Red Hat Ceph Storage 3

Ceph Block Device to OpenStack Guide

Configuring Ceph, QEMU, libvirt and OpenStack to use Ceph as a back end for OpenStack.

Last Updated: 2021-05-05

Red Hat Ceph Storage 3 Ceph Block Device to OpenStack Guide

Configuring Ceph, QEMU, libvirt and OpenStack to use Ceph as a back end for OpenStack.

Legal Notice

Abstract

This document describes how to configure OpenStack and Ceph to use Ceph as a back end for Glance, Cinder, Cinder Backup and Nova.

Table of Contents

PREFACE

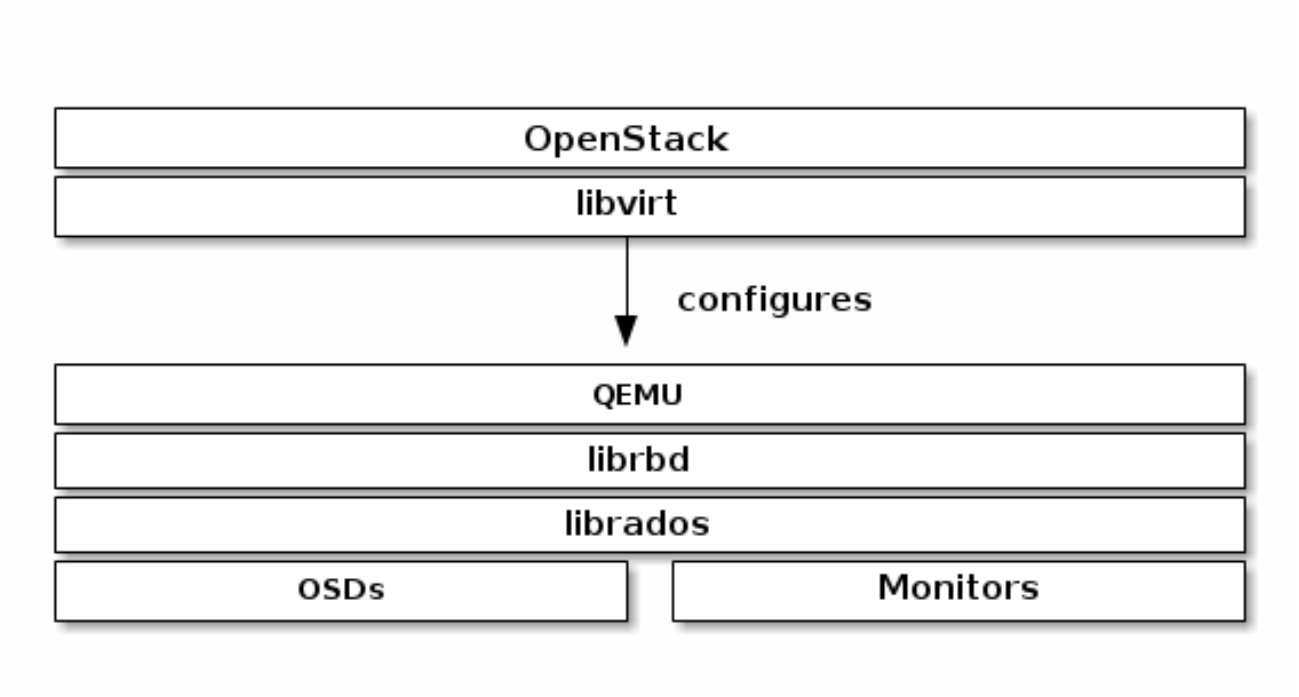
The Red Hat OpenStack Platform Director provides two methods for using Ceph as a backend for Glance, Cinder, Cinder Backup and Nova:

1. **OpenStack Creates the Ceph Cluster:** OpenStack Director can spin up a Ceph cluster, which requires configuring templates for the Ceph OSDs. OpenStack handles the installation and configuration of Ceph nodes. In this scenario, OpenStack will install Ceph monitors with OpenStack controller nodes.

2. **OpenStack Connects to an Existing Ceph Cluster:** OpenStack Director (Red Hat OpenStack Platform Director version 9 and beyond) can connect to a Ceph monitor and configure the Ceph cluster for use as a backend for OpenStack.

The foregoing methods are the preferred methods for configuring Ceph as a backend for OpenStack, because they will handle much of the installation and configuration automatically. See the Red Hat OpenStack Platform documentation for additional details.

This document details the manual procedure for configuring Ceph, QEMU, libvirt and OpenStack to use Ceph as a backend. This document is intended for use for those who do not intend to use the Red Hat OpenStack Platform Director.



NOTE

A running Ceph storage cluster and at least one OpenStack node is required to use Ceph Block Devices as a backend for OpenStack.

Three parts of OpenStack integrate with Ceph's block devices:

- **Images:** OpenStack Glance manages images for VMs. Images are immutable. OpenStack treats images as binary blobs and downloads them accordingly.

- **Volumes:** Volumes are block devices. OpenStack uses volumes to boot VMs, or to attach volumes to running VMs. OpenStack manages volumes using Cinder services. Ceph can serve as a black end for OpenStack Cinder and Cinder Backup.

- **Guest Disks:** Guest disks are guest operating system disks. By default, when booting a virtual machine, its disk appears as a file on the filesystem of the hypervisor (usually under **/var/lib/nova/instances/<uuid>/**`). OpenStack Glance can store images in a Ceph Block Device, and can use Cinder to boot a VM using a copy-on-write clone of an image.

IMPORTANT

Ceph doesn't support QCOW2 for hosting a virtual machine disk. To boot virtual machines in Ceph (ephemeral backend or boot from volume), the Glance image format must be RAW.

OpenStack can use Ceph for images, volumes or guest disks VMs. There is no requirement to use all three.

CHAPTER 1. CREATING CEPH POOLS

By default, Ceph block devices use the **rbd** pool. You may use any available pool. The following example creates pools for Cinder, Cinder backups, Glance and Nova respectively. Ensure the Ceph cluster is running, then create the pools.

Red Hat recommends using the Ceph Placement Group's per Pool Calculator to calculate a suitable number of placement groups for the pools. See the Pools chapter in the *Storage Strategies* guide for Red Hat Ceph Storage 3 for details on creating pools. In the following example, **128** is the number of placement groups.

```
# ceph osd pool create volumes 128
# ceph osd pool create backups 128
# ceph osd pool create images 128
# ceph osd pool create vms 128
```

CHAPTER 2. INSTALLING AND CONFIGURING CEPH CLIENTS

The **nova-compute**, **cinder-backup** and on the **cinder-volume** node require both the Python bindings and the client command line tools:

```
# yum install python-rbd
# yum install ceph-common
```

The **glance-api** node requires the Python bindings for **librbd**:

```
# yum install python-rbd
```

2.1. COPYING CEPH CONFIGURATION FILE TO OPENSTACK NODES

The nodes running **glance-api**, **cinder-volume**, **nova-compute** and **cinder-backup** act as Ceph clients. Each requires the Ceph configuration file. Copy the Ceph configuration file from the monitor node to the OSP nodes.

```
# scp /etc/ceph/ceph.conf osp:/etc/ceph
```

2.2. SETTING UP CEPH CLIENT AUTHENTICATION

From a Ceph monitor node, create new users for Cinder, Cinder Backup and Glance.

```
# ceph auth get-or-create client.cinder mon 'allow r' osd 'allow class-read object_prefix rbd_children,
allow rwx pool=volumes, allow rwx pool=vms, allow rx pool=images'

# ceph auth get-or-create client.cinder-backup mon 'allow r' osd 'allow class-read object_prefix
rbd_children, allow rwx pool=backups'

# ceph auth get-or-create client.glance mon 'allow r' osd 'allow class-read object_prefix rbd_children,
allow rwx pool=images'
```

Add the keyrings for **client.cinder**, **client.cinder-backup** and **client.glance** to the appropriate nodes and change their ownership:

```
# ceph auth get-or-create client.cinder | ssh {your-volume-server} sudo tee
/etc/ceph/ceph.client.cinder.keyring
# ssh {your-cinder-volume-server} chown cinder:cinder /etc/ceph/ceph.client.cinder.keyring

# ceph auth get-or-create client.cinder-backup | ssh {your-cinder-backup-server} tee
/etc/ceph/ceph.client.cinder-backup.keyring
# ssh {your-cinder-backup-server} chown cinder:cinder /etc/ceph/ceph.client.cinder-backup.keyring

# ceph auth get-or-create client.glance | ssh {your-glance-api-server} sudo tee
/etc/ceph/ceph.client.glance.keyring
# ssh {your-glance-api-server} chown glance:glance /etc/ceph/ceph.client.glance.keyring
```

Nodes running **nova-compute** need the keyring file for the **nova-compute** process:

```
# ceph auth get-or-create client.cinder | ssh {your-nova-compute-server} tee
/etc/ceph/ceph.client.cinder.keyring
```

Nodes running **nova-compute** also need to store the secret key of the **client.cinder** user in **libvirt**. The **libvirt** process needs it to access the cluster while attaching a block device from Cinder. Create a temporary copy of the secret key on the nodes running **nova-compute**:

```
# ceph auth get-key client.cinder | ssh {your-compute-node} tee client.cinder.key
```

If the storage cluster contains Ceph Block Device images that use the **exclusive-lock** feature, ensure that all Ceph Block Device users have permissions to blacklist clients:

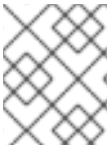
```
# ceph auth caps client.{ID} mon 'allow r, allow command "osd blacklist"' osd '{existing-OSD-user-capabilities}'
```

Return to the compute node.

```
# ssh {your-compute-node}
```

Generate a UUID for the secret, and save the UUID of the secret for configuring **nova-compute** later.

```
# uuidgen > uuid-secret.txt
```



NOTE

You don't necessarily need the UUID on all the compute nodes. However from a platform consistency perspective, it's better to keep the same UUID.

Then, on the compute nodes, add the secret key to **libvirt** and remove the temporary copy of the key:

```
cat > secret.xml <<EOF
<secret ephemeral='no' private='no'>
  <uuid>`cat uuid-secret.txt`</uuid>
  <usage type='ceph'>
    <name>client.cinder secret</name>
  </usage>
</secret>
EOF
```

```
# virsh secret-define --file secret.xml
# virsh secret-set-value --secret $(cat uuid-secret.txt) --base64 $(cat client.cinder.key) && rm client.cinder.key secret.xml
```

CHAPTER 3. CONFIGURING OPENSTACK TO USE CEPH

3.1. CONFIGURING CINDER

The **cinder-volume** nodes require the Ceph block device driver, the **volume** pool, the user and the UUID of the secret to interact with Ceph block devices. To configure Cinder, perform the following steps:

1. Open the Cinder configuration file.

```
# vim /etc/cinder/cinder.conf
```

2. In the **[DEFAULT]** section, enable Ceph as a backend for Cinder.

```
enabled_backends = ceph
```

3. Ensure that the Glance API version is set to 2. If you are configuring multiple cinder back ends in **enabled_backends**, the **glance_api_version = 2** setting must be in the **[DEFAULT]** section and not the **[ceph]** section.

```
glance_api_version = 2
```

4. Create a **[ceph]** section in the **cinder.conf** file. Add the Ceph settings in the following steps under the **[ceph]** section.

5. Specify the **volume_driver** setting and set it to use the Ceph block device driver. For example:

```
volume_driver = cinder.volume.drivers.rbd.RBDDriver
```

6. Specify the cluster name and Ceph configuration file location. In typical deployments the Ceph cluster has a cluster name of **ceph** and a Ceph configuration file at **/etc/ceph/ceph.conf**. If the Ceph cluster name is not **ceph**, specify the cluster name and configuration file path appropriately. For example:

```
rbd_cluster_name = us-west
rbd_ceph_conf = /etc/ceph/us-west.conf
```

7. By default, OSP stores Ceph volumes in the **rbd** pool. To use the **volumes** pool created earlier, specify the **rbd_pool** setting and set the **volumes** pool. For example:

```
rbd_pool = volumes
```

8. OSP does not have a default user name or a UUID of the secret for volumes. Specify **rbd_user** and set it to the **cinder** user. Then, specify the **rbd_secret_uuid** setting and set it to the generated UUID stored in the **uuid-secret.txt** file. For example:

```
rbd_user = cinder
rbd_secret_uuid = 4b5fd580-360c-4f8c-abb5-c83bb9a3f964
```

9. Specify the following settings:

```
rbd_flatten_volume_from_snapshot = false
```

```
rbd_max_clone_depth = 5
rbd_store_chunk_size = 4
rados_connect_timeout = -1
```

The resulting configuration should look something like this:

```
[DEFAULT]
enabled_backends = ceph
glance_api_version = 2
...

[ceph]
volume_driver = cinder.volume.drivers.rbd.RBDDriver
rbd_cluster_name = ceph
rbd_pool = volumes
rbd_user = cinder
rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_flatten_volume_from_snapshot = false
rbd_secret_uuid = 4b5fd580-360c-4f8c-abb5-c83bb9a3f964
rbd_max_clone_depth = 5
rbd_store_chunk_size = 4
rados_connect_timeout = -1
```

NOTE

Consider removing the default **[lvm]** section and its settings.

3.2. CONFIGURING CINDER BACKUP

The **cinder-backup** node requires a specific daemon. To configure Cinder backup, perform the following steps:

1. Open the Cinder configuration file.

```
# vim /etc/cinder/cinder.conf
```

2. Go to the **[ceph]** section of the configuration file.
3. Specify the **backup_driver** setting and set it to the Ceph driver.

```
backup_driver = cinder.backup.drivers.ceph
```

4. Specify the **backup_ceph_conf** setting and specify the path to the Ceph configuration file.

```
backup_ceph_conf = /etc/ceph/ceph.conf
```

NOTE

The Cinder backup Ceph configuration file may be different from the Ceph configuration file used for Cinder. For example, it may point to a different Ceph cluster.

5. Specify the Ceph pool for backups.

```
backup_ceph_pool = backups
```

NOTE

While it is possible to use the same pool for Cinder Backups as used with Cinder, it is NOT recommended. Consider using a pool with a different CRUSH hierarchy.

6. Specify the **backup_ceph_user** setting and specify the user as **cinder-backup**.

```
backup_ceph_user = cinder-backup
```

7. Specify the following settings:

```
backup_ceph_chunk_size = 134217728
backup_ceph_stripe_unit = 0
backup_ceph_stripe_count = 0
restore_discard_excess_bytes = true
```

With the Cinder settings included, the **[ceph]** section of the **cinder.conf** file should look something like this:

```
[ceph]
volume_driver = cinder.volume.drivers.rbd.RBDDriver
rbd_cluster_name = ceph
rbd_pool = volumes
rbd_user = cinder
rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_flatten_volume_from_snapshot = false
rbd_secret_uuid = 4b5fd580-360c-4f8c-abb5-c83bb9a3f964
rbd_max_clone_depth = 5
rbd_store_chunk_size = 4
rados_connect_timeout = -1

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

Check to see if Cinder backup is enabled under **/etc/openstack-dashboard/**. The setting should be in a file called **local_settings**, or **local_settings.py**. For example:

```
cat  /etc/openstack-dashboard/local_settings | grep enable_backup
```

If **enable_backup** is set to **False**, set it to **True**. For example:

```
OPENSTACK_CINDER_FEATURES = {
    'enable_backup': True,
}
```

3.3. CONFIGURING GLANCE

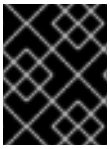
To use Ceph block devices by default, edit the **/etc/glance/glance-api.conf** file. Uncomment the following settings if necessary and change their values accordingly. If you used different pool, user or Ceph configuration file settings apply the appropriate values.

```
# vim /etc/glance/glance-api.conf
```

```
stores = rbd
default_store = rbd
rbd_store_chunk_size = 8
rbd_store_pool = images
rbd_store_user = glance
rbd_store_ceph_conf = /etc/ceph/ceph.conf
```

To enable copy-on-write (CoW) cloning set **show_image_direct_url** to **True**.

```
show_image_direct_url = True
```



IMPORTANT

Enabling CoW exposes the back end location via Glance's API, so the endpoint should not be publicly accessible.

Disable cache management if necessary. The **flavor** should be set to **keystone** only, not **keystone+cachemanagement**.

```
flavor = keystone
```

Red Hat recommends the following properties for images:

```
hw_scsi_model=virtio-scsi
hw_disk_bus=scsi
hw_qemu_guest_agent=yes
os_require_quiesce=yes
```

The **virtio-scsi** controller gets better performance and provides support for discard operations. For systems using SCSI/SAS drives, connect every cinder block device to that controller. Also, enable the QEMU guest agent and send **fs-freeze/thaw** calls through the QEMU guest agent.

3.4. CONFIGURING NOVA

On every **nova-compute** node, edit the Ceph configuration file to configure the ephemeral backend for Nova and to boot all the virtual machines directly into Ceph.

1. Open the Ceph configuration file.

```
# vim /etc/ceph/ceph.conf
```

2. Add the following section to the **[client]** section of the Ceph configuration file:

```
[client]
rbd cache = true
rbd cache writethrough until flush = true
rbd concurrent management ops = 20
admin socket = /var/run/ceph/guests/$cluster-$type.$id.$pid.$cctid.asok
log file = /var/log/ceph/qemu-guest-$pid.log
```

3. Make directories for the admin socket and log file, and change their permissions to use the **qemu** user and **libvirtd** group.

```
mkdir -p /var/run/ceph/guests/ /var/log/ceph/
chown qemu:libvirt /var/run/ceph/guests /var/log/ceph/
```

NOTE

The directories must be allowed by SELinux or AppArmor.

On every **nova-compute** node, edit the **/etc/nova/nova.conf** file under the **[libvirt]`** section and configure the following settings:

```
[libvirt]
images_type = rbd
images_rbd_pool = vms
images_rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_user = cinder
rbd_secret_uuid = 4b5fd580-360c-4f8c-abb5-c83bb9a3f964
disk_cachemodes="network=writeback"
inject_password = false
inject_key = false
inject_partition = -2
live_migration_flag="VIR_MIGRATE_UNDEFINE_SOURCE,VIR_MIGRATE_PEER2PEER,VIR_MIGR
ATE_LIVE,VIR_MIGRATE_PERSIST_DEST,VIR_MIGRATE_TUNNELLED"
hw_disk_discard = unmap
```

If the Ceph configuration file is not **/etc/ceph/ceph.conf**, provide the correct path. Replace the UUID in **rbd_user_secret** with the UUID in the **uuid-secret.txt** file.

3.5. RESTARTING OPENSTACK SERVICES

To activate the Ceph block device drivers, load the block device pool names and Ceph user names into the configuration, restart the appropriate OpenStack services after modifying the corresponding configuration files.

```
# systemctl restart openstack-cinder-volume
# systemctl restart openstack-cinder-backup
# systemctl restart openstack-glance-api
# systemctl restart openstack-nova-compute
```