



## OpenStack Cinder Driver

# 配置指南

文档版本 01

发布日期 2017-08-04

版权所有 © 华为技术有限公司 2017。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 华为技术有限公司

地址：深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址：<http://www.huawei.com>

客户服务邮箱：[support@huawei.com](mailto:support@huawei.com)

客户服务电话：4008302118

## 目 录

<b>1 简介</b>	<b>1</b>
<b>2 版本配套关系</b>	<b>2</b>
<b>3 Cinder Driver 安装与部署</b>	<b>7</b>
3.1 Cinder Driver 获取	8
3.2 Ubuntu 环境部署	8
3.3 RedHat 环境部署	9
<b>4 Cinder Driver 基本属性配置</b>	<b>10</b>
4.1 iSCSI 组网 Cinder Driver 配置	11
4.1.1 配置 T 系列 V1 iSCSI 协议 Cinder Driver	11
4.1.2 配置 T 系列 V2 iSCSI 协议 Cinder Driver	12
4.1.3 配置 V3 iSCSI 协议 Cinder Driver	13
4.1.4 配置 18000 系列 iSCSI 协议 Cinder Driver	14
4.1.4.1 配置 18000 系列 V1 iSCSI 协议 Cinder Driver	14
4.1.4.2 配置 18000 系列 V3 iSCSI 协议 Cinder Driver	16
4.1.5 配置 Dorado 系列 iSCSI 协议 Cinder Driver	17
4.1.6 配置文件参数属性说明	18
4.1.7 配置 iSCSI 多路径	20
4.1.8 配置 CHAP 和 ALUA 特性	20
4.2 配置 FC 组网 Cinder Driver	21
4.2.1 配置 T 系列 V1 FC 协议 Cinder Driver	21
4.2.2 配置 T 系列 V2 FC 协议 Cinder Driver	22
4.2.3 配置 V3 FC 协议 Cinder Driver	23
4.2.4 配置 18000 系列 FC 协议 Cinder Driver	24
4.2.4.1 配置 18000 系列 V1 FC 协议 Cinder Driver	24
4.2.4.2 配置 18000 系列 V3 FC 协议 Cinder Driver	25
4.2.5 配置 Dorado 系列 FC 协议 Cinder Driver	26
4.2.6 配置智能划域	27
4.2.7 配置文件参数属性说明	28
4.2.8 配置 FC 多路径	29
4.3 配置多存储支持	30
<b>5 Cinder Driver 高级特性配置</b>	<b>31</b>
5.1 配置 SmartQoS 属性	32

5.1.1 配置前端 QoS.....	32
5.1.2 QoS 配置版本说明.....	33
5.1.3 配置控制 IOPS.....	33
5.1.3.1 配置控制最大 IOPS.....	33
5.1.3.2 配置控制最小 IOPS.....	34
5.1.4 配置控制带宽.....	35
5.1.4.1 配置控制最大带宽.....	35
5.1.4.2 配置控制最小带宽.....	36
5.1.5 配置控制时延.....	37
5.1.6 配置控制多策略.....	38
5.2 配置 SmartPartition 属性.....	39
5.3 配置 Thick 属性.....	40
5.4 配置 Thin 属性.....	41
5.5 配置 SmartCache 属性.....	41
5.6 配置 SmartTier 属性.....	42
5.7 配置 HyperMetro 属性.....	43
5.8 配置 Replication V2.1.....	45
5.9 指定 Storage Pool 创建 LUN.....	47
5.10 指定 Disk Type 创建 LUN.....	48
5.11 指定 Volume Backend 创建 LUN.....	49
5.12 配置 Consistency Groups.....	50
5.13 配置 Backup Snapshot.....	50
5.14 配置证书认证.....	50
<b>6 最佳实践.....</b>	<b>52</b>
6.1 快速对接华为存储.....	53

# 1 简介

---

介绍Cinder Driver定义。

Cinder Driver是部署在OpenStack Cinder模块上的一个插件，该插件用于向OpenStack中的虚拟机提供逻辑卷及快照等功能。Cinder Driver支持iSCSI和FC两种协议。

# 2 版本配套关系

介绍Cinder Driver与华为存储系统、OpenStack的版本配套关系。

表 2-1 Cinder Driver 存储产品版本支持说明

OpenStack版本	存储产品版本
Juno	T系列 V100R005 C02/C30
Kilo	T系列 V200R002 C00/C20/C30
Liberty	OceanStor V3 V300R001C10/C20
Mitaka	OceanStor V3 V300R002C10
Newton	OceanStor V3 V300R003C00/C10/C20
Ocata	OceanStor 2200 V3 V300R005C00
	OceanStor 2600 V3 V300R005C00
	OceanStor V3 V300R006C00
	OceanStor 18500/18800
	V100R001C00/C20/C30
	OceanStor 18500/18800 V300R003C00
	OceanStor 18500/18800 V300R006C00
	OceanStor Dorado V3 V300R001C00

表 2-2 Cinder Driver 特性支持说明（√：支持，x：不支持）

特性	Juno	Kilo	Liberty	Mitaka	Newton	Ocata
Create Volume	√	√	√	√	√	√
Delete Volume	√	√	√	√	√	√
Attach Volume	√	√	√	√	√	√
Detach Volume	√	√	√	√	√	√
Extend Volume	√	√	√	√	√	√

特性	Juno	Kilo	Liberty	Mitaka	Newton	Ocata
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	√	√	√	√	√	√
Re-type	√	√	√	√	√	√
Manage/Unmanage Volume	√	√	√	√	√	√
Manage/Unmanage Snapshot	x	x	x	√	√	√
Replication V2.1	x	x	x	√	√	√
HyperMetro Consistency Group	x	x	√	√	√	√
Create a volume on a certain storage pool	√	√	√	√	√	√
Create a volume with certain disk type	√	√	√	√	√	√
Backup Snapshot	x	x	x	√	√	√
Snapshot Consistency Group	x	x	x	√	√	√
Multipath	√	√	√	√	√	√

特性	Juno	Kilo	Liberty	Mitaka	Newton	Ocata
Consistency Group	x	x	√	√	√	√

表 2-3 Cinder Driver 存储产品版本与特性支持说明

特性	存储产品版本
Create, delete, expand, attach, and detach volumes Copy an image to a volume Copy a volume to an image Auto Zoning SmartThin Create and delete snapshots	T系列 V100R005 C02/C30 T系列 V200R002 C00/C20/C30 OceanStor V3 V300R001C10/C20 OceanStor V3 V300R002C10 OceanStor V3 V300R003C00/C10/C20 OceanStor 2200 V3 V300R005C00 OceanStor 2600 V3 V300R005C00 OceanStor V3 V300R006C00 OceanStor 2200 V3 V300R006C00 OceanStor 2600 V3 V300R006C00 OceanStor 18500/18800 V100R001C00/C20/C30 OceanStor 18500/18800 V300R003C00 OceanStor 18500/18800 V300R006C00 OceanStor Dorado V3 V300R001C00
Create a volume on a certain storage pool Create a volume with certain disk type Create, delete and update a consistency group Manage and unmanage volumes Volume Migration Create and delete a cgsnapshot Manage, unmanage, and backup snapshots Replication V2.1	T系列 V200R002 C00/C20/C30 OceanStor V3 V300R001C10/C20 OceanStor V3 V300R002C10 OceanStor V3 V300R003C00/C10/C20 OceanStor 2200 V3 V300R005C00 OceanStor 2600 V3 V300R005C00 OceanStor V3 V300R006C00 OceanStor 2200 V3 V300R006C00 OceanStor 2600 V3 V300R006C00 OceanStor 18500/18800 V100R001C00/C20/C30 OceanStor 18500/18800 V300R003C00 OceanStor 18500/18800 V300R006C00 OceanStor Dorado V3 V300R001C00



特性	存储产品版本
Clone a volume Create volume from snapshot SmartThick	T系列 V100R005 C02/C30 T系列 V200R002 C00/C20/C30 OceanStor V3 V300R001C10/C20 OceanStor V3 V300R002C10 OceanStor V3 V300R003C00/C10/C20 OceanStor 2200 V3 V300R005C00 OceanStor 2600 V3 V300R005C00 OceanStor V3 V300R006C00 OceanStor 2200 V3 V300R006C00 OceanStor 2600 V3 V300R006C00 OceanStor 18500/18800 V100R001C00/C20/C30 OceanStor 18500/18800 V300R003C00 OceanStor 18500/18800 V300R006C00
QoS Retype SmartTier SmartCache	T系列 V200R002 C00/C20/C30 OceanStor V3 V300R001C10/C20 OceanStor V3 V300R002C10 OceanStor V3 V300R003C00/C10/C20 OceanStor 2200 V3 V300R005C00 OceanStor 2600 V3 V300R005C00 OceanStor V3 V300R006C00 OceanStor 2200 V3 V300R006C00 OceanStor 2600 V3 V300R006C00 OceanStor 18500/18800 V100R001C00/C20/C30 OceanStor 18500/18800 V300R003C00 OceanStor 18500/18800 V300R006C00
SmartPartition	T系列 V200R002 C00/C20/C30 OceanStor V3 V300R001C10/C20 OceanStor V3 V300R002C10 OceanStor V3 V300R003C00/C10/C20 OceanStor 2600 V3 V300R005C00 OceanStor V3 V300R006C00 OceanStor 2600 V3 V300R006C00 OceanStor 18500/18800 V100R001C00/C20/C30 OceanStor 18500/18800 V300R003C00 OceanStor 18500/18800 V300R006C00

特性	存储产品版本
Hypermetro	OceanStor V3 V300R003C00/C10/C20
Hypermetro consistency group	OceanStor 2600 V3 V300R005C00
	OceanStor V3 V300R006C00
	OceanStor 2600 V3 V300R006C00
	OceanStor 18500/18800 V300R003C00
	OceanStor 18500/18800 V300R006C00

# 3 Cinder Driver 安装与部署

---

[3.1 Cinder Driver获取](#)

[3.2 Ubuntu环境部署](#)

[3.3 RedHat环境部署](#)

## 3.1 Cinder Driver 获取

您可以通过两种途径获取到OpenStack Driver:

第一种是通过OpenStack社区仓库。从Kilo版本开始，华为存储就已经将华为存储驱动完全贡献给OpenStack开源社区，用户可自由下载贡献到OpenStack社区的OpenStack Driver版本。安装完成指定的OpenStack版本后，指定的OpenStack Driver放置在../cinder/cinder/volume/drivers/huawei目录下。安装完成后未发现对应的安装文件，您也可以通过OpenStack官网仓库下载指定的OpenStack Driver使用，下载地址为：<https://github.com/openstack/cinder>。

第二种是通过华为自有OpenStack Driver仓库。仓库地址为：[https://github.com/huaweistorage/OpenStack\\_Driver](https://github.com/huaweistorage/OpenStack_Driver)，访问到该仓库地址后，您可以下载到和OpenStack社区版本相对应的OpenStack Driver。

### 说明

OpenStack社区版本发布后，不允许合入新的特性，bug修复周期长且存在风险。

华为自有OpenStack Driver仓库版本，在框架支持的前提下能够保证新特性及时发布，bug及时修复，相较社区版本更稳定完善。

OpenStack社区仅维护两个稳定版本，华为自有OpenStack Driver仓库维护六个稳定版本，能够保障历史版本长期稳定运行。

**强烈推荐使用华为自有OpenStack Driver仓库版本替换社区版本。**

Cinder Driver获取方法具体操作如下：

- 步骤1** 打开网页浏览器，键入上述仓库地址：[https://github.com/huaweistorage/OpenStack\\_Driver](https://github.com/huaweistorage/OpenStack_Driver)。
- 步骤2** 点击“Download ZIP”按钮，驱动将以压缩包的方式下载到本地，对该压缩包解压。
- 步骤3** 在解压完成后的目录下找到Cinder目录，其下包含多个OpenStack版本的 OpenStack Driver插件，选择相应版本OpenStack插件即可。

----结束

## 3.2 Ubuntu 环境部署

OpenStack社区标准部署步骤如下：

- 步骤1** 安装前需删除默认华为OpenStack Driver安装目录下所有文件，默认安装路径为/usr/lib/python2.7/dist-packages/cinder/volume/drivers/huawei。

### 说明

python2.7为本机使用的Python版本，如果为其余版本，该处为相应Python版本号。可通过如下方法，搜索OpenStack Driver安装目录：

```
root@ubuntu:~# find / -name huawei_driver.py
/usr/lib/python2.7/dist-packages/cinder/volume/drivers/huawei/huawei_driver.py
```

- 步骤2** 将获取到的OpenStack Cinder Driver拷贝到Cinder节点驱动安装目录。
- 步骤3** 按照章节4和章节5进行配置。
- 步骤4** 待配置完毕后，重启Cinder-Volume服务，启动命令为：

```
service cinder-volume restart
```

**步骤5** 使用cinder service-list查看服务启动详情，查看Cinder-Volume服务的State状态为up状态表明服务已经正常启动。

```
root@ubuntuL004:/# cinder service-list
```

Updated_at	Binary	Host	Zone	Status	State
2016-02-01T16:26:00.000000	cinder-scheduler	ubuntuL004	nova	enabled	up
2016-02-01T16:25:53.000000	cinder-volume	ubuntuL004@v3r3	nova	enabled	up

----结束

## 3.3 RedHat 环境部署

RedHat OpenStack部署步骤如下：

**步骤1** 安装前需删除默认华为OpenStack Driver安装目录下所有文件，默认安装路径为/usr/lib/python2.7/site-packages/cinder/volume/drivers/huawei。

 **说明**

python2.7为本机使用的Python版本，如果为其余版本，该处为相应Python版本号。可通过如下方法，搜索OpenStack Driver安装目录：

```
root@redhatL004:~# find / -name huawei_driver.py
/usr/lib/python2.7/dist-packages/cinder/volume/drivers/huawei/huawei_driver.py
```

**步骤2** 将获取到的OpenStack Cinder Driver拷贝到Cinder节点驱动安装目录，默认路径参考步骤1。

**步骤3** 按照章节4和章节5进行配置。

**步骤4** 待配置完毕后，重启Cinder-Volume服务，启动命令为：

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

**步骤5** 使用cinder service-list查看服务启动详情，查看Cinder-Volume服务的State状态为up状态表明服务已经正常启动。

```
root@redhatL004:/# cinder service-list
```

Updated_at	Binary	Host	Zone	Status	State
2016-02-01T16:26:00.000000	cinder-scheduler	redhatL004	nova	enabled	up
2016-02-01T16:25:53.000000	cinder-volume	redhatL004@v3r3	nova	enabled	up

----结束

# 4 Cinder Driver 基本属性配置

---

分别介绍在iSCSI协议和FC协议下如何配置华为Cinder Driver。

[4.1 iSCSI组网Cinder Driver配置](#)

[4.2 配置FC组网Cinder Driver](#)

[4.3 配置多存储支持](#)

## 4.1 iSCSI 组网 Cinder Driver 配置

介绍在iSCSI协议下如何配置不同华为存储产品的Cinder Driver。

### 前提条件

若使用从镜像创建卷功能，需安装多路径软件，且在Cinder Volume节点/etc/cinder/cinder.conf文件里面的指定后端里增加如下配置信息：

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

### 4.1.1 配置 T 系列 V1 iSCSI 协议 Cinder Driver

#### 操作步骤

**步骤1** 在“/etc/cinder”路径下创建华为自定义Driver配置文件，文件格式为XML格式。

可根据实际情况修改Driver配置文件名称。例如，Driver配置文件名为“cinder\_huawei\_conf.xml”。

**步骤2** 配置华为自定义Driver配置文件参数，分为必填参数和选填参数。

必填参数：

```
<?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>
    <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" />
</config>
```

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

```
<LUN>
  <LUNType>xxx</LUNType>
  <StripUnitSize>xxx</StripUnitSize>
  <WriteType>xxx</WriteType>
  <Prefetch Type="xxx" Value="xxx" />
</LUN>
<iSCSI>
  <Initiator Name="xxx" TargetIP="x.x.x.x" />
</iSCSI>
```



说明

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的拥有者及用户组与“/etc/cinder/cinder.conf”文件的拥有者及用户组一致：

```
-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
```

请参考“4.1.6”来配置参数属性。

### 步骤3 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为TV1\_iSCSI后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT]区块中修改以下内容，配置TV1\_iSCSI后端：

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

### 步骤4 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

## 4.1.2 配置 T 系列 V2 iSCSI 协议 Cinder Driver

### 操作步骤

#### 步骤1 在“/etc/cinder”路径下创建华为自定义Driver配置文件，文件格式为XML。

可根据实际情况修改Driver配置文件名称。例如，Driver配置文件名为“cinder\_huawei\_conf.xml”。

#### 步骤2 配置华为自定义Driver配置文件参数，分为必填参数和选填参数。

必填参数：

```
<?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</UserName>
    <UserPassword>xxx</UserPassword>
  </Storage>
  <LUN>
    <StoragePool>xxx</StoragePool>
  </LUN>
  <iSCSI>
    <DefaultTargetIP>x.x.x.x</DefaultTargetIP>
  </iSCSI>
</config>
```

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

```
<LUN>
  <LUNType>xxx</LUNType>
```



```
<WriteType>xxx</WriteType>
<Prefetch Type="xxx" Value="xxx" />
<LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
<Timeout>432000</Timeout>
</LUN>
<iSCSI>
  <Initiator Name="xxx" TargetIP="x. x. x. x" />
</iSCSI>
```

#### 说明

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的拥有者及用户组与“/etc/cinder/cinder.conf”文件的拥有者及用户组一致：

```
-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
```

请参考“4.1.6”来配置参数属性。

#### 步骤3 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为TV2\_iSCSI后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT]区块中修改以下内容，配置TV2\_iSCSI后端：

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

#### 步骤4 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

## 4.1.3 配置 V3 iSCSI 协议 Cinder Driver

### 操作步骤

#### 步骤1 在“/etc/cinder”路径下创建华为自定义Driver配置文件，文件格式为XML。

可根据实际情况修改Driver配置文件名称。例如，Driver配置文件名为“cinder\_huawei\_conf.xml”。

#### 步骤2 配置华为自定义Driver配置文件参数，分为必填参数和选填参数。

必填参数：

```
<?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>
    <StoragePool>xxx</StoragePool>
  </LUN>
  <iSCSI>
    <DefaultTargetIP>x. x. x. x</DefaultTargetIP>
```

```
</iSCSI>
</config>
```

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

```
<LUN>
  <LUNType>xxx</LUNType>
  <WriteType>xxx</WriteType>
  <Prefetch Type="xxx" Value="xxx" />
  <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
  <Timeout>432000</Timeout>
</LUN>
<iSCSI>
  <Initiator Name="xxx" TargetIP="x.x.x.x" />
</iSCSI>
```

#### 说明

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的拥有者及用户组与“/etc/cinder/cinder.conf”文件的拥有者及用户组一致：

```
-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
```

请参考“4.1.6”来配置参数属性。

#### 步骤3 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为V3\_iSCSI后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT]区块中修改以下内容，配置V3\_iSCSI后端：

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

#### 步骤4 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

## 4.1.4 配置 18000 系列 iSCSI 协议 Cinder Driver

介绍在iSCSI协议下配置OceanStor 18000系列V1和V3存储系统Cinder Driver。

### 4.1.4.1 配置 18000 系列 V1 iSCSI 协议 Cinder Driver

#### 操作步骤

**步骤1** 在“/etc/cinder”路径下创建华为自定义Driver配置文件，文件格式为XML格式。

可根据实际情况修改Driver配置文件名称。例如，Driver配置文件名为“cinder\_huawei\_conf.xml”。

**步骤2** 配置华为自定义Driver配置文件参数，分为必填参数和选填参数。

必填参数：

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

```
<Storage>
  <Product>18000</Product>
  <Protocol>iSCSI</Protocol>
  <RestURL>https://x.x.x/deviceManager/rest/</RestURL>
  <UserName>xxx</UserName>
  <UserPassword>xxx</UserPassword>
</Storage>
<LUN>
  <StoragePool>xxx</StoragePool>
</LUN>
<iSCSI>
  <DefaultTargetIP>x.x.x</DefaultTargetIP>
</iSCSI>
</config>
```

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

```
<LUN>
  <LUNType>xxx</LUNType>
  <WriteType>xxx</WriteType>
  <Prefetch Type="xxx" Value="xxx" />
  <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
  <Timeout>432000</Timeout>
</LUN>
<iSCSI>
  <Initiator Name="xxx" TargetIP="x.x.x.x" />
</iSCSI>
```

#### 说明

- 确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的拥有者及用户组与“/etc/cinder/cinder.conf”文件的拥有者及用户组一致：  

```
-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
```

  
参数请参考“4.1.6”来配置参数属性。
- 配置Rest接口访问地址时，若全新灌装SVP，需要加上8088端口；若是从C99升级上来，默认不加端口号。

### 步骤3 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为18000V1\_iSCSI后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT]区块中修改以下内容，配置18000V1\_iSCSI后端：

```
[DEFAULT]
...
enabled_backends=18000V1_iSCSI
```

### 步骤4 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

### 4.1.4.2 配置 18000 系列 V3 iSCSI 协议 Cinder Driver

#### 操作步骤

**步骤1** 在“/etc/cinder”路径下创建华为自定义Driver配置文件，文件格式为XML格式。

可根据实际情况修改Driver配置文件名称。例如，Driver配置文件名为“cinder\_huawei\_conf.xml”。

**步骤2** 配置华为自定义Driver配置文件参数，分为必填参数和选填参数。

必填参数：

```
<?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>
    <StoragePool>xxx</StoragePool>
  </LUN>
  <iSCSI>
    <DefaultTargetIP>x.x.x.x</DefaultTargetIP>
  </iSCSI>
</config>
```

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

```
<LUN>
  <LUNType>xxx</LUNType>
  <WriteType>xxx</WriteType>
  <Prefetch Type="xxx" Value="xxx" />
  <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
  <Timeout>432000</Timeout>
</LUN>
<iSCSI>
  <Initiator Name="xxx" TargetIP="x.x.x.x" />
</iSCSI>
```

#### 说明

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的拥有者及用户组与“/etc/cinder/cinder.conf”文件的拥有者及用户组一致：

```
-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
```

请参考“4.1.6”来配置参数属性。

**步骤3** 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为18000V3\_iSCSI后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT]区块中修改以下内容，配置18000V3\_iSCSI后端：

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

**步骤4** 执行 `service cinder-volume restart` 命令，重新启动Cinder服务。

----结束

## 4.1.5 配置 Dorado 系列 iSCSI 协议 Cinder Driver

### 操作步骤

**步骤1** 在“/etc/cinder”路径下创建华为自定义Driver配置文件，文件格式为XML。

可根据实际情况修改Driver配置文件名称。例如，Driver配置文件名为“cinder\_huawei\_conf.xml”。

**步骤2** 配置华为自定义Driver配置文件参数，分为必填参数和选填参数

必填参数：

```
<?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>
  <LUN>
    <LUNType>Thin</LUNType>
    <StoragePool>xxx</StoragePool>
  </LUN>
  <iSCSI>
    <DefaultTargetIP>x.x.x.x</DefaultTargetIP>
  </iSCSI>
</config>
```

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

```
<LUN>
  <WriteType>xxx</WriteType>
  <Prefetch Type="xxx" Value="xxx" />
</LUN>
<iSCSI>
  <Initiator Name="xxx" TargetIP="x.x.x.x" />
</iSCSI>
```

#### 说明

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的所有者及用户组与“/etc/cinder/cinder.conf”文件的所有者及用户组一致：

```
-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
```

请参考“4.1.6”来配置参数属性。

**步骤3** 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为Dorado\_iSCSI后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT] 区块中修改以下内容，配置Dorado\_iSCSI后端：

```
[DEFAULT]  
...  
enabled_backends=Dorado_iSCSI
```

**步骤4** 执行**service cinder-volume restart**命令，重新启动Cinder服务。

----结束

## 4.1.6 配置文件参数属性说明

表 4-1 必填参数说明

属性名称	默认值	说明	适用产品
Product	-	存储产品类型，可选值为“T”、“18000”、“TV2”、“V3”和“Dorado”。	全部
Protocol	-	连接协议类型，选值为“iSCSI”。	全部
ControllerIP0	-	T系列V100R005存储设备主控制器IP地址。	TV1
ControllerIP1	-	T系列V100R005存储设备从控制器IP地址。	TV1
RestURL	-	Rest接口访问地址。参考配置： <a href="https://x.x.x.x:8088/deviceManager/rest/">https://x.x.x.x:8088/ deviceManager/rest/</a> x.x.x.x代表管理IP地址，18000系列V1参考4.1.4.1。如果需要配置多个RestURL，以分号“;”进行分割。	TV2 V3 18000 Dorado V3
UserName	-	存储管理员用户名。	全部
UserPassword	-	存储管理员密码。	全部
StoragePool	-	需要使用的存储池名称，需提前在目标阵列上创建。如果需要配置多个资源池，以分号“;”进行分割。TV1系列参考4.1.1	全部
DefaultTargetIP	-	存储系统提供给计算节点的默认iSCSI目标器端口IP。	全部



小心

StoragePool不允许配置为中文。

表 4-2 选填参数说明

属性名称	默认值	说明	适用产品
LUNType	Thick（除Dorado） Thin（Dorado）	创建LUN的类型，可选值为“Thick”或者“Thin”。Dorado系列仅支持“Thin”。	全部
StripUnitSize	64	创建LUN的分条深度大小，可选值为：4、8、16、32、64、128、256、512，单位为KB。创建精简LUN时，此项无效。	TV1
WriteType	1	Cache写方式，可选值为“1”（回写）、“2”（透写）。	全部
Prefetch Type	3	Cache预取策略，可选值为“0”（不预取）、“1”（固定预取）、“2”（可变预取）或者“3”（智能预取）。	全部
Prefetch Value	0	Cache预取值。	全部
LUNcopyWaitInterval	5	插件在启动LUN拷贝后需要不断查询LUN拷贝的拷贝进度，该值用于用户指定间隔多长时间查询一次。	TV2 V3 18000
Timeout	432000	等待存储设备执行LUN拷贝的超时时间，单位为秒（s）。	TV2 V3 18000
Initiator Name	-	计算节点启动器iqn。	全部
Initiator Target IP	-	存储系统提供给计算节点的iSCSI目标器端口IP。	全部
Initiator TargetPortGroup	-	存储系统提供给计算节点的iSCSI目标器端口组名称。	TV2 V3 18000 Dorado V3
HostIP	-	Nova compute节点的主机IP。	TV1
OSType	Linux	Nova compute节点的主机操作系统类型。	TV1

## 4.1.7 配置 iSCSI 多路径

如果需要为iSCSI配置多路径，可参考以下配置进行。

### 操作步骤

**步骤1** 通过DeviceManager界面在存储设备上创建端口组，将需要使用多路径的业务链路加入该端口组。

**步骤2** 在华为自定义的Cinder Driver配置文件中增加端口组设置，配置启动器端所需要使用的端口组名称。

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

**步骤3** 打开OpenStack Nova模块多路径开关。

如果OpenStack版本为Juno、Kilo、Liberty或Mitaka，则在Nova Compute节点“/etc/nova/nova.conf”的[libvirt]中增加“iscsi\_use\_multipath = True”  
如果OpenStack版本为Newton或Ocata，则在Nova Compute节点“/etc/nova/nova.conf”的[libvirt]中增加“volume\_use\_multipath = True”

**步骤4** 执行service nova-compute restart，重启nova-compute服务。

----结束

## 4.1.8 配置 CHAP 和 ALUA 特性

在公共网络中，与存储系统iSCSI主机端口在同一网络的应用服务器都可以访问存储系统，并对其进行读写操作。因此，数据的安全性存在隐患。为了保证存储系统的安全，可以通过配置CHAP安全性认证，限制应用服务器对存储系统的访问权限。

### 操作步骤

**步骤1** 配置CHAP和ALUA。

在Cinder Driver配置文件中进行如下配置：

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

“ALUA”表示一种多路径模式，“0”表示不启用ALUA，“1”表示启用ALUA。

“CHAPinfo”表示CHAP认证时用户自行配置的用户名和密码。格式为“mm-user;mm-user@storage1”，用户名和密码以分号隔开。

#### 说明

- TV1或V3存储都要保证主机的启动器处于空闲可用状态（即没有添加给其他主机）；
- 对于V3存储若该启动器第一次配置Chap，则必须保证阵列上该启动器没有添加给主机或者在阵列上不存在。



#### 注意

- CHAP名称长度范围是4到223位；



- 密码长度范围是12到16位;
- 密码必须包含大写字母、小写字母、数字和特殊符号（  
`~!@#\$%^&\*()-\_+=\|[{ }];:'.<.>/?和空格）中任意三者的组合;
- 密码不能和CHAP名称或者CHAP名称的倒写一样。

----结束

## 4.2 配置 FC 组网 Cinder Driver

介绍在FC协议下如何配置不同华为存储产品的Cinder Driver。

### 前提条件

FC组网时，要使用存储资源，需要安装sg工具。

若使用从镜像创建卷功能，需安装多路径软件，且在Cinder volume节点/etc/cinder/cinder.conf文件里面的指定后端里增加如下配置信息：

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

### 4.2.1 配置 T 系列 V1 FC 协议 Cinder Driver

#### 操作步骤

**步骤1** 在“/etc/cinder”路径下创建华为自定义Cinder Driver配置文件，文件格式为XML格式。

可根据实际情况修改Cinder Driver配置文件名称。例如，Cinder Driver配置文件名为“cinder\_huawei\_conf.xml”。

**步骤2** 配置华为自定义Cinder Driver配置文件参数，分为必填参数和选填参数。

必填参数：

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

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

```
<LUN>
  <LUNType>xxx</LUNType>
  <StripUnitSize>xxx</StripUnitSize>
  <WriteType>xxx</WriteType>
```

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

#### 说明

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的拥有者及用户组与“/etc/cinder/cinder.conf”文件的拥有者及用户组一致：

```
-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
```

请参考“4.2.7”来配置参数属性。

#### 步骤3 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为TV1\_FC后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT]区块中修改以下内容，配置TV1\_FC后端：

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

#### 步骤4 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

## 4.2.2 配置 T 系列 V2 FC 协议 Cinder Driver

### 操作步骤

#### 步骤1 在“/etc/cinder”路径下创建华为自定义Cinder Driver配置文件，文件格式为XML格式。

可根据实际情况修改Cinder Driver配置文件名称。例如，Cinder Driver配置文件名为“cinder\_huawei\_conf.xml”。

#### 步骤2 配置华为自定义Cinder Driver配置文件参数，分为必填参数和选填参数。

必填参数：

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
  <Storage>
    <Product>TV2</Product>
    <Protocol>FC</Protocol>
    <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
    <UserName>xxx</UserName>
    <UserPassword>xxx</UserPassword>
  </Storage>
  <LUN>
    <StoragePool>xxx</StoragePool>
  </LUN>
</config>
```

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

```
<LUN>
  <LUNType>xxx</LUNType>
```

```
<WriteType>xxx</WriteType>
<Prefetch Type="xxx" Value="xxx" />
<LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
<Timeout>432000</Timeout>
</LUN>
```

#### 说明

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的拥有者及用户组与“/etc/cinder/cinder.conf”文件的拥有者及用户组一致：

```
-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
```

请参考“4.2.7”来配置参数属性。

#### 步骤3 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为TV2\_FC后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT]区块中修改以下内容，配置TV2\_FC后端：

```
[DEFAULT]
...
enabled_backends=TV2_FC
```

#### 步骤4 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

## 4.2.3 配置 V3 FC 协议 Cinder Driver

### 操作步骤

#### 步骤1 在“/etc/cinder”路径下创建华为自定义Cinder Driver配置文件，文件格式为XML格式。

可根据实际情况修改Cinder Driver配置文件名称。例如，Cinder Driver配置文件名为“cinder\_huawei\_conf.xml”。

#### 步骤2 配置cinder.conf文件，分为必填参数和选填参数。

必填参数：

```
<?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>
    <StoragePool>xxx</StoragePool>
  </LUN>
</config>
```

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

```
<LUN>
  <LUNType>xxx</LUNType>
  <WriteType>xxx</WriteType>
  <Prefetch Type="xxx" Value="xxx" />
  <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
  <Timeout>432000</Timeout>
</LUN>
```

#### 说明

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的拥有者及用户组与“/etc/cinder/cinder.conf”文件的拥有者及用户组一致：

```
-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
```

请参考“4.2.7”来配置参数属性。

#### 步骤3 修改cinder配置文件“cinder.conf”，增加并配置下述参数。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为V3\_FC后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT]区块中修改以下内容，配置V3\_FC后端：

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

#### 步骤4 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

## 4.2.4 配置 18000 系列 FC 协议 Cinder Driver

介绍在FC协议下配置OceanStor 18000系列V1和V3存储系统Cinder Driver。

### 4.2.4.1 配置 18000 系列 V1 FC 协议 Cinder Driver

#### 操作步骤

##### 步骤1 在“/etc/cinder”路径下创建华为自定义Driver配置文件，文件格式为XML格式。

可根据实际情况修改配Driver置文件名称。例如，Driver配置文件名为“cinder\_huawei\_conf.xml”。

##### 步骤2 配置华为自定义Driver配置文件参数，分为必填参数和选填参数。

必填参数：

```
<?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>
</LUN>
```

```
<StoragePool>xxx/StoragePool>
</LUN>
</config>
```

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

```
<LUN>
  <LUNType>xxx</LUNType>
  <WriteType>xxx</WriteType>
  <Prefetch Type="xxx" Value="xxx" />
  <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
  <Timeout>432000</Timeout>
</LUN>
```

#### 说明

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的拥有者及用户组与“/etc/cinder/cinder.conf”文件的拥有者及用户组一致：

```
-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
```

参数请参考“4.2.7”来配置参数属性。

### 步骤3 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为18000V1\_FC后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT]区块中修改以下内容，配置18000V1\_FC后端：

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

### 步骤4 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

## 4.2.4.2 配置 18000 系列 V3 FC 协议 Cinder Driver

### 操作步骤

#### 步骤1 在“/etc/cinder”路径下创建华为自定义Driver配置文件，文件格式为XML格式。

可根据实际情况修改配Driver置文件名称。例如，Driver配置文件名为“cinder\_huawei\_conf.xml”。

#### 步骤2 配置华为自定义Driver配置文件参数，分为必填参数和选填参数

必填参数：

```
<?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>
  <StoragePool>xxx</StoragePool>
</LUN>
</config>
```

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

```
<LUN>
  <LUNType>xxx</LUNType>
  <WriteType>xxx</WriteType>
  <Prefetch Type="xxx" Value="xxx" />
  <LUNcopyWaitInterval>xxx</LUNcopyWaitInterval>
  <Timeout>432000</Timeout>
</LUN>
```

#### 说明

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的拥有者及用户组与“/etc/cinder/cinder.conf”文件的拥有者及用户组一致：

```
-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
```

请参考“4.2.7”来配置参数属性。

#### 步骤3 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为18000V3\_FC后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT]区块中修改以下内容，配置18000V3\_FC后端：

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

#### 步骤4 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

## 4.2.5 配置 Dorado 系列 FC 协议 Cinder Driver

### 操作步骤

#### 步骤1 在“/etc/cinder”路径下创建华为自定义Driver配置文件，文件格式为XML。

可根据实际情况修改Driver配置文件名称。例如，Driver配置文件名为“cinder\_huawei\_conf.xml”。

#### 步骤2 配置华为自定义Driver配置文件参数，分为必填参数和选填参数。

必填参数：

```
<?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>
<UserName>xxx</UserName>
<UserPassword>xxx</UserPassword>
</Storage>
<LUN>
  <LUNType>Thin</LUNType>
  <StoragePool>xxx</StoragePool>
</LUN>
</config>
```

选填参数（在自定义配置文件必填参数的基础上，在对应的位置增加需要使用的选填参数）：

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

#### 说明

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的拥有者及用户组与“/etc/cinder/cinder.conf”文件的拥有者及用户组一致：

```
-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
```

请参考“4.2.7”来配置参数属性。

#### 步骤3 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下项，为Dorado\_FC后端配置Huawei Driver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT]区块中修改以下内容，配置Dorado\_V3后端：

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

#### 步骤4 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

## 4.2.6 配置智能划域

智能划域功能为可选项，当需要使用智能划域功能时，进行如下配置。

### 操作步骤

#### 步骤1 配置cinder.conf文件。

参考<https://docs.openstack.org/ocata/config-reference/block-storage/fc-zoning.html>配置智能化域相关参数。

以Brocade交换机为例：

```
[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.BrcdFCSanLookupService
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
```

**步骤2** 执行**service cinder-volume restart**命令，重新启动Cinder服务。

----结束

## 4.2.7 配置文件参数属性说明

表 4-3 必填参数说明

属性名称	默认值	说明	适用产品
Product	-	存储产品类型，可选值为“T”、“18000”、“TV2”、“V3”和“Dorado”。	全部
Protocol	-	连接协议类型，选值为“FC”。	全部
ControllerIP0	-	T系列V100R005存储设备主控制器IP地址。	TV1
ControllerIP1	-	T系列V100R005存储设备从控制器IP地址。	TV1
RestURL	-	Rest接口访问地址。参考配置： <a href="https://x.x.x.x:8088/deviceManager/rest/">https://x.x.x.x:8088/deviceManager/rest/</a> x.x.x.x代表管理IP地址，18000系列V1参考4.2.4.1。如果需要配置多个RestURL，以分号“;”进行分割。	TV2 V3 18000 Dorado V3
UserName	-	管理员用户名。	全部
UserPassword	-	管理员密码。	全部
StoragePool	-	需要使用的存储池名称，需提前在目标阵列上创建。如果需要配置多个资源池，以分号“;”进行分割。TV1系列参考4.2.1	全部



表 4-4 选填参数说明

属性名称	默认值	说明	适用产品
LUNType	Thick（除Dorado） Thin（Dorado）	创建LUN的类型，可选值为“Thick”或者“Thin”。Dorado系列仅支持“Thin”。	全部
StripUnitSize	64	创建LUN的分条深度大小，可选值为：4、8、16、32、64、128、256、512，单位为KB。创建精简LUN时，此项无效。	TV1
WriteType	1	Cache写方式，可选值为“1”（回写）、“2”（透写）。	全部
Prefetch Type	3	Cache预取策略，可选值为“0”（不预取）、“1”（固定预取）、“2”（可变预取）或者“3”（智能预取）。	全部
Prefetch Value	0	Cache预取值。	全部
LUNcopyWaitInterval	5	插件在启动LUN拷贝后需要不断查询LUN拷贝的拷贝进度，该值用于用户指定间隔多长时间查询一次。	TV2 V3 18000
Timeout	432000	等待存储设备执行LUN拷贝的超时时间，单位为秒（s）。	TV2 V3 18000
HostIP	-	Nova compute节点的主机IP。	TV1
OSType	Linux	Nova compute节点的主机操作系统类型。	TV1

## 4.2.8 配置 FC 多路径

如果需要为FC配置多路径，可参考以下配置进行。

### 操作步骤

#### 步骤1 打开OpenStack Nova模块多路径开关。

如果OpenStack版本为Liberty或Mitaka，则在Nova Compute节点“/etc/nova/nova.conf”的[libvirt]中增加“iscsi\_use\_multipath = True”  
如果OpenStack版本为Newton或Ocata，则在Nova Compute节点“/etc/nova/nova.conf”的[libvirt]中增加“volume\_use\_multipath = True”

#### 步骤2 执行service nova-compute restart，重启nova-compute服务。

----结束

## 4.3 配置多存储支持

在“/etc/cinder/cinder.conf”中配置多个存储系统，举例如下：

```
[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 高级特性配置

---

华为存储支持系列的smartx和双活等高级特性，这些高级特性能通过与特定的卷类型关联的形式，在OpenStack环境中关联起来，实现OpenStack与存储高级特性的结合。

使用华为存储提供的某项高级特性，需提前导入特性license。

[5.1 配置SmartQoS属性](#)

[5.2 配置SmartPartition属性](#)

[5.3 配置Thick属性](#)

[5.4 配置Thin属性](#)

[5.5 配置SmartCache属性](#)

[5.6 配置SmartTier 属性](#)

[5.7 配置HyperMetro属性](#)

[5.8 配置Replication V2.1](#)

[5.9 指定Storage Pool创建LUN](#)

[5.10 指定Disk Type创建LUN](#)

[5.11 指定Volume Backend创建LUN](#)

[5.12 配置Consistency Groups](#)

[5.13 配置Backup Snapshot](#)

[5.14 配置证书认证](#)

## 5.1 配置 SmartQoS 属性

### 关于本章

OpenStack的QoS特性主要依赖于前端的Hypervisor和后端存储来实现，华为OpenStack Cinder Driver同时支持“前端QoS”和“后端QoS”特性。前端QoS包含以下选项：

total\_bytes\_sec、read\_bytes\_sec、total\_iops\_sec、read\_iops\_sec、write\_iops\_sec

华为存储支持如下QoS属性，这些属性可单个或多个同时关联到一个QoS属性中。

保护策略：latency、minIOPS、minBandWidth

限制策略：maxIOPS、maxBandWidth



小心

保护策略和限制策略为一对互斥的特性，如果同时配置会导致创建卷失败。

QoS属性中，IOType为必选项，如果QoS策略中未配置IOType会导致创建卷失败。

### 5.1.1 配置前端 QoS

介绍配置前端QoS的操作步骤。

#### 操作步骤

**步骤1** 执行**cinder type-create XXX**命令，创建卷类型。XXX代表卷类型名称，用户可任意指定。

```
root@ubuntu:/# cinder type-create high-iops
```

ID	Name	Description	Is_Public
84b189d3-8984-4e92-aab0-fa4a913126bf	high-iops	-	True

**步骤2** 执行**cinder qos-create xxx consumer="front-end" read\_iops\_sec=xxx write\_iops\_sec=xxx**命令，创建前端QoS控制属性参数。

以“**read\_iops\_sec=2000 write\_iops\_sec=1000**”为例进行说明。

```
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' }

**步骤3** 建立卷类型和QoS控制属性关联。

1. 查看关联命令**cinder qos-associate**用法。

关联命令格式为：**cinder qos-associate <qos控制/属性ID> <卷类型ID>**。

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

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

----结束

## 操作结果

- 执行完以后可以通过**cinder qos-list**命令查看QoS策略配置详情。

```
root@lcl404:~# cinder qos-list
+-----+-----+-----+-----+
| ID                               | Name      | Consumer |
| specs                            |           |           |
+-----+-----+-----+-----+
| 7dc73b5b-1b19-4371-8dee-8edbc52cf625 | high-iops | front-end | {'write_iops_sec':
| '1000', 'read_iops_sec': '2000'} |           |           |
+-----+-----+-----+-----+
```

### 说明

qos中，配置consumer="front-end"将会创建前端QoS，未配置consumer时，consumer默认为“back-end”，即后端QoS。

## 5.1.2 QoS 配置版本说明

表 5-1 支持同时配置多 QoS 属性存储系统

存储产品型号	存储系统版本
18000	V300R003C00 V300R006C00
V3	V300R003C00/C10/C20 V300R006C00 2200 V300R005C00 2600 V300R005C00 2200 V300R006C00 2600 V300R006C00

## 5.1.3 配置控制 IOPS

包括配置控制最大IOPS和配置控制最小IOPS。

### 5.1.3.1 配置控制最大 IOPS

介绍配置控制最大IOPS操作步骤。

## 操作步骤

**步骤1** 执行**cinder type-create XXX**命令，创建卷类型。XXX代表卷类型名称，用户可任意指定。

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

ID	Name	Description	Is_Public
3d9cc52e-069b-4245-b201-945e0ef571cf	maxIOPS	-	True

**步骤2** 执行以下命令，配置SmartQos属性为“true”的键值对。

```
root@ubuntu:~# cinder type-key maxIOPS set capabilities:QoS_support='<is> true'
```

**步骤3** 执行**cinder qos-create xxx maxIOPS=xxx IOType=xxx**命令，创建QoS控制属性参数。

- 第一个xxx表示已创建的QoS控制属性名称。
- maxIOPS：最大IOPS限制，数值为大于0的整数。
- IOType：控制读写类型。“0”表示控制读IO；“1”表示控制写IO；“2”表示控制读写IO。

以“**maxIOPS=100 IOType=2**”为例进行说明。

```
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' }

**步骤4** 建立卷类型和QoS控制属性关联。

1. 查看关联命令**cinder qos-associate**用法。

关联命令格式为：**cinder qos-associate <qos控制属性ID> <卷类型ID>**。

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

2. 执行命令**cinder qos-associate 1f772258-49f0-47a9-aa9c-d8f32d844bb1 3d9cc52e-069b-4245-b201-945e0ef571cf**。

----结束

### 5.1.3.2 配置控制最小 IOPS

介绍配置控制最小IOPS操作步骤。

## 操作步骤

**步骤1** 执行**cinder type-create XXX**命令，创建卷类型。XXX代表卷类型名称，用户可任意指定。

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

ID	Name	Description	Is_Public
1a1dcee8-d9ae-4de9-b1c6-2317a4ad7219	minIOPS	-	True

**步骤2** 执行以下命令，配置SmartQos属性为“true”的键值对。

```
root@ubuntu:~# cinder type-key minIOPS set capabilities:QoS_support='<is> true'
```

**步骤3** 执行 **cinder qos-create xxx minIOPS=xxx IOType=xxx** 命令，创建QoS控制属性参数。

- 第一个xxx表示已创建的QoS控制属性名称。
- minIOPS：最小IOPS限制，数值为大于0的整数。
- IOType：控制读写类型。“0”表示控制读IO；“1”表示控制写IO；“2”表示控制读和写IO。

以“**minIOPS=50 IOType=2**”为例进行说明。

```
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' }

**步骤4** 建立卷类型和QoS控制属性关联。

1. 查看关联命令**cinder qos-associate**用法。

关联命令格式为：**cinder qos-associate <qos控制属性ID> <卷类型ID>**。

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

2. 执行命令**cinder qos-associate 8deaf13c-bf1c-4a71-94d0-1149805693ce 1a1dcee8-d9ae-4de9-b1c6-2317a4ad7219**。

----结束

## 5.1.4 配置控制带宽

包括配置控制最大带宽和配置控制最小带宽。

### 5.1.4.1 配置控制最大带宽

介绍配置控制最大带宽的操作步骤。

#### 操作步骤

**步骤1** 执行**cinder type-create XXX**命令，创建卷类型。XXX代表卷类型名称，用户可任意指定。

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

ID	Name	Description	Is_Public
24200fbb-c984-4d99-9465-a820464662d6	maxBandWidth	-	True

**步骤2** 执行以下命令，配置SmartQos属性为“true”的键值对。

```
root@ubuntu:~# cinder type-key maxBandWidth set capabilities:QoS_support='<is> true'
```

**步骤3** 执行**cinder qos-create xxx maxBandWidth=xxx IOType=xxx**命令，创建QoS控制属性参数。

- 第一个xxx表示已创建的QoS控制属性名称。
- maxBandWidth: 最大带宽限制, 数值为大于0的整数, 单位MB/s。
- IOType: 控制读写类型。“0”表示控制读IO; “1”表示控制写IO; “2”表示控制读和写IO。

以“maxBandWidth=100 IOType=2”为例进行说明。

```
root@ubuntu:/# cinder qos-create maxbandwidth maxBandWidth=100 IOType=2
```

property	value
consumer	back-end
id	2f51c955-e029-48ca-aa8c-5d7ae462481e
name	maxbandwidth
specs	{u' maxBandWidth': u'100', u' IOType': u'2' }

**步骤4** 建立卷类型和QoS控制属性关联。

1. 查看关联命令**cinder qos-associate**用法。

关联命令格式为: **cinder qos-associate** <qos控制/属性ID> <卷类型ID>。

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

2. 执行命令**cinder qos-associate 2f51c955-e029-48ca-aa8c-5d7ae462481e 24200fbb-c984-4d99-9465-a820464662d6**。

----结束

### 5.1.4.2 配置控制最小带宽

介绍配置控制最小带宽的操作步骤。

## 操作步骤

**步骤1** 执行**cinder type-create XXX**命令, 创建卷类型。XXX代表卷类型名称, 用户可任意指定。

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

ID	Name	Description	Is_Public
271bed0a-1cce-4e8f-a65e-b85ccb3b9a25	minBandWidth	-	True

**步骤2** 执行以下命令, 配置SmartQos属性为“true”的键值对。

```
root@ubuntu:~# cinder type-key minBandWidth set capabilities:QoS_support='<is> true'
```

**步骤3** 执行**cinder qos-create xxx minBandWidth=xxx IOType=xxx**命令, 创建QoS控制属性参数。

- 第一个xxx表示已创建的QoS控制属性名称。
- minBandWidth: 最小带宽限制, 数值为大于0的整数, 单位MB/s。
- IOType: 控制读写类型。“0”表示控制读IO; “1”表示控制写IO; “2”表示控制读和写IO。

以“minBandWidth= 100 IOType= 2”为例进行说明。



```
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' }

**步骤4** 建立卷类型和QoS控制属性关联。

1. 查看关联命令**cinder qos-associate**用法。

关联命令格式为：**cinder qos-associate <qos控制属性ID> <卷类型ID>**。

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

2. 执行命令**cinder qos-associate 59c583d8-69d2-4c39-8c75-c1b21ef85f2e 271bed0a-1cce-4e8f-a65e-b85ccb3b9a25**。

----结束

## 5.1.5 配置控制时延

介绍配置控制时延的操作步骤。

### 操作步骤

**步骤1** 执行**cinder type-create XXX**命令，创建类型。XXX代表卷类型名称，用户可任意指定。

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

ID	Name	Description	Is_Public
ae34870a-21c0-4e50-8148-1e26f6ab6eab	latency	-	True

**步骤2** 执行以下命令，配置SmartQos属性为“true”的键值对。

```
root@ubuntu:/# cinder type-key latency set capabilities:QoS_support='<is> true'
```

**步骤3** 执行**cinder qos-create xxx latency=xxx IOType=xxx**命令，创建QoS控制属性参数。

- 第一个xxx表示已创建的QoS控制属性名称。
- latency: 时延限制，数值为大于0的整数，单位为ms。
- IOType: 控制读写类型。“0”表示控制读IO；“1”表示控制写IO；“2”表示控制读和写IO。

以“**latency=100 IOType=2**”为例进行说明。

```
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' }

**步骤4** 建立卷类型和QoS控制属性关联。

1. 查看关联命令**cinder qos-associate**用法。

关联命令格式为：**cinder qos-associate** <qos控制属性ID> <卷类型ID>。

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

2. 执行命令**cinder qos-associate 6f305e9f-698b-4e9e-aa68-8efb80b43036 ae34870a-21c0-4e50-8148-1e26f6ab6eab**。

----结束

## 5.1.6 配置控制多策略

介绍配置控制多策略的操作步骤。

### 操作步骤

- 步骤1** 执行**cinder type-create XXX**命令，创建类型。XXX代表卷类型名称，用户可任意指定。

```
root@ubuntu:/# cinder type-create multi-strategy
```

ID	Name	Description	Is_Public
df53d9d7-b1db-4e6b-847a-7a5150c39489	multi-strategy	-	True

- 步骤2** 执行以下命令，配置SmartQos属性为“true”的键值对。

```
root@ubuntu:~# cinder type-key multi-strategy set capabilities:QoS_support='<is> true'
```

- 步骤3** 执行**cinder qos-create xxx latency=xxx minBandWidth=xxx IOType=xxx**命令，创建QoS控制属性参数。

- 第一个xxx表示已创建的QoS控制属性名称。
- latency: 时延限制，数值为大于0的整数，单位为ms。
- minBandWidth: 最小带宽限制，数值为大于0的整数，单位MB/s。
- IOType: 控制读写类型。“0”表示控制读IO；“1”表示控制写IO；“2”表示控制读和写IO。

以“**latency=100 minBandWidth=100 IOType=2**”为例进行说明。

```
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', 'latency': '100'}

- 步骤4** 建立卷类型和QoS控制属性关联。

1. 查看关联命令**cinder qos-associate**用法。

关联命令格式为：**cinder qos-associate** <qos控制属性ID> <卷类型ID>。

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

2. 执行命令 **cinder qos-associate 36f0fc4a-d5f5-4862-ab1f-e7fedeeddc41 df53d9d7-b1db-4e6b-847a-7a5150c39489**。

----结束

## 操作结果

- 执行完毕后可以通过 **cinder qos-list** 命令查看QoS策略配置详情。

```
root@ubuntu:~# cinder qos-list
```

ID	specs	Name	Consumer
36f0fc4a-d5f5-4862-ab1f-e7fedeeddc41	multi-strategy	back-end	{ 'IOType': '2', 'minBandWidth': '100', 'latency': '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	back-end	{ 'latency': '100', 'IOType': '2' }
9b34fbdf-91b5-473d-a5ff-0b4ea2ed9fd0	maxiops	back-end	{ 'IOType': '2', 'maxIOPS': '100' }
f60a3374-cdf3-45cc-86c4-12c06a3db8d4	minbandwidth	back-end	{ 'IOType': '2', 'minBandWidth': '100' }

- 执行完毕后可以通过 **cinder type-list** 命令查看设置的卷类型。

```
root@ubuntu:~# cinder type-list
```

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

## 5.2 配置 SmartPartition 属性

以举例的方式，介绍配置SmartPartition属性的操作步骤。

### 操作步骤

- 步骤1** 执行 **cinder type-create XXX** 命令，创建卷类型。XXX代表卷类型名称，由用户指定。

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

ID	Name	Description	Is_Public
a7648f77-fb56-41f2-9d50-3c014534ef66	SmartPartition	-	True

- 步骤2** 执行以下命令，配置SmartPartition属性为“true”的键值对。

```
root@ubuntu:~# cinder type-key SmartPartition set capabilities:smartpartition='<is> true'
```

- 步骤3** 配置存储设备上已经存在的SmartPartition的名称，将partitionname关联到卷类型中。

```
root@ubuntu:~# cinder type-key SmartPartition set smartpartition:partitionname='test_partition'
```

**步骤4** 效果展示，创建一个带SmartPartition属性的卷。

```
root@ubuntu-204:~# cinder create 1 --display-name test001 --volume-type SmartPartition
```

property	value
attachments	[]
availability_zone	nova
bootable	false
created_at	2015-06-26t13:40:10.483450
display_description	none
display_name	test001
encrypted	false
id	1b2a87e4-eb8c-46ab-922c-63fcd7db8f20
metadata	{}
multiattach	false
size	1
snapshot_id	none
source_vol_id	none
status	creating
volume_type	SmartPartition

----结束

## 5.3 配置 Thick 属性

以举例的方式，介绍配置Thick属性的操作步骤。



**小心**

Dorado V3不支持创建Thick属性的卷。

### 操作步骤

**步骤1** 执行**cinder type-create XXX**命令，创建卷类型。XXX代表卷类型名称，由用户指定。

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

ID	Name	Description	Is_Public
a7648f77-fb56-41f2-9d50-3c014534ef66	Thick	-	True

**步骤2** 执行**cinder type-key Thick set capabilities:thick\_provisioning\_support='<is> true'**命令，设置Thick卷类型键值对。

**步骤3** 执行**cinder type-key Thick set provisioning:type='thick'**命令，设置创建Thick卷时调度器的调度机制。

**步骤4** 效果展示，创建一个Thick属性卷。

```
root@ubuntu-204:~# cinder create 1 --display-name test001 --volume-type Thick
```

property	value
attachments	[]
availability_zone	nova
bootable	false
created_at	2015-06-26t13:40:10.483450
display_description	none

display_name	test001
encrypted	false
id	1b2a87e4-eb8c-46ab-922c-63fcd7db8f20
metadata	{}
multiattach	false
size	1
snapshot_id	none
source_volid	none
status	creating
volume_type	Thick

----结束

## 5.4 配置 Thin 属性

以举例的方式，介绍配置Thin属性的操作步骤。

### 操作步骤

**步骤1** 执行**cinder type-create XXX**命令，创建卷类型。XXX代表卷类型名称，由用户指定。

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

ID	Name	Description	Is_Public
ba648f77-fb56-41f2-9d50-3c014534eedd	Thin	-	True

**步骤2** 执行**cinder type-key Thin set capabilities:thin\_provisioning\_support='<is> true'**命令，设置Thin卷类型键值对。

**步骤3** 效果展示，创建一个Thin属性的卷。

```
root@ubuntu-204:~# cinder create 1 --display-name test001 --volume-type Thin
```

property	value
attachments	[]
availability_zone	nova
bootable	false
created_at	2015-06-26t13:40:10.483450
display_description	none
display_name	test001
encrypted	false
id	1b2a87e4-eb8c-46ab-922c-63fcd7db8f20
metadata	{}
multiattach	false
size	1
snapshot_id	none
source_volid	none
status	creating
volume_type	Thin

----结束

## 5.5 配置 SmartCache 属性

以举例的方式，介绍配置SmartCache属性的操作步骤。

## 操作步骤

**步骤1** 执行**cinder type-create XXX**命令，创建卷类型。XXX代表卷类型名称，由用户指定。

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

ID	Name	Description	Is_Public
a7648f77-fb56-41f2-9d50-3c014534ef66	SmartCache	-	True

**步骤2** 执行以下命令，配置SmartCache属性为“true”的键值对。

```
root@ubuntu:~#cinder type-key SmartCache set capabilities:smartcache='<is> true'
```

**步骤3** 配置存储设备上已经存在的SmartCache名称，将cachename关联到卷类型中。

```
root@ubuntu:~#cinder type-key SmartCache set smartcache:cachename='test_cache'
```

**步骤4** 效果展示，创建一个带SmartCache属性的卷。

```
root@ubuntu-204:~# cinder create 1 --display-name test001 --volume-type Smartcache
```

property	value
attachments	[]
availability_zone	nova
bootable	false
created_at	2015-06-26t13:40:10.483450
display_description	none
display_name	test001
encrypted	false
id	1b2a87e4-eb8c-46ab-922c-63fcd7db8f20
metadata	{}
multiattach	false
size	1
snapshot_id	none
source_volid	none
status	creating
volume_type	SmartCache

----结束

## 5.6 配置 SmartTier 属性

以举例的方式，介绍配置SmartTier属性的操作步骤。

### 操作步骤

**步骤1** 执行**cinder type-create XXX**命令，创建卷类型。XXX代表卷类型名称，由用户指定。

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

ID	Name	Description	Is_Public
fb3ec327-84f2-4af9-acc9-744dd6163dcf	SmartTier	-	True

**步骤2** 执行以下命令，设置SmartTier卷类型键值对。

```
root@ubuntu:~#cinder type-key SmartTier set capabilities:smarttier='<is> true'
```

**步骤3** 配置SmartTier迁移策略

```
root@ubuntu:~#cinder type-key SmartTier set smarttier:policy='xx'
```

**Policy:** 表示迁移策略。“0”表示不迁移，“1”表示自动迁移，“2”表示向高性能迁移，“3”表示向低性能迁移。

**步骤4** 效果展示，创建一个带SmartTier属性的卷。

```
root@ubuntu-204:~# cinder create 1 --display-name test001 --volume-type Smarttier
```

property	value
attachments	[]
availability_zone	nova
bootable	false
created_at	2015-06-26t13:40:10.483450
display_description	none
display_name	test001
encrypted	false
id	1b2a87e4-eb8c-46ab-922c-63fcd7db8f20
metadata	{}
multiattach	false
size	1
snapshot_id	none
source_vol_id	none
status	creating
volume_type	SmartTier

----结束

## 5.7 配置 HyperMetro 属性

以举例的方式，介绍配置HyperMetro属性的操作步骤。

### 前提条件

**步骤1** 在“/etc/cinder/cinder.conf”的指定后端中配置远端设备信息。

```
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
```

双活远端使用iscsi多路径，在远端配置信息里面增加如下配置：

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



### 注意

各关键字段之间用“，”分割，“iscsi\_info”中多条启动器信息，用“；”分割。

表 5-2 参数含义参考下表属性说明：

属性名称	默认值	说明	适用产品及版本
san_address	--	Rest接口访问地址。参考配置： <a href="https://x.x.x.x:8088/deviceManager/rest/">https://x.x.x.x:8088/deviceManager/rest/</a> ， x.x.x.x 代表管理IP地址	V3R3 2600 V3R5 18000 V3R3
san_user	--	双活远端存储管理员用户名	V3R3 2600 V3R5 18000 V3R3
san_password	--	双活远端存储管理员密码	V3R3 2600 V3R5 18000 V3R3
domain_name	--	通过存储管理界面（DeviceManager）上配置的域名名称	V3R3 2600 V3R5 18000 V3R3
storage_pools	--	双活远端资源池名称	V3R3 2600 V3R5 18000 V3R3
iscsi_default_target_ip	--	双活远端业务端口IP	V3R3 2600 V3R5 18000 V3R3
Initiator Name	-	计算节点启动器iqn。	V3R3 2600 V3R5 18000 V3R3
Initiator Target IP	-	存储系统提供给计算节点的iSCSI目标器端口IP。	V3R3 2600 V3R5 18000 V3R3
Initiator TargetPortGroup	-	存储系统提供给计算节点的iSCSI目标器端口组名称。	V3R3 2600 V3R5 18000 V3R3

**步骤2** 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

## 操作步骤

**步骤1** 执行cinder type-create XXX命令，创建卷类型。XXX代表卷类型名称，由用户指定。

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



ID	Name	Description	Is_Public
0913e3dd-8b68-4b46-80ea-cf255617b13a	HyperMetro	-	True

**步骤2** 执行 **cinder type-key HyperMetro set capabilities:hypermetro='<is> true'**命令，设置 HyperMetro 卷类型键值对。

**步骤3** 执行创卷命令，创建双活卷，其中 **volume-type** 指定步骤1中定义的名称。

```
root@ubuntu-204:~# cinder create 1 --display-name test001 --volume-type Hypermetro
```

property	value
attachments	[]
availability_zone	nova
bootable	false
created_at	2015-06-26t13:40:10.483450
display_description	none
display_name	test001
encrypted	false
id	1b2a87e4-eb8c-46ab-922c-63fcd7db8f20
metadata	{}
multiattach	false
size	1
snapshot_id	none
source_vol_id	none
status	creating
volume_type	HyperMetro

----结束

## 5.8 配置 Replication V2.1

以举例的方式，介绍配置 Replication V2.1 的操作步骤。

### 前提条件

**步骤1** 在 “/etc/cinder/cinder.conf” 的指定后端中配置复制远端设备信息。

```
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
```

复制远端使用 **iscsi** 多路径，在远端配置信息里面增加如下配置：

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



### 注意

各关键字段之间用 “，” 分割，“iscsi\_info” 中多条启动器信息，用 “；” 分割。

表 5-3 参数含义参考下表属性说明：

属性名称	默认值	说明	适用产品及版本
backend_id	--	目标设备id，failover时需要指定的远端设备	TV2、V3、18000、Dorado V3
storage_pool	--	复制failover目标后端存储池名称。	TV2、V3、18000、Dorado V3
san_address	--	Rest接口访问地址。参考配置： <a href="https://x.x.x.x:8088/deviceManager/rest/x.x.x.x">https://x.x.x.x:8088/deviceManager/rest/x.x.x.x</a> 代表管理IP地址，18000系列V1参考4.1.4.1或4.2.4.1。	TV2、V3、18000、Dorado V3
san_user	--	复制远端存储管理员用户名	TV2、V3、18000、Dorado V3
san_password	--	复制远端存储管理员密码	TV2、V3、18000、Dorado V3
iscsi_default_target_ip	--	远端业务端口IP	TV2、V3、18000、Dorado V3
Initiator Name	-	计算节点启动器iqn。	TV2、V3、18000、Dorado V3
Initiator Target IP	-	存储系统提供给计算节点的iSCSI目标器端口IP。	TV2、V3、18000、Dorado V3
Initiator TargetPortGroup	-	存储系统提供给计算节点的iSCSI目标器端口组名称。	TV2、V3、18000、Dorado V3

**步骤2** 执行service cinder-volume restart命令，重新启动Cinder服务。

----结束

## 操作步骤

**步骤1** 执行cinder type-create XXX命令，创建卷类型。XXX代表卷类型名称，由用户指定。

```
root@ubuntu-204:~# cinder type-create Replication_V2.1
```

ID	Name	Description	Is_Public
b9ff6a0d-7581-4e50-8ba9-0ab2703e42e7	Replication_V2.1	-	True

**步骤2** 执行cinder type-key Replication\_V2.1 set capabilities:replication\_enabled='<is> true'命令，设置Replication V2.1卷类型键值对。

**步骤3** （可选）执行 **cinder type-key Replication\_V2.1 set replication\_type='<in> sync'** 命令，设置复制类型。

**replication\_type** 可选值为 “sync”、“async”，其中 “sync” 代表同步复制，“async” 代表异步复制。未配置时，默认为异步复制。

**步骤4** 效果展示，创建一个复制卷。

```
root@ubuntu-204:~# cinder create 1 --display-name test001 --volume-type Replication_V2.1
```

property	value
attachments	[]
availability_zone	nova
bootable	false
created_at	2015-06-26t13:42:10.483450
display_description	none
display_name	test001
encrypted	false
id	1b2a87e4-eb8c-46ab-922c-63fcd7db8f31
metadata	{}
multiattach	false
size	1
snapshot_id	none
source_vol_id	none
status	creating
volume_type	Replication_V2.1

----结束

## 5.9 指定 Storage Pool 创建 LUN

以举例的方式，介绍配置指定存储池创建卷操作步骤。

### 操作步骤

**步骤1** 执行 **cinder type-create XXX** 命令，创建卷类型。XXX代表卷类型名称，由用户指定。

```
root@ubuntu-175:~# cinder type-create target-pool
```

ID	Name	Description	Is_Public
2ecf5589-507e-4ee5-995f-4aald0aaff83	target-pool	-	True

**步骤2** 配置目标存储池。

- 配置单个目标存储池

```
cinder type-key target-pool set pool_name=StoragePool001
```

- 配置多个目标存储池

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

**步骤3** 效果展示，指定存储池创建卷。

```
root@ubuntu-175:~# cinder create --volume-type target-pool 1
```

Property	Value
attachments	[]
availability_zone	nova
bootable	false
consistencygroup_id	None
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_vol_id	None
status	creating
updated_at	None
user_id	ec5b1bd5fa8646109381755037aacc82
volume_type	target-pool

----结束

## 5.10 指定 Disk Type 创建 LUN

以举例的方式，介绍配置指定存储池磁盘类型创建卷操作步骤。

### 操作步骤

**步骤1** 执行**cinder type-create XXX**命令，创建卷类型。XXX代表卷类型名称，由用户指定。

```
root@ubuntu-175:~# cinder type-create disk-type
```

ID	Name	Description	Is_Public
c47fffc4-3510-4c24-b4ac-1880d44f12ff	disk-type	-	True

**步骤2** 配置磁盘类型。

- 配置一种磁盘类型

```
root@ubuntu-175:~# cinder type-key disk-type set disk_type=sas
```

- 配置多种磁盘类型

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

**disk\_type**可选值为“ssd”、“sas”、“nl\_sas”和“mix”，其中“mix”是由“ssd”、“sas”、“nl\_sas”中的2种及以上混合。

**步骤3** 效果展示，指定磁盘类型创建卷。

```
root@ubuntu-175:~# cinder create --volume-type disk-type 1
```

Property	Value
attachments	[]
availability_zone	nova
bootable	false
consistencygroup_id	None
created_at	2016-08-12T08:24:09.000000
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

----结束

## 5.11 指定 Volume Backend 创建 LUN

以举例的方式，介绍配置指定后端创建卷操作步骤。

### 操作步骤

**步骤1** 执行 **cinder type-create XXX** 命令，创建卷类型。XXX代表卷类型名称，由用户指定。

```
root@ul404:~# cinder type-create volume-backend
```

ID	Name	Description	Is_Public
97cb1474-6019-41fc-a07c-0eefb19ecb30	volume-backend	-	True

**步骤2** 配置卷后端名称并查看。

```
root@ul404:~# cinder type-key volume-backend set volume_backend_name=240_sh
```

```
root@ul404:~# cinder extra-specs-list
```

ID	Name	extra_specs
97cb1474-6019-41fc-a07c-0eefb19ecb30	volume-backend	{'volume_backend_name': '240_sh'}

**步骤3** 效果展示，指定后端创建卷。

```
root@ul404:~# cinder create --volume-type volume-backend 1
```

Property	Value
attachments	[]
availability_zone	nova
bootable	false
consistencygroup_id	None
created_at	2017-04-17T19:36:11.000000
description	None
encrypted	False
id	01d1df8c-45f8-4af3-8ef6-62a708233474
metadata	{}
migration_status	None
multiattach	False
name	None
os-vol-host-attr:host	ul404@240_sh#Cinder-HXL
os-vol-mig-status-attr:migstat	None
os-vol-mig-status-attr:name_id	None
os-vol-tenant-attr:tenant_id	3692e4d8455741a8b0c47a6859f23e1c
replication_status	disabled

size	1
snapshot_id	None
source_volid	None
status	creating
updated_at	2017-04-17T19:36:12.000000
user_id	3fdf4c2ee2964e23bb09decdeb24caa5
volume_type	volume-backend

----结束

## 5.12 配置 Consistency Groups

介绍配置一致性组和快照一致性组。

使用一致性组及快照一致性组功能前，需要将api接口打开（/etc/cinder/policy.json），详见<http://docs.openstack.org/admin-guide/blockstorage-consistency-groups.html>。

## 5.13 配置 Backup Snapshot

介绍配置备份快照。

### 操作步骤

**步骤1** 配置备份driver。

参考社区文档配置备份driver（如nas、swift、ceph），详见<https://docs.openstack.org/ocata/config-reference/block-storage/backup-drivers.html>，并确保备份服务cinder-backup正常启动；

**步骤2** 配置cinder.conf文件。

在cinder.conf文件的[DEFAULT]区块中增加如下配置：

```
backup_use_same_host = True
```

在对应后端区块增加如下配置：

```
backup_use_temp_snapshot = True
```

**步骤3** 重启cinder-volume和cinder-backup服务。

----结束

## 5.14 配置证书认证

介绍配置证书认证方式连接存储。

### 操作步骤

**步骤1** 配置华为自定义Driver配置文件。

在<Storage>配置段中增加SSLCertVerify和SSLCertPath配置：

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

```
        <SSLCertPath>xxx</SSLCertPath>
        ...
    </Storage>
    ...
</config>
```

SSLCertVerify表示是否打开证书认证，有效值为True或False。未显示配置默认False。

SSLCertPath指定用于认证的证书路径，仅在SSLCertVerify为True的情况下有效。

**步骤2** 重启cinder-volume服务。

----结束

# 6 最佳实践

---

## 6.1 快速对接华为存储



## 6.1 快速对接华为存储

### 配置流程

配置流程介绍了在OpenStack上通过配置OpenStack Huawei Cinder Driver对接华为存储系统配置步骤。

#### 步骤1 获取Cinder Driver

OpenStack社区Kilo版本后已自带华为存储OpenStack Driver，也可通过华为自有OpenStack Driver仓库获取最新的Driver代码，详见3。

##### 说明

OpenStack社区版本发布后，不允许合入新的特性，bug修复周期长且存在风险。

华为自有OpenStack Driver仓库版本，在框架支持的前提下能够保证新特性及时发布，bug及时修复，相较社区版本更稳定完善。

OpenStack社区仅维护两个稳定版本，华为自有OpenStack Driver仓库维护六个稳定版本，能够保障历史版本长期稳定运行。

**强烈推荐使用华为自有OpenStack Driver仓库版本替换社区版本。**

#### 步骤2 配置cinder.conf文件和华为自定义Driver配置文件（此处以iSCSI组网对接V3系列存储为例，详见4）。

- 在“/etc/cinder”路径下创建华为自定义Driver配置文件，文件格式为XML。可根据实际情况修改Driver配置文件名称。例如，Driver配置文件名为“cinder\_huawei\_conf.xml”。

- 配置华为自定义Driver配置文件参数。

```
<?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>
    <StoragePool>xxx</StoragePool>
  </LUN>
  <iSCSI>
    <DefaultTargetIP>x.x.x.x</DefaultTargetIP>
  </iSCSI>
</config>
```

- 检查XML文件所有者及用户组。

确保“/etc/cinder/cinder\_huawei\_conf.xml”文件的所有者及用户组与“/etc/cinder/cinder.conf”文件的所有者及用户组一致：

```
-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
```

- 配置cinder.conf文件。

在“/etc/cinder/cinder.conf”文件的最后添加如下配置项，为V3\_iSCSI后端配置HuaweiDriver，其中volume\_driver表示加载的Driver文件，cinder\_huawei\_conf\_file表示指定的华为自定义配置文件：

```
[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
```

在[DEFAULT] 区块中修改以下内容，配置V3\_iSCSI后端：

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

**步骤3** 执行service cinder-volume restart命令，重新启动Cinder服务。

**步骤4** 检查服务状态。

```
root@u1404:~# cinder service-list
```

Binary	Host	Zone	Status	State	Updated_at
Disabled Reason					
cinder-volume	u1404@38_sd	nova	enabled	down	2016-11-17T15:26:32.000000
cinder-volume	u1404@V3_iSCSI	nova	enabled	up	2017-04-18T15:44:09.000000
cinder-volume	u1404@dorado	nova	enabled	down	2017-03-17T11:15:43.000000

其中u1404@V3\_iSCSI的服务状态为up，代表服务已正常启动。

----结束