A Report on the Project

OpenStack- Orchestrate Public and Private Cloud using OpenStack/Vcenter integration

Submitted by

Steave Dsouza

In partial fulfillment for the award of the degree

Master of Science in Internetworking

(From University of Alberta)

Under the guidance of

Muhammad Durrani

September 2016 - March2017

ABSTRACT

With the advent of cloud computing technology, many industries have been debating the business case scenarios for hybrid clouds. The opportunities and capabilities offered by cloud computing are tremendous taking into account the on-demand access of IT resources and the scope for application development. With different vendors like Amazon, OpenStack, VMware, salesforce etc. offering cloud solutions, it all boils down to the application that best suits a business needs. Integrating the cloud technology with network virtualization presents benefits and makes a business case very hard to ignore. Programmatic API access to infrastructure and vendor neutral API facilities are two of the most important factors to weigh in on when deploying hybrid cloud scenarios over a virtual network. For my project, I will be considering the integration of OpenStack's cloud platform with the network virtualization and management capabilities of VMware Suite.

ACKNOWLEDGEMENT

I offer my most sincere thanks to my project guide Mr. Muhammad Durrani for his expert help that has guided me throughout the making of this project. His invaluable guidance and support has always motivated me to work harder. His approachability, cooperation and encouragement provided me with great enthusiasm and confidence throughout this project.

I also wish to express my deep gratitude to Mr. Shahnawaz Mir for his valuable advice, suggestions and helping out with the lab equipment during the course of the project.

I extend thanks to all my classmates whom I have approached for academic help with regard to this project.

Steave Dsouza

Contents

Chapter 1 Introduction 9
1.1 Importance of project
1.2 Motivation
1.3 Organization of report

Chapter 2 Literature Review 11
2.1 Amazon Web Services
2.2 Microsoft Azure
2.3 Google Cloud
2.4 2.4 Openstack vs AWS, Microsoft Azure and Google cloud

Chapter 3 Overview of Openstack 13

Chapter 4 Pre-requisites, Resources and expected results 18
4.1 Pre-requisites
4.2 Flow Diagram
4.3 Deployment Sequence
4.4 Expected Results

Chapter 5 Actual Implementation 21
5.1 Verifying Vsphere & Esxi setup
5.2 Vcenter Deployment
5.3 Creating Datacenters and Adding Hosts
5.4 Vmware Integrated Openstack OVA deployment
5.5 Vmware Integrated Openstack Instance deployment
5.6 Openstack Configuration

5.7 Snort IDS

Chapter 6 Actual vs Expected Results & Conclusion 56

Chapter 7 References 58

List OF Abbreviations

VIO	Vmware Integrated Openstack
OVA	Open Virtualization Appliance
AWS	Amazon Web Services
Iaas	Infrastructure-as-a-service
SDN	Software Defined Networking
SDDC	Software Defined Data Center
DRS	Disaster Recovery Server
HA	High Availability
VSAN	Virtual Storage Area Network
VDS	Virtual Distributed Switch
IDS	Intrusion Detection System
IPS	Intrusion Prevention System

List of Figures

Figure No.	Figure Captions	Page No.
Fig 3.1	Openstack Components [4]	13
Fig 3.2	VIO Components and their integration [5]	15
Fig 3.3	VMware and Openstack implementation before VIO [5]	16
Fig 3.4	VIO internal architecture [5]	17
Fig 4.1	Flow Diagram	19
Fig 5.1	Vsphere and Esxi version	21
Fig 5.2	Vcenter Setup page	22
Fig 5.3	Target Server Details	23
Fig 5.4	Embedded Vcenter install	24
Fig 5.5	Selecting the Storage.	24
Fig 5.6	Vcenter Network settings	25
Fig 5.7	Final overview before deployment	26
Fig 5.8	Vcenter Logon screen	26
Fig 5.9	Vcenter home page	27
Fig 5.10	Datacenter creation	27
Fig 5.11	Cluster specifications [6]	28
Fig 5.12	OVA source from Local Directory	29
Fig 5.13	Verify the OVA details	30
Fig 5.14	Provide the folder for deployment	30
Fig 5.15	Select Storage and Thin provision it	31
Fig 5.16	Provide Networking Details	31
Fig 5.17	Review before final deployment	32
Fig 5.18	Original values of omjs.properties	33
Fig 5.19	New values of omjs.properties	34
Fig 5.20	Status of the Plugin	37
Fig 5.21	Extension manager for Vcenter	38
Fig 5.22	VIO plugin on Vcenter Homepage.	38
Fig 5.23	Connection verification of VIO	39
Fig 5.24	Deploy Openstack homepage	39
Fig 5.25	Select Compact Deployment type	40
Fig 5.26	Vcenter Credentials to Deploy VIO	40
Fig 5.27	Networking Details	41
Fig 5.28	Glance Datastore	41
Fig 5.29	Virtual distributed switch networking	42
Fig 5.30	Openstack Credentials to access Webpage	42

Fig 5.31	Review VIO instance details	43
Fig 5.32	Deployment Progress.	43
Fig 5.33	Status of VIO deployment	44
Fig 5.34	VIO module status	44
Fig 5.35	Virtual Distributed Switch	45
Fig 5.36	esxi VDS	45
Fig 5.37	Openstack login page	46
Fig 5.38	Openstack Projects	47
Fig 5.39	Openstack Cloud admins & users	47
Fig 5.39	Functionalities available to administrator	48
Fig 5.40	Functionalities available to Users	48
Fig 5.41	VIO images	49
Fig 5.42	VIO Networks	49
Fig 5.43	VIO router	49
Fig 5.44	VIO router interfaces	50
Fig 5.45	VIO Network topology-1	50
Fig 5.46	VIO Network topology- 2	51
Fig 5.47	VIO Key pairs	51
Fig 5.48	VIO Security Groups	52
Fig 5.49	Access-To-Network Security Group	52
Fig 5.50	Allow-SSH-from-Local	53
Fig 5.51	SQL security group	53
Fig 5.52	Secret-Backdoor-Communication security group	54
Fig 5.53	Web and HTTPS security group.	54
Fig 5.54	Snort running on 10.3.32.116	55
Fig 6.1	Neutron Error	56

Chapter 1

Introduction

1.1 Importance of the Project

Cloud computing has seen tremendous amount of growth in the past few years. The advantages that this technology brings to the table is very hard to ignore. Cloud computing is a continuity to the internet itself. Before cloud computing, we would download an application on our physical devices like computers, cellphones etc. to avail certain services, however this had to be done for every single customer of that business which resulted in security lapses, uses of computing resources which were ill managed and overall caused a depreciation in the service being provided to the customers. Cloud computing did away with all of this. Some of the few examples are checking your bank balance. Every time you log on to the bank website to check your balance you are using cloud, updating your status on Facebook is done by means of a cloud. The basic idea of introducing cloud was to simplify this simple tasks while ensuring proper management of resources. Present day there are numerous vendors that offer custom cloud solution with each one having their own patented cloud technology and services and it is very essential for a business to study and evaluate them carefully to deem which one of them fits their needs and is the most effective solution for them.

1.2 Motivation

There are many vendors who offer cloud solutions. Most prominent among them are Amazon, Google, Microsoft, Rackspace, HP and IBM. Among these, Rackspace is the only one offering an open source solution. Rackspace in collaboration with NASA founded the Openstack initiative. A cloud solution which is available to everyone with experts around the world contributing codes and features. It is updated yearly with new features in the form of modules added to it. OpenStack is an open to all set of tools used to govern and create different aspects of a cloud. OpenStack provides a framework to develop and deploy Infrastructure-as-a-service (Iaas) cloud. The cloud style API's thus created are perfect and justify the consumption of virtual infrastructure technologies. However, OpenStack itself does not provide any of the virtual technology like a hypervisor, networking or storage. It also leverages most of the cloud management tools such as policies, governance, monitoring etc. from the underlying hypervisor/network virtualization vendors. OpenStack cannot provide full functionality on its own and must rely on different vendors in order to provide the most optimized solution for a business. Present day VMware is the most widely used virtualization platform and such implementation of Openstack on VMware provides numerous opportunities to optimize the way in which a cloud can be managed.

1.3 Organization of the Report

Chapter 2: A brief comparison between the different cloud based solutions out there and how Openstack compares to them. The discussion is limited to the prominent ones who have a considerable market share in terms of cloud industry.

Chapter 3: A brief overview on the Openstack suite that includes its features, its advantages and some of the features that were added in past couple of years

Chapter 4: A list down of all the pre-requisites for implementing this project, the sequence in which each of them will be executed and the expected results.

Chapter 5: The actual steps, snapshots, and issues faced in implementing them, the workarounds implemented to resolve them and the final result. The captured logs and mark down critical entries for future reference

Chapter 6: This deals with conclusion of the result and a brief explanation regarding expected vs actual results.

Chapter 7: This includes the references and a list of resources that were used for successful completion of the project and the report.

Chapter 2

Literature review

2.1 AWS (Amazon web services)

Among all the vendors for cloud computing technology Amazon web services occupy the biggest market share with more than 30% as per the survey conducted by Synergy research group [1]. The year over year growth being 53%. AWS offers its proprietary solution for cloud computing and takes advantage of its years of web service dominance. Some of the benefits that AWS advocates and the reason why it has managed to garner such a huge market share are its ease of use, flexibility, Reliability, Scalable and high performance, security and cost effectiveness. One of the primary reason why AWS seems to be so successful and popular is its no commitment policy. Almost all its server backed services are charged hourly and terminating a server stops the billing the very next hour. This approach also applies to companies looking to leverage commercial software which are very expensive and difficult to procure. Amazon web services offer Amazon marketplace where majority of this software's are available on an hourly charged basis. Since its inception in 2006, AWS has come a long way with offerings for every business model. The importance of AWS as a division to the company can be recognised by the fact that in the first quarter of 2016, Amazon experienced a 42% rise in stock value as a result of increased earnings, of which AWS contributed 56% to company's profit. With a 50% increase in revenues the past few years, it is predicted AWS will have $13 billion in revenue in 2017 which is very substantial.

2.2 Microsoft Azure

Microsoft Azure with a market share of 11% leads the chase towards cloud market dominance and is second in race after AWS [1]. Microsoft Azure is built around Microsoft's proprietary technology i.e. the fabric layer. It is a cluster based service hosted on their datacenters which provides software as a service, platform as a service and infrastructure as a service and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems. It also runs its own customised version of Hypervisor called the Azure hypervisor to provide virtualization services [2]. One look at the company's website on why Azure should be adapted and we quickly realise that its main business focus is large scale institutions and government installations. With features like artificial intelligence available at a premium price to improve the overall customer experience and build an app that truly delivers on the company's values Azure does deliver on the price point and hence has seen a tremendous growth in recent years. As per Microsoft's official statement its cloud business comprising of Azure and office 365 has a run rate of $12 billion. With a 100% year over year growth Azure has made a substantial breakthrough in the cloud marketspace and is evident from different results.

2.3 Google Cloud

Initially released on October 6th 2011 google cloud platform has made some headway into the cloud computing market. Currently ranked 4th with a market share of approximately 5% google is still playing catch up to Amazon and Microsoft [1]. However, the pest part about Google's cloud platform is that it leverages the same internal infrastructure used for google search and YouTube to provide the cloud based services. It approaches or markets itself as modular cloud based service with a number of development and management tools. The google cloud is very similar to the amazon Azure offering in the sense that it has modules and services matching every single one of the AWS suite. However the introduction of the platform isn't as good as google would have expected. Over the years google had to introduce more features like the managed virtual machines to overcome the limitation of google app engine. Google has been marketing the cloud platform to all levels of business include start-up's. This was evident from the massive price drop that google introduced in March 2014 affecting all products in the range of 30 – 85%.

2.4 Openstack vs AWS, Microsoft Azure & Google cloud

There are many resources out there that would make a strong business case for using either AWS, Azure or google. However the most compelling one that I found was provided on the Openstack website and simply put the source code for Openstack is freely available under the apache 2.0 licence [3]. If we were building a cloud platform for 5000 to 10,000 hosts using solutions like VMware or Citrix is beneficial, however when building a platform for 100,000+ hosts this solutions become unsustainable. Using AWS, Azure or google is one solution however each one has their own special solution and we are restricted by functionality offered by that particular vendor. Even though Openstack is open source there is also the case that a third party has to be hired to provide support for the application itself. The one advantage about Openstack that I find compelling enough is that the rate at which it has grown. With developers all around the world and even well-known companies like VMware, IBM contributing the new releases and modules being added and the functionality they provide has been astounding. Currently, Openstack follows a six month three module cycle. Every six months there is a summit held where developers from all around the world contribute to suggestions, the ongoing issues and challenges and ways to counter them. The Openstack growth has been tremendous. Virtualization vendors like VMware introduced the VMware integrated Openstack module to simplify the deployment and also allow customers using VMware suites to leverage their existing infrastructure and deploy Openstack. With features like VMware's high availability and migration with zero downtime Openstack may soon become a market leader and promises a strong potential for any organization.

Chapter 3

Overview of Openstack

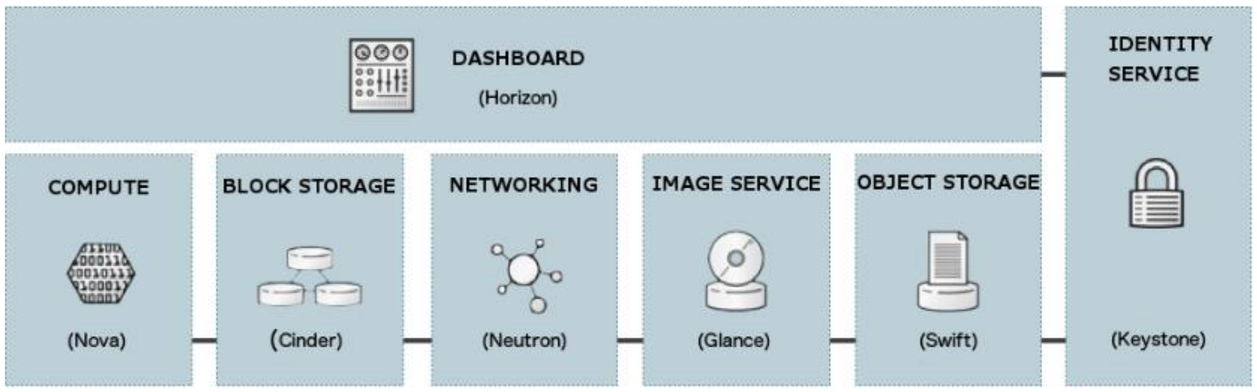


Fig 3.1 Openstack Components [4]

The picture above depicts the basic components of Openstack. Openstack essentially is a modular service and allows its users to deploy and configure any of its service or components as and when needed. The ones depicted above are the core components essential to Openstack's function. A brief overview of each of this component is provided below. All of the information on this component has been referenced form Openstack's knowledge base articles and documentation provided on the official website. Please follow the reference [4] for an in-depth explanation for each of it.

1. Compute (nova)

OpenStack Compute service provides services to support the management of virtual machine instances at scale, instances that host multi-tiered applications, dev/test environments, "Big Data" crunching Hadoop clusters, and/or high performance computing. The Compute service facilitates this management through an abstraction layer that interfaces with supported hypervisors. The security of Compute is critical for an OpenStack deployment. Hardening techniques should include support for strong instance isolation, secure communication between Compute sub-components, and resiliency of public-facing API endpoints.

2. Object Storage (swift)

The OpenStack Object Storage service provides support for storing and retrieving arbitrary data in the cloud. The Object Storage service provides both a native API and an Amazon Web Services S3 compatible API. The service provides a high degree of resiliency through data replication and can handle petabytes of data. It is important to understand that object storage differs from traditional file system storage. It is best used for static data such as media files (MP3s, images, and videos), virtual machine images, and backup files. Object security should focus on access control and encryption of data in transit and at rest. Other concerns may relate to system abuse, illegal or malicious content storage, and cross authentication attack vectors.

3. Block Storage (cinder)

The OpenStack Block Storage service provides persistent block storage for compute instances. The Block Storage service is responsible for managing the life-cycle of block devices, from the creation and attachment of volumes to instances, to their release. Security considerations for block storage are similar to that of object storage.

4. Shared File Systems (manila)

The Shared File Systems service provides a set of services for management of shared file systems in a multi-tenant cloud environment, similar to how OpenStack provides for block-based storage management through the OpenStack Block Storage service project. With the Shared File Systems service, you can create a remote file system, mount the file system on your instances, and then read and write data from your instances to and from your file system.

5. Networking (neutron)

The OpenStack Networking service provides various networking services to cloud users (tenants) such as IP address management, DNS, DHCP, load balancing, and security groups (network access rules, like firewall policies). It provides a framework for software defined networking (SDN) that allows for pluggable integration with various networking solutions. OpenStack Networking allows cloud tenants to manage their guest network configurations. Security concerns with the networking service include network traffic isolation, availability, integrity and confidentiality.

6. Dashboard (horizon)

The OpenStack Dashboard provides a web-based interface for both cloud administrators and cloud tenants. Through this interface administrators and tenants can provision, manage, and monitor cloud resources. Horizon is commonly deployed in a public facing manner with all the usual security concerns of public web portals.

7. Identity service (keystone)

The OpenStack Identity service is a shared service that provides authentication and authorization services throughout the entire cloud infrastructure. The Identity service has pluggable support for multiple forms of authentication. Security concerns here pertain to trust in authentication, management of authorization tokens, and secure communication.

8. Image service (glance)

The OpenStack Image service provides disk image management services. The Image service provides image discovery, registration, and delivery services to the Compute service, as needed. Trusted processes for managing the life cycle of disk images are required, as are all the previously mentioned issues with respect to data security.

9. Data processing service (Sahara)

The Data Processing service provides a platform for the provisioning, management, and usage of clusters running popular processing frameworks. Security considerations for data processing should focus on data privacy and secure communications to provisioned clusters.

VMware integrated Openstack (VIO)

VMware for a long time has been Openstack's top contributor since they joined the project in 2012. Their main focus was to integrate as many features as possible of the NSX virtualization suite with Openstack's open API's. However the fact remained that the Openstack deployment had to be done on different platforms like CentOS, Ubuntu etc. and even though VMware provided a differentiating value, the VMware administrators still had to learn the skill set to deploy it on different machines and then integrate it with their current infrastructure. In order to simplify this VMware launched the VMware integrated Openstack (VIO) module in March 2015. Ever since the first release VMware has made substantial changes to the module and they most recently launched the version 3.1 on 21st February 2017.

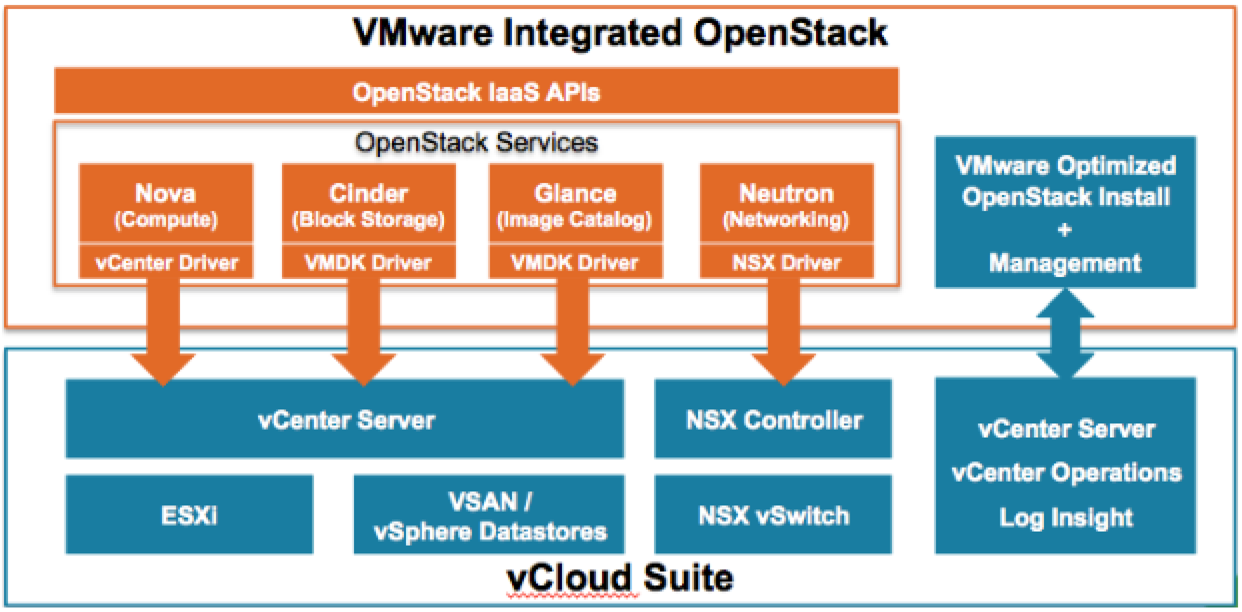


Fig 3.2 VIO Components and their integration [5]

Prior to VIO the integration of Openstack with VMware was a bit complex the difference is evident from the below diagrams that outline the architecture and integration differences before and after introduction of VIO i.e. figures 3.3 & 3.4

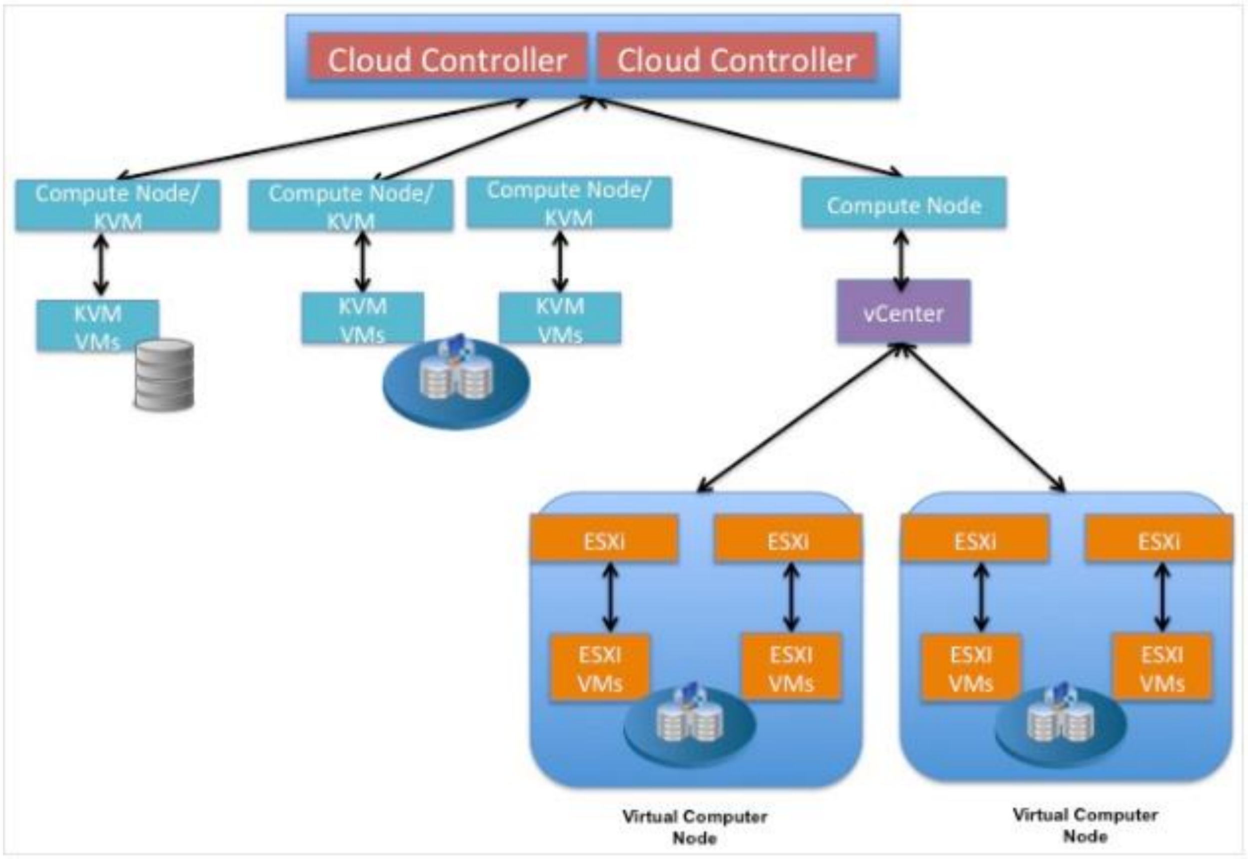


Fig 3.3 VMware and Openstack implementation before VIO [5]

The VIO package was developed by VMware as an OVA and was delivered as a vApp (Virtual Appliance) to the VMware infrastructure. In this scenario Openstack was deployed as an Overcloud while the VMware infrastructure acted as an Under-cloud. All of Openstack's control plane services and compute modules enabled administrators to use various plugins and networking/storage technologies from different vendors for their Openstack project. This made the Openstack Administrators independent of their reliance on admin assistance for infrastructure assistance. However the best capability to emerge out of this integration module was that admin's were now able to leverage underlying Vsphere capabilities like High availability, Disaster recovery and Vmotion.

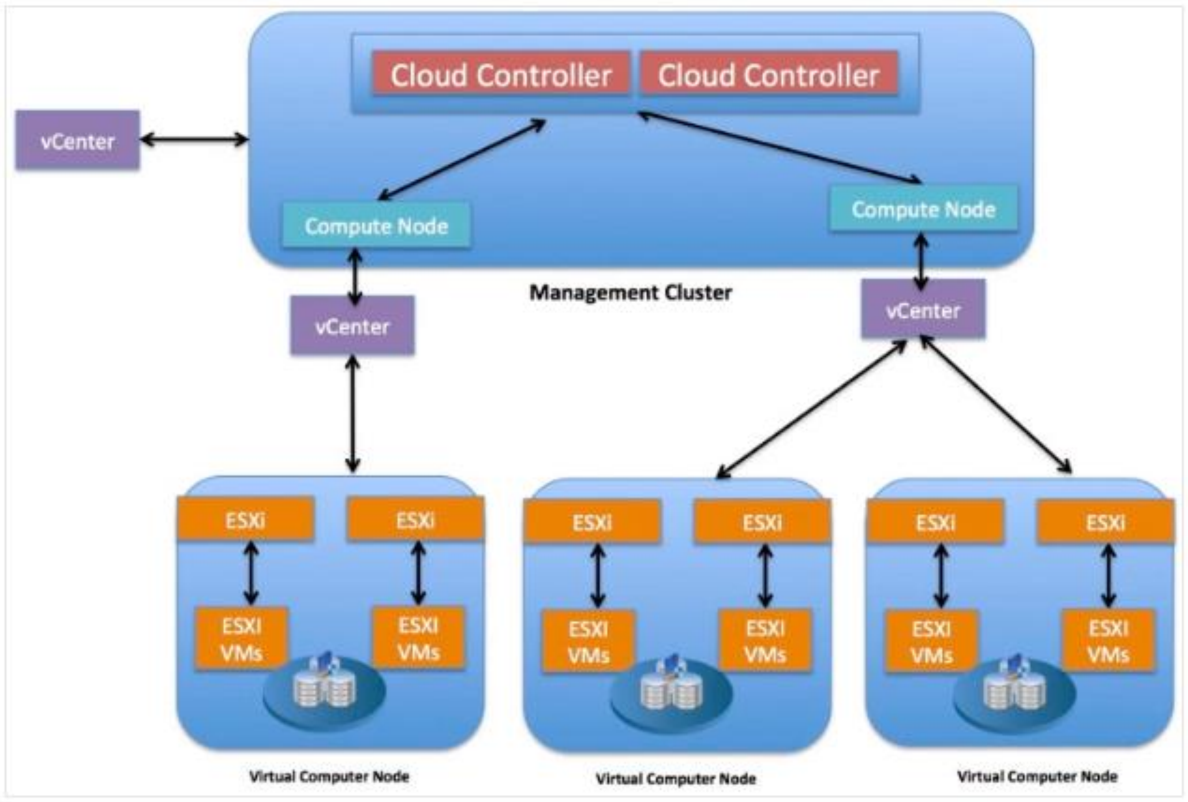


Fig 3.4 VIO internal architecture [5]

Even though Openstack under the Apache license is free, to make it a production environment for an enterprise grade deployment requires a lot of resources that includes third party vendors and support personnel. With VIO this was all reduced to a fraction of the cost. Their main objectives as outlined by the company are as below [5]:

1. VIO will empower Administrators to successfully deliver and operate production grade Openstack
2. Openstack delivered on best of breed SDDC products (software defined Data centers)
3. Proprietary features of NSX suite available to Openstack Admins
4. Leverage existing infrastructure to manage and deploy Openstack
5. Built in Automated workflows
6. Completes support for all Openstack components and its underlying infrastructure.

Chapter 4

Pre-requisites, Resources and expected results

4.1 Pre-requisites

Most of the documentation that is available on VMware websites and Openstack's official deployment guides are all based on deploying a production grade environment. For my project I am going to deploy VIO for a home lab or a small lab environment. VIO allows two types of deployment i.e. HA (High availability) & Compact. I have deployed a Compact Openstack because it requires very less resources, resources that are available to me in the lab. The main disadvantage of Compact deployment is that not all of Openstack's components are installed. And installing all of them at once causes the Hypervisor to run out of memory and reboot. I have only installed the components to justify my implementation for a small cloud.

1. **Hypervisor**: The server on which esxi runs. The lab servers run the latest esxi version 6.0 which is available as a free product from VMware. However, The Vsphere required to access the esxi has a licensing cost.
2. **VMware Vsphere**: I have installed the latest Vsphere 6.0 version. The license for this product is available on the universities one hub website.
3. **Vcenter:** The latest version of Vcenter i.e. 6.0 is also available for download and licensing from the one hub website.
4. **VMware integrated Openstack 3.0**: As of writing this report VMware launched a new version of VIO i.e. 3.1, but for the scope of this project I have used version 3.0. It is available for free download from the VMware website, the link for which is Link.
5. **Vmanager**: I would have preferred using Vmanager for my deployment. But unfortunately it is not available for licensing on the universities website. Instead as an alternative I have used a Vsphere distributed switch for my deployment which is part of the VMware Vcenter suite.
6. **Vshield**: Vshield helps secure the overall environment and is one of the major component for my project, unfortunately it is not available as a part of the licensing deal that university has with VMware and hence alternative approach was used.
7. **Datastore**: VMware & Openstack recommend the use of SSD's (Solid state drives) for data storage. I have used the VMFS Datastore available to us which provides substantial capacity. It does affects the fluidity of the operation but it will do from a lab environment deployment point of view.
8. **VMware Plugins**: There are certain plugins required to install the Vcenter, launch the console windows for the servers and for deploying Openstack itself.
9. **IP blocks**: Openstack deployment requires the use of a block of IP reserved specifically for its use to deploy different modules as virtual machines. This block can also be used for floating IP's in the Openstack console.

10. **DRS**: Disaster recovery servers are a must and an essential as per the deployment guidelines specified by VMware and Openstack. I only had the one server available so I used another server for the deployment and turned DRS off as soon as the deployment was complete.

4.2 Flow diagram

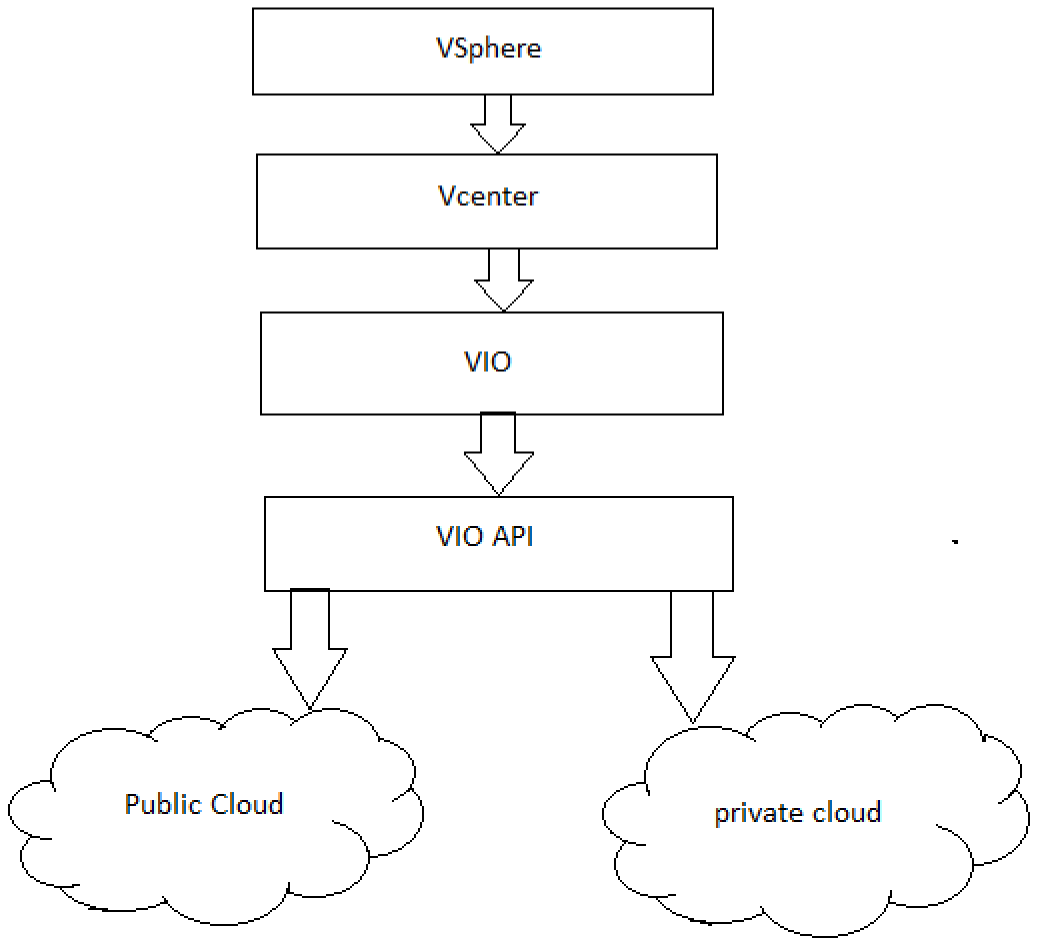


Fig 4.1 Flow Diagram

4.3 Deployment Sequence

The step by step execution of this deployment will be as below.

1. The initial step was to verify that proper access was provided to the server. The server was pre-installed with the esxi version 6.0 which served the purpose for this project.
2. Installing Vsphere 6.0 and assigning proper license to that instance was the next step.
3. Vcenter comes next. It includes, downloading the setup for Vcenter, deploying it in Vsphere server and assigning proper roles, access and license.
4. Creating a datacenter and adding the hosts to it was the next step. The importance of datacenter and their use is explained while listing the actual steps of execution.
5. Deploying VMware integrated Openstack management server on the datacenter was the next part. This server is just a manager that helps gain access to the actual VIO plugin which is then used to deploy Openstack instances.
6. Creating a Vsphere Distributed switch for deploying VIO instance to aid in networking.
7. Deploy the VIO instance on Vcenter while assigning proper resources.
8. Access the Openstack API, create a private and a public cloud.
9. Do proper networking on the router for both the clouds.
10. Create Admin and user accounts for the respective clouds.
11. Secure the clouds by means of rules and IPS/IDS deployment.

4.4 Expected Results

An expected deployment should have proper hierarchy in terms of VIO deployment and its modules. The networking using Vsphere distributed switch should be reflected on the host. API access to Openstack and a network topology view of the deployed cloud should be visible. Also, roles of different users, the security of cloud and the access to the system should be verified.

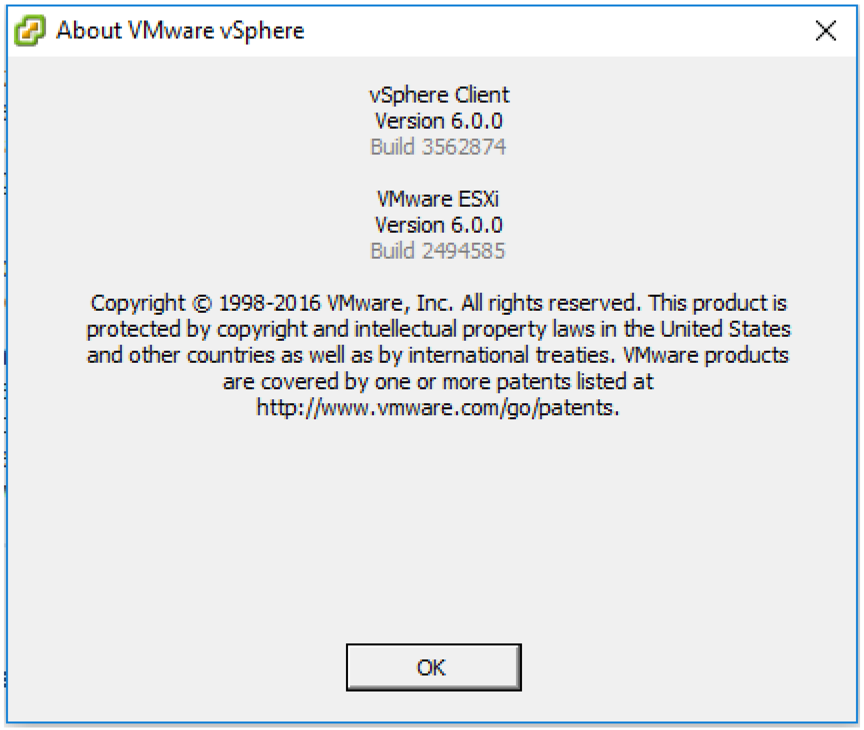
Chapter 5

Actual Implementation

I have divided this chapter in 5 sections. Each section deals with installation, issues and workaround of a specific module in the project

5.1 Verifying Vsphere and esxi setup.

The esxi was already installed on the Lab server and consisted of the latest version. The esxi is a free software/OS available through the official VMware website for download. Its installation is pretty easy and straightforward and there are numerous guides and videos available as such for reference. VMware has their own official guide for deploying esxi on a server and its best practices are outlined. The figure below details the version of Vsphere and esxi used.


About VMware vSphere

vSphere Client
Version 6.0.0
Build 3562874

VMware ESXi
Version 6.0.0
Build 2494585

Copyright © 1998-2016 VMware, Inc. All rights reserved. This product is protected by copyright and intellectual property laws in the United States and other countries as well as by international treaties. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents.

Fig 5.1 Vsphere and Esxi version

Some of the technical details regarding the IP addressing are as below:

IP address: 10.3.32.107/25

Gateway: 10.3.32.126/25

Host & Domain name: esxi.mint.local

DNS Servers: 10.3.31.10 & 129.128.5.233

IP addressing plays a very important role while deploying Vcenter and VIO. They are essentially required to be on the same subnet for the deployment to satisfy all possible conditions. Apart from this, I also reserved an IP block 10.3.34.0/27 for cloud deployments if required.

5.2 Vcenter Deployment

The next step is to deploy Vcenter on the esxi host. As mentioned previously in Chapter 4, Vcenter is readily available for download on the VMware website however a valid license has to be purchased and the free trial doesn't provide access to all functionalities of the suite. The one hub website enabled me to download a copy and a free license for a year.

An iso file for Vcenter is what is required. Once the iso file is opened the setup is pretty straightforward and a web based setup manager is available to guide through the installation. A client integration plugin installs automatically once the web base setup manager is launched.

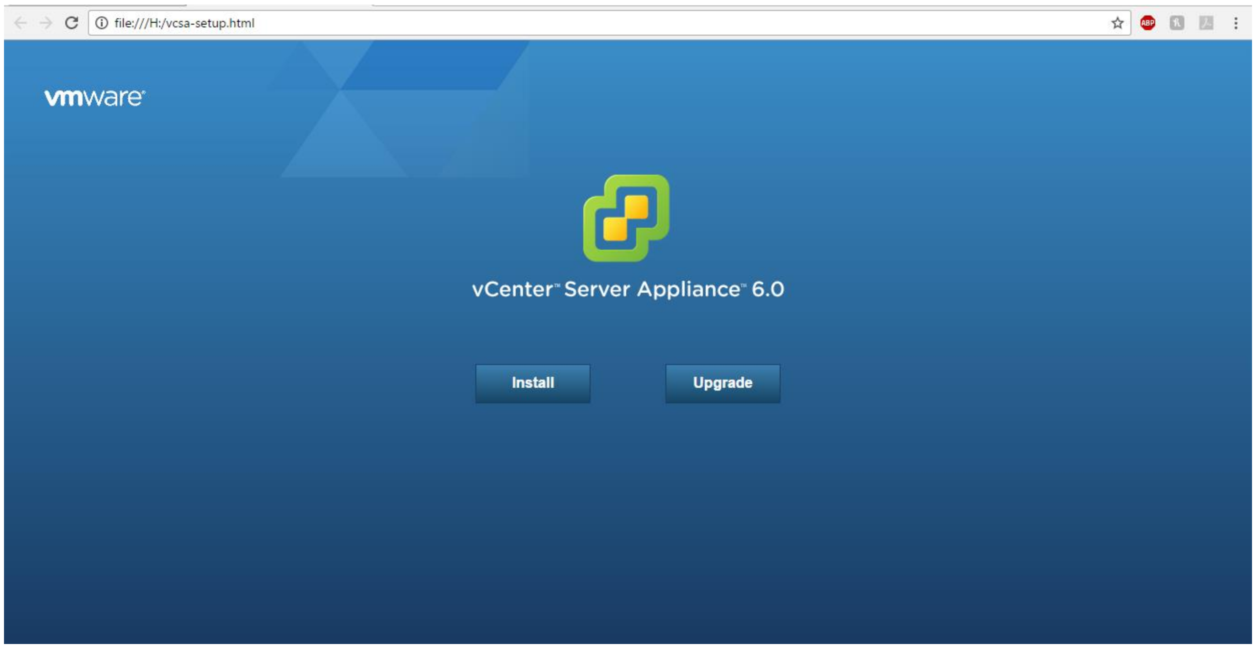
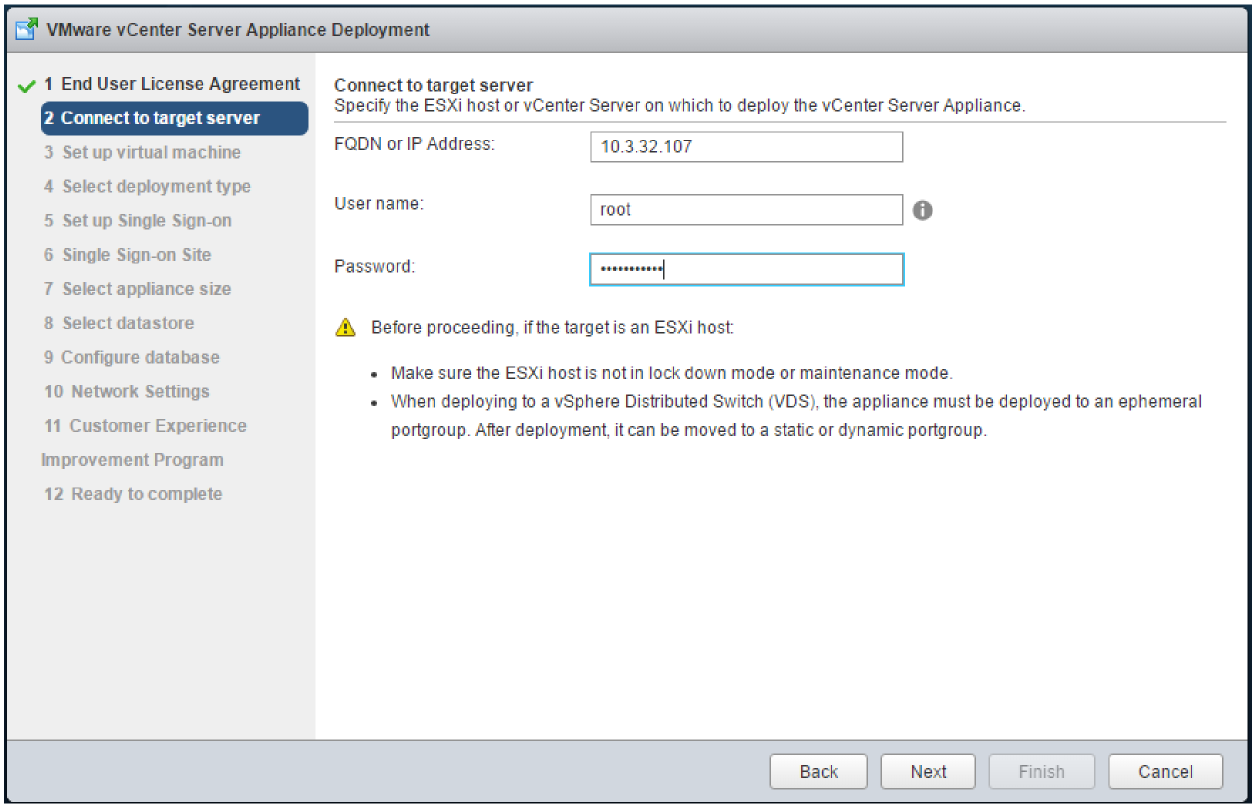


Fig 5.2 Vcenter Setup page

It's a fresh install and hence we go with the first option. It's very important to understand that Vcenter is not a standalone installation, rather it is installed as a virtual machine on the esxi host. Once installed, Vcenter provides a centralized console to access, manage and deploy on the esxi host. A single Vcenter instance located on one host can be connected and used to centrally manage any number of hosts in the same domain.



Connect to target server
Specify the ESXi host or vCenter Server on which to deploy the vCenter Server Appliance.

FQDN or IP Address: 10.3.32.107

User name: root

Password: ••••••••••

Before proceeding, if the target is an ESXi host:

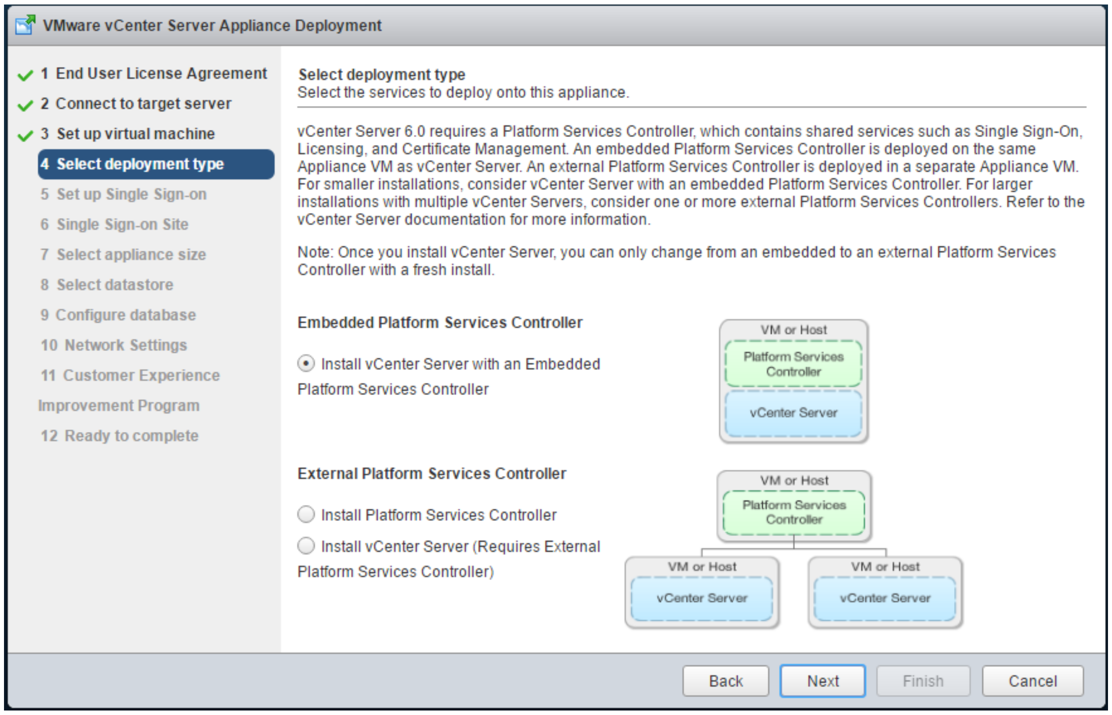
- Make sure the ESXi host is not in lock down mode or maintenance mode.
- When deploying to a vSphere Distributed Switch (VDS), the appliance must be deployed to an ephemeral portgroup. After deployment, it can be moved to a static or dynamic portgroup.

Fig 5.3 Target Server Details

After appropriately naming the appliance and specifying the passwords, we select the type of deployment. This gives us multiple options, for the purpose of this project we go with an embedded deployment type. The difference between embedded and platform service controller is out of the scope from this project point of view, but it is explained in quite detail in the VMware deployment guide.

A new SSO domain was setup with a new password to provide Single Sign On functionality.

The deployment type is Tiny i.e. up to 10 hosts and a 100 virtual machines. For the project a maximum of two servers were used and this deployment type suffices the requirements.


VMware vCenter Server Appliance Deployment

- ✓ 1 End User License Agreement
- ✓ 2 Connect to target server
- ✓ 3 Set up virtual machine
- 4 Select deployment type
- 5 Set up Single Sign-on
- 6 Single Sign-on Site
- 7 Select appliance size
- 8 Select datastore
- 9 Configure database
- 10 Network Settings
- 11 Customer Experience Improvement Program
- 12 Ready to complete

Select deployment type
Select the services to deploy onto this appliance.

vCenter Server 6.0 requires a Platform Services Controller, which contains shared services such as Single Sign-On, Licensing, and Certificate Management. An embedded Platform Services Controller is deployed on the same Appliance VM as vCenter Server. An external Platform Services Controller is deployed in a separate Appliance VM. For smaller installations, consider vCenter Server with an embedded Platform Services Controller. For larger installations with multiple vCenter Servers, consider one or more external Platform Services Controllers. Refer to the vCenter Server documentation for more information.

Note: Once you install vCenter Server, you can only change from an embedded to an external Platform Services Controller with a fresh install.

Embedded Platform Services Controller

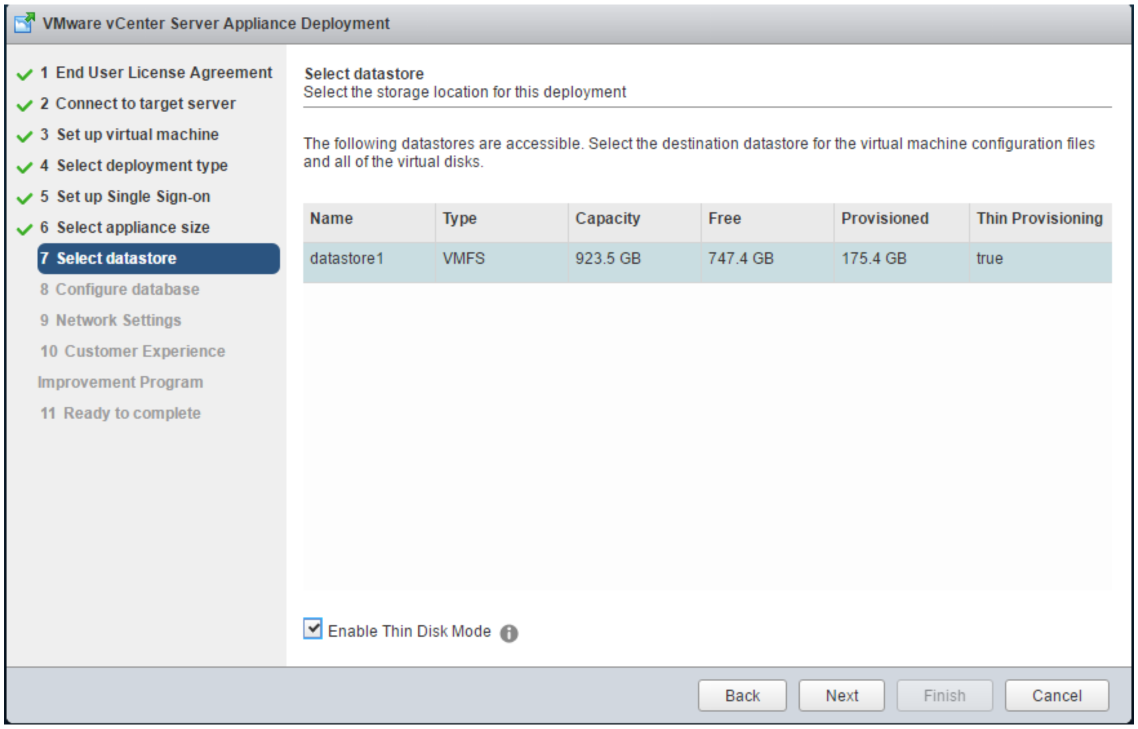
● Install vCenter Server with an Embedded Platform Services Controller

External Platform Services Controller

○ Install Platform Services Controller

○ Install vCenter Server (Requires External Platform Services Controller)

Fig 5.4 Embedded Vcenter install


VMware vCenter Server Appliance Deployment

- ✓ 1 End User License Agreement
- ✓ 2 Connect to target server
- ✓ 3 Set up virtual machine
- ✓ 4 Select deployment type
- ✓ 5 Set up Single Sign-on
- ✓ 6 Select appliance size
- 7 Select datastore
- 8 Configure database
- 9 Network Settings
- 10 Customer Experience Improvement Program
- 11 Ready to complete

Select datastore
Select the storage location for this deployment

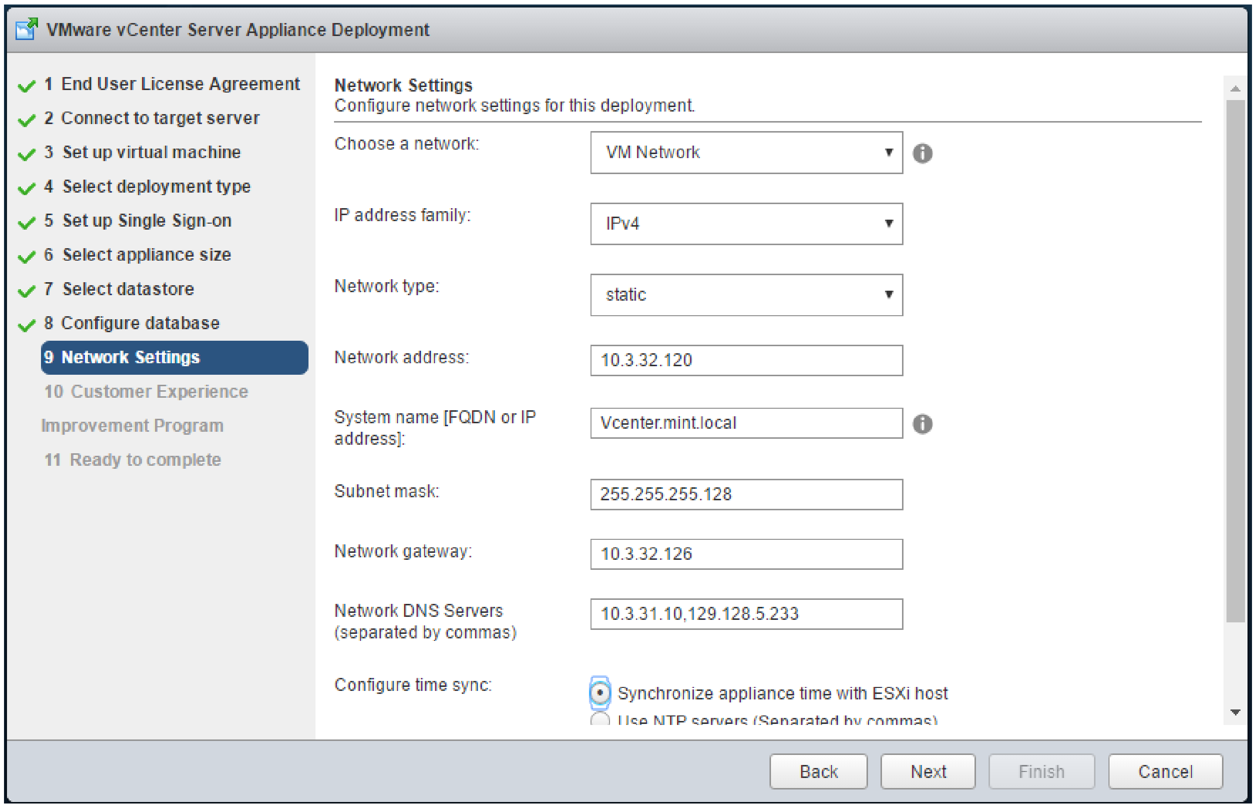
The following datastores are accessible. Select the destination datastore for the virtual machine configuration files and all of the virtual disks.

Name	Type	Capacity	Free	Provisioned	Thin Provisioning
datastore1	VMFS	923.5 GB	747.4 GB	175.4 GB	true

✓ Enable Thin Disk Mode

Fig 5.5 Selecting the Storage.

VMware recommends the use of SSD's for implementation, I used the VMFS Datastore available to me in a thin disk mode configuration. The thin disk mode doesn't pre-allocate the space required, instead it assigns space dynamically when needed. A Vcenter deployment also allows for external oracle database to be used, but for the sake of this project the embedded PostgreSQL was used.


VMware vCenter Server Appliance Deployment

✓ 1 End User License Agreement
✓ 2 Connect to target server
✓ 3 Set up virtual machine
✓ 4 Select deployment type
✓ 5 Set up Single Sign-on
✓ 6 Select appliance size
✓ 7 Select datastore
✓ 8 Configure database
9 Network Settings
10 Customer Experience Improvement Program
11 Ready to complete

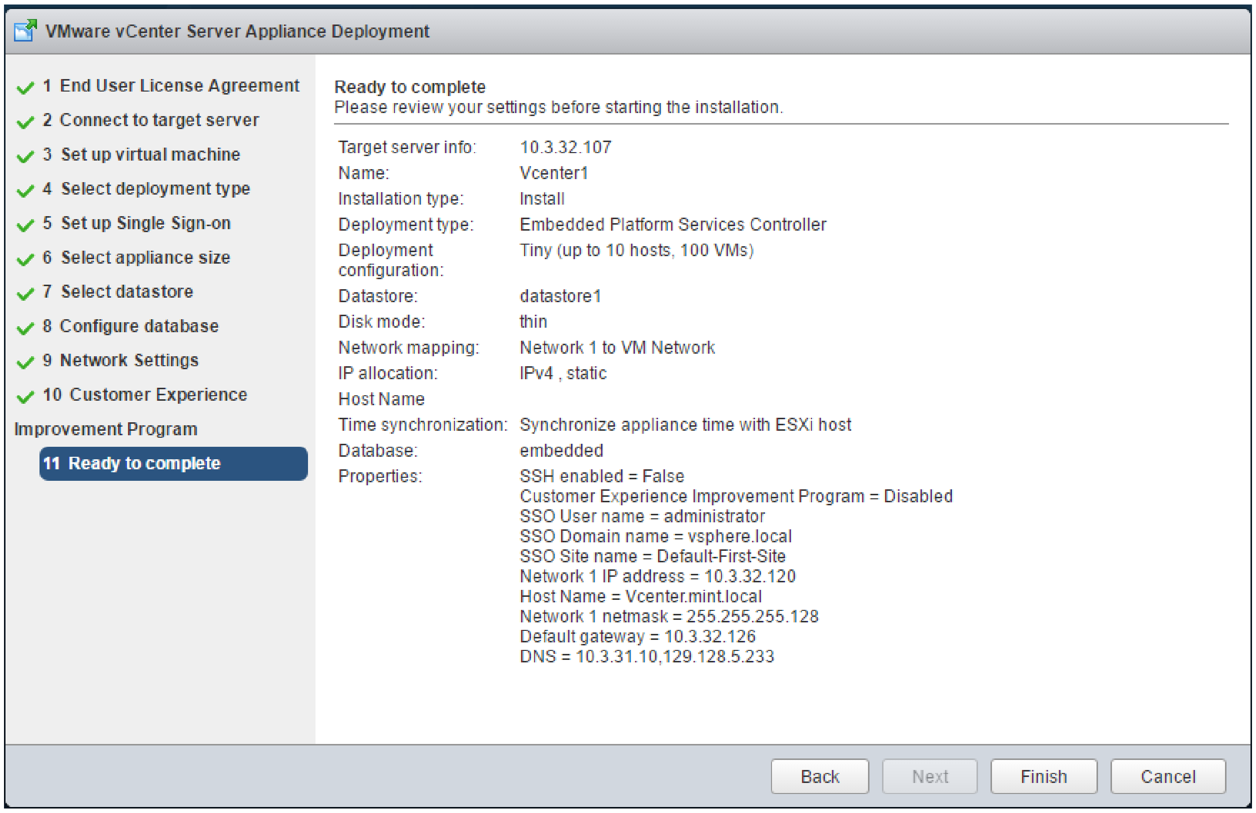
Network Settings
Configure network settings for this deployment.

Choose a network:	VM Network
IP address family:	IPv4
Network type:	static
Network address:	10.3.32.120
System name [FQDN or IP address]:	Vcenter.mint.local
Subnet mask:	255.255.255.128
Network gateway:	10.3.32.126
Network DNS Servers (separated by commas)	10.3.31.10,129.128.5.233
Configure time sync:	[✓ Synchronize appliance time with ESXi host] Use NTP servers (Separated by commas)

Fig 5.6 Vcenter Network settings

Probably the most important part of the whole Vcenter deployment. I had numerous failures when trying to deploy Vcenter and some of them are listed below

1. Vcenter deployment failed because the IP address assigned to the Vcenter didn't belong to the same subnet as the esxi host
2. The VM network selected didn't have an active network adapter that failed the deployment
3. The system name used didn't match the domain name. For the sake of this implementation it should be same as the esxi host. In my case the hostnames for esxi and Vcenter had to be esxi.mint.local & Vcenter.mint.local
4. Since no NTP server was used it was essential that both Vcenter and esxi are synchronized to the same time.


VMware vCenter Server Appliance Deployment

✓ 1 End User License Agreement
✓ 2 Connect to target server
✓ 3 Set up virtual machine
✓ 4 Select deployment type
✓ 5 Set up Single Sign-on
✓ 6 Select appliance size
✓ 7 Select datastore
✓ 8 Configure database
✓ 9 Network Settings
✓ 10 Customer Experience Improvement Program
11 Ready to complete

Ready to complete
Please review your settings before starting the installation.

Target server info:	10.3.32.107
Name:	Vcenter1
Installation type:	Install
Deployment type:	Embedded Platform Services Controller
Deployment configuration:	Tiny (up to 10 hosts, 100 VMs)
Datastore:	datastore1
Disk mode:	thin
Network mapping:	Network 1 to VM Network
IP allocation:	IPv4 , static
Host Name	
Time synchronization:	Synchronize appliance time with ESXi host
Database:	embedded
Properties:	SSH enabled = False Customer Experience Improvement Program = Disabled SSO User name = administrator SSO Domain name = vsphere.local SSO Site name = Default-First-Site Network 1 IP address = 10.3.32.120 Host Name = Vcenter.mint.local Network 1 netmask = 255.255.255.128 Default gateway = 10.3.32.126 DNS = 10.3.31.10,129.128.5.233

Fig 5.7 Final overview before deployment

Once all the details have been verified a Vcenter server can be successfully deployed. It takes about 30 -45 minutes to deploy and an addition 20 minutes to initialize the web page for Vcenter. Once done the web browser can be used to access Vcenter

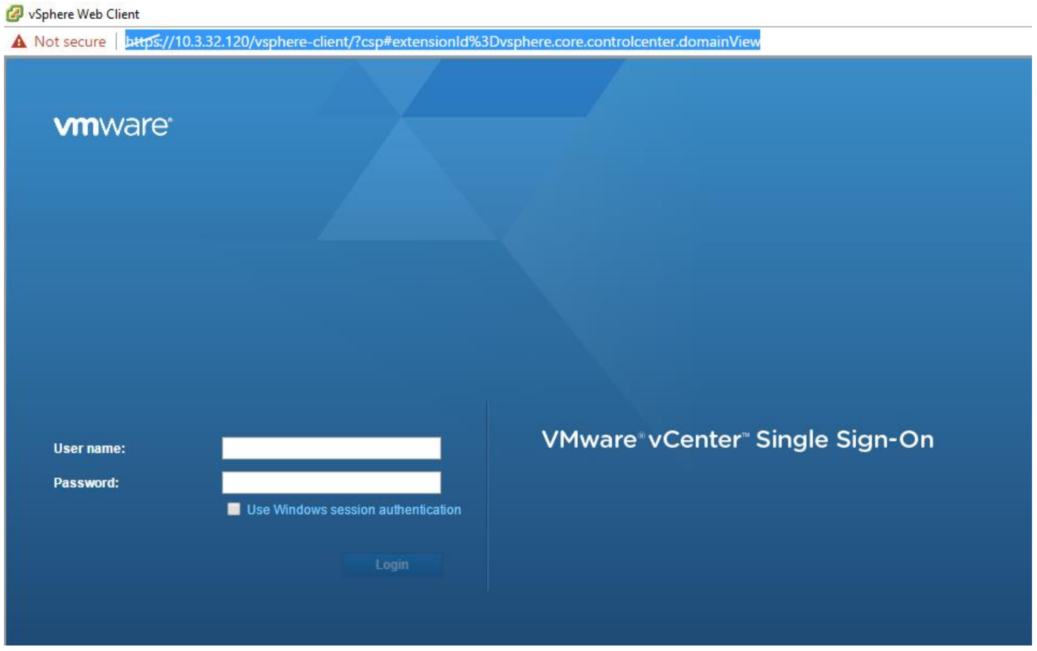

vSphere Web Client
Not secure
https://10.3.32.120/vsphere-client/?csp#extensionId%3Dvsphere.core.controlcenter.domainView
vmware
User name:
Password:
Use Windows session authentication
Login
VMware vCenter Single Sign-On

Fig 5.8 Vcenter Logon screen

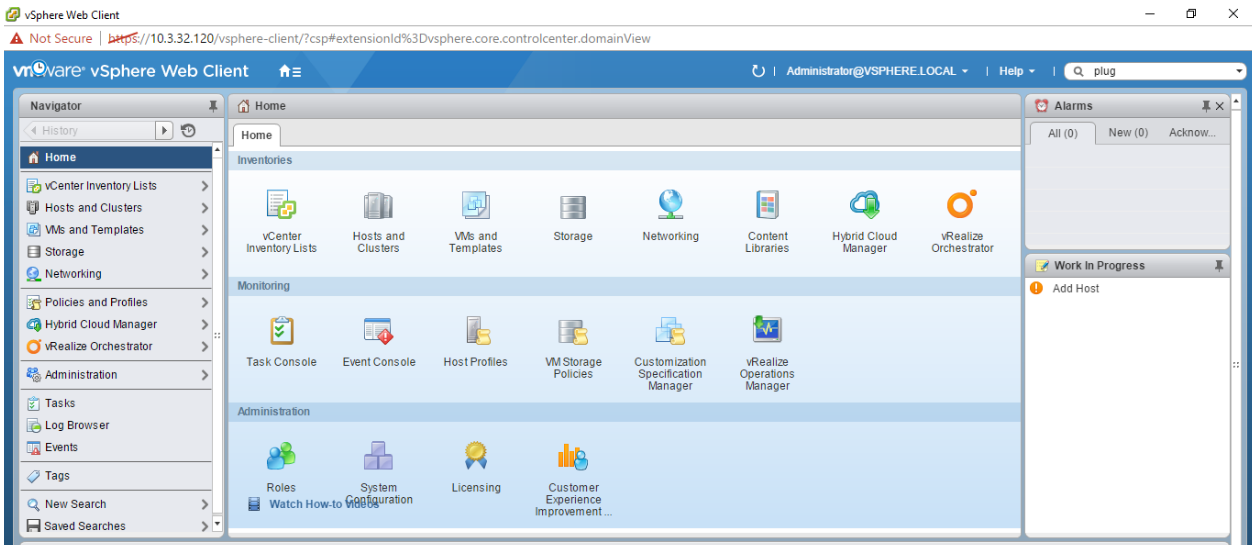
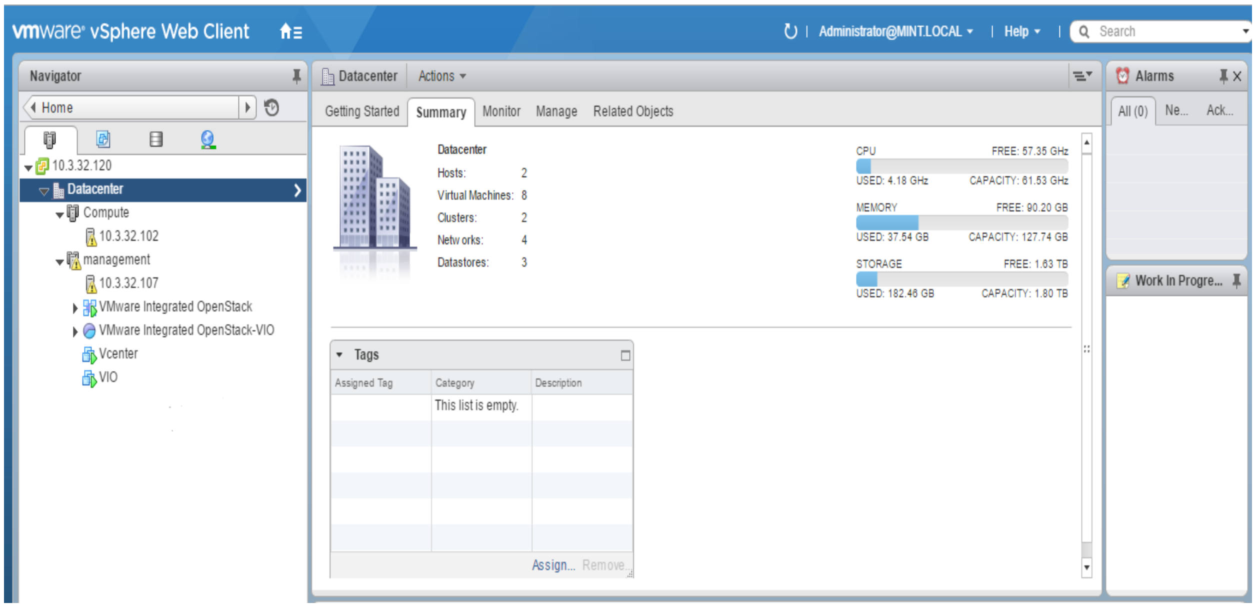


Fig 5.9 Vcenter home page

5.3 Creating Datacenters and adding hosts

Once the Vcenter is up and operational the next step is to create a datacenter and add appropriate hosts to it.



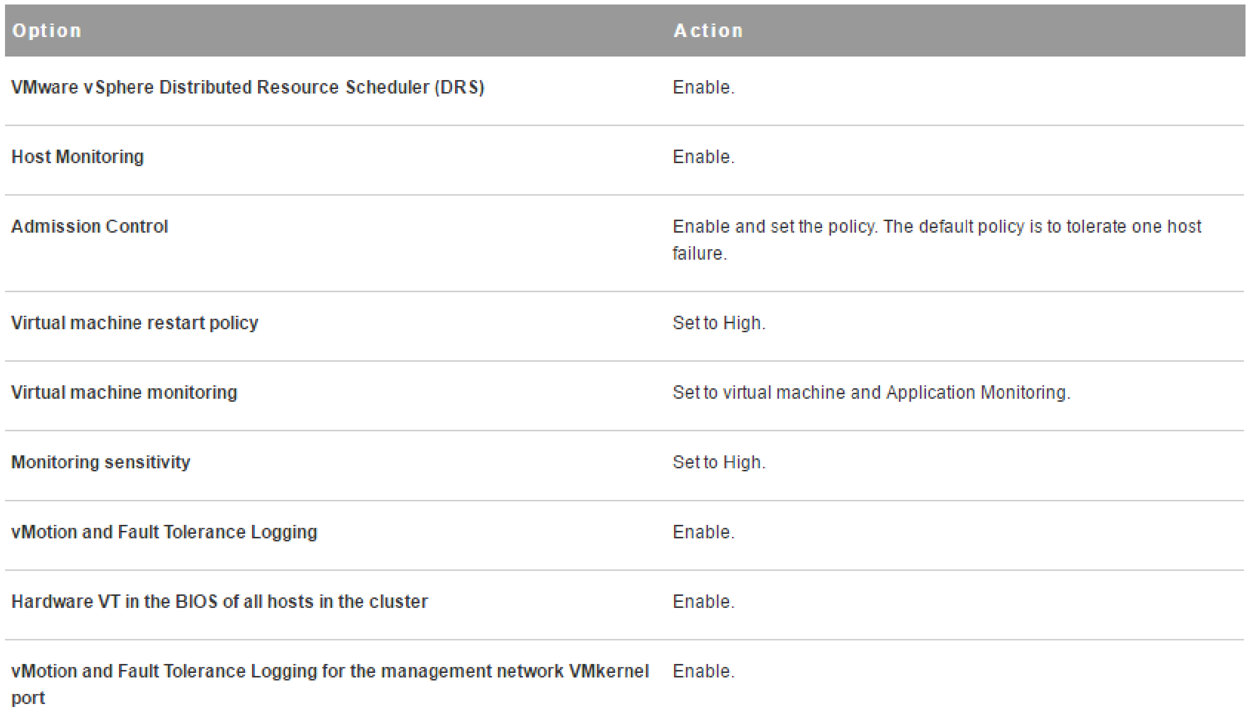
Datacenter
Hosts: 2
Virtual Machines: 8
Clusters: 2
Networks: 4
Datastores: 3

CPU FREE: 57.35 GHz
USED: 4.18 GHz CAPACITY: 61.53 GHz
MEMORY FREE: 90.20 GB
USED: 37.54 GB CAPACITY: 127.74 GB
STORAGE FREE: 1.63 TB
USED: 182.46 GB CAPACITY: 1.80 TB

Fig 5.10 Datacenter creation

The deployment is a cluster based deployment and for deploying Openstack two clusters namely Management and compute are needed. The process is quite simple and straightforward for creating the datacenter however the cluster should be configured to the specifications mentioned by

VMware in their deployment guides. Right clicking on the Vcenter IP in the hosts & cluster section gives the option for creating a new datacenter. The default values selected should be sufficient to successfully create one without any additional data needed. Once the datacenter is created the next step is to deploy the two clusters on it. Below are the specs required for configuring the clusters [6]



Option	Action
VMware vSphere Distributed Resource Scheduler (DRS)	Enable.
Host Monitoring	Enable.
Admission Control	Enable and set the policy. The default policy is to tolerate one host failure.
Virtual machine restart policy	Set to High.
Virtual machine monitoring	Set to virtual machine and Application Monitoring.
Monitoring sensitivity	Set to High.
vMotion and Fault Tolerance Logging	Enable.
Hardware VT in the BIOS of all hosts in the cluster	Enable.
vMotion and Fault Tolerance Logging for the management network VMkernel port	Enable.

Fig 5.11 Cluster specifications [6]

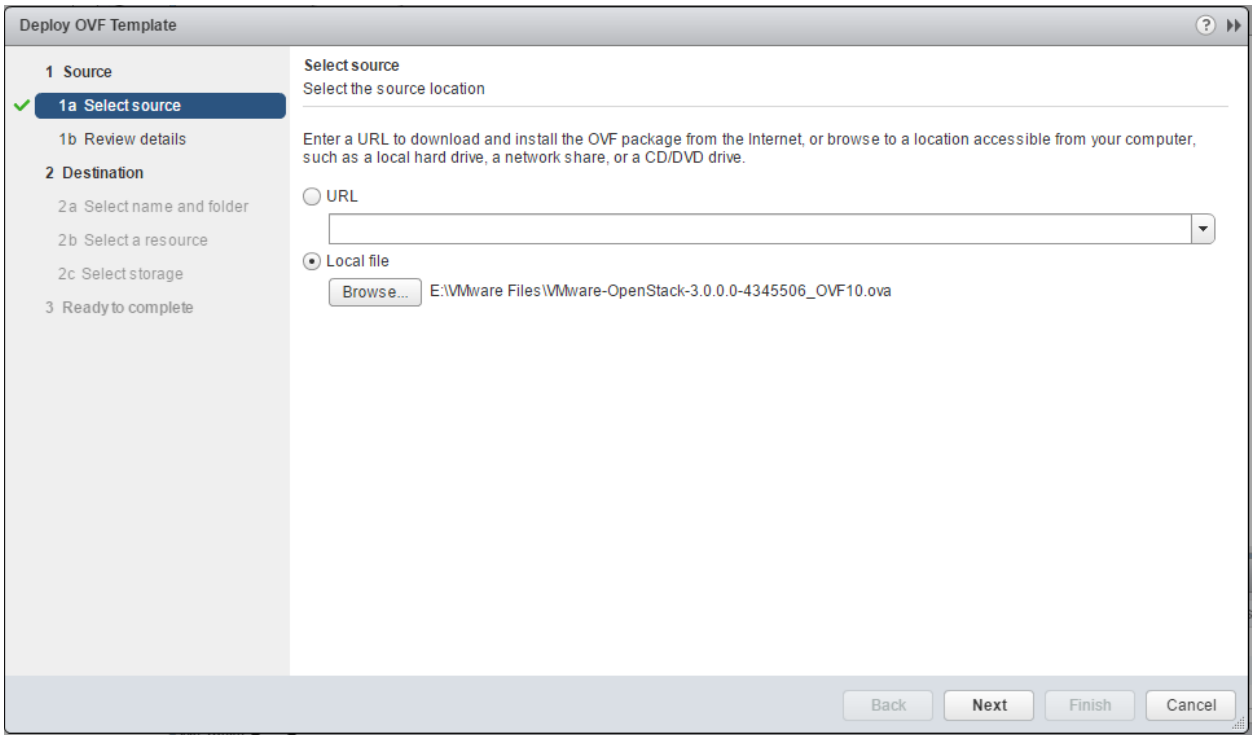
The minimum requirement is that at least one host be added to each of the clusters. In this scenario I have added the hosts 10.3.32.107 & 10.3.32.102 to management and compute cluster respectively as observed in Fig 5.10 Once the environment is ready Openstack manager can be installed on the datacenter Created

5.4 VMware Integrated Openstack OVA deployment

VMware recommends minimum configuration to deploy Openstack, however these specs are from a production point of view. However at the minimum a Disaster recovery server, Virtual SAN and an additional Datastore to store images and glance data is required all of which weren't at my disposal. Thus to make the deployment work there were certain changes that were made to the config files to ignore all of this properties. The first step is to acquire the OVA (Open Virtualization Appliance) file for Openstack. The latest version launched on 22nd Feb 2017 is 3.1

I have used the version 3.0 launched in September 2016. The OVA is available as a free download from the VMware website the link for which is Link.

There are many guides and tutorials available to help with the installation. Right clicking on the datacenter gives the option for deploying OVF. The rest of the steps are outlined below


Deploy OVF Template

1 Source
✓ 1a Select source
1b Review details
2 Destination
2a Select name and folder
2b Select a resource
2c Select storage
3 Ready to complete

Select source
Select the source location

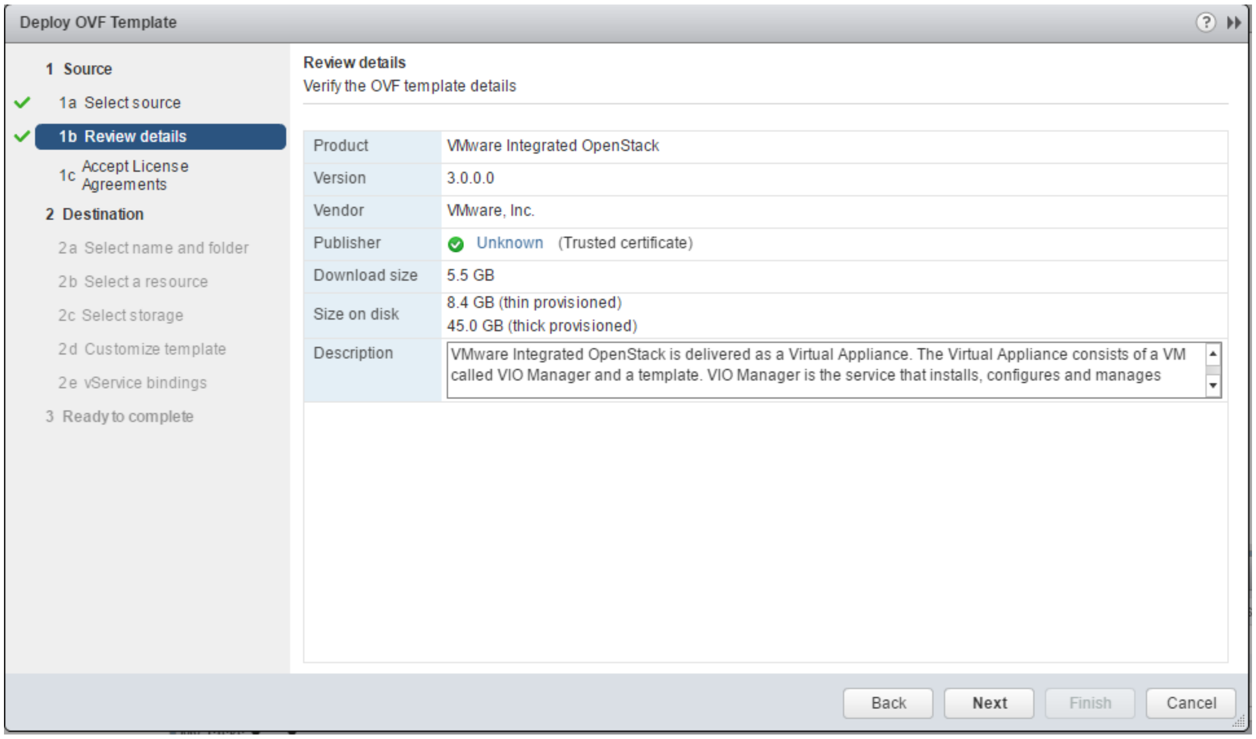
Enter a URL to download and install the OVF package from the Internet, or browse to a location accessible from your computer, such as a local hard drive, a network share, or a CD/DVD drive.

○ URL

● Local file

Browse... E:\VMware Files\VMware-OpenStack-3.0.0.0-4345506_OVF10.ova

Fig 5.12 OVA source from Local Directory


Deploy OVF Template

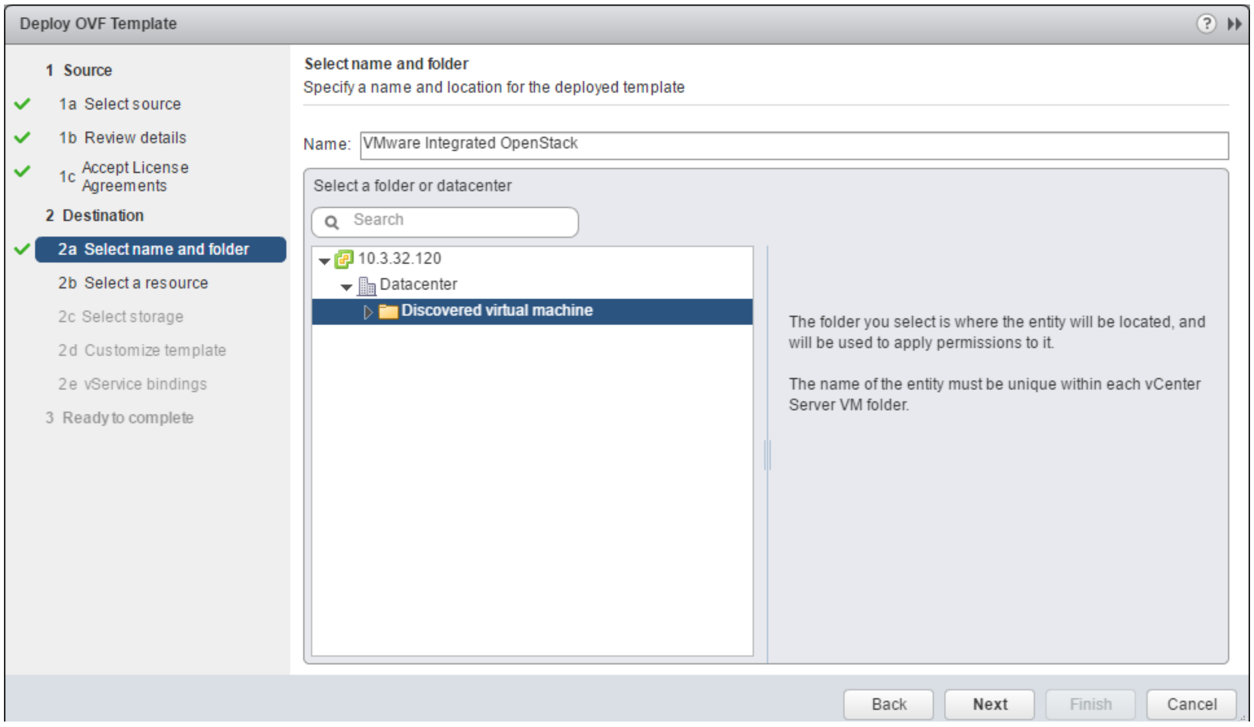
1 Source
✓ 1a Select source
✓ 1b Review details
1c Accept License Agreements
2 Destination
2a Select name and folder
2b Select a resource
2c Select storage
2d Customize template
2e vService bindings
3 Ready to complete

Review details
Verify the OVF template details

Product	VMware Integrated OpenStack
Version	3.0.0.0
Vendor	VMware, Inc.
Publisher	✓ Unknown (Trusted certificate)
Download size	5.5 GB
Size on disk	8.4 GB (thin provisioned) 45.0 GB (thick provisioned)
Description	VMware Integrated OpenStack is delivered as a Virtual Appliance. The Virtual Appliance consists of a VM called VIO Manager and a template. VIO Manager is the service that installs, configures and manages

Back | Next | Finish | Cancel

Fig 5.13 Verify the OVA details


Deploy OVF Template

1 Source
✓ 1a Select source
✓ 1b Review details
✓ 1c Accept License Agreements
2 Destination
✓ 2a Select name and folder
2b Select a resource
2c Select storage
2d Customize template
2e vService bindings
3 Ready to complete

Select name and folder
Specify a name and location for the deployed template

Name: VMware Integrated OpenStack

Select a folder or datacenter

Search

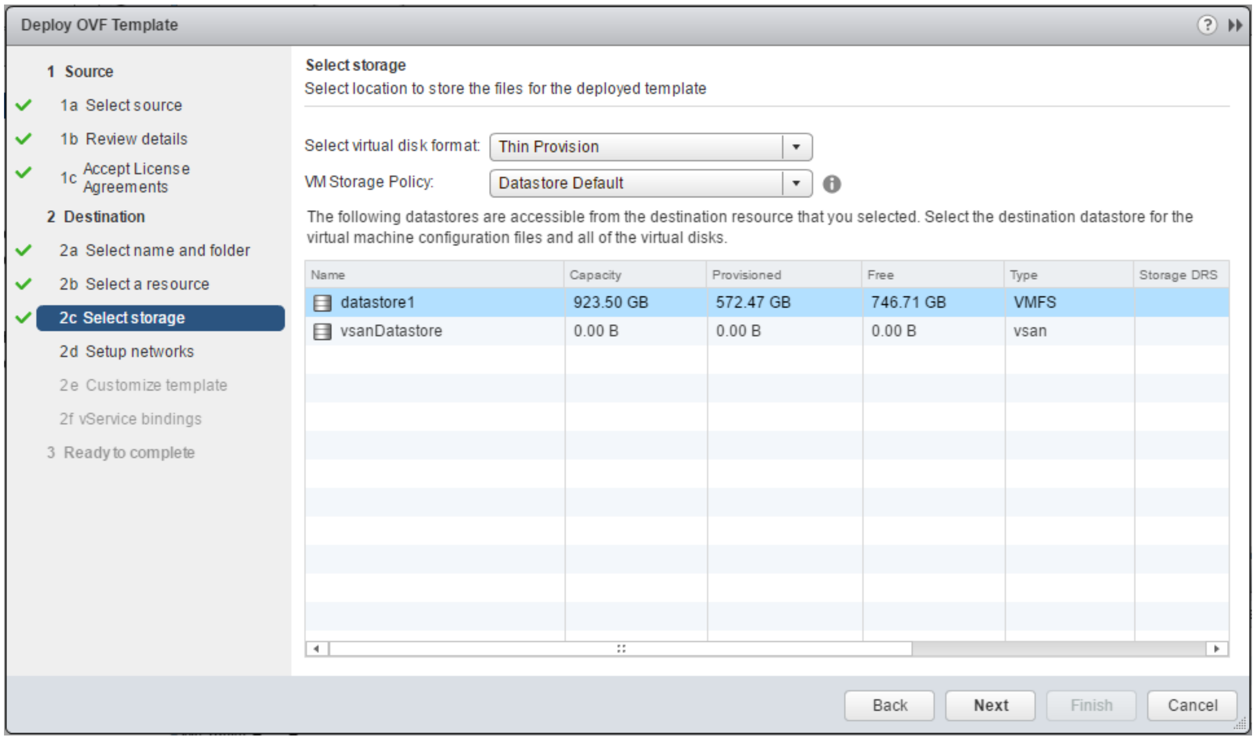
10.3.32.120
Datacenter
Discovered virtual machine

The folder you select is where the entity will be located, and will be used to apply permissions to it.

The name of the entity must be unique within each vCenter Server VM folder.

Back | Next | Finish | Cancel

Fig 5.14 Provide the folder for deployment



Select storage
Select location to store the files for the deployed template

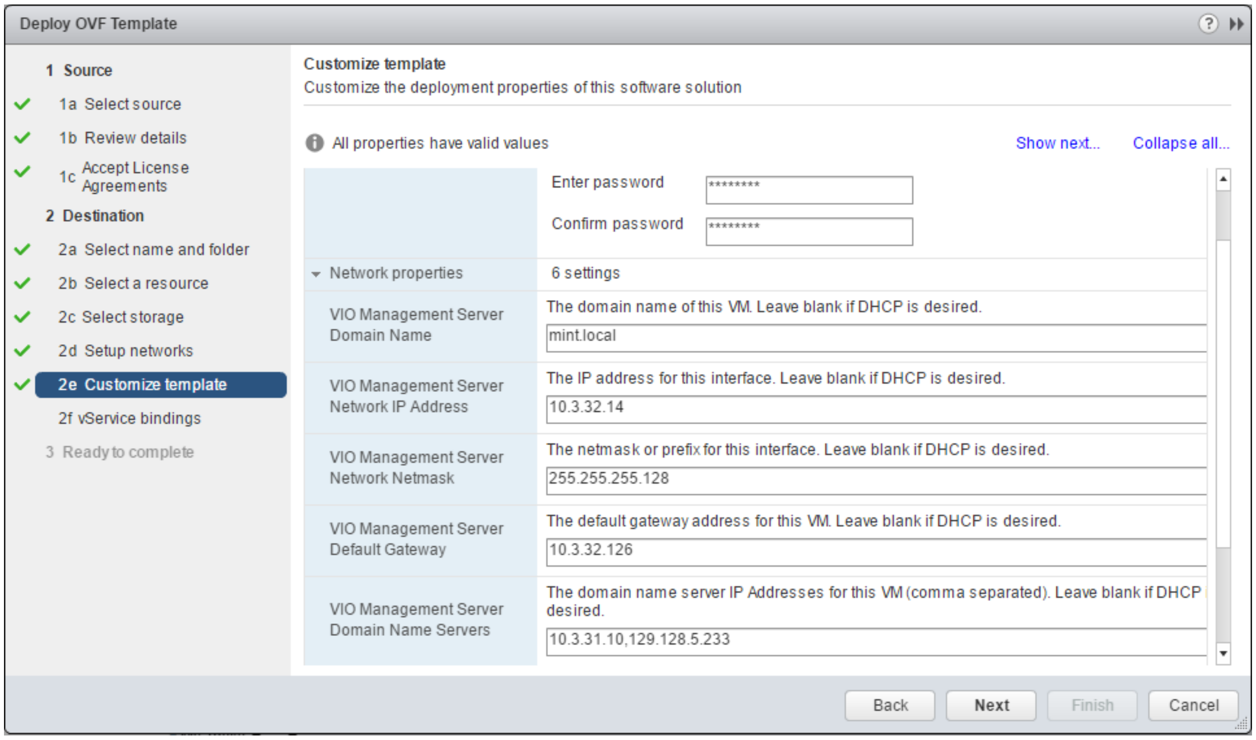
Select virtual disk format: Thin Provision

VM Storage Policy: Datastore Default

The following datastores are accessible from the destination resource that you selected. Select the destination datastore for the virtual machine configuration files and all of the virtual disks.

Name	Capacity	Provisioned	Free	Type	Storage DRS
datastore1	923.50 GB	572.47 GB	746.71 GB	VMFS	
vsanDatastore	0.00 B	0.00 B	0.00 B	vsan	

Fig 5.15 Select Storage and Thin provision it



Customize template
Customize the deployment properties of this software solution

All properties have valid values

Enter password ********
Confirm password ********

Network properties: 6 settings

VIO Management Server Domain Name — The domain name of this VM. Leave blank if DHCP is desired. mint.local

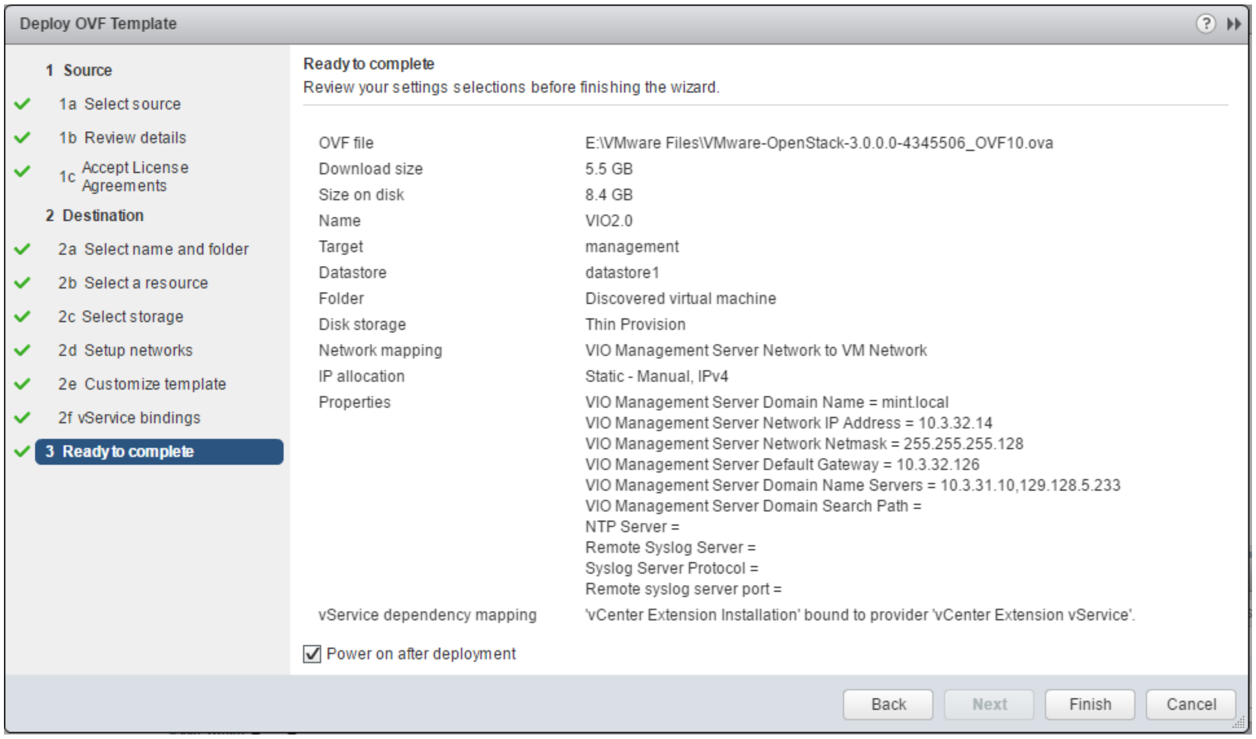
VIO Management Server Network IP Address — The IP address for this interface. Leave blank if DHCP is desired. 10.3.32.14

VIO Management Server Network Netmask — The netmask or prefix for this interface. Leave blank if DHCP is desired. 255.255.255.128

VIO Management Server Default Gateway — The default gateway address for this VM. Leave blank if DHCP is desired. 10.3.32.126

VIO Management Server Domain Name Servers — The domain name server IP Addresses for this VM (comma separated). Leave blank if DHCP desired. 10.3.31.10,129.128.5.233

Fig 5.16 Provide Networking Details



Ready to complete
Review your settings selections before finishing the wizard.

OVF file	E:\VMware Files\VMware-OpenStack-3.0.0.0-4345506_OVF10.ova
Download size	5.5 GB
Size on disk	8.4 GB
Name	VIO2.0
Target	management
Datastore	datastore1
Folder	Discovered virtual machine
Disk storage	Thin Provision
Network mapping	VIO Management Server Network to VM Network
IP allocation	Static - Manual, IPv4
Properties	VIO Management Server Domain Name = mint.local VIO Management Server Network IP Address = 10.3.32.14 VIO Management Server Network Netmask = 255.255.255.128 VIO Management Server Default Gateway = 10.3.32.126 VIO Management Server Domain Name Servers = 10.3.31.10,129.128.5.233 VIO Management Server Domain Search Path = NTP Server = Remote Syslog Server = Syslog Server Protocol = Remote syslog server port =
vService dependency mapping	'vCenter Extension Installation' bound to provider 'vCenter Extension vService'.

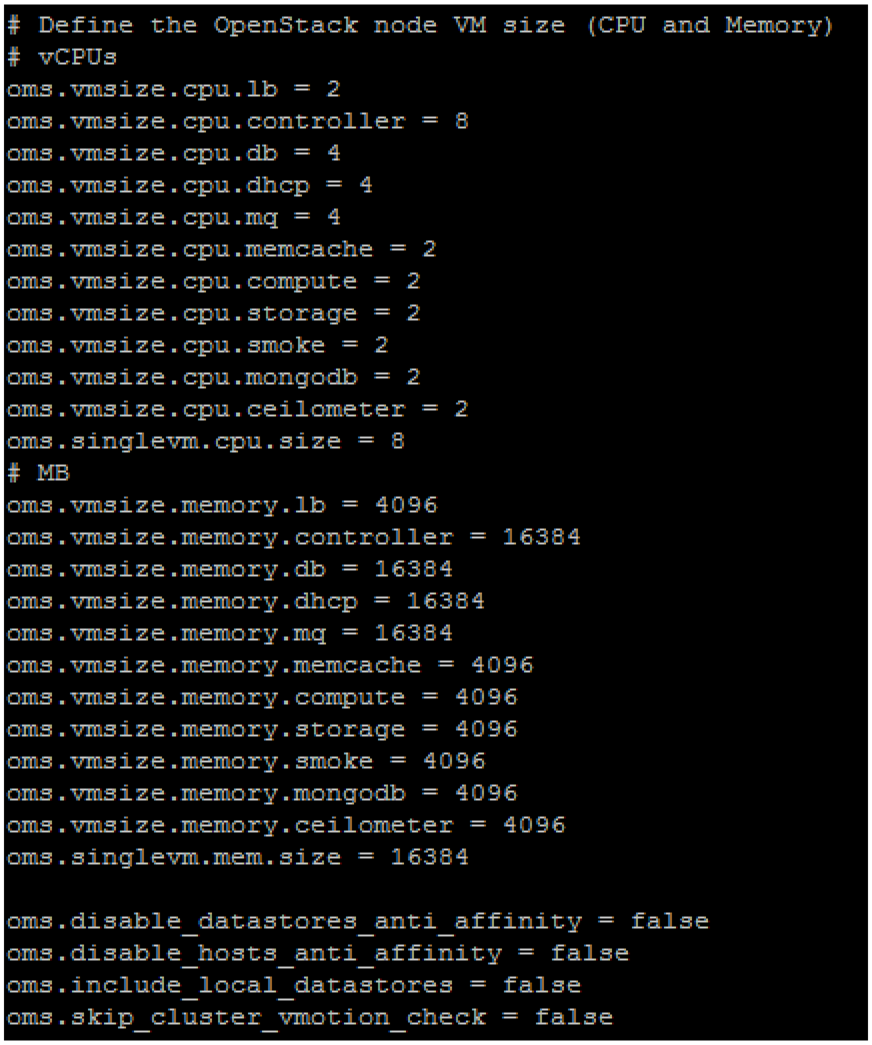
☑ Power on after deployment

Fig 5.17 Review before final deployment

The two most important things to verify before deploying are that the IP address lies in the same subnet as the Vcenter server, the network assigned should have active network adapters having connectivity without which the deployment will fail. The domain name is another important thing and to avoid conflicts due to the name servers being configured the domain name i.e. mint.local is maintained throughout for all virtual machines created. Once the deployment is completed, the server is accessible via SSH using putty. As mentioned earlier, resources like disaster recovery, additional Datastore, Load balancer or VSAN (Virtual Storage Area Network) are not available. This requires additional changes to the properties file [7]

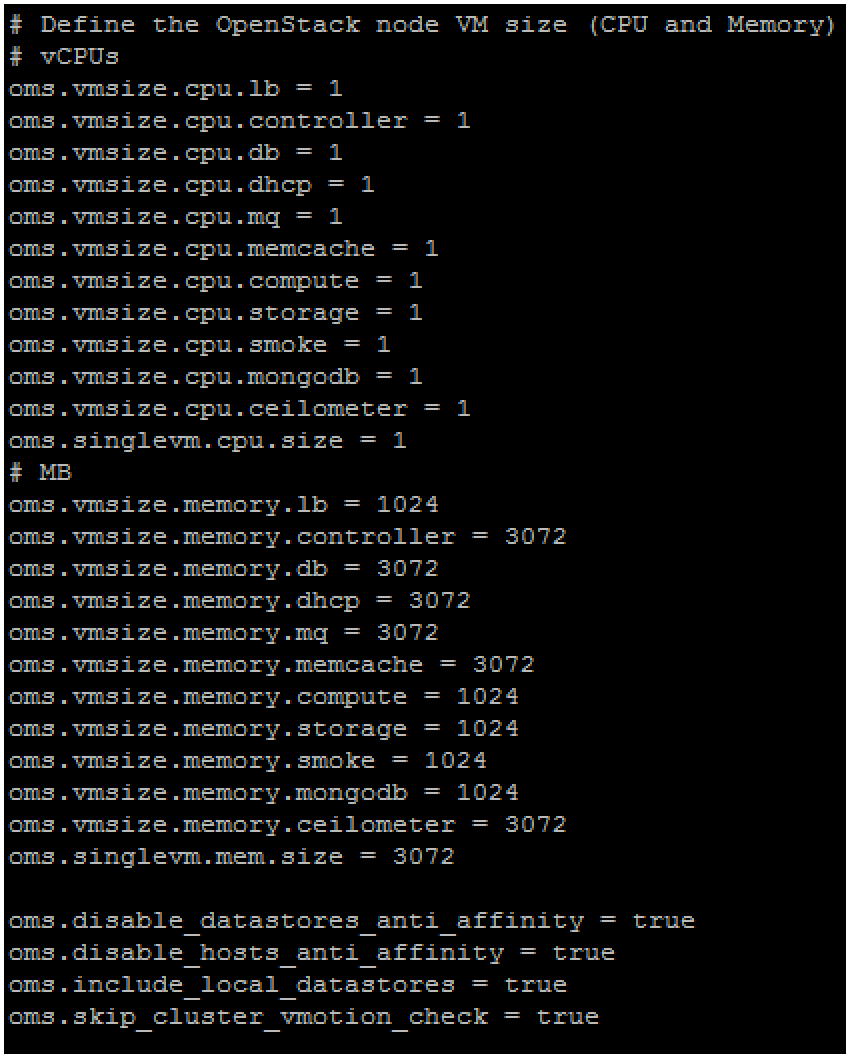
Once the newly deployed server is accessed the below steps were followed to configure it

1. SSH the server
2. enter the command "sudo -s" to gain root access
3. cd /opt/vmware/vio/etc
4. Edit the file omjs.properties to reflect the following values.



```
# Define the OpenStack node VM size (CPU and Memory)
# vCPUs
oms.vmsize.cpu.lb = 2
oms.vmsize.cpu.controller = 8
oms.vmsize.cpu.db = 4
oms.vmsize.cpu.dhcp = 4
oms.vmsize.cpu.mq = 4
oms.vmsize.cpu.memcache = 2
oms.vmsize.cpu.compute = 2
oms.vmsize.cpu.storage = 2
oms.vmsize.cpu.smoke = 2
oms.vmsize.cpu.mongodb = 2
oms.vmsize.cpu.ceilometer = 2
oms.singlevm.cpu.size = 8
# MB
oms.vmsize.memory.lb = 4096
oms.vmsize.memory.controller = 16384
oms.vmsize.memory.db = 16384
oms.vmsize.memory.dhcp = 16384
oms.vmsize.memory.mq = 16384
oms.vmsize.memory.memcache = 4096
oms.vmsize.memory.compute = 4096
oms.vmsize.memory.storage = 4096
oms.vmsize.memory.smoke = 4096
oms.vmsize.memory.mongodb = 4096
oms.vmsize.memory.ceilometer = 4096
oms.singlevm.mem.size = 16384

oms.disable_datastores_anti_affinity = false
oms.disable_hosts_anti_affinity = false
oms.include_local_datastores = false
oms.skip_cluster_vmotion_check = false
```

Fig 5.18 Original values of omjs.properties



```
# Define the OpenStack node VM size (CPU and Memory)
# vCPUs
oms.vmsize.cpu.lb = 1
oms.vmsize.cpu.controller = 1
oms.vmsize.cpu.db = 1
oms.vmsize.cpu.dhcp = 1
oms.vmsize.cpu.mq = 1
oms.vmsize.cpu.memcache = 1
oms.vmsize.cpu.compute = 1
oms.vmsize.cpu.storage = 1
oms.vmsize.cpu.smoke = 1
oms.vmsize.cpu.mongodb = 1
oms.vmsize.cpu.ceilometer = 1
oms.singlevm.cpu.size = 1
# MB
oms.vmsize.memory.lb = 1024
oms.vmsize.memory.controller = 3072
oms.vmsize.memory.db = 3072
oms.vmsize.memory.dhcp = 3072
oms.vmsize.memory.mq = 3072
oms.vmsize.memory.memcache = 3072
oms.vmsize.memory.compute = 1024
oms.vmsize.memory.storage = 1024
oms.vmsize.memory.smoke = 1024
oms.vmsize.memory.mongodb = 1024
oms.vmsize.memory.ceilometer = 3072
oms.singlevm.mem.size = 3072

oms.disable_datastores_anti_affinity = true
oms.disable_hosts_anti_affinity = true
oms.include_local_datastores = true
oms.skip_cluster_vmotion_check = true
```

Fig 5.19 New values of omjs.properties

5. Change the following lines in the same file

   oms.use_linked_clone = false to oms.use_linked_clone = true

   oms.datadisk_size = 60 to oms.datadisk_size = 20

6. Go to the path using command “cd /var/lib/vio/ansible/roles/neutron-server/templates/etc/neutron/plugins/vmware”
7. Add the line “backup_edge_pool = service:large:1:3,service:compact:1:3,vdr:large:1:3” to the nsxv.ini file

The above configuration is very essential. With the amount of resources available to me Compact mode is the only kind of deployment that can be done and the above changes make sure that the deployment can take necessary values from the properties file and not fail during deployment. The backup_edge_pool setting ensures that the VIO instance can run in Compact mode.

Once the server has been restarted we should be able to see the VIO plugin on the Vcenter home page, however this is where the real issues lie. Even though the deployment was successful and we can verify that by accessing the VIO management server this does not ensure that we can deploy a VIO instance. Vmware uses the VIO plugin to deploy VIO instance and there are many reasons why the plugin might not appear. This is one of the major issues I faced while implementing the project and it took a whole lot of reading forums and knowledge base articles to resolve it. I had to take help from the Vmware forums and start my own discussion thread to get my logs analysed and find a resolution. Below is the link to the discussion I opened on Vmware forum

https://code.vmware.com/forums/5494#552920|3655877

There were many errors that can be seen in the oms.log file that can tell the status of the deployment. Some of them do not relate to the environment and can be ignored, however there are a few which are absolutely necessary to be fixed.

```
[2017-01-27T16:23:05.226+0000] INFO localhost-startStop-1| org.hibernate.engine.jdbc.internal.LobCreatorBuilder: HHH000424: Disabling contextual LOB creation as createClob() method threw error : java.lang.reflect.InvocationTargetException

[2017-01-27T16:23:06.213+0000] INFO localhost-startStop-1| org.hibernate.tool.hbm2ddl.TableMetadata: HHH000037: Columns: [error_message, vc_rp_id, rack, vc_datastores, memory, volumes, cpu_number, action_failed, version, moid, node_group_id, action, guest_host_name, id, vm_name, host_name, status, power_status_changed]

[2017-01-27T16:23:17.511+0000] INFO VcEventListener| com.vmware.aurora.vc.vcevent.VcEventListener: vim.event.VmMessageErrorEvent

[2017-01-27T16:23:20.738+0000] INFO localhost-startStop-1| com.vmware.aurora.vc.vcevent.VcEventHandlers: Installed external event handler: VmMessageError -> com.vmware.aurora.vc.vcevent.VcEventRouter$1@72a6be88

[2017-01-27T16:23:21.780+0000] INFO localhost-startStop-1| com.vmware.aurora.vc.vcevent.VcEventHandlers: Installed external event handler: VmMessageError -> com.vmware.openstack.service.event.VmEventManager$1@6eb565b

[2017-01-27T16:23:21.791+0000] INFO localhost-startStop-1| com.vmware.aurora.vc.vcevent.VcEventHandlers: Installed internal event handler: VmMessageError -> com.vmware.openstack.service.event.VmEventManager$2@79164b0c

[2017-01-27T16:23:35.038+0000] ERROR localhost-startStop-1| com.vmware.openstack.security.sso.utils.SecurityUtils: Authentication error :null

[2017-01-27T16:23:35.039+0000] ERROR localhost-startStop-1| com.vmware.openstack.manager.PluginRegisterManager: SSO server is misconfigured
```

The ones highlighted in yellow are a major concern and have to be resolved in order for the plugin to be displayed properly. Vmware has posted a solution for this particular issue, however it only fixes a part of the issue and not the whole thing. Vmware knowledge based article was referred for the below solution [8]

1. Create a snapshot of VMware Integrated OpenStack vApp.

2. Shutdown and power on the VMware Integrated OpenStack vApp.

3. Connect to the management-server through the SSH

4. Stop oms services by running commands:

   service oms stop

   service osvmw stop

5. Take a backup of the /opt/vmware/vio/etc file.

6. cp -r /opt/vmware/vio/etc /root/etc_backup

7. Take a snapshot of the management-server.

8. Remove files that block certificate regeneration by running command:

   rm /opt/vmware/vio/etc/oms.lock /opt/vmware/vio/etc/guard.key

9. Clean up the old certificate values by running this command:

   sed -i '/cms.*/d' /opt/vmware/vio/etc/vio_system.properties

10. Run script to generate new certificate:

   /opt/vmware/vio/sysctl/scripts/generate-certs.sh`cat /opt/vmware/vio/etc/keystore.properties | grep keystorePass | awk -F"=" '{print $2}' | tr -d "\n"` oms oms_server /opt/vmware/vio/etc

11. Modify the /opt/vmware/vio/etc/omjs.properties file and set the oms.extension.registered to false to allow oms to register with vCenter Server with new certificate.

12. Restore original guard.key and password:

   cp /root/etc_backup/guard.key /opt/vmware/vio/etc/

13. Open the /opt/vmware/vio/etc/vio_system.properties file and locate the entry starting with:

   "cms.guard_keystore_pswd"

14. Replace it with same entry from /root/etc_backup/vio_system.properties file.

15. Stop and restart the VIO vApp.

16. Log out from the vSphere Web Client.

17. Clear the browser cache and log in to the vSphere Web Clinet.

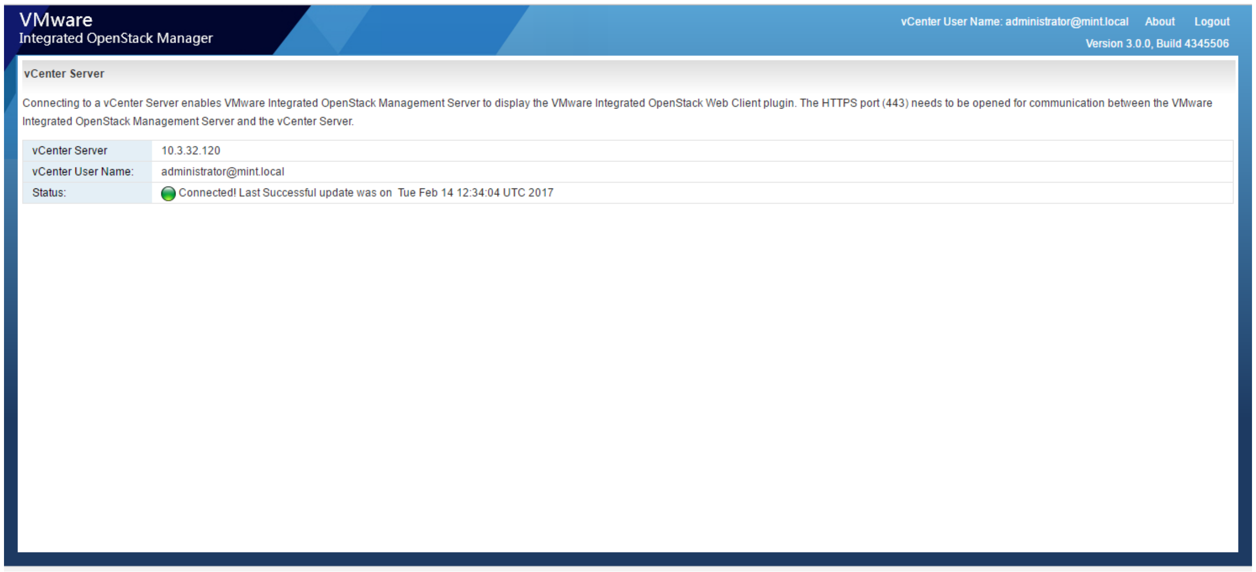
18. Go to the VMware Integrated Openstack plugin and reconnect the OMS.

19. Verify if you can run viocli commands and delete the snapshot.

However this does not fully solve the issue. A proper sequence must be followed to forcefully register the VIO plugin in Vcenter, The steps are as below.

1. Reboot the esxi host on which the Vcenter is running.
2. Once the Vcenter is started verify that the VIO management server is on.
3. SSH access the management server
4. Stop the oms service, then stop the osvmw service
5. Start the oms service followed by osvmw service
6. Login after an hour and the plugin will be present.

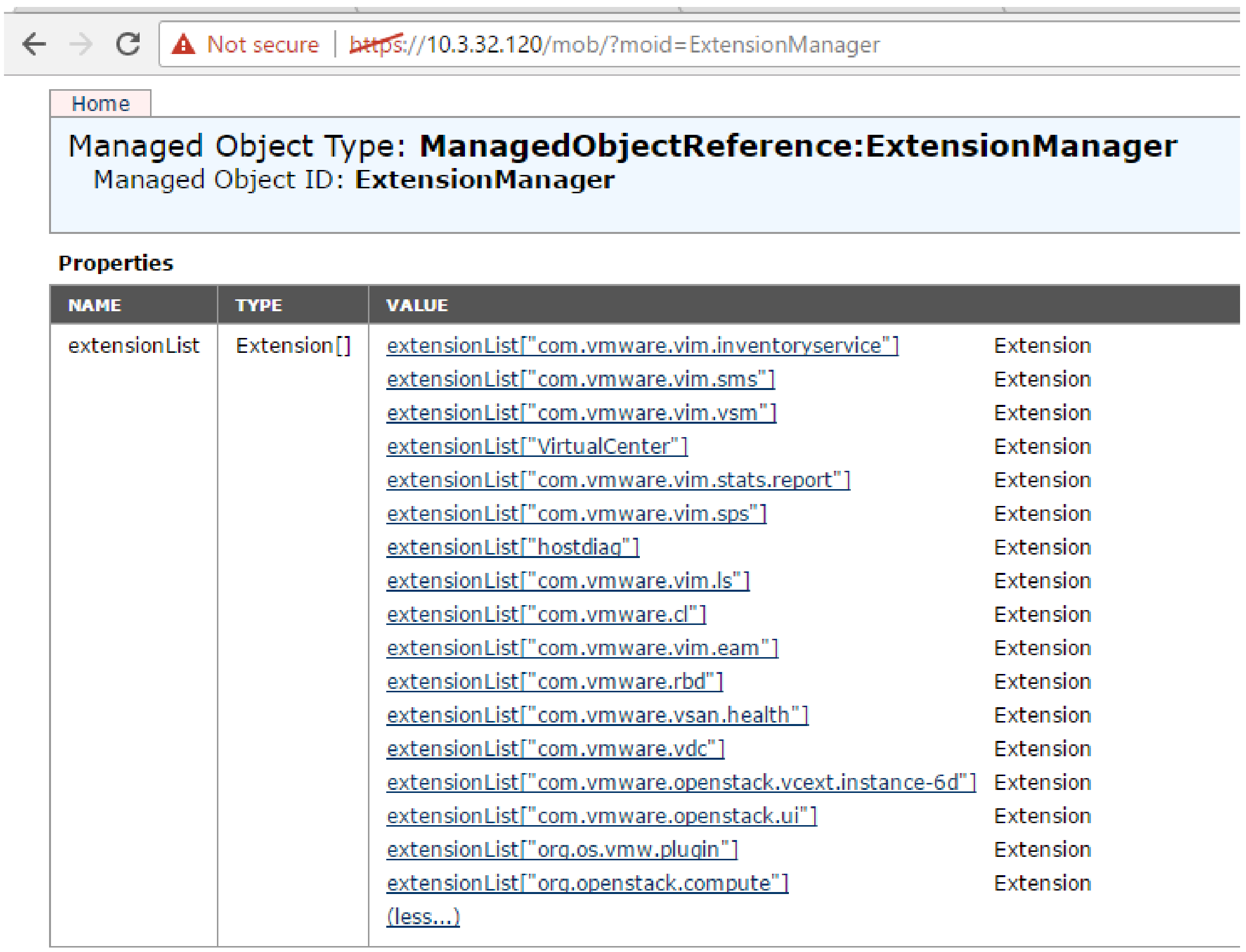
The below snapshots help in verifying that the Openstack plugin has been registered and connected to Vcenter


VMware
Integrated OpenStack Manager
vCenter User Name: administrator@mint.local
About
Logout
Version 3.0.0, Build 4345506
vCenter Server

Connecting to a vCenter Server enables VMware Integrated OpenStack Management Server to display the VMware Integrated OpenStack Web Client plugin. The HTTPS port (443) needs to be opened for communication between the VMware Integrated OpenStack Management Server and the vCenter Server.

vCenter Server	10.3.32.120
vCenter User Name:	administrator@mint.local
Status:	● Connected! Last Successful update was on Tue Feb 14 12:34:04 UTC 2017

Fig 5.20 Status of the Plugin



Not secure | ~~https~~://10.3.32.120/mob/?moid=ExtensionManager

Home

Managed Object Type: **ManagedObjectReference:ExtensionManager**
Managed Object ID: **ExtensionManager**

Properties

NAME	TYPE	VALUE	
extensionList	Extension[]	extensionList["com.vmware.vim.inventoryservice"]	Extension
		extensionList["com.vmware.vim.sms"]	Extension
		extensionList["com.vmware.vim.vsm"]	Extension
		extensionList["VirtualCenter"]	Extension
		extensionList["com.vmware.vim.stats.report"]	Extension
		extensionList["com.vmware.vim.sps"]	Extension
		extensionList["hostdiag"]	Extension
		extensionList["com.vmware.vim.ls"]	Extension
		extensionList["com.vmware.cl"]	Extension
		extensionList["com.vmware.vim.eam"]	Extension
		extensionList["com.vmware.rbd"]	Extension
		extensionList["com.vmware.vsan.health"]	Extension
		extensionList["com.vmware.vdc"]	Extension
		extensionList["com.vmware.openstack.vcext.instance-6d"]	Extension
		extensionList["com.vmware.openstack.ui"]	Extension
		extensionList["org.os.vmw.plugin"]	Extension
		extensionList["org.openstack.compute"]	Extension
		(less...)	

Fig 5.21 Extension manager for Vcenter

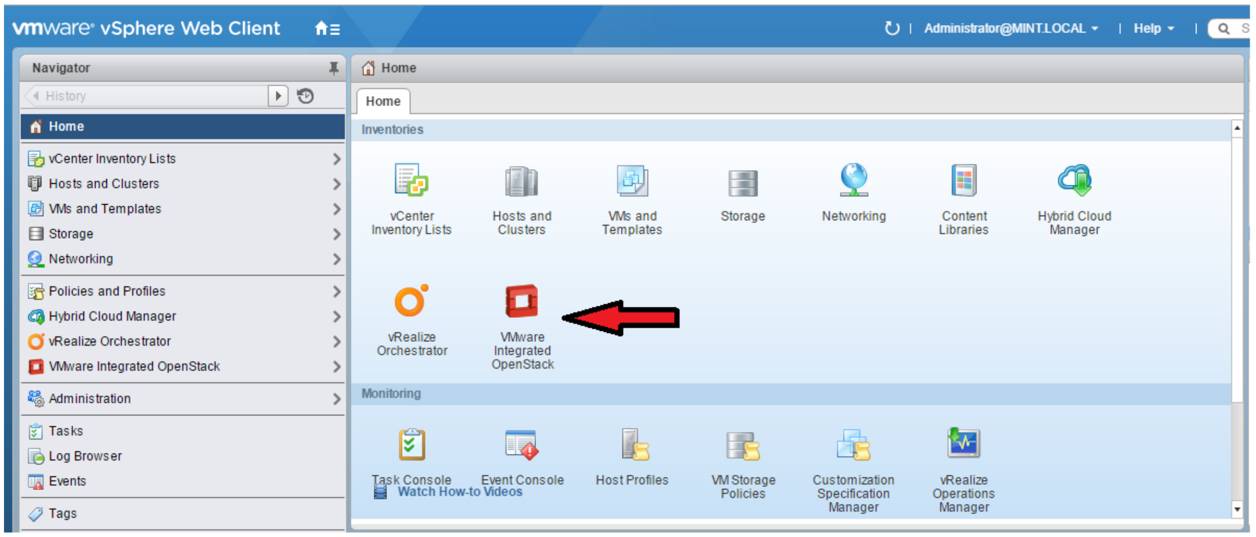
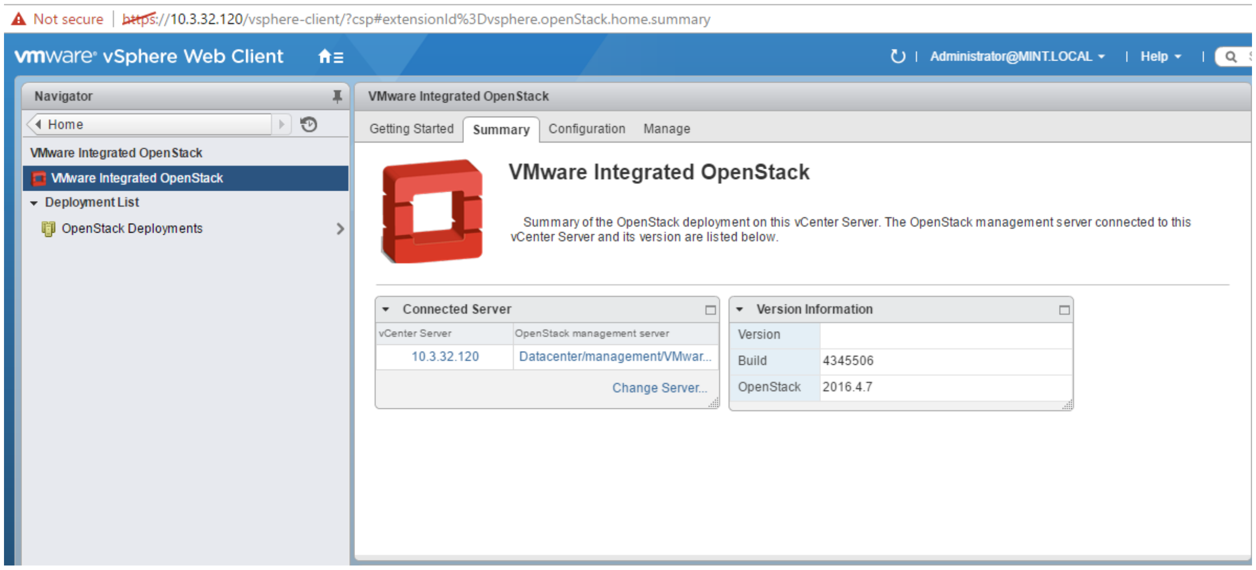


Fig 5.22 VIO plugin on Vcenter Homepage.

The next step is to verify connectivity of the VIO plugin with the Vcenter which can be done accessing the summary tab of the VIO plugin on the home page


Not secure | https://10.3.32.120/vsphere-client/?csp#extensionId%3Dvsphere.openStack.home.summary
vmware vSphere Web Client
Administrator@MINT.LOCAL
Help
Navigator
Home
VMware Integrated OpenStack
VMware Integrated OpenStack
Deployment List
OpenStack Deployments
Getting Started
Summary
Configuration
Manage

VMware Integrated OpenStack

Summary of the OpenStack deployment on this vCenter Server. The OpenStack management server connected to this vCenter Server and its version are listed below.

Connected Server

vCenter Server	OpenStack management server
10.3.32.120	Datacenter/management/VMwar...

Change Server...

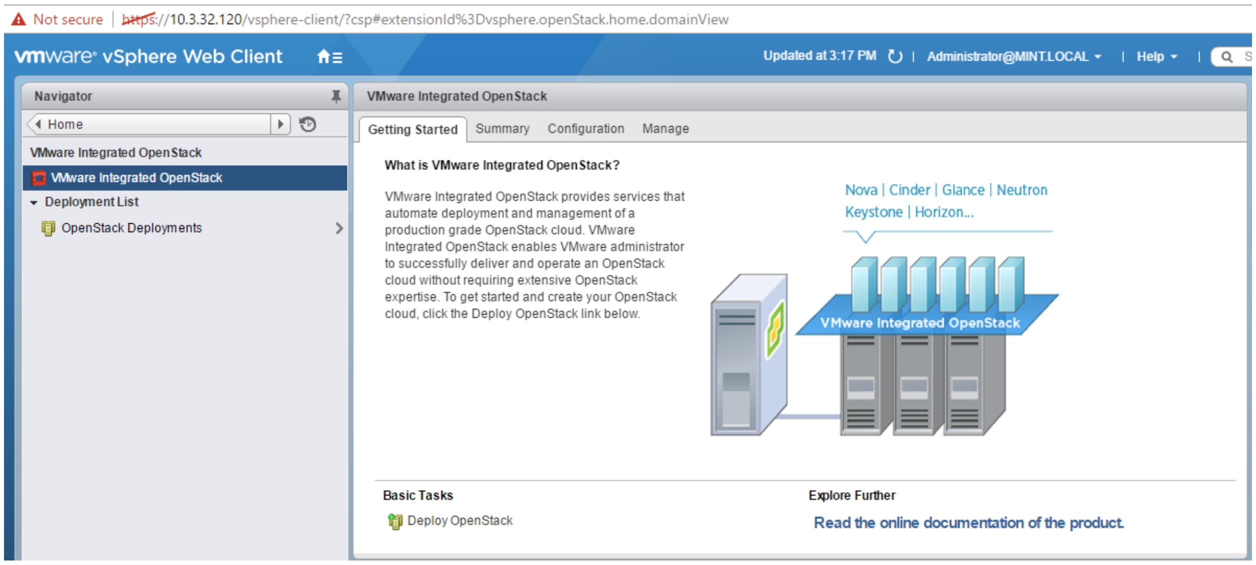
Version Information

Version	
Build	4345506
OpenStack	2016.4.7

Fig 5.23 Connection verification of VIO

5.5 Vmware integrated Openstack Instance deployment

Once the Openstack plugin has established connection with the management server VIO instance can be deployed. The snaps below list the steps and the config used to deploy the VIO instance


Not secure | https://10.3.32.120/vsphere-client/?csp#extensionId%3Dvsphere.openStack.home.domainView
vmware vSphere Web Client
Updated at 3:17 PM
Administrator@MINT.LOCAL
Help
Navigator
Home
VMware Integrated OpenStack
VMware Integrated OpenStack
Deployment List
OpenStack Deployments
Getting Started
Summary
Configuration
Manage
Nova | Cinder | Glance | Neutron
Keystone | Horizon...
VMware Integrated OpenStack

What is VMware Integrated OpenStack?

VMware Integrated OpenStack provides services that automate deployment and management of a production grade OpenStack cloud. VMware Integrated OpenStack enables VMware administrator to successfully deliver and operate an OpenStack cloud without requiring extensive OpenStack expertise. To get started and create your OpenStack cloud, click the Deploy OpenStack link below.

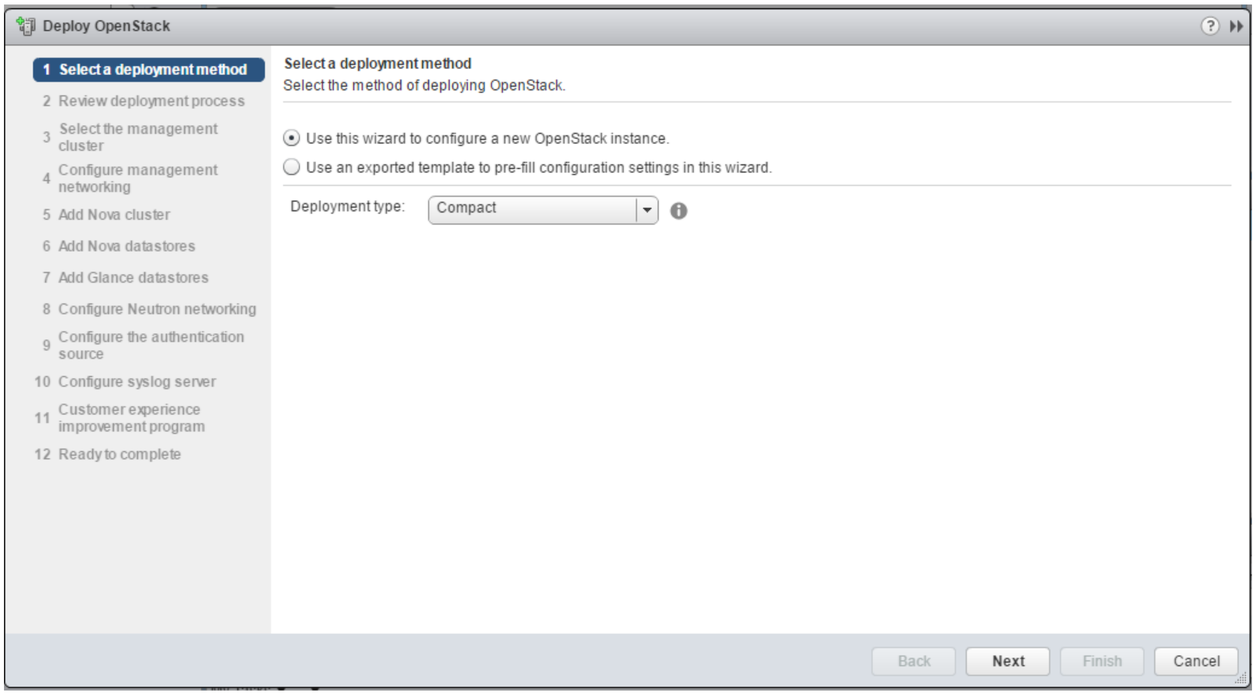
Basic Tasks

Deploy OpenStack

Explore Further

Read the online documentation of the product.

Fig 5.24 Deploy Openstack homepage


Deploy OpenStack
1 Select a deployment method
2 Review deployment process
3 Select the management cluster
4 Configure management networking
5 Add Nova cluster
6 Add Nova datastores
7 Add Glance datastores
8 Configure Neutron networking
9 Configure the authentication source
10 Configure syslog server
11 Customer experience improvement program
12 Ready to complete
Back
Next
Finish
Cancel

Select a deployment method

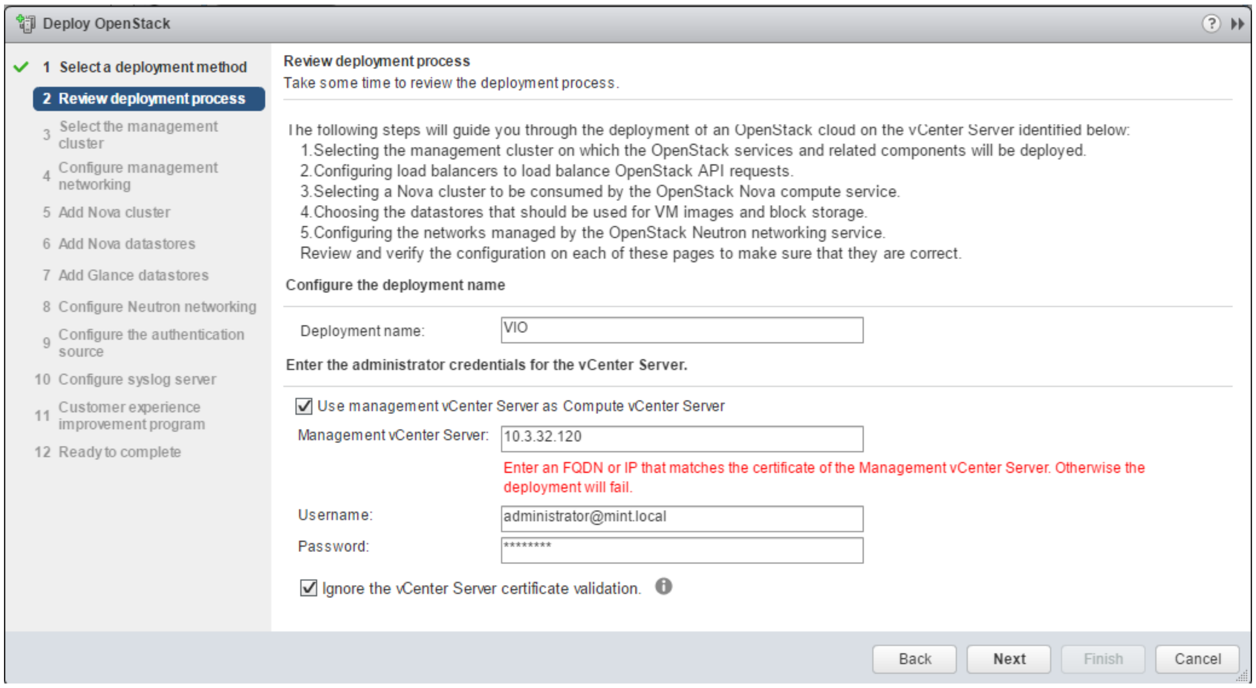
Select the method of deploying OpenStack.

(●) Use this wizard to configure a new OpenStack instance.

( ) Use an exported template to pre-fill configuration settings in this wizard.

Deployment type: Compact

Fig 5.25 Select Compact Deployment type


Deploy OpenStack
✓ 1 Select a deployment method
2 Review deployment process
3 Select the management cluster
4 Configure management networking
5 Add Nova cluster
6 Add Nova datastores
7 Add Glance datastores
8 Configure Neutron networking
9 Configure the authentication source
10 Configure syslog server
11 Customer experience improvement program
12 Ready to complete
Back
Next
Finish
Cancel

Review deployment process

Take some time to review the deployment process.

The following steps will guide you through the deployment of an OpenStack cloud on the vCenter Server identified below:
1.Selecting the management cluster on which the OpenStack services and related components will be deployed.
2.Configuring load balancers to load balance OpenStack API requests.
3.Selecting a Nova cluster to be consumed by the OpenStack Nova compute service.
4.Choosing the datastores that should be used for VM images and block storage.
5.Configuring the networks managed by the OpenStack Neutron networking service.
Review and verify the configuration on each of these pages to make sure that they are correct.

Configure the deployment name

Deployment name: VIO

Enter the administrator credentials for the vCenter Server.

☑ Use management vCenter Server as Compute vCenter Server

Management vCenter Server: 10.3.32.120

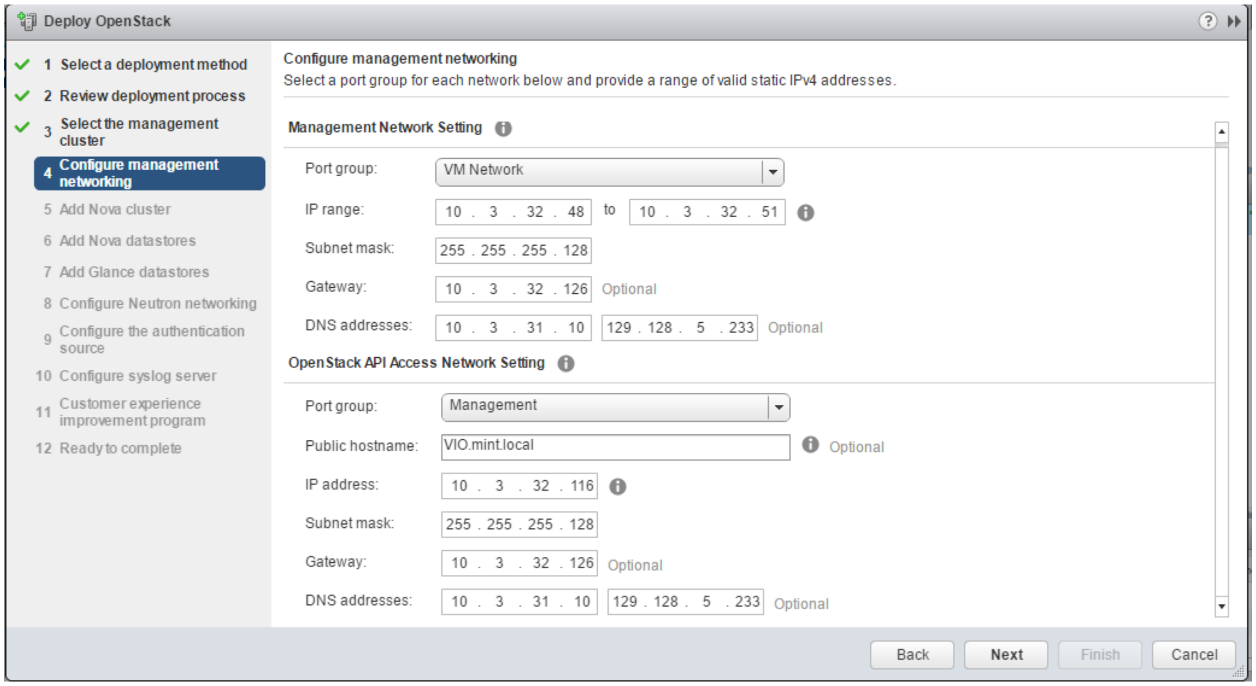
Enter an FQDN or IP that matches the certificate of the Management vCenter Server. Otherwise the deployment will fail.

Username: administrator@mint.local

Password: ********

☑ Ignore the vCenter Server certificate validation.

Fig 5.26 Vcenter Credentials to Deploy VIO


Deploy OpenStack
1 Select a deployment method
2 Review deployment process
3 Select the management cluster
4 Configure management networking
5 Add Nova cluster
6 Add Nova datastores
7 Add Glance datastores
8 Configure Neutron networking
9 Configure the authentication source
10 Configure syslog server
11 Customer experience improvement program
12 Ready to complete
Back
Next
Finish
Cancel

Configure management networking
Select a port group for each network below and provide a range of valid static IPv4 addresses.

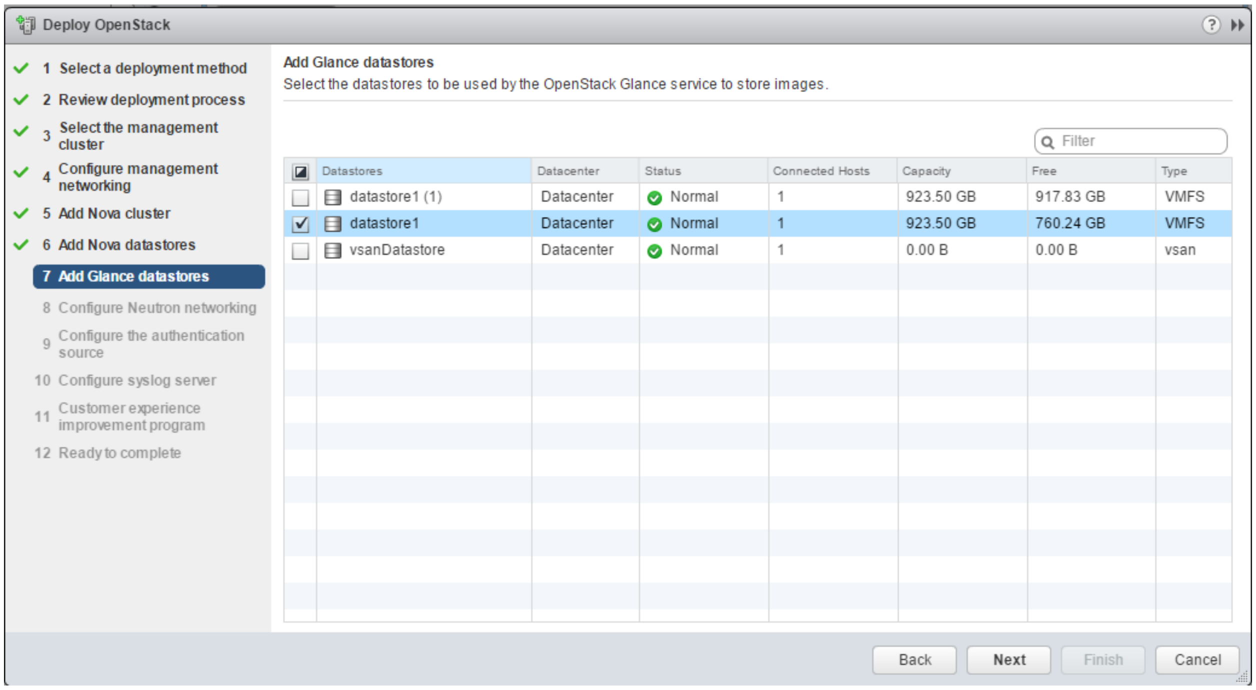
Management Network Setting

Port group:	VM Network
IP range:	10 . 3 . 32 . 48 to 10 . 3 . 32 . 51
Subnet mask:	255 . 255 . 255 . 128
Gateway:	10 . 3 . 32 . 126 Optional
DNS addresses:	10 . 3 . 31 . 10 129 . 128 . 5 . 233 Optional

OpenStack API Access Network Setting

Port group:	Management
Public hostname:	VIO.mint.local Optional
IP address:	10 . 3 . 32 . 116
Subnet mask:	255 . 255 . 255 . 128
Gateway:	10 . 3 . 32 . 126 Optional
DNS addresses:	10 . 3 . 31 . 10 129 . 128 . 5 . 233 Optional

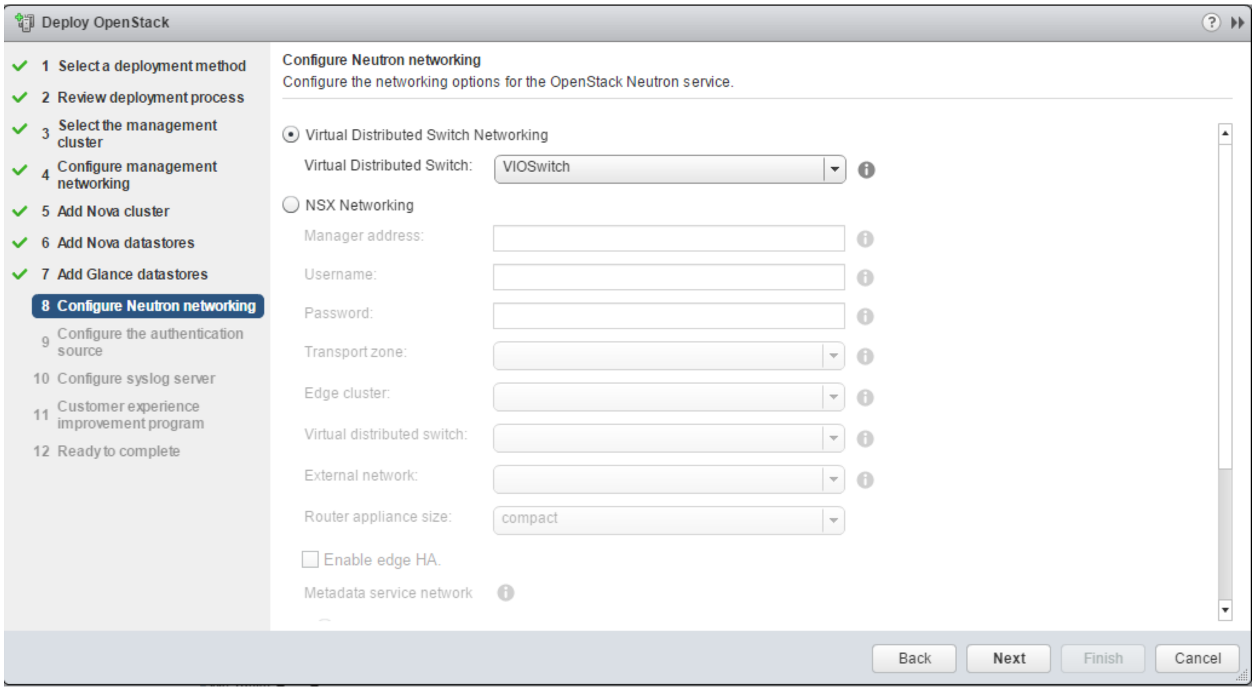
Fig 5.27 Networking Details


Deploy OpenStack
1 Select a deployment method
2 Review deployment process
3 Select the management cluster
4 Configure management networking
5 Add Nova cluster
6 Add Nova datastores
7 Add Glance datastores
8 Configure Neutron networking
9 Configure the authentication source
10 Configure syslog server
11 Customer experience improvement program
12 Ready to complete
Filter
Back
Next
Finish
Cancel

Add Glance datastores
Select the datastores to be used by the OpenStack Glance service to store images.

	Datastores	Datacenter	Status	Connected Hosts	Capacity	Free	Type
☐	datastore1 (1)	Datacenter	Normal	1	923.50 GB	917.83 GB	VMFS
✓	datastore1	Datacenter	Normal	1	923.50 GB	760.24 GB	VMFS
☐	vsanDatastore	Datacenter	Normal	1	0.00 B	0.00 B	vsan

Fig 5.28 Glance Datastore


Deploy OpenStack
1 Select a deployment method
2 Review deployment process
3 Select the management cluster
4 Configure management networking
5 Add Nova cluster
6 Add Nova datastores
7 Add Glance datastores
8 Configure Neutron networking
9 Configure the authentication source
10 Configure syslog server
11 Customer experience improvement program
12 Ready to complete
Back
Next
Finish
Cancel

Configure Neutron networking
Configure the networking options for the OpenStack Neutron service.

[✓ Virtual Distributed Switch Networking]

Virtual Distributed Switch:	VIOSwitch

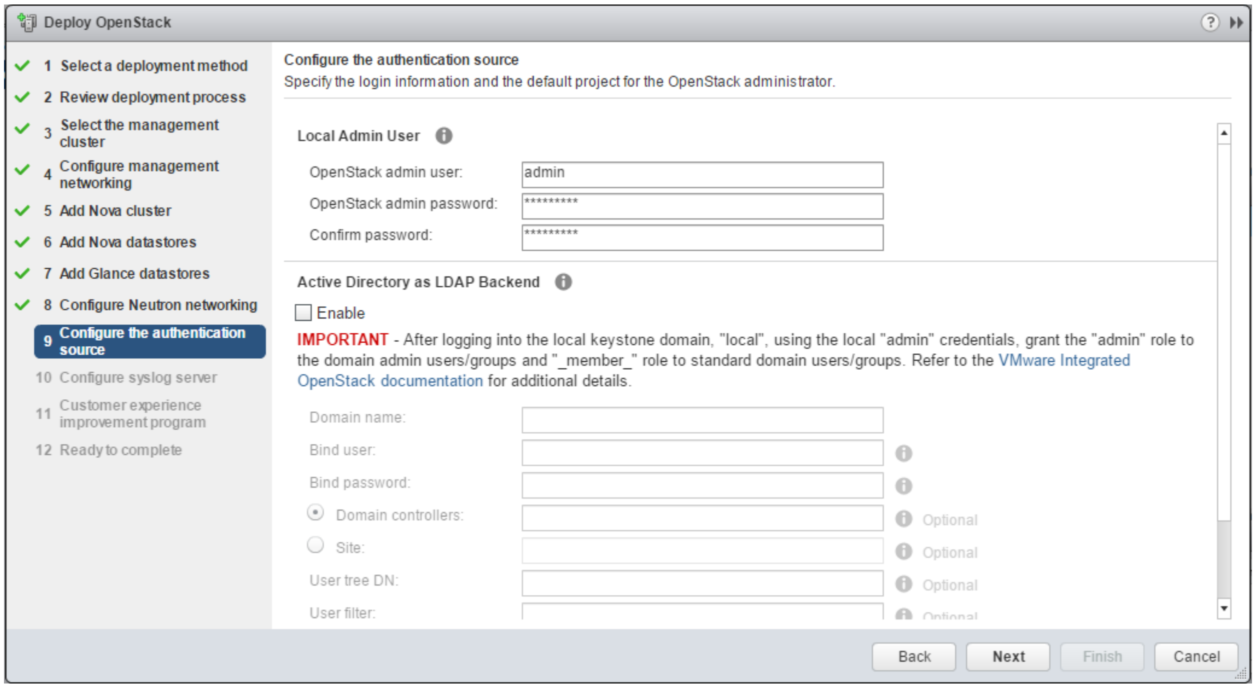
NSX Networking

Manager address:	
Username:	
Password:	
Transport zone:	
Edge cluster:	
Virtual distributed switch:	
External network:	
Router appliance size:	compact

Enable edge HA.

Metadata service network

Fig 5.29 Virtual distributed switch networking


Deploy OpenStack
1 Select a deployment method
2 Review deployment process
3 Select the management cluster
4 Configure management networking
5 Add Nova cluster
6 Add Nova datastores
7 Add Glance datastores
8 Configure Neutron networking
9 Configure the authentication source
10 Configure syslog server
11 Customer experience improvement program
12 Ready to complete
Back
Next
Finish
Cancel

Configure the authentication source
Specify the login information and the default project for the OpenStack administrator.

Local Admin User

OpenStack admin user:	admin
OpenStack admin password:	*********
Confirm password:	*********

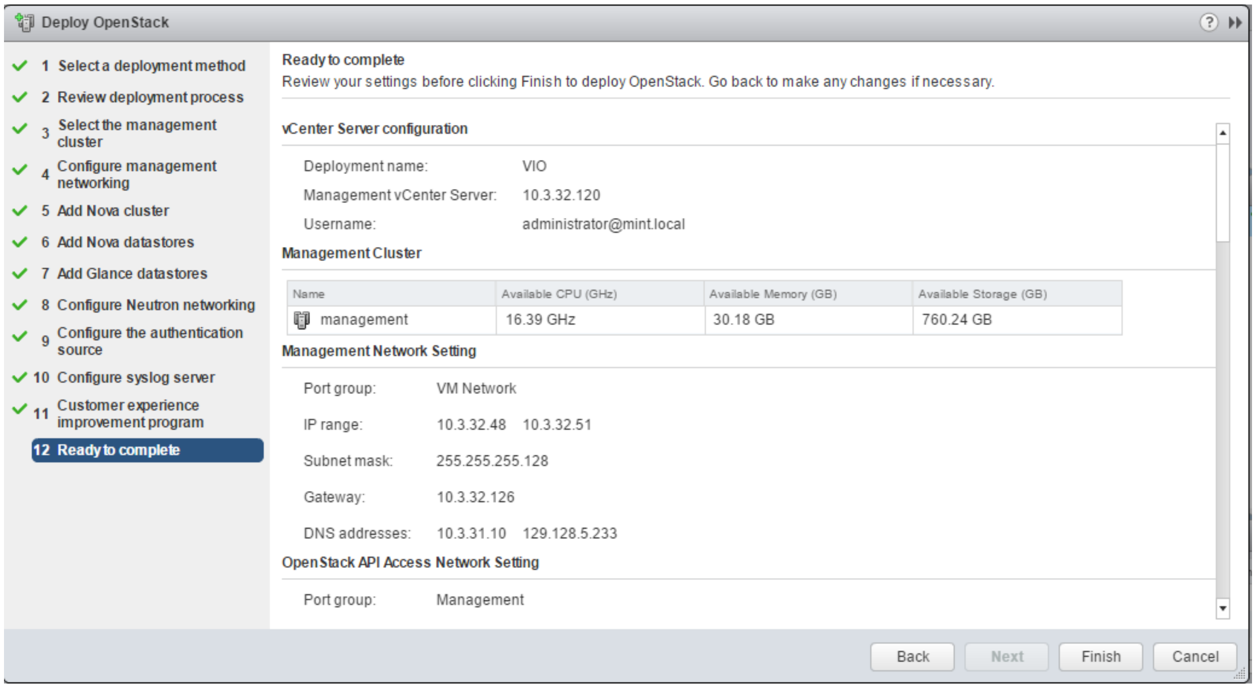
Active Directory as LDAP Backend

Enable

IMPORTANT - After logging into the local keystone domain, "local", using the local "admin" credentials, grant the "admin" role to the domain admin users/groups and "_member_" role to standard domain users/groups. Refer to the VMware Integrated OpenStack documentation for additional details.

Domain name:		
Bind user:		
Bind password:		
Domain controllers:		Optional
Site:		Optional
User tree DN:		Optional
User filter:		Optional

Fig 5.30 Openstack Credentials to access Webpage


Deploy OpenStack
1 Select a deployment method
2 Review deployment process
3 Select the management cluster
4 Configure management networking
5 Add Nova cluster
6 Add Nova datastores
7 Add Glance datastores
8 Configure Neutron networking
9 Configure the authentication source
10 Configure syslog server
11 Customer experience improvement program
12 Ready to complete
Back
Next
Finish
Cancel

Ready to complete
Review your settings before clicking Finish to deploy OpenStack. Go back to make any changes if necessary.

vCenter Server configuration

Deployment name:	VIO
Management vCenter Server:	10.3.32.120
Username:	administrator@mint.local

Management Cluster

Name	Available CPU (GHz)	Available Memory (GB)	Available Storage (GB)
management	16.39 GHz	30.18 GB	760.24 GB

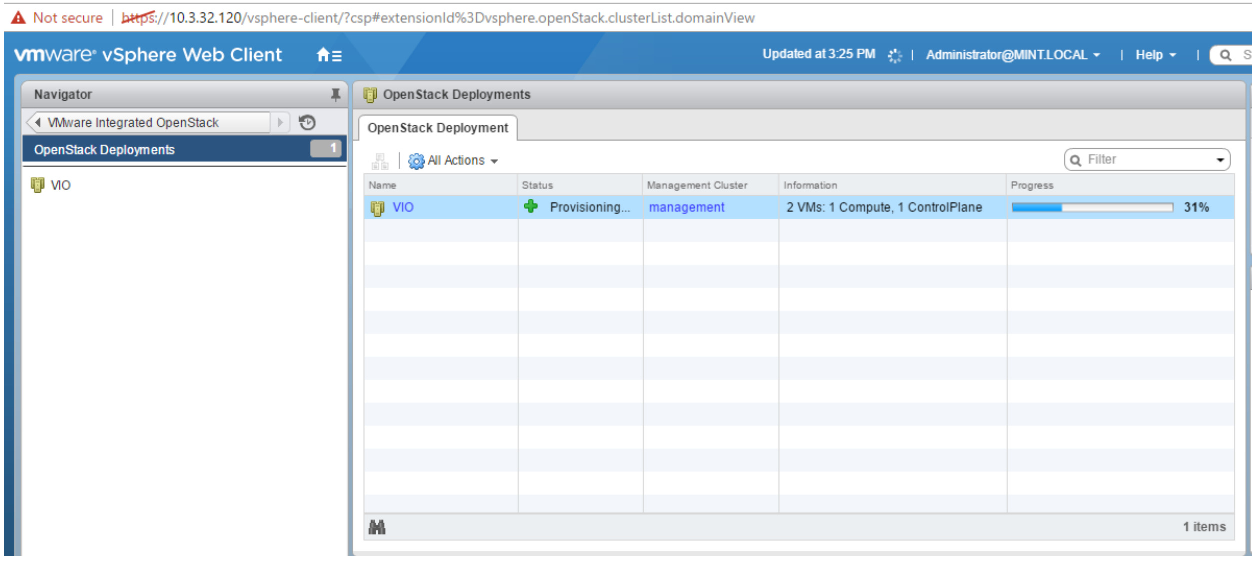
Management Network Setting

Port group:	VM Network
IP range:	10.3.32.48 10.3.32.51
Subnet mask:	255.255.255.128
Gateway:	10.3.32.126
DNS addresses:	10.3.31.10 129.128.5.233

OpenStack API Access Network Setting

Port group:	Management

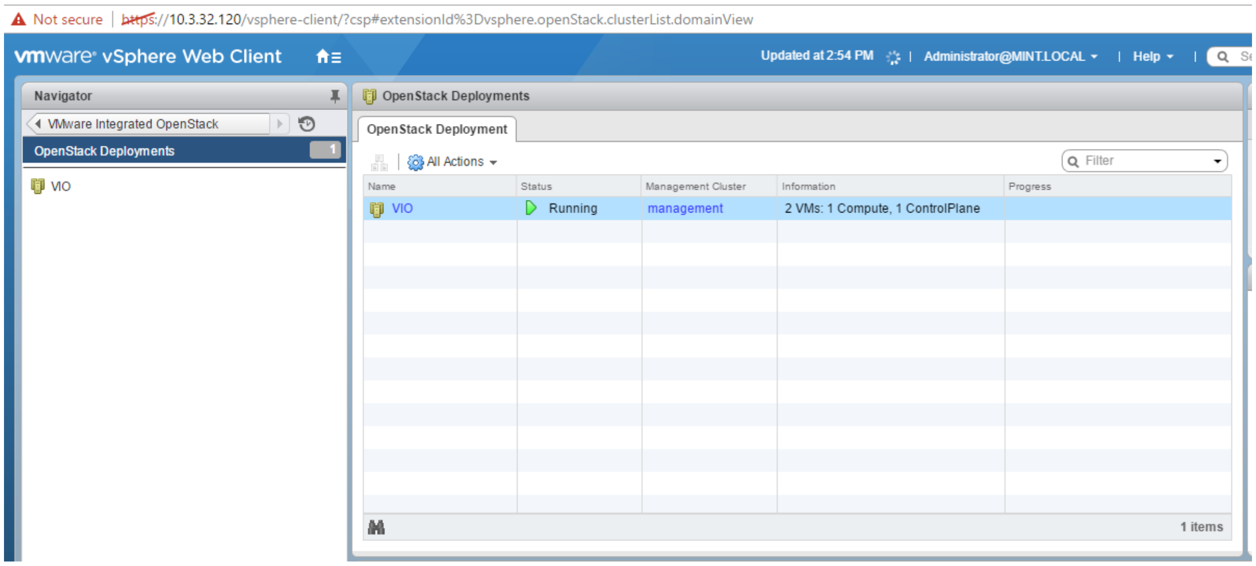
Fig 5.31 Review VIO instance details



Name	Status	Management Cluster	Information	Progress
VIO	Provisioning...	management	2 VMs: 1 Compute, 1 ControlPlane	31%

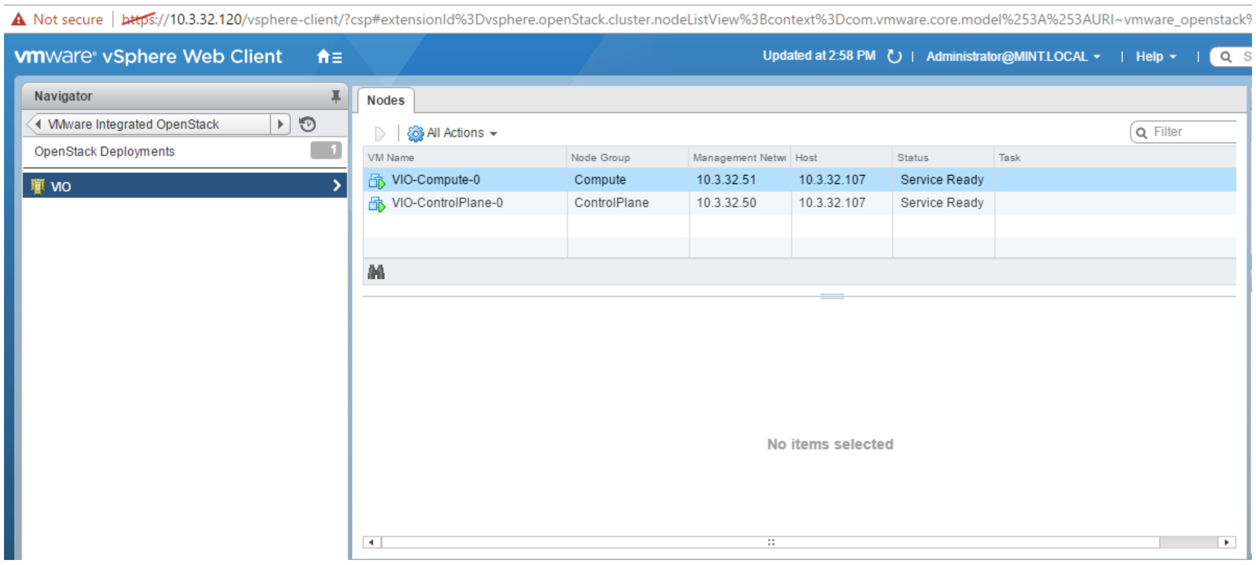
Fig 5.32 Deployment Progress.

Once the deployment is completed we can verify that the instance was created successfully and that all its components and modules are service ready. This can be done by checking the below snaps


Not secure | https://10.3.32.120/vsphere-client/?csp#extensionId%3Dvsphere.openStack.clusterList.domainView
vmware vSphere Web Client
Updated at 2:54 PM | Administrator@MINT.LOCAL | Help
Navigator
VMware Integrated OpenStack
OpenStack Deployments 1
VIO
OpenStack Deployments
OpenStack Deployment
All Actions
Filter
1 items

Name	Status	Management Cluster	Information	Progress
VIO	Running	management	2 VMs: 1 Compute, 1 ControlPlane	

Fig 5.33 Status of VIO deployment


Not secure | https://10.3.32.120/vsphere-client/?csp#extensionId%3Dvsphere.openStack.cluster.nodeListView%3Bcontext%3Dcom.vmware.core.model%253A%253AURI~vmware_openstack
vmware vSphere Web Client
Updated at 2:58 PM | Administrator@MINT.LOCAL | Help
Navigator
VMware Integrated OpenStack
OpenStack Deployments 1
VIO
Nodes
All Actions
Filter
No items selected

VM Name	Node Group	Management Netw	Host	Status	Task
VIO-Compute-0	Compute	10.3.32.51	10.3.32.107	Service Ready	
VIO-ControlPlane-0	ControlPlane	10.3.32.50	10.3.32.107	Service Ready	

Fig 5.34 VIO module status

The Openstack API should be now accessible using the IP address configured. The horizon dashboard will be accessible at http://10.3.32.116/dashboard . Another important thing for this deployment to work is the Virtual distributed switch. There are three alternatives to configure Openstack deployment networking namely, VDS (Virtual Distributed Switch), Vmware Network manager and Neutron networking. I created a VDS for this project to suffice the networking conditions of the Openstack deployment. The configuration for which is shown below. The configuration should also reflect on the esxi host under configuration > Networking > VDS

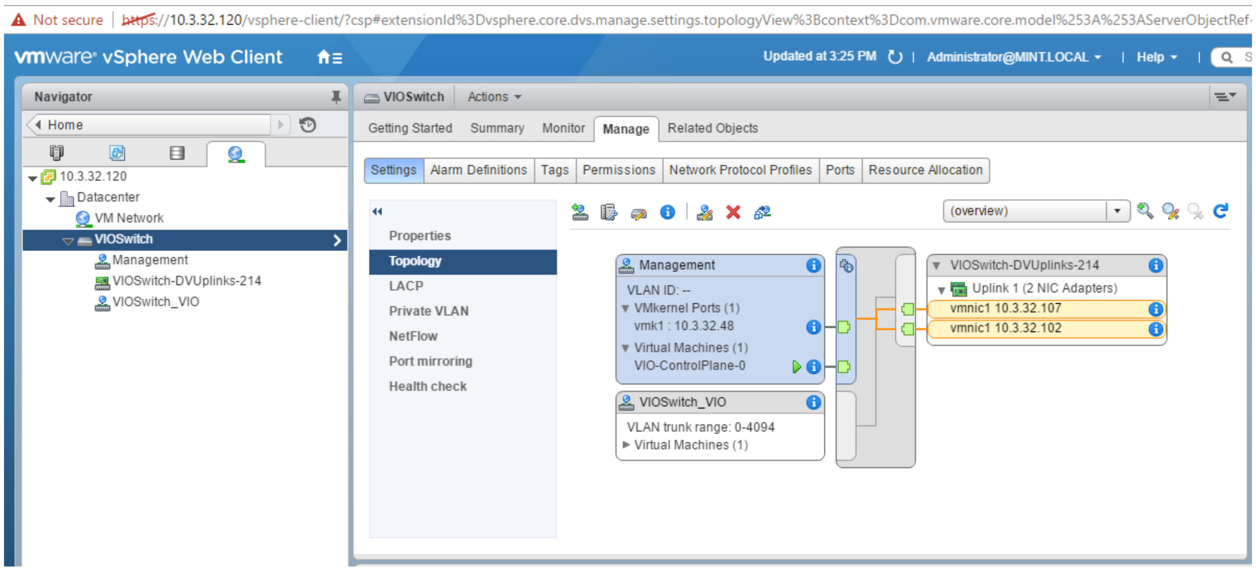


Fig 5.35 Virtual Distributed Switch

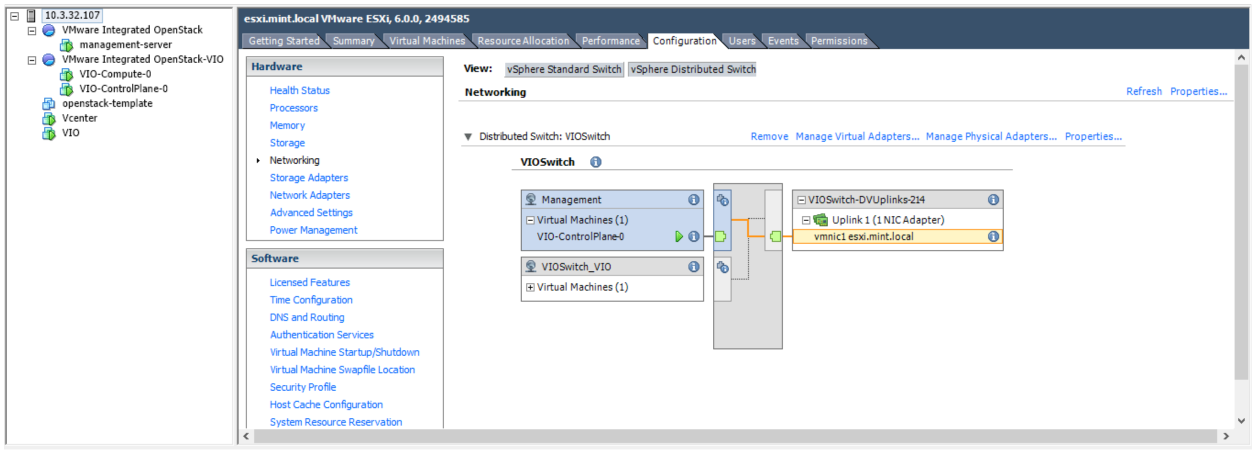


Fig 5.36 esxi VDS

This concludes the deployment process. The next part deals with the Openstack cloud.

5.6 Openstack Configuration

The deployment of VIO compromises a majority of the project. The downside of implementing VIO on a single host is the lack of resources to create virtual machines in cloud using images like Ubuntu server or Windows operating systems. The main steps to configure Openstack clouds is as below

1. Create a new project
2. Create administrator user accounts and public user accounts for both public and private cloud

3. Define and create the public, private and the external cloud networks
4. Create a router and configure it to establish communication
5. Create appropriate rules for both instances
6. Create virtual machines for both public and private clouds.
7. Secure the instance.

I have used the CirrOS image for the spawned virtual machines on private and public clouds.

About CirrOS, it is a minimal Linux distribution that was designed for use as a test image on clouds such as OpenStack Compute [9] and is only 12.7 MB in size. It requires minimum resources to operate and serve the purpose of establishing the cloud scenario we require. A separate flavour was created by me specifically for deploying CirrOS in Openstack. CirrOS is available as a free download on the Openstack webpage [9].

As per my configuration Openstack is accessible via horizon dashboard at http://10.3.32.116/dashboard

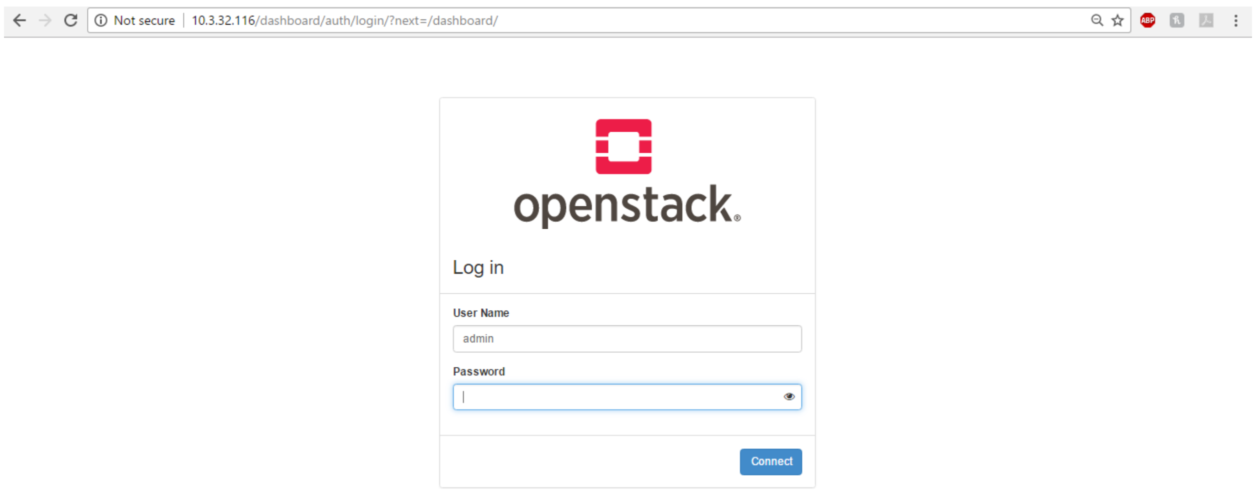
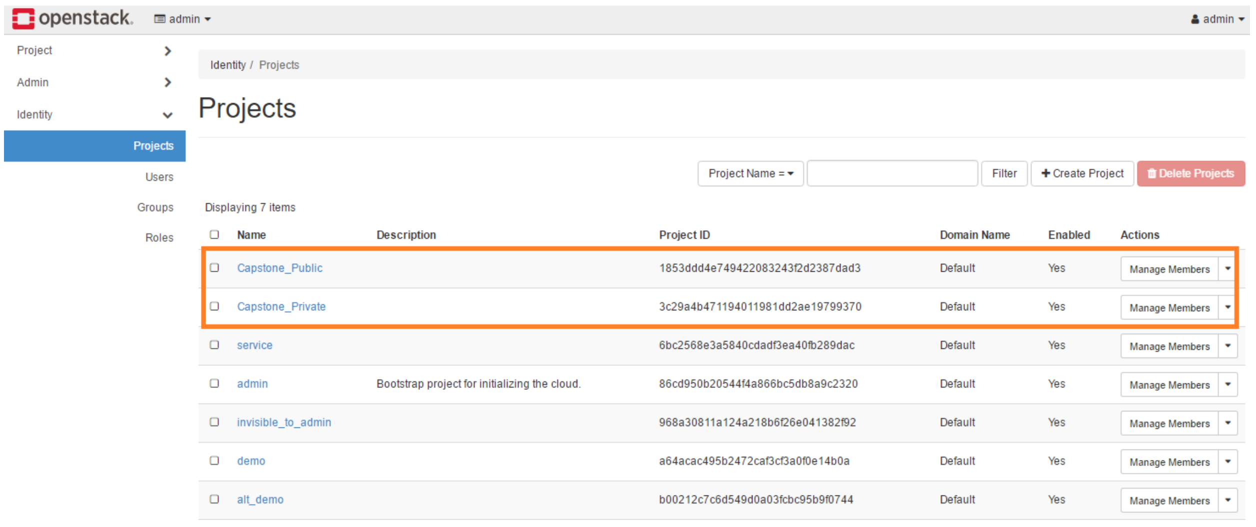


Fig 5.37 Openstack login page

The new projects for public and private clouds created are as below in Fig 5.38. The best resources available to study and prepare for Openstack deployment are readily available. The sample deployment guide provided by Openstack is one of the best guides to refer to while creating your own cloud scenario [10]. The best part about Openstack is that it readily provides plugins to integrate itself with web services from different vendors. There are also paid implementations available that charge on an hourly basis for access to their resources and its utilization. Projects and users can be created and assigned role by the Openstack application admin only. The cloud admins that are created have the ability to deploy networks and manage the cloud itself, however, they cannot delete/create projects or change the assigned roles of the users. The cloud users can create/deploy virtual machines on the cloud and manage them. There is a huge difference in the number of functionalities that an admin possesses as opposed to a user.

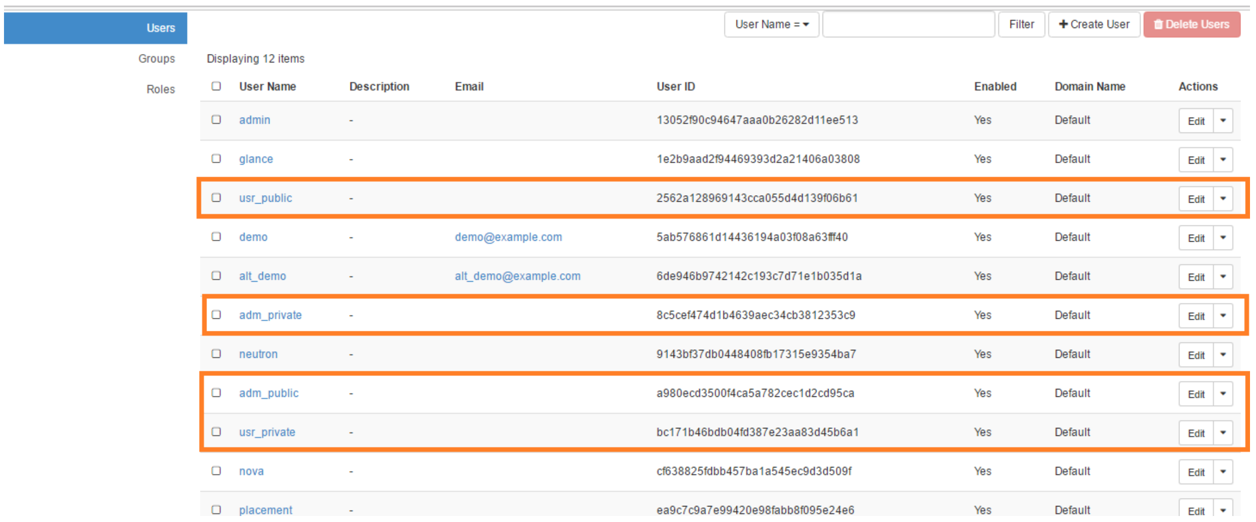


Projects

Displaying 7 items

Name	Description	Project ID	Domain Name	Enabled	Actions
Capstone_Public		1853ddd4e749422083243f2d2387dad3	Default	Yes	Manage Members
Capstone_Private		3c29a4b471194011981dd2ae19799370	Default	Yes	Manage Members
service		6bc2568e3a5840cdadf3ea40fb289dac	Default	Yes	Manage Members
admin	Bootstrap project for initializing the cloud.	86cd950b20544f4a866bc5db8a9c2320	Default	Yes	Manage Members
invisible_to_admin		968a30811a124a218b6f26e041382f92	Default	Yes	Manage Members
demo		a64acac495b2472caf3cf3a0f0e14b0a	Default	Yes	Manage Members
alt_demo		b00212c7c6d549d0a03fcbc95b9f0744	Default	Yes	Manage Members

Fig 5.38 Openstack Projects

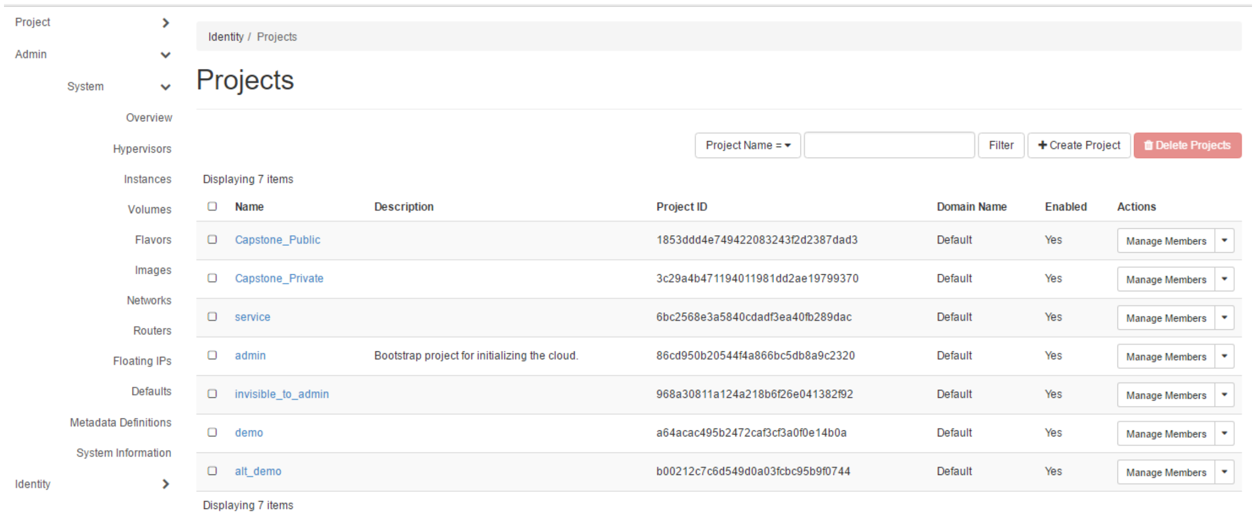


Displaying 12 items

User Name	Description	Email	User ID	Enabled	Domain Name	Actions
admin	-		13052f90c94647aaa0b26282d11ee513	Yes	Default	Edit
glance	-		1e2b9aad2f94469393d2a21406a03808	Yes	Default	Edit
usr_public	-		2562a128969143cca055d4d139f06b61	Yes	Default	Edit
demo	-	demo@example.com	5ab576861d14436194a03f08a63ff40	Yes	Default	Edit
alt_demo	-	alt_demo@example.com	6de946b9742142c193c7d71e1b035d1a	Yes	Default	Edit
adm_private	-		8c5cef474d1b4639aec34cb3812353c9	Yes	Default	Edit
neutron	-		9143bf37db0448408fb17315e9354ba7	Yes	Default	Edit
adm_public	-		a980ecd3500f4ca5a782cec1d2cd95ca	Yes	Default	Edit
usr_private	-		bc171b46bdb04fd387e23aa83d45b6a1	Yes	Default	Edit
nova	-		cf638825fdbb457ba1a545ec9d3d509f	Yes	Default	Edit
placement	-		ea9c7c9a7e99420e98fabb8f095e24e6	Yes	Default	Edit

Fig 5.39 Openstack Cloud admins & users

For the purpose of this project 4 users were create as highlighted in the figure above. The users “adm_private” & “adm_public” serve as cloud administrators for private and public clouds respectively. The users “usr_public” & “usr_private” serve as members of the respective clouds. The difference between a user and an admin account is quite evident from the number of functions available to each one of them. Please refer to figures 5.40 & 5.41

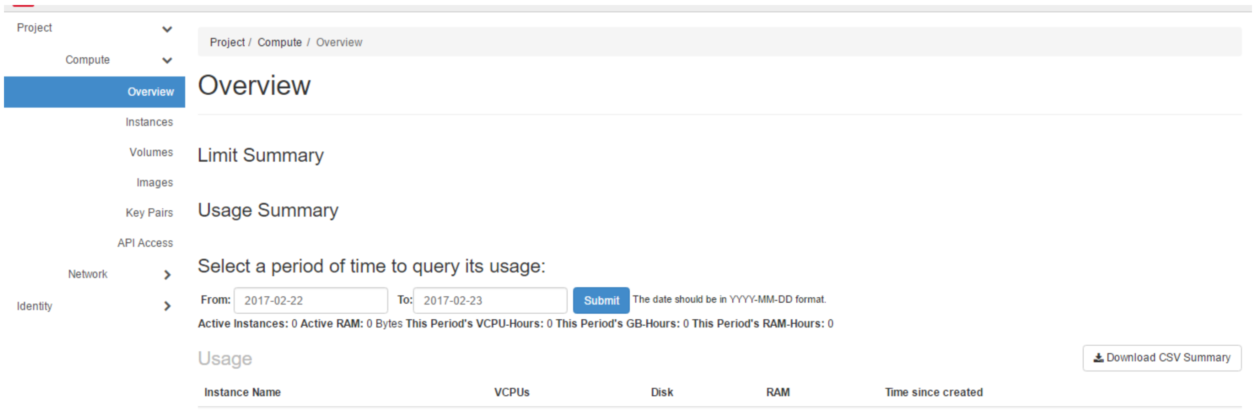


Displaying 7 items

Name	Description	Project ID	Domain Name	Enabled	Actions
Capstone_Public		1853ddd4e749422083243f2d2387dad3	Default	Yes	Manage Members
Capstone_Private		3c29a4b471194011981dd2ae19799370	Default	Yes	Manage Members
service		6bc2568e3a5840cdadf3ea40fb289dac	Default	Yes	Manage Members
admin	Bootstrap project for initializing the cloud.	86cd950b20544f4a866bc5db8a9c2320	Default	Yes	Manage Members
invisible_to_admin		968a30811a124a218b6f26e041382f92	Default	Yes	Manage Members
demo		a64acac495b2472caf3cf3a0f0e14b0a	Default	Yes	Manage Members
alt_demo		b00212c7c6d549d0a03fcbc95b9f0744	Default	Yes	Manage Members

Displaying 7 items

Fig 5.39 Functionalities available to administrator



Select a period of time to query its usage:

From: 2017-02-22 To: 2017-02-23 The date should be in YYYY-MM-DD format.

Active Instances: 0 Active RAM: 0 Bytes This Period's VCPU-Hours: 0 This Period's GB-Hours: 0 This Period's RAM-Hours: 0

Instance Name	VCPUs	Disk	RAM	Time since created

Fig 5.40 Functionalities available to Users

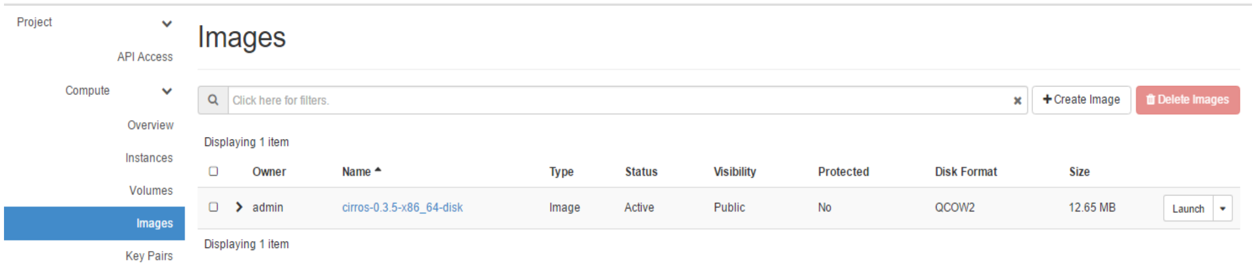
The cloud implementation is pretty basic and simple to understand, the snapshots below are self-explanatory. I will not include the implementation steps for the scenario or the clouds. The Openstack Administrator guide [11] is a great resource and have a detailed explanation of each of the steps and how they impact the installation.

Starting off, once the users are created and appropriate permissions are in place below is the sequence in which the cloud is implemented.

1. Ensure that proper images and flavours are in place.
2. Create Public, Private and Web networks including a router and properly configure them
3. Create Key pair for public and private networks
4. Create security groups and write appropriate rules

5. Launch instances using proper images and flavours for both public and private networks
6. Assigning proper security measures for each type of network
7. Secure the system.

For the purpose of implementation I have used the CirrOS images available on Openstack

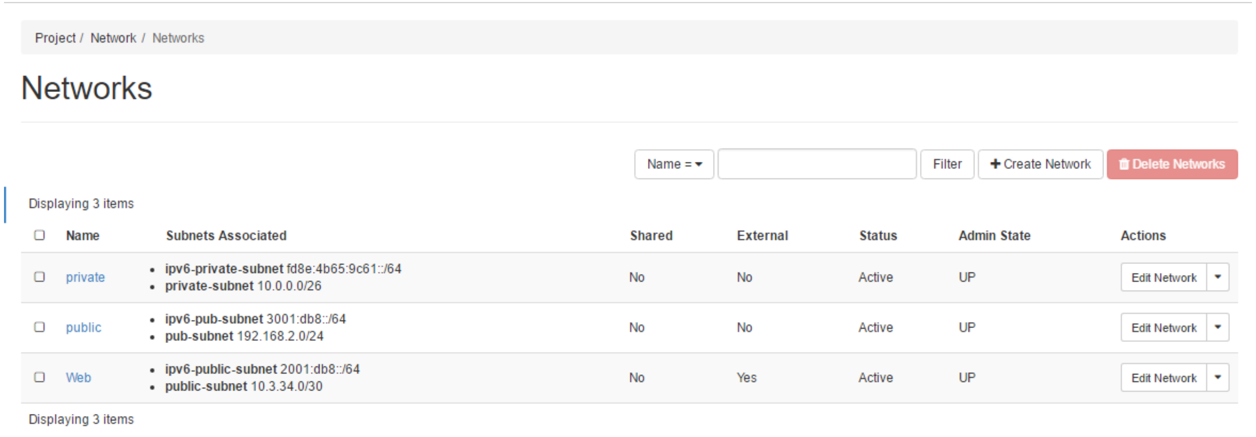


Displaying 1 item

Owner	Name	Type	Status	Visibility	Protected	Disk Format	Size
admin	cirros-0.3.5-x86_64-disk	Image	Active	Public	No	QCOW2	12.65 MB

Displaying 1 item

Fig 5.41 VIO images



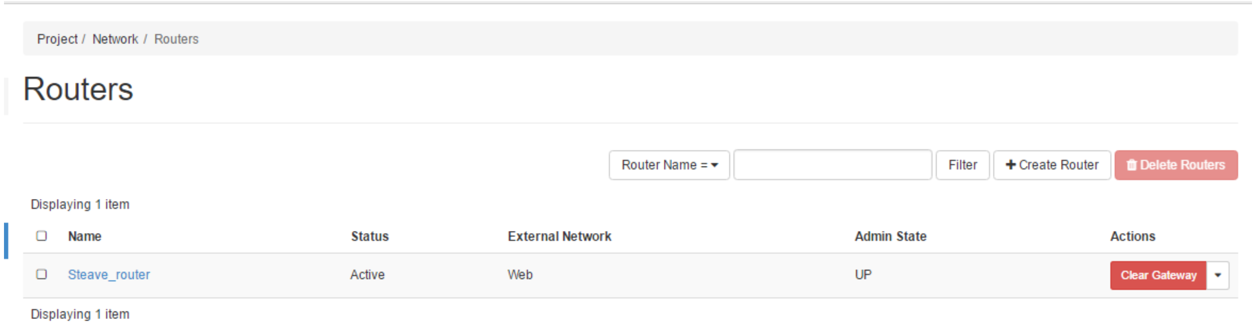
Displaying 3 items

Name	Subnets Associated	Shared	External	Status	Admin State	Actions
private	• ipv6-private-subnet fd8e:4b65:9c61::/64 • private-subnet 10.0.0.0/26	No	No	Active	UP	Edit Network
public	• ipv6-pub-subnet 3001:db8::/64 • pub-subnet 192.168.2.0/24	No	No	Active	UP	Edit Network
Web	• ipv6-public-subnet 2001:db8::/64 • public-subnet 10.3.34.0/30	No	Yes	Active	UP	Edit Network

Displaying 3 items

Fig 5.42 VIO Networks

Three networks are created one for running machines in public domain, one for running machines in private domain and one for accessing the web. All of these networks are interfaced to the router. Static routes can be created to ascertain certain communication between public and private cloud machines.

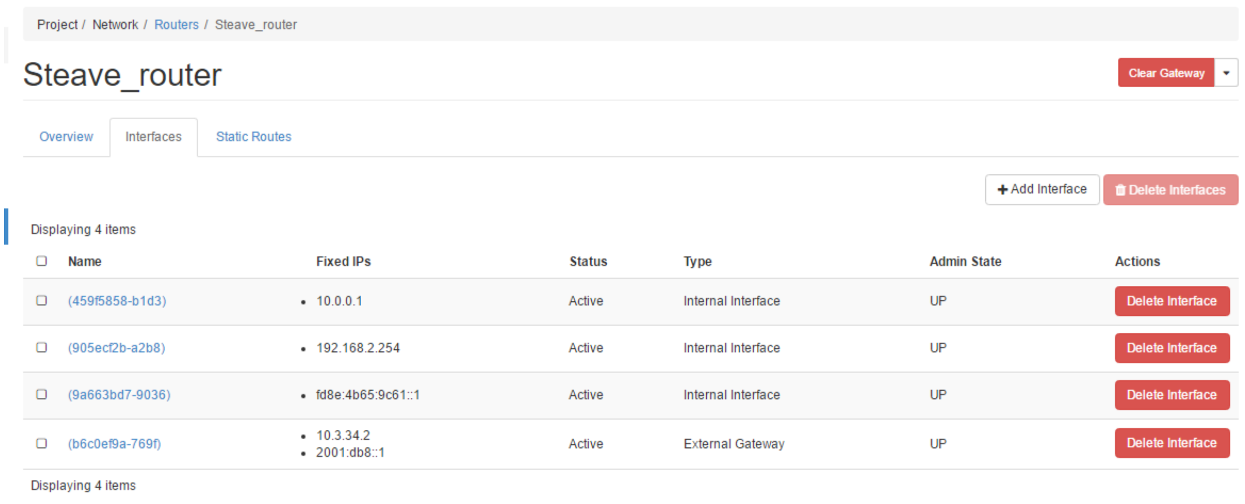


Displaying 1 item

Name	Status	External Network	Admin State	Actions
Steave_router	Active	Web	UP	Clear Gateway

Displaying 1 item

Fig 5.43 VIO router



Displaying 4 items

Name	Fixed IPs	Status	Type	Admin State	Actions
(459f5858-b1d3)	• 10.0.0.1	Active	Internal Interface	UP	Delete Interface
(905ecf2b-a2b8)	• 192.168.2.254	Active	Internal Interface	UP	Delete Interface
(9a663bd7-9036)	• fd8e:4b65:9c61::1	Active	Internal Interface	UP	Delete Interface
(b6c0ef9a-769f)	• 10.3.34.2 • 2001:db8::1	Active	External Gateway	UP	Delete Interface

Displaying 4 items

Fig 5.44 VIO router interfaces

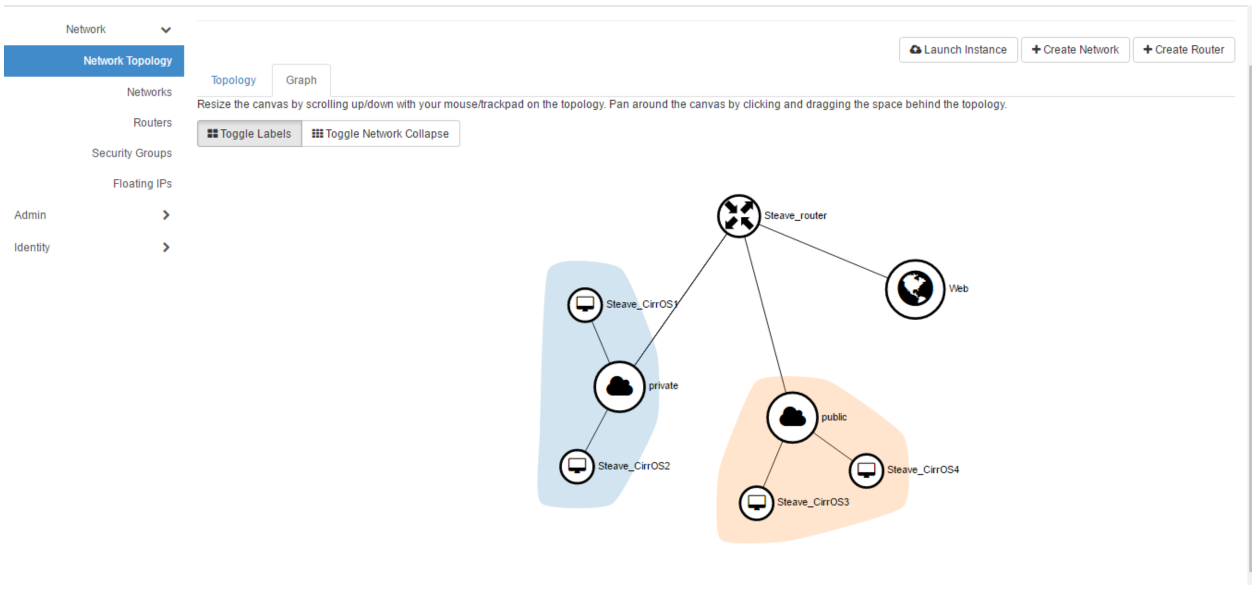


Fig 5.45 VIO Network topology-1

Openstack provides a very rich representation of the implemented scenario both in a topology and a graphical view. The instances in public and private clouds are implemented with appropriate rule sets concerning the security measures and only the respective cloud administrators or users can make changes to them. The web network uses a range of floating IP's that allow each of the machines in public and private clouds to communicate with the Internet.

By default all the rules allow outgoing communication and restrictions are placed on incoming requests or data

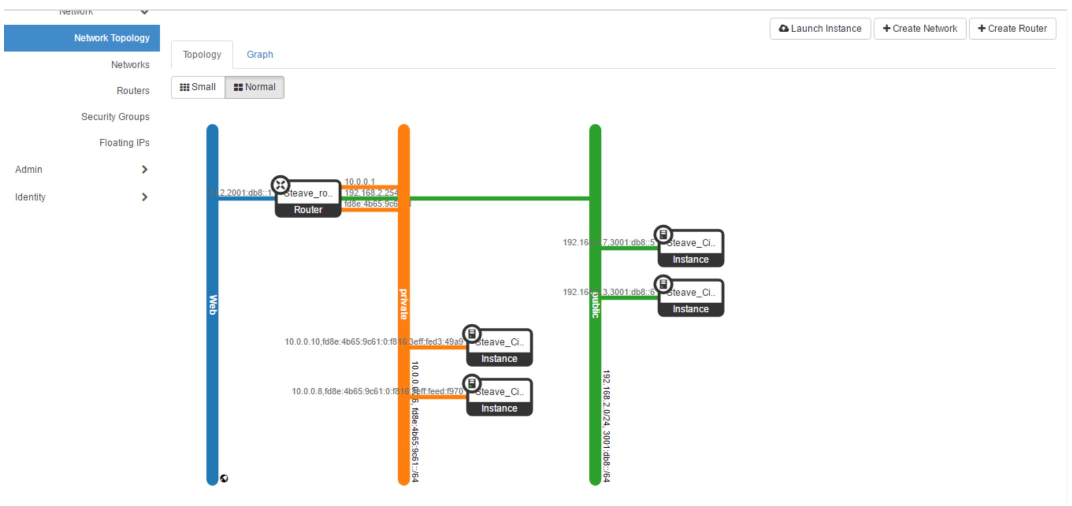
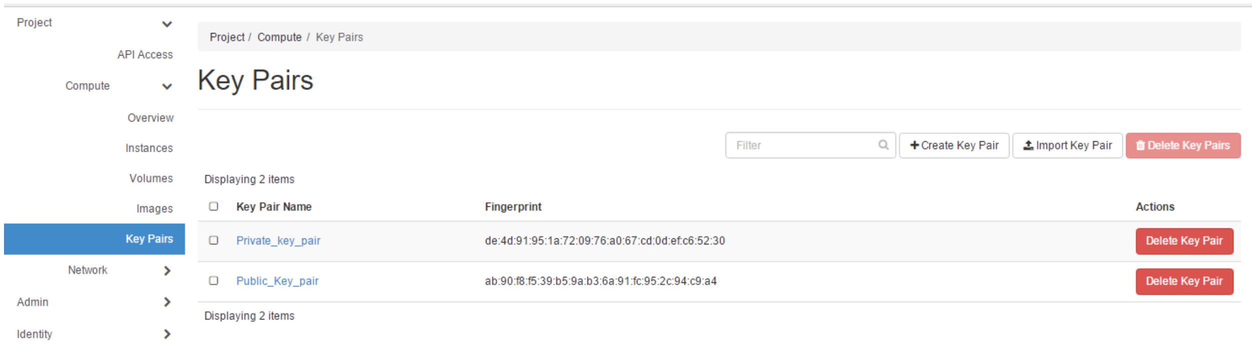


Fig 5.46 VIO Network topology- 2



Key Pairs

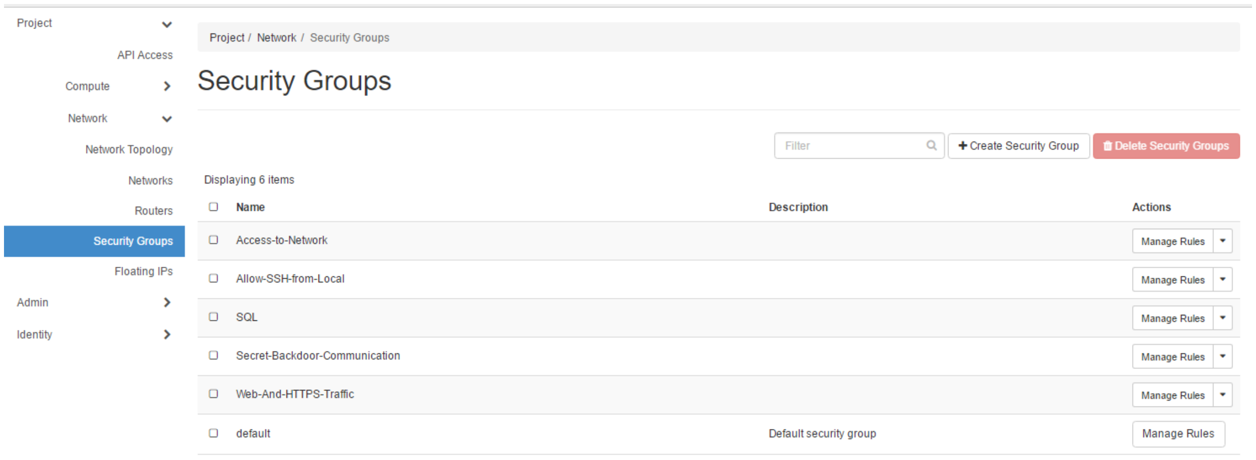
Displaying 2 items

Key Pair Name	Fingerprint	Actions
Private_key_pair	de:4d:91:95:1a:72:09:76:a0:67:cd:0d:ef:c6:52:30	Delete Key Pair
Public_Key_pair	ab:90:f8:f5:39:b5:9a:b3:6a:91:fc:95:2c:94:c9:a4	Delete Key Pair

Displaying 2 items

Fig 5.47 VIO Key pairs

Key pairs are essentially used for authentication. The moment a key pair is configured in the system a local copy of it is available for download. The Key pairs are public and private keys using the RSA algorithm to provide encryption capabilities to the cloud users. A cloud user has to be very careful in handling the keys generated and it serves the basis for both encrypting the outgoing data and decrypting the incoming one.



Security Groups

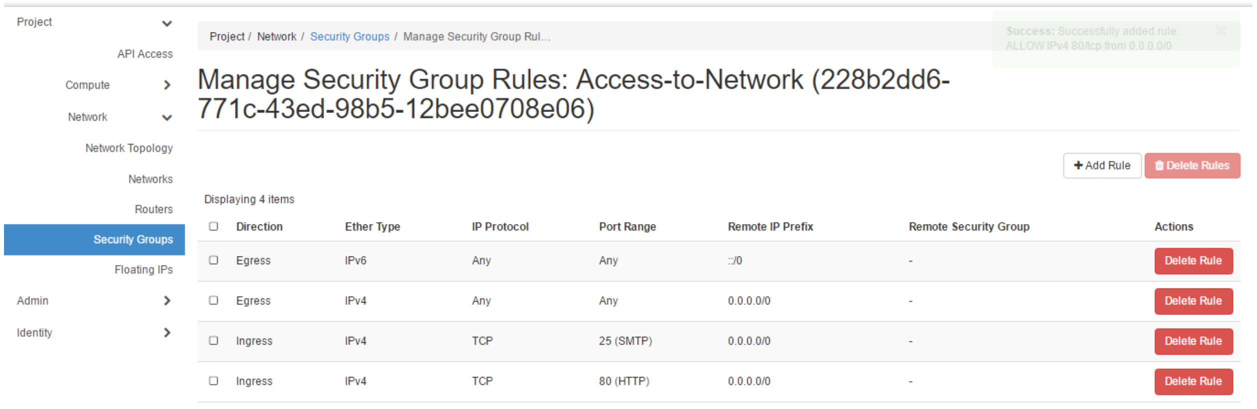
Displaying 6 items

	Name	Description	Actions
☐	Access-to-Network		Manage Rules
☐	Allow-SSH-from-Local		Manage Rules
☐	SQL		Manage Rules
☐	Secret-Backdoor-Communication		Manage Rules
☐	Web-And-HTTPS-Traffic		Manage Rules
☐	default	Default security group	Manage Rules

Fig 5.48 VIO Security Groups

There are many guides and tutorial available that outline the best practices when securing a cloud. Unlike a firewall which is used at a perimeter to filter the traffic passing through it the cloud cannot have a similar implementation and the reason is the vast overhead. Apart from data the cloud exposes a vast amount of resources that can be exploited. Security groups are similar to the rules written for a firewall. They dictate the type of traffic, services or ports that are allowed or banned from the cloud. Using the guidelines outlined [12] I have 5 security groups apart from the default one configured by the system.

1. Access-to-Network: The rules defined are quite simple and straight forward. This security group applies to both private and public cloud. The rules simply state that all the traffic entering the cloud for the ports 25 (Simple mail transfer protocol) & port 80 (Hyper Text Transfer Protocol). This traffic is quite common and will be required by the users in both the clouds.



Success: Successfully added rule: ALLOW IPv4 80/tcp from 0.0.0.0/0

Manage Security Group Rules: Access-to-Network (228b2dd6-771c-43ed-98b5-12bee0708e06)

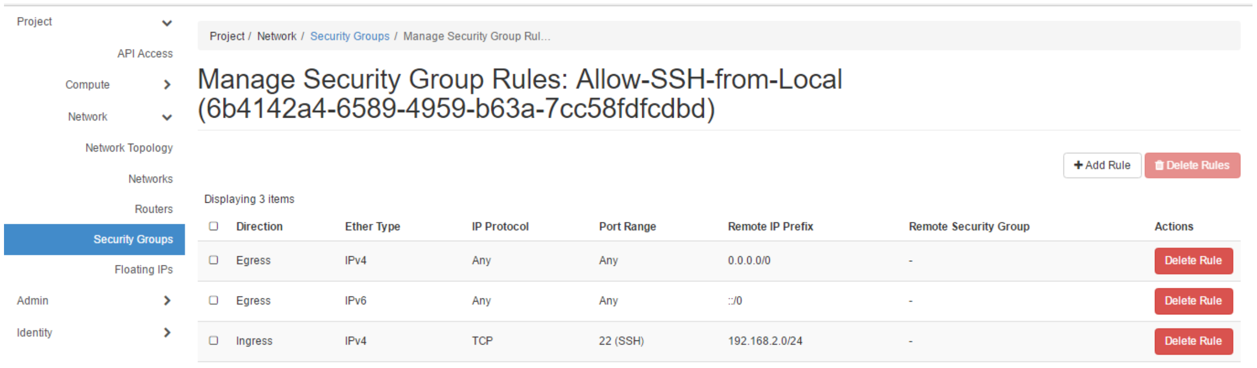
Displaying 4 items

	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
☐	Egress	IPv6	Any	Any	::/0	-	Delete Rule
☐	Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
☐	Ingress	IPv4	TCP	25 (SMTP)	0.0.0.0/0	-	Delete Rule
☐	Ingress	IPv4	TCP	80 (HTTP)	0.0.0.0/0	-	Delete Rule

Fig 5.49 Access-To-Network Security Group

2. Allow-SSH-From-Local: This security group was especially created for the public cloud users. This is evident by the mentioned Remote IP Prefix specified. The rule clearly states

to grant remote access to terminals or systems having IP address in the subnet of 192.168.2.0/24



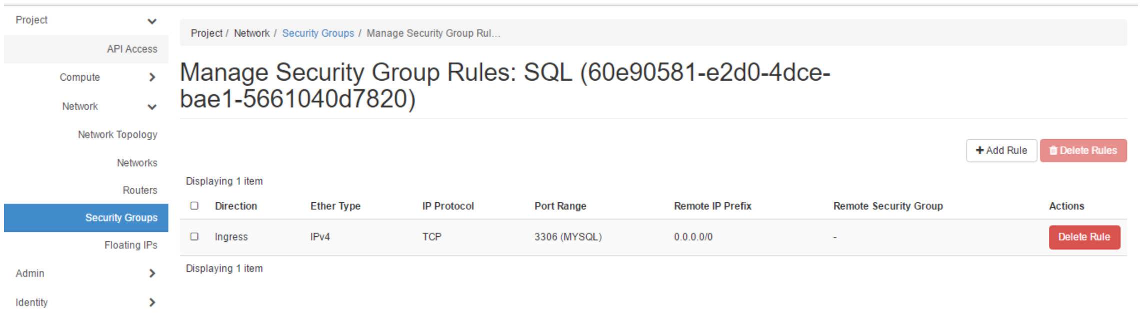
Manage Security Group Rules: Allow-SSH-from-Local (6b4142a4-6589-4959-b63a-7cc58fdfcdbd)

Displaying 3 items

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
Egress	IPv6	Any	Any	::/0	-	Delete Rule
Ingress	IPv4	TCP	22 (SSH)	192.168.2.0/24	-	Delete Rule

Fig 5.50 Allow-SSH-from-Local

3. SQL: This rule applies to both private and public clouds and as a best practice can be a part of the default security group. The main intention is to avoid attacks like SQL injections on the system. The rule really is for systems that need to query databases outside the cloud to retrieve information.



Manage Security Group Rules: SQL (60e90581-e2d0-4dce-bae1-5661040d7820)

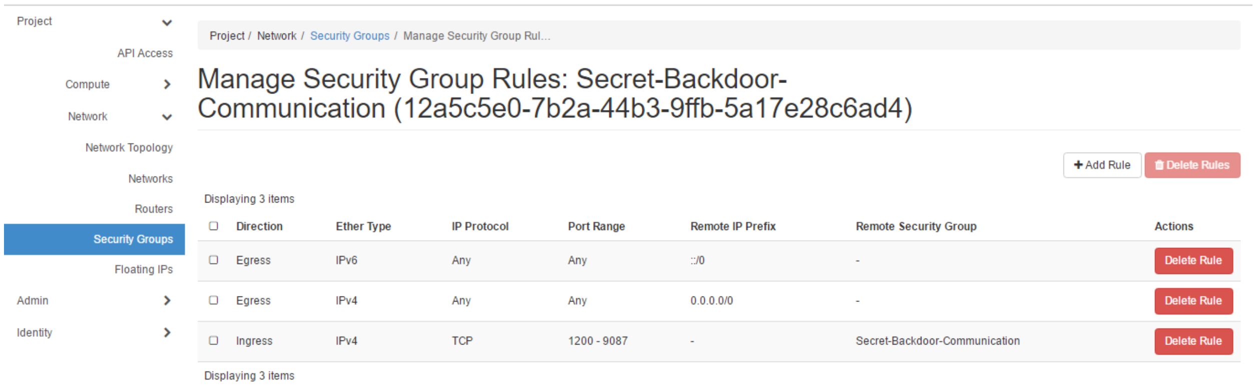
Displaying 1 item

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
Ingress	IPv4	TCP	3306 (MYSQL)	0.0.0.0/0	-	Delete Rule

Displaying 1 item

Fig 5.51 SQL security group

4. Secure-Backdoor-Communication: A security rule that allows Ingress communication to any instance that is part of the same security group, on any port between 1200-9087. Since this is a necessary requirement, you can control and manage your potential exposure by limiting the communication only to those instances that also have the same security rule applied.[12]



Manage Security Group Rules: Secret-Backdoor-Communication (12a5c5e0-7b2a-44b3-9ffb-5a17e28c6ad4)

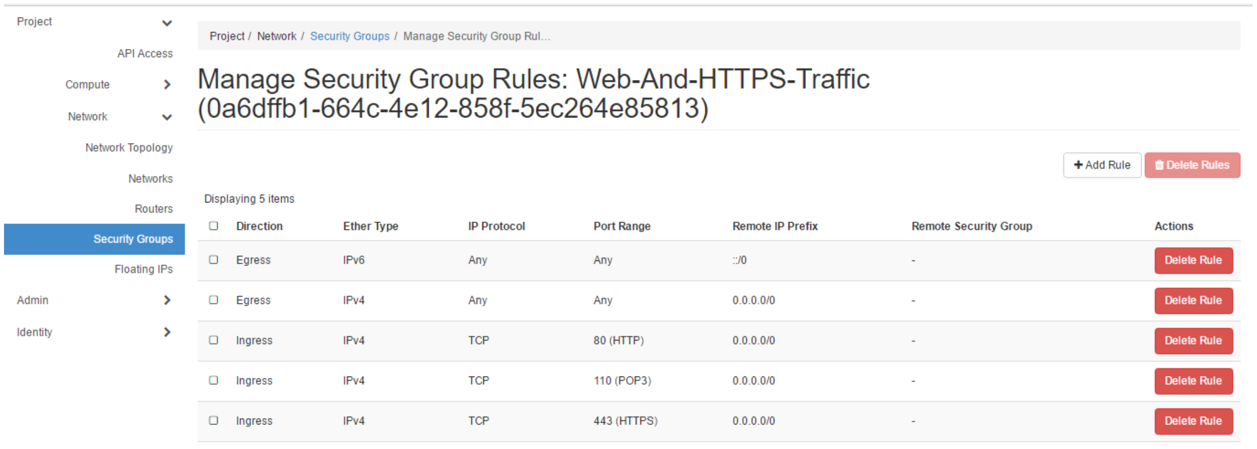
Displaying 3 items

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
Egress	IPv6	Any	Any	::/0	-	Delete Rule
Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	TCP	1200 - 9087	-	Secret-Backdoor-Communication	Delete Rule

Displaying 3 items

Fig 5.52 Secret-Backdoor-Communication security group

5. Web-And-HTTPS-Traffic: The group deals with rules regarding the Web traffic and HTTPS. Almost 50% of all the data used is hosted online and accessible via the web. The web based applications form an essential part of an organization and hence it is necessary to have appropriate rules in place for the same.



Manage Security Group Rules: Web-And-HTTPS-Traffic (0a6dffb1-664c-4e12-858f-5ec264e85813)

Displaying 5 items

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
Egress	IPv6	Any	Any	::/0	-	Delete Rule
Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	TCP	80 (HTTP)	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	TCP	110 (POP3)	0.0.0.0/0	-	Delete Rule
Ingress	IPv4	TCP	443 (HTTPS)	0.0.0.0/0	-	Delete Rule

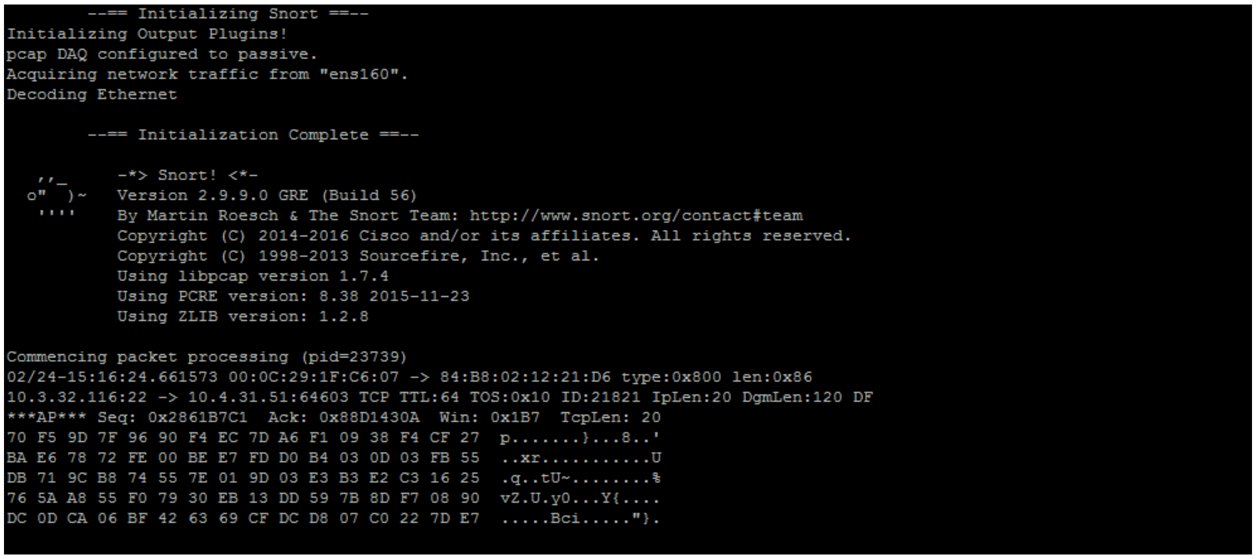
Fig 5.53 Web and HTTPS security group.

The key pair along with the security groups form a very strong case from a security perspective for the cloud. There are many third party tools and plugins available that handle the security of clouds, and in regards to the implementation being the Vshield module from Vmware. However since it’s not available for licensing there are various other alternatives that can be used. For this project apart from Security groups and key pairs that protect the clouds data and resources, Snort is installed as an intrusion detection system to secure the server itself.

5.7 SNORT IDS (Intrusion Detection System)

Snort developed in 1988 is a free and open source Network intrusion prevention/Detection system. Snort is really popular and is widely used. Real time traffic analysis, Packet logging on IP networks, protocol analysis, content searching and matching and detecting probes, network scans and attacks are some of its features. It can be operated to either detect, sniff or prevent any attacks. For the scope of this project Snort is operated in Sniffer mode, in which it constantly checks the network traffic and displays the output on the console. The best part about snort is that it can be readily installed on almost all operating systems. The installation for Snort is quite straight forward with a list of commands available [13]. A very basic form of rules are configured for this instance of Snort as below

1. alert icmp any any -> $HOME_NET any (msg:"ICMP test detected - Steave"; GID:1; sid:10000001; rev:001; classtype:icmp-event;)
2. alert icmp any any -> $HOME_NET any (msg:"IP test detected - Steave"; GID:2; sid:10000002; rev:002; classtype:ip-event;)
3. alert icmp any any -> $HOME_NET any (msg:"UDP test detected - Steave"; GID:3; sid:10000003; rev:003; classtype:udp-event;)



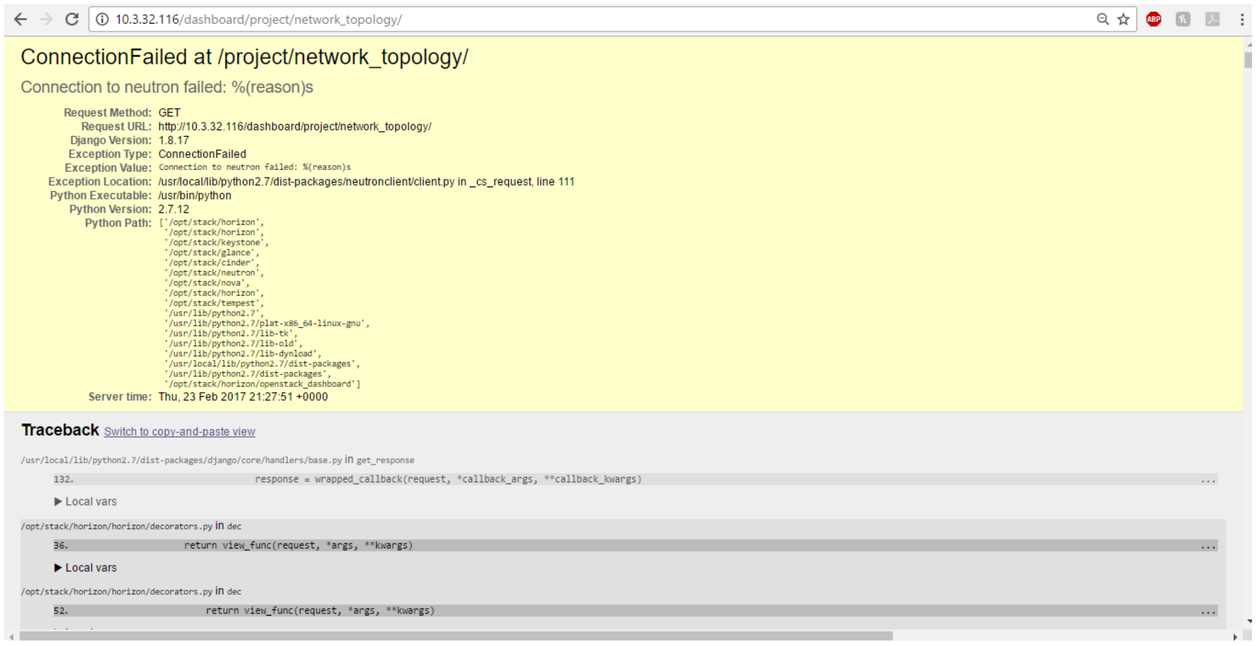
```
        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens160".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.9.0 GRE (Build 56)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2016 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.7.4
           Using PCRE version: 8.38 2015-11-23
           Using ZLIB version: 1.2.8

Commencing packet processing (pid=23739)
02/24-15:16:24.661573 00:0C:29:1F:C6:07 -> 84:B8:02:12:21:D6 type:0x800 len:0x86
10.3.32.116:22 -> 10.4.31.51:64603 TCP TTL:64 TOS:0x10 ID:21821 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0x2861B7C1  Ack: 0x88D1430A  Win: 0x1B7  TcpLen: 20
70 F5 9D 7F 96 90 F4 EC 7D A6 F1 09 38 F4 CF 27  p.......}...8..'
BA E6 78 72 FE 00 BE E7 FD D0 B4 03 0D 03 FB 55  ..xr...........U
DB 71 9C B8 74 55 7E 01 9D 03 E3 B3 E2 C3 16 25  .q..tU~........%
76 5A A8 55 F0 79 30 EB 13 DD 59 7B 8D F7 08 90  vZ.U.y0...Y{....
DC 0D CA 06 BF 42 63 69 CF DC D8 07 C0 22 7D E7  .....Bci....."}.
```

Fig 5.54 Snort running on 10.3.32.116

Chapter 6

Actual vs Expected results.

I did meet the expected result. At the end of the project I had a fully functional cloud as was intended at the beginning and during the proposal stage. Vmware integrated Openstack module deployment was the most essential part of the project and being able to deploy it successfully in an environment with resources substantially less than what is recommended was the best part. Orchestrating the cloud itself on Openstack was pretty easy. The only downside of this implementation was that it is not stable enough. After a host reboot or if more than 5 instances were launched the VIO deployment would crash with the neutron service disconnecting and not being able to communicate with the compute modules. Snap shot is shown in figure 6.1, and every time the service crashed the entire deployment had to be redone and the whole configuration on Openstack had to be done from Scratch. I would have very much liked to install Vshield on the system but the licensing prevented me. Alternative like the Snort Intrusion detection system works well and provides good security.



```
10.3.32.116/dashboard/project/network_topology/

ConnectionFailed at /project/network_topology/
Connection to neutron failed: %(reason)s

      Request Method: GET
         Request URL: http://10.3.32.116/dashboard/project/network_topology/
      Django Version: 1.8.17
      Exception Type: ConnectionFailed
     Exception Value: Connection to neutron failed: %(reason)s
  Exception Location: /usr/local/lib/python2.7/dist-packages/neutronclient/client.py in _cs_request, line 111
   Python Executable: /usr/bin/python
      Python Version: 2.7.12
         Python Path: ['/opt/stack/horizon',
                       '/opt/stack/horizon',
                       '/opt/stack/keystone',
                       '/opt/stack/glance',
                       '/opt/stack/cinder',
                       '/opt/stack/neutron',
                       '/opt/stack/nova',
                       '/opt/stack/horizon',
                       '/opt/stack/tempest',
                       '/usr/lib/python2.7',
                       '/usr/lib/python2.7/plat-x86_64-linux-gnu',
                       '/usr/lib/python2.7/lib-tk',
                       '/usr/lib/python2.7/lib-old',
                       '/usr/lib/python2.7/lib-dynload',
                       '/usr/local/lib/python2.7/dist-packages',
                       '/usr/lib/python2.7/dist-packages',
                       '/opt/stack/horizon/openstack_dashboard']
         Server time: Thu, 23 Feb 2017 21:27:51 +0000

Traceback Switch to copy-and-paste view

/usr/local/lib/python2.7/dist-packages/django/core/handlers/base.py in get_response
    132.                        response = wrapped_callback(request, *callback_args, **callback_kwargs)  ...
    ▶ Local vars
/opt/stack/horizon/horizon/decorators.py in dec
    36.             return view_func(request, *args, **kwargs)  ...
    ▶ Local vars
/opt/stack/horizon/horizon/decorators.py in dec
    52.                 return view_func(request, *args, **kwargs)  ...
```

Fig 6.1 Neutron Error

Conclusion and scope for future work

Cloud is very essential and provides great flexibility to business. Implementing cloud in not sufficient and underlying virtualization hardware is just as important to guarantee the robustness of a cloud infrastructure. With numerous vendors providing a host of services and each with their proprietary technology provide a lot of options and it is very essential for a business to analyze this before settling on a solution. Openstack is a great tool and its advantages lie in the simplicity of its design and its various function. However it is ever changing, with developers from all around the world contributing new functions and modules are being added as fast as every 6 months and it has a lot of potential to grow and provide the kind of flexibility and ease that every business desires

The future scope may involve deploying VIO and then managing it with VMware's own hybrid cloud managers. VMware's functionalities like disaster recovery, Vmotion and high availability can be put to the test. Vmware has some novel solutions like the Vmanager which provides a centralized networking and management for clouds which can be used.

Chapter 7

References

[1] https://rcpmag.com/articles/2016/08/02/microsoft-behind-aws-in-cloud.aspx

[2] https://en.wikipedia.org/wiki/Microsoft_Azure

[3] https://www.openstack.org/assets/pdf-downloads/business-perspectives.pdf

[4] https://docs.openstack.org/security-guide/introduction/introduction-to-openstack.html

[5] https://cloudarchitectmusings.com/2014/08/25/vio-an-openstack-that-vmware-admins-can-love/

[6] https://pubs.vmware.com/integrated-openstack-1/index.jsp?topic=%2Fcom.vmware.openstack.getstart.doc%2FGUID-4B404DEC-BBE3-49FB-BEC4-022A7DAC5E02.html

[7] http://www.virtually-limitless.com/vio/deploying-vmware-integrated-openstack-vio/

[8] https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2146872

[9] https://docs.openstack.org/image-guide/obtain-images.html

[10] https://docs.openstack.org/draft/networking-guide/deploy.html

[11] https://docs.openstack.org/admin-guide/

[12] http://www.stratoscale.com/blog/openstack/openstack-security-groups-best-practices/

[13] http://www.thegeekstuff.com/2010/08/snort-tutorial