

OpenStack Cinder Driver Configuration Guide

Issue 01

Date 2016-08-26



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	I
2 Version Mappings	2
3 Cinder Driver Installation&deployment	10
3.1 Obtain Cinder Driver.	
3.2 Ubuntu Environment Deployment.	11
3.3 RedHat OpenStack Deployment	12
4 Cinder Driver Basis Properties Configuration	13
4.1 Configuring the Cinder Driver for iSCSI Storage	14
4.1.1 Configuring the Cinder Driver for OceanStor T Series V1 (iSCSI)	14
4.1.2 Configuring the Cinder Driver for OceanStor T Series V2 (iSCSI)	15
4.1.3 Configuring the Cinder Driver for OceanStor V3 (iSCSI)	16
4.1.4 Configuring the Cinder Driver for OceanStor 18000 Series(iSCSI)	17
4.1.4.1 Configuring the Cinder Driver for OceanStor 18000 Series V1(iSCSI)	17
4.1.4.2 Configuring the Cinder Driver for OceanStor 18000 Series V3(iSCSI)	18
4.1.5 Configuring the Cinder Driver for Dorado Series V3(iSCSI)	19
4.1.6 Parameters in the Configuration File.	20
4.1.7 Configuring iSCSI Multipathing	23
4.1.8 Configuring CHAP and ALUA	24
4.2 Configuring the Cinder Driver for Fibre Channel Storage	24
4.2.1 Configuring the Cinder Driver for OceanStor T Series V1 (Fibre Channel)	25
4.2.2 Configuring the Cinder Driver for OceanStor T Series V2 (Fibre Channel)	26
4.2.3 Configuring the Cinder Driver for OceanStor V3 (Fibre Channel)	27
4.2.4 Configuring the Cinder Driver of OceanStor 18000 Series (Fibre Channel)	28
4.2.4.1 Configuring the Cinder Driver for OceanStor 18000 Series V1(Fibre Channel)	28
4.2.4.2 Configuring the Cinder Driver for OceanStor 18000 Series V3(Fibre Channel)	29
4.2.5 Configuring the Cinder Driver for Dorado Series V3(Fibre Channel)	30
4.2.6 Configuring the Auto Zoning.	31
4.2.7 Parameters in the Configuration File	32
4.2.8 Configuring FC Multipathing.	
4.3 Configuring Multi-Storage Support	34
5 Cinder Driver Advanced Properties Configuration	36
5.1 Configuring the SmartQoS Property	37

5.1.1 Configuring the Front-end QoS	37
5.1.2 QoS Configuration Versions.	
5.1.3 Configuring the Control IOPS	38
5.1.3.1 Configuring the Maximum Control IOPS	38
5.1.3.2 Configuring the Minimum Control IOPS	39
5.1.4 Configuring the Control Bandwidth	40
5.1.4.1 Configuring the Maximum Control Bandwidth	40
5.1.4.2 Configuring the Minimum Control Bandwidth	41
5.1.5 Configuring the Control Latency	42
5.1.6 Configuring Multiple Control Policies	43
5.2 Configuring the SmartPartition Property	45
5.3 Configuring the Thick Property	46
5.4 Configuring the Thin Property	47
5.5 Configuring the SmartCache Property	47
5.6 Configuring the SmartTier Property	48
5.7 Configuring the HyperMetro Property	49
5.8 Configuring the Replication V2.1	51
5.9 Configuring Assign StoragePool.	53
5.10 Configuring Assign DiskType	54
5.11 Configuring Consistency Groups	55
5 12 Configuring Backup Spanshot	55

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. (\checkmark : support, x: unsupport)

Huawei storage system	IceHouse	Juno	Kilo	Liberty	Mitaka	Newton
T V100R005 C02/C30	Х	√	√	√	√	√
T V200R002 C00	√	√	√	√	√	√
T V200R002 C20	√	√	√	√	√	√
T V200R002 C30	√	√	√	√	√	√
OceanStor V3 V300R001 C10	√	√	√	√	√	√
OceanStor V3 V3R1C20	√	√	√	√	√	√
OceanStor V3 V300R002 C10	√	√	√	√	√	√

Huawei storage system	IceHouse	Juno	Kilo	Liberty	Mitaka	Newton
OceanStor V3 V300R003 C00	х	√	√	√	√	√
OceanStor 2200 V3 V300R005 C00	x	√	√	√	1	√
OceanStor 2600 V3 V300R005 C00	x	√	√	√	1	√
OceanStor 18500/188 00 V100R001 C00	√	√	√	√	√	√
OceanStor 18500/188 00 V100R001 C20	√	√	√	√	√	√
OceanStor 18500/188 00 V100R001 C30	√	√	√	√	√	√
OceanStor 18500/188 00 V300R003 C00	х	√	√	√	√	√
OceanStor Dorado V3 V300R001 C00	Х	1	√	√	1	√

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

Features	IceHouse	Juno	Kilo	Liberty	Mitaka	Newton
Create Volume	√	√	√	√	√	√
Delete Volume	√	√	✓	√	✓	✓
Attach Volume	√	√	√	✓	√	✓
Detach Volume	✓	√	√	√	√	✓
Extend Volume	√	√	√	√	√	√
Create Snapshot	√	√	1	√	√	√
Delete Snapshot	√	√	√	√	√	√
List Snapshots	√	√	√	√	√	√
Create Volume from Snapshot	√	√	√	√	√	√
Create Volume from Image	√	√	√	√	√	√
Create Volume from Volume	√	√	√	√	√	1
Create Image from Volume	√ 	√	√	√	√	√
Volume Migration	x	√	1	√	√	√
QoS	√	√	√	√	√	√
Auto zoning	х	√	1	√	√	√

Features	IceHouse	Juno	Kilo	Liberty	Mitaka	Newton
SmartTier	X	√	√	√	√	√
SmartCach e	x	√	√	√	1	√
SmartThin/ Thick	x	✓	✓	√	√	✓
SmartPartit ion	X	√	√	√	√	√
HyperMetr o	x	✓	✓	√	√	✓
Retype	x	√	√	√	√	√
Manage/ Unmanage Volume	X	√	√	√	√	√
Manage/ Unmanage Snapshot	X	X	X	X	√	√
Replication V2.1	X	X	X	x	1	√
HyperMetr o Consistenc y Group	х	X	x	√	√	√
Assign StoragePoo	х	√	√	√	√	√
Assign DiskType	х	√	√	√	√	√
Backup Snapshot	х	х	х	х	√	√
Snapshot Consistenc y Group	х	x	х	х	√	√
Multipath	√	√	√	√	√	√
Consistenc y Group	х	X	х	√	√	√

Table 2-3 Version mappings among the Cinder Driver, Features and Huawei storage system. ($\sqrt{\ }$: support, x: unsupport)

Featu res	Т			OceanStor V3			Oceans 18500/1		Ocea nStor Dora do V3
	V1R5	V2R2	V3R1/ V3R2	V3R3	2200 V3R5	2600 V3R5	V1R1	V3R3	V3R1
Create Volum e	1	√	√	√	√	1	√	√	√
Delete Volum e	1	√	√	√	√	1	√	1	√
Attach Volum e	√	√	√	√	√	√	√	√	√
Detac h Volum e	√	√	√	√	√	√	√	√	√
Exten d Volum e	√	√	√	√	√	√	√	√	√
Create Snaps hot	√	√	√	√	√	√	√	√	√
Delete Snaps hot	√	√	√	√	√	√	√	√	√
List Snaps hots	√	√	√	√	√	√	√	√	√
Create Volum e from Snaps hot	√	√	√	√	√	√	√	√	x

Featu res	Т	Γ		Stor V3			Oceans 18500/1		Ocea nStor Dora do V3
Create Volum e from Image	√	√	√	√	√	√	√	√	√
Create Volum e from Volum e	√	✓	✓	√	√	√	√	√	х
Create Image from Volum e	√	√	√	√	√	√	√	√	√
Volum e Migrat ion	X	1	√	√	√	√	√	√	√
QoS	X	√	√	√	√	√	√	√	X
Auto zoning	√	√	1	√	√	√	√	√	√
Smart Tier	х	√	√	√	√	√	√	√	Х
Smart Cache	X	√	√	√	√	√	√	√	Х
Smart Thin	√	√	√	√	√	√	√	√	√
Smart Thick	√	√	√	√	√	√	√	√	х
Smart Partiti on	X	√	√	√	X	√	√	√	х
Hyper Metro	х	х	х	√	х	√	х	√	Х
Retyp e	х	√	√	√	√	√	√	√	х

Featu res	T	Т		Stor V3			Ocean5 18500/1		Ocea nStor Dora do V3
Mana ge/ Unma nage Volum e	x	√ 	√	√	√	√	√	√	√
Mana ge/ Unma nage Snaps hot	x	√ 	√	√	√	√	√	√	√
Replic ation V2.1	X	√	√	√	√	√	√	1	✓
Hyper Metro Consis tency Group	x	x	X	√	X	√	x	√	x
Assig n Storag ePool	X	1	√	√	√	√	√	√	√
Assig n DiskT ype	X	√	√	√	√	√	√	√	~
Backu p Snaps hot	X	√	√	√	√	√	√	√	✓
Snaps hot Consis tency Group	х	√	√	√	√	√	√	√	√
Multip ath	Х	√	√	√	√	√	√	√	√

Featu res	Т		Oceans	OceanStor V3			Ocean S 18500/1		Ocea nStor Dora do V3
Consis tency Group	X	√	√	√	√	√	√	√	√

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

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:

MOTE

After Kilo, Huawei OpenStack Driver is included in OpenStack community version, ignore the subsequent steps, jump to the relevant section to configure properties.

- **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.

- **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.

Except the above method, check /var/log/cinder/cinder-volume.log to confirm whether 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.

- **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 start 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

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:

```
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.

```
<?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>
       <LUN>
          <LUNType>xxx</LUNType>
          <StripUnitSize>xxx</StripUnitSize>
          <WriteType>xxx</WriteType>
         <Prefetch Type="xxx" Value="xxx" />
          <StoragePool Name="xxx" />
          <StoragePool Name="xxx" />
       </LUN>
       <iSCSI>
         <DefaultTargetIP>x.x.x.x/DefaultTargetIP>
          <Initiator Name="xxx" TargetIP="x.x.x.x" />
       <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x" />
 </config>
```

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.

```
<?xml version='1.0' encoding='UTF-8'?>
 <config>
      <Storage>
        <Product>TV2</Product>
        <Protocol>iSCSI</Protocol>
        <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
        <UserName>xxx
         <UserPassword>xxx</UserPassword>
      </Storage>
      <T.UN>
        <LUNType>xxx</LUNType>
        <WriteType>xxx</WriteType>
        <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
        <Timeout>432000</Timeout>
        <StoragePool>xxx;xxx;xxx</StoragePool>
      </LUN>
      <iSCSI>
        <DefaultTargetIP>x.x.x.x/DefaultTargetIP>
        <Initiator Name="xxx" TargetIP="x.x.x.x" />
      <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x" />
 </config>
```

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 (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.

```
<?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>
      <T.UN>
         <LUNType>xxx</LUNType>
         <WriteType>xxx</WriteType>
        <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
         <Timeout>432000</Timeout>
         <StoragePool>xxx;xxx;xxx</StoragePool>
      </LUN>
      <iscst>
       <DefaultTargetIP>x.x.x.x/DefaultTargetIP>
        <Initiator Name="xxx" TargetIP="x.x.x.x" />
      </iscst>
      <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x" />
</config>
```

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.

The driver configuration file of OceanStor 18000 series V1 storage system is shown as follows (x indicates parameters that can be set).

```
<?xml version='1.0' encoding='UTF-8'?>
 <config>
      <Storage>
         <Product>18000</Product>
         <Protocol>iSCSI</Protocol>
         <RestURL>https://x.x.x.x/deviceManager/rest/</RestURL>
         <UserName>xxx</UserName>
          <UserPassword>xxx</UserPassword>
      </Storage>
      <LUN>
          <LUNType>xxx</LUNType>
         <WriteType>xxx</WriteType>
         <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
         <Timeout>432000</Timeout>
          <StoragePool>xxx;xxx;xxx</StoragePool>
      </LUN>
      <iscst>
          <DefaultTargetIP>x.x.x.x/DefaultTargetIP>
         <Initiator Name="xxx" TargetIP="x.x.x.x" />
      <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x" />
 </config>
```

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.

The driver configuration file of OceanStor 18000 series V3 storage system is shown as follows (x indicates parameters that can be set).

```
<?xml version='1.0' encoding='UTF-8'?>
 <config>
      <Storage>
         <Product>18000</Product>
         <Protocol>iSCSI</Protocol>
         <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
         <UserName>xxx</UserName>
         <UserPassword>xxx</UserPassword>
      </Storage>
      <LUN>
         <LUNType>xxx</LUNType>
         <WriteType>xxx</WriteType>
         <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
         <Timeout>432000</Timeout>
         <StoragePool>xxx;xxx;xxx</StoragePool>
      </IJN>
      <iscsi>
```

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 "4.1.6".

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.

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
     <Storage>
        <Product>Dorado</Product>
         <Protocol>iSCSI</Protocol>
        <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
        <UserName>xxx</UserName>
        <UserPassword>xxx</UserPassword>
     </Storage>
        <LUNType>xxx</LUNType>
         <WriteType>xxx</WriteType>
        <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
        <Timeout>432000</Timeout>
         <StoragePool>xxx;xxx;xxx</StoragePool>
     </T.IJN>
      <DefaultTargetIP>x.x.x/DefaultTargetIP>
        <Initiator Name="xxx" TargetIP="x.x.x.x" />
```

```
</iscsi>
<Host OsType="Linux" HostIP="x.x.x.x, x.x.x.x" />
</config>
```

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
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 4.1.4.1 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



L CAUTION

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

Table 4-2 Optional parameters

Parameter	Default Value	Description	Applicable To
LUNType	Thick	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
LUNcopyWaitInterv al	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

Parameter	Default Value	Description	Applicable To
Initiator Target IP	-	IP address of the iSCSI target port that is provided for computing nodes.	All
Initiator TargetPortGroup	-	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.	All
OSType	Linux	Operating system of the Nova compute node's host.	All
DefaultTargetIP	-	Default IP address of the iSCSI target port that is provided for computing nodes.	All

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.

```
<iSCSI>
  <DefaultTargetIP>x.x.x.x</DefaultTargetIP>
  <Initiator Name="xxxxxx" TargetPortGroup="xxxx" />
  </iSCSI>
```

Step 3 Enable the multipathing switch of the OpenStack Nova module.

If the version of OpenStack is Havana and IceHouse, add libvirt_iscsi_use_multipath = True in [default] of /etc/nova/nova.conf.

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

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

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

service nova-compute restart.

----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:

```
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.

The driver configuration file of OceanStor T series V100R005 storage system is shown as follows (x indicates parameters that can be set).

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
    <Storage>
       <Product>T</Product>
       <Protocol>FC</Protocol>
        <ControllerIP0>x.x.x.x</ControllerIP0>
       <ControllerIP1>x.x.x.x</ControllerIP1>
       <UserName>xxx</UserName>
       <UserPassword>xxx</UserPassword>
     </Storage>
     <T.UN>
        <LUNType>xxx</LUNType>
        <StripUnitSize>xxx</StripUnitSize>
        <WriteType>xxx</WriteType>
       <Prefetch Type="xxx" Value="xxx" />
       <StoragePool Name="xxx" />
        <StoragePool Name="xxx" />
     <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x" />
```

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.

The driver configuration file of OceanStor T series V200R002 storage system is shown as follows (x indicates parameters that can be set).

```
<?xml version='1.0' encoding='UTF-8'?>
  <confia>
 <Storage>
        <Product>TV2</Product>
        <Protocol>FC</Protocol>
        <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
        <UserName>xxx
        <UserPassword>xxx</UserPassword>
 </Storage>
 <LUN>
        <LUNType>xxx</LUNType>
        <WriteType>xxx</WriteType>
        <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
      <Timeout>432000</Timeout>
        <StoragePool>xxx;xxx;xxx</StoragePool>
 <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x" />
  </config>
```

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 [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 (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.

The driver configuration file of OceanStor V3 storage system is shown as follows (x indicates parameters that can be set).

```
<?xml version='1.0' encoding='UTF-8'?>
  <config>
 <Storage>
        <Product>V3</Product>
        <Protocol>FC</protocol>
        <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
        <UserName>xxx</UserName>
        <UserPassword>xxx</UserPassword>
 </Storage>
 <LUN>
        <LUNType>xxx</LUNType>
        <WriteType>xxx</WriteType>
        <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
        <Timeout>432000</Timeout>
        <StoragePool>xxx;xxx;xxx</StoragePool>
 <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x" />
```

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.

The driver configuration file of OceanStor 18000 series V1 storage system is shown as follows (x indicates parameters that can be set).

```
<?xml version='1.0' encoding='UTF-8'?>
  <config>
       <Storage>
          <Product>18000</Product>
          <Protocol>FC</Protocol>
         <RestURL>https://x.x.x.x/deviceManager/rest/</RestURL>
          <UserName>xxx</UserName>
          <UserPassword>xxx</UserPassword>
       </Storage>
       <T.UN>
          <LUNType>xxx</LUNType>
          <WriteType>xxx</WriteType>
        <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
        <Timeout>432000</Timeout>
         <StoragePool>xxx;xxx;xxx</StoragePool>
       <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x" />
 </config>
```

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 [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.

The driver configuration file of OceanStor 18000 series V3 storage system is shown as follows (x indicates parameters that can be set).

```
<?xml version='1.0' encoding='UTF-8'?>
 <config>
       <Storage>
         <Product>18000</Product>
          <Protocol>FC</Protocol>
         <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
         <UserName>xxx</UserName>
          <UserPassword>xxx</UserPassword>
       </Storage>
       <LUN>
          <LUNType>xxx</LUNType>
          <WriteType>xxx</WriteType>
         <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
       <Timeout>432000</Timeout>
          <StoragePool>xxx;xxx;xxx</StoragePool>
       <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x" />
 </config>
```

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.

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
      <Storage>
         <Product>Dorado</Product>
         <Protocol>FC</Protocol>
         <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
         <!!serName>xxx</!!serName>
         <UserPassword>xxx</UserPassword>
      </Storage>
      <T.UN>
         <LUNType>xxx</LUNType>
         <WriteType>xxx</WriteType>
         <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
         <Timeout>432000</Timeout>
         <StoragePool>xxx;xxx;xxx</StoragePool>
      </LUN>
      <iscsi>
       <DefaultTargetIP>x.x.x.x/DefaultTargetIP>
        <Initiator Name="xxx" TargetIP="x.x.x.x" />
      <Host OSType="Linux" HostIP="x.x.x.x, x.x.x.x" />
</config>
```

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 http://docs.openstack.org/mitaka/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
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 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
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

Parameter	Default Value	Description	Applicable To
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	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
LUNcopyWaitInterv al	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.	All
OSType	Linux	Operating system of the Nova compute node's host.	All

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 and Mitaka, add **iscsi_use_multipath** = **True** in **[libvirt]** of /etc/nova/nova.conf.

If the version of OpenStack is Newton, 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.

service nova-compute restart.

----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 Configuring Assign StoragePool
- 5.10 Configuring Assign DiskType
- 5.11 Configuring Consistency Groups
- 5.12 Configuring Backup Snapshot

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 gos.

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 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 gos-associate' for more information.

2. Run the **cinder qos-associate** 7dc73b5b-1b19-4371-8dee-8edbc52cf625 84b189d3-8984-4e92-aab0-fa4a913126bf。

----End

Results

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

NOTE

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	
V3	V300R003C00	

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.

- **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 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.

```
root@ubuntu:/# cinder qos-create miniops minIOPS=50 IOType=2
+-----+
| property | value |
+-----+
| consumer | back-end |
| id | 8deaf13c-bf1c-4a71-94d0-1149805693ce |
| name | miniops |
| specs | {u'IOType': u'2', u'minIOPS': u'50'} |
+-----+
```

- **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** 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

```
root@ubuntu:/# cinder type-create maxBandWidth

+------+

+ ID | Name | Description | Is_Public
```

- 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

```
root@ubuntu:/# cinder type-create minBandWidth
+------+

| ID | Name | Description | Is_Public |
+-----+
```

- 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 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** 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.

- **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 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.

```
root@ubuntu:/# cinder qos-create multi-strategy latency=100 minBandWidth=100

IOType=2
+------+
| Property | Value | |
+------+
| consumer | back-end | |
| id | 36f0fc4a-d5f5-4862-ab1f-e7fedeeddc41 | |
| name | multi-strategy | |
| specs | {'IOType': '2', 'minBandWidth': '100', 'latenct': '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 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 gos-list
         ID
                       | Name | Consumer
| 36f0fc4a-d5f5-4862-ab1f-e7fedeeddc41 | multi-strategy | back-end |
{'IOType': '2', 'minBandWidth': '100', 'latenct': '100'} |
{'IOType': '2', 'minIOPS': '50'}
| 3fd0971e-f5fd-48d7-89d9-d2efb6fc44cf | maxbandwidth | back-end
   {'maxBandWidth': '100', 'IOType': '2'}
| 6a4e24af-0e8c-4a64-91ee-859434414400 | latency
                                         | back-end
| {'latency': '100', 'IOType': '2'}
| 9b34fbdf-91b5-473d-a5ff-0b4ea2ed9fd0 | maxiops | back-end
    {'IOType': '2', 'maxIOPS': '100'}
{'IOType': '2', 'minBandWidth': '100'}
```

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

ID	-	Name		Description	1
Is_Public					
+	-+-		-+-		•
++					
3f19ccd1-3ac2-4ce7-b907-d532b8ddbe52		maxBandWidth		-	1
True					
92dbab58-07d8-46e6-9327-532a9a3b30b6	-	maxIOPS		-	1
True					
98d5ad9c-4591-4d18-ba09-199f6dba95a9		minBandWidth		-	1
True					
df53d9d7-b1db-4e6b-847a-7a5150c39489		multi-strategy		-	1
True					
e28d430e-ccc5-4f18-914a-b0c8008dc42d		latency		-	1
True					
f63f6eb4-7036-4452-bc46-9d31cba095d1		minIOPS		-	1
True					
+	-+-		-+-		-
++					

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**.

	size		1		
	snapshot_id	- 1	none		
	source_volid	- 1	none		
1	status		creating		
	volume_type	1	SmartPartition		
+		+		+	

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**.

----End

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.

- 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.

----End

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**.

root	:@ubuntu-204:~# cir	nder cre	ate 1	display-name	test001	volume-type	SmartTier
	property			value			
+-	attachments availability zone	+ 		[] nova		 	

bootable	false	
created_at	2015-06-26t13:40:10.483450	1
display_description	none	
display_name	test001	1
encrypted	false	
id	1b2a87e4-eb8c-46ab-922c-63fcd7db8f20	1
metadata	{}	1
multiattach	false	1
size	1	1
snapshot_id	none	1
source_volid	none	1
status	creating	
volume_type	SmartTier	1
+	+	-+

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

Parameter	Default Value	Description	Applicable To
metro_san_user		User name of a storage administrator of hypermetro remote device.	V3R3 2600 V3R5 18000 V3R3
metro_san_passwor d		Password of a storage administrator of hypermetro remote device.	V3R3 2600 V3R5 18000 V3R3
metro_domain_nam e		Hypermetro domain name configured on ISM.	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
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.

Procedure

```
| 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**.

```
root@ubuntu-204:~# cinder create 1 --display-name test001 --volume-type
Hypermetro
       property |
      attachments |
                                   []
    availability zone |
                                   nova
   bootable | false created_at | 2015-06-26t13:40:10.483450 display_description | none
                        none
test001
      display_name |
      snapshot id
                                   none
      source_volid |
                                   none
       status | volume_type |
                                 creating
                                HyperMetro
```

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
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.

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.**

----End

5.9 Configuring Assign StoragePool

This section describes how to configure the assign storagepool to create a volume.

Procedure

```
+-----+
| 2ecf5589-507e-4ee5-995f-4aa1d0aaff83 | target-pool | - | True |
```

Step 2 Run the following command to configure the assign storagepool.

Configure a single assign pool

cinder type-key target-pool set pool_name=StoragePool001

• Configure multiple assign pools.

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

Step 3 Create a volume that supports **Assign StorgePool**.

root@ubuntu-175:~# cinder create	volume-type target-pool 1
Property	Value
+ attachments	+ []
availability zone	l nova l
bootable	false
consistencygroup id	None I
created at	2016-08-12T08:14:21.000000
description	None
encrypted	False
id	3023e949-ec25-4005-8bbc-6aa2c6860235
metadata	[{ }
migration status	None
multiattach	False
name	None
os-vol-host-attr:host	None
os-vol-mig-status-attr:migstat	None
os-vol-mig-status-attr:name_id	None
os-vol-tenant-attr:tenant_id	53c46df66a084916876a08d7bcc31d87
replication_status	disabled
size	1
snapshot_id	None
source_volid	None
status	creating
updated_at	None
user_id	ec5b1bd5fa8646109381755037aacc82
volume_type	target-pool

----End

5.10 Configuring Assign DiskType

This section describes how to configure the assign disk type to create a 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 assign 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 that supports Assign **DiskType**.

root@ubuntu-175:~# cinder create	volume-type disk-type 1	
Property	Value	
attachments	+ []	+
availability zone	nova	1
bootable	false	1
consistencygroup id	None	1
created at	2016-08-12T08:24:09.000000	1
description	None	-
encrypted	False	-
id	eca3a81b-0ef0-49b6-9202-d1bf1c16a96f	-
metadata	{}	-
migration_status	None	-
multiattach	False	-
name	None	-
os-vol-host-attr:host	ubuntu-175@234#StoragePool002	-
os-vol-mig-status-attr:migstat	None	-
os-vol-mig-status-attr:name_id	None	-
os-vol-tenant-attr:tenant_id	53c46df66a084916876a08d7bcc31d87	
replication_status	disabled	
size	1	-
snapshot_id	None	
source_volid	None	-
status	creating	-
updated_at	2016-08-12T08:24:11.000000	-
user_id	ec5b1bd5fa8646109381755037aacc82	-
volume_type	disk-type	-

----End

5.11 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.

5.12 Configuring Backup Snapshot

This section describes how to configure backup snapshot.

Procedure

Step 1 Configure backup driver.

For more details please refer to http://docs.openstack.org/mitaka/config-reference/block-storage/backup-drivers.html, and makee sure cinder-backup service is working.

Step 2 Configure the cinder.conffile.

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