

OpenStack Manila Driver Configuration Guide

Issue 01

Date 2016-08-26



Copyright © Huawei Technologies Co., Ltd. 2016. 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

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base

Bantian, Longgang Shenzhen 518129

People's Republic of China

Website: http://www.huawei.com
Email: support@huawei.com

Contents

1 Overview	1
2 Version Mapping	2
3 Manila Driver Installation&deployment	6
3.1 Obtain Manila Driver	7
3.2 Ubuntu Environment Deployment	7
4 Manila Driver Basis Properties Configuration	9
5 Manila Driver Advanced Properties Configuration	12
5.1 Property Configuration	14
5.2 SmartDedupe Property Configuration	14
5.3 SmartCompression Property Configuration	14
5.4 SmartCache Property Configuration	15
5.5 SmartPartition Property Configuration	15
5.6 SmartQoS Property Configuration	15
5.6.1 QoS Configuration Versions	16
5.6.2 Configuring the Control IOPS	16
5.6.2.1 Configuring the Maximum Control IOPS	16
5.6.2.2 Configuring the Minimum Control IOPS	16
5.6.3 Configuring the Control Bandwidth	17
5.6.3.1 Configuring the Maximum Control Bandwidth	17
5.6.3.2 Configuring the Minimum Control Bandwidth	17
5.6.4 Configuring the Control Latency	18
5.6.5 Configuring Multiple Control Policies	
5.7 Assign StoragePool Configuration	19
5.8 Assign DiskType Configuration	19
5.9 SectorSize Configuration.	19
5.10 Replication Configuration.	20

$\mathbf{1}$ Overview

This chapter describes the definition of Manila Driver.

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

2 Version Mapping

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

Table 2-1 Version mappings among the Manila Driver, Huawei storage system and OpenStack. (\checkmark : support, x: unsupport)

存储产品版本	Kilo	Liberty	Mitaka	Newton
OceanStor V3 V300R002C10	√	√	✓	√
OceanStor V3 V300R003C00	√	√	✓	√
OceanStor 2200 V3 V300R005C00	X	X	x	X
OceanStor 2600 V3 V300R005C00	√	√	√	√
OceanStor 18500/18800 V300R003C00	√	√	√	√

Table 2-2 Version mappings among the Manila Driver, Features and OpenStack. (\checkmark : support, x: unsupport)

特性	Kilo	Liberty	Mitaka	Newton
Create Share	√	✓	√	✓
Delete Share	√	✓	√	√
Allow access	√	✓	√	√
Deny access	√	✓	√	√
Create Snapshot	√	√	√	√

特性	Kilo	Liberty	Mitaka	Newton
Delete Snapshot	√	√	√	√
Manage/ Unmanage Share	X	√	√	√
Extend Share	X	√	√	√
Shrink Share	х	√	√	√
SmartCompression	X	√	√	√
SmartDedupe	X	√	√	√
SmartCache	X	√	√	√
SmartThin/Thick	Х	√	√	√
SmartPartition	Х	√	√	√
SmartQoS	х	X	√	√
Multi-tenancy	х	X	√	√
Ensure Share	X	Х	√	√
Create Share from Snapshot	X	X	√	√
Manage/ Unmanage Snapshot	х	х	х	√
Assign StoragePool	х	X	√	√
Assign DiskType	X	X	√	√
SectorSize	х	X	√	√
Replication	x	X	X	√

Table 2-3 Version mappings among the Manila Driver, Features and Huawei storage system. (\checkmark : support, x: unsupport)

特性	OceanStor V3			OceanStor 18500/1880 0	
	V3R2	V3R3	2200 V3R5	2600 V3R5	V3R3
Create Share	√	√	X	√	√
Delete Share	√	√	x	√	✓

特性	OceanStor V3				OceanStor 18500/1880 0
Allow access	√	√	x	√	√
Deny access	√	√	X	√	√
Create Snapshot	√	1	X	√	1
Delete Snapshot	√	√	X	✓	✓
Manage Share	√	√	x	√	✓
Extend Share	√	1	x	✓	✓
Shrink Share	√	√	X	√	√
SmartCompr ession	√	1	X	√	1
SmartDedup e	√	√	х	√	√
SmartCache	√	√	X	√	√
SmartThin/ Thick	√	√	X	✓	√
SmartPartiti on	√	√	x	1	✓
SmartQoS	X	√	X	√	√
Multi- tenancy	√	√	x	√	√
Ensure Share	✓	✓	X	1	√
Create Share from Snapshot	√	√	Х	√	√
Manage/ Unmanage Snapshot	√	√	х	√	√
Assign StoragePool	√	√	X	√	√

特性	OceanStor V3			OceanStor 18500/1880 0	
Assign DiskType	√	√	X	√	√
SectorSize	√	√	X	√	√
Replication	√	√	X	√	√

3 Manila Driver Installation&deployment

- 3.1 Obtain Manila Driver
- 3.2 Ubuntu Environment Deployment

3.1 Obtain Manila Driver

Two ways to obtain OpenStack Driver:

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

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

Steps to get Manila Driver are as follows:

MOTE

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

- **Step 1** Enter the above warehouse address in the browser, for example Huawei warehouse address: https://github.com/huaweistorage/OpenStack_Driver.
- **Step 2** Click the "Download ZIP" bottom to download Driver, and unzip it.
- **Step 3** Find "Manila" catalog in which the Diver is extracted, there are multiple OpenStack Driver for different OpenStack version, choose the corresponding Driver.

----End

3.2 Ubuntu Environment Deployment

The OpenStack standard deployment steps are as follows:

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



CAUTION

On my host, the version of Python is 2.7, if other version is used, make corresponding changes to the Driver path.

- **Step 2** Copy OpenStack Manila Driver to Manila Driver installation directory, the default directory refer to step 1.
- **Step 3** Refer to chapter 4 and 5 to do the configuration.
- **Step 4** After configuration, restart Manila-Share service:

service manila-share restart

Step 5 Check the status of services by inputting manila service-list, if the "State" status of Manila-Share is up, that means Manila-Share is OK.



Except the above method, check /var/log/manila/manila-share.log to confirm whether Manila-Share is OK

4 Manila Driver Basis Properties Configuration

This chapter describes how to configure Huawei Manila Driver.

Procedure

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

manila huawei conf.xml is used as an example.

• Example for configuring a storage system:

```
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf.xml
driver handles share servers = False
```

• Example for configuring multiple storage systems:

```
enabled_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
driver_handles_share_servers = False
[huawei_manila_2]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_2.xml
driver_handles_share_servers = False
```

NOTE

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

Details about the driver configuration file are as follows:

```
<UserPassword>xxxxxxxxxx</UserPassword>
         <SnapshotSupport>True</SnapshotSupport>
        <ReplicationSupport>False
  </Storage>
  <Filesystem>
        <StoragePool>xxxxxxxxxx;xxxxxxxx</StoragePool>
        <SectorSize>64</SectorSize>
        <WaitInterval>3</WaitInterval>
        <Timeout>60</Timeout>
        <NFSClient>
             <IP>x.x.x.x</IP>
        </NFSClient>
         <CIFSClient>
            <UserName>xxxxxxxxxx</UserName>
            <UserPassword>xxxxxxxxx</UserPassword>
        </CIFSClient>
  </Filesystem>
</Config>
```

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

Table 4-1 Parameters in the configuration file

Parameter	Default Value	Description	Туре
Product	V3	Storage product model.	Mandatory.
LogicalPortIP	-	Logical port IP address.	Mandatory when not in multi-tenancy mode.
Port	-	Port name list of bond port or ETH port, used to create vlan and logical port.	If <port> is not configured, then will choose an online port on the array.</port>
RestURL	-	Access address of the REST interface.	Mandatory.
UserName	-	User name of an administrator.	Mandatory.
UserPassword	-	Password of an administrator.	Mandatory.
StoragePool	-	Name of a storage pool to be used.	Mandatory.
SectorSize	64	The size of the disk blocks, optional value can be "4", "8", "16", "32" or "64", and the units is KB.	Optional.
SnapshotSupport	True	Support snapshot or not.	Optional.

Parameter	Default Value	Description	Туре
ReplicationSupport	False	Support replication or not.	Optional.
WaitInterval	3	Interval for querying file system status. The unit is second.	Optional.
Timeout	60	Timeout interval for waiting command execution of a storage device to complete. The unit is second.	Optional.
NFSClient\IP	-	Backend IP in admin network to use for mounting NFS share.	Mandatory when create a share from snapshot.
CIFSClient \UserName	-	Backend user name in admin network to use for mounting CIFS share.	Mandatory when create a share from snapshot.
CIFSClient \UserPassword	-	Backend password in admin network to use for mounting CIFS share.	Mandatory when create a share from snapshot.

NOTE

- You can configure multiple RestURL, storage pools and ports in the configuration file and separate them using semicolons(;).
- For details about share configuration information, see the **show share** command in the specific command-line interface (CLI) document.
- Snapshot and Replication can't be configured at the same time.

Step 3 Restart the Manila services.

5 Manila Driver Advanced Properties Configuration

This chapter describes how to configure advanced storage properties.

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

Prerequisites:

You need to run the **manila type-create test_share_type False** command to create a share type.

Creating a share type named **test_share_type** is used as an example.

- 5.1 Property Configuration
- 5.2 SmartDedupe Property Configuration
- 5.3 SmartCompression Property Configuration
- 5.4 SmartCache Property Configuration
- 5.5 SmartPartition Property Configuration
- 5.6 SmartQoS Property Configuration
- 5.7 Assign StoragePool Configuration
- 5.8 Assign DiskType Configuration
- 5.9 SectorSize Configuration

5.10 Replication Configuration

5.1 Property Configuration

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

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

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

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

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

----End

NOTE

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

5.2 SmartDedupe Property Configuration

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

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

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

----End

NOTE

Only "Thin" share supports SmartDedupe.

Refer to section 4.1 to configure thin property.

5.3 SmartCompression Property Configuration

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

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

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

NOTE

Only "Thin" share supports SmartCompression. Refer to section 4.1 to configure thin property.

5.4 SmartCache Property Configuration

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

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

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

```
root@ubuntu:~#manila type-key test_share_type set
huawei smartcache:cachename='test name'
```

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

----End

5.5 SmartPartition Property Configuration

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

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

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

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

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

----End

5.6 SmartQoS Property Configuration

About This Chapter

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

Protection policies: latency, minIOPS, minBandWidth

Restriction policies: maxIOPS, maxBandWidth



CAUTION

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

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

5.6.1 QoS Configuration Versions

Table 5-1 Support for configurations of storage systems with multiple QoS properties

Storage Product Model	Storage System Version
18000	V300R003C00
V3	V300R003C00

5.6.2 Configuring the Control IOPS

Configuring the control IOPS involves configuring the maximum control IOPS and minimum control IOPS.

5.6.2.1 Configuring the Maximum Control IOPS

This section describes how to configure the maximum Control IOPS.

Procedure

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

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

Run the following command to configure QoS control property parameters.

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

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

----End

5.6.2.2 Configuring the Minimum Control IOPS

This section describes how to configure the minimum Control IOPS.

Procedure

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

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

Run the following command to configure QoS control property parameters.

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

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

----End

5.6.3 Configuring the Control Bandwidth

Configuring the control bandwidth involves configuring the maximum control bandwidth and minimum control bandwidth.

5.6.3.1 Configuring the Maximum Control Bandwidth

This section describes how to configure the maximum control bandwidth.

Procedure

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

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

Run the following command to configure QoS control property parameters.

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

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

----End

5.6.3.2 Configuring the Minimum Control Bandwidth

This section describes how to configure the minimum control bandwidth.

Procedure

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

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

Run the following command to configure QoS control property parameters.

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

• minBandWidth: indicates the minimum BANDWIDTH. The value is an integer larger than 0 and expressed in MB/s.

- **IOType** (mandatory): indicates the read and write type.0 indicates the control read I/Os, 1 indicates the control write I/Os, 2 indicates the control read and write I/Os.
- Step 2 Run manila create --name test001 NFS 2 --share-type test_share_type to create a share that supports the preceding properties.

----End

5.6.4 Configuring the Control Latency

This section describes how to configure the minimum control latency.

Procedure

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

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

Run the following command to configure QoS control property parameters.

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

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

----End

5.6.5 Configuring Multiple Control Policies

This section describes how to configure multiple control policies.

Procedure

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

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

Run the following command to configure QoS control property parameters.

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

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

5.7 Assign StoragePool Configuration

This section describes how to configure the assign storagepool to create a share.

Procedure

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

 "target_pool_type"indicates the name of a share type. "False"indicates that the multi-tenant mode is not supported, when configured to "True" means that the multi-tenant mode is supported.
- **Step 2** Run the following command to configure the assign storagepool.
 - Configure a single assign pool root@ubuntu:~# manila type-key target pool type set pool name=StoragePool001
 - Configure multiple assign pools.

 root@ubuntu:~# manila type-key target_pool_type set pool_name="<or>
 StoragePool001 <or> StoragePool002"
- Step 3 Run manila create --name test001 NFS 2 --share-type target_pool_type to create a share that supports the preceding properties.

----End

5.8 Assign DiskType Configuration

This section describes how to configure the assign disk type to create a share.

Procedure

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

 "disk_type"indicates the name of a share type. "False"indicates that the multi-tenant mode is not supported, when configured to "True" means that the multi-tenant mode is supported.
- **Step 2** Run the following command to configure the assign disk type.
 - Configure one type of disk type.

 root@ubuntu:~# manila type-key disk_type set huawei_disk_type=sas
 - Configure multiple types of disk type.

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

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

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

----End

5.9 SectorSize Configuration

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

Procedure

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

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

Configure the value of sectorsize:

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

NOTE

'SectorSize' is the size of the disk blocks, optional value can be "4", "8", "16", "32" or "64", and the units is KB. If "sectorsize" is configured in both share_type and xml file, the value of sectorsize in the share_type will be used. If "sectorsize" is configured in neither share_type nor xml file, huawei storage backends will provide a default value(64) when creating a new share.

----End

5.10 Replication Configuration

This section describes how to configure Replication.

Prerequisites

Step 1 Configure two back-ends in "/etc/manila/manila.conf" file for replication.

```
[DEFAULT]
...
enabled_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
driver_handles_share_servers = False
replication_domain = huawei_domain

[huawei_manila_2]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_2.xml
driver_handles_share_servers = False
replication_domain = huawei_domain
```

NOTE

- The replication_domain option is a backend specific StrOpt option to be used within manila.conf.
 The value can be any ASCII string. Two backends that can replicate between each other would have the same replication_domain.
- For more details about "manila_huawei_conf_1.xml" and "manila_huawei_conf_2.xml" please refer to chapter 4.

Step 2 Restart the Manila services.

Procedure

- **Step 1** Run the **manila type-create replication_type False** command to create a share type. "replication_type" indicates the name of a share type. "False" indicates that the multi-tenant mode is not supported, when configured to "True" means that the multi-tenant mode is supported.
- Step 2 Run the following command to configure the type of replication.

 root@ubuntu:~# manila type-key replication type set replication type=dr
- Step 3 Run manila create --name test001 NFS 2 --share-type replication_type to create a share that supports the preceding properties.
- Step 4 Run manila share-replica-create test001 to create replication for share test001.

NOTE

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