**eSDK Enterprise Storage Plugins
2.5.4**

# OpenStack Cinder Driver Configuration Guide

**Issue**     01

**Date**     2023-04-17

**HUAWEI TECHNOLOGIES CO., LTD.**

# Huawei Technologies Co., Ltd.

| | |
|---|---|
| Address: | Huawei Industrial Base<br>Bantian, Longgang<br>Shenzhen 518129<br>People's Republic of China |
| Website: | https://e.huawei.com |

# About This Document

## Intended Audience

This document is intended for:

- Technical support engineers
- O&M engineers
- Engineers with basic knowledge of storage and OpenStack

## Symbol Conventions

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

| Symbol | Description |
|---|---|
| ⚠ **DANGER** | Indicates a hazard with a high level of risk which, if not avoided, will result in death or serious injury. |
| ⚠ **WARNING** | Indicates a hazard with a medium level of risk which, if not avoided, could result in death or serious injury. |
| ⚠ **CAUTION** | Indicates a hazard with a low level of risk which, if not avoided, could result in minor or moderate injury. |
| **NOTICE** | Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury. |
| 📖 NOTE | Supplements the important information in the main text. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration. |

# Change History

| Issue | Date | Description |
|-------|------|-------------|
| 01 | 2023-04-17 | This issue is the first official release. |

# Contents

# 1 Overview

Huawei Cinder Driver is a plug-in deployed on the OpenStack Cinder module. It connects to Huawei storage to provide volume storage capabilities for the OpenStack platform. Huawei Cinder Driver supports iSCSI and FC protocols.

# 2 Version Mappings

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

## 2.1 Storage Product Versions Supported by Cinder Driver

**Table 2-1** Storage product versions supported by Cinder Driver

| OpenStack Version | Storage Product Version |
|---|---|
| Mitaka/Newton/Ocata/Pike/ Queens/Rocky/Stein/Train/Ussuri/ Victoria/Wallaby/Xena/Yoga/Zed | <ul><li>OceanStor V3 V300R006C60/ V300R006C61</li><li>OceanStor V5 V500R007C61/ V500R007C70/V500R007C71</li><li>OceanStor V6 6.1.3/6.1.5</li><li>OceanStor Dorado V3 V300R002C10/ V300R002C20</li><li>OceanStor Dorado V6 6.0.0/6.0.1/6.1.0/6.1.2/6.1.3/6.1.5</li></ul> |

☐ NOTE

Huawei only maintains Cinder Driver for 14 stable OpenStack versions (M-Z). Earlier versions can still be obtained from Huawei warehouse but are no longer maintained. Exercise caution when selecting the version.

# 2.2 Support for Cinder Driver Features

**Table 2-2** Support for Cinder Driver features (√: supported; x: not supported)

| Feature | Normal LUN | HyperMetro LUN | Remarks |
|---|---|---|---|
| OpenStack | Mitaka/ Newton/ Ocata/Pike/ Queens/ Rocky/Stein/ Train/Ussuri/ Victoria/ Wallaby/ Xena/ Yoga/Zed | Mitaka/ Newton/ Ocata/Pike/ Queens/ Rocky/Stein/ Train/Ussuri/ Victoria/ Wallaby/ Xena/ Yoga/Zed | ● The following features are applicable to scenarios where HyperMetro storage is normal. <br> ● √ indicates that a feature is supported by all OpenStack versions listed on the left but not supported by other OpenStack versions. <br> ● If some features are not supported by a specific version, they will be listed separately in brackets (). <br> ● x indicates that a feature is not supported by any OpenStack version. |
| Create Volume | √ | √ | N/A |
| Delete Volume | √ | √ | N/A |
| Attach Volume | √ | √ | N/A |
| Detach Volume | √ | √ | N/A |
| Extend Volume | √ | √ | N/A |
| Create Snapshot | √ | √ | Snapshots can be created only for the local HyperMetro LUN. |
| Delete Snapshot | √ | √ | Snapshots can be deleted only for the local HyperMetro LUN. |
| Create Volume from Snapshot | √ | √ | Volumes can be created using snapshots only for the local HyperMetro LUN. |
| Create Volume from Image | √ | √ | N/A |

| Feature | Normal LUN | HyperMetro LUN | Remarks |
|---|---|---|---|
| Create Volume from Volume | √ | √ | N/A |
| Create Image from Volume | √ | √ | N/A |
| Volume Migration | √ | x | N/A |
| Thin Volume | √ | √ | N/A |
| Thick Volume | √ | √ | N/A |
| QoS | √ | √ | SmartQoS can be configured only for the local HyperMetro LUN. |
| FC Zoning | √ | √ | N/A |
| Manage/ Unmanage Volume | √ | x | N/A |
| Manage/ Unmanage Snapshot | √ | x | N/A |
| Replication V2.1 | √ | x | N/A |
| Backup Snapshot | √ | √ | Snapshots can be backed up only for the local HyperMetro LUN. |
| Revert to Snapshot | √ (not supported by the Mitaka version) | x | N/A |
| Retype | √ | √ | N/A |
| Multipath | √ | √ | N/A |
| Multi-Attach | √ | √ | N/A |
| Consistency Group | √ | x | N/A |
| Snapshot Consistency Group | √ | √ | Snapshots can be created only for the local HyperMetro LUN. |
| HyperMetro Consistency Group | x | √ | N/A |
| Replication Consistency Group | √ | x | N/A |

| Feature | Normal LUN | HyperMetro LUN | Remarks |
|---|---|---|---|
| SmartTier | √ | √ | SmartTier can be configured only for the local HyperMetro LUN. |
| SmartCache | √ | √ | SmartCache can be configured only for the local HyperMetro LUN. |
| SmartPartition | √ | √ | SmartPartition can be configured only for the local HyperMetro LUN. |
| Create a volume on a certain storage pool | √ | x | N/A |
| Create a volume with certain disk type | √ | x | N/A |
| Tempest test_force_detach | √ | √ | The community Tempest script can be forcibly detached. |

> **NOTICE**
>
> ● OceanStor Dorado does not support Thick Volume, SmartTier, SmartCache, and SmartPartition.
>
> ● Entry-level storage devices do not support some value-added features. For details, see the feature specifications of the specific storage device model.
>
> For details about the specifications of different storage models, see **https://info.support.huawei.com/storage/spec/#/home**.

# 2.3 Mappings Between Cinder Driver Features and Storage Licenses

**Table 2-3** Mappings between Cinder Driver features and storage licenses

| Feature | Storage License |
|---|---|
| Create Volume | N/A |
| Delete Volume | N/A |
| Attach Volume | N/A |

| Feature | Storage License |
|---|---|
| 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 |
| 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 |

| Feature | Storage License |
|---|---|
| HyperMetro Consistency Group | HyperMetro |
| Replication Consistency Group | HyperReplication |

# 3 Specifications

**Table 3-1** Specifications

| Feature | Sub-feature | Description | Remarks |
|---|---|---|---|
| Mapping platform | Native OpenStack platform | OpenStack versions: Mitaka/Newton/ Ocata/Pike/ Queens/Rocky/ Stein/Train/ Ussuri/Victoria/ Wallaby/Xena/ Yoga/Zed | • Resources created using OpenStack, such as hosts, host groups, LUNs, LUN groups, snapshots, mapping views, QoS policies, and HyperMetro pairs, cannot be shared with other platform applications or manually modified.<br>• The storage access address, user name, password, and storage pool information provided for OpenStack cannot be modified. |

| Feature | Sub-feature | Description | Remarks |
|---|---|---|---|
| Configuration | XML configuration file | Ensure that the storage pool used for configuring Cinder Driver exists on Huawei storage. Otherwise, you need to manually create one. In addition, the type of this storage pool must be the 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 Huawei Cinder Driver configuration file cannot contain the following XML special characters: <>&'" | - |
| | | Set **Product** based on the actual back-end storage model. | - |

| Feature | Sub-feature | Description | Remarks |
|---------|-------------|-------------|---------|
|  |  | The value of **HostName** in the **Initiator** configuration item is a regular expression. | ● For details about how to configure it, see **Regular expression**.<br>● When **HostName** is set to **\***, the common configuration is used and takes effect on hosts with any name. When **HostName** is set to another value, the general configuration is used.<br>● Configuration policy rules:<br>  – Priority: General host name configuration > Common host name configuration.<br>  – In the general configuration, use the first ALUA section that meets the configuration policy. |
|  | Multipathing configuration | Host multipathing: You need to manually install multipathing software and enable the multipathing service. | - |

| Feature | Sub-feature | Description | Remarks |
|---------|-------------|-------------|---------|
|  |  | 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. | - |
| Policy | SmartQoS | Different protection policy parameters or restriction policy parameters can be specified at the same time. However, protection policy parameters cannot coexist with restriction policy parameters. | ● Restriction policy parameters: **maxIOPS** and **maxBandWidth**<br>● Protection policy parameters: **minIOPS**, **minBandWidth**, and **latency** |
|  |  | For OceanStor Dorado, **IOType** must be set to **2**, and only restriction policy parameters are supported. |  |
| Data copy | FastClone | Only some storage devices support FastClone (fast data clone). | ● OceanStor Dorado: 6.1.0 and later versions<br>● OceanStor Dorado V3: V300R001C20 and later versions<br>● OceanStor V6: 6.1.3/6.1.5<br><br>When FastClone is used, there are restrictions on the HyperMetro, remote replication, snapshot, and clone features. |

| Feature | Sub-feature | Description | Remarks |
|---------|-------------|-------------|---------|
| | Full data copy | The copy speed can be configured. The default LUN copy speed is medium. | <ul><li>Low: 0 MB/s to 5 MB/s</li><li>Medium: 10 MB/s to 20 MB/s</li><li>High: 50 MB/s to 70 MB/s</li><li>Highest: above 100 MB/s</li></ul> |
| License | Creating a snapshot | The license of HyperSnap is required. | - |
| | Creating a QoS volume | The license of SmartQoS is required. | - |
| | Creating a HyperMetro volume | The license of HyperMetro is required. | - |
| | Creating a volume using a snapshot | The license of HyperCopy/ HyperClone is required. | - |
| | Creating a mirrored volume | The license of HyperSnap & HyperCopy/ HyperClone is required. | - |
| | Creating a clone volume | The license of HyperSnap & HyperCopy/ HyperClone is required. | - |
| RESTful request | Maximum number of RESTful requests that can be sent by a Cinder Driver to a storage device at the same time | 20 | It is recommended that each storage device be connected to two Cinder Drivers. |

# 4 Installing Cinder Driver

> **NOTICE**
>
> - Resources created using OpenStack, such as hosts, host groups, LUNs, LUN groups, snapshots, mapping views, QoS policies, 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 Huawei Cinder Driver

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

- Method 1: OpenStack community warehouse.

  Since the Kilo version, Huawei Cinder Driver has been integrated into the OpenStack community warehouse. After OpenStack is installed, Cinder Driver is provided and stored in the *../cinder/cinder/volume/drivers/huawei* directory.

- Method 2: Huawei warehouse. Perform the following steps.

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

**Step 2** Click **Download ZIP** to download Huawei Cinder Driver package to a local PC.

**Step 3** Decompress the package.

**Step 4** In the decompressed directory, find the **Cinder** directory (path: *../ OpenStack_Driver-master/Cinder*), which contains Huawei 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 Cinder 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 2 stable versions. Huawei OpenStack Cinder Driver warehouse maintains 14 stable versions (M-Z), ensuring long-term stable running of historical versions.
- It is strongly recommended that OpenStack community versions be replaced by Huawei warehouse versions.

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

**Step 1** Use a remote access tool (for example, PuTTY) to log in to an OpenStack Cinder Volume node through the management IP address, and search for the original Huawei Cinder Driver code in the system. For details about how to search for the code, see the following note.

◻ **NOTE**

The absolute path may vary in different systems. You can search for the code directory of Huawei Cinder Driver as follows:

- Method 1:

  Run the following command. In the command output, **/usr/lib/python2.7/site-packages/cinder/volume/drivers/huawei** is the code directory.
  ```
  # python -c "from cinder.volume.drivers import huawei; print (huawei.__path__)"
  ['/usr/lib/python2.7/site-packages/cinder/volume/drivers/huawei']
  ```

- Method 2:

  Run the following command. In the command output, **/usr/lib/python3.7/site-packages/cinder/volume/drivers/huawei** is the code directory.
  ```
  # python3 -c "from cinder.volume.drivers import huawei; print (huawei.__path__)"
  ['/usr/lib/python3.7/site-packages/cinder/volume/drivers/huawei']
  ```

**Step 2** Delete all Huawei Cinder Driver code files in the directory returned in **Step 1**.

**Step 3** Copy all Huawei Cinder Driver code files of the corresponding version to the directory returned in **Step 1**. The minimum permission required for the Cinder Driver code files is **644**. (**644** is a Linux system permission. **-rw-r--r--** indicates the minimum permission required.)

```
# 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 hypermetro.py
-rw-r--r-- 1 root root      0 Mar  4 23:40 __init__.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 a node that has a Cinder Volume container image and run the **docker image ls** command to query the Cinder Volume container image.

```
[root@control huawei]# docker_image ls | grep cinder
kolla/centos-source-cinder-volume         train-centos8    a9e9af6cca02    15 months ago    1.25GB
kolla/centos-source-cinder-api            train-centos8    2dbb505cf434    15 months ago    1.1GB
kolla/centos-source-cinder-backup         train-centos8    e34a428212b3    15 months ago    1.09GB
kolla/centos-source-cinder-scheduler      train-centos8    c5329adb17b9    15 months ago    1.06GB
[root@control huawei]#
```

**Step 2**  Run the **docker save -o cinder_volume.tar kolla/centos-source-cinder-volume** command to back up the Cinder Volume container image to an image file.

```
[root@control ~]# docker save -o cinder_volume.tar  kolla/centos-source-cinder-volume
```

📖 **NOTE**

> **cinder_volume.tar** is a user-defined name of the backup image, and **kolla/centos-source-cinder-volume** is the name of the Cinder Volume container image.

**Step 3**  Create a temporary directory named **huawei** in any directory.

```
# mkdir huawei
```

**Step 4**  Copy the obtained Huawei Cinder Driver code files of the corresponding version to the **huawei** directory. The minimum permission required for the Cinder Driver code files is **644**.

```
# 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 hypermetro.py
-rw-r--r-- 1 root root      0 Mar  4 23:40 __init__.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
```

**Step 5**  Run the **docker ps | grep cinder** command to list Cinder Volume containers. In the command output, **cinder_volume** indicates the container name.

```
[root@control ~]# docker ps | grep cinder
681a72350b38    kolla/centos-source-cinder-backup:train-centos8      "dumb-init --single-…"   15 months ago    Up 25 hours                  cinder_backup
2a1e28d9a4a7    kolla/centos-source-cinder-volume:train-centos8      "dumb-init --single-…"   15 months ago    Up 25 hours                  cinder_volume
d2da14bd6c17    kolla/centos-source-cinder-scheduler:train-centos8   "dumb-init --single-…"   15 months ago    Up 25 hours                  cinder_scheduler
0d0d27771db5    kolla/centos-source-cinder-api:train-centos8         "dumb-init --single-…"   15 months ago    Up 25 hours                  cinder_api
```

**Step 6**  Run the **docker exec -it -u 0 cinder_volume bash** command to access the **cinder_volume** container.

```
[root@control ~]# docker exec -it -u 0 cinder_volume bash
(cinder-volume)[root@control /]#
```

**Step 7** Run the **python -c "from cinder.volume import drivers; print (drivers.\_\_path\_\_)"** command to query the path of the original Huawei Cinder Driver code.

```
(cinder-volume)[root@control /]# python -c "from cinder.volume import drivers; print (drivers.__path__)"
['/var/lib/kolla/venv/lib/python3.6/site-packages/cinder/volume/drivers']
(cinder-volume)[root@control /]#
```

📖 NOTE

- **cinder_volume** is the container name obtained in **Step 5**.
- **/var/lib/kolla/venv/lib/python3.6/site-packages/cinder/volume/drivers** is the path of the original Huawei Cinder Driver code.

**Step 8** Run the **exit** command to exit the **cinder_volume** container, go to the upper-level directory of the **huawei** directory created in **Step 3**, and copy the **huawei** directory to the path of the original Huawei Cinder Driver code in **Step 7**.

```
(cinder-volume)[root@control /]# exit
exit
[root@control ~]# docker cp huawei/ cinder_volume:/var/lib/kolla/venv/lib/python3.6/site-packages/cinder/volume/drivers
[root@control ~]#
```

docker cp huawei/ cinder_volume:/var/lib/kolla/venv/lib/python3.6/site-packages/cinder/volume/drivers

**Step 9** Run the following command to restart the Cinder Volume service (**cinder_volume** indicates the container name obtained in **Step 5**).

docker restart cinder_volume

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+---------------+-----------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| Binary        | Host                  | Zone | Status  | State | Updated_at                 | Cluster | Disabled Reason | Backend State |
+---------------+-----------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| cinder-volume | openstack-wallaby-1@v6 | nova | enabled | up    | 2021-12-03T07:56:23.000000 | -       | -               | up            |
+---------------+-----------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
```

**----End**

# 5 Upgrading Cinder Driver

**NOTICE**

- To upgrade OpenStack Cinder Driver, perform the operations described in this chapter. If you install Cinder Driver for the first time, skip this chapter.
- During the upgrade, do not perform other operations on the management plane, such as delivering VMs or creating volumes.

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

**Procedure**

**Step 1** Use a remote access tool (for example, PuTTY) to log in to an OpenStack Cinder Volume node through the management IP address, and search for the original Huawei Cinder Driver code in the system. For details about how to search for the code, see the following note.

📖 **NOTE**

The absolute path may vary in different systems. You can search for the code directory of Huawei Cinder Driver as follows:

- Method 1:

  Run the following command. In the command output, **/usr/lib/python2.7/site-packages/cinder/volume/drivers/huawei** is the code directory.
  ```
  # python -c "from cinder.volume.drivers import huawei; print (huawei.__path__)"
  ['/usr/lib/python2.7/site-packages/cinder/volume/drivers/huawei']
  ```

- Method 2:

  Run the following command. In the command output, **/usr/lib/python3.7/site-packages/cinder/volume/drivers/huawei** is the code directory.
  ```
  # python3 -c "from cinder.volume.drivers import huawei; print (huawei.__path__)"
  ['/usr/lib/python3.7/site-packages/cinder/volume/drivers/huawei']
  ```

**Step 2** Back up all Huawei Cinder Driver code files in the directory returned in **Step 1**.

The backup command is as follows:

```
cp -rf Source directory  Backup directory
```

📖 NOTE

- *Source directory*: directory returned in **Step 1**.
- *Backup directory*: directory to which data is backed up.

**Step 3** Copy all Huawei Cinder Driver code files of the corresponding version to the directory returned in **Step 1**. The minimum permission required for the Cinder Driver code files is **644**. (**644** is a Linux system permission. **-rw-r--r--** indicates the minimum permission required.)

```
# 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 hypermetro.py
-rw-r--r-- 1 root root      0 Mar  4 23:40 __init__.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
```

**Step 4** Run the following command to restart the Cinder Volume service.

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

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+-----------------+-------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| Binary          | Host              | Zone | Status  | State | Updated_at                 | Cluster | Disabled Reason | Backend State |
+-----------------+-------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| cinder-volume   | openstack-wallaby-1@v6 | nova | enabled | up    | 2021-12-03T07:56:23.000000 | -       | -               | up            |
```

**----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 a node that has a Cinder Volume container image and run the **docker image ls** command to query the Cinder Volume container image.



**Step 2** Run the **docker save -o cinder_volume.tar kolla/centos-source-cinder-volume** command to back up the Cinder Volume container image to an image file.

📖 **NOTE**

**cinder_volume.tar** is a user-defined name of the backup image, and **kolla/centos-source-cinder-volume** is the name of the Cinder Volume container image.

**Step 3** Create a temporary directory named **huawei** in any directory.

```
# mkdir huawei
```

**Step 4** Copy the obtained Huawei Cinder Driver code files of the corresponding version to the **huawei** directory. The minimum permission required for the Cinder Driver code files is **644**.

```
# 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 hypermetro.py
-rw-r--r-- 1 root root      0 Mar  4 23:40 __init__.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
```

**Step 5** Run the **docker ps | grep cinder** command to list Cinder Volume containers. In the command output, **cinder_volume** indicates the container name.



**Step 6** Run the **docker exec -it -u 0 cinder_volume bash** command to access the **cinder_volume** container.



**Step 7** Run the **python -c "from cinder.volume import drivers; print (drivers.__path__)"** command to query the path of the original Huawei Cinder Driver code.



📖 **NOTE**

- **cinder_volume** is the container name obtained in **Step 5**.
- **/var/lib/kolla/venv/lib/python3.6/site-packages/cinder/volume/drivers** is the path of the original Huawei Cinder Driver code.

**Step 8** Exit the **cinder_volume** container, go to the upper-level directory of the **huawei** directory created in **Step 3**, and copy the **huawei** directory to the path of the original Huawei Cinder Driver code in **Step 7**.



```
docker cp huawei/ cinder_volume:/var/lib/kolla/venv/lib/python3.6/site-packages/cinder/volume/drivers
```

**----End**

NOTE

If the upgrade fails, restore the container image backed up in **Step 2**.

# 6 Configuring Basic Properties of Cinder Driver

This chapter describes how to configure Huawei Cinder Driver to connect to Huawei storage on an iSCSI or FC network.

> **NOTICE**
>
> - Ensure that the storage pool used for configuring Cinder Driver exists on Huawei storage. Otherwise, you need to manually create one. In addition, the type of this storage pool must be the 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 Huawei Cinder Driver configuration file cannot contain the following XML special characters: <>&'"
> - When FastClone is used for LUN cloning, the mutually exclusive relationship between the clone feature and other features or functions varies depending on the product model. For details about the specifications, see the *Product Description* of the corresponding product model. You can log in to Huawei technical support website (**https://support.huawei.com/enterprise/**) and enter the product model + document name in the search box to search for, browse, and download the desired documents.

## 6.1 Configuring Cinder Driver in Non-Containerized Mode

### 6.1.1 Configuring Cinder Driver on an iSCSI Network

This section describes how to configure Cinder Driver for Huawei storage products when the iSCSI protocol is used.

## Procedure

**Step 1** Use a remote access tool, such as PuTTY, to log in to an OpenStack Cinder Volume node through the management IP address.

**Step 2** Run the following commands to go to the **/etc/cinder** directory and create a Huawei Cinder Driver configuration file in xml format. You can customize the configuration file name, for example, **cinder_huawei_conf.xml**.

```
cd /etc/cinder
vi cinder_huawei_conf.xml
```

**Step 3** Configure mandatory parameters in the Huawei Cinder Driver configuration file. Use the Linux text editor to open the configuration file created in **Step 2**. Take the default vi editor of Linux as an example. Run the **vi cinder_huawei_conf.xml** command to open and edit the configuration file. The following lists the parameters in the configuration file. For details about the parameter values and more information, see **Table 6-1** and **Table 6-2**.

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
  <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="*"
TargetIP="x.x.x.x y.y.y.y"/>
  </iSCSI>
</config>
```

Alternatively, you can set **Name** in **<iSCSI>** for OceanStor V3, OceanStor V5, and OceanStor Dorado V3.

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

The ALUA policies of OceanStor Dorado and OceanStor V6 are different. 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 | Storage product type. The value can be **V3**, **V5**, **V6**, or **Dorado**. |
| Protocol | Connection protocol type. Set this parameter to **iSCSI** in iSCSI scenarios. |
| RestURL | Access address of the REST interface. If you need to configure multiple addresses, separate them by using semicolons (;). |
| UserName | User name of a storage administrator. The user name must be an administrator or super administrator. |
| UserPassword | Password of the storage administrator. |
| StoragePool | Name of a storage pool to be used. The storage pool must be created on the target storage in advance. If you need to configure multiple storage pools, separate them by using semicolons (;). |
| LUNType | Type of a LUN to be created. The value can be **Thick** or **Thin**. OceanStor Dorado supports only **Thin**. |
| LUNCopySpeed | LUN copy speed. The value can be:<br>● **1**: low<br>● **2**: medium<br>● **3**: high<br>● **4**: highest |
| Initiator | For parameter details, see **Table 6-2**.<br>**NOTE**<br>OceanStor V3, OceanStor V5, OceanStor V6, OceanStor Dorado V3, and OceanStor Dorado storage systems support **Name** or **HostName** (they cannot be configured at the same time). **HostName** is recommended. |

**Table 6-2** Initiator configuration description

| Parameter | Description |
|---|---|
| HostName | The value of **HostName** is the regular expression of the host name of the node where Nova-compute or Cinder-volume resides. Specify either **HostName** or **Name**.<br><br>In Linux, you can run the **cat /etc/hostname** or **hostname** command to obtain the host name.<br><br>● If the host names obtained using the two methods are the same, configure the host name by referring to **Regular expression**.<br>● If the host names obtained using the two methods are different, configure the host name in any of the following ways:<br>  – Change the host names obtained using the two methods to be the same, and then configure the host name by referring to **Regular expression**.<br>  – Configure the host names obtained using the two modes separately by referring to **Regular expression**.<br>  – Use the common configuration **HostName="*"**.<br>  – Specify **Name** rather than **HostName**. |
| Name | The value of **Name** is the host iSCSI initiator name of the node where Nova-compute or Cinder-volume resides. Specify either **HostName** or **Name**.<br><br>You can run the **cat /etc/iscsi/initiatorname.iscsi** command to obtain the value. |
| ALUA parameters | ALUA parameters that can be configured include **ALUA**, **FAILOVERMODE**, **SPECIALMODETYPE**, **PATHTYPE**, **ACCESSMODE**, and **HYPERMETROPATHOPTIMIZED**. For details, see **11.1 ALUA Parameter Description**. |

| Parameter | Description |
|-----------|-------------|
| TargetIP | **TargetIP** indicates the IP address of the iSCSI target port. You can enter multiple IP addresses and separate them with spaces. Different **TargetIP** values can be configured for different hosts.<br><br>You can log in to DeviceManager to obtain the IP address of the iSCSI target port.<br><br>● OceanStor V3/V5 series: To obtain the IP address of an Ethernet port on a specified storage device, choose **Provisioning** > **Port** > **Ethernet Ports** on DeviceManager.<br><br>● OceanStor Dorado V3 series: To obtain the IP address of an Ethernet port on a specified storage device, choose **Provisioning** > **Port** > **Ethernet Ports** on DeviceManager.<br><br>● OceanStor Dorado series: To obtain the IP address of a logical port on a specified storage device, choose **Services** > **Network** > **Logical Ports** on DeviceManager and obtain the IP address whose data protocol is iSCSI.<br><br>● OceanStor V6 series: To obtain the IP address of a logical port on a specified storage device, choose **Services** > **Network** > **Logical Ports** on DeviceManager and obtain the IP address whose data protocol is iSCSI. |

> **NOTICE**

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

  For OceanStor V3 series, enter **V3**. For OceanStor V5 series, enter **V5**. For OceanStor V6 series, enter **V6**. For OceanStor Dorado series, enter **Dorado**.

- The value of **HostName** in the **Initiator** configuration item is a regular expression. For details about how to configure it, see **Regular expression**.

  - When **HostName** is set to **\***, the common configuration is used and takes effect on hosts 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 about the limitation, see **Specifications Query** and search for **Maximum number of iSCSI connections per controller enclosure**. If the number of host connections is less than or equal to the upper limit, you are advised to use the common configuration. If the number of host connections is greater than the upper limit, use the general configuration.

- Configuration policy rules:

  - Priority: General host name configuration > Common host name configuration. For details, see example 1 in **9.2 Initiator Configuration Policy Examples**.

  - In the general configuration, use the first ALUA section that meets the configuration policy. For details, see example 2 in **9.2 Initiator Configuration Policy Examples**.

  - In the general configuration, if you need to exactly match a host, refer to example 5 in **9.2 Initiator Configuration Policy Examples**.

**Step 4**   (Optional) Configure optional parameters in the Huawei 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 6-3** Optional parameters

| Parameter | Default Value | Description |
|-----------|---------------|-------------|
| WriteType | 1 | Cache write mode. The value can be:<br>● **1**: write back<br>● **2**: write through<br>Volumes created on OceanStor Dorado series storage support write back only. |

| Parameter | Default Value | Description |
|---|---|---|
| Prefetch Type | 3 | Cache prefetch policy. The value can be:<br>● **0**: no prefetch<br>● **1**: constant prefetch<br>● **2**: variable prefetch<br>● **3**: intelligent prefetch |
| Prefetch Value | 0 | Cache prefetch value. |
| LUNcopyWaitInterval | 5 | After starting the LUN copy, the plug-in needs to continuously query the LUN copy progress. This parameter is used to specify the query interval, expressed in seconds. |
| Timeout | 2592000 | Timeout period of waiting for a storage device to perform the LUN copy, expressed in seconds. |
| LUNCloneMode | luncopy | LUN clone mode. The value can be:<br>● fastclone<br>● luncopy<br>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 | Synchronization speed of a HyperMetro pair. The value can be:<br>● **1**: low<br>● **2**: medium<br>● **3**: high<br>● **4**: highest |
| ReplicaSyncSpeed | 2 | Synchronization speed of a remote replication pair. The value can be:<br>● **1**: low<br>● **2**: medium<br>● **3**: high<br>● **4**: highest |
| ForceDeleteVolume | False | Whether to forcibly delete a LUN that exists in a mapping view. The value can be:<br>● **True**: forcibly<br>● **False**: not forcibly |

| Parameter | Default Value | Description |
|---|---|---|
| HyperEnforceMulti-path | True | Whether to enable multipathing in HyperMetro scenarios. |
| SnapshotRollback-Speed | 3 | Snapshot rollback speed. The value can be:<br><br>● **1**: low<br><br>● **2**: medium<br><br>● **3**: high<br><br>● **4**: highest |

**Step 5** 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. If they are different, run Linux command **chown** to change the owner and user group of the **cinder_huawei_conf.xml** file to be the same as those of the **/etc/cinder/cinder.conf** file. After the change, the result is similar to the following:

```
-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 6** Add the following information to the end of the **/etc/cinder/cinder.conf** 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
```

📖 **NOTE**

- **huawei_backend** indicates the backend name. It can be customized.
- **volume_driver** indicates the type of Cinder Driver to be used. Set this parameter to **cinder.volume.drivers.huawei.huawei_driver.HuaweiISCSIDriver**.
- **cinder_huawei_conf_file** indicates the path of the Huawei Cinder Driver configuration file.
- **retain_storage_mapping** indicates whether to retain the storage mapping relationship 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]** section, set **enabled_backends** to **huawei_backend**.

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

**Step 7** If Cinder Volume is deployed on multiple nodes, you need to configure a distributed lock in the **[coordination]** section. 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, if the configuration of **[database]** is as follows:

```
[database]
connection = mysql+pymysql://cinder:302824058e9a4f31@127.0.0.1/cinder
```

You can configure **[database]** as follows:

```
[coordination]
backend_url = mysql://cinder:302824058e9a4f31@127.0.0.1/cinder
```

**Step 8**  Run the following command to restart the Cinder Volume service.

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

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+---------------+---------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| Binary        | Host                | Zone | Status  | State | Updated_at                 | Cluster | Disabled Reason | Backend State |
+---------------+---------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| cinder-volume | openstack-wallaby-1@v6 | nova | enabled | up    | 2021-12-03T07:56:23.000000 | -       | -               | up            |
```

**----End**

# 6.1.2 Configuring Cinder Driver on an FC Network

This section describes how to configure Cinder Driver for Huawei storage products when the FC protocol is used.

## Procedure

**Step 1**  Use a remote access tool, such as PuTTY, to log in to an OpenStack Cinder Volume node through the management IP address.

**Step 2**  Run the following commands to go to the **/etc/cinder** directory and create a Huawei Cinder Driver configuration file in xml format. You can customize the configuration file name, for example, **cinder_huawei_conf.xml**.

```
cd /etc/cinder
vi cinder_huawei_conf.xml
```

**Step 3**  Configure mandatory parameters in the Huawei Cinder Driver configuration file. Use the Linux text editor to open the configuration file created in **Step 2**. Take the default vi editor of Linux as an example. Run the **vi cinder_huawei_conf.xml** command to open and edit the configuration file. The following lists the parameters in the configuration file. For details about the parameter values and more information, see **Table 6-4** and **Table 6-5**.

```xml
<?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>
```

Alternatively, you can set **Name** in **<FC>** for OceanStor V3, OceanStor V5, and OceanStor Dorado V3.

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

The ALUA policies of OceanStor Dorado and OceanStor V6 are different. See the following.

```
…
 <FC>
   <MinOnlineFCInitiator>xx</MinOnlineFCInitiator>
   <Initiator HostName="xxx" ACCESSMODE="xxx" HYPERMETROPATHOPTIMIZED="xxx"/>
   <Initiator HostName="xxx" ACCESSMODE="xxx" HYPERMETROPATHOPTIMIZED="xxx"
 </FC>
…
```

**Table 6-4** Mandatory parameters

| Parameter | Description |
|---|---|
| Product | Storage product type. The value can be **V3**, **V5**, **V6**, or **Dorado**. |
| Protocol | Connection protocol type. Set this parameter to **FC** in FC scenarios. |
| RestURL | Access address of the REST interface. If you need to configure multiple addresses, separate them by using semicolons (;). |
| UserName | User name of a storage administrator. The user name must be an administrator or super administrator. |
| UserPassword | Password of the storage administrator. |
| StoragePool | Name of a storage pool to be used. The storage pool must be created on the target storage in advance. If you need to configure multiple storage pools, separate them by using semicolons (;). |
| LUNType | Type of a LUN to be created. The value can be **Thick** or **Thin**. OceanStor Dorado supports only **Thin**. |
| LUNCopySpeed | LUN copy speed. The value can be:<br>● **1**: low<br>● **2**: medium<br>● **3**: high<br>● **4**: highest |
| MinOnlineFCInitiator | Minimum number of available FC initiators. The value is an Arabic numeral. The default value indicates that all initiators must be online to complete volume mounting. |

| Parameter | Description |
|---|---|
| Initiator | For parameter details, see **Table 6-5**.<br>**NOTE**<br>    OceanStor V3, OceanStor V5, OceanStor V6, OceanStor Dorado V3, and OceanStor Dorado storage systems support **Name** or **HostName** (they cannot be configured at the same time). **HostName** is recommended. |

**Table 6-5** Initiator configuration description

| Parameter | Description |
|---|---|
| HostName | The value of **HostName** is the regular expression of the host name of the node where Nova-compute or Cinder-volume resides. Specify either **HostName** or **Name**.<br>In Linux, you can run the **cat /etc/hostname** or **hostname** command to obtain the host name.<br>● If the host names obtained using the two methods are the same, configure the host name by referring to **Regular expression**.<br>● If the host names obtained using the two methods are different, configure the host name in any of the following ways:<br>  – Change the host names obtained using the two methods to be the same, and then configure the host name by referring to **Regular expression**.<br>  – Configure the host names obtained using the two modes separately by referring to **Regular expression**.<br>  – Use the common configuration **HostName="*"**.<br>  – Specify **Name** rather than **HostName**. |
| Name | The value of **Name** is the host FC initiator WWN of the node where Nova-compute or Cinder-volume resides. Specify either **HostName** or **Name**.<br>You can run the **cat /sys/class/fc_host/*/port_name** command to obtain the value. |
| ALUA parameters | ALUA parameters that can be configured include **ALUA**, **FAILOVERMODE**, **SPECIALMODETYPE**, **PATHTYPE**, **ACCESSMODE**, and **HYPERMETROPATHOPTIMIZED**. For details, see **11.1 ALUA Parameter Description**. |

> **NOTICE**
>
> ● Set **Product** based on the actual back-end storage model.
>
>   For OceanStor V3 series, enter **V3**. For OceanStor V5 series, enter **V5**. For OceanStor V6 series, enter **V6**. For OceanStor Dorado series, enter **Dorado**.
>
> ● The value of **HostName** in the **Initiator** configuration item is a regular expression. For details about how to configure it, see **Regular expression**.
>
>   – When **HostName** is set to **\***, the common configuration is used and takes effect on hosts 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 about the limitation, see **Specifications Query** and search for **Maximum number of Fibre Channel connections per controller enclosure**. If the number of host connections is less than or equal to the upper limit, you are advised to use the common configuration. If the number of host connections is greater than the upper limit, use the general configuration.
>
> ● Configuration policy rules:
>
>   – Priority: General host name configuration > Common host name configuration. For details, see example 3 in **9.2 Initiator Configuration Policy Examples**.
>
>   – In the general configuration, use the first ALUA section that meets the configuration policy. For details, see example 4 in **9.2 Initiator Configuration Policy Examples**.
>
>   – In the general configuration, if you need to exactly match a host, refer to example 6 in **9.2 Initiator Configuration Policy Examples**.

**Step 4** (Optional) Configure optional parameters in the Huawei 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 6-6** Optional parameters

| Parameter | Default Value | Description |
|-----------|---------------|-------------|
| WriteType | 1 | Cache write mode. The value can be: <br> • **1**: write back <br> • **2**: write through <br> Volumes created on OceanStor Dorado series storage support write back only. |
| Prefetch Type | 3 | Cache prefetch policy. The value can be: <br> • **0**: no prefetch <br> • **1**: constant prefetch <br> • **2**: variable prefetch <br> • **3**: intelligent prefetch |
| Prefetch Value | 0 | Cache prefetch value. |
| LUNcopyWaitInterval | 5 | After starting the LUN copy, the plug-in needs to continuously query the LUN copy progress. This parameter is used to specify the query interval, expressed in seconds. |
| Timeout | 2592000 | Timeout period of waiting for a storage device to perform the LUN copy, expressed in seconds. |
| LUNCloneMode | luncopy | LUN clone mode. The value can be: <br> • fastclone <br> • luncopy <br> 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 | Synchronization speed of a HyperMetro pair. The value can be: <br> • **1**: low <br> • **2**: medium <br> • **3**: high <br> • **4**: highest |

| Parameter | Default Value | Description |
|---|---|---|
| ReplicaSyncSpeed | 2 | Synchronization speed of a remote replication pair. The value can be:<br>● **1**: low<br>● **2**: medium<br>● **3**: high<br>● **4**: highest |
| ForceDeleteVolume | False | Whether to forcibly delete a LUN that exists in a mapping view. The value can be:<br>● **True**: forcibly<br>● **False**: not forcibly |
| HyperEnforceMulti-path | True | Whether to enable multipathing in HyperMetro scenarios. |
| SnapshotRollback-Speed | 3 | Snapshot rollback speed. The value can be:<br>● **1**: low<br>● **2**: medium<br>● **3**: high<br>● **4**: highest |

**Step 5** 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. If they are different, run Linux command **chown** to change the owner and user group of the **cinder_huawei_conf.xml** file to be the same as those of the **/etc/cinder/cinder.conf** file. After the change, the result is similar to the following:

```
-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 6** Add the following information to the end of the **/etc/cinder/cinder.conf** 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**

- **huawei_backend** indicates the backend name. It can be customized.
- **volume_driver** indicates the type of Cinder Driver to be used. Set this parameter to **cinder.volume.drivers.huawei.huawei_driver.HuaweiFCDriver**.
- **cinder_huawei_conf_file** indicates the path of the Huawei Cinder Driver configuration file.
- **retain_storage_mapping** indicates whether to retain the storage mapping relationship 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]** section, set **enabled_backends** to **huawei_backend**.

```
[DEFAULT]
…
enabled_backends=huawei_backend
```

**Step 7** If Cinder Volume is deployed on multiple nodes, you need to configure a distributed lock in the **[coordination]** section. 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
```

**Step 8** Run the following command to restart the Cinder Volume service.

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

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+--------------+----------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| Binary       | Host                 | Zone | Status  | State | Updated_at                 | Cluster | Disabled Reason | Backend State |
+--------------+----------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| cinder-volume| openstack-wallaby-1@v6| nova | enabled | up   | 2021-12-03T07:56:23.000000 | -       | -               | up            |
```

**----End**

# 6.2 Configuring Cinder Driver in Containerized Mode

This section describes how to configure Cinder Driver for Huawei storage products in containerized mode. The following uses OpenStack Kolla as an example.

## 6.2.1 Configuring Cinder Driver on an iSCSI Network

This section describes how to configure Cinder Driver for Huawei storage products when the iSCSI protocol is used.

### Procedure

**Step 1** Use a remote access tool, such as PuTTY, to log in to an OpenStack Cinder Volume node through the management IP address. Run the following command on a host to find the **cinder.conf** file and configure parameters by referring to **6.1.1 Configuring Cinder Driver on an iSCSI Network**.

```
find / -name cinder.conf | grep volume
```

**Step 2** Run the **docker ps | grep cinder** command to query the Cinder Volume container. Then run the **docker exec -it -u 0 cinder_volume bash** command to access the Cinder Volume container.

```
[root@control ~]# docker ps | grep cinder
681a72350b38    kolla/centos-source-cinder-backup:train-centos8       "dumb-init --single-_ "   15 months ago   Up 25 hours                     cinder_backup
2a1e28d9a4a7    kolla/centos-source-cinder-volume:train-centos8       "dumb-init --single-_ "   15 months ago   Up 25 hours                     cinder_volume
d2da14bd6c17    kolla/centos-source-cinder-scheduler:train-centos8    "dumb-init --single-_ "   15 months ago   Up 25 hours                     cinder_scheduler
0d0d27771db5    kolla/centos-source-cinder-api:train-centos8          "dumb-init --single-_ "   15 months ago   Up 25 hours                     cinder_api
```

```
[root@control ~]# docker exec -it -u 0 cinder_volume bash
(cinder-volume)[root@control /]#
```

**NOTE**

**cinder_volume** is the name of the Cinder Volume container.

**Step 3** Create a Huawei Cinder Driver configuration file in xml format in the **/etc/cinder** directory of the Cinder Volume container and configure parameters by referring to

**6.1.1 Configuring Cinder Driver on an iSCSI Network**. 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**  Run the following command to restart the Cinder Volume service.

```
docker restart cinder_volume
```

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+-----------------+-------------------------+------+---------+-------+-------------------------+---------+-----------------+---------------+
| Binary          | Host                    | Zone | Status  | State | Updated_at              | Cluster | Disabled Reason | Backend State |
+-----------------+-------------------------+------+---------+-------+-------------------------+---------+-----------------+---------------+
| cinder-volume   | openstack-wallaby-1@v6  | nova | enabled | up    | 2021-12-03T07:56:23.000000 | -    | -               | up            |
```

**----End**

# 6.2.2 Configuring Cinder Driver on an FC Network

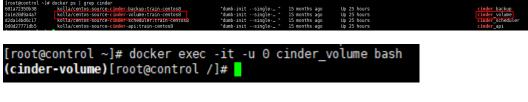This section describes how to configure Cinder Driver for Huawei storage products when the FC protocol is used.

## Procedure

**Step 1**  Use a remote access tool, such as PuTTY, to log in to an OpenStack Cinder Volume node through the management IP address. Run the following command on a host to find the **cinder.conf** file and configure parameters by referring to **6.1.2 Configuring Cinder Driver on an FC Network**.

```
find / -name cinder.conf | grep volume
```

**Step 2**  Run the **docker ps | grep cinder** command to query the Cinder Volume container. Then run the **docker exec -it -u 0 cinder_volume bash** command to access the Cinder Volume container.



📖 **NOTE**

**cinder_volume** is the name of the Cinder Volume container.

**Step 3**  Create a Huawei Cinder Driver configuration file in xml format in the **/etc/cinder** directory of the Cinder Volume container and configure parameters by referring to **6.1.2 Configuring Cinder Driver on an FC Network**. 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** Run the following command to restart the Cinder Volume service.

docker restart cinder_volume

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+----------------+-------------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| Binary         | Host                    | Zone | Status  | State | Updated_at                 | Cluster | Disabled Reason | Backend State |
+----------------+-------------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| cinder-volume  | openstack-wallaby-1@v6  | nova | enabled | up    | 2021-12-03T07:56:23.000000 | -       | -               | up            |
```

**----End**

# 7 Configuring Multipathing

Multipathing is configured to improve the link reliability of LUNs on SAN storage. If multipathing is incorrectly configured, I/O errors will occur when a single link is faulty. As a result, VM file systems are 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

This section describes how to install the multipathing tool package.

**Prerequisites**

OpenStack nodes can access the Internet (only used to download the multipathing tool package).

**Precautions**

The host multipathing service needs to be configured only when storage devices are connected over an iSCSI or FC network.
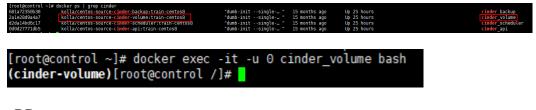
**Procedure**

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

**Step 2** Install the multipathing tool package based on the OS.

- CentOS:
  ```
  yum install -y device-mapper-multipath
  ```
- Ubuntu:

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

- SUSE:
```
zypper install -y multipath-tools
```

**Step 3** 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
```

- SUSE:
```
systemctl restart multipath-tools.service
chkconfig multipathd on
```

**Step 4** Repeat **Step 1** to **Step 3** to install the multipathing tool on all Nova nodes.

**----End**

## 7.1.2 Configuring the Multipathing Service

Multipathing is configured to improve the link reliability of LUNs on SAN storage. If multipathing is incorrectly configured, I/O errors will occur when a single link is faulty. As a result, the file systems or disks managed by OpenStack are read-only or faulty, affecting I/O delivery.

### Precautions

The multipathing service needs to be configured only when storage devices are connected over an iSCSI or FC network.

### Procedure

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

**Step 2** Run the **vi /etc/multipath.conf** command to modify the **multipath.conf** file. Press **I** or **Insert** to enter the editing mode and modify related parameters. After the modification is complete, press **Esc** and enter **:wq!** to save the modification.

> **NOTICE**
>
> - Load balancing mode: During service read and write, the I/O paths from a host to all controllers on a storage device are the same. For details, see **Configuring Multipathing** > **Concepts** in *Huawei SAN Storage Host Connectivity Guide for Red Hat*.
>
> - Local preferred mode: When a host delivers I/Os to controllers, the storage device with better performance is accessed because the service link distances from storage devices are different. For details, see **Configuring Multipathing** > **Concepts** in *Huawei SAN Storage Host Connectivity Guide for Red Hat*.

- If the load balancing mode is used, you are advised to add the following content to the **devices** field in the multipathing configuration file (**/etc/multipath.conf**). The following is a typical example. For details about all scenarios and examples, see *OceanStor Dorado Host Connectivity Guide for Red Hat*.

```
devices {
    device {
            vendor                "HUAWEI"
            product               "XSG1"
            path_grouping_policy      multibus
            path_checker          tur
            prio                  const
            path_selector            "service-time 0"
            failback              immediate
            no_path_retry            15
    }
}
```

- If the local preferred mode is used, you are advised to add the following content to the **devices** field in the multipathing configuration file (**/etc/multipath.conf**). The following is a typical example. For details about all scenarios and examples, 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            15
    }
}
```

**Step 3** After the configuration is complete, run the following command to restart the multipathd service.

```
systemctl restart multipathd.service
```

**Step 4** Repeat **Step 1** to **Step 3** to configure the multipathing service for all Nova nodes.

**----End**

📖 **NOTE**

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

# 7.2 Configuring OpenStack Multipathing

## 7.2.1 Configuring Multipathing for OpenStack Nova Nodes

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

**Step 2** Configure the **/etc/nova/nova.conf** file.

Add **volume_use_multipath = True** to **[libvirt]** in the **/etc/nova/nova.conf** file of the Nova Compute node.

📖 **NOTE**

- If the OpenStack version is Mitaka or earlier, add **iscsi_use_multipath = True** to **[libvirt]** in the **/etc/nova/nova.conf** file of the Nova Compute node.
- If the OpenStack version is Newton or earlier, add **volume_use_multipath = True** to **[libvirt]** in the **/etc/nova/nova.conf** file of the Nova Compute node.

**Step 3** Restart the Nova-compute service.

systemctl restart openstack-nova-compute.service

**----End**

## 7.2.2 Configuring Multipathing for OpenStack Cinder Nodes

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

**Step 2** 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 3** Restart the Cinder-volume service.

systemctl restart openstack-cinder-volume.service

**----End**

# 8 Configuring Advanced Features of Cinder Driver

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

**NOTICE**

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

# 8.1 Configuring the Thick Property

This section describes how to configure the Thick property.

> **NOTICE**
>
> OceanStor Dorado does not support Thick volumes.

## Procedure

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

**Step 2** Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3** Set the Thick volume type. *<vtype>* indicates the name of the volume type created in **Step 2**.

```
# cinder type-key <vtype> set capabilities:thick_provisioning_support='<is> true'
```

**Step 4** Set the scheduling mechanism for the Thick volume. *<vtype>* indicates the name of the volume type created in **Step 2**.

```
# cinder type-key <vtype> set provisioning:type='thick'
```

**Step 5** Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 6** Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.2 Configuring the Thin Property

This section describes how to configure the Thin property.

**Procedure**
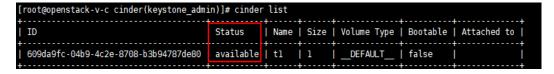
**Step 1**  Use a remote access tool, such as PuTTY, to log in to an OpenStack Cinder node through the management IP address.

**Step 2**  Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3**  Set the Thin volume type. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

**Step 4**  Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 5**  Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.



**----End**

# 8.3 Configuring the SmartQoS Property

The QoS feature of OpenStack depends on the front-end hypervisor and back-end storage.

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

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

**Procedure**

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

**Step 2**  Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3**  Enable the SmartQoS property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

**Step 4** Create a QoS specification. *<name>* indicates the name of the QoS specification.

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

**Table 8-1** IOType parameters

| Parameter | Description | Remarks |
|---|---|---|
| IOType | Read/write type. | The value can be:<br>● **0**: read I/O<br>● **1**: write I/O<br>● **2**: read and write I/Os |
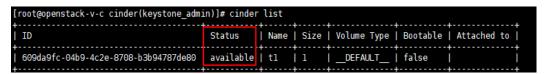
**Table 8-2** Optional qos_key parameters

| Parameter | Description | Remarks |
|---|---|---|
| maxIOPS | Maximum IOPS. This is a restriction policy parameter. | The value is an integer greater than 0. For OceanStor Dorado series, the value is an integer greater than or equal to 100. |
| minIOPS | Minimum IOPS. This is a protection policy parameter. | The value is an integer greater than 0. For OceanStor Dorado series, the value is an integer greater than or equal to 100. |
| maxBandWidth | Maximum bandwidth. This is a restriction policy parameter. | The value is an integer greater than 0, expressed in MB/s. |
| minBandWidth | Minimum bandwidth. This is a protection policy parameter. | The value is an integer greater than 0, expressed in MB/s. |
| latency | Maximum latency. This is a protection policy parameter. | The value is an integer greater than 0, expressed in ms. For OceanStor Dorado series, the value is 500 or 1500. |
| burstIOPS | Maximum burst IOPS. | Only OceanStor Dorado and OceanStor V6 series are supported. The value is an integer greater than or equal to 100. When setting **burstIOPS**, you also need to set **maxIOPS** and **burstTime**. Moreover, the value of this parameter must be greater than that of **maxIOPS**. |

| Parameter | Description | Remarks |
|-----------|-------------|---------|
| burstBandWidth | Maximum burst bandwidth. | Only OceanStor Dorado and OceanStor V6 series are supported. The unit is MB/s. The value is an integer greater than 0. When setting **burstBandWidth**, you also need to set **maxBandWidth** and **burstTime**. Moreover, the value of this parameter must be greater than that of **maxBandWidth**. |
| burstTime | Burst duration. | Only OceanStor Dorado and OceanStor V6 series are supported. The unit is s. The value is an integer greater than 0. When setting **burstIOPS** or **burstBandWidth**, you must also set **burstTime**. |

☐ **NOTE**

- Different protection policy parameters or restriction policy parameters can be specified at the same time. However, protection policy parameters cannot coexist with restriction policy parameters.

- For OceanStor Dorado and OceanStor V6 storage, **IOType** must be set to **2**, the protection policy and restriction policy parameters can be specified at the same time, but **latency** cannot be configured.

- **latency**, **burstIOPS**, **burstBandWidth**, **burstTime**, and protection policy parameters support only OpenStack R to Y versions.

- For OceanStor Dorado and OceanStor V6 storage, versions earlier than R do not support **minIOPS**, **minBandWidth**, and **latency**.

**Step 5**  Associate the QoS specification with the volume type. *<qos_specs>* indicates the ID of the QoS specifications created in **Step 4**, and *<volume_type_id>* indicates the ID of the volume type created in **Step 2**.

# cinder qos-associate <qos_specs> <volume_type_id>

**Step 6**  Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

# cinder create 10 --name <volume-name> --volume-type <name>

**Step 7**  Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.4 Configuring the SmartPartition Property

This section describes how to configure the SmartPartition property.

## Procedure

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

**Step 2**  Create a volume type. *<name>* indicates the name of the volume type.

# cinder type-create <name>

**Step 3**  Enable the SmartPartition property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

**Step 4**  Configure the name of existing SmartPartition on the back-end storage.

# cinder type-key <vtype> set smartpartition:partitionname='***'

**Step 5**  Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

# cinder create 10 --name <volume-name> --volume-type <name>

**Step 6**  Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.5 Configuring the SmartCache Property

This section describes how to configure the SmartCache property.

## Procedure
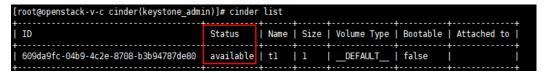
**Step 1**  Use a remote access tool, such as PuTTY, to log in to an OpenStack Cinder node through the management IP address.

**Step 2**  Create a volume type. *<name>* indicates the name of the volume type.

# cinder type-create <name>

**Step 3**  Enable the SmartCache property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

**Step 4**  Configure the name of existing SmartCache on the back-end storage.

# cinder type-key <vtype> set smartcache:cachename='***'

**Step 5** Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 6** Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.6 Configuring the SmartTier Property

This section describes how to configure the SmartTier property.

## Procedure

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

**Step 2** Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3** Enable the SmartTier property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

**Step 4** Configure a SmartTier migration policy.

```
# cinder type-key <vtype> set smarttier:policy=***
```
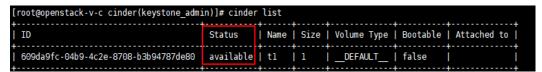
> 📖 **NOTE**
>
> **policy** indicates a migration policy. The value can be:
> - **0**: no migration
> - **1**: automatic migration
> - **2**: migration to a high-performance tier
> - **3**: migration to a low-performance tier

**Step 5** Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 6** Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.7 Configuring the HyperMetro Property

This section describes how to configure the HyperMetro property.

## Procedure

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

**Step 2** Configure the remote device information in the specified backend in **/etc/cinder/cinder.conf**.

```
hypermetro_device =
    storage_pool:***,
    san_address:https://*.*.*.*:8088/deviceManager/rest/,
    san_user:***,
    san_password:***,
    vstore_name:***,
    metro_domain:***,
    metro_sync_completed:***,
    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};
            {HostName:xxx;ALUA:xxx;FAILOVERMODE:xxx;PATHTYPE:xxx}
```

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

```
…
    iscsi_info: {HostName:xxx;ACCESSMODE:xxx;HYPERMETROPATHOPTIMIZED:xxx;TargetIP:x.x.x.x y.y.y.y};
            {HostName:xxx;ACCESSMODE:xxx;HYPERMETROPATHOPTIMIZED:xxx;TargetIP:x.x.x.x z.z.z.z},
    fc_info: {HostName:xxx;ACCESSMODE:xxx;HYPERMETROPATHOPTIMIZED:xxx};
            {HostName:xxx;ACCESSMODE:xxx;HYPERMETROPATHOPTIMIZED:xxx}
```

**Table 8-3** Parameter description

| Parameter | Description | Mandatory |
|---|---|---|
| storage_pool | Name of the remote storage pool. | Yes |
| san_address | Access address of the REST interface. | Yes |
| san_user | User name of a storage administrator or vStore of the remote device. | Yes |
| san_password | Password of a storage administrator or vStore of the remote device. | Yes |
| vstore_name | vStore name of the remote device. This parameter needs to be configured only when a vStore user is used. | No |
| metro_domain | HyperMetro domain name. | Yes |
| metro_sync_completed | Whether to wait for the synchronization to complete during HyperMetro creation. The value can be **True** or **False**. The default value is **True**. | No |

| Parameter | Description | Mandatory |
|---|---|---|
| iscsi_info | **HostName** is the regular expression of the host name of the Nova or Cinder node.<br><br>Configuration rule and priority description:<br><br>● **ALUA**, **FAILOVERMODE**, **PATHTYPE**, **ACCESSMODE**, and **HYPERMETROPATHOPTIMIZED** indicate ALUA parameters of the storage device. For details, see **11.1 ALUA Parameter Description**.<br><br>● **TargetIP** indicates the IP address of the iSCSI target port. You can enter multiple IP addresses and separate them with spaces. Different **TargetIP** values can be configured for different hosts. | Mandatory for iSCSI networking |
| fc_info | **HostName** is the regular expression of the host name of the Nova or Cinder node.<br><br>Configuration rule and priority description:<br><br>● **ALUA**, **FAILOVERMODE**, **PATHTYPE**, **ACCESSMODE**, and **HYPERMETROPATHOPTIMIZED** indicate ALUA parameters of the storage device. For details, see **11.1 ALUA Parameter Description**.<br><br>● This parameter can be set for different hosts. | Mandatory for FC networking |

**Step 3**  Restart the Cinder Volume service.

**Step 4**  Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 5**  Enable the HyperMetro property. *<vtype>* indicates the name of the volume type created in **Step 4**.

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

**Step 6**  Use the volume type created in **Step 4** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 7**  Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.8 Configuring the Remote Replication Property

This section describes how to configure the remote replication property.

**Procedure**

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

**Step 2** Configure the remote device information in the specified backend in **/etc/cinder/cinder.conf**.

```
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};
            {HostName:xxx;ALUA:xxx;FAILOVERMODE:xxx;PATHTYPE:xxx}
```

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

```
...
    iscsi_info: {HostName:xxx;ACCESSMODE:xxx;HYPERMETROPATHOPTIMIZED:xxx;TargetIP:x.x.x.x y.y.y.y};
            {HostName:xxx;ACCESSMODE:xxx;HYPERMETROPATHOPTIMIZED:xxx;TargetIP:x.x.x.x z.z.z.z},
    fc_info: {HostName:xxx;ACCESSMODE:xxx;HYPERMETROPATHOPTIMIZED:xxx};
            {HostName:xxx;ACCESSMODE:xxx;HYPERMETROPATHOPTIMIZED:xxx}
```

**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 vStore of the remote device. | Yes |
| san_password | Password of a storage administrator or vStore of the remote device. | Yes |
| vstore_name | vStore name of the remote device. This parameter needs to be configured only when a vStore user is used. | No |

| Parameter | Description | Mandatory |
|---|---|---|
| iscsi_info | **HostName** is the regular expression of the host name of the Nova or Cinder node.<br><br>Configuration rule and priority description:<br><br>● **ALUA**, **FAILOVERMODE**, **PATHTYPE**, **ACCESSMODE**, and **HYPERMETROPATHOPTIMIZED** indicate ALUA parameters of the storage device. For details, see **11.1 ALUA Parameter Description**.<br><br>● **TargetIP** indicates the IP address of the iSCSI target port. You can enter multiple IP addresses and separate them with spaces. Different **TargetIP** values can be configured for different hosts. | Mandatory for iSCSI networking |
| fc_info | **HostName** is the regular expression of the host name of the Nova or Cinder node.<br><br>Configuration rule and priority description:<br><br>● **ALUA**, **FAILOVERMODE**, **PATHTYPE**, **ACCESSMODE**, and **HYPERMETROPATHOPTIMIZED** indicate ALUA parameters of the storage device. For details, see **11.1 ALUA Parameter Description**.<br><br>● This parameter can be set for different hosts. | Mandatory for FC networking |

**Step 3** Restart the Cinder Volume service.

**Step 4** Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 5** Enable the remote replication property. *<vtype>* indicates the name of the volume type created in **Step 4**.

```
# cinder type-key <vtype> set capabilities:replication_enabled='<is> true'
```

**Step 6** (Optional) Set the replication type.

```
# cinder type-key <vtype> set replication_type='<in> ***'
```

📖 NOTE

The value of *\*\*\** can be:

● **sync**: synchronous replication

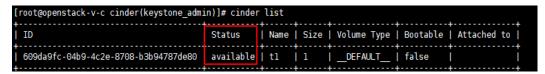● **async**: asynchronous replication

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

**Step 7** Use the volume type created in **Step 4** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 8**    Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.9 Creating a Volume in a Specified Storage Pool

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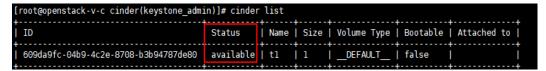
## Procedure

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

**Step 2**    Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3**    Configure a single storage pool. *<vtype>* indicates the name of the volume type created in **Step 2**.

```
# cinder type-key <vtype> set pool_name=StoragePool001
```

Alternatively, configure multiple storage pools.

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

**Step 4**    Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 5**    Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.10 Creating a Volume on a Disk of a Specified Type

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

## Procedure

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

**Step 2**    Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3**   Configure a disk type. *<vtype>* indicates the name of the volume type created in **Step 2**.

```
# cinder type-key <vtype> set disk_type=sas
```

Alternatively, configure multiple disk types.
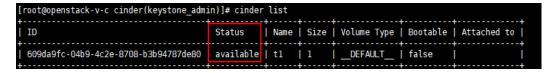```
# cinder type-key <vtype> set disk_type="<or> sas <or> ssd"
```

> ☐ **NOTE**
>
> 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 4**   Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 5**   Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.11 Creating a Volume on a Specified Backend

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

## Procedure

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

**Step 2**   Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3**   Configure the backend name *<vtype>* indicates the name of the volume type created in **Step 2**.

```
# 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 4**   Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 5**   Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.12 Configuring Snapshot Backup

This section describes how to configure the snapshot backup function.

## Procedure

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

**Step 2**  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 3**  Run the **vi /etc/cinder/cinder.conf** command to configure the backend of the configuration file. Add the following configuration to the file.

backup_use_temp_snapshot = True

**Step 4**  Restart the Cinder Volume and Cinder Backup services.

systemctl restart openstack-cinder-backup.service
systemctl restart openstack-cinder-volume.service

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume and Cinder Backup services are successfully started.

```
+----------------+---------------------+------+---------+-------+--------------------------+---------+-----------------+---------------+
| Binary         | Host                | Zone | Status  | State | Updated_at               | Cluster | Disabled Reason | Backend State |
+----------------+---------------------+------+---------+-------+--------------------------+---------+-----------------+---------------+
| cinder-back    | openstack-wallaby-1 | nova | enabled | up    | 2021-12-03T07:56:23.000000 | -     | -               | --            |
| cinder-volume  | openstack-wallaby-1@v6 | nova | enabled | up  | 2021-12-03T07:56:23.000000 | -     | -               | up            |
+----------------+---------------------+------+---------+-------+--------------------------+---------+-----------------+---------------+
```

**----End**

# 8.13 Configuring Certificate Authentication

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

## Procedure

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

**Step 2**  Set the configuration file of Huawei Cinder Driver. Add **SSLCertVerify** and **SSLCertPath** in the **<Storage>** section.

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
  <Storage>
    ...
    <SSLCertVerify>True</SSLCertVerify>
    <SSLCertPath>***</SSLCertPath>
```

```
</Storage>
...
</config>
```

📖 **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 3** Run the following command to restart the Cinder Volume service.

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

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+----------------+------------------------+------+---------+-------+---------------------------+---------+-----------------+---------------+
| Binary         | Host                   | Zone | Status  | State | Updated_at                | Cluster | Disabled Reason | Backend State |
+----------------+------------------------+------+---------+-------+---------------------------+---------+-----------------+---------------+
| cinder-volume  | openstack-wallaby-1@v6 | nova | enabled | up    | 2021-12-03T07:56:23.000000 | -       | -               | up            |
+----------------+------------------------+------+---------+-------+---------------------------+---------+-----------------+---------------+
```

**----End**

# 8.14 Configuring the Owning Controller of a LUN

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

## Procedure

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

**Step 2** Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3** Enable the **huawei_controller** property. *<vtype>* indicates the name of the volume type created in **Step 2**.

```
# cinder type-key <vtype> set capabilities:huawei_controller='<is> true'
```
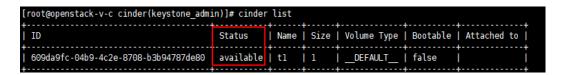
**Step 4** Configure the owning controller name.

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

📖 **NOTE**

The format of the owning controller name is **CTE0.A**. Take OceanStor Dorado V6 as an example. On DeviceManager, choose **System** > **Hardware** > **Controllers** to obtain the controller name.

**Step 5** Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 6** Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.
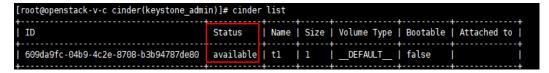
```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.15 Configuring the SmartDedupe Property

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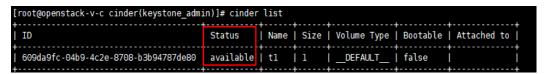
## Enabling SmartDedupe

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

**Step 2** Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3** Enable the SmartDedupe property.

- For Newton and later versions, run the following command to enable the SmartDedupe property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

- For Mitaka and earlier versions, run the following command to enable the SmartDedupe property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

**Step 4** Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 5** Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

## ◯◯ NOTE

OceanStor Dorado supports only Thin volumes. The **capabilities:thin_provisioning_support** parameter in **Step 3** is optional.

## Disabling SmartDedupe

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

**Step 2**    Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3**    Disable the SmartDedupe property.

- For Newton and later versions, run the following command to disable the SmartDedupe property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

- For Mitaka and earlier versions, run the following command to disable the SmartDedupe property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

**Step 4**    Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 5**    Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

📖 **NOTE**

OceanStor Dorado supports only Thin volumes. The **capabilities:thin_provisioning_support** parameter in **Step 3** is optional.

# 8.16 Configuring the SmartCompression Property

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

## Enabling SmartCompression

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

**Step 2**    Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3**    Enable the SmartCompression property.

- For Newton and later versions, run the following command to enable the SmartCompression property. *<vtype>* indicates the name of the volume type created in **Step 2**.

  ```
  # cinder type-key <vtype> set capabilities:thin_provisioning_support='<is> true'
  capabilities:compression='<is> true'
  ```
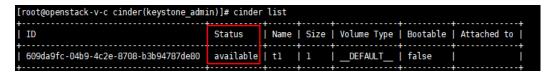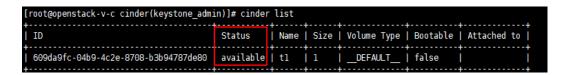
- For Mitaka and earlier versions, run the following command to enable the SmartCompression property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

**Step 4** Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 5** Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

📖 **NOTE**

> OceanStor Dorado supports only Thin volumes. The **capabilities:thin_provisioning_support** parameter in **Step 3** is optional.
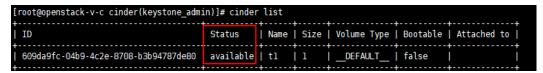
## Disabling SmartCompression

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

**Step 2** Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3** Disable the SmartCompression property.

- For Newton and later versions, run the following command to disable the SmartCompression property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

- For Mitaka and earlier versions, run the following command to disable the SmartCompression property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

**Step 4** Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 5** Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

📖 **NOTE**

> OceanStor Dorado supports only Thin volumes. The **capabilities:thin_provisioning_support** parameter in **Step 3** is optional.

# 8.17 Creating a Volume Using FastClone

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

## Procedure

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

**Step 2**  When creating volumes from snapshots or cloning volumes, specify the **--metadata** parameter with the **fastclone** item as follows:
```
# cinder create --snapshot-id *** --metadata fastclone=True
# cinder create --source-volid *** --metadata fastclone=True
```

**Step 3**  Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

📖 **NOTE**

- If **fastclone** is not set, LUN copy is used by default.
- Only OceanStor Dorado V3/V6 supports FastClone. For details, contact Huawei storage R&D engineers.
- When FastClone is used, there are restrictions on the HyperMetro, remote replication, snapshot, and clone features.

# 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.

## Procedure

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

**Step 2** When creating volumes from snapshots or cloning volumes, specify the **--metadata** parameter with the **copyspeed** item as follows: For details about the value of **copyspeed**, see **Table 8-5**.

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

**Table 8-5** copyspeed parameter description

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

**----End**

◫ 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.

◫ NOTE

- This configuration applies only to OceanStor Dorado V300R001C20 and later versions.
- This configuration also includes compression and deduplication. Their priorities are higher than those of SmartDedupe and SmartCompression. If both of them are configured, the configuration of the application type takes precedence.
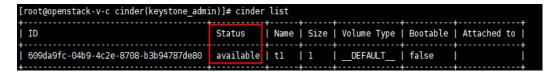
**Procedure**

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

**Step 2** Create a volume type. *<name>* indicates the name of the volume type.

```
# cinder type-create <name>
```

**Step 3** Enable the **huawei_application_type** property. *<vtype>* indicates the name of the volume type created in **Step 2**.

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

**Step 4** Configure the application type name of the LUN.

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

**Step 5** Use the volume type created in **Step 2** to create a volume. *10* indicates the volume size (unit: GB), *<volume-name>* indicates the volume name, and *<name>* indicates the name of the volume type.

```
# cinder create 10 --name <volume-name> --volume-type <name>
```

**Step 6** Run the **cinder list** command. Check the command output. If the volume status is **available**, the volume is successfully created.

```
[root@openstack-v-c cinder(keystone_admin)]# cinder list
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 609da9fc-04b9-4c2e-8708-b3b94787de80 | available | t1   | 1    | __DEFAULT__ | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

**----End**

# 8.20 Configuring a vStore User

This section describes how to configure a vStore user.

## Prerequisites

For OceanStor Dorado storage, only OceanStor Dorado 6.1.3 and later versions support this feature.

## Procedure

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

**Step 2** Run the **vi /etc/cinder/cinder_huawei_conf.xml** command. Modify the **cinder_huawei_conf.xml** configuration file as follows:

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
  <Storage>
    …
    <UserName>xxx</UserName>
    <UserPassword>xxx</UserPassword>
    <vStoreName>xxx</vStoreName>
  </Storage>
  …
</config>
```

**Table 8-6** vStore parameters

| Parameter | Description |
|---|---|
| UserName | User name of the storage vStore. |
| UserPassword | User password of the storage vStore. |
| vStoreName | vStore name. |

> **NOTICE**
>
> - For details about how to configure other parameters in the **cinder_huawei_conf.xml** configuration file, see **Step 3**.
> - vStore users can be configured only when the storage system version supports the multi-tenant feature.

**Step 3** Run the following command to restart the Cinder Volume service.

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

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+---------------+------------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| Binary        | Host                   | Zone | Status  | State | Updated_at                 | Cluster | Disabled Reason | Backend State |
+---------------+------------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| cinder-volume | openstack-wallaby-1@v6 | nova | enabled | up    | 2021-12-03T07:56:23.000000 | -       | -               | up            |
```

**----End**

# 8.21 Configuring Multiple Backends

This section describes how to configure multiple backends.

## Procedure

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

**Step 2** Run the **vi /etc/cinder/cinder.conf** command. Modify the **cinder.conf** configuration file. 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
```

**Step 3** Run the following command to restart the Cinder Volume service.

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

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+---------------+------------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| Binary        | Host                   | Zone | Status  | State | Updated_at                 | Cluster | Disabled Reason | Backend State |
+---------------+------------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| cinder-volume | openstack-wallaby-1@v6 | nova | enabled | up    | 2021-12-03T07:56:23.000000 | -       | -               | up            |
```

**----End**

# 8.22 Configuring Automatic Zoning

The automatic zoning function is optional. To use it, perform the following operations.

> **NOTICE**
>
> vStore users do not support automatic zoning.

## Procedure

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

**Step 2**   Configure the **cinder.conf** file. Configure switch parameters by referring to **https://docs.openstack.org/cinder/latest/configuration/block-storage/fc-zoning.html**.

**Step 3**   Run the following command to restart the Cinder Volume service.

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

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+----------------+-----------------------+-------+---------+-------+-------------------------+---------+-----------------+---------------+
| Binary         | Host                  | Zone  | Status  | State | Updated_at              | Cluster | Disabled Reason | Backend State |
+----------------+-----------------------+-------+---------+-------+-------------------------+---------+-----------------+---------------+
| cinder-volume  | openstack-wallaby-1@v6|  nova | enabled | up    | 2021-12-03T07:56:23.000000 | -    | -               | up            |
+----------------+-----------------------+-------+---------+-------+-------------------------+---------+-----------------+---------------+
```

**----End**

# 8.23 Configuring the CHAP Feature

On a public network, any application server that resides on the same network as the iSCSI front-end port of a storage system can access the storage system and perform read and write operations on the system. Therefore, data security risks arise. To ensure the security of storage systems, you can configure CHAP authentication to restrict application servers' access permissions on storage systems.

## Procedure

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

**Step 2**   Add the following information to the **Initiator** configuration item in the Huawei Cinder Driver configuration file:

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

**Table 8-7** Parameter description

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

**Step 3** Run the following command to restart the Cinder Volume service.

systemctl restart openstack-cinder-volume.service

Wait about 30 seconds and run the **cinder service-list** command. If the following information is displayed and the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+----------------+------------------------+------+---------+-------+---------------------------+---------+-----------------+---------------+
| Binary         | Host                   | Zone | Status  | State | Updated_at                | Cluster | Disabled Reason | Backend State |
+----------------+------------------------+------+---------+-------+---------------------------+---------+-----------------+---------------+
| cinder-volume  | openstack-wallaby-1@v6 | nova | enabled | up    | 2021-12-03T07:56:23.000000 | -      | -               | up            |
```

**----End**

# 8.24 Configuring the In-band Feature

The Cinder Driver plug-in adapts to the in-band command channel (service data and management and control data is transmitted through the same channel). In this way, requests delivered by Cinder can be sent to a storage array using the host agent to complete go-online, go-offline, and resource switchover requests.

This section describes how to configure the in-band feature.

> **NOTICE**
>
> The in-band feature is supported only by OceanStor Dorado 6.1.3 and later versions. The OpenStack version must be version O. The host agent version must be In-Band_Agent 1.0.RC1.

## Prerequisites

The host agent has been configured on the host. For details about the host agent, see **https://support.huawei.com/enterprise/en/centralized-storage/ultrapath-pid-8576127**.

---

⚠ **CAUTION**

After an in-band LUN is configured, this plug-in does not clear the in-band LUN resource of the host during resource clearing. You need to manually clear resources after the storage device where the LUN resides no longer provides services to reclaim resources in time.

---

## Procedure

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

**Step 2** Run the following commands to open the xml file (**cinder_huawei_conf.xml** is used as an example) created in **6 Configuring Basic Properties of Cinder Driver**:

```
cd /etc/cinder
vi cinder_huawei_conf.xml
```

**Step 3** Modify and add parameters under **<Storage>** as follows. **Table 8-8** describes the parameters.

```
…
  <Storage>
    <InBandOrNot>True</InBandOrNot>
    <Storagesn>xxxxxxxxxxxx</Storagesn>
    <RestURL>https://127.0.0.1:8889/deviceManager/rest/</RestURL>
    <UserName>***</UserName>
    <UserPassword>***</UserPassword>
  </Storage>
…
```

**Table 8-8** In-band parameters

| Parameter | Optional/ Mandatory/ Conditionally Mandatory | Description |
|---|---|---|
| InBandOrNot | Optional | Whether in-band commands are supported.<br>● The value can be **True** or **False**.<br>● The default value is **False**. |

| Parameter | Optional/ Mandatory/ Conditionally Mandatory | Description |
|---|---|---|
| Storagesn | Conditionally mandatory | Storage device SN.<br>• When the value of **InBandOrNot** is **True**:<br>Enter the SN of the storage device, which can be queried on the storage device to be connected.<br>• When the value of **InBandOrNot** is **False**:<br>This parameter is optional. |
| RestURL | Conditionally mandatory | URL for logging in to the storage device.<br>• When the value of **InBandOrNot** is **True**:<br>This parameter indicates the address for accessing the host agent. If the host agent is not modified, set this parameter to **https://127.0.0.1:8889/ deviceManager/rest/**.<br>• When the value of **InBandOrNot** is **False**:<br>Enter the URL of the storage device. |
| UserName | Conditionally mandatory | User name for logging in to the storage device.<br>• When the value of **InBandOrNot** is **True**:<br>Enter the user name of the in-band account. For details about how to create and obtain the in-band account, see the host agent documentation.<br>• When the value of **InBandOrNot** is **False**:<br>Enter the user name of the storage administrator. The user name must be an administrator or super administrator. |

| Parameter | Optional/ Mandatory/ Conditionally Mandatory | Description |
|---|---|---|
| UserPassword | Conditionally mandatory | Password for logging in to the storage device. This parameter is mandatory.<br><br>● When the value of **InBandOrNot** is **True**:<br>Enter the password of the in-band account. For details about how to create and obtain the in-band account, see the host agent documentation.<br><br>● When the value of **InBandOrNot** is **False**:<br>Enter the password of the storage administrator. |
| san_address | Conditionally mandatory | This parameter is available only for configuration items that require HyperMetro or remote replication.<br><br>● When the value of **InBandOrNot** is **True**:<br>This parameter indicates the address for accessing the host agent. If the host agent is not modified, set this parameter to **https://127.0.0.1:8889/ deviceManager/rest/**.<br><br>● When the value of **InBandOrNot** is **False**:<br>Enter the URL of the storage device. |

| Parameter | Optional/ Mandatory/ Conditionally Mandatory | Description |
|---|---|---|
| san_user | Conditionally mandatory | This parameter is available only for configuration items that require HyperMetro or remote replication.<br><br>● When the value of **InBandOrNot** is **True**:<br>Enter the user name of the in-band account of the remote storage device. For details about how to create and obtain the in-band account, see the host agent documentation.<br><br>● When the value of **InBandOrNot** is **False**:<br>Enter the user name or vStore user name of the remote storage administrator. |
| san_password | Conditionally mandatory | This parameter is available only for configuration items that require HyperMetro or remote replication.<br><br>● When the value of **InBandOrNot** is **True**:<br>Enter the password of the in-band account of the remote storage device. For details about how to create and obtain the in-band account, see the host agent documentation.<br><br>● When the value of **InBandOrNot** is **False**:<br>Enter the password or vStore user password of the remote storage administrator. |

**Step 4** (Optional) Configure the HyperMetro property.

Run the **vi /etc/cinder/cinder.conf** command to modify the information about the remote device configured in the specified backend. Modify the in-band command parameters as follows. For details about the parameters, see **Table 8-8**.

After the configuration is modified, if you want to use the HyperMetro property, restart the Cinder Volume service, and create a volume type, see **8.7 Configuring the HyperMetro Property**.

```
hypermetro_device =
  ...
  in_band_or_not:True,
  storage_sn:xxxxxxxx,
  san_address:https://127.0.0.1:8889/deviceManager/rest/,
```

```
    san_user:***,
    san_password:***,
    …
```

**Step 5**    (Optional) Configure the remote replication property.

Run the **vi /etc/cinder/cinder.conf** command to modify the information about the remote device configured in the specified backend. Modify the in-band command parameters as follows. For details about the parameters, see **Table 8-8**.

After the configuration is modified, if you want to use the remote replication property, restart the Cinder Volume service, and create a volume type, see **8.8 Configuring the Remote Replication Property**.

```
replication_device =
    …
    in_band_or_not:True,
    storage_sn:xxxxxxxx,
    san_address:https://127.0.0.1:8889/deviceManager/rest/,
    san_user:***,
    san_password:***,
    …
```

**Step 6**    Run the following command to restart the Cinder Volume service.

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

Wait about 30 seconds and run the **cinder service-list** command. If the value of **State** is **up**, the Cinder Volume service is successfully started.

```
+----------------+------------------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| Binary         | Host                         | Zone | Status  | State | Updated_at                 | Cluster | Disabled Reason | Backend State |
+----------------+------------------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
| cinder-volume  | openstack-wallaby-1@v6       | nova | enabled | up    | 2021-12-03T07:56:23.000000 | -       | -               | up            |
+----------------+------------------------------+------+---------+-------+----------------------------+---------+-----------------+---------------+
```

**----End**

# 9 Best Practices

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

This section describes the best practices of the xml file for interconnection between iSCSI and OceanStor Dorado V3.

## Prerequisites

The following configuration must be performed after the operations in **4 Installing Cinder Driver** and **7.1 Configuring Host Multipathing** are complete.

## Procedure

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

**Step 2** Configure the **/etc/cinder/cinder_huawei_conf.xml** file as follows (for iSCSI networking).

```
<?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"
PATHTYPE="0" TargetIP="192.168.128.100 192.168.128.101"/>
```

```
      <Initiator HostName="host[5-9]" ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0"
PATHTYPE="0" TargetIP="192.168.128.102 192.168.128.103"/>
    </iSCSI>
</config>
```

📖 **NOTE**

- This configuration example applies to iSCSI networking. For details about FC networking, see **Step 3**.
- For details about the parameter description and procedure, see **6.1.1 Configuring Cinder Driver on an iSCSI Network**.
- ALUA parameter settings vary depending on the OS type. For details, see **11.1 ALUA Parameter Description**.

**Step 3** Configure the **/etc/cinder/cinder_huawei_conf.xml** file as follows (for FC networking).

```
<?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>
  <FC>
    <MinOnlineFCInitiator>1</MinOnlineFCInitiator>
    <Initiator HostName="xxx" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
    <Initiator HostName="xxx" ALUA="xxx" FAILOVERMODE="XXX" PATHTYPE="xxx"/>
  </FC>
</config>
```

📖 **NOTE**

- This configuration example applies to FC networking. For details about iSCSI networking, see **Step 2**.
- For details about the parameter description and procedure, see **6.1.2 Configuring Cinder Driver on an FC Network**.
- ALUA parameter settings vary depending on the OS type. For details, see **11.1 ALUA Parameter Description**.

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

**Step 5** Configure the **/etc/cinder/cinder.conf** 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
use_multipath_for_image_xfer = true
enforce_multipath_for_image_xfer = true
```

Add backend **huawei_backend** in the **[DEFAULT]** section.

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

**Step 6** Configure OpenStack Nova multipathing.

Add the following content to the **[libvirt]** section in the **/etc/nova/nova.conf** file.

```
[libvirt]
…
 volume_use_multipath = True
```

**Step 7** 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 Examples

## Example 1

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**, both of the preceding configuration items can be used to configure information for the initiator, such as ALUA and port group. According to the rules (subject to the precise configuration), the priority of the second configuration item (**HostName="host"**) is higher than that of the first configuration item (**HostName="*"**).

## Example 2

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**, both of the preceding configuration items meet the configuration rules. According to the rules (subject to the first configuration item), the first configuration item is used to configure the initiator.

## Example 3

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**, both of the preceding configuration items can be used to configure information for the initiator, such as ALUA and port group. According to the rules (subject to the precise configuration), the priority of the second configuration item (**HostName="host"**) is higher than that of the first configuration item (**HostName="*"**).

## Example 4

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**, both of the preceding configuration items meet the configuration rules. According to the rules (subject to the first configuration item), the first configuration item is used to configure the initiator.

## Example 5:

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 item is used to configure the initiator. If the host name is **Host10**, the second configuration item is used to configure the initiator. **^** matches the beginning of a character string, and **$** matches the end of a character string.

## Example 6:

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 item is used to configure the initiator. If the host name is **Host10**, the second configuration item is used to configure the initiator. **^** matches the beginning of a character string, and **$** matches the end of a character string.

# 10 FAQ

## 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 the associated objects and mapping views. If these objects are manually created on the storage, the mapping logic of Cinder Driver will be affected. Therefore, ensure that these objects are deleted before mapping volumes using Cinder Driver.

# 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 the execute permission.

# 10.3 Failed to Create a Volume from an Image After verify_glance_signatures = enabled Is Configured

## Symptom

OpenStack creates a volume from an image. After the image is copied for a period of time, the error message **Failed to copy metadata to volume: Glance metadata cannot be updated, key signature_verified exists for volume id b75f957c-0ba4-4099-8706-866c68012779** is displayed. As a result, the volume fails to be created.

## Environment Configuration

- OS: CentOS 7.6
- Server model: 5288 V5
- OpenStack version: Rocky

## Root Cause Analysis

Community bug link: **https://bugs.launchpad.net/cinder/+bug/1823445**

In Rocky and later versions, the **verify_glance_signatures** field is added to **cinder.image.glance.py**. The default value is **enabled**. When **CONF.verify_glance_signatures != 'disabled'** is configured and **signature_verification** exists in the image, the DB API does not allow **signature_verification** to be written repeatedly. As a result, an exception is thrown.

## Solution or Workaround

In the **[DEFAULT]** section of the **/etc/cinder/cinder.conf** file on the OpenStack Cinder node, set the value of the **verify_glance_signatures** field to **disabled**. An example is provided as follows:

```
[DEFAULT]
…
verify_glance_signatures = disabled
…
```

# 10.4 "alua not supported" Is Displayed in the Command Output When I Run the multipath -ll Command to Query Multipathing Information

## Symptom

After host multipathing and OpenStack multipathing are configured, a VM is created or migrated and the **multipath -ll** command is run to query multipathing information. As a result, **alua not supported** is displayed for some drive letters.

## Environment Configuration

OpenStack version: all

## Root Cause Analysis

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

## Solution or Workaround

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

2. Set the host multipathing parameters by referring to **7.1 Configuring Host Multipathing**. Perform this step on all compute nodes.

3. Configure ALUA by referring to the **Initiator** field in **6.1.1 Configuring Cinder Driver on an iSCSI Network** or **6.1.2 Configuring Cinder Driver on an FC Network**. For details about the parameters, see **11.1 ALUA Parameter Description**. Perform this step on all Cinder Volume nodes.

4. Restart the multipathd service.
   ```
   systemctl restart multipathd.service
   ```

5. Restart the Cinder-volume service.
   ```
   systemctl restart openstack-cinder-volume.service
   ```

6. Re-map all created VMs (unmap volumes and then map volumes again). You can perform operations such as rebuilding and migration.

# 10.5 Failed to Delete or Live Migrate a VM

## Symptom

A VM fails to be deleted or live migrated. The following information is displayed in the Nova Compute log:
```
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 query multipathing information. **failed faulty** is displayed for some paths.

### Environment Configuration

OpenStack version: all

### Root Cause Analysis

The **/dev/sd\*** drive letter forming **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 or Workaround

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

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

2. Configure the **filter** field in the **devices** section of the **/etc/lvm/lvm.conf** file. The following is an example. Perform this step on all compute nodes.
   ```
   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.

### Symptom

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 Configuration

- OpenStack version: all
- Storage versions:
  - OceanStor Dorado 6.1.0/6.1.2/6.1.3/6.1.5
  - OceanStor 6.1.3/6.1.5

### Root Cause Analysis

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 or Workaround

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

# 10.7 How Do I Manually Map a LUN to an OpenStack Node Without Affecting the Volume Mapping of Cinder Driver?

📖 **NOTE**

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 node based on the following rules.

## Solution

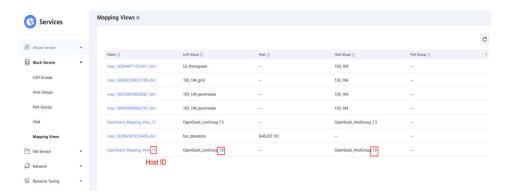Manually create host groups, LUN groups, and mapping views on the storage based on the following Cinder Driver rules.

📖 **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 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 OpenStack 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())
   0e4ed842f01b44143dd03dab6e04739f
   ```

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 **1**.

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

– A host group is named as follows:

**OpenStack_HostGroup_**_host_id_

Example: **OpenStack_HostGroup_1**

– A LUN group is named as follows:

**OpenStack_LunGroup_**_host_id_

Example: **OpenStack_LunGroup_1**

– A mapping view is named as follows:

**OpenStack_Mapping_View_**_host_id_

Example: **OpenStack_Mapping_View_1**

📖 **NOTE**

Mapping view names of OceanStor Dorado V6 cannot be changed on DeviceManager. Perform the following steps to change the name:

1. Run the **ssh** *Username***@***IP* command to go to the CLI of the storage device. (The password for logging in to the CLI is the same as that for logging in to the GUI.)

2. Run the **change mapping_view mapping_view_name=**_Old_map_ **name=**_New_map_ command to change the name of a mapping view.

Parameters in the command are described as follows:

● *Username*: user name for logging in to the storage device.

● *IP*: IP address of the storage device.

● *Old_map*: old name.

● *New_map*: new name.

# 10.8 Image Volumes Fail to Be Created in Batches for the Mitaka Version Due to Residual Paths

## Symptom

When the system is interconnected with OpenStack Mitaka, image volumes fail to be created in batches due to residual paths.

## Environment Configuration

OpenStack version: Mitaka

## Root Cause Analysis

For details about problems in disk scanning and detaching on the OpenStack cloud platform, see **https://bugs.launchpad.net/os-brick/+bug/1553730**.

## Solution or Workaround

Contact the corresponding cloud platform vendor for maintenance and modification, or manually clear the paths that do not exist.

# 11 Appendix

## 11.1 ALUA Parameter Description

### 11.1.1 ALUA Parameter Description for OceanStor 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 *Huawei SAN Storage Host Connectivity Guide for Red Hat* and choose **Configuring Multipathing** > **Configuring Multipathing in Non-HyperMetro Scenarios**.

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

**Table 11-1** ALUA parameter description

| Parameter | Default Value | Description |
|-----------|---------------|-------------|
| ALUA | 0 | Multipathing type. The value can be:<br>- **0**: default (Huawei UltraPath)<br>- **1**: third-party multipathing |

| Parameter | Default Value | Description |
|---|---|---|
| FAILOVERMODE | - | Initiator switchover mode. This parameter needs to be specified only when third-party multipathing is used. The value can be:<br>● **0**: early-version ALUA<br>● **1**: common ALUA<br>● **2**: ALUA not used<br>● **3**: special ALUA |
| SPECIALMODETYPE | - | Special mode type of the initiator. This parameter needs to be specified only when the initiator switchover mode is special ALUA. The value can be:<br>● **0**: mode 0 (special mode 0)<br>● **1**: mode 1 (special mode 1)<br>● **2**: mode 2 (special mode 2)<br>● **3**: mode 3 (special mode 3) |
| PATHTYPE | - | Initiator path type. This parameter needs to be specified only when third-party multipathing is used. The value can be:<br>● **0**: preferred path<br>● **1**: non-preferred path<br>**NOTE**<br>Set this parameter to **0** for the storage device that is accessed preferentially and to **1** for the storage device that is not accessed preferentially. |

> **NOTICE**

**HyperMetro scenarios:**

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

- For hosts running CentOS/RHEL series:
    - Storage with the preferred path: **ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0"**
    - Storage with the non-preferred path: **ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="1"**
- For hosts running Ubuntu/SUSE/Debain series:
    - Storage with the preferred path: **ALUA="1" FAILOVERMODE="1" PATHTYPE="0"**
    - Storage with the non-preferred path: **ALUA="1" FAILOVERMODE="1" PATHTYPE="1"**

**Non-HyperMetro scenarios:**

For OceanStor V3/V5 and OceanStor Dorado V3:

- Method 1 (recommended): Configure ALUA as follows:
    - For hosts running CentOS/RHEL series:

      **ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0"**
    - For hosts running Ubuntu/SUSE/Debain series:

      **ALUA="1" FAILOVERMODE="1" PATHTYPE="0"**
- Method 2: Do not configure ALUA.

# 11.1.2 ALUA Parameter Description for OceanStor Dorado V6

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

**Table 11-2** ALUA parameter description

| Parameter | Default Value | Description |
|---|---|---|
| ACCESSMODE | 0 | Host access mode. This parameter is applicable only to OceanStor Dorado V6. The value can be:<br><br>● **0**: balanced (default)<br><br>● **1**: asymmetric<br><br>If both **ACCESSMODE** and **ALUA** are configured, **ACCESSMODE** takes precedence. |
| HYPERMETROPATHO PTIMIZED | - | HyperMetro preferred path. This parameter is optional when the host access mode is asymmetric. The value can be:<br><br>● **0**: no<br><br>● **1**: yes |

## Creating OceanStor Dorado V6 All-Flash Storage

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

**Table 11-3** ALUA parameters for OceanStor Dorado V6

| Customer Scenario | Host OS type | Parameter Configuration |
|---|---|---|
| HyperMetro scenario: load balancing | CentOS/RHEL/Ubuntu/ SUSE/Debain | You do not need to configure ALUA. That is, delete the **ACCESSMODE** and **HYPERMETROPATHOPTI-MIZED** parameters from the **initiator** property. |
| HyperMetro scenario: local preferred – storage with the preferred path | CentOS/RHEL/Ubuntu/ SUSE/Debain | ACCESSMODE="1" HYPERMETROPATHOPTI-MIZED="1" |
| HyperMetro scenario: local preferred – storage with the non-preferred path | 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 | You do not need to configure ALUA. That is, delete the **ACCESSMODE** and **HYPERMETROPATHOPTI-MIZED** parameters from the **initiator** property. |

### Coexistence of OceanStor Converged Storage and OceanStor Dorado V6 All-Flash Storage

If you use OceanStor converged storage and then expand the capacity using OceanStor Dorado all-flash 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** ALUA parameters for OceanStor Dorado V6

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

# 11.2 Configuring a 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**.

For details about the configuration, see OpenStack official documentation of the desired version.

# 11.3 Configuring the 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**.

For details about the configuration, see OpenStack official documentation of the desired version.