DevStack Basic Administration

Configuration and Management of OpenStack Horizon with DevStack

Lab 1: DevStack Installation

Purpose

In this lab you will deploy DevStack in order to be able to perform the relevant OpenStack related portions of this lab guide.

DevStack is NOT PERSISTENT!!! If you restart the VM running DevStack, all configuration is lost!!!

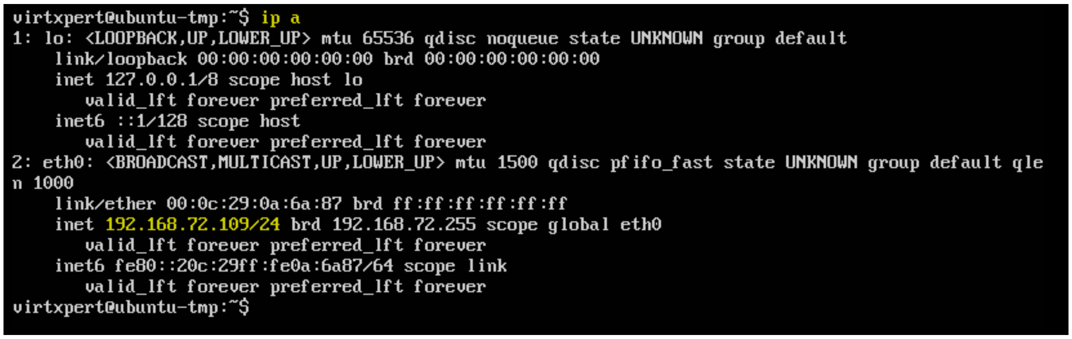
You must have access to an Ubuntu 14.04 virtual machine to perform these labs with at least 4GB RAM and 2vCPU

Lab 1 Exercise 1: Install DevStack

1. Power on an Ubuntu 14.04 virtual machine and log in as a user with sudo access. By default, the user created during the installation of the OS has sudo access.

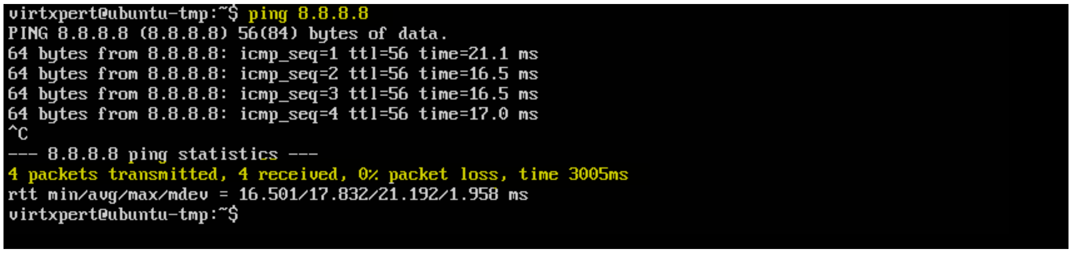
2. Once logged in, verify that the virtual machine has a valid IP address on your network and has access to the internet:

   ```
   ip a
   ```



```
virtxpert@ubuntu-tmp:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qle
n 1000
    link/ether 00:0c:29:0a:6a:87 brd ff:ff:ff:ff:ff:ff
    inet 192.168.72.109/24 brd 192.168.72.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe0a:6a87/64 scope link
       valid_lft forever preferred_lft forever
virtxpert@ubuntu-tmp:~$
```

   ```
   ping 8.8.8.8
   ```



```
virtxpert@ubuntu-tmp:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=21.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=16.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=16.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=56 time=17.0 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 16.501/17.832/21.192/1.958 ms
virtxpert@ubuntu-tmp:~$
```

3. SSH to the IP address of virtual machine using PuTTY or other SSH client, accept the key and minimize the virtual machine console (VMware Workstaion/Fusion, VirtualBox, etc) as this is no longer needed.

4. Install git:

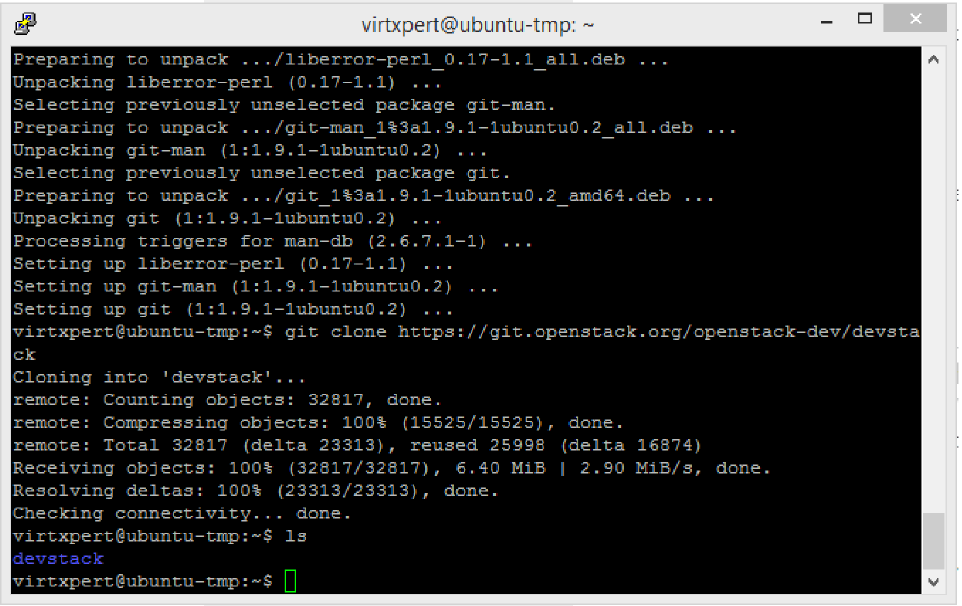
```
sudo apt-get update && sudo apt-get install git
```

Enter the users password when prompted, type Y and then press enter

5. Clone DevStack from GitHub:

```
git clone https://git.openstack.org/openstack-
dev/devstack
```

6. You should now have a devstack directory



virtxpert@ubuntu-tmp: ~

```
Preparing to unpack .../liberror-perl_0.17-1.1_all.deb ...
Unpacking liberror-perl (0.17-1.1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a1.9.1-1ubuntu0.2_all.deb ...
Unpacking git-man (1:1.9.1-1ubuntu0.2) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a1.9.1-1ubuntu0.2_amd64.deb ...
Unpacking git (1:1.9.1-1ubuntu0.2) ...
Processing triggers for man-db (2.6.7.1-1) ...
Setting up liberror-perl (0.17-1.1) ...
Setting up git-man (1:1.9.1-1ubuntu0.2) ...
Setting up git (1:1.9.1-1ubuntu0.2) ...
virtxpert@ubuntu-tmp:~$ git clone https://git.openstack.org/openstack-dev/devsta
ck
Cloning into 'devstack'...
remote: Counting objects: 32817, done.
remote: Compressing objects: 100% (15525/15525), done.
remote: Total 32817 (delta 23313), reused 25998 (delta 16874)
Receiving objects: 100% (32817/32817), 6.40 MiB | 2.90 MiB/s, done.
Resolving deltas: 100% (23313/23313), done.
Checking connectivity... done.
virtxpert@ubuntu-tmp:~$ ls
devstack
virtxpert@ubuntu-tmp:~$ 
```

7. Create a user to run devstack

```
devstack/tools/create-stack-user.sh; su stack
```

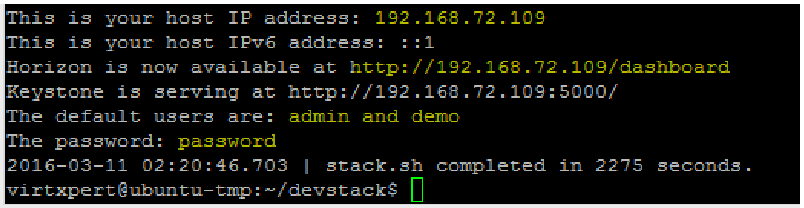
8. Install DevStack

```
cd devstack; ./stack.sh
```

9. When prompted, type a password and press the enter key. password will be used throughout this guide.

10. The installation will take several minutes depending on the resources assigned and available to the virtual machine. (on a physical 8-core server with 32GB RAM, and a 2vCPU/4GB VM the installation took ~35 minutes)

11. Once the installation completes, make note of the information provided



```
This is your host IP address: 192.168.72.109
This is your host IPv6 address: ::1
Horizon is now available at http://192.168.72.109/dashboard
Keystone is serving at http://192.168.72.109:5000/
The default users are: admin and demo
The password: password
2016-03-11 02:20:46.703 | stack.sh completed in 2275 seconds.
virtxpert@ubuntu-tmp:~/devstack$ 
```

12. Shutdown DevStack

```
./unstack.sh
```

13. Create local.conf to enable Neutron

```
vi local.conf
```

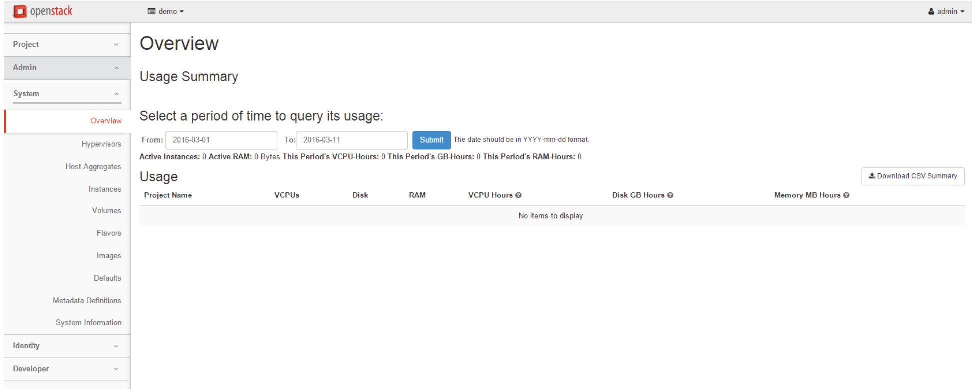
14. Add the following lines

```
[[local|localrc]]
disable_service n-net
enable_service q-svc
enable_service q-agt
enable_service q-dhcp
enable_service q-l3
enable_service q-meta
enable_service neutron
# Optional, to enable tempest configuration as part of
devstack
enable_service tempest
```

15. Restart DevStack

```
./stack.sh
```

16. Open a browser and navigate to the Horizon URL. Log in as admin with the password you set during installation, for example password. You are now logged into the OpenStack Horizon UI



Overview

Usage Summary

Select a period of time to query its usage:

From: 2016-03-01 To: 2016-03-11 The date should be in YYYY-mm-dd format.

Active Instances: 0 Active RAM: 0 Bytes This Period's VCPU-Hours: 0 This Period's GB-Hours: 0 This Period's RAM-Hours: 0

Usage

Project Name	VCPUs	Disk	RAM	VCPU Hours	Disk GB Hours	Memory MB Hours
No items to display.						

If for any reason you need to reboot the VM after the installation run the following to restart DevStack (may take up to 10 min):

```
./devstack/stack.sh
```

Lab 2: Configure DevStack

Purpose

In this lab you will explore and configure your DevStack environment to prepare for launching instances.

Lab 2 Exercise 1: Explore Projects in DevStack

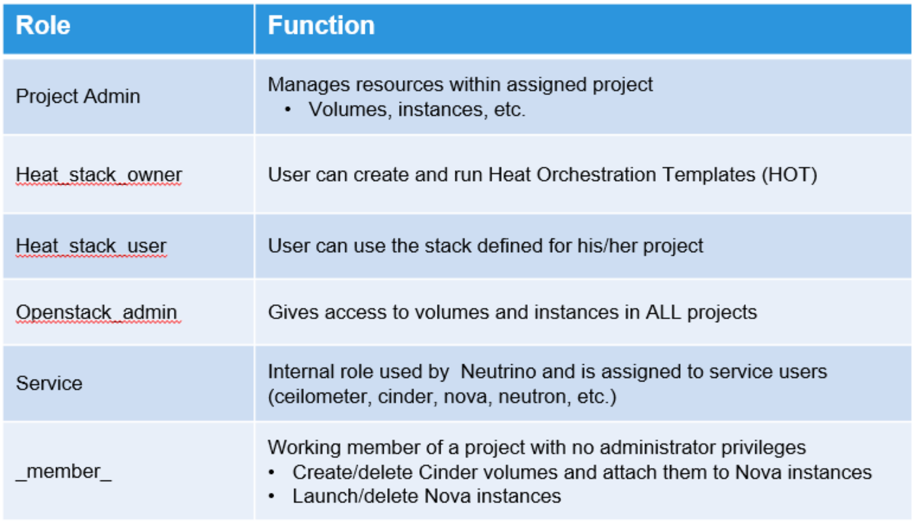
1. Log into the Horizon dashboard as admin / password
2. Click **System >> Defaults**

   Q: What are the items listed here used for?

   A: These are the default quotas used when you create projects.
3. Navigate to **Identity >> Projects**
4. In the dropdown next to **demo,** choose **Modify Quotas**
5. Change **VCPU to 10**
6. Change **Instances to 3**
7. Change **Volumes to 3**
8. Change **Floating IPs to 5**
9. Click **Save.**
10. Log out as admin and log back in as demo / password
11. Navigate to **Identity >> Projects**

    Q: Are you able to modify the quotas? Why or why not?

    A: the demo user does not have the appropriate OpenStack role



Role	Function
Project Admin	Manages resources within assigned project • Volumes, instances, etc.
Heat_stack_owner	User can create and run Heat Orchestration Templates (HOT)
Heat_stack_user	User can use the stack defined for his/her project
Openstack_admin	Gives access to volumes and instances in ALL projects
Service	Internal role used by Neutrino and is assigned to service users (ceilometer, cinder, nova, neutron, etc.)
member	Working member of a project with no administrator privileges • Create/delete Cinder volumes and attach them to Nova instances • Launch/delete Nova instances

Lab 2 Exercise 2: Create a new Project in DevStack

1. Log into the Horizon dashboard as admin / password (sign out as demo from the previous lab)
2. Click **Identity >> Users**
3. Click **Create User,** enter the following, and click **Create User**:

Parameter	Value
Domain ID	Default
Domain Name	Default
User Name	qa-admin
Password / Confirm Password	password
Enabled	checked

4. Click **Identity >> Projects**
5. Click **Create Project** and enter the following:

Parameter	Value
Domain ID	Default
Domain Name	Default
Name	QA-demo
Enabled	Checked

6. Click on the **Project Members** tab, click the + sign next to qa-admin
7. Click the **Member** pull down next to qa-admin and add the role admin
8. Click the **Quota tab**
9. Change **VCPU to 10**
10. Change **Instances to 3**
11. Change **Volumes to 3**
12. Change **Floating IPs to 5**
13. Click **Create Project**
14. Click On **Users**
15. Click **Create User,** enter the following, and click **Create User**:

Parameter	Value
Domain ID	Default
Domain Name	Default

User Name	qa-user
Password / Confirm Password	password
Primary Project	QA-demo
Role	Member
Enabled	Checked

16. While assigning access to individual users is possible, this would become an administrative overhead. Click on **Groups**
17. Click **Create Group,** name the group **QA-users** and click **Create Group**:
18. Click **Projects**, for the QA-demo project click **Manage Members**
19. Click the **Project Groups** tab, add QA-users, and click save.
20. Now, any user added to the group QA-users will have the member role in the project. The same theory is possible when using LDAP, LDAP users can be added to a group and the group assigned membership to a project or an Account.

Lab 2 Exercise 3: Create an image

1. Log into the Horizon dashboard as qa-admin / password (sign out as admin from the previous lab)

2. Open a new tab and navigate to http://cloud.centos.org/centos/7/images/ and download CentOS-7-x86_64-GenericCloud-1601.qcow2 to c:\tmp

3. Navigate to **Project >> Images**.

   Q: What OpenStack service provides images?

   A: Glance

4. Click **+Create Image**.

5. Enter the following information.

Build Information Tab	
Parameter	**Value**
Name	linux-qa
Description	Development Linux Image
Image Source	Image File
Image File	C:\tmp\CentOS-7-x86_64-GenericCloud-1601.qcow2
Format	QCOW2
Architecture	x86_64
Minimum Disk	Leave Blank
Minimum RAM	Leave Blank
Public	Unchecked
Protected	Unchecked

6. Click **Create Image.**

Lab 2 Exercise 4: Create private networks

1. Log into the Horizon dashboard as qa-admin / password (if you were signed out after the previous lab)

2. Navigate to **Project >> Network >> Networks**

3. Click +**Create Network** button

4. Enter the following information:

Network Tab	
Parameter	Value
Network_Name	qa_web_internal
Admin State	UP

Subnet Tab	
Parameter	Value
Create Subnet	Checked
Subnet Name	qa_web_internal
Network Adress	172.16.1.0/24
Gateway IP	172.16.1.1
Disable Gateway	Unchecked

Subnet Detail Tab	
Parameter	Value
Enable DHCP	Checked
Allocation pools	172.16.1.10,172.16.1.50
DNS Name Server	Blank
Host Routes	Blank

5. Click **Create**

6. Click **+Create Network** button

7. Enter the following information:

Network Tab	
Parameter	Value
Network_Name	qa_app_internal
Admin State	UP

Subnet Tab	
Parameter	Value
Create Subnet	Checked
Subnet Name	qa_app_internal

Network Adress	172.16.2.0/24
Gateway IP	172.16.2.1
Disable Gateway	Unchecked

Subnet Detail Tab	
Parameter	Value
Enable DHCP	Checked
Allocation pools	172.16.2.10,172.16.2.50
DNS Name Server	Blank
Host Routes	Blank

8. Click **Create**
9. Click **+Create Network** button
10. Enter the following information:

Network Tab	
Parameter	Value
Network_Name	qa_db_internal
Admin State	UP

Subnet Tab	
Parameter	Value
Create Subnet	Checked
Subnet Name	qa_db_internal
Network Adress	172.16.3.0/24
Gateway IP	172.16.3.1
Disable Gateway	Unchecked

Subnet Detail Tab	
Parameter	Value
Enable DHCP	Checked
Allocation pools	172.16.3.10,172.16.3.50
DNS Name Server	Blank
Host Routes	Blank

11. Click **Create**

Lab 2 Exercise 5: Create a router

1. Log into the Horizon dashboard as qa-admin / password (if you were signed out after the previous lab)
2. Navigate to **Project >> Network >> Routers**
3. Click +**Create Router**
4. Enter the following information:

Router Tab	
Parameter	Value
Router Name	qa_router01
Admin State	UP
External Network	public

5. Click **Create Router**
6. Click on the name of the router, **qa_router01**
7. Click the **Interfaces** tab
8. Click **+Add Interface** button
9. Enter the following information:

Add Interface	
Parameter	Value
Subnet	qa_web_internal
IP Address (optional)	Leave blank
Router Name	Leave default
Router ID	Leave default

10. Click **Add Interface**
11. Click **+Add Interface** button
12. Enter the following information:

Add Interface	
Parameter	Value
Subnet	qa_app_internal
IP Address	Leave blank
Router Name	Leave default
Router ID	Leave default

13. Click **Add Interface**
14. Click **+Add Interface** button
15. Enter the following information:

Add Interface	
Parameter	**Value**
Subnet	qa_db_internal
IP Address	Leave blank
Router Name	Leave default
Router ID	Leave default

Q: What are the IPs of the new interfaces?

A: 172.16.1.1, 172.16.2.1, 172.16.3.1

Q: Where did it get the IP address from?

A: The default gatweay address configured when creating the network

Lab 2 Exercise 6: Create private networks

1. Log into the Horizon dashboard as qa-admin / password (if you were signed out after the previous lab)

2. Navigate to **Project >> Compute >> Access & Security.**

3. Click the **Security Groups** tab if it is not selected

   Q: What is the purpose of Security Groups?

   A: Filter traffic to and from instances

4. Click **+Create Security Group**

Parameter	Value
Name	Linux Web Servers
Description	Rules for public facing Web Servers

5. Click **+Create Security Group**

Parameter	Value
Name	Linux App Servers
Description	Rules for App Servers

6. Click **+Create Security Group**

Parameter	Value
Name	Linux DB Servers
Description	Rules for DB Servers

7. Click **Manage Rules** next to **Linux Web Servers**

8. Click **+Add Rule**

Add Rule Tab	
Parameter	**Value**
Rule	HTTP
Remote	CIDR
CIDR	0.0.0.0/0

9. Click **Add**

10. Click **+Add Rule**

Add Rule Tab	
Parameter	**Value**
Rule	HTTPS
Remote	CIDR

CIDR	0.0.0.0/0

11. Click **Add**

12. Click **+Add Rule**

Add Rule Tab	
Parameter	**Value**
Rule	SSH
Remote	CIDR
CIDR	0.0.0.0/0

13. Click **Add**

14. Click **+Add Rule**

Add Rule Tab	
Parameter	**Value**
Rule	SSH
Remote	Security Group
Security Group	Linux App Servers
Ether Type	IPV4

15. Click **Add**

16. Click **+Add Rule**

Add Rule Tab	
Parameter	**Value**
Rule	All ICMP
Direction	Ingress
Remote	CIDR
CIDR	0.0.0.0/0

17. Click **Add**

18. Click **Access & Security**

19. Click **Manage Rules** next to **Linux DB Servers**

20. Click **+Add Rule**

Add Rule Tab	
Parameter	**Value**
Rule	All ICMP
Direction	Ingress
Remote	CIDR
CIDR	0.0.0.0/0

21. Click **Add**
22. Click **+Add Rule**

Add Rule Tab	
Parameter	**Value**
Rule	All TCP
Direction	Ingress
Remote	Security Group
Security Group	Linux App Servers
Ether Type	IPV4

23. Click **Add**
24. Click **+Add Rule**

Add Rule Tab	
Parameter	**Value**
Rule	All TCP
Direction	Egress
Remote	Security Group
Security Group	Linux App Servers
Ether Type	IPV4

25. Click **Add**
26. Click **Access & Security**
27. Click **Manage Rules** next to **Linux App Servers**
28. Click **+Add Rule**

Add Rule Tab	
Parameter	**Value**
Rule	SSH
Remote	Security Group
Security Group	Linux Web Servers
Ether Type	IPV4

29. Click **Add**
30. Click **+Add Rule**

Add Rule Tab	
Parameter	**Value**
Rule	All TCP
Direction	Ingress

Remote	Security Group
Security Group	Linux DB Servers
Ether Type	IPV4

31. Click **Add**

32. Click **+Add Rule**

Add Rule Tab	
Parameter	Value
Rule	All TCP
Direction	Egress
Remote	Security Group
Security Group	Linux DB Servers
Ether Type	IPV4

33. Click **Add**

34. Click **+Add Rule**

Add Rule Tab	
Parameter	Value
Rule	All ICMP
Direction	Ingress
Remote	CIDR
CIDR	0.0.0.0/0

35. Click **Add**

36. Click **Access & Security**

37. Click the **Key Pairs** tab

38. Click **+Create Key Pair**

Create Key Pair Tab	
Parameter	Value
Key Pair Name	qakeypair

39. When prompted save the file, **qakeypair.pem** and note the location

Lab 2 Exercise 7: Allocate floating IPs

1. Log into the Horizon dashboard as qa-admin / password (if you were signed out after the previous lab)
2. Navigate to **Project >> Compute >> Access & Security.**
3. Click the **Floating IPs** tab
4. Click **Allocate IP To Project**
5. Choose **public** for the **Pool**
6. Click **Allocate IP**
7. Repeat steps 5-7 until you have 5 IPs allocated.
8. You now have Floating IPs available to assign to routers and instances for the project

Lab 2 Exercise 8: Launch Instances

1. Log into the Horizon dashboard as qa-admin / password (if you were signed out after the previous lab)
2. Navigate to ***Project >> Compute >> Overview*** what is the quota allowed for the number of instances?
3. Navigate to **Project >> Compute >> Instances**
4. Click **Launch Instance**
5. Follow the wizard and select the following information:

Details Tab	
Parameter	**Value**
Availability Zone	nova
Instance Name	web01
Flavor	M1.micro
Instance Count	1
Instance Boot Source	Boot from image
Image Name	cirros-0.3.4-x86_64-uec
Key Pair	devkeypair
Security groups	Linux Web Servers
Selected Networks	qa_web_internal
Customization Script	#cloud-config chpasswd: list: | root:P@ssw0rd centos:P@ssw0rd expire: False
Disk Partition	Automatic
Configuration Drive	Unchecked

6. Click **Launch**
7. Navigate to **Project >> Compute >> Instances**
8. Click **Launch Instance**
9. Enter the following information:

Details Tab	
Parameter	**Value**
Availability Zone	nova
Instance Name	app01
Flavor	M1.micro
Instance Count	1
Instance Boot Source	Boot from image
Image Name	cirros-0.3.4-x86_64-uec

Key Pair	devkeypair
Security groups	Linux App Servers
Selected Networks	qa_app_internal
Direct Input	#cloud-config chpasswd: list: | root:P@ssw0rd centos:P@ssw0rd expire: False ssh_pwauth: True
Disk Partition	Automatic
Configuration Drive	Unchecked

10. Navigate to **Project >> Compute >> Instances**

11. Click **Launch Instance**

12. Enter the following information:

Details Tab	
Parameter	**Value**
Availability Zone	nova
Instance Name	db01
Flavor	M1.micro
Instance Count	1
Instance Boot Source	Boot from image
Image Name	cirros-0.3.4-x86_64-uec
Key Pair	devkeypair
Security groups	Linux DB Servers
Selected Networks	qa_db_internal
Direct Input	#cloud-config chpasswd: list: | root:P@ssw0rd centos:P@ssw0rd expire: False ssh_pwauth: True
Disk Partition	Automatic
Configuration Drive	Unchecked

13. Click **Launch Instance**.

14. Click the **Actions** drop down next to **web01**

15. Select **Associate Floating IP, c**hoose an address from the dropdown

16. Click **Associate**.

Lab 2 Exercise 9: Create Volumes and Snapshots

1. Log into the Horizon dashboard as qa-admin / password (if you were signed out after the previous lab)
2. Navigate to **Project >> Compute >> Volumes**.
3. Click **Create Volume**.
4. Enter the following information.

Details Tab	
Parameter	Value
Volume Name	web01-data
Description	web server data volume
Volume Source	No source, empty volume
Type	No volume type
Size	1 GB
Availability Zone	Any availability zone

5. Click **Create Volume**.
6. Navigate to **Project >> Compute >> Volumes**
7. In the **Edit Volume** dropdown menu choose **Manage Attachments**
8. In the **Attach to Instance** dropdown, choose **web01**
9. Click **Attach Volume**
10. Wait for the volume attachment to complete.
11. Navigate to **Project >> Compute >> Volumes**
12. From the Dropdown next to web01-data, choose **Create Snapshot**.
13. Enter the following information.

Create Volume Snapshot Tab	
Parameter	Value
Snapshot Name	snap-web01
Description	web01 data snapshot

14. Click **Create Volume Snapshot (Force)**
15. Navigate to **Project >> Compute >> Volumes**
16. Click the **Volume Snapshots** tab
17. Choose **Create Volume** next to the snapshot
18. Enter the following information:

Create Volume	
Parameter	Value
Snapshot Name	snap-from-web01

Description	web01 data volume from snapshot
Use snapshot as a source	snap-web01 (1GB)
Type	No volume type
Size (GB)	1

19. Click **Create Volume**
20. Navigate to **Project >> Compute >> Volumes**
21. In the **Edit Volume** dropdown menu choose **Manage Volume Attachments** next to **snap-from-web01**
22. In the **Attach to Instance** dropdown, choose **app01**
23. Click **Attach Volume**.

Lab 2 Exercise 10: Create Instance Snapshots

1. Log into the Horizon dashboard as qa-admin / password (if you were signed out after the previous lab)
2. Navigate to **Project >> Compute >> Instances**
3. In the dropdown next to **DB01**, choose **Create Snapshot**
4. For the snapshot name, type **DB01-snapshot**
5. Click **Create Snapshot.**