

**eSDK Enterprise Storage Plugins
2.5.3**

OpenStack Manila Driver 配置指南

文档版本	01
发布日期	2022-12-30



版权所有 © 华为技术有限公司 2022。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暗示的声明或保证。

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： <https://e.huawei.com>

前言






读者对象

本文档主要适用于以下读者对象：

- 技术支持工程师
- 运维工程师
- 具备存储和OpenStack基础知识的工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
 警告	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
 注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。
 须知	用于传递设备或环境安全警示信息。如不避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
 说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修改记录

文档版本	发布日期	修改说明
01	2022-12-30	第一次正式发布。

目录

前言.....	ii
1 简介.....	1
2 版本配套关系.....	2
3 规格与限制.....	5
4 Manila Driver 安装与部署.....	6
4.1 Manila Driver 获取.....	6
4.2 Manila Share 服务华为 Driver 部署.....	6
5 Manila Driver 基本属性配置.....	8
5.1 配置 manila driver.....	8
5.2 验证 manila share 服务.....	11
6 Manila Driver 高级属性配置（可选）.....	13
6.1 配置 Thin/Thick 属性.....	13
6.2 配置 SmartDedupe 属性.....	14
6.3 配置 SmartCompression 属性.....	14
6.4 配置 SmartCache 属性.....	15
6.5 配置 SmartPartition 属性.....	15
6.6 配置 SmartQoS 属性.....	16
6.6.1 配置控制最大 IOPS.....	17
6.6.2 配置控制最小 IOPS.....	17
6.6.3 配置控制最大带宽.....	18
6.6.4 配置控制最小带宽.....	18
6.6.5 配置控制时延.....	19
6.6.6 配置控制多策略.....	19
6.7 指定 Storage Pool 创建共享点.....	20
6.8 指定 Disk Type 创建共享点.....	20
6.9 配置 SectorSize.....	20
6.10 配置 Replication.....	21
6.11 配置 FS 归属控制器.....	22
6.12 配置 NFS 客户端权限.....	23
6.13 配置 HyperMetro.....	23
6.14 配置 filesystem:mode.....	25

7 最佳实践	26
7.1 快速对接华为存储	26

1 简介

华为Manila Driver是部署在OpenStack Manila模块上的一个插件，用于对接华为存储向OpenStack平台提供共享存储能力。

2 版本配套关系

介绍华为Manila Driver版本与华为存储、OpenStack的版本配套关系。

表 2-1 Manila Driver 存储产品版本支持说明

OpenStack版本	存储产品版本
Rocky/Train/Ussuri/ Victoria/Wallaby/Xena/ Yoga	OceanStor V3/F V3 V300R006C60/V300R006C61 OceanStor V5/F V5 V500R007C61/V500R007C70 OceanStor V6 6.1.3/6.1.5 OceanStor Dorado V3 V300R002C10/V300R002C20 OceanStor Dorado V6 6.1.0/6.1.2/6.1.3/6.1.5

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

特性	Normal Share	Hypermetro Share	备注
OpenStack	Rocky/Train/ Ussuri/ Victoria/ Wallaby/ Xena/Yoga	Rocky/Train/ Ussuri/ Victoria/ Wallaby/ Xena/Yoga	NA
Create Share	√	√	NA
Delete Share	√	√	NA
Allow access	√	√	NA
Deny access	√	√	NA
Create Snapshot	√	√	Hypermetro Share只支持在双活激活端创建快照
Delete Snapshot	√	√	Hypermetro Share只支持在双活激活端删除快照

特性	Normal Share	Hypermetro Share	备注
Revert Snapshot	✓	✓	Hypermetro Share只支持在双活激活端回滚快照
Extend Share	✓	✓	NA
Shrink Share	✓	✓	NA
Create Share from Snapshot	✓	✓	Hypermetro Share只支持在双活激活端从快照创建
Manage/Unmanage Share	✓	x	Manila管理的文件系统和共享关系必须是一对一 只支持管理非多租户的文件系统和共享
Manage/Unmanage Snapshot	✓	x	NA
Replication	✓	x	NA
QoS	✓	✓	Hypermetro Share只支持在双活激活端配置SmartQoS
Multi-tenancy	✓	x	NA
SmartThin/SmartThick	✓	✓	NA
SmartCompression	✓	✓	NA
SmartDedupe	✓	✓	NA
Ensure Share	✓	✓	NA
Create a share on a certain storage pool	✓	✓	NA
Create a share with a certain disk type	✓	✓	NA
SectorSize	✓	✓	NA
SmartPartition	✓	✓	Hypermetro Share只支持在双活激活端配置SmartPartition
SmartCache	✓	✓	Hypermetro Share只支持在双活激活端配置SmartCache

说明

OceanStor Dorado NAS架构发生变化，以上涉及双活激活端的特性以实际存储为准。

对dorado V6 6.1.2及以后的版本，Hypermetro Share支持在存储主从切换后创建和删除share，创建删除回滚快照，扩容缩容share。

3 规格与限制

主特性	子特性	描述	备注
平台配套	原生OpenStack平台	OpenStack版本 Rocky/Train/Ussuri/ Victoria/Wallaby/ Xena/Yoga	-
配置	xml配置文件	Driver配置使用的存储池，需要事先保证在华为存储上存在，否则需要手动创建，存储池类型必须是“文件存储服务”，且存储池名不能包含中文	-
		华为Driver配置文件所有参数值中不能含有XML特殊字符< > & ' "	-
		Product配置项务必根据实际对接的后端存储型号填写	-
策略	SmartQoS	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share失败	限制策略：maxIOPS，maxBandWidth 保护策略：minIOPS，minBandWidth，latency
		QoS属性中，IOType为必选项，如果QoS策略中未配置IOType会导致创建share失败	
		OceanStor Dorado 6.1.0暂只支持限制策略，不支持保护策略	

4 Manila Driver 安装与部署

4.1 Manila Driver获取

4.2 Manila Share服务华为Driver部署

4.1 Manila Driver 获取

步骤1 打开浏览器，访问仓库地址：https://github.com/Huawei/OpenStack_Driver。

步骤2 单击“Download ZIP”按钮，华为Driver将以压缩包的形式下载到本地。

步骤3 解压该压缩包。

步骤4 在解压出的目录下找到Manila目录，其下即包含多个OpenStack版本的华为Driver。

----结束

说明

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。

4.2 Manila Share 服务华为 Driver 部署

OpenStack社区标准部署步骤如下：

步骤1 查找系统中原有的华为Driver代码，相对路径为.../manila/share/drivers/huawei。

说明

完整路径在不同的系统下可能有所不同，可通过以下方法搜索华为Driver代码目录：

- 方式1：

执行以下命令，回显中的“/usr/lib/python2.7/site-packages/manila/share/drivers/huawei”即为代码目录。

```
# python -c "from manila.share.drivers import huawei; print (huawei.__path__)"  
['/usr/lib/python2.7/site-packages/manila/share/drivers/huawei']
```

- 方式2：

执行以下命令，回显中的“/usr/lib/python3.7/site-packages/manila/share/drivers/huawei”即为代码目录。

```
# python3 -c "from manila.share.drivers import huawei; print (huawei.__path__)"  
['/usr/lib/python3.7/site-packages/manila/share/drivers/huawei']
```

步骤2 将获取到的OpenStack Manila Driver拷贝到Manila节点驱动安装目录，默认路径参考步骤1，以X版本为例，复制后的结果如下。（只拷贝覆盖，切勿删除）

```
# ls -l  
-rw-r--r-- 1 root root 4453 Mar 30 10:51 constants.py  
-rw-r--r-- 1 root root 41592 Mar 30 10:51 helper.py  
-rw-r--r-- 1 root root 10467 Mar 30 10:51 huawei_config.py  
-rw-r--r-- 1 root root 92785 Mar 30 10:51 huawei_nas.py  
-rw-r--r-- 1 root root 9791 Mar 30 10:51 huawei_utils.py  
-rw-r--r-- 1 root root 7875 Mar 30 10:51 hypermetro.py  
-rw-r--r-- 1 root root 23 Mar 30 10:51 __init__.py  
-rw-r--r-- 1 root root 3380 Mar 30 10:51 manager.py  
-rw-r--r-- 1 root root 13173 Mar 30 10:51 replication.py  
-rw-r--r-- 1 root root 8171 Mar 30 10:51 rpcapi.py  
-rw-r--r-- 1 root root 4491 Mar 30 10:51 smartx.py
```

----结束

5 Manila Driver 基本属性配置

介绍如何配置华为Manila Driver。

须知

- OpenStack Ocata Manila创建共享点时，如未指定share type，会使用"/etc/manila/manila.conf"文件中默认share type。
- 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支持“文件存储服务”。
- 华为Driver配置文件所有参数值中不能含有XML特殊字符< > & ' "。
- 快照功能和复制功能不能同时开启

5.1 配置manila driver

5.2 验证manila share服务

5.1 配置 manila driver

操作步骤

步骤1 在“/etc/manila”路径下创建华为Driver配置文件，文件格式为XML。配置文件名称可自定义，例如manila_huawei_conf.xml。

步骤2 编辑manila_huawei_conf.xml，配置华为Driver配置文件必填参数

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
  <Storage>
    <Product>V3</Product>
    <LogicalPortIP>x.x.x.x</LogicalPortIP>
    <DNS>x.x.x.x</DNS>
    <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
    <UserName>xxx</UserName>
    <UserPassword>xxx</UserPassword>
  </Storage>
  <Filesystem>
    <StoragePool>xxx</StoragePool>
  </Filesystem>
</config>
```

表 5-1 必填参数属性说明

属性名称	默认值	说明	备注
Product	V3	存储产品类型, 可选值为V3、V5、V6或Dorado。	-
LogicalPortIP	-	逻辑端口IP。支持配置多个IP, 以分号;分隔。	DNS不配置时必须选。同时配置时优先级低于DNS
DNS	-	DNS域名。支持配置多个DNS, 以分号;分隔。	LogicalPortIP不配置时必须选。同时配置时优先级高于LogicalPortIP
RestURL	-	Rest接口访问地址。	-
UserName	-	阵列管理员用户名。	-
UserPassword	-	阵列管理员密码。	-
StoragePool	-	需要使用的存储池名称。	-

步骤3 编辑manila_huawei_conf.xml, 配置华为Driver配置文件选填参数。

```
<?xml version='1.0' encoding='UTF-8'?>
<Config>
  <Storage>
    <Port>abc;CTE0.A.H1</Port>
    <SnapshotSupport>True</SnapshotSupport>
    <ReplicationSupport>False</ReplicationSupport>
  </Storage>
  <Filesystem>
    <SectorSize>64</SectorSize>
    <WaitInterval>3</WaitInterval>
    <Timeout>60</Timeout>
    <SnapshotReserve>20</SnapshotReserve>
    <NFSCClient>
      <IP>x.x.x.x</IP>
    </NFSCClient>
    <CIFSCClient>
      <UserName>xxx</UserName>
      <UserPassword>xxx</UserPassword>
    </CIFSCClient>
  </Filesystem>
</Config>
```

表 5-2 选填参数属性说明

属性名称	默认值	说明	类型
Port	-	可用的业务端口（绑定端口或者物理端口）。	多租户模式下 (driver_handles_share_servers为True)可选，如不配置，则使用所有online的端口。
SectorSize	64	文件系统磁盘块大小，可选值“4”、“8”、“16”、“32”、“64”，单位KB。	可选。
SnapshotSupport	True	是否提供快照功能。	除OceanStor Dorado外，其余存储不可与ReplicationSupport同时开启
ReplicationSupport	False	是否提供远程复制功能。	除OceanStor Dorado外，其余存储不可与SnapshotSupport同时开启
WaitInterval	3	查询文件系统状态间隔时间，单位为秒（s）。	可选。
Timeout	60	等待存储设备执行命令的超时时间，单位为秒（s）。	可选。
SnapshotReserve	20	快照空间比例（%）。（范围为0~50）	可选
NFSClient\IP	-	从快照创建共享时，管理节点上用于挂载NFS共享点所用IP。	从快照创建共享点必选。
CIFSClient\UserName	-	从快照创建共享时，管理节点上用于挂载CIFS共享点所用用户名。	从快照创建共享点必选。使用OceanStor Dorado存储，用户名需要加入具有管理员权限的本地认证用户组
CIFSClient\UserPassword	-	从快照创建共享时，管理节点上用于挂载CIFS共享点所用用户密码。	从快照创建共享点必选。

步骤4 配置“/etc/manila/manila.conf”文件，在文件最后添加如下后端配置

```
[huawei_backend]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf.xml
```

```
share_backend_name = huawei_backend
driver_handles_share_servers = False
```

说明

- 后端名huawei_backend可自定义。
- share_driver表示待使用的Driver类型，此处填写manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver。
- manila_huawei_conf_file表示配置的华为Driver配置文件路径。
- driver_handles_share_servers表示是否启用多租户。

在[DEFAULT]区块中修改enabled_share_backends配置项，新增huawei_backend后端。

```
[DEFAULT]
...
enabled_share_backends=xxx,huawei_backend
```

步骤5 修改/etc/manila/manila_huawei_conf.xml文件权限，使其与/etc/manila/manila.conf保持一致。

```
# ls /etc/manila/
...
-rw-r----- 1 manila manila 82179 Jan 20 14:34 manila.conf
-rw-r----- 1 manila manila 1005 Jan 20 10:55 manila_huawei_conf.xml
...
```

步骤6 重新启动Manila服务。

```
systemctl restart openstack-manila-share.service
```

----结束

5.2 验证 manila share 服务

操作步骤

步骤1 创建通用share类型。“general”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

```
# manila type-create general False
```

步骤2 设置share类型后端。“huawei_backend”代表share的后端名称，为[5.1 配置manila driver](#)中步骤4的share_backend_name字段。

```
# manila type-key general set share_backend_name=huawei_backend
```

步骤3 设置快照属性。

```
# manila type-key general set snapshot_support=True
```

步骤4 设置快照回滚属性。

```
# manila type-key general set revert_to_snapshot_support=True
```

步骤5 设置从快照创建共享属性。

```
# manila type-key general set create_share_from_snapshot_support=True
```

步骤6 执行manila create --name test001 NFS 2 --share-type general，创建支持以上属性的share。

----结束

说明

- 快照属性：该特性与远程复制互斥，两者不能同时配置。（Dorado V6 NAS除外）

6 Manila Driver 高级属性配置（可选）

介绍配置存储高级属性的操作步骤。

华为存储支持一系列的smartx高级特性，这些高级特性能通过与特定的share类型关联的方式，在OpenStack环境中使用。

[6.1 配置Thin/Thick属性](#)

[6.2 配置SmartDedupe属性](#)

[6.3 配置SmartCompression属性](#)

[6.4 配置SmartCache属性](#)

[6.5 配置SmartPartition属性](#)

[6.6 配置SmartQoS属性](#)

[6.7 指定Storage Pool创建共享点](#)

[6.8 指定Disk Type创建共享点](#)

[6.9 配置SectorSize](#)

[6.10 配置Replication](#)

[6.11 配置FS归属控制器](#)

[6.12 配置NFS 客户端权限](#)

[6.13 配置HyperMetro](#)

[6.14 配置filesystem:mode](#)

6.1 配置 Thin/Thick 属性

介绍配置Thin/Thick属性操作步骤。

操作步骤

步骤1 执行**manila type-create thin_type False**命令，创建share类型。“thin_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

步骤2 配置Thin属性：执行以下命令，配置thin_provisioning属性为“true”的键值对。

```
root@ubuntu:~# manila type-key thin_type set capabilities:thin_provisioning='<is> true'
```

配置Thick属性：执行以下命令，配置thin_provisioning属性为“false”的键值对。

```
root@ubuntu:~# manila type-key thin_type set capabilities:thin_provisioning='<is> false'
```

步骤3 执行**manila create --name test001 NFS 2 --share-type thin_type**，创建支持以上属性的share。

----结束

说明

- share-type中配置“thin_provisioning”属性为“true”，共享点类型为“Thin”；“thin_provisioning”属性为“false”时，共享点类型为“Thick”。
- share-type中未配置“thin_provisioning”，共享点默认类型为“Thin”。
- Dorado V6.x 和 融合V6.x不支持配置类型为“Thick”的共享点

6.2 配置 SmartDedupe 属性

介绍配置SmartDedupe属性操作步骤。

操作步骤

步骤1 执行**manila type-create dedupe_type False**命令，创建share类型。

“dedupe_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

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

```
root@ubuntu:~# manila type-key dedupe_type set capabilities:thin_provisioning='<is> true'
```

步骤3 执行以下命令，配置SmartDedupe属性为“true”的键值对。

```
root@ubuntu:~# manila type-key dedupe_type set capabilities:dedupe='<is> true'
```

步骤4 执行**manila create --name test001 NFS 2 --share-type dedupe_type**，创建支持以上属性的share。

----结束

说明

仅“Thin”共享点支持配置SmartDedupe。

6.3 配置 SmartCompression 属性

介绍配置SmartCompression属性操作步骤。

操作步骤

步骤1 执行**manila type-create compression_type False**命令，创建share类型。

“compression_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

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

```
root@ubuntu:~# manila type-key compression_type set capabilities:thin_provisioning='<is> true'
```

步骤3 执行以下命令，配置SmartCompression属性为“true”的键值对。

```
root@ubuntu:~# manila type-key compression_type set capabilities:compression='<is> true'
```

步骤4 执行**manila create --name test001 NFS 2 --share-type compression_type**，创建支持以上属性的share。

----结束

说明

仅“Thin”共享点支持配置SmartCompression。

6.4 配置 SmartCache 属性

介绍配置SmartCache属性操作步骤。

操作步骤

步骤1 执行**manila type-create cache_type False**命令，创建share类型。“cache_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

步骤2 执行以下命令，配置SmartCache属性为“true”的键值对。

```
root@ubuntu:~# manila type-key cache_type set capabilities:huawei_smartcache='<is> true'
```

配置存储设备上已经存在的SmartCache名称，将cache关联到share类型中。

```
root@ubuntu:~# manila type-key cache_type set huawei_smartcache:cache_name='test_name'
```

步骤3 执行**manila create --name test001 NFS 2 --share-type cache_type**，创建支持以上属性的share。

----结束

6.5 配置 SmartPartition 属性

介绍配置SmartPartition属性操作步骤。

操作步骤

步骤1 执行**manila type-create partition_type False**命令，创建share类型。“partition_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

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

```
root@ubuntu:~# manila type-key partition_type set capabilities:huawei_smartpartition='<is> true'
```

配置存储设备上已经存在的SmartPartition名称，将partition关联到share类型中。

```
root@ubuntu:~# manila type-key partition_type set huawei_smartpartition:partition_name='test_name'
```

步骤3 执行`manila create --name test001 NFS 2 --share-type partition_type`，创建支持以上属性的share。

----结束

6.6 配置 SmartQoS 属性

关于本章

华为支持如下QoS属性，同一类属性可单个或多个同时配置到一个QoS属性中。

保护策略：latency、minIOPS、minBandWidth

限制策略：maxIOPS、maxBandWidth

表 6-1 IOType 参数说明

参数名称	说明	备注
IOType	控制读写类型。	可选值如下： <ul style="list-style-type: none">• 0：读I/O• 1：写I/O• 2：读写I/O

表 6-2 QoS 属性说明

参数名称	说明	备注
maxIOPS	最大IOPS限制策略。	有效值为大于0的整数。对于OceanStor Dorado系列，有效值为大于等于100的整数。
minIOPS	最小IOPS保护策略。	有效值为大于0的整数。对于OceanStor Dorado系列，有效值为大于等于100的整数。
maxBandWidth	最大带宽限制策略。	单位为MB/s，有效值为大于0的整数。
minBandWidth	最小带宽保护策略。	单位为MB/s，有效值为大于0的整数。
latency	最大时延保护策略。	单位为ms，有效值为大于0的整数。对于OceanStor Dorado系列，有效值为500或者1500的整数。

说明

- 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share失败。
- OceanStor Dorado 6.1.0及以上版本暂只支持限制策略，不支持保护策略。
- QoS属性中，IOType为必选项，如果QoS策略中未配置IOType会导致创建share失败。对于OceanStor Dorado存储，仅支持IOType设置为“2”。

6.6.1 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**manila type-create maxiops_type False**命令，创建share类型。
“maxiops_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

步骤2 执行以下命令，配置SmartQoS属性为“true”的键值对。

```
root@ubuntu:~# manila type-key maxiops_type set capabilities:qos='<is> true'
```

执行以下命令，配置QoS控制属性参数。

```
root@ubuntu:~# manila type-key maxiops_type set qos:IOType=0 qos:maxIOPS=50
```

- maxIOPS：最大IOPS限制，数值为大于0的整数。对于OceanStor Dorado系列，有效值为大于等于100的整数。
- IOType（必选）：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。

步骤3 执行**manila create --name test001 NFS 2 --share-type maxiops_type**，创建支持以上属性的share。

----结束

6.6.2 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**manila type-create miniops_type False**命令，创建share类型。
“miniops_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

步骤2 执行以下命令，配置SmartQoS属性为“true”的键值对。

```
root@ubuntu:~# manila type-key miniops_type set capabilities:qos='<is> true'
```

执行以下命令，配置QoS控制属性参数。

```
root@ubuntu:~# manila type-key miniops_type set qos:IOType=0 qos:minIOPS=500
```

- minIOPS：最小IOPS限制，有效值为大于0的整数。对于OceanStor Dorado系列，有效值为大于等于100的整数。
- IOType（必选）：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。

步骤3 执行**manila create --name test001 NFS 2 --share-type minio_type**，创建支持以上属性的share。

----结束

6.6.3 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**manila type-create maxbandwidth_type False**命令，创建share类型。
“maxbandwidth_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

步骤2 执行以下命令，配置SmartQoS属性为“true”的键值对。

```
root@ubuntu:~# manila type-key maxbandwidth_type set capabilities:qos='<is> true'
```

执行以下命令，配置QoS控制属性参数。

```
root@ubuntu:~# manila type-key maxbandwidth_type set qos:IOType=0 qos:maxBandWidth=50
```

- maxBandWidth：最大带宽限制，数值为大于0的整数，单位为MB/s。
- IOType（必选）：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。

步骤3 执行**manila create --name test001 NFS 2 --share-type maxbandwidth_type**，创建支持以上属性的share。

----结束

6.6.4 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**manila type-create minbandwidth_type False**命令，创建share类型。
“minbandwidth_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

步骤2 执行以下命令，配置SmartQoS属性为“true”的键值对。

```
root@ubuntu:~# manila type-key minbandwidth_type set capabilities:qos='<is> true'
```

执行以下命令，配置QoS控制属性参数。

```
root@ubuntu:~# manila type-key minbandwidth_type set qos:IOType=0 qos:minBandWidth=50
```

- minBandWidth：最小带宽限制，数值为大于0的整数，单位为MB/s。
- IOType（必选）：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。

步骤3 执行**manila create --name test001 NFS 2 --share-type minbandwidth_type**，创建支持以上属性的share。

----结束

6.6.5 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**manila type-create latency_type False**命令，创建share类型。
“latency_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

步骤2 执行以下命令，配置SmartQoS属性为“true”的键值对。

```
root@ubuntu:~# manila type-key latency_type set capabilities:qos='<is> true'
```

执行以下命令，配置QoS控制属性参数。

```
root@ubuntu:~# manila type-key latency_type set qos:IOType=0 qos:latency=500
```

- latency：时延限制，数值为500或者1500的整数，单位为ms。
- IOType（必选）：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。

步骤3 执行**manila create --name test001 NFS 2 --share-type latency_type**，创建支持以上属性的share。

----结束

6.6.6 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**manila type-create multiple_strategy_type False**命令，创建share类型。
“multiple_strategy_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

步骤2 执行以下命令，配置SmartQoS属性为“true”的键值对。

```
root@ubuntu:~# manila type-key multiple_strategy_type set capabilities:qos='<is> true'
```

执行以下命令，配置QoS控制属性参数。

```
root@ubuntu:~# manila type-key multiple_strategy_type set qos:IOType=0 qos:latency=500  
qos:minIOPS=500 qos:minBandWidth=50
```

- latency：时延限制，数值为500或者1500的整数，单位为ms。
- minIOPS：最小IOPS限制，数值为大于99的整数。
- minBandWidth：最小带宽限制，数值为大于0的整数。
- IOType（必选）：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。

步骤3 执行**manila create --name test001 NFS 2 --share-type multiple_strategy_type**，创建支持以上属性的share。

----结束

6.7 指定 Storage Pool 创建共享点

介绍配置指定存储池创建共享点操作步骤。

操作步骤

- 步骤1** 执行**manila type-create target_pool_type False**命令，创建share类型。
“target_pool_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。
- 步骤2** 执行以下命令，配置目标存储池。
- 配置单个目标存储池

```
root@ubuntu:~# manila type-key target_pool_type set pool_name=StoragePool001
```
 - 配置多个目标存储池

```
root@ubuntu:~# manila type-key target_pool_type set pool_name="<or> StoragePool001 <or> StoragePool002"
```
- 步骤3** 执行**manila create --name test001 NFS 2 --share-type target_pool_type**，创建支持以上属性的share。
- 结束

6.8 指定 Disk Type 创建共享点

介绍配置指定存储池磁盘类型创建共享点操作步骤。

操作步骤

- 步骤1** 执行**manila type-create disk_type False**命令，创建share类型。“disk_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。
- 步骤2** 执行以下命令，配置存储池类型。
- 配置一种磁盘类型

```
root@ubuntu:~# manila type-key disk_type set huawei_disk_type=sas
```
 - 配置多种磁盘类型

```
root@ubuntu:~# manila type-key disk_type set huawei_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** 执行**manila create --name test001 NFS 2 --share-type disk_type**，创建支持以上属性的share。
- 结束

6.9 配置 SectorSize

介绍配置共享点源文件系统磁盘块大小操作步骤。

操作步骤

步骤1 执行**manila type-create sectorsize_type False**命令，创建share类型。
“sectorsize_type”代表share类型名称，由用户指定；“False”表示不支持多租户，
若需配置多租户，该参数配置为“True”。

步骤2 执行以下命令，配置SectorSize属性为“true”的键值对。

```
root@ubuntu:~# manila type-key sectorsize_type set capabilities:huawei_sectorsize='<is> true'
```

配置SectorSize取值。

```
root@ubuntu:~# manila type-key sectorsize_type set huawei_sectorsize:sectorsize=4
```

说明

针对不同的应用场景，华为存储系统提供了5个可选的SectorSize值（4/8/16/32/64），单位KB。

步骤3 执行**manila create --name test001 NFS 2 --share-type sectorsize_type**，创建支持以上属性的share。

----结束

说明

- share-type和xml文件中同时配置了SectorSize，优先使用share-type中的SectorSize值；
- share-type中无可取值且xml文件中同时配置了SectorSize，取xml文件中的SectorSize值。
- share-type和xml文件均未配置SectorSize，使用阵列提供的默认值64KB。
- Rocky版本及以后版本不支持SectorSize

6.10 配置 Replication

以举例的方式，介绍配置Replication的操作步骤。

前提条件

步骤1 在“/etc/manila/manila.conf”文件中配置有远程复制关系的两个后端。

```
[DEFAULT]
...
enabled_share_backends = huawei_manila_1, huawei_manila_2
...

[huawei_manila_1]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_1.xml
share_backend_name = huawei_manila_1
driver_handles_share_servers = False
replication_domain = huawei_domain
local_replication = False
replica_backend = host@huawei_manila_2

[huawei_manila_2]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_2.xml
share_backend_name = huawei_manila_2
driver_handles_share_servers = False
replication_domain = huawei_domain
local_replication = False
replica_backend = host@huawei_manila_1
```

说明

- “replication_domain” 在需要使用远程复制功能时设置，参数值为自定义字符串。有远程复制关系的后端之间需要设置相同的参数值；
- “replica_backend” 在需要使用远程复制功能时设置，参数值为自定义字符串。具有远程复制关系的后端名称互为replica_backend，格式为主机名@后端名；
- “manila_huawei_conf_1.xml”、“manila_huawei_conf_2.xml” 为driver自定义的配置文件，请参考章节5 Manila Driver基本属性配置。
- 要使用阵列内远程复制功能，如上添加“local_replication”参数并设置为True，“manila_huawei_conf_1.xml”、“manila_huawei_conf_2.xml”配置对接同一阵列。如不设置该参数或参数值设置为False，表示阵列间远程复制。

步骤2 重新启动Manila服务。

----结束

操作步骤

步骤1 执行**manila type-create replication_type False**命令，创建share类型。
“replication_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

步骤2 执行以下命令，配置远程复制类型。

```
root@ubuntu:~# manila type-key replication_type set replication_type=dr
```

步骤3 执行**manila create --name test001 NFS 2 --share-type replication_type**，创建支持以上属性的share。

步骤4 执行**manila share-replica-create test001**，针对share创建复制。

说明

目前Manila Driver仅支持“dr”类型的远程复制。关于“dr”详见http://docs.openstack.org/developer/manila/devref/share_replication.html。

----结束

6.11 配置 FS 归属控制器

介绍配置FS归属控制器操作步骤。

操作步骤

步骤1 执行**manila type-create controller_type False**命令，创建share类型。
“controller_type”代表share控制器类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

```
root@ubuntu:~# manila type-create controller_type_A False
```

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

```
root@ubuntu:~# manila type-key controller_type_A set capabilities:huawei_controller='<is> true'
```

配置存储设备上已经存在的controllername名称，将controller_type_A 关联到share类型中。

```
root@ubuntu:~# manila type-key controller_type_A set huawei_controller:controllername='CTE0.A'
```

步骤3 执行`manila create --name test001 NFS 2 --share-type controller_type_A`，创建支持以上属性的share。

----结束

6.12 配置 NFS 客户端权限

介绍配置NFS客户端权限操作步骤。

操作步骤

步骤1 执行`manila type-create share_privilege_type False`命令，创建share类型。
“share_privilege_type”代表share控制器类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”

```
root@ubuntu:~# manila type-create share_privilege_type False
```

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

```
root@ubuntu:~# manila type-key share_privilege_type set capabilities:huawei_share_privilege='<is> true'
```

配置存储设备上已经存在的属性类型名称，将share_privilege_type关联到share类型中。

表 6-3 参数取值说明

参数名称	参数说明	参数取值
sync	写入模式	0: 同步 1: 异步
allsquash	权限限制	0: all_squash 1: no_all_squash
rootsquash	Root权限限制	0: root_squash 1: no_root_squash
secure	源端口校验限制	0: secure 1: insecure

```
root@ubuntu:~# manila type-key share_privilege_type set huawei_share_privilege:sync=0
root@ubuntu:~# manila type-key share_privilege_type set huawei_share_privilege:allsquash=0
root@ubuntu:~# manila type-key share_privilege_type set huawei_share_privilege:rootsquash=0
root@ubuntu:~# manila type-key share_privilege_type set huawei_share_privilege:secure=0
```

步骤3 执行`manila create --name test001 NFS 2 --share-type share_privilege_type`，创建支持以上属性的share。

----结束

6.13 配置 HyperMetro

介绍配置HyperMetro属性的操作步骤。

前提条件

需要用户提前在本端和远端阵列上完成双活域、租户、租户pair以及租户可见的逻辑端口的配置。

操作步骤

步骤1 在“/etc/manila/manila.conf”文件中配置有NAS双活关系的两个后端。

```
[DEFAULT]
...
enabled_share_backends = huawei_manila_1, huawei_manila_2
...

[huawei_manila_1]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_1.xml
share_backend_name = huawei_manila_1
driver_handles_share_servers = False
metro_info =
    metro_domain: huawei_domain,
    local_vStore_name: local_vstore,
    remote_vStore_name: remote_vstore,
    remote_backend: host@huawei_manila_2,
    metro_logic_ip: metro_logic_ip

[huawei_manila_2]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_2.xml
share_backend_name = huawei_manila_2
driver_handles_share_servers = False
metro_info =
    metro_domain: huawei_domain,
    local_vStore_name: local_vstore,
    remote_vStore_name: remote_vstore,
    remote_backend: host@huawei_manila_1,
    metro_logic_ip: metro_logic_ip
```

说明

- “metro_info”在需要使用NAS双活功能时设置，参数值为自定义列表，列表包含多个元素，每个元素包含“metro_domain”、“local_vStore_name”、“remote_vStore_name”、“remote_backend”和“metro_logic_ip”字段，用“,”隔开。具体说明参考[表6-4](#)；
- “manila_huawei_conf_1.xml”，“manila_huawei_conf_2.xml”为driver自定义的配置文件，请参考[章节5 Manila Driver基本属性配置](#)；
- 暂不支持从双活share的快照创建share的操作。
- 对dorado V6 6.1.2及以后的版本，Hypermetro Share支持在存储主从切换后创建和删除share，创建删除回滚快照，从快照创建share，扩容缩容share。
- 对dorado V6 6.1.2及以后的版本，如果使用双活后，且volume_type中强制指定后端，请将type中后端名称设置为：<or> backend1 <or> backend2（backend1,backend2为双活后端名称），配置的后端存储池名称要求一致，不然双活主从切换后将没法通过快照创建share。

表 6-4 参数含义说明

属性名称	说明
metro_domain	双活域名称。有双活关系的后端之间，需要设置相同的参数值

属性名称	说明
local_vStore_name	双活租户Pair的本端租户名称。
remote_vStore_name	双活租户Pair的远端租户名称。
remote_backend	具有NAS双活关系的远端Manila后端的名称。构成NAS双活的两个后端互为“remote_backend”，格式为 主机名@后端名
metro_logic_ip	租户下的逻辑IP，构成NAS双活的两个后端配置同一个IP。可填写多个IP，使用分号（";"）进行分割

步骤2 重新启动Manila服务。

```
systemctl restart openstack-manila-share.service
```

步骤3 执行**manila type-create HyperMetro False**命令，创建share类型。

“HyperMetro”代表share类型名称，由用户指定；“False”表示不支持多租户，HyperMetro场景不支持多租户，即不支持该参数配置为“True”。

步骤4 执行以下命令，配置HyperMetro类型键值对。

```
root@ubuntu:~# manila type-key HyperMetro set capabilities:hypermetro='<is> true'
```

步骤5 执行**manila create --name metro001 NFS 2 --share-type HyperMetro**，创建双活share。

----结束

6.14 配置 filesystem:mode

介绍配置共享点源文件系统的文件系统语义层分布算法操作步骤。仅支持OceanStor Dorado存储。

操作步骤

步骤1 执行**manila type-create filesystem_mode_type False**命令，创建share类型。

“filesystem_mode_type”代表share类型名称，由用户指定；“False”表示不支持多租户，若需配置多租户，该参数配置为“True”。

步骤2 执行以下命令，配置filesystem:mode取值。

```
root@ubuntu:~# manila type-key filesystem_mode_type set filesystem:mode=0
```

说明

filesystem:mode=0表示性能模式，filesystem:mode=2表示目录均衡模式

步骤3 执行**manila create --name test001 NFS 2 --share-type filesystem_mode_type**，创建支持以上属性的share。

----结束

7 最佳实践

7.1 快速对接华为存储

7.1 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Huawei Manila Driver对接华为存储系统配置步骤。

步骤1 参考[4.1 Manila Driver获取](#)获取Manila Driver。

步骤2 在华为存储上创建或者查看待使用的文件存储池。

步骤3 配置manila.conf文件和华为自定义Driver配置文件（[5 Manila Driver基本属性配置](#)）。

- 在“/etc/manila”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“manila_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>
    <LogicalPortIP>x.x.x.x</LogicalPortIP>
    <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
    <UserName>xxx</UserName>
    <UserPassword>xxx</UserPassword>
    <SnapshotSupport>True</SnapshotSupport>
  </Storage>
  <Filesystem>
    <StoragePool>xxx</StoragePool>
  </Filesystem>
</Config>
```

- 配置manila.conf文件。

在“/etc/manila/manila.conf”文件的最后添加如下配置项，其中volume_driver表示加载的Driver文件，share_backend_name表示的是manila share 后端的名称，manila_huawei_conf_file表示指定的华为自定义配置文件：

```
[huawei]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
```

```
manila_huawei_conf_file = /etc/manila/manila_huawei_conf.xml
share_backend_name = huawei
driver_handles_share_servers = False
```

在[DEFAULT] 区块中修改以下内容，配置huawei后端：

```
[DEFAULT]
...
enabled_share_backends=huawei
```

步骤4 重新启动Manila服务。

步骤5 检查服务状态。

```
root@openstack:~# manila service-list
+-----+-----+-----+-----+-----+-----+
| Id | Binary      | Host          | Zone | Status | State | Updated_at          |
+-----+-----+-----+-----+-----+-----+
| 1  | manila-scheduler | openstack     | nova | enabled | up    | 2020-10-30T01:26:35.000000 |
| 2  | manila-data      | openstack     | nova | enabled | up    | 2020-10-30T01:26:34.000000 |
| 3  | manila-share     | openstack@huawei | nova | enabled | up    | 2020-10-30T01:26:34.000000 |
+-----+-----+-----+-----+-----+-----+
```

服务状态为**up**，代表服务已正常启动。

----结束