

HP Helion OpenStack 1.1 and 1.1.1: Configuring Ceph Cluster and Client Nodes using Ansible Playbooks

Contents

HP Helion OpenStack® 1.1 and 1.1.1: Configuring Ceph Cluster and Client Nodes using Ansible Playbooks.....3

HP Helion OpenStack® 1.1 and 1.1.1: Configuring Ceph Cluster and Client Nodes using Ansible Playbooks

This section covers the following topics:

1. [Installing Ceph](#)
2. [Scaling Ceph cluster](#)
 - a. [Add Monitor](#)
 - b. [Add OSD](#)
3. [Heartbeat Monitoring Tool \(Optional\)](#)
4. [Helion OpenStack Ceph cluster validation](#)
5. [Helion OpenStack Ceph Client Validation](#)
 - a. [Glance](#)
 - b. [Ceph Glance Clone Copy On Write](#)
 - c. [Cinder](#)
 - d. [Creating a Cinder Backup](#)
6. [Volume Snapshots](#)
7. [Ceph RADOS Gateway Validation](#)

Installing the Ceph Solution

To install the Ceph solution:

1. Copy and untar the `helion-ceph.tar` file on one of the provisioned nodes identified as `ceph-admin` under `/helion-ceph/ansible-playbooks`. If `/helion-ceph/` does not exist, then create a directory named `/helion-ceph/`.

Unpacking the tar file creates the `cephconfiguration` directory under `/helion-ceph`.

2. Ensure the client setup scripts tar ball (also containing the Ceph Debian distribution for installing on Helion nodes) has been copied under `/helion-ceph/cephconfiguration/ansible-playbooks/roles/ceph-client/files`.

For example:

```
`/helion-ceph/cephconfiguration/ansible-playbooks/roles/ceph-client/
files/ceph_client_setup-0.80.7_h1.1.fix6_newdebs.tar`
```

3. Once the nodes are provisioned (for more information on nodes provisioning, refer [Running the Provisioning Tool](#)), create the following configuration files under `/helion-ceph/cephconfiguration/ansible-playbooks/` on the `ceph-admin` node

cephcluster.csv:

```
<ip-address>, <ceph-identifier>, <type>,
<user>,<osd-partition-type [osd nodes only]>, <osd-disk-path [osd nodes
only]>, <journal-partition-type[osd nodes only]>, <journal-disk-path [osd
nodes only]>
ip-address
```

where:

node ip: is the address used by configuration process to set up `ceph-cluster`. This can either be a PXE, management or some other network.

ceph-identifier: is the unique name to identify an OSD, mon, admin or RADOS. This will be used by the bootstrap script to generate Ansible metadata.

IMPORTANT: The RADOS nodes must be given the identifier `rados-1` and `rados-2`.

Type: can be `mon-master`, `mon`, `osd`, `admin`, `radosgw`, `radosgw-master`, `controllers`, `computes`

IMPORTANT: Ensure that `mon-master` is the first node to be defined in `cephcluster.csv`.

Also, if there is only one RADOS node, then the type must be `radosgw-master`.

user: used to login to Ceph nodes for performing the configuration. In this case it is **hlinux**. (**hlinux** refers to HP Linux for HP Helion OpenStack)

osd-partition-type: partition type to be used while formatting the disk for OSD storage (This attribute exists only for OSD type node).

osd-disk-path: disk to be used for OSD storage (This attribute exists only for OSD type nodes) . This will be an entire physical disk to be used as a data disk. For example: `/dev/sda` or it can be a partition on a large disk. For example: `/dev/sda5`

journal-partition-type: partition type to be used while formatting the disk (SSD) for journaling (This attribute exists only for OSD type nodes). For journaling make sure the entire disk is specified. For example: `/dev/sde`. Do not use a partition for journaling. For example: `/dev/sde1` is not allowed.

journal-disk-path: disk to be used for journaling (This attribute exists only for OSD type nodes).

The following is a sample:

```
192.168.51.90,mon-master-1,mon-master,hlinux
192.168.51.100,mon2,mon,hlinux
192.168.51.102,mon3,mon,hlinux
192.168.51.98,admin-1,admin,hlinux
192.168.51.93,rados-1,radosgw-master,hlinux
192.168.51.94,rados-2,radosgw,hlinux
192.168.51.95,ceph-osd-1,osd,hlinux,xfs,/dev/sdc,xfs,/dev/sdb
192.168.51.95,ceph-osd-2,osd,hlinux,xfs,/dev/sdd,xfs,/dev/sdb
192.168.51.95,ceph-osd-3,osd,hlinux,xfs,/dev/sde,xfs,/dev/sdb
192.168.51.95,ceph-osd-4,osd,hlinux,xfs,/dev/sdf,xfs,/dev/sdb
192.168.51.95,ceph-osd-5,osd,hlinux,xfs,/dev/sdg,xfs,/dev/sdb
192.168.51.95,ceph-osd-6,osd,hlinux,xfs,/dev/sdh,xfs,/dev/sdb
192.168.51.95,ceph-osd-7,osd,hlinux,xfs,/dev/sdi,xfs,/dev/sdb
192.168.51.96,ceph-osd-8,osd,hlinux,xfs,/dev/sdc,xfs,/dev/sdb
192.168.51.96,ceph-osd-9,osd,hlinux,xfs,/dev/sdd,xfs,/dev/sdb
192.168.51.96,ceph-osd-10,osd,hlinux,xfs,/dev/sde,xfs,/dev/sdb
192.168.51.96,ceph-osd-11,osd,hlinux,xfs,/dev/sdf,xfs,/dev/sdb
192.168.51.96,ceph-osd-12,osd,hlinux,xfs,/dev/sdg,xfs,/dev/sdb
192.168.51.96,ceph-osd-13,osd,hlinux,xfs,/dev/sdh,xfs,/dev/sdb
192.168.51.96,ceph-osd-14,osd,hlinux,xfs,/dev/sdi,xfs,/dev/sdb
192.168.51.97,ceph-osd-15,osd,hlinux,xfs,/dev/sdc,xfs,/dev/sdb
192.168.51.97,ceph-osd-16,osd,hlinux,xfs,/dev/sdd,xfs,/dev/sdb
192.168.51.97,ceph-osd-17,osd,hlinux,xfs,/dev/sde,xfs,/dev/sdb
192.168.51.97,ceph-osd-18,osd,hlinux,xfs,/dev/sdf,xfs,/dev/sdb
192.168.51.97,ceph-osd-19,osd,hlinux,xfs,/dev/sdg,xfs,/dev/sdb
192.168.51.97,ceph-osd-20,osd,hlinux,xfs,/dev/sdh,xfs,/dev/sdb
192.168.51.97,ceph-osd-21,osd,hlinux,xfs,/dev/sdi,xfs,/dev/sdb
192.168.51.88,compute0,computes
192.168.51.89,compute1,computes
192.168.51.87,controller0,controllers
192.168.51.86,controller1,controllers
192.168.51.85,controller2,controllers
```

4. Edit the `group_vars` as per your set up:

```
/group_vars/all
```

```
#
cephmon_user: root
cephmon_group: root
runrados: 0 # Set this to 0 if you do not have rados nodes, to 1 if you
  have the rados nodes.
radosgwHA: 1 # Set this to 1 if you want to setup rados in HA mode where
  you need min two rados nodes. If you have only 1 node set this value to
  0.
secretuuid: 457eb676-33da-42ec-9a8c-9293d545c337 # This the UUID that will
  be used to setup the helion nodes. Change this prior to running the ceph-
  client and ceph-admin roles, if you wish to newly generated UUID. The same
  UUID will work too.
clienttarname: ceph_client_setup-0.80.7_h1.1.fix6_newdebs.tar # Set this
  to the tar ball name that is being used for helion client setup. Make
  sure the tarball has been copied under roles/ceph-client/files folder
rados_1_fqdn: # Set the actual fqdn of the rados 1 node here
rados_2_fqdn: # Set the actual fqdn of the rados 2 node here
/group_vars/ceph-cluster
---
---
# Variables here are applicable to the ceph-cluster host group
osd_journal_size: 1024
mon_master: c1mon1-overcloud-ceph-cluster
fsid: 514bb61e-80e1-11e4-9461-000c2966c4ff
fssize: 2048
env: baremetal
journal: 1
dependencies:
```



Note: The fsid (File System ID) is a unique identifier for the cluster. Execute `uuidgen` command to generate fsid.

5. Copy the `cephadmin.pem` key file to the Seed node.
6. Copy the public key of `heat-admin` to enable SSH from the `ceph-admin` node to Helion nodes by entering:
 - a. Start from the KVM host after the helion and Ceph nodes are up.
 - b. From the root shell on the KVM host, SSH to the seed node, by entering:

```
`ssh <IP of seed>`
```

c. Enter: `scp -i <keypair> /root/.ssh/id_rsa hlinux@<IP of ceph-admin>:/home/hlinux/seed-id_rsa.private`

d. Enter: `scp -i <keypair> /root/.ssh/id_rsa.pub hlinux@<IP of ceph-admin>:/home/hlinux/seed-id_rsa.public`

e. Copy the `seed-id_rsa.private` under the `ansible-playbooks` folder on your admin node.

f. Copy the `seed-id_rsa.public` under the `ansible-playbooks` folder on your admin node.

7. If journaling is enabled, in `/group_vars/ceph-cluster`, run following scripts from the admin node from `helion-ceph/ansible-playbooks/`

```
sudo ./createjournalpartitions <cephadmin keypair>
```

This script will create partitions on the journal drive, as specified `cephcluster.csv` and appends the partition number and journal size to each node in this file.

```
<ipaddress>, <ceph-identifier>, <node-type>,<user>,<osd-partition-type>,<osd-disk-path>,<journal-partition-type>,<journal-disk-path><partition no>,<
```

The following is sample output:

```
192.168.51.90,mon-master-1,mon-master,hlinux
192.168.51.100,mon2,mon,hlinux
192.168.51.102,mon3,mon,hlinux
192.168.51.98,admin-1,admin,hlinux
192.168.51.93,rados-1,radosgw-master,hlinux
192.168.51.94,rados-2,radosgw,hlinux
192.168.51.95,ceph-osd-1,osd,hlinux,xfs,/dev/sdc,xfs,/dev/sdb5
192.168.51.95,ceph-osd-2,osd,hlinux,xfs,/dev/sdd,xfs,/dev/sdb6
192.168.51.95,ceph-osd-3,osd,hlinux,xfs,/dev/sde,xfs,/dev/sdb7
192.168.51.95,ceph-osd-4,osd,hlinux,xfs,/dev/sdf,xfs,/dev/sdb8
192.168.51.95,ceph-osd-5,osd,hlinux,xfs,/dev/sdg,xfs,/dev/sdb9
192.168.51.95,ceph-osd-6,osd,hlinux,xfs,/dev/sdh,xfs,/dev/sdb10
192.168.51.95,ceph-osd-7,osd,hlinux,xfs,/dev/sdi,xfs,/dev/sdb11
192.168.51.96,ceph-osd-8,osd,hlinux,xfs,/dev/sdc,xfs,/dev/sdb5
192.168.51.96,ceph-osd-9,osd,hlinux,xfs,/dev/sdd,xfs,/dev/sdb6
192.168.51.96,ceph-osd-10,osd,hlinux,xfs,/dev/sde,xfs,/dev/sdb7
192.168.51.96,ceph-osd-11,osd,hlinux,xfs,/dev/sdf,xfs,/dev/sdb8
192.168.51.96,ceph-osd-12,osd,hlinux,xfs,/dev/sdg,xfs,/dev/sdb9
192.168.51.96,ceph-osd-13,osd,hlinux,xfs,/dev/sdh,xfs,/dev/sdb10
192.168.51.96,ceph-osd-14,osd,hlinux,xfs,/dev/sdi,xfs,/dev/sdb11
192.168.51.97,ceph-osd-15,osd,hlinux,xfs,/dev/sdc,xfs,/dev/sdb5
192.168.51.97,ceph-osd-16,osd,hlinux,xfs,/dev/sdd,xfs,/dev/sdb6
192.168.51.97,ceph-osd-17,osd,hlinux,xfs,/dev/sde,xfs,/dev/sdb7
192.168.51.97,ceph-osd-18,osd,hlinux,xfs,/dev/sdf,xfs,/dev/sdb8
192.168.51.97,ceph-osd-19,osd,hlinux,xfs,/dev/sdg,xfs,/dev/sdb9
192.168.51.97,ceph-osd-20,osd,hlinux,xfs,/dev/sdh,xfs,/dev/sdb10
192.168.51.97,ceph-osd-21,osd,hlinux,xfs,/dev/sdi,xfs,/dev/sdb11
192.168.51.88,compute0,computes
192.168.51.89,compute1,computes
192.168.51.87,controller0,controllers
192.168.51.86,controller1,controllers
192.168.51.85,controller2,controllers
```



Note: Before running this script, make sure that no partition numbers are associated with <journal-disk-path> in the `cephcluster.csv` file. This script will append the partition numbers once it is executed successfully.

8. Execute the following command.

```
sudo ./bootstrap.sh <cephadmin keypair>
```

This script creates Ansible metadata from the `cephcluster.csv` manifest file, and configures the `/etc/hosts` file on Ceph nodes.

After the bootstrap is run, verify that the `hosts` file and `host_vars/*` files are generated.

The following is a sample host file:

```
[mon]
mon-2 ansible_ssh_private_key_file=clcephadmin.pem
mon-3 ansible_ssh_private_key_file=clcephadmin.pem
[osd]
ceph-osd-1 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-2 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-3 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-4 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-5 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-6 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-7 ansible_ssh_private_key_file=clcephadmin.pem

ceph-osd-8 ansible_ssh_private_key_file=clcephadmin.pem
```

```

ceph-osd-9 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-10 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-11 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-12 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-13 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-14 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-15 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-16 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-17 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-18 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-19 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-20 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-21 ansible_ssh_private_key_file=clcephadmin.pem

[ceph-cluster]
mon-master-1 ansible_ssh_private_key_file=clcephadmin.pem
admin-1 ansible_ssh_private_key_file=clcephadmin.pem
rados-1 ansible_ssh_private_key_file=clcephadmin.pem
rados-2 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-1 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-2 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-3 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-4 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-5 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-6 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-7 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-8 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-9 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-10 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-11 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-12 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-13 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-14 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-15 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-16 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-17 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-18 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-19 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-20 ansible_ssh_private_key_file=clcephadmin.pem
ceph-osd-21 ansible_ssh_private_key_file=clcephadmin.pem

[mon-master]
mon-master-1 ansible_ssh_private_key_file=clcephadmin.pem

[admin]
admin-1 ansible_ssh_private_key_file=clcephadmin.pem

[radosgw]
rados-1 ansible_ssh_private_key_file=clcephadmin.pem
rados-2 ansible_ssh_private_key_file=clcephadmin.pem

[computes]
compute0 ansible_ssh_private_key_file=seed-id_rsa.private
compute1 ansible_ssh_private_key_file=seed-id_rsa.private

[helionnodes:children]
computes
controllers

[controllers]
controller0 ansible_ssh_private_key_file=seed-id_rsa.private
controller1 ansible_ssh_private_key_file=seed-id_rsa.private
controller2 ansible_ssh_private_key_file=seed-id_rsa.private

```

Running the Ansible playbook

Make sure that the following three keys are present or copied from their respective nodes to the admin node under `ansible-playbook` folder:

- `cephadmin.pem` key is copied (from Undercloud) to the `ansible-playbooks` folder.
- `seed-id_rsa.private` key copied from the seed to the `ansible-playbooks` folder.
- `seed-id_rsa.public` key is copied from the seed to the `ansible-playbooks` folder.



Note: All three key ring permissions should be set to **`sudo chmod 600 <key>`**

```
4 -rw----- 1 hlinux hlinux 1676 Feb 24 07:46 cephadmin.pem
4 -rw----- 1 hlinux hlinux 1675 Feb 24 07:45 seed-id_rsa.private
4 -rw----- 1 hlinux hlinux 393 Feb 24 07:45 seed-id_rsa.public
```

Ceph is deployed in three different stages

You can deploy Ceph in three different stages:

1. Set up Monitors and OSD nodes.
2. Integrate Ceph on Helion nodes
3. Set up RADOS Gateway nodes in the Ceph cluster

To deploy Ceph in three different stages:

I. Set up Monitors and OSD nodes using the following command.

```
ansible-playbook -i hosts osd_mon_setup.yml
```

Following is the sample output.

```
2015-02-24 08:13:44,462 p=15079 u=hlinux | PLAY RECAP
*****
2015-02-24 08:13:44,463 p=15079 u=hlinux | admin-1 : ok=2 changed=1
unreachable=0 failed=0
2015-02-24 08:13:44,463 p=15079 u=hlinux | ceph-osd-1 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,463 p=15079 u=hlinux | ceph-osd-10 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,463 p=15079 u=hlinux | ceph-osd-11 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,464 p=15079 u=hlinux | ceph-osd-12 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,464 p=15079 u=hlinux | ceph-osd-13 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,464 p=15079 u=hlinux | ceph-osd-14 : ok=20 changed=17
unreachable=0 failed=0
2015-02-24 08:13:44,464 p=15079 u=hlinux | ceph-osd-15 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,465 p=15079 u=hlinux | ceph-osd-16 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,465 p=15079 u=hlinux | ceph-osd-17 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,465 p=15079 u=hlinux | ceph-osd-18 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,465 p=15079 u=hlinux | ceph-osd-19 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,465 p=15079 u=hlinux | ceph-osd-2 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,466 p=15079 u=hlinux | ceph-osd-20 : ok=20 changed=14
unreachable=0 failed=0
```



```

2015-02-24 08:13:44,466 p=15079 u=hlinux | ceph-osd-21 : ok=20 changed=17
unreachable=0 failed=0
2015-02-24 08:13:44,466 p=15079 u=hlinux | ceph-osd-3 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,466 p=15079 u=hlinux | ceph-osd-4 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,466 p=15079 u=hlinux | ceph-osd-5 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,467 p=15079 u=hlinux | ceph-osd-6 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,467 p=15079 u=hlinux | ceph-osd-7 : ok=20 changed=16
unreachable=0 failed=0
2015-02-24 08:13:44,467 p=15079 u=hlinux | ceph-osd-8 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,468 p=15079 u=hlinux | ceph-osd-9 : ok=20 changed=14
unreachable=0 failed=0
2015-02-24 08:13:44,468 p=15079 u=hlinux | mon-master-1 : ok=24 changed=20
unreachable=0 failed=0
2015-02-24 21:27:24,358 p=15079 u=hlinux | mon-1 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,358 p=15079 u=hlinux | mon-2 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 08:13:44,468 p=15079 u=hlinux | rados-1 : ok=2 changed=1
unreachable=0 failed=0
2015-02-24 08:13:44,468 p=15079 u=hlinux | rados-2 : ok=2 changed=1
unreachable=0 failed=0

```

II. Integrate Ceph on Helion nodes (Controller and Compute nodes) to support the Ceph backend] by entering:

```
ansible-playbook -i hosts client_setup.yml
```

The following is sample output.

```

....
2015-02-24 21:27:24,351 p=18817 u=hlinux | PLAY RECAP
*****
2015-02-24 21:27:24,352 p=18817 u=hlinux | admin-1 : ok=16 changed=14
unreachable=0 failed=0
2015-02-24 21:27:24,352 p=18817 u=hlinux | ceph-osd-1 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,352 p=18817 u=hlinux | ceph-osd-10 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,353 p=18817 u=hlinux | ceph-osd-11 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,353 p=18817 u=hlinux | ceph-osd-12 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,353 p=18817 u=hlinux | ceph-osd-13 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,353 p=18817 u=hlinux | ceph-osd-14 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,353 p=18817 u=hlinux | ceph-osd-15 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,354 p=18817 u=hlinux | ceph-osd-16 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,354 p=18817 u=hlinux | ceph-osd-17 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,354 p=18817 u=hlinux | ceph-osd-18 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,354 p=18817 u=hlinux | ceph-osd-19 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,355 p=18817 u=hlinux | ceph-osd-2 : ok=2 changed=0
unreachable=0 failed=0

```

```

2015-02-24 21:27:24,355 p=18817 u=hlinux | ceph-osd-20 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,355 p=18817 u=hlinux | ceph-osd-21 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,355 p=18817 u=hlinux | ceph-osd-3 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,355 p=18817 u=hlinux | ceph-osd-4 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,356 p=18817 u=hlinux | ceph-osd-5 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,356 p=18817 u=hlinux | ceph-osd-6 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,356 p=18817 u=hlinux | ceph-osd-7 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,356 p=18817 u=hlinux | ceph-osd-8 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,356 p=18817 u=hlinux | ceph-osd-9 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,357 p=18817 u=hlinux | compute0 : ok=27 changed=23
unreachable=0 failed=0
2015-02-24 21:27:24,357 p=18817 u=hlinux | compute1 : ok=27 changed=23
unreachable=0 failed=0
2015-02-24 21:27:24,357 p=18817 u=hlinux | controller0 : ok=13 changed=11
unreachable=0 failed=0
2015-02-24 21:27:24,357 p=18817 u=hlinux | controller1 : ok=13 changed=11
unreachable=0 failed=0
2015-02-24 21:27:24,357 p=18817 u=hlinux | controller2 : ok=13 changed=11
unreachable=0 failed=0
2015-02-24 21:27:24,358 p=18817 u=hlinux | mon-master-1 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,358 p=18817 u=hlinux | mon-1 : ok=2 changed=0
unreachable=0 failed=0
2015-02-24 21:27:24,358 p=18817 u=hlinux | mon-2 : ok=2 changed=0
unreachable=0 failed=0

```

III. Set up RADOS Gateway nodes in the Ceph cluster by entering:

```
ansible-playbook -i hosts rados_setup.yml
```

If any changes are made to `ceph.conf` file on any node manually, copy `ceph.conf` file as `ceph_master.conf` from that node to the Ceph admin node `/helion-ceph/ansible-playbooks/roles/ceph-mon-master/files/ceph_master.conf`.

Run the following command to sync the `ceph.conf` file with all the Ceph cluster and Helion client nodes.

```
ansible-playbook -i hosts sync_all_nodes.yml
```

Scaling Ceph cluster

Add monitor

- Provision the new node with the `ceph-cluster` image using the node-provisioning tool (as mentioned above) from the seed host. You need to have three monitor nodes. This example shows adding two additional monitor nodes.

Sample Baremetal.csv

```

f0:92:1c:05:47:21,helioncse1,m0ng00s3,192.168.51.104,150,163840,200,overcloud-
ceph-cluster,clmon4,002
f0:92:1c:05:35:58,helioncse1,m0ng00s3,192.168.51.106,163840,200,overcloud-
ceph-cluster,clmon5,002

```

Once the node is up, from the admin node modify `/helion-ceph/ansible-playbooks/cephcluster.csv` with the new mon node.

```
192.168.51.90,mon-master-1,mon-master,hlinux
192.168.51.100,mon2,mon,hlinux
192.168.51.102,mon3,mon,hlinux
192.168.51.104,mon4,mon,hlinux
192.168.51.106,mon5,mon,hlinux
```

- Run `bootstrap.sh` from `ansible-playbook` folder as mentioned above.
- Run Ansible playbook to add the new monitors

```
ansible-playbook -i hosts osd_mon_setup.yml
```

- Verify, if you are seeing all five monitors by entering:

```
sudo ceph -w
```

- For NTP-related warnings, check the following section on how to sync all the servers in the Ceph cluster with the NTP server.

Add OSD

To add an Object Storage Daemon (OSD):

- Provision the new
- node with the `ceph-cluster` image using the node-provisioning tool from the seed host as mentioned above.

Sample Baremetal.csv

The following is a sample of a `baremetal.csv` file:

```
f0:92:1c:05:56:98,helioncsel,m0ng00s3,10.1.67.123,178,163840,200,overcloud-
ceph-cluster,closd4,002
f0:92:1c:05:33:12,helioncsel,m0ng00s3,10.1.67.123,179,163840,200,overcloud-
ceph-cluster,closd5,002
```

This example shows that two OSD nodes are added and under each node there are four OSDs.

- Once the node is up, from the admin node modify `/helion-ceph/ansible-playbooks/cephcluster.csv` with the new monitor node.

```
192.168.51.90,mon-master-1,mon-master,hlinux
192.168.51.121,ceph-osd-22,osd,hlinux,xfs,/dev/sdc,xfs,/dev/sdb
192.168.51.121,ceph-osd-23,osd,hlinux,xfs,/dev/sdd,xfs,/dev/sdb
192.168.51.121,ceph-osd-24,osd,hlinux,xfs,/dev/sde,xfs,/dev/sdb
192.168.51.121,ceph-osd-25,osd,hlinux,xfs,/dev/sdf,xfs,/dev/sdb
192.168.51.125,ceph-osd-26,osd,hlinux,xfs,/dev/sdc,xfs,/dev/sdb
192.168.51.125,ceph-osd-27,osd,hlinux,xfs,/dev/sdd,xfs,/dev/sdb
192.168.51.125,ceph-osd-28,osd,hlinux,xfs,/dev/sde,xfs,/dev/sdb
192.168.51.125,ceph-osd-29,osd,hlinux,xfs,/dev/sdf,xfs,/dev/sdb
```

- If you are enabling journaling, in `/group_vars/ceph-cluster` on the admin node, run the following scripts in `helion-ceph/ansible-playbooks/`:

```
sudo ./createjournalpartitions <cephadmin keypair>
192.168.51.90,mon-master-1,mon-master,hlinux
192.168.51.121,ceph-osd-22,osd,hlinux,xfs,/dev/sdc,xfs,/dev/sdb5
192.168.51.121,ceph-osd-23,osd,hlinux,xfs,/dev/sdd,xfs,/dev/sdb6
192.168.51.121,ceph-osd-24,osd,hlinux,xfs,/dev/sde,xfs,/dev/sdb7
192.168.51.121,ceph-osd-25,osd,hlinux,xfs,/dev/sdf,xfs,/dev/sdb8
192.168.51.125,ceph-osd-26,osd,hlinux,xfs,/dev/sdc,xfs,/dev/sdb5
```

```
192.168.51.125,ceph-osd-27,osd,hlinux,xfs,/dev/sdd,xfs,/dev/sdb6
192.168.51.125,ceph-osd-28,osd,hlinux,xfs,/dev/sde,xfs,/dev/sdb7
192.168.51.125,ceph-osd-29,osd,hlinux,xfs,/dev/sdf,xfs,/dev/sdb8
```

- Run `bootstrap.sh` from `ansible-playbook` folder as mentioned above.

```
sudo ./bootstrap.sh <cephadmin keypair>
```

This script creates Ansible metadata from `cephcluster.csv` manifest file, and configures `/etc/hosts` file on ceph nodes.

- Run the Ansible playbook to add the new OSD nodes by entering:

```
ansible-playbook -i hosts osd_mon_setup.yml --tags "add-osd"
```

- Verify that you are seeing newly added OSD nodes and daemons from the `sudo ceph -w` command return.

Heartbeat Monitoring Tool (Optional)

The Heartbeat tool is written in Python and it is an open ended tool so other 3rd party service and status checks can be added or extended. This tool is run on the seed node from the HP Helion setup. This tool continuously queries for HP Helion OpenStack services and provides notifications to the pre-defined administrator when config file mismatches are detected, or when any of the specified services in config file are down. The Configuration service automatically reapplies the known good config files when the checksum does not match. It then restarts the associated services specified in the config file and triggers the validation script to do a minimal check for that service. Even if the config files are overwritten when the checksum does not match, a backup of the existing file from the current folder (named `[filename].timestamp`) is also saved in the backup folder. This allows the admin to verify the changes at any time.

The Heartbeat tar files are:

```
heartbeat-tool/Readme.md
heartbeat-tool/conf/heartbeat.conf
heartbeat-tool/conf/sample.conf heartbeat-tool/src/checks.py
heartbeat-tool/src/run.py
heartbeat-tool/src/sendemail.py

heartbeat-tool/src/osclients.py
heartbeat-tool/src/nodeaccess.py
```

These tarballs can be obtained from:

<https://helion.hpwsportal.com/>

[Return to Top](#)

Helion OpenStack Ceph cluster validation

To validate the Helion OpenStack Ceph cluster, perform the following:

1. Verify Ceph default pools by entering:

```
ceph osd lspools
```

For example:

```
0 data, 1 metadata, 2rbd
```

2. Verify that the monitor is running by entering:

```
ceph -s
```

For example:

```
root@ceph-mon1gw1:/var/lib/ceph/mon# ceph -s
cluster e0f2ad6b-588f-432c-99c1-d81f0f71cb77
health HEALTH_ERR 192 pgs stuck inactive; 192 pgs stuck unclean; no osds
monmap e1: 1 mons at {ceph-mon1gw1=192.168.116.54:6789/0}, election epoch
  2, quorum 0
ceph-mon1gw1
osdmap e1: 0 osds: 0 up, 0 in
pgmap v2: 192 pgs, 3 pools, 0 bytes data, 0 objects
  0 kB used, 0 kB / 0 kB avail
  192 creating
```

3. Ensure that the OSD Daemon is running by verifying the output from running `sudo ceph -w`.
4. Ensure that Ceph health and status is OK by entering:

```
ceph health
HEALTH_OK
ceph status
```

For example:

```
root@ceph-mon1:/home/ceph# ceph -s
cluster 6a710689-5b19-4ba3-b2c5-c23ddd26dce9
health HEALTH_OK
monmap e1: 1 mons at {ceph-mon1=192.168.116.54:6789/0}, election epoch 1,
  quorum 0 ceph-mon1
osdmap e336: 39 osds: 39 up, 39 in
pgmap v106607: 11456 pgs, 17 pools, 7878 MB data, 1319 kobjects
83315 MB used, 99199 GB / 99280 GB avail
11456 active+clean
```

5. Monitor and correct XFS fragmentation by entering:

```
xfs_db -c frag -r /dev/sdb1
```

which returns information such as:

```
Fragmentation on /dev/sdb1 - osd3
actual 22722, ideal 22557, fragmentation factor 0.73%
```

For example, to check the fragmentation on OSDs, enter:

```
#!/bin/sh
echo "Fragmentation on /dev/sdb1 - osd1"
xfs_db -c frag -r /dev/sdb1
echo "Fragmentation on /dev/sdc1 - osd1"
xfs_db -c frag -r /dev/sdc1
echo "Fragmentation on /dev/sdd1 - osd1"
xfs_db -c frag -r /dev/sdd1
echo "Fragmentation on /dev/sde1 - osd1"
xfs_db -c frag -r /dev/sde1
echo "Fragmentation on /dev/sdf1 - osd1"
xfs_db -c frag -r /dev/sdf1
echo "Fragmentation on /dev/sdg1 - osd1"
xfs_db -c frag -r /dev/sdg1
echo "Fragmentation on /dev/sdh1 - osd1"
xfs_db -c frag -r /dev/sdh1
echo "Fragmentation on /dev/sdi1 - osd1"
```

6. Depending on workloads, you can edit the Ceph tuning parameters:

```
osd op threads = 8
osd max backfills = 1
osd recovery max active = 1
filestore max sync interval = 100
filestore min sync interval = 50
filestore queue max ops = 2000
filestore queue max bytes = 536870912
filestore queue committing max ops = 2000
filestore queue committing max bytes = 536870912
```

[Return to Top](#)

Helion OpenStack Ceph client validation

This section explains how to validate Glance.

Glance

- Create a sample Glance Raw image on any controller node as shown below. Use the Raw data format with RBD for instant image snapshots and protection. For more details, refer to <http://ceph.com/docs/master/rbd/qemu-rbd/?highlight=raw>
- Use a conversion tool like `qemu-img` to convert from one image format to another.

For example:

```
qemu-img convert -f {source-format} -O {output-format} {source-filename}
{output-filename}
qemu-img convert -f qcow2 -O raw cirros-0.3.2-x86_64-disk.img
cirros-0.3.2-x86_64-disk.raw
glance image-create --name RImg --is-public=true --disk-format=raw --
container-format=bare --file cirros-0.3.2-x86_64-disk.raw
```

```

root@overcloud-controller0-mm73ivs47rh:~# glance image-create --name RImg --is-public=true --file cirros-0.3.2-x86_64-disk.raw
+-----+-----+
| Property          | Value                               |
+-----+-----+
| checksum          | cf2392db1f59d59ed69a8f8491b670e0  |
| container_format  | bare                               |
| created_at        | 2014-08-15T18:14:03                |
| deleted           | False                              |
| deleted_at        | None                               |
| disk_format       | raw                                |
| id                | c40e3ac7-d1d8-4f44-aef3-7c75a16d534f |
| is_public         | True                               |
| min_disk          | 0                                  |
| min_ram           | 0                                  |
| name              | RImg                               |
| owner             | 5ffdf19acff547d4ae1fbfc1eafbe6cf  |
| protected         | False                              |
| size              | 41126400                           |
| status            | active                             |
| updated_at        | 2014-08-15T18:14:09                |
| virtual_size      | None                               |
+-----+-----+
root@overcloud-controller0-mm73ivs47rh:~#

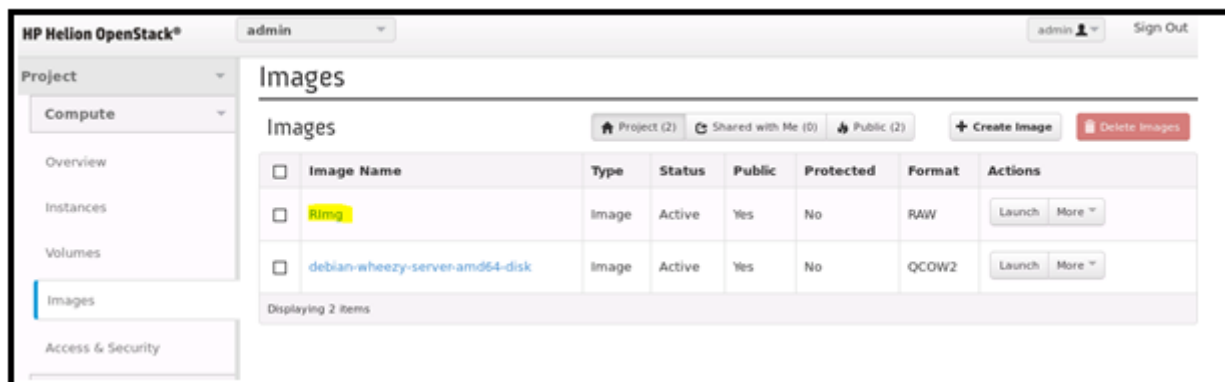
```

- Make sure that the uploaded Glance image is available in the Horizon UI and is correctly stored in the appropriate pool in Ceph by entering;

```

rbd ls -l <glance pool name>
glance image-list

```



- Enable logging in `glance-api.conf`. If you encounter a problem in any of the above steps, restart Glance services and re-run the problem step. If the problem persists, check Glance debug logs in the `/var/log/glance` directory and contact the HP support team for help.

Ceph Glance Clone Copy On Write

Note that Clone copy-on-write (COW) is achieved when an image is in RAW format. Use a conversion tool like `qemu-img` to convert from one format to another.

- Create a Glance image using the `glance image-create` command.

For example:

```
qemu-img convert -f {source-format} -O {output-format} {source-filename}
{output-filename}
qemu-img convert -f qcow2 -O raw cirros-0.3.2-x86_64-disk.img cirros-0.3.2-
x86_64-disk.raw
glance image-create --name RImg --is-public=true --disk-format=raw --
container-format=bare --file cirros-0.3.2-x86_64-disk.raw
```

- Create a Cinder volume on any controller node from a Glance image created above by entering:

```
cinder create -image-id <glance image id> --display-name RVol 2
```

```
root@overcloud-controller0-mmnr73ivs47rh:~# cinder create --image-id c40e3ac7-d1d8-4f44-aef3-7c75a16d534f
+-----+-----+
| Property | Value |
+-----+-----+
| attachments | [] |
| availability_zone | nova |
| bootable | false |
| created_at | 2014-08-15T18:36:25.690791 |
| display_description | None |
| display_name | RVol |
| encrypted | False |
| id | ed1bb437-126f-4842-87e7-4bafb5fe2398 |
| image_id | c40e3ac7-d1d8-4f44-aef3-7c75a16d534f |
| metadata | {} |
| size | 2 |
| snapshot_id | None |
| source_vol_id | None |
| status | creating |
| volume_type | None |
+-----+-----+
root@overcloud-controller0-mmnr73ivs47rh:~#
```

- Make sure the Cinder volume created is available in the rbd pool by entering:

```
rbd ls -l <cinder pool name>
cinder list
```

```
root@overcloud-controller0-mmnr73ivs47rh:~# rbd ls -l cinder
cNAME                               SIZE PARENT
volume-ed1bb437-126f-4842-87e7-4bafb5fe2398 2048M glance/c40e3ac7-d1d8-4f44-aef3-7c75a16d534f
root@overcloud-controller0-mmnr73ivs47rh:~# cinder list
+-----+-----+-----+-----+-----+-----+
| ID | Status | Display Name | Size | Volume Type |
+-----+-----+-----+-----+-----+-----+
| ed1bb437-126f-4842-87e7-4bafb5fe2398 | available | RVol | 2 | None |
| f48a0506-dbed-418a-a70c-9abcada237c5 | available | vol1 | 1 | None |
+-----+-----+-----+-----+-----+-----+
root@overcloud-controller0-mmnr73ivs47rh:~#
```


- Track clones to demonstrate copy-on-write feature by first listing snapshots of the Glance image, and then listing the children of the snapshot by entering:

```

rbd --pool <glance pool name> snap ls <glance image id>
rbd --pool <glance pool name> children --image <glance image id> --snap
<snap name>
rbd children <glance pool name>/<glance-image id>@<snap name>

```

```

root@overcloud-controller0-mm73ivs47rh:~# rbd --pool glance snap ls c40e3ac7-d1d8-4f4
SNAPID NAME      SIZE
      8 snap 40162 kB
root@overcloud-controller0-mm73ivs47rh:~# rbd --pool glance children --image c40e3ac7
cinder/volume-ed1bb437-126f-4842-87e7-4bafb5fe2398
root@overcloud-controller0-mm73ivs47rh:~# rbd children glance/c40e3ac7-d1d8-4f44-aef3
cinder/volume-ed1bb437-126f-4842-87e7-4bafb5fe2398
root@overcloud-controller0-mm73ivs47rh:~# █

```

- Enable logging in `glance-api.conf` and `cinder.conf` if you encounter a problem in any of the above steps. Restart Glance and Cinder services and re-run the problem step. If necessary, gather Glance debug logs in `/var/log/glance` directory and Cinder debug logs in `/var/log/upstart` directory and contact HP support team for help.

Creating a Cinder volume

There are two ways to create a volume:

- Horizon overcloud dashboard
- The Command Line Interface (CLI)

Using the Horizon Overcloud dashboard

To create a volume see [OpenStack User Guide](#)

Using the CLI

To create a volume using the command-line interface (CLI), from the Overcloud controller node running the Ceph client, run the following command:

```
Cinder create -display-name <name of the volume> <volume size>
```

For example:

```
# cinder create -- display-name vol2-RBD 1
```

Output:

```

+-----+
+-----+-----+
| Property           | Value |
+-----+-----+
| attachments        | []    |
| availability_zone   | nova  |

```

```

| bootable          | false
| created_at        | 2014-08-01T14:56:21.423821
| display_description | None
| display_name       | vol2-RBD
| encrypted          | False
| id                 | d6064822-d1c1-4e72-b496-ee807174ef96
| metadata           | {}
| size               | 1
| snapshot_id        | None
| source_vol_id      | None
| status             | creating
| volume_type        | None
+-----+
+-----+

```

Creating a Cinder backup

Once the Cinder service is restarted, you can create a backup of a Cinder volume. There are two ways to create a backup:

- The Horizon Overcloud Dashboard
- The Command Line Interface (CLI)



Note: For Cinder backup, the Cinder volume to be backed up must be in a detached state. The volume should not be attached to any of the instances or Virtual Machines.

Using the Horizon overcloud dashboard

Create a volume backup see [OpenStack User Guide](#)

Using the CLI

1. Login to the Overcloud using the `controllermanagement` command.
2. To create a cinder backup, enter:

```
cinder backup-create [--container <container>] --display-name <display-name>] [--display-description <display-description>] <volume>
```

where:

volume is then name or ID of the volume to backup.

container <container> is the optional Backup container name. (Default=None)

display-name <display-name> is the optional backup name. (Default=None)

display-description <display-description> is the optional backup description. (Default=None)

The following example shows how to create a backup with the name of **deb7rawbackup** for an existing Cinder volume with the ID **0a2c6c62-627f-42d3-9b66-e4ba56db0ba7**:

```
cinder backup-create --display-name deb7rawbackup 0a2c6c62-627f-42d3-9b66-
e4ba56db0ba7.
```

```
cinder backup-create --display-name cindervol_backup ff8d13a5-3083-424b-  
a626-0b75cbe8cf66
```

Output:

Property	Value
id	60764712-c456-465a-828b-5f45d3a14ff5
name	cindervol_backup
volume_id	ff8d13a5-3083-424b-a626-0b75cbe8cf66

- ### 3. View a list of Cinder backups, enter:

```
cinder backup-list
```

Output:

Container	Status	ID	Name	Size	Object Count	Volume ID
244aa3a1-b291-4cfe-9999-438f7611da2b	available		Rvol6backup	15		eb170c5e-d227-40ef-b515-b84b82c38eb0
32ea7668-9179-433c-8fe3-44b98cd9d85b	available		ubuntu1404-backup	10		8aefafcc-4171-4c11-b900-362fbda40015
60764712-c456-465a-828b-5f45d3a14ff5	creating		cindervol_backup	15		ff8d13a5-3083-424b-a626-0b75cbe8cf66
beeccb71-81e7-4860-8d38-add05a2e610d	available		Rvol6backup	15		eb170c5e-d227-40ef-b515-b84b82c38eb0

- 4. To view details of a selected volume, enter:**

```
cinder backup-show 60764712-c456-465a-828b-5f45d3a14ff5
```

Output:

Property	Value
----------	-------

availability_zone	None
container	None
created_at	2014-10-01T18:14:50.000000
description	None
fail_reason	None
id	60764712-c456-465a-828b-5f45d3a14ff5
name	cindervol_backup
object_count	None
size	15
status	creating
volume_id	ff8d13a5-3083-424b-a626-0b75cbe8cf66

5. Enter:

```
rbd ls -l helion-ceph-backups
```

Output:

NAME	SIZE	PARENT	FMT
PROT LOCK			
volume-0a2c6c62-627f-42d3-9b66-e4ba56db0ba7.backup.base	10240M		2
volume-0a2c6c62-627f-42d3-9b66-e4ba56db0ba7.backup.base@backup.02c6df2c-d03a-44ad-847a-ce03b580ee23.snap.1412106395.75	10240M	2	
volume-0a2c6c62-627f-42d3-9b66-e4ba56db0ba7.backup.base@backup.c9c20a09-403e-4011-a3f8-2fea11a560ee.snap.1412042130.16	10240M	2	
volume-3adf1c83-2efa-4a1e-bef6-cdaffd13b489.backup.base	3072M		2
volume-3adf1c83-2efa-4a1e-bef6-cdaffd13b489.backup.base@backup.cdd27130-1791-45f6-8b6e-cc284922b02e.snap.1412041965.31	3072M	2	

6. To view cluster utilization, enter:

```
rados df
```

Output:

pool name	category	rd	KB	objects	clones
degraded	unfound		rd KB	wr	wr
KB					
.rgw	-		9411	51459	
0	0	110460	85700	130906	
51952					
.rgw.buckets	-		20004	188711	
0	0	6466145	4329147	8000069	
1563078					
helion-ceph-backups	-		54343272	13300	
2190	0	16295	40106569	38157	
70071922					



Note: You can delete a Cinder volume by executing: `cinder backup-delete`.

7. Attach the Cinder volume to a Nova instance by entering:

```
nova volume-attach <instance id> <volume ID>
```

8. Select the instance and provide the device name by entering:

```
nova volume-attach <server> <volume> <device>
```

Mounting the volume and Copying a new image

To mount the volume and copy the new image file from the VM, perform these steps:

1. Log in as root.
2. List the block devices by entering:

```
lsblk
```

Output:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
vda	254:0	0	10G	0	disk	
aavda1	254:1	0	243M	0	part	/boot
aavda2	254:2	0	1K	0	part	
aavda5	254:5	0	7.8G	0	part	
aadebian-root (dm-0)	253:0	0	7.4G	0	lvm	/
aadebian-swap_1 (dm-1)	253:1	0	376M	0	lvm	[SWAP]
vdb	254:16	0	26G	0	disk	
vdc	254:32	0	10G	0	disk	
vdd	254:48	0	10G	0	disk	
vde	254:64	0	10G	0	disk	
vdf	254:80	0	10G	0	disk	
vdg	254:96	0	10G	0	disk	
vdi	254:128	0	15G	0	disk	
vdj	254:144	0	15G	0	disk	

3. Mount dev to vol by entering:

```
mount /dev/vdj /mnt/vol
```

Output:

```
mount: mount point /mnt/vol does not exist
```

4. Mount dev to vol1 by entering:

```
mount /dev/vdj /mnt/vol1
```

5. Change the directory by entering:

```
cd /mnt/vol1
```

6. List all the sub-directories by entering:

```
ls -ltr
```

7. To view disk usage, enter:

```
df -h
```

Output:

Filesystem	Size	Used	Avail	Use%	Mounted on
rootfs	7.3G	6.7G	248M	97%	/
udev	10M	0	10M	0%	/dev
tmpfs	397M	208K	397M	1%	/run

```

/dev/mapper/debian-root 7.3G 6.7G 248M 97% /
tmpfs                  5.0M 0 5.0M 0% /run/lock
tmpfs                  794M 0 794M 0% /run/shm
/dev/vda1              228M 18M 199M 9% /boot
/dev/vdj               15G 847M 14G 6% /mnt/vol1

```

8. To mount, enter:

```
mount
```

Output:

```

sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs
  (rw,relatime,size=10240k,nr_inodes=506385,mode=755)
devpts on /dev/pts type devpts
  (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=406364k,mode=755)
/dev/mapper/debian-root on / type ext4 (rw,relatime,errors=remount-
ro,user_xattr,barrier=1,data=ordered)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /run/shm type tmpfs
  (rw,nosuid,nodev,noexec,relatime,size=812720k)
/dev/vda1 on /boot type ext2 (rw,relatime,errors=continue)
/dev/vdj on /mnt/vol1 type ext4
  (rw,relatime,user_xattr,barrier=1,data=ordered)

```

Now the Cinder volume has additional file system changes within the volume.

9. To list the additional file system changes, enter:

```
ls-ltr
```

Output:

```

total 6328752
drwx----- 2 root root      16384 Oct  1 05:22 lost+found
-rwxr-x--- 1 root root 702939136 Oct  1 05:23 CentOS_65.qcow2
-rw-r--r-- 1 root root 10870593 Oct  1 05:24 initrd.img-3.2.0-4-amd64
-rw-r--r-- 1 root root 5766807552 Oct  2 00:34 Debian_7.raw

```

10. To view disk usage, enter:

```
df -h
```

Output:

```

Filesystem      Size  Used Avail Use% Mounted on
rootfs          7.3G  6.7G  248M  97% /
udev            10M    0   10M   0% /dev
tmpfs           397M 208K  397M   1% /run
/dev/mapper/debian-root 7.3G  6.7G  248M  97% /
tmpfs           5.0M    0   5.0M   0% /run/lock
tmpfs           794M    0   794M   0% /run/shm
/dev/vda1       228M  18M  199M   9% /boot
/dev/vdj        15G   6.2G   8.6G  42% /mnt/vol1

```

11. To unmount the volume, enter:

```
umount /dev/vdj
```

12. To mount the volume, enter:

```
mount
```

Output:

```
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs
  (rw,relatime,size=10240k,nr_inodes=506385,mode=755)
devpts on /dev/pts type devpts
  (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=406364k,mode=755)
/dev/mapper/debian-root on / type ext4 (rw,relatime,errors=remount-
ro,user_xattr,barrier=1,data=ordered)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /run/shm type tmpfs
  (rw,nosuid,nodev,noexec,relatime,size=812720k)
/dev/vda1 on /boot type ext2 (rw,relatime,errors=continue)
```

Restoring data from a Cinder backup

You can restore the backed up volume to a new volume or an existing volume.

In the following example, a new volume is created and the data is restored to it.

To create a new volume and to restore data backup perform the following steps.

1. To create a volume, enter:

```
cinder create --display-name restore_volume1 15
```

Output:

Property	Value
attachments	[]
availability_zone	nova
bootable	false
created_at	2014-10-01T21:48:42.727440
display_description	None
display_name	restore_volume1
encrypted	False
id	1b4614f8-8069-4211-8a3e-797be5641964
metadata	{}
size	15
snapshot_id	None
source_volid	None
status	creating
volume_type	None

2. Execute the following command to the Cinder backup restore:

```
cinder backup-restore --volume-id restore_volume1 60764712-
c456-465a-828b-5f45d3a14ff5
```

3. View the Cinder backup by entering:

```
cinder backup-list
```

Output:

```

+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|      |      ID      |      |      Volume ID      | | | |
|      | Status |      Name      | Size | Object Count |      Container      |
|      |      |      |      |      |      |      |
+-----+-----+-----+-----+
| 02c6df2c-d03a-44ad-847a-ce03b580ee23 | 0a2c6c62-627f-42d3-9b66-
e4ba56db0ba7 | available | deb7rawbackup | 10 |      None      |
helion-ceph-backups |
| 244aa3a1-b291-4cfe-9999-438f7611da2b | eb170c5e-d227-40ef-b515-
b84b82c38eb0 | available | Rvol6backup | 15 |      None      |
helion-ceph-backups |
| 32ea7668-9179-433c-8fe3-44b98cd9d85b | 8aefafcc-4171-4c11-
b900-362fbd40015 | available | ubuntu1404-backup | 10 |      None      |
helion-ceph-backups |
| 60764712-c456-465a-828b-5f45d3a14ff5 | ff8d13a5-3083-424b-
a626-0b75cbe8cf66 | restoring | cindervol_backup | 15 |      None      |
helion-ceph-backups |
| beecb71-81e7-4860-8d38-add05a2e610d | eb170c5e-d227-40ef-b515-
b84b82c38eb0 | available | Rvol6backup | 15 |      None      |
helion-ceph-backups |
| c9c20a09-403e-4011-a3f8-2fea11a560ee | 0a2c6c62-627f-42d3-9b66-
e4ba56db0ba7 | available | RDebian7_backup | 10 |      None      |
helion-ceph-backups |
| cdd27130-1791-45f6-8b6e-cc284922b02e | 3adflc83-2efa-4ale-bef6-
cdaffd13b489 | available | Rbackup1 | 3 |      None      |
helion-ceph-backups |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

4. View the Cinder list by entering:

```
cinder list
```

Output:

```

+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
|      ID      |      Status      |      Display      | | |
| Name      | Size | Volume Type | Bootable |      Attached to      |
|      |      |      |      |      |
+-----+-----+-----+-----+-----+
| 0219d66e-d69d-4e28-bf4f-cb5f096696e3 | available |      |
Rsmallvol4 | 1 |      None      | false |
| 0285ee63-4ebd-4cd1-915c-933c48503d00 | in-use |      |
Rvol1 | 10 |      None      | false | 54938de0-49dd-4b01-931e-
dafcddc41518 |
| 054bfa98-1d69-4cb8-b195-9b9481f5b8c7 | in-use |      |
Rwin2012Cowrawvol1 | 26 |      None      | true | 0bf98387-b0c9-4814-
a2ef-f81clef1322e |

```



```
| 1b4614f8-8069-4211-8a3e-797be5641964 | restoring-backup |
restore_volumes | 15 | None | false |
|
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Once the backup is created, the volume name remains the same as the Cinder-backup name.

- 5. Verify the volume name by entering:**

cinder list

Output:

```

+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          ID          | Status |          Display          |
| Name                | Size  | Volume Type | Bootable | Attached to |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| 0219d66e-d69d-4e28-bf4f-cb5f096696e3 | available |          |          |          |
| Rsmallvol4                          | 1         | None     | false   |          |
+-----+-----+-----+-----+
| 0285ee63-4ebd-4cd1-915c-933c48503d00 | in-use    |          |          |          |
|                                | 10        | None     | false   |          |
| 54938de0-49dd-4b01-931e-dafcd41518 |          |          |          |          |
| 1b4614f8-8069-4211-8a3e-797be5641964 | available |          |          |          |
| cindervol_forbackup                | 15        | None     | false   |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

Verify the attachment of volume to a VM

Perform the following to verify the attachment of a volume to a VM and verify the content.

1. Change the directory by entering:

```
cd /mnt/vol1
```

2. Verify the content by entering:

$$1s - 1s$$

Output

```
total 697100
686468 -rwxr-x--- 1 root root 702939136 Oct 1 05:23 CentOS_65.qcow2
10616 -rw-r--r-- 1 root root 10870593 Oct 1 05:24 initrd.img-3.2.0-4-amd64
16 drwx----- 2 root root 16384 Oct 1 05:22 lost+found
```

[Return to Top](#)

Volume snapshots

Volume snapshots are saved in a Cinder pool.

Creating a volume snapshot for backup

You can create new and identical volumes by taking a snapshot of the volume.

There are two ways to create a snapshot backup:

1. Horizon Overcloud dashboard
2. The Command Line Interface (CLI)

Notes for taking snapshots

- The volume must be detached and must be in an **available** state to take a snapshot of it. An error occurs if you try to snapshot a used volume.
- To function properly, keep the original volume, whose snapshot was taken. If the original volume is deleted then the snapshot becomes unusable.

Using the Horizon Overcloud dashboard

To create a snapshot from volume, see [OpenStack User Guide](#)

Using the CLI

The following steps show how to create a snapshot using the CLI.

1. To create a snapshot, enter:

```
nova volume-snapshot-create --force [TRUE or FALSE] --display_name
[DISPLAY_NAME] --display_description [DISPLAY_DESCRIPTION] [VOLUME_ID]
```



Note: This is a base command without the variable set.

2. To view the snapshot, enter:

```
nova volume-snapshot-list
```

Output

```
+-----+
+-----+-----+
+-----+-----+
| ID | Volume ID | Status | Display Name | Size |
+-----+-----+
+-----+-----+
| 1f9cae44-e3c9-4326-a1f9-68aeea34d672 |
0285ee63-4ebd-4cd1-915c-933c48503d00 | available | snapshot for
Rinstgeneral_snapshot | 10 |
| 2758b62c-8f2a-482c-bf2c-9183c8304227 | f628002a-6cc4-4e70-a98f-
d575e36fca75 | available | snapshot for Rinstgeneral_snapshot | 10 |
| 32267733-bf04-4180-a1ea-bc133726bb7b | 525bd6a2-05cc-4bba-9a6a-
f8e8a3f6ce68 | available | snapshot for Rinstgeneral_snapshot | 10 |
| 4b1e37f6-04f6-41c4-a80c-34ce8d8c743a | 386ed069-71c4-428b-9ecc-
f21d572d74b2 | available | snapshot for Rdeb8cowinst7_snapshot | 10 |
+-----+-----+
+-----+-----+
+-----+-----+
```

Working with Nova

There are two ways to list the Nova instance:

- The Horizon Overcloud Dashboard

- The Command Line Interface (CLI)



Note: For Cinder backups, the Cinder volume to be backed up must be in a detached state. The volume should not be attached to any of the instances or Virtual Machines.

Using the Horizon Overcloud dashboard

1. Log into the Overcloud Horizon.
2. From the left panel, click the **Projects** tab and then **Instances** to view the list of instances.
3. Click on the **Instance Name** to view the instance console log.

Using the Command Line Interface

From the CLI, perform the following:

1. List all of the Nova instances

```
nova list
```

```
root@overcloud-controller0-4rooy/pzo4dz:~# nova list
+-----+-----+-----+-----+
| ID                | Name                | Status | Task |
+-----+-----+-----+-----+
| 249f0727-d4a8-4f11-b6c7-0cce447a5116 | instance1_RBD | ACTIVE | -    |
0.5, 192.0.8.24 |
| b0867155-e48d-4023-b84a-6a676310d58a | instance2_RBD | ACTIVE | -    |
0.6, 192.0.8.23 |
| 34dff699-2156-4a83-b7b4-8d678bd8e602 | instance3_RBD | ACTIVE | -    |
0.7                |
| 839019a9-c7aa-44b0-a226-46a1e97bb371 | instance4_RBD | ACTIVE | -    |
0.9                |
+-----+-----+-----+-----+
```

2. Verify instance status by entering:

```
sudo ceph -w
```

```

root@overcloud-controller0-4rooy7pzo4dz:~# ceph -w
cluster f6d33f33-7278-4387-afe1-f5f8983cb7c8
health HEALTH_OK
monmap e1: 1 mons at {ceph-mon=192.168.122.110:6789/0}, election e
osdmap e60: 3 osds: 3 up, 3 in
pgmap v11494: 620 pgs, 7 pools, 1067 MB data, 282 objects
                22641 MB used, 105 GB / 134 GB avail
                620 active+clean

2014-08-14 21:17:22.348164 mon.0 [INF] pgmap v11494: 620 pgs: 620 activ
05 GB / 134 GB avail; 2617 B/s rd, 7852 B/s wr, 4 op/s

```

Attaching the Cinder volume to the Nova instance

There are two ways to attach a Cinder volume to a Nova instance.

- The Horizon overcloud dashboard
- The Command Line Interface (CLI)

Attaching a Cinder volume from Horizon Overcloud dashboard

To attach a Cinder volume to a Nova instance perform the following steps:

1. In the Horizon Dashboard, click the **Project** Tab.
2. Click **Compute** and then **Volume** to open the Volume page.
3. Click the **More Action** tab, and select **Edit Attachments**.
4. Click the **Attach to Instance** drop-down list and select the instance.
5. In the **Device Name** box, enter the name of the selected instance.
6. Click **Attach Volume** to attach the Cinder volume to the Nova instance. To undo these changes, click **Cancel**.

Attaching a Cinder volume using the Command Line Interface (CLI)

To attach a Cinder volume to a Nova instance using the CLI:

1. Execute the following command to attach the volume to a Nova instance

```
nova volume-attach <instance id> <volume ID>
```

2. To view all the volumes, enter:

```
# cinder list
```

3. To view the Nova instance, enter:

```
nova list
```

4. To view the details of the attached volume, enter:

```
# cinder show <volume ID>
```

For example:

```
cinder show 580d3e95-970f-4a9c-92ea-284799dcbc82
```

Output:

```

+-----+
+-----+
+
| Property                                | Value
|
+-----+
+-----+
+
| attachments                            | [{u'device': u'/dev/
vde', u'server_id': u'd6c98de0-b65e-4e43-bd5e-04c81ad26cd1',
u'id': u'580d3e95-970f-4a9c-92ea-284799dcbc82',
u'host_name': None, u'volume_id':
u'580d3e95-970f-4a9c-92ea-284799dcbc82'}]}
|
| availability_zone                      | nova
|
| bootable                              | false
|
| created_at                            | 2014-08-13T03:38:27.000000
|
| display_description                    | None
|
| display_name                          | volume2_RBD
|
| encryptd                              | False
|
| id                                    |
580d3e95-970f-4a9c-92ea-284799dcbc82
|
| metadata                              | {u'readonly': u'False',
u'attached_mode': u'rw'}
|
| os-vol-host-attr:host                  | overcloud-controller1-
thg43e77ptei
|
| os-vol-mig-status-attr:migstat         | None
|
| os-vol-mig-status-attr:name_id         | None
|
| os-vol-tenant-attr:tenant_id          | 98ae295c1958428a890cf6441d70db08
|
| size                                  | 2
|
| snapshot_id                           | None
|
| source_volid                           | None
|

```

```
| status | in-use
|
| volume_type | None
|
+-----+
+-----+
+
```

5. To view the details of the Nova instance, enter:

```
nova show < nova instance ID>
```

For example:

```
# nova show d6c98de0-b65e-4e43-bd5e-04c81ad26cd1
```

Output:

```
+-----+
+-----+
+
| Property | Value
+-----+
+-----+
+
| OS-EXT-AZ:availability_zone | nova
|
| OS-EXT-SRV-ATTR:host | overcloud-novacompute0-
k3kakatgtgb2
|
| OS-EXT-SRV-ATTR:hypervisor_hostname | overcloud-novacompute0-
k3kakatgtgb2.novalocal
|
| OS-EXT-SRV-ATTR:instance_name | instance-00000087
|
| OS-EXT-STS:power_state | 1
|
| OS-EXT-STS:task_state | -
|
| OS-EXT-STS:vm_state | active
|
| accessIPv4 |
|
| accessIPv6 |
|
| config_drive |
|
| created | 2014-08-12T23:43:50Z
|
| default-net network | 10.0.0.43, 192.168.100.108
|
| flavor | m1.tiny (1)
|
```

```

| hostId                                     |
| cf6bb4eb58517b0e06246628e3d0559267a2594c06ea44100e2fae1e |
| |                                         |
| id                                         | d6c98de0-b65e-4e43-
bd5e-04c81ad26cd1                         |
| image                                     | debian-wheezy-server-amd64-disk
(39565ba5-bfe7-4ee7-be2b-abab70eeb989) |
| key_name                                  | default
| |                                         |
| metadata                                 | {}
| |                                         |
| name                                     | vm1
| |                                         |
| progress                                 | 0
| |                                         |
| security_groups                          | default
| |                                         |
| status                                   | ACTIVE
| |                                         |
| tenant_id                               | 98ae295c1958428a890cf6441d70db08
| |                                         |
| updated                                  | 2014-08-12T23:44:23Z
| |                                         |
| user_id                                  | 835261faa1454b56bfab6cd07edfd433
| |                                         |
+-----+
+-----+
+

```

[Return to Top](#)

Ceph RADOS gateway validation

To validate the RADOS gateway, make a GET request to the gateway server using the FQDN or IP address of the gateway server.

For example:

```
curl -k (https://gateway.ex.com)
```

• GET Response

```

<?xml version="1.0" encoding="UTF-8"?><ListAllMyBucketsResult
xmlns="http://s3.amazonaws.com/doc/2006-03-01/"><Owner><ID>anonymous</
ID><DisplayName></DisplayName></Owner><Buckets></Buckets></
ListAllMyBucketsResult>

```

This response indicates that gateway instance is working as expected.

- If there is an error, ensure `radosgw` is executed in debug mode and watch out for errors.
- If there is a permission issue on `/var/run/ceph/ceph-client.radosgw.gateway.asok`, change file permission accordingly.
- If there is error with Apache2 or FastCGI, look for debug logs in the `/var/log/apache2/error.log`. Changing permissions accordingly on `/var/www` directory or `/var/www/s3gw.fcgi` file should fix the problem.

Creating Users

To create users, execute the following:

```
radosgw-admin user create --subuser=s3User:swiftUser --display-name="First User" --key-type=swift --access=full
```


- Make sure that user `s3User` and subuser `s3User:swiftUser` are stored in respective `.users.uid` and `.users.swift` pool.

```
gateway@gateway:~$ rados -p .users.uid ls
s3User
gateway@gateway:~$ rados -p .users.swift ls
s3User:swiftUser
gateway@gateway:~$
```

- S3 users and swifts users need to have access and secret keys to enable end users to interact with the gateway instance. To create the access and secret key for `s3User`, enter:

```
radosgw-admin key create --uid=s3User --key-type=s3 --gen-access-key --gen-secret
```

```
gateway@gateway:~$ radosgw-admin key create --uid=s3User --key-type=s3 --gen-access-key --gen-secret
{ "user_id": "s3User",
  "display_name": "First User",
  "email": "",
  "suspended": 0,
  "max_buckets": 1000,
  "auid": 0,
  "subusers": [],
  "keys": [
    { "user": "s3User",
      "access_key": "2KASLCG8C1XVYERMD2WQ",
      "secret_key": "UpUVhhhdAemX779ZAECVQBNBxEtCaqASzjDJW8ZB"}],
  "swift_keys": [
    { "user": "s3User:swiftUser",
      "secret_key": "Pp3YqVoyqOpFF28kby03e55j3akd0wEE3NYGjXsK"}],
  "caps": [],
  "op_mask": "read, write, delete",
  "default_placement": "",
  "placement_tags": [],
  "bucket_quota": { "enabled": false,
    "max_size_kb": -1,
    "max_objects": -1},
  "user_quota": { "enabled": false,
    "max_size_kb": -1,
    "max_objects": -1},
  "temp_url_keys": []}
```


- Make sure that keys generated are free of JSON escape () characters,
- If the User or Application will write more than 1k Containers, then you must modify the `max_buckets` variable. Also, right-sizing of Placement Groups per Pool may be required. Make sure that `max_buckets` is set to unlimited size by setting it to 0. This is important in order to write unlimited containers into the `.rgw.buckets` default pool during workload testing.

```
radosgw-admin user modify --uid=s3User --max-buckets=0
```

Working with the Swift client

- The gateway instance and Swift users can be verified on gateway node or Ceph client using Swift Client by making Swift v1.0 requests
- Create `creds.py` with the following file contents

```
#Auth url pointing to gateway node
export ST_AUTH=http://gateway.ex.com/auth/v1.0
#Swift user
export ST_USER=s3User:swiftUser
#Swift user - secret key
export ST_KEY= abd
```

- Source swift credentials by entering:

```
source creds.py
```

- List the container by entering:

```
swift --insecure -V 1.0 -A http://gateway.ex.com/auth/v1.0 -U
s3User:swiftUser -K abc list
```

OR

```
swift list
```

```
root@overcloud-controller0-mmnr73ivs47rh:~# swift list
my-Bucket
```

- Display container information by entering:

```
swift stat <container>
```

```
root@overcloud-controller0-mmnr73ivs47rh:~# swift stat my-Bucket
Account: v1
Container: my-Bucket
Objects: 0
Bytes: 0
Read ACL:
Write ACL:
Sync To:
Sync Key:
Server: Apache/2.4.7 (Ubuntu)
X-Container-Bytes-Used-Actual: 0
Content-Type: text/plain; charset=utf-8
```

- Upload image into the container by entering:

```
swift upload <container> <image to upload>
```

```
root@overcloud-controller0-mm73ivs47rh:~# swift upload my-Bucket cirros-0.3.2-x86_64-
cirros-0.3.2-x86_64-disk.raw
root@overcloud-controller0-mm73ivs47rh:~# swift list my-Bucket
cirros-0.3.2-x86_64-disk.raw
```

- Verify the upload using stat by entering:

```
swift stat <container>
```

```
root@overcloud-controller0-mm73ivs47rh:~# swift stat my-Bucket
Account: v1
Container: my-Bucket
Objects: 1
Bytes: 41126400
Read ACL:
Write ACL:
Sync To:
Sync Key:
Server: Apache/2.4.7 (Ubuntu)
X-Container-Bytes-Used-Actual: 41127936
Content-Type: text/plain; charset=utf-8
root@overcloud-controller0-mm73ivs47rh:~#
```

- Verify that the uploaded image is residing in the RGW pool by entering:

```
rados -p .rgw.buckets ls
```

```
gateway@gateway:~$ rados -p .rgw.buckets ls
default.39808.1__shadow_.KC4qQI4QM_6J4-JN-TrCpruZom3vOyE_6
default.39808.1__shadow_.KC4qQI4QM_6J4-JN-TrCpruZom3vOyE_1
default.39808.1__shadow_.KC4qQI4QM_6J4-JN-TrCpruZom3vOyE_9
default.40982.1__shadow_.-HH5IMmIVqFjCg1xHkEmrq2y8aVX-25_7
default.40982.1__shadow_.-HH5IMmIVqFjCg1xHkEmrq2y8aVX-25_1
default.39808.1__shadow_.KC4qQI4QM_6J4-JN-TrCpruZom3vOyE_7
default.40982.1_cirros-0.3.2-x86_64-disk.raw
```

Next Steps

[Ceph Monitoring](#)

[Return to Top](#)