

**eSDK Enterprise Storage Plugins
2.7.4**

OpenStack Manila Driver 配置指南

文档版本	01
发布日期	2025-03-28



版权所有 © 华为技术有限公司 2025。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>

安全声明

产品生命周期政策

华为公司对产品生命周期的规定以“产品生命周期终止政策”为准，该政策的详细内容请参见如下网址：
<https://support.huawei.com/ecolumnsweb/zh/warranty-policy>

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：
<https://www.huawei.com/cn/psirt/vul-response-process>
如企业客户须获取漏洞信息，请参见如下网址：
<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

华为初始证书权责说明

华为公司对随设备出厂的初始数字证书，发布了“华为设备初始数字证书权责说明”，该说明的详细内容请参见如下网址：
<https://support.huawei.com/enterprise/zh/bulletins-service/ENEWS2000015766>

华为企业业务最终用户许可协议(EULA)

本最终用户许可协议是最终用户（个人、公司或其他任何实体）与华为公司就华为软件的使用所缔结的协议。最终用户对华为软件的使用受本协议约束，该协议的详细内容请参见如下网址：
<https://e.huawei.com/cn/about/eula>

产品资料生命周期策略

华为公司针对随产品版本发布的售后客户资料（产品资料），发布了“产品资料生命周期策略”，该策略的详细内容请参见如下网址：
<https://support.huawei.com/enterprise/zh/bulletins-website/ENEWS2000017760>





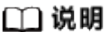
前言

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	2025-03-28	第一次正式发布。

目 录

前言.....	iii
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 非容器化 Manila Share 服务华为 Manila Driver 部署.....	6
4.3 容器化 Manila Share 服务华为 Manila Driver 部署.....	7
5 Manila Driver 基本属性配置.....	9
5.1 非容器化场景配置 Manila Driver.....	9
5.2 容器化场景配置 Manila Driver.....	13
5.3 验证 Manila Share 服务.....	14
6（可选）Manila Driver 高级属性配置.....	15
6.1 配置 Thin/Thick 属性.....	15
6.2 配置 SmartDedupe 属性.....	16
6.3 配置 SmartCompression 属性.....	17
6.4 配置 SmartCache 属性.....	17
6.5 配置 SmartPartition 属性.....	17
6.6 配置 SmartQoS 属性.....	18
6.6.1 配置控制最大 IOPS.....	19
6.6.2 配置控制最小 IOPS.....	19
6.6.3 配置控制最大带宽.....	20
6.6.4 配置控制最小带宽.....	20
6.6.5 配置控制最大时延.....	21
6.6.6 配置控制多属性参数.....	21
6.7 指定存储池创建共享.....	22
6.8 指定磁盘类型创建共享.....	22
6.9 配置 SectorSize 属性.....	23
6.10 配置远程复制属性.....	24
6.11 配置文件系统归属控制器.....	25
6.12 配置 NFS 客户端权限.....	25

6.13 配置 HyperMetro 属性.....	26
6.14 配置 filesystem:mode.....	28
6.15 配置快照预留空间百分比.....	28
6.16 配置文件系统根目录 UNIX 权限.....	28
6.17 配置自定义快照目录可见.....	29
6.18 配置证书认证.....	29
7 最佳实践.....	31
7.1 快速对接华为存储.....	31

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
Mitaka/Newton/ Ocata/Pike/Queens/ Rocky/Stein/Train/ Ussuri/Victoria/ Wallaby/Xena/ Yoga/Zed/Antelope/ Bobcat/Caracal/ Dalmatian	<ul style="list-style-type: none">• OceanStor V5: V500R007C61/V500R007C70• OceanStor 6.1.3/6.1.5/6.1.6/6.1.7/6.1.8/ V700R001C00• OceanStor Dorado 6.1.0/6.1.2/6.1.3/6.1.5/6.1.6/6.1.7/6.1.8/ V700R001C00

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	Mitaka/ Newton/ Ocata/Pike/ Queens/ Rocky/Stein/ Train/Ussuri/ Victoria/ Wallaby/Xena/ Yoga/Zed/ Antelope/ Bobcat/ Caracal/ Dalmatian	Mitaka/ Newton/ Ocata/Pike/ Queens/ Rocky/Stein/ Train/Ussuri/ Victoria/ Wallaby/Xena/ Yoga/Zed/ Antelope/ Bobcat/ Caracal/ Dalmatian	N/A
Create Share	√	√	N/A

特性	Normal Share	HyperMetro Share	备注
Delete Share	√	√	N/A
Allow access	√	√	N/A
Deny access	√	√	N/A
Create Snapshot	√	√	HyperMetro share只支持在双活本端创建快照。
Delete Snapshot	√	√	HyperMetro share只支持在双活本端删除快照。
Revert Snapshot	√	√	HyperMetro share只支持在双活本端回滚快照。
Extend Share	√	√	N/A
Shrink Share	√	√	N/A
Create Share from Snapshot	√	√	HyperMetro share只支持在双活本端从快照创建共享。
Manage/Unmanage Share	√	x	Manila Driver管理的文件系统和共享关系必须是一一对应。 只支持管理系统租户下的文件系统和共享。
Manage/Unmanage Snapshot	√	x	N/A
Replication	√	x	N/A
QoS	√	√	HyperMetro share只支持在双活本端配置SmartQoS。
Multi-tenancy	√	x	N/A
SmartThin/ SmartThick	√	√	N/A
SmartCompression	√	√	N/A
SmartDedupe	√	√	N/A
Ensure Share	√	√	N/A
Create a share on a certain storage pool	√	√	N/A

特性	Normal Share	HyperMetro Share	备注
Create a share with a certain disk type	√	√	N/A
SectorSize	√	√	N/A
SmartPartition	√	√	HyperMetro share只支持在双活本端配置SmartPartition。
SmartCache	√	√	HyperMetro share只支持在双活本端配置SmartCache。

说明

- OceanStor Dorado NAS架构发生变化，以上涉及双活本端的特性以实际存储为准。
- 对OceanStor Dorado 6.1.2及以后的版本，HyperMetro share支持在存储主从切换后创建/删除share、创建/删除/回滚快照、从快照创建share、扩容/缩容share。

表 2-3 Manila Driver 各特性与存储关系配套列表

特性	OceanStor V5	OceanStor 6.1.x	OceanStor Dorado
Thin类型共享	支持	支持	支持
Thick类型共享	支持	不支持	不支持
SmartDedupe	支持(仅Thin类型的share)	支持(仅Thin类型的share)	支持(仅Thin类型的share)
SmartCompression	支持(仅Thin类型的share)	支持(仅Thin类型的share)	支持(仅Thin类型的share)
SmartCache	支持	不支持	支持
SmartPartition	支持	不支持	不支持
SmartQoS <ul style="list-style-type: none">• 保护策略：latency、minIOPS、minBandWidth• 限制策略：maxIOPS、maxBandWidth	支持，保护策略和限制策略不能同时配置	支持，保护策略和限制策略不能同时配置	支持，保护策略和限制策略不能同时配置

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： Mitaka/Newton/Ocata/Pike/ Queens/Rocky/Stein/Train/ Ussuri/Victoria/Wallaby/ Xena/Yoga/Zed/Antelope	-
配置	XML配置文件	<ul style="list-style-type: none">Manila 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Manila 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	<ul style="list-style-type: none">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。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。	<ul style="list-style-type: none">限制策略： maxIOPS， maxBandWidth保护策略： minIOPS， minBandWidth ， latency

4 Manila Driver 安装与部署

4.1 Manila Driver获取

4.2 非容器化Manila Share服务华为Manila Driver部署

4.3 容器化Manila Share服务华为Manila Driver部署

4.1 Manila Driver 获取

操作步骤

- 步骤1** 打开浏览器，访问仓库地址：https://github.com/Huawei/OpenStack_Driver。
- 步骤2** 选择“Code”>“Download ZIP”，华为Manila 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，目录中包含多个OpenStack版本的华为Manila Driver。

说明

华为自有OpenStack Manila Driver仓库维护M-D总共18个稳定版本，能够保障历史版本长期稳定运行。

----结束

4.2 非容器化 Manila Share 服务华为 Manila 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Manila Driver代码目录。
- 相对路径为：.../manila/share/drivers/huawei。
 - 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Manila Driver代码目录：
 - 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']
```
- 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，复制后结果如下。

#### 须知

只需拷贝覆盖，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

## 4.3 容器化 Manila Share 服务华为 Manila Driver 部署

**步骤1** 登录具有Manila Share容器镜像的节点，执行**docker image ls**命令查询Manila Share容器镜像。

```
[root@openstack-node1 cinder]# vi oceanstor_vs_system.xml
[root@openstack-node1 cinder]# docker image ls | grep manila
quay.io/openstack.kolla/manila-share 2023.1-rocky-9 0158ef4bc21b 3 weeks ago 1.59GB
quay.io/openstack.kolla/manila-data 2023.1-rocky-9 7ce721d4aae8 3 weeks ago 1.36GB
quay.io/openstack.kolla/manila-scheduler 2023.1-rocky-9 9e688df52db6 3 weeks ago 1.33GB
quay.io/openstack.kolla/manila-api 2023.1-rocky-9 e2da71544592 3 weeks ago 1.33GB
```

**步骤2** 执行**docker save -o manila\_share.tar quay.io/openstack.kolla/manila-share**命令备份Manila Share容器镜像到镜像文件。

```
[root@openstack-node1 cinder]# docker save -o manila_share.tar quay.io/openstack.kolla/manila-share
```

#### 说明

**manila\_share.tar**为自定义备份镜像的名称，**quay.io/openstack.kolla/manila-share**为Manila Share容器镜像名称。

**步骤3** 在任意目录下创建临时目录**huawei**。

```
mkdir huawei
```

**步骤4** 将获取到的对应版本的华为Manila Driver代码文件拷贝到创建的**huawei**目录下。Manila Driver代码文件最小权限要求为644。

```
ls -l
-rw-r--r-- 1 root root 4661 Dec 7 19:14 constants.py
-rw-r--r-- 1 root root 44837 Dec 7 19:14 helper.py
-rw-r--r-- 1 root root 12795 Dec 7 19:14 huawei_config.py
-rw-r--r-- 1 root root 98273 Dec 7 19:13 huawei_nas.py
```

```
-rw-r--r--. 1 root root 12082 Dec 7 19:13 huawei_utils.py
-rw-r--r--. 1 root root 7684 Dec 7 19:13 hypermetro.py
-rw-r--r--. 1 root root 21 Dec 7 19:14 __init__.py
-rw-r--r--. 1 root root 4285 Dec 7 19:13 manager.py
-rw-r--r--. 1 root root 12870 Dec 7 19:13 replication.py
-rw-r--r--. 1 root root 10835 Dec 7 19:13 rpcapi.py
-rw-r--r--. 1 root root 4357 Dec 7 19:14 smartx.py
```

**步骤5** 执行**docker ps | grep manila**命令列出Manila Share容器，**manila\_share**为容器名称。

```
root@openstack-node1:~# docker ps | grep manila
20f1ca414865 quay.io/openstack.kolla/manila-share:2023.1-rocky-9 "dumb-init --single-" 3 weeks ago Up 2 days (healthy) manila_share
cb316d62078d quay.io/openstack.kolla/manila-scheduler:2023.1-rocky-9 "dumb-init --single-" 3 weeks ago Up 2 days (healthy) manila_scheduler
5a5376fe8171 quay.io/openstack.kolla/manila-data:2023.1-rocky-9 "dumb-init --single-" 3 weeks ago Up 2 days (healthy) manila_data
ecb9b65f9011 quay.io/openstack.kolla/manila-api:2023.1-rocky-9 "dumb-init --single-" 3 weeks ago Up 2 days (healthy) manila_api
```

**步骤6** 执行**docker exec -it -u 0 manila\_share bash**命令进入**manila\_share**容器。

```
[root@control ~]# docker exec -it -u 0 cinder_volume bash
(cinder-volume)[root@control /]#
```

**步骤7** 执行**python -c "from manila.share import drivers; print (drivers.\_\_path\_\_)"**命令查询原有的华为Manila Driver代码的路径。

```
(manila-share)[root@openstack-node1 /]# python -c "from manila.share import drivers; print (drivers.__path__)"
['/var/lib/kolla/venv/lib64/python3.9/site-packages/manila/share/drivers']
```

#### 说明

- **manila\_share**为**步骤5**查询出来的容器名称。
- **/var/lib/kolla/venv/lib64/python3.9/site-packages/manila/share/drivers**为原有的华为Manila Driver代码的路径。
- 如果执行上述命令报错，请根据Python环境变量，变更命令为**python3 -c "from manila.share import drivers; print (drivers.\_\_path\_\_)"**。

**步骤8** 执行**exit**命令，退出**manila\_share**容器，进入**步骤3**创建的**huawei**目录的上级目录，将**huawei**目录拷贝到**步骤7**中原有的华为Manila Driver代码的路径。

```
(manila-share)[root@openstack-node1 /]# exit
exit
[root@openstack-node1 huawei]# cd ..
[root@openstack-node1 huangpeng]# docker cp huawei/ manila_share:/var/lib/kolla/venv/lib64/python3.9/site-packages/manila/share/drivers
```

```
docker cp huawei/ manila_share:/var/lib/kolla/venv/lib64/python3.9/site-packages/manila/share/drivers
```

**步骤9** 执行以下命令，重启Manila Share服务，**manila\_share**为**步骤5**获取到的容器名称。

```
docker restart manila_share
```

等待一段时间后（一般30秒左右），执行**manila service-list**命令。有以下回显时，查看**State**的值为**up**，表示Manila Share服务启动成功。

```
[root@openstack-node1 huangpeng]# manila service-list
manila CLI is deprecated and will be removed in the future. Use openstack CLI instead. The equivalent command is "openstack share service list"
+-----+-----+-----+-----+-----+-----+-----+
| Id | Binary | Host | Zone | Status | State | Updated_at |
+-----+-----+-----+-----+-----+-----+-----+
3	manila-data	openstack-node1	nova	enabled	up	2023-12-14T08:24:58.141786
9	manila-data	openstack-node3	nova	enabled	up	2023-12-14T08:25:02.549368
15	manila-scheduler	openstack-node1	nova	enabled	up	2023-12-14T08:25:04.463347
18	manila-scheduler	openstack-node3	nova	enabled	up	2023-12-14T08:25:01.462897
20	manila-share	openstack-node1	nova	enabled	up	2023-12-04T13:34:35.978287
23	manila-share	openstack-node1@dorado-v6-system-iscsi-primary	nova	enabled	up	2023-12-14T08:25:02.575048
26	manila-share	openstack-node1@dorado-v6-system-iscsi-secondary	nova	enabled	up	2023-12-14T08:25:06.677289
29	manila-share	openstack-node1@oceanstor-v5-system-iscsi-primary	nova	enabled	up	2023-12-14T08:25:02.773277
32	manila-share	openstack-node1@oceanstor-v5-system-iscsi-secondary	nova	enabled	up	2023-12-14T08:25:02.972588
```

----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创建共享时，如未指定share type，会使用“/etc/manila/manila.conf”文件中的默认share type。
- Manila 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Manila 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 非容器化场景配置Manila Driver

### 5.2 容器化场景配置Manila Driver

### 5.3 验证Manila Share服务

## 5.1 非容器化场景配置 Manila Driver

### 操作步骤

**步骤1** 在“/etc/manila”路径下创建华为Manila 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”文件，配置华为Manila Driver配置文件必填参数。

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
 <Storage>
 <Product>Dorado</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 必填参数说明

参数	说明	默认值	备注
Product	存储产品类型，可选值为V5、V6或Dorado。	-	-
LogicalPortIP	逻辑端口的IP地址。支持配置多个IP地址，以“;”分隔。	-	“DNS”参数不配置时必选。两个参数同时配置时，本参数的优先级低于“DNS”参数。
DNS	DNS域名。支持配置多个DNS域名，以“;”分隔。	-	“LogicalPortIP”参数不配置时必选。两个参数同时配置时，本参数的优先级高于“LogicalPortIP”参数。
RestURL	REST接口访问地址。	-	-
UserName	存储管理员用户名。 <b>须知</b> 请避免直接使用“超级管理员”权限账户。	-	-
UserPassword	存储管理员密码。	-	-
StoragePool	需要使用的存储池名称。支持配置多个存储池，以“;”分隔。	-	-

步骤3 编辑“manila\_huawei\_conf.xml”文件，配置华为Manila 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>
 <UnixPermission>755</UnixPermission>
 <ShowSnapshotDir>true</ShowSnapshotDir>
 <NFSClient>
 <IP>x.x.x.x</IP>
 </NFSClient>
 <CIFSCClient>
 <UserName>xxx</UserName>
 <UserPassword>xxx</UserPassword>
 </CIFSCClient>
 </Filesystem>
</Config>
```



表 5-2 选填参数说明

参数	说明	默认值	类型
Port	可用的业务端口（绑定端口或者物理端口）。支持配置多个端口，以“;”分隔。	-	“driver_handles_share_servers”参数为“True”时可选，如不配置，则使用所有的online端口。
SectorSize	文件系统磁盘块的大小，可选值“4”、“8”、“16”、“32”、“64”，单位KB。	64	可选。
SnapshotSupport	是否提供快照功能。	True	除OceanStor Dorado外，其余存储不可同时开启SnapshotSupport和ReplicationSupport功能。
ReplicationSupport	是否提供远程复制功能。	False	除OceanStor Dorado外，其余存储不可同时开启SnapshotSupport和ReplicationSupport功能。
WaitInterval	查询文件系统状态的间隔时间，单位为秒（s）。	3	可选。
Timeout	等待存储设备执行命令的超时时间，单位为秒（s）。	60	可选。
SnapshotReserve	快照预留空间百分比（%）。范围为0~50。	0	可选。

参数	说明	默认值	类型
UnixPermission	文件系统根目录权限	755	可选，取值共3个部分，由3个数字表示，含义如下：第1个数字表示拥有者权限；第2个数字表示所属组用户权限；第3个数字表示任何人权限。每个部分的取值为数字0-7，含义如下：0：没有权限；1：可执行；2：可写；3：可写+可执行；4：可读；5：可读+可执行；6：可读+可写；7：可读+可写+可执行。
ShowSnapshotDir	快照目录是否可见	true	可选，取值为： true: 可见 false: 不可见
NFSClient\IP	从快照创建共享时，管理节点上用于挂载NFS共享所用的IP地址。	-	从快照创建共享时必选。
CIFSClient\UserName	从快照创建共享时，管理节点上用于挂载CIFS共享所用的用户名。	-	从快照创建共享时必选。使用OceanStor Dorado存储时，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共享所用的用户密码。	-	从快照创建共享时必选。

**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 Manila 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 Manila Driver 配置文件路径。
- **driver\_handles\_share\_servers** 为 **True** 时创建 share 需要指定 **share-network**。

**步骤5** 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6** 修改 “/etc/manila/manila\_huawei\_conf.xml” 文件的权限，使其与 “/etc/manila/manila.conf” 文件的权限一致。

```
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7** 执行以下命令，重新启动 Manila 服务。

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

----结束

## 5.2 容器化场景配置 Manila Driver

### 操作步骤

**步骤1** 使用远程访问工具（以 PuTTY 为例），通过管理 IP 地址，登录 OpenStack 的 Manila Share 节点。在主机上执行如下命令查找 **manila.conf**，并按照 **步骤4** 和 **步骤5** 配置相应参数。

```
find / -name manila.conf | grep manila
```

**步骤2** 执行命令 **docker ps | grep manila** 查询 Manila Share 容器，然后执行命令 **docker exec -it -u 0 manila\_share bash** 进入 Manila Share 容器。

```
[root@openstack-node1 huangpeng]# docker ps | grep manila
20f1ca-f140e9 quay.io/openstack.kolla/manila-share:2023.1-rocky-9 "dumb-init --single..." 3 weeks ago Up 2 days (healthy) manila_share
cb316d62078d quay.io/openstack.kolla/manila-scheduler:2023.1-rocky-9 "dumb-init --single..." 3 weeks ago Up 2 days (healthy) manila_scheduler
5a537f6e8171 quay.io/openstack.kolla/manila-data:2023.1-rocky-9 "dumb-init --single..." 3 weeks ago Up 2 days (healthy) manila_data
ecb9b65f9011 quay.io/openstack.kolla/manila-api:2023.1-rocky-9 "dumb-init --single..." 3 weeks ago Up 2 days (healthy) manila_api
```

```
[root@openstack-node1 huangpeng]# docker exec -it -u 0 manila_share bash
(manila-share)[root@openstack-node1 /]#
```

### 说明

**manila\_share** 为查询出来的 Manila Share 容器名称。

**步骤3** 在 Manila Share 容器中 “/etc/manila” 路径下创建华为 Manila Driver 配置文件，文件格式为 xml，并按照 **步骤2** 和 **步骤3** 配置相应参数。配置文件名称可自定义，例如 “manila\_huawei\_conf.xml”。

**步骤4** 修改华为 Manila Driver 配置文件的拥有者及用户组与 “/etc/manila/manila.conf” 文件的拥有者及用户组一致。

```
-rw----- 1 manila manila 2962 Sep 22 07:17 manila.conf
-rw----- 1 manila manila 443 Sep 14 03:43 manila_huawei_conf.xml
```

**步骤5** 执行以下命令，重启Manila Share服务。

```
docker restart manila_share
```

等待一段时间后（一般30秒左右），执行**manila service-list**命令。有以下回显时，查看**State**的值为**up**，表示Manila Share服务启动成功。

```
[root@openstack-node1 huangpengj]# manila service-list
manila CLI is deprecated and will be removed in the future. Use openstack CLI instead. The equivalent command is 'openstack share service list'
+-----+-----+-----+-----+-----+-----+
| Id | Binary | Host | Zone | Status | State | Updated_at |
+-----+-----+-----+-----+-----+-----+
| 3 | manila-data | openstack-node1 | nova | enabled | up | 2023-12-14T09:58:10.390376 |
| 9 | manila-data | openstack-node3 | nova | enabled | up | 2023-12-14T09:58:12.746916 |
| 15 | manila-scheduler | openstack-node1 | nova | enabled | up | 2023-12-14T09:58:15.878819 |
| 18 | manila-scheduler | openstack-node3 | nova | enabled | up | 2023-12-14T09:58:11.826777 |
| 20 | manila-share | openstack-node1 | nova | enabled | up | 2023-12-04T13:34:35.978287 |
```

----结束

## 5.3 验证 Manila Share 服务

### 注意事项

除OceanStor Dorado NAS外，快照属性与远程复制属性互斥，两者不能同时配置。

### 操作步骤

**步骤1** 创建通用share类型。

其中，**general**为share类型名称，由用户指定。**False**表示不启用  
**driver\_handles\_share\_servers**功能，若需启用此功能，该参数配置为**True**。  
# manila type-create general False

**步骤2** 设置share类型后端。

其中，**huawei\_backend**为**步骤4**中配置的share的后端名称。  
# 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** 创建支持以上属性的share。

# manila create --name test001 NFS 2 --share-type general

----结束

# 6（可选）Manila Driver 高级属性配置

---

本章介绍配置高级存储属性的操作步骤。

华为存储支持多种Smart系列高级特性，这些高级特性通过与特定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 指定存储池创建共享](#)

[6.8 指定磁盘类型创建共享](#)

[6.9 配置SectorSize属性](#)

[6.10 配置远程复制属性](#)

[6.11 配置文件系统归属控制器](#)

[6.12 配置NFS客户端权限](#)

[6.13 配置HyperMetro属性](#)

[6.14 配置filesystem:mode](#)

[6.15 配置快照预留空间百分比](#)

[6.16 配置文件系统根目录UNIX权限](#)

[6.17 配置自定义快照目录可见](#)

[6.18 配置证书认证](#)

## 6.1 配置 Thin/Thick 属性

介绍配置Thin/Thick属性的操作步骤。

## 操作步骤

**步骤1** 执行**manila type-create thin\_type False**命令，创建share类型。

其中，thin\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，该参数配置为True。

**步骤2** 配置Thin/Thick属性。

### 说明

- 配置thin\_provisioning属性为true时，共享类型为Thin；配置thin\_provisioning属性为false时，共享类型为Thick。
- share-type中未配置thin\_provisioning时，共享默认类型为Thin。
- OceanStor Dorado 6.x 和OceanStor 6.x不支持配置类型为Thick的共享。

- 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**命令，创建支持以上属性的share。

----结束

## 6.2 配置 SmartDedupe 属性

本节介绍配置SmartDedupe属性的操作步骤。

### 注意事项

仅共享类型为Thin时，支持配置SmartDedupe。

## 操作步骤

**步骤1** 执行**manila type-create dedupe\_type False**命令，创建share类型。

其中，dedupe\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，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**命令，创建支持以上属性的share。

----结束

## 6.3 配置 SmartCompression 属性

本节介绍配置SmartCompression属性的操作步骤。

### 注意事项

仅共享类型为Thin时，支持配置SmartCompression。

### 操作步骤

**步骤1** 执行**manila type-create compression\_type False**命令，创建share类型。

其中，compression\_type为share类型名称，由用户指定。False表示不启用**driver\_handles\_share\_servers**功能，若需启用此功能，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**命令，创建支持以上属性的share。

----结束

## 6.4 配置 SmartCache 属性

本节介绍配置SmartCache属性的操作步骤。

### 操作步骤

**步骤1** 执行**manila type-create cache\_type False**命令，创建share类型。

其中，cache\_type为share类型名称，由用户指定。False表示不启用**driver\_handles\_share\_servers**功能，若需启用此功能，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'
```

**步骤3** 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4** 执行**manila create --name test001 NFS 2 --share-type cache\_type**命令，创建支持以上属性的share。

----结束

## 6.5 配置 SmartPartition 属性

本节介绍配置SmartPartition属性的操作步骤。

## 操作步骤

**步骤1** 执行**manila type-create partition\_type False**命令，创建share类型。

其中，partition\_type为share类型名称，由用户指定。False表示不启用**driver\_handles\_share\_servers**功能，若需启用此功能，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'
```

**步骤3** 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:partitionname='test_name'
```

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

----结束

## 6.6 配置 SmartQoS 属性

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。

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”。

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"><li>• 0：读I/O</li><li>• 1：写I/O</li><li>• 2：读写I/O</li></ul>

表 6-2 QoS 参数说明

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。



参数名称	说明	备注
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。

## 6.6.1 配置控制最大 IOPS

本节介绍配置控制最大IOPS的操作步骤。

### 操作步骤

**步骤1** 执行**manila type-create maxiops\_type False**命令，创建share类型。

其中，maxiops\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，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'
```

**步骤3** 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I/O。
  - “1”表示控制写I/O。
  - “2”表示控制读写I/O。

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

----结束

## 6.6.2 配置控制最小 IOPS

本节介绍配置控制最小IOPS的操作步骤。

### 操作步骤

**步骤1** 执行**manila type-create miniops\_type False**命令，创建share类型。

其中，miniops\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，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'
```

**步骤3** 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I/O。
  - “1”表示控制写I/O。
  - “2”表示控制读写I/O。

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

----结束

## 6.6.3 配置控制最大带宽

本节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。

其中，maxbandwidth\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，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'
```

**步骤3** 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I/O。
  - “1”表示控制写I/O。
  - “2”表示控制读写I/O。

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

----结束

## 6.6.4 配置控制最小带宽

本节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。

其中，minbandwidth\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，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'
```

**步骤3** 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I/O。
  - “1”表示控制写I/O。
  - “2”表示控制读写I/O。

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

----结束

## 6.6.5 配置控制最大时延

本节介绍配置控制最大时延的操作步骤。

### 操作步骤

**步骤1** 执行**manila type-create latency\_type False**命令，创建share类型。

其中，latency\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，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'
```

**步骤3** 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：最大时延限制，有效值为大于0的整数，单位为ms。对于OceanStor Dorado系列，数值为500或者1500的整数。
- IOType（必选）：控制读写类型。
  - “0”表示控制读I/O。
  - “1”表示控制写I/O。
  - “2”表示控制读写I/O。

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

----结束

## 6.6.6 配置控制多属性参数

本节介绍配置控制多属性参数的操作步骤。

### 操作步骤

**步骤1** 执行**manila type-create multiple\_strategy\_type False**命令，创建share类型。

其中，multiple\_strategy\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，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'
```

**步骤3** 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：最大时延限制，有效值为大于0的整数，单位为ms。对于OceanStor Dorado系列，数值为500或者1500的整数。
- minIOPS：最小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。
- 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I/O。
  - “1”表示控制写I/O。
  - “2”表示控制读写I/O。

**步骤4** 执行**manila create --name test001 NFS 2 --share-type multiple\_strategy\_type**命令，创建支持以上属性的share。

----结束

## 6.7 指定存储池创建共享

本节介绍配置指定存储池创建共享的操作步骤。

### 操作步骤

**步骤1** 执行**manila type-create target\_pool\_type False**命令，创建share类型。

其中，target\_pool\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，该参数配置为True。

**步骤2** 执行以下命令，配置目标存储池。

- 若配置单个目标存储池，执行以下命令：

```
root@ubuntu:~# manila type-key target_pool_type set pool_name=StoragePool001
```
- 若配置多个目标存储池，执行以下命令：

```
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**命令，创建支持以上属性的share。

----结束

## 6.8 指定磁盘类型创建共享

本节介绍配置指定磁盘类型创建共享的操作步骤。

## 操作步骤

**步骤1** 执行**manila type-create disk\_type False**命令，创建share类型。

其中，disk\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，该参数配置为True。

**步骤2** 执行以下命令，配置磁盘类型。

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

- 若配置一种磁盘类型，执行以下命令：

```
root@ubuntu:~# manila type-key disk_type set huawei_disk_type=sas
```

- 若配置多种磁盘类型，执行以下命令：

```
root@ubuntu:~# manila type-key disk_type set huawei_disk_type="<or> sas <or> ssd"
```

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

----结束

## 6.9 配置 SectorSize 属性

本节介绍配置共享源文件系统的SectorSize属性（磁盘块大小）操作步骤。

### 注意事项

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

### 操作步骤

**步骤1** 执行**manila type-create sectorsize\_type False**命令，创建share类型。

其中，sectorsize\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，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'
```

**步骤3** 配置SectorSize的值。

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

----结束

## 6.10 配置远程复制属性

本节介绍配置远程复制属性的操作步骤。

### 前提条件

在“/etc/manila/manila.conf”文件中配置有远程复制关系的两个后端并重新启动Manila服务。

- `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。具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互为`replica_backend`，名称格式为：`主机名@后端名`。
- “`manila_huawei_conf_1.xml`”和“`manila_huawei_conf_2.xml`”为Manila Driver的自定义配置文件，具体请参考[5 Manila Driver基本属性配置](#)章节。
- 要使用存储阵列内的远程复制功能，如上添加“`local_replication`”参数并设置为True，且“`manila_huawei_conf_1.xml`”和“`manila_huawei_conf_2.xml`”设置相同的存储阵列参数。如不设置“`local_replication`”参数或该参数设置为False，表示使用存储阵列间的远程复制功能。

```
[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
```

### 操作步骤

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

其中，`replication_type`为share类型名称，由用户指定。False表示不启用`driver_handles_share_servers`功能，若需启用此功能，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
```

### 说明

目前Manila Driver仅支持“dr”类型的远程复制。关于“dr”详细说明：[点此前往](#)。

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

**步骤4** 执行**manila share-replica-create test001**命令，创建share replica，使其与**步骤3**中创建的share构成远程复制关系。

----结束

## 6.11 配置文件系统归属控制器

本节介绍配置文件系统归属控制器的操作步骤。

### 操作步骤

**步骤1** 执行**manila type-create controller\_type False**命令，创建share类型。

其中，controller\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，该参数配置为True。  
root@ubuntu:~# manila type-create controller\_type\_A False

**步骤2** 配置存储设备上已经存在的控制器名称，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**命令，创建支持以上属性的share。

----结束

## 6.12 配置 NFS 客户端权限

本节介绍配置NFS客户端权限的操作步骤。

### 操作步骤

**步骤1** 执行**manila type-create share\_privilege\_type False**命令，创建share类型。

其中，share\_privilege\_type为share类型名称，由用户指定。False表示不启用driver\_handles\_share\_servers功能，若需启用此功能，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 参数说明

参数名称	参数说明	参数取值
sync	写入模式。	<ul style="list-style-type: none"><li>0：同步</li><li>1：异步</li></ul>

参数名称	参数说明	参数取值
allsquash	权限限制。	<ul style="list-style-type: none"><li>0: all_squash</li><li>1: no_all_squash</li></ul>
rootsquash	root权限限制。	<ul style="list-style-type: none"><li>0: root_squash</li><li>1: no_root_squash</li></ul>
secure	源端口校验限制。	<ul style="list-style-type: none"><li>0: secure</li><li>1: insecure</li></ul>

```
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**命令，创建支持以上属性的share。

----结束

## 6.13 配置 HyperMetro 属性

本节介绍配置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。

- “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”和“manila\_huawei\_conf\_2.xml”为Manila Driver的自定义配置文件，具体请参考5 [Manila Driver基本属性配置](#)章节。
- 暂不支持从双活share的快照创建share的操作。
- 对OceanStor Dorado 6.1.2及以后的版本，HyperMetro share支持在存储主从切换后创建/删除share、创建/删除/回滚快照、从快照创建share、扩容/缩容share。
- 对OceanStor Dorado 6.1.2及以后的版本，如果使用双活，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必须一致，否则双活主从切换后将无法通过快照创建share。

[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
```

表 6-4 参数说明

参数名称	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。
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，名称格式为： <i>主机名@后端名</i> 。
metro_logic_ip	租户下的逻辑IP地址，构成NAS双活的两个后端配置同一个IP地址。可填写多个IP地址，使用分号";"进行分割。

**步骤2** 执行以下命令，重新启动Manila服务。

```
root@ubuntu:~# systemctl restart openstack-manila-share.service
```

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

其中，HyperMetro为share类型名称，由用户指定。False表示不启用**driver\_handles\_share\_servers**功能，HyperMetro不支持该参数配置为True。

**步骤4** 执行以下命令，配置HyperMetro属性为“true”的键值对。

```
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**命令，创建支持以上属性的share。

----结束

## 6.14 配置 filesystem:mode

本节介绍配置共享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。

其中，filesystem\_mode\_type为share类型名称，由用户指定。False表示不启用**driver\_handles\_share\_servers**功能，若需启用此功能，该参数配置为True。

**步骤2** 执行以下命令，配置filesystem:mode的值。

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

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

**步骤3** 执行**manila create --name test001 NFS 2 --share-type filesystem\_mode\_type**命令，创建支持以上属性的share。

----结束

## 6.15 配置快照预留空间百分比

本节介绍配置快照预留空间百分比的操作步骤。

### 操作步骤

**步骤1** 执行**manila type-create snapshot\_reserve False**命令，创建share类型。

其中，snapshot\_reserve为share类型名称，由用户指定。False表示不启用**driver\_handles\_share\_servers**功能，若需启用此功能，该参数配置为True。

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

**步骤2** 执行以下命令，配置snapshot\_reserve\_percentage的值。

```
root@ubuntu:~# manila type-key snapshot_reserve set
huawei_snapshotreserveper:snapshot_reserve_percentage=20
```

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

----结束

## 6.16 配置文件系统根目录 UNIX 权限

本节介绍配置文件系统根目录UNIX权限的操作步骤。

### 操作步骤

**步骤1** 执行**manila type-create unix\_permission False**命令，创建share类型。

其中，`unix_permission`为share类型名称，由用户指定。False表示不启用 **driver\_handles\_share\_servers**功能，若需启用此功能，该参数配置为True。

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

**步骤2** 执行以下命令，配置`unix_permission`的值。

```
root@ubuntu:~# manila type-key unix_permission set huawei_unixpermission:unix_permission=755
```

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

----结束

## 6.17 配置自定义快照目录可见

本节介绍配置自定义快照目录可见的操作步骤。

### 操作步骤

**步骤1** 执行**manila type-create show\_snapshot\_dir False**命令，创建share类型。

其中，`show_snapshot_dir`为share类型名称，由用户指定。False表示不启用 **driver\_handles\_share\_servers**功能，若需启用此功能，该参数配置为True。

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

**步骤2** 执行以下命令，配置`show_snapshot_dir`的值。

```
root@ubuntu:~# manila type-key show_snapshot_dir set huawei_showsnapshotdir:show_snapshot_dir=true
```

**步骤3** 执行**manila create --name test001 NFS 2 --share-type show\_snapshot\_dir**命令，创建支持以上属性的share。

----结束

## 6.18 配置证书认证

本节介绍配置通过证书认证方式连接后端存储。

### 操作步骤

**步骤1** 使用远程访问工具（以PuTTY为例），通过管理IP地址，登录OpenStack的Malina节点。

**步骤2** 配置华为Malina 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>***</SSLCertPath>
 </Storage>
 ...
</config>
```

#### 说明

- **SSLCertVerify**表示是否开启证书认证。有效值为True或False，未配置时默认为False，建议开启证书校验。
- **SSLCertPath**指定用于认证的证书路径，仅在**SSLCertVerify**为True的情况下有效。

----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

介绍在OpenStack上通过配置华为Manila Driver对接华为存储系统。

### 配置流程

**步骤1** 参考[4.1 Manila Driver获取](#)章节，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** 参考[5 Manila Driver基本属性配置](#)章节，配置“manila.conf”文件和华为自定义Manila Driver配置文件。

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

- 配置华为自定义Manila Driver配置文件参数。

```
<?xml version='1.0' encoding='UTF-8'?>
<Config>
 <Storage>
 <Product>Dorado</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Manila Driver文件。
- share\_backend\_name表示Manila share后端的名称。
- manila\_huawei\_conf\_file表示华为自定义Manila Driver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。

如果服务状态为up，代表服务已正常启动。

```
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 |
+-----+-----+-----+-----+-----+-----+
```

----结束