UNIVERSITY OF ALBERTA

MINT 709

Capstone Project Report

April , 2014

Building a Multivendor Hybrid Network Consisting of Physical and Virtual Routing and Switching Devices for Cloud Deployment

AHSAN AHMED SHAIKH

Master of Science in Internetworking

Supervisor: Muhammad Durrani

Brocade Communications Systems

Abstract

With the exponentially increasing volume of mobile Internet data traffic driven mainly by voice, video and various other social and business applications, network operators are struggling to reduce their capital and operational expenses and remain profitable. With the emergence of data center virtualization technologies, more and more Service providers as well as Enterprises are building public and/or private cloud infrastructures to offer cloud based services. Cost efficiencies as well as scale and elasticity are being realized via compute and storage virtualization enabling better resource utilization, multi-tenancy, energy optimization and management flexibility etc. While data center virtualization technologies have shown a good promise in reducing both OPEX and CAPEX, so far data centers have been a preferred choice for hosting numerous service applications only. Service providers especially Mobile Network Operators are driving the new era of transformation in the form of Network Functions Virtualization (NFV) which can play a pivotal role in achieving these goals extending the virtualization technologies beyond compute and storage to networking functions. Several vendors are starting to offer broad range of virtual network functions ranging from routers, switches, load balancers and firewalls to base stations and mobile gateways etc. in the virtual form factor that can run on commodity high volume x86 servers and can grow and shrink on a on-demand basis adapting to the traffic demands.

In this project we have developed a Multivendor hybrid network simulating public and private cloud infrastructure consisting of physical and virtual network devices from multiple vendors including Cisco, Brocade, Arista and Juniper. We provide detailed steps for life cycle management of Network Functions Virtualization (NFV) components including instantiation, configuration and provisioning of virtual leveraging VMware hypervisor as the underlying virtualization platform. Several NFV use cases such as virtual Internet Edge router, Route Reflector, Core router and DC Top Of the Rack (TOR) switch use cases are highlighted. We also demonstrate end-to-end connectivity and examine redundancy and security features in this environment.

For our project we use the Master of Science in Internetworking (MINT) lab equipment such as physical routers, switches and servers considering as an enterprise data center in our network topology. We perform an experiment by virtualizing this enterprise data center so that it can be beneficial and helpful for future MINT program graduate students to learn and implement this emerging virtualization technology in the MINT lab.

Acknowledgments

I would like to special thank the MINT program director Dr. M. H. MacGregor and MINT program coordinator Mr. Shanawaz Mir for giving me the opportunity to work on this emerging technology and also providing the equipment in the MINT lab required to implement this project. Their continuing interest in providing the latest equipment to the MINT lab will greatly benefit the present and future graduates at the University of Alberta in Master of Science in Internetworking program.

Table of Contents

Table of Figures

1. INTRODUCTION

With the advent of virtualization technologies, more and more Service providers as well as Enterprises are virtualizing the data centers and building public and/or private Cloud Infrastructures to offer cloud based services. Some enterprises are moving applications to cloud for OPEX/CAPEX reduction as traditional data center architectures are not ideal because they lack the flexibility to support or hosting numerous network application. Traditional enterprise data centers use dedicated servers to run applications which results in a high operational cost and poor server utilization. Others enterprises are hosting applications in the cloud for disaster recovery and/or cloud bursting purposes as well. Either of these scenarios require a secure and dedicated access to cloud infrastructure. One way to provide such an access is via Layer 3 routing gateway devices. Network Functions Virtualization or NFV is another industry movement with a goal to virtualize several network functions such as routing and forwarding for virtualized data centers.

The objective of the project is to build a Multivendor hybrid network for physical enterprise data center and virtualized data center that consists of physical and virtual network devices and demonstrate connectivity, redundancy and security features between them for cloud deployment. In the hybrid network topology, physical routers residing in a enterprise data center will establish BGP peering on the one side with virtual routers instantiated in the public (e.g. Amazon) cloud and on the other side with private cloud so as to extend the underlying enterprise data center.

For virtual routers and switches, open source router softwares such as Brocade Vyatta 5400 Router, Cisco Cloud services Router 1000v, Juniper vSRX virtual services gateway and Arista vEOS switch appliances will be used. VMware hypervisor (ESXI) and vCenter server will be used for carving out VMs on x86 platform and orchestration. All virtual router instances on the private cloud side will run OSPF and BGP with each other and one of the router will be configured as BGP route reflector. On the pubic cloud side two Cisco Cloud Services Router 1000V will establish eBGP peering with Juniper vSRX . Arista network switches will be used for inter-VLAN routing connecting with enterprise data center.

On the physical topology side, physical routers will run OSPF and BGP with each other. There will be physical connectivity via a Ethernet cable between the physical routers and the two servers (i.e. virtual routers and switches). Two physical routers will establish eBGP peering with two virtual routers towards public and private clouds. Each router will advertise a subnet. Routing and forwarding information will be verified and connectivity will be validated using ping and trace commands. Finally, firewall and redundancy features will be tested and verified ensuring highly secured networking with high availability.

1.1 SCOPE

This project presents the design considerations, best practices and implementation guidelines for integrating VMware's ESXI 5.1 infrastructure into the Enterprise data center network, in conjunction with deploying Brocade Vyatta 5400 vRouter at the virtualized data center, deploying Juniper vSRX firewall and Arista vSwitch at the Enterprise data center, and Cisco cloud services

router CSR1000v deployment at the public cloud. We will also Verify end-to-end connectivity for Multivendor topology from private cloud to public cloud and also examine some security and high availability features between virtualized and physical environment.

1.2 Data Centers before Virtualization

Before the age of virtualization, the physical data center typically housed stacks of physical machines accompanied by a range of networking and storage devices with miles of cabling. Each physical machine generally had a dedicated function, for example, a database server, mail server, backup server. applications server and so on. This meant that each machine had its on operating system, application and data. The cost of installing, running and maintaining hardware, software and the cost of engineering resources were extremely high in physical data center.

Following are some information of the costs of running a data center before virtualization:

Design
Lengthy and bespoke design was required for any large scale infrastructure solution project. A virtual network on the other hand is designed to scale. This means that once the application requirements are understood the solution can be sized and quickly deployed**.**

Lead Time
Procuring and provision hardware and software in the traditional data center often took days and sometime weeks. Lead time on network resources was also significant. Virtual machines on the other hand can be deployed within minutes.

Maintenance
The ongoing power costs of running and maintaining the physical data center were high. Different operating systems and storage systems also required ongoing maintenance by a range of experts with different skillsets

Downtime
Patching was difficult on a single system because it required downtime. IT could not provide high availability to all internal customer as it was too expensive. Also hardware and software upgrades required major downtime. Regularly hardware refreshes meant more design more downtime and more costs.

2. TERMINOLOGY AND CONCEPTS

2.1 Virtualization

Virtualization is an abstraction layer meaning that operating system has no longer to be bound to the server that it runs on. The operating system is abstracted from the hardware. In other words, operating system is not directly installed on the hardware instead there is a layer between the physical server and the operating system that we normally install called as Virtualization layer. Once the operating system is separated from the hardware we can utilize the hypervisor (ESXI) to present the complete x86 platform to many virtual machines (OS).

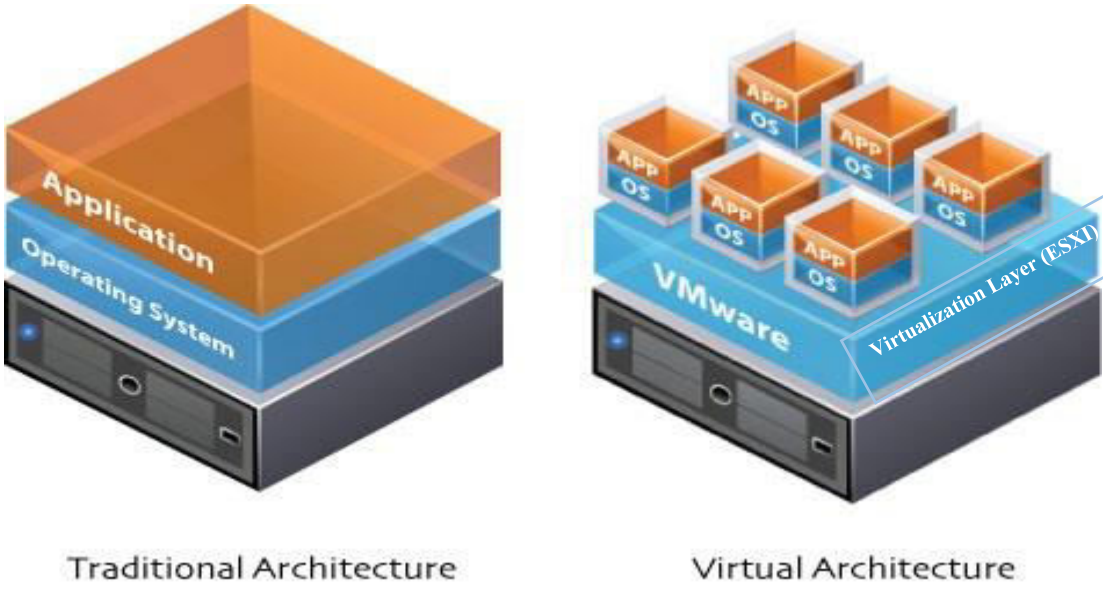


Figure 1: *Traditional Architecture vs Virtual Architecture*

In the above figure, at the right side of the virtual architecture the physical server is at the bottom, virtualization layer is above that which is actually ESXI and then above that there are little blocks called virtual machines. In those virtual machines we actually install the operating system that we used to install on the physical server as shown above in the traditional architecture and then inside that operating system we can install applications. So this allows to run multiple virtual machines each with their own operating system on the same physical server. It is also known as server virtualization which is mandatory for implementation Network virtualization as well in which all virtual machines have to connect with each other and also with the physical network and it is all possible through server virtualization.

2.2 Concept of Network Functions Virtualization

The concept of Network Functions Virtualization basically came from service providers who were interested to accelerate the deployment of new network services to support their growth and revenue objects. They observed the limitations of hardware-based equipment and appliances, so they wanted to implement standard IT virtualization technologies to their networks

Network functions virtualization (NFV) eliminates the need of physical appliance allowing network functions to be completely virtualized running on virtual machines utilizing standard x86 platform. NFV offers a new way to manage, design and deploy networking services. NFV separates the network functions, such as firewalling domain name service (DNS), network address translation (NAT), intrusion detection, caching, etc., from proprietary hardware appliances, so that they can run in software. It's designed to consolidate and deliver the networking components needed to support a fully virtualized infrastructure including virtual servers, routers, switches, firewalls, load balancers, storage and even other networks.

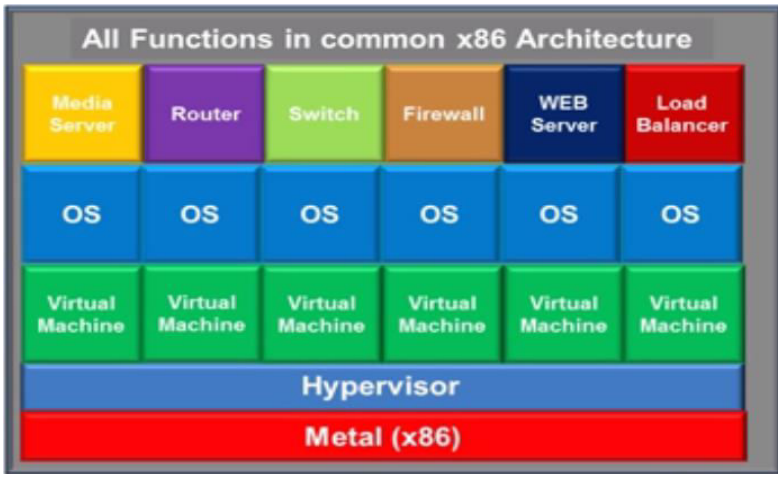


Figure2: *Architecture of Network Functions Virtualization*

In the above diagram, using virtualization on a x86 architecture having a single server running x86 platform and on top of it the hypervisor is running. On the top of hypervisor install all the virtual machines and each virtual machine associated with its own independent operating system and in turn running router, switch, firewall, load balancer, web server or media server etc.

Benefits of Network Functions Virtualization

- Orchestration by managing thousands of devices
- Lower CAPEX and OPEX through reduced power consumption and reduced equipment cost.
- Openness to the virtual appliance market and pure software entrants
- Less complex architecture and very flexible to implement
- Reduced time-to-market to deploy new network services
- Opportunities to trial, test and deploy new innovative services at lower risk
- Automation
- Optimizing network device utilization

2.3 Virtual Networking

Virtual Networking is the building block of an IP network that is built primarily for virtual machines. This virtual IP network helps to connect virtual machines to the physical network. Just like physical machines require physical network for communication, in the same way virtual machines require virtual network to communicate with each other and also to integrate virtual environment with the physical environment and allows seamless connectivity of virtual devices with physical devices. Virtual networking requires the same components as the physical networking. Physical machine requires a physical NIC card and a virtual machine requires virtual NIC card. If a physical machine requires a physical switch in order to have communication with other devices, in the same way virtual machine requires virtual switch for communication with other virtual devices. So in virtual networking all the hardware is virtual.

2.4 Hypervisor

Hypervisor creates the virtualization layer for making the server virtualization possible. It contains the Virtual Machine Manager (VMM) whose task is to manage multiple virtual machines that are running on a single virtual host or single physical server. Examples of hypervisors are VMware ESXI, Microsoft Hyper-V, Citrix XenServer, etc. There are two types of hypervisors: Type 1 hypervisor and Type 2 hypervisor. Type 1 hypervisor is loaded directly on the hardware while Type 2 hypervisor is loaded in an operating system running on the hardware. Type 1 hypervisors are basically used in data centers having a dedicated physical server on which hypervisors are installed directly and create virtual machines according to the requirements. We will use Type 1 hypervisor for our project.

2.5 Virtual machine

Virtual machine (VM) is a software implementation of a computing environment in which an operating system (OS) can be installed and run. Like the physical machine, a virtual machine requests for CPU, memory, hard disk, network and other hardware resources that are completely managed by a virtualization layer which translates these requests to the underlying physical hardware. Virtual machines are created within a virtualization layer, such as a hypervisor. The virtualization layer can be used to create many individual, isolated VM environments.

2.6 Virtual Standard Switch

A virtual standard switch is modelled on a physical Ethernet switch responsible for connecting virtual machines in a virtual network. But it is software based exists on ESXI server and can be managed at the ESXI host level using vCenter server. Just as the physical switch consists of ports so does a virtual switch. Ports on virtual switch provide logical connection point among virtual devices to communicate with each other and also to communicate between virtual and physical devices. These ports are like virtual RJ45 connectors. By connecting virtual machines to the ports

of virtual standard switch so that they can communicate with other virtual machines on the same ESXI host. It is useful when two vital machines want to communicate with each other without connecting to the outside world or physical environment. These are known as port groups of virtual standard switch. If the virtual machines want to connect to the physical network or vice versa then it needs to be connected to the uplink adaptors which are also called pNICs. An uplink adaptor is a physical NIC associated with the host server. Uplink adapters use virtual objects called vmnics, or virtual network adapters, to interface with the vSwitch. It connects the physical network to the virtual network

On a more technical level, a vSwitch attaches to the VMkernel (vmknic) inside a host server. The vSwitch is responsible for routing network traffic to the VMkernel, the VM network, and the Service Console. The VMkernel is used to manage features like vMotion, fault tolerance, network file system (NFS), and Internet small computer system interface (iSCSI); the VM network enables virtual machines running on an ESXi host to connect to the virtual and physical network; and the Service Console is used for remote management. In ESXi, the VMkernel instead serves as the management front-end.

The virtual machines must have a virtual NIC (vNIC) mapped to it in order to connect to a vSwitch. Just like the physical machines can't connect to a network without a working network adapter. Like pNICs, vNICs have both a MAC address and an IP address. Each virtual machine interfaces with the vSwitch via a port. vSwitch can consist of one or more port groups, which describe how the virtual switch should route traffic between the virtual network and the virtual machines connected to the specified ports. A vSwitch starts out with 120 ports, by default, but can be configured to use up to 4,088 ports, and up to 20 network adapters can be associated with a host.

A vSwitch that is associated with two or more adapters is called a teamed vSwitch; these switches provide an added layer of protection to a network and are used for fault tolerance and load balancing.

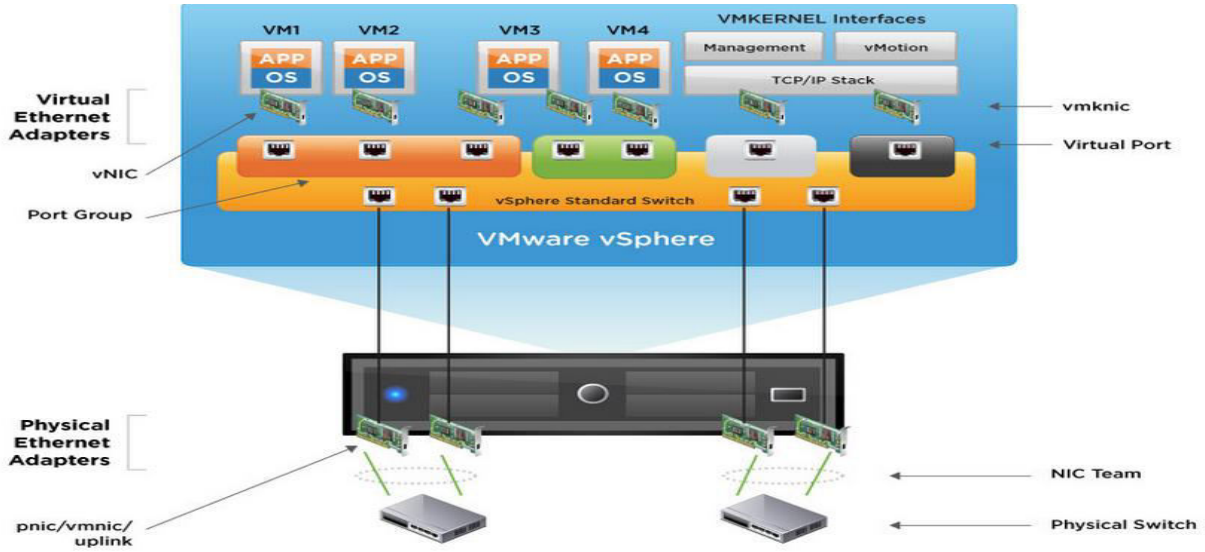


Figure 3: *Virtual Standard Switch Architecture*

2.7 Network Virtualization Challenges to Date

The adoption of server virtualization over the past decade has resulted in a completely new operational model for provisioning and managing application workloads in the data center. However, the operating system model of the network to which these dynamic workloads are connected has not kept pace. The network is now a barrier to achieving the full benefits of virtualization because application provisioning is manually intensive taking days or weeks to provision even simple network topologies to support applications. Workload placement and mobility is limited by physical network constraints and topology. The network is operational intensive, requiring significant ongoing manual hardware configuration and maintenance and vendor specific expertise. The network operational model is the same as it has been for 25 years, designed in a time when workloads were static and ran directly on physical servers. This antiquated operational model is broken and is now a barrier to achieving the full benefits of virtualization

2.8 Considerations when Deploying Virtual Network Components

We can achieve many of the same features, benefits and guarantees with virtual network components as one can get with the physical network components. In fact, tasks that take a long time to achieve in a physical network are performed much faster in a virtual network such as design application setup, server provisioning, maintenance and disaster recovery. Deployment of virtual network can be done without interacting with the underlying network hardware. So, there is no need to run a cable, or rack, a server for example, Virtual switches and routers run their own operating system software. The key difference in a virtual network is that it's possible for this software to run on any hardware from different vendors.

3. VIRTUALIZATION TOOLS FOR PROJECT

3.1 VMware ESXI

VMware ESXI also known as VMware hypervisor is an operating System specially built for Virtualization. We will be using ESXI 5.1 version that will only run on servers with 64-bit x86 CPUs and need atleast Quad cores with 16 GB RAM. ESXI Operating system has a small footprint that can be easily installed on a USB drive or embedded memory. For 64-bit virtual machines, support for hardware virtualization must be enabled on x86 CPUs. Once the ESXI is installed and configured it can be completely managed by VMware vSphere client. ESXI supports windows, Linux, BSD virtual machines and some other OS based virtual machines. ESXI needs atleast two NICs to separate the management traffic from the virtual machine traffic.

Figure 4: *VMware ESXI*

3.2 VMware vSphere Client

VMware vSphere client also known as VMware virtual Infrastructure client is a primary interface for managing all aspects of virtual infrastructure environment and provides access directly to VMware ESXI server so that virtual machines can be configured and managed. vSphere Client is a windows application that allows for connecting directly to VMware ESXI or to a vCenter Server. It allows to manage the ESXI or vCenter server directly on windows machine. VMware vSphere client can be installed on any number of windows machine . When we connect directly to the ESXI and wants to access it through vSphere client it requires the Administrative username and password for the login for that specific ESXI host. The username and password will be the same which we will configure for ESXI during the installation of ESXI.

Figure 5: *VMware vSphere Client*

3.3 VMware vSphere Web Client

The VMware vSphere web client operates in the same way as VMware vSphere client does for accessing the ESXI server. The only difference is that, for login credentials in vSphere web client it can be accessed through web browser by providing the IP address of the ESXI server.

3.4 VMware vCenter Server

The VMware vCenter server is a management platform that allows to centralize the management of all the ESXI hosts and the virutal machines running on those ESXI hosts. There are upto 1,000 ESXI hosts can exist per vCenter server instance as well as upto 10,000 powered-on virtual machines per vCenter instance. It can be deployed by two methods:

1. vCenter Server can be deployed as a virtual appliance that runs the SUSE Linux operating system.
2. It can also be deployed on a physical host as Windows based vCenter server.

The vCenter server also requires some additional components like database server and Active directory. vSphere client application is used to access and manage the vCenter server environment.

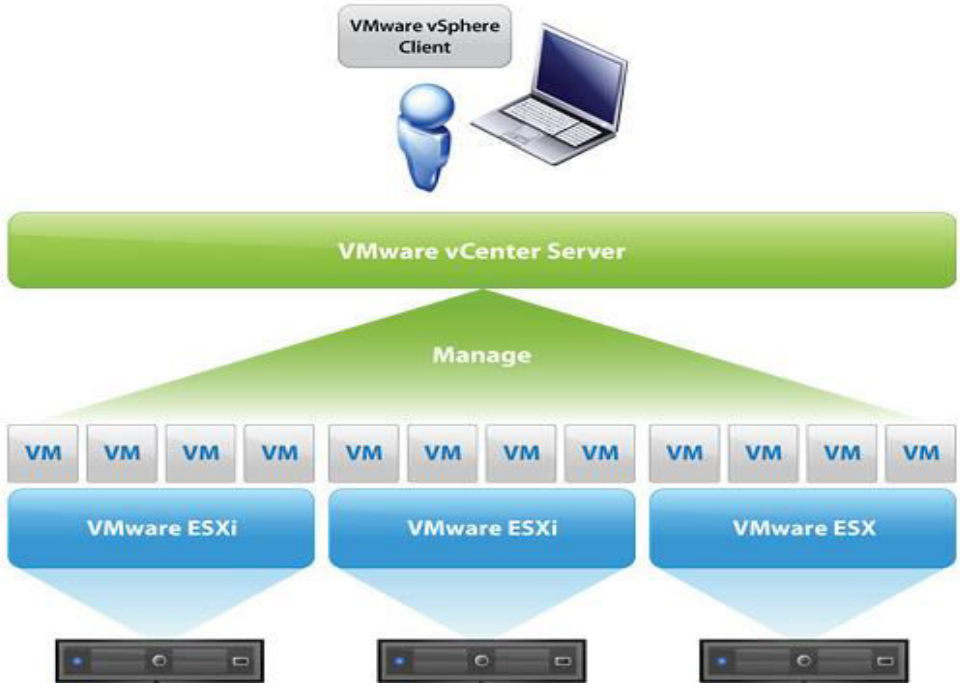


Figure 6: *vCenter Server Management Platform*

3.5 Brocade Vyatta 5400 vRouter (Virual Router)

The Brocade Vyatta 5400 vRouter is virtual machine based virtual router that delivers advanced routing for physical, virtual, and cloud networking environments. It includes dynamic routing, Policy-Based Routing (PBR), Multicast, stateful firewall, IPv6-compatible, IPsec and SSL-based Open VPN, DMVPN support, and traffic management in a solution optimized for virtualized environments. All features can be configured through a familiar, network-centric Command Line Interface (CLI), a Web-based GUI, or external management systems using the Brocade Vyatta Remote Access API. The Brocade Vyatta 5400 vRouter supports and can be deployed on all commonly used hypervisors such as VMware EXSI, Microsoft Hyper-V, Citrix Xenserver, Red Hat KVM etc. and can be installed on any standard x86-based system. It delivers a robust, Linux-based, and extensible OS.

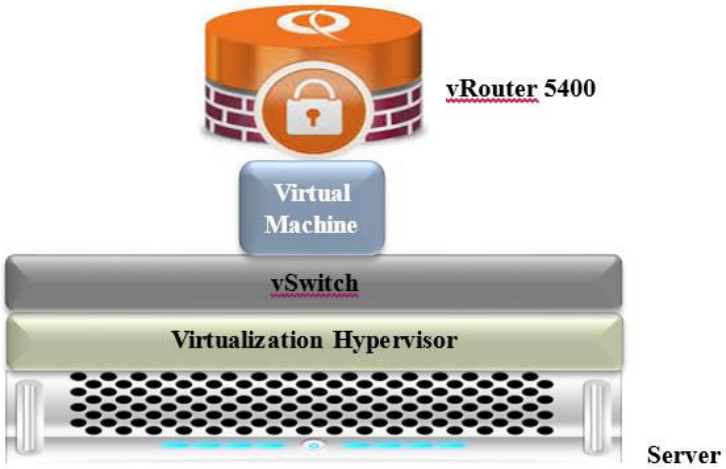


Figure 7: *Brocade Vyatta 5400 vRouter as a Virtual Router*

Key Features of Brocade Vyatta 5400 vRouter

Network Connectivity
With full support of IPv4 and IPv6 dynamic routing protocols (BGP Multipath, high performance BGP routing, Virtual route reflector, OSPF, RIP, Multicast) and PBR. This includes support for 802.11 wireless, serial WAN interfaces, and a wide variety of 10/100 Mbps through 10 Gbps Ethernet NICs.

Firewall Protection
The Brocade Vyatta vRouter firewall features IPv4/IPv6 stateful packet inspection to intercept and inspect network activity and protect critical data. Advanced firewall capabilities include zone- and time-based firewalling and P2P filtering.

Secure Connectivity
Organizations can establish secure site-to-site VPN tunnels with a standards-based IPsec VPN between two or more Brocade Vyatta vRouters or any IPsec VPN device. The vRouters also provide network access to remote users via SSL-based Open VPN functionality. In addition, they support

Dynamic Multipoint VPN (DMVPN) and the ability to represent policy-based IPsec tunnels as virtual interfaces (Virtual Tunnel Interface, or VTI).

High Availability
Mission-critical networks can deploy Brocade Vyatta vRouters with confidence, knowing that industry-standard failover and configuration synchronization mechanisms will provide high availability and system redundancy.

3.6 Arista vEOS (Virtual Switch)

Arista vEOS (Virtualized Extensible Operating System) is a virtual-machine-based virtual switch or switch-integrated implementation of Arista EOS that integrates with VMware vSphere to provide the network operator visibility and configuration access to the virtual switches within the virtual server environment. The Arsita vEOS is similar to the Extensible Operating System (EOS) of Arista Networks 7000 family switches.

Arista vEOS integrates with the VMware vNetwork Distributed Switch framework. vEOS combines management for all physical and virtual switching through a single network operating system image, enabling consistency of policy between physical and virtual assets. Arista simplifies operations for the server administrator while giving the network administrator visibility into and control over the existing virtual switches. vEOS brings consistent operational models and policy to the physical, virtual, and cloud network.

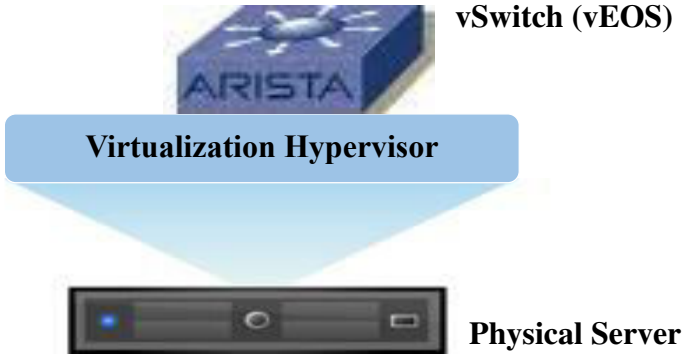


Figure 8: *Arista vEOS vSwitch as a Virtual Switch*

Arista vEOS Key Attributes

- Arista vEOS is an implementation of Arista EOS that manages VMware Distributed Switches.
- Extends a familiar industry-standard CLI to the vSwitches.
- Integrates with VMware vSphere virtualization platform.
- Separation of control and data plane enables hitless software upgrades.
- Auto discovers virtual infrastructure.

- Automates network provisioning during VM migration.
- Creates and manages distributed port profiles for physical-virtual service mapping.
- Supports extensible OVF framework for multi-vendor and cloud interoperability.
- Consistent software with EOS running across all Arista switches and now as a virtual appliance.

3.7 Juniper Firefly Suite (Virtual Firewall)

Firefly Suite provides security features to protect both inter-virtual machine (VM) traffic and traffic between VMs and external networks, including physical network and the Internet. It is designed to address the need for robust security for diverse Virtualized environments. Firefly suite consists of following products:

- **Firefly Host**

Firefly Host is basically hypervisor-based security solution that is purpose-built for the virtualized environment. It protects virtual machines (VMs) and their traffic in the virtualized data center.

- **Junos Space Virtual Director**

Junos Space Virtual Director provides rapidly provision and automatically deploy Firefly Perimeter instances into the VMware vCenter environment. After deploying Firefly Perimeter, it can be monitored through Virtual Director and also to efficiently manage their lifecycle.

- **Firefly Perimeter (Juniper vSRX Services Gateway virtual Firewall)**

Firefly Perimeter is a complete virtual firewall which is deployed as a virtual machine (VM) form on hypervisors. It is based on Junos OS and Juniper SRX Series Services Gateways. It protects virtualized network and the VM traffic at the tenant virtual network edge Firefly Perimeter is optimized to deliver strong security with both the performance and scale needed in a virtual world. Firefly perimeter can be deployed and configured through Junos OS J-Web and CLI.

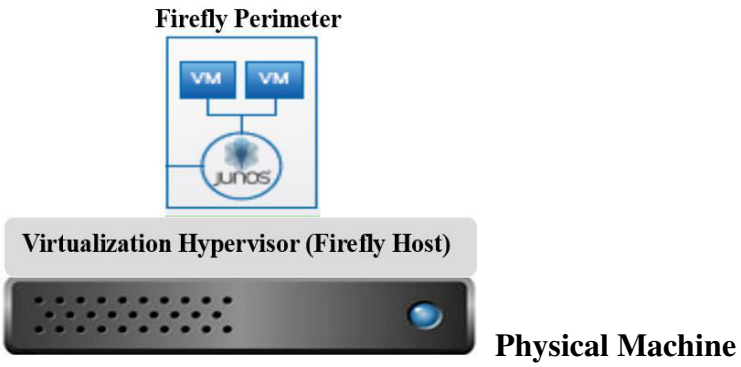


Figure 9: *Juniper Firefly Suite Components*

Key Benefits and Elements of Firefly Perimeter

Some of the benefits and key elements that Firefly Perimeter provides for virtualized environments are stated below:

- Routing and networking capabilities for virtualized environments.
- Stateful firewall protection at the tenant virtual network edge.
- Faster deployment of virtual firewalls than is possible with physical systems.
- Centralized and local management capability.
- Rich connectivity features based on a powerful Junos OS foundation, including NAT, and VPN
- Provisioning of security between zones, creating boundaries between organizations, public cloud, private cloud and applications

3.8 Cisco Cloud Services Router CSR1000v

Cisco Cloud services router CSR1000v is a fully virtualized software router that an enterprise or cloud provider deploy as a virtual machine in a provider hosted cloud. The CSR 1000V enables enterprises to transparently extend their WANs into external provider-hosted clouds and cloud providers to offer their tenants enterprise-class networking services. The CSR1000v takes advantage of Cisco ASR 1000 series routers design and cisco leading industry IOS networking and security features and capabilities. CSR1000v is a hardware agnostic so it is not dependent on any server implementation. It can run on VMware hypervisor, Citrix Xenserver and also support other most commonly used hypervisors

Features

- Routing
- VPN
- Firewall
- Qos
- Wan optimization
- High Availability
- Traffic redirection
- RESTful APIs.

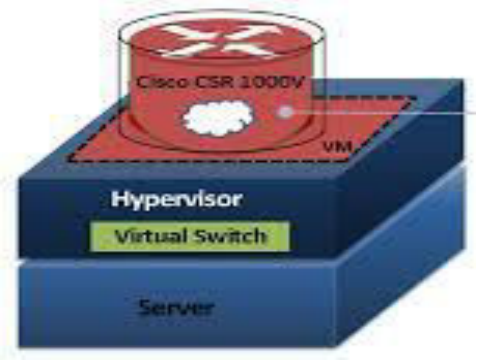


Figure 10: *Cisco CSR1000v as a Virtual Router*

3.9 Hardware Equipment Required for the Project

We will use the following hardware equipment for the physical routers and servers in our hybrid network topology.

- **Four- Cisco 2900 series Routers**
- **Two- Dell PowerEdge R420 Rack Servers**
- **Ethernet Cables with RJ-45 Connectors**

Major Specifications required for one Dell R420 rack Server are specified below:

- **Processor- Quad Core**
- **Operating System- Virtualization support of VMware ESXI**
- **Memory- 16GB RAM**
- **Storage- 1TB Hard drive**
- **Network Controller- Two 1GB Ethernet Ports**

4. NETWORK TOPOLOGY DESIGN CONSIDERATIONS

4.1 Multivendor Hybrid Network Topology for Cloud Deployment

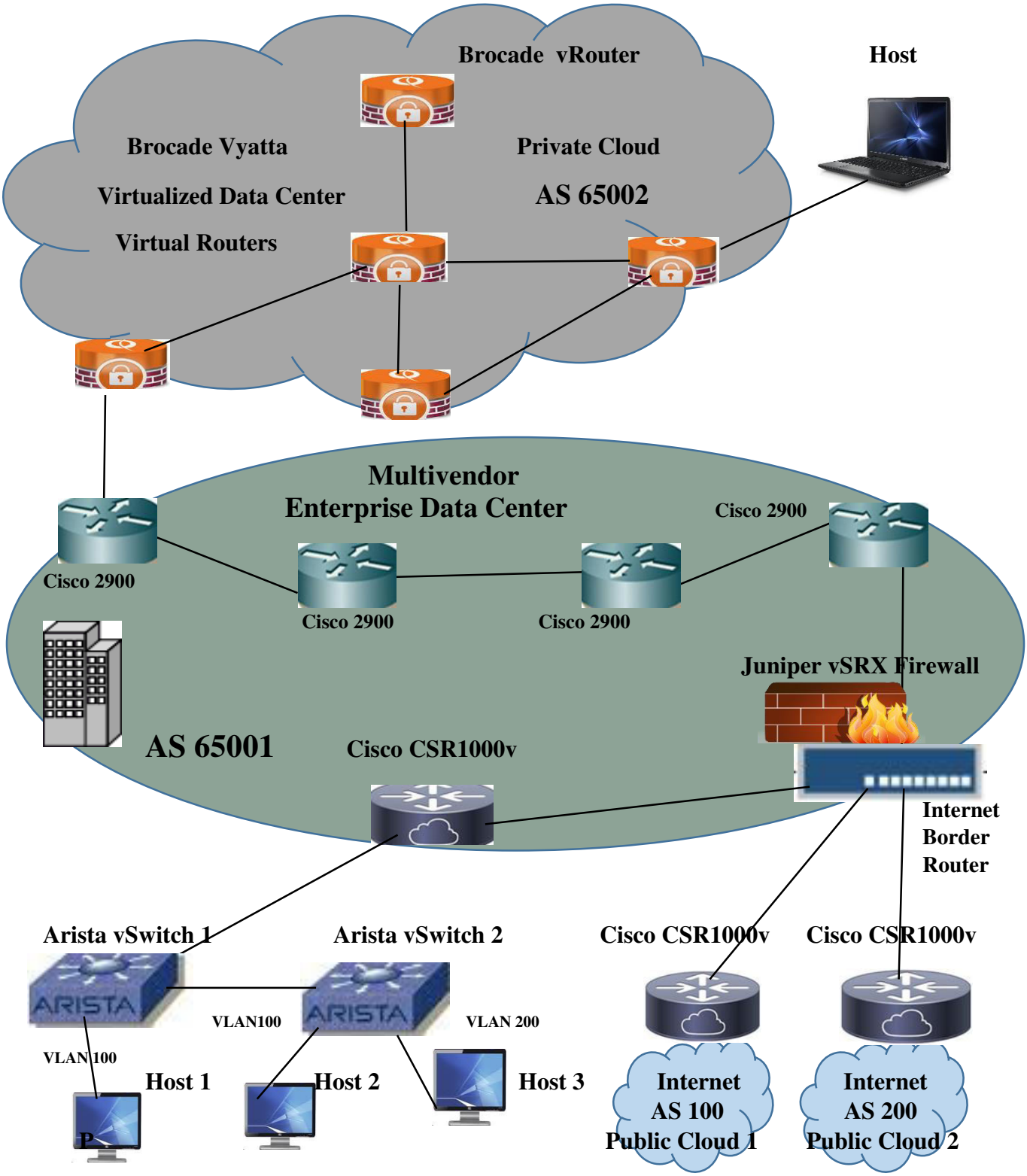


Figure 11: *Multivendor Hybrid Network Topology for Cloud Deployment*

4.2 Logical Hybrid Network Topology for Building Virtualized Data Center (Private Cloud)

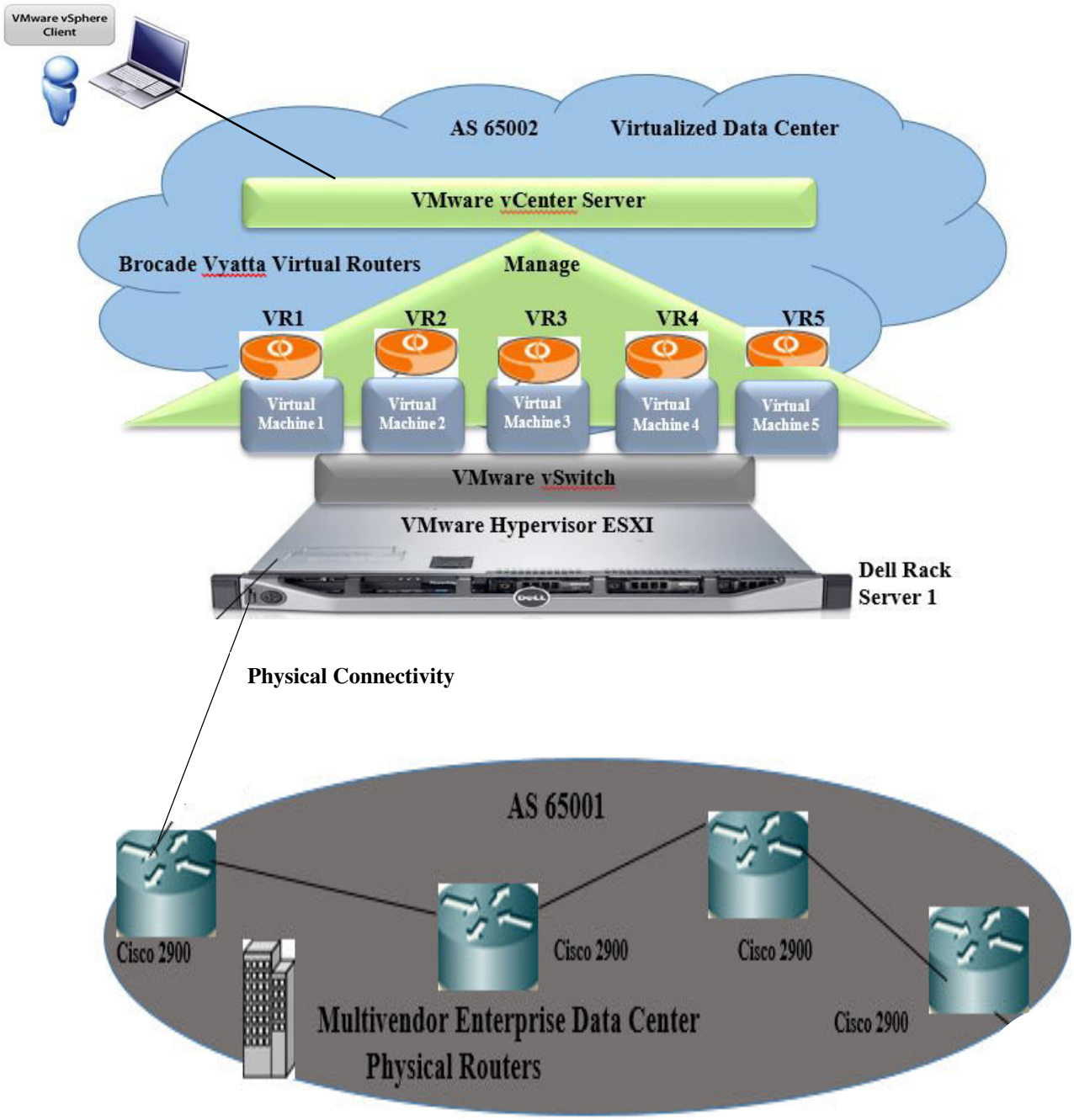


Figure 12: *Logical Hybrid Network Topology for Building Virtualized Data Center (Private Cloud)*

4.3 Logical Hybrid Network Topology for Deploying Public Cloud and Enterprise VLAN

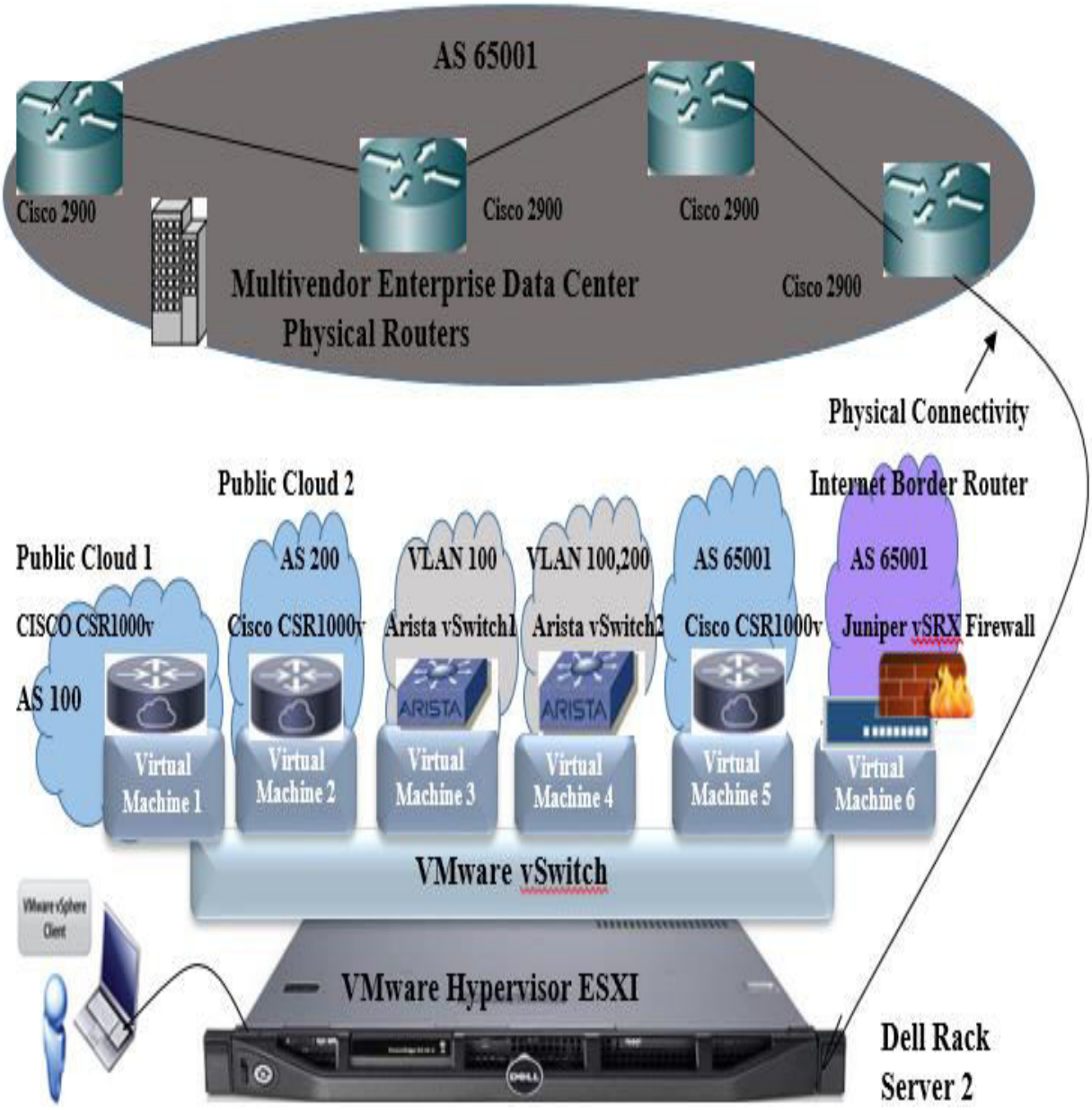


Figure 13: *Logical Hybrid Network Topology for Deploying Public Cloud and Enterprise VLAN*

4.4 Logical Network Topology for Building Hybrid Network

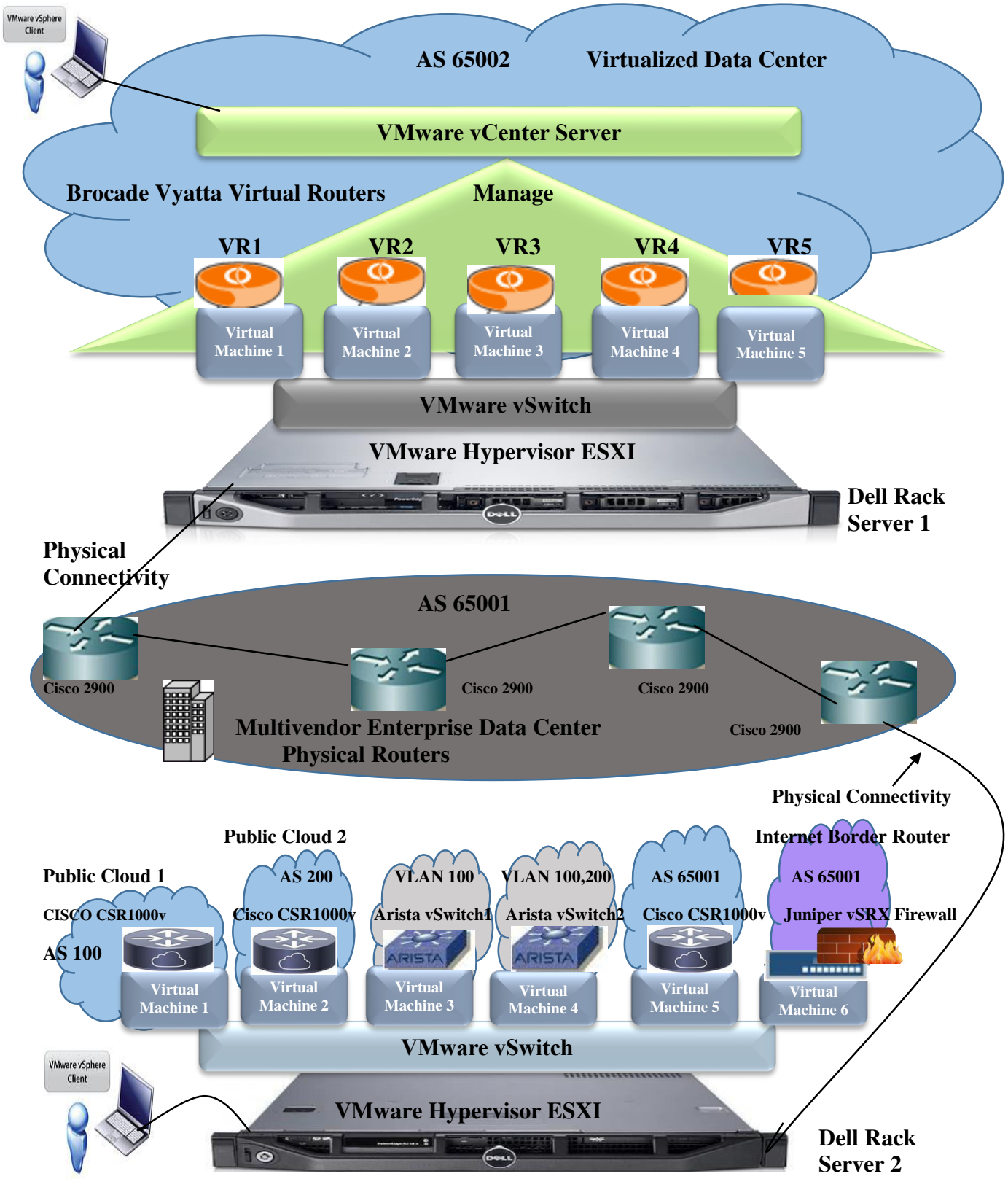


Figure 14: *Logical Network Topology for Building Hybrid Network*

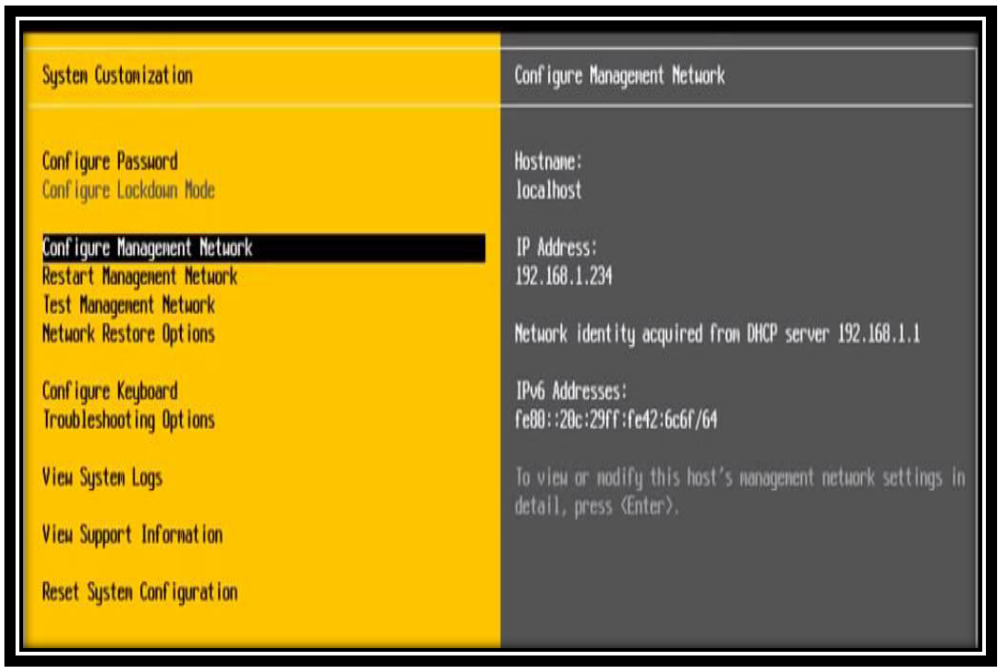
5. IMPLEMENTING VIRTUAL INFRASTRUCTURE

5.1 Installation and Configuring VMware ESXI 5.1 (Hypervisor)

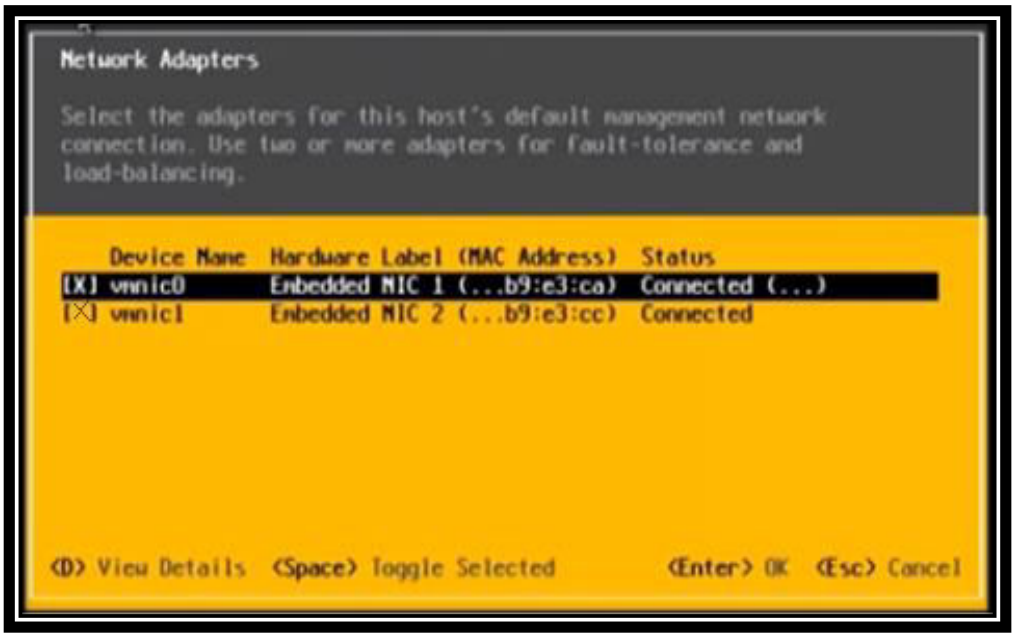
Step 1: First download the trial version of hypervisor VMware ESXI 5.1 disk image file from VMware website https://my.vmware.com/group/vmware/evalcenter?lp=default&p=free-esxi5 and install it on Dell PowerEdge R420 rack server.

Step 2: When the ESXI has booted up, we will customize the ESXI server by first logging in with the login name and password we set during ESXI installation i-e. Login name: root and Password: Mint709?

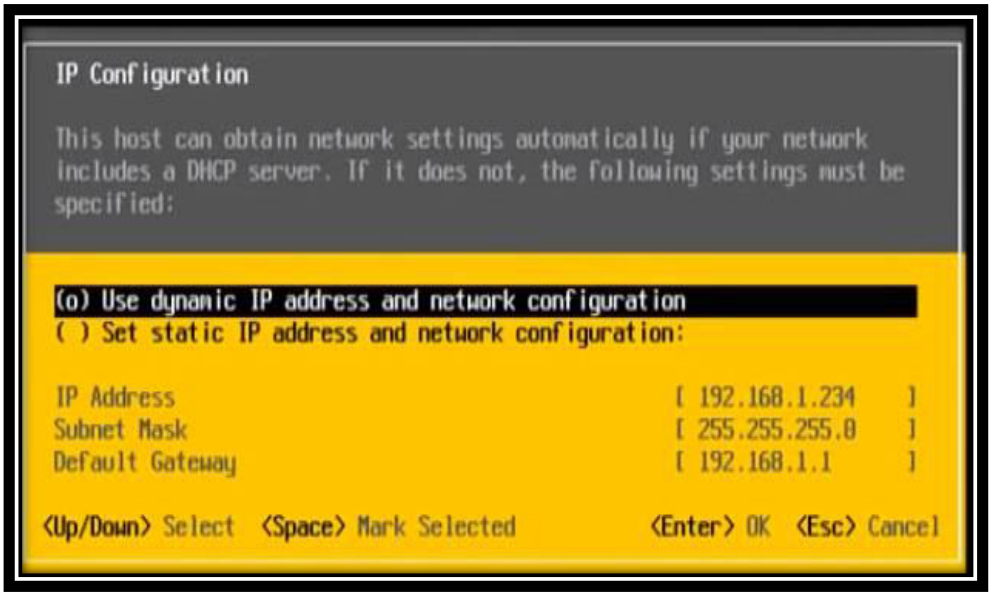
Step 3: We will select the option Configure Management Network and configure its parameters.



Step 4: When we select network adaptors, we can see that we have two network adaptors that can be used. Select both vmnic adaptors i-e. vmnic0 and vmnic1 because we will use vmnic0 for management and getting the IP from DHCP and vmnic1 will be used for networking and connecting the virtual world with the physical world.

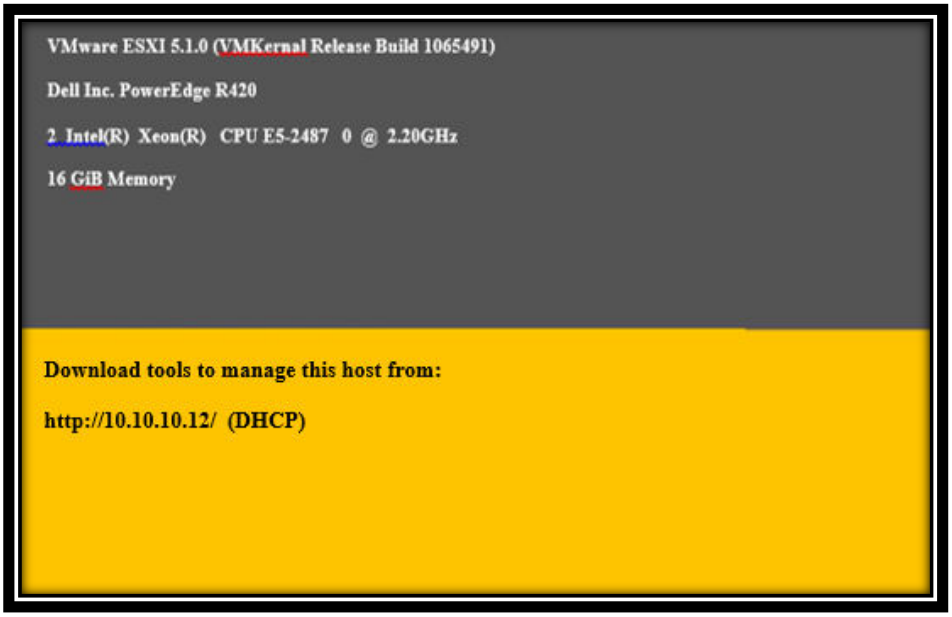


Step 5: Go to IP configuration and select the dynamic IP address option to get the IP from DHCP. Press enter to apply changes and restart management network for the changes to take effect.



Step 6: We have been assigned the IP address 10.3.31.38 from the DHCP server ensuring the successful installation and configuration of ESXI 5.1 on Dell rack server 1.

Repeat the similar steps from step 1 to 6 for installing and configuring the Dell R420 rack server 2. For the second server we have been assigned the IP address 10.10.10.12 from the DHCP server as shown below:



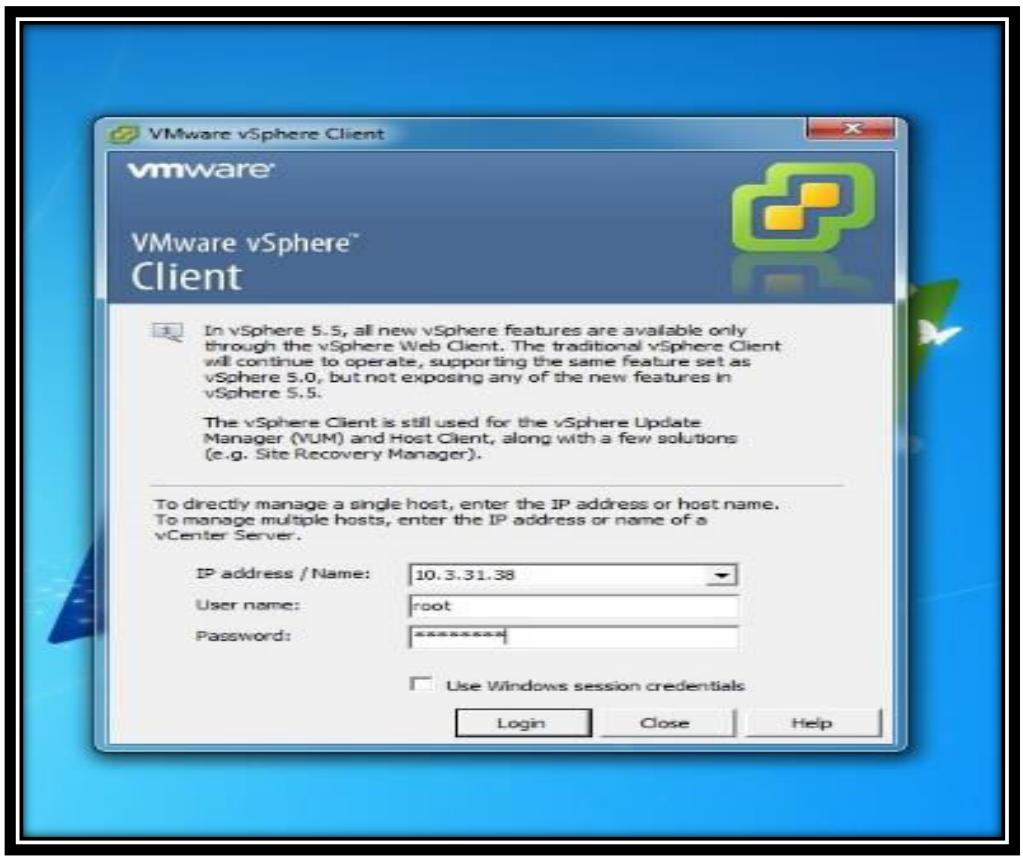
5.2 Accessing ESXI through VMware vSphere Client

Step 1:

Open the Internet browser and type the ESXI server IP i-e. http://10.3.31.38/ on the browser and it will direct towards the download components page from which we have to download VMware vsphere client

Step 2:

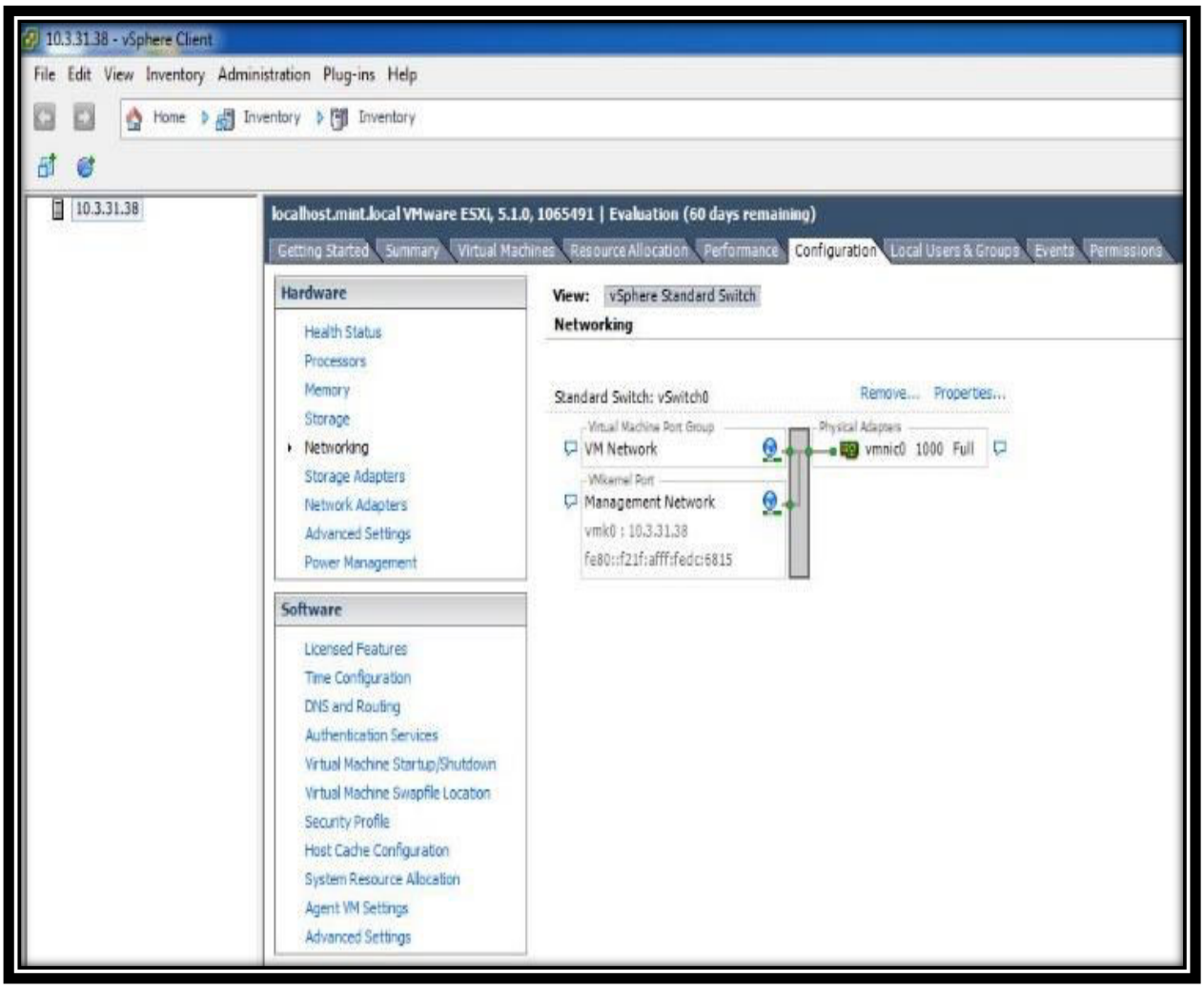
Run the downloaded VMware vsphere client setup and install it on your machine. Fo log in to the VMware ESXi server using vsphre client we will open the VMware vsphere client and it will ask for the User name, Password and IP address of the ESXI. Type the User name: root, Password: Mint709? and IP address the same we configured for ESXI server i-e. 10.3.31.38 and click login.



5.3 Creation of VMware Virtual Standard Switch

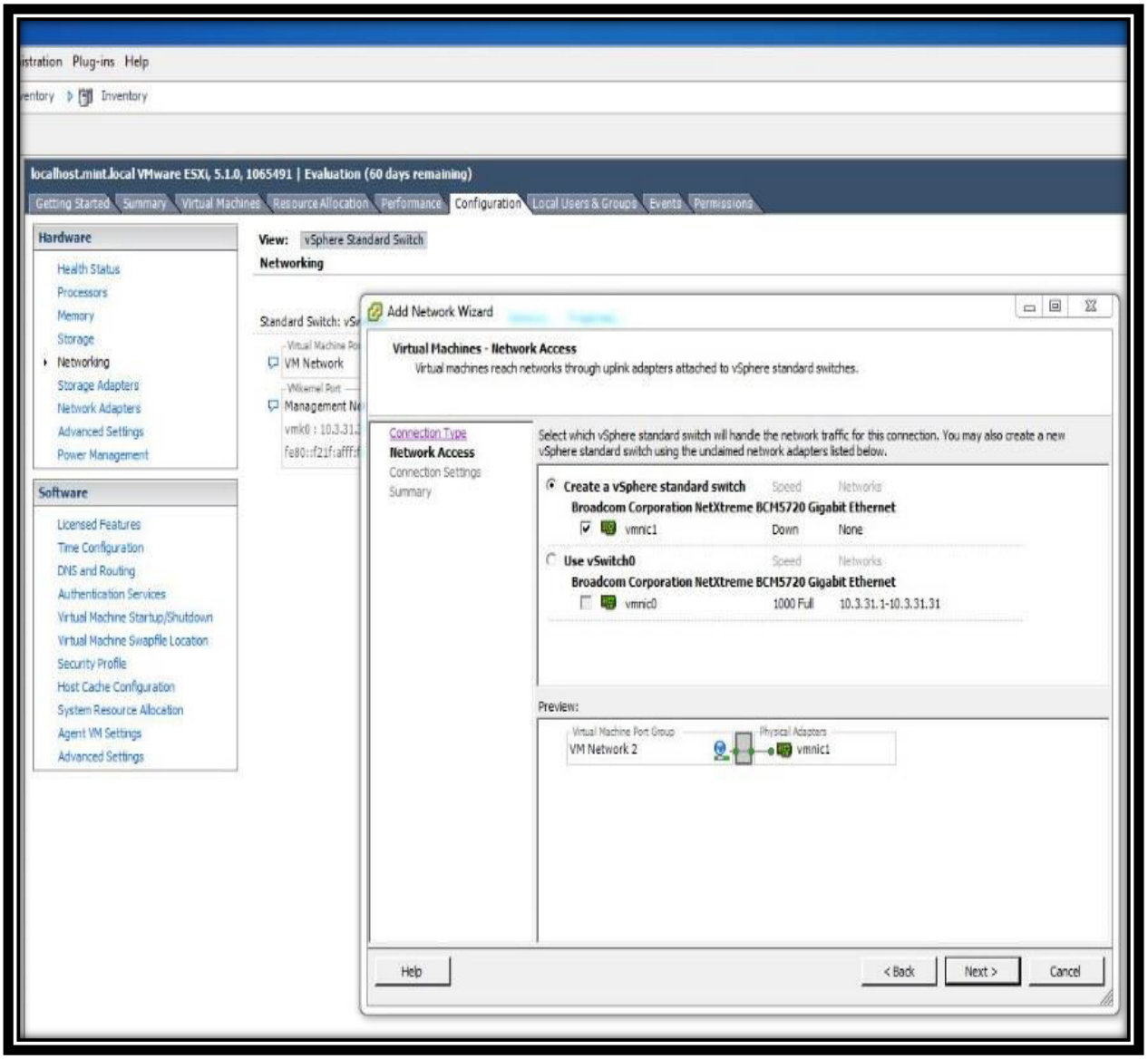
When we first time login to the vSphere client we will see that we already have virtual standard switch vswitch0 for management network as shown in the figure below which is basically a part of the installation of ESXI server. Currently this vSwitch has virtual machine port group, a management port interface as well as physical network adaptor identified as vmnic0 associated with this virtual switch vSwitch0.

Step 1: Login to the vSphere client and navigate under the hardware parameters to networking under the configuration tab and select the view as vSphere Standard Switch.



Step 2: Click add networking and select the connection type as virtual machine.

Step 3: Select the vmnic1 from the available physical adaptors and leave the virtual switch and vmnic0 i-e. a physical NIC attached to the virtual switch dedicated to the management of ESXI server as it is not a best practice to put the virtual mchines on this virtual switch.



Step 4: In the port group properties assign the network label name as virtual standard switch. Verify all the parameters and click finish.

As shown below a new virtual standard switch vSwitch1 has been created associated with vmnic1 and virtual machine port group called Virtual standard Switch. The status of vmnic1 is showing down because we are not connected to any physical device to that port yet. When we connect any physical device to that port it will go up showing up with green dot.

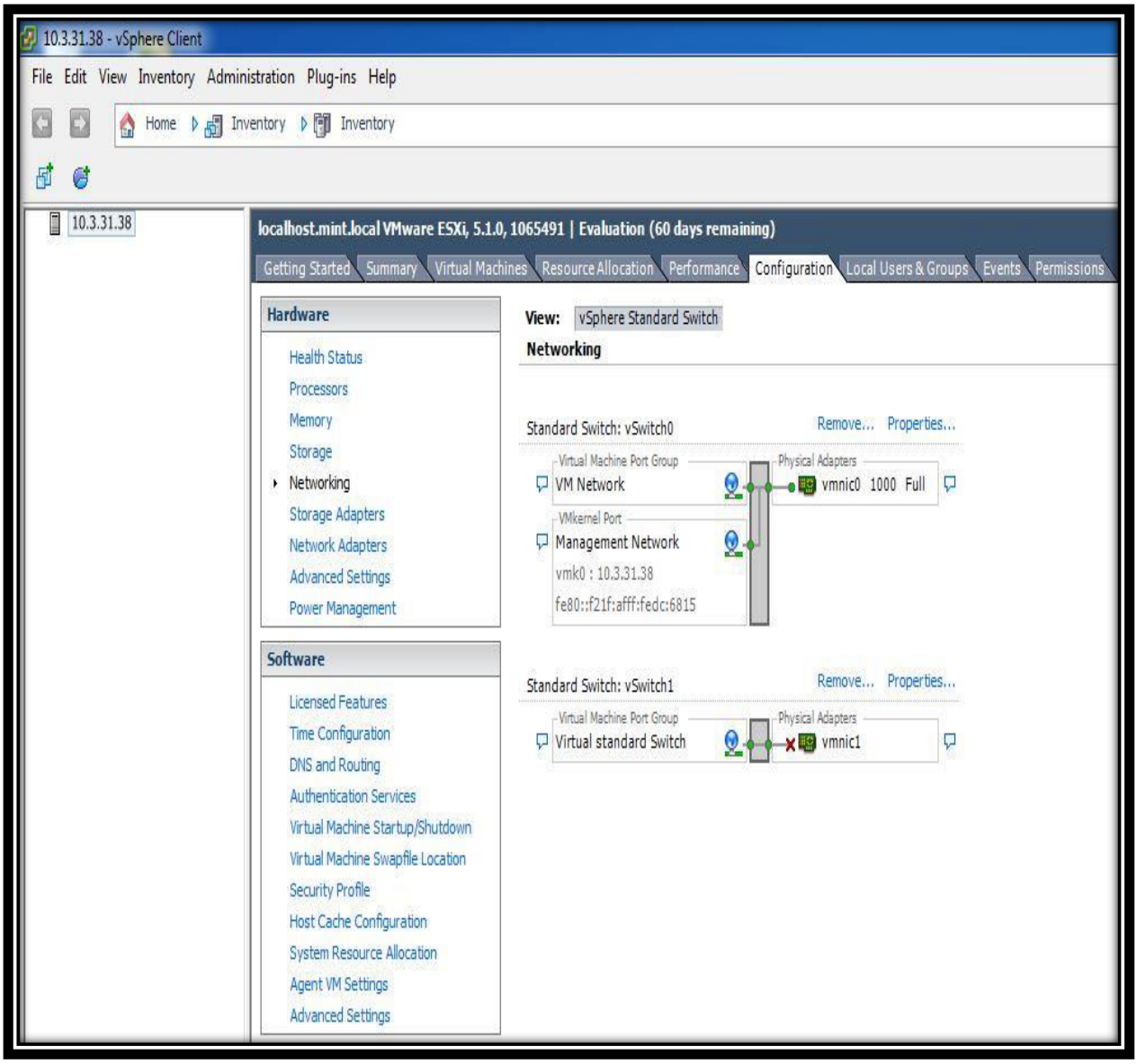


Figure15: *View of Created Virtual Standard Switch*

In the same fashion we will create virtual standard switch on ESXI server2.

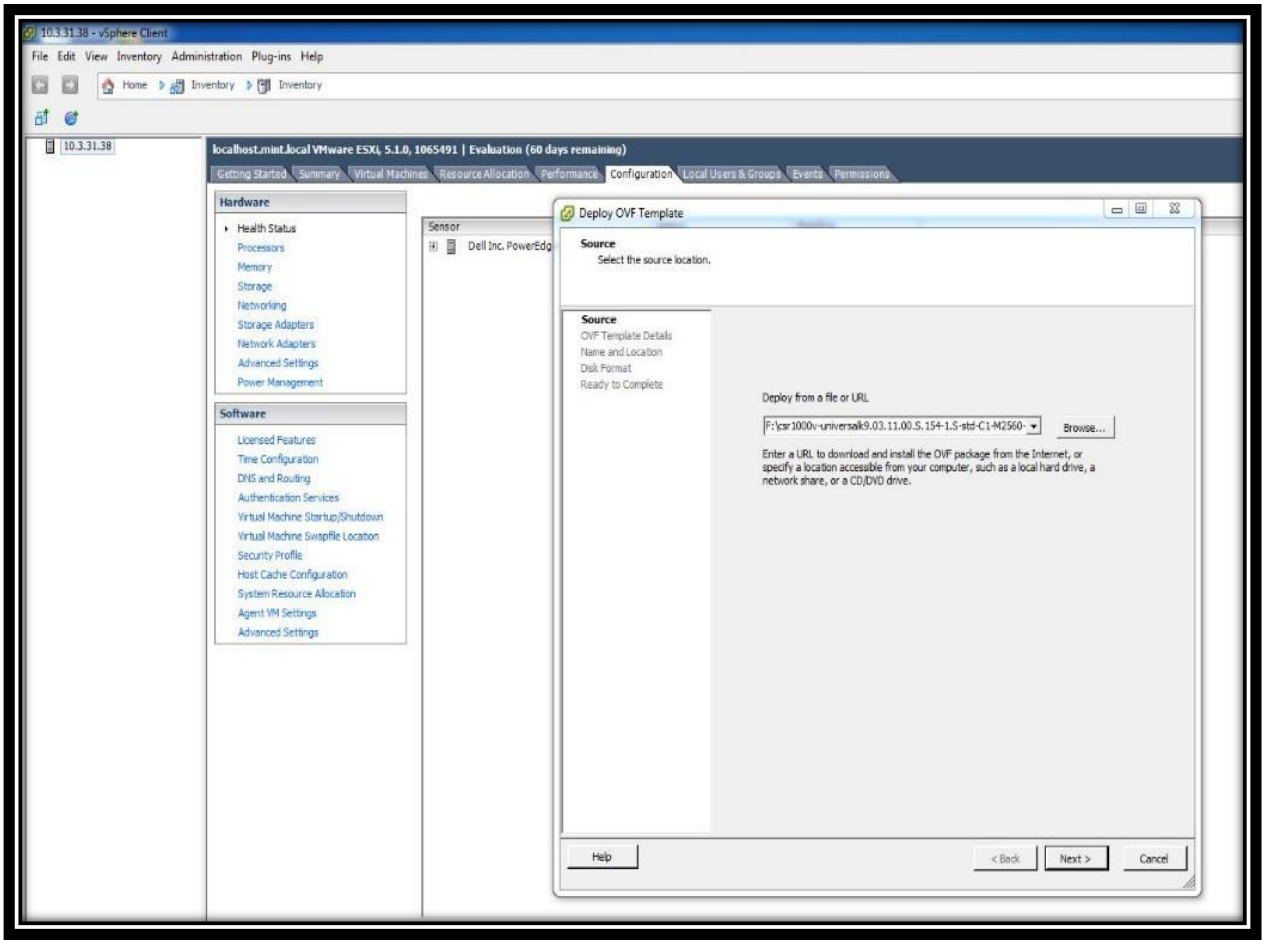
5.4 Creation and Deployment of Virtual Machine Cisco CSR1000v

Step1: Download the trial version of Cloud Services Router Cisco CSR1000v in the Open Virtualization Appliance (OVA) format file from the Cisco website given below:

http://software.cisco.com/download/release.html?mdfid=284364978&softwareid=282046477&release=3.11.1S&flowid=39582

Step 2: In the vSphere client right Click on the file menu and select deploy OVF template.

Step 3: Click browse button to locate the OVA file which we downloaded initially and click next.

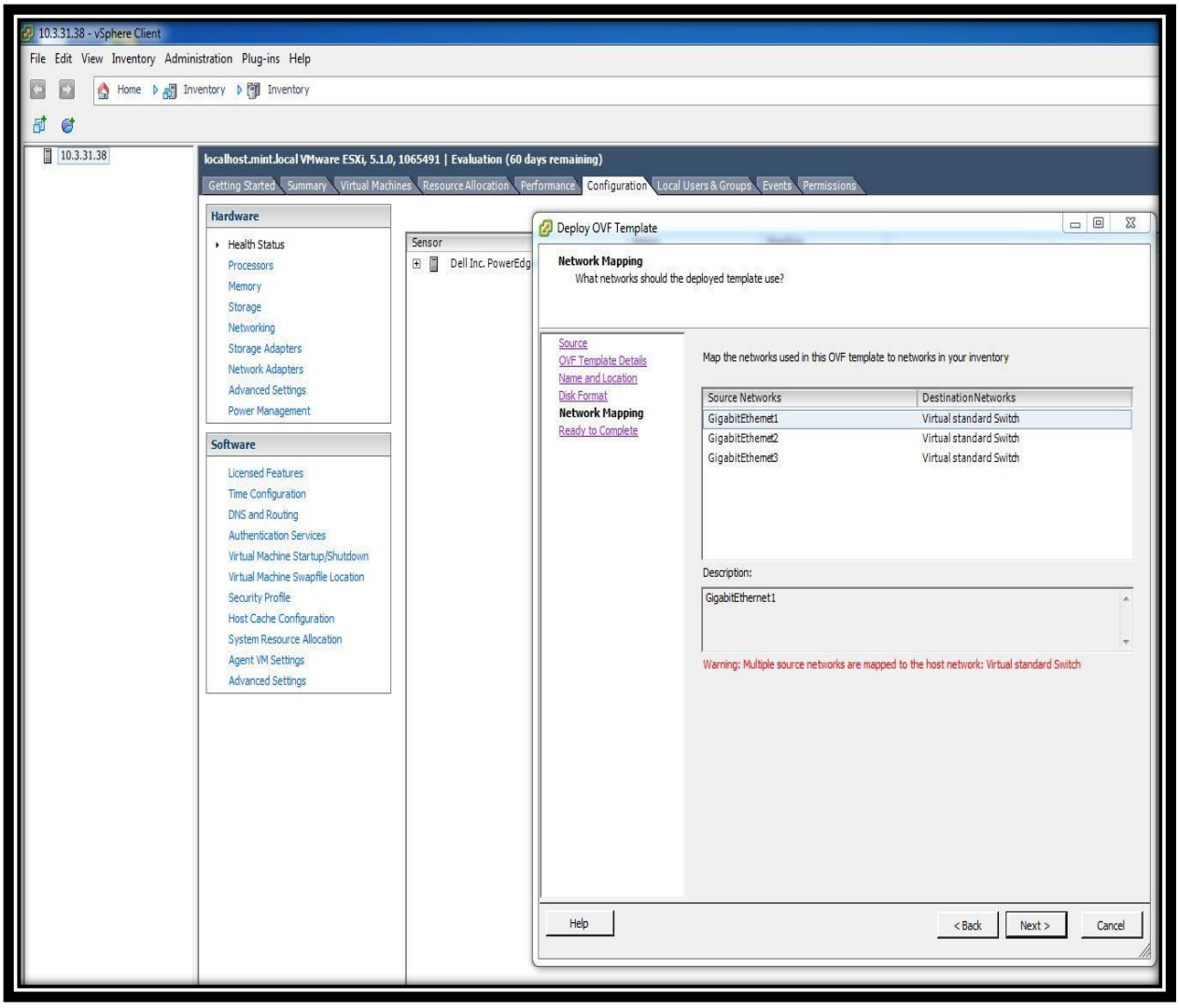


Step 4: Name the virtual machine as Cisco Cloud Services Router

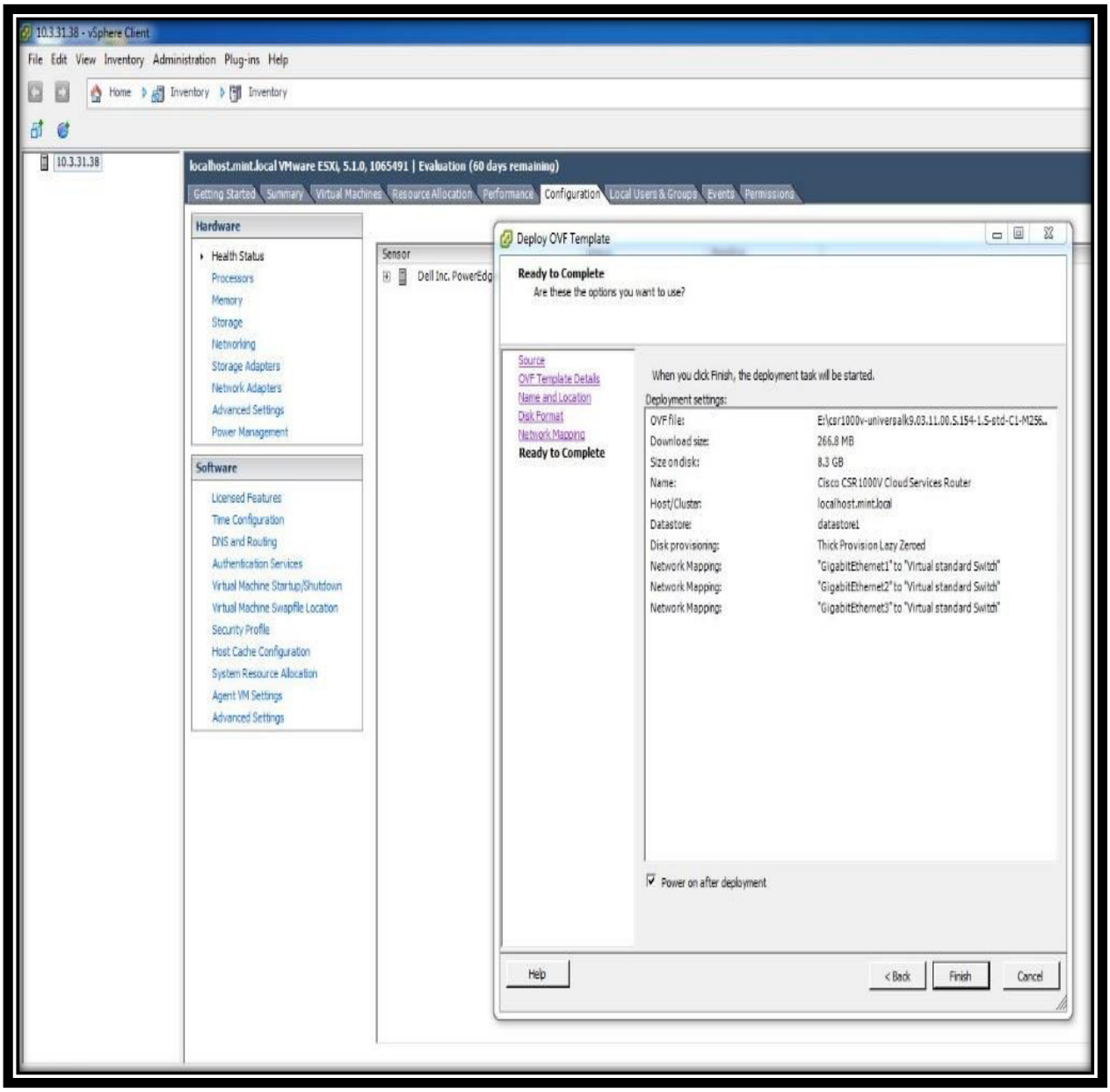
Step 5: For the storage specifications click the datastore1 which is already created in ESXI.

Step 6: For the disk format select Thick Provision Lazy Zero.

Step 7: In the network mapping select the virtual standard switch as their destination networks for all Gigabit Ethernet interfaces. Here, virtual standard switch is a port group which we created during the creation of vSwitch1. Meaning that, we are mapping our virtual machines to vSwitch1which is a desired method and leaving vSwitch0 for management network.



Step 8: Verify and confirm all the deployment settings and click next for the virtual machine creation.



A new virtual machine (virtual router) Cisco Cloud Services Router has been created as displayed under the ESXI server and also at the bottom showing the status of deploy OVF template as completed.

Step 9: Now, select the newly created VM and click power on button to start the VM.

Step 10: After that, click on the console button to access the console of deployed Cisco Cloud Services Router 1000v. Once it has booted successfully we can start the configuration task.

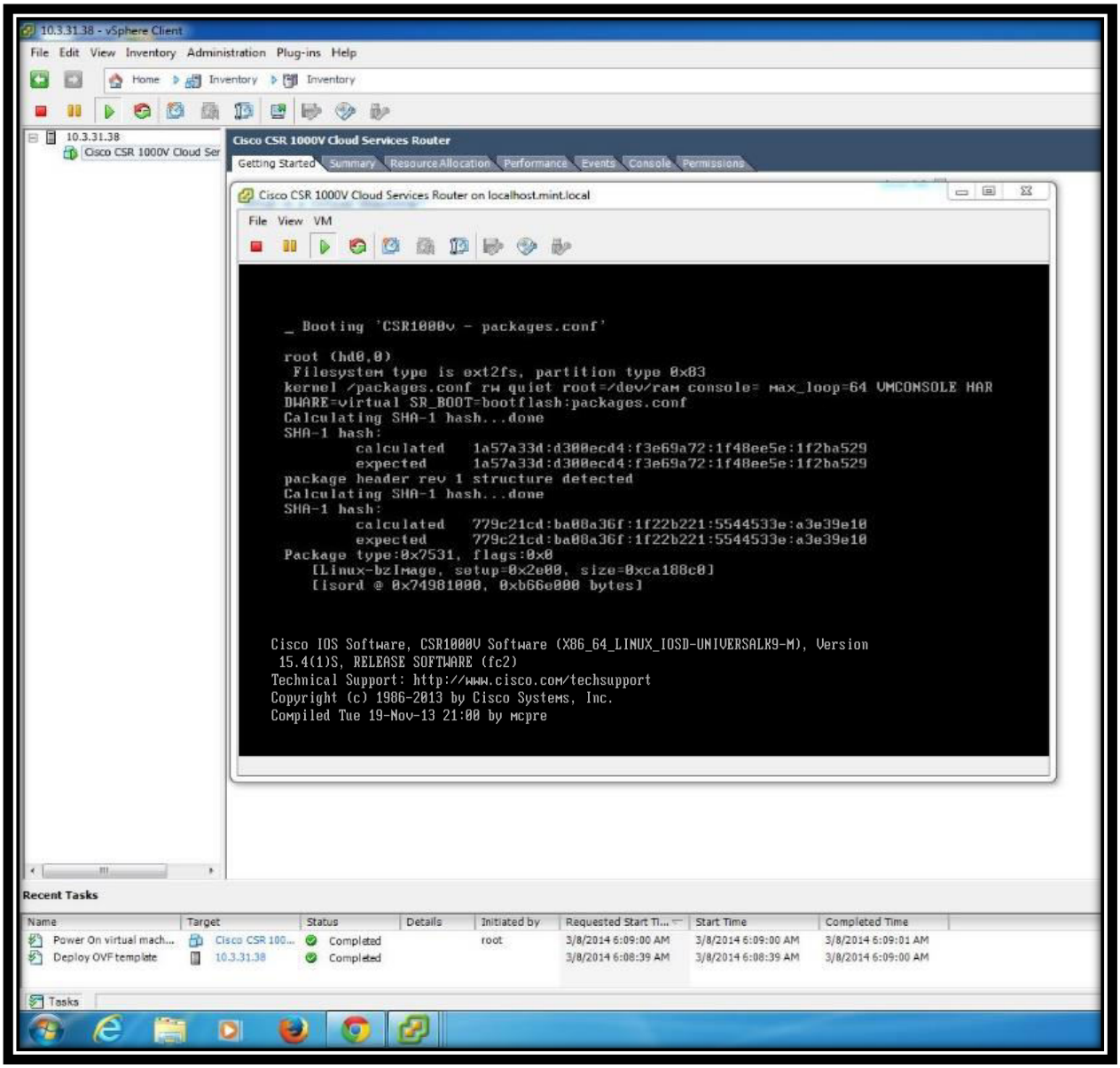


Figure16: *View of Deployed Cisco CSR1000v Virtual Machine*

By following above steps from 1 to 10, we will create and deploy two more Cisco Cloud Services Routers on this ESXI server.

5.5 Deployment of Juniper vSRX Services Gateway Virtual Firewall

Initially, download the trial version of firefly perimeter in the Open Virtualization Format (OVA) file which is basically a Juniper JUNOS based virtual firewall/router from the Juniper website given as:

http://www.juniper.net/support/downloads/?p=junosvfirefly-eval#sw

The creation and deployment steps of Juniper vSRX Services Gateway virtual appliance are the same as we have deployed for Cisco CSR1000v. After going through steps from 1 to 10, we have deployed a new virtual machine named Juniper vSRX router and it's running successfully as shown in the figure below:

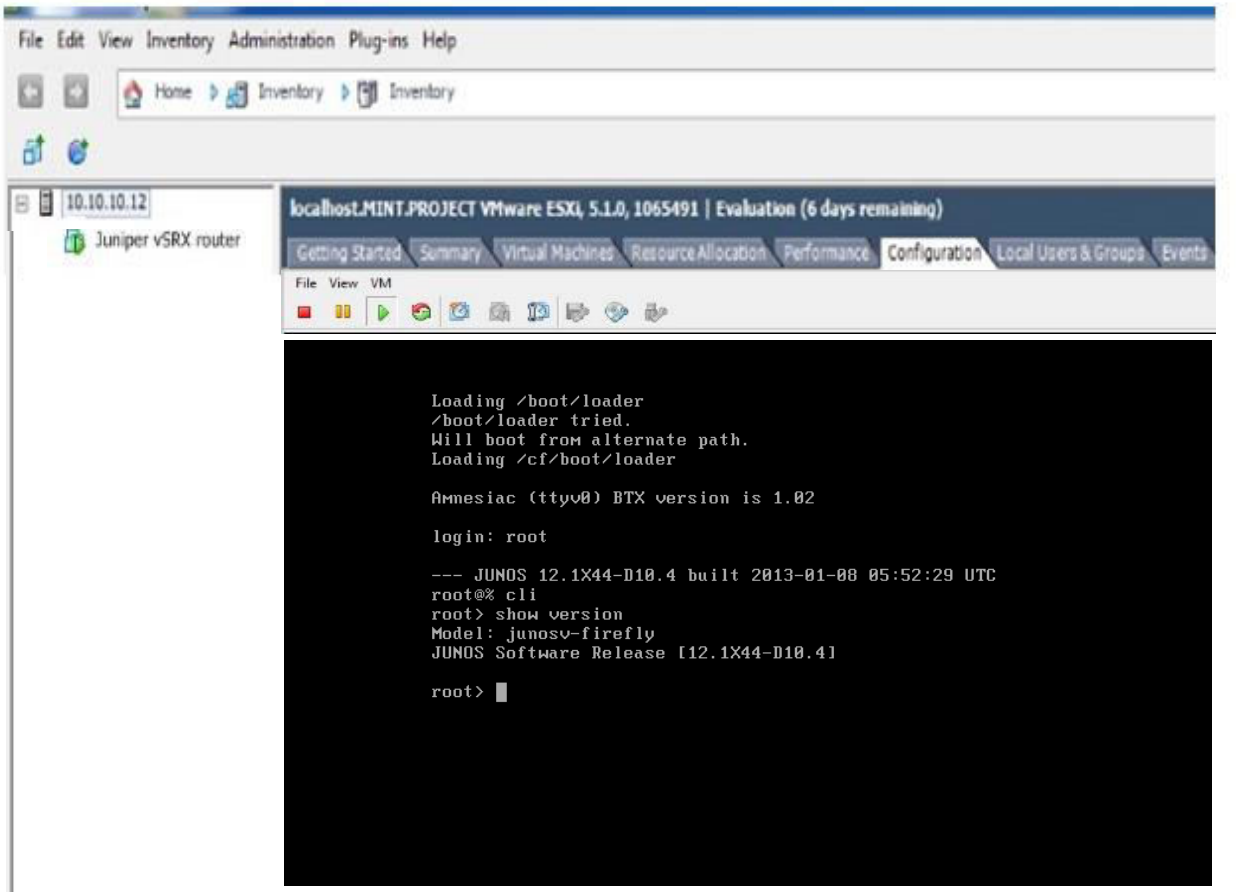


Figure17: *View of Deployed Juniper vSRX Virtual Machine*

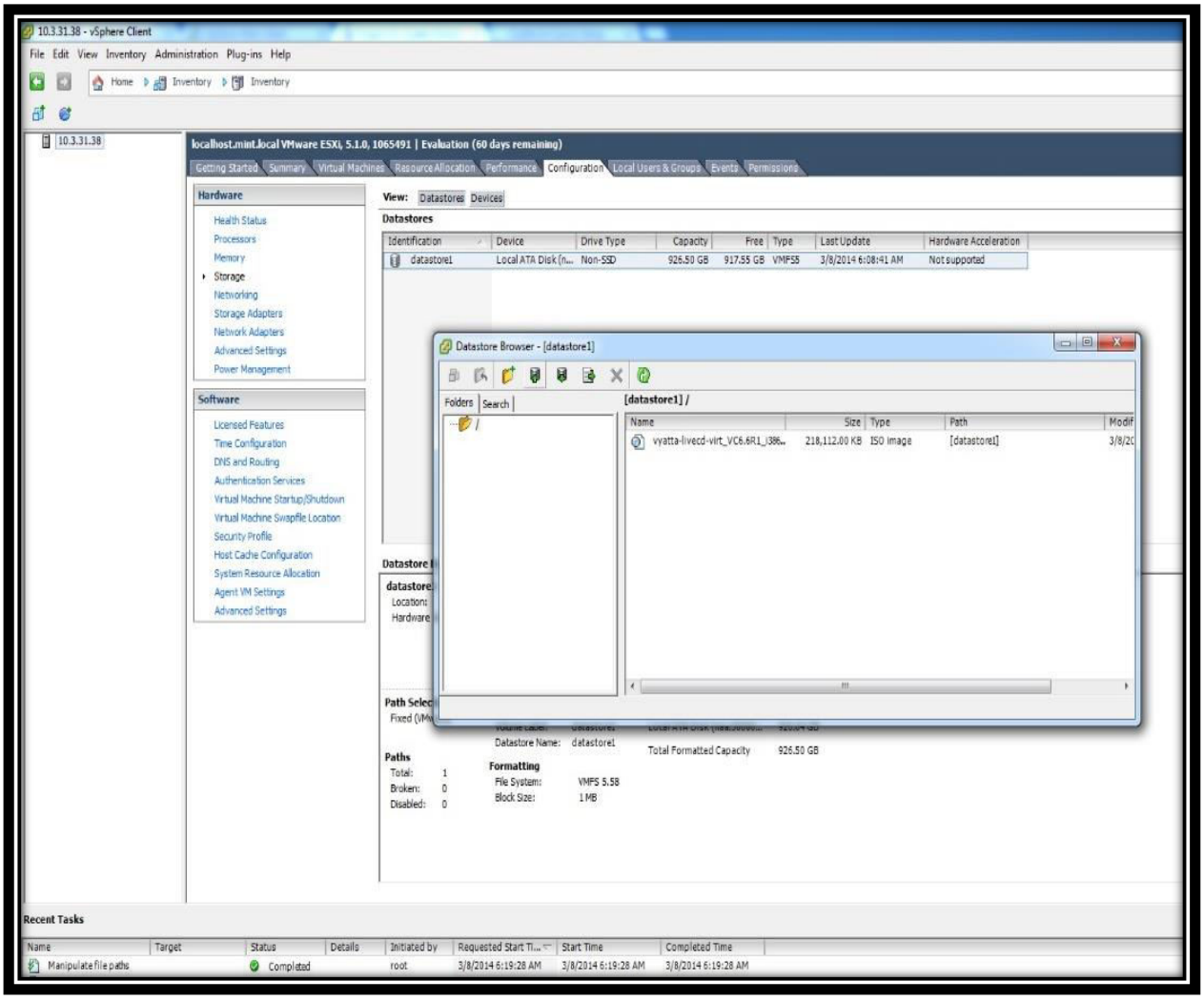
5.6 Creation and Deployment of Brocade Vyatta 5400 vRouter

The virtual machine creation and deployment method of Brocade Vyatta router is bit different from Cisco CSR1000v.

Step 1: First of all, download the trial version of Brocade Vyatta 5400 vRouter ISO file from the Brocade official website given below

http://www.brocade.com/forms/jsp/vyatta=download/index.jsp?src=WS&lsd=Banner&lst=BRCD&cn=SDN-GDG-14Q1-EVAL-WS-Vyatta-Download&gcn=&ggeo=

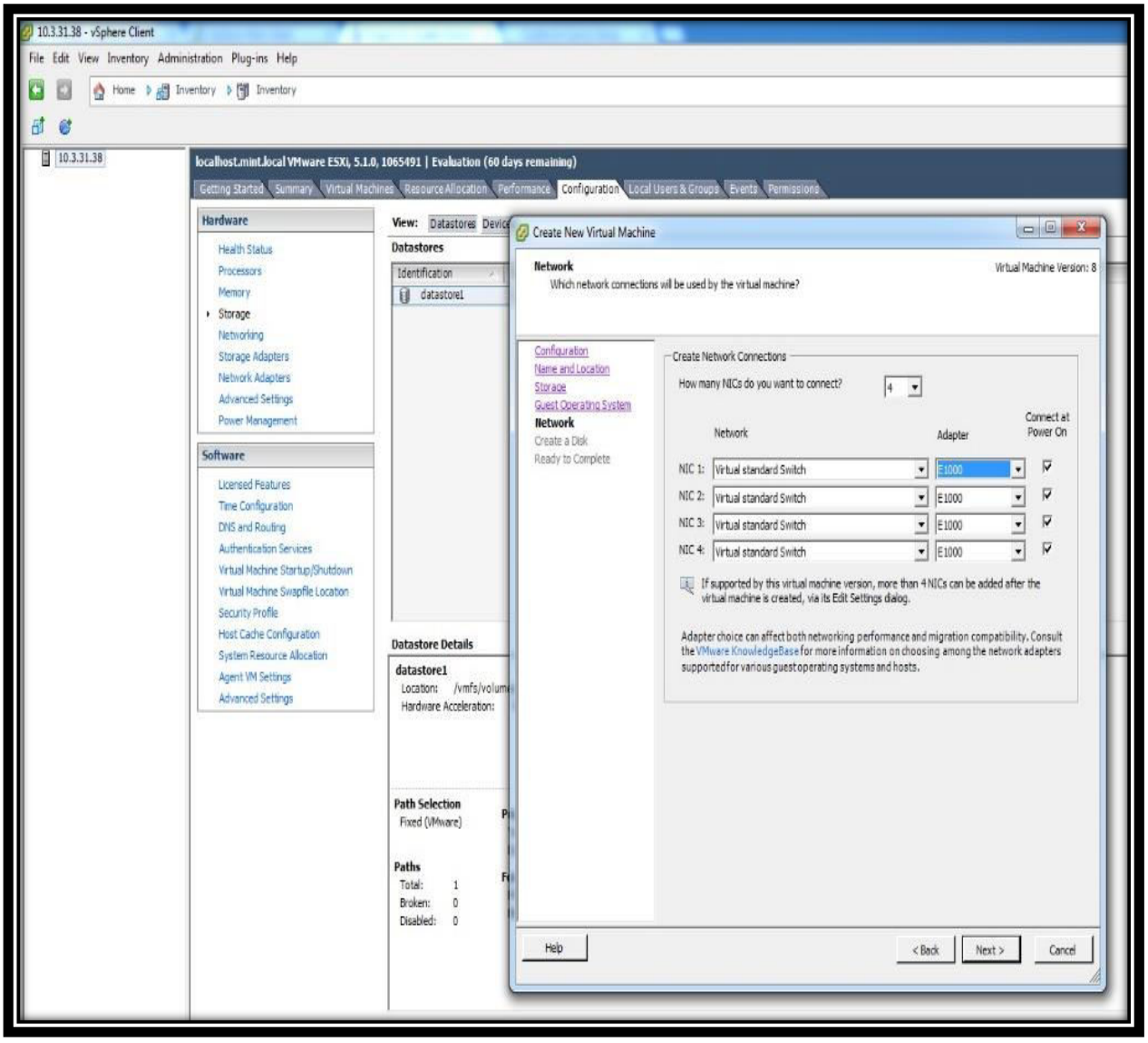
Step 2: Upload the downloaded Vyatta IOS file under the vSphere client datastore1by clicking the upload files tab and browse the downloaded file of Vyatta ISO image.



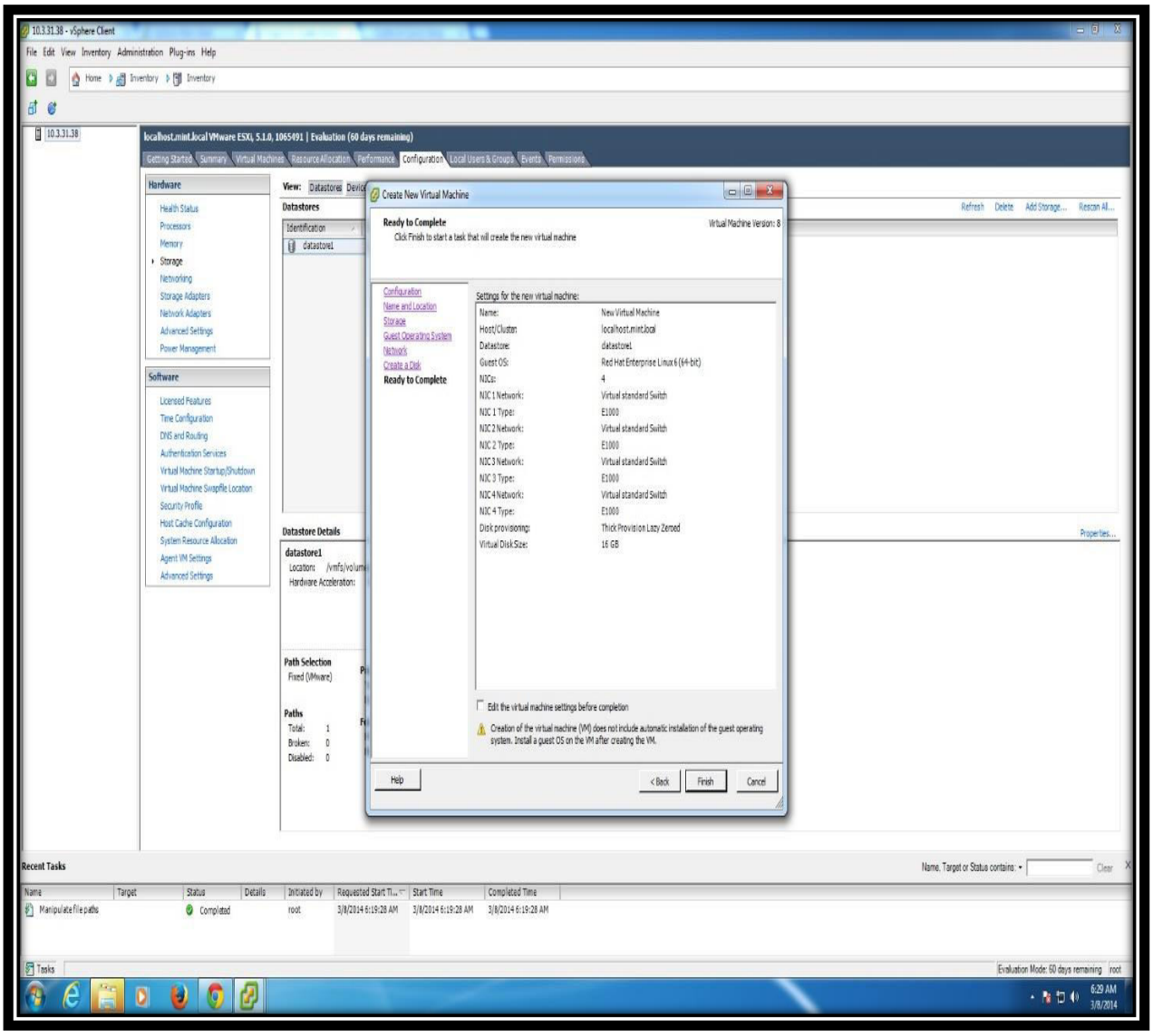
Step 3: Within the vSphere client click on create a new virtual machine and select typical settings under the configuration tab.

Step 4: Select the datastore1 for storage and choose the operating system as Linux 6 (64 bit) under the guest operating system tab.

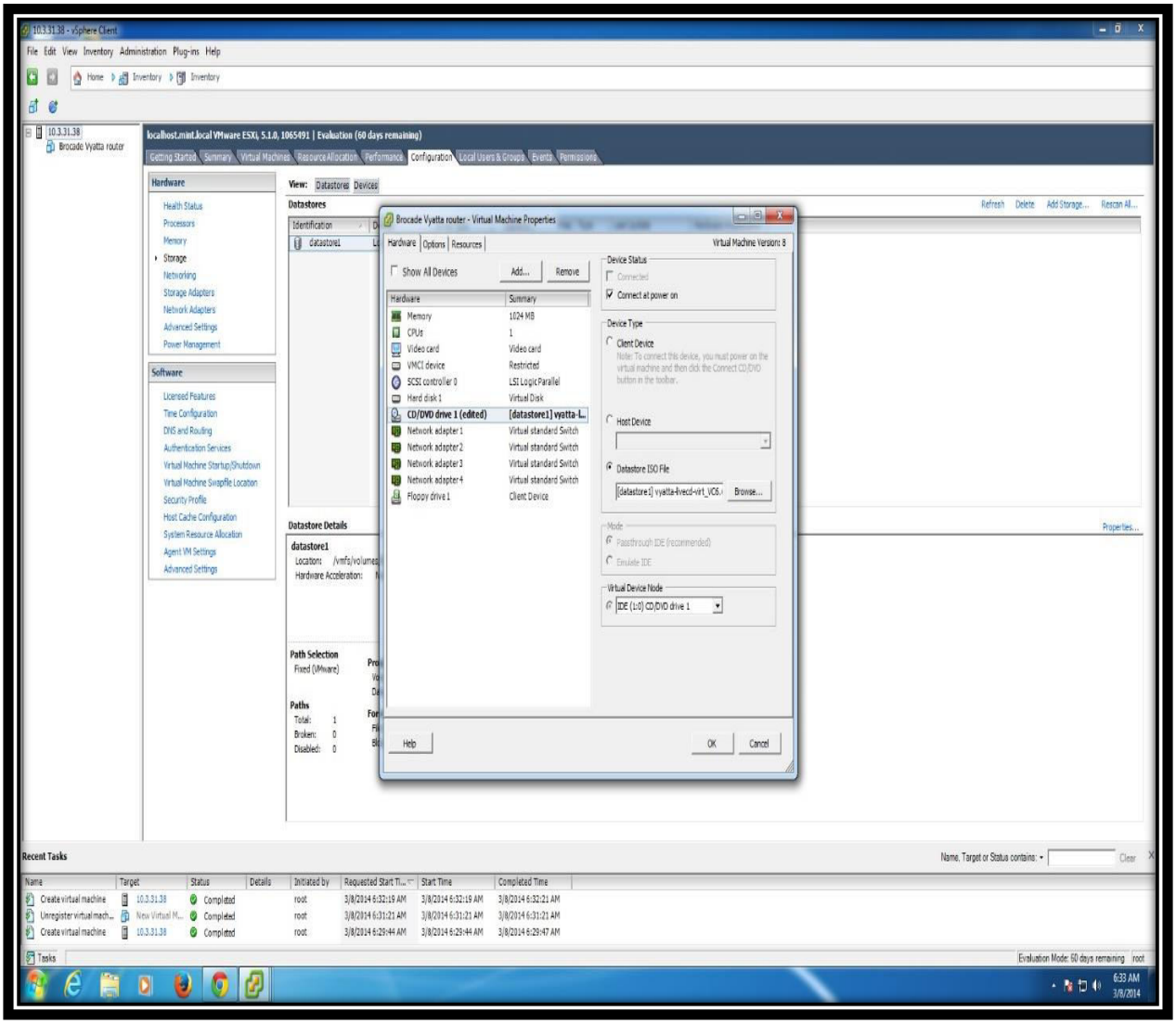
Step 5: Under the network section, increase the number of NICs to 4 and choose E1000 as the Adapter for all of them. Connect each NIC to a port-group of vSwitch1 i-e. virtual standard switch and click next.



Step 6: Verify all the settings for the new virtual machine and click finish to start a task that will create the new virtual machine of Brocade Vyatta router.



Step 7: Although we have created the new VM for Brocade Vyatta router but we still have to upload that ISO image to run the VM. Right click on the newly created VM of Brocade Vyatta router and select virtual machine settings. Map the CD/DVD drive1 to the Vyatta ISO bootloader image residing on the datastore1 file which we already uploaded in datastore1. This file needs to remain mounted and it is not only used for installation, but for all future boot cycles. Finally, Check the connect at power on option so that when we power on the VM it will run that ISO image. Click Ok for the changes to take effect.



Step 8: Power up the Brocade Vyatta vrouter VM and open the console. It will take around one minute for Vyatta router to complete the first boot. Once the boot process is complete we can login with the username "vyatta" and password "vyatta" to get the full benefits of Brocade Vyatta CLI.

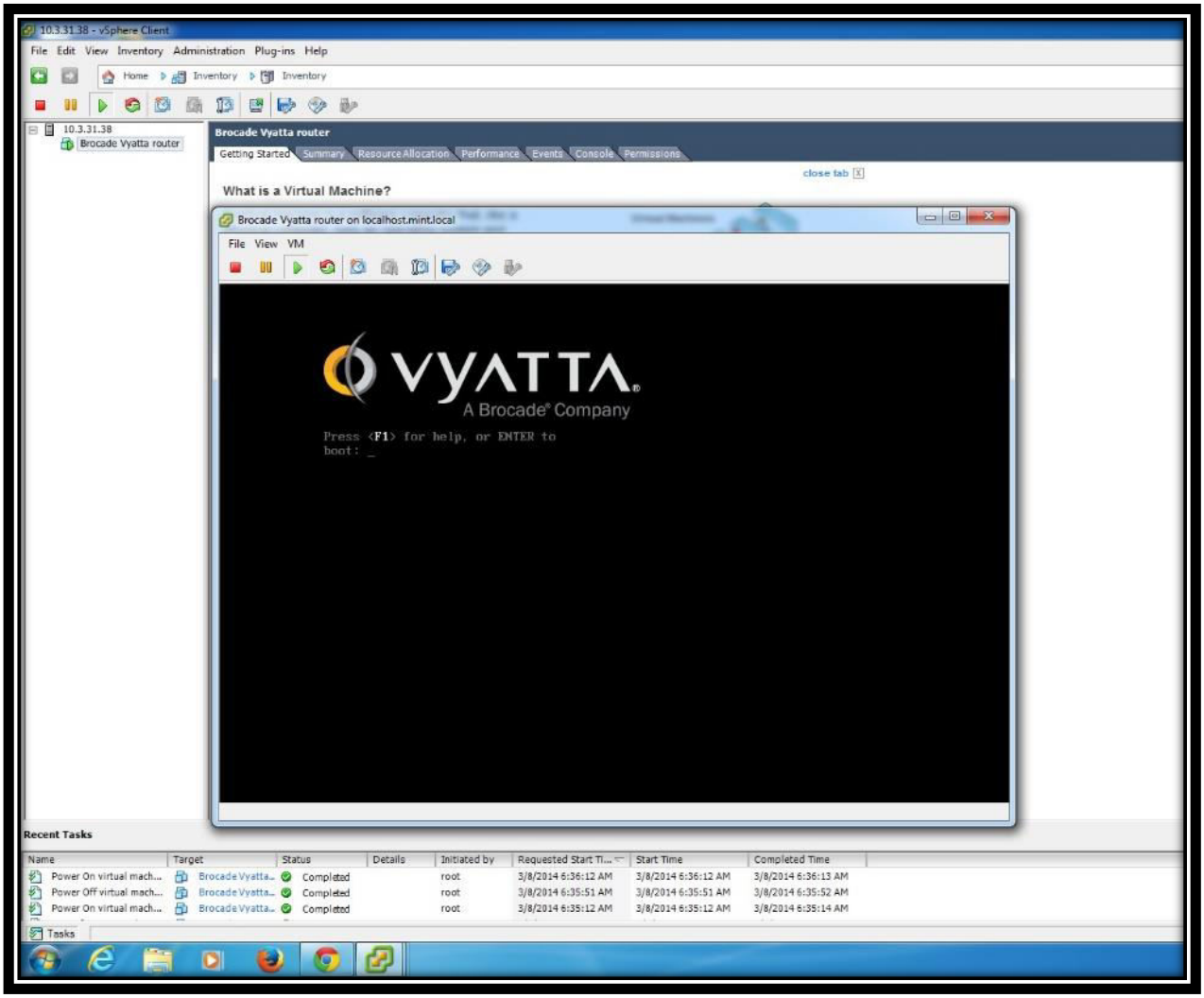


Figure18: *View of Deployed Brocade Vyatta 5400 vRouter Virtual Machine*

Follow the above steps from 1 to 8 to create and deploy four more Brocade Vyatta virtual routers on ESXI server1.

5.7 Arista vSwitch vEOS (virtual Extensible Operating System)

The creation and deployment method for Arista vSwitch vEOS is same as we have deployed for Brocade Vyatta vRouter. The only difference is that for Arista vSwitch we have to download two files as given below:

1. Bootloader: Aboot-veos-2.0.8.iso
2. Actual vEOS image as a VMDK: EOS-4.12.5-veos.vmdk

Trial version of these files can be downloaded from Arista Networks website mentioned below: http://www.aristanetworks.com/en/support/gettingstarted

For Arista vSwitch we have to upload above both files in the datastore1 instead of one we uploaded for Brocade Vyatta. After that, for Arista vSwitch creation follow the same steps from 1 to 10 as we followed for Brocade vRouter implementation.

Once the Arista vSwitch has been created we still will not be able to see all the network traffic traversing the virtual switch. To solve this problem, we have to go in vSwitch1 properties and accept promiscuous mode at the port group level which is rejected by default. After enabling the promiscuous mode all the interfaces and virtual machines within that port group i-e. virtual standard switch in our case will be able to see all the traffic passing on the vswitch1.

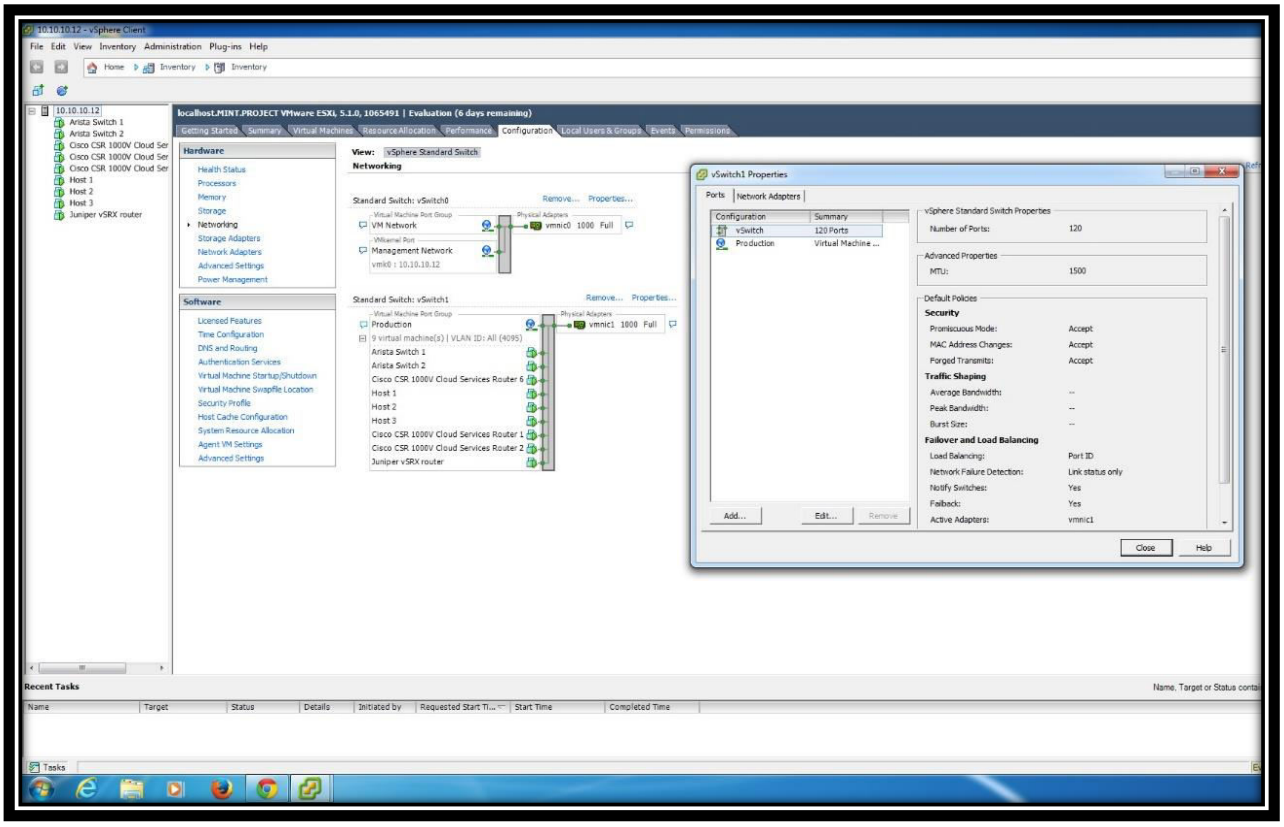


Figure19: *View of Deployed Arista vEOS vSwitch Virtual Machine*

5.8 Networking view for ESXI Server2

We have successfully deployed the following virtual machines and created vSwitch on ESXI server 2.

- **Virtual Standard Switch (vSwitch1)**
- **Arista Switch 1**
- **Arista Switch 2**
- **Cisco CSR 1000v Cloud Services Router 1**
- **Cisco CSR 1000v Cloud Services Router 2**
- **Cisco CSR 1000v Cloud Services Router 6**
- **Juniper vSRX Router**
- **Host 1**
- **Host 2**
- **Host 3**

All the above deployed virtual machines are up and running as shown in the following figure by green dot and all virtual machines are connected to the same port group named Production of vSwitch1 and that vSwitch1 is connected to the vmnic1 which is uplink a physical Ethernet adaptor. While vmnic0 is for the management network to manage the ESXI server.

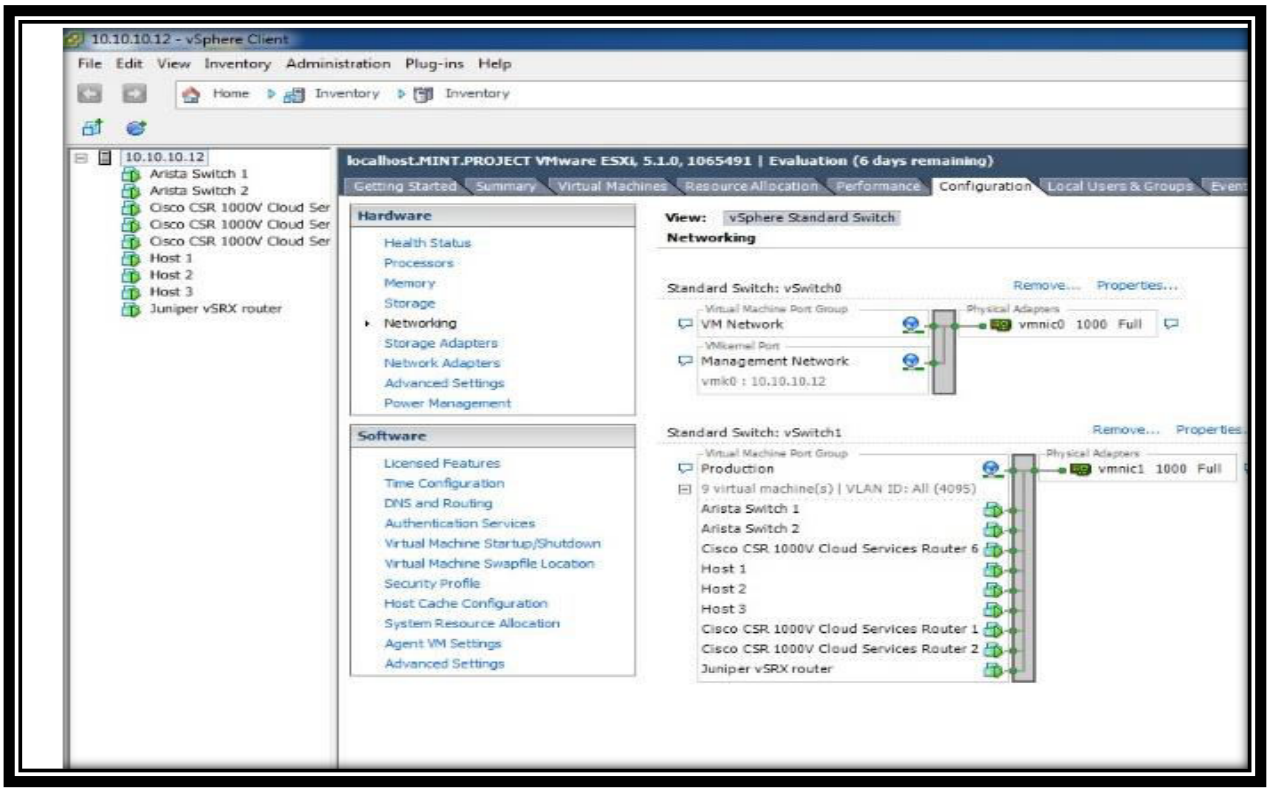


Figure20: *Networking View of ESXI Server 2*

5.9 Networking view for ESXI Server1

We have deployed the VMware vCenter Server virtual appliance (OVA) format in the same way as we deployed for Cisco CSR1000v. The vCenter virtual appliance is available from official VMware download website.

On the ESXI 1 we have deployed the following VMs and created vSwitch1.

- **Brocade Vyatta Router 1**
- **Brocade Vyatta Router 2**
- **Brocade Vyatta Router 3**
- **Brocade Vyatta Router 4**
- **Brocade Vyatta Router 5**
- **Host**
- **VMware vCenter Server Appliance**
- **Virtual Standard Switch (vSwitch1)**

The vSwitch1 is associated with the vmnic1 or pnic1 and all the deployed virtual machines of Brocade Vyatta are connected to the same port group of vSwitch1. But the vCenter server is connected to the port group named "VM Network" of vSwitch0 which is basically for management network. By doing this, we can manage all the virtual machines, multiple ESXI servers and network through vCenter Server.

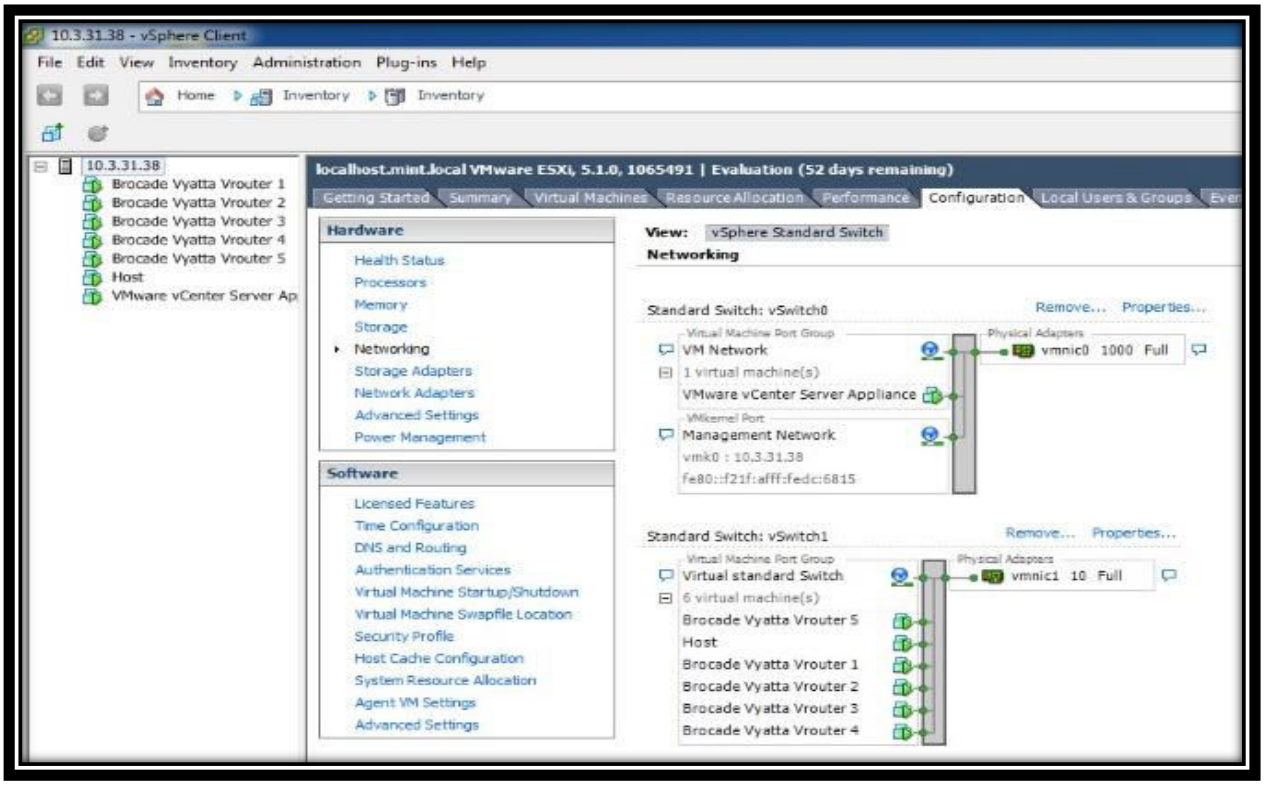
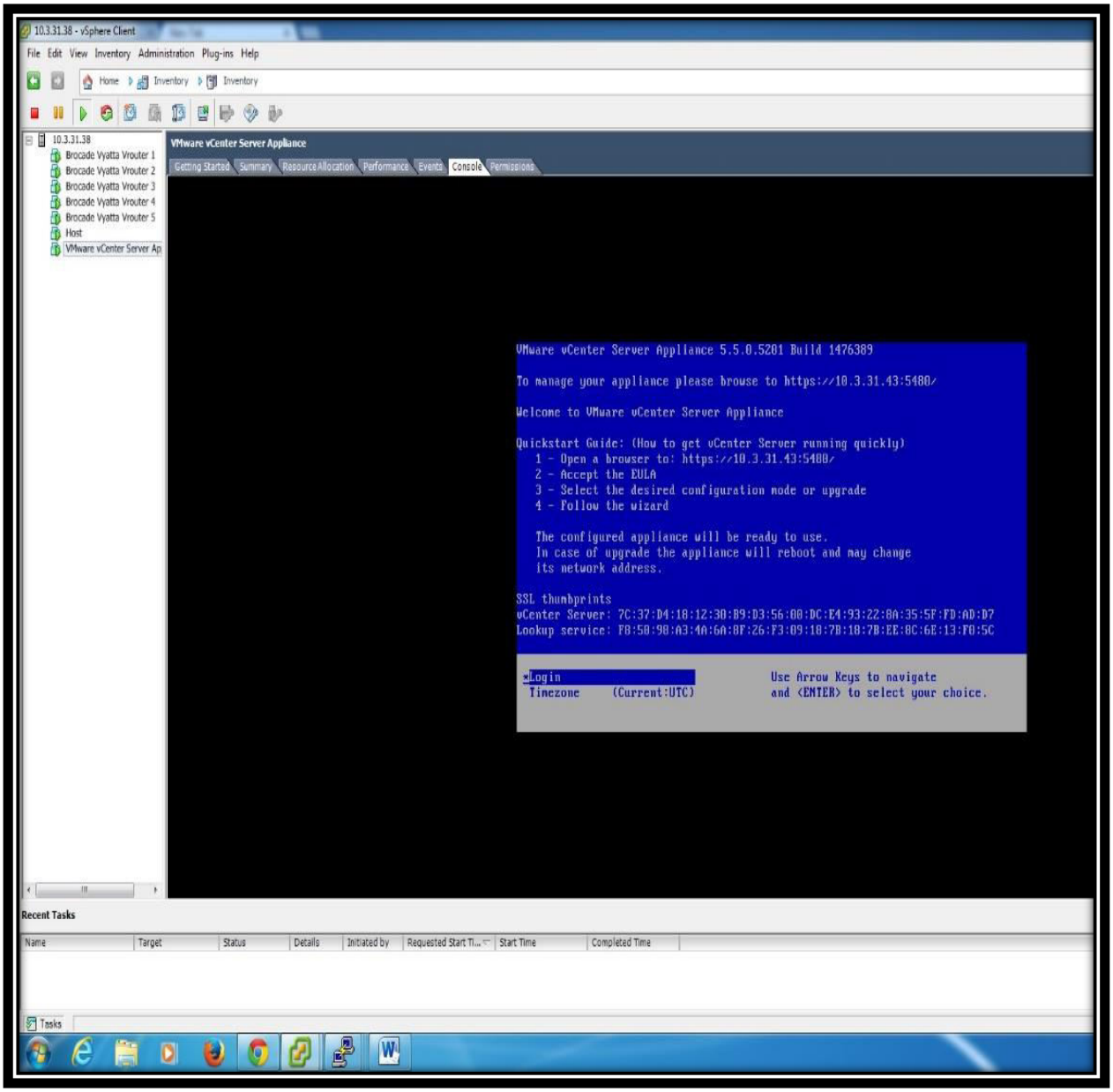


Figure21: *Networking View of ESXI Server1*

5.10 Accessing and Configuring vCenter Server

The vCenter server appliance is a pre-configured Linux based virtual machine optimized for running vCenter server and associated services.

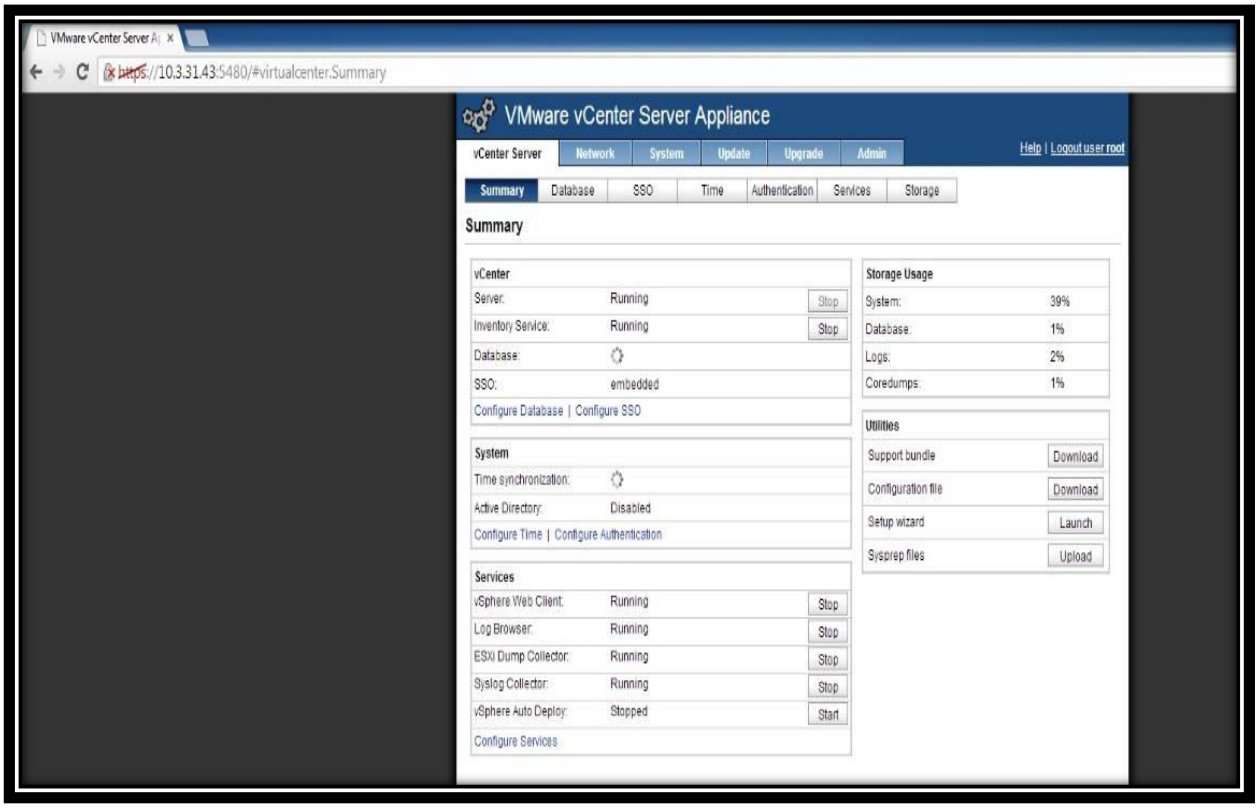
Step1: As we have already deployed vCenter server appliance and it is powered on. When we open the console of the vCenter appliance we can see the IP address 10.3.31.43 assigned to it from the DHCP server. This address will allow us to configure and manage the vCenter server appliance. We will use the secure browsing using https.



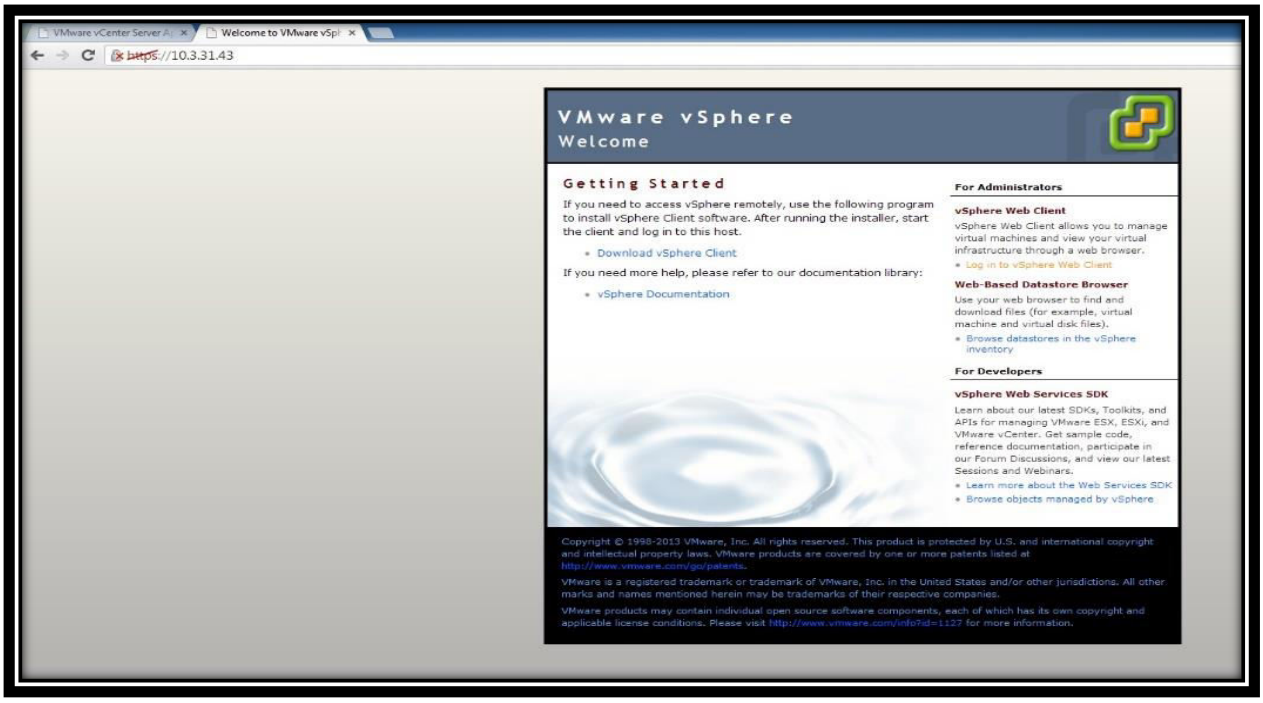
Step 2: Go to the internet browser and type the ip address and port number that was assigned for vCenter Server appliance through secure browsing i-e https://10.3.31.43:5480/. This will direct us towards the login page of vCenter server appliance.

Step 3: Login with the default username and password i-e. Username "root" and Password "vmware".

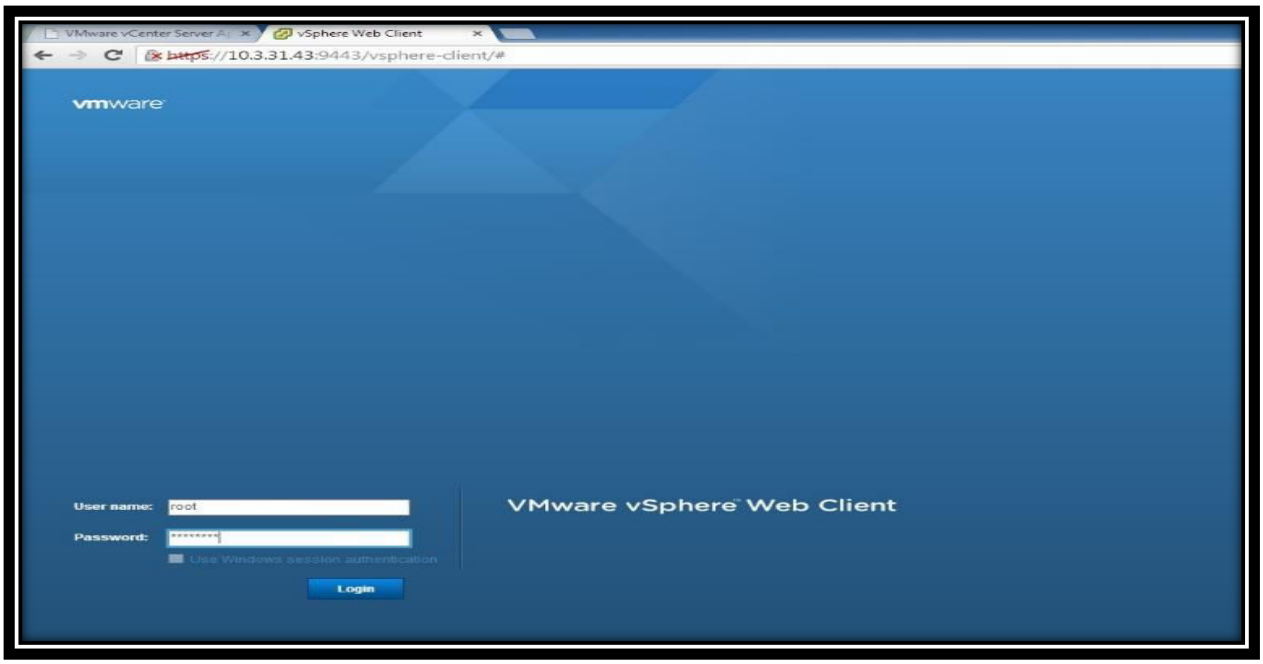
Step 4: Select all the default settings including the embedded database. Accept the end user license agreement and select the default configuration and click start to startup all the services. The embedded database supports upto 5 ESXI hosts and 50 virutal machines. Make sure that the server, inventory service and all the parameters in the Services column should be running.



Step 5: Once the vCenter server appliance is configured, we can browse to the home page of vCenter server using the same IP address "10.3.31.43" but without the port this time. Open the vSphere web client from this home page associated with vCenter server.



Step 6: After opening the vSphere web client login with the same username and password that we used for configuring vCenter server appliance. Through vSphere web client we can access the vCenter server to manage our network.

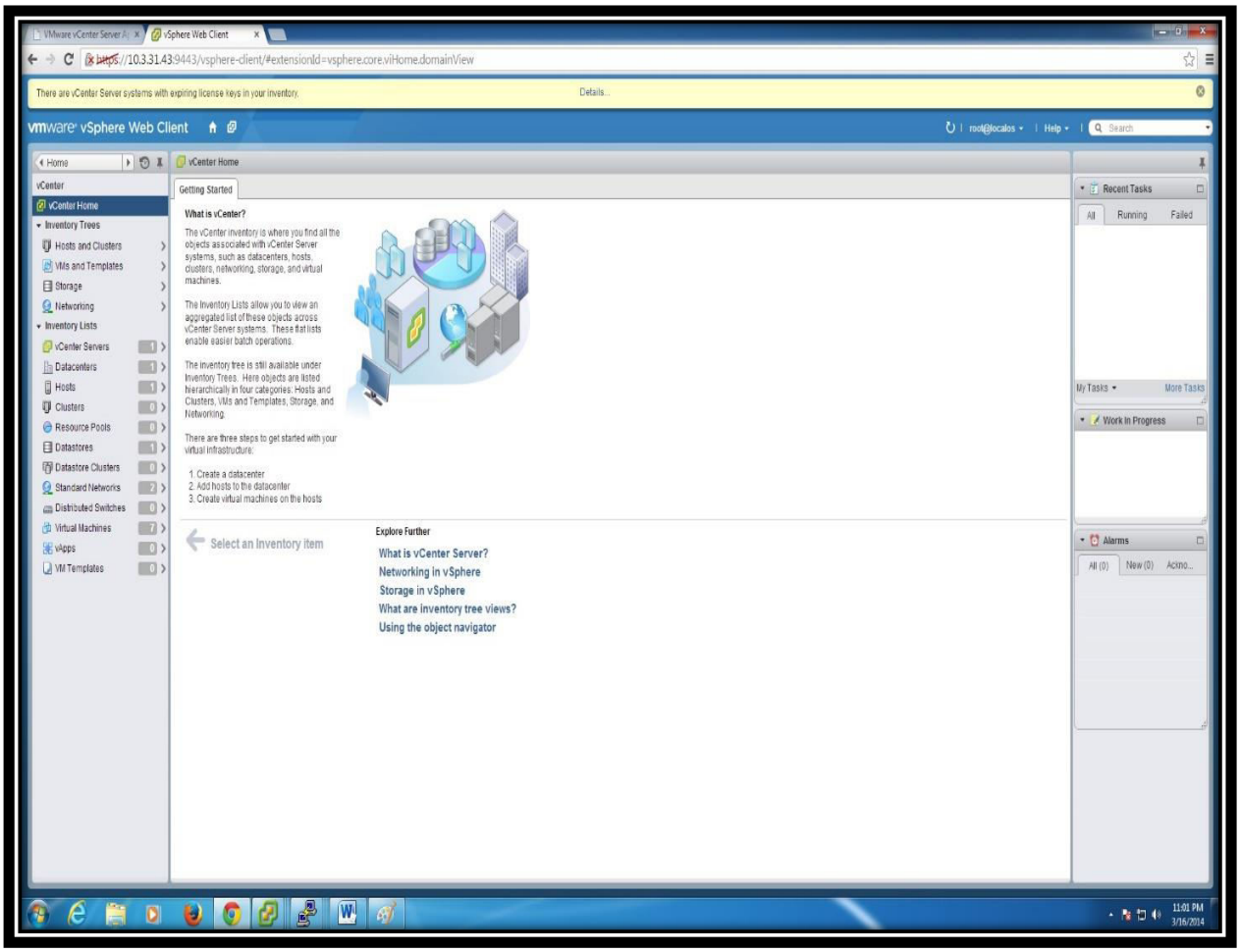


5.11 Creating vCenter Inventory

In vCenter Inventory we can find all the objects associated with vCenter server such as datacenter, host, cluster, networking, storage and virtual machines. All these objects can be managed through vCenter server if we put them in the vCenter Inventory.

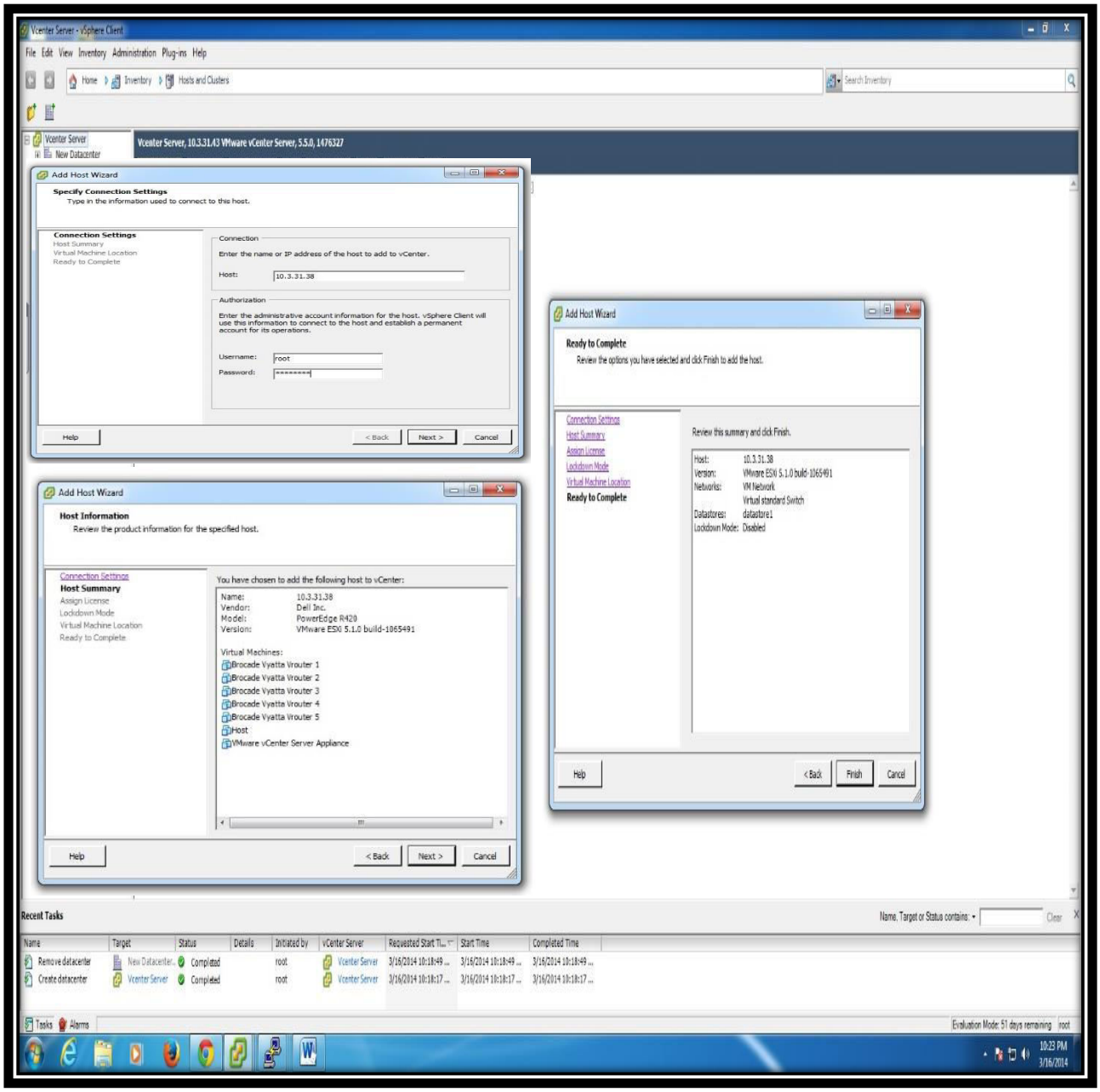
5.11.1 Creating Data Center in vCenter Inventory

In data center all the ESXI hosts and the virtual machines reside. For creating data center select the "Data Center" icon from the inventory list and click "create a new Data Center" and name it "New Datacenter". After that we will select the vCenter server where we want to create a data center and click ok to complete the task of creating new data center.



5.11.2 Adding ESXI Server 1 in Data Center

Firstly, we have to connect with vCenter Server using the vSphere client. It can be seen that New Datacenter has been created under the vCenter Server tab in the figure below. Now, right click on New Datacenter and select “Add Host”. Give the IP address of the ESXI (host) which you want to add in your data center. In the next step, it will ask for the username and password for the authentication purpose. Provide the username “root” and password “Mint709?” of the ESXI server 1. After that, select the location of ESXI as “New Datacenter”. Review the summary of adding the ESXI host in data center and click finish for the task to be completed.



5.11.3 Summary of Created Data Center

The data center has been created successfully and all the objects like hosts, virtual machines, networks and datastore are now residing under the newly created data center as shown below:

Figure22: *Summary of Virtualized Data Center*

The right side of the above figure shows the total CPU usage, memory usage and storage utilization by the data center and its residing objects.

Now, we can manage and modify the data center, add or remove ESXI host, deploy or delete all virtual machines and virtual switches under the centralized vCenter Server as shown below:

Figure23: *View of Deployed vCenter Server Virtual Machine*

We can also add another host in the vCenter Server environment for the high availability feature. Suppose, if the running ESXI server fails or shutdown for some reason then all the virtual machines and virtual switches can be transferred automatically to other ESXi server through vCenter server without losing any data.

5.12 vCenter Server Topology Map

vCenter server map is a visual representation of the vCenter Server topology. The vCenter map shows the relationship between the virtual and physical resources that are available to vCenter Server. To view the map of vCenter topology just select the vCenter server and click the map tab. vCenter map helps to determine the things like which clusters or hosts are most densely populated, which networks are most critical, and which storage devices are being utilized

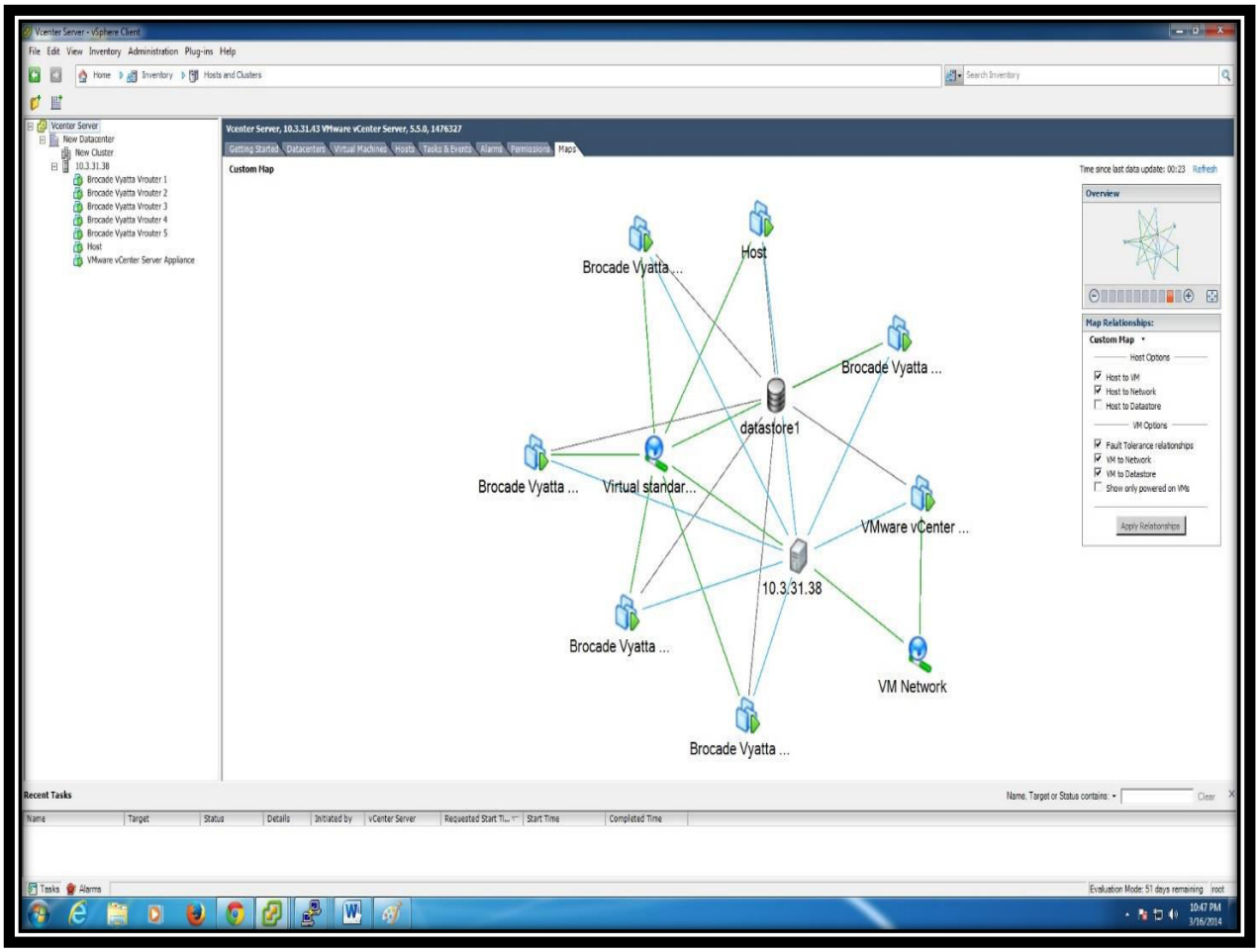


Figure24: *Topology Map of vCenter Server*

The above figure displays the following relationships:

- The virtual machine centric relationships of all Brocade Vyatta virtual machines with datastore1, ESXI host and virtual standard switch.
- The host centric relationship of ESXI host with all virtual machines, virtual standard switch, vCenter server and VM network.
- The datastore centric relationship of datastore1 with all virtual machines and vCenter server.
- The vCenter centric relationship of VMware vCenter server with ESXI host, VM Network and with Datastore1.

6. LAB EXPERIMENT DEMO WITH RESULTS

6.1 Configuring Enterprise Data Center Physical Routers

We are taking four Cisco 2900 series physical routers for our Enterprise Data Center topology residing in the MINT lab giving Autonomous system number 65001 as demonstrated in the network topology below:

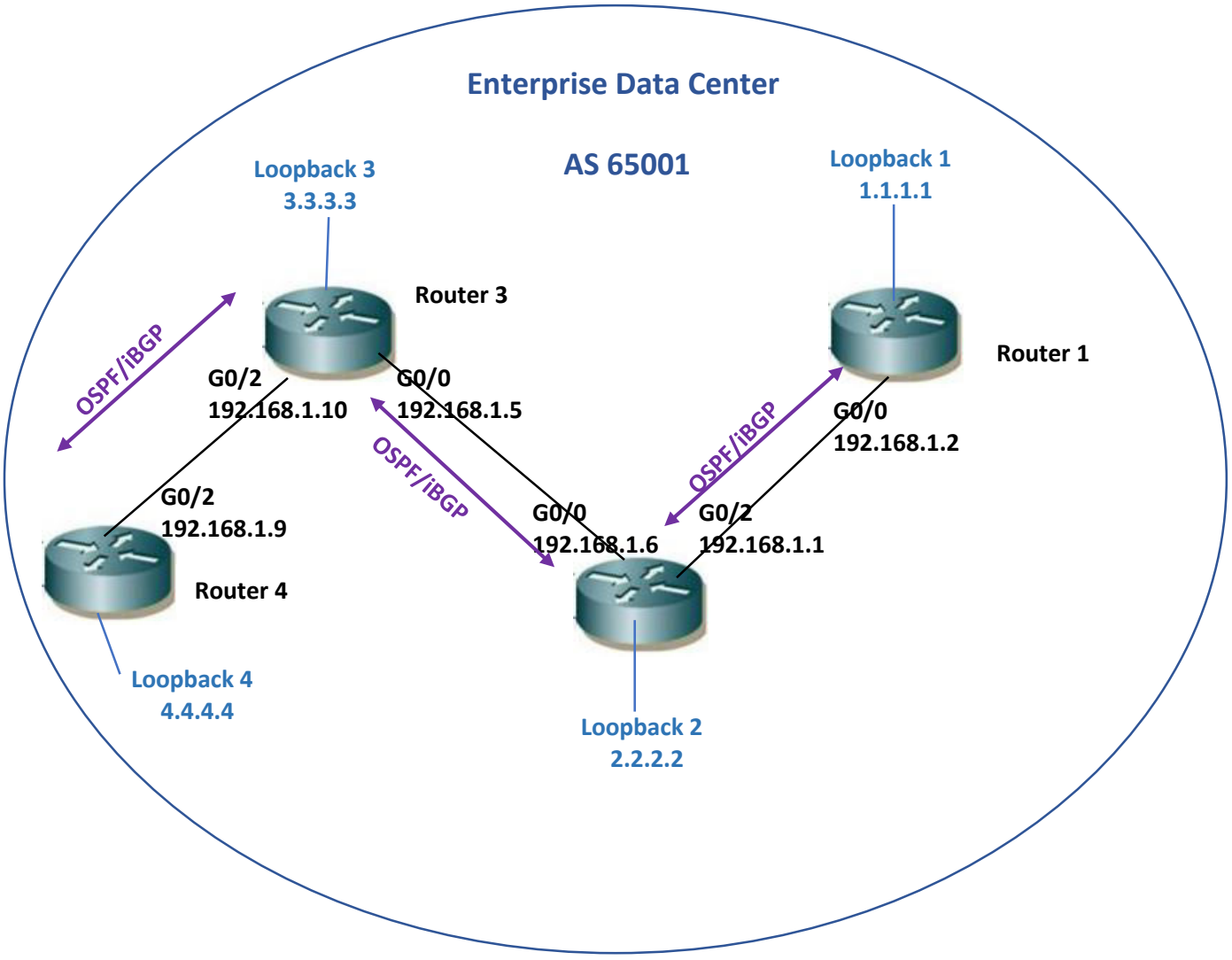


Figure 25: *Physical Data Center Routers Topology*

Configuration Steps:

- Configuring Interfaces of all Cisco routers
- Configuring loopbacks of all routers
- Configuring OSPF area0 between all routers and also for loopbacks
- Configuring iBGP between all routers through loopbacks

6.1.1 Configuration Demo of Router 2

Here we are just showing the configuration of router 2 as the configuration of other routers will be similar to router 2.

```
Router_2#show running-configuration
hostname Router_2
!
interface Loopback2
 ip address 2.2.2.2 255.255.255.255
!
interface GigabitEthernet0/0
 description To-Router_3
 ip address 192.168.1.6 255.255.255.252
!
interface GigabitEthernet0/2
 description To-Router_1
 ip address 192.168.1.1 255.255.255.252
!
router ospf 1
 network 2.2.2.2 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.3 area 0
 network 192.168.1.4 0.0.0.3 area 0
!
router bgp 65001
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 65001
 neighbor 1.1.1.1 update-source Loopback2
 neighbor 3.3.3.3 remote-as 65001
 neighbor 3.3.3.3 update-source Loopback2
 neighbor 4.4.4.4 remote-as 65001
 neighbor 4.4.4.4 update-source Loopback2
 neighbor 5.5.5.5 remote-as 65001
 neighbor 5.5.5.5 update-source Loopback2
 neighbor 6.6.6.6 remote-as 65001
 neighbor 6.6.6.6 update-source Loopback2
!
end
```

6.2 Configuring Virtual Routers for Virtualized Brocade Vyatta Data Center

We have already created the five virtual machines and deployed five Brocade Vyatta 5400 vRouters. Now, we will configure them using Autonomous system number 65002. The Network topology for virtualized Data center is given below:

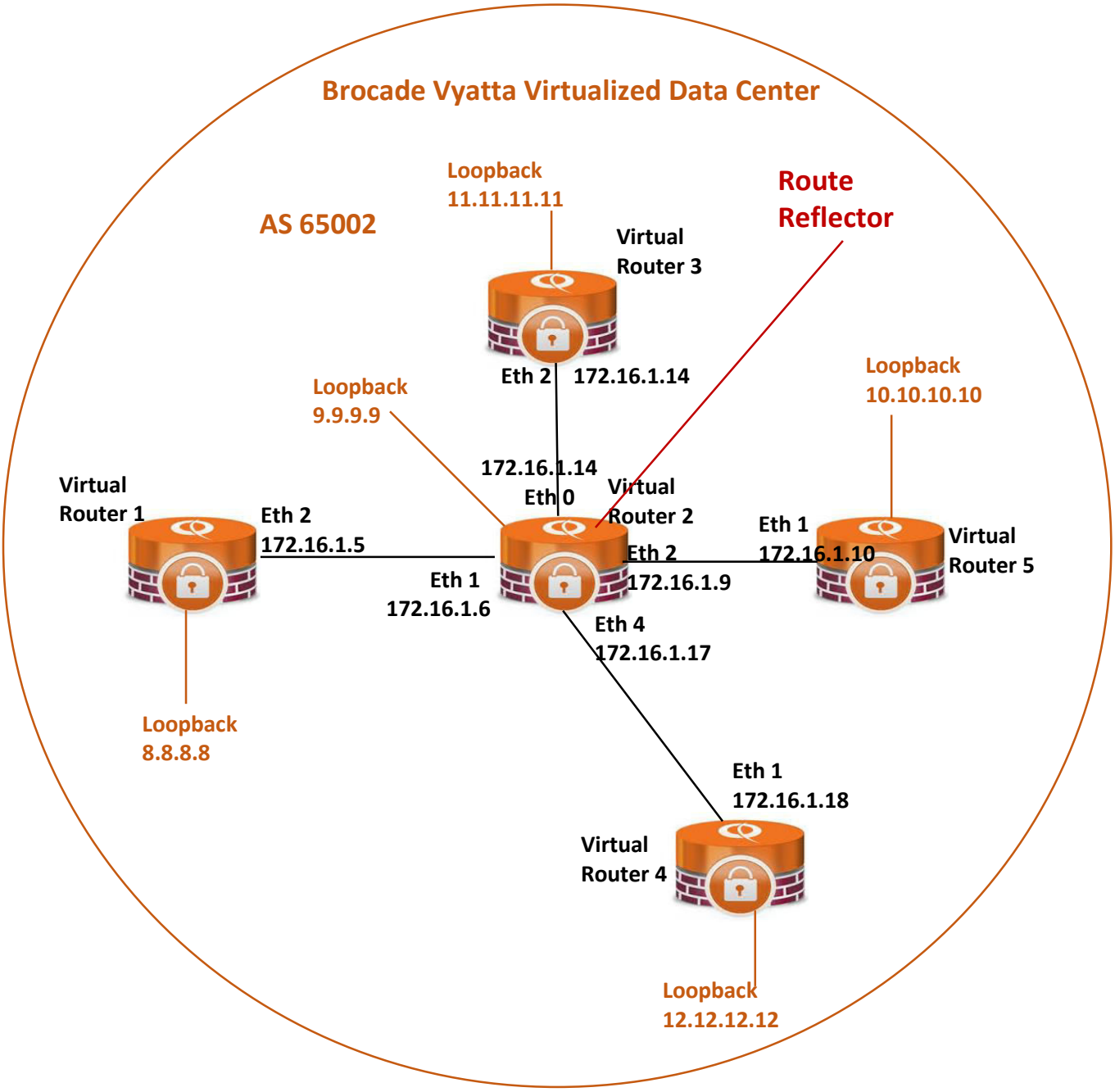


Figure 26: *Brocade Vyatta Virtualized Data Center Topology*

Configuration Steps:

- Configuring Ethernet Interfaces of all Brocade Vyatta routers
- Configuring loopbacks of all routers
- Configuring OSPF area0 between all links and also for loopbacks
- Configuring iBGP according to following:
  - Between virtual router 1 and virtual router 2 which is acting as a Route Reflector (RR).
  - Between virtual router 3 and virtual router 2 (RR)
  - Between virtual router 4 and virtual router 2 (RR)
  - Between virtual router 5 and virtual router 2 (RR)
- Configuring virtual router 2 as a router reflector.

6.2.1 Virtual Router 2 as a Route Reflector

In the above network topology Virtual Router 2 is serving as a Route Reflector. Let's say, there are thousands of routers within the one Autonomous System and it is totally impractical for each BGP speaker to form a full mesh topology with every other BGP speaker in that Autonomous System. Through Route Reflector it is easier to administrate the large number of routers instead of each router advertising directly to all the other routers in the same Autonomous System. One router will serve as a central hub for all the other routers to advertise networks towards it. The central hub is called the Route Reflector. All the routers connected to the router reflector are known as router reflector clients. Virtual routers 1, 3, 4 and 5 are route reflector clients in our case.

6.2.2 Configuration Demo of Virtual Router 2 as a Route Reflector

```
vyatta@Vrouter2# run show configuration _


interfaces {
    ethernet eth0 {
        address 172.16.1.13/30
        description To-Vrouter_3
        duplex auto
        hw-id 00:50:56:9f:4f:5f
        smp_affinity auto
        speed auto
    }
    ethernet eth1 {
        address 172.16.1.6/30
        description To-Vrouter_1
        duplex auto
        hw-id 00:50:56:9f:c9:68
        smp_affinity auto
        speed auto
    }
    ethernet eth2 {
        address 172.16.1.9/30
        description To-Vrouter_4
        duplex auto
        hw-id 00:50:56:9f:22:5a
        smp_affinity auto
        speed auto
```

```
    }
    ethernet eth4 {
        address 172.16.1.17/30
        description To-Vrouter_5
        hw-id 00:50:56:bb:7c:a0
    }
    loopback lo {
        address 9.9.9.9/32
    }
}
protocols {
    bgp 65002 {
        neighbor 8.8.8.8 {
            remote-as 65002
            route-reflector-client
            update-source 9.9.9.9
        }
        neighbor 10.10.10.10 {
            remote-as 65002
            route-reflector-client
            update-source 9.9.9.9
        }
        neighbor 11.11.11.11 {
            remote-as 65002
```

```
            route-reflector-client
            update-source 9.9.9.9
        }
        neighbor 12.12.12.12 {
            remote-as 65002
            route-reflector-client
            update-source 9.9.9.9
        }
    }
    ospf {
        area 0 {
            network 172.16.1.4/30
            network 172.16.1.8/30
            network 172.16.1.16/30
            network 172.16.1.12/30
            network 9.9.9.9/32
        }
    }
```

6.2.3 Verifying Neighbors at Virtual Router2

After configuring virtual router 2 as a route reflector, it should have four BGP neighbors. We can verify it by "**run show ip bgp summary"** command on virtual router 2 as given below:

```
vyatta@Vrouter2# run show ip bg summary
BGP router identifier 9.9.9.9, local AS number 65002
IPv4 Unicast - max multipaths: ebgp 1 ibgp 1
RIB entries 73, using 4672 bytes of memory
Peers 4, using 10096 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
8.8.8.8         4 65002    9020    8961        0    0    0 6d04h58m       36
10.10.10.10     4 65002    8897    8963        0    0    0 6d04h14m        1
11.11.11.11     4 65002    8958    9031        0    0    0 6d04h58m        0
12.12.12.12     4 65002    8922    8988        0    0    0 6d04h40m        1

Total number of neighbors 4
```

6.3 Configuration between Virtualized and Physical Data Center

We will configure eBGP between Physical Data center router1 which is in AS 65001 and Virtual data center virtual router1 residing in AS 65002.

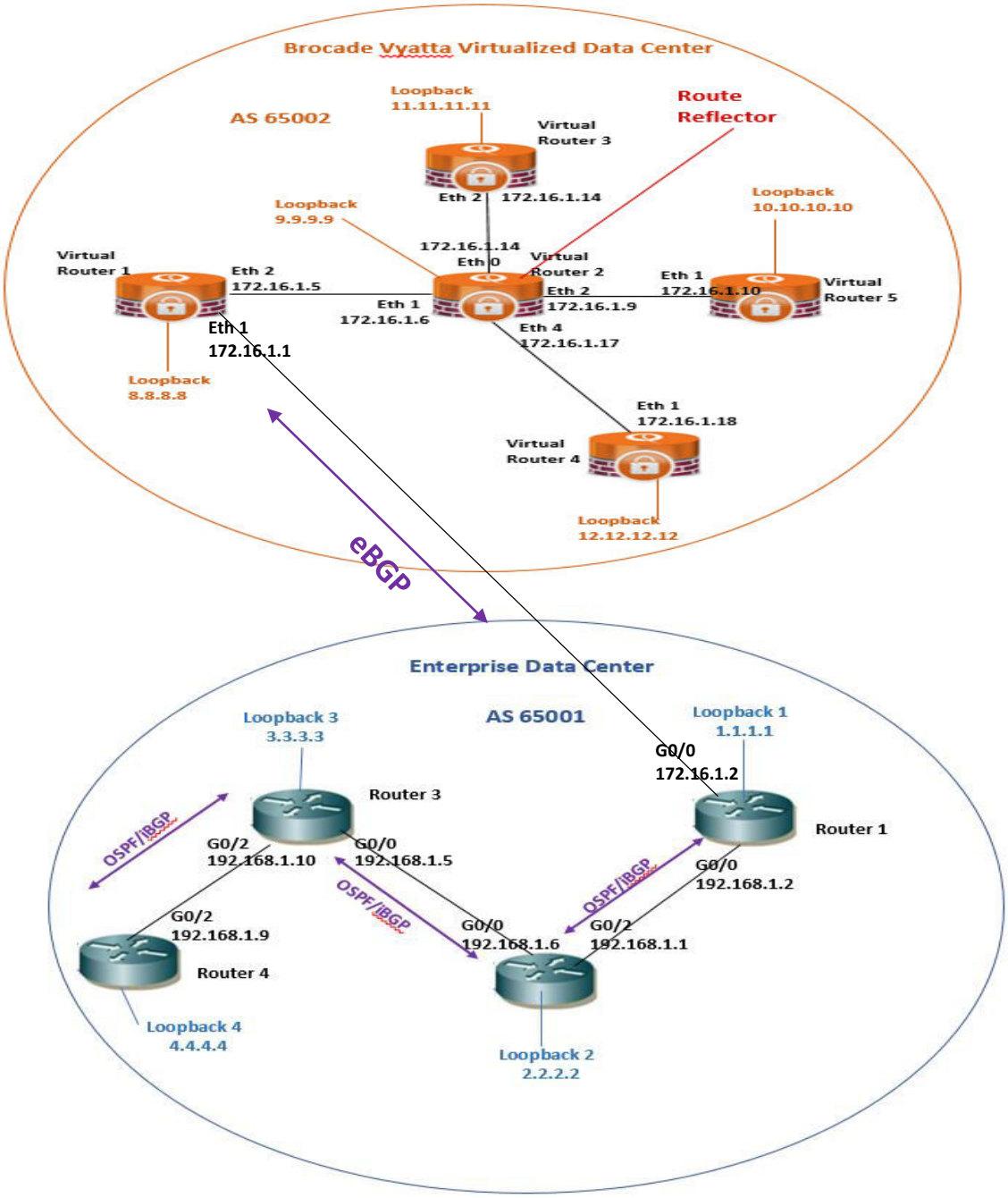
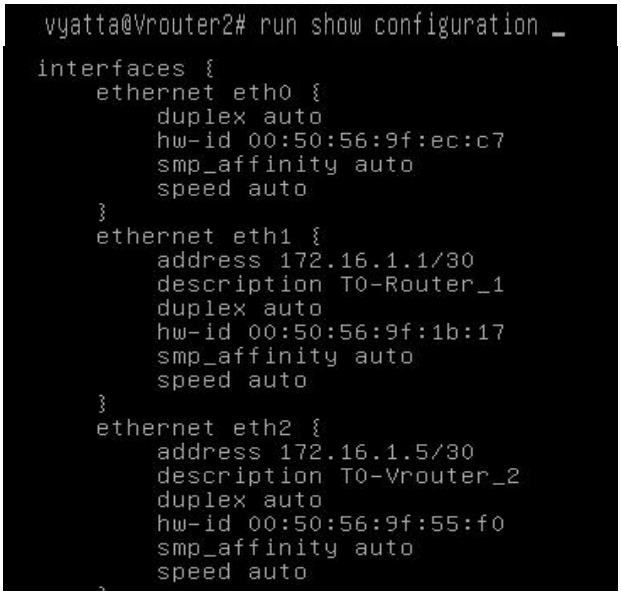


Figure 27: *eBGP between Physical and Virtual Router*

Configuration Steps:

- Configure interfaces towards eBGP neighbor on physical router1 and virtual router1.
- Configure eBGP between physical router1 (AS65001) and virtual router1 (AS65002).
- Redistribute connected networks into BGP on both routers
- Redistribute OSPF routes into BGP on both physical and virtual routers1
- Configure iBGP neighbors as next-hop-self

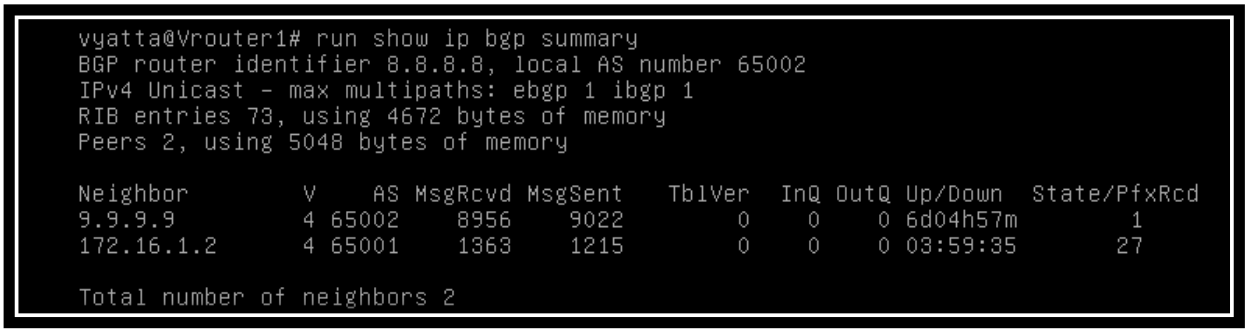
6.3.1 Configuration demo of Virtual Router 1



The same configuration steps will be done at Cisco platform physical router1.

6.3.2 Verifying Neighbors at Virtual Router1

It should have two neighbors i-e one iBGP peering with route reflector virtual router2 and one eBGP peering with Cisco physical router1.



6.3.3 Verifying Reachability between Virtualized Data center (AS 65002) and Physical Enterprise data Center (AS 65001)

Now, we should be able to ping from any virtual router residing in virtualized data center to any physical router residing in physical enterprise data center and vice versa.

Verifying reachability from physical router4 to virtual router2 (Route Reflector):

```
Router_4#ping 9.9.9.9
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 9.9.9.9, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Verifying reachability from physical router4 to virtual router2 (Route Reflector):

```
Vyatta@Vrouter5# ping 4.4.4.4
PING 4.4.4.4 (4.4.4.4) 56(84) bytes of data .
64 bytes from 4.4.4.4: icmp_req=1 ttl=59 time=5.71 ms
64 bytes from 4.4.4.4: icmp_req=1 ttl=59 time=2.30 ms
64 bytes from 4.4.4.4: icmp_req=1 ttl=59 time=5.39 ms

- - - 4.4.4.4 ping statistics - - -
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 2.506/4.538/5.711/1.442 ms
```

Form the above results it is verified that the physical data center routers and virtual data center routers are reachable and can ping each other.

6.4 Configuring Juniper vSRX Services Gateway Firewall

After the successful deployment of virtual machine for Juniper vSRX virtual Firewall we will include Juniper vSRX virtual Firewall in our Enterprise Data center which will be acting as a Internet Border Router and it is connected to the physical router4 in Enterprise data center (AS 65001) as shown in the diagram below:

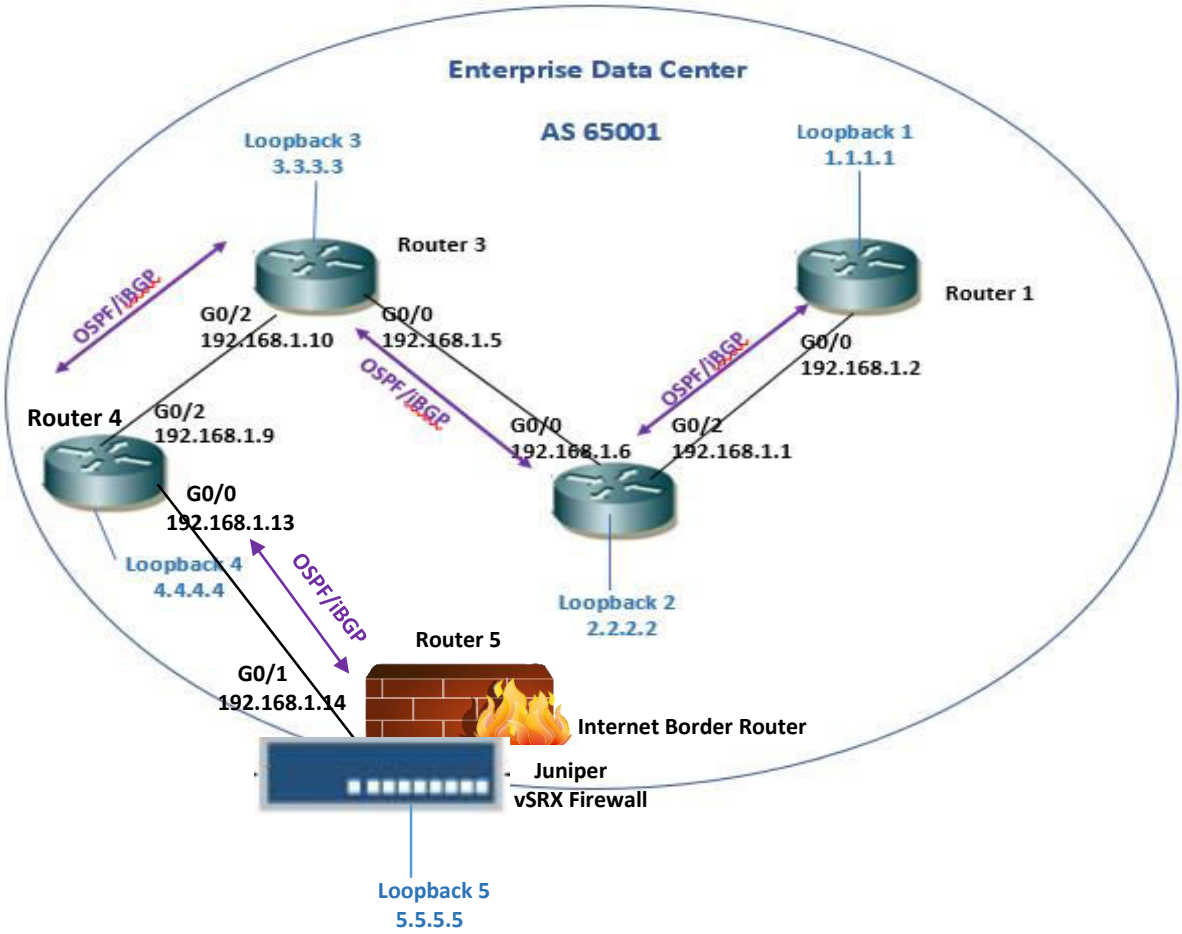


Figure 28: *Configuring Juniper vSRX Firewall in Enterprise Data Center (AS 65001)*

Configuration Steps:

- Configuring Interfaces of Juniper firewall (Router5) in trusted security zone and untrusted security zone and enable ping under host-inbound traffic.
- Configuring loopbacks of Juniper firewall (Router5) in trusted security zone.
- Configuring OSPF area0 between Juniper router5 and Cisco router4 and also include juniper interfaces and OSPF protocol in trusted security zone.
- Configuring iBGP between Juniper router5 and all Cisco routers through loopbacks.

6.5 Configuring Cisco Cloud Services Router CSR1000v for Public Cloud

The virtual machine deployment of Cisco Cloud Services Router CSR1000v has already been done. Now, we will add two Cisco CSR1000v routers in our topology for the two Public Clouds having AS100 and AS200, which will be acting as a service providers over the internet. Both Internet CSR1000v routers will be connected to Juniper vSRX router via eBGP peering as shown in the diagram below:

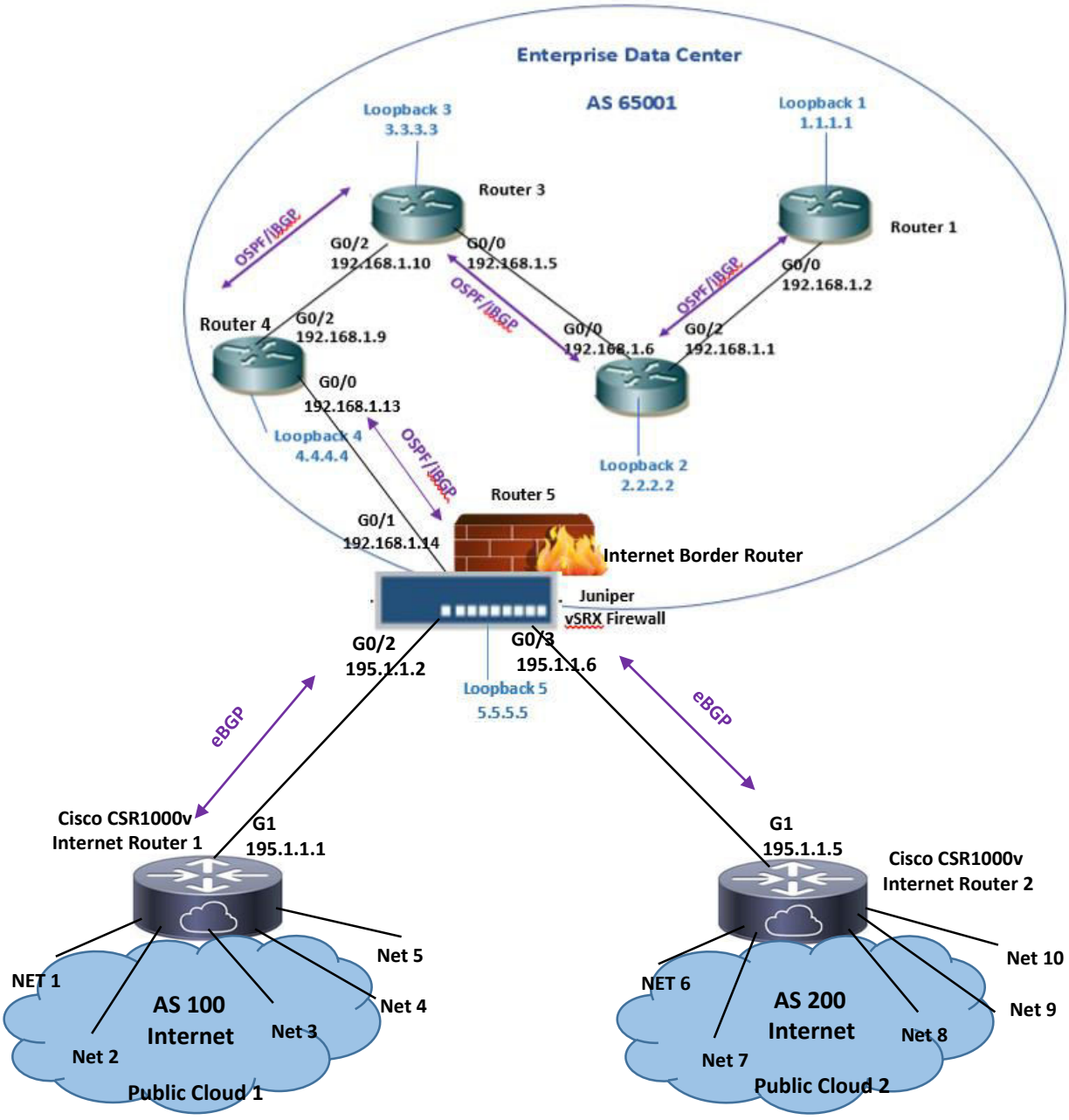


Figure 29: *Topology of Cisco Cloud Services Router in Public Clouds*

Following are the loopbacks and networks pointing toward the internet at CSR1000v Internet router 1.

Loopbacks	Loopback IP Address	Internet-Network
Loopback100	13.13.13.13/23	Net1 - 13.13.0.0/23
Loopback101	14.14.14.14/22	Net2 - 14.14.0.0/22
Loopback102	15.15.15.15/21	Net3 - 15.15.0.0/21
Loopback103	16.16.16.16/20	Net4 - 16.16.0.0/20
Loopback104	17.17.17.17/19	Net5 - 17.17.0.0/19

Following are the loopbacks and networks pointing toward the internet at CSR1000v Internet router 2.

Loopbacks	Loopback IP Addresses	Internet-Networks
Loopback105	18.18.18.18/18	Net6 - 18.18.0.0/18
Loopback106	19.19.19.19/17	Net7 - 19.19.0.0/17
Loopback107	20.20.20.20/16	Net8 - 20.20.0.0/16
Loopback108	21.21.21.21/15	Net9 - 21.0.0.0/15
Loopback109	22.22.22.22/14	Net10 - 22.0.0.0/14

Configuration Steps:

- Configuring interfaces of CSR1000v internet router 1 and 2.
- Configuring loopbacks at CSR1000v internet router 1 and 2 according to above IP addressing schema.
- Configuring eBGP according to the following:
  - Between Juniper router and Cisco CSR1000v internet router 1.
  - Between Juniper router and Cisco CSR1000v internet router 1.

At Internet Border Router:

- Create policy statements for redistributing directly connected routes into BGP, redistributing OSPF routes into BGP and policy statement for next-hop self at Juniper router and apply it to internal and external peer groups.

At Internet Router1:

- Advertise the Internet-Networks from Net1 to Net5 in BGP to simulate the internet routes in AS100.
- Redistribute connected networks into BGP

At Internet Router1:

- Create 5 static routes for the Internet-Networks from Net6 to Net10 pointing to Null0 so as to simulate the internet routes in AS200.
- Redistribute static networks into BGP

6.5.1 Configuration demo of Juniper vSRX

```
## Last commit: 2014-03-13 19:50:20 UTC by root
version 12.1X46-D10.2;
system {
    host-name Internet-border-router;
    root-authentication {
        encrypted-password "$1$aIEir5.8$EN9pEYxhMwYcnmjg6IBfU."; ## SECRET-DATA
    }
    services {
        ssh;
        web-management {
            http {
                interface ge-0/0/0.0;
            }
            https {
                system-generated-certificate;
                interface ge-0/0/0.0;
    }
    interfaces {
        ge-0/0/0 {
            description To-Router_6;
            unit 0 {
                family inet {
                    address 192.168.1.25/30;
                }
            }
        }
        ge-0/0/1 {
            description To-Router_4;
            unit 0 {
                family inet {
                    address 192.168.1.14/30;
                }
            }
        }
        ge-0/0/2 {
            description To-Internet_Router_1;
            unit 0 {
                family inet {
                    filter {
                        input icmp-filter;
                    }
                    address 195.1.1.2/30;
                    address 100.100.100.1/24;
                }
            }
        }
        ge-0/0/3 {
            description To-Internet_Router_2;
            unit 0 {
                family inet {
                    address 195.1.1.6/30;
                }
            }
        }
        ge-0/0/4 {
            unit 0 {
                family inet;
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 5.5.5.5/32;
                }
            }
        }
    }
    routing-options {
        static {
            route 192.168.2.0/24 next-hop 192.168.1.26;
            route 192.168.3.0/24 next-hop 192.168.1.26;
        }
        autonomous-system 65001;
    }
    protocols {
        bgp {
            export [ ospf-into-bgp next-hop-self ];
            group external-peers {
                type external;
                peer-as 100;
                neighbor 195.1.1.1;
            }
            group internal-peers {
                type internal;
                local-address 5.5.5.5;
                export send-direct;
                neighbor 4.4.4.4;
                neighbor 3.3.3.3;
                neighbor 2.2.2.2;
                neighbor 1.1.1.1;
                neighbor 6.6.6.6;
            }
            group external-peers2 {
                type external;
                peer-as 200;
                neighbor 195.1.1.5;
            }
        }
        ospf {
            area 0.0.0.0 {
                interface ge-0/0/1.0;
                interface lo0.0;
                interface ge-0/0/0.0;
            }
        }
    }
    policy-options {
        policy-statement exp2bgp {
            then accept;
        }
        policy-statement next-hop-self {
            then {
                next-hop self;
                accept;
            }
        }
        policy-statement ospf-into-bgp {
            term ospf-into-bgp {
                from {
                    protocol ospf;
                    area 0.0.0.0;
                }
                then accept;
            }
        }
        policy-statement send-direct {
            term send-direct {
                from protocol direct;
                then accept;
            }
        }
    }
    security {
        screen {
            ids-option untrust-screen {
                icmp {
                    ping-death;
                }
            }
        }
        zones {
            security-zone trust {
                tcp-rst;
            host-inbound-traffic {
                system-services {
                    http;
                    https;
                }
                protocols {
                    ospf;
                }
            }
            interfaces {
                ge-0/0/1.0 {
                    host-inbound-traffic {
                        system-services {
                            ping;
                            ssh;
                            ftp;
                            telnet;
                        }
                        protocols {
                            ospf;
                        }
                    }
                }
                lo0.0 {
```

```
        host-inbound-traffic {
            system-services {
                ping;
                ftp;
                telnet;
                ssh;
            }
            protocols {
                ospf;
            }
        }
    }
    ge-0/0/0.0 {
        host-inbound-traffic {
            system-services {
                ping;
                telnet;
                ssh;
                ftp;
            }
        }
    }
}
```

```
    security-zone untrust {
      interfaces {
         ge-0/0/2.0 {
            host-inbound-traffic {
               system-services {
                  ping;
                  ssh;
                  ftp;
                  telnet;
               }
            }
               ge-0/0/3.0 {
                  host-inbound-traffic {
                     system-services {
                        ping;
                        ssh;
                        telnet;
                        ftp;
                     }
                  }
```

6.5.2 Verifying reachability from Enterprise and Virtualized Data Centers to Public Cloud Internet Routers

Verifying reachability from Juniper router to CSR1000v Internet router1 (NET-1)

```
root@Internet-border-router# run ping 13.13.13.13
PING 13.13.13.13 (13.13.13.13): 56 data bytes
64 bytes from 13.13.13.13: icmp_seq=0 ttl=255 time=10.062 ms
64 bytes from 13.13.13.13: icmp_seq=1 ttl=255 time=10.407 ms
64 bytes from 13.13.13.13: icmp_seq=2 ttl=255 time=5.353 ms
^C
--- 13.13.13.13 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.353/8.607/10.407/2.305 ms
```

Verifying reachability from Juniper router to CSR1000v Internet router2 (NET-9)

```
root@Internet-border-router# run ping 21.21.21.21
PING 21.21.21.21 (21.21.21.21): 56 data bytes
64 bytes from 21.21.21.21: icmp_seq=0 ttl=255 time=355.158 ms
64 bytes from 21.21.21.21: icmp_seq=1 ttl=255 time=15.711 ms
64 bytes from 21.21.21.21: icmp_seq=2 ttl=255 time=335.380 ms
^C
--- 21.21.21.21 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 15.711/235.416/355.158/155.565 ms
```

Verifying reachability from Brocade Vyatta router5 to CSR1000v Internet router2(NET-8)

```
vyatta@Vrouter5# ping 20.20.20.20
PING 20.20.20.20 (20.20.20.20) 56(84) bytes of data.
64 bytes from 20.20.20.20: icmp_req=1 ttl=248 time=272 ms
64 bytes from 20.20.20.20: icmp_req=2 ttl=248 time=691 ms
64 bytes from 20.20.20.20: icmp_req=3 ttl=248 time=247 ms
64 bytes from 20.20.20.20: icmp_req=4 ttl=248 time=56.7 ms
^C
--- 20.20.20.20 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4001ms
rtt min/avg/max/mdev = 56.700/317.166/691.969/231.943 ms
```

Verifying reachability and path from Brocade Vyatta virtual router3 to Cisco CSR1000v Internet router1 (NET-2)

```
vyatta@Vrouter3# run traceroute 14.14.14.14
traceroute to 14.14.14.14 (14.14.14.14), 30 hops max, 60 byte packets
 1  172.16.1.13 (172.16.1.13)  0.520 ms  0.512 ms  0.496 ms
 2  172.16.1.5 (172.16.1.5)  0.982 ms  0.987 ms  0.976 ms
 3  172.16.1.2 (172.16.1.2)  1.670 ms  1.666 ms  1.626 ms
 4  192.168.1.1 (192.168.1.1)  1.872 ms  1.912 ms  1.985 ms
 5  192.168.1.5 (192.168.1.5)  2.042 ms  2.148 ms  2.183 ms
 6  192.168.1.9 (192.168.1.9)  2.558 ms  1.913 ms  1.900 ms
 7  192.168.1.14 (192.168.1.14)  6.668 ms  2.904 ms  2.886 ms
 8  195.1.1.1 (195.1.1.1)  7.241 ms * *
```

The above results clearly depict that the routers residing in Brocade Vyatta virtualized data center (AS 65002) and routers in Enterprise data center (AS 65001) can reach and ping the both Internet routers residing in Public Clouds at AS100 and AS200.

6.6 Configuring Arista vEOS vSwitch

The final addition in our network topology is to add two Arista Networks virtual switch. As we have already created virtual machines for Arista vSwitch and deployed them successfully. Now, we can configure them to observe the Arista Network vSwitch features. We will add one more Cisco CSR1000v which will be configured as a "Router on a Stick" for inter-vlan routing in AS 65001. For inter-vlan routing switch will use a router to route the network traffic between vlans.

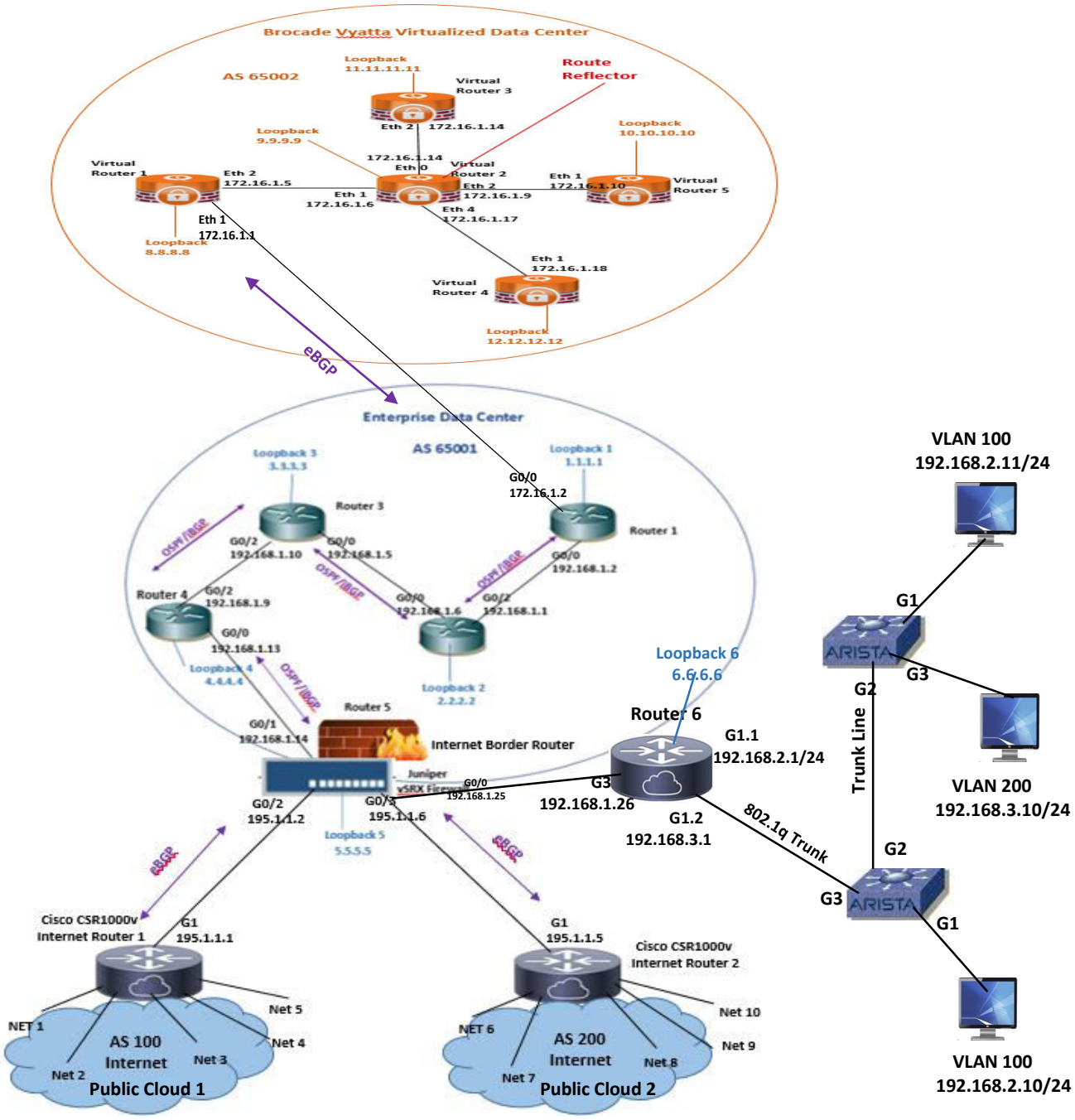


Figure 30: *Topology of Arista Virtual Switch for Inter-VLAN Routing*

Configuration Steps:

- Create VLANs 100 and 200 on both Arista virtual switches.
- Configure switch interfaces as an access mode that are towards the hosts.
- Configure the trunk link between the virtual switches.
- Configure the trunk link between Arista virtual switch1 and CiscoCSR1000v router6.
- Configure sub interfaces with encapsulation dot1Q on CSR1000v router6 to use a router interface as trunk port to a switch.
- Configure OSPF in area0 between Cisco CS1000v router6 and Juniper router5.
- Configure iBGP using loopbacks between Cisco CSR1000v router6 and all 5 enterprise data center routers in AS 65001.
- Configure two static routes on Juniper router5 pointing towards the VLAN networks via router6 interface.
- Redistribute static routes into BGP at Juniper router by configuring policy statement and applying it to internal and external peer groups.

6.6.1 Verification of Trunking at Arista Virtual Switch 1 and 2

```
Name: Et1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
MAC Address Learning: enabled
Access Mode VLAN: 100 (VLAN_100)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Administrative private VLAN mapping: ALL
Trunking VLANs Enabled: 100,200
Trunk Groups:

Name: Et2
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
MAC Address Learning: enabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Administrative private VLAN mapping: ALL
Trunking VLANs Enabled: 100,200
Trunk Groups:

Name: Et3
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
MAC Address Learning: enabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Administrative private VLAN mapping: ALL
Trunking VLANs Enabled: 100,200
Trunk Groups:

Arista-Switch1#_
```

```
Name: Et1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
MAC Address Learning: enabled
Access Mode VLAN: 100 (VLAN_100)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Administrative private VLAN mapping: ALL
Trunking VLANs Enabled: 100,200
Trunk Groups:

Name: Et2
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
MAC Address Learning: enabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Administrative private VLAN mapping: ALL
Trunking VLANs Enabled: 100,200
Trunk Groups:

Name: Et3
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
MAC Address Learning: enabled
Access Mode VLAN: 200 (VLAN_200)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: disabled
Administrative private VLAN mapping: ALL
Trunking VLANs Enabled: 100,200
Trunk Groups:
```

6.6.2 Verifying reachability between Enterprise VLANs and towards Virtualized Data Centers and Public Cloud Internet Routers

Now we should be able to ping from host in vlan100 to host in vlan200 and vice versa.

Verifying reachability from host1 in vlan100 to host3 in vlan200:

```
Host_1#ping 192.168.3.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

Verifying reachability from host3 in vlan200 to host2 in vlan100:

```
Host_3#ping 192.168.2.11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.11, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms
Host_3#_
```

Verifying reachability from host1 in vlan100 to Internet router1 (NET-1):

```
Host_1#ping 13.13.13.13
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 13.13.13.13, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/9/15 ms
```

Verifying reachability from host1 in vlan100 to host in Virtualized data center:

```
Host_1#ping 172.16.2.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/9/11 ms
```

7. TESTING SECURITY AND HIGH AVAILABILITY FEATURES

7.1 High Availability feature of VMware vSphere

High availability of VMware vSphere enables the cluster of ESXI host to work together so as to provide higher levels of availability for virtual machines rather than just the ESXI host by itself. In the event of physical server failure, the affected virtual machines will be automatically started on other ESXI servers that are also in the same cluster and having the spare capacity. In the case of operating system failure of virtual machines, VMware vSphere high availability feature restarts the affected virtual machines on the same physical server. VMware vSphere protects against three types of failures:

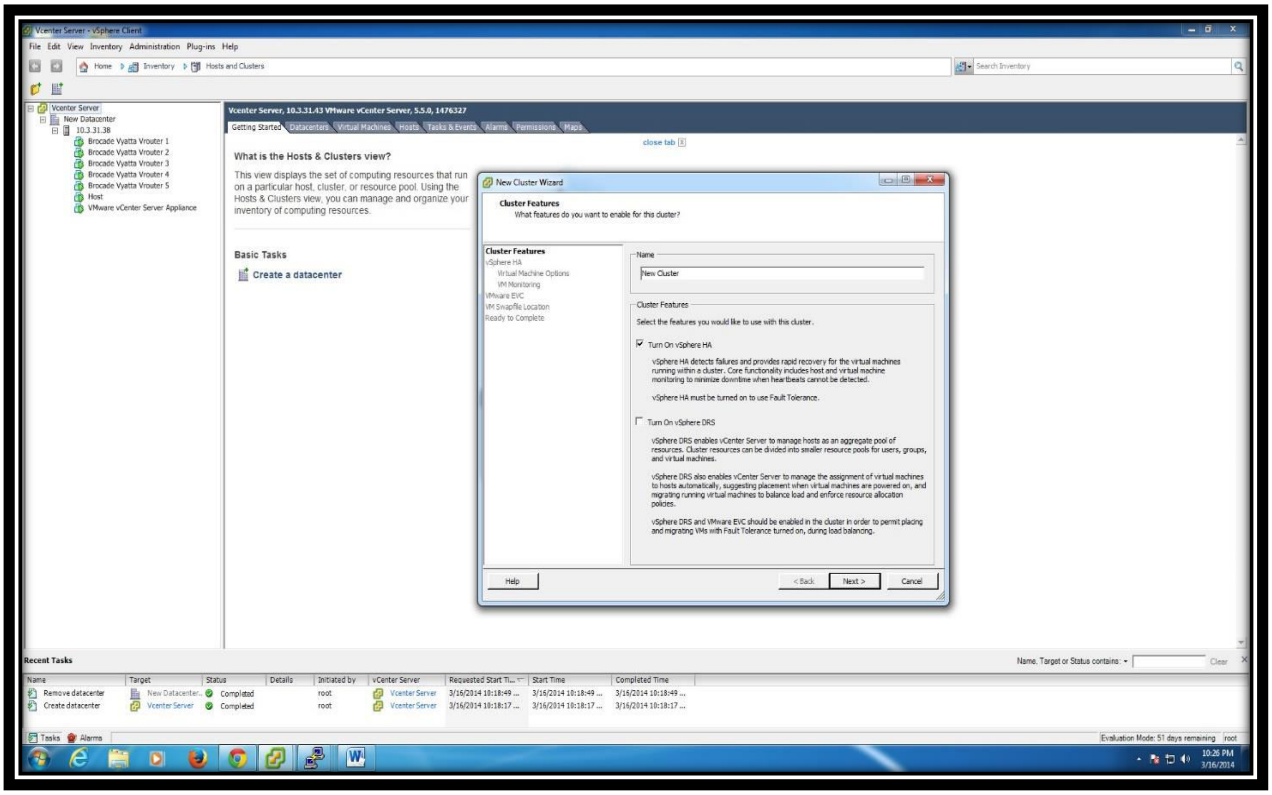
- ESXI host failure
- Virtual machine operating system failure
- Application failure

7.1.1 Configuring High Availability feature of VMware vSphere

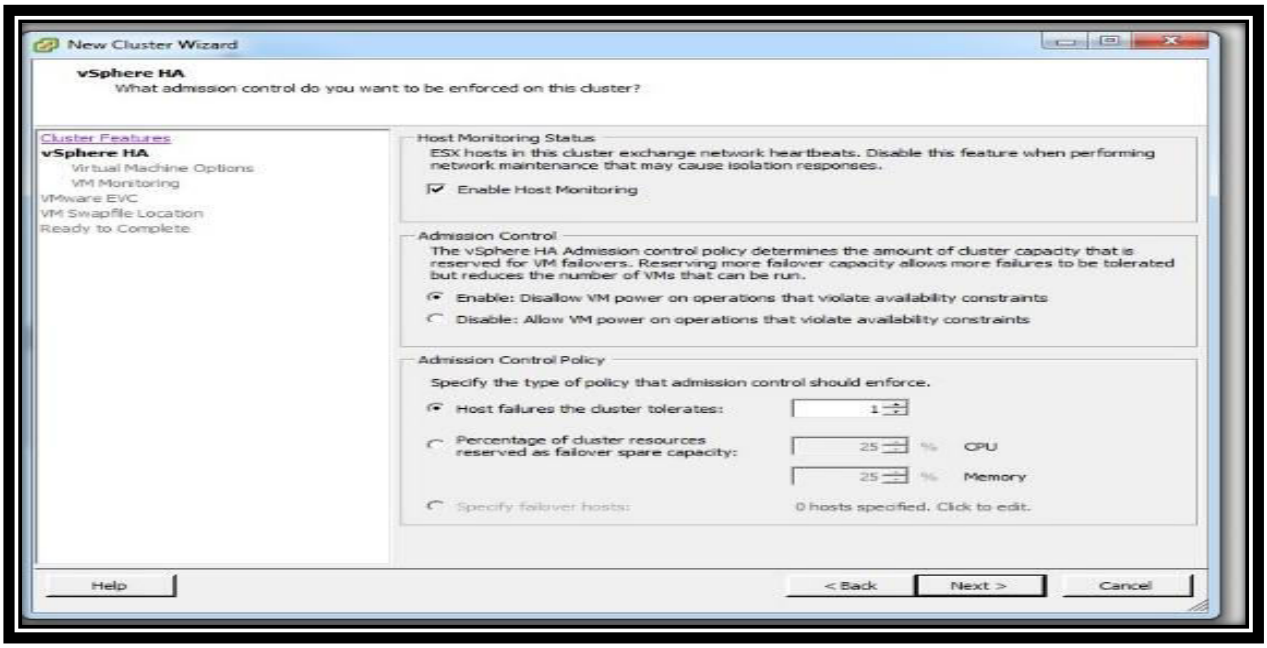
We will configure high availability feature on ESXI server 1 for Brocade Vyatta virtual machines.

Step 1: Login to vCenter Server and right click the New data center under getting started tab and select create a new cluster.

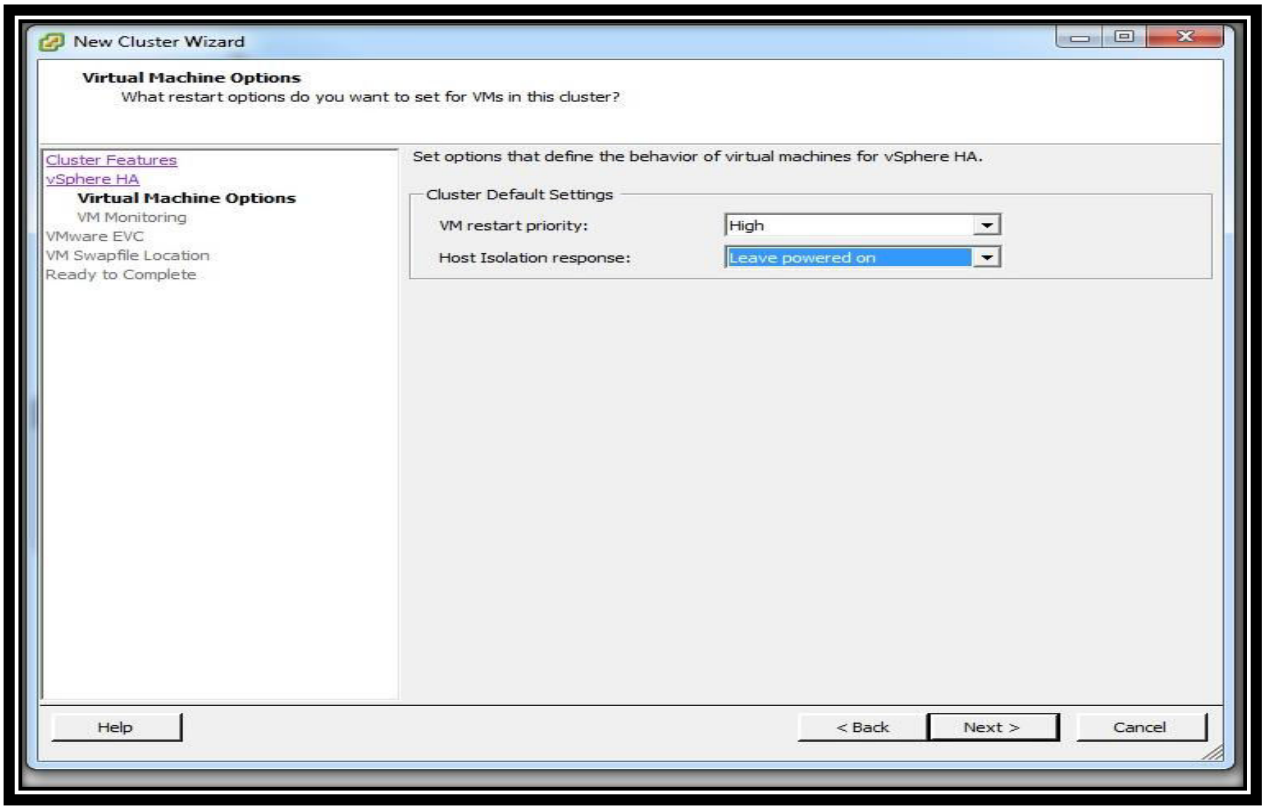
Step2: Under the Cluster features create the name for cluster and turn on vSphere HA feature. Then click next.



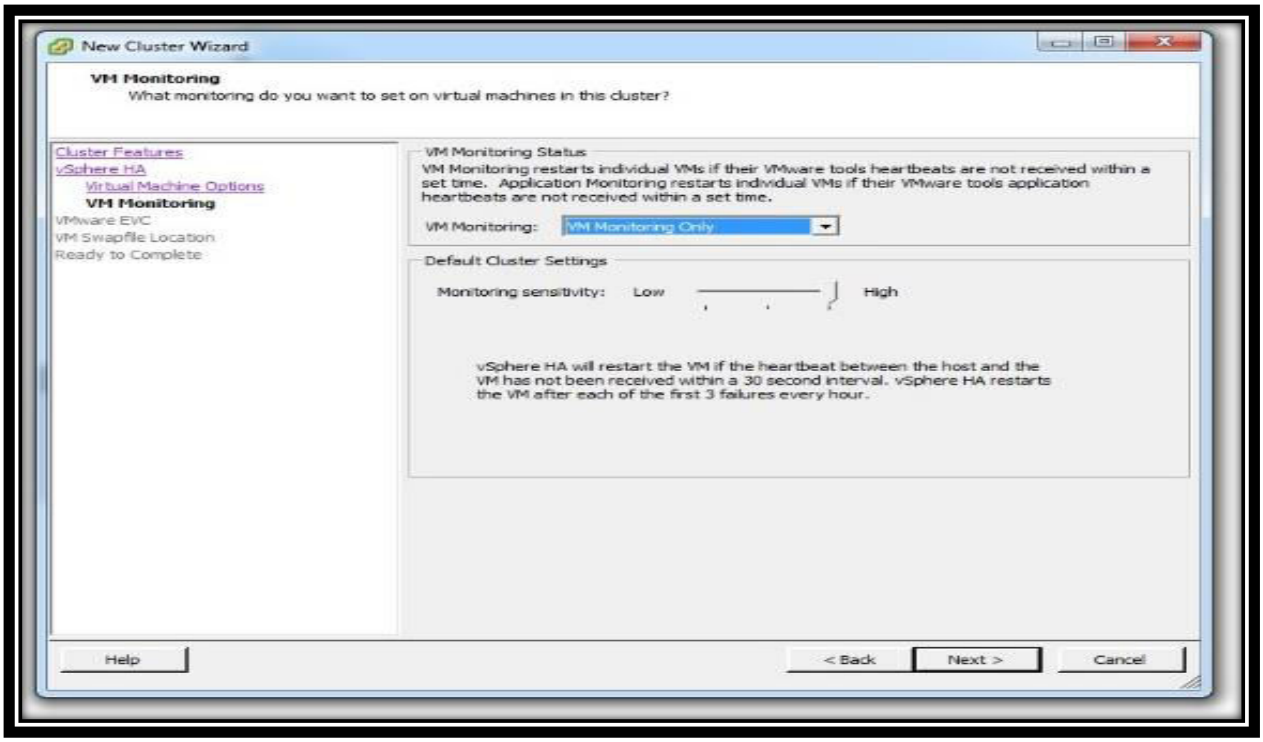
Step 3: Enable the host monitoring under the vSphere HA so that the ESXI hosts in the cluster can exchange the network heartbeat.



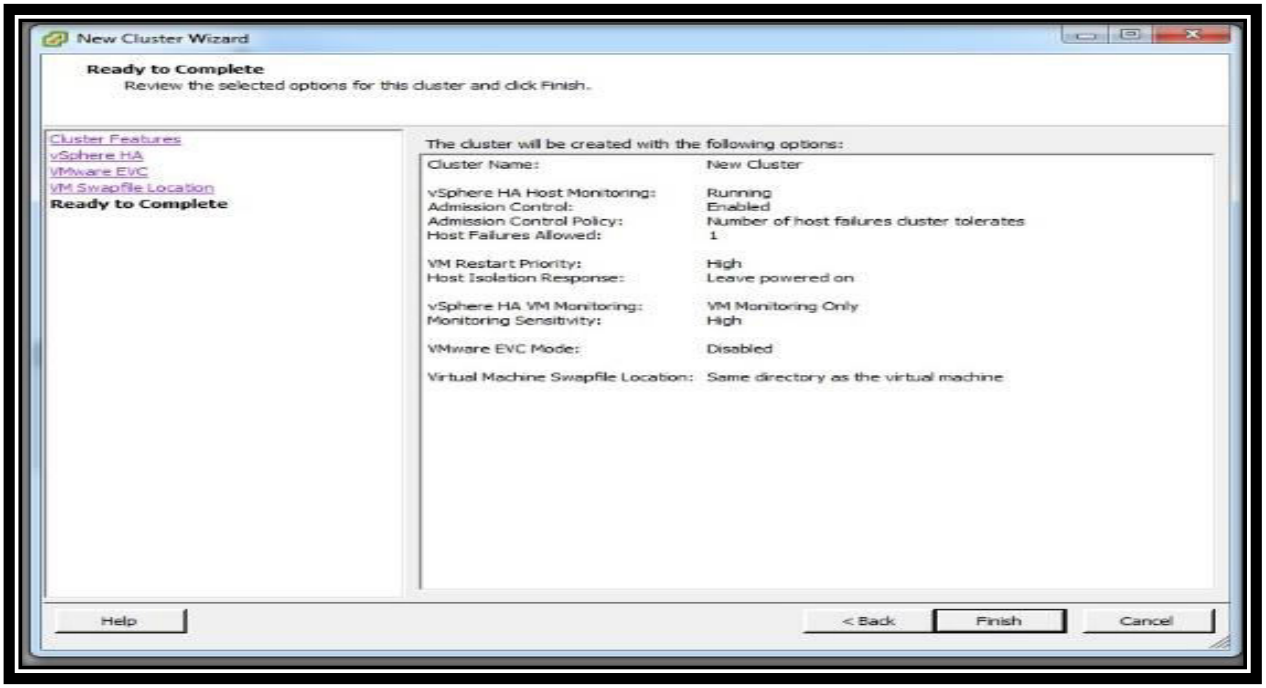
Step 4: In the virtual machine options set the VM restart priority high for the virtual machines.



Step 5: Under the VM monitoring select the "VM Monitoring Only" so that the virtual machines will restart if their VMware tools heartbeats are not received within a set time. Set the Monitor sensitivity to high it will restart the VM if the heartbeat between the host and VM has not been received within a 30 second interval.

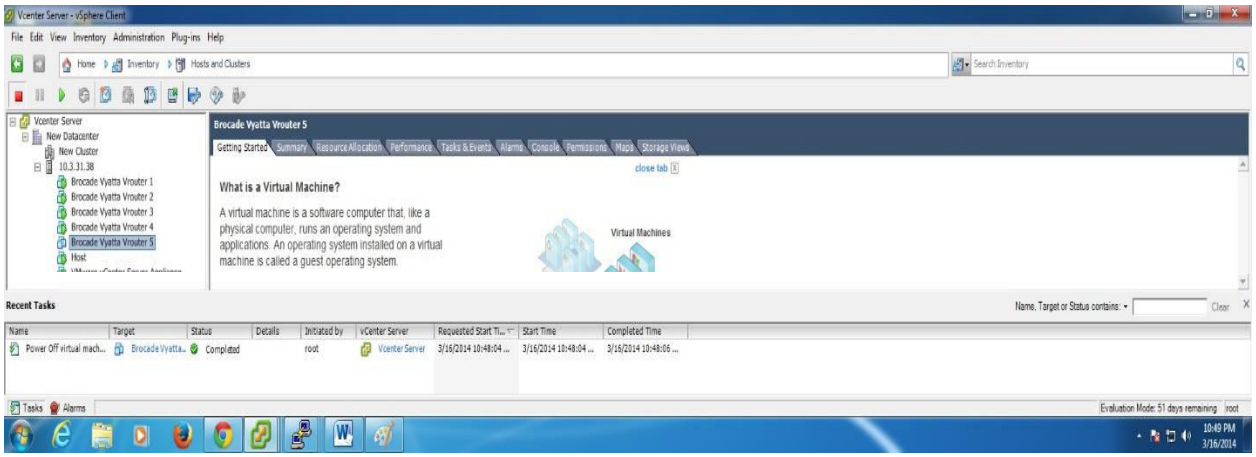


Step 6: Verify all the setting for the cluster creation and click finish for the cluster to be created with the following settings.

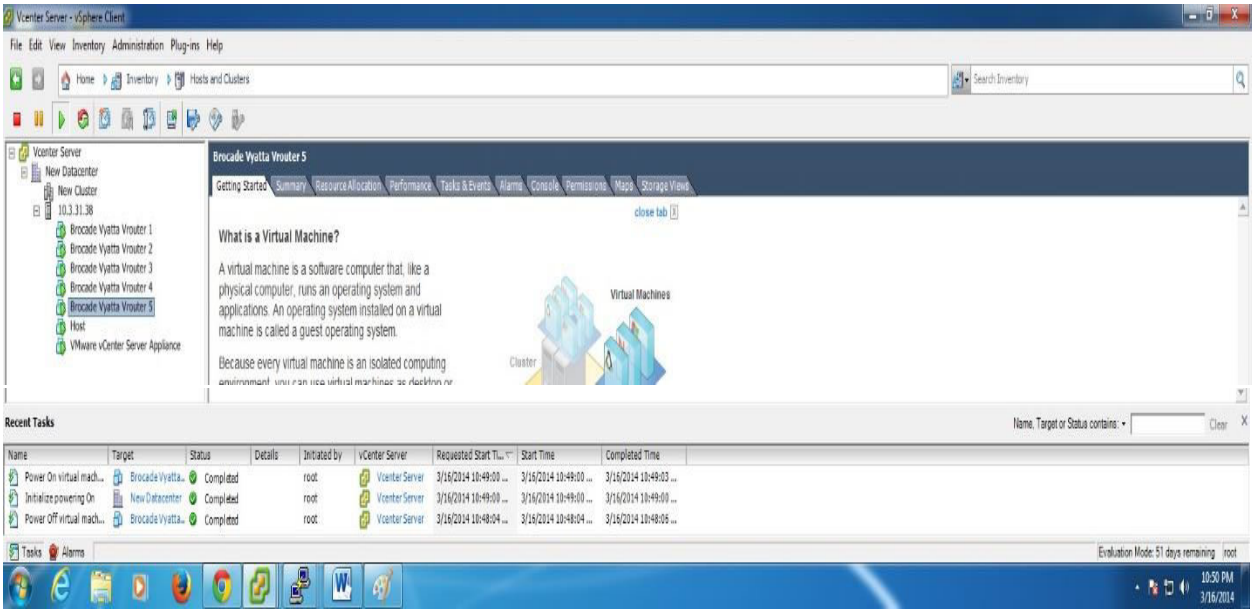


7.1.2 Failure Scenario of Virtual Machine Operating System

We have configured HA feature on five Brocade Vyatta virtual machines for high availability in case of their operating system failure. Let's examine the failure of virtual machine operating system by selecting any virtual machine and power off the virtual machine. A virtual machine has been powered off as shown below in the recent tasks as well



By configuring and enabling the high availability feature the heartbeat is sent between the virtual machine and vCenter server. If the operating system of virtual machine fails then the VMware tools which are already installed in the virtual machine would also fails which results in heartbeat no longer being sent to the vCenter server. After 30 seconds when the vCenter Server detects the heartbeat is no longer being received so it will automatically restart the virtual machines on the same physical server which will take around 60 seconds as shown below in the recent tasks.



7.2 Virtual Routing Redundancy Protocol (VRRP)

Virtual Router Redundancy Protocol (VRRP) is a first hop redundancy protocol for providing redundancy to the hosts in which virtual IP is used as a gateway IP address for the hosts to communicate. A virtual IP address is shared among the routers. In VRRP one router is set as a Master or active router and the other router is backup or standby. In case the Master router fails, the virtual IP address is mapped to the backup router's IP address and the backup router becomes active.

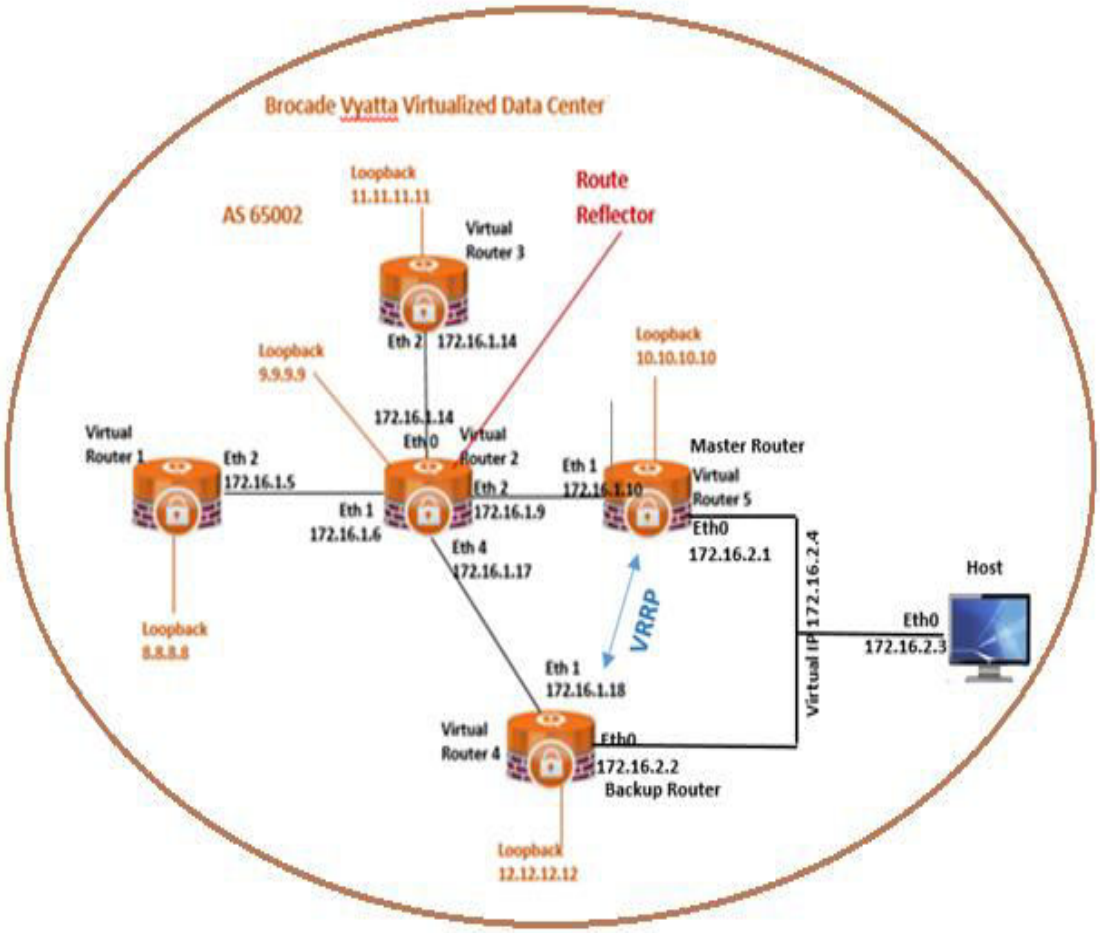


Figure 31: *Brocade Vyatta Virtualized Data Center Topology with VRRP*

We will configure VRRP in our Virtualized data center topology as given above for the provisioning of redundancy to the host. For configuring VRRP virtual router 4, virtual router 5 and the host are in the same subnet i-e 172.16.2.0/24. In the above topology virtual router 4 is Master router and virtual router 4 is backup.

7.2.1 Configuring VRRP

- Configure VRRP group on the same interfaces of virtual router 4 and virtual router 5. The group number should also be same.
- Configure higher priority for the Master router (virtual router 4) and lower for the backup router (virtual router 5).
- Configure same virtual IP address on both routers.
- Configure preempt as true to take effect the priority.
- On the host side assign the gateway address as virtual IP address of both routers.
- Configure iBGP between the virtual router 4 and host.
- Configure iBGP between the virtual router 5 and host as we are taking router as a host.

Verifying details at Master Route

```
vyatta@Vrouter4# run show vrrp detail
Use of uninitialized value in printf at /opt/vyatta/share/perl5/Vyatta/VRRP/OPMo
de.pm line 249.
--------------------------------------------------
Interface: eth0
--------------
  Group: 200
  ----------
  State:                        MASTER
  Last transition:              6d6h57m32s

  Source Address:
  Priority:                     200
  Advertisement interval:       1 sec
  Authentication type:          none
  Preempt:                      enabled

  VIP count:                    1
    172.16.2.4/24
```

Verifying details at Backup Router:

```
vyatta@Vrouter5# run show vrrp detail
Use of uninitialized value in printf at /opt/vyatta/share/perl5/Vyatta/VRRP/OPMo
de.pm line 249.
--------------------------------------------------
Interface: eth0
--------------
  Group: 200
  ----------
  State:                        BACKUP
  Last transition:              6d6h58m55s

  Master router:                172.16.2.1
  Master priority:              200

  Source Address:
  Priority:                     100
  Advertisement interval:       1 sec
  Authentication type:          none
  Preempt:                      enabled

  VIP count:                    1
    172.16.2.4/24
```

7.2.2 Testing VRRP

Initially, we will verify that which path the host is following for reaching the other virtual routers. As virtual router 4 is a Master router so host should follow the path via virtual router 4 as shown below:

Verifying path from host to virtual router 3:

```
vyatta@Host# run traceroute 11.11.11.11
traceroute to 11.11.11.11 (11.11.11.11), 30 hops max, 60 byte packets
 1  172.16.2.1 (172.16.2.1)  0.741 ms  0.706 ms  0.688 ms
 2  172.16.1.9 (172.16.1.9)  1.032 ms  1.054 ms  1.037 ms
 3  11.11.11.11 (11.11.11.11)  1.405 ms  1.394 ms  1.377 ms
```

Now, shutdown the virtual router 4 (Master router) and verify the path. Host should follow the other path via backup router when the Master router fails as shown below:

Verifying path from host to virtual router 3:

```
vyatta@Host# run traceroute 11.11.11.11
traceroute to 11.11.11.11 (11.11.11.11), 30 hops max, 60 byte packets
 1  172.16.2.2 (172.16.2.2)  0.899 ms  0.848 ms  0.834 ms
 2  172.16.1.17 (172.16.1.17)  1.364 ms  1.370 ms  1.358 ms
 3  11.11.11.11 (11.11.11.11)  1.700 ms  1.704 ms  1.636 ms
```

Verifying the status of Backup router after the failure of Master router:

It can be verified from the “run show VRRP detail” at virtual router 5 that when the master router fails the backup router becomes the master router to provide the redundancy.

```
vyatta@Vrouter5# run show vrrp detail
Use of uninitialized value in printf at /opt/vyatta/share/perl5/Vyatta/VRRP/OPMo
de.pm line 249.
------------------------------------------------
Interface: eth0
-------------
  Group: 200
  ----------
  State:                        MASTER
  Last transition:              9m9s

  Source Address:
  Priority:                     100
  Advertisement interval:       1 sec
  Authentication type:          none
  Preempt:                      enabled

  VIP count:                    1
    172.16.2.4/24
```

7.3 Configuring Juniper vSRX as Internet Firewall

In the final multivendor network topology, we will configure Juniper vSRX Services Gateway router as an Internet Firewall in which the internal network will be our Enterprise Data Center (AS 65001) and Virtualized Data Center (AS 65002) residing in the trusted zone. Whereas the Public Clouds in AS100 and AS200 will be in untrusted zone or Internet zone. We want our all network traffic to be allowed from internal network to Internet but we will block some network traffic from the Public Clouds (Internet) to the Internal network by applying firewall rules. We will also apply Network Address Translation (NAT) rules so that our internal network Private IP addresses that needs to get out to the Internet will be source NAT to a Public IP addresses.

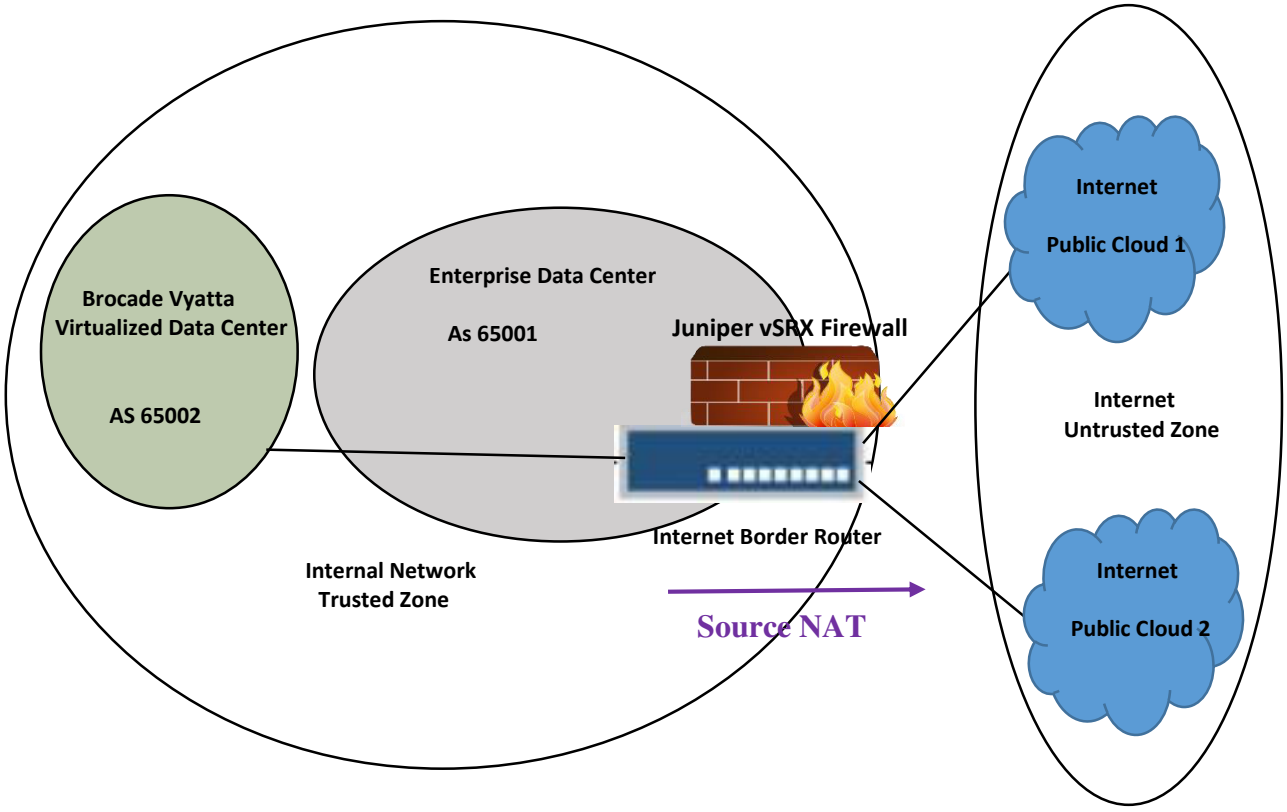


Figure 32: *Juniper vSRX as Internet Firewall*

Configuration Steps for Firewall Rule:

- Configure security zones for internal network (Trust) and for Internet (Untrust).
- Create Address book in the untrusted zone to match the source and destination IP addresses.
- Create firewall rule or policy to allow all traffic from trusted zone to untrusted zone.
- Create firewall rule or policy to allow Internet Networks (NET1-4) from untrusted zone to trusted zone and also block Internet Networks (NET5-10) from untrusted zone to trusted zone.

Configuration Steps for Source NAT:

- Configure loopback on Juniper vSRX router from the Public network subnet 100.100.100.0/24 which will be used for Source IP pool.
- Configure address pools for Source NAT.
- Configure Source NAT using IP pool so that all traffic from the trust zone to the untrust zone is translated to the source IP pool.

7.3.1 Configuration demo of Juniper vSRX Firewall and Source NAT

```
policy allow-internet-network-1 {
    match {
        source-address internet-network-1;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
policy allow-internet-network-2 {
    match {
        source-address internet-network-2;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
policy allow-internet-network-3 {
    match {
        source-address internet-network-3;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
policy allow-internet-network-4 {
    match {
        source-address internet-network-4;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
policy deny-internet-network-5 {
    match {
        source-address internet-network-5;
        destination-address any;
        application any;
    }
    then {
        deny;
    }
}
policy deny-internet-network-6 {
    match {
        source-address internet-network-6;
        destination-address any;
        application any;
    }
    then {
        deny;
    }
}
policy deny-internet-network-7 {
    match {
        source-address internet-network-7;
        destination-address any;
        application any;
    }
    then {
        deny;
    }
}
policy deny-internet-network-8 {
    match {
        source-address internet-network-8;
        destination-address any;
        application any;
    }
    then {
        deny;
    }
}
policy deny-internet-network-9 {
    match {
        source-address internet-network-9;
        destination-address any;
        application any;
    }
    then {
        deny;
    }
}
policy deny-internet-network-10 {
    match {
        source-address internet-network-10;
        destination-address any;
        application any;
    }
    then {
        deny;
    }
}
from-zone trust to-zone untrust {
    policy Allow-Access {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
default-policy {
    permit-all;
}
zones {
    security-zone trust {
        tcp-rst;

security-zone untrust {
    address-book {
        address internet-network-1 13.13.0.0/23;
        address internet-network-2 14.14.0.0/22;
        address internet-network-3 15.15.0.0/21;
        address internet-network-4 16.16.0.0/20;
        address internet-network-5 17.17.0.0/19;
        address internet-network-6 18.18.0.0/18;
        address internet-network-7 19.19.0.0/17;
        address internet-network-8 20.20.0.0/16;
        address internet-network-9 21.0.0.0/15;
        address internet-network-10 22.0.0.0/14;
    }
```

Verifying reachability from NET1 in Internet to Physical router1 of Enterprise Data Center:

```
Internet_Router_1#ping 1.1.1.1 source 13.13.13.13
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 13.13.13.13
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/10/11 ms
```

Verifying reachability from NET2 in Internet to virtual router3 of Virtualized Data Center:

```
Internet_Router_1#ping 11.11.11.11 source 14.14.14.14
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.11.11.11, timeout is 2 seconds:
Packet sent with a source address of 14.14.14.14
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/10 ms
```

As we can see from the above results that our firewall rules are working correctly because the above Internet networks were allowed to reach the internal network.

Verifying reachability from NET8 in Internet to Physical router2 of Enterprise Data Center:

```
Internet_Router_2#ping 2.2.2.2 source 20.20.20.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 20.20.20.20
.....
Success rate is 0 percent (0/5)
```

Verifying reachability from NET6 in Internet to Juniper router of Enterprise Data Center:

```
Internet_Router_2#ping 5.5.5.5 source 18.18.18.18
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
Packet sent with a source address of 18.18.18.18
.....
Success rate is 0 percent (0/5)
```

From the above results it is verified that the above Internet Networks are not reachable to the Enterprise Data Center because we have blocked that traffic by implementing firewall rules from untrusted zone to trusted zone.

7.3.3 Verification of Security NAT Source

```
root@Internet-border-router# run show security nat source pool all
Total pools: 1

Pool name          : Public_Nat_Range
Pool id            : 4
Routing instance   : default
Host address base  : 0.0.0.0
Port               : [1024, 63487]
Port overloading   : 1
Address assignment : no-paired
Total addresses    : 100
Translation hits   : 243
Address range                          Single Ports   Twin Ports
      100.100.100.1 - 100.100.100.100    0              0

Total rules: 1
Total referenced IPv4/IPv6 ip-prefixes: 7/0

source NAT rule: admin-access         Rule-set: Internet-Nat
  Rule-Id                   : 1
  Rule position             : 1
  From zone                 : trust
  To zone                   : untrust
  Match
    Source addresses        : 192.168.1.0     - 192.168.1.255
                              192.168.2.0     - 192.168.2.255
                              192.168.3.0     - 192.168.3.255
                              172.16.1.0      - 172.16.1.255
                              172.16.2.0      - 172.16.2.255
                              0.0.0.0         - 255.255.255.255
    Destination addresses   : 0.0.0.0         - 255.255.255.255
    Destination port        : 0               - 0
  Action                      : Public_Nat_Range
    Persistent NAT type       : N/A
    Persistent NAT mapping type : address-port-mapping
    Inactivity timeout        : 0
    Max session number        : 0
  Translation hits          : 243
    Successful sessions     : 228
```

8. SUMMARY AND CONCLUSION

Service providers as well as Enterprises are leveraging virtualization technologies to build highly efficient Data Centers to offer cloud based services. Some enterprises are moving applications to public service provider clouds for cost efficiencies. Others are hosting applications in the cloud for disaster recovery and/or cloud bursting purposes. Either of these scenarios requires a secure and dedicated access to cloud infrastructure and can be fulfilled by instantiating a virtual layer 3 VPN and/or routing gateway in the public cloud that can run on x86 commodity server platform. Such a deployment of virtual function is one of the common use cases of Network Functions Virtualization or NFV. In this project, we have demonstrated several additional NFV deployment use cases applicable in both service provider and enterprise environments.

The main objective of this project was to build an end-to-end Multivendor Hybrid Cloud network encompassing campus, DC, WAN and Public Internet domains. The network built for the project consisted of physical and virtual network devices from various hardware and software vendors and included functions such as routing, switching, firewall and NAT etc. We highlighted several deployment models and successfully demonstrated the coexistence of virtual and physical functions. In addition, we validated interoperability between virtual and traditional hardware based network functions using intranet and extranet routing protocols e.g. OSPF and BGP. In the topology, OSPF routing was chosen for internal routing both for physical and virtual router domains whole BGP peering relationship was established between physical as well as between virtual routers.

We also demonstrated the agility that NFV brings to the table through flexible virtual network function instantiation and rapid provisioning. Several design considerations and best practices including BGP scale via route reflector, optimal layer 2 switching between host virtual machines, redundancy and resiliency via VRRP were incorporated in the overall design of the network. Connectivity between each element of the network and appropriate traffic flow according to the defined routing policies for optimal path selection was thoroughly verified. In addition, security and high availability features were also tested especially for the traffic destined to or from Internet using firewall.

With respect to virtualization, we demonstrated the ease of implementation and management of virtualized infrastructure with VMware's ESXI 5.1 and deployed Brocade Vyatta 5400 vRouter, Juniper vSRX virtual firewall, Arista virtual switch (vSwitch), and Cisco Cloud Services Router CSR1000v.

In conclusion, Network Functions Virtualization (NFV) is emerging as a promising paradigm for network operators to build networks in a highly cost effective way leveraging commodity servers and extending virtualization beyond compute and storage to networking functions. In addition, with the architectural elasticity, the capacity of virtual functions can grow and shrink adapting to the traffic loads without needing fork lift upgrades. Network function virtualization is still in its infancy

but given that we have utilized virtual products from four different vendors in this project, it is highly encouraging to see that several vendors are starting to offer broad range of virtual network functions ranging from routers, switches, load balancers and firewalls etc. While major emphasis in this project was more around the operation, provisioning and management of virtual functions along with their coexistence and interoperability with physical devices, the future work could focus on the performance, throughput and scalability aspects of the virtual functions in some of the deployment models discussed in this project.

Bibliography & References:

Books

[1] Danielle Ruest and Nelson Ruest, "*Virtualization: A Beginner's Guide"*, McGraw-Hill, 2009.

[2] Scott Lowe, "*Mastering VMware vSphere 5",* John Wiley & Sons, 2011.

[3] Matthew Portnoy, "*Virtualization Essentials"*, John Wiley & Sons, 2012.

[4] Vyatta, Inc. "*Vyatta System Quick Start Guide"*, 2012.

[5] VMware, Inc. "*vSphere Networking ESXI 5.1, vCenter Server 5.1",* 2012.

[6] Cisco Systems, Inc. "*Cisco CSR1000v Series Cloud Services Router Software Configuration Guide",* 2014.

[7] Juniper Networks, Inc. "*Juniper Firefly Perimeter Getting Started Guide for VMware",* 2014.

[8] Arista Networks, Inc. *"Arista Networks User Manual Configuration Guide",* Arista EOS version 4.13.5F, 2014.

White Papers

[9] Brocade Communication Systems. (2014). "*Network Functions Virtualization. Cloud Networking: Scaling Data Centers and Connecting Users"* [White paper]. Retrieved from http://www.brocade.com/downloads/documents/white_papers/brocade-vyatta-cloud-networking-wp.pdf

[10] Juniper Networks, Inc. (2010). "*The Important Role of the Network in a Virtualized World"* [White paper]. Retrieved from http://www.juniper.net/Downloads/IDqePT8u_Juniper_Virtualization_Datacenter_wp%20(1).pdf

[11] The Green Grid (2009). "*Using Virtualization to Improve Data Center Efficiency"* [White paper]. Retrieved from http://www.thegreengrid.org/~/media/WhitePapers/White%20Paper%2019%20-%20Using%20Virtualization%20to%20Improve%20Data%20Center%20Efficiency.pdf?lang=en

[12] VMware, Inc. (2006). "*VMware Infrastructure Architecture"* [White paper]. Retrieved from https://www.vmware.com/pdf/vi_architecture_wp.pdf

Web Links

[13] ***http://www.vmware.com/ca/en/products/vsphere/features-network***

[14] ***http://blog.pluralsight.com/virtual-networking-101-understanding-vmware-networking***

[15] ***http://www.brocade.com/downloads/documents/data_sheets/product_data_sheets/brocade-vyatta-5400vrouter-ds.pdf***

[16] ***http://www.101datasolutions.co.uk/manufacturers/arista-networks/arista-veos/***

[17] ***http://www.juniper.net/us/en/local/pdf/solutionbriefs/3510497-en.pdf***

[18] ***http://www.juniper.net/us/en/local/pdf/case-studies/3520505-en.pdf***

[19] ***http://www.cisco.com/c/dam/en/us/products/collateral/routers/cloud-services-router-1000v-series/at-a-glance-c45-730864.pdf***

[20] ***http://www.etsi.org/technologies-clusters/technologies/689-network-functions-virtualisation?highlight=YToxOntpOjA7czozOiJuZnYiO30=***

[21] ***http://www.sdncentral.com/whats-network-functions-virtualization-nfv/***

[22] ***http://www.vmware.com/files/pdf/press-kit/vmware-nsx-media-backgrounder.pdf***