3. Private Cloud

a. Setup a Private Cloud by performing the procedure using a Single node Openstack/Opennebula implementation.

AIM:

To execute the procedure to launch virtual machine using Openstack

OpenStack is an open-source software cloud computing platform. OpenStack is primarily used for deploying an infrastructure as a service (IaaS) solution like Amazon Web Service (AWS). In other words, you can make your own AWS by using OpenStack. If you want to try out OpenStack, **TryStack** is the easiest and free way to do it.

In order to try OpenStack in TryStack, you must register yourself by joining TryStack FacebookGroup. The acceptance of group needs a couple days because it's approved manually. After you have been accepted in the TryStack Group, you can log in TryStack.

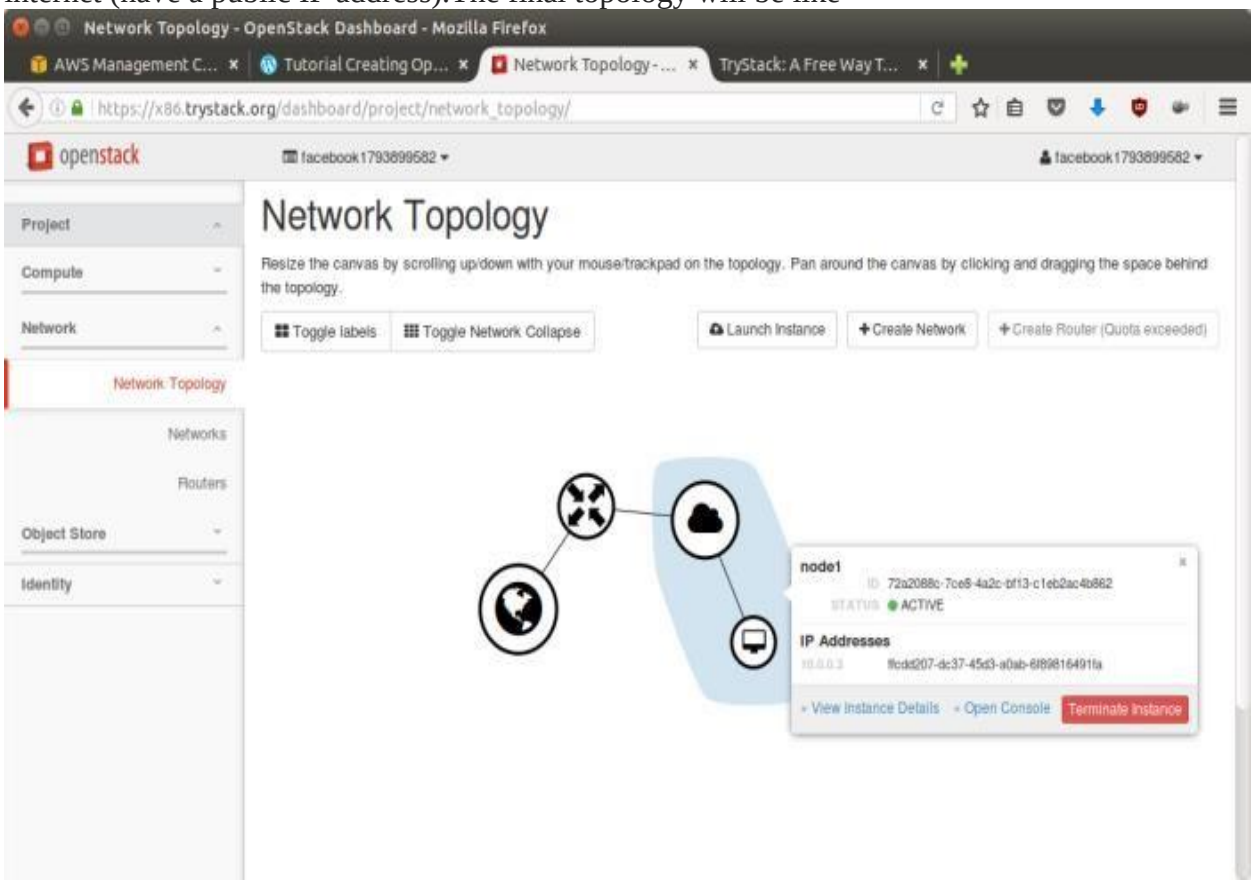
TryStack.org Homepage

I assume that you already join to the Facebook Group and login to the dashboard. After youlog in to the TryStack, you will see the Compute Dashboard like:

OpenStack Compute Dashboard

Overview: What we will do?

In this post, I will show you how to run an OpenStack instance. The instance will be accessible through the internet (have a public IP address).The final topology will be like



Network topology

As you see from the image above, the instance will be connected to a local network and the local network will be connected to internet.

Step 1: Create Network

Network? Yes, the network in here is our own local network. So, your instances will be not mixed up with the others. You can imagine this as your own LAN (Local Area Network) in the cloud.

1. Go to **Network > Networks** and then click **CreateNetwork**.
2. In **Network** tab, fill **Network Name** for example internal and then click**Next**.
3. In **Subnet**tab,
1. Fill **Network Address** with appropriate CIDR, for example 192.168.1.0/24. Use private network CIDR block as the bestpractice.
2. Select **IP Version** with appropriate IP version, in this caseIPv4.
3. Click**Next**.
4. In **Subnet Details** tab, fill **DNS Name Servers** with 8.8.8.8 (Google DNS) and then click**Create**.

Step 2: Create Instance

Now, we will create an instance. The instance is a virtual machine in the cloud, like AWS EC2. You need the instance to connect to the network that we just created in the previous step.

1. Go to **Compute > Instances** and then click **LaunchInstance**.
2. In **Details**tab,
1. Fill **Instance Name**, for example Ubuntu1.
2. Select **Flavor**, for examplem1.medium.
3. Fill **Instance Count** with**1**.
4. Select **Instance Boot Source** with **Boot fromImage**.
5. Select **Image Name** with **Ubuntu 14.04 amd64 (243.7 MB)** if you want installUbuntu 14.04 in your virtual machine.
3. In **Access & Security**tab,
1. Click [+] button of **Key Pair** to import key pair. This key pair is a public and private keythat we will use to connect to the instance from ourmachine.
2. In **Import Key Pair**dialog,
1. Fill **Key Pair Name** with your machine name (for exampleEdward-Key).
2. Fill **Public Key** with your **SSH public key** (usually is in ~/.ssh/id_rsa.pub). See description in Import Key Pair dialog box for more information. If you are using Windows, you can use **Puttygen** to generate keypair.
3. Click **Import keypair**.
3. In **Security Groups**, mark/check**default**.
4. In **Networking**tab,
1. In **Selected Networks**, select network that have been created in Step 1, for exampleinternal.
5. Click**Launch**.
6. If you want to create multiple instances, you can repeat step 1-5. I created one more instance with instance name Ubuntu2.

Step 3: Create Router

I guess you already know what router is. In the step 1, we created our network, but it is isolated. It doesn't connect to the internet. To make our network has an internet connection, we need a router that running as the gateway to the internet.

1. Go to **Network > Routers** and then click **CreateRouter**.
2. Fill **Router Name** for example router1 and then click **Createrouter**.
3. Click on your **router name link**, for example router1, **Router Details**page.
4. Click **Set Gateway** button in upperright:
1. Select **External networks** with**external**.
2. Then**OK**.
5. Click **Add Interface**button.
1. Select **Subnet** with the network that you have been created in Step 1.
2. Click **Addinterface**.
6. Go to **Network > Network Topology**. You will see the network topology. In the example, there are two network, i.e. external and internal, those are bridged by a router. Thereare

instances those are joined to internal network.

Step 4: Configure Floating IP Address

Floating IP address is public IP address. It makes your instance is accessible from the internet. When you launch your instance, the instance will have a private network IP, but no public IP. In OpenStack, the public IPs is collected in a pool and managed by admin (in our case is TryStack). You need to request a public (floating) IP address to be assigned to your instance.

1. Go to **Compute >Instance**.
2. In one of your instances, click **More > Associate FloatingIP**.
3. In **IP Address**, click Plus[+].
4. Select **Pool** to **external** and then click **AllocateIP**.
5. Click**Associate**.
6. Now you will get a public IP, e.g. 8.21.28.120, for yourinstance.

Step 5: Configure Access & Security

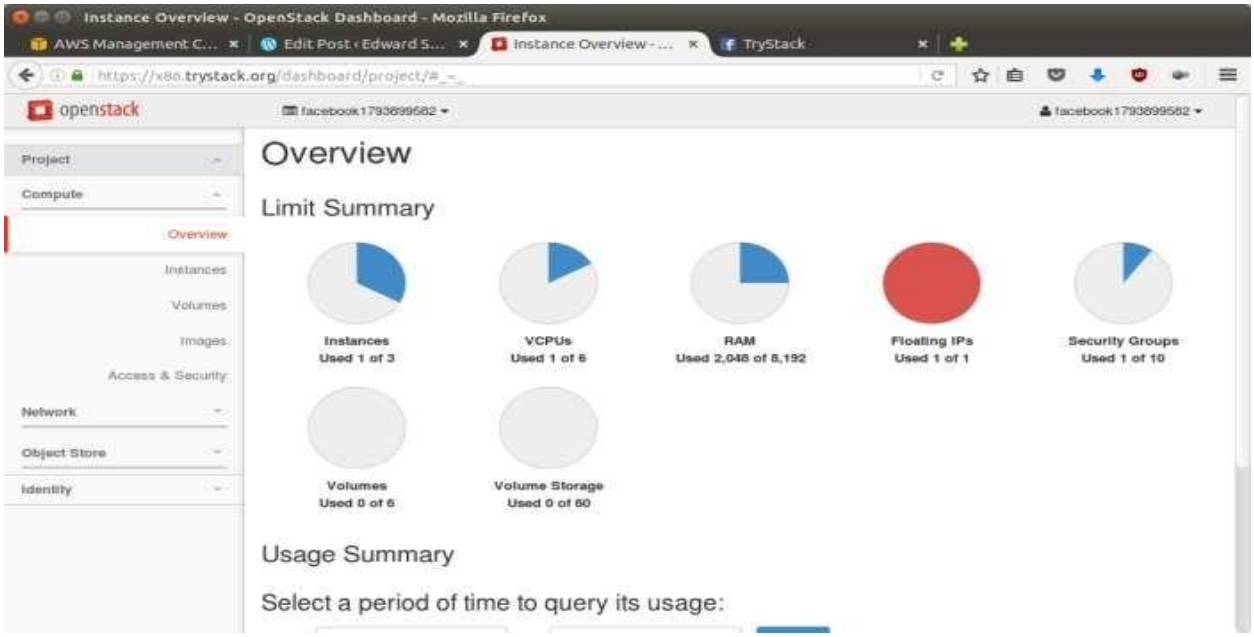
OpenStack has a feature like a firewall. It can whitelist/blacklist your in/out connection. It is called Security Group.

1. Go to **Compute > Access & Security** and then open **Security Groups**tab.
2. In **default** row, click **ManageRules**.
3. Click **Add Rule**, choose **ALL ICMP** rule to enable ping into your instance, and then click**Add**.
4. Click **Add Rule**, choose **HTTP** rule to open HTTP port (port 80), and then click**Add**.
5. Click **Add Rule**, choose **SSH** rule to open SSH port (port 22), and then click**Add**.
6. You can open other ports by creating newrules.

Step 6: SSH to Your Instance

Now, you can SSH your instances to the floating IP address that you got in the step 4. If you are using Ubuntu image, the SSH user will be ubuntu.

Output:



Result:

Thus the above open stack Application created and Executed successfully.