

OpenStack Cinder Driver

Configuration Guide

Issue 01

Date 2017-12-08



Copyright © Huawei Technologies Co., Ltd. 2017. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base

Bantian, Longgang Shenzhen 518129

People's Republic of China

Website: http://www.huawei.com
Email: support@huawei.com

Contents

1 Overview	1
2 Version Mappings	2
3 Cinder Driver Installation&deployment	8
3.1 Obtain Cinder Driver	
3.2 Ubuntu Environment Deployment	
3.3 RedHat OpenStack Deployment	
4 Cinder Driver Basis Properties Configuration	
4.1 Configuring the Cinder Driver for iSCSI Storage	
4.1.1 Configuring the Cinder Driver for OceanStor T Series V1 (iSCSI)	
4.1.2 Configuring the Cinder Driver for OceanStor T Series V2 (iSCSI)	
4.1.3 Configuring the Cinder Driver for OceanStor V3/V5(iSCSI)	14
4.1.4 Configuring the Cinder Driver for OceanStor 18000 Series(iSCSI)	15
4.1.4.1 Configuring the Cinder Driver for OceanStor 18000 Series V1(iSCSI)	15
4.1.4.2 Configuring the Cinder Driver for OceanStor 18000 Series V3(iSCSI)	16
4.1.5 Configuring the Cinder Driver for Dorado Series V3(iSCSI)	17
4.1.6 Parameters in the Configuration File.	18
4.1.7 Configuring iSCSI Multipathing	20
4.1.8 Configuring CHAP and ALUA	21
4.2 Configuring the Cinder Driver for Fibre Channel Storage	22
4.2.1 Configuring the Cinder Driver for OceanStor T Series V1 (Fibre Channel)	22
4.2.2 Configuring the Cinder Driver for OceanStor T Series V2 (Fibre Channel)	23
4.2.3 Configuring the Cinder Driver for OceanStor V3/V5 (Fibre Channel)	24
4.2.4 Configuring the Cinder Driver of OceanStor 18000 Series (Fibre Channel)	25
4.2.4.1 Configuring the Cinder Driver for OceanStor 18000 Series V1(Fibre Channel)	25
4.2.4.2 Configuring the Cinder Driver for OceanStor 18000 Series V3(Fibre Channel)	26
4.2.5 Configuring the Cinder Driver for Dorado Series V3(Fibre Channel)	28
4.2.6 Configuring the Auto Zoning.	29
4.2.7 Parameters in the Configuration File	29
4.2.8 Configuring FC Multipathing.	31
4.3 Configuring Multi-Storage Support	31
5 Cinder Driver Advanced Properties Configuration	32
5.1 Configuring the SmartQoS Property	33

5.1.1 Configuration the Front and Oos	22
5.1.1 Configuring the Front-end QoS	
5.1.2 QoS Configuration Versions	
5.1.3 Configuring the Control IOPS.	
5.1.3.1 Configuring the Maximum Control IOPS	
5.1.3.2 Configuring the Minimum Control IOPS	
5.1.4 Configuring the Control Bandwidth	
5.1.4.1 Configuring the Maximum Control Bandwidth	37
5.1.4.2 Configuring the Minimum Control Bandwidth	38
5.1.5 Configuring the Control Latency	39
5.1.6 Configuring Multiple Control Policies.	40
5.2 Configuring the SmartPartition Property.	41
5.3 Configuring the Thick Property	42
5.4 Configuring the Thin Property	43
5.5 Configuring the SmartCache Property	44
5.6 Configuring the SmartTier Property	44
5.7 Configuring the HyperMetro Property	45
5.8 Configuring the Replication V2.1	47
5.9 Create a volume on a certain storage pool	50
5.10 Create a volume with a certain disk type	51
5.11 Create a volume on a certain volume back end.	
5.12 Configuring Consistency Groups	53
5.13 Configuring Backup Snapshot.	
5.14 Configuring Certificate Verify	
6 Best Practices	55
6.1 Quick Interconnection with Huawei Storage	55
7 FAQ	57
7.1 Need to manually create the mapping associated objects like Host, Host Group or Lun Group on Huawei storag	
beforehand before attaching volume via Cinder Driver?	57
7.2 Nova log prints "Ishlk " command execution failed while attaching volume	57

$oldsymbol{1}$ Overview

This chapter describes the definition of the Cinder Driver.

Cinder Driver is a plug-in that is deployed on the OpenStack Cinder module. The plug-in can be used to provide functions such as the logical volume and snapshot for virtual machines (VMs) in the OpenStack Cinder Driver that supports iSCSI and Fibre Channel protocols.

2 Version Mappings

This chapter describes the version mappings among the Cinder Driver, Huawei storage system, and OpenStack.

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

OpenStack	Huawei storage system
Kilo	T V100R005 C02/C30
Liberty	T V200R002 C00/C20/C30
Mitaka	OceanStor V3 V300R001C10/C20
Newton	OceanStor V3 V300R002C10
Ocata	OceanStor V3 V300R003C00/C10/C20
Pike	OceanStor 2200/2600 V3 V300R005C00
	OceanStor V3 V300R006C00/C01
	OceanStor V500R007C00
	OceanStor 18500/18800 V100R001C00/C20/C30
	OceanStor 18500/18800 V300R003C00
	OceanStor 18500/18800 V300R006C00/C01
	OceanStor Dorado V3 V300R001C00/C01
	OceanStor 18500F/18800F V3 high-end all-flash storage system
	OceanStor 6800F V3 high-end all-flash storage system
	OceanStor 2600F/5500F/5600F/5800F V3 all-flash storage system

Table 2-2 Version mappings among the Cinder Driver, Features and OpenStack. (\checkmark : support, x: unsupport)

Features	Kilo	Liberty	Mitaka	Newton	Ocata	Pik e
Create Volume	√	√	√	√	√	√

Features	Kilo	Liberty	Mitaka	Newton	Ocata	Pik e
Delete Volume	√	√	√	√	√	√
Attach Volume	√	√	√	√	√	√
Detach Volume	√	√	√	√	√	√
Extend Volume	√	√	√	√	√	√
Create Snapshot	√	√	√	√	√	√
Delete Snapshot	√	√	√	√	√	√
List Snapshots	√	√	√	√	√	√
Create Volume from Snapshot	√	√	√	√	√	√
Create Volume from Image	√	√	√	√	√	√
Create Volume from Volume	√	√	√	√	√	√
Create Image from Volume	√	√	√	√	√	√
Volume Migration	√	√	√	√	√	√
QoS	√	√	√	√	√	√
Auto zoning	√	√	√	√	√	√
SmartTier	√	√	√	√	√	√
SmartCache	√	√	√	√	√	√
SmartThin/Thick	√	√	√	√	√	√
SmartPartition	√	√	√	√	√	√
HyperMetro	√	√	√	√	√	√
Retype	√	√	√	√	√	√
Manage/Unmanage Volume	√	√	√	√	√	√
Manage/Unmanage Snapshot	X	X	√	√	√	√
Replication V2.1	X	X	√	√	√	√
HyperMetro Consistency Group	х	√	√	√	√	√
Assign StoragePool	√	√	√	√	√	√
Assign DiskType	√	√	√	√	√	√

Features	Kilo	Liberty	Mitaka	Newton	Ocata	Pik e
Backup Snapshot	X	X	√	√	√	√
Snapshot Consistency Group	X	X	√	√	√	√
Multipath	√	√	√	√	√	√
Consistency Group	X	√	√	√	√	√

Table 2-3 Version mappings among the Cinder Driver, Features and Huawei storage system.

Features	Huawei storage system
Create, delete, expand, attach, and	T V100R005 C02/C30
detach volumes	T V200R002 C00/C20/C30
Copy an image to a volume	OceanStor V3 V300R001C10/C20
Copy a volume to an image	OceanStor V3 V300R002C10
Auto Zoning	OceanStor V3 V300R003C00/C10/C20
SmartThin	OceanStor 2200/2600 V3 V300R005C00
Create and delete snapshots	OceanStor V3 V300R006C00/C01
	OceanStor V500R007C00
	OceanStor 18500/18800 V100R001C00/C20/C30
	OceanStor 18500/18800 V300R003C00
	OceanStor 18500/18800 V300R006C00/C01
	OceanStor Dorado V3 V300R001C00/C01
	OceanStor 18500F/18800F V3 high-end all-flash storage system
	OceanStor 6800F V3 high-end all-flash storage system
	OceanStor 2600F/5500F/5600F/5800F V3 all-flash storage system

Features	Huawei storage system
Create volumes with assigned	T V200R002 C00/C20/C30
storage pools	OceanStor V3 V300R001C10/C20
Create volumes with assigned disk	OceanStor V3 V300R002C10
types	OceanStor V3 V300R003C00/C10/C20
Create, delete and update a consistency group	OceanStor 2200/2600 V3 V300R005C00
Manage and unmanage volumes	OceanStor V3 V300R006C00/C01
Volume Migration	OceanStor V500R007C00
Create and delete a cgsnapshot	OceanStor 18500/18800 V100R001C00/C20/C30
Manage, unmanage, and backup	OceanStor 18500/18800 V300R003C00
snapshots	OceanStor 18500/18800 V300R006C00/C01
Replication V2.1	OceanStor Dorado V3 V300R001C00/C01
	OceanStor 18500F/18800F V3 high-end all-flash storage system
	OceanStor 6800F V3 high-end all-flash storage system
	OceanStor 2600F/5500F/5600F/5800F V3 all-flash storage system
Clone a volume	T V100R005 C02/C30
Create volume from snapshot	T V200R002 C00/C20/C30
SmartThick	OceanStor V3 V300R001C10/C20
	OceanStor V3 V300R002C10
	OceanStor V3 V300R003C00/C10/C20
	OceanStor 2200/2600 V3 V300R005C00
	OceanStor V3 V300R006C00/C01
	OceanStor V500R007C00
	OceanStor 18500/18800 V100R001C00/C20/C30
	OceanStor 18500/18800 V300R003C00
	OceanStor 18500/18800 V300R006C00/C01
	OceanStor 18500F/18800F V3 high-end all-flash storage system
	OceanStor 6800F V3 high-end all-flash storage system
	OceanStor 2600F/5500F/5600F/5800F V3 all-flash storage system

Features	Huawei storage system
QoS	T V200R002 C00/C20/C30
Retype	OceanStor V3 V300R001C10/C20
	OceanStor V3 V300R002C10
	OceanStor V3 V300R003C00/C10/C20
	OceanStor 2200/2600 V3 V300R005C00
	OceanStor V3 V300R006C00/C01
	OceanStor V500R007C00
	OceanStor 18500/18800 V100R001C00/C20/C30
	OceanStor 18500/18800 V300R003C00
	OceanStor 18500/18800 V300R006C00/C01
	OceanStor 18500F/18800F V3 high-end all-flash storage system
	OceanStor 6800F V3 high-end all-flash storage system
	OceanStor 2600F/5500F/5600F/5800F V3 all-flash storage system
SmartTier	T V200R002 C00/C20/C30
SmartCache	OceanStor V3 V300R001C10/C20
	OceanStor V3 V300R002C10
	OceanStor V3 V300R003C00/C10/C20
	OceanStor 2200/2600 V3 V300R005C00
	OceanStor V3 V300R006C00/C01
	OceanStor V500R007C00
	OceanStor 18500/18800 V100R001C00/C20/C30
	OceanStor 18500/18800 V300R003C00
	OceanStor 18500/18800 V300R006C00/C01

Features	Huawei storage system
SmartPartition	T V200R002 C00/C20/C30
	OceanStor V3 V300R001C10/C20
	OceanStor V3 V300R002C10
	OceanStor V3 V300R003C00/C10/C20
	OceanStor 2600 V3 V300R005C00
	OceanStor V3 V300R006C00/C01
	OceanStor 2600 V3 V300R006C00/C01
	OceanStor V500R007C00
	OceanStor 18500/18800 V100R001C00/C20/C30
	OceanStor 18500/18800 V300R003C00
	OceanStor 18500/18800 V300R006C00/C01
	OceanStor 18500F/18800F V3 high-end all-flash storage system
	OceanStor 6800F V3 high-end all-flash storage system
	OceanStor 2600F/5500F/5600F/5800F V3 all-flash storage system
Hypermetro	OceanStor V3 V300R003C00/C10/C20
Hypermetro consistency group	OceanStor 2600 V3 V300R005C00
	OceanStor V3 V300R006C00/C01
	OceanStor 2600 V3 V300R006C00/C01
	OceanStor V500R007C00
	OceanStor 18500/18800 V300R003C00
	OceanStor 18500/18800 V300R006C00/C01
	OceanStor Dorado V3 V300R001C00/C01
	OceanStor 18500F/18800F V3 high-end all-flash storage system
	OceanStor 6800F V3 high-end all-flash storage system
	OceanStor 2600F/5500F/5600F/5800F V3 all-flash storage system

3 Cinder Driver Installation&deployment

- 3.1 Obtain Cinder Driver
- 3.2 Ubuntu Environment Deployment
- 3.3 RedHat OpenStack Deployment
- 3.1 Obtain Cinder Driver
- 3.2 Ubuntu Environment Deployment
- 3.3 RedHat OpenStack Deployment

3.1 Obtain Cinder Driver

Two ways to obtain OpenStack Driver:

One is through the OpenStack community warehouse. From Kilo, Huawei has contributed Huawei Storage Driver to OpenStack, users can download OpenStack Driver from OpenStack community for free. After installing the specified OpenStack, OpenStack Driver will be placed under the catalog of "../cinder/cinder/volume/drivers/huawei". If you don't find the corresponding installation files, you can download the OpenStack Driver from OpenStack community warehouse at https://github.com/openstack/cinder.

Another is through Huawei OpenStack Driver warehouse. By visiting https://github.com/huaweistorage/OpenStack_Driver, you can download OpenStack Driver that corresponds to OpenStack community version.

Steps to get Cinder Driver are as follows:

NOTE

- A community version cannot be integrated with new features once being released and has security risks because its debug process is time-consuming.
- However, OpenStack Driver obtained from OpenStack Driver repository can be integrated with newly launched features and debugged in a timely manner, when a supportive framework is used.
- The community only maintains two stable versions. However, Huawei OpenStack Driver library maintains six stable versions, ensuring long-term stable running of historical versions.
- You are strongly recommended to use Huawei OpenStack Driver library versions, instead of community versions.

- **Step 1** Enter the above warehouse address in the browser, for example Huawei warehouse address: https://github.com/huaweistorage/OpenStack_Driver.
- Step 2 Click the "Download ZIP" bottom to download Driver, and unzip it.
- **Step 3** Find "Cinder" catalog in which the Diver is extracted, there are multiple OpenStack Driver for different OpenStack version, choose the corresponding Driver.

----End

3.2 Ubuntu Environment Deployment

The OpenStack standard deployment steps are as follows:

Step 1 Before installation, delete all the installation files of Huawei OpenStack Drver, the default path is /usr/lib/python2.7/dist-packages/cinder/volume/drivers/huawei.



CAUTION

On my host, the version of Python is 2.7, if other version is used, make corresponding changes to the Driver path. You can also get the Cinder Driver installation directory by the following method:

root@ubuntu:~# find / -name huawei driver.py

/usr/lib/python2.7/dist-packages/cinder/volume/drivers/huawei/huawei driver.py

- **Step 2** Copy OpenStack Cinder Driver to Cinder Driver installation directory, the default directory refer to step 1.
- **Step 3** Refer to chapter 4 and 5 to do the configuration.
- **Step 4** After configuration, restart Cinder-Volume service:

service cinder-volume restart

Step 5 Check the status of services by inputting cinder service-list, if the "State" status of Cinder-Volume is up, that means Cinder-Volume is OK.

----End

3.3 RedHat OpenStack Deployment

RedHat OpenStack deployment steps are as follows:

Step 1 Before installation, delete all the installation files of Huawei OpenStack Drver, the default path is /usr/lib/python2.7/disk-packages/cinder/volume/drivers/huawei.



CAUTION

On my host, the version of Python is 2.7, if other version is used, make corresponding changes to the Driver path. You can also get the Cinder Driver installation directory by the following method:

root@redhatL004:~# find / -name huawei_driver.py

/usr/lib/python2.7/dist-packages/cinder/volume/drivers/huawei_driver.py

- **Step 2** Copy OpenStack Cinder Driver to Cinder Driver installation directory, the default directory refer to step 1.
- **Step 3** Refer to chapter 4 and 5 to do the configuration.
- **Step 4** After configuration, restart Cinder-Volume service: systemctl restart openstack-cinder-volume.service
- **Step 5** Check the status of services by inputting cinder service-list, if the "State" status of Cinder-Volume is up, that means Cinder-Volume is OK.

----End

4 Cinder Driver Basis Properties Configuration

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

- 4.1 Configuring the Cinder Driver for iSCSI Storage
- 4.2 Configuring the Cinder Driver for Fibre Channel Storage
- 4.3 Configuring Multi-Storage Support
- 4.1 Configuring the Cinder Driver for iSCSI Storage
- 4.2 Configuring the Cinder Driver for Fibre Channel Storage
- 4.3 Configuring Multi-Storage Support

4.1 Configuring the Cinder Driver for iSCSI Storage

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

Prerequisites

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

```
use_multipath_for_image_xfer = True
enforce multipath for image xfer = True
```

4.1.1 Configuring the Cinder Driver for OceanStor T Series V1 (iSCSI)

Procedure

Step 1 In /etc/cinder, create a Huawei-customized driver configuration file. The file format is XML. Change the name of the driver configuration file based on site requirements, for example, cinder huawei conf.xml.

Step 2 Configure parameters in the driver configuration file, including mandatory and optional parameters.

Mandatory parameters:

```
<?xml version='1.0' encoding='UTF-8'?>
 <config>
      <Storage>
         <Product>T</Product>
         <Protocol>iSCSI</Protocol>
         <ControllerIP0>x.x.x.x</ControllerIP0>
          <ControllerIP1>x.x.x.x/ControllerIP1>
         <UserName>xxx</UserName>
         <UserPassword>xxx</UserPassword>
      </Storage>
      <T.UN>
         <StoragePool Name="xxx" />
         <StoragePool Name="xxx" />
      </LUN>
      <iSCSI>
         <DefaultTargetIP>x.x.x.x/DefaultTargetIP>
      </iscsi>
      <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x" />
```

Optional parameters(Pick and add the optional parameter in the above XML file):

NOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

```
-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
```

For details about the parameters in the configuration file, see section "4.1.6".

Step 3 Configure the **cinder.conf** file.

Add a [TV1_iSCSI] section at the end of "/etc/cinder/cinder.conf" file, configure the TV1_iSCSI back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder_huawei_conf_file** indicates the specified Huawei-customized configuration file.

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

In the [DEFAULT] section, enable the TV1_iSCSI back end:

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

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.1.2 Configuring the Cinder Driver for OceanStor T Series V2 (iSCSI)

Procedure

- **Step 1** In /etc/cinder, create a Huawei-customized driver configuration file. The file format is XML. Change the name of the driver configuration file based on site requirements, for example, cinder_huawei_conf.xml.
- **Step 2** Configure parameters in the driver configuration file, including mandatory and optional parameters.

Mandatory parameters:

Optional parameters(Pick and add the optional parameter in the above XML file):

NOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

```
-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
```

For details about the parameters in the configuration file, see section "4.1.6".

Step 3 Configure the **cinder.conf** file.

Add a [TV2_iSCSI] section at the end of "/etc/cinder/cinder.conf" file, configure the TV2_iSCSI back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder_huawei_conf_file** indicates the specified Huawei-customized configuration file.

```
[TV2_iSCSI]

volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiISCSIDriver

cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml

volume_backend_name = TV2_iSCSI
```

In the [DEFAULT] section, enable the TV2 iSCSI back end:

```
[DEFAULT]
...
enabled_backends=TV2_iSCSI
```

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.1.3 Configuring the Cinder Driver for OceanStor V3/V5(iSCSI)

Procedure

- **Step 1** In /etc/cinder, create a Huawei-customized driver configuration file. The file format is XML. Change the name of the driver configuration file based on site requirements, for example, cinder_huawei_conf.xml.
- **Step 2** Configure parameters in the driver configuration file, including mandatory and optional parameters.

Mandatory parameters:

```
<?xml version='1.0' encoding='UTF-8'?>
 <config>
     <Storage>
         <Product>V3</Product>
         <Protocol>iSCSI</Protocol>
         <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
         <UserName>xxx</UserName>
         <UserPassword>xxx</UserPassword>
      </Storage>
      <LUN>
         <LUNType>Thin</LUNType>
         <StoragePool>xxx</StoragePool>
      </LUN>
         <DefaultTargetIP>x.x.x.x/DefaultTargetIP>
      </iscst>
 </config>
```

Optional parameters(Pick and add the optional parameter in the above XML file):

NOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

```
-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
```

For details about the parameters in the configuration file, see section "4.1.6".

Step 3 Configure the **cinder.conf** file.

Add a [V3_iSCSI] section at the end of "/etc/cinder/cinder.conf" file, configure the V3_iSCSI back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder huawei conf file** indicates the specified Huawei-customized configuration file.

```
[V3_iSCSI]
volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiISCSIDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml
volume_backend_name = V3_iSCSI
```

In the [DEFAULT] section, enable the V3 iSCSI back end:

```
[DEFAULT]
...
enabled_backends=V3_iSCSI
```

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.1.4 Configuring the Cinder Driver for OceanStor 18000 Series(iSCSI)

This section describes how to configure the Cinder Driver for 18000 series V1 and V3 for iSCSI storage products.

4.1.4.1 Configuring the Cinder Driver for OceanStor 18000 Series V1(iSCSI)

Procedure

- **Step 1** In /etc/cinder, create a Huawei-customized driver configuration file. The file format is XML. Change the name of the driver configuration file based on site requirements, for example, cinder huawei conf.xml.
- **Step 2** Configure parameters in the driver configuration file, including mandatory and optional parameters.

Mandatory parameters:

Optional parameters(Pick and add the optional parameter in the above XML file):

NOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

-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

For details about the parameters in the configuration file, see section "4.1.6".

When configuring the access address of the REST interface, if newly installed SVP, add the 8088 port, for example, https://x.x.x.x:8088/devicemanager/rest/; if the system is update from C99, you should not add 8088 port.

Step 3 Configure the cinder.conf file.

Add a [18000V1_iSCSI] section at the end of "/etc/cinder/cinder.conf" file, configure the 18000V1_iSCSI back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder_huawei_conf_file** indicates the specified Huawei-customized configuration file.

```
[18000V1_iSCSI]

volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiISCSIDriver

cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml

volume_backend_name = 18000V1_iSCSI
```

In the [DEFAULT] section, enable the 18000V1_iSCSI back end:

```
[DEFAULT]
...
enabled backends=18000V1 iSCSI
```

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.1.4.2 Configuring the Cinder Driver for OceanStor 18000 Series V3(iSCSI)

Procedure

- **Step 1** In /etc/cinder, create a Huawei-customized driver configuration file. The file format is XML. Change the name of the driver configuration file based on site requirements, for example, cinder_huawei_conf.xml.
- **Step 2** Configure parameters in the driver configuration file, including mandatory and optional parameters.

Mandatory parameters:

Optional parameters(Pick and add the optional parameter in the above XML file):

NOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

```
-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
```

For details about the parameters in the configuration file, see section "4.1.6".

Step 3 Configure the **cinder.conf** file.

Add a [18000V3_iSCSI] section at the end of "/etc/cinder/cinder.conf" file, configure the 18000V3_iSCSI back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder_huawei_conf_file** indicates the specified Huawei-customized configuration file.

```
[18000V3_iSCSI]

volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiISCSIDriver

cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml

volume_backend_name = 18000V3_iSCSI
```

In the [DEFAULT] section, enable the 18000V3 iSCSI back end:

```
[DEFAULT]
...
enabled_backends=18000V3_iSCSI
```

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.1.5 Configuring the Cinder Driver for Dorado Series V3(iSCSI)

Procedure

- **Step 1** In /etc/cinder, create a Huawei-customized driver configuration file. The file format is XML. Change the name of the driver configuration file based on site requirements, for example, cinder huawei conf.xml.
- **Step 2** Configure parameters in the driver configuration file, including mandatory and optional parameters.

Mandatory parameters:

Optional parameters(Pick and add the optional parameter in the above XML file):

NOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

```
-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
```

For details about the parameters in the configuration file, see section "4.1.6".

Step 3 Configure the **cinder.conf** file.

Add a [Dorado_iSCSI] section at the end of "/etc/cinder/cinder.conf" file, configure the Dorado_iSCSI back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder_huawei_conf_file** indicates the specified Huawei-customized configuration file.

```
[Dorado_iSCSI]

volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiISCSIDriver

cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml

volume_backend_name = Dorado_iSCSI
```

In the [DEFAULT] section, enable the Dorado_iSCSI back end:

```
[DEFAULT]
...
enabled backends=Dorado iSCSI
```

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.1.6 Parameters in the Configuration File

Table 4-1 Mandatory parameters

Parameter	Default Value	Description	Applicable To
Product	-	Type of a storage product. Possible values are T, TV2, 18000, V3 and Dorado.	All
Protocol	-	Type of a connection protocol. The possible value is iSCSI .	All

Parameter	Default Value	Description	Applicable To
ControllerIP0	-	IP address of the primary controller on an OceanStor T series V100R005 storage device.	T series V1
ControllerIP1	-	IP address of the secondary controller on an OceanStor T series V100R005 storage device.	T series V1
RestURL	-	Access address of the REST interface, for example, https://x.x.x.x:8088/devicemanager/rest/.x.x.x.x indicates the management IP address. Refer to 4.1.4.1 to configure 18000 series V1. If you need to configure multiple RestURL, separate them by semicolons (;).	T series V2 V3 18000 Dorado V3
UserName	-	User name of a storage administrator.	All
UserPassword	-	Password of a storage administrator.	All
StoragePool	-	Name of a storage pool to be used. If you need to configure multiple storage pools, separate them by semicolons (;). Refer to 4.1.1 to configure TV1 series.	All
DefaultTargetI P	-	Default IP address of the iSCSI target port that is provided for computing nodes.	All



CAUTION

The value of **StoragePool** cannot contain Chinese characters.

Table 4-2 Optional parameters

Parameter	Default Value	Description	Applicable To
LUNType	Thick(except Dorado) Thin(Dorado)	Type of the LUNs to be created. The value can be Thick or Thin . Dorado only support Thin LUNs.	All

Parameter	Default Value	Description	Applicable To
StripUnitSize	64	Stripe depth of a LUN to be created, optional values: 4, 8,16, 32, 64, 128, 256, 512. The unit is KB. This parameter is invalid when a	T series V1
		thin LUN is created.	
WriteType	1	Cache write type. Possible values are: 1 (write back), 2 (write through).	All
Prefetch Type	3	Cache prefetch policy. Possible values are: 0 (no prefetch), 1 (fixed prefetch), 2 (variable prefetch) or 3 (intelligent prefetch).	All
Prefetch Value	0	Cache prefetch value.	All
LUNcopyWaitI nterval	5	After LUN copy is enabled, the plug-in frequently queries the copy progress. You can set a value to specify the query interval.	T series V2 V3 18000
Timeout	432000	Timeout interval for waiting LUN copy of a storage device to complete. The unit is second.	T series V2 V3 18000
Initiator Name	-	Name of a computing node initiator.	All
Initiator Target IP	-	IP address of the iSCSI target port that is provided for computing nodes.	All
Initiator TargetPortGrou p	-	IP address of the iSCSI target port that is provided for computing nodes.	T series V2 V3 18000 Dorado V3
HostIP	-	IP address of the Nova compute node's host.	T series V1
OSType	Linux	Operating system of the Nova compute node's host.	T series V1

4.1.7 Configuring iSCSI Multipathing

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

Procedure

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

```
<iscs!>
  <Initiator Name="xxxxxx" TargetPortGroup="xxxx" />
  </iscs!>
```

Step 3 If the version of OpenStack is Juno, Kilo, Liberty or Mitaka, add iscsi_use_multipath = True in [libvirt] of /etc/nova/nova.conf.

If the version of OpenStack is Newton or Ocata, add **volume_use_multipath** = **True** in [libvirt] of /etc/nova/nova.conf.

Enable the multipathing switch of the OpenStack Nova module.

Step 4 Run the **service nova-compute restart** command to restart the nova-compute service.

----End

4.1.8 Configuring CHAP and ALUA

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

Procedure

Configure CHAP and ALUA.

Configure the driver configuration file as follows:

```
<iSCSI>
     <Initiator ALUA="xxx" CHAPinfo="xxx" Name="xxx" TargetIP="x.x.x.x"/>
</iSCSI>
```

ALUA indicates a multipathing mode. **0** indicates that ALUA is disabled. **1** indicates that ALUA is enabled.

CHAPinfo indicates the user name and password authenticated by CHAP. The format is **mm-user;mm-user@storage1**. The user name and password are separated by semicolons (;).

NOTE

- For TV1 or V3 storage system ensure the status of the initiator is free.
- For V3 storage system, if this is the first time to configure chap for the initiator, ensure the initiator is not added to host or dose exist on the array.



• CHAP name must contain 4 to 223 characters.

- The password must contain 12 to 16 characters.
- The password must contain any three types of uppercase letters, lowercase letters, digits, and special characters including '~! @ # \$ % ^ & *() _ = + \ [{ }];:' " < . > / ? and spaces.
- The password cannot be the same as the account or mirror writing of the account.

4.2 Configuring the Cinder Driver for Fibre Channel Storage

This section describes how to configure Cinder Driver for different products for the Fibre Channel products.

Prerequisites

For a Fibre Channel network, the **sg** tool must be installed so that storage resources can be used.

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

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

4.2.1 Configuring the Cinder Driver for OceanStor T Series V1 (Fibre Channel)

Procedure

Step 1 In /etc/cinder, create a Huawei-customized diver configuration file. The file format is XML.

Change the name of the driver configuration file based on site requirements, for example, **cinder_huawei_conf.xml**.

Step 2 Configure parameters in the driver configuration file, including mandatory and optional parameters.

Mandatory parameters:

Optional parameters(Pick and add the optional parameter in the above XML file):

NOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

```
-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
```

For details about the parameters in the configuration file, see section "4.2.7".

Step 3 Configure the **cinder.conf** file.

Add a [TV1_FC] section at the end of "/etc/cinder/cinder.conf" file, configure the TV1_FC back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder_huawei_conf_file** indicates the specified Huawei-customized configuration file.

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

In the [DEFAULT] section, enable the TV1_FC back end:

```
[DEFAULT]
...
enabled backends=TV1 FC
```

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.2.2 Configuring the Cinder Driver for OceanStor T Series V2 (Fibre Channel)

Procedure

Step 1 In /etc/cinder, create a Huawei-customized diver configuration file. The file format is XML.

Change the name of the driver configuration file based on site requirements, for example, cinder huawei conf.xml.

Step 2 Configure parameters in the driver configuration file, including mandatory and optional parameters.

Mandatory parameters:

Optional parameters(Pick and add the optional parameter in the above XML file):

MOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

```
-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
```

For details about the parameters in the configuration file, see section "4.2.7".

Step 3 Configure the **cinder.conf** file.

Add a [TV2_FC] section at the end of "/etc/cinder/cinder.conf" file, configure the TV2_FC back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder_huawei_conf_file** indicates the specified Huawei-customized configuration file.

```
[TV2_FC]
volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiFCDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml
volume backend name = TV2 FC
```

In the [DEFAULT] section, enable the TV2 FC back end:

```
[DEFAULT]
...
enabled backends=TV2 FC
```

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.2.3 Configuring the Cinder Driver for OceanStor V3/V5 (Fibre Channel)

Procedure

Step 1 In /etc/cinder, create a Huawei-customized driver configuration file. The file format is XML.

Change the name of the driver configuration file based on site requirements, for example, **cinder_huawei_conf.xml**.

Step 2 Configure parameters in the driver configuration file, including mandatory and optional parameters.

Mandatory parameters:

Optional parameters(Pick and add the optional parameter in the above XML file):

NOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

```
-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
```

For details about the parameters in the configuration file, see section "4.2.7".

Step 3 Configure the **cinder.conf** file.

Add a [V3_FC] section at the end of "/etc/cinder/cinder.conf" file, configure the V3_FC back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder_huawei_conf_file** indicates the specified Huawei-customized configuration file.

```
[V3_FC]
volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiFCDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml
volume_backend_name = V3_FC
```

In the [DEFAULT] section, enable the V3 FC back end:

```
[DEFAULT]
...
enabled_backends=V3_FC
```

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.2.4 Configuring the Cinder Driver of OceanStor 18000 Series (Fibre Channel)

This section describes how to configure the Cinder Driver for 18000 series V1 and V3 for FC storage products.

4.2.4.1 Configuring the Cinder Driver for OceanStor 18000 Series V1(Fibre Channel)

Procedure

Step 1 In /etc/cinder, create a Huawei-customized driver configuration file. The file format is XML.

Change the name of the driver configuration file based on site requirements, for example, **cinder huawei conf.xml**.

Step 2 Configure the **cinder.conf** file, including mandatory and optional parameters.

Mandatory parameters:

Optional parameters(Pick and add the optional parameter in the above XML file):

NOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

```
-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
```

For details about the parameters in the configuration file, see section"4.2.7 ".

Step 3 Configure the **cinder.conf** file.

Add a [18000V1_FC] section at the end of "/etc/cinder/cinder.conf" file, configure the 18000V1_FC back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder_huawei_conf_file** indicates the specified Huawei-customized configuration file.

```
[18000V1_FC]

volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiFCDriver

cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml

volume_backend_name = 18000V1_FC
```

In the [DEFAULT] section, enable the 18000V1 FC back end:

```
[DEFAULT]
...
enabled_backends=18000V1_FC
```

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.2.4.2 Configuring the Cinder Driver for OceanStor 18000 Series V3(Fibre Channel)

Procedure

Step 1 In /etc/cinder, create a Huawei-customized driver configuration file. The file format is XML.

Change the name of the driver configuration file based on site requirements, for example, **cinder huawei conf.xml**.

Step 2 Configure the **cinder.conf** file, including mandatory and optional parameters.

Mandatory parameters:

Optional parameters(Pick and add the optional parameter in the above XML file):

NOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

```
-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
```

For details about the parameters in the configuration file, see section 4.2.7 ".

Step 3 Configure the **cinder.conf** file.

Add a [18000V3_FC] section at the end of "/etc/cinder/cinder.conf" file, configure the 18000V3_FC back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder_huawei_conf_file** indicates the specified Huawei-customized configuration file.

```
[18000V3_FC]

volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiFCDriver

cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml

volume_backend_name = 18000V3_FC
```

In the [DEFAULT] section, enable the 18000V3 FC back end:

```
[DEFAULT]
...
enabled_backends=18000V3_FC
```

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.2.5 Configuring the Cinder Driver for Dorado Series V3(Fibre Channel)

Procedure

- Step 1 In /etc/cinder, create a Huawei-customized driver configuration file. The file format is XML. Change the name of the driver configuration file based on site requirements, for example, cinder huawei conf.xml.
- **Step 2** Configure parameters in the driver configuration file, including mandatory and optional parameters.

Mandatory parameters:

Optional parameters(Pick and add the optional parameter in the above XML file):

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

NOTE

Make sure the owner and group of "/etc/cinder/cinder_huawei_conf.xml" file are the same as "/etc/cinder/cinder.conf" file's owner and group.

```
-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
```

For details about the parameters in the configuration file, see section "4.2.7".

Step 3 Configure the **cinder.conf** file.

Add a [Dorado_FC] section at the end of "/etc/cinder/cinder.conf" file, configure the Dorado_FC back end with the Huawei driver. The **volume_driver** indicates the loaded driver file, and **cinder_huawei_conf_file** indicates the specified Huawei-customized configuration file.

```
[Dorado_FC]

volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiFCDriver

cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml

volume_backend_name = Dorado_FC
```

In the [DEFAULT] section, enable the Dorado_FC back end:

```
[DEFAULT]
...
enabled_backends=Dorado_FC
```

Step 4 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.2.6 Configuring the Auto Zoning

Auto zoning is not compulsory, When you need to use this function, do the following configuration.

Procedure

Step 1 Configure the **cinder.conf** file.

For more details please refer to https://docs.openstack.org/ocata/config-reference/block-storage/fc-zoning.html

For example, configure the Brocade Exchanger:

```
[DEFAULT]
zoning_mode = fabric
 [fc-zone-manager]
fc fabric names = swd77
zoning policy = initiator
brcd sb connector :
cinder.zonemanager.drivers.brocade.brcd fc zone client cli.BrcdFCZoneClientCLI
fc san lookup service =
cinder.Zonemanager.drivers.brocade.brcd_fc_san_lookup_service.BrcdFCSanLookupServi
zone driver =
cinder.zonemanager.drivers.brocade.brcd fc zone driver.BrcdFCZoneDriver
[swd77]
fc fabric address = x.x.x.x
fc fabric password = xxx
fc_fabric_port = 22
fc_fabric_user = xxx
principal switch wwn = xxx
zone activate = True
```

Step 2 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

4.2.7 Parameters in the Configuration File

Table	4-3	Mandatory	y parameters
--------------	-----	-----------	--------------

Parameter	Default Value	Description	Applicable To
Product	-	Type of a storage product. Possible values are T, TV2, 18000, V3 and Dorado.	All
Protocol	-	Type of a connection protocol. Possible value is FC .	All
ControllerIP0	-	IP address of the primary controller on an OceanStor T series V100R005 storage device.	T series V1

Parameter	Default Value	Description	Applicable To
ControllerIP1	-	IP address of the secondary controller on an OceanStor T series V100R005 storage device.	T series V1
RestURL	-	Access address of the REST interface, for example, https://x.x.x.x:8088/devicemanager/rest/.x.x.x.x indicates the management IP address. Refer to 4.2.4.1 to configure 18000 series V1. If you need to configure multiple RestURL, separate them by semicolons (;).	T series V2 V3 18000 Dorado V3
UserName	-	User name of an administrator.	All
UserPassword	-	Password of an administrator.	All
StoragePool	-	Name of a storage pool to be used. If you need to configure multiple storage pools, separate them by semicolons (;). Refer to 4.1.1 to configure TV1 series.	All

Table 4-4 Optional parameters

Parameter	Default Value	Description	Applicable To
LUNType	Thick(except Dorado) Thin(Dorado)	Type of the LUNs to be created. The value can be Thick or Thin . Dorado only support Thin LUNs.	All
StripUnitSize	64	Stripe depth of a LUN to be created, optional values: 4, 8,16, 32, 64, 128, 256, 512. The unit is KB. This parameter is invalid when a thin LUN is created.	T series V1
WriteType	1	Cache write type. Possible values are: 1 (write back), 2 (write through).	All
Prefetch Type	3	Cache prefetch policy. Possible values are: 0 (no prefetch), 1 (fixed prefetch), 2 (variable prefetch) or 3 (intelligent prefetch).	All
Prefetch Value	0	Cache prefetch value.	All

Parameter	Default Value	Description	Applicable To
LUNcopyWaitI nterval	5	After LUN copy is enabled, the plug-in frequently queries the copy progress. You can set a value to specify the query interval.	T series V2 V3 18000
Timeout	432000	Timeout interval for waiting LUN copy of a storage device to complete. The unit is second.	T series V2 V3 18000
HostIP	-	IP address of the Nova compute node's host.	T series V1
OSType	Linux	Operating system of the Nova compute node's host.	T series V1

4.2.8 Configuring FC Multipathing

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

Procedure

Step 1 Enable the multipathing switch of the OpenStack Nova module.

If the version of OpenStack is Liberty or Mitaka, add **iscsi_use_multipath = True** in **[libvirt]** of /etc/nova/nova.conf.

If the version of OpenStack is Newton or Ocata, add **volume_use_multipath** = **True** in **[libvirt]** of /etc/nova/nova.conf.

Step 2 Run the service nova-compute restart command to restart the nova-compute service.

----End

4.3 Configuring Multi-Storage Support

Example for configuring multiple storage systems in /etc/cinder/cinder.conf:

```
[DEFAULT]
...
enabled_backends = t_fc, 18000_fc
[t_fc]
volume_driver = cinder.volume.drivers.huawei.huawei_t.HuaweiTFCDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf_t_fc.xml
volume_backend_name = t_fc
[18000_fc]
volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiFCDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf_18000_fc.xml
volume_backend_name = 18000_fc
```

5 Cinder Driver Advanced Properties Configuration

Huawei storage supports advanced properties, such as Smartx and HyperMetro. By associating with specified volume types in OpenStack, these properties make the combination of OpenStack and storage advanced properties possible.

- 5.1 Configuring the SmartQoS Property
- **5.2 Configuring the SmartPartition Property**
- **5.3 Configuring the Thick Property**
- **5.4 Configuring the Thin Property**
- 5.5 Configuring the SmartCache Property
- **5.6 Configuring the SmartTier Property**
- 5.7 Configuring the HyperMetro Property
- **5.8 Configuring the Replication V2.1**
- 5.9 Create a volume on a certain storage pool
- 5.10 Create a volume with a certain disk type
- 5.11 Create a volume on a certain volume back end
- **5.12 Configuring Consistency Groups**
- 5.13 Configuring Backup Snapshot
- **5.14 Configuring Certificate Verify**
- 5.1 Configuring the SmartQoS Property
- 5.2 Configuring the SmartPartition Property
- 5.3 Configuring the Thick Property
- 5.4 Configuring the Thin Property
- 5.5 Configuring the SmartCache Property
- 5.6 Configuring the SmartTier Property

- 5.7 Configuring the HyperMetro Property
- 5.8 Configuring the Replication V2.1
- 5.9 Create a volume on a certain storage pool
- 5.10 Create a volume with a certain disk type
- 5.11 Create a volume on a certain volume back end
- 5.12 Configuring Consistency Groups
- 5.13 Configuring Backup Snapshot
- 5.14 Configuring Certificate Verify

5.1 Configuring the SmartQoS Property

About This Chapter

Qos in OpenStack mainly depends on the front-end Hypervisor and the back-end storage. Huawei OpenStack Cinder Driver supports "frontend QoS" and "backend QoS". Front-end QoS options are:

total_bytes_sec, read_bytes_sec, write_bytes_sec, total_iops_sec, read_iops_sec, write iops_sec

Huawei storage backends support the following QoS properties. One or multiple properties can be associated with one QoS property simultaneously.

Protection policies: latency, minIOPS, minBandWidth

Restriction policies: maxIOPS, maxBandWidth



CAUTION

Protection policies and Restriction policies are mutually exclusive. If they are configured together, volumes will fail to be created.

"IOType" is mandatory. If it isn't configured in qos, volumes will fail to be created.

5.1.1 Configuring the Front-end QoS

This section describes how to configure the front-end qos.

Procedure

Step 1 Run the cinder type-create XXX command to create a volume type. XXX indicates the name of a volume type.

```
| 84b189d3-8984-4e92-aab0-fa4a913126bf | high-iops | - | True |
```

Step 2 Run the cinder qos-create xxx consumer="front-end" read_iops_sec=xxx write_iops_sec=xxx command to create front QoS control property parameters.

read iops sec=2000 write iops sec=1000 is used as an example.

```
root@ubuntu:/# cinder qos-create high-iops consumer="front-end"
read_iops_sec=2000 write_iops_sec=1000
+-----+
| Property | Value |
+-----+
| consumer | front-end |
| id | 7dc73b5b-1b19-4371-8dee-8edbc52cf625 |
| name | high-iops |
| specs | {'write_iops_sec': '1000', 'read_iops_sec': '2000'} |
+------+
```

Step 3 Associate the volume type with QoS control properties.

1. Check the usage of the **cinder qos-associate qos** command.

The association command format is **cinder qos-associate** <ID of QoS specifications> <ID of volume type>.

root@ubuntu:/# cinder qos-associate

usage: cinder qos-associate <qos_specs> <volume_type_id>

try 'cinder help qos-associate' for more information.

2. Run the **cinder qos-associate** 7dc73b5b-1b19-4371-8dee-8edbc52cf625 84b189d3-8984-4e92-aab0-fa4a913126bf_o

----End

Results

• Run the **cinder qos-list** command to view details about the QoS policy configuration.

MOTE

If consumer="front-end" is configured in qos, front-end QoS will be created; if not, back-end QoS will be created.

5.1.2 QoS Configuration Versions

Table 5-1 Support for configurations of storage systems with multiple QoS properties

Storage Product Model	Storage System Version
18000	V300R003C00
	V300R006C00

Storage Product Model	Storage System Version
V3	V300R003C00/C10/C20
	V300R006C00
	2200 V300R005C00
	2600 V300R005C00
	2200 V300R006C00
	2600 V300R006C00

5.1.3 Configuring the Control IOPS

Configuring the control IOPS involves configuring the maximum control IOPS and minimum control IOPS.

5.1.3.1 Configuring the Maximum Control IOPS

This section describes how to configure the maximum Control IOPS.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

- Step 2 Run the cinder type-key maxIOPS set capabilities:QoS_support='<is> true' command to set a key-value pair for the maxIOPS volume type.
- **Step 3** Run the **cinder qos-create xxx maxIOPS=***xxx* **IOType=***xxx* command to create QoS control property parameters.
 - The first **xxx** indicates the created QoS control properties name.
 - maxIOPS: indicates the maximum IOPS. The value is an integer larger than 0.
 - **IOType**: indicates the read and write type.0 indicates the control read I/Os, 1 indicates the control write I/Os, 2 indicates the control read and write I/Os.

maxIOPS=100 IOType=2 is used as an example.

```
root@ubuntu:/# cinder qos-create maxiops maxIOPS=100 IOType=2
+------+
| property | value |
+------+
| consumer | back-end |
| id | 1f772258-49f0-47a9-aa9c-d8f32d844bb1 |
| name | maxiops |
| specs | {u'IOType': u'2', u'maxIOPS': u'100'} |
+-------+
```

Step 4 Associate the volume type with QoS control properties.

1. Check the usage of the **cinder qos-associate qos** command.

The association command format is **cinder qos-associate** <ID of QoS specifications> <ID of volume type>.

root@ubuntu:/# cinder qos-associate

usage: cinder qos-associate <qos specs> <volume type id>

try 'cinder help qos-associate' for more information.

2. Run the **cinder qos-associate** 1f772258-49f0-47a9-aa9c-d8f32d844bb1 3d9cc52e-069b-4245-b201-945e0ef571cf.

----End

5.1.3.2 Configuring the Minimum Control IOPS

This section describes how to configure the minimum Control IOPS.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

- Step 2 Run the cinder type-key minIOPS set capabilities:QoS_support='<is> true' command to set a key-value pair for the minIOPS volume type.
- Step 3 Run the cinder qos-create xxx minIOPS=xxx IOType=xxx command to create QoS control property parameters.
 - The first **xxx** indicates the created QoS control properties name.
 - minIOPS: indicates the minimum IOPS. The value is an integer larger than 0.
 - **IOType**: indicates the read and write type.0 indicates the control read I/Os, 1 indicates the control write I/Os, 2 indicates the control read and write I/Os.

minIOPS=100 IOType=2 is used as an example.

- **Step 4** Associate the volume type with QoS control properties.
 - 1. Check the usage of the **cinder gos-associate gos** command.

The association command format is **cinder qos-associate** <ID of QoS specifications> <ID of volume type>.

root@ubuntu:/# cinder qos-associate

usage: cinder qos-associate <qos specs> <volume type id>

try 'cinder help qos-associate' for more information.

2. Run the **cinder qos-associate** 8deaf13c-bf1c-4a71-94d0-1149805693ce 1a1dcee8-d9ae-4de9-b1c6-2317a4ad7219.

----End

5.1.4 Configuring the Control Bandwidth

Configuring the control bandwidth involves configuring the maximum control bandwidth and minimum control bandwidth.

5.1.4.1 Configuring the Maximum Control Bandwidth

This section describes how to configure the maximum control bandwidth.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

- Step 2 Run the cinder type-key maxBandWidth set capabilities:QoS_support='<is> true' command to set a key-value pair for the maxBandWidth volume type.
- **Step 3** Run the **cinder qos-create xxx maxBandWidth=***xxx* **IOType=***xxx* command to create QoS control property parameters.
 - The first **xxx** indicates the created QoS control properties name.
 - maxBandWidth: indicates the maximum BANDWIDTH. The value is an integer larger than 0 and expressed in MB/s.
 - **IOType**: indicates the read and write type.0 indicates the control read I/Os, 1 indicates the control write I/Os, 2 indicates the control read and write I/Os.

maxBandWidth=100 IOType=2 is used as an example.

- **Step 4** Associate the volume type with QoS control properties.
 - 1. Check the usage of the **cinder qos-associate qos** command.

The association command format is **cinder qos-associate** <ID of QoS specifications> <ID of volume type>.

root@ubuntu:/# cinder qos-associate
usage: cinder qos-associate <qos_specs> <volume_type_id>
try 'cinder help qos-associate' for more information.

2. Run the **cinder qos-associate** 2f51c955-e029-48ca-aa8c-5d7ae462481e 24200fbb-c984-4d99-9465-a820464662d6 command.

----End

5.1.4.2 Configuring the Minimum Control Bandwidth

This section describes how to configure the minimum control bandwidth.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

- Step 2 Run the cinder type-key minBandWidth set capabilities:QoS_support='<is> true' command to set a key-value pair for the minBandWidth volume type.
- **Step 3** Run the **cinder qos-create xxx minBandWidth=***xxx* **IOType=***xxx* command to create QoS control property parameters.
 - The first **xxx** indicates the created QoS control properties name.
 - **minBandWidth**: indicates the minimum BANDWIDTH. The value is an integer larger than **0** and expressed in MB/s.
 - IOType: indicates the read and write type.0 indicates the control read I/Os, 1 indicates the control write I/Os, 2 indicates the control read and write I/Os.

minBandWidth=100 IOType=2 is used as an example.

```
root@ubuntu:/# cinder qos-create minbandwidth minBandWidth=100 IOType=2
+------+
| property | value |
+------+
| consumer | back-end |
| id | 59c583d8-69d2-4c39-8c75-c1b21ef85f2e |
| name | minbandwidth |
| specs | {u'IOType': u'2', u'minBandWidth': u'100'} |
+------+
```

- **Step 4** Associate the volume type with QoS control properties.
 - 1. Check the usage of the **cinder qos-associate qos** command.

The association command format is **cinder qos-associate** <ID of QoS specifications> <ID of volume type>.

root@ubuntu:/# cinder gos-associate

usage: cinder qos-associate <qos specs> <volume type id>

try 'cinder help qos-associate' for more information.

2. Run the **cinder qos-associate** 59c583d8-69d2-4c39-8c75-c1b21ef85f2e 271bed0a-1cce-4e8f-a65e-b85ccb3b9a25.

----End

5.1.5 Configuring the Control Latency

This section describes how to configure the control latency.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

- Step 2 Run the cinder type-key latency set capabilities:QoS_support='<is> true' command to set a key-value pair for the latency volume type.
- **Step 3** Run the **cinder qos-create xxx latency**=*xxx* **IOType**=*xxx* command to create QoS control property parameters.
 - The first **xxx** indicates the created QoS control properties name.
 - latency: indicates LATENCY. The value is an integer larger than 0 and expressed in ms.
 - IOType: indicates the read and write type.0 indicates the control read I/Os, 1 indicates the control write I/Os, 2 indicates the control read and write I/Os.

latency=100 **IOType**=2 is used as an example.

```
root@ubuntu:/# cinder qos-create latency latency=100 IOType=2
+-------+
| property | value |
+------+
| consumer | back-end |
| id | 6f305e9f-698b-4e9e-aa68-8efb80b43036 |
| name | latency |
| specs | {u'latency': u'100', u'IOType': u'2'} |
+-------+
```

- **Step 4** Associate the volume type with QoS control properties.
 - 1. Check the usage of the **cinder qos-associate qos** command.

The association command format is **cinder qos-associate** <ID of QoS specifications> <ID of volume type>.

root@ubuntu:/# cinder qos-associate

usage: cinder qos-associate <qos_specs> <volume_type_id>

try 'cinder help qos-associate' for more information.

2. Run the **cinder qos-associate** 6f305e9f-698b-4e9e-aa68-8efb80b43036 ae34870a-21c0-4e50-8148-1e26f6ab6eab command.

----End

5.1.6 Configuring Multiple Control Policies

This section describes how to configure multiple control policies.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

- Step 2 Run the cinder type-key multi-strategy set capabilities:QoS_support='<is> true' command to set a key-value pair for the multi-strategy volume type.
- Step 3 Run the cinder qos-create xxx latency=xxx minBandWidth=xxx IOType=xxx command to create QoS control property parameters.
 - The first **xxx** indicates the created QoS control properties name.
 - latency: indicates LATENCY. The value is an integer larger than **0** and expressed in ms.
 - **minBandWidth**: indicates the minimum BANDWIDTH. The value is an integer larger than **0** and expressed in MB/s.
 - IOType: indicates the read and write type.0 indicates the control read I/Os, 1 indicates the control write I/Os, 2 indicates the control read and write I/Os.

latency=100 minBandWidth=100 IOType=2 is used as an example.

- **Step 4** Associate the volume type with QoS control properties.
 - 1. Check the usage of the **cinder gos-associate gos** command.

The association command format is **cinder qos-associate** <ID of QoS specifications> <ID of volume type>.

root@ubuntu:/# cinder qos-associate

usage: cinder qos-associate <qos specs> <volume type id>

try 'cinder help gos-associate' for more information.

2. Run the **cinder qos-associate** 36f0fc4a-d5f5-4862-ab1f-e7fedeeddc41 df53d9d7-b1db-4e6b-847a-7a5150c39489.

----End

Results

• Run the **cinder qos-list** command to view details about the QoS policy configuration.

```
root@ubuntu:~# cinder qos-list
                                                 +----
| 36f0fc4a-d5f5-4862-ab1f-e7fedeeddc41 | multi-strategy | back-end |
{'IOType': '2', 'minBandWidth': '100', 'latenct': '100'} |
| 3e2b2fa5-aebb-4ea0-8dd2-52a577015769 | miniops | back-end
           {'IOType': '2', 'minIOPS': '50'}
| 3fd0971e-f5fd-48d7-89d9-d2efb6fc44cf | maxbandwidth | back-end
  {'maxBandWidth': '100', 'IOType': '2'}
| 6a4e24af-0e8c-4a64-91ee-859434414400 | latency
| {'latency': '100', 'IOType': '2'}
| 9b34fbdf-91b5-473d-a5ff-0b4ea2ed9fd0 | maxiops | back-end
{'IOType': '2', 'maxIOPS': '100'}
        {'IOType': '2', 'maxIOPS': '100'}
                                                 | f60a3374-cdf3-45cc-86c4-12c06a3db8d4 | minbandwidth | back-end
{'IOType': '2', 'minBandWidth': '100'}
+-----
```

• Run the **cinder type-list** command to view the volume type.

5.2 Configuring the SmartPartition Property

The following example describes how to configure the SmartPartition property.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

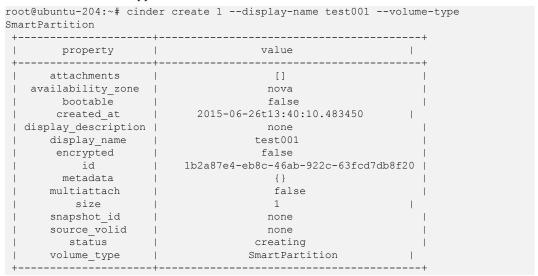
```
root@ubuntu-204:~# cinder type-create SmartPartition
+------
+-----+
| ID | Name | Description | Is_Public |
+------+
| a7648f77-fb56-41f2-9d50-3c014534ef66 | SmartPartition | - | True
```

+-----+

- Step 2 Run the cinder type-key SmartPartition set capabilities:smartpartition='<is> true' command to set a key-value pair for the SmartPartition volume type.
- Step 3 Run the cinder type-key SmartPartition set smartpartition:partitionname ='test partition'

partitionname indicates the name of a SmartPartition partition that has been configured on the storage system.

Step 4 Create a volume that supports **SmartPartition**.



----End

5.3 Configuring the Thick Property

The following example describes how to configure the Thick property.



CAUTION

Dorado V3 dones't support Thick volume.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

Step 2 Run the cinder type-key Thick set capabilities:thick_provisioning_support='<is> true' command to set a key-value pair for the Thick volume type.

- **Step 3** Run the **cinder type-key Thick set provisioning:type='thick'** command to set the scheduling mechanism for the Thick volume.
- **Step 4** Create a volume that supports **Thick**.

5.4 Configuring the Thin Property

The following example describes how to configure the Thin property.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

```
root@ubuntu-204:~# cinder type-create Thin

+------+

| ID | Name | Description | Is_Public |

+-----+

| ba648f77-fb56-41f2-9d50-3c014534eedd | Thin | - | True |

+-----+
```

- Step 2 Run the cinder type-key Thin set capabilities:thin_provisioning_support='<is> true' command to set key-value pairs of the Thin volume type.
- **Step 3** Create a volume that supports **Thin**.

	status		creating		
- 1	volume_type	1	Thin		
+				+	

5.5 Configuring the SmartCache Property

The following example describes how to configure the SmartCache property.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

- Step 2 Run the cinder type-key SmartCache set capabilities:smartcache='<is> true' command to set a key-value pair for the SmartCache volume type.
- Step 3 Run the cinder type-key SmartCache set smartcache:cachename='test_cache' command to config smartcache name.

cachename is the name of SmartCache partition that has been configured on the storage system.

Step 4 Create a volume that supports **SmartCache**.

----End

5.6 Configuring the SmartTier Property

The following example describes how to configure the SmartTier property.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

- Step 2 Run the cinder type-key SmartTier set capabilities:smarttier='<is> true' command to set a key-value pair for the SmartTier volume type.
- Step 3 Run the cinder type-key SmartTier set smarttier:policy= 'xx' command to set a key-value pair for the SmartTier volume type.

Policy: indicates migration policy. **0** indicates no migration, **1** indicates automatic migration, **2** indicates migration to high performance, **3** indicates migration to low performance.

Step 4 Create a volume that supports **SmartTier**.

----End

5.7 Configuring the HyperMetro Property

The following example describes how to configure the HyperMetro property.

Prerequisites

Step 1 Add information about remote devices in /etc/cinder/cinder.conf in target back end section.

```
hypermetro_device =
   storage_pool:StoragePool001,
   san_address:https://IP:port/deviceManager/rest/,
   san_user:admin,
   san_password:XXXX,
   iscsi_default_target_ip:x.x.x.x,
   metro_domain:hypermetro-domain
```

Add information about remote devices for iscsi multipath:



Separate the key words by ","; separate the initiator informations in "iscsi_info" by ";".

Table 5-2 Parameter specification

Parameter	Default Value	Description	Applicable To
metro_san_address		Access address of the REST interface, for example, https:// x.x.x.x:8088/ deviceManager/rest/, x.x.x.x indicates the management IP address.	V3R3 2600 V3R5 18000 V3R3
metro_san_user		User name of a storage administrator of hypermetro remote device.	V3R3 2600 V3R5 18000 V3R3
metro_san_passwor		Password of a storage administrator of hypermetro remote device.	V3R3 2600 V3R5 18000 V3R3
metro_domain_nam e		Hypermetro domain name configured on DeviceManager.	V3R3 2600 V3R5 18000 V3R3
metro_storage_pools		Remote storage pool for hypermetro.	V3R3 2600 V3R5 18000 V3R3
iscsi_default_target_ ip		Remote transaction port IP	V3R3 2600 V3R5 18000 V3R3
Initiator Name	-	Name of a computing node initiator.	V3R3 2600 V3R5 18000 V3R3

Parameter	Default Value	Description	Applicable To
Initiator Target IP	-	IP address of the iSCSI target port that is provided for computing nodes.	V3R3 2600 V3R5 18000 V3R3
Initiator TargetPortGroup	-	IP address of the iSCSI target port that is provided for computing nodes.	V3R3 2600 V3R5 18000 V3R3

Step 2 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

root@ubuntu-204:~# cinder type-create HyperMetro					
ID Name Description Is_Public					
0913e3dd-8b68-4b46-80ea-cf255617b13a	HyperMetro		True		

- Step 2 Run the cinder type-key HyperMetro set capabilities:hypermetro='<is> true' command to set a key-value pair for the HyperMetro volume type.
- **Step 3** Create a volume that supports **HyperMetro**.

----End

5.8 Configuring the Replication V2.1

The following example describes how to configure the Replication V2.1.

Prerequisites

Step 1 Add information about remote devices in /etc/cinder/cinder.conf in target back end section.

```
replication_device =
   backend_id:huawei-replica-1,
   storage_pool:StoragePool001,
   san_address:https://IP:port/deviceManager/rest/,
   san_user:admin,
   san_password:XXXX,
   iscsi_default_target_ip:x.x.x.x
```

Add information about remote devices for iscsi multipath:



Separate the key words by ","; separate the initiator informations in "iscsi_info" by ";".

Table 5-3 Parameter specification

Parameter	Default Value	Description	Applicable To
backend_id		Target device id	TV2、V3、18000、 Dorado V3
storage_pool		Pool name of target backend when failover.	TV2、V3、18000、 Dorado V3
san_address		Access address of the REST interface, for example, https://x.x.x.x:8088/deviceManager/rest/,x.x.x.x indicates the management IP address. Refer to 4.1.4.1 or 4.2.4.1 to configure 18000 serises V1.	TV2、V3、18000、 Dorado V3
san_user		User name of a storage administrator of Replication V2.1 remote device	TV2、V3、18000、 Dorado V3

Parameter	Default Value	Description	Applicable To
san_password		Password of a storage administrator of Replication V2.1 remote device.	TV2、V3、18000、 Dorado V3
iscsi_default_target_ip		Remote transaction port IP	TV2、V3、18000、 Dorado V3
Initiator Name	-	Name of a computing node initiator.	TV2、V3、18000、 Dorado V3
Initiator Target IP	-	IP address of the iSCSI target port that is provided for computing nodes.	TV2、V3、18000、 Dorado V3
Initiator TargetPortGroup	-	IP address of the iSCSI target port that is provided for computing nodes.	TV2、V3、18000、 Dorado V3

Step 2 Run the **service cinder-volume restart** command to restart the Cinder service.

----End

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

```
root@ubuntu-204:~# cinder type-create Replication_V2.1
+-------+
| ID | Name | Description |
Is_Public |
+------+
| b9ff6a0d-7581-4e50-8ba9-0ab2703e42e7 | Replication_V2.1 | - |
True |
+-----+
```

- Step 2 Run the cinder type-key Replication_V2.1 set capabilities:replication_enabled='<is> true' command to set a key-value pair for the Replication V2.1 volume type.
- Step 3 (Optional)Run the cinder type-key Replication_V2.1 set replication_type='<in> sync' command to set the type of Replication V2.1.

Optional values of "replication_type" are "sync" and "async". "sync" indicates synchronous replication; "async" indicates asynchronous replication. when skip this step, asynchronous replication will be applied.

Step 4 Create a volume that supports **Replication V2.1.**

root@ubuntu-204:~# cinder create 1 --display-name test001 --volume-type Replication V2.1

++		-+
property	value	į
attachments	[]	
availability zone	nova	- 1
bootable	false	-
created at	2015-06-26t13:42:10.483450	-
display description	none	- 1
display name	test001	- [
encrypted	false	- 1
id	1b2a87e4-eb8c-46ab-922c-63fcd7db8f31	- [
metadata	{}	-
multiattach	false	-
size	1	-
snapshot_id	none	-
source_volid	none	
status	creating	-
volume_type	Replication_V2.1	-
++		-+

5.9 Create a volume on a certain storage pool

This section describes how to create a volume on a certain storage pool in a back end that manages multiple pools. A volume type with a extra spec specified storage pool should be created first, then the user can use this volume type to create the volume.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

- **Step 2** Run the following command to configure the storage pool.
 - Configure a single storge pool
 cinder type-key target-pool set pool_name=StoragePool001
 - Configure multiple storage pools. cinder type-key target-pool set pool_name="<or> StoragePool001 <or> StoragePool002"
- **Step 3** Create a volume on a certain **storage pool**.

multiattach	False	
name	None	
os-vol-host-attr:host	None	1
os-vol-mig-status-attr:migstat	None	
os-vol-mig-status-attr:name id	None	
os-vol-tenant-attr:tenant_id	53c46df66a084916876a08d7bcc31d87	1
replication_status	disabled	
size	1	1
snapshot_id	None	
source_volid	None	
status	creating	
updated at	None	
user id	ec5b1bd5fa8646109381755037aacc82	
volume type	target-pool	
_ 	+	+

5.10 Create a volume with a certain disk type

This section describes how to create a volume with a certain disk type.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

- **Step 2** Run the following command to configure the certain disk type.
 - Configure one type of disk type.
 root@ubuntu-175:~# cinder type-key disk-type set disk_type=sas
 - Configure multiple types of disk type.

 root@ubuntu-175:~# cinder type-key disk-type set disk_type="<or> sas <or> ssd"

optional values of disk type can be "ssd", "sas", "nl_sas" or "mix", the "mix" is a mixture of two or more of "ssd", "sas", "nl_sas".

Step 3 Create a volume with a certain **disk type**.

1	replication_status	disabled	
1	size	1	
1	snapshot_id	None	1
1	source_volid	None	1
1	status	creating	1
	updated_at	2016-08-12T08:24:11.000000	
1	user_id	ec5b1bd5fa8646109381755037aacc82	1
	volume_type	disk-type	1
+		+	+

5.11 Create a volume on a certain volume back end

This section describes how to create a volume on a certain volume back end.

Procedure

Step 1 Run the **cinder type-create XXX** command to create a volume type. XXX indicates the name of a volume type.

Step 2 Configure the volume back end name.

Step 3 Create a volume on the volume back end.

```
root@u1404:~# cinder create --volume-type volume-backend 1
                                                                                 _____
             -----+
                                        attachments
                                                                                                                                                                                                                                    []
                                availability_zone
                                                                                                                                                                                                                                  nova
                             bootable | None | Created_at | 2017-04-17T19:36:11.000000 | None 
                                               bootable
                                                                                                                                                                                                                               false
                                              description encrypted
                                                                                                                                                                                                                             False
                                                                                                                                          | 01d1df8c-45f8-4af3-8ef6-62a708233474 |
                                                               id
                                                     metadata
                                                                                                                                                                                                                                      { }
                                     migration_status
multiattach
                                                                                                                                                                                                                                 None
                                         multiattach
                                                                                                                                                                                                                          False
                                                                                                                                                                                                                                  None
                                                              name
os-vol-host-attr:host | u1404@240_sh#Cinder-HXL
```

```
| os-vol-mig-status-attr:migstat | None | Os-vol-mig-status-attr:name_id | None | Os-vol-mig-status-attr:name_id | None | Os-vol-tenant-attr:tenant_id | 3692e4d8455741a8b0c47a6859f23e1c | Os-vol-tenant-attr:tenant_id | disabled | Os-vol-tenant-attr:tenant_id | disabled | Os-vol-tenant-attr:tenant_id | Disabled | Os-vol-tenant-attr:tenant_id | None | Os-vol-tenant-attr:tenant_id | Os-vol-tenant-attr:
```

5.12 Configuring Consistency Groups

This section describes how to configure consistency groups and snapshot consistency groups.

Before using consistency groups, you must change policies for the consistency group APIs in the /etc/cinder/policy.json file. For more details please refer to http://docs.openstack.org/admin-guide/blockstorage-consistency-groups.html.

5.13 Configuring Backup Snapshot

This section describes how to configure backup snapshot.

Procedure

Step 1 Configure backup driver.

For more details please refer to https://docs.openstack.org/ocata/config-reference/block-storage/backup-drivers.html, and ensure cinder-backup service is working.

Step 2 Configure the **cinder.conf**file.

In the [DEFAULT] section add the following configuration.

```
backup_use_same_host = True
```

Add the following configuration in the target back end configuration group

```
backup use temp snapshot = True
```

Step 3 Run the **service cinder-volume restart** and **service cinder-back restart** command to restart the Cinder service and Backup service.

----End

5.14 Configuring Certificate Verify

This section describes how to enable SSL certificate verify for Huawei storage connection.

Procedure

Step 1 Configure Huawei-defined configuration file of Driver.

Add SSLCertVerify and SSLCertPath in <Storage>, as follow:

SSLCertVerify denotes whether enable certificate verify, which's valid options are True/False. Default False if not explicitly specified.

SSLCertPath specifies the path to the certificate to use, only works when SSLCertVerify is True.

Step 2 Restart cinder-volume service.

----End

6 Best Practices

6.1 Quick Interconnection with Huawei Storage

6.1 Quick Interconnection with Huawei Storage

6.1 Quick Interconnection with Huawei Storage

Configuration Process

This section demonstrates how to configure OpenStack Huawei Cinder Driver on OpenStack to interconnect with Huawei Storage.

Step 1 Obtain Cinder Driver.

OpenStack Kilo and later versions released in the OpenStack community are delivered with Huawei OpenStack Driver. You can obtain the latest codes of Driver through Huawei's OpenStack Driver repository. For details, see 3.

NOTE

- A community version cannot be integrated with new features once being released and has security risks because its debug process is time-consuming.
- However, OpenStack Driver obtained from OpenStack Driver repository can be integrated with newly launched features and debugged in a timely manner, when a supportive framework is used.
- The community only maintains two stable versions. However, Huawei OpenStack Driver library maintains six stable versions, ensuring long-term stable running of historical versions.
- You are strongly recommended to use Huawei OpenStack Driver library versions, instead of community versions.
- **Step 2** Configure file **cinder.conf** and Huawei-defined configuration file of Driver. (The following uses Huawei OceanStor V3 series on the iSCSI network as an example. For details, see 4).
 - In /etc/cinder, create a Huawei-defined Driver configuration file in .xml format. In this
 example, cinder_huawei_conf.xml is used as the file name that can be changed based
 on actual conditions.
 - 2. Set parameters for the created file.

3. Check the owner and owning group of the file.

Ensure that the owner and owning group of file /etc/cinder/cinder_huawei_conf.xml is the same as those of file /etc/cinder/cinder.conf.

```
-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
```

4. Configure file **cinder.conf**.

At the end of file /etc/cinder/cinder.conf, add the following configuration item to add HuaweiDriver for V3_iSCSI. In this configuration item, volume_driver indicates the loaded Driver file, and cinder_huawei_conf_file indicates the Huawei-defined configuration file.

```
[V3_iSCSI]
volume_driver = cinder.volume.drivers.huawei.huawei_driver.HuaweiISCSIDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml
volume backend name = V3 iSCSI
```

In the **[DEFAULT]** area, modify the configuration as follows to enable the V3_iSCSI back-end:

```
[DEFAULT]
...
enabled_backends=V3_iSCSI
```

- **Step 3** Run theservice cinder-volume restart command to restart the Cinder service.
- **Step 4** Check the service status.

In this example, the service status of **u1404@V3_iSCSI** is **up**, indicating that the service is started correctly.

----End

 7_{FAQ}

- 7.1 Need to manually create the mapping associated objects like Host, Host Group or Lun Group on Huawei storage beforehand before attaching volume via Cinder Driver?
- 7.2 Nova log prints "Isblk " command execution failed while attaching volume.

7.1 Need to manually create the mapping associated objects like Host, Host Group or Lun Group on Huawei storage beforehand before attaching volume via Cinder Driver?

No, because Cinder Driver will create these mapping associated objects and the Mapping View. On the contrary, if these objects are manually created on Huawei storage, will cause Cinder Driver not properly function, so that please make sure these objects deleted before attaching volume.

7.2 Nova log prints "lsblk " command execution failed while attaching volume.

Nova will utilize "lsblk" tool to query the attached volume information, this error generally occurs due to this tool is not installed, please make sure it's installed correctly in Nova system.