Red Hat OpenStack Platform 16.1

Integrating an Overcloud with an Existing Red Hat Ceph Cluster

Configuring an Overcloud to Use Stand-Alone Red Hat Ceph Storage

Last Updated: 2020-09-25

Red Hat OpenStack Platform 16.1 Integrating an Overcloud with an Existing Red Hat Ceph Cluster

Configuring an Overcloud to Use Stand-Alone Red Hat Ceph Storage

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Copyright © 2020 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 ® is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js ® 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 ® 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

This guide describes how to use Red Hat OpenStack Platform director to integrate an Overcloud with an existing, stand-alone Red Hat Ceph cluster.

Table of Contents

CHAPTER 1. INTRODUCTION

Red Hat OpenStack Platform director creates a cloud environment called the *overcloud*. The director provides the ability to configure extra features for an overcloud, including integration with Red Hat Ceph Storage (both Ceph Storage clusters created with the director or existing Ceph Storage clusters).

1.1. INTRODUCTION TO CEPH STORAGE

Red Hat Ceph Storage is a distributed data object store designed to provide excellent performance, reliability, and scalability. Distributed object stores are the future of storage, because they accommodate unstructured data, and because clients can use modern object interfaces and legacy interfaces simultaneously. At the core of every Ceph deployment is the Ceph Storage cluster, which consists of several types of daemons, but primarily, these two:

Ceph OSD (Object Storage Daemon)

Ceph OSDs store data on behalf of Ceph clients. Additionally, Ceph OSDs utilize the CPU and memory of Ceph nodes to perform data replication, rebalancing, recovery, monitoring and reporting functions.

Ceph Monitor

A Ceph monitor maintains a master copy of the Ceph storage cluster map with the current state of the storage cluster.

For more information about Red Hat Ceph Storage, see the Red Hat Ceph Storage Architecture Guide .

1.2. DEFINING THE SCENARIO

This guide provides instructions on integrating an existing Ceph Storage cluster with an overcloud. This means the director configures the overcloud to use the Ceph Storage cluster for storage needs. You manage and scale the cluster itself outside of the Overcloud configuration.

1.3. DEPLOY THE SHARED FILE SYSTEMS SERVICE WITH EXTERNAL CEPHFS THROUGH NFS

When Red Hat OpenStack Platform director deploys the Shared File Systems service with CephFS through NFS, it deploys the NFS-Ganesha gateway on Controller nodes managed by Pacemaker (PCS). PCS manages cluster availability using an active-passive configuration.

This feature enables director to deploy the Shared File System with an external Ceph Storage cluster. In this type of deployment, Ganesha still runs on the Controller nodes managed by PCS.

You must edit the parameters in **ceph-ansible-external.yaml** to integrate the Shared File Systems with an external Ceph Storage cluster.

NOTE

You must edit the **ceph-ansible-external.yaml** file to configure any OpenStack Platform service to use an external Ceph Storage cluster.

For more information about how to configure the **ceph-ansible-external.yaml** file, see Integrating with the existing Ceph cluster .

NOTE

This feature is supported with Ceph Storage 4.1 or later. After you upgrade to Ceph Storage 4.1, you must install the latest version of the **ceph-ansible** package on the undercloud. For more information about how to determine the Ceph Storage release installed on your system, see Red Hat Ceph Storage releases and corresponding Ceph package versions.

For more information about how to update the **ceph-ansible** package on the undercloud, see Installing the ceph-ansible package.

Prerequisites

The following prerequisites are required to configure the Shared File Systems service (manila) with an external Ceph Storage cluster:

- The external Ceph Storage cluster must have an active MDS.
- The external Ceph Storage cluster must have a CephFS file system based on the values of the CephFS data (**ManilaCephFSDataPoolName**) and CephFS metadata pools (**ManilaCephFSMetadataPoolName**). For more information, see Creating a custom environment file.
- The external Ceph Storage cluster must have a **cephx** client key for the Shared File Systems service and NFS-Ganesha. For more information, see Creating a custom environment file.
- You must have the **cephx** ID and client key to configure the Shared File Systems service and NFS-Ganesha. For more information, see Creating a custom environment file.

For more information about Red Hat Ceph Storage, see the Red Hat Ceph Storage File System Guide .

CHAPTER 2. PREPARING OVERCLOUD NODES

The scenario described in this chapter consists of six nodes in the Overcloud:

- Three Controller nodes with high availability.
- Three Compute nodes.

The director integrates a separate Ceph Storage cluster with its own nodes into the overcloud. You manage this cluster independently from the overcloud. For example, you scale the Ceph Storage cluster using the Ceph management tools, not through the OpenStack Platform director. For more information, see the Red Hat Ceph Storage documentation library.

2.1. CONFIGURING THE EXISTING CEPH STORAGE CLUSTER

1. Create the following pools in your Ceph cluster relevant to your environment:
   - **volumes**: Storage for OpenStack Block Storage (cinder)
   - **images**: Storage for OpenStack Image Storage (glance)
   - **vms**: Storage for instances
   - **backups**: Storage for OpenStack Block Storage Backup (cinder-backup)
   - **metrics**: Storage for OpenStack Telemetry Metrics (gnocchi)
     Use the following commands as a guide:

     ```
     [root@ceph ~]# ceph osd pool create volumes PGNUM
     [root@ceph ~]# ceph osd pool create images PGNUM
     [root@ceph ~]# ceph osd pool create vms PGNUM
     [root@ceph ~]# ceph osd pool create backups PGNUM
     [root@ceph ~]# ceph osd pool create metrics PGNUM
     ```

     If your overcloud deploys the Shared File System (manila) backed by CephFS, also create CephFS data and metadata pools:

     ```
     [root@ceph ~]# ceph osd pool create manila_data PGNUM
     [root@ceph ~]# ceph osd pool create manila_metadata PGNUM
     ```

   Replace *PGNUM* with the number of placement groups. Red Hat recommends approximately 100 placement groups per OSD. For example, the total number of OSDs multiplied by 100, divided by the number of replicas (**osd pool default size**). You can also use the Ceph Placement Groups (PGs) per Pool Calculator to determine a suitable value.

2. Create a **client.openstack** user in your Ceph cluster with the following capabilities:
   - **cap_mgr:** "allow *"
   - **cap_mon:** profile rbd
   - **cap_osd:** profile rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd pool=backups, profile rbd pool=metrics
     Use the following command as a guide:

```
[root@ceph ~]# ceph auth add client.openstack mgr 'allow *' mon 'profile rbd' osd 'profile
rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd
pool=backups, profile rbd pool=metrics'
```

3. Note the Ceph client key created for the **client.openstack** user:

```
[root@ceph ~]# ceph auth list
...
[client.openstack]
 key = AQC+vYNXgDAgAhAAc8UoYt+OTz5uhV7ItLdwUw==
 caps mgr = "allow *"
 caps mon = "profile rbd"
 caps osd = "profile rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile
rbd pool=backups, profile rbd pool=metrics"
...
```

   The **key** value in the example, **AQC+vYNXgDAgAhAAc8UoYt+OTz5uhV7ItLdwUw==**, is your Ceph client key.

4. If your overcloud deploys the Shared File System backed by CephFS, create the **client.manila** user in your Ceph cluster with the following capabilities:

   - **cap_mds: allow ***
   - **cap_mgr: allow ***
   - **cap_mon: allow r, allow command "auth del", allow command "auth caps", allow command "auth get", allow command "auth get-or-create"**
   - **cap_osd: allow rw** Use the following command as a guide:

```
[root@ceph ~]# ceph auth add client.manila mon 'allow r, allow command "auth del",
allow command "auth caps", allow command "auth get", allow command "auth get-or-
create"' osd 'allow rw' mds 'allow *' mgr 'allow *'
```

5. Note the manila client name and the key value to use in overcloud deployment templates:

```
[root@ceph ~]# ceph auth get-key client.manila
AQDQ991cAAAAABAA0aXFrTnjH9aO39P0iVvYyg==
```

6. Note the file system ID of your Ceph Storage cluster. This value is specified with the **fsid** setting in the configuration file of your cluster (in the **[global]** section):

```
[global]
fsid = 4b5c8c0a-ff60-454b-a1b4-9747aa737d19
...
```

NOTE

For more information about the Ceph Storage cluster configuration file, see Ceph configuration in the *Red Hat Ceph Storage Configuration Guide* .

The Ceph client key and file system ID, as well as the manila client IDS and key, will all be used later in Chapter 3, *Integrating with the existing Ceph Storage cluster* .

2.2. INITIALIZING THE STACK USER

- Log into the director host as the **stack** user and enter the following command to initialize your director configuration:

```
$ source ~/stackrc
```

Result

The environment variables that contain the authentication details to access the director CLI tools are configured.

2.3. REGISTERING NODES

The following example shows a node definition template, **instackenv.json**, which is a JSON format file and contains the hardware and power management details to register nodes:

```
{
    "nodes":[
        {
            "mac":[
                "bb:bb:bb:bb:bb:bb"
            ],
            "cpu":"4",
            "memory":"6144",
            "disk":"40",
            "arch":"x86_64",
            "pm_type":"pxe_ipmitool",
            "pm_user":"admin",
            "pm_password":"p@55w0rd!",
            "pm_addr":"192.0.2.205"
        },
        {
            "mac":[
                "cc:cc:cc:cc:cc:cc"
            ],
            "cpu":"4",
            "memory":"6144",
            "disk":"40",
            "arch":"x86_64",
            "pm_type":"pxe_ipmitool",
            "pm_user":"admin",
            "pm_password":"p@55w0rd!",
            "pm_addr":"192.0.2.206"
        },
        {
            "mac":[
                "dd:dd:dd:dd:dd:dd"
            ],
            "cpu":"4",
            "memory":"6144",
            "disk":"40",
            "arch":"x86_64",
            "pm_type":"pxe_ipmitool",
            "pm_user":"admin",
```

```
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.0.2.207"
      },
      {
        "mac":[
          "ee:ee:ee:ee:ee:ee"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"pxe_ipmitool",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.0.2.208"
      }
      {
        "mac":[
          "ff:ff:ff:ff:ff:ff"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"pxe_ipmitool",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.0.2.209"
      }
      {
        "mac":[
          "gg:gg:gg:gg:gg:gg"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"pxe_ipmitool",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.0.2.210"
      }
    ]
}
```

Procedure

1. After you create the inventory file, save the file to the home directory of the stack user (/**home/stack/instackenv.json**).

2. Initialize the stack user, then import the **instackenv.json** inventory file into director:

```
$ source ~/stackrc
$ openstack overcloud node import ~/instackenv.json
```

The **openstack overcloud node import** command imports the inventory file and registers each node with the director.

3. Assign the kernel and ramdisk images to each node:

```
$ openstack overcloud node configure <node>
```

Result

The nodes are registered and configured in director.

2.4. MANUALLY TAGGING THE NODES

After you register each node, you must inspect the hardware and tag the node into a specific profile. Use profile tags to match your nodes to flavors, and then assign flavors to deployment roles.

To inspect and tag new nodes, complete the following steps:

1. Trigger hardware introspection to retrieve the hardware attributes of each node:

```
$ openstack overcloud node introspect --all-manageable --provide
```

- The **--all-manageable** option introspects only the nodes that are in a managed state. In this example, all nodes are in a managed state.
- The **--provide** option resets all nodes to an **active** state after introspection.

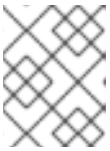
IMPORTANT

Ensure that this process completes successfully. This process usually takes 15 minutes for bare metal nodes.

2. Retrieve a list of your nodes to identify their UUIDs:

```
$ openstack baremetal node list
```

3. Add a profile option to the **properties/capabilities** parameter for each node to manually tag a node to a specific profile. The addition of the **profile** option tags the nodes into each respective profile.



NOTE

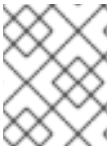
As an alternative to manual tagging, use the Automated Health Check (AHC) Tools to automatically tag larger numbers of nodes based on benchmarking data.

For example, to tag three nodes to use the **control** profile and another three nodes to use the **compute** profile, enter:

```
$ ironic node-update 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 6faba1a9-e2d8-4b7c-95a2-c7fbdc12129a add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 5e3b2f50-fcd9-4404-b0a2-59d79924b38e add
properties/capabilities='profile:control,boot_option:local'
```

```
$ ironic node-update 484587b2-b3b3-40d5-925b-a26a2fa3036f add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d010460b-38f2-4800-9cc4-d69f0d067efe add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d930e613-3e14-44b9-8240-4f3559801ea6 add
properties/capabilities='profile:compute,boot_option:local'
```

The addition of the **profile** option tags the nodes into each respective profiles.



NOTE

As an alternative to manual tagging, use the Automated Health Check (AHC) Tools to automatically tag larger numbers of nodes based on benchmarking data.

CHAPTER 3. INTEGRATING WITH THE EXISTING CEPH STORAGE CLUSTER

The Red Hat OpenStack Platform director uses **ceph-ansible** to integrate with an existing Ceph Storage cluster, but **ceph-ansible** is not installed by default on the undercloud.

3.1. INSTALLING THE CEPH-ANSIBLE PACKAGE

Run the following command to install the **ceph-ansible** package on the undercloud:

```
sudo dnf install -y ceph-ansible
```

3.2. CREATING A CUSTOM ENVIRONMENT FILE

Director supplies parameters to **ceph-ansible** to integrate with an external Ceph Storage cluster through the environment file:

- **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml**

If you are deploying the Shared File System with CephFS via NFS, the director also supplies parameters through another environment file:

- **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml**

These environment files are invoked during deployment (Section 3.6, "Deploying the overcloud") to integrate an existing Ceph Storage cluster with the overcloud being deployed.

To configure integration, you must supply the details of your Ceph Storage cluster to the director. To do this, use a custom environment file, which allows you to override the default settings supplied by these environment files.

Procedure

1. Create a custom environment file: **/home/stack/templates/ceph-config.yaml**

2. Add a **parameter_defaults:** header to the file:

   ```
   parameter_defaults:
   ```

3. Under this header, set all of the parameters that you want to override in **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml**. At a minimum, you must set:

   - **CephClientKey**: the Ceph client key of your Ceph Storage cluster. This is the value of **key** you retrieved earlier in Section 2.1, "Configuring the existing Ceph Storage cluster" (for example, **AQDLOh1VgEp6FRAAFzT7Zw+Y9V6JJExQAsRnRQ==**).

   - **CephClusterFSID**: the file system ID of your Ceph Storage cluster. This is the value of **fsid** in your Ceph Storage cluster configuration file, which you retrieved earlier in Section 2.1, "Configuring the existing Ceph Storage cluster" (for example, **4b5c8c0a-ff60-454b-a1b4-**

9747aa737d19).

- **CephExternalMonHost**: a comma-delimited list of the IPs of all MON hosts in your Ceph Storage cluster (for example, **172.16.1.7, 172.16.1.8**).
  For example:

```
parameter_defaults:
  CephClientKey: AQDLOh1VgEp6FRAAFzT7Zw+Y9V6JJExQAsRnRQ==
  CephClusterFSID: 4b5c8c0a-ff60-454b-a1b4-9747aa737d19
  CephExternalMonHost: 172.16.1.7, 172.16.1.8
```

4. If necessary, override the default pool names or the name of the OpenStack Platform client user to match your Ceph Storage cluster:

   - **CephClientUserName: openstack**
   - **NovaRbdPoolName: vms**
   - **CinderRbdPoolName: volumes**
   - **GlanceRbdPoolName: images**
   - **CinderBackupRbdPoolName: backups**
   - **GnocchiRbdPoolName: metrics**

5. If you are deploying the Shared File System backed by CephFS, set the names of the data and metadata pools:

```
ManilaCephFSDataPoolName: manila_data
ManilaCephFSMetadataPoolName: manila_metadata
```

NOTE

Make sure these names match the names of the pools you created earlier.

6. Set the client key that you created for manila and the name of the Ceph user for that key:

```
ManilaCephFSCephFSAuthId: 'manila'
CephManilaClientKey: 'AQDQ991cAAAAABAA0aXFrTnjH9aO39P0iVvYyg=='
```

NOTE

By default this key belongs to a user called **manila** and is stored as the **client.manila** key.

7. You can also add overcloud parameters to your custom environment file. For example, to set **vxlan** as the **neutron** network type, add the following to **parameter_defaults**:

```
NeutronNetworkType: vxlan
```

After you create the custom environment file, you must include it when you deploy the overcloud. For more information about deploying the overcloud, see Deploying the overcloud.

3.3. COMPATIBILITY WITH OLDER VERSIONS OF RED HAT CEPH STORAGE

If you are integrating Red Hat OpenStack Platform (RHOSP) with an external Ceph Storage Cluster from an earlier version, you might need to enable backwards compatibility.

If you are running Red Hat Ceph Storage 1.3, you must add the following line to **parameter_defaults** in your custom environment file, in this case, **/home/stack/templates/ceph-config** from Chapter 3, *Integrating with the existing Ceph Storage cluster* :

```
parameter_defaults:
  RbdDefaultFeatures: 1
```

If you are running an external Red Hat Ceph Storage 2.x cluster, then it is not necessary to use this parameter. Although this version of RHOSP uses Red Hat Ceph Storage 3.x clients, those clients remain compatible with a Red Hat Ceph Storage 2.x server.

3.4. ASSIGNING NODES AND FLAVORS TO ROLES

Planning an overcloud deployment involves specifying how many nodes and which flavors to assign to each role. Like all Heat template parameters, these role specifications are declared in the **parameter_defaults** section of your custom environment file (in this case, **/home/stack/templates/ceph-config** from Chapter 3, *Integrating with the existing Ceph Storage cluster*):

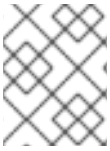
For this purpose, use the following parameters:

Table 3.1. Roles and Flavors for Overcloud Nodes

Heat Template Parameter	Description
ControllerCount	The number of Controller nodes to scale out
OvercloudControlFlavor	The flavor to use for Controller nodes (**control**)
ComputeCount	The number of Compute nodes to scale out
OvercloudComputeFlavor	The flavor to use for Compute nodes (**compute**)

For example, to configure the overcloud to deploy three nodes for each role (Controller and Compute), add the following to your **parameter_defaults**:

```
parameter_defaults:
  ControllerCount: 3
  ComputeCount: 3
  OvercloudControlFlavor: control
  OvercloudComputeFlavor: compute
```



NOTE

See Creating the Overcloud with the CLI Tools from the Director Installation and Usage guide for a more complete list of Heat template parameters.

3.5. CEPH CONTAINERS FOR RED HAT OPENSTACK PLATFORM WITH CEPH STORAGE

A Ceph container is required to configure OpenStack Platform to use Ceph, even with an external Ceph cluster. To be compatible with Red Hat Enterprise Linux 8, OpenStack Platform 15 requires Red Hat Ceph Storage 4. The Ceph Storage 4 container is hosted at registry.redhat.io, a registry which requires authentication.

You can use the heat environment parameter **ContainerImageRegistryCredentials** to authenticate at **registry.redhat.io**, as described in Container image preparation parameters.

3.6. DEPLOYING THE OVERCLOUD

NOTE

During undercloud installation, set **generate_service_certificate=false** in the **undercloud.conf** file. Otherwise, you must inject a trust anchor when you deploy the overcloud. For more information about how to inject a trust anchor, see Enabling SSL/TLS on Overcloud Public Endpoints in the *Advanced Overcloud Customization* guide.

The creation of the overcloud requires additional arguments for the **openstack overcloud deploy** command. For example:

```
$ openstack overcloud deploy --templates \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml \
  -e /home/stack/templates/ceph-config.yaml \
  -e --ntp-server pool.ntp.org \
```

The above command uses the following options:

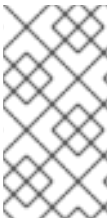
- **--templates** - Creates the overcloud from the default Heat template collection (namely, **/usr/share/openstack-tripleo-heat-templates/**).
- **-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml** - Sets the director to integrate an existing Ceph cluster to the overcloud.
- **-e /home/stack/templates/ceph-config.yaml** - Adds a custom environment file to override the defaults set by **-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml**. In this case, it is the custom environment file you created in Chapter 3, *Integrating with the existing Ceph Storage cluster* .
- **--ntp-server pool.ntp.org** - Sets the NTP server.

As noted earlier, deployment of the overcloud with the Shared File System backed by CephFS via NFS requires an additional **manila-cephfsganesha-config.yaml** environment file.

Additionally, you must deploy a custom Controller role to run the Ganesha CephFS to NFS gateway and to set up an isolated StorageNFS network to deliver shares to clients.

In this configuration, the **openstack overcloud deploy** command has the following form:

```
$ openstack overcloud deploy --templates \
  -n /usr/share/openstack-tripleo-heat-templates/network_data_ganesha.yaml \
  -r /home/stack/custom_roles.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-
external.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml \
  -e /home/stack/templates/ceph-config.yaml \
  -e --ntp-server pool.ntp.org
```



NOTE

The custom **ceph-config.yaml** environment file overrides parameters in the **ceph-ansible-external.yaml** file and the **manila-cephfsganesha-config.yaml** file. Therefore, include the custom **ceph-config.yaml** environment file in the deployment command after **ceph-ansible-external.yaml** and **manila-cephfsganesha-config.yaml**.

For information about the StorageNFS network and the custom Controller role required to deploy the Shared File System backed by CephFS via NFS, see Deploying the updated environment.

TIP

You can also use an *answers file* to invoke all your templates and environment files. For example, you can use the following command to deploy an identical overcloud:

```
$ openstack overcloud deploy \
  --answers-file /home/stack/templates/answers.yaml \
  --ntp-server pool.ntp.org
```

In this case, the answers file **/home/stack/templates/answers.yaml** contains:

```
templates: /usr/share/openstack-tripleo-heat-templates/
environments:
  - /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml
\
  - /home/stack/templates/ceph-config.yaml \
```

See Including environment files in an overcloud deployment for more details.

For a full list of options, run:

```
$ openstack help overcloud deploy
```

For more information, see Configuring a basic overcloud with the CLI tools in the *Director Installation and Usage* guide.

The overcloud creation process begins and the director provisions your nodes. This process takes some time to complete. To view the status of the Overcloud creation, open a separate terminal as the **stack** user and run:

```
$ source ~/stackrc
$ openstack stack list --nested
```

This configures the overcloud to use your external Ceph Storage cluster. You manage this cluster independently from the overcloud. For example, you scale the Ceph Storage cluster by using the Ceph management tools and not through the Red Hat OpenStack Platform director.

CHAPTER 4. ACCESSING THE OVERCLOUD

The director generates a script to configure and help authenticate interactions with your overcloud from the undercloud. The director saves this file (**overcloudrc**) in your **stack** user's home directory. Run the following command to use this file:

```
$ source ~/overcloudrc
```

This loads the necessary environment variables to interact with your overcloud from the undercloud CLI. To return to interacting with the undercloud, run the following command:

```
$ source ~/stackrc
```