Version 1.1



Dhany Ong Jesvinder Singh

S10019565 s10024982

ApaC Communications

Table of Contents

[Final Report Template 4](#_Toc11082080)

[Executive Summary 4](#_Toc11082081)

[Product Scope 4](#_Toc11082082)

[Nova 4](#_Toc11082083)

[Cinder 4](#_Toc11082084)

[Neutron 4](#_Toc11082085)

[Horizon 5](#_Toc11082086)

[Team structure 5](#_Toc11082087)

[Project communication table 5](#_Toc11082088)

[Team goals 5](#_Toc11082089)

[Background 6](#_Toc11082090)

[Infographic Timeline 7](#_Toc11082091)

[Project management timeline 7](#_Toc11082092)

[Detail Breakdown of Assigment 8](#_Toc11082093)

[Start-Up Cost 10](#_Toc11082094)

[Server 10](#_Toc11082095)

[Switch 10](#_Toc11082096)

[Hardware setup services 10](#_Toc11082097)

[Total cost 10](#_Toc11082098)

[REVENUE STREAM 11](#_Toc11082099)

[evaluation 11](#_Toc11082100)

[Risks and issues management 12](#_Toc11082101)

[Obstacles and solution 12](#_Toc11082102)

[Topology Diagram 13](#_Toc11082103)

[Proof of concept 13](#_Toc11082104)

[Logical diagram for production 13](#_Toc11082105)

[Logical diagram for bhi demonstration 15](#_Toc11082106)

[Behind the Scenes illustration 15](#_Toc11082107)

[Outcome 16](#_Toc11082108)

[Hardware requirement 16](#_Toc11082109)

[Production 16](#_Toc11082110)

[DEMONSTRATION / PROOF OF CONCEPT 17](#_Toc11082111)

[NETWORKING OPTION 1 : PROVIDER NETWORK 17](#_Toc11082112)

[NETWORKING OPTION 2: SELF-SERVICE NETWORKS 17](#_Toc11082113)

[DEMONSTRATION NETWORKING 17](#_Toc11082114)

[Change management process 19](#_Toc11082115)

[Lab For Demonstration 19](#_Toc11082116)

[Ubuntu Installation 20](#_Toc11082117)

[Configuration 26](#_Toc11082118)

[Testing and evaluation 29](#_Toc11082119)

[OpenStack Dashboard 29](#_Toc11082120)

[Create VM image 29](#_Toc11082121)

[Launch an Instance 31](#_Toc11082122)

[Storage Volume 38](#_Toc11082123)

[Reference 40](#_Toc11082124)

[Weekly Status report 40](#_Toc11082125)

# Final Report Template

## Executive Summary

Universities and schools today is vastly different from a technological standpoint even from comparison of how a typical school would have been run even as little as 10 years ago, PCloud is a simple yet powerful cloud platform that enables student and staff to be able to obtain a mix of web services for a fraction of a cost or no cost depending on the capability of piggybacking on existing hardware. Services include (IaaS) Infrastructure as a service, (PaaS) Platform as a service and (SaaS) Software as a service.

## Product Scope

This project uses OpenStack software to deploy and offer Infrastructure and Storage as a service. The main components involved in this project are:

* Nova
* Cinder
* Neutron
* Horizon

### Nova

Nova is an important component in OpenStack that allows user to provision virtual machines and use it as part of cloud services. Apart from creating virtual machines, user can also create baremetal servers and system container. However, system container has limited support in this component. The tools that are required for end user to use Nova are Horizon, OpenStack Client, and Nova Client. Hence, these components must be installed on the client site.

### Cinder

To offer OpenStack Storage as a Service, we are required to deploy OpenStack Cinder. It provides volume to Nova Virtual Machine, Ironic Baremetal Hosts, containers and more. On client site or the end user, there are 3 ways to manage the volume, which are Horizon Dashboard, Python-CinderClient or directly using REST-API.

### Neutron

Neutron offers network connectivity as a service. However, we do not emphasize this feature in this project. Neutron will only be used to provide necessary network connectivity for client to access the compute and storage services.

### Horizon

Lastly, graphic user interface for client to access and manage compute and storage services. In another words, OpenStack Dashboard. A screenshot of the dashboard of our project is shown below:

## Team structure

From planning to delivery

Dhany Ong – Technical architect

Jesvinder Singh – Technical architect

Internal Stakeholders

Boxhill Institute

To provide server rack for storage and users to utilize services

ICT320 – Students

The first batch of users to use services upon project completion

Jesvinder Singh

Dhany Ong

### Project communication table

| Document | Recipients | Responsibilities | Update frequency |
| --- | --- | --- | --- |
| Executive status report | Dhany Ong | Responsibility | Weekly |
| Risk management document | Jesvinder Singh | Responsibility | Fortnightly |
| Issue management document | Dhany Ong | Responsibility | Fortnightly |
| Change control document | Dhany Ong | Responsibility | Monthly |
| Project schedule | Jesvinder Singh | Responsibility | Weekly |
|  |  |  |  |

### Team goals

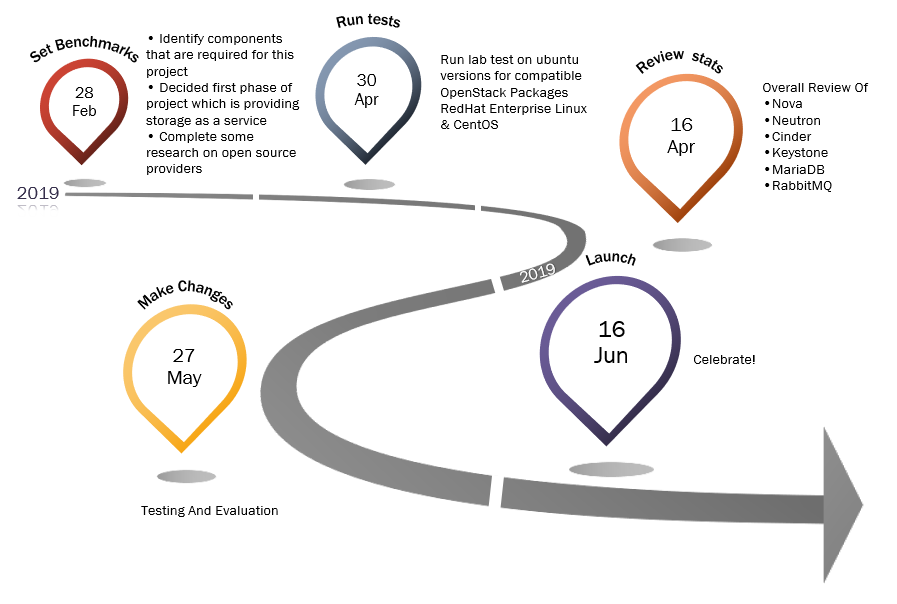
* To Be the first Cloud service provider within the Box Hill Institute
* To be able to monetarize on this platform in the long run upon successful deployment of services.
* To add PaaS as an add-on upon service rollout
* To be one of the top 10 cloud service providers within Asia Pacific
* To be able to give back to the community through charity organizations
* To provide internship programs for fresh graduates who would like to gain skills and knowledge within the managed services industry.
* To offer affordable cloud services to students

### Background

Two of us who are participating in this project are final year students for our bachelor’s degree in Box Hill. This will be the last semester for Dhany Ong while my second last semester for our bachelors. The concept here is to provide storage as a service to students within the Box Hill environment. This project will test us in terms of our knowledge of storage, networking, and virtualization. The main challenge here is that we are using Open stack for our project which is something new for both of us.

The idea came around when we both were both enrolled in cloud computing and desktop virtualization. Upon gaining knowledge on cloud computing from Sudipto and Frank for desktop virtualization, we thought let’s challenge our selves and instead of running hyper-v or Microsoft products, lets reach out and use all opensource platforms for this project. The outcome of this project will provide BHI students with free access to storage and Vm’s within the BHI environment. We have also included some information in the event if we decide to scale out.

### Infographic Timeline



### Project management timeline

The above timeline is provided based on our project schedule. Currently we are well ahead of the project delivery assuption.

### Detail Breakdown of Assigment

|  |  |
| --- | --- |
| **Week** | **Task** |
| **1** | Project preliminary discussion (decided to work on Cloud Infrastructure and Storage) |
| **2** | * Discussed with Umesh on our idea of having cloud services provider * Identify components that are required for this project * Decided first phase of project which is providing storage as a service |
| **3** | * Complete some research on open source providers * Identify deliverables needed for this project |
| **4** | Run lab test on ubuntu versions for compatible OpenStack Packages   1. openSUSE:    * OpenStack Stein    * OpenStack Rocky    * OpenStack Queens    * OpenStack Pike 2. RedHat Enterprise Linux & CentOS    * OpenStack Rocky    * OpenStack Queens    * OpenStack Pike 3. Ubuntu 18.04 LTS    * OpenStack Stein    * OpenStack Rocky 4. Ubuntu 16.04 LTS    * OpenStack Queens    * OpenStack Pike 5. Tested on other Ubuntu alternative installation:    * MAAS and JUJU |
| **5** | Installed Ubuntu 18.04 with OpenStack Rocky  Install OpenStack services:   * NTP Server on Controller Node * NTP Client on Compute and Storage Node |
| **6** | Install OpenStack services:   * SQL Database (MariaDB) on Controller Node * Message Queue (RabbitMQ) * Memcached |
| **7** | Install OpenStack services:   * Identity Service (Keystone) on Controller Node * Image Service (Glance) on Controller Node |
| **8** | Install OpenStack services:   * Compute Service (Nova) on Controller Node * Compute Service (Nova) on Compute Node * Networking Service (Neutron) on Controller Node * Networking Service (Neutron) on Compute Node |
| **9** | Install OpenStack services:   * Dashboard Service (Horizon) on Controller Node * Block Storage (Cinder) on Controller Node * Block Storage (Cinder) on Storage Node |
| **10** | Testing and Evaluation |

## Start-Up Cost

### Server

|  |  |  |
| --- | --- | --- |
| **Item** | **Cost (AU$)** | **Image** |
| Lenovo ST50 Server | 2,000 |  |
| Lenovo ST50 Server | 2,000 |
| Lenovo ST50 Server | 2,000 |
| **Total** | **6,000** |

### Switch

|  |  |  |
| --- | --- | --- |
| **Item** | **Cost (AU$)** | **Image** |
| Cisco Catalyst 2960-L 8 ports | 2,000 | Catalyst 2960-L  with 8 ports |

### Hardware setup services

|  |  |
| --- | --- |
| **Item** | **Cost (AU$)** |
| Server | 2,000 |
| Switch | 2,000 |
| **Total** | **4,000** |

### Total cost

Costs of estimation required for running the business per year: **AU$ 12,000**

This excludes all labor charges. This is also the initial cost as we will be using existing infrastructure of Box Hill institute. Upon successfully running this program internally, we will then plan to scale out whereby we will require additional structure as well as location. This will only be determined after we complete our first year inhouse availability. Below is the potential cost for scaling out in the event project is successful internally.

|  |  |
| --- | --- |
| Components | Cost |
| Domain ( pcloud.com.au ) | 28 AUD per year |
| 1GB Primary link via 5G Network | 10,800 AUD per year on a 4-year contract. |
| 1GB Secondary link via TPG | 10,300 AUD Per year on 4-year contract. |
| VEEM backup solution license | 680 USD per year for single license |
| MDC ( Melbourne data center) location rent Rackspace | 13000 per year |

### REVENUE STREAM

Due to the nature of the services and scope. There won’t be any revenue stream available as the services will be provided for free to students. How ever we will be engaging our suppliers which are cisco and veem to see if we can use their services for free in return, we do some advertising on the sites. Upon completing the first year. We will then provide paid services to students for little as 10 dollars per month. This will only be available when students leave Box Hill and would like to uphold their VM’s. Again, this service will only be available upon us scaling out from box hill environment. The scale out will solely depend on the success and popularity of this program. We will also be engaging with other institutes to join venture on this journey to provide similar services to students. In terms of hosting as a website. Will we will be hosting the services over a sub domain which again is free of charge. Again, the purpose of this project was to provide free services to students that are currently enrolled within the box hill environment

## evaluation

To ensure the demonstration lab runs smoothly, we adopted the evaluation technique to meet stages of requirement before proceeding to the next one. Below are the requirements that are required to be met to reach the final task.

| Requirement | Purpose |
| --- | --- |
| OpenStack Lab Test Linux OS Version Test | To decide which Linux OS and OpenStack repository to install.  After Linux OS has been decided, test on that Linux OS’s version. |
| Test SQL Database, Identity and Image Services are running | Test the Keystone and Glance connectivity |
| Test Dashboard service | Test OpenStack dashboard by logging in |
| Test Compute Service | Test Instance creation |
| Test Storage Service | Test Cinder by creating volume and attach the volume to the instance |

## Risks and issues management

### Obstacles and solution

Below is the risk and issues that have been registered as of today

| Date recorded | Risk description | Probability | Impact | Mitigation plan |
| --- | --- | --- | --- | --- |
| 03/03/2019 | Compute usage will outgrow current infrastructure | HIGH | HIGH | Upon completing the rollout. We will need to arrange for hardware expansion. In the event that funding is not available, we will then limit the number or Vm’s that can be accessed based on the current bare metal capacity. |
| 14/03/2019 | Insufficient amount of bandwidth available to access VM’s from external site. | LOW | HIGH | Arrange a meeting with all stakeholders to address the issue. Speak to the networking team to allocate x amount of bandwidth to run Vm’s from external access. |
| 15/05/2019 | Data loss recovery. Caused my hardware failure or natural disaster | LOW | HIGH | To enable incremental backup to AWS via VEEM. Currently still in the process to finalize cost involved . cost is also dependent on the amount of storage and bandwidth |
| 27/04/2019 | Issue with limited hardware resources. OpenStack requires at least 3 physical servers | HIGH | HIGH | Try Ubuntu MAAS and Juju for pre-automated OpenStack installation |
| 05/05/2019 | Issue installing OpenStack with Ubuntu MAAS and Juju | LOW | LOW | Seek for Frank’s Help  Found another method to test OpenStack, that is Devstack. |
| 08/05/2019 | OpenStack crashed when Devstack VM rebooted. This is one of Devstack | HIGH | HIGH | Pause or suspend the VM, instead of turning it off. Enabled additional virtual memory |
| 08/05/2019 | limitation, where the VM must not be rebooted. | LOW | LOW | Set VM not to hibernate under no use |
| 06/06/2019 | IP configuration changes every time the physical machine connects to new network. | HIGH | HIGH | Used NAT network instead of Bridge network. |

## Topology Diagram

### Proof of concept

Two architectures were designed for this project, one meant for Production and the other one for Demonstration or Proof of Concept.

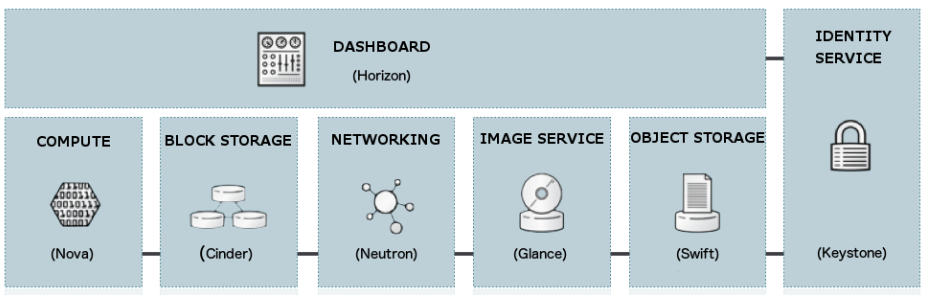
### Logical diagram for production



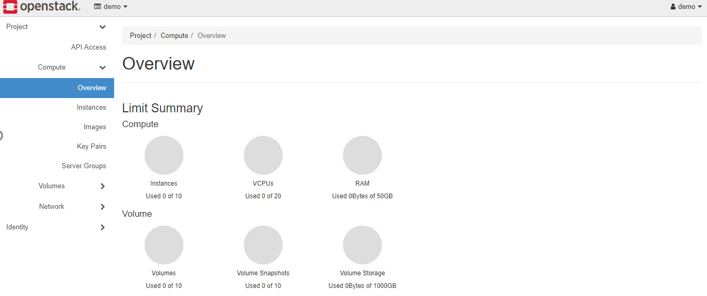
### Logical diagram for bhi demonstration



### Behind the Scenes illustration



### Outcome



Upon completing all the required task and installation, the final visability will be the openstack dashboard as illustrated above. this is where each user will be able to obtain their very on computer services that will be deployed within the organization.

## Hardware requirement

### Production

here are 3 physical nodes involved in this design, which are Controller Node, Compute Node, and Storage Node. Each of the node serves their purposes, this will be discussed further in OpenStack Components section. The hardware requirements are as follows: [a]

| Server Name | CPU | Memory | Storage | NIC |
| --- | --- | --- | --- | --- |
| Controller Node | 1 - 2 | 8GB | 100GB | 2 |
| Compute Node | 2 – 4 + | 8GB + | 100GB + | 2 |
| Storage Node | 1 – 2 | 4GB | 100GB Min | 1 |
| Switch | Cisco Catalyst 2960-L 8 Ports | | | |

### DEMONSTRATION / PROOF OF CONCEPT

This setup will be simple; hence we only require a notebook with minimum 8GB of ram and at least 40GB of hard disk space. [b]

| Machine Type | Software Required | OS | CPU / vCPU | Memory | Storage |
| --- | --- | --- | --- | --- | --- |
| Notebook | VMware  PuTTY  Web Browser | ANY | Quad Core | 8GB MIN | 40 GB MIN |
| VM | VMware | Ubuntu  CentOS  RHEL  SUSE | 2 vCPU | 4GB MIN | 20GB MIN |

Networking

Based on OpenStack documentation, 2 virtual networking options are recommended.

### NETWORKING OPTION 1 : PROVIDER NETWORK

Virtual networks are bridge to physical networks and relies on the physical network infrastructure of layer-3 routing services. DHCP services will provides IP address to instances. [a]

### NETWORKING OPTION 2: SELF-SERVICE NETWORKS

VXLAN is used in this networking option for self-service capability. [a]

## DEMONSTRATION NETWORKING

Meanwhile below is our network setup for our demonstration lab:

The 2 options can be used to setup VMware networking are bridge and NAT. However, we found that bridge will constantly change the IP configuration when the physical machine is connected to new network. Hence, NAT is used in our Lab to ensure the IP configuration unchanged when the physical machine connects to new network.

Supported OS

1. SUSE:
   * OpenStack Stein
   * OpenStack Rocky
   * OpenStack Queens
   * OpenStack Pike
2. RedHat Enterprise Linux & CentOS
   * OpenStack Rocky
   * OpenStack Queens
   * OpenStack Pike
3. Ubuntu 18.04 LTS
   * OpenStack Stein
   * OpenStack Rocky
4. Ubuntu 16.04 LTS
   * OpenStack Queens
   * OpenStack Pike
5. Ubuntu alternative installation with:
   * MAAS and JUJU

Openstack components

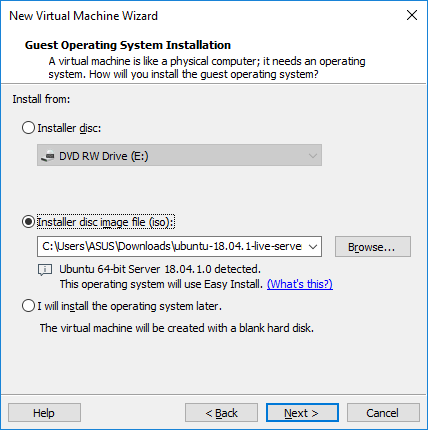
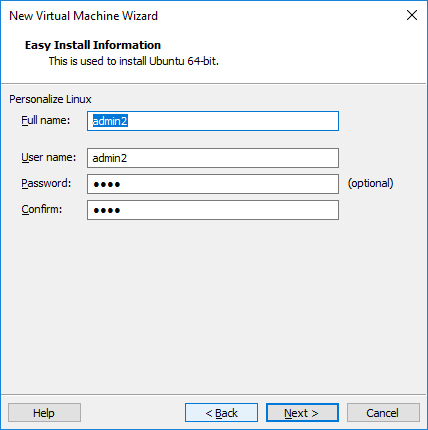
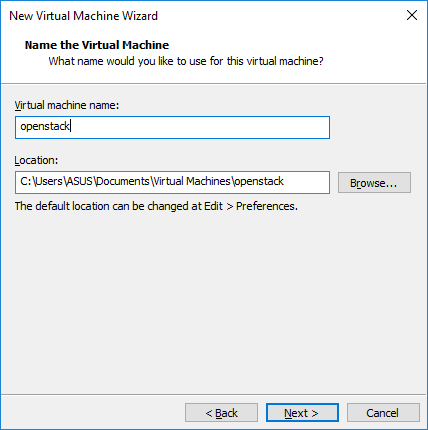
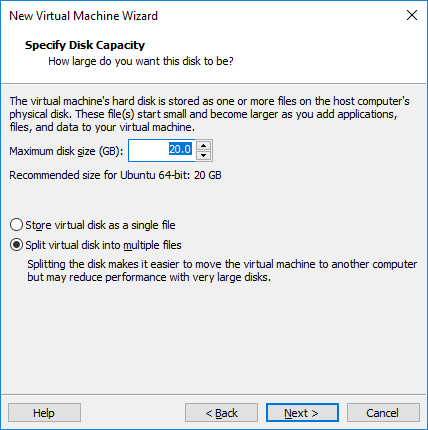
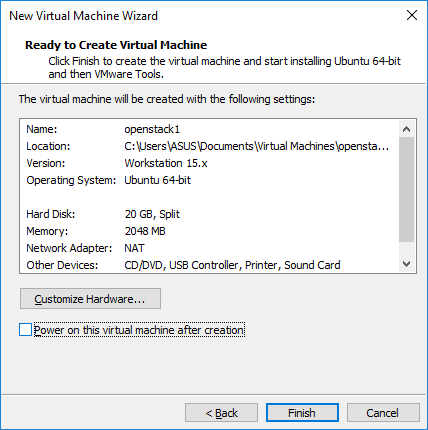
| Component | Use |
| --- | --- |
| Chrony | Synchronize the NTP on all nodes. Controller node is the NTP Server while others are NTP Client. |
| MariaDB | Is a SQL Database for OpenStack to store information. Run on Controller node. |
| RabbitMQ | OpenStack uses message queue to coordinate operations and status information among services. Run on Controller node. |
| Memcache | Is an identity service authentication mechanism used by OpenStack to cache tokens. Run on Controller node. |
| Etcd | a distributed reliable key-value store for distributed key locking, storing configuration, keeping track of service live-ness and other scenarios. Run on Controller node. |
| Keystone | OpenStack Identity service - provides API client authentication, service discovery, and distributed multi-tenant authorization |
| Glance | OpenStack Image service – Allow users to upload and discover data assets. These include discovering, registering, and retrieving virtual machine (VM) images |
| Nova | OpenStack Compute service – Provide on-demand access to users for compute resources. |
| Neutron | OpenStack Networking service – Allow users to offer “network as a service” to the cloud tenants. |
| Horizon | OpenStack Dashboard – An intuitive Graphical Interface that provides interactive user interface to users. |
| Cinder | OpenStack Storage service – Users can manage volumes on storage devices while volumes can be easily added or removed from cloud tenants. |

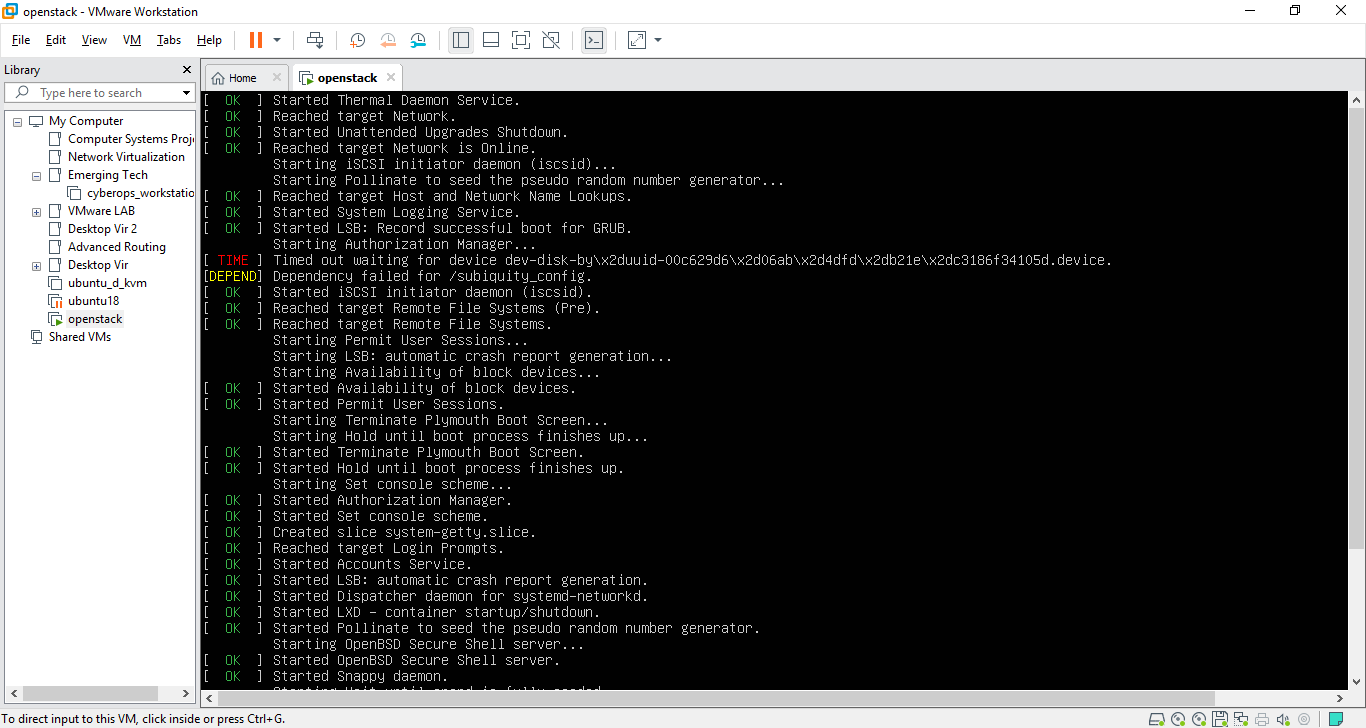
## Change management process

A change request form must be submitted in the event if a change is needed. This form is used only when changes that are out of the project scope are required.

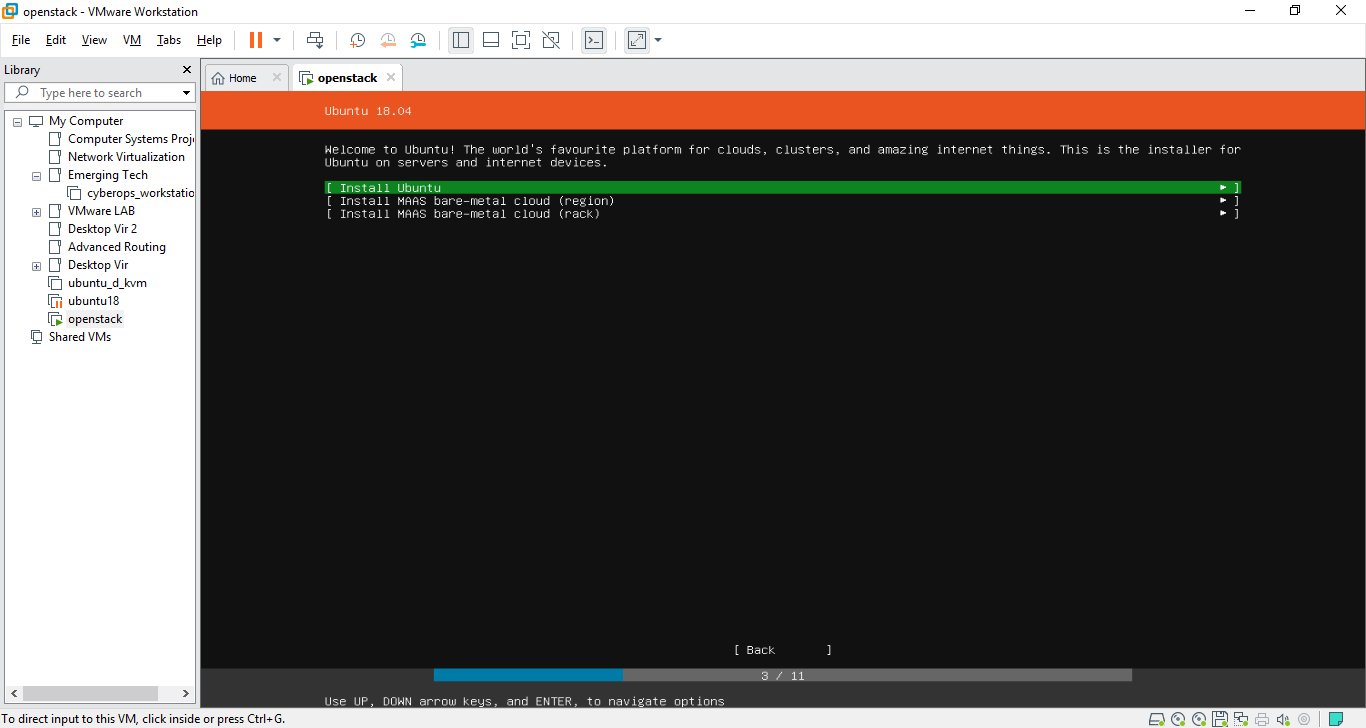
### Lab For Demonstration

## Ubuntu Installation

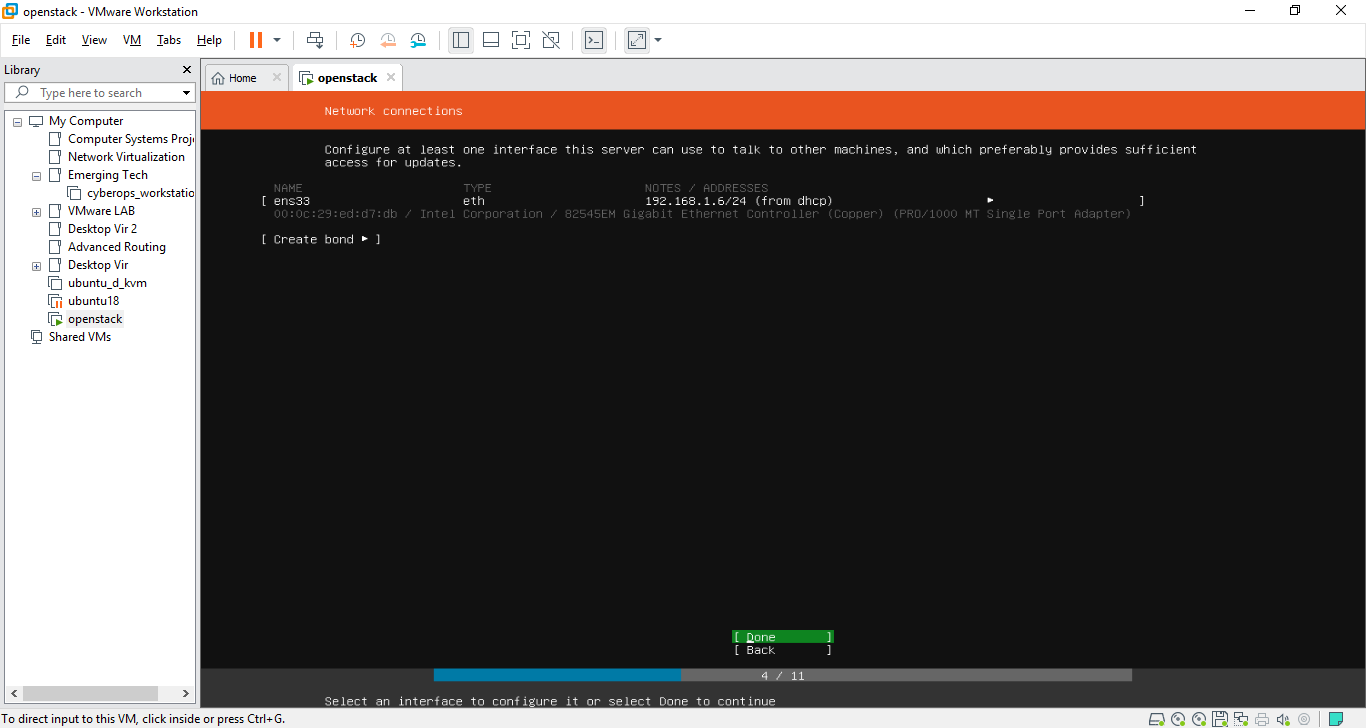
1. First, we will need to install uBuntu 18.04.2 on a blank virtual machine.
2. Select uBuntu 18.04.1 iso file. (This version will be upgraded to 18.04.2 in the later steps)
3. Provide Full name and Username.
4. Name your virtual machine and provide the location to save your VM.
5.  Increase the disk size to at least 40GB.
6. Review summary and click Finish to start the installation.
7. Installation will start and wait for the ubuntu installation wizard screen.



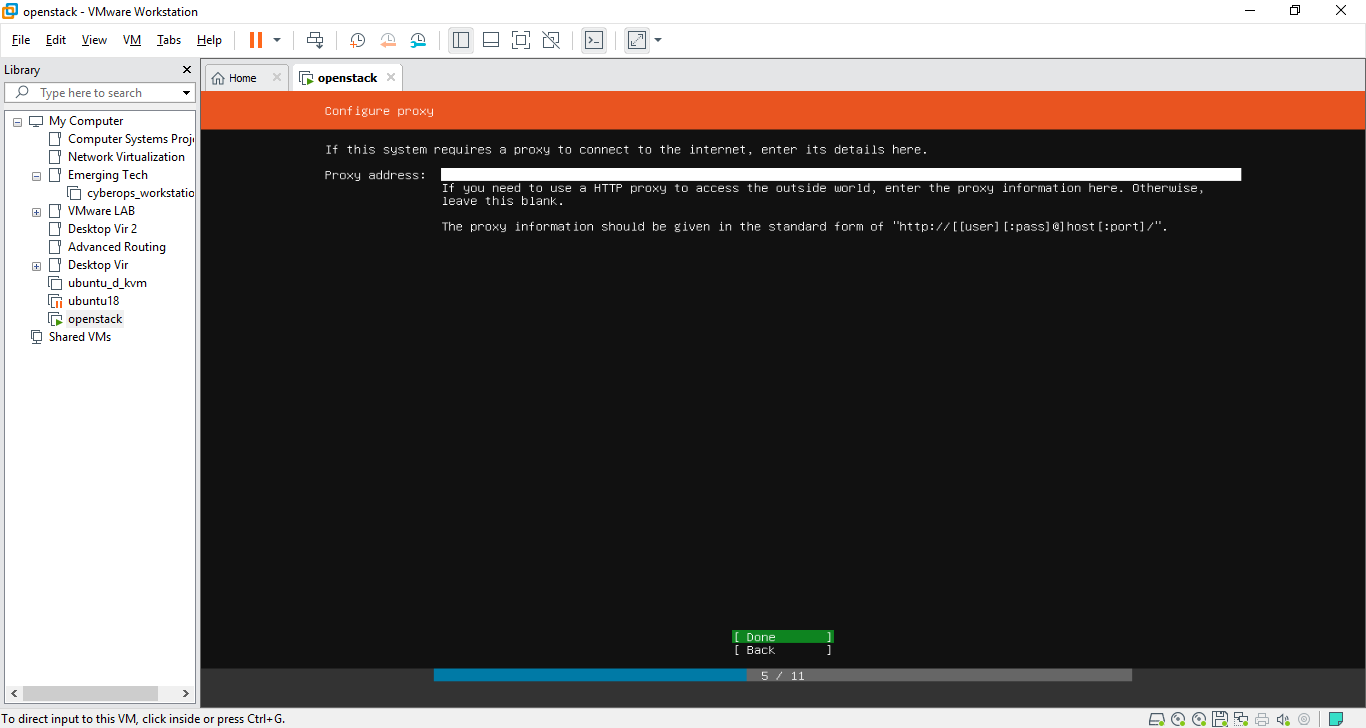
1. Select 1st option: Install Ubuntu



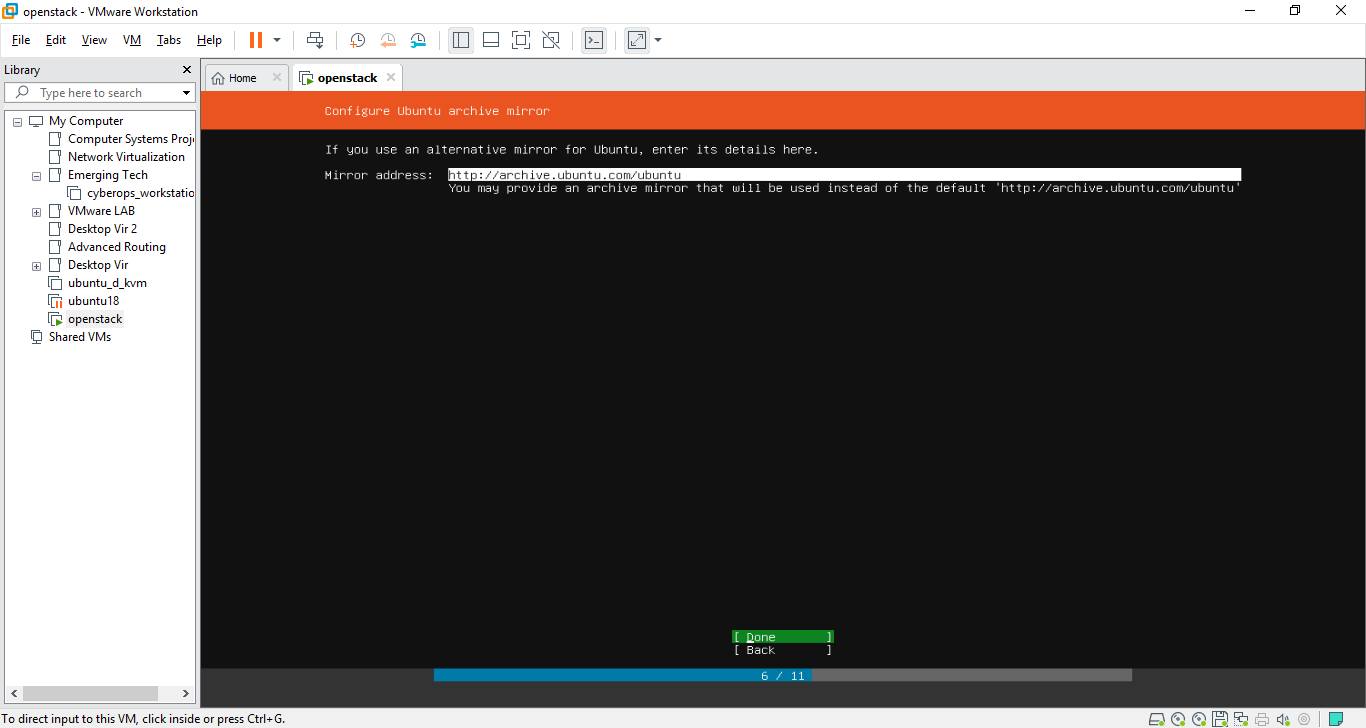
1. Review the network interface and click done



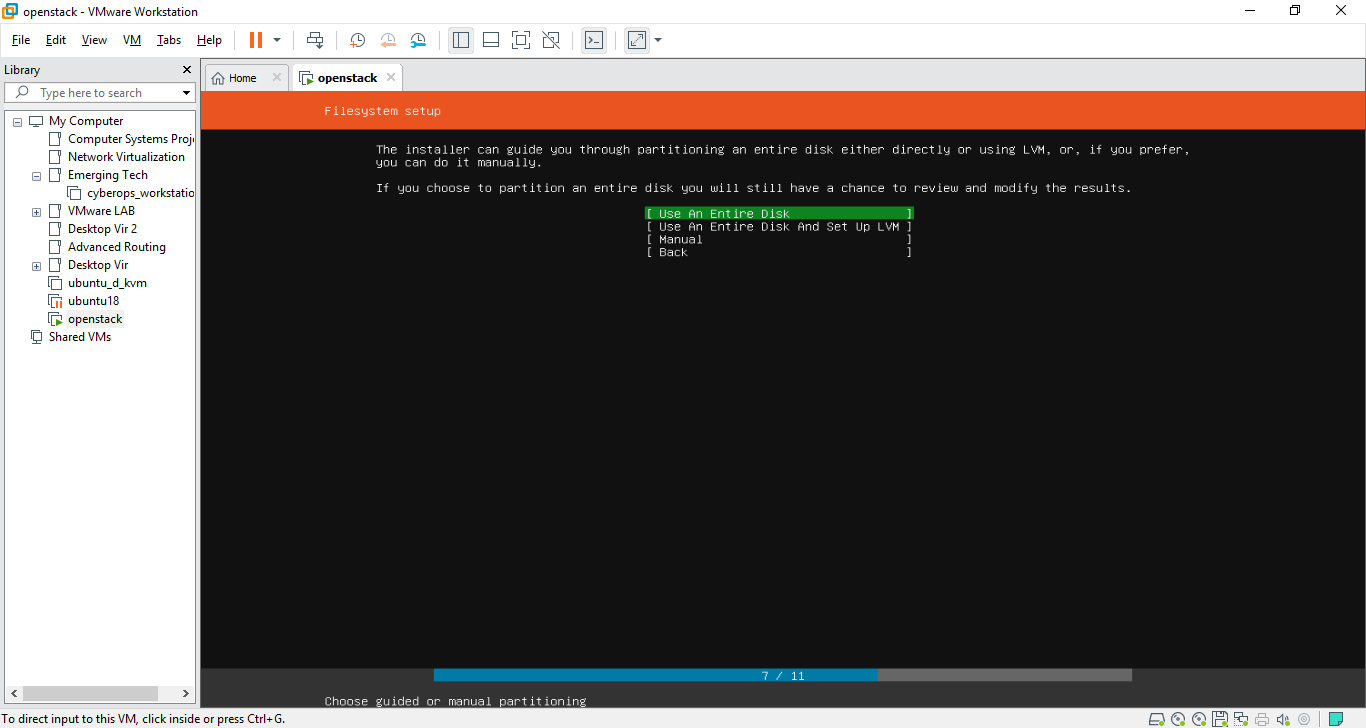
1. Leave the Proxy address blank



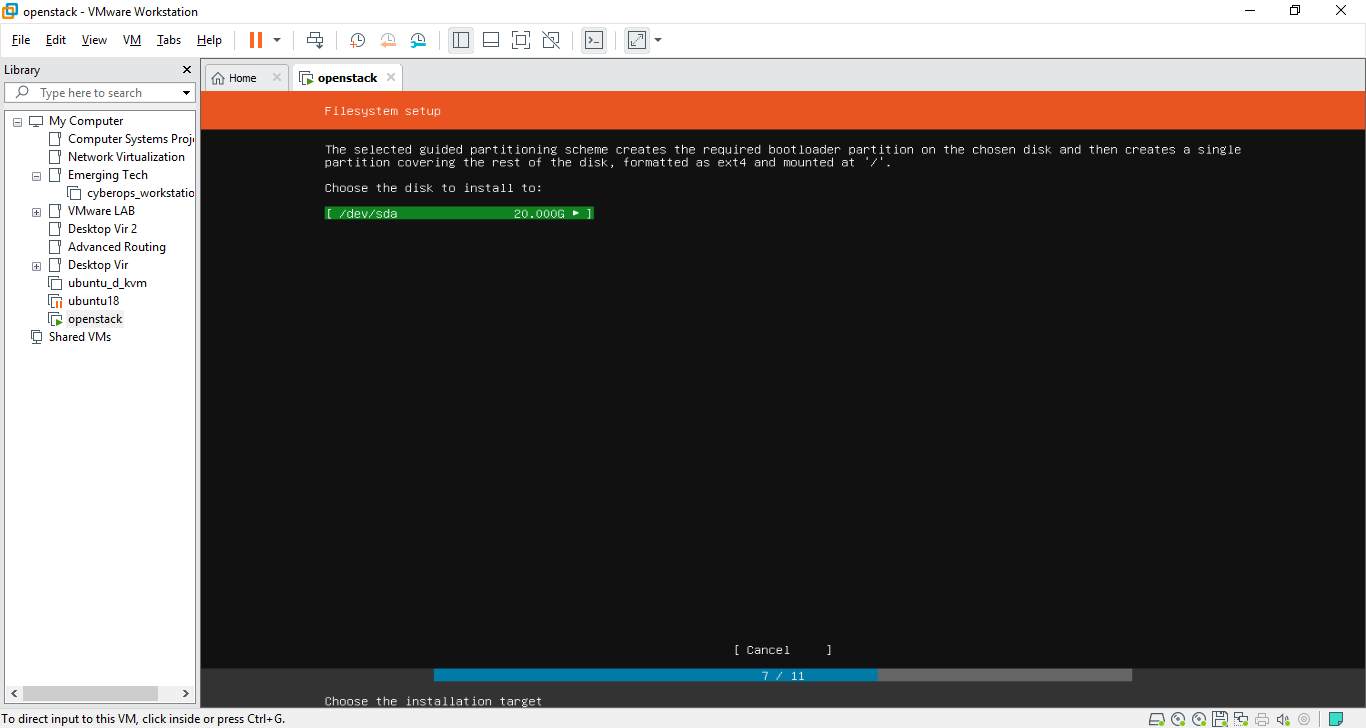
1. Leave the mirror address as default



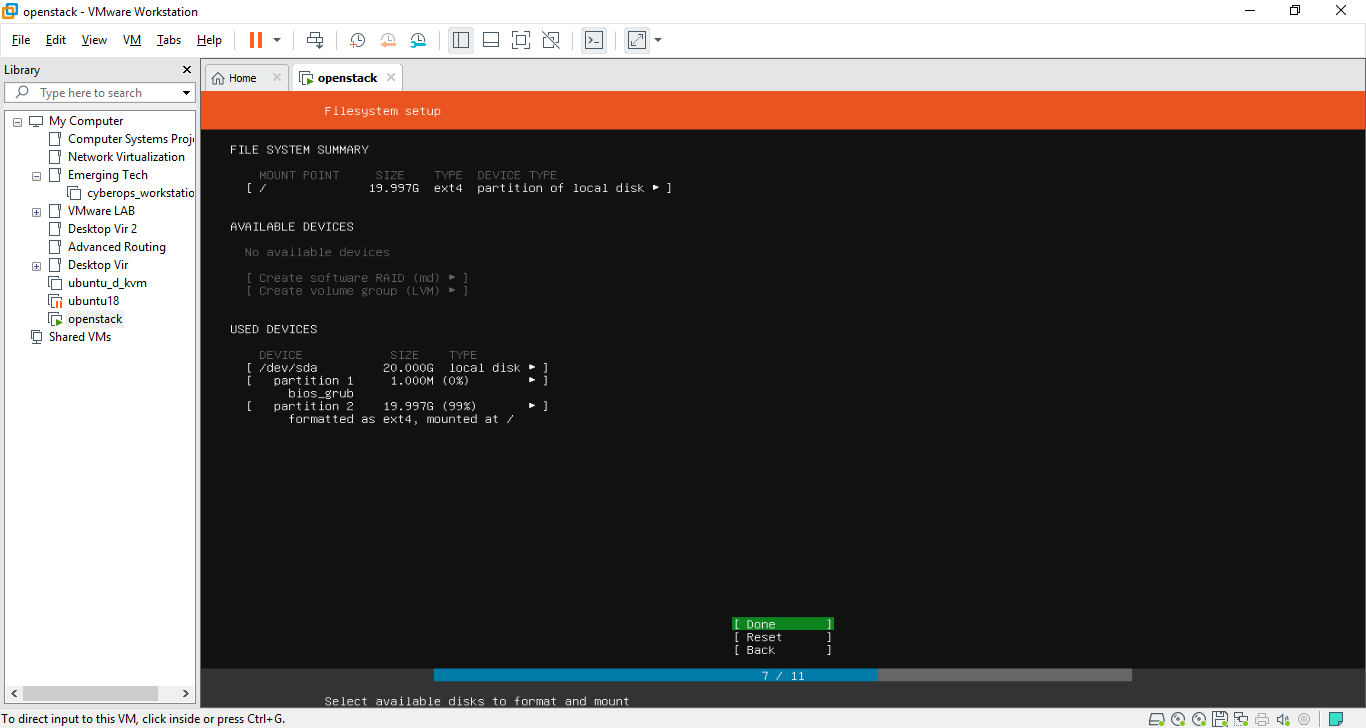
1. Use An Entire Disk for this setting.



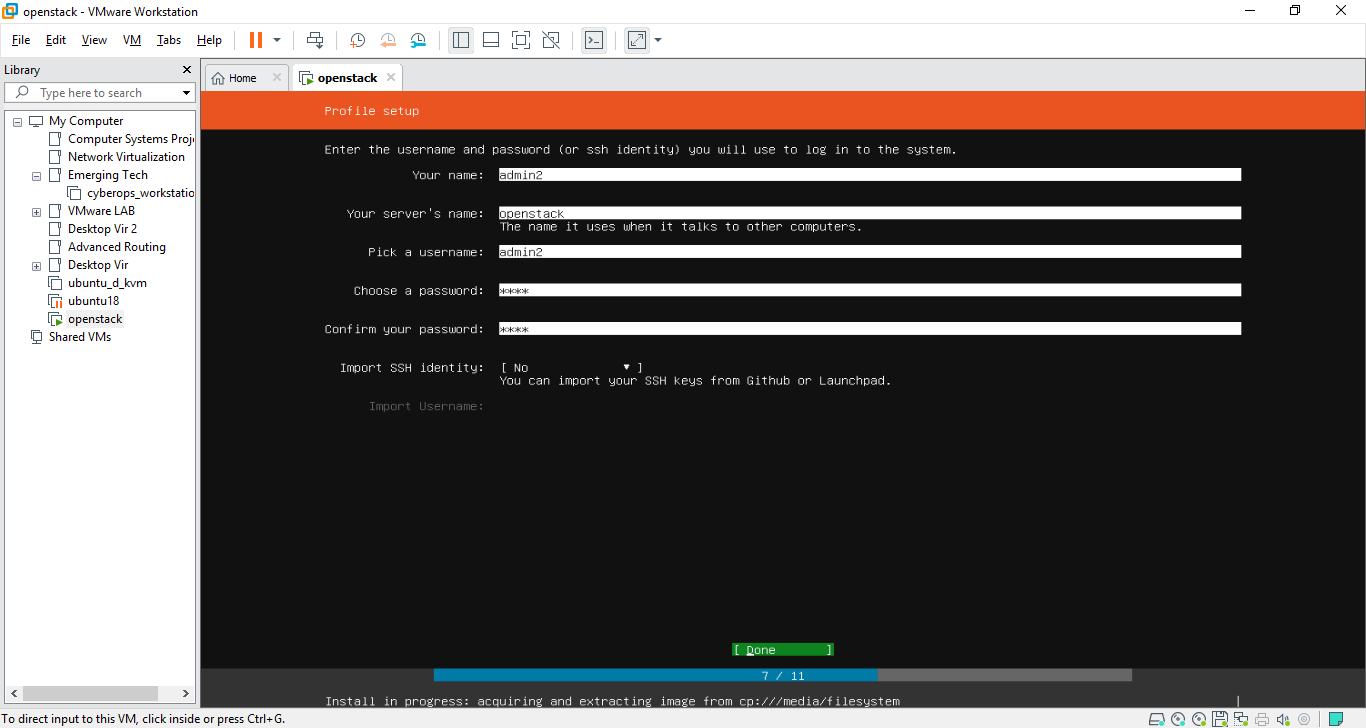
1. Select the partition you want to install.



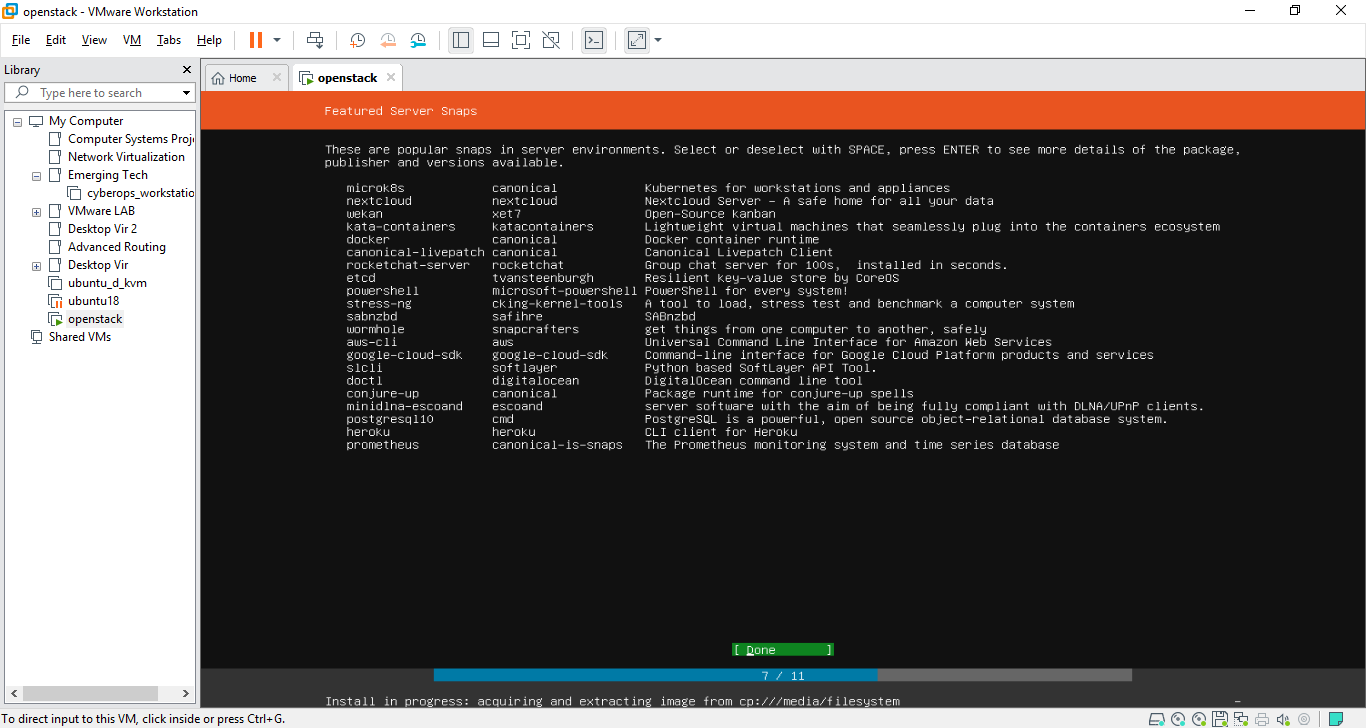
1. Review the File System summary



1. Enter the Username details for Ubuntu login.



1. Other components are not needed for now. Click Done to proceed and wait for the installation to finish.



## Configuration

1. Log in once the installation is completed. Update and upgrade the Ubuntu version.



1. Enter dist-upgrade to finish the Ubuntu upgrade



1. Reboot the system



1. create DevStack deployment user. 
2. Enable sudo privileges for this user without need for a password



1. Logout and login as stack user.

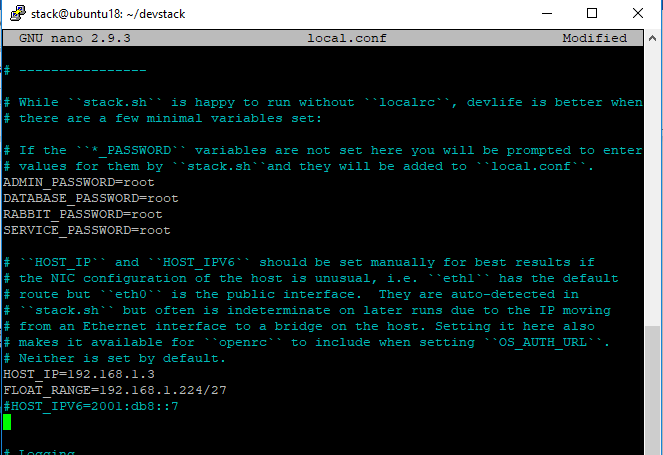
Install Ubuntu git component if it does not exist.



1. Clone Destack deployment code from Github 
2. Change directory to devstack. Create a local.conf file with 4 passwords and Host IP address.

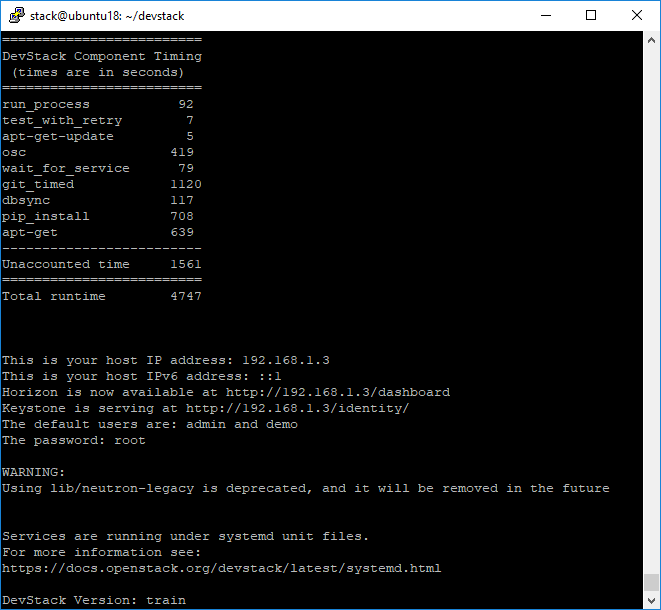


1. Change the passwords, IP address and Float Range as shown in the red boxes. You may use your own password and your host IP address.



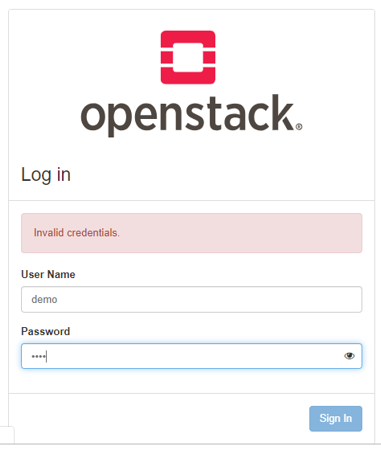
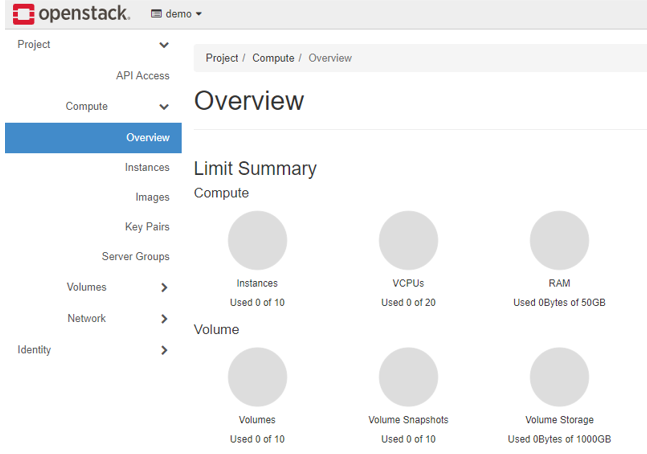
1. Now that you have configured the minimum devstack configuration and to get started, hit the below command.



1. The installation will take about 30 – 60 minutes, depending on your internet speed. At the end of the installation, you should see similar screen below: 

## Testing and evaluation

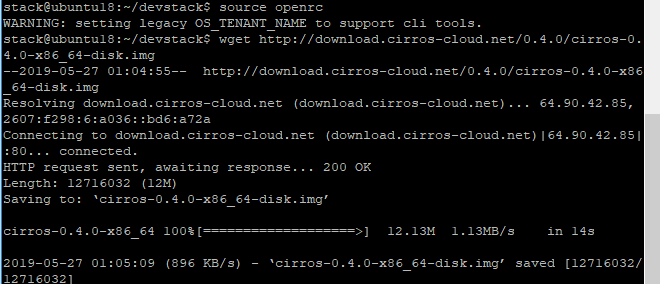
## OpenStack Dashboard

1. With internet browser, key in <http://192.168.1.6/dashboard>
2. Upon successful login, you will see the Overview page.

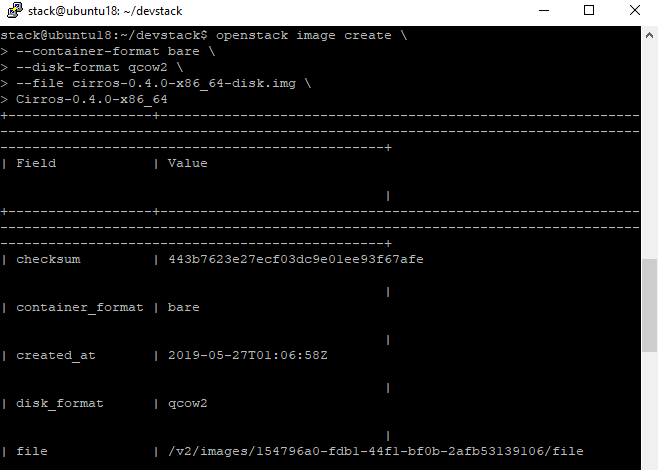
## Create VM image

1. To test, we will need an OS image for the new VM creation. To start, we will need to enter OpenStack command mode as shown in the red box. Type in “source openrc”.

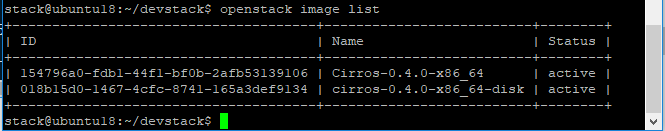
Then issue in wget command as shown in the blue box.



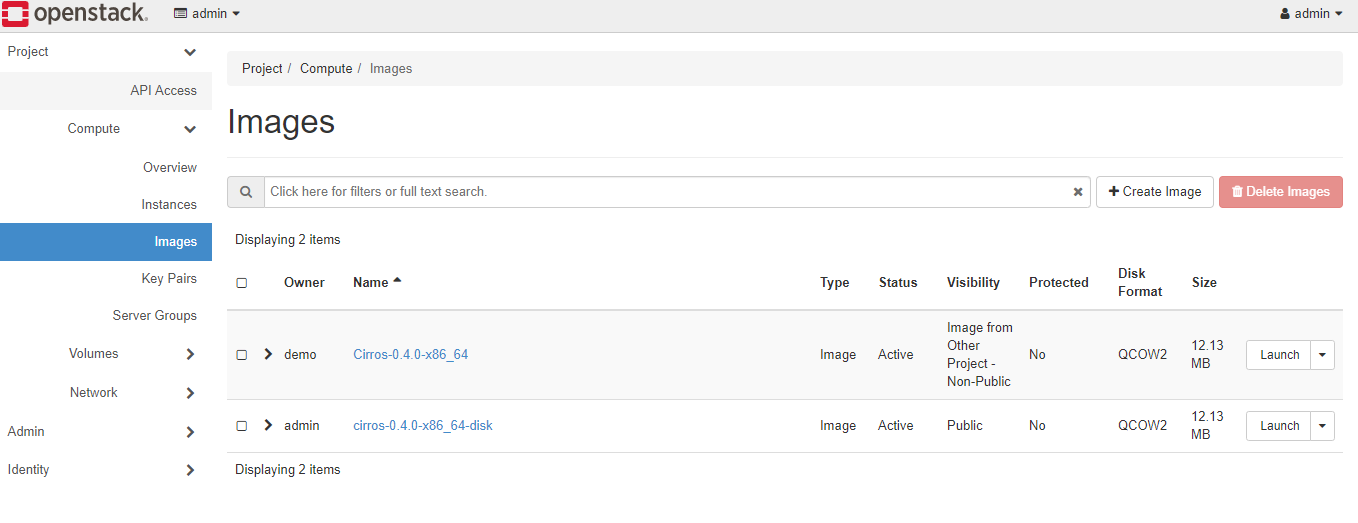
1. Issue command as below to add the cirros image.



1. Issue the command to list the images

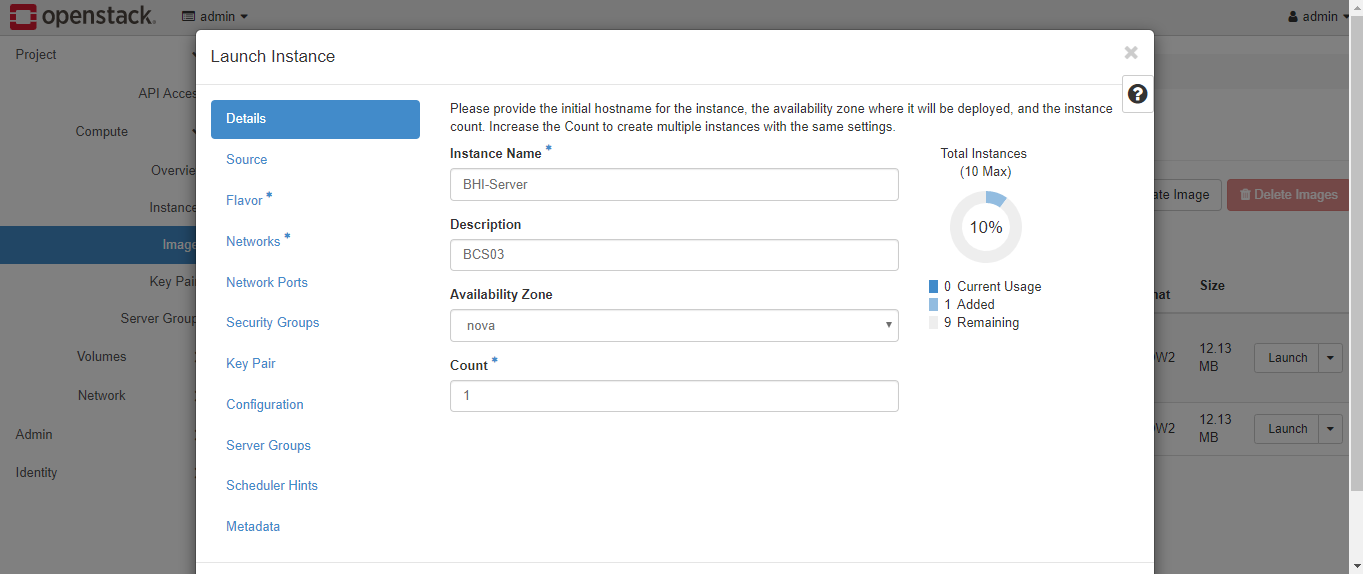


1. Back to OpenStack dashboard, and navigate to Images. You will see the images created as shown below.

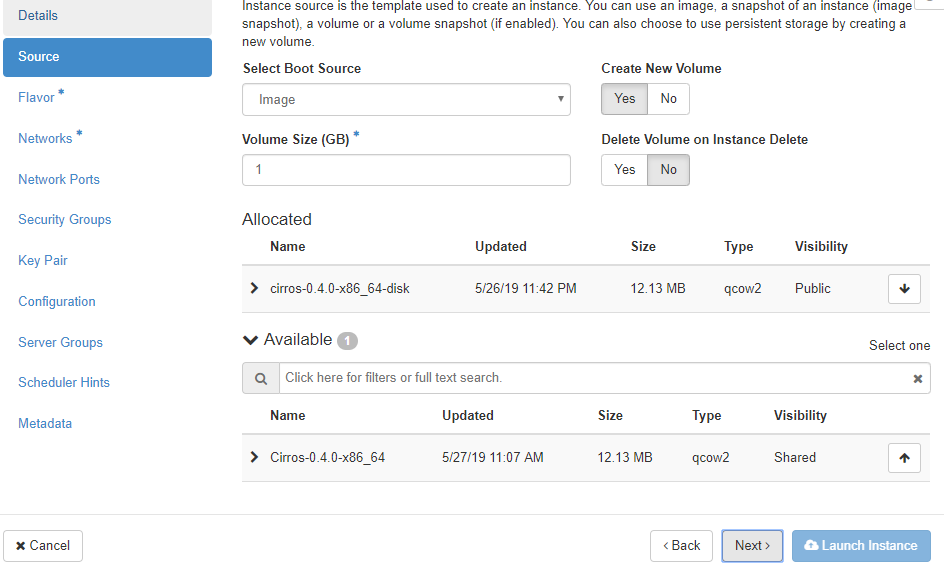


## Launch an Instance

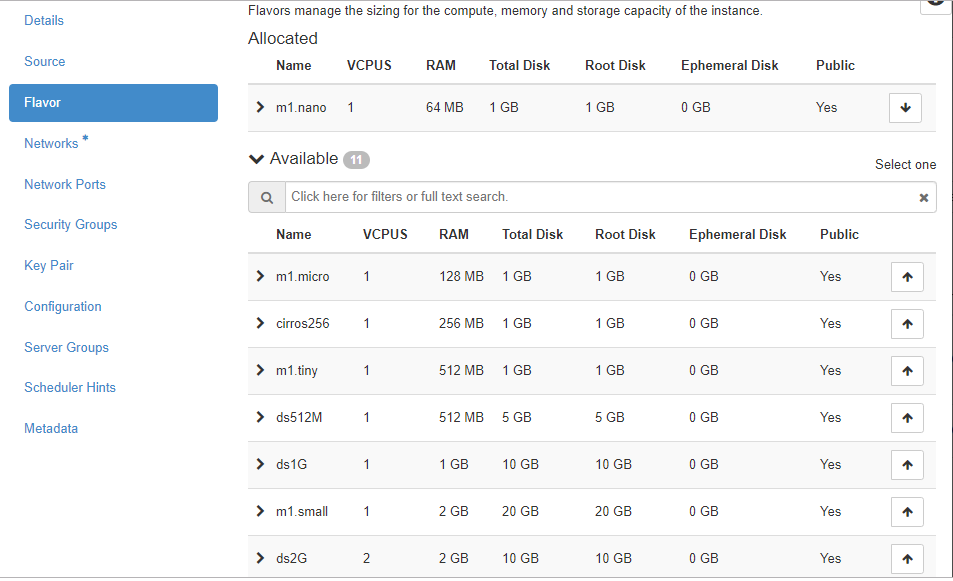
1. Click the Launch button on the image that was created earlier, as shown below. Name your instance.



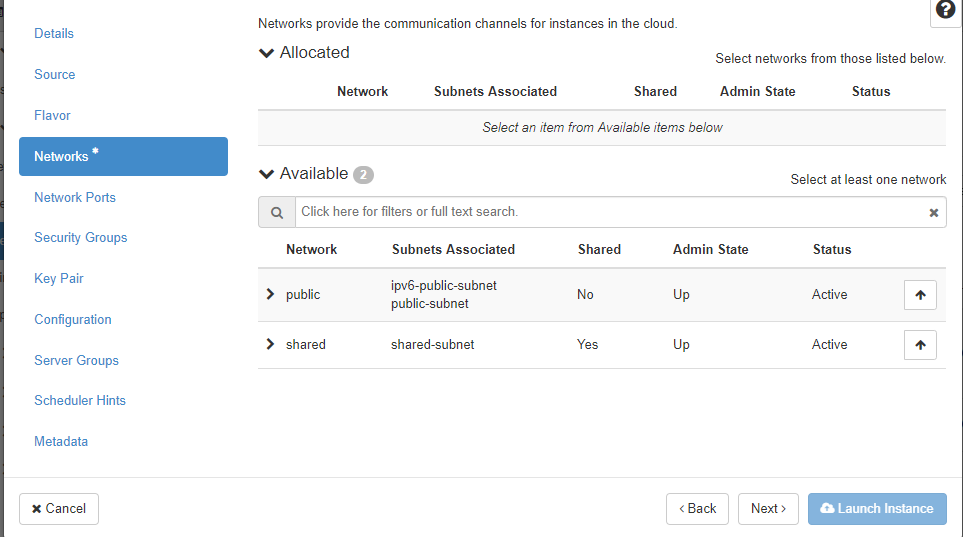
1. Select the source.



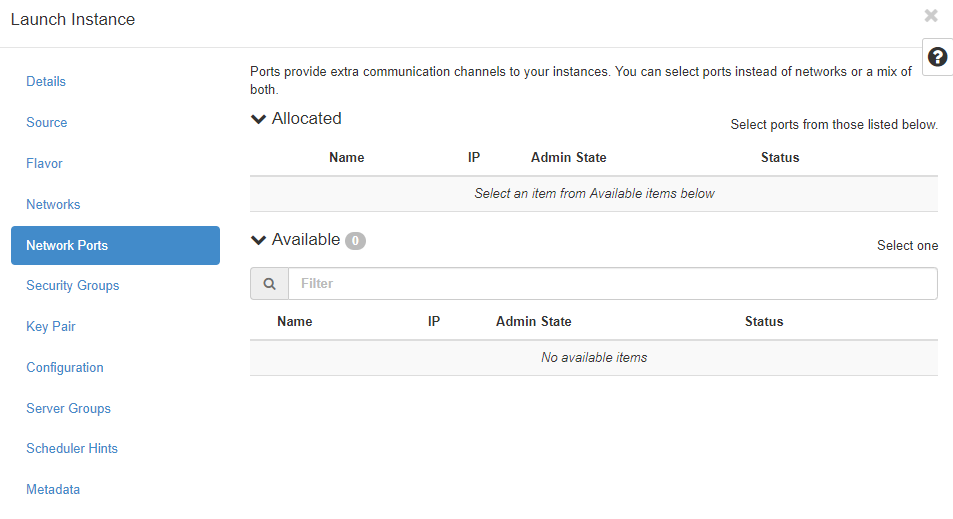
1. Select the Flavor (vCPU, RAM, Total Disk, Root Disk, etc)



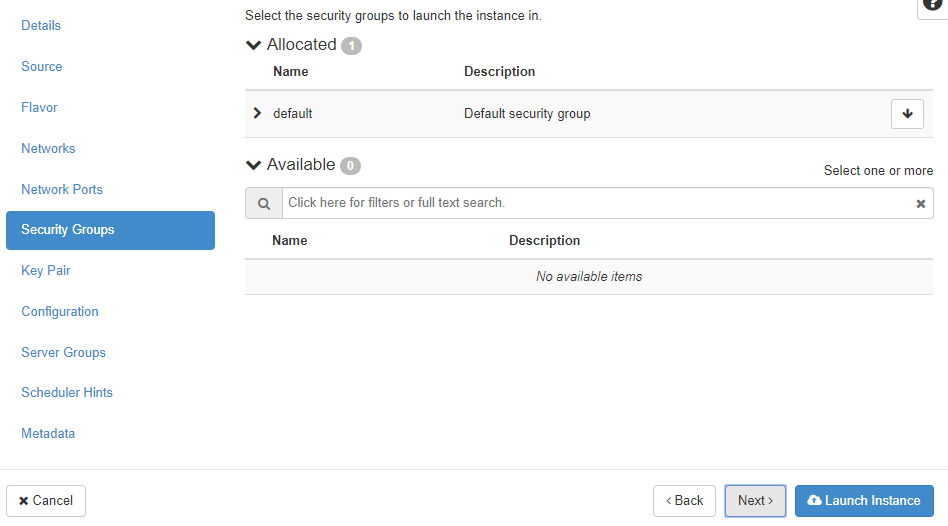
1. Assign the necessary network to this instance.



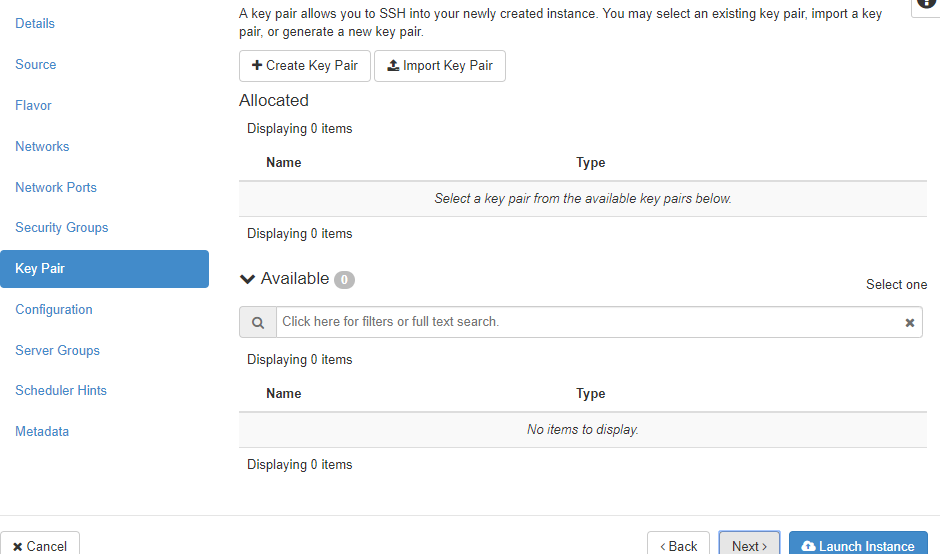
1. Select network ports if any



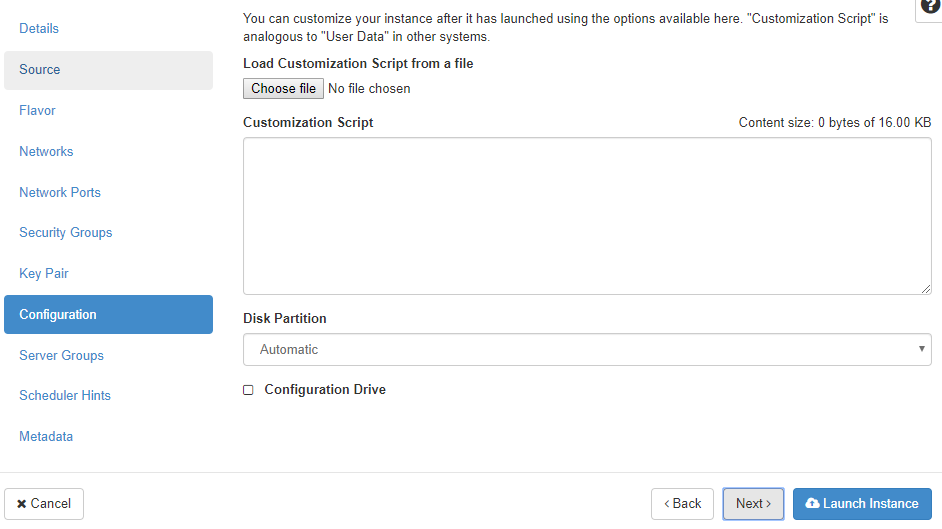
1. Proceed with default security group or other security group if any.



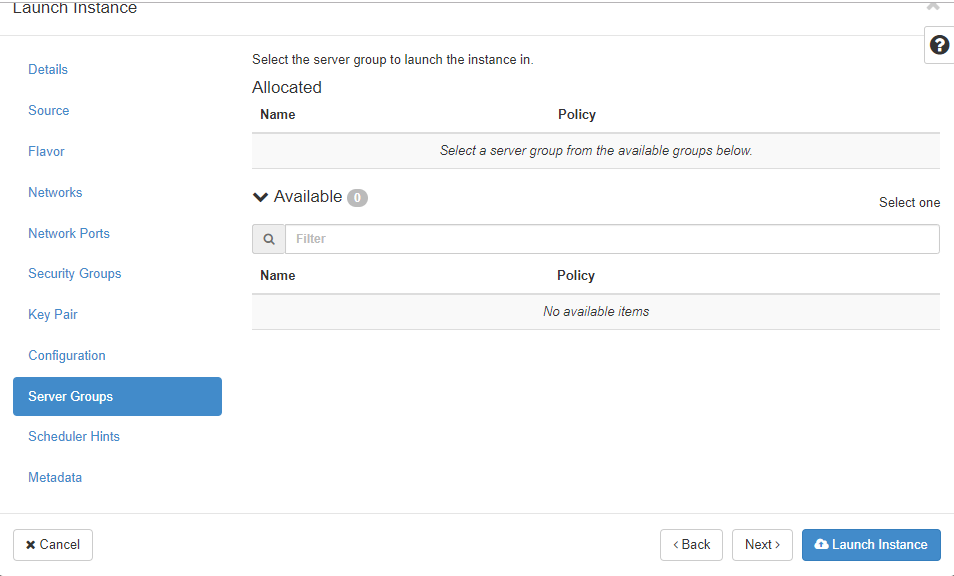
1. If you have an SSH key, you may insert here.



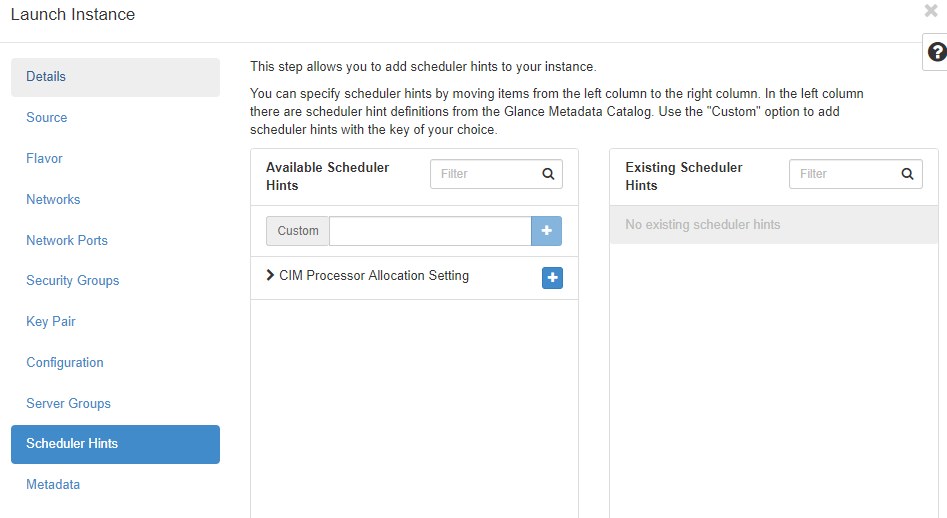
1. This screen allow you to customize your VM with scripting.



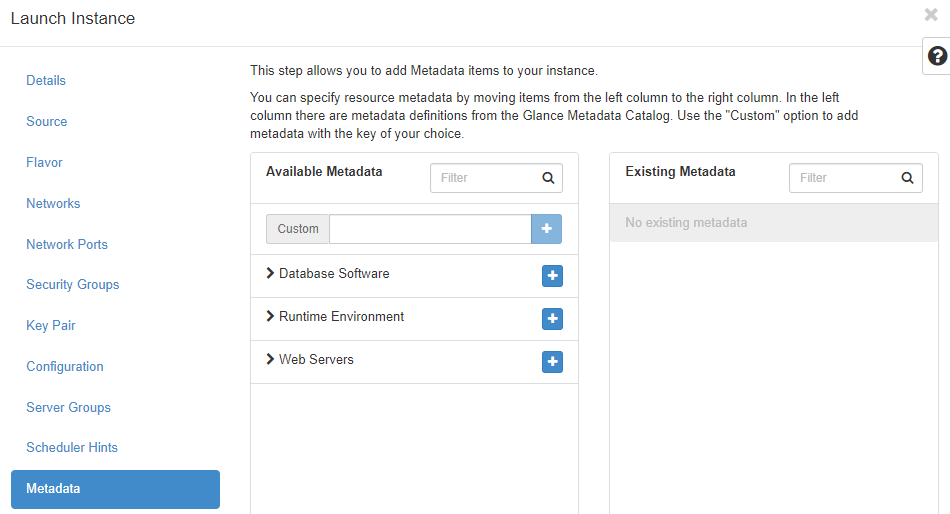
1. Select server group to launch the instance.



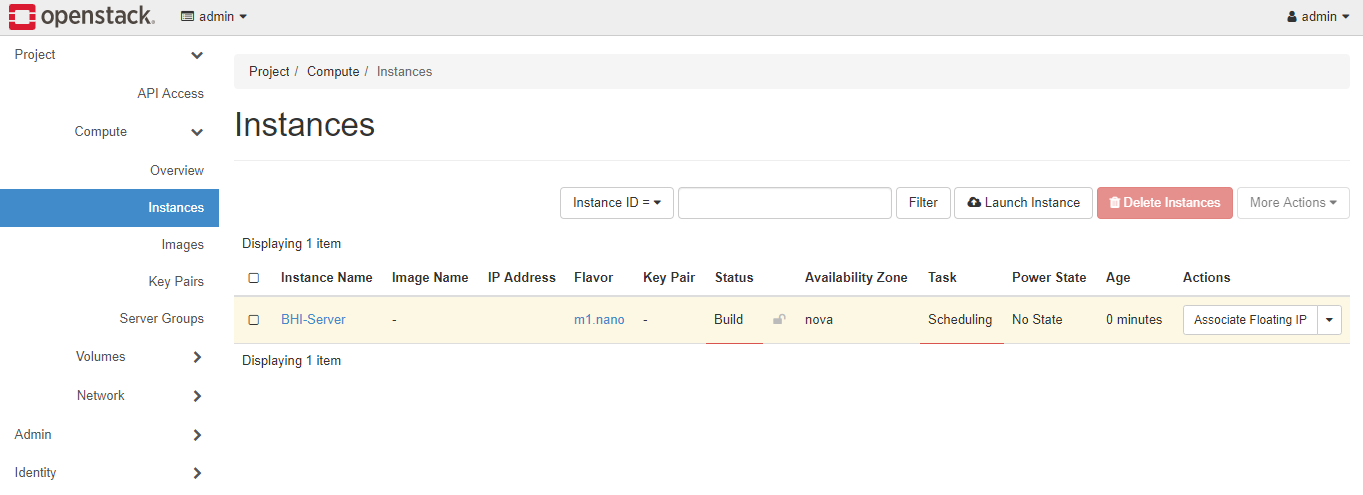
1. This page allows you to select Scheduler Hints.



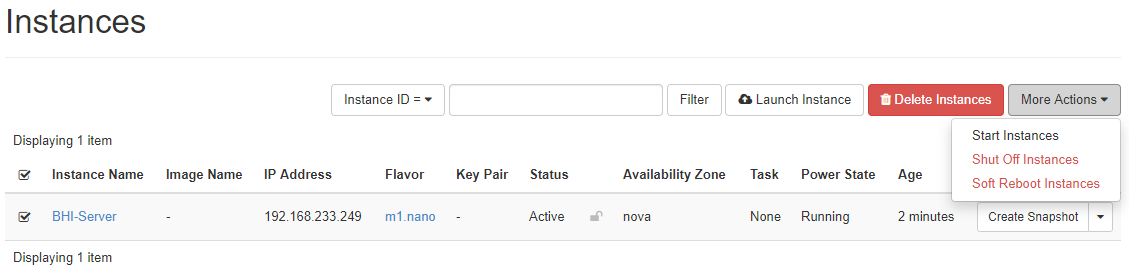
1. Finally, add Metadeta item if any.



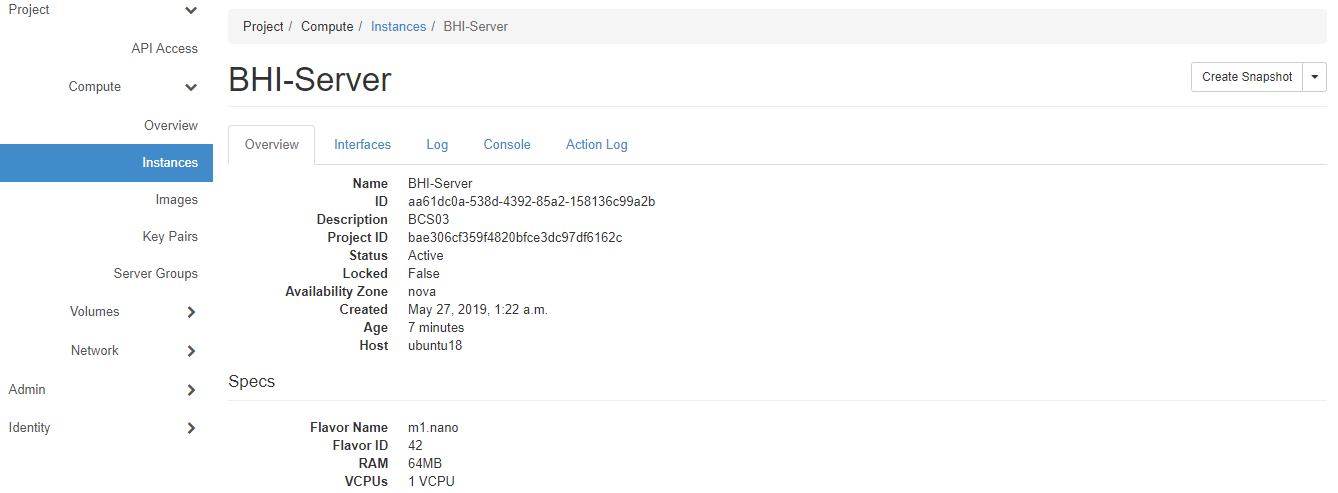
1. After the Launch Instance configuration completed, you will see the newly created instance appear in the list.



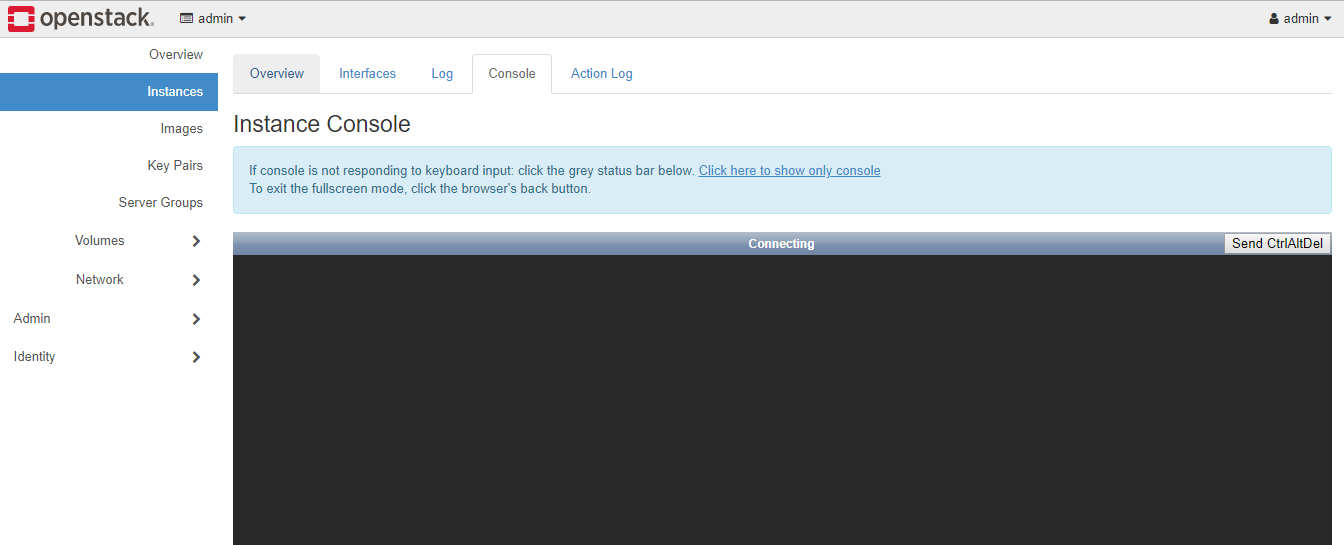
1. Wait for the instance to finish the task. Then start the instance.



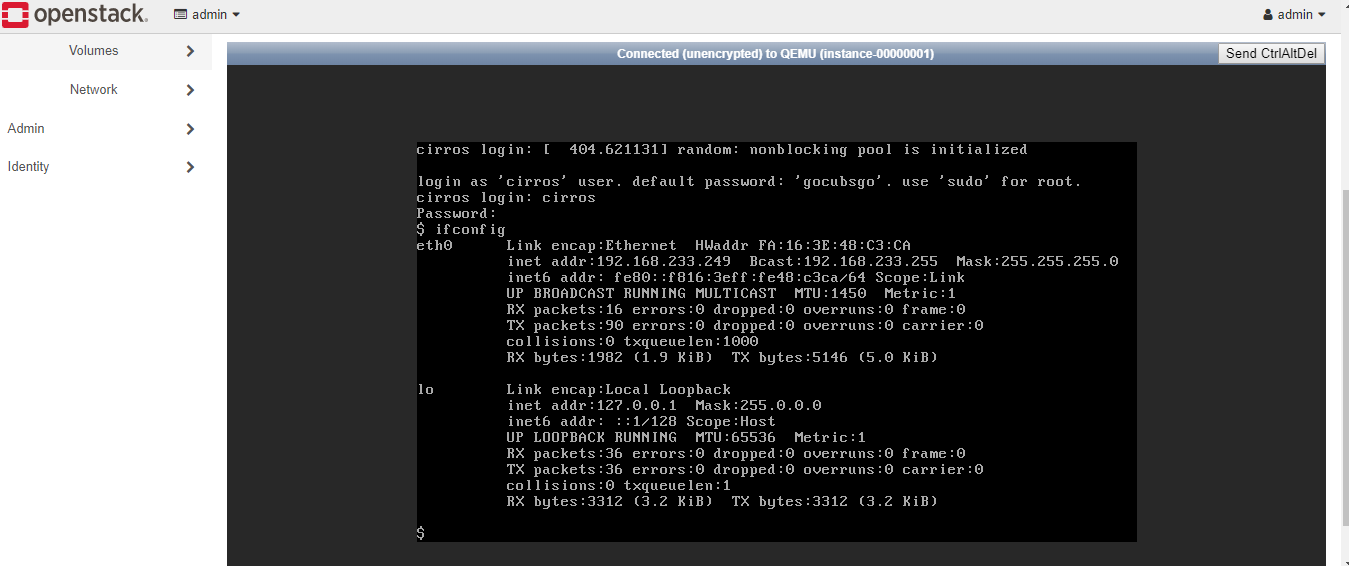
1. Navigate to Instance tab.



1. Click on Console

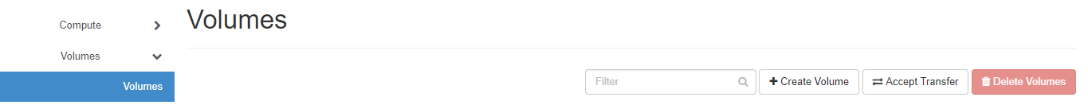


1. Wait for the VM to boot up. Then login with the username and password given. (VM was successfully booted with IP address as shown below)

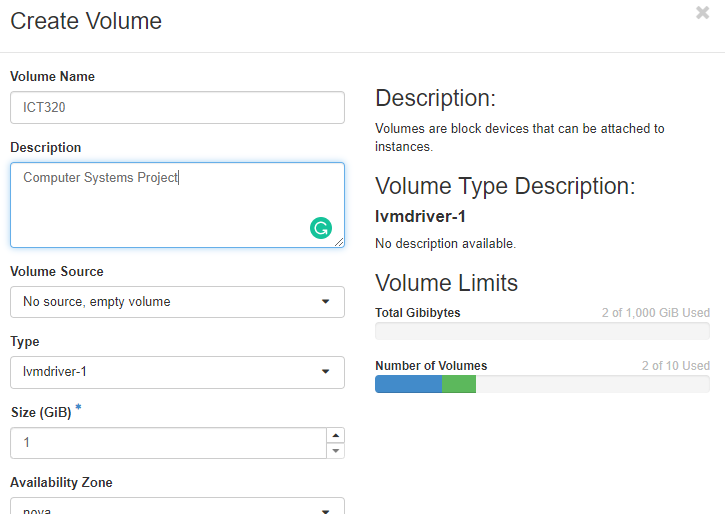


## Storage Volume

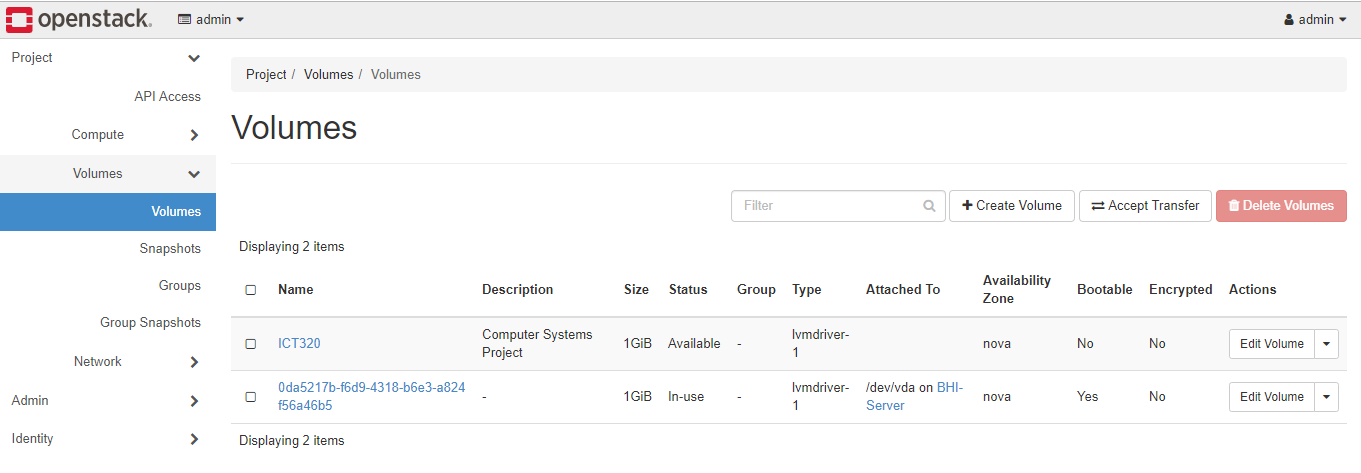
1. To create new volume, click on the Volumes tab. Then Create Volume as shown in the red box.



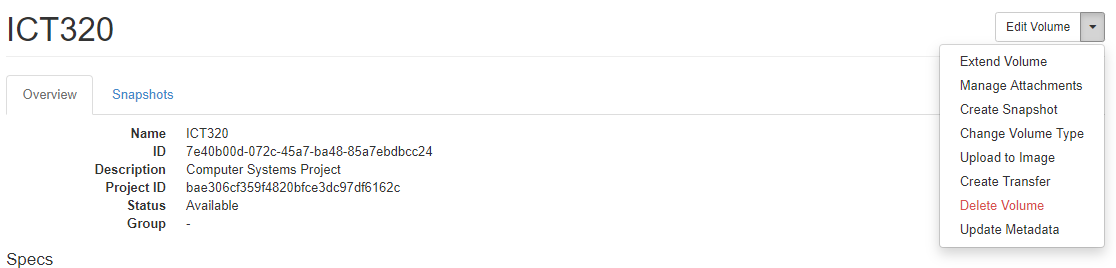
1. Key in the Volume details. You can enter the size from the available space.



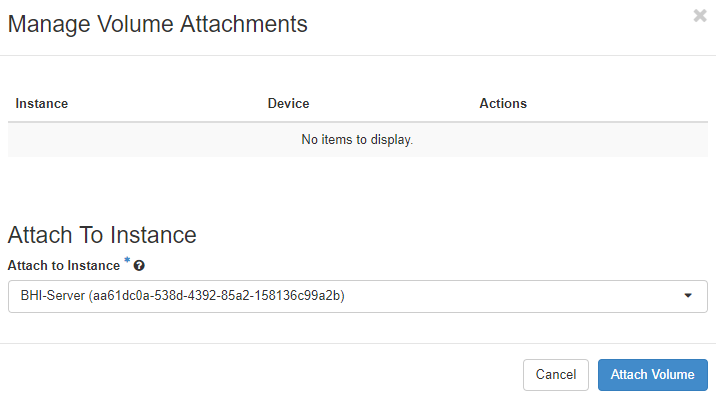
1. After you have finished the configuration, you will the volume in the list. Click the volume name as shown in the red box.



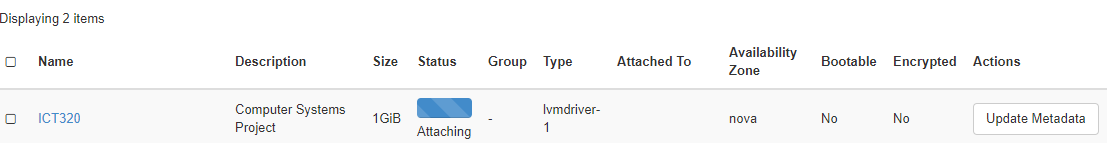
1. Click the drop down and click Manage Attachments



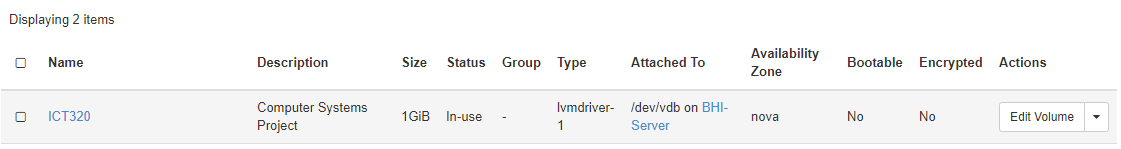
1. Under Attach To Instance, select the Cirros VM that was created earlier.



1. Wait for the status indicator.



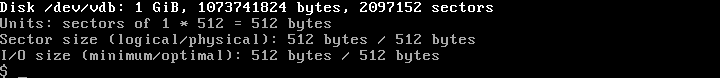
1. Volume is ready to use when you see the “In-use” status.



1. To verify, go to Cirros VM console. Enter fdisk command.



1. You will find the created volume as shown below.



## Reference

[a] Openstack (Mar, 2019) Openstack Newton Install Guide [Online] Available: <https://docs.openstack.org/newton/install-guide-ubuntu/>

[a] Openstack (April, 2019) Openstack Devstack Install Guide [Online] Available: <https://docs.openstack.org/install-guide/environment-packages.html>

END