eSDK Enterprise Storage Plugins 2.3.RC1

OpenStack Manila Driver Configuration Guide

Issue 01

Date 2021-07-09





Copyright © Huawei Technologies Co., Ltd. 2021. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base

Bantian, Longgang Shenzhen 518129

People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Contents

1 Overview	1
2 Version Mapping	2
3 Specifications and Restrictions	5
4 Deployment	7
4.1 Obtaining Manila Driver	
4.2 Deploying Manila Driver for the Manila Share Service	8
5 Configuring Basic Properties	9
5.1 Configuring Manila Driver	9
5.2 Verifying the Manila Share Service	13
6 Configuring Advanced Properties	14
6.1 Configuring Thin/Thick	14
6.2 Configuring SmartDedupe	15
6.3 Configuring SmartCompression	15
6.4 Configuring SmartCache	15
6.5 Configuring SmartPartition	16
6.6 Configuring SmartQoS	16
6.6.1 Configuring the Maximum Control IOPS	16
6.6.2 Configuring the Minimum Control IOPS	17
6.6.3 Configuring the Maximum Control Bandwidth	17
6.6.4 Configuring the Minimum Control Bandwidth	18
6.6.5 Configuring the Control Latency	18
6.6.6 Configuring Multiple Control Policies	18
6.7 Creating a Share on a Specified Storage Pool	19
6.8 Creating a Share with a Specified Disk Type	19
6.9 Configuring SectorSize	20
6.10 Configuring Replication	21
6.11 Configuring the Owning Controller of an FS	22
6.12 Configuring the NFS Client Permission	
6.13 Configuring HyperMetro	
6.14 Configuring filesystem:mode	25
7 Best Practices	26

SDK Enterprise Storage Plugins	
DpenStack Manila Driver Configuration Guid	e

Contents

1 Overview

This chapter describes the definition of Manila Driver.

Manila Driver is a plug-in that is deployed on the OpenStack Manila module. The plug-in can be used to provide functions such as share and snapshot for virtual machines (VMs) in OpenStack.

2 Version Mapping

This chapter describes versions and functions supported by Manila Driver, and its version mappings with Huawei storage systems and OpenStack.

Table 2-1 Version mappings among the Manila Driver, Huawei storage system and OpenStack

OpenStack	Huawei storage system
Rocky/Stein/	OceanStor Dorado V6 6.1.0/6.1.2RC1
Train/Ussuri/ Victoria	OceanStor Dorado V3 V300R002
VICTORIA	OceanStor F V5/V5 V500R007/V500R007 Kunpeng
	OceanStor F V3/V3 V300R003/V300R006

Table 2-2 Mappings among Manila Driver, features and the OpenStack version (√: supported, x: not supported)

Feature	Normal Share	Hypermetro Share	Remarks
OpenStack	Rocky/Stein/ Train/Ussuri/ Victoria	Rocky/Stein/ Train/Ussuri/ Victoria	NA
Create Share	√	√	NA
Delete Share	√	√	NA
Allow access	√	√	NA
Deny access	√	√	NA
Create Snapshot	√	✓	Hypermetro Share supports snapshot creation only on the local end of HyperMetro.

Feature	Normal Share	Hypermetro Share	Remarks
Delete Snapshot	√	✓	Hypermetro Share supports snapshot deletion only on the local end of HyperMetro.
Revert Snapshot	√	√	Hypermetro Share supports snapshot revertion only on the local end of HyperMetro.
Extend Share	√	√	NA
Shrink Share	√	√	NA
Create Share from Snapshot	√	✓	Hypermetro Share supports snapshot-based share creation only on the local end of HyperMetro.
Manage/ Unmanage Share	√	x	The file system and sharing relationship managed by manila must be one-to-one
Manage/ Unmanage Snapshot	√	x	NA
Replication	√	х	NA
QoS	√	√	Hypermetro Share supports SmartQoS configuration only on the local end of HyperMetro.
Multi-tenancy	√	x	NA
SmartThin/ SmartThick	√	√	NA
SmartCompression	√	√	NA
SmartDedupe	√	√	NA

Feature	Normal Share	Hypermetro Share	Remarks
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 supports SmartPartition configuration only on the local end of HyperMetro.
SmartCache	✓	√	Hypermetro Share supports SmartCache configuration only on the local end of HyperMetro.

CAUTION

The **Manage/Unmanage Share** function has not been delivered in OceanStor Dorado V6 6.1.0. If you need to use it, please contact Huawei engineers in time.

3 Specifications and Restrictions

Feature	Sub-feature	Description	Remarks
Mapping platform	Native OpenStack platform	OpenStack versions: Rocky/Stein/Train/ Ussuri/Victoria	
Configur ation	XML configuration file	Ensure that the storage pool used for configuring Cinder Driver exists on Huawei storage. Otherwise, create a storage pool. In addition, the type of this storage pool must be file storage service.	
		All parameter values in the Cinder Driver configuration file cannot contain the following special characters: <>&'''	
		Set Product based on the actual backend storage model.	
Policy	SmartQoS	Protection policies and restriction policies are mutually exclusive. If they are configured together, shares will fail to be created.	 Limit policies: maxIOPS and maxBandWidth Protection policies: minIOPS, minBandWidth, and latency

IOType is mandatory. If it is not configured in a QoS policy, shares will fail to be created.	
OceanStor Dorado V6 6.1.0 supports only restriction policies and does not support protection policies.	

4 Deployment

- 4.1 Obtaining Manila Driver
- 4.2 Deploying Manila Driver for the Manila Share Service

4.1 Obtaining Manila Driver

Two ways to obtain OpenStack Driver:

One is through the OpenStack community warehouse. From Kilo, Huawei has contributed Huawei Storage Driver to OpenStack, users can download OpenStack Driver from OpenStack community for free. After installing the specified OpenStack, OpenStack Driver will be placed under the catalog of "../manila/manila/share/drivers/huawei". If you don't find the corresponding installation files, you can download the OpenStack Driver from OpenStack community warehouse at https://github.com/openstack/manila.

Another is through Huawei OpenStack Driver warehouse. By visiting https://github.com/Huawei/OpenStack_Driver, you can download OpenStack Driver that corresponds to OpenStack community version.

Steps to get Manila Driver are as follows:

□ NOTE

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

Step 1 Enter the above warehouse address in the browser, for example Huawei warehouse address:

https://github.com/huaweistorage/OpenStack_Driver.

- **Step 2** Click the "Download ZIP" bottom to download Driver, and unzip it.
- **Step 3** Find "Manila" catalog in which the Diver is extracted, there are multiple OpenStack Driver for different OpenStack version, choose the corresponding Driver.

4.2 Deploying Manila Driver for the Manila Share Service

The OpenStack standard deployment steps are as follows:

Step 1 Before installation, delete all the installation files of Huawei OpenStack Driver. The default installation path is /usr/lib/python2.7/dist-packages/manila/share/drivers/huawei.

⚠ CAUTION

The complete path may vary depending on the operating system. You can search for the Huawei Driver code directory as follows:

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

In the preceding information, the Huawei Driver code directory is /usr/lib/python2.7/site-packages/manila/share/drivers/huawei/.

Step 2 Copy OpenStack Manila Driver to the Manila Driver installation directory. (Just overwrite)

```
# Is -l
-rw-r--r-- 1 root root 4615 Oct 21 14:55 base.py
-rw-r--r-- 1 root root 5348 Oct 21 14:55 constants.py
-rw-r--r-- 1 root root 12721 Oct 21 14:55 huawei_nas.py
-rw-r--r-- 1 root root 4302 Oct 21 14:55 huawei_utils.py
-rw-r--r-- 1 root root 0 Oct 21 14:55 __init__.py
drwxr-xr-x 2 root root 4096 Oct 30 10:24 oceanstor9000
-rw-r--r-- 1 root root 193777 Oct 21 14:55 test_huawei_nas.py
drwxr-xr-x 2 root root 4096 Oct 21 12:28 v3
```

If the distributed OceanStor 9000 NAS driver is not included:

5 Configuring Basic Properties

This chapter describes how to configure Huawei Manila Driver.

NOTICE

- In OpenStack Ocata Manila, when you create a share without a share type, the default share type in the /etc/manila/manila.conf file will be used.
- Ensure that the storage pool used for configuring Cinder Driver exists in Huawei storage. Otherwise, create a storage pool. In addition, the type of this storage pool must be file storage service.
- All of the parameter values cannot include XML special character < > & ' "
- Snapshot and Replication can't be configured at the same time
- 5.1 Configuring Manila Driver
- 5.2 Verifying the Manila Share Service

5.1 Configuring Manila Driver

Procedure

- **Step 1** In /etc/manila, create a Huawei Manila Driver configuration file in .xml format. You can customize the configuration file name, for example, manila_huawei_conf.xml.
- **Step 2** Edit the **manila_huawei_conf.xml** file and set **mandatory** parameters in the Huawei Driver configuration file.

</Filesystem> </config>

Table 5-1 Parameters in the configuration file

Parameter	Default Value	Description	Remarks
Product	V3	Storage product model. The value can be V3 , V5 , or Dorado .	
LogicalPortIP	-	Logical port IP address, conditionally mandatory. This parameter is mandatory when DNS is not configured. If DNS is configured at the same time, the priority of LogicalPortIP is lower than that of DNS. Support multiple IPs, separate them by semicolons (;).	This parameter is mandatory when DNS is not configured. If they are both configured, the priority of the parameter is lower than that of DNS.
DNS	-	DNS domain name, conditionally mandatory. This parameter is mandatory when Logical port IP is not configured. If Logical port IP is configured at the same time, the priority of DNS is higher than that of Logical port IP. Support multiple DNSes, separate them by semicolons (;).	This parameter is mandatory when LogicalPortIP is not configured. If they are both configured, the priority of the parameter is higher than that of LogicalPortIP.
RestURL	-	Access address of the REST interface.	
UserName	-	User name of an administrator.	
UserPassword	-	Password of an administrator.	
StoragePool	-	Name of a storage pool to be used.	

Step 3 Edit the **manila_huawei_conf.xml** file and configure **optional** parameters in the Huawei Driver configuration file.

<?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>
  <NFSClient>
   <IP>x.x.x.x</IP>
  </NFSClient>
  <CIFSClient>
   <UserName>xxx</UserName>
   <userPassword>xxx</userPassword>
  </CIFSClient>
 </Filesystem>
</Config>
```

Table 5-2 Parameters in the configuration file

Parameter	Defau lt Value	Description	Туре
Port	-	Port name list of bond port or ETH port, used to create vlan and logical port.	If <port> is not configured, then will choose an online port on the array.</port>
SectorSize	64	The size of the disk blocks, optional value can be "4", "8", "16", "32" or "64", and the units is KB.	Optional.
SnapshotSuppor t	True	Support snapshot or not.	This parameter cannot be enabled together with ReplicationSuppor t.
ReplicationSupp ort	False	Support replication or not.	This parameter cannot be enabled together with SnapshotSupport.
WaitInterval	3	Interval for querying file system status. The unit is second.	Optional.
Timeout	60	Timeout interval for waiting command execution of a storage device to complete. The unit is second.	Optional.
SnapshotReserv e	20	The ratio of the snapshot size to the file system size(%). The range is (0~50).	Optional.

Parameter	Defau lt Value	Description	Туре
NFSClient\IP	-	Backend IP in admin network to use for mounting NFS share.	Mandatory when create a share from snapshot.
CIFSClient \UserName	-	Backend user name in admin network to use for mounting CIFS share.	Mandatory when create a share from snapshot. If OceanStor Dorado V6 is used, the user name must be added to a local authentication user group with administrator rights.
CIFSClient \UserPassword	-	Backend password in admin network to use for mounting CIFS share.	Mandatory when create a share from snapshot.

Step 4 Configure the /etc/manila/manila.conf file. Add the following backend configuration to the end of the file:

[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

- Backend name huawei_backend can be customized.
- **share_driver** indicates the type of the driver to be used. Set this parameter to **manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver**.
- manila_huawei_conf_file indicates the path of the Manila Driver configuration file.
- driver_handles_share_servers indicates whether to enable multi-tenant.

In the **DEFAULT** area, add **huawei_backend** to **enabled_share_backends** item.

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

Step 5 Change the permission on the /etc/manila/manila_huawei_conf.xml file to be the same as that on /etc/manila/manila.conf.

```
# 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
...
```

Step 6 Restart the Manila Share service.

systemctl restart openstack-manila-share.service

----End

5.2 Verifying the Manila Share Service

Procedure

Step 1 Create a common share type. **general** indicates the name of the share type, which is specified by users. **False** indicates that multi-tenancy is not supported. To configure multi-tenancy, set this parameter to **True**.

manila type-create general False

Step 2 Set the backend of the share type. huawei_backend indicates the backend name of the share, which is the value of share_backend_name in step 4 in 5.1 Configuring Manila Driver.

manila type-key general set share_backend_name=huawei_backend

Step 3 Set snapshot properties.

manila type-key general set snapshot_support=True

Step 4 Set snapshot rollback properties.

manila type-key general set revert_to_snapshot_support=True

Step 5 Sets the properties for creating a share using a snapshot.

manila type-key general set create_share_from_snapshot_support=True

Step 6 Run the **manila create --name test001 NFS 2 --share-type general** command to create a share that supports the preceding properties.

----End

□ NOTE

Snapshot properties: This feature is mutually exclusive with remote replication and they cannot be configured at the same time (excluding OceanStor Dorado V6 NAS).

6 Configuring Advanced Properties

This chapter describes how to configure advanced storage properties.

Huawei storage supports advanced properties, such as Smartx. By associating with specified share types, these properties can be used in OpenStack.

- 6.1 Configuring Thin/Thick
- 6.2 Configuring SmartDedupe
- 6.3 Configuring SmartCompression
- 6.4 Configuring SmartCache
- 6.5 Configuring SmartPartition
- 6.6 Configuring SmartQoS
- 6.7 Creating a Share on a Specified Storage Pool
- 6.8 Creating a Share with a Specified Disk Type
- 6.9 Configuring SectorSize
- 6.10 Configuring Replication
- 6.11 Configuring the Owning Controller of an FS
- 6.12 Configuring the NFS Client Permission
- 6.13 Configuring HyperMetro
- 6.14 Configuring filesystem:mode

6.1 Configuring Thin/Thick

Step 1 Configure Thin property:Run the following command to configure the key-value pair whose Thin property is **true**:

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

Configure Thick property:Run the following command to configure the key-value pair whose Thin property is **false**:

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

Step 2 Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

□ NOTE

- If capabilities:thin_provisioning is set in the share type extra-specs, it will be used (thin share will be created if capabilities:thin_provisioning=<is> true, thick share will be created if capabilities: thin provisioning=<is> false).
- If capabilities:thin_provisioning is not set in the share type extra-specs, thin share will be created.

6.2 Configuring SmartDedupe

Step 1 Run the following command to configure the key-value pair whose SmartDedupe property is **true**:

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

Step 2 Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

◯ NOTE

Only Thin share supports SmartDedupe.

Refer to section 4.1 to configure thin property.

6.3 Configuring SmartCompression

Step 1 Run the following command to configure the key-value pair whose SmartCompression is **true**:

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

Step 2 Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

◯ NOTE

Only **Thin** share supports SmartComprssion.

Refer to section 4.1 to configure thin property.

6.4 Configuring SmartCache

Step 1 Run the following command to configure the key-value pair whose SmartCache property is **true**:

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

Configure a name for the SmartCache existing on the storage device.

root@ubuntu:~#manila type-key test_share_type set huawei_smartcache:cachename='test_name'

Step 2 Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

6.5 Configuring SmartPartition

Step 1 Run the following command to configure the key-value pair whose SmartPartition property is **true**:

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

Configure a name for the SmartPartition existing on the storage device.

root@ubuntu:~#manila type-key test_share_type set huawei_smartpartition:partitionname='test_name'

Step 2 Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

6.6 Configuring SmartQoS

About This Chapter

Huawei supports the following QoS properties. One or multiple properties in the same class can be configured in one QoS property simultaneously.

Protection policies: latency, minIOPS, minBandWidth

Restriction policies: maxIOPS, maxBandWidth

<u>A</u> CAUTION

- Protection policies and restriction policies are mutually exclusive. If they are configured together, shares will fail to be created.
- OceanStor Dorado V6 6.1.0 supports only restriction policies and does not support protection policies.
- **IOType** is mandatory. If it is not configured in a QoS policy, shares will fail to be created.

6.6.1 Configuring the Maximum Control IOPS

This section describes how to configure the maximum Control IOPS.

Procedure

Step 1 Run the following command to configure the key-value pair whose QoS property is **true**.

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

Run the following command to configure QoS control property parameters.

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

- maxIOPS: indicates the maximum IOPS. The value is an integer larger than 0.
- **IOType** (mandatory): indicates the read and write type.0 indicates the control read I/Os, 1 indicates the control write I/Os, 2 indicates the control read and write I/Os.
- **Step 2** Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

6.6.2 Configuring the Minimum Control IOPS

This section describes how to configure the minimum Control IOPS.

Procedure

Step 1 Run the following command to configure the key-value pair whose QoS property is **true**.

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

Run the following command to configure QoS control property parameters.

root@ubuntu:~# manila type-key test_share_type set qos:IOType=0 qos:minIOPS=50

- minIOPS: indicates the minimum IOPS. The value is an integer larger than **0**.
- **IOType** (mandatory): indicates the read and write type.**0** indicates the control read I/Os, **1** indicates the control write I/Os, **2** indicates the control read and write I/Os.
- **Step 2** Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

6.6.3 Configuring the Maximum Control Bandwidth

This section describes how to configure the maximum control bandwidth.

Procedure

Step 1 Run the following command to configure the key-value pair whose QoS property is **true**

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

Run the following command to configure QoS control property parameters.

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

- maxBandWidth: indicates the maximum BANDWIDTH. The value is an integer larger than **0** and expressed in MB/s.
- **IOType** (mandatory): indicates the read and write type.**0** indicates the control read I/Os, **1** indicates the control write I/Os, **2** indicates the control read and write I/Os.

Step 2 Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

6.6.4 Configuring the Minimum Control Bandwidth

This section describes how to configure the minimum control bandwidth.

Procedure

Step 1 Run the following command to configure the key-value pair whose QoS property is **true**.

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

Run the following command to configure QoS control property parameters.

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

- **minBandWidth**: indicates the minimum BANDWIDTH. The value is an integer larger than **0** and expressed in MB/s.
- **IOType** (mandatory): indicates the read and write type.**0** indicates the control read I/Os, **1** indicates the control write I/Os, **2** indicates the control read and write I/Os.
- **Step 2** Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

6.6.5 Configuring the Control Latency

This section describes how to configure the minimum control latency.

Procedure

Step 1 Run the following command to configure the key-value pair whose QoS property is **true**.

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

Run the following command to configure QoS control property parameters.

root@ubuntu:~# manila type-key test_share_type set qos:IOType=0 qos:latency=50

- latency: indicates the maxinum LATENCY. The value is an integer larger than
 o and expressed in ms.
- **IOType** (mandatory): indicates the read and write type.**0** indicates the control read I/Os, **1** indicates the control write I/Os, **2** indicates the control read and write I/Os.
- **Step 2** Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

6.6.6 Configuring Multiple Control Policies

This section describes how to configure multiple control policies.

Procedure

Step 1 Run the following command to configure the key-value pair whose QoS property is **true**

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

Run the following command to configure QoS control property parameters.

root@ubuntu:~# manila type-key test_share_type set qos:IOType=0 qos:latency=50

- latency: indicates the maxinum LATENCY. The value is an integer larger than
 0
- minIOPS: indicates the minimum IOPS. The value is an integer larger than 0.
- **minBandWidth**: indicates the minimum BANDWIDTH. The value is an integer larger than **0**.
- **IOType** (mandatory): indicates the read and write type.0 indicates the control read I/Os, 1 indicates the control write I/Os, 2 indicates the control read and write I/Os.
- **Step 2** Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

6.7 Creating a Share on a Specified Storage Pool

This section describes how to create a share on a certain storage pool.

Procedure

- **Step 1** Run the **manila type-create target_pool_type False** command to create a share type. "target_pool_type"indicates the name of a share type. "False"indicates that the multi-tenant mode is not supported, when configured to "True" means that the multi-tenant mode is supported.
- **Step 2** Run the following command to configure the assign storagepool.
 - Configure a single assign pool root@ubuntu:~# manila type-key target_pool_type set pool_name=StoragePool001
 - Configure multiple assign pools.
 root@ubuntu:~# manila type-key target_pool_type set pool_name="<or>
 StoragePool002"
- **Step 3** Run manila create --name test001 NFS 2 --share-type target_pool_type to create a share that supports the preceding properties.

----End

6.8 Creating a Share with a Specified Disk Type

This section describes how to create a share with a certain disk type.

Procedure

- **Step 1** Run the **manila type-create disk_type False** command to create a share type. "disk_type"indicates the name of a share type. "False"indicates that the multitenant mode is not supported, when configured to "True" means that the multitenant mode is supported.
- **Step 2** Run the following command to configure the assign disk type.
 - Configure one type of disk type.
 root@ubuntu:~# manila type-key disk_type set huawei_disk_type=sas
 - Configure multiple types of disk type.
 root@ubuntu:~# manila type-key disk_type set huawei_disk_type="<or> sas <or> ssd"

optional values of disk type can be "ssd", "sas", "nl_sas" or "mix", and the "mix" is a mixture of two or more in "ssd", "sas", "nl sas".

Step 3 Run manila create --name test001 NFS 2 --share-type disk_type to create a share that supports the preceding properties.

----End

6.9 Configuring SectorSize

This section describes how to configure the size of blocks of filesystem.

Procedure

- **Step 1** Run the **manila type-create sectorsize_type False** command to create a share type. "sectorsize_type"indicates the name of a share type. "False"indicates that the multi-tenant mode is not supported, when configured to "True" means that the multi-tenant mode is supported.
- **Step 2** Run the following command to configure the key-value pair whose SectorSize property is **true**:

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

Configure the value of sectorsize:

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

□ NOTE

`SectorSize` is the size of the disk blocks, optional value can be "4", "8", "16", "32" or "64", and the units is KB.

Step 3 Run manila create --name test001 NFS 2 --share-type sectorsize_type to create a share that supports the above attributes.

□ NOTE

- If "sectorsize" is configured in both share_type and xml file, the value of sectorsize in the share_type will be used.
- If there is no value available in share-type and SectorSize is configured in the xml file at the same time. Take the SectorSize value in the xml file.
- If "sectorsize" is configured in neither share_type nor xml file, huawei storage backends will provide a default value(64) when creating a new share.
- Rocky version and later versions do not support SectorSize

6.10 Configuring Replication

This section describes how to configure replication.

Prerequisites

Step 1 In file /etc/manila/manila.conf, configure two back ends with the remote replication relationship.

```
[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
```


- The replication_domain option is a backend specific StrOpt option to be used within manila.conf. The value can be any ASCII string. Two backends that can replicate between each other would have the same replication_domain.
- The replica_backend option is set when the remote replication function is required. The
 value is a user-defined character string. The backends of the remote replication
 relationship are replica_backend.
- manila_huawei_conf_1.xml and manila_huawei_conf_2.xml are user-defined configuration files of the driver. For details, see 5 Configuring Basic Properties.
- The 'local_replication' option should set to 'True' when using replication within array, and configure using the same array in 'manila_huawei_conf_1.xml' and 'manila_huawei_conf_2.xml'. If 'local_replication' option is not configured or its value is False, it means replication between arrays.

Step 2 Restart the Manila services.

Procedure

- **Step 1** Run the manila type-create replication_type False command to create a share type. "replication_type" indicates the name of a share type. "False" indicates that the multi-tenant mode is not supported, when configured to "True" means that the multi-tenant mode is supported.
- **Step 2** Run the following command to configure the type of replication. root@ubuntu:~# manila type-key replication_type set replication_type=dr
- **Step 3** Run manila create --name test001 NFS 2 --share-type replication_type to create a share that supports the preceding properties.
- **Step 4** Run manila share-replica-create test001 to create replication for share test001.

Now Huawei Manila Driver only supports **dr** style replication. For more details about **dr**, please refer to **http://docs.openstack.org/developer/manila/devref/share_replication.html**.

----End

6.11 Configuring the Owning Controller of an FS

This section describedsf s how to configure the owning controller of a FS.

Procedure

- Step 1 In this command, controller_type indicates the type of the shared controller, which is specified by the user. Falseindicates that multi-tenant is not supported. If you want to enable multi-tenant, change this value to True.

 root@ubuntu:~# manila type-create controller type A False
- Step 2 Run the following command to configure the key-value pair whose Huawei_controllerattribute is true.

 root@ubuntu:~# manila type-key controller type A set capabilities:huawei_controller='<is> true'
- Step 3 Configure controllername that exists on the storage device and associate controller_type_A to the share type.

 root@ubuntu:~# manila type-key controller_type_A set huawei_controller:controllername='CTE0.A'
- **Step 4** Run the manila create --name test001 NFS 2 --share-type controller_type_A command to create a share with the owning controller attribute specified.

----End

6.12 Configuring the NFS Client Permission

This section describes how to configure the NFS client permission.

Procedure

Step 1 In this command, **share_privilege_type** indicates the name of the share type, which is specified by the user. **False** indicates that multi-tenant is not supported. If you want to enable multi-tenant, change this value to **True**.

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

- **Step 2** Run the following command to configure the key-value pair whose **huawei_share_privilege** attribute is **true**.
 - root@ubuntu:~# manila type-key share_privilege_type set capabilities:huawei_share_privilege='<is> true'
- **Step 3** Configure the attribute name that exists on the storage device and associate to the share type.

Table 6-1

Name	Description	Value
sync	Write mode	0: Synchronous 1: Asynchronous
allsquash	Permission restriction	0: all_squash 1: no_all_squash
rootsquash	Root permission restriction	0: root_squash 1: no_root_squash
secure	Source Port Verification	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

Step 4 Run the manila create --name test001 NFS 2 --share-type share_privilege_type command to create a share with the owning controller attribute specified.

----End

----End

6.13 Configuring HyperMetro

This section describes how to configure HyperMetro.

Prerequisites

You have configured HyperMetro domains, vStores, vStore pairs, and logical ports available to vStores on the local and remote storage arrays.

Procedure

Step 1 In file **/etc/manila/manila.conf**, configure two backends with the NAS HyperMetro relationship.

[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
```

■ NOTE

- metro_info is set when the NAS HyperMetro function is required. Its value is a user-defined list composed of multiple elements. Each element contains the metro_domain, local_vStore_name, remote_vStore_name, remote_backend, and metro_logic_ip fields, which are separated by commas (,). For details, see Table 6-2.
- manila_huawei_conf_1.xml and manila_huawei_conf_2.xml are user-defined configuration files of the driver. For details, see 5 Configuring Basic Properties.
- Currently, shares cannot be created from the snapshots of HyperMetro shares.

Table 6-2 Parameter description

Parameter	Description
metro_domain	HyperMetro domain name. You must use the same value for backends with the HyperMetro relationship.
local_vStore_name	Local tenant name of HyperMetro.
remote_vStore_name	Remote tenant name of HyperMetro.
remote_backend	Name of the remote backend which belongs to the Manila backends composing a HyperMetro pair. Two backends with the NAS HyperMetro relationship are remote backends for each other.
metro_logic_ip	Logical IP address of the vStore configured for NAS HyperMetro. Two backends with the NAS HyperMetro relationship must be configured with the same IP address. Support multiple IPs, separate them by semicolons (;).

Step 2 Run the following command to restart the Manila service:

systemctl restart openstack-manila-volume.service

- **Step 3** Run the **manila type-create HyperMetro False** command to create the share type. **HyperMetro** indicates the name of the share type, which is specified by the user. **False** indicates that multi-tenant is not supported. If you want to enable multi-tenant, change this value to **True**.
- **Step 4** Run the following command to configure the key-value pair for the HyperMetro type:

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

Step 5 Run the manila create --name metro001 NFS 2 --share-type HyperMetro command to create a share.

----End

6.14 Configuring filesystem:mode

This section describes how to configure the distribution algorithm for a shared source file system. Only OceanStor Dorado v6 supports the configuration.

Procedure

- **Step 1** Run the manila type-create filesystem_mode_type False command to create a share type. filesystem_mode_type indicates the name of the share type, which is specified by users. False indicates that multi-tenant is not supported. To configure multi-tenant, change it to **True**.
- **Step 2** Run the following command to set the value of **filesystem:mode**:

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

■ NOTE

filesystem:mode=0 indicates the performance mode, and **filesystem:mode=2** indicates the directory balance mode.

Step 3 Run the manila create --name test001 NFS 2 --share-type filesystem_mode_type command to create a share that supports the filesystem:mode attribute.

7 Best Practices

7.1 Quick Interconnection with Huawei Storage

7.1 Quick Interconnection with Huawei Storage

Configuration Process

This section demonstrates how to configure Huawei Manila Driver on OpenStack to interconnect with Huawei Storage.

- **Step 1** Obtain Manila Driver(See chapter 3.1 for details).
- **Step 2** Create or query the file storage pool which will be used in Huawei storage.
- **Step 3** Configure file **manila.conf** and Huawei-defined configuration file of Driver. (For details, see chapter 4).
 - 1. In /etc/manila, create a Huawei-defined Driver configuration file in .xml format. In this example, manila_huawei_conf.xml is used as the file name that can be changed based on actual conditions.
 - 2. Set parameters for the created file.

3. Configure file manila.conf.

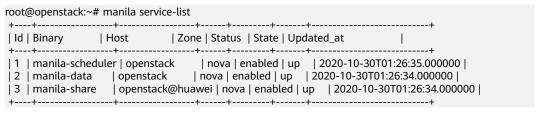
At the end of file /etc/manila/manila.conf, add the following configuration item. In this configuration item, volume_driver indicates the loaded Driver file, and manila_huawei_conf_file indicates the Huawei-defined configuration file.

[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

In the **[DEFAULT]** area, modify the configuration as follows to enable the huawei back-end:

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

- Step 4 Restart the Manila service.
- **Step 5** Check the service status.



In this example, the service status is **up**, indicating that the service is started correctly.