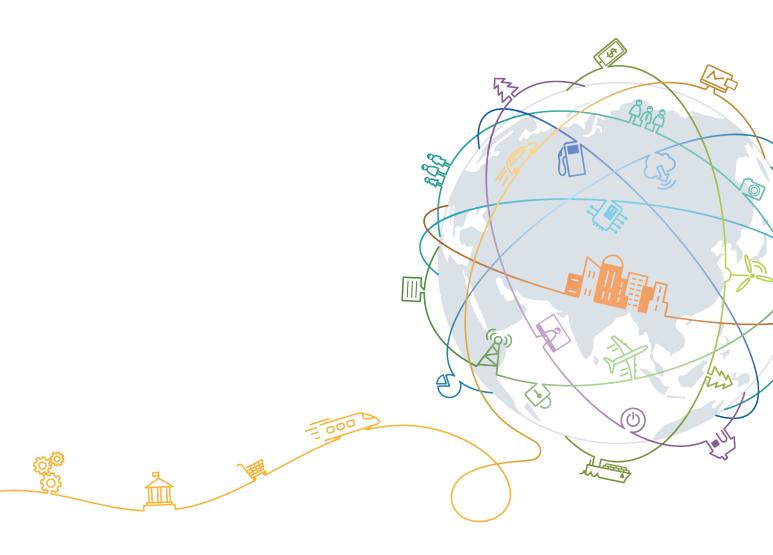
OpenStack Manila Driver

Configuration Guide

Issue 01

Date 2019-09-26





Copyright © Huawei Technologies Co., Ltd. 2019. 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: http://www.huawei.com
Email: support@huawei.com

Contents

1 Overview	1
2 Version Mapping	2
3 Deployment	4
3.1 Obtaining Manila Driver	4
3.2 Ubuntu Environment Deployment	5
3.3 Red Hat Environment Deployment	
4 Configuring Basic Properties	7
5 Configuring Advanced Properties	11
5.1 Configuring Thin/Thick	11
5.2 Configuring SmartDedupe	12
5.3 Configuring SmartCompression	12
5.4 Configuring SmartCache	12
5.5 Configuring SmartPartition	
5.6 Configuring SmartQoS	13
5.6.1 Configuring the Maximum Control IOPS	
5.6.2 Configuring the Minimum Control IOPS	14
5.6.3 Configuring the Maximum Control Bandwidth	14
5.6.4 Configuring the Minimum Control Bandwidth	14
5.6.5 Configuring the Control Latency	15
5.6.6 Configuring Multiple Control Policies	15
5.7 Creating a Share on a Specified Storage Pool	16
5.8 Creating a Share with a Specified Disk Type	16
5.9 Configuring SectorSize	17
5.10 Configuring Replication	
5.11 Configuring the Owning Controller of an FS	18
5.12 Configuring the NFS Client Permission	19
5.13 Configuring HyperMetro	20
6 Best Practices	22
6.1 Quick Interconnection with Huawei Storage	22

$oldsymbol{1}$ 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
Newton	OceanStor V3 V300R003/V300R006
Ocata	OceanStor V5 V500R007
Pike	OceanStor 2600 V3 V300R005
Queens	OceanStor 18500/18800 V300R003/V300R006/V500R007
Rocky	OceanStor Dorado V3 V300R002C10
Stein	

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

Feature	Newto n	Ocata	Pike	Queens	Rocky	Stein
Create Share	√	√	√	√	√	√
Delete Share	√	√	√	√	√	√
Allow access	√	√	√	√	√	√
Deny access	√	√	√	√	√	√
Create Snapshot	√	√	√	√	√	√
Delete Snapshot	√	√	√	√	√	√
Revert Snapshot	X	X	X	√	√	√

Feature	Newto n	Ocata	Pike	Queens	Rocky	Stein
Manage/Unmanage Share	√	√	√	✓	√	~
Extend Share	√	√	√	√	√	√
Shrink Share	√	√	√	√	√	√
SmartCompression	√	√	√	√	√	✓
SmartDedupe	√	√	√	√	√	✓
SmartCache	√	√	√	√	√	✓
SmartThin/Thick	√	√	√	√	√	~
SmartPartition	√	√	√	√	√	~
SmartQoS	√	√	√	√	√	~
Multi-tenancy	√	√	√	√	√	~
Ensure Share	√	√	√	√	√	~
Create Share from Snapshot	√	√	√	✓	√	~
Manage/Unmanage Snapshot	X	√	√	√	√	√
Create a share on a certain storage pool	X	√	√	√	√	√
Create a share with a certain disk type	х	√	√	√	√	√
SectorSize	Х	√	√	√	√	√
Replication	X	√	√	√	√	√

3 Deployment

- 3.1 Obtaining Manila Driver
- 3.2 Ubuntu Environment Deployment
- 3.3 Red Hat Environment Deployment

3.1 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:

MOTE

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.

3.2 Ubuntu Environment Deployment

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

On the host, the version of Python is 2.7. If another version is used, use the correct version number.

- **Step 2** Copy OpenStack Manila Driver to the Manila Driver installation directory.
- **Step 3** Make configuration by referring to **4 Configuring Basic Properties** and **5 Configuring Advanced Properties**.
- **Step 4** After configuration, restart the Manila-Share service by running the following command: service manila-share restart
- **Step 5** Check the status of service restart by running the **manila service-list** command. If **State** is **up**, the Manila-Share service has been restarted.

```
root@u1404:~# manila service-list
| Id | Binary
               | Host
                           | Zone | Status | State |
Updated at
                 +---+
| 1 | manila-scheduler | u1404
                           | nova | enabled | up |
2016-03-15T01:43:48.000000 |
| 2 | manila-data | u1404
                           | nova | enabled | up
2016-03-15T01:43:50.000000 |
| 3 | manila-share | u1404@v3r3 | nova | enabled | up
2016-03-15T01:43:41.000000 |
+---+-----
+----+
```

----End

3.3 Red Hat Environment Deployment

The OpenStack standard deployment steps are as follows:

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

⚠ CAUTION

On the host, the version of Python is 2.7. If another version is used, use the correct version number.

- **Step 2** Copy OpenStack Manila Driver to the Manila Driver installation directory.
- **Step 3** Make configuration by referring to **4 Configuring Basic Properties** and **5 Configuring Advanced Properties**.
- **Step 4** After configuration, restart the Manila-Share service by running the following command: systemctl restart openstack-manila-volume.service
- Step 5 Check the status of service restart by running the manila service-list command. If State is up, the Manila-Share service has been restarted.

```
root@u1404:~# manila service-list
| Id | Binary | Host
                          | Zone | Status | State |
Updated_at
                - 1
| 1 | manila-scheduler | u1404
2016-03-15T01:43:48.000000 |
                          | nova | enabled | up |
2016-03-15T01:43:48.000000 |
| 2 | manila-data | u1404
                          | nova | enabled | up
2016-03-15T01:43:50.000000 |
| 3 | manila-share | u1404@v3r3 | nova | enabled | up
2016-03-15T01:43:41.000000 |
+---+
+----+
```

4 Configuring Basic Properties

This chapter describes how to configure Huawei Manila Driver.

NOTICE

- In OpenStack Ocata Manila, when create a share without a share type, the default share type in "/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.

If you want to use the snapshot feature of shared files, set **snapshot_support** to **True** in **share type**.

root@ubuntu-001:~# manila type-key default share type set snapshot support=True

If you want to use snapshots to create shared files, set create share from snapshot support to True in share type.

root@ubuntu-001:~# manila type-key default_share_type set create_share_from_snapshot_support=True

Procedure

Step 1 Modify the manila.conf configuration file of Manila. Add share_driver, share_backend_name and manila_huawei_conf_file,and modify driver handles share servers.

manila_huawei_conf.xml is used as an example.

• Example for configuring a storage system:

```
[DEFAULT]
enabled_share_backends = huawei_manila_backend
[huawei_manila_backend]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf.xml
share_backend_name = huawei_manila_backend
driver_handles_share_servers = False
```

• Example for configuring multiple storage systems:

```
[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
[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
```

NOTE

- The parameter value of **driver_handles_share_servers** can be **True** or **False**. **True** indicates that the multi-tenant mode is supported and **False** indicates that the multi-tenant mode is not supported.
- Step 2 In /etc/manila, create a driver configuration file named manila_huawei_conf.xml. The driver configuration file name must be the same as value of the manila_huawei_conf_file item in the manila.conf configuration file.

Details about the driver configuration file are as follows:

```
<?xml version='1.0' encoding='UTF-8'?>
  <Config>
   <Storage>
     <Product>V3</Product>
      <LogicalPortIP>x.x.x.x</LogicalPortIP>
     <Port>abc;CTE0.A.H1</Port>
      <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
     <UserName>xxxxxxxxxx</UserName>
     <UserPassword>xxxxxxxxxx</UserPassword>
     <SnapshotSupport>True</SnapshotSupport>
      <ReplicationSupport>False</ReplicationSupport>
    </Storage>
    <Filesystem>
      <StoragePool>xxxxxxxxxx;xxxxxxxx</StoragePool>
      <SectorSize>64</SectorSize>
      <WaitInterval>3</WaitInterval>
      <Timeout>60</Timeout>
      <SnapshotReserve>20</SnapshotReserve>
```

Table 4-1 lists all parameters in the configuration file.

Table 4-1 Parameters in the configuration file

Parameter	Defaul t Value	Description	Туре
Product	V3	Storage product model.	Mandatory.
LogicalPortIP	-	Logical port IP address. Support multiple IPs, separate them by semicolons (;).	Mandatory when not in multi-tenancy mode.
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>
RestURL	-	Access address of the REST interface.	Mandatory.
UserName	-	User name of an administrator.	Mandatory.
UserPassword	-	Password of an administrator.	Mandatory.
StoragePool	-	Name of a storage pool to be used.	Mandatory.
SectorSize	64	The size of the disk blocks, optional value can be "4", "8", "16", "32" or "64", and the units is KB.	Optional.
SnapshotSupport	True	Support snapshot or not.	Optional.
ReplicationSuppo rt	False	Support replication or not.	Optional.
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.
SnapshotReserve	20	The ratio of the snapshot size to the file system size(%). The range is $(0\sim50)$.	Optional.

Parameter	Defaul t 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.
CIFSClient \UserPassword	-	Backend password in admin network to use for mounting CIFS share.	Mandatory when create a share from snapshot.

NOTE

- You can configure multiple RestURL, storage pools and ports in the configuration file and separate them using semicolons(;)
- For details about share configuration information, see the **show share** command in the specific command-line interface (CLI) document
- Snapshot and Replication can't be configured at the same time
- All of the parameter values cannot include XML special character <> & ' "

Step 3 Restart the Manila service.

5 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.

- 5.1 Configuring Thin/Thick
- 5.2 Configuring SmartDedupe
- 5.3 Configuring SmartCompression
- 5.4 Configuring SmartCache
- 5.5 Configuring SmartPartition
- 5.6 Configuring SmartQoS
- 5.7 Creating a Share on a Specified Storage Pool
- 5.8 Creating a Share with a Specified Disk Type
- 5.9 Configuring SectorSize
- 5.10 Configuring Replication
- 5.11 Configuring the Owning Controller of an FS
- 5.12 Configuring the NFS Client Permission
- 5.13 Configuring HyperMetro

5.1 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.

5.2 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.

5.3 Configuring SmartCompression

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

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.

5.4 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

5.5 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

5.6 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

⚠ CAUTION

Protection policies and Restriction policies are mutually exclusive. If they are configured together, shares will fail to be created.

"IOType" is mandatory. If it isn't configured in qos, shares will fail to be created.

5.6.1 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

5.6.2 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

5.6.3 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

5.6.4 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

5.6.5 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 maximum LATENCY. The value is an integer larger than **0** 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

5.6.6 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 maximum 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

5.7 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>
 StoragePool001 <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

5.8 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 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 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

5.9 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. If "sectorsize" is configured in both share_type and xml file, the value of sectorsize in the share_type will be used. 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.

----End

5.10 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

[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
```

NOTE

- 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.
- manila_huawei_conf_1.xml and manila_huawei_conf_2.xml are user-defined configuration files
 of the driver. For details, see 4 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.

----End

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.

MOTE

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

5.11 Configuring the Owning Controller of an FS

This section describeds f 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. **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 controller type A False
```

Step 2 Run the following command to configure the key-value pair whose **Huawei controller**attribute 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='CTEO.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

5.12 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.

Tr-	1. 1 .	. = 1
ıа	nie	. D-I

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

5.13 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: remote backend,
   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: remote backend,
   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 5-2.
- manila_huawei_conf_1.xml and manila_huawei_conf_2.xml are user-defined configuration files
 of the driver. For details, see 4 Configuring Basic Properties.
- Currently, shares cannot be created from the snapshots of HyperMetro shares.

Table 5-2 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.

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.

6 Best Practices

6.1 Quick Interconnection with Huawei Storage

6.1 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).
 - 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. Check the owner and owning group of the file.

Ensure that the owner and owning group of file /etc/manila/manila_huawei_conf.xml is the same as those of file /etc/manila/manila.conf.

```
-rw-r--r-- 1 manila manila 2662 Jul 29 02:13 manila.conf
-rw-r--r-- 1 manila manila 778 Jul 30 02:56 manila_huawei_conf.xml
```

4. 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 backend:

```
[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.