UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

DESIGN AND DEPLOYMENT OF REAL SCENARIOS OF TCP/IP NETWORKING AND IT SECURITY FOR SOFTWARE DEFINED NETWORKS WITH NEXT GENERATION TOOLS

A Master's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Clàudia Palomas Riera

In partial fulfilment

of the requirements for the degree of

MASTER IN TELECOMMUNICATIONS ENGINEERING

Advisor: Esparza Martín, Óscar

Barcelona, May 2018

Title of the thesis: Design and Deployment of Real scenarios of TCP/IP Networking and IT Security for Software Defined Networks with Next Generation Tools.

Author: Clàudia Palomas Riera

Advisor: Óscar Esparza Martín

Abstract

This thesis is about NSX, a Software Defined tool provided by VMware, to deploy and design virtual networks. The recent growth in the marked pushed companies to invest and use this kind of technology. This thesis explains three main NSX concepts and the basis to perform some deployments. Some use cases regarding networking and security are included in this document. The purpose of these use cases is to use them in real scenarios, which is the main purpose of the thesis. The budget to deploy these use cases is included as an estimation about how much a project like this would cost for the company. Finally, there are some conclusions and tips for best practices.

Per la Judit i en Roger. Sense ells, res hauria estat possible.

Acknowledgements

First of all, I have to mention Judit, my mother, the person who encouraged and supported me in everything of my life.

Roger, thank you for respecting my decisions, for waiting minutes and minutes before to leave because I want to finish something, for your good tips. Also, for encourage me in my life, to be successful no matter what I try.

Óscar, my advisor and a friend, thank you for being the first one to teach me SDN, for being there when I have questions, need help with new concepts and to correct my mistakes.

Anja, thank you for your positive attitude and for encouraging me at all times.

Lluís A., one of my bosses, thank you for providing me a server and NSX licenses, moreover, for hire me for this job and project.

Raúl R., my actual boss, thank you for having patience with me, my project time and for providing me experience.

Albert F., my colleague, thank you for helping me in this NSX world, for teaching me.

Juanma, my systems colleague, thank you for helping me in the initial steps of this project, for install with me Delly (server) and deploying the hypervisor together.

Gerard, Eric, Arnau, thank you for talking to me when I am obfuscated and helping me in the office.

Finally, thank the other members of CSIA.

Revision history and approval record

Revision	Date	Purpose
0	20/03/2018	Document creation
1	18/05/2018	Document revision
2	20/05/2018	Document revision

Written by:		Reviewed and approved by:	
Date	20/03/2018	Date	20/05/2018
Name	Clàudia Palomas Riera	Name	Óscar Esparza Martín
Position	Project Author	Position	Project Supervisor

Table of contents

The table of contents must be given in detail. Each chapter and main section of the thesis must be listed in the "Table of Contents" and duly paginated to facilitate the location of a particular text.

List of Figures

List of Tables

1. Introduction

The main purpose of this thesis is to learn and experiment how SDN behaves, in particular, NSX. Moreover, to know its advantages and of course, drawbacks. For my company, it is very important to be skilled in networking but also in SDN, which is probably the future of networking. This project results on a hands-on experience with NSX, which allows to participate in other kind of projects.

The idea of this project started in the second year of the master, with Communication Networks (traditional networks) and Overlay Networks; the fusion of this subjects results in how networks will be in 5 years: hybrid networks made of traditional aspects and virtual ones.

1.1. Requirements and specifications

The principal requirements and specifications to perform this project are described below. It is required some hardware and software.

1.1.1. Hardware

The hardware required it is a physical server, for example, Dell PowerEdge R420 Server Series. The one used in this project has the following specifications: 4x600GB Storage, 128GB Memory and 8+8 CPU Cores. It has to be mentioned, that although the hardware is not the one in charge of the functions of the virtual network, it has to be well dimensioned; on the contrary, the virtual network will be slow and things will not work properly. This is something that happened and will be explained.

It is also needed a router, a device that provides the connectivity between the server and the PC used to design and control everything.

1.1.2. Software

About the software, it is required an hypervisor, the licenses and the virtual machines that will be used for the network.

Licenses

The licenses required for this project are mentioned in the table below specifying the license name and where it is used:

Element to be Licensed	License Name
ESXi	VS6-EPL-C VMware vSphere 6 Enterprise Plus - 32 CPU
vCenter	VCS6-STD-C VMware vCenter Server 6 Standard for vSphere 6 - 2 instance key
NSX	NX-ENT-C VMware NSX Enterprise per Processor - 16 CPUs

Table 1. Licenses required for this project.

Each ESXi needs to be license, in this project there are 3 ESXi but no more than one license is needed due to the number of CPUs does not reach 32.

For the case of vCenter, each one has to be licensed. For this project there are 2 vCenter, so, a 2-instance key is required.

Finally, NSX, the element that provides networking and security virtualization, has three kinds of licenses: Standard, Advanced or Enterprise depending on the options that the network has.

The following table shows detailed the 3 available license options:

Features	Standard	Advanced	Enterprise
Distributed switching and routing	X	X	X
NSX Edge Firewall	X	X	X
NAT	X	X	X
SW L2 bridging to physical environment	X	X	X
Dynamic routing with EMP (active-active)	X	X	X
API-driven automation	X	X	X
Integration with vRealize and OpenStack	X	X	X
Automtion of security policy with vRealize		X	X
NSX edge load balancing		X	X
Distributed firewall		X	X

Integration with Active Directory	X	X
Server activity monitoring	X	X
Service insertion (third-party integration)	X	X
Cross vCenter NSX		X
Multi-site NSX optimizations		X
VPN (IPSec and SSL)		X
Remote Gateway		X
Integration with hardware VTEPs		X

Table 2. Differences between vCenter licenses.

It must be mentioned that although in this project requires Cross vCenter NSX, for this reason the Enterprise this license is required.

Hypervisor

The hypervisor used in this project for use cases and deployment it is the one provided by VMware. The reason for choosing this hypervisor and not another one it was to test the full VMware proposal for SDN.

Virtual Machines

The virtual machines were provided by a company's department. Usually, in VMware projects, the engineer that designs the network it is not in charge of building or preparing the virtual machines that reside inside of the network.

1.2. Statement of purpose

The main purpose of this SDN project is to investigate about SDN and to learn how to build networks with this technology using VMware. There are many other options for current SDN networking but, the tool used was decided by a company's boss.

The reason for learning VMware is that there are projects where engineers must know how to use it properly. On the contrary, there are other projects that require cisco ACI, another SDN company, instead of VMware.

The principal objectives are:

- Perform a SDN deployment.
- Learn NSX from VMware.
- Develop and test real scenarios/use cases.
- Extract conclusions.
- Write a best practices document for the company.

Finally, it is important to say that as virtual networks work in a different way from the traditional ones, the acquisition of a new way of thinking it is another objective of the project.

1.3. Methods and procedures

This thesis it is not a continuation of another project. All the methods and deployments have been performed by the author.

There are future lines for this project which will be detailed in a section in last pages of the document.

1.4. Work plan and Gantt diagram

The work plan for this project was difficult, not only for the project itself, but also the job. It must be said that job hours are approximately 6 per day, so the combination with the project is complicated. Even though, the project had some scheduled tasks that can be appreciated in the Gantt diagram, which is included as a file in this section.

In order to better comprehend the Gantt some comments are made:

- The server was ubicated inside the office without remote connection.
- There are weeks without anything that correspond to holidays.
- The server had an expiration date of 6 months, so, in the third week of February was bounced.
- During critical job moments, the job had priority.

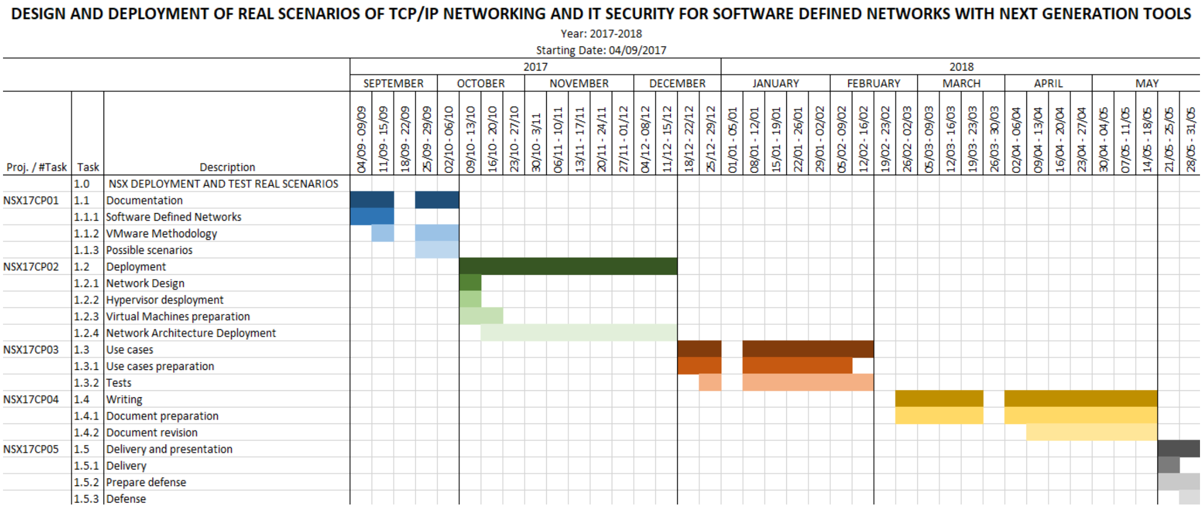


Figure 1. Gantt Diagram.

Gantt.xlsb.xlsx

1.5. Deviations and eventualities arisen

In a project there are always deviations and eventualities. As it is commented in the previous section, the Gantt one, this project was combined with an almost full-time job.

The first deviation is temporal; theoretically, the project had to be presented in January but due to the amount of hours in the office the project had to be prorogued. This temporal deviation was caused to because in the work some troubles have been experienced, problems that required to be solved immediately.

Moreover, the fact that the server was rented supposed to have limited time to perform tests or develop other use cases.

Finally, about deviations regarding specific use cases were related with licenses. For example, one of the Ideas was to include AirWatch from VMware but this supposed to acquire an annual license that costs about 2000€. Another idea was to include in a security section a Trend Micro block, but to get the licenses was difficult because they had to be bought. The same happened with third parties: although NSX has a distributed firewall and balancers, one idea was to include a Palo Alto Networks (firewall) and a F5 (load balancer module) but again, for license stuff could not be implemented.

1.6. Thesis Chapters

To conclude the introduction, here there is a brief explanation of each topic that the thesis contains.

1.6.1. State Of The Art

State of the Art it is the most general part. It explains the basis of SDN networks, with its pros and cons. Furthermore, the different market competitors (CISCO, VMware), besides, the companies that uses these technologies.

In addition to that, State of the art contains an explanation of main NSX (VMware) components with the purpose to comprehend the global configurations and system.

1.6.2. Methodology and procedures

This section explains how network was built, moreover some tips regarding on basic configurations.

1.6.3. Use cases

The use cases chapter is the one with the “experiments”. There it can be found different techniques regarding networking and security. For example, different ways to implement microsegmentation or how to add a load balancer in a SDN network. It also contains some conclusions.

1.6.4. Budget

The budget estimates the cost of the implantation of NSX in a company.

2. State of the art

As mentioned in the abstract, the main technology used in this thesis is SDN, which means Software Defined Networks; in particular, all deployments and use cases have been done with VMware NSX.

This section presents different topics that are required to comprehend the work done, these topics are:

- SDN.
- Competitors and Position of Technology players within a specific market.
- Companies using SDN technologies.
- VMware: NSX main ideas, pros, cons, challenges and research.

2.1. SDN

2.1.1. What is SDN?

SDN, known as Software Defined Networks, is basically the decoupling of the control plane and data plane having the network intelligence and state logically centralized in the underlaying network infrastructure, by contrast of the traditional model which is decentralized and complex. Besides, the underlaying network infrastructure is abstracted from the applications. Figure 2 shows the basic SDN Model:

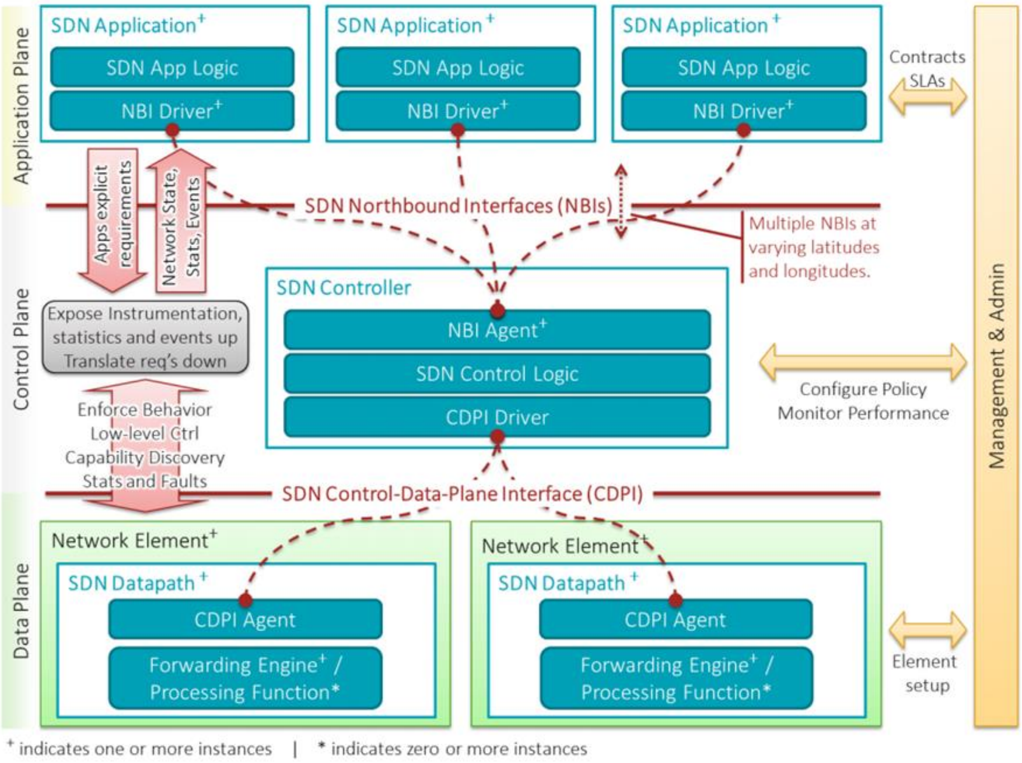


Figure 2. SDN Model.

By way of introduction and a better comprehension, the different planes and the elements that compose SDN are going to be described briefly:

- Application Plane: The application plane is where the applications reside, they request their network requirements and desired network behavior to the SDN Controller via NBIs (Northbound Interfaces).
- Control Plane: The control plane is in charge to manage the network. It is composed by the SDN Controller.
    - SDN Controller: it is the intelligence of the network, it provides a centralized view of the overall network and translates the demands from the SDN Application Layer and relays them to the SDN Datapaths. As a result, the Application layer has an abstract view of the network.
- Data Plane: The data plane is where logical network devices reside. These devices main function is to manage the forward and processing data. It is mainly composed by the SDN Datapaths.

SDN appears due to the need of having an architecture manageable and adaptable, cost-effective and dynamic. So, the aim of that is to make network engineers and administrators life easy, in terms of architecture, troubleshooting and to respond quickly and effectiveness to changing business requirements. For this simple reason, in SDN a network administrator can shape traffic from a centralized control console without having to touch individual switches, and can deliver services to wherever they are needed in the network, moreover, the admin does not have to concern to what specific devices a server or other hardware components are connected to. The main characteristics of this kind of architectures are:

- Agile: Decouple control from forwarding allowing administrators dynamically to adjust the network to the traffic demands.
- Centrally Managed: As it is mentioned, the SDN Controller is the network intelligence.
- Programmatically configured: Managers can configure, manage, optimize and secure the network easily using automated SDN programs.
- Directly programmable: Network control is directly programmable because of the decoupling of it from the forwarding functions.

2.1.2. PROS AND CONS

This innovative technology approach offers some pros and cons that are going to be listed to clarify the advantages and disadvantages:

PROS	CONS
▪ **Centralized network provisioning** ▪ **Augmented automation** ▪ **Enhanced security** ▪ **Reduced operating costs** ▪ **Reduced hardware management and costs** ▪ **Cloud-ready infrastructure** ▪ **Network Flexibility** ▪ **Simplified management wth a central management console** ▪ **Easy Troubleshooting** ▪ **Microsegmentation** ▪ **Dynamic scaling** ▪ **White-box network**	▪ Transition and reconfiguration from legacy network to the SDN one ▪ Single point of failure ▪ Adapt the physical network to the virtual one, in terms of that virtual grows faster. ▪ Lack generic diversity: if a bug takes down the primary controller it may take down the others. ▪ High volumes of traffic between virtual networks and physical systems ▪ Not mature enough ▪ New trainings for staff ▪ New management tools

Table 3. SDN Advantages and Disadvantages.

2.1.3. SDN Market Evolution

There are many studies and predictions about the SDN market and all of them coincides in one premise: SDN is growing. Markets are evolving constantly but when it comes to technology, changes are faster and bigger if the technology has benefits and interests for the end user.

The most common example is the Moore's Law applied in the transistors count, a known image for Telecommunications students.

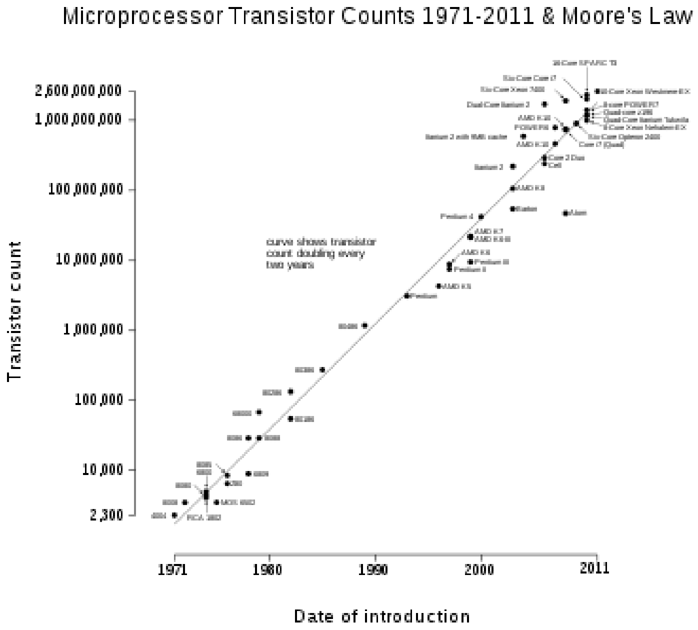


Figure 3. Microprocessor Transistor Counts 1971-2011 Moore's Law.

With SDN happens something similar, not only in terms of invests for this technology but also the know, trust and technology adaptation. The number of companies of SDN in 2009 was zero and, nowadays it is higher than 200; this is reflected in the amount of money that refers to SDN. By the 2018, the SDN Market Share is about $35 billion but it is estimated to cross $88 billion by 2024.

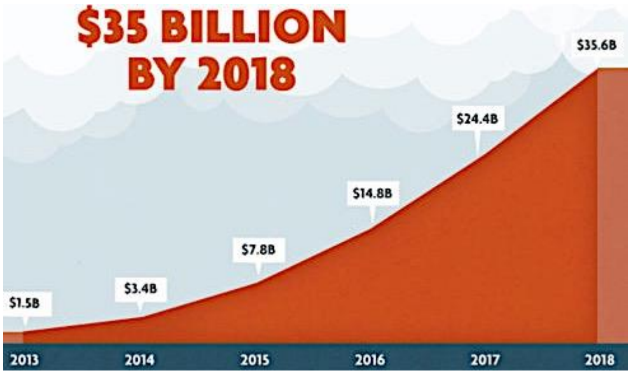


Figure 4. SDN Market Share Evolution.

As commented before, the know, trust and technology adaptation has been changed too, the paradigm 4 years ago of SDN adoption it is shown in the Figure 5:

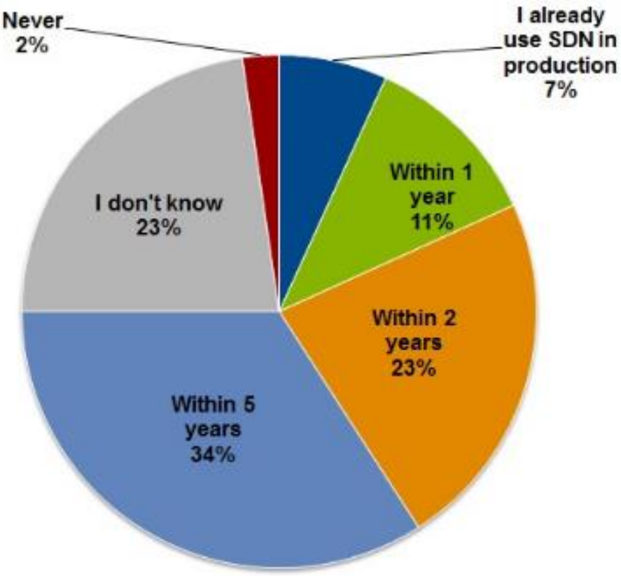


Figure 5. SDN questionnaire in 2014.

While in 2014 the current percentage of companies that already had SDN in production was 7%, 3 years later this value grew to the X%. Furthermore, the number of companies that pretended to use this technology was about 67%, subsequently, recent studies predicts that in 2019/2020 almost x% of the companies will have SDN partial or totally in their network. Nowadays IT business/companies now what SDN is.

Thanks to the increase of trust, mature and benefits of SDN, companies are more and more deploying this technology and leaving traditional networking. The tendency of that is the following one:

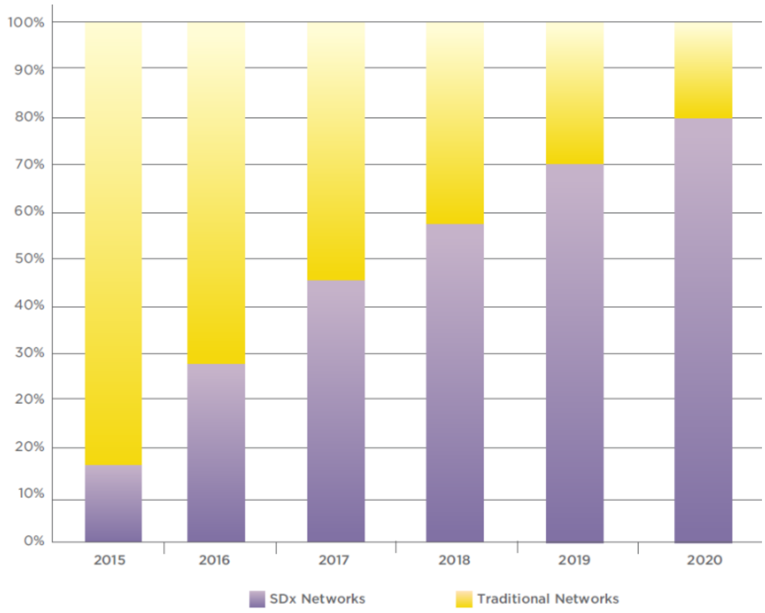


Figure 6. Evolution of Traditional vs SDx Networks.

The figure above reflects the incremental tendency of SDx Networks. As a result, in a couple of years the actual roles will be exchanged and the kind of services too.

2.1.4. SDN Standards

The significant impact and the current grown of SDN networks forces IEEE to standardize it; in fact, it must be said that SDN is highly related with NFV (Network Functions Virtualization) which is out of scope of this project.

For this reason, the IEEE is working on these standards. From the IEEE organization, it can be extracted which projects and research groups are involved in this business.

The current standardization projects are the following ones:

- IEEE P1903.1 - Standard for Content Delivery Protocols of Next Generation Service Overlay Network (NGSON).
- IEEE P1913.1 - Software-Defined Quantum Communication.
- IEEE P1915.1 - Security for Virtualized Environments.
- IEEE P1916.1 - Performance for Virtualized Environments.
- IEEE P1917.1 - Reliability for Virtualized Environments.
- IEEE P1921.1 - Software-Defined Networking Bootstrapping Procedures.
- IEEE P1930.1 - SDN based Middleware for Control and Management of Networks.
- IEEE P802.1CF - Recommended Practice for Network Reference Model and Functional Description of IEEE 802 Access Network.

Most of these standards do not focus in network performance, on the contrary, they are intended for satisfying customers, operators and improve network experience.

2.2. SDN Competitors and Market Position

As it is said, another door opens when one is shut, there are some traditional network companies that are involved in this virtual network world although there are others that born due to SDN. There are many ones, however, this section describes the Top SDN vendors and products.

But first, the magic quadrant by Gartner it is going to be presented:

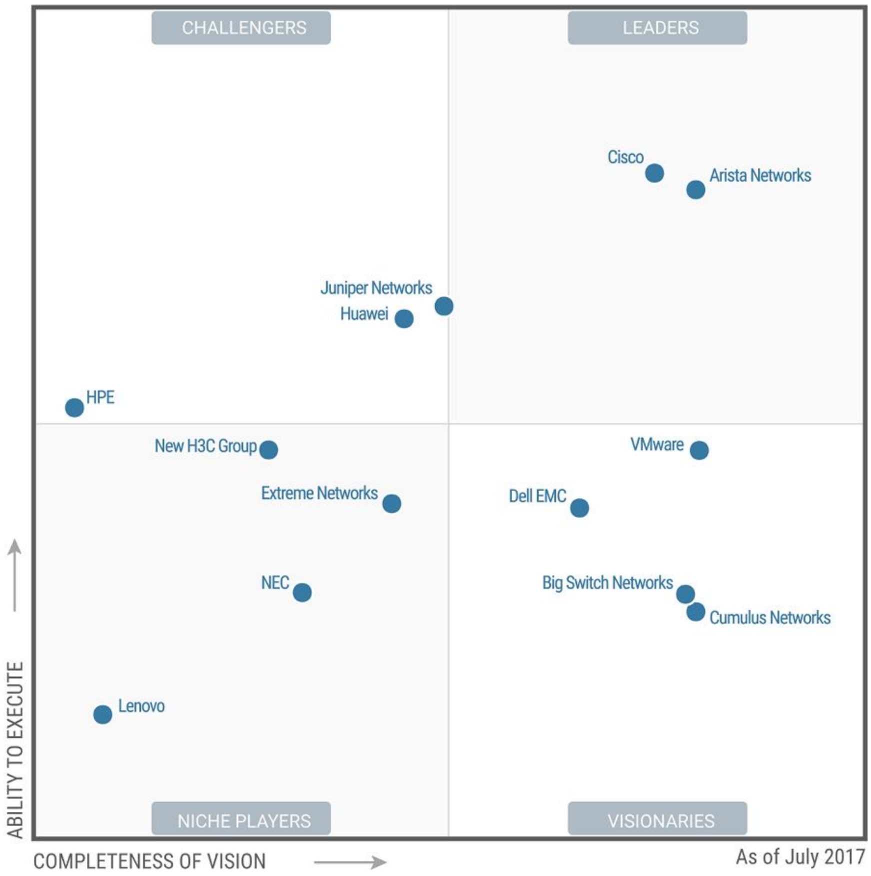


Figure 7. Gartner Magic Quadrant.

The picture above reveals which vendors are the most visionaries and leaders in SDN market. The Gartner Magic Quadrant it is a tool used by networking managers to analyze and apprise the different options that they have.

It can be clearly seen that VMware and CISCO are well positioned as visionaries and leaders, which means that they are executing well their vision and they are well positioned for tomorrow; moreover, they are the most known vendors in this sector, they have approximately the same amount of networks deployed in SDN besides VMware revenue is higher.

As visionaries, cumulus networks and VMware are the ones that are in the top of this section, they are characterized for having a good vision about where the market is going or a vision for changing market rules although they are not leaders because they do not yet execute well. It is true that before there is commented that VMware is executing well, from the Gartner they are almost leaders.

Furthermore, it can be seen that Lenovo, a well-known company, and NEC remain in the Niche Players section, due to they work in a small segment of the whole SDN.

Finally, the challengers, those who are doing a good work, designing and developing interesting ideas even though they do not comprehend the market tendency.

The Top SDN companies in the market are going to be mentioned with their most featured aspects.

2.2.1. CISCO

Product: ACI (Application Centric Infrastructure)

Figure 8. CISCO Logo.

They stand out for their custom chips that gather application traffic flow data. Their major fact is the innovation around the SDN underlays.

2.2.2. VMware

Product: NSX (Network Virtualization and Security Platform)

Figure 9. NSX Logo.

The network virtualization idea has become a reality thanks to VMware, the ones that nowadays are leaders in terms of revenue. The biggest hurdle for VMware is the fact that physical network in the data center is still important and their focus is the overlays concept.

2.2.3. Huawei

Product: Huawei SDN/NFV: software and hardware.

Figure 10. Huawei Logo.

They offer the full product which includes software and hardware. Their purpose is that client can use their software, orchestration, virtualization and servers of storage. Moreover, Huawei has a very strong R&D team that works in new components, with silicon, to improve hardware. For example, a programmable switch in Silicon.

Huawei’s main drawback is their market position; while in China they are a parent entity in North America they are not well positioned.

2.2.4. Hewlett Packard Enterprise

Product: SDN Applications, SDN Open Ecosystem, Cloud Solution.

Figure 11. HPE Logo.

HPE offers a wide range of SDN applications that can be easy and quickly download on SDN environments with the purpose of testing and live deployment. They define themselves as the first SDN App Store. Additionally, HPE offers network optimizer and security and cloud orchestrators. Likewise, the HPE Helion OpenStack accelerates the transition from the traditional network to hybrid networking.

2.2.5. Juniper Networks

Product: NorthStar Controller and NFX Series Network Services Platform.

Figure 12. Juniper Networks Logo.

The first vendor that open source their controller; Juniper Networks use an open source model for overlays. They want to promote and provide an open development environment. They use a bare metal switching which disaggregates the switch from the software. As a result, the main products are the NorthStar Controller, a programmable network that provides visibility and optimization giving service providers the freedom to customize services and deliver them consistency; also, the commented NFX that provides secure, standards-compliant CPE devices that simplify the creation and delivery of network services.

2.2.6. Cumulus Networks

Product: Cumulus Linux

Figure 13. Cumulus Networks Logo.

Three years ago, Cumulus was selected for being one of the best startups. Their basis are SDN solutions based on bare metal switching and Linux. They work with the Dev Ops, IT folks, providing a switch OS that's familiar to many Linux developers with the purpose of having a disaggregated switching.

Cumulus Linux is an open network operating system that allows to customize, automate and scale using web-scale principles.

2.2.7. Red Hat

Product: Red Hat Enterprise Virtualization.

Figure 14. Red Hat Logo.

Red Hat offers the possibility to manage a network, servers and workstations that had been virtualized, from one interface using their Red Had Linux Enterprise and their virtualization tool. It is very useful for those companies that have already Red Had Linux inside.

The vendors described in this section are, in my opinion, the most popular or innovative that provide better solutions of the market. There are many others like: Arista Networks, Big Switch Networks, Dell EMC, Extreme Networks, New H3C Group, NEC and Lenovo, which are doing a magnificent work in SDN world.

2.2.8. Arista Networks

Product: Software Driven Cloud Networking.

Figure 15. Arista Networks Logo.

Arista offers a wide range of products, the one based in SDN is the SDCN (Software Driven Cloud Networking), combining the principles of cloud computing and network virtualization. This fact simplifies management and provisioning providing high speed.

2.3. <u>Companies with a network based in SDN</u>

An effective way to know if SDN is accepted and implemented for the managers and technicians is to perform a market research, not only based in revenues, but also investigating which companies use an SDN model or network.

- Big Fish Games (CISCO ACI)
- AT&T (VMware NSX)
- Check Point Software Technologies (VMware NSX)
- Brookhaven National Laboratory (Red Hat Enterprise Virtualization)

From the above list, it can be extracted that this technology does not only concern the Telecommunication industry, financial services, computer software and others use it. It must be mentioned, that some companies are now betting for combined SDN structures: for example, CISCO ACI as a base and NSX as a software. The following image shows different industries using NSX VMware:

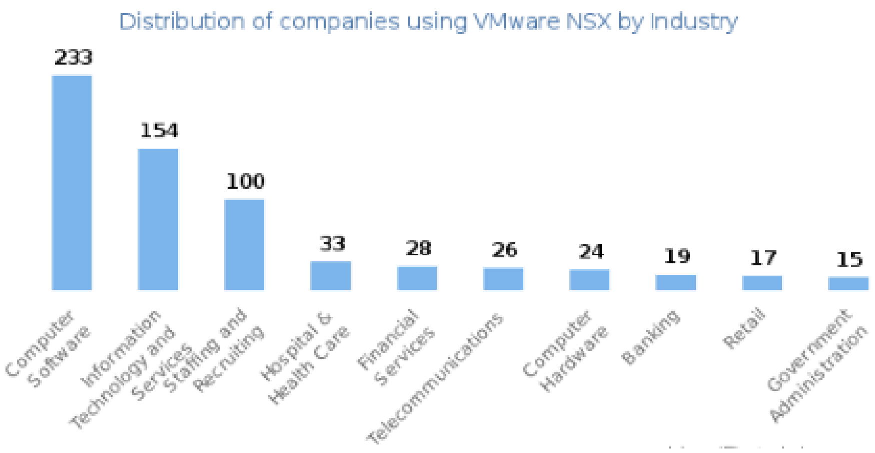


Figure 16. Distribution of companies using VMware NSX by Industry.

The geographical distribution for this technology it is still immature. The most companies that use SDN are not quite extended around the globe, it is commonly used in United States. The next figure represents the geo distribution for NSX, although CISCO and other vendors have a similar tendency.

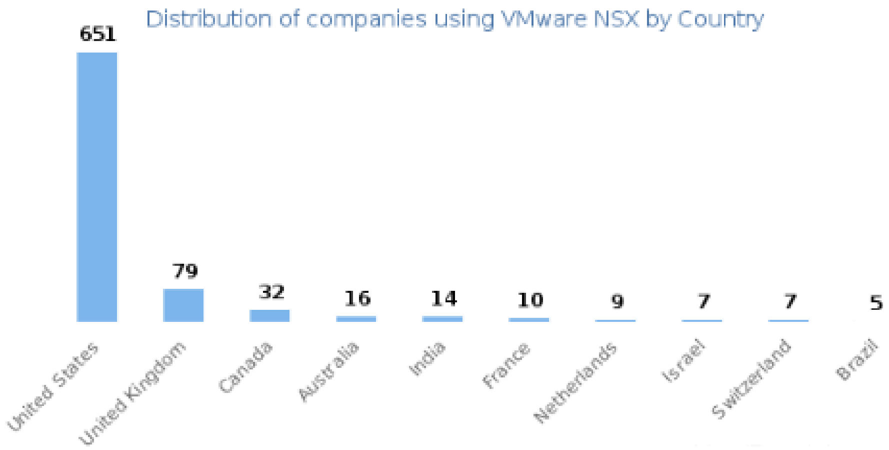


Figure 17 Distribution of companies using VMware NSX by Country.

2.4. VMware

What this section pretends is to introduce the audience about VMware; VMware does not only offer SDN technology, they offer a wide range of possibilities among the networking world. Moreover, this point explains the premises of NSX, its main ideas, the pros and cons, challenges and the research that they are currently performing.

2.4.1. NSX

NSX is a network virtualization platform designed to build a set of logical networking and security services in a SDDC. These services are: logical switching, logical routing, logical firewall, logical load balancer and logical VPN which will be deployed and explained in the current project.

NSX is service inside vSphere; vSphere is main composed by an Hypervisor (ESXi) and a Centralized management component (vCenter Server). Prior to analyze and explain NSX itself, there are some structures that have to be explained because NSX is built over there.

Architecture

To perceive the main differences between the Physical structure and the virtual one the following image is presented:

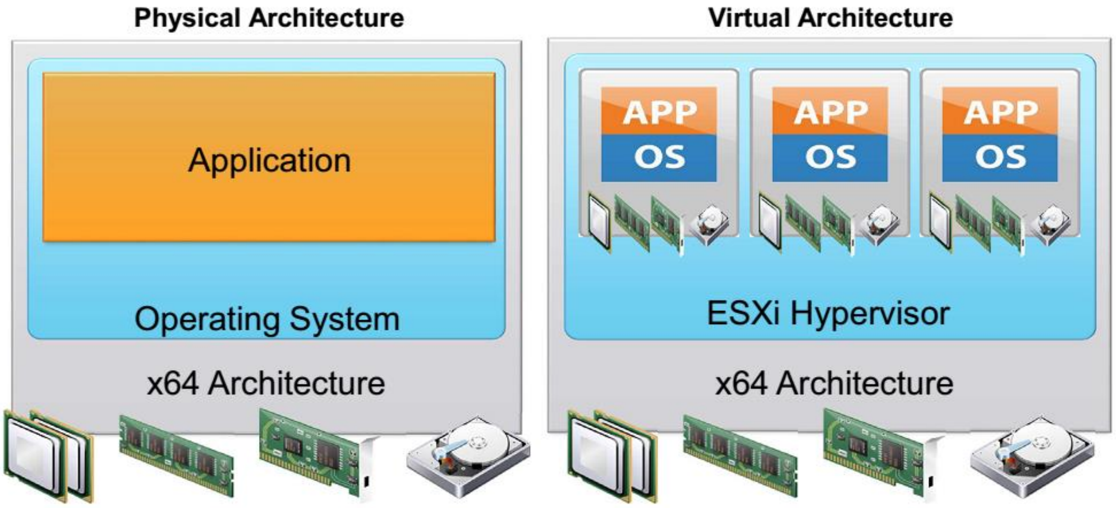


Figure 18. Differences between the Physical structure and the virtual one.

It can be seen in the above figure that, while in the physical architecture the Operating System is installed on the physical hardware, in the Virtual one, the Hypervisor is installed over the hardware and then, over that, the Virtual machines with diverse kinds of OS and applications. Basically, the Hypervisor is the lowest level, the requirement number one to start deploying virtual machines. This results in a complete control of the underlying hardware and the possibility of managing and sharing that physical resources. It is basically a decoupling between the OS and the hardware than in the traditional structure it is not independent. The next deploying step is to deploy the different ESXi hosts; this will be explained later in this project.

Using an hypervisor, several VMs can run in parallel and, although they share the same physical resources, the ones assigned to a VM can be different.

The principal disadvantages of the traditional structure are the mobility of the workload, backups, disaster recovery, restore and cost; on the contrary, the virtual architecture improves all the mentioned disadvantages and, moreover, the cost is lower and the speed is higher.

Centralized Management

It has been mentioned before that SDN and VMware has a centralized management and distributed services (vMotion, High Availability…) which are provided by the vCenter Server. vCenter Server is made by several components:

- Platform Services Controller: it includes infrastructure components, license and certificate management and lookup service.
- Database Server
- Application Server
- vSphere Web Client

To have an order about what an vCenter server can handle, some general data are going to be presented. A vCenter server can support 1.000 ESXi hosts and up to 10.000 powered VMs.

The major distributed services, which reinforces the central management idea, are:

- VMware vSphere Distributed Switches: consistency of network management and configuration. It provides the central management across multiple ESXi.
- VMware vSphere vMotion: It allows to move a VM from one physical server to another without being unavailable and running perfectly, there is no downtime during the migration process what means the VM preserves its connections.
- VMware vSphere HA: In case of ESXi failure, VMs in that ESXi are automatically restarted and before that, moved to another available ESXi.

NSX Main Elements

Before start deploying, some elements must be presented and studied in order to comprehend how things work in NSX. This section it is purely theoretical, further on there is a section in which deployments are detailed.

Logical Switch

A virtual switch is a switch that is not a physical one, it runs over software. One of its main functions is to provide connectivity between VMs and hypervisor to the physical network. It also allows several types of traffic, as well as, Management, IP Storage, Fault Tolerance Logging, vMotion…

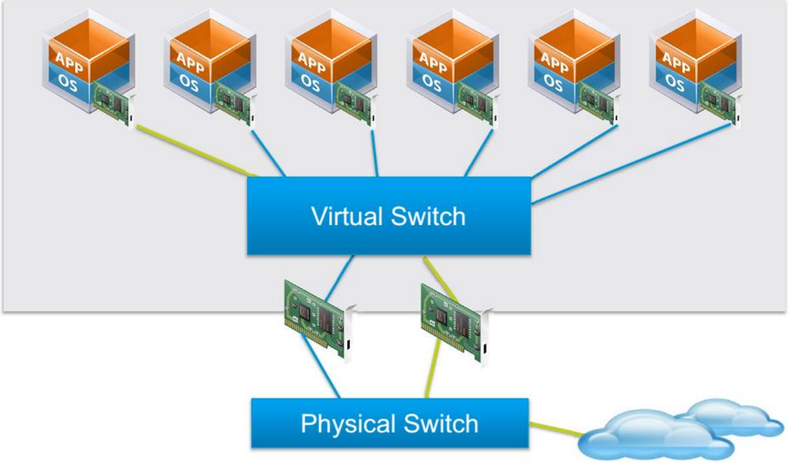


Figure 19. Virtual Switch Scheme.

As it can be appreciated in the picture above, virtual switches are connected to physical switches by using physical Ethernet adapters, also referred as uplink adapters.

In addition, a vSwitch it is used inside of a virtual network to connect VMs, no matter if they are in the same ESXi or not. As a physical switch, the vSwitch identifies which VMs are connected to its ports, then, it forwards the traffic to the properly destination. As a result, there are two kinds of vSwitch:

- vSphere Standard Switch: Standard switch manages virtual machine and host networking at the host level. This means that, it can bridge traffic internally between virtual machines in the same standard PortGroup[1] and repeating: this vSwitch is created in the ESXi host and manages VMs that reside in the same host.
- vSphere Distributed Switch: Distributed switch manages virtual machine and host networking at vCenter Server level. It works across associated hosts, while maintaining communications between VMs in the same distributed port group, that they need to be communicated but remain in different hosts.

It is scalable across the datacenter and reduces VLAN ID usage.

Logical Routing

Routing is necessary to provide communication across isolated logical layer 2. There are two elements to perform logical routing:

- NSX Edge Services Gateway: it is commonly used as a perimeter to the “external world”. It has dynamic routing capability; moreover, it provides network services such as DHCP, NAT, Load Balancer, Firewall and VPN. In this project there are 2 edges deployed; the first one placed in the perimeter of the network and, the second one, inside the network as a load balancer for two web servers. There are different “sizes” of NSX Edge Services Gateway, the size depends on its firewall capabilities.

[1] PortGroup: It is a way to combine multiple ports under a common configuration and provide a stable anchor point of virtual machines connecting to labelled networks. A PortGroup connects to a vSwitch, and a vSwitch connects a physical network interface.

- Logical (distributed router): it is used inside of the network. It supports distribute and dynamic routing. It allows to route traffic between logical switches. The routing capability is distributed in the hypervisor. A benefit is that can replace complex routing topologies in the logical space. In SDN it is used east-west routing instead of north-south which is the most common in traditional networks. What determines if it is a logical router or logical distributed network is if the routers route flows in a ESXi o between different ESXis.

NSX routing supports OSPF and BGP routing protocols.

Distributed Firewall

The distributed firewall from NSX is an innovative firewall that it is not placed in the network and flows pass through it. The NSX distributed firewall is located in the vNIC of each VM which allows easily to deploy microsegmentation. As it is embedded at vNIC layer, it analyzes the traffic before and after encapsulation.

The next figure shows an example of a scenario with DFW, it can be appreciated that although there is a firewall from the NSX Edge Gateway, and underneath of each virtual machine there is a firewall, a NSX Distributed Firewall.

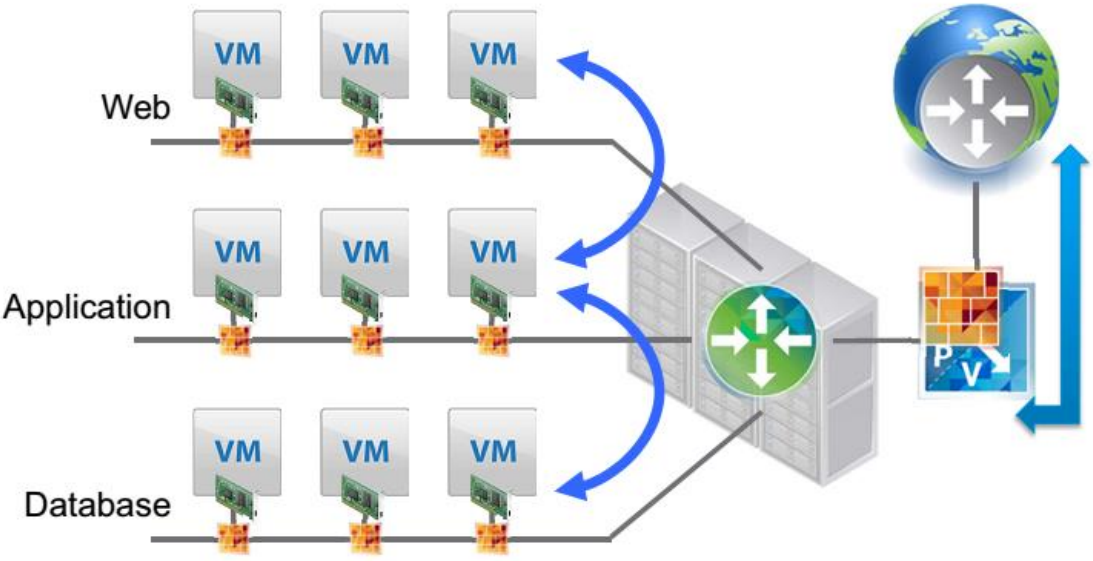


Figure 20. DFW Example.

It allows segmentation of virtual data centers and specially if virtual machines are based on names and attributes, user identity, vCenter Server objects like port groups, security groups, cluster, logical switch, resource pool, data center, VM name and VM operating system. It also provides firewall filtering with traditional networking elements: IP addresses/ranges and ports. Moreover, firewall rules can be dynamic or based in identity.

Firewall rules are configured and managed by the vSphere Web Client or REST APIs.

The Edge firewall it is not a distributed firewall, it is included, as commented in the NSX Edge services gateway. It provides firewall capabilities for flows moving through the data centre. In most of cases, this firewall is used for north-south traffic.

NSX Manager

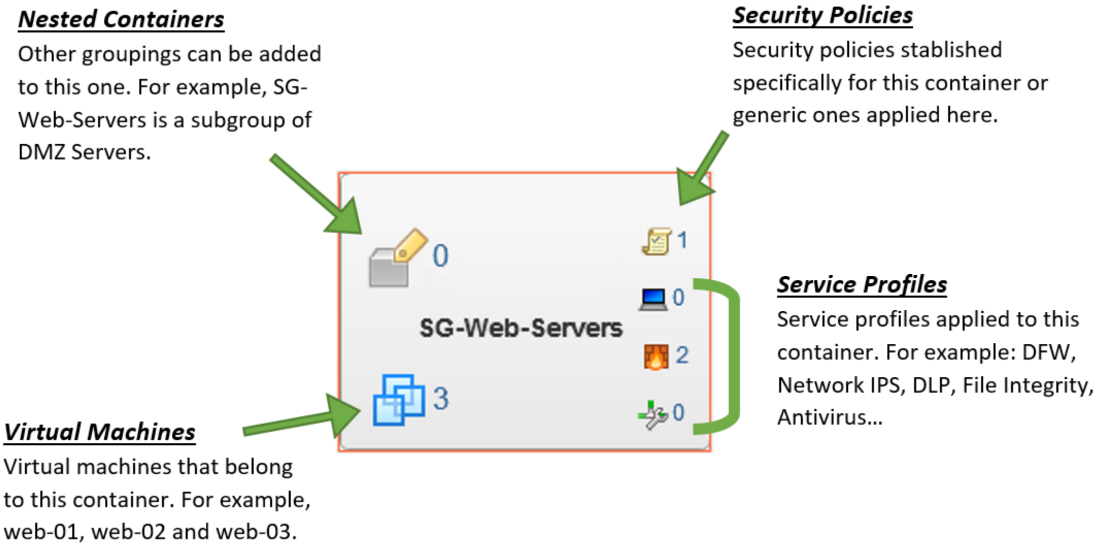
The NSX Manager is the management plane of the solution. It is the single point of configuration and REST API entry points. It has to be registered in the vCenter Server. It is explained and deployed in future sections of this project.

NSX Controller

It is a cluster of controllers; the recommended number is three controllers. The controllers provide control plane functions for switching and routing. It has information about all hosts, logical switches and distributed logical routers. It is basically the control plane of the solution. It is deployed in future sections of this project.

Service Composer

The service composer is a container that an administrator can graphically manage security policies and security groups. It is composed by a set of VMs, IP Addresses or other network components. These components have to have something in common or something generic that has to be applied to everyone in that set. For example, for a Quarantine Zone, Web services/servers… The next figure shows what the service composer is and what each icon offers.



Nested Containers

Other groupings can be added to this one. For example, SG-Web-Servers is a subgroup of DMZ Servers.

Security Policies

Security policies stablished specifically for this container or generic ones applied here.

Service Profiles

Service profiles applied to this container. For example: DFW, Network IPS, DLP, File Integrity, Antivirus…

Virtual Machines

Virtual machines that belong to this container. For example, web-01, web-02 and web-03.

Figure 21. Service composer structure.

It is clearly seen that this NSX element it is for Security instead to build a network. A couple of definition are going to be described in order to better comprehend why it is so useful the Service composer and how NSX associates behaviours and components.

- Security Group: It is a collection of assets, such as VMs, or grouping objects from the vSphere inventory.
- Security Policy: It is a set of services, for example firewall rules or network introspection services, applied to a security group.

Other NSX Concepts

- VXLAN: VXLAN stands for virtual extended LAN that allows running an overlay logical network. While in traditional networks the LAN limit is 4000, in NSX this limit grows up to 16 million networks.
- Transport Zone: A transport zone is like a tag that controls which hosts can reach a switch. In conclusion, it determines which clusters and VMs can participate in the use of a particular network. There are two kinds of TZ; Global scope or Universal scope depending if the TZ spans one or more vSphere clusters. An NSX environment can contain more than one TZ although, a logical switch can belong only to one TZ. Other considerations are that virtual machines that reside in different transport zones cannot be on the same layer 2 network. TZ have to be well-assigned, for example, the next figure shows a TZ that is misaligned and the consequences are going to be commented.

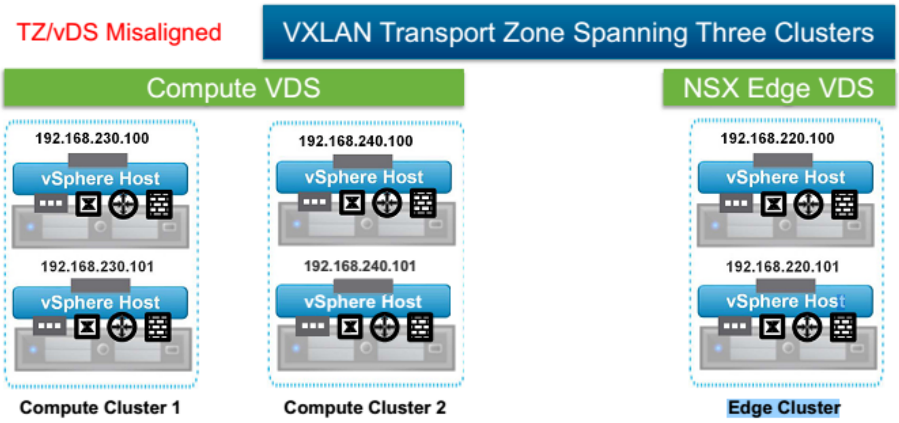


Figure 22. TZ misaligned.

In the above figure it can be clearly seen that Compute Cluster 1 is not included in the TZ although that cluster is attached to Compute VDS. A logical switch is essentially a port group on the VDS and the compute cluster 2 is a member of the TZ. In this case, the VXLAN connectivity will work fine. The problems come when it deals about DLR.

The DLR is only created by the NSX Manager on each host in the TZ, thus, VMs that reside in Computer Cluster 1 will be able to communicate at layer 2, by contrast, layer 3 connectivity will be broken.

- vNIC: It stands for virtual network interface card, so, the NIC of a VM.

2.4.2. Other VMware products

As commented before, VMware it is not focused only in NSX, they have other kinds of products although they are related. The most known of them are Networking and Security (NSX), vSAN, Site Recovery Manager and vRealize. The Networking and security will be described later with details, other ones will be commented briefly.

- vSphere: It is the market leader hypervisor. It offers a simple and efficient management at scale. It is built in security and policy management. vSphere can be integrated with other VMware products, such as: vSAN, NSX, among others. It is a Universal Application Platform with existing applications and new ones. For example: Exchange, SAP, Oracle, OpenStack and Outlook and revolutionary workloads like Big Data, High Performance Computing, Artificial Intelligence and Machine Learning.

- vSAN: It is a virtual storage structure. Its principles are efficiency, reduce costs and transform physical servers in a hybrid converge structure. vSAN allows to eliminate the complexity of a traditional structure whereas there is no need to deploy separate arrays, storage networks and cabling thanks to a policy based in rules made of specific requirements for different VMs; these rules are related with performance and availability.
- Site Recovery Manager: The principal function is to deliver current services without interruption whereas it is a disaster avoidance. While legacy resources require technical and specialized teams with complex and resources difficult to test, the SRM it is easy to manage and improves reliability and reduces cost. This is used and called as SDDC (Software Defined Data Center) and as commented before it provides the availability and a private cloud automation software with a fast and reliable recovery. In case of a failure event there is zero downtime application mobility (with vCenter vMotion, it is verified in this project).

2.4.3. NSX Benefits

The major benefit of having NSX is that things are easier having more security, speed and agility. All of this spending less money in comparison with actual networks which are based in the traditional network model.

There is centralized management and all is virtualized by an hypervisor. Network components run over software instead of hardware.

Automation is possible so, NSX provides a highly resilient, available and automated network.

The network can be placed in a cloud, no matters if is the NSX cloud or AWS.

In particular, the descripted elements (logical switches, logical routers and DFW) , offer several benefits:

Logical Switching benefits

- Logical networks are decoupled from physical ones due to VXLAN-based overlay.
- VLAN ID usage is reduced.
- Scalable multitenancy across the datacenter.
- Layer 2 over Layer 3 infrastructure.

Logical Routing benefits

- Distributed hypervisor-based logical routing.
- Support for OSPF and BGP routing protocols.
- Optimized north-south and east-west traffic flows.

Distributed Firewall benefits

- Distributed at hypervisor level.
- vNIC segmentation.
- High line rate; about 20 Gbps per host.
- Rules based on VM name, vCenter Server objects, identity-based rules…

3. Project development

This section explains how the network was built, and some tips regarding basic configurations.

There are 3 steps in an NSX project: the first one is to install the hypervisor; the second one is the network design; and finally the third one is, the optimization and managing of the full architecture.

The first and second steps are going to be explained in this section. This section also contains a description of each component, the deployment method and some characteristics and recommendations. The optimization and managing of the full architecture are explained in the next section by explaining some deployed use cases.

To put the audience in context, the network maps are shown. IP addresses and virtual machines distribution are located in an annex.

The figure below represents the diverse levels that are required to develop the principal scenario.

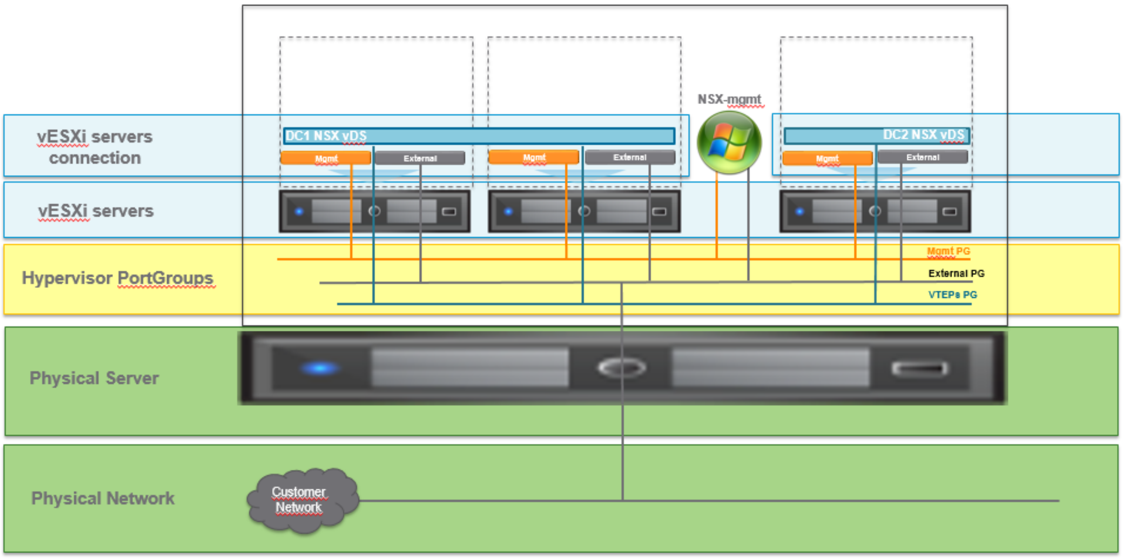


Figure 23. Required levels to develop the principal scenario.

The customer physical network is the one responsible for providing connectivity between the technician and the physical server. Hence, the customer physical server is a Dell R420. The Customer created PortGroups are needed to communicate the physical server with the different vESXi and network components that will be deployed. And Finally, the vESXi and the network designed.

There are 3 vESXi so, as they remain in the same physical server, each vESXi has to be virtualized transforming into vESXi. Moreover, there are 3 PortGroups created: Management PG, External PG and VTEPs PG, they will be detailed after.

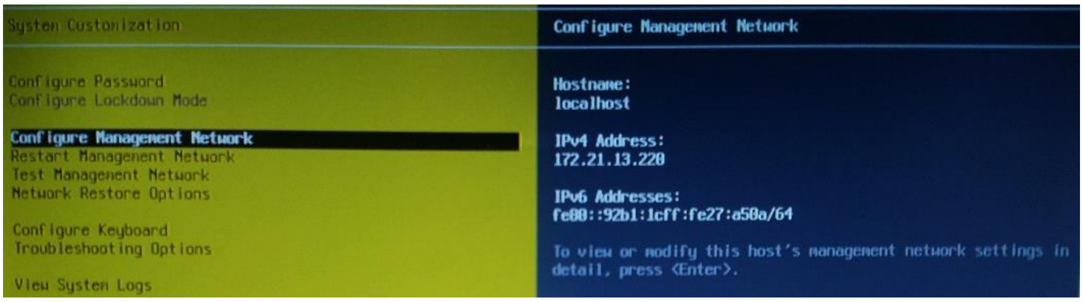
The technician accesses the server where the hypervisor is installed through the customer physical server. From there, the customer can access the virtual world using a hop machine which avoids to work directly on the physical server.

VMs used in this project are provided by a department of the company. The network design and configurations have been performed by the student as part of the project.

3.1. Hypervisor

The hypervisor is the most basic level, so before starting to deploy virtual machines corresponding to web or app servers, vESXi have to be deployed.

Once the hypervisor is installed, a “Welcome to the VMware NSX” message appears. After that, an IP address has to be assigned; this is the IP address used by the engineer to access the server. In this case it is: 172.21.13.220.



```
Configure Management Network

Hostname:
localhost

IPv4 Address:
172.21.13.220

IPv6 Addresses:
fe80::92b1:1cff:fe27:a50a/64

To view or modify this host's management network settings in
detail, press <Enter>.
```

Figure 24. Hypervisor Management Network configuration.

Regarding on Figure 23, there are three kinds of PortGroups that must be included: External, VTEPs and Mgmt. A PortGroup (PG) is an aggregation of multiple ports under a common configuration that provides a stable anchor point for virtual machines connecting to labelled networks.

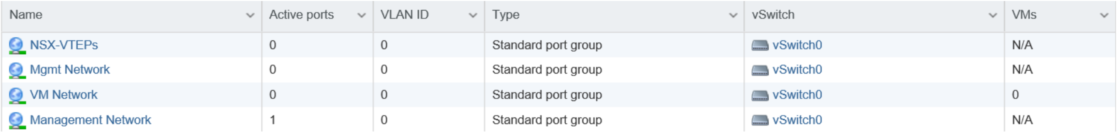
- External Network PG: Provides communication between the virtual environment with customer networks. It is used by the management station (NSX-mgmt) and NSX Edges. It requires a physical NIC associated as Uplink traffic.
- Management Network: Provides communication between management components; so, it is intended for traffic with management purposes. It is not mapped to any physical NIC because is internal to the ESXi.
- NSX-VTEPS: Provides communication between ESXi VTEP interfaces; it is used to send VTEP traffic, for example, packets between virtual machines. As the management network, it is not mapped to any physical NIC due to is internal to the ESXi.

Due to a confusion when naming PortGroups in the Hypervisor, this is the relationship between Figure 23 PortGroups and the ones defined in the Hypervisor is[2]:

Initial PortGroup	Hypervisor PortGroup definition
Internal Management	Mgmt Internal
VTEPs	VTEPs Internal
External	Mgmt Network

Table 4. PortGroups equivalences.

Next figure represents defined PortGroups in the Hypervisor:



Name	Active ports	VLAN ID	Type	vSwitch	VMs
NSX-VTEPs	0	0	Standard port group	vSwitch0	N/A
Mgmt Network	0	0	Standard port group	vSwitch0	N/A
VM Network	0	0	Standard port group	vSwitch0	0
Management Network	1	0	Standard port group	vSwitch0	N/A

Figure 25. PortGroups defined in the Hypervisor.

After having the PG created, the hypervisor is ready to embrace a hop VM and the 3 vESXi.

For this project, there is only one vNIC configured in the physical server, but there could be more if necessary. Moreover, DNS configuration it is not required although if it is necessary, it can be done.

3.1.1. Hop Virtual Machine: NSX-mgmt

The main purpose of this machine is to provide an entrance to the virtual world without being connected directly to the server. It is the VM that allows to control, configure and manage the virtual network, for this reason, the hop machine it is used to control and restrict users that can access to the virtual world.

The hop machine has an .OVA format; this fact allows an easy deployment in the vSphere Client (Hypervisor). During its deployment, there are some things that must be configured:

- Virtual machine name.
- Map the networks used by this VM. (It is important to create the networks before, on the contrary, the VM cannot be deployed)[3].
    - Network adapter 1 corresponds to Mgmt Internal.
    - Network adapter 2 corresponds to Mgmt Network (External).

[2] In the virtual world, as PG have to be created again, the nomenclature mistake will not be present.

[3] The order when allocating network adapters is important, this fact determines the network configuration inside of the VM.

Due to lack of experience this deployment took around 10h because the VM has a size of 40GB and the External Disk that contained the VM was connected through the PC and not the server. For big size VMs, it is recommended to deploy them connecting directly the External Disk on the physical server.

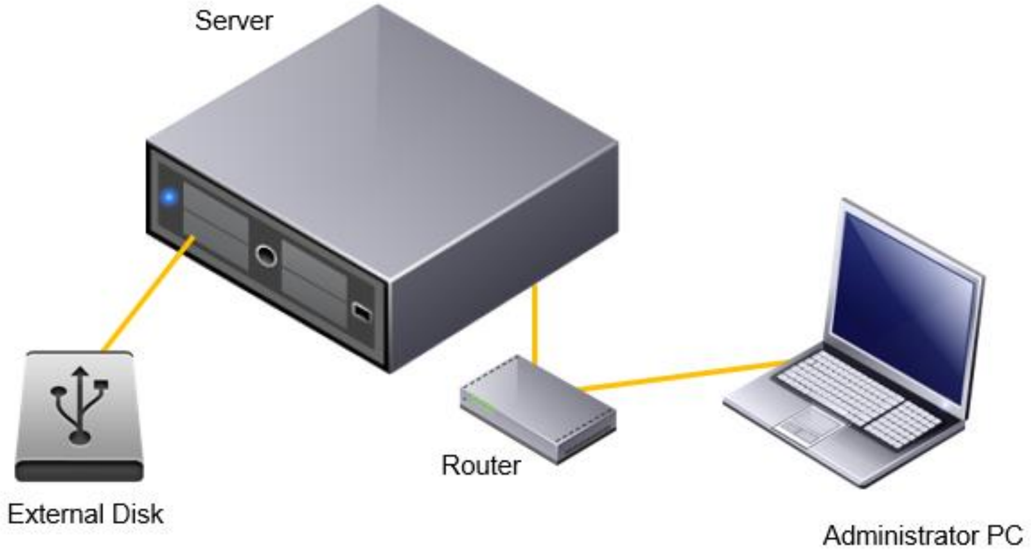
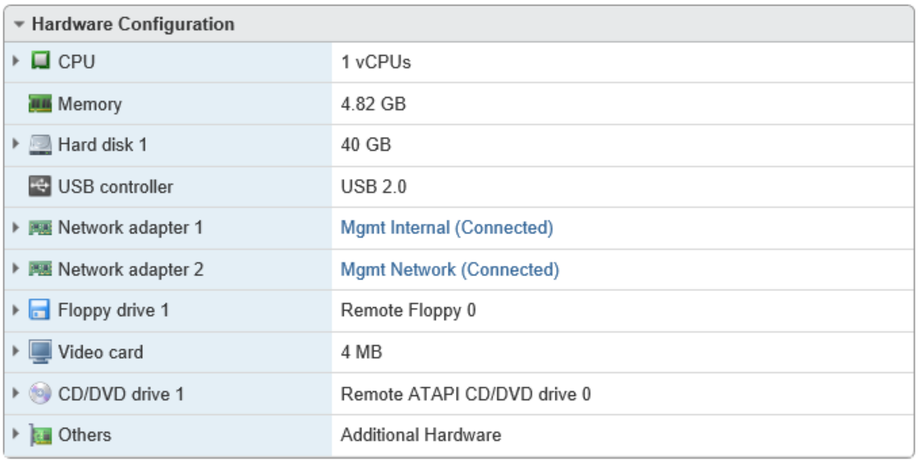


Figure 26. Recommended connections to deploy VM stored in an External Disk.

Once the deployment is done, the hop machine has the following configuration in the vSphere (Hypervisor):

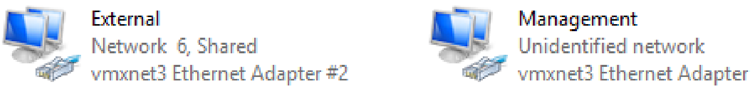


Hardware Configuration	
CPU	1 vCPUs
Memory	4.82 GB
Hard disk 1	40 GB
USB controller	USB 2.0
Network adapter 1	Mgmt Internal (Connected)
Network adapter 2	Mgmt Network (Connected)
Floppy drive 1	Remote Floppy 0
Video card	4 MB
CD/DVD drive 1	Remote ATAPI CD/DVD drive 0
Others	Additional Hardware

Figure 27. Hop machine, NSX-mgmt, configuration.

Hence it can be appreciated that the first network adapter is the Management Internal and the second one, although it is called Mgmt Network (connection with virtual environment), corresponds to the External Network (connection with physical environment).

The network configuration inside the hop machine is:

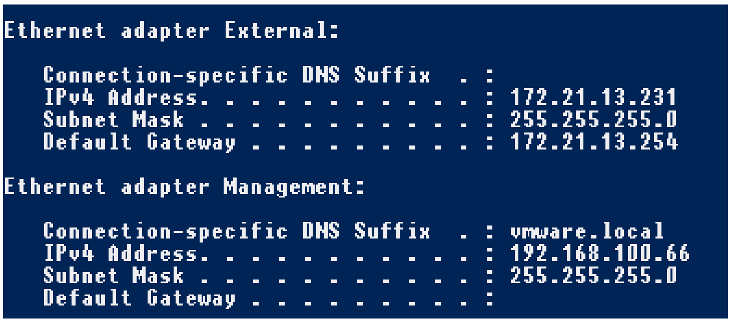


External
Network 6, Shared
vmxnet3 Ethernet Adapter #2

Management
Unidentified network
vmxnet3 Ethernet Adapter

Figure 28. Hop machine Network adapters.

Regarding Ethernet adapters, the configuration is:



```
Ethernet adapter External:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.21.13.231
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.21.13.254

Ethernet adapter Management:

   Connection-specific DNS Suffix  . : vmware.local
   IPv4 Address. . . . . . . . . . . : 192.168.100.66
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

Figure 29. Ethernet adapters configuration.

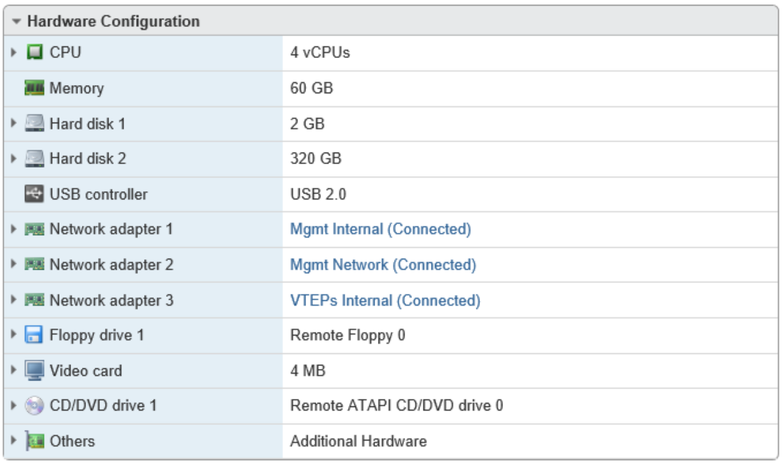
There is so emphasis with that because, when assigning Network adapters in the Hypervisor, the order was altered. This drove in a configuration error where DHCP requests from the virtual environment arrived to the physical one. This caused a mix of DHCP requests in the physical router and anyone connected to that router, lost connectivity. It should be mentioned that the affected router provides internet connectivity to some employees.

3.1.2. vESXi

The virtual ESXi's have to be deployed too; its deployment is performed using the same method as the one for the hop machine. For network adapters, the configuration is kind of different because instead of having a couple of network adapters, there are three.

Due to the big size of each vESXi, the USB had to be connected to the physical server. The elapsed time for deploying was of 8, 7, and 5 hours depending on the vESXi.

The final vESXi configuration in vSphere is:



Hardware Configuration	
CPU	4 vCPUs
Memory	60 GB
Hard disk 1	2 GB
Hard disk 2	320 GB
USB controller	USB 2.0
Network adapter 1	Mgmt Internal (Connected)
Network adapter 2	Mgmt Network (Connected)
Network adapter 3	VTEPs Internal (Connected)
Floppy drive 1	Remote Floppy 0
Video card	4 MB
CD/DVD drive 1	Remote ATAPI CD/DVD drive 0
Others	Additional Hardware

Figure 30. vESXI Network Adapters configuration.

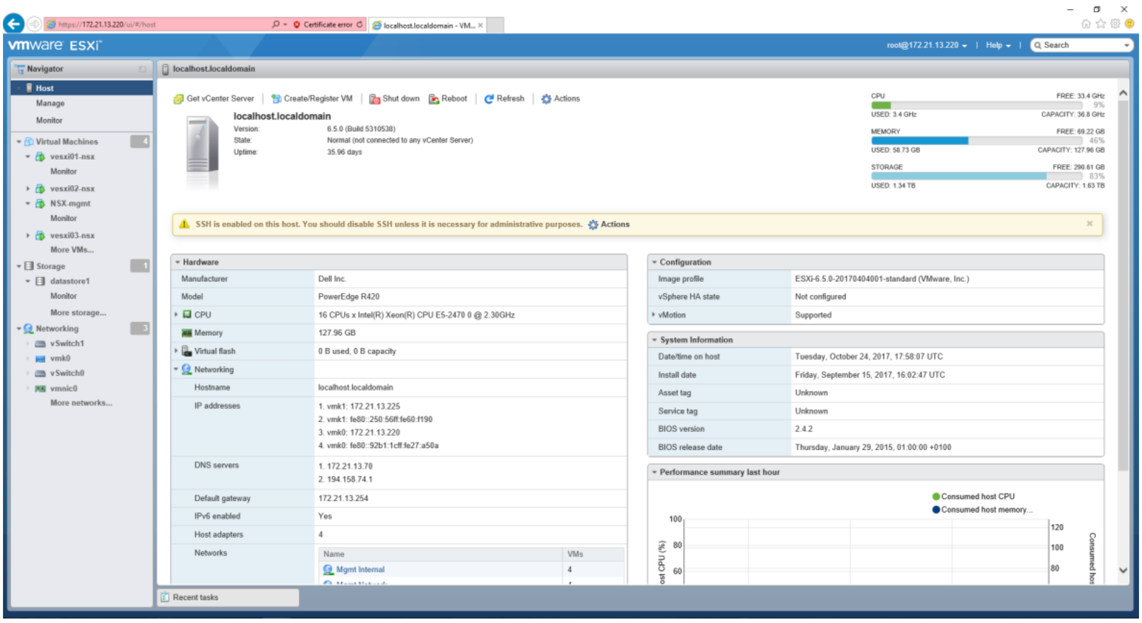
This network adapter configuration is replicated in the other vESXi's. It has to be commented that the first network mapping was temporary because it drove to a problem with DHCP propagation that will be explained later.

Network mapping notes:

- NSX interna ⬄ Mgmt Internal
- NSX DMZ ⬄ Mgmt Network (the one connecting to the external world)
- NSX Transport ⬄ VTEPs Internal

3.1.3. Hypervisor Networking

Once the hop machine and the vESXi are deployed this is how the hypervisor looks:



localhost.localdomain

Version: 6.5.0 (Build 5310538)
State: Normal (not connected to any vCenter Server)
Uptime: 35.96 days

CPU FREE: 33.4 GHz 9% USED: 3.4 GHz CAPACITY: 36.8 GHz

MEMORY FREE: 69.22 GB 46% USED: 58.73 GB CAPACITY: 127.96 GB

STORAGE FREE: 290.61 GB 83% USED: 1.34 TB CAPACITY: 1.63 TB

SSH is enabled on this host. You should disable SSH unless it is necessary for administrative purposes. Actions

Hardware	
Manufacturer	Dell Inc.
Model	PowerEdge R420
CPU	16 CPUs x Intel(R) Xeon(R) CPU E5-2470 0 @ 2.30GHz
Memory	127.96 GB
Virtual flash	0 B used, 0 B capacity
Networking	
Hostname	localhost.localdomain
IP addresses	1. vmk1: 172.21.13.225 2. vmk1: fe80::250:56ff:fe60:f190 3. vmk0: 172.21.13.220 4. vmk0: fe80::92b1:1cff:fe27:a50a
DNS servers	1. 172.21.13.70 2. 194.158.74.1
Default gateway	172.21.13.254
IPv6 enabled	Yes
Host adapters	4
Networks	

Name	VMs
Mgmt Internal	4

Configuration	
Image profile	ESXi-6.5.0-20170404001-standard (VMware, Inc.)
vSphere HA state	Not configured
vMotion	Supported

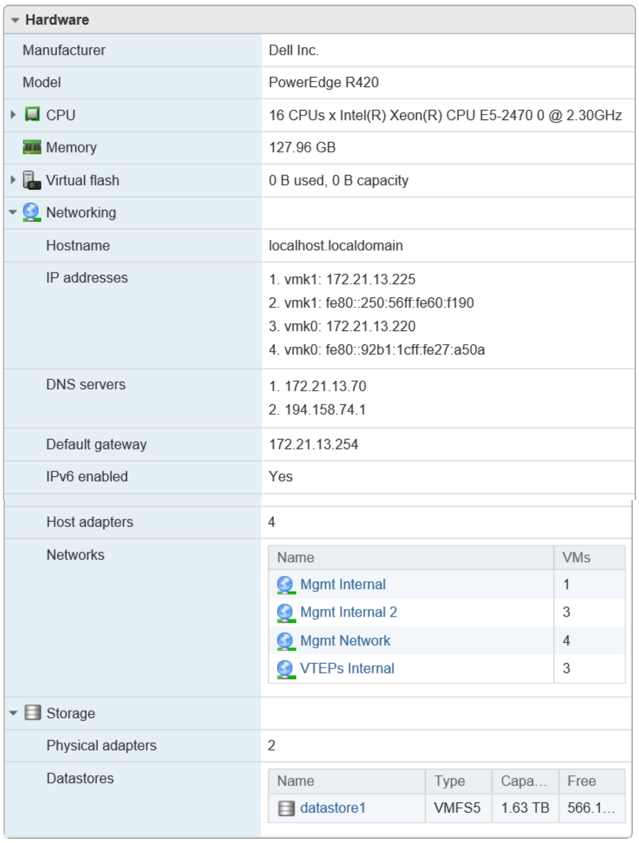
System Information	
Date/time on host	Tuesday, October 24, 2017, 17:58:07 UTC
Install date	Friday, September 15, 2017, 16:02:47 UTC
Asset tag	Unknown
Service tag	Unknown
BIOS version	2.4.2
BIOS release date	Thursday, January 29, 2015, 01:00:00 +0100

Performance summary last hour

● Consumed host CPU
● Consumed host memory...

Figure 31. Hypervisor main screen.

Hypervisor Hardware Tab:



Hardware	
Manufacturer	Dell Inc.
Model	PowerEdge R420
CPU	16 CPUs x Intel(R) Xeon(R) CPU E5-2470 0 @ 2.30GHz
Memory	127.96 GB
Virtual flash	0 B used, 0 B capacity
Networking	
Hostname	localhost.localdomain
IP addresses	1. vmk1: 172.21.13.225 2. vmk1: fe80::250:56ff:fe60:f190 3. vmk0: 172.21.13.220 4. vmk0: fe80::92b1:1cff:fe27:a50a
DNS servers	1. 172.21.13.70 2. 194.158.74.1
Default gateway	172.21.13.254
IPv6 enabled	Yes
Host adapters	4
Networks	
Storage	
Physical adapters	2
Datastores	

Networks:

Name	VMs
Mgmt Internal	1
Mgmt Internal 2	3
Mgmt Network	4
VTEPs Internal	3

Datastores:

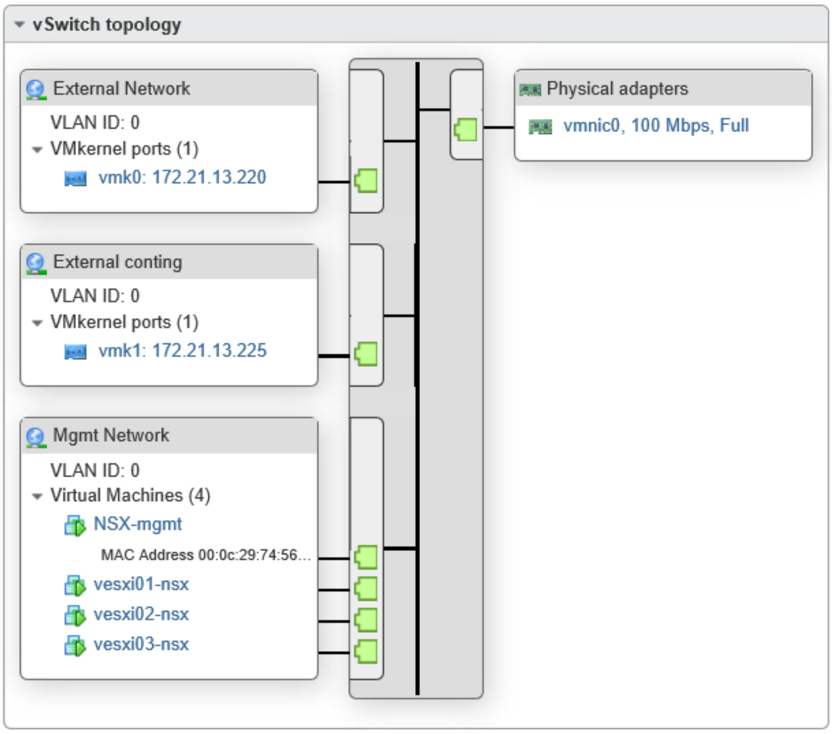
Name	Type	Capa...	Free
datastore1	VMFS5	1.63 TB	566.1...

Figure 32. Hypervisor Hardware.

The figure above shows the main characteristics of the physical server where the hypervisor is installed. Regarding IP addresses, VMware always shows the IP address in IPv4 and IPv6. The DNS servers are 2, one is internal and the other one is managed by Andorra Telecom. The tap host also shows some statistics of the machine and the networks configured (Mgmt Internal, Mgmt Network…).

In the Hypervisor it is necessary to create the link between the physical and the virtual world. The properly way to do that is creating two virtual switches decoupling the physical network, the hop machine and the virtual environment. These are the implemented vSwitches.

vSwitch 0 links physical with virtual worlds:



vSwitch topology

External Network
VLAN ID: 0
VMkernel ports (1)
vmk0: 172.21.13.220

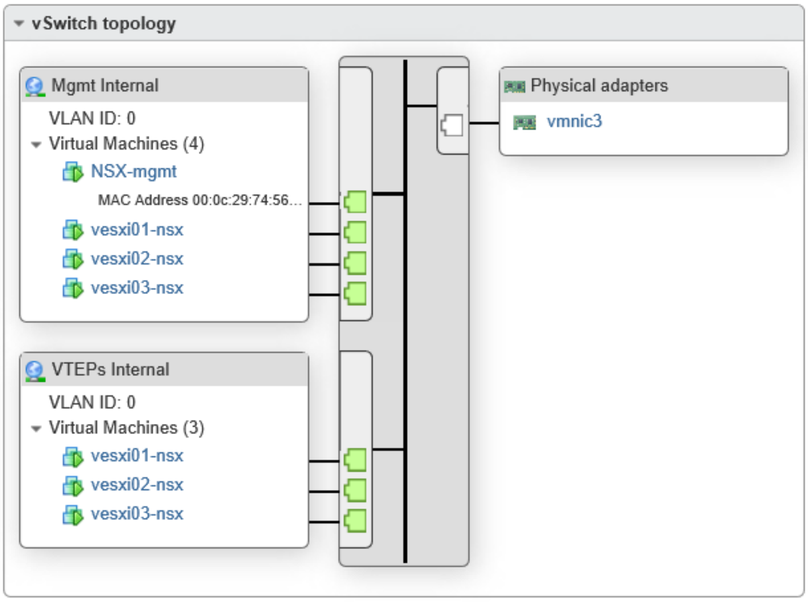
External conting
VLAN ID: 0
VMkernel ports (1)
vmk1: 172.21.13.225

Mgmt Network
VLAN ID: 0
Virtual Machines (4)
NSX-mgmt
MAC Address 00:0c:29:74:56...
vesxi01-nsx
vesxi02-nsx
vesxi03-nsx

Physical adapters
vmnic0, 100 Mbps, Full

Figure 33. vSwitch 0.

vSwitch 1 links the hop machine with the management and data plane of the virtual network:



▾ vSwitch topology

Mgmt Internal
VLAN ID: 0
▾ Virtual Machines (4)
NSX-mgmt
MAC Address 00:0c:29:74:56...
vesxi01-nsx
vesxi02-nsx
vesxi03-nsx

VTEPs Internal
VLAN ID: 0
▾ Virtual Machines (3)
vesxi01-nsx
vesxi02-nsx
vesxi03-nsx

Physical adapters
vmnic3

Figure 34. vSwitch 1.

Looking at the previous two figures it is easy to determine which vSwitch belongs to the virtual environment due to the physical adapter is disabled, as the name says, it must be virtual.

To comprehend the above configuration, it is represented in a network map:

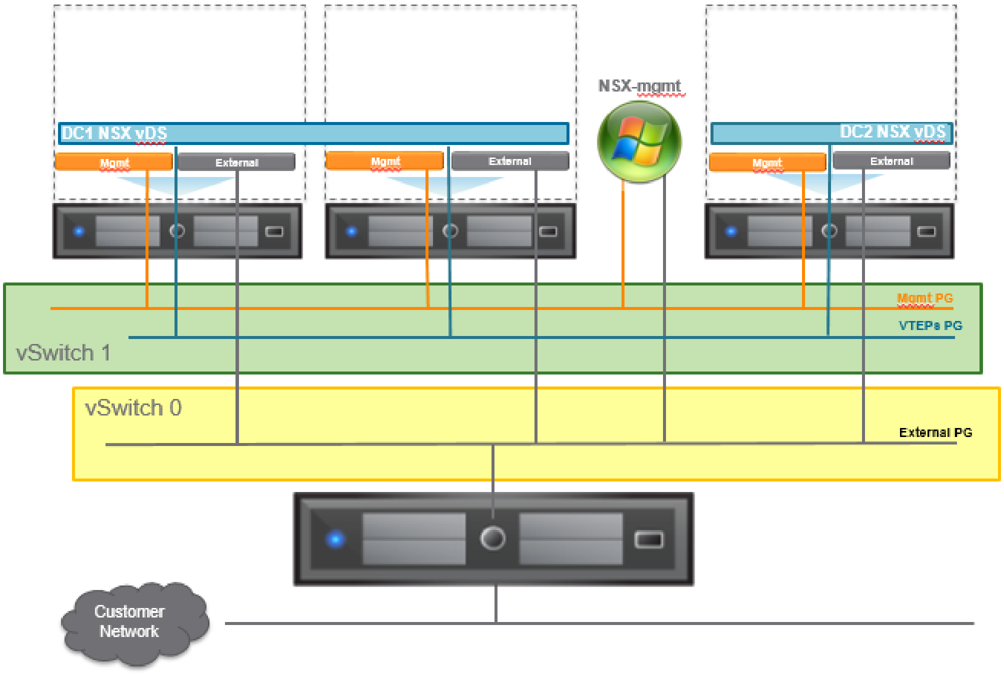
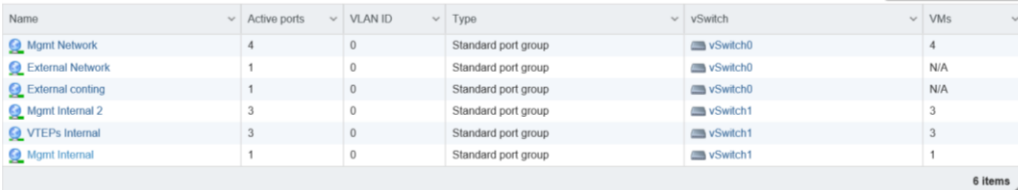


Figure 35. vSwitch 0 and vSwitch 1 representation.

Components that compound Hypervisor:

- Port Groups: As commented before, there are three port groups in the Hypervisor that are necessary. In the figure below, there are shown the port groups with the number of VMs attached.

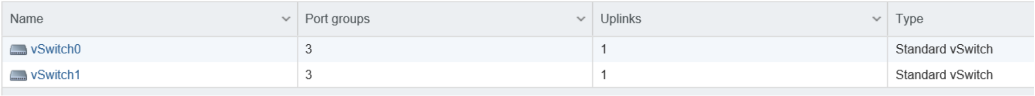


Name	Active ports	VLAN ID	Type	vSwitch	VMs
Mgmt Network	4	0	Standard port group	vSwitch0	4
External Network	1	0	Standard port group	vSwitch0	N/A
External conting	1	0	Standard port group	vSwitch0	N/A
Mgmt Internal 2	3	0	Standard port group	vSwitch1	3
VTEPs Internal	3	0	Standard port group	vSwitch1	3
Mgmt Internal	1	0	Standard port group	vSwitch1	1

6 items

Figure 36. PortGroups defined in the Hypervisor[4].

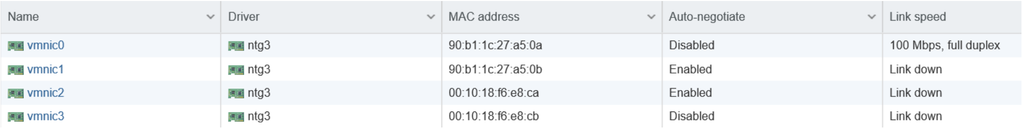
- Virtual Switches: There are two virtual switches. The design allows to decouple physical environment from the virtual one.



Name	Port groups	Uplinks	Type
vSwitch0	3	1	Standard vSwitch
vSwitch1	3	1	Standard vSwitch

Figure 37. vSwitches defined in the Hypervisor.

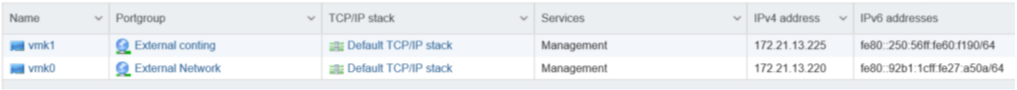
- Physical NICs: There is used only one physical NIC due to the single link with the physical world. The other physical NICs remain unused.



Name	Driver	MAC address	Auto-negotiate	Link speed
vmnic0	ntg3	90:b1:1c:27:a5:0a	Disabled	100 Mbps, full duplex
vmnic1	ntg3	90:b1:1c:27:a5:0b	Enabled	Link down
vmnic2	ntg3	00:10:18:f6:e8:ca	Enabled	Link down
vmnic3	ntg3	00:10:18:f6:e8:cb	Disabled	Link down

Figure 38. Physical NICs in the Hypervisor.

- VMKernel NICs: They are special constructs used by the vSphere host to communicate with the outside world.

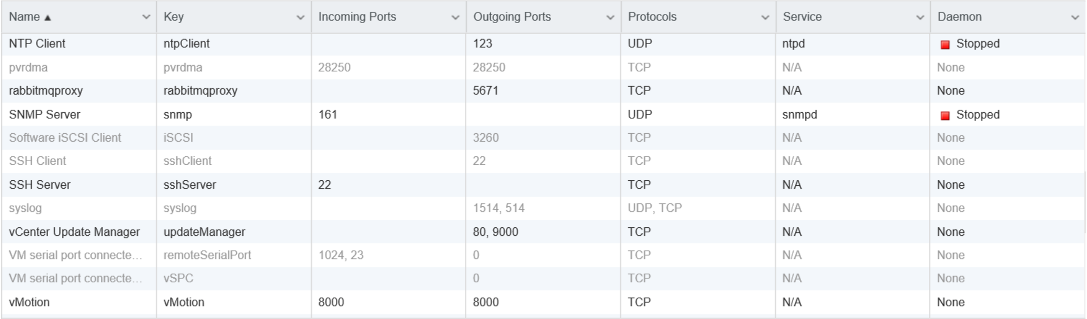


Name	Portgroup	TCP/IP stack	Services	IPv4 address	IPv6 addresses
vmk1	External conting	Default TCP/IP stack	Management	172.21.13.225	fe80::250:56ff:fe60:f190/64
vmk0	External Network	Default TCP/IP stack	Management	172.21.13.220	fe80::92b1:1cff:fe27:a50a/64

Figure 39. VMKernel NICs in the Hypervisor.

[4] Used port groups are Mgmt Network, Mgmt Internal and VTEPs Internal, the other ones are for testing purposes.

- Hypervisor Firewall Rules: Minimum connections that need to be allowed at hypervisor level.



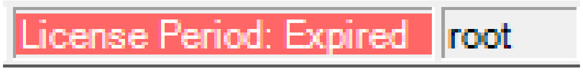
Name ▲	Key	Incoming Ports	Outgoing Ports	Protocols	Service	Daemon
NTP Client	ntpClient		123	UDP	ntpd	🟥 Stopped
pvrdma	pvrdma	28250	28250	TCP	N/A	None
rabbitmqproxy	rabbitmqproxy		5671	TCP	N/A	None
SNMP Server	snmp	161		UDP	snmpd	🟥 Stopped
Software iSCSI Client	iSCSI		3260	TCP	N/A	None
SSH Client	sshClient		22	TCP	N/A	None
SSH Server	sshServer	22		TCP	N/A	None
syslog	syslog		1514, 514	UDP, TCP	N/A	None
vCenter Update Manager	updateManager		80, 9000	TCP	N/A	None
VM serial port connecte...	remoteSerialPort	1024, 23	0	TCP	N/A	None
VM serial port connecte...	vSPC		0	TCP	N/A	None
vMotion	vMotion	8000	8000	TCP	N/A	None

Figure 40. Hypervisor firewall rules[5].

3.1.4. Licensing

Increasingly, there is the tendency that more money is made with licenses instead of hardware. This fact arises out that companies have to buy a piece of hardware in more or less time depending if its broken or not; however, licenses are annual and there is a constant income.

As commented previously, for this project 3 kinds of licenses are required. When turning on the vESXi hosts, a message on the down-right part of the screen appeared:



License Period: Expired root

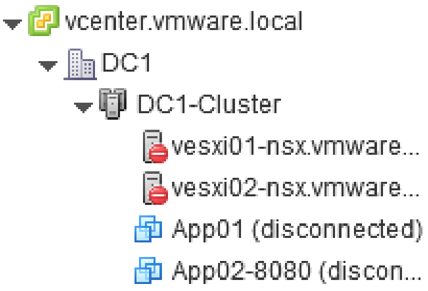
Figure 41. Expired License.

This was the first indicator that the machines are not licensed and thus, VMs inside of a vESXi cannot turned on. From vSphere client a vESXi host can be licensed. As an appointment, there are two ways to access the vSphere, vSphere web client (from an internet browser) and vSphere client. Something that was noticed during the licensing period was that although vESXi hosts were licensed on vSphere Client, when accessing vSphere web client SSO[6], vESXi hosts and clusters had to be re-licensed; this was observed due to although the order to turn vESXi hosts was performed, they remained switched off and licenses were not applied.

[5] There are more firewall rules although these are the ones regarding services that before, were reviewed for being up.

[6] SSO stands for Single Sign On: In the scenario there are two vCenters and both have to be licensed and well connected.

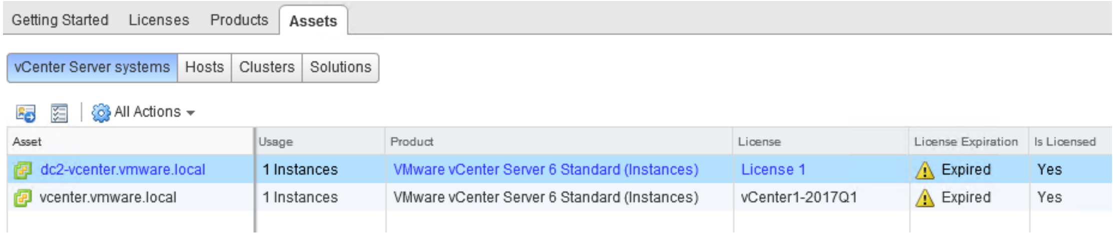
The following image shows how an vESXi remains off with its VMs disconnected:



```
vcenter.vmware.local
  DC1
    DC1-Cluster
      vesxi01-nsx.vmware...
      vesxi02-nsx.vmware...
      App01 (disconnected)
      App02-8080 (discon...
```

Figure 42. vESXi and VMs disconnected.

The licensing and re-licensing on vSphere Web Client has to be done in the administration tab and then, licensing. First of all, licenses have to be uploaded and then, they must be assigned to an asset; for example, an vESXi host or NSX, also the vCenter… If this assignment is not performed, the status of an assed remains as expired because the license is still expired.

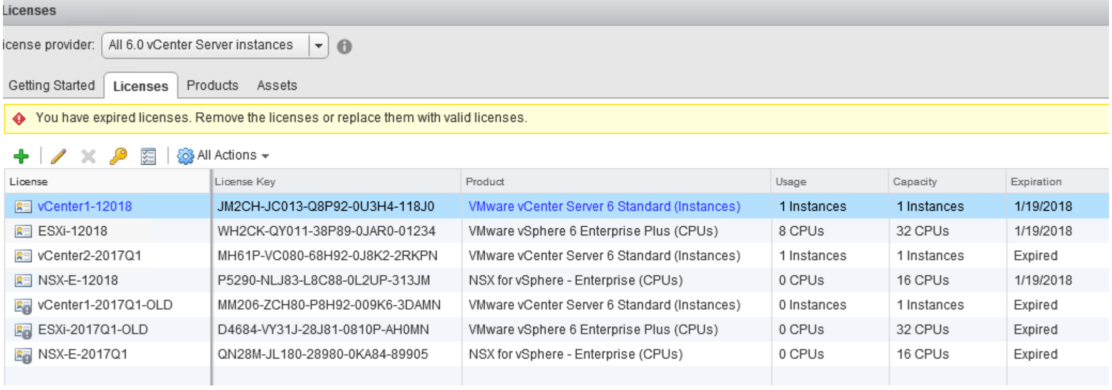


Asset	Usage	Product	License	License Expiration	Is Licensed
dc2-vcenter.vmware.local	1 Instances	VMware vCenter Server 6 Standard (Instances)	License 1	⚠ Expired	Yes
vcenter.vmware.local	1 Instances	VMware vCenter Server 6 Standard (Instances)	vCenter1-2017Q1	⚠ Expired	Yes

Figure 43. Non-assigned and expired licenses.

When a license is about to expire, vSphere Web Client notifies it.

This is how licenses look:

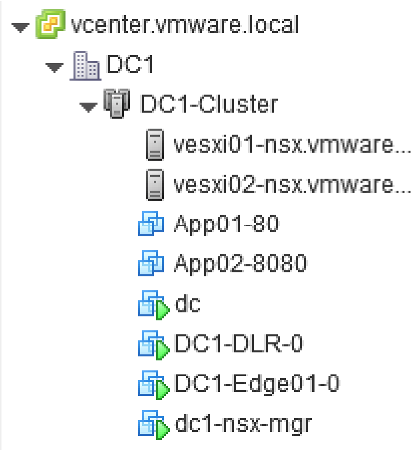


You have expired licenses. Remove the licenses or replace them with valid licenses.

License	License Key	Product	Usage	Capacity	Expiration
vCenter1-12018	JM2CH-JC013-Q8P92-0U3H4-118J0	VMware vCenter Server 6 Standard (Instances)	1 Instances	1 Instances	1/19/2018
ESXi-12018	WH2CK-QY011-38P89-0JAR0-01234	VMware vSphere 6 Enterprise Plus (CPUs)	8 CPUs	32 CPUs	1/19/2018
vCenter2-2017Q1	MH61P-VC080-68H92-0J8K2-2RKPN	VMware vCenter Server 6 Standard (Instances)	1 Instances	1 Instances	Expired
NSX-E-12018	P5290-NLJ83-L8C88-0L2UP-313JM	NSX for vSphere - Enterprise (CPUs)	0 CPUs	16 CPUs	1/19/2018
vCenter1-2017Q1-OLD	MM206-ZCH80-P8H92-009K6-3DAMN	VMware vCenter Server 6 Standard (Instances)	0 Instances	1 Instances	Expired
ESXi-2017Q1-OLD	D4684-VY31J-28J81-0810P-AH0MN	VMware vSphere 6 Enterprise Plus (CPUs)	0 CPUs	32 CPUs	Expired
NSX-E-2017Q1	QN28M-JL180-28980-0KA84-89905	NSX for vSphere - Enterprise (CPUs)	0 CPUs	16 CPUs	Expired

Figure 44. Licenses list.

Once all is licensed, hosts can be connected and VMs administered. Now, hosts and VMs appear like this:



- vcenter.vmware.local
  - DC1
    - DC1-Cluster
      - vesxi01-nsx.vmware...
      - vesxi02-nsx.vmware...
      - App01-80
      - App02-8080
      - dc
      - DC1-DLR-0
      - DC1-Edge01-0
      - dc1-nsx-mgr

Figure 45. vESXis and VM after being licensed.

The figure above shows that the scenario is ready for applying NSX components.

3.2. <u>NSX</u>

After deploying vESXi and licensing everything, the environment is ready to deploy what is missing: NSX Manager, NSX Controllers, vSwitches and routers.

The scenario of the current project is the following one:

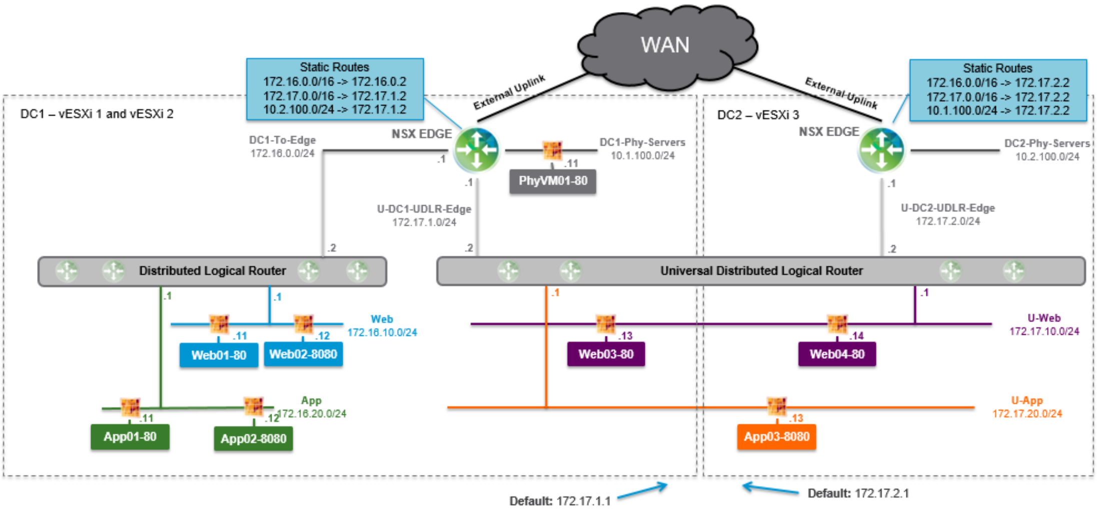


Figure 46. Network Map.

The IP addresses inventory can be found in the Appendix.

3.2.1. NSX Manager

In State of the Art, it is said that NSX Manager is the management plane of the network and it has to be registered in the vCenter Server. This deployment is going to be explained with more details because requires verifications before and when it is finished. It is a critical step of the deployment.

The NSX Manager responsibilities are:

- Provide management UI and VMWare NSX API.
- Install user world agents[7], VXLAN, distributed routing, distributed firewall kernel modules.
- Deploy NSX Controller cluster nodes.
- Configure hosts through a message bus.
- Generate certificates to secure control plane communications.
- Deploy logical networks and services.

Requirements for a successful deployment:

- VMware vCenter Server and vESXi clusters with 5.5 and higher.
- Physical network supporting MTU of at least 1600 bytes, this value corresponds to a NSX requirement.
- vSphere distributed switch.
- NSX Manager needs network connectivity to vCenter Server and management network of vESXi hosts.

The NSX Manager is a virtual machine that is deployed on an vESXi host that belongs to a cluster. Once is deployed, the engineer has to ensure that the VMware Client Integration Plug-in is installed; on the contrary it will not work.

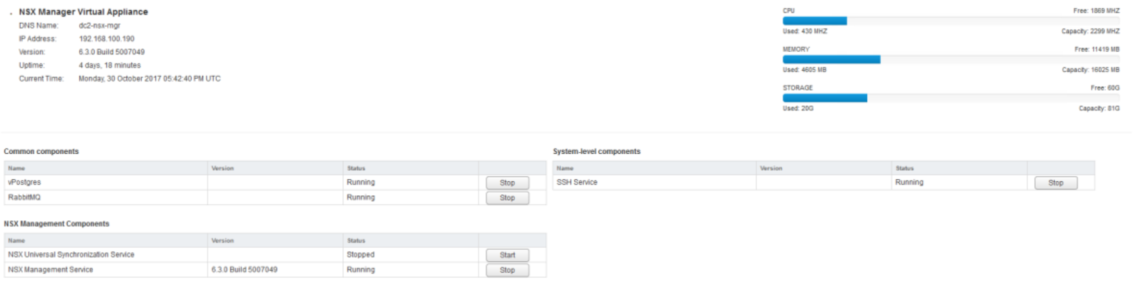
During configuration there are some parameters that have to be configured: DNS server, domain, Hostname, IP Address and Default IPv4 Gateway.

Parameters:

- DNS SERVER: 192.168.100.10
- Domain search list: vmware.local
- Default IPv4 gateway: 192.168.100.254
- Hostname: dc1-nsx-mgr
- Network 1 IP Address: 192.168.100.19 255.255.255.0
- Enable SSH.

[7] There are two user world agents, netcpa and vsfwd. Netcpa collects network information: VMs connected to logical switches, VMs IP and MAC addresses and reports to the VMware NSX Controller instances, moreover uses SSL to secure the control plane communication. The other agent, vsfwd, interacts with NSX Manager to retrieve distributed firewall policy rules, gathers DFW statistics and sends them to NSX Manager, and the same with audit logs that are sent to NSX Manager, moreover, it is the one that receives the configuration from the NSX Manager when creating or deleting DLR or EDGE Services Gateway.

Once parameters are set, the NSX Manager is ready be switched on, and, afterwards access it; this is how NSX Manager looks like:



NSX Manager Virtual Appliance

DNS Name: dc2-nsx-mgr
IP Address: 192.168.100.190
Version: 6.3.0 Build 5007049
Uptime: 4 days, 18 minutes
Current Time: Monday, 30 October 2017 05:42:40 PM UTC

CPU — Free: 1869 MHZ — Used: 430 MHZ — Capacity: 2299 MHZ
MEMORY — Free: 11419 MB — Used: 4605 MB — Capacity: 16025 MB
STORAGE — Free: 60G — Used: 20G — Capacity: 81G

Common components

Name	Version	Status	
vPostgres		Running	Stop
RabbitMQ		Running	Stop

System-level components

Name	Version	Status	
SSH Service		Running	Stop

NSX Management Components

Name	Version	Status	
NSX Universal Synchronization Service		Stopped	Start
NSX Management Service	6.3.0 Build 5007049	Running	Stop

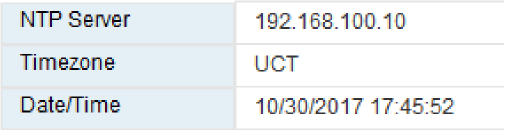
Figure 47. NSX Manager web interface.

NSX Manager Verifications

Processes that have to be running are: vPostgres, RabbitMQ and NSX Management Service.

The NSX Manager is divided in different sections that contain different parameters:

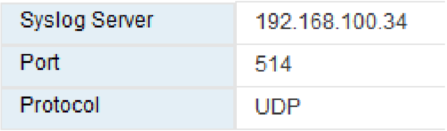
- Time Settings: NTP server is specified. For SSO configuration to work correctly it is required that the time on this virtual appliance and NTP server should be in sync. It is recommended to use the same NTP server used by the SSO server.



NTP Server	192.168.100.10
Timezone	UCT
Date/Time	10/30/2017 17:45:52

Figure 48. NSX Manager time settings.

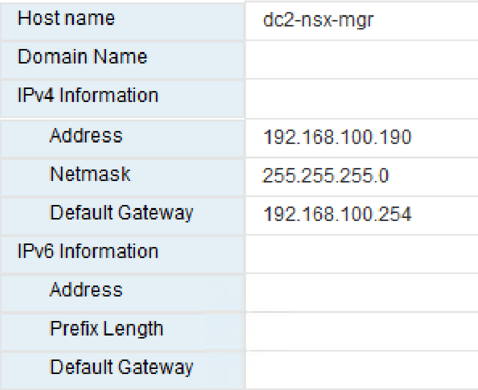
- Syslog Server: IP address of the syslog server.



Syslog Server	192.168.100.34
Port	514
Protocol	UDP

Figure 49. NSX Manager syslog server.

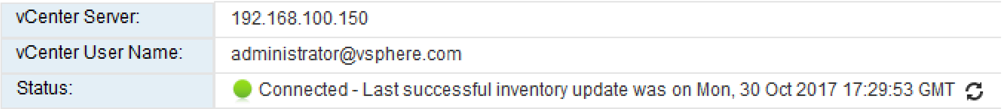
- Network: Verify that IP and Default Gateway are the correct ones.



Host name	dc2-nsx-mgr
Domain Name	
IPv4 Information	
Address	192.168.100.190
Netmask	255.255.255.0
Default Gateway	192.168.100.254
IPv6 Information	
Address	
Prefix Length	
Default Gateway	

Figure 50. NSX Manager network settings.

- NSX Management Service: Verify that vCenter Server Status is connected. Connecting to a vCenter server enables NSX Management Service to display the VMware Infrastructure inventory. HTTPS port (443) needs to be opened for communication between NSX Management Service, vESXi. It is important to ensure that appropriate CPU and memory reservation is given to this appliance VM. After successful configuration of vCenter on NSX Manager, it is needed to log out of any active client sessions on vSphere Web Client and log back in to enable NSX user interface components.



vCenter Server:	192.168.100.150
vCenter User Name:	administrator@vsphere.com
Status:	🟢 Connected - Last successful inventory update was on Mon, 30 Oct 2017 17:29:53 GMT

Figure 51. NSX Manager vCenter status.

Additionally, in NSX Manager some backups could be configured. They can be hourly, daily or weekly basis.

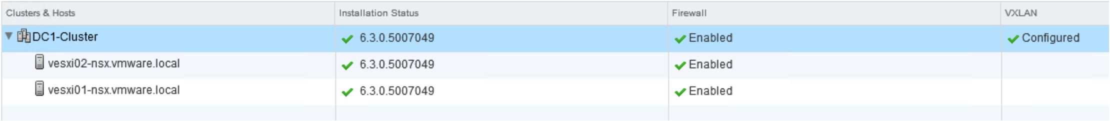
NSX Manager vSphere Web Client Verifications

First of all, it must be verified that NSX Manager VM is ON.

Then in Home → Network & Security → Installation, there are some verifications and configurations that are required for a properly behaviour of NSX management plane.

Host Preparation

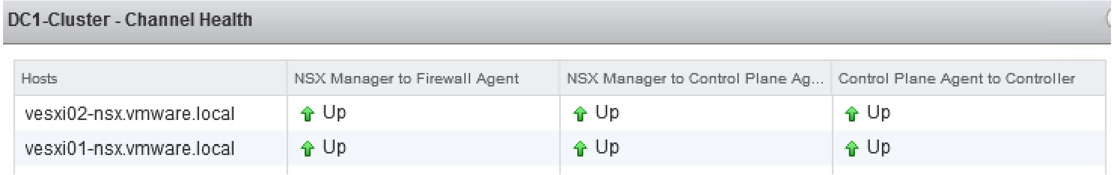
Depending on how much Managers are deployed, there are one or more roles, for the primary one, it must be verified that Installation Status, Firewall and VXLAN are enabled either configured.



Clusters & Hosts	Installation Status	Firewall	VXLAN
▼ DC1-Cluster	✓ 6.3.0.5007049	✓ Enabled	✓ Configured
vesxi02-nsx.vmware.local	✓ 6.3.0.5007049	✓ Enabled	
vesxi01-nsx.vmware.local	✓ 6.3.0.5007049	✓ Enabled	

Figure 52. Host preparation.

Moreover, right clicking on the DC1, there is the Channel Health that shows if the NSX Manager is Up with different agents (Firewall, Control Plane and Controller).



DC1-Cluster - Channel Health

Hosts	NSX Manager to Firewall Agent	NSX Manager to Control Plane Ag...	Control Plane Agent to Controller
vesxi02-nsx.vmware.local	⇧ Up	⇧ Up	⇧ Up
vesxi01-nsx.vmware.local	⇧ Up	⇧ Up	⇧ Up

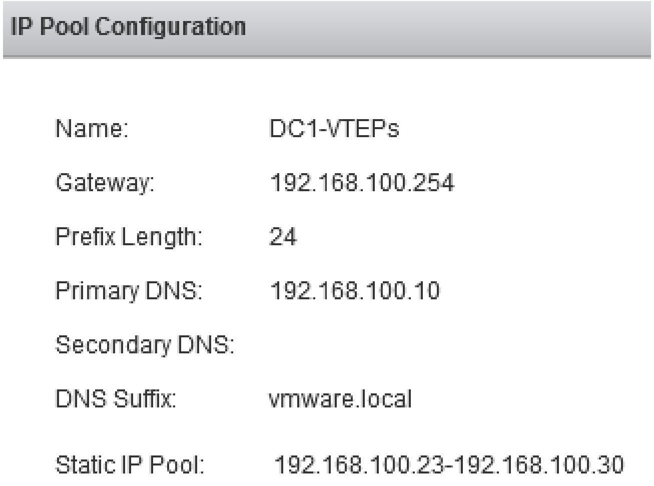
Figure 53. Channel Health.

Host preparation must be done as many times as roles exist.

Logical Network Preparation

The main purpose is to prepare properly the network for an NSX Manager. It stands for configuring the VXLAN, Segment ID and Transport Zones.

To prepare the VXLAN it is necessary to create an IP pool for each VMKNic, for this project it is:



IP Pool Configuration

Name: DC1-VTEPs

Gateway: 192.168.100.254

Prefix Length: 24

Primary DNS: 192.168.100.10

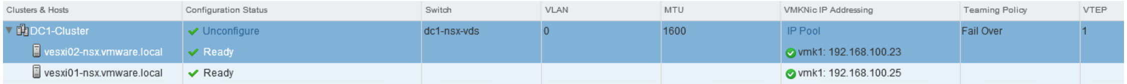
Secondary DNS:

DNS Suffix: vmware.local

Static IP Pool: 192.168.100.23-192.168.100.30

Figure 54. NSX Manager pool configuration.

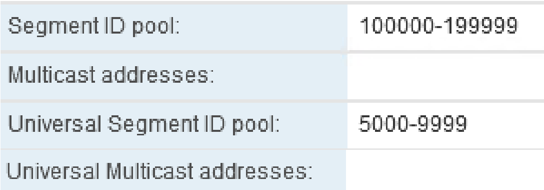
Then, this pool is applied for the VXLAN transport configuration. It is highly important to set MTU equal to 1600. Moreover there is only one VTEP per host, because it is virtual.



Clusters & Hosts	Configuration Status	Switch	VLAN	MTU	VMKNic IP Addressing	Teaming Policy	VTEP
▼ DC1-Cluster	✓ Unconfigure	dc1-nsx-vds	0	1600	IP Pool	Fail Over	1
vesxi02-nsx.vmware.local	✓ Ready				✓ vmk1: 192.168.100.23		
vesxi01-nsx.vmware.local	✓ Ready				✓ vmk1: 192.168.100.25		

Figure 55. NSX Manager VXLAN transport configuration.

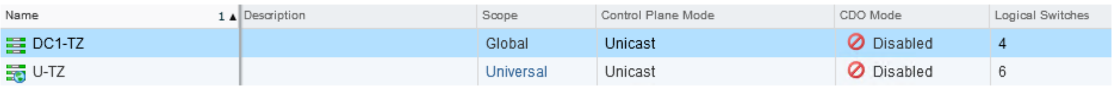
Moreover, a Segment ID has to be configured, it specifies a range of VNIs for use when building network segments.



Segment ID pool:	100000-199999
Multicast addresses:	
Universal Segment ID pool:	5000-9999
Universal Multicast addresses:	

Figure 56. NSX Manager segment ID.

Finally, transport Zones where any logical switches created in a TZ will automatically be added to the clusters selected here. It is important to remember that, a transport zone specifies the hosts and clusters that are associated with logical switches that are created in the zone; this process is similar to manually adding hosts to the distributed switch.



Name 1 ▲	Description	Scope	Control Plane Mode	CDO Mode	Logical Switches
DC1-TZ		Global	Unicast	⊘ Disabled	4
U-TZ		Universal	Unicast	⊘ Disabled	6

Figure 57. NSX Manager Transport Zones.

With previous steps and recommendations, the deployment of NSX Manager can be concluded. Now, the scenario is ready to deploy NSX Controllers.

3.2.2. NSX Controller(s)

The NSX controller is a distributed management system that provide control plane functions for NSX logical switching and routing functions. It serves as the central control point for all logical witches within a network while maintaining information about all hosts, logical switches and VXLANs, and distributed logical routers. In summary, the NSX Controller provides the following benefits:

- VXLAN and logical routing network information distribution to vESXi hosts.
- High availability.
- Workload distribution among VMware NSX Controller cluster nodes.
- Maintenance of tables for VXLAN and distributed logical routers. For VXLAN are VTEP, MAC and ARP tables and for DLRs are routes, logical interfaces, ARP and MACs needed in the DLR bridging.

Controllers are strictly necessary if distributed logical routers or VXLANs are deployed. Depending if the scenario is in cross-vCenter, when the NSX Manager is assigned the primary role, its controller cluster becomes the universal one for the whole cross-vCenter NSX environment.

Regarding on the number of NSX Controllers it is important to deploy more than one controller for contingency. In a production environment it is recommended that the 3 NSX Controllers reside on different hosts (vESXi/ESXi) with DRS anti-affinity rules to avoid multiple failure of NSX Controllers due to host outage.

As it is commented in this section, workloads are distributed across NSX Controller cluster nodes. There are several slices and defined objects that are going to be sliced; Logical Switches and Logical Routers.

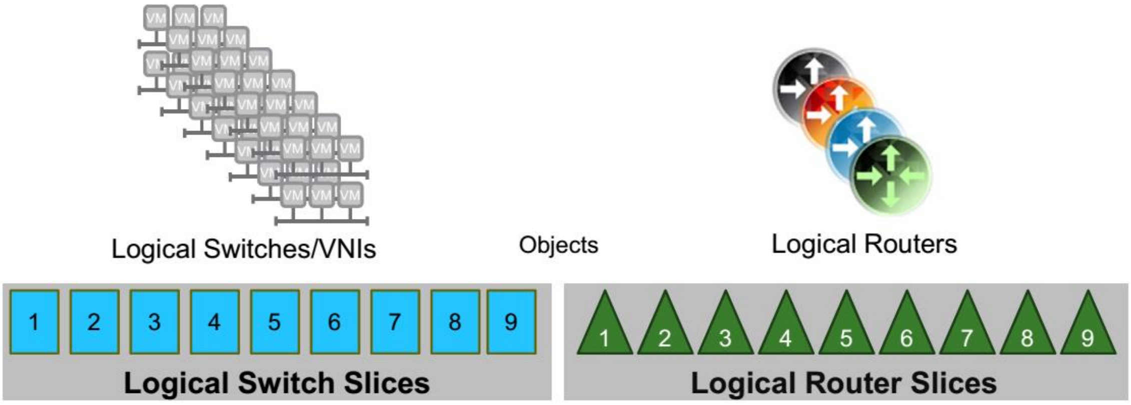


Figure 58. Logical Switches and Logical Routers sliced.

After slicing these network components, each slice is assigned to an NSX Controller. Controllers can have multiple slices. An example of workloads distributed in an scenario with 3 controllers may be like the next figure:

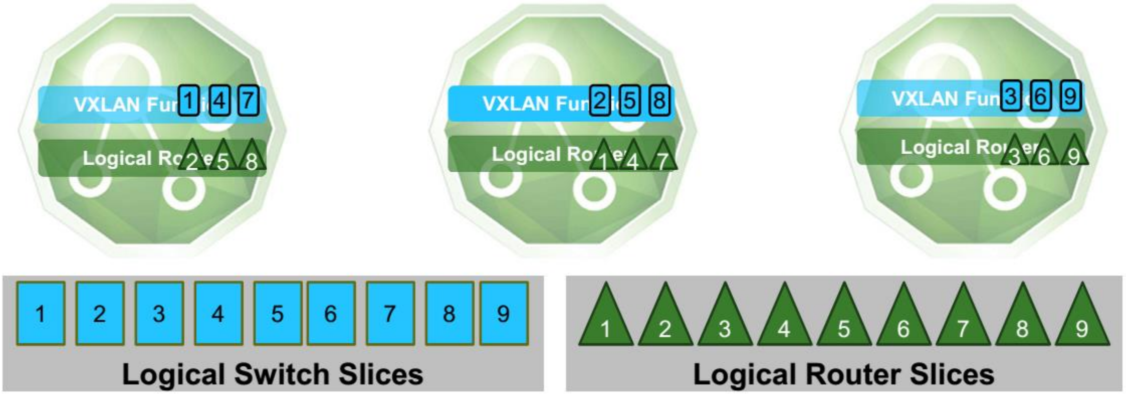


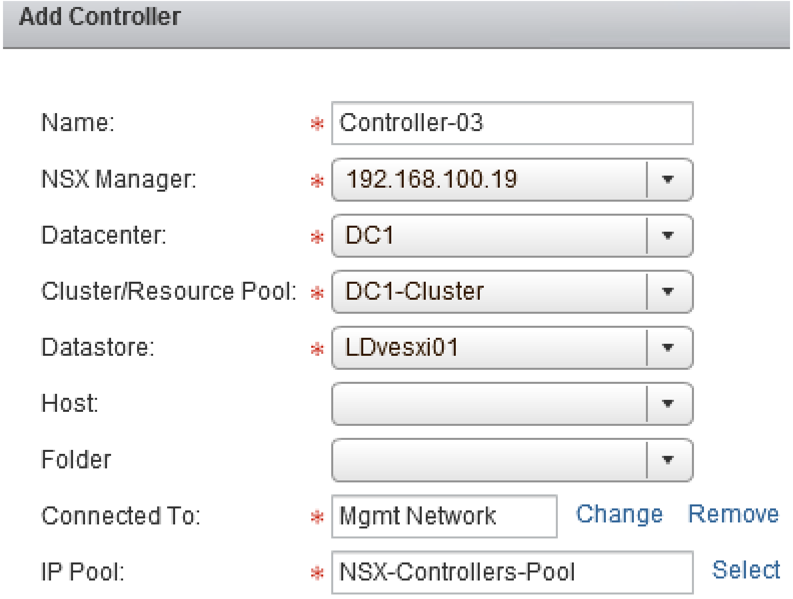
Figure 59. Distributed slices among NSX Controllers.

If a Controller fails, the slices are redistributed among the remaining nodes. Redistribution it is also use when the number of controllers increases or decreases.

NSX Controller deployment

Before to start, it has to be mentioned that NSX Controller clusters must be deployed in the same vCenter system where NSX Manager is connected.

De deployment is in: Home → Networking & Security → Installation → Management → NSX Controller Nodes. When adding a Controller, administrators configure the following options:


Add Controller

Name:	* Controller-03	
NSX Manager:	* 192.168.100.19	
Datacenter:	* DC1	
Cluster/Resource Pool:	* DC1-Cluster	
Datastore:	* LDvesxi01	
Host:		
Folder		
Connected To:	* Mgmt Network	Change Remove
IP Pool:	* NSX-Controllers-Pool	Select

Figure 60. NSX Controller deployment parameters.

From the previous Figure, it can be appreciated that these options correspond to the controller number 3, which is associated with the primary Manager (the secondary manager has 192.168.100.190 IP Address). In addition, this controller is placed in DC1 as well as in cluster and Datastore 1. An important detail is to attach the controller to the management port group, the one intended to connect management elements. Finally, there is an IP pool for controllers, needed for having more than one controller.

Due to design purposes, the third controller, showed before, is not included in the current scenario; there is only one controller deployed. NSX Controllers appear as VM in the VM inventory.

Once the NSX Controller(s) is deployed, the administrator has to ensure that all deployed nodes have the normal status and they are managed by same NSX Manager.

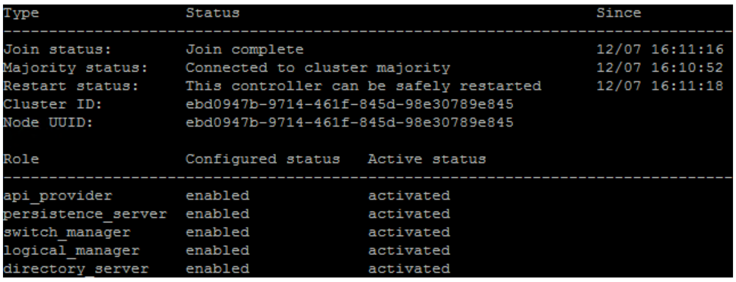
Typical deployment issues:

- Insufficient resources. One NSX Controller uses these resources: 4 CPUs, Memory 4096MB and a Hard Disk of 20GB.
- DNS misconfigured in vESXi hosts, vCenter Server and NSX Manager.
- Connectivity between NSX Manager and NSX Controller.

NSX Controller Verifications

There are some verifications that have to be performed to ensure NSX Controller connectivity, active services.

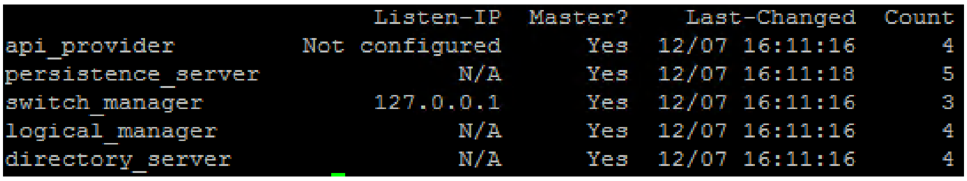
Accessing via PuTTY and entering the command `show control-cluster status` shows which status are activated and configurated:



```
Type                 Status                                     Since
--------------------------------------------------------------------------------
Join status:         Join complete                              12/07 16:11:16
Majority status:     Connected to cluster majority              12/07 16:10:52
Restart status:      This controller can be safely restarted    12/07 16:11:18
Cluster ID:          ebd0947b-9714-461f-845d-98e30789e845
Node UUID:           ebd0947b-9714-461f-845d-98e30789e845

Role                 Configured status   Active status
--------------------------------------------------------------------------------
api_provider         enabled             activated
persistence_server   enabled             activated
switch_manager       enabled             activated
logical_manager      enabled             activated
directory_server     enabled             activated
```

Figure 61. NSX Controller status.

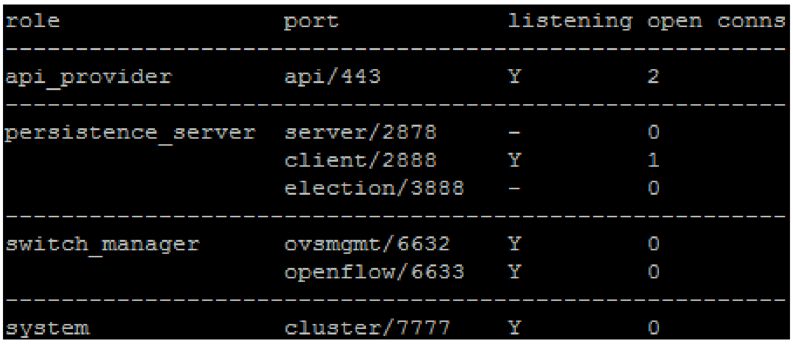
Restart status is important to be like in the previous figure. As an extra comment, there are 5 roles well configured. To see them in better detail, the command used is `show control-cluster roles`.



```
                          Listen-IP  Master?    Last-Changed  Count
api_provider         Not configured      Yes  12/07 16:11:16      4
persistence_server              N/A      Yes  12/07 16:11:18      5
switch_manager            127.0.0.1      Yes  12/07 16:11:16      3
logical_manager                 N/A      Yes  12/07 16:11:16      4
directory_server                N/A      Yes  12/07 16:11:16      4
```

Figure 62. NSX Controller roles.

Another interesting command is `show control-cluster connections`, that specifies the components actively listening on a network port. In this case are the following ones:



```
role                port            listening open conns
--------------------------------------------------------
api_provider        api/443         Y          2
--------------------------------------------------------
persistence_server  server/2878     -          0
                    client/2888     Y          1
                    election/3888   -          0
--------------------------------------------------------
switch_manager      ovsmgmt/6632    Y          0
                    openflow/6633   Y          0
--------------------------------------------------------
system              cluster/7777    Y          0
```

Figure 63. NSX Controller connections.

3.2.3. Logical Switching

A logical switch performs same functions as a physical one besides it is made by software, so, it is virtualized.

Transport Zones have a vital role when dealing with vSwitches; they specify the hosts and clusters that are associated with logical switches that are created in the zone. The administrator can create as much as transport zones as she needs.

The NSX logical switch creates logical broadcast domains or segments to which an application or tenant virtual machine can be logically wired.

Logical switching offers several benefits:

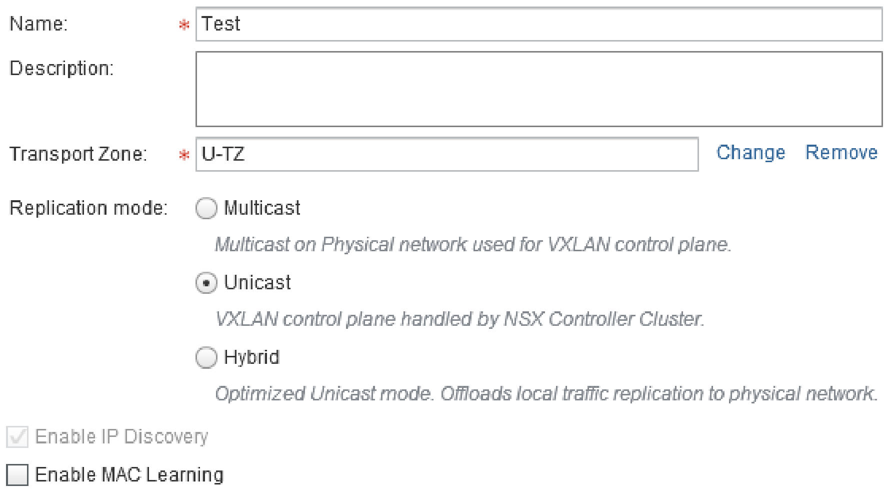
- Enable L2 over L3 infrastructure.
- VXLAN based overlay: logical networks are decoupled from physical ones.
- Scalable multitenancy across the data centre.
- Reduces VLAN ID usage.

There are 4 steps to follow to prepare a virtual switch for being used:

- Create vSwitch.
- Attach vSwitch to an NSX Services Gateway or logical router.
- Attach VM to a vSwitch.
- Verify connectivity between VMs attached to a vSwitch.

vSwitch Creation

Through Home navigation pane → Networking & Security → Logical Switches and Add. The parameters required for the creation of the logical switch are three: a name, TZ that will belong and replication mode that can be Multicast, Unicast or Hybrid.



Name: * Test

Description:

Transport Zone: * U-TZ

Replication mode:

○ Multicast

Multicast on Physical network used for VXLAN control plane.

● Unicast

VXLAN control plane handled by NSX Controller Cluster.

○ Hybrid

Optimized Unicast mode. Offloads local traffic replication to physical network.

☑ Enable IP Discovery

□ Enable MAC Learning

Figure 64. vSwitch parameters.

Unicast replication mode overloads the host that performs the ARP request, so this mode is recommended to use for small and medium environments. Besides, multicast mode only sends ARP requests to those VMs that joined, so, VMs with VTEPs. It is less efficient for SDN, due to not all VMs are discovered. Finally the Hybrid mode that combines both modes: multicast is used inside of a segment without needing multicast routing, and unicast out of the segment. Hybrid mode is commonly used in big environments.

As it can be seen IP Discovery is enabled; IP Discovery enables ARP suppression because there is a globally shared ARP table although ARP replies are unicast. The ARP flow in NSX is the following one:

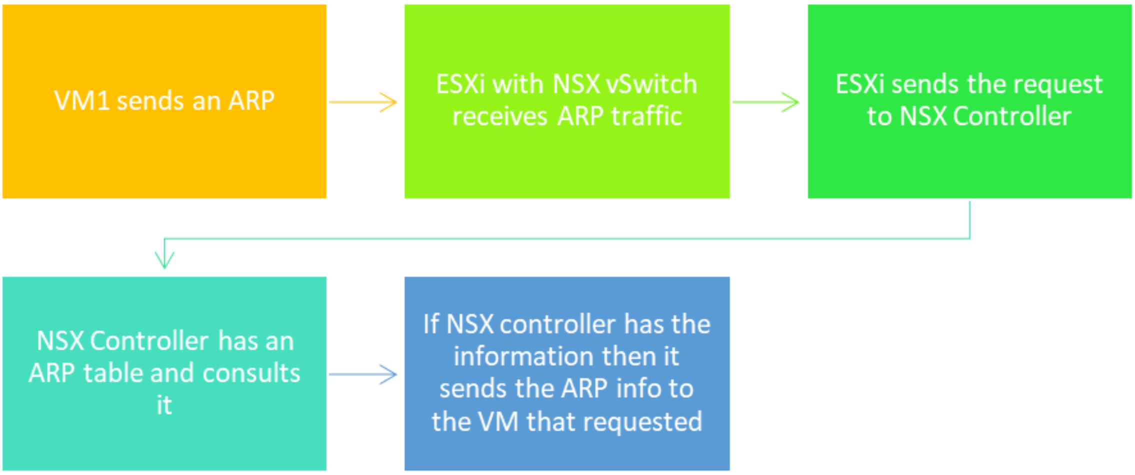

VM1 sends an ARP
ESXi with NSX vSwitch receives ARP traffic
ESXi sends the request to NSX Controller
NSX Controller has an ARP table and consults it
If NSX controller has the information then it sends the ARP info to the VM that requested

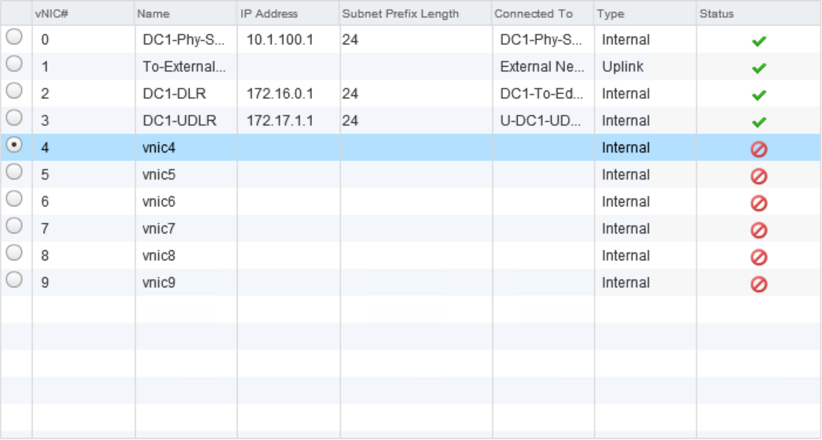
Figure 65. NSX ARP Suppression.

The VM performs an ARP Request but if the controller has the information, it answers the information to the requesting VM without forwarding the packet. Then, the requesting VM only has to update its ARP table.

Attaching vSwitch to a NSX Edge services gateway or logical router

Supposing that an Edge services gateway or a logical router is already deployed, both procedures will be explained lately, the new virtual switch has to be attached to one of commented elements.

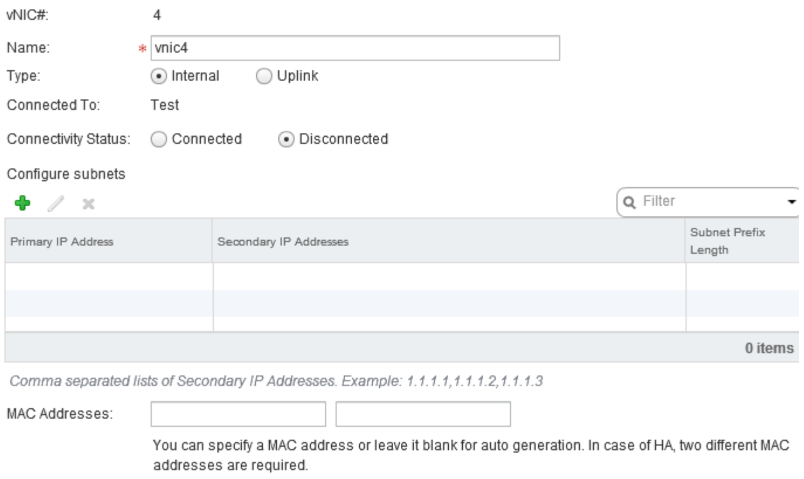
To attach the switch to an Edge, the administrator has to select the vSwitch, right click, Connect Edge and then, select the NSX Edge. A pop-up window appears asking for which Edge or DLR has to be attached to the vSwitch. Then, it appears the following screen showing which Edge interfaces are free:



	vNIC#	Name	IP Address	Subnet Prefix Length	Connected To	Type	Status
○	0	DC1-Phy-S...	10.1.100.1	24	DC1-Phy-S...	Internal	✓
○	1	To-External...			External Ne...	Uplink	✓
○	2	DC1-DLR	172.16.0.1	24	DC1-To-Ed...	Internal	✓
○	3	DC1-UDLR	172.17.1.1	24	U-DC1-UD...	Internal	✓
●	4	vnic4				Internal	⊘
○	5	vnic5				Internal	⊘
○	6	vnic6				Internal	⊘
○	7	vnic7				Internal	⊘
○	8	vnic8				Internal	⊘
○	9	vnic9				Internal	⊘

Figure 66. Available Edge or DLR interfaces.

After selecting an interface, there are some parameters that have to be configured; for example, the interface name, the type of the switch and subnets.



vNIC#: 4

Name: * vnic4

Type: ● Internal ○ Uplink

Connected To: Test

Connectivity Status: ○ Connected ● Disconnected

Configure subnets

Primary IP Address	Secondary IP Addresses	Subnet Prefix Length

0 items

Comma separated lists of Secondary IP Addresses. Example: 1.1.1.1,1.1.1.2,1.1.1.3

MAC Addresses:

You can specify a MAC address or leave it blank for auto generation. In case of HA, two different MAC addresses are required.

Figure 67. Parameters to be set when attaching a vSwitch to an Edge or DLR.

For the current project the parameters are:

- vSwitch name: App
- Type: Internal[8]
- Subnet: 172.16.0.1/24. Components belonging to vSwitch App use 172.16.20.x. 172.16.0.1 is the address where VMs from App switch send packets that have to be forwarded to another IP domain.

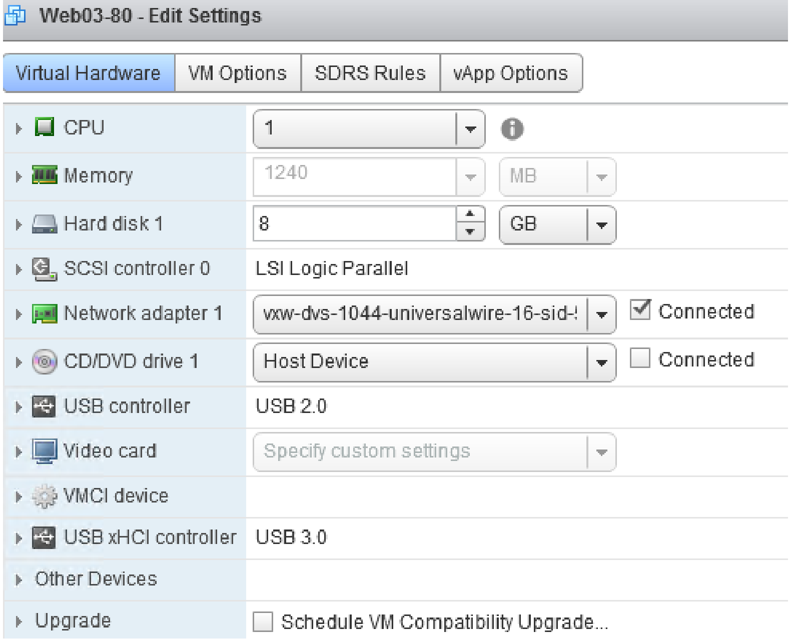
When vSwitch is attached to an Edge/DLR, the scenario is ready to attach VMs to that vSwitch.

Attaching VMs to a vSwitch

There are two ways to attack a VM to a vSwitch, as they are similar, only one it is going to be explained.

Before doing that, the administrator has to ensure that the VM network adapter is enabled and connects to the correct vSwitch/vDistributed Switch. To do that, in VMs and templates, right click on the VM and select Edit Settings. The network adapter has to be connected and the universal wire has to be the vSwitch where the VM will be attached.

[8] Type can be internal or external. Internal means that all components belonging to the vSwitch are part of the virtual network.



Web03-80 - Edit Settings

- CPU: 1
- Memory: 1240 MB
- Hard disk 1: 8 GB
- SCSI controller 0: LSI Logic Parallel
- Network adapter 1: vxw-dvs-1044-universalwire-16-sid-! ☑ Connected
- CD/DVD drive 1: Host Device □ Connected
- USB controller: USB 2.0
- Video card: Specify custom settings
- VMCI device
- USB xHCI controller: USB 3.0
- Other Devices
- Upgrade: □ Schedule VM Compatibility Upgrade...

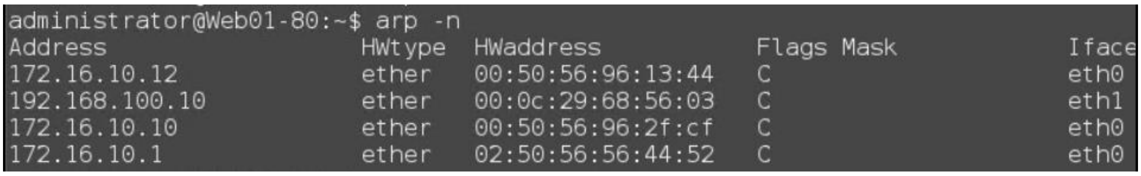
Figure 68. VM Edit Settings.

The VM is ready to be attached so, the admin navigates to Networking & Security → Logical Switches → Right click al Switch → Add VM → Select VM → Select VM vNICs for being attached and Ready to complete.

Verify connectivity between VMs attached to a vSwitch

Before doing any ping between VM machines, it has to be verified if VM vNICs are up and routing tables have entries. To see VM entries, enter via PuTTy to VM and execute `arp -n`.

The arp table example is from VM-web-01:



```
administrator@Web01-80:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.16.10.12             ether   00:50:56:96:13:44   C                     eth0
192.168.100.10           ether   00:0c:29:68:56:03   C                     eth1
172.16.10.10             ether   00:50:56:96:2f:cf   C                     eth0
172.16.10.1              ether   02:50:56:56:44:52   C                     eth0
```

Figure 69. VM Web-01 ARP table.

The routes showed correspond to:

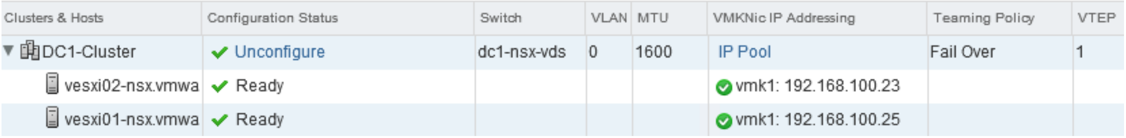
IP Address	VM
172.16.10.12	VM Web-02 (attached to same vSwitch as Web-01)
192.168.100.10	Domain Controller
172.16.10.10	VM Web-03 (temporary attached to same vSwitch as Web-03)
172.16.10.1	DLR interface viewed from vSwitch.

Table 5. Web-01 ARP table.

Due to Web-01 and Web-02, pings are supposed to work because they belong to the same subnet, besides the ping between Web-02 and App-01 (attached to another vSwitch) will not work as there are no firewall rules allowing this traffic.

Verify connectivity between VTEPs and vESXi hosts

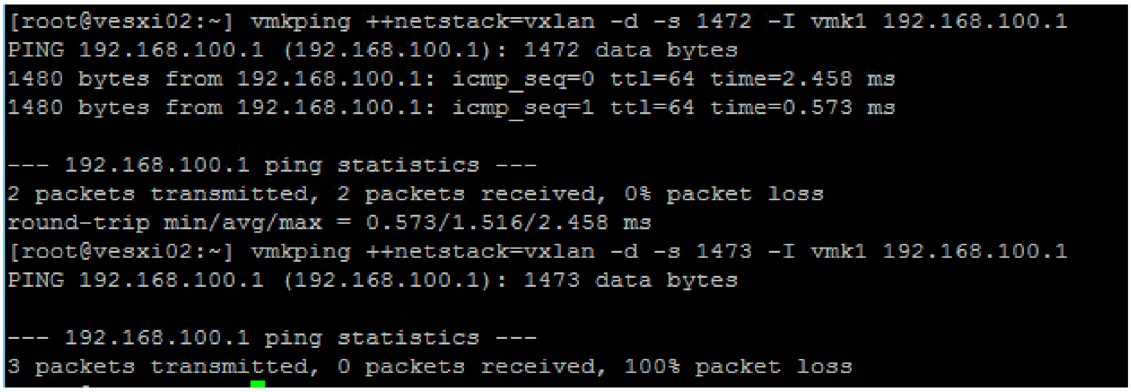
Inside of DC1-Cluster there are 2 vESXis (vesxi01 and vesxi02), they have a unique VMKNic each one from DC1 IP pool and for primary NSX Manager.



Clusters & Hosts	Configuration Status	Switch	VLAN	MTU	VMKNic IP Addressing	Teaming Policy	VTEP
▼ DC1-Cluster	✓ Unconfigure	dc1-nsx-vds	0	1600	IP Pool	Fail Over	1
vesxi02-nsx.vmwa	✓ Ready				✓ vmk1: 192.168.100.23		
vesxi01-nsx.vmwa	✓ Ready				✓ vmk1: 192.168.100.25		

Figure 70. vESXI's VMKnic.

Through these IP addresses, vESXIs can be accessed with PuTTy. Once the administrator has entered, for example in vESXi-02 it can test connectivity to the other vESXi (not VMKnic!), vesxi-01.



```
[root@vesxi02:~] vmkping ++netstack=vxlan -d -s 1472 -I vmk1 192.168.100.1
PING 192.168.100.1 (192.168.100.1): 1472 data bytes
1480 bytes from 192.168.100.1: icmp_seq=0 ttl=64 time=2.458 ms
1480 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=0.573 ms

--- 192.168.100.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.573/1.516/2.458 ms
[root@vesxi02:~] vmkping ++netstack=vxlan -d -s 1473 -I vmk1 192.168.100.1
PING 192.168.100.1 (192.168.100.1): 1473 data bytes

--- 192.168.100.1 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
```

Figure 71. VMKping from vESXi-02 to vESXi-01.

Conclusions are extracted from this vmkping:

- Although accessing via PuTTy to vESXi vMKNic IP Address, the vmkping is performed to vESXi-01 IP address (192.168.100.1).
- MTU is an important parameter, the vmkping works with 1472 Bytes but not for 1473.

- If the ping is successful, this indicates that VTEPs from each vESXI can communicate with each other and the physical network is properly configured to support VXLAN frames (remember that VTEP stands for VXLAN tunnel end point).
- When frames travel from an ESXi to another there is an encapsulation and decapsulation of frames. In L2, VTEP adds a header to go through L3 that when the frame enters to the other ESXi and downs to L2, the frame is decapsulate without suffering any change.

3.2.4. Logical Routers

Logical routers provide East-West[9] connectivity between VMs that although they belong to different networks they reside inside the virtual environment, also they provide North-South[10] connectivity to access public networks.

As commented in State of the Art, there are three elements that provide logical routing: Distributed Logical Router, Universal Distributed Logical Router or NSX Edge Services Gateway. The device selected depends on where is placed and what it offers:

- DLR or UDLR: They are in charge to provide East-West routing. They support distributed[11] and dynamic routing[12]. It allows to route traffic between logical switches. The main difference between them is that the UDLR is used for routing across several domains.
- NSX Edge Services Gateway: it is commonly used as a perimeter to the “external world”. It has dynamic routing capability; moreover, it provides network services such as DHCP, NAT, Load Balancer, Firewall and VPN.

NSX routing supports OSPF and BGP routing protocols.

The current scenario has one distributed logical router, one universal distributed logical router and 2 NSX Edge Services Gateways (indicated in Figure 46 as NSX EDGE), both performing routing. Due to DLR and UDLR deployments are similar, only the explanation of the deployment of the DLR and its configuration is included,

DLR Deployment

The first step for creating a routing element is the same: through the navigation pane, NSX Edges and then add. After that, a window opens asking for the name of the element and the installation type (DLR, UDLR or Edge Services Gateway). During the creation it is important to enable SSH, this way it can be controlled and monitored via CLI.

One of the most important steps during the deployment is when configuring its interfaces. It has to be said that more interfaces can be added lately.

[9] East-west traffic refers to traffic within a data centre, i.e. VM to VM.

[10] North-south traffic is client to server traffic, the one between the data centre and the rest of the network, that includes anything outside the data centre.

[11] Distributed routing: Allows the communication between two virtual machines that belong to different vSwitches without having to pass through the physical world.

[12] Dynamic Routing: it is a process where a router can forward data via a different route or given destination based on the current conditions of the network.

For example, interface that goes from DLR to DC1-Web vSwitch, it is represented in next figure:

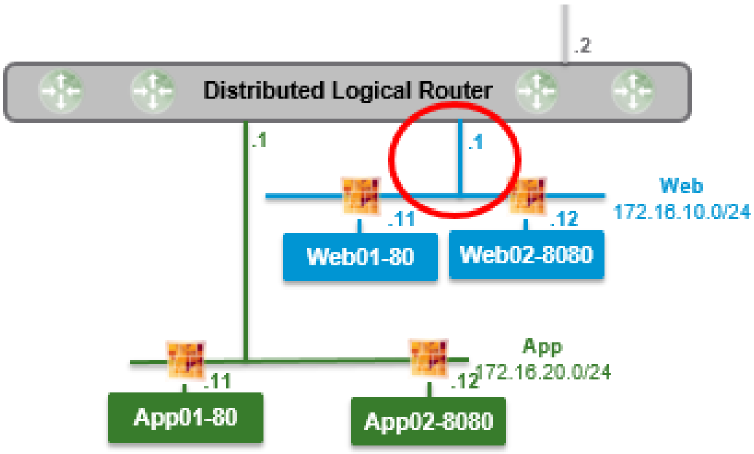
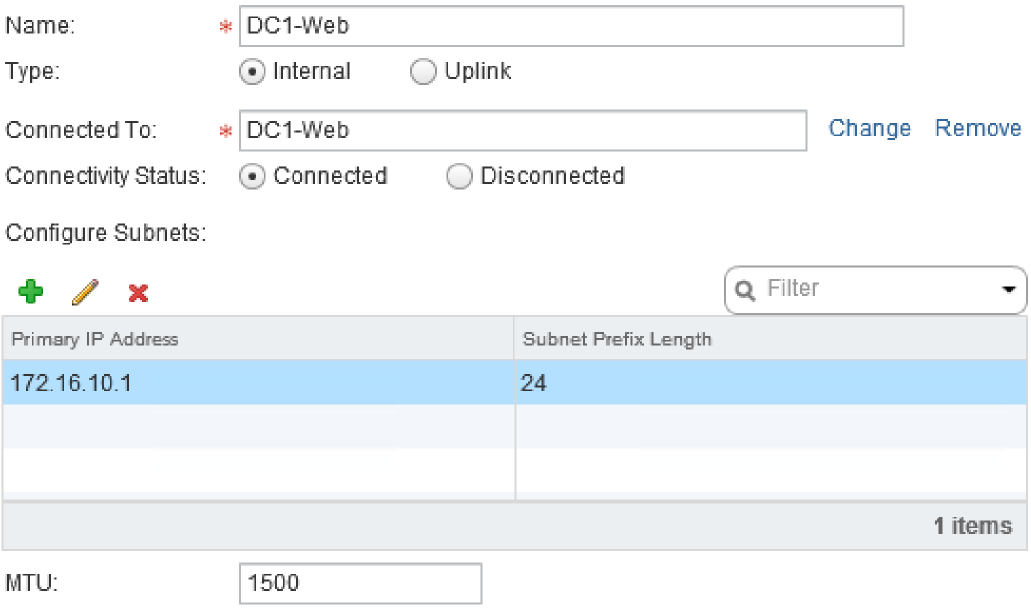


Figure 72. Router interface from DLR to DC1-Web vSwitch.

And the configuration is:



Name:	* DC1-Web
Type:	● Internal ○ Uplink
Connected To:	* DC1-Web
Connectivity Status:	● Connected ○ Disconnected
Configure Subnets:	

Primary IP Address	Subnet Prefix Length
172.16.10.1	24

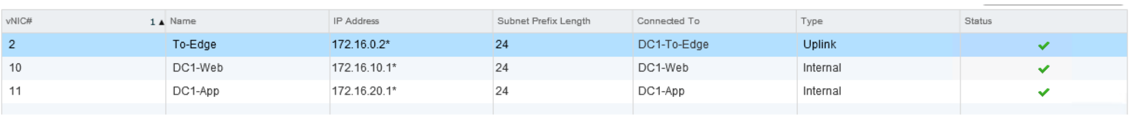
1 items

MTU: 1500

Figure 73. DLR Interface example.

As it can be seen, the primary IP address is the same than the one in the principal network map. Moreover, the “Connected To” field refers to which vSwitch is this interface connected. The MTU is important to be set to 1500 Bytes.

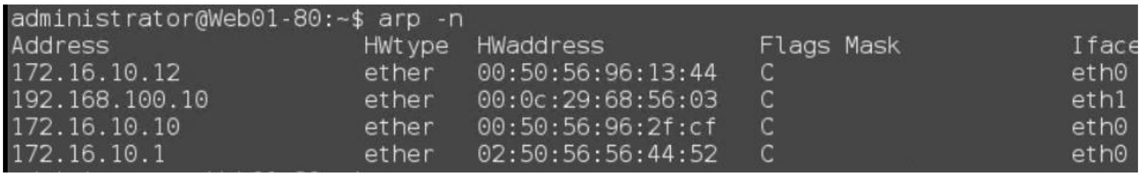
But this router does not only have one interface, it has a total number of three:



vNIC#	Name	IP Address	Subnet Prefix Length	Connected To	Type	Status
2	To-Edge	172.16.0.2*	24	DC1-To-Edge	Uplink	✓
10	DC1-Web	172.16.10.1*	24	DC1-Web	Internal	✓
11	DC1-App	172.16.20.1*	24	DC1-App	Internal	✓

Figure 74. DLR Interfaces.

After the creation, the DLR appears as a VM in the inventory. Now, its corresponding IP address will appear when performing `arp -n` in VM machines currently attached to vSwitches that are in the interfaces of the DLR. For example, Web-01 that is attached to vSwitch DC1-Web and this vSwitch belongs to an interface of the DLR.



```
administrator@Web01-80:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
172.16.10.12             ether   00:50:56:96:13:44   C                     eth0
192.168.100.10           ether   00:0c:29:68:56:03   C                     eth1
172.16.10.10             ether   00:50:56:96:2f:cf   C                     eth0
172.16.10.1              ether   02:50:56:56:44:52   C                     eth0
```

Figure 75. Web-01 ARP table.

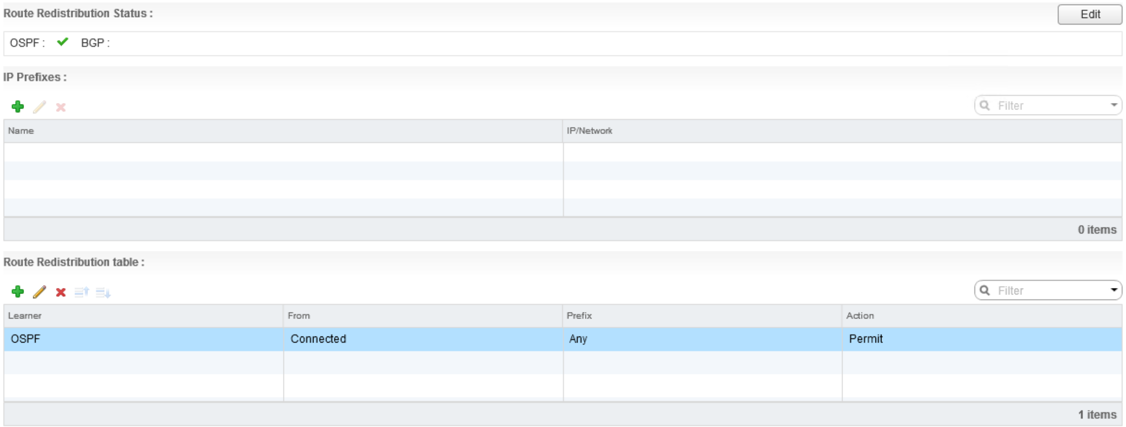
After enabling a FW rule from Web-01 to App-01, the ping can be performed successfully.

Routing protocols and default gateway

The question now is, which is its default gateway? Is there any routing protocol enabled?

Through the navigation pane Manage → Settings → Routing; here it can be found different ways to configure routing: static routes, OSPF, BGP and also route redistribution.

The default gateway of DLR is 172.16.0.1, this IP address corresponds to the Edge one. For this DLR there are no static routes as OSPF will be used. Moreover, route redistribution is enabled too as it is shown in next figure:



Route Redistribution Status :

OSPF : ✓ BGP :

IP Prefixes :

Name	IP/Network

0 items

Route Redistribution table :

Learner	From	Prefix	Action
OSPF	Connected	Any	Permit

1 items

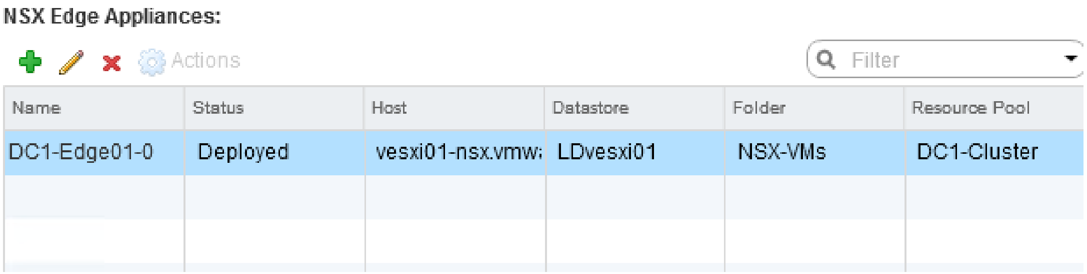
Figure 76. DLR Routing Configuration.

This concludes the deployment of the distributed logical router.

NSX Edge Services Gateway Deployment

As previously explained, an Edge has more functions than a simple DLR/UDLR; for example: DHCP, NAT, Load Balancer, Firewall and VPN.

The EDGE must be deployed with the NSX Edge appliance. After creating the NSX Edge following the same steps as for the router, this is how the NSX Edge appliance looks:



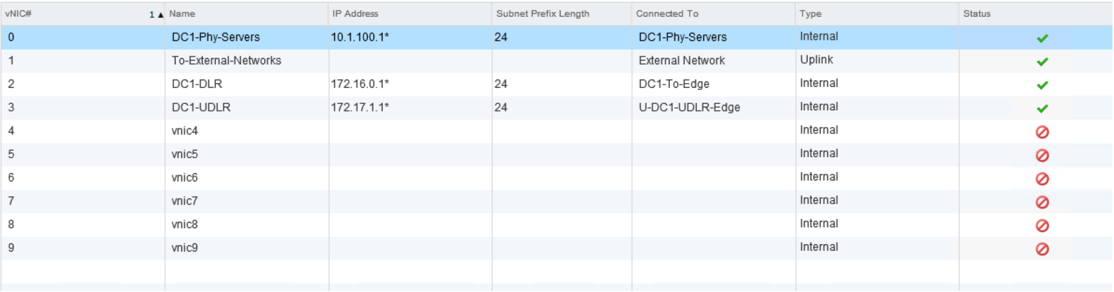
NSX Edge Appliances:

Name	Status	Host	Datastore	Folder	Resource Pool
DC1-Edge01-0	Deployed	vesxi01-nsx.vmw:	LDvesxi01	NSX-VMs	DC1-Cluster

Figure 77. NSX Appliance setup.

Due to the actual Edge belongs to different domains, it has to be taken into account which of the NSX Manager roles are being configured.

And now, the interfaces that are configured:

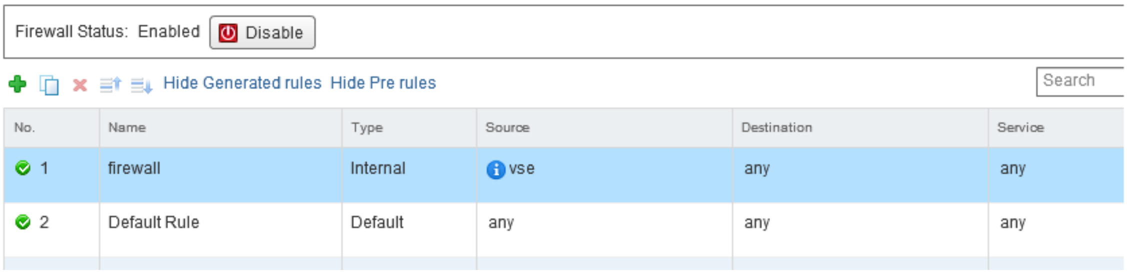


vNIC# 1 ▲	Name	IP Address	Subnet Prefix Length	Connected To	Type	Status
0	DC1-Phy-Servers	10.1.100.1*	24	DC1-Phy-Servers	Internal	✓
1	To-External-Networks			External Network	Uplink	✓
2	DC1-DLR	172.16.0.1*	24	DC1-To-Edge	Internal	✓
3	DC1-UDLR	172.17.1.1*	24	U-DC1-UDLR-Edge	Internal	✓
4	vnic4				Internal	⊘
5	vnic5				Internal	⊘
6	vnic6				Internal	⊘
7	vnic7				Internal	⊘
8	vnic8				Internal	⊘
9	vnic9				Internal	⊘

Figure 78. NSX Edge configured interfaces.

From the previous image it can be concluded that the number of interfaces is limited to 10. The MTU parameter it is set to 1500 for virtual interfaces and 1600 if they are physical ones.

Firewall in NSX Edge is enabled allowing all traffic. This actual environment does not require block rules due to there are no critical elements nor databases inside of it.

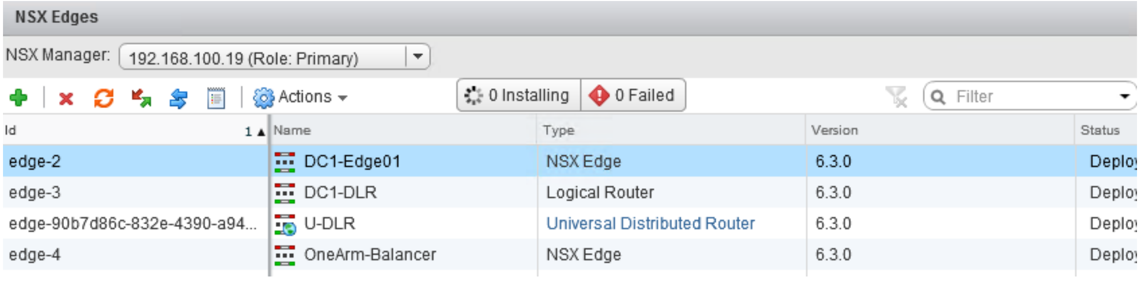


Firewall Status: Enabled

No.	Name	Type	Source	Destination	Service
✓ 1	firewall	Internal	vse	any	any
✓ 2	Default Rule	Default	any	any	any

Figure 79. NSX Edge firewall rules.

Once all deployment is done, it is recommended to verify that the configuration is correct.


NSX Edges

NSX Manager: 192.168.100.19 (Role: Primary)

Actions | 0 Installing | 0 Failed | Filter

Id	Name	Type	Version	Status
edge-2	DC1-Edge01	NSX Edge	6.3.0	Deploy
edge-3	DC1-DLR	Logical Router	6.3.0	Deploy
edge-90b7d86c-832e-4390-a94...	U-DLR	Universal Distributed Router	6.3.0	Deploy
edge-4	OneArm-Balancer	NSX Edge	6.3.0	Deploy

Figure 80. NSX Edges deployed in the actual scenario.

It is confirmed that DC1-Edge01 the type is NSX Edge.

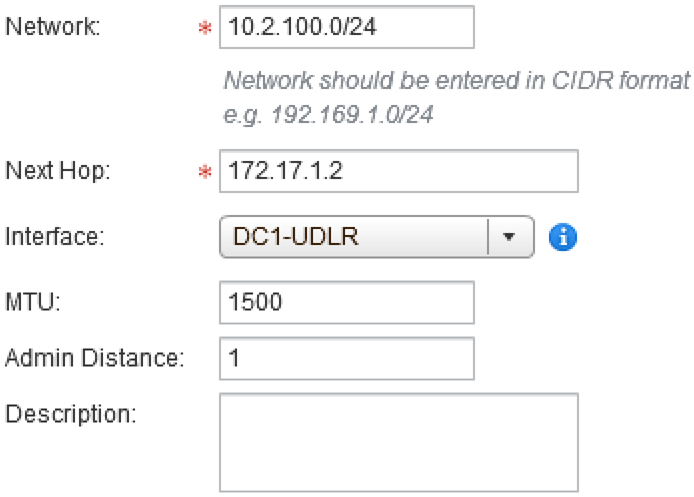
Routing protocols, dynamic and static routes

Static Routes

Dynamic and static routes can be configured in edges and routers, depending on network requirements.

A network can be robust and stable with OSPF routes; although, static routes could be included to reach other network segments. From principal network scheme, it can be deduced that the Edge services gateway has both: dynamic and static routes. In this section it is shown how routes have been configured.

For example, the route 10.2.100.0/24 -> 172.17.1.2. Through the navigation pane when entering to the desired Edge, in configuration → Static routes the route must be added this way:


Network: * 10.2.100.0/24

Network should be entered in CIDR format e.g. 192.169.1.0/24

Next Hop: * 172.17.1.2

Interface: DC1-UDLR

MTU: 1500

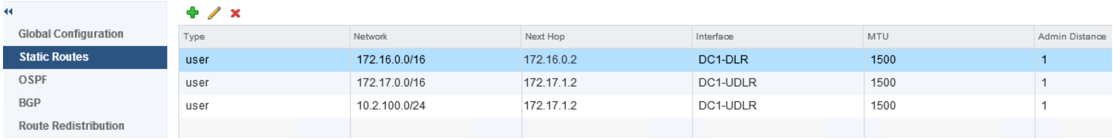
Admin Distance: 1

Description:

Figure 81. NSX Edge Static Route configuration.

It is important to describe an origin network, on the contrary, for destination it is required a unique hop. Regarding on the interface, it is the one where the traffic will go through. Remember that MTU is 1500Bytes.

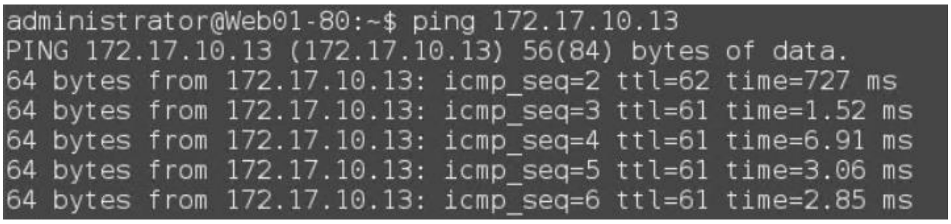
After configuring all static routes this is the result:



Type	Network	Next Hop	Interface	MTU	Admin Distance
user	172.16.0.0/16	172.16.0.2	DC1-DLR	1500	1
user	172.17.0.0/16	172.17.1.2	DC1-UDLR	1500	1
user	10.2.100.0/24	172.17.1.2	DC1-UDLR	1500	1

Figure 82. NSX Edge Static Routes.

To test that static routes work properly, a ping from Web-01 to Web-03 is done successfully:



```
administrator@Web01-80:~$ ping 172.17.10.13
PING 172.17.10.13 (172.17.10.13) 56(84) bytes of data.
64 bytes from 172.17.10.13: icmp_seq=2 ttl=62 time=727 ms
64 bytes from 172.17.10.13: icmp_seq=3 ttl=61 time=1.52 ms
64 bytes from 172.17.10.13: icmp_seq=4 ttl=61 time=6.91 ms
64 bytes from 172.17.10.13: icmp_seq=5 ttl=61 time=3.06 ms
64 bytes from 172.17.10.13: icmp_seq=6 ttl=61 time=2.85 ms
```

Figure 83. Ping from Web-01 to Web-03 to test static routes configured in the NSX Edge.

The previous test validates that static routes are well configured.

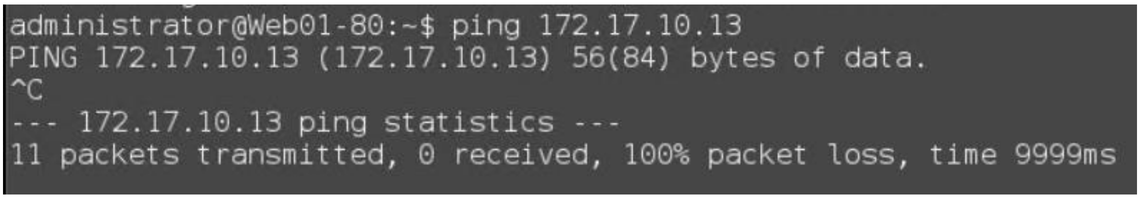
Sometimes can happen that although OSPF is disabled, route redistribution OSPF is enabled. This implies that the routes are static but that they are redistributed via OSPF.

Dynamic Routes

Before starting dynamic routes configuration, it is important to describe the concept of routing Areas.

Routing Area: routing is divided into different areas to optimize the traffic. An area is a logical collection of OSPF networks, routers and links that have the same area identification.

To test that OSPF works properly static routes configured previously have to be deleted. Now, the ping between Web-01 and Web-03 does not work:

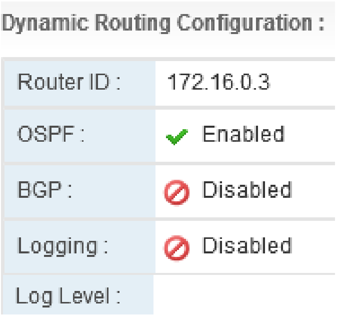


```
administrator@Web01-80:~$ ping 172.17.10.13
PING 172.17.10.13 (172.17.10.13) 56(84) bytes of data.
^C
--- 172.17.10.13 ping statistics ---
11 packets transmitted, 0 received, 100% packet loss, time 9999ms
```

Figure 84. Ping from Web-01 to Web-03 failed because routes are not configured.

The steps for configuring dynamic routes are described in detail because are kind of complex:

1) First of all, OSPF must be enabled in NSX Edge and DLR. It is done by going through Edge → Manage → Routing → Global Configuration and enter a router ID and enable OSPF.



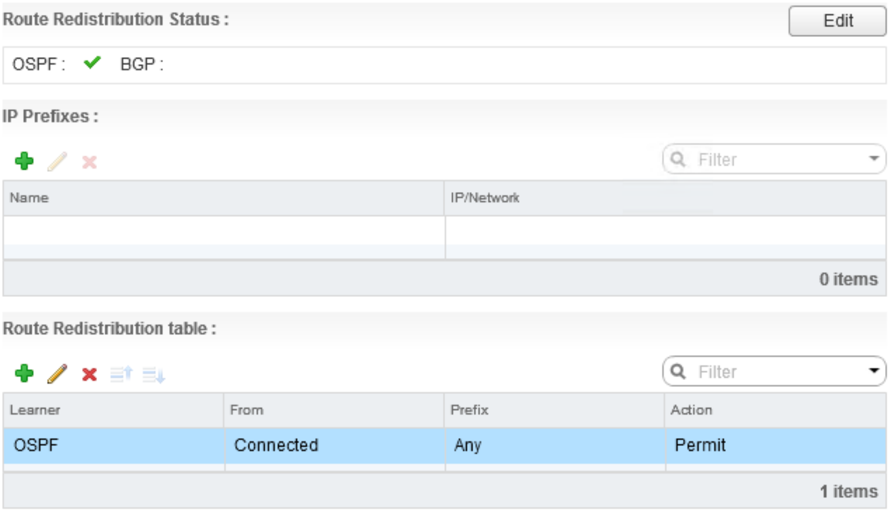
Dynamic Routing Configuration :

Router ID :	172.16.0.3
OSPF :	✓ Enabled
BGP :	⊘ Disabled
Logging :	⊘ Disabled
Log Level :	

Figure 85. Enabling OSPF in NSX Edge.

2) Create a Routing Area in addition to the 0 that already exists. Edge → Manage → Routing → OSPF. For example, new area is 51.

3) Enable route redistribution status and redistribution table. Edge → Manage → Routing → Route Redistribution.



Route Redistribution Status :

OSPF : ✓ BGP :

IP Prefixes :

Name	IP/Network

0 items

Route Redistribution table :

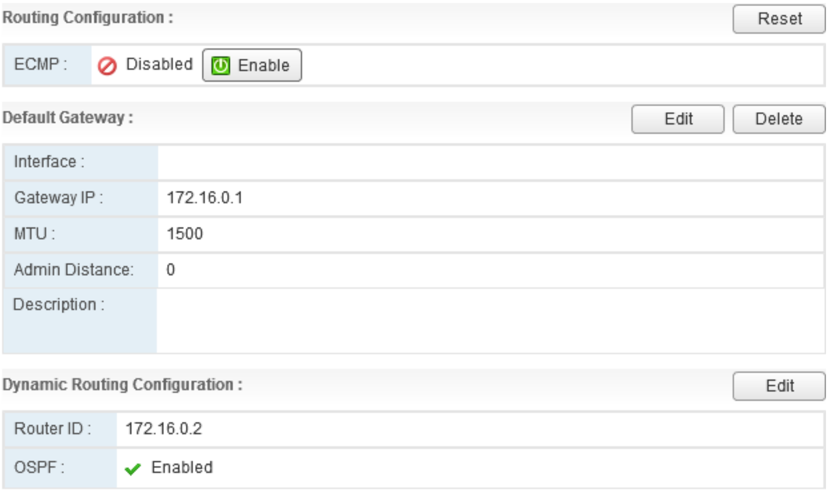
Learner	From	Prefix	Action
OSPF	Connected	Any	Permit

1 items

Figure 86. Routing Area and Route Redistribution Table.

It important to highlight that route redistribution table learns OSPF and its status is "Connected", which means that subnets that are connected to the perimetral Gateway can be learned.

4) Although this point is about NSX Edge, if dynamic routes are configured in the Edge they must be also configured in the DLR/UDLR. When enabling OSPF in DLR the steps are the same as for the NSX Edge, these steps are Enable OSPF and add a Router ID:



Routing Configuration : Reset

ECMP :	Disabled Enable

Default Gateway : Edit Delete

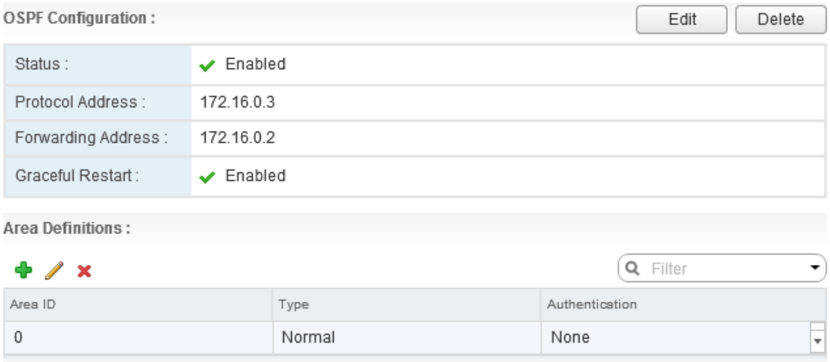
Interface :	
Gateway IP :	172.16.0.1
MTU :	1500
Admin Distance:	0
Description :	

Dynamic Routing Configuration : Edit

Router ID :	172.16.0.2
OSPF :	✓ Enabled

Figure 87. Enabling OSPF in DLR.

5) Finish to enable OSPF and add both protocol and forwarding addresses, also to set Area 0 as routing area. Protocol address = 172.16.0.2 (uplink interface where Edge sends routes); forwarding address = Routing ID.



OSPF Configuration : Edit Delete

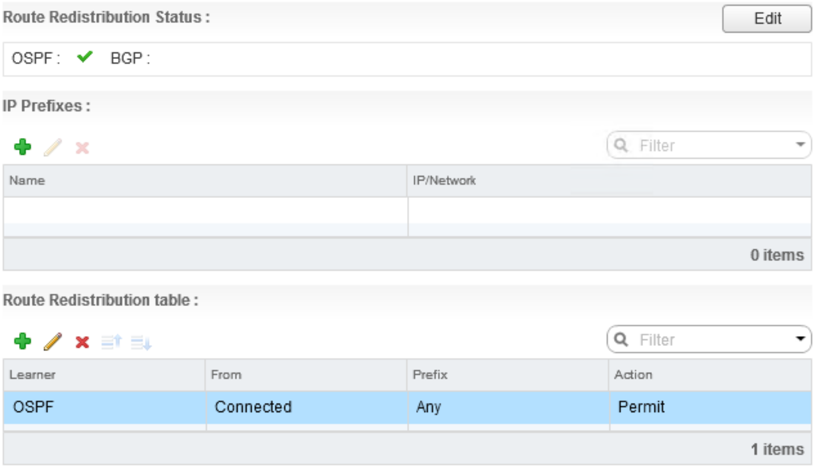
Status :	✓ Enabled
Protocol Address :	172.16.0.3
Forwarding Address :	172.16.0.2
Graceful Restart :	✓ Enabled

Area Definitions :

Area ID	Type	Authentication
0	Normal	None

Figure 88. Setting OSPF parameters in DLR.

6) Configure route redistribution: OSPF must be included.


Route Redistribution Status :

OSPF : ✓ BGP :

IP Prefixes :

Name	IP/Network

0 items

Route Redistribution table :

Learner	From	Prefix	Action
OSPF	Connected	Any	Permit

1 items

Figure 89. Configuring Route Redistribution in DLR.

Once dynamic routes are configured in both devices (Edge + DLR) Web-01 knows how to arrive to Edge. If the user wants to reach Web-03, same steps used in the DLR configuration have to be applied to UDLR too. To demonstrate the previous configuration it is enough this test:

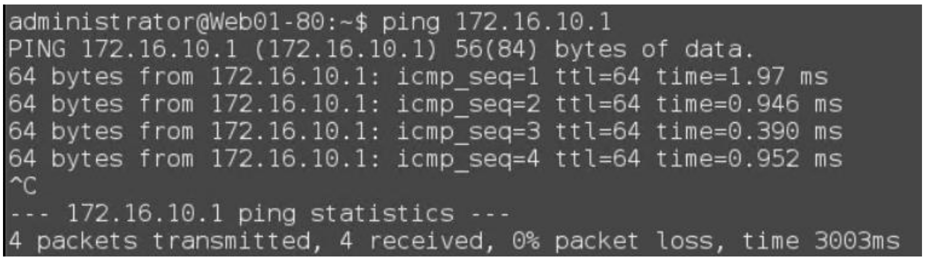

```
administrator@Web01-80:~$ ping 172.16.10.1
PING 172.16.10.1 (172.16.10.1) 56(84) bytes of data.
64 bytes from 172.16.10.1: icmp_seq=1 ttl=64 time=1.97 ms
64 bytes from 172.16.10.1: icmp_seq=2 ttl=64 time=0.946 ms
64 bytes from 172.16.10.1: icmp_seq=3 ttl=64 time=0.390 ms
64 bytes from 172.16.10.1: icmp_seq=4 ttl=64 time=0.952 ms
^C
--- 172.16.10.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
```

Figure 90. Verifying OSPF configuration.

For the project scenario, it is seen that when traffic flows from Web-01 to App-01, packets arrive until DLR; in contrast, traffic from App-01 to App-03 has to pass through DLR, then, Edge and, finally, UDLR.

3.2.5. Other components

This section contains other functions that NSX offers that are useful not only to control network elements also for monitoring, a help for future administrators. For example, NSX Manager Health and Process.

Logical Firewall

A logical firewall is a tool that provides security for dynamic virtual data centers. There are two different kinds of firewall in an NSX Environment: Distributed Firewall that is intended for East-West traffic and, Edge Firewall, focused on the North-South traffic for datacentre perimeter.

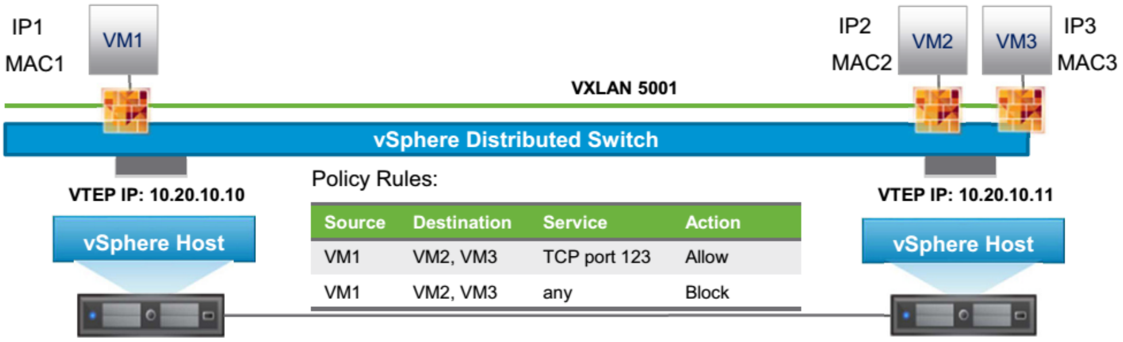
In Software-Defined data centers, security services are managed more efficiently. It is easy to deploy, in terms of provision; also, it is easy to apply security policies and finally automation.

Apart from the logical firewall provided by VMware, the integration with third-party services is easy and fast, Palo Alto Networks is an example: they provide an extra firewall.

There are some features that NSX Firewalls offer such us microsegmentation because DFW ensures that every vNIC is subject to policy processing at ingress and egress. Moreover, policies are based on vCenter objects, not only source IP and destination IP, vSwitches, datacenters, VM names are other parameters that can define a firewall rule.

DFW can offer a high line rate, about 20Gbps per host.

A Firewall rule is, for example, the following one: There are 3 VM that belong to same distributed switch and network. The traffic between VM1 to VM2 and VM3 it is only allowed for tcp/123, and the rest is blocked. In this case, there are many ways to define source or destination, using IP addresses or VM names (VM1, VM2, VM3…).



Policy Rules:

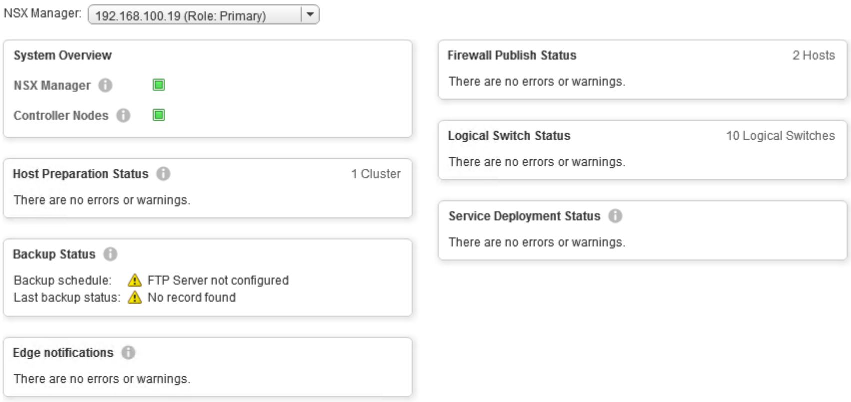
Source	Destination	Service	Action
VM1	VM2, VM3	TCP port 123	Allow
VM1	VM2, VM3	any	Block

Figure 91. DFW rule example.

During the scenario deployment, an error in these rules caused to lose the access to the scenario, which is going to be explained in the following section in order not to repeat this configuration.

Manager Health and Process

It is a functionality from vSphere Web Client that allows the administrator to get an instant view of the overall health of the NSX environment. The dashboard alerts about any potential issues with the NSX Manager, Controllers, Hosts, Firewall and logical switches.



NSX Manager: 192.168.100.19 (Role: Primary)

System Overview
NSX Manager 🟩
Controller Nodes 🟩

Host Preparation Status — 1 Cluster
There are no errors or warnings.

Backup Status
Backup schedule: ⚠ FTP Server not configured
Last backup status: ⚠ No record found

Edge notifications
There are no errors or warnings.

Firewall Publish Status — 2 Hosts
There are no errors or warnings.

Logical Switch Status — 10 Logical Switches
There are no errors or warnings.

Service Deployment Status
There are no errors or warnings.

Figure 92. NSX Manager Health and Process.

If the mouse is placed in the icon “i” close to NSX Manager, it is showed the status and disc usage of NSX Manager besides the running state of services currently in use.

Doing the same but for Controller Nodes, the info displayed is each controller status and its connectivity between NSX Manager and controller peers. In addition to that, the overall health of a controller.

3.3. Deployment Problems

3.3.1. DHCP Problem

The initial configuration in the first shield of virtualization, the one that paths the physical world to the virtual one was erroneous. The hop machine and the 3 vESXi were implemented in the same level, connected to the physical servers using same PortGroups which caused a DHCP propagation problem.

DHCP requests from the virtual world travelled to the physical world causing a DHCP problem in the physical router. This physical router was not only used to access the server, but also for company users to access to the Internet.

The connections that surround the physical router were the corrupted ones. In order to comprehend better the situation, the connections scheme is showed:

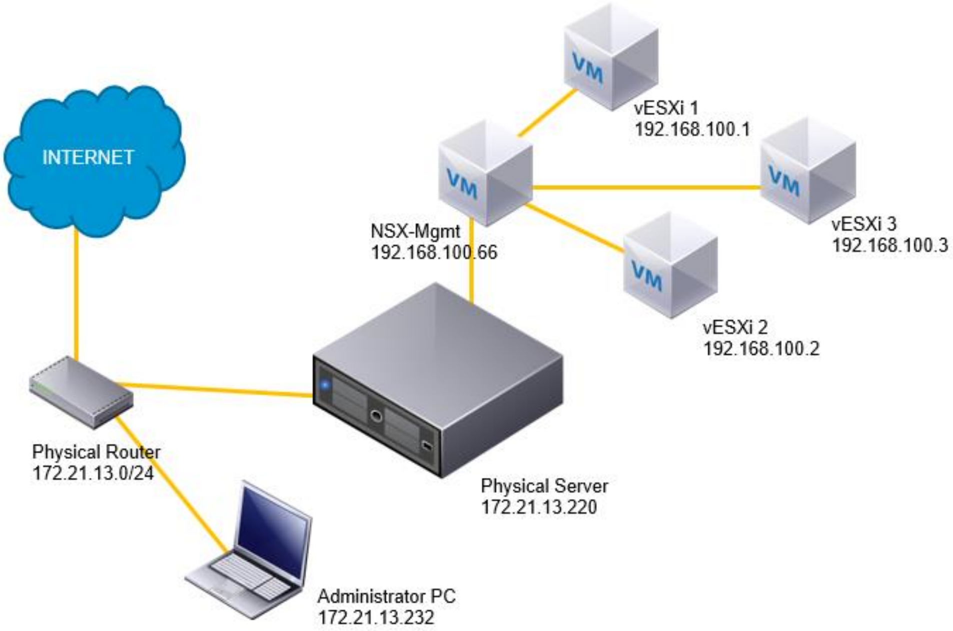
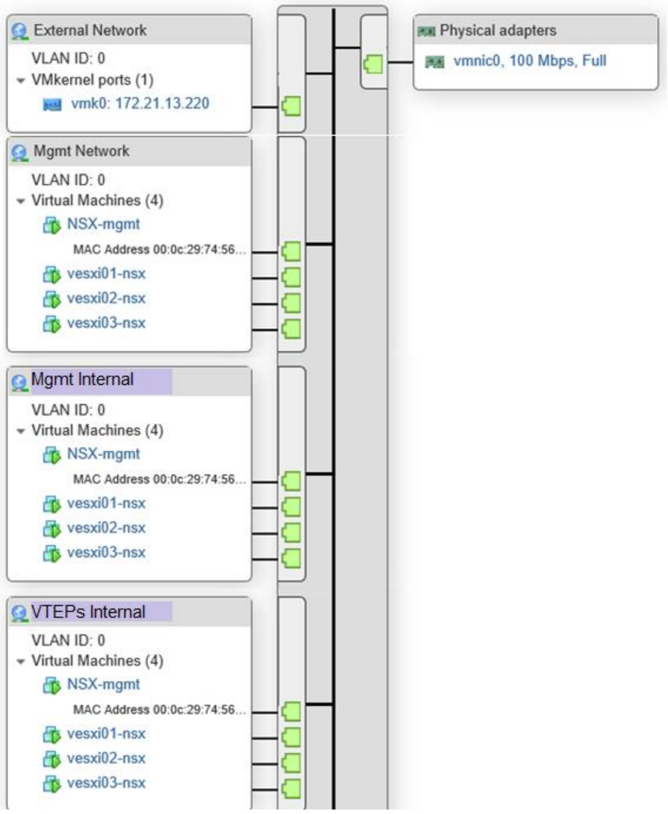


Figure 93. Environment connections.

The Hypervisor connections were:



External Network
VLAN ID: 0
VMkernel ports (1)
vmk0: 172.21.13.220

Physical adapters
vmnic0, 100 Mbps, Full

Mgmt Network
VLAN ID: 0
Virtual Machines (4)
NSX-mgmt
MAC Address 00:0c:29:74:56...
vesxi01-nsx
vesxi02-nsx
vesxi03-nsx

Mgmt Internal
VLAN ID: 0
Virtual Machines (4)
NSX-mgmt
MAC Address 00:0c:29:74:56...
vesxi01-nsx
vesxi02-nsx
vesxi03-nsx

VTEPs Internal
VLAN ID: 0
Virtual Machines (4)
NSX-mgmt
MAC Address 00:0c:29:74:56...
vesxi01-nsx
vesxi02-nsx
vesxi03-nsx

Figure 94. Erroneous hypervisor logical switch configuration.

These connections were bad designed in first place due to a misunderstanding, for this reason, there was a unique vSwitch instead of two.

The major consequence was the physical router assigning dynamic IPs from the VMware Environment to the company users because VMware DHCP requests were faster than the ones performed by the physical router.

This configuration it is not valid if the physical router is shared for different kind of purposes. Although, it is a good configuration if the physical router is dedicated and only used for the VMware Server and the PCs accessing it.

There are two ways to fix this problem, besides one is temporary and the other no.

The temporary way consists on disabling the DHCP virtual server that sends the requests. It is a server inside vESXi1. It is a temporary solution because these requests are necessary for the virtual network.

The ultimate solution consists on decoupling the hypervisor vSwitch as it is explained in Hypervisor Networking section, to isolate the virtual world to the physical one.

3.3.2. DFW Problem

Taking into account that rules applied in DFW are distributed around the network, all traffic flows are over control. A simple deny all at the end of the DFW causes a general failover if properly rules are not applied.

This happened during a use case, while the only traffic allowed was from DC1-Web vSwitch to DC1-Web vSwitch and other flows should be blocked.

The thing is that after allowing the mentioned traffic, a default deny rule was set. Ten seconds later access to VMs and servers were down causing a general failover.

All of the scenario had to be deployed again. Thanks to snapshots of vESXi, the task was fast and easier.

Recommendation: Be careful with default deny rules.

3.3.3. Hypervisor datastores full of snapshots

The hypervisor contains all vESXi because in the current scenario everything it is virtualized. The available storage of the physical server after installing the hypervisor was 1.63TB, a reasonable value considering the actual scenario.

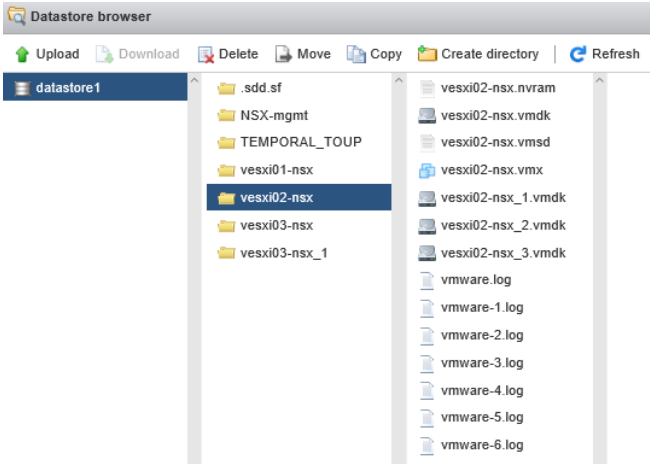
After having some network and deployment problems, it was decided to do snapshots of virtual machines while deploying and experimenting.

An snapshot is a detailed table of contents that provides the user accessible copies of data that can roll back; in other words, it is a copy of a virtual machine. When deploying a virtual machine from an snapshot, configurations and virtual machine states are those in the moment of performing the snapshot.

Snapshots are considered security copies of virtual machines; when use cases were deployed, some errors forced to re-deploy virtual machines from existing snapshots. A problem in physical datastores was noticed when they were almost 100% full.

After looking through the virtual environment and discarding other options, it was found, in the hypervisor, that its datastore (datastore1) contained too many versions and too big snapshot files from vESXi's. An snapshot is composed by several files; if one of those files is deleted accidentally, it might corrupt the snapshot.

The next figure shows an excessive quantity of files belonging to vESXi2:



datastore1

.sdd.sf
NSX-mgmt
TEMPORAL_TOUP
vesxi01-nsx
vesxi02-nsx
vesxi03-nsx
vesxi03-nsx_1

vesxi02-nsx.nvram
vesxi02-nsx.vmdk
vesxi02-nsx.vmsd
vesxi02-nsx.vmx
vesxi02-nsx_1.vmdk
vesxi02-nsx_2.vmdk
vesxi02-nsx_3.vmdk
vmware.log
vmware-1.log
vmware-2.log
vmware-3.log
vmware-4.log
vmware-5.log
vmware-6.log

Figure 95. Excessive files from vESXi2 which collapsed storage in physical server.

The problem was that there were too many files that were not deleted automatically, so they should be deleted manually. The fact was that there was no way to figure which files could be removed and which not. That results in deleting files that should not be removed and all the scenario was corrupted. The solution was to deploy again all the network with each network component and reconfiguring snapshots.

4. Use Cases

As previously mentioned, included in the thesis, there are some use cases that have been developed. For a better understanding, these use cases have been divided into two subsections: those that deal with networks and those ones with security.

Every Use Case has its own configuration, which is going to be presented later with its results and final recommendation.

Before to start, the network diagram is attached:

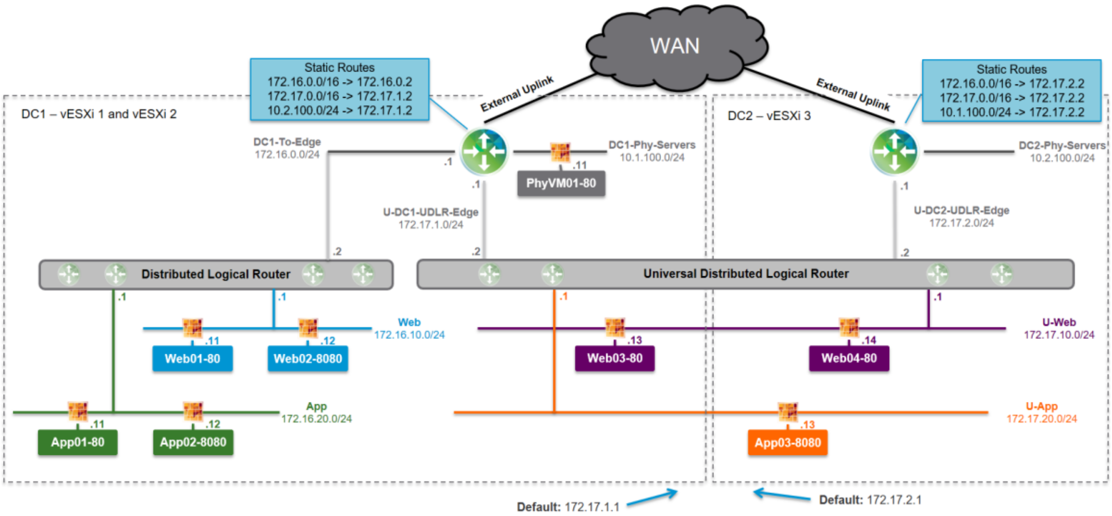


Figure 96. Network Architecture.

4.1. Networking Use Cases

Networking use cases are those ones intended to modify the network to obtain better results or specific behaviours for an specific configuration. There is a use case that includes a new NSX functionality for the current version. The networking use cases are the following ones:

- Creation of a Virtual Network.
- Cross vCenter: Migration of a VM between vCenters.
- Load Balancer Deployment.
- Native bridging. (New functionality)

4.1.1. Creation of a virtual network

In companies it is common to have different VLANs to segregate users, for example in the bank there are different departments like: Backoffice, Technology and Systems, Cybersecurity... but also, the network has multiple interfaces to separate different kinds of machines, AIX, Murex… This use case is intended to demonstrate how easily it is to create and propagate a virtual network.

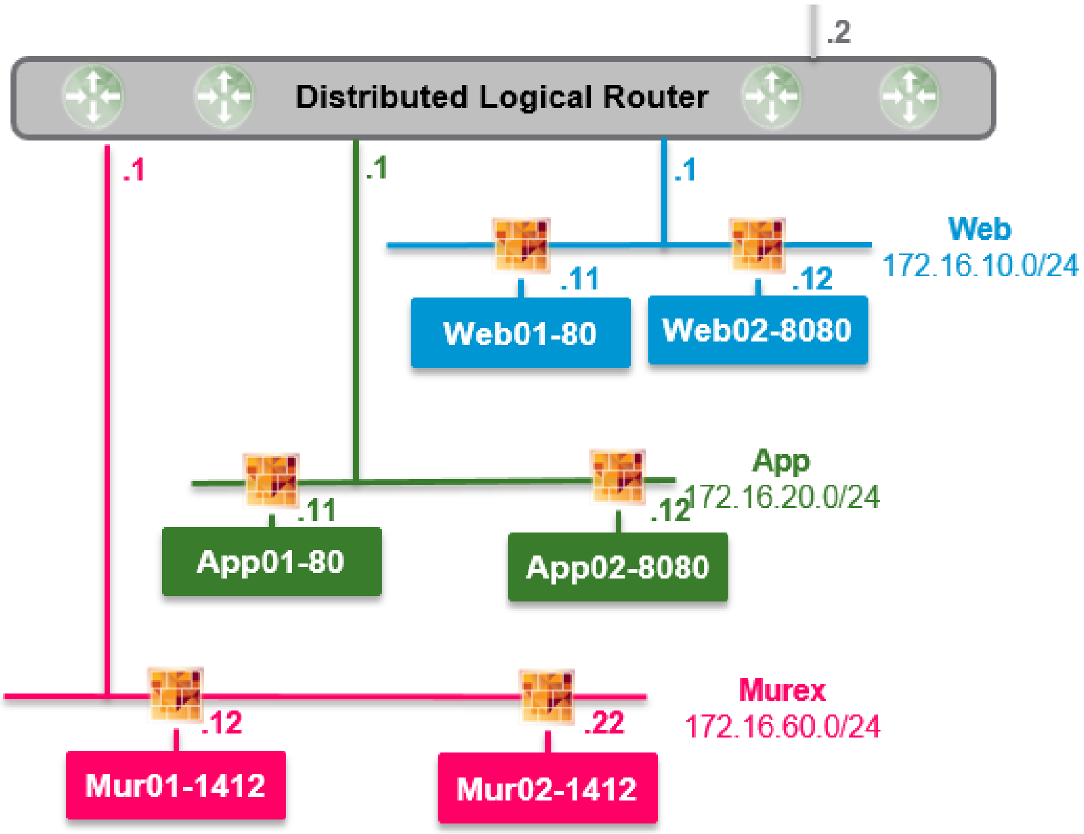


Figure 97. New Murex VLAN.

The figure above represents how a new network, the pink one (Murex), it is going to be created. These are the steps to build it:

1) Define a Transport Zone (global or universal), in this case, global.

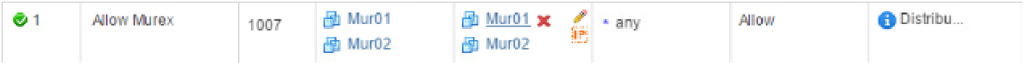
2) Create a Logical Switch, here it is the Murex one (in Figure 97 it is the pink network).

3) Associate the virtual switch to the TZ.

4) Assign VMs to the vSwitch. It has to be mentioned that it is the vNIC of a VM what is attached to the vSwitch.

5) Review that VMs IP addresses are in the same segment; for example: 172.16.60.12 and 172.16.60.22.

6) Apply firewall rules required for the communication of VMs. For example: allow traffic between vSwitch Murex or from VMs containing Mur in its virtual machine name.

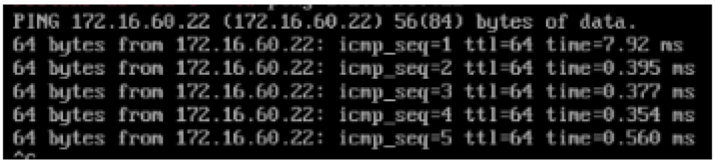


✓ 1	Allow Murex	1007	Mur01 Mur02	Mur01 ✗ Mur02	any	Allow	Distribu...

Figure 98, Firewall rule allowing traffic between Murex VMs.

7) Configure the default gateway and interfaces in distributed logical router.

8) Verify connectivity between VMs: Ping from Mur01 to Mur02.



```
PING 172.16.60.22 (172.16.60.22) 56(84) bytes of data.
64 bytes from 172.16.60.22: icmp_seq=1 ttl=64 time=7.92 ms
64 bytes from 172.16.60.22: icmp_seq=2 ttl=64 time=0.395 ms
64 bytes from 172.16.60.22: icmp_seq=3 ttl=64 time=0.377 ms
64 bytes from 172.16.60.22: icmp_seq=4 ttl=64 time=0.354 ms
64 bytes from 172.16.60.22: icmp_seq=5 ttl=64 time=0.560 ms
```

Figure 99. Connectivity between Murex VMs verification.

In the end, it is the creation of a new virtual switch, a new VXLAN.

4.1.2. Cross vCenter: Migration of a VM between vCenters.

The main functionality of VMware vSphere vMotion is that the migration of a virtual machine from a vCenter to another can be made without affecting service nor performance. In production environments, failovers are critical specially when applications have a great impact over the final user.

This case is intended to proof what VMware says, the non-loss of any ping during the migration of a VM between vCenters.

Web-03 is the VM candidate to be migrated. At the initial point it resides in DC1, after migration it will belong to DC2.

Situation before migration:

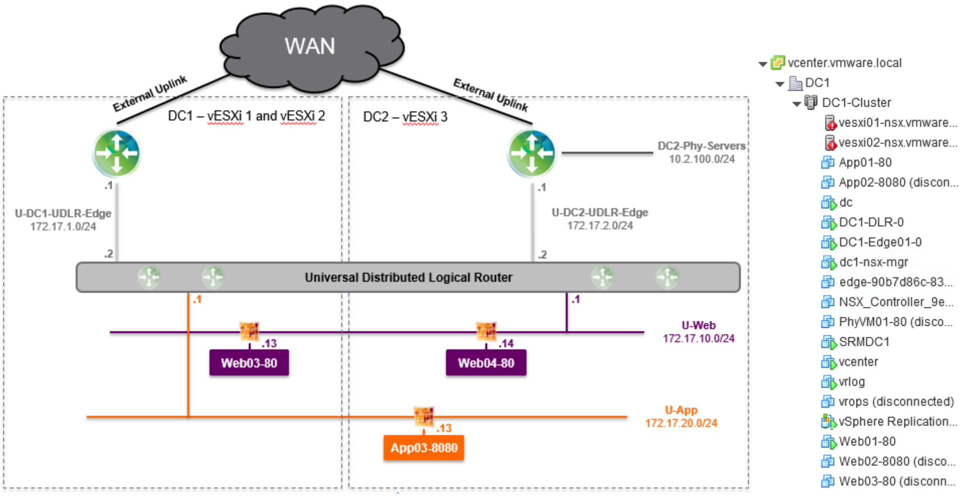


Figure 100. Web-03 vESXi ubication before being migrated.

After migrating Web-03 from DC1 to DC2 the desired situation is showed below:

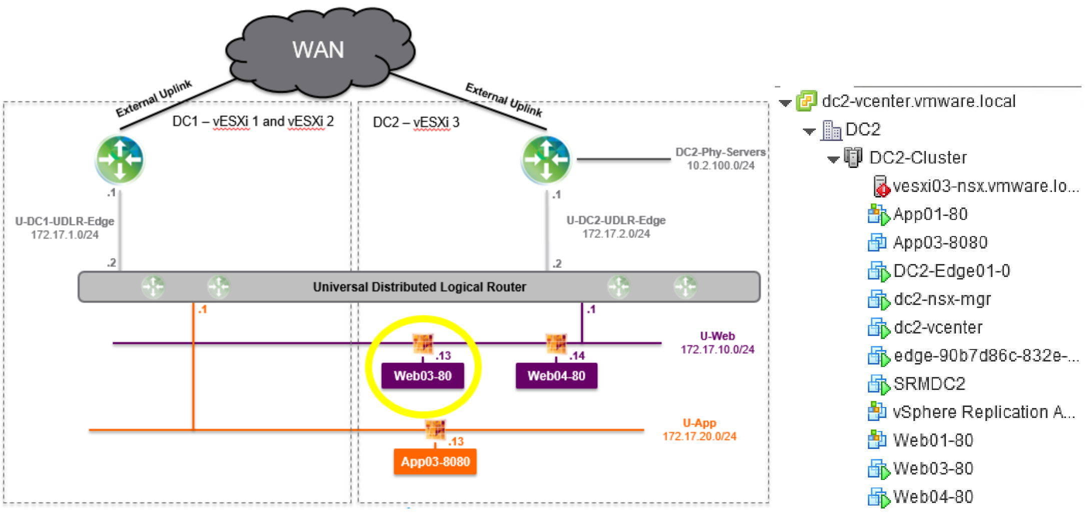


Figure 101. Web-03 ubication after being migrated.

The steps followed during the migration are described below:

1) Right click on Web-03 and select Migrate.

2) A pop-menu appears and some actions have to be taken, these actions are:

- Migration type selection: for the current case it is “Change both compute resources and storage”. This stands for migrating the virtual machine to a specific host and its storage to a specific datastore. There are other migration types, for example, “Change compute resource only” or “Change storage Only”. The reason for choosing change both compute resources and storage is to prove that the full migration works perfectly.

- Compute resource selection: it is the destination compute resource for the virtual machine. Web-03 before resided in vESXi-02; after migration it will be part of vESXi-03.

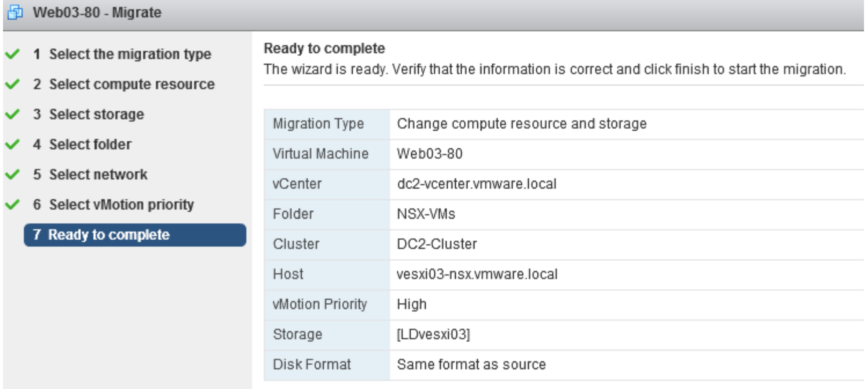
- Storage selection: it is the destination storage for the virtual machine. Due to migration, the selected datastore will be the dedicated one for vESXi-03, so LDvesxi03.

- Folder selection: it is the destination virtual machine folder for the virtual machine migration. For the actual case it is the NSX-VMs, a folder created by the administrator.

- Network selection: it is the destination network. As it is represented in Figure 101, the destination network is U-Web vSwitch.

- vMotion priority selection: defines the allocation of CPU resources to maintain performance of those VM that are in migration. In this case, the selected one is “Schedule vMotion with high priority”. This option is the one chosen because vMotion receives higher CPU scheduling preference relative to normal priority migrations; also, vMotion might complete quickly and then service would not be affected.

The summary of previous actions is:


Web03-80 - Migrate

- ✓ 1 Select the migration type
- ✓ 2 Select compute resource
- ✓ 3 Select storage
- ✓ 4 Select folder
- ✓ 5 Select network
- ✓ 6 Select vMotion priority
- 7 Ready to complete

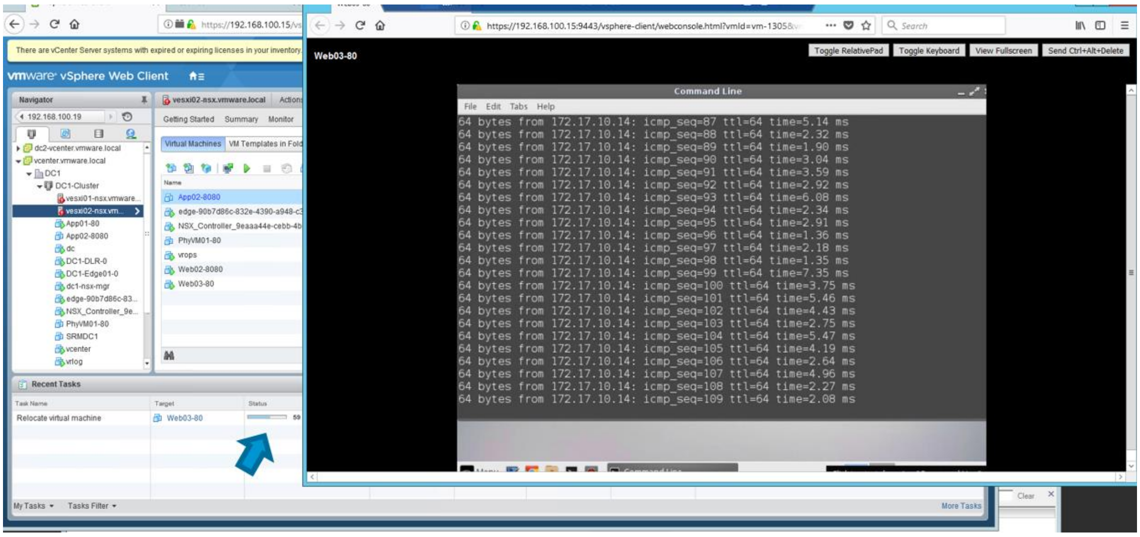
Ready to complete
The wizard is ready. Verify that the information is correct and click finish to start the migration.

Migration Type	Change compute resource and storage
Virtual Machine	Web03-80
vCenter	dc2-vcenter.vmware.local
Folder	NSX-VMs
Cluster	DC2-Cluster
Host	vesxi03-nsx.vmware.local
vMotion Priority	High
Storage	[LDvesxi03]
Disk Format	Same format as source

Figure 102. vMotion preferences.

Once actions have been taken, migration can start. Before that, a continuous ping from Web-03 to Web-04 is thrown.

The next figure shows the ping and the task bar showing the current process of virtual machine migration:



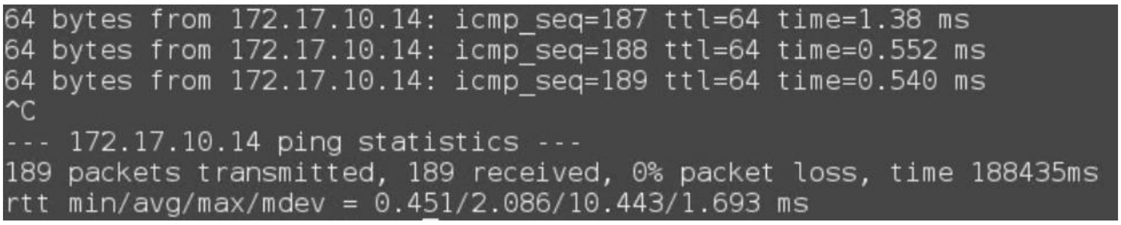
```
64 bytes from 172.17.10.14: icmp_seq=87 ttl=64 time=5.14 ms
64 bytes from 172.17.10.14: icmp_seq=88 ttl=64 time=2.32 ms
64 bytes from 172.17.10.14: icmp_seq=89 ttl=64 time=1.90 ms
64 bytes from 172.17.10.14: icmp_seq=90 ttl=64 time=3.04 ms
64 bytes from 172.17.10.14: icmp_seq=91 ttl=64 time=3.59 ms
64 bytes from 172.17.10.14: icmp_seq=92 ttl=64 time=2.92 ms
64 bytes from 172.17.10.14: icmp_seq=93 ttl=64 time=6.08 ms
64 bytes from 172.17.10.14: icmp_seq=94 ttl=64 time=2.34 ms
64 bytes from 172.17.10.14: icmp_seq=95 ttl=64 time=2.91 ms
64 bytes from 172.17.10.14: icmp_seq=96 ttl=64 time=1.36 ms
64 bytes from 172.17.10.14: icmp_seq=97 ttl=64 time=2.18 ms
64 bytes from 172.17.10.14: icmp_seq=98 ttl=64 time=1.35 ms
64 bytes from 172.17.10.14: icmp_seq=99 ttl=64 time=7.35 ms
64 bytes from 172.17.10.14: icmp_seq=100 ttl=64 time=3.75 ms
64 bytes from 172.17.10.14: icmp_seq=101 ttl=64 time=5.46 ms
64 bytes from 172.17.10.14: icmp_seq=102 ttl=64 time=4.43 ms
64 bytes from 172.17.10.14: icmp_seq=103 ttl=64 time=2.75 ms
64 bytes from 172.17.10.14: icmp_seq=104 ttl=64 time=5.47 ms
64 bytes from 172.17.10.14: icmp_seq=105 ttl=64 time=4.19 ms
64 bytes from 172.17.10.14: icmp_seq=106 ttl=64 time=2.64 ms
64 bytes from 172.17.10.14: icmp_seq=107 ttl=64 time=4.96 ms
64 bytes from 172.17.10.14: icmp_seq=108 ttl=64 time=2.27 ms
64 bytes from 172.17.10.14: icmp_seq=109 ttl=64 time=2.08 ms
```

Figure 103. Migration intermediate steps.

It is important to note that during migration, the pings duration is higher in comparison when there is no migration in process. Despite, there are no pings lost during the migration of a virtual machine from a vCenter to another one.

It should be especially considered that, as Web-03 console is opened from vCenter 1 and Web-03 is no longer in that vCenter, when migration is completed that console is closed. After that, what administration has to do is to open vCenter2 and from there, open a new Web-03 console. Once a new console is open, it can be appreciated that ping stills working.

When the ping is finished, by the user, these are the statistics:



```
64 bytes from 172.17.10.14: icmp_seq=187 ttl=64 time=1.38 ms
64 bytes from 172.17.10.14: icmp_seq=188 ttl=64 time=0.552 ms
64 bytes from 172.17.10.14: icmp_seq=189 ttl=64 time=0.540 ms
^C
--- 172.17.10.14 ping statistics ---
189 packets transmitted, 189 received, 0% packet loss, time 188435ms
rtt min/avg/max/mdev = 0.451/2.086/10.443/1.693 ms
```

Figure 104. Ping statistics during the migration.

The previous figure proves that no pings were lost, so connections, services are kept during a migration of a VM from an vCenter to another.

4.1.3. Load Balancer Deployment

A load balancer is a device that distributes network or application traffic across several servers. Load balancers are used to increase capacity, concurrent users and reliability of applications besides an improvement of the overall performance. In addition, a balancer contributes reducing response times and ensures service redundancy.

Load balancers ensure reliability and availability by only sending requests to servers and applications that can respond in a timely manner.

Load balancing is one of the NSX Edge functionalities. This use case shows how a load balancer is build and configured.

The ubication of NSX Load Balancer will be the following one:

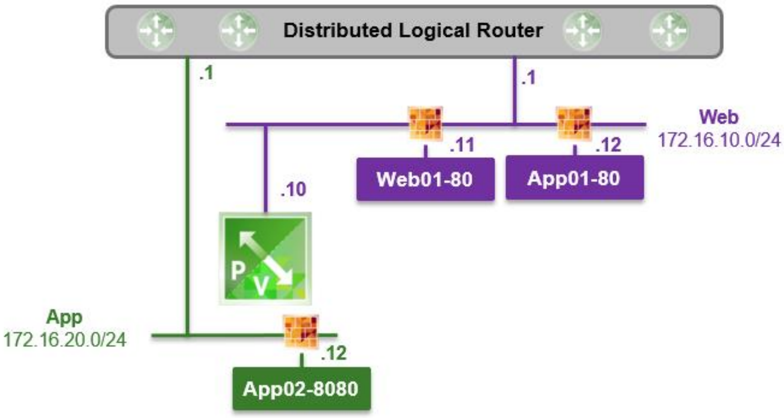


Figure 105. Load balancer scenario.

CONSIDERATIONS:

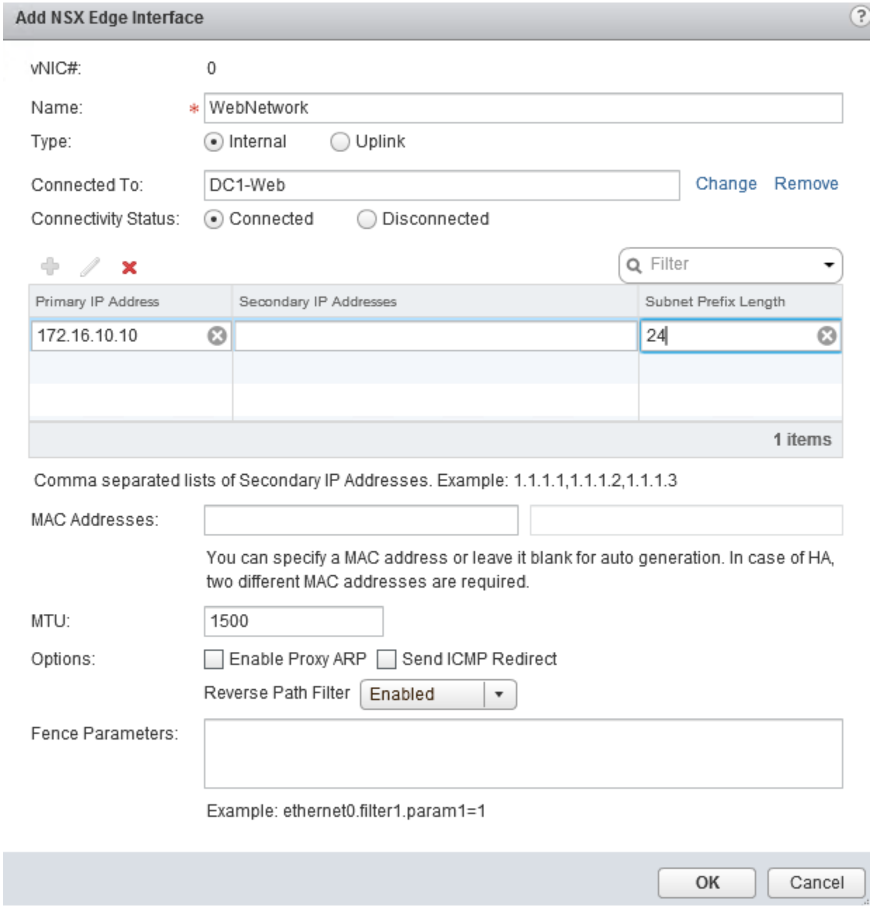
- It should be especially considered that the balancing will be done between Web-01 and App-01. The reason for this is that both virtual machines are accessible from port 80. In production environments, it is common to perform balancing between same kind of machines, both web or app. In this case, it is not possible because of the issue of ports that has mentioned earlier.
- App-01 IP address has been changed from 172.16.20.11 to 172.16.10.11.

- App-01 is temporarily attached to vSwitch DC1-Web

In the following are the steps how to build it:

1) Create a new NSX Edge. It can be created going to Networking & Security NSX Edges and add. There, the installation type is set to “Edge Services Gateway” and the name is “OneArm-Balancer”. It is important to enable SSH, this would let the administrator to control the balancer via CLI.

2) Configure Load Balancer interfaces. It is important to define which interface is going to connect the load balancer and the vSwitch.



Add NSX Edge Interface

vNIC#: 0

Name: * WebNetwork

Type: ● Internal ○ Uplink

Connected To: DC1-Web

Connectivity Status: ● Connected ○ Disconnected

Primary IP Address	Secondary IP Addresses	Subnet Prefix Length
172.16.10.10		24

1 items

Comma separated lists of Secondary IP Addresses. Example: 1.1.1.1,1.1.1.2,1.1.1.3

MAC Addresses:

You can specify a MAC address or leave it blank for auto generation. In case of HA, two different MAC addresses are required.

MTU: 1500

Options: □ Enable Proxy ARP □ Send ICMP Redirect

Reverse Path Filter: Enabled

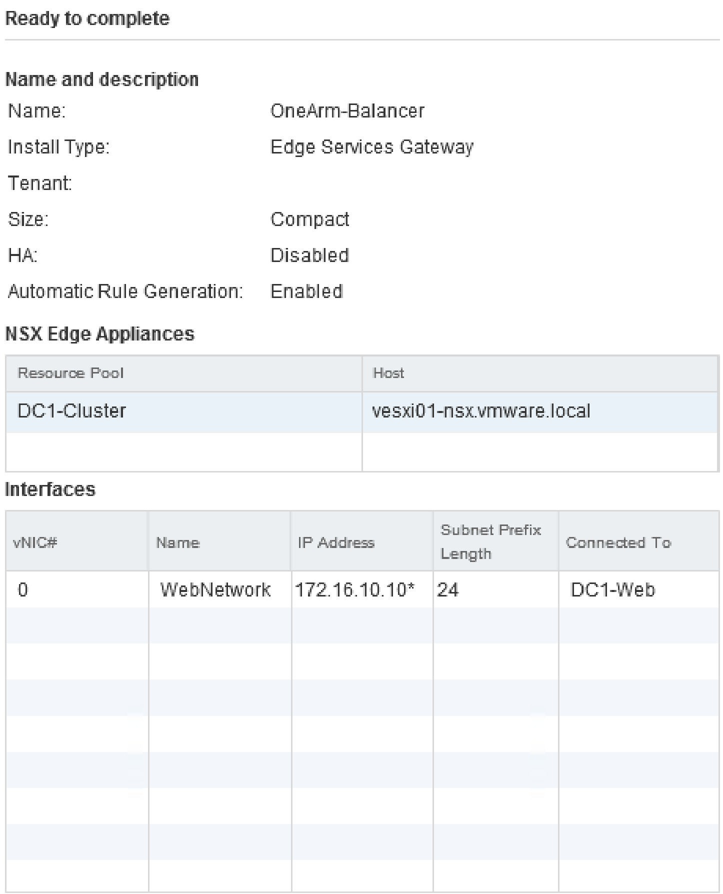
Fence Parameters:

Example: ethernet0.filter1.param1=1

Figure 106. Load Balancer interface configuration.

It can be appreciated that load balancer is connected to DC1-Web, which is the virtual switch that has both VMs and the balancer attached. The primary IP Address corresponds to the one that will receive the requests.

This is the Load Balancer recommended configuration:



Ready to complete

Name and description

Name:	OneArm-Balancer
Install Type:	Edge Services Gateway
Tenant:	
Size:	Compact
HA:	Disabled
Automatic Rule Generation:	Enabled

NSX Edge Appliances

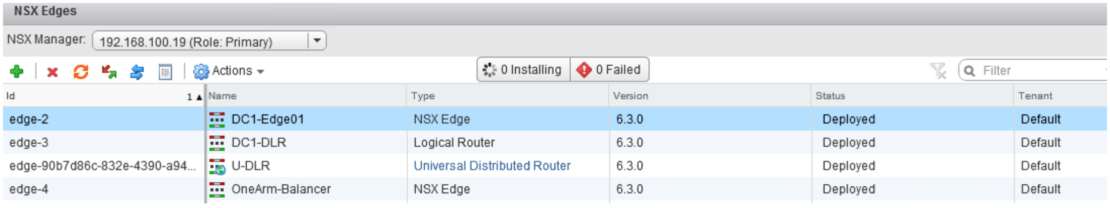
Resource Pool	Host
DC1-Cluster	vesxi01-nsx.vmware.local

Interfaces

vNIC#	Name	IP Address	Subnet Prefix Length	Connected To
0	WebNetwork	172.16.10.10*	24	DC1-Web

Figure 107. Load Balancer deployment parameters.

One the balancer is deployed (it takes a few minutes), in the NSX Edger section appears a new instance corresponding to OneArm-Balancer.



NSX Edges

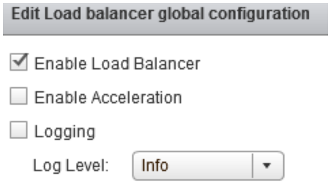
NSX Manager: 192.168.100.19 (Role: Primary)

Id	Name	Type	Version	Status	Tenant
edge-2	DC1-Edge01	NSX Edge	6.3.0	Deployed	Default
edge-3	DC1-DLR	Logical Router	6.3.0	Deployed	Default
edge-90b7d86c-832e-4390-a94...	U-DLR	Universal Distributed Router	6.3.0	Deployed	Default
edge-4	OneArm-Balancer	NSX Edge	6.3.0	Deployed	Default

Figure 108. NSX Edges deployed in principal scenario.

Apart from the initial configuration, balancing services have to be defined.

3) Enable Load balancer services. They are enabled through Networking & Security → NSX Edges → Select EDGE → Manage → Load Balancer → Global Configuration → Edit… and activate the checkbox corresponding to Enable Load Balancer.



Edit Load balancer global configuration

☑ Enable Load Balancer

□ Enable Acceleration

□ Logging

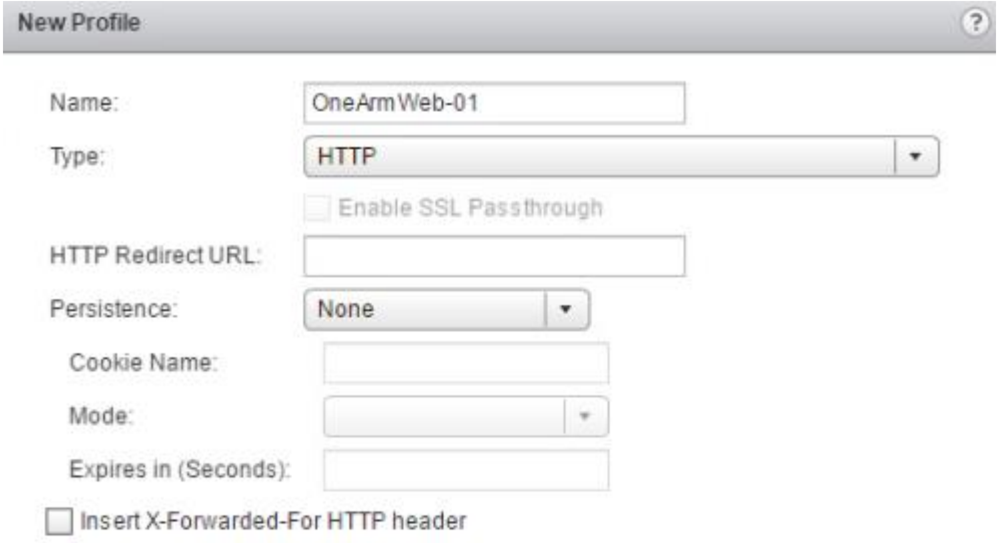
Log Level: Info

Figure 109. Enabling Load Balancer.

4) Set Load Balancer parameters. There are many parameters and steps to be done before a balancer can be used:

- Define an Application Profile.
- Configure service monitor.
- Define pools.
- Configure a virtual server.

5) Define an Application Profile. Application Profile are in charge of the behaviour of a typical type of network traffic. These profiles are applied to a virtual server (VIP) which handles traffic based on the values specified in the Application Profile. Profiles can make traffic-management tasks less error-prone and more efficient.



New Profile

Name: OneArmWeb-01

Type: HTTP

□ Enable SSL Passthrough

HTTP Redirect URL:

Persistence: None

Cookie Name:

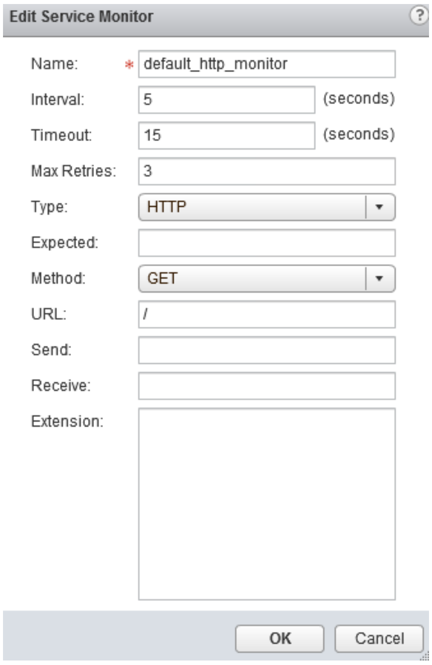
Mode:

Expires in (Seconds):

□ Insert X-Forwarded-For HTTP header

Figure 110. Load Balancer Profile definition.

6) Configure service monitor. Monitors ensure that pool members serving virtual servers are up and working. For instance, the default HTTPS monitor will simply do a "GET" at "/". We will modify the default monitor to do a health check at application specific URL. This will help determine that not only the pool member servers are up and running but the application is as well.

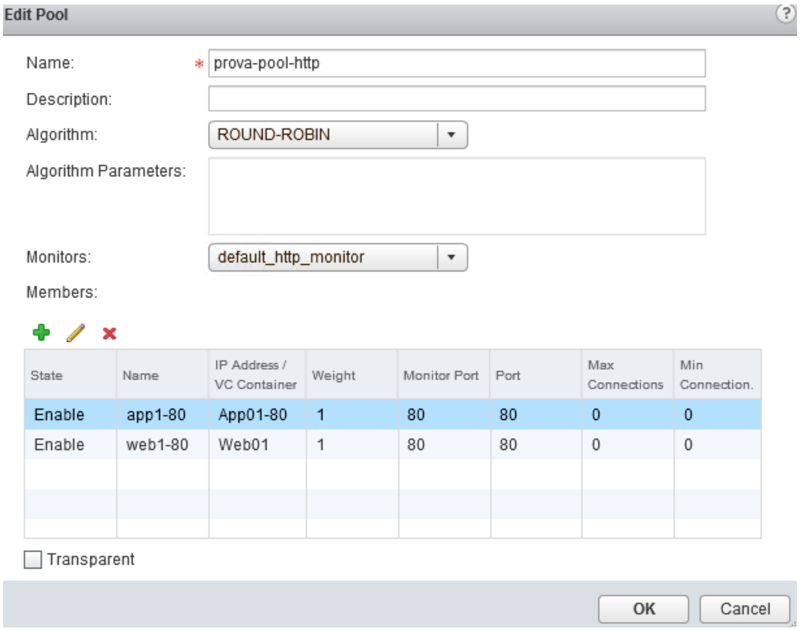


Edit Service Monitor

Name: *	default_http_monitor
Interval:	5 (seconds)
Timeout:	15 (seconds)
Max Retries:	3
Type:	HTTP
Expected:	
Method:	GET
URL:	/
Send:	
Receive:	
Extension:	

Figure 111. Load Balancer Service Monitor.

7) Define Pools. A Pool is the entity that represents the nodes that traffic is getting load balanced to. Web-01 and App-01 are the virtual machines added to the new pool.



Edit Pool

Name: *	prova-pool-http
Description:	
Algorithm:	ROUND-ROBIN
Algorithm Parameters:	
Monitors:	default_http_monitor

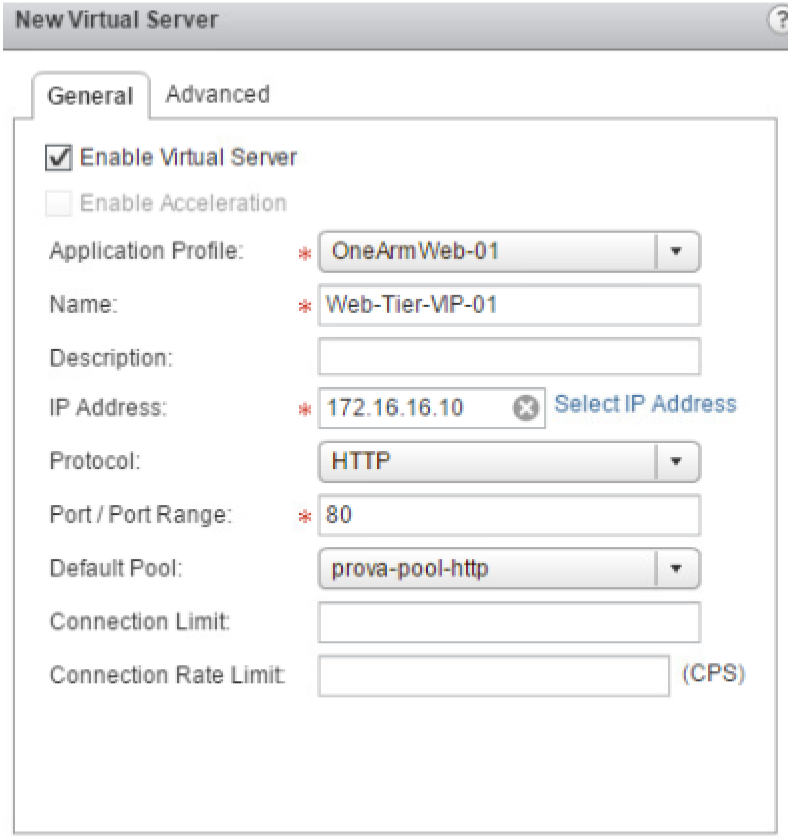
Members:

State	Name	IP Address / VC Container	Weight	Monitor Port	Port	Max Connections	Min Connection.
Enable	app1-80	App01-80	1	80	80	0	0
Enable	web1-80	Web01	1	80	80	0	0

☐ Transparent

Figure 112. Load Balancer Pool configuration.

8) Configure a Virtual Server. This virtual server is the one that users will request. It does not matter if the port is the same as the VM that would be behind it.



New Virtual Server

☑ Enable Virtual Server

□ Enable Acceleration

Application Profile: *	OneArmWeb-01
Name: *	Web-Tier-VIP-01
Description:	
IP Address: *	172.16.16.10
Protocol:	HTTP
Port / Port Range: *	80
Default Pool:	prova-pool-http
Connection Limit:	
Connection Rate Limit:	(CPS)

Figure 113. Load Balancer Virtual Server configuration.

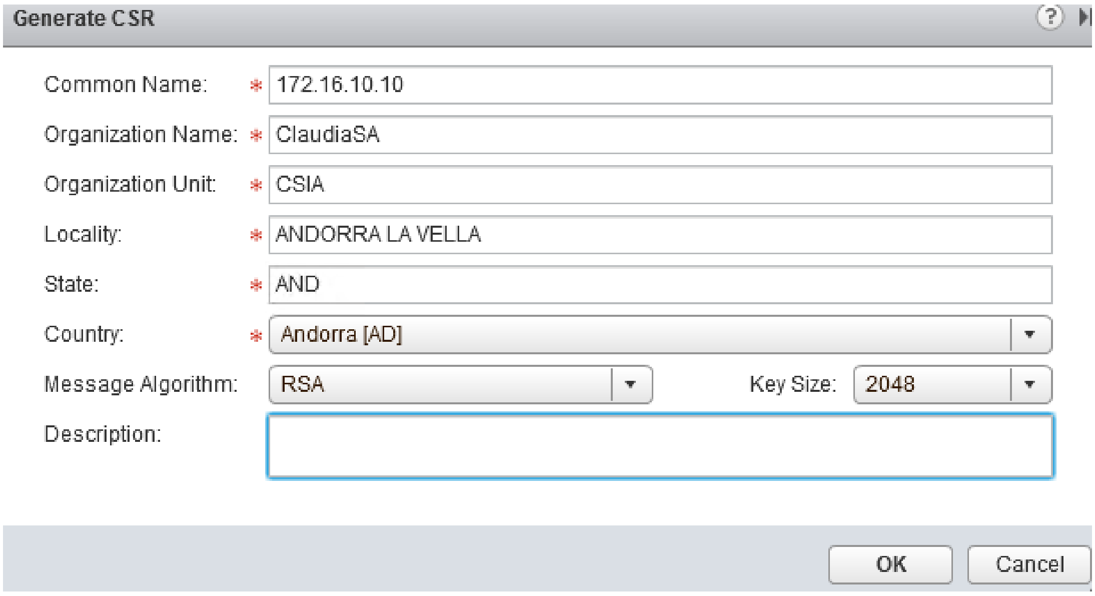
From the configuration it can be deduced that the VS that allocate users requests is Web-Tier-VIP-01 with an IP address of 172.16.10.10. Likewise, it is a virtual server that only accepts HTTP requests and, behind, there are hosted virtual machines defined in the previous step.

This step concludes the deployment of the One-Arm Load balancer. In the following are the verifications to check that balancer is working properly; moreover, there is included a brief explanation of how to create self-signed certificates, if necessary.

9) Self-Signed Certificate (if necessary)

A Virtual Server certificate is required when virtual servers are accessed via HTTPS. It is created in Networking & Security → NSX Edges → Select EDGE → Manage → Settings -→ Certificates → Actions → Generate CSR.

This is the CSR certificate:



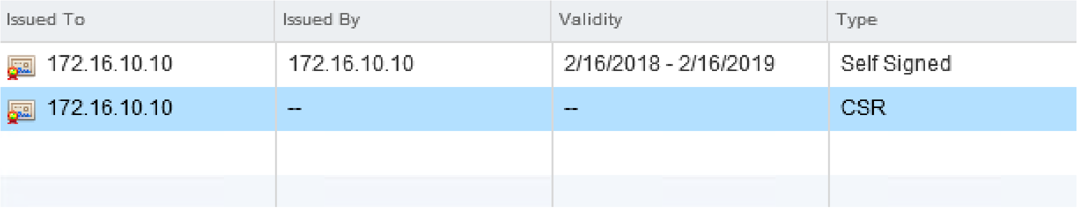
Generate CSR

Common Name: *	172.16.10.10
Organization Name: *	ClaudiaSA
Organization Unit: *	CSIA
Locality: *	ANDORRA LA VELLA
State: *	AND
Country: *	Andorra [AD]
Message Algorithm:	RSA
Key Size:	2048
Description:	

Figure 114. CSR Certificate.

As the name indicates, the previous certificate is a CSR one. Due to the objective is a self-signed certificate, to obtain a self-signed one, the administrator has to edit the CSR introducing a time period that specifies the validity of that certificate.

These are the certificates created:



Issued To	Issued By	Validity	Type
172.16.10.10	172.16.10.10	2/16/2018 - 2/16/2019	Self Signed
172.16.10.10	--	--	CSR

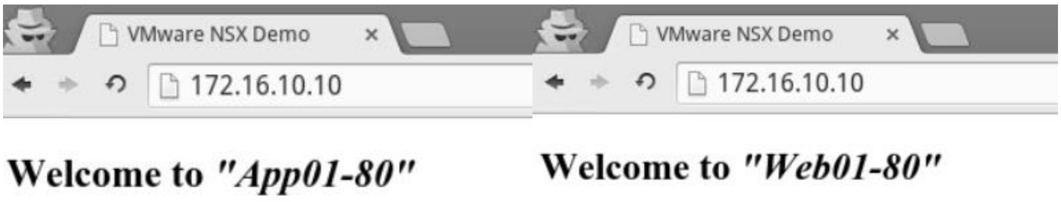
Figure 115. Load Balancer Certificates.

After certificates are generated, they should be uploaded in balancer applications profiles.

10) Verification of balancing services.

Requests to load balances IP are done from App-02. Before to try it, a firewall rule enabling traffic from App-02 to the balancer should be enabled.

After that, from App-02 console, request can be done using a browser. The request is: http://172.16.10.10. Making consecutive requests it is appreciated that Web-01 and App-01 are alternating.



Welcome to "App01-80"

Welcome to "Web01-80"

Figure 116. Balanced Requests.

It should be mentioned that for the developed case, the Virtual Server does not require a certificate. Nevertheless, if the virtual server is HTTPS, a certificate has to be imported or generated, even if it is self-signed.

With last entry, the Load Balancer chapter ends.

4.1.4. Native Bridging

NSX provides L2 Bridging capabilities that allow organizations to seamlessly connect traditional workloads and legacy VLANs to virtualized networks using VXLAN. L2 Bridging simplifies the introduction of logical networks and other scenarios involving physical systems that require L2 connectivity to virtual machines.

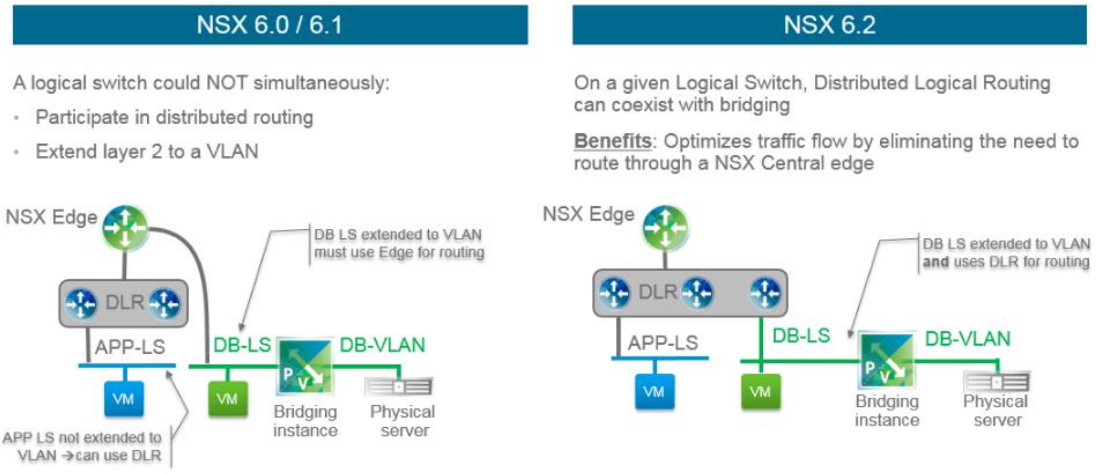
Logical routers can provide L2 bridging from the logical networking to the physical VLAN-backed network. This fact, allows the creation of an L2 bridge between a logical switch and a VLAN, which enables the migration of virtual workloads to physical devices without causing any impact on IP addresses. As a result, a logical network can leverage a physical L3 gateway and access existing physical networks and security resources by bridging the logical switch broadcast domain to the VLAN broadcast domain. This is a new functionality introduced in NSX version 6.2 which was not permitted in previous versions of NSX. Moreover, this function has been enhanced because bridged Logical Switches can be connected to Distributed Logical Routers.

The main objective of this case is to configure a L2 Bridging instance from a traditional VLAN to a NSX Logical Switch. An advantage of L2 bridging is that optimizes East-West traffic. Moreover, it reduces the trombone effect, that collapses Edges with traffic that, in case of being everything virtual, the traffic would not go through there.

In order to put the audience in context, before starting any configuration, it has to be explained how physical machines use to access the virtual network and how is it done after NSX 6.2 release.

Next figure is extracted from VMware directly:

Previous NSX versions (6.0/6.1) NSX 6.2 version



NSX 6.0 / 6.1

A logical switch could NOT simultaneously:

- Participate in distributed routing
- Extend layer 2 to a VLAN

NSX 6.2

On a given Logical Switch, Distributed Logical Routing can coexist with bridging

Benefits: Optimizes traffic flow by eliminating the need to route through a NSX Central edge

NSX Edge
DLR
APP-LS
DB-LS
DB-VLAN
VM
VM
Bridging instance
Physical server
DB LS extended to VLAN must use Edge for routing
APP LS not extended to VLAN →can use DLR
NSX Edge
DLR
APP-LS
DB-LS
DB-VLAN
VM
VM
Bridging instance
Physical server
DB LS extended to VLAN and uses DLR for routing

Figure 117. Comparison between L2 bridging in current and older NSX versions.

For this use case Web-01 it is not a virtual machine. It has to be supposed that it is a physical workstation. DPortGroup1 is the portgroup intended when dealing with physical environments.

This use case has a clear objective: connect Web-01 to a DLR instead of a NSX Edge creating a L2 bridge.

In lower NSX Versions (6.0/6.1) the scenario was the following one:

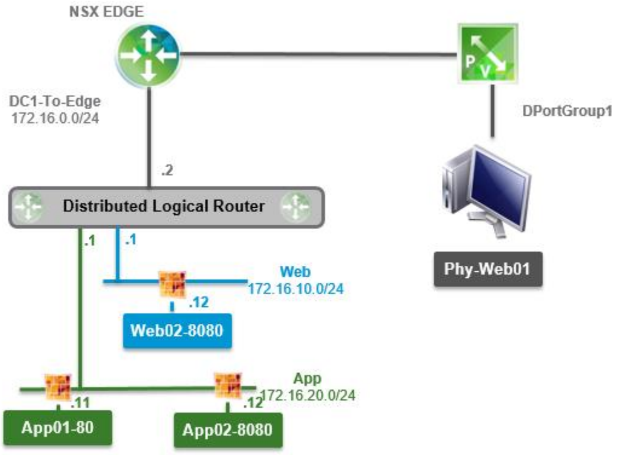


Figure 118. L2 bridging in old NSX Versions.

On the contrary, in NSX Version 6.2 it is not necessary to connect the physical machine to an Edge, it can be directly connected to a DLR as in next figure:

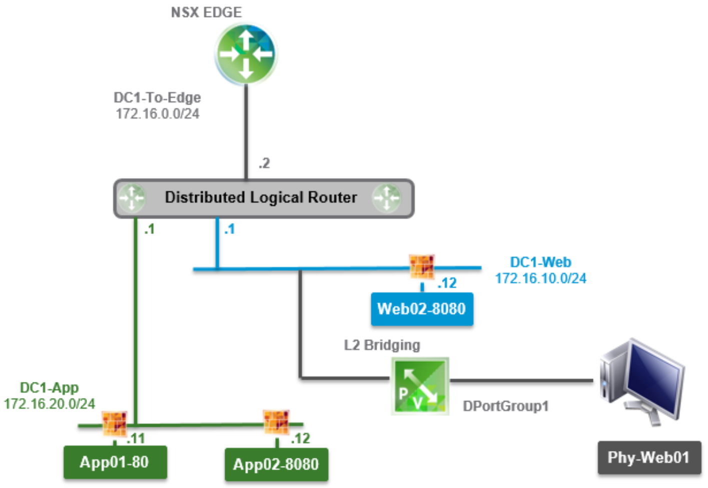
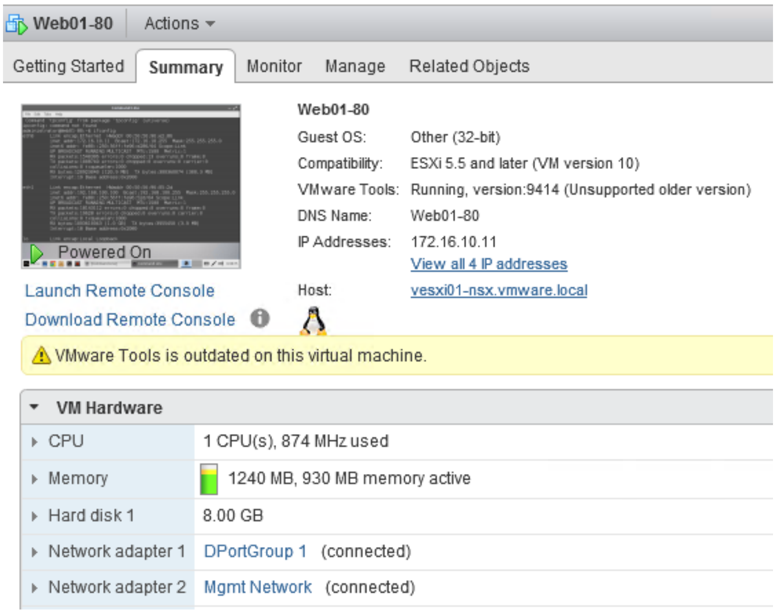


Figure 119. Improved L2 bridging in NSX 6.2.

Actions to be taken to deploy L2 bridging:

1) Create a new PortGroup from Phy-Web01 to L2 Bridge. (DPortGroup1)

2) Relate Phy-Web01 with DPortGroup1 and verify it. Next figure ensures that Web-01 as network adapter 1 has DPortGroup1.



Web01-80

Guest OS:	Other (32-bit)
Compatibility:	ESXi 5.5 and later (VM version 10)
VMware Tools:	Running, version:9414 (Unsupported older version)
DNS Name:	Web01-80
IP Addresses:	172.16.10.11 View all 4 IP addresses
Host:	vesxi01-nsx.vmware.local

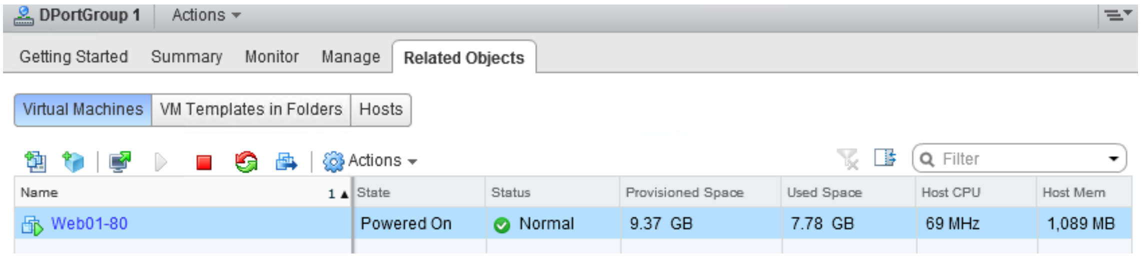
⚠ VMware Tools is outdated on this virtual machine.

VM Hardware

CPU	1 CPU(s), 874 MHz used
Memory	1240 MB, 930 MB memory active
Hard disk 1	8.00 GB
Network adapter 1	DPortGroup 1 (connected)
Network adapter 2	Mgmt Network (connected)

Figure 120. Web-01 Network adapter 1 configured as DPortGroup1.

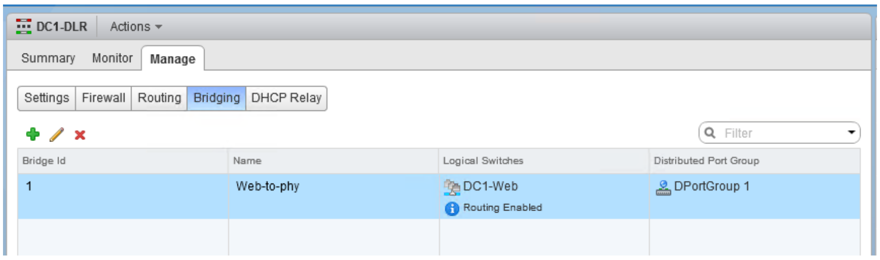
3) Verify that DPortGroup contains Web-01:



Name	State	Status	Provisioned Space	Used Space	Host CPU	Host Mem
Web01-80	Powered On	Normal	9.37 GB	7.78 GB	69 MHz	1,089 MB

Figure 121. Verifying that Web-01 is attached to DPortGroup1.

4) Configure L2 bridging in the Distributed Logical Router. In NSX Edges, double click DC1-DLR. When DC1-DLR options appear, navigate to Manage Bridging. A new bridge has to be created. To create it, the administrator has to specify which port group is the one that contains the Physical machine. Moreover, it is also specified to which Logical Switch is the L2 Bridge attached, in this case DC1-Web.

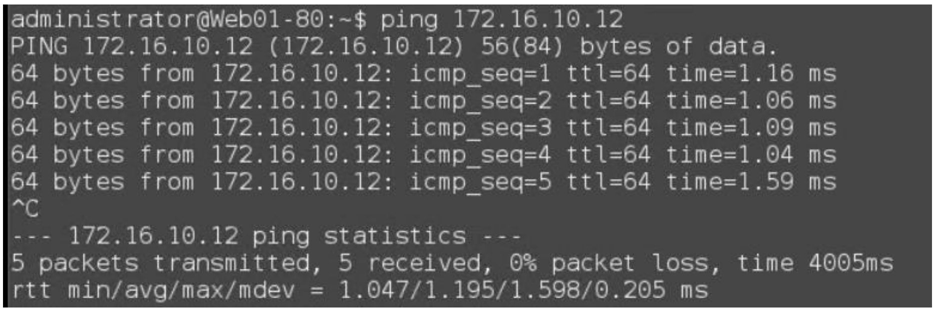


Bridge Id	Name	Logical Switches	Distributed Port Group
1	Web-to-phy	DC1-Web Routing Enabled	DPortGroup 1

Figure 122. DLR bridging configuration.

5) Verifications: it has to be verified that connectivity between Web-01 and Web-02, and Web-01 and App-01 is done without problems.

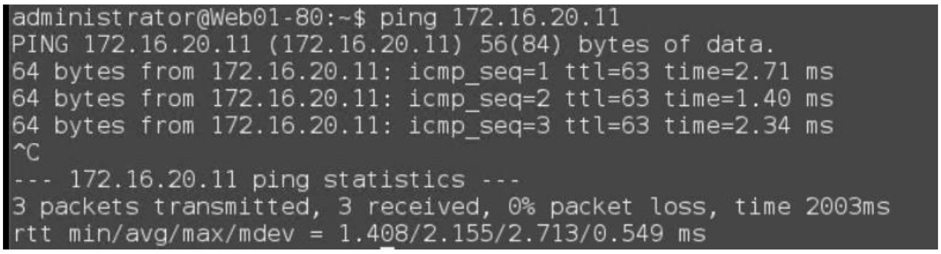
Ping from Web-01 to Web-02



```
administrator@Web01-80:~$ ping 172.16.10.12
PING 172.16.10.12 (172.16.10.12) 56(84) bytes of data.
64 bytes from 172.16.10.12: icmp_seq=1 ttl=64 time=1.16 ms
64 bytes from 172.16.10.12: icmp_seq=2 ttl=64 time=1.06 ms
64 bytes from 172.16.10.12: icmp_seq=3 ttl=64 time=1.09 ms
64 bytes from 172.16.10.12: icmp_seq=4 ttl=64 time=1.04 ms
64 bytes from 172.16.10.12: icmp_seq=5 ttl=64 time=1.59 ms
^C
--- 172.16.10.12 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.047/1.195/1.598/0.205 ms
```

Figure 123. L2 Bridging: successful ping between Web-01 and Web-02.

Ping from Web-01 to App-01



```
administrator@Web01-80:~$ ping 172.16.20.11
PING 172.16.20.11 (172.16.20.11) 56(84) bytes of data.
64 bytes from 172.16.20.11: icmp_seq=1 ttl=63 time=2.71 ms
64 bytes from 172.16.20.11: icmp_seq=2 ttl=63 time=1.40 ms
64 bytes from 172.16.20.11: icmp_seq=3 ttl=63 time=2.34 ms
^C
--- 172.16.20.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.408/2.155/2.713/0.549 ms
```

Figure 124. L2 Bridging: successful ping between Web-01 and App-01.

Both pings are successful.

To emphasize the importance of new L2 Bridging functionality, the following figures are a comparison of before and after where traffic passes from a physical machine to a virtual one.

Traffic flows from Web-01 to Web-2

Before After

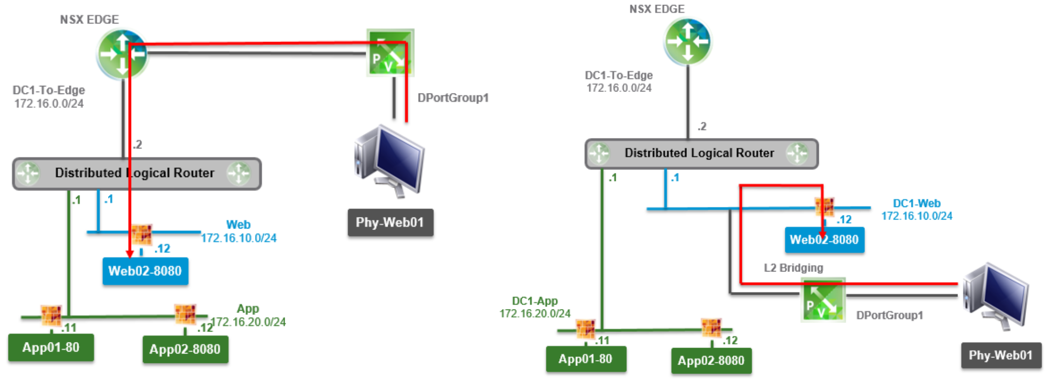


Figure 125. L2 Bridging: Comparison of traffic paths due to improvement of L2 bridging in NSX.

In conclusion, in the new NSX version, the Edge traffic is clearly lower compared to older versions. This supposes an optimization of resources besides improving the efficiency of the network

This use case concludes networking use case section.

4.2. Security Use Cases

Security use cases consist on the management of security policies and some practices adopted to prevent and monitor unauthorized access, illicit network flows, and misuse the network-accessible resources.

The networking use cases are the following ones:

- Microsegmentation:

    - Default rule to isolate traffic flows.
    - Single Security Group.
    - Multiple Security Groups with excluded members.
    - Collapsed tier.
- Service composer.
- Visibility and Operations:
    - Log insight
    - Flow monitoring
    - Trace Flow
    - Endpoint monitoring.

4.2.1. Microsegmentation

Microsegmentation is a method that consists on creating secure zones in data centers and cloud deployments that allows companies to isolate workloads and, secure them individually. The objective is making network security more granular. Microsegmentation is achieved by creating VLANs (use case 10.1.1. Creation of a virtual network) and designing and applying the appropriate firewall rules.

The distribute firewall has an important role when dealing with microsegmentation because every packet sent from a virtual machine can be inspected before the packet even touches the network. Security is omnipresent, at the first and last hop.

As discussed earlier in this report, firewall rules are based on vCenter objects, so security policies are decoupled from IP addressing. For this reason, it becomes possible to consider more simplified, flatter application architectures. For example, Web and App “tiers” could be on the same network segment, separated by their container membership.

There many ways to implement microsegmentation. Some of them will be shown.

Microsegmentation using a default rule to isolate traffic flows

In traditional firewalls, virtual or non-virtual machines remaining to the same VLAN or network segment, can communicate each other by default, without having to apply any allow firewall rule.

In NSX, although VMs are attached to the same vSwitch, due to firewall is deployed at vNIC level, 2 VMs in a single network can be isolated by a firewall rule.

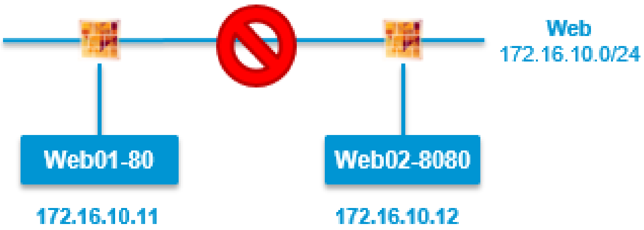
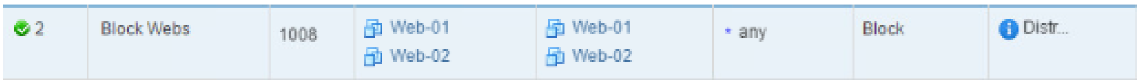


Figure 126. Isolating two VMs remaining to a single network using a firewall rule.

The scenario consists of two virtual machines: Web01 and Web02, and the objective is using a firewall rule to block the traffic between them.

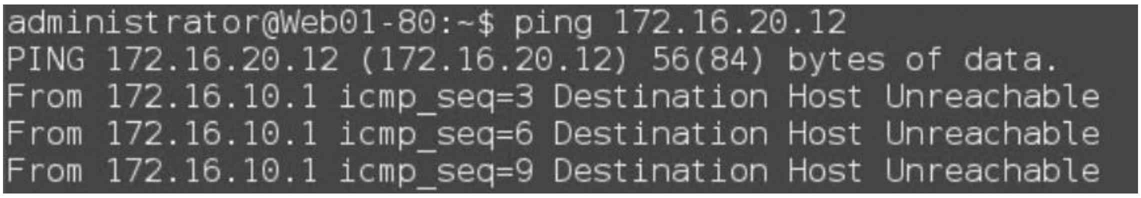
The firewall rule is:



✓ 2	Block Webs	1008	Web-01 Web-02	Web-01 Web-02	* any	Block	Distr...

Figure 127. Bidirectional firewall rule blocking traffic between Web01 and Web02.

The ping from Web-01 to Web-02 does not work although they are in same subnet and vSwitch:



```
administrator@Web01-80:~$ ping 172.16.20.12
PING 172.16.20.12 (172.16.20.12) 56(84) bytes of data.
From 172.16.10.1 icmp_seq=3 Destination Host Unreachable
From 172.16.10.1 icmp_seq=6 Destination Host Unreachable
From 172.16.10.1 icmp_seq=9 Destination Host Unreachable
```

Figure 128. Ping between Web01 and Web-02 without reaching destination.

It is important to notice that Web-01's gateway is who answers the ping request. Ping reply is "Destination Host Unreachable", it means that the traffic is blocked.

Microsegmentation using an specific firewall rule is the easiest option but the less practical one. If architecture changes, to maintain the same initial condition (block traffic flows between two VMs), for any network change an instance of the firewall rule has to be modified. So, although this option is valid, there are many other ones that are much better. For instance, creating Security groups, composed by VMs and associating them to firewall instances. These options are shown and developed in other use cases.

Microsegmentation using a single security group

Already commented in the project, security rules not necessarily have to be defined only by IP addresses, they can be created using clusters, datacentre, port groups… and even security groups.

A Security group is a collection of assets or grouping objects from the vSphere inventory. Its aim is to have a granular traffic control and demonstrate how to change security policies without changing network configuration.

In this use case, there is a single Security Group that contain VMs from same subnet; the security Group is called SG-Web-Servers and it contains Web01 and Web-02. The main objective is to permit traffic only between Security Group members. Other traffic flows are blocked: for example, from App01 to Web Servers. Using a security group, the flow management it is easier than using individual VMs. The administrator does not have to review the firewall rule by rule, and VM by VM. With Security Groups the firewall is well organized and its performance is better.

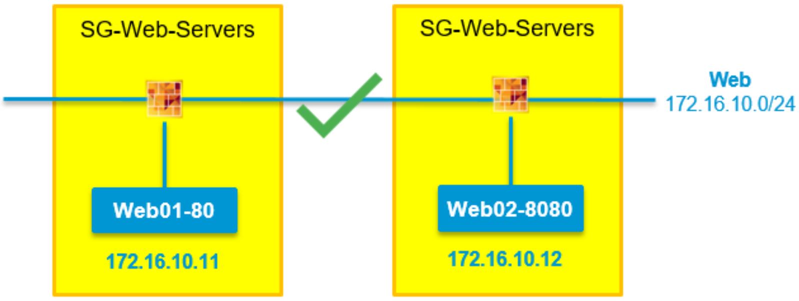
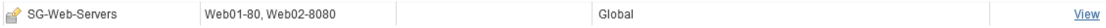


Figure 129. Microsegmentation allowing traffic between Security Group Members.

The first step is to create a Security Group. It is configurated in Networking&Security → NSX Manager → Select primary role NSX Manager (192.168.100.19) → Manage tab → Grouping Objects → Security Group. It is used the primary NSX Manager because VM used correspond to vESXi1 and vESXi2.

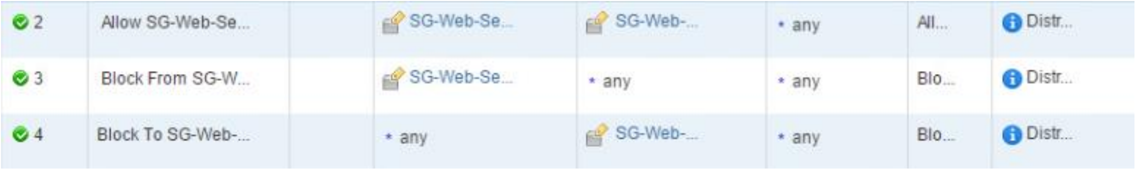
The security group created is:



SG-Web-Servers	Web01-80, Web02-8080		Global		View

Figure 130. Web servers Security Group.

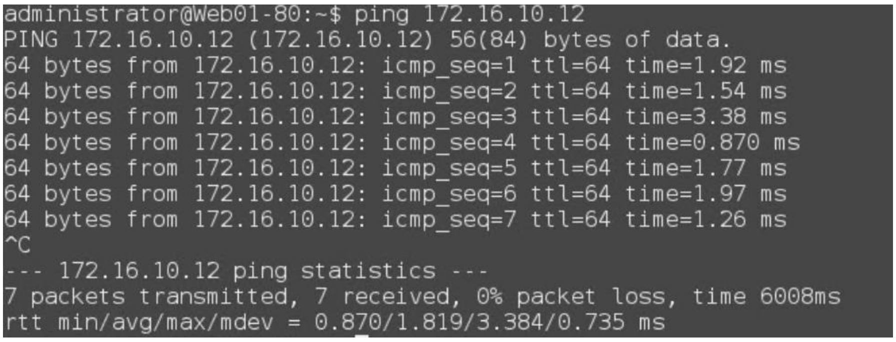
The second step is to create FW rules. The only rule necessary is the one that allows traffic between members of a security group. Due to issues regarding a deny default rule, two extra rules are added to deny traffic from web servers and to web servers.



✓ 2	Allow SG-Web-Se...		SG-Web-Se...	SG-Web-...	any	All...	Distr...
✓ 3	Block From SG-W...		SG-Web-Se...	any	any	Blo...	Distr...
✓ 4	Block To SG-Web-...		any	SG-Web-...	any	Blo...	Distr...

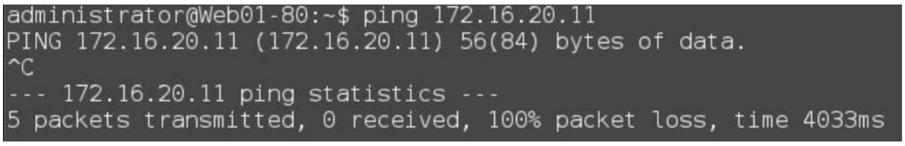
Figure 131. Firewall rules to only allow traffic between SG-Web-Servers.

To prove that, it is necessary to try a ping from Web-01 to Web-02 and verify that result is successful and the other test is to perform a ping from Web-01 to App-01 and verify that host is unreachable.



```
administrator@Web01-80:~$ ping 172.16.10.12
PING 172.16.10.12 (172.16.10.12) 56(84) bytes of data.
64 bytes from 172.16.10.12: icmp_seq=1 ttl=64 time=1.92 ms
64 bytes from 172.16.10.12: icmp_seq=2 ttl=64 time=1.54 ms
64 bytes from 172.16.10.12: icmp_seq=3 ttl=64 time=3.38 ms
64 bytes from 172.16.10.12: icmp_seq=4 ttl=64 time=0.870 ms
64 bytes from 172.16.10.12: icmp_seq=5 ttl=64 time=1.77 ms
64 bytes from 172.16.10.12: icmp_seq=6 ttl=64 time=1.97 ms
64 bytes from 172.16.10.12: icmp_seq=7 ttl=64 time=1.26 ms
^C
--- 172.16.10.12 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6008ms
rtt min/avg/max/mdev = 0.870/1.819/3.384/0.735 ms
```

Figure 132. Successful between SG-Web-Servers virtual machines.



```
administrator@Web01-80:~$ ping 172.16.20.11
PING 172.16.20.11 (172.16.20.11) 56(84) bytes of data.
^C
--- 172.16.20.11 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4033ms
```

Figure 133. Unsuccessful between SG-Web-Servers and virtual machines not included in that SG.

This is the simplest use case regarding the use of security groups. Hereafter, there are variations of this use case.

Microsegmentation using multiple Security Groups with excluded members

In the previous use case, the security group used only contained members from a same network segment with all members allowed.

This use case stands for two security groups where the elements included are from different subnets. In addition, one of the SG's has an excluded member.

The main advantage of using exclude members is, for example, that if a vSwitch has 10 VMs and all the flows allowed have to be the same excepting for a single VM, it is not necessary to create particular rules for the 9 members that have flows in common. The solution is, to assign all vSwitch to a security group and exclude as many virtual machines that do not share same behaviour.

Another case for using exclude members is, for example, the one represented here: Web-02 and App-01 belong to a same SG for designing purposes, but suddenly, it is found a vulnerability in App-02. Instead of removing App-02 from specific firewall rules, the vulnerable machine is removed from included members to excluded ones; when vulnerability is parched, with a single click, App-02 can be part of the security group as an included member again.

So, the scenario is the following one: 2 Security Groups, one of them with one excluding member, and the only traffic allowed between them is icmp, ssh and http.

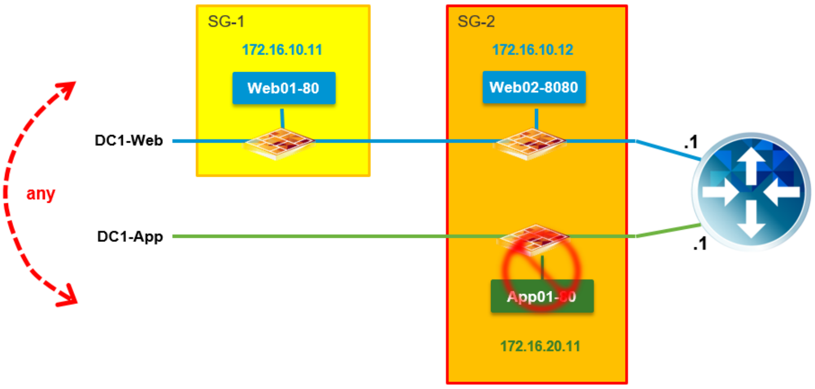
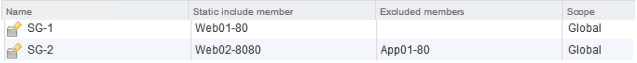


Figure 134. Microsegmentation allowing traffic between security groups with one excluded member.

Security groups created for this use case are two:

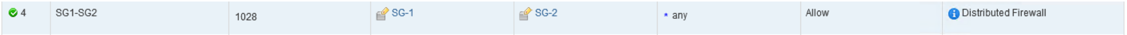


Name	Static include member	Excluded members	Scope
SG-1	Web01-80		Global
SG-2	Web02-8080	App01-80	Global

Figure 135. Security Groups.

It can be appreciated that App-01 figures as an excluded member due to it has a vulnerability and it could be infected. The scope of Security Groups is global due to virtual machines remain in same DC.

There is only one firewall rule that allows any kind of traffic between SG-1 and SG-2, other traffic flows are denied.

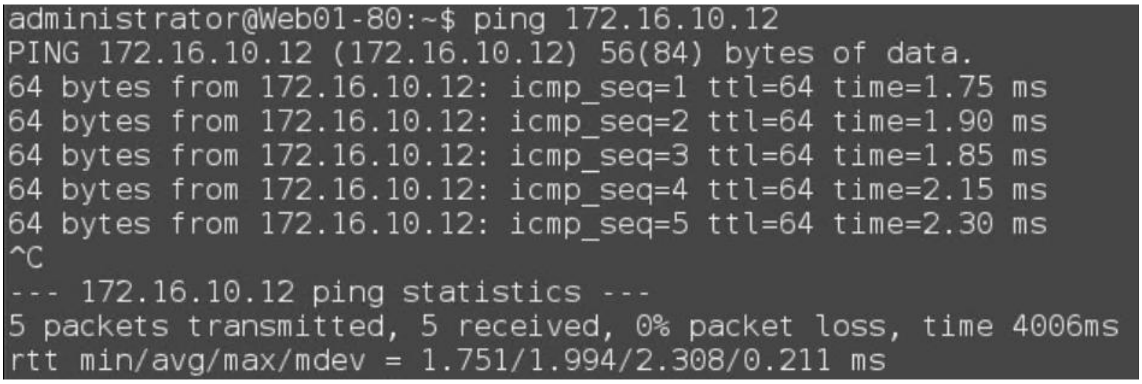


✓ 4	SG1-SG2	1028	SG-1	SG-2	* any	Allow	Distributed Firewall

Figure 136. Firewall Rule.

To prove that everything is well configured two pings are performed: the first one, from Web-01 to Web-02 that should be successful, and the second one, from Web-01 to App-01 that should fail because there is the excluded member, so traffic should be blocked.

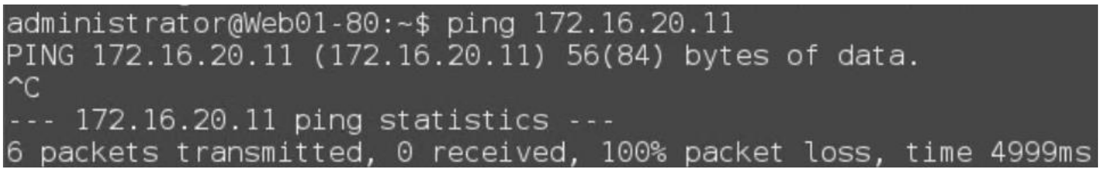
Ping Web-01 → Web-02:



```
administrator@Web01-80:~$ ping 172.16.10.12
PING 172.16.10.12 (172.16.10.12) 56(84) bytes of data.
64 bytes from 172.16.10.12: icmp_seq=1 ttl=64 time=1.75 ms
64 bytes from 172.16.10.12: icmp_seq=2 ttl=64 time=1.90 ms
64 bytes from 172.16.10.12: icmp_seq=3 ttl=64 time=1.85 ms
64 bytes from 172.16.10.12: icmp_seq=4 ttl=64 time=2.15 ms
64 bytes from 172.16.10.12: icmp_seq=5 ttl=64 time=2.30 ms
^C
--- 172.16.10.12 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.751/1.994/2.308/0.211 ms
```

Figure 137. Successful ping between security groups where traffic is allowed.

Ping Web-01 → App-01:



```
administrator@Web01-80:~$ ping 172.16.20.11
PING 172.16.20.11 (172.16.20.11) 56(84) bytes of data.
^C
--- 172.16.20.11 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 4999ms
```

Figure 138. Unsuccessful ping between security groups where traffic is allowed but there is an excluded member.

From previous example, it is demonstrated how easily is to offer security using Security Groups. They are easy to create while they provide easy administration of network flows. Security is independent from network structure.

Microsegmentation in a collapsed tier

This use case is performed in a different environment where all VM belong to the same vSwitch although they are used by different departments providing different services. The thing is that all flows should not be permitted between them, something that happens in traditional networks. Traditional architectures and systems cannot offer microsegmentation between elements from the same subnet. Virtual networks can do this, and it is represented here.

To sum up, NSX allows segmentation in layer 2.

The scenario that is going to be used in this us case is the following one:

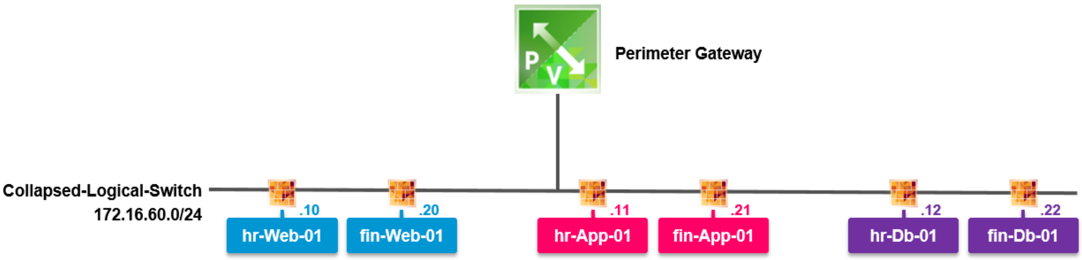
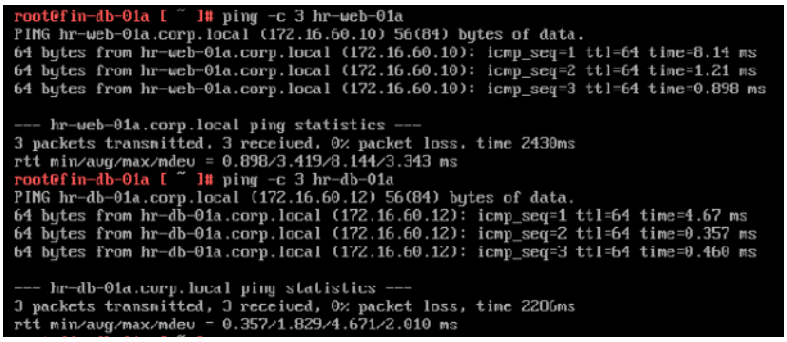


Figure 139. Collapsed tier scenario.

This example is different from the other ones because security groups are created following a dynamic membership criterion. The first SG contain virtual machines that their name has “hr” standing for human resources department, while, the second one, “fin” corresponding to financial department. Due to security reasons, departments have to be separated although it is not performed using different VLANs. Instead of using different VLANs, SG will help to decouple human resources from finances and vice versa. VMs that belong to a same department can exchange any kind of flows.

Before creating any Security Group nor firewall rules, connectivity between “hr” and “fin” is successful:



```
root@fin-db-01a [ ~ ]# ping -c 3 hr-web-01a
PING hr-web-01a.corp.local (172.16.60.10) 56(84) bytes of data.
64 bytes from hr-web-01a.corp.local (172.16.60.10): icmp_seq=1 ttl=64 time=8.14 ms
64 bytes from hr-web-01a.corp.local (172.16.60.10): icmp_seq=2 ttl=64 time=1.21 ms
64 bytes from hr-web-01a.corp.local (172.16.60.10): icmp_seq=3 ttl=64 time=0.898 ms

--- hr-web-01a.corp.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2430ms
rtt min/avg/max/mdev = 0.898/3.419/8.144/3.343 ms
root@fin-db-01a [ ~ ]# ping -c 3 hr-db-01a
PING hr-db-01a.corp.local (172.16.60.12) 56(84) bytes of data.
64 bytes from hr-db-01a.corp.local (172.16.60.12): icmp_seq=1 ttl=64 time=4.67 ms
64 bytes from hr-db-01a.corp.local (172.16.60.12): icmp_seq=2 ttl=64 time=0.357 ms
64 bytes from hr-db-01a.corp.local (172.16.60.12): icmp_seq=3 ttl=64 time=0.460 ms

--- hr-db-01a.corp.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2206ms
rtt min/avg/max/mdev = 0.357/1.829/4.671/2.010 ms
```

Figure 140. Testing connectivity before starting the use case.

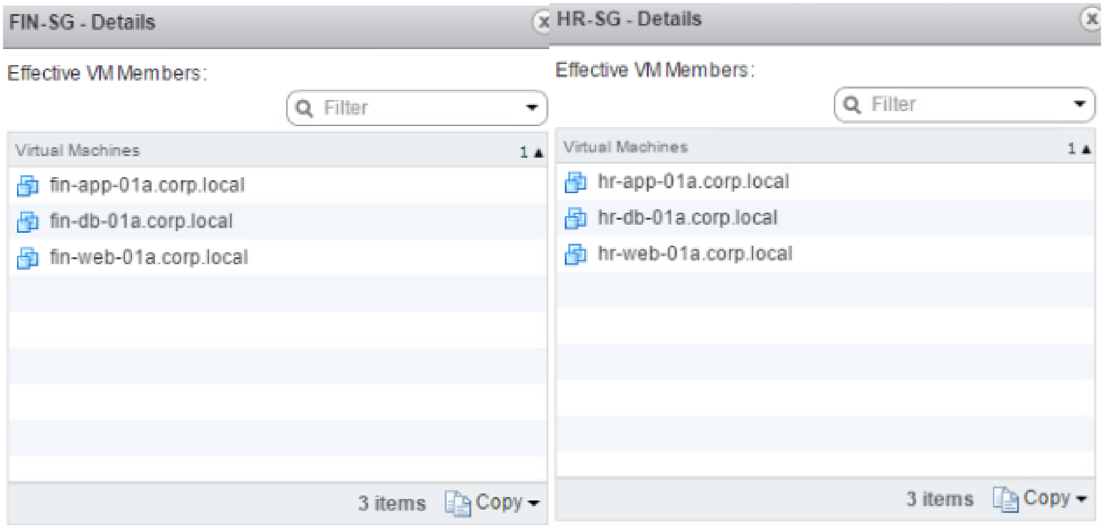
If all virtual machines are connected to a unique vSwitch, they will have connectivity among them by default.

As commented, the main functionality of created dynamic SG’s, is to restrict traffic between departments.

Dynamic created SG are:

Financial Security Group

Human Resources Security Group



FIN-SG - Details

Effective VM Members:

Virtual Machines
fin-app-01a.corp.local
fin-db-01a.corp.local
fin-web-01a.corp.local

3 items

HR-SG - Details

Effective VM Members:

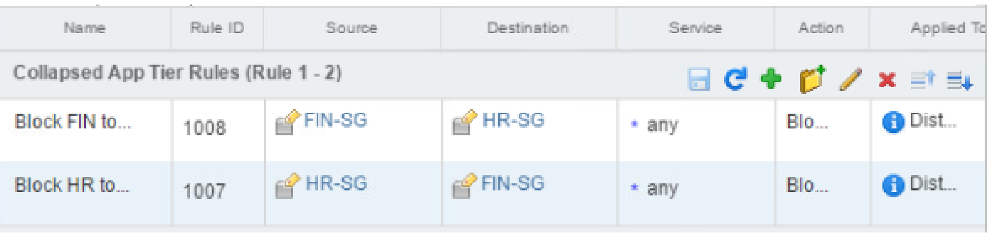
Virtual Machines
hr-app-01a.corp.local
hr-db-01a.corp.local
hr-web-01a.corp.local

3 items

Figure 141. Financial and Human Resources Dynamic Security Groups.

To remember, the designed parameter to build these dynamic security groups is: include virtual machines that their name contains “hr” or “fin”.

The firewall rules to block inter-department traffic are:

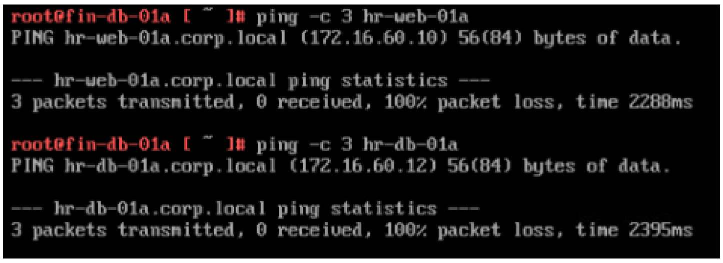


Name	Rule ID	Source	Destination	Service	Action	Applied To
Collapsed App Tier Rules (Rule 1 - 2)						
Block FIN to...	1008	FIN-SG	HR-SG	any	Blo...	Dist...
Block HR to...	1007	HR-SG	FIN-SG	any	Blo...	Dist...

Figure 142. Firewall rules to block inter-department traffic using dynamic SG.

Once SG are created and FW are rules applied, it has to be proven if inter-department traffic is blocked and intra-department traffic is allowed.

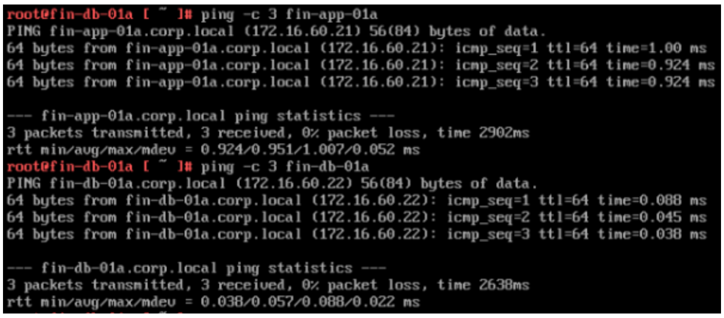
FIN → HR is blocked



```
root@fin-db-01a [ ~ ]# ping -c 3 hr-web-01a
PING hr-web-01a.corp.local (172.16.60.10) 56(84) bytes of data.

--- hr-web-01a.corp.local ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2288ms

root@fin-db-01a [ ~ ]# ping -c 3 hr-db-01a
PING hr-db-01a.corp.local (172.16.60.12) 56(84) bytes of data.

--- hr-db-01a.corp.local ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2395ms
```

Figure 143. Blocking inter-department traffic.

FIN → FIN



```
root@fin-db-01a [ ~ ]# ping -c 3 fin-app-01a
PING fin-app-01a.corp.local (172.16.60.21) 56(84) bytes of data.
64 bytes from fin-app-01a.corp.local (172.16.60.21): icmp_seq=1 ttl=64 time=1.00 ms
64 bytes from fin-app-01a.corp.local (172.16.60.21): icmp_seq=2 ttl=64 time=0.924 ms
64 bytes from fin-app-01a.corp.local (172.16.60.21): icmp_seq=3 ttl=64 time=0.924 ms

--- fin-app-01a.corp.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2902ms
rtt min/avg/max/mdev = 0.924/0.951/1.007/0.052 ms
root@fin-db-01a [ ~ ]# ping -c 3 fin-db-01a
PING fin-db-01a.corp.local (172.16.60.22) 56(84) bytes of data.
64 bytes from fin-db-01a.corp.local (172.16.60.22): icmp_seq=1 ttl=64 time=0.088 ms
64 bytes from fin-db-01a.corp.local (172.16.60.22): icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from fin-db-01a.corp.local (172.16.60.22): icmp_seq=3 ttl=64 time=0.038 ms

--- fin-db-01a.corp.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2638ms
rtt min/avg/max/mdev = 0.038/0.057/0.088/0.022 ms
```

Figure 144. Allowing intra-department traffic.

To conclude this section about microsegmentation and Security Groups an appointment has to be made: security groups can include other security groups. This fact is useful for example in an scenario with HR department that includes production and development virtual machines.

For example:

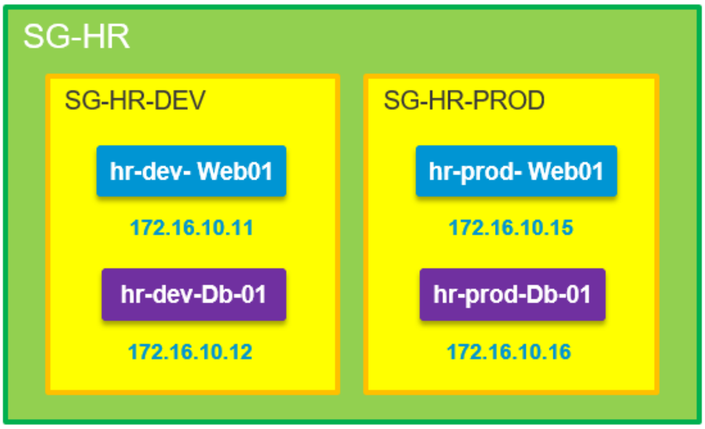


Figure 145. Generic Security Group containing an smaller Security Group.

Collapsed tier use case is the last one about microsegmentation. The other ones, are about Service composer and ASM (Automatic Security Manager) two different ways to manage security policies.

4.2.2. Service Composer

Already introduced in State of The Art, service composer is a container that helps the administrator with the security policies and security groups management. In previous microsegmentation use cases, it has been demonstrated that using SG, firewall rules are simplified and better-managed. The question is, is there any other method to know in a simple view how SG are organized and which security policies and service profiles are associated to one or multiple SGs? The answer is yes and Service Composer is the tool to know which VMs have these configurations applied.

The Service composer is defined at NSX Manager level, so, in the actual scenario if there are security groups that englobe VM from different DCs, security policies and services will be duplicated.

A security policy is a set of endpoints, firewall and network introspection services that can be applied to a security group. The security policies have an associated weight with the policy which determines the order in which security policies are displayed. It has to be commented that, by default, a new policy is assigned the highest weight so that it is at the top of the table. However, the administration can modify the default suggested weight in order to change the order assigned to the new policy.

This use case is intended for critical company applications/departments such as: Bloomberg, Murex or Online Banking. The current case is for Online Banking web servers, they are critical because they are ubicated in DMZ, so the traffic must be restricted as much as possible and extra-measures have to be taken, such us IDS (Intrusion Detection System)[13].

[13] In this thesis it is only shown the properly configuration for IDS, it cannot be demonstration due to a lack of license.

Prior annotations:

- All firewall rules applied in other use cases have been deleted.
- There is a new Security Group called SG-Web-Servers that contains Web-01, Web-02 and Web-03. Web VMs belonging to DC1.
- In the case of an existing FW, the new one defined in service composer will be duplicated. Although it seems unuseful, in terms of visibility, it is practical to know what is defined and what applies which components.

There are two options when working with service composer. One is more efficient and useful than the other one; they are exposed below from less to more useful.

- Option 1: security policy with flow restriction applied to a specific SG.
- Option 2: security policy with flow restriction that applies the SG associated to Security policy[14].

Extrapolating two options mentioned earlier in a real case should be:

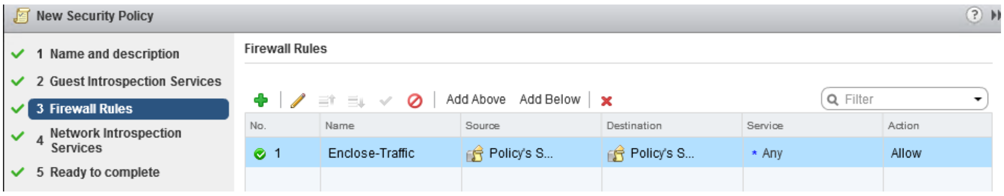
- Option 1: “The security policy only allows web servers to communicate with each other and no one else”.
- Option 2: “Traffic is restricted to members that security policy is applied”. Members can be, web servers but also other Security Groups.

Option 2 it the one chosen for being developed.

1) Set a name and a description of the configuration. For example: Traffic restricted in Security Profile – Security group. Moreover, the administrator can choose whether this is a ‘child’ policy of an existing security policy.

2) Guest introspection services: if there is an available license, prior to define a security policy it is required the creation of a Security Service. Security services are those based on a vendor’s solution, for example TrendMicro. Moreover, there are some service definitions such us Antivirus, File Integrity Monitoring, IDS… they are applied to the Security Policy Security Groups. Guest introspection services can take measures if detect something unusual, for example: after a number of tries failing (brute force attack) it quarantines a VM. It is not developed due to a lack of license.

3) Define Firewall rules that apply the current security group associated to that Security Policy.



New Security Policy

- ✓ 1 Name and description
- ✓ 2 Guest Introspection Services
- ✓ 3 Firewall Rules
- ✓ 4 Network Introspection Services
- ✓ 5 Ready to complete

Firewall Rules

No.	Name	Source	Destination	Service	Action
✓ 1	Enclose-Traffic	Policy's S...	Policy's S...	Any	Allow

Figure 146. Firewall rules defined in a Security Policy.

[14] In option 1, all security policies are intended for a particular SG. In contrast, in Option 2, all security policies are for SG currently associated. When SG is changed, all configurations point to the new SG indicated.

4) Canvas: Canvas is the graphical recompilation for previous steps.

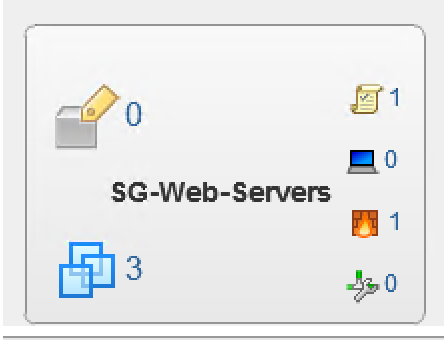


Figure 147. Service Composer: Canvas.

The previous canvas shows exactly what have been configured: one security policy with a single firewall rule, applied to a security group that contains three virtual machines.

Although there is a security policy that has not been created during previous steps it can always be created apart and then added. As security policies could have associated firewall rules, when applying a security rule, the canvas is automatically modified.

For example, a second security policy with a couple firewall rules is added in the example before, the result is:

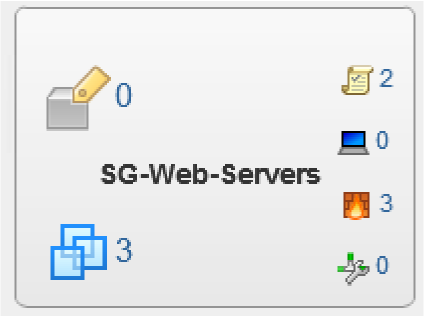


Figure 148. Canvas with multiple security policies.

It results in a new canvas, with two security policies instead of one. And, three firewall rules instead of two. It is accumulative.

An extra verification from this use case is to determine who has a higher priority: distributed firewall or firewall rule applied to a security policy. So, two opposite firewall rules are applied, on in DFW and the other one in the security policy definition:

- Distributed firewall rule: It allows traffic between web servers.
- Security policy firewall rule: It blocks traffic between web servers.

Who has a bigger priority?

A ping is thrown from Web-01 to Web-02 and it is successful. In conclusion, DFW has priority over policy rules.

4.2.3. Visibility and Operations

For security companies, the protection of the environment and the network is everything. These tasks are not only based on using firewalls, anti-malware software, Intrusion Prevention Systems, but also one of the most critical of their essentials is the collection and regular review of event logs.

The continuous revision of logs allows technicians to know which activity patterns are normal and which are not, taking into account that nowadays most of information leaks do not come from outside. Even so, logs play an important role when dealing with prevention and attacks identification. Without a good log collection, retention and analysis, an organization's security will rest on very unstable ground.

Large corporate environments may generate high log volumes and, in a future, logs will be increased due to mobile devices and IoT endpoints. How will such a quantity of logs be managed?

Actual log companies, such as Splunk, AlienVault or ArcSight produce software for searching, monitoring and analysing machine-generate big data, but they cannot interact with the environment. If they could, the management of applications would be more efficient and optimal.

So, the purpose of this use case is to comprehend which management tools are suitable for a dynamic hybrid cloud environment, which are the differences between them and most important, how to work with them.

For these tools, the main objective is to collect machine data strategically to generate insights and troubleshoot any IT infrastructure issues.

There are 4 tools directly related with VMware:

- Log insight: It analyses massive amounts of log data and delivers near real-time monitoring, search and log analytics. It delivers innovative indexing and machine learning based Intelligent grouping. The administrator can design dashboards for stored queries, reports and alerts. It speeds correlation of events across an IT environment.
- Flow Monitoring: It analyses the traffic to and from a vNIC of a virtual machine. It provides a detailed view of traffic when the administrator enables flow monitoring; data provided includes packets transmitted per session and the number of session. Moreover, session details such as: sources, destinations, ports and applications being used. That information allows the creation in situ of firewall rules. ARM (Application Rule Manager is an extension of flow monitoring that will also be tested).
- Trace Flow: It is a traceroute. It tracks the path taken by a packet at every hope. It allows to inject a packet into an vNIC and it shows all the hops of the packet in the virtual network. Moreover, it specifies is the packet is received, delivered or forwarded, also the host, the component type and component name.
- Endpoint Monitoring: It provides information about application processes and associated network connections inside of a guest. Due to it gives end-to-end visibility, it is easy to create firewall rules and perform microsegmentation easily.

Once these tools have been introduced, it is time to demonstrate how they work.

Log Insight

Before to start Log Insight, it has to be said that Log insight analysis and deployment have not been done in the thesis scenario. Log insight requires an extra license that the company could not provide for being used in a use case.

For this reason, an alternative scenario is used. It is not necessary to describe in detail the architecture. This section contains a brief explanation of what Log Insight is, which kind of logs are analysed and how queries performed.

Log Insight is a management logs tool that allows the administrator to create dashboards and reports from queries. The aim of this tool is to have a general and in the same time, a clear vision of what is going on through network elements. Network elements could be routers, firewalls, load balancers…

The steps to deploy and start using vRealize Log Insight are very easy.

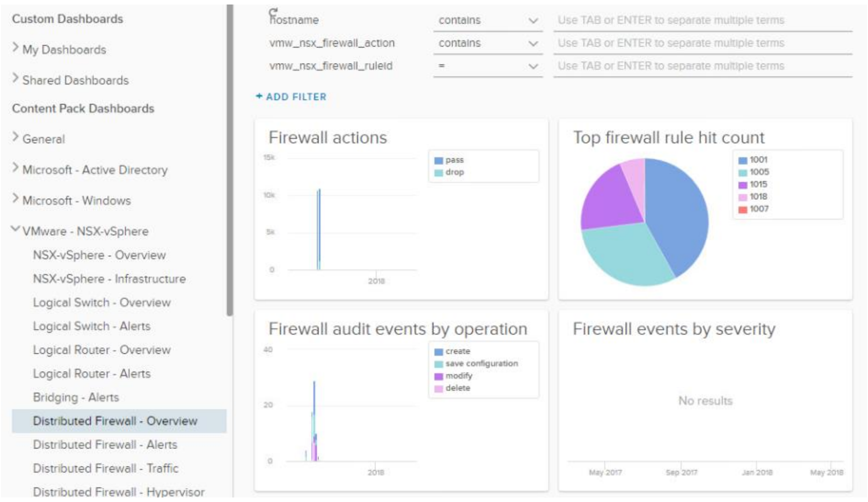
1) Deployment: From VMware the virtual appliance has to be downloaded. After that, deploy the OVA in the vCenter and accepting the End User License.

2) Assign an IP address to the virtual machine. Once it is done, the administrator can access Log Insight via web interface.

3) Login to vRealize Log Insight and select “ingest data from vSphere, Agents and Syslog”. Once is performed, there are content packs that must be downloaded or imported (NSX) through the menu Content Packs and “+ Import content Pack”, then the one desired is selected to be imported.

4) Once the administrator has imported the content pack and the installation is complete, there are so many dashboards and widgets specifics for NSX that appear. Although there are predefined dashboards, they are fully customizable.

5) Now the user can do queries, configure dashboards and create reports. On the top of the page, the tab dashboards can be found:



hostname contains Use TAB or ENTER to separate multiple terms
vmw_nsx_firewall_action contains Use TAB or ENTER to separate multiple terms
vmw_nsx_firewall_ruleid = Use TAB or ENTER to separate multiple terms

Figure 149. vRealize Log Insight Firewall overview dashboard.

From previous figure, it can be extracted that dashboards are focused in main network elements: Logical Routers, Logical Switches, Distributed Firewall, NSX Edge…[15]

6) Analyse data obtained by the system. How logs look like? Is there any option to interact with logs in real time? To show all the logs related to an event and filter out the incidents, what administrator has to do is work with interactive analytics. Interactive analytics section appears after clicking an event from a dashboard:

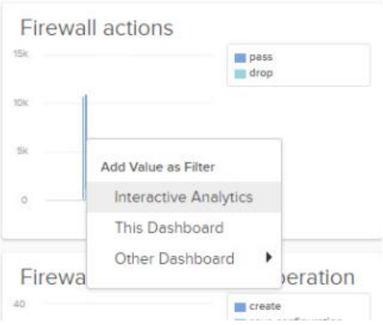
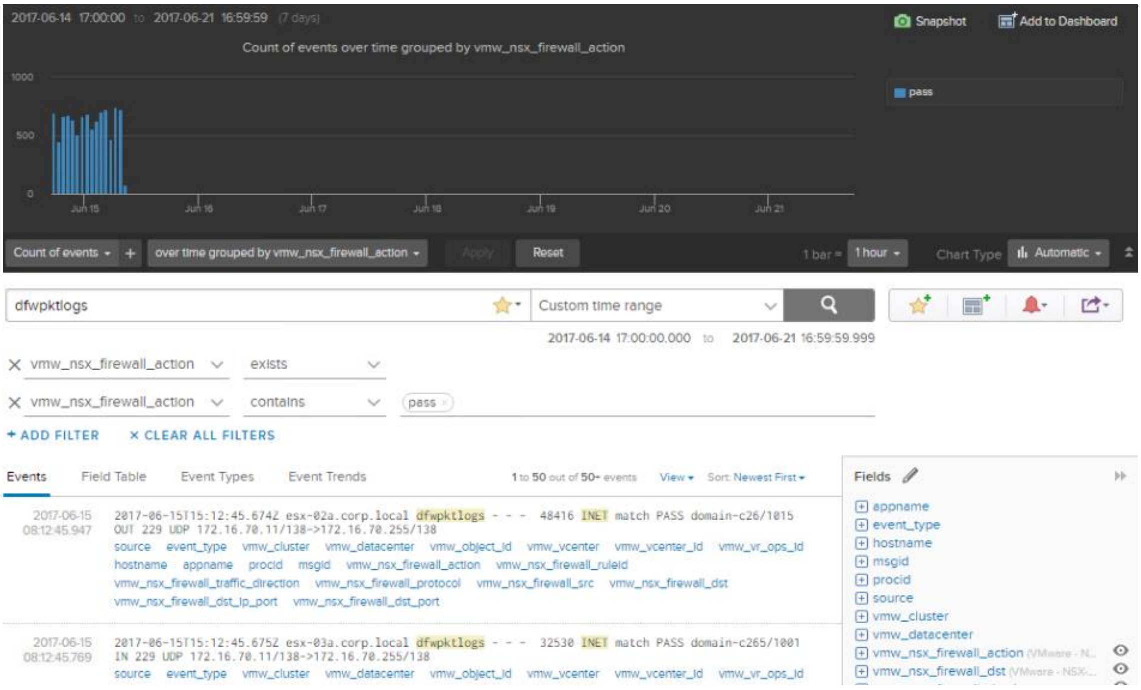


Figure 150. Interactive Analytics selection.

Once Interactive Analytics is selected, a new window opens. This is how Interactive Analytics looks like:



dfwpktlogs — Custom time range

2017-06-14 17:00:00.000 to 2017-06-21 16:59:59.999

× vmw_nsx_firewall_action exists

× vmw_nsx_firewall_action contains pass

```
2017-06-15    2017-06-15T15:12:45.674Z esx-02a.corp.local dfwpktlogs - - -  48416 INET match PASS domain-c26/1015
08:12:45.947  OUT 229 UDP 172.16.70.11/138->172.16.70.255/138
              source  event_type  vmw_cluster  vmw_datacenter  vmw_object_id  vmw_vcenter  vmw_vcenter_id  vmw_vr_ops_id
              hostname  appname  procid  msgid  vmw_nsx_firewall_action  vmw_nsx_firewall_ruleid
              vmw_nsx_firewall_traffic_direction  vmw_nsx_firewall_protocol  vmw_nsx_firewall_src  vmw_nsx_firewall_dst
              vmw_nsx_firewall_dst_ip_port  vmw_nsx_firewall_dst_port

2017-06-15    2017-06-15T15:12:45.675Z esx-03a.corp.local dfwpktlogs - - -  32530 INET match PASS domain-c265/1001
08:12:45.769  IN 229 UDP 172.16.70.11/138->172.16.70.255/138
              source  event_type  vmw_cluster  vmw_datacenter  vmw_object_id  vmw_vcenter  vmw_vcenter_id  vmw_vr_ops_id
```

Figure 151. Interactive Analytics window.

Interactive Analytics allow the user to customize time ranges, add new filters and determine the logs format. It is useful due to logs can be analysed almost in real time; for this reason, the administrator can monitor events in live mode to find out the cause for incidents.

[15] Logs are from last year due to the environment, this environment it is not the one where other use cases are developed.

For example, filters in the previous example are: log has to contain a field with a firewall action and, in addition, the action has to be pass. Other firewall action is drop, which is commonly used to analyse non-allowed traffic and to determine if it is legitimate or not.

Flow Monitoring

Flow Monitoring, as commented before, allows to see and analyse all traffic flows between two virtual machines with the aim of determine if traffic is legitimate or not and act in consequence. For this reason, it is a good tool to analyse and create firewall rules.

Flow monitoring is accessed through vCenter → Networking and Security → Tools → Flow Monitoring. The analysis is restricted to one source vNIC. If more vNIC flows want to be analysed, with this tool it is not possible. The tool that offers the monitoring of simultaneous source vNICs is ARM (Application Rule Manager) which will be explained lately.

The steps to use flow monitoring are:

1) Allow traffic from source vNIC: before configuring flow monitoring parameters, the administrator has to configure a firewall rule allowing all kinds of traffic between vNICs (VMs) that will be analysed. For this case, the rule is: allow any kind of traffic from Web-01 to Web-02 and App-01.

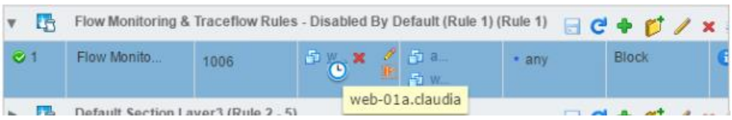

Flow Monitoring & Traceflow Rules - Disabled By Default (Rule 1) (Rule 1)
1
Flow Monito...
1006
w...
a...
w...
any
Block
web-01a.claudia
Default Section Layer3 (Rule 2 - 5)

Figure 152. Flow monitoring firewall rule.

2) Configure flow monitoring parameters in Networking and Security → Tools → Flow Monitoring → Live Flow as the following image shows:

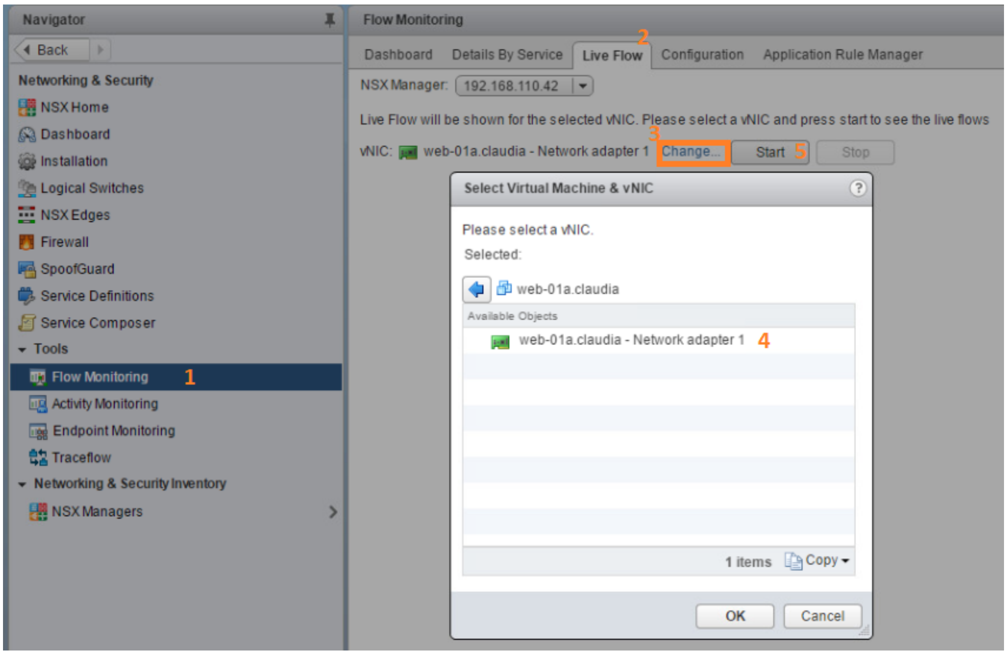
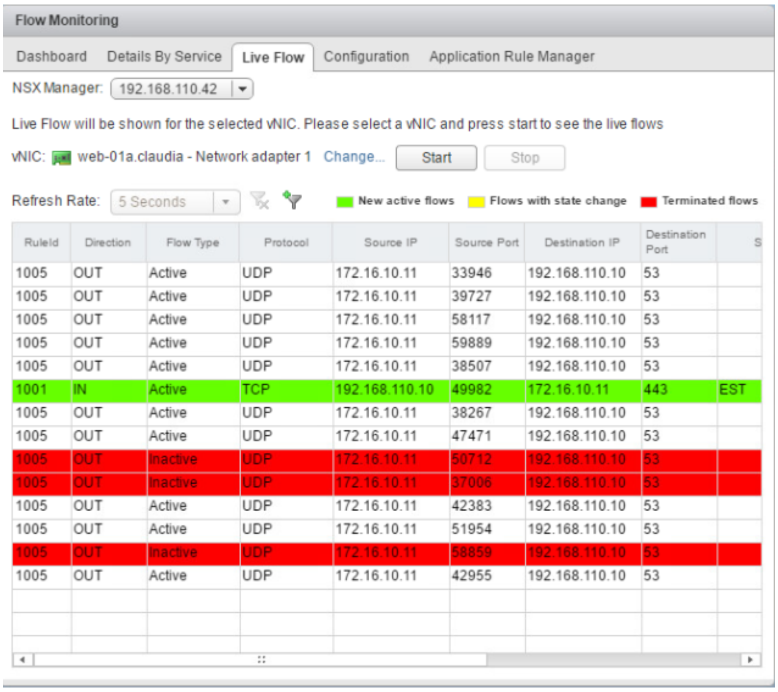


Figure 153. Flow monitoring configuration.

3) Start flow monitoring and analyse live flows. Once monitoring started, flows start to appear- They are classified in three different colours as: new active flows, flows with state change and terminated flows. To generate flows, the administrator has to use the analysed VM, perform habitual queries and functionalities.



NSX Manager: 192.168.110.42

Live Flow will be shown for the selected vNIC. Please select a vNIC and press start to see the live flows

vNIC: web-01a.claudia - Network adapter 1

Refresh Rate: 5 Seconds

🟩 New active flows 🟨 Flows with state change 🟥 Terminated flows

RuleId	Direction	Flow Type	Protocol	Source IP	Source Port	Destination IP	Destination Port	S
1005	OUT	Active	UDP	172.16.10.11	33946	192.168.110.10	53	
1005	OUT	Active	UDP	172.16.10.11	39727	192.168.110.10	53	
1005	OUT	Active	UDP	172.16.10.11	58117	192.168.110.10	53	
1005	OUT	Active	UDP	172.16.10.11	59889	192.168.110.10	53	
1005	OUT	Active	UDP	172.16.10.11	38507	192.168.110.10	53	
1001	IN	Active	TCP	192.168.110.10	49982	172.16.10.11	443	EST
1005	OUT	Active	UDP	172.16.10.11	38267	192.168.110.10	53	
1005	OUT	Active	UDP	172.16.10.11	47471	192.168.110.10	53	
1005	OUT	Inactive	UDP	172.16.10.11	50712	192.168.110.10	53	
1005	OUT	Inactive	UDP	172.16.10.11	37006	192.168.110.10	53	
1005	OUT	Active	UDP	172.16.10.11	42383	192.168.110.10	53	
1005	OUT	Active	UDP	172.16.10.11	51954	192.168.110.10	53	
1005	OUT	Inactive	UDP	172.16.10.11	58859	192.168.110.10	53	
1005	OUT	Active	UDP	172.16.10.11	42955	192.168.110.10	53	

Figure 154. Live flow monitoring.

4) Determine which flows are necessary and which not. After stopping live flow, the administrator is able to determine which flows must be allowed and which ones blocked.

The next point is about ARM (Application Rule Manager). It is like flow monitoring. The main difference between flow monitoring and ARM is that ARM can monitor more than one VM.

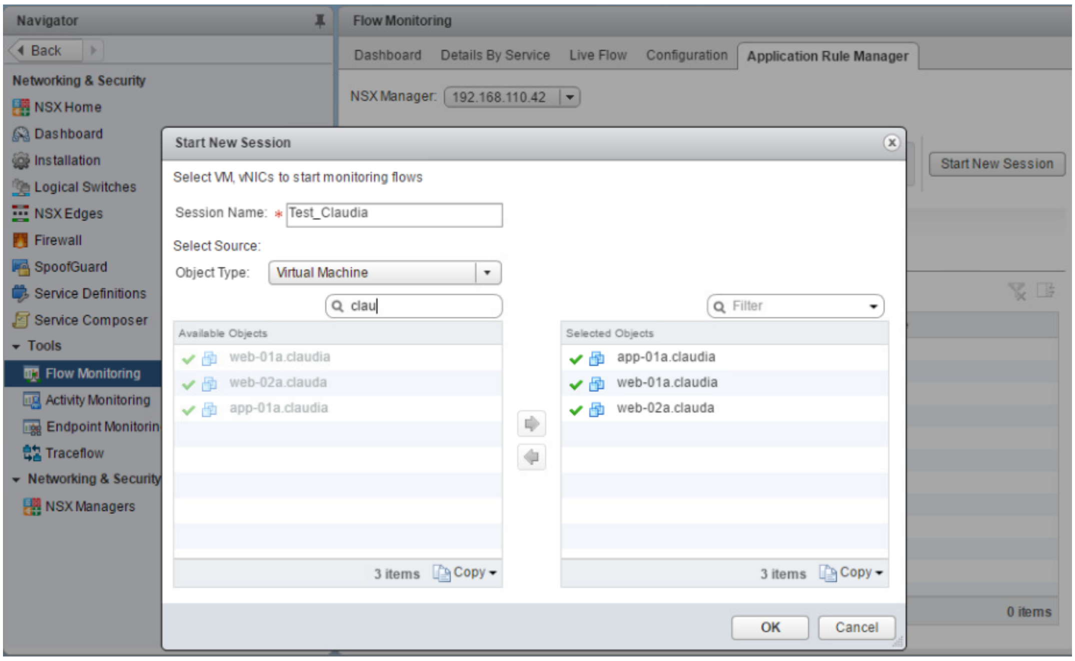
ARM (Application Rule Manager)

Application Rule Manager is an extension of flow monitoring. The main difference is, while live flow is constantly updating displayed data, ARM shows the captured flows when it is stopped, summarizing what has been flowing through the network.

Steps to use ARM are similar to the previous ones performed for live flow:

1) Allow traffic between VMs that will be analysed; for this case, the VMs are Web-01, Web-02 and App-01.

2) Configure ARM: the administrator has to set a session name and select the virtual machines that will be analysed.



NSX Manager: 192.168.110.42

Start New Session

Select VM, vNICs to start monitoring flows

Session Name: * Test_Claudia

Select Source:

Object Type: Virtual Machine

clau

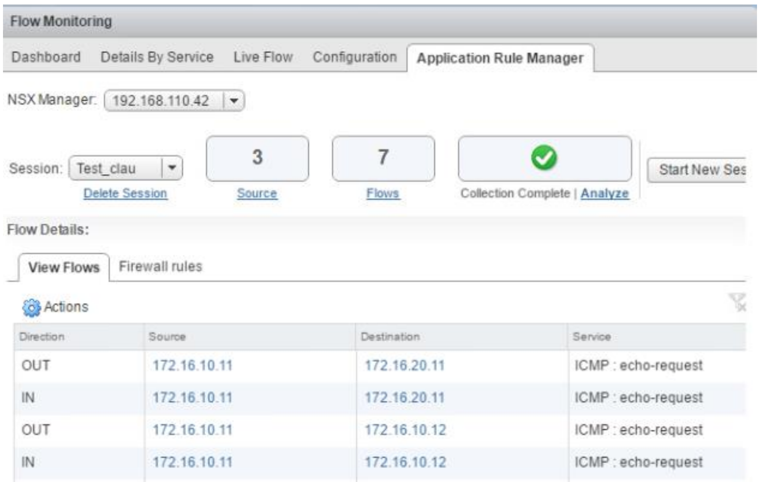
Available Objects
web-01a.claudia
web-02a.clauda
app-01a.claudia
3 items

Selected Objects
app-01a.claudia
web-01a.claudia
web-02a.clauda
3 items

Figure 155. ARM configuration parameters.

3) Start new session. Once session is started, the user has to generate habitual traffic between the selected VMs. For example, as VM selected are Web and App servers, using the navigator access different Web servers. After that, web servers may do some service requests to app ones. This would determine flows that are necessary for an appropriate application behaviour. For this particular case, some pings from Web-01 have been thrown.

4) Finish session and Analyze. After a session is finished, the user has to click “Analyze”. These are flow details that ARM shows:



NSX Manager: 192.168.110.42

Session: Test_clau

3 Source

7 Flows

Collection Complete | Analyze

Flow Details:

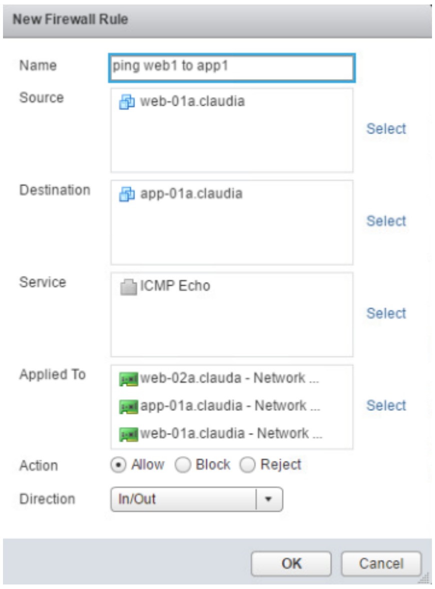
Direction	Source	Destination	Service
OUT	172.16.10.11	172.16.20.11	ICMP : echo-request
IN	172.16.10.11	172.16.20.11	ICMP : echo-request
OUT	172.16.10.11	172.16.10.12	ICMP : echo-request
IN	172.16.10.11	172.16.10.12	ICMP : echo-request

Figure 156. ARM flow details.

As a reminder: 172.16.10.11 is Web-01, 172.16.10.12 is Web-02 and 172.16.20.11 is App-02.

From the previous image it can be extracted that there are some flows that correspond to the pings thrown from Web-01 to Web-02 and App-01. The service is ICMP but it could have been HTTP, HTTPS or whatever.

Once flows are reviewed the user can create firewall rules easily. The flow has to be selected, then, actions and finally, create firewall rule. For example, the administrator decides that the first flow, which corresponds a ping from Web-01 to App-01, has to be admitted. The new firewall rule is created:



New Firewall Rule

Name: ping web1 to app1

Source: web-01a.claudia

Destination: app-01a.claudia

Service: ICMP Echo

Applied To: web-02a.clauda - Network ...
app-01a.claudia - Network ...
web-01a.claudia - Network ...

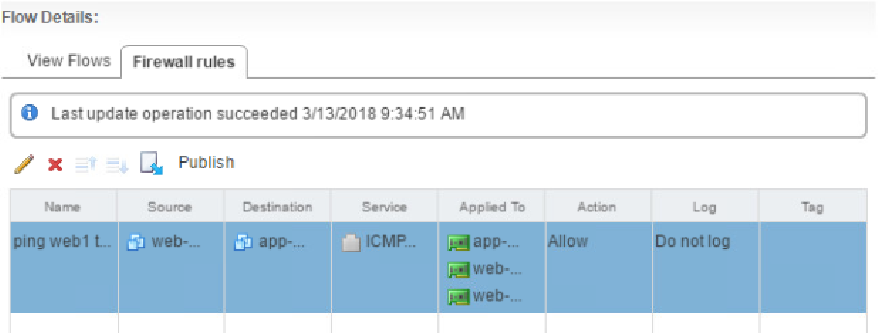
Action: ● Allow ○ Block ○ Reject

Direction: In/Out

Figure 157. ARM firewall rule creation.

The previous rule, as commented, allow pings from Web-01 to App-01. Moreover, the administrator can decide where this rule is going to be applied. In previous case, it is applied in vNICs of VMs analysed although it is only necessary in Web-01 and App-01.

After that, the new firewall rule appears in the Distributed Firewall.



Flow Details:

Last update operation succeeded 3/13/2018 9:34:51 AM

Name	Source	Destination	Service	Applied To	Action	Log	Tag
ping web1 t...	web-...	app-...	ICMP...	app-... web-... web-...	Allow	Do not log	

Figure 158. Firewall rule created with ARM.

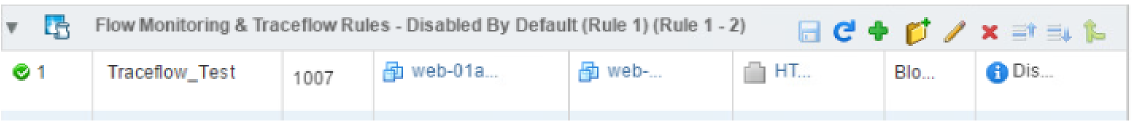
A reflection from live flow and ARM is that both tools are very useful not only for the creation and flow analysis. They are useful to clean and optimizing firewall rules. In production environments it is critical to clean firewalls because most of the applications are complex. For this reason, firewall rules configurations are degraded over time. Maybe there are too permissive rules or other ones that are not used anymore. Nowadays there is a similar tool which is called “ALGOSEC”, it performs firewall rule grouping and optimization after analysing firewall traffic flows. In conclusion, this is a tool that it is interesting to know how it works because it offers functionalities not only for security purpose, it is also for networking.

Traceflow

Traceflow tracks the path taken by a packet at every hope. It allows to inject a packet into an vNIC and it shows all the hops of the packet in the virtual network.

This use case is to determine which route is taken by an HTTP packet that goes from Web-01 to Web-02. It has to be mentioned that the following example performs a traceflow between VM pertaining to a same L2 segment. The steps to configure and use traceflow are:

1) Allow/deny traffic from virtual machines. Before to start traceflow, HTTP traffic from Web-01 to Web-02 is blocked; the objective of blocking traffic is to figure out which VMware components discard the packets.

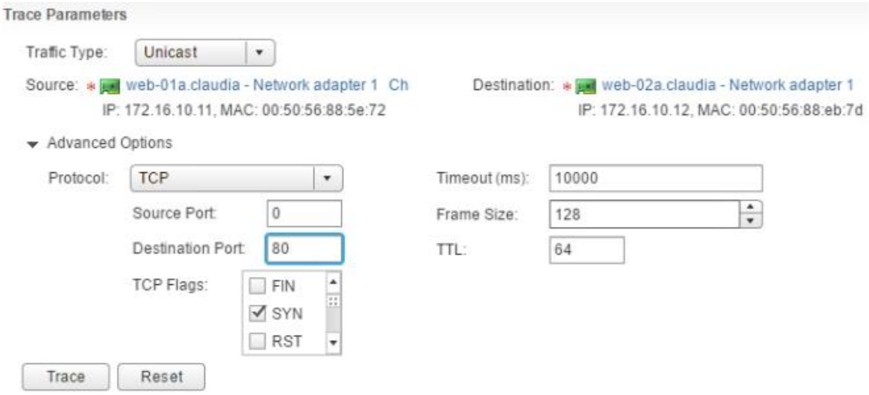


Flow Monitoring & Traceflow Rules - Disabled By Default (Rule 1) (Rule 1 - 2)

✓ 1	Traceflow_Test	1007	web-01a...	web-...	HT...	Blo...	Dis...

Figure 159. Traceflow firewall rule.

2) Configure trace flow. To configure it, the administrator has to go to Networking & Security → Tools → Traceflow. There, vNICs (source and destination) have to be selected. In advanced options, the administrator has the possibility to configure some parameters for the packer: Protocol, destination port, frame size, TCP flags… As the purpose of this example is to see where an HTTP packet is blocked, the configuration is the next one:



Trace Parameters

Trafic Type: Unicast

Source: * web-01a.claudia - Network adapter 1 Ch
IP: 172.16.10.11, MAC: 00:50:56:88:5e:72

Destination: * web-02a.claudia - Network adapter 1
IP: 172.16.10.12, MAC: 00:50:56:88:eb:7d

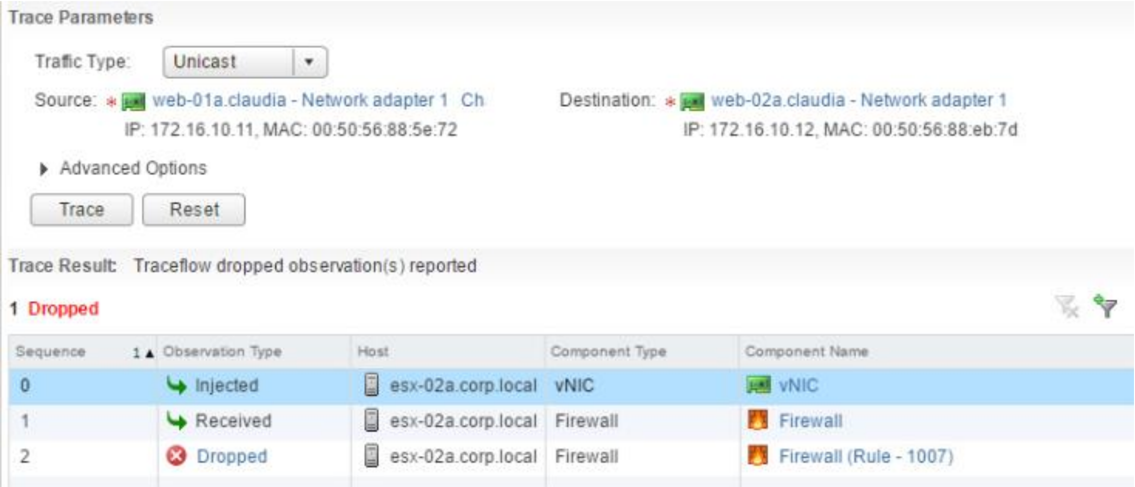
Advanced Options

Protocol: TCP
Source Port: 0
Destination Port: 80
TCP Flags: □ FIN ☑ SYN □ RST

Timeout (ms): 10000
Frame Size: 128
TTL: 64

Figure 160. Traceflow configuration.

3) Start trace and analyse results. The result is predictable: the packet should be blocked somewhere due to the blocking firewall rule (FW rule ID: 1007).



Trace Parameters

Trafic Type: Unicast

Source: * web-01a.claudia - Network adapter 1 Ch
IP: 172.16.10.11, MAC: 00:50:56:88:5e:72

Destination: * web-02a.claudia - Network adapter 1
IP: 172.16.10.12, MAC: 00:50:56:88:eb:7d

Advanced Options

Trace Result: Traceflow dropped observation(s) reported

1 Dropped

Sequence 1 ▲	Observation Type	Host	Component Type	Component Name
0	Injected	esx-02a.corp.local	vNIC	vNIC
1	Received	esx-02a.corp.local	Firewall	Firewall
2	Dropped	esx-02a.corp.local	Firewall	Firewall (Rule - 1007)

Figure 161. Traceroute results.

As expected, the packet is dropped in the firewall because the HTTP traffic was blocked previously by Rule 1007.

This tool is very useful when troubleshooting. Sometimes packets are lost in the network and no one knows the reasons. With traceroute it is easy to find the exact point where a packet is dropped. Moreover, it is used to quickly identify problems and determine if there is any issue in the NSX data path. An actual tool that is similar is NetBrain; NetBrain not always performs traceroutes, it also can perform network discovery.

Endpoint Monitoring

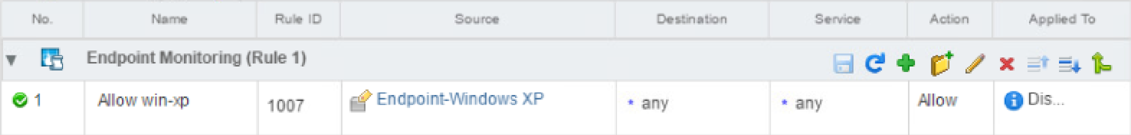
Endpoint monitoring provides information about application processes and associated network connections inside of a guest. Due to it gives end-to-end visibility, it is easy to create firewall rules and perform microsegmentation. It also analyses intra and inter Security Group communication.

This use case is performed in an alternative environment. For this reason, IP Addresses, VM names could be different.

Requirements for this experiment:

- Windows XP virtual machine.
- Security group containing multiple Windows XP virtual machines.
- Endpoint monitoring supports up to 20 VMs in monitored Security Group.

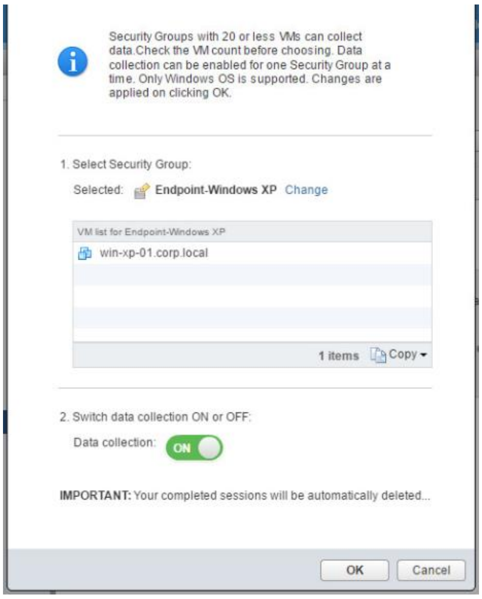
1) Apply a firewall rule allowing any traffic from SG that will be monitored:



No.	Name	Rule ID	Source	Destination	Service	Action	Applied To
▾	Endpoint Monitoring (Rule 1)						
✓ 1	Allow win-xp	1007	Endpoint-Windows XP	any	any	Allow	Dis...

Figure 162. Endpoint Monitoring: Firewall rule allowing traffic from Enpoint-Windows XP SG.

2) Configure Endpoint Monitoring: to configure it, the administrator has to go to Networking & Security → Tools → Endpoint Monitoring. After that, a Security group has to be chosen.



Security Groups with 20 or less VMs can collect data.Check the VM count before choosing. Data collection can be enabled for one Security Group at a time. Only Windows OS is supported. Changes are applied on clicking OK.

1. Select Security Group:

Selected: Endpoint-Windows XP Change

VM list for Endpoint-Windows XP

win-xp-01.corp.local

1 items

2. Switch data collection ON or OFF:

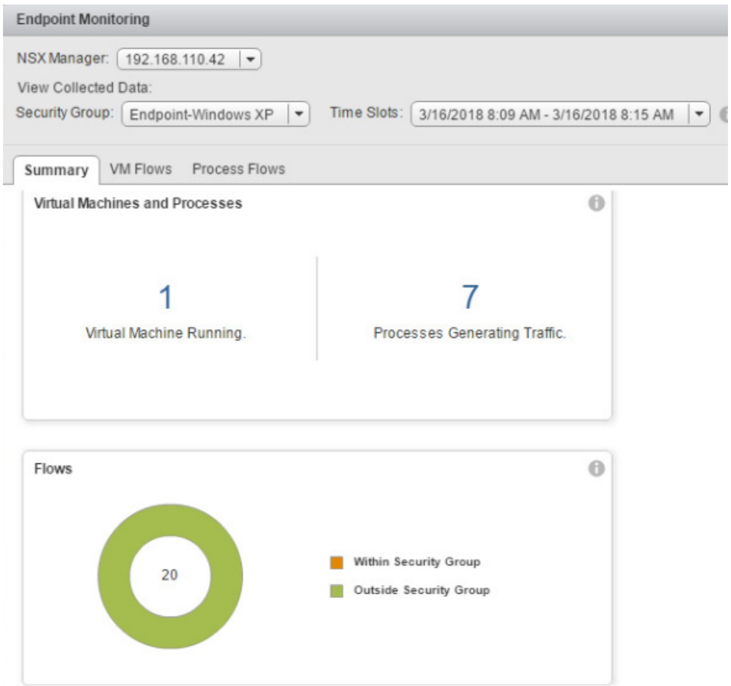
Data collection: ON

IMPORTANT: Your completed sessions will be automatically deleted...

Figure 163. Endpoint monitoring configuration.

It can be appreciated that Endpoint-Windows XP security group is composed by a single virtual machine, the one that is going to be monitored. Once the security group is selected, the monitorization starts.

3) Stop monitoring and results analysis.



Endpoint Monitoring

NSX Manager: 192.168.110.42

View Collected Data:

Security Group: Endpoint-Windows XP Time Slots: 3/16/2018 8:09 AM - 3/16/2018 8:15 AM

Summary VM Flows Process Flows

Virtual Machines and Processes

1
Virtual Machine Running.

7
Processes Generating Traffic.

Flows

20

🟧 Within Security Group
🟩 Outside Security Group

Figure 164. Endpoint monitoring results.

When the monitorization finishes, results are the ones showed in the previous figure. It can be appreciated that the VM monitored has seven processes that generate traffic with a total amount of twenty flows. A flow is any unique stream of network traffic as identified by its packet type, source and destination IP, and port. All flows are outside the security group. These results make sense since there are no other members in the SG.

Another interesting information that this monitoring brings us is the representation through bubbles. The bubbles show the directions of the flows besides indicate if the destination is a member of the origin security group or not. The detailed flow view includes the process name, version and number of flows being generated by each process.

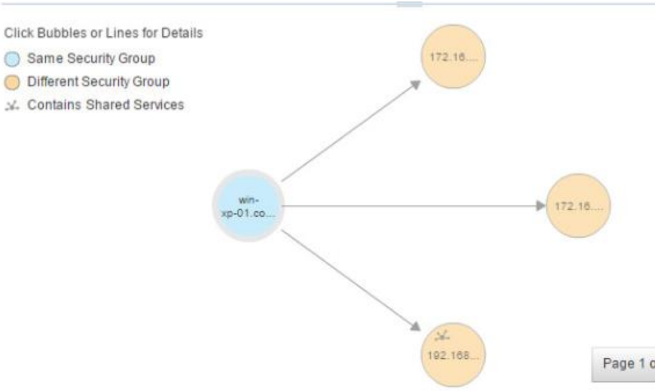


Figure 165. Endpoint monitoring: bubble diagram showing flows from VMs of the monitored SG.

These diagrams help us determine if there are process relationships between virtual machines, this fact contributes with the creation of new security groups due to more than VM could have process in common and they can be treated equally. In virtual environments everything works with tags. Tags are useful because it is a way to generalize. It is true that some things may have particularities but usually there are common things; for shared behaviours is better to combine resources.

Finally, the process flows tab. This tab contains a list of process that the virtual machine is executing. Moreover, there is information about process versions. An interesting thing is to figure out what which is the destination of flows, in particular, if flows are outside or inside the security group.

Endpoint monitoring concludes the use cases section.

5. Budget

This section summarizes a financial plan concerning the costs of hardware, licenses and engineer hours.

As a remember, the hardware and licenses are the following ones:

- Physical Server: Dell R430.
- Licenses:
  - ESXi’s: VS6-EPL-C VMware vSphere 6 Enterprise Plus - 32 CPU
  - vCenter: VCS6-STD-C VMware vCenter Server 6 Standard for vSphere 6 - 2 instance key
  - NSX: NX-ENT-C VMware NSX Enterprise per Processor - 16 CPUs

VMware determines their licenses in function of processors (sockets) or Virtual Machines instead of the number of cores.

The project budget is:

PROJECT DETAILS		
Project name: Deploy a SDN Network using VMware. Project code: NSX-0001 Project Manager: R. Rico Date: MAY 2018		
EXPENSES	2018 Cost Estimate	
Hardware Resources	Units	Price/u
Server: DELL R430 2*8 CORE 2.60GHZ E5-2640 V3 128GB 4*960GB SAS SSD H730	1	7.655,69€
Licenses	Instance Key	Price/key
vSphere: VS6-EPL-C VMware vSphere 6 Enterprise Plus - 32 CPU	1	4.917,69€
vCenter: VCS6-STD-C VMware vCenter Server 6 Standard for vSphere 6*	2	8.683,50€
NSX Enterprise: NX-ENT-C VMware NSX Enterprise per Processor - 16 CPUs	1	7.500€
Staff Resources	Total Hours	Price/h
Project Manager	16	100€
Project Engineer	160	65€
TOTAL COST		
Hardware		7.655,69€
Licenses		29.784,69€
Staff		12.000€
TOTAL		**49.440,38€**

Table 6. Budget.

6. Conclusions and future development

During this project it has been possible to investigate and learn from this wonderful world of networking. Something that, during my years of the degree has not been my priority however this will change undouble.

First of all, I would like to emphasize that the virtual world does not only include SDN but also it is closely linked with NFV (Network Function Virtualization), SD-WAN (Software Defined- Wide Area Network), IoT (Internet of Things) and Cloud.

This point summarizes a little the conclusions that have been appearing throughout the project, not only for the SDN technology, but also for the preparation of the environment as well as the different use cases. In addition, there is a small section of things that can seriously affect the environment directly related with problems that have arisen during the development of the thesis.

Finally, there are some proposals for a future development, either of what has been done as new options that are directly related.

6.1. Conclusions

SDN it is based in decoupling the control plane and data plane which centralises intelligence and management. This is something that is the traditional networks is decentralized and therefore more complex.

SDN allows easy management that can be remotely accessible and, also provides scalability. One thing that should be especially highlighted is that you can make big changes in the architecture with a single click.

Another advantage of SDN is that organizations can meet business demands efficiently and flexibly.

Another thing that has been appreciated and that is very important, is the growing trend of the logs. Increasingly, more things can be monitored and in more points of the network, fact that supposes a considerable augment in the quantity that there is to handle. So, it is important to have good tools that allow the user not only to manage, but to visualize in an easy way such amount of information.

Focusing in VMware and NSX, there are other advantages that must be highlighted. For instance, NSX can increase data centre security by enabling a rich set of security services and microsegmentation. This will lead us to the subject of the tags. In virtual environments, the symmetry of the network is not prioritized, but in what elements they have what things in common. It is about configuring generically everything that can be associated in, for example, security groups. Something that clearly makes management more comfortable, faster and therefore, optimal.

Besides, in NSX, virtual networks that are created can provide a complete set of network services.

Another benefit provided by VMware NSX is the optimization of the data centre traffic, where the north-south traffic is reduced as well as the latency. All of this is achieved thanks to the optimization of east-west traffic patterns, load-balancing capabilities and distributed software firewall in the software.

The thesis has shown that with simple actions from a console or a client, the administrator can make significant changes very comfortably, such as the extension of a network or the addition of a load balancer. In the case of the balancer it is vital to include it as a conclusion and advantage since it is a very useful tool / device that in traditional environments should be added as an extra part, when on the contrary, in NSX it is part of the whole solution. Although if desired, NSX can incorporate third-parties such as F5.

Another aspect that was included as a use case was microsegmentation. In such a case, it has been demonstrated that can be performed in layer 2. Something that it is not possible in traditional networks.

6.2. Tips

However, at it is the case of all failures, good learnings have been the outcome. Here are some tips that I have been learned from the mistakes I made.

It is very important to size and estimate the matter of deployment. A good and adequate management of resources is the key to handle and work with the scenario properly.

Also, the link from the physical to the virtual world should be well configured. Otherwise, the isolation between the two worlds will be bad and will mix requests that should not be mixed. This happened at the beginning of the project where virtual DHCP requests ended up arriving at the physical router and the VMware management was lost as well as the internet access, since the purpose of that router was to provide internet to some users of the company.

The order of assignment of the network adapters must be respected since this affects the virtual machine to which they are assigned. Likewise, when PortGroups are created in the Hypervisor, it is very important which ones they should be and how they want to be called since once they are created their names cannot be modified.

Something that was noticed during the licensing period was that although vESXi hosts were licensed on vSphere Client, when accessing vSphere web client, that vESXi hosts had to be re-licensed; this was observed due to although the order to turn vESXi hosts was performed, they remained switched off.

Another tip for the initial deployment is that there are virtual machines that weigh a lot. So, when they are deployed, it is better doing it with the server's own USB port. Otherwise, it could happen that they were not deployed or that they took many hours, or even days.

Snapshots are a very good tool that can save you, returning the virtual machine to the state it had before becoming corrupted. Even so, it has been seen that if they are managed in a wrong way the consequences can be terrible, overflowing the storage and having to reset everything completely.

Particular attention must be paid when applying default deny firewall rules. If previously the administrator has not defined the minimum flows that the environment needs to work, such as: allow connection between vCenter and managers, etc. The environment remains collapsed since the basic flows are not enabled. In case this happens, it is very likely that the administrator loses access and has to do a redeployment of the virtual network. Here it is appreciated that having snapshots are a very useful thing.

6.3. Future development

From this project can come others, directly related to VMware or others more generic that are based on SDN.

Another project, very similar, using VMware could be done but deploying the datacentres on two different physical servers.

This would cause that instead of working with vESXi, it would work with ESXi. That is, ESXi hosts used would not be virtualized.

Another project that would be very interesting, could be the deployment of the same network but using a different product, for example, Cisco ACI, and compare performance, manageability, security and costs.

Bibliography

[1] Tech Target (2013, March). Software-defined networking (SDN), from https://searchsdn.techtarget.com/definition/software-defined-networking-SDN

[2] Open Networking. Software-Defined Networking (SDN) Definition, from https://www.opennetworking.org/sdn-definition/

[3] Sdx Central. What is Software Defined Networking (SDN)? From, https://www.sdxcentral.com/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/

[4] Webopedia. SDN - software defined networking from, https://www.webopedia.com/TERM/S/software_defined_networking.html

[5] ONF (Open Networking Foundation) (2013, December 12). SDN Architecture Overview from, https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf

[6] Cisco. The seven benefits of software-defined networking from, https://www.cisco.com/c/en/us/solutions/software-defined-networking/benefits.html

[7] Ángel Leonardo Valdivieso Caraguay, Lorena Isabel Barona López, Luís Javier García Villalba, “Evolution and Challenges of Software Defined Networking”, (Quito, Ecuador), 2012.

[8] Sdx central. (2013, April). SDN market tendency from, https://www.sdxcentral.com/wp-content/uploads/2013/04/plexxi_sdn_final5.jpg

[9] Andrew Lerner. (2014, December 8). Predicting SDN Adoption from, https://blogs.gartner.com/andrew-lerner/2014/12/08/predicting-sdn-adoption/

[10] Mandi Nowitz. (2018, March 2). SDN Market Poised to Cross $88 Billion by 2024 from, http://www.transformingnetworkinfrastructure.com/topics/virtualization/articles/437269-sdn-market-poised-cross-88-billion-2024.htm

[11] IEEE. Standardization from, https://sdn.ieee.org/standardization

[12] Mark Haranas. (2016, February 9). Top 10 SDN Market Leaders In The Data Center And Enterprise In 2016 from, https://www.crn.com/slide-shows/networking/300079644/top-10-sdn-market-leaders-in-the-data-center-and-enterprise-in-2016.htm/pgno/0/15?itc=refresh

[13] CISCO Solutions: Data Center Virtualization from, https://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/index.html

[14] VMware products from, https://www.vmware.com/products/nsx.html

[15] Huawei SDN solutions from, http://e.huawei.com/en/solutions/technical/sdn

[16] HPE Networking applications from, https://www.hpe.com/my/en/networking/applications.html

[17] Juniper SDN products and services from, https://www.juniper.net/us/en/products-services/sdn/

[18] Cumulus Networks from, https://cumulusnetworks.com

[19] RedHat SDN blog from, https://www.redhat.com/blog/verticalindustries/sdn-fundamentals-for-nfv-openstack-and-containers/

[20] Arista Software Driven Cloud Networking from, https://www.arista.com/en/products/software-driven-cloud-networking

[21] iDatalabs (2018). Companies using NSX from, https://idatalabs.com/tech/products/vmware-nsx

[22] VMware Products from, https://www.vmware.com/products/nsx.html

[23] VMware Community from, https://communities.vmware.com/welcome

[24] vRealize Log Insight from, https://www.vmware.com/products/vrealize-log-insight.html

[25] VMware Licenses from, https://store.vmware.com/store?Action=home&Env=BASE&Locale=en_IE&SiteID=vmwde

[26] Dell R430 EdgePower Server from, https://www.dell.com/learn/us/en/04/shared-content~data-sheets~en/documents~dell-poweredge-r430-spec-sheet.pdf

Appendices

The next figure represents how virtual machines are distributed among vESXi's:

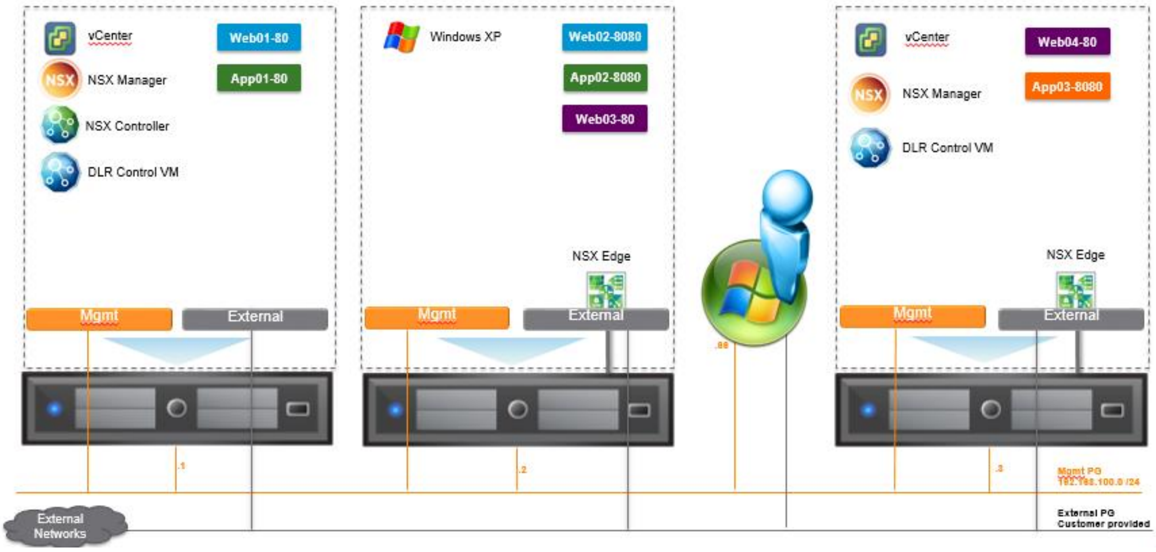
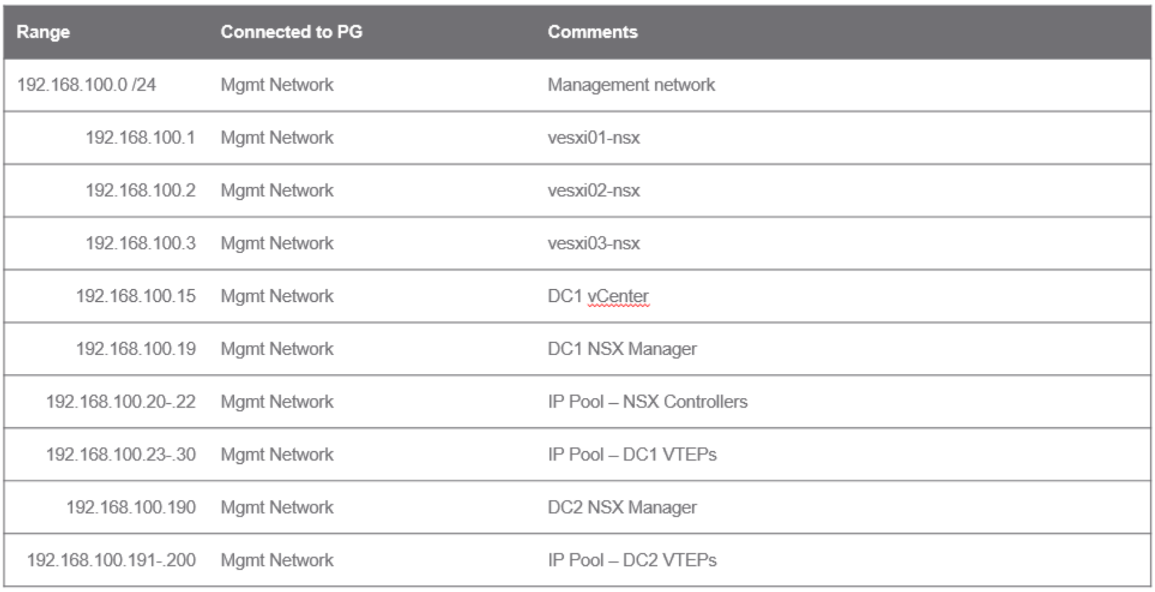


Figure 166. Virtual Machines distribution.

In the following tables, there are included all IP Addresses used in the main scenario of this project:



Range	Connected to PG	Comments
192.168.100.0 /24	Mgmt Network	Management network
192.168.100.1	Mgmt Network	vesxi01-nsx
192.168.100.2	Mgmt Network	vesxi02-nsx
192.168.100.3	Mgmt Network	vesxi03-nsx
192.168.100.15	Mgmt Network	DC1 vCenter
192.168.100.19	Mgmt Network	DC1 NSX Manager
192.168.100.20-.22	Mgmt Network	IP Pool – NSX Controllers
192.168.100.23-.30	Mgmt Network	IP Pool – DC1 VTEPs
192.168.100.190	Mgmt Network	DC2 NSX Manager
192.168.100.191-.200	Mgmt Network	IP Pool – DC2 VTEPs

Table 7. IP Addressing.

Glossary

Distributed Firewall (DFW): it is a firewall that provides visibility and control for virtualized workloads and networks. It applies firewall rules between a VM's vNIC and its connectivity to the Distributed vSwitch.

Distributed Logical Router (DLR): it is a virtual appliance that contains the routing.

ESXi: Bare-metal hypervisor installed over the physical server. It handles VMs with different Operation Systems. ESXi is the exclusive naming for hypervisors in VMware.

Load Balancer: it is device that distributes network or application traffic across several servers.

Logical switch (vSwitch): it is a virtual switch that its main function is to provide connectivity between VMs and hypervisor to the physical network,

Microsegmentation: It is a method that consists on creating secure zones in data centers and cloud deployments that allows to isolate workloads from one another and, secure them individually.

Native Bridging: it is a capability that to connect traditional workloads and legacy VLANs to virtualized networks using VXLAN.

NSX: VMware NSX is a virtual networking and security software product family created from VMware's.

NSX Controller: it provides control plane functions for switching and routing.

NSX Edge Services Gateway: it is a virtual device that provides gateway services and network edge security to isolate a virtual network. It provides routing services, moreover, it also provides common gateway services such as DHCP, NAT, VPN and Load Balancing.

NSX Manager: it is the management plane of the solution.

Port group (PG): it is an aggregation of multiple ports under a common configuration that provides a stable anchor point for virtual machines connecting to labeled networks.

Routing Area: an area is a logical collection of OSPF networks, routers, and links that have the same area identification.

Service Composer: it is a container that helps the administrator with the security policies and security groups management.

Security Group (SG): it is a virtual container that can contain multiple object types, for example, logical switch, IPset, vNIC, VM…

Security Policy: it is a group of network and security services.

Security Service: it is a service definition, for instance, intrusion prevention system, that it is applied in a security policy.

Transport Zone (TZ): it is like a tag, a parameter that controls to which hosts a logical switch can reach.

vCenter Server: it is a tool to manage the virtual infrastructure from a single console.

Virtual Machine (VM): Software computer that runs an Operating System and a set of applications. It provides the same functionalities than physical hardware although it is more portable, secure and easier to manage. It consists of several files.

Vmkernel NIC: It is a special construct used by the vSphere host to communicate with the outside world.

vNIC: Virtual Network Adapter.

vSphere Distributed Switch: virtual switch that provides connectivity between virtual machines from different DCs.

vSphere Standard Switch: virtual switch that provides connectivity between virtual machines that reside in same DC.

VTEP: VXLAN Tunnel Endpoint.

VXLAN: Framework for overlying virtualized layer 2 networks over layer 3 networks.