

Manila OpenStack service - How to share filesystem in OpenStack

- [Introduction](#)
- [How to create and use a CEPHFS_TYPE share](#)
 - [Request to be enabled to the service](#)
 - [Create the share](#)
 - [Mount the share on the VMs](#)

Introduction

This guide is for users that aim to create a **shared filesystem in their virtual machines** using the OpenStack Manila service.

The OpenStack Manila service allows the creation of a filesystem that can be shared among virtual machines in the same tenant (intra-tenant) or in different tenants (extra-tenant).

More details on Manila's service can be found [here](#).

Important: In ADA Cloud, it is enabled only the CEPHFS_TYPE share, that allows the creation of a share in the CEPH storage.

Important: Only intra-tenant shares are allowed on ADA Cloud.

How to create and use a CEPHFS_TYPE share

Request to be enabled to the service

A user that would like to make use of the Manila service needs to **send an email** to superc@cineca.it, communicating **how many shares** are needed and **for each share**:

- its **dimensions** (GB)
- the **instance name** of the virtual machines (VMs) that will share that filesystem
- the **tenant's name**.

Once enabled by the User Support Team, the user needs to create the share and mount it in a special network interface on the VMs attached to a dedicated storage network.

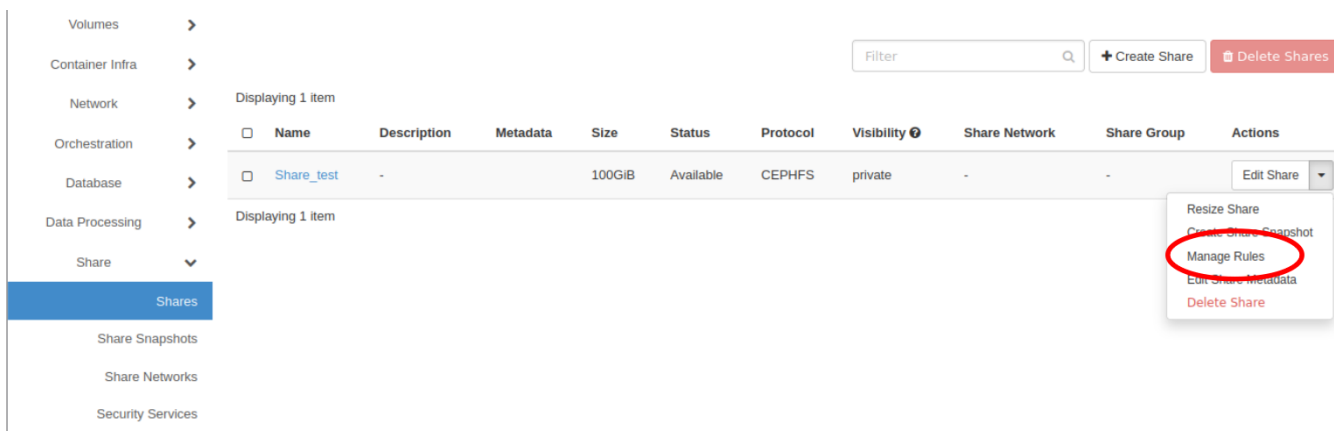
As an example, in what follows, we will create a share in common between VM_alice and VM_bob belonging to the same tenant.

Create the share

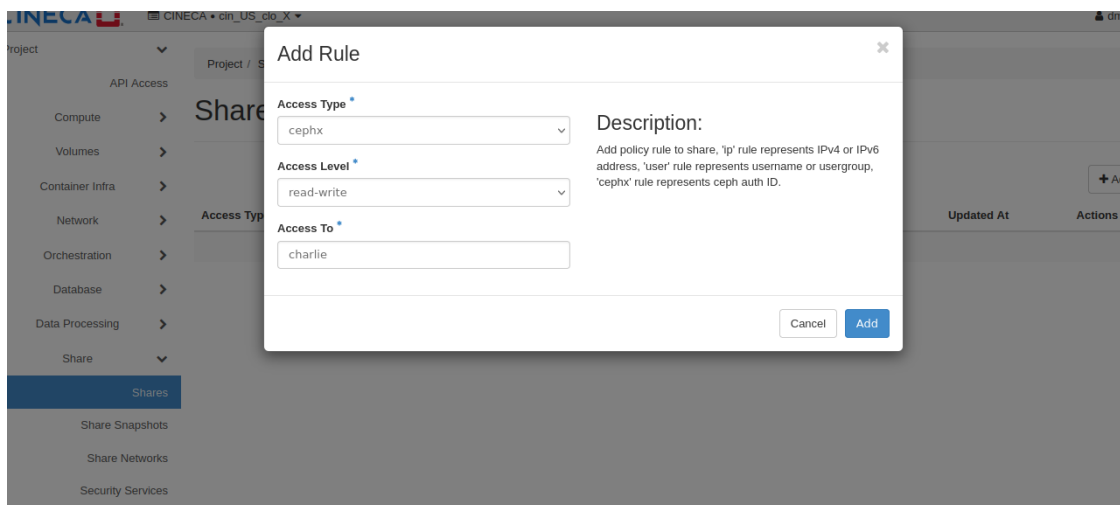
You can create a share by following the below steps:

- Create the share by clicking on "[Project](#) [Share](#) [Shares](#) **Create Share**" and set:
 - **share name**
 - **share protocol** == "**CephFS**" (other cases not allowed)
 - **size** (on the right side is visualized information about the actual available and used space within the tenant)
 - **Type** == "cephfs_type"
 - Leave blank the option "Make visible for all projects" because it is not enabled
 - In the end, click on the "**save**" button.

- Set the access rule on the share just created. On the OpenStack dashboard click on "[Project](#) [Share](#) [Shares](#)", then select the share just created and in the menu on the right select "**Manage Rules**".



- Click on **"Add rule"** and set:
 - **access type:** cephx
 - **access level:** read-write or read-only (depending on your needs)
 - **access to:** write the name of the client (in our example "charlie")



By clicking on the **"add"** button the dashboard will show the "access key" and "access to" keys that must be used to mount the share on the virtual machines.

Mount the share on the VMs

The user is now ready to mount the share on VMs. In the following example, we will consider two VM with Ubuntu 20.04 OS.

Please refer to the network guide of the operating system of your VM to be sure about the operations to be done.

Login in the first VM, then:

- configure the network interface attached to the storage network
 - **"ip a"** command lists all the network interfaces. Find the new interface, attached to the storage network, and refer to the mac-address of the interface to be sure.

Compute VM_bob Create Snapshot

Overview Instances Images Key Pairs Server Groups Volumes Container Infra Network

Overview Interfaces Log Console Action Log

Displaying 2 Items

Name	Network	Fixed IPs	MAC Address	Status	Admin State	Actions
(c6970aa1-91b5)	-	10.47.4.18	fa:16:3e:a4:b3:66	Active	UP	Edit Security Groups
(d9d545e8-d499)	tenant_admin_net	192.168.0.116	fa:16:3e:4f:fb:f1	Active	UP	Edit Security Groups

Displaying 2 Items

```

ubuntu@vm-bob:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc fq_codel state UP group default qlen 1000
    link/ether fa:16:3e:4f:fb:f1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.116/24 brd 192.168.0.255 scope global dynamic ens3
        valid_lft 83977sec preferred_lft 83977sec
    inet6 fe80::f816:3eff:fe4f:fbf1/64 scope link
        valid_lft forever preferred_lft forever
3: ens7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether fa:16:3e:a4:b3:66 brd ff:ff:ff:ff:ff:ff
  
```

- Create a new file in the /etc/netplan directory to configure such new interface (in our example "ens7"), and enable it

```

$ sudo su
$ cd /etc/netplan
$ cp <file-cloud-init> <file-ens-number> - example: cp 50-cloud-init.yaml ens7.yaml -

$ vim <file-ens-number> ==> Modify the value of the fields "ens", "mtu <value>", "macaddress <value>" and "set-
name <value>" with the values shown by "ip a" command.

-- example of the ens7.yaml is --

network:
  version: 2
  ethernets:
    ens7:
      dhcp4: true
      match:
        macaddress: <your MAC address>
        mtu: <MTU value>
      set-name: ens7

$ netplan apply ==> to enable the new interface
$ ip a ==> check that the interface is enabled
  
```

```

# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  version: 2
  ethernets:
    ens7:
      dhcp4: true
      match:
        macaddress: fa:16:3e:a4:b3:66
        mtu: 1500
      set-name: ens7
  
```

- Install the client, by installing the package named "ceph-common" (on Ubuntu)
- Create the mount point in the virtual machines (in our example "/mnt/share_manila") and mount the share

To mount the share you will need some informations contained in the "Share Overview" page on OpenStack dashboard, in particular you will need the values of PATH, ACCESS_TO and ACCESS KEY (here an example):

The screenshot shows the 'Share Overview' page in OpenStack. The left sidebar has 'Shares' selected. The main content area displays details for a share named 'Share_test'. The 'Export locations' section shows a 'Path' field with a long string of IP addresses and a volume ID, circled in red. The 'Access Rules' section shows a rule for 'charlie' with an 'Access Key' circled in green.

Beware that different versions of ceph-common are available for different versions of Ubuntu and the syntax of the mount command could change.

For **Ubuntu 20.04**, the command is:

```
sudo mount -t ceph -v <mount PATH> <mount point on the VM> -o name=<CLIENT_ACCESS_TO>,secret=<ACCESS_KEY>
```

An example of the complete command is:

```
sudo mount -t ceph -v 10.35.1.9:6789,10.35.1.10:6789,10.35.1.11:6789,10.35.1.12:6789,10.35.1.13:6789:/volumes/_nogroup/43aa4ecc-1db6-4952-b2dd-6336b45075d5 /mnt/share_manila/ -o name=my-client-name,secret=AQBP07Nejv/RLhAABYqQ5tvgePh2EP7EL0UuhQ==
```

While, for **Ubuntu 22.04** and higher the command is:

```
sudo mount.ceph <CLIENT_ACCESS_TO>@482d24d4-df47-11eb-8d80-0c42a1f53648.g100_fs=<FS_ADDRESS> <mount point on the VM> -o mon_addr=<MON_IPS>,secretfile=<SECRET_FILE>
```

Where <FS_ADDRESS> and <MON_IPS> are the two parts of the "Path" string on OpenStack:

- <MON_IPS> is the first numeric part of the "Path" string, up to ":/volumes", where each IP has to be separated using the character "/" instead of ","
- <FS_ADDRESS> is everything else, from "/volumes/" to the end of the string

Finally, the <SECRET_FILE> is the path to a text file that contains the string <ACCESS_KEY>. Following the same example that uses the picture from above:

```
sudo mount.ceph charlie@482d24d4-df47-11eb-8d80-0c42a1f53648.g100_fs=/volumes/_nogroup/43aa4ecc-1db6-4952-b2dd-6336b45075d5 /mnt/share_manila/ -o mon_addr=10.35.1.9:6789/10.35.1.10:6789/10.35.1.11:6789/10.35.1.12:6789/10.35.1.13:6789,secretfile=/home/ubuntu/my_secret_file.txt
```

NB: If you are using a different Linux distribution, please **refer to the ceph user manual** to be sure that the syntax you are using is appropriate for the ceph version installed.

Then **repeat the same steps for the second VM as well**. Now the two VMs share the same filesystem.