

eSDK Cloud Storage Plugins
25.2.0

OpenStack Manila Driver 配置指南

文档版本 01
发布日期 2025-09-30



版权所有 © 华为技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址：深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址：<https://e.huawei.com>

安全声明

产品生命周期政策

华为公司对产品生命周期的规定以“产品生命周期终止政策”为准，该政策的详细内容请参见如下网址：
<https://support.huawei.com/ecolumnsweb/zh/warranty-policy>

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：
<https://www.huawei.com/cn/psirt/vul-response-process>
如企业客户须获取漏洞信息，请参见如下网址：
<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

华为初始证书权责说明

华为公司对随设备出厂的初始数字证书，发布了“华为设备初始数字证书权责说明”，该说明的详细内容请参见如下网址：
<https://support.huawei.com/enterprise/zh/bulletins-service/ENEWS2000015766>

华为企业业务最终用户许可协议(EULA)

本最终用户许可协议是最终用户（个人、公司或其他任何实体）与华为公司就华为软件的使用所缔结的协议。最终用户对华为软件的使用受本协议约束，该协议的详细内容请参见如下网址：
<https://e.huawei.com/cn/about/eula>

产品资料生命周期策略

华为公司针对随产品版本发布的售后客户资料（产品资料），发布了“产品资料生命周期策略”，该策略的详细内容请参见如下网址：
<https://support.huawei.com/enterprise/zh/bulletins-website/ENEWS2000017760>





前言

读者对象

- 本文档主要适用于以下读者对象：
- 技术支持工程师
 - 运维工程师
 - 具备存储和OpenStack基础知识的工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
 警告	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
 注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。
须知	用于传递设备或环境安全警示信息。如不避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
 说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修改记录

文档版本	发布日期	修改说明
01	2025-09-30	第一次正式发布。

目 录

前言..... iii

1 简介..... 1

2 版本配套关系..... 2

3 规格与限制..... 4

4 Manila Driver 安装与部署..... 5

4.1 Manila Driver 获取..... 5

4.2 非容器化 Manila Share 服务华为 Manila Driver 部署..... 5

4.3 容器化 Manila Share 服务华为 Manila Driver 部署..... 6

4.4 容器化场景 Manila Share 服务华为 Manila Driver 部署（中国移动苏研）..... 7

5 Manila Driver 基本属性配置..... 12

5.1 非容器化场景配置 Manila Driver..... 12

5.2 容器化场景配置 Manila Driver..... 15

5.3 验证 Manila Share 服务..... 16

6 Manila Driver 升级..... 17

7（可选）Manila Driver 高级属性配置..... 18

7.1 指定存储池创建共享..... 18

7.2 指定 ACL 策略类型创建共享..... 19

7.3 配置单向认证..... 20

7.4 配置双向认证..... 20

8 附录..... 24

8.1 安全加固..... 24

1 简介

华为Manila Driver是部署在OpenStack Manila模块上的一个插件，用于对接华为存储向OpenStack平台提供共享存储能力。

2 版本配套关系

介绍华为Manila Driver与华为存储、OpenStack的版本配套关系。

表 2-1 Manila Driver 存储产品版本支持说明

OpenStack版本	存储产品版本
Train	OceanStor Pacific 8.1.5/8.2.0/8.2.1/ V800R001C10

表 2-2 Manila Driver 特性支持说明（√：支持，x：不支持）

特性	NFS Share	CIFS Share	备注
OpenStack	Train	Train	N/A
Create Share	√	√	N/A
Delete Share	√	√	N/A
Allow access	√	√	支持对IPv4/IPv6地址和地址段进行授权
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
Reload QoS	√	√	中兴云平台定制特性

 说明

对非OpenStack社区标准的特性进行说明如下：

- **Reload QoS**：支持对share添加、删除、更新QoS策略，限制如下：
 - QoS参数只支持如下字段: ["maxIOPS", "maxMBPS", "total_ops_sec", "total_iops_sec", "total_bytes_sec"]。
 - maxIOPS、total_ops_sec、total_iops_sec同时设定时，只有一个生效，优先级为: maxIOPS > total_ops_sec > total_iops_sec。
 - maxMBPS和total_bytes_sec同时设定时以maxMBPS为准。
 - maxIOPS、total_ops_sec、total_iops_sec必须设置其中一个；maxMBPS和total_bytes_sec必须设置其中一个。
 - 具体参数的取值范围详见[表2-3](#)。

表 2-3 QoS 参数详解

参数名称	说明	备注	示例
maxIOPS	最大 IOPS。	有效值为0-1073741824000的整数或字符类型的整数，0表示无限制	<ul style="list-style-type: none">● 1000000● '1000000'
maxMBPS	最大带宽。	单位MB/s，有效值为0-1073741824的整数或字符类型的整数，0表示无限制	<ul style="list-style-type: none">● 30000● '30000'
total_iops_sec	最大 IOPS。	有效值为0-1073741824000的整数或字符类型的整数，0表示无限制	<ul style="list-style-type: none">● 1000000● '1000000'
total_bytes_sec	最大带宽。	单位byte/s，有效值为0-1125899906842624的整数或字符类型的整数，0表示无限制	<ul style="list-style-type: none">● 3145728000● '3145728000'
total_ops_sec	最大 IOPS。	有效值为0-1073741824000的整数或字符类型的整数，0表示无限制	<ul style="list-style-type: none">● 1000000● '1000000'

3 规格与限制

主特性	子特性	描述	备注
平台配套	原生OpenStack平台	OpenStack版本：Train	-
配置	XML配置文件	Manila Driver配置使用的存储池，需要事先保证在华为存储上存在，否则需要手动创建。	华为Manila Driver配置文件所有参数值中不能含有XML特殊字符< > & ' "。
存储设备	存储池	支持对接多个存储池。	-
规格	share配额	受限于OceanStor Pacific存储限制，最大支持100,000个的share的创建。	-
Driver后端	同一套存储支持对接的Driver后端数量	针对同一套存储，一个manila-share节点对接一个后端，最多支持三个manila-share节点。	单个存储可对接的Driver后端数量与存储管理平面支持的最大并发请求相关。对接的Driver后端过多，会导致存储过载。

4 Manila Driver 安装与部署

4.1 Manila Driver获取

4.2 非容器化Manila Share服务华为Manila Driver部署

4.3 容器化Manila Share服务华为Manila Driver部署

4.4 容器化场景Manila Share服务华为Manila Driver部署（中国移动苏研）

4.1 Manila Driver 获取

操作步骤

- 步骤1 打开浏览器，访问仓库地址：https://github.com/Huawei/FusionStorage_OpenStack_Driver
 - 步骤2 选择“Code > Download ZIP”，华为Manila Driver将以压缩包的形式下载到本地。
 - 步骤3 解压该压缩包。
 - 步骤4 在解压出的目录下找到Manila目录。
- 结束

4.2 非容器化 Manila Share 服务华为 Manila Driver 部署

操作步骤

OpenStack社区标准部署步骤如下：

- 步骤1 查找系统中原有的华为Manila Driver代码目录。

完整路径在不同的系统下可能有所不同，可通过以下方法搜索华为Manila Driver代码安装目录：

- 执行以下命令，回显中的“/usr/lib/python2.7/site-packages/manila/share/drivers/huawei”即为代码安装目录。

```
# python -c "from manila.share.drivers import huawei; print (huawei.__path__)"  
['/usr/lib/python2.7/site-packages/manila/share/drivers/huawei']
```

- 执行以下命令，回显中的“/usr/lib/python3.7/site-packages/manila/share/drivers/huawei”即为代码安装目录。

```
# python3 -c "from manila.share.drivers import huawei; print (huawei.__path__)"  
['/usr/lib/python3.7/site-packages/manila/share/drivers/huawei']
```

步骤2 将从[4.1 Manila Driver获取](#)获取到的OpenStack Manila Driver的huawei目录下的所有内容拷贝到Manila Share节点华为Manila Driver代码安装目录下，默认路径参考[步骤1](#)，复制后结果如下。Driver所有目录的权限最小为755，所有文件的权限最小为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
```

----结束

4.3 容器化 Manila Share 服务华为 Manila Driver 部署

操作步骤

OpenStack社区标准部署步骤如下：

步骤1 登录具有Manila Share容器镜像的节点，执行**docker image ls**命令查询Manila Share容器镜像。

```
centos7:4.1700 31200035e02c 3 years ago 1.97GB  
[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
```

步骤2 执行**docker save -o manila_share.tar kolla/centos-source-manila-share**命令备份Manila Share容器镜像到镜像文件。

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

说明

manila_share.tar为自定义备份镜像的名称，**kolla/centos-source-manila-share**为Manila Share容器镜像名称。

步骤3 在任意目录下创建临时目录**huawei**。

```
# mkdir huawei
```

步骤4 将获取到的对应版本的华为Manila Driver代码文件拷贝到创建的**huawei**目录下。Manila Driver目录的权限最小为755，所有文件的权限最小为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
```

步骤5 执行**docker ps -a | grep manila**命令列出Manila Share容器，**manila_share**为容器名称。

```
exit  
[root@T-kolla-node01 oceanstor_hp1]# docker ps -a | grep manila  
17726a632522 kolla/centos-source-manila-share:train "dumb-init --single-" 3 years ago Up 35 minutes  
92279b863194 kolla/centos-source-manila-scheduler:train "dumb-init --single-" 3 years ago Up 5 days  
c8baef59305b kolla/centos-source-manila-data:train "dumb-init --single-" 3 years ago Up 5 days  
7f93e43b0173 kolla/centos-source-manila-api:train "dumb-init --single-" 3 years ago Up 5 days  
manila_share  
manila_scheduler  
manila_data  
manila_api
```

步骤6 执行**docker exec -it -u 0 manila_share bash**命令进入**manila_share**容器。

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

步骤7 执行`python -c "from manila.share import drivers; print (drivers.__path__)"`命令查询原有的华为Manila Driver代码的路径。

```
(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']
```

说明

- `manila_share`为步骤5查询出来的容器名称。
- `/var/lib/kolla/venv/lib/python2.7/site-packages/manila/share/drivers`为原有的华为Manila Driver代码的路径。
- 如果执行上述命令报错，请根据Python环境变量，变更命令为`python3 -c "from manila.share import drivers; print (drivers.__path__)"`

步骤8 执行`exit`命令，退出`manila_share`容器，进入步骤3创建的`huawei`目录的上级目录，将`huawei`目录拷贝到步骤7中原有的华为Manila Driver代码的路径。

```
[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
```

步骤9 执行以下命令，重启Manila Share服务，`manila_share`为步骤5获取到的容器名称。

```
docker restart manila_share
```

等待一段时间后（一般30秒左右），执行`manila service-list`命令。有以下回显时，查看`State`的值为`up`，表示Manila Share服务启动成功。

```
[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 |
+-----+-----+-----+-----+-----+-----+-----+
2024-08-20T08:18:28.000000 |
```

----结束

4.4 容器化场景 Manila Share 服务华为 Manila Driver 部署（中国移动苏研）

步骤1 使用远程访问工具（以PuTTY为例），通过管理IP地址，登录要部署Manila Share的节点。

步骤2 进入`/apps`目录，将4.1 Manila Driver获取中的`file_driver_dir`目录上传至`/apps`目录下。

```
[root@bcecn03 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
```

步骤3 执行`cd file_driver_dir`命令，进入“file_driver_dir”目录，执行`chmod +x file_driver_init.sh`命令，赋予file_driver_init.sh脚本可执行权限。

```
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.oel.x86_64.rpm
-rw-r--r-- 1 root root 1.2M May 5 14:12 python2-xml-4.5.2-3.oel.x86_64.rpm
```

步骤4 使用远程访问工具（以PuTTY为例），通过管理IP地址，登录中国移动苏研K8s管理的节点，其上有bcec-workspace相关部署脚本（此脚本为中国移动苏研环境特有），假设其Manila工具目录为：`/root/bcec-workspace/2017.4.7.6/bcec-workspace/manila`。进入中国移动苏研提供的默认yaml文件路径，假设为：`/root/bcec-workspace/2017.4.7.6/bcec-workspace/manila/deploy/manila-share.yaml`

步骤5 对该yaml文件进行修改，对containers下volumeMounts下的字段，进行如下修改：

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

添加黑体部分

containers:

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

步骤6 对该yaml文件进行修改，对containers下volumes下的字段，进行如下修改：

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

添加黑体部分

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
```

步骤7 对该yaml文件进行修改，对containers下volumeMounts下的字段，进行如下修改：

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

添加黑体部分

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

步骤8 对该yaml文件进行修改，对containers下volumes下的字段，进行如下修改：

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

添加黑体部分

```
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
```

步骤9 对该yaml文件进行修改，对containers下volumeMounts下的字段，进行如下修改：

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

添加黑体部分

```
containers:
  .....
  volumeMounts:
    .....
    - mountPath: /file_driver_dir
      name: file-driver-dir
```

步骤10 对该yaml文件进行修改，对containers下volumes下的字段，进行如下修改：

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

添加黑体部分

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

步骤11 使用远程访问工具（以PuTTY为例），通过管理IP地址，登录中国移动苏研K8s管理的节点，其上有bcec-workspace相关部署脚本（此脚本为中国移动苏研环境特有），假设其Manila工具目录为：`/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
```

步骤12 执行`cd /root/bcec-workspace/2017.4.7.6/bcec-workspace/manila/conf`命令，修改`manila-share.conf`配置文件，在文件最后添加如下后端配置，该文件在bcec-workspace脚本工具（中国移动苏研环境自带）路径下，以`/root/bcec-workspace/2017.4.7.6/bcec-workspace/manila/conf`为例。

```
[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
```

说明

- 后端名`huawei_backend`可自定义。
- `share_driver`表示待使用的Manila Driver类型，此处填写`manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver`。
- `manila_huawei_conf_file`表示配置的华为Manila Driver配置文件路径。
- `driver_handles_share_servers`为`True`时创建share需要指定share-network。

步骤13 在[DEFAULT]区块中修改`enabled_share_backends`配置项，新增`huawei_backend`后端。

```
[DEFAULT]
...
enabled_share_backends=xxx,huawei_backend
```

步骤14 在`/root/bcec-workspace/2017.4.7.6/bcec-workspace/manila/conf`目录下配置`manila_driver.xml`，配置参数参考[manila_driver.xml配置](#)。

步骤15 修改、执行`create_volume.sh`。

```
function create_common_volumes()
/app/bin/kubectl create configmap cm-ec-manila-control-conf --from-file=$work_path/./conf/manila-control-conf --dry-run -o yaml | /app/bin/kubectl -n bcec apply -f -
/app/bin/kubectl create configmap cm-ec-manila-share-conf --from-file=$work_path/./conf/manila-share-conf --dry-run -o yaml | /app/bin/kubectl -n bcec apply -f -
/app/bin/kubectl create configmap cm-ec-manila-graceful-shutdown-sh --from-file=$work_path/./tools/manila_mq_operation.sh --dry-run -o yaml | /app/bin/kubectl -n bcec apply -f -
/app/bin/kubectl create configmap cm-ec-manila-check-graceful-shutdown-sh --from-file=$work_path/./tools/manila_mq_confirm.sh --dry-run -o yaml | /app/bin/kubectl -n bcec apply -f -
/app/bin/kubectl create configmap cm-ec-manila-mq-healthy-sh --from-file=$work_path/./tools/manila_mq_healthy.sh --dry-run -o yaml | /app/bin/kubectl -n bcec apply -f -
/app/bin/kubectl create configmap cm-ec-manila-share-start-sh --from-file=$work_path/./tools/manila_share_start.sh --dry-run -o yaml | /app/bin/kubectl -n bcec apply -f -
/app/bin/kubectl create configmap cm-ec-manila-driver-xml --from-file=$work_path/./conf/manila_driver.xml --dry-run -o yaml | /app/bin/kubectl -n bcec apply -f -
/app/bin/kubectl create configmap cm-ec-admin-openrc-sh --from-file=$work_path/./admin-openrc.sh --dry-run -o yaml | /app/bin/kubectl -n bcec apply -f -

```

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

函数`create_common_volumes`中添加`configmap`后保存：

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

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

步骤16 运行`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
```

步骤17 执行如下命令完成容器的启动。

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

步骤18 如果使用的命名空间为bcec，执行如下命令检查Manila Share容器是否启动成功。

```
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]#
```

说明

- 返回状态为Running，且READY为1/1表示启动成功。
- 如果Manila Share容器之前已经启动，会先停止已经启动的Manila Share容器，停止时间大概需要2分钟。

----结束

5 Manila Driver 基本属性配置

介绍如何配置华为Manila Driver。

须知

- OpenStack Ocata Manila创建共享时，如未指定share type，会使用“manila.conf”文件中的“default_share_type”的配置值。
- Manila Driver配置使用的存储池，需要事先保证在华为存储上存在，否则需要手动创建。
- 华为Manila Driver配置文件所有参数值中不能含有XML特殊字符< > & ' "。

5.1 非容器化场景配置 Manila Driver

5.2 容器化场景配置 Manila Driver

5.3 验证Manila Share服务

5.1 非容器化场景配置 Manila Driver

操作步骤

步骤1 在“/etc/manila”路径下创建华为Manila Driver配置文件，文件格式为xml。配置文件名称可自定义，例如“manila_huawei_conf.xml”。

步骤2 编辑“manila_huawei_conf.xml”文件，配置华为Manila Driver配置文件参数。

```
<?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>
```

表 5-1 参数说明

参数	说明	示例
Product	必填参数，存储产品类型，当前仅支持如下配置： <ul style="list-style-type: none">• Pacific	Pacific
RestURL	必填参数，REST接口访问地址。	此处填写OceanStor Pacific存储浮动IP地址和端口
UserName	必填参数，存储管理员角色的用户。	manilauser 须知 请避免直接使用“超级管理员”权限账户。
UserPassword	必填参数，存储管理员角色的用户密码。	-
Reserved_percentage	选填参数，空间预留百分比，不填写默认为15，仅支持数字。 用户可通过修改该参数来实现容量预留。	15
Max_over_subscription_ratio	选填参数，最大超分比，不填写默认为1，仅支持大于等于1的浮点数。 用户可通过修改该参数来实现容量超分配。	1
Semaphore	选填参数，用于控制调用存储RESTful接口的并发量，不配置时默认为20，最大不能超过20	20
StoragePool	必填参数，存储池的ID，仅支持数字，支持配置多个，使用“;”隔开。	-
AccountName	必填参数，存储租户名称，建议使用系统租户system	system
ClusterDomainName	必填参数，用户挂载NFS/CIFS共享的业务IP或域名 说明 创建share时，Driver会将该参数设置到share的export_locations中	-

参数	说明	示例
RollbackRate	选填参数，用于设置快照回滚的回滚速率，不配置时默认为2，仅支持四个等级： [1, 2, 3, 4]	2

步骤3 修改“/etc/manila/manila_huawei_conf.xml”文件的权限，使其与“/etc/manila/manila.conf”文件的拥有者及用户组一致。

```
# 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
...
```

步骤4 配置“/etc/manila/manila.conf”文件，在文件最后添加如下后端配置。配置参数请参考表5-2。

表 5-2 参数说明

参数	说明	示例
share_driver	必填参数，表示对接的华为后端存储的driver类型，用于启动服务时找到对应的华为存储后端驱动。	固定填写:manila.share.drivers.huawei.oceanstorPacific.oceanstorpacific_nas.HuaweiNasDriver
manila_huawei_conf_file	必填参数，5.1 非容器化场景配置Manila Driver配置的华为Manila Driver配置文件路径。	/etc/manila/manila_huawei_conf.xml
share_backend_name	必填参数，share后端名称，用于后续在创建share_type时指定share_backend_name的值。	huawei_backend
driver_handles_share_servers	必填参数，是否开启OpenStack租户网络隔离。当前仅支持设置为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
```

步骤5 在[DEFAULT]区块中修改enabled_share_backends配置项，新增huawei_backend后端。

```
[DEFAULT]
...
enabled_share_backends=xxx,huawei_backend
```

步骤6 执行以下命令，重新启动Manila服务。

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

- 步骤7** 等待一段时间后（一般30秒左右），执行如下命令。查看对应的华为后端share服务State的值为up，表示Manila Share服务启动成功。

```
# manila service-list
```

----结束

5.2 容器化场景配置 Manila Driver

操作步骤

- 步骤1** 使用远程访问工具（以PuTTY为例），通过管理IP地址，登录OpenStack的Manila Share节点。在主机上执行如下命令查找manila.conf，并按照步骤4和步骤5配置相应参数。

```
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
/usr/lib/docker/overlay2/01414b5fd42277e120f3272f320b756d3e1e3d30c90b80fb90f300a3e70b8a/
```

- 步骤2** 执行命令docker ps -a | grep manila查询Manila Share容器，然后执行命令docker exec -it -u 0 manila_share bash进入Manila Share容器。

```
centos
[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 /]#
```

说明

manila_share为查询出来的Manila Share容器名称。

- 步骤3** 在Manila Share容器中“/etc/manila”路径下创建华为Manila Driver配置文件，文件格式为xml，并按照步骤2配置相应参数。配置文件名称可自定义，例如“manila_huawei_conf.xml”。

- 步骤4** 修改华为Manila Driver配置文件的拥有者及用户组与“/etc/manila/manila.conf”文件的拥有者及用户组一致。

```
-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
```

- 步骤5** 执行以下命令，重启Manila Share服务。

```
docker restart manila_share
```

等待一段时间后（一般30秒左右），执行manila service-list命令。有以下回显时，查看State的值为up，表示Manila Share服务启动成功。

```
[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 |
```

----结束

5.3 验证 Manila Share 服务

操作步骤

步骤1 创建通用share类型。

其中，general为share类型名称，由用户指定。False表示不启用
driver_handles_share_servers功能，若需启用此功能，该参数配置为True。
manila type-create general False

步骤2 设置share类型后端。其中，huawei_backend为[步骤4](#)中配置的share_backend_name值。

```
# manila type-key general set share_backend_name=huawei_backend
```

步骤3 设置快照属性。

```
# manila type-key general set snapshot_support=True
```

步骤4 设置快照回滚属性。

```
# manila type-key general set revert_to_snapshot_support=True
```

步骤5 创建支持以上属性的share。

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

步骤6 执行manila list | grep test001查看share的状态为available则表明Manila Share的华为后端服务正常。

----结束

6 Manila Driver 升级

OpenStack Manila Driver升级场景，请参考该章节；如果是第一次安装部署Manila Driver，请跳过该章节。

操作步骤

步骤1 查找系统中原有的华为Manila Driver代码目录。

完整路径在不同的系统下可能有所不同，可通过以下方法搜索华为Manila Driver代码目录：

- 执行以下命令，回显中的“/usr/lib/python2.7/site-packages/manila/share/drivers/huawei”即为代码目录。

```
# python -c "from manila.share.drivers import huawei; print (huawei.__path__)"  
['/usr/lib/python2.7/site-packages/manila/share/drivers/huawei']
```
- 执行以下命令，回显中的“/usr/lib/python3.7/site-packages/manila/share/drivers/huawei”即为代码目录。

```
# python3 -c "from manila.share.drivers import huawei; print (huawei.__path__)"  
['/usr/lib/python3.7/site-packages/manila/share/drivers/huawei']
```

步骤2 备份**步骤1**中返回目录下所有的华为Manila Driver代码文件。

备份命令参考如下：

```
cp -rf 源目录 备份目录
```

说明

- 源目录：**步骤1**中返回的目录。
- 备份目录：用户希望备份到的目录。

步骤3 将对应版本的华为Manila Driver代码文件全部拷贝到**步骤1**中返回的目录中。

步骤4 执行如下命令，重启Manila Share服务。

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

等待一段时间后（一般30秒左右），执行**manila service-list**命令。查看对应的华为后端share服务**State**的值为**up**，表示Manila Share服务启动成功。

----结束

7（可选）Manila Driver 高级属性配置

介绍如何配置华为Manila Driver。

须知

- OpenStack Ocata Manila创建共享时，如未指定share type，会使用“/etc/manila/manila.conf”文件中的默认share type。
- Manila Driver配置使用的存储池，需要事先保证在华为存储上存在，否则需要手动创建，存储池类型必须是“文件存储服务”，且存储池名不能包含中文。
- 华为Manila Driver配置文件所有参数值中不能含有XML特殊字符< > & ' "。

7.1 指定存储池创建共享

7.2 指定ACL策略类型创建共享

7.3 配置单向认证

7.4 配置双向认证

7.1 指定存储池创建共享

本节介绍配置指定存储池创建共享的操作步骤。

操作步骤

步骤1 执行**manila type-create target_pool_type False**命令，创建share类型。

其中，target_pool_type为share类型名称，由用户指定。False表示不启用driver_handles_share_servers功能，此处只支持设置为False

步骤2 执行以下命令，配置目标存储池。

- 若配置单个目标存储池，执行以下命令：
root@ubuntu:~# manila type-key target_pool_type set pool_name=StoragePool001
- 若配置多个目标存储池，执行以下命令：
root@ubuntu:~# manila type-key target_pool_type set pool_name="<or> StoragePool001 <or> StoragePool002"

 说明

- 此处指定的存储池为存储池名称，不是存储池ID
- 存储池名称可通过manila pool-list --detail查询

```
[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 |
```

步骤3 执行**manila create --name test001 NFS 2 --share-type target_pool_type**命令，创建支持以上属性的share。

----结束

7.2 指定 ACL 策略类型创建共享

本节介绍配置ACL策略类型创建共享的操作步骤。

操作步骤

步骤1 执行**manila type-create acl_policy_type False**命令，创建share类型。

其中，**acl_policy_type**为share类型名称，由用户指定。False表示不启用**driver_handles_share_servers**功能，此处只支持设置为False

步骤2 执行以下命令，配置想要指定的ACL策略类型, 参数配置值说明见[表7-1](#)。

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

表 7-1 ACL 策略类型参数说明

参数名称	说明	配置值	说明
acl_policy	ACL 策略类型, 对应命名空间的安全模式	0	“Mixed”类型: 适用于CIFS客户端（使用SMB协议）的用户和UNIX客户端(使用NFS/HDFS/DPC协议)的用户都可以访问并控制命名空间的场景。该模式下，权限以CIFS客户端或UNIX客户端最后设置的权限为准，不会同时存在CIFS权限（NT ACL）和UNIX权限（Unix Mode/Posix ACL/NFSv4 ACL）。
		1	“UNIX”类型：适用于Unix Mode/Posix ACL/NFSv4 ACL权限控制场景。
		2	“NTFS”类型：适用于Windows NT ACL权限控制的场景。

参数名称	说明	配置值	说明
		3	“Native”类型：与Mixed模式适用于相同的场景。与Mixed模式区别在于，Native模式可以同时存在CIFS权限（NT ACL）和UNIX权限（Unix Mode/Posix ACL/NFSv4 ACL），两者互不影响，不存在互相同步。

步骤3 执行**manila create --name test001 NFS 5 --share-type acl_policy_type**命令，创建支持以上属性的share。

----结束

7.3 配置单向认证

本节介绍配置通过证书认证方式连接后端存储。

操作步骤

步骤1 使用远程访问工具（以PuTTY为例），通过管理IP地址，登录OpenStack的Manila节点。

步骤2 配置华为Cinder Driver配置文件。在<Storage>配置段中增加**SSLCertVerify**和**SSLCertPath**配置。

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

说明

- **SSLCertVerify**表示是否开启证书认证。有效值为True或False，未配置时默认为False，建议开启证书校验。
- **SSLCertPath**指定用于认证的证书路径，仅在**SSLCertVerify**为True的情况下有效。

步骤3 执行以下命令，重启Manila Share服务。

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

步骤4 执行**manila service-list**命令。查看**State**的值为**up**，表示Manila Share服务启动成功。

----结束

7.4 配置双向认证

本节介绍如何使用插件配置双向证书。配置完毕，支持插件与存储进行双向认证。

 说明

本节操作以非容器化部署为示例，容器化部署需将证书放置到容器内进行配置认证。

前提条件

- 已获取双向证书的文件。
- 已登录存储后端并开启证书双向认证。

须知

存储开启证书双向认证后，Manila必须使用双向认证。

操作步骤

- 步骤1** 使用远程访问工具（以PuTTY为例），通过管理IP地址，登录OpenStack的Manila节点。
- 步骤2** 将已获取的证书放入到主机端的一个目录下，对目录与文件权限的要求如下：
- 目录：支持被Manila访问，例如“/etc/manila/”。
 - 文件权限：权限600及以上，属组均为manila。
- 步骤3** 在[5.1 非容器化场景配置Manila Driver](#)中创建后端时添加的“/etc/manila/manila.conf”文件中，补充如下信息，参数说明请参见[表7-2](#)。

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

表 7-2 双向认证参数说明

参数	可选/必选/条件必选	填写说明
storage_ssl_two_way_auth	可选	是否支持双向认证。可选值： <ul style="list-style-type: none">• True：表示使用双向认证。• False：表示不使用双向认证。 默认值：False。

参数	可选/必选/条件必选	填写说明
storage_ca_filepath	条件必选	<p>服务端（存储）CA证书的绝对路径。</p> <ul style="list-style-type: none"> 当“storage_ssl_two_way_auth”参数的值为“True”时： 填写服务端（存储）CA证书的路径。 当“storage_ssl_two_way_auth”参数的值为“False”时： 此参数为非必填项。 <p>示例如下： storage_ca_filepath=/etc/manila/ storage_ca.crt</p>
storage_cert_filepath	条件必选	<p>客户端的证书的绝对路径。</p> <ul style="list-style-type: none"> 当“storage_ssl_two_way_auth”参数的值为“True”时： 填写客户端的证书的路径。 当“storage_ssl_two_way_auth”参数的值为“False”时： 此参数为非必填项。 <p>示例如下： storage_cert_filepath=/etc/manila/ client.crt</p>
storage_key_filepath	条件必选	<p>客户端的证书的私有key的绝对路径。</p> <ul style="list-style-type: none"> 当“storage_ssl_two_way_auth”参数的值为“True”时： 填写客户端的证书的私有key的路径。 当“storage_ssl_two_way_auth”参数的值为“False”时： 此参数为非必填项。 <p>示例如下： storage_key_filepath=/etc/manila/ client.key</p>
storage_key_pwd	可选	<p>客户端的证书的私有key的密码。</p> <ul style="list-style-type: none"> 当“storage_ssl_two_way_auth”参数的值为“True”且 storage_key_file已加密时： 填写客户端的证书的私有key的密码。 当“storage_ssl_two_way_auth”参数的值为“False”时： 此参数为非必填项。 <p>示例如下： storage_key_pwd=xxxxxx</p>

步骤4 执行以下命令，重启Manila Share服务。

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

步骤5 执行**manila service-list**命令。查看**State**的值为**up**，表示Manila Share服务启动成功。

----**结束**

8 附录

8.1 安全加固

说明

由于OpenStack平台针对配置文件中的明文密码，当前没有良好的解决方案去实现加解密，因此，在开源场景下，驱动针对配置文件中的密码都是明文保存的。

驱动为Python源码发布，提供了安全加固的入口，可供用户基于该入口进行二次开发以满足自己的安全诉求。

二次开发入口代码如下：

找到驱动代码目录oceanstorPacific/utils下中的“cipher.py”文件，修改该文件中如下代码即可。

```
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
```

说明

代码修改后，配置密码时请直接将加密后的密文配置到指定位置，驱动会在对应位置调用代码中的"decrypt_cipher"方法实现密文解密。

涉及密文配置的配置项如下：

- 驱动xml配置文件（例如：manila_driver.xml）中的“UserPassword”配置项。
- manila.conf配置文件中客户端的证书私钥文件的密码“storage_key_pwd”配置项。