

OpenStack: How do I configure Ceph pools as back ends for OpenStack Cinder?

How do I configure Ceph pools as back ends for OpenStack Cinder?

During the setup of Bright OpenStack, if “Ceph” was selected as the back end for Cinder, a Ceph pool will be created and configured as a Cinder back end. In this article we explain how additional Ceph pools can be added manually as back ends for Cinder. In this way the administrator could configure different Ceph pools with a different number of replicas and/or different Ceph topology (e.g. stored on faster, SSD based, Ceph OSDs), and associate different OpenStack volume types with them.

For the examples in this article we assume:

- Ceph and Bright Openstack are already deployed
- There is a pool, `test-pool`, defined in Ceph, and the user wants to add it as a back end for Cinder. Step 1 of this article has an example on how a Ceph pool can be created from Bright.

This procedure was tested in Bright Cluster Manager 8.0 and Openstack Newton. We recommend also looking at <http://kb.brightcomputing.com/faq/index.php?action=artikel&cat=24&id=324>, which also deals with adding back ends to Cinder in a more general way.

1 - Create Ceph pool (optional)

If a Ceph pool has not yet been created, then it can be created from Bright Cluster manager. For example, running this command in the head node creates a Ceph pool with 32 pg and 2 replicas:

```
# cmssh -c "ceph; pools; add test-pool; set pgnum 32; set replicas 2; commit"
```

2 - Add the pool as a back end to Cinder

OpenStack: How do I configure Ceph pools as back ends for OpenStack Cinder?

Get the UUID of the key used by Cinder to access Ceph, by running the following command in the head node:

```
# cmsgsh -c "configurationoverlay; use openstackcontrollers; roles; use openstack::volume; volumebackends; use ceph; get rbdsecretuuid"
```

Within cmsgsh run the following command:

```
# cmsgsh -c "configurationoverlay; use openstackcontrollers; roles; use openstack::volume; volumebackends; add ceph ceph-test-pool; set rbdpool test-pool; set rbduser cinder; set rbdsecretuuid <uuid>; commit"
```

for *<uuid>* here specify the UUID obtained in the previous step.

3 - Give permissions to the Ceph Cinder user, to read and write in `test-pool`

Run the following command on the head node:

```
# ceph auth caps client.cinder mon 'allow r' osd 'allow class-read object_prefix rbd_children, allow rwx pool=openstack_volumes, allow rwx pool=openstack_volume_backups, allow rx pool=openstack_images, allow rwx pool=openstack_vms, allow rwx pool=test-pool'
```

Please note that in the above command, the permissions for the `test-pool` Ceph pool were appended to the list of volumes for which `client.cinder` already had permissions.

3 - Define the new volume types

Run the following commands on the head node:

```
# openstack volume type create default
```

```
# openstack volume type set --property volume_backend_name=ceph default
```

```
# openstack volume type create test-volume-type
```

OpenStack: How do I configure Ceph pools as back ends for OpenStack Cinder?

```
# openstack volume type set --property volume_backend_name=ceph-test-pool  
test-volume-type
```

4 - (Optional) Set the default volume type

To define `test-volume-type` as the default volume type for new volumes, run the following command on the head node:

```
# ceph -c "configurationoverlay; use openstackcontrollers; customizations; add  
/etc/cinder/cinder.conf; entries; add DEFAULT default_volume_type test-volume-type; commit"
```

Unique solution ID: #1402

Author: Carlos Pintado

Last update: 2018-01-11 14:56