Linux Academy

Live! Lab

Creating and Mounting a Block Storage Volume

Contents

Lab Connection Information

- Labs may take up to five minutes to build
- Access to an AWS Console is provided on the Live! Lab page, along with your login credentials
- Ensure you are using the N. Virginia region
- Labs will automatically end once the alloted amount of time finishes

Related Courses

OpenStack Foundation Certified OpenStack Administrator

Related Videos

Launch a New Instance

Block Storage - Cinder Overview

Create and Mount a Block Storage Volume

Need Help?

Linux Academy Community

... and you can always send in a support ticket on our website to talk to an instructor!

Introduction

This lab reviews creating a block storage volume with OpenStack and mounting it to an available instance. We begin by generating the necessary key pair and instance itself, then move on volume creation and attachment.

Log in to your terminal using the SSH details provided on the **Live! Lab** page. Also, log in to the **Horizon Dashboard** using the *demo* user.

Creating an Instance

Before we begin, we need to ensure that our credentials are properly sorted. From the **Horizon Dashboard**, with the *demo* project selected, select **Access & Security**, then **Download OpenStack RC File**, under **API Access**.

Either open and copy the contents of the file into a *demo.sh* file on your OpenStack server, or `scp` the file up to your server, changing the name to *demo.sh*.

Source the file, inputting the OpenStack password for *demo* when prompted:

```
root@ubuntu-openstack:~# source demo.sh
```

Before we can attach a block storage volume, we need an instance from which to work. Before we do this, we must generate a key pair and change it to have the appropriate permissions:

```
root@ubuntu-openstack:~# nova keypair-add KEY > KEY.pem
root@ubuntu-openstack:~# chmod 600 KEY.pem
```

You can check your available key pairs by running:

```
root@ubuntu-openstack:~# nova keypair-list
+------+------+-------------------------------------------------+
| Name | Type | Fingerprint                                     |
+------+------+-------------------------------------------------+
| KEY  | ssh  | 25:85:9b:6b:74:eb:d7:dd:ea:6b:86:48:d1:58:7b:d3 |
+------+------+-------------------------------------------------+
```

Review the available instance images and flavors before creating your instance:

```
root@ubuntu-openstack:~# nova image-list
+------------------------------------+----------------------------------+--------+--------+
| ID                                 | Name                             | Status | Server |
+------------------------------------+----------------------------------+--------+--------+
| ab2a33-cdd2-4656-b961-227005af755f | cirros-0.3.4-x86_64-uec          | ACTIVE |        |
| 94711e-0040-4daa-9a20-b4d2d92d7eb6 | cirros-0.3.4-x86_64-uec-kernel  | ACTIVE |        |
| 44fed9-0bb2-488a-9567-e6acbc85bce1 | cirros-0.3.4-x86_64-uec-ramdisk | ACTIVE |        |
+------------------------------------+---------------------------------+--------+--------+
root@ubuntu-openstack:~# nova flavor-list
+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| ID | Name      | Memory_MB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor | Is_Public |
+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| 1  | m1.tiny   | 512       | 1    | 0         |      | 1     | 1.0         | True      |
| 2  | m1.small  | 2048      | 20   | 0         |      | 1     | 1.0         | True      |
| 3  | m1.medium | 4096      | 40   | 0         |      | 2     | 1.0         | True      |
| 4  | m1.large  | 8192      | 80   | 0         |      | 4     | 1.0         | True      |
| 5  | m1.xlarge | 16384     | 160  | 0         |      | 8     | 1.0         | True      |
+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
```

We want to create an instance using the *cirros-0.3.4-x86_64-uec* image, and *m1.tiny* flavor, with *instance2* being the instance name:

```
root@ubuntu-openstack:~# nova boot --image cirros-0.3.4-x86_64-uec --flavor m1.tiny --key_
name KEY instance2
+--------------------------------------+-------------------------------------------------+
| Property                             | Value                                           |
+--------------------------------------+-------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                          |
| OS-EXT-AZ:availability_zone          |                                                 |
| OS-EXT-STS:power_state               | 0                                               |
| OS-EXT-STS:task_state                | scheduling                                      |
| OS-EXT-STS:vm_state                  | building                                        |
| OS-SRV-USG:launched_at               | -                                               |
| OS-SRV-USG:terminated_at             | -                                               |
| accessIPv4                           |                                                 |
| accessIPv6                           |                                                 |
| adminPass                            | tv7pwTzMS4CK                                    |
| config_drive                         |                                                 |
| created                              | 2016-04-05T17:27:44Z                            |
| flavor                               | m1.tiny (1)                                     |
| hostId                               |                                                 |
| id                                   | 253619cf-127c-46a1-a2aa-a273cedf6a85            |
| image                                | cirros-0.3.4-x86_64-uec (ab259a33-227005af755f) |
| key_name                             | KEY                                             |
| metadata                             | {}                                              |
| name                                 | instance2                                       |
| os-extended-volumes:volumes_attached | []                                              |
| progress                             | 0                                               |
| security_groups                      | default                                         |
| status                               | BUILD                                           |
| tenant_id                            | 32f8a0f02393481cba2f0a30f5c00dd8                |
| updated                              | 2016-04-05T17:27:44Z                            |
| user_id                              | de9b797d09c64040a1ee2463f02c3e3e                |
+--------------------------------------+-------------------------------------------------+
```

Creating and Attaching a Volume

Our goal is to create a new volume based on the *cirros-0.3.4-x86_64-uec* image and then attach it to our *instance2* instance.

Since we need the **ID** of the CirrOS image, we must run `nova image-list` again:

```
root@ubuntu-openstack:~# nova image-list
+------------------------------------+---------------------------------+--------+--------+
| ID                                 | Name                            | Status | Server |
+------------------------------------+---------------------------------+--------+--------+
| ab259a33-cdd2-46-b961-227005af755f | cirros-0.3.4-x86_64-uec         | ACTIVE |        |
| 9444711e-0040-4d-9a20-b4d2d92d7eb6 | cirros-0.3.4-x86_64-uec-kernel  | ACTIVE |        |
| 4afed9-0bb2-488a-9567-e6acbc85bce1 | cirros-0.3.4-x86_64-uec-ramdisk | ACTIVE |        |
+------------------------------------+---------------------------------+--------+--------+
```

Now confirm our default availability zone:

```
root@ubuntu-openstack:~# cinder availability-zone-list
+------+-----------+
| Name |   Status  |
+------+-----------+
| nova | available |
+------+-----------+
```

We are using the *nova* availability zone.

Using `cinder` we want to create *1* volume in the *nova* zone using the *cirros ID*. This resembles the following, with the `--image-id` replaced accordingly:

```
root@ubuntu-openstack:~# cinder create 1 --display-name my-new-volume --image-id ab259a33-
cdd2-4656-b961-227005af755f --availability-zone nova
+---------------------------------------+--------------------------------------+
|                Property               |                Value                 |
+---------------------------------------+--------------------------------------+
|              attachments              |                  []                  |
|           availability_zone           |                 nova                 |
|                bootable               |                false                 |
|          consistencygroup_id          |                 None                 |
|               created_at              |      2016-04-05T17:33:36.000000      |
|              description              |                 None                 |
|               encrypted               |                False                 |
|                   id                  | 2429b44b-cc42-4ebd-847c-fde00ea96649 |
|                metadata               |                  {}                  |
|              multiattach              |                False                 |
|                  name                 |            my-new-volume             |
|      os-vol-tenant-attr:tenant_id     |   32f8a0f02393481cba2f0a30f5c00dd8   |
|   os-volume-replication:driver_data   |                 None                 |
| os-volume-replication:extended_status |                 None                 |
|          replication_status          |               disabled               |
|                 size                 |                  1                   |
|              snapshot_id             |                 None                 |
|              source_volid            |                 None                 |
|                status                |               creating               |
|                user_id               |   de9b797d09c64040a1ee2463f02c3e3e   |
|              volume_type             |              lvmdriver-1             |
+--------------------------------------+--------------------------------------+
```

If we now run `cinder list` our volume is displayed:

```
root@ubuntu-openstack:~# cinder list


+---------------------------+---------+--------+------+-------------+-------+--------------+
|            ID             |  Status |  Name  | Size | Volume Type |  Boot | Attached to |
+---------------------------+---------+--------+------+-------------+-------+--------------+
| 2429b44b-cc42-4ebd-847c   | availab | new-vo |  1   | lvmdriver-1 |  true |              |
+---------------------------+---------+--------+------+-------------+-------+--------------+
```

Before we attach this volume to our instance, we need to know the instance's **ID**. As with the image ID above, we discover this through a `list` command:

```
root@ubuntu-openstack:~# nova list
+--------------------+-----------+--------+-------+---------+------------------------------+
| ID                 | Name      | Status | State | P. State| Networks                     |
+--------------------+-----------+--------+-------+---------+------------------------------+
| 25369cf-127c-46a1  | instance2 | ACTIVE | -     | Running | private=10.0.0.3, fd04:cc38 |
+--------------------+-----------+--------+-------+---------+------------------------------+
```

To attach our volume, we use the `nova volume-attach` command, followed by our **instance ID**, our **volume ID**, then the location where we want to attach our volume. This this lab, we are attaching our volume to */dev/vdb*.

```
root@ubuntu-openstack:~# nova volume-attach 253619cf-127c-46a1-a2aa-a273cedf6a85 2429b44b-
cc42-4ebd-847c-fde00ea96649 /dev/vdb
+----------+--------------------------------------+
| Property | Value                                |
+----------+--------------------------------------+
| device   | /dev/vdb                             |
| id       | 2429b44b-cc42-4ebd-847c-fde00ea96649 |
| serverId | 253619cf-127c-46a1-a2aa-a273cedf6a85 |
| volumeId | 2429b44b-cc42-4ebd-847c-fde00ea96649 |
+----------+--------------------------------------+
```

To confirm that the volume is attached run `cinder show 2429b44b-cc42-4ebd-847c-fde00ea96649`, replacing the ID with the ID for your own attached volume. The **Status** of the volume should be listed as *in-use*.

To confirm that the volume is attached, we can SSH into our new instance. First, retreived the private IP of the instance, located in the **Networks** section of the output:

```
root@ubuntu-openstack:~# nova list
+-------------+----------+-------+-----------+------------+-------------------------+
| ID          | Name     | Status | Task State | Power State | Networks               |
+-------------+----------+-------+-----------+------------+-------------------------+
| 253619cf-12 | instance2 | ACTIVE | -          | Running     | private=10.0.0.3, fd04:c |
+-------------+----------+-------+-----------+------------+-------------------------+
```

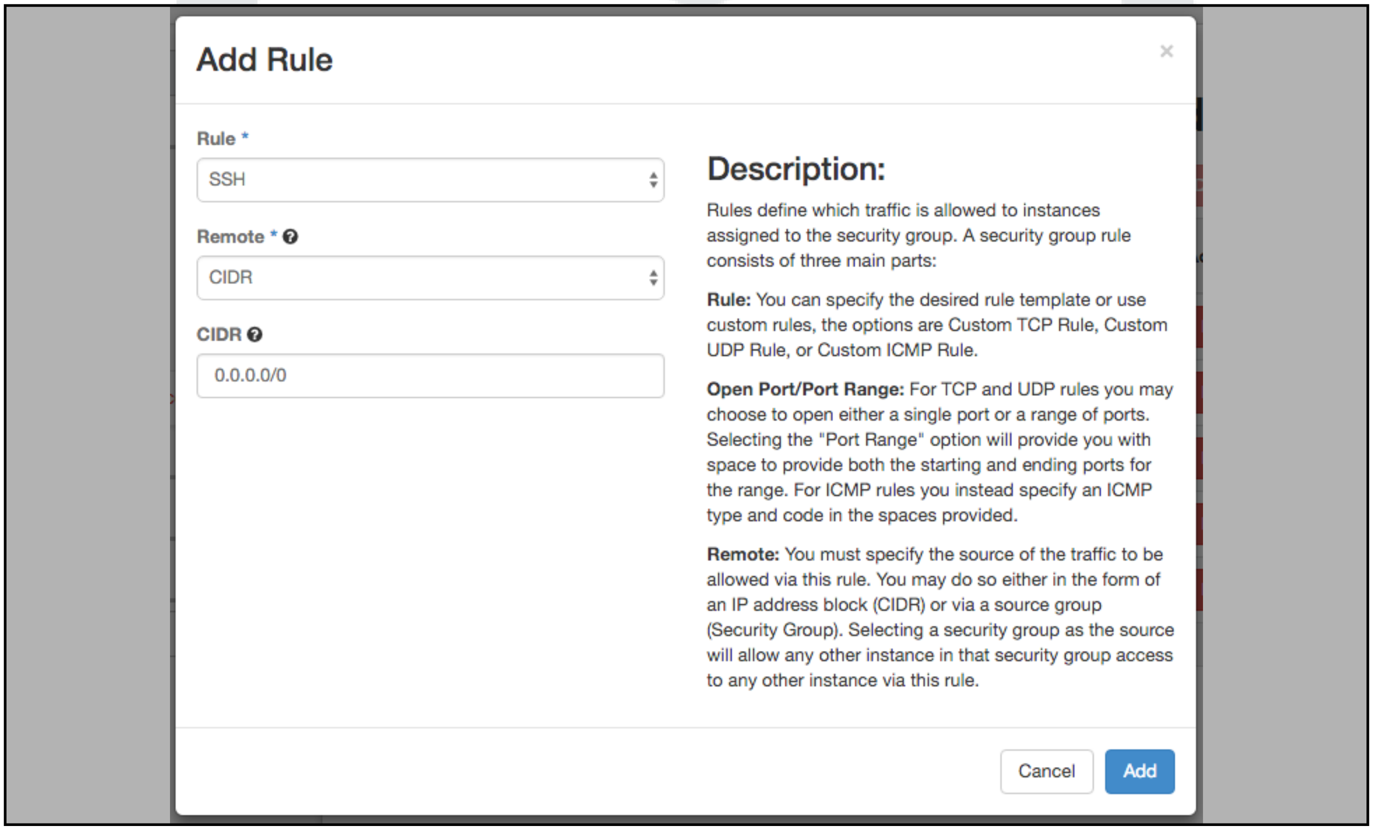
In this example, the IP is *10.0.0.3*.

The user for our CirrOS instance is *cirros*; this is what we use to log in:

```
root@ubuntu-openstack:~# ssh -i KEY.pem cirros@10.0.0.3
```

However, because we do not have port 22 open for our server, we are unable to log in. Return to the **Horizon Dashboard** to change permissions to allow SSH connection.

Under **Access & Security**, ensure the **Security Groups** tab is selected. Press **Manage Rules**, then **Add Rule**.



Rule * : SSH

Remote * : CIDR

CIDR: 0.0.0.0/0

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

For the **Rule** type, select *SSH*. You can leave the CIDR block range as-is. It allows connections from any location.

Return to your terminal and re-run the ssh command:

```
root@ubuntu-openstack:~# ssh -i KEY.pem cirros@10.0.0.3
```

You are taken to a prompt. Type mount to see the attached volumes, then use fdisk to list available disks:

```
$ mount
rootfs on / type rootfs (rw)
/dev on /dev type devtmpfs (rw,relatime,size=248056k,nr_inodes=62014,mode=755)
/dev/vda on / type ext3 (rw,relatime,errors=continue,user_attr,acl,barrier=1,data=ordered)
/proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /dev/shm type tmpfs (rw,relatime,mode=777)
tmpfs on /run type tmpfs (rw,nosuid,relatime,size=200k,mode=755)
$ sudo fdisk -l

Disk /dev/vda: 1073 MB, 1073741824 bytes


16 heads, 63 sectors/track, 2080 cylinders, total 2097152 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: 0x00000000

Disk /dev/vda doesn't contain a valid partition table

Disk /dev/vdb: 1073 MB, 1073741824 bytes
16 heads, 63 sectors/track, 2080 cylinders, total 2097152 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: 0x00000000

Disk /dev/vdb doesn't contain a valid partition table
```

We are interested in working with the new */dev/vdb* disk. Create a partition table:

```
$ sudo fdisk /dev/vdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xfe6a63c2.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): w
```

Press w to write.

Mount the disk to the */mnt* directory:

```
$ sudo mount /dev/vdb /mnt
```

To unmount, you can use:

```
$ sudo umount /mnt
```