

# NSX End-User Computing Design Guide

---

## ABSTRACT

This document details the design and deployment details of VMware NSX in a Horizon End-User Computing environment .

## Table of Contents

<b>1</b>	<b>Intended Audience.....</b>	<b>4</b>
<b>2</b>	<b>Introduction .....</b>	<b>4</b>
<b>3</b>	<b>NSX and Horizon Virtual Desktop Deployment.....</b>	<b>5</b>
3.1	Deployment Topology for Horizon – Physical View .....	6
3.2	Deployment Topology for Horizon – Logical View .....	7
3.3	Deployment Topology with NSX & Horizon.....	7
<b>4</b>	<b>Deploying Horizon with NSX Network Virtualization .....</b>	<b>9</b>
4.1	Logical View .....	10
4.2	Scalable Topology .....	11
<b>5</b>	<b>Securing Horizon with NSX Micro-Segmentation.....</b>	<b>12</b>
5.1	Creating Micro-segments with NSX for Horizon end-user computing .....	16
5.2	Introduction to Service Composer .....	19
5.3	Security Groups and Policy.....	21
5.4	Deploying distributed firewall to protect Horizon infrastructure.....	28
5.5	Securing desktop pools using distributed firewall .....	29
5.6	Identity based micro-segmentation for desktop pools.....	29
<b>6</b>	<b>Deploying NSX Load-Balancing for Horizon .....</b>	<b>30</b>
6.1	Building The Infrastructure .....	30
6.2	Certificates .....	31
6.3	User Types.....	31
6.4	Access Point .....	31
6.5	Security Server .....	32
6.6	Connection Servers.....	32
6.7	Datacenter Edge Firewalls .....	34
6.8	Considerations for Deployment.....	34
6.9	Network Topology .....	35
6.10	Edge Services Gateway – Deployment.....	36
6.11	NSX-V Load balancing .....	39
6.12	NSX-V Load Balancing Access Point.....	45
6.13	Understanding Access Point packet Flow.....	45
6.14	Deploying Access Point Servers with NSX Load Balancing.....	46
6.15	Deploying the Access Point in BLAST EXTREME Mode – port sharing.....	50
6.16	Deploying the Access Point Servers in Direct Mode .....	52
6.17	NSX-V Load Balancing Security Servers.....	55
6.18	Understanding Security Server packet flow .....	55
6.19	Deploying the Security Servers in Tunnel Mode.....	56
6.20	Deploying the Security Servers in Direct Mode.....	58
6.21	NSX-V Load Balancing Connection Servers .....	61
6.22	Deploying Connection Server Direct Mode .....	61
6.23	Deploying Connection Server – Tunnel Mode .....	63
<b>7</b>	<b>Appendix .....</b>	<b>66</b>
7.1	Application Rules Used .....	66
7.2	NSX Edge Command Line Interface Load balancer Troubleshooting .....	66
7.3	Configuring the Pool Behavior to Support Transparent topologies.....	69

7.4 Log Insight..... 69  
7.5 Edge configuration customization..... 70

## 1 Intended Audience

This document is targeted towards virtualization, networking and security architects interested in deploying Horizon for virtual desktops and NSX in a vSphere environment. This document highlights design and deployment considerations for utilizing NSX to implement network virtualization, create a secure end-user environment, and load-balance for Horizon infrastructure.

*Note: A solid understanding based on hands on experience with both NSX-v and Horizon products are a pre-requisite for successfully understanding this design guide.*

Revision History:

Version	Updates	Comments
1.0	none	First Release

## 2 Introduction

The Software Defined Data Center (SDDC) is defined by server virtualization, storage virtualization and network virtualization. Server Virtualization has already proved the value of SDDC architecture in reducing costs and complexity of compute infrastructure. VMware NSX provides the third critical pillar of SDDC and extends the same benefits obtained from the virtualization compute to the data center. NSX accelerates core network provisioning, networking and security services provisioning, thus, simplifying network operations and improve network economics.

With network virtualization, the functional equivalent of a “network hypervisor” , NSX reproduces the complete set of layer 2 to layer 7 networking services (e.g., switching, routing, firewalling and load balancing) in software. As a result, these services can be programmatically assembled in any arbitrary combination, to produce unique, isolated virtual networks in a matter of seconds. NSX also provides a platform for various security services – both network based as well as endpoint based. NSX provides various built-in services like L2-L4 firewall, sensitive data detection, activity monitoring. Additionally, security vendors can leverage its guest introspection and network introspection frameworks to deliver next-gen firewall, IDS/IPS, Agentless AV, File Integrity Monitoring and Vulnerability Management. More information about these functions and design considerations are available in the NSX Design Guide [here](#).

Horizon delivers hosted virtual desktops and applications to end users through a single platform. These desktop and application services—including RDS-hosted applications, packaged applications with VMware ThinApp®, software-as-a-service (SaaS) applications, and even virtualized applications from Citrix—can all be accessed from one unified workspace across devices, locations, media, and

connections. Leveraging closed-loop management and optimized for the software-defined data center, Horizon helps IT control, manage, and protect the Windows resources that end users want at the speed they expect and with the efficiency that business demands.

This design guide provides recommended practices and topologies to optimize interoperability between the NSX platform and the Horizon platform to deploy a secure end-user environment in their SDDC. This NSX end-user computing design guide is intended for customers who would like to utilize the benefits of Network Virtualization, Micro-Segmentation, and load-balancing in their brownfield/greenfield Horizon virtual desktop environment.

### 3 NSX and Horizon Virtual Desktop Deployment

For a Secure End User Environment, NSX is used to deliver micro-segmentation, networking services and network virtualization in a Horizon virtual desktop deployment.

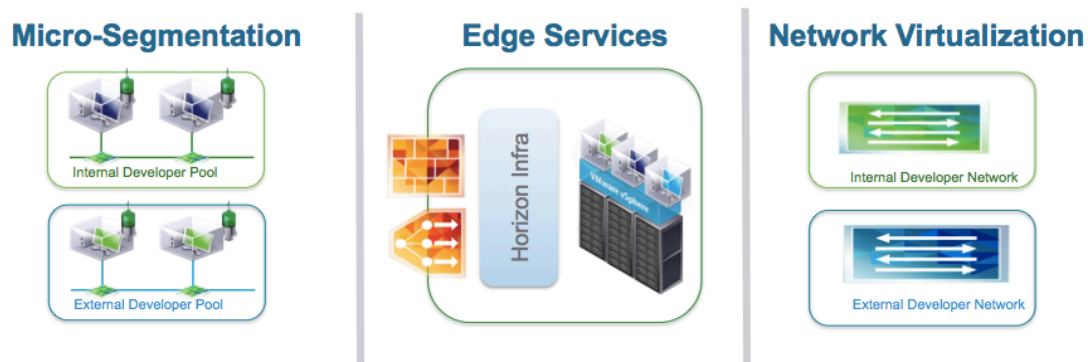


Figure 1: Functionalities delivered by NSX for Horizon virtual desktop deployment.

Figure 1 depicts the various functionalities delivered by NSX for a secure end user environment. They are the following:

- i. **Micro-segmentation**  
NSX as a security platform enables micro-segmentation in the Horizon with View deployment. NSX enables fine-grained desktop to desktop and desktop to enterprise application controls using distributed firewall and edge firewall. In addition to that, NSX can protect Horizon – View infrastructure from attacks. NSX platform provides third party vendors to utilize guest and network introspection frameworks to provide next-generation firewall, IDS/IPS, agentless AV.
- ii. **Networking and Edge Services**  
NSX provides software based networking services like load balancers, VPN termination, SSL offloads, NAT. In addition to that, NSX integrates

with physical services like load balancers, DHCP servers to allow customers to use their existing infrastructure or use NSX built in services to deploy virtual desktops.

- iii. Network Virtualization  
NSX provides core-networking services in software like switching and routing that can be automatically provisioned to create various topologies as required. This allows for an elastic way to spin up or down new desktop pools or expand existing desktop pools on an existing infrastructure. Network Virtualization is also a key tenant of Micro-segmentation, providing the rapid provisioning of isolated network segments.

### 3.1 Deployment Topology for Horizon – Physical View

Horizon deployments are deployed in pods. Each pod contains two physical portions – Horizon Management Block and Horizon Desktop Block. Figure 2 shows the physical Horizon pod topology.

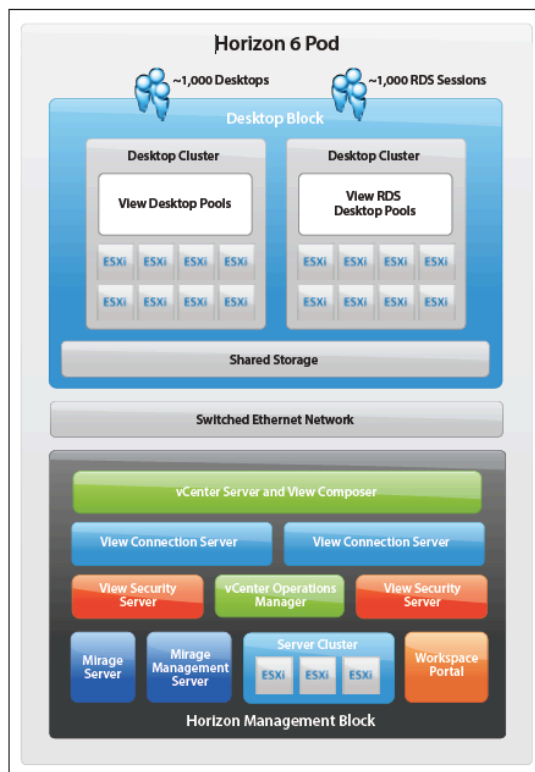


Figure 2: Deployment Topology - Physical

Horizon Management Block contains ESXi hosts that will contain all the management components deployed for the Horizon infrastructure. Horizon Desktop Block contains the ESXi hosts where virtual desktops will be created. Each of the blocks is managed by a separate vCenter Server. Each pod can support up to 2000 concurrent users or desktops.

### 3.2 Deployment Topology for Horizon – Logical View

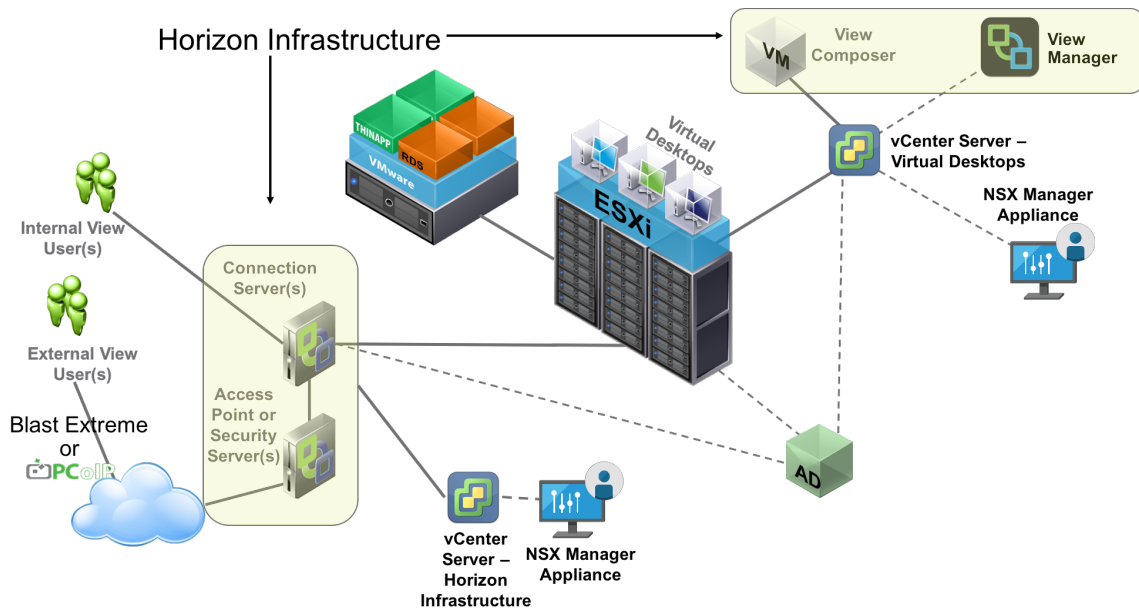


Figure 3: Deployment Topology - Logical

Figure 2 shows the deployment topology of NSX in a Horizon virtual desktop deployment. Horizon virtual desktop deployments have a control plane notion and a data plane notion. Connection Servers and View Composers are primarily part of the control plane. The Security servers (utilized in Horizon 6) or access points (used in Horizon 7 and later) are primarily part of the data plane. NSX Manager and View Manager will be part of the management plane.

View Users will use the Horizon virtual desktop client from their devices to request for a desktop. The access point or security servers in the DMZ zone will handle this request. The security servers will then ask the connection server for a connection to a virtual desktop. Connection servers will authenticate the user and the connectivity with Active Directory. Once authenticated, Connection servers will be notified about the AD groups that will reflect the desktop pool the user is supposed to get the desktop from. Connection servers will either then give a desktop that is currently un-utilized or will request the View Composer to create a new desktop. This desktop connection is then sent back to the requesting Horizon client via the Security Servers. Further communication between the user and the virtual desktop provided primarily happens over the PCoIP or Blast Extreme protocol via the Security Servers or Access Points, respectively.

### 3.3 Deployment Topology with NSX & Horizon

There will be two NSX Managers created in a deployment with NSX and Horizon. The first NSX manager will be connected to the vCenter Server managing the

Horizon Infrastructure while the second NSX Manager will be connected to the vCenter Server managing the desktop pools. Figure 4 shows the logical deployment topology.

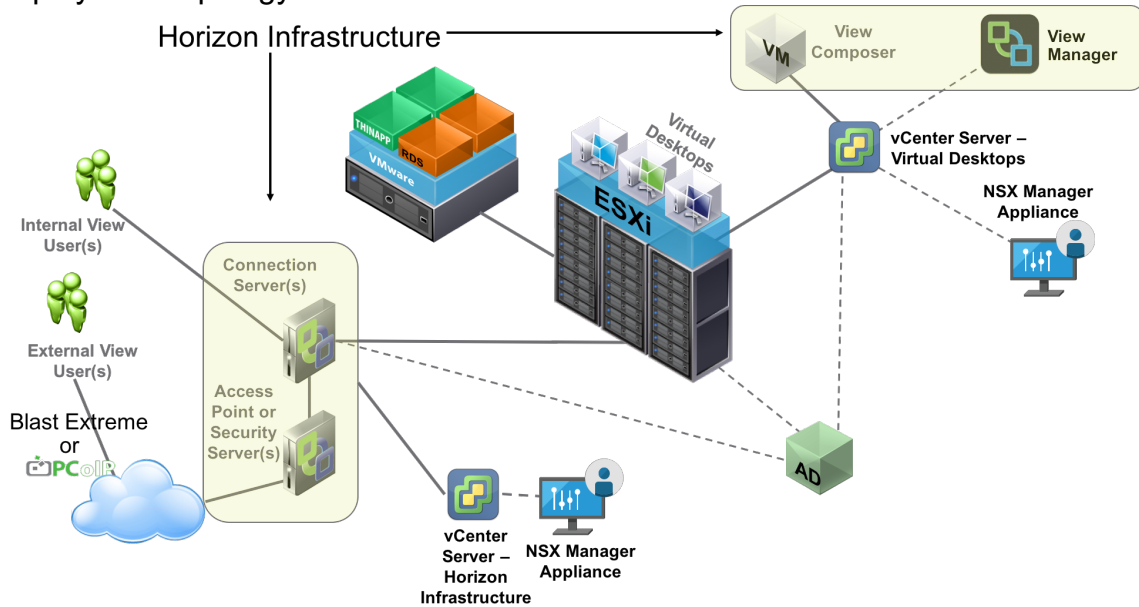


Figure 4: Deployment Topology with NSX - Logical

From a Physical topology perspective, we will expand the topology shown in Figure 2. NSX design guide requires three different clusters for most deployments, they are, management cluster, compute cluster and edge cluster. We will define a Combined Horizon + NSX management cluster for both the Infrastructure side and the Desktop side. This cluster will contain most of the management appliances from both Horizon, NSX and any Operational or Security Management consoles. Compute and Edge clusters will be separate for infrastructure and desktop environments. Connection Servers will be deployed as part of the compute clusters in the infrastructure environment while the Security servers will be deployed at the edge clusters in the infrastructure environment. Figure 5a shows the building blocks for a physical deployment topology while Figure 5b shows the physical infrastructure and how a NSX with Horizon deployment will look in a datacenter.



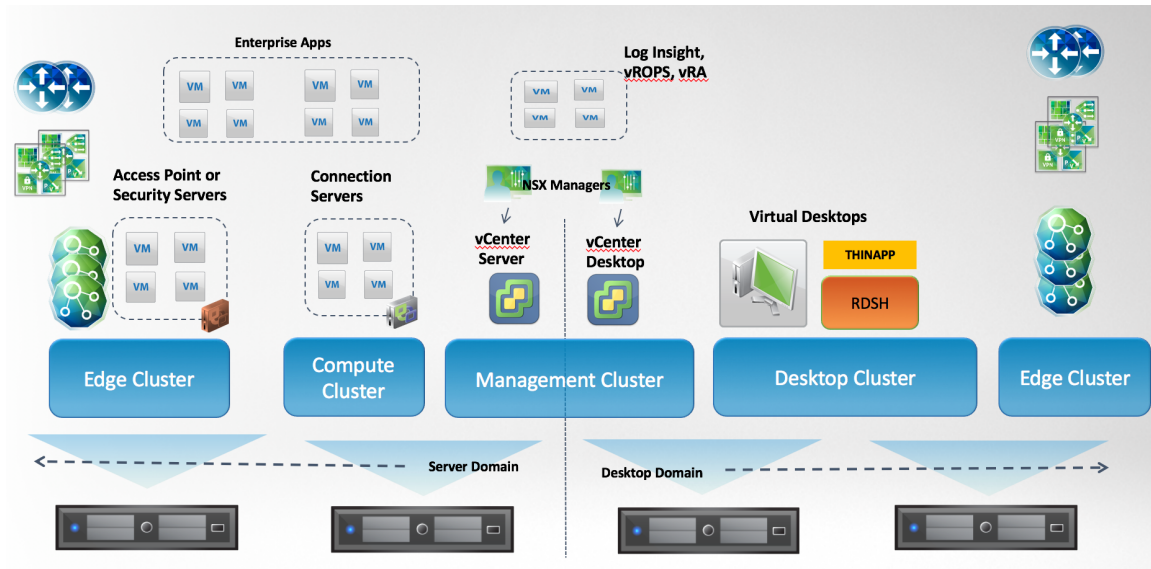


Figure 5a: Deployment Topology with NSX – Physical

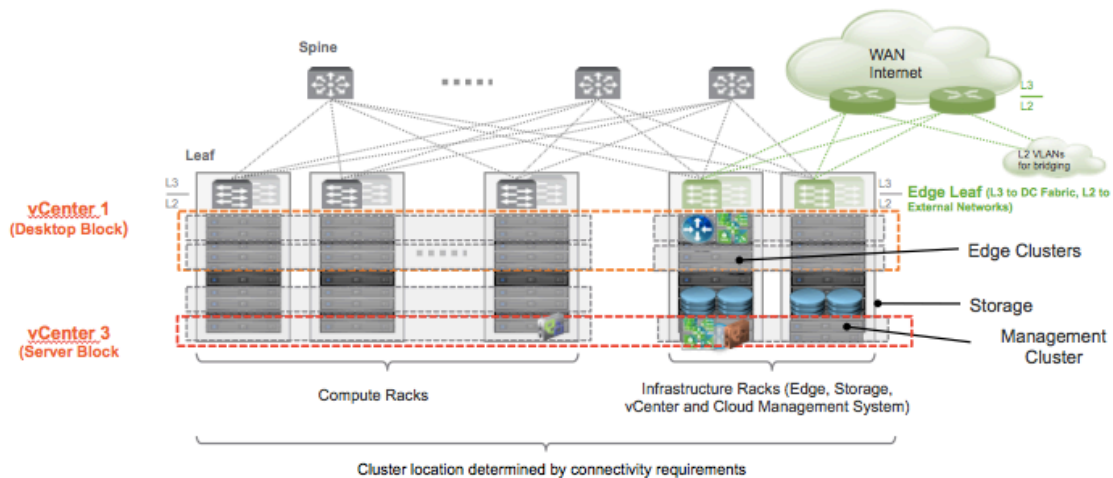


Figure 5b: Deployment Topology with NSX – Physical

## 4 Deploying Horizon with NSX Network Virtualization

This section shows a logical topology of how Horizon can be deployed over overlay networks. This will include deploying both the Horizon infrastructure components and Horizon desktop pools connected to overlays created by NSX. We will discuss various design considerations as part of this section.

## 4.1 Logical View

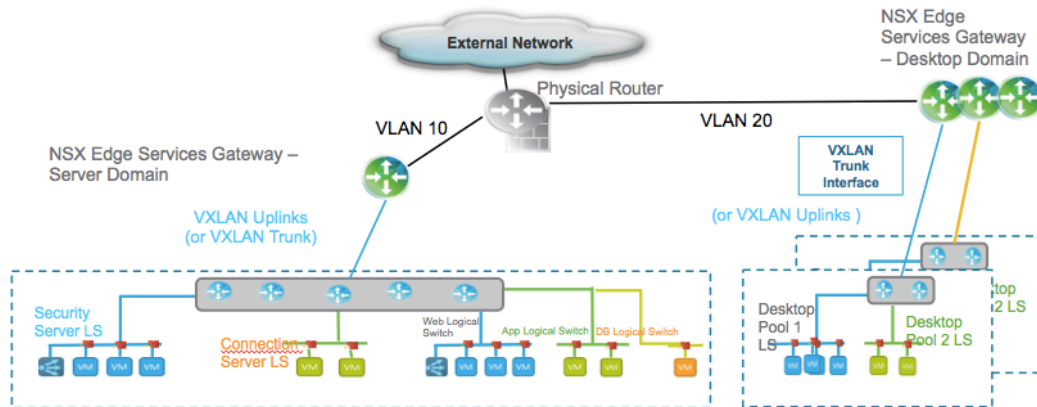


Figure 6: Deploying Horizon with NSX Overlays – Logical Topology

Figure 6 contains the logical view of the Horizon virtual desktop deployment using overlay networks. Access point/Security Servers and Connection Servers will be deployed in separate logical switches in the Server domain of NSX. This will be separated from the other enterprise applications. Enterprise applications will be deployed under separate logical switches. A distributed logical router(DLR) will connect these security servers and connections servers so that they can talk to each other. The DLR will then be uplinked to Edge Services Gateway or a set of Edge Services Gateway. The design considerations for the edge services gateway will be similar to as described in the NSX design guide.

Desktop Pools can be created over one or many logical switches. Most organizations either create the desktops based on user departments or desktops based on functionality provided by the desktops.

Horizon provides 4 basic types of users and bandwidth requirements. Table 1 describes the details of various workers and bandwidth usage.

Type of Worker	Average Bandwidth Consumption	Peak Burst Bandwidth Consumption
Task Worker	70 Kbps	500 Kbps
Basic Office Worker	150Kbps - 2.5Mbps	750 Kbps – 10 Mbps
Video User	7 Mbps	30 Mbps
Power User	30-80 Mbps	50-120 Mbps

Desktops typically do not require east-west communication with other desktops in their pools or with other desktops in other pools. Most of the connectivity for desktops is limited to:

1. Connecting to particular applications in the datacenter for which they are authorized.
2. Connecting to the Internet.
3. Connecting to the users to either the internal network or the external network.

Network administrators can group desktop pools in specific logical switches by user types consuming certain amount of bandwidth. This will help them in designing the edge service gateway capacity planning. Network administrators can also group desktop pools in logical switches based on applications or desktop types created. In all of these cases, logical switches can directly be trunked to edge gateways for north-south communication. Desktop pools will connect to edge gateways deployed as part of the desktop domain. Figure 7 provides three different ways of designing desktop pools. The main advantage of using NSX overlays is the ability to create all the three different topologies in the same environment.

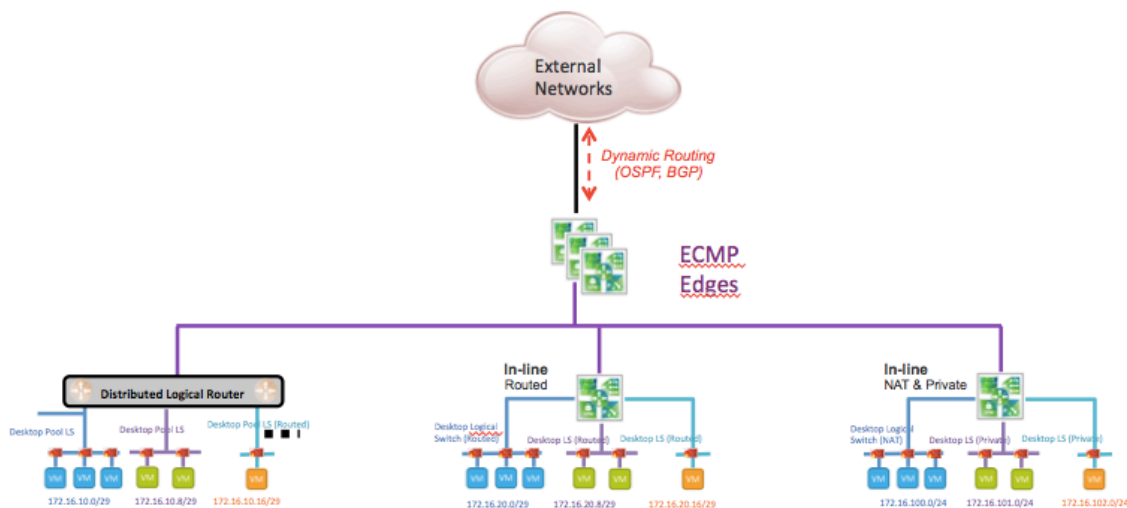


Figure 7: Different ways for deploying Horizon desktops with NSX Overlays

## 4.2 Scalable Topology

NSX provides a highly scalable desktop pool deployment topology. Figure 8 provides a view of a scalable topology.

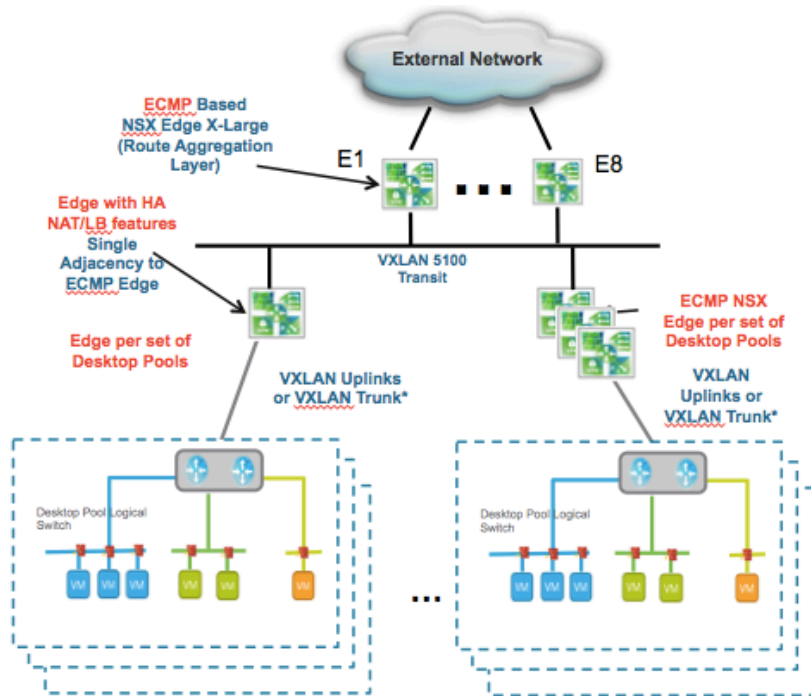


Figure 8: Scalable Topology for Deploying Horizon with NSX Overlays

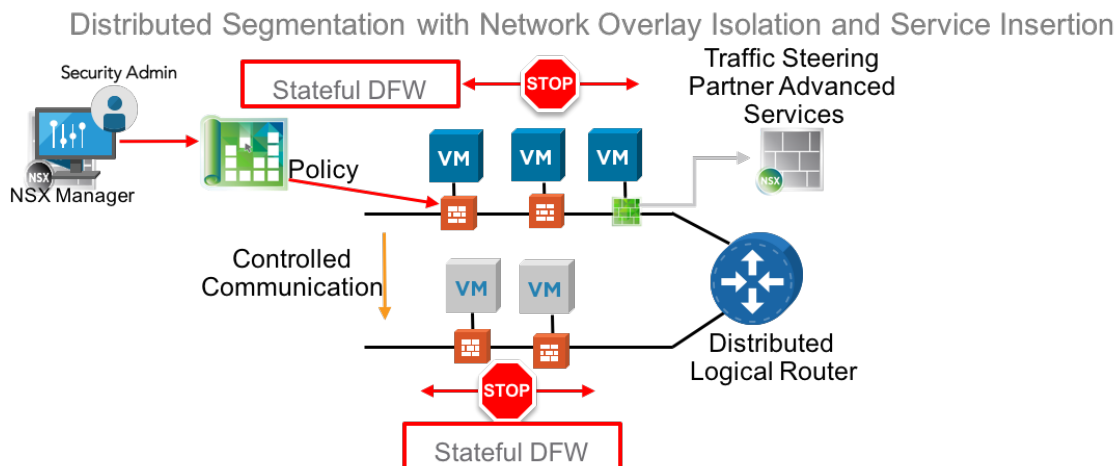
In this model, each desktop pool or a combination of desktop pools can be uplinked to an NSX Edge Service Gateway (ESG) or a set of Equal Cost Mutlipath (ECMP) Edge Service Gateways. The traffic can then be aggregated via VXLAN uplinks to a Tier 0 set of ESGs that is connected to the external network. This model has an added advantage of providing overlapping IP addresses to different desktop pools. Each desktop pool can have similar set of IP Addresses. However, edges at Tier 1 Edge Services Gateway (connected to the desktop pool uplinks) will need to be in HA mode and have requisite NAT configured. This will enable flows from overlapping IP addresses differentiated in the aggregation layer of edges.

## 5 Securing Horizon with NSX Micro-Segmentation

NSX provides a security platform that enables secure Horizon infrastructure as well as secure desktop pools via micro-segmentation. Generally, in a virtual desktop infrastructure, it is very difficult to provide security at a granular level. NSX helps in securing desktop pools as well as limit communication between desktops and unauthorized applications. NSX enables the following capabilities to provide micro-segmentation for end-user computing environments.

- Distributed stateful firewalling for topology agnostic segmentation

- Reducing the attack surface within the datacenter perimeter through distributed stateful firewalling and [ALGs \(Application Level Gateway\)](#) on a per-workload granularity regardless of the underlying L2 network topology (i.e. possible on either VXLAN overlays or VLAN)
- Centralized ubiquitous policy control of distributed services
  - Enabling the ability to programmatically create and provision security policy through a RESTful API or integrated cloud management platform (CMP)
- Granular unit-level controls implemented by high-level policy objects
  - Enabling the ability to utilize security groups for object-based policy application (i.e. security groups can use dynamic constructs such as OS type, VM name or static constructs such active-directory groups, logical switches, VMs, port groups IPsets, etc.). See the [DFW Policy Rules Whitepaper](#) for more information.
- Network overlay based isolation and segmentation
  - Logical Network based isolation and segmentation that can span across racks or data-centers regardless of underlying network hardware, centrally managed multi-datacenter security policy
- Policy driven unit-level service insertion and traffic steering
  - Enabling Integration with 3<sup>rd</sup> party solutions for advanced IDS/IPS and guest introspection capabilities



NSX has two built-in services; distributed firewall and identity-based firewall that helps provide granular access at the data plane layer. It also provides the frameworks for guest introspection and network introspection that enables NSX automated agentless anti-virus, next-generation firewall and intrusion-detection system/intrusion-prevention systems from eco-system partners for advanced Layer 7 application capabilities. .

NSX also provides a policy layer known as Service Composer that enables a centralized management plane to orchestrate distributed services. Table 2 provides a table of capabilities.




	Component	Description
	<b>Service Composer</b>	Within NSX Manager, built-in tool that allows provisioning and assignment of firewall policies and security services to applications
	<b>Firewall Services</b>	Virtual, distributed firewall that manages and monitors network traffic based on VM role or user identity.
	<b>Guest and Network Introspection Frameworks</b>	Deployed as part of host preparation for NSX. Enables agentless AV as well as NGFW, IDS, IPS and may other endpoint based and network based services.

Table 2: NSX Platform features to provide micro-segmentation

There are three different aspects of a Horizon end-user computing environment that NSX helps secure via micro-segmentation. Figure 9 below illustrates the various areas that NSX can help secure in a Horizon virtual desktop deployment.

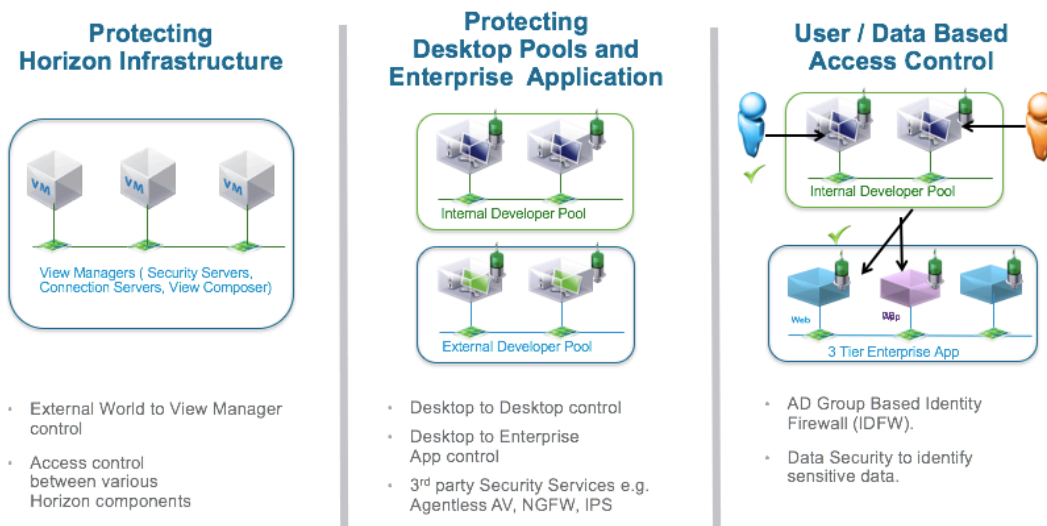


Figure 9: Securing Horizon virtual desktop Deployment with NSX Micro-Segmentation

Horizon virtual desktop deployment can be divided into 3 major areas that are crucial to protect.

1. Horizon virtual desktop Infrastructure: Horizon infrastructure contains various components whose interaction allows administrators to provision desktops and the users to connect to the desktops. NSX protects the horizon infrastructure allowing secure inter-communication between horizon components.
2. Interaction between virtual desktops and enterprise applications: Virtual desktops contain applications that allow users to connect to various enterprise applications inside the datacenter. NSX secures and provides limited access to applications inside the datacenter. This enables fine grained control to the datacenter from each and every desktop.
3. User and Data Based access control: NSX allows user identity based micro-segmentation for the Horizon desktops. This enables creating a fine grained access control and visibility for each desktop based on the user accessing the desktop.

Figure 10 provides the complete micro-segmentation picture for the entire Horizon virtual desktop deployment. This includes deploying all aspects of the Horizon Infrastructure.

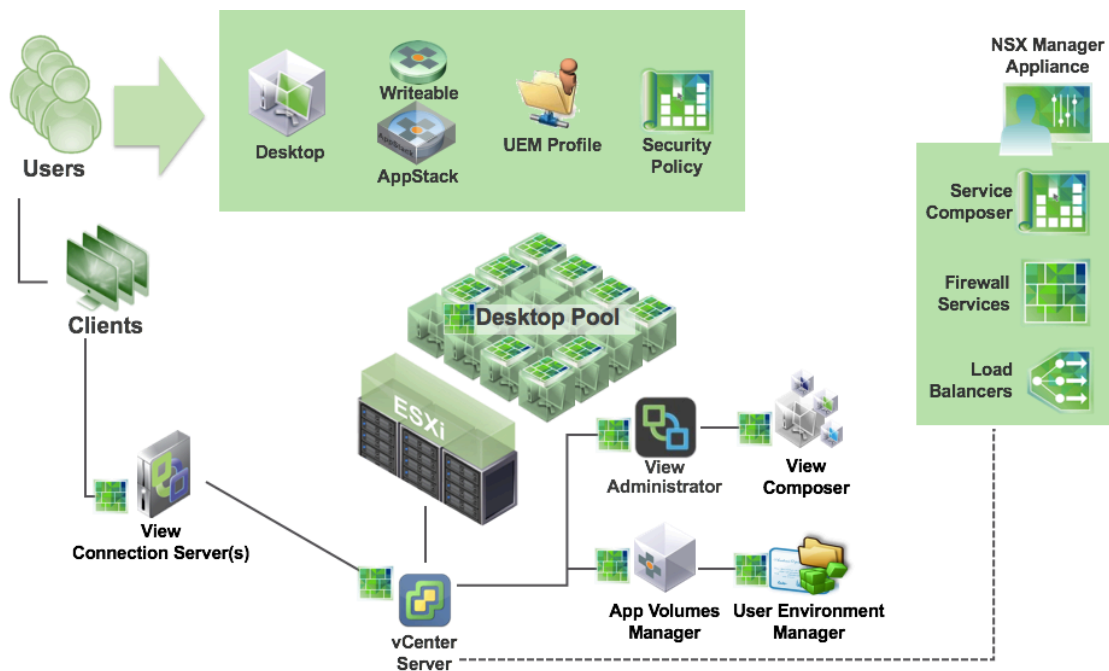




Figure 10: Securing Horizon virtual desktop Deployment with NSX Micro-Segmentation

## 5.1 Creating Micro-segments with NSX for Horizon end-user computing

Micro-segmentation is comprised of three foundational principals; isolation, segmentation and service insertion.

### 5.1.1 Network Isolation

Isolation is the foundation of most network security, whether for compliance, containment or simply keeping development, test and production environments from interacting. While manually configured and maintained routing, ACLs and/or firewall rules on physical devices have traditionally been used to establish and enforce isolation and multi-tenancy, those properties are inherent to network virtualization.

Virtual networks (leveraging VXLAN technology) are isolated from any other virtual network and from the underlying physical infrastructure by default, delivering the security principle of least privilege. Virtual networks are created in isolation and remain isolated unless specifically connected together. No physical subnets, no VLANs, no ACLs, no firewall rules are required to enable this isolation.

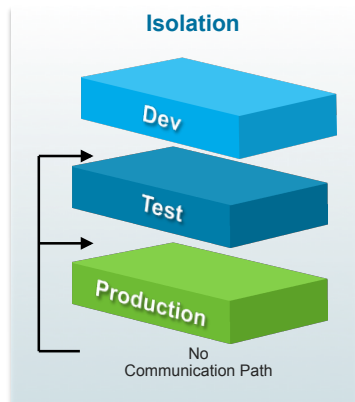


Figure 3 – Network Isolation

Any isolated virtual network can be made up of workloads distributed anywhere in the data center. Workloads in the same virtual network can reside on the same or separate hypervisors. Additionally, workloads in several multiple isolated virtual networks can reside on the same hypervisor. Case in point isolation between virtual networks allows for overlapping IP addresses, making it possible to have isolated development, test and production virtual networks, each with different application versions, but with the same IP addresses, all operating at the same time and on the same underlying physical infrastructure.



Virtual networks are also isolated from the underlying physical network. Because traffic between hypervisors is encapsulated, physical network devices operate in a completely different address space than the workloads connected to the virtual networks. For example, a virtual network could support IPv6 application workloads on top of an IPv4 physical network. This isolation protects the underlying physical infrastructure from any possible attack initiated by workloads in any virtual network. Again, independent from any VLANs, ACLs, or firewall rules that would traditionally be required to create this isolation.

### 5.1.2 Network Segmentation

Related to isolation, but applied within a multi-tier virtual network, is segmentation. Traditionally, network segmentation is a function of a physical firewall or router, designed to allow or deny traffic between network segments or tiers. For example, segmenting traffic between a Web tier, Application tier and Database tier. Traditional processes for defining and configuring segmentation are time consuming and highly prone to human error, resulting in a large percentage of security breaches. Implementation requires deep and specific expertise in device configuration syntax, network addressing, application ports and protocols.

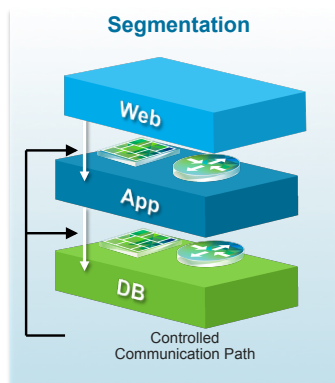


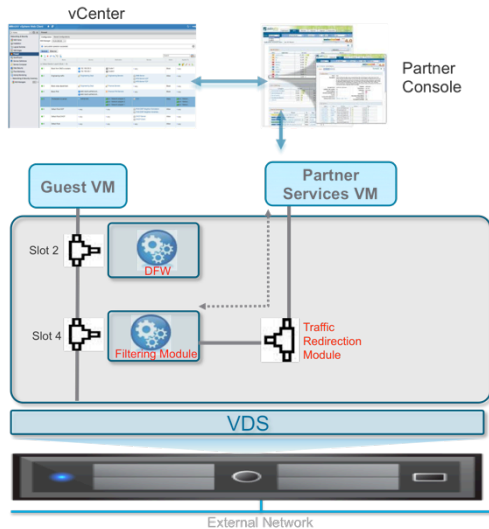
Figure 4 – Network Segmentation

Network segmentation, like isolation, is a core capability of VMware NSX network virtualization. A virtual network can support a multi-tier network environment, meaning multiple L2 segments with L3 segmentation or a single-tier network environment where workloads are all connected to a single L2 segment using distributed firewall rules. Both scenarios achieve the same goal of micro-segmenting the virtual network to offer workload-to-workload traffic protection (also referred to as east-west protection).

### 5.1.3 Advanced Services

NSX as a security platform provides L2-L4 stateful firewalling features to deliver segmentation within virtual networks. In some environments, there is a

requirement for more advanced network security capabilities. In these instances, customers can leverage VMware NSX to distribute, enable and enforce advanced network security services in a virtualized network environment. NSX distributes network services into the vNIC context to form a logical pipeline of services applied to virtual network traffic. Third party network services can be inserted into this logical pipeline, allowing physical or virtual services to be consumed in the logical pipeline.



**Figure 5 – Service Insertion, Chaining and Steering**

Between guest VM and logical network (Logical Switch or DVS port-group VLAN-backed), there is a service space implemented into the vNIC context. Slot-ID materializes service connectivity to the VM. As depicted in the above **Error! Reference source not found.**, slot 2 is allocated to DFW, slot 4 to the Palo Alto Networks VM-series FW. Another set of slots is available to plug more third-party services.

Traffic exiting the guest VM always follows the path with increasing slot-ID number (i.e. packet is redirected to slot 2 and then slot 4). Traffic reaching the guest VM follows the path in the reverse slot-ID order (slot 4 and then slot 2).

Every security team uses a unique combination of network security products to meet the needs of their environment. The VMware NSX platform is being leveraged by VMware's entire ecosystem of security solution providers. Network security teams are often challenged to coordinate network security services from multiple vendors in relationship to each other. Another powerful benefit of the NSX approach is its ability to build policies that leverage NSX service insertion, chaining and steering to drive service execution in the logical services pipeline, based on the result of other services, making it possible to coordinate otherwise completely unrelated network security services from multiple vendors.

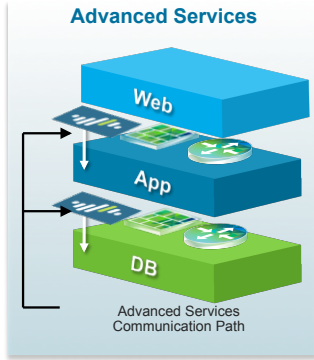


Figure 6 – Network Segmentation with Advanced Services provided by third party vendor.

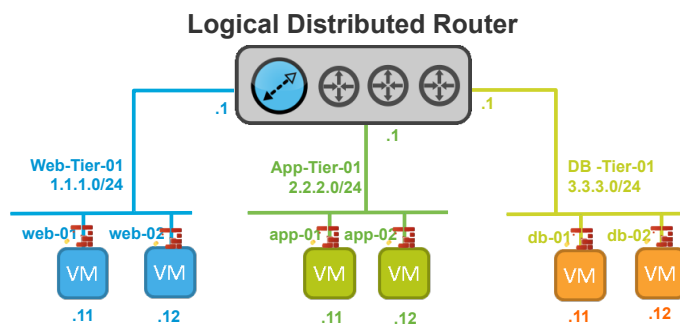


Figure 7 - Micro-segmentation with NSX DFW – Multiple Layer-2 Segment

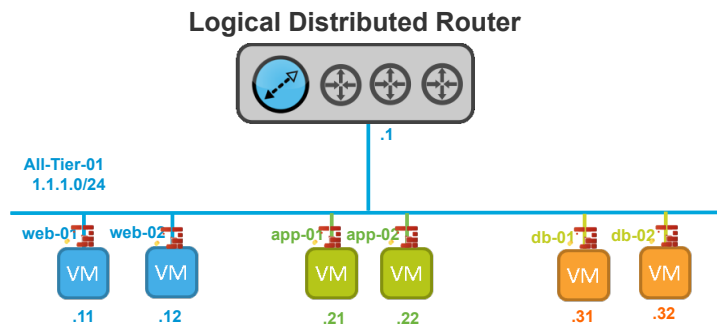


Figure 8 - Micro-segmentation with NSX DFW – Single Layer-2 Segment

## 5.2 Introduction to Service Composer

NSX introduces a way of deploying security services that are independent of the underlying topology. Services like firewall, or advanced services like agentless AV, L7 Firewall, IPS and monitoring traffic can be deployed independent of the underlying (physical or logical) networking topologies. This enables a significant shift in planning and deploying services in the datacenter. Services no longer need to be tied down to networking topology. NSX provides a framework called Service Composer to enable deployment of security services for the datacenter.

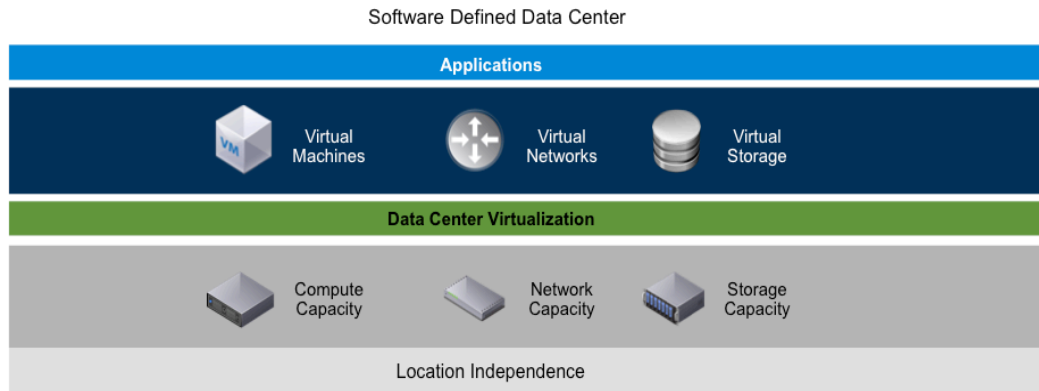


Figure 9 – Software Defined Data Center

Service Composer contains three broad parts to it. They are:

1. Intelligent Grouping via Security Groups: NSX achieves decoupling of workloads from the underlying topology via creation of these security groups.
2. Security Policies: Security Policies enable NSX to apply a consolidated way of creating and providing rules to protect the workloads. Policies include governing of not just the built-in NSX security technologies but also third party vendor services.
3. 3<sup>rd</sup> Party Service Registration and Deployment: It enables 3<sup>rd</sup> party vendor registration with NSX and deployment of vendor security technologies throughout the datacenter.

Service Composer Model is a very simple model as represented in the figure below:

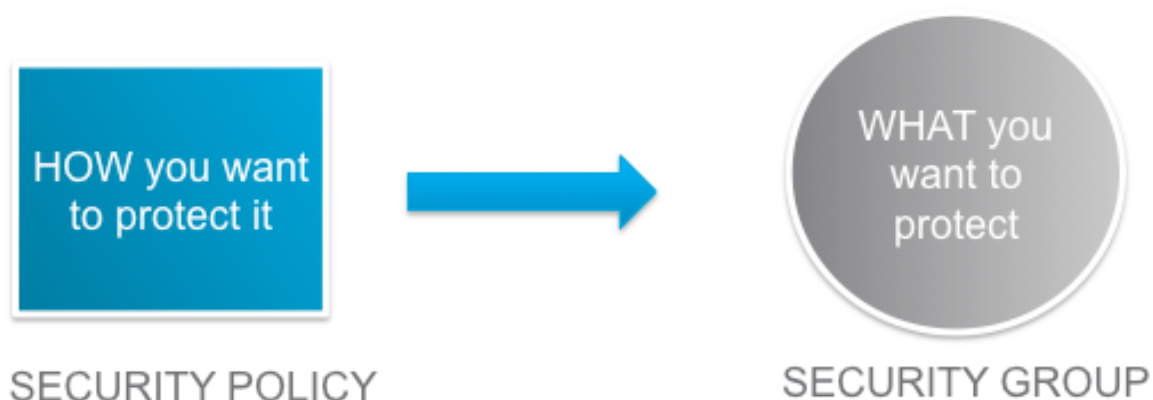


Figure 10 – Decoupling of Rules and Policy

There are various advantages in decoupling the service and rule creation from the underlying topologies:

1. *Distribution of Services*

The services layer of NSX allows distributing and embedding services across your datacenter. This allows workloads to move across the datacenter without creating bottlenecks or hair pinning of traffic. However, granular traffic inspection is done for all workloads wherever they are residing in the datacenter.

2. *Policies are Workload-Centric*

Policies can now be truly workload centric instead of translating from workloads to virtual machines to their basic networking topology and IP address constructs. Policies can be now configured to say that a group of database workloads will be allowed these specific operations without explicitly calling out networking centric language like IP subnets, MAC and Ports.

3. *Truly Agile and Adaptive Security Controls*

With the above #1 and #2, the datacenter security capability is now truly agile and flexible. Workloads do not have to be designed based on the underlying physical networking topologies. Logical networking topologies can be created at scale on demand and provisioned with security controls that are independent of these topologies. When workloads migrate from one place in the datacenter to a different place, security controls and policies migrate with them. Additional workloads do not require provisioning same kind of security as the policies automatically adapt.

4. *Service Chaining is Policy-based and Vendor Independent*

With NSX, service chaining is based on a policy across various security controls. Service chaining has evolved from manually hardwiring various security controls from one or more vendors in the underlying network. With NSX, policies can be created, modified and deleted based on the requirement for service chaining.

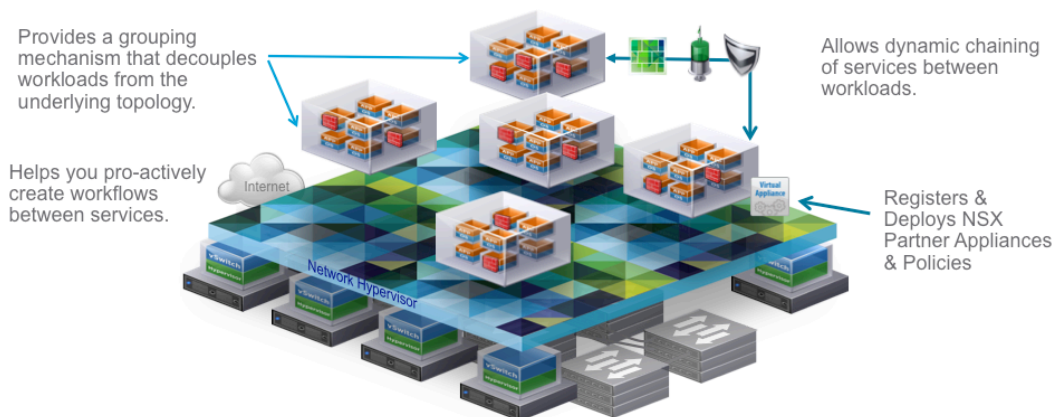


Figure 11 – Advantages of Service Composer

### 5.3 Security Groups and Policy

To implement micro-segmentation in an end-user computing environment, NSX provides various ways of grouping VMs and applying security policies to the groups.

- **Traditional network-based policies**
  - Vlan or IPset maps to security grouping (e.g. VLAN 100 = VoIP or 10.10.10.0/26 assigned to engineering group)
  - *Best suited for static environments or specific use cases. Is coupled to physical network constructs*
- **Infrastructure-based policies**
  - Based on software-defined infrastructure constructs like clusters, logical switches, distributed port groups etc
  - *Best suited if there are clearly defined physical or logical boundaries in your SDDC where mobility of the workloads across said boundaries won't jeopardize your security posture*
    - (e.g. common View pod/cluster dedicated to PCI and another cluster for desktops used by the rest of the company)
- **Application-based policies**
  - Based on the application type (e.g: VMs tagged as “Web\_Servers”), application environment (e.g: all resources tagged as “Production\_Zone”) and application security posture

Not tied to either network constructs or SDDC infrastructure. Security policies can move with the application irrespective of network or infrastructure boundaries. Most flexible implementation of micro-segmentation

### 5.3.1 Intelligent Grouping

Intelligent Grouping in NSX can be created in many ways. There are many customized grouping criteria possible as show in the figure below:

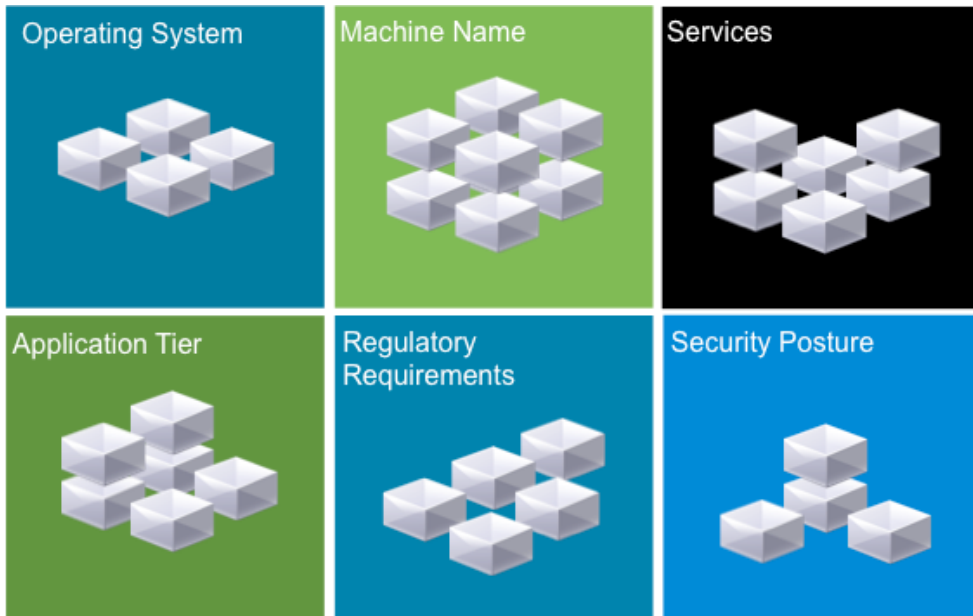


Figure 12 – Types of intelligent Grouping

NSX provides grouping mechanism criteria can be based on vCenter Objects (e.g. VMs, Distributed Switches, Clusters, etc.), VM Properties (e.g. vNICs, VM names, VM operating Systems, etc.) or NSX Objects (e.g. Logical Switches, Security Tags, Logical Routers, etc.)

Grouping mechanisms can be either static or dynamic in nature. Grouping mechanisms use either vCenter objects (e.g. VM, Cluster, vAPP, DVS, vNIC), or NSX Objects (Logical Switch, Security Tags, IP Sets, MacSets) or VM Properties (VM Name, VM Operating System) or Identity Manager objects (AD Groups). A complete list of objects that can be used is listed in the figure below.

A group can be any combination of objects. A security group in NSX is based on all the inclusion (static & dynamic) criteria and static exclusion criteria defined by a user. Figure below details the security group construct.

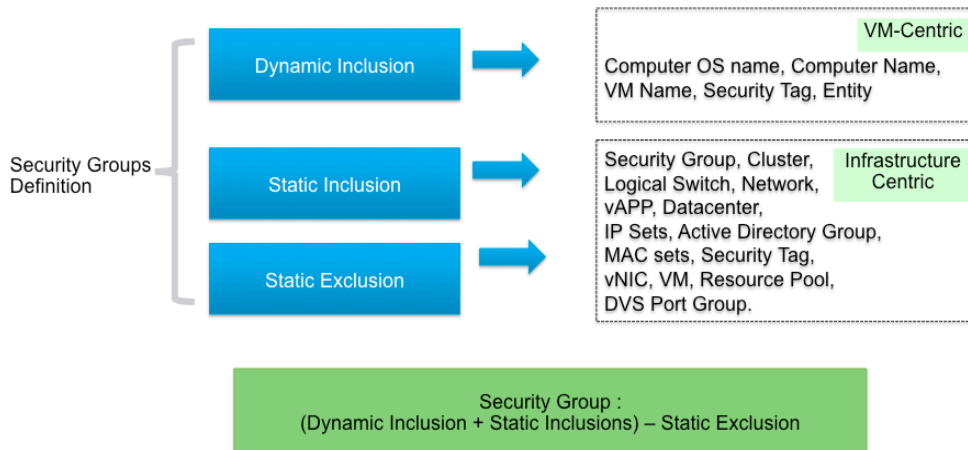


Figure 13 – Scope of Security of Group Attributes

Static grouping mechanisms are based on collection of virtual machines that conform to set criteria. vCenter Objects defines datacenter components and NSX objects defines core networking components are used in static grouping criteria. Combining different objects for the grouping criteria results in the creation of the “AND” expression in the NSX System.

Dynamic grouping mechanisms are more flexible and are characterized by allowing expressions that evaluates to defining the virtual machines for a group. The core difference is the ability to define “AND/OR” or “ANY/ALL” criteria for the grouping.

Evaluation of VMs that are part of a group is also different. In a static grouping mechanism, the criteria tells the NSX Manager, which objects to look for in terms of change. In dynamic grouping, NSX manager evaluates each and every change in the datacenter environment to determine if this affects each of the group.

### 5.3.2 Security Tags

NSX provides Security Tags that can be applied to any virtual machine. This allows for classification of virtual machines in any desirable form.

Security Group	Dynamic Membership
SG-FIN-WEB	Security Tag contains 'FIN-TAG-WEB'
SG-HR-WEB	Security Tag contains 'HR-TAG-WEB'

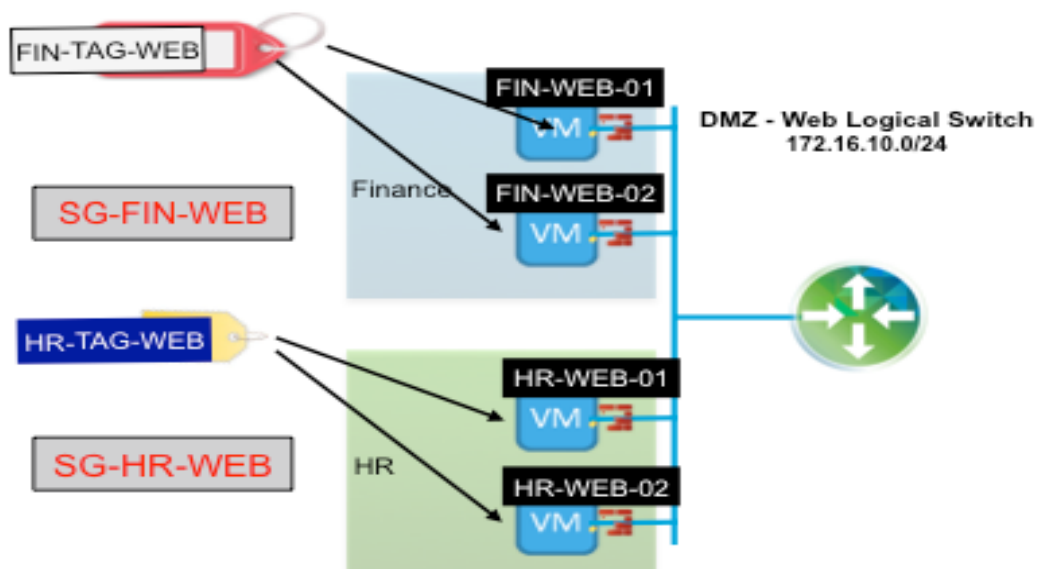




Figure 14 – Tag Used as Base Isolation Method

Some of the most common forms of classification for using security tags are:

1. Classification for departments
2. Data-Type classification (e.g. PCI Data).
3. Type of environment (Production, DevOps)
4. Geo-Location of the virtual machine.

Security Tags are intended for providing more contextual information about the workload. This allows for better security posture for the workload with this additional context. In addition to users creating and defining their own tags, third party security vendors can use the same tag for advance workload actions. In addition vendors set tags to indicate trigger the flag on malware found, workload is under threat based on IPS detection or update the CVSS scores. This allows for context sharing across different vendors.

### 1.1.1 Introduction to Security Policy

NSX provides security policy that contains rules for security controls that will be applied to an intelligent group created in your datacenter. So if we expand the service composer model:



Figure 15 – What goes into Security Policy and Security Group

A security policy is comprised of two broad parts. They are:

1. Services: The security controls that will be provisioned for a policy. Examples of services are Firewall, Antivirus, Vulnerability management, IPS.
2. Profiles: Security vendors publish policies back to the NSX platform. NSX defines them as service profiles for a particular service.

Security Policies can be written in multiple ways. They are

1. Traditional Method: In this method, the rules and policies written in the traditional firewall table format. This is available both for built-in distributed firewall as well as third party advanced services for network introspection services like IPS, traffic monitoring etc.
2. NSX Policy Method: The Service Composer method provides a construct called security policy that enables you to create rules and controls in a

new way. These policies can be created as a template and then applied to as many security groups possible.

In both these methods, NSX consumes all the intelligent groupings discussed above. The firewall rule table method enables users to create rules in traditional ways of just IP addresses. This is the major difference between the traditional methods of writing a security rule vs. NSX security policy method. Security Policies only allow security groups as way of creating rules and decouples the rule management and the context.

NSX provides creation of sections in a firewall rule table. Sections allow better management and grouping of firewall rules. A single security policy is essentially a section in a firewall rule table. This is maintained so that rules in a firewall rule table and those written via the security policy are always in sync and understand each other.

Since a security policy is written for a particular application or workload, its rules are organized into a section in a firewall rule table. Multiple security policies can be applied to a single application. In that scenario, the order of the sections is important to determine that precedence of rules that will be applied.

### 1.1.2 Anatomy of a Security Policy

In this section we will explain in detail about a security policy. Figure below contains an anatomy of a policy.

Rules	Weights	Inheritance
<ul style="list-style-type: none"><li>• Rules for various services available in the platform.</li><li>• Rules have precedence.</li></ul>	<ul style="list-style-type: none"><li>• Determines the policy that needs to be applied first.</li><li>• Rank and Weight of the policy are the same.</li></ul>	<ul style="list-style-type: none"><li>• If it contains rules from a parent policy.</li><li>• Parent and Child policy weights are automatically adjusted.</li></ul>

Figure 16 –Anatomy Security Policy

A security policy contains the following:

#### 1. **Rules of a policy**

Each policy contains a set of rules that define the security controls behavior. A single policy may contain one or more security controls as shown in the figure below. There are three types of security control namely,

- a. Firewall: NSX built in distributed firewall
- b. Guest Introspection Control: These are VM based controls like Anti-virus, Anti-Malware, File Integrity Monitoring, Data Security etc. controls.
- c. Network Introspection Controls: L7 Firewall, IDS/IPS

Vendors integrate with NSX using the Guest Introspection framework and/or the Network Introspection framework to provide the requisite controls for those technologies. Any policy created on the vendor platform is published on the NSX platform. These policies are termed as service profiles in NSX.

For each of the rule that contains a security control from a vendor, a service profile has to be associated with the rule.

Rules within a given security control can have precedence. Rules will be executed in a top-down order.

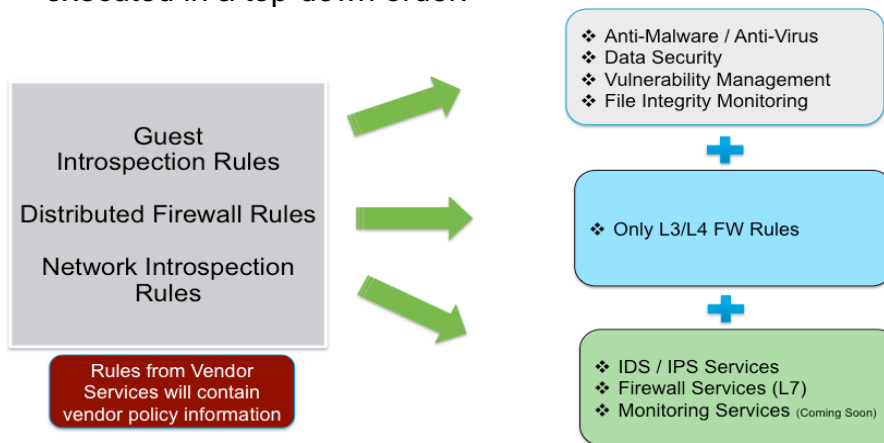


Figure 17 – Types of Rules and Properties

## 2. Weight of the Policy

A given security group can have multiple security policies applied to it. Weight of a policy provides NSX with the information about the order of rules to be applied. Weights of a policy determine the rank of the policy vis-à-vis other policies in the NSX eco-system.

## 3. Inheritance

A policy can be inherited from multiple security policies. NSX provides a mechanism to create base policies and child policies. Applying a child policy to a security group automatically applies the base/parent policies to the security group.

Security policies can be inherited from base policies and added more rules. A child security policy will contain all the rules in the order created in the base policy and all the other child rules will be added

after that. Applying a child policy to a security group will automatically apply the parent policy.

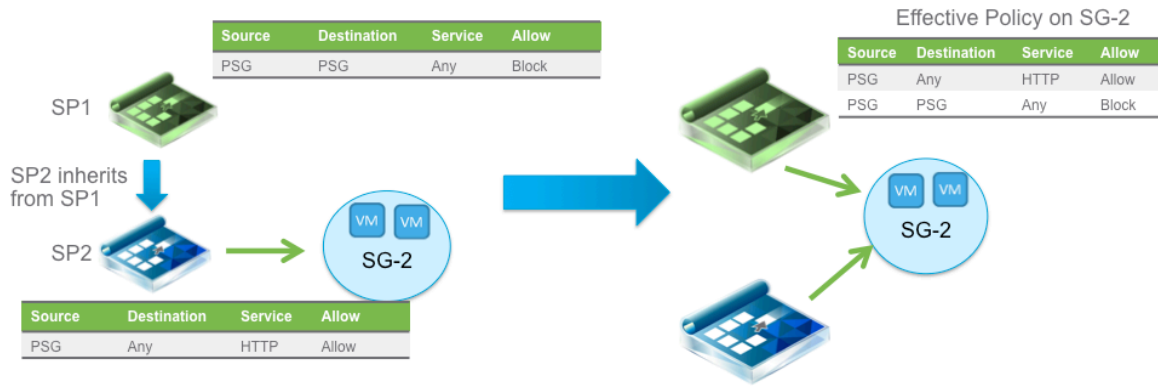


Figure 18 – Rule Inheritance

## 5.4 Deploying distributed firewall to protect Horizon infrastructure

The vSphere cluster must have been prepared by NSX in order to activate DFW, VXLAN and Distributed Logical Router modules within the ESXi kernel. VTEP configuration is needed to utilize VXLAN logical switches:

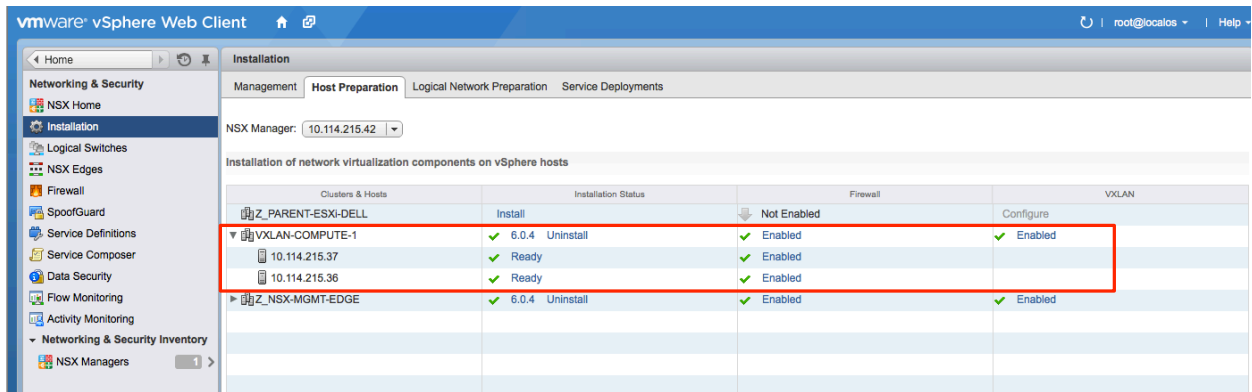


Figure 19 – Host Preparation Window.

- NSX Controllers must be deployed to support VXLAN Logical Switches and Distributed Logical Router:

The screenshot shows the VMware vSphere Web Client interface. The left sidebar contains a navigation menu with categories like 'Networking & Security', 'NSX Home', 'Installation', 'Logical Switches', 'NSX Edges', 'Firewall', 'SpoofGuard', 'Service Definitions', 'Service Composer', 'Data Security', 'Flow Monitoring', 'Activity Monitoring', and 'Networking & Security Inventory'. The main content area is titled 'Installation' and has tabs for 'Management', 'Host Preparation', 'Logical Network Preparation', and 'Service Deployments'. Under the 'Management' tab, there are two tables:

**NSX Manager**

NSX Manager	IP Address	vCenter	Version
10.114.215.42	10.114.215.42	10.114.215.35	6.0.4.1752381

**NSX Controller nodes**

Name	Node	NSX Manager	Cluster/Resource Pool	Datastore	Host	Software Version	Status
controller-10	10.114.215.61	10.114.215.42	Resources	Datastore-NFS-UBUN...	10.114.215.38	6.0	✓ Normal
controller-11	10.114.215.62	10.114.215.42	Resources	Datastore-NFS-UBUN...	10.114.215.38	6.0	✓ Normal
controller-12	10.114.215.59	10.114.215.42	Resources	Datastore-NFS-UBUN...	10.114.215.38	6.0	✓ Normal

Figure 20 – NSX Controllers.

Note: VMTools must be installed on all guest VMs to support Identify Based Firewall

## 5.5 Securing desktop pools using distributed firewall

The VMware NSX platform includes two firewall functionalities: a centralized firewall service offered by the NSX Edge Services Gateway and a Distributed Firewall (DFW) enabled in the kernel as a VIB package (similarly to the distributed routing function previously described) on all the ESXI hosts part of a given NSX domain. The DFW provides firewalling with near line rate performance, virtualization and identity aware with activity monitoring, among the isolation and segmentation network security features native to network virtualization.

## 5.6 Identity based micro-segmentation for desktop pools

DFW offers Identity Based Firewall (IDFW) functionalities:

- Specific group of users can access particular group of applications
- Policy rules defined based on Active Directory (AD) group that user belongs to
- Define a Security Group that contains AD group and apply it as source of the DFW policy rule
- User can use any desktop or client but the destination MUST be a VM... like a virtual desktop
- Activity Monitoring provides monitoring of user activity (which application has been accessed for instance, top users by permitted/denied traffic, top applications, ...)

## 6 Deploying NSX Load-Balancing for Horizon

Load-balancing Horizon with NSX utilizes the NSX Edge Services Gateway (ESG)

All configurations in this document can be performed on VLAN or VXLAN connected systems. VMware NSX components can be consumed in whole or in part, allowing customers to deploy the services required for their specific environments. VMware recommends that in addition to NSX Load Balancing customers deploy NSX distributed firewall micro-segmentation for VDI. By using these components customers will have both a highly available, redundant and secure environment.

This document provides guidance on how to deploy VMware Horizon (Connection Servers, Access Point Servers or Security Servers) and enable load balancing using VMware NSX for vSphere.

### 6.1 Building The Infrastructure

In this section we will build the basic infrastructure that will be leveraged later. We will show you how to create a profile, pool, virtual server and monitor. You will need to repeat these steps based on the charts in the respective View topology sections of this document.

#### 6.1.1 Deciding the View Topology

In allowing access from external clients we need to determine the topology we will use for the gateway. Horizon uses Security Servers or Access Point Servers for external client access. If this is a new Horizon 7 deployment VMware recommends the deployment of Access Point Servers for external client access.

Internal clients normally leverage the Connection Servers and these systems normally have a different application flow as compared to external users that are using Access Point Servers or Security Servers.

When designing the load balancing solution one must start with the Horizon configuration and then build a load balancing solution to meet the requirements. Each customer will have variances in their individual requirements.

VMware Horizon supports not only multiple access topologies and systems, but you can also choose different protocols, such as Blast Extreme, PCoIP, and RDP. When selecting these protocols there are implications to the configuration size and how load balancing with persistence is maintained.

## 6.2 Certificates

We are not leveraging SSL offload in this deployment. To prevent your end users from reporting certificate errors you will need to deploy the same SSL certificate to each of your Connection Servers, Security Servers or Access Point Servers. You can choose to leverage SAN or Wildcard certificates on your View servers if you so choose.

## 6.3 User Types

A deployment will have different types of users. There will be users that are inside the corporate firewall, outside the firewall, standard users, and 3D users. The combination of user type, access method, and concurrency will create the final design for accessing the EUC environment. When designing a load balancing solution for your environment you will need to account for the aggregate bandwidth of the concurrent users, as well as the aggregate connection counts that will be generated. See the section on Considerations for deployment.

## 6.4 Access Point

VMware Access Points are hardened Linux virtual machines that accept client connections. Access Points support multiple modes of client connections; there is the Standard (Tunnel) Mode, Blast Extreme, Blast Extreme Port Sharing and Direct mode. We will provide load balancing configuration guidance for each.

### 6.4.1 Access Point Standard protocol topology – tunneled

In this mode the client systems use the following ports and protocols – TCP 4172, UDP 4172, TCP 443. The client systems authenticate on TCP 443, and then use PCoIP (TCP/UDP 4172). This mode is very similar to configuring the Connection Server in Tunnel Mode or Security Server Tunnel Mode load balancing, and requires more ports and protocols to be configured to it is a more complex load-balancing configuration.

### 6.4.2 Access Point Blast Extreme protocol topology - tunneled

In this mode the Access points proxy all the client connections over TCP using TCP 443 and TCP 8443. This mode provides load balancing of the Blast Extreme protocol, and only leverages two ports, and one protocol, and is a simpler load-balancing configuration.

### 6.4.3 Access Point Blast Extreme Mode – Port Sharing protocol topology - tunneled

In this mode the Access Points function as a reverse proxy listening on a single port (usually 443) for the client connections. This mode is commonly used because many organizations have outgoing firewall rules that may block Ports 4172, and 8443. This mode provides the simplest load-balancing configuration.

#### 6.4.4 Access Point protocol topology - bypass

In bypass mode the users authenticate to the VIP but all PCOIP and BLAST EXTREME traffic bypasses the Load Balancing function and goes direct to an Access Point Server. Each Access Point server is configured to respond to the client with its unique information. In this topology you will only need to load balance services on port TCP:443

### 6.5 Security Server

Security Servers require a Windows Server to host the application. With these servers we can load balance connections between the Clients and the Security Servers, and the Security Servers have a one to one relationship with the Connection Servers. We will need to configure load balancing for TCP 4172, 443, and UDP 4172 between the Clients and the Security Servers.

#### 6.5.1 Security Server protocol topology - tunneled

In this mode the client systems use the following ports and protocols – TCP 4172, UDP 4172, TCP 443. The client systems authenticate on TCP 443, and then use PCoIP (TCP/UDP 4172). This mode is very similar to configuring the Connection Server in Tunnel mode.

#### 6.5.2 Security Server protocol topology - bypass

In bypass mode the users authenticate to the VIP but all PCOIP, RDP, or BLAST EXTREME traffic bypasses the Load Balancing function and goes direct to a Security Server. Each Security Server is configured to respond to the client with its unique information. In this topology you will only need to load balance services on port TCP:443

### 6.6 Connection Servers

Connection servers are normally used for internal (clients have direct routing to the desktop servers) access, and are normally deployed in Direct mode. You can elect to deploy connection servers in Tunnel mode if you desire. NSX-V load balancing supports both of these topologies.



### 6.6.1 Connection Servers protocol topology - tunneled

In this mode the client systems use the following ports and protocols – TCP 4172, UDP 4172, TCP 443. The client systems authenticate on TCP 443, and then use PCoIP (TCP/UDP 4172). This mode is very similar to configuring the Access Point Server and Security Server load balancing, and requires more ports and protocols to be configured. It is a more complex load-balancing configuration.

### 6.6.2 Connection Server protocol topology - bypass

In this mode clients authenticate to the Connection Servers Virtual Server on port 443, and then connect directly to the desktop or application images as required. This model requires that clients have IP connectivity to the servers hosting the desktops but does not generate a high load on the load balancer in front of the connection servers.

### 6.6.3 Connection Server for Access Point Services

When using Access Point servers, they will proxy client requests to the Connection Servers via the Connection Server Virtual Server on port 443. This is a beneficial topology as it adds scalability and redundancy between the Access Point servers and the connection servers.

### 6.6.4 Connection Server for Security Server Services

When using Security Servers we do not need to provide load balancing between the Security Servers and the connection servers as they have a 1:1 mapping.

### 6.6.5 Configuring the Access Points, Security Servers, or Connection Servers

Once you have decided the server types and protocols that will be supported you will need to configure the servers to return the proper connection information that clients will receive. Each type of server will have differences in the exact method to configure the following: Authentication URL, PCoIP IP address, Blast Extreme URL. Please see the chart below for guidance:

Server Type	IP_1:443	IP_1:PCoIP	IP_1:BLAST EXTREME	NAT
Access Point	x	x	x	X*
Access Point – Direct Mode	x			X*
Security Server	x	x	x	X*
Security Server Direct Mode	x			X*

Connection Server - Tunnel Mode	x	x	x	X*
Connection Server Direct	x			X*

- See section on Data Center Edge Firewalls

The NAT column is only applicable if the IP address that the virtual server or Connection Servers have is not routable to the client systems; see the next section. If the IP addresses are routable the NAT column is not applicable.

## 6.7 Datacenter Edge Firewalls

If you are using RFC 1918 private address space on your Access Points or Security Servers and the Virtual Server and you need to enable access to these services from the internet without a VPN you will need to configure the required NATs on your datacenter firewall, as well as configure the Access Points to return the correct EXTERNAL information. When you are using the PCOIP gateway it must return an IP address that is reachable by the client systems.

For example, if your Virtual Servers is 192.16.1.1, and your servers are 192.168.2.1 and 192.168.2.2 and your external NAT for the Virtual server is 10.1.1.1 the Access Point / Security Servers will need to return connection information to 10.1.1.1 for BLAST EXTREME or PCOIP servers.

How to make these configuration changes is dependent on the use of Security Servers or Access Point servers and you will need to refer to the relevant documentation.

## 6.8 Considerations for Deployment

Key questions in deploying Horizon with NSX load balancing include:

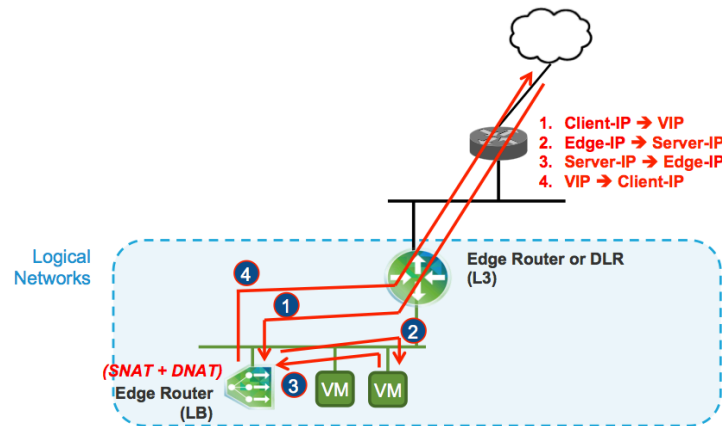
1. How many concurrent sessions will be required?
  - a. Depending on the protocol used each user will consume two to four concurrent connections. We should target 500K or less concurrent connections.
2. What type of users will we have? How many of that type concurrently?
  - a. One NSX LB can scale to 10 Gb/S, but actual throughput may vary depending on packet size and network characteristics.
  - b. Standard users?
  - c. 3D users?
3. What type of access will the users need? What topology?
  - a. Connection Servers (Internal Clients)?
  - b. Access Point?

- c. Security Servers?
- 4. How many Access Point / Security Server deployments will you have?
  - a. Will you have server deployments in each datacenter? One datacenter?
- 5. Certificates
  - a. When load-balancing decisions need to be made on how to leverage SSL. When NOT using SSL offload each Access Point, Connection Server, or Security server must host the same certificate. This certificate and it's Certificate Authority should be trusted by the client. Refer to the relevant documentation.

## 6.9 Network Topology

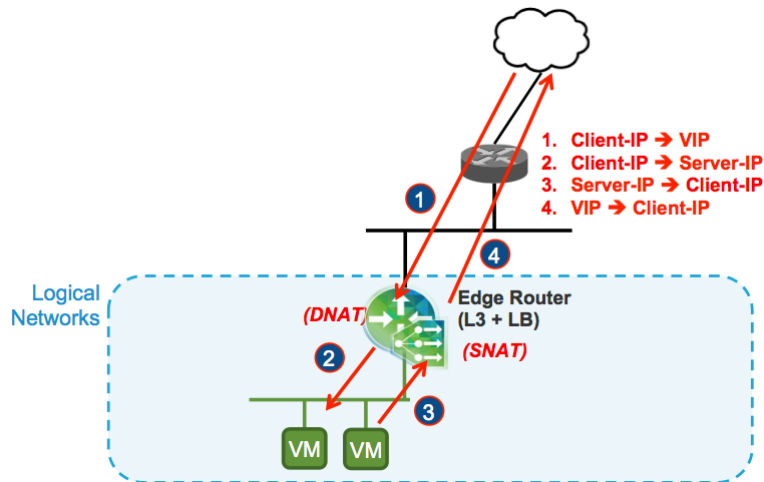
When deploying load-balancing services you have the option to place the load balancing platform inline (AKA two armed mode or transparent mode) or Proxy mode (aka One Armed, or SNAT mode). With NSX Load Balancing you can choose to deploy in either mode. You will need to decide the mode as you build your topology. The notes below will help you make your decision.

### 6.9.1.1 One Arm - aka SNAT Mode



This mode does not require any changes to the server gateways, but you will lose the source IP address of the client. Since changes to the network are minimal it is the easiest to deploy.

### 6.9.1.2 Transparent or Two Armed Mode

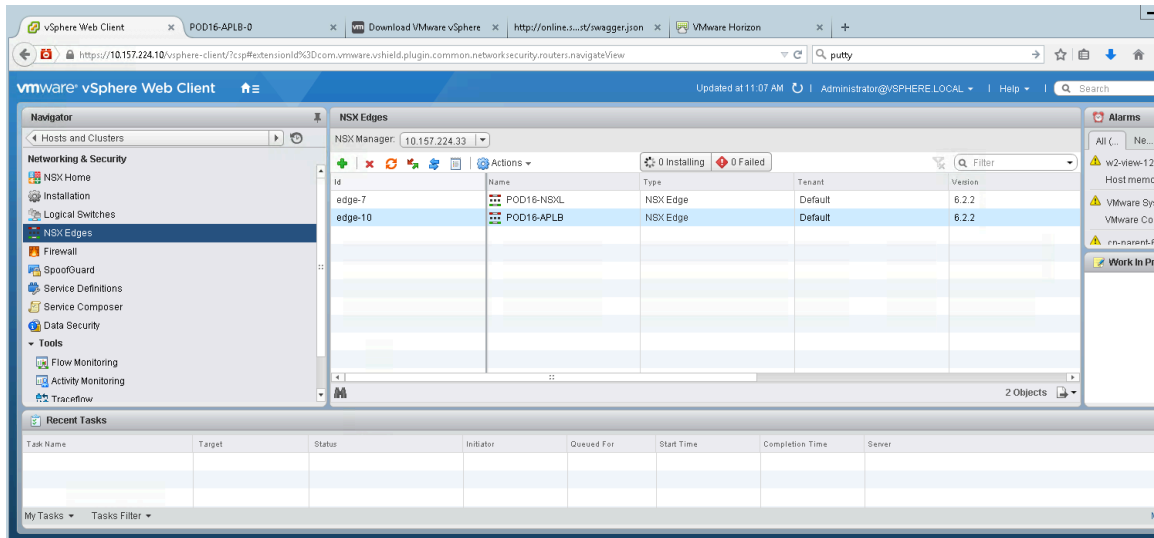


This mode requires that the ESG be the gateway (or in the bidirectional traffic path) between the clients and the servers. It allows retention of the client IP address, but requires the network topology to support it. Please see the Appendix on how to configure pools for transparent topologies.

## 6.10 Edge Services Gateway – Deployment

Prior to configuration of load balancing we will be required to either select an existing Edge Services Gateway (ESG) to perform the load balancing functions or deploy an ESG. NSX differs from the traditional load balancing platforms; with the simple licensing model your organization can now easily chose to build load-balancing systems per application service, or per micro service without incurring any additional license expense. Additionally, NSX does not impose bandwidth limitations on the license supporting bandwidth up to 10 Gb/S.

Log into vCenter, and Navigate to NSX → NSX Edges



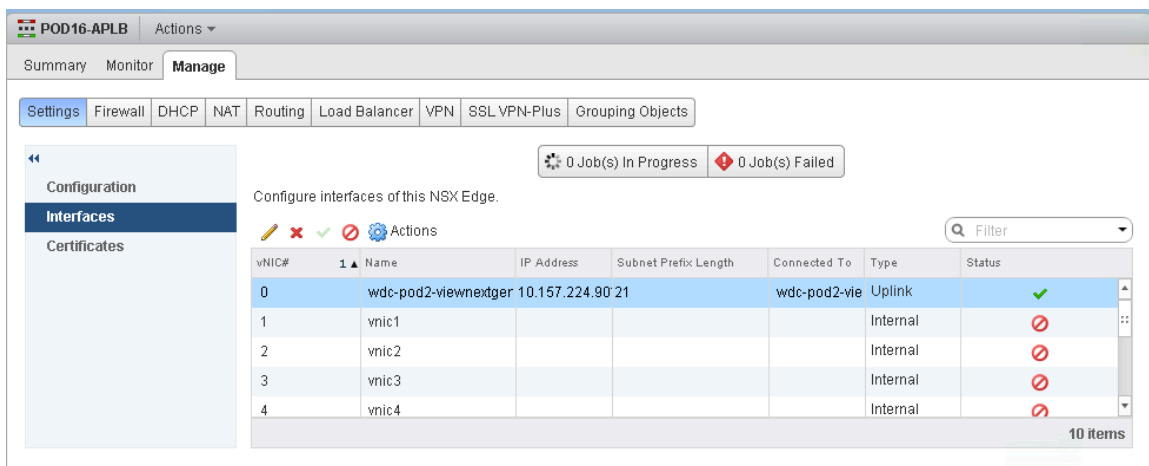
If the ESG you intend to use is already created click on it, and follow the steps in this guide.

If you need to create an ESG click on the green + to create a new edge, and select Edge Service Gateway and follow the steps in the administration guide - <https://pubs.vmware.com/NSX-62/index.jsp#com.vmware.nsx.install.doc/GUID-1EA25D37-F1C7-45C8-AEBA-A555ACC972BC.html>

We will be using an existing ESG in this document, and will be referred to as the desired ESG.

### 6.10.1 Adding IP addresses to be used for Virtual Servers

When creating virtual servers, the IP address that is used must be added to the desired ESG. Navigate to the desired ESG, select manage → settings → interface

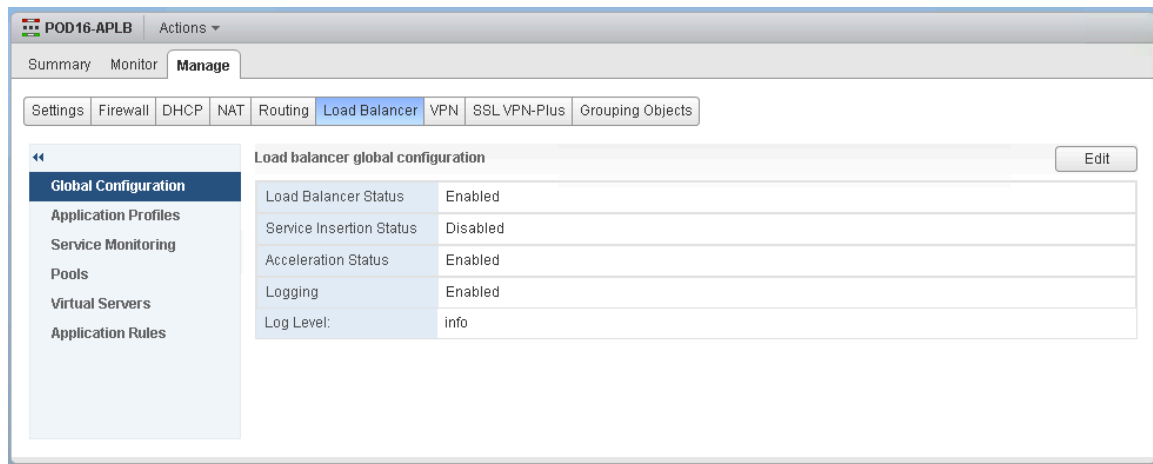


Highlight the interface we want to apply the IP address for the Virtual Server and select the pencil, adding the IP addresses you intend to use as Secondary IP addresses and then click on OK. Additionally, you can choose to use the ESG IP address. VMware recommends that you use a separate IP address for your virtual server so you can security the device from all external access.

### 6.10.2 Enable the ESG for Load Balancing

Go to the desired ESG and navigate to the Load Balancer tab

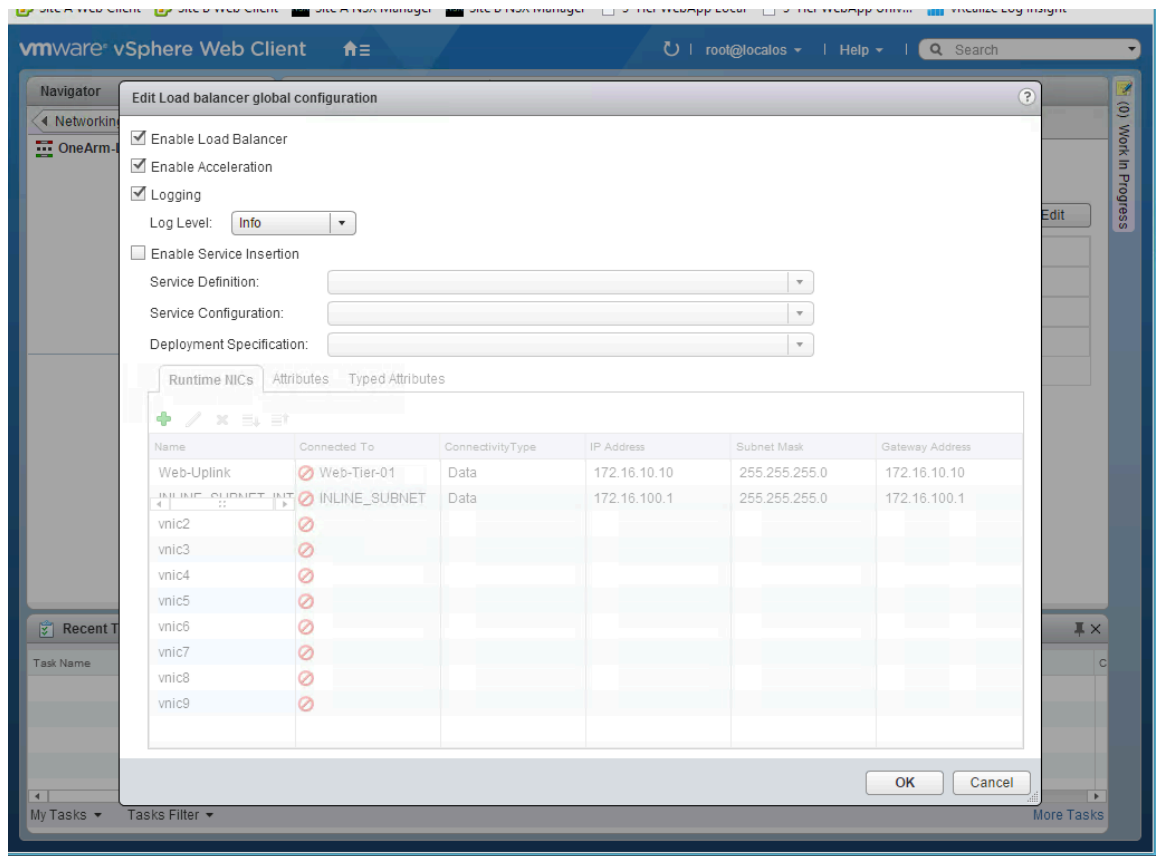
Enable: Load Balancer, Acceleration and Logging, click OK.



The screenshot shows a network management interface for a device named 'POD16-APLB'. The 'Manage' tab is active, and the 'Load Balancer' sub-tab is selected. The 'Load balancer global configuration' section is displayed, showing the following settings:

Setting	Value
Load Balancer Status	Enabled
Service Insertion Status	Disabled
Acceleration Status	Enabled
Logging	Enabled
Log Level:	info

The interface also includes a left-hand navigation menu with options like 'Global Configuration', 'Application Profiles', 'Service Monitoring', 'Pools', 'Virtual Servers', and 'Application Rules'. An 'Edit' button is visible in the top right corner of the configuration area.



## 6.11 NSX-V Load balancing

VMware Horizon uses the following features from the NSX load balancer:

1. Load Balancing
2. Custom Monitors
3. Address Translation (topology dependent)
4. Application Rules

### 6.11.1 How to Create the Different Load Balancing components

When we create Virtual Servers on an Edge Services Gateway we will have the following components in use

- Application Profile aka Profile: this defines the protocol characteristics of our virtual server
- Pool – this the pool of resources that are being load balanced
- Virtual Server – this consists of, or bundles, the other components such as a profile and a pool
- Monitor – a pool may have a standard or custom monitor applied to it.

We will give an example of how to create the relevant components below

### 6.11.1.1 How to create a profile

To create a profile you navigate to the desired ESG → Load Balancing → Application Profiles. Click on the green + sign

**New Profile**

Name:

Type:

Enable SSL Passthrough

HTTP Redirect URL:

Persistence:

Cookie Name:

Mode:

Expires in (Seconds):

Insert X-Forwarded-For HTTP header

Enable Pool Side SSL

Virtual Server Certifica... Pool Certificates

Service Certificates CA Certificates CRL

Configure Service Certificate

Common Name	Issuer	Validity

Cipher:

Client Authentication:

OK Cancel

### 6.11.1.2 How to create a Service Monitor

To create a service monitor navigate to the desired ESG → Load Balancing → Service Monitors and click on the green + sign



**New Service Monitor** ?

Name: \*

Interval:  (seconds)

Timeout:  (seconds)

Max Retries:

Type:  ▼

Expected:

Method:  ▼

URL:

Send:

Receive:

Extension:

OK Cancel

### 6.11.1.3 How to create a Pool

To create a pool navigate to the desired ESG → Load Balancer → Pools and click on the green + sign

**New Pool** ?

Name: \*

Description:

Algorithm:

Algorithm Parameters:

Monitors:

Members:

+ ✎ ✕

Enabled	Name	IP Address / VC Container	Weight	Monitor Port	Port	Max Connections	Min Connection.

Transparent

#### 6.11.1.4 How to create a Virtual Server

To create a virtual server navigate to the desired ESG → Load Balancer → Virtual Servers and click on the green + sign

**New Virtual Server**

**General** | Advanced

Enable Virtual Server

Enable Acceleration

Application Profile: \*

Name: \*

Description:

IP Address: \*  [Select IP Address](#)

Protocol:

Port: \*

Default Pool:

Connection Limit:

Connection Rate Limit:  (CPS)

**OK** **Cancel**

### 6.11.2 Creating the Redirection Virtual Server

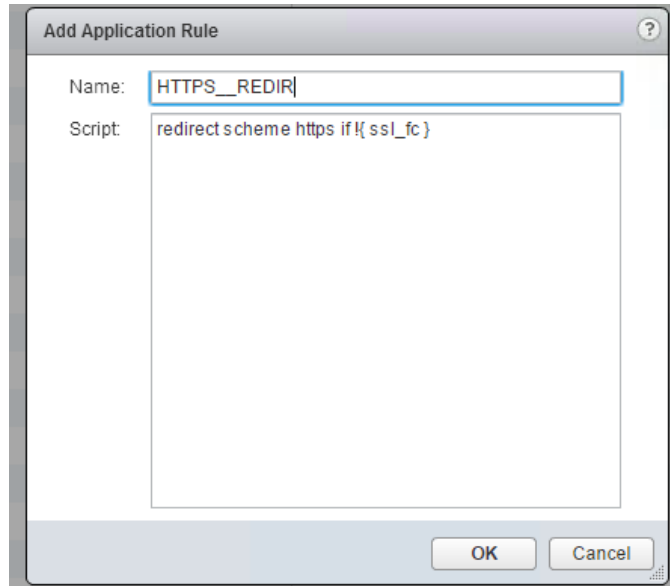
We will deploy a redirection virtual server for Security Server, Connection Server, and Access Point server topologies. With this being a universal item we will show the configuration and then move to the specifics of the other connection systems.

For the redirection virtual server and all the other virtual servers we will need to use the same IP address. We will refer to this address as IP\_ADDRESS\_1 in the document, please make a note of what address you intend on using.

NAME	VALUE
IP_ADDRESS_1	

### 6.11.3 Create the Redirection Application Rule

Navigate to the ESG in question → Load Balancer → Application Rules – click on the green sign



Rule Name	Rule Script	Results
HTTPS__REDIR	Redirect scheme https if ! { ssl_fc }	Redirects all client connections on the applied virtual server to HTTPS

### 6.11.4 Create the Redirection Profile

On the desired ESG navigate to Load balancer → Profile, click on the green + sign

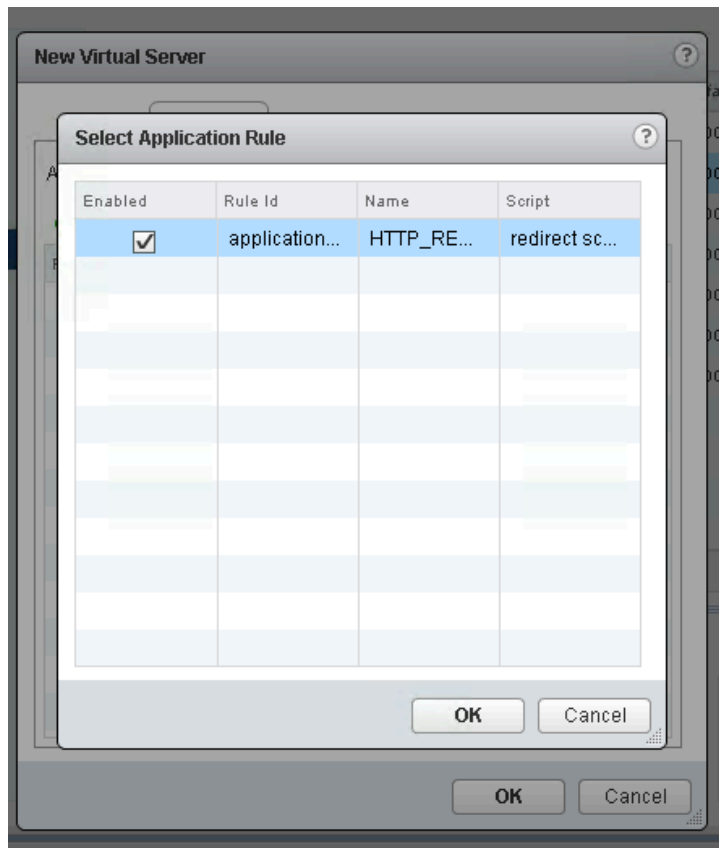
Name	Protocol	Persistence
VIEW__REDIR__PROFILE	HTTP	NONE

### 6.11.5 Create the Redirection Virtual Server

Navigate to the desired ESG → Load Balancer → Virtual Servers. Click on the green + sign

NAME	IP ADDRESS	PORT	PROFILE
VIEW__REDIR__VS	IP_ADDRESS_1	80	VIEW__REDIR__PROFILE

Next select advanced and select the redirection App Rule, and click OK. This will create the redirection virtual server



This will ensure that if a user forgets to type HTTPS:// they will be redirected to the secure site.

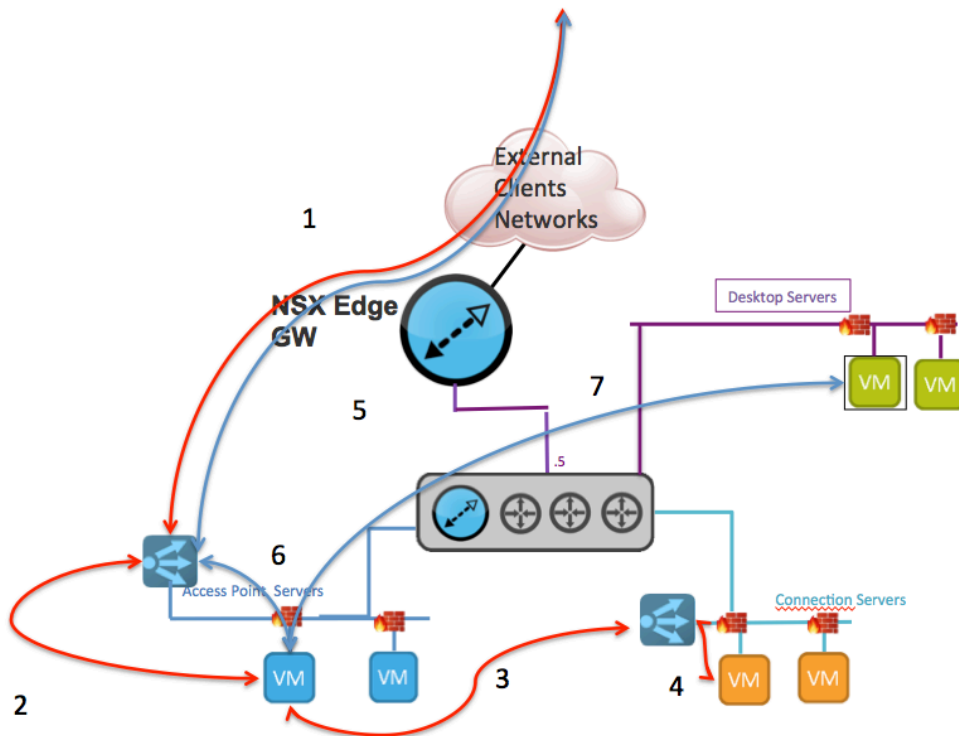
## 6.12 NSX-V Load Balancing Access Point

In this section we will cover the different configurations that you can have to load balancing Access Point Servers. You will select one of these methods for your deployment. You will need to speak with the End User Compute team to understand if they are using Access Point Standard, Access Point Blast Extreme, or Access Point Blast Extreme port sharing, or other topology variants.

## 6.13 Understanding Access Point packet Flow

If we look at the following image we can see how the connections flow

# AP Connection Flow



1. Client connects to the Access Point Virtual Server on the ESG.
2. The ESG selects an Access Point server and connects
3. The Access Point server authenticates the users and connects to the internal Connection Server VIP.
4. The ESG selects a Connection Server and a list of resources is presented back to the AP, and then to the client.
5. The Client connects back to the Access Point Virtual Server over PCoIP/BLAST EXTREME/RDP.
6. The ESG sends the new connection to the same Access Point server as the initial connection.
7. The Access Point server connects to the Desktop Servers, and proxies between the client and the resources.

## 6.14 Deploying Access Point Servers with NSX Load Balancing

Now that we have created the redirect virtual server used by the Access Point Servers in all modes we will create the Access Point – standard mode services

### 6.14.1 Creating the Custom Monitors

Navigate to the desired ESG → Load Balancer → Service Monitors. We are going to create a custom monitor that retrieves the default webpage of the Access Point servers. We will set the interval as 5, timeout at 15 and we will add the no-body extension to minimize the parsing. We will only accept a response code of 200.

Field	Value
NAME	ACCESS-POINT-HTTPS-MON
INTERVAL	5
TIMEOUT	15
MAX RETRIES	3
TYPE	HTTPS
EXPECTED	200
METHOD	GET
URL	/
SEND	
RECIEVE	
EXTENSIONS	no-body

#### 6.14.2 Creating the Profiles

Navigate to the desired ESG → Load Balancer → Application profiles. We are going to create a TCP profile and a UDP profile.

Click on the green + sign and create the following profiles

Name	Protocol	Persistence
AP-UDP	UDP	NONE – Persistence provided by IP HASH
AP-TCP	TCP	NONE – Persistence provided by IP HASH

#### 6.14.3 Create the AP Pool

Create the pool with a NAME, an Algorithm of IP-HASH, and the monitor we created. Note that if the ESG is also the default gateway of the servers you can select the “Transparent” box.

NAME	ALGORITHM	MONITOR	TRANSPARENT
AP-POOL	IP-HASH	ACCESS-POINT-HTTPS-MON	CHECK/UNCHECK

Now we will add pool members, for each pool member in the pool:

FIELD	VALUE
ENABLED	CHECK/UNCHECK
NAME	Name for each server
IP ADDRESS	IP Address for the server
PORT	Leave blank
MONITOR PORT	443
WEIGHT	1
MAX CONNECTIONS	
MIN CONNECTIONS	

#### 6.14.4 Creating the Virtual Servers

Now we will create five virtual servers that make up the application. Access Points in standard mode are a complex application that requires we have all the same ports and protocols listening on the same IP address; TCP 80, TCP 443, TCP 8443, TCP 4172, UDP 4172. All of the TCP services will use the TCP profile and the TCP Pool, and the UDP services will use the UDP profile that we created. All of the services will use the AP Pool.

Virtual Server ID	Name	Description	Default Pool	IP Address	Protocol	Port
virtualServer-7	POD16_AP_TCP_VS_8443		pool-1	10.157.224.90	http	8443
virtualServer-9	POD16_AP_TCP_VS_4172		pool-1	10.157.224.90	tcp	4172
virtualServer-8	POD16_AP_TCP_VS_443		pool-1	10.157.224.90	tcp	443
virtualServer-2	POD16-AP-UDP-VS-4172		pool-2	10.157.224.90	udp	4172

Name/Function	Pool	IP Address	Protocol	Port	Accelerated
PCOIP-TCP	AP-POOL	IP-ADDRESS-1	TCP	4172	Yes
PCOIP-UDP	AP-POOL	IP-ADDRESS-1	UDP	4172	Yes
WEB-PORTAL	AP-POOL	IP-ADDRESS-1	TCP	443	Yes
BLAST EXTREME- PORTAL	AP-POOL	IP-ADDRESS-1	TCP	8443	Yes



#### 6.14.5 Deploying the Access Point in BLAST EXTREME only Mode

Now that we have created the redirect virtual server used by the Access Point Servers in all modes we will create the Access Point – only leveraging BLAST EXTREME. The packet flow is the same as the Standard mode diagram, but we are using a smaller set of connectivity protocols.

#### 6.14.6 Creating the Custom Monitors

Navigate to the desired ESG → Load Balancer → Service Monitors. We are going to create a custom monitor that retrieves the default webpage of the Access Point servers. We will set the interval as 5, timeout at 15 and we will add the no-body extension to minimize the parsing. We will only accept a response code of 200.

Field	Value
NAME	ACCESS-POINT-HTTPS-MON
INTERVAL	5
TIMEOUT	15
MAX RETRIES	3
TYPE	HTTPS
EXPECTED	200
METHOD	GET
URL	/
SEND	
RECEIVE	
EXTENSIONS	no-body

#### 6.14.7 Creating the Profiles

Navigate to the desired ESG → Load Balancer → Application profiles. We are going to create a TCP profile.

Click on the green + sign and create the following profiles

Name	Protocol	Persistence
AP-TCP	TCP	NONE – Persistence provided by IP HASH
AP-UDP	UDP	NONE- Persistence provided by IP HASH

#### 6.14.8 Create the AP Pool

Create the pool with a NAME, an Algorithm of IP-HASH, and the monitor we created. Note that if the ESG is also the default gateway of the servers you can select the “Transparent” box.

NAME	ALGORITHM	MONITOR	TRANSPARENT
AP-POOL	IP-HASH	ACCESS-POINT-HTTPS-MON	CHECK/UNCHECK

Now we will add pool members, for each pool member in the pool:

FIELD	VALUE
ENABLED	CHECK/UNCHECK
NAME	Name for each server
IP ADDRESS	IP Address for the server
PORT	Leave blank
MONITOR PORT	443
WEIGHT	1
MAX CONNECTIONS	
MIN CONNECTIONS	

#### 6.14.9 Creating the Virtual Servers

Now we will create two virtual servers that make up the application. Access Points in BLAST EXTREME mode are a complex application that requires we have all the same ports and protocols listening on the same IP address; TCP 80, TCP 443, TCP 8443. All of the services will use the AP Pool.

Name/Function	Pool	IP Address	Protocol	Port	Accelerated
WEB-PORTAL	AP-POOL	IP-ADDRESS-1	TCP	443	Yes
BLAST EXTREME-PORTAL	AP-POOL	IP-ADDRESS-1	TCP	8443	Yes

#### 6.15 Deploying the Access Point in BLAST EXTREME Mode – port sharing

Now that we have created the redirect virtual server used by the Access Point Servers in all modes we will create the Access Point using only BLAST EXTREME services on port 443. The packet flow is the same as the Standard mode diagram, but we are using a smaller set of connectivity protocols.

### 6.15.1 Creating the Custom Monitors

Navigate to the desired ESG → Load Balancer → Service Monitors. We are going to create a custom monitor that retrieves the default webpage of the Access Point servers. We will set the interval as 5, timeout at 15 and we will add the no-body extension to minimize the parsing. We will only accept a response code of 200.

Field	Value
NAME	ACCESS-POINT-HTTPS-MON
INTERVAL	5
TIMEOUT	15
MAX RETRIES	3
TYPE	HTTPS
EXPECTED	200
METHOD	GET
URL	/
SEND	
RECEIVE	
EXTENSIONS	no-body

### 6.15.2 Creating the Profiles

Navigate to the desired ESG → Load Balancer → Application profiles. We are going to create a TCP profile.

Click on the green + sign and create the following profiles

Name	Protocol	Persistence
AP-TCP	TCP	Source IP

### 6.15.3 Create the AP Pool

Create the pool with a NAME, an Algorithm of IP-HASH, and the monitor we created. Note that if the ESG is also the default gateway of the servers you can select the "Transparent" box.

NAME	ALGORITHM	MONITOR	TRANSPARENT
AP-POOL	LEAST CONNECTIONS	ACCESS-POINT-HTTPS-MON	CHECK/UNCHECK

Now we will add pool members, for each pool member in the pool:

FIELD	VALUE
ENABLED	CHECK/UNCHECK
NAME	Name for each server
IP ADDRESS	IP Address for the server
PORT	Leave blank
MONITOR PORT	443
WEIGHT	1
MAX CONNECTIONS	
MIN CONNECTIONS	

#### 6.15.4 Creating the Virtual Servers

Now we will create one virtual servers that makes up the application. Access Points in BLAST EXTREME mode requires services on TCP:443. All of the services will use the AP Pool.

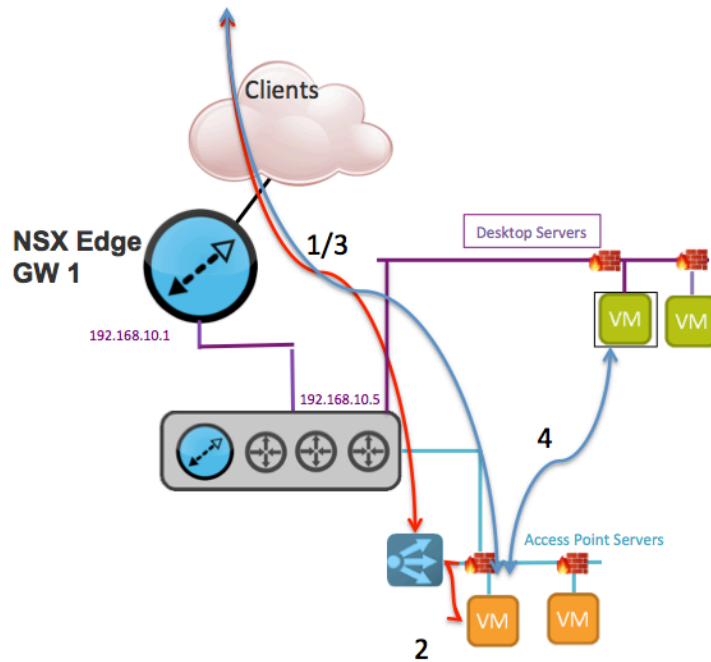
Name/Function	Pool	IP Address	Protocol	Port	Accelerated
WEB-PORTAL/BLAST EXTREME SERVICES	AP-POOL	IP-ADDRESS-1	TCP	443	Yes

#### 6.16 Deploying the Access Point Servers in Direct Mode

In this section we will deploy the Access Point servers in Direct mode. Direct mode is not the most common deployment method for Access Point, but can be supported by NSX-V Load Balancing

##### 6.16.1 Understanding Access Point Server Direct mode Packet Flow

# Access Point- Direct Connection Flow



1. Clients connect to the Virtual Server:443
2. ESG connects to the Access Point Server
3. Clients connect directly to the unique Access Point Server
4. Access Point server connects to resources

## 6.16.2 Creating the Custom Monitors

Navigate to the desired ESG → Load Balancer → Service Monitors. We are going to create a custom monitor that retrieves the default webpage of the Access Point servers. We will set the interval as 5, timeout at 15 and we will add the no-body extension to minimize the parsing. We will only accept a response code of 200.

Field	Value
NAME	ACCESS-POINT-HTTPS-MON
INTERVAL	5
TIMEOUT	15
MAX RETRIES	3
TYPE	HTTPS
EXPECTED	200
METHOD	GET

URL	/
SEND	
RECIEVE	
EXTENSIONS	no-body

### 6.16.3 Creating the Profiles

Navigate to the desired ESG → Load Balancer → Application profiles. We are going to create a TCP profile.

Click on the green + sign and create the following profiles

Name	Protocol	Persistence
AP-TCP	TCP	Source IP

### 6.16.4 Create the AP Pool

Create the pool with a NAME, an Algorithm of IP-HASH, and the monitor we created. Note that if the ESG is also the default gateway of the servers you can select the “Transparent” box.

NAME	ALGORITHM	MONITOR	TRANSPARENT
AP-POOL	LEAST CONNECTIONS	ACCESS-POINT-HTTPS-MON	CHECK/UNCHECK

Now we will add pool members, for each pool member in the pool:

FIELD	VALUE
ENABLED	CHECK/UNCHECK
NAME	Name for each server
IP ADDRESS	IP Address for the server
PORT	Leave blank
MONITOR PORT	443
WEIGHT	1
MAX CONNECTIONS	
MIN CONNECTIONS	

### 6.16.5 Creating the Virtual Servers

Now we will create one virtual servers that makes up the application. Access Points in Direct mode requires services on TCP:443. All of the services will use the AP Pool.

Name/Function	Pool	IP Address	Protocol	Port	Accelerated
WEB-PORTAL/BLAST EXTREME SERVICES	AP-POOL	IP-ADDRESS-1	TCP	443	Yes

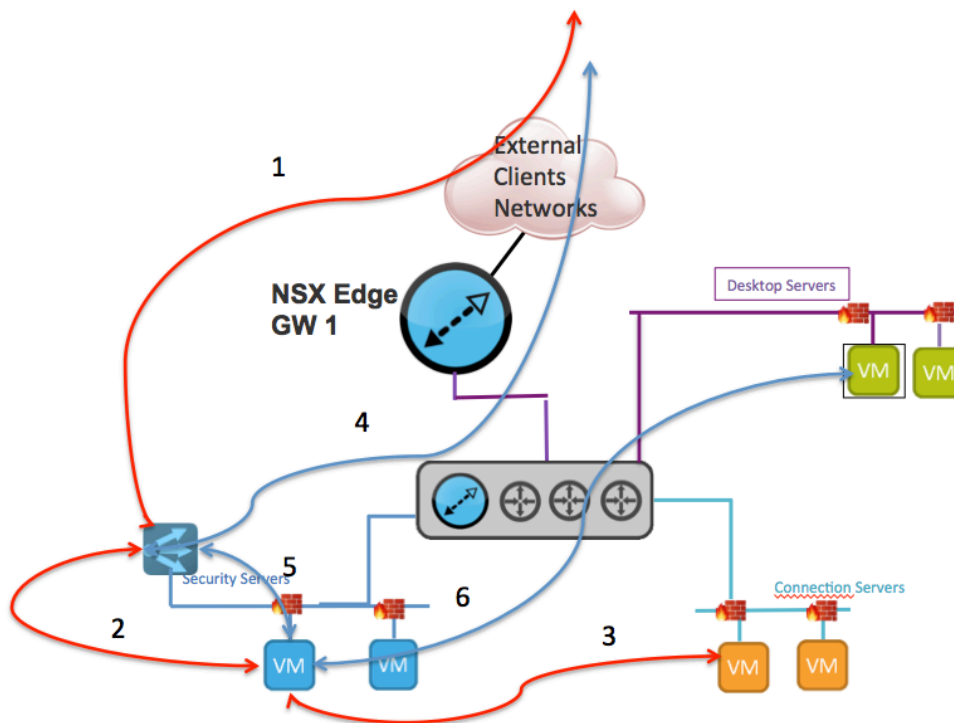
### 6.17 NSX-V Load Balancing Security Servers

If you are using Security Servers for remote employees to access the VDI environment you can use this section to configure load balancing services.

### 6.18 Understanding Security Server packet flow

If we look at the following image we can see how the packets flow.

# Security Server Connection Flow



1. Client connects to Security Server Virtual Server on 443.
2. ESG connects to Security Server on 443
3. Security Server proxies calls to Connection Servers to get inventory, and replies back to client with information.
4. Client connects over PCOIP/Blast Extreme to the Security Server Virtual Server.
5. ESG connects to Security Server.
6. Security Server connects to Desktop Servers.

## 6.19 Deploying the Security Servers in Tunnel Mode

In this configuration the Security Servers will accept all client connections, and then connect to internal resources.

### 6.19.1 Creating the Custom Monitors

Navigate to the desired ESG → Load Balancer → Service Monitors. We are going to create a custom monitor that retrieves the default webpage of the Security Servers. We will set the interval as 5, timeout at 15 and we will add the no-body extension to minimize the parsing. We will only accept a response code of 200.



Field	Value
NAME	SECURITY-SERVER-HTTPS-MON
INTERVAL	5
TIMEOUT	15
MAX RETRIES	3
TYPE	HTTPS
EXPECTED	200
METHOD	GET
URL	/
SEND	
RECEIVE	
EXTENSIONS	no-body

### 6.19.2 Create the Security Server Pool

Now we will need to create the pool needed by the application. Navigate to the desired ESG → Load Balancer → Pools and click on the green + sign

NAME	ALGORITHM	MONITOR	TRANSPARENT
SECURITY-SERVER-POOL	IP-HASH	SECURITY-SERVER-HTTPS-MON	CHECK/UNCHECK

Now we will add pool members, for each pool member in the pool:

FIELD	VALUE
ENABLED	CHECK/UNCHECK
NAME	Name for each server
IP ADDRESS	IP Address for the server
PORT	Leave blank
MONITOR PORT	443
WEIGHT	1
MAX CONNECTIONS	
MIN CONNECTIONS	

### 6.19.3 Creating the Profiles

Navigate to the desired ESG → Load Balancer → Application profiles. We are going to create a TCP profile and a UDP profile.

Click on the green + sign and create the following profiles

Name	Protocol	Persistence
SECURITY-SERVER-TCP	TCP	NONE – Persistence provided by IP HASH
SECURITY-SERVER-UDP	UDP	NONE – Persistence provided by IP HASH

#### 6.19.4 Creating the Virtual Servers

Now we will create five virtual servers that make up the application. Access Points in standard mode are a complex application that requires we have all the same ports and protocols listening on the same IP address; TCP 80, TCP 443, TCP 8443, TCP 4172, UDP 4172, and TCP 3389 (RDP). All of the TCP services will use the TCP profile, and the UDP services will use the UDP profile we created. All virtual servers will use the SECURITY-SERVER-POOL.

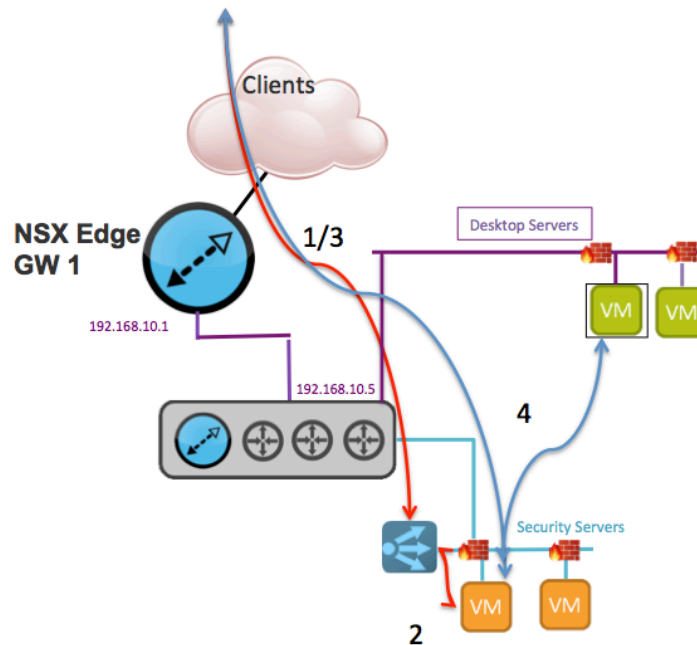
Name/Function	Pool	IP Address	Protocol	Port	Accelerated
PCOIP-TCP	SECURITY-SERVER-POOL	IP-ADDRESS-1	TCP	4172	Yes
PCOIP-UDP	SECURITY-SERVER-POOL	IP-ADDRESS-1	UDP	4172	Yes
WEB-PORTAL	SECURITY-SERVER-POOL	IP-ADDRESS-1	TCP	443	Yes
BLAST EXTREME-PORTAL	SECURITY-SERVER-POOL	IP-ADDRESS-1	TCP	8443	Yes

### 6.20 Deploying the Security Servers in Direct Mode

In this section we will deploy the Security Servers in Direct mode. Direct mode is not the most common deployment method for Security Servers, but can be supported by NSX-V Load Balancing

#### 6.20.1 Understanding Security Server Direct mode Packet Flow

# Security Server - Direct Connection Flow



5. Clients connect to the Virtual Server:443
6. ESG connects to the Security Server
7. Clients connect directly to the unique Security Server
8. Security Server connects to resources

## 6.20.2 Creating the Custom Monitors

Navigate to the desired ESG → Load Balancer → Service Monitors. We are going to create a custom monitor that retrieves the default webpage of the Security Servers. We will set the interval as 5, timeout at 15 and we will add the no-body extension to minimize the parsing. We will only accept a response code of 200.

Field	Value
NAME	SECURITY-SERVER-HTTPS-MON
INTERVAL	5
TIMEOUT	15
MAX RETRIES	3
TYPE	HTTPS
EXPECTED	200
METHOD	GET
URL	/

SEND	
RECIEVE	
EXTENSIONS	no-body

### 6.20.3 Creating the Profiles

Navigate to the desired ESG → Load Balancer → Application profiles. We are going to create a TCP profile.

Click on the green + sign and create the following profiles

Name	Protocol	Persistence
SECURITY-SERVER-TCP	TCP	Source IP

### 6.20.4 Create the Security Server Pool

Create the pool with a NAME, an Algorithm of IP-HASH, and the monitor we created. Note that if the ESG is also the default gateway of the servers you can select the “Transparent” box.

NAME	ALGORITHM	MONITOR	TRANSPARENT
SECURITY-SERVER-POOL	LEAST CONNECTIONS	SECURITY-SERVER-HTTPS-MON	CHECK/UNCHECK

Now we will add pool members, for each pool member in the pool:

FIELD	VALUE
ENABLED	CHECK/UNCHECK
NAME	Name for each server
IP ADDRESS	IP Address for the server
PORT	Leave blank
MONITOR PORT	443
WEIGHT	1
MAX CONNECTIONS	
MIN CONNECTIONS	

### 6.20.5 Creating the Virtual Servers

Now we will create one virtual servers that makes up the application TCP:443.

Name/Function	Pool	IP Address	Protocol	Port	Accelerated
WEB-PORTAL	SECURITY-SERVER-POOL	IP-ADDRESS-1	TCP	443	Yes

## 6.21 NSX-V Load Balancing Connection Servers

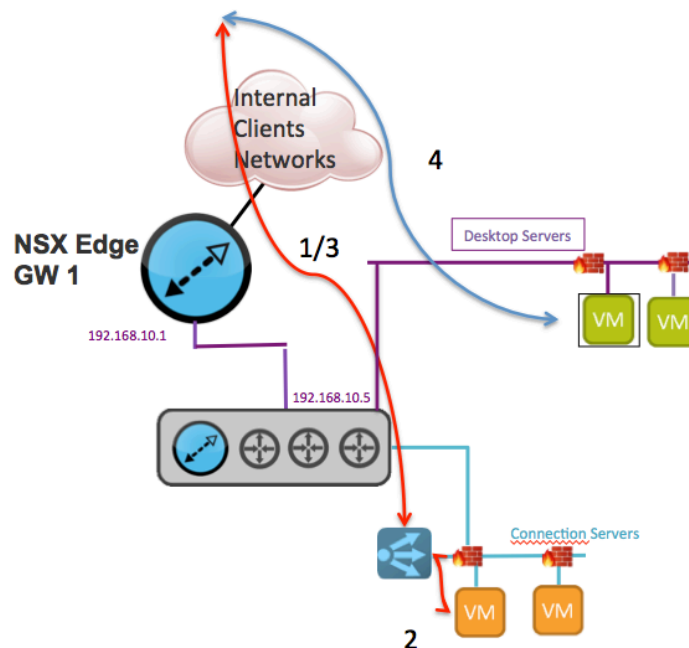
## 6.22 Deploying Connection Server Direct Mode

In Connection Server Direct mode clients authenticate to the Connection Servers, and then connect directly to the desktop servers.

### 6.22.1 Understanding Connection Server Direct Mode packet flow

If we look at the following image we can see how the packets flow

## Connection Server - Direct Connection Flow



1. Client Connects to Connection Server Virtual Server on 443
2. ESG connects to Connection Server
3. The Connection Server replies to client with Desktop server info
4. Client makes direct connection to Desktop servers

### 6.22.2 Creating the Custom Monitors

Navigate to the desired ESG → Load Balancer → Service Monitors. We are going to create a custom monitor that retrieves the default webpage of the Connection Servers. We will set the interval as 5, timeout at 15 and we will add the no-body extension to minimize the parsing. We will only accept a response code of 200.

Field	Value
NAME	CS-HTTPS-MON
INTERVAL	5
TIMEOUT	15
MAX RETRIES	3
TYPE	HTTPS
EXPECTED	200
METHOD	GET
URL	/
SEND	
RECEIVE	
EXTENSIONS	no-body

### 6.22.3 Creating the Profiles

Navigate to the desired ESG → Load Balancer → Application profiles. We are going to create a TCP profile.

Click on the green + sign and create the following profiles

Name	Protocol	Persistence
CS-TCP	TCP	Source IP

### 6.22.4 Creating the Connection Server Pool

Now we will need to create the two pools needed by the application. Navigate to the desired ESG → Load Balancer → Pools and click on the green + sign

NAME	ALGORITHM	MONITOR	TRANSPARENT
CS-POOL	LEAST CONNECTIONS	CS-HTTPS-MON	CHECK/UNCHECK

Now we will add pool members, for each pool member in the pool:

FIELD	VALUE
ENABLED	CHECK/UNCHECK
NAME	Name for each server
IP ADDRESS	IP Address for the server
PORT	Leave blank
MONITOR PORT	443
WEIGHT	1
MAX CONNECTIONS	
MIN CONNECTIONS	

#### 6.22.5 Creating the Connection Server Virtual Servers

Name/Function	Pool	IP Address	Protocol	Port	Accelerated
WEB-PORTAL	CS-POOL	IP-ADDRESS-1	TCP	443	Yes

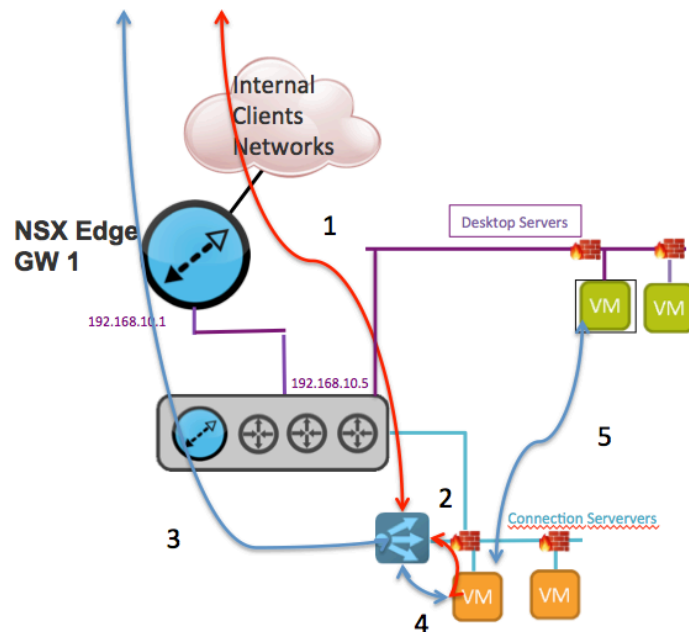
### 6.23 Deploying Connection Server – Tunnel Mode

In tunnel mode all the client connections are via the Connection Server, which then proxies to the Desktop Servers.

#### 6.23.1 Understanding Connection Server Tunnel Mode packet flow

If we look at the following image we can see how the packets flow

# Connection Server - Tunnel Connection Flow



1. Client connects to Connection Server Virtual Server on 443
2. ESG connects to Connection Server on 443, Connection Server instructs clients to connect back to the Virtual Server with relevant protocols (PCOIP, BLAST EXTREME)
3. Client connects back to Virtual Server with BLAST EXTREME/PCOIP
4. ESG connects to same Connection Server
5. The Connection Server proxies to Desktop Servers

## 6.23.2 Creating the Custom Monitors

Navigate to the desired ESG → Load Balancer → Service Monitors. We are going to create a custom monitor that retrieves the default webpage of the Security Servers. We will set the interval as 5, timeout at 15 and we will add the no-body extension to minimize the parsing. We will only accept a response code of 200.

Field	Value
NAME	CS-HTTPS-MON
INTERVAL	5
TIMEOUT	15



MAX RETRIES	3
TYPE	HTTPS
EXPECTED	200
METHOD	GET
URL	/
SEND	
RECIEVE	
EXTENSIONS	no-body

### 6.23.3 Creating the Profiles

Navigate to the desired ESG → Load Balancer → Application profiles. We are going to create a TCP profile and a UDP profile.

Click on the green + sign and create the following profiles

Name	Protocol	Persistence
CS-UDP	UDP	NONE – Persistence provided by IP HASH
CS-TCP	TCP	NONE – Persistence provided by IP HASH

### 6.23.4 Creating the Connection Server Pool

Now we will need to create the two pools needed by the application. Navigate to the desired ESG → Load Balancer → Pools and click on the green + sign

NAME	ALGORITHM	MONITOR	TRANSPARENT
CS-POOL	IP HASH	CS-HTTPS-MON	CHECK/UNCHECK

Now we will add pool members, for each pool member in the pool:

FIELD	VALUE
ENABLED	CHECK/UNCHECK
NAME	Name for each server
IP ADDRESS	IP Address for the server
PORT	Leave blank
MONITOR PORT	443
WEIGHT	1
MAX CONNECTIONS	
MIN CONNECTIONS	

## 6.23.5 Creating the Security Server Virtual Servers

To operate the Connection Servers in tunnel mode you will need to configure the following virtual servers.

Name/Function	Pool	IP Address	Protocol	Port	Accelerated
PCOIP-TCP	CS- POOL	IP- ADDRESS- 1	TCP	4172	Yes
PCOIP-UDP	CS- POOL	IP- ADDRESS- 1	UDP	4172	Yes
WEB-PORTAL	CS- POOL	IP- ADDRESS- 1	TCP	443	Yes
BLAST EXTREME- PORTAL	CS- POOL	IP- ADDRESS- 1	TCP	8443	Yes

## 7 Appendix

### 7.1 Application Rules Used

In this document we used 1 application rule that redirected from HTTP to HTTPS

Name: HTTP\_REDIR

Action: redirect scheme https if !{ ssl\_fc }

### 7.2 NSX Edge Command Line Interface Load balancer Troubleshooting

You can use the ESG console form vSphere or a SSH to an ESG to gather information to verify if the configuration is working as desired.

## 7.2.1 Accessing the Edge

Using your preferred SSH client SSH to the DNS name or IP address of the edge in question. You will log in with the admin account (the edge admin account is READ ONLY).

## 7.2.2 Verify the Load Balancing Software is running from the CLI

```
NSX-edge-10-0> show service loadbalancer
```

```
-----  
Loadbalancer Services Status:
```

```
L7 Loadbalancer : running
```

```
-----  
L7 Loadbalancer Status Information:
```

STATUS	PID	MAX_MEM_MB	MAX SOCK	MAX_CONN	MAX_PIPE	CUR_CONN	CONN_RATE
running	27355	0	16424	8192	0	0	0

```
-----  
L4 Loadbalancer Statistics:
```

```
Prot LocalAddress:Port Scheduler Flags
```

```
-> RemoteAddress:Port Forward Weight ActiveConn InActConn
```

TCP	10.157.224.90:443 sh						
->	10.157.224.53:443	Masq	1	0	0		
->	10.157.224.54:443	Masq	1	1	0		
TCP	10.157.224.90:4172 sh						
->	10.157.224.53:4172	Masq	1	0	0		
->	10.157.224.54:4172	Masq	1	0	0		
TCP	10.157.224.90:8443 sh						
->	10.157.224.53:8443	Masq	1	0	0		
->	10.157.224.54:8443	Masq	1	0	0		
TCP	10.157.224.90:32111 sh						
->	10.157.224.53:32111	Masq	1	0	0		
->	10.157.224.54:32111	Masq	1	0	0		
TCP	10.157.224.91:443 sh						
TCP	10.157.224.91:4172 sh						
TCP	10.157.224.91:8443 sh						
TCP	10.157.224.92:443 sh						
->	192.168.10.5:443	Masq	1	0	0		
->	192.168.10.6:443	Masq	1	0	0		
TCP	10.157.224.92:4172 sh						
->	192.168.10.5:4172	Masq	1	0	0		
->	192.168.10.6:4172	Masq	1	0	0		
TCP	10.157.224.92:8443 sh						
->	192.168.10.5:8443	Masq	1	0	0		
->	192.168.10.6:8443	Masq	1	0	0		
TCP	10.157.224.93:443 sh						
TCP	10.157.224.93:4172 sh						
TCP	10.157.224.93:8443 sh						
UDP	10.157.224.90:4172 sh						
->	10.157.224.53:4172	Masq	1	0	0		
->	10.157.224.54:4172	Masq	1	0	1		
UDP	10.157.224.91:4172 sh						
UDP	10.157.224.92:4172 sh						
->	192.168.10.5:4172	Masq	1	0	0		
->	192.168.10.6:4172	Masq	1	0	0		
UDP	10.157.224.93:4172 sh						

## 7.2.3 Looking at the Pool from the CLI

```
NSX-edge-10-0> show service loadbalancer pool POD16-AP-OA-POOL
```

-----  
Loadbalancer Pool Statistics:

```
POOL POD16-AP-OA-POOL
| LB METHOD ip-hash
| LB PROTOCOL L4
| Transparent disabled
| SESSION (cur, cps, total) = (5, 0, 8)
| BYTES in = (4767722), out = (3414714)
+>POOL MEMBER: POD16-AP-OA-POOL/INT-AP4, STATUS: UP
| | HEALTH MONITOR = MONITOR SERVICE, AP-HTTPS-MONITOR:OK
| | | LAST STATE CHANGE: 2016-06-02 03:01:20
| | | LAST CHECK: 2016-06-02 23:59:45
| | SESSION (cur, cps, total) = (1, 0, 0)
| | BYTES in = (4743857), out = (3375849)
+>POOL MEMBER: POD16-AP-OA-POOL/INT-AP3, STATUS: UP
| | HEALTH MONITOR = MONITOR SERVICE, AP-HTTPS-MONITOR:OK
| | | LAST STATE CHANGE: 2016-06-02 03:01:35
| | | LAST CHECK: 2016-06-02 23:59:45
| | SESSION (cur, cps, total) = (4, 0, 8)
| | BYTES in = (23865), out = (38865)
```

NSX-edge-10-0>

## 7.2.4 Looking at the Virtual Server from the CLI

NSX-edge-10-0> show service loadbalancer virtual

-----  
Loadbalancer VirtualServer Statistics:

```
VIRTUAL POD16-SS-TCP-VS-4172
| ADDRESS [10.157.224.92]:4172
| SESSION (total) = (0)
| RATE (cur) = (0)
| BYTES in = (0), out = (0)
+>POOL SECURITY_SERVER_POOL
| LB METHOD ip-hash
| LB PROTOCOL L4
| Transparent enabled
| SESSION (cur, cps, total) = (0, 0, 0)
| BYTES in = (0), out = (0)
+>POOL MEMBER: SECURITY_SERVER_POOL/SECURITY-SERVER-10-6, STATUS: UP
| | HEALTH MONITOR = MONITOR SERVICE, SECURITY-SERVER-SSL:OK
| | | LAST STATE CHANGE: 2016-06-02 08:29:50
| | | LAST CHECK: 2016-06-02 23:50:24
| | SESSION (cur, cps, total) = (0, 0, 0)
| | BYTES in = (0), out = (0)
+>POOL MEMBER: SECURITY_SERVER_POOL/SECURITY-SERVER-10-5, STATUS: UP
| | HEALTH MONITOR = MONITOR SERVICE, SECURITY-SERVER-SSL:OK
| | | LAST STATE CHANGE: 2016-06-02 08:29:50
| | | LAST CHECK: 2016-06-02 23:50:25
| | SESSION (cur, cps, total) = (0, 0, 0)
| | BYTES in = (0), out = (0)
```

## 7.2.5 Checking the status of all Load Balancer Monitors

NSX-edge-10-0> show service loadbalancer monitor

-----  
Loadbalancer Health Check Statistics:

MONITOR PROVIDER	POOL	MEMBER	HEALTH STATUS
built-in	AP-HTTP	int-ap3	default_http_monitor:L7OK
built-in	AP-HTTP	int-ap4	default_http_monitor:L7OK
built-in	HTTPS_POOL	AP-3-SSL-POOL	default_https_monitor:L7OK
built-in	HTTPS_POOL	AP-4-SSL-POOL	default_https_monitor:L7OK
monitor service	POD16-AP-OA-POOL	INT-AP4	AP-HTTPS-MONITOR:OK

```

monitor service POD16-AP-OA-POOL INT-AP3 AP-HTTPS-MONITOR:OK
monitor service SECURITY_SERVER_POOL SECURITY-SERVER-10-6 SECURITY-SERVER-SSL:OK
monitor service SECURITY_SERVER_POOL SECURITY-SERVER-10-5 SECURITY-SERVER-SSL:OK
NSX-edge-10-0>

```

### 7.2.6 Validating Cross Virtual Server connections (PCoiP)

A given client should be connected to the same backend servers. In the example below the client IP is 10.113.226.77, Virtual Server IP is 10.157.224.90, and the backend server IP is 10.157.224.53.

```
#show service loadbalancer session | include %CLEINTIPADDRESS%
```

```

NSX-edge-10-0> show service loadbalancer session | include 10.113.226.77
UDP 00:58 UDP 10.113.226.77:50002 10.157.224.90:4172 10.157.224.53:4172
TCP 59:57 ESTABLISHED 10.113.226.77:60098 10.157.224.90:443 10.157.224.53:443
TCP 59:33 ESTABLISHED 10.113.226.77:60102 10.157.224.90:443 10.157.224.53:443
TCP 59:25 ESTABLISHED 10.113.226.77:60101 10.157.224.90:443 10.157.224.53:443
TCP 59:25 ESTABLISHED 10.113.226.77:60094 10.157.224.90:443 10.157.224.53:443
NSX-edge-10-0>

```

## 7.3 Configuring the Pool Behavior to Support Transparent topologies

If you decide to deploy the ESG in a transparent topology you will need to configure the Load Balancing pool for this topology.

Enabled	Name	IP Address / VC Container	Weight	Monitor Port	Port	Max Connections	Min Connection.
✓	web-03a	172.17.1...	1	443	443	0	0
✓	web-01b	172.17.1...	1	443	443	0	0

Transparent

## 7.4 Log Insight

vCenter, ESXi hosts, the ESGs, and firewall rules have all been configured to send syslog traffic to VMware LogInsight™. LogInsight has powerful dashboards that can be used to support your environment.

## 7.4.1 Configure an ESG to use LogInsight

Navigate to the desired ESG → Settings → Configuration

The screenshot shows the configuration page for 'OneArm-LoadBalancer-01'. The 'Manage' tab is active, and the 'Settings' sub-tab is selected. The 'Load Balancer' configuration is displayed with the following details:

Details:		Action
Size:	Compact	
Host Name:	OneArm-LoadBalancer-01	
Auto generate rules:	Enabled	
Syslog servers:		<a href="#">Change</a>
Server 1:	192.168.110.24	
Server 2:		

## 7.5 Edge configuration customization

Not Applicable