Red Hat Ceph Storage 1.3

Installation Guide for Red Hat Enterprise Linux

Installing Calamari and Red Hat Ceph Storage on Red Hat Enterprise Linux

Last Updated: 2018-03-20

Red Hat Ceph Storage 1.3 Installation Guide for Red Hat Enterprise Linux

Installing Calamari and Red Hat Ceph Storage on Red Hat Enterprise Linux

Legal Notice

Abstract

This document provides instructions for preparing nodes before installation, for downloading Red Hat Ceph Storage, for setting up a local Red Hat Ceph Storage repository, for configuring Calamari, and for creating an initial Red Hat Ceph Storage cluster on Red Hat Enterprise Linux 7 running on AMD64 and Intel 64 architectures.

Table of Contents

CHAPTER 1. OVERVIEW

Designed for cloud infrastructures and web-scale object storage, Red Hat® Ceph Storage is a massively scalable, open, software-defined storage platform that combines the most stable version of Ceph with a Ceph management platform, deployment tools and support services. Providing the tools to flexibly and cost-effectively manage petabyte-scale data deployments in the enterprise, Red Hat Ceph Storage manages cloud data so enterprises can focus on managing their businesses.

This document provides procedures for installing Red Hat Ceph Storage v1.3 for **x86_64** architecture on Red Hat Enterprise Linux (RHEL) 7.

Red Hat® Ceph Storage clusters consist of the following types of nodes:

- **Administration node:** We expect that you will have a dedicated administration node that will host the Calamari monitoring and administration server, your cluster's configuration files and keys, and optionally local repositories for installing Ceph on nodes that cannot access the internet for security reasons.
- **Monitor nodes:** Ceph can run with one monitor; however, for high availability, we expect that you will have at least three monitor nodes to ensure high availability in a production cluster.
- **OSD nodes:** Ceph can run with very few OSDs (3, by default), but production clusters realize better performance beginning at modest scales (e.g., 50 OSDs). Ideally, a Ceph cluster will have multiple OSD nodes, allowing you to create a CRUSH map to isolate failure domains.

For minimum recommended hardware, see the Hardware Guide.

1.1. PREREQUISITES

Before installing Red Hat Ceph Storage, review the following prerequisites first and prepare the cluster nodes.

1.1.1. Operating System

Red Hat Ceph Storage 1.3 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.

1.1.2. 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 udpate
```

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.

1.1.3. Enable Ceph Repositories

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, you may use Red Hat Subscription Manager on each node to enable the required Ceph repositories for Calamari, Ceph CLI tools, monitors and OSDs as needed.

- **Local Repository:** For Ceph Storage clusters where security measures preclude nodes from accessing the internet, you may install Red Hat Ceph Storage v1.3 from a single software build delivered as an ISO with the **ice_setup** package, which installs the **ice_setup** program.

When you execute the **ice_setup** program, it will install local repositories, the Calamari monitoring and administration server and the Ceph installation scripts, including a hidden **.cephdeploy.conf** file pointing **ceph-deploy** to the local repositories.

For CDN-based installations, enable the appropriate repository(ies) for each node.

1. For your administration node, enable the Calamari, installer (**ceph-deploy**) and tools repositories:

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

```
# yum update
```

2. For Ceph monitor nodes, enable the monitor repository:

```
# subscription-manager repos --enable=rhel-7-server-rhceph-1.3-mon-
rpms
```

```
# yum update
```

3. For OSD nodes, enable the OSD repository:

```
# subscription-manager repos --enable=rhel-7-server-rhceph-1.3-osd-
rpms
```

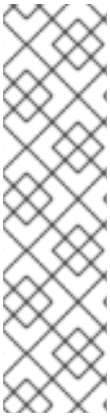
```
# yum update
```

For ISO-based installations, enable ALL repositories on the administration node ONLY.

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

```
# yum update
```

Finally, disconnect your admin node from the internet.



NOTE

With ISO-based installations, **ceph-deploy** accesses local repositories on your administration node, so the Ceph Storage nodes can retrieve all the required packages without a need to access the internet. If the admin node can access the internet, you can receive online updates and publish them to the rest of the cluster from your admin node with **ice_setup update**. If the admin node cannot access the internet, you must use ISOs to handle any updates. So, the above step to enable all **Ceph** repositories to receive online updates is optional but it is mandatory to enable the **rhel-7-server-rhceph-1.3-tools-rpms** repo to get **ceph-common** package. The ISO doesn't include the **tools** repo.

1.1.4. 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's 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.

1.1.5. Network Interface Cards

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 OSDs. You SHOULD 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.

We DO NOT RECOMMEND using a single NIC for both a public and private network.

1.1.6. Network

Ensure that you configure your network interfaces and make them persistent so that the settings are identical on reboot. For example:

- **`BOOTPROTO`** will usually be **`none`** for static IP addresses.
- **`IPV6{opt}`** settings MUST be set to **`yes`** except for **`FAILURE_FATAL`** if you intend to use IPv6. You must also set your Ceph configuration file to tell Ceph to use IPv6 if you intend to use it. Otherwise, Ceph will use IPv4.
- **`ONBOOT`** MUST be set to **`yes.`** If it is set to **`no`**, Ceph may fail to peer on reboot.

Navigate to **`/etc/sysconfig/network-scripts`** and ensure that the **`ifcfg-<iface>`** settings for your public and cluster interfaces (assuming you will use a cluster network too [RECOMMENDED]) are properly configured.

For details on configuring network interface scripts for RHEL 7, see Configuring a Network Interface Using ifcfg Files.

1.1.7. Firewall

Red Hat® Ceph Storage v1.3 uses **`firewalld`**. Start the firewall and ensure that you enable it to run on boot.

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

Ensure **`firewalld`** is running:

```
# systemctl status firewalld.service
```

The default firewall configuration for RHEL is fairly strict. You **MUST** adjust your firewall settings on the Calamari node to allow inbound requests on port **`80`** so that clients in your network can access the Calamari web user interface.

Calamari also communicates with Ceph nodes via ports **2003**, **4505** and **4506**. For **firewalld**, add port **80**, **4505**, **4506** and **2003** to the public zone of your Calamari administration node and ensure that you make the setting permanent so that it is enabled on reboot.

You **MUST** open ports **80**, **2003**, and **4505-4506** on your Calamari node. First, open the port to ensure it opens immediately at runtime. Then, rerun the command with **--permanent** to ensure that the port opens on reboot.

```
# firewall-cmd --zone=public --add-port=80/tcp
# firewall-cmd --zone=public --add-port=80/tcp --permanent
# firewall-cmd --zone=public --add-port=2003/tcp
# firewall-cmd --zone=public --add-port=2003/tcp --permanent
# firewall-cmd --zone=public --add-port=4505-4506/tcp
# firewall-cmd --zone=public --add-port=4505-4506/tcp --permanent
```

You **MUST** open port **6789** on your public network on **ALL Ceph monitor nodes**.

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

Finally, you **MUST** also open ports for OSD traffic (**6800-7300**). **Each OSD on each Ceph node** needs a few ports: one for talking to clients and monitors (public network); one for sending data to other OSDs (cluster network, if available; otherwise, public network); and, one for heartbeating (cluster network, if available; otherwise, public network). To get started quickly, open up the default port range. For example:

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

For additional details on **firewalld**, see Using Firewalls.

1.1.8. Network Time Protocol

You MUST install Network Time Protocol (NTP) on all Ceph monitor nodes and admin nodes. Ensure that ceph nodes are NTP peers. You SHOULD consider installing NTP on Ceph OSD nodes, but it is not required. NTP helps preempt issues that arise from clock drift.

1. Install NTP

   ```
   # yum install ntp
   ```

2. Make sure NTP starts on reboot.

   ```
   # systemctl enable ntpd.service
   ```

3. Start the NTP service and ensure it's running.

   ```
   # systemctl start ntpd
   ```

   Then, check its status.

   ```
   # systemctl status ntpd
   ```

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

```
ntpq -p
```

For additional details on NTP for RHEL 7, see Configuring NTP Using ntpd.

1.1.9. Install SSH Server

For **ALL** Ceph Nodes perform the following steps:

1. Install an SSH server (if necessary) on each Ceph Node:

```
# yum install openssh-server
```

2. Ensure the SSH server is running on **ALL** Ceph Nodes.

1.1.10. Create a Ceph Deploy User

The **ceph-deploy** utility must login to a Ceph node as a user that has passwordless **sudo** privileges, because it needs to install software and configuration files without prompting for passwords.

ceph-deploy supports a **--username** option so you can specify any user that has password-less **sudo** (including **root**, although this is **NOT** recommended). To use **ceph-deploy --username <username>**, the user you specify must have password-less SSH access to the Ceph node, because **ceph-deploy** will not prompt you for a password.

We recommend creating a Ceph Deploy user on **ALL** Ceph nodes in the cluster. Please do **NOT** use "ceph" as the user name. A uniform user name across the cluster may improve ease of use (not required), but you should avoid obvious user names, because hackers typically use them with brute force hacks (for example, **root**, **admin**, or **<productname>**). The following procedure, substituting **<username>** for the user name you define, describes how to create a Ceph Deploy user with passwordless **sudo** on a Ceph node.

NOTE

In a future Ceph release, the "ceph" user name will be reserved for the Ceph daemons. If the "ceph" user already exists on the Ceph nodes, removing this user must be done before attempting an upgrade to future Ceph releases.

1. Create a Ceph Deploy user on each Ceph Node.

```
ssh root@<hostname>
adduser <username>
passwd <username>
```

Replace **<hostname>** with the hostname of your Ceph node.

2. For the Ceph Deploy user you added to each Ceph node, ensure that the user has **sudo** privileges and disable **requiretty**.

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

```
EOF
```

Ensure the correct file permissions.

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

1.1.11. Enable Password-less SSH

Since **ceph-deploy** will not prompt for a password, you must generate SSH keys on the admin node and distribute the public key to each Ceph node. **ceph-deploy** will attempt to generate the SSH keys for initial monitors.

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 with Create a Ceph Deploy User.

```
ssh-copy-id <username>@node1
ssh-copy-id <username>@node2
ssh-copy-id <username>@node3
```

3. (Recommended) Modify the **~/.ssh/config** file of your **ceph-deploy** admin node so that **ceph-deploy** can log in to Ceph nodes as the user you created without requiring you to specify **--username <username>** each time you execute **ceph-deploy**. This has the added benefit of streamlining **ssh** and **scp** usage. Replace **<username>** with the user name you created:

```
Host node1
   Hostname node1
   User <username>
Host node2
   Hostname node2
   User <username>
Host node3
   Hostname node3
   User <username>
```

After editing the **~/.ssh/config** file on the **admin node**, execute the following to ensure the permissions are correct:

```
chmod 600 ~/.ssh/config
```

1.1.12. Adjust `ulimit` on Large Clusters

For users that will run Ceph administrator commands on large clusters (for example, 1024 OSDs or more), create an **/etc/security/limits.d/50-ceph.conf** file on your admin node with the following contents:

```
<username>       soft    nproc     unlimited
```

Replace **<username>** with the name of the non-root account that you will use to run Ceph administrator commands.

NOTE

The **root** user's **ulimit** is already set to "unlimited" by default on RHEL.

1.1.13. Configuring RAID Controllers

If a RAID controller with 1-2 GB of cache is installed on a host, then enabling write-back caches might result in increased small I/O write throughput. In order for this to be done safely, 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 of them might 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 might 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 SAS or SATA connected SSDs are also present on the controller, it is worth investigating whether your controller and firmware support passthrough mode. This will avoid the caching logic, and generally result in much lower latencies for fast media.

1.1.14. Adjust PID Count

Hosts with high numbers of OSDs (more than 12) may spawn a lot of threads, especially during recovery and re-balancing. The standard RHEL 7 kernel defaults to a relatively small maximum number of threads (**32768**). Check your default settings to see if they are suitable.

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

Consider setting **kernel.pid_max** to a higher number of threads. The theoretical maximum is 4,194,303 threads. For example, you could add the following to the **/etc/sysctl.conf** file to set it to the maximum:

```
kernel.pid_max = 4194303
```

To see the changes you made without a reboot, execute:

```
# sysctl -p
```

To verify the changes, execute:

```
# sysctl -a | grep kernel.pid_max
```

1.1.15. Adjust Netfilter conntrack Limits

When using a firewall and running several OSDs on a single host, busy clusters might create a lot of network connections and overflow the kernel **nf_conntrack** table on the OSD and monitor hosts. To find the current values, execute the following commands:

```
cat /proc/sys/net/netfilter/nf_conntrack_buckets
cat /proc/sys/net/netfilter/nf_conntrack_max
```

The **nf_conntrack_max** value defaults to the **nf_conntrack_buckets** value multiplied by 8. Consider setting **nf_conntrack_buckets** to a higher number on the OSD and monitor hosts. To do so, create a new **/etc/modprobe.d/ceph.conf** file with the following content:

```
options nf_conntrack hashsize=<size>
```

Where **<size>** specifies the new size of the **nf_conntrack_buckets** value. For example:

```
options nf_conntrack hashsize=128000
```

Having this option specified loads the **nf_conntrack** module with a maximum table size of 1024000 (128000 * 8).

To see the changes you made without a reboot, execute the following commands as **root**:

```
# systemctl stop firewalld
# modprobe -rv nf_conntrack
# systemctl start firewalld
```

To verify the changes, execute the following commands as **root**:

```
# sysctl -a | grep conntrack_buckets
# sysctl -a | grep conntrack_max
```

1.1.16. SELinux

SELinux is set to **enforcing** mode by default. For **Red Hat Ceph Storage 1.3** and **1.3.1**, set SELinux to **permissive** mode, or disable it entirely and ensure that your installation and cluster are working properly. To set SELinux to **permissive**, execute the following command:

```
# setenforce 0
```

To configure SELinux persistently, modify the **/etc/selinux/config** configuration file.

With **Red Hat Ceph Storage 1.3.2** or later, a new **ceph-selinux** package can be installed on Ceph nodes. This package provides SELinux support for Ceph, and SELinux therefore no longer needs to be in **permissive** or **disabled** mode. See the Install ceph-selinux section for detailed information on installing **ceph-selinux** and enabling SELinux.

1.1.17. Disable EPEL on Cluster Nodes

Some Ceph package dependencies require versions that differ from the package versions from EPEL. Disable EPEL to ensure that you install the packages required for use with Ceph.

To disable **epel**, execute:

```
# yum-config-manager --disable epel
```

The above command will disable **epel.repo** in **/etc/yum.repos.d/**.

1.2. SETTING UP THE ADMINISTRATION SERVER

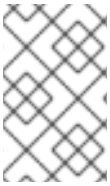
You are supposed to have a dedicated administration node that hosts the Calamari monitoring and administration server.

The administration server hardware requirements vary with the size of the cluster. A minimum recommended hardware configuration for a Calamari server includes:

- at least 4 GB of RAM,
- a dual core CPU on AMD64 or Intel 64 architecture,
- enough network throughput to handle communication with Ceph hosts.

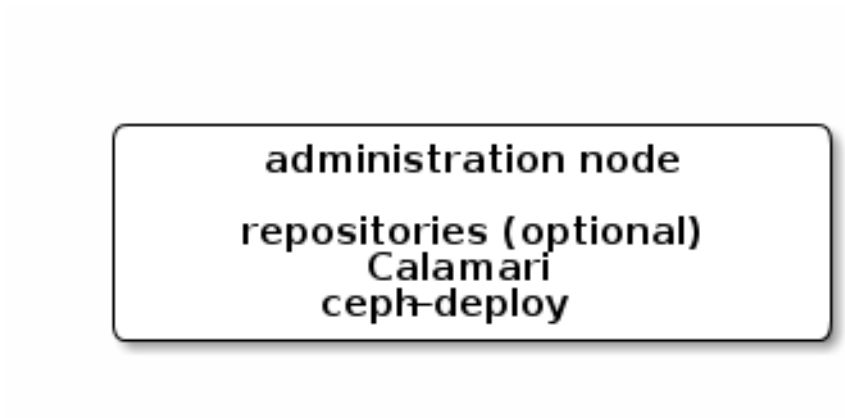
The hardware requirements scale linearly with the number of Ceph servers, so if you intend to run a fairly large cluster, ensure that you have enough RAM, processing power, and network throughput for the administration node.

Red Hat Ceph Storage uses an administration server for the Calamari monitoring and administration server, and the cluster's Ceph configuration file and authentication keys. If you install Red Hat Ceph Storage from an ISO image and require local repositories for the Ceph packages, the administration server will also contain the Red Hat Ceph Storage repositories.



NOTE

To use the HTTPS protocol with Calamari, set up the Apache web server first. See the Setting Up an SSL Server chapter in the System Administrator's Guide for Red Hat Enterprise Linux 7.



1.2.1. Create a Working Directory

Create a working directory for the Ceph cluster configuration files and keys. Then, navigate to that directory. For example:

```
mkdir ~/ceph-config
cd ~/ceph-config
```

The **ice-setup** and **ceph-deploy** utilities must be executed within this working directory. See the Installation by ISO and Executing ceph-deploy sections for details.

1.2.2. Installation by CDN

To install the **ceph-deploy** utility and the Calamari server by using the Red Hat Content Delivery Network (CDN), execute the following command as **root**:

```
# yum install ceph-deploy calamari-server calamari-clients
```

1.2.3. Installation by ISO

To install the **ceph-deploy** utility and the Calamari server by using the ISO image, perform the following steps:

1. Log in to the Red Hat Customer Portal.
2. Click **Downloads** to visit the **Software & Download** center.
3. In the Red Hat Ceph Storage area, click **Download Software** to download the latest version of the software.
4. As **root**, mount the downloaded ISO image to the **/mnt/** directory, for example:

   ```
   # mount <path_to_iso>/rhceph-1.3.2-rhel-7-x86_64-dvd.iso /mnt
   ```

5. As **root**, install the **ice_setup** program:

   ```
   # yum install /mnt/Installer/ice_setup-*.rpm
   ```

6. Navigate to the working directory that you created in the Create a Working Directory section, for example:

   ```
   $ cd ~/ceph-config
   ```

7. Withing this directory, run **ice_setup** as **root**:

   ```
   # ice_setup -d /mnt
   ```

   The **ice_setup** program performs the following operations:

   - moves the RPM packages to the **/opt/calamari/** directory
   - creates a local repository for the **ceph-deploy** and **calamari** packages
   - installs the Calamari server on the administration node

- installs the **ceph-deploy** package on the administration node
- writes the **.cephdeploy.conf** file to the **/root/** directory and to the current working directory, for example, **~/ceph-config**
- prints further instructions regarding **ceph-deploy** to the console

1.2.4. Initialize Calamari

Once you have installed the Calamari package by using either the Content Delivery Network or the ISO image, initialize the Calamari monitoring and administration server:

```
# calamari-ctl initialize
```

As **root**, update existing cluster nodes that report to Calamari.

```
# salt '*' state.highstate
```

At this point, you should be able to access the Calamari web server using a web browser. Proceed to the Storage Cluster Quick Start.

NOTE

The initialization program implies that you can only execute **ceph-deploy** when pointing to a remote site. You may also direct **ceph-deploy** to your Calamari administration node for example, **ceph-deploy admin <admin-hostname>**. You can also use the Calamari administration node to run a Ceph daemon, although this is not recommended.

1.3. UPDATING THE ADMINISTRATION SERVER

Red Hat provides updated packages for Red Hat Ceph Storage periodically. For CDN-based installations, execute:

```
# yum update
```

For ISO-based installations, get the latest version of **ice_setup** and upgrade the administration server with the latest packages. To update the administration server, perform the following steps:

1. As **root**, update the Calamari administration node to the latest version of **ice_setup**. Note to have at least version 0.3.0:

   ```
   # yum update ice_setup
   ```

2. As **root**, run **ice_setup** with the **update all** subcommand. The **ice_setup** utility synchronizes the new packages from the Red Hat CDN onto the local repositories on the Calamari administration node.

   ```
   # ice_setup update all
   ```

3. The updated packages are now available to the nodes in the cluster with **yum update**:

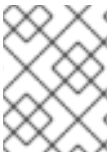
   ```
   # yum update
   ```

If the updates contain new packages for the Ceph cluster, upgrade the cluster too. See Upgrading Ceph Storage for details.

1.3.1. Notes for Update After Upgrading Red Hat Enterprise Linux 6 to 7

Upgrading from Red Hat Enterprise Linux 6 to 7 requires either removing the PostgreSQL data or migrating it. Consider migrating this data if you have services other than Calamari using PostgreSQL. See the How to Migrate PostgreSQL Databases from RHEL6 to RHEL7 article for details.

If you do not have services other than Calamari using the PostgreSQL database, proceed as follows. In Red Hat Ceph Storage 1.2, Calamari stores only crash-recovery data in PostgreSQL. All that data will be rebuilt when first connected to the Ceph cluster. It is harmless to delete the data during this transition.



NOTE

Ensure to recreate the Calamari account when running the **`calamari-ctl initialize`** command.

All the commands in the following steps must be run as **`root`**.

On the Calamari node perform these steps:

1. Remove the PostgreSQL data:

```
# rm -rf /var/lib/pgsql/*
```

2. Proceed with Calamari installation as described in Section 1.3, “Updating the Administration Server”. Note that during updating of the **`calamari-server`** package, errors similar to the following can appear:

```
Updating   : calamari-server-1.3-11.el7cp.x86_64
1/4
mv: cannot stat ‘/etc/httpd/conf.d/welcome.conf’: No such file or
directory
Redirecting to /bin/systemctl restart  salt-master.service
Warning: salt-master.service changed on disk. Run 'systemctl daemon-
reload' to reload units.
Job for salt-master.service failed because the control process
exited with error code. See "systemctl status salt-master.service"
and "journalctl -xe" for details.
Redirecting to /bin/systemctl stop  supervisord.service
Warning: supervisord.service changed on disk. Run 'systemctl daemon-
reload' to reload units.
Redirecting to /bin/systemctl start  supervisord.service
Warning: supervisord.service changed on disk. Run 'systemctl daemon-
reload' to reload units.
Job for supervisord.service failed because the control process
exited with error code. See "systemctl status supervisord.service"
and "journalctl -xe" for details.
Thank you for installing Calamari.
```

3. Reload all services:

```
# systemctl daemon-reload
```

4. Ensure that the **salt-master** service is not running:

```
# killall salt-master
```

5. Start the **salt-master** service:

```
# systemctl start salt-master.service
```

6. Restart the **supervisord** service:

```
# systemctl restart supervisord.service
```

7. Identify cluster nodes previously connected to Calamari:

```
# salt-key -L
```

8. Delete all keys:

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

On nodes that were previously connected to Calamari and that you identified in step 7, perform the following steps:

1. Check if the old **/var/lock/subsys/diamond** lock file exists and if so, delete the file and restart the **diamond** service:

```
# rm /var/lock/subsys/diamond
# systemctl restart diamond.service
```

2. Remove the old **salt-master** public key:

```
# rm /etc/salt/pki/minion/minion_master.pub
```

3. Disable all repositories and enable the **rhel-7-server-rhceph-1.3-calamari-rpms** repository. Then, update the node:

```
# yum --disablerepo=* --enablerepo=rhel-7-server-rhceph-1.3-
calamari-rpms update
```

4. Reload all services:

```
# systemctl daemon-reload
```

5. Restart the **salt-minion** service:

```
# systemctl restart salt-minion
```

On the Calamari node, perform the following steps:

1. Add the **salt** minions to Calamari:

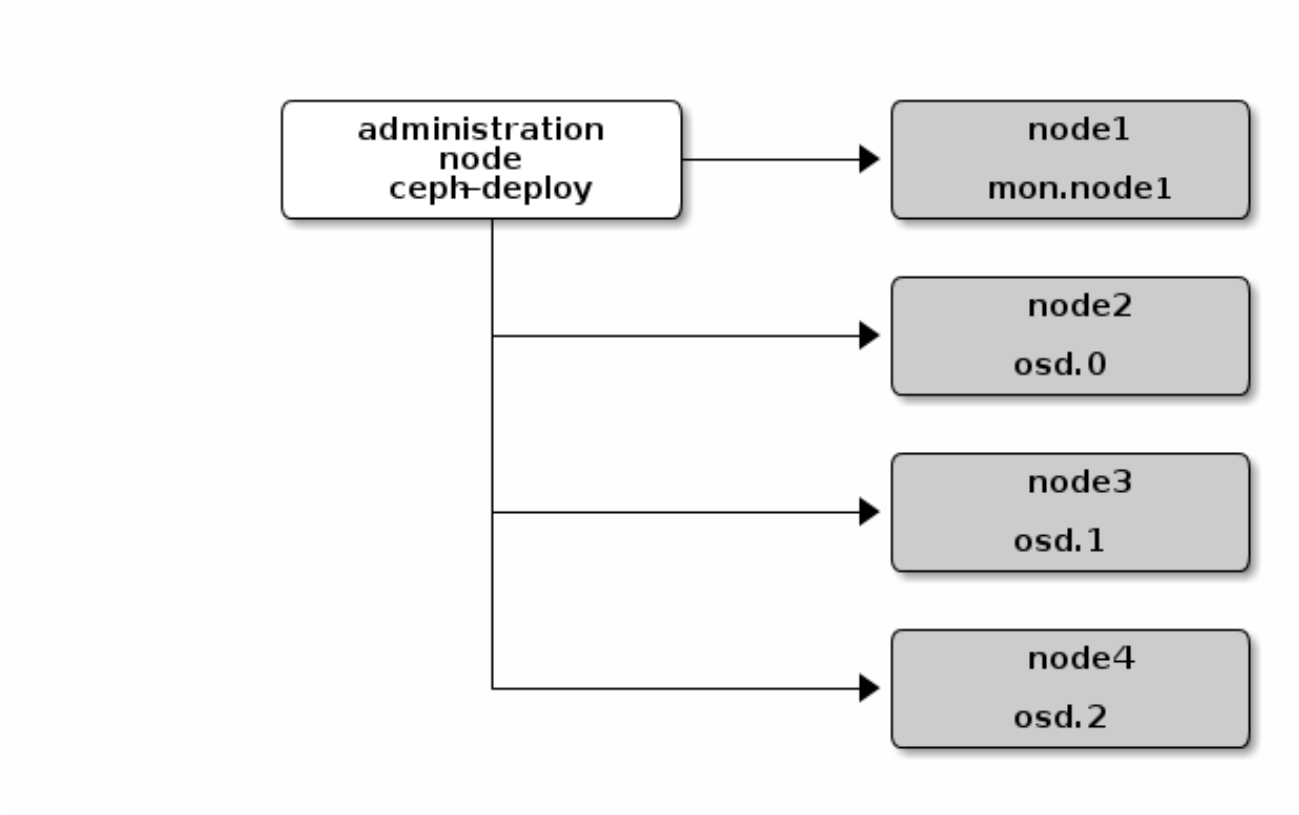
```
# salt-key -A
```

2. Update the Ceph module:

```
# salt '*' state.highstate
```

CHAPTER 2. STORAGE CLUSTER QUICK START

This **Quick Start** sets up a Red Hat Ceph Storage cluster using **`ceph-deploy`** on your Calamari admin node. Create a small Ceph cluster so you can explore Ceph functionality. As a first exercise, create a Ceph Storage Cluster with one Ceph Monitor and some Ceph OSD Daemons, each on separate nodes. Once the cluster reaches an **`active + clean`** state, you can use the cluster.



2.1. EXECUTING `CEPH-DEPLOY`

When executing **`ceph-deploy`** to install the Red Hat Ceph Storage, **`ceph-deploy`** retrieves Ceph packages from the **`/opt/calamari/`** directory on the Calamari administration host. To do so, **`ceph-deploy`** needs to read the **`.cephdeploy.conf`** file created by the **`ice_setup`** utility. Therefore, ensure to execute **`ceph-deploy`** in the local working directory created in the Create a Working Directory section, for example **`~/ceph-config/`**:

```
cd ~/ceph-config
```

IMPORTANT

Execute **`ceph-deploy`** commands as a regular user not as **`root`** or by using **`sudo`**. The Create a Ceph Deploy User and Enable Password-less SSH steps enable **`ceph-deploy`** to execute as **`root`** without **`sudo`** and without connecting to Ceph nodes as the **`root`** user. You might still need to execute **`ceph`** CLI commands as **`root`** or by using **`sudo`**.

2.2. CREATE A CLUSTER

If at any point you run into trouble and you want to start over, execute the following to purge the configuration:

-

```
ceph-deploy purge <ceph-node> [<ceph-node>]
ceph-deploy purgedata <ceph-node> [<ceph-node>]
ceph-deploy forgetkeys
```

If you execute the foregoing procedure, you must re-install Ceph.

On your Calamari admin node from the directory you created for holding your configuration details, perform the following steps using **ceph-deploy**.

1. Create the cluster:

   ```
   ceph-deploy new <initial-monitor-node(s)>
   ```

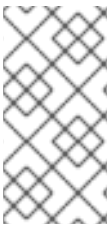
   For example:

   ```
   ceph-deploy new node1
   ```

   Check the output of **ceph-deploy** with **ls** and **cat** in the current directory. You should see a Ceph configuration file, a monitor secret keyring, and a log file of the **ceph-deploy** procedures.

2.3. MODIFY THE CEPH CONFIGURATION FILE

At this stage, you may begin editing your Ceph configuration file (**ceph.conf**).



NOTE

If you choose not to use **ceph-deploy** you will have to deploy Ceph manually or configure a deployment tool (e.g., Chef, Juju, Puppet, etc.) to perform each operation that **ceph-deploy** performs for you. To deploy Ceph manually, please see our Knowledgebase article.

1. Add the **public_network** and **cluster_network** settings under the **[global]** section of your Ceph configuration file.

   ```
   public_network = <ip-address>/<netmask>
   cluster_network = <ip-address>/<netmask>
   ```

   These settings distinguish which network is public (front-side) and which network is for the cluster (back-side). Ensure that your nodes have interfaces configured for these networks. We do not recommend using the same NIC for the public and cluster networks. Please see the Network Configuration Settings for details on the public and cluster networks.

2. Turn on IPv6 if you intend to use it.

   ```
   ms_bind_ipv6 = true
   ```

   Please see Bind for more details.

3. Add or adjust the **osd journal size** setting under the **[global]** section of your Ceph configuration file.

   ```
   osd_journal_size = 10000
   ```

We recommend a general setting of 10GB. Ceph's default **osd_journal_size** is **0**, so you will need to set this in your **ceph.conf** file. A journal size should be the product of the **filestore_max_sync_interval** option and the expected throughput, and then multiply the resulting product by two. The expected throughput number should include the expected disk throughput (i.e., sustained data transfer rate), and network throughput. For example, a 7200 RPM disk will likely have approximately 100 MB/s. Taking the **min()** of the disk and network throughput should provide a reasonable expected throughput. Please see Journal Settings for more details.

4. Set the number of copies to store (default is **3**) and the default minimum required to write data when in a **degraded** state (default is **2**) under the **[global]** section of your Ceph configuration file. We recommend the default values for production clusters.

```
osd_pool_default_size = 3
osd_pool_default_min_size = 2
```

For a quick start, you may wish to set **osd_pool_default_size** to **2**, and the **osd_pool_default_min_size** to 1 so that you can achieve and **active+clean** state with only two OSDs.

These settings establish the networking bandwidth requirements for the cluster network, and the ability to write data with eventual consistency (i.e., you can write data to a cluster in a degraded state if it has **min_size** copies of the data already). Please see Settings for more details.

5. Set a CRUSH leaf type to the largest serviceable failure domain for your replicas under the **[global]** section of your Ceph configuration file. The default value is **1**, or host, which means that CRUSH will map replicas to OSDs on separate separate hosts. For example, if you want to make three object replicas, and you have three racks of chassis/hosts, you can set **osd_crush_chooseleaf_type** to **3**, and CRUSH will place each copy of an object on OSDs in different racks.

```
osd_crush_chooseleaf_type = 3
```

The default CRUSH hierarchy types are:

- type 0 osd
- type 1 host
- type 2 chassis
- type 3 rack
- type 4 row
- type 5 pdu
- type 6 pod
- type 7 room
- type 8 datacenter
- type 9 region
- type 10 root

Please see Settings for more details.

6. Set **max_open_files** so that Ceph will set the maximum open file descriptors at the OS level to help prevent Ceph OSD Daemons from running out of file descriptors.

```
max_open_files = 131072
```

Please see the General Configuration Reference for more details.

In summary, your initial Ceph configuration file should have at least the following settings with appropriate values assigned after the **=** sign:

```
[global]
fsid = <cluster-id>
mon_initial_members = <hostname>[, <hostname>]
mon_host = <ip-address>[, <ip-address>]
public_network = <network>[, <network>]
cluster_network = <network>[, <network>]
ms_bind_ipv6 = [true | false]
max_open_files = 131072
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_crush_chooseleaf_type = <n>
```

2.4. INSTALL CEPH WITH THE ISO

To install Ceph from a local repository, use the **--repo** argument first to ensure that **ceph-deploy** is pointing to the **.cephdeploy.conf** file generated by **ice_setup** (e.g., in the exemplary **~/ceph-config** directory, the **/root** directory, or **~**). Otherwise, you may not receive packages from the local repository. Specify **--release=<daemon-name>** to specify the daemon package you wish to install. Then, install the packages. Ideally, you should run **ceph-deploy** from the directory where you keep your configuration (e.g., the exemplary **~/ceph-config**) so that you can maintain a **{cluster-name}.log** file with all the commands you have executed with **ceph-deploy**.

```
ceph-deploy install --repo --release=[ceph-mon|ceph-osd] <ceph-node>
[<ceph-node> ...]
ceph-deploy install --<daemon> <ceph-node> [<ceph-node> ...]
```

For example:

```
ceph-deploy install --repo --release=ceph-mon monitor1 monitor2 monitor3
ceph-deploy install --mon monitor1 monitor2 monitor3
```

```
ceph-deploy install --repo --release=ceph-osd srv1 srv2 srv3
ceph-deploy install --osd srv1 srv2 srv3
```

The **ceph-deploy** utility will install the appropriate Ceph daemon on each node.

NOTE

If you use **`ceph-deploy purge`**, you must re-execute this step to re-install Ceph.

2.5. INSTALL CEPH BY USING CDN

When installing Ceph on remote nodes from the CDN (not ISO), you must specify which Ceph daemon you wish to install on the node by passing one of **`--mon`** or **`--osd`** to **`ceph-deploy`**.

```
ceph-deploy install [--mon|--osd] <ceph-node> [<ceph-node> ...]
```

For example:

```
ceph-deploy install --mon monitor1 monitor2 monitor3
```

```
ceph-deploy install --osd srv1 srv2 srv3
```

NOTE

If you use **`ceph-deploy purge`**, you must re-execute this step to re-install Ceph.

2.6. INSTALL `CEPH-SELINUX`

With **Red Hat Ceph Storage 1.3.2** or later, a new **`ceph-selinux`** package can be installed on Ceph nodes. This package provides SELinux support for Ceph and SELinux therefore no longer needs to be in **`permissive`** or **`disabled`** mode.

Once installed, **`ceph-selinux`** adds the SELinux policy for Ceph and also relabels files on the cluster accordingly. Ceph processes are labeled with the **`ceph_exec_t`** SELinux context.

To install **`ceph-selinux`**, use the following command:

```
ceph-deploy pkg --install ceph-selinux <nodes>
```

For example:

```
ceph-deploy pkg --install ceph-selinux node1 node2 node3
```

NOTE

All Ceph daemons will be down for the time the **`ceph-selinux`** package is being installed. Therefore, your cluster will not be able to serve any data at this point. This operation is necessary in order to update the metadata of the files located on the underlying file system and to make Ceph daemons run with the correct context. This operation may take several minutes depending on the size and speed of the underlying storage.

If SELinux was in **`permissive`**, run the following command as **`root`** to set it to **`enforcing`** again:

```
# setenforce 1
```

To configure SELinux persistently, modify the **/etc/selinux/config** configuration file.

For more information about SELinux, see the SELinux User's and Administrator's Guide for Red Hat Enterprise Linux 7.

2.7. ADD INITIAL MONITORS

Add the initial monitor(s) and gather the keys.

```
ceph-deploy mon create-initial
```

Once you complete the process, your local directory should have the following keyrings:

- **<cluster-name>.client.admin.keyring**
- **<cluster-name>.bootstrap-osd.keyring**
- **<cluster-name>.bootstrap-mds.keyring**
- **<cluster-name>.bootstrap-rgw.keyring**

2.8. CONNECT MONITOR HOSTS TO CALAMARI

Once you have added the initial monitor(s), you need to connect the monitor hosts to Calamari. From your admin node, execute:

```
ceph-deploy calamari connect --master '<FQDN for the Calamari admin node>'
<ceph-node>[<ceph-node> ...]
```

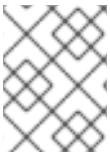
For example, using the exemplary **node1** from above, you would execute:

```
ceph-deploy calamari connect --master '<FQDN for the Calamari admin node>'
node1
```

If you expand your monitor cluster with additional monitors, you will have to connect the hosts that contain them to Calamari, too.

2.9. MAKE YOUR CALAMARI ADMIN NODE A CEPH ADMIN NODE

After you create your initial monitors, you can use the Ceph CLI to check on your cluster. However, you have to specify the monitor and admin keyring each time with the path to the directory holding your configuration, but you can simplify your CLI usage by making the admin node a Ceph admin client.



NOTE

You will also need to install **ceph-common** on the Calamari node. **ceph-deploy install --cli** does this.

```
ceph-deploy install --cli <node-name>
ceph-deploy admin <node-name>
```

For example:

```
ceph-deploy install --cli admin-node
ceph-deploy admin admin-node
```

The **ceph-deploy** utility will copy the **ceph.conf** and **ceph.client.admin.keyring** files to the **/etc/ceph** directory. When **ceph-deploy** is talking to the local admin host (**admin-node**), it must be reachable by its hostname (e.g., **hostname -s**). If necessary, modify **/etc/hosts** to add the name of the admin host. If you do not have an **/etc/ceph** directory, you should install **ceph-common**.

You may then use the Ceph CLI.

Once you have added your new Ceph monitors, Ceph will begin synchronizing the monitors and form a quorum. You can check the quorum status by executing the following as **root**:

```
# ceph quorum_status --format json-pretty
```

NOTE

Your cluster will not achieve an **active + clean** state until you add enough OSDs to facilitate object replicas. This is inclusive of CRUSH failure domains.

2.10. ADJUST CRUSH TUNABLES

Red Hat Ceph Storage CRUSH tunables defaults to **bobtail**, which refers to an older release of Ceph. This setting guarantees that older Ceph clusters are compatible with older Linux kernels. However, if you run a Ceph cluster on Red Hat Enterprise Linux 7, reset CRUSH tunables to **optimal**. As **root**, execute the following:

```
# ceph osd crush tunables optimal
```

See the CRUSH Tunables chapter in the Storage Strategies guides for details on the CRUSH tunables.

2.11. ADD OSDS

Before creating OSDs, consider the following:

- We recommend using the XFS file system, which is the default file system.

WARNING

Use the default XFS file system options that the **ceph-deploy** utility uses to format the OSD disks. Deviating from the default values can cause stability problems with the storage cluster.

For example, setting the directory block size higher than the default value of 4096 bytes can cause memory allocation deadlock errors in the file system. For more details, view the Red Hat Knowledgebase article regarding these errors.

- Red Hat recommends using SSDs for journals. It is common to partition SSDs to serve multiple OSDs. Ensure that the number of SSD partitions does not exceed the SSD's sequential write limits. Also, ensure that SSD partitions are properly aligned, or their write performance will suffer.
- Red Hat recommends to delete the partition table of a Ceph OSD drive by using the **ceph-deploy disk zap** command before executing the **ceph-deploy osd prepare** command:

  ```
  ceph-deploy disk zap <ceph_node>:<disk_device>
  ```

  For example:

  ```
  ceph-deploy disk zap node2:/dev/sdb
  ```

From your administration node, use **ceph-deploy osd prepare** to prepare the OSDs:

```
ceph-deploy osd prepare <ceph_node>:<disk_device> [<ceph_node>:
<disk_device>]
```

For example:

```
ceph-deploy osd prepare node2:/dev/sdb
```

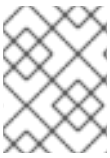
The **prepare** command creates two partitions on a disk device; one partition is for OSD data, and the other is for the journal.

Once you prepare OSDs, activate the OSDs:

```
ceph-deploy osd activate <ceph_node>:<data_partition>
```

For example:

```
ceph-deploy osd activate node2:/dev/sdb1
```



NOTE

In the **ceph-deploy osd activate** command, specify a particular disk partition, for example **/dev/sdb1**.

It is also possible to use a disk device that is wholly formatted without a partition table. In that case, a partition on an additional disk must be used to serve as the journal store:

```
ceph-deploy osd activate <ceph_node>:<disk_device>:<data_partition>
```

In the following example, **sdd** is a spinning hard drive that Ceph uses entirely for OSD data. **ssdb1** is a partition of an SSD drive, which Ceph uses to store the journal for the OSD:

```
ceph-deploy osd activate node{2,3,4}:sdd:ssdb1
```

To achieve the **active + clean** state, you must add as many OSDs as the **osd pool default size = <n>** parameter specifies in the Ceph configuration file.

For information on creating encrypted OSD nodes, see the Encrypted OSDs subsection in the Adding OSDs by Using ceph-deploy section in the Administration Guide for Red Hat Ceph Storage 2.

2.12. CONNECT OSD HOSTS TO CALAMARI

Once you have added the initial OSDs, you need to connect the OSD hosts to Calamari.

```
ceph-deploy calamari connect --master '<FQDN for the Calamari admin node>'
<ceph-node>[<ceph-node> ...]
```

For example, using the exemplary **node2**, **node3** and **node4** from above, you would execute:

```
ceph-deploy calamari connect --master '<FQDN for the Calamari admin node>'
node2 node3 node4
```

As you expand your cluster with additional OSD hosts, you will have to connect the hosts that contain them to Calamari, too.

2.13. CREATE A CRUSH HIERARCHY

You can run a Ceph cluster with a flat node-level hierarchy (default). This is NOT RECOMMENDED. We recommend adding named buckets of various types to your default CRUSH hierarchy. This will allow you to establish a larger-grained failure domain, usually consisting of racks, rows, rooms and data centers.

```
ceph osd crush add-bucket <bucket-name> <bucket-type>
```

For example:

```
ceph osd crush add-bucket dc1 datacenter
ceph osd crush add-bucket room1 room
ceph osd crush add-bucket row1 row
ceph osd crush add-bucket rack1 rack
ceph osd crush add-bucket rack2 rack
ceph osd crush add-bucket rack3 rack
```

Then, place the buckets into a hierarchy:

```
ceph osd crush move dc1 root=default
ceph osd crush move room1 datacenter=dc1
ceph osd crush move row1 room=room1
ceph osd crush move rack1 row=row1
ceph osd crush move node2 rack=rack1
```

2.14. ADD OSD HOSTS/CHASSIS TO THE CRUSH HIERARCHY

Once you have added OSDs and created a CRUSH hierarchy, add the OSD hosts/chassis to the CRUSH hierarchy so that CRUSH can distribute objects across failure domains. For example:

```
ceph osd crush set osd.0 1.0 root=default datacenter=dc1 room=room1
row=row1 rack=rack1 host=node2
ceph osd crush set osd.1 1.0 root=default datacenter=dc1 room=room1
```

```
row=row1 rack=rack2 host=node3
ceph osd crush set osd.2 1.0 root=default datacenter=dc1 room=room1
row=row1 rack=rack3 host=node4
```

The foregoing example uses three different racks for the exemplary hosts (assuming that is how they are physically configured). Since the exemplary Ceph configuration file specified "rack" as the largest failure domain by setting **`osd_crush_chooseleaf_type = 3`**, CRUSH can write each object replica to an OSD residing in a different rack. Assuming **`osd_pool_default_min_size = 2`**, this means (assuming sufficient storage capacity) that the Ceph cluster can continue operating if an entire rack were to fail (e.g., failure of a power distribution unit or rack router).

2.15. CHECK CRUSH HIERARCHY

Check your work to ensure that the CRUSH hierarchy is accurate.

```
ceph osd tree
```

If you are not satisfied with the results of your CRUSH hierarchy, you may move any component of your hierarchy with the **`move`** command.

```
ceph osd crush move <bucket-to-move> <bucket-type>=<parent-bucket>
```

If you want to remove a bucket (node) or OSD (leaf) from the CRUSH hierarchy, use the **`remove`** command:

```
ceph osd crush remove <bucket-name>
```

2.16. CHECK CLUSTER HEALTH

To ensure that the OSDs in your cluster are peering properly, execute:

```
ceph health
```

You may also check on the health of your cluster using the Calamari dashboard.

2.17. LIST AND CREATE A POOL

You can manage pools using Calamari, or using the Ceph command line. Verify that you have pools for writing and reading data:

```
ceph osd lspools
```

You can bind to any of the pools listed using the **`admin`** user and **`client.admin`** key. To create a pool, use the following syntax:

```
ceph osd pool create <pool-name> <pg-num> [<pgp-num>] [replicated] [crush-
ruleset-name]
```

For example:

```
ceph osd pool create mypool 512 512 replicated replicated_ruleset
```

NOTE

To find the rule set names available, execute **`ceph osd crush rule list`**. To calculate the **`pg-num`** and **`pgp-num`** see Ceph Placement Groups (PGs) per Pool Calculator.

2.18. STORING AND RETRIEVING OBJECT DATA

To perform storage operations with Ceph Storage Cluster, all Ceph clients regardless of type must:

1. Connect to the cluster.
2. Create an I/O contest to a pool.
3. Set an object name.
4. Execute a read or write operation for the object.

The Ceph Client retrieves the latest cluster map and the CRUSH algorithm calculates how to map the object to a placement-group, and then calculates how to assign the placement group to a Ceph OSD Daemon dynamically. Client types such as Ceph Block Device and the Ceph Object Gateway perform the last two steps transparently.

To find the object location, all you need is the object name and the pool name. For example:

```
ceph osd map <poolname> <object-name>
```

NOTE

The **`rados`** CLI tool in the following example is for Ceph administrators only.

Exercise: Locate an Object

As an exercise, lets create an object. Specify an object name, a path to a test file containing some object data and a pool name using the **`rados put`** command on the command line. For example:

```
echo <Test-data> > testfile.txt
rados put <object-name> <file-path> --pool=<pool-name>
rados put test-object-1 testfile.txt --pool=data
```

To verify that the Ceph Storage Cluster stored the object, execute the following:

```
rados -p data ls
```

Now, identify the object location:

```
ceph osd map <pool-name> <object-name>
ceph osd map data test-object-1
```

Ceph should output the object's location. For example:

```
osdmap e537 pool 'data' (0) object 'test-object-1' -> pg 0.d1743484 (0.4)
-> up [1,0] acting [1,0]
```

To remove the test object, simply delete it using the **`rados rm`** command. For example:

```
rados rm test-object-1 --pool=data
```

As the cluster size changes, the object location may change dynamically. One benefit of Ceph's dynamic rebalancing is that Ceph relieves you from having to perform the migration manually.

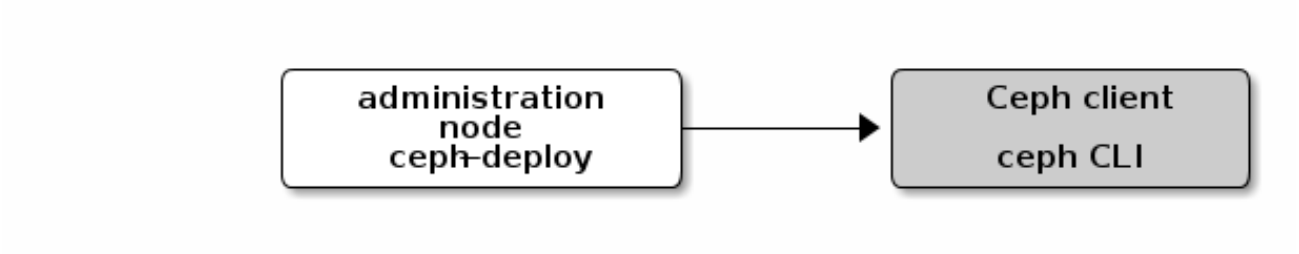
CHAPTER 3. CLIENT QUICK START

Red Hat Ceph Storage supports three types of Ceph clients:

- **Ceph CLI**: The **ceph** command-line interface (CLI) enables Ceph administrators to execute Ceph administrative commands such as creating a CRUSH hierarchy, monitoring cluster health, or managing users from the command line.
- **Ceph Block Device:** Red Hat Ceph Storage supports mounting a thin-provisioned, re-sizable block device. While the most popular use case for Ceph Block Device is to use its **librbd** library with QEMU and **libvirt** to serve as a back end for cloud platforms like the Red Hat Open Stack Platform, we also support a kernel block device (RHEL 7.1 x86_64 and later releases only).
- **Ceph Object Gateway:** Red Hat Ceph Storage supports a Ceph Object Gateway with its own user management and Swift- and S3-compliant APIs.

To use Ceph clients, you must first have a Ceph Storage Cluster running, preferably in the **active + clean** state.

Ceph clients typically run on nodes separate from the Ceph Storage Cluster. You can use **ceph-deploy** on your Calamari administration to configure a Ceph client node.



3.1. EXECUTE THE PRE-INSTALLATION PROCEDURE

For streamlined installation and deployment, execute the pre-installation procedures on your Ceph client node. Specifically, disable **requiretty**, set SELinux to **permissive** mode (with Red Hat Ceph Storage 1.3.2 or later, SELinux can run in **enforcing** mode, see the SELinux and Install ceph-selinux sections for more details), and set up a Ceph Deploy user with password-less **sudo** (see the Create a Ceph Deploy User and Enable Password-less sudo sections). For Ceph Object Gateways, open the ports that Civetweb uses in production (by default port **80** and port **7480**).

3.2. ENABLE CEPH CLIENT REPOSITORY

Red Hat includes Ceph Storage clients in the **rhel-7-server-rhceph-1.3-tools-rpms** repository. To ensure you are using the same version of the Ceph client as your storage cluster, execute the following to enable the repository:

```
# subscription-manager repos --enable=rhel-7-server-rhceph-1.3-tools-rpms
```

3.3. INSTALL THE CEPH COMMON PACKAGE

The Ceph Common packages provides the Ceph CLI tools, the Ceph Block Device and the Ceph Object Store daemon.

To install **ceph-common** CLI tools, go to the working directory of Calamari administration server and execute:

```
ceph-deploy install --cli <node-name>
```

NOTE

Using **ceph-deploy** requires you to execute the Pre Installation procedure first.

The **ceph** CLI tools are intended for administrators. To make your Ceph client node an administrator node, execute the following from the working directory of your administration server.

```
ceph-deploy admin <node-name>
```

The CLI tools include:

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

3.4. BLOCK DEVICE QUICK START

The following quick start describes how to mount a thin-provisioned, resizable block device for RHEL 7.1 x86_64 and later releases only. You must install **ceph-common** first before using this procedure.

Execute the following procedures on a separate physical node (or within a VM) from the Ceph monitor and OSD nodes. Running Linux kernel clients and kernel server daemons on the same node can lead to kernel deadlocks.

1. Create a user for your block device. This step requires the Ceph CLI interface with administrative privileges. To create a user, execute **ceph auth get-or-create** and output the result to a keyring file.

   ```
   # ceph auth get-or-create USERTYPE.USERID {daemon} \
    'allow <r|w|x|*|...> [pool={pool-name}]' \
    -o /etc/ceph/rbd.keyring
   ```

   A block device user should have **rwx** permissions on OSDs, because block devices use classes and therefore require execute **x** permissions. The following example limits the **client.rbd** user to the default **rbd** pool. For example, on the **ceph-client** node, execute:

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

   See the Red Hat Ceph Storage Administration Guide for additional details on user management.

2. On the **ceph-client** node, create a block device image.

```
# rbd create foo --size 4096 --pool rbd \
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

3. On the **ceph-client** node, map the image to a block device.

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

IMPORTANT

Kernel block devices currently only support the legacy **straw** bucket algorithm in your CRUSH map. If you have set your CRUSH tunables to **optimal**, you may have to set them to **legacy** or an earlier major release; otherwise, you will not be able to map the image. Alternatively, refer to the Ceph Storage Strategies guide and the section on editing a CRUSH map to ensure that you are not using **straw2** (replace **straw2** with **straw**).

4. Use the block device by creating a file system on the **ceph-client** node.

```
# mkfs.ext4 -m0 /dev/rbd/rbd/foo
```

This may take a few moments.

5. Mount the file system on the **ceph-client** node.

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

See the Red Hat Ceph Storage Block Device guide for additional details.

3.5. OBJECT GATEWAY QUICK START

Red Hat Ceph Storage v1.3 dramatically simplifies installing and configuring a Ceph Object Gateway. The Gateway daemon embeds Civetweb, so you do not have to install a web server or configure FastCGI. Additionally, **ceph-deploy** can install the gateway package, generate a key, configure a data directory and create a gateway instance for you.

TIP

Civetweb uses port **7480** by default. You must either open port **7480**, or set the port to a preferred port (typically port **80**) in your Ceph configuration file.

To start a Ceph Object Gateway, follow this procedure:

1. Execute the pre-installation steps on your **client-node**. If you intend to use Civetweb's default port **7480**, you must open it using **firewall-cmd**.

2. Enable the **rhel-7-server-rhceph-1.3-tools-rpms** repository on the **client-node** node, if you haven't done so already.

```
# subscription-manager repos --enable=rhel-7-server-rhceph-1.3-
tools-rpms
```

3. From the working directory of your administration server, install the Ceph Object Gateway package on the **client-node** node. For example:

```
ceph-deploy install --rgw <client-node> [<client-node> ...]
```

4. From the working directory of your administration server, create an instance of the Ceph Object Gateway on the **client-node**. For example:

```
ceph-deploy rgw create <client-node>
```

Once the gateway is running, you should be able to access it on port **7480**. (for example, **http://client-node:7480**).

5. To change the default port (for example, to port **80**), modify your Ceph configuration file. Add a section entitled **[client.rgw.<client-node>]**, replacing **<client-node>** with the short node name of your Ceph client node (**$ hostname -s**). For example, if your node name is **client-node**, add a section like this after the **[global]** section:

```
[client.rgw.client-node]
rgw_frontends = "civetweb port=80"
```

NOTE

Ensure that you leave no whitespace between **port=<port-number>** in the **rgw_frontends** key/value pair.

6. To make the new port setting take effect, restart the Ceph Object Gateway.

```
# systemctl restart ceph-radosgw.service
```

Finally, check to ensure that the port you selected is open on the node's firewall (for example, port **80**). If it is not open, add the port and reload the firewall configuration. For example:

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

You should be able to make an unauthenticated request, and receive a response. For example, a request with no parameters like this:

```
http://<client-node>:80
```

Should result in a response like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
 <Owner>
  <ID>anonymous</ID>
```

```
   <DisplayName></DisplayName>
  </Owner>
  <Buckets>
  </Buckets>
</ListAllMyBucketsResult>
```

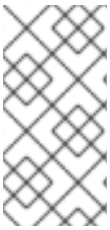
See the Ceph Object Storage Guide for RHEL **x86_64** for additional administration and API details.

CHAPTER 4. UPGRADING THE STORAGE CLUSTER

To keep your administration server and your Ceph Storage cluster running optimally, upgrade them when Red Hat provides bug fixes or delivers major updates.

There is only one supported upgrade path to upgrade your cluster to the latest 1.3 version:

- Upgrading 1.3.2 to 1.3.3



NOTE

For all the upgrade paths, SELinux needs to be in **`permissive`** or **`disabled`** mode in your cluster. With **Red Hat Ceph Storage 1.3.2** or later, SELinux no longer needs to be in **`permissive`** or **`disabled`** mode and can be run in its default mode, that is **`enforcing`**. See Section 2.6, “Install **`ceph-selinux`**” for more information.

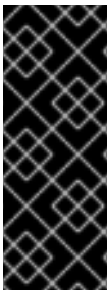
4.1. UPGRADING 1.3.2 TO 1.3.3

There are two ways to upgrade Red Hat Ceph Storage 1.3.2 to 1.3.3:

- CDN or online-based installations
- ISO-based installations

For upgrading Ceph with the CDN or an ISO-based installation method, Red Hat recommends upgrading in the following order:

- Administration Node
- Monitor Nodes
- OSD Nodes
- Object Gateway Nodes



IMPORTANT

Due to changes in encoding of the OSD map in the **`ceph`** package version 0.94.7, upgrading Monitor nodes to Red Hat Ceph Storage 1.3.3 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 1.3.3 from previous versions.

4.1.1. Administration Node

Using CDN

To upgrade the administration node, reinstall the **`ceph-deploy`** package, upgrade the Calamari server, reinitialize the Calamari and Salt services, and upgrade Ceph. Finally, upgrade the administration node.

To do so, run the following commands as **`root`**:

```
# yum install ceph-deploy
# yum install calamari-server calamari-clients
```

```
# calamari-ctl initialize
# salt '*' state.highstate
# ceph-deploy install --cli <administration-node>
# yum update
```

NOTE

The repositories are the same as in Red Hat Ceph Storage 1.3. These repositories include all the updated packages.

Using an ISO

For ISO-based upgrades, remove **Calamari**, **Installer**, and **Tools** repositories from the **/etc/yum.repos.d/** directory, remove the **cephdeploy.conf** file from the Ceph working directory, for example **~/ceph-config/**, remove the **.cephdeploy.conf** file from the home directory, download and mount the latest Red Hat Ceph Storage 1.3 ISO image, run the **ice_setup** utility, reinitialize the Calamari service, and upgrade Ceph. Finally, upgrade the administration node.

1. Remove the Ceph related repositories from **/etc/yum.repos.d/**. Execute the following commands as **root**:

   ```
   # cd /etc/yum.repos.d
   # rm -rf Calamari.repo Installer.repo Tools.repo
   ```

2. Remove the existing **cephdeploy.conf** file from the Ceph working directory:

   Syntax

   ```
   # rm -rf <directory>/cephdeploy.conf
   ```

   Example

   ```
   # rm -rf /home/example/ceph/cephdeploy.conf
   ```

3. Remove the existing **.cephdeploy.conf** file from the home directory:

   Syntax

   ```
   $ rm -rf <directory>/.cephdeploy.conf
   ```

   Example

   ```
   # rm -rf /home/example/ceph/.cephdeploy.conf
   ```

To install the **ceph-deploy** utility and Calamari using an ISO image, visit the Software & Download Center on the Red Hat Customer Portal to download the latest Red Hat Ceph Storage installation ISO image file.

1. Mount the ISO by using the following command as **root**:

   ```
   # mount /<path_to_iso>/rhceph-1.3.3-rhel-7-x86_64-dvd.iso /mnt
   ```

2. Install the setup program by running the following command as **root**:

```
# yum install /mnt/Installer/ice_setup-*.rpm
```

3. Change to the Ceph working directory. For example:

```
# cd /home/example/ceph
```

4. Run the **ice_setup** utility as **root**:

```
# ice_setup -d /mnt
```

The **ice_setup** program installs the upgraded version of the **ceph-deploy** utility and the Calamari server, creates new local repositories, and a new **.cephdeploy.conf** file.

5. Restart Calamari and set Salt to **state.highstate** by running the following commands as **root**:

```
# calamari-ctl initialize
# salt '*' state.highstate
```

6. Upgrade Ceph:

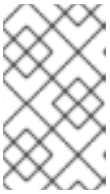
```
# ceph-deploy install --cli <administration-node>
```

7. Finally, update the administration node:

```
# yum update
```

8. Optionally, enable the Tools repository:

```
# subscription-manager repos --enable=rhel-7-server-rhceph-1.3-
tools-rpms
```



NOTE

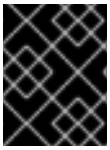
It is mandatory to enable the **rhel-7-server-rhceph-1.3-tools-rpms** repository to obtain the **ceph-common** package. The Red Hat Ceph Storage ISO image does not include the **tools** repository packages.

You can also enable the administration node to receive online updates and publish them to the rest of the cluster with the **ice_setup update** command. To do so, execute the following commands as **root**:

```
# subscription-manager repos --enable=rhel-7-server-rpms --enable=rhel-7-
server-rhceph-1.3-calamari-rpms --enable=rhel-7-server-rhceph-1.3-
installer-rpms --enable=rhel-7-server-rhceph-1.3-mon-rpms --enable=rhel-7-
server-rhceph-1.3-osd-rpms --enable=rhel-7-server-rhceph-1.3-tools-rpms
# yum update
```

4.1.2. Monitor Nodes

To upgrade a Monitor node, reinstall the Monitor daemon from the administration node and reconnect the Monitor node to Calamari. Finally, upgrade and restart the Monitor daemon.



IMPORTANT

Only upgrade one Monitor node at a time, and allow the Monitor to come up and in, rejoining the Monitor quorum, before proceeding to upgrade the next Monitor.

Using CDN

1. From the administration node, execute:

```
# ceph-deploy install --mon <monitor-node>
# ceph-deploy calamari connect --master '<FQDN for the Calamari
administration node>' <monitor-node>
```

2. Upgrade and restart the Monitor daemon. From the Monitor node, execute the following commands as **root**:

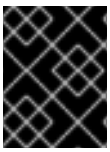
```
# yum update
# /etc/init.d/ceph [options] restart mon.[id]
```

NOTE

The repositories are the same as in Red Hat Ceph Storage 1.3. These repositories include all the updated packages.

Using an ISO

To upgrade a Monitor node, log in to the node and stop the Monitor daemon. Remove the **ceph-mon** repository from the **/etc/yum.repos.d/** directory. Then, reinstall the Ceph Monitor daemon from the administration node and reconnect the Monitor node with Calamari. Finally, update the Monitor node and restart the Monitor daemon.



IMPORTANT

Only upgrade one Monitor node at a time, and allow the Monitor to come up and in, rejoining the Monitor quorum, before proceeding to upgrade the next Monitor.

1. On the Monitor node, execute the following commands as **root**:

```
# /etc/init.d/ceph [options] stop mon.[id]
# rm /etc/yum.repos.d/ceph-mon.repo
```

1. From the administration node, execute:

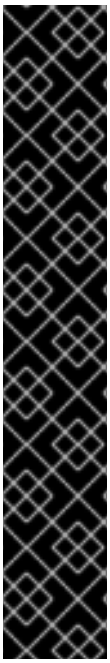
```
# ceph-deploy install --repo --release=ceph-mon <monitor-node>
# ceph-deploy install --mon <monitor-node>
# ceph-deploy calamari connect --master '<FQDN for the Calamari
administration node>' <monitor-node>
```

1. From the Monitor node, update to the latest packages and start the Ceph Monitor daemon. Run the following commands as **root**:

```
# yum update
# /etc/init.d/ceph [options] restart mon.[id]
```

4.1.3. OSD Nodes

To upgrade a Ceph OSD node, reinstall the OSD daemon from the administration node, and reconnect OSD node to Calamari. Finally, upgrade the OSD node and restart the OSDs.



IMPORTANT

Only upgrade one OSD node at a time, and preferably within a CRUSH hierarchy. Allow the OSDs to come up and in, and the cluster achieving the **active + clean** state, before proceeding to upgrade the next OSD node.

Before starting the upgrade of the OSD nodes, set the **noout** and the **norebalance** flags:

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

Once all the OSD nodes are upgraded in the storage cluster, unset the the **noout** and the **norebalance** flags:

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

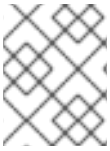
Using CDN

1. From the administration node, execute:

```
# ceph-deploy install --osd <osd-node>
# ceph-deploy calamari connect --master '<FQDN for the Calamari
administration node>' <osd-node>
```

2. Finally, update the OSD node and restart the OSD daemon by running the following commands as **root**:

```
# yum update
# /etc/init.d/ceph [options] restart
```



NOTE

The repositories are the same as in Red Hat Ceph Storage 1.3. These repositories include all the updated packages.

Using an ISO

1. On the OSD node, execute the following commands as **root**:

```
# /etc/init.d/ceph [options] stop
# rm /etc/yum.repos.d/ceph-osd.repo
```

2. From the administration node, execute:

```
# ceph-deploy install --repo --release=ceph-osd <osd-node>
# ceph-deploy install --osd <osd-node>
# ceph-deploy calamari connect --master '<FQDN for the Calamari
administration node>' <osd-node>
```

3. From the OSD node, update to the latest packages and start the Ceph OSD daemons by running the following commands as **root**:

```
# yum update
# /etc/init.d/ceph [options] start
```

4.1.4. Object Gateway Nodes

To upgrade a Ceph Object Gateway node, reinstall the Ceph Object Gateway daemon from the administration node, upgrade the Ceph Object Gateway node, and restart the gateway.

Using CDN

1. From the administration node, execute:

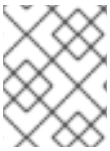
```
# ceph-deploy install --rgw <rgw-node>
```

1. For federated deployments, from the Ceph Object Gateway node, execute the following command as **root**:

```
# yum install radosgw-agent
```

1. Upgrade the Ceph Object Gateway node and restart the gateway by running the following commands as **root**:

```
# yum update
# systemctl restart ceph-radosgw
```



NOTE

The repositories are the same as in Red Hat Ceph Storage 1.3. These repositories include all the updated packages.

Using an ISO

The Ceph Object Gateway is not shipped with ISO installations. To upgrade the Ceph Object Gateway, the CDN-based installation must be used.