# **OpenStack Manila Driver Configuration Guide**

# OpenStack Manila Driver Configuration Guide

Issue 01

**Date** 2023-10-08





#### Copyright © Huawei Technologies Co., Ltd. 2023. 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: <a href="https://www.huawei.com">https://www.huawei.com</a>

Email: <a href="mailto:support@huawei.com">support@huawei.com</a>

# **About This Document**

# **Intended Audience**

This document is intended for:

- Technical support engineers
- O&M engineers
- Engineers with basic knowledge of storage and OpenStack

# **Symbol Conventions**

The symbols that may be found in this document are defined as follows.

Symbol	Description		
▲ DANGER	Indicates a hazard with a high level of risk which, if not avoided, will result in death or serious injury.		
<b>⚠ WARNING</b>	Indicates a hazard with a medium level of risk which, if not avoided, could result in death or serious injury.		
<b>⚠</b> CAUTION	Indicates a hazard with a low level of risk which, if not avoided, could result in minor or moderate injury.		
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results.  NOTICE is used to address practices not related to personal injury.		
<b>◯</b> NOTE	Supplements the important information in the main text.  NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.		

# **Change History**

Issue	Date	Description
01	2023-09-28	This issue is the first official release.

# **Contents**

About This Document	ii
1 Overview	1
2 Version Mappings	2
3 Specifications	6
4 Installing Manila Driver	8
4.1 Obtaining Manila Driver	8
4.2 Deploying Huawei Manila Driver for the Manila Share Service	8
5 Configuring Basic Properties of Manila Driver	10
5.1 Configuring Manila Driver	10
5.2 Verifying the Manila Share Service	15
6 (Optional) Configuring Advanced Properties of Manila Driver	16
6.1 Configuring the Thin/Thick Property	17
6.2 Configuring the SmartDedupe Property	17
6.3 Configuring the SmartCompression Property	18
6.4 Configuring the SmartCache Property	18
6.5 Configuring the SmartPartition Property	19
6.6 Configuring the SmartQoS Property	19
6.6.1 Configuring the Maximum IOPS	20
6.6.2 Configuring the Minimum IOPS	21
6.6.3 Configuring the Maximum Bandwidth	21
6.6.4 Configuring the Minimum Bandwidth	22
6.6.5 Configuring the Maximum Latency	23
6.6.6 Configuring Multiple Parameters	23
6.7 Creating a Share in a Specified Storage Pool	24
6.8 Creating a Share on a Disk of a Specified Type	24
6.9 Configuring the SectorSize Property	25
6.10 Configuring the Remote Replication Property	26
6.11 Configuring the Owning Controller of a File System	
6.12 Configuring the NFS Client Permission	
6.13 Configuring the HyperMetro Property	
6.14 Configuring filesystem:mode	30

OpenStack Ma	ınila Driver	· Configuration	Guide
OpenStack Ma	nila Driver	· Configuration	Guide

6.15 Configuring the Percentage of the Reserved Snapshot Space	31
6.16 Configuring the UNIX Permission on the Root Directory of a File System	31
6.17 Configuring a Snapshot Directory to Be Visible	32
6.18 Configuring Certificate Authentication	32
7 Best Practices	33
7.1 Quick Interconnection with Huawei Storage	33

# 1 Overview

Huawei Manila Driver is a plug-in deployed on the OpenStack Manila module. It connects to Huawei storage to provide share storage capabilities for the OpenStack platform.

# **2** Version Mappings

This chapter describes the version mappings among Huawei Manila Driver, Huawei storage, and OpenStack.

Table 2-1 Storage product versions supported by Manila Driver

OpenStack Version	Storage Product Version	
Mitaka/Newton/ Ocata/Pike/Queens/ Rocky/Stein/Train/ Ussuri/Victoria/ Wallaby/Xena/ Yoga/Zed	<ul> <li>OceanStor V3: V300R006C60/V300R006C61</li> <li>OceanStor V5: V500R007C61/V500R007C70</li> <li>OceanStor V6: 6.1.3/6.1.5/6.1.6</li> <li>OceanStor Dorado V6: 6.1.0/6.1.2/6.1.3/6.1.5/6.1.6</li> </ul>	

**Table 2-2** Support for Manila Driver features (√: supported; x: not supported)

Feature	Normal Share	HyperMetro Share	Remarks
OpenStack	Mitaka/ Newton/ Ocata/Pike/ Queens/ Rocky/Stein/ Train/Ussuri/ Victoria/ Wallaby/ Xena/ Yoga/Zed	Mitaka/ Newton/ Ocata/Pike/ Queens/ Rocky/Stein/ Train/Ussuri/ Victoria/ Wallaby/ Xena/ Yoga/Zed	N/A
Create Share	√	√	N/A
Delete Share	√	√	N/A
Allow access	√	√	N/A
Deny access	√	√	N/A

Feature	Normal Share	HyperMetro Share	Remarks
Create Snapshot	√	√	Snapshots can be created only for the local HyperMetro share.
Delete Snapshot	√	√	Snapshots can be deleted only for the local HyperMetro share.
Revert Snapshot	√	√	Snapshots can be used for rollback only for the local HyperMetro share.
Extend Share	√	√	N/A
Shrink Share	√	√	N/A
Create Share from Snapshot	√	√	A share can be created using a snapshot only for the local HyperMetro share.
Manage/Unmanage Share	√	х	The file systems and shares managed by Manila Driver must be in one-to-one mapping. Only file systems and shares of the system vStore can be managed.
Manage/Unmanage Snapshot	√	х	N/A
Replication	√	х	N/A
QoS	√	√	SmartQoS can be configured only for the local HyperMetro share.
Multi-tenancy	√	х	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
Create a share with a certain disk type	√	√	N/A

Feature	Normal Share	HyperMetro Share	Remarks
SectorSize	√	√	N/A
SmartPartition	√	<b>√</b>	SmartPartition can be configured only for the local HyperMetro share.
SmartCache	√	<b>√</b>	SmartCache can be configured only for the local HyperMetro share.

#### □ NOTE

- The architecture of OceanStor Dorado NAS has changed. The features of the local HyperMetro share are subject to the actual storage.
- For OceanStor Dorado 6.1.2 and later versions, you can create or delete a HyperMetro share, create, delete, or roll back a snapshot, create a share using a snapshot, and expand or reduce the capacity of a share after a primary/secondary switchover.

**Table 2-3** Mappings between Manila Driver features and storage devices

Feature	OceanStor V3	OceanStor V5	OceanStor V6	OceanStor Dorado V6
Thin share	Supported	Supported	Supported	Supported
Thick share	Supported	Supported	Not supported	Not supported
SmartDedupe	Supported (only for thin share)	Supported (only for thin share)	Supported (only for thin share)	Supported (only for thin share)
SmartCompre ssion	Supported (only for thin share)	Supported (only for thin share)	Supported (only for thin share)	Supported (only for thin share)
SmartCache	Supported	Supported	Not supported	Supported
SmartPartitio n	Supported	Supported	Not supported	Not supported

Feature	OceanStor	OceanStor	OceanStor	OceanStor
	V3	V5	V6	Dorado V6
SmartQoS  Protection policy parameters : latency, minIOPS, and minBand Width  Restriction policy parameters : maxIOPS and maxBand Width	Supported. Protection policy parameters and restriction policy parameters cannot be configured at the same time.	Supported. Protection policy parameters and restriction policy parameters cannot be configured at the same time.	Supported. Protection policy parameters and restriction policy parameters cannot be configured at the same time.	Supported. Protection policy parameters and restriction policy parameters cannot be configured at the same time.

# 3 Specifications

Feature	Sub-feature	Description	Remarks
Mapping platform	Native OpenStack platform	OpenStack versions: Mitaka/Newton/ Ocata/Pike/Queens/ Rocky/Stein/Train/ Ussuri/Victoria/ Wallaby/Xena/ Yoga/Zed	

Feature	Sub-feature	Description	Remarks
Configuration	XML configuration file	<ul> <li>Ensure that the storage pool used for configuring Manila Driver exists on Huawei storage.         Otherwise, you need to manually create one. In addition, the type of this storage pool must be the file storage service.</li> <li>All parameter values in the Huawei Manila Driver configuration file cannot contain the following XML special characters: &lt;&gt;&amp;""</li> <li>Set Product based on the actual back-end storage model.</li> </ul>	
Policy	SmartQoS	<ul> <li>Protection policy parameters and restriction policy parameters are mutually exclusive. If they are configured at the same time, the share will fail to be created.</li> <li>In the QoS property, IOType is mandatory. If IOType is not configured in a QoS policy, the share will fail to be created.</li> </ul>	<ul> <li>Restriction policy parameters:         maxIOPS and maxBandWidth</li> <li>Protection policy parameters:         minIOPS,         minBandWidth,         and latency</li> </ul>

# 4 Installing Manila Driver

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

# 4.1 Obtaining Manila Driver

#### **Procedure**

- **Step 1** Open a browser and enter <a href="https://github.com/Huawei/OpenStack\_Driver">https://github.com/Huawei/OpenStack\_Driver</a> in the address box.
- **Step 2** Choose **Code** > **Download ZIP** to download the Huawei Manila Driver package to a local PC.
- **Step 3** Decompress the package.
- **Step 4** In the directory generated upon decompression, find the **Manila** directory, which contains Huawei Manila Driver of multiple OpenStack versions.

#### □ NOTE

Huawei OpenStack Manila Driver warehouse maintains 14 stable versions (M-Z), ensuring long-term stable running of historical versions.

----End

# 4.2 Deploying Huawei Manila Driver for the Manila Share Service

The standard deployment procedure of the OpenStack community is as follows:

- **Step 1** Search for the original Huawei Manila Driver code directory in the system.
  - The relative path is .../manila/share/drivers/huawei.
  - The absolute path may vary in different systems. You can search for the code directory of Huawei Manila Driver as follows:

- Run the following command. In the command output, /usr/lib/ python2.7/site-packages/manila/share/drivers/huawei is the code directory.
  - # python -c "from manila.share.drivers import huawei; print (huawei.\_\_path\_\_)" ['/usr/lib/python2.7/site-packages/manila/share/drivers/huawei']
- Run the following command. In the command output, /usr/lib/ python3.7/site-packages/manila/share/drivers/huawei is the code directory.
  - # python3 -c "from manila.share.drivers import huawei; print (huawei.\_\_path\_\_)" ['/usr/lib/python3.7/site-packages/manila/share/drivers/huawei']
- **Step 2** Copy the obtained OpenStack Manila Driver to the driver installation directory of the Manila node. For details about the default path, see **Step 1**. The following uses version X as an example.

#### **NOTICE**

You only need to copy and overwrite the driver. Do not delete the original one.

```
# Is -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
```

# 5 Configuring Basic Properties of Manila Driver

This chapter describes how to configure Huawei Manila Driver.

#### NOTICE

- In OpenStack Ocata Manila, when you create a share without specifying 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 Manila Driver exists on Huawei storage. Otherwise, you need to manually create one. In addition, the type of this storage pool must be the file storage service.
- All parameter values in the Huawei Manila Driver configuration file cannot contain the following XML special characters: <>&'''
- The snapshot and replication functions cannot be enabled 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 Manila Driver configuration file.

- <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
- <UserName>xxx</UserName>
- <userPassword>xxx</userPassword>
- </Storage>
- <Filesystem>
- <StoragePool>xxx</StoragePool>
- </Filesystem>
- </config>

**Table 5-1** Mandatory parameters

Parameter	Description	Default Value	Remarks
Product	Storage product type. The value can be <b>V3</b> , <b>V5</b> , <b>V6</b> , or <b>Dorado</b> .	V3	-
LogicalPortIP	IP address of a logical port. You can configure multiple IP addresses separated by semicolons (;).	-	This parameter is mandatory when the <b>DNS</b> parameter is not specified. If both parameters are configured, the priority of this parameter is lower than that of the <b>DNS</b> parameter.
DNS	DNS domain name. You can configure multiple DNS domain names separated by semicolons (;).	-	This parameter is mandatory when the LogicalPortIP parameter is not specified. If both parameters are configured, the priority of this parameter is higher than that of the LogicalPortIP parameter.
RestURL	Access address of the REST interface.	-	-
UserName	User name of a storage administrator.	-	-
UserPassword	Password of the storage administrator.	-	-
StoragePool	Name of a storage pool to be used. You can configure multiple storage pools separated by semicolons (;).	-	-

**Step 3** Edit the **manila\_huawei\_conf.xml** file and set optional parameters in the Huawei Manila 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>
   <UnixPermission>755</UnixPermission>
   <ShowSnapshotDir>true</ShowSnapshotDir>
   <NFSClient>
    <IP>x.x.x.x</IP>
   </NFSClient>
   <CIFSClient>
    <UserName>xxx</UserName>
    <UserPassword>xxx</UserPassword>
   </CIFSClient>
  </Filesystem>
 </Config>
```

Table 5-2 Optional parameters

Parameter	Description	Default Value	Туре
Port	Available service port (bond port or physical port). You can configure multiple ports separated by semicolons (;).	-	This parameter is optional when driver_handles_share_se rvers is set to True. If this parameter is not set, all online ports are used.
SectorSize	Size of the file system disk block. The value can be 4, 8, 16, 32, or 64 KB.	64	Optional
SnapshotSupp ort	Whether to enable the snapshot function.	True	Except for OceanStor Dorado, SnapshotSupport and ReplicationSupport cannot be enabled at the same time.
ReplicationSu pport	Whether to enable the remote replication function.	False	Except for OceanStor Dorado, SnapshotSupport and ReplicationSupport cannot be enabled at the same time.
WaitInterval	Interval for querying file system status, expressed in seconds (s).	3	Optional

Parameter	Description	Default Value	Туре
Timeout	Timeout period of waiting for a storage device to execute commands, expressed in seconds (s).	60	Optional
SnapshotRese rve	Percentage (%) of the reserved snapshot space. The value ranges from 0 to 50.	0	Optional
UnixPermissio	Permission on the root directory of a file system.	755	Optional. The value consists of three digits whose meanings are as follows: The first digit refers to the permissions of the owner. The second digit refers to the permissions of the user group to which the file belongs. The last digit refers to the permissions of everyone else. The digits are from 0 to 7 whose meanings are as follows: 0: No permission; 1: Execute permission; 2: Write permission; 3: Write and execute permissions; 4: Read permissions; 5: Read and execute permissions; 7: All permissions (read, write, and execute).
ShowSnapsho tDir	Whether the snapshot directory is visible.	true	Optional. The value can be:  true: visible  false: invisible
NFSClient\IP	IP address used by a management node to mount an NFS share when the share is created using a snapshot.	-	This parameter is mandatory when a share is created using a snapshot.

Parameter	Description	Default Value	Туре
CIFSClient \UserName	User name used by a management node to mount a CIFS share when the share is created using a snapshot.	-	This parameter is mandatory when a share is created using a snapshot. If OceanStor Dorado is used, the user name must be added to a local authentication user group that has administrator rights.
CIFSClient \UserPasswor d	User password used by a management node to mount a CIFS share when the share is created using a snapshot.	-	This parameter is mandatory when a share is created using a snapshot.

#### **Step 4** Add the following information to the end of the /etc/manila/manila.conf 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

#### □ NOTE

- huawei backend indicates the backend name. It can be customized.
- share\_driver indicates the type of Manila Driver to be used. In this example, the parameter is set to manila.share.drivers.huawei.huawei\_nas.HuaweiNasDriver.
- manila\_huawei\_conf\_file indicates the path of the Huawei Manila Driver configuration file.
- If driver\_handles\_share\_servers is set to True, you need to specify share-network
  when creating a share.

# **Step 5** In the **[DEFAULT]** section, modify the **enabled\_share\_backends** configuration item and add the **huawei backend** backend.

[DEFAULT]

enabled\_share\_backends=xxx,huawei\_backend

**Step 6** Change the permission on the /etc/manila/manila\_huawei\_conf.xml file to be the same as that on the /etc/manila/manila.conf file.

# 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 7** Run the following command to restart the Manila service.

# systemctl restart openstack-manila-share.service

# 5.2 Verifying the Manila Share Service

#### **Precautions**

Except for OceanStor Dorado V6 NAS, the snapshot and remote replication properties are mutually exclusive and cannot be configured at the same time.

#### Procedure

**Step 1** Create a common share type.

In the following command, **general** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

# manila type-create general False

**Step 2** Set the backend of the share type.

In the following command, **huawei\_backend** indicates the backend name of the share configured in **Step 4**.

# manila type-key general set share\_backend\_name=huawei\_backend

**Step 3** Set the snapshot property.

# manila type-key general set snapshot\_support=True

**Step 4** Set the snapshot rollback property.

# manila type-key general set revert\_to\_snapshot\_support=True

**Step 5** Set the property of creating a share using a snapshot.

# manila type-key general set create\_share\_from\_snapshot\_support=True

**Step 6** Create a share that supports the preceding properties.

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

# 6 (Optional) Configuring Advanced Properties of Manila Driver

This chapter describes how to configure advanced storage properties.

Huawei storage supports multiple advanced features of the Smart series. These features can be used in OpenStack environments by associating with specific share types.

- 6.1 Configuring the Thin/Thick Property
- 6.2 Configuring the SmartDedupe Property
- 6.3 Configuring the SmartCompression Property
- 6.4 Configuring the SmartCache Property
- 6.5 Configuring the SmartPartition Property
- 6.6 Configuring the SmartQoS Property
- 6.7 Creating a Share in a Specified Storage Pool
- 6.8 Creating a Share on a Disk of a Specified Type
- 6.9 Configuring the SectorSize Property
- 6.10 Configuring the Remote Replication Property
- 6.11 Configuring the Owning Controller of a File System
- 6.12 Configuring the NFS Client Permission
- 6.13 Configuring the HyperMetro Property
- 6.14 Configuring filesystem:mode
- 6.15 Configuring the Percentage of the Reserved Snapshot Space
- 6.16 Configuring the UNIX Permission on the Root Directory of a File System
- 6.17 Configuring a Snapshot Directory to Be Visible
- 6.18 Configuring Certificate Authentication

# 6.1 Configuring the Thin/Thick Property

This section describes how to configure the Thin/Thick property.

#### **Procedure**

**Step 1** Run the **manila type-create thin\_type False** command to create a share type.

In the preceding command, **thin\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Configure the Thin/Thick property.

#### **□** NOTE

- If **thin\_provisioning** is set to **true**, the share type is **Thin**. If **thin\_provisioning** is set to **false**, the share type is **Thick**.
- If thin\_provisioning is not configured in share-type, the default share type Thin is used.
- OceanStor Dorado 6.x and OceanStor 6.x do not support Thick shares.
- Configuring the Thin property

Run the following command to configure a key-value pair whose **thin\_provisioning** property is **true**.

root@ubuntu:~# manila type-key thin\_type set capabilities:thin\_provisioning='<is> true'

Configuring the Thick property

Run the following command to configure a key-value pair whose **thin\_provisioning** property is **false**.

root@ubuntu:~# manila type-key thin\_type set capabilities:thin\_provisioning='<is> false'

**Step 3** Run the **manila create --name test001 NFS 2 --share-type thin\_type** command to create a share that supports the preceding properties.

----End

# 6.2 Configuring the SmartDedupe Property

This section describes how to configure the SmartDedupe property.

#### **Precautions**

SmartDedupe can be configured only when the share type is Thin.

#### **Procedure**

**Step 1** Run the **manila type-create dedupe\_type False** command to create a share type.

In the preceding command, **dedupe\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure a key-value pair whose **thin\_provisioning** property is **true**.

root@ubuntu:~# manila type-key dedupe\_type set capabilities:thin\_provisioning='<is> true'

**Step 3** Run the following command to configure a key-value pair whose SmartDedupe property is **true**.

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

**Step 4** Run the manila create --name test001 NFS 2 --share-type dedupe\_type command to create a share that supports the preceding properties.

----End

# 6.3 Configuring the SmartCompression Property

This section describes how to configure the SmartCompression property.

#### **Precautions**

SmartCompression can be configured only when the share type is Thin.

#### **Procedure**

**Step 1** Run the **manila type-create compression\_type False** command to create a share type.

In the preceding command, **compression\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure a key-value pair whose **thin\_provisioning** property is **true**.

root@ubuntu:~# manila type-key compression\_type set capabilities:thin\_provisioning='<is> true'

**Step 3** Run the following command to configure a key-value pair whose SmartCompression property is **true**.

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

**Step 4** Run the manila create --name test001 NFS 2 --share-type compression\_type command to create a share that supports the preceding properties.

----End

## 6.4 Configuring the SmartCache Property

This section describes how to configure the SmartCache property.

#### Procedure

**Step 1** Run the **manila type-create cache\_type False** command to create a share type.

In the preceding command, **cache\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure a key-value pair whose SmartCache property is **true**.

root@ubuntu:~# manila type-key cache type set capabilities:huawei smartcache='<is> true'

- **Step 3** Configure the SmartCache name that already exists on the storage device and associate the cache with the share type.
  - root@ubuntu:~# manila type-key cache\_type set huawei\_smartcache:cachename='test\_name'
- **Step 4** Run the manila create --name test001 NFS 2 --share-type cache\_type command to create a share that supports the preceding properties.

----End

# 6.5 Configuring the SmartPartition Property

This section describes how to configure the SmartPartition property.

#### **Procedure**

- **Step 1** Run the **manila type-create partition\_type False** command to create a share type.
  - In the preceding command, **partition\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.
- **Step 2** Run the following command to configure a key-value pair whose SmartPartition property is **true**.
  - root@ubuntu:~# manila type-key partition\_type set capabilities:huawei\_smartpartition='<is> true'
- **Step 3** Configure the SmartPartition name that already exists on the storage device and associate the partition with the share type.

  root@ubuntu:~# manila type-key partition type set huawei smartpartition:partitionname='test name'
- **Step 4** Run the manila create --name test001 NFS 2 --share-type partition\_type command to create a share that supports the preceding properties.

----End

## 6.6 Configuring the SmartQoS Property

Protection policy parameters and restriction policy parameters are mutually exclusive. If they are configured at the same time, the share will fail to be created.

In the QoS property, **IOType** is mandatory. If **IOType** is not configured in a QoS policy, the share will fail to be created. For OceanStor Dorado storage, **IOType** must be set to **2**.

Huawei supports the following QoS properties. One or more properties of the same type can be configured in one QoS policy.

- Protection policy parameters: latency, minIOPS, and minBandWidth
- Restriction policy parameters: maxIOPS and maxBandWidth

**Table 6-1** IOType parameter

Parameter	Description	Remarks
ЮТуре	Read/write type.	The value can be:
		• <b>0</b> : read I/O
		• 1: write I/O
		• <b>2</b> : read and write I/O

Table 6-2 QoS parameters

Parameter	Description	Remarks
maxIOPS	Maximum IOPS. This is a restriction policy parameter.	The value is an integer greater than 0. For OceanStor Dorado series, the value is an integer greater than or equal to 100.
minIOPS	Minimum IOPS. This is a protection policy parameter.	The value is an integer greater than 0. For OceanStor Dorado series, the value is an integer greater than or equal to 100.
maxBandWidth	Maximum bandwidth. This is a restriction policy parameter.	The value is an integer greater than 0, expressed in MB/s.
minBandWidth	Minimum bandwidth. This is a protection policy parameter.	The value is an integer greater than 0, expressed in MB/s.
latency	Maximum latency. This is a protection policy parameter.	The value is an integer greater than 0, expressed in ms. For OceanStor Dorado series, the value is 500 or 1500.

## 6.6.1 Configuring the Maximum IOPS

This section describes how to configure the maximum IOPS.

#### **Procedure**

**Step 1** Run the **manila type-create maxiops\_type False** command to create a share type.

In the preceding command, **maxiops\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure a key-value pair whose SmartQoS property is **true**.

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

**Step 3** Run the following command to configure QoS parameters.

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

- maxIOPS: maximum IOPS. The value is an integer greater than 0. For OceanStor Dorado series, the value is an integer greater than or equal to 100.
- **IOType** (mandatory): read/write type.
  - 0 indicates the read I/O.
  - 1 indicates the write I/O.
  - 2 indicates the read and write I/O.
- **Step 4** Run the manila create --name test001 NFS 2 --share-type maxiops\_type command to create a share that supports the preceding properties.

----End

## 6.6.2 Configuring the Minimum IOPS

This section describes how to configure the minimum IOPS.

#### **Procedure**

**Step 1** Run the **manila type-create miniops\_type False** command to create a share type.

In the preceding command, **miniops\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure a key-value pair whose SmartQoS property is **true**.

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

**Step 3** Run the following command to configure QoS parameters.

root@ubuntu:~# manila type-key miniops\_type set qos:IOType=0 qos:minIOPS=500

- minIOPS: minimum IOPS. The value is an integer greater than 0. For OceanStor Dorado series, the value is an integer greater than or equal to 100.
- **IOType** (mandatory): read/write type.
  - 0 indicates the read I/O.
  - 1 indicates the write I/O.
  - 2 indicates the read and write I/O.
- **Step 4** Run the manila create --name test001 NFS 2 --share-type miniops\_type command to create a share that supports the preceding properties.

----End

### 6.6.3 Configuring the Maximum Bandwidth

This section describes how to configure the maximum bandwidth.

#### **Procedure**

**Step 1** Run the **manila type-create maxbandwidth\_type False** command to create a share type.

In the preceding command, **maxbandwidth\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure a key-value pair whose SmartQoS property is **true**.

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

**Step 3** Run the following command to configure QoS parameters.

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

- maxBandWidth: maximum bandwidth. The value is an integer greater than 0, expressed in MB/s.
- **IOType** (mandatory): read/write type.
  - 0 indicates the read I/O.
  - 1 indicates the write I/O.
  - 2 indicates the read and write I/O.
- **Step 4** Run the manila create --name test001 NFS 2 --share-type maxbandwidth\_type command to create a share that supports the preceding properties.

----End

## 6.6.4 Configuring the Minimum Bandwidth

This section describes how to configure the minimum bandwidth.

#### **Procedure**

**Step 1** Run the **manila type-create minbandwidth\_type False** command to create a share type.

In the preceding command, **minbandwidth\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure a key-value pair whose SmartQoS property is **true**.

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

**Step 3** Run the following command to configure QoS parameters.

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

- **minBandWidth**: minimum bandwidth. The value is an integer greater than 0, expressed in MB/s.
- **IOType** (mandatory): read/write type.
  - 0 indicates the read I/O.
  - 1 indicates the write I/O.
  - 2 indicates the read and write I/O.

**Step 4** Run the manila create --name test001 NFS 2 --share-type minbandwidth\_type command to create a share that supports the preceding properties.

----End

### 6.6.5 Configuring the Maximum Latency

This section describes how to configure the maximum latency.

#### **Procedure**

**Step 1** Run the **manila type-create latency\_type False** command to create a share type.

In the preceding command, **latency\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure a key-value pair whose SmartQoS property is **true**.

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

**Step 3** Run the following command to configure QoS parameters.

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

- **latency**: maximum latency. The value is an integer greater than 0, expressed in ms. For OceanStor Dorado series, the value is 500 or 1500.
- **IOType** (mandatory): read/write type.
  - 0 indicates the read I/O.
  - 1 indicates the write I/O.
  - 2 indicates the read and write I/O.
- **Step 4** Run the manila create --name test001 NFS 2 --share-type latency\_type command to create a share that supports the preceding properties.

----End

## **6.6.6 Configuring Multiple Parameters**

This section describes how to configure multiple parameters.

#### **Procedure**

**Step 1** Run the **manila type-create multiple\_strategy\_type False** command to create a share type.

In the preceding command, **multiple\_strategy\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure a key-value pair whose SmartQoS property is **true**.

root@ubuntu:~# manila type-key multiple\_strategy\_type set capabilities:qos='<is> true'

**Step 3** Run the following command to configure QoS parameters.

root@ubuntu:~# manila type-key multiple\_strategy\_type set qos:IOType=0 qos:latency=500 qos:minIOPS=500 qos:minBandWidth=50

- **latency**: maximum latency. The value is an integer greater than 0, expressed in ms. For OceanStor Dorado series, the value is 500 or 1500.
- **minIOPS**: minimum IOPS. The value is an integer greater than 0. For OceanStor Dorado series, the value is an integer greater than or equal to 100.
- minBandWidth: minimum bandwidth. The value is an integer greater than 0.
- **IOType** (mandatory): read/write type.
  - 0 indicates the read I/O.
  - 1 indicates the write I/O.
  - 2 indicates the read and write I/O.
- **Step 4** Run the manila create --name test001 NFS 2 --share-type multiple\_strategy\_type command to create a share that supports the preceding properties.

----End

# 6.7 Creating a Share in a Specified Storage Pool

This section describes how to create a share in a specified storage pool.

#### **Procedure**

**Step 1** Run the **manila type-create target\_pool\_type False** command to create a share type.

In the preceding command, **target\_pool\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

- **Step 2** Run the following command to configure the target storage pool.
  - To configure a single target storage pool, run the following command: root@ubuntu:~# manila type-key target\_pool\_type set pool\_name=StoragePool001
- **Step 3** Run the manila create --name test001 NFS 2 --share-type target\_pool\_type command to create a share that supports the preceding properties.

----End

# 6.8 Creating a Share on a Disk of a Specified Type

This section describes how to create a share on a disk of a specified type.

#### **Procedure**

**Step 1** Run the manila type-create disk\_type False command to create a share type.

In the preceding command, **disk\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure the disk type.

The value of **disk\_type** can be **ssd**, **sas**, **nl\_sas**, or **mix**, where **mix** indicates that two or more of **ssd**, **sas**, and **nl sas** are used.

- To configure one disk type, run the following command: root@ubuntu:~# manila type-key disk\_type set huawei\_disk\_type=sas
- To configure multiple disk types, run the following command: root@ubuntu:~# manila type-key disk\_type set huawei\_disk\_type="<or> sas <or> ssd"
- Step 3 Run the manila create --name test001 NFS 2 --share-type disk\_type command to create a share that supports the preceding properties.

----End

# **6.9 Configuring the SectorSize Property**

This section describes how to configure the SectorSize property (disk block size) of a share source file system.

#### **Precautions**

- If **SectorSize** is configured in both **share-type** and the XML file, the value of **SectorSize** in **share-type** is preferentially used.
- If **share-type** does not have an available value and **SectorSize** is configured in the XML file, the value of **SectorSize** in the XML file is used.
- If **SectorSize** is not configured in **share-type** and the XML file, the default value 64 KB provided by the storage array is used.
- In Rocky and later versions, **SectorSize** cannot be configured.

#### **Procedure**

**Step 1** Run the **manila type-create sectorsize\_type False** command to create a share type.

In the preceding command, **sectorsize\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure a key-value pair whose **SectorSize** property is **true**.

root@ubuntu:~# manila type-key sectorsize\_type set capabilities:huawei\_sectorsize='<is> true'

**Step 3** Configure the value of **SectorSize**.

root@ubuntu:~# manila type-key sectorsize\_type set huawei\_sectorsize:sectorsize=4

□ NOTE

Huawei storage systems provide five **SectorSize** values (4/8/16/32/64 KB) to adapt to different application scenarios.

**Step 4** Run the manila create --name test001 NFS 2 --share-type sectorsize\_type command to create a share that supports the preceding properties.

----End

# 6.10 Configuring the Remote Replication Property

This section describes how to configure the remote replication property.

#### **Prerequisites**

In the /etc/manila/manila.conf file, two backends in the remote replication relationship have been configured and the Manila service has been restarted.

- **replication\_domain** is set when the remote replication function is required. The parameter value is a user-defined character string. The same parameter value must be set for backends in the remote replication relationship.
- **replica\_backend** is set when the remote replication function is required. The parameter value is a user-defined character string. Backends in the remote replication relationship are **replica\_backend** of each other. The name format is *Host name@Backend name*.
- manila\_huawei\_conf\_1.xml and manila\_huawei\_conf\_2.xml are custom configuration files of Manila Driver. For details, see 5 Configuring Basic Properties of Manila Driver.
- To use the remote replication function within a storage array, add the local\_replication parameter and set it to True, and set the same storage array parameters in manila\_huawei\_conf\_1.xml and manila\_huawei\_conf\_2.xml. If the local\_replication parameter is not specified or is set to False, the remote replication function between storage arrays is used.

```
[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
```

#### Procedure

**Step 1** Run the **manila type-create replication\_type False** command to create a share type.

In the preceding command, **replication\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

**Step 2** Run the following command to configure the remote replication type.

root@ubuntu:~# manila type-key replication\_type set replication\_type=dr

#### □ NOTE

Currently, Manila Driver supports only remote replication of the **dr** type. For details about **dr**, **click here**.

- **Step 3** Run the manila create --name test001 NFS 2 --share-type replication\_type command to create a share that supports the preceding properties.
- **Step 4** Run the **manila share-replica-create test001** command to create a share replica to establish a remote replication relationship between it and the share created in **Step 3**.

----End

# 6.11 Configuring the Owning Controller of a File System

This section describes how to configure the owning controller of a file system.

#### Procedure

**Step 1** Run the **manila type-create controller\_type False** command to create a share type.

In the preceding command, **controller\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**. root@ubuntu:~# manila type-create controller\_type\_A False

- **Step 2** Configure the controller name that already exists on the storage device and associate the **controller\_type\_A** with the share type.
  - root@ubuntu:~# manila type-key controller\_type\_A set huawei\_controller:controllername='CTE0.A'
- Step 3 Run the manila create --name test001 NFS 2 --share-type controller\_type\_A command to create a share that supports the preceding properties.

----End

## 6.12 Configuring the NFS Client Permission

This section describes how to configure the NFS client permission.

#### Procedure

**Step 1** Run the **manila type-create share\_privilege\_type False** command to create a share type.

In the preceding command, **share\_privilege\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**. root@ubuntu:~# manila type-create share\_privilege\_type False

**Step 2** Run the following command to configure the property type name that already exists on the storage device and associate **share\_privilege\_type** with the share type.

**Table 6-3** Parameters

Parameter	Description	Value
sync	Write mode.	<ul><li>0: synchronous</li><li>1: asynchronous</li></ul>
allsquash	Permission constraint.	<ul><li>0: all_squash</li><li>1: no_all_squash</li></ul>
rootsquash	root permission constraint.	<ul><li>0: root_squash</li><li>1: no_root_squash</li></ul>
secure	Source port verification constraint.	<ul><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

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

----End

## 6.13 Configuring the HyperMetro Property

This section describes how to configure the HyperMetro property.

#### **Prerequisites**

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

#### Procedure

- **Step 1** In the /etc/manila/manila.conf file, configure two backends in the NAS HyperMetro relationship.
  - metro\_info is set when the NAS HyperMetro function is required. The
    parameter value is a custom list containing 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 6-4**.

- manila\_huawei\_conf\_1.xml and manila\_huawei\_conf\_2.xml are custom configuration files of Manila Driver. For details, see 5 Configuring Basic Properties of Manila Driver.
- A share cannot be created using a snapshot of a HyperMetro share.
- For OceanStor Dorado 6.1.2 and later versions, you can create or delete a HyperMetro share, create, delete, or roll back a snapshot, create a share using a snapshot, and expand or reduce the capacity of a share after a primary/secondary switchover.
- For OceanStor Dorado 6.1.2 and later versions, if HyperMetro is used and the backends are forcibly specified in volume\_type, set the backend names in type to "<or> backend1 <or> backend2" (backend1 and backend2 are HyperMetro backend names). The backend storage pool names must be the same. Otherwise, a share cannot be created using a snapshot after a HyperMetro primary/secondary switchover.

```
[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
```

#### Table 6-4 Parameters

Parameter	Description
metro_domain	HyperMetro domain name. The same parameter value must be set for backends in the HyperMetro relationship.
local_vStore_name	Local vStore name of a HyperMetro vStore pair.
remote_vStore_name	Remote vStore name of a HyperMetro vStore pair.

Parameter	Description
remote_backend	Names of the remote Manila backends in the NAS HyperMetro relationship. The two backends in the NAS HyperMetro relationship are <b>remote_backend</b> of each other. The name format is <i>Host name@Backend name</i> .
metro_logic_ip	Logical IP address under a vStore. The same IP address is configured for the two backends in the NAS HyperMetro relationship. You can enter multiple IP addresses separated by semicolons (;).

- **Step 2** Run the following command to restart the Manila service. root@ubuntu:~# systemctl restart openstack-manila-share.service
- **Step 3** Run the manila type-create HyperMetro False command to create a share type.

In the preceding command, **HyperMetro** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. This parameter cannot be set to **True** for HyperMetro.

**Step 4** Run the following command to configure a key-value pair whose HyperMetro property is **true**.

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 that supports the preceding properties.

----End

# 6.14 Configuring filesystem:mode

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

#### **Procedure**

**Step 1** Run the **manila type-create filesystem\_mode\_type False** command to create a share type.

In the preceding command, **filesystem\_mode\_type** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**.

- **Step 2** Run the following command to configure the value of **filesystem:mode**.
  - **filesystem:mode=0** indicates the performance mode.
  - filesystem:mode=2 indicates the directory balance mode.

root@ubuntu:~# manila type-key filesystem\_mode\_type set filesystem:mode=0

Step 3 Run the manila create --name test001 NFS 2 --share-type filesystem\_mode\_type command to create a share that supports the preceding properties.

# 6.15 Configuring the Percentage of the Reserved Snapshot Space

This section describes how to configure the percentage of the reserved snapshot space.

#### **Procedure**

**Step 1** Run the **manila type-create snapshot\_reserve False** command to create a share type.

In the preceding command, **snapshot\_reserve** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**. root@ubuntu:~# manila type-create snapshot\_reserve False

**Step 2** Run the following command to configure the value of **snapshot\_reserve\_percentage**.

root@ubuntu:~# manila type-key snapshot\_reserve set huawei\_snapshotreserveper:snapshot\_reserve\_percentage=20

**Step 3** Run the manila create --name test001 NFS 2 --share-type snapshot\_reserve command to create a share that supports the preceding properties.

----End

# 6.16 Configuring the UNIX Permission on the Root Directory of a File System

This section describes how to configure the UNIX permission on the root directory of a file system.

#### **Procedure**

**Step 1** Run the **manila type-create unix\_permission False** command to create a share type.

In the preceding command, **unix\_permission** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**. root@ubuntu:~# manila type-create unix\_permission False

- **Step 2** Run the following command to configure the value of **unix\_permission**. root@ubuntu:~# manila type-key unix\_permission set huawei\_unixpermission:unix\_permission=755
- Step 3 Run the manila create --name test001 NFS 2 --share-type unix\_permission command to create a share that supports the preceding properties.

## 6.17 Configuring a Snapshot Directory to Be Visible

This section describes how to configure a snapshot directory to be visible.

#### **Procedure**

**Step 1** Run the **manila type-create show\_snapshot\_dir False** command to create a share type.

In the preceding command, **show\_snapshot\_dir** indicates the share type name, which is specified by users. **False** indicates that the **driver\_handles\_share\_servers** function is disabled. To enable the function, change it to **True**. root@ubuntu:~# manila type-create show\_snapshot\_dir False

- **Step 2** Run the following command to configure the value of **show\_snapshot\_dir**. root@ubuntu:~# manila type-key show\_snapshot\_dir set huawei\_showsnapshotdir:show\_snapshot\_dir=true
- Step 3 Run the manila create --name test001 NFS 2 --share-type show\_snapshot\_dir command to create a share that supports the preceding properties.

----End

# 6.18 Configuring Certificate Authentication

This section describes how to configure certificate authentication to connect to back-end storage.

#### **Procedure**

- **Step 1** Use a remote access tool, such as PuTTY, to log in to an OpenStack Cinder node through the management IP address.
- **Step 2** Set the configuration file of Huawei Cinder Driver. Add **SSLCertVerify** and **SSLCertPath** in the **<Storage>** section.

#### 

- **SSLCertVerify** indicates whether to enable certificate authentication. The value can be **True** or **False**. If the parameter is not set, the value is **False** by default. You are advised to enable certificate authentication.
- **SSLCertPath** indicates the certificate path for authentication and is valid only when **SSLCertVerify** is set to **True**.

# **7** Best Practices

7.1 Quick Interconnection with Huawei Storage

# 7.1 Quick Interconnection with Huawei Storage

This section describes how to configure Huawei Manila Driver on OpenStack to interconnect with Huawei storage systems.

#### **Configuration Process**

- **Step 1** Obtain Manila Driver. For details, see **4.1 Obtaining Manila Driver**.
- **Step 2** Create or view the file storage pool to be used on Huawei storage.
- **Step 3** Configure the **manila.conf** file and Huawei custom Manila Driver configuration file. For details, see **5 Configuring Basic Properties of Manila Driver**.
  - In /etc/manila, create a Huawei custom Manila Driver configuration file in .xml format. You can change the name of the Manila Driver configuration file based on site requirements. For example, the Manila Driver configuration file name is manila\_huawei\_conf.xml.
  - Set parameters in the Huawei custom Manila Driver configuration file.

• Configure the manila.conf file.

Add the following configuration items to the end of the /etc/manila/manila.conf file:

volume driver indicates the loaded Manila Driver file.

- **share\_backend\_name** indicates the name of the Manila share backend.
- manila\_huawei\_conf\_file indicates the Huawei custom Manila Driver 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] section, add the huawei backend.

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

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

If the service status is up, the service is started properly.

