

OpenStack Havana All-in-One Lab: Cinder addition for Block Storage

It seems like forever ago that we began, and it's time to keep the ball rolling with our OpenStack Havana All-in-One build on VMware Workstation series that had become idle in the last little while.

Lot's of great feedback has been coming in and one of the key things that we wanted to do to get to our next step was to enable OpenStack Block Storage, also known as Cinder.

I'm going to assume that you've gotten through the first few steps, and just in case you haven't already gotten on board with our lab build, here are the posts to use to get caught up:

- OpenStack Havana All-in-One lab on VMware Workstation
- OpenStack Havana All-in-One Lab: Booting your first Nova instance
- OpenStack Havana All-in-One Lab: Using the Horizon Dashboard

Now that we are all up to the same point, let's get some block storage happening!

We can enable Cinder internally on our existing machine, but I wanted to show you how to do this by using a secondary volume. Since we only have one virtual disk on our virtual machine, we need to power down your virtual machine and then we will follow these simple steps to add our volume before installing Cinder.

Once powered down, open your Virtual Machine Settings (right click: Edit Settings) and click the **Add...** button to add a new device:



We select **Hard Disk** as the hardware type and click **Next**



Leave the default option of SCSI and click **Next**



Choose **Create a new virtual disk** and click **Next**



Increase the **Maximum disk size** to **40 GB**, and choose **Store virtual disk as a single file** before clicking **Next**



Let's name this as **AIO-HAVANA-CINDER01.vmdk** so that we know what it's for and then click **Finish**



Now we will boot our AIO-HAVANA virtual machine and do the following:

- Log in to the console using your regular account
- Enter the elevated access with **sudo su -**
- Change directory into our GitHub source folder with **cd OpenStack-All-in-One-Havana**
- Update our GitHub source by running **git pull**



What we have now with our updated source code is the addition of the **cinder.sh** shell script which will deploy and configure our Cinder services and dependencies. You can confirm the file is there with the **ls** command.

Launch the script by typing this:

```
sh cinder.sh
```



The installation will take a few minutes, and when it is all done you will see the final commands which show the services being restarted as shown here:



Now we need to test out that our Cinder services are working which we will be doing with the Python Cinder client that was deployed as part of our script.

The command to create a Cinder volume is **cinder create -display_name <volumename> <size>** which for our example to do a quick test is going to be a 1 GB volume named test:

```
cinder create -display_name test 1
```



Provided everything goes properly, you will see the results above and you can confirm the results with an easy command:

```
cinder list
```



Now we have confirmed our volume has been created and the status will show as **available** once it is fully built. Because we want to clean up after ourselves, we will use the following command to delete our test volume:

```
cinder delete test
```



By running **cinder list** we can see the volume as it is being deleted. This happens quickly with a 1

GB volume, so you may not see it as above, but as we continue to use our environment these are the typical commands for creating and listing volumes.

Horizon and Cinder

Let's recap where we have come from. We installed our virtual lab on the all-in-one virtual machine, created a Nova instance, viewed our OpenStack cloud in our Horizon dashboard, and now we have installed our OpenStack Cinder Block Storage environment. Not bad with only a few configuration steps thanks to our deployment scripts.

The command line is only one part of our Cinder work of course, so next we will look at the same process of creating a Cinder volume in the Horizon environment.

We will quickly go through a few steps in our dashboard by browsing to the site and logging in with our Admin account (remember our password is **openstack** for our lab build)



In your dashboard, click on the **Project** tab, then click the **Volumes** link in the left hand pane. This brings you to the Volumes screen where you will click on the **Create Volume** button



We don't need much information to create a volume as we saw in the command line version of this process. For our simple example, we will use just the **Volume Name** option, the **Size (GB)** option, and the **Volume Source** option with simple options:

- Name: **test**
- Size: **2 GB**
- Source: **No source, empty volume**

Once you select those options you will click the **Create Volume** button. We will revisit all of the other options in a later post, but our initial goal is just getting our environment up and running to start exploring more ☐



In your Volumes view you will see the volume show up in the list with a status of **Creating**



In a few short seconds (that's why we use small volumes to test) you will see the status change to **Available** and once that happens you will have a button in the Actions column which lets you either do a Create Snapshot or Delete Volume action.



We also have an option here to **Edit Attachment** but I'm going to leave you in suspense a little tiny bit longer because that is the subject of our next post ☐