



Red Hat Ceph Storage

1.2.3

Installation Guide for RHEL (x86 64)

Installing Calamari and Ceph Storage on RHEL x86_64.

Red Hat Customer Content
Services

Red Hat Ceph Storage 1.2.3 Installation Guide for RHEL (x86 64)

Installing Calamari and Ceph Storage on RHEL x86_64.

Legal Notice

Copyright © 2017 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

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 RHEL 6 and RHEL 7.

Table of Contents

PART I. INSTALLATION	4
CHAPTER 1. SUBSCRIBE TO THE CONTENT DELIVERY NETWORK (CDN)	5
CHAPTER 2. PRE-INSTALLATION REQUIREMENTS	7
2.1. OPERATING SYSTEM	7
2.2. DNS NAME RESOLUTION	7
2.3. NICS	7
2.4. NETWORK	7
2.5. FIREWALL FOR RHEL 6	8
2.6. FIREWALL FOR RHEL 7	8
2.7. NTP	9
2.8. INSTALL SSH SERVER	10
2.9. CREATE A CEPH USER	10
2.10. ENABLE PASSWORD-LESS SSH	11
2.11. ADJUST ULIMIT ON LARGE CLUSTERS	12
2.12. DISABLE RAID	12
2.13. ADJUST PID COUNT	12
2.14. HARD DRIVE PREP ON RHEL 6	12
2.15. SELINUX	13
2.16. DISABLE EPEL ON CEPH CLUSTER NODES	13
2.17. INSTALL XFSPROGS (RHEL 6)	13
CHAPTER 3. SETTING UP YOUR ADMINISTRATION SERVER	14
CHAPTER 4. UPGRADING YOUR ADMINISTRATION SERVER	16
PART II. STORAGE CLUSTER QUICK START	17
CHAPTER 5. EXECUTING CEPH-DEPLOY	18
CHAPTER 6. CREATE A CLUSTER	19
CHAPTER 7. INSTALL CEPH	22
CHAPTER 8. ADD INITIAL MONITORS	23
CHAPTER 9. CONNECT MONITOR HOSTS TO CALAMARI	24
CHAPTER 10. MAKE YOUR CALAMARI ADMIN NODE A CEPH ADMIN NODE	25
CHAPTER 11. ADJUST CRUSH TUNABLES	26
CHAPTER 12. ADD OSDS	27
CHAPTER 13. CONNECT OSD HOSTS TO CALAMARI	28
CHAPTER 14. CREATE A CRUSH HIERARCHY	29
CHAPTER 15. ADD OSD HOSTS/CHASSIS TO THE CRUSH HIERARCHY	30
CHAPTER 16. CHECK CRUSH HIERARCHY	31
CHAPTER 17. CHECK CLUSTER HEALTH	32
CHAPTER 18. LIST/CREATE A POOL	33
CHAPTER 19. STORING/RETRIEVING OBJECT DATA	34

PART III. UPGRADING CEPH	36
CHAPTER 20. UPGRADING YOUR CLUSTER FROM V1.2.2 TO V1.2.3	37
CHAPTER 21. STORAGE CLUSTER UPGRADE	39
CHAPTER 22. REVIEWING CRUSH TUNABLES	41

PART I. INSTALLATION

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 v1.2.3 for x86_64 architecture on Red Hat Enterprise Linux (RHEL) 6 and RHEL 7.

To simplify installation and to support deployment scenarios where security measures preclude direct Internet access, Red Hat Ceph Storage v1.2.3 is installed from a single software build delivered as an ISO with the **ice_setup** package, which installs the **ice_setup** script. When you execute the **ice_setup** script, it will install a local repository, the Calamari monitoring and administration server and the Ceph installation scripts, including a **cephdeploy.conf** file pointing **ceph-deploy** to the local repository.

We expect that you will have a dedicated administration node that will host the local repository and the Calamari monitoring and administration server. The following instructions assume you will install (or update) the repository on the dedicated administration node.

The administration/Calamari server hardware requirements vary with the size of your cluster. A minimum recommended hardware configuration for a Calamari server includes at least 4GB of RAM, a dual core CPU on x86_64 architecture and 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.

CHAPTER 1. SUBSCRIBE TO THE CONTENT DELIVERY NETWORK (CDN)

Red Hat Ceph Storage installation requires that the Calamari administration node be subscribed/registered to a number of Subscription Management Service repositories. These repositories are used to retrieve both the initial installation packages and later updates as they become available.

1. Register the Calamari node with Subscription Management Service.

Run the following command and enter your Red Hat Network user name and password to register the system with the Red Hat Network:

```
sudo subscription-manager register
```

2. Identify available entitlement pools.

Using **sudo**, run the following command to find entitlement pools containing the repositories required to install Red Hat Ceph Storage:

```
sudo subscription-manager list --available | grep -A8 "Red Hat Ceph Storage"
```

3. Attach entitlement pools to the Calamari node.

Use the pool identifiers located in the previous step to attach following entitlements to the Calamari node:

- ✳ Red Hat Enterprise Linux Server
- ✳ Red Hat Ceph Storage Installer
- ✳ Red Hat Ceph Storage Calamari
- ✳ Red Hat Ceph Storage MON
- ✳ Red Hat Ceph Storage OSD

Run the following command to attach each of the entitlements:

```
sudo subscription-manager attach --pool=[POOLID]
```

4. Enable the required repositories.

For Red Hat Ceph Storage v1.2.3, enable all of the Red Hat Ceph repositories on the Calamari node.

```
sudo subscription-manager repos --enable=[RH-Ceph-Storage-Repo-Name]
```

For RHEL 6, execute:

```
sudo subscription-manager repos --enable=rhel-6-server-rpms --enable=rhel-scalefs-for-rhel-6-server-rpms --enable=rhel-6-server-rhceph-1.2-calamari-rpms --enable=rhel-6-server-rhceph-
```

```
1.2-installer-rpms --enable=rhel-6-server-rhceph-1.2-mon-rpms --  
enable=rhel-6-server-rhceph-1.2-osd-rpms
```

For RHEL 7, execute:

```
sudo subscription-manager repos --enable=rhel-7-server-rpms --  
enable=rhel-7-server-rhceph-1.2-calamari-rpms --enable=rhel-7-  
server-rhceph-1.2-installer-rpms --enable=rhel-7-server-rhceph-  
1.2-mon-rpms --enable=rhel-7-server-rhceph-1.2-osd-rpms
```

5. Verify if the repositories are enabled.

Run the following command to verify if the repositories are enabled:

```
yum repolist
```

6. Update the repositories.

```
sudo yum update
```

CHAPTER 2. PRE-INSTALLATION REQUIREMENTS

If you are installing Red Hat Ceph Storage v1.2.3 for the first time, you should review the pre-installation requirements first. Depending on your Linux distribution, you may need to adjust default settings and install required software before setting up a local repository and installing Calamari and Ceph.

2.1. OPERATING SYSTEM

Red Hat Ceph Storage v1.2.3 and beyond requires a homogeneous operating system distribution and version (e.g., RHEL 6, RHEL7) on x86_64 architecture for all Ceph nodes, including the Calamari cluster. We do not support clusters with heterogeneous operating systems and versions.

2.2. 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.

2.3. NICS

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.

2.4. 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 6, see [Ethernet Interfaces](#).

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

2.5. FIREWALL FOR RHEL 6

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**. 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., **6800-7100**). **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). For example, if you have 4 OSDs, open **4 x 3** ports (**12**).

```
sudo iptables -I INPUT 1 -i <iface> -m multiport -p tcp -s <ip-
address>/<netmask> --dports 6800:6811 -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. For example:

```
/sbin/service iptables save
```

2.6. FIREWALL FOR RHEL 7

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 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.

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

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

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

Finally, you **MUST** also open ports for OSD traffic (e.g., **6800-7100**). **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). For example, if you have 4 OSDs, open **4 x 3** ports (**12**).

```
sudo firewall-cmd --zone=public --add-port=6800-6811/tcp --permanent
```

Once the foregoing procedures are complete, reload the firewall configuration to ensure that the changes take effect.

```
sudo firewall-cmd --reload
```

For additional details on **firewalld**, see [Using Firewalls](#).

2.7. NTP

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 yum install ntp
```

2. Make sure NTP starts on reboot.

For RHEL 6, execute:

```
sudo chkconfig ntpd on
```

For RHEL 7, execute:

```
systemctl enable ntpd.service
```

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

For RHEL 6, execute:

```
sudo /etc/init.d/ntpd start
```

For RHEL 7, execute:

```
sudo systemctl start ntpd
```

Then, check its status.

For RHEL 6, execute:

```
sudo /etc/init.d/ntpd status
```

For RHEL 7, execute:

```
sudo systemctl status ntpd
```

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

```
ntpq -p
```

For additional details on NTP for RHEL 6, see [Network Time Protocol Setup](#).

For additional details on NTP for RHEL 7, see [Configuring NTP Using ntpd](#).

2.8. INSTALL SSH SERVER

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

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

```
sudo yum install openssh-server
```

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

For additional details on OpenSSH for RHEL 6, see [OpenSSH](#).

For additional details on OpenSSH for RHEL 7, see [OpenSSH](#).

2.9. CREATE A CEPH 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 user on **ALL** Ceph nodes in the cluster. 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 (e.g., **root**, **admin**, **<productname>**). The following procedure, substituting **<username>** for the user name you define, describes how to create a user with passwordless **sudo** on a node called **ceph-server**.

1. Create a user on each Ceph Node. :

```
ssh user@ceph-server
sudo useradd -d /home/<username> -m <username>
sudo passwd <username>
```

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

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

Ensure the file permissions are correct.

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

2.10. 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 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>
```

2.11. ADJUST ULIMIT ON LARGE CLUSTERS

For users that will run Ceph administrator commands on large clusters (e.g., 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.

2.12. DISABLE RAID

If you have RAID (not recommended), configure your RAID controllers to RAID 0 (JBOD).

2.13. ADJUST PID COUNT

Hosts with high numbers of OSDs (e.g., > 20) may spawn a lot of threads, especially during recovery and re-balancing. Many Linux kernels 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:

```
sudo sysctl -p
```

To verify the changes, execute:

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

2.14. HARD DRIVE PREP ON RHEL 6

Ceph aims for data safety, which means that when the Ceph Client receives notice that data was written to a storage drive, that data was actually written to the storage drive (i.e., it's not in a journal or drive cache, but yet to be written). On RHEL 6, disable the write cache if the journal is on a raw drive.

Use `hdparm` to disable write caching on OSD storage drives:

```
-
```



```
sudo hdparm -W 0 /<path-to>/<disk> 0
```

RHEL 7 has a newer kernel that handles this automatically.

2.15. SELINUX

SELinux is set to **Enforcing** by default. For Ceph Storage v1.2.3, set SELinux to **Permissive** or disable it entirely and ensure that your installation and cluster is working properly. To set SELinux to **Permissive**, execute the following:

```
sudo setenforce 0
```

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

2.16. DISABLE EPEL ON CEPH 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.

2.17. INSTALL XFSPROGS (RHEL 6)

Red Hat Ceph Storage for RHEL 6 requires **xfspgros** for OSD nodes.



Note

You should ensure that your Calamari node has already run **subscription-manager** to enable the Red Hat Ceph Storage repositories before enabling the Scalable File System repository.

As part of the Red Hat Ceph Storage product, Red Hat includes an entitlement to the Scalable File System set of packages for RHEL6, which includes **xfspgros**. On each Ceph Node, using **sudo**, enable the Scalable File System repo and install **xfspgros**:

```
sudo subscription-manager repos --enable=rhel-scalefs-for-rhel-6-  
server-rpms  
sudo yum install xfspgros
```

CHAPTER 3. SETTING UP YOUR ADMINISTRATION SERVER

Red Hat Ceph Storage uses an administration server for a Red Hat Ceph Storage repository, the Calamari monitoring and administration server, and your cluster's Ceph configuration and authentication keys.

Visit the Software & Download Center in the Red Hat Customer Service Portal (<https://access.redhat.com/downloads>) to obtain the Red Hat Ceph Storage installation ISO image files. Use a valid Red Hat Subscription to download the full installation files, obtain a free evaluation installation, or follow the links in this page to purchase a new Red Hat Subscription. To download the Red Hat Ceph Storage installation files using a Red Hat Subscription or a Red Hat Evaluation Subscription:

1. Visit the Red Hat Customer Service Portal at <https://access.redhat.com/login> and enter your user name and password to log in.
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. Using **sudo**, mount the image:

```
sudo mount -o loop <path_to_iso>/rhceph-1.2.3-rhel-6-x86_64.iso /mnt
```

OR

```
sudo mount <path_to_iso>/rhceph-1.2.3-rhel-7-x86_64.iso /mnt
```

5. Using **sudo**, copy each Ceph *.pem product certificates from **/mnt** to **/etc/pki/product**. For example:

```
sudo cp /mnt/RHCeph-Calamari-1.2-x86_64-c1e8ca3b6c57-285.pem /etc/pki/product/285.pem
sudo cp /mnt/RHCeph-Installer-1.2-x86_64-8ad6befe003d-281.pem /etc/pki/product/281.pem
sudo cp /mnt/RHCeph-MON-1.2-x86_64-d8afd76a547b-286.pem /etc/pki/product/286.pem
sudo cp /mnt/RHCeph-OSD-1.2-x86_64-25019bf09fe9-288.pem /etc/pki/product/288.pem
```

6. Using **sudo**, install the setup script.

```
sudo yum install /mnt/ice_setup-*.rpm
```

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

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

- Using **sudo**, run the setup script in the working directory you created at step 7 of [Setting Up Your Administration Server](#). **NOTE:** You cannot run the setup script in `/mnt` or a read-only directory or the script will crash. The script will output a **cephdeploy.conf** file, which **ceph-deploy** will use to point to the local repository.

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

The setup script performs the following operations:

- ✦ It moves the RPMs to `/opt/ICE` and `/opt/calamari`
- ✦ It creates a `.repo` file for the **ceph-deploy** and **calamari** packages pointing to a local path
- ✦ It installs the Calamari server packages on the admin node
- ✦ It installs the **ceph-deploy** package on the admin node; and
- ✦ It writes a **cephdeploy.conf** file to `/opt/ICE`.

- To receive updates to **calamari**, **ceph-deploy** and **ice_setup** on the admin node, using **sudo**, execute:

```
sudo yum update
```

- Using **sudo**, run **ice_setup** with the **update all** sub-command. This will synchronize new packages (if any) from the Red Hat CDN into the local repository on your Calamari admin node.

```
sudo ice_setup update all
```

- Using **sudo**, initialize the Calamari monitoring and administration server.

```
sudo calamari-ctl initialize
```



Note

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

At this point, you should be able to access the Calamari web server via a web browser. Proceed to the [Storage Cluster Quick Start](#).

CHAPTER 4. UPGRADING YOUR ADMINISTRATION SERVER

Periodically, Red Hat will provide updated packages for Ceph Storage. You may get the latest version of **ice_setup** and upgrade your administration server with the latest packages. To upgrade your administration server, perform the following steps:

1. Using **sudo**, update your Calamari admin node to the latest version of **ice_setup**. (You will need at least version 0.3.0.)

```
sudo yum update ice_setup
```

2. Using **sudo**, run **ice_setup** with the **update all** sub-command. **ice_setup** will synchronize the new packages from the Red Hat CDN onto the local repository on your Calamari admin node.

```
sudo ice_setup update all
```

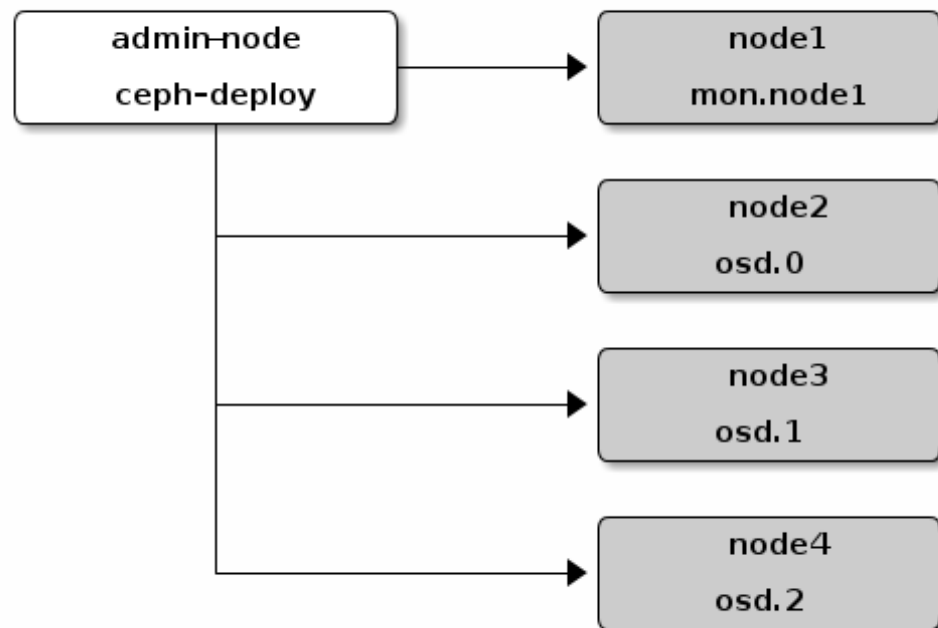
3. The updated packages will now be available to the nodes in your cluster with **yum update**.

```
sudo yum update
```

If the updates contain new packages for your Ceph Storage Cluster, you should upgrade your cluster too. See [Storage Cluster Upgrade](#) for details.

PART II. 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.



CHAPTER 5. EXECUTING CEPH-DEPLOY

When executing **ceph-deploy** with Red Hat Ceph Storage, **ceph-deploy** will need to retrieve Ceph packages from the **/opt/ICE** directory on your Calamari admin host, so you need to ensure that **ceph-deploy** has access to the **cephdeploy.conf** file that was written to your local working directory when you executed **calamari-ctl initialize**.

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



Important

The **ceph-deploy** utility does not issue **sudo** commands needed on the remote host. Execute **ceph-deploy** commands as a regular user (not as **root** or using **sudo**). The [Create a Ceph 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.

The **ceph-deploy** utility will output files to the current directory. Ensure you are in this directory when executing **ceph-deploy**, and ensure that **ceph-deploy** points to the **cephdeploy.conf** file generated by **calamari-ctl initialize** when installing Red Hat Ceph Storage packages.



Note

On RHEL 6, you may see a backtrace after **ceph-deploy** runs. This is cosmetic/harmless and does not affect the operation of **ceph-deploy**.

CHAPTER 6. 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 purgedata <ceph-node> [<ceph-node>]
ceph-deploy forgetkeys
```

To purge the Ceph packages too, you may also execute:

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

If you execute **purge**, 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.

At this stage, you may begin editing your Ceph configuration file.



Note

If you choose not to use **ceph-deploy** you will have to deploy Ceph manually or refer to Ceph manual deployment documentation and configure a deployment tool (e.g., Chef, Juju, Puppet, etc.) to perform each operation **ceph-deploy** performs for you.

2. 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.

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

```
ms_bind_ipv6 = true
```

4. 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 find the product of the **filestore_max_sync_interval** and the expected throughput, and multiply the product by two (2). 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.

5. Set the number of copies to store (default is **3**) and the default minimum required 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).

6. Set the maximum number of placement groups per OSD. The Ceph Storage Cluster has a default maximum value of 300 placement groups per OSD. You can set a different maximum value in your Ceph configuration file (i.e., where **n** is the maximum number of PGs per OSD).

```
mon_pg_warn_max_per_osd = n
```

Multiple pools can use the same CRUSH ruleset. When an OSD has too many placement groups associated to it, Ceph performance may degrade due to resource use and load. This setting warns you, but you may adjust it to your needs and the capabilities of your hardware.

7. 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. For example:

```
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

8. 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
```

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>
```

CHAPTER 7. INSTALL CEPH

Ensure that **ceph-deploy** is pointing to the **cephdeploy.conf** file generated by **calamari-ctl initialize** (e.g., in the exemplary **~/ceph-config** directory, the **/opt/ICE** directory, etc.). Otherwise, you may not receive packages from the local repository. 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**. To install Ceph on remote nodes, first use **--repo** option with **ceph-deploy** to install the repo files on remote nodes.

For **admin node**, execute:

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

For example:

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

For other nodes, execute:

```
ceph-deploy install --repo <ceph-node> [<ceph-node> ...]  
ceph-deploy install <ceph-node> [<ceph-node> ...]
```

For example:

```
ceph-deploy install --repo node1 node2 node3 node4  
ceph-deploy install node1 node2 node3 node4
```

The **ceph-deploy** utility will install Ceph on each node. **NOTE:** If you use **ceph-deploy purge**, you must re-execute these steps to re-install Ceph.

CHAPTER 8. 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**

CHAPTER 9. CONNECT MONITOR HOSTS TO CALAMARI

Once you have added the initial monitor(s), you need to connect the monitor hosts to Calamari.

```
ceph-deploy calamari connect <ceph-node>[<ceph-node> ...]
```

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

```
ceph-deploy calamari connect node1
```

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

CHAPTER 10. 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** or **ceph-common** on the Calamari node.

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

For example:

```
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:

```
sudo ceph quorum_status --format json-pretty
```

Ensure that you have acceptable permissions for the **/etc/ceph/ceph.client.admin.keyring**. You can use **sudo** when executing the **ceph** command, or you can change the keyring permissions to enable a specific user or group. Keyring permissions provide administrative capability to the Red Hat Ceph Storage cluster. So exercise caution if many users have access to the Ceph nodes and admin node.

```
sudo chmod +r /etc/ceph/ceph.client.admin.keyring
```



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.

CHAPTER 11. ADJUST CRUSH TUNABLES

Red Hat Ceph Storage CRUSH tunables defaults to **bobtail**, which refers to an older release of Ceph. This setting guarantees older Ceph clusters are compatible with older Linux kernels. However, new Ceph clusters running on RHEL 7 should reset CRUSH tunables to **optimal**. For example:

```
ceph osd crush tunables optimal
```

Please see the [Storage Strategies Guide, Chapter 9, Tunables](#) for more details on the CRUSH tunables.

CHAPTER 12. ADD OSDS

Before creating OSDs, consider the following:

- ✦ We recommend using the XFS filesystem (default).
- ✦ We recommend using SSDs for journals. It is common to partition SSDs to serve multiple OSDs. Ensure that the number of SSD partitions does not exceed your SSD's sequential write limits. Also, ensure that SSD partitions are properly aligned, or their write performance will suffer.
- ✦ We recommend using **ceph-deploy disk zap** on a Ceph OSD drive before executing **ceph-deploy osd create**. For example:

```
ceph-deploy disk zap <ceph-node>:<data-drive>
```

From your admin node, use **ceph-deploy** to prepare the OSDs.

```
ceph-deploy osd prepare <ceph-node>:<data-drive>[:<journal-partition>]
[<ceph-node>:<data-drive>[:<journal-partition>]]
```

For example:

```
ceph-deploy osd prepare node2:sdb:ssdb node3:sdd:ssdb node4:sdd:ssdb
```

In the foregoing example, **sdb** is a spinning hard drive. Ceph will use the entire drive for OSD data. **ssdb** is a partition on an SSD drive, which Ceph will use to store the journal for the OSD.

Once you prepare OSDs, use **ceph-deploy** to activate the OSDs.

```
ceph-deploy osd activate <ceph-node>:<data-drive>:<journal-partition>
[<ceph-node>:<data-drive>:<journal-partition>]
```

For example:

```
ceph-deploy osd activate node2:sdb:ssdb node3:sdd:ssdb node4:sdd:ssdb
```

To achieve an **active + clean** state, you must add as many OSDs as the value of **osd pool default size = <n>** from your Ceph configuration file.

CHAPTER 13. 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 <ceph-node>[<ceph-node> ...]
```

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

```
ceph-deploy calamari connect 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.

CHAPTER 14. 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
```

CHAPTER 15. 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).

CHAPTER 16. 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>
```

CHAPTER 17. 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.

CHAPTER 18. LIST/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**.

CHAPTER 19. STORING/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 context 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, let's 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.

PART III. UPGRADING CEPH

You may upgrade your administration server and your Ceph Storage cluster when Red Hat provides fixes or delivers a major release.

CHAPTER 20. UPGRADING YOUR CLUSTER FROM V1.2.2 TO V1.2.3

To obtain the Red Hat Ceph Storage installation ISO image files for the newer version, visit the Software & Download Center in the Red Hat Customer Service Portal (<https://access.redhat.com/downloads>). Use a valid Red Hat Subscription to download the full installation files. To download the Red Hat Ceph Storage installation files using a Red Hat Subscription or a Red Hat Evaluation Subscription:

1. Visit the Red Hat Customer Service Portal at <https://access.redhat.com/login> and enter your user name and password to log in.
2. Click **Downloads** to visit the **Software & Download** Center.
3. In the Red Hat Ceph Storage area, click **Download Software** and download the version of the software you want to upgrade to.



Important

Do not downgrade to an earlier version, as it may introduce compatibility issues.

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

```
sudo mount -o loop <path_to_iso>/rhceph-1.2.3-rhel-6-x86_64.iso /mnt
```

OR

```
sudo mount <path_to_iso>/rhceph-1.2.3-rhel-7-x86_64.iso /mnt
```

2. Using **sudo**, copy each Ceph *.pem product certificates from **/mnt** to **/etc/pki/product**. For example:

```
sudo cp /mnt/RHCeph-Calamari-1.2-x86_64-c1e8ca3b6c57-285.pem /etc/pki/product/285.pem
sudo cp /mnt/RHCeph-Installer-1.2-x86_64-8ad6befe003d-281.pem /etc/pki/product/281.pem
sudo cp /mnt/RHCeph-MON-1.2-x86_64-d8afd76a547b-286.pem /etc/pki/product/286.pem
sudo cp /mnt/RHCeph-OSD-1.2-x86_64-25019bf09fe9-288.pem /etc/pki/product/288.pem
```

3. Using **sudo**, install the setup script.

```
sudo yum install /mnt/ice_setup-*.rpm
```

4. Back up your Ceph configuration, log and key files.
5. Using **sudo**, run the setup script in the working directory you created at step 7 of **Setting Up Your Administration Server**.

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



Note

You cannot run the setup script in `/mnt` or a read-only directory or the script will crash. The script will output a `cephdeploy.conf` file, which `ceph-deploy` will use to point to the local repository. Remember to backup the original `cephdeploy.conf` file too.

- Remove any **priority** settings in `/etc/yum.repos.d/ceph.repo`. If you are upgrading from ICE 1.2.2, you may have a `priority=` setting in `ceph.repo`. This should not be used in Red Hat Ceph Storage 1.2.3 or any later versions.

```
sudo grep ^priority ` /etc/yum.repos.d/ceph.repo `
sudo vi /etc/yum.repos.d/ceph.repo
```

- Restart Apache and Calamari.
- Update your system.

```
sudo yum update
```

ICE v1.2.2 can run on RHEL 7.0, but not on RHEL 7.1. If you deferred an upgrade from RHEL 7.0 to RHEL 7.1, because you were running ICE v1.2.2, you MAY upgrade to RHEL 7.1 once you have upgraded to Red Hat Ceph Storage v1.2.3.

- Update your system to RHEL 7.1 by subscribing to the Content Delivery Network (CDN) and enabling respective repositories for RHEL 7.
- In `/etc/yum/pluginconf.d/priorities.conf`, add the following line:

```
check_obsoletes=1
```

- In `/etc/yum/repos.d` remove the following line:

```
priority=1
```

- Update your system.

```
sudo yum update
```

- To upgrade the Ceph daemons running on your cluster hosts, see **Storage Cluster Upgrade** for details.

CHAPTER 21. STORAGE CLUSTER UPGRADE

Upgrading Ceph daemons involves installing the upgraded packages, and restarting each Ceph daemon. We recommend upgrading in this order:

- ✦ Ceph Deploy
- ✦ Ceph Monitors
- ✦ Ceph OSD Daemons
- ✦ Ceph Object Gateways

- ✦ To upgrade **ceph-deploy**, execute:

```
sudo yum install ceph-deploy
```

- ✦ To upgrade monitors, execute the following on your monitor nodes:

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

ceph-deploy will install the latest version of Ceph.

Restart your monitors one at a time. Give each daemon time to come **up** and **in**, rejoining the quorum before you restart the next instance. To restart a monitor, execute **ceph** with the **restart** command. Use the following syntax:

```
sudo /etc/init.d/ceph [options] restart mon.[id]
```

- ✦ To upgrade OSDs, execute the following on your OSD nodes:

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

ceph-deploy will install the latest version of Ceph.

We recommend upgrading OSDs by CRUSH hierarchy—i.e., by failure domain or performance domain. Give each daemon time to come **up** and **in** with the cluster reaching a **HEALTH_OK** state before proceeding to the next CRUSH hierarchy. To restart an OSD, execute **ceph** with the **restart** command. Use the following syntax:

```
sudo /etc/init.d/ceph [options] restart osd.[id]
```

- ✦ To upgrade a Ceph Object Gateway daemon, execute the following:

```
sudo yum install ceph-radosgw
```

To upgrade the Ceph Object Gateway synchronization agent, execute the following:

```
sudo yum install radosgw-agent
```

Restart each Ceph Object gateway daemon. To do so, execute the following on each host:

On RHEL 7:

```
sudo systemctl restart ceph-radosgw
```

On RHEL 6:

```
sudo service ceph-radosgw restart
```

If you are running a federated architecture, restart your sync agent(s). For data replication agents, go to the terminal and execute **ctrl + c**; then, execute:

```
radosgw-agent -c [config-file]
```

For metadata replication agents, go to the terminal and execute **ctrl + c**; then, execute:

```
radosgw-agent -c [config-file] --metadata-only
```

CHAPTER 22. REVIEWING CRUSH TUNABLES

If you have been using Ceph for awhile and you are using an older CRUSH tunables setting such as **bobtail**, you should investigate setting 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
```