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: 2022-06-10

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

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

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message . :leveloffset: +0

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Tell us how we can make it better.

Using the Direct Documentation Feedback (DDF) function

Use the **Add Feedback** DDF function for direct comments on specific sentences, paragraphs, or code blocks.

1. View the documentation in the *Multi-page HTML* format.
2. Ensure that you see the **Feedback** button in the upper right corner of the document.
3. Highlight the part of text that you want to comment on.
4. Click **Add Feedback**.
5. Complete the **Add Feedback** field with your comments.
6. Optional: Add your email address so that the documentation team can contact you for clarification on your issue.
7. Click **Submit**.

CHAPTER 1. INTEGRATING AN OVERCLOUD WITH CEPH STORAGE

Red Hat OpenStack Platform director creates a cloud environment called the overcloud. You can use director to configure extra features for an overcloud, such as integration with Red Hat Ceph Storage. You can integrate your overcloud with Ceph Storage clusters created with director or with existing Ceph Storage clusters.

This guide includes instructions about how to integrate an existing Ceph Storage cluster with an overcloud. This means that 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.

1.1. ABOUT CEPH STORAGE

Red Hat Ceph Storage is a distributed data object store that provides performance, reliability, and scalability. Distributed object stores accommodate unstructured data, and 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. The following are the primarily two types of daemons:

Ceph OSD (Object Storage Daemon)

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

Ceph Monitor

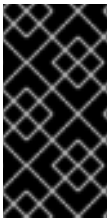
A Ceph monitor maintains a master copy of the Ceph storage cluster map with the current state of the storage cluster.

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

1.2. DEPLOY THE SHARED FILE SYSTEMS SERVICE WITH EXTERNAL CEPHFS

Red Hat OpenStack Platform director can deploy the Shared File Systems service (manila) with CephFS. CephFS can be consumed either through the native CephFS protocol or through the NFS protocol.

For more information about these storage protocols, see Ceph File System architecture in *CephFS Back End Guide for the Shared Files Systems Service*.



IMPORTANT

You cannot use the Shared File Systems service (manila) with the CephFS native driver to serve shares to Red Hat OpenShift Container Platform through Manila CSI. Red Hat does not support this type of deployment. For more information, contact Red Hat Support.

IMPORTANT

The Shared File Systems service (manila) with CephFS through NFS fully supports serving shares to Red Hat OpenShift Container Platform through Manila CSI. This solution is not intended for large scale deployments. For important recommendations, see https://access.redhat.com/articles/6667651.

IMPORTANT

To use native CephFS shared file systems, clients require access to the Ceph public network. When you integrate an overcloud with an existing Ceph cluster, director does not create an isolated Storage network to designate as the Ceph public network. This network is assumed to already exist. Do not provide direct access to the Ceph public network, instead, allow tenants to create a router to connect to the Ceph public network.

For more information about security considerations, see Security considerations in the *CephFS Back End Guide for the Shared Files Systems Service* guide.

When you use CephFS through the NFS protocol, director deploys the NFS-Ganesha gateway on Controller nodes managed by Pacemaker (PCS). PCS manages cluster availability by using an active-passive configuration.

NOTE

This feature is supported with Ceph Storage 4.1 or later in the Ceph 4 cycle or Ceph Storage 5.0 or later in the Ceph 5 cycle. 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 Section 3.1, "Installing the ceph-ansible package" .

Prerequisites

The following prerequisites are required to configure the Shared File Systems service 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 Section 3.2, "Creating a custom environment file".
- The external Ceph Storage cluster must have a **cephx** client name and key for the Shared File Systems service. For more information, see Section 3.2, "Creating a custom environment file".

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

1.3. CONFIGURE CEPH OBJECT STORE TO USE EXTERNAL CEPH OBJECT GATEWAY

Red Hat OpenStack Platform (RHOSP) director supports configuring an external Ceph Object Gateway (RGW) as an Object Store service. To authenticate with the external RGW service, you must configure RGW to verify users and their roles in the Identity service (keystone).

For more information about how to configure an external Ceph Object Gateway, see Configuring the Ceph Object Gateway to use Keystone authentication in the *Using Keystone with the Ceph Object Gateway Guide*.

CHAPTER 2. PREPARING OVERCLOUD NODES

The overcloud deployed in this scenario consists of six nodes:

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

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 with the Ceph management tools, not through director. For more information, see the Red Hat Ceph Storage documentation library.

2.1. PRE-DEPLOYMENT VALIDATIONS FOR CEPH STORAGE

To help avoid overcloud deployment failures, verify that the required packages exist on your servers.

2.1.1. Verifying the ceph-ansible package version

The undercloud contains Ansible-based validations that you can run to identify potential problems before you deploy the overcloud. These validations can help you avoid overcloud deployment failures by identifying common problems before they happen.

Procedure

Verify that the correction version of the **ceph-ansible** package is installed:

```
$ ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/ansible/validation-playbooks/ceph-ansible-installed.yaml
```

2.1.2. Verifying packages for pre-provisioned nodes

Ceph can service only overcloud nodes that have a certain set of packages. When you use pre-provisioned nodes, you can verify the presence of these packages.

For more information about pre-provisioned nodes, see Configuring a basic overcloud with pre-provisioned nodes.

Procedure

Verify that the servers contained the required packages:

```
ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/ansible/validation-playbooks/ceph-dependencies-installed.yaml
```

2.2. CONFIGURING THE EXISTING CEPH STORAGE CLUSTER

Create OSD pools, define capabilities, and create keys and IDs for your Ceph Storage cluster.

Procedure

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 Systems (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. Approximately 100 placement groups per OSD is the best practice. 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 Systems service 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* .

Use the Ceph client key and file system ID; and the manila client IDS and key in the following procedure: Section 3.1, “Installing the ceph-ansible package” .

2.3. INITIALIZING THE STACK USER

Initialize the stack user to configure the authentication details used to access director CLI tools.

Procedure

1. Log in to the director host as the **stack** user.
2. Enter the following command to initialize your director configuration:

   ```
   $ source ~/stackrc
   ```

2.4. REGISTERING NODES

An inventory file contains hardware and power management details about nodes. Create an inventory file to configure and register nodes in director.

Procedure

1. Create an inventory file. Use the example node definition template, **instackenv.json** as a reference:

```
{
    "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"
        }
    ]
}
```

2. Save the file to the home directory of the stack user: **/home/stack/instackenv.json**.

3. 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.

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

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

Result

The nodes are registered and configured in director.

2.5. MANUALLY TAGGING NODES

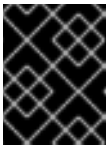
After you register each node, you must inspect the hardware and tag the node into a specific profile. Use profile tags to match your nodes to flavors and then assign flavors to deployment roles.

Procedure

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.
As an alternative to manual tagging, you can configure 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, create the following **profile** options:

```
$ 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'
```

CHAPTER 3. INTEGRATE WITH AN EXISTING CEPH STORAGE CLUSTER

To integrate the Red Hat OpenStack Platform with an existing Ceph Storage cluster, you must install the **ceph-ansible** package. After that, you can create custom environment files and assign nodes and flavors to roles.

3.1. INSTALLING THE CEPH-ANSIBLE PACKAGE

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.

Procedure

Enter 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 deploy the Shared File Systems service with external CephFS, separate environment files supply additional parameters.

- For native CephFS the environment file is **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml**.
- For CephFS through NFS, the environment file is **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml**.

Director invokes these environment files during deployment to integrate an existing Ceph Storage cluster with the overcloud. For more information, Section 3.5, “Deploying the overcloud”.

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

Procedure

1. Create a custom environment file: **/home/stack/templates/ceph-config.yaml**
2. Add a **parameter_defaults:** section to the file:

   ```
   parameter_defaults:
   ```

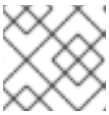
3. Use **parameter_defaults** to set all of the parameters that you want to override in **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml**. You must set the following parameters at a minimum:

- **CephClientKey**: the Ceph client key of your Ceph Storage cluster. This is the value of **key** you retrieved in Section 2.2, "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 in Section 2.2, "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 Red Hat 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 Systems service backed by CephFS, set the names of the data and metadata pools:

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



NOTE

Ensure that these names match the names of the pools you created.

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

The default client user name **ManilaCephFSCephFSAuthId** is **manila**, unless you overide it. **CephManilaClientKey** is always required.

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 Section 3.5, "Deploying the overcloud".

3.3. 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**.

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 **parameter_defaults**:

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

NOTE

For more information and a more complete list of heat template parameters, see Creating the Overcloud with the CLI Tools in the *Director Installation and Usage* guide.

3.4. 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, Red Hat OpenStack Platform (RHOSP) 15 requires Red Hat Ceph Storage 4. The Ceph Storage 4 container is hosted at registry.redhat.io, a registry that requires authentication.

You can use the heat environment parameter **ContainerImageRegistryCredentials** to authenticate at **registry.redhat.io**. For more information, see Container image preparation parameters.

3.5. 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.

Procedure

- The creation of the overcloud requires additional arguments for the **openstack overcloud deploy** command:

```
$ 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 \
```

This example command uses the following options:

- **--templates** - Creates the overcloud from the default heat template collection, **/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 Section 3.1, “Installing the ceph-ansible package” .
- **--ntp-server pool.ntp.org** - Sets the NTP server.

3.5.1. Adding an additional environment file for the Shared File Systems service backed by CephFS

If you deploy an overcloud that uses a the Shared File Systems service backed by CephFS, you must add an additional environment file.

Procedure

1. Use one of the following options to create and add an additional environment file:
   - If you deploy an overcloud that uses the native CephFS back end driver, use **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml**.

- If you deploy an overcloud that uses CephFS through NFS, use **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml**.
  For CephFS through NFS, you must also deploy a custom Controller role to run the Ganesha CephFS to NFS gateway. This role also configures isolated StorageNFS network to deliver shares to clients. For more information about the StorageNFS network and the custom Controller role, see Deploying the updated environment in the *CephFS via NFS Back End Guide for the Shared File Systems Service*.

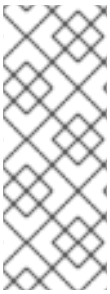
2. Alter the form of the **openstack overcloud deploy** command depending on the CephFS back end that you use.

   - For native CephFS:

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

   - For CephFS through NFS:

```
$ 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 either the **manila-cephfsnative-config.yaml** file or 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 either **manila-cephfsnative-config.yaml** or **manila-cephfsganesha-config.yaml**.

Example environment file

```
parameter_defaults:
  CinderEnableIscsiBackend: false
  CinderEnableRbdBackend: true
  CinderEnableNfsBackend: false
  NovaEnableRbdBackend: true
  GlanceBackend: rbd
  CinderRbdPoolName: "volumes"
  NovaRbdPoolName: "vms"
  GlanceRbdPoolName: "images"
  CinderBackupRbdPoolName: "backups"
```

```
GnocchiRbdPoolName: "metrics"
CephClusterFSID: <cluster_ID>
CephExternalMonHost: <IP_address>,<IP_address>,<IP_address>
CephClientKey: "<client_key>"
CephClientUserName: "openstack"
ManilaCephFSDataPoolName: manila_data
ManilaCephFSMetadataPoolName: manila_metadata
ManilaCephFSCephFSAuthId: 'manila'
CephManilaClientKey: '<client_key>'
ExtraConfig:
  ceph::profile::params::rbd_default_features: '1'
```

- Replace the variables **<cluster_ID>**, **<IP_address>**, and **<client_key>** with values that are suitable for your environment.

3.5.2. Adding an additional environment file for external Ceph Object Gateway (RGW) for Object storage

If you deploy an overcloud that uses an already existing RGW service for Object storage, you must add an additional environment file.

Procedure

1. Add the following **parameter_defaults** to a custom environment file, for example, **swift-external-params.yaml**. Replace the values to suit your deployment:

```
parameter_defaults:
  ExternalSwiftPublicUrl: 'http://<Public RGW endpoint or
loadbalancer>:8080/swift/v1/AUTH_%(project_id)s'
  ExternalSwiftInternalUrl: 'http://<Internal RGW endpoint>:8080/swift/v1/AUTH_%
(project_id)s'
  ExternalSwiftAdminUrl: 'http://<Admin RGW endpoint>:8080/swift/v1/AUTH_%(project_id)s'
  ExternalSwiftUserTenant: 'service'
  SwiftPassword: 'choose_a_random_password'
```

NOTE

The example code snippet contains parameter values that might differ from values that you use in your environment:

- The default port where the remote RGW instance listens is **8080**. The port might be different depending on how the external RGW is configured.
- The **swift** user created in the overcloud uses the password defined by the **SwiftPassword** parameter. You must configure the external RGW instance to use the same password to authenticate with the Identity service by using the **rgw_keystone_admin_password**.

2. Add the following code to the Ceph config file to configure RGW to use the Identity service. Replace the variable values to suit your environment:

```
rgw_keystone_api_version = 3
rgw_keystone_url = http://<public Keystone endpoint>:5000/
rgw_keystone_accepted_roles = member, Member, admin
```

```
    rgw_keystone_accepted_admin_roles = ResellerAdmin, swiftoperator
    rgw_keystone_admin_domain = default
    rgw_keystone_admin_project = service
    rgw_keystone_admin_user = swift
    rgw_keystone_admin_password =
<password_as_defined_in_the_environment_parameters>
    rgw_keystone_implicit_tenants = true
    rgw_keystone_revocation_interval = 0
    rgw_s3_auth_use_keystone = true
    rgw_swift_versioning_enabled = true
    rgw_swift_account_in_url = true
```

NOTE

Director creates the following roles and users in the Identity service by default:

- rgw_keystone_accepted_admin_roles: ResellerAdmin, swiftoperator
- rgw_keystone_admin_domain: default
- rgw_keystone_admin_project: service
- rgw_keystone_admin_user: swift

3. Deploy the overcloud with the additional environment files with any other environment files that are relevant to your deployment::

```
openstack overcloud deploy --templates \
-e <your_environment_files>
-e /usr/share/openstack-tripleo-heat-templates/environments/swift-external.yaml
-e swift-external-params.yaml
```

3.5.3. Invoking templates and environment files

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 \
```

For more information, see Including environment files in an overcloud deployment in the *Director Installation and Usage* guide.

3.5.4. OpenStack overcloud deploy command options

You can enter the following command to see a full list of options that are available to use with the **openstack overcloud deploy** command:

```
$ openstack help overcloud deploy
```

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

3.5.5. Viewing the status of overcloud creation

The overcloud creation process begins and director provisions your nodes. This process takes some time to complete.

Procedure

To view the status of the overcloud creation, open a separate terminal as the **stack** user and enter the following commands:

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

CHAPTER 4. VERIFY EXTERNAL CEPH STORAGE CLUSTER INTEGRATION

After you deploy the overcloud, confirm that Red Hat OpenStack Platform (RHOSP) services can write to the Ceph Storage cluster.

WARNING

RHOSP does not support the use of Ceph clone format v2 or later. Deleting images or volumes from a Ceph cluster that has Ceph clone format v2 enabled might cause unpredictable behavior and potential loss of data. Therefore, do not use either of the following methods that enable Ceph clone format v2:

- Setting **rbd default clone format = 2**
- Running **ceph osd set-require-min-compat-client mimic**

4.1. GATHERING IDS

To verify that you integrated a Ceph Storage cluster, you must first create an image, a Compute instance, and a volume and gather their respective IDs.

Procedure

1. Create an image with the Image service (glance).
   For more information about how to create an image, see Import an image in the *Creating and Managing Images* guide.
2. Record the glance image ID for later use.
3. Create a Compute (nova) instance.
   For more information about how to create an instance, see Launch an instance in the *Creating and Managing Instances* guide.
4. Record the nova instance ID for later use.
5. Create a Block Storage (cinder) volume.
   For more information about how to create a Block Storage volume, see Create a volume in the *Storage Guide*.
6. Record the cinder volume ID for later use.

4.2. VERIFYING THE CEPH STORAGE CLUSTER

When you configure an external Ceph Storage cluster, you create pools and a **client.openstack** user to access those pools. After you deploy the overcloud, you can use the file that contains the credentials of the **client.openstack** user to list the contents of Red Hat OpenStack Platform (RHOSP) pools.

List the contents of the pools and confirm that the IDs of the glance image, Compute instance, and cinder volume exist on the Ceph Storage cluster.

Procedure

1. Source the undercloud credentials:

```
[stack@undercloud-0 ~]$ source stackrc
```

2. List the available servers to retrieve the IP addresses of nodes on the system:

```
(undercloud) [stack@undercloud-0 ~]$ openstack server list

+--------------+---------------+--------------+
| ID | Name | Status | Networks | Image | Flavor |
+--------------+---------------+--------------+
| d5a621bd-d109-41ae-a381-a42414397802 | compute-0 | ACTIVE | ctlplane=192.168.24.31
| overcloud-full | compute |
| 496ab196-d6cb-447d-a118-5bafc5166cf2 | controller-0 | ACTIVE | ctlplane=192.168.24.37 |
overcloud-full | controller |
| c01e730d-62f2-426a-a964-b31448f250b3 | controller-2 | ACTIVE | ctlplane=192.168.24.55 |
overcloud-full | controller |
| 36df59b3-66f3-452e-9aec-b7e7f7c54b86 | controller-1 | ACTIVE | ctlplane=192.168.24.39 |
overcloud-full | controller |
| f8f00497-246d-4e40-8a6a-b5a60fa66483 | compute-1 | ACTIVE | ctlplane=192.168.24.10 |
overcloud-full | compute |
```

3. Use SSH to log in to any Compute node:

```
(undercloud) [stack@undercloud-0 ~]$ ssh heat-admin@192.168.24.31
```

4. Switch to the root user:

```
[heat-admin@compute-0 ~]$ sudo su -
```

5. Confirm that the files **/etc/ceph/ceph.conf** and **/etc/ceph/ceph.client.openstack.keyring** exist:

```
[root@compute-0 ~]# ls -l /etc/ceph/ceph.conf

-rw-r--r--. 1 root root 1170 Sep 29 23:25 /etc/ceph/ceph.conf
[root@compute-0 ~]# ls -l /etc/ceph/ceph.client.openstack.keyring

-rw-------. 1 ceph ceph 253 Sep 29 23:25 /etc/ceph/ceph.client.openstack.keyring
```

6. Enter the following command to force the **nova_compute** container to use the **rbd** command to list the contents of the appropriate pool.

```
# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls vms
```

The pool name must match the pool names of the images, VMs, and volumes that you created when you configured the Ceph Storage cluster. For more information, see Configuring the existing Ceph Storage cluster. The IDs of the image, Compute instance, and volume must match

the IDs that you recorded in Section 4.1, "Gathering IDs".

NOTE

The example command is prefixed with **podman exec nova_compute** because **/usr/bin/rbd**, which is provided by the ceph-common package, is not installed on overcloud nodes by default. However, it is available in the **nova_compute** container. The command lists block device images. For more information, see Listing the block device images in the *Ceph Storage Block Device Guide* .

The following examples show how to confirm whether an ID for each service is present for each pool by using the IDs from Section 4.1, "Gathering IDs".

```
# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls images | grep
4485d4c0-24c3-42ec-a158-4d3950fa020b

# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls vms | grep
64bcb731-e7a4-4dd5-a807-ee26c669482f

# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls volumes | grep
aeac15e8-b67f-454f-9486-46b3d75daff4
```

4.3. TROUBLESHOOTING FAILED VERIFICATION

If the verification procedures fail, verify that the Ceph key for the **openstack.client** user and the Ceph Storage monitor IPs or hostnames can be used together to read, write, and delete from the Ceph Storage pools that you created for the Red Hat OpenStack Platform (RHOSP).

Procedure

1. To shorten the amount of typing you must do in this procedure, log in to a Compute node and create an alias for the **rbd** command:

```
# alias rbd="podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack"
```

2. Confirm that you can write test data to the pool as a new object:

```
# rbd create --size 1024 vms/foo
```

3. Confirm that you can see the test data:

```
# rbd ls vms | grep foo
```

4. Delete the test data:

```
# rbd rm vms/foo
```

NOTE

If this procedure fails, contact your Ceph Storage administrator for assistance. If this procedure succeeds, but you cannot create Compute instances, glance images, or cinder volumes, contact Red Hat Support.

CHAPTER 5. ACCESSING THE OVERCLOUD

Director generates a script to configure and authenticate interactions with your overcloud from the undercloud. Director saves this file, **overcloudrc**, in the **stack** user home directory.

Procedure

1. Enter the following command to use the **overcloudrc** file:

```
$ source ~/overcloudrc
```

   This loads the environment variables that are required to interact with your overcloud from the undercloud CLI.

2. To return to to the undercloud, enter the following command:

```
$ source ~/stackrc
```