eSDK Cloud Storage Plugins
25.2.0

OpenStack Manila Driver Configuration Guide

Issue	01
Date	2025-09-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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	2025-09-30	This issue is the first official release.

Contents

About This Document..iii

1 Overview..1

2 Version Mappings..2

3 Specifications...5

4 Installing Manila Driver..7

4.1 Obtaining Manila Driver...7

4.2 Deploying Huawei Manila Driver for the Manila Share Service in Non-Containerized Mode....................7

4.3 Deploying Huawei Manila Driver for the Manila Share Service in Containerized Mode.............................8

4.4 Deploying Huawei Manila Driver for the Manila Share Service in Containerized Mode (for China Mobile Suzhou Research Center)...10

5 Configuring Basic Properties of Manila Driver...15

5.1 Configuring Manila Driver in Non-Containerized Mode...15

5.2 Configuring Manila Driver in Containerized Mode...19

5.3 Verifying the Manila Share Service..20

6 Upgrading Manila Driver...21

7 (Optional) Configuring Advanced Properties of Manila Driver...................................22

7.1 Creating a Share in a Specified Storage Pool...22

7.2 Creating a Share Using a Specified ACL Policy Type...23

7.3 Configuring One-Way Authentication..24

7.4 Configuring Two-Way Authentication..25

8 Appendix...29

8.1 Security Hardening..29

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
Train	OceanStor Pacific 8.1.5/8.2.0/8.2.1/ V800R001C10

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

Feature	NFS Share	CIFS Share	Remarks
OpenStack	Train	Train	N/A
Create Share	√	√	N/A
Delete Share	√	√	N/A
Allow access	√	√	Authorization of IPv4/IPv6 addresses and address segments is supported.
Deny access	√	√	N/A
Create Snapshot	√	√	N/A
Delete Snapshot	√	√	N/A
Revert Snapshot	√	√	N/A
Extend Share	√	√	N/A
Ensure Share	√	√	N/A
Get share pool	√	√	N/A

Feature	NFS Share	CIFS Share	Remarks
Reload QoS	√	√	Feature customized for the ZTE cloud platform

NOTE

The non-standard features of the OpenStack community are described as follows:

- **Reload QoS**: QoS policies can be added, deleted, and updated for a share. The restrictions are as follows:
  - QoS parameters support only the following fields: ["maxIOPS", "maxMBPS", "total_ops_sec", "total_iops_sec", "total_bytes_sec"]
  - If **maxIOPS**, **total_ops_sec**, and **total_iops_sec** are set at the same time, only one of them takes effect. The priority is: **maxIOPS** > **total_ops_sec** > **total_iops_sec**.
  - If both **maxMBPS** and **total_bytes_sec** are set, the value of **maxMBPS** is used.
  - One of **maxIOPS**, **total_ops_sec**, and **total_iops_sec** must be set. One of **maxMBPS** and **total_bytes_sec** must be set.
  - For details about the value ranges of these parameters, see **Table 2-3**.

Table 2-3 QoS parameters

Parameter	Description	Remarks	Example
maxIOPS	Maximum IOPS.	The value is an integer or an integer of the character type ranging from 0 to 1073741824000. Value 0 indicates unlimited.	• 1000000 • '1000000'
maxMBPS	Maximum bandwidth.	The unit is MB/s. The value is an integer or an integer of the character type ranging from 0 to 1073741824. Value 0 indicates unlimited.	• 30000 • '30000'
total_iops_sec	Maximum IOPS.	The value is an integer or an integer of the character type ranging from 0 to 1073741824000. Value 0 indicates unlimited.	• 1000000 • '1000000'
total_bytes_sec	Maximum bandwidth.	The unit is byte/s. The value is an integer or an integer of the character type ranging from 0 to 1125899906842624. Value 0 indicates unlimited.	• 31457280000 • '31457280000'

Parameter	Description	Remarks	Example
total_ops_sec	Maximum IOPS.	The value is an integer or an integer of the character type ranging from 0 to 1073741824000. Value 0 indicates unlimited.	● 1000000 ● '1000000'

3 Specifications

Feature	Sub-feature	Description	Remarks
Mapping platform	Native OpenStack platform	OpenStack version: Train	-
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.	All parameter values in the Huawei Manila Driver configuration file cannot contain the following XML special characters: <>&'"
Storage device	Storage pool	Multiple storage pools can be connected.	-
Specifications	Share quota	Due to the limitation of OceanStor Pacific, a maximum of 100,000 shares can be created.	-

Feature	Sub-feature	Description	Remarks
Driver backend	Number of driver backends that can be connected to the same storage system	For the same storage system, one manila-share node connects to one backend and a maximum of three manila-share nodes are supported.	The number of driver backends that can be connected to a single storage system is related to the maximum number of concurrent requests supported by the management plane of the storage system. If too many driver backends are connected, the storage system will be overloaded.

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

4.4 Deploying Huawei Manila Driver for the Manila Share Service in Containerized Mode (for China Mobile Suzhou Research Center)

4.1 Obtaining Manila Driver

Procedure

Step 1 Open a browser and visit **https://github.com/Huawei/FusionStorage_OpenStack_Driver**.

Step 2 Choose **Code** > **Download ZIP** to download the Huawei Manila Driver package to a local path.

Step 3 Decompress the package.

Step 4 In the directory generated upon decompression, find the **Manila** directory.

----End

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

Procedure

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 absolute path may vary in different systems. You can search for the code installation 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 installation 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 installation 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 all contents in the **huawei** directory of OpenStack Manila Driver obtained in **4.1 Obtaining Manila Driver** to the code installation directory of Huawei Manila Driver on the Manila Share node. For details about the default path, see **Step 1**. The copy result is as follows: The minimum permission on all Driver directories is 755, and the minimum permission on all files is 644.

```
[root@pnode1 huawei(keystone_admin)]# pwd
/usr/lib/python2.7/site-packages/manila/share/drivers/huawei
[root@pnode1 huawei(keystone_admin)]# ls -l
drwxr-xr-x 1 root root  204 Mar 07 10:51 oceanstorPacific
-rw-r--r-- 1 root root   23 Mar 07 10:51 __init__.py
```

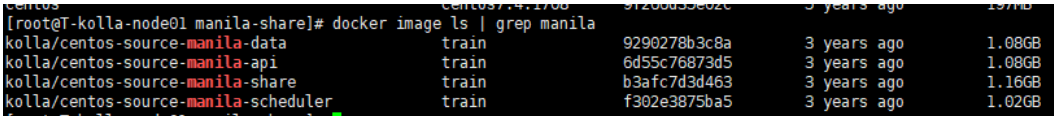
----End

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

Procedure

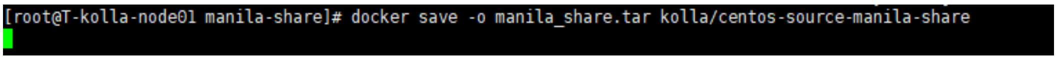
The standard deployment procedure of the OpenStack community is as follows:

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.



```
[root@T-kolla-node01 manila-share]# docker image ls | grep manila
kolla/centos-source-manila-data           train      9290278b3c8a     3 years ago     1.08GB
kolla/centos-source-manila-api            train      6d55c76873d5     3 years ago     1.08GB
kolla/centos-source-manila-share          train      b3afc7d3d463     3 years ago     1.16GB
kolla/centos-source-manila-scheduler      train      f302e3875ba5     3 years ago     1.02GB
```

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



```
[root@T-kolla-node01 manila-share]# docker save -o manila_share.tar kolla/centos-source-manila-share
```

NOTE

manila_share.tar is a user-defined name of the backup image, and **kolla/centos-source-manila-share** is the name of the Manila Share container image.

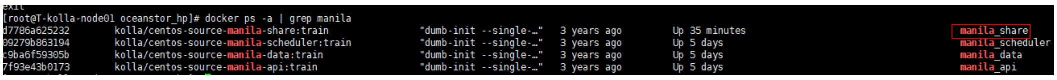
Step 3 Create a temporary directory named **huawei** in any directory.

```
# mkdir huawei
```

Step 4 Copy the obtained Huawei Manila Driver code files of the corresponding version to the **huawei** directory. The minimum permission on the Manila Driver directory is 755, and the minimum permission on all files is 644.

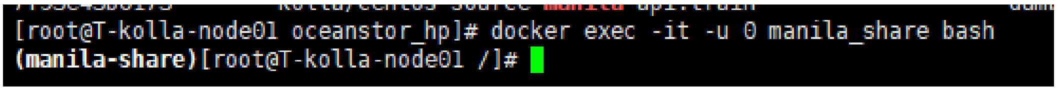
```
# ls -l
drwxr-xr-x. 5 root root  186 Aug 20 14:37 oceanstorPacific
-rw-r--r-- 1 root root   186 Aug 20 14:37 __init__.py
```

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



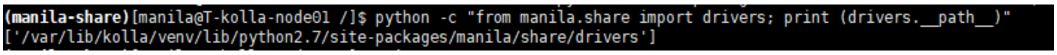
```
exit
[root@T-kolla-node01 oceanstor_hp]# docker ps -a | grep manila
d7786a625232        kolla/centos-source-manila-share:train              "dumb-init --single-…"   3 years ago         Up 35 minutes                           manila_share
09279b863194        kolla/centos-source-manila-scheduler:train          "dumb-init --single-…"   3 years ago         Up 5 days                               manila_scheduler
c9ba6f59305b        kolla/centos-source-manila-data:train               "dumb-init --single-…"   3 years ago         Up 5 days                               manila_data
7f93e43b0173        kolla/centos-source-manila-api:train                "dumb-init --single-…"   3 years ago         Up 5 days                               manila_api
```

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



```
[root@T-kolla-node01 oceanstor_hp]# docker exec -it -u 0 manila_share bash
(manila-share)[root@T-kolla-node01 /]#
```

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.

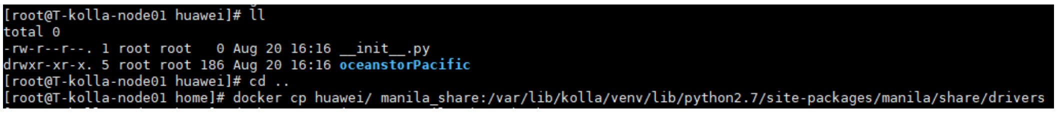


```
(manila-share)[manila@T-kolla-node01 /]$ python -c "from manila.share import drivers; print (drivers.__path__)"
['/var/lib/kolla/venv/lib/python2.7/site-packages/manila/share/drivers']
```

NOTE

- **manila_share** is the container name obtained in **Step 5**.
- **/var/lib/kolla/venv/lib/python2.7/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**.



```
[root@T-kolla-node01 huawei]# ll
total 0
-rw-r--r--. 1 root root   0 Aug 20 16:16 __init__.py
drwxr-xr-x. 5 root root 186 Aug 20 16:16 oceanstorPacific
[root@T-kolla-node01 huawei]# cd ..
[root@T-kolla-node01 home]# docker cp huawei/ manila_share:/var/lib/kolla/venv/lib/python2.7/site-packages/manila/share/drivers
```

```
docker cp huawei/ manila_share:/var/lib/kolla/venv/lib/python2.7/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.

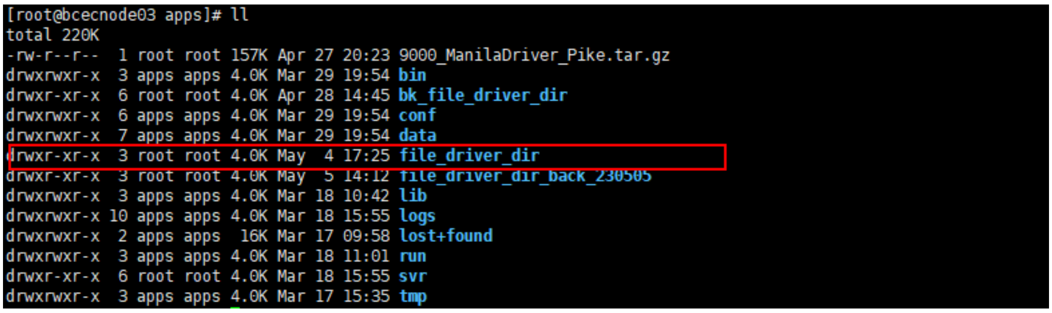
```
[root@T-kolla-node01 home]# manila service-list
+----+------------------+------------------------------------+------+---------+-------+----------------------------+
| Id | Binary           | Host                               | Zone | Status  | State | Updated_at                 |
+----+------------------+------------------------------------+------+---------+-------+----------------------------+
| 1  | manila-data      | T-kolla-node01                     | nova | enabled | up    | 2024-08-20T08:18:33.000000 |
| 2  | manila-scheduler | T-kolla-node01                     | nova | enabled | up    | 2024-08-20T08:18:35.000000 |
| 3  | manila-share     | T-kolla-node01@pacific_backend      | nova | enabled | up     |
2024-08-20T08:18:28.000000 |
```

----End

4.4 Deploying Huawei Manila Driver for the Manila Share Service in Containerized Mode (for China Mobile Suzhou Research Center)

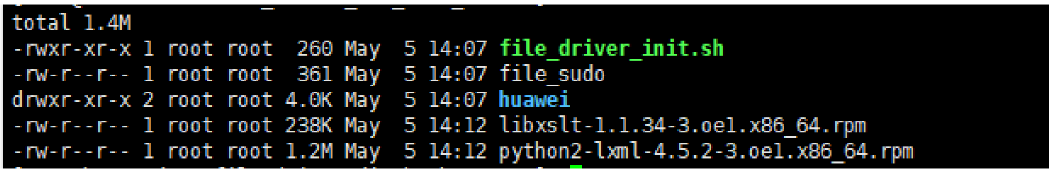
Step 1 Use a remote access tool, such as PuTTY, to log in to the node where Manila Share is to be deployed through the management IP address.

Step 2 Go to the **/apps** directory and upload the **file_driver_dir** directory obtained in **4.1 Obtaining Manila Driver** to **/apps**.



```
[root@bcecnode03 apps]# ll
total 220K
-rw-r--r--  1 root root 157K Apr 27 20:23 9000_ManilaDriver_Pike.tar.gz
drwxrwxr-x  3 apps apps 4.0K Mar 29 19:54 bin
drwxr-xr-x  6 root root 4.0K Apr 28 14:45 bk_file_driver_dir
drwxrwxr-x  6 apps apps 4.0K Mar 29 19:54 conf
drwxrwxr-x  7 apps apps 4.0K Mar 29 19:54 data
drwxr-xr-x  3 root root 4.0K May  4 17:25 file_driver_dir
drwxr-xr-x  3 root root 4.0K May  5 14:12 file_driver_dir_back_230505
drwxrwxr-x  3 apps apps 4.0K Mar 18 10:42 lib
drwxrwxr-x 10 apps apps 4.0K Mar 18 15:55 logs
drwxrwxr-x  2 apps apps  16K Mar 17 09:58 lost+found
drwxrwxr-x  3 apps apps 4.0K Mar 18 11:01 run
drwxr-xr-x  6 root root 4.0K Mar 18 15:55 svr
drwxrwxr-x  3 apps apps 4.0K Mar 17 15:35 tmp
```

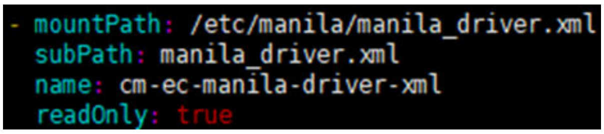
Step 3 Run the **cd file_driver_dir** command to go to the **file_driver_dir** directory and run the **chmod +x file_driver_init.sh** command to grant the execute permission on the **file_driver_init.sh** script.



```
total 1.4M
-rwxr-xr-x 1 root root  260 May  5 14:07 file_driver_init.sh
-rw-r--r-- 1 root root  361 May  5 14:07 file_sudo
drwxr-xr-x 2 root root 4.0K May  5 14:07 huawei
-rw-r--r-- 1 root root 238K May  5 14:12 libxslt-1.1.34-3.oe1.x86_64.rpm
-rw-r--r-- 1 root root 1.2M May  5 14:12 python2-lxml-4.5.2-3.oe1.x86_64.rpm
```

Step 4 Use a remote access tool (for example, PuTTY) to log in to a node managed by K8s for China Mobile Suzhou Research Center through the management IP address. The **bcec-workspace** deployment script (available only for China Mobile Suzhou Research Center) exists on the node. Assume that the Manila tool directory is **/root/bcec-workspace/2017.4.7.6/bcec-workspace/manila**. Go to the default .yaml file path provided by the China Mobile Suzhou Research Center, for example, **/root/bcec-workspace/2017.4.7.6/bcec-workspace/manila/deploy/manila-share.yaml**.

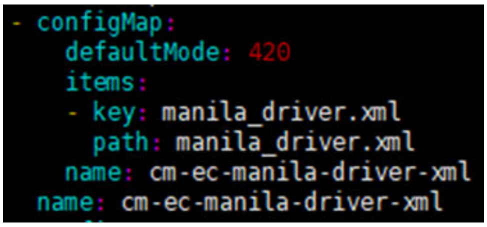
Step 5 Modify the .yaml file. Specifically, modify the fields under **volumeMounts** in **containers** as follows:



```
- mountPath: /etc/manila/manila_driver.xml
  subPath: manila_driver.xml
  name: cm-ec-manila-driver-xml
  readOnly: true
```

```
Add the following text in bold.
containers:
 ......
 volumeMounts:
  ......
 - mountPath: /etc/manila/manila_driver.xml
   subPath: manila_driver.xml
   name: cm-ec-manila-driver-xml
   readOnly: true
```

Step 6 Modify the .yaml file. Specifically, modify the fields under **volumes** in **containers** as follows:

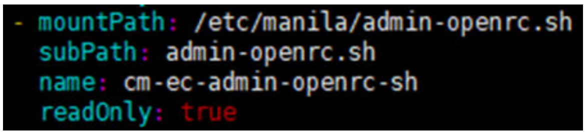


```
- configMap:
    defaultMode: 420
    items:
    - key: manila_driver.xml
      path: manila_driver.xml
    name: cm-ec-manila-driver-xml
  name: cm-ec-manila-driver-xml
```

Add the following text in bold.

```
volumes:
   ......
 - configMap:
     defaultMode: 420
     items:
     - key: manila_driver.xml
       path: manila_driver.xml
     name: cm-ec-manila-driver-xml
   name: cm-ec-manila-driver-xml
```

Step 7 Modify the .yaml file. Specifically, modify the fields under **volumeMounts** in **containers** as follows:

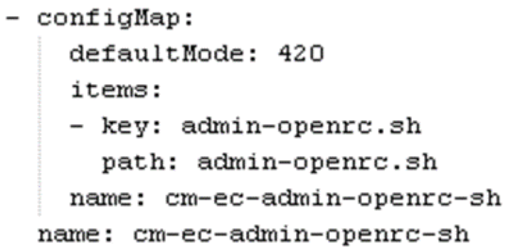


```
- mountPath: /etc/manila/admin-openrc.sh
  subPath: admin-openrc.sh
  name: cm-ec-admin-openrc-sh
  readOnly: true
```

Add the following text in bold.

```
containers:
   ......
 volumeMounts:
   ......
  - mountPath: /etc/manila/admin-openrc.sh
    subPath: admin-openrc.sh
    name: cm-ec-admin-openrc-sh
    readOnly: true
```

Step 8 Modify the .yaml file. Specifically, modify the fields under **volumes** in **containers** as follows:



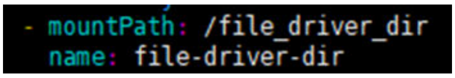
```
- configMap:
    defaultMode: 420
    items:
    - key: admin-openrc.sh
      path: admin-openrc.sh
    name: cm-ec-admin-openrc-sh
  name: cm-ec-admin-openrc-sh
```

Add the following text in bold.

```
volumes:
   ......
 - configMap:
     defaultMode: 420
     items:
     - key: admin-openrc.sh
       path: admin-openrc.sh
```

```
    name: cm-ec-admin-openrc-sh
  name: cm-ec-admin-openrc-sh
```

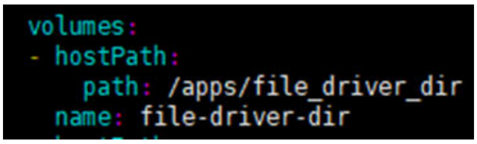
Step 9 Modify the .yaml file. Specifically, modify the fields under **volumeMounts** in **containers** as follows:



```
- mountPath: /file_driver_dir
  name: file-driver-dir
```

```
Add the following text in bold.
containers:
  ......
 volumeMounts:
   ......
  - mountPath: /file_driver_dir
    name: file-driver-dir
```

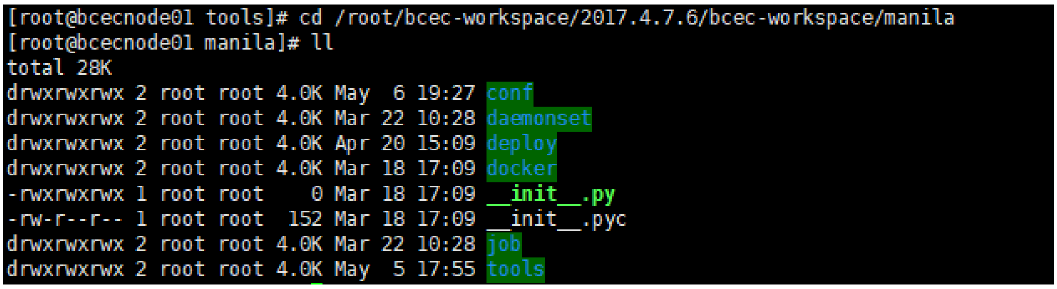
Step 10 Modify the .yaml file. Specifically, modify the fields under **volumes** in **containers** as follows:



```
volumes:
- hostPath:
    path: /apps/file_driver_dir
  name: file-driver-dir
```

```
Add the following text in bold.
volumes:
  ......
 - hostPath:
     path: /apps/file_driver_dir
   name: file-driver-dir
```

Step 11 Use a remote access tool (for example, PuTTY) to log in to a node managed by K8s for China Mobile Suzhou Research Center through the management IP address. The **bcec-workspace** deployment script (available only for China Mobile Suzhou Research Center) exists on the node. Assume that the Manila tool directory is **/root/bcec-workspace/2017.4.7.6/bcec-workspace/manila**.



```
[root@bcecnode01 tools]# cd /root/bcec-workspace/2017.4.7.6/bcec-workspace/manila
[root@bcecnode01 manila]# ll
total 28K
drwxrwxrwx 2 root root 4.0K May  6 19:27 conf
drwxrwxrwx 2 root root 4.0K Mar 22 10:28 daemonset
drwxrwxrwx 2 root root 4.0K Apr 20 15:09 deploy
drwxrwxrwx 2 root root 4.0K Mar 18 17:09 docker
-rwxrwxrwx 1 root root    0 Mar 18 17:09 __init__.py
-rw-r--r-- 1 root root  152 Mar 18 17:09 __init__.pyc
drwxrwxrwx 2 root root 4.0K Mar 22 10:28 job
drwxrwxrwx 2 root root 4.0K May  5 17:55 tools
```

Step 12 Run the **cd /root/bcec-workspace/2017.4.7.6/bcec-workspace/manila/conf** command and modify the **manila-share.conf** configuration file. Specifically, add the following backend configuration to the end of the file. The file is stored in the path of the **bcec-workspace** script tool (provided by China Mobile Suzhou Research Center). The following uses **/root/bcec-workspace/2017.4.7.6/bcec-workspace/manila/conf** as an example.

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

NOTE

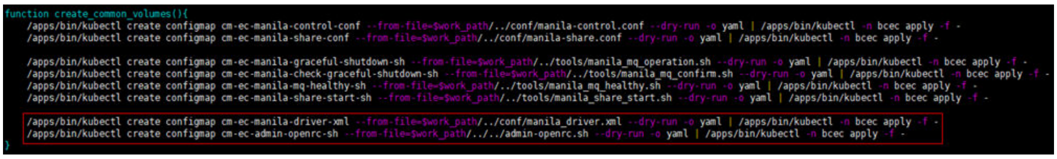
- **huawei_backend** indicates the backend name. It can be customized.
- **share_driver** indicates the type of Manila Driver to be used. In this example, the parameter is set to **manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver**.
- **manila_huawei_conf_file** indicates the path of the Huawei Manila Driver configuration file.
- If **driver_handles_share_servers** is set to **True**, you need to specify **share-network** when creating a share.

Step 13 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 14 Configure the **manila_driver.xml** file in the **/root/bcec-workspace/2017.4.7.6/bcec-workspace/manila/conf** directory. For details about related parameters, see manila_driver.xml configuration.

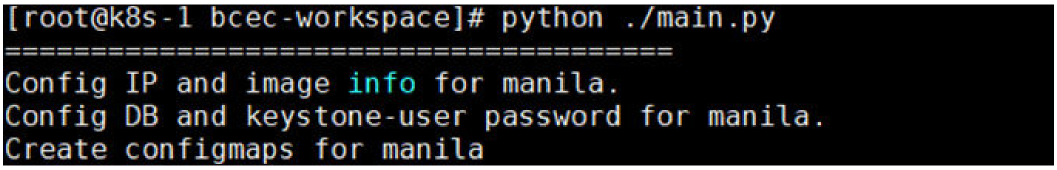
Step 15 Modify and execute **create_volume.sh**.



```
function create_common_volumes(){
    /apps/bin/kubectl create configmap cm-ec-manila-control-conf --from-file=$work_path/../conf/manila-control.conf --dry-run -o yaml | /apps/bin/kubectl -n bcec apply -f -
    /apps/bin/kubectl create configmap cm-ec-manila-share-conf --from-file=$work_path/../conf/manila-share.conf --dry-run -o yaml | /apps/bin/kubectl -n bcec apply -f -

    /apps/bin/kubectl create configmap cm-ec-manila-graceful-shutdown-sh --from-file=$work_path/../tools/manila_mq_operation.sh --dry-run -o yaml | /apps/bin/kubectl -n bcec apply -f -
    /apps/bin/kubectl create configmap cm-ec-manila-check-graceful-shutdown-sh --from-file=$work_path/../tools/manila_mq_confirm.sh --dry-run -o yaml | /apps/bin/kubectl -n bcec apply -f -
    /apps/bin/kubectl create configmap cm-ec-manila-mq-healthy-sh --from-file=$work_path/../tools/manila_mq_healthy.sh --dry-run -o yaml | /apps/bin/kubectl -n bcec apply -f -
    /apps/bin/kubectl create configmap cm-ec-manila-share-start-sh --from-file=$work_path/../tools/manila_share_start.sh --dry-run -o yaml | /apps/bin/kubectl -n bcec apply -f -

    /apps/bin/kubectl create configmap cm-ec-manila-driver-xml --from-file=$work_path/../conf/manila_driver.xml --dry-run -o yaml | /apps/bin/kubectl -n bcec apply -f -
    /apps/bin/kubectl create configmap cm-ec-admin-openrc-sh --from-file=$work_path/../../admin-openrc.sh --dry-run -o yaml | /apps/bin/kubectl -n bcec apply -f -
}
```

```
vim /root/bcec-workspace/2017.4.7.6/bcec-workspace/manila/tools/create_volume.sh

Add configmap to the create_common_volumes function and save the file.
/apps/bin/kubectl create configmap cm-ec-manila-driver-xml --from-file=$work_path/../conf/
manila_driver.xml --dry-run -o yaml | /apps/bin/kubectl -n bcec apply -f -
/apps/bin/kubectl create configmap cm-ec-admin-openrc-sh --from-file=$work_path/../../admin-openrc.sh --
dry-run -o yaml | /apps/bin/kubectl -n bcec apply -f -

sh /root/bcec-workspace/2017.4.7.6/bcec-workspace/manila/tools/create_volume.sh
```

Step 16 Run **main.py**.



```
[root@k8s-1 bcec-workspace]# python ./main.py
==========================================
Config IP and image info for manila.
Config DB and keystone-user password for manila.
Create configmaps for manila
```

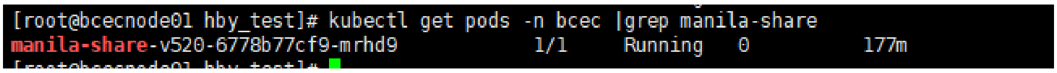
```
cd /root/bcec-workspace/2017.4.7.6/bcec-workspace/
python ./main.py
```

Step 17 Run the following commands to start the container.

```
cd /root/bcec-workspace/2017.4.7.6/bcec-workspace/manila/deploy/
kubectl delete -f manila-share.yaml
kubectl create -f manila-share.yaml
```

Step 18 If the **bcec** namespace is used, run the following command to check whether the Manila Share container is started successfully.

```
kubectl get pods -n bcec|grep manila-share
```



```
[root@bcecnode01 hby_test]# kubectl get pods -n bcec |grep manila-share
manila-share-v520-6778b77cf9-mrhd9          1/1     Running   0          177m
[root@bcecnode01 hby_test]#
```

NOTE

- If the returned status is **Running** and the value of **READY** is **1/1**, the start is successful.
- If the Manila Share container has already been started, it will be stopped first. It takes about 2 minutes to stop the Manila Share container.

----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 value of **default_share_type** in the **manila.conf** file is used.
- Ensure that the storage pool used for configuring Manila Driver exists on Huawei storage. Otherwise, you need to manually create one.
- All parameter values in the Huawei Manila Driver configuration file cannot contain the following XML special characters: <>&'"

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 parameters in the Huawei Manila Driver configuration file.

```
<?xml version='1.0' encoding='UTF-8'?>
 <config>
  <Storage>
   <Product>Pacific</Product>
   <RestURL>https://ip:8088/api/v2/</RestURL>
   <UserName>xxx</UserName>
```

```
    <UserPassword>xxx</UserPassword>
    <Reserved_percentage>15</Reserved_percentage>
    <Max_over_subscription_ratio>1</Max_over_subscription_ratio>
    <Semaphore>10</Semaphore>
    <SslCertVerify>False</SslCertVerify>
    <SslCertPath>xxx</SslCertPath>
  </Storage>
  <Filesystem>
    <StoragePool>0</StoragePool>
    <AccountName>system</AccountName>
    <ClusterDomainName>nfs-ip</ClusterDomainName>
    <RollbackRate>3</RollbackRate>
  </Filesystem>
</config>
```

Table 5-1 Parameter description

Parameter	Description	Example
Product	Storage product type. This parameter is mandatory. Currently, this parameter can be set only to: ● Pacific	Pacific
RestURL	Access address of the REST interface. This parameter is mandatory.	Enter the floating IP address and port number of OceanStor Pacific.
UserName	User of the storage administrator role. This parameter is mandatory.	manilauser **NOTICE** Do not directly use the super administrator account.
UserPassword	Password of the user of the storage administrator role. This parameter is mandatory.	-
Reserved_percentage	Percentage of reserved space. This parameter is optional. If this parameter is not set, the default value **15** is used. The value can contain only digits. You can modify this parameter to reserve capacity.	15
Max_over_subscription_ratio	Maximum oversubscription ratio. This parameter is optional. If this parameter is not specified, the default value 1 is used. Only floating-point numbers greater than or equal to 1 are supported. You can modify this parameter to implement capacity overcommitment.	1

Parameter	Description	Example
Semaphore	Controls the number of concurrent requests for invoking storage RESTful interfaces. This parameter is optional. If this parameter is not specified, the default value 20 is used. The maximum value cannot exceed 20.	20
StoragePool	Storage pool ID. This parameter is mandatory. The value can contain only digits. Multiple IDs can be configured and separated by semicolons (;).	-
AccountName	Name of the storage tenant. This parameter is mandatory. The system tenant **system** is recommended.	system
ClusterDomainName	Service IP address or domain name of the NFS/CIFS share mounted by a user. This parameter is mandatory. **NOTE** When a share is created, the driver sets this parameter to **export_locations** of the share.	-
RollbackRate	Snapshot rollback speed. This parameter is optional. If this parameter is not specified, the default value 2 is used. Only the following four levels are supported: [1, 2, 3, 4]	2

Step 3 Change the permission on the **/etc/manila/manila_huawei_conf.xml** file to be the same as the permission of the owner and user group 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 4 Add the following information to the end of the **/etc/manila/manila.conf** file. For details about the parameters, see **Table 5-2**.

Table 5-2 Parameter description

Parameter	Description	Example
share_driver	Driver type of the interconnected Huawei backend storage. This parameter is mandatory. It is used to find the corresponding Huawei storage backend driver when the service is started.	The value is fixed at **manila.share.drivers.huawei.oceanstorPacific.oceanstorpacific_nas.HuaweiNasDriver**.
manila_huawei_conf_file	Path of the Huawei Manila Driver configuration file configured in **5.1 Configuring Manila Driver in Non-Containerized Mode**. This parameter is mandatory.	/etc/manila/manila_huawei_conf.xml
share_backend_name	Share backend name, which is used to specify the value of **share_backend_name** when **share_type** is created. This parameter is mandatory.	huawei_backend
driver_handles_share_servers	Whether to enable OpenStack tenant network isolation. This parameter is mandatory. Currently, the value can only be **False**.	False

```
[huawei_backend]
share_driver = manila.share.drivers.huawei.oceanstorPacific.oceanstorpacific_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf.xml
share_backend_name = huawei_backend
driver_handles_share_servers = False
```

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 Run the following command to restart the Manila service.

```
# systemctl restart openstack-manila-share.service
```

Step 7 Wait about 30 seconds and run the following command. If the value of **State** of the Huawei backend share service is **up**, the Manila Share service is started successfully.

```
# manila service-list
```

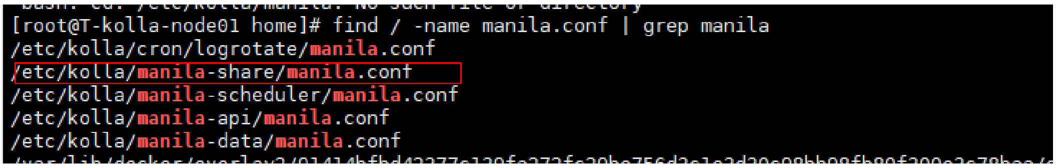
----End

5.2 Configuring Manila Driver in Containerized Mode

Procedure

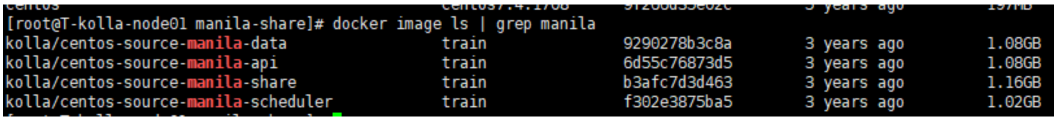
Step 1 Use a remote access tool, such as PuTTY, to log in to an OpenStack Manila Share node through the management IP address. Run the following command on a host to find the **manila.conf** file and configure parameters by referring to **Step 4** and **Step 5**.

```
find / -name manila.conf | grep manila
```

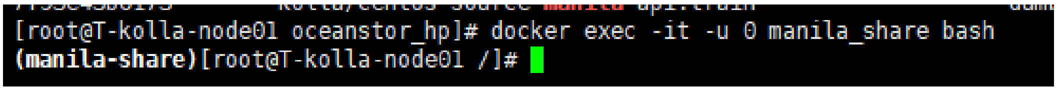


```
bash: cd: /etc/kolla/manila: No such file or directory
[root@T-kolla-node01 home]# find / -name manila.conf | grep manila
/etc/kolla/cron/logrotate/manila.conf
/etc/kolla/manila-share/manila.conf
/etc/kolla/manila-scheduler/manila.conf
/etc/kolla/manila-api/manila.conf
/etc/kolla/manila-data/manila.conf
```

Step 2 Run the **docker ps -a | 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.



```
[root@T-kolla-node01 manila-share]# docker image ls | grep manila
kolla/centos-source-manila-data          train     9290278b3c8a     3 years ago     1.08GB
kolla/centos-source-manila-api           train     6d55c76873d5     3 years ago     1.08GB
kolla/centos-source-manila-share         train     b3afc7d3d463     3 years ago     1.16GB
kolla/centos-source-manila-scheduler     train     f302e3875ba5     3 years ago     1.02GB
```



```
[root@T-kolla-node01 oceanstor_hp]# docker exec -it -u 0 manila_share bash
(manila-share)[root@T-kolla-node01 /]#
```

NOTE

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**. 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 Aug 20 15:30 manila.conf
-rw-r--r--. 1 manila manila  443 Aug 20 11:32 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.

```
[root@T-kolla-node01 home]# manila service-list
+----+------------------+------------------------------------+------+---------+-------+----------------------------+
| Id | Binary           | Host                               | Zone | Status  | State | Updated_at                 |
+----+------------------+------------------------------------+------+---------+-------+----------------------------+
| 1  | manila-data      | T-kolla-node01                     | nova | enabled | up    | 2024-08-20T08:18:33.000000 |
| 2  | manila-scheduler | T-kolla-node01                     | nova | enabled | up    | 2024-08-20T08:18:35.000000 |
| 3  | manila-share     | T-kolla-node01@pacific_backend      | nova | enabled | up    |
2024-08-20T08:18:28.000000 |
```

----End

5.3 Verifying the Manila Share Service

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** is the value of **share_backend_name** 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 Create a share that supports the preceding properties.

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

Step 6 Run the **manila list | grep test001** command to check the share status. If the status is **available**, Huawei back-end services of Manila Share are normal.

----End

6 Upgrading Manila Driver

To upgrade OpenStack Manila Driver, perform the operations described in this chapter. If you install Manila Driver for the first time, skip this chapter.

Procedure

Step 1 Search for the original Huawei Manila Driver code directory in the system.

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 Back up all Huawei Manila Driver code files in the directory returned in **Step 1**.

The backup command is as follows:

```
cp -rf Source directory  Backup directory
```

NOTE

- *Source directory*: directory returned in **Step 1**.
- *Backup directory*: directory to which data is backed up.

Step 3 Copy all Huawei Manila Driver code files of the corresponding version to the directory returned in **Step 1**.

Step 4 Run the following command to restart the Manila Share service.

```
systemctl restart openstack-manila-share.service
```

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

----End

7 (Optional) Configuring Advanced 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: <>&'"

7.1 Creating a Share in a Specified Storage Pool

7.2 Creating a Share Using a Specified ACL Policy Type

7.3 Configuring One-Way Authentication

7.4 Configuring Two-Way Authentication

7.1 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. In addition, **False** indicates that the **driver_handles_share_servers** function is disabled. The value can only be **False**.

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
```

- To configure multiple target storage pools, run the following command:

```
root@ubuntu:~# manila type-key target_pool_type set pool_name="<or> StoragePool001 <or>
StoragePool002"
```

NOTE

- The storage pool name instead of the storage pool ID is specified.
- The storage pool name can be queried by running the **manila pool-list --detail** command.

```
[root@T-kolla-node01 manila-share]# manila pool-list --detail
+-----------------------------------+------------------------------------+
| Property                          | Value                              |
+-----------------------------------+------------------------------------+
| pool_name                         | pool                               |
| qos                               | True                               |
| create_share_from_snapshot_support | False                             |
| timestamp                         | 2024-08-20T07:19:10.401718         |
| pool_id                           | 0                                  |
| driver_version                    | 2.7.2                              |
| total_capacity_gb                 | 8845.0                             |
| revert_to_snapshot_support        | True
```

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

7.2 Creating a Share Using a Specified ACL Policy Type

This section describes how to create a share using an ACL policy type.

Procedure

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

In the preceding command, **acl_policy_type** indicates the share type name, which is specified by users. In addition, **False** indicates that the **driver_handles_share_servers** function is disabled. The value can only be **False**.

Step 2 Run the following command to configure the ACL policy type. For details about the parameter values, see **Table 7-1**.

```
root@ubuntu:~# manila type-key acl_policy_type set acl_policy='<is> 1'
```

Table 7-1 ACL policy type parameter

Parameter	Description	Value	Remarks
acl_policy	ACL policy type, which corresponds to the security style of a namespace.	0	**Mixed**: applies to scenarios where users of CIFS clients (using the SMB protocol) and UNIX clients (using the NFS, HDFS, or DPC protocol) can access and control namespaces. In this style, CIFS permissions (NT ACLs) and UNIX permissions (UNIX mode bits, POSIX ACLs, or NFSv4 ACLs) cannot co-exist. Only the latest configured permissions on either CIFS or UNIX clients take effect.
		1	**UNIX**: applies to scenarios where UNIX mode bits, POSIX ACLs, or NFSv4 ACLs control users' permissions.
		2	**NTFS**: applies to scenarios where Windows NT ACLs control users' permissions.
		3	**Native**: applies to the same scenarios as the **Mixed** style. The difference with **Mixed** is that in the **Native** style, CIFS permissions (NT ACLs) and UNIX permissions (UNIX mode bits, POSIX ACLs, or NFSv4 ACLs) can co-exist, and will neither affect nor synchronize with each other.

Step 3 Run the **manila create --name test001 NFS 5 --share-type acl_policy_type** command to create a share that supports the preceding properties.

----End

7.3 Configuring One-Way 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 Manila node through the management IP address.

Step 2 Set the configuration file of Huawei Cinder Driver. Add **SSLCertVerify** and **SSLCertPath** in the **<Storage>** section.

```
<?xml version='1.0' encoding='UTF-8'?>
<config>
  <Storage>
    ...
    <SslCertVerify>False</SslCertVerify>
    <SslCertPath>******</SslCertPath>
  </Storage>
  ...
</config>
```

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**.

Step 3 Run the following command to restart the Manila Share service.

```
systemctl restart openstack-manila-share.service
```

Step 4 Run the **manila service-list** command. If the value of **State** is **up**, the Manila Share service is successfully started.

----End

7.4 Configuring Two-Way Authentication

This section describes how to use the plug-in to configure the two-way certificate. After the configuration is complete, two-way authentication can be performed between the plug-in and storage.

NOTE

This section uses non-containerized deployment as an example. For containerized deployment, you need to place the certificate in the container for authentication.

Prerequisites

- You have obtained the two-way certificate file.
- You have logged in to a storage backend and enabled two-way certificate authentication.

NOTICE

After two-way certificate authentication is enabled for the storage backend, Manila must use two-way authentication.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to an OpenStack Manila node through the management IP address.

Step 2 Save the obtained certificate to a directory on the host. The requirements for the directory and file permission are as follows:

- Directory: can be accessed by Manila, for example, **/etc/manila/**.
- File permission: 600 or higher. The owner group is **manila**.

Step 3 Add the following information to the **/etc/manila/manila.conf** file added in **5.1 Configuring Manila Driver in Non-Containerized Mode**. **Table 7-2** describes the parameters.

```
storage_ssl_two_way_auth = True
storage_cert_filepath=*****
storage_ca_filepath=*****
storage_key_filepath=******
storage_key_pwd=******
```

Table 7-2 Two-way authentication parameters

Parameter	Optional/ Mandatory/ Conditionally Mandatory	Description
storage_ssl_two_way_auth	Optional	Whether two-way authentication is supported. The value can be: ● **True**: Two-way authentication is used. ● **False**: Two-way authentication is not used. The default value is **False**.
storage_ca_filepath	Conditionally mandatory	Absolute path of the CA certificate on the server (storage). ● When the value of **storage_ssl_two_way_auth** is **True**: Enter the path of the CA certificate on the server (storage). ● When the value of **storage_ssl_two_way_auth** is **False**: This parameter is optional. Example: storage_ca_filepath=/etc/manila/storage_ca.crt

Parameter	Optional/ Mandatory/ Conditionally Mandatory	Description
storage_cert_filepath	Conditionally mandatory	Absolute path of the certificate on the client. ● When the value of **storage_ssl_two_way_auth** is **True**: Enter the path of the certificate on the client. ● When the value of **storage_ssl_two_way_auth** is **False**: This parameter is optional. Example: storage_cert_filepath=/etc/manila/client.crt
storage_key_filepath	Conditionally mandatory	Absolute path of the private key of the certificate on the client. ● When the value of **storage_ssl_two_way_auth** is **True**: Enter the path of the private key of the certificate on the client. ● When the value of **storage_ssl_two_way_auth** is **False**: This parameter is optional. Example: storage_key_filepath=/etc/manila/client.key
storage_key_pwd	Optional	Password of the private key of the certificate on the client. ● When the value of **storage_ssl_two_way_auth** is **True** and **storage_key_file** is encrypted: Enter the password of the private key of the certificate on the client. ● When the value of **storage_ssl_two_way_auth** is **False**: This parameter is optional. Example: storage_key_pwd=xxxxxx

Step 4 Run the following command to restart the Manila Share service.

```
systemctl restart openstack-manila-share.service
```

Step 5 Run the **manila service-list** command. If the value of **State** is **up**, the Manila Share service is successfully started.

----End

8 Appendix

8.1 Security Hardening

8.1 Security Hardening

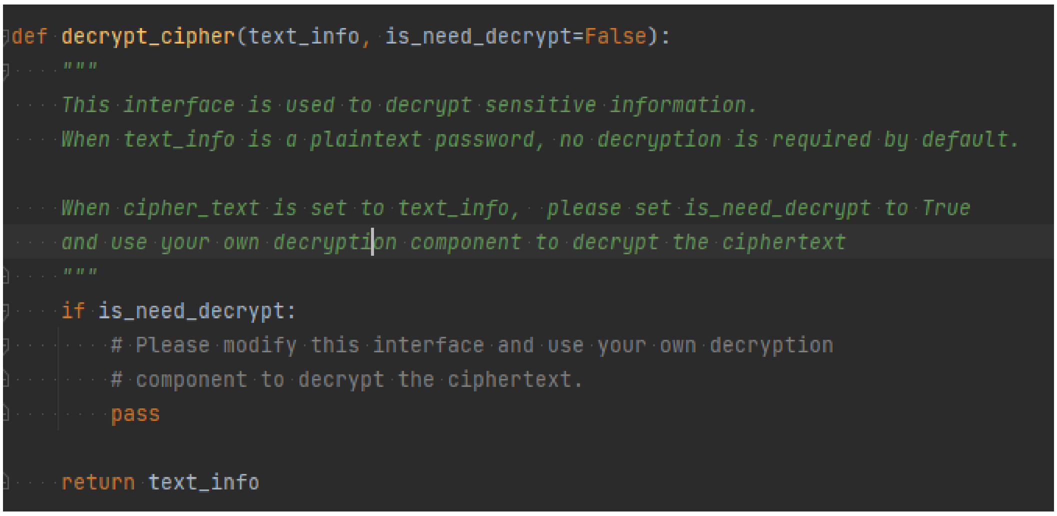
NOTE

The OpenStack platform does not have a good solution to encrypting and decrypting plaintext passwords in configuration files. Therefore, in open-source scenarios, the driver stores plaintext passwords in configuration files.

The driver is released as Python source code and provides a security hardening entry for users to perform secondary development to meet their security requirements.

The secondary development entry code is as follows:

Find the **cipher.py** file in driver code directory **oceanstorPacific/utils** and modify the following code in the file.



```
def decrypt_cipher(text_info, is_need_decrypt=False):
    """
    This interface is used to decrypt sensitive information.
    When text_info is a plaintext password, no decryption is required by default.

    When cipher_text is set to text_info,  please set is_need_decrypt to True
    and use your own decryption component to decrypt the ciphertext
    """
    if is_need_decrypt:
        # Please modify this interface and use your own decryption
        # component to decrypt the ciphertext.
        pass

    return text_info
```

NOTE

When configuring a password upon code modification, configure the encrypted ciphertext to the specified position. The driver will call the **decrypt_cipher** method in the corresponding position to decrypt the ciphertext.

The following configuration item involves ciphertext configuration:

- **UserPassword** configuration item in the .xml configuration file (for example, **manila_driver.xml**) of the driver
- **storage_key_pwd** (password of the private key file of the certificate on the client) in the **manila.conf** configuration file