Red Hat Ceph Storage 1.3

Installation Guide for Ubuntu

Installing Calamari and Red Hat Ceph Storage on Ubuntu

Last Updated: 2018-03-20

Red Hat Ceph Storage 1.3 Installation Guide for Ubuntu

Installing Calamari and Red Hat Ceph Storage on Ubuntu

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 Ceph Storage Cluster on Ubuntu 14.04 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 Ubuntu `Trusty` 14.04.

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

IMPORTANT

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

1.1.2. Kernel

Red Hat Ceph Storage v1.3 is supported on Ubuntu 14.04 configurations including the original 14.04 LTS kernel (3.13). Configurations including the latest HWE kernel (3.19) are also supported. We recommend customers default their choice to the LTS kernel where the HWE is not required by their hardware in light of its longer support lifecycle.

1.1.3. 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.

1.1.4. 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.

1.1.5. Firewall

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`**. You **MUST** open ports **`80`**, **`2003`**, and **`4505-4506`** on your Calamari node.

```
sudo iptables -I INPUT 1 -i <iface> -p tcp -s <ip-address>/<netmask> --
dport 80 -j ACCEPT
sudo iptables -I INPUT 1 -i <iface> -p tcp -s <ip-address>/<netmask> --
dport 2003 -j ACCEPT
sudo iptables -I INPUT 1 -i <iface> -m multiport -p tcp -s <ip-
address>/<netmask> --dports 4505:4506 -j ACCEPT
```

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

```
sudo iptables -I INPUT 1 -i <iface> -p tcp -s <ip-address>/<netmask> --
dport 6789 -j ACCEPT
```

Finally, you **MUST** also open ports for OSD traffic (e.g., **`6800-7300`**). **Each OSD on each Ceph node** needs three 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). The OSDs will bind to the next available port if they are restarted in certain scenarios, so open the entire port range 6800 through 7300 for completeness.

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

Once you have finished configuring **`iptables`**, ensure that you make the changes persistent on each node so that they will be in effect when your nodes reboot.

Execute:

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

A terminal UI will open up. Select **`yes`** for the prompts to save current **`IPv4`** iptables rules to **`/etc/iptables/rules.v4`** and current **`IPv6`** iptables rules to **`/etc/iptables/rules.v6`**.

The **`IPv4`** iptables rules that you set in the earlier steps will be loaded in **`/etc/iptables/rules.v4`** and will be persistent across reboots.

If you add a new **IPv4** iptables rule after installing **iptables-persistent** you will have to add it to the rule file. In such case, execute the following as a **root** user:

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

1.1.6. Network Time Protocol

You MUST install Network Time Protocol (NTP) on all Ceph monitor hosts and ensure that monitor hosts 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

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

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

   ```
   sudo service ntp start
   sudo service ntp status
   ```

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

   ```
   ntpq -p
   ```

1.1.7. Install SSH Server

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

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

   ```
   sudo apt-get install openssh-server
   ```

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

1.1.8. 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>
   ```

   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>
   ```

1.1.9. 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
```

1.1.10. 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.

1.1.11. Adjust PID Count

Hosts with high numbers of OSDs (e.g., > 20) may spawn a lot of threads, especially during recovery and re-balancing. Ubuntu 14.04 kernels (3.13 and 3.16) default to a relatively small maximum number of threads (e.g., **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
```

1.1.12. Enable Ceph Repositories

Starting from v1.3.1, Red Hat Ceph Storage supports two installation methods for Ubuntu Trusty:

- **Online Repositories**: For Ceph Storage clusters with Ceph nodes that can connect directly to the internet, you can use online repositories for **`Calamari`**, **`Installer`**, **`Monitors`**, **`OSDs`** and **`RGW`** from https://rhcs.download.redhat.com/ubuntu. You will need your **`Customer Name`** and **`Customer Password`** received from https://rhcs.download.redhat.com to be able to use the repos.

  IMPORTANT

  Please contact your account manager to obtain credentials for **`https://rhcs.download.redhat.com`**.

- **Local Repository:** For Ceph Storage clusters where security measures preclude nodes from accessing the internet, 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 installation via online repositories, configure the Installer repository on your administration node in order to install **`ceph-deploy`**, then use **`ceph-deploy`** to configure all other RHCS repositories.

1. For your administration node, set the Installer (**`ceph-deploy`**) repository then use **`ceph-deploy`** to enable the Calamari and Tools repositories.

   ```
   sudo bash -c 'umask 0077; echo deb
   https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu/
   1.3-updates/Installer $(lsb_release -sc) main | tee
   /etc/apt/sources.list.d/Installer.list'
   sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt
   | apt-key add -'
   sudo apt-get update
   sudo apt-get install ceph-deploy
   ceph-deploy repo --repo-url
   'https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu
   /1.3-updates/Calamari' Calamari `hostname -f`
   ceph-deploy repo --repo-url
   'https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu
   /1.3-updates/Tools' Tools `hostname -f`
   sudo apt-get update
   ```

2. For Ceph monitor nodes, install the MON repository by running the following from the Admin node:

```
ceph-deploy repo --repo-url
'https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu
/1.3-updates/MON' --gpg-url
https://www.redhat.com/security/fd431d51.txt ceph-mon <MONHOST1>
[<MONHOST2> [<MONHOST3> ...]]
```

3. Update monitor nodes:

   ```
   sudo apt-get update
   ```

4. For Ceph OSD nodes, install the OSD repository by running the following from the Admin node:

   ```
   ceph-deploy repo --repo-url
   'https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu
   /1.3-updates/OSD' --gpg-url
   https://www.redhat.com/security/fd431d51.txt ceph-osd <OSDHOST1>
   [<OSDHOST2> [<OSDHOST3> ...]]
   ```

5. Update OSD nodes:

   ```
   sudo apt-get update
   ```

6. For gateway nodes, install the Tools repository by running the following from the Admin node:

   ```
   ceph-deploy repo --repo-url
   'https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu
   /1.3-updates/Tools' --gpg-url
   https://www.redhat.com/security/fd431d51.txt Tools <RGWNODE1>
   [RGWNODE2 [RGWNODE3 ...]]
   ```

7. Update gateway nodes:

   ```
   sudo apt-get update
   ```

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:

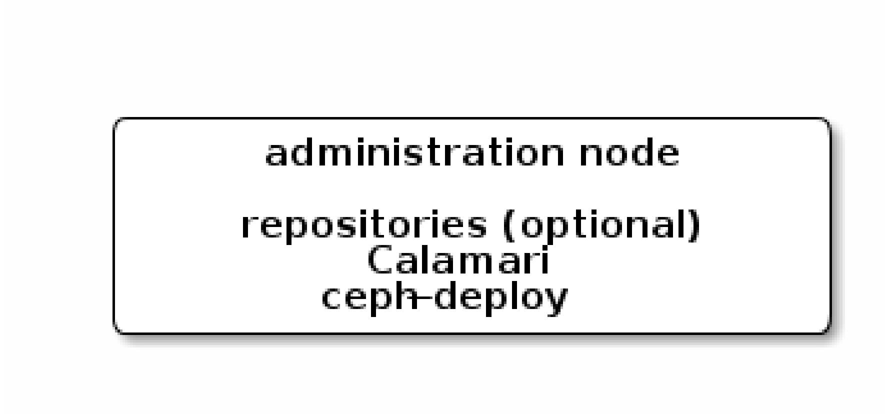
- at least 4 GB of RAM,
- a dual core CPU on AMD64 or Intel 64 architecture,
- enough network throughput to handle communication with Ceph hosts.

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

If you have correctly set the online repositories for **`Calamari`** and **`Installer`** in section Set Online Ceph Repositories, execute the following to install Calamari:

```
sudo apt-get install 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, execute following steps:

1. Visit the Red Hat Ceph Storage for Ubuntu page on the Customer Portal to obtain the Red Hat Ceph Storage installation ISO image files.

2. Using **`sudo`**, mount the downloaded ISO image to the **`/mnt/`** directory, for example:

   ```
   $ sudo mount <path_to_iso>/rhceph-1.3.2-ubuntu-x86_64-dvd.iso /mnt
   ```

3. Using **sudo**, install the **ice_setup** program:

   ```
   $ sudo dpkg -i /mnt/ice-setup_*.deb
   ```

   NOTE

   If you receive an error that the **python-pkg-resources** package is missing, run the **sudo apt-get -f install** command to install the missing **python-pkg-resources** package.

4. Navigate to the working directory that you created in the Create a Working Directory section:

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

5. Within this directory, run **ice_setup** using **sudo**:

   ```
   $ sudo ice_setup -d /mnt
   ```

   Follow the instructions in the interactive shell.

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

   - creates a local repository for the **ceph-deploy** and **calamari** packages
   - installs the Calamari server packages on the administration node
   - installs the **ceph-deploy** package on the administration node
   - creates the **/opt/ICE/** and **/opt/calamari/** directories
   - writes the **.cephdeploy.conf** file to the **/root/** directory and to the current working directory, for example, **~/ceph-config**

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
```

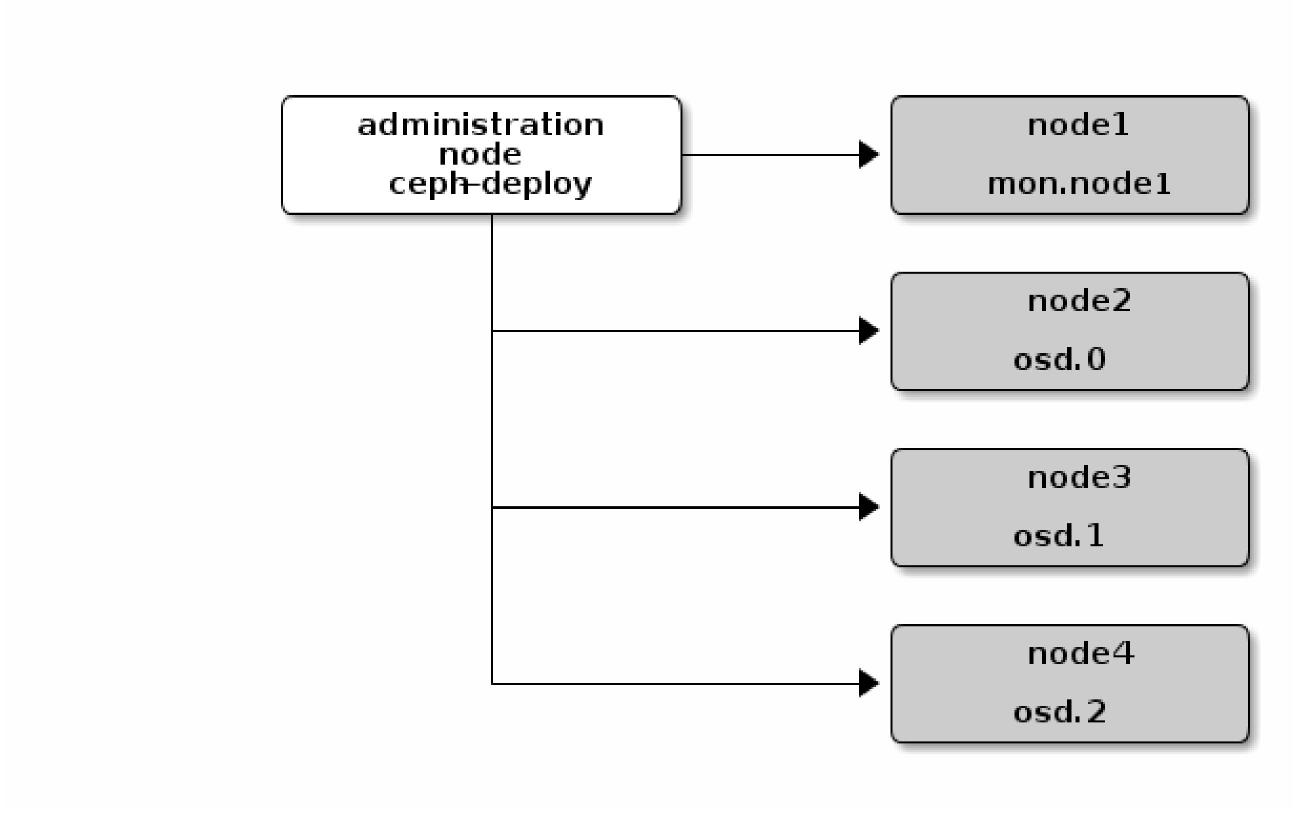
At this point, you should be able to access the Calamari web server using a web browser. Proceed to the Storage Cluster Quick Start.

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.

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. 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**

2.7. 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.

2.8. 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.

2.9. 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 Ubuntu 14.04 or 16.04, 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.

2.10. 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.

2.11. 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.

2.12. 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
```

2.13. 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).

2.14. 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>
```

2.15. 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.

2.16. 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.

2.17. 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. 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

If your cluster nodes run Ubuntu Precise 12.04, you must upgrade your operating systems to Ubuntu Trusty 14.04. Red Hat Ceph Storage 1.3 is only supported on Ubuntu Trusty. Please see the separate **`Upgrade Ceph Cluster on Ubuntu Precise to Ubuntu Trusty`** document if your cluster is running on Ubuntu Precise.

3.1. UPGRADING 1.3.X 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 an online 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.

3.1.1. Administration Node

Using the Online Repositories

To upgrade admin node, remove **`Calamari`**, **`Installer`**, and **`Tools`** repositories under **`/etc/apt/sources.list.d/`**, remove **`cephdeploy.conf`** from the working directory, for example **`/home/example/ceph/`**, remove **`.cephdeploy.conf`** from the home directory, set Installer (**`ceph-deploy`**) online repository, upgrade **`ceph-deploy`**, enable Calamari and Tools online repositories, upgrade **`calamari-server`**, **`calamari-clients`**, re-initialize Calamari/Salt and upgrade Ceph.

1. Remove existing Ceph repositories:

```
$ cd /etc/apt/sources.list.d/
$ sudo rm -rf Calamari.list Installer.list Tools.list
```

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

   Syntax

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

   Example

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

4. Set the Installer (**ceph-deploy**) repository then use **ceph-deploy** to enable the Calamari and Tools repositories.:

   ```
   $ sudo bash -c 'umask 0077; echo deb
   https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu/
   1.3-updates/Installer $(lsb_release -sc) main | tee
   /etc/apt/sources.list.d/Installer.list'
   $ sudo bash -c 'wget -O -
   https://www.redhat.com/security/fd431d51.txt | apt-key add -'
   $ sudo apt-get update
   $ sudo apt-get install ceph-deploy
   $ ceph-deploy repo --repo-url
   'https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu
   /1.3-updates/Calamari' Calamari `hostname -f`
   $ ceph-deploy repo --repo-url
   'https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu
   /1.3-updates/Tools' Tools `hostname -f`
   $ sudo apt-get update
   ```

5. Upgrade Calamari:

   ```
   $ sudo apt-get install calamari-server calamari-clients
   ```

6. Re-initialize Calamari:

   ```
   $ sudo calamari-ctl initialize
   ```

7. Update existing cluster nodes that report to Calamari:

   -

```
$ sudo salt '*' state.highstate
```

8. Upgrade Ceph:

```
$ ceph-deploy install --no-adjust-repos --cli  <admin-node>
$ sudo apt-get upgrade
$ sudo restart ceph-all
```

Using an ISO

To upgrade admin node, remove **`Calamari`**, **`Installer`**, and **`Tools`** repositories under **`/etc/apt/sources.list.d/`**, remove **`cephdeploy.conf`** from the working directory, for example **`ceph-config`**, remove **`.cephdeploy.conf`** from the home directory, download and mount the latest Ceph ISO, run **`ice_setup`**, re-initialize Calamari and upgrade Ceph.

> **IMPORTANT**
>
> To support upgrading the other Ceph daemons, you must upgrade the Administration node first.

1. Remove existing Ceph repositories:

```
$ cd /etc/apt/sources.list.d/
$ sudo rm -rf Calamari.list Installer.list Tools.list
```

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

Syntax

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

Example

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

4. Visit the Red Hat Customer Portal to obtain the Red Hat Ceph Storage ISO image file.

5. Download **`rhceph-1.3.3-ubuntu-x86_64-dvd.iso`** file.

6. Using **sudo**, mount the image:

```
$ sudo mount /<path_to_iso>/rhceph-1.3.3-ubuntu-x86_64-dvd.iso /mnt
```

7. Using **sudo**, install the setup program:

   ```
   $ sudo dpkg -i /mnt/ice-setup_*.deb
   ```

   NOTE

   if you receive an error about missing **python-pkg-resources**, run **sudo apt-get -f install** to install the missing **python-pkg-resources** dependency.

8. Navigate to the working directory:

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

9. Using **sudo**, run the setup script in the working directory:

   ```
   $ sudo ice_setup -d /mnt
   ```

   The **ice_setup** program will install upgraded version of **ceph-deploy**, **calamari-server**, **calamari-clients**, create new local repositories and a **.cephdeploy.conf** file.

10. Initialize Calamari and update existing cluster nodes that report to Calamari:

    ```
    $ sudo calamari-ctl initialize
    $ sudo salt '*' state.highstate
    ```

11. Upgrade Ceph:

    ```
    $ ceph-deploy install --no-adjust-repos --cli <admin-node>
    $ sudo apt-get upgrade
    $ sudo restart ceph-all
    ```

3.1.2. Monitor Nodes

To upgrade a Monitor node, log in to the node, remove **ceph-mon** repository under **/etc/apt/sources.list.d/**, install online repository for Monitor from the admin node, re-install Ceph and reconnect Monitor node to Calamari. Finally, upgrade and restart the Ceph 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.

Online Repository

1. Remove existing Ceph repositories in Monitor node:

   ```
   $ cd /etc/apt/sources.list.d/
   $ sudo rm -rf ceph-mon.list
   ```

2. Set online Monitor repository in Monitor node from admin node:

```
$ ceph-deploy repo --repo-url
'https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu
/1.3-updates/MON' --gpg-url
https://www.redhat.com/security/fd431d51.txt ceph-mon <monitor-node>
```

3. Reinstall Ceph in Monitor node from the admin node:

```
$ ceph-deploy install --no-adjust-repos --mon <monitor-node>
```

NOTE

You need to specify **`--no-adjust-repos`** with **`ceph-deploy`** so that **`ceph-deploy`** does not create **`ceph.list`** file on Monitor node.

4. Reconnect the Monitor node to Calamari. From the admin node, execute:

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

5. Upgrade and restart the Ceph Monitor daemon. From the Monitor node, execute:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo restart ceph-mon id={hostname}
```

Using an ISO

To upgrade a Monitor node, log in to the node, remove **`ceph-mon`** repository under **`/etc/apt/sources.list.d/`**, re-install Ceph from the administration node and reconnect 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.

1. Execute on the Monitor node:

```
$ cd /etc/apt/sources.list.d/
$ sudo rm -rf ceph-mon.list
```

2. From the administration node, execute:

```
$ ceph-deploy repo ceph-mon <monitor-node>
$ ceph-deploy install --no-adjust-repos --mon <monitor-node>
```

3. Reconnect the Monitor node to Calamari. From the administration node, execute:

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

4. Upgrade and restart Ceph Monitor daemon. From the Monitor node, execute:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo restart ceph-mon id={hostname}
```

3.1.3. 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 the Online Repositories

1. Remove existing Ceph repositories in the OSD node:

```
$ cd /etc/apt/sources.list.d/
$ sudo rm -rf ceph-osd.list
```

2. Set online OSD repository on OSD node from administration node:

```
$ ceph-deploy repo --repo-url
'https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu
/1.3-updates/OSD' --gpg-url
https://www.redhat.com/security/fd431d51.txt ceph-osd <osd-node>
```

3. Reinstall Ceph on OSD node from the administration node:

```
$ ceph-deploy install --no-adjust-repos --osd <osd-node>
```

NOTE

You need to specify **`--no-adjust-repos`** with **`ceph-deploy`** so that **`ceph-deploy`** does not create **`ceph.list`** file on OSD node.

4. Reconnect the OSD node to Calamari. From the administration node, execute:

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

5. Update and restart the Ceph OSD daemon. From the OSD node, execute:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo restart ceph-osd id={id}
```

Using an ISO

To upgrade a OSD node, log in to the node, remove **ceph-osd** repository under **/etc/apt/sources.list.d/**, re-install Ceph from the administration node and reconnect OSD node to Calamari. Finally, upgrade and restart the OSD daemon(s).

1. Execute on the OSD node:

```
$ cd /etc/apt/sources.list.d/
$ sudo rm -rf ceph-osd.list
```

2. From the administration node, execute:

```
$ ceph-deploy repo ceph-osd <osd-node>
$ ceph-deploy install --no-adjust-repos --osd <osd-node>
```

3. Reconnect the OSD node to Calamari. From the administration node, execute:

```
$ ceph-deploy calamari connect --master '<FQDN_Calamari_admin_node>'
<osd-node>
```

4. Upgrade and restart the Ceph OSD daemon. From the OSD node, execute:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo restart ceph-osd id=<id>
```

3.1.4. Object Gateway Nodes

To upgrade a Ceph Object Gateway node, log in to the node, remove **ceph-mon** or **ceph-osd** repository, whichever was installed for the **radosgw** package in Red Hat Ceph Storage 1.3.0 or 1.3.1, under **/etc/apt/sources.list.d/**, set the online **Tools** repository from the administration node, and re-install the Ceph Object Gateway daemon. Finally, upgrade and restart Ceph Object Gateway.

Using the Online Repositories

1. Remove existing Ceph repository on the Object Gateway node:

```
$ cd /etc/apt/sources.list.d/
```

```
$ sudo rm -rf ceph-mon.list
```

OR

```
$ sudo rm -rf ceph-osd.list
```

NOTE

For Red Hat Ceph Storage v1.3.1, you had to install either **ceph-mon** or **ceph-osd** repository for the **radosgw** package. Remove the repository that was previous installed before setting the **Tools** repository for Red Hat Ceph Storage v1.3.3.

If upgrading from Red Hat Ceph Storage 1.3.2, then this step can be skipped.

2. Set the online Tools repository from administration node:

```
$ ceph-deploy repo --repo-url
'https://customername:customerpasswd@rhcs.download.redhat.com/ubuntu
/1.3-updates/Tools' --gpg-url
https://www.redhat.com/security/fd431d51.txt Tools <rgw-node>
```

3. Reinstall Object Gateway from the administration node:

```
$ ceph-deploy install --no-adjust-repos --rgw <rgw-node>
```

4. For federated deployments, from the Object Gateway node, execute:

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

5. Upgrade and restart the Object Gateway:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo service radosgw restart id=rgw.<short-hostname>
```

NOTE

If you modify the **ceph.conf** file for **radosgw** to run on port **80** then run **sudo service apache2 stop** before restarting the gateway.

Using an ISO

To upgrade a Ceph Object Gateway node, log in to the node, remove the **ceph** repository under **/etc/apt/sources.list.d/**, stop the Ceph Object Gateway daemon (radosgw) and stop the Apache/FastCGI instance. From the administration node, re-install the Ceph Object Gateway daemon. Finally, restart Ceph Object Gateway.

1. Remove existing Ceph repository in the Ceph Object Gateway node:

```
$ cd /etc/apt/sources.list.d/
$ sudo rm -rf ceph.list
```

2. Stop Apache/Radosgw:

```
$ sudo service apache2 stop
$ sudo /etc/init.d/radosgw stop
```

3. From the administration node, execute:

```
$ ceph-deploy repo ceph-mon <rgw-node>
$ ceph-deploy install --no-adjust-repos --rgw <rgw-node>
```

NOTE

Both **ceph-mon** and **ceph-osd** repository contains the **radosgw** package. So, you can use anyone of them for the Object Gateway upgrade.

4. For federated deployments, from the Ceph Object Gateway node, execute:

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

5. Finally, from the Ceph Object Gateway node, restart the gateway:

```
$ sudo service radosgw restart
```

To upgrade a Ceph Object Gateway node, log in to the node and remove **ceph-mon** or **ceph-osd** repository under **/etc/apt/sources.list.d/**, whichever was previous installed for the **radosgw** package in Red Hat Cecph Storage 1.3.0. From the administration node, re-install the Ceph Object Gateway daemon. Finally, upgrade and restart Ceph Object Gateway.

1. Remove existing Ceph repository in the Ceph Object Gateway node:

```
$ cd /etc/apt/sources.list.d/
```

```
$ sudo rm -rf ceph-mon.list
```

OR

```
$ sudo rm -rf ceph-osd.list
```

NOTE

For Red Hat Ceph Storage v1.3.1, you had to install either **ceph-mon** or **ceph-osd** repository for the **radosgw** package. You have to remove the repository that was previous installed before setting the new repo for RHCS v1.3.2.

2. From the administration node, execute:

```
$ ceph-deploy repo ceph-mon <rgw-node>
$ ceph-deploy install --no-adjust-repos --rgw <rgw-node>
```

NOTE

Both **ceph-mon** and **ceph-osd** repo contains the **radosgw** package. So, you can use anyone of them for the gateway upgrade.

3. For federated deployments, from the Object Gateway node, execute:

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

4. Upgrade and restart the Object Gateway:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo service radosgw restart id=rgw.<short-hostname>
```

NOTE

If you modify the **ceph.conf** file for **radosgw** to run on port **80** then run **sudo service apache2 stop** before restarting the gateway.

3.2. REVIEWING CRUSH TUNABLES

If you have been using Ceph for a while and you are using an older CRUSH tunables setting such as **bobtail**, you should investigate and set your CRUSH tunables to **optimal**.

NOTE

Resetting your CRUSH tunables may result in significant rebalancing. See the Storage Strategies Guide, Chapter 9, Tunables for additional details on CRUSH tunables.

For example:

```
ceph osd crush tunables optimal
```