Installing Red Hat Ceph Storage on Ubuntu
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? ........................................... 3.

## CHAPTER 2. PREREQUISITES ................................................................. 5.
1. OPERATING SYSTEM ............................................................................. 6
   1.1. Adjusting the PID Count ................................................................. 6
2. ENABLING CEPH REPOSITORIES ...................................................... 7
   2.1. Online Repositories ................................................................. 7
   2.2. Local Repository ..................................................................... 8
3. CONFIGURING RAID CONTROLLERS ................................................. 9
4. CONFIGURING NETWORK .................................................................... 9
5. SETTING DNS NAME RESOLUTION .................................................. 10
6. CONFIGURING FIREWALL .................................................................. 10
7. CONFIGURING NETWORK TIME PROTOCOL ..................................... 12
8. CREATING AN ANSIBLE USER (ANSIBLE DEPLOYMENT ONLY) ............ 12
9. ENABLING PASSWORD-LESS SSH (ANSIBLE DEPLOYMENT ONLY) .......... 13

## CHAPTER 3. STORAGE CLUSTER INSTALLATION .......................................... 15.
1. INSTALLING RED HAT CEPH STORAGE USING THE RED HAT STORAGE CONSOLE 15
   1.1. Installing and Configuring the Red Hat Storage Console Agent ............... 15
   1.2. Importing an existing Ceph Storage Cluster ....................................... 19
2. INSTALLING RED HAT CEPH STORAGE USING ANSIBLE ......................... 19
3. INSTALLING RED HAT CEPH STORAGE BY USING THE COMMAND-LINE INTERFACE 19
   3.3. Calamari Server Installation ...................................................... 31

## CHAPTER 4. CLIENT INSTALLATION ............................................................ 33.
1. INSTALLING THE CEPH-CLIENT ROLE .............................................. 33
2. CEPH COMMAND-LINE INTERFACE INSTALLATION ................................. 33
3. CEPH BLOCK DEVICE INSTALLATION ............................................... 34
   Before you start ........................................................................... 34
   Installing Ceph Block Devices by Using the Command Line ....................... 34
4. CEPH OBJECT GATEWAY INSTALLATION ........................................ 37
   4.4.1. Installing Ceph Object Gateway by using Ansible ......................... 37
   4.4.2. Installing Ceph Object Gateway Manually ..................................... 37

## CHAPTER 5. UPGRADING CEPH STORAGE CLUSTER ..................................... 41.
1. UPGRADING FROM RED HAT CEPH STORAGE 1.3 TO 2 ............................ 41
   1.1. Upgrading a Ceph Monitor Node ................................................. 41
   1.2. Upgrading a Ceph OSD Node .................................................... 45
   1.3. Upgrading the Ceph Object Gateway Nodes ................................. 48
      Before You Start ..................................................................... 48
      Procedure: Upgrading the Ceph Object Gateway Node ......................... 49
      See Also ............................................................................. 51
   1.4. Upgrading a Ceph Client Node .................................................... 51
   1.5. Repurposing the Ceph Administration Node .................................... 52
2. UPGRADING BETWEEN MINOR VERSIONS AND APPLYING ASYNCHRONOUS UPDATES 53
   2.1. Changes Between Ansible 2.1 and 2.2 ........................................... 54

## CHAPTER 6. WHAT TO DO NEXT? ............................................................ 56.
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 Storage Console and Ansible 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
  - Red Hat Storage Console
  - Ansible administration
  - Ceph Client Command line interface
- The Ceph configuration files and keys
- Optionally, local repositories for installing Ceph on nodes that cannot access the Internet for security reasons

**NOTE**

In Red Hat Ceph Storage 1.3.x, the Ceph Administration node hosted the Calamari monitoring and administration server, and the `ceph-deploy` utility, which has been deprecated in Red Hat Ceph Storage 2. Use the Red Hat Storage Console, Ceph command-line utilities or Ansible automation utility instead. See Section 5.1.5, “Repurposing the Ceph Administration Node” for details on repurposing the legacy Ceph Administration node.

**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.

Ceph can run with one monitor; however, to ensure high availability in a production cluster, Red Hat recommends to deploy 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.

MDS and CephFS are Technology Preview features and as such they are not fully supported yet. For information on MDS installation and configuration, see the [Ceph File System Guide (Technology Preview)](https://docs.ceph.com/docs/v10.0.0/ceph-filesystem-guide/).

**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 RADOS 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](https://docs.ceph.com/docs/v10.0.0/architectural-overview/).

For minimum recommended hardware, see the [Hardware Guide](https://docs.ceph.com/docs/v10.0.0/architectural-overview/).
Figure 2.1. Prerequisite Workflow

Before installing Red Hat Ceph Storage, review the following prerequisites first and prepare each Ceph Monitor, OSD, and client nodes accordingly.

Table 2.1. Prerequisites Checks

<table>
<thead>
<tr>
<th>Task</th>
<th>Required</th>
<th>Section</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifying the operating system version</td>
<td>Yes</td>
<td>Section 2.1, “Operating System”</td>
<td>Verify the PID count</td>
</tr>
<tr>
<td>Enabling Ceph software repositories</td>
<td>Yes</td>
<td>Section 2.2, “Enabling Ceph Repositories”</td>
<td>Two installation methods:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Content Delivery Network (CDN)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Local Repository (ISO)</td>
</tr>
<tr>
<td>Using a RAID controller</td>
<td>No</td>
<td>Section 2.3, “Configuring RAID Controllers”</td>
<td>For OSD nodes only.</td>
</tr>
<tr>
<td>Configuring network Interface</td>
<td>Yes</td>
<td>Section 2.4, “Configuring Network”</td>
<td>Using a public network is required. Having a private network for cluster communication is optional, but recommended.</td>
</tr>
<tr>
<td>Resolving short host names</td>
<td>Yes</td>
<td>Section 2.5, “Setting DNS Name Resolution”</td>
<td></td>
</tr>
<tr>
<td>Configuring a firewall</td>
<td>No</td>
<td>Section 2.6, “Configuring Firewall”</td>
<td></td>
</tr>
</tbody>
</table>
2.1. OPERATING SYSTEM

Red Hat Ceph Storage 2 and later requires Ubuntu 16.04 with a homogeneous version running on AMD64 and Intel 64 architectures for all Ceph nodes, including the Red Hat Ceph Storage node.

**IMPORTANT**

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

_Return to prerequisite checklist_

2.1.1. Adjusting the PID Count

Hosts with high numbers of OSDs, that being more than 12, may spawn a lot of threads, especially during recovery and re-balancing events. The kernel defaults to a relatively small maximum number of threads, typically 32768.

1. Check the current `pid_max` settings:

   ```
   # cat /proc/sys/kernel/pid_max
   ```

2. As `root`, consider setting `kernel.pid_max` to a higher number of threads. The theoretical maximum is 4,194,303 threads. For example, add the following to the `/etc/sysctl.conf` file to set it to the maximum value:

   ```
   kernel.pid_max = 4194303
   ```

3. As `root`, to load the changes without a rebooting:

   ```
   # sysctl -p
   ```

4. As `root`, verify the changes:

   ```
   # sysctl -a | grep kernel.pid_max
   ```
2.2. ENABLING CEPH REPOSITORIES

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

- **Online Repositories**
  For Ceph Storage clusters with Ceph nodes that can connect directly to the Internet, you can use online repositories from the https://rhcs.download.redhat.com/ubuntu site. You will need your **Customer Name** and **Customer Password** received from https://rhcs.download.redhat.com to be able to use the repositories.

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

- **Local Repository**
  For Ceph Storage clusters where security measures preclude nodes from accessing the Internet, install Red Hat Ceph Storage 2 from a single software build delivered as an ISO image, which will allow you to install local repositories.

### 2.2.1. Online Repositories

**Online Installations for...**

- **Monitor Nodes**
  As **root**, enable the Red Hat Ceph Storage 2 Monitor repository:

  ```bash
  $ sudo bash -c 'umask 0077; echo deb https://customername:customerpasswd@rhcs.download.redhat.com/2-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
  ```

- **OSD Nodes**
  As **root**, enable the Red Hat Ceph Storage 2 OSD repository:

  ```bash
  $ sudo bash -c 'umask 0077; echo deb https://customername:customerpasswd@rhcs.download.redhat.com/2-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
  ```

- **RADOS Gateway and Client Nodes**
  As **root**, enable the Red Hat Ceph Storage 2 Tools repository:

  ```bash
  $ sudo bash -c 'umask 0077; echo deb https://customername:customerpasswd@rhcs.download.redhat.com/2-updates/Tools $(lsb_release -sc) main | tee
  ```
Red Hat Storage Console Agent
For all Ceph Monitor and OSD nodes being managed by Red Hat Storage Console, as root, enable the Red Hat Storage Console 2 Agent repository:

$ sudo bash -c 'umask 0077; echo deb https://customername:customerpasswd@rhcs.download.redhat.com/2-updates/Agent $(lsb_release -sc) main | tee /etc/apt/sources.list.d/Agent.list'
$ sudo apt-get update

2.2.2. Local Repository
ISO Installations

• Download the Red Hat Ceph Storage ISO
  2. Copy the ISO image to the node.
  3. As root, mount the copied ISO image to the /mnt/rhcs2/ directory:

        $ sudo mkdir -p /mnt/rhcs2
        $ sudo mount -o loop /<path_to_iso>/rhceph-2.0-ubuntu-x86_64.iso /mnt/rhcs2

      NOTE
      For ISO installations using Ansible to install Red Hat Ceph Storage 2, mounting the ISO and creating a local repository is not required.

• Download the Red Hat Storage Console Agent ISO
  2. Copy the ISO image to the node.
  3. As root, mount the copied ISO image to the /mnt/rhscon2_agent/ directory:

        $ sudo mkdir -p /mnt/rhscon2_agent
        $ sudo mount -o loop /<path_to_iso>/rhscon-2.0-rhel-7-x86_64.iso /mnt/rhscon2_agent
Create a Local Repository

1. Copy the ISO image to the node.

2. As `root`, mount the copied ISO image:

   ```
   $ sudo mkdir -p /mnt/<new_directory>
   $ sudo mount -o loop /<path_to_iso_file> /mnt/<new_directory>
   ```

3. As `root`, add the ISO image as a software source:

   ```
   $ sudo apt-get install software-properties-common
   $ sudo add-apt-repository "deb file:/mnt/<new_directory> $(lsb_release -sc) main"
   ```

**NOTE**

With ISO-based installations, the Red Hat Storage Console can host the local repositories, so the Red Hat Ceph Storage nodes can retrieve all the required packages without needing to access the Internet. If the Red Hat Storage Console node can access the Internet, then you can receive online updates and publish them to the rest of the storage cluster.

If you are completely disconnected from the Internet, then you must use ISO images to receive any updates.

**2.3. CONFIGURING RAID CONTROLLERS**

If a RAID controller with 1-2 GB of cache is installed on a host, enabling write-back caches might result in increased small I/O write throughput. To prevent this problem, 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 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 or some firmware do not provide such information, so verify that disk level caches are disabled to avoid file system corruption.

Create a single RAID 0 volume with write-back for each 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 controller, investigate whether your controller and firmware support passthrough mode. **Passthrough** mode helps avoid caching logic, and generally results in much lower latency for fast media.

**2.4. CONFIGURING NETWORK**
All Ceph clusters 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.

**IMPORTANT**

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

For additional information on network configuration see the Network Configuration Reference chapter in the Configuration Guide for Red Hat Ceph Storage 2.

**2.5. SETTING DNS NAME RESOLUTION**

Ceph nodes must be able to resolve short host names, not just fully qualified domain names. Set up a default search domain to resolve short host names. To retrieve a Ceph node short host name, execute:

```
$ hostname -s
```

Each Ceph node must be able to ping every other Ceph node in the cluster by its short host name.

**2.6. CONFIGURING FIREWALL**

Red Hat Ceph Storage 2 uses the `iptables` service, which you must configure to suit your environment.

Monitor nodes use port `6789` for communication within the Ceph cluster. The monitor where the `calamari-lite` is running uses port `8002` for access to the Calamari REST-based API.

On each Ceph OSD node, the OSD daemon uses 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

Ceph object gateway nodes use port `7480` by default. However, you can change the default port, for example to port `80`. To use the SSL/TLS service, open port `443`.

For more information about public and cluster network, see Network.

**Configuring Access**

1. As `root`, on all Ceph Monitor nodes, open port `6789` on the public network:
2. If **calamari-lite** is running on the Ceph Monitor node, as **root**, open port **8002** on the public network:

```
$ sudo iptables -I INPUT 1 -i <iface> -p tcp -s <IP-address>/<prefix> --dport 8002 -j ACCEPT
```

3. If you use Red Hat Storage Console, as **root**, limit the traffic to port **8002** on the Ceph Monitor nodes to accept only traffic from the Red Hat Storage Console administration node:

```
$ sudo iptables -I INPUT 1 -i <iface> -p tcp -s <RHCS-IP-address> --dport 8002 -j ACCEPT
$ sudo iptables -I INPUT 2 -i <iface> -p tcp -s 0.0.0.0/0 --dport 8002 -j DROP
```

Repeat these commands for IPv6 addressing if necessary:

```
$ sudo ip6tables -I INPUT 1 -i <iface> -p tcp -s <RHCS-IP-address> --dport 8002 -j ACCEPT
$ sudo ip6tables -I INPUT 2 -i <iface> -p tcp -s 0.0.0.0/0 --dport 8002 -j DROP
```

4. As **root**, on all OSD nodes, open ports **6800-7300**:

```
$ sudo iptables -I INPUT 1 -i <iface> -m multiport -p tcp -s <IP-address>/<prefix> --dports 6800:7300 -j ACCEPT
```

Where `<ip-address>` is the network address of the OSD nodes.

5. As **root**, on all object gateway nodes, open the relevant port or ports on the public network.
   a. To open the default port **7480**:

```
$ sudo iptables -I INPUT 1 -i <iface> -p tcp -s <IP-address>/<prefix> --dport 7480 -j ACCEPT
```

   b. Optionally, as **root**, if you changed the default Ceph object gateway port, for example to port **80**, open this port:

```
$ sudo iptables -I INPUT 1 -i <iface> -p tcp -s <IP-address>/<prefix> --dport 80 -j ACCEPT
```

   c. Optionally, as **root**, to use SSL/TLS, open port **443**:

```
$ sudo iptables -I INPUT 1 -i <iface> -p tcp -s <IP-address>/<prefix> --dport 443 -j ACCEPT
```

6. As **root**, make the changes persistent on each node:
   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:

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

Return to prerequisite checklist

2.7. CONFIGURING NETWORK TIME PROTOCOL

NOTE

If using Ansible to deploy a Red Hat Ceph Storage cluster, then the installation, configuration, and enabling NTP is done automatically during the deployment.

You must configure the Network Time Protocol (NTP) on all Ceph Monitor and OSD nodes. Ensure that Ceph nodes are NTP peers. NTP helps preempt issues that arise from clock drift.

1. As root, install the ntp package:

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

2. As root, start the NTP service and ensure it is running:

```bash
$ sudo service ntp start
$ sudo service ntp status
```

3. Ensure that NTP is synchronizing Ceph monitor node clocks properly:

```bash
$ ntpq -p
```

Return to prerequisite checklist

2.8. CREATING AN ANSIBLE USER (ANSIBLE DEPLOYMENT ONLY)

Ansible must login to Ceph nodes as a user that has passwordless root privileges, because Ansible needs to install software and configuration files without prompting for passwords.

Red Hat recommends creating an Ansible user on all Ceph nodes in the cluster.
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 to for brute force attacks. For example, root, admin, or <productname> are not advised.

The following procedure, substituting <username> for the user name you define, describes how to create an Ansible user with passwordless root privileges on a Ceph node.

1. Use the ssh command to log in to a Ceph node:

   $ ssh <user_name>@<hostname>

   Replace <hostname> with the host name of the Ceph node.

2. Create a new Ansible user and set a new password for this user:

   $sudo useradd <username>
   $sudo passwd <username>

3. Ensure that the user you added has the root privileges:

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

4. Ensure the correct file permissions:

   $ sudo chmod 0440 /etc/sudoers.d/<username>

Return to prerequisite checklist

2.9. ENABLING PASSWORD-LESS SSH (ANSIBLE DEPLOYMENT ONLY)

Since Ansible will not prompt for a password, you must generate SSH keys on the administration node and distribute the public key to each Ceph node.

1. Generate the SSH keys, but do not use sudo or the root user. Instead, use the Ansible user you created in Create an Ansible User. Leave the passphrase empty:

   $ ssh-keygen

   Generating public/private key pair.
   Enter file in which to save the key (/ceph-admin/.ssh/id_rsa):
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /ceph-admin/.ssh/id_rsa.
   Your public key has been saved in /ceph-admin/.ssh/id_rsa.pub.
2. Copy the key to each Ceph Node, replacing `<username>` with the user name you created in Create an Ansible User and `<hostname>` with a host name of a Ceph node:

   $ ssh-copy-id <username>@<hostname>

3. Modify or create (using a utility such as vi) the `~/.ssh/config` file of the Ansible administration node so that Ansible can log in to Ceph nodes as the user you created without requiring you to specify the `-u <username>` option each time you execute the `ansible-playbook` command. Replace `<username>` with the name of the user you created and `<hostname>` with a host name of a Ceph node:

   Host node1
   Hostname <hostname>
   User <username>

   Host node2
   Hostname <hostname>
   User <username>

   Host node3
   Hostname <hostname>
   User <username>

After editing the `~/.ssh/config` file on the Ansible administration node, ensure the permissions are correct:

   $ chmod 600 ~/.ssh/config

Return to prerequisite checklist
CHAPTER 3. STORAGE CLUSTER INSTALLATION

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

There are three ways to install a Red Hat Ceph Storage cluster:

- Red Hat Storage Console
- Ansible automation application
- Command line interface

3.1. INSTALLING RED HAT CEPH STORAGE USING THE RED HAT STORAGE CONSOLE

The Red Hat Storage Console is a web-based interface utility, and a unified storage management platform for Red Hat Storage products, such as Red Hat Ceph Storage. The Red Hat Storage Console provides a flexible, pluggable framework to deploy, manage, and monitor software-defined storage technologies.

To install the Red Hat Storage Console, see the Red Hat Storage Console Quick Start Guide.

3.1.1. Installing and Configuring the Red Hat Storage Console Agent

To use the management capabilities of the Red Hat Storage Console, each node participating in the Ceph storage cluster must be prepared by installing and configuring the Red Hat Storage Console agent. Once this is done, the Red Hat Storage Console can create and manage a Ceph storage cluster.

Before the Red Hat Storage Console agent can be installed, an operational Red Hat Storage Console server must be running. See the Red Hat Storage Console Quick Start Guide for details on installing and configuring the Red Hat Storage Console.

IMPORTANT

Trying to use local repositories to install the Red Hat Storage Console agent will fail. At this time, using online repositories is required to install the Red Hat Storage Console agent.

Do the following on each Ceph Monitor and OSD nodes in the Ceph storage cluster:
Preparing

1. For importing existing Ceph storage cluster nodes, skip to step 3.

2. For new Ceph Monitor and OSD nodes, go through the prerequisite checks in Figure 2.1, “Prerequisite Workflow” before installing the Red Hat Storage Console agent. The prerequisite Section 2.2, “Enabling Ceph Repositories” can be skipped. Enabling the correct repositories is in the procedures below. Once done with the prerequisite checks, skip to step 5.

3. Install the latest updates for Ubuntu 16.04 Xenial:

   $ sudo apt-get update
   $ sudo apt-get upgrade

4. Verify that the Network Time Protocol (NTP) is enabled and the local time is synchronized on each node in the storage cluster:

   $ sudo ntpq -p
   $ sudo date

   For more details about NTP, see Note.

5. Enable the Red Hat Storage Console Agent repository on the Ceph Monitor and OSD nodes:

   $ sudo bash -c 'umask 0077; echo deb
   https://customername:customerpasswd@rhcs.download.redhat.com/2-updates/Agent $(lsb_release -sc) main | tee
   /etc/apt/sources.list.d/Agent.list'
   $ sudo bash -c 'wget -O -
   https://www.redhat.com/security/fd431d51.txt | apt-key add -'
   $ sudo apt-get update

   a. For Monitor nodes, enable the Ceph Monitor repository:

   $ sudo bash -c 'umask 0077; echo deb
   https://customername:customerpasswd@rhcs.download.redhat.com/2-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

   b. For the OSD nodes, enable the Ceph OSD repository:

   $ sudo bash -c 'umask 0077; echo deb
   https://customername:customerpasswd@rhcs.download.redhat.com/2-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

Installing and Configuring
1. On Ceph Monitor and OSD nodes, as `root`, install and configure the Red Hat Storage Console Agent:

   ```bash
   $ curl <FQDN_RHS_Console_node>:8181/setup/agent/ | sudo bash
   ```

   **Example**

   ```bash
   $ curl rhsc.example.com:8181/setup/agent/ | sudo bash
   % Total    % Received % Xferd  Average Speed   Time    Time
   Time  Current
   Left  Speed
   100  1647  100 1647  0    0   98k   0    0    0    0    87k
   --> creating new user with disabled password: ceph-installer
   Removing password for user ceph-installer.
   passwd: Success
   --> adding provisioning key to the ceph-installer user
   authorized_keys
   --> ensuring correct permissions on .ssh/authorized_keys
   --> ensuring ceph-installer user will be able to sudo
   --> ensuring ceph-installer user does not require a tty
   --> installing and configuring agent
   ```

   ```JSON
   {"endpoint": "/api/agent/", "succeeded": false, "stdout": null,
   "started": null, "request": "", "exit_code": null, "ended": null,
   "http_method": "", "command": null, "user_agent": "", "stderr":
   null, "identifier": "eaf260b4-4474-4e0a-863d-58331b56cbb5"}
   ```

   **NOTE**

   The installing and configuring process of the Red Storage Console Agent will take several minutes to complete, even after the command returns you back to the command prompt. During the configuring process password-less SSH will be configured on the node. To view the status of this process:

   ```bash
   curl <FQDN_RHS_Console_node>:8181/api/tasks/
   ```

2. Open a web browser from a workstation, and go to the URL for the Red Hat Storage Console web interface. Log in as the administrator using the "admin" user name, and "admin" as the password.
3. In the top-right corner of the web interface, click on the small computer icon. This opens a page with a list of discovered systems. Click on the "Accept" button to add the new storage host.

4. After a few seconds, a green check mark appears next to storage host name. The host is fully recognized by the Storage Console and available for use in a storage cluster:
If a red “X” appears next to the storage host name, check the salt-minion service and the salt-minion configuration. You can also view the /var/log/salt/minion and /var/log/skynet/skynet.log logs for more details.

Once you have all your Ceph storage nodes prepared, proceed to create a new Ceph storage cluster or import an existing Ceph storage cluster.

3.1.2. Importing an existing Ceph Storage Cluster

To import an existing Red Hat Ceph Storage 2 cluster into the Red Hat Storage Console, see the Red Hat Storage Console Quick Start Guide for details.

3.2. INSTALLING RED HAT CEPH STORAGE USING ANSIBLE

Currently, Red Hat does not provide the ceph-ansible package for Ubuntu. If you want to deploy Red Hat Ceph Storage 2 in an Ubuntu environment using Ansible, then a Red Hat Enterprise Linux node must be used. To install and configure Ansible, see the Red Hat Ceph Storage 2 Installation Guide for Red Hat Enterprise Linux for more details.

To add more Monitors or OSDs to an existing storage cluster, see the Red Hat Ceph Storage Administration Guide for details:

- Adding a Monitor
- Adding an OSD

3.3. INSTALLING RED HAT CEPH STORAGE BY USING THE COMMAND-LINE INTERFACE

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

**IMPORTANT**

Red Hat does not support or test upgrading manually deployed clusters. Currently, the only supported way to upgrade to a minor version of Red Hat Ceph Storage 2 is to use the Ansible automation application as described in Section 5.2, "Upgrading Between Minor Versions and Applying Asynchronous Updates". Therefore, Red Hat recommends to use Ansible or Red Hat Storage Console to deploy a new cluster with Red Hat Ceph Storage 2. See Section 3.2, “Installing Red Hat Ceph Storage using Ansible” and Section 3.1, “Installing Red Hat Ceph Storage using the Red Hat Storage Console” 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.

### 3.3.1. 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 2 Monitor repository. For ISO-based installations, see the ISO installation section.

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:
Syntx

# 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 must set the ms bind ipv6 option to true. See the Red Hat Ceph Storage Configuration Guide for more details.

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

**Syntax**

```bash
# ceph-authtool --create-keyring /etc/ceph/<cluster_name>.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
```

**Example**

```bash
# 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
```

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

**Syntax**

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

**Example**

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

importing contents of /etc/ceph/ceph.client.admin.keyring into /tmp/ceph.mon.keyring

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

**Syntax**

```bash
# ceph-authtool --create-keyring /etc/ceph/<cluster_name>.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
```

**Example**

```bash
# ceph-authtool --create-keyring /etc/ceph/<cluster_name>.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
```

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**

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

**Example**

```bash
$ 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 Red Hat Ceph Storage Configuration Guide. 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>
filestore xattr use omap = true
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 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. Verify that Ceph created the default pools:

$ ceph osd lspools
0 rbd,

20. Verify that the Monitor is running. The status output will look similar to the following example. The Monitor is up and running, but the cluster health will be in a HEALTH_ERR state. This error is indicating that placement groups are stuck and inactive. Once OSDs are added to the cluster and active, the placement group health errors will disappear.

Example

$ ceph -s
cluster a7f64266-0894-4f1e-a635-d0aeaca0e993
health HEALTH_ERR 192 pgs stuck inactive; 192 pgs stuck unclean; no osds
monmap e1: 1mons at {node1=192.168.0.120:6789/0}, election epoch 1, quorum 0 node1
osdmap e1: 0 osds: 0 up, 0 in
pgmap v2: 192 pgs, 3 pools, 0 bytes data, 0 objects
0 kB used, 0 kB / 0 kB avail
192 creating

To add more Red Hat Ceph Storage Monitors to the storage cluster, see the Red Hat Ceph Storage Administration Guide

3.3.2. 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 Red Hat Ceph Storage Configuration Guide.

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 2 OSD repository. For ISO-based installations, see the ISO installation section.

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

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

   Example

   ```bash
   # 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

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

   Example

   ```bash
   # 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:
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:

```bash
# 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**

```bash
# 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:

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

**Example**

```bash
# ceph-osd -i 0 --mkfs --mkkey --osd-uuid b367c360-b364-4b1d-8fc6-09408a9dca7a
... 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 Red Hat Ceph Storage Storage Strategies Guide 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

<table>
<thead>
<tr>
<th>ID</th>
<th>WEIGHT</th>
<th>TYPE</th>
<th>NAME</th>
<th>UP/DOWN</th>
<th>REWEIGHT</th>
<th>PRIMARY-AFFINITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>2</td>
<td>root</td>
<td>default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>2</td>
<td>host</td>
<td>node2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>osd.0</td>
<td></td>
<td>up</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-3</td>
<td>1</td>
<td>host</td>
<td>node3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>osd.1</td>
<td></td>
<td>up</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

To expand the storage capacity by adding new OSDs to the storage cluster, see the Red Hat Ceph Storage Administration Guide for more details.

3.3.3. Calamari Server Installation

The Calamari server provides a RESTful API for monitoring Ceph storage clusters.

NOTE

The Red Hat Storage Console replaces the Calamari graphical user interface application.

To install calamari-server, perform the following steps on all Monitor nodes.

1. As root, enable the Red Hat Ceph Storage Monitor repository

2. As root, install calamari-server:

   $ sudo apt-get install calamari-server

3. As root, initialize the calamari-server:

   Syntax

   $ sudo calamari-ctl clear --yes-i-am-sure
   $ sudo calamari-ctl initialize --admin-username <uid> --admin-password <pwd> --admin-email <email>

   Example

   $ sudo calamari-ctl clear --yes-i-am-sure
   $ sudo calamari-ctl initialize --admin-username admin --admin-password admin --admin-email cephadm@example.com
IMPORTANT

The `calamari-ctl clear --yes-i-am-sure` command is only necessary for removing the database of old Calamari server installations. Running this command on a new Calamari server results in an error.

NOTE

During initialization, the `calamari-server` will generate a self-signed certificate and a private key and place them in the `/etc/calamari/ssl/certs/` and `/etc/calamari/ssl/private` directories respectively. Use HTTPS when making requests. Otherwise, user names and passwords are transmitted in clear text.

The `calamari-ctl initialize` process generates a private key and a self-signed certificate, which means there is no need to purchase a certificate from a Certificate Authority (CA).

To verify access to the HTTPS API through a web browser, go to the following URL. Click through the untrusted certificate warnings, because the auto-generated certificate is self-signed:

```
https://<calamari_hostname>:8002/api/v2/cluster
```

To use a key and certificate from a CA:

1. Purchase a certificate from a CA. During the process, you will generate a private key and a certificate for CA. Or you can also use the self-signed certificate generated by Calamari.

2. Save the private key associated to the certificate to a path, preferably under `/etc/calamari/ssl/private/`.

3. Save the certificate to a path, preferably under `/etc/calamari/ssl/certs/`.

4. Open the `/etc/calamari/calamari.conf` file.

5. Under the `[calamari_web]` section, modify `ssl_cert` and `ssl_key` to point to the respective certificate and key path, for example:

```
[calamari_web]
...
ssl_cert = /etc/calamari/ssl/certs/calamari-lite-bundled.crt
ssl_key = /etc/calamari/ssl/private/calamari-lite.key
```

6. As `root`, re-initialize Calamari:

```
$ sudo calamari-ctl initialize
```
CHAPTER 4. CLIENT INSTALLATION

Red Hat Ceph Storage supports the following types of Ceph clients:

Ceph CLI
The Ceph command-line interface (CLI) enables administrators to execute Ceph administrative commands. See Section 4.2, “Ceph Command-line Interface Installation” for information on installing the Ceph CLI.

Block Device
The Ceph Block Device is a thin-provisioned, resizable block device. See Section 4.3, “Ceph Block Device Installation” for information on installing Ceph Block Devices.

Object Gateway
The Ceph Object Gateway provides its own user management and Swift- and S3-compliant APIs. See Section 4.4, “Ceph Object Gateway Installation” for information on installing Ceph Object Gateways.

In addition, the ceph-ansible utility provides the ceph-client role that copies the Ceph configuration file and the administration keyring to nodes. See Section 4.1, “Installing the ceph-client role” for details.

IMPORTANT
To use Ceph clients, you must have a Ceph cluster storage running, preferably in the active + clean state.

In addition, before installing the Ceph clients, ensure to perform the tasks listed in the Figure 2.1, “Prerequisite Workflow” section.

4.1. INSTALLING THE CEPH-CLIENT ROLE

The ceph-client role copies the Ceph configuration file and administration keyring to a node. In addition, you can use this role to create custom pools and clients.

To deploy the ceph-client role by using Ansible, see the Red Hat Ceph Storage 2 Installation Guide for Red Hat Enterprise Linux.

4.2. CEPH COMMAND-LINE INTERFACE INSTALLATION

The Ceph command-line interface (CLI) is provided by the ceph-common package and includes the following utilities:

- ceph
- ceph-auth tool
- ceph-dencoder
- rados

To install the Ceph CLI:

1. On the client node, enable the Tools repository.
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.

### 4.3. CEPH BLOCK DEVICE INSTALLATION

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.

**Before you start**

- Ensure to perform the tasks listed in the Section 4.2, “Ceph Command-line Interface Installation” 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.

**Installing Ceph Block Devices by Using the Command Line**

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 'allow r' osd 'allow rwx
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 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:
  
  ```bash
  rbd feature disable <image_name> <feature_name>
  
  For example:
  ```
  ```bash
  # 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 by default in the Ceph configuration file:

  ```bash
  rbd_default_features = 1
  ```

  This is a known issue, for details see the Release Notes Red Hat Ceph Storage 2.2.

  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:

  ```bash
  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 Red Hat Ceph Storage Block Device Guide.

### 4.4. CEPH OBJECT GATEWAY INSTALLATION

The Ceph object gateway, also known as the RADOS gateway, is an object storage interface built on top of the [librados](https://librados.github.io) API to provide applications with a RESTful gateway to Ceph storage clusters.

For more information about the Ceph object gateway, see the Object Gateway Guide for Ubuntu.

There are two ways to install the Ceph object gateway:

- Using the Ansible automation application, see Section 4.4.1, "Installing Ceph Object Gateway by using Ansible" for details
- Using the command-line interface, see Section 4.3.2, "Installing Ceph Object Gateway Manually" for details

#### 4.4.1. Installing Ceph Object Gateway by using Ansible

To deploy the Ceph Object Gateway using Ansible, see the Red Hat Ceph Storage 2 Installation Guide for Red Hat Enterprise Linux.

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.

#### 4.4.2. Installing Ceph Object Gateway Manually
1. Enable the Red Hat Ceph Storage 2 Tools repository. For ISO-based installations, see the ISO installation section.

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**

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

**Syntax**

```bash
$ 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**

```bash
$ 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**

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

**Example**

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

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

```bash
$ 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**

```bash
$ 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.
CHAPTER 5. UPGRADING CEPH STORAGE CLUSTER

There are two main upgrading paths:

- from Red Hat Ceph Storage 1.3 to 2 (Section 5.1, “Upgrading from Red Hat Ceph Storage 1.3 to 2”)
- between minor versions of Red Hat Ceph Storage 2 or between asynchronous updates (Section 5.2, “Upgrading Between Minor Versions and Applying Asynchronous Updates”)

5.1. UPGRADING FROM RED HAT CEPH STORAGE 1.3 TO 2

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 1.3.2 to 2.0:

- 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. See the Red Hat Ceph Storage Strategies Guide for more information.

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.

5.1.1. Upgrading a Ceph Monitor Node
Red Hat recommends a minimum of three Monitors for a production storage cluster. There must be an odd number of Monitors. While you are upgrading one Monitor, the storage cluster will still have quorum.

Upgrading Red Hat Ceph Storage from version 1.3.2 to version 2 running on Ubuntu 14.04 Trusty to Ubuntu 16.04 Xenial has two main tasks. The Red Hat Ceph Storage packages will be upgraded first, then the Ubuntu operating system will be upgraded next. These two main tasks will need to be done on each Monitor node in the storage cluster. Perform the following steps on each Monitor node in the storage cluster, sequentially upgrading one Monitor node at a time.

**IMPORTANT**

Red Hat does not support running Red Hat Ceph Storage 2 clusters on Ubuntu 14.04 Trusty in a production environment. This is only a transitional step to get to Red Hat Ceph Storage 2 on Ubuntu 16.04 Xenial, which is the supported platform. Red Hat recommends having a full system backup before proceeding with these upgrade procedures.

1. As `root`, disable any Red Hat Ceph Storage 1.3.x repositories:

   If the following lines exist in the `/etc/apt/sources.list` or in the `/etc/apt/sources.list.d/ceph.list` files, then comment out the online repositories for Red Hat Ceph Storage 1.3 by adding a `#` to the beginning of the line.

   ```
   deb https://<customer_name>:<customer_password>@rhcs.download.redhat.com/ubuntu/1.3-updates/Installer
   deb https://<customer_name>:<customer_password>@rhcs.download.redhat.com/ubuntu/1.3-updates/Calamari
   deb https://<customer_name>:<customer_password>@rhcs.download.redhat.com/ubuntu/1.3-updates/Tools
   ```

   Also, check for the following files in `/etc/apt/sources.list.d/`:

   ```
   Calamari.list
   ceph-mon.list
   ceph-osd.list
   Installer.list
   Tools.list
   ```

   **NOTE**

   Remove any reference to Red Hat Ceph Storage 1.3.x in the APT source file(s). If an ISO-based installation was performed for Red Hat Ceph Storage 1.3.x, then skip this first step.

2. Enable the Red Hat Ceph Storage 2 Monitor repository. For ISO-based installations, see the ISO installation section.

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

   **Syntax**

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

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

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

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

a. Verify the latest Red Hat version is installed:

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

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

**Syntax**

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

**Example**

```bash
# 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:

```bash
# 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:

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

   ```bash
   # udevadm trigger
   ```


   a. Configure **update-manager** for the Red Hat Ceph Storage packages:

      i. Create a new file

         ```bash
         $ sudo touch /etc/update-manager/release-upgrades.d/rhcs.cfg
         ```

      ii. Add the following lines to the new file

         ```bash
         [Sources]
         AllowThirdParty=yes
         ```

   b. Start the Ubuntu upgrade:

      ```bash
      $ sudo do-release-upgrade -d
      ```

   c. Follow the on screen instructions

   d. Verify the Ceph package versions:

      ```bash
      $ dpkg -s ceph-base | grep Version
      Version: 10.2.2-19redhat1xenial
      ```

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

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

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

      ```bash
      # shutdown -r now
      ```

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

      ```bash
      # ceph -s
      ```

To add more Red Hat Ceph Storage Monitors to the storage cluster, see the Red Hat Ceph Storage Administration Guide.
5.1.2. Upgrading a Ceph OSD Node

Upgrading Red Hat Ceph Storage from version 1.3.2 to version 2 running on Ubuntu 14.04 Trusty to Ubuntu 16.04 Xenial has two main tasks. The Red Hat Ceph Storage packages will be upgraded first, then the Ubuntu operating system will be upgraded next. These two main tasks will need to be done on each OSD node in the storage cluster. Perform the following steps on each OSD node in the storage cluster, sequentially upgrading one OSD node at a time.

**IMPORTANT**

Red Hat does not support running Red Hat Ceph Storage 2 clusters on Ubuntu 14.04 Trusty in a production environment. This is only a transitional step to get to Red Hat Ceph Storage 2 on Ubuntu 16.04 Xenial, which is the supported platform. Red Hat recommends having a full system backup before proceeding with these upgrade procedures.

During the upgrade of an OSD node, some placement groups will become degraded, because the OSD might be down or restarting. You will need to tell the storage cluster not to mark an OSD out, because you do not want to trigger a recovery. The default behavior is to mark an OSD out of the CRUSH map after five minutes.

On a Monitor node, set `noout` and `norebalance` flags for the OSDs:

```bash
# ceph osd set noout
# ceph osd set norebalance
```

Perform the following steps on each OSD node in the storage cluster. Sequentially upgrading one OSD node at a time. If an ISO-based installation was performed for Red Hat Ceph Storage 1.3, then skip this first step.

1. **As root**, disable the Red Hat Ceph Storage 1.3 repositories:
   
   If the following lines exist in the `/etc/apt/sources.list` or in the `/etc/apt/sources.list.d/ceph.list` files, then comment out the online repositories for Red Hat Ceph Storage 1.3 by adding a `#` to the beginning of the line.

   ```bash
deb https://<customer_name>:@rhcs.download.redhat.com/ubuntu/1.3-updates/Installer
   deb https://<customer_name>:@rhcs.download.redhat.com/ubuntu/1.3-updates/Calamari
deb https://<customer_name>:@rhcs.download.redhat.com/ubuntu/1.3-updates/Tools
   
   Also, check for the following files in `/etc/apt/sources.list.d/`:
   ```

   ```bash
   Calamari.list
   ceph-mon.list
   ceph-osd.list
   Installer.list
   Tools.list
   ```
NOTE
Remove any reference to Red Hat Ceph Storage 1.3.x in the APT source file(s). If an ISO-based installation was performed for Red Hat Ceph Storage 1.3.x, then skip this first step.

2. Enable the Red Hat Ceph Storage 2 OSD repository. For ISO-based installations, see the ISO installation section.

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
Running the following find command might speed up the process of changing ownership by running 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

   a. Configure `update-manager` for the Red Hat Ceph Storage packages:
      i. Create a new file
         
         $ sudo touch /etc/update-manager/release-upgrades.d/rhcs.cfg
      ii. Add the following lines to the new file
         
         [Sources]
         AllowThirdParty=yes
   b. Start the Ubuntu upgrade:
      
      $ sudo do-release-upgrade -d
   c. Follow the on screen instructions
   d. Verify the Ceph package versions:
      
      $ dpkg -s ceph-base | grep Version
      Version: 10.2.2-19redhat1xenial
   e. As `root`, enable the `ceph-osd` process:
      
      $ sudo systemctl enable ceph-osd.target
      $ sudo systemctl enable ceph-osd@<osd_id>
   f. As `root`, reboot the OSD node:
      
      # shutdown -r now

9. Move to the next OSD node.
NOTE

While the `noout` and `norebalance` flags are set, the storage cluster will have the `HEALTH_WARN` status:

```bash
$ 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:

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

In addition, set the `require_jewel_osds` flag. This flag ensures that no more OSDs with Red Hat Ceph Storage 1.3 can be added to the storage cluster. If you do not set this flag, the storage status will be `HEALTH_WARN`.

```bash
# ceph osd set require_jewel_osds
```

To expand the storage capacity by adding new OSDs to the storage cluster, see the Red Hat Ceph Storage Administration Guide for more details.

### 5.1.3. Upgrading the Ceph Object Gateway Nodes

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

**IMPORTANT**

Red Hat does not support running Red Hat Ceph Storage 2 clusters on Ubuntu 14.04 in a production environment. Therefore, upgrading Red Hat Ceph Storage from 1.3 to 2 includes:

- Upgrading the Red Hat Ceph Storage packages from 1.3 to 2
- Upgrading the Ubuntu operation system from 14.04 to 16.04

Perform these steps on each Ceph Object Gateway node in use, sequentially upgrading one node at a time.

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

**Before You Start**

- 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: Upgrading the Ceph Object Gateway Node**

1. If you used online repositories to install Red Hat Ceph Storage, disable the 1.3 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/1.3-updates/Installer
      # deb https://<customer_name>:<customer_password>@rhcs.download.redhat.com/ubuntu/1.3-updates/Calamari
      # deb https://<customer_name>:<customer_password>@rhcs.download.redhat.com/ubuntu/1.3-updates/Tools
      ```
   
   b. Remove the following files from the `/etc/apt/sources.list.d/` directory.
      
      ```
      # rm /etc/apt/sources.list.d/Calamari.list
      # rm /etc/apt/sources.list.d/ceph-mon.list
      # rm /etc/apt/sources.list.d/ceph-osd.list
      # rm /etc/apt/sources.list.d/Installer.list
      # rm /etc/apt/sources.list.d/Tools.list
      ```

2. Enable the Red Hat Ceph Storage 2 Tools repository. For ISO-based installations, see the ISO Installation section.

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`. 
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. Upgrade from Ubuntu 14.04 to 16.04.

   a. Configure the `update-manager` utility for the Red Hat Ceph Storage packages:

      i. Create a new `rhcs.cfg` file in the `/etc/update-manager/release-upgrades.d/` directory.

         $ sudo touch /etc/update-manager/release-upgrades.d/rhcs.cfg

      ii. Add the following lines to the file.

         [Sources]
         AllowThirdParty=yes

   b. Start the upgrading process and follow the instructions on the screen.

      $ sudo do-release-upgrade -d

   c. Verify the Ceph package versions:

      $ dpkg -s ceph-base | grep Version
      Version: 10.2.2-19redhat1xenial

8. 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

9. Reboot the Ceph Object Gateway node:

    # shutdown -r now
10. If you use a load balancer, add the Ceph Object Gateway node back to the load balancer.

11. Repeat these steps on a next Ceph Object Gateway node.

See Also

- The Ceph Object Gateway Guide for Ubuntu

5.1.4. Upgrading a Ceph Client Node

Ceph clients can be the RADOS Gateway, RADOS block devices, the Ceph command-line interface (CLI), Nova compute nodes, qemu-kvm, or any custom application using the Ceph client-side libraries. Red Hat recommends all Ceph clients to be running the same version as the Ceph storage cluster.

IMPORTANT

Red Hat recommends stopping all IO running against a Ceph client node while the packages are being upgraded. Not stopping all IO might cause unexpected errors to occur.

1. As root, disable any Red Hat Ceph Storage 1.3 repositories:
   If the following lines exist in the /etc/apt/sources.list or in the /etc/apt/sources.list.d/ceph.list files, then comment out the online repositories for Red Hat Ceph Storage 1.3 by adding a # to the beginning of the line.

   ```bash
   deb https://<customer_name>:<customer_password>@rhcs.download.redhat.com/ubuntu/1.3-updates/Installer
   deb https://<customer_name>:<customer_password>@rhcs.download.redhat.com/ubuntu/1.3-updates/Calamari
   deb https://<customer_name>:<customer_password>@rhcs.download.redhat.com/ubuntu/1.3-updates/Tools
   ```

   Also, check for the following files in /etc/apt/sources.list.d/:

   ```bash
   Calamari.list
   ceph-mon.list
   ceph-osd.list
   Installer.list
   Tools.list
   ```

   **NOTE**

   Remove any reference to Red Hat Ceph Storage 1.3.x in the APT source file(s).

2. On the client node, enable the Tools repository.

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

   ```bash
   $ sudo apt-get install ceph-common
   ```
Any application depending on the Ceph client-side libraries will have to be restarted after upgrading the Ceph client package.

NOTE
For Nova compute nodes with running \texttt{qemu-kvm} instances or if using a dedicated \texttt{qemu-kvm} client, then stopping and starting the \texttt{qemu-kvm} instance processes is required. A simple restart will not work here.

5.1.5. Repurposing the Ceph Administration Node

Red Hat expects a dedicated Ceph Administration node was used with the previous versions of Red Hat Ceph Storage, which might have hosted a Calamari server, the storage cluster’s configuration files and keys, and optionally, local repositories for installing Ceph on nodes that cannot access the Internet for security reasons. The legacy Ceph Administration node can be repurposed as the new Red Hat Storage Console and Ansible administration servers.

With previous versions of the Calamari server, it can be hosted on any node outside the Ceph storage cluster, but starting with Red Hat Ceph Storage 2.0, the Calamari server must be hosted on a Ceph Monitor node.

If the legacy Ceph Administration node hosted an old version of the Calamari server, then perform these steps before repurposing:

1. As \texttt{root}, remove the old Calamari packages:

\begin{verbatim}
$ sudo apt-get purge calamari-server calamari-client
$ sudo apt-get purge salt-master salt-minion salt
$ sudo apt-get purge diamond
$ sudo apt-get purge graphite
\end{verbatim}

2. As \texttt{root}, delete the saved Salt keys:

\begin{verbatim}
# salt-key -D
\end{verbatim}

3. As \texttt{root}, remove the Calamari Salt files from all nodes in the Ceph storage cluster:

\begin{verbatim}
# rm /etc/salt/minion.d/calamari.conf
# rm /etc/salt/pki/minion/*
\end{verbatim}

4. After upgrading to Red Hat Ceph Storage 2, repurpose the legacy Ceph Administration node by seeing the Red Hat Storage Console \textit{Quick Start Guide} to install the Red Hat Storage Console or see \textit{Section 3.2, “Installing Red Hat Ceph Storage using Ansible”} for installing Ansible.

5. If using the Red Hat Storage Console, then install the Red Hat Storage Console Agent on all nodes in the storage cluster. See \textit{Section 3.1.1, “Installing and Configuring the Red Hat Storage Console Agent”} for details.

To install and configure the new Calamari server, see \textit{Section 3.3.3, “Calamari Server Installation”} for details.

Once the Red Hat Storage Console Agent is installed and configured on each node in the storage cluster and the Calamari server is installed and configured, you can import the Ceph storage cluster into the Red Hat Storage Console. See the Red Hat Storage Console \textit{Quick Start Guide} for more details.
5.2. UPGRADING BETWEEN MINOR VERSIONS AND APPLYING ASYNCHRONOUS UPDATES

Use the Ansible `rolling_update.yml` playbook located in the `infrastructure-playbooks` directory from the administration node to upgrade between two minor versions of Red Hat Ceph Storage 2 or to apply asynchronous updates.

Currently, this is the only supported way to upgrade to a minor version. If you use a cluster that was not deployed by using Ansible, see Taking Over an Existing Cluster for details on configuring Ansible to use a cluster that was deployed without it.

**NOTE**

The administration node must use Red Hat Enterprise Linux because the `ceph-ansible` package is not supported on Ubuntu. See the Installing Red Hat Ceph Storage using Ansible chapter in the Red Hat Ceph Storage 2 Installation Guide for Red Hat Enterprise Linux.

Ansible upgrades the Ceph nodes in the following order:

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

**NOTE**

Upgrading encrypted OSD nodes is the same as upgrading OSD nodes that are not encrypted.

**Before you Start**

- On the Ansible Administration node, ensure the latest version of `ceph-ansible` is installed:
  
  ```bash
  $ sudo apt-get update ceph-ansible
  ```

- In the `rolling_update.yml` playbook, change the `health_osd_check_retries` and `health_osd_check_delay` values to `40` and `30` respectively. For each OSD node, Ansible will wait up to 20 minutes. Also, Ansible will check the cluster health every 30 seconds, waiting before continuing the upgrade process. Set the following values:

  ```yaml
  health_osd_check_retries: 40
  health_osd_check_delay: 30
  ```

- 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 2.2, “Enabling Ceph Repositories” for details.
If you upgrade from Red Hat Ceph Storage 2.1 to 2.2, review the Section 5.2.1, “Changes Between Ansible 2.1 and 2.2” section first. Ansible 2.2 uses slightly different file names and setting.

Procedure: Updating the Ceph Storage Cluster by using Ansible

1. On the Ansible administration node, navigate to the /usr/share/ceph-ansible/ directory:
   ```bash
   # cd /usr/share/ceph-ansible
   ```

2. In the group_vars/all.yml file, uncomment the upgrade_ceph_packages option and set it to True:
   ```yaml
   upgrade_ceph_packages: True
   ```

3. Run the rolling_update.yml playbook:
   ```bash
   # cp infrastructure-playbooks/rolling_update.yml .
   # ansible-playbook rolling_update.yml
   ```

4. After you upgrade all nodes in the cluster, set the recovery_deletes flag. Use the following command from a Monitor node.
   ```bash
   # ceph osd set recovery_deletes
   ```

   **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 hosts will be upgraded one by one.

5.2.1. Changes Between Ansible 2.1 and 2.2

Red Hat Ceph Storage 2.2 includes Ansible 2.2 that introduces the following changes:

- Files in the group_vars directory have the .yml extension. Before updating to 2.2, you must rename them. To do so:
  Navigate to the Ansible directory:
  ```bash
  # cd /usr/share/ceph-ansible
  ```

  Change the names of the files in group_vars:
  ```bash
  # mv groups_vars/all groups_vars/all.yml
  # mv groups_vars/mons groups_vars/mons.yml
  # mv groups_vars/osds groups_vars/osds.yml
  # mv groups_vars/mdss groups_vars/mdss.yml
  # mv groups_vars/rgws groups_vars/rgws.yml
  ```

- Ansible 2.2 uses different variable names and handles this change automatically when updating to version 2.2. See Table 5.1, “Differences in Variable Names Between Ansible 2.1 and 2.2” table for details.
<table>
<thead>
<tr>
<th>Ansible 2.1 variable name</th>
<th>Ansible 2.2 variable name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ceph_stable_rh_storage</td>
<td>ceph_rhcs</td>
</tr>
<tr>
<td>ceph_stable_rh_storage_version</td>
<td>ceph_rhcs_version</td>
</tr>
<tr>
<td>ceph_stable_rh_storage_cdn_install</td>
<td>ceph_rhcs_cdn_install</td>
</tr>
<tr>
<td>ceph_stable_rh_storage_iso_install</td>
<td>ceph_rhcs_iso_install</td>
</tr>
<tr>
<td>ceph_stable_rh_storage_iso_path</td>
<td>ceph_rhcs_iso_path</td>
</tr>
<tr>
<td>ceph_stable_rh_storage_mount_path</td>
<td>ceph_rhcs_mount_path</td>
</tr>
<tr>
<td>ceph_stable_rh_storage_repository_path</td>
<td>ceph_rhcs_repository_path</td>
</tr>
</tbody>
</table>
CHAPTER 6. 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 Red Hat Ceph Storage Administration Guide.
- Creating and managing snapshots, see the Red Hat Ceph Storage Block Device Guide.
- Expanding the Red Hat Ceph Storage cluster, see the Red Hat Ceph Storage Administration Guide.
- Mirroring RADOS Block Devices, see the Red Hat Ceph Storage Block Device Guide.
- Process management, enabling debug logging, and related topics, see the Red Hat Ceph Storage Administration Guide.
- Tunable parameters, see the Red Hat Ceph Storage Configuration Guide.
- Using Ceph as the back end storage for OpenStack, see the Red Hat OpenStack Platform Storage Guide.