

OpenStack Cinder Driver

配置指南

文档版本 01

发布日期 2018-08-07



版权所有 © 华为技术有限公司 2018。 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

客户服务电话: 4008302118

目 录

1 简介	1
2 版本配套关系	2
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	
4 Cinder Driver 基本属性配置	8
4.1 iSCSI 组网 Cinder Driver 配置	8
4.1.1 配置 T 系列 V1 iSCSI 协议 Cinder Driver	8
4.1.2 配置 T 系列 V2 iSCSI 协议 Cinder Driver	10
4.1.3 配置 V3/V5 iSCSI 协议 Cinder Driver	11
4.1.4 配置 18000 系列 iSCSI 协议 Cinder Driver	12
4.1.5 配置 Dorado 系列 iSCSI 协议 Cinder Driver	13
4.1.6 配置文件参数属性说明	14
4.1.7 配置 iSCSI 多路径	16
4.1.8 配置 CHAP 和 ALUA 特性	17
4.2 配置 FC 组网 Cinder Driver	18
4.2.1 配置 T 系列 V1 FC 协议 Cinder Driver	18
4.2.2 配置 T 系列 V2 FC 协议 Cinder Driver	19
4.2.3 配置 V3/V5 FC 协议 Cinder Driver	20
4.2.4 配置 18000 系列 FC 协议 Cinder Driver	21
4.2.5 配置 Dorado 系列 FC 协议 Cinder Driver	
4.2.6 配置智能划域	23
4.2.7 配置文件参数属性说明	24
4.2.8 配置 FC 多路径	25
4.2.9 配置 ALUA 特性	
4.3 配置多存储支持	26
4.4 配置租户用户	27
5 Cinder Driver 高级特性配置	
5.1 配置 SmartQoS 属性	29
5.1.1 配置前端 QoS	
5.1.2 配置控制最大 IOPS	30

5.1.3 配置控制最小 IOPS	31
5.1.4 配置控制最大带宽	32
5.1.5 配置控制最小带宽	33
5.1.6 配置控制时延	34
5.1.7 配置控制多策略	35
5.2 配置 SmartPartition 属性	36
5.3 配置 Thick 属性	37
5.4 配置 Thin 属性	38
5.5 配置 SmartCache 属性	39
5.6 配置 SmartTier 属性	39
5.7 配置 HyperMetro 属性	40
5.8 配置 Replication V2.1	42
5.9 指定 Storage Pool 创建 LUN	44
5.10 指定 Disk Type 创建 LUN	44
5.11 指定 Volume Backend 创建 LUN	45
5.12 配置 Consistency Groups	46
5.13 配置 Backup Snapshot.	46
5.14 配置证书认证	47
5.15 配置 LUN 归属控制器	47
5.16 配置 SmartDedupe 属性	48
5.17 配置 SmartCompression 属性	49
6 最佳实践	51
6.1 快速对接华为存储	
7 FAQ	53
7.1 通过 Cinder Driver 映射卷是否需要事先在华为存储上创建主机、主机组、LUN 组等映射需要	更的关联对象?
	53
7.2 映射卷时 Nova 日志中打印 lsblk 命令执行失败导致映射失败。	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
Liberty	T系列 V100R005/V200R002
Mitaka	OceanStor V3 V300R001/V300R002/V300R003/V300R006
Newton	OceanStor 2200/2600 V3 V300R005
Ocata	OceanStor V5 V500R007
Pike	OceanStor 18500/18800 V100R001/V300R003/V300R006/
Queens	V500R007
	OceanStor Dorado V3 V300R001
	OceanStor 6800F/18500F/18800F V3高端全闪存存储系统
	OceanStor 2600F/5500F/5600F/5800F V3全闪存存储系统

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

特性	Liber ty	Mita ka	Newt on	Ocata	Pike	Quee ns
Create Volume	√	√	√	√	√	√
Delete Volume	√	√	√	√	√	√
Attach Volume	√	√	√	√	√	√
Detach Volume	√	√	√	√	√	√
Extend Volume	√	√	√	√	√	√
Create Snapshot	√	√	√	√	√	√
Delete Snapshot	√	√	√	√	√	√

特性	Liber ty	Mita ka	Newt on	Ocata	Pike	Quee ns
Create Volume from Snapshot	√	√	√	√	√	√
Create Volume from Image	√	√	√	√	√	√
Create Volume from Volume	√	√	√	√	√	√
Create Image from Volume	√	√	√	√	√	√
Volume Migration	√	√	√	√	√	√
QoS	√	√	√	√	√	√
Auto zoning	√	√	√	√	√	√
SmartTier	√	√	√	√	√	√
SmartCache	√	√	√	√	√	√
SmartThin	√	√	√	√	√	√
SmartThick	√	√	√	√	√	√
SmartPartition	√	√	√	√	√	√
HyperMetro	√	√	√	√	√	√
Retype	√	√	√	√	√	√
Manage/Unmanage Volume	√	√	√	√	√	√
Manage/Unmanage Snapshot	x	√	√	√	√	√
Replication V2.1	x	√	√	√	√	√
HyperMetro Consistency Group	√	√	√	√	√	√
Create a volume on a certain storage pool	√	√	√	√	√	√
Create a volume with certain disk type	√	√	√	√	√	√
Backup Snapshot	x	√	√	√	√	√
Snapshot Consistency Group	х	√	√	√	√	√
Multipath	√	√	√	√	√	√
Consistency Group	√	√	√	√	√	√



注意

- OceanStor Dorado不支持Thick卷、SmartTier、SmartCache和SmartPartition特性。
- 低端存储不支持某些增值特性,请参考具体存储型号的支持特性规格。

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,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。

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 Drvier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。

----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 Drvier安装目录:

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

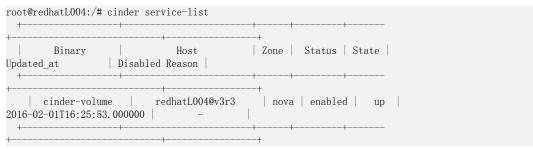
步骤2 将获取到的OpenStack Cinder Driver拷贝到Cinder节点驱动安装目录,默认路径参考步骤1。

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。



----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。



注意

Driver配置使用的存储池,需要事先保证在华为存储上存在,否则请手动创建,并且存储池类型必须是"块存储服务"。

- 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
- 4.4 配置租户用户

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>
      <ControllerIPO>x. x. x. x/ControllerIPO>
      <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" />
```

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):

□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 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:

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):

Mixe

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 重新启动Cinder服务。

----结束

4.1.3 配置 V3/V5 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:

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):

◯◯₩睭

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 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系列存储系统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:

□ 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18000_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:

```
[18000_iSCSI]

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

cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml

volume backend name = 18000 iSCSI
```

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

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

步骤4 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>
     \langle / Storage \rangle
     <LUN>
        <LUNType>Thin</LUNType>
        <StoragePool>xxx</StoragePool>
     </LUN>
     <iSCSI>
        <DefaultTargetIP>x.x.x.x/DefaultTargetIP>
     </iscsi>
```

```
<LUN>
<WriteType>xxx</WriteType>
```

□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 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	说明	适用产品
Product	存储产品类型,可选值为T、TV2、18000、V3、V5和 Dorado。	全部
Protocol	连接协议类型,选值为"iSCSI"。	全部
ControllerIP 0	T系列V100R005存储设备主控制器IP地址。	TV1
ControllerIP	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: https://x.x.x.x:8088/deviceManager/rest/ x.x.x.x代表管理IP地址。如果需要配置多个RestURL,以分号";"进行分割。	TV2 V3 V5 18000 Dorado
UserName	存储管理员用户名。	全部

属性名称	说明	适用产品
UserPasswor d	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
DefaultTarge tIP	存储系统提供给计算节点的默认iSCSI目标器端口IP。	全部

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
LUNcopyWaitInter val	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 V5 18000 Dorado
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 V5 18000 Dorado
Initiator Name	-	计算节点启动器iqn。	全部

属性名称	默认值	说明	适用产品
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	-	存储系统提供给计算节点的iSCSI目	TV2
TargetPortGroup		标器端口组名称。	V3
			V5
			18000
			Dorado
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



داندار

- StoragePool不允许配置为中文
- 所有参数值中不能含有XML特殊字符<>&'"

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。

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 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:

表 4-3 参数说明

参数	默认值	说明
Name	-	ISCSI启动器名称
ALUA	0	0: 默认(华为自研多路径) 1: 第三方多路径
FAILOVERMODE	-	启动器的切换模式,当开启第三方多路径时该参 数才需要下发。
SPECIALMODETYPE	-	启动器的特殊模式类型, 当启动器的切换模式为 特殊模式时该参数才需要下发。
РАТНТҮРЕ	-	启动器的路径类型。
CHAPinfo	-	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;
		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。

∭说明

由于FAILOVERMODE、SPECIALMODETYPE和PATHTYPE参数的取值范围在不同存储型号上存在差异,请参考相应存储型号的API文档进行配置。

----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:

□ 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 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:

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

□ 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 重新启动Cinder服务。

----结束

4.2.3 配置 V3/V5 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:

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

```
<WriteType>xxx</WriteType>
  <Prefetch Type="xxxx" Value="xxxx" />
    <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 重新启动Cinder服务。

----结束

4.2.4 配置 18000 系列 FC 协议 Cinder Driver

"cinder huawei conf.xml" .

介绍在FC协议下配置OceanStor 18000系列存储系统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

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:

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>
```

□□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18000_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:

```
[18000_FC]
volume_driver = cinder.volume.drivers.huawei_huawei_driver.HuaweiFCDriver
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml
volume_backend_name = 18000_FC
```

在[DEFAULT] 区块中修改以下内容,配置18000V3_FC后端:

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

步骤4 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:

```
</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):

□ 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 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 重新启动Cinder服务。

----结束



注意

● 使用租户用户不支持自动划域特性。

4.2.7 配置文件参数属性说明

表 4-4 必填参数说明

属性名称	说明	适用产品
Product	存储产品类型,可选值为T、TV2、18000、V3、V5和Dorado。	全部
Protocol	连接协议类型,选值为"FC"。	全部
ControllerIP 0	T系列V100R005存储设备主控制器IP地址。	TV1
ControllerIP	T系列V100R005存储设备从控制器IP地址。	TV1
RestURL		
	deviceManager/rest/ x.x.x.x代表管理IP地址。如果需要 配置多个RestURL,以分号";"进行分割。	V3
	ELD RESIDEL, SA J , MITAIN	V5
		18000
		Dorado
UserName	管理员用户名。	全部
UserPasswo rd	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 4-5 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
LUNcopyWait Interval	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 V5 18000 Dorado
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 V5 18000 Dorado
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



小心

- StoragePool不允许配置为中文
- 所有参数值中不能含有XML特殊字符<>&'"

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 重启nova-compute服务。

----结束

4.2.9 配置 ALUA 特性

如果需要为FC配置ALUA特性,可参考以下配置进行。

操作步骤

步骤1 配置ALUA。

在Cinder Driver配置文件中进行如下配置:

表 4-6 参数说明

参数	默认值	说明
Name	-	FC启动器WWN
ALUA	0	0: 默认(华为自研多路径) 1: 第三方多路径
FAILOVERMODE	-	启动器的切换模式,当开启第三方多路径时该参 数才需要下发。
SPECIALMODETYPE	-	启动器的特殊模式类型,当启动器的切换模式为 特殊模式时该参数才需要下发。
РАТНТҮРЕ	-	启动器的路径类型。

□说明

由于FAILOVERMODE、SPECIALMODETYPE和PATHTYPE参数的取值范围在不同存储型号上存在差异,请参考相应存储型号的API文档进行配置。

----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
```

4.4 配置租户用户

在华为自定义Driver配置文件中配置租户用户对接后端存储,配置方式如下:

表 4-7 租户参数说明

参数名称	说明
UserName	租户用户名
UserPassword	租户用户密码
vStoreName	vstore名



注意

- 其他相关参数请参考前面章节。
- 仅对接具有多租户功能的存储版本时支持配置租户用户。

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
- 5.15 配置LUN归属控制器
- 5.16 配置SmartDedupe属性
- 5.17 配置SmartCompression属性

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



/////

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。

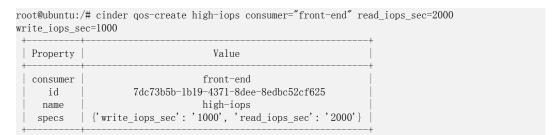
操作步骤

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-ic	ops	1	ı
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。



步骤3 建立卷类型和QoS控制属性关联。

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>。

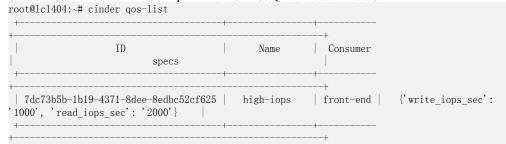
root@ubuntu:/# cinder qos-associate
usage: cinder qos-associate <qos_specs> <volume_type_id>
try 'cinder help qos-associate' for more information.

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

----结束

操作结果

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



∭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 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。

r	oot@ubuntu:/# cinder type-create maxIOPS	S		
	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。



注意

Dorado阵列IOType仅支持"2"。

以 "maxIOPS=100 IOType=2" 为例进行说明。

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 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。

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。



注意

Dorado阵列IOType仅支持"2"。

以 "minIOPS=50 IOType=2" 为例进行说明。

root@ubuntu:/# cinder qos-create miniops minIOPS=50 IOType=2

+	·
property	value
consumer id name specs	back-end 8deaf13c-bf1c-4a71-94d0-1149805693ce miniops {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。

步骤2 执行以下命令,配置SmartOos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。



注意

Dorado阵列IOType仅支持"2"。

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 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

5.1.5 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。

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。



注意

Dorado阵列IOType仅支持"2"。

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OoS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

5.1.6 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。

oot@ubuntu:/# cinder type-create latency	7		
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。



注意

Dorado阵列IOType仅支持"2"。

以 "latency=100 IOType=2" 为例进行说明。

root@ubuntu:/# ci	nder qos-create latency la	atency=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

5.1.7 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。

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。



注意

Dorado阵列IOType仅支持"2"。

以 "latency=100 minBandWidth=100 IOType=2" 为例进行说明。

rc	oot@ubuntu:/#	${\tt cinder \ qos-create}$	$\verb multi-strategy $	latency=100	${\tt minBandWidth=100}$	IOType=2
+	+				+	
	Property		Value			
4						
ĺ	consumer		back-end		İ	

```
| id | 36f0fc4a-d5f5-4862-ab1f-e7fedeeddc41
| name | multi-strategy
| specs | {'IOType': '2', 'minBandWidth': '100', 'latenct': '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 gos-list命令查看QoS策略配置详情。

```
root@ubuntu:~# cinder qos-list
                                                       Consumer
                                             Name
                          specs
  36f0fc4a-d5f5-4862-ab1f-e7fedeeddc41 | multi-strategy | back-end | {'IOType': '2',
'minBandWidth': '100', 'latenct': '100'}
  3e2b2fa5-aebb-4ea0-8dd2-52a577015769
                                           miniops
                                                       back-end
                                                                               {'IOType':
'2', 'minIOPS': '50'}
  3fd0971e-f5fd-48d7-89d9-d2efb6fc44cf
                                         maxbandwidth | back-end
{'maxBandWidth': '100', 'IOType': '2'}
 6a4e24af-0e8c-4a64-91ee-859434414400
                                                       back-end
                                                                              {'latency':
                                           latency
100', 'IOType': '2'}
 9b34fbdf-91b5-473d-a5ff-0b4ea2ed9fd0
                                                       back-end
                                                                              {'IOType':
                                           maxions
'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-d532b8ddbe52maxBandWidth True 92 dbab58 - 07 d8 - 46 e6 - 9327 - 532 a9 a3 b30 b6maxIOPSTrue 98d5ad9c-4591-4d18-ba09-199f6dba95a9minBandWidth True df53d9d7-b1db-4e6b-847a-7a5150c39489multi-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

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-63fcd7db8f2
metadata	{}
multiattach	false
size	1
snapshot_id	none
source_volid	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。

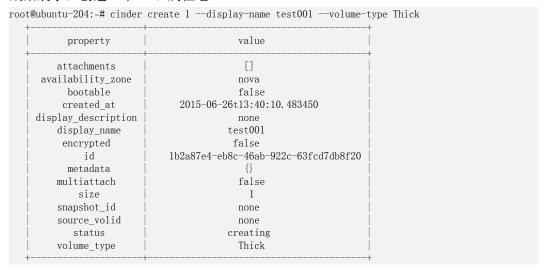
root@ubuntu-204:~# cinder type-create Thick

+	·	·	·
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。



----结束

5.4 配置 Thin 属性

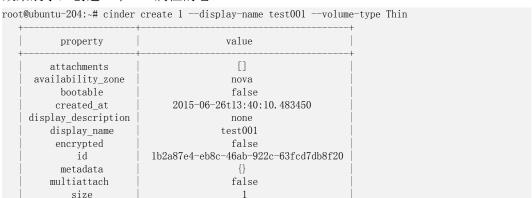
以举例的方式,介绍配置Thin属性的操作步骤。

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。

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。



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 1b2a87e4-eb8c-46ab-922c-63fcd7db8f20id 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-accc-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。

 ${\tt root@ubuntu-204:} {\tt ~\#~cinder~create~1~--display-name~test001~--volume-type~Smarttier}$

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_volid	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:xxx,
san_password:xxx,
vstore_name:xxx,
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:



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 5-1 参数含义参考下表属性说明:

属性名称	说明
san_address	Rest接口访问地址。参考配置: https://x.x.x.x:8088/deviceManager/rest/, x.x.x.x代表管理IP地址
san_user	双活远端存储管理员用户名或租户用户名
san_password	双活远端存储管理员密码或租户用户密码
vstore_name	双活远端存储vstore名(仅在使用租户用户场景下需要配置)
metro_domain	双活域名称
storage_pool	双活远端资源池名称
iscsi_default_target_ip	双活远端业务端口IP
Initiator Name	计算节点启动器iqn。
Initiator Target IP	存储系统提供给计算节点的iSCSI目标器端口IP。
Initiator TargetPortGroup	存储系统提供给计算节点的iSCSI目标器端口组名 称。

步骤2 重启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 1display-name test001volume-	type Hypermetro
	property	value	+
=	attachments availability_zone bootable	[] nova false	*

created at	2015-06-26±13:40:10, 483450
display description	none
	test001
display_name	
encrypted	false
id	1b2a87e4-eb8c-46ab-922c-63fcd7db8f20
metadata	{}
multiattach	false
size	1
snapshot_id	none
source_volid	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:xxx,
   san_password:xxx,
   vstore_name:xxx,
   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:



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	说明
backend_id	目标设备id,failover时需要指定的远端设备
storage_pool	复制failover目标后端存储池名称。
san_address	Rest接口访问地址。参考配置: https://x.x.x.x8088/deviceManager/rest/ x.x.x.x代表管理IP地址
san_user	复制远端存储管理员用户名或租户用户名

属性名称	说明
san_password	复制远端存储管理员密码或租户用户密码
vstore_name	复制远端存储租户名(仅在使用租户用户场景下需要配置)
iscsi_default_target_ip	远端业务端口IP
Initiator Name	计算节点启动器iqn。
Initiator Target IP	存储系统提供给计算节点的iSCSI目标器端口IP。
Initiator TargetPortGroup	存储系统提供给计算节点的iSCSI目标器端口组名 称。

步骤2 重启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 Rep	olication_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:~# c	nder create 1display-name test001volume-type Replication_V2.1
property	value
attachments availability_ze bootable created_at display_descrip display_nam encrypted id	false 2015-06-26t13:42:10.483450 ion none
metadata multiattach size snapshot_id source_voli	{} false l none none creating



----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 ta	arget-pool	ı	ı
ID	Name	Description	Is_Public
2ecf5589-507e-4ee5-995f-4aa1d0aaff83	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 false bootable consistencygroup_id None 2016-08-12T08:14:21.000000 created at description None encrypted 3023 e 949 - e c 25 - 4005 - 8 b b c - 6 a a 2 c 6860235id 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 53c46df66a084916876a08d7bcc31d87 os-vol-tenant-attr:tenant_id replication_status disabled size 1 snapshot_id None source_volid None status creating updated at None ec5b1bd5fa8646109381755037aacc82 user_id 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 id eca 3a 81b - 0ef 0 - 49b 6 - 920 2 - d1b f1c 16a 96fmetadata {} None migration_status False multiattach 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 53c46df66a084916876a08d7bcc31d87 os-vol-tenant-attr:tenant_id disabled replication_status size 1 $snapshot_id$ None source volid None status creating 2016-08-12T08:24:11.000000 updated at user_id ec5b1bd5fa8646109381755037aacc82 disk-type volume_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。

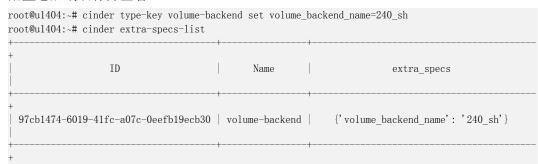
操作步骤

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@u1404	:~# cinder type-create	volume-back	kend	
+		+		++
	ID		Name	Description Is_Public

		+	++	
-41fc-a07c-0eefb19ecb30 volume-backend - True	cb1474-6019-41fc-a07c-0eefb19ecb30	volume-backend	-	True

步骤2 配置卷后端名称并查看。



步骤3 效果展示,指定后端创建卷。

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	u1404@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:

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

5.15 配置 LUN 归属控制器

以举例的方式,介绍配置LUN 归属控制器操作步骤。

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-p-200:~# cinder type-create controller_type_C
+-----+
```

ID	Name	Description	Is_Public
a6f65c73-ff74-4809-9e63-086052169e42	controller_type_C	-	True

步骤2 执行以下命令,配置huawei controller属性为"true"的键值对。

 ${\tt root@ubuntu-p-200:~\#~cinder~type-key~controller_type_C~set~capabilities:huawei_controller='`<is>true'}$

步骤3 配置存储设备上已经存在的huawei_controller的名称,将controllername关联到卷类型中

root@ubuntu-p-200:~# cinder type-key controller_type_C set huawei controller:controllername='CTEO.C'

步骤4 基于该volume type创建一个指定归属控制器属性的卷。

----结束

5.16 配置 SmartDedupe 属性

介绍配置SmartDedupe属性操作步骤。仅Thin卷支持。

开启 SmartDedupe 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。

ro	ot@ubuntu-p-200:~# cinder type-create	Dedup 		L
į:	ID	Name	Description	Is_Public
;	a6f65c73-ff74-4809-9e63-086052169e42	Dedup	_	True

步骤2 对于Newton及之后版本,执行cinder type-key Dedup set capabilities:thin_provisioning_support='<is> true' capabilities:dedup='<is> true'命令设置卷类型参数。

对于Liberty和Mitaka版本,执行**cinder type-key Dedup set capabilities:thin_provisioning_support='<is> true' capabilities:dedup='<in> true'命令设置卷类型参数。**

步骤3 使用Dedup类型创建一个开启SmartDedupe的卷。

----结束

关闭 SmartDedupe 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。

ID	root@ubuntu-p-200:~# cinder type-create	NoDedup		
a6f65c73-ff74-4809-9e63-086052169e42 NoDedup	ID	Name	Description	Is_Public
·	a6f65c73-ff74-4809-9e63-086052169e42	NoDedup	-	True

步骤2 对于Newton及之后版本,执行**cinder type-key NoDedup set capabilities:thin_provisioning_support='<is> true' capabilities:dedup='<is> false'命令设置卷类型参数。**

对于Liberty和Mitaka版本,执行**cinder type-key NoDedup set** capabilities:thin_provisioning_support='<is> true' capabilities:dedup='<in> false'命令 设置卷类型参数。

步骤3 使用NoDedup类型创建一个关闭SmartDedupe的卷。

----结束

□□说明

Dorado系列存储仅支持Thin卷,步骤2中capabilities:thin_provisioning_support配置可选。

5.17 配置 SmartCompression 属性

介绍配置SmartCompression属性操作步骤。仅Thin卷支持。

开启 SmartCompression 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-p-200:~# cinder type-create	Compression		
ID	Name	Description	Is_Public
a6f65c73-ff74-4809-9e63-086052169e42	Compression	-	True

步骤2 对于Newton及之后版本,执行cinder type-key Compression set capabilities:thin_provisioning_support='<is> true' capabilities:compression='<is> true'命令设置卷类型参数。

对于Liberty和Mitaka版本,执行**cinder type-key Compression set capabilities:thin_provisioning_support='<is> true' capabilities:compression='<in> true'命令设置卷类型参数。**

步骤3 使用Compression类型创建一个开启SmartCompression的卷。

----结束

关闭 SmartCompression 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-p-200:~# cinder type-create	NoCompression		L
ID	Name	Description	Is_Public
a6f65c73-ff74-4809-9e63-086052169e42	NoCompression	-	True

步骤2 对于Newton及之后版本,执行cinder type-key NoCompression set capabilities:thin_provisioning_support='<is> true' capabilities:compression='<is> false'命令设置卷类型参数。

对于Liberty和Mitaka版本,执行**cinder type-key NoCompression set capabilities:thin_provisioning_support='<is> true' capabilities:compression='<in>false'命令设置卷类型参数。**

步骤3 使用NoCompression类型创建一个关闭SmartCompression的卷。

----结束

∭说明

Dorado系列存储仅支持Thin卷,步骤2中capabilities:thin_provisioning_support配置可选。

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 在华为存储上创建或者查看待使用的块存储池。

步骤3 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。

● 检查XML文件拥有者及用户组。

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
```

● 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4 重新启动Cinder服务。

步骤5 检查服务状态。



其中u1404@V3_iSCSI的服务状态为up,代表服务已正常启动。

----结束

 7_{FAQ}

7.1 通过Cinder Driver映射卷是否需要事先在华为存储上创建主机、主机组、LUN组等映射需要的关联对象?

7.2 映射卷时Nova日志中打印lsblk命令执行失败导致映射失败。

7.1 通过 Cinder Driver 映射卷是否需要事先在华为存储上创建主机、主机组、LUN 组等映射需要的关联对象?

不需要,因为Driver会自动创建这些卷映射需要的关联对象,以及映射视图。相反,如 果手动在存储上创建了这些对象,反而会影响Driver的映射逻辑,请确保在使用Cinder 映射前删除这些对象。

7.2 映射卷时 Nova 日志中打印 lsblk 命令执行失败导致映射失败。

Nova在卷映射流程中会调用lsblk命令查询映射卷的信息,出现此错误一般是未安装 lsblk工具,请确保Nova系统中安装了此工具并且有相应的执行权限。