Linux Academy.com

Hands On Labs

Creating an Instance From the CLI

Table of Contents

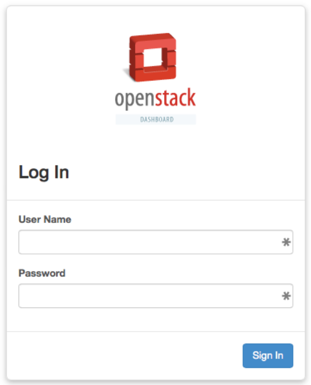
Introduction 1
Getting your OpenStack RC File 1
Using the Nova CLI client 5
Available Images and Flavors 5
Creating an SSH key pair for OpenStack Images 6
Add a rule to the default security group for SSH access 7
Booting an instance with the nova CLI 7
How to connect to an instance with SSH 9
How to delete an instance with the Nova CLI 10

Introduction

In this Lab, we are going to learn to use the nova compute CLI in order to launch an instance instead of from the web Dashboard. We are first going to connect to our instance in the Dashboard in order to learn how to download from any OpenStack environment what is called the OpenStack RC file. An OpenStack RC file is used to set environment variables in our Linux terminal session in order to set project-specific credentials for the OpenStack CLI services such as Compute, Image, and Identity services.

Getting your OpenStack RC File

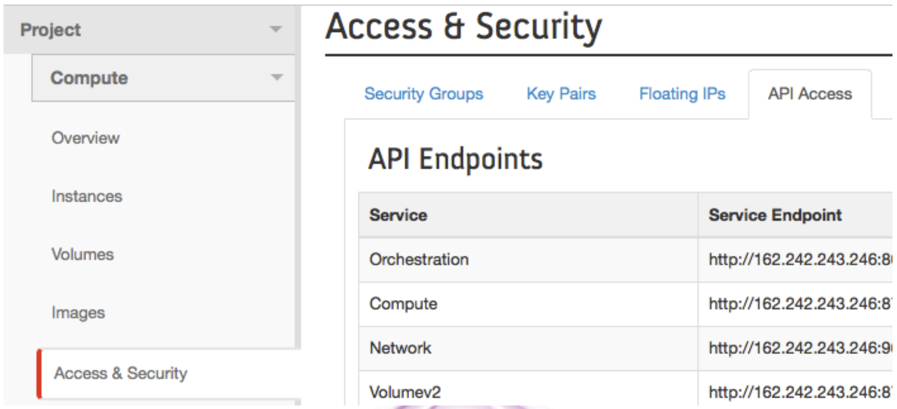
After you start the lab, you can click on the Horizon link on the Live! Lab page and you will be given a page like follows:



NOTE

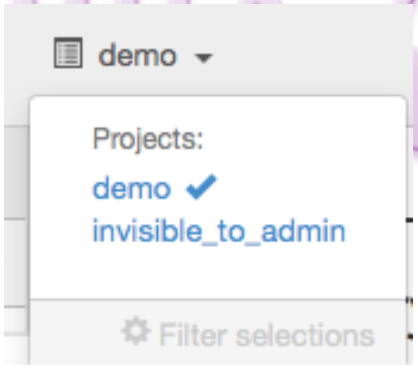
In this Lab, we are going to login to OpenStack with the DEMO tenant. You can login using the username **demo** and the password **openstack**.

Step 1: Navigate to Project → Compute → Access & Security



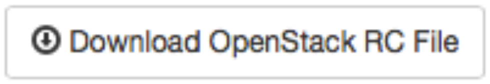
Service	Service Endpoint
Orchestration	http://162.242.243.246:8
Compute	http://162.242.243.246:8
Network	http://162.242.243.246:9
Volumev2	http://162.242.243.246:8

Be sure that at the top of Horizon you have selected from the drop down **demo** and not **invisible_to_admin.**



Step 2: Then select the API Access tab like shown above.

Step 3: Now click on 'Download OpenStack RC File'



Step 4: Now navigate to wherever you downloaded the RC file; the filename should be demo-openrc.sh

Open the file in a text editor of your choice.

The file will look similar to the following with the exception of your Tenant ID, which will be different each time you start a lab. I have highlighted the specific lines of the script, which set environment variables for tenant AUTH API URL, ID and Tenant username etc.

```
##begin##

#!/bin/bash

# To use an Openstack cloud you need to authenticate against keystone, which

# returns a **Token** and **Service Catalog**.  The catalog contains the

# endpoint for all services the user/tenant has access to - including nova,

# glance, keystone, swift.

#

# *NOTE*: Using the 2.0 *auth api* does not mean that compute api is 2.0.  We

# will use the 1.1 *compute api*

export OS_AUTH_URL=http://162.242.243.246:5000/v2.0


# With the addition of Keystone we have standardized on the term **tenant**

# as the entity that owns the resources.

export OS_TENANT_ID=69870678594c4fd2a8a36a1a5123bbae

export OS_TENANT_NAME="demo"


# In addition to the owning entity (tenant), openstack stores the entity

# performing the action as the **user**.

export OS_USERNAME="demo"


# With Keystone you pass the keystone password.

echo "Please enter your OpenStack Password: "

read -sr OS_PASSWORD_INPUT

export OS_PASSWORD=$OS_PASSWORD_INPUT


# If your configuration has multiple regions, we set that information here.

# OS_REGION_NAME is optional and only valid in certain environments.

export OS_REGION_NAME="RegionOne"

# Don't leave a blank variable, unset it if it was empty

if [ -z "$OS_REGION_NAME" ]; then unset OS_REGION_NAME; fi
```

## END ###

Step 5: SSH to your server as root using the generated password you received in the Live Lab! console:

```
$ ssh root@YOURSSHIP
```

Example: ***Note be sure to use the IP for YOUR lab***

```
$ ssh root@162.242.243.246
The authenticity of host '162.242.243.246 (162.242.243.246)' can't be established.
RSA key fingerprint is 3e:b6:c3:6a:db:ca:bc:8d:9e:db:d9:32:3f:0d:8f:ac.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '162.242.243.246' (RSA) to the list of known hosts.
root@162.242.243.246's password:
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-53-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
root@openstack29062015-155406:~#
```

Step 6: Now let's use the vim editor to create our environment variable RC file:

```
# vim demorc.sh
```

Then insert the exact script that you downloaded and opened in an editor on your local host.

Save and close this file.

Now you can source this environment variable and source it work with the OpenStack Nova CLI. In fact, you now know how to quickly create an RC file for this on any OpenStack environment that you have access credentials to.

Step 7: Let's source our file now:

```
# source demorc.sh
```

Please enter your OpenStack Password:

When asked above, enter the password for **demo** which is just **openstack**

Step 8: We can verify that the environments variables are active with the following command:

```
# export | grep OS
```

Example: I have highlighted the results that are in the script for the environment variables that we just sourced.

```
root@openstack29062015-155406:~# export | grep OS
declare -x LESSCLOSE="/usr/bin/lesspipe %s %s"
declare -x OS_AUTH_URL="http://162.242.243.246:5000/v2.0"
declare -x OS_PASSWORD="openstack"
declare -x OS_REGION_NAME="RegionOne"
declare -x OS_TENANT_ID="69870678594c4fd2a8a36a1a5123bbae"
declare -x OS_TENANT_NAME="demo"
declare -x OS_USERNAME="demo"
root@openstack29062015-155406:~#
```

Using the Nova CLI client

The Nova client CLI offer a vast amount of help. You can get help by simply typing:

```
# nova help
```

Furthermore, we can get subcommand help by selecting a subcommand like boot such as:

```
# nova help boot
```

This gives us the break down of options that we need when booting an instance on the CLI.

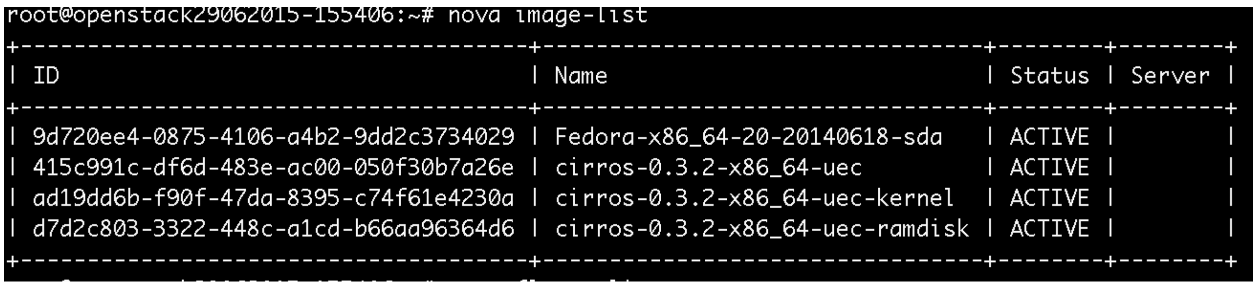
Feel free to take a look at some of the options now.

Available Images and Flavors

Much like we did in the Horizon Dashboard, we need to see what images we have in order to boot an instance from an available image.

Step 1: Use the nova image-list command to see what images you have available:

```
# nova image-list
```

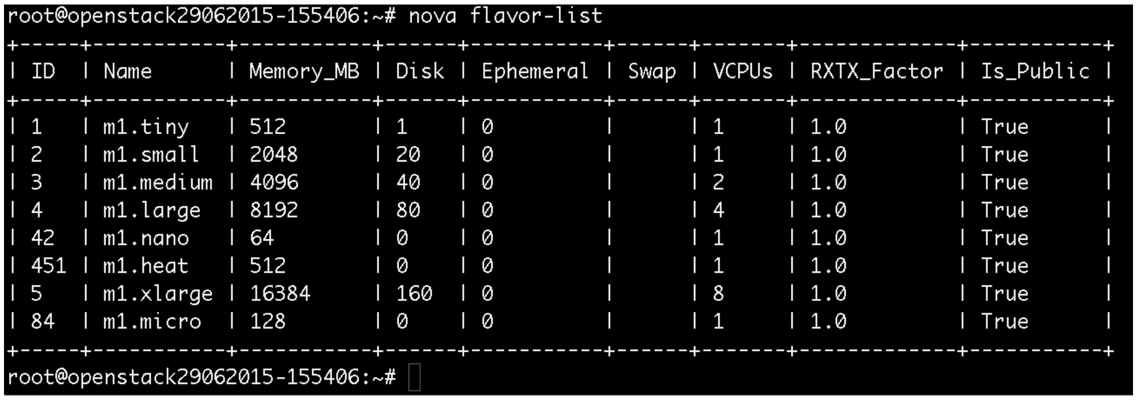


```
root@openstack29062015-155406:~# nova image-list
+--------------------------------------+---------------------------------+--------+--------+
| ID                                   | Name                            | Status | Server |
+--------------------------------------+---------------------------------+--------+--------+
| 9d720ee4-0875-4106-a4b2-9dd2c3734029 | Fedora-x86_64-20-20140618-sda   | ACTIVE |        |
| 415c991c-df6d-483e-ac00-050f30b7a26e | cirros-0.3.2-x86_64-uec         | ACTIVE |        |
| ad19dd6b-f90f-47da-8395-c74f61e4230a | cirros-0.3.2-x86_64-uec-kernel  | ACTIVE |        |
| d7d2c803-3322-448c-a1cd-b66aa96364d6 | cirros-0.3.2-x86_64-uec-ramdisk | ACTIVE |        |
+--------------------------------------+---------------------------------+--------+--------+
```

The Image that we are interested in is the cirros uec image.

Step 2: Now we need to see what flavors we have available to boot an image with:

nova flavor-list



```
root@openstack29062015-155406:~# 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      |
| 42  | m1.nano   | 64        | 0    | 0         |      | 1     | 1.0         | True      |
| 451 | m1.heat   | 512       | 0    | 0         |      | 1     | 1.0         | True      |
| 5   | m1.xlarge | 16384     | 160  | 0         |      | 8     | 1.0         | True      |
| 84  | m1.micro  | 128       | 0    | 0         |      | 1     | 1.0         | True      |
+-----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
root@openstack29062015-155406:~# 
```

Creating an SSH key pair for OpenStack Images

In the Horizon Lab, we used VNC to access our Virtual Machine Instance from the console. When working within the CLI, clients cannot use VNC. We need to manage our instances using SSH. To do this, we need an SSH key pair that will inject into the instance when it is launched.

Step 1: We need to create an SSH key pair

# ssh-keygen -q -f ssh-key

When prompted to enter a password just hit enter to not set one.

```
# ssh-keygen -q -f ssh-key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
root@openstack29062015-155406:~#
```

After key creation in your working directory, you should see the following two files:

ls

ssh-key ssh-key.pub

Step 2: Now we have to actually add the keys that we created (the public key to be precise) in to OpenStack

bellow the *novakey* is the name we are giving the keypair and the *ssh-key.pub* is the path to the key we generated

# nova keypair-add novakey --pub-key ssh-key.pub

Step 3: Now verify that OpenStack has accepted and shows your new key pair:

```
# nova keypair-list
```

```
root@openstack29062015-155406:~# nova keypair-list
+---------+-------------------------------------------------+
| Name    | Fingerprint                                     |
+---------+-------------------------------------------------+
| novakey | 06:45:20:fe:8c:78:74:da:7b:b9:b9:a5:f8:59:c9:b9 |
+---------+-------------------------------------------------+
```

Add a rule to the default security group for SSH access

Furthermore we did not need any security group rules changed in the Horizon lesson because we were not going to access our instance from an SSH terminal. In this lesson, we need to add to the default security group port 22 for SSH connections.

We need to port 22 to the CIDR range of the Public subnet address range which is 172.24.4.0/24

Step 1:

```
$ nova secgroup-add-rule default tcp 22 22 172.24.4.0/24
```

```
root@openstack29062015-155406:~# nova secgroup-add-rule default tcp 22 22 172.24.4.0/24
+-------------+-----------+---------+---------------+--------------+
| IP Protocol | From Port | To Port | IP Range      | Source Group |
+-------------+-----------+---------+---------------+--------------+
| tcp         | 22        | 22      | 172.24.4.0/24 |              |
+-------------+-----------+---------+---------------+--------------+
```

Booting an instance with the nova CLI

Using the nova commands earlier (you may need to scroll up to see the outputs again), we are going to boot an image cirros-0.3.2-x86_64-uec and the m1.nano flavor. We are also going to inject the novakey key that we created and added to OpenStack earlier on in the Lab.

Step 1:

```
$ nova boot instance2 --image cirros-0.3.2-x86_64-uec --flavor m1.tiny --key-name novakey
```

```
+--------------------------------------+----------------------------------------------------------------+
| Property                             | Value                                                          |
+--------------------------------------+----------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                         |
| OS-EXT-AZ:availability_zone          | nova                                                           |
| 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                            | En6W648vEY2d                                                   |
| config_drive                         |                                                                |
| created                              | 2015-06-30T04:23:53Z                                           |
| flavor                               | m1.tiny (1)                                                    |
| hostId                               |                                                                |
| id                                   | 8d30f5a3-0c69-42d9-8e68-5a7ba9bb3dcb                           |
| image                                | cirros-0.3.2-x86_64-uec (415c991c-df6d-483e-ac00-050f30b7a26e) |
| key_name                             | novakey                                                        |
| metadata                             | {}                                                             |
| name                                 | instance2                                                      |
| os-extended-volumes:volumes_attached | []                                                             |
| progress                             | 0                                                              |
| security_groups                      | default                                                        |
| status                               | BUILD                                                          |
| tenant_id                            | 69870678594c4fd2a8a36a1a5123bbae                               |
| updated                              | 2015-06-30T04:23:53Z                                           |
| user_id                              | 704822425ba64e57814447afa111d2c1                               |
+--------------------------------------+----------------------------------------------------------------+
root@openstack29062015-155406:~#
```

Step 2: every few moments we can take a look at the build status by using the nova list command to see our instances. Once the Status changes from BUILD to ACTIVE then your machine is running.

```
$ nova list
```

```
root@openstack29062015-155406:~# nova list
+--------------------------------------+-----------+--------+------------+-------------+----------+
| ID                                   | Name      | Status | Task State | Power State | Networks |
+--------------------------------------+-----------+--------+------------+-------------+----------+
| c77a21b5-ab99-4f09-bcbb-4e47673cdff5 | instance2 | BUILD  | spawning   | NOSTATE     |          |
+--------------------------------------+-----------+--------+------------+-------------+----------+
root@openstack29062015-155406:~# nova list
+--------------------------------------+-----------+--------+------------+-------------+------------------+
| ID                                   | Name      | Status | Task State | Power State | Networks         |
+--------------------------------------+-----------+--------+------------+-------------+------------------+
| c77a21b5-ab99-4f09-bcbb-4e47673cdff5 | instance2 | ACTIVE | -          | Running     | private=10.0.0.4 |
+--------------------------------------+-----------+--------+------------+-------------+------------------+
```

How to connect to an instance with SSH

Now while still being connected via SSH to our OpenStack node server, we can now SSH further in to our instance we just created. We are going to use the ssh-key flag to include the private portion of our SSH key and then using the private IP address of our Virtual machine. Your Private IP of the virtual machine you just created is shown when you use the nova list command, which will show under the Networks print out. Again the user for the cirros image is simply cirros as shown below.

Step 1:

```
$ ssh -i ssh-key cirros@10.0.0.4
```

```
root@openstack29062015-155406:~# ssh -i ssh-key cirros@10.0.0.4
The authenticity of host '10.0.0.4 (10.0.0.4)' can't be established.
RSA key fingerprint is 9f:11:0e:03:c1:36:46:7d:a0:19:70:f8:79:2a:76:c8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.4' (RSA) to the list of known hosts.
$
```

Once we see the prompt shown above, we have successfully connected to our instance via SSH.

Furthermore, we can even see that we are on our instance 2 host by using the hostname command.

Step 2:

```
$ hostname
instance2
$
```

Step 3: You can now close the SSH connection to the instance2 you created.

```
$ exit
```

```
$ exit
Connection to 10.0.0.4 closed.
root@openstack29062015-155406:~#
```

How to delete an instance with the Nova CLI

You can delete an instance using nova delete

(nova delete NAMEOFINSTANCE)

Step 1:

```
 root@openstack29062015-155406:~# nova delete instance2
Request to delete server instance2 has been accepted.
root@openstack29062015-155406:~#
```

Step 2: Now verify that the instance has been deleted with the nova list command

```
root@openstack29062015-155406:~# nova list
+----+------+--------+------------+-------------+----------+
| ID | Name | Status | Task State | Power State | Networks |
+----+------+--------+------------+-------------+----------+
+----+------+--------+------------+-------------+----------+
root@openstack29062015-155406:~#
```

If your output looks like the above and no instances are running, you have successfully deleted your instance.

Congratulations you have now created an instance using the nova CLI client and deleted it!