



Red Hat OpenStack Platform 16.2

Integrating an Overcloud with an Existing Red Hat Ceph Storage Cluster

Configuring an overcloud to use standalone Red Hat Ceph Storage

Red Hat OpenStack Platform 16.2 Integrating an Overcloud with an Existing Red Hat Ceph Storage Cluster

Configuring an overcloud to use standalone Red Hat Ceph Storage

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Copyright © 2024 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

You can use Red Hat OpenStack Platform (RHOSP) director to integrate an overcloud with an existing, standalone Red Hat Ceph Storage cluster.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	3
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	4
CHAPTER 1. INTEGRATING AN OVERCLOUD WITH CEPH STORAGE	5
1.1. RED HAT CEPH STORAGE COMPATIBILITY	5
1.2. DEPLOYING THE SHARED FILE SYSTEMS SERVICE WITH EXTERNAL CEPHFS	5
NFS-Ganesha gateway	5
1.3. CONFIGURING CEPH OBJECT STORE TO USE EXTERNAL CEPH OBJECT GATEWAY	6
CHAPTER 2. PREPARING OVERCLOUD NODES	7
2.1. VERIFYING AVAILABLE RED HAT CEPH STORAGE PACKAGES	7
2.1.1. Verifying the ceph-ansible package version	7
2.1.2. Verifying packages for pre-provisioned nodes	7
2.2. CONFIGURING THE EXISTING RED HAT CEPH STORAGE CLUSTER	7
CHAPTER 3. INTEGRATING WITH AN EXISTING CEPH STORAGE CLUSTER	10
3.1. INSTALLING THE CEPH-ANSIBLE PACKAGE	10
3.2. CREATING A CUSTOM ENVIRONMENT FILE	10
3.3. CEPH CONTAINERS FOR RED HAT OPENSTACK PLATFORM WITH CEPH STORAGE	12
3.4. DEPLOYING THE OVERCLOUD	12
3.4.1. Adding environment files for the Shared File Systems service with CephFS	12
3.4.2. Adding an additional environment file for external Ceph Object Gateway (RGW) for Object storage	14
CHAPTER 4. VERIFYING EXTERNAL RED HAT CEPH STORAGE CLUSTER INTEGRATION	16
4.1. GATHERING IDS	16
4.2. VERIFYING THE RED HAT CEPH STORAGE CLUSTER	16
4.3. TROUBLESHOOTING FAILED VERIFICATION	18

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](#).

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

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

Providing documentation feedback in Jira

Use the [Create Issue](#) form to provide feedback on the documentation. The Jira issue will be created in the Red Hat OpenStack Platform Jira project, where you can track the progress of your feedback.

1. Ensure that you are logged in to Jira. If you do not have a Jira account, create an account to submit feedback.
2. Click the following link to open a the **Create Issue** page: [Create Issue](#)
3. Complete the **Summary** and **Description** fields. In the **Description** field, include the documentation URL, chapter or section number, and a detailed description of the issue. Do not modify any other fields in the form.
4. Click **Create**.

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.

For more information about Red Hat Ceph Storage, see the [Red Hat Ceph Storage Architecture Guide](#).

1.1. RED HAT CEPH STORAGE COMPATIBILITY

RHOSP 16.2 supports connection to external Red Hat Ceph Storage 4 and Red Hat Ceph Storage 5 clusters.

1.2. DEPLOYING THE SHARED FILE SYSTEMS SERVICE WITH EXTERNAL CEPHFS

You can deploy the Shared File Systems service (manila) with CephFS by using Red Hat OpenStack Platform (RHOSP) director. You can use the Shared File Systems service with the NFS protocol or the native CephFS protocol.



IMPORTANT

You cannot use the Shared File Systems service 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.

The Shared File Systems service 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 more information about CSI workload recommendations, see <https://access.redhat.com/articles/6667651>.

To use native CephFS shared file systems, clients require access to the Ceph public network. When you integrate an overcloud with an existing Ceph Storage 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.

NFS-Ganesha gateway

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.

The NFS-Ganesha gateway is supported with Red Hat Ceph Storage 4.x (Ceph package 14.x) and Red Hat Ceph Storage 5.x (Ceph package 16.x). For information about how to determine the Ceph Storage release installed on your system, see [Red Hat Ceph Storage releases and corresponding Ceph package versions](#).

You must install the latest version of the **ceph-ansible** package on the undercloud, as described in [Installing the ceph-ansible package](#).

Prerequisites

Before you configure the Shared File Systems service with an external Ceph Storage cluster, complete the following prerequisites:

- Verify that your external Ceph Storage cluster has an active Metadata Server (MDS):

```
$ ceph -s
```

- The external Ceph Storage cluster must have a CephFS file system that is supported by CephFS data and metadata pools.
 - Verify the pools in the CephFS file system:

```
$ ceph fs ls
```

- Note the names of these pools to configure the director parameters, **ManilaCephFSDataPoolName** and **ManilaCephFSMetadataPoolName**. For more information about this configuration, see [Creating a custom environment file](#).
- The external Ceph Storage cluster must have a **cephx** client name and key for the Shared File Systems service.
 - Verify the keyring:

```
$ ceph auth get client.<client name>
```

- Replace <client name> with your **cephx** client name.

1.3. CONFIGURING 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 deployment that is used to demonstrate how to integrate with a Red Hat Ceph Storage cluster consists of Controller nodes with high availability and Compute nodes to host workloads. The Red Hat Ceph Storage cluster has its own nodes that you manage independently from the overcloud by using the Ceph management tools, not through director. For more information about Red Hat Ceph Storage, see [Red Hat Ceph Storage](#).

2.1. VERIFYING AVAILABLE RED HAT CEPH STORAGE PACKAGES

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 **ceph-ansible** package version you want 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

Red Hat Ceph Storage (RHCS) 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 pre-provisioned nodes contain 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 RED HAT CEPH STORAGE CLUSTER

To configure your Red Hat Ceph Storage cluster, you create object storage daemon (OSD) pools, define capabilities, and create keys and IDs directly on the Ceph Storage cluster. You can execute commands from any machine that can reach the Ceph Storage cluster and has the Ceph command line client installed.

Procedure

1. Create the following pools in your Ceph Storage cluster, relevant to your environment:

- Storage for OpenStack Block Storage (cinder):

-

```
[root@ceph ~]# ceph osd pool create volumes <pgnum>
```

- Storage for OpenStack Image Storage (glance):

```
[root@ceph ~]# ceph osd pool create images <pgnum>
```

- Storage for instances:

```
[root@ceph ~]# ceph osd pool create vms <pgnum>
```

- Storage for OpenStack Block Storage Backup (cinder-backup):

```
[root@ceph ~]# ceph osd pool create backups <pgnum>
```

- Optional: Storage for OpenStack Telemetry Metrics (gnocchi):

```
[root@ceph ~]# ceph osd pool create metrics <pgnum>
```

Use this storage option only if metrics are enabled through OpenStack. If your overcloud deploys OpenStack Telemetry Metrics with CephFS, create CephFS data and metadata pools.

2. If your overcloud deploys the Shared File Systems service (manila) with Red Hat Ceph 4 (Ceph package 14) or earlier, 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 in the cluster, divided by the number of replicas (**osd pool default size**). For example, if there are 10 OSDs, and the cluster has the **osd pool default size** set to 3, use 333 placement groups. You can also use the [Ceph Placement Groups \(PGs\) per Pool Calculator](#) to determine a suitable value.

3. If your overcloud deploys the Shared File Systems service (manila) with Red Hat Ceph 5 (Ceph package 16) or later, you do not need to create data and metadata pools for CephFS. You can create a filesystem volume. For more information, see [Management of MDS service using the Ceph Orchestrator](#) in the Red Hat Ceph Storage *Operations Guide*.

4. 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,

```
[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
```

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

```
[root@ceph ~]# ceph auth list
...
[client.openstack]
key = <AQC+vYNXgDAgAhAAc8UoYt+OTz5uhV7ltLdwUw==>
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
...
```

The **key** value in the example, **AQC+vYNXgDAgAhAAc8UoYt+OTz5uhV7ltLdwUw==**, is your Ceph client key.

- If your overcloud deploys the Shared File Systems service with CephFS, create the **client.manila** user in your Ceph Storage 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

```
[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 *'
```

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

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

- Note the file system ID of your Ceph Storage cluster. This value is specified in the **fsid** field, under the **[global]** section of the configuration file for your cluster:

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



NOTE

Use the Ceph client key and file system ID, and the Shared File Systems service client IDs and key when you create the custom environment file.

Additional resources

- [Creating a custom environment file](#)
- [Red Hat Ceph Storage releases and corresponding Ceph package versions](#)
- [Ceph configuration](#) in the Red Hat Ceph Storage *Configuration Guide*.

CHAPTER 3. INTEGRATING WITH AN EXISTING CEPH STORAGE CLUSTER

To integrate Red Hat OpenStack Platform (RHOSP) with an existing Red Hat Ceph Storage cluster, you must install the **ceph-ansible** package. After that, you can create custom environment files to override and provide values for configuration options within OpenStack components.

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 Red Hat 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 (manila) 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**.

To configure integration of an existing Ceph Storage cluster with the overcloud, you must supply the details of your Ceph Storage cluster to director by using a custom environment file. Director invokes these environment files during deployment.

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 for the **client.openstack** user in your Ceph Storage cluster. This is the value of **key** you retrieved in [Configuring the existing Ceph Storage](#)

cluster. This is the value of `key` you retrieved in [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 [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>
```

- Optional: You can override the Red Hat OpenStack Platform (RHOSP) client username and the following default pool names to match your Ceph Storage cluster:

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

- Optional: If you are deploying the Shared File Systems service with CephFS, you can override the following default data and metadata pool names:

```
ManilaCephFSDataPoolName: <manila_data>
ManilaCephFSMetadataPoolName: <manila_metadata>
```

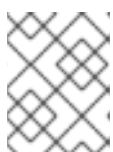


NOTE

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

- Set the client key that you created for the Shared File Systems service. You can override the default Ceph client username for that key:

```
ManilaCephFSCephFSAuthId: <manila>
CephManilaClientKey: <AQDQ991cAAAAABAA0aXFrTnjH9aO39P0iVvYyg==>
```



NOTE

The default client username `ManilaCephFSCephFSAuthId` is `manila`, unless you override it. `CephManilaClientKey` is always required.

After you create the custom environment file, you must include it when you deploy the overcloud.

Additional resources

- [Deploying the overcloud](#)

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

To configure Red Hat OpenStack Platform (RHOSP) to use Red Hat Ceph Storage with NFS Ganesha, you must have a Ceph container.

To be compatible with Red Hat Enterprise Linux 8, RHOSP 16 requires Red Hat Ceph Storage 4 or 5 (Ceph package 14.x or Ceph package 16.x). The Ceph Storage 4 and 5 containers are hosted at registry.redhat.io, a registry that requires authentication. For more information, see [Container image preparation parameters](#).

3.4. DEPLOYING THE OVERCLOUD

Deploy the overcloud with the environment file that you created.

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 [Installing the ceph-ansible package](#).
- **--ntp-server pool.ntp.org** - Sets the NTP server.

3.4.1. Adding environment files for the Shared File Systems service with CephFS

If you deploy an overcloud that uses the Shared File Systems service (manila) with CephFS, you must add additional environment files.

Procedure

1. Create and add additional environment files:

- If you deploy an overcloud that uses the native CephFS back-end driver, add **`/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml`**.
 - If you deploy an overcloud that uses CephFS through NFS, add **`/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml`**.
Red Hat recommends that you deploy the Ceph-through-NFS driver with an isolated StorageNFS network where shares are exported. You must deploy the isolated network to overcloud controller nodes. To enable this deployment, director includes the following file and role:
 - An example custom network configuration file that includes the StorageNFS network (**`/usr/share/openstack-tripleo-heat-templates/network_data_ganesha.yaml`**). Review and customize this file as necessary.
 - A ControllerStorageNFS role.
2. Modify 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:
  CinderEnableScsiBackend: 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 **<cluster_ID>**, **<IP_address>**, and **<client_key>** with values that are suitable for your environment.

Additional resources

- For more information about generating a custom roles file, see [Deploying the Shared File Systems service with CephFS through NFS](#).

3.4.2. 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**, and adjust 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
```

CHAPTER 4. VERIFYING EXTERNAL RED HAT CEPH STORAGE CLUSTER INTEGRATION

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



WARNING

RHOSP does not support the use of Ceph clone format v2 or later. Deleting images or volumes from a Ceph Storage 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 Red Hat Ceph Storage cluster, you must first create an image, a Compute instance, a Block Storage volume, and a file share 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 image ID for later use.
3. Create a Compute (nova) instance. For more information about how to create an instance, see [Creating an instance](#) in the *Creating and Managing Instances* guide.
4. Record the 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 volume ID for later use.
7. Create a file share by using the Shared File Systems service (manila). For more information about how to create a file share, see [Creating a share](#) in the *Storage Guide*.
8. List the export path of the share and record the UUID in the suffix for later use. For more information about how to list the export path of the share, see [Listing shares and exporting information](#) in the *Storage Guide*.

4.2. VERIFYING THE RED HAT CEPH STORAGE CLUSTER

When you configure an external Red Hat 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 Image service (glance) image, the Compute (nova) instance, the Block Storage (cinder) volume, and the Shared File Systems service (manila) file share exist on the Ceph Storage cluster.

Procedure

1. Log in to the undercloud as the **stack** user and source the **stackrc** credentials file:

```
$ source ~/stackrc
```

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

```
$ 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:

```
$ 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 **rbid** 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, volumes, and shares that you created when you configured the Ceph Storage cluster. The IDs of the image, Compute instance, volume, and share must match the IDs that you recorded in [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 about listing block device images, 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 [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
```

- To verify the existence of the Shared File Systems service share, you must log into a Controller node:

```
# podman exec openstack-manila-share-podman-0 ceph -n client.manila fs subvolume ls
cephfs | grep ec99db3c-0077-40b7-b09e-8a110e3f73c1
```

4.3. TROUBLESHOOTING FAILED VERIFICATION

If the verification procedures fail, verify that the Ceph key for the **openstack.client** user and the Red Hat 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

- 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"
```

- 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 (nova) instances, Image service (glance) images, Block Storage (cinder) volumes, or Shared File Systems service (manila) shares, contact Red Hat Support.