Red Hat Ceph Storage 2 Installation Guide for Red Hat Enterprise Linux

Installing Red Hat Ceph Storage on Red Hat Enterprise Linux

Red Hat Ceph Storage Documentation Team

Red Hat Ceph Storage 2 Installation Guide for Red Hat Enterprise Linux

Installing Red Hat Ceph Storage on Red Hat Enterprise Linux

Legal Notice

Abstract

This document provides instructions on installing Red Hat Ceph Storage on Red Hat Enterprise Linux 7 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 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).

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

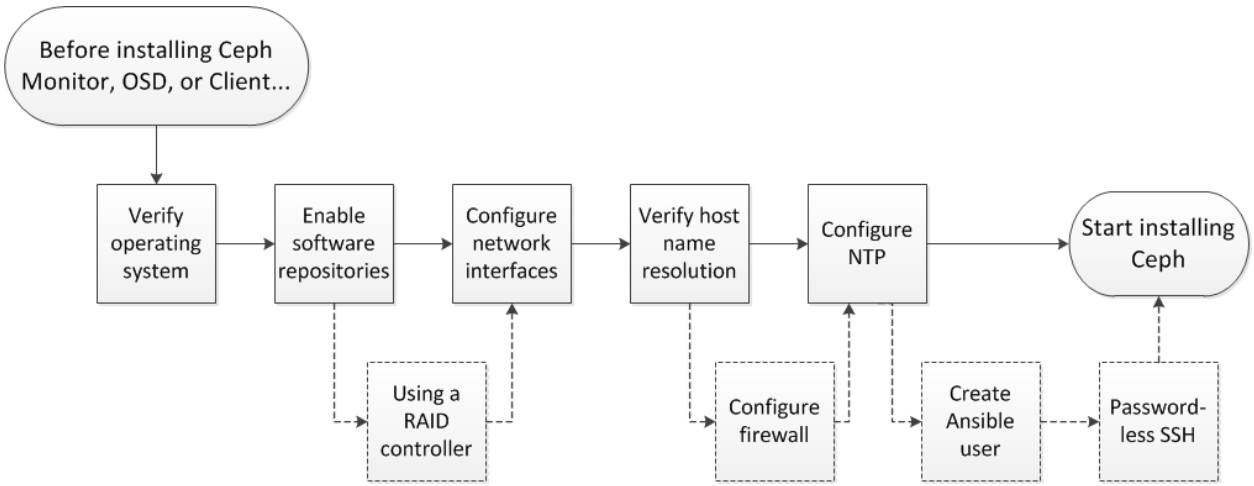
Provides object storage functionality with an interface that is compatible with a large subset of the OpenStack Swift API.

For details on the Ceph architecture, see the Architecture Guide.

For minimum recommended hardware, see the Hardware Guide.

CHAPTER 2. PREREQUISITES

Figure 2.1. Prerequisite Workflow



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

Table 2.1. Prerequisites Checks

Task	Required	Section	Recommendation
Verifying the operating system version	Yes	Section 2.1, “Operating System”	
Registering Ceph nodes	Yes	Section 2.2, “Registering to CDN”	
Enabling Ceph software repositories	Yes	Section 2.3, “Enabling Ceph Repositories”	Two installation methods: - Content Delivery Network (CDN) - Local Repository (ISO)
Using a RAID controller	No	Section 2.4, “Configuring RAID Controllers”	For OSD nodes only.
Configuring network Interface	Yes	Section 2.5, “Configuring Network”	Using a public network is required. Having a private network for cluster communication is optional, but recommended.

Task	Required	Section	Recommendation
Resolving short host names	Yes	Section 2.6, “Setting DNS Name Resolution”	
Configuring a firewall	No	Section 2.7, “Configuring Firewall”	
Configuring the Network Time Protocol	Yes	Section 2.8, “Configuring Network Time Protocol”	
Creating an Ansible user	No	Section 2.9, “Creating an Ansible User (Ansible Deployment Only)”	Ansible deployment only. Creating the Ansible user is required on all Ceph nodes.
Enabling password-less SSH	No	Section 2.10, “Enabling Password-less SSH (Ansible Deployment Only)”	Ansible deployment only.

2.1. OPERATING SYSTEM

Red Hat Ceph Storage 2 and later requires Red Hat Enterprise Linux 7 Server with a homogeneous version, for example, Red Hat Enterprise Linux 7.2 running on AMD64 and Intel 64 architectures for all Ceph nodes, including the Red Hat Storage Console node.

Important

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

Return to prerequisite checklist

2.2. REGISTERING TO CDN

Ceph relies on packages in the Red Hat Enterprise Linux 7 Base content set. Each Ceph node must be able to access the full Red Hat Enterprise Linux 7 Base content.

To do so, register Ceph nodes that can connect to the Internet to the Red Hat Content Delivery Network (CDN) and attach appropriate Ceph subscriptions to the nodes:

Registering Ceph Nodes to CDN

Run all commands in this procedure as **root**.

1. Register a node with the Red Hat Subscription Manager. Run the following command and when prompted, enter your Red Hat Customer Portal credentials:

```
# subscription-manager register
```

2. Pull the latest subscription data from the CDN server:

```
# subscription-manager refresh
```

3. List all available subscriptions and find the appropriate Red Hat Ceph Storage subscription and determine its Pool ID.

```
# subscription-manager list --available
```

4. Attach the subscriptions:

```
# subscription-manager attach --pool=<pool-id>
```

   Replace **<pool-id>** with the Pool ID determined in the previous step.

5. Enable the Red Hat Enterprise Linux 7 Server Base repository:

```
# subscription-manager repos --enable=rhel-7-server-rpms
```

6. Update the node:

```
# yum update
```

Once you register the nodes, enable repositories that provide the Red Hat Ceph Storage packages.

Note

For nodes that cannot access the Internet during the installation, provide the Base content by other means. Either use the Red Hat Satellite server in your environment or mount a local Red Hat Enterprise Linux 7 Server ISO image and point the Ceph cluster nodes to it. For additional details, contact the Red Hat Support.

For more information on registering Ceph nodes with the Red Hat Satellite server, see the How to Register Ceph with Satellite 6 and How to Register Ceph with Satellite 5 articles on the Customer Portal.

Return to prerequisite checklist

2.3. 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:

- 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 repositories on each node.

- 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.

2.3.1. Content Delivery Network (CDN)

CDN Installations for…

- **Monitor Nodes**

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

  ```
  # subscription-manager repos --enable=rhel-7-server-rhceph-2-mon-rpms
  ```

- **OSD Nodes**

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

  ```
  # subscription-manager repos --enable=rhel-7-server-rhceph-2-osd-rpms
  ```

- **RADOS Gateway and Client Nodes**

  As **root**, enable the Red Hat Ceph Storage 2 Tools repository:

  ```
  # subscription-manager repos --enable=rhel-7-server-rhceph-2-tools-
  rpms
  ```

- **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:

  ```
  # subscription-manager repos --enable=rhel-7-server-rhscon-2-agent-
  rpms
  ```

- **Red Hat Storage Console and Ansible Installer**

  For Ansible deployment of Red Hat Ceph Storage nodes, as **root**, enable the Red Hat Storage Console 2 Installer repository:

  ```
  # subscription-manager repos --enable=rhel-7-server-rhscon-2-
  installer-rpms
  ```

Return to prerequisite checklist

2.3.2. Local Repository

For ISO Installations…

- **Download the Red Hat Ceph Storage ISO**

  - Log in to the Red Hat Customer Portal.
  - Click **Downloads** to visit the **Software & Download** center.
  - In the Red Hat Ceph Storage area, click **Download Software** to download the latest version of the software.
  - Copy the ISO image to the node.
  - As **root**, mount the copied ISO image to the **/mnt/rhcs2/** directory:

```
# mkdir -p /mnt/rhcs2
# mount -o loop /<path_to_iso>/rhceph-2.0-rhel-7-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 ISO**

  - Log in to the Red Hat Customer Portal.
  - Click **Downloads** to visit the **Software & Download** center.
  - In the Red Hat Ceph Storage area, click **Download Software** to download the latest version of the software.
  - Copy the ISO image to the node.
  - As **root**, mount the copied ISO image to the **/mnt/rhscon2/** directory:

```
# mkdir -p /mnt/rhscon2
# mount -o loop /<path_to_iso>/rhscon-2.0-rhel-7-x86_64.iso
/mnt/rhscon2
```

- **Create a Local Repository**

  - Copy the ISO image to the node.
  - Follow the steps in this Knowledgebase solution.

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.

Return to prerequisite checklist

2.4. 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.

Return to prerequisite checklist

2.5. 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.

Configure the network interfaces and ensure to make the changes persistent so that the settings are identical on reboot. Configure the following settings:

- The **BOOTPROTO** parameter is usually set to **none** for static IP addresses.
- The **ONBOOT** parameter must be set to **yes.** If it is set to **no**, Ceph might fail to peer on reboot.
- If you intend to use IPv6 addressing, the IPv6 parameters, for example **IPV6INIT** must be set to **yes** except for the **IPV6_FAILURE_FATAL** parameter. Also, edit the Ceph configuration file to instruct Ceph to use IPv6. Otherwise, Ceph will use IPv4.

Navigate to the **/etc/sysconfig/network-scripts/** directory and ensure that the **ifcfg-<iface>** settings for the public and cluster interfaces are properly configured.

For details on configuring network interface scripts for Red Hat Enterprise Linux 7, see the Configuring a Network Interface Using ifcfg Files chapter in the Networking Guide for Red Hat Enterprise Linux 7.

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

Return to prerequisite checklist

2.6. 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.

Return to prerequisite checklist

2.7. CONFIGURING FIREWALL

Red Hat Ceph Storage 2 uses the **firewalld** 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. On all Ceph nodes, as **root**, start the **firewalld** service, enable it to run on boot, and ensure that it is running:

```
# systemctl enable firewalld
# systemctl start firewalld
# systemctl status firewalld
```

2. As **root**, on all Ceph Monitor nodes, open port **6789** on the public network:

```
# firewall-cmd --zone=public --add-port=6789/tcp
# firewall-cmd --zone=public --add-port=6789/tcp --permanent
```

   To limit access based on the source address, run the following commands:

```
# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4"
\
source address="<IP-address>/<prefix>" port protocol="tcp" \
port="6789" accept"
# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4"
\
source address="<IP-address>/<prefix>" port protocol="tcp" \
port="6789" accept" --permanent
```

3. If **calamari-lite** is running on the Ceph Monitor node, as **root**, open port **8002** on the public network:

```
# firewall-cmd --zone=public --add-port=8002/tcp
# firewall-cmd --zone=public --add-port=8002/tcp --permanent
```

   To limit access based on the source address, run the following commands:

```
# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4"
\
source address="<IP-address>/<prefix>" port protocol="tcp" \
port="8002" accept"
# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4"
\
source address="<IP-address>/<prefix>" port protocol="tcp" \
port="8002" accept" --permanent
```

4. 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:

```
# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4"
\
source address="<RHSC-IP-address>" port protocol="tcp" \
port="8002" accept"
```

```
# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4"
\
source address="<RHSC-IP-address>" port protocol="tcp" \
port="8002" accept" --permanent
```

Repeat these commands for IPv6 addressing if necessary:

```
# firewall-cmd --zone=public --add-rich-rule="rule family="ipv6"
\
source address="<RHSC-IP-address>" port protocol="tcp" \
port="8002" accept"
# firewall-cmd --zone=public --add-rich-rule="rule family="ipv6"
\
source address="<RHSC-IP-address>" port protocol="tcp" \
port="8002" accept" --permanent
```

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

```
# firewall-cmd --zone=public --add-port=6800-7300/tcp
# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
```

If you have a separate cluster network, repeat the commands with the appropriate zone.

6. As **root**, on all object gateway nodes, open the relevant port or ports on the public network.

   a. To open the default port **7480**:

   ```
   # firewall-cmd --zone=public --add-port=7480/tcp
   # firewall-cmd --zone=public --add-port=7480/tcp --
   permanent
   ```

   To limit access based on the source address, run the following commands:

   ```
   # firewall-cmd --zone=public \
   --add-rich-rule="rule family="ipv4" \
   source address="<IP-address>/<prefix>" \
   port protocol="tcp" port="7480" accept"
   # firewall-cmd --zone=public \
   --add-rich-rule="rule family="ipv4" \
   source address="<IP-address>/<prefix>" \
   port protocol="tcp" port="7480" accept" --permanent
   ```

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

   ```
   # firewall-cmd --zone=public --add-port=80/tcp
   # firewall-cmd --zone=public --add-port=80/tcp --permanent
   ```

   To limit access based on the source address, run the following commands:

   ```
   # firewall-cmd --zone=public \
   --add-rich-rule="rule family="ipv4" \
   source address="<IP-address>/<prefix>" \
   port protocol="tcp" port="80" accept"
   ```

```
# firewall-cmd --zone=public \
--add-rich-rule="rule family="ipv4" \
source address="<IP-address>/<prefix>" \
port protocol="tcp" port="80" accept" --permanent
```

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

```
# firewall-cmd --zone=public --add-port=443/tcp
# firewall-cmd --zone=public --add-port=443/tcp --permanent
```

To limit access based on the source address, run the following commands:

```
# firewall-cmd --zone=public \
--add-rich-rule="rule family="ipv4" \
source address="<IP-address>/<prefix>" \
port protocol="tcp" port="443" accept"
# firewall-cmd --zone=public \
--add-rich-rule="rule family="ipv4" \
source address="<IP-address>/<prefix>" \
port protocol="tcp" port="443" accept" --permanent
```

For additional details on **firewalld**, see the Using Firewalls chapter in the Security Guide for Red Hat Enterprise Linux 7.

Return to prerequisite checklist

2.8. CONFIGURING NETWORK TIME PROTOCOL

You must configure 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:

```
# yum install ntp
```

2. As **root**, enable the NTP service to be persistent across a reboot:

```
# systemctl enable ntpd
```

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

```
# systemctl start ntpd
# systemctl status ntpd
```

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

```
$ ntpq -p
```

For additional details on NTP for Red Hat Enterprise Linux 7, see the Configuring NTP Using ntpd chapter in the System Administrator's Guide for Red Hat Enterprise Linux 7.

Return to prerequisite checklist

2.9. 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:

   ```
   # useradd <username>
   # passwd <username>
   ```

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

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

4. Ensure the correct file permissions:

   ```
   # chmod 0440 /etc/sudoers.d/<username>
   ```

Return to prerequisite checklist

2.10. 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. 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 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>
```

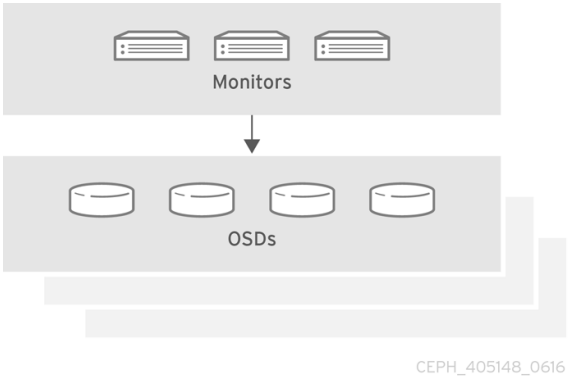
After editing the **~/.ssh/config** file on the Ansible administration node, ensure the permissions are correct:

```
$ 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 2.3, “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. Update to the latest release for Red Hat Enterprise Linux 7:

```
# yum update
```

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

```
# ntpq -p
# date
```

   For more details about NTP, see Section 2.8, “Configuring Network Time Protocol”.

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

```
# subscription-manager repos --enable=rhel-7-server-rhscon-2-
agent-rpms
```

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

```
# subscription-manager repos --enable=rhel-7-server-rhceph-
2-mon-rpms
```

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

```
# subscription-manager repos --enable=rhel-7-server-rhceph-2-osd-
rpms
```

Installing and Configuring

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

   Syntax

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

   Example

```
# curl rhsc.example.com:8181/setup/agent/ | bash
% Total    % Received % Xferd  Average Speed   Time    Time
```

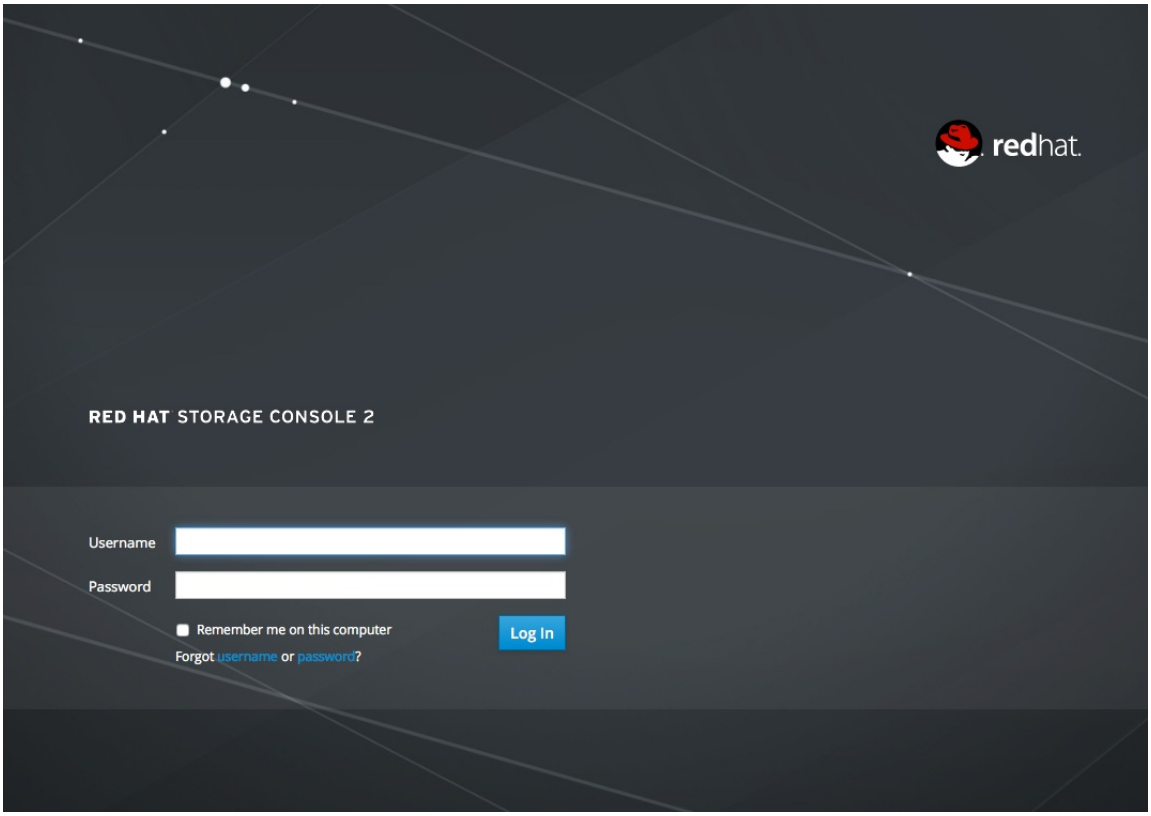
```
Time  Current
                                 Dload  Upload   Total   Spent
Left  Speed
100  1647  100  1647    0     0    98k      0 --:--:-- --:--:--
--:--:--  107k
--> creating new user with disabled password: ceph-installer
Removing password for user ceph-installer.
passwd: Success
--> adding provisioning key to the ceph-installer user
authorized_keys
--> ensuring correct permissions on .ssh/authorized_keys
--> ensuring that ceph-installer user will be able to sudo
--> ensuring ceph-installer user does not require a tty
--> installing and configuring agent
{"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

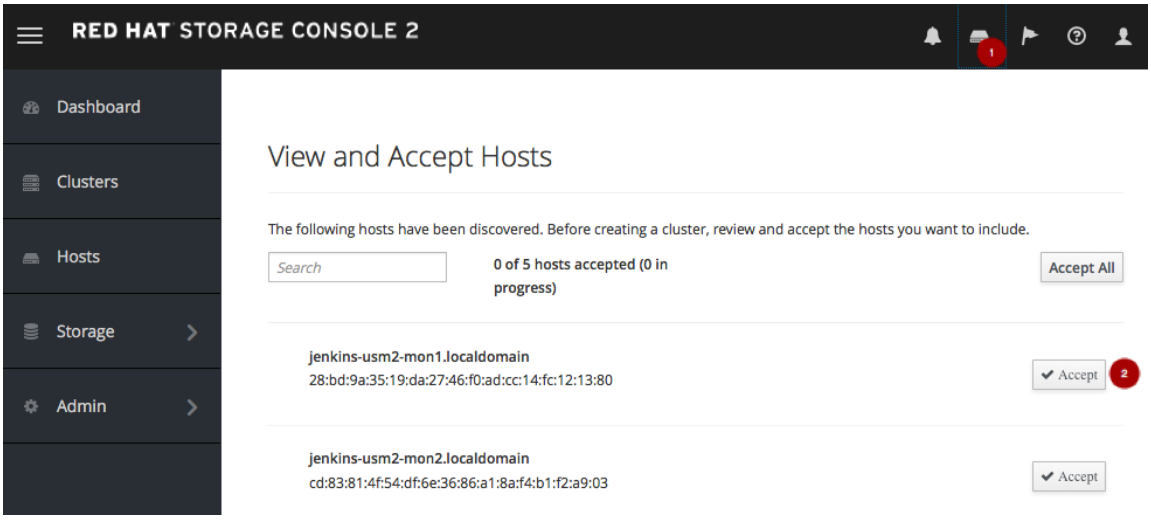
The installing and configuring process of the Red Storage Console Agent will take several minutes to complete, even after the command returns you back to the command prompt. During the configuring process password-less SSH will be configured on the node. To view the status of this process:

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

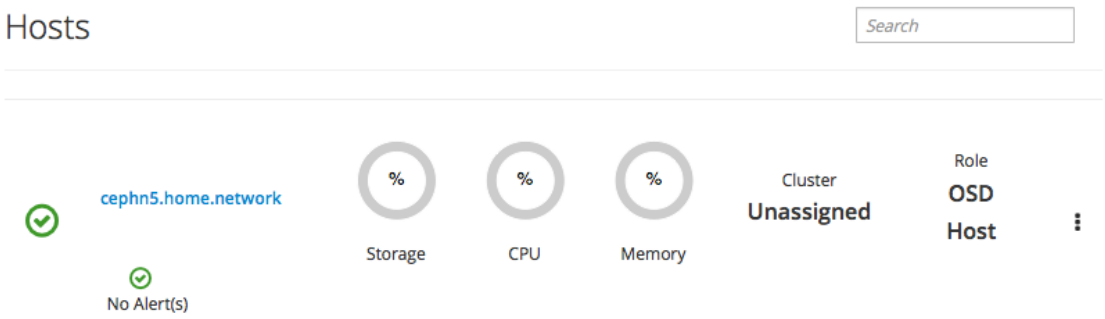
2. Open a web browser from a workstation, and go to the URL for the Red Hat Storage Console web interface. Log in as the administrator using the "admin" user name, and "admin" as the password.



3. In the top-right corner of the web interface, click on the small computer icon (1). This opens a page with a list of discovered systems. Click on the "Accept" button (2) 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:



Note

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. Creating a Ceph Storage Cluster

To create a Red Hat Ceph Storage cluster using the Red Hat Storage Console, see the Red Hat Storage Console Quick Start Guide for details.

3.1.3. Creating a Storage Pool

To create a new object storage pool, see the Red Hat Storage Console Quick Start Guide for details.

3.1.4. 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

You can use the Ansible automation application to deploy Red Hat Ceph Storage. Execute the procedures in Figure 2.1, “Prerequisite Workflow” first.

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

3.2.1. Installing Ceph Ansible

1. Enable the Red Hat Storage Console 2 Installer repository. For ISO-based installations, see

the ISO installation section.

2. Install the **ceph-ansible** package:

```
# yum install ceph-ansible
```

3. As **root**, add the Ceph hosts to the **/etc/ansible/hosts** file. Remember to comment out example hosts.

   If the Ceph hosts have sequential naming, consider using a range:

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

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

   b. Add OSD nodes under the **[osds]** section:

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

   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'
]"
```

   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. See Section A.1, “Ansible Stops Installation Because It Detects Less Devices Than It Expected” for more details.

4. Ensure that Ansible can reach the Ceph hosts:

```
# ansible all -m ping
```

3.2.2. Configuring Ceph Global Settings

1. Create a directory under the home directory so Ansible can write the keys:

```
# cd ~
# mkdir ceph-ansible-keys
```

2. Navigate to the Ceph Ansible **group_vars** directory:

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

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

```
# cp all.sample all
# vim all
```

4. Set the **generate_fsid** setting to **false**:

```
generate_fsid: false
```

Note

With **generate_fsid** set to **false**, then you must specify the value for the Universally Unique Identifier (UUID), uncomment the **fsid** setting and specify the generated UUID:

```
fsid: <generated_uuid>
```

With **generate_fsid** set to **true**, then the UUID will be automatically generated. This removes the need to specify the UUID in the **fsid** setting.

5. Uncomment the **fetch_directory** setting under the **GENERAL** section. Then, point it to the directory you created in step 1:

```
fetch_directory: ~/ceph-ansible-keys
```

6. Uncomment the **ceph_stable_rh_storage** setting and set it to **true**:

```
ceph_stable_rh_storage: true
```

7. Select the installation method. There are two approaches:

    a. If Ceph hosts have connectivity to the Red Hat Content Delivery Network (CDN), uncomment the **ceph_stable_rh_storage_cdn_install** setting and set it to **true**. This is the most common approach to installing Ceph.

    ```
    ceph_stable_rh_storage_cdn_install: true
    ```

    b. If Ceph nodes cannot connect to the Red Hat Content Delivery Network (CDN), uncomment the **ceph_stable_rh_storage_iso_install** setting and set it to **true**. This approach is most frequently used in high security environments.

    ```
    ceph_stable_rh_storage_iso_install: true
    ```

    Then, uncomment the **ceph_stable_rh_storage_iso_path** setting and specify the path to the ISO image:

```
ceph_stable_rh_storage_iso_path
```

Example

```
ceph_stable_rh_storage_iso_path: /path/to/ISO_file.iso
```

8. To enable authentication, uncomment the **cephx** setting under the Ceph Configuration section. Red Hat recommends running Ceph with authentication enabled:

```
cephx: true
```

9. Uncomment the **monitor_interface** setting and specify the network interface:

```
monitor_interface: eth0
```

Note

The **monitor_interface** setting will use the IPv4 address. To use an IPv6 address, use the **monitor_address** setting instead.

10. Set the **journal_size** setting:

```
journal_size: <size_in_MB>
```

See Journal Settings for additional details.

11. Set the **public_network** setting:

```
public_network: <public_network>
```

See Section 2.5, “Configuring Network” and Network Configuration Reference for additional details.

3.2.3. Configuring Monitor Settings

Ansible will create monitors without any additional configuration steps. However, you may override default settings for authentication, and for use with OpenStack. By default, the Calamari API is disabled.

To configure monitors, perform the following:

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

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

2. As **root**, create an **mons** file from **mons.sample** file and open it for editing:

```
# cp mons.sample mons
# vim mons
```

3. To enable the Calamari API, uncomment the **calamari** setting and set it to **true**:

```
calamari: true
```

4. To configure other settings, uncomment them and set appropriate values.

3.2.4. Configuring Ceph OSD Settings

To configure OSDs:

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

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

2. As **root**, create a new **osds** file from the **osds.sample** file and open it for editing:

```
# cp osds.sample osds
# vim osds
```

3. Uncomment and set settings that are relevant for your use case. See Table 3.1, “What settings are needed for my use case?” for details.

4. Once you are done editing the file, save your changes and close the file.

Table 3.1. What settings are needed for my use case?

I want:	Relevant Options	Comments
to have the Ceph journal and OSD data co-located on the same device and to specify OSD disks on my own.	**devices** **journal_collocation: true**	The **devices** setting excepts a list of devices. Ensure that the specified devices correspond to the storage devices on the OSD nodes.
to have the Ceph journal and OSD data co-located on the same device and **ceph-ansible** to detect and configure all the available devices.	**osd_auto_discovery: true** **journal_collocation: true**	

I want:	Relevant Options	Comments
to use one or more dedicated devices to store the Ceph journal.	**`devices`** **`raw_multi_journal: true`** **`raw_journal_devices`**	The **`devices`** and **`raw_journal_devices`** settings except a list of devices. Ensure that the devices specified correspond to the storage devices on the OSD nodes.
to use directories instead of disks.	**`osd_directory: true`** **`osd_directories`**	The **`osd_directories`** setting excepts a list of directories.
to use the BlueStore back end instead of the FileStore back end.	**`devices`** **`bluestore: true`**	The **`devices`** setting excepts a list of devices. For details on OSD BlueStore, see the OSD BlueStore (Technology Preview) chapter in the Administration Guide for Red Hat Ceph Storage.

For additional settings, see the **`osds.sample`** file located in **`/usr/share/ceph-ansible/group_vars/`**.

Warning

Some OSD options will conflict with each other. Avoid enabling these sets of options together:

- **`journal_collocation`** and **`raw_multi_journal`**
- **`journal_collocation`** and **`osd_directory`**
- **`raw_multi_journal`** and **`osd_directory`**

3.2.5. Overriding Ceph Default Settings

Unless otherwise specified in the Ansible configuration files, Ceph will use its default settings. Ceph Ansible manages the Ceph configuration file. So any changes to the Ceph configuration file should be made in the **`/usr/share/ceph-ansible/group_vars/all`** file instead of the Ceph configuration file under **`/etc/ceph`** file. To change Ceph's default settings, open the **`/usr/share/ceph-ansible/group_vars/all`** file, and scroll down to:

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

The **ceph_conf_overrides:** setting allows you to override Ceph configuration defaults. Like the Ceph configuration file, **ceph-ansible** supports the sections of the Ceph configuration file; namely, **[global]**, **[mon]**, **[osd]**, **[mds]**, **[rgw]**, and so on. You may also override particular instances, such as a particular gateway instance:

```
ceph_conf_overrides:
   global:
      osd_pool_default_size: 2
      osd_pool_default_min_size: 1
      cluster_network: 10.0.0.1/24
   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.

3.2.6. Deploying a Ceph Cluster

1. Navigate to the Ansible configuration directory:

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

2. As **root**, create a **site.yml** file from the **site.yml.sample** file:

   ```
   # cp site.yml.sample site.yml
   ```

3. Run the Ansible playbook:

   ```
   # ansible-playbook site.yml [-u <user_name>]
   ```

Once the playbook runs, it creates a running Ceph cluster.

3.2.7. Taking over an Existing Cluster

Ansible can be configured to use a cluster deployed without Ansible. For example, Red Hat Ceph Storage 1.3.x clusters upgraded to version 2 manually can be configured to use Ansible. Use the following procedure:

1. After manually upgrading from version 1.3.x to version 2, see Section 3.2.1, "Installing Ceph Ansible" to install and configure Ansible, this is the node where the master Ceph configuration file is maintained.

2. Ensure the Ansible node has passwordless **ssh** access to all Ceph nodes in the cluster. See Section 2.10, “Enabling Password-less SSH (Ansible Deployment Only)” for more details.
3. Change the directory to **/usr/share/ceph-ansible**.
4. Copy the sample **group_vars/all.sample** to **group_vars/all**.
5. Set the **generate_fsid** setting to **false** in **group_vars/all**.
6. Get the current cluster **fsid** by executing **ceph fsid**.
7. Set the retrieved **fsid** in **group_vars/all**.
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. From the **/usr/share/ceph-ansible** directory run the playbook.

```
# ansible-playbook take-over-existing-cluster.yml -u <username>
```

3.3. INSTALLING RED HAT CEPH STORAGE 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 Yum, 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**:

```
# yum 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 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>'
```

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 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 -R ceph:ceph /etc/ceph
    ```

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

    Syntax

    ```
    # echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
    ```

    Example

    ```
    # echo "CLUSTER=test123" >> /etc/sysconfig/ceph
    ```

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

    Syntax

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

Example

```
# systemctl enable ceph-mon.target
# systemctl enable ceph-mon@node1
# 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: 1 mons 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:

```
# yum 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 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 the following line:

    Syntax

    ```
    # echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
    ```

    Example

    ```
    # echo "CLUSTER=test123" >> /etc/sysconfig/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

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

    Example

    ```
    # systemctl enable ceph-osd.target
    # systemctl enable ceph-osd@0
    # 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 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. The Calamari server runs on Monitor nodes only, and only on one Monitor node per storage cluster.

Note

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

To install **`calamari-server`**, perform the following steps on a Monitor node.

1. As **`root`**, enable the Red Hat Ceph Storage 2 Monitor repository
2. As **`root`**, install **`calamari-server`**:

   ```
   # yum install calamari-server
   ```

   Important

   The Calamari server runs on Monitor nodes only, and only on one Monitor node per storage cluster.

3. As **`root`**, initialize the **`calamari-server`**:

   Syntax

   ```
   # calamari-ctl clear --yes-i-am-sure
   # calamari-ctl initialize --admin-username <uid> --admin-
   ```

```
password <pwd> --admin-email <email>
```

Example

```
# calamari-ctl clear --yes-i-am-sure
# 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

Currently, the Calamari administrator user name and password is hard-coded as **`admin`** and **`admin`** respectively.

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, usernames and passwords are transmitted in clear text.

4. As **`root`**, enable and restart the **`supervisord`** service:

```
# systemctl enable supervisord
# systemctl restart supervisord
```

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. You will need to click through the untrusted certificate warnings, since the auto-generated certificate is self-signed:

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

To use a key and certificate from a CA, perform the following:

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:

```
# calamari-ctl initialize
```

CHAPTER 4. CLIENT INSTALLATION

Red Hat Ceph Storage supports three types of Ceph clients:

Ceph CLI

The Ceph command-line interface (CLI) enables administrators to execute Ceph administrative commands. See Section 4.1, “Ceph Command-line Interface Installation” for information on installing the Ceph CLI.

Block Device

Ceph block device is a thin-provisioned, resizable block device. See Section 4.2, “Ceph Block Device Installation” for information on installing Ceph block devices.

Object Gateway

Ceph object gateway provides its own user management and Swift- and S3-compliant APIs. See Section 4.3, “Ceph Object Gateway Installation” for information on installing Ceph object gateways.

Note

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

Important

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

4.1. 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-authtool`**
- **`ceph-dencoder`**
- **`rados`**

Currently, there is only one way to install the Ceph CLI:

- Using the native operating system tools

4.1.1. Installing Ceph Command-line Interface Manually

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

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

```
# yum 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.

4.2. 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 4.1, “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**:

```
# 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

These features are enabled by default when creating a block device:

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

Users utilizing the kernel RBD client will not be able to map the block device image. You must first disable all these features, except **layering**.

Syntax

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

Example

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

Using the **--image-feature layering** option on the **rbd create** command only enables **layering** on newly created block device images.

This is a known issue, see the Red Hat Ceph Storage 2.0 Release Notes for more details.

All these features work for users utilizing 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:

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

Important

Kernel block devices currently only support the legacy straw bucket algorithm in the CRUSH map. If you have set the CRUSH tunables to optimal, you must set them to legacy or an earlier major release, otherwise, you will not be able to map the image.

Alternatively, replace **`straw2`** with **`straw`** in the CRUSH map. For details, see the Editing a CRUSH Map chapter in the Storage Strategies Guide for Red Hat Ceph Storage 2.

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:

```
# 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:

```
# mkdir /mnt/ceph-block-device
# mount /dev/rbd/rbd/image2 /mnt/ceph-block-device
```

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

4.3. CEPH OBJECT GATEWAY INSTALLATION

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.

For more information about the Ceph object gateway, see the Object Gateway Guide for Red Hat Enterprise Linux.

There are two ways to install the Ceph object gateway:

- Using the Ansible automation application, see Section 4.3.1, “Installing Ceph Object Gateway using Ansible” for details
- Using the comand-line interface, see Section 4.3.2, "Installing Ceph Object Gateway Manually for details

4.3.1. Installing Ceph Object Gateway using Ansible

Perform the following tasks on the Ansible administration node, see Install Ceph Ansible for details.

1. Uncomment the **radosgw_frontend** setting in the **/usr/share/ceph-ansible/group_vars/all** file:

```
radosgw_frontend: civetweb
```

> **Warning**
>
> Do not change the default value of **radosgw_frontend** because the Ceph Object Gateway only supports the CivetWeb web server with Red Hat Ceph Storage.

2. 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/rgws** file:

```
copy_admin_key: true
```

3. You can specify a different default port than the default port **7480**. For example:

```
radosgw_civetweb_port: 80
```

   For SSL/TLS, you can specify port **443** with or without **s** appended, and add the **ssl_certificate** setting with the path to the **.pem** file. For example:

```
rgw frontends = civetweb port=443s
ssl_certificate=/etc/ceph/private/cert.pem
```

4. You can add additional settings to the **ceph_conf_overrides:**. For example, set the **rgw_dns_name:** with the host of your DNS server and ensure your DNS server is configured for wild cards to enable S3 subdomains.

```
ceph_conf_overrides:
   global:
      osd_pool_default_size: 2
      osd_pool_default_min_size: 1
   client.rgw.rgw1:
      rgw_dns_name: {hostname}
```

5. 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, you can use a range. For example:

```
[rgws]
<rgw-host-name-1>
<rgw-host-name-2>
<rgw-host-name[3..10]>
```

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

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

7. Run the Ansible playbook:

```
$ ansible-playbook site.yml
```

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 Red Hat Enterprise Linux for details on configuring a cluster for multi-site.

4.3.2. 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 **ceph-radosgw** package:

```
# yum install ceph-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

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

   Example

```
# 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

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

Example

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

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

Syntax

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

Example

```
# 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

```
# 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

```
# 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

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

Example

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

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

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

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

Syntax

```
# echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
```

Example

```
# echo "CLUSTER=test123" >> /etc/sysconfig/ceph
```

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

```
# firewall-cmd --zone=public --add-port=80/tcp
# firewall-cmd --zone=public --add-port=80/tcp --permanent
```

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

Syntax

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

Example

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.node1
# 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

Important

Due to changes in encoding of the OSD map in the **ceph** package version 10.2.2, upgrading Monitor nodes to Red Hat Ceph Storage 2.0 before OSD nodes can lead to serious performance issues on large clusters that contain hundreds of OSDs.

To work around this issue, upgrade the OSD nodes before the Monitor nodes when upgrading to Red Hat Ceph Storage 2.0 from previous versions.

This issue will be fixed in future releases of Red Hat Ceph Storage 2.

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.

For Red Hat Ceph Storage 1.3.2 Monitor nodes running on Red Hat Enterprise Linux 7, perform the following steps on each Monitor node in the storage cluster. Sequentially upgrading one Monitor node at a time.

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

   ```
   # subscription-manager repos --disable=rhel-7-server-rhceph-1.3-
   mon-rpms --disable=rhel-7-server-rhceph-1.3-installer-rpms --
   disable=rhel-7-server-rhceph-1.3-calamari-rpms
   ```

   Note

   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

   ```
   # service ceph stop <daemon_type>.<monitor_host_name>
   ```

   Example

   ```
   # service ceph stop mon.node1
   ```

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

   ```
   # yum update ceph-mon
   ```

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 -R ceph:ceph /etc/ceph
```

6. If SELinux is set to enforcing mode, then set a relabelling of the SELinux context on files for the next reboot:

```
# touch /.autorelabel
```

> **Warning**
>
> Relabeling will take a long time to complete, because SELinux must traverse every file system and fix any mislabeled files.

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

```
# udevadm trigger
```

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

```
# systemctl enable ceph-mon.target
# 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
```

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

Red Hat recommends having a minimum of three OSD nodes in the Ceph storage cluster. For Red Hat Ceph Storage 1.3.2 OSD nodes running on Red Hat Enterprise Linux 7, perform the following steps on each OSD node in the storage cluster. Sequentially upgrading one OSD node at a time.

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:

```
# 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:

   ```
   # subscription-manager repos --disable=rhel-7-server-rhceph-1.3-
   osd-rpms --disable=rhel-7-server-rhceph-1.3-installer-rpms --
   disable=rhel-7-server-rhceph-1.3-calamari-rpms
   ```

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

   ```
   # service ceph stop <daemon_type>.<osd_id>
   ```

   Example

   ```
   # service ceph stop osd.0
   ```

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

   ```
   # yum update ceph-osd
   ```

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
   ```

6. If SELinux is set to enforcing mode, then set a relabelling of the SELinux context on files for the next reboot:

```
# touch /.autorelabel
```

> **Warning**
>
> Relabeling will take a long time to complete, because SELinux must traverse every file system and fix any mislabeled files.

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

```
# udevadm trigger
```

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

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

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

```
# shutdown -r now
```

10. Move to the next OSD node.

> **Note**
>
> While the **noout** and **norebalance** flags are set, the storage cluster will have a **HEALTH_WARN** status:
>
> ```
> $ ceph health
> HEALTH_WARN noout,norebalance flag(s) set
> ```

Once you are done upgrading the Ceph storage cluster, the previously set OSD flags need to be unset, and you need to 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
```

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 a Ceph Object Gateway Node

Red Hat recommends putting a RADOS Gateway behind a load balancer, such as HAProxy. Remove the RADOS Gateway from the load balancer once no requests are being served, upgrade

the RADOS Gateway node, and then add the RADOS Gateway node back to the load balancer.

1. As **root**, disable the Red Hat Ceph Storage 1.3 repositories:

```
# subscription-manager repos --disable=rhel-7-server-rhceph-1.3-
tools-rpms --disable=rhel-7-server-rhceph-1.3-installer-rpms --
disable=rhel-7-server-rhceph-1.3-calamari-rpms
```

Note

If an ISO-based installation was performed for Red Hat Ceph Storage 1.3.2, then skip this first step.

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

3. As **root**, stop the RADOS Gateway process:

```
# service ceph-radosgw stop
```

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

```
# yum update ceph-radosgw
```

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

6. If SELinux is set to enforcing mode, then set a relabelling of the SELinux context on files for the next reboot:

```
# touch /.autorelabel
```

Warning

Relabeling will take a long time to complete, because SELinux must traverse every file system and fix any mislabeled files.

7. As **root**, enable the **ceph-radosgw** process:

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.<rgw_hostname>
```

8. As **root**, reboot the RADOS Gateway node:

```
# shutdown -r now
```

9. If using a load balancer, then add the node back to the load balancer once the RADOS Gateway node is up.

10. Move to the next RADOS Gateway node.

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:

```
# subscription-manager repos --disable=rhel-7-server-rhceph-1.3-
tools-rpms --disable=rhel-7-server-rhceph-1.3-installer-rpms --
disable=rhel-7-server-rhceph-1.3-calamari-rpms
```

Note

If an ISO-based installation was performed for Red Hat Ceph Storage 1.3.x clients, then skip this first step.

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

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

```
# yum update 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 **qemu-kvm** instances or if using a dedicated **qemu-kvm** client, then stopping and starting the **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 **root**, remove the old Calamari packages:

```
# yum remove calamari-server calamari-client
# yum remove salt-master salt-minion salt
# yum remove diamond
# yum remove graphite
```

2. As **root**, delete the saved Salt keys:

```
# salt-key -D
```

3. As **root**, remove the Calamari Salt files from all nodes in the Ceph storage cluster:

```
# rm /etc/salt/minion.d/calamari.conf
# rm /etc/salt/pki/minion/*
```

4. After upgrading to Red Hat Ceph Storage 2, repurpose the legacy Ceph Administration node by seeing the Red Hat Storage Console Quick Start Guide to install the Red Hat Storage Console or see 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 Section 3.1.1, "Installing and Configuring the Red Hat Storage Console Agent" for details.

To install and configure the new Calamari server, see 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 Quick Start Guide for more details.

5.2. UPGRADING BETWEEN MINOR VERSIONS AND APPLYING ASYNCHRONOUS UPDATES

Use the Ansible **`rolling_update.yml`** playbook 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 Section 3.2.7, “Taking over an Existing Cluster” for details on configuring Ansible to use a cluster that was deployed without it.

Ansible upgrades the Ceph nodes in the following order:

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

Before you start

- 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, Ceph Ansible will wait up to 20 minutes. Ceph Ansible will check the cluster health every 30 seconds, waiting before continuing the upgrade process. Set the following values:

```
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 2.3, “Enabling Ceph Repositories” for details.

Updating the Ceph Storage Cluster by using Ansible

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

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

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

```
upgrade_ceph_packages: True
```

3. Run the **`rolling_update.yml`** playbook:

```
$ ansible-playbook rolling_update.yml
```

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.

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.

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
   shell: "sgdisk --zap-all --clear --mbrtogpt -g -- {{ item.1 }} ||
sgdisk   --zap-all --clear --mbrtogpt -g -- {{ item.1 }}"
   with_together:
    - combined_osd_partition_status_results.results
    - devices
   changed_when: false
   when:
    (journal_collocation or raw_multi_journal) and not
    osd_auto_discovery and
    item.0.rc != 0
```

What this means:

When the **`osd_auto_discovery`** parameter is set to **`true`** in the **`/usr/share/ceph-ansible/group_vars/osds/`** 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 Installing Ceph Ansible for details.