IMPLEMENTATION OF IAAS PRIVATE CLOUD USING OPENSTACK CLOUD MANAGEMENT PLATFORM

BY

MD KAMRUL HASAN
ID: 132-25-307

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Computer Science and Engineering

Supervised By

Dr Syed Akhter Hossain
Professor and Head
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised By

A B M Moniruzzaman
Lecturer
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

APRIL 2015

©Daffodil International University

APPROVAL

This Project titled **“Implementation of IaaS Private Cloud using OpenStack Cloud Management Platform**”, submitted by Md. Kamrul Hasan to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 21 April, 2015.

BOARD OF EXAMINERS

____

Dr Syed Akhter Hossain **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Dr Sheak Rashed Haider Noori **Internal Examiner**
Asstt. Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

A.H.M. Saiful Islam **Internal Examiner**
Asstt. Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Dr Muhammad Shorif Uddin **External Examiner**
Professor and Chairperson
Department of Computer Science and Engineering
Jahagirnagar University

DECLARATION

I hereby declare that, this project has been done by us under the supervision of **Dr Syed Akhter Hossain, Professor and Head, Department of CSE** and co-supervised by **A B M Moniruzzaman, Lecturer, Department of CSE** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

____

Dr Syed Akhter Hossain
Professor and Head
Department of CSE
Daffodil International University

Co-Supervised by:

____

A B M Moniruzzaman
Lecturer
Department of CSE
Daffodil International University

Submitted by:

____

(Md. Kamrul Hasan)
ID: 132-25-307
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First I express my heartiest thanks and gratefulness to almighty Allah for His divine blessing makes me possible to complete this project successfully.

I fell grateful to and wish my indebtedness to **Dr Syed Akhter Hossain, Professor and Head**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of Computer Science influenced me to carry out this project .His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **A B M Moniruzzaman** Lecturer**,** Department of CSE, for his kind help to finish my project and also to other faculty member and the staff of CSE department of Daffodil International University.

I would like to thank my entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

In the field of cloud computing, infrastructure as a service (IaaS) is one of the most vital parts for building any types of cloud. The popularity of cloud management software is increasing day by day for smooth operation and administration of private cloud, public cloud or hybrid cloud. There are several open-source and commercial cloud management software exists in the market. Although the installation and configuration of those open source cloud management software is really difficult compare to commercial cloud management product nevertheless it gains a lot of popularity in the research and production network because of its free of cost, downloadable source code, community support etc.

OpenStack is one of the most widely open-source cloud computing platforms that support all types of cloud environments. It provides an Infrastructure-as-a-Service(IaaS) solution through a variety of services. The aim of implementation of OpenStack cloud management software is to build a private cloud in a virtual environment. To create the virtual environment, we use VMware workstation for deploying the virtual machine and then we install the Ubuntu operating system to install OpenStack on top of that. Since the installation of OpenStack cloud management software t is really a challenging task that is why we try use both manual configuration as well as scripting to make easy installation. After the proper installation of OpenStack we can define different types of service, allocate different resources using web interface as well as command line interface.

In this project the installation steps of OpenStack will help any organization like research institutions, government organizations, universities, cloud service providers who are interested to build their cloud in the test environment before building their real life cloud in the production network.

TABLE OF CONTENTS

CONTENTS **PAGE**

CHAPTER 1
INTRODUCTION

This is the introductory chapter of the project. In this chapter I will first describe the project working area. Then I will mention and provide brief discussion about the problem of the project goals and the methodology for doing the project. Finally, the short description the project structure will be mentioned at the end of the section.

1.1 Background

According to NIST, “Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [1]. Cloud computing is a method for delivering different kinds of services over the Internet. It can be a software application (e.g., Gmail) delivered over the Internet. It can be a platform (e.g., Google App Engine) on top of which users can write an application. It can be an operating system (e.g., Amazon EC2 provides OS) where users can create their own platform and software. All these services can be accessed through an Internet connection [2]. Cloud computing solves the demand of computing resources (e.g., CPU, RAM, etc) for processing jobs. In recent years, the demand for processing jobs has increased to hundreds or thousands of CPU cores.

Infrastructure-as-a-Service (IaaS) cloud is one of the models of cloud computing used to deliver computing resources, and this is accomplished by provisioning virtual resources to customers. For example, if a person wants to run an online business, there is no need to set up his/her own IT infrastructure. Instead he/she can rent a virtual machine from Amazon and provide the service [3]. In addition to delivering the virtual resources, IaaS brings another advantage of “pay-per-use”. The pay-per-use model can be easily understood through comparison with an electricity payment scheme, paying only for the amount of electricity consumed. Similarly, IaaS allows a customer to pay only for the used computing resource. This helps the business as it no longer requires investing capital in IT infrastructure and allows an enterprise to focus more on the business growth instead of thinking about over-provisioning and under-provisioning of the computing resource [1].

The actors involved in cloud computing related to IaaS are called cloud stakeholders. The cloud stakeholders are cloud providers and cloud users. The cloud providers are responsible for setting up the IT infrastructure and the cloud users use the infrastructure provided to complete their tasks. The private-IaaS-cloud model is selected where the cloud infrastructure is provided by a single organization for its different business units. The cloud management software is a piece of software that converts the existing physical IT infrastructure into a cloud infrastructure that supports the private-IaaS-cloud. The number of commercial cloud management software increases as the popularity of private-IaaS-cloud increases, such as VMware vCloud, CloudForms, etc [4].

1.2 Problem of the Project

Numerous contributions have been made by the open-source community related to private-IaaS-cloud. Some of the open-source cloud management software are Eucalyptus, Nimbus, CloudStack and OpenNebula [5,6]. The number of cloud management software is increasing day-by-day. One of the major problems for cloud consumers is choosing the cloud management software, as features of the cloud management software vary significantly. This makes it difficult for the cloud consumers to choose the software based on their business requirements. For example, if a cloud customer thinks that power management is important for his/her business, the cloud consumer have to read the information provided in the software website to identify whether the power management is provided with the software or not.

1.3 Goal of the Project

The major goal of the project is to implement OpenStack IaaS cloud management software in a virtual environment using command line interface and using web browser.

1.4 Methodology

The purpose of this project is to study the characteristics of cloud computing and how OpenStack can be implemented as a private cloud. The following steps are as follows:

1. The first step of the methodology involves literature study on the current industry trends of cloud computing.
2. Second step includes installation of VMware workstation, creation of virtual network, creation of virtual machines, installation of Ubuntu, update and upgrade of Ubuntu operating system, installation of NTP server, installation of MySql Server etc.
3. Third step includes installation and configuration of different OpenStack modules on top of Ubuntu operating system hosted on virtual machines.
4. Finally, we will configure the OpenStack according to user needs using web browser.

1.5 Intended Audience

The project work deals with the installation of VMware workstation, creation of virtual machines, installation of Ubuntu, update and upgrade of Ubuntu operating system, installation of NTP server, installation of Mysql Server Operating system and installation of different OpenStack software. So, a person with a background of information technology along with having strong knowledge of any linux operating systems, scripting knowledge and networking concepts would be an appropriate audience for this project works. Moreover, the report will be suitable for students, researchers and industrials professional and general audience who have keen interest in implementing OpenStack as a IaaS.

1.6 Limitations

The focus has been on implementing OpenStack cloud management system in a private network and in a virtual environment. The cloud management system is implemented and configured with couple of services because of resource constraints. But the full implementation in the public cloud can be done as a future work.

1.7 Project Structure

- Chapter 1 is the introduction and background of the thesis. The rest chapters are structured as follows:

- Chapter 2 contains an overview of cloud computing, different types of cloud services and benefits and finally different cloud deployment model existing in the market.

- Chapter 3 starts with the introduction of OpenStack and then why IaaS is needed in cloud computing. In the chapter the architecture of the OpenStack is shown and finally components are briefly described.
- Chapter 4 contains the step by step implementation of OpenStack in a virtual machine environment.

- Chapter 5 contains conclusion of this chapter.

CHAPTER 2
SIMILAR STANDARDS AND TECHNOLOGIES

This chapter describes the relevant cloud management software which will help us to acquire the knowledge of other existing cloud management software in the industry. By studying these standards and technologies, we will be able to understand the advantages and disadvantages of different cloud management software. There are mainly two types of cloud management software exists in the market such as open-source and commercial product. In the open-source software, we have OpenStack, Eucalyptus, CloudStack OpenNebula, Ganeti etc. On the other hand in the commercial product there are VMware vCloud, HP Cloud Matrix, Microsoft System Center etc. Couple of cloud management software is briefly explained below.

2.1 Eucalyptus

Eucalyptus is open source software for building private clouds that are compatible with AWS APIs. Eucalyptus is the acronym for Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems [23]. It started as a research project at University of California, Santa Barbara, USA and founded company in 2009 to commercialize the project. Eucalyptus cloud software pools together compute, network, and storage resources within your IT environment to create on-demand, self-service private cloud resources. It can dynamically scale up or down depending on application workloads and is uniquely suited for enterprise clouds, delivering production-ready software. The benefits of this open source software for private clouds are highly efficient scalability, organization agility, and increased trust and control for IT [24][30]. The strengths and weaknesses of Eucalyptus are mentioned below.

Eucalyptus strengths are mentioned below [27][30].

- Excellent commercial support
- Fault-tolerance
- Offers a hybrid-cloud solution with AWS

Eucalyptus weaknesses are mentioned below [27][30].

- It is configurable but not very customizable
- Complex installation requirements
- Codebase is complicated

2.2 CloudStack

Apache CloudStack is open source software designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure as a Service (IaaS) cloud computing platform. CloudStack is used by a number of service providers to offer public cloud services, and by many companies to provide an on-premises (private) cloud offering, or as part of a hybrid cloud solution [25][26]. CloudStack currently supports the most popular hypervisors: VMware, KVM, XenServer, Xen Cloud Platform (XCP) and Hyper-V [26]. The strengths and weaknesses of CloudStack are briefly mentioned below.

CloudStack strengths are mention below [26][27].

- Well-rounded GUI
- Stack is fairly simple
- Customization of the storage backend

CloudStack weaknesses are mentioned below [26][27].

- Very GUI centric
- Single java core
- AWS integration weak

2.3 OpenNebula

OpenNebula is a cloud computing platform for managing heterogeneous distributed data center infrastructures. The OpenNebula platform manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service. OpenNebula is free and open-source software, subject to the requirements of the

Apache License[28]. OpenNebula provides features at the two main layers of Data Center Virtualization and Cloud Infrastructure [29][30][31].

The strengths of OpenNebula are mentioned below[32][33].

- Amazon API
- Fully platform independent
- On-demand provision of Virtual Data Centers
- Powerful CLI

The weakness of the OpenNubula is mentioned below.

- The development is managed by a small group in Europe

2.4 OpenStack

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface [34][35].

The strengths of OpenStack are mentioned below[33].

- Single codebase
- Excellent APIs
- Growing community
- Corporate support
- Excellent for large deployment

The weaknesses of the OpenStack are mentioned below[34][35][36].

- Young codebase
- Web interface is young and limited
- Uncertain future

CHAPTER 3
OVERVIEW OF CLOUD COMPUTING

This is the third chapter of the project report. In this chapter I will first introduce the definition and different attributed of cloud computing. Then I will discuss various types of cloud service models exist in the market. Finally, the short description the different cloud delivery models will be mentioned at the end of the section.

3.1 Cloud Computing

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configuration computing resources(e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.[1]

The definition of cloud computing is based on five important attributes such as

- Multitenancy (shared resources)
- Massive scalability
- Rapid Elasticity
- Pay as you go and
- On-demand self-service of resources.

The brief description of mentioned attributes is given below.

1. **Multitenancy :** In a multi-tenant model, cloud computing is based on a business model where computing resources such as storage, processing, memory, network bandwidth, virtual machines etc. are shared. In this model multiple users are sharing the resources in the network level, host level and application level.
2. **Massive scalability:** In cloud computing a company will have the ability to scale to tens of thousands of systems along with the ability of massively scale of

bandwidth and storage space. This available capability can be accessed through standard mechanism that promotes us by heterogeneous thin or thick cline platforms such as mobile phones, laptops, PDA etc.

3. **Elasticity of resources:** The cloud users can rapidly increase and decrease their computing resources whenever it is needed. In addition to that they can release the resources for the other cloud users when it is no longer needed to them.[7]

4. **Pay as you go:** The cloud users will pay for only the resources they actually consume. The cloud system will automatically control monitor and report to the provider and consumer of the utilized services.
5. **Self-provisioning of resources:** Users can self-provision computing resources, such as processing capability, software, storage and network resources without requiring human interaction with each service provider.

3.2 Cloud Computing Models

Cloud computing can be divided into three service delivery models [7][8] such as

- Software as a Service(SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service(IaaS)

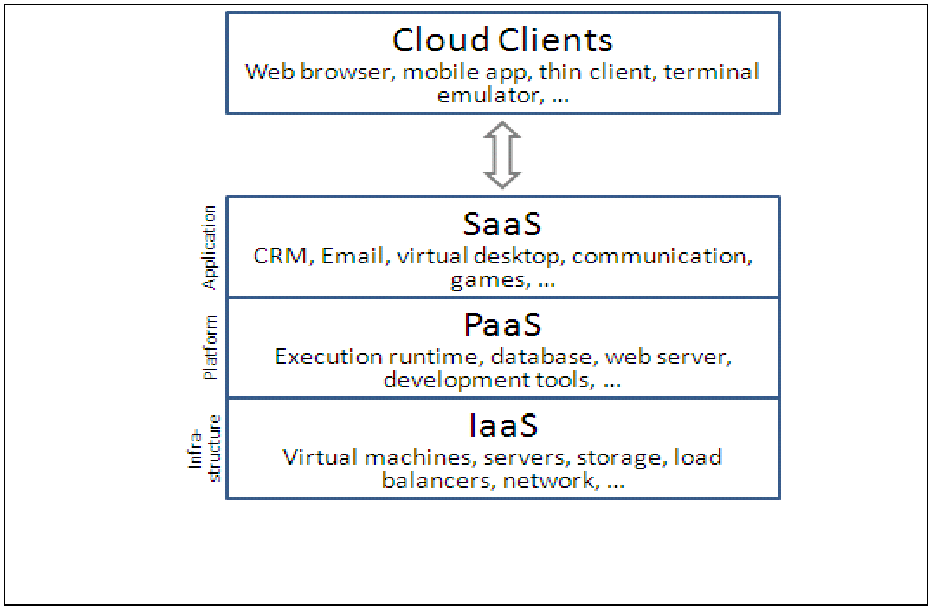

Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
SaaS
CRM, Email, virtual desktop, communication, games, ...
PaaS
Execution runtime, database, web server, development tools, ...
IaaS
Virtual machines, servers, storage, load balancers, network, ...
Application
Platform
Infra-structure

Figure 3. 1: Cloud Computing Models.[9]

3.2.1 Software as a Service (SaaS)

In this Software as a Service delivery model users are provided access to application software and databases. These application software and associated data are hosted to the third party cloud service provider. The responsibilities of the cloud providers are to manage the cloud infrastructure and platform where the application runs. In this model cloud service providers install and operate the software and on the other hand cloud users access the software on demand using the cloud clients such as browser on a computer or mobile device.

The key benefits of Softwae as a Service are given below.

- SaaS enables the organization to reduce IT costs by outsourcing the hosting, maintenance and support of the software to the cloud provider[10][11] SaaS enables software vendors to control and limit use, prohibits copying and distribution, and facilitates the control of all derivative versions of their software.
- There is no requirement of any hardware for the deployment of Software as a service. This deployment model can work on top the existing Internet access

infrastructure.[10][11]

- There is no need to install any patches to download or install in the client side. The SaaS provider manages all updates and upgrades. [11][12]
- Accessibility can easily be achieved just only using browser and an internet connection to access the SaaS application, which can therefore be made available on a wide range of desktop and mobile devices.[10]
- Management of a SaaS application is supported by the vendor from the end user perspective, whereby a SaaS application can be configured using an API, but SaaS applications cannot be completely customized.

3.2.2 Platform as a Service

Platform as a service is a delivery cloud service model where service provider offers a software development environment as a service to application developer. This service model can also be represents as a pre-defined ready-to-use environment typically comprised of already deployed and configured IT resources. Cloud users can get rid of burden of application development management and can lessen the cost of the services.[8][13] [21][22]

3.2.3 Infrastructure as a Service

In this cloud service model, service provider will provide physical or more often virtual machines along with other resources. This IaaS model is similar to utility computing in which the idea is to offer computing resources to the clients in the same way as utilities. In this model the service provider will have the control of the whole infrastructure. The important features of IaaS are as follows [14] [18][19].

Scalability

This is the ability to scale infrastructure requirements, such as computing resources, memory and storage based on usage requirements.

Pay as you go
In this feature the cloud users will have the ability of purchasing the exact amount of infrastructure required at any specific time.

3.3 Cloud Deployment Models

3.3.1 Public Clouds

The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services. The public cloud is hosted, operated and maintained by a third party vendor. The services of the public cloud are renders over a network that is open for public use.[1][9][20]

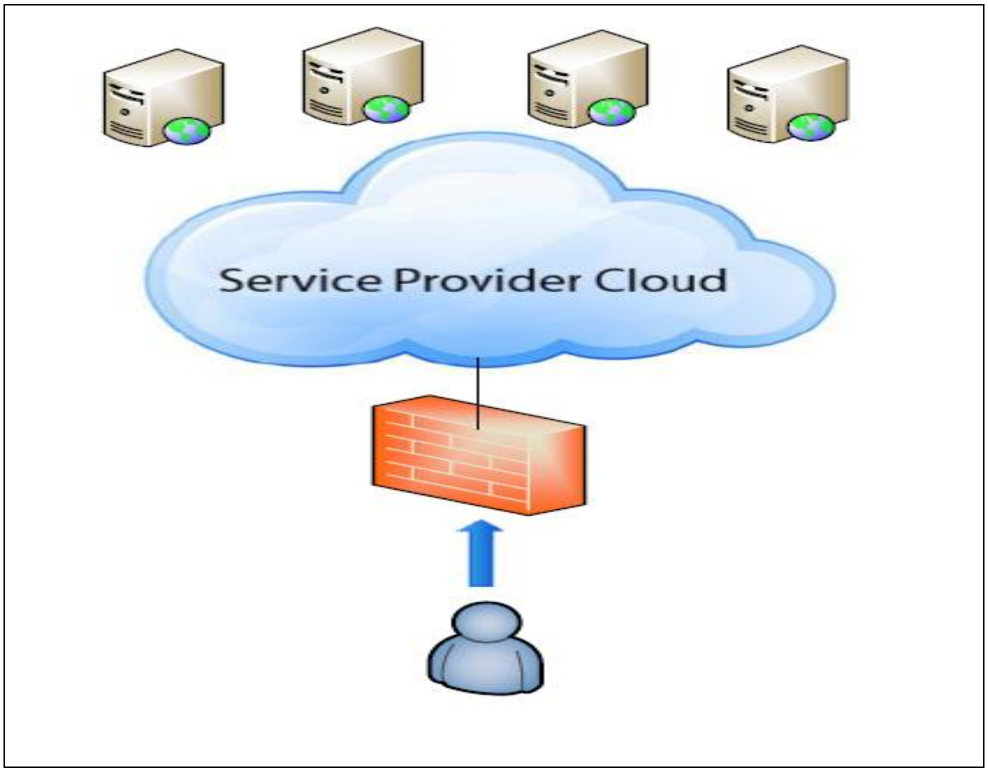


Figure 3. 2: Public Cloud

3.3.2 Private Clouds

The private cloud is a particular cloud model which is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise. In this model the pool of resources of the cloud is only accessible by a particular single organization and only the organization has the greater control and privacy on the resources.[15][20]

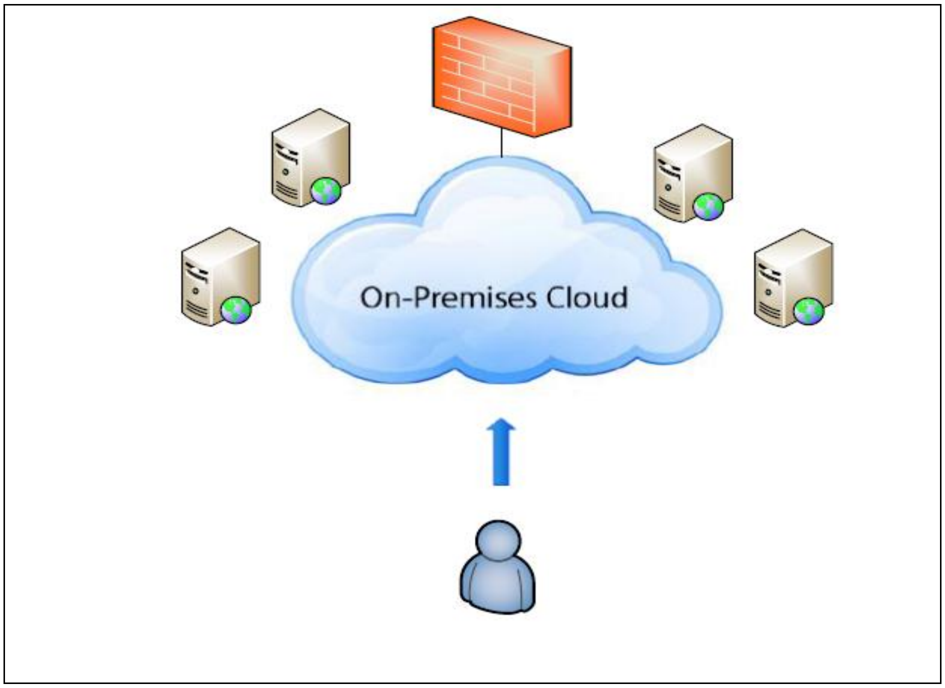


Figure 3. 3: Private Cloud

The features and benefits of private clouds therefore are mentioned below.

- **Higher security and privacy**: Public clouds services can implement a certain level of security but private clouds - using techniques such as distinct pools of resources with access restricted to connections made from behind one organization's firewall, dedicated leased lines and/or on-site internal hosting - can ensure that operations are kept out of the reach of prying eyes.
- **More control:** as a private cloud is only accessible by a single organization, that organization will have the ability to configure and manage it inline with their

needs to achieve a tailored network solution. However, this level of control removes some the economies of scale generated in public clouds by having centralized management of the hardware

- **Cost and energy efficiency:** implementing a private cloud model can improve the allocation of resources within an organization by ensuring that the availability of resources to individual departments/business functions can directly and flexibly respond to their demand. Therefore, although they are not as cost effective as a public cloud services due to smaller economies of scale and increased management costs, they do make more efficient use of the computing resource than traditional LANs as they minimize the investment into unused capacity. Not only does this provide a cost saving but it can reduce an organization's carbon footprint too.
- **Improved reliability:** even where resources (servers, networks etc.) are hosted internally, the creation of virtualized operating environments means that the network is more resilient to individual failures across the physical infrastructure. Virtual partitions can, for example, pull their resource from the remaining unaffected servers. In addition, where the cloud is hosted with a third party, the organization can still benefit from the physical security afforded to infrastructure hosted within data centers.

- **Cloud bursting:** some providers may offer the opportunity to employ cloud bursting, within a private cloud offering, in the event of spikes in demand. This service allows the provider to switch certain non-sensitive functions to a public cloud to free up more space in the private cloud for the sensitive functions that require it. Private clouds can even be integrated with public cloud services to form hybrid clouds where non-sensitive functions are always allocated to the public cloud to maximize the efficiencies on offer.

3.3.3 Hybrid Clouds

A hybrid cloud environment consisting of multiple internal and/or external providers is a possible deployment for organizations. With a hybrid cloud, organizations might run non-core applications in a public cloud, while maintaining core applications and sensitive data in-house in a private cloud.[20]

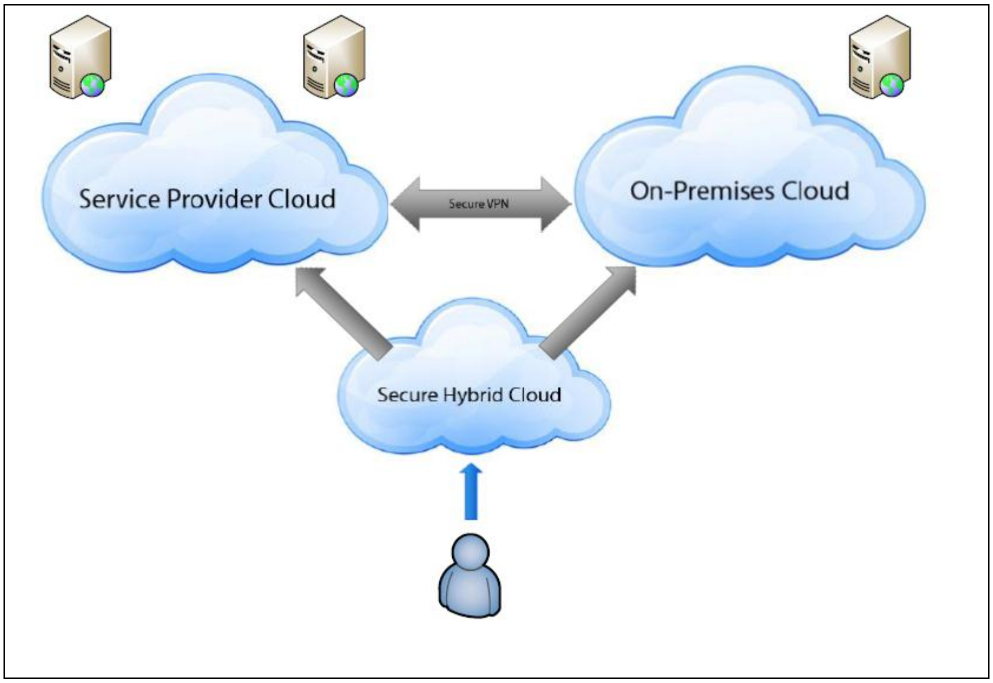


Figure 3. 4: Hybrid Cloud

CHAPTER 4
OPENSTACK OVERVIEW

This is the fourth chapter of the project report. In this chapter I will first provide a brief overview of OpenStack. Then I will discuss the necessity of Infrastructure of a service (IaaS). Then I show the different architecture of OpenStack. Finally, the short description the different OpenStack components will be described at the end of the section.

4.1 OpenStack Overview

OpenStack is a Cloud ecosystem that controls large pools of storage, compute, and networking resources throughout a datacenter. This is all managed with a dashboard (horizon) that gives the administrator control while yet empowering their users to provision resources through a web interface.

OpenStack consist of a global collaboration of developers and cloud computing technologist producing the ubiquitous open source computing platform for both public and private clouds. The OpenStack project aims to deliver solutions for all types of clouds by being simple to implement, massively scalable, and feature rich. The technology consists of a series of interrelated projects delivering various components for a cloud infrastructure solution.

OpenStack was founded by Rackspace Hosting and NASA. OpenStack has grown to be a global software community of developers collaborating on a standard and massively scalable open source cloud operating system. This mission is to enable any organization to create and offer cloud computing services running on standard hardware.

4.2 Why IaaS

Infrastructure as a Service (IaaS) comprises computing, storage, desktop and networking resources delivered on demand. Infrastructure as a Service is that part of cloud computing that allows leasing and managing computing infrastructure for business, office on needs. Computing infrastructure includes virtual machines (VMs), operating systems, middleware, runtime components, network, storage, data and applications. Cloud computing vendors provide the necessary underlying physical hardware (servers, network, and storage) that they own and manage transparently in the background. In general, organizations are embracing IaaS for one or more of the following reasons.

4.2.1 Capital Cost Avoidance

The difference between the capital cost of purchasing servers that an organization must write down over a period of years and the operational cost of renting the same server capabilities from a cloud provider is a core argument for IaaS. But every organization’s needs are different, and IaaS doesn’t always make better financial sense over in-house infrastructure solutions.

4.2.2 Fresh Technology

With more governments holding on longer to their PCs, servers, storage and networking gear, the prospect of renting infrastructure in the cloud can be a welcome idea. IaaS providers are more like to employ the latest technology to attract customers. They can afford to do so because they lease IaaS resources to many different organizations at the same time.

4.2.3 Resources on Demand

In a traditional infrastructure model, if the IT team needs, for example, more storage to support a new initiative, it must procure the storage system, have it delivered, set it up, configure and test it, and then make it operational. In the cloud, such IaaS is provisioned online and available in a fraction of the time.

4.2.4 Rapid Response

On-demand IaaS resources also make it possible for IT teams to pull together entire computing platforms rapidly. For example, if an organization must add hundreds of employees to meet a specific seasonal demand, the IT department can procure desktop resources, storage and the server environments and networking to support them from a centralized console.

4.2.5 Low Maintenance

Best of all, with IaaS, IT staff don’t have to put their hands on as many systems in order to troubleshoot and maintain them all. Significant system upgrades can be time- and resource-consuming for the IT group. But in the cloud, an IaaS provider handles upgrades, support and maintenance.

4.3 OpenStack Architecture

The Cloud Computing Architecture of a cloud solution is the structure of the system, which comprise on-premise and cloud resources, services, middleware, and software components, geo-location, the externally visible properties of those, and the relationships between them. The term also refers to documentation of a system's cloud computing architecture. Documenting facilitates communication between stakeholders, documents early decisions about high-level design, and allows reuse of design components and patterns between projects.

4.4 Conceptual architecture

OpenStack conceptual architecture shows the relationships among the OpenStack services in Figure 4.1.

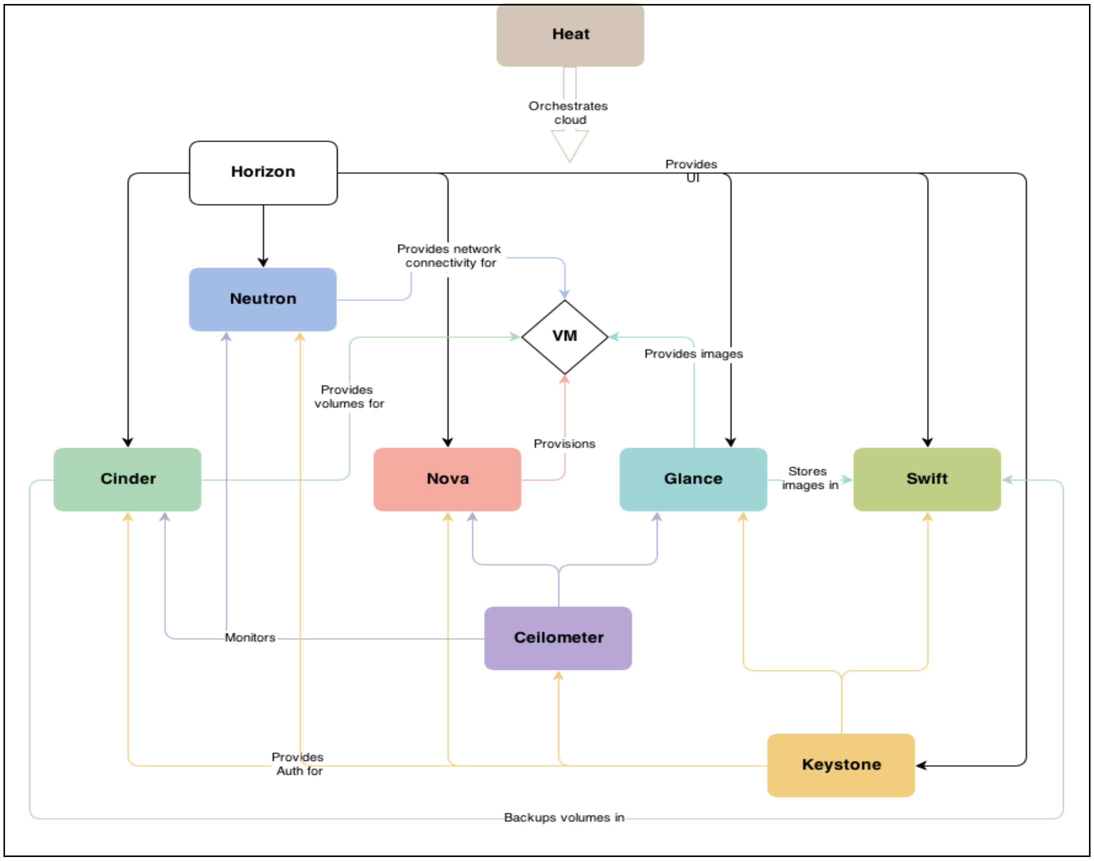


Figure 4. 1: Conceptual Architecture

4.5 Logical architecture

To design, install, and configure a cloud, cloud administrators must understand the logical Architecture.

OpenStack modules are one of the following types:

- **Daemon:** Runs as a daemon. On Linux platforms, a daemon is usually installed as a service.
- **Script:** Installs and tests of a virtual environment.
- **Command Line Interface (CLI):** Enables users to submit API calls to OpenStack services through easy-to-use commands.

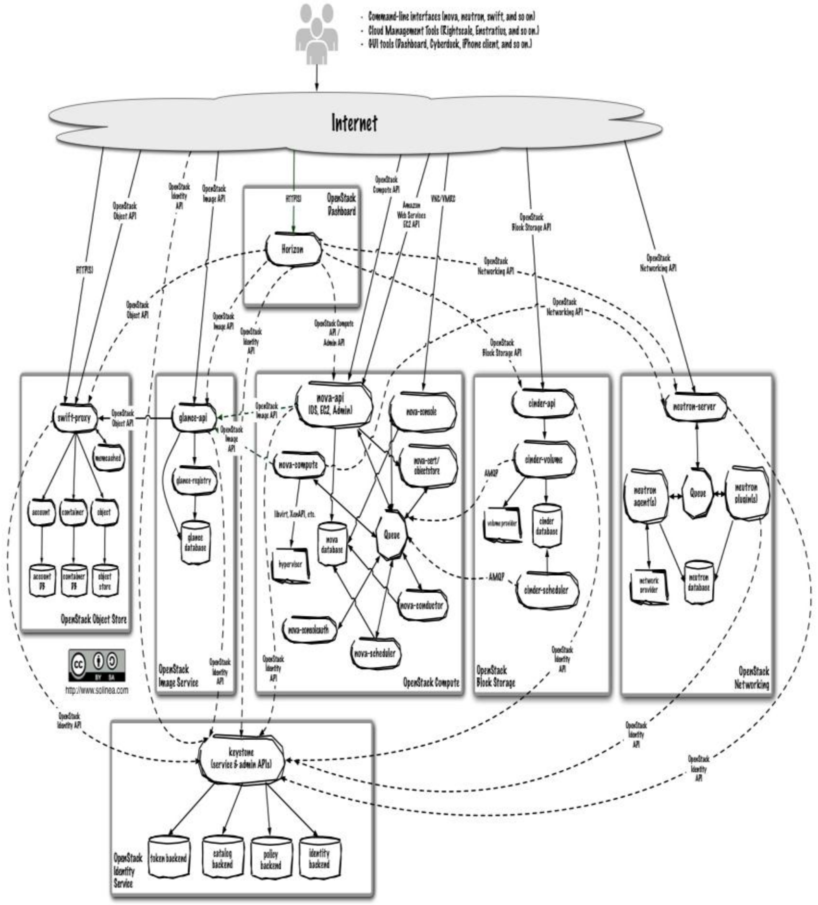


Figure 4. 2: Logical Architecture

4.6 OpenStack Components

Compute (Nova)

OpenStack Compute is a tool to orchestrate a cloud, including running instances, managing networks, and controlling access to the cloud through users and projects. The code name of OpenStack compute service is Nova. Nova provides the software that can control Infrastructure as a Service (IaaS) cloud computing platform . Nova does not include any virtualization software but it defines drivers to work with widely available virtualization technologies that run on host operating system[15][17].

Block Storage (Cinder)

The codename of the OpenStack Block Storage service is Cinder. It provides persistent block-level storage devices for use with OpenStack compute instances. The block storage system manages the creation, attaching and detaching of the block devices to servers. Block storage volumes are fully integrated into OpenStack Compute and the Dashboard allowing for cloud users to manage their own storage needs.[15][17]

Dashboard (Horizon)

The OpenStack dashboard, also known as Horizon, is a Web interface that provides administrators and users a graphical interface to manage various OpenStack resources, services and also automate cloud-bases resources. It uses database to store sessions. The dashboard is one of the several ways users can interact with OpenStack resources. The design of the dashboard can accommodates third party products and services such as monitoring and management tools. Developers can automate access or build tools to manage resources using the native OpenStack API or the EC2 compatibility API. [17]

Identity Service (Keystone)

OpenStack Identity (Keystone) provides a central directory of users mapped to the OpenStack services they can access. It is a single point of integration for Openstack policy catalog, token and authentication. It acts as a common authentication system across the cloud operating system and can integrate with existing backend directory services like LDAP. It supports multiple forms of authentication including standard username and password credentials, token-based systems and AWS-style (i.e. Amazon Web Services) logins. Additionally, the catalog provides a query able list of all of the services deployed in an OpenStack cloud in a single registry. Keystone handles API requests as well as providing configurable catalog, policy, token and identity services. It communicates via OpenStack Identity API version 2. Each Keystone function has a pluggable backend which allows different ways to use the particular service. Standard backend includes LDAP or SQL, as well as Key Value Stores (KVS). Most people will use this as a point of customization for their current authentication services.[15][17]

Image Service (Glance)

The image service module of Openstack is also known as glance which enables users to discover, register, and retrieve virtual machine images. The image of Openstack is an operating system that is installed on a virtual machine(VM). The glance module does not does not store images, variations, or instances—but rather catalogs them and holds their metadata from Swift or a storage backend data store. Glance has a RESTful API that allows querying of VM image metadata as well as retrieval of the actual image.[15] Glance can provide addition, deletion, sharing or duplication of images.[17]

CHAPTER 5
IMPLEMENTATION DETAILS

The aim of this chapter is to provide step by step processes which are required performing a complete installation of OpenStack. We build a lab using VMware Workstation version 10. We chose Ubuntu LTS ISO to create virtual machines and install OpenStack on top of it and finally we used manual and scripting process to complete the installation.

5.1 Implementation of the OpenStack

The implementation of OpenStack requires a lot of steps. Before installation of OpenStack we need to install a lot of pre-requisite services such as MySQL, RabbitMQ, NTP, OpenSSH server for remote access into VM etc. The following are the OpenStack components which we are going to install.

- OpenStack Identity (Keystone)
- OpenStack Image Service (Glance)
- OpenStack Compute (Nova)
- OpenStack Dashboard (Horizon)
- OpenStack Storage (Cinder)

5.2 Creating Virtual Networks

The Virtual Machine (VM) which will be installed in VMware Workstation requires two networks: one public and one private network. Public network will have the internet access to download all the necessary installation components. On the other hand private network will be used for nested virtual guests. In VMware Workstation, after clicking Virtual Network Editor from the edit menu of the VMware workstation then new Virtual Network Editor window will appear. The Virtual Network Editor is shown in Figure 4.1.

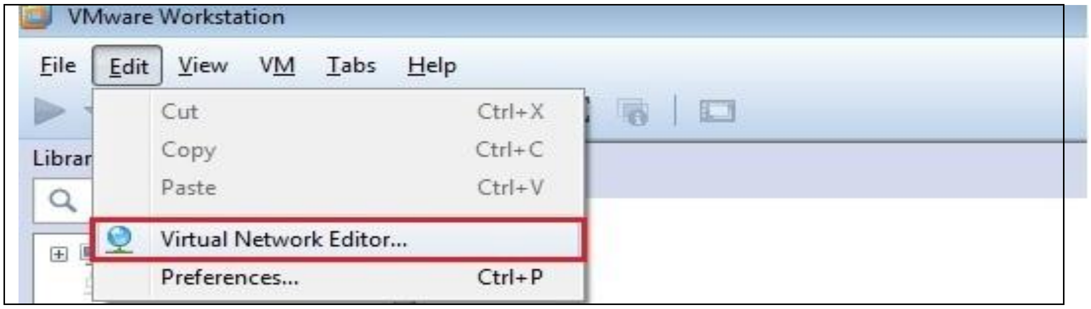
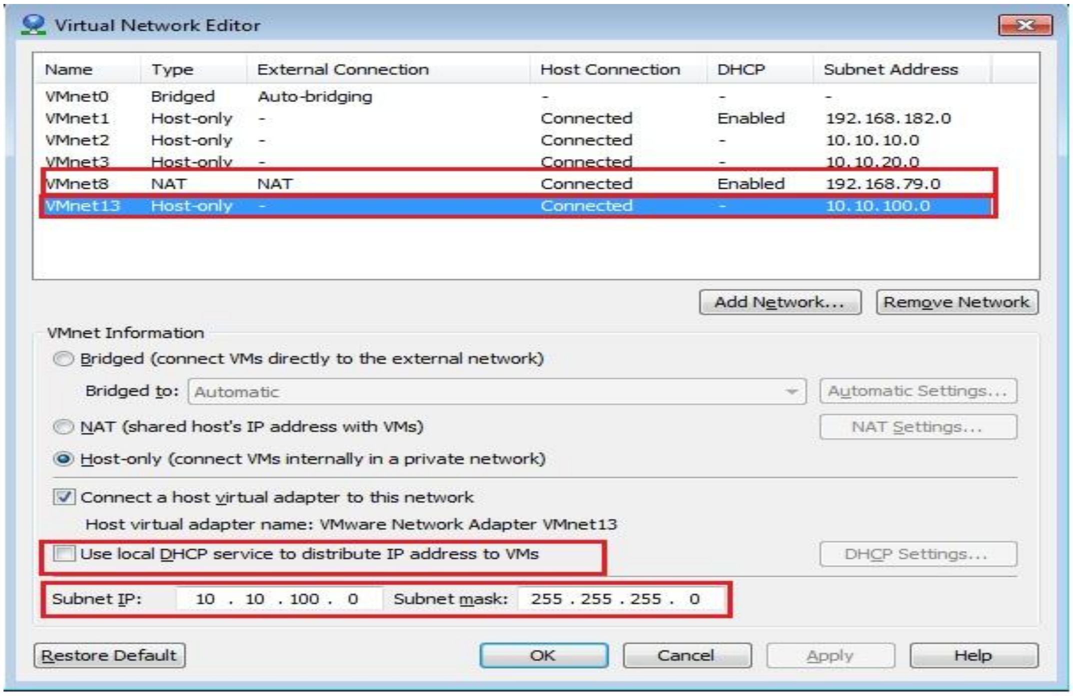


Figure 5. 1: Virtual Network Editor

Once the editor is launched, we can see different types of VM networks in the Virtual Network Editor. In the Virtual Network Editor, we create a public network which name is VMnet8 and type is Network Address Translation (NAT). The internet protocol (IP) address block for this public network is 192.168.79.0/24. Then we will create a private network which name is VMnet13 and type is Host-only. The internet protocol (IP) address block is 10.10.100.0/24. We need to unchecked the local DHCP service to distribute IP address to VMs and checked Subnet IP matches that is shown in Figure 4.2.



Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Auto-bridging	-	-	-
VMnet1	Host-only	-	Connected	Enabled	192.168.182.0
VMnet2	Host-only	-	Connected	-	10.10.10.0
VMnet3	Host-only	-	Connected	-	10.10.20.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.79.0
VMnet13	Host-only	-	Connected	-	10.10.100.0

VMnet Information

○ Bridged (connect VMs directly to the external network)

Bridged to: Automatic

○ NAT (shared host's IP address with VMs)

● Host-only (connect VMs internally in a private network)

☑ Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet13

☐ Use local DHCP service to distribute IP address to VMs

Subnet IP: 10 . 10 . 100 . 0 Subnet mask: 255 . 255 . 255 . 0

Figure 5. 2: IP Address Allocation in VMnet

5.3 Creating New Virtual Machine in VMware Workstation

This is the step-by-step of the process of deploying a new Ubuntu Server Virtual Machine (VM) guest on VMware Workstation 10 using the installation wizard. First, we have to click on File tab and it will open a drop down menu. From drop down menu we need to select New Virtual Machine. The snapshot of the process is shown in Figure 5.3.

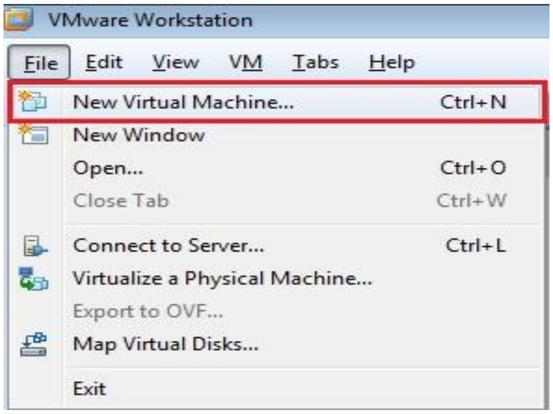
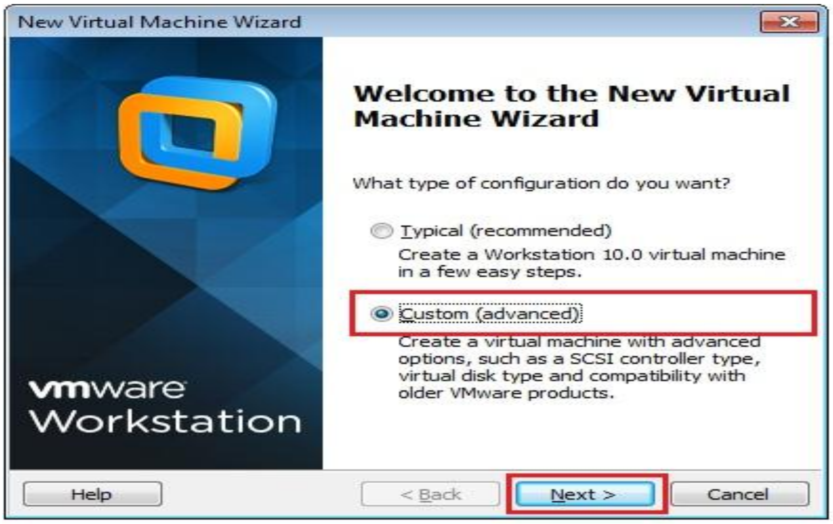


Figure 5. 3: New Virtual Machine

After clicking New Virtual Machine, New Virtual Machine Wizard will appear. Then we need to select the Custom (advanced) radio button and click next. The Figure 5.4 shows a custom Virtual Machine.



Welcome to the New Virtual Machine Wizard

What type of configuration do you want?

○ Typical (recommended)

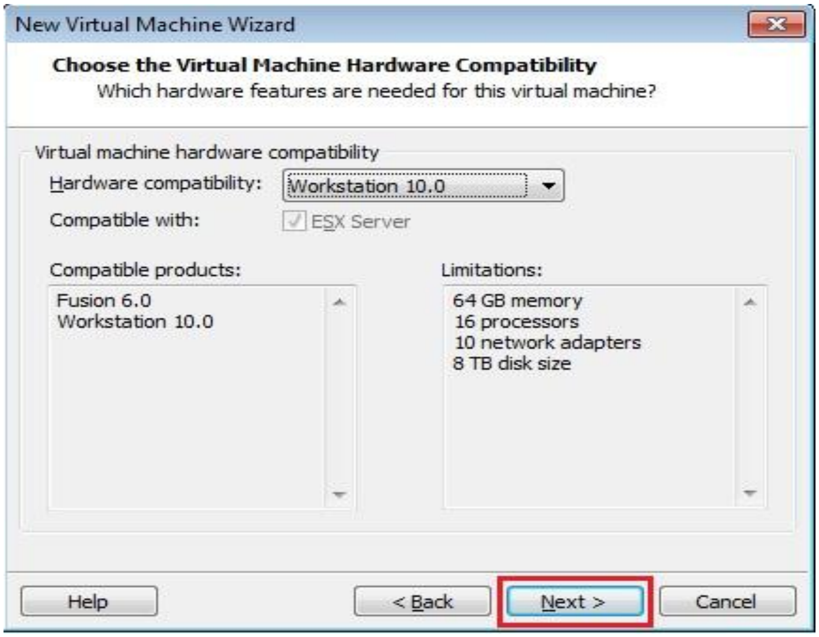
Create a Workstation 10.0 virtual machine in a few easy steps.

● Custom (advanced)

Create a virtual machine with advanced options, such as a SCSI controller type, virtual disk type and compatibility with older VMware products.

Figure 5. 4: New Virtual Machine Wizard

We will take the default settings of Workstation 10.0 for Hardware Compatibility and click next which are shown below in the Figure 5.5.



New Virtual Machine Wizard

Choose the Virtual Machine Hardware Compatibility

Which hardware features are needed for this virtual machine?

Virtual machine hardware compatibility

Hardware compatibility: Workstation 10.0

Compatible with: ☑ ESX Server

Compatible products:

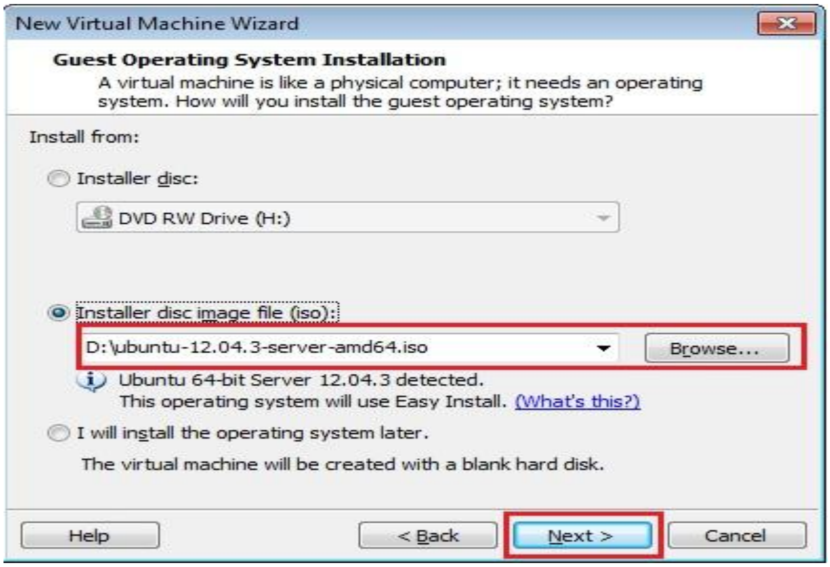
Fusion 6.0
Workstation 10.0

Limitations:

64 GB memory
16 processors
10 network adapters
8 TB disk size

Figure 5. 5: Hardware Compatibility

After Clicking the Browse button we will show the path of Ubuntu Server ISO to it in VMware workstation. Then we will click next. The process is shown in Figure 5.6.



New Virtual Machine Wizard

Guest Operating System Installation

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

○ Installer disc:

DVD RW Drive (H:)

● Installer disc image file (iso):

D:\ubuntu-12.04.3-server-amd64.iso

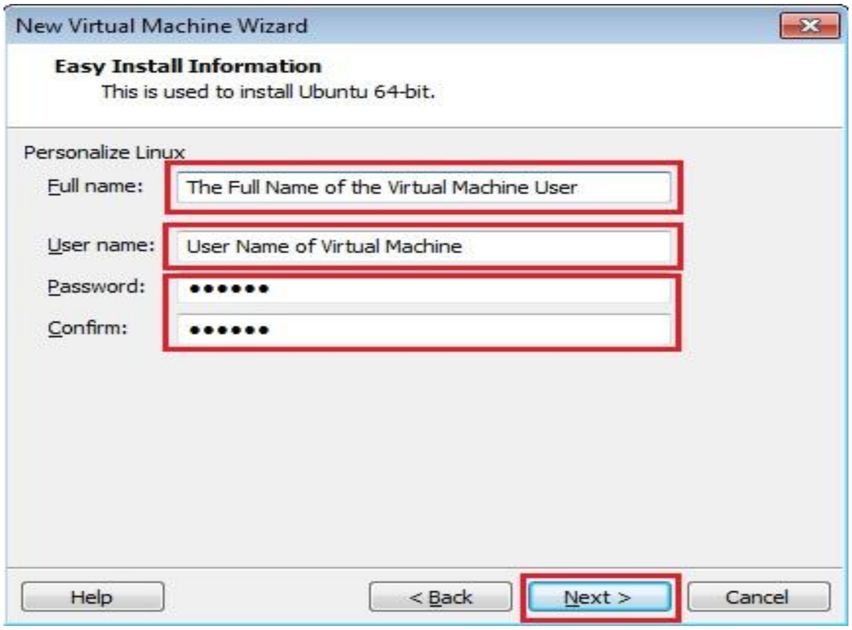
Ubuntu 64-bit Server 12.04.3 detected.
This operating system will use Easy Install. (What's this?)

○ I will install the operating system later.

The virtual machine will be created with a blank hard disk.

Figure 5. 6: Browse Disc Image

In the Easy Installation wizard we need to provide the full name of the virtual machine user and then user name and password which will be used to login to the virtual machine. After providing the required username and password we will click next which is show in the Figure 5.7.



New Virtual Machine Wizard

Easy Install Information

This is used to install Ubuntu 64-bit.

Personalize Linux

Full name: The Full Name of the Virtual Machine User

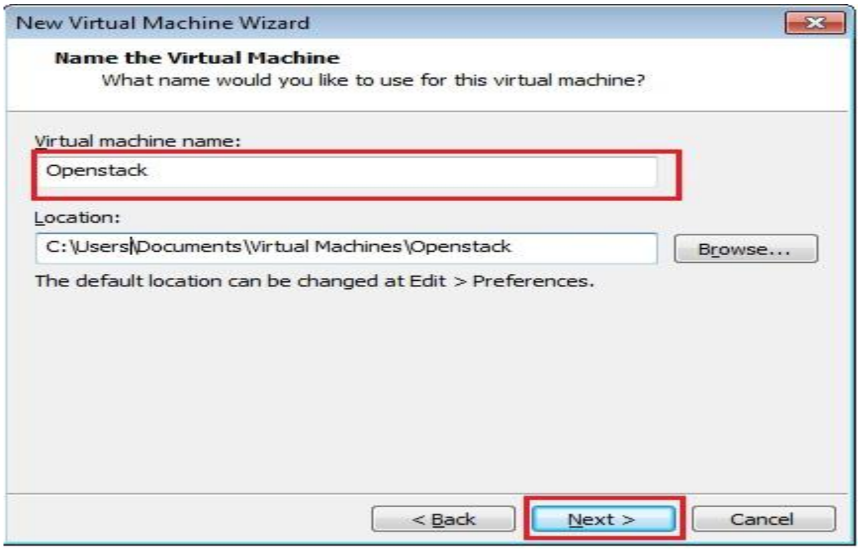
User name: User Name of Virtual Machine

Password: ●●●●●●

Confirm: ●●●●●●

Figure 5. 7:Define Virtual Machine User

After clicking the next button a new wizard will appear where we need to define the name of Virtual machine which is shown in figure 5.8. Then click next button.


New Virtual Machine Wizard

Name the Virtual Machine

What name would you like to use for this virtual machine?

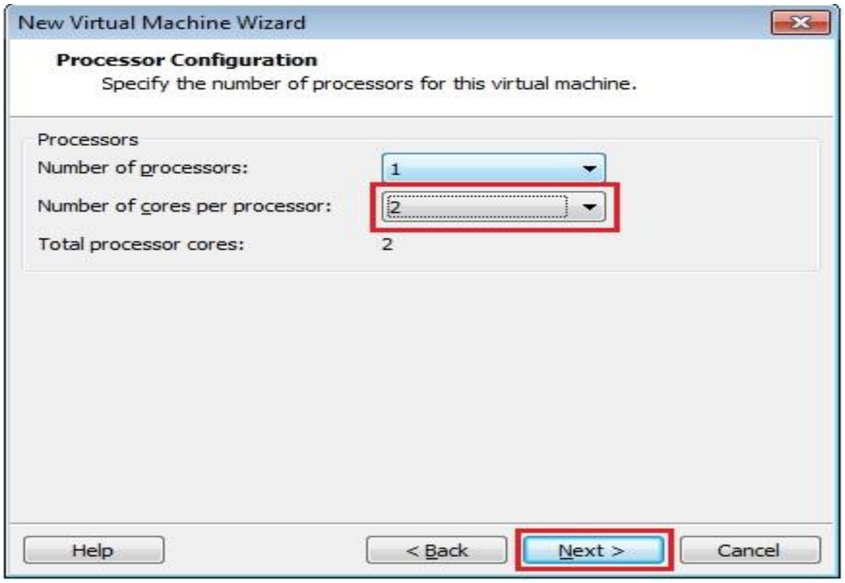
Virtual machine name: Openstack

Location: C:\Users\Documents\Virtual Machines\Openstack

The default location can be changed at Edit > Preferences.

Figure 5. 8: Virtual Machine Naming

Then we need to select the number of processor and cores. The smallest build that would recommend is a 1 CPU with 2 cores. Selecting Number of Processor shown in Figure 5.9.


New Virtual Machine Wizard

Processor Configuration

Specify the number of processors for this virtual machine.

Processors

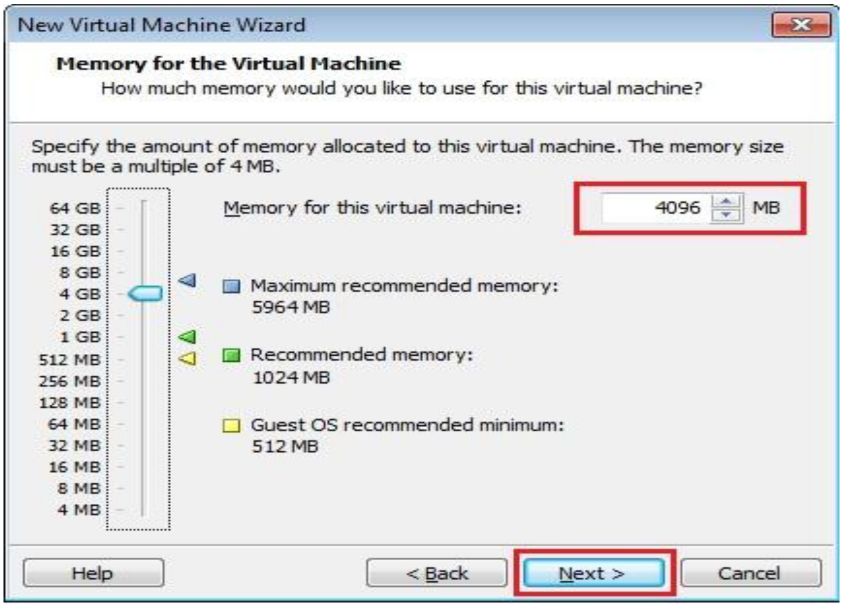
Number of processors: 1

Number of cores per processor: 2

Total processor cores: 2

Figure 5. 9: Selecting Process Number

In the memory for the virtual machine wizard we will select 4 GB or ram which is recommended for testing environment. Figure 5.10 shows the amount of memory which was allocated for lab environment.



New Virtual Machine Wizard

Memory for the Virtual Machine

How much memory would you like to use for this virtual machine?

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 4096 MB

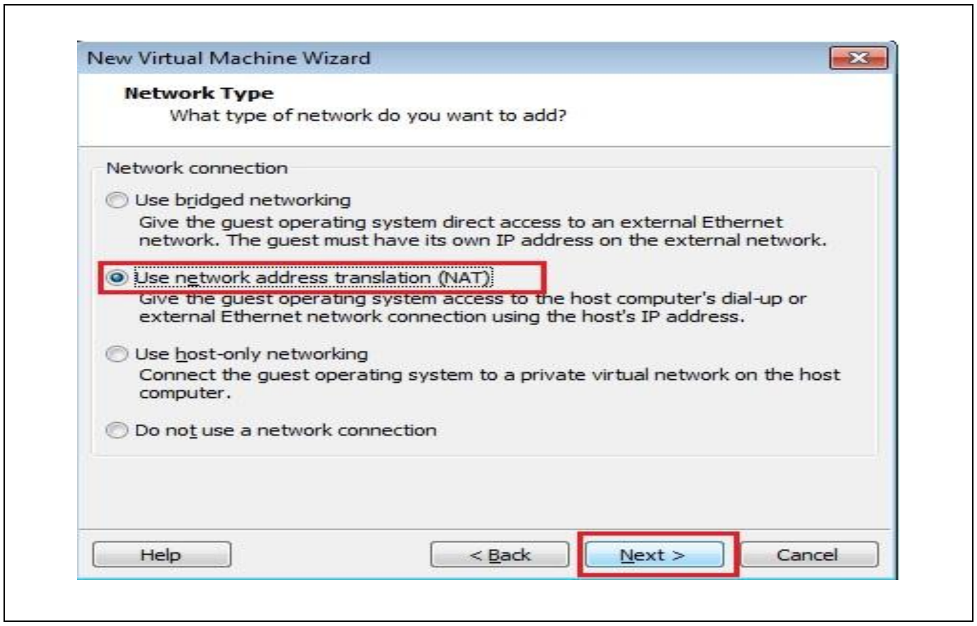
🟦 Maximum recommended memory:
5964 MB

🟩 Recommended memory:
1024 MB

🟨 Guest OS recommended minimum:
512 MB

Figure 5.10: Virtual Machine Memory Allocation

In the network type section we will select Network Address Translation (NAT) for network interface card for public facing interface which will be used for OpenStack packages. The network selection process is shown in Figure 5.11.



New Virtual Machine Wizard

Network Type

What type of network do you want to add?

Network connection

○ Use bridged networking
Give the guest operating system direct access to an external Ethernet network. The guest must have its own IP address on the external network.

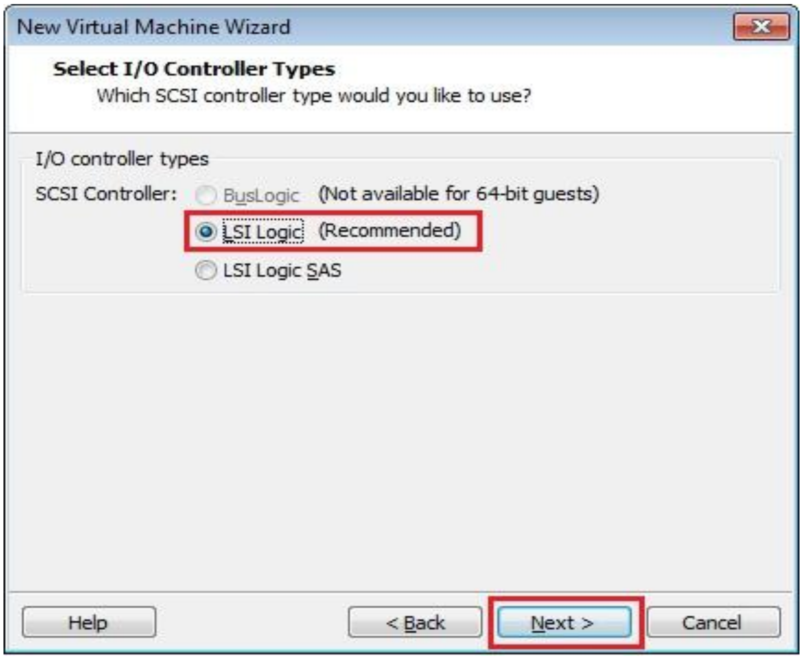
● Use network address translation (NAT)
Give the guest operating system access to the host computer's dial-up or external Ethernet network connection using the host's IP address.

○ Use host-only networking
Connect the guest operating system to a private virtual network on the host computer.

○ Do not use a network connection

Figure 5. 11: Network Connection Types

From the controllers type we will select LSI Logic and click next which is shown in Figure: 5.12.



New Virtual Machine Wizard

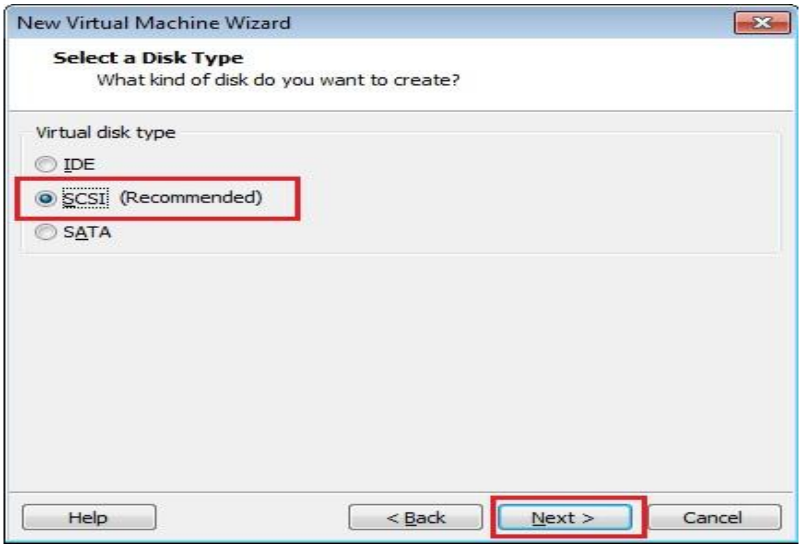
Select I/O Controller Types
Which SCSI controller type would you like to use?

I/O controller types

SCSI Controller: ○ BusLogic (Not available for 64-bit guests)
● LSI Logic (Recommended)
○ LSI Logic SAS

Figure 5. 12: Selecting I/O Controller

We will select SCSI for the virtual disk type which is recommended and then click next which is given in Figure 5.13.



New Virtual Machine Wizard

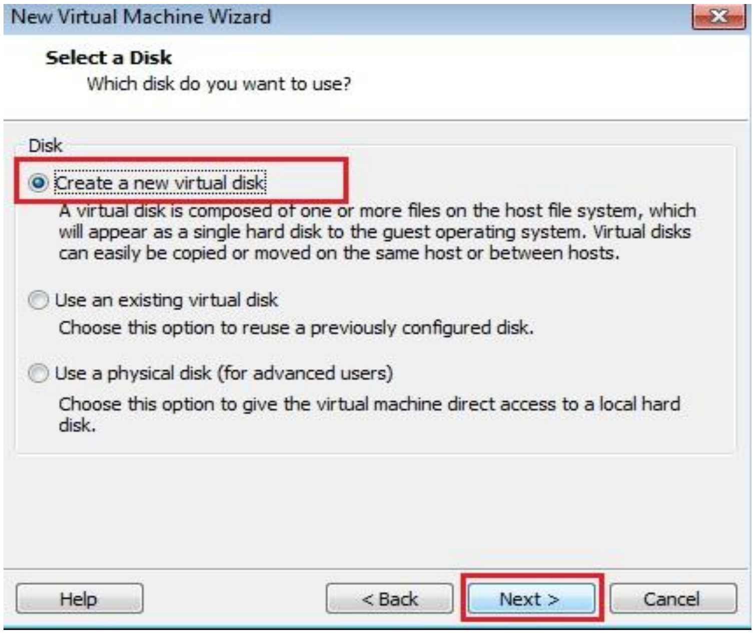
Select a Disk Type
What kind of disk do you want to create?

Virtual disk type

○ IDE
● SCSI (Recommended)
○ SATA

Figure 5.13:Selection of Disk Type

We will select “Create a new virtual disk” from Disk selection options and then click next which is shown in Figure 5.14.



New Virtual Machine Wizard

Select a Disk
Which disk do you want to use?

Disk

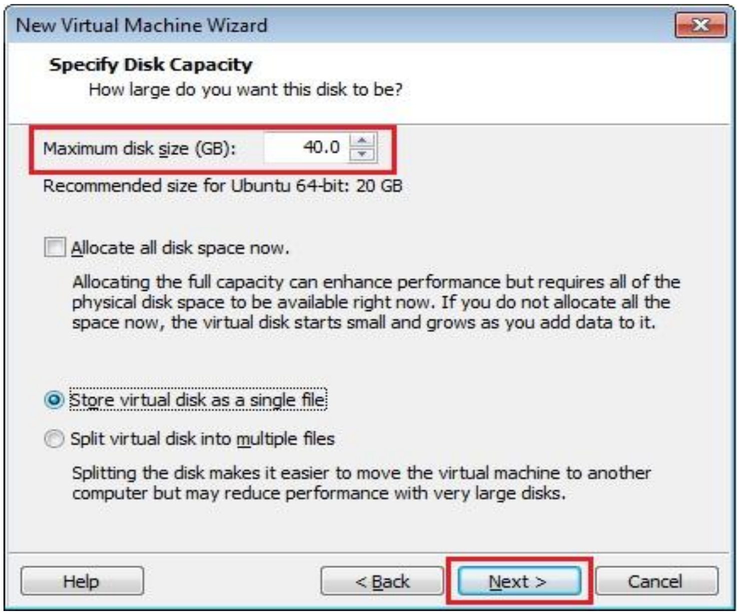
● Create a new virtual disk
A virtual disk is composed of one or more files on the host file system, which will appear as a single hard disk to the guest operating system. Virtual disks can easily be copied or moved on the same host or between hosts.

○ Use an existing virtual disk
Choose this option to reuse a previously configured disk.

○ Use a physical disk (for advanced users)
Choose this option to give the virtual machine direct access to a local hard disk.

Figure 5. 14: Creation of New Virtual Disk

From “Disk Capacity” we will select 40 GB which is recommended size for deployment of Ubuntu in Virtual Machine. For ease of movement and management, we need to select “Store virtual disk as a single file”. After completion of the selection process click next.



New Virtual Machine Wizard

Specify Disk Capacity
How large do you want this disk to be?

Maximum disk size (GB): 40.0

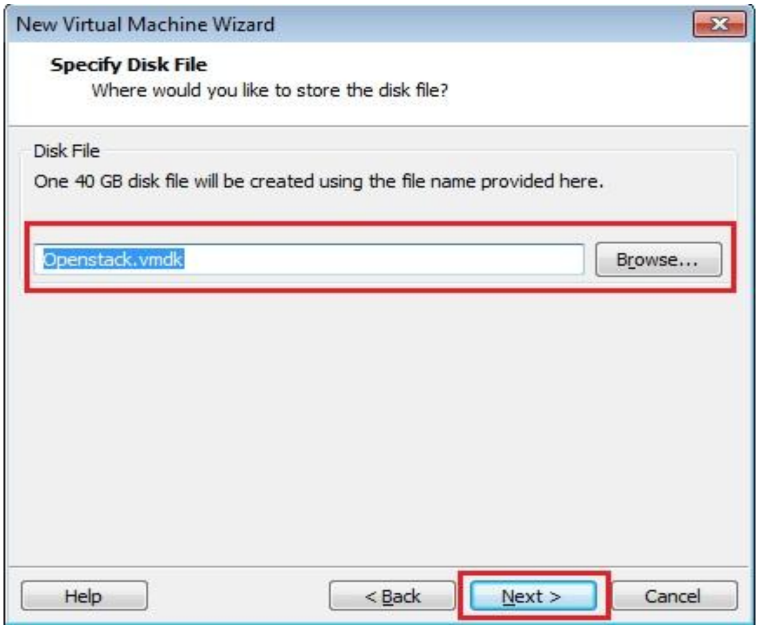
Recommended size for Ubuntu 64-bit: 20 GB

□ Allocate all disk space now.
Allocating the full capacity can enhance performance but requires all of the physical disk space to be available right now. If you do not allocate all the space now, the virtual disk starts small and grows as you add data to it.

● Store virtual disk as a single file
○ Split virtual disk into multiple files
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

Figure 5. 15: Disk Capacity Specification

In the “Disk File” any name can be provided as the file name which is shown in Figure 5.16.



Specify Disk File

Where would you like to store the disk file?

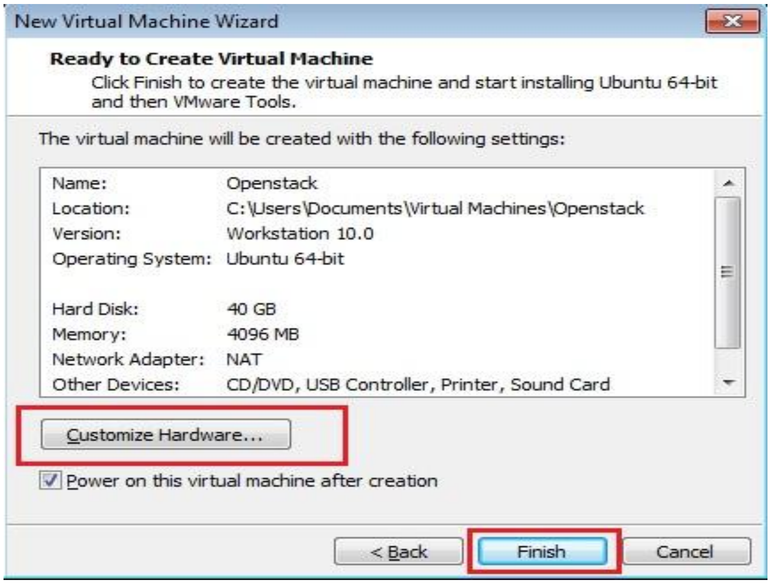
Disk File

One 40 GB disk file will be created using the file name provided here.

Openstack.vmdk

Figure 5. 16: Specify Disk File

We need to click on Customize Hardware option to customize the settings of virtual machine.



Ready to Create Virtual Machine

Click Finish to create the virtual machine and start installing Ubuntu 64-bit and then VMware Tools.

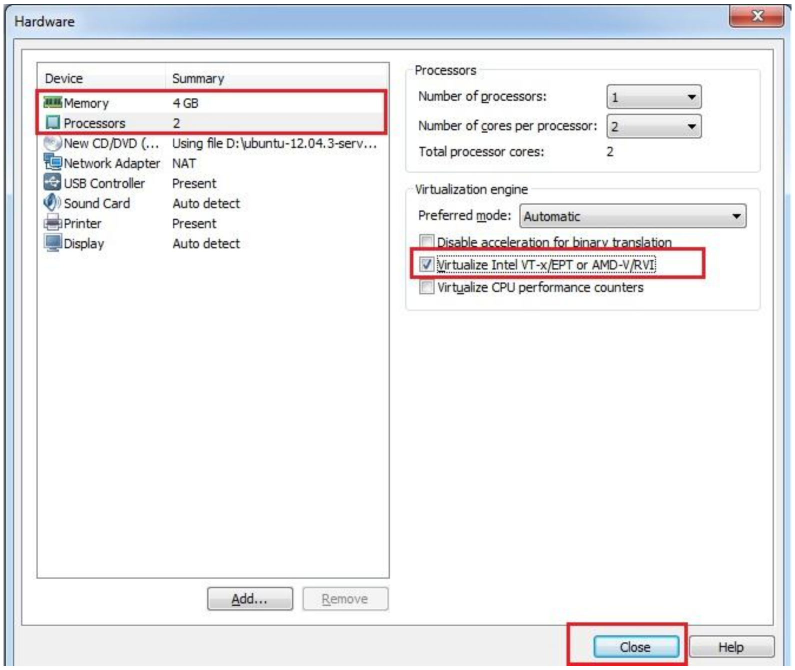
The virtual machine will be created with the following settings:

Name:	Openstack
Location:	C:\Users\Documents\Virtual Machines\Openstack
Version:	Workstation 10.0
Operating System:	Ubuntu 64-bit
Hard Disk:	40 GB
Memory:	4096 MB
Network Adapter:	NAT
Other Devices:	CD/DVD, USB Controller, Printer, Sound Card

☑ Power on this virtual machine after creation

Figure 5. 17: Customize Hardware

From the hardware customization pane we will click processor on the left hand pane and on the right hand pane we will put check mark on “Virtualize Intel VT-x/EPT or AMD-V/RVI” and then click close button.


Hardware

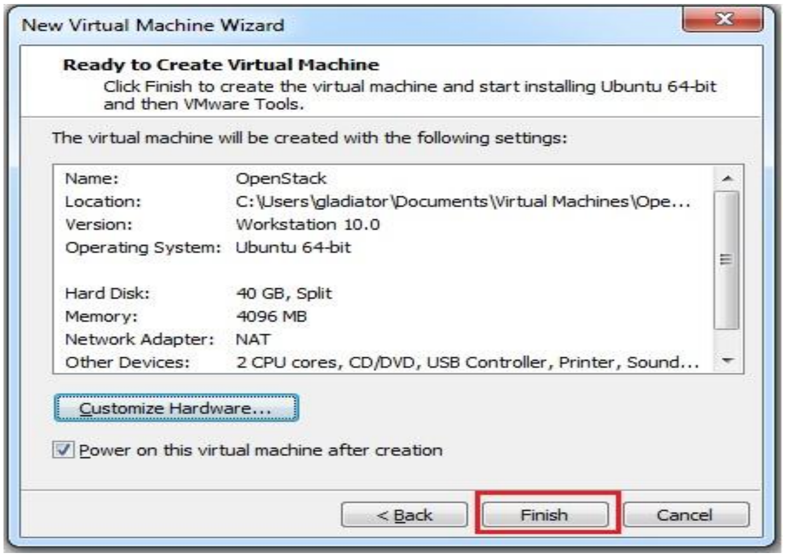
Device	Summary
Memory	4 GB
Processors	2
New CD/DVD (...	Using file D:\ubuntu-12.04.3-serv...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Processors
Number of processors: 1
Number of cores per processor: 2
Total processor cores: 2

Virtualization engine
Preferred mode: Automatic
□ Disable acceleration for binary translation
☑ Virtualize Intel VT-x/EPT or AMD-V/RVI
□ Virtualize CPU performance counters

Figure 5. 18: Enable Virtualize Intel VT

After customization of the hardware settings we will click finis button.


New Virtual Machine Wizard

Ready to Create Virtual Machine
Click Finish to create the virtual machine and start installing Ubuntu 64-bit and then VMware Tools.

The virtual machine will be created with the following settings:

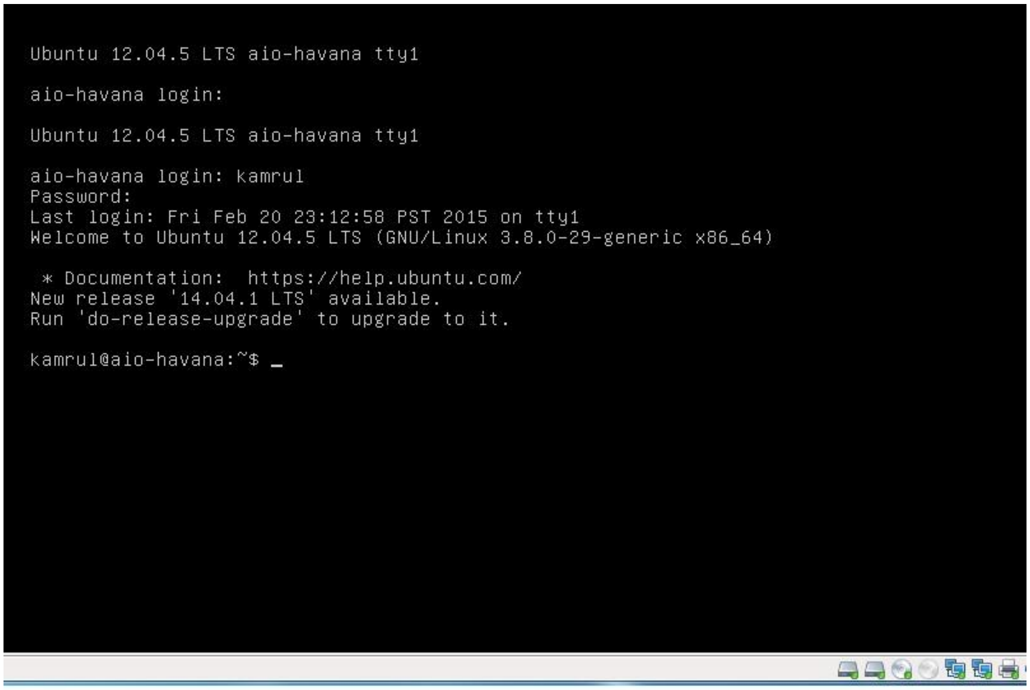
Name:	OpenStack
Location:	C:\Users\gladiator\Documents\Virtual Machines\Ope...
Version:	Workstation 10.0
Operating System:	Ubuntu 64-bit
Hard Disk:	40 GB, Split
Memory:	4096 MB
Network Adapter:	NAT
Other Devices:	2 CPU cores, CD/DVD, USB Controller, Printer, Sound...

☑ Power on this virtual machine after creation

Figure 5. 19: Finish Virtual Machine Wizard

5.4 Login to Ubuntu Operating System

We will login to our brand new virtual machine of Ubuntu operating system.

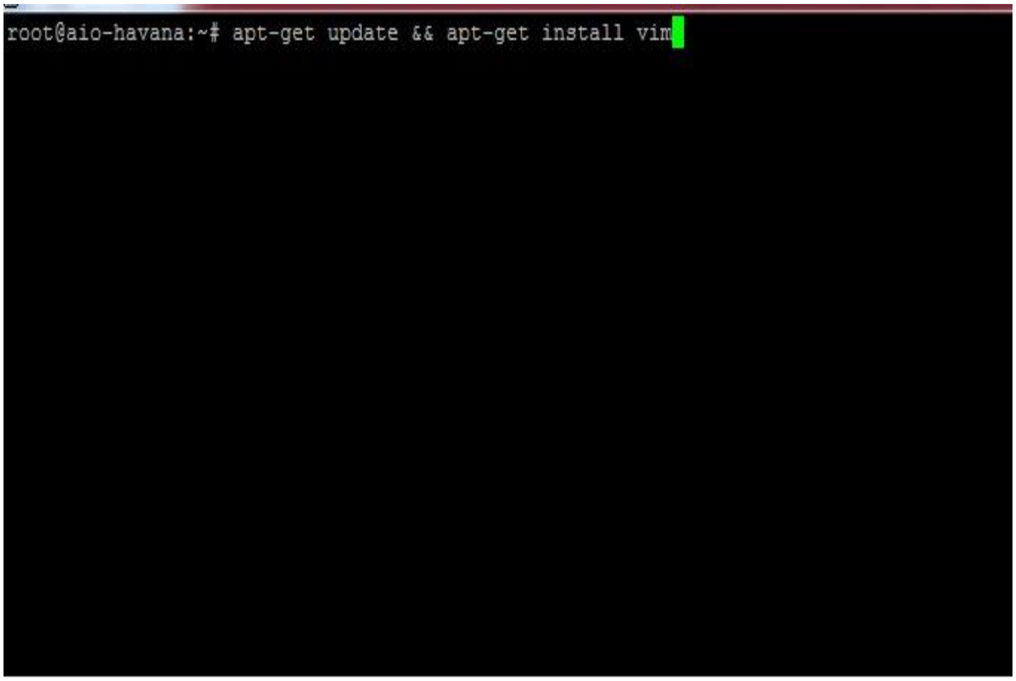


```
Ubuntu 12.04.5 LTS aio-havana tty1

aio-havana login:

Ubuntu 12.04.5 LTS aio-havana tty1

aio-havana login: kamrul
Password:
Last login: Fri Feb 20 23:12:58 PST 2015 on tty1
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.8.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

kamrul@aio-havana:~$ _
```

Figure 5. 20: Log in prompt of Ubuntu in Virtual Machine

The first thing is we need to update our Ubuntu operating system. Then we install vim packages using the following commands.

1. apt-get update
2. apt-get install vim

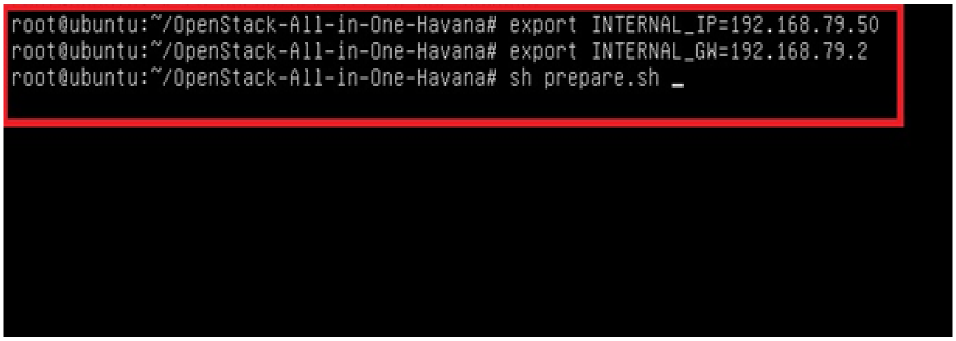


```
root@aio-havana:~# apt-get update && apt-get install vim
```

Figure 5. 21: Installation of Update and Vim Packages

Once the mentioned packages are downloaded and updated from the internet then we will enter in Ubuntu operating system as a root. Then we write a shell script which contains installation procedures. The written shell script is the installation procedure which can be modified according to the user need.

Then the system needs to set some system environment variables which can be installed using the scripts. The system also needs to set manually the IP address of machine and the gateway. Once the setup of the environment variables then we need to run the scripts to setup the machine and reboot the machine according to the need of the OpenStack user.

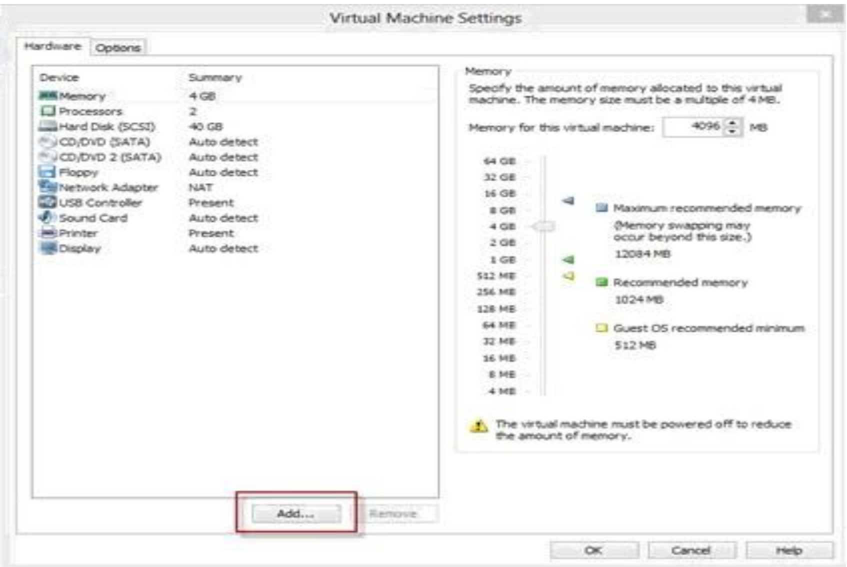


```
root@ubuntu:~/OpenStack-All-in-One-Havana# export INTERNAL_IP=192.168.79.50
root@ubuntu:~/OpenStack-All-in-One-Havana# export INTERNAL_GW=192.168.79.2
root@ubuntu:~/OpenStack-All-in-One-Havana# sh prepare.sh _
```

Figure 5. 22: IP address Export

5.5 Adding Network Adapters

After rebooting the virtual machine then we need to edit the virtual machine settings and click the add button to add network adapter.



Virtual Machine Settings

Device	Summary
Memory	4 GB
Processors	2
Hard Disk (SCSI)	40 GB
CD/DVD (SATA)	Auto detect
CD/DVD 2 (SATA)	Auto detect
Floppy	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 4096 MB

Maximum recommended memory (Memory swapping may occur beyond this size.) 12084 MB

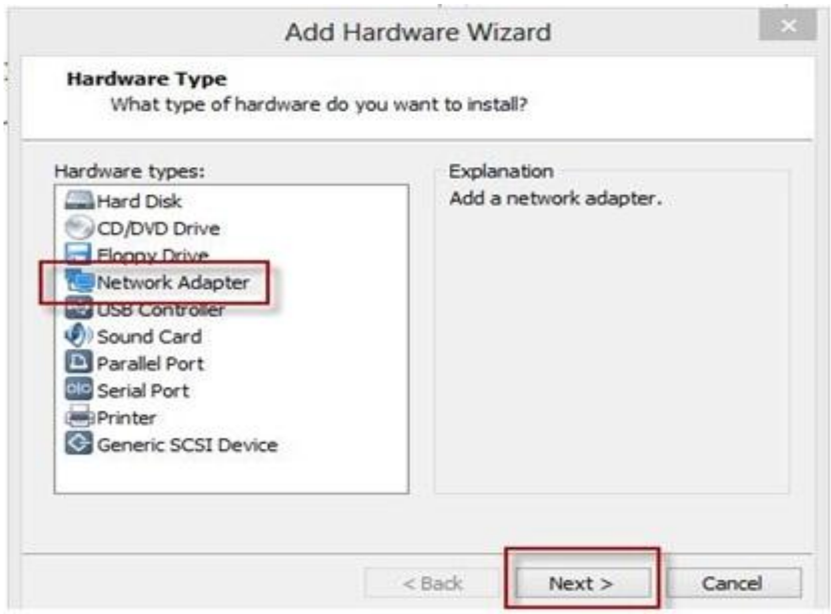
Recommended memory 1024 MB

Guest OS recommended minimum 512 MB

The virtual machine must be powered off to reduce the amount of memory.

Figure 5. 23: Virtual Machine Settings

After clicking on the add button we will get “Add Hardware Wizard” pane. From the pane we will select “Network Adapter” and click next.



Add Hardware Wizard

Hardware Type

What type of hardware do you want to install?

Hardware types:

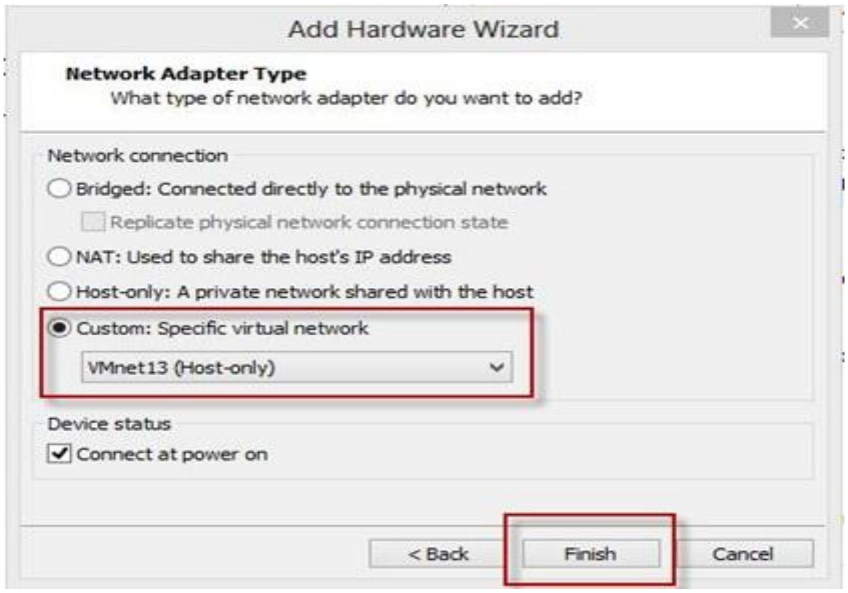
- Hard Disk
- CD/DVD Drive
- Floppy Drive
- Network Adapter
- USB Controller
- Sound Card
- Parallel Port
- Serial Port
- Printer
- Generic SCSI Device

Explanation

Add a network adapter.

Figure 5. 24: Add Network Adapter

For the “Network Adapter Type”, we need to choose “Custom: Specific virtual network” and pick the network “VMnet13 (Host-only)” from the drop list which is our internal network and the address is 10.10.100.0/24 segment. After doing all the actions click finish.



Add Hardware Wizard

Network Adapter Type

What type of network adapter do you want to add?

Network connection

○ Bridged: Connected directly to the physical network

□ Replicate physical network connection state

○ NAT: Used to share the host's IP address

○ Host-only: A private network shared with the host

● Custom: Specific virtual network

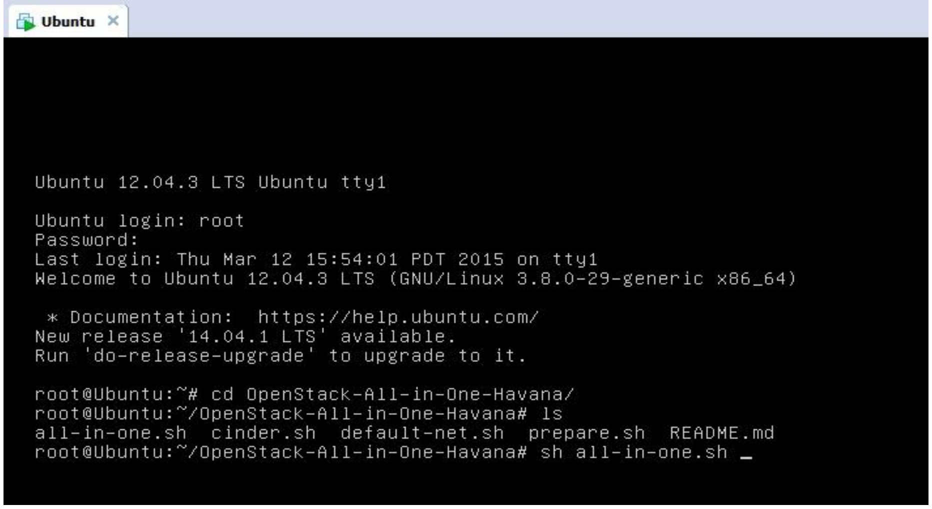
VMnet13 (Host-only)

Device status

☑ Connect at power on

Figure 5. 25: Selection Network Adapter Type

After rebooting the virtual machine the login prompt will be provided. We need to provide username and password in the machine to login. After successful login we need to run the shell script which we wrote to install OpenStack.



```
Ubuntu 12.04.3 LTS Ubuntu tty1

Ubuntu login: root
Password:
Last login: Thu Mar 12 15:54:01 PDT 2015 on tty1
Welcome to Ubuntu 12.04.3 LTS (GNU/Linux 3.8.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

root@Ubuntu:~# cd OpenStack-All-in-One-Havana/
root@Ubuntu:~/OpenStack-All-in-One-Havana# ls
all-in-one.sh  cinder.sh  default-net.sh  prepare.sh  README.md
root@Ubuntu:~/OpenStack-All-in-One-Havana# sh all-in-one.sh _
```

Figure 5. 26: Installation of OpenStack

5.6 Configuring Network in OpenStack

In OpenStack installation, we need to configure the network and add firewall rules to reach our instances over ICMP and SSH. To do this we need to create a script file called default.sh where we will write down our rules. After writing down all the rules in that script files we will run the script. The rules of file are given below.

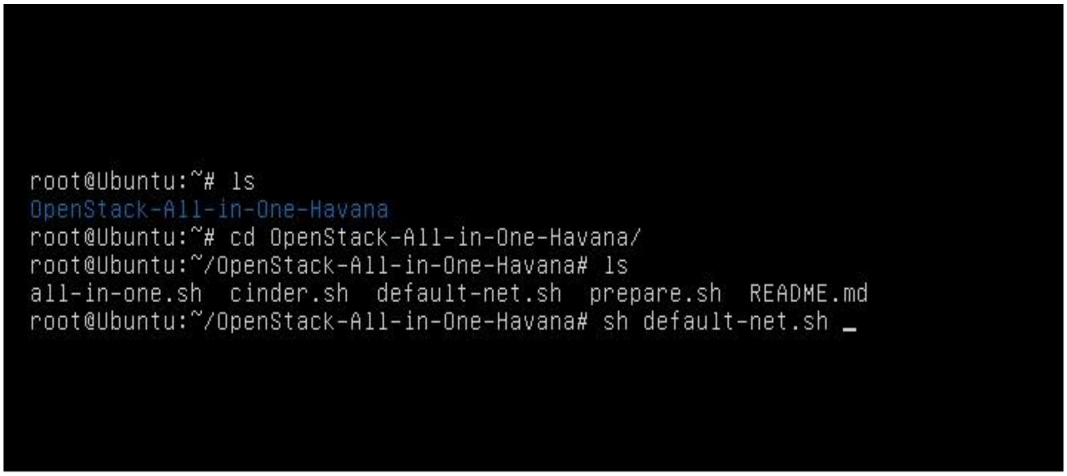
Rule to add our network:

Nova network-create vmnet –fixed-range-v4=10.10.100.0/24 –bridge-interface=br100 –multi-host=T

Firewall rules to reach our instances over ICMP and SSH:

nova secgroup-add-rule default tcp 22 22 0.0.0.0/0

nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0



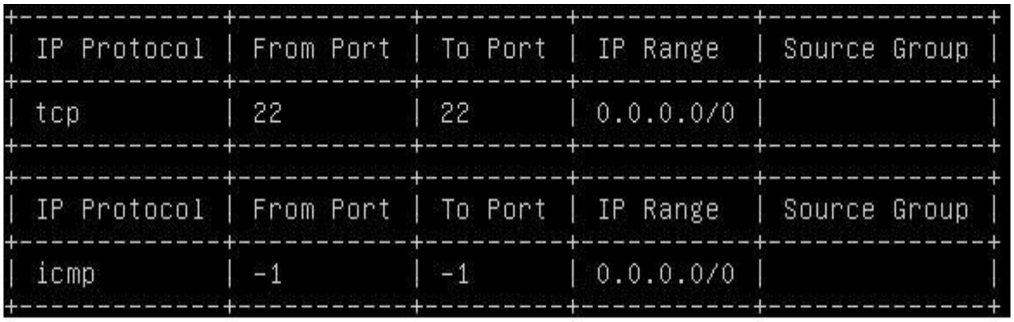
```
root@Ubuntu:~# ls
OpenStack-All-in-One-Havana
root@Ubuntu:~# cd OpenStack-All-in-One-Havana/
root@Ubuntu:~/OpenStack-All-in-One-Havana# ls
all-in-one.sh  cinder.sh  default-net.sh  prepare.sh  README.md
root@Ubuntu:~/OpenStack-All-in-One-Havana# sh default-net.sh _
```

Figure 5. 27: Installation of default script

That script does the following:

- Create a flat DHCP network for guest VMs (10.10.100.0/24)
- Add a firewall rule to allow telnet (TCP 22) from all source IP addresses
- Add a firewall rule to allow ping (ICMP) from all source IP addresses

After running the script it will give the following output which is shown below.



```
+-------------+-----------+---------+-----------+--------------+
| IP Protocol | From Port | To Port | IP Range  | Source Group |
+-------------+-----------+---------+-----------+--------------+
| tcp         | 22        | 22      | 0.0.0.0/0 |              |
+-------------+-----------+---------+-----------+--------------+
+-------------+-----------+---------+-----------+--------------+
| IP Protocol | From Port | To Port | IP Range  | Source Group |
+-------------+-----------+---------+-----------+--------------+
| icmp        | -1        | -1      | 0.0.0.0/0 |              |
+-------------+-----------+---------+-----------+--------------+
```

Figure 5. 28: Output of the default.sh file

5.7 Creating SSH Keypair

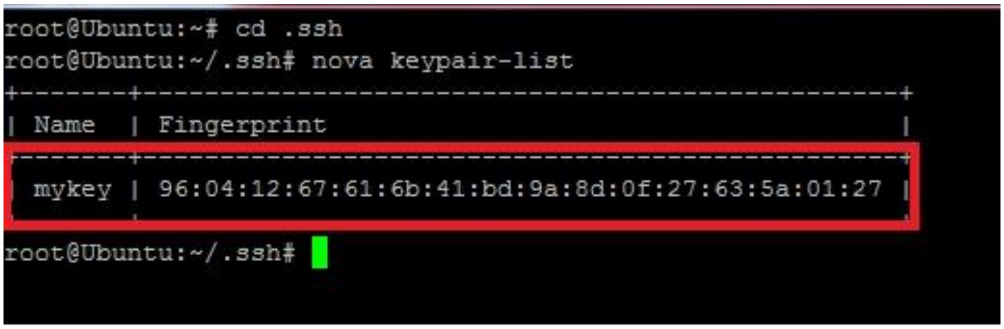
Generating an SSH key is quite simple. Entering the following commands will create SSH keypair. During the SSH keygen creation, it will prompt for different questions and we just need to press enter to take the defaults for file name and location. The file will be stored in a hidden directory (/root/.ssh) and the name will be id_rsa.pub. We need to change the name by any name and in our case we used “mykey”. The following command will be used to create keypair list.

1. nova keypair-add –pub_key id_rsa.pub mykey
2. nova keypair-list

Now we need to check the status of creation of the keypair list. To check that we will use the following in command prompt.

- nova keypair-list

The nova keypair-list command confirmed that key has been created and loaded into Nova. The output of the command is shown below.

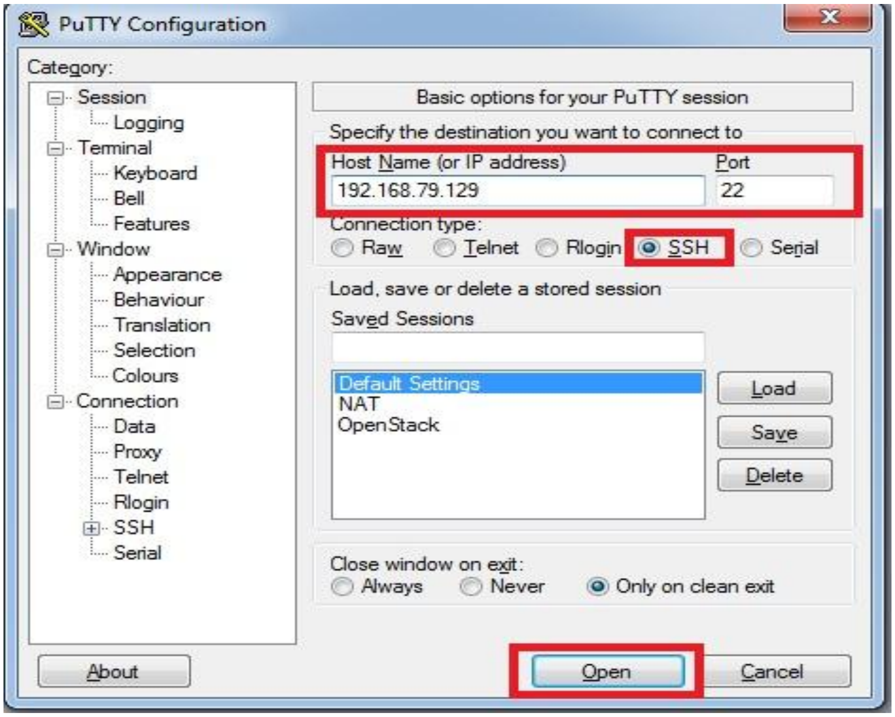


```
root@Ubuntu:~# cd .ssh
root@Ubuntu:~/.ssh# nova keypair-list
+-------+-------------------------------------------------+
| Name  | Fingerprint                                     |
+-------+-------------------------------------------------+
| mykey | 96:04:12:67:61:6b:41:bd:9a:8d:0f:27:63:5a:01:27 |
+-------+-------------------------------------------------+
root@Ubuntu:~/.ssh#
```

Figure 5. 29: Output of nova keypair-list

5.8 Remote Access Using Putty

We need to download putty for remote access to the virtual machines because it will make the interaction easier. It is not possible to copy and paste anything from outside of the virtual machines to inside of the virtual machine. In that case putty will help us to copy and paste any configuration to our Ubuntu virtual machine. After launching the putty we need to provide the internet protocol address of virtual machine to access the machine remotely using SSH protocol. The steps are shown in the figure below.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.79.129

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

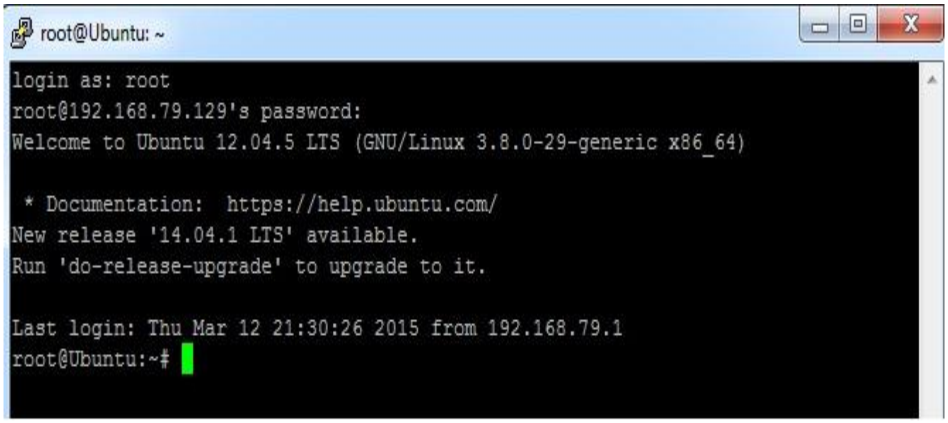
Saved Sessions:

Default Settings
NAT
OpenStack

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 5. 30: Login Ubuntu machine from remotely using Putty

After successful login with root using putty from remotely to our Ubuntu virtual machine we will see the command prompt like the following. Now we can insert any command to our Ubuntu virtual machine using putty. The output is shown below.



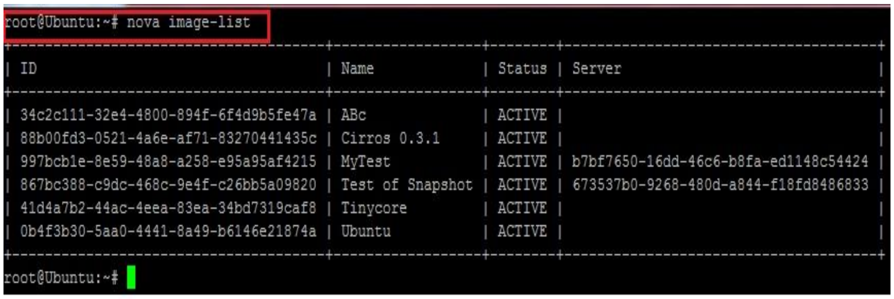
```
login as: root
root@192.168.79.129's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.8.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Mar 12 21:30:26 2015 from 192.168.79.1
root@Ubuntu:~#
```

Figure 5. 31: Shell prompt access using Putty

5.9 Important things to remember for booting VM instance

- Instance flavor : By default there are 5 flavors. We can create according to out need.
- SSH Key name – This keypair has already been created.
- Image name – This are the boot images which will be used for installation in virtual machine. A CirrOS Linux machine image has already installed during the OpenStack installation.
- Security Group – By default only one group has created which has access rules.
- Instance name – This is name of the virtual machine.

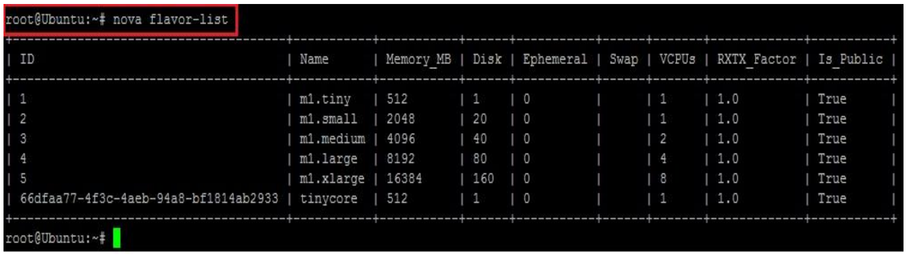
The several nova command and its outputs are show below. The output of the "nova image-list" command is shown below.



```
root@Ubuntu:~# nova image-list
+--------------------------------------+------------------+--------+--------------------------------------+
| ID                                   | Name             | Status | Server                               |
+--------------------------------------+------------------+--------+--------------------------------------+
| 34c2c111-32e4-4800-894f-6f4d9b5fe47a | ABc              | ACTIVE |                                      |
| 88b00fd3-0521-4a6e-af71-83270441435c | Cirros 0.3.1     | ACTIVE |                                      |
| 997bcb1e-8e59-48a8-a258-e95a95af4215 | MyTest           | ACTIVE | b7bf7650-16dd-46c6-b8fa-ed1148c54424 |
| 867bc388-c9dc-468c-9e4f-c26bb5a09820 | Test of Snapshot | ACTIVE | 673537b0-9268-480d-a844-f18fd8486833 |
| 41d4a7b2-44ac-4eea-83ea-34bd7319caf8 | Tinycore         | ACTIVE |                                      |
| 0b4f3b30-5aa0-4441-8a49-b6146e21874a | Ubuntu           | ACTIVE |                                      |
+--------------------------------------+------------------+--------+--------------------------------------+
root@Ubuntu:~#
```

Figure 5. 32: Nova image list

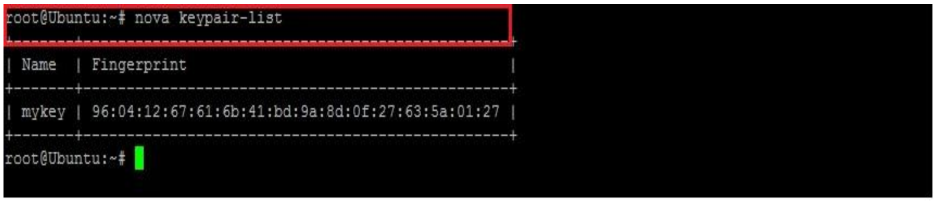
The output of the “nova flavor-list” command is shown below.



```
root@Ubuntu:~# nova flavor-list
+--------------------------------------+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| ID                                   | Name      | Memory_MB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor | Is_Public |
+--------------------------------------+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| 1                                    | m1.tiny   | 512       | 1    | 0         |      | 1     | 1.0         | True      |
| 2                                    | m1.small  | 2048      | 20   | 0         |      | 1     | 1.0         | True      |
| 3                                    | m1.medium | 4096      | 40   | 0         |      | 2     | 1.0         | True      |
| 4                                    | m1.large  | 8192      | 80   | 0         |      | 4     | 1.0         | True      |
| 5                                    | m1.xlarge | 16384     | 160  | 0         |      | 8     | 1.0         | True      |
| 66dfaa77-4f3c-4aeb-94a8-bf1814ab2933 | tinycore  | 512       | 1    | 0         |      | 1     | 1.0         | True      |
+--------------------------------------+-----------+-----------+------+-----------+------+-------+-------------+-----------+
root@Ubuntu:~#
```

Figure 5. 33: Nova flavor list

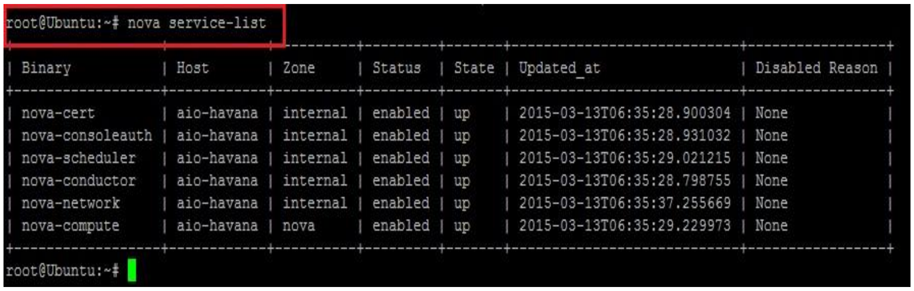
The output of the “nova keypair-list” command is shown below.



```
root@Ubuntu:~# nova keypair-list
+-------+-------------------------------------------------+
| Name  | Fingerprint                                     |
+-------+-------------------------------------------------+
| mykey | 96:04:12:67:61:6b:41:bd:9a:8d:0f:27:63:5a:01:27 |
+-------+-------------------------------------------------+
root@Ubuntu:~#
```

Figure 5. 34: Nova Keypair list

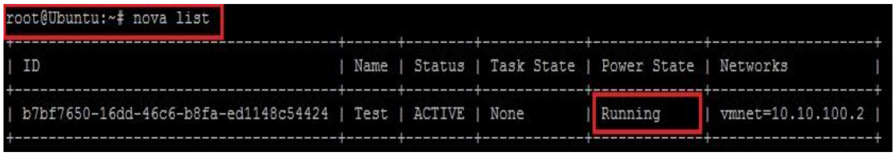
The output of the “nova service-list” command is shown below.



```
root@Ubuntu:~# nova service-list
+------------------+------------+----------+---------+-------+----------------------------+-----------------+
| Binary           | Host       | Zone     | Status  | State | Updated_at                 | Disabled Reason |
+------------------+------------+----------+---------+-------+----------------------------+-----------------+
| nova-cert        | aio-havana | internal | enabled | up    | 2015-03-13T06:35:28.900304 | None            |
| nova-consoleauth | aio-havana | internal | enabled | up    | 2015-03-13T06:35:28.931032 | None            |
| nova-scheduler   | aio-havana | internal | enabled | up    | 2015-03-13T06:35:29.021215 | None            |
| nova-conductor   | aio-havana | internal | enabled | up    | 2015-03-13T06:35:28.798755 | None            |
| nova-network     | aio-havana | internal | enabled | up    | 2015-03-13T06:35:37.255669 | None            |
| nova-compute     | aio-havana | nova     | enabled | up    | 2015-03-13T06:35:29.229973 | None            |
+------------------+------------+----------+---------+-------+----------------------------+-----------------+
root@Ubuntu:~#
```

Figure 5. 35: Nova service list

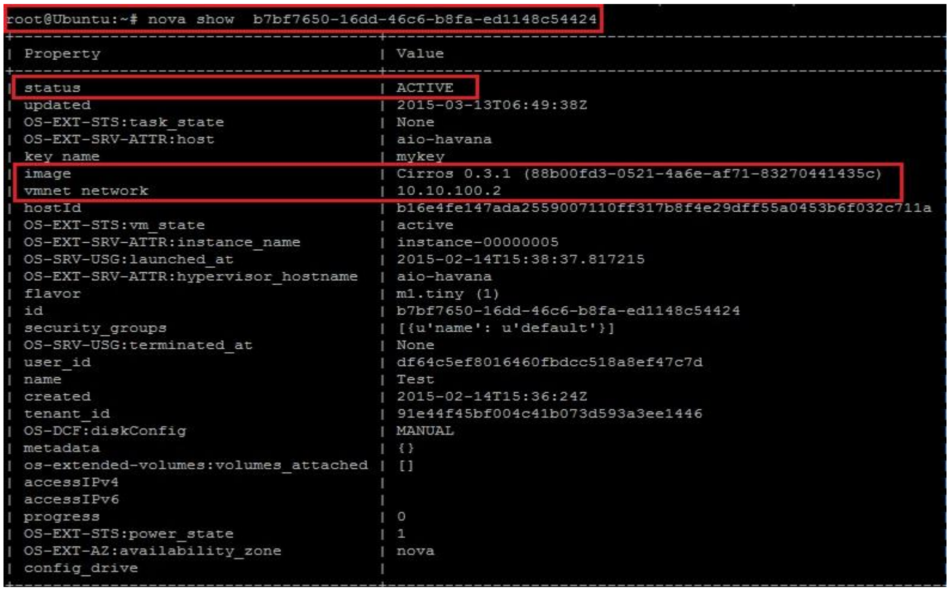
The command “nova list” will show the different properties of the virtual instance created inside the OpenStacck. The output of the “nova list” command also shows the identity of the instance, the name of the instance, the task of the instance, the state of the instance and finally the IP address of the instance.



```
root@Ubuntu:~# nova list
+--------------------------------------+------+--------+------------+-------------+-------------------+
| ID                                   | Name | Status | Task State | Power State | Networks          |
+--------------------------------------+------+--------+------------+-------------+-------------------+
| b7bf7650-16dd-46c6-b8fa-ed1148c54424 | Test | ACTIVE | None       | Running     | vmnet=10.10.100.2 |
+--------------------------------------+------+--------+------------+-------------+-------------------+
```

Figure 5. 36: Virtual Machine state

The “nova show” command shows all the properties of the created instance in details. The output of the command is shown below.

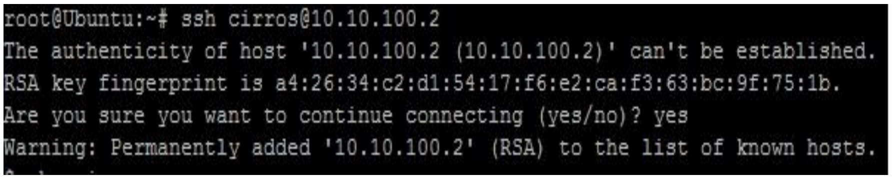


```
root@Ubuntu:~# nova show  b7bf7650-16dd-46c6-b8fa-ed1148c54424
+--------------------------------------+-----------------------------------------------------------------
| Property                             | Value
+--------------------------------------+-----------------------------------------------------------------
| status                               | ACTIVE
| updated                              | 2015-03-13T06:49:38Z
| OS-EXT-STS:task_state                | None
| OS-EXT-SRV-ATTR:host                 | aio-havana
| key name                             | mykey
| image                                | Cirros 0.3.1 (88b00fd3-0521-4a6e-af71-83270441435c)
| vmnet network                        | 10.10.100.2
| hostId                               | b16e4fe147ada2559007110ff317b8f4e29dff55a0453b6f032c711a
| OS-EXT-STS:vm_state                  | active
| OS-EXT-SRV-ATTR:instance_name        | instance-00000005
| OS-SRV-USG:launched_at               | 2015-02-14T15:38:37.817215
| OS-EXT-SRV-ATTR:hypervisor_hostname  | aio-havana
| flavor                               | m1.tiny (1)
| id                                   | b7bf7650-16dd-46c6-b8fa-ed1148c54424
| security_groups                      | [{u'name': u'default'}]
| OS-SRV-USG:terminated_at             | None
| user_id                              | df64c5ef8016460fbdcc518a8ef47c7d
| name                                 | Test
| created                              | 2015-02-14T15:36:24Z
| tenant_id                            | 91e44f45bf004c41b073d593a3ee1446
| OS-DCF:diskConfig                    | MANUAL
| metadata                             | {}
| os-extended-volumes:volumes_attached | []
| accessIPv4                           |
| accessIPv6                           |
| progress                             | 0
| OS-EXT-STS:power_state               | 1
| OS-EXT-AZ:availability_zone          | nova
| config_drive                         |
+--------------------------------------+-----------------------------------------------------------------
```

Figure 5. 37: The properties of the created instance

5.10 Connecting to Nova Instance

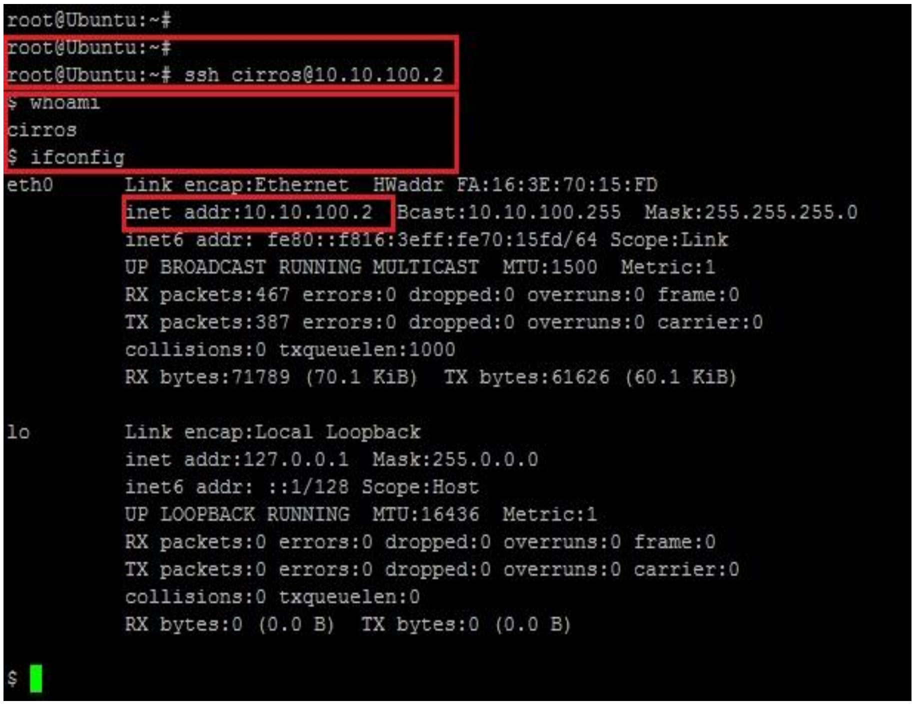
To confirm that the created instance is actually running we need to access using SSH in host virtual machine remotely. The boot image is CirrOS and the default username is cirros with no password.



```
root@Ubuntu:~# ssh cirros@10.10.100.2
The authenticity of host '10.10.100.2 (10.10.100.2)' can't be established.
RSA key fingerprint is a4:26:34:c2:d1:54:17:f6:e2:ca:f3:63:bc:9f:75:1b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.100.2' (RSA) to the list of known hosts.
```

Figure 5. 38: Cirros instance login

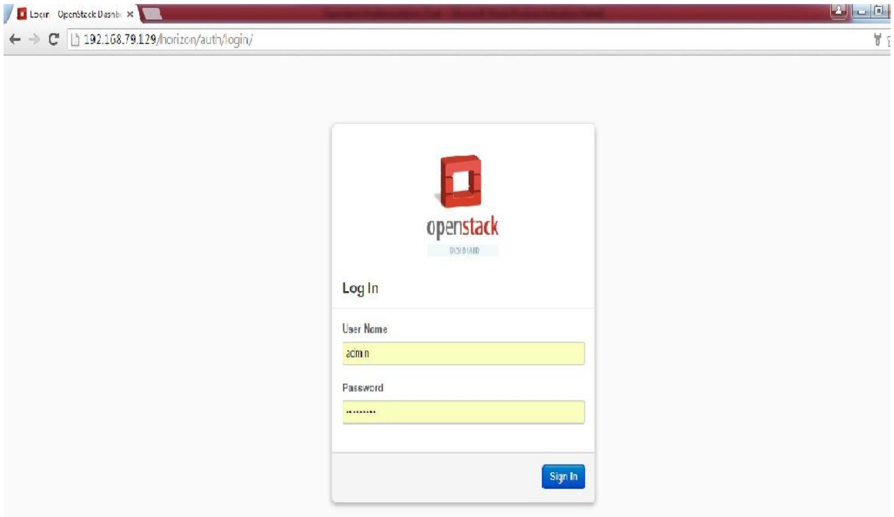
After successful login to the cirros operating system we need to check different configuration of the instance. First the "whoami" command is used to check the name of the user of the instance. Then "ifconfig" command is used to check the internet protocol address of the instance



```
root@Ubuntu:~#
root@Ubuntu:~#
root@Ubuntu:~# ssh cirros@10.10.100.2
$ whoami
cirros
$ ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:70:15:FD
          inet addr:10.10.100.2  Bcast:10.10.100.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fe70:15fd/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:467 errors:0 dropped:0 overruns:0 frame:0
          TX packets:387 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:71789 (70.1 KiB)  TX bytes:61626 (60.1 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

$
```

Figure 5. 39: Cirros configuration checking

After completing the entire configuration in cli mode we need to access OpenStack dashboard (which is also called Horizon) using the web browser. The uniform resource locator of the OpenStack dashboard will as follows: http://192.168.79.129/horizon.



192.168.79.129/horizon/auth/login/

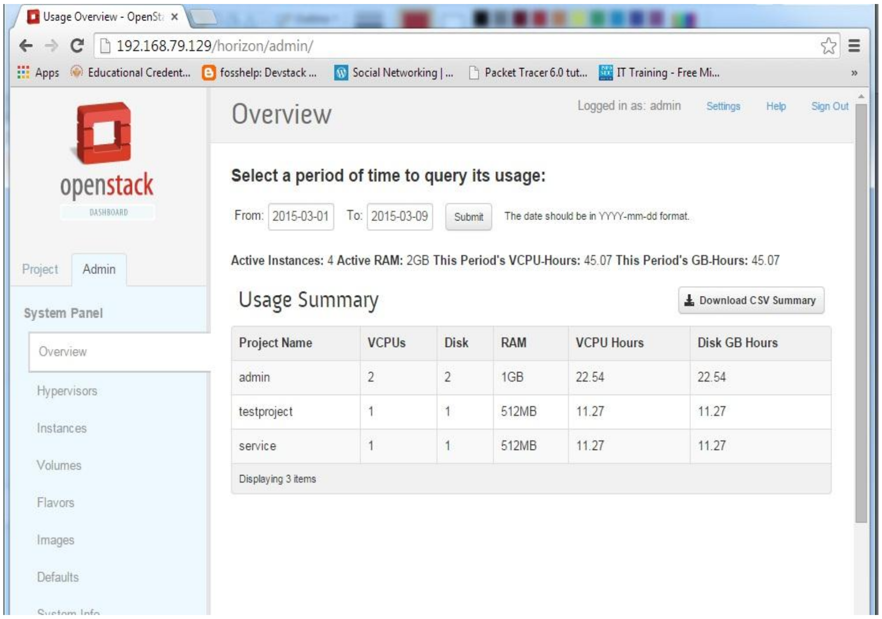
Log In

User Name: admin

Password: ••••••••

Figure 5. 40: Horizon login page

At the login screen there are two different accounts that can use to login. The first one is admin account. In admin account the username is admin and the password is OpenStack. After successful login as admin, the admin panel overview is shown below.



192.168.79.129/horizon/admin/

Overview

Logged in as: admin

Select a period of time to query its usage:

From: 2015-03-01 To: 2015-03-09

The date should be in YYYY-mm-dd format.

Active Instances: 4 **Active RAM:** 2GB **This Period's VCPU-Hours:** 45.07 **This Period's GB-Hours:** 45.07

Usage Summary

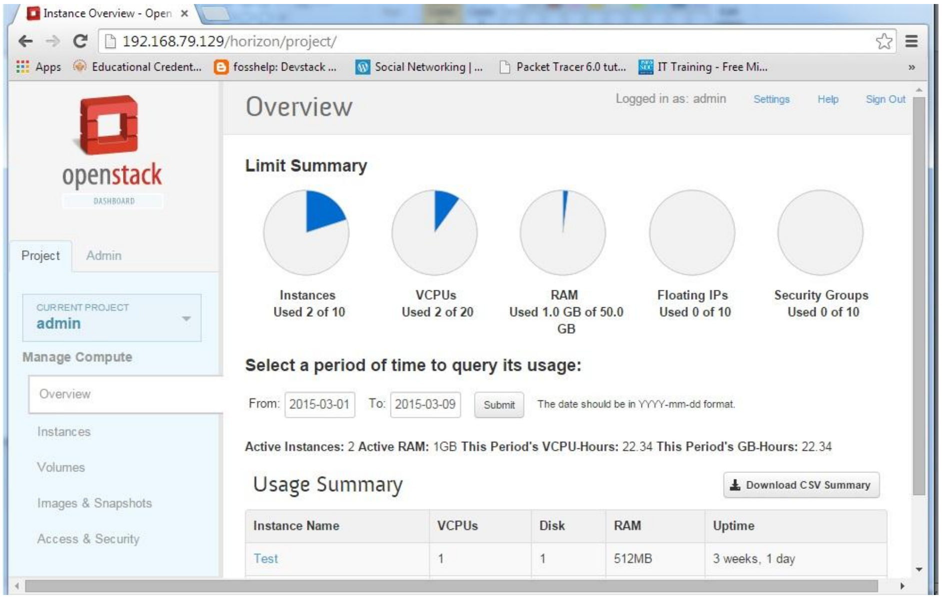
Project Name	VCPUs	Disk	RAM	VCPU Hours	Disk GB Hours
admin	2	2	1GB	22.54	22.54
testproject	1	1	512MB	11.27	11.27
service	1	1	512MB	11.27	11.27

Displaying 3 items

Figure 5. 41: Overview of the Admin Panel

If we click on the Project tab in the left hand pane then the usage summary will Usage

Summary for admin project will be displayed for the all instances. The pie charts are showing the running instances. The following figure shows the project overview.



Overview

Limit Summary

Instances	VCPUs	RAM	Floating IPs	Security Groups
Used 2 of 10	Used 2 of 20	Used 1.0 GB of 50.0 GB	Used 0 of 10	Used 0 of 10

Select a period of time to query its usage:

From: 2015-03-01 To: 2015-03-09 Submit The date should be in YYYY-mm-dd format.

Active Instances: 2 Active RAM: 1GB This Period's VCPU-Hours: 22.34 This Period's GB-Hours: 22.34

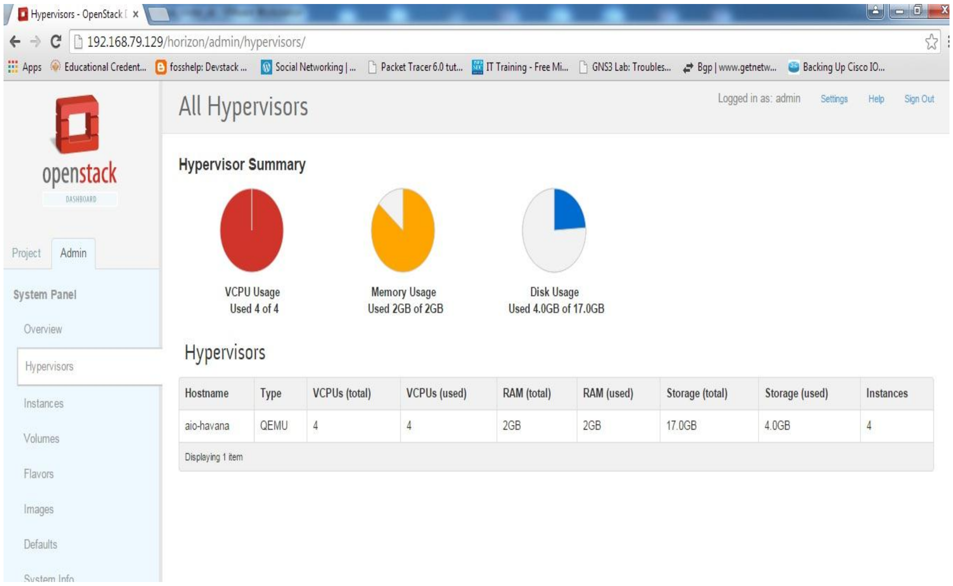
Usage Summary

Download CSV Summary

Instance Name	VCPUs	Disk	RAM	Uptime
Test	1	1	512MB	3 weeks, 1 day

Figure 5. 42: Project overview

In the admin panel all the hypervisors summary report can be displayed. The total resources used by the hypervisors are shown in the pie chart and in the tabular format.



All Hypervisors

Hypervisor Summary

VCPU Usage	Memory Usage	Disk Usage
Used 4 of 4	Used 2GB of 2GB	Used 4.0GB of 17.0GB

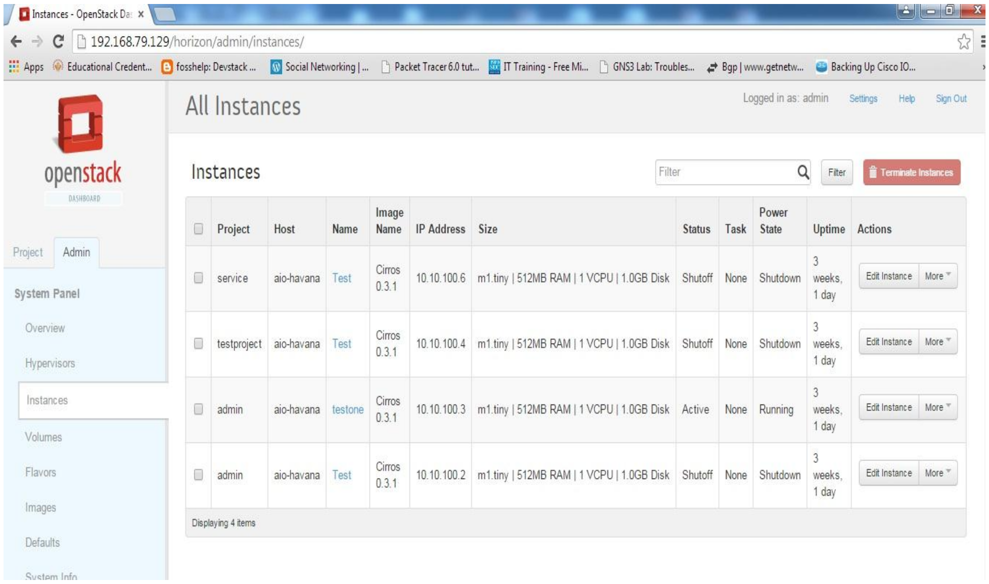
Hypervisors

Hostname	Type	VCPUs (total)	VCPUs (used)	RAM (total)	RAM (used)	Storage (total)	Storage (used)	Instances
aio-havana	QEMU	4	4	2GB	2GB	17.0GB	4.0GB	4

Displaying 1 item

Figure 5. 43: Hypervisors summary

In the admin panel user can see the total statistics of all the instances in a nutshell. The project name, hostname, image name, ip address, status, state of the instance and different actions can be seen in the instances tab.



All Instances

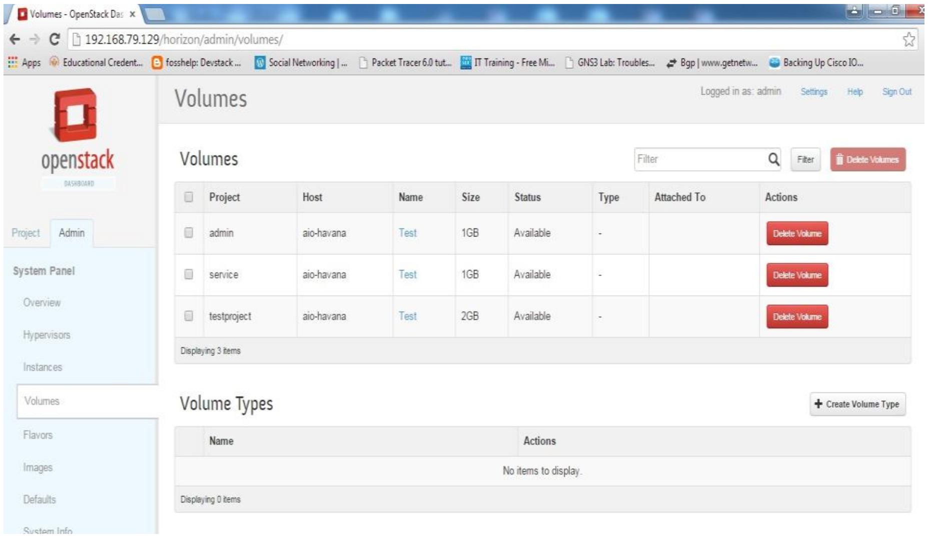
Instances

□	Project	Host	Name	Image Name	IP Address	Size	Status	Task	Power State	Uptime	Actions
□	service	aio-havana	Test	Cirros 0.3.1	10.10.100.6	m1.tiny | 512MB RAM | 1 VCPU | 1.0GB Disk	Shutoff	None	Shutdown	3 weeks, 1 day	Edit Instance More
□	testproject	aio-havana	Test	Cirros 0.3.1	10.10.100.4	m1.tiny | 512MB RAM | 1 VCPU | 1.0GB Disk	Shutoff	None	Shutdown	3 weeks, 1 day	Edit Instance More
□	admin	aio-havana	testone	Cirros 0.3.1	10.10.100.3	m1.tiny | 512MB RAM | 1 VCPU | 1.0GB Disk	Active	None	Running	3 weeks, 1 day	Edit Instance More
□	admin	aio-havana	Test	Cirros 0.3.1	10.10.100.2	m1.tiny | 512MB RAM | 1 VCPU | 1.0GB Disk	Shutoff	None	Shutdown	3 weeks, 1 day	Edit Instance More

Displaying 4 items

Figure 5. 44: Instances statistics

In the volume tab of the admin panel all the volume statistical data can be seen. User can create and delete volumes according to their needs.



Volumes

Volumes

□	Project	Host	Name	Size	Status	Type	Attached To	Actions
□	admin	aio-havana	Test	1GB	Available	-		Delete Volume
□	service	aio-havana	Test	1GB	Available	-		Delete Volume
□	testproject	aio-havana	Test	2GB	Available	-		Delete Volume

Displaying 3 items

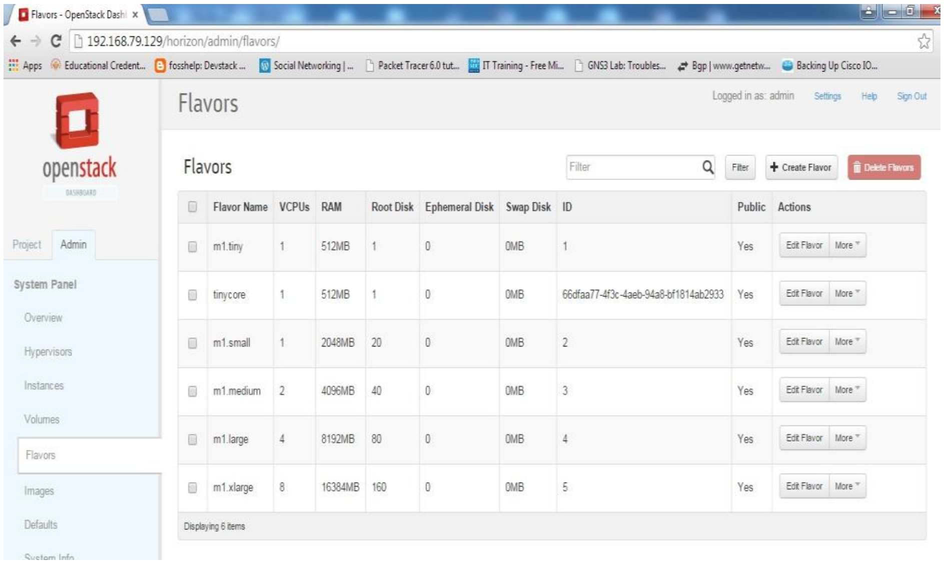
Volume Types

Name	Actions
No items to display.	

Displaying 0 items

Figure 5. 45: Volume statistics

In the flavors tab there are by default five different flavors which created during the install process of the OpenStack. The user can create different flavors based on their need. They can also delete and modify the existing flavors.



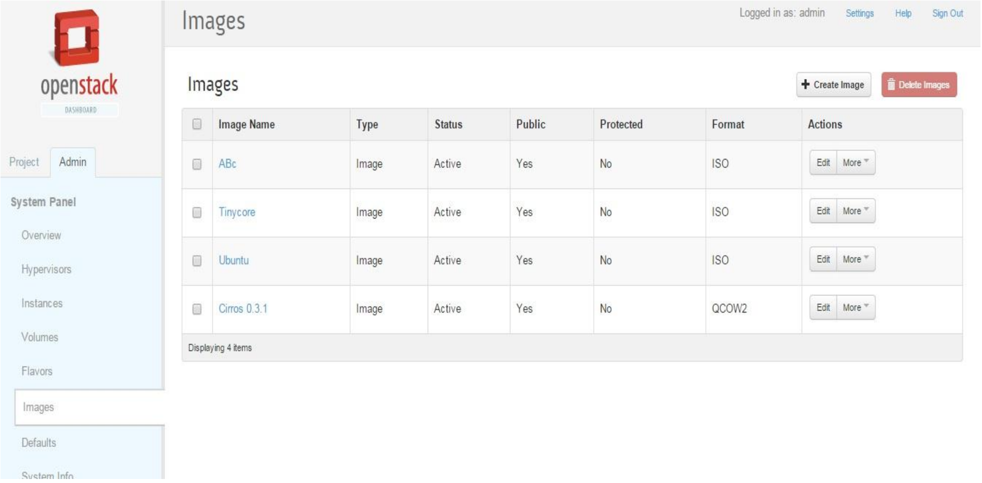
Flavors

□	Flavor Name	VCPUs	RAM	Root Disk	Ephemeral Disk	Swap Disk	ID	Public	Actions
□	m1.tiny	1	512MB	1	0	0MB	1	Yes	Edit Flavor More
□	tinycore	1	512MB	1	0	0MB	66dfaa77-4f3c-4aeb-94a8-bf1814ab2933	Yes	Edit Flavor More
□	m1.small	1	2048MB	20	0	0MB	2	Yes	Edit Flavor More
□	m1.medium	2	4096MB	40	0	0MB	3	Yes	Edit Flavor More
□	m1.large	4	8192MB	80	0	0MB	4	Yes	Edit Flavor More
□	m1.xlarge	8	16384MB	160	0	0MB	5	Yes	Edit Flavor More

Displaying 6 items

Figure 5. 46: Flavors statistics

Using the image tab, user can create their own image which will be used next to create the virtual instances. They can also delete and modify the images according to their need.



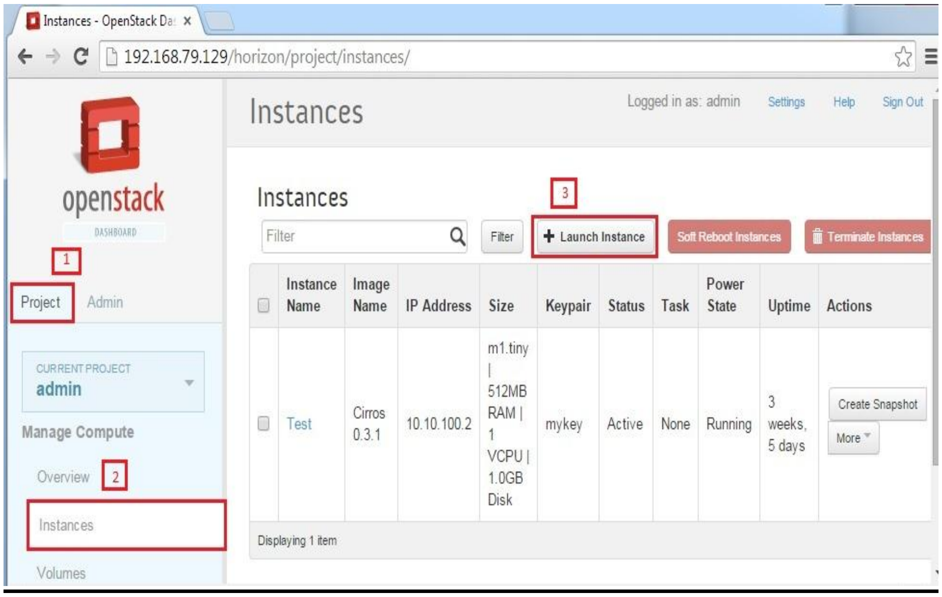
Images

□	Image Name	Type	Status	Public	Protected	Format	Actions
□	ABc	Image	Active	Yes	No	ISO	Edit More
□	Tinycore	Image	Active	Yes	No	ISO	Edit More
□	Ubuntu	Image	Active	Yes	No	ISO	Edit More
□	Cirros 0.3.1	Image	Active	Yes	No	QCOW2	Edit More

Displaying 4 items

Figure 5. 47: Image statistics

The dashboard is place where user can launch the virtual machine instance. In the left hand pane user need to click on the instances menu options and then user need click on the “Launce Instance” button at the top right of screen. The steps are shown in the diagram.

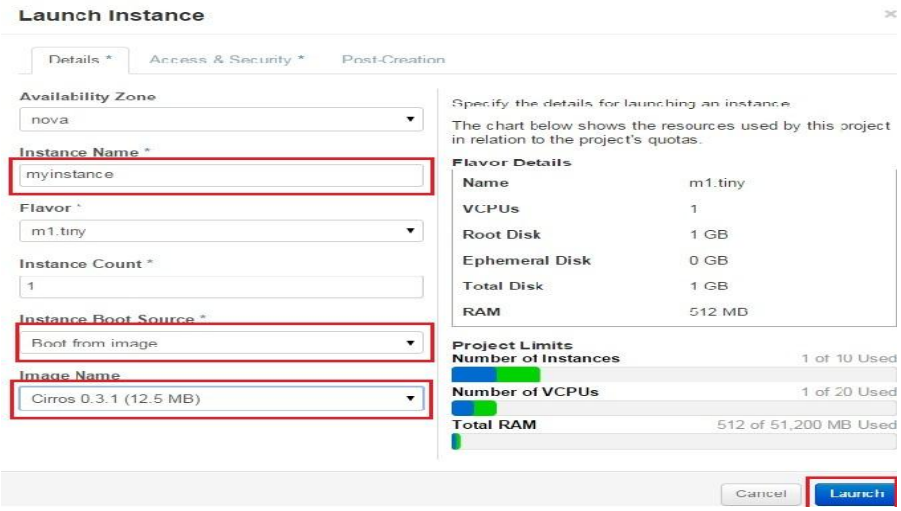


□	Instance Name	Image Name	IP Address	Size	Keypair	Status	Task	Power State	Uptime	Actions
□	Test	Cirros 0.3.1	10.10.100.2	m1.tiny | 512MB RAM | 1 VCPU | 1.0GB Disk	mykey	Active	None	Running	3 weeks, 5 days	Create Snapshot More

Displaying 1 item

Figure 5. 48: Launch Instances steps

On the Details tab there are a few fields that need to fill in to prepare instance. First user need to fill the name then from the instance boot source user need to select appropriate options. Then user will select the flavor and choose the boot source option. Finally user should choose the instance which will be installed in OpenStack cloud.



Launch Instance

Availability Zone: nova

Instance Name *: myinstance

Flavor *: m1.tiny

Instance Count *: 1

Instance Boot Source *: Boot from image

Image Name: Cirros 0.3.1 (12.5 MB)

Specify the details for launching an instance.

The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	m1.tiny
VCPUs	1
Root Disk	1 GB
Ephemeral Disk	0 GB
Total Disk	1 GB
RAM	512 MB

Project Limits

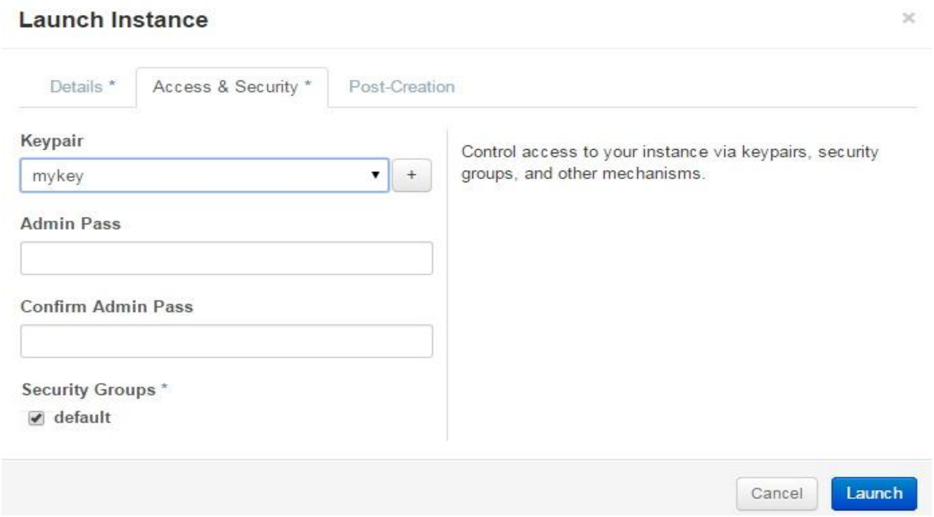
Number of Instances 1 of 10 Used

Number of VCPUs 1 of 20 Used

Total RAM 512 of 51,200 MB Used

Figure 5. 49: Preparing instances and selection of image

On the Access & Security tab, user needs to select the appropriate "keypair" to use. The "Security Groups" checkbox need to select for default. After providing all the information user need to click the "Launch" button.



Launch Instance

Keypair: mykey

Admin Pass:

Confirm Admin Pass:

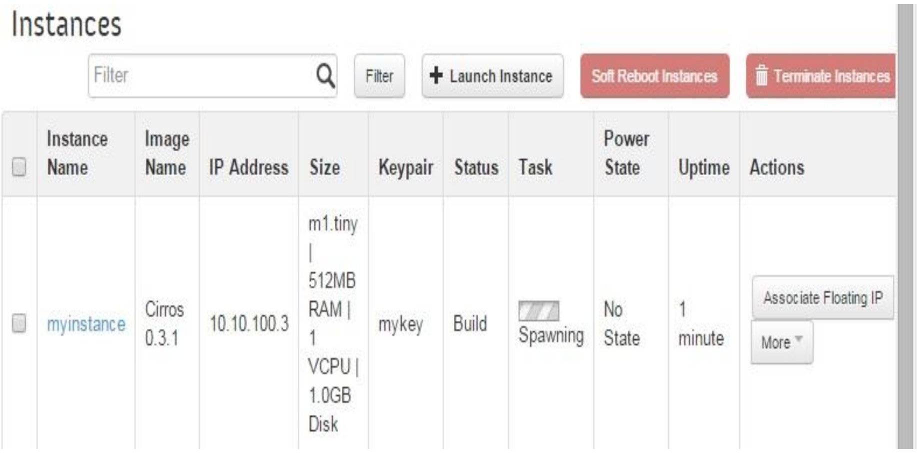
Security Groups *

☑ default

Control access to your instance via keypairs, security groups, and other mechanisms.

Figure 5. 50: Launching the instance

In the "Task" option of the instances it is showing "Spawning". After compliting the install thepower state will be running.



☐	Instance Name	Image Name	IP Address	Size	Keypair	Status	Task	Power State	Uptime	Actions
☐	myinstance	Cirros 0.3.1	10.10.100.3	m1.tiny | 512MB RAM | 1 VCPU | 1.0GB Disk	mykey	Build	Spawning	No State	1 minute	Associate Floating IP More ▾

Figure 5. 51: Booting myinstance

CHAPTER 6
CONCLUSION

Cloud Computing is the latest technology of provisioning scalable and reliable services over the internet. Different organizations rely on massively scalable IT infrastructure providers like Amazon for their cloud services. In this project we try to focus on one goal and that is to implement OpenStack IaaS cloud management software in a virtual environment using command line interface and using web browser.

The goal is achieved using the following several steps. The steps are: (1) installation and configuration of VMware workstation; (2) configuration of VMware network editor; (3) installation of Ubuntu server in VMware workstation and installation of different packages for preparing Ubuntu to hold OpenStack; (4) installation of different services of OpenStack and finally (5) configuration of OpenStack using command line interface and web interface.

Cloud computing is an attractive concept for small to lager enterprises as it brings lot of benefits on many levels. Now-a-days open source products play an increasingly important role in this market. There are several open source software solutions to implement IaaS (infrastructure as a service) cloud management software with rich feature sets which are comparable to commercial/closed sourced products.

REFERENCES

[1] The NIST Definition of Cloud Computing. [Online]. Available at http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. Last access on 2/02/2015.

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, "Above the Clouds: A View of Cloud Computing". [Online]. Available at http://mars.ing.unimo.it/didattica/ingss/Lec_SaS/CloudView.pdf. Last access on 2/05/2015.

[3] P. Sempolinski and D. Thain, "A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus". [Online]. Available at http://ccl.cse.nd.edu/research/papers/psempoli-cloudcom.pdf. Last access on 2/02/2015

[4] T. Cordeiro, D. Damalio, N. Pereira, P. Endo, A. Palhares, G. Gonçalves, D. Sadok, J. Kelner, B. Melander, V. Souza and J. Mångs, "Open Source Cloud Computing Platforms," in 9th International Conference on Grid and Cooperative Computing (GCC), Nanjing , Nov. 2010, pp.366-371.

[5] Cloud management software. Cloud management software list. [Online]. Available at: http://gigaom.com/2009/11/06/10-top-open-source-resources-for-cloud-computing/.

[6] Cloud management software. Cloud management software list. [Online]. Available at: http://socializedsoftware.com/2010/01/20/eleven-open-source-cloud-computing-projects-to-watch/.

[7] Elasticity in Cloud Computing: What It Is, and What It Is Not.[Online]. Available at: https://sdqweb.ipd.kit.edu/publications/pdfs/HeKoRe2013-ICAC-Elasticity.pdf

[8] Cloud Computing Service Models.[Online]. Available: http://cloud.cio.gov/topics/cloud-computing-service-models.

[9] Cloud Computing. [Online]. Available: http://en.wikipedia.org/wiki/Cloud_computing

[10] SaaS: Pros, cons and leading vendors.[Online]. Available at: http://www.zdnet.com/article/saas-pros-cons-and-leading-vendors.

[11]Benefits of SaaS. [Online]. Available: http://www.salesforce.com/au/saas/benefits-of-saas

[12] IaaS, PaaS, SaaS (Explained and Compared).[Online]. Available at: http://apprenda.com/library/paas/iaas-paas-saas-explained-compared/

[13]Platform-as-a-Service(PaaS).[Online].Available at: http://whatiscloud.com/cloud_delivery_models/platform_as_a_service.

[14] Cloud Security and Privacy by Tim Mather, Subra Kumaraswamy, and Shahed Latif, O'Reilly Media, Inc.,2009.

[15] What is a Private Cloud?.[Online]. Available at: http://www.interoute.com/cloud-article/what-private-cloud

[16] OPENSTACK TECHNOLOGY101 Boston OpenStack Meet-Up February 19, 2014. [Online]. Available at: http://www.slideshare.net.

[17] OpenStack. [Online]. Available at: http://en.wikipedia.org/wiki/OpenStack

[18] Amazon, "Amazon Elastic Compute Cloud (EC2)," Amazon Web Services LLC, Tech. Rep., 2009. [Online]. Available at: http://aws.amazon.com/ec2/

[19] Mosso, "Deploy and scale websites, servers and storage in minutes," Rackspace, Tech. Rep., 2009. [Online]. Available at: http://www.mosso.com/

[20] Hwang, J.,Park, J.,"Decision Factors of Enterprises for Adopting Grid Computing, Grid Economics and Business Models", 4th International Workshop, GECON, 2007.

[21] R. Buyya, C. Yeo, and S. Venugopal, "Market-oriented Cloud Computing: Vision, hype, and reality for delivering it services as computing utilities," in High Performance Computing and Communications. IEEE Press, Sept 2008.

[22] Google, "Google App Engine: Run your web apps on Google's infrastructure." Google, Tech. Rep., 2009. [Online]. Available at: http://code.google.com/appengine

[23] http://en.wikipedia.org/wiki/Eucalyptus_(software)

[24] https://www.eucalyptus.com/eucalyptus-cloud/iaas

[25] https://cloudstack.apache.org/

[26] http://en.wikipedia.org/wiki/Apache_CloudStack

[27] http://www.slideshare.net

[28] http://en.wikipedia.org/wiki/OpenNebula

[29] http://opennebula.org/about/technology/

[30] http://ccl.cse.nd.edu/research/papers/psempoli-cloudcom.pdf

[31]http://opennebula.org/openstack-cloudstack-eucalyptus-and-opennebula-which- cloud-platform-is-the-most-open/

[32] http://opennebula.org/about/key-features/

[33] Thomas J. Hacker, "Exploring the Use of Virtual Machines and Virtual Clusters for High Performance Computing Education", 2011 Annual Conference & Exposition, June 2011

[34] Robayet Nasim, Andreas J. Kassler, “Deploying OpenStack: Virtual Infrastructure or Dedicated Hardware”, Computer Software and Applications Conference Workshops (COMPSACW), 84 – 89, July 2014

[35] http://en.wikipedia.org/wiki/OpenStack

[36] http://www.openstack.org/software/

TABLE OF FIGURES