Red Hat Ceph Storage 3

Installation Guide for Ubuntu

Installing Red Hat Ceph Storage on Ubuntu

Last Updated: 2019-05-13

Red Hat Ceph Storage 3 Installation Guide for Ubuntu

Installing Red Hat Ceph Storage on Ubuntu

Legal Notice

Abstract

This document provides instructions on installing Red Hat Ceph Storage on Ubuntu 16.04 running on AMD64 and Intel 64 architectures.

Table of Contents

CHAPTER 1. WHAT IS RED HAT CEPH STORAGE?

Red Hat Ceph Storage is a scalable, open, software-defined storage platform that combines the most stable version of the Ceph storage system with a Ceph management platform, deployment utilities, and support services.

Red Hat Ceph Storage is designed for cloud infrastructure and web-scale object storage. Red Hat Ceph Storage clusters consist of the following types of nodes:

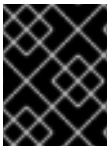
Red Hat Ceph Storage Ansible administration node

This type of node acts as the traditional Ceph Administration node did for previous versions of Red Hat Ceph Storage. This type of node provides the following functions:

- Centralized storage cluster management
- The Ceph configuration files and keys
- Optionally, local repositories for installing Ceph on nodes that cannot access the Internet for security reasons

Monitor nodes

Each monitor node runs the monitor daemon (**ceph-mon**), which maintains a master copy of the cluster map. The cluster map includes the cluster topology. A client connecting to the Ceph cluster retrieves the current copy of the cluster map from the monitor which enables the client to read from and write data to the cluster.



IMPORTANT

Ceph can run with one monitor; however, to ensure high availability in a production cluster, Red Hat will only support deployments with at least three monitor nodes.

OSD nodes

Each Object Storage Device (OSD) node runs the Ceph OSD daemon (**ceph-osd**), which interacts with logical disks attached to the node. Ceph stores data on these OSD nodes.
Ceph can run with very few OSD nodes, which the default is three, but production clusters realize better performance beginning at modest scales, for example 50 OSDs in a storage cluster. Ideally, a Ceph cluster has multiple OSD nodes, allowing isolated failure domains by creating the CRUSH map.

MDS nodes

Each Metadata Server (MDS) node runs the MDS daemon (**ceph-mds**), which manages metadata related to files stored on the Ceph File System (CephFS). The MDS daemon also coordinates access to the shared cluster.

Object Gateway node

Ceph Object Gateway node runs the Ceph RADOS Gateway daemon (**ceph-radosgw**), and is an object storage interface built on top of **librados** to provide applications with a RESTful gateway to Ceph Storage Clusters. The Ceph Object Gateway supports two interfaces:

S3

Provides object storage functionality with an interface that is compatible with a large subset of the Amazon S3 RESTful API.

Swift

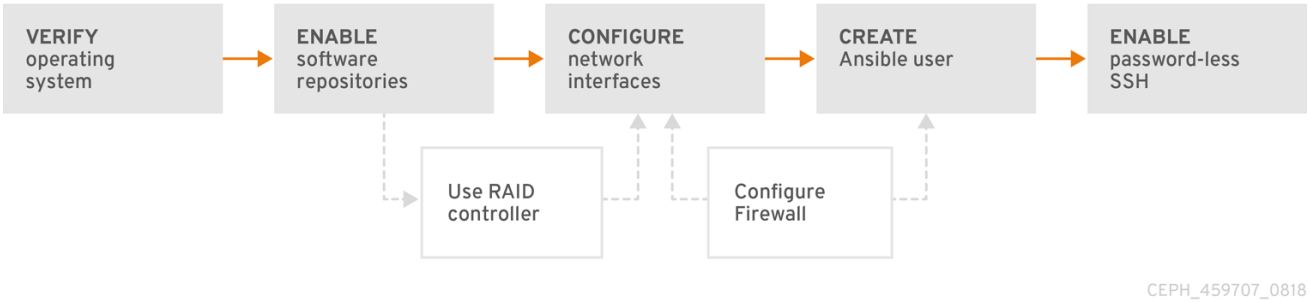
Provides object storage functionality with an interface that is compatible with a large subset of the OpenStack Swift API.

For details on the Ceph architecture, see the *Architecture Guide* for Red Hat Ceph Storage 3.

For minimum recommended hardware, see the Red Hat Ceph Storage Hardware Selection Guide 3.

CHAPTER 2. REQUIREMENTS FOR INSTALLING RED HAT CEPH STORAGE

Figure 2.1. Prerequisite Workflow



Before installing Red Hat Ceph Storage (RHCS), review the following requirements and prepare each Monitor, OSD, Metadata Server, and client nodes accordingly.

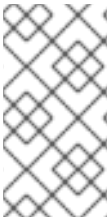
2.1. PREREQUISITES

- Verify the hardware meets the minimum requirements. For details, see the Hardware Guide for Red Hat Ceph Storage 3.

2.2. REQUIREMENTS CHECKLIST FOR INSTALLING RED HAT CEPH STORAGE

Task	Required	Section	Recommendation
Verifying the operating system version	Yes	Section 2.3, "Operating system requirements for Red Hat Ceph Storage"	
Enabling Ceph software repositories	Yes	Section 2.4, "Enabling the Red Hat Ceph Storage Repositories"	
Using a RAID controller with OSD nodes	No	Section 2.5, "Considerations for Using a RAID Controller with OSD Nodes (optional)"	Enabling write-back caches on a RAID controller might result in increased small I/O write throughput for OSD nodes.
Configuring the network	Yes	Section 2.7, "Verifying the Network Configuration for Red Hat Ceph Storage"	At minimum, a public network is required. However, a private network for cluster communication is recommended.

Task	Required	Section	Recommendation
Configuring a firewall	No	Section 2.8, "Configuring a firewall for Red Hat Ceph Storage"	A firewall can increase the level of trust for a network.
Creating an Ansible user	Yes	Section 2.9, "Creating an Ansible user with **sudo** access"	Creating the Ansible user is required on all Ceph nodes.
Enabling password-less SSH	Yes	Section 2.10, "Enabling Password-less SSH for Ansible"	Required for Ansible.



NOTE

By default, **ceph-ansible** installs NTP as a requirement. If NTP is customized, refer to *Configuring the Network Time Protocol for Red Hat Ceph Storage* in Manually Installing Red Hat Ceph Storage to understand how NTP must be configured to function properly with Ceph.

2.3. OPERATING SYSTEM REQUIREMENTS FOR RED HAT CEPH STORAGE

Red Hat Ceph Storage 3 requires Ubuntu 16.04.04 with a homogeneous version, such as AMD64 or Intel 64 architectures, running on all Ceph nodes in the storage cluster.

IMPORTANT

Red Hat does not support clusters with heterogeneous operating systems and versions.

Additional Resources

- The Installation Guide for Red Hat Enterprise Linux 7.
- The System Administrator's Guide for Red Hat Enterprise Linux 7.

Return to requirements checklist

2.4. ENABLING THE RED HAT CEPH STORAGE REPOSITORIES

Before you can install Red Hat Ceph Storage, you must choose an installation method. Red Hat Ceph Storage supports two installation methods:

- Content Delivery Network (CDN)
  For Ceph Storage clusters with Ceph nodes that can connect directly to the internet, use Red Hat Subscription Manager to enable the required Ceph repository.
- Local Repository

For Ceph Storage clusters where security measures preclude nodes from accessing the internet, install Red Hat Ceph Storage 3.2 from a single software build delivered as an ISO image, which will allow you to install local repositories.

Access to the RHCS software repositories requires a valid Red Hat login and password on the Red Hat Customer Portal.

IMPORTANT

Contact your account manager to obtain credentials for https://rhcs.download.redhat.com.

Prerequisites

- Valid customer subscription.
- For CDN installations, RHCS nodes must be able to connect to the internet.

Procedure

For CDN installations:

On the **Ansible administration node**, enable the Red Hat Ceph Storage 3 Tools repository:

```
$ sudo bash -c 'umask 0077; echo deb
https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/Tools $(lsb_release -
sc) main | tee /etc/apt/sources.list.d/Tools.list'

$ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-key add -'

$ sudo apt-get update
```

For ISO installations:

1. Log in to the Red Hat Customer Portal.
2. Click **Downloads** to visit the **Software & Download** center.
3. In the Red Hat Ceph Storage area, click **Download Software** to download the latest version of the software.

Additional Resources

- The Registering and Managing Subscriptions chapter in the System Administrator's Guide for Red Hat Enterprise Linux.

Return to the requirements checklist

2.5. CONSIDERATIONS FOR USING A RAID CONTROLLER WITH OSD NODES (OPTIONAL)

If an OSD node has a RAID controller with 1-2GB of cache installed, enabling the write-back cache might result in increased small I/O write throughput. However, the cache must be non-volatile.

Modern RAID controllers usually have super capacitors that provide enough power to drain volatile memory to non-volatile NAND memory during a power loss event. It is important to understand how a particular controller and its firmware behave after power is restored.

Some RAID controllers require manual intervention. Hard drives typically advertise to the operating system whether their disk caches should be enabled or disabled by default. However, certain RAID controllers and some firmware do not provide such information. Verify that disk level caches are disabled to avoid file system corruption.

Create a single RAID 0 volume with write-back for each Ceph OSD data drive with write-back cache enabled.

If Serial Attached SCSI (SAS) or SATA connected Solid-state Drive (SSD) disks are also present on the RAID controller, then investigate whether the controller and firmware support *pass-through* mode. Enabling *pass-through* mode helps avoid caching logic, and generally results in much lower latency for fast media.

Return to requirements checklist

2.6. CONSIDERATIONS FOR USING NVME WITH OBJECT GATEWAY (OPTIONAL)

If you plan to use the Object Gateway feature of Red Hat Ceph Storage and your OSD nodes have NVMe based SSDs or SATA SSDs, consider following the procedures in Ceph Object Gateway for Production to use NVMe with LVM optimally. These procedures explain how to use specially designed Ansible playbooks which will place journals and bucket indexes together on SSDs, which can increase performance compared to having all journals on one device. The information on using NVMe with LVM optimally should be referenced in combination with this Installation Guide.

Return to requirements checklist

2.7. VERIFYING THE NETWORK CONFIGURATION FOR RED HAT CEPH STORAGE

All Red Hat Ceph Storage (RHCS) nodes require a public network. You must have a network interface card configured to a public network where Ceph clients can reach Ceph monitors and Ceph OSD nodes.

You might have a network interface card for a cluster network so that Ceph can conduct heart-beating, peering, replication, and recovery on a network separate from the public network.

Configure the network interface settings and ensure to make the changes persistent.

IMPORTANT

Red Hat does not recommend using a single network interface card for both a public and private network.

Additional Resources

- For more information on network configuration see the *Network Configuration Reference* chapter in the *Configuration Guide* for Red Hat Ceph Storage 3.

Return to requirements checklist

2.8. CONFIGURING A FIREWALL FOR RED HAT CEPH STORAGE

Red Hat Ceph Storage (RHCS) uses the **iptables** service.

The Monitor daemons use port **6789** for communication within the Ceph storage cluster.

On each Ceph OSD node, the OSD daemons use several ports in the range **6800-7300**:

- One for communicating with clients and monitors over the public network
- One for sending data to other OSDs over a cluster network, if available; otherwise, over the public network
- One for exchanging heartbeat packets over a cluster network, if available; otherwise, over the public network

The Ceph Manager (**ceph-mgr**) daemons use ports in range **6800-7300**. Consider colocating the **ceph-mgr** daemons with Ceph Monitors on same nodes.

The Ceph Metadata Server nodes (**ceph-mds**) use port **6800**.

The Ceph Object Gateway nodes are configured by Ansible to use port **8080** by default. However, you can change the default port, for example to port **80**.

To use the SSL/TLS service, open port **443**.

Prerequisite

- Network hardware is connected.

Procedure

Run the following commands as the **root** user.

1. On all Monitor nodes, open port **6789** on the public network:

   ```
   iptables -I INPUT 1 -i iface -p tcp -s IP_address/netmask_prefix --dport 6789 -j ACCEPT
   ```

   Replace

   - **iface** with the name of network interface card on the public network.
   - **IP_address** with the network address of the Monitor node.
   - **netmask_prefix** with the netmask in Classless Inter-domain Routing (CIDR) notation.

   Example

   ```
   $ sudo iptables -I INPUT 1 -i enp6s0 -p tcp -s 192.168.0.11/24 --dport 6789 -j ACCEPT
   ```

2. On all OSD nodes, open ports **6800-7300** on the public network:

   ```
   iptables -I INPUT 1 -i iface -m multiport -p tcp -s IP_address/netmask_prefix --dports
   6800:7300 -j ACCEPT
   ```

Replace

- **iface** with the name of network interface card on the public network.
- **IP_address** with the network address of the OSD nodes.
- **netmask_prefix** with the netmask in CIDR notation.

Example

```
$ sudo iptables -I INPUT 1 -i enp6s0 -m multiport -p tcp -s 192.168.0.21/24 --dports 6800:7300 -j ACCEPT
```

3. On all Ceph Manager (**ceph-mgr**) nodes (usually the same nodes as Monitor ones), open ports **6800-7300** on the public network:

```
iptables -I INPUT 1 -i iface -m multiport -p tcp -s IP_address/netmask_prefix --dports 6800:7300 -j ACCEPT
```

Replace

- **iface** with the name of network interface card on the public network.
- **IP_address** with the network address of the OSD nodes.
- **netmask_prefix** with the netmask in CIDR notation.

Example

```
$ sudo iptables -I INPUT 1 -i enp6s0 -m multiport -p tcp -s 192.168.0.21/24 --dports 6800:7300 -j ACCEPT
```

4. On all Ceph Metadata Server (**ceph-mds**) nodes, open port **6800** on the public network:

```
iptables -I INPUT 1 -i iface -m multiport -p tcp -s IP_address/netmask_prefix --dports 6800 -j ACCEPT
```

Replace

- **iface** with the name of network interface card on the public network.
- **IP_address** with the network address of the OSD nodes.
- **netmask_prefix** with the netmask in CIDR notation.

Example

```
$ sudo iptables -I INPUT 1 -i enp6s0 -m multiport -p tcp -s 192.168.0.21/24 --dports 6800 -j ACCEPT
```

5. On all Ceph Object Gateway nodes, open the relevant port or ports on the public network.
   a. To open the default Ansible configured port of **8080**:

```
iptables -I INPUT 1 -i iface -p tcp -s IP_address/netmask_prefix --dport 8080 -j ACCEPT
```

Replace

- **iface** with the name of the network interface card on the public network.
- **IP_address** with the network address of the object gateway node.
- **netmask_prefix** with the netmask in CIDR notation.

Example

```
$ sudo iptables -I INPUT 1 -i enp6s0 -p tcp -s 192.168.0.31/24 --dport 8080 -j ACCEPT
```

b. Optional. If you installed Ceph Object Gateway using Ansible and changed the default port that Ansible configures Ceph Object Gateway to use from **8080**, for example, to port **80**, open this port:

```
iptables -I INPUT 1 -i iface -p tcp -s IP_address/netmask_prefix --dport 80 -j ACCEPT
```

Replace

- **iface** with the name of network interface card on the public network.
- **IP_address** with the network address of the object gateway node.
- **netmask_prefix** with the netmask in CIDR notation.

Example

```
$ sudo iptables -I INPUT 1 -i enp6s0 -p tcp -s 192.168.0.31/24 --dport 80 -j ACCEPT
```

c. Optional. To use SSL/TLS, open port **443**:

```
iptables -I INPUT 1 -i iface -p tcp -s IP_address/netmask_prefix --dport 443 -j ACCEPT
```

Replace

- **iface** with the name of network interface card on the public network.
- **IP_address** with the network address of the object gateway node.
- **netmask_prefix** with the netmask in CIDR notation.

Example

```
$ sudo iptables -I INPUT 1 -i enp6s0 -p tcp -s 192.168.0.31/24 --dport 443 -j ACCEPT
```

6. Make the changes persistent on all RHCS nodes in the storage cluster.

a. Install the **iptables-persistent** package:

```
$ sudo apt-get install iptables-persistent
```

b. In the terminal UI that appears, select **yes** to save current **IPv4 iptables** rules to the **/etc/iptables/rules.v4** file and current **IPv6 iptables** rules to the **/etc/iptables/rules.v6** file.

NOTE

If you add a new **iptables** rule after installing **iptables-persistent**, add the new rule to the **rules** file:

```
$ sudo iptables-save >> /etc/iptables/rules.v4
```

Additional Resources

- For more information about public and cluster network, see Verifying the Network Configuration for Red Hat Ceph Storage.

Return to requirements checklist

2.9. CREATING AN ANSIBLE USER WITH SUDO ACCESS

Ansible must be able to log into all the Red Hat Ceph Storage (RHCS) nodes as a user that has **root** privileges to install software and create configuration files without prompting for a password. You must create an Ansible user with password-less **root** access on all nodes in the storage cluster when deploying and configuring a Red Hat Ceph Storage cluster with Ansible.

Prerequisite

- Having **root** or **sudo** access to all nodes in the storage cluster.

Procedure

1. Log in to a Ceph node as the **root** user:

   ```
   ssh root@$HOST_NAME
   ```

   Replace

   - **$HOST_NAME** with the host name of the Ceph node.

   Example

   ```
   # ssh root@mon01
   ```

   Enter the **root** password when prompted.

2. Create a new Ansible user:

   ```
   adduser $USER_NAME
   ```

   Replace

- **$USER_NAME** with the new user name for the Ansible user.

Example

```
$ sudo adduser admin
```

Enter the password for this user twice when prompted.

IMPORTANT

Do not use **ceph** as the user name. The **ceph** user name is reserved for the Ceph daemons. A uniform user name across the cluster can improve ease of use, but avoid using obvious user names, because intruders typically use them for brute-force attacks.

3. Configure **sudo** access for the newly created user:

```
cat << EOF >/etc/sudoers.d/$USER_NAME
$USER_NAME ALL = (root) NOPASSWD:ALL
EOF
```

Replace

- **$USER_NAME** with the new user name for the Ansible user.

Example

```
$ sudo cat << EOF >/etc/sudoers.d/admin
admin ALL = (root) NOPASSWD:ALL
EOF
```

4. Assign the correct file permissions to the new file:

```
chmod 0440 /etc/sudoers.d/$USER_NAME
```

Replace

- **$USER_NAME** with the new user name for the Ansible user.

Example

```
$ sudo chmod 0440 /etc/sudoers.d/admin
```

Additional Resources

- The Adding a New User section in the *System Administrator's Guide* for Red Hat Enterprise Linux 7.

Return to the requirements checklist

2.10. ENABLING PASSWORD-LESS SSH FOR ANSIBLE

Generate an SSH key pair on the Ansible administration node and distribute the public key to each node in the storage cluster so that Ansible can access the nodes without being prompted for a password.

Prerequisites

- Create an Ansible user with **sudo** access.

Procedure

Do the following steps from the Ansible administration node, and as the Ansible user.

1. Generate the SSH key pair, accept the default file name and leave the passphrase empty:

```
[user@admin ~]$ ssh-keygen
```

2. Copy the public key to all nodes in the storage cluster:

```
ssh-copy-id $USER_NAME@$HOST_NAME
```

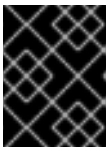
Replace

- **$USER_NAME** with the new user name for the Ansible user.
- **$HOST_NAME** with the host name of the Ceph node.

Example

```
[user@admin ~]$ ssh-copy-id ceph-admin@ceph-mon01
```

3. Create and edit the ~/**.ssh**/**config** file.



IMPORTANT

By creating and editing the ~/**.ssh**/**config** file you do not have to specify the **-u $USER_NAME** option each time you execute the **ansible-playbook** command.

a. Create the SSH **config** file:

```
[user@admin ~]$ touch ~/.ssh/config
```

b. Open the **config** file for editing. Set the **Hostname** and **User** options for each node in the storage cluster:

```
Host node1
  Hostname $HOST_NAME
  User $USER_NAME
Host node2
  Hostname $HOST_NAME
  User $USER_NAME
...
```

Replace

- **$HOST_NAME** with the host name of the Ceph node.
- **$USER_NAME** with the new user name for the Ansible user.

Example

```
Host node1
  Hostname monitor
  User admin
Host node2
  Hostname osd
  User admin
Host node3
  Hostname gateway
  User admin
```

4. Set the correct file permissions for the **~/.ssh/config** file:

```
[admin@admin ~]$ chmod 600 ~/.ssh/config
```

Additional Resources

- The **ssh_config(5)** manual page
- The OpenSSH chapter in the *System Administrator's Guide* for Red Hat Enterprise Linux 7

Return to requirements checklist

CHAPTER 3. DEPLOYING RED HAT CEPH STORAGE

This chapter describes how to use the Ansible application to deploy a Red Hat Ceph Storage cluster and other components, such as Metadata Servers or the Ceph Object Gateway.

- To install a Red Hat Ceph Storage cluster, see Section 3.2, "Installing a Red Hat Ceph Storage Cluster".
- To install Metadata Servers, see Section 3.4, "Installing Metadata Servers".
- To install the **ceph-client** role, see Section 3.5, "Installing the Ceph Client Role".
- To install the Ceph Object Gateway, see Section 3.6, "Installing the Ceph Object Gateway".
- To configure a multisite Ceph Object Gateway, see Section 3.6.1, "Configuring a multisite Ceph Object Gateway".
- To learn about the Ansible **--limit** option, see Section 3.8, "Understanding the **limit** option".

Previously, Red Hat did not provide the **ceph-ansible** package for Ubuntu. In Red Hat Ceph Storage version 3 and later, you can use the Ansible automation application to deploy a Ceph cluster from an Ubuntu node.

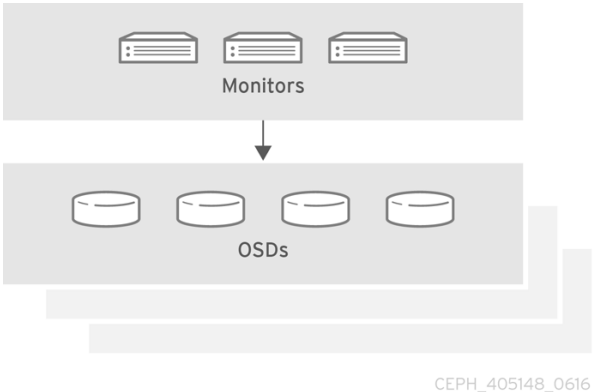
3.1. PREREQUISITES

- Obtain a valid customer subscription.
- Prepare the cluster nodes. On each node:
  - Enable the appropriate software repositories.
  - Create an Ansible user.
  - Enable passwordless SSH access.
  - Optional. Configure firewall.

3.2. INSTALLING A RED HAT CEPH STORAGE CLUSTER

Use the Ansible application with the **ceph-ansible** playbook to install Red Hat Ceph Storage 3.

Production Ceph storage clusters start with a minimum of three monitor hosts and three OSD nodes containing multiple OSD daemons.



Prerequisites

- On the Ansible administration node, install the **ceph-ansible** package:

```
[user@admin ~]$ sudo apt-get install ceph-ansible
```

Procedure

Run the following commands from the Ansible administration node unless instructed otherwise.

1. As the Ansible user, create the **ceph-ansible-keys** directory where Ansible stores temporary values generated by the **ceph-ansible** playbook.

```
[user@admin ~]$ mkdir ~/ceph-ansible-keys
```

2. As root, create a symbolic link to the **/usr/share/ceph-ansible/group_vars** directory in the **/etc/ansible/** directory:

```
[root@admin ~]# ln -s /usr/share/ceph-ansible/group_vars /etc/ansible/group_vars
```

3. Navigate to the **/usr/share/ceph-ansible/** directory:

```
[root@admin ~]$ cd /usr/share/ceph-ansible
```

4. Create new copies of the **yml.sample** files:

```
[root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
[root@admin ceph-ansible]# cp site.yml.sample site.yml
```

5. Edit the copied files.

   a. Edit the **group_vars/all.yml** file. See the table below for the most common required and optional parameters to uncomment. Note that the table does not include all parameters.

IMPORTANT

Do not set the **cluster: ceph** parameter to any value other than **ceph** because using custom cluster names is not supported.

Table 3.1. General Ansible Settings

Option	Value	Required	Notes

Option	Value	Required	Notes
ceph_origin	**repository** or **distro** or **local**	Yes	The **repository** value means Ceph will be installed through a new repository. The **distro** value means that no separate repository file will be added, and you will get whatever version of Ceph that is included with the Linux distribution. The **local** value means the Ceph binaries will be copied from the local machine.
ceph_repository_type	**cdn** or **iso**	Yes	
ceph_rhcs_iso_path	The path to the ISO image	Yes if using an ISO image	
ceph_rhcs_cdn_debian_repo	The credentials to access the online Ubuntu Ceph repositories. For example, **https://username:password@rhcs.download.redhat.com**.	Yes	
ceph_rhcs_cdn_debian_repo_version	Use **/3-release/** for new installations; use **/3-updates/** for updates.	Yes	
monitor_interface	The interface that the Monitor nodes listen to	**monitor_interface**, **monitor_address**, or **monitor_address_block** is required	
monitor_address	The address that the Monitor nodes listen to		

Option	Value	Required	Notes
monitor_address_block	The subnet of the Ceph public network		Use when the IP addresses of the nodes are unknown, but the subnet is known
ip_version	**ipv6**	Yes if using IPv6 addressing	
public_network	The IP address and netmask of the Ceph public network, or the corresponding IPv6 address if using IPv6	Yes	Section 2.7, "Verifying the Network Configuration for Red Hat Ceph Storage"
cluster_network	The IP address and netmask of the Ceph cluster network	No, defaults to **public_network**	

An example of the **all.yml** file can look like:

```
ceph_origin: repository
ceph_repository: rhcs
ceph_repository_type: cdn
ceph_rhcs_version: 3
monitor_interface: eth0
public_network: 192.168.0.0/24
```

NOTE

Having the **ceph_rhcs_version** option set to **3** will pull in the latest version of Red Hat Ceph Storage 3.

For additional details, see the **all.yml** file.

b. Edit the **group_vars/osds.yml** file. See the table below for the most common required and optional parameters to uncomment. Note that the table does not include all parameters.

IMPORTANT

Use a different physical device to install an OSD than the device where the operating system is installed. Sharing the same device between the operating system and OSDs causes performance issues.

Table 3.2. OSD Ansible Settings

Option	Value	Required	Notes
osd_scenario	**collocated** to use the same device for write-ahead logging and key/value data (BlueStore) or journal (FileStore) and OSD data **non-collocated** to use a dedicated device, such as SSD or NVMe media to store write-ahead log and key/value data (BlueStore) or journal data (FileStore) **lvm** to use the Logical Volume Manager to store OSD data	Yes	When using **osd_scenario: non-collocated**, **ceph-ansible** expects the variables **devices** and **dedicated_devices** to match. For example, if you specify 10 disks in **devices**, you must specify 10 entries in **dedicated_devices**.
osd_auto_discovery	**true** to automatically discover OSDs	Yes if using **osd_scenario: collocated**	Cannot be used when **devices** setting is used
devices	List of devices where **ceph data** is stored	Yes to specify the list of devices	Cannot be used when **osd_auto_discovery** setting is used. When using **lvm** as the **osd_scenario** and setting the **devices** option, **ceph-volume lvm batch** mode creates the optimized OSD configuration.
dedicated_devices	List of dedicated devices for non-collocated OSDs where **ceph journal** is stored	Yes if **osd_scenario: non-collocated**	Should be nonpartitioned devices
dmcrypt	**true** to encrypt OSDs	No	Defaults to **false**

Option	Value	Required	Notes
lvm_volumes	A list of FileStore or BlueStore dictionaries	Yes if using **osd_scenario: lvm** and storage devices are not defined using **devices**	Each dictionary must contain a **data**, **journal** and **data_vg** keys. Any logical volume or volume group must be the name and not the full path. The **data**, and **journal** keys can be a logical volume (LV) or partition, but do not use one journal for multiple **data** LVs. The **data_vg** key must be the volume group containing the **data** LV. Optionally, the **journal_vg** key can be used to specify the volume group containing the journal LV, if applicable. See the examples below for various supported configurations.
osds_per_device	The number of OSDs to create per device.	No	Defaults to **1**
osd_objectstore	The Ceph object store type for the OSDs.	No	Defaults to **bluestore**. The other option is **filestore**. Required for upgrades.

The following are examples of the **osds.yml** file when using the three OSD scenarios: **collocated**, **non-collocated**, and **lvm**. The default OSD object store format is BlueStore, if not specified.

Collocated

```
osd_objectstore: filestore
osd_scenario: collocated
devices:
  - /dev/sda
  - /dev/sdb
```

Non-collocated - BlueStore

```
osd_objectstore: bluestore
osd_scenario: non-collocated
devices:
 - /dev/sda
 - /dev/sdb
 - /dev/sdc
 - /dev/sdd
dedicated_devices:
 - /dev/nvme0n1
 - /dev/nvme0n1
 - /dev/nvme1n1
 - /dev/nvme1n1
```

This non-collocated example will create four BlueStore OSDs, one per device. In this example, the traditional hard drives (**sda**, **sdb**, **sdc**, **sdd**) are used for object data, and the solid state drives (SSDs) (/**dev**/**nvme0n1**, /**dev**/**nvme1n1**) are used for the BlueStore databases and write-ahead logs. This configuration pairs the /**dev**/**sda** and /**dev**/**sdb** devices with the /**dev**/**nvme0n1** device, and pairs the /**dev**/**sdc** and /**dev**/**sdd** devices with the /**dev**/**nvme1n1** device.

Non-collocated - FileStore

```
osd_objectstore: filestore
osd_scenario: non-collocated
devices:
  - /dev/sda
  - /dev/sdb
  - /dev/sdc
  - /dev/sdd
dedicated_devices:
   - /dev/nvme0n1
   - /dev/nvme0n1
   - /dev/nvme1n1
   - /dev/nvme1n1
```

LVM simple

```
osd_objectstore: bluestore
osd_scenario: lvm
devices:
  - /dev/sda
  - /dev/sdb
```

or

```
osd_objectstore: bluestore
osd_scenario: lvm
devices:
  - /dev/sda
  - /dev/sdb
  - /dev/nvme0n1
```

With these simple configurations **ceph-ansible** uses batch mode (**ceph-volume lvm batch**) to create the OSDs.

In the first scenario, if the **devices** are traditional hard drives or SSDs, then one OSD per device is created.

In the second scenario, when there is a mix of traditional hard drives and SSDs, the data is placed on the traditional hard drives (**sda**, **sdb**) and the BlueStore database (**block.db**) is created as large as possible on the SSD (**nvme0n1**).

LVM advance

```
osd_objectstore: filestore
osd_scenario: lvm
lvm_volumes:
  - data: data-lv1
    data_vg: vg1
    journal: journal-lv1
    journal_vg: vg2
  - data: data-lv2
    journal: /dev/sda
    data_vg: vg1
```

or

```
osd_objectstore: bluestore
osd_scenario: lvm
lvm_volumes:
  - data: data-lv1
    data_vg: data-vg1
    db: db-lv1
    db_vg: db-vg1
    wal: wal-lv1
    wal_vg: wal-vg1
  - data: data-lv2
    data_vg: data-vg2
    db: db-lv2
    db_vg: db-vg2
    wal: wal-lv2
    wal_vg: wal-vg2
```

With these advance scenario examples, the volume groups and logical volumes must be created beforehand. They will not be created by **ceph-ansible**.

NOTE

If using all NVMe SSDs set the **osd_scenario: lvm** and **osds_per_device: 4** options. For more information, see *Configuring OSD Ansible settings for all NVMe Storage* for Red Hat Enterprise Linux or *Configuring OSD Ansible settings for all NVMe Storage* for Ubuntu in the Red Hat Ceph Storage *Installation Guides*.

For additional details, see the comments in the **osds.yml** file.

6. Edit the Ansible inventory file located by default at **/etc/ansible/hosts**. Remember to comment out example hosts.

   a. Add the Monitor nodes under the **[mons]** section:

```
[mons]
<monitor-host-name>
<monitor-host-name>
<monitor-host-name>
```

b. Add OSD nodes under the **[osds]** section. If the nodes have sequential naming, consider using a range:

```
[osds]
<osd-host-name[1:10]>
```

NOTE

For OSDs in a new installation, the default object store format is BlueStore.

Optionally, use the **devices** parameter to specify devices that the OSD nodes will use. Use a comma-separated list to list multiple devices.

```
[osds]
<ceph-host-name> devices="[ '<device_1>', '<device_2>' ]"
```

For example:

```
[osds]
ceph-osd-01 devices="[ '/dev/sdb', '/dev/sdc' ]"
ceph-osd-02 devices="[ '/dev/sdb', '/dev/sdc', '/dev/sdd' ]"
```

When specifying no devices, set the **osd_auto_discovery** option to **true** in the **osds.yml** file.

NOTE

Using the **devices** parameter is useful when OSDs use devices with different names or when one of the devices failed on one of the OSDs.

Optionally, if you want **ansible-playbook** to create a custom CRUSH hierarchy, specify where you want the OSD hosts to be in the CRUSH map's hierarchy by using the **osd_crush_location** parameter. You must specify at least two CRUSH bucket types to specify the location of the OSD, and one bucket **type** must be **host**. By default, these include **root**, **datacenter**, **room**, **row**, **pod**, **pdu**, **rack**, **chassis** and **host**.

```
[osds]
<ceph-host-name> osd_crush_location="{ 'root': '<root-bucket>', 'rack': '<rack-bucket>',
'pod': '<pod-bucket>', 'host': '<ceph-host-name>' }"
```

For example:

```
[osds]
ceph-osd-01 osd_crush_location="{ 'root': 'mon-root', 'rack': 'mon-rack', 'pod': 'monpod',
'host': 'ceph-osd-01' }"
```

c. Add the Ceph Manager (**ceph-mgr**) nodes under the **[mgrs]** section. Colocate the Ceph Manager daemon with Monitor nodes.

```
[mgrs]
<monitor-host-name>
<monitor-host-name>
<monitor-host-name>
```

7. As the Ansible user, ensure that Ansible can reach the Ceph hosts:

```
[user@admin ~]$ ansible all -m ping
```

8. Add the following line to the **/etc/ansible/ansible.cfg** file:

```
retry_files_save_path = ~/
```

9. As **root**, create the **/var/log/ansible/** directory and assign the appropriate permissions for the **ansible** user:

```
[root@admin ~]# mkdir /var/log/ansible
[root@admin ~]# chown ansible:ansible  /var/log/ansible
[root@admin ~]# chmod 755 /var/log/ansible
```

a. Edit the **/usr/share/ceph-ansible/ansible.cfg** file, updating the **log_path** value as follows:

```
log_path = /var/log/ansible/ansible.log
```

10. As the Ansible user, change to the **/usr/share/ceph-ansible/** directory:

```
[user@admin ~]$ cd /usr/share/ceph-ansible/
```

11. Run the **ceph-ansible** playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site.yml
```

12. Using the root account on a Monitor node, verify the status of the Ceph cluster:

```
[root@monitor ~]# ceph health
HEALTH_OK
```

13. Verify the cluster is functioning using **rados**.

a. From a monitor node, create a test pool with eight placement groups:
Syntax

```
[root@monitor ~]# ceph osd pool create <pool-name> <pg-number>
```

Example

```
[root@monitor ~]# ceph osd pool create test 8
```

b. Create a file called **hello-world.txt**:

Syntax

```
[root@monitor ~]# vim <file-name>
```

Example

```
[root@monitor ~]# vim hello-world.txt
```

c. Upload **hello-world.txt** to the test pool using the object name **hello-world**:
Syntax

```
[root@monitor ~]# rados --pool <pool-name> put <object-name> <object-file>
```

Example

```
[root@monitor ~]# rados --pool test put hello-world hello-world.txt
```

d. Download **hello-world** from the test pool as file name **fetch.txt**:
Syntax

```
[root@monitor ~]# rados --pool <pool-name> get <object-name> <object-file>
```

Example

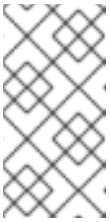
```
[root@monitor ~]# rados --pool test get hello-world fetch.txt
```

e. Check the contents of **fetch.txt**:

```
[root@monitor ~]# cat fetch.txt
```

The output should be:

```
"Hello World!"
```



NOTE

In addition to verifying the cluster status, you can use the **ceph-medic** utility to overall diagnose the Ceph Storage Cluster. See the *Installing and Using **ceph-medic** to Diagnose a Ceph Storage Cluster* chapter in the Red Hat Ceph Storage 3 *Troubleshooting Guide*.

3.3. CONFIGURING OSD ANSIBLE SETTINGS FOR ALL NVME STORAGE

To optimize performance when using only non-volatile memory express (NVMe) devices for storage, configure four OSDs on each NVMe device. Normally only one OSD is configured per device, which will underutilize the throughput of an NVMe device.

NOTE

If you mix SSDs and HDDs, then SSDs will be used for either journals or **block.db**, not OSDs.

NOTE

In testing, configuring four OSDs on each NVMe device was found to provide optimal performance. It is recommended to set **osds_per_device: 4**, but it is not required. Other values may provide better performance in your environment.

Prerequisites

- Satisfying all software and hardware requirements for a Ceph cluster.

Procedure

1. Set **osd_scenario: lvm** and **osds_per_device: 4** in **group_vars/osds.yml**:

```
osd_scenario: lvm
osds_per_device: 4
```

2. List the NVMe devices under **devices**:

```
devices:
  - /dev/nvme0n1
  - /dev/nvme1n1
  - /dev/nvme2n1
  - /dev/nvme3n1
```

3. The settings in **group_vars/osds.yml** will look similar to this example:

```
osd_scenario: lvm
osds_per_device: 4
devices:
  - /dev/nvme0n1
  - /dev/nvme1n1
  - /dev/nvme2n1
  - /dev/nvme3n1
```

NOTE

You must use **devices** with this configuration, not **lvm_volumes**. This is because **lvm_volumes** is generally used with pre-created logical volumes and **osds_per_device** implies automatic logical volume creation by Ceph.

Additional Resources

- Installing a Red Hat Ceph Storage Cluster on Red Hat Enterprise Linux
- Installing a Red Hat Ceph Storage Cluster on Ubuntu

3.4. INSTALLING METADATA SERVERS

Use the Ansible automation application to install a Ceph Metadata Server (MDS). Metadata Server daemons are necessary for deploying a Ceph File System.

Prerequisites

- A working Red Hat Ceph Storage cluster.

Procedure

Perform the following steps on the Ansible administration node.

1. Add a new section **[mdss]** to the **/etc/ansible/hosts** file:

```
[mdss]
hostname
hostname
hostname
```

   Replace *hostname* with the host names of the nodes where you want to install the Ceph Metadata Servers.

2. Navigate to the **/usr/share/ceph-ansible** directory:

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

3. Optional. Change the default variables.

   a. Create a copy of the **group_vars/mdss.yml.sample** file named **mdss.yml**:

```
[root@admin ceph-ansible]# cp group_vars/mdss.yml.sample group_vars/mdss.yml
```

   b. Optionally, edit parameters in **mdss.yml**. See **mdss.yml** for details.

4. As the Ansible user, run the Ansible playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit mdss
```

5. After installing Metadata Servers, configure them. For details, see the *Configuring Metadata Server Daemons* chapter in the Ceph File System Guide for Red Hat Ceph Storage 3.

Additional Resources

- The *Ceph File System Guide* for Red Hat Ceph Storage 3
- *Understanding the **limit** option*

3.5. INSTALLING THE CEPH CLIENT ROLE

The **ceph-ansible** utility provides the **ceph-client** role that copies the Ceph configuration file and the administration keyring to nodes. In addition, you can use this role to create custom pools and clients.

Prerequisites

- A running Ceph storage cluster, preferably in the **active + clean** state.

- Perform the tasks listed in Chapter 2, *Requirements for Installing Red Hat Ceph Storage* .

Procedure

Perform the following tasks on the Ansible administration node.

1. Add a new section **[clients]** to the **/etc/ansible/hosts** file:

```
[clients]
<client-hostname>
```

Replace **<client-hostname>** with the host name of the node where you want to install the **ceph-client** role.

2. Navigate to the **/usr/share/ceph-ansible** directory:

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

3. Create a new copy of the **clients.yml.sample** file named **clients.yml**:

```
[root@admin ceph-ansible ~]# cp group_vars/clients.yml.sample group_vars/clients.yml
```

4. Open the **group_vars/clients.yml** file, and uncomment the following lines:

```
keys:
  - { name: client.test, caps: { mon: "allow r", osd: "allow class-read object_prefix
rbd_children, allow rwx pool=test" },  mode: "{{ ceph_keyring_permissions }}" }
```

   a. Replace **client.test** with the real client name, and add the client key to the client definition line, for example:

```
key: "ADD-KEYRING-HERE=="
```

   Now the whole line example would look similar to this:

```
- { name: client.test, key: "AQAin8tUMICVFBAALRHNrV0Z4MXupRw4v9JQ6Q==", caps:
{ mon: "allow r", osd: "allow class-read object_prefix rbd_children, allow rwx pool=test" },
mode: "{{ ceph_keyring_permissions }}" }
```

NOTE

The **ceph-authtool --gen-print-key** command can generate a new client key.

5. Optionally, instruct **ceph-client** to create pools and clients.

   a. Update **clients.yml**.

   - Uncomment the **user_config** setting and set it to **true**.
   - Uncomment the **pools** and **keys** sections and update them as required. You can define custom pools and client names altogether with the **cephx** capabilities.

   b. Add the **osd_pool_default_pg_num** setting to the **ceph_conf_overrides** section in the **all.yml** file:

```
ceph_conf_overrides:
  global:
    osd_pool_default_pg_num: <number>
```

Replace **<number>** with the default number of placement groups.

6. Run the Ansible playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site.yml --limit clients
```

Additional Resources

- Section 3.8, "Understanding the **limit** option"

3.6. INSTALLING THE CEPH OBJECT GATEWAY

The Ceph Object Gateway, also know as the RADOS gateway, is an object storage interface built on top of the **librados** API to provide applications with a RESTful gateway to Ceph storage clusters.

Prerequisites

- A running Red Hat Ceph Storage cluster, preferably in the **active + clean** state.
- On the Ceph Object Gateway node, perform the tasks listed in Chapter 2, *Requirements for Installing Red Hat Ceph Storage*.

Procedure

Perform the following tasks on the Ansible administration node.

1. Add gateway hosts to the **/etc/ansible/hosts** file under the **[rgws]** section to identify their roles to Ansible. If the hosts have sequential naming, use a range, for example:

```
[rgws]
<rgw_host_name_1>
<rgw_host_name_2>
<rgw_host_name[3..10]>
```

2. Navigate to the Ansible configuration directory:

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

3. Create the **rgws.yml** file from the sample file:

```
[root@ansible ~]# cp group_vars/rgws.yml.sample group_vars/rgws.yml
```

4. Open and edit the **group_vars/rgws.yml** file. To copy the administrator key to the Ceph Object Gateway node, uncomment the **copy_admin_key** option:

```
copy_admin_key: true
```

5. The **rgws.yml** file may specify a different default port than the default port **7480**. For example:

```
ceph_rgw_civetweb_port: 80
```

6. The **all.yml** file **MUST** specify a **radosgw_interface**. For example:

```
radosgw_interface: eth0
```

Specifying the interface prevents Civetweb from binding to the same IP address as another Civetweb instance when running multiple instances on the same host.

7. Generally, to change default settings, uncomment the settings in the **rgw.yml** file, and make changes accordingly. To make additional changes to settings that are not in the **rgw.yml** file, use **ceph_conf_overrides:** in the **all.yml** file. For example, set the **rgw_dns_name:** with the host of the DNS server and ensure the cluster's DNS server to configure it for wild cards to enable S3 subdomains.

```
ceph_conf_overrides:
  client.rgw.rgw1:
    rgw_dns_name: <host_name>
    rgw_override_bucket_index_max_shards: 16
    rgw_bucket_default_quota_max_objects: 1638400
```

For advanced configuration details, see the Red Hat Ceph Storage 3 *Ceph Object Gateway for Production* guide. Advanced topics include:

- *Configuring Ansible Groups*
- *Developing Storage Strategies*. See the *Creating the Root Pool*, *Creating System Pools*, and *Creating Data Placement Strategies* sections for additional details on how create and configure the pools.
  See Bucket Sharding for configuration details on bucket sharding.

8. Uncomment the **radosgw_interface** parameter in the **group_vars/all.yml** file.

```
radosgw_interface: <interface>
```

Replace:

- **<interface>** with the interface that the Ceph Object Gateway nodes listen to

For additional details, see the **all.yml** file.

9. Run the Ansible playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site.yml --limit rgws
```

NOTE

Ansible ensures that each Ceph Object Gateway is running.

For a single site configuration, add Ceph Object Gateways to the Ansible configuration.

For multi-site deployments, you should have an Ansible configuration for each zone. That is, Ansible will create a Ceph storage cluster and gateway instances for that zone.

After installation for a multi-site cluster is complete, proceed to the Multi-site chapter in the *Object Gateway Guide for Ubuntu* for details on configuring a cluster for multi-site.

Additional Resources

- Section 3.8, "Understanding the **limit** option"
- The *Object Gateway Guide for Ubuntu*

3.6.1. Configuring a multisite Ceph Object Gateway

Ansible will configure the realm, zonegroup, along with the master and secondary zones for a Ceph Object Gateway in a multisite environment.

Prerequisites

- Two running Red Hat Ceph Storage clusters.
- On the Ceph Object Gateway node, perform the tasks listed in the *Requirements for Installing Red Hat Ceph Storage* found in the *Red Hat Ceph Storage Installation Guide*.
- Install and configure one Ceph Object Gateway per storage cluster.

Procedure

1. Do the following steps on Ansible node for the primary storage cluster:
   a. Generate the system keys and capture their output in the **multi-site-keys.txt** file:

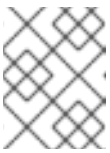
```
[root@ansible ~]# echo system_access_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 20 | head -n 1) > multi-site-keys.txt
[root@ansible ~]# echo system_secret_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 40 | head -n 1) >> multi-site-keys.txt
```

   b. Navigate to the Ansible configuration directory, **/usr/share/ceph-ansible**:

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

   c. Open and edit the **group_vars/all.yml** file. Enable multisite support by adding the following options, along with updating the **$ZONE_NAME**, **$ZONE_GROUP_NAME**, **$REALM_NAME**, **$ACCESS_KEY**, and **$SECRET_KEY** options accordingly:

```
rgw_multisite: true
rgw_zone: $ZONE_NAME
rgw_zonemaster: true
rgw_zonesecondary: false
rgw_multisite_endpoint_addr: "{{ ansible_fqdn }}"
rgw_zonegroup: $ZONE_GROUP_NAME
rgw_zone_user: zone.user
rgw_realm: $REALM_NAME
system_access_key: $ACCESS_KEY
system_secret_key: $SECRET_KEY
```



NOTE

The **ansible_fqdn** domain name must be resolvable from the secondary storage cluster.

d. Run the Ansible playbook:

```
[user@ansible ceph-ansible]$ ansible-playbook site.yml --limit rgws
```

e. Restart the Ceph Object Gateway daemon:

```
[root@rgw ~]# systemctl restart ceph-radosgw@rgw.`hostname -s`
```

2. Do the following steps on Ansible node for the secondary storage cluster:

a. Navigate to the Ansible configuration directory, **/usr/share/ceph-ansible**:

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

b. Open and edit the **group_vars/all.yml** file. Enable multisite support by adding the following options, along with updating the **$ZONE_NAME**, **$ZONE_GROUP_NAME**, **$REALM_NAME**, **$ACCESS_KEY**, and **$SECRET_KEY** options accordingly: The **rgw_zone_user**, **system_access_key**, and **system_secret_key** must be the same value as used in the master zone configuration. The **rgw_pullhost** option must be the Ceph Object Gateway for the master zone:

```
rgw_multisite: true
rgw_zone: $ZONE_NAME
rgw_zonemaster: false
rgw_zonesecondary: true
rgw_multisite_endpoint_addr: "{{ ansible_fqdn }}"
rgw_zonegroup: $ZONE_GROUP_NAME
rgw_zone_user: zone.user
rgw_realm: $REALM_NAME
system_access_key: $ACCESS_KEY
system_secret_key: $SECRET_KEY
rgw_pull_proto: http
rgw_pull_port: 8080
rgw_pullhost: $MASTER_RGW_NODE_NAME
```

NOTE

The **ansible_fqdn** domain name must be resolvable from the primary storage cluster.

c. Run the Ansible playbook:

```
[user@ansible ceph-ansible]$ ansible-playbook site.yml --limit rgws
```

3. After running the Ansible playbook on the master and secondary storage clusters, you will have a running active-active Ceph Object Gateway configuration.

4. Verify the multisite Ceph Object Gateway configuration:

a. From the Ceph Monitor and Object Gateway nodes at each site, primary and secondary, must be able to **curl** the other site.

b. Run the **radosgw-admin sync status** command on both sites.

3.7. INSTALLING THE NFS-GANESHA GATEWAY

The Ceph NFS Ganesha Gateway is an NFS interface built on top of the Ceph Object Gateway to provide applications with a POSIX filesystem interface to the Ceph Object Gateway for migrating files within filesystems to Ceph Object Storage.

Prerequisites

- A running Ceph storage cluster, preferably in the **active + clean** state.
- At least one node running a Ceph Object Gateway.
- At least one S3 user with an access key and secret.
- Perform the Before You Start procedure.

Procedure

Perform the following tasks on the Ansible administration node.

1. Create the **nfss** file from the sample file:

```
[root@ansible ~]# cd /etc/ansible/group_vars
[root@ansible ~]# cp nfss.yml.sample nfss.yml
```

2. Add gateway hosts to the **/etc/ansible/hosts** file under an **[nfss]** group to identify their group membership to Ansible. If the hosts have sequential naming, use a range. For example:

```
[nfss]
<nfs_host_name_1>
<nfs_host_name_2>
<nfs_host_name[3..10]>
```

3. Navigate to the Ansible configuration directory, **/etc/ansible/**:

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

4. To copy the administrator key to the Ceph Object Gateway node, uncomment the **copy_admin_key** setting in the **/usr/share/ceph-ansible/group_vars/nfss.yml** file:

```
copy_admin_key: true
```

5. Configure the FSAL (File System Abstraction Layer) sections of the **/usr/share/ceph-ansible/group_vars/nfss.yml** file. Provide an ID, S3 user ID, S3 access key and secret. For NFSv4, it should look something like this:

```
####################
# FSAL RGW Config #
####################
#ceph_nfs_rgw_export_id: <replace-w-numeric-export-id>
#ceph_nfs_rgw_pseudo_path: "/"
#ceph_nfs_rgw_protocols: "3,4"
#ceph_nfs_rgw_access_type: "RW"
#ceph_nfs_rgw_user: "cephnfs"
# Note: keys are optional and can be generated, but not on containerized, where
```

```
# they must be configered.
#ceph_nfs_rgw_access_key: "<replace-w-access-key>"
#ceph_nfs_rgw_secret_key: "<replace-w-secret-key>"
```

6. Run the Ansible playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site.yml --limit nfss
```

Additional Resources

- Section 3.8, "Understanding the **limit** option"
- The *Object Gateway Guide for Ubuntu*

3.8. UNDERSTANDING THE LIMIT OPTION

This section contains information about the Ansible **--limit** option.

Ansible supports the **--limit** option that enables you to use the **site**, **site-docker**, and **rolling_upgrade** Ansible playbooks for a particular section of the inventory file.

```
$ ansible-playbook site.yml|rolling_upgrade.yml|site-docker.yml --limit
osds|rgws|clients|mdss|nfss|iscsigws
```

For example, to redeploy only OSDs on bare metal, run the following command as the Ansible user:

```
$ ansible-playbook /usr/share/ceph-ansible/site.yml --limit osds
```

IMPORTANT

If you colocate Ceph components on one node, Ansible applies a playbook to all components on the node despite that only one component type was specified with the **limit** option. For example, if you run the **rolling_update** playbook with the **--limit osds** option on a node that contains OSDs and Metadata Servers (MDS), Ansible will upgrade both components, OSDs and MDSs.

3.9. ADDITIONAL RESOURCES

- The Ansible Documentation

CHAPTER 4. UPGRADING A RED HAT CEPH STORAGE CLUSTER

This section describes how to upgrade to a new major or minor version of Red Hat Ceph Storage.

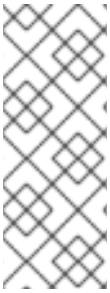
Previously, Red Hat did not provide the **ceph-ansible** package for Ubuntu. In Red Hat Ceph Storage version 3 and later, you can use the Ansible automation application to upgrade a Ceph cluster from an Ubuntu node.

- To upgrade a storage cluster, see Section 4.1, “Upgrading the Storage Cluster” .

Use the Ansible **rolling_update.yml** playbook located in the **/usr/share/ceph-ansible/infrastructure-playbooks/** directory from the administration node to upgrade between two major or minor versions of Red Hat Ceph Storage, or to apply asynchronous updates.

Ansible upgrades the Ceph nodes in the following order:

- Monitor nodes
- MGR nodes
- OSD nodes
- MDS nodes
- Ceph Object Gateway nodes
- All other Ceph client nodes

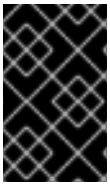


NOTE

Red Hat Ceph Storage 3 introduces several changes in Ansible configuration files located in the **/usr/share/ceph-ansible/group_vars/** directory; certain parameters were renamed or removed. Therefore, make backup copies of the **all.yml** and **osds.yml** files before creating new copies from the **all.yml.sample** and **osds.yml.sample** files after upgrading to version 3. For more details about the changes, see Appendix I, *Changes in Ansible Variables Between Version 2 and 3*.

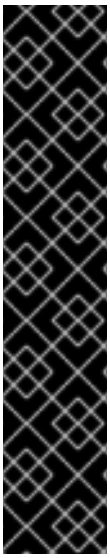
NOTE

Red Hat Ceph Storage 3.1 and later introduces new Ansible playbooks to optimize storage for performance when using Object Gateway and high speed NVMe based SSDs (and SATA SSDs). The playbooks do this by placing journals and bucket indexes together on SSDs, which can increase performance compared to having all journals on one device. These playbooks are designed to be used when installing Ceph. Existing OSDs continue to work and need no extra steps during an upgrade. There is no way to upgrade a Ceph cluster while simultaneously reconfiguring OSDs to optimize storage in this way. To use different devices for journals or bucket indexes requires reprovisioning OSDs. For more information see Using NVMe with LVM optimally in Ceph Object Gateway for Production .



IMPORTANT

The **rolling_update.yml** playbook includes the **serial** variable that adjusts the number of nodes to be updated simultaneously. Red Hat strongly recommends to use the default value (**1**), which ensures that Ansible will upgrade cluster nodes one by one.



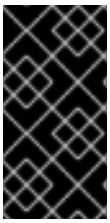
IMPORTANT

When using the **rolling_update.yml** playbook to upgrade to any Red Hat Ceph Storage 3.x version, users who use the Ceph File System (CephFS) must manually update the Metadata Server (MDS) cluster. This is due to a known issue.

Comment out the MDS hosts in **/etc/ansible/hosts** before upgrading the entire cluster using **ceph-ansible rolling-upgrade.yml**, and then upgrade MDS manually. In the **/etc/ansible/hosts** file:

```
#[mdss]
#host-abc
```

For more details about this known issue, including how to update the MDS cluster, refer to the Red Hat Ceph Storage 3.0 Release Notes.



IMPORTANT

When upgrading a Red Hat Ceph Storage cluster from a previous version to 3.2, the Ceph Ansible configuration will default the object store type to BlueStore. If you still want to use FileStore as the OSD object store, then explicitly set the Ceph Ansible configuration to FileStore. This ensures newly deployed and replaced OSDs are using FileStore.

IMPORTANT

When using the **rolling_update.yml** playbook to upgrade to any Red Hat Ceph Storage 3.x version, and if you are using a multisite Ceph Object Gateway configuration, then you do not have to manually update the **all.yml** file to specify the multisite configuration.

Prerequisites

- If the Ceph nodes are not connected to the Red Hat Content Delivery Network (CDN) and you used an ISO image to install Red Hat Ceph Storage, update the local repository with the latest version of Red Hat Ceph Storage. See Section 2.4, "Enabling the Red Hat Ceph Storage Repositories" for details.
- If upgrading from Red Hat Ceph Storage 2.x to 3.x, on the Ansible administration node and the RBD mirroring node, enable the Red Hat Ceph Storage 3 Tools repository:

```
[root@admin ~]$ sudo bash -c 'umask 0077; echo deb
https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/Tools
$(lsb_release -sc) main | tee /etc/apt/sources.list.d/Tools.list'
[root@admin ~]$ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-
key add -'
[root@admin ~]$ sudo apt-get update
```

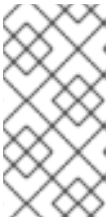
- If upgrading from RHCS 2.x to 3.x, or from RHCS 3.x to the latest version, on the Ansible administration node, ensure the latest version of the **ceph-ansible** package is installed.

```
[root@admin ~]$ sudo apt-get install ceph-ansible
```

- In the **rolling_update.yml** playbook, change the **health_osd_check_retries** and **health_osd_check_delay** values to **50** and **30** respectively.

```
health_osd_check_retries: 50
health_osd_check_delay: 30
```

With these values set, for each OSD node, Ansible will wait up to 25 minutes, and will check the storage cluster health every 30 seconds, waiting before continuing the upgrade process.



NOTE

Adjust the **health_osd_check_retries** option value up or down based on the used storage capacity of the storage cluster. For example, if you are using 218 TB out of 436 TB, basically using 50% of the storage capacity, then set the **health_osd_check_retries** option to **50**.

- If the cluster you want to upgrade contains Ceph Block Device images that use the **exclusive-lock** feature, ensure that all Ceph Block Device users have permissions to blacklist clients:

```
ceph auth caps client.<ID> mon 'allow r, allow command "osd blacklist"' osd '<existing-OSD-user-capabilities>'
```

4.1. UPGRADING THE STORAGE CLUSTER

Procedure

Use the following commands from the Ansible administration node.

1. As the **root** user, navigate to the **/usr/share/ceph-ansible/** directory:

```
[root@admin ~]# cd /usr/share/ceph-ansible/
```

2. Skip this step when upgrading from Red Hat Ceph Storage version 3.x to the latest version. Back up the **group_vars/all.yml** and **group_vars/osds.yml** files.

```
[root@admin ceph-ansible]# cp group_vars/all.yml group_vars/all_old.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml group_vars/osds_old.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml group_vars/clients_old.yml
```

3. Skip this step when upgrading from Red Hat Ceph Storage version 3.x to the latest version. When upgrading from Red Hat Ceph Storage 2.x to 3.x, create new copies of the **group_vars/all.yml.sample**, **group_vars/osds.yml.sample** and **group_vars/clients.yml.sample** files, and rename them to **group_vars/all.yml**, **group_vars/osds.yml**, and **group_vars/clients.yml** respectively. Open and edit them accordingly. For details, see Appendix I, *Changes in Ansible Variables Between Version 2 and 3* and Section 3.2, “Installing a Red Hat Ceph Storage Cluster” .

```
[root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml.sample group_vars/clients.yml
```

4. Skip this step when upgrading from Red Hat Ceph Storage version 3.x to the latest version. When upgrading from Red Hat Ceph Storage 2.x to 3.x, open the **group_vars/clients.yml** file, and uncomment the following lines:

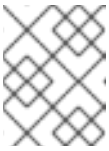
```
keys:
  - { name: client.test, caps: { mon: "allow r", osd: "allow class-read object_prefix
rbd_children, allow rwx pool=test" },  mode: "{{ ceph_keyring_permissions }}" }
```

a. Replace **client.test** with the real client name, and add the client key to the client definition line, for example:

```
key: "ADD-KEYRING-HERE=="
```

Now the whole line example would look similar to this:

```
- { name: client.test, key: "AQAin8tUMICVFBAALRHNrV0Z4MXupRw4v9JQ6Q==", caps:
{ mon: "allow r", osd: "allow class-read object_prefix rbd_children, allow rwx pool=test" },
mode: "{{ ceph_keyring_permissions }}" }
```



NOTE

To get the client key, run the **ceph auth get-or-create** command to view the key for the named client.

5. In the **group_vars/all.yml** file, uncomment the **upgrade_ceph_packages** option and set it to **True**.

```
upgrade_ceph_packages: True
```

6. Add the **fetch_directory** parameter to the **group_vars/all.yml** file.

```
fetch_directory: <full_directory_path>
```

Replace:

- **<full_directory_path>** with a writable location, such as the Ansible user's home directory.

7. If the cluster you want to upgrade contains any Ceph Object Gateway nodes, add the **radosgw_interface** parameter to the **group_vars/all.yml** file.

```
radosgw_interface: <interface>
```

Replace:

- **<interface>** with the interface that the Ceph Object Gateway nodes listen to.

8. Starting with Red Hat Ceph Storage 3.2, the default OSD object store is BlueStore. To keep the traditional OSD object store, you must explicitly set the **osd_objectstore** option to **filestore** in the **group_vars/all.yml** file.

```
osd_objectstore: filestore
```

NOTE

With the **osd_objectstore** option set to **filestore**, replacing an OSD will use FileStore, instead of BlueStore.

9. In the Ansible inventory file located at **/etc/ansible/hosts**, add the Ceph Manager (**ceph-mgr**) nodes under the **[mgrs]** section. Colocate the Ceph Manager daemon with Monitor nodes. Skip this step when upgrading from version 3.x to the latest version.

```
[mgrs]
<monitor-host-name>
<monitor-host-name>
<monitor-host-name>
```

10. Copy **rolling_update.yml** from the **infrastructure-playbooks** directory to the current directory.

```
[root@admin ceph-ansible]# cp infrastructure-playbooks/rolling_update.yml .
```

11. Create the **/var/log/ansible/** directory and assign the appropriate permissions for the **ansible** user:

```
[root@admin ceph-ansible]# mkdir /var/log/ansible
[root@admin ceph-ansible]# chown ansible:ansible  /var/log/ansible
[root@admin ceph-ansible]# chmod 755 /var/log/ansible
```

    a. Edit the **/usr/share/ceph-ansible/ansible.cfg** file, updating the **log_path** value as follows:

```
log_path = /var/log/ansible/ansible.log
```

12. As the Ansible user, run the playbook:

```
[user@admin ceph-ansible]$ ansible-playbook rolling_update.yml
```

    To use the playbook only for a particular group of nodes on the Ansible inventory file, use the **--limit** option. For details, see Section 3.8, “Understanding the **limit** option”.

13. While logged in as the **root** user on the RBD mirroring daemon node, upgrade **rbd-mirror** manually:

```
$ sudo apt-get upgrade rbd-mirror
```

    Restart the daemon:

```
# systemctl restart  ceph-rbd-mirror@<client-id>
```

14. Verify that the cluster health is OK. ..Log into a monitor node as the **root** user and run the ceph status command.

```
[root@monitor ~]# ceph -s
```

1. If working in an OpenStack environment, update all the **cephx** users to use the RBD profile for pools. The following commands must be run as the **root** user:

    - Glance users

```
ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd pool=<glance-pool-name>'
```

Example

```
[root@monitor ~]# ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd pool=images'
```

- Cinder users

```
ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd pool=<cinder-volume-pool-name>, profile rbd pool=<nova-pool-name>, profile rbd-read-only pool=<glance-pool-name>'
```

Example

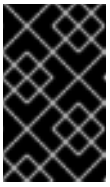
```
[root@monitor ~]# ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```

- OpenStack general users

```
ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-only pool=<cinder-volume-pool-name>, profile rbd pool=<nova-pool-name>, profile rbd-read-only pool=<glance-pool-name>'
```

Example

```
[root@monitor ~]# ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-only pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```



IMPORTANT

Do these CAPS updates before performing any live client migrations. This allows clients to use the new libraries running in memory, causing the old CAPS settings to drop from cache and applying the new RBD profile settings.

CHAPTER 5. WHAT TO DO NEXT?

This is only the beginning of what Red Hat Ceph Storage can do to help you meet the challenging storage demands of the modern data center. Here are links to more information on a variety of topics:

- Benchmarking performance and accessing performance counters, see the Benchmarking Performance chapter in the Administration Guide for Red Hat Ceph Storage 3.
- Creating and managing snapshots, see the Snapshots chapter in the Block Device Guide for Red Hat Ceph Storage 3.
- Expanding the Red Hat Ceph Storage cluster, see the Managing Cluster Size chapter in the Administration Guide for Red Hat Ceph Storage 3.
- Mirroring Ceph Block Devices, see the Block Device Mirroring chapter in the Block Device Guide for Red Hat Ceph Storage 3.
- Process management, see the Process Management chapter in the Administration Guide for Red Hat Ceph Storage 3.
- Tunable parameters, see the Configuration Guide for Red Hat Ceph Storage 3.
- Using Ceph as the back end storage for OpenStack, see the Back-ends section in the Storage Guide for Red Hat OpenStack Platform.

APPENDIX A. TROUBLESHOOTING

A.1. ANSIBLE STOPS INSTALLATION BECAUSE IT DETECTS LESS DEVICES THAN IT EXPECTED

The Ansible automation application stops the installation process and returns the following error:

```
- name: fix partitions gpt header or labels of the osd disks (autodiscover disks)
  shell: "sgdisk --zap-all --clear --mbrtogpt -- '/dev/{{ item.0.item.key }}' || sgdisk --zap-all --clear --
mbrtogpt -- '/dev/{{ item.0.item.key }}'"
  with_together:
    - "{{ osd_partition_status_results.results }}"
    - "{{ ansible_devices }}"
  changed_when: false
  when:
    - ansible_devices is defined
    - item.0.item.value.removable == "0"
    - item.0.item.value.partitions|count == 0
    - item.0.rc != 0
```

What this means:

When the **osd_auto_discovery** parameter is set to **true** in the **/etc/ansible/group_vars/osds.yml** file, Ansible automatically detects and configures all the available devices. During this process, Ansible expects that all OSDs use the same devices. The devices get their names in the same order in which Ansible detects them. If one of the devices fails on one of the OSDs, Ansible fails to detect the failed device and stops the whole installation process.

Example situation:

1. Three OSD nodes (**host1**, **host2**, **host3**) use the **/dev/sdb**, **/dev/sdc**, and **dev/sdd** disks.
2. On **host2**, the **/dev/sdc** disk fails and is removed.
3. Upon the next reboot, Ansible fails to detect the removed **/dev/sdc** disk and expects that only two disks will be used for **host2**, **/dev/sdb** and **/dev/sdc** (formerly **/dev/sdd**).
4. Ansible stops the installation process and returns the above error message.

To fix the problem:

In the **/etc/ansible/hosts** file, specify the devices used by the OSD node with the failed disk (**host2** in the Example situation above):

```
[osds]
host1
host2 devices="[ '/dev/sdb', '/dev/sdc' ]"
host3
```

See Chapter 3, *Deploying Red Hat Ceph Storage* for details.

APPENDIX B. MANUALLY INSTALLING RED HAT CEPH STORAGE

IMPORTANT

Red Hat does not support or test upgrading manually deployed clusters. Therefore, Red Hat recommends to use Ansible to deploy a new cluster with Red Hat Ceph Storage 3. See Chapter 3, *Deploying Red Hat Ceph Storage* for details.

You can use command-line utilities, such as **apt-get**, to upgrade manually deployed clusters, but Red Hat does not support or test this.

All Ceph clusters require at least one monitor, and at least as many OSDs as copies of an object stored on the cluster. Red Hat recommends using three monitors for production environments and a minimum of three Object Storage Devices (OSD).

Bootstrapping the initial monitor is the first step in deploying a Ceph storage cluster. Ceph monitor deployment also sets important criteria for the entire cluster, such as:

- The number of replicas for pools
- The number of placement groups per OSD
- The heartbeat intervals
- Any authentication requirement

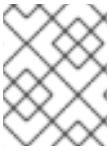
Most of these values are set by default, so it is useful to know about them when setting up the cluster for production.

Installing a Ceph storage cluster by using the command line interface involves these steps:

- Bootstrapping the initial Monitor node
- Adding an Object Storage Device (OSD) node

Configuring the Network Time Protocol for Red Hat Ceph Storage

All Ceph Monitor and OSD nodes requires configuring the Network Time Protocol (NTP). Ensure that Ceph nodes are NTP peers. NTP helps preempt issues that arise from clock drift.



NOTE

When using Ansible to deploy a Red Hat Ceph Storage cluster, Ansible automatically installs, configures, and enables NTP.

Prerequisites

- Network access to a valid time source.

Procedure: Configuring the Network Time Protocol for RHCS

Do the following steps on the all RHCS nodes in the storage cluster, as the **root** user.

1. Install the **ntp** package:

```
$ sudo apt-get install ntp
```

```
$ sudo systemctl start ntp
$ sudo systemctl enable ntp
```

1. Ensure that NTP is synchronizing clocks properly:

```
$ ntpq -p
```

Additional Resources

- The Configuring NTP Using ntpd chapter in the System Administrator's Guide for Red Hat Enterprise Linux 7.

MONITOR BOOTSTRAPPING

Bootstrapping a Monitor and by extension a Ceph storage cluster, requires the following data:

Unique Identifier

The File System Identifier (**fsid**) is a unique identifier for the cluster. The **fsid** was originally used when the Ceph storage cluster was principally used for the Ceph file system. Ceph now supports native interfaces, block devices, and object storage gateway interfaces too, so **fsid** is a bit of a misnomer.

Cluster Name

Ceph clusters have a cluster name, which is a simple string without spaces. The default cluster name is **ceph**, but you can specify a different cluster name. Overriding the default cluster name is especially useful when you work with multiple clusters.
When you run multiple clusters in a multi-site architecture, the cluster name for example, **us-west**, **us-east** identifies the cluster for the current command-line session.

NOTE

To identify the cluster name on the command-line interface, specify the Ceph configuration file with the cluster name, for example, **ceph.conf**, **us-west.conf**, **us-east.conf**, and so on.

Example:

```
# ceph --cluster us-west.conf ...
```

Monitor Name

Each Monitor instance within a cluster has a unique name. In common practice, the Ceph Monitor name is the node name. Red Hat recommend one Ceph Monitor per node, and no co-locating the Ceph OSD daemons with the Ceph Monitor daemon. To retrieve the short node name, use the **hostname -s** command.

Monitor Map

Bootstrapping the initial Monitor requires you to generate a Monitor map. The Monitor map requires:

- The File System Identifier (**fsid**)
- The cluster name, or the default cluster name of **ceph** is used

- At least one host name and its IP address.

Monitor Keyring

Monitors communicate with each other by using a secret key. You must generate a keyring with a Monitor secret key and provide it when bootstrapping the initial Monitor.

Administrator Keyring

To use the **ceph** command-line interface utilities, create the **client.admin** user and generate its keyring. Also, you must add the **client.admin** user to the Monitor keyring.

The foregoing requirements do not imply the creation of a Ceph configuration file. However, as a best practice, Red Hat recommends creating a Ceph configuration file and populating it with the **fsid**, the **mon initial members** and the **mon host** settings at a minimum.

You can get and set all of the Monitor settings at runtime as well. However, the Ceph configuration file might contain only those settings which overrides the default values. When you add settings to a Ceph configuration file, these settings override the default settings. Maintaining those settings in a Ceph configuration file makes it easier to maintain the cluster.

To bootstrap the initial Monitor, perform the following steps:

1. Enable the Red Hat Ceph Storage 3 Monitor repository:

```
$ sudo bash -c 'umask 0077; echo deb
https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/MON
$(lsb_release -sc) main | tee /etc/apt/sources.list.d/MON.list'
$ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-key add -'
$ sudo apt-get update
```

2. On your initial Monitor node, install the **ceph-mon** package as **root**:

```
$ sudo apt-get install ceph-mon
```

3. As **root**, create a Ceph configuration file in the **/etc/ceph/** directory. By default, Ceph uses **ceph.conf**, where **ceph** reflects the cluster name:

   Syntax

```
# touch /etc/ceph/<cluster_name>.conf
```

   Example

```
# touch /etc/ceph/ceph.conf
```

4. As **root**, generate the unique identifier for your cluster and add the unique identifier to the **[global]** section of the Ceph configuration file:

   Syntax

```
# echo "[global]" > /etc/ceph/<cluster_name>.conf
# echo "fsid = `uuidgen`" >> /etc/ceph/<cluster_name>.conf
```

   Example

```
# echo "[global]" > /etc/ceph/ceph.conf
# echo "fsid = `uuidgen`" >> /etc/ceph/ceph.conf
```

5. View the current Ceph configuration file:

```
$ cat /etc/ceph/ceph.conf
[global]
fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
```

6. As **root**, add the initial Monitor to the Ceph configuration file:

   Syntax

```
# echo "mon initial members = <monitor_host_name>[,<monitor_host_name>]" >>
/etc/ceph/<cluster_name>.conf
```

   Example

```
# echo "mon initial members = node1" >> /etc/ceph/ceph.conf
```

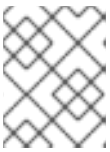
7. As **root**, add the IP address of the initial Monitor to the Ceph configuration file:

   Syntax

```
# echo "mon host = <ip-address>[,<ip-address>]" >> /etc/ceph/<cluster_name>.conf
```

   Example

```
# echo "mon host = 192.168.0.120" >> /etc/ceph/ceph.conf
```



NOTE

To use IPv6 addresses, you set the **ms bind ipv6** option to **true**. For details, see the Bind section in the Configuration Guide for Red Hat Ceph Storage 3.

8. As **root**, create the keyring for the cluster and generate the Monitor secret key:

   Syntax

```
# ceph-authtool --create-keyring /tmp/<cluster_name>.mon.keyring --gen-key -n mon. --cap
mon '<capabilites>'
```

   Example

```
# ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --cap mon 'allow *'
creating /tmp/ceph.mon.keyring
```

9. As **root**, generate an administrator keyring, generate a **<cluster_name>.client.admin.keyring** user and add the user to the keyring:

   Syntax

```
# ceph-authtool --create-keyring /etc/ceph/<cluster_name>.client.admin.keyring --gen-key -n
client.admin --set-uid=0 --cap mon '<capabilites>' --cap osd '<capabilites>' --cap mds
'<capabilites>'
```

Example

```
# ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring --gen-key -n
client.admin --set-uid=0 --cap mon 'allow *' --cap osd 'allow *' --cap mds 'allow'
creating /etc/ceph/ceph.client.admin.keyring
```

10. As **root**, add the **<cluster_name>.client.admin.keyring** key to the **<cluster_name>.mon.keyring**:

Syntax

```
# ceph-authtool /tmp/<cluster_name>.mon.keyring --import-keyring
/etc/ceph/<cluster_name>.client.admin.keyring
```

Example

```
# ceph-authtool /tmp/ceph.mon.keyring --import-keyring /etc/ceph/ceph.client.admin.keyring
importing contents of /etc/ceph/ceph.client.admin.keyring into /tmp/ceph.mon.keyring
```

11. Generate the Monitor map. Specify using the node name, IP address and the **fsid**, of the initial Monitor and save it as **/tmp/monmap**:

Syntax

```
$ monmaptool --create --add <monitor_host_name> <ip-address> --fsid <uuid>
/tmp/monmap
```

Example

```
$ monmaptool --create --add node1 192.168.0.120 --fsid a7f64266-0894-4f1e-a635-
d0aeaca0e993 /tmp/monmap
monmaptool: monmap file /tmp/monmap
monmaptool: set fsid to a7f64266-0894-4f1e-a635-d0aeaca0e993
monmaptool: writing epoch 0 to /tmp/monmap (1 monitors)
```

12. As **root** on the initial Monitor node, create a default data directory:

Syntax

```
# mkdir /var/lib/ceph/mon/<cluster_name>-<monitor_host_name>
```

Example

```
# mkdir /var/lib/ceph/mon/ceph-node1
```

13. As **root**, populate the initial Monitor daemon with the Monitor map and keyring:

Syntax

```
# ceph-mon [--cluster <cluster_name>] --mkfs -i <monitor_host_name> --monmap
/tmp/monmap --keyring /tmp/<cluster_name>.mon.keyring
```

Example

```
# ceph-mon --mkfs -i node1 --monmap /tmp/monmap --keyring /tmp/ceph.mon.keyring
ceph-mon: set fsid to a7f64266-0894-4f1e-a635-d0aeaca0e993
ceph-mon: created monfs at /var/lib/ceph/mon/ceph-node1 for mon.node1
```

14. View the current Ceph configuration file:

```
# cat /etc/ceph/ceph.conf
[global]
fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
mon_initial_members = node1
mon_host = 192.168.0.120
```

For more details on the various Ceph configuration settings, see the Configuration Guide for Red Hat Ceph Storage 3. The following example of a Ceph configuration file lists some of the most common configuration settings:

Example

```
[global]
fsid = <cluster-id>
mon initial members = <monitor_host_name>[, <monitor_host_name>]
mon host = <ip-address>[, <ip-address>]
public network = <network>[, <network>]
cluster network = <network>[, <network>]
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd journal size = <n>
osd pool default size = <n>  # Write an object n times.
osd pool default min size = <n> # Allow writing n copy in a degraded state.
osd pool default pg num = <n>
osd pool default pgp num = <n>
osd crush chooseleaf type = <n>
```

15. As **root**, create the **done** file:

Syntax

```
# touch /var/lib/ceph/mon/<cluster_name>-<monitor_host_name>/done
```

Example

```
# touch /var/lib/ceph/mon/ceph-node1/done
```

16. As **root**, update the owner and group permissions on the newly created directory and files:

Syntax

```
# chown -R <owner>:<group> <path_to_directory>
```

Example

```
# chown -R ceph:ceph /var/lib/ceph/mon
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown ceph:ceph /etc/ceph/ceph.client.admin.keyring
# chown ceph:ceph /etc/ceph/ceph.conf
# chown ceph:ceph /etc/ceph/rbdmap
```

NOTE

If the Ceph Monitor node is co-located with an OpenStack Controller node, then the Glance and Cinder keyring files must be owned by **glance** and **cinder** respectively. For example:

```
# ls -l /etc/ceph/
...
-rw-------.  1 glance glance      64 <date> ceph.client.glance.keyring
-rw-------.  1 cinder cinder      64 <date> ceph.client.cinder.keyring
...
```

17. For storage clusters with custom names, as **root**, add the the following line:

Syntax

```
$ sudo echo "CLUSTER=<custom_cluster_name>" >> /etc/default/ceph
```

Example

```
$ sudo echo "CLUSTER=test123" >> /etc/default/ceph
```

18. As **root**, start and enable the **ceph-mon** process on the initial Monitor node:

Syntax

```
$ sudo systemctl enable ceph-mon.target
$ sudo systemctl enable ceph-mon@<monitor_host_name>
$ sudo systemctl start ceph-mon@<monitor_host_name>
```

Example

```
$ sudo systemctl enable ceph-mon.target
$ sudo systemctl enable ceph-mon@node1
$ sudo systemctl start ceph-mon@node1
```

19. As **root**, verify the monitor daemon is running:

Syntax

```
# sudo systemctl status ceph-mon@<monitor_host_name>
```

Example

```
# sudo systemctl status ceph-mon@node1
● ceph-mon@node1.service - Ceph cluster monitor daemon
   Loaded: loaded (/usr/lib/systemd/system/ceph-mon@.service; enabled; vendor preset:
disabled)
   Active: active (running) since Wed 2018-06-27 11:31:30 PDT; 5min ago
 Main PID: 1017 (ceph-mon)
   CGroup: /system.slice/system-ceph\x2dmon.slice/ceph-mon@node1.service
           └─1017 /usr/bin/ceph-mon -f --cluster ceph --id node1 --setuser ceph --setgroup ceph

Jun 27 11:31:30 node1 systemd[1]: Started Ceph cluster monitor daemon.
Jun 27 11:31:30 node1 systemd[1]: Starting Ceph cluster monitor daemon...
```

To add more Red Hat Ceph Storage Monitors to the storage cluster, see the Adding a Monitor section in the Administration Guide for Red Hat Ceph Storage 3.

OSD BOOTSTRAPPING

Once you have your initial monitor running, you can start adding the Object Storage Devices (OSDs). Your cluster cannot reach an **active + clean** state until you have enough OSDs to handle the number of copies of an object.

The default number of copies for an object is three. You will need three OSD nodes at minimum. However, if you only want two copies of an object, therefore only adding two OSD nodes, then update the **osd pool default size** and **osd pool default min size** settings in the Ceph configuration file.

For more details, see the *OSD Configuration Reference* section in the *Configuration Guide* for Red Hat Ceph Storage 3.

After bootstrapping the initial monitor, the cluster has a default CRUSH map. However, the CRUSH map does not have any Ceph OSD daemons mapped to a Ceph node.

To add an OSD to the cluster and updating the default CRUSH map, execute the following on each OSD node:

1. Enable the Red Hat Ceph Storage 3 OSD repository:

```
$ sudo bash -c 'umask 0077; echo deb
https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/OSD
$(lsb_release -sc) main | tee /etc/apt/sources.list.d/OSD.list'
$ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-key add -'
$ sudo apt-get update
```

2. As **root**, install the **ceph-osd** package on the Ceph OSD node:

```
$ sudo apt-get install ceph-osd
```

3. Copy the Ceph configuration file and administration keyring file from the initial Monitor node to the OSD node:

   Syntax

```
# scp <user_name>@<monitor_host_name>:<path_on_remote_system>
<path_to_local_file>
```

Example

```
# scp root@node1:/etc/ceph/ceph.conf /etc/ceph
# scp root@node1:/etc/ceph/ceph.client.admin.keyring /etc/ceph
```

4. Generate the Universally Unique Identifier (UUID) for the OSD:

```
$ uuidgen
b367c360-b364-4b1d-8fc6-09408a9cda7a
```

5. As **root**, create the OSD instance:

Syntax

```
# ceph osd create <uuid> [<osd_id>]
```

Example

```
# ceph osd create b367c360-b364-4b1d-8fc6-09408a9cda7a
0
```

NOTE

This command outputs the OSD number identifier needed for subsequent steps.

6. As **root**, create the default directory for the new OSD:

Syntax

```
# mkdir /var/lib/ceph/osd/<cluster_name>-<osd_id>
```

Example

```
# mkdir /var/lib/ceph/osd/ceph-0
```

7. As **root**, prepare the drive for use as an OSD, and mount it to the directory you just created. Create a partition for the Ceph data and journal. The journal and the data partitions can be located on the same disk. This example is using a 15 GB disk:

Syntax

```
# parted <path_to_disk> mklabel gpt
# parted <path_to_disk> mkpart primary 1 10000
# mkfs -t <fstype> <path_to_partition>
# mount -o noatime <path_to_partition> /var/lib/ceph/osd/<cluster_name>-<osd_id>
# echo "<path_to_partition>  /var/lib/ceph/osd/<cluster_name>-<osd_id>   xfs
defaults,noatime 1 2" >> /etc/fstab
```

Example

```
# parted /dev/sdb mklabel gpt
# parted /dev/sdb mkpart primary 1 10000
# parted /dev/sdb mkpart primary 10001 15000
# mkfs -t xfs /dev/sdb1
# mount -o noatime /dev/sdb1 /var/lib/ceph/osd/ceph-0
# echo "/dev/sdb1 /var/lib/ceph/osd/ceph-0  xfs defaults,noatime 1 2" >> /etc/fstab
```

8. As **root**, initialize the OSD data directory:

Syntax

```
# ceph-osd -i <osd_id> --mkfs --mkkey --osd-uuid <uuid>
```

Example

```
# ceph-osd -i 0 --mkfs --mkkey --osd-uuid b367c360-b364-4b1d-8fc6-09408a9cda7a
... auth: error reading file: /var/lib/ceph/osd/ceph-0/keyring: can't open /var/lib/ceph/osd/ceph-0/keyring: (2) No such file or directory
... created new key in keyring /var/lib/ceph/osd/ceph-0/keyring
```

NOTE

The directory must be empty before you run **ceph-osd** with the **--mkkey** option. If you have a custom cluster name, the **ceph-osd** utility requires the **--cluster** option.

9. As **root**, register the OSD authentication key. If your cluster name differs from **ceph**, insert your cluster name instead:

Syntax

```
# ceph auth add osd.<osd_id> osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/<cluster_name>-<osd_id>/keyring
```

Example

```
# ceph auth add osd.0 osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/ceph-0/keyring
added key for osd.0
```

10. As **root**, add the OSD node to the CRUSH map:

Syntax

```
# ceph [--cluster <cluster_name>] osd crush add-bucket <host_name> host
```

Example

```
# ceph osd crush add-bucket node2 host
```

11. As **root**, place the OSD node under the **default** CRUSH tree:

    Syntax

    ```
    # ceph [--cluster <cluster_name>] osd crush move <host_name> root=default
    ```

    Example

    ```
    # ceph osd crush move node2 root=default
    ```

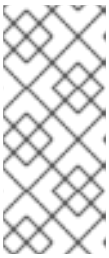
12. As **root**, add the OSD disk to the CRUSH map

    Syntax

    ```
    # ceph [--cluster <cluster_name>] osd crush add osd.<osd_id> <weight> [<bucket_type>=
    <bucket-name> ...]
    ```

    Example

    ```
    # ceph osd crush add osd.0 1.0 host=node2
    add item id 0 name 'osd.0' weight 1 at location {host=node2} to crush map
    ```

    NOTE

    You can also decompile the CRUSH map, and add the OSD to the device list. Add the OSD node as a bucket, then add the device as an item in the OSD node, assign the OSD a weight, recompile the CRUSH map and set the CRUSH map. For more details, see the Editing a CRUSH map section in the *Storage Strategies Guide* for Red Hat Ceph Storage 3. for more details.

13. As **root**, update the owner and group permissions on the newly created directory and files:

    Syntax

    ```
    # chown -R <owner>:<group> <path_to_directory>
    ```

    Example

    ```
    # chown -R ceph:ceph /var/lib/ceph/osd
    # chown -R ceph:ceph /var/log/ceph
    # chown -R ceph:ceph /var/run/ceph
    # chown -R ceph:ceph /etc/ceph
    ```

14. For storage clusters with custom names, as **root**, add the following line to the **/etc/default/ceph** file:

    Syntax

    ```
    $ sudo echo "CLUSTER=<custom_cluster_name>" >> /etc/default/ceph
    ```

    Example

```
$ sudo echo "CLUSTER=test123" >> /etc/default/ceph
```

15. The OSD node is in your Ceph storage cluster configuration. However, the OSD daemon is **down** and **in**. The new OSD must be **up** before it can begin receiving data. As **root**, enable and start the OSD process:

    Syntax

    ```
    $ sudo systemctl enable ceph-osd.target
    $ sudo systemctl enable ceph-osd@<osd_id>
    $ sudo systemctl start ceph-osd@<osd_id>
    ```

    Example

    ```
    $ sudo systemctl enable ceph-osd.target
    $ sudo systemctl enable ceph-osd@0
    $ sudo systemctl start ceph-osd@0
    ```

    Once you start the OSD daemon, it is **up** and **in**.

Now you have the monitors and some OSDs up and running. You can watch the placement groups peer by executing the following command:

```
$ ceph -w
```

To view the OSD tree, execute the following command:

```
$ ceph osd tree
```

Example

```
ID  WEIGHT   TYPE NAME       UP/DOWN  REWEIGHT  PRIMARY-AFFINITY
-1       2   root default
-2       2       host node2
 0       1           osd.0        up        1              1
-3       1       host node3
 1       1           osd.1        up        1              1
```

To expand the storage capacity by adding new OSDs to the storage cluster, see the Adding an OSD section in the *Administration Guide* for Red Hat Ceph Storage 3.

APPENDIX C. INSTALLING THE CEPH COMMAND LINE INTERFACE

The Ceph command-line interface (CLI) enables administrators to execute Ceph administrative commands. The CLI is provided by the **ceph-common** package and includes the following utilities:

- **ceph**
- **ceph-authtool**
- **ceph-dencoder**
- **rados**

Prerequisites

- A running Ceph storage cluster, preferably in the **active + clean** state.

Procedure

1. On the client node, enable the Red Hat Ceph Storage 3 Tools repository:

```
$ sudo bash -c 'umask 0077; echo deb
https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/Tools
$(lsb_release -sc) main | tee /etc/apt/sources.list.d/Tools.list'
$ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-key add -'
$ sudo apt-get update
```

2. On the client node, install the **ceph-common** package:

```
$ sudo apt-get install ceph-common
```

3. From the initial monitor node, copy the Ceph configuration file, in this case **ceph.conf**, and the administration keyring to the client node:

   ### Syntax

```
# scp /etc/ceph/<cluster_name>.conf <user_name>@<client_host_name>:/etc/ceph/
# scp /etc/ceph/<cluster_name>.client.admin.keyring
<user_name>@<client_host_name:/etc/ceph/
```

   ### Example

```
# scp /etc/ceph/ceph.conf root@node1:/etc/ceph/
# scp /etc/ceph/ceph.client.admin.keyring root@node1:/etc/ceph/
```

   Replace **<client_host_name>** with the host name of the client node.

APPENDIX D. MANUALLY INSTALLING CEPH MANAGER

Usually, the Ansible automation utility installs the Ceph Manager daemon (**ceph-mgr**) when you deploy the Red Hat Ceph Storage cluster. However, if you do not use Ansible to manage Red Hat Ceph Storage, you can install Ceph Manager manually. Red Hat recommends to colocate the Ceph Manager and Ceph Monitor daemons on a same node.

Prerequisites

- A working Red Hat Ceph Storage cluster
- **root** or **sudo** access
- The **rhel-7-server-rhceph-3-mon-rpms** repository enabled
- Open ports **6800-7300** on the public network if firewall is used

Procedure

Use the following commands on the node where **ceph-mgr** will be deployed and as the **root** user or with the **sudo** utility.

1. Install the **ceph-mgr** package:

```
[user@node1 ~]$ sudo apt-get install ceph-mgr
```

2. Create the **/var/lib/ceph/mgr/ceph-*hostname*/** directory:

```
mkdir /var/lib/ceph/mgr/ceph-hostname
```

Replace *hostname* with the host name of the node where the **ceph-mgr** daemon will be deployed, for example:

```
[user@node1 ~]$ sudo mkdir /var/lib/ceph/mgr/ceph-node1
```

3. In the newly created directory, create an authentication key for the **ceph-mgr** daemon:

```
[user@node1 ~]$ sudo ceph auth get-or-create mgr.`hostname -s` mon 'allow profile mgr'
osd 'allow *' mds 'allow *' -o /var/lib/ceph/mgr/ceph-node1/keyring
```

4. Change the owner and group of the **/var/lib/ceph/mgr/** directory to **ceph:ceph**:

```
[user@node1 ~]$ sudo chown -R ceph:ceph /var/lib/ceph/mgr
```

5. Enable the **ceph-mgr** target:

```
[user@node1 ~]$ sudo systemctl enable ceph-mgr.target
```

6. Enable and start the **ceph-mgr** instance:

```
systemctl enable ceph-mgr@hostname
systemctl start ceph-mgr@hostname
```

Replace *hostname* with the host name of the node where the **ceph-mgr** will be deployed, for example:

```
[user@node1 ~]$ sudo systemctl enable ceph-mgr@node1
[user@node1 ~]$ sudo systemctl start ceph-mgr@node1
```

7. Verify that the **ceph-mgr** daemon started successfully:

```
ceph -s
```

The output will include a line similar to the following one under the **services:** section:

```
    mgr: node1(active)
```

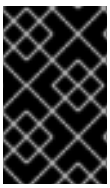
8. Install more **ceph-mgr** daemons to serve as standby daemons that become active if the current active daemon fails.

Additional resources

- *Requirements for Installing Red Hat Ceph Storage*

APPENDIX E. MANUALLY INSTALLING CEPH BLOCK DEVICE

The following procedure shows how to install and mount a thin-provisioned, resizable Ceph Block Device.



IMPORTANT

Ceph Block Devices must be deployed on separate nodes from the Ceph Monitor and OSD nodes. Running kernel clients and kernel server daemons on the same node can lead to kernel deadlocks.

Prerequisites

- Ensure to perform the tasks listed in the Appendix C, *Installing the Ceph Command Line Interface* section.
- If you use Ceph Block Devices as a back end for virtual machines (VMs) that use QEMU, increase the default file descriptor. See the Ceph - VM hangs when transferring large amounts of data to RBD disk Knowledgebase article for details.

Procedure

1. Create a Ceph Block Device user named **client.rbd** with full permissions to files on OSD nodes (**osd 'allow rwx'**) and output the result to a keyring file:

```
ceph auth get-or-create client.rbd mon 'profile rbd' osd 'profile rbd pool=<pool_name>' \
-o /etc/ceph/rbd.keyring
```

   Replace **<pool_name>** with the name of the pool that you want to allow **client.rbd** to have access to, for example **rbd**:

```
$ sudo ceph auth get-or-create \
client.rbd mon 'allow r' osd 'allow rwx pool=rbd' \
-o /etc/ceph/rbd.keyring
```

   See the *User Management* section in the Red Hat Ceph Storage 3 *Administration Guide* for more information about creating users.

2. Create a block device image:

```
rbd create <image_name> --size <image_size> --pool <pool_name> \
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

   Specify **<image_name>**, **<image_size>**, and **<pool_name>**, for example:

```
$ rbd create image1 --size 4096 --pool rbd \
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

WARNING

The default Ceph configuration includes the following Ceph Block Device features:

- **layering**
- **exclusive-lock**
- **object-map**
- **deep-flatten**
- **fast-diff**

If you use the kernel RBD (**krbd**) client, you will not be able to map the block device image because the current kernel version included in Red Hat Enterprise Linux 7.3 does not support **object-map**, **deep-flatten**, and **fast-diff**.

To work around this problem, disable the unsupported features. Use one of the following options to do so:

- Disable the unsupported features dynamically:

```
rbd feature disable <image_name> <feature_name>
```

  For example:

```
# rbd feature disable image1 object-map deep-flatten fast-diff
```

- Use the **--image-feature layering** option with the **rbd create** command to enable only **layering** on newly created block device images.
- Disable the features be default in the Ceph configuration file:

```
rbd_default_features = 1
```

This is a known issue, for details see the *Known Issues* chapter in the *Release Notes* for Red Hat Ceph Storage 3.

All these features work for users that use the user-space RBD client to access the block device images.

3. Map the newly created image to the block device:

```
rbd map <image_name> --pool <pool_name>\
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

For example:

-

```
$ sudo rbd map image1 --pool rbd --name client.rbd \
--keyring /etc/ceph/rbd.keyring
```

4. Use the block device by creating a file system:

```
mkfs.ext4 -m5 /dev/rbd/<pool_name>/<image_name>
```

Specify the pool name and the image name, for example:

```
$ sudo mkfs.ext4 -m5 /dev/rbd/rbd/image1
```

This can take a few moments.

5. Mount the newly created file system:

```
mkdir <mount_directory>
mount /dev/rbd/<pool_name>/<image_name> <mount_directory>
```

For example:

```
$ sudo mkdir /mnt/ceph-block-device
$ sudo mount /dev/rbd/rbd/image1 /mnt/ceph-block-device
```

For additional details, see the Block Device Guide for Red Hat Ceph Storage 3.

APPENDIX F. MANUALLY INSTALLING CEPH OBJECT GATEWAY

The Ceph object gateway, also know as the RADOS gateway, is an object storage interface built on top of the **librados** API to provide applications with a RESTful gateway to Ceph storage clusters.

Prerequisites

- A running Ceph storage cluster, preferably in the **active + clean** state.
- Perform the tasks listed in Chapter 2, *Requirements for Installing Red Hat Ceph Storage* .

Procedure

1. Enable the Red Hat Ceph Storage 3 Tools repository:

```
$ sudo bash -c 'umask 0077; echo deb
https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/Tools
$(lsb_release -sc) main | tee /etc/apt/sources.list.d/Tools.list'
$ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-key add -'
$ sudo apt-get update
```

2. On the Object Gateway node, install the **radosgw** package:

```
$ sudo apt-get install radosgw
```

3. On the initial Monitor node, do the following steps.

   a. Update the Ceph configuration file as follows:

```
[client.rgw.<obj_gw_hostname>]
host = <obj_gw_hostname>
rgw frontends = "civetweb port=80"
rgw dns name = <obj_gw_hostname>.example.com
```

   Where **<obj_gw_hostname>** is a short host name of the gateway node. To view the short host name, use the **hostname -s** command.

   b. Copy the updated configuration file to the new Object Gateway node and all other nodes in the Ceph storage cluster:

   Syntax

```
$ sudo scp /etc/ceph/<cluster_name>.conf
<user_name>@<target_host_name>:/etc/ceph
```

   Example

```
$ sudo scp /etc/ceph/ceph.conf root@node1:/etc/ceph/
```

   c. Copy the **<cluster_name>.client.admin.keyring** file to the new Object Gateway node:

   Syntax

   -

```
$ sudo scp /etc/ceph/<cluster_name>.client.admin.keyring
<user_name>@<target_host_name>:/etc/ceph/
```

Example

```
$ sudo scp /etc/ceph/ceph.client.admin.keyring root@node1:/etc/ceph/
```

4. On the Object Gateway node, create the data directory:

   Syntax

   ```
   $ sudo mkdir -p /var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname -s`
   ```

   Example

   ```
   $ sudo mkdir -p /var/lib/ceph/radosgw/ceph-rgw.`hostname -s`
   ```

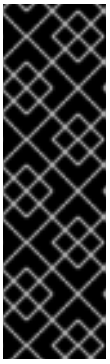
5. On the Object Gateway node, add a user and keyring to bootstrap the object gateway:

   Syntax

   ```
   $ sudo ceph auth get-or-create client.rgw.`hostname -s` osd 'allow rwx' mon 'allow rw' -o
   /var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname -s`/keyring
   ```

   Example

   ```
   $ sudo ceph auth get-or-create client.rgw.`hostname -s` osd 'allow rwx' mon 'allow rw' -o
   /var/lib/ceph/radosgw/ceph-rgw.`hostname -s`/keyring
   ```



IMPORTANT

When you provide capabilities to the gateway key you must provide the read capability. However, providing the Monitor write capability is optional; if you provide it, the Ceph Object Gateway will be able to create pools automatically.

In such a case, ensure to specify a reasonable number of placement groups in a pool. Otherwise, the gateway uses the default number, which might not be suitable for your needs. See Ceph Placement Groups (PGs) per Pool Calculator for details.

6. On the Object Gateway node, create the **done** file:

   Syntax

   ```
   $ sudo touch /var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname -s`/done
   ```

   Example

   ```
   $ sudo touch /var/lib/ceph/radosgw/ceph-rgw.`hostname -s`/done
   ```

7. On the Object Gateway node, change the owner and group permissions:

```
$ sudo chown -R ceph:ceph /var/lib/ceph/radosgw
$ sudo chown -R ceph:ceph /var/log/ceph
$ sudo chown -R ceph:ceph /var/run/ceph
$ sudo chown -R ceph:ceph /etc/ceph
```

8. For storage clusters with custom names, as **root**, add the following line:

   Syntax

   ```
   $ sudo echo "CLUSTER=<custom_cluster_name>" >> /etc/default/ceph
   ```

   Example

   ```
   $ sudo echo "CLUSTER=test123" >> /etc/default/ceph
   ```

9. On the Object Gateway node, open TCP port 80:

   ```
   $ sudo iptables -I INPUT 1 -i <network_interface> -p tcp -s <ip_address>/<netmask> --dport
   80 -j ACCEPT
   ```

10. On the Object Gateway node, start and enable the **ceph-radosgw** process:

    Syntax

    ```
    $ sudo systemctl enable ceph-radosgw.target
    $ sudo systemctl enable ceph-radosgw@rgw.<rgw_hostname>
    $ sudo systemctl start ceph-radosgw@rgw.<rgw_hostname>
    ```

    Example

    ```
    $ sudo systemctl enable ceph-radosgw.target
    $ sudo systemctl enable ceph-radosgw@rgw.node1
    $ sudo systemctl start ceph-radosgw@rgw.node1
    ```

Once installed, the Ceph Object Gateway automatically creates pools if the write capability is set on the Monitor. See the Pools chapter in the Storage Strategies Guide for information on creating pools manually.

Additional Details

- The Red Hat Ceph Storage 3 the Object Gateway Guide for Ubuntu

APPENDIX G. OVERRIDING CEPH DEFAULT SETTINGS

Unless otherwise specified in the Ansible configuration files, Ceph uses its default settings.

Because Ansible manages the Ceph configuration file, edit the **/etc/ansible/group_vars/all.yml** file to change the Ceph configuration. Use the **ceph_conf_overrides** setting to override the default Ceph configuration.

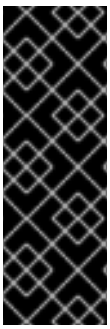
Ansible supports the same sections as the Ceph configuration file; **[global]**, **[mon]**, **[osd]**, **[mds]**, **[rgw]**, and so on. You can also override particular instances, such as a particular Ceph Object Gateway instance. For example:

```
###################
# CONFIG OVERRIDE #
###################

ceph_conf_overrides:
  client.rgw.rgw1:
    log_file: /var/log/ceph/ceph-rgw-rgw1.log
```

NOTE

Ansible does not include braces when referring to a particular section of the Ceph configuration file. Sections and settings names are terminated with a colon.



IMPORTANT

Do not set the cluster network with the **cluster_network** parameter in the **CONFIG OVERRIDE** section because this can cause two conflicting cluster networks being set in the Ceph configuration file.

To set the cluster network, use the **cluster_network** parameter in the **CEPH CONFIGURATION** section. For details, see Section 3.2, “Installing a Red Hat Ceph Storage Cluster”.

APPENDIX H. MANUALLY UPGRADING FROM RED HAT CEPH STORAGE 2 TO 3

This chapter provides information on how to manually upgrade from Red Hat Ceph storage version 2.4 to 3.0.

You can upgrade the Ceph Storage Cluster in a rolling fashion and while the cluster is running. Upgrade each node in the cluster sequentially, only proceeding to the next node after the previous node is done.

Red Hat recommends upgrading the Ceph components in the following order:

- Monitor nodes
- OSD nodes
- Ceph Object Gateway nodes
- All other Ceph client nodes

Two methods are available to upgrade a Red Hat Ceph Storage 2.3 to 3:

- Using Red Hat's Content Delivery Network (CDN)
- Using a Red Hat provided ISO image file

After upgrading the storage cluster you might have a health warning regarding the CRUSH map using legacy tunables. For details, see the CRUSH Tunables section in the Storage Strategies guide for Red Hat Ceph Storage 3.

Example

```
$ ceph -s
    cluster 848135d7-cdb9-4084-8df2-fb5e41ae60bd
     health HEALTH_WARN
            crush map has legacy tunables (require bobtail, min is firefly)
     monmap e1: 1 mons at {ceph1=192.168.0.121:6789/0}
            election epoch 2, quorum 0 ceph1
     osdmap e83: 2 osds: 2 up, 2 in
      pgmap v1864: 64 pgs, 1 pools, 38192 kB data, 17 objects
            10376 MB used, 10083 MB / 20460 MB avail
                  64 active+clean
```

IMPORTANT

Red Hat recommends all Ceph clients to be running the same version as the Ceph storage cluster.

Prerequisites

- If the cluster you want to upgrade contains Ceph Block Device images that use the **exclusive-lock** feature, ensure that all Ceph Block Device users have permissions to blacklist clients:

  ```
  ceph auth caps client.<ID> mon 'allow r, allow command "osd blacklist"' osd '<existing-OSD-user-capabilities>'
  ```

Upgrading Monitor Nodes

This section describes steps to upgrade a Ceph Monitor node to a later version. There must be an odd number of Monitors. While you are upgrading one Monitor, the storage cluster will still have quorum.

Procedure

Do the following steps on each Monitor node in the storage cluster. Upgrade only one Monitor node at a time.

1. If you installed Red Hat Ceph Storage 2 by using software repositories, disable the repositories:

   a. If the following lines exist in the **/etc/apt/sources.list** or **/etc/apt/sources.list.d/ceph.list** files, comment out the online repositories for Red Hat Ceph Storage 2 by adding a hash sign (**#**) to the beginning of the line.

   ```
   deb https://<customer_name>:
   <customer_password>@rhcs.download.redhat.com/ubuntu/2-updates/Installer
   deb https://<customer_name>:
   <customer_password>@rhcs.download.redhat.com/ubuntu/2-updates/Tools
   ```

   b. Remove the following files from the **/etc/apt/sources.list.d/** directory:

   ```
   Installer.list
   Tools.list
   ```

2. Enable the Red Hat Ceph Storage 3 Monitor repository:

   ```
   $ sudo bash -c 'umask 0077; echo deb
   https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/MON
   $(lsb_release -sc) main | tee /etc/apt/sources.list.d/MON.list'
   $ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-key add -'
   $ sudo apt-get update
   ```

3. As **root**, stop the Monitor process:

   Syntax

   ```
   $ sudo stop ceph-mon id=<monitor_host_name>
   ```

   Example

   ```
   $ sudo stop ceph-mon id=node1
   ```

4. As **root**, update the **ceph-mon** package:

   ```
   $ sudo apt-get update
   $ sudo apt-get dist-upgrade
   $ sudo apt-get install ceph-mon
   ```

   a. Verify the latest Red Hat version is installed:

   ```
   $ dpkg -s ceph-base | grep Version
   Version: 10.2.2-19redhat1trusty
   ```

5. As **root**, update the owner and group permissions:

   Syntax

   ```
   # chown -R <owner>:<group> <path_to_directory>
   ```

   Example

   ```
   # chown -R ceph:ceph /var/lib/ceph/mon
   # chown -R ceph:ceph /var/log/ceph
   # chown -R ceph:ceph /var/run/ceph
   # chown ceph:ceph /etc/ceph/ceph.client.admin.keyring
   # chown ceph:ceph /etc/ceph/ceph.conf
   # chown ceph:ceph /etc/ceph/rbdmap
   ```

   NOTE

   If the Ceph Monitor node is colocated with an OpenStack Controller node, then the Glance and Cinder keyring files must be owned by **glance** and **cinder** respectively. For example:

   ```
   # ls -l /etc/ceph/
   ...
   -rw-------.  1 glance glance      64 <date> ceph.client.glance.keyring
   -rw-------.  1 cinder cinder      64 <date> ceph.client.cinder.keyring
   ...
   ```

6. Remove packages that are no longer needed:

   ```
   $ sudo apt-get purge ceph ceph-osd
   ```

7. As **root**, replay device events from the kernel:

   ```
   # udevadm trigger
   ```

8. As **root**, enable the **ceph-mon** process:

   ```
   $ sudo systemctl enable ceph-mon.target
   $ sudo systemctl enable ceph-mon@<monitor_host_name>
   ```

9. As **root**, reboot the Monitor node:

   ```
   # shutdown -r now
   ```

10. Once the Monitor node is up, check the health of the Ceph storage cluster before moving to the next Monitor node:

    ```
    # ceph -s
    ```

Upgrading OSD Nodes

This section describes steps to upgrade a Ceph OSD node to a later version.

Prerequisites

When upgrading an OSD node, some placement groups will become degraded because the OSD might be down or restarting. To prevent Ceph from starting the recovery process, on a Monitor node, set the **noout** and **norebalance** OSD flags:

```
[root@monitor ~]# ceph osd set noout
[root@monitor ~]# ceph osd set norebalance
```

Procedure

Do the following steps on each OSD node in the storage cluster. Upgrade only one OSD node at a time. If an ISO-based installation was performed for Red Hat Ceph Storage 2.3, then skip this first step.

1. As **root**, disable the Red Hat Ceph Storage 2 repositories:

   a. If the following lines exist in the **/etc/apt/sources.list** or **/etc/apt/sources.list.d/ceph.list** files, comment out the online repositories for Red Hat Ceph Storage 2 by adding a hash sign (**#**) to the beginning of the line.

   ```
   deb https://<customer_name>:
   <customer_password>@rhcs.download.redhat.com/ubuntu/2-updates/Installer
   deb https://<customer_name>:
   <customer_password>@rhcs.download.redhat.com/ubuntu/2-updates/Tools
   ```

   b. Remove the following files from the /etc/apt/sources.list.d/ directory:

   ```
   Installer.list
   Tools.list
   ```

   NOTE

   Remove any reference to Red Hat Ceph Storage 2 in the APT source file(s). If an ISO-based installation was performed for Red Hat Ceph Storage 2, then skip this first step.

2. Enable the Red Hat Ceph Storage 3 OSD repository:

   ```
   $ sudo bash -c 'umask 0077; echo deb
   https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/OSD
   $(lsb_release -sc) main | tee /etc/apt/sources.list.d/OSD.list'
   $ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-key add -'
   $ sudo apt-get update
   ```

3. As **root**, stop any running OSD process:

   Syntax

   ```
   $ sudo stop ceph-osd id=<osd_id>
   ```

   Example

   ```
   $ sudo stop ceph-osd id=0
   ```

4. As **root**, update the **ceph-osd** package:

```
$ sudo apt-get update
$ sudo apt-get dist-upgrade
$ sudo apt-get install ceph-osd
```

   a. Verify the latest Red Hat version is installed:

```
$ dpkg -s ceph-base | grep Version
Version: 10.2.2-19redhat1trusty
```

5. As **root**, update the owner and group permissions on the newly created directory and files:

   Syntax

```
# chown -R <owner>:<group> <path_to_directory>
```

   Example

```
# chown -R ceph:ceph /var/lib/ceph/osd
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown -R ceph:ceph /etc/ceph
```

NOTE

Using the following **find** command might quicken the process of changing ownership by using the **chown** command in parallel on a Ceph storage cluster with a large number of disks:

```
# find /var/lib/ceph/osd -maxdepth 1 -mindepth 1 -print | xargs -P12 -n1 chown -R ceph:ceph
```

6. Remove packages that are no longer needed:

```
$ sudo apt-get purge ceph ceph-mon
```

NOTE

The **ceph** package is now a meta-package. Only the **ceph-mon** package is needed on the Monitor nodes, only the **ceph-osd** package is needed on the OSD nodes, and only the **ceph-radosgw** package is needed on the RADOS Gateway nodes.

7. As **root**, replay device events from the kernel:

```
# udevadm trigger
```

8. As **root**, enable the **ceph-osd** process:

```
$ sudo systemctl enable ceph-osd.target
$ sudo systemctl enable ceph-osd@<osd_id>
```

9. As **root**, reboot the OSD node:

```
# shutdown -r now
```

10. Move to the next OSD node.

NOTE

If the **noout** and **norebalance** flags are set, the storage cluster is in **HEALTH_WARN** state

```
$ ceph health
HEALTH_WARN noout,norebalance flag(s) set
```

Once you are done upgrading the Ceph Storage Cluster, unset the previously set OSD flags and verify the storage cluster status.

On a Monitor node, and after all OSD nodes have been upgraded, unset the **noout** and **norebalance** flags:

```
# ceph osd unset noout
# ceph osd unset norebalance
```

In addition, execute the **ceph osd require-osd-release <release>** command. This command ensures that no more OSDs with Red Hat Ceph Storage 2.3 can be added to the storage cluster. If you do not run this command, the storage status will be **HEALTH_WARN**.

```
# ceph osd require-osd-release luminous
```

Additional Resources

- To expand the storage capacity by adding new OSDs to the storage cluster, see the *Add an OSD* section in the *Administration Guide* for Red Hat Ceph Storage 3

Upgrading the Ceph Object Gateway Nodes

This section describes steps to upgrade a Ceph Object Gateway node to a later version.

IMPORTANT

Red Hat recommends to back up the system before proceeding with these upgrade procedures.

Prerequisites

- Red Hat recommends putting a Ceph Object Gateway behind a load balancer, such as HAProxy. If you use a load balancer, remove the Ceph Object Gateway from the load balancer once no requests are being served.

- If you use a custom name for the region pool, specified in the **rgw_region_root_pool**

parameter, add the **rgw_zonegroup_root_pool** parameter to the **[global]** section of the Ceph configuration file. Set the value of **rgw_zonegroup_root_pool** to be the same as **rgw_region_root_pool**, for example:

```
[global]
rgw_zonegroup_root_pool = .us.rgw.root
```

Procedure

Do the following steps on each Ceph Object Gateway node in the storage cluster. Upgrade only one node at a time.

1. If you used online repositories to install Red Hat Ceph Storage, disable the 2 repositories.

   a. Comment out the following lines in the **/etc/apt/sources.list** and **/etc/apt/sources.list.d/ceph.list** files.

   ```
   # deb https://<customer_name>:
   <customer_password>@rhcs.download.redhat.com/ubuntu/2-updates/Installer
   # deb https://<customer_name>:
   <customer_password>@rhcs.download.redhat.com/ubuntu/2-updates/Tools
   ```

   b. Remove the following files from the **/etc/apt/sources.list.d/** directory.

   ```
   # rm /etc/apt/sources.list.d/Installer.list
   # rm /etc/apt/sources.list.d/Tools.list
   ```

2. Enable the Red Hat Ceph Storage 3 Tools repository:

   ```
   $ sudo bash -c 'umask 0077; echo deb
   https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/Tools
   $(lsb_release -sc) main | tee /etc/apt/sources.list.d/Tools.list'
   $ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-key add -'
   $ sudo apt-get update
   ```

3. Stop the Ceph Object Gateway process (**ceph-radosgw**):

   ```
   $ sudo stop radosgw id=rgw.<hostname>
   ```

   Replace **<hostname>** with the name of Ceph Object Gateway host, for example **gateway-node**.

   ```
   $ sudo stop radosgw id=rgw.node
   ```

4. Update the **ceph-radosgw** package:

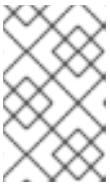
   ```
   $ sudo apt-get update
   $ sudo apt-get dist-upgrade
   $ sudo apt-get install radosgw
   ```

5. Change the owner and group permissions on the newly created **/var/lib/ceph/radosgw/** and **/var/log/ceph/** directories and their content to **ceph**.

```
# chown -R ceph:ceph /var/lib/ceph/radosgw
# chown -R ceph:ceph /var/log/ceph
```

6. Remove packages that are no longer needed.

```
$ sudo apt-get purge ceph
```



NOTE

The **ceph** package is now a meta-package. Only the **ceph-mon**, **ceph-osd**, and **ceph-radosgw** packages are required on the Monitor, OSD, and Ceph Object Gateway nodes respectively.

7. Enable the **ceph-radosgw** process:

```
$ sudo systemctl enable ceph-radosgw.target
$ sudo systemctl enable ceph-radosgw@rgw.<hostname>
```

Replace **<hostname>** with the name of the Ceph Object Gateway host, for example **gateway-node**.

```
$ sudo systemctl enable ceph-radosgw.target
$ sudo systemctl enable ceph-radosgw@rgw.gateway-node
```

8. Reboot the Ceph Object Gateway node:

```
# shutdown -r now
```

9. If you use a load balancer, add the Ceph Object Gateway node back to the load balancer.

See Also

- The *Ceph Object Gateway Guide for Ubuntu*

Upgrading a Ceph Client Node

Ceph clients are:

- Ceph Block Devices
- OpenStack Nova compute nodes
- QEMU/KVM hypervisors
- Any custom application that uses the Ceph client-side libraries

Red Hat recommends all Ceph clients to be running the same version as the Ceph storage cluster.

Prerequisites

- Stop all I/O requests against a Ceph client node while upgrading the packages to prevent unexpected errors to occur

Procedure

1. If you installed Red Hat Ceph Storage 2 clients by using software repositories, disable the repositories:

   a. If the following lines exist in the **/etc/apt/sources.list** or **/etc/apt/sources.list.d/ceph.list** files, comment out the online repositories for Red Hat Ceph Storage 2 by adding a hash sign (**#**) to the beginning of the line.

   ```
   deb https://<customer_name>:
   <customer_password>@rhcs.download.redhat.com/ubuntu/2-updates/Installer
   deb https://<customer_name>:
   <customer_password>@rhcs.download.redhat.com/ubuntu/2-updates/Tools
   ```

   b. Remove the following files from the /etc/apt/sources.list.d/ directory:

   ```
   Installer.list
   Tools.list
   ```

   NOTE

   Remove any reference to Red Hat Ceph Storage 2 in the APT source file(s).

2. On the client node, enable the Red Hat Ceph Storage Tools 3 repository:

   ```
   $ sudo bash -c 'umask 0077; echo deb
   https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/Tools
   $(lsb_release -sc) main | tee /etc/apt/sources.list.d/Tools.list'
   $ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt | apt-key add -'
   $ sudo apt-get update
   ```

3. On the client node, update the **ceph-common** package:

   ```
   $ sudo apt-get install ceph-common
   ```

Restart any application that depends on the Ceph client-side libraries after upgrading the **ceph-common** package.

NOTE

If you are upgrading OpenStack Nova compute nodes that have running QEMU/KVM instances or use a dedicated QEMU/KVM client, stop and start the QEMU/KVM instance because restarting the instance does not work in this case.

APPENDIX I. CHANGES IN ANSIBLE VARIABLES BETWEEN VERSION 2 AND 3

With Red Hat Ceph Storage 3, certain variables in the configuration files located in the **/usr/share/ceph-ansible/group_vars/** directory have changed or have been removed. The following table lists all the changes. After upgrading to version 3, copy the **all.yml.sample** and **osds.yml.sample** files again to reflect these changes. See Upgrading a Red Hat Ceph Storage Cluster for details.

Old Option	New Option	File
ceph_rhcs_cdn_install	**ceph_repository_type: cdn**	**all.yml**
ceph_rhcs_iso_install	**ceph_repository_type: iso**	**all.yml**
ceph_rhcs	**ceph_origin: repository** and **ceph_repository: rhcs** (enabled by default)	**all.yml**
journal_collocation	**osd_scenario: collocated**	**osds.yml**
raw_multi_journal	**osd_scenario: non-collocated**	**osds.yml**
raw_journal_devices	**dedicated_devices**	**osds.yml**
dmcrytpt_journal_collocation	**dmcrypt: true** + **osd_scenario: collocated**	**osds.yml**
dmcrypt_dedicated_journal	**dmcrypt: true** + **osd_scenario: non-collocated**	**osds.yml**

APPENDIX J. IMPORTING AN EXISTING CEPH CLUSTER TO ANSIBLE

You can configure Ansible to use a cluster deployed without Ansible. For example, if you upgraded Red Hat Ceph Storage 1.3 clusters to version 2 manually, configure them to use Ansible by following this procedure:

1. After manually upgrading from version 1.3 to version 2, install and configure Ansible on the administration node.

2. Ensure that the Ansible administration node has passwordless **ssh** access to all Ceph nodes in the cluster. See Section 2.10, "Enabling Password-less SSH for Ansible" for more details.

3. As **root**, create a symbolic link to the Ansible **group_vars** directory in the **/etc/ansible/** directory:

```
# ln -s /usr/share/ceph-ansible/group_vars /etc/ansible/group_vars
```

4. As **root**, create an **all.yml** file from the **all.yml.sample** file and open it for editing:

```
# cd /etc/ansible/group_vars
# cp all.yml.sample all.yml
# vim all.yml
```

5. Set the **generate_fsid** setting to **false** in **group_vars/all.yml**.

6. Get the current cluster **fsid** by executing **ceph fsid**.

7. Set the retrieved **fsid** in **group_vars/all.yml**.

8. Modify the Ansible inventory in **/etc/ansible/hosts** to include Ceph hosts. Add monitors under a **[mons]** section, OSDs under an **[osds]** section and gateways under an **[rgws]** section to identify their roles to Ansible.

9. Make sure **ceph_conf_overrides** is updated with the original **ceph.conf** options used for **[global]**, **[osd]**, **[mon]**, and **[client]** sections in the **all.yml** file.
   Options like **osd journal**, **public_network** and **cluster_network** should not be added in **ceph_conf_overrides** because they are already part of **all.yml**. Only the options that are not part of **all.yml** and are in the original **ceph.conf** should be added to **ceph_conf_overrides**.

10. From the **/usr/share/ceph-ansible/** directory run the playbook.

```
# cd /usr/share/ceph-ansible/
# cp infrastructure-playbooks/take-over-existing-cluster.yml .
$ ansible-playbook take-over-existing-cluster.yml -u <username>
```

APPENDIX K. PURGING A CEPH CLUSTER BY USING ANSIBLE

If you deployed a Ceph cluster using Ansible and you want to purge the cluster, then use the **purge-cluster.yml** Ansible playbook located in the **infrastructure-playbooks** directory.

IMPORTANT

Purging a Ceph cluster will lose data stored on the cluster's OSDs.

Before purging the Ceph cluster...

Check the **osd_auto_discovery** option in the **osds.yml** file. Having this option set to **true** will cause the purge to fail. To prevent the failure, do the following steps before running the purge:

1. Declare the OSD devices in the **osds.yml** file. See Section 3.2, "Installing a Red Hat Ceph Storage Cluster" for more details.
2. Comment out the **osd_auto_discovery** option in the **osds.yml** file.

To purge the Ceph cluster...

1. As **root**, navigate to the **/usr/share/ceph-ansible/** directory:

```
# cd /usr/share/ceph-ansible
```

2. As **root**, copy the **purge-cluster.yml** Ansible playbook to the current directory:

```
# cp infrastructure-playbooks/purge-cluster.yml .
```

3. Run the **purge-cluster.yml** Ansible playbook:

```
$ ansible-playbook purge-cluster.yml
```