Red Hat Hyperconverged Infrastructure for Cloud 13

Operations Guide

Operating a Red Hat Hyperconverged Infrastructure for Cloud Solution

Last Updated: 2018-07-13

Red Hat Hyperconverged Infrastructure for Cloud 13 Operations Guide

Operating a Red Hat Hyperconverged Infrastructure for Cloud Solution

Legal Notice

Abstract

This document provides instructions for operating a Red Hat Hyperconverged Infrastructure for Cloud, using Red Hat OpenStack Platform 13 and Red Hat Ceph Storage 3, all running on AMD64 and Intel 64 architectures.

Table of Contents

CHAPTER 1. PERFORMING RED HAT HYPERCONVERGED INFRASTRUCTURE CLOUD OPERATIONAL TASKS

The Red Hat Hyperconverged Infrastructure (RHHI) Cloud solution has three basic operational tasks:

- Updating the overcloud configuration
- Adding nodes to the overcloud
- Removing nodes from the overcloud

CHAPTER 2. UPDATING THE OVERCLOUD CONFIGURATION

At times you will need to update the Red Hat Hyperconverged Infrastructure (RHHI) for Cloud configuration to add new features, or change the way the overcloud functions.

Prerequisite

- A running RHHI for Cloud solution.

Procedure

Do the following step on the Red Hat OpenStack Platform director node, as the **stack** user.

1. Rerun the **openstack overcloud deploy** command with the same TripleO Heat templates from the initial overcloud deployment:

   ### Example

   ```
   [stack@director ~]$ openstack overcloud deploy --templates \
   -r ~/custom-templates/custom-roles.yaml \
   -e /usr/share/openstack-tripleo-heat-templates/environments/puppet-
   pacemaker.yaml \
   -e /usr/share/openstack-tripleo-heat-templates/environments/network-
   isolation.yaml \
   -e /usr/share/openstack-tripleo-heat-templates/environments/storage-
   environment.yaml \
   -e ~/custom-templates/network.yaml \
   -e ~/custom-templates/ceph.yaml \
   -e ~/custom-templates/compute.yaml \
   -e ~/custom-templates/layout.yaml
   ```

   NOTE

   If adding a new environment file to the overcloud, then add an additional **-e** argument to the **openstack overcloud deploy** command.

Additional Resources

- The Red Hat Hyperconverged Infrastructure Cloud Deployment Guide.
- The Red Hat OpenStack Platform 10 Director Installation and Usage Guide.

CHAPTER 3. ADDING A NODE TO THE OVERCLOUD

The overcloud can grow to meet an increase in demand by adding a new Nova compute and Ceph OSD node to the overcloud.

Prerequisites

- A running RHHI Cloud solution.
- The MAC addresses for the network interface cards (NICs).
- IPMI User name and password

Procedure

Do the following steps on the Red Hat OpenStack Platform director node, as the **stack** user.

1. Create and populate a host definition file for the Ironic service to manage the new node.

   a. Create a new JSON host definition file:

   ```
   [stack@director ~]$ touch ~/new_node.json
   ```

   b. Add a definition block for the new node between the **nodes** stanza square brackets (**{"nodes": []}**) using this template:

   ```
   {
     "pm_password": "$IPMI_USER_PASSWORD",
     "name": "$NODE_NAME",
     "pm_user": "$IPMI_USER_NAME",
     "pm_addr": "$IPMI_IP_ADDR",
     "pm_type": "pxe_ipmitool",
     "mac": [
            "$NIC_MAC_ADDR"
           ],
     "arch": "x86_64",
     "capabilities": "node:$NODE_ROLE-
   INSTANCE_NUM,boot_option:local"
   }
   ```

   Replace…

   - **$IPMI_USER_PASSWORD** with the IPMI password.
   - **$NODE_NAME** with a descriptive name of the node. This is an optional parameter.
   - **$IPMI_USER_NAME** with the IPMI user name that has access to power the node on or off.
   - **$IPMI_IP_ADDR** with the IPMI IP address.
   - **$NIC_MAC_ADDR** with the network card MAC address handling the PXE boot.
   - **$NODE_ROLE-INSTANCE_NUM** with the node's role, along with a node number. This solution uses two roles: **controller** and **osd-compute**.

```
{
  "nodes": [
     {
         "pm_password": "AbC1234",
         "name": "m630_slot2",
         "pm_user": "ipmiadmin",
         "pm_addr": "10.19.143.62",
         "pm_type": "pxe_ipmitool",
         "mac": [
             "c8:1f:66:65:33:42"
         ],
         "arch": "x86_64",
          "capabilities": "node:osd-compute-
3,boot_option:local"
     }
  ]
}
```

2. Import the nodes into the Ironic database:

```
[stack@director ~]$ openstack baremetal import ~/new_node.json
```

   a. Verify that the **`openstack baremetal import`** command populated the Ironic database with the new node:

```
[stack@director ~]$ openstack baremetal node list
```

3. Set the new node into maintenance mode:

```
ironic node-set-maintenance $UUID true
```

Replace…

- **`$UUID`** with the UUID of the new node. See the output from step 2a to get the new node's UUID.

  Example

```
[stack@director ~]$ ironic node-set-maintenance 7250678a-a575-
4159-840a-e7214e697165 true
```

4. Inspect the new node's hardware:

```
openstack baremetal introspection start $UUID
```

Replace…

- **`$UUID`** with the UUID of the new node. See the output from step 2a to get the new node's UUID.

  Example

```
[stack@director ~]$ openstack baremetal introspection start
7250678a-a575-4159-840a-e7214e697165 true
```

a. The introspection process can take some time to complete. Verify that the status of the introspection process:

```
[stack@director ~]$ openstack baremetal introspection bulk
status
```

Example Output

```
+--------------------------------------+----------+------
-+
| Node UUID                            | Finished | Error
|
+--------------------------------------+----------+------
-+
| a94b75e3-369f-4b2d-b8cc-8ab272e23e89 | True     | None  |
| 7ace7b2b-b549-414f-b83e-5f90299b4af3 | True     | None  |
| 8be1d83c-19cb-4605-b91d-928df163b513 | True     | None  |
| e8411659-bc2b-4178-b66f-87098a1e6920 | True     | None  |
| 04679897-12e9-4637-9998-af8bee30b414 | True     | None  |
| 48b4987d-e778-48e1-ba74-88a08edf7719 | True     | None  |
| 7250678a-a575-4159-840a-e7214e697165 | True     | None  |
+--------------------------------------+----------+------
-+
```

5. Disable maintenance mode on the new node:

```
ironic node-set-maintenance $UUID false
```

Replace…

- **`$UUID`** with the UUID of the new node. See the output from step 2a to get the new node's UUID.

Example

```
[stack@director ~]$ ironic node-set-maintenance 7250678a-a575-
4159-840a-e7214e697165 false
```

6. Assign the full overcloud kernel and ramdisk image to the new node:

```
[stack@director ~]$ openstack baremetal configure boot
```

7. Open the **`~/custom-templates/layout.yaml`** file for editing.

a. Under the **`parameter_defaults`** section, change the **`OsdComputeCount`** option from **`3`** to **`4`**.

b. Under the **OsdComputeIPs** section, add the new node's IP addresses for each isolated network.

8. Apply the new overcloud configuration by rerunning the **openstack overcloud deploy** command with the same TripleO Heat templates from the initial overcloud deployment:

Example

```
[stack@director ~]$ openstack overcloud deploy --templates \
-r ~/custom-templates/custom-roles.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/puppet-
pacemaker.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-
isolation.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/storage-
environment.yaml \
-e ~/custom-templates/network.yaml \
-e ~/custom-templates/ceph.yaml \
-e ~/custom-templates/compute.yaml \
-e ~/custom-templates/layout.yaml
```

9. Verify the addition of the new node:

```
[stack@director ~]$ openstack server list
```

NOTE

If the node status is **ACTIVE**, then the new node was added successfully to the overcloud.

CHAPTER 4. REMOVING A NODE FROM THE OVERCLOUD

The Red Hat OpenStack Platform director (RHOSP-d) does not support the removal of a Red Hat Ceph Storage (RHCS) node automatically.

4.1. PREREQUISITES

- Verify there will be enough CPU and RAM to service the workloads.
- Migrate the compute workloads off of the node being removed.
- Verify that the storage cluster has enough reserve storage capacity to maintain a status of **HEALTH_OK**.

4.2. REMOVING THE CEPH OSD SERVICES FROM THE STORAGE CLUSTER

This procedure removes the Ceph OSD services from being a member of the storage cluster.

Prerequisite

- A healthy Ceph storage cluster.

Procedure

Do the following steps on one of the Controller/Monitor nodes, and as the **root** user, unless otherwise stated.

1. Verify the health status of the Ceph storage cluster:

```
[root@controller ~]# ceph health
```

The health status must be **HEALTH_OK** before continuing on with this procedure.

WARNING

If the **ceph health** command reports that the storage cluster is **near full**, then removing a Ceph OSD could result in exceeding or reaching the full ratio limit. This could cause data loss. If the storage cluster is **near full**, then contact Red Hat Support before proceeding.

2. Determine the number of Ceph OSDs for removal:

```
[root@controller ~]# ceph osd tree
```

Example Output

```
ID WEIGHT   TYPE NAME                          UP/DOWN REWEIGHT
PRIMARY-AFFINITY
```

```
-1 52.37256 root default
-2 13.09314     host overcloud-osd-compute-3
 0  1.09109         osd.0                          up  1.00000
1.00000
 4  1.09109         osd.4                          up  1.00000
1.00000
 8  1.09109         osd.8                          up  1.00000
1.00000
12  1.09109         osd.12                         up  1.00000
1.00000
16  1.09109         osd.16                         up  1.00000
1.00000
20  1.09109         osd.20                         up  1.00000
1.00000
24  1.09109         osd.24                         up  1.00000
1.00000
28  1.09109         osd.28                         up  1.00000
1.00000
32  1.09109         osd.32                         up  1.00000
1.00000
36  1.09109         osd.36                         up  1.00000
1.00000
40  1.09109         osd.40                         up  1.00000
1.00000
44  1.09109         osd.44                         up  1.00000
1.00000
...
```

To view the total number of OSDs *up* and *in*:

```
[root@controller ~]# ceph osd stat
```

Example Output

```
osdmap e173: 48 osds: 48 up, 48 in
            flags sortbitwise
```

3. Monitor the Ceph storage cluster from a new terminal session:

```
[root@controller ~]# ceph -w
```

In this terminal session, you can watch as the OSD is removed from the storage cluster. Go back to the original terminal session for the next step.

4. Mark the OSD *out*:

```
ceph osd out $OSD_NUM
```

Replace…

- **`$OSD_NUM`** with the number portion of the OSD name.

  Example

```
[root@controller ~]# ceph osd out 0
marked out osd.0.
```

Set all OSDs on the node to *out*.

NOTE

If scripting this step to handle multiple OSDs sequentially, then set a **`sleep`** command of at least 10 seconds in between the running of each **`ceph osd out`** command.

5. Wait for all the placement groups to become **`active+clean`** and the storage cluster is in a **`HEALTH_OK`** state. You can watch the placement group migration from the new terminal session from step 3. This rebalancing of data can take some time to complete.

6. Verify the health status of the Ceph storage cluster:

```
[root@controller ~]# ceph health
```

7. From the Compute/OSD node, and as the **`root`** user, disable and stop all OSD daemons:

```
[root@osdcompute ~]# systemctl disable ceph-osd.target
[root@osdcompute ~]# systemctl stop ceph-osd.target
```

8. Remove the OSD from the CRUSH map:

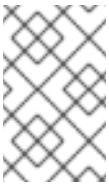
```
ceph osd crush remove osd.$OSD_NUM
```

Replace…

- **`$OSD_NUM`** with the number portion of the OSD name.

Example

```
[root@controller ~]# ceph osd crush remove osd.0
removed item id 0 name 'osd.0' from crush map
```



NOTE

Removing an OSD from the CRUSH map, causes CRUSH to recompute which OSDs get the placement groups, and rebalances the data accordingly.

9. Remove the OSD authentication key:

```
ceph auth del osd.$OSD_NUM
```

Replace…

- **`$OSD_NUM`** with the number portion of the OSD name.

Example

```
[root@controller ~]# ceph osd auth del osd.0
updated
```

10. Remove the OSD:

```
ceph osd rm $OSD_NUM
```

Replace…

- **$OSD_NUM** with the number portion of the OSD name.

Example

```
[root@controller ~]# ceph osd rm 0
removed osd.0
```

4.3. REMOVING THE NOVA COMPUTE SERVICES FROM THE OVERCLOUD

This procedure removes the Nova compute services from being a member of the overcloud, and powers off the hardware.

Prerequisite

- Migrate any running instances to another compute node in the overcloud.

Procedure

Do the following steps on the Red Hat OpenStack Platform director (RHOSP-d) node, as the **stack** user.

1. Verify the status of the compute node:

```
[stack@director ~]$ nova service-list
```

2. Disable the compute service:

```
nova service-disable $HOST_NAME nova-compute
```

Replace…

- **$HOST_NAME** with the compute's host name.

Example

```
[stack@director ~]$ nova service-disable overcloud-osd-compute-
3.localdomain nova-compute
+--------------------------------------+--------------+------
----+
| Host                                 | Binary       | Status
```

```
|
+--------------------------------------+-------------+------
----+
| overcloud-osd-compute-3.localdomain | nova-compute | disabled
|
+--------------------------------------+-------------+------
----+
```

3. Collect the Nova ID of the compute node:

```
[stack@director ~]$ openstack server list
```

Write down the Nova UUID, which is in the first column of the command output.

4. Collect the OpenStack Platform name:

```
[stack@director ~]$ heat stack-list
```

Write down the **stack_name**, which is the second column of the command output.

5. Delete the compute node by UUID from the overcloud:

```
openstack overcloud node delete --stack $OSP_NAME $NOVA_UUID
```

Replace…

- **$OSP_NAME** with the `stack_name`from the previous step.
- **$NOVA_UUID** with the Nova UUID from the previous step.

Example

```
[stack@director ~]$ openstack overcloud node delete --stack
overcloud 6b2a2e71-f9c8-4d5b-aaf8-dada97c90821
deleting nodes [u'6b2a2e71-f9c8-4d5b-aaf8-dada97c90821'] from
stack overcloud
Started Mistral Workflow. Execution ID: 396f123d-df5b-4f37-
b137-83d33969b52b
```

6. Verify that the compute node was removed from the overcloud:

```
[stack@director ~]$ openstack server list
```

If the compute node was successfully removed, then it will not be listed in the above command output.

```
[stack@director ~]$ nova service-list
```

The removed Nova compute node's status will be **disabled** and **down**.

7. Verify that Ironic has powered off the node:

```
[stack@director ~]$ openstack baremetal node list
```

The compute node's power state and availability will be **`power off`** and **`available`** respectively. Write down the Nova compute service ID, which is the value in the first column of the above command output.

8. Remove the compute node from the **`nova-compute`** service from the Nova scheduler:

```
nova service-delete $COMPUTE_SERVICE_ID
```

Replace…

- **`$COMPUTE_SERVICE_ID`** with the Nova compute service ID from the previous step.

Example

```
[stack@director ~]$ nova service-delete 145
```

4.4. ADDTIONAL RESOURCES

- The Red Hat Ceph Storage Administration Guide.

CHAPTER 5. USING CEPH BLOCK DEVICE MIRRORING

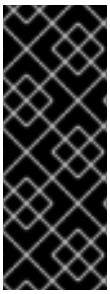
As a technician, you can mirror Ceph Block Devices to protect the data storage in the block devices.

5.1. PREREQUISITES

- A running Red Hat Ceph Storage cluster.
- Access to a Ceph client's command-line interface.

5.2. AN OVERVIEW OF CEPH BLOCK DEVICE MIRRORING

Ceph Block Device mirroring is a process of asynchronous replication of Ceph block device images between two or more Ceph clusters. Mirroring ensures point-in-time consistent replicas of all changes to an image, including reads and writes, block device resizing, snapshots, clones and flattening. Mirroring serves primarily for recovery from a disaster. Mirroring can run in either an active-passive or active-active configuration; that is, using mandatory exclusive locks and the journaling feature, Ceph records all modifications to an image in the order in which they occur. This ensures that a crash-consistent mirror of the remote image is available locally. Therefore, before an image can be mirrored to a peer cluster, you must enable journaling.



IMPORTANT

The CRUSH hierarchies supporting local and remote pools that mirror block device images SHOULD have the same capacity and performance characteristics, and SHOULD have adequate bandwidth to ensure mirroring without excess latency. For example, if you have X MiB/s average write throughput to images in the primary cluster, the network must support N * X throughput in the network connection to the secondary site, plus a safety factor of Y% to mirror N images.

The `rbd-mirror` Daemon

The **`rbd-mirror`** daemon is responsible for synchronizing images from one Ceph cluster to another. The **`rbd-mirror`** package provides the **`rbd-mirror`** daemon. Depending on the type of replication, **`rbd-mirror`** runs either on a single cluster or on all clusters that participate in mirroring:

One-way Replication

When data is mirrored from a primary cluster to a secondary cluster that serves as a backup,**`rbd-mirror`** runs ONLY on the backup cluster. RBD mirroring may have multiple secondary sites in an active-passive configuration.

Two-way Replication

When the data is mirrored from mirrored from a primary cluster to a secondary cluster and the secondary cluster can mirror back to the primary and each other, both clusters must have **`rbd-mirror`** running. Currently, two-way replication, also known as an active-active configuration, is supported only between two sites.

IMPORTANT

In two-way replication, each instance of **`rbd-mirror`** must be able to connect to the other Ceph cluster simultaneously. Additionally, the network must have sufficient bandwidth between the two data center sites to handle mirroring.

WARNING

Only run a single `rbd-mirror` daemon per a Ceph Storage cluster.

Mirroring Modes

Mirroring is configured on a per-pool basis within peer clusters. Red Hat Ceph Storage supports two modes, depending on what images in a pool are mirrored:

Pool Mode

All images in a pool with the journaling feature enabled are mirrored.

Image Mode

Only a specific subset of images within a pool is mirrored and you must enable mirroring for each image separately.

Image States

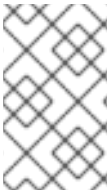
In an active-passive configuration, the mirrored images are:

- Primary
  - These mirrored images **can** be modified.
- Non-primary
  - These mirrored images **cannot** be modified.

Images are automatically promoted to primary when mirroring is first enabled on an image. Image promotion can happen implicitly or explicitly based on the mirroring mode. Image promotion happens implicitly when mirroring is enabled in pool mode. Image promotion happens explicitly when mirroring is enabled in image mode. It is also possible to demote primary images and promote non-primary images.

Asynchronous Red Hat Ceph Storage Updates

When doing a asynchronous update to a storage cluster using Ceph Block Device mirroring, follow the installation instruction for the update. After the update completes successfully, restart the Ceph Block Device instances.



NOTE

There is no required order for restarting the `ceph-rbd-mirror` instances. Red Hat recommends restarting the `ceph-rbd-mirror` instance pointing to the pool with primary images, followed by the `ceph-rbd-mirror` instance pointing to the mirrored pool.

Additional Resources

- See Section 5.4, “Enabling Ceph Block Device Journaling for Mirroring” for details.
- See Section 5.10, “Recovering From a Disaster” for details.
- See Section 5.5.2, “Enabling Mirroring on a Pool” for details.
- See Section 5.7, “Enabling Image Mirroring” for details.

5.3. CONFIGURING MIRRORING BETWEEN STORAGE CLUSTERS WITH THE SAME NAME

Creating Ceph Storage clusters using the same cluster name, by default the storage cluster name is **ceph**, can cause a challenge for Ceph Block Device mirroring. For example, some Ceph functions expect a storage cluster named of **ceph**. When both clusters have the same name, currently you must perform additional steps to configure **rbd-mirror**:

Prerequisites

- Two running Red Hat Ceph Storage clusters located at different sites.
- Access to the storage cluster or client node where the **rbd-mirror** daemon will be running.

Procedure

1. As **root**, on both storage clusters, specify the storage cluster name by adding the **CLUSTER** option to the appropriate file.

   #### Example

   ```
   CLUSTER=master
   ```

   #### Red Hat Enterprise Linux

   Edit the **/etc/sysconfig/ceph** file and add the **CLUSTER** option with the Ceph Storage cluster name as the value.

   #### Ubuntu

   Edit the **/etc/default/ceph** file and add the **CLUSTER** option with the Ceph Storage cluster name as the value.

2. As **root**, and only for the node running the **rbd-mirror** daemon, create a symbolic link to the **ceph.conf** file:

   ```
   [root@monitor ~]# ln -s /etc/ceph/ceph.conf /etc/ceph/master.conf
   ```

3. Now when referring to the storage cluster, use the symbolic link name with the **--cluster** flag.

   #### Example

   ```
   --cluster master
   ```

5.4. ENABLING CEPH BLOCK DEVICE JOURNALING FOR MIRRORING

There are two ways to enable the Ceph Block Device journaling feature:

- On image creation.
- Dynamically on already existing images.

IMPORTANT

Journaling depends on the **`exclusive-lock`** feature which must be enabled too.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- Access to a Ceph client command-line interface.

Procedure

Enable on Image Creation

1. As a normal user, execute the following command to enable journaling on image creation creating:

```
rbd create $IMAGE_NAME --size $MEGABYTES --pool $POOL_NAME --image-
feature $FEATURE_NAME[,$FEATURE_NAME]
```

Example

```
[user@rbd-client ~]$ rbd create image-1 --size 1024 --pool pool-1 --
image-feature exclusive-lock,journaling
```

Enable on an Existing Image

1. As a normal user, execute the following command to enable journaling on an already existing image:

```
rbd feature enable $POOL_NAME/$IMAGE_NAME $FEATURE_NAME
```

Example

```
[user@rbd-client ~]$ rbd feature enable pool-1/image-1 exclusive-
lock
[user@rbd-client ~]$ rbd feature enable pool-1/image-1 journaling
```

Setting the Default

1. To enable journaling on all new images by default, add the following line to the Ceph configuration file, **`/etc/ceph/ceph.conf`** by default:

```
rbd default features = 125
```

Additional Resources

- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide for more details:
  - Red Hat Enterprise Linux
  - Ubuntu

5.5. CONFIGURING CEPH BLOCK DEVICE MIRRORING ON A POOL

As a technician, you can enable or disable mirroring on a pool, add or remove a cluster peer, and view information on peers and pools.

- Section 5.5.2, “Enabling Mirroring on a Pool”
- Section 5.5.3, “Disabling Mirroring on a Pool”
- Section 5.5.4, “Adding a Cluster Peer”
- Section 5.5.5, “Removing a Cluster Peer”
- Section 5.5.6, “Viewing Information About the Cluster Peers”
- Section 5.5.7, “Viewing Mirroring Status for a Pool”

5.5.1. Prerequisites

- A running Red Hat Ceph Storage cluster.
- Access to a Ceph client command-line interface.

5.5.2. Enabling Mirroring on a Pool

When enabling mirroring on an object pool, you must specify which mirroring mode to use.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- Access to a Ceph client command-line interface.
- An existing object pool.

Procedure

1. As a normal user, execute the following command to enable mirroring on a pool:

```
rbd mirror pool enable $POOL_NAME $MODE
```

Example

To enable pool mode:

```
[user@rbd-client ~]$ rbd mirror pool enable data pool
```

To enable image mode:

```
[user@rbd-client ~]$ rbd mirror pool enable data image
```

Additional Resources

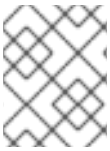
- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide for more details:

- Red Hat Enterprise Linux
- Ubuntu

- See the section called “Mirroring Modes” for details.

5.5.3. Disabling Mirroring on a Pool

Before disabling mirroring on a pool, you must remove the cluster peer.



NOTE

Disabling mirroring on a pool, also disables mirroring on any images within the pool for which mirroring was enabled separately in image mode.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- Access to a Ceph client command-line interface.
- Removed as a cluster peer.
- An existing object pool.

Procedure

1. As a normal user, execute the following command to disable mirroring on a pool:

```
rbd mirror pool disable $POOL_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror pool disable data
```

Additional Resources

- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide for more details:
  - Red Hat Enterprise Linux
  - Ubuntu
- See Section 5.5.5, “Removing a Cluster Peer” for details.

5.5.4. Adding a Cluster Peer

In order for the **rbd-mirror** daemon to discover its peer cluster, you must register the peer to the pool.

Prerequisites

- Two running Red Hat Ceph Storage clusters located at different sites.
- Access to a Ceph client command-line interface.

- An existing object pool.

Procedure

1. As a normal user, execute the following command to add a cluster peer:

```
rbd --cluster $CLUSTER_NAME mirror pool peer add $POOL_NAME
$CLIENT_NAME@$TARGET_CLUSTER_NAME
```

Examples

Adding the **remote** cluster as a peer to the **local** cluster:

```
[user@rbd-client ~]$ rbd --cluster local mirror pool peer add data
client.remote@remote
```

Additional Resources

- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide for more details:
  - Red Hat Enterprise Linux
  - Ubuntu

5.5.5. Removing a Cluster Peer

In order for the **rbd-mirror** daemon to discover its peer cluster, you must register the peer to the pool.

Prerequisites

- Two running Red Hat Ceph Storage clusters located at different sites.
- Access to a Ceph client command-line interface.
- An existing cluster peer.

Procedure

1. Record the peer's Universally Unique Identifier (UUID) for use in the next step. To view the peer's UUID, execute the following command as a normal user:

```
rbd mirror pool info $POOL_NAME
```

2. As a normal user, execute the following command to remove a cluster peer:

```
rbd mirror pool peer remove $POOL_NAME $PEER_UUID
```

Example

```
[user@rbd-client ~]$ rbd mirror pool peer remove data 55672766-c02b-
4729-8567-f13a66893445
```

Additional Resources

- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide

for more details:

- Red Hat Enterprise Linux
- Ubuntu

5.5.6. Viewing Information About the Cluster Peers

You can view basic information about the cluster peers by doing this procedure.

Prerequisites

- Two running Red Hat Ceph Storage clusters located at different sites.
- Access to a Ceph client command-line interface.
- An existing cluster peer.

Procedure

1. As a normal user, execute the following command to view information about the cluster peers:

```
rbd mirror pool info $POOL_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror pool info data
Enabled: true
Peers:
  UUID                                 NAME        CLIENT
    786b42ea-97eb-4b16-95f4-867f02b67289 ceph-remote client.admin
```

Additional Resources

- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide for more details:
  - Red Hat Enterprise Linux
  - Ubuntu

5.5.7. Viewing Mirroring Status for a Pool

You can view the Ceph Block Device mirroring status for a pool by doing this procedure.

Prerequisites

- Access to a Ceph client command-line interface.
- An existing cluster peer.
- An existing object storage pool.

Procedure

1. As a normal user, execute the following command to view the mirroring status for a pool:

```
rbd mirror pool status <pool-name>
```

Example

```
[user@rbd-client ~]$ rbd mirror pool status data
health: OK
images: 1 total
```

NOTE

To output more details for every mirrored image in a pool, use the **`--verbose`** option.

Additional Resources

- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide for more details:
  - Red Hat Enterprise Linux
  - Ubuntu

5.5.8. Additional Resources

- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide for more details:
  - Red Hat Enterprise Linux
  - Ubuntu

5.6. CONFIGURING CEPH BLOCK DEVICE MIRRORING ON AN IMAGE

As a technician, you can enable or disable mirroring on an image, add or remove a cluster peer, and view information on peers and pools.

5.6.1. Prerequisites

- A running Red Hat Ceph Storage cluster.
- Access to a Ceph client command-line interface.

5.7. ENABLING IMAGE MIRRORING

This procedure enables Ceph Block Device mirroring on images.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- Access to a Ceph client command-line interface.
- An existing image.

Procedure

1. As a normal user, execute the following command to enable mirroring on an image:

```
rbd mirror image enable $POOL_NAME/$IMAGE_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror image enable data/image2
```

Additional Resources

- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide for more details:
  - Red Hat Enterprise Linux
  - Ubuntu
- See the section called “Mirroring Modes” for more details.

included::modules/procedure_rbd_disabling-image-mirroring_en-us.adoc[leveloffset=+1]

included::modules/procedure_rbd_promoting-and-demoting-an-image_en-us.adoc[leveloffset=+1]

included::modules/procedure_rbd_resynchronizing-an-image_en-us.adoc[leveloffset=+1]

included::modules/procedure_rbd_viewing-the-mirroring-status-for-a-single-image_en-us.adoc[leveloffset=+1]

5.7.1. Additional Resources

- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide for more details:
  - Red Hat Enterprise Linux
  - Ubuntu

5.8. CONFIGURING TWO-WAY MIRRORING FOR CEPH BLOCK DEVICES

Two-way mirroring is an effective active-active mirroring solution suitable for automatic failover.

Prerequisites

- Two running Red Hat Ceph Storage clusters located at different sites.
  - Each storage cluster has the corresponding configuration files in the **`/etc/ceph/`** directory.
- One Ceph client, with a connection to both storage clusters.
  - Access to the Ceph client’s command-line interface.
- An existing object storage pool and an image.

- The same object pool name exists on each storage cluster.

Procedure

1. Verify that all images within the object storage pool have **`exclusive-lock`** and **`journaling`** enabled:

```
rbd info $POOL_NAME/$IMAGE_NAME
```

Example

```
[user@rbd-client ~]$ rbd info data/image1
```

2. The **`rbd-mirror`** package is provided by the Red Hat Ceph Storage Tools repository. As **`root`**, on a Ceph Monitor node of the local and the remote storage clusters, install the **`rbd-mirror`** package:
Red Hat Enterprise Linux

```
[root@monitor-remote ~]# yum install rbd-mirror
```

Ubuntu

```
[user@monitor-remote ~]$ sudo apt-get install rbd-mirror
```

NOTE

The **`rbd-mirror`** daemon can run on any node in the storage cluster. It does not have to be a Ceph Monitor or OSD node. However, only one **`rbd-mirror`** daemon per storage cluster.

3. As **`root`**, on both storage clusters, specify the storage cluster name by adding the **`CLUSTER`** option to the appropriate file.

Example

```
CLUSTER=local
```

Red Hat Enterprise Linux

Edit the **`/etc/sysconfig/ceph`** file and add the **`CLUSTER`** option with the Ceph Storage cluster name as the value.

Ubuntu

Edit the **`/etc/default/ceph`** file and add the **`CLUSTER`** option with the Ceph Storage cluster name as the value.

NOTE

See the procedure on handling Ceph Block Device mirroring between two Ceph Storage clusters with the same name.

4. As a normal user, on both storage clusters, create users with permissions to access the object storage pool and output their keyrings to a file.

```
ceph auth get-or-create client.$STORAGE_CLUSTER_NAME mon 'profile
rbd' osd 'profile rbd pool=$POOL_NAME' -o $PATH_TO_KEYRING_FILE --
cluster $STORAGE_CLUSTER_NAME
```

   a. On the Ceph Monitor node in the local storage cluster, create the **client.local** user and output the keyring to the **local.client.local.keyring** file:

      Example

```
[user@monitor-local ~]$ ceph auth get-or-create client.local mon
'profile rbd' osd 'profile rbd pool=data' -o
/etc/ceph/local.client.local.keyring --cluster local
```

   b. On the Ceph Monitor node in the remote storage cluster, create the **client.remote** user and output the keyring to the **remote.client.remote.keyring** file:

      Example

```
[user@monitor-remote ~]$ ceph auth get-or-create client.remote
mon 'profile rbd' osd 'profile rbd pool=data' -o
/etc/ceph/remote.client.remote.keyring --cluster remote
```

5. As **root**, copy the Ceph configuration file and the newly created keyring file for each storage cluster between each storage cluster and to any Ceph client nodes in the both storage clusters.

```
scp $PATH_TO_STORAGE_CLUSTER_CONFIG_FILE_NAME
$SSH_USER_NAME@$MON_NODE:/etc/ceph/
scp $PATH_TO_STORAGE_CLUSTER_KEYRING_FILE_NAME
$SSH_USER_NAME@$CLIENT_NODE:/etc/ceph/
```

Copying Local to Remote Example

```
[root@monitor-local ~]# scp /etc/ceph/local.conf
example@remote:/etc/ceph/
[root@monitor-local ~]# scp /etc/ceph/local.client.local.keyring
example@remote:/etc/ceph/
```

Copying Remote to Local Example

```
[root@monitor-remote ~]# scp /etc/ceph/remote.conf
example@local:/etc/ceph/
[root@monitor-remote ~]# scp /etc/ceph/remote.client.remote.keyring
example@local:/etc/ceph/
```

Copying both Local and Remote to Clients Example

```
[root@monitor-local ~]# scp /etc/ceph/local.conf example@rbd-
client:/etc/ceph/
[root@monitor-local ~]# scp /etc/ceph/local.client.local.keyring
```

```
example@rbd-client:/etc/ceph/
```

```
[root@monitor-remote ~]# scp /etc/ceph/remote.conf example@rbd-
client:/etc/ceph/
[root@monitor-remote ~]# scp /etc/ceph/remote.client.remote.keyring
example@rbd-client:/etc/ceph/
```

6. As **root**, on the Ceph Monitor node of both storage clusters, enable and start the **rbd-mirror** daemon:

```
systemctl enable ceph-rbd-mirror.target
systemctl enable ceph-rbd-mirror@$CLIENT_ID
systemctl start ceph-rbd-mirror@$CLIENT_ID
```

The **$CLIENT_ID** is the Ceph Storage cluster user that the **rbd-mirror** daemon will use.

Example:

```
[root@monitor-remote ~]# systemctl enable ceph-rbd-mirror.target
[root@monitor-remote ~]# systemctl enable ceph-rbd-mirror@remote
[root@monitor-remote ~]# systemctl start ceph-rbd-mirror@remote
```

NOTE

The **$CLIENT_ID** user must have the appropriate **cephx** authentication access to the storage cluster.

Configuring Two Way Mirroring for Pool Mode

1. As a normal user, from any Ceph client node that has access to each storage cluster, enable pool mirroring of the object storage pool residing on both storage clusters:

```
rbd mirror pool enable $POOL_NAME $MIRROR_MODE --cluster
$STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror pool enable data pool --cluster
local
[user@rbd-client ~]$ rbd mirror pool enable data pool --cluster
remote
```

a. Verify that mirroring has been successfully enabled:

```
rbd mirror pool status $POOL_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror pool status data
health: OK
images: 1 total
```

2. As a normal user, add the storage clusters as a peer of the other storage cluster:

```
rbd mirror pool peer add $POOL_NAME
$CLIENT_NAME@$STORAGE_CLUSTER_NAME --cluster
$PEER_STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror pool peer add data
client.local@local --cluster remote
[user@rbd-client ~]$ rbd mirror pool peer add data
client.remote@remote --cluster local
```

a. Verify that the storage cluster peer was successfully added:

Example

```
[user@rbd-client ~]$ rbd mirror pool info data --cluster local
Mode: pool
Peers:
  UUID                                 NAME  CLIENT
  de32f0e3-1319-49d3-87f9-1fc076c83946 remote client.remote

[user@rbd-client ~]$ rbd mirror pool info data --cluster remote
Mode: pool
Peers:
  UUID                                 NAME   CLIENT
  87ea0826-8ffe-48fb-b2e8-9ca81b012771 local client.local
```

Configuring One Way Mirroring for Image Mode

1. As a normal user, enable image mirroring of the object storage pool on both storage clusters:

```
rbd mirror pool enable $POOL_NAME $MIRROR_MODE --cluster
$STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror pool enable data image --cluster
local
[user@rbd-client ~]$ rbd mirror pool enable data image --cluster
remote
```

a. Verify that mirroring has been successfully enabled:

```
rbd mirror pool status $POOL_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror pool status data
health: OK
images: 1 total
```

2. As a normal user, add the storage clusters as a peer of the other storage cluster:

```
rbd mirror pool peer add $POOL_NAME
$CLIENT_NAME@$STORAGE_CLUSTER_NAME --cluster
$PEER_STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror pool peer add data
client.local@local --cluster remote
[user@rbd-client ~]$ rbd mirror pool peer add data
client.remote@remote --cluster local
```

a. Verify that the storage cluster peer was successfully added:

```
rbd mirror pool info --cluster $STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror pool info --cluster remote
Mode: image
Peers:
  UUID                                 NAME   CLIENT
  87ea0826-8ffe-48fb-b2e8-9ca81b012771 local client.local

[user@rbd-client ~]$ rbd mirror pool info --cluster local
Mode: image
Peers:
  UUID                                 NAME   CLIENT
  de32f0e3-1319-49d3-87f9-1fc076c83946 remote client.remote
```

3. As a normal user, on the local storage cluster, explicitly enable mirroring for the images:

```
rbd mirror image enable $POOL_NAME/$IMAGE_NAME --cluster
$STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror image enable data/image1 --cluster
local
Mirroring enabled
```

a. Verify that mirroring has been successfully enabled:

```
[user@rbd-client ~]$ rbd mirror image status data/image1 --
cluster local
image1:
  global_id:   2c928338-4a86-458b-9381-e68158da8970
  state:       up+replaying
  description: replaying, master_position=[object_number=6,
tag_tid=2,
entry_tid=22598], mirror_position=[object_number=6, tag_tid=2,
entry_tid=29598], entries_behind_master=0
```

```
    last_update: 2018-04-28 18:47:39

[user@rbd-client ~]$ rbd mirror image status data/image1 --
cluster remote
image1:
  global_id:   2c928338-4a86-458b-9381-e68158da8970
  state:       up+replaying
  description: replaying, master_position=[object_number=6,
tag_tid=2,
entry_tid=22598], mirror_position=[object_number=6, tag_tid=2,
entry_tid=29598], entries_behind_master=0
  last_update: 2018-04-28 18:47:39
```

Additional Resources

- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide for more details:
  - Red Hat Enterprise Linux
  - Ubuntu
- See the *Pools* chapter in the *Storage Strategies* guide for more details.
- See the *User Management* chapter in the *Administration Guide* for more details.
- See Section 5.5, “Configuring Ceph Block Device Mirroring on a Pool” for more details.
- See Section 5.6, “Configuring Ceph Block Device Mirroring on an Image” for more details.
- See Section 5.4, “Enabling Ceph Block Device Journaling for Mirroring” for more details.

5.9. DELAYING REPLICATION BETWEEN STORAGE CLUSTERS

Whether you are using one- or two-way replication, you can delay replication between Ceph Block Device mirroring images. You can implement a replication delay strategy as a cushion of time before unwanted changes to the primary image are propagated to the replicated secondary image. The replication delay can be configured globally or on individual images and must be configured on the destination storage cluster.

Prerequisites

- Two running Red Hat Ceph Storage clusters located at different sites.
- Access to the storage cluster or client node where the **rbd-mirror** daemon will be running.

Procedure

Setting the Replication Delay Globally

1. As **root**, edit the Ceph configuration file, on the node running the **rbd-mirror** daemon, and add the following line:

   ```
   rbd_mirroring_replay_delay = $MINIMUM_DELAY_IN_SECONDS
   ```

   Example

```
rbd_mirroring_replay_delay = 600
```

Setting the Replication Delay on an Image

1. As a normal user, on a Ceph client node, set the replication delay for a specific primary image, execute the following command:

```
rbd image-meta set $POOL_NAME/$IMAGE_NAME conf_rbd_mirroring_replay_delay
$MINIMUM_DELAY_IN_SECONDS
```

+ .Example

```
[user@rbd-client ~]$ rbd image-meta set data/image1
conf_rbd_mirroring_replay_delay 600
```

Additional Resources

- See Section 5.2, “An Overview of Ceph Block Device Mirroring” for more details.

5.10. RECOVERING FROM A DISASTER

The following procedure shows how to failover to the mirrored data on a secondary storage cluster after the primary storage cluster terminated in an orderly or non-orderly manner.

Prerequisites

- Two running Red Hat Ceph Storage clusters located at different sites.
- One Ceph client, with a connection to both storage clusters.
  - Access to the Ceph client’s command-line interface.

Procedure

Failover After an Orderly Shutdown

1. Stop all clients that use the primary image. This step depends on which clients are using the image.
2. As a normal user, on a Ceph client node, demote the primary image located on the local storage cluster:

```
rbd mirror image demote $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror image demote data/image1 --
cluster=local
```

3. As a normal user, on a Ceph client node, promote the non-primary image located on the remote storage cluster:

```
rbd mirror image promote $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror image promote data/image1 --
cluster=remote
```

4. Resume the access to the peer image. This step depends on which clients are using the image.

Failover After a Non-Orderly Shutdown

1. Verify that the primary storage cluster is down.
2. Stop all clients that use the primary image. This step depends on which clients are using the image.
3. As a normal user, on a Ceph client node, promote the non-primary image located on the remote storage cluster. Use the **`--force`** option, because the demotion cannot be propagated to the local storage cluster:

```
rbd mirror image promote --force $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

Example

```
$ rbd mirror image promote --force data/image1 --cluster=remote
```

4. Resume the access to the peer image. This step depends on which clients are using the image.

Failing Back to the Primary Storage Cluster

1. Verify the primary storage cluster is available.
2. If there was a non-orderly shutdown, as a normal user, on a Ceph client node, demote the primary image located on the local storage cluster:

```
rbd mirror image demote $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror image demote data/image1 --
cluster=local
```

3. Resynchronize the image **ONLY** if there was a non-orderly shutdown. As a normal user, on a Ceph client node, resynchronize the image:

```
rbd mirror image resync $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror image resync data/image1 --
cluster=local
```

4. Verify that resynchronization is complete and is in the **`up+replaying`** state. As a normal user, on a Ceph client node, check the resynchronization status of the image:

```
rbd mirror image status $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror image status data/image1 --
cluster=local
```

5. As a normal user, on a Ceph client node, demote the secondary image located on the remote storage cluster:

```
rbd mirror image demote $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

Example

```
[user@rbd-client ~]$ rbd mirror image demote data/image1 --
cluster=remote
```

6. As a normal user, on a Ceph client node, promote the formerly primary image located on the local storage cluster:

```
rbd mirror image promote $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

Example

```
$ rbd mirror image promote data/image1 --cluster=local
```

Additional Resources

- See the *Block Storage and Volumes* chapter in the *Storage Guide* for the Red Hat OpenStack Platform.

5.11. ADDITIONAL RESOURCES

- See the *Installing the Ceph Client Role* section in the Red Hat Ceph Storage Installation Guide for more details:
  - Red Hat Enterprise Linux
  - Ubuntu