



# Red Hat Ceph Storage 3

## Container Guide

Deploying and Managing Red Hat Ceph Storage in Containers



# Red Hat Ceph Storage 3 Container Guide

---

Deploying and Managing Red Hat Ceph Storage in Containers

## Legal Notice

Copyright © 2019 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, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

The OpenStack<sup>®</sup> 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 describes how to deploy and manage Red Hat Ceph Storage in containers.

## Table of Contents

<b>CHAPTER 1. DEPLOYING RED HAT CEPH STORAGE IN CONTAINERS</b> .....	<b>4</b>
1.1. PREREQUISITES	4
1.1.1. Registering Red Hat Ceph Storage Nodes to the CDN and Attaching Subscriptions	4
Prerequisites	4
Procedure	5
Additional Resources	5
1.1.2. Creating an Ansible user with sudo access	6
1.1.3. Enabling Password-less SSH for Ansible	7
Prerequisites	7
Procedure	8
Additional Resources	9
1.1.4. Configuring a firewall for Red Hat Ceph Storage	9
1.1.5. Using a HTTP Proxy	13
Prerequisites	13
Procedure	13
1.2. INSTALLING A RED HAT CEPH STORAGE CLUSTER IN CONTAINERS	13
Prerequisites	14
Procedure	14
1.3. CONFIGURING OSD ANSIBLE SETTINGS FOR ALL NVME STORAGE	21
1.4. INSTALLING THE CEPH OBJECT GATEWAY IN A CONTAINER	23
1.5. INSTALLING METADATA SERVERS	25
1.6. INSTALLING THE CEPH ISCSI GATEWAY IN A CONTAINER	26
1.6.1. Configuring the Ceph iSCSI gateway in a container	28
1.6.2. Removing the Ceph iSCSI gateway in a container	32
1.6.3. Optimizing the performance of the iSCSI Target	33
1.7. UNDERSTANDING THE LIMIT OPTION	34
1.8. ADDITIONAL RESOURCES	35
<b>CHAPTER 2. COLOCATION OF CONTAINERIZED CEPH DAEMONS</b> .....	<b>36</b>
2.1. HOW COLOCATION WORKS AND ITS ADVANTAGES	36
How Colocation Works	36
2.2. SETTING DEDICATED RESOURCES FOR COLOCATED DAEMONS	38
2.3. ADDITIONAL RESOURCES	39
<b>CHAPTER 3. ADMINISTERING CEPH CLUSTERS THAT RUN IN CONTAINERS</b> .....	<b>40</b>
3.1. STARTING, STOPPING, AND RESTARTING CEPH DAEMONS THAT RUN IN CONTAINERS	40
3.2. VIEWING LOG FILES OF CEPH DAEMONS THAT RUN IN CONTAINERS	41
3.3. PURGING CLUSTERS DEPLOYED BY ANSIBLE	42
<b>CHAPTER 4. UPGRADING RED HAT CEPH STORAGE WITHIN CONTAINERS</b> .....	<b>44</b>
4.1. PREREQUISITES	44
4.2. UPGRADING A RED HAT CEPH STORAGE CLUSTER THAT RUNS IN CONTAINERS	44
Prerequisites	45
4.3. UPGRADING THE STORAGE CLUSTER	46
Procedure	46
4.4. UPGRADING RED HAT CEPH STORAGE DASHBOARD	49
<b>CHAPTER 5. MONITORING CEPH CLUSTERS RUNNING IN CONTAINERS WITH THE RED HAT CEPH STORAGE DASHBOARD</b> .....	<b>51</b>
Prerequisites	51
5.1. THE RED HAT CEPH STORAGE DASHBOARD	51
5.2. INSTALLING THE RED HAT CEPH STORAGE DASHBOARD	54

5.3. ACCESSING THE RED HAT CEPH STORAGE DASHBOARD	56
Prerequisites	56
Procedure	56
Additional Resources	57
5.4. CHANGING THE DEFAULT RED HAT CEPH STORAGE DASHBOARD PASSWORD	57
5.5. THE PROMETHEUS PLUGIN FOR RED HAT CEPH STORAGE	58
5.5.1. Prerequisites	58
5.5.2. The Prometheus plugin	58
5.5.3. Managing the Prometheus environment	58
5.5.4. Working with the Prometheus data and queries	60
5.5.5. Using the Prometheus expression browser	61
5.5.6. Additional Resources	61
5.6. THE RED HAT CEPH STORAGE DASHBOARD ALERTS	62
5.6.1. Prerequisites	62
5.6.2. About Alerts	62
5.6.3. Accessing the Alert Status dashboard	63
Procedure	63
5.6.4. Configuring the Notification Target	63
Procedure	63
5.6.5. Changing the Default Alerts and Adding New Ones	63
Procedure	63
Additional Resources	63
<b>APPENDIX A. CHANGES IN ANSIBLE VARIABLES BETWEEN VERSION 2 AND 3</b> .....	<b>64</b>



# CHAPTER 1. DEPLOYING RED HAT CEPH STORAGE IN CONTAINERS

This chapter describes how to use the Ansible application with the **ceph-ansible** playbook to deploy Red Hat Ceph Storage 3 in containers.

- To install the Red Hat Ceph Storage, see [Section 1.2, “Installing a Red Hat Ceph Storage Cluster in Containers”](#).
- To install the Ceph Object Gateway, see [Section 1.4, “Installing the Ceph Object Gateway in a Container”](#).
- To install Metadata Servers, see [Section 1.5, “Installing Metadata Servers”](#).
- To learn about the Ansible **--limit** option, see [Section 1.7, “Understanding the limit option”](#).

## 1.1. PREREQUISITES

- Obtain a valid customer subscription.
- Prepare the cluster nodes. On each node:
  - [Register the node to the Content Delivery Network \(CDN\)](#) .
  - [Create an Ansible user with \*\*sudo\*\* access](#).
  - [Enable passwordless SSH access](#) .
  - Optional. [Configure firewall](#).
  - Optional. [Using a HTTP Proxy](#) .

### 1.1.1. Registering Red Hat Ceph Storage Nodes to the CDN and Attaching Subscriptions

Register each Red Hat Ceph Storage (RHCS) node to the Content Delivery Network (CDN) and attach the appropriate subscription so that the node has access to software repositories. Each RHCS node must be able to access the full Red Hat Enterprise Linux 7 base content and the extras repository content.

#### Prerequisites

- A valid Red Hat subscription
- RHCS nodes must be able to connect to the Internet.
- For RHCS nodes that cannot access the internet during installation, you must first follow these steps on a system with internet access:
  1. Start a local Docker registry:

```
# docker run -d -p 5000:5000 --restart=always --name registry registry:2
```

2. Pull the Red Hat Ceph Storage 3.x image from the Red Hat Customer Portal:



```
# docker pull registry.access.redhat.com/rhceph/rhceph-3-rhel7
```

3. Tag the image:

```
# docker tag registry.access.redhat.com/rhceph/rhceph-3-rhel7 <local-host-fqdn>:5000/cephimageinlocalreg
```

Replace **<local-host-fqdn>** with your local host FQDN.

4. Push the image to the local Docker registry you started:

```
# docker push <local-host-fqdn>:5000/cephimageinlocalreg
```

Replace **<local-host-fqdn>** with your local host FQDN.

### Procedure

Perform the following steps on all nodes in the storage cluster as the **root** user.

1. Register the node. When prompted, enter your Red Hat Customer Portal credentials:

```
# subscription-manager register
```

2. Pull the latest subscription data from the CDN:

```
# subscription-manager refresh
```

3. List all available subscriptions for Red Hat Ceph Storage:

```
# subscription-manager list --available --all --matches="*Ceph*"
```

Identify the appropriate subscription and retrieve its Pool ID.

4. Attach the subscription:

```
# subscription-manager attach --pool=$POOL_ID
```

### Replace

- **\$POOL\_ID** with the Pool ID identified in the previous step.

5. Disable the default software repositories. Then, enable the Red Hat Enterprise Linux 7 Server and Red Hat Enterprise Linux 7 Server Extras repositories:

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

6. Update the system to receive the latest packages:

```
# yum update
```

### Additional Resources

- See the [Registering a System and Managing Subscriptions](#) chapter in the System Administrator's Guide for Red Hat Enterprise Linux 7.

## 1.1.2. Creating an Ansible user with sudo access

Ansible must be able to log into all the Red Hat Ceph Storage (RHCS) nodes as a user that has **root** privileges to install software and create configuration files without prompting for a password. You must create an Ansible user with password-less **root** access on all nodes in the storage cluster when deploying and configuring a Red Hat Ceph Storage cluster with Ansible.

### Prerequisite

- Having **root** or **sudo** access to all nodes in the storage cluster.

### Procedure

1. Log in to a Ceph node as the **root** user:

```
ssh root@$HOST_NAME
```

#### Replace

- **\$HOST\_NAME** with the host name of the Ceph node.

### Example

```
# ssh root@mon01
```

Enter the **root** password when prompted.

2. Create a new Ansible user:

```
adduser $USER_NAME
```

#### Replace

- **\$USER\_NAME** with the new user name for the Ansible user.

### Example

```
# adduser admin
```



#### IMPORTANT

Do not use **ceph** as the user name. The **ceph** user name is reserved for the Ceph daemons. A uniform user name across the cluster can improve ease of use, but avoid using obvious user names, because intruders typically use them for brute-force attacks.

3. Set a new password for this user:

```
# passwd $USER_NAME
```

#### Replace

- **\$USER\_NAME** with the new user name for the Ansible user.

#### Example

```
# passwd admin
```

Enter the new password twice when prompted.

4. Configure **sudo** access for the newly created user:

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

#### Replace

- **\$USER\_NAME** with the new user name for the Ansible user.

#### Example

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

5. Assign the correct file permissions to the new file:

```
chmod 0440 /etc/sudoers.d/$USER_NAME
```

#### Replace

- **\$USER\_NAME** with the new user name for the Ansible user.

#### Example

```
# chmod 0440 /etc/sudoers.d/admin
```

#### Additional Resources

- The [Adding a New User](#) section in the *System Administrator's Guide* for Red Hat Enterprise Linux 7.

### 1.1.3. Enabling Password-less SSH for Ansible

Generate an SSH key pair on the Ansible administration node and distribute the public key to each node in the storage cluster so that Ansible can access the nodes without being prompted for a password.

#### Prerequisites

- Create an Ansible user with **sudo** access.

### Procedure

Do the following steps from the Ansible administration node, and as the Ansible user.

1. Generate the SSH key pair, accept the default file name and leave the passphrase empty:

```
[user@admin ~]$ ssh-keygen
```

2. Copy the public key to all nodes in the storage cluster:

```
ssh-copy-id $USER_NAME@$HOST_NAME
```

### Replace

- **\$USER\_NAME** with the new user name for the Ansible user.
- **\$HOST\_NAME** with the host name of the Ceph node.

### Example

```
[user@admin ~]$ ssh-copy-id ceph-admin@ceph-mon01
```

3. Create and edit the `~/.ssh/config` file.



### IMPORTANT

By creating and editing the `~/.ssh/config` file you do not have to specify the `-u $USER_NAME` option each time you execute the `ansible-playbook` command.

- a. Create the SSH **config** file:

```
[user@admin ~]$ touch ~/.ssh/config
```

- b. Open the **config** file for editing. Set the **Hostname** and **User** options for each node in the storage cluster:

```
Host node1
  Hostname $HOST_NAME
  User $USER_NAME
Host node2
  Hostname $HOST_NAME
  User $USER_NAME
...
```

### Replace

- **\$HOST\_NAME** with the host name of the Ceph node.
- **\$USER\_NAME** with the new user name for the Ansible user.

### Example

```

Host node1
  Hostname monitor
  User admin
Host node2
  Hostname osd
  User admin
Host node3
  Hostname gateway
  User admin

```

4. Set the correct file permissions for the `~/.ssh/config` file:

```
[admin@admin ~]$ chmod 600 ~/.ssh/config
```

### Additional Resources

- The `ssh_config(5)` manual page
- The [OpenSSH](#) chapter in the *System Administrator's Guide* for Red Hat Enterprise Linux 7

## 1.1.4. Configuring a firewall for Red Hat Ceph Storage

Red Hat Ceph Storage (RHCS) uses the `firewalld` service.

The Monitor daemons use port **6789** for communication within the Ceph storage cluster.

On each Ceph OSD node, the OSD daemons use several ports in the range **6800-7300**:

- One for communicating with clients and monitors over the public network
- One for sending data to other OSDs over a cluster network, if available; otherwise, over the public network
- One for exchanging heartbeat packets over a cluster network, if available; otherwise, over the public network

The Ceph Manager (`ceph-mgr`) daemons use ports in range **6800-7300**. Consider colocating the `ceph-mgr` daemons with Ceph Monitors on same nodes.

The Ceph Metadata Server nodes (`ceph-mds`) use port **6800**.

The Ceph Object Gateway nodes are configured by Ansible to use port **8080** by default. However, you can change the default port, for example to port **80**.

To use the SSL/TLS service, open port **443**.

### Prerequisite

- Network hardware is connected.

### Procedure

Run the following commands as the `root` user.

1. On all RHCS nodes, start the `firewalld` service. Enable it to run on boot, and ensure that it is running:

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

- On all Monitor nodes, open port **6789** on the public network:

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6789/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6789/tcp --permanent
```

To limit access based on the source address:

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="6789" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="6789" accept" --permanent
```

### Replace

- IP\_address** with the network address of the Monitor node.
- netmask\_prefix** with the netmask in CIDR notation.

### Example

```
[root@monitor ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.11/24" port protocol="tcp" \
port="6789" accept"
```

```
[root@monitor ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.11/24" port protocol="tcp" \
port="6789" accept" --permanent
```

- On all OSD nodes, open ports **6800-7300** on the public network:

```
[root@osd ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp
[root@osd ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
```

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

- On all Ceph Manager (**ceph-mgr**) nodes (usually the same nodes as Monitor ones), open ports **6800-7300** on the public network:

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
```

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

- On all Ceph Metadata Server (**ceph-mds**) nodes, open port **6800** on the public network:

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800/tcp --permanent
```

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

6. On all Ceph Object Gateway nodes, open the relevant port or ports on the public network.
  - a. To open the default Ansible configured port of **8080**:

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=8080/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

To limit access based on the source address:

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="8080" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="8080" accept" --permanent
```

#### Replace

- **IP\_address** with the network address of the object gateway node.
- **netmask\_prefix** with the netmask in CIDR notation.

#### Example

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="8080" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="8080" accept" --permanent
```

- b. Optional. If you installed Ceph Object Gateway using Ansible and changed the default port that Ansible configures Ceph Object Gateway to use from **8080**, for example, to port **80**, open this port:

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=80/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=80/tcp --permanent
```

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

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="80" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="80" accept" --permanent
```

### Replace

- **IP\_address** with the network address of the object gateway node.
- **netmask\_prefix** with the netmask in CIDR notation.

### Example

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="80" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="80" accept" --permanent
```

- c. Optional. To use SSL/TLS, open port **443**:

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=443/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=443/tcp --permanent
```

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

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="443" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="443" accept" --permanent
```

### Replace

- **IP\_address** with the network address of the object gateway node.
- **netmask\_prefix** with the netmask in CIDR notation.

### Example

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="443" accept"
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="443" accept" --permanent
```



- For more information about public and cluster network, see [Verifying the Network Configuration for Red Hat Ceph Storage](#).
- For additional details on **firewalld**, see the [Using Firewalls](#) chapter in the Security Guide for Red Hat Enterprise Linux 7.

### 1.1.5. Using a HTTP Proxy

If the Ceph nodes are behind a HTTP/HTTPS proxy, then docker will need to be configured to access the images in the registry. Do the following procedure to configure access for docker using a HTTP/HTTPS proxy.

#### Prerequisites

- A running HTTP/HTTPS proxy

#### Procedure

1. As **root**, create a systemd directory for the docker service:

```
# mkdir /etc/systemd/system/docker.service.d/
```

2. As **root**, create the HTTP/HTTPS configuration file.

- a. For HTTP, create the **/etc/systemd/system/docker.service.d/http-proxy.conf** file and add the following lines to the file:

```
[Service]
Environment="HTTP_PROXY=http://proxy.example.com:80/"
```

- b. For HTTPS, create the **/etc/systemd/system/docker.service.d/https-proxy.conf** file and add the following lines to the file:

```
[Service]
Environment="HTTPS_PROXY=https://proxy.example.com:443/"
```

3. As **root**, copy the HTTP/HTTPS configuration file to all Ceph nodes in the storage cluster before running the **ceph-ansible** playbook.

## 1.2. INSTALLING A RED HAT CEPH STORAGE CLUSTER IN CONTAINERS

Use the Ansible application with the **ceph-ansible** playbook to install Red Hat Ceph Storage 3 in containers.

A Ceph cluster used in production usually consists of ten or more nodes. To deploy Red Hat Ceph Storage as a container image, Red Hat recommends to use a Ceph cluster that consists of at least three OSD and three Monitor nodes.



#### IMPORTANT

Ceph can run with one monitor; however, to ensure high availability in a production cluster, Red Hat will only support deployments with at least three monitor nodes.

## Prerequisites

- Using the root user account on the Ansible administration node, enable the Red Hat Ceph Storage 3 Tools repository and Ansible repository:

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-rpms --enable=rhel-7-server-ansible-2.6-rpms
```

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

```
[root@admin ~]# yum install ceph-ansible
```

## Procedure

Run the following commands from the Ansible administration node unless instructed otherwise.

- As the Ansible user, create the **ceph-ansible-keys** directory where Ansible stores temporary values generated by the **ceph-ansible** playbook.

```
[user@admin ~]$ mkdir ~/ceph-ansible-keys
```

- As root, create a symbolic link to the **/usr/share/ceph-ansible/group\_vars** directory in the **/etc/ansible/** directory:

```
[root@admin ~]# ln -s /usr/share/ceph-ansible/group_vars /etc/ansible/group_vars
```

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

```
[root@admin ~]$ cd /usr/share/ceph-ansible
```

- Create new copies of the **yml.sample** files:

```
[root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
[root@admin ceph-ansible]# cp site-docker.yml.sample site-docker.yml
```

- Edit the copied files.

- Edit the **group\_vars/all.yml** file. See the table below for the most common required and optional parameters to uncomment. Note that the table does not include all parameters.

**Table 1.1. General Ansible Settings**

Option	Value	Required	Notes
<b>monitor_interface</b>	The interface that the Monitor nodes listen to	<b>monitor_interface</b> , <b>monitor_address</b> , or <b>monitor_address_block</b> is required	
<b>monitor_address</b>	The address that the Monitor nodes listen to		

Option	Value	Required	Notes
<b>monitor_address_block</b>	The subnet of the Ceph public network		Use when the IP addresses of the nodes are unknown, but the subnet is known
<b>ip_version</b>	<b>ipv6</b>	Yes if using IPv6 addressing	
<b>journal_size</b>	The required size of the journal in MB	No	
<b>public_network</b>	The IP address and netmask of the Ceph public network	Yes	The <a href="#">Verifying the Network Configuration for Red Hat Ceph Storage</a> section in the <i>Installation Guide for Red Hat Enterprise Linux</i>
<b>cluster_network</b>	The IP address and netmask of the Ceph cluster network	No	
<b>ceph_docker_image</b>	<b>rhceph/rhceph-3-rhel7</b> , or <b>cephimageinlocalreg</b> if using a local Docker registry	Yes	
<b>containerized_deployment</b>	<b>true</b>	Yes	
<b>ceph_docker_registry</b>	<b>registry.access.redhat.com</b> , or <b>&lt;local-host-fqdn&gt;</b> if using a local Docker registry	Yes	

An example of the **all.yml** file can look like:

```
monitor_interface: eth0
journal_size: 5120
monitor_interface: eth0
public_network: 192.168.0.0/24
ceph_docker_image: rhceph/rhceph-3-rhel7
containerized_deployment: true
ceph_docker_registry: registry.access.redhat.com
```

For additional details, see the **all.yml** file.

- b. Edit the **group\_vars/osds.yml** file. See the table below for the most common required and optional parameters to uncomment. Note that the table does not include all parameters.



### IMPORTANT

Use a different physical device to install an OSD than the device where the operating system is installed. Sharing the same device between the operating system and OSDs causes performance issues.

Table 1.2. OSD Ansible Settings

Option	Value	Required	Notes
<b>osd_scenario</b>	<p><b>collocated</b> to use the same device for write-ahead logging and key/value data (BlueStore) or journal (FileStore) and OSD data</p> <p><b>non-collocated</b> to use a dedicated device, such as SSD or NVMe media to store write-ahead log and key/value data (BlueStore) or journal data (FileStore)</p> <p><b>lvm</b> to use the Logical Volume Manager to store OSD data</p>	Yes	When using <b>osd_scenario: non-collocated, ceph-ansible</b> expects the variables <b>devices</b> and <b>dedicated_devices</b> to match. For example, if you specify 10 disks in <b>devices</b> , you must specify 10 entries in <b>dedicated_devices</b> .
<b>osd_auto_discovery</b>	<b>true</b> to automatically discover OSDs	Yes if using <b>osd_scenario: collocated</b>	Cannot be used when <b>devices</b> setting is used
<b>devices</b>	List of devices where <b>ceph data</b> is stored	Yes to specify the list of devices	Cannot be used when <b>osd_auto_discovery</b> setting is used. When using <b>lvm</b> as the <b>osd_scenario</b> and setting the <b>devices</b> option, <b>ceph-volume lvm batch</b> mode creates the optimized OSD configuration.

Option	Value	Required	Notes
<b>dedicated_devices</b>	List of dedicated devices for non-collocated OSDs where <b>ceph journal</b> is stored	Yes if <b>osd_scenario: non-collocated</b>	Should be nonpartitioned devices
<b>dmccrypt</b>	<b>true</b> to encrypt OSDs	No	Defaults to <b>false</b>
<b>lvm_volumes</b>	A list of FileStore or BlueStore dictionaries	Yes if using <b>osd_scenario: lvm</b> and storage devices are not defined using <b>devices</b>	Each dictionary must contain a <b>data</b> , <b>journal</b> and <b>data_vg</b> keys. Any logical volume or volume group must be the name and not the full path. The <b>data</b> , and <b>journal</b> keys can be a logical volume (LV) or partition, but do not use one journal for multiple <b>data</b> LVs. The <b>data_vg</b> key must be the volume group containing the <b>data</b> LV. Optionally, the <b>journal_vg</b> key can be used to specify the volume group containing the journal LV, if applicable. See the examples below for various supported configurations.
<b>osds_per_device</b>	The number of OSDs to create per device.	No	Defaults to <b>1</b>
<b>osd_objectstore</b>	The Ceph object store type for the OSDs.	No	Defaults to <b>bluestore</b> . The other option is <b>filestore</b> . Required for upgrades.

The following are examples of the **osds.yml** file when using the three OSD scenarios: **collocated**, **non-collocated**, and **lvm**. The default OSD object store format is BlueStore, if not specified.

## Collocated

```
osd_objectstore: filestore
osd_scenario: collocated
devices:
- /dev/sda
- /dev/sdb
```

## Non-collocated - BlueStore

```
osd_objectstore: bluestore
osd_scenario: non-collocated
devices:
- /dev/sda
- /dev/sdb
- /dev/sdc
- /dev/sdd
dedicated_devices:
- /dev/nvme0n1
- /dev/nvme0n1
- /dev/nvme1n1
- /dev/nvme1n1
```

This non-collocated example will create four BlueStore OSDs, one per device. In this example, the traditional hard drives (**sda**, **sdb**, **sdc**, **sdd**) are used for object data, and the solid state drives (SSDs) (**/dev/nvme0n1**, **/dev/nvme1n1**) are used for the BlueStore databases and write-ahead logs. This configuration pairs the **/dev/sda** and **/dev/sdb** devices with the **/dev/nvme0n1** device, and pairs the **/dev/sdc** and **/dev/sdd** devices with the **/dev/nvme1n1** device.

## Non-collocated - FileStore

```
osd_objectstore: filestore
osd_scenario: non-collocated
devices:
- /dev/sda
- /dev/sdb
- /dev/sdc
- /dev/sdd
dedicated_devices:
- /dev/nvme0n1
- /dev/nvme0n1
- /dev/nvme1n1
- /dev/nvme1n1
```

## LVM simple

```
osd_objectstore: bluestore
osd_scenario: lvm
devices:
- /dev/sda
- /dev/sdb
```

or

```

osd_objectstore: bluestore
osd_scenario: lvm
devices:
- /dev/sda
- /dev/sdb
- /dev/nvme0n1

```

With these simple configurations **ceph-ansible** uses batch mode (**ceph-volume lvm batch**) to create the OSDs.

In the first scenario, if the **devices** are traditional hard drives or SSDs, then one OSD per device is created.

In the second scenario, when there is a mix of traditional hard drives and SSDs, the data is placed on the traditional hard drives (**sda, sdb**) and the BlueStore database (**block.db**) is created as large as possible on the SSD (**nvme0n1**).

### LVM advance

```

osd_objectstore: filestore
osd_scenario: lvm
lvm_volumes:
- data: data-lv1
  data_vg: vg1
  journal: journal-lv1
  journal_vg: vg2
- data: data-lv2
  journal: /dev/sda
  data_vg: vg1

```

or

```

osd_objectstore: bluestore
osd_scenario: lvm
lvm_volumes:
- data: data-lv1
  data_vg: data-vg1
  db: db-lv1
  db_vg: db-vg1
  wal: wal-lv1
  wal_vg: wal-vg1
- data: data-lv2
  data_vg: data-vg2
  db: db-lv2
  db_vg: db-vg2
  wal: wal-lv2
  wal_vg: wal-vg2

```

With these advance scenario examples, the volume groups and logical volumes must be created beforehand. They will not be created by **ceph-ansible**.

**NOTE**

If using all NVMe SSDs set the **osd\_scenario: lvm** and **osds\_per\_device: 4** options. For more information, see the [Configuring OSD Ansible settings for all NVMe Storage](#) section in the Red Hat Ceph Storage *Container Guide*.

For additional details, see the comments in the **osds.yml** file.

6. Edit the Ansible inventory file located by default at **/etc/ansible/hosts**. Remember to comment out example hosts.

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

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

- b. Add OSD nodes under the **[osds]** section. If the nodes have sequential naming, consider using a range:

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

**NOTE**

For OSDs in a new installation, the default object store format is BlueStore.

Alternatively, you can colocate Monitors with the OSD daemons on one node by adding the same node under the **[mons]** and **[osds]** sections. See [Chapter 2, Colocation of Containerized Ceph Daemons](#) for details.

Optionally, if you want **ansible-playbook** to create a custom CRUSH hierarchy, specify where you want the OSD hosts to be in the CRUSH map's hierarchy by using the **osd\_crush\_location** parameter. You must specify at least two CRUSH bucket types to specify the location of the OSD, and one bucket **type** must be **host**. By default, these include **root**, **datacenter**, **room**, **row**, **pod**, **pdu**, **rack**, **chassis** and **host**.

```
[osds]
<ceph-host-name> osd_crush_location="{ 'root': '<root-bucket>', 'rack': '<rack-bucket>',
'pod': '<pod-bucket>', 'host': '<ceph-host-name>' }"
```

For example:

```
[osds]
ceph-osd-01 osd_crush_location="{ 'root': 'mon-root', 'rack': 'mon-rack', 'pod': 'monpod',
'host': 'ceph-osd-01' }"
```

- c. Add the Ceph Manager (**ceph-mgr**) nodes under the **[mgrs]** section. Colocate the Ceph Manager daemon with Monitor nodes.

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



```
<monitor-host-name>
<monitor-host-name>
```

7. As the Ansible user, ensure that Ansible can reach the Ceph hosts:

```
[user@admin ~]$ ansible all -m ping
```

8. As **root**, create the **/var/log/ansible/** directory and assign the appropriate permissions for the **ansible** user:

```
[root@admin ~]# mkdir /var/log/ansible
[root@admin ~]# chown ansible:ansible /var/log/ansible
[root@admin ~]# chmod 755 /var/log/ansible
```

- a. Edit the **/usr/share/ceph-ansible/ansible.cfg** file, updating the **log\_path** value as follows:

```
log_path = /var/log/ansible/ansible.log
```

9. As the Ansible user, change to the **/usr/share/ceph-ansible/** directory:

```
[user@admin ~]$ cd /usr/share/ceph-ansible/
```

10. Run the **ceph-ansible** playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml
```



#### NOTE

If you deploy Red Hat Ceph Storage to Red Hat Enterprise Linux Atomic Host hosts, use the **--skip-tags=with\_pkg** option:

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --skip-
tags=with_pkg
```

11. Using the root account on a Monitor node, verify the status of the Ceph cluster:

```
docker exec ceph-<mon|mgr>-<id> ceph health
```

*Replace:*

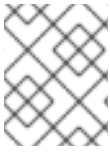
- **<id>** with the host name of the Monitor node:

For example:

```
[root@monitor ~]# docker exec ceph-mon-mon0 ceph health
HEALTH_OK
```

## 1.3. CONFIGURING OSD ANSIBLE SETTINGS FOR ALL NVME STORAGE

To optimize performance when using only non-volatile memory express (NVMe) devices for storage, configure four OSDs on each NVMe device. Normally only one OSD is configured per device, which will underutilize the throughput of an NVMe device.

**NOTE**

If you mix SSDs and HDDs, then SSDs will be used for either journals or **block.db**, not OSDs.

**NOTE**

In testing, configuring four OSDs on each NVMe device was found to provide optimal performance. It is recommended to set **osds\_per\_device: 4**, but it is not required. Other values may provide better performance in your environment.

**Prerequisites**

- Satisfying all software and hardware requirements for a Ceph cluster.

**Procedure**

1. Set **osd\_scenario: lvm** and **osds\_per\_device: 4** in **group\_vars/osds.yml**:

```
osd_scenario: lvm
osds_per_device: 4
```

2. List the NVMe devices under **devices**:

```
devices:
- /dev/nvme0n1
- /dev/nvme1n1
- /dev/nvme2n1
- /dev/nvme3n1
```

3. The settings in **group\_vars/osds.yml** will look similar to this example:

```
osd_scenario: lvm
osds_per_device: 4
devices:
- /dev/nvme0n1
- /dev/nvme1n1
- /dev/nvme2n1
- /dev/nvme3n1
```

**NOTE**

You must use **devices** with this configuration, not **lvm\_volumes**. This is because **lvm\_volumes** is generally used with pre-created logical volumes and **osds\_per\_device** implies automatic logical volume creation by Ceph.

**Additional Resources**

- [Installing a Red Hat Ceph Storage Cluster on Red Hat Enterprise Linux](#)

- [Installing a Red Hat Ceph Storage Cluster on Ubuntu](#)

## 1.4. INSTALLING THE CEPH OBJECT GATEWAY IN A CONTAINER

Use the Ansible application with the **ceph-ansible** playbook to install the Ceph Object Gateway in a container.

### Prerequisites

- A working Red Hat Ceph Storage cluster.

### Procedure

Run the following commands from the Ansible administration node unless specified otherwise.

1. As the **root** user, navigate to the `/usr/share/ceph-ansible/` directory.

```
[root@admin ~]# cd /usr/share/ceph-ansible/
```

2. Uncomment the **radosgw\_interface** parameter in the **group\_vars/all.yml** file.

```
radosgw_interface: interface
```

Replace *interface* with the interface that the Ceph Object Gateway nodes listen to.

3. Optional. Change the default variables.
  - a. Create a new copy of the **rgws.yml.sample** file located in the **group\_vars** directory.

```
[root@admin ceph-ansible]# cp group_vars/rgws.yml.sample group_vars/rgws.yml
```

- b. Edit the **group\_vars/rgws.yml** file. For additional details, see the **rgws.yml** file.

4. Add the host name of the Ceph Object Gateway node to the **[rgws]** section of the Ansible inventory file located by default at **/etc/ansible/hosts**.

```
[rgws]
gateway01
```

Alternatively, you can colocate the Ceph Object Gateway with the OSD daemon on one node by adding the same node under the **[osds]** and **[rgws]** sections. See [Colocation of containerized Ceph daemons](#) for details.

5. As the Ansible user, run the **ceph-ansible** playbook.

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit rgws
```

**NOTE**

If you deploy Red Hat Ceph Storage to Red Hat Enterprise Linux Atomic Host hosts, use the **--skip-tags=with\_pkg** option:

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --skip-tags=with_pkg
```

6. Verify that the Ceph Object Gateway node was deployed successfully.

a. Connect to a Monitor node as the **root** user:

```
ssh hostname
```

Replace *hostname* with the host name of the Monitor node, for example:

```
[user@admin ~]$ ssh root@monitor
```

b. Verify that the Ceph Object Gateway pools were created properly:

```
[root@monitor ~]# docker exec ceph-mon-mon1 rados lspools
rbd
cephfs_data
cephfs_metadata
.rgw.root
default.rgw.control
default.rgw.data.root
default.rgw.gc
default.rgw.log
default.rgw.users.uid
```

c. From any client on the same network as the Ceph cluster, for example the Monitor node, use the **curl** command to send an HTTP request on port 8080 using the IP address of the Ceph Object Gateway host:

```
curl http://IP-address:8080
```

Replace *IP-address* with the IP address of the Ceph Object Gateway node. To determine the IP address of the Ceph Object Gateway host, use the **ifconfig** or **ip** commands:

```
[root@client ~]# curl http://192.168.122.199:8080
<?xml version="1.0" encoding="UTF-8"?><ListAllMyBucketsResult
xmlns="http://s3.amazonaws.com/doc/2006-03-01/"><Owner><ID>anonymous</ID>
<DisplayName></DisplayName></Owner><Buckets></Buckets>
</ListAllMyBucketsResult>
```

d. List buckets:

```
[root@monitor ~]# docker exec ceph-mon-mon1 radosgw-admin bucket list
```

**Additional Resources**

- The Red Hat Ceph Storage 3 [Ceph Object Gateway Guide for Red Hat Enterprise Linux](#)

- [Understanding the \*limit\* option](#)

## 1.5. INSTALLING METADATA SERVERS

Use the Ansible automation application to install a Ceph Metadata Server (MDS). Metadata Server daemons are necessary for deploying a Ceph File System.

### Prerequisites

- A working Red Hat Ceph Storage cluster.

### Procedure

Perform the following steps on the Ansible administration node.

1. Add a new section **[mdss]** to the **/etc/ansible/hosts** file:

```
[mdss]
hostname
hostname
hostname
```

Replace *hostname* with the host names of the nodes where you want to install the Ceph Metadata Servers.

Alternatively, you can colocate the Metadata Server with the OSD daemon on one node by adding the same node under the **[osds]** and **[mdss]** sections. See [Colocation of containerized Ceph daemons](#) for details.

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

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

3. Optional. Change the default variables.

- a. Create a copy of the **group\_vars/mdss.yml.sample** file named **mdss.yml**:

```
[root@admin ceph-ansible]# cp group_vars/mdss.yml.sample group_vars/mdss.yml
```

- b. Optionally, edit parameters in **mdss.yml**. See **mdss.yml** for details.

4. As the Ansible user, run the Ansible playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit mdss
```

5. After installing Metadata Servers, configure them. For details, see the [Configuring Metadata Server Daemons](#) chapter in the Ceph File System Guide for Red Hat Ceph Storage 3.

### Additional Resources

- The [Ceph File System Guide](#) for Red Hat Ceph Storage 3
- [Understanding the \*limit\* option](#)

## 1.6. INSTALLING THE CEPH ISCSI GATEWAY IN A CONTAINER

The Ansible deployment application installs the required daemons and tools to configure a Ceph iSCSI gateway in a container.

### Prerequisites

- A working Red Hat Ceph Storage cluster.

### Procedure

1. As the root user, open and edit the `/etc/ansible/hosts` file. Add a node name entry in the iSCSI gateway group:

#### Example

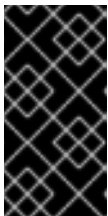
```
[iscsigws]
ceph-igw-1
ceph-igw-2
```

2. Navigate to the `/usr/share/ceph-ansible` directory:

```
[root@admin ~]# cd /usr/share/ceph-ansible/
```

3. Create a copy of the `iscsigws.yml.sample` file and name it `iscsigws.yml`:

```
[root@admin ceph-ansible]# cp group_vars/iscsigws.yml.sample group_vars/iscsigws.yml
```



#### IMPORTANT

The new file name (`iscsigws.yml`) and the new section heading (`[iscsigws]`) are only applicable to Red Hat Ceph Storage 3.1 or higher. Upgrading from previous versions of Red Hat Ceph Storage to 3.1 will still use the old file name (`iscsi-gws.yml`) and the old section heading (`[iscsi-gws]`).



#### IMPORTANT

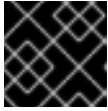
Currently, Red Hat does not support the following options for container-based deployments:

- `gateway_iqn`
- `rbd_devices`
- `client_connections`

4. Open the `iscsigws.yml` file for editing.
5. Configure the `gateway_ip_list` option by adding the iSCSI gateway IP addresses, using IPv4 or IPv6 addresses:

#### Example

```
gateway_ip_list: 192.168.1.1,192.168.1.2
```



### IMPORTANT

You cannot use a mix of IPv4 and IPv6 addresses.

6. Optionally, uncomment the **trusted\_ip\_list** option and add the IPv4 or IPv6 addresses accordingly, if you want to use SSL. You will need **root** access to the iSCSI gateway containers to configure SSL. To configure SSL, do the following steps:

- a. If needed, install the **openssl** package within all the iSCSI gateway containers.
- b. On the primary iSCSI gateway container, create a directory to hold the SSL keys:

```
# mkdir ~/ssl-keys
# cd ~/ssl-keys
```

- c. On the primary iSCSI gateway container, create the certificate and key files:

```
# openssl req -newkey rsa:2048 -nodes -keyout iscsi-gateway.key -x509 -days 365 -out
iscsi-gateway.crt
```



### NOTE

You will be prompted to enter the environmental information.

- d. On the primary iSCSI gateway container, create a PEM file:

```
# cat iscsi-gateway.crt iscsi-gateway.key > iscsi-gateway.pem
```

- e. On the primary iSCSI gateway container, create a public key:

```
# openssl x509 -inform pem -in iscsi-gateway.pem -pubkey -noout > iscsi-gateway-
pub.key
```

- f. From the primary iSCSI gateway container, copy the **iscsi-gateway.crt**, **iscsi-gateway.pem**, **iscsi-gateway-pub.key**, and **iscsi-gateway.key** files to the **/etc/ceph/** directory on the other iSCSI gateway containers.

7. Optionally, review and uncomment any of the following iSCSI target API service options accordingly:

```
#api_user: admin
#api_password: admin
#api_port: 5000
#api_secure: false
#loop_delay: 1
#trusted_ip_list: 192.168.122.1,192.168.122.2
```

8. Optionally, review and uncomment any of the following resource options, updating them according to the workload needs:

```
# TCMU_RUNNER resource limitation
#ceph_tcmu_runner_docker_memory_limit: 1g
#ceph_tcmu_runner_docker_cpu_limit: 1

# RBD_TARGET_GW resource limitation
#ceph_rbd_target_gw_docker_memory_limit: 1g
#ceph_rbd_target_gw_docker_cpu_limit: 1

# RBD_TARGET_API resource limitation
#ceph_rbd_target_api_docker_memory_limit: 1g
#ceph_rbd_target_api_docker_cpu_limit: 1
```

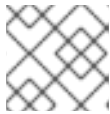
- As the Ansible user, run the Ansible playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit iscsigws
```

For Red Hat Enterprise Linux Atomic, add the `--skip-tags=with_pkg` option:

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit iscsigws --skip-
tags=with_pkg
```

- Once the Ansible playbook has finished, open TCP ports **3260** and the **api\_port** specified in the **iscsigws.yml** file on each node listed in the **trusted\_ip\_list** option.



#### NOTE

If the **api\_port** option is not specified, the default port is **5000**.

#### Additional Resources

- For more information on installing Red Hat Ceph Storage in a container, see the [Installing a Red Hat Ceph Storage cluster in containers](#) section.
- For more information on Ceph's iSCSI gateway options, see [Table 8.1](#) in the Red Hat Ceph Storage Block Device Guide.
- For more information on the iSCSI target API options, see [Table 8.2](#) in the Red Hat Ceph Storage Block Device Guide.
- For an example of the **iscsigws.yml** file, see [Appendix A](#) the Red Hat Ceph Storage Block Device Guide.

### 1.6.1. Configuring the Ceph iSCSI gateway in a container

The Ceph iSCSI gateway configuration is done with the **gwcli** command-line utility for creating and managing iSCSI targets, Logical Unit Numbers (LUNs) and Access Control Lists (ACLs).

#### Prerequisites

- A working Red Hat Ceph Storage cluster.
- Installation of the iSCSI gateway software.



## Procedure

1. As the **root** user, start the iSCSI gateway command-line interface:

```
# docker exec -it rbd-target-api gwcli
```

2. Create the iSCSI gateways using either IPv4 or IPv6 addresses:

### Syntax

```
>/iscsi-target create iqn.2003-01.com.redhat.iscsi-gw:$TARGET_NAME
> goto gateways
> create $ISCSI_GW_NAME $ISCSI_GW_IP_ADDR
> create $ISCSI_GW_NAME $ISCSI_GW_IP_ADDR
```

### Example

```
>/iscsi-target create iqn.2003-01.com.redhat.iscsi-gw:ceph-igw
> goto gateways
> create ceph-gw-1 10.172.19.21
> create ceph-gw-2 10.172.19.22
```



### IMPORTANT

You cannot use a mix of IPv4 and IPv6 addresses.

3. Add a RADOS Block Device (RBD):

### Syntax

```
> cd /disks
>/disks/ create $POOL_NAME image=$IMAGE_NAME size=$IMAGE_SIZE[m|g|t]
max_data_area_mb=$BUFFER_SIZE
```

### Example

```
> cd /disks
>/disks/ create rbd image=disk_1 size=50g max_data_area_mb=32
```



### IMPORTANT

There can not be any periods (.) in the pool name or in the image name.



## WARNING

Do NOT adjust the **max\_data\_area\_mb** option, unless Red Hat Support has instructed you to do so.

The **max\_data\_area\_mb** option controls the amount of memory in megabytes that each image can use to pass SCSI command data between the iSCSI target and the Ceph cluster. If this value is too small, then it can result in excessive queue full retries which will affect performance. If the value is too large, then it can result in one disk using too much of the system's memory, which can cause allocation failures for other subsystems. The default value is 8.

This value can be changed using the **reconfigure** command. The image must not be in use by an iSCSI initiator for this command to take effect.

### Syntax

```
>/disks/ reconfigure max_data_area_mb $NEW_BUFFER_SIZE
```

### Example

```
>/disks/ reconfigure max_data_area_mb 64
```

4. Create a client:

### Syntax

```
> goto hosts  
> create iqn.1994-05.com.redhat:$CLIENT_NAME  
> auth chap=$USER_NAME/$PASSWORD
```

### Example

```
> goto hosts  
> create iqn.1994-05.com.redhat:rh7-client  
> auth chap=iscsiuser1/temp12345678
```

**IMPORTANT**

Disabling CHAP is only supported on Red Hat Ceph Storage 3.1 or higher. Red Hat does not support mixing clients, some with CHAP enabled and some CHAP disabled. All clients must have either CHAP enabled or have CHAP disabled. The default behavior is to only authenticate an initiator by its initiator name.

If initiators are failing to log into the target, then the CHAP authentication might be a misconfigured for some initiators.

**Example**

```
o- hosts ..... [Hosts: 2: Auth: MISCONFIG]
```

Do the following command at the **hosts** level to reset all the CHAP authentication:

```
/> goto hosts
/iscsi-target...csi-igw/hosts> auth nochap
ok
ok
/iscsi-target...csi-igw/hosts> ls
o- hosts ..... [Hosts: 2: Auth: None]
o- iqn.2005-03.com.ceph:esx ..... [Auth: None, Disks: 4(310G)]
o- iqn.1994-05.com.redhat:rh7-client .. [Auth: None, Disks: 0(0.00Y)]
```

5. Add disks to a client:

**Syntax**

```
>/iscsi-target..eph-igw/hosts> cd iqn.1994-05.com.redhat:$CLIENT_NAME
> disk add $POOL_NAME.$IMAGE_NAME
```

**Example**

```
>/iscsi-target..eph-igw/hosts> cd iqn.1994-05.com.redhat:rh7-client
> disk add rbd.disk_1
```

6. Run the following command to verify the iSCSI gateway configuration:

```
> ls
```

7. Optionally, confirm that the API is using SSL correctly, look in the **/var/log/rbd-target-api.log** file for **https**, for example:

```
Aug 01 17:27:42 test-node.example.com python[1879]: * Running on https://0.0.0.0:5000/
```

8. The next step is to configure an iSCSI initiator.

**Additional Resources**

- For more information on installing Red Hat Ceph Storage in a container, see the [Installing a Red Hat Ceph Storage cluster in containers](#) section.
- For more information on installing the iSCSI gateway software in a container, see the [Installing the Ceph iSCSI gateway in a container](#) section.
- For more information on connecting an iSCSI initiator, see the [Configuring the iSCSI Initiator](#) section in the *Red Hat Ceph Storage Block Device Guide*.

## 1.6.2. Removing the Ceph iSCSI gateway in a container

The Ceph iSCSI gateway configuration can be removed using **gwcli**, the iSCSI gateway command line utility.

### Prerequisites

- A working Red Hat Ceph Storage cluster.
- Installation of the iSCSI gateway software.
- Exported RBD images.

### Procedure

1. All iSCSI initiators need to be disconnected before purging the iSCSI gateway configuration. Follow the procedures below for the appropriate operating system:

#### Red Hat Enterprise Linux initiators:

Run the following command as the **root** user:

```
iscsiadm -m node -T $TARGET_NAME --logout
```

Replace **\$TARGET\_NAME** with the configured iSCSI target name.

#### Example output

```
# iscsiadm -m node -T iqn.2003-01.com.redhat.iscsi-gw:ceph-igw --logout
Logging out of session [sid: 1, target: iqn.2003-01.com.redhat.iscsi-gw:iscsi-igw, portal:
10.172.19.21,3260]
Logging out of session [sid: 2, target: iqn.2003-01.com.redhat.iscsi-gw:iscsi-igw, portal:
10.172.19.22,3260]
Logout of [sid: 1, target: iqn.2003-01.com.redhat.iscsi-gw:iscsi-igw, portal:
10.172.19.21,3260] successful.
Logout of [sid: 2, target: iqn.2003-01.com.redhat.iscsi-gw:iscsi-igw, portal:
10.172.19.22,3260] successful.
```

#### Windows initiators:

See the [Microsoft documentation](#) for more details.

#### VMware ESXi initiators:

See the [VMware documentation](#) for more details.

2. As the **root** user, run the iSCSI gateway command line utility:

```
# gwcli
```

- Remove the hosts:

```
/> cd /iscsi-target/iqn.2003-01.com.redhat.iscsi-gw:$TARGET_NAME/hosts
/> /iscsi-target...$TARGET_NAME/hosts> delete $CLIENT_NAME
```

Replace **\$TARGET\_NAME** with the configured iSCSI target name, and replace **\$CLIENT\_NAME** with iSCSI initiator name.

### Example

```
/> cd /iscsi-target/iqn.2003-01.com.redhat.iscsi-gw:ceph-igw/hosts
/> /iscsi-target...eph-igw/hosts> delete iqn.1994-05.com.redhat:rh7-client
```

- Remove the disks:

```
/> cd /disks/
/disks> delete $POOL_NAME.$IMAGE_NAME
```

Replace **\$POOL\_NAME** with the name of the pool, and replace the **\$IMAGE\_NAME** with the name of the image.

```
/> cd /disks/
/disks> delete rbd.disk_1
```

- Remove the iSCSI target and gateway configuration:

```
/> cd /iscsi-target/
/iscsi-target> clearconfig confirm=true
```

### Additional Resources

- For more information on installing Red Hat Ceph Storage in a container, see the [Installing a Red Hat Ceph Storage cluster in containers](#) section.
- For more information on installing the iSCSI gateway software in a container, see the [Installing the Ceph iSCSI gateway in a container](#) section.

### 1.6.3. Optimizing the performance of the iSCSI Target

There are many settings that control how the iSCSI Target transfers data over the network. These settings can be used to optimize the performance of the iSCSI gateway.



#### WARNING

Only change these settings if instructed to by Red Hat Support or as specified in this document.

## The gwcli reconfigure subcommand

The **gwcli reconfigure** subcommand controls the settings that are used to optimize the performance of the iSCSI gateway.

### Settings that affect the performance of the iSCSI target

- `max_data_area_mb`
- `cmds_n_depth`
- `immediate_data`
- `initial_r2t`
- `max_outstanding_r2t`
- `first_burst_length`
- `max_burst_length`
- `max_recv_data_segment_length`
- `max_xmit_data_segment_length`

### Additional Resources

- Information about **max\_data\_area\_mb**, including an example showing how to adjust it using **gwcli reconfigure**, is in the section [Configuring the iSCSI Target using the Command Line Interface](#) for the *Block Device Guide*, and [Configuring the Ceph iSCSI gateway in a container](#) for the *Container Guide*.

## 1.7. UNDERSTANDING THE LIMIT OPTION

This section contains information about the Ansible **--limit** option.

Ansible supports the **--limit** option that enables you to use the **site**, **site-docker**, and **rolling\_upgrade** Ansible playbooks for a particular section of the inventory file.

```
$ ansible-playbook site.yml|rolling_upgrade.yml|site-docker.yml --limit
osds|rgws|clients|mdss|nfss|iscsigws
```

For example, to redeploy only OSDs on containers, run the following command as the Ansible user:

```
$ ansible-playbook /usr/share/ceph-ansible/site-docker.yml --limit osds
```



### IMPORTANT

If you colocate Ceph components on one node, Ansible applies a playbook to all components on the node despite that only one component type was specified with the **limit** option. For example, if you run the **rolling\_update** playbook with the **--limit osds** option on a node that contains OSDs and Metadata Servers (MDS), Ansible will upgrade both components, OSDs and MDSs.

## 1.8. ADDITIONAL RESOURCES

- The [Getting Started with Containers](#) guide for Red Hat Enterprise Linux Atomic Host

## CHAPTER 2. COLOCATION OF CONTAINERIZED CEPH DAEMONS

This section describes:

- [How colocation works and its advantages](#)
- [How to set dedicated resources for colocated daemons](#)

### 2.1. HOW COLOCATION WORKS AND ITS ADVANTAGES

You can colocate containerized Ceph daemons on the same node. Here are the advantages of collocating some of Ceph's services:

- Significant improvement in total cost of ownership (TCO) at small scale
- Reduction from six nodes to three for the minimum configuration
- Easier upgrade
- Better resource isolation

#### How Colocation Works

You can colocate one daemon from the following list with an OSD daemon by adding the same node to appropriate sections in the Ansible inventory file.

- The Ceph Object Gateway (**radosgw**)
- Metadata Server (MDS)
- RBD mirror (**rbd-mirror**)
- Monitor and the Ceph Manager daemon (**ceph-mgr**)
- NFS Ganesha

The following example shows how the inventory file with colocated daemons can look like:

#### Example 2.1. Ansible inventory file with colocated daemons

```
[mons]
<hostname1>
<hostname2>
<hostname3>

[mgrs]
<hostname1>
<hostname2>
<hostname3>

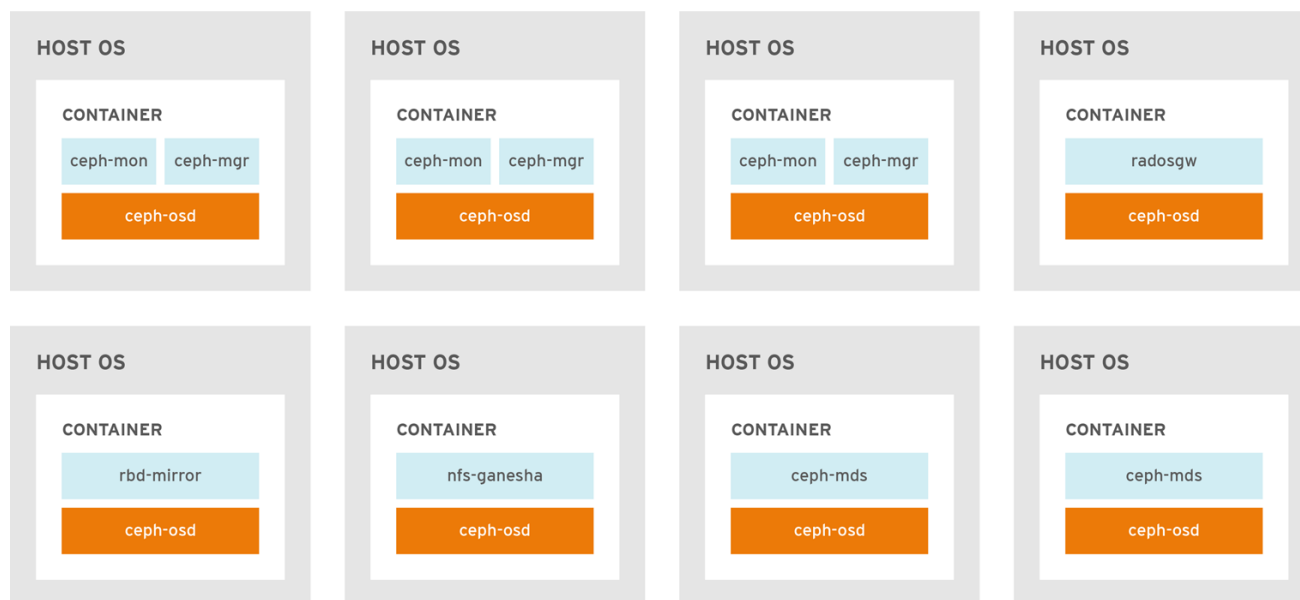
[osds]
<hostname4>
<hostname5>
<hostname6>
```



```
[rgws]
<hostname4>
<hostname5>
```

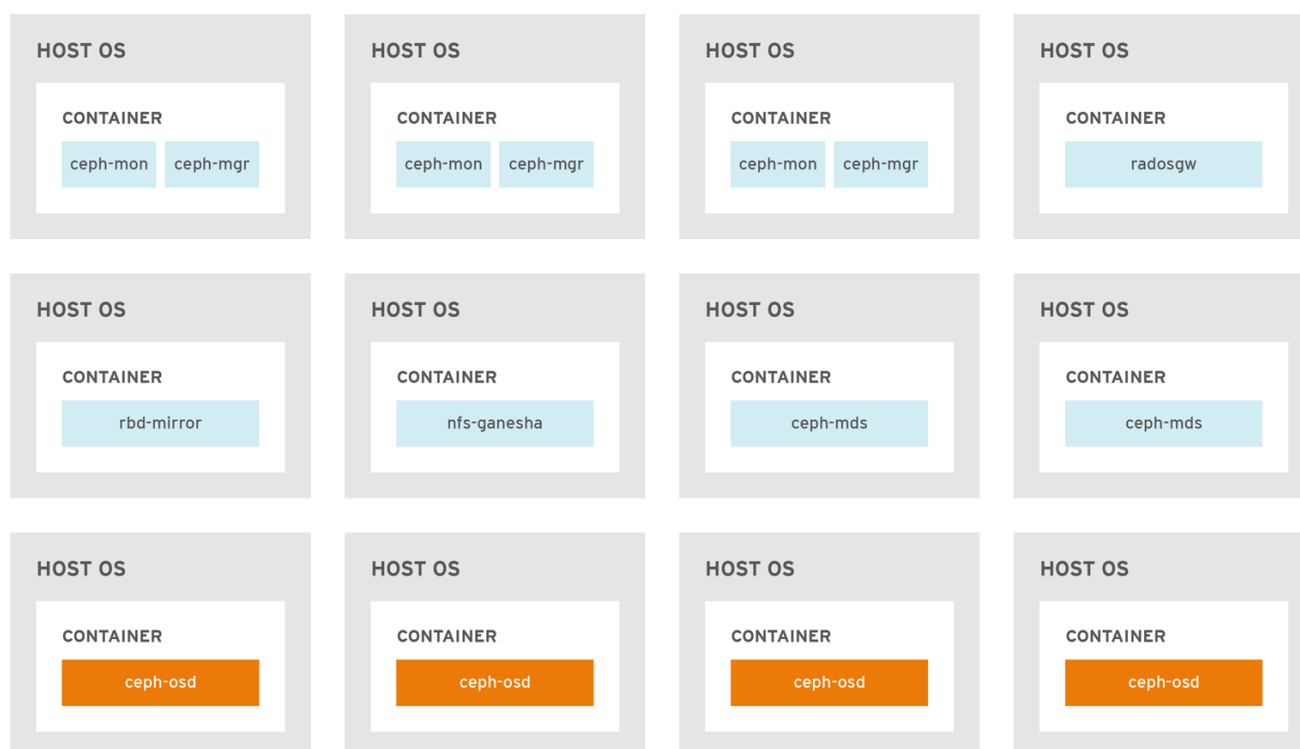
The [Figure 2.1, “Colocated Daemons”](#) and [Figure 2.2, “Non-colocated Daemons”](#) images shows the difference between clusters with colocated and non-colocated daemons.

**Figure 2.1. Colocated Daemons**



CEPH\_459072\_1017

**Figure 2.2. Non-colocated Daemons**



CEPH\_459072\_1017

When you colocate two containerized Ceph daemons on a same node, the **ceph-ansible** playbook reserves dedicated CPU and RAM resources to each. By default, **ceph-ansible** uses values listed in the

[Recommended Minimum Hardware](#) chapter in the Red Hat Ceph Storage Hardware Selection Guide 3. To learn how to change the default values, see the [Setting Dedicated Resources for Colocated Daemons](#) section.

## 2.2. SETTING DEDICATED RESOURCES FOR COLOCATED DAEMONS

When colocating two Ceph daemons on the same node, the **ceph-ansible** playbook reserves CPU and RAM resources for each daemon. The default values that **ceph-ansible** uses are listed in the [Recommended Minimum Hardware](#) chapter in the Red Hat Ceph Storage Hardware Selection Guide. To change the default values, set the needed parameters when deploying Ceph daemons.

### Procedure

1. To change the default CPU limit for a daemon, set the **ceph\_daemon-type\_docker\_cpu\_limit** parameter in the appropriate **.yml** configuration file when deploying the daemon. See the following table for details.

Daemon	Parameter	Configuration file
OSD	<b>ceph_osd_docker_cpu_limit</b>	<b>osds.yml</b>
MDS	<b>ceph_mds_docker_cpu_limit</b>	<b>mdss.yml</b>
RGW	<b>ceph_rgw_docker_cpu_limit</b>	<b>rgws.yml</b>

For example, to change the default CPU limit to 2 for the Ceph Object Gateway, edit the **/usr/share/ceph-ansible/group\_vars/rgws.yml** file as follows:

```
ceph_rgw_docker_cpu_limit: 2
```

2. To change the default RAM for OSD daemons, set the **osd\_memory\_target** in the **/usr/share/ceph-ansible/group\_vars/all.yml** file when deploying the daemon. For example, to limit the OSD RAM to 6 GB:

```
ceph_conf_overrides:
  osd:
    osd_memory_target=6000000000
```

### IMPORTANT

In a hyperconverged infrastructure (HCI) configuration, you can also use the **ceph\_osd\_docker\_memory\_limit** parameter in the **osds.yml** configuration file to change the Docker memory CGroup limit. In this case, set **ceph\_osd\_docker\_memory\_limit** to 50% higher than **osd\_memory\_target**, so that the CGroup limit is more constraining than it is by default for an HCI configuration. For example, if **osd\_memory\_target** is set to 6 GB, set **ceph\_osd\_docker\_memory\_limit** to 9 GB:

```
ceph_osd_docker_memory_limit: 9g
```

### Additional Resources

- The sample configuration files in the `/usr/share/ceph-ansible/group_vars/` directory

## 2.3. ADDITIONAL RESOURCES

- [Deploying Red Hat Ceph Storage in containers](#)
- The [Red Hat Ceph Storage Hardware Selection Guide](#)

## CHAPTER 3. ADMINISTERING CEPH CLUSTERS THAT RUN IN CONTAINERS

This chapter describes basic administration tasks to perform on Ceph clusters that run in containers, such as:

- [Section 3.1, “Starting, Stopping, and Restarting Ceph Daemons That Run in Containers”](#)
- [Section 3.2, “Viewing Log Files of Ceph Daemons That Run in Containers”](#)
- [Section 3.3, “Purging Clusters Deployed by Ansible”](#)
- [Section 4.2, “Upgrading a Red Hat Ceph Storage Cluster That Runs in Containers”](#)

### 3.1. STARTING, STOPPING, AND RESTARTING CEPH DAEMONS THAT RUN IN CONTAINERS

Use the **systemctl** command `start`, `stop`, or `restart` Ceph daemons that run in containers.

#### Procedure

1. To start, stop, or restart a Ceph daemon running in a container, run a **systemctl** command as **root** composed in the following format:

```
systemctl action ceph-daemon@ID
```

Where:

- *action* is the action to perform; **start**, **stop**, or **restart**
- *daemon* is the daemon; **osd**, **mon**, **mids**, or **rgw**
- *ID* is either
  - The short host name where the **ceph-mon**, **ceph-mids**, or **ceph-rgw** daemons are running
  - The ID of the **ceph-osd** daemon if it was deployed the **osd\_scenario** parameter set to **lvm**
  - The device name that the **ceph-osd** daemon uses if it was deployed with the **osd\_scenario** parameter set to **collocated** or **non-collocated**

For example, to restart a **ceph-osd** daemon with the ID **osd01**:

```
# systemctl restart ceph-osd@osd01
```

To start a **ceph-mon** demon that runs on the **ceph-monitor01** host:

```
# systemctl start ceph-mon@ceph-monitor01
```

To stop a **ceph-rgw** daemon that runs on the **ceph-rgw01** host:

```
# systemctl stop ceph-radosgw@ceph-rgw01
```

- 2. Verify that the action was completed successfully.

```
systemctl status ceph-daemon@_ID
```

For example:

```
# systemctl status ceph-mon@ceph-monitor01
```

### Additional Resources

- The [Running Ceph as a systemd Service](#) section in the *Administration Guide* for Red Hat Ceph Storage 3.

## 3.2. VIEWING LOG FILES OF CEPH DAEMONS THAT RUN IN CONTAINERS

Use the **journald** daemon from the container host to view a log file of a Ceph daemon from a container.

### Procedure

1. To view the entire Ceph log file, run a **journalctl** command as **root** composed in the following format:

```
journalctl -u ceph-daemon@ID
```

Where:

- *daemon* is the Ceph daemon; **osd**, **mon**, or **rgw**
- *ID* is either
  - The short host name where the **ceph-mon**, **ceph-mds**, or **ceph-rgw** daemons are running
  - The ID of the **ceph-osd** daemon if it was deployed the **osd\_scenario** parameter set to **lvm**
  - The device name that the **ceph-osd** daemon uses if it was deployed with the **osd\_scenario** parameter set to **collocated** or **non-collocated**

For example, to view the entire log for the **ceph-osd** daemon with the ID **osd01**:

```
# journalctl -u ceph-osd@osd01
```

2. To show only the recent journal entries, use the **-f** option.

```
journalctl -fu ceph-daemon@ID
```

For example, to view only recent journal entries for the **ceph-mon** daemon that runs on the **ceph-monitor01** host:

```
# journalctl -fu ceph-mon@ceph-monitor01
```



## NOTE

You can also use the **sosreport** utility to view the **journald** logs. For more details about SOS reports, see the [What is a sosreport and how to create one in Red Hat Enterprise Linux 4.6 and later?](#) solution on the Red Hat Customer Portal.

### Additional Resources

- The **journalctl(1)** manual page

## 3.3. PURGING CLUSTERS DEPLOYED BY ANSIBLE

If you no longer want to use a Ceph cluster, use the **purge-docker-cluster.yml** playbook to purge the cluster. Purging a cluster is also useful when the installation process failed and you want to start over.



### WARNING

After purging a Ceph cluster, all data on the OSDs are lost.

### Prerequisites

- Ensure that the **/var/log/ansible.log** file is writable.

### Procedure

Use the following commands from the Ansible administration node.

1. As the **root** user, navigate to the **/usr/share/ceph-ansible/** directory.

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

2. Copy the **purge-docker-cluster.yml** playbook from the **/usr/share/infrastructure-playbooks/** directory to the current directory:

```
[root@admin ceph-ansible]# cp infrastructure-playbooks/purge-docker-cluster.yml .
```

3. As the Ansible user, use the **purge-docker-cluster.yml** playbook to purge the Ceph cluster.

- a. To remove all packages, containers, configuration files, and all the data created by the **ceph-ansible** playbook:

```
[user@admin ceph-ansible]$ ansible-playbook purge-docker-cluster.yml
```

- b. To specify a different inventory file than the default one (**/etc/ansible/hosts**), use **-i** parameter:

```
ansible-playbook purge-docker-cluster.yml -i inventory-file
```

Replace *inventory-file* with the path to the inventory file.

For example:

```
[user@admin ceph-ansible]$ ansible-playbook purge-docker-cluster.yml -i  
~/ansible/hosts
```

- c. To skip the removal of the Ceph container image, use the **--skip-tags="remove\_img"** option:

```
[user@admin ceph-ansible]$ ansible-playbook --skip-tags="remove_img" purge-docker-  
cluster.yml
```

- d. To skip the removal of the packages that were installed during the installation, use the **--skip-tags="with\_pkg"** option:

```
[user@admin ceph-ansible]$ ansible-playbook --skip-tags="with_pkg" purge-docker-  
cluster.yml
```

## CHAPTER 4. UPGRADING RED HAT CEPH STORAGE WITHIN CONTAINERS

The Ansible application preforms the upgrade of Red Hat Ceph Storage running within containers.

### 4.1. PREREQUISITES

- A running Red Hat Ceph Storage cluster.

### 4.2. UPGRADING A RED HAT CEPH STORAGE CLUSTER THAT RUNS IN CONTAINERS

This section describes how to upgrade to a newer minor or major version of the Red Hat Ceph Storage container image.

- To upgrade a storage cluster, see [Section 4.3, “Upgrading the Storage Cluster”](#).
- To upgrade Red Hat Ceph Storage Dashboard, see [Section 4.4, “Upgrading Red Hat Ceph Storage Dashboard”](#).

Use the Ansible **rolling\_update.yml** playbook located in the **/usr/share/ceph-ansible/infrastructure-playbooks/** directory from the administration node to upgrade between two major or minor versions of Red Hat Ceph Storage, or to apply asynchronous updates.

Ansible upgrades the Ceph nodes in the following order:

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



#### NOTE

Red Hat Ceph Storage 3 introduces several changes in Ansible configuration files located in the **/usr/share/ceph-ansible/group\_vars/** directory; certain parameters were renamed or removed. Therefore, make backup copies of the **all.yml** and **osds.yml** files before creating new copies from the **all.yml.sample** and **osds.yml.sample** files after upgrading to version 3. For more details about the changes, see [Appendix A, \*Changes in Ansible Variables Between Version 2 and 3\*](#).





## NOTE

Red Hat Ceph Storage 3.1 and later introduces new Ansible playbooks to optimize storage for performance when using Object Gateway and high speed NVMe based SSDs (and SATA SSDs). The playbooks do this by placing journals and bucket indexes together on SSDs, which can increase performance compared to having all journals on one device. These playbooks are designed to be used when installing Ceph. Existing OSDs continue to work and need no extra steps during an upgrade. There is no way to upgrade a Ceph cluster while simultaneously reconfiguring OSDs to optimize storage in this way. To use different devices for journals or bucket indexes requires reprovisioning OSDs. For more information see [Using NVMe with LVM optimally](#) in [Ceph Object Gateway for Production](#).



## IMPORTANT

The **rolling\_update.yml** playbook includes the **serial** variable that adjusts the number of nodes to be updated simultaneously. Red Hat strongly recommends to use the default value (**1**), which ensures that Ansible will upgrade cluster nodes one by one.



## IMPORTANT

When using the **rolling\_update.yml** playbook to upgrade to any Red Hat Ceph Storage 3.x version, users who use the Ceph File System (CephFS) must manually update the Metadata Server (MDS) cluster. This is due to a known issue.

Comment out the MDS hosts in **/etc/ansible/hosts** before upgrading the entire cluster using **ceph-ansible rolling-upgrade.yml**, and then upgrade MDS manually. In the **/etc/ansible/hosts** file:

```
#[mdss]
#host-abc
```

For more details about this known issue, including how to update the MDS cluster, refer to the Red Hat Ceph Storage 3.0 [Release Notes](#).



## IMPORTANT

When upgrading a Red Hat Ceph Storage cluster from a previous version to 3.2, the Ceph Ansible configuration will default the object store type to BlueStore. If you still want to use FileStore as the OSD object store, then explicitly set the Ceph Ansible configuration to FileStore. This ensures newly deployed and replaced OSDs are using FileStore.



## IMPORTANT

When using the **rolling\_update.yml** playbook to upgrade to any Red Hat Ceph Storage 3.x version, and if you are using a multisite Ceph Object Gateway configuration, then you do not have to manually update the **all.yml** file to specify the multisite configuration.

## Prerequisites

- Log in as the **root** user on all nodes in the storage cluster.
- On all nodes in the storage cluster, enable the **rhel-7-server-extras-rpms** repository.

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

- If upgrading from Red Hat Ceph Storage 2.x to 3.x, on the Ansible administration node and the RBD mirroring node, enable the Red Hat Ceph Storage 3 Tools repository:

```
# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-rpms
```

- On the Ansible administration node, enable the Ansible repository:

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-ansible-2.6-rpms
```

- On the Ansible administration node, ensure the latest version of the **ansible** and **ceph-ansible** packages are installed.

```
[root@admin ~]# yum update ansible ceph-ansible
```

## 4.3. UPGRADING THE STORAGE CLUSTER

### Procedure

Use the following commands from the Ansible administration node.

1. As the **root** user, navigate to the **/usr/share/ceph-ansible/** directory:

```
[root@admin ~]# cd /usr/share/ceph-ansible/
```

2. Skip this step when upgrading from Red Hat Ceph Storage version 3.x to the latest version. Back up the **group\_vars/all.yml** and **group\_vars/osds.yml** files.

```
[root@admin ceph-ansible]# cp group_vars/all.yml group_vars/all_old.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml group_vars/osds_old.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml group_vars/clients_old.yml
```

3. Skip this step when upgrading from Red Hat Ceph Storage version 3.x to the latest version. When upgrading from Red Hat Ceph Storage 2.x to 3.x, create new copies of the **group\_vars/all.yml.sample**, **group\_vars/osds.yml.sample** and **group\_vars/clients.yml.sample** files, and rename them to **group\_vars/all.yml**, **group\_vars/osds.yml**, and **group\_vars/clients.yml** respectively. Open and edit them accordingly. For details, see [Appendix A, Changes in Ansible Variables Between Version 2 and 3](#) and [Section 1.2, "Installing a Red Hat Ceph Storage Cluster in Containers"](#).

```
[root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml.sample group_vars/clients.yml
```

4. Skip this step when upgrading from Red Hat Ceph Storage version 3.x to the latest version. When upgrading from Red Hat Ceph Storage 2.x to 3.x, open the **group\_vars/clients.yml** file, and uncomment the following lines:

```
keys:
- { name: client.test, caps: { mon: "allow r", osd: "allow class-read object_prefix
  rbd_children, allow rwx pool=test" }, mode: "{{ ceph_keyring_permissions }}" }
```

- a. Replace **client.test** with the real client name, and add the client key to the client definition line, for example:

■

```
key: "ADD-KEYRING-HERE=="
```

Now the whole line example would look similar to this:

```
- { name: client.test, key: "AQAIN8tUMICVFBAALRHNRV0Z4MXupRw4v9JQ6Q==", caps:
  { mon: "allow r", osd: "allow class-read object_prefix rbd_children, allow rwx pool=test" },
  mode: "{{ ceph_keyring_permissions }}" }
```



#### NOTE

To get the client key, run the **ceph auth get-or-create** command to view the key for the named client.

- When upgrading from 2.x to 3.x, in the **group\_vars/all.yml** file change the **ceph\_docker\_image** parameter to point to the Ceph 3 container version.

```
ceph_docker_image: rhceph/rhceph-3-rhel7
```

- Add the **fetch\_directory** parameter to the **group\_vars/all.yml** file.

```
fetch_directory: <full_directory_path>
```

*Replace:*

- <full\_directory\_path>** with a writable location, such as the Ansible user's home directory.

- If the cluster you want to upgrade contains any Ceph Object Gateway nodes, add the **radosgw\_interface** parameter to the **group\_vars/all.yml** file.

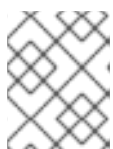
```
radosgw_interface: <interface>
```

*Replace:*

- <interface>** with the interface that the Ceph Object Gateway nodes listen to.

- Starting with Red Hat Ceph Storage 3.2, the default OSD object store is BlueStore. To keep the traditional OSD object store, you must explicitly set the **osd\_objectstore** option to **filestore** in the **group\_vars/all.yml** file.

```
osd_objectstore: filestore
```



#### NOTE

With the **osd\_objectstore** option set to **filestore**, replacing an OSD will use FileStore, instead of BlueStore.

- In the Ansible inventory file located at **/etc/ansible/hosts**, add the Ceph Manager (**ceph-mgr**) nodes under the **[mgrs]** section. Colocate the Ceph Manager daemon with Monitor nodes. Skip this step when upgrading from version 3.x to the latest version.

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

```
<monitor-host-name>
<monitor-host-name>
```

10. Copy **rolling\_update.yml** from the **infrastructure-playbooks** directory to the current directory.

```
[root@admin ceph-ansible]# cp infrastructure-playbooks/rolling_update.yml .
```

11. Create the **/var/log/ansible/** directory and assign the appropriate permissions for the **ansible** user:

```
[root@admin ceph-ansible]# mkdir /var/log/ansible
[root@admin ceph-ansible]# chown ansible:ansible /var/log/ansible
[root@admin ceph-ansible]# chmod 755 /var/log/ansible
```

- a. Edit the **/usr/share/ceph-ansible/ansible.cfg** file, updating the **log\_path** value as follows:

```
log_path = /var/log/ansible/ansible.log
```

12. As the Ansible user, run the playbook:

```
[user@admin ceph-ansible]$ ansible-playbook rolling_update.yml
```

To use the playbook only for a particular group of nodes on the Ansible inventory file, use the **--limit** option. For details, see [Section 1.7, "Understanding the limit option"](#).

13. While logged in as the **root** user on the RBD mirroring daemon node, upgrade **rbd-mirror** manually:

```
# yum upgrade rbd-mirror
```

Restart the daemon:

```
# systemctl restart ceph-rbd-mirror@<client-id>
```

14. Verify that the cluster health is OK.

- a. Log into a monitor node as the **root** user and list all running containers.

```
[root@monitor ~]# docker ps
```

- b. Verify the cluster health is OK.

```
[root@monitor ~]# docker exec ceph-mon-<mon-id> ceph -s
```

*Replace:*

- **<mon-id>** with the name of the Monitor container found in the first step.

For example:

```
[root@monitor ~]# docker exec ceph-mon-monitor ceph -s
```

15. If working in an OpenStack environment, update all the **cephx** users to use the RBD profile for pools. The following commands must be run as the **root** user:

- Glance users

```
ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd pool=<glance-pool-name>'
```

#### Example

```
[root@monitor ~]# ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd pool=images'
```

- Cinder users

```
ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd pool=<cinder-volume-pool-name>, profile rbd pool=<nova-pool-name>, profile rbd-read-only pool=<glance-pool-name>'
```

#### Example

```
[root@monitor ~]# ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```

- OpenStack general users

```
ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-only pool=<cinder-volume-pool-name>, profile rbd pool=<nova-pool-name>, profile rbd-read-only pool=<glance-pool-name>'
```

#### Example

```
[root@monitor ~]# ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-only pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```



#### IMPORTANT

Do these CAPS updates before performing any live client migrations. This allows clients to use the new libraries running in memory, causing the old CAPS settings to drop from cache and applying the new RBD profile settings.

## 4.4. UPGRADING RED HAT CEPH STORAGE DASHBOARD

The following procedure outlines the steps to upgrade Red Hat Ceph Storage Dashboard from version 3.1 to 3.2.

Before upgrading, ensure Red Hat Ceph Storage is upgraded from version 3.1 to 3.2. See [4.1. Upgrading the Storage Cluster](#) for instructions.

**WARNING**

The upgrade procedure will remove historical Storage Dashboard data.

**Procedure**

1. As the **root** user, update the **cephmetrics-ansible** package from the Ansible administration node:

```
[root@admin ~]# yum update cephmetrics-ansible
```

2. Change to the **/usr/share/cephmetrics-ansible** directory:

```
[root@admin ~]# cd /usr/share/cephmetrics-ansible
```

3. Install the updated Red Hat Ceph Storage Dashboard:

```
[root@admin cephmetrics-ansible]# ansible-playbook -v playbook.yml
```

## CHAPTER 5. MONITORING CEPH CLUSTERS RUNNING IN CONTAINERS WITH THE RED HAT CEPH STORAGE DASHBOARD

The Red Hat Ceph Storage Dashboard provides a monitoring dashboard to visualize the state of a Ceph Storage Cluster. Also, the Red Hat Ceph Storage Dashboard architecture provides a framework for additional modules to add functionality to the storage cluster.

- To learn about the Dashboard, see [Section 5.1, “The Red Hat Ceph Storage Dashboard”](#) .
- To install the Dashboard, see [Section 5.2, “Installing the Red Hat Ceph Storage Dashboard”](#) .
- To access the Dashboard, see [Section 5.3, “Accessing the Red Hat Ceph Storage Dashboard”](#) .
- To change the default password after installing the Dashboard, see [Section 5.4, “Changing the default Red Hat Ceph Storage dashboard password”](#).
- To learn about the Prometheus plugin, see [Section 5.5, “The Prometheus plugin for Red Hat Ceph Storage”](#).
- To learn about the Red Hat Ceph Storage Dashboard alerts and how to configure them, see [Section 5.6, “The Red Hat Ceph Storage Dashboard alerts”](#) .

### Prerequisites

- A Red Hat Ceph Storage cluster running in containers

## 5.1. THE RED HAT CEPH STORAGE DASHBOARD

The Red Hat Ceph Storage Dashboard provides a monitoring dashboard for Ceph clusters to visualize the storage cluster state. The dashboard is accessible from a web browser and provides a number of metrics and graphs about the state of the cluster, Monitors, OSDs, Pools, or the network.

With the previous releases of Red Hat Ceph Storage, monitoring data was sourced through a **collectd** plugin, which sent the data to an instance of the Graphite monitoring utility. Starting with Red Hat Ceph Storage 3.2, monitoring data is sourced directly from the **ceph-mgr** daemon, using the **ceph-mgr** Prometheus plugin.

The introduction of Prometheus as the monitoring data source simplifies deployment and operational management of the Red Hat Ceph Storage Dashboard solution, along with reducing the overall hardware requirements. By sourcing the Ceph monitoring data directly, the Red Hat Ceph Storage Dashboard solution is better able to support Ceph clusters deployed in containers.



### NOTE

With this change in architecture, there is no migration path for monitoring data from Red Hat Ceph Storage 2.x and 3.0 to Red Hat Ceph Storage 3.2.

The Red Hat Ceph Storage Dashboard uses the following utilities:

- The Ansible automation application for deployment.
- The embedded Prometheus **ceph-mgr** plugin.

- The Prometheus **node-exporter** daemon, running on each node of the storage cluster.
- The Grafana platform to provide a user interface and alerting.

The Red Hat Ceph Storage Dashboard supports the following features:

### General Features

- Support for Red Hat Ceph Storage 3.1 and higher
- SELinux support
- Support for FileStore and BlueStore OSD back ends
- Support for encrypted and non-encrypted OSDs
- Support for Monitor, OSD, the Ceph Object Gateway, and iSCSI roles
- Initial support for the Metadata Servers (MDS)
- Drill down and dashboard links
- 15 second granularity
- Support for Hard Disk Drives (HDD), Solid-state Drives (SSD), Non-volatile Memory Express (NVMe) interface, and Intel® Cache Acceleration Software (Intel® CAS)

### Node Metrics

- CPU and RAM usage
- Network load

### Configurable Alerts

- Out-of-Band (OOB) alerts and triggers
- Notification channel is automatically defined during the installation
- The Ceph Health Summary dashboard created by default  
See the [Red Hat Ceph Storage Dashboard Alerts](#) section for details.

### Cluster Summary

- OSD configuration summary
- OSD FileStore and BlueStore summary
- Cluster versions breakdown by role
- Disk size summary
- Host size by capacity and disk count
- Placement Groups (PGs) status breakdown
- Pool counts



- Device class summary, HDD vs. SSD

### Cluster Details

- Cluster flags status (**noout**, **nodown**, and others)
- OSD or Ceph Object Gateway hosts **up** and **down** status
- Per pool capacity usage
- Raw capacity utilization
- Indicators for active scrub and recovery processes
- Growth tracking and forecast (raw capacity)
- Information about OSDs that are **down** or **near full**, including the OSD host and disk
- Distribution of PGs per OSD
- OSDs by PG counts, highlighting the over or under utilized OSDs

### OSD Performance

- Information about I/O operations per second (IOPS) and throughput by pool
- OSD performance indicators
- Disk statistics per OSD
- Cluster wide disk throughput
- Read/write ratio (client IOPS)
- Disk utilization heat map
- Network load by Ceph role

### The Ceph Object Gateway Details

- Aggregated load view
- Per host latency and throughput
- Workload breakdown by HTTP operations

### The Ceph iSCSI Gateway Details

- Aggregated views
- Configuration
- Performance
- Per Gateway resource utilization
- Per client load and configuration

- Per Ceph Block Device image performance

## 5.2. INSTALLING THE RED HAT CEPH STORAGE DASHBOARD

The Red Hat Ceph Storage Dashboard provides a visual dashboard to monitor various metrics in a running Ceph Storage Cluster.



### NOTE

For information on upgrading the Red Hat Ceph Storage Dashboard see [Upgrading Red Hat Ceph Storage Dashboard](#) in the [Installation Guide for Red Hat Enterprise Linux](#).

### Prerequisites

- A Ceph Storage cluster running in containers deployed with the Ansible automation application.
- The storage cluster nodes use Red Hat Enterprise Linux 7.  
For details, see [Section 1.1.1, “Registering Red Hat Ceph Storage Nodes to the CDN and Attaching Subscriptions”](#).
- A separate node, the Red Hat Ceph Storage Dashboard node, for receiving data from the cluster nodes and providing the Red Hat Ceph Storage Dashboard.
- Prepare the Red Hat Ceph Storage Dashboard node:
  - Register the system with the Red Hat Content Delivery Network (CDN), attach subscriptions, and enable Red Hat Enterprise Linux repositories. For details, see [Section 1.1.1, “Registering Red Hat Ceph Storage Nodes to the CDN and Attaching Subscriptions”](#).
  - Enable the Tools repository on all nodes.  
For details, see the [Enabling the Red Hat Ceph Storage Repositories](#) section in the Red Hat Ceph Storage 3 *Installation Guide for Red Hat Enterprise Linux*.
  - If using a firewall, then ensure that the following TCP ports are open:

Table 5.1. TCP Port Requirements

Port	Use	Where?
<b>3000</b>	Grafana	The Red Hat Ceph Storage Dashboard node.
<b>9090</b>	Basic Prometheus graphs	The Red Hat Ceph Storage Dashboard node.
<b>9100</b>	Prometheus' <b>node-exporter</b> daemon	All storage cluster nodes.
<b>9283</b>	Gathering Ceph data	All <b>ceph-mgr</b> nodes.
<b>9287</b>	Ceph iSCSI gateway data	All Ceph iSCSI gateway nodes.

For more details see the [Using Firewalls](#) chapter in the *Security Guide* for Red Hat Enterprise Linux 7.

## Procedure

Run the following commands on the Ansible administration node as the **root** user.

1. Install the **cephmetrics-ansible** package.

```
[root@admin ~]# yum install cephmetrics-ansible
```

2. Using the Ceph Ansible inventory as a base, add the Red Hat Ceph Storage Dashboard node under the **[ceph-grafana]** section of the Ansible inventory file, by default located at **/etc/ansible/hosts**.

```
[ceph-grafana]
$HOST_NAME
```

Replace:

- **\$HOST\_NAME** with the name of the Red Hat Ceph Storage Dashboard node

For example:

```
[ceph-grafana]
node0
```

3. Change to the **/usr/share/cephmetrics-ansible/** directory.

```
[root@admin ~]# cd /usr/share/cephmetrics-ansible
```

4. Run the Ansible playbook.

```
[root@admin cephmetrics-ansible]# ansible-playbook -v playbook.yml
```



### IMPORTANT

Every time you update the cluster configuration, for example, you add or remove a MON or OSD node, you must re-run the **cephmetrics** Ansible playbook.



## NOTE

The **cephmetrics** Ansible playbook does the following actions:

- Updates the **ceph-mgr** instance to enable the prometheus plugin and opens TCP port 9283.
- Deploys the Prometheus **node-exporter** daemon to each node in the storage cluster.
  - Opens TCP port 9100.
  - Starts the **node-exporter** daemon.
- Deploys Grafana and Prometheus containers under Docker/systemd on the Red Hat Ceph Storage Dashboard node.
  - Prometheus is configured to gather data from the ceph-mgr nodes and the node-exporters running on each ceph host
  - Opens TCP port 3000.
  - The dashboards, themes and user accounts are all created in Grafana.
  - Outputs the URL of Grafana for the administrator.

## 5.3. ACCESSING THE RED HAT CEPH STORAGE DASHBOARD

Accessing the Red Hat Ceph Storage Dashboard gives you access to the web-based management tool for administrating Red Hat Ceph Storage clusters.

### Prerequisites

- [Install the Red Hat Ceph Storage Dashboard](#) .
- Ensure that NTP is synchronizing clocks properly because a time lag can occur among the Ceph Storage Dashboard node, cluster nodes, and a browser when the nodes are not properly synced. See the *Configuring the Network Time Protocol for Red Hat Ceph Storage* section in the Red Hat Ceph Storage 3 [Installation Guide for Red Hat Enterprise Linux](#) or [Ubuntu](#).

### Procedure

1. Enter the following URL to a web browser:

```
http://$HOST_NAME:3000
```

Replace:

- **\$HOST\_NAME** with the name of the Red Hat Ceph Storage Dashboard node

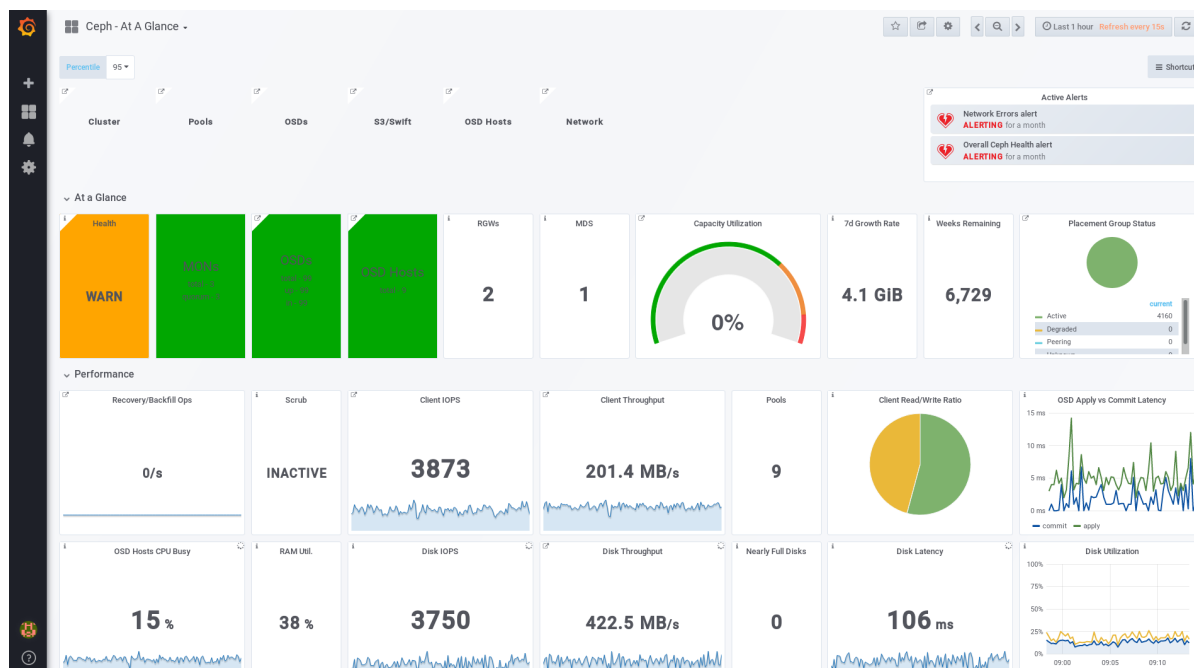
For example:

```
http://cephmetrics:3000
```

2. Enter the password for the **admin** user. If you did not set the password during the installation, use **admin**, which is the default password.

Once logged in, you are automatically placed on the *Ceph At a Glance* dashboard. The *Ceph At a Glance* dashboard provides a high-level overview of capacity, performance, and node-level performance information.

### Example



### Additional Resources

- See the [Changing the Default Red Hat Ceph Storage Dashboard Password](#) section in the Red Hat Ceph Storage Administration Guide.

## 5.4. CHANGING THE DEFAULT RED HAT CEPH STORAGE DASHBOARD PASSWORD

The default user name and password for accessing the Red Hat Ceph Storage Dashboard is set to **admin** and **admin**. For security reasons, you might want to change the password after the installation.



### NOTE

If you redeploy the Red Hat Ceph Storage dashboard using Ceph Ansible, then the password will be reset to the default value. Update the Ceph Ansible inventory file (**/etc/ansible/hosts**) with the custom password to prevent the password from resetting to the default value.

### Prerequisites

- [Install the Red Hat Ceph Storage Dashboard](#) .
- [Log in to the Red Hat Ceph Storage Dashboard](#) .

### Procedure

1. Click the Grafana icon in the upper-left corner.
2. Hover over the user name you want to modify the password for. In this case **admin**.

3. Click **Profile**.
4. Click **Change Password**.
5. Enter the new password twice and click **Change Password**.

#### Additional Resource

- If you forgot the password, follow the [Reset admin password](#) procedure on the Grafana web pages.

## 5.5. THE PROMETHEUS PLUGIN FOR RED HAT CEPH STORAGE

As a storage administrator, you can gather performance data, export that data using the Prometheus plugin module for the Red Hat Ceph Storage Dashboard, and then perform queries on this data. The Prometheus module allows **ceph-mgr** to expose Ceph related state and performance data to a Prometheus server.

### 5.5.1. Prerequisites

- Running Red Hat Ceph Storage 3.1 or higher.
- Installation of the Red Hat Ceph Storage Dashboard.

### 5.5.2. The Prometheus plugin

The Prometheus plugin provides an exporter to pass on Ceph performance counters from the collection point in **ceph-mgr**. The Red Hat Ceph Storage Dashboard receives **MMgrReport** messages from all **MgrClient** processes, such as Ceph Monitors and OSDs. A circular buffer of the last number of samples contains the performance counter schema data and the actual counter data. This plugin creates an HTTP endpoint and retrieves the latest sample of every counter when polled. The HTTP path and query parameters are ignored; all extant counters for all reporting entities are returned in a text exposition format.

#### Additional Resources

- See the [Prometheus documentation](#) for more details on the text exposition format.

### 5.5.3. Managing the Prometheus environment

To monitor a Ceph storage cluster with Prometheus you can configure and enable the Prometheus exporter so the metadata information about the Ceph storage cluster can be collected.

#### Prerequisites

- A running Red Hat Ceph Storage 3.1 cluster
- Installation of the Red Hat Ceph Storage Dashboard

#### Procedure

1. As the **root** user, open and edit the `/etc/prometheus/prometheus.yml` file.

- a. Under the **global** section, set the **scrape\_interval** and **evaluation\_interval** options to 15 seconds.

**Example**

```
global:
  scrape_interval: 15s
  evaluation_interval: 15s
```

- b. Under the **scrape\_configs** section, add the **honor\_labels: true** option, and edit the **targets**, and **instance** options for each of the **ceph-mgr** nodes.

**Example**

```
scrape_configs:
  - job_name: 'node'
    honor_labels: true
    static_configs:
      - targets: ['node1.example.com:9100']
        labels:
          instance: "node1.example.com"
      - targets: ['node2.example.com:9100']
        labels:
          instance: "node2.example.com"
```



**NOTE**

Using the **honor\_labels** option enables Ceph to output properly-labelled data relating to any node in the Ceph storage cluster. This allows Ceph to export the proper **instance** label without Prometheus overwriting it.

- c. To add a new node, simply add the **targets**, and **instance** options in the following format:

**Example**

```
- targets: ['new-node.example.com:9100']
  labels:
    instance: "new-node"
```



**NOTE**

The **instance** label has to match what appears in Ceph’s OSD metadata **instance** field, which is the short host name of the node. This helps to correlate Ceph stats with the node’s stats.

- 2. Add Ceph targets to the **/etc/prometheus/ceph\_targets.yml** file in the following format.

**Example**

```
[
  {
    "targets": [ "cephnode1.example.com:9283" ],
```

```

    "labels": {}
  }
]

```

3. Enable the Prometheus module:

```
# ceph mgr module enable prometheus
```

#### 5.5.4. Working with the Prometheus data and queries

The statistic names are exactly as Ceph names them, with illegal characters translated to underscores, and **ceph\_** prefixed to all names. All Ceph daemon statistics have a **ceph\_daemon** label that identifies the type and ID of the daemon they come from, for example: **osd.123**. Some statistics can come from different types of daemons, so when querying you will want to filter on Ceph daemons starting with **osd** to avoid mixing in the Ceph Monitor and RocksDB stats. The global Ceph storage cluster statistics have labels appropriate to what they report on. For example, metrics relating to pools have a **pool\_id** label. The long running averages that represent the histograms from core Ceph are represented by a pair of sum and count performance metrics.

The following example queries can be used in the Prometheus expression browser:

##### Show the physical disk utilization of an OSD

```
(irate(node_disk_io_time_ms[1m]) /10) and on(device,instance)
ceph_disk_occupation{ceph_daemon="osd.1"}
```

##### Show the physical IOPS of an OSD as seen from the operating system

```
irate(node_disk_reads_completed[1m]) + irate(node_disk_writes_completed[1m]) and on (device,
instance) ceph_disk_occupation{ceph_daemon="osd.1"}
```

##### Pool and OSD metadata series

Special data series are output to enable the displaying and the querying on certain metadata fields. Pools have a **ceph\_pool\_metadata** field, for example:

```
ceph_pool_metadata{pool_id="2",name="cephfs_metadata_a"} 1.0
```

OSDs have a **ceph\_osd\_metadata** field, for example:

```
ceph_osd_metadata{cluster_addr="172.21.9.34:6802/19096",device_class="ssd",ceph_daemon="osd.0",public_addr="172.21.9.34:6801/19096",weight="1.0"} 1.0
```

##### Correlating drive statistics with node\_exporter

The Prometheus output from Ceph is designed to be used in conjunction with the generic node monitoring from the Prometheus node exporter. Correlation of Ceph OSD statistics with the generic node monitoring drive statistics, special data series are output, for example:

```
ceph_disk_occupation{ceph_daemon="osd.0",device="sdd", exported_instance="node1"}
```

To get disk statistics by an OSD ID, use either the **and** operator or the asterisk (\*) operator in the Prometheus query. All metadata metrics have the value of **1** so they act neutral with asterisk operator.



Using asterisk operator allows to use **group\_left** and **group\_right** grouping modifiers, so that the resulting metric has additional labels from one side of the query. For example:

```
rate(node_disk_bytes_written[30s]) and on (device,instance)
ceph_disk_occupation{ceph_daemon="osd.0"}
```

### Using label\_replace

The **label\_replace** function can add a label to, or alter a label of, a metric within a query. To correlate an OSD and its disks write rate, the following query can be used:

```
label_replace(rate(node_disk_bytes_written[30s]), "exported_instance", "$1", "instance", "(.*):.*") and
on (device,exported_instance) ceph_disk_occupation{ceph_daemon="osd.0"}
```

### Additional Resources

- See Prometheus [querying basics](#) for more information on constructing queries.
- See Prometheus' [label\\_replace documentation](#) for more information.

## 5.5.5. Using the Prometheus expression browser

Use the builtin Prometheus expression browser to run queries against the collected data.

### Prerequisites

- A running Red Hat Ceph Storage 3.1 cluster
- Installation of the Red Hat Ceph Storage Dashboard

### Procedure

1. Enter the URL for the Prometheus web browser:

```
http://$DASHBOARD_SEVER_NAME:9090/graph
```

Replace...

- **\$DASHBOARD\_SEVER\_NAME** with the name of the Red Hat Ceph Storage Dashboard server.
2. Click on *Graph*, then type in or paste the query into the query window and press the *Execute* button.
    - a. View the results in the console window.
  3. Click on *Graph* to view the rendered data.

### Additional Resources

- See the [Prometheus expression browser](#) documentation on the Prometheus web site for more information.

## 5.5.6. Additional Resources

- The [Prometheus web page](#)

## 5.6. THE RED HAT CEPH STORAGE DASHBOARD ALERTS

This section includes information about alerting in the Red Hat Ceph Storage Dashboard.

- To learn about the Red Hat Ceph Storage Dashboard alerts, see [Section 5.6.2, "About Alerts"](#).
- To view the alerts, see [Section 5.6.3, "Accessing the Alert Status dashboard"](#).
- To configure the notification target, see [Section 5.6.4, "Configuring the Notification Target"](#).
- To change the default alerts or add new ones, see [Section 5.6.5, "Changing the Default Alerts and Adding New Ones"](#).

### 5.6.1. Prerequisites

- [Install the Red Hat Ceph Storage Dashboard](#) .
- [Log in to the Red Hat Ceph Storage Dashboard](#) .

### 5.6.2. About Alerts

The Red Hat Ceph Storage Dashboard supports alerting mechanism that is provided by the Grafana platform. You can configure the dashboard to send you a notification when a metric that you are interested in reaches certain value. Such metrics are in the **Alert Status** dashboard.

By default, **Alert Status** already includes certain metrics, such as *Overall Ceph Health*, *OSDs Down*, or *Pool Capacity*. You can add metrics that you are interested in to this dashboard or change their trigger values.

Here is a list of the pre-defined alerts that are included with Red Hat Ceph Storage Dashboard:

- Overall Ceph Health
- Disks Near Full (>85%)
- OSD Down
- OSD Host Down
- PG's Stuck Inactive
- OSD Host Less - Free Capacity Check
- OSD's With High Response Times
- Network Errors
- Pool Capacity High
- Monitors Down
- Overall Cluster Capacity Low
- OSDs With High PG Count

### 5.6.3. Accessing the Alert Status dashboard

Certain Red Hat Ceph Storage Dashboard alerts are configured by default in the **Alert Status** dashboard. This section shows two ways to access it.

#### Procedure

To access the dashboard:

- In the main **At the Glance** dashboard, click the **Active Alerts** panel in the upper-right corner.

Or..

- Click the dashboard menu from in the upper-left corner next to the Grafana icon. Select **Alert Status**.

### 5.6.4. Configuring the Notification Target

A notification channel called **cephmetrics** is automatically created during installation. All preconfigured alerts reference the **cephmetrics** channel but before you can receive the alerts, complete the notification channel definition by selecting the desired notification type. The Grafana platform supports a number of different notification types including email, Slack, and PagerDuty.

#### Procedure

- To configure the notification channel, follow the instructions in the [Alert Notifications](#) section on the Grafana web page.

### 5.6.5. Changing the Default Alerts and Adding New Ones

This section explains how to change the trigger value on already configured alerts and how to add new alerts to the **Alert Status** dashboard.

#### Procedure

- To change the trigger value on alerts or to add new alerts, follow the [Alerting Engine & Rules Guide](#) on the Grafana web pages.



#### IMPORTANT

To prevent overriding custom alerts, the **Alert Status** dashboard will not be updated when upgrading the Red Hat Ceph Storage Dashboard packages when you change the trigger values or add new alerts.

### Additional Resources

- The [Grafana web page](#)

## APPENDIX A. CHANGES IN ANSIBLE VARIABLES BETWEEN VERSION 2 AND 3

With Red Hat Ceph Storage 3, certain variables in the configuration files located in the `/usr/share/ceph-ansible/group_vars/` directory have changed or have been removed. The following table lists all the changes. After upgrading to version 3, copy the `all.yml.sample` and `osds.yml.sample` files again to reflect these changes. See [Upgrading a Red Hat Ceph Storage Cluster That Runs in Containers](#) for details.

Old Option	New Option	File
<code>mon_containerized_deployment</code>	<code>containerized_deployment</code>	<code>all.yml</code>
<code>ceph_mon_docker_interface</code>	<code>monitor_interface</code>	<code>all.yml</code>
<code>ceph_rhcs_cdn_install</code>	<code>ceph_repository_type: cdn</code>	<code>all.yml</code>
<code>ceph_rhcs_iso_install</code>	<code>ceph_repository_type: iso</code>	<code>all.yml</code>
<code>ceph_rhcs</code>	<code>ceph_origin: repository</code> and <code>ceph_repository: rhcs</code> (enabled by default)	<code>all.yml</code>
<code>journal_collocation</code>	<code>osd_scenario: collocated</code>	<code>osds.yml</code>
<code>raw_multi_journal</code>	<code>osd_scenario: non-collocated</code>	<code>osds.yml</code>
<code>raw_journal_devices</code>	<code>dedicated_devices</code>	<code>osds.yml</code>
<code>dmccrypt_journal_collocation</code>	<code>dmccrypt: true</code> + <code>osd_scenario: collocated</code>	<code>osds.yml</code>
<code>dmccrypt_dedicated_journal</code>	<code>dmccrypt: true</code> + <code>osd_scenario: non-collocated</code>	<code>osds.yml</code>