Installing Ceph

Contents

Installing Ceph

This page describes configuration, installation, and the integration of Ceph Block Storage with HP Helion OpenStack 2.0. After the installation of Ceph you can perform the cinder operation as documented in this page.

Attention: Beta1 Workaround

You must perform the following step after Step **3** in the section ***Configure Ceph as a Cinder backend***.

```
cp /usr/lib/python2.7/dist-packages/rbd.py /opt/stack/venv/cinder-
<time>/lib/python2.7/site-packages/
cp /usr/lib/python2.7/dist-packages/rados.py /opt/stack/venv/cinder-
<time>/lib/python2.7/site-packages/
```

The functionality to attach RBD volume to a Nova instance is not working.

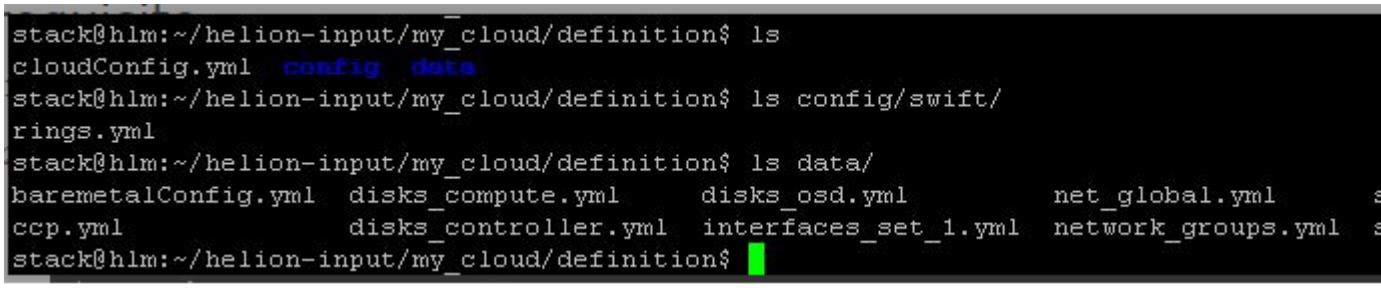
Prerequisite

The deployer node must be setup before deploying Ceph. For more details on the installation of deployer node, refer to *installation guide*.

Installation Procedure

Perform the following procedures to configure, install, and the integrate the Ceph Block Storage with HP Helion OpenStack 2.0

1. Login to the Deployer node.
2. Create the Ceph configuration files (**manually**) at `~/helion-input/my_cloud/definition`. The example configuration files are available at *Ceph Configuration files*.
3. Copy the configuration files as shown in the screeshot below.



You must copy the **yml** files in the appropriate folder. For example: Copy the ring.yml file in `config/swift/`.

4. Edit the configuration files, based on your environment, to deploy Ceph.
5. Edit the file `disks_osd.yml` and enter the details for the additional disks meant for OSD data and journal filesystems.

```
vi disks_osd.yml
```

6. Commit your configuration to a *local repository*:

```
cd ~/helion/hos/ansible
git add -A
git commit -m "<commit message>"
```

Note: Enter your commit message <commit message>

7. Run the configuration processor

```
cd ~/helion/hos/ansible
ansible-playbook -i hosts/localhost config-processor-run.yml
```

8. Change the directory to `~/scratch/ansible/next/hos/ansible` and run the following command to create a deployment directory.

```
ansible-playbook -i hosts/localhost ready-deployment.yml
```

9. Modify `~/helion/hlm/ansible/hlm-deploy.yml` and uncomment the line containing `ceph-deploy.yml` to enable deployment of ceph.
10. Run the following ansible playbook:

```
ansible-playbook -i hosts/verb_hosts site.yml
```

Ceph Monitor service is deployed on the Controller Nodes and OSD's are deployed as separate nodes (Resource Nodes).

Run Ceph Client Packages

Execute the following command to install the Ceph client packages on controller nodes.

```
cd ~/scratch/ansible/next/hos/ansible
ansible-playbook -i hosts/verb_hosts ceph-client-prepare.yml
```

This will also create ceph users and Ceph pools on the resource nodes:

Configure Ceph as a Cinder backend

Perform the following procedure on the Deployer node to configure Ceph as a Cinder backend:

1. Edit `~/helion/hos/ansible/roles/_CND-CMN/templates/cinder.conf.j2` to add ceph configuration data as shown below:

```
enabled_backends=ceph1
```

2. Copy the following configurations:

```
[ceph1]
rbd_max_clone_depth = 5
rbd_flatten_volume_from_snapshot = False
rbd_uuid = 457eb676-33da-42ec-9a8c-9293d545c337
rbd_user = cinder
rbd_pool = volumes
rbd_ceph_conf = /etc/ceph/ceph.conf
volume_driver = cinder.volume.drivers.rbd.RBDDriver
volume_backend_name = ceph
```

Note: The **rbd_uuid** is available in `/home/stack/helion/hos/ansible/roles/ceph-client-prepare/vars/ceph_user_model.yml`

3. To enable cinder backup to Ceph, modify `cinder.conf.j2` at `~/helion/hos/ansible/roles/_CND-CMN/templates/cinder.conf.j2` with the following values:

```
backup_driver = cinder.backup.drivers.ceph
backup_ceph_conf = /etc/ceph/ceph.conf
backup_ceph_user = cinder-backup
backup_ceph_chunk_size = 134217728
backup_ceph_pool = backups
backup_ceph_stripe_unit = 0
backup_ceph_stripe_count = 0
restore_discard_excess_bytes = true
```

4. Copy `ceph.client.cinder.keyring` to the controller nodes:
   a. Login to controller node as a root user and execute the following command.

   ```
   ceph auth get-or-create client.cinder | tee /etc/ceph/
   ceph.client.cinder.keyring
   ```

   OR

   You can copy the keyring from the deployer node

   ```
   scp /etc/ceph/ceph.client.cinder.keyring
   ```

   to `/etc/ceph` folder on all the controller nodes.
5. Commit your configuration to the local repository to configure cinder on the deployer node

   ```
   cd /home/stack/helion/hos/ansible
   git add -A
   git commit -m "<your commit message>"
   ```

   Note: Enter your commit message <commit message>
6. Change the directory to `~/scratch/ansible/next/hos/ansible` and run the following command to create a deployment directory.

   ```
   ansible-playbook -i hosts/localhost ready-deployment.yml
   ```
7. Run the following ansible playbook:

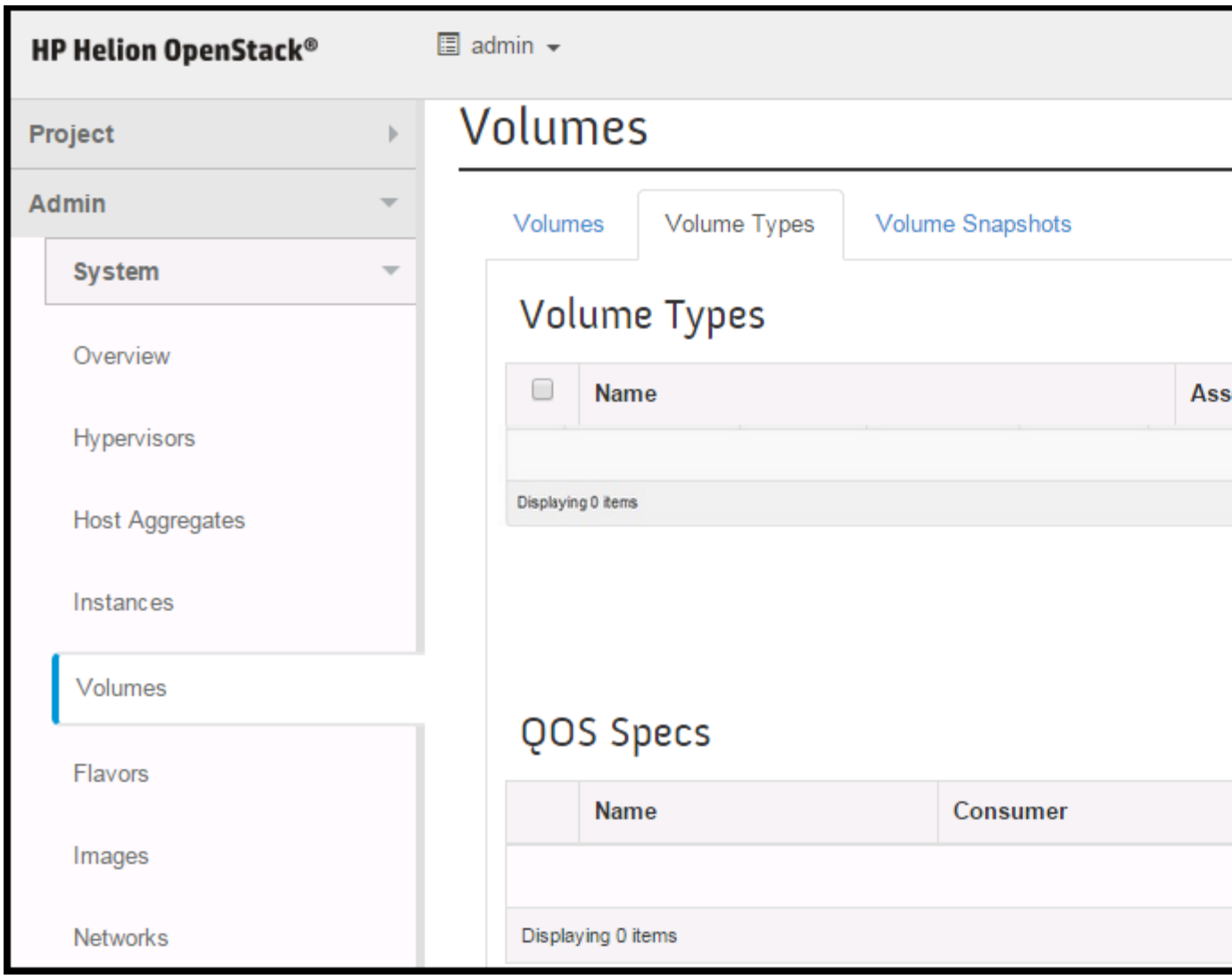
   ```
   ansible-playbook -i hosts/verb_hosts cinder-reconfigure.yml
   ```

   Once cinder is configured, launch the Horizon dashboard to create a cinder volume type.

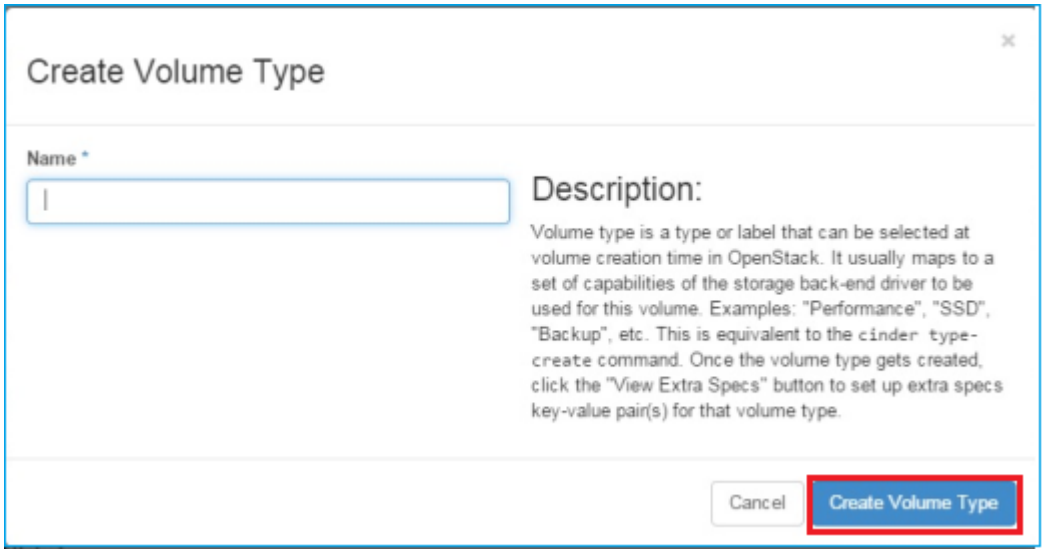
Creating Cinder Volume Type

To create a volume type using the Horizon dashboard, do the following:

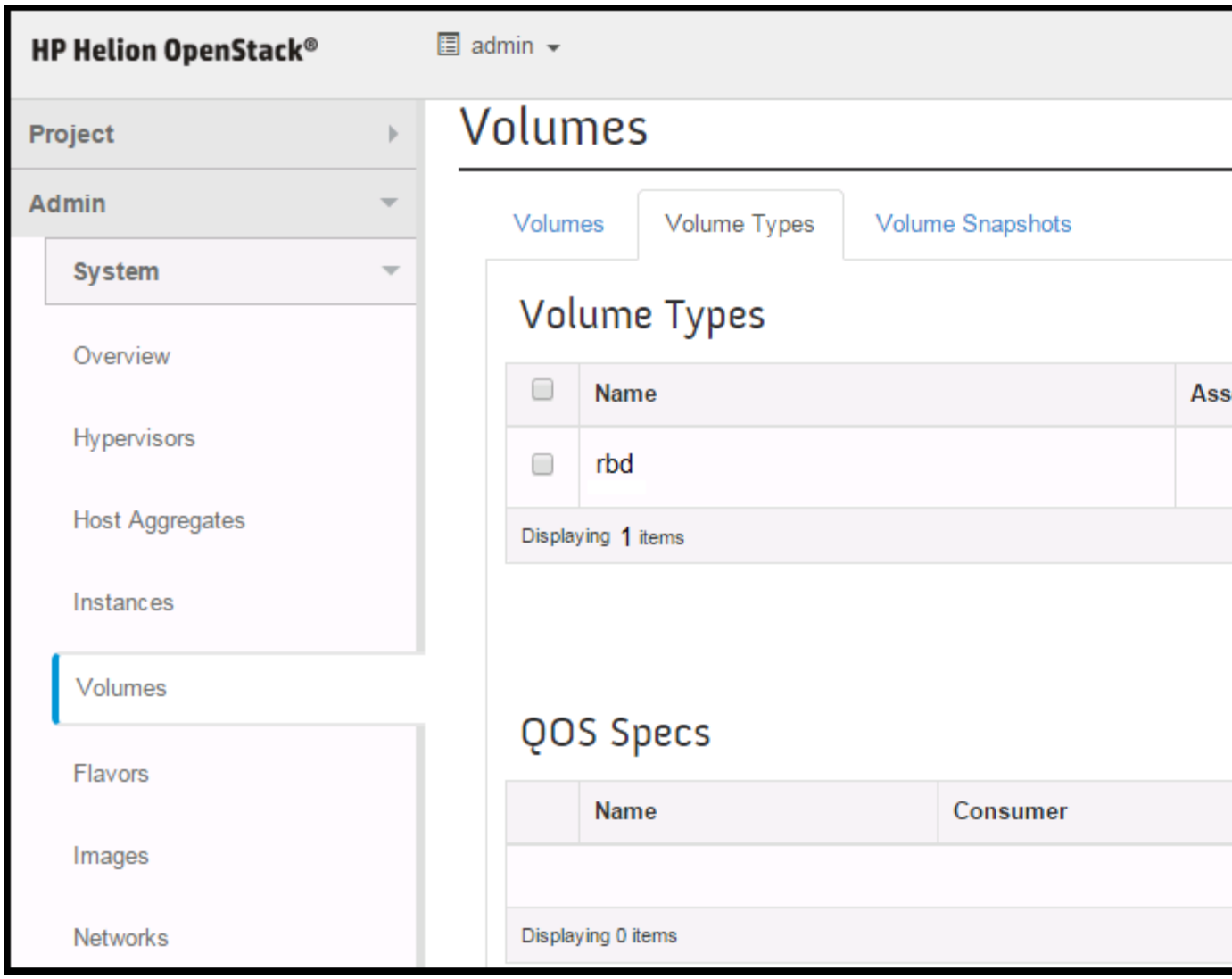
1. Log into the Horizon dashboard. The Horizon dashboard displays with the options in the left panel.
2. From the left panel, click the **Admin** tab and then click the **Volumes** tab to display the Volumes page.



3. Click **Create Volume Type** to display a dialog box.



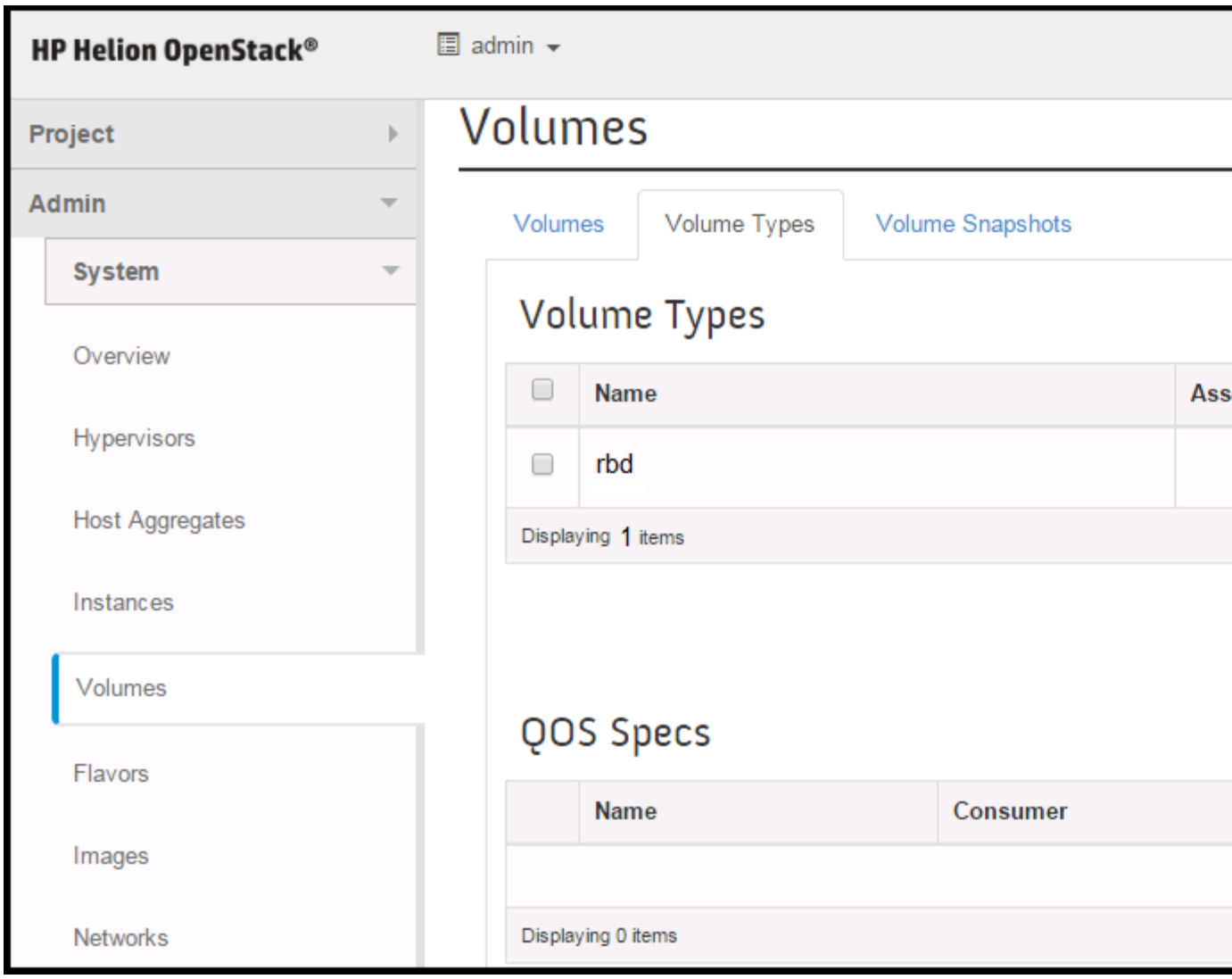
4. Enter the name of the volume type.
5. Click **Create Volume Type**. The newly created volume displays in the Volumes page.



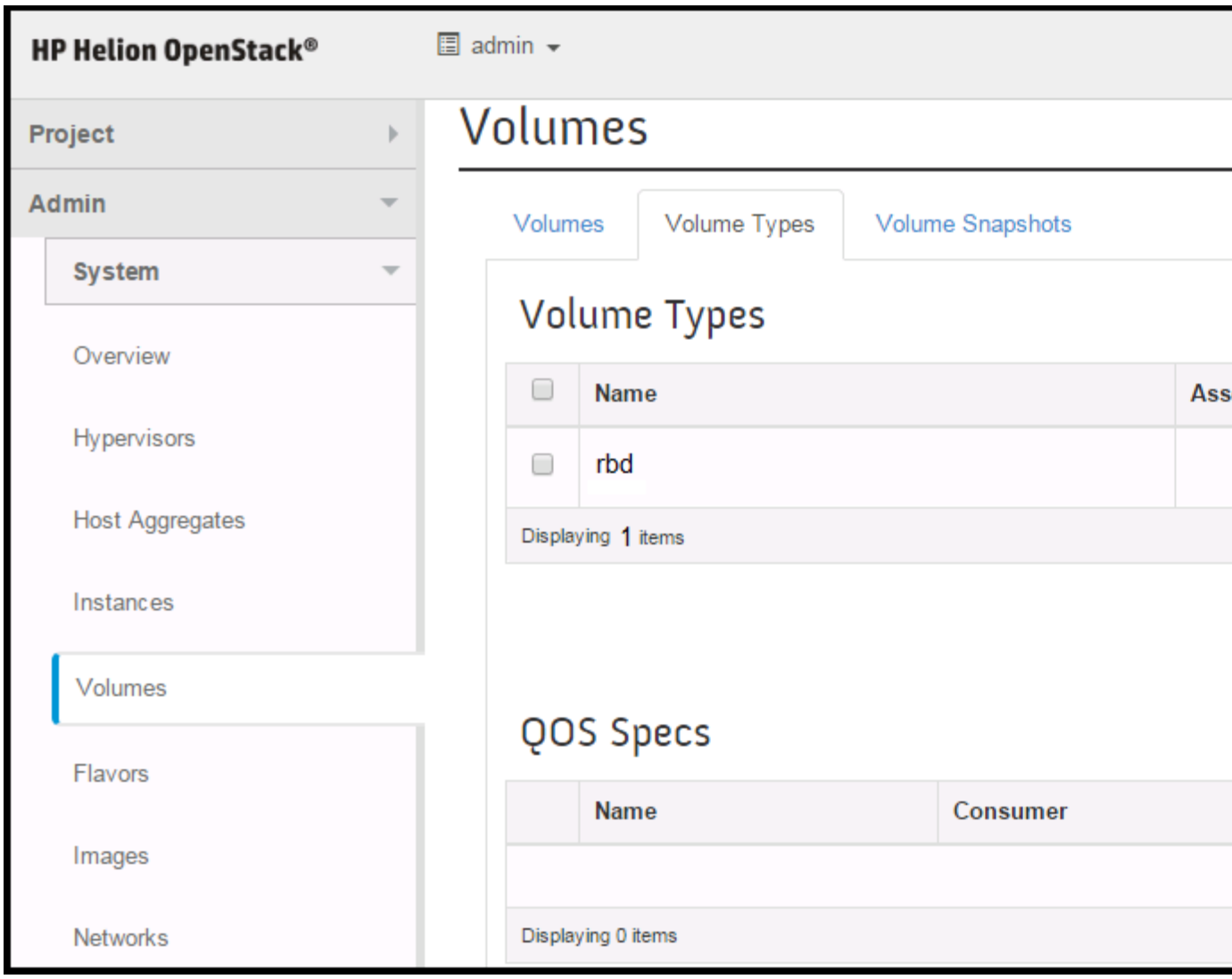
Associate the volume type to a backend

To map a volume type to a backend, do the following:

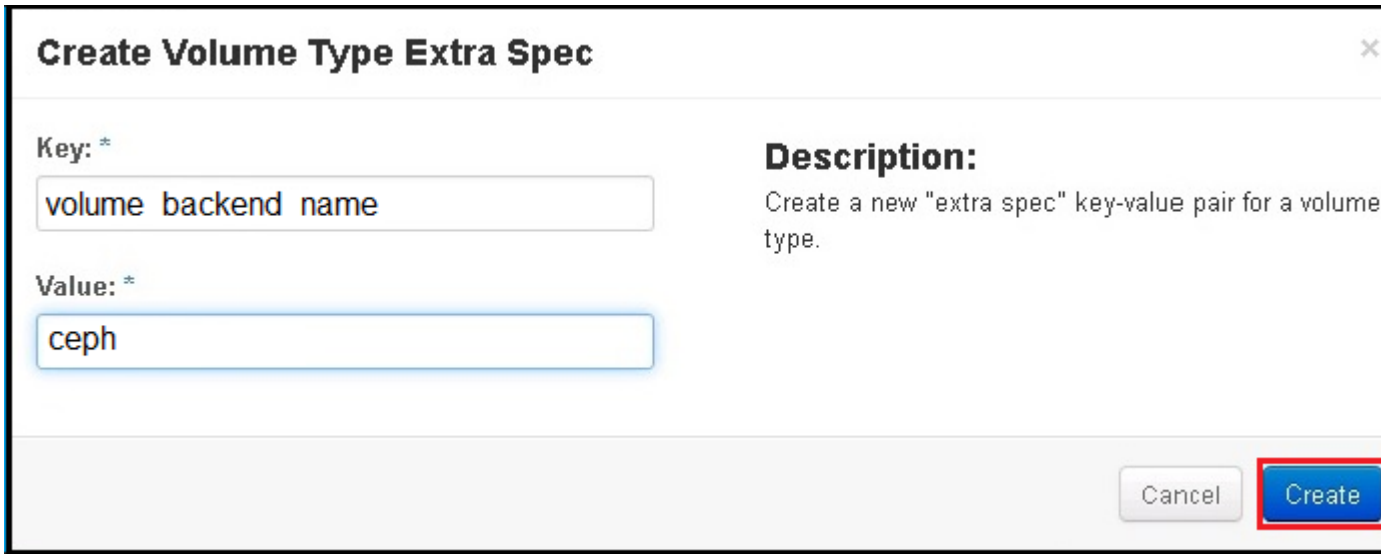
1. Login to the Overcloud Horizon dashboard. The Overcloud dashboard displays with the options in the left panel.
2. From the left panel, click the **Admin** tab and then click the **Volumes** tab to display the Volumes page.



3. Click **View Extra Specs** displayed against the volume type which you want to associate to the backend.



The **Create Volume Type Extra Specs** dialog box displays.



4. In the **Key** box, enter *ceph* as volume backend name. This is the name of the key used to specify the storage backend when provisioning volumes of this volume type.

5. In the **Value** box, enter the name of the backend to which you want to associate the volume type. For example:*ceph*.
6. Click **Create** to create the extra volume type specs.

Note: Once the volume type is mapped to the backend, you can create volumes with this volume type.

Ceph Operations

After the successful deployment of Ceph, you can perform the following ceph operations:

- Check the status of Ceph OSD and Monitor Services
- Start OSD Nodes and Monitor Services
- Stop OSD Nodes and Monitor Services

Check the status of Ceph OSD and Monitor Services

Perform the following steps to check the stays of Ceph OSD Nodes and Monitor Services:

1. Login to Deployer Node.
2. Change to the following directory:

```
~/scratch/ansible/next/hos/ansible
```

3. Run the ansible playbook:

```
ansible-playbook -i hosts/verb_hosts ceph-status.yml
```

Start OSD and Monitor Services

Perform the following steps to start OSD Nodes and Monitor Services:

1. Login to Deployer Node.
2. Change to the following directory:

```
cd ~/scratch/ansible/next/hos/ansible
```

3. Run the ansible playbook:

```
ansible-playbook -i hosts/verb_hosts ceph-start.yml
```

Stop OSD Nodes and Monitor Services

Perform the following steps to stop OSD Nodes and Monitor Services:

1. Login to Deployer Node.
2. Change to the following directory:

```
cd ~/scratch/ansible/next/hos/ansible
```

3. Run the ansible playbook:

```
ansible-playbook -i hosts/verb_hosts ceph-stop.yml
```