Ex. No: 9 Google App Engine Launcher to launch the web applications

Aim:

To find the procedure to launch the web applications using Google App Engine.

Procedure:

To use GAE launcher to launch the web applications.

You can use Google App Engine to host a static website.

Static web pages can contain client-side technologies such as HTML, CSS, and JavaScript.

Hosting your static site on App Engine can cost less than using a traditional hosting provider, as App Engine provides a free tier.

Sites hosted on App Engine are hosted on the REGION_ID.r.appspot.com subdomain, such as [my-project-id].uc.r.appspot.com. After you deploy your site, you can map your own domain name to your App Engine-hosted website.

Before you begin

Before you can host your website on Google App Engine:

1. Create a new Cloud Console project or retrieve the project ID of an existing project to use: Go to the Project page

2. Install and then initialize the Google Cloud SDK:

Download the SDK

Creating a website to host on Google App Engine

Basic structure for the project

This guide uses the following structure for the project:

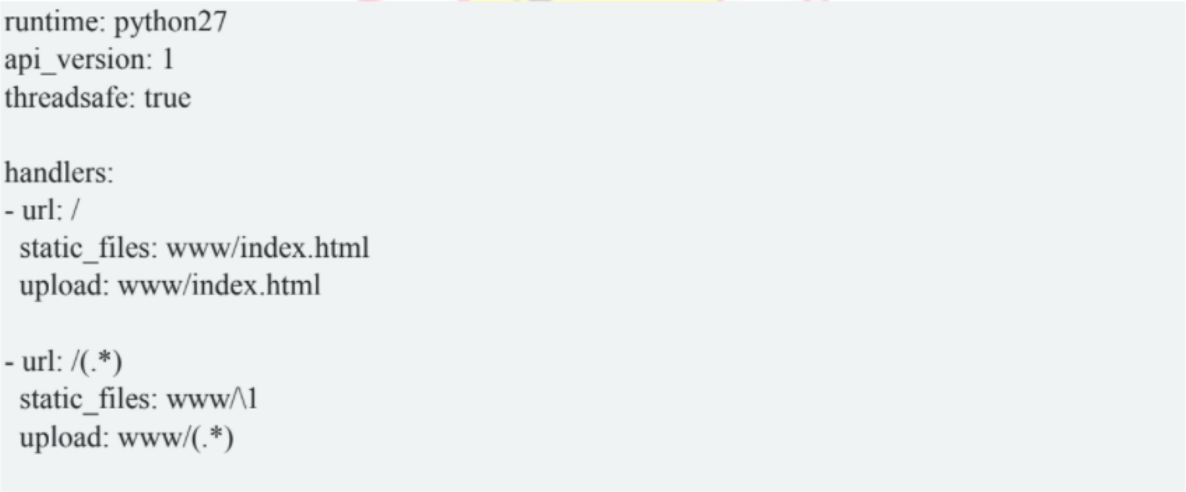
- app.yaml: Configure the settings of your App Engine application.
- www/: Directory to store all of your static files, such as HTML, CSS, images, and JavaScript.
- css/: Directory to store stylesheets.
- style.css: Basic stylesheet that formats the look and feel of your site.
- images/: Optional directory to store images.
- index.html: An HTML file that displays content for your website.
- js/: Optional directory to store JavaScript files.
- Other asset directories.

Creating the app.yaml file

The app.yaml file is a configuration file that tells App Engine how to map URLs to your static files. In the following steps, you will add handlers that will load www/index.html when someone visits your website, and all static files will be stored in and called from the www directory.

Create the app.yaml file in your application's root directory:

1. Create a directory that has the same name as your project ID. You can find your project ID in the Console.
2. In directory that you just created, create a file named app.yaml.
3. Edit the app.yaml file and add the following code to the file:



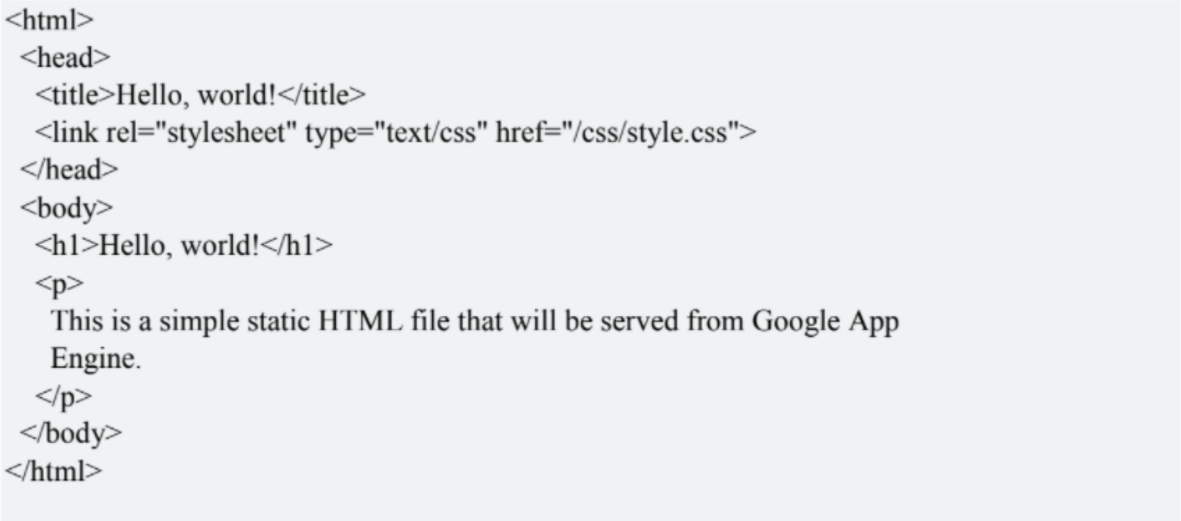
```
runtime: python27
api_version: 1
threadsafe: true

handlers:
- url: /
  static_files: www/index.html
  upload: www/index.html

- url: /(.*)
  static_files: www/\1
  upload: www/(.*)
```

Creating the index.html file

Create an HTML file that will be served when someone navigates to the root page of your website. Store this file in your www directory.



```
<html>
  <head>
    <title>Hello, world!</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <p>
      This is a simple static HTML file that will be served from Google App
      Engine.
    </p>
  </body>
</html>
```

Deploying your application to App Engine

When you deploy your application files, your website will be uploaded to App Engine. To deploy your app, run the following command from within the root directory of your application where the app.yaml file is located:

Optional flags:

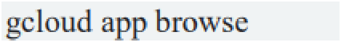
- Include the --project flag to specify an alternate Cloud Console project ID to what you initialized as the default in the gcloud tool.
    - Example: --project [YOUR_PROJECT_ID]
- Include the -v flag to specify a version ID, otherwise one is generated for you. Example: -v

[YOUR_VERSION_ID]

To learn more about deploying your app from the command line, see Deploying a Python 2 App.

Viewing your application

To launch your browser and view the app at https://PROJECT_ID.REGION_ID.r.appspot.com, run the following command:



```
gcloud app browse
```

Result:

The procedure to launch the web applications using Google App Engine is described.

Ex. No: 10 Procedure to transfer the files from one virtual machine to another virtual machine

Aim:

To find the procedure to transfer the files from one virtual machine to another virtual machine

Procedure:

This page provides information on transferring files between different types of virtual machines provided by the SCS. It is often useful to be able to transfer files between the guest machine and the host machine. On this page, you will find instructions for the following methods:

- Creating a Shared Folder in VirtualBox
- Dragging and Dropping Files in VirtualBox
- Managing Files with NextCloud.

Creating a Shared Folder in VirtualBox

A shared folder is a folder that makes its files available on both the guest machine and the host machine at the same time. Creating a shared folder between the guest and the host allows you to easily manage files that should be present on both machines. The course virtual machines are ready to use shared folders right away, but if you are using the virtual machine on your personal computer, then you will need to specify which folder to use as shared storage.

Shared Folders on Personal Computers

If you are using your own personal machine, you will need to configure VirtualBox to look in the right place for your shared files.

- Video Tutorial (Windows)
- Quick Start Guide

First, click on the guest machine you intend to share files with. From there, you can select the guest *Settings* and navigate to *Shared Folders* on the left side menu. To create a new shared folder, either click the *New Folder* icon on the right menu **or** right-click the empty list of shared folders and click *Add Shared Folder.* From here, there are six options:

- **Folder Path:** The folder name on the **host** machine. Click the drop-down menu and navigate to the folder you would like to share.
- **Folder Name:** This is the name of the folder as it will appear on the **guest** machine.
- **Read-Only:** If you check read-only, the **guest** machine will be unable to write changes to the folder. This is valuable when you only want to send files *to* the virtual machine, but do not want to risk having the files modified by the guest.
- **Auto-Mount:** When any external storage is connected to a computer, it must be *mounted* in order to be used. It is recommended that you turn on auto-mounting unless you are familiar with the process of mounting a drive yourself.
- **Mount Point:** Unless you already know about mount points, leave this blank.

- **Make Permanent:** If you check this, the shared folder will be a permanent **machine folder**. If it is not checked, the folder will not be shared after a shutdown.

Dragging and Dropping Files in VirtualBox

If you only need to transfer a few files quickly, you can simply drag and drop the files in. On the top bar of the running guest machine, click on *Devices > Drag and Drop* and make sure that *Bidirectional* is selected. This means that you will be able to drag files from the host to the guest and from the guest to the host. Once bidirectional drag and drop is checked, you should be able to begin dragging and dropping files.

NOTE: Sometimes when dragging files *into* **the course VM**, you may not be able to drag them into the file browser directly. If you encounter this issue, you should drag your files onto the *Desktop* and move the files around from there. You should see the cursor change when it is ready to drop files.
You can also drag files from the guest machine into the host. To do this, simply open the file browser on the host to where you would like to drop the files and drag the files from the virtual machine into the file browser of the host. File transfers should be pretty quick; if the virtual machine seems stuck when transferring, simply cancel the transfer and try again.

Managing Files with NextCloud

On any virtual machine, including VirtualBox, VMWare, or the virtual machines hosted on the SCS OpenStack, you can access the SCS NextCloud services to move files between multiple machines and your SCS Windows Account storage. NextCloud offers you all of your SCS storage in one remote location, similar to how you might use other file hosting services like Dropbox or Google Drive. Before trying to use NextCloud, you should check that you can access the service by logging in here.

If you can access the NextCloud services, you can browse the various file storage services available to you:

- **Linux Home:** These are the files from your SCS Linux Account
- **Windows Home:** These are the files from your SCS Windows Account and your lab Z:\ drive.
- **NextCloud:** In addition to the other storage accounts provided to you by the SCS, you can also upload up to 20GB of files directly to NextCloud.

With NextCloud, you can upload your files from any machine with an internet connection and download them onto any other machine with an internet connection. For example, you can move project files off of your virtual machine, onto the NextCloud storage, and then download them on your personal laptop. Alternatively, you can upload files from your personal PC onto the NextCloud storage, place them into the *Windows Home* folder, and access those files from either the lab Z:\ drive or download them on a virtual machine like VirtualBox or OpenStack.

Result:

The procedure for transferring the files from one virtual machine to another virtual machine is described.

Ex. No: 11 Procedure to launch virtual machine using trystack (Online Openstack Demo Version)

AIM:

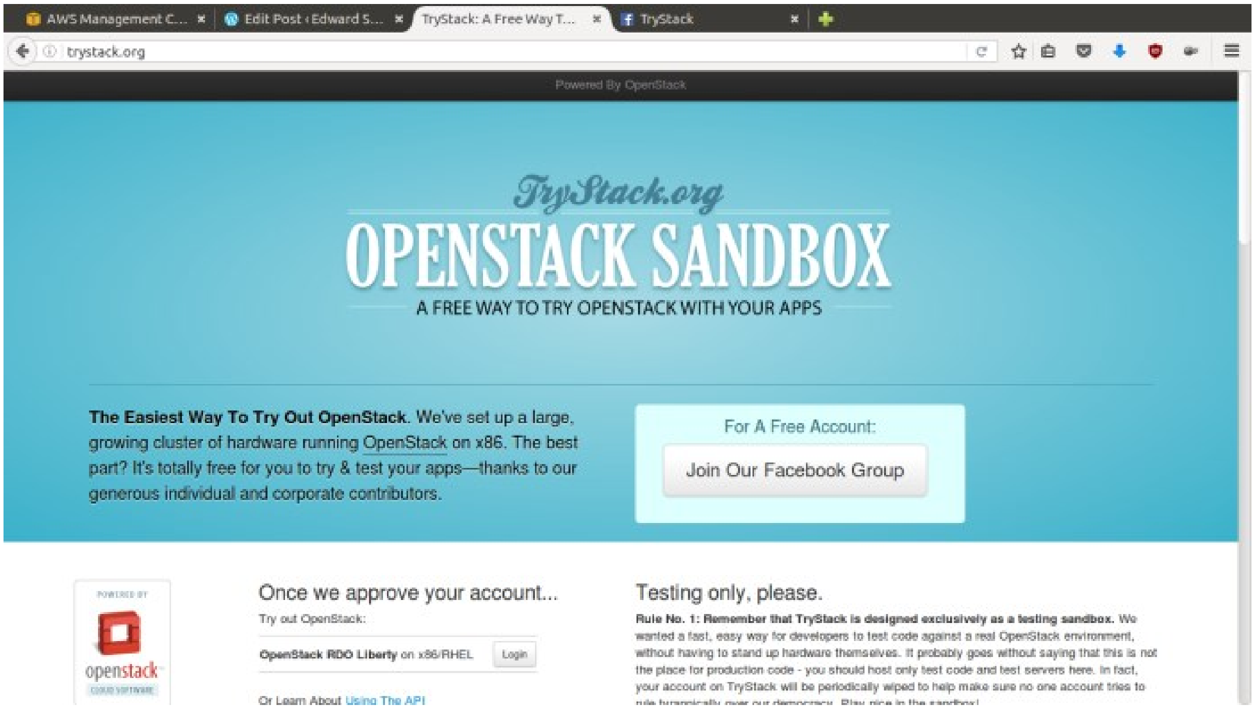
To find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

DESCRIPTION:

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.

PROCEDURE:

In order to try OpenStack in TryStack, you must register yourself by joining TryStack Facebook Group. 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

OPENSTACK SANDBOX

A FREE WAY TO TRY OPENSTACK WITH YOUR APPS

The Easiest Way To Try Out OpenStack. We've set up a large, growing cluster of hardware running OpenStack on x86. The best part? It's totally free for you to try & test your apps—thanks to our generous individual and corporate contributors.

For A Free Account:

Once we approve your account...

Try out OpenStack:

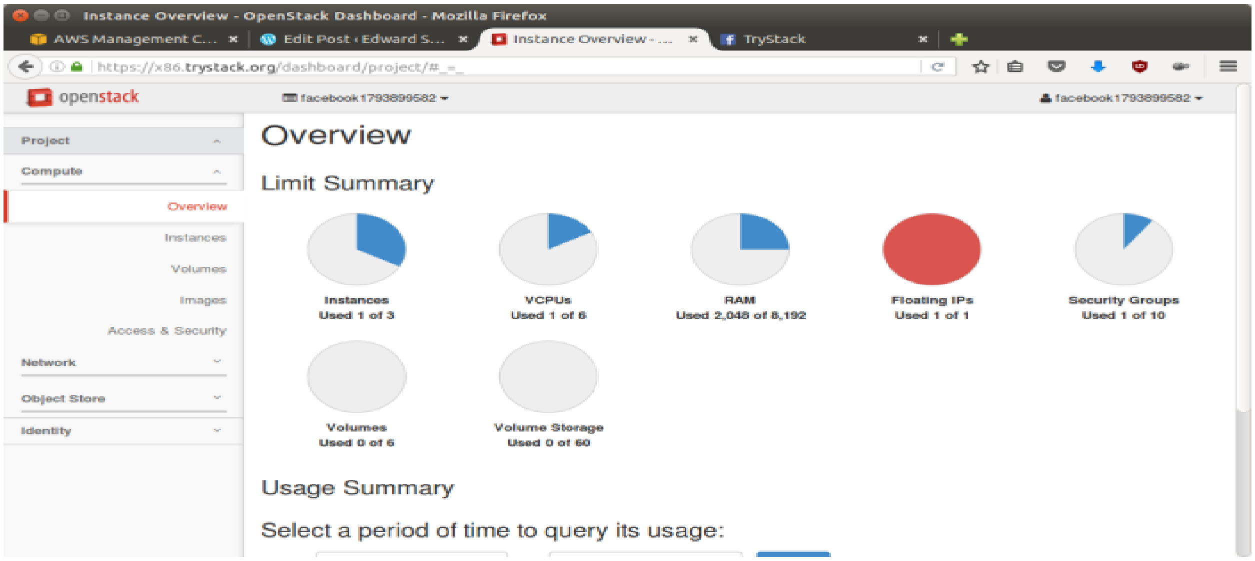
OpenStack RDO Liberty on x86/RHEL

Or Learn About Using The API

Testing only, please.

Rule No. 1: Remember that TryStack is designed exclusively as a testing sandbox. We wanted a fast, easy way for developers to test code against a real OpenStack environment, without having to stand up hardware themselves. It probably goes without saying that this is not the place for production code - you should host only test code and test servers here. In fact, your account on TryStack will be periodically wiped to help make sure no one account tries to rule tyrannically over our democracy. Play nice in the sandbox!

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

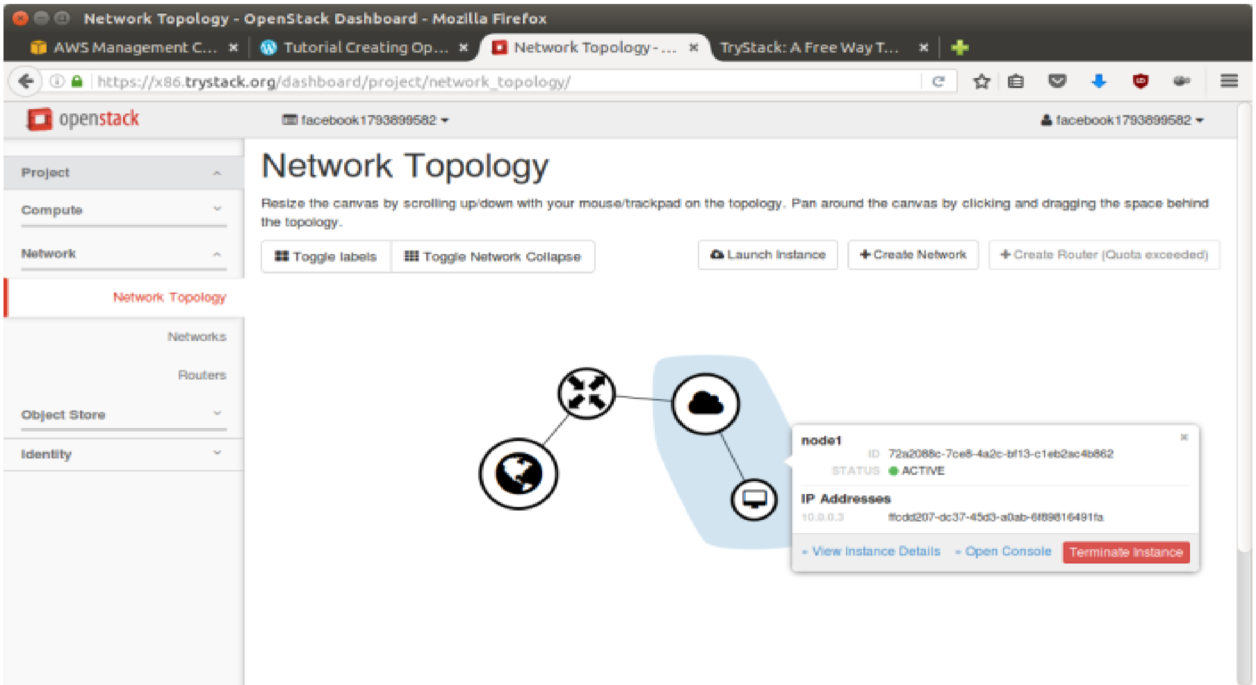


Select a period of time to query its usage:

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 like:



Resize the canvas by scrolling up/down with your mouse/trackpad on the topology. Pan around the canvas by clicking and dragging the space behind the topology.

node1
ID 72a2088c-7ce8-4a2c-bf13-c1eb2ac4b862
STATUS ● ACTIVE
IP Addresses
10.0.0.3 ffcdd207-dc37-45d3-a0ab-6f898164 91fa

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.

Go to Network > Networks and then click Create Network.

In Network tab, fill Network Name for example internal and then click Next.

In Subnet tab,

Fill Network Address with appropriate CIDR, for example 192.168.1.0/24. Use private network CIDR block as the best practice.

Select IP Version with appropriate IP version, in this case IPv4.

Click Next.

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.

Go to Compute > Instances and then click Launch Instance.

In Details tab,

Fill Instance Name, for example Ubuntu 1.

Select Flavor, for example m1.medium.

Fill Instance Count with 1.

Select Instance Boot Source with Boot from Image.

Select Image Name with Ubuntu 14.04 amd64 (243.7 MB) if you want install Ubuntu 14.04 in your virtual machine.

In Access & Security tab,

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.

In Import Key Pair dialog,

Fill Key Pair Name with your machine name (for example Edward-Key).

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.

Click Import key pair.

In Security Groups, mark/check default.

In Networking tab,

In Selected Networks, select network that have been created in Step 1, for example internal.

Click Launch.

If you want to create multiple instances, you can repeat step 1-5. I created one more instance with instance name Ubuntu 2.

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.

Go to Network > Routers and then click Create Router.

Fill Router Name for example router1 and then click Create router.

Click on your router name link, for example router1, Router Details page.

Click Set Gateway button in upper right:

Select External networks with external.

Then OK.

Click Add Interface button.

Select Subnet with the network that you have been created in Step 1.

Click Add interface.

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.

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.

Go to Compute > Instance.

In one of your instances, click More > Associate Floating IP.

In IP Address, click Plus [+].

Select Pool to external and then click Allocate IP.

Click Associate.

Now you will get a public IP, e.g. 8.21.28.120, for your instance.

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.

Go to Compute > Access & Security and then open Security Groups tab.

In default row, click Manage Rules.

Click Add Rule, choose ALL ICMP rule to enable ping into your instance, and then click Add.

Click Add Rule, choose HTTP rule to open HTTP port (port 80), and then click Add.

Click Add Rule, choose SSH rule to open SSH port (port 22), and then click Add.

You can open other ports by creating new rules.

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.

RESULT:

The procedure to launch virtual machine using trystack (Online Openstack Demo Version) is described.