**Assignment No-5**

**Aim:** Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

# Objective:

* To launch virtual machine on cloud
* Creating openstack instance (virtual machine) in trystack
* To learn deployment of virtual machine on open-source cloud computing platform

# Theory:

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.

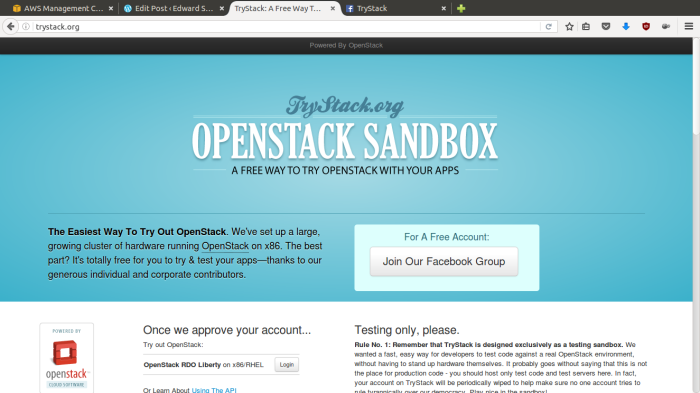
Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster.

**TASK 1: Launch Virtual Machine (VM) on AWS**

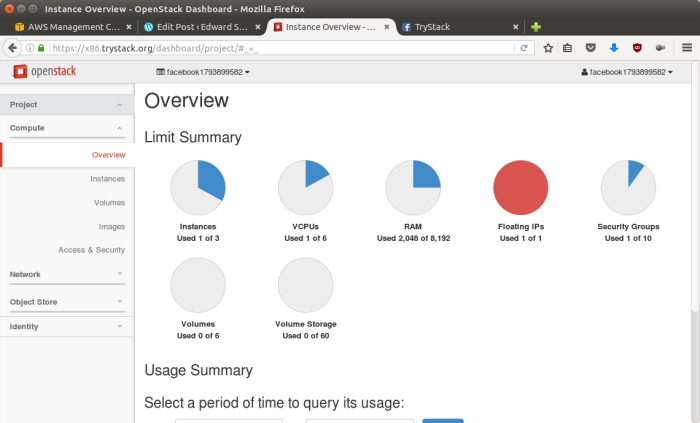
1. Launch an Amazon EC2 Instance :
2. Open Amazon EC2 console and then click Launch Instance to create and configure your virtual machine.
3. Configure your Instance

**TASK 2: CREATING OPENSTACK INSTANCE (Virtual Machine) IN TRYSTACK**

In order to try OpenStack in TryStack, you must register yourself by joining TryStack Facebook Group. After you have been accepted in the TryStack Group, you can log in TryStack.

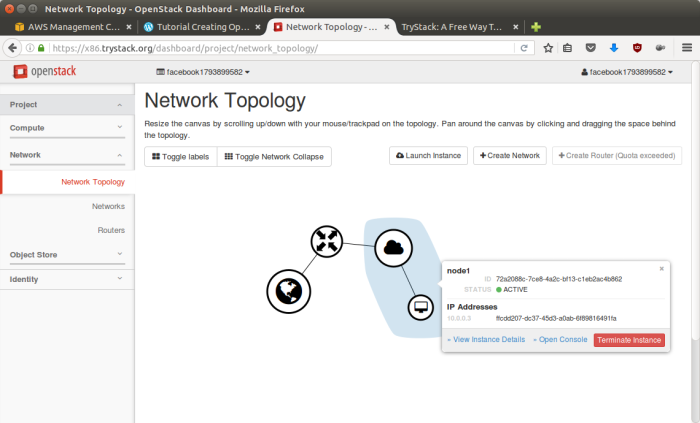


Join to the Facebook Group and login to the dashboard. After you log in to the TryStack, you will see the Compute Dashboard like:

  
OpenStack Compute Dashboard

**TASK 3: Run an OpenStack instance**

The instance will be accessible through the internet (have a public IP address). The final topology will 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.

**TASK 4: Create Network**

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 **Create Network**.
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 best practice.
   2. Select **IP Version** with appropriate IP version, in this case IPv4.
   3. Click **Next**.
4. In **Subnet Details** tab, fill **DNS Name Servers** with 8.8.8.8 (Google DNS) and then click **Create**.

**TASK 5: Create Instance**

Now, we will create an instance. The instance is a virtual machine in the cloud, like AWS EC2 (created in TASK 1). 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 **Launch Instance**.
2. In **Details** tab,
   1. Fill **Instance Name**, for example Ubuntu 1.
   2. Select **Flavor**, for example m1.medium.
   3. Fill **Instance Count** with **1**.
   4. Select **Instance Boot Source** with **Boot from Image**.
   5. Select **Image Name** with **Ubuntu 14.04 amd64 (243.7 MB)** if you want install Ubuntu 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 key that we will use to connect to the instance from our machine.
   2. In **Import Key Pair** dialog,
      1. Fill **Key Pair Name** with your machine name (for example Edward-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 key pair.
      3. Click **Import key pair**.
   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 example internal.
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 Ubuntu 2.

**TASK 6: 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 **Create Router**.
2. Fill **Router Name**for example router1 and then click **Create router**.
3. Click on your **router name link**, for example router1, **Router Details** page.
4. Click **Set Gateway** button in upper right:
   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 **Add interface**.
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. There are instances those are joined to internal network.

**TASK 7: 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 Floating IP**.
3. In **IP Address**, click Plus [**+**].
4. Select **Pool** to **external** and then click **Allocate IP**.
5. Click **Associate**.
6. Now you will get a public IP, e.g. 8.21.28.120, for your instance.

**TASK 8: 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 **Manage Rules**.
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 new rules.

**TASK 9: 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.

**Conclusion:** Thus, we have studied implementation of virtualization and understood the procedure to launch virtual machine using trystack.