



# Red Hat OpenStack Platform 16.0

## Deploying the Shared File Systems service with CephFS through NFS

Understanding, using, and managing the Shared File Systems service with CephFS  
through NFS in Red Hat OpenStack Platform



# Red Hat OpenStack Platform 16.0 Deploying the Shared File Systems service with CephFS through NFS

---

Understanding, using, and managing the Shared File Systems service with CephFS through NFS in Red Hat OpenStack Platform

OpenStack Team  
rhos-docs@redhat.com

## Legal Notice

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

Install, configure, and verify the Shared File Systems service (manila) with the Red Hat Ceph File System (CephFS) through NFS for the Red Hat OpenStack Platform (RHOSP) environment.

# Table of Contents

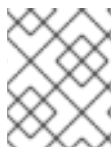
<b>PREFACE</b> .....	<b>3</b>
<b>CHAPTER 1. THE SHARED FILE SYSTEMS SERVICE WITH CEPHFS THROUGH NFS</b> .....	<b>4</b>
1.1. BENEFITS OF USING THE SHARED FILE SYSTEMS SERVICE WITH CEPHFS THROUGH NFS	4
1.2. CEPH FILE SYSTEM ARCHITECTURE	5
1.2.1. CephFS with native driver	5
1.2.2. CephFS through NFS	6
1.2.2.1. Ceph services and client access	7
1.2.2.2. Shared File Systems service with CephFS through NFS fault tolerance	8
<b>CHAPTER 2. CEPHFS THROUGH NFS INSTALLATION</b> .....	<b>9</b>
2.1. CEPHFS WITH NFS-GANESHA DEPLOYMENT	9
2.1.1. Requirements for CephFS through NFS	9
2.1.2. File shares	10
2.1.3. Isolated network used by CephFS through NFS	10
2.2. INSTALLING RED HAT OPENSTACK PLATFORM WITH CEPHFS THROUGH NFS AND A CUSTOM NETWORK_DATA FILE	10
2.2.1. Installing the ceph-ansible package	11
2.2.2. Preparing overcloud container images	11
2.2.2.1. Generating the custom roles file	12
2.2.3. Deploying the updated environment	12
2.2.3.1. StorageNFS and network_data_ganesha.yaml file	13
2.2.3.2. manila-cephfsganesha-config.yaml	14
2.2.4. Completing post-deployment configuration	15
2.2.4.1. Configuring the isolated network	15
2.2.4.2. Configuring the shared provider StorageNFS network	16
2.2.4.2.1. Configuring the shared provider StorageNFS IPv4 network	16
2.2.4.2.2. Configuring the shared provider StorageNFS IPv6 network	16
2.2.4.3. Configuring a default share type	17
<b>CHAPTER 3. VERIFYING SUCCESSFUL CEPHFS THROUGH NFS DEPLOYMENT</b> .....	<b>18</b>
3.1. VERIFYING CREATION OF ISOLATED STORAGENFS NETWORK	18
3.2. VERIFYING CEPH MDS SERVICE	18
3.3. VERIFYING CEPH CLUSTER STATUS	19
3.4. VERIFYING NFS-GANESHA AND MANILA-SHARE SERVICE STATUS	19
3.5. VERIFYING MANILA-API SERVICES ACKNOWLEDGES SCHEDULER AND SHARE SERVICES	20



## PREFACE

Red Hat OpenStack Platform (RHOSP) provides the foundation to build a private or public Infrastructure-as-a-Service (IaaS) cloud on top of Red Hat Enterprise Linux. It offers a massively scalable, fault-tolerant platform for the development of cloud-enabled workloads.

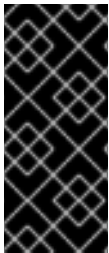
With the Shared File Systems service (manila) with Ceph File System (CephFS) through NFS, you can use the same Ceph cluster that you use for block and object storage to provide file shares through the NFS protocol. For more information, [Shared File Systems service](#) in the *Storage Guide*.



### NOTE

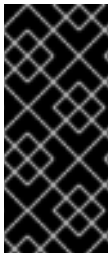
For the complete suite of documentation for Red Hat OpenStack Platform, see [Red Hat OpenStack Platform Documentation](#).

# CHAPTER 1. THE SHARED FILE SYSTEMS SERVICE WITH CEPHFS THROUGH NFS



## IMPORTANT

The Red Hat OpenStack Platform (RHOSP) Shared File Systems service with CephFS via NFS for RHOSP 16.0 and later is supported for use with Red Hat Ceph Storage version 4.1 or later. For more information about how to determine the version of Ceph Storage installed on your system, see [Red Hat Ceph Storage releases and corresponding Ceph package versions](#).



## IMPORTANT

The Red Hat OpenStack Platform (RHOSP) Shared File Systems service with CephFS through NFS for RHOSP 16.0 and later is supported for use with Red Hat Ceph Storage version 4.1 or later. For more information about how to determine the version of Ceph Storage installed on your system, see [Red Hat Ceph Storage releases and corresponding Ceph package versions](#).

CephFS is the highly scalable, open-source distributed file system component of Ceph, a unified distributed storage platform. Ceph implements object, block, and file storage using Reliable Autonomic Distributed Object Store (RADOS). CephFS, which is POSIX compatible, provides file access to a Ceph storage cluster.

You can use the Shared File Systems service to create shares in CephFS and access them with NFS 4.1 through NFS-Ganesha. NFS-Ganesha controls access to the shares and exports them to clients through the NFS 4.1 protocol. The Shared File Systems service manages the life cycle of these shares from within Red Hat OpenStack Platform (RHOSP). When cloud administrators configure the service to use CephFS through NFS, these file shares come from the CephFS cluster, but are created and accessed as familiar NFS shares.

For more information, see [Shared File Systems service](#) in the *Storage Guide*.

## 1.1. BENEFITS OF USING THE SHARED FILE SYSTEMS SERVICE WITH CEPHFS THROUGH NFS

- **Familiarity:** You can use the Shared File Systems service (manila) with CephFS through the NFS protocol to provide file shares through the NFS protocol, which is available by default on most operating systems. CephFS maximizes Ceph clusters that are already used as storage back ends for other services in the OpenStack cloud, such as Block Storage (cinder), object storage, and other services.



## NOTE

With this release, adding CephFS to an externally deployed Ceph cluster, which was not configured by Red Hat OpenStack (RHOSP) director, is supported. Currently, you can define only one CephFS back end in RHOSP director. For more information, see [Integrating with the existing Ceph Storage cluster](#) in the *Integrating an Overcloud with an Existing Red Hat Ceph Cluster* guide.

This version of Red Hat OpenStack Platform fully supports the CephFS NFS driver (NFS-Ganesha), unlike the CephFS native driver, which is a Technology Preview feature.





## IMPORTANT

Red Hat CephFS native driver is available only as a *Technology Preview*, and therefore is not fully supported by Red Hat.

For more information about Technology Preview features, see [Scope of Coverage Details](#).

- **Security:** In CephFS through NFS deployments, the Ceph Storage back end is separated from the user network. This configuration ensures that the underlying Ceph storage is less vulnerable to malicious attacks and inadvertent mistakes.
- **Security:** File storage is more secure because data-plane traffic and APIs use separate networks to communicate with control plane services, such as Shared File Systems services.
- **Control:** The Ceph client is under administrative control. The end user controls an NFS client, for example, an isolated user VM, that has no direct access to the Ceph cluster storage back end.

## 1.2. CEPH FILE SYSTEM ARCHITECTURE

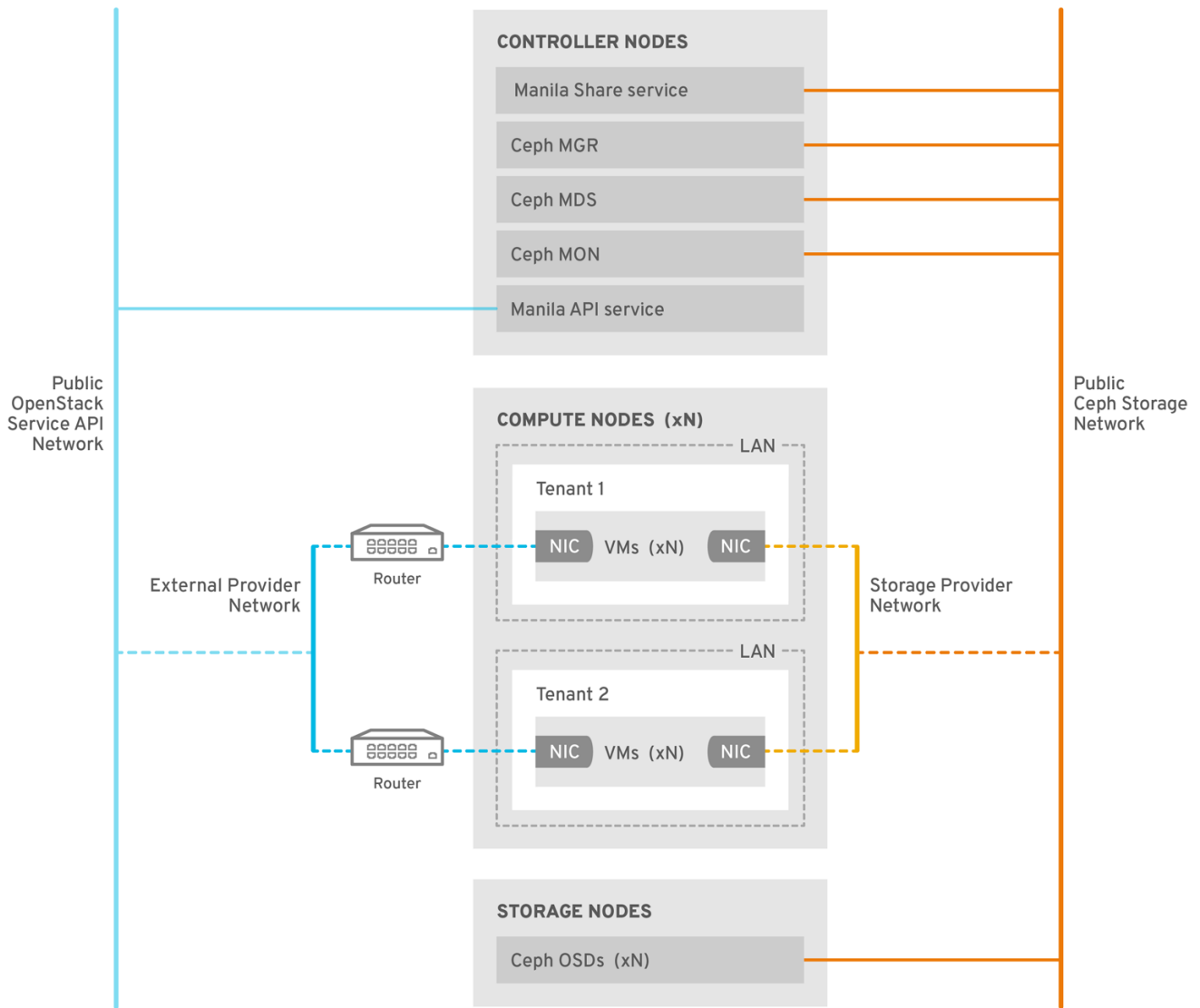
Ceph File System (CephFS) is a distributed file system that you can use with either NFS-Ganesha using the NFS v4 protocol (supported) or CephFS native driver.

### 1.2.1. CephFS with native driver

The CephFS native driver combines the OpenStack Shared File Systems service (manila) and Red Hat Ceph Storage. When you use Red Hat OpenStack (RHOSP) director, the Controller nodes host the Ceph daemons, such as the manager, metadata servers (MDS), and monitors (MON) and the Shared File Systems services.

Compute nodes can host one or more projects. Projects (formerly known as tenants), which are represented in the following graphic by the white boxes, contain user-managed VMs, which are represented by gray boxes with two NICs. To access the ceph and manila daemons projects connect to the daemons over the public Ceph storage network. On this network, you can access data on the storage nodes provided by the Ceph Object Storage Daemons (OSDs). Instances (VMs) that are hosted on the project boot with two NICs: one dedicated to the storage provider network and the second to project-owned routers to the external provider network.

The storage provider network connects the VMs that run on the projects to the public Ceph storage network. The Ceph public network provides back end access to the Ceph object storage nodes, metadata servers (MDS), and Controller nodes. Using the native driver, CephFS relies on cooperation with the clients and servers to enforce quotas, guarantee project isolation, and for security. CephFS with the native driver works well in an environment with trusted end users on a private cloud. This configuration requires software that is running under user control to cooperate and work correctly.



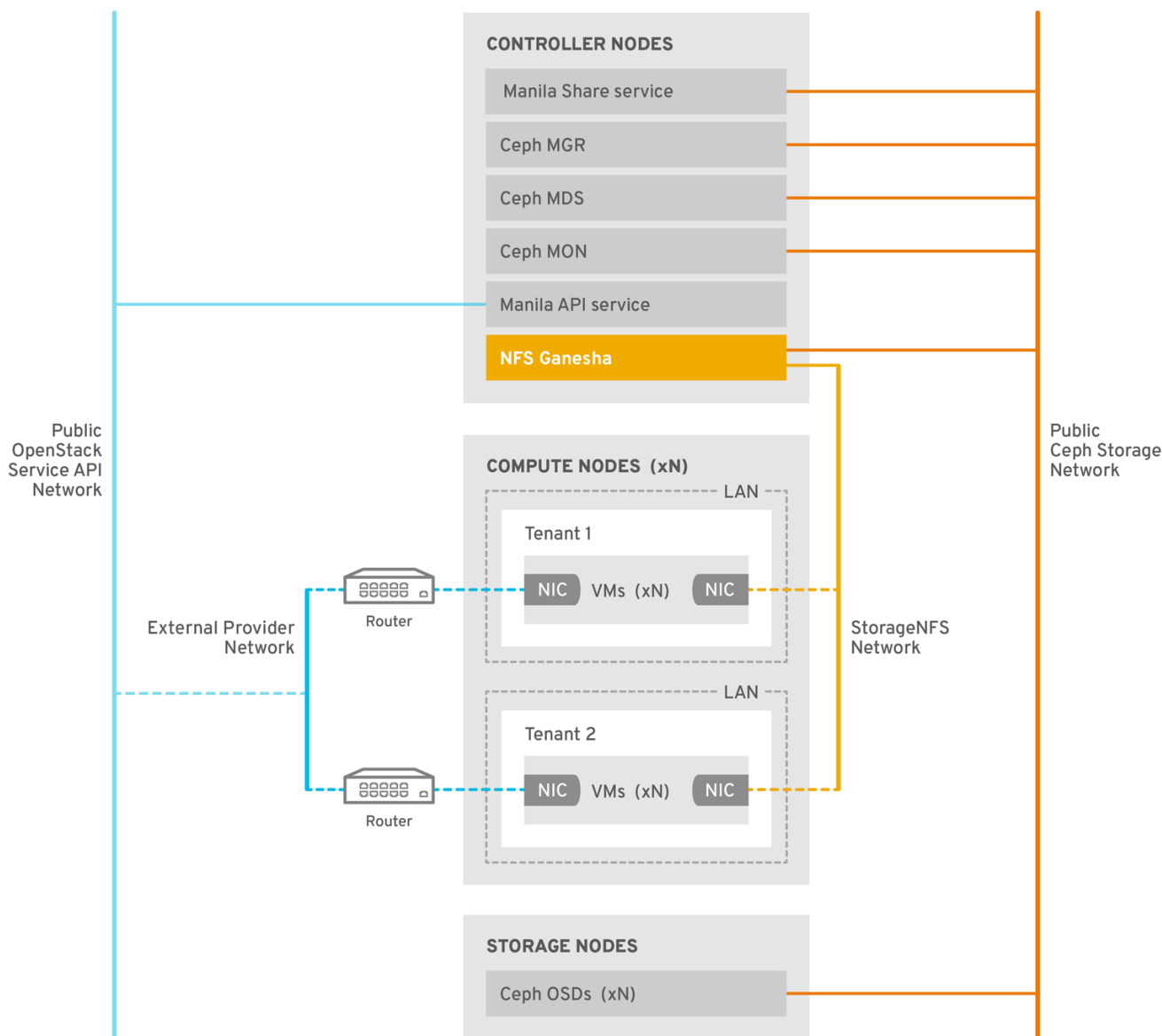
OPENSTACK\_476233\_0818

### 1.2.2. CephFS through NFS

The CephFS through NFS back end in the Shared File Systems service (manila) is composed of Ceph metadata servers (MDS), the CephFS through NFS gateway (NFS-Ganesha), and the Ceph cluster service components. The Shared File Systems service CephFS NFS driver uses NFS-Ganesha gateway to provide NFSv4 protocol access to CephFS shares. The Ceph MDS service maps the directories and file names of the file system to objects that are stored in RADOS clusters. NFS gateways can serve NFS file shares with different storage back ends, such as Ceph. The NFS-Ganesha service runs on the Controller nodes with the Ceph services.

Instances are booted with at least two NICs: one NIC connects to the project router and the second NIC connects to the StorageNFS network, which connects directly to the NFS-Ganesha gateway. The instance mounts shares by using the NFS protocol. CephFS shares that are hosted on Ceph OSD nodes are provided through the NFS gateway.

NFS-Ganesha improves security by preventing user instances from directly accessing the MDS and other Ceph services. Instances do not have direct access to the Ceph daemons.



OPENSTACK\_476233\_0818

### 1.2.2.1. Ceph services and client access

In addition to the monitor, OSD, Rados Gateway (RGW), and manager services deployed when Ceph provides object and block storage, a Ceph metadata service (MDS) is required for CephFS and an NFS-Ganesha service is required as a gateway to native CephFS using the NFS protocol. For user-facing object storage, an RGW service is also deployed. The gateway runs the CephFS client to access the Ceph public network and is under administrative rather than end-user control.

NFS-Ganesha runs in its own container that interfaces both to the Ceph public network and to a new isolated network, StorageNFS. The composable network feature of Red Hat OpenStack Platform (RHOSP) director deploys this network and connects it to the Controller nodes. As the cloud administrator, you can configure the network as a Networking (neutron) provider network.

NFS-Ganesha accesses CephFS over the Ceph public network and binds its NFS service using an address on the StorageNFS network.

To access NFS shares, provision user VMs, Compute (nova) instances, with an additional NIC that connects to the Storage NFS network. Export locations for CephFS shares appear as standard NFS **IP: <path>** tuples that use the NFS-Ganesha server VIP on the StorageNFS network. The network uses the IP address of the user VM to perform access control on the NFS shares.

Networking (neutron) security groups prevent the user VM that belongs to project 1 from accessing a user VM that belongs to project 2 over the StorageNFS network. Projects share the same CephFS file system but project data path separation is enforced because user VMs can access files only under export trees: **/path/to/share1/...., /path/to/share2/....**

### 1.2.2.2. Shared File Systems service with CephFS through NFS fault tolerance

When Red Hat OpenStack Platform (RHOSP) director starts the Ceph service daemons, they manage their own high availability (HA) state and, in general, there are multiple instances of these daemons running. By contrast, in this release, only one instance of NFS-Ganesha can serve file shares at a time.

To avoid a single point of failure in the data path for CephFS through NFS shares, NFS-Ganesha runs on a RHOSP Controller node in an active-passive configuration managed by a Pacemaker-Corosync cluster. NFS-Ganesha acts across the Controller nodes as a virtual service with a virtual service IP address.

If a Controller node fails or the service on a particular Controller node fails and cannot be recovered on that node, Pacemaker-Corosync starts a new NFS-Ganesha instance on a different Controller node using the same virtual IP address. Existing client mounts are preserved because they use the virtual IP address for the export location of shares.

Using default NFS mount-option settings and NFS 4.1 or later, after a failure, TCP connections are reset and clients reconnect. I/O operations temporarily stop responding during failover, but they do not fail. Application I/O also stops responding but resumes after failover completes.

New connections, new lock-state, and so on are refused until after a grace period of up to 90 seconds during which time the server waits for clients to reclaim their locks. NFS-Ganesha keeps a list of the clients and exits the grace period earlier if all clients reclaim their locks.



#### NOTE

The default value of the grace period is 90 seconds. To change this value, edit the NFSv4 **Grace\_Period** configuration option.

## CHAPTER 2. CEPHFS THROUGH NFS INSTALLATION

### 2.1. CEPHFS WITH NFS-GANESHA DEPLOYMENT

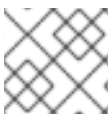
A typical Ceph file system (CephFS) through NFS installation in a Red Hat OpenStack Platform (RHOSP) environment includes the following configurations:

- OpenStack Controller nodes running containerized Ceph metadata server (MDS), Ceph monitor (MON), manila, and NFS-Ganesha services. Some of these services can coexist on the same node or can have one or more dedicated nodes.
- Ceph storage cluster with containerized object storage daemons (OSDs) running on Ceph storage nodes.
- An isolated StorageNFS network that provides access from projects to the NFS-Ganesha services for NFS share provisioning.

The Shared File Systems service (manila) provides APIs that allow the projects to request file system shares, which are fulfilled by driver modules. The driver for Red Hat CephFS, **manila.share.drivers.cephfs.driver.CephFSDriver**, means that you can use the Shared File Systems service as a CephFS as a back end. RHOSP director configures the driver to deploy the NFS-Ganesha gateway so that the CephFS shares are presented through the NFS 4.1 protocol.

Using RHOSP director to deploy the Shared File Systems service with a CephFS back end on the overcloud automatically creates the required storage network defined in the heat template. For more information about network planning, see [Overcloud networks](#) in the *Director Installation and Usage* guide.

Although you can manually configure the Shared File Systems service by editing its node `/etc/manila/manila.conf` file, RHOSP director can override any settings in future overcloud updates. The recommended method for configuring a Shared File System back end is through director.



#### NOTE

Currently, you can define only one CephFS back end at a time in director.

CephFS through NFS

#### 2.1.1. Requirements for CephFS through NFS

CephFS through NFS requires a Red Hat OpenStack Platform (RHOSP) version 13 or later environment, which can be an existing or a new environment.

- For RHOSP versions 13, 14, and 15, CephFS works with Red Hat Ceph Storage (RHCS) version 3.
- For RHOSP version 16 or later, CephFS works with Red Hat Ceph Storage (RHCS) version 4.1 or later.

For more information, see the [Deploying an Overcloud with Containerized Red Hat Ceph Guide](#) .

#### Prerequisites

- You install the Shared File Systems service on Controller nodes, as is the default behavior.

- You install the NFS-Ganesha gateway service on Pacemaker cluster of the Controller node.
- You configure only a single instance of a CephFS back end to use the Shared File Systems service. You can use other non-CephFS back ends with the single CephFS back end.
- You use RHOSP director to create an extra network (StorageNFS) for the storage traffic.
- You configure a new RHCS version 4.1 or later cluster at the same time as CephFS through NFS.

### 2.1.2. File shares

File shares are handled differently in the Shared File Systems service (manila), Ceph File System (CephFS), and Ceph through NFS.

The Shared File Systems service provides shares. A share is an individual file system namespace and a unit of storage or sharing and a defined size for example, subdirectories with quotas. Shared file system storage enables multiple clients because the file system is configured before access is requested unlike block storage, which is configured when it is requested.

With CephFS, a share is considered a directory with a defined quota and a layout that points to a particular storage pool or namespace. CephFS quotas limit the size of a directory to the size share that the Shared File Systems service creates. Access to Ceph shares is determined by MDS authentication capabilities.

With CephFS through NFS, file shares are provisioned and accessed through the NFS protocol. The NFS protocol also handles security.

### 2.1.3. Isolated network used by CephFS through NFS

CephFS through NFS deployments use an extra isolated network, StorageNFS. This network is deployed so users can mount shares over NFS on that network without accessing the Storage or Storage Management networks which are reserved for infrastructure traffic.

For more information about isolating networks, see [Basic network isolation](#) in the *Advanced Overcloud Customization* guide.

## 2.2. INSTALLING RED HAT OPENSTACK PLATFORM WITH CEPHFS THROUGH NFS AND A CUSTOM NETWORK\_DATA FILE

To install CephFS through NFS, complete the following procedures:

1. Install the `ceph-ansible` package. See [Section 2.2.1, "Installing the ceph-ansible package"](#)
2. Prepare the overcloud container images with the **openstack overcloud image prepare** command. See [Section 2.2.2, "Preparing overcloud container images"](#)
3. Generate the custom roles file, **roles\_data.yaml**, and **network\_data.yaml** file. See [Section 2.2.2.1, "Generating the custom roles file"](#)
4. Deploy Ceph, Shared File Systems service (manila), and CephFS using the **openstack overcloud deploy** command with custom roles and environments. See [Section 2.2.3, "Deploying the updated environment"](#)

5. Configure the isolated StorageNFS network and create the default share type. See [Section 2.2.4, “Completing post-deployment configuration”](#)

Examples use the standard **stack** user in the Red Hat Platform (RHOSP) environment.

Perform tasks as part of a RHOSP installation or environment update.

## 2.2.1. Installing the ceph-ansible package

Install the **ceph-ansible** package to be installed on an undercloud node to deploy containerized Ceph.

### Procedure

1. Log in to an undercloud node as the **stack** user.
2. Install the ceph-ansible package:

```
[stack@undercloud-0 ~]$ sudo dnf install -y ceph-ansible
[stack@undercloud-0 ~]$ sudo dnf list ceph-ansible
...
Installed Packages
ceph-ansible.noarch 3.1.0-0.1.el7
```

## 2.2.2. Preparing overcloud container images

Because all services are containerized in Red Hat OpenStack Platform (RHOSP), you must prepare container images for the overcloud by using the **openstack overcloud image prepare** command. Enter this command with the additional options to add default images for the **ceph** and **manila** services to the container registry. **Ceph MDS** and **NFS-Ganesha** services use the same Ceph base container image.

For more information about container images, see [Container Images for Additional Services](#) in the *Director Installation and Usage* guide.

### Procedure

1. From the undercloud as the **stack** user, enter the **openstack overcloud image prepare** command with **-e** to include the following environment files:

```
$ openstack overcloud container image prepare \
...
-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/services/manila.yaml \
...
```

2. Use `grep` to verify that the default images for the ceph and manila services are available in the **containers-default-parameters.yaml** file.

```
[stack@undercloud-0 ~]$ grep -E 'ceph|manila' composable_roles/docker-images.yaml
DockerCephDaemonImage: 192.168.24.1:8787/rhceph-beta/rhceph-4-rhel8-4-12
DockerManilaApiImage: 192.168.24.1:8787/rhosp-rhel8/openstack-manila-api:2019-01-16
DockerManilaConfigImage: 192.168.24.1:8787/rhosp-rhel8/openstack-manila-api:2019-01-16
DockerManilaSchedulerImage: 192.168.24.1:8787/rhosp-rhel8/openstack-manila-
```

```
scheduler:2019-01-16
```

```
DockerManilaShareImage: 192.168.24.1:8787/rhosp-rhel8/openstack-manila-share:2019-01-16
```

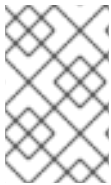
### 2.2.2.1. Generating the custom roles file

The **ControllerStorageNFS** custom role configures the isolated **StorageNFS** network. This role is similar to the default **Controller.yaml** role file with the addition of the **StorageNFS** network and the **CephNfs** service, indicated by the **OS::TripleO::Services::CephNfs** command.

```
[stack@undercloud ~]$ cd /usr/share/openstack-tripleo-heat-templates/roles
[stack@undercloud roles]$ diff Controller.yaml ControllerStorageNfs.yaml
16a17
> - StorageNFS
50a45
> - OS::TripleO::Services::CephNfs
```

For more information about the **openstack overcloud roles generate** command, see [Roles](#) in the *Advanced Overcloud Customization* guide.

The **openstack overcloud roles generate** command creates a custom **roles\_data.yaml** file including the services specified after **-o**. In the following example, the **roles\_data.yaml** file created has the services for **ControllerStorageNfs**, **Compute**, and **CephStorage**.



#### NOTE

If you have an existing **roles\_data.yaml** file, modify it to add **ControllerStorageNfs**, **Compute**, and **CephStorage** services to the configuration file. For more information, see [Roles](#) in the *Advanced Overcloud Customization* guide.

#### Procedure

1. Log in to an undercloud node as the **stack** user,
2. Use the **openstack overcloud roles generate** command to create the **roles\_data.yaml** file:

```
[stack@undercloud ~]$ openstack overcloud roles generate --roles-path
/usr/share/openstack-tripleo-heat-templates/roles -o /home/stack/roles_data.yaml
ControllerStorageNfs Compute CephStorage
```

### 2.2.3. Deploying the updated environment

When you are ready to deploy your environment, use the **openstack overcloud deploy** command with the custom environments and roles required to run CephFS with NFS-Ganesha.

The overcloud deploy command has the following options in addition to other required options.



Action	Option	Additional information
Add the updated default containers from the <b>overcloud container image prepare</b> command.	<code>-e /home/stack/containers-default-parameters.yaml`</code>	<a href="#">Section 2.2.2, “Preparing overcloud container images”</a>
Add the extra StorageNFS network with <b>network_data_ganesha.yaml</b> .	<code>-n /usr/share/openstack-tripleo-heat-templates/network_data_ganesha.yaml`</code>	<a href="#">Section 2.2.3.1, “StorageNFS and network_data_ganesha.yaml file”</a>
Add the custom roles defined in <b>roles_data.yaml</b> file from the previous section.	<code>-r /home/stack/roles_data.yaml.</code>	<a href="#">Section 2.2.2.1, “Generating the custom roles file”</a>
Deploy the Ceph daemons with <b>ceph-ansible.yaml</b> .	<code>-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml</code>	<a href="#">Initiating Overcloud Deployment</a> in the <i>Deploying an Overcloud with Containerized Red Hat Ceph</i> guide
Deploy the Ceph metadata server with <b>ceph-mds.yaml</b> .	<code>-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-mds.yaml.</code>	<a href="#">Initiating Overcloud Deployment</a> in the <i>Deploying an Overcloud with Containerized Red Hat Ceph</i> guide
Deploy the <b>manila</b> service with the CephFS through NFS back end. Configure NFS-Ganesha with director.	<code>-e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml</code>	<a href="#">Section 2.2.3.2, “manila-cephfsganesha-config.yaml”</a>

The following example shows an **openstack overcloud deploy** command with options to deploy CephFS through NFS-Ganesha, Ceph cluster, Ceph MDS, and the isolated StorageNFS network:

```
[stack@undercloud ~]$ openstack overcloud deploy \
--templates /usr/share/openstack-tripleo-heat-templates \
-n /usr/share/openstack-tripleo-heat-templates/network_data_ganesha.yaml \
-r /home/stack/roles_data.yaml \
-e /home/stack/containers-default-parameters.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/net-single-nic-with-vlans.yaml \
-e /home/stack/network-environment.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-mds.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml
```

For more information about the **openstack overcloud deploy** command, see [Deployment command](#) in the *Director Installation and Usage* guide.

### 2.2.3.1. StorageNFS and network\_data\_ganesha.yaml file

Use composable networks to define custom networks and assign them to any role. Instead of using the standard **network\_data.yaml** file, you can configure the StorageNFS composable network with the **network\_data\_ganesha.yaml** file. Both of these roles are available in the **/usr/share/openstack-tripleo-heat-templates** directory.

The **network\_data\_ganesha.yaml** file contains an additional section that defines the isolated StorageNFS network. Although the default settings work for most installations, you must edit the YAML file to add your network settings, including the VLAN ID, subnet, and other settings.

```
name: StorageNFS
enabled: true
vip: true
name_lower: storage_nfs
vlan: 70
ip_subnet: '172.16.4.0/24'
allocation_pools: [{'start': '172.16.4.4', 'end': '172.16.4.149'}]
ipv6_subnet: 'fd00:fd00:fd00:7000::/64'
ipv6_allocation_pools: [{'start': 'fd00:fd00:fd00:7000::10', 'end': 'fd00:fd00:fd00:7000:ffff:ffff:ffff:ffff'}]
```

For more information about composable networks, see [Using Composable Networks](#) in the *Advanced Opencloud Customization* guide.

### 2.2.3.2. manila-cephfsganesha-config.yaml

The integrated environment file for defining a CephFS back end is located in the following path of an undercloud node:

```
/usr/share/openstack-tripleo-heat-templates/environments/
```

The **manila-cephfsganesha-config.yaml** environment file contains settings relevant to the deployment of the Shared File Systems service. The back end default settings work for most environments. The following example shows the default values that director uses during deployment of the Shared File Systems service:

```
[stack@undercloud ~]$ cat /usr/share/openstack-tripleo-heat-templates/environments/manila-
cephfsganesha-config.yaml
# A Heat environment file which can be used to enable a
# a Manila CephFS-NFS driver backend.
resource_registry:
  OS::TripleO::Services::ManilaApi: ../deployment/manila/manila-api-container-puppet.yaml
  OS::TripleO::Services::ManilaScheduler: ../deployment/manila/manila-scheduler-container-
puppet.yaml
# Only manila-share is pacemaker managed:
OS::TripleO::Services::ManilaShare: ../deployment/manila/manila-share-pacemaker-puppet.yaml
OS::TripleO::Services::ManilaBackendCephFs: ../deployment/manila/manila-backend-cephfs.yaml
# ceph-nfs (ganesha) service is installed and configured by ceph-ansible
# but it's still managed by pacemaker
OS::TripleO::Services::CephNfs: ../deployment/ceph-ansible/ceph-nfs.yaml

parameter_defaults:
  ManilaCephFSBackendName: cephfs 1
  ManilaCephFSDriverHandlesShareServers: false 2
  ManilaCephFSCephFSAuthId: 'manila' 3
  ManilaCephFSCephFSEnableSnapshots: false 4
```

```
# manila cephfs driver supports either native cephfs backend - 'CEPHFS'
# (users mount shares directly from ceph cluster), or nfs-ganesha backend -
# 'NFS' (users mount shares through nfs-ganesha server)
ManilaCephFSCephFSProtocolHelperType: 'NFS'
```

The **parameter\_defaults** header signifies the start of the configuration. In this section, you can edit settings to override default values set in **resource\_registry**. This includes values set by **OS::Tripleo::Services::ManilaBackendCephFs**, which sets defaults for a CephFS back end.

- 1 **ManilaCephFSBackendName** sets the name of the manila configuration of your CephFS backend. In this case, the default back end name is **cephfs**.
- 2 **ManilaCephFSDriverHandlesShareServers** controls the lifecycle of the share server. When set to **false**, the driver does not handle the lifecycle. This is the only supported option.
- 3 **ManilaCephFSCephFSAuthId** defines the Ceph auth ID that the director creates for the **manila** service to access the Ceph cluster.
- 4 **ManilaCephFSCephFSEnableSnapshots** controls snapshot activation. The **false** value indicates that snapshots are not enabled. This feature is currently not supported.

For more information about environment files, refer to the [Environment Files](#) section in the *Director Installation and Usage Guide*.

## 2.2.4. Completing post-deployment configuration

You must complete two post-deployment configuration tasks before you create NFS shares, grant user access, and mount NFS shares.

- Map the neutron StorageNFS network to the isolated data center Storage NFS network. See [Section 2.2.4.1, "Configuring the isolated network"](#)
- Create the default share type. See [Section 2.2.4.3, "Configuring a default share type"](#)

### 2.2.4.1. Configuring the isolated network

Map the new isolated StorageNFS network to a neutron-shared provider network. The Compute VMs attach to this neutron network to access share export locations provided by the NFS-Ganesha gateway.

For more information about network security with the Shared File Systems service, see [Hardening the Shared File System Service](#) in the *Security and Hardening Guide*.

The **openstack network create** command defines the configuration for the StorageNFS neutron network. You can enter this command with the following options:

- For **--provider-network-type**, use the value **vlan**.
- For **--provider-physical-network**, use the default value **datacentre**, unless you set another tag for the br-isolated bridge through **NeutronBridgeMappings** in your tripleo-heat-templates.
- For **--provider-segment**, use the VLAN value set for the StorageNFS isolated network in the heat template, **/usr/share/openstack-tripleo-heat-templates/network\_data\_ganesha.yaml**. This value is 70, unless the deployer modified the isolated network definitions.

## Procedure

1. On an undercloud node as the **stack** user, enter the following command:

```
[stack@undercloud ~]$ source ~/overcloudrc
```

2. On an undercloud node, enter the **openstack network create** command to create the StorageNFS network:

```
(overcloud) [stack@undercloud-0 ~]$ openstack network create StorageNFS --share --
provider-network-type vlan --provider-physical-network datacentre --provider-segment 70
```

### 2.2.4.2. Configuring the shared provider StorageNFS network

Create a corresponding **StorageNFSSubnet** on the neutron-shared provider network. Ensure that the subnet is the same as the **storage\_nfs** network definition in the **network\_data.yml** file and ensure that the allocation range for the **StorageNFS** subnet and the corresponding undercloud subnet do not overlap. No gateway is required because the **StorageNFS** subnet is dedicated to serving NFS shares.

#### Prerequisites

- The start and ending IP range for the allocation pool.
- The subnet IP range.

#### 2.2.4.2.1. Configuring the shared provider StorageNFS IPv4 network

##### Procedure

1. Log in to an overcloud node.
2. Source your overcloud credentials.
3. Use the example command to provision the network and make the following updates:
  - a. Replace the **start=172.16.4.150,end=172.16.4.250** IP values with the IP values for your network.
  - b. Replace the **172.16.4.0/24** subnet range with the subnet range for your network.

```
[stack@undercloud-0 ~]$ openstack subnet create --allocation-pool
start=172.16.4.150,end=172.16.4.250 --dhcp --network StorageNFS --subnet-range 172.16.4.0/24 --
gateway none StorageNFSSubnet
```

#### 2.2.4.2.2. Configuring the shared provider StorageNFS IPv6 network

This feature is available in this release as a *Technology Preview*, and therefore is not fully supported by Red Hat. It should only be used for testing, and should not be deployed in a production environment. For more information about Technology Preview features, see [Scope of Coverage Details](#).

##### Procedure

1. Log in to an overcloud node.
2. Use the sample command to provision the network, updating values as needed.

- Replace the **fd00:fd00:fd00:7000::/64** subnet range with the subnet range for your network.

```
[stack@undercloud-0 ~]$ openstack subnet create --ip-version 6 --dhcp --network StorageNFS --  
subnet-range fd00:fd00:fd00:7000::/64 --gateway none --ipv6-ra-mode dhcpv6-stateful --ipv6-  
address-mode dhcpv6-stateful StorageNFSSubnet -f yaml
```

### 2.2.4.3. Configuring a default share type

You can use the Shared File Systems service to define share types that you can use to create shares with specific settings. Share types work like Block Storage volume types. Each type has associated settings, for example, extra specifications. When you invoke the type during share creation the settings apply to the shared file system.

Red Hat OpenStack Platform (RHOSP) director expects a default share type. You must create the default share type before you open the cloud for users to access. For CephFS with NFS, use the **manila type-create** command:

```
manila type-create default false
```

For information about share types, see [Creating and managing shares](#) in the *Storage Guide*.

## CHAPTER 3. VERIFYING SUCCESSFUL CEPHFS THROUGH NFS DEPLOYMENT

When you deploy CephFS through NFS as a back end of the Shared File Systems service (manila), you add the following new elements to the overcloud environment:

- StorageNFS network
- Ceph MDS service on the controllers
- NFS-Ganesha service on the controllers

For more information about using the Shared File Systems service with CephFS through NFS, see [Shared File Systems service](#) in the *Storage Guide*.

As the cloud administrator, you must verify the stability of the CephFS through NFS environment before you make it available to service users.

### 3.1. VERIFYING CREATION OF ISOLATED STORAGE NFS NETWORK

The **network\_data\_ganesha.yaml** file used to deploy CephFS through NFS as a Shared File Systems service back end creates the StorageNFS VLAN. Complete the following steps to verify the existence of the isolated StorageNFS network.

#### Prerequisites

- Complete the steps in [Chapter 2, CephFS through NFS installation](#)

#### Procedure

1. Log in to one of the controllers in the overcloud.
2. Enter the following command to check the connected networks and verify the existence of the VLAN as set in **network\_data\_ganesha.yaml**:

```
$ ip a
15: vlan310: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/ether 32:80:cf:0e:11:ca brd ff:ff:ff:ff:ff:ff
    inet 172.16.4.4/24 brd 172.16.4.255 scope global vlan310
        valid_lft forever preferred_lft forever
    inet 172.16.4.7/32 brd 172.16.4.255 scope global vlan310
        valid_lft forever preferred_lft forever
    inet6 fe80::3080:cfff:fe0e:11ca/64 scope link
        valid_lft forever preferred_lft forever
```

### 3.2. VERIFYING CEPH MDS SERVICE

Use the **systemctl status** command to verify the Ceph MDS service status.

#### Procedure

1. Enter the following command on all Controller nodes to check the status of the MDS container:

-

```
$ systemctl status ceph-mds@<CONTROLLER-HOST>
```

```
ceph-mds@controller-0.service - Ceph MDS
  Loaded: loaded (/etc/systemd/system/ceph-mds@.service; enabled; vendor preset:
disabled)
  Active: active (running) since Tue 2018-09-18 20:11:53 UTC; 6 days ago
  Main PID: 65066 (conmon)
```

### 3.3. VERIFYING CEPH CLUSTER STATUS

Complete the following steps to verify Ceph cluster status.

#### Procedure

1. Log in to the active Controller node.
2. Enter the following command:

```
$ sudo ceph -s

cluster:
  id: 3369e280-7578-11e8-8ef3-801844eeec7c
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum overcloud-controller-1,overcloud-controller-2,overcloud-
controller-0
  mgr: overcloud-controller-1(active), standbys: overcloud-controller-2, overcloud-controller-0
  mds: cephfs-1/1/1 up {0=overcloud-controller-0=up:active}, 2 up:standby
  osd: 6 osds: 6 up, 6 in
```

#### Result

Notice there is one active MDS and two MDSs on standby.

3. To check the status of the Ceph file system in more detail, enter the following command and replace **<cephfs>** with the name of the Ceph file system:

```
$ sudo ceph fs ls

name: cephfs, metadata pool: manila_metadata, data pools: [manila_data]
```

### 3.4. VERIFYING NFS-GANESHA AND MANILA-SHARE SERVICE STATUS

Complete the following step to verify the status of NFS-Ganesha and manila-share service.

#### Procedure

- Enter the following command from one of the Controller nodes to confirm that **ceph-nfs** and **openstack-manila-share** started:

```
$ pcs status
```

```
ceph-nfs    (systemd:ceph-nfs@pacemaker): Started overcloud-controller-1

podman container: openstack-manila-share [192.168.24.1:8787/rhosp-rhel8/openstack-
manila-share:pcmklatest]
  openstack-manila-share-podman-0    (ocf::heartbeat:podman): Started overcloud-
controller-1
```

### 3.5. VERIFYING MANILA-API SERVICES ACKNOWLEDGES SCHEDULER AND SHARE SERVICES

Complete the following steps to confirm the **manila-api** service acknowledges the scheduler and share services.

#### Procedure

1. Log in to the undercloud.
2. Enter the following command:

```
$ source /home/stack/overcloudrc
```

3. Enter the following command to confirm **manila-scheduler** and **manila-share** are enabled:

```
$ manila service-list
```

```
| Id | Binary          | Host           | Zone | Status | State | Updated_at |
```

```
| 2 | manila-scheduler | hostgroup     | nova | enabled | up | 2018-08-08T04:15:03.000000 |
| 5 | manila-share    | hostgroup@cephfs | nova | enabled | up | 2018-08-08T04:15:03.000000 |
```