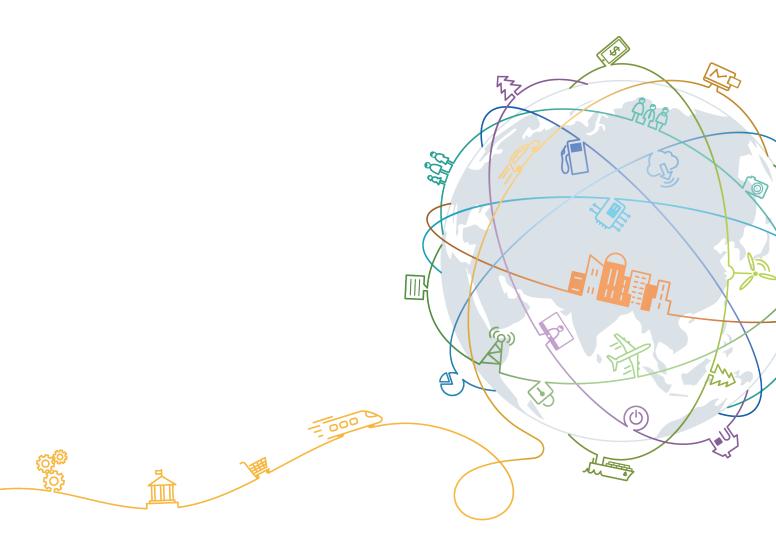
OpenStack Cinder Driver

Configuration Guide

Issue 01

Date 2019-06-21





Copyright © Huawei Technologies Co., Ltd. 2019. 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: http://www.huawei.com
Email: support@huawei.com

Contents

| 1 Overview | 1 |
|---|----|
| 2 Version Mappings | 2 |
| 3 Deployment | 5 |
| 3.1 Obtaining Cinder Driver | 5 |
| 3.2 Deploying Cinder Driver for the Cinder Volume Service in Non-Containerized Mode | |
| 3.3 Deploying Cinder Driver for the Cinder Volume Service in Containerized Mode | 6 |
| 4 Configuring Basic Properties | 8 |
| 4.1 Configuring Cinder Driver for iSCSI-based Storage | 8 |
| 4.1.1 Configuring Cinder Driver for iSCSI-based OceanStor TV1 Storage | 9 |
| 4.1.2 Configuring Cinder Driver for Other iSCSI-based Storage | 11 |
| 4.1.3 Configuring iSCSI Multipathing. | 13 |
| 4.1.4 Configuring CHAP and ALUA | 14 |
| 4.2 Configuring Cinder Driver for FC-based Storage | 16 |
| 4.2.1 Configuring Cinder Driver for FC-based OceanStor TV1 Storage | 16 |
| 4.2.2 Configuring Cinder Driver for Other FC-based Storage. | 18 |
| 4.2.3 Configuring Auto Zoning. | 20 |
| 4.2.4 Configuring FC Multipathing. | 21 |
| 4.2.5 Configuring ALUA | 21 |
| 4.3 Configuring Multiple Backends | 22 |
| 4.4 Configuring a Tenant User | 22 |
| 5 Configuring Advanced Properties | 24 |
| 5.1 Configuring the Thick Property. | 25 |
| 5.2 Configuring the Thin Property. | 25 |
| 5.3 Configuring the SmartQoS Property | 25 |
| 5.4 Configuring the SmartPartition Property. | 27 |
| 5.5 Configuring the SmartCache Property. | 27 |
| 5.6 Configuring the SmartTier Property | 27 |
| 5.7 Configuring the HyperMetro Property | 28 |
| 5.8 Configuring the Remote Replication Property. | 29 |
| 5.9 Creating a LUN in a Specified Storage Pool | 30 |
| 5.10 Creating a LUN on a Disk of a Specified Type | 31 |
| 5.11 Creating a LUN in on a Specified Backend | 31 |

| 5.12 Configuring a Backup Snapshot | 31 |
|--|----|
| 5.13 Configuring Certificate Authentication. | 32 |
| 5.14 Configuring the Owning Controller of a LUN. | 32 |
| 5.15 Configuring the SmartDedupe Property. | 33 |
| 5.16 Configuring the SmartCompression Property | 34 |
| 5.17 Creating a Volume Using fastclone | 35 |
| 5.18 Configuring the LUN Copy Speed. | 35 |
| 5.19 Configuring the Application Type of a LUN | 35 |
| 6 FAQs | 37 |
| 6.1 Do I Need to Create Hosts, Host Groups, LUN Groups, and Other Associated Objects on Huawei S Using Cinder Driver to Map Volumes? | _ |
| 6.2 What Can I Do if Mapping Fails Because Nova Logs Indicate that the lsblk Command Fails to Be E Volume Mapping? | |
| 6.3 Initiator Configuration Policy Example | 37 |

$oldsymbol{1}$ 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.

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

| OpenStack | Huawei Storage |
|-----------|---|
| • Newton | • T Series V100R005/V200R002 |
| • Ocata | OceanStor V3 V300R001/V300R002/V300R003/V300R006 |
| • Pike | OceanStor 2200/2600 V300R005 |
| • Queens | OceanStor V5 V500R007 |
| • Rocky | OceanStor 18500/18800 V100R001/V300R003/V300R006/V500R007 |
| • Stein | OceanStor Dorado V3 V300R001/V300R002 |
| | OceanStor F V3/V5 all-flash storage |

NOTE

- Huawei T series storage devices are no longer supported from the Rocky version.
- 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.

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

| Feature | Mitaka | Newton | Ocata | Pike | Queens | Rocky |
|---------------|--------|--------|----------|----------|--------|----------|
| Create Volume | √ | √ | √ | √ | √ | √ |
| Delete Volume | √ | √ | √ | √ | √ | √ |
| Attach Volume | √ | √ | √ | √ | √ | √ |
| Detach Volume | √ | √ | √ | √ | √ | √ |

| Feature | Mitaka | Newton | Ocata | Pike | Queens | Rocky |
|---|----------|----------|----------|----------|----------|----------|
| Extend Volume | √ | √ | √ | √ | √ | √ |
| Create Snapshot | √ | √ | √ | √ | √ | √ |
| Delete Snapshot | √ | √ | √ | √ | √ | √ |
| Create Volume from Snapshot | √ | ✓ | √ | √ | √ | √ |
| Create Volume from Image | √ | √ | √ | √ | √ | √ |
| Create Volume from Volume | √ | √ | √ | √ | √ | √ |
| Create Image from Volume | √ | √ | √ | √ | √ | √ |
| Volume Migration | √ | √ | √ | √ | √ | √ |
| QoS | √ | √ | √ | √ | √ | √ |
| FC zoning | √ | √ | √ | √ | √ | √ |
| SmartTier | √ | √ | √ | √ | √ | √ |
| SmartCache | √ | √ | √ | √ | √ | √ |
| Thin Volume | √ | √ | √ | √ | √ | √ |
| Thick Volume | √ | √ | √ | √ | √ | √ |
| SmartPartition | √ | √ | √ | √ | √ | √ |
| HyperMetro | √ | √ | √ | √ | √ | √ |
| Retype | √ | √ | √ | √ | √ | √ |
| Manage/Unmanage Volume | √ | √ | √ | √ | √ | √ |
| Manage/Unmanage Snapshot | √ | √ | √ | √ | √ | √ |
| Replication V2.1 | √ | √ | √ | √ | √ | √ |
| Create a volume on a certain storage pool | √ | √ | 1 | √ | √ | √ |
| Create a volume with certain disk type | √ | √ | √ | 1 | √ | 1 |
| Backup Snapshot | √ | √ | √ | √ | √ | √ |
| Multipath | √ | √ | √ | √ | √ | √ |
| Consistency Group | √ | √ | √ | √ | √ | √ |
| Snapshot Consistency Group | √ | √ | √ | √ | √ | √ |

| Feature | Mitaka | Newton | Ocata | Pike | Queens | Rocky |
|----------------------------------|----------|--------|----------|------|--------|----------|
| HyperMetro Consistency Group | √ | √ | √ | 1 | √ | √ |
| Replication Consistency Group | √ | √ | √ | √ | √ | √ |

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.

3 Deployment

- 3.1 Obtaining Cinder Driver
- 3.2 Deploying Cinder Driver for the Cinder Volume Service in Non-Containerized Mode
- 3.3 Deploying Cinder Driver for the Cinder Volume Service in Containerized Mode

3.1 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

NOTE

- 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.

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

Step 1 Delete existing Cinder Driver code in the Cinder Driver code directory. The relative path is .../ *cinder/volume/drivers/huawei*.

MOTE

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

```
# find / -name huawei_driver.py
/usr/lib/python2.7/dist-packages/cinder/volume/drivers/huawei/huawei driver.py
```

- **Step 2** Copy obtained Cinder Driver code files to the directory.
- **Step 3** Make configuration by referring to **4 Configuring Basic Properties**.
- Step 4 Restart Cinder Volume services.

----End

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

Step 1 (Optional) Run the docker save command to back up the Cinder Volume container image.

MOTE

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

Step 2 Run the following command to create temporary directory **build** in any directory and go to the created directory:

```
# mkdir build;cd build
```

Step 3 Run the following commands to 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.

Step 5 Run the following command to build an image:

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

NOTE

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

- **Step 6** Make configuration by referring to **4 Configuring Basic Properties**.
- **Step 7** Restart the Cinder Volume container.

----End

4 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 create a volume from image, the multipath tool must be installed, and add the following configuration keys in the target back end configuration group of the /etc/cinder/ cinder.conf file in Cinder Volume node:

```
use_multipath_for_image_xfer = True
enforce_multipath_for_image_xfer = True
```

- 4.1 Configuring Cinder Driver for iSCSI-based Storage
- 4.2 Configuring Cinder Driver for FC-based Storage
- 4.3 Configuring Multiple Backends
- 4.4 Configuring a Tenant User

4.1 Configuring Cinder Driver for iSCSI-based Storage

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

4.1.1 Configuring Cinder Driver for iSCSI-based OceanStor TV1 Storage

NOTICE

Only OpenStack versions earlier than Rocky support OceanStor TV1 Storage.

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

Table 4-1 Mandatory parameters

| Parameter | Description |
|-----------------|--|
| Product | Type of a storage product. Set this parameter to T for an OceanStor TV1 storage device. |
| Protocol | Type of a connection protocol. Set this parameter to iSCSI for iSCSI scenarios. |
| ControllerIP0 | IP address of the primary controller on the OceanStor TV1 storage device. |
| ControllerIP1 | IP address of the secondary controller on the OceanStor TV1 storage device. |
| UserName | User name of a storage administrator. |
| UserPassword | Password of the storage administrator. |
| StoragePool | Name of a storage pool to be used. |
| DefaultTargetIP | Default IP address of the iSCSI target port provided for compute nodes by the storage device. |

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

Table 4-2 Optional parameters

| Parameter | Default Value | Description |
|----------------|---------------|--|
| LUNType | Thick | Type of the LUN to be created by default. The value can be Thick or Thin . |
| StripUnitSize | 64 | Stripe depth of the LUN. The value can be 4, 8, 16, 32, 64, 128, 256, or 512, expressed in KB. This parameter is invalid when a thin LUN is created. |
| 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. |
| HostIP | - | IP address of the Nova compute node. |
| OSType | Linux | Operating system of the Nova compute node. |

Step 4 Change the owner and user group of the Cinder Driver configuration file to 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:

```
[TV1_iSCSI]
volume_driver = cinder.volume.drivers.huawei.huawei_t.HuaweiTISCSIDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml
volume_backend_name = TV1_iSCSI
```

NOTE

- Backend name TV1_iSCSI can be customized.
- **volume_driver** indicates the type of the driver to be used. Set this parameter to **cinder.volume.drivers.huawei_t.HuaweiTISCSIDriver**.
- **cinder_huawei_conf_file** indicates the path of the Cinder Driver configuration file.

In the DEFAULT area, set enabled_backends to TV1_iSCSI.

```
[DEFAULT]
...
enabled_backends=TV1_iSCSI
```

Step 6 Restart the Cinder Volume service.

----End

4.1.2 Configuring Cinder Driver for Other iSCSI-based Storage

- **Step 1** In /etc/cinder, create a Cinder Driver configuration file in XML format. The configuration file name can be customized, for example, cinder huawei conf.xml.
- **Step 2** Configure mandatory parameters in the Cinder Driver configuration file.

Table 4-3 Mandatory parameters

| Parameter | Description |
|-----------------|--|
| Product | Type of a storage product. The value can be TV2 , 18000 , 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 (;). |
| DefaultTargetIP | Default IP address of the iSCSI target port provided for compute nodes by the storage device. For Rocky and later versions, multiple IP addresses can be configured and are separated by semicolons (;). |

NOTICE

Set **Product** based on the actual back-end storage model.

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

```
<Timeout>***</Timeout>
<LUNCopySpeed>***</LUNCopySpeed>
<LUNCloneMode>***</LUNCloneMode>
<HyperSyncSpeed>***</HyperSyncSpeed>
<ReplicaSyncSpeed>***</ReplicaSyncSpeed>
</LUN>
```

Table 4-4 Optional parameters

| Parameter | Default Value | Description |
|---------------------|--|---|
| LUNType | Thin (OceanStor Dorado)Thick (Others) | Type of the LUN to be created by default. The value can be Thick or Thin . For OceanStor Dorado, the value must be Thin . |
| 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. |
| LUNCopySpeed | 2 | LUN copy speed. The value can be 1 (low), 2 (medium), 3 (high), or 4 (highest). |
| LUNCloneMode | luncopy | LUN clone mode. The value can be fastclone or luncopy. Only OceanStor Dorado supports fastclone. OceanStor Dorado V300R001C20 and later versions support luncopy. Other storage devices support luncopy only. |
| 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). |

Step 4 Change the owner and user group of the Cinder Driver configuration file to 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
```

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.HuaweiISCSIDriver.
- **cinder_huawei_conf_file** indicates the path of the Cinder Driver configuration file.

In the **DEFAULT** area, set **enabled** backends to huawei backend.

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

Step 6 Restart the Cinder Volume service.

----End

4.1.3 Configuring iSCSI Multipathing

If you need to configure multipathing for iSCSI, configure it as follows:

Procedure for Queens and Earlier Versions

- **Step 1** Create a port group on the storage device using DeviceManager and add service links that require multipathing to the port group.
- **Step 2** Add the port group settings in the Cinder Driver configuration file and configure the port group name needed by an initiator.

Method 1: Use the initiator name for exact configuration.

Method 2: Use the regular expression of the host name for configuration.

```
<isCSI>
     <Initiator HostName="*****" TargetPortGroup="***"/>
     <Initiator HostName="*****" TargetPortGroup="***"/>
</isCSI>
```

- **Step 3** Enable multipathing for the OpenStack Nova module.
 - 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 4 Restart the nova-compute service.

----End

Procedure for Rocky and Later Versions

Step 1 Configure multiple IP addresses in the **DefaultTargetIP** item of the Cinder Driver configuration file.

Method 1: Configure multiple IP addresses in the **DefaultTargetIP** item in the Cinder Driver configuration file.

Method 2: Configure multiple IP addresses in the **TargetIP** item for individual initiators.

```
<isCSI>
     <Initiator Name="*****" TargetIP="x.x.x.x;y.y.y.y"/>
     <Initiator Name="*****" TargetIP="x.x.x.x;y.y.y.y"/>
</isCSI>
```

Method 3: For multiple iSCSI initiators, use the regular expression of the host name to configure IP addresses.

Step 2 Enable multipathing for the OpenStack Nova module.

Add volume_use_multipath = True in [libvirt] of /etc/nova/nova.conf.

Step 3 Restart the nova-compute service.

NOTE

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.

Configuration rules:

- Use exact configuration. For details, see example 1 in 6.3 Initiator Configuration Policy Example.
 - For the same host, if the corresponding configuration can be found by both the initiator name and host name, use the initiator configuration.
 - For the same host, if both the common configuration and general configuration are met, use the general configuration.
 - Priority: Initiator name configuration > General configuration of host name > Common configuration of host name
- Use the first configuration item (for the common configuration of the host name). For details, see example 2 in 6.3 Initiator Configuration Policy Example.

----End

4.1.4 Configuring CHAP and ALUA

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.

Step 1 Add the following contents to the Cinder Driver configuration file:

Table 4-5 Parameter description

| Parameter | Default Value | Description |
|-----------------|---------------|---|
| Name | - | iSCSI initiator name. This parameter is optional. Either this parameter or HostName must be specified. |
| HostName | - | Regular expression of the host name. This parameter is optional. Either this parameter or Name must be specified. |
| ALUA | 0 | 0: Huawei UltraPath (default value) 1: third-party multipathing software |
| FAILOVERMODE | - | Initiator switchover mode. This parameter needs to be delivered only when third-party multipathing software is used. |
| SPECIALMODETYPE | - | Special mode type of the initiator. This parameter needs to be delivered only when initiator switchover mode is special ALUA. |
| РАТНТҮРЕ | - | Initiator path type. |
| CHAPinfo | - | User name and password authenticated by CHAP in the format of mm-user;mm-user@storage1. The user name and password are separated by a semicolon (;). The CHAP name must contain 4 to 223 characters. |
| | | The password must contain 12 to 16 characters |
| | | The password must contain any three of the following types: uppercase letters, lowercase letters, digits, and special characters including `~!@#\$%^*()=+\ [{}];:,./? and spaces |
| | | The password cannot be the same as the CHAP name or CHAP name spelled backwards. |

NOTE

Because the value ranges of parameters **FAILOVERMODE**, **SPECIALMODETYPE**, and **PATHTYPE** vary according to storage devices, configure the parameters by referring to the API document of the corresponding Huawei storage device.

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.

Configuration rules:

- Use exact configuration. For details, see example 1 in 6.3 Initiator Configuration Policy Example.
 - For the same host, if the corresponding configuration can be found by both the initiator name and host name, use the initiator configuration.
 - For the same host, if both the common configuration and general configuration are met, use the general configuration.
 - Priority: Initiator name configuration > General configuration of host name > Common configuration of host name
- Use #EN-US_TOPIC_0163269088/li944112795617 (when the host name is not configured in common mode).
- Use the first configuration item (for the common configuration of the host name). For details, see example 2 in 6.3 Initiator Configuration Policy Example.

----End

4.2 Configuring Cinder Driver for FC-based Storage

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

4.2.1 Configuring Cinder Driver for FC-based OceanStor TV1 Storage

NOTICE

Only OpenStack versions earlier than Rocky support OceanStor TV1 Storage.

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

Table 4-6 Mandatory parameters

| Parameter | Description |
|---------------|---|
| Product | Type of a storage product. Set this parameter to T for an OceanStor TV1 storage device. |
| Protocol | Type of a connection protocol. Set this parameter to FC for FC scenarios. |
| ControllerIP0 | IP address of the primary controller on the OceanStor TV1 storage device. |
| ControllerIP1 | IP address of the secondary controller on the OceanStor TV1 storage device. |
| UserName | User name of a storage administrator. |
| UserPassword | Password of the storage administrator. |
| StoragePool | Name of a storage pool to be used. |

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

Table 4-7 Optional parameters

| Parameter | Default Value | Description |
|----------------|---------------|--|
| LUNType | Thick | Type of the LUN to be created by default. The value can be Thick or Thin . |
| StripUnitSize | 64 | Stripe depth of the LUN. The value can be 4, 8, 16, 32, 64, 128, 256, or 512, expressed in KB. This parameter is invalid when a thin LUN is created. |
| 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. |
| HostIP | - | IP address of the Nova compute node. |
| OSType | Linux | Operating system of the Nova compute node. |

Step 4 Change the owner and user group of the Cinder Driver configuration file to 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:

```
[TV1_FC]
volume_driver = cinder.volume.drivers.huawei.huawei_t.HuaweiTFCDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml
volume_backend_name = TV1_FC
```

NOTE

- Backend name TV1 FC can be customized.
- **volume_driver** indicates the type of the driver to be used. Set this parameter to **cinder.volume.drivers.huawei_t.HuaweiTFCDriver**.
- cinder_huawei_conf_file indicates the path of the Cinder Driver configuration file.

In the DEFAULT area, set enabled_backends to TV1_FC.

```
[DEFAULT]
...
enabled_backends=TV1_FC
```

Step 6 Restart the Cinder Volume service.

----End

4.2.2 Configuring Cinder Driver for Other FC-based Storage

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

Table 4-8 Mandatory parameters

| Parameter | Description |
|-----------|---|
| Product | Type of a storage product. The value can be TV2 , 18000 , V3 , V5 , or Dorado . |
| Protocol | Type of a connection protocol. Set this parameter to FC for FC 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. |

| Parameter | Description |
|--------------|--|
| 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 (;). |

NOTICE

Set **Product** based on the actual back-end storage model.

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

Table 4-9 Optional parameters

| Parameter | Default Value | Description |
|---------------------|---------------------------------|--|
| LUNType | • Thin (OceanStor Dorado) | Type of the LUN to be created by default. The value can be Thick or Thin . For OceanStor Dorado, the value must be Thin . |
| | • Thick (Others) | |
| 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 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. |
| LUNCopySpeed | 2 | LUN copy speed. The value can be 1 (low), 2 (medium), 3 (high), or 4 (highest). |

| Parameter | Default Value | Description |
|------------------|---------------|---|
| LUNCloneMode | luncopy | LUN clone mode. The value can be fastclone or luncopy . Only OceanStor Dorado supports fastclone . OceanStor Dorado V300R001C20 and later versions support luncopy . Other storage devices support luncopy only. |
| 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). |

Step 4 Change the owner and user group of the Cinder Driver configuration file to 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
```

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.

In the DEFAULT area, set enabled_backends to huawei_backend.

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

Step 6 Restart the Cinder Volume service.

----End

4.2.3 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.

4.2.4 Configuring FC Multipathing

If you need to configure multipathing for FC, configure it as follows:

- **Step 1** Enable multipathing for the OpenStack Nova module.
 - 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.

----End

4.2.5 Configuring ALUA

If you need to configure ALUA for FC, configure it as follows:

Step 1 Add the following contents to the Cinder Driver configuration file:

Table 4-10 Parameter description

| Parameter | Default Value | Description |
|-----------------|---------------|---|
| Name | - | FC initiator WWN. This parameter is optional. Either this parameter or HostName must be specified. |
| HostName | - | Regular expression of the host name. This parameter is optional. Either this parameter or Name must be specified. |
| ALUA | 0 | 0: Huawei UltraPath (default value) 1: third-party multipathing software |
| FAILOVERMODE | - | Initiator switchover mode. This parameter needs to be delivered only when third-party multipathing software is used. |
| SPECIALMODETYPE | - | Special mode type of the initiator. This parameter needs to be delivered only when initiator switchover mode is special ALUA. |

| Parameter | Default Value | Description |
|-----------|---------------|----------------------|
| РАТНТҮРЕ | - | Initiator path type. |

NOTE

Because the value ranges of parameters **FAILOVERMODE**, **SPECIALMODETYPE**, and **PATHTYPE** vary according to storage devices, configure the parameters by referring to the API document of the corresponding Huawei storage device.

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.

Configuration rules:

- Use exact configuration. For details, see example 1 in 6.3 Initiator Configuration Policy Example.
 - For the same host, if the corresponding configuration can be found by both the initiator name and host name, use the initiator configuration.
 - For the same host, if both the common configuration and general configuration are met, use the general configuration.
 - Priority: Initiator name configuration > General configuration of host name > Common configuration of host name
- Use the first configuration item (for the common configuration of the host name). For details, see example 2 in 6.3 Initiator Configuration Policy Example.

----End

4.3 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
```

4.4 Configuring a Tenant User

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

Table 4-11 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 multitenant feature.

5 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.

- 5.1 Configuring the Thick Property
- 5.2 Configuring the Thin Property
- 5.3 Configuring the SmartQoS Property
- 5.4 Configuring the SmartPartition Property
- 5.5 Configuring the SmartCache Property
- 5.6 Configuring the SmartTier Property
- 5.7 Configuring the HyperMetro Property
- 5.8 Configuring the Remote Replication Property
- 5.9 Creating a LUN in a Specified Storage Pool
- 5.10 Creating a LUN on a Disk of a Specified Type
- 5.11 Creating a LUN in on a Specified Backend
- 5.12 Configuring a Backup Snapshot
- 5.13 Configuring Certificate Authentication
- 5.14 Configuring the Owning Controller of a LUN
- 5.15 Configuring the SmartDedupe Property
- 5.16 Configuring the SmartCompression Property
- 5.17 Creating a Volume Using fastclone
- 5.18 Configuring the LUN Copy Speed

5.19 Configuring the Application Type of a LUN

5.1 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

5.2 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

5.3 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 5-1 IOType parameter

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

Table 5-2 Optional qos_key parameters

| Parameter | Description | Remarks |
|--------------|--------------------|---|
| maxIOPS | Maximum IOPS. | The value is a positive integer. |
| 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. |

NOTE

- 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 qos-associate <qos_specs> <volume_type_id>
- **Step 5** Use the volume type in **Step 1** to create a volume.
 - ----End

5.4 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

5.5 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

5.6 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=***
```

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

5.7 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.

```
hypermetro_device =
storage_pool:***,
san_address:https://*.*.*.*:8088/deviceManager/rest/,
san_user:***,
san_password:***,
vstore_name:***,
iscsi_default_target_ip:*.*.*,
metro_domain:***
```

Table 5-3 Parameter description

| Parameter | Description |
|-------------------------|--|
| storage_pool | Name of the HyperMetro remote storage pool. |
| san_address | Access address of the REST interface. |
| san_user | User name of a storage administrator or tenant of the HyperMetro remote device. |
| san_password | Password of a storage administrator or tenant of the HyperMetro remote device. |
| vstore_name | vStore name of the HyperMetro remote device. This parameter needs to be configured only when a tenant user is used. |
| metro_domain | HyperMetro domain name. |
| iscsi_default_target_ip | IP address of the HyperMetro remote service port. For Rocky and later versions, multiple IP addresses can be configured and are separated by semicolons (;). |

If the HyperMetro remote device needs to use iSCSI multipathing, add the following contents in the configuration information of the remote device:

```
TargetIP:x.x.x.x;TargetPortGroup:xxx};
{HostName:xxx;CHAPinfo:user#password;ALUA:1;
   TargetIP:*.*.*.;TargetPortGroup:***};
{HostName:xxx;CHAPinfo:user#password;ALUA:1;
   TargetIP:*.*.*.;TargetPortGroup:***}
```

MOTE

For details about **iscsi_info** parameters and configuration rules, see **4.1.3 Configuring iSCSI Multipathing** and **4.1.4 Configuring CHAP and ALUA**.

- **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

5.8 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 default target ip:*.*.**
```

Table 5-4 Parameter description

| Parameter | Description |
|--------------|---|
| backend_id | Target device ID. This parameter must be specified in the event of failover. |
| storage_pool | Storage pool name of the target backend in the event of failover. |
| san_address | Access address of the REST interface. |
| san_user | User name of a storage administrator or tenant of the remote replication remote device. |
| san_password | Password of a storage administrator or tenant of the remote replication remote device. |
| vstore_name | vStore name of the remote replication remote device. This parameter needs to be configured only when a tenant user is used. |

| Parameter | Description |
|-------------------------|--|
| iscsi_default_target_ip | IP address of the remote replication remote service port. For Rocky and later versions, multiple IP addresses can be configured and are separated by semicolons (;). |

If the remote replication remote device needs to use iSCSI multipathing, add the following contents in the configuration information of the remote device:

NOTE

For details about **iscsi_info** parameters and configuration rules, see **4.1.3 Configuring iSCSI Multipathing** and **4.1.4 Configuring CHAP and ALUA**.

- **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

5.9 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

5.10 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"
```

MOTE

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

5.11 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=***
```

MNOTE

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

5.12 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

5.13 Configuring Certificate Authentication

This section describes how to configure certificate authentication to connect back-end storage.

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

MNOTE

- 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

5.14 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

5.15 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

MNOTE

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

5.16 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

MNOTE

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

5.17 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

NOTE

- If **fastclone** is not set, LUN copy is used by default.
- Only OceanStor Dorado supports fastclone.

5.18 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.

Step 1 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
```

----End

Table 5-5 copyspeed parameter value description

| Parameter Value | Description |
|-----------------|-------------|
| 1 | Low |
| 2 | Medium |
| 3 | High |
| 4 | Highest |

NOTE

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

5.19 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
 - 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 configed, Configuration the application type takes precedence.

6 FAQs

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

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

6.3 Initiator Configuration Policy Example

6.1 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.

6.2 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.

6.3 Initiator Configuration Policy Example

Configuration file example 1:

```
<iSCSI>
<Initiator Name="iqn" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx" CHAPinfo="xxx"
TargetPortGroup="***"/>
<Initiator HostName="*" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"
CHAPinfo="xxx" TargetPortGroup="***"/>
```

```
<Initiator HostName="host" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"
CHAPinfo="xxx" TargetPortGroup="***"/>
</iscsi>
```

If the host name is **host** and the initiator name is **iqn**, the preceding three configuration items can be used to configure ALUA and port group information for the initiator. According to the rules (based on the exact configuration), we will definitely select the initiator name (the first configuration item), then (when the first configuration item does not exist) the general configuration of the host name (the third configuration item), and finally (when the first and third configuration items do not exist) the common configuration of the host name (the second configuration item).

Configuration file example 2:

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

If the host whose 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.

NOTE

For details about the configuration rules, see 4.1.3 Configuring iSCSI Multipathing and 4.1.4 Configuring CHAP and ALUA.