eSDK Enterprise Storage Plugins 25.1.0

OpenStack Manila Driver Configuration Guide

Issue 01

Date 2025-06-30





Copyright © Huawei Technologies Co., Ltd. 2025. All rights reserved.

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

Trademarks and Permissions

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

Notice

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

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

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base

Bantian, Longgang Shenzhen 518129

People's Republic of China

Website: https://e.huawei.com

Security Declaration

Product Lifecycle

Huawei's regulations on product lifecycle are subject to the *Product End of Life Policy.* For details about this policy, visit the following web page:

https://support.huawei.com/ecolumnsweb/en/warranty-policy

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:

https://www.huawei.com/en/psirt/vul-response-process

For vulnerability information, enterprise customers can visit the following web page:

https://securitybulletin.huawei.com/enterprise/en/security-advisory

Initial Digital Certificate

The Initial digital certificates on Huawei devices are subject to the *Rights and Responsibilities of Initial Digital Certificates on Huawei Devices.* For details about this document, visit the following web page: https://support.huawei.com/enterprise/en/bulletins-service/ENEWS2000015789

Huawei Enterprise End User License Agreement

This agreement is the end user license agreement between you (an individual, company, or any other entity) and Huawei for the use of the Huawei Software. Your use of the Huawei Software will be deemed as your acceptance of the terms mentioned in this agreement. For details about this agreement, visit the following web page:

https://e.huawei.com/en/about/eula

Lifecycle of Product Documentation

Huawei after-sales user documentation is subject to the *Product Documentation Lifecycle Policy.* For details about this policy, visit the following web page:

https://support.huawei.com/enterprise/en/bulletins-website/ENEWS2000017761

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.		
⚠ WARNING	Indicates a hazard with a medium level of risk which, if not avoided, could result in death or serious injury.		
⚠ 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.		
◯ 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	2025-06-30	This issue is the first official release.

Contents

About This Document	III
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 in Non-Containerized Mode	8
4.3 Deploying Huawei Manila Driver for the Manila Share Service in Containerized Mode	S
5 Configuring Basic Properties of Manila Driver	12
5.1 Configuring Manila Driver in Non-Containerized Mode	
5.2 Configuring Manila Driver in Containerized Mode	18
5.3 Verifying the Manila Share Service	19
6 (Optional) Configuring Advanced Properties of Manila Driver	20
6.1 Configuring the Thin/Thick Property	
6.2 Configuring the SmartDedupe Property	21
6.3 Configuring the SmartCompression Property	22
6.4 Configuring the SmartCache Property	22
6.5 Configuring the SmartPartition Property	23
6.6 Configuring the SmartQoS Property	23
6.6.1 Configuring the Maximum IOPS	24
6.6.2 Configuring the Minimum IOPS	25
6.6.3 Configuring the Maximum Bandwidth	25
6.6.4 Configuring the Minimum Bandwidth	26
6.6.5 Configuring the Maximum Latency	27
6.6.6 Configuring Multiple Parameters	27
6.7 Creating a Share in a Specified Storage Pool	28
6.8 Creating a Share on a Disk of a Specified Type	28
6.9 Configuring the SectorSize Property	29
6.10 Configuring the Remote Replication Property	30
6.11 Configuring the Owning Controller of a File System	31
6.12 Configuring the NFS Client Permission	31

6.13 Configuring the HyperMetro Property	32
6.14 Configuring filesystem:mode	34
6.15 Configuring the Percentage of the Reserved Snapshot Space	35
6.16 Configuring the UNIX Permission on the Root Directory of a File System	35
6.17 Configuring a Snapshot Directory to Be Visible	36
6.18 Configuring the Advanced Parameter for File Systems	36
6.19 Configuring Certificate Authentication	36
7 Best Practices	38
7.1 Quick Interconnection with Huawei Storage	38

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/Antelope/ Bobcat/Caracal/ Dalmatian	 OceanStor V5: V500R007C61/V500R007C70 OceanStor 6.1.3/6.1.5/6.1.6/6.1.7/6.1.8/ V700R001C00 OceanStor Dorado 6.1.0/6.1.2/6.1.3/6.1.5/6.1.6/6.1.7/6.1.8/ V700R001C00

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

Feature	Normal Share	HyperMetro Share	Remarks
Create Share	√	√	N/A
Delete Share	√	√	N/A
Allow access	√	√	N/A
Deny access	√	√	N/A
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	√	x	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

Feature	Normal Share	HyperMetro Share	Remarks
Ensure Share	√	√	N/A
Create a share on a certain storage pool	√	√	N/A
Create a share with a certain disk type	√	√	N/A
SectorSize	√	√	N/A
SmartPartition	√	√	SmartPartition can be configured only for the local HyperMetro share.
SmartCache	√	√	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 or delete a snapshot, perform a rollback with 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 V5	OceanStor	OceanStor Dorado
Thin share	Supported	Supported	Supported
Thick share	Supported	Not supported	Not supported
SmartDedupe	Supported (only for thin share)	Supported (only for thin share)	Supported (only for thin share)
SmartCompressio n	Supported (only for thin share)	Supported (only for thin share)	Supported (only for thin share)
SmartCache	Supported	Not supported	Supported
SmartPartition	Supported	Not supported	Not supported

Feature	OceanStor V5	OceanStor	OceanStor Dorado
SmartQoS • Protection policy parameters: latency, minIOPS, and minBandWidt h	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.
 Restriction policy parameters: maxIOPS and maxBandWidt h 			

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/Antelope	-

Feature	Sub-feature	Description	Remarks
Configuration	XML configuration file	 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: <>&"" Set Product based 	-
Policy	SmartQoS	on the actual back-end storage model. • Protection policy	Restriction policy
roucy		parameters and restriction policy parameters are mutually exclusive. If they are configured at the same time, the share will fail to be created.	parameters: maxIOPS and maxBandWidth Protection policy parameters: minIOPS, minBandWidth, and latency
		• In the QoS property, IOType is mandatory. If IOType is not configured in a QoS policy, the share will fail to be created.	

4 Installing Manila Driver

- 4.1 Obtaining Manila Driver
- 4.2 Deploying Huawei Manila Driver for the Manila Share Service in Non-Containerized Mode
- 4.3 Deploying Huawei Manila Driver for the Manila Share Service in Containerized Mode

4.1 Obtaining Manila Driver

Procedure

- **Step 1** Open a browser and enter https://github.com/Huawei/OpenStack_Driver 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.

Huawei OpenStack Manila Driver warehouse maintains 18 stable versions (M-D), ensuring long-term stable running of historical versions.

----End

4.2 Deploying Huawei Manila Driver for the Manila Share Service in Non-Containerized Mode

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.

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

----End

4.3 Deploying Huawei Manila Driver for the Manila Share Service in Containerized Mode

Step 1 Log in to a node that has a Manila Share container image and run the **docker image ls** command to query the Manila Share container image.

```
| Trootegopenstack-nodel cinder]# docker image ls | grep manila | quay.io/openstack.kolla/manila-share | 2023.l-rocky-9 | 0158ef4bc2lb | 3 weeks ago | 1.59GB | quay.io/openstack.kolla/manila-data | 2023.l-rocky-9 | 7ce72ld4ae8 | 3 weeks ago | 1.36GB | quay.io/openstack.kolla/manila-scheduler | 2023.l-rocky-9 | 9e688df52db6 | 3 weeks ago | 1.33GB | quay.io/openstack.kolla/manila-api | 2023.l-rocky-9 | e2da71544592 | 3 weeks ago | 1.33GB |
```

Step 2 Run the **docker save -o manila_share.tar quay.io/openstack.kolla/manila-share** command to back up the Manila Share container image to an image file.

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

NOTE
```

manila_share.tar is a user-defined name of the backup image, and quay.io/ openstack.kolla/manila-share is the name of the Manila Share container image. **Step 3** Create a temporary directory named **huawei** in any directory.

mkdir huawei

Step 4 Copy the obtained Huawei Manila Driver code files of the corresponding version to the **huawei** directory. The minimum permission required for the Manila Driver code files is **644**.

```
# ls -l
-rw-r--r--. 1 root root 4661 Dec 7 19:14 constants.py
-rw-r--r--. 1 root root 44837 Dec 7 19:14 helper.py
-rw-r--r--. 1 root root 12795 Dec 7 19:14 huawei_config.py
-rw-r--r--. 1 root root 98273 Dec 7 19:13 huawei_nas.py
-rw-r--r--. 1 root root 12082 Dec 7 19:13 huawei_utils.py
-rw-r--r--. 1 root root 7684 Dec 7 19:13 hypermetro.py
-rw-r--r--. 1 root root 21 Dec 7 19:14 __init__.py
-rw-r--r--. 1 root root 4285 Dec 7 19:13 manager.py
-rw-r--r--. 1 root root 12870 Dec 7 19:13 replication.py
-rw-r--r--. 1 root root 10835 Dec 7 19:14 smartx.py
```

Step 5 Run the **docker ps | grep manila** command to list Manila Share containers. In the command output, **manila_share** indicates the container name.

Step 6 Run the **docker exec -it -u 0 manila_share bash** command to access the **manila_share** container.

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

Step 7 Run the python -c "from manila.share import drivers; print (drivers.__path__)" command to query the path of the original Huawei Manila Driver code.

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

□ NOTE

- manila_share is the container name obtained in Step 5.
- /var/lib/kolla/venv/lib64/python3.9/site-packages/manila/share/drivers is the path of the original Huawei Manila Driver code.
- If an error is reported, change the preceding command to **python3 -c "from manila.share import drivers; print (drivers.__path__)"** based on the Python environment variables.
- **Step 8** Run the **exit** command to exit the **manila_share** container, go to the upper-level directory of the **huawei** directory created in **Step 3**, and copy the **huawei** directory to the path of the original Huawei Manila Driver code in **Step 7**.

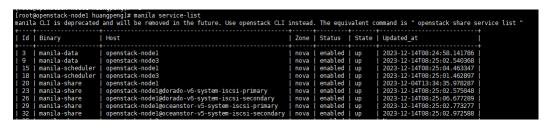
```
(manila-share)[root@openstack-nodel /]# exit
exit
[root@openstack-nodel huawei]# cd ..
[root@openstack-nodel huawei]# docker cp huawei/ manila_share:/var/lib/kolla/venv/lib64/python3.9/site-packages/manila/share/drivers
```

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

Step 9 Run the following command to restart the Manila Share service (**manila_share** indicates the container name obtained in **Step 5**).

docker restart manila_share

Wait about 30 seconds and run the **manila service-list** command. If the following information is displayed and the value of **State** is **up**, the Manila Share service is successfully started.



----End

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 in Non-Containerized Mode
- 5.2 Configuring Manila Driver in Containerized Mode
- 5.3 Verifying the Manila Share Service

5.1 Configuring Manila Driver in Non-Containerized Mode

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.

<?xml version='1.0' encoding='UTF-8'?>
<config>

- <Storage>
- <Product>Dorado</Product>
- <LogicalPortIP>x.x.x.x</LogicalPortIP>
- <DNS>x.x.x.x</DNS>
- <RestURL>https://x.x.x.x:8088/deviceManager/rest/</RestURL>
- <UserName>xxx</UserName>
- <UserPassword>xxx</UserPassword>
- </Storage>
- <Filesystem>
- <StoragePool>xxx</StoragePool>
- </Filesystem>
- </config>

Table 5-1 Mandatory parameters

Parameter	Description	Default Value	Remarks
Product	Storage product type. The value can be V5 , V6 , or Dorado .	-	-
LogicalPortIP	IP address of a logical port. You can configure multiple IP addresses separated by semicolons (;).	-	This parameter is mandatory when the DNS parameter is not specified. If both parameters are configured, the priority of this parameter is lower than that of the DNS 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. NOTICE Do not directly use the super administrator account.	-	-
UserPassword	Password of the storage administrator.	-	-

Parameter	Description	Default Value	Remarks
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>
    <CAPACITYTHRESHOLD>90</CAPACITYTHRESHOLD>
   </Advanced>
   <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

Parameter	Description	Default Value	Туре
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
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

Parameter	Description	Default Value	Туре
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 permission; 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
Advanced	Advanced parameter set when a file system is created. For details about the supported parameter, see Table 5-3.	-	Optional
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.

Table 5-3 Advanced parameter

Parameter	Description	Remarks
CAPACITYTHRESH OLD	Capacity alarm threshold percentage of file systems.	The default value and value range are subject to the product documentation.

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

- 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

----End

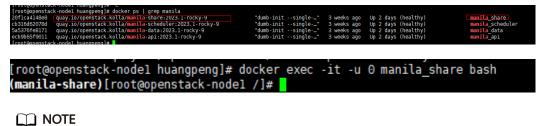
5.2 Configuring Manila Driver in Containerized Mode

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to an OpenStack Manila Share node through the management IP address. Run the following command on a host to find the manila.conf file and configure parameters by referring to Step 4 and Step 5.

find / -name manila.conf | grep manila

Step 2 Run the **docker ps | grep manila** command to query the Manila Share container. Then run the docker exec -it -u 0 manila share bash command to access the Manila Share container.



manila_share is the name of the Manila Share container.

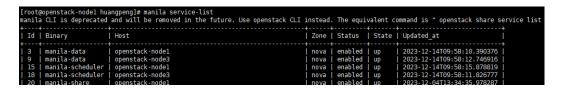
- Step 3 Create a Huawei Manila Driver configuration file in xml format in the /etc/manila directory of the Manila Share container and configure parameters by referring to Step 2 and Step 3. You can customize the configuration file name, for example, manila huawei conf.xml.
- **Step 4** Change the owner and user group of the Huawei Manila Driver configuration file to be the same as those of the /etc/manila/manila.conf file.

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

Step 5 Run the following command to restart the Manila Share service.

docker restart manila share

Wait about 30 seconds and run the manila service-list command. If the following information is displayed and the value of **State** is **up**, the Manila Share service is successfully started.



----End

5.3 Verifying the Manila Share Service

Precautions

Except for OceanStor Dorado 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

----End

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 the Advanced Parameter for File Systems
- 6.19 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:
		• 0 : read I/O
		• 1: write I/O
		• 2: 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.	0: synchronous1: asynchronous
allsquash	Permission constraint.	0: all_squash1: no_all_squash
rootsquash	root permission constraint.	0: root_squash1: no_root_squash
secure	Source port verification constraint.	 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 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 remote_backend 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.

----End

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.

----End

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 the Advanced Parameter for File Systems

Procedure

Step 1 Run the **manila type-create filesystem_advanced False** command to create a share type.

In the preceding command, **filesystem_advanced** 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 filesystem_advanced False

- Step 2 Run the following command to configure the advanced parameter for file systems. In the preceding command, filesystem_advanced: is a fixed prefix, and CAPACITYTHRESHOLD is the advanced parameter to be configured. For details about the supported advanced parameter, see Table 5-3.

 manila type-key filesystem advanced set filesystem advanced:CAPACITYTHRESHOLD=90
- Step 3 Run the manila create --name test001 NFS 2 --share-type filesystem_advanced command to create a share that supports the preceding properties.

----End

6.19 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 Malina node through the management IP address.
- **Step 2** Set the configuration file of Huawei Malina Driver. Add **SSLCertVerify** and **SSLCertPath** in the **<Storage>** section.

□ NOTE

- **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**.

----End

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.

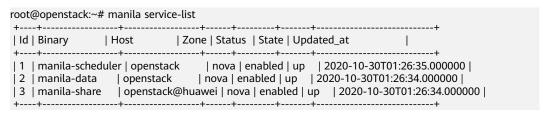
[huawei] share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver manila_huawei_conf_file = /etc/manila/manila_huawei_conf.xml share_backend_name = huawei driver_handles_share_servers = False

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.



----End