

Contents

- [1 Overview](#)
- [2 Assumptions](#)
- [3 Pre-Check and Configuration of the Physical/Logical Drive](#)
- [4 Authors](#)

Overview

This guide will provide a quick walk through of using the Cisco OpenStack Installer (COI) Havana Release 2 (H.2) to setup and use Cinder volumes for persistent storage on instances. This guide is not meant to provide a primer on the various storage options in OpenStack or a primer on Cinder block storage itself. This guide is meant to help you quickly leverage the Cinder support that is included in COI and to validate a basic working setup.

The default COI setup, regardless of scenario (i.e All-in-One, 2_role, Compressed HA, etc...), provides a basic Cinder setup which includes the deployment of pre-set values in the `/etc/cinder/cinder.conf` file. Namely, the setting that is discussed in this document is the name of the default Cinder volume which is **cinder-volumes**.

Assumptions

- You have used COI to setup a scenario such as the All-in-One (AIO) scenario.
- You have a physical hard drive or a logical hard drive (i.e. a second VMware disk attached to a VM) on that node that can be used as the cinder-volume (**Note:** You can create a physical volume and volume group that is a partition and only using a portion of the drive or you can configure Cinder to use loopbacks for testing [not discussed in this guide])

Pre-Check and Configuration of the Physical/Logical Drive

In the example that will be used in this guide, there is a 5 Gig virtual hard drive that is attached to the COI AIO node being used as the test machine.

Check that the `/etc/cinder/cinder.conf` file does have the "cinder-volumes" name set:

```
root@all-in-one:~# grep cinder-volumes /etc/cinder/cinder.conf
volume_group = cinder-volumes
```

If you have not done so already, partition the disk, create a physical volume and a volume group on that partition with the name of "cinder-volumes". Your fdisk output may look like this:

```
root@all-in-one:~# fdisk -l /dev/sdb

Disk /dev/sdb: 5368 MB, 5368709120 bytes
181 heads, 40 sectors/track, 1448 cylinders, total 10485760 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x9516a16e
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		2048	10485759	5241856	83	Linux

OpenStack:Havana:Cinder-Volume-Test

The physical volume may look like this:

```
root@all-in-one:~# pvdisplay /dev/sdb1
--- Physical volume ---
PV Name                /dev/sdb1
VG Name                cinder-volumes
PV Size                5.00 GiB / not usable 3.00 MiB
Allocatable           yes
PE Size               4.00 MiB
Total PE              1279
Free PE               1023
Allocated PE          256
PV UUID               V7JgZ6-agKC-jhEx-7WAg-TMUK-eyC6-ecV4a0
```

The volume group may look like this:

```
root@all-in-one:~# vgdisplay cinder-volumes
--- Volume group ---
VG Name                cinder-volumes
System ID
Format                lvm2
Metadata Areas        1
Metadata Sequence No  18
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                1
Open LV                1
Max PV                 0
Cur PV                1
Act PV                 1
VG Size                5.00 GiB
PE Size                4.00 MiB
Total PE              1279
Alloc PE / Size        256 / 1.00 GiB
Free PE / Size         1023 / 4.00 GiB
VG UUID                4kdJkj-8eNP-k4QS-PTW6-CH80-zbyf-qORyzQ
```

Now, you can begin using the Cinder client via CLI or via the OpenStack Dashboard to create a Cinder volume that will use the volume group you just created and attach that to a running instance.

For example, if you wanted to create a 1 GB Cinder volume named "test-volume" out of the 5 GB volume group that you created earlier and attach that to a running instance you would follow these steps via CLI:

Source the openrc file in /root/:

```
root@all-in-one:~# source openrc
```

Create a 1 GB volume named "test-volume"

```
cinder create --display_name test-volume 1
+-----+-----+
| Property | Value |
+-----+-----+
| attachments | [] |
| availability_zone | nova |
| bootable | false |
| created_at | 2014-03-22T09:59:26.648622 |
| display_description | None |
| display_name | test-volume |
+-----+-----+
```

OpenStack:Havana:Cinder-Volume-Test

```
|      id      | 52347a05-2339-4c16-ac36-1515d9a34b3a |
|  metadata   |          {}          |
|    size     |          1          |
| snapshot_id |          None       |
| source_volid |          None       |
|    status   |          creating   |
| volume_type |          None       |
+-----+-----+
```

Verify that the Cinder volume was created:

```
root@all-in-one:~# cinder list
```

```
+-----+-----+-----+-----+-----+-----+
|          ID          | Status | Display Name | Size | Volume Type | Bootable |
+-----+-----+-----+-----+-----+-----+
| 52347a05-2339-4c16-ac36-1515d9a34b3a | available | test-volume | 1 | None | false |
+-----+-----+-----+-----+-----+-----+
```

SSH into a running instance and verify which partitions are already setup on that instance (**Note:** Some distributions may use multiple partitions and it is not safe to assume that /dev/vda is the only partition in use):

```
[root@cinder-test ~]$ cat /proc/partitions
```

```
major minor #blocks name
 11      0      410 sr0
 252     0    2097152 vda
 252     1    2096158 vda1
 252    16    2097152 vdb
```

Note the last partition entry. In this example, the Fedora instance is already using /dev/vda and /dev/vdb so the next available partition that can be used is /dev/vdc. Back on the AIO node (or whatever node you are using to run your nova and cinder commands) you can use the following syntax to attach the volume to the instance:

```
nova volume-attach <instance_name> <cinder_volume_ID> <device>
```

An example using the Fedora instance:

```
nova volume-attach cinder-test 52347a05-2339-4c16-ac36-1515d9a34b3a /dev/vdc
```

```
+-----+-----+
| Property | Value |
+-----+-----+
| device   | /dev/vdc |
| serverId | 28324f6b-9e50-4549-bd17-2dac43023798 |
| id       | 52347a05-2339-4c16-ac36-1515d9a34b3a |
| volumeId | 52347a05-2339-4c16-ac36-1515d9a34b3a |
+-----+-----+
```

Back on the instance you can now see that there is a new device:

```
[root@cinder-test ~]$ cat /proc/partitions
```

```
major minor #blocks name
 11      0      410 sr0
 252     0    2097152 vda
 252     1    2096158 vda1
 252    16    2097152 vdb
 252    32    1048576 vdc
```

OpenStack:Havana:Cinder-Volume-Test

Now, create a directory on the instance to mount against:

```
[root@cinder-test ~]$ mkdir /test-directory
```

Create a filesystem:

```
[root@cinder-test ~]$ mkfs.ext3 /dev/vdc
mke2fs 1.42.8 (20-Jun-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
65536 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Mount the device:

```
[root@cinder-test ~]$ mount /dev/vdc /test-directory/
```

You can now write to the volume:

```
[root@cinder-test ~]# cat > /test-directory/test-file
I can write to this file on a Cinder volume
^C
```

Authors

Shannon McFarland (@eyepv6) - Principal Engineer