Red Hat Ceph Storage 3

Container Guide

Deploying and Managing Red Hat Ceph Storage in Containers

Last Updated: 2018-09-17

Red Hat Ceph Storage 3 Container Guide

Deploying and Managing Red Hat Ceph Storage in Containers

Legal Notice

Abstract

This document describes how to deploy and manage Red Hat Ceph Storage in containers.

Table of Contents

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 1.3, “Installing the Ceph Object Gateway in a Container”.
- To install Metadata Servers, see Section 1.4, “Installing Metadata Servers”.
- To learn about the Ansible **--limit** option, see Section 1.5, “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.

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:

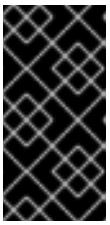
```
adduser $USER_NAME
```

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.

```
# 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
```

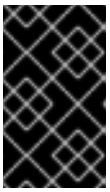
Replace

- **$USER_NAME** with the new user name for the Ansible user.
- **$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 use port **7480** by default. However, you can change the default port, for example to port **80**.

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

Prerequisite

- Network hardware is connected.

Procedure

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
```

2. 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="$IP_ADDR/$NETMASK_PREFIX" port protocol="tcp" \
port="6789" accept"
```

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

Replace

- **$IP_ADDR** with the network address of the Monitor node.
- **$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
```

3. 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.

4. 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.

5. 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 port **7480**:

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=7480/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=7480/tcp
--permanent
```

   To limit access based on the source address:

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="$IP_ADDR/$NETMASK_PREFIX" port protocol="tcp" \
port="7480" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="$IP_ADDR/$NETMASK_PREFIX" port protocol="tcp" \
port="7480" accept" --permanent
```

   Replace

   - **$IP_ADDR** with the network address of the object gateway node.
   - **$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="7480" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-
rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="7480" accept" --permanent
```

   b. Optional. If you changed the default Ceph Object Gateway port, 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="$IP_ADDR/$NETMASK_PREFIX" port protocol="tcp" \
port="80" accept"
```

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

Replace

- **$IP_ADDR** with the network address of the object gateway node.
- **$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="$IP_ADDR/$NETMASK_PREFIX" port protocol="tcp" \
port="443" accept"
```

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

Replace

- **$IP_ADDR** 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
```

Additional Resources

- 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.

1.2. 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

- 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-2.4-rpms
```

- On the Ansible administration node, install the **ceph-ansible** package:

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

Procedure

Use the following commands from the Ansible administration node if not instructed otherwise.

1. In the user's home directory, create the **ceph-ansible-keys** directory where Ansible stores temporary values generated by the **ceph-ansible** playbook.

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

2. 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
```

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

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

4. 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
```

5. Edit the copied files.

   a. 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.

Option	Value	Required	Notes
monitor_interface	The interface that the Monitor nodes listen to	**monitor_interface**, **monitor_address**, or **monitor_address_block** is required	
monitor_address	The address that the Monitor nodes listen to		
monitor_address_block	The subnet of the Ceph public network		Use when the IP addresses of the nodes are unknown, but the subnet is known
ip_version	**ipv6**	Yes if using IPv6 addressing	
journal_size	The required size of the journal in MB	No	

Option	Value	Required	Notes
public_network	The IP address and netmask of the Ceph public network	Yes	The *Verifying the Network Configuration for Red Hat Ceph Storage* section in the *Installation Guide for Red Hat Enterprise Linux*
cluster_network	The IP address and netmask of the Ceph cluster network	No	
ceph_docker_image	**rhceph/rhceph-3-rhel7**, or **cephimageinlocalreg** if using a local Docker registry	Yes	
containerized_deployment	**true**	Yes	
ceph_docker_registry	**registry.access.redhat.com**, or **<local-host-fqdn>** 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.

Table 1.1. OSD Ansible Settings

Option	Value	Required	Notes

Option	Value	Required	Notes
osd_scenario	**collocated** to use the same device for journal and OSD data **non-collocated** to use a dedicated device to store journal data **lvm** to use the Logical Volume Manager to store OSD data	Yes	When using **osd_scenario: non-collocated**, **ceph-ansible** expects the variables **devices** and **dedicated_devices** to match. For example, if you specify 10 disks in **devices**, you must specify 10 entries in **dedicated_devices**. Currently, Red Hat only supports dedicated journals when using **osd_scenario: lvm**, not collocated journals.
osd_auto_discovery	**true** to automatically discover OSDs	Yes if using **osd_scenario: collocated**	Cannot be used when **devices** setting is used
devices	List of devices where **ceph data** is stored	Yes to specify the list of devices	Cannot be used when **osd_auto_discovery** setting is used
dedicated_devices	List of dedicated devices for non-collocated OSDs where **ceph journal** is stored	Yes if **osd_scenario: non-collocated**	Should be nonpartitioned devices
dmcrypt	**true** to encrypt OSDs	No	Defaults to **false**

Option	Value	Required	Notes
lvm_volumes	a list of dictionaries	Yes if using **osd_scenario: lvm**	Each dictionary must contain a **data**, **journal** and **data_vg** keys. The **data** key must be a logical volume. The **journal** key can be a logical volume (LV), device or partition, but do not use one journal for multiple **data** LVs. The **data_vg** key must be the volume group containing the **data** LV. Optionally, the **journal_vg** key can be used to specify the volume group containing the journal LV, if applicable.

The following are examples of the **osds.yml** file using these three **osd_scenario:**: **collocated**, **non-collocated**, and **lvm**.

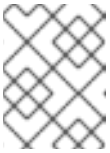
```
osd_scenario: non-collocated
devices:
  - /dev/sda
  - /dev/sdb
  - /dev/sdc
  - /dev/sdd
dedicated_devices:
   - /dev/nvme0n1
   - /dev/nvme0n1
   - /dev/nvme0n1
   - /dev/nvme0n1
```

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

```
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
```

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



NOTE

Currently, **ceph-ansible** does not create the volume groups or the logical volumes. This must be done before running the Anisble playbook.

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]>
   ```

   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.

   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 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
```

9. As the Ansible user, 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 --skip-
tags=with_pkg site-docker.yml
```

10. From 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
```

NOTE

In addition to verifying the cluster status, you can use the **ceph-medic** utility to overall diagnose the Ceph Storage Cluster. See the *Installing and Using **ceph-medic** to Diagnose a Ceph Storage Cluster* chapter in the Red Hat Ceph Storage 3 *Troubleshooting Guide*.

1.3. 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. See Section 1.2, “Installing a Red Hat Ceph Storage Cluster in Containers” for details.

Procedure

Use the following commands from the Ansible administration node.

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

```
[user@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

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

3. 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
```

4. Optional. Edit the **group_vars/rgws.yml** file. For additional details, see the **rgws.yml** file.

5. 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 Chapter 2, *Colocation of Containerized Ceph Daemons* for details.

6. 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 --skip-
tags=with_pkg site-docker.yml
```

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

   a. Connect to a Monitor node:

```
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.

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*
- Section 1.5, “Understanding the **limit** option”

1.4. 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.

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 Chapter 2, *Colocation of Containerized Ceph Daemons* for details.

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

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

3. 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
```

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

5. Run the Ansible playbook:

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

6. 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
- Section 1.5, “Understanding the **limit** option”

1.5. 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
```

For example, to redeploy only OSDs:

```
$ ansible-playbook /usr/share/ceph-ansible/site.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.

1.6. 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 colocating 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

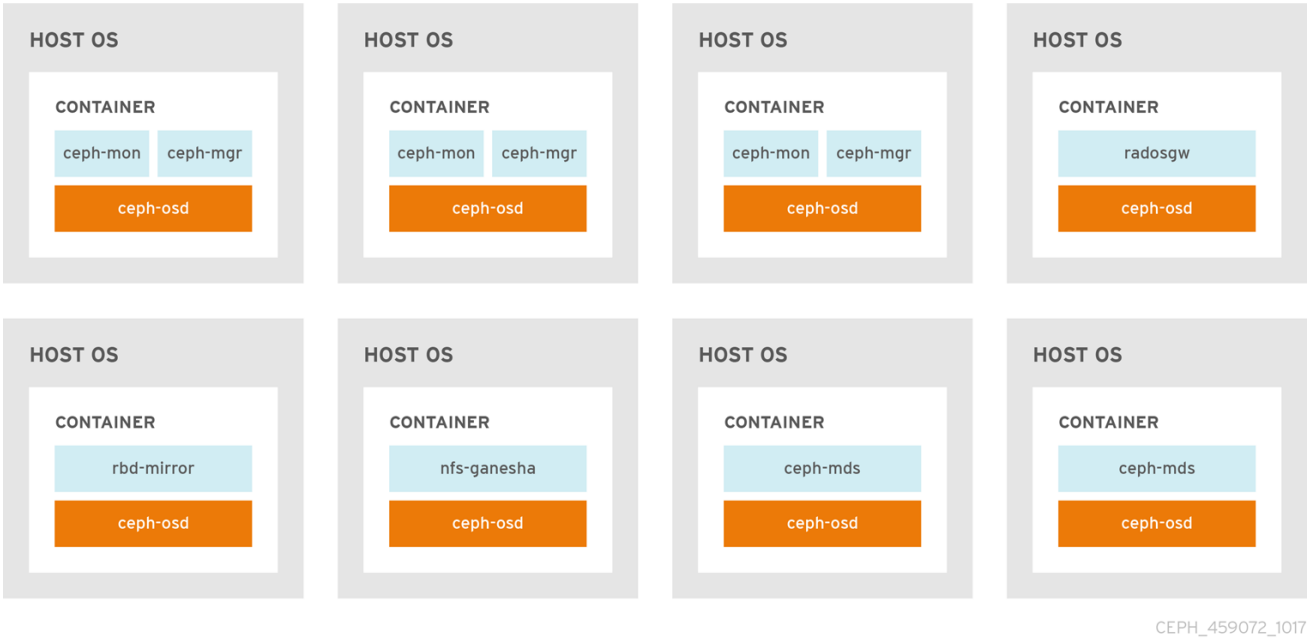
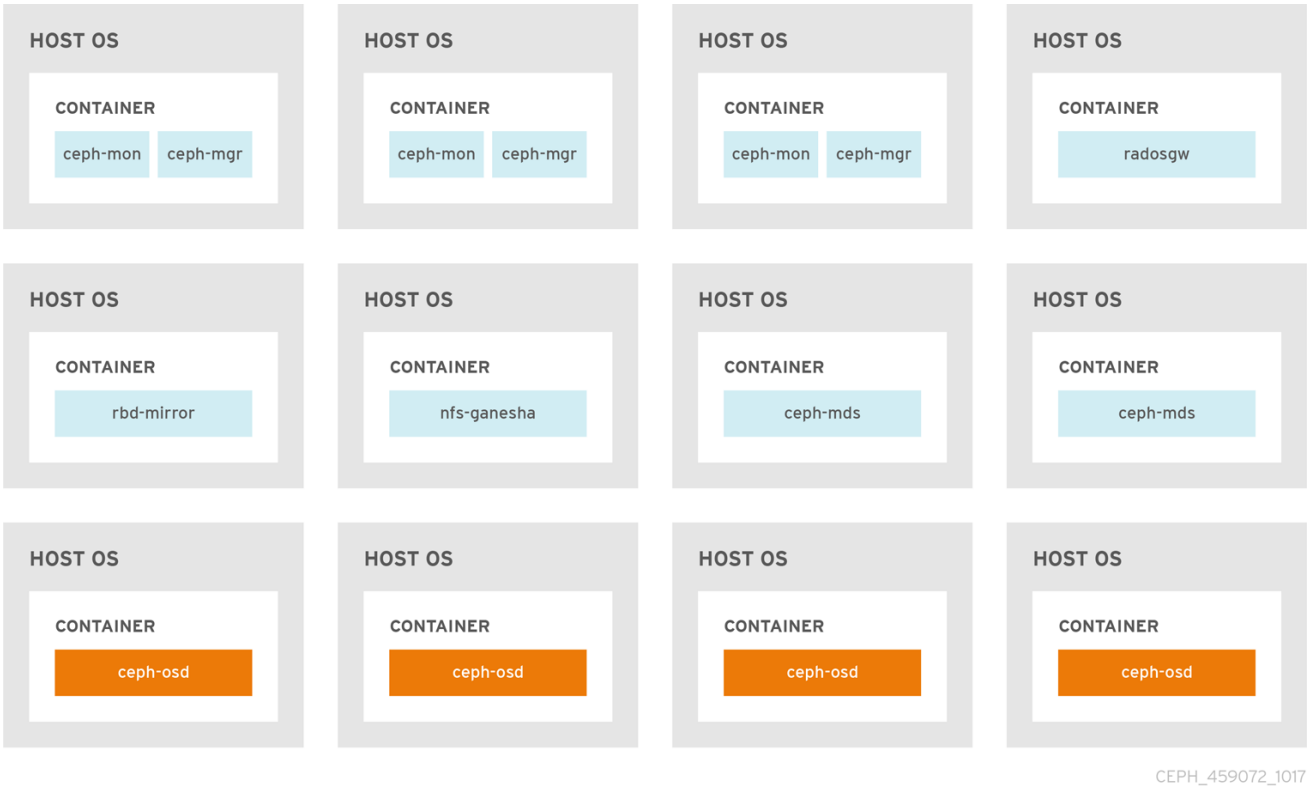


Figure 2.2. Non-colocated Daemons



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 daemon on a same node, the **ceph-ansible** playbook reserves CPU and RAM resources to each. By default, **ceph-ansible** uses values listed in the {hardware_guide}#ceph-hardware-min-recommend[Recommended Minimum Hardware] chapter in the Red Hat Ceph Storage Hardware Selection Guide. This section describes how to change the default values.

Procedure

- To change the default RAM and CPU limit for a daemon, set the **ceph_<daemon-type>_docker_memory_limit** and **ceph_<daemon-type>_docker_cpu_limit** parameters in the appropriate **.yml** configuration file when deploying the daemon.
  For example, to change the default RAM limit to 2 GB and the CPU limit to 2 for the Ceph Object Gateway, edit the **/usr/share/ansible/group_vars/rgws.yml** file as follows:

```
ceph_rgw_docker_memory_limit: 2g
ceph_rgw_docker_cpu_limit: 2
```

Additional Resources

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

2.3. ADDITIONAL RESOURCES

- Chapter 1, *Deploying Red Hat Ceph Storage in Containers*
- The Red Hat Ceph Storage Hardware Selection Guide 3

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 3.4, “Upgrading a Red Hat Ceph Storage Cluster That Runs in Containers”

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

This section describes how to start, stop, or restart Ceph daemons that run in containers

Procedure

- To start, stop, or restart a Ceph daemon running in a container:

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

  Where:

  - **<action>** is the action to perform; **start**, **stop**, or **restart**
  - **<daemon>** is the daemon; **osd**, **mon**, **mds**, or **rgw**
  - **<ID>** is either
    - The device name that the **ceph-osd** daemon uses
    - The short host name where the **ceph-mon**, **ceph-mds**, or **ceph-rgw** daemons are running

  For example, to restart a **ceph-osd** daemon that uses the **/dev/sdb** device:

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

  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-rgw@ceph-rgw01
  ```

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: Viewing Log Files of Ceph Daemons That Run in Containers

- To view the entire Ceph log file.

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

  Where:

  - **<daemon>** is the Ceph daemon; **osd**, **mon**, or **rgw**
  - **<ID>** is either
    - The device name that the **ceph-osd** daemon uses
    - The short host name where the **ceph-mon** or **ceph-rgw** daemons are running

  For example, to view the entire log for the **ceph-osd** daemon that uses the **/dev/sdb** device:

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

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

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

2. Copy the **purge-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. Use the **purge-docker-cluster.yml** playbook to purge the Ceph cluster.

   - 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
```

   - 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
```

   - 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
```

   - 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
```

3.4. 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.

IMPORTANT

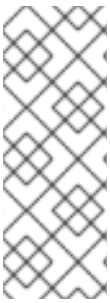
Please contact Red Hat support prior to upgrading, if you have a large Ceph Object Gateway storage cluster with millions of objects present in buckets.

For more details refer to the Red Hat Ceph Storage 3.0 Release Notes, under the *Slow OSD startup after upgrading to Red Hat Ceph Storage 3.0* heading.

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

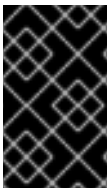


NOTE

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 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 Red Hat Ceph Storage 3.0 and from version 3.0 to other zStream releases of 3.0, users who use the Ceph File System (CephFS) must manually update the Metadata Server (MDS) cluster. This is due to a known issue.

Comment 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.

Prerequisites

- On all nodes in the 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 and Ansible repository:

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-
rhceph-3-tools-rpms --enable=rhel-7-server-ansible-2.4-rpms
```

- If upgrading from Red Hat Ceph Storage 3.0 to 3.1 and using Red Hat Ceph Storage Dashboard, before upgrading the cluster, purge the old cephmetrics installation from the cluster. This avoids an issue where the dashboard won't display data after upgrade.

  1. If the **`cephmetrics-ansible`** package isn't already updated, update it:

     ■

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

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

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

3. Purge the existing cephmetrics installation.

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

4. Install the updated Red Hat Ceph Storage Dashboard

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

- 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
```

Procedure

Use the following commands from the Ansible administration node.

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

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

2. Back up the **group_vars/all.yml** and **group_vars/osds.yml** files. Skip this step when upgrading from version 3.x to the latest version.

```
[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
```

3. Create new copies of the **group_vars/all.yml.sample** and **group_vars/osds.yml.sample** named **group_vars/all.yml** and **group_vars/osds.yml** respectively and edit them according to you deployment. Skip this step when upgrading from version 3.x to the latest version. 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
```

4. 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
```

5. 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.

6. 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.

7. 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>
```

8. Copy **rolling_update.yml** from the **infrastructure-playbooks** directory to the current directory.

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

9. 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
```

10. 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 1.5, "Understanding the **limit** option".

11. From the RBD mirroring daemon node, upgrade **rbd-mirror** manually:

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

Restart the daemon:

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

12. Verify that the cluster health is OK.

    a. From a monitory node, list all running containers.

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

    b. Verify that 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
    ```

CHAPTER 4. 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 4.1, “The Red Hat Ceph Storage Dashboard”.
- To install the Dashboard, see Section 4.2, “Installing the Red Hat Ceph Storage Dashboard”.
- To access the Dashboard, see Section 4.3, “Accessing the Red Hat Ceph Storage Dashboard”.
- To change the default password after installing the Dashboard, see Section 4.4, “Changing the default Red Hat Ceph Storage dashboard password”.
- To learn about the Prometheus plugin, see Section 4.5, “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 4.6, “The Red Hat Ceph Storage Dashboard alerts”.

Prerequisites

- A Red Hat Ceph Storage cluster running in containers

4.1. 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 3.1, 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 3.1.

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

4.2. 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.

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.

    ```
    [root@admin ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-rpms --enable=rhel-7-server-ansible-2.4-rpms
    ```

  - If using a firewall, then ensure that the following TCP ports are open:

    Table 4.1. TCP Port Requirements

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

Use 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 to 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
```

Replace:

- **<password>** with the new password

+ For example:

+

```
grafana:
  admin_password: CGqf5HhUaZ
```

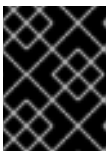
1. Use the Ansible playbook.

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

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.



IMPORTANT

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

4.3. 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.

Procedure

1. Enter the following URL to a web browser:

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

Replace:

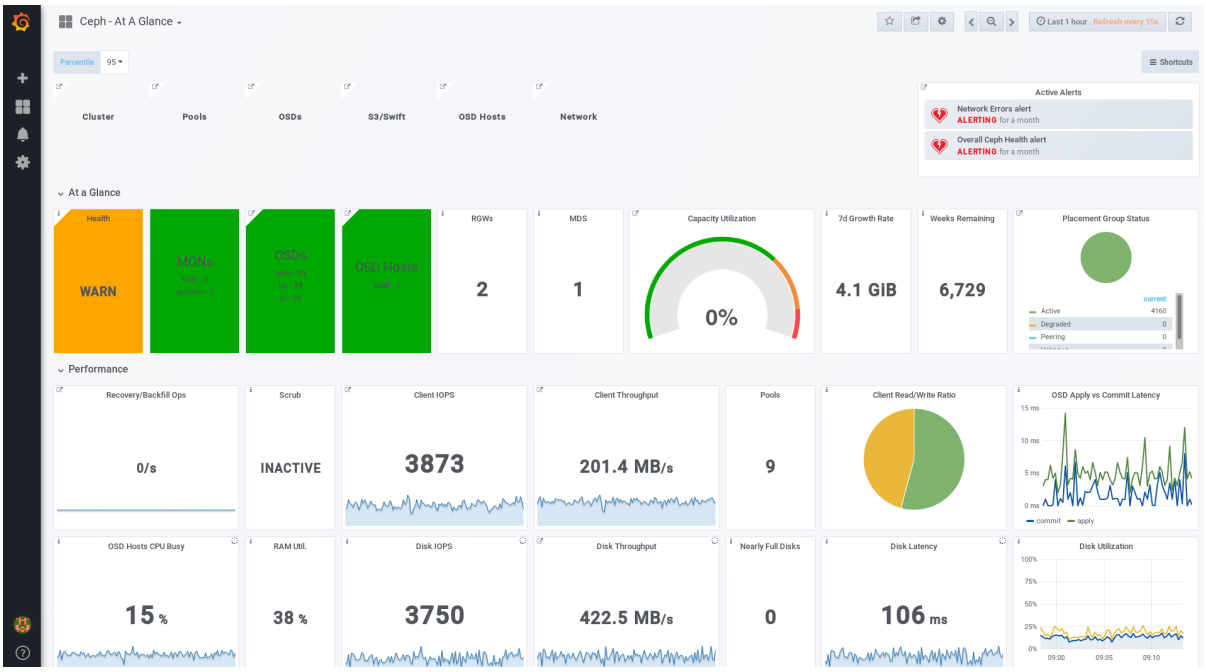
- **$HOST_NAME** with the name of the Red Hat Ceph Storage Dashboard node

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 overviews of capacity, performance, and node-level performance information.

Example

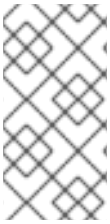


Additional Resources

- See the Changing the Default Red Hat Ceph Storage Dashboard Password section in the Red Hat Ceph Storage Administration Guide.

4.4. 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.

4.5. 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.

4.5.1. Prerequisites

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

4.5.2. 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.

4.5.3. 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. 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
```

4.5.4. 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.

4.5.5. 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 Prometh the 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.

4.5.6. Additional Resources

- The Prometheus web page

4.6. 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 4.6.2, “About Alerts”.
- To view the alerts, see Section 4.6.3, “Accessing the Alert Status dashboard”.
- To configure the notification target, see Section 4.6.4, “Configuring the Notification Target”.
- To change the default alerts or add new ones, see Section 4.6.5, “Changing the Default Alerts and Adding New Ones”.

4.6.1. Prerequisites

- Install the Red Hat Ceph Storage Dashboard .
- Log in to the Red Hat Ceph Storage Dashboard.

4.6.2. 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

4.6.3. 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**.

4.6.4. 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.

4.6.5. 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 Section 3.4, “Upgrading a Red Hat Ceph Storage Cluster That Runs in Containers” for details.

Old Option	New Option	File
mon_containerized_deployment	**containerized_deployment**	**all.yml**
ceph_mon_docker_interface	**monitor_interface**	**all.yml**
ceph_rhcs_cdn_install	**ceph_repository_type: cdn**	**all.yml**
ceph_rhcs_iso_install	**ceph_repository_type: iso**	**all.yml**
ceph_rhcs	**ceph_origin: repository** and **ceph_repository: rhcs** (enabled by default)	**all.yml**
journal_collocation	**osd_scenario: collocated**	**osds.yml**
raw_multi_journal	**osd_scenario: non-collocated**	**osds.yml**
raw_journal_devices	**dedicated_devices**	**osds.yml**
dmcrytpt_journal_collocation	**dmcrypt: true** + **osd_scenario: collocated**	**osds.yml**
dmcrypt_dedicated_journal	**dmcrypt: true** + **osd_scenario: non-collocated**	**osds.yml**