eSDK Enterprise Storage Plugins 2.3.RC2

OpenStack Cinder Driver Configuration Guide

Issue 01

Date 2021-09-16





Copyright © Huawei Technologies Co., Ltd. 2021. 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

HUAWEI 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

Contents

i Overview	I
2 Version Mappings	2
2.1 Version Mappings Among Cinder Driver, Huawei Storage, and OpenStack	2
2.2 Mappings Among Cinder Driver, Features, and OpenStack Versions	3
2.3 Mappings Between Cinder Driver and Storage Licenses	5
3 Specifications and Restrictions	7
4 Deployment	11
4.1 Obtaining Cinder Driver	
4.2 Deploying Cinder Driver for the Cinder Volume Service in Non-Containerized Mode	12
4.3 Deploying Cinder Driver for the Cinder Volume Service in Containerized Mode	13
5 Upgrade	14
5.1 Upgrading Cinder Driver for the Cinder Volume Service in Non-Containerized Mode	
5.2 Upgrading Cinder Driver for the Cinder Volume Service in Containerized Mode	15
6 Configuring Basic Properties	17
6.1 Configuring Cinder Driver in Non-Containerized Mode	17
6.1.1 Configuring Cinder Driver for iSCSI-based Storage	17
6.1.2 Configuring Cinder Driver for FC-based Storage	22
6.2 Configuring Cinder Driver in Containerized Mode	26
6.2.1 Configuring Cinder Driver for iSCSI-based Storage	26
6.2.2 Configuring Cinder Driver for FC-based Storage	27
7 Configuring the Multipathing Software	28
7.1 Configuring Host Multipathing	28
7.1.1 Installing the Multipathing Tool Package	28
7.1.2 Configuring the Multipathing Service	29
7.2 Configuring OpenStack Multipathing	29
7.2.1 Configuring Multipathing for OpenStack Nova Nodes	29
7.2.2 Configuring Multipathing for OpenStack Cinder Nodes	30
8 Configuring Advanced Properties	31
8.1 Configuring the Thick Property	32
8.2 Configuring the Thin Property	32

8.3 Configuring the SmartQoS Property	33
8.4 Configuring the SmartPartition Property	34
8.5 Configuring the SmartCache Property	34
8.6 Configuring the SmartTier Property	35
8.7 Configuring the HyperMetro Property	35
8.8 Configuring the Remote Replication Property	37
8.9 Creating a LUN in a Specified Storage Pool	39
8.10 Creating a LUN on a Disk of a Specified Type	39
8.11 Creating a LUN in on a Specified Backend	39
8.12 Configuring a Backup Snapshot	40
8.13 Configuring Certificate Authentication	40
8.14 Configuring the Owning Controller of a LUN	41
8.15 Configuring the SmartDedupe Property	41
8.16 Configuring the SmartCompression Property	42
8.17 Creating a Volume Using fastclone	43
8.18 Configuring the LUN Copy Speed	43
8.19 Configuring the Application Type of a LUN	44
8.20 Configuring a Tenant User	44
8.21 Configuring Multiple Backends	45
8.22 Configuring Auto Zoning	45
8.23 Configuring CHAP	46
9 Best Practices	47
9.1 Configuring Basic Properties for Interconnection Between iSCSI and OceanStor Dorado V3	47
9.2 Initiator Configuration Policy Example	49
10 FAQs	51
10.1 Do I Need to Create Hosts, Host Groups, LUN Groups, and Other Associated Objects on Huawei Storage Before Using Cinder Driver to Map Volumes?	
10.2 What Can I Do if Mapping Fails Because Nova Logs Indicate that the lsblk Command Fails to Be Executed During Volume Mapping?	
10.3 Failed to Create Volumes From Images When verify_glance_signatures=enabled	52
10.4 There Are Some "alua not supported" Errors When You Run the "multipath -ll" Command	53
10.5 Failed to Delete or Live Migrate a VM	53
10.6 After a Volume is Deleted from OpenStack, the Volume Enters the Recycle Bin. When You Try to Create the Volume Again, the Volume Fails to Be Created Due to Duplicate Names	54
10.7 How Do I Manually Map LUNs to an OpenStack Host Without Affecting Cinder Driver's Volume Mapping?	55
11 Appendix	. 57
11.1 Appendix 1: ALUA Parameter Description	57
11.1.1 ALUA Parameter Description for Converged Storage	57
11.1.2 ALUA Parameter Description for OceanStor Dorado V6 Storage	59
11.2 Appendix 2: Configuring Volume-backend image	61
11.3 Appendix 3: Configuring Image-Volume Cache	62

1 Overview

Huawei Cinder Driver is a plug-in that is deployed on the OpenStack Cinder module. The plug-in connects Huawei storage to provide the volume storage capability for the OpenStack platform. Cinder Driver supports iSCSI and Fibre Channel (FC) protocols.

2 Version Mappings

This chapter describes the version mappings among Cinder Driver, Huawei storage, and OpenStack as well as mappings between Cinder Driver and storage licenses.

- 2.1 Version Mappings Among Cinder Driver, Huawei Storage, and OpenStack
- 2.2 Mappings Among Cinder Driver, Features, and OpenStack Versions
- 2.3 Mappings Between Cinder Driver and Storage Licenses

2.1 Version Mappings Among Cinder Driver, Huawei Storage, and OpenStack

Table 2-1 Version mappings among Cinder Driver, Huawei storage, and OpenStack

OpenStack	Huawei Storage
Mitaka/Rocky/Stein/Train/ Ussuri/Victoria/Wallaby	 OceanStor Dorado V6 6.0.0/6.0.1/6.1.0/6.1.2 OceanStor Dorado V3 V300R001/V300R002
	OceanStor F V5/V5 V500R007/V500R007 Kunpeng
	OceanStor F V3/V3 V300R003/V300R006

□ NOTE

Huawei only maintains the drivers for the latest 6 stable OpenStack versions. Though the earlier versions can still be retrieved from Huawei warehouse website, they are out of maintenance. Be wary of using them.

2.2 Mappings Among Cinder Driver, Features, and OpenStack Versions

Table 2-2 Mappings among Cinder Driver, features, and OpenStack versions (√: supported, x: not supported)

Feature	Normal LUN	HyperMetro LUN	Remarks
OpenStack	Mitaka/Rocky/ Stein/Train/ Ussuri/Victoria/ Wallaby	Mitaka/Rocky/ Stein/Train/ Ussuri/Victoria/ Wallaby	The following features are applicable to the scenario where HyperMetro storage is normal.
Create Volume	√	√	N/A
Delete Volume	√	√	N/A
Attach Volume	√	√	N/A
Detach Volume	√	√	N/A
Extend Volume	√	√	N/A
Create Snapshot	√	√	HyperMetro LUN supports snapshot creation only on the local end of HyperMetro.
Delete Snapshot	√	√	HyperMetro LUN supports snapshot deletion only on the local end of HyperMetro.
Create Volume from Snapshot	√	√	HyperMetro LUN supports snapshot-based volume creation only on the local end of HyperMetro.
Create Volume from Image	√	√	N/A
Create Volume from Volume	√	√	N/A
Create Image from Volume	√	√	N/A
Volume Migration	√	х	N/A
Thin Volume	√	√	N/A
Thick Volume	√	√	N/A
QoS	√	√	HyperMetro LUN supports SmartQoS configuration only on the local end of HyperMetro.

Feature	Normal LUN	HyperMetro LUN	Remarks
FC zoning	√	√	N/A
Manage/Unmanage Volume	√	х	N/A
Manage/Unmanage Snapshot	√	х	N/A
Replication V2.1	√	х	N/A
Backup Snapshot	√	√	HyperMetro LUN supports snapshots backup only on the local end of HyperMetro.
Revert to Snapshot	√ (not supported by Mitaka)	X	N/A
Retype	√	х	N/A
Multipath	√	√	N/A
Multi-Attach	√	√	N/A
Consistency Group	√	x	N/A
Snapshot Consistency Group	√	√	HyperMetro LUN supports snapshot creation only on the local end of HyperMetro.
HyperMetro Consistency Group	х	√	N/A
Replication Consistency Group	√	х	N/A
SmartTier	√	√	HyperMetro LUN supports SmartTier configuration only on the local end of HyperMetro.
SmartCache	√	√	HyperMetro LUN supports SmartCache configuration only on the local end of HyperMetro.
SmartPartition	√	√	HyperMetro LUN supports SmartPartition configuration only on the local end of HyperMetro.
Create a volume on a certain storage pool	√	х	N/A
Create a volume with certain disk type	√	х	N/A

NOTICE

- OceanStor Dorado does not support Thick Volume, SmartTier, SmartCache, and SmartPartition.
- Entry-level storage does not support some value-added features. For details, refer to the specifications of the corresponding storage model.

2.3 Mappings Between Cinder Driver and Storage Licenses

Table 2-3 Mappings between Cinder Driver and storage licenses

Feature	Storage License
Create Volume	N/A
Delete Volume	N/A
Attach Volume	N/A
Detach Volume	N/A
Extend Volume	N/A
Create Snapshot	HyperSnap
Delete Snapshot	N/A
Create Volume from Snapshot	HyperCopy/HyperClone
Create Volume from Image	HyperSnap&(HyperCopy/HyperClone)
Create Volume from Volume	HyperSnap&(HyperCopy/HyperClone)
Create Image from Volume	N/A
Volume Migration	SmartMigration
QoS	SmartQoS
FC zoning	N/A
SmartTier	SmartTier
SmartCache	SmartCache
Thin Volume	SmartThin
Thick Volume	N/A
SmartPartition	SmartPartition
HyperMetro	HyperMetro
Retype	N/A

Feature	Storage License
Manage/Unmanage Volume	N/A
Manage/Unmanage Snapshot	N/A
Replication V2.1	HyperReplication
Create a volume on a certain storage pool	N/A
Create a volume with certain disk type	N/A
Backup Snapshot	HyperCopy/HyperClone
Revert to Snapshot	HyperSnap
Multipath	N/A
Multi-Attach	N/A
Consistency Group	N/A
Snapshot Consistency Group	HyperSnap
HyperMetro Consistency Group	HyperMetro
Replication Consistency Group	HyperReplication

3 Specifications and Restrictions

Feature	Sub-feature	Description	Remarks
Mapping platform	Native OpenStack platform	OpenStack versions: Rocky/Stein/Train/ Ussuri/Victoria/ Wallaby	 Resources created using OpenStack, such as hosts, host groups, LUNs, LUN groups, snapshots, mapping views, QoS, and HyperMetro pairs, cannot be shared with other platform applications or manually modified. The storage access address, user name, password, and storage pool information provided for OpenStack cannot be modified.
Configur ation	XML configuration file	Ensure that the storage pool used for configuring Cinder Driver exists on Huawei storage. Otherwise, create a storage pool. In addition, the type of this storage pool must be block storage service.	
		For an FC network, the sg tool must be installed so that storage resources can be used.	

Feature	Sub-feature	Description	Remarks
		All parameter values in the Cinder Driver configuration file cannot contain the following special characters: <>&'''	
		Set Product based on the actual backend storage model.	
		The value of HostName in the Initiator configuration item is a regular expression.	For details about the configuration method, visit Regular expression. When HostName is set to *, the common configuration is used and takes effect on the host with any name. When HostName is set to another value, the general configuration is used. Configure policy rules: Priority: General host name configuration > Common host name configuration Use the first configuration item (for the general host name configuration).
	Multipathing configuration	Host multipathing: Manually install the multipathing software and enable the multipathing service.	
		OpenStack multipathing: OpenStack multipathing is disabled by default. You need to manually modify Nova and Cinder configurations.	
	LUN type	OceanStor Dorado supports only thin volumes.	

Feature	Sub-feature	Description	Remarks
Policy	SmartQoS	maxBandWidth or maxIOPS cannot coexist with minBandWidth, minIOPS, or latency.	 Limit policies: maxIOPS and maxBandWidth Protection policies: minIOPS, minBandWidth, and
		For OceanStor Dorado, IOType must be set to 2 and minBandWidth, minIOPS, and latency are unavailable.	latency
Data copy	fastclone	Some storage devices support fastclone.	 OceanStor Dorado V6: 6.1.0 and later OceanStor Dorado V3: V300R001C20 and later When fastclone is used, there are restrictions on HyperMetro, remote replication, and snapshots.
	Full data copy	The copy speed can be configured. The default LUN copy speed is Medium .	 Low: 0 MB/s to 5 MB/s Medium: 10 MB/s to 20 MB/s High: 50 MB/s to 70 MB/s Highest: over 100 MB/s
License	Creating a snapshot	The storage HyperSnap license is required.	
	Creating a volume with a QoS policy	The storage SmartQoS license is required.	
	Creating a HyperMetro volume	The storage HyperMetro license is required.	
	Creating a volume using a snapshot	The storage HyperCopy/ HyperClone license is required.	
	Creating a mirrored volume	The storage HyperSnap&(HyperC opy/HyperClone) license is required.	

Feature	Sub-feature	Description	Remarks
	Creating a clone volume	The storage HyperSnap&(HyperC opy/HyperClone) license is required.	
RESTful request	Maximum number of RESTful requests sent by a Cinder driver to storage at the same time	20	

4 Deployment

NOTICE

- Resources created using OpenStack, such as hosts, host groups, LUNs, LUN groups, snapshots, mapping views, QoS, and HyperMetro pairs, cannot be shared with other platform applications or manually modified.
- The storage access address, user name, password, and storage pool information provided for OpenStack cannot be modified.
- 4.1 Obtaining Cinder Driver
- 4.2 Deploying Cinder Driver for the Cinder Volume Service in Non-Containerized Mode
- 4.3 Deploying Cinder Driver for the Cinder Volume Service in Containerized Mode

4.1 Obtaining Cinder Driver

You can obtain Cinder Driver in either of the following ways:

- OpenStack community warehouse. Since the Kilo version, Cinder Driver has been integrated into the OpenStack community warehouse. After OpenStack is installed, Cinder Driver is provided. Cinder Driver is stored in the ../cinder/ cinder/volume/drivers/huawei directory.
- Huawei OpenStack Driver warehouse
- **Step 1** Open a browser and enter https://github.com/Huawei/OpenStack_Driver in the address box.
- **Step 2** Click the **Download ZIP** button to download the Cinder Driver package to a local host.
- **Step 3** Decompress the package.
- **Step 4** In the decompressed directory, find the **Cinder** directory, which contains Cinder Driver of multiple OpenStack versions.

----End

- After OpenStack community versions are released, new features cannot be integrated.
 The bug fixing period is long and risks exist. Huawei OpenStack Driver warehouse
 versions ensure that new features are released and bugs are fixed in time. These
 versions are more stable than OpenStack community versions.
- The OpenStack community maintains only two stable versions. Huawei OpenStack
 Driver warehouse maintains six stable versions, ensuring long-term stable running of
 historical versions.
- It is strongly recommended that OpenStack community versions be replaced by Huawei OpenStack Driver warehouse versions.

4.2 Deploying Cinder Driver for the Cinder Volume Service in Non-Containerized Mode

Step 1 Search for the original Huawei Driver code in the system. The relative path is .../ cinder/volume/drivers/huawei.

The absolute path may vary in different systems. You can obtain the Cinder Driver code directory by running the following commands:

```
# python -c "from cinder.volume.drivers import huawei; print (huawei.__path__)"
['/usr/lib/python2.7/site-packages/cinder/volume/drivers/huawei']
```

The driver path is:

/usr/lib/python2.7/site-packages/cinder/volume/drivers/huawei/

Alternatively, run the following commands:

```
# python3 -c "from cinder.volume.drivers import huawei; print (huawei.__path__)"
['/usr/lib/python3.7/site-packages/cinder/volume/drivers/huawei']
```

The driver path is:

/usr/lib/python3.7/site-packages/cinder/volume/drivers/huawei/

- **Step 2** Delete all Huawei Driver code files in the directory returned in **Step 1**.
- **Step 3** Copy all Huawei Driver code files of the corresponding version to the directory returned in **Step 1**.

```
# ls -l
-rw-r--r-- 1 root root 4787 Mar 4 23:40 constants.py
-rw-r--r-- 1 root root 10316 Mar 4 23:40 fc_zone_helper.py
-rw-r--r-- 1 root root 17905 Mar 4 23:40 huawei_conf.py
-rw-r--r-- 1 root root 125657 Mar 4 23:48 huawei_driver.py
-rw-r--r-- 1 root root 24276 Mar 4 23:40 huawei_t.py
-rw-r--r-- 1 root root 6698 Mar 4 23:40 huawei_utils.py
-rw-r--r-- 1 root root 18437 Mar 4 23:40 huawei_utils.py
-rw-r--r-- 1 root root 38948 Mar 4 23:40 replication.py
-rw-r--r-- 1 root root 97236 Mar 4 23:40 rest_client.py
-rw-r--r-- 1 root root 8834 Mar 4 23:40 smartx.py
```

```
Or
```

```
# ls -l
-rw-r--r-- 1 root root 5631 Mar 18 11:43 constants.py
-rw-r--r-- 1 root root 33401 Mar 18 11:43 huawei_base_driver.py
-rw-r--r-- 1 root root 21773 Mar 18 11:43 huawei_conf.py
-rw-r--r-- 1 root root 12116 Mar 18 11:43 huawei_driver.py
-rw-r--r-- 1 root root 99727 Mar 18 11:43 huawei_flow.py
-rw-r--r-- 1 root root 19599 Mar 18 11:43 huawei_utils.py
-rw-r--r-- 1 root root 14733 Mar 18 11:43 hypermetro.py
```

```
-rw-r--r-- 1 root root 23 Mar 18 11:43 __init__.py
-rw-r--r-- 1 root root 20056 Mar 18 11:43 replication.py
-rw-r--r-- 1 root root 67292 Mar 18 11:43 rest_client.py
-rw-r--r-- 1 root root 5082 Mar 18 11:43 smartx.py
```

----End

4.3 Deploying Cinder Driver for the Cinder Volume Service in Containerized Mode

Step 1 Log in to the node with the Cinder Volume container image. Run the **docker save** command to back up the Cinder Volume container image.

Ⅲ NOTE

To obtain the image version, run the docker image ls command.

- **Step 2** Create temporary directory **build** in any directory and go to the created directory: # mkdir build;cd build
- **Step 3** Copy the obtained Cinder Driver code files to the current directory:

Step 4 Create the Dockerfile file in the current directory and edit the following content:

```
FROM ***:***
COPY *.py /huawei/cinder/driver/path/
```

□ NOTE

- Replace ***:*** with the name and version of the original Cinder Volume container image.
- Replace /huawei/cinder/driver/path/ with the path of Cinder Driver in the container.
 You can obtain the Cinder Driver code directory by running the following commands:
 # python -c "from cinder.volume.drivers import huawei; print (huawei.__path__)"
 ['/usr/lib/python2.7/site-packages/cinder/volume/drivers/huawei']
- **Step 5** Run the following command to build an image:

```
docker build -t ***:*** .
```

Replace ***:*** with the name and version of the original Cinder Volume container image.

----End

5 Upgrade

NOTICE

- 1. For details about how to upgrade OpenStack Cinder Driver, see this section. If you install and deploy Cinder Driver for the first time, skip this section.
- 2. During the upgrade, do not perform other operations on the management plane, such as VM creating and LUN creating.
- 5.1 Upgrading Cinder Driver for the Cinder Volume Service in Non-Containerized Mode
- 5.2 Upgrading Cinder Driver for the Cinder Volume Service in Containerized Mode

5.1 Upgrading Cinder Driver for the Cinder Volume Service in Non-Containerized Mode

Step 1 Search for the original Huawei Driver code in the system. The relative path is .../ cinder/volume/drivers/huawei.

■ NOTE

The absolute path may vary in different systems. You can obtain the Cinder Driver code directory by running the following commands:

python -c "from cinder.volume.drivers import huawei; print (huawei.__path__)" ['/usr/lib/python2.7/site-packages/cinder/volume/drivers/huawei']

The driver path is:

/usr/lib/python2.7/site-packages/cinder/volume/drivers/huawei/

Alternatively, run the following commands:

python3 -c "from cinder.volume.drivers import huawei; print (huawei.__path__)" ['/usr/lib/python3.7/site-packages/cinder/volume/drivers/huawei']

The driver path is:

/usr/lib/python3.7/site-packages/cinder/volume/drivers/huawei/

- **Step 2** Back up all Huawei Driver code files in the directory returned in **Step 1**.
- **Step 3** Copy all Huawei Driver code files of the corresponding version to the directory returned in **Step 1**.

Step 4 Restart the Cinder Volume service.

systemctl restart openstack-cinder-volume.service

----End

NOTE

If the upgrade fails, restore the code backed up in **Step 2** and restart the Cinder Volume service

5.2 Upgrading Cinder Driver for the Cinder Volume Service in Containerized Mode

Step 1 Log in to the node with the Cinder Volume container image. Run the **docker save** command to back up the Cinder Volume container image.

To obtain the image version, run the **docker image Is** command.

- **Step 2** Create temporary directory **build** in any directory and go to the created directory: # mkdir build;cd build
- **Step 3** Copy the obtained Cinder Driver code files to the current directory:

Step 4 Create the **Dockerfile** file in the current directory and edit the following content:

```
FROM ***:***
COPY *.py /huawei/cinder/driver/path/
```

□ NOTE

- Replace ********* with the name and version of the original Cinder Volume container image.
- Replace /huawei/cinder/driver/path/ with the path of Cinder Driver in the container.
 You can obtain the Cinder Driver code directory by running the following commands:

python -c "from cinder.volume.drivers import huawei; print (huawei.__path__)" ['/usr/lib/python2.7/site-packages/cinder/volume/drivers/huawei']

Step 5 Run the following command to build an image:

docker b	ouild -t ***:*** .
₩ N	IOTE
Re	place ***:*** with the name and version of the original Cinder Volume container image.
En	d
₩ N	IOTE
If t	the upgrade fails, restore the container image backed up in Step 1.

6 Configuring Basic Properties

This chapter describes how to configure Huawei Cinder Driver for iSCSI-and FC-based storage.

NOTICE

- Ensure that the storage pool used for configuring Cinder Driver exists on Huawei storage. Otherwise, create a storage pool. In addition, the type of this storage pool must be block storage service.
- For an FC network, the sg tool must be installed so that storage resources can be used.
- All parameter values in the Cinder Driver configuration file cannot contain the following special characters: <>&'"
- When fastclone is used, there are restrictions on HyperMetro, remote replication, and snapshots.

6.1 Configuring Cinder Driver in Non-Containerized Mode

6.2 Configuring Cinder Driver in Containerized Mode

6.1 Configuring Cinder Driver in Non-Containerized Mode

6.1.1 Configuring Cinder Driver for iSCSI-based Storage

This section describes how to configure Cinder Driver for iSCSI-based storage products.

- **Step 1** In /etc/cinder, create a Huawei Cinder Driver configuration file in .xml format. You can customize the configuration file name, for example, cinder_huawei_conf.xml.
- **Step 2** Configure mandatory parameters in the Cinder Driver configuration file.

<?xml version='1.0' encoding='UTF-8'?>
<confia>

```
<Storage>
    <Product>***</Product>
    <Protocol>iSCSI</Protocol>
    <RestURL>https://*.**.*:8088/deviceManager/rest/;https://*.**.*:8088/deviceManager/rest/</RestURL>
<UserName>***</UserName>
    <UserPassword>***</UserPassword>
  </Storage>
  <LUN>
    <StoragePool>***</StoragePool>
    <LUNType>***</LUNType>
    <LUNCopySpeed>***</LUNCopySpeed>
  </LUN>
  <iSCSI>
    <Initiator HostName="****** ALUA="*" FAILOVERMODE="*" SPECIALMODETYPE="*" PATHTYPE="*"
TargetIP="x.x.x.x y.y.y.y"/>
   <Initiator HostName="******" ALUA="*" FAILOVERMODE="*" SPECIALMODETYPE="*" PATHTYPE="*"</pre>
TargetIP="x.x.x.x y.y.y.y"/>
 </iSCSI>
</config>
```

Name can be configured in <iSCSI> for OceanStor V3, V5, and Dorado V3 storage.

```
...
<iSCSI>
<initiator Name="*****" ALUA="*" FAILOVERMODE="*" SPECIALMODETYPE="*" PATHTYPE="*"
TargetIP="x.x.x.x y.y.y.y"/>
</iSCSI>
...
```

The ALUA policy of OceanStor Dorado V6 is different. Only **HostName** can be configured for OceanStor Dorado V6. See the following.

```
...
<iSCSI>
<Initiator HostName="******" ACCESSMODE="*" HYPERMETROPATHOPTIMIZED="*" TargetIP="x.x.x.x
y.y.y.y"/>
<Initiator HostName="******" ACCESSMODE="*" HYPERMETROPATHOPTIMIZED="*" TargetIP="x.x.x.x
y.y.y.y"/>
</iSCSI>
...
```

Table 6-1 Mandatory parameters

Parameter	Description
Product	Type of a storage product. The value can be V3 , V5 , or Dorado .
Protocol	Type of a connection protocol. Set this parameter to iSCSI for iSCSI scenarios.
RestURL	Access address of the REST interface. If you need to configure multiple RestURL parameters, separate them by semicolons (;).
UserName	User name of a storage administrator.
UserPassword	Password of the storage administrator.
StoragePool	Name of a storage pool to be used. If you need to configure multiple storage pools, separate them by semicolons (;).
LUNType	Type of a LUN. The value can be Thick or Thin . Dorado supports only Thin .
LUNCopySpee d	LUN copy speed. The value can be 1 (low speed), 2 (medium speed), 3 (high speed), or 4 (highest speed).

Parameter	Description
Initiator	The value of HostName is the host name of the node where Nova-compute or Cinder-volume resides. You can run the cat /etc/hostname command to obtain the value. Regular expression matching is supported. For details about the configuration rules and priorities, see the notice.
	The value of Name is the host iSCSI initiator name of the node where Nova-compute or Cinder-volume resides. You can run the cat /etc/iscsi/initiatorname.iscsi command to obtain the value.
	Note: OceanStor V3, V5, and Dorado V3 storage systems support Name or HostName (not configured at the same time). OceanStor Dorado V6 support only HostName. HostName is recommended.
	For details about ALUA, FAILOVERMODE, SPECIALMODETYPE, PATHTYP, ACCESSMODE, HYPERMETROPATHOPTIMIZED storage ALUA policy, see 11.1 Appendix 1: ALUA Parameter Description. TargetIP indicates the IP address of the iSCSI target port. You can enter multiple IP addresses and separate them with spaces. This parameter can be set for different hosts.

NOTICE

- 1. Set **Product** based on the actual back-end storage model.
- 2. The value of **HostName** in the **Initiator** configuration item is a regular expression. For details about the configuration method, visit **Regular** expression.

When **HostName** is set to *, the common configuration is used and takes effect on the host with any name. When **HostName** is set to another value, the general configuration is used. When you configure **HostName**, the number of host connections is limited. For details, see the **Maximum number of iSCSI connections per controller enclosure** specification item in **Specifications Query**. If the number of host connections is less than or equal to the upper limit, the common configuration is recommended. If the number of host connections exceeds the upper limit, the general configuration is used.

Configure policy rules:

- Priority: General host name configuration > Common host name configuration.
 For details, see Example 1 in 9.2 Initiator Configuration Policy Example.
- Use the first configuration item (for the general host name configuration). For details, see **Example 2** in **9.2 Initiator Configuration Policy Example**.
- Match a host exactly (for the general host name configuration). For details, see
 Example 5 in 9.2 Initiator Configuration Policy Example.

Step 3 (Optional) Configure optional parameters in the Cinder Driver configuration file.

```
<LUN>
    <WriteType>***</WriteType>
    <Prefetch Type="***" Value="***"/>
```

<LUNcopyWaitInterval>***</LUNcopyWaitInterval>
<Timeout>***</Timeout>
<LUNCloneMode>***</LUNCloneMode>
<HyperSyncSpeed>***</HyperSyncSpeed>
<ReplicaSyncSpeed>***</ReplicaSyncSpeed>
<ForceDeleteVolume>***</ForceDeleteVolume>
<HyperEnforceMultipath>***</HyperEnforceMultipath>
<SnapshotRollbackSpeed>
</LUN>

Table 6-2 Optional parameters

Parameter	Default Value	Description
WriteType	1	Cache write type. The value can be 1 (write back) or 2 (write through). For OceanStor Dorado, the value must be 1.
Prefetch Type	3	Cache prefetch policy. The value can be 0 (no prefetch), 1 (fixed prefetch), 2 (variable prefetch), or 3 (intelligent prefetch).
Prefetch Value	0	Cache prefetch value.
LUNcopyWaitInterval	5	After LUN copy is enabled, the plug- in frequently queries the LUN copy progress. You can set a value to specify the query interval.
Timeout	2592000	Timeout interval for waiting the LUN copy of a storage device to complete. The unit is second.
LUNCloneMode	luncopy	LUN clone mode. The value can be fastclone or luncopy. OceanStor Dorado V300R001C20 and later versions and OceanStor Dorado 6.1.0 and later versions support fastclone and luncopy. Other storage devices support only luncopy.
HyperSyncSpeed	2	HyperMetro pair synchronization speed. The value can be 1 (low), 2 (medium), 3 (high), or 4 (highest).
ReplicaSyncSpeed	2	Replication pair synchronization speed. The value can be 1 (low), 2 (medium), 3 (high), or 4 (highest).
ForceDeleteVolume	False	When deleting a LUN, if the LUN is in the mapping view, whether to delete it forcibly. Optional values: "True" (forcibly), "False" (not forcibly)

Parameter	Default Value	Description
HyperEnforceMultipath	True	Whether to enable multipathing in HyperMetro scenarios. The default value is True .
SnapshotRollbackSpeed	3	Snapshot rollback speed. The value can be 1 (low speed), 2 (medium speed), 3 (high speed), or 4 (highest speed).

Step 4 Change the owner and user group of the Huawei Cinder Driver configuration file to be the same as those of the /etc/cinder/cinder.conf file.

```
-rw-r--r-- 1 cinder cinder 2662 Jul 29 02:13 cinder.conf
-rw-r--r-- 1 cinder cinder 778 Jul 30 02:56 cinder_huawei_conf.xml
```

Step 5 Configure the /etc/cinder/cinder.conf file. Add the following backend configuration to the end of the file:

[huawei_backend] volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiISCSIDriver cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml volume_backend_name = huawei_backend retain_storage_mapping = False

- Backend name huawei_backend can be customized.
- **volume_driver** indicates the type of the driver to be used. Set this parameter to **cinder.volume.drivers.huawei.huawei driver.HuaweiISCSIDriver**.
- **cinder_huawei_conf_file** indicates the path of the Cinder Driver configuration file.
- **retain_storage_mapping** indicates whether to retain the storage mapping when the last volume on the host is not mapped. **False** indicates no and **True** indicates yes. This parameter is optional. The default value is **False**.

In the [DEFAULT] area, set enabled_backends to huawei_backend.

[DEFAULT]

enabled_backends=huawei_backend

Step 6 If the Cinder Volume service is deployed on multiple nodes, you need to configure a distributed lock in the [coordination] area. For details about <user>, <password>, <host>, and <database>, see the value of connection under [database] in the configuration file.

[coordination]

backend_url = mysql://<user>:<password>@<host>/<database>?charset=utf8

For example, [database] in the configuration file is configured as follows:

[database]

connection = mysql+pymysql://cinder:302824058e9a4f31@127.0.0.1/cinder

Then configure [coordination] as follows:

[coordination]

backend_url = mysql://cinder:302824058e9a4f31@127.0.0.1/cinder

Step 7 Restart the Cinder Volume service.

systemctl restart openstack-cinder-volume.service

----End

6.1.2 Configuring Cinder Driver for FC-based Storage

This section describes how to configure Cinder Driver for FC-based storage products.

- **Step 1** In /etc/cinder, create a Huawei Cinder Driver configuration file in .xml format. You can customize the configuration file name, for example, cinder_huawei_conf.xml.
- **Step 2** Configure mandatory parameters in the Cinder Driver configuration file.

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
  <Storage>
   <Product>***</Product>
   <Protocol>FC</Protocol>
   <RestURL>https://*.*.*:8088/deviceManager/rest/;https://*.*.*:8088/deviceManager/rest/</RestURL>
   <UserName>***</UserName>
   <UserPassword>***</UserPassword>
 </Storage>
  <LUN>
   <StoragePool>***</StoragePool>
   <LUNType>***</LUNType>
   <LUNCopySpeed>***</LUNCopySpeed>
  </LUN>
  <FC>
   <MinOnlineFCInitiator>xx</MinOnlineFCInitiator>
   <Initiator HostName="xxx" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
   <Initiator HostName="xxx" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
  </FC>
</config>
```

Name can be configured in <FC> for OceanStor V3, V5, and Dorado V3 storage.

```
...
<FC>
<MinOnlineFCInitiator>xx</MinOnlineFCInitiator>
<Initiator Name="xxx" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
</FC>
...
```

The ALUA policy of OceanStor Dorado V6 is different. Only **HostName** can be configured for OceanStor Dorado V6. See the following.

Table 6-3 Mandatory parameters

Parameter	Description
Product	Type of a storage product. The value can be V3 , V5 , or Dorado .
Protocol	Type of a connection protocol. Set this parameter to FC for FC scenarios.

Parameter	Description	
RestURL	Access address of the REST interface. If you need to configure multiple RestURL parameters, separate them by semicolons (;).	
UserName	User name of a storage administrator.	
UserPassword	Password of the storage administrator.	
StoragePool	Name of a storage pool to be used.	
LUNType	Type of a LUN. The value can be Thick or Thin . Dorado supports only Thin .	
LUNCopySpeed	LUN copy speed. The value can be 1 (low speed), 2 (medium speed), 3 (high speed), or 4 (highest speed).	
MinOnlineFCInitiator	Minimum available FC initiators. The type is Arabic numerals. The Default value means that all FC initiators are required to be online.	
Initiator	The value of HostName is the host name of the node where Nova-compute or Cinder-volume resides. You can run the cat /etc/hostname command to obtain the value. Regular expression matching is supported. For details about the configuration rules and priorities, see the notice.	
	The value of Name is the host FC initiator WWN of the node where Nova-compute or Cinder-volume resides. You can run the cat /sys/class/fc_host/*/port_name command to obtain the value.	
	Note : OceanStor V3, V5, and Dorado V3 storage systems support Name or HostName (not configured at the same time). OceanStor Dorado V6 support only HostName . HostName is recommended.	
	For details about ALUA, FAILOVERMODE, SPECIALMODETYPE, PATHTYP, ACCESSMODE, HYPERMETROPATHOPTIMIZED storage ALUA policy, see 11.1 Appendix 1: ALUA Parameter Description. This parameter can be set for different hosts.	

NOTICE

- 1. Set **Product** based on the actual back-end storage model.
- 2. The value of HostName in the Initiator configuration item is a regular expression. For details about the configuration method, visit **Regular expression**.

When **HostName** is set to *, the common configuration is used and takes effect on the host with any name. When **HostName** is set to another value, the general configuration is used. When you configure **HostName**, the number of host connections is limited. For details, see the **Maximum number of Fibre Channel connections per controller enclosure** specification item in **Specifications Query**. If the number of host connections is less than or equal to the upper limit, the common configuration is recommended. If the number of host connections exceeds the upper limit, the general configuration is used.

Configure policy rules:

- Priority: General host name configuration > Common host name configuration For details, see **Example 3** in **9.2 Initiator Configuration Policy Example**.
- Use the first configuration item (for the general host name configuration). For details, see **Example 4** in **9.2 Initiator Configuration Policy Example**.
- Match a host exactly (for the general host name configuration). For details, see **Example 6** in **9.2 Initiator Configuration Policy Example**.

Step 3 (Optional) Configure optional parameters in the Cinder Driver configuration file.

```
<LUN>
<WriteType>***</WriteType>
<Prefetch Type="***" Value="***"/>
<LUNcopyWaitInterval>***</LUNcopyWaitInterval>
<Timeout>***</Timeout>
<LUNCloneMode>***</LUNCloneMode>
<HyperSyncSpeed>***</HyperSyncSpeed>
<ReplicaSyncSpeed>***</ReplicaSyncSpeed>
<ForceDeleteVolume>***</ForceDeleteVolume>
<HyperEnforceMultipath>***</HyperEnforceMultipath>
<SnapshotRollbackSpeed>***</SnapshotRollbackSpeed>
</LUN>
```

Table 6-4 Optional parameters

Parameter	Default Value	Description
WriteType	1	Cache write type. The value can be 1 (write back) or 2 (write through).
Prefetch Type	3	Cache prefetch policy. The value can be 0 (no prefetch), 1 (fixed prefetch), 2 (variable prefetch), or 3 (intelligent prefetch).
Prefetch Value	0	Cache prefetch value.
LUNcopyWaitInterval	5	After LUN copy is enabled, the plug-in frequently queries the copy progress. You can set a value to specify the query interval.

Parameter	Default Value	Description
Timeout	2592000	Timeout period for waiting LUN copy of an array to complete, in seconds
LUNCloneMode	luncopy	LUN clone mode. The value can be fastclone or luncopy .
		OceanStor Dorado V300R001C20 and later versions and OceanStor Dorado 6.1.0 and later versions support fastclone and luncopy . Other storage devices support only luncopy .
HyperSyncSpeed	2	HyperMetro pair synchronization speed. The value can be 1 (low speed), 2 (medium speed), 3 (high speed), or 4 (highest speed).
ReplicaSyncSpeed	2	Remote replication pair synchronization speed. The value can be 1 (low speed), 2 (medium speed), 3 (high speed), or 4 (highest speed).
ForceDeleteVolume	False	When deleting a LUN, if the LUN is in the mapping view, whether to delete it forcibly. Optional values: "True" (forcibly), "False" (not forcibly)
HyperEnforceMulti- path	True	Whether to enable multipathing in HyperMetro scenarios. The default value is True .
SnapshotRollback- Speed	3	Snapshot rollback speed. The value can be 1 (low speed), 2 (medium speed), 3 (high speed), or 4 (highest speed).

Step 4 Change the owner and user group of the Huawei Cinder Driver configuration file to be the same as those of the /etc/cinder/cinder.conf file.

-rw-r--r-- 1 cinder cinder 2662 Jul 29 02:13 cinder.conf -rw-r--r-- 1 cinder cinder 778 Jul 30 02:56 cinder_huawei_conf.xml

Step 5 Configure the /etc/cinder/cinder.conf file. Add the following backend configuration to the end of the file:

[huawei_backend]
volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiFCDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml
volume_backend_name = huawei_backend
retain_storage_mapping = False

□ NOTE

- Backend name huawei backend can be customized.
- **volume_driver** indicates the type of the driver to be used. Set this parameter to **cinder.volume.drivers.huawei.huawei_driver.HuaweiFCDriver**.
- **cinder_huawei_conf_file** indicates the path of the Cinder Driver configuration file.
- **retain_storage_mapping** indicates whether to retain the storage mapping when the last volume on the host is not mapped. **False** indicates no and **True** indicates yes. This parameter is optional. The default value is **False**.

In the **DEFAULT** area, set **enabled_backends** to **huawei_backend**.

[DEFAULT]

enabled backends=huawei backend

If Cinder Volume is deployed on multiple nodes, you can configure distributed locks in the **coordination** area. For details about **<user>**, **<password>**, **<host>**, and **<database>**, see the value of **connection** in **[database]** in the configuration file.

```
[coordination] backend_url = mysql://<user>:<password>@<host>/<database>?charset=utf8
```

Step 6 Restart the Cinder Volume service.

systemctl restart openstack-cinder-volume.service

----End

6.2 Configuring Cinder Driver in Containerized Mode

This section describes how to configure Cinder Driver for Huawei storage products in container-based deployment. The following uses the kolla deployment method as an example.

6.2.1 Configuring Cinder Driver for iSCSI-based Storage

Step 1 Configure the cinder.conf file in the corresponding path on the host and set related parameters by referring to 6.1.1 Configuring Cinder Driver for iSCSIbased Storage.

find / -name cinder.conf | grep volume

Step 2 Run the following commands to go to the Cinder Volume container:

```
docker ps | grep cinder
docker exec -it -u 0 <cinder_volume_docker_name> bash
```

- Step 3 In the /etc/cinder directory of the Cinder Volume container, create a Huawei Cinder Driver configuration file in .xml format and set related parameters by referring to 6.1.1 Configuring Cinder Driver for iSCSI-based Storage. You can customize the configuration file name, for example, cinder huawei conf.xml.
- **Step 4** Change the owner and user group of the Huawei Cinder Driver configuration file to be the same as those of the /etc/cinder/cinder.conf file.

```
-rw-----. 1 cinder cinder 2962 Sep 22 07:17 cinder.conf
-rw----. 1 cinder cinder 443 Sep 14 03:43 cinder_huawei_conf.xml
```

Step 5 Restart the Cinder Volume service.

docker restart <cinder_volume_docker_name>

----End

6.2.2 Configuring Cinder Driver for FC-based Storage

Step 1 Configure the **cinder.conf** file in the corresponding path on the host and set related parameters by referring to **6.1.2 Configuring Cinder Driver for FC-based Storage**.

find / -name cinder.conf | grep volume

Step 2 Run the following commands to go to the Cinder Volume container:

docker ps | grep cinder docker exec -it -u 0 <cinder_volume_docker_name> bash

- Step 3 In the /etc/cinder directory of the Cinder Volume container, create a Huawei Cinder Driver configuration file in .xml format and set related parameters by referring to 6.1.2 Configuring Cinder Driver for FC-based Storage. You can customize the configuration file name, for example, cinder_huawei_conf.xml.
- **Step 4** Change the owner and user group of the Huawei Cinder Driver configuration file to be the same as those of the /etc/cinder/cinder.conf file.

```
-rw-----. 1 cinder cinder 2962 Sep 22 07:17 cinder.conf
-rw----. 1 cinder cinder 443 Sep 14 03:43 cinder huawei conf.xml
```

Step 5 Restart the Cinder Volume service.

docker restart < cinder_volume_docker_name>

----End

Configuring the Multipathing Software

Multipathing is configured to improve the link reliability of LUNs of SAN storage. If the multipathing configuration is incorrect, an I/O error occurs when a single link is faulty. As a result, the VM file system is read-only or faulty, affecting VM service delivery. When configuring multipathing, install the multipathd service on the host and configure proper parameters. Then, configure multipathing parameters on OpenStack.

7.1 Configuring Host Multipathing

7.2 Configuring OpenStack Multipathing

7.1 Configuring Host Multipathing

7.1.1 Installing the Multipathing Tool Package

Step 1 Install the Multipath tool package based on the operating system.

CentOS:

yum install -y device-mapper-multipath

Ubuntu:

apt-get install -y multipath-tools apt-get install -y multipath-tools-boot

Step 2 Enable the host multipathing service.

CentOS:

/sbin/mpathconf --enable systemctl start multipathd.service systemctl enable multipathd.service systemctl restart multipathd.service

Ubuntu:

systemctl restart multipath-tools.service

----End

7.1.2 Configuring the Multipathing Service

If the AA mode is used, add the following content to the **devices** field in the multipathing configuration file (/etc/multipath.conf). For details, see *OceanStor Dorado V6 Host Connectivity Guide for Red Hat*.

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

If the ALUA mode is used, add the following content to the **devices** field in the multipathing configuration file (/etc/multipath.conf). For details, see *Huawei* SAN Storage Host Connectivity Guide for Red Hat.

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

After the configuration is complete, restart the multipathd service.

systemctl restart multipathd.service

□ NOTE

When OceanStor converged storage and OceanStor Dorado V6 storage coexist, that is, you use OceanStor converged storage and then expand the capacity using OceanStor Dorado storage, retain the multipathing configuration file (/etc/multipath.conf).

7.2 Configuring OpenStack Multipathing

7.2.1 Configuring Multipathing for OpenStack Nova Nodes

Step 1 Configure the /etc/nova/nova.conf file.

Add volume use multipath = True in [libvirt] of /etc/nova/nova.conf.

□ NOTE

- If the version of OpenStack is Mitaka or earlier, add iscsi_use_multipath = True in [libvirt] of /etc/nova/nova.conf.
- If the version of OpenStack is Newton or later, add volume_use_multipath = True in [libvirt] of /etc/nova/nova.conf.

Step 2 Restart the nova-compute service.

systemctl restart openstack-nova-compute.service

----End

7.2.2 Configuring Multipathing for OpenStack Cinder Nodes

Step 1 Add the following content to the end of **BACKEND** in the /etc/cinder/cinder.conf file:

[huawei_backend]

use_multipath_for_image_xfer = true enforce_multipath_for_image_xfer = true

Step 2 Restart the cinder-volume service.

systemctl restart openstack-cinder-volume.service

----End

8 Configuring Advanced Properties

Huawei storage supports a series of advanced features. You can create volumes with advanced features by configuring specific volume types.

NOTICE

Licenses for advanced features must be imported onto Huawei storage devices in advance.

- 8.1 Configuring the Thick Property
- 8.2 Configuring the Thin Property
- 8.3 Configuring the SmartQoS Property
- 8.4 Configuring the SmartPartition Property
- 8.5 Configuring the SmartCache Property
- 8.6 Configuring the SmartTier Property
- 8.7 Configuring the HyperMetro Property
- 8.8 Configuring the Remote Replication Property
- 8.9 Creating a LUN in a Specified Storage Pool
- 8.10 Creating a LUN on a Disk of a Specified Type
- 8.11 Creating a LUN in on a Specified Backend
- 8.12 Configuring a Backup Snapshot
- 8.13 Configuring Certificate Authentication
- 8.14 Configuring the Owning Controller of a LUN
- 8.15 Configuring the SmartDedupe Property
- 8.16 Configuring the SmartCompression Property
- 8.17 Creating a Volume Using fastclone

- 8.18 Configuring the LUN Copy Speed
- 8.19 Configuring the Application Type of a LUN
- 8.20 Configuring a Tenant User
- 8.21 Configuring Multiple Backends
- 8.22 Configuring Auto Zoning
- 8.23 Configuring CHAP

8.1 Configuring the Thick Property

This section describes how to configure the Thick property.

NOTICE

OceanStor Dorado does not support Thick volumes.

- **Step 1** Run the following command to create a volume type. *<name>* indicates the name of the volume type.
 - # cinder type-create <name>
- **Step 2** Run the following command to set the Thick volume type. *<vtype>* indicates the volume type name configured in **Step 1**.
 - # cinder type-key <vtype> set capabilities:thick_provisioning_support='<is> true'
- **Step 3** Run the following command to set the scheduling mechanism for the Thick volume. <*vtype>* indicates the volume type name configured in **Step 1**.
 - # cinder type-key <vtype> set provisioning:type='thick'
- **Step 4** Use the volume type in **Step 1** to create a volume.
 - ----End

8.2 Configuring the Thin Property

This section describes how to configure the Thin property.

- **Step 1** Run the following command to create a volume type. *<name>* indicates the name of the volume type.
 - # cinder type-create <name>
- **Step 2** Run the following command to set the Thin volume type. *<vtype>* indicates the volume type name configured in **Step 1**.
 - # cinder type-key <vtype> set capabilities:thin_provisioning_support='<is> true'
- **Step 3** Use the volume type in **Step 1** to create a volume.
 - ----End

8.3 Configuring the SmartQoS Property

QoS in OpenStack mainly depends on front-end Hypervisor and back-end storage.

For details about how to configure front-end QoS, see https://docs.openstack.org/cinder/latest/admin/blockstorage-basic-volume-qos.html.

This section describes how to configure back-end QoS for Huawei storage devices.

Step 1 Run the following command to create a volume type. *<name>* indicates the name of the volume type.

cinder type-create <name>

Step 2 Run the following command to enable QoS. *<vtype>* indicates the volume type name configured in **Step 1**.

cinder type-key <vtype> set capabilities:QoS_support='<is> true'

Step 3 Run the following command to create a QoS specification. <*name>* indicates the name of a customized QoS specification.

cinder qos-create <name> IOType=* <qos_key>=***

Table 8-1 IOType parameter

Parameter	Description	Remarks
ЮТуре	Read/write type.	Valid values are as follows:
		• 0 : read I/O
		• 1: write I/O
		• 2: read and write I/Os

Table 8-2 Optional qos_key parameters

Parameter	Description	Remarks
maxIOPS	Maximum IOPS.	The value is a positive integer, value should not be less than 100 if its type is Dorado.
minIOPS	Minimum IOPS.	The value is a positive integer.
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.
latency	Maximum latency.	The value is a positive integer, expressed in ms.

- maxBandWidth or maxIOPS cannot coexist with minBandWidth, minIOPS, or latency.
- For OceanStor Dorado, IOType must be set to 2 and minBandWidth, minIOPS, and latency are unavailable.
- **Step 4** Associate the QoS specification with the volume type. <qos_specs> indicates the QoS specification ID created in **Step 3** and <volume_type_id> indicates the volume type ID created in **Step 1**.

cinder gos-associate <gos_specs> <volume_type_id>

Step 5 Use the volume type in **Step 1** to create a volume.

----End

8.4 Configuring the SmartPartition Property

This section describes how to configure the SmartPartition property.

- **Step 1** Run the following command to create a volume type. *<name>* indicates the name of the volume type.
 - # cinder type-create <name>
- **Step 2** Run the following command to enable SmartPartition. <*vtype>* indicates the volume type name configured in **Step 1**.
 - # cinder type-key <vtype> set capabilities:smartpartition='<is> true'
- **Step 3** Run the following command to configure the name of existing SmartPartition on the back-end storage:
 - # cinder type-key <vtype> set smartpartition:partitionname='***'
- **Step 4** Use the volume type in **Step 1** to create a volume.

----End

8.5 Configuring the SmartCache Property

This section describes how to configure the SmartCache property.

- **Step 1** Run the following command to create a volume type. *<name>* indicates the name of the volume type.
 - # cinder type-create <name>
- **Step 2** Run the following command to enable SmartCache. *<vtype>* indicates the volume type name configured in **Step 1**.
 - # cinder type-key <vtype> set capabilities:smartcache='<is> true'
- **Step 3** Run the following command to configure the name of existing SmartCache on the back-end storage:
 - # cinder type-key <vtype> set smartcache:cachename='***'
- **Step 4** Use the volume type in **Step 1** to create a volume.

----End

8.6 Configuring the SmartTier Property

This section describes how to configure the SmartTier property.

Step 1 Run the following command to create a volume type. *<name>* indicates the name of the volume type.

cinder type-create <name>

Step 2 Run the following command to enable SmartTier. <*vtype>* indicates the volume type name configured in **Step 1**.

cinder type-key <vtype> set capabilities:smarttier='<is> true'

Step 3 Run the following command to configure the SmartTier migration policy:

cinder type-key <vtype> set smarttier:policy=***

Ⅲ NOTE

policy indicates a migration policy. The value can be:

- 0: no migration
- 1: automatic migration
- 2: migration to a high-performance tier
- 3: migration to a low-performance tier

Step 4 Use the volume type in **Step 1** to create a volume.

----End

8.7 Configuring the HyperMetro Property

This section describes how to configure the HyperMetro property.

Step 1 Add remote device information to the specified backend in the **/etc/cinder/cinder.conf** file.

The ALUA policy of OceanStor Dorado V6 is different. See the following.

Table 8-3 Parameter description

Parameter	Description	Mandatory
storage_pool	Name of the HyperMetro remote storage pool.	Yes
san_address	Access address of the REST interface.	Yes
san_user	User name of a storage administrator or tenant of the HyperMetro remote device.	Yes
san_password	Password of a storage administrator or tenant of the HyperMetro remote device.	Yes
vstore_name	vStore name of the HyperMetro remote device. This parameter needs to be configured only when a tenant user is used.	No
metro_domain	HyperMetro domain name.	Yes
metro_sync_co mpleted	Whether to wait for the synchronization to complete during HyperMetro creation. The value can be True or False . The default value is True .	No
iscsi_info	HostName is the regular expression of the Nova or Cinder node host name (configuration rule and priority description). For details about ALUA, FAILOVERMODE, PATHTYPE, ACCESSMODE, and HYPERMETROPATHOPTIMIZED storage ALUA policy, see 11.1 Appendix 1: ALUA Parameter Description. TargetIP indicates the IP address of the iSCSI target port. You can enter multiple IP addresses and separate them with spaces. This parameter can be set for different hosts.	Mandatory for iSCSI networking
fc_info	HostName is the regular expression of the Nova or Cinder node host name (configuration rule and priority description). For details about ALUA, FAILOVERMODE, PATHTYPE, ACCESSMODE, and HYPERMETROPATHOPTIMIZED storage ALUA policy, see 11.1 Appendix 1: ALUA Parameter Description. This parameter can be set for different hosts.	Mandatory for FC networking

- **Step 2** Restart the Cinder Volume service.
- **Step 3** Run the following command to create a volume type. *<name>* indicates the name of the volume type.

cinder type-create <name>

Step 4 Run the following command to enable HyperMetro. *<vtype>* indicates the volume type name configured in **Step 3**.

cinder type-key <vtype> set capabilities:hypermetro='<is> true'

Step 5 Use the volume type in **Step 3** to create a volume.

----End

8.8 Configuring the Remote Replication Property

This section describes how to configure the remote replication property.

Step 1 Add remote device information to the specified backend in the **/etc/cinder/cinder.conf** file.

```
replication_device =
backend_id:***,
storage_pool:***,
san_address:https://*.**.*:8088/deviceManager/rest/,
san_user:***,
san_password:******,
vstore_name:***,
iscsi_info: {HostName:xxx;ALUA:xxx;FAILOVERMODE:xxx;PATHTYPE:xxx;TargetIP:x.x.x.x y.y.y.y};
{HostName:xxx;ALUA:xxx;FAILOVERMODE:xxx;PATHTYPE:xxx;TargetIP:x.x.x.x z.z.z.z},
fc_info: {HostName:xxx;ALUA:xxx;FAILOVERMODE:xxx;PATHTYPE:xxx};
```

The ALUA policy of OceanStor Dorado V6 is different. See the following.

Table 8-4 Parameter description

Parameter	Description	Mandatory
backend_id	Target device ID. This parameter must be specified in the event of failover.	Yes
storage_pool	Storage pool name of the target backend in the event of failover.	Yes
san_address	Access address of the REST interface.	Yes
san_user	User name of a storage administrator or tenant of the remote replication remote device.	Yes

Parameter	Description	Mandatory
san_password	Password of a storage administrator or tenant of the remote replication remote device.	Yes
vstore_name	vStore name of the remote replication remote device. This parameter needs to be configured only when a tenant user is used.	No
iscsi_info	HostName is the regular expression of the Nova or Cinder node host name (configuration rule and priority description). For details about ALUA, FAILOVERMODE, PATHTYPE, ACCESSMODE, and HYPERMETROPATHOPTIMIZED storage ALUA policy, see 11.1 Appendix 1: ALUA Parameter Description. TargetIP indicates the IP address of the iSCSI target port. You can enter multiple IP addresses and separate them with spaces. This parameter can be set for different hosts.	Mandatory for iSCSI networking
fc_info	HostName is the regular expression of the Nova or Cinder node host name (configuration rule and priority description). For details about ALUA, FAILOVERMODE, PATHTYPE, ACCESSMODE, and HYPERMETROPATHOPTIMIZED storage ALUA policy, see 11.1 Appendix 1: ALUA Parameter Description. This parameter can be set for different hosts.	Mandatory for FC networking

- **Step 2** Restart the Cinder Volume service.
- **Step 3** Run the following command to create a volume type. *<name>* indicates the name of the volume type.

cinder type-create <name>

- **Step 4** Run the following command to enable remote replication. *<vtype>* indicates the volume type name configured in **Step 3**.
 - # cinder type-key <vtype> set capabilities:replication_enabled='<is> true'
- **Step 5** (Optional) Run the following command to set the replication type: # cinder type-key <vtype> set replication_type='<in> ***'

□ NOTE

The value of *** can be:

- **sync**: synchronous replication
- **async**: asynchronous replication

If **replication_type** is not set, asynchronous replication is used by default.

Step 6 Use the volume type in **Step 3** to create a volume.

----End

8.9 Creating a LUN in a Specified Storage Pool

This section describes how to create a volume in a specified storage pool.

Step 1 Run the following command to create a volume type. *<name>* indicates the name of the volume type.

cinder type-create <name>

Step 2 Run the following command to configure a single storage pool. *<vtype>* indicates the volume type name configured in **Step 1**.

cinder type-key <vtype> set pool_name=StoragePool001

Alternatively, run the following command to configure multiple storage pools:

cinder type-key <vtype> set pool_name="<or> StoragePool001 <or> StoragePool002"

Step 3 Use the volume type in **Step 1** to create a volume.

----End

8.10 Creating a LUN on a Disk of a Specified Type

This section describes how to create a volume on a disk of a specified type.

Step 1 Run the following command to create a volume type. *<name>* indicates the name of the volume type.

cinder type-create <name>

Step 2 Run the following command to configure a disk type. *<vtype>* indicates the volume type name configured in **Step 1**.

cinder type-key <vtype> set disk_type=sas

Alternatively, run the following command to configure multiple disk types:

cinder type-key <vtype> set disk_type="<or> sas <or> ssd"

The value of **disk_type** can be **ssd**, **sas**, **nl_sas**, or **mix**, where **mix** indicates that two or more of **ssd**, **sas**, and **nl_sas** are used.

Step 3 Use the volume type in **Step 1** to create a volume.

----End

8.11 Creating a LUN in on a Specified Backend

This section describes how to create a volume on a specified backend.

Step 1 Run the following command to create a volume type. *<name>* indicates the name of the volume type.

cinder type-create <name>

Step 2 Run the following command to configure the backend name. *<vtype>* indicates the volume type name configured in **Step 1**.

cinder type-key <vtype> set volume_backend_name=***

□ NOTE

Set **volume_backend_name** to the value of **volume_backend_name** in the **/etc/cinder/cinder.conf** configuration file of the backend.

Step 3 Use the volume type in **Step 1** to create a volume.

----End

8.12 Configuring a Backup Snapshot

This section describes how to configure a backup snapshot.

- **Step 1** Configure a backup driver by referring to https://docs.openstack.org/ocata/config-reference/block-storage/backup-drivers.html. Ensure that the Cinder Backup service is running properly.
- **Step 2** Add the following information to the specified backend in the **/etc/cinder/ cinder.conf** file:

backup use temp snapshot = True

Step 3 Restart the Cinder Volume and Cinder Backup services.

----End

8.13 Configuring Certificate Authentication

This section describes how to configure certificate authentication to connect backend storage.

Step 1 Configure the Cinder Driver configuration file. Add **SSLCertVerify** and **SSLCertPath** in **<Storage>** as follows:

□ NOTE

- **SSLCertVerify** indicates whether to enable certificate authentication. The value can be **True** or **False**. If the parameter is not set, the value is **False** by default.
- SSLCertPath indicates the certificate path for authentication and is valid only when SSLCertVerify is set to True.
- **Step 2** Restart the Cinder Volume service.

----End

8.14 Configuring the Owning Controller of a LUN

This section describes how to configure the owning controller of a LUN.

- **Step 1** Run the following command to create a volume type. *<name>* indicates the name of the volume type.
 - # cinder type-create <name>
- **Step 2** Run the following command to enable the huawei_controller property. <*vtype>* indicates the volume type name configured in **Step 1**.
 - # cinder type-key <vtype> set capabilities:huawei_controller='<is> true'
- **Step 3** Run the following command to configure the owning controller name:

cinder type-key <vtype> set huawei_controller:controllername=***

□ NOTE

The format of the owning controller name is **CTE0.A**.

Step 4 Use the volume type in **Step 1** to create a volume.

----End

8.15 Configuring the SmartDedupe Property

This section describes how to configure the SmartDedupe property. Only Thin volumes support this property.

Procedure of Enabling SmartDedupe

- **Step 1** Run the following command to create a volume type. *<name>* indicates the name of the volume type.
 - # cinder type-create <name>
- **Step 2** For OpenStack Newton and later versions, run the following command to enable SmartDedupe. <*vtype>* indicates the volume type name configured in **Step 1**.

cinder type-key <vtype> set capabilities:thin_provisioning_support='<is> true' capabilities:dedup='<is> true'

For OpenStack Mitaka and earlier versions, run the following command to enable SmartDedupe. <*vtype>* indicates the volume type name configured in Step 1.

cinder type-key <vtype> set capabilities:thin_provisioning_support='<is> true' capabilities:dedup='<in> true'

Step 3 Use the volume type in **Step 1** to create a volume.

----End

□ NOTE

OceanStor Dorado supports only Thin volumes. The **capabilities:thin_provisioning_support** parameter in **Step 2** is optional.

Procedure of Disabling SmartDedupe

Step 1 Run the following command to create a volume type. *<name>* indicates the name of the volume type.

cinder type-create <name>

Step 2 For OpenStack Newton and later versions, run the following command to disable SmartDedupe. <*vtype>* indicates the volume type name configured in **Step 1**.

cinder type-key <vtype> set capabilities:thin_provisioning_support='<is> true' capabilities:dedup='<is> false'

For OpenStack Mitaka and earlier versions, run the following command to disable SmartDedupe. <*vtype>* indicates the volume type name configured in **Step 1**.

cinder type-key <vtype> set capabilities:thin_provisioning_support='<is> true' capabilities:dedup='<in> false'

Step 3 Use the volume type in **Step 1** to create a volume.

----End

■ NOTE

OceanStor Dorado supports only Thin volumes. The **capabilities:thin_provisioning_support** parameter in **Step 2** is optional.

8.16 Configuring the SmartCompression Property

This section describes how to configure the SmartCompression property. Only Thin volumes support this property.

Procedure of Enabling SmartCompression

Step 1 Run the following command to create a volume type. *<name>* indicates the name of the volume type.

cinder type-create <name>

Step 2 For OpenStack Newton and later versions, run the following command to enable SmartCompression. <*vtype>* indicates the volume type name configured in **Step 1**.

cinder type-key <vtype> set capabilities:thin_provisioning_support='<is> true'
capabilities:compression='<is> true'

For OpenStack Mitaka and earlier versions, run the following command to enable SmartCompression. vtype indicates the volume type name configured in Step 1.

cinder type-key <vtype> set capabilities:thin_provisioning_support='<is> true'
capabilities:compression='<in> true'

Step 3 Use the volume type in **Step 1** to create a volume.

----End

□ NOTE

OceanStor Dorado supports only Thin volumes. The **capabilities:thin_provisioning_support** parameter in **Step 2** is optional.

Procedure of Disabling SmartCompression

Step 1 Run the following command to create a volume type. *<name>* indicates the name of the volume type.

cinder type-create <name>

Step 2 For OpenStack Newton and later versions, run the following command to disable SmartCompression. *<vtype>* indicates the volume type name configured in **Step 1**.

cinder type-key <vtype> set capabilities:thin_provisioning_support='<is> true'
capabilities:compression='<is> false'

For OpenStack Mitaka and earlier versions, run the following command to disable SmartCompression. <vtype> indicates the volume type name configured in Step 1.

cinder type-key <vtype> set capabilities:thin_provisioning_support='<is> true'
capabilities:compression='<in> false'

Step 3 Use the volume type in **Step 1** to create a volume.

----End

□ NOTE

OceanStor Dorado supports only Thin volumes. The **capabilities:thin_provisioning_support** parameter in **Step 2** is optional.

8.17 Creating a Volume Using fastclone

This section describes how to configure **fastclone** to create volumes from snapshots or clone volumes.

Step 1 When creating volumes from snapshots or cloning volumes, specify the -- metadata parameter with the fastclone key as follows:

```
# cinder create --snapshot-id *** --metadata fastclone=True
# cinder create --source-volid *** --metadata fastclone=True
```

----End

- If **fastclone** is not set, LUN copy is used by default.
- Only OceanStor Dorado V3/V6 supports **fastclone**. For details, please contact Huawei storage R&D engineers.

8.18 Configuring the LUN Copy Speed

When you create volumes from snapshots or clone volumes, LUN copy is used by default. This section describes how to configure the LUN copy speed.

When creating volumes from snapshots or cloning volumes, specify the -- **metadata** parameter with the **copyspeed** key as follows:

```
# cinder create --snapshot-id *** --metadata copyspeed=2
# cinder create --source-volid *** --metadata copyspeed=2
```

Table 8-5 copyspeed parameter value description

Parameter Value	Description
1	Low
2	Medium

Parameter Value	Description
3	High
4	Highest

□ NOTE

If copyspeed is not set, the medium speed is used by default.

8.19 Configuring the Application Type of a LUN

This section describes how to configure the application type of a LUN.

Step 1 Run the following command to create a volume type. *<name>* indicates the name of the volume type.

cinder type-create <name>

Step 2 Run the following command to enable the huawei_application_type property. <*vtype>* indicates the volume type name configured in **Step 1**.

cinder type-key <vtype> set capabilities:huawei_application_type='<is> true'

Step 3 Run the following command to configure the application type name:

cinder type-key <vtype> set huawei_application_type:applicationname='***'

Step 4 Use the volume type in **Step 1** to create a volume.

----End

Ⅲ NOTE

- This configuration is only available for Dorado V300R001C20 and later versions.
- This configuration includes the configuration of Compression and Dedupe. And its
 priority is the highest among SmartCompression, SmartDedupe and Application type.
 Once both configured, Configuration the application type takes precedence.

8.20 Configuring a Tenant User

Configure tenant user parameters in the Cinder Driver configuration file as follows:

Table 8-6 Parameter description

Parameter	Description
UserName	Storage tenant user name.
UserPassword	Storage tenant user password.
vStoreName	vStore name.

NOTICE

- For details about how to configure other parameters, refer to preceding sections.
- Tenant users can be configured only when the storage system version supports the multi-tenant feature.

8.21 Configuring Multiple Backends

Configure multiple storage systems in **/etc/cinder/cinder.conf**. The following is an example.

```
[DEFAULT]
...
enabled_backends = v3_fc,v5_fc
[v3_fc]
volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiFCDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf_v3_fc.xml
volume_backend_name = v3_fc
[v5_fc]
volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiFCDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf_v5_fc.xml
volume_backend_name = v5_fc
```

8.22 Configuring Auto Zoning

Auto zoning is not compulsory. When you need this function, perform the following configuration.

- **Step 1** Configure the **cinder.conf** file. For details about how to set switch parameters, see https://docs.openstack.org/cinder/latest/configuration/block-storage/fc-zoning.html.
- **Step 2** Restart the Cinder Volume service.

----End

NOTICE

Auto zoning is not supported for tenant users.

8.23 Configuring CHAP

On a public network, any application server whose IP address resides on the same network segment as that of the storage system's iSCSI host port can access the storage system and perform read and write operations. This poses risks to data security. To ensure storage system access security, you can configure CHAP authentication to control application servers' access to the storage system.

Add the following configuration to the Initiator configuration item in the Huawei Driver configuration file:

```
<iSCSI>
<Initiator ... CHAPinfo="xxx"/>
<Initiator ... CHAPinfo="xxx"/>
</iSCSI>
```

Table 8-7 Parameter description

Parameter	Default Value	Description
CHAPinfo	-	User-defined user name and password for CHAP authentication. The format is mmuser;mm-user@storage1. The user name and password are separated by a semicolon (;).
		The CHAP name contains 4 to 223 characters.
		The password contains 12 to 16 characters.
		The password must contain any three types of uppercase letters, lowercase letters, digits, and special characters (` \sim ! @ # \$ % \wedge * () = + \ [{ }];:,./? and spaces)
		The password must not be the same as the CHAP name or the reverse of the CHAP name.

9Best Practices

- 9.1 Configuring Basic Properties for Interconnection Between iSCSI and OceanStor Dorado V3
- 9.2 Initiator Configuration Policy Example

9.1 Configuring Basic Properties for Interconnection Between iSCSI and OceanStor Dorado V3

The following configurations must be performed after 4 **Deployment** and **7.1 Configuring Host Multipathing** are complete.

Step 1 Configure /etc/cinder/cinder huawei conf.xml (iSCSI-based storage).

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
  <Storage>
   <Product>Dorado</Product>
   <Protocol>iSCSI</Protocol>
   <RestURL>https://192.168.128.100:8088/deviceManager/rest/</RestURL>
   <UserName>user</UserName>
   <UserPassword>password</UserPassword>
 </Storage>
  <LUN>
   <StoragePool>StoragePool001</StoragePool>
   <LUNType>Thin</LUNType>
   <LUNCopySpeed>3</LUNCopySpeed>
  </LUN>
 <iSCSI>
   <Initiator HostName="host[0-4]" ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0"</pre>
PATHTYPE="0" TargetIP="128.115.128.100 128.115.128.101"/>
   <Initiator HostName="host[5-9]" ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0"
PATHTYPE="0" TargetIP="128.115.128.102 128.115.128.103"/>
 </iSCSI>
</config>
```

∩ NOTE

- This configuration is for iSCSI-based storage. For details about FC-based storage, see Step 2.
- For details about the parameter meanings and procedure, see 6.1.1 Configuring Cinder Driver for iSCSI-based Storage.
- ALUA parameter settings vary with OS types. For details, see 11.1 Appendix 1: ALUA Parameter Description.

Step 2 Configure /etc/cinder/cinder_huawei_conf.xml (FC-based storage).

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
  <Storage>
   <Product>Dorado</Product>
   <Protocol>FC</Protocol>
   <RestURL>https://192.168.128.100:8088/deviceManager/rest/</RestURL>
   <UserName>user</UserName>
   <userPassword>password</userPassword>
 </Storage>
  <LUN>
   <StoragePool>StoragePool001</StoragePool>
   <LUNType>Thin</LUNType>
   <LUNCopySpeed>3</LUNCopySpeed>
  </LUN>
   <MinOnlineFCInitiator>1</MinOnlineFCInitiator>
   <Initiator HostName="xxx" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
   <Initiator HostName="xxx" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
</config>
```


- This configuration is for FC-based storage. For details about iSCSI-based storage, see Step 1.
- For details about the parameter meanings and procedure, see 6.1.2 Configuring Cinder Driver for FC-based Storage.
- ALUA parameter settings vary with OS types. For details, see 11.1 Appendix 1: ALUA
 Parameter Description.
- **Step 3** Modify the owner and owning group of Huawei driver configuration file is the same as those of file /etc/cinder/cinder.conf.

```
-rw-r--r- 1 cinder cinder 2662 Jul 11 11:25 cinder.conf
-rw-r--r- 1 cinder cinder 778 Jul 11 11:25 cinder_huawei_conf.xml
```

Step 4 Configure /etc/cinder/cinder.conf.

```
[huawei_backend]
volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiISCSIDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml
volume_backend_name = huawei_backend
use_multipath_for_image_xfer = true
enforce_multipath_for_image_xfer = true
```

Modify the following content in the [DEFAULT] area and add huawei_backend:

```
[DEFAULT]
...
enabled_backends=..., huawei_backend
```

Step 5 Configure multipathing for OpenStack Nova, Add the following content to **[libvirt]** in the **/etc/nova/nova.conf** file

```
[libvirt]
...
volume_use_multipath = True
```

Step 6 Restart the **nova-compute** and **cinder-volume** services.

```
systemctl restart openstack-nova-compute.service
systemctl restart openstack-cinder-volume.service
```

----End

9.2 Initiator Configuration Policy Example

Example 1: (iSCSI) The configuration file content is as follows:

```
<iSCSI>
<Initiator HostName="*" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx" TargetIP="x.x.x.x y.y.y.y"/>
<Initiator HostName="host" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx" TargetIP="x.x.x.x y.y.y.y"/>
</iSCSI>
```

If the host name is **host**, you can configure ALUA and port groups for the initiator. According to the rule (subject to the precise configuration), the priority of the second record (**HostName="host"**) is higher than that of the first record (**HostName="*"**).

Example 2: (iSCSI) The configuration file content is as follows:

```
<iSCSI>
<Initiator HostName="Host[0-9]" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx" TargetIP="x.x.x.x y.y.y.y"/>
<Initiator HostName="Host[5-7]" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx" TargetIP="x.x.x.x y.y.y.y"/>
</iSCSI>
```

If the host name is **Host6**, the preceding two configuration items meet the configuration rules. According to the rules, the **first** configuration item is used to set the initiator.

Example 3: (FC) The configuration file content is as follows:

```
<FC>
<Initiator HostName="*" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
<Initiator HostName="host" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
</FC>
```

If the host name is **host**, you can configure ALUA and port groups for the initiator. According to the rule (subject to the precise configuration), the priority of the second record (**HostName="host"**) is higher than that of the first record (**HostName="*"**).

Example 4: (FC) The configuration file content is as follows:

```
<FC>
<Initiator HostName="Host[0-9]" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
<Initiator HostName="Host[5-7]" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
</FC>
```

If the host name is **Host6**, the preceding two configuration items meet the configuration rules. According to the rules, the **first** configuration item is used to set the initiator.

Example 5: (iSCSI) The configuration file content is as follows:

```
<iSCSI>
<Initiator HostName="^Host1$" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx" TargetIP="x.x.x.x y.y.y.y"/>
<Initiator HostName="^Host10$" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx" TargetIP="x.x.x.x y.y.y.y"/>
</iSCSI>
```

If the host name is **Host1**, the first configuration parameter is selected to set the initiator. If the host name is **Host10**, the secondconfiguration parameter is selected to set the initiator. ^ indicates the start of the matching string, and \$ indicates the end of the matching string.

Example 6: (FC) The configuration file content is as follows:

```
<FC>
<Initiator HostName="^Host1$" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
<Initiator HostName="^Host10$" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
</FC>
```

If the host name is **Host1**, the first configuration parameter is selected to set the initiator. If the host name is **Host10**, the secondconfiguration parameter is selected to set the initiator. ^ indicates the start of the matching string, and \$ indicates the end of the matching string.

10_{FAQs}

10.1 Do I Need to Create Hosts, Host Groups, LUN Groups, and Other Associated Objects on Huawei Storage Before Using Cinder Driver to Map Volumes?

10.2 What Can I Do if Mapping Fails Because Nova Logs Indicate that the lsblk Command Fails to Be Executed During Volume Mapping?

10.3 Failed to Create Volumes From Images When verify_glance_signatures=enabled

10.4 There Are Some "alua not supported" Errors When You Run the "multipath - ll" Command

10.5 Failed to Delete or Live Migrate a VM

10.6 After a Volume is Deleted from OpenStack, the Volume Enters the Recycle Bin. When You Try to Create the Volume Again, the Volume Fails to Be Created Due to Duplicate Names.

10.7 How Do I Manually Map LUNs to an OpenStack Host Without Affecting Cinder Driver's Volume Mapping?

10.1 Do I Need to Create Hosts, Host Groups, LUN Groups, and Other Associated Objects on Huawei Storage Before Using Cinder Driver to Map Volumes?

No, because Cinder Driver will automatically create associated objects and mapping views. If these objects are manually created on Huawei storage, the mapping logic of Cinder Driver will be adversely affected. Therefore, ensure that these objects are deleted before mapping volumes.

10.2 What Can I Do if Mapping Fails Because Nova Logs Indicate that the lsblk Command Fails to Be Executed During Volume Mapping?

Nova will invoke the **lsblk** command in the volume mapping process to query the information of the mapped volume. This error occurs generally because the lsblk

tool is not installed. Ensure that the tool is installed in the Nova system and you have execute permission.

10.3 Failed to Create Volumes From Images When verify_glance_signatures=enabled

Issue Description

OpenStack executes creating a volume from an image. After copying the image metadata for a period of time, an error is reported: Failed to copy metadata to volume: Glance metadata cannot be updated, key signature_verified exists for volume id b75f957c-0ba4-4099-8706-866c68012779, which eventually causes the volume creation to fail.

Environment

Operating System: CentOS 7.6

Server: Huawei 5288 V5

OpenStack Version: Rocky

Root Cause

Bug Link: https://bugs.launchpad.net/cinder/+bug/1823445

A new field verify_glance_signatures has been added to cinder.image.glance.py since Rocky. The default value is enabled. When CONF.verify_glance_signatures! = 'disabled' and signature_verification in the image also exists, then the DB API does not allow duplicate writes signature_verification flag, thereby throwing an exception.

Solution

In [DEFAULT] of the cinder.conf file, set the verify_glance_signatures field to disabled, for example:

[DEFAULT]

...

verify_glance_signatures = disabled

...

10.4 There Are Some "alua not supported" Errors When You Run the "multipath -ll" Command

Issue Description

After host multipathing and OpenStack multipathing are configured, a VM is created or migrated and the multipath -ll command is run to query path information. Some drive letters are displayed as "alua not supported".

Environment

OpenStack Version: All versions

Root Cause

Host's multipathing (/etc/multipath.conf) and ALUA (../cinder/cinder_huawei_conf.xml) are incorrectly configured.

Solution

- 1. Perform the following step on all compute nodes. Configure the host multipath service. See **Configuring Host Multipathing**.
- On all cinder volume nodes, configure storage ALUA. See Configuring Cinder Driver for iSCSI-based Storage or Configuring Cinder Driver for FC-based Storage. For details about the configuration parameters, see Appendix 1 ALUA Parameter Description.
- Restart the multipathd service. systemctl restart multipathd.service
- 4. Restart the cinder-volume service. systematl restart openstack-cinder-volume.service
- 5. Remap all created VMs (detach volumes and attach volumes). You can perform operations such as rebuild instance or migrate instance.

10.5 Failed to Delete or Live Migrate a VM

Issue Description

Failed to delete or live migrate a VM. According to the nova compute log, the following information is displayed:

```
Command: multipath -f mpath**
Exit code: 1
Stdout: ***** | mpath**: map in use\n ***** | failed to remove multipath map mpath**
```

Run the multipath -ll command to check the multipathing information. Some paths are displayed as "failed fault".

Environment

OpenStack Version: All versions

Root Cause

The /dev/sd* drive letter constituting dm-* has an LVM partition on the host machine. When the dm-* drive letter is removed, the LVM partition is not cleared on the host machine. As a result, the drive letter fails to be removed.

Solution

Take the OpenStack Train as an example. The official recommended configuration is https://docs.openstack.org/cinder/train/install/cinder-storage-install-rdo.html#install-and-configure-components.

On all compute nodes, configure the filter field in the device option in the /etc/lvm/lvm.conf file. For example:

devices { ... filter=["a/sdb/", "r/.*/"]

10.6 After a Volume is Deleted from OpenStack, the Volume Enters the Recycle Bin. When You Try to Create the Volume Again, the Volume Fails to Be Created Due to Duplicate Names.

Issue Description

Creating a HyperMetro volume fails due to timeout. You need to delete the volume and then create it again. The deleted volume is in the recycle bin. As a result, the volume with the same name fails to be created again.

Environment

OpenStack version: all versions

Storage version: Dorado V6 6.1.0/6.1.2

Root Cause

The recycle bin function is added to the storage system, and the default interface for deleting volumes is changed. When a volume is deleted, it is placed in the recycle bin by default. When a volume with the same name is created, an error message is displayed indicating that the volume already exists.

Solution

Disable the recycle bin function of the storage system. For details, see **Configure** > **Basic Storage Service Configuration Guide** > **Managing Basic Storage Services** > **Managing LUNs** > **Managing the Recycle Bin** > **Configuring the Recycle Bin** in *OceanStor Dorado 3000, Dorado 5000, and Dorado 6000 Product Documentation*.

10.7 How Do I Manually Map LUNs to an OpenStack Host Without Affecting Cinder Driver's Volume Mapping?

If you use the LUNs of the OpenStack Glance remote storage to create volumes or use the LUNs of the remote storage to create the **image_cache** partition on the system disk, you need to manually create LUNs on the storage and map them to the OpenStack host based on the following rules.

Solution

Manually create host groups, LUN groups, and mapping views on the storage device based on the following Cinder Driver rules.

■ NOTE

In principle, all LUNs and hosts on a storage device connected to the OpenStack platform are created, managed, and deleted by Cinder Driver based on specified rules. Therefore, if you want to manually create these objects, ensure that the rules for manually creating them are the same as the creation rules of Cinder Driver so that the platform will not fail to invoke the Cinder Driver to manage them due to conflicts.

- 1. Create a host. If the host name contains 32 or fewer characters, use the host name. If the host name contains more than 32 characters, use the **hashlib.md5()** method to process the host name. The details are as follows:
 - a. Log in to the host node and run the **cat /etc/hostname** command to obtain the host name.
 - b. Run the **python** or **python3** command.
 - c. Run the **import hashlib** command.
 - d. Run the print(hashlib.md5(("openstack-w").encode()).hexdigest()) command.

```
[root@openstack-w ~]# cat /etc/hostname
openstack-w
[root@openstack-w ~]# python3
Python 3.6.8 (default, May 8 2021, 09:11:34)
[GCC 8.4.1 20210423 (Red Hat 8.4.1-2)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import hashlib
>>> print(hashlib.md5(("openstack-w").encode()).hexdigest())
0e4ed842f0lb44143dd03dab6e04739f
```

2. Create a mapping view.

□ NOTE

Create a separate host group, LUN group, and mapping view for each host. The naming rules are as follows. *host_id* is the host ID automatically generated by the storage device after the host is created in **Step 1**.

Cinder Driver identifies the mapping views of OpenStack host nodes based on the following naming rules.

A host group is named as follows:

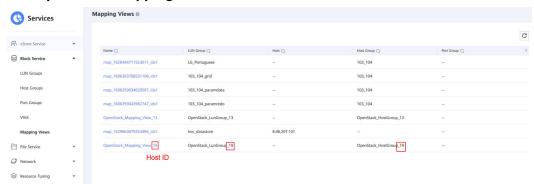
OpenStack_HostGroup_host_id. For example, OpenStack_HostGroup_1

A LUN is named as follows:

OpenStack_LunGroup_host_id. For example, OpenStack_LunGroup_1

- A mapping view is named as follows:

OpenStack_Mapping_View_host_id. For example, OpenStack_Mapping_View_1



Ⅲ NOTE

Mapping view names of OceanStor Dorado V6 cannot be changed on DeviceManager. You need to run the **ssh** *Username@IP* command to go to the storage CLI (the CLI login password is the same as the DeviceManager login password) and run the following command to change the name:

change mapping_view mapping_view_name=Old_map name=New_map

Username indicates the storage login user name, *IP* indicates the storage IP address, *Old_map* indicates the old name, and *New_map* indicates the new name.

11 Appendix

11.1 Appendix 1: ALUA Parameter Description

11.2 Appendix 2: Configuring Volume-backend image

11.3 Appendix 3: Configuring Image-Volume Cache

11.1 Appendix 1: ALUA Parameter Description

11.1.1 ALUA Parameter Description for Converged Storage

To improve the performance of links between hosts and storage, configure ALUA. This section describes how to set ALUA parameters in different scenarios.

- In non-HyperMetro scenarios, see Configuring Multipathing > Configuring Multipathing in Non-HyperMetro Scenarios in *Huawei SAN Storage Host Connectivity Guide for Red Hat*.
- In HyperMetro scenarios, see Configuring Multipathing > Configuring Multipathing in HyperMetro Scenarios in Huawei SAN Storage Host Connectivity Guide for Red Hat.

Table 11-1 ALUA parameter description

Parameter	Default Value	Description
ALUA	0	• 0 : Huawei UltraPath (default value)
		• 1: third-party multipathing software
FAILOVERMO DE	-	Initiator switchover mode. This parameter needs to be delivered only when third-party multipathing software is used.
		• 0 : early-version ALUA
		• 1: common ALUA
		• 2: ALUA not used
		• 3: special ALUA

Parameter	Default Value	Description
SPECIALMODE TYPE	-	Special mode type of the initiator. This parameter needs to be delivered only when initiator switchover mode is special ALUA.
		• 0 : mode 0
		• 1 : mode 1
		• 2 : mode 2
		• 3 : mode 3
PATHTYPE	-	Initiator path type. This parameter needs to be delivered only when third-party multipathing software is used.
		• 0 : optimal path
		• 1: non-optimal path

NOTICE

1. HyperMetro scenarios:

For OceanStor V3, V5, and Dorado V3, ALUA must be configured as follows:

- For hosts running CentOS or RHEL series:
 - Optimal path: ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0"
 - Non-optimal path: ALUA="1" FAILOVERMODE="3"
 SPECIALMODETYPE="0" PATHTYPE="1"
- For hosts running Ubuntu, SUSE, or Debain series:
 - Optimal path: ALUA="1" FAILOVERMODE="1" PATHTYPE="0"
 - Non-optimal path: ALUA="1" FAILOVERMODE="1" PATHTYPE="1"

2. Non-HyperMetro scenarios:

For OceanStor V3, V5, and Dorado V3:

Method 1: (Recommended) Configure ALUA as follows:

• For hosts running CentOS or RHEL series:

ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0"

• For hosts running Ubuntu, SUSE, or Debain series:

ALUA="1" FAILOVERMODE="1" PATHTYPE="0"

Method 2: Do not configure ALUA.

Note: Set **PATHTYPE** to 0 for the storage that is preferentially accessed, and set **PATHTYPE** to 1 for the storage that is not preferentially accessed.

11.1.2 ALUA Parameter Description for OceanStor Dorado V6 Storage

To improve the performance of links between hosts and storage, configure ALUA. This section describes how to set ALUA parameters in different scenarios.

- In non-HyperMetro scenarios, see Configuring Multipathing > Non-HyperMetro Scenarios in OceanStor Dorado Host Connectivity Guide for Red Hat.
- In HyperMetro scenarios, see Configuring Multipathing > HyperMetro Scenarios in OceanStor Dorado Host Connectivity Guide for Red Hat.

Coexistence of OceanStor Converged Storage and OceanStor Dorado V6 Storage

Table 11-2 ALUA parameter description

Parameter	Default Value	Description
ACCESSMODE	0	Host access mode. This parameter applies only to OceanStor Dorado V6 storage devices. If both ACCESSMODE and ALUA are configured, the value of ACCESSMODE prevails. • 0: balanced (default value) • 1: asymmetric
HYPERMETRO PATHOPTIMIZ ED	-	HyperMetro optimal path. This parameter is optional when ACCESSMODE is set to 1 . • 0 : no • 1 : yes

Creating OceanStor Dorado V6 Storage

If you connect only the OceanStor Dorado V6 back-end storage, configure the **initiator** parameter in the Cinder Driver configuration file by referring to **Table 11-3**.

Table 11-3 OceanStor Dorado V6 ALUA parameter configurations

Customer Scenario	Host OS Type	Parameter Configuration
HyperMetro scenario: load balancing mode	CentOS/ RHEL/ Ubuntu/ SUSE/ Debain	You do not need to configure ALUA. That is, delete the ACCESSMODE and HYPERMETROPATHOPTIMIZED parameters from the initiator attribute.

Customer Scenario	Host OS Type	Parameter Configuration
HyperMetro scenario: local preferred mode – preferred path storage	CentOS/ RHEL/ Ubuntu/ SUSE/ Debain	ACCESSMODE="1" HYPERMETROPATHOPTI-MIZED="1"
HyperMetro scenario: local preferred mode – non-preferred path storage	CentOS/ RHEL/ Ubuntu/ SUSE/ Debain	ACCESSMODE="1" HYPERMETROPATHOPTI-MIZED="0"
Non-HyperMetro scenario	CentOS/ RHEL/ Ubuntu/ SUSE/ Debain	You do not need to configure ALUA. That is, delete the ACCESSMODE and HYPERMETROPATHOPTIMIZED parameters from the initiator attribute.

Coexistence of OceanStor Converged Storage and OceanStor Dorado V6 Storage

If you use OceanStor converged storage and then expand the capacity using OceanStor Dorado storage, that is, multiple back-end storage devices are connected, configure the **initiator** parameter in the Cinder Driver configuration file by referring to **Table 11-4**.

Table 11-4 OceanStor Dorado V6 ALUA parameter configurations

Customer Scenario	Host OS Type	Parameter Configuration
HyperMetro scenario: load balancing mode	CentOS/ RHEL/ Ubuntu/ SUSE/ Debain	ACCESSMODE="1" HYPERMETROPATHOPTI-MIZED="1"
HyperMetro scenario: local preferred mode – preferred path storage	CentOS/ RHEL/ Ubuntu/ SUSE/ Debain	ACCESSMODE="1" HYPERMETROPATHOPTI-MIZED="1"
HyperMetro scenario: local preferred mode – non-preferred path storage	CentOS/ RHEL/ Ubuntu/ SUSE/ Debain	ACCESSMODE="1" HYPERMETROPATHOPTI-MIZED="0"

Customer Scenario	Host OS Type	Parameter Configuration
Non-HyperMetro scenario	CentOS/ RHEL/ Ubuntu/ SUSE/ Debain	ACCESSMODE="1" HYPERMETROPATHOPTI-MIZED="1"

11.2 Appendix 2: Configuring Volume-backend image

This function is provided by the native OpenStack platform. It is used to store images in the storage backend of Cinder to accelerate the speed of creating volumes from images. Official link: https://docs.openstack.org/cinder/stein/admin/blockstorage-volume-backed-image.html.

Step 1 Edit the /etc/glance/glance-api.conf file and complete the following configurations:

Configure Cinder as the storage backend of glance and configure the following content in [glance store]:

```
[glance_store]
...
stores = file,http,swift,cinder
default_store = cinder
```

Configure **show_multiple_locations** in [DEFAULT] to parse the locations URL information of Cinder.

```
[DEFAULT]
...
show_multiple_locations = True
```

Step 2 Edit the /etc/cinder/cinder.conf file and complete the following configurations:

Configure **allowed_direct_url_schemes** and **glance_api_version** to use Cinder as the backend of glance.

```
[DEFAULT]
...
glance_api_version=2
allowed_direct_url_schemes = cinder
```

Configure **image_upload_use_cinder_backend** at the backend, for example, [huawei_backend].

```
[huawei_backend]
...
image_upload_use_cinder_backend = True
```

Step 3 After the preceding configuration is complete, restart the cinder and glance services.

```
systemctl restart openstack-glance-*
systemctl restart openstack-cinder-*
```

----End

11.3 Appendix 3: Configuring Image-Volume Cache

This function is provided by the native OpenStack platform. Image volume cache can significantly improve the performance of creating EVS disks from images. Official link: https://docs.openstack.org/cinder/stein/admin/blockstorage-image-volume-cache.html.

Step 1 Obtain the project ID, for example, admin.

openstack project list | grep admin | 366d9f386ecd4defb91df5b0830e7267 | admin |

Step 2 Obtain a Cinder user ID.

openstack user list | grep cinder | 1b2a016cb4a74326a469834dfbc7a841 | cinder

Step 3 Edit the /etc/cinder/cinder.conf file.

Add cinder_internal_tenant_project_id and cinder_internal_tenant_user_id to [DEFAULT].

[DEFAULT] cinder_internal_tenant_project_id = 366d9f386ecd4defb91df5b0830e7267 cinder_internal_tenant_user_id = 1b2a016cb4a74326a469834dfbc7a841

Configure **image_volume_cache_enabled** at the backend, for example, [huawei_backend].

[huawei_backend]
...
image_volume_cache_enabled = True

Step 4 After the preceding configuration is complete, restart the cinder and glance services.

systemctl restart openstack-cinder-*

----End