Network Configuration Example

MetaFabric Architecture 2.0: Configuring Virtual Chassis Fabric and VMware NSX

Modified: 2017-04-14

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Network Configuration Example MetaFabric Architecture 2.0: Configuring Virtual Chassis Fabric and VMware NSX

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

Table of Contents

CHAPTER 1

MetaFabric Architecture 2.0 Configuration

About This Network Configuration Example

This network configuration example provides an overview of MetaFabric Architecture 2.0, offers configurations for Virtual Chassis Fabric (VCF) and VMware NSX, and provides a sample use case showing how these technologies support virtual machines, VM movement, and applications in a data center network.

Understanding MetaFabric Architecture 2.0

MetaFabric Architecture 2.0 is based on the integration of the Juniper Networks® Virtual Chassis Fabric and VMware NSX for vSphere to create an enterprise private cloud environment. This architecture gives you a single point of control for both the physical and virtual networking components of a network. This topic explains these MetaFabric features and concepts:

High-Performance Network Fabric

Enterprise private clouds require high-speed and high-performance network fabrics with low latency. The network fabric also needs to be easy to manage. Adding more capacity, configuration changes, and software upgrades must be plug-and-play, simple, and seamless.

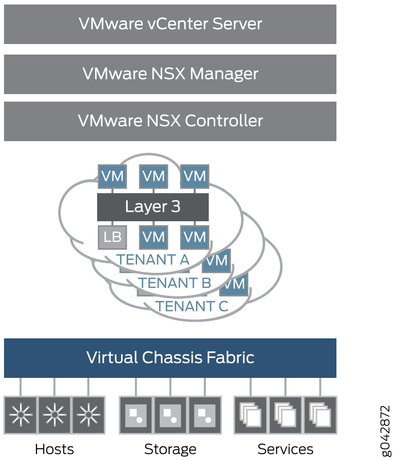
Virtual Chassis Fabric is a plug-and-play Ethernet fabric technology that allows you to combine a set of switches into a single logical switch. A Virtual Chassis Fabric has the following benefits:

- Single point of management
- Supports software-defined networking (SDN) with Virtual Extensible LAN (VXLAN) integration
- Supports Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, and 40-Gigabit Ethernet interfaces
- Full Layer 2, Layer 3, and multicast support
- Equal-cost multipath (ECMP)
- Scales up to 20 switches
- Nonstop software upgrade (NSSU)

Virtual Chassis Fabric allows you to easily manage your data center environment. All devices are—at most—only three hops away, regardless of where they are located in the data center. Virtual Chassis Fabric offers line-rate performance for all types of workloads, whether virtual machines or bare metal servers. One of the key benefits is that Virtual Chassis Fabric supports software-defined data centers (SDDC) and SDN with VMware NSX and Juniper Networks Contrail.

Virtual Chassis Fabric is a next-generation Ethernet fabric that delivers a high-speed and high-performance network fabric through a spine and leaf architecture as shown in Figure 1 on page 6.

Figure 1: MetaFabric Architecture 2.0



End-to-end over-subscription is user-configurable from 1:1 to 6:1, and the end-to-end latency is less than 3 microseconds. Virtual Chassis Fabric allows the entire network fabric to be managed and operated as a single logical switch. All configuration changes and software upgrades are performed in a single place. Virtual Chassis Fabric also supports NSSU, which allows you to upgrade all member switches on a VCF with minimal network traffic disruption during the upgrade. Adding new capacity to the Virtual Chassis Fabric is as simple as cabling the new Juniper Networks QFX5100 switch and powering it up. The QFX5100 switch is automatically discovered and added to the Virtual Chassis Fabric as a new line card.

Software-Defined Networking

Enterprise private clouds need to quickly deploy applications in a multi-tier network which enables better scale and security segmentation. Each application requires custom network segmentation and security policies. VMware NSX for vSphere is an SDN solution that allows you to quickly create virtual networks that are multi-tier with segmentation and security policies.

VMware NSX for vSphere has full integration with other VMware tools such as vCenter Server, vCenter Operations Manager, and vCloud Director.

Virtualized Hosts

Enterprise private clouds need to quickly deliver virtual machines to their end users. One of the most important things is having server hardware that is listed in the VMware hardware compatibility list (HCL). For the MetaFabric Architecture 2.0 lab, we used Supermicro servers and Intel network cards.

Network-Attached Storage

One of the final tenets of enterprise private cloud is doing more with less. When it comes to storage, we want to converge the storage traffic and application traffic onto the same hardware. This means that we used network-attached storage (NAS). It is also important to choose a NAS device that is listed in the VMware HCL to ensure that everything works properly. For the MetaFabric Architecture 2.0 lab, we used a Synology NAS with iSCSI protocol. Because iSCSI uses IP, we can run both the storage traffic and application traffic across the same Virtual Chassis Fabric network.

MetaFabric Architecture 2.0 Sizing

MetaFabric Architecture 2.0 allows you to build a small enterprise private cloud or use a scale out architecture to build a hyper-scale private cloud with over 1,000,000 VMs as shown in Table 1 on page 7. Whether you use the small architecture or the scale out architecture, MetaFabric Architecture 2.0 is seamless and uses the same products and technologies: QFX5100 family switches and Virtual Chassis Fabric.

The assumption is that each host is connected to the Virtual Chassis Fabric with two 10-Gigabit Ethernet connections and can support 100 VMs per host.

Table 1: MetaFabric Architecture 2.0 Sizes

Size	Network Ports	Hosts	Virtual Machines
Small	96x10-Gigabit Ethernet	48	4,800
Medium	768x10-Gigabit Ethernet	384	38,400

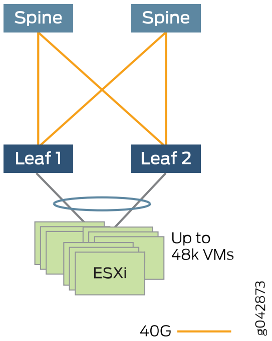
Table 1: MetaFabric Architecture 2.0 Sizes *(continued)*

Size	Network Ports	Hosts	Virtual Machines
Large - 4 PODs	3,072x10-Gigabit Ethernet	1,536	153,600
Scale Out - 8 PODs	6,144x10-Gigabit Ethernet	3,072	307,200
Scale Out - 16 PODs	12,288x10-Gigabit Ethernet	6,144	614,400
Scale Out - 32 PODs	24,576x10-Gigabit Ethernet	12,288	1,228,800

Small-Sized Virtual Chassis Fabric

The smallest possible Virtual Chassis Fabric you can create consists of four switches: two spine switches and two leaf switches as shown in Figure 2 on page 8.

Figure 2: Small-Sized Virtual Chassis Fabric with Four Members

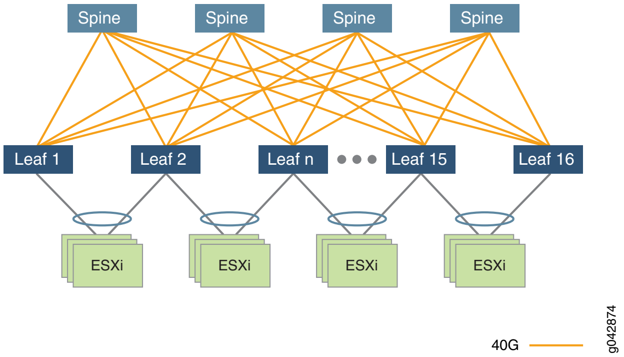


The small Virtual Chassis Fabric can support up to 96 10-Gigabit Ethernet ports with 48 hosts and 4,800 VMs.

Medium-Sized Virtual Chassis Fabric

Using the same Virtual Chassis Fabric technology, you can add up to 20 members into the Ethernet fabric as shown in Figure 3 on page 9. Although there are 20 members in the Virtual Chassis Fabric, it is managed as a single, logical switch.

Figure 3: Medium-Sized Virtual Chassis Fabric with 20 Members

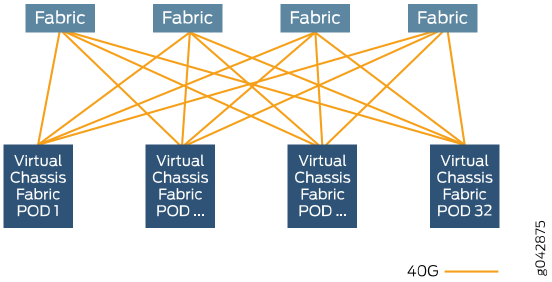


The medium-sized, 20-member Virtual Chassis Fabric supports 768 10-Gigabit Ethernet ports with hosts and 38,400 VMs.

Scale Out Architecture

When a single Virtual Chassis Fabric is not large enough, you can simply move to a scale out architecture. Each Virtual Chassis Fabric becomes a point of delivery (POD). In a scale out architecture, there are many PODs that are connected through a fabric layer as shown in Figure 4 on page 9.

Figure 4: Scale Out Architecture with Virtual Chassis Fabric PODs


Fabric
Fabric
Fabric
Fabric
Virtual Chassis Fabric POD 1
Virtual Chassis Fabric POD ...
Virtual Chassis Fabric POD ...
Virtual Chassis Fabric POD 32
40G
g042875

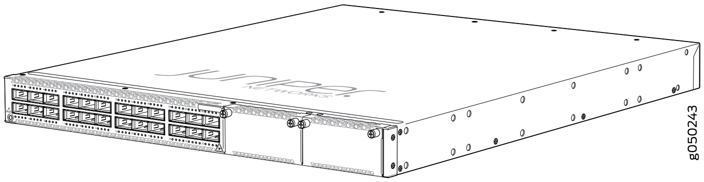
The fabric switches that are connecting the PODs of Virtual Chassis Fabric are Juniper Networks QFX5100-24Q switches, which support 32 40-Gigabit Ethernet interfaces. A scale-out architecture allows MetaFabric Architecture 2.0 to scale beyond a single Virtual Chassis Fabric and support up to 24,576 10-Gigabit Ethernet ports with 12,288 hosts and over 1,000,000 VMs. Other benefits of a scale-out architecture are high availability and resiliency. Each POD is managed separately and is treated as a separate fault domain. A failure in one POD does not impact the performance and availability of the other 31 PODs in MetaFabric Architecture 2.0.

Virtual Chassis Fabric Platforms and Topology

A Virtual Chassis Fabric is constructed using a set of switches. Virtual Chassis Fabric offers the best performance and features with the QFX5100 series which comes in two models, the QFX5100-24Q switch and the QFX5100-48S switch.

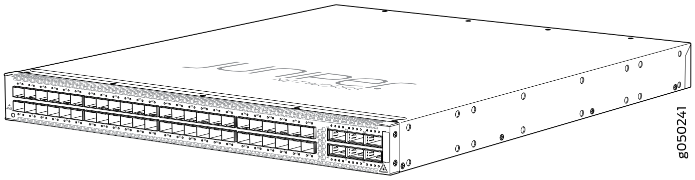
As shown in Figure 5 on page 10, the QFX5100-24Q switch offers 24 built-in QSFP+ ports and two modules that can support 4 ports of QSFP+; this brings the total number of 40-Gigabit Ethernet ports to 32. One nice feature of the QSFP+ ports is that you can break a single port into four 10-Gigabit Ethernet ports.

Figure 5: Juniper Networks QFX5100-24Q Switch



The QFX5100-48S switch offers 48 10-Gigabit Ethernet ports and six 40-Gigabit Ethernet ports as shown in Figure 6 on page 10.

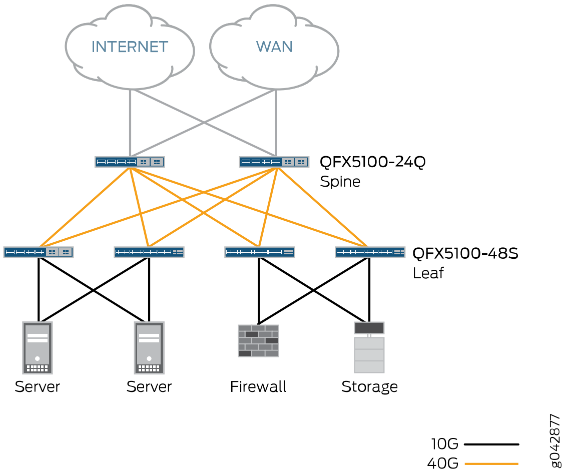
Figure 6: Juniper Networks QFX5100-48S Switch



Virtual Chassis Fabric is most often deployed using the QFX5100-24Q and QFX5100-48S switches; these switches complement each other when building a simple 3-stage topology.

The Virtual Chassis Fabric topology is a 3-stage Clos architecture which offers the best latency, performance, and scale. The QFX5100-24Q switch is most commonly used as a spine switch, and the QFX5100-48S is most commonly used as a leaf switch as shown in Figure 7 on page 11.

Figure 7: Virtual Chassis Fabric Architecture



One benefit of a Virtual Chassis Fabric is that there are no physical restrictions on where you can connect devices; you can use both the spine and leaf switches to connect servers, routers, or any other device. When creating a small to medium-sized data center, port flexibility creates a distinct advantage because a single fabric can connect all servers, storage, firewalls, and even the Internet and WAN.

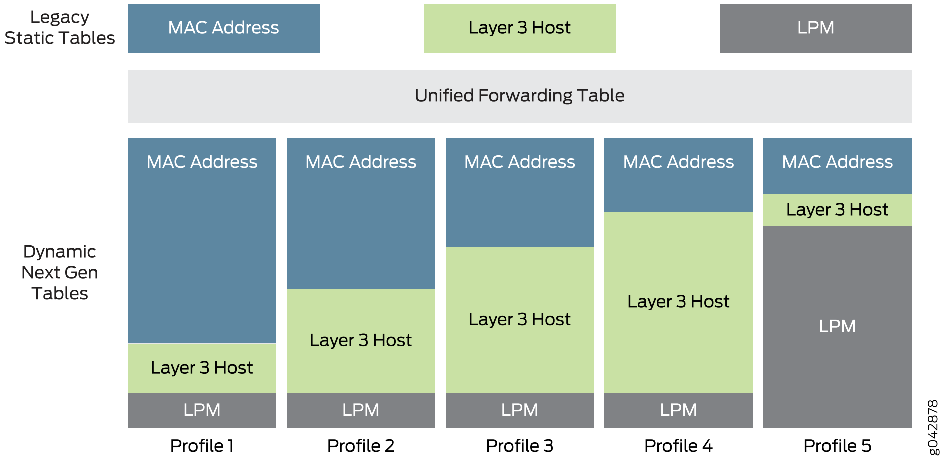
Virtual Chassis Fabric Performance and Scale

Virtual Chassis Fabric is a high-speed Ethernet fabric for every device in the data center. The very nature of the spine and leaf topology enables deterministic traffic patterns and latency, which means that applications are lightning fast. Some of the performance characteristics of Virtual Chassis Fabric are as follows:

- End-to-end latency of 2.5 microseconds
- Line-rate performance of 10-Gigabit Ethernet and 40-Gigabit Ethernet
- 1.28Tbps of forwarding capacity per switch
- 25.6Tbps of forwarding capacity for the entire fabric

As you can see, there is enough scale and performance in a Virtual Chassis Fabric to support any servers and applications that you are using. Virtual Chassis Fabric uses a new technology called the Unified Forwarding Table to give you the power to adjust the logical scale of the Ethernet fabric. Virtual Chassis Fabric uses next-generation flexible tables as shown in Figure 8 on page 12.

Figure 8: Unified Forwarding Table



There are five fabric profiles that you can choose from—each profile incrementally increases the amount of Layer 3 scale, as shown in Table 2 on page 12.

Table 2: Unified Forwarding Table - Fabric Profiles

Profile	MAC Addresses	Layer 3 Hosts	Longest Prefix Match
l2-profile-one	288,000	16,000	16,000
l2-profile-two	224,000	56,000	16,000
l2-profile-three	160,000	88,000	16,000
l3-profile	96,000	120,000	16,000
lpm-profile	32,000	16,000	128,000

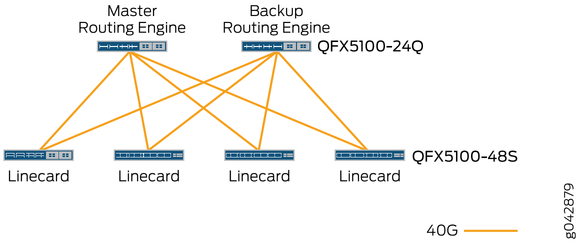
High Availability

Virtual Chassis Fabric leverages the functionality from carrier-class routers such as the Juniper Networks MX Series and T Series to provide high availability in the Ethernet fabric.

- Graceful Routing Engine switchover (GRES)—Keeps the operating system state synchronized between the master and backup Routing Engines
- Nonstop active routing (NSR)—Keeps the routing protocol state synchronized between the master and backup Routing Engines
- Nonstop bridging (NSB)—Keeps the Layer 2 protocol state synchronized between the master and backup Routing Engines

The spine switches act as the master and backup Routing Engines in a Virtual Chassis Fabric as shown in Figure 9 on page 13.

Figure 9: Virtual Chassis Fabric Roles



The other leaf switches in a Virtual Chassis Fabric act as simple line cards. If the master Routing Engine experiences a failure, the backup Routing Engine immediately becomes the new master Routing Engine. Traffic will not be interrupted because GRES, NSR, and NSB keep the two Routing Engines continuously synchronized.

Virtual Chassis Fabric is an easy to manage, high-performance, and highly-available Ethernet fabric that allows you to build a best-of-class data center network.

Related Documentation

- Understanding Network Virtualization with VMware NSX on page 13
- Example: Configuring Virtual Chassis Fabric and VMware NSX for MetaFabric Architecture 2.0 on page 19

Understanding Network Virtualization with VMware NSX

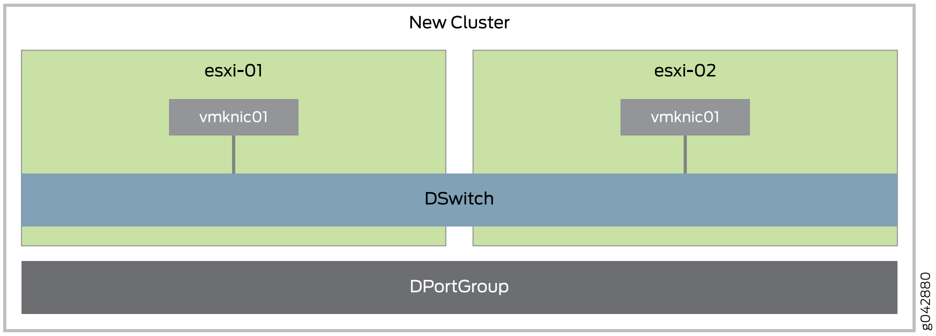
Understanding how physical networks and virtual networks come together to provide an end-to-end solution is critical to running a stable production environment. The physical and virtual networks interact with each other to provide different functionality. There are some additional layers with the introduction of overlay networks such as VMware NSX. This topic explains the following concepts:

- VMware vSphere Architecture on page 13
- VMware NSX on page 14
- VMware NSX Edge Gateway on page 16
- Overlay Architecture on page 18

VMware vSphere Architecture

The VMware vSphere architecture is very simple. There are two ESXi hosts in a cluster called “New Cluster.” Both of the hosts have a distributed virtual switch called “DSwitch” with a single port group “DPortGroup” as shown in Figure 10 on page 14. The cluster is assigned to a data center called “Datacenter.”

Figure 10: VMware vSphere Architecture



All NSX appliance VMs are placed into the distributed virtual switch “DSwitch.” The local vSwitch0 is no longer used for any type of traffic; all underlay traffic will ingress and egress the distributed virtual switch.

VMware NSX

To enable SDN, there are many functions from management to packet forwarding that need to be performed. Each functional component is described next.

VMware NSX Manager

The VMware NSX Manager provides integration with VMware vCenter Server which allows you to manage the VMware NSX environment through VMware vCenter. All VMware NSX operations and configuration is done through VMware vCenter, which communicates with VMware NSX Manager through APIs to delegate tasks to the responsible owner.

VMware NSX Controller

All virtual network provisioning and MAC address learning is handled through the VMware NSX Controller. You can think of the VMware NSX Controller as the virtualized control plane of the SDN network.

VMware NSX Logical Distributed Router

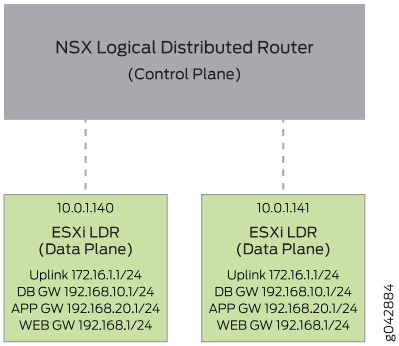
The VMware NSX Logical Distributed Router (LDR) is responsible for forwarding and routing all packets through the virtualized SDN networks. It provides the following functionality:

- Default gateway for all VMs
- MAC address learning and flooding
- Bridging and routing all packets between different virtual networks
- Peers with the VMware NSX Edge to progress egress traffic outside of the virtual networks
- Virtual tunnel end-point (VTEP) termination
- Security policies and enforcement
- Multi-tenancy

The NSX LDR is a great tool for coarse or fine-grained virtual network segmentation. Multiple VMware NSX LDRs can be created to enable multi-tenancy or a completely separate security zone for regularity requirements such as Payment Card Industry Data Security Standard (PCI DSS). Each VMware NSX LDR can create virtual switches, which are just VXLAN Network Identifiers (VNIs). You can treat virtual switches just like you used to use VLANs in a physical network. Virtual switches are an easy way to create multi-tiered networks for applications.

The VMware NSX LDR is split into two components: a control plane and data plane. The control plane is responsible for the management and provisioning of changes. The VMware NSX LDR is also installed into each VMware ESXi host to handle the traffic forwarding and routing as shown in Figure 11 on page 15.

Figure 11: VMware NSX LDR Control Plane and Data Plane



Each VMware host has a copy of the VMware NSX LDR running in the hypervisor. All of the gateway interfaces and IP addresses are distributed throughout the VMware cluster. This allows VMs to directly access their default gateway at the local hypervisor. VMware NSX supports three methods for MAC address learning:

- Unicast—Each VMware host has a TCP connection to the VMware NSX Controller for MAC address learning.
- Multicast—The physical network – in this case, the Virtual Chassis Fabric – uses multicast to replicate broadcast, unknown unicast, and multicast traffic between all VMware hosts participating in the same VNI.
- Hybrid—Each VMware host has a TCP connection to the VMware NSX Controller for MAC address learning, but uses the physical network for local process of broadcast, unknown unicast, and multicast traffic for performance.

If your environment is 100 percent virtualized, we recommend that you use either unicast or hybrid mode. If you need to integrate physical servers – such as mainframes – into the VMware NSX virtual environment, you need to use multicast mode for MAC address learning. Virtual Chassis Fabric allows you to configure multicast with a single IGMP command and not have to worry about designing and maintaining multicast protocols such as PIM.

All of the VMware NSX virtual switches are associated with a VNI. Depending on the traffic profile, the LDR can either locally forward or route the traffic. If the destination is on another host or outside of the virtual NSX networks, the VMware NSX LDR can route the traffic out to the VMware NSX Edge Gateway.

Each hypervisor has a virtual tunnel end-point (VTEP) that is responsible for encapsulating VM traffic inside of a VXLAN header and routing the packet to a destination VTEP for further processing. Traffic can be routed to another VTEP on a different host or the VMware NSX Edge Gateway to access the physical network.

In Table 3 on page 16, you can see all of the possible traffic patterns and how the VMware NSX LDR handles them.

Table 3: Traffic Patterns Handled by VMWare LDR

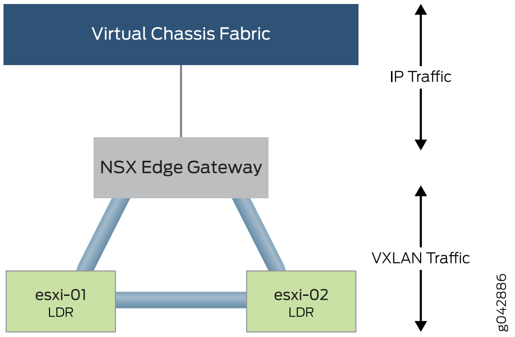
Source	Destination	Action
Local VM	Local VM, same network	Locally switch traffic
Local VM	Local VM, different network	Locally route traffic
Local VM	Remote VM, same network	Encapsulate traffic with VXLAN header and route to destination VTEP
Local VM	Remote VM, different network	Encapsulate traffic with VXLAN header and route to destination VTEP
Local VM	Internet	Encapsulate traffic with VXLAN header and route to VMware NSX Edge Gateway
Local VM	Physical server outside of NSX virtual networks	Encapsulate traffic with VXLAN header and route to VMware NSX Edge Gateway

VMware NSX Edge Gateway

The VMware NSX Edge Gateway is responsible for bridging the virtual networks with the outside world. It acts as a virtual WAN router that is able to peer with physical networking equipment so that all of the internal virtual networks can access the Internet, WAN, or any other physical resources in the network. The VMware NSX Edge Gateway can also provide centralized security policy enforcement between the physical network and the virtualized networks.

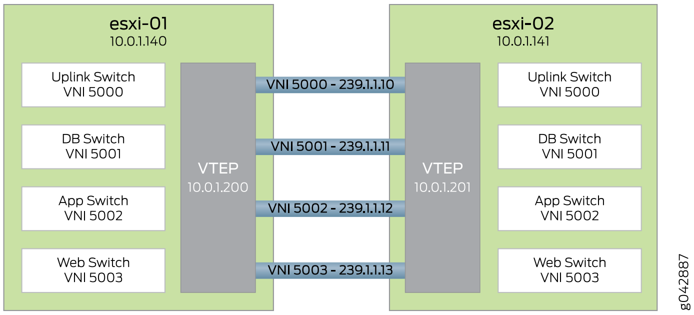
The VMware NSX Edge Gateway and LDR have a full mesh of VXLAN tunnels as shown in Figure 12 on page 17. This enables any VMware ESXi host to communicate directly through the VXLAN tunnels when they need to switch or route traffic. If traffic needs to enter or exit the VMware NSX environment, the VMware NSX Edge Gateway removes the VXLAN header and routes the traffic through its “Uplink” interface and into the Virtual Chassis Fabric.

Figure 12: VXLAN Tunnels



Each virtual switch on the VMware NSX LDR needs to be mapped to a VNI and multicast group to enable the data plane and control plane. It is as simple as choosing a different VNI and multicast group per virtual switch as shown in Figure 13 on page 17.

Figure 13: Overlay Tunnels and Multicast Groups



As VM traffic from esxi-01 needs to reach esxi-02, it simply passes through the VXLAN tunnels. Depending on the type of traffic, there are different actions that can take place as shown in Table 4 on page 17.

Table 4: Traffic Types and Actions

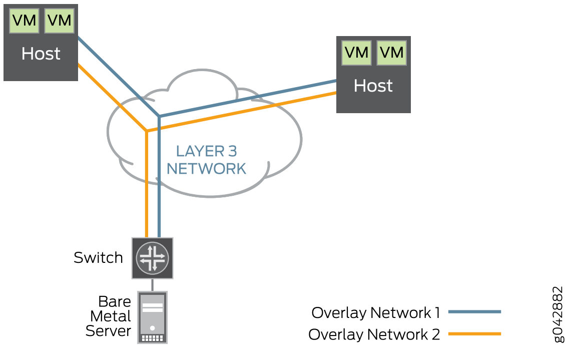
Traffic Type	Action
Unicast	Route directly to the remote VTEP through the Virtual Chassis Fabric
Unknown Unicast	Flood through multicast in the Virtual Chassis Fabric
Multicast	Flood through multicast in the Virtual Chassis Fabric

It is possible to associate multiple VNIs with the same multicast group; however, in the MetaFabric Architecture 2.0 lab, we assigned each VNI a separate multicast group for simplicity.

Overlay Architecture

The term “underlay” refers to the physical networking equipment; in this case, it is the Virtual Chassis Fabric. The term “overlay” refers to any virtual networks created by VMware NSX. Virtual networks are created with a MAC-over-IP encapsulation called VXLAN. This encapsulation allows two VMs on the same network to talk to each other, even if the path between the VMs needs to be routed as shown in Figure 14 on page 18.

Figure 14: Overlay Architecture



All VM-to-VM traffic is encapsulated in VXLAN and transmitted over a routed network to the destination host. Once the destination host receives the VXLAN encapsulated traffic, it can remove the VXLAN header and forward the original Ethernet packet to the destination VM. The same traffic pattern occurs when a VM talks to a bare metal server. The exception is that the top-of-rack switch handles the VXLAN encapsulation on behalf of the physical server, as opposed to the hypervisor.

The advantage of VXLAN encapsulation is that it allows you to build a network that is based on Layer 3. The most common underlay architecture for SDN is the IP fabric. All switches communicate with each other through a typical routing protocol such as BGP. There is no requirement for Layer 2 protocols such as Spanning Tree Protocol (STP).

One of the drawbacks to building an IP fabric is that it requires more network administration. Each switch needs to be managed separately. Another drawback is that to allow the integration of bare metal servers with VMware NSX for vSphere, multicast is required for the integration of VXLAN and MAC address learning. This means that in addition to building an IP fabric with BGP, you also need to design and manage a multicast infrastructure with Protocol Independent Multicast (PIM) protocols.

The advantage of Virtual Chassis Fabric is that you can build an IP fabric and multicast infrastructure without having to worry about BGP and PIM. Because the Virtual Chassis Fabric behaves like a single, logical switch, you simply create integrated routing and bridging (IRB) interfaces and all traffic is automatically routed between all networks because each network appears as a directly connected network. To enable multicast on a Virtual Chassis Fabric, you simply enable Internet Group Management Protocol (IGMP) on any interfaces requiring multicast. There is no need to design and configure PIM or any other multicast protocols.

Related Documentation

- Understanding MetaFabric Architecture 2.0 on page 5
- Example: Configuring Virtual Chassis Fabric and VMware NSX for MetaFabric Architecture 2.0 on page 19

Example: Configuring Virtual Chassis Fabric and VMware NSX for MetaFabric Architecture 2.0

The power of SDN enabled through Juniper Networks Virtual Chassis Fabric and VMware NSX allows you to quickly build an enterprise private cloud. You can now build multi-tier applications and deploy them within seconds. Virtual Chassis Fabric provides high performance and an easy-to-use network to support SDN with VMware NSX. There is no need to worry about multicast protocols or spanning tree with Virtual Chassis Fabric, because the entire fabric works like a single, logical switch.

This example shows how to configure a QFX5100-only Virtual Chassis Fabric (VCF) and VMware NSX for MetaFabric Architecture 2.0. For more details on the MetaFabric architecture, see the MetaFabric™ Architecture Virtualized Data Center Design and Implementation Guide and MetaFabric™ Architecture 1.1: Configuring Virtual Chassis Fabric and Network Director 1.6.

- Requirements on page 19
- Overview and Topology on page 20
- Configuring a Virtual Chassis Fabric for MetaFabric Architecture 2.0 on page 21
- Configuring VMware NSX for MetaFabric Architecture 2.0 on page 24
- Verification on page 45

Requirements

This example uses the following hardware and software components:

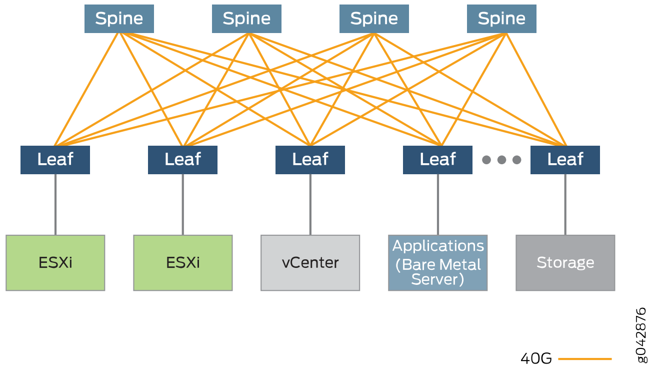
- Four QFX5100-24Q switches used as the spine layer in the VCF
- Six QFX5100-48S switches used in the leaf layer in the VCF
- Junos OS Release 14.1X53-D10 or later for all QFX Series QFX5100 switches participating in the VCF
- VMware ESXi 5.5.0.update2-2068190.x86_64
- VMware vCenter Appliance 5.5.0.20200-2183109_OVF10.ova
- VMware NSX Manager 6.1.0-2107742.ova
- VMware Client Integration Plugin 5.5.0.mac64
- Four servers with Supermicro X9SCM-iiF motherboards, 3.3GHz Intel Xeon E3-1230V2, 32GB Samsung DDR-1600 Memory, and 128GB SSD Crucial M4
- 48TB Synology RS2414(RP)+ and DSM 5.1 U2 for the network-attached storage (NAS) device

Overview and Topology

MetaFabric Architecture 2.0 continues to provide the proper foundation for a virtualized environment that supports virtual machine movement, robust application hosting, and storage in a data center environment. However, this evolving architecture now includes a QFX5100-only VCF and VMware NSX 6.1.0 for virtualization.

The MetaFabric Architecture 2.0 topology used in this example consists of a VCF with 10 members as shown in Figure 15 on page 20.

Figure 15: MetaFabric Architecture 2.0 Topology



There are also the following components:

- Two servers for VMware virtualization (ESXi)
- A separate physical server for VMware vCenter to manage the clusters, virtual machines, and VMware NSX services
- A physical server to host applications that do not support virtualization
- A NAS device using the iSCSI protocol, so that each host has adequate storage for VMs and file storage for images and other media

In this example, a QFX5100-only VCF replaces the mixed-mode VCF seen in the MetaFabric Architecture 1.1 solution. As before, the VCF connects directly to servers and storage on the access side (also known as the *leaf layer* in a VCF), and edge devices on the data center network side (also known as the *spine layer* in a VCF).

The VCF used in this example is a same-mode fabric that implements four QFX5100-24Q switches in the spine layer and six QFX5100-48S switches in the leaf layer for a total of 10 VCF devices. All server, storage, and network destinations are a maximum of two hops from each other to keep latency to a minimum and application performance to a maximum.

The configuration tasks for MetaFabric Architecture 2.0 integrate the VCF with the VMware NSX software suite. This document assumes that you have already installed your VCF and you are ready to begin configuring it. This document also assumes that you are familiar with VMware vSphere, but still new to the VMware NSX software suite.

- For more information about Virtual Chassis Fabric, see Virtual Chassis Fabric or MetaFabric™ Architecture 1.1: Configuring Virtual Chassis Fabric and Network Director 1.6.
- For more information about VMware vSphere, see the VMware vSphere Documentation.

To configure the MetaFabric Architecture 2.0 network, perform the following tasks:

1. Set up your Virtual Chassis Fabric to provide basic IP connectivity.
2. Configure the VMware NSX Manager and integrate it with the VMware vCenter server.
3. Configure the VMware NSX components through the VMware vCenter Web client.
4. Create logical switches inside of VMware NSX to provide the connectivity to the components.
5. Create and configure a VMware NSX Edge Gateway and a VMware NSX LDR.
6. Integrate your Virtual Chassis Fabric with VMware NSX.

Configuring a Virtual Chassis Fabric for MetaFabric Architecture 2.0

The minimal configuration tasks for Virtual Chassis Fabric fall into four areas: VLANs, interfaces, IGMP, and OSPF. One of the benefits of VCF is that you can configure the fabric from the master Routing Engine – a single point of management for all the VCF devices. It is also very easy to configure multicast support in VCF with a single IGMP command. As a result, there is no need to worry about multicast protocols such as Protocol Independent Multicast (PIM).

If you want to provide additional redundancy to the VMware ESXi hosts or physical servers, you can choose as an option to set up IEEE 802.1AC / LACP between physical switches. Because VCF works like a single, logical switch, there is no requirement to set up additional protocols, such as multichassis link aggregation (MC-LAG) or Spanning Tree Protocol (STP).

This example explains how to configure a VCF to support the MetaFabric Architecture 2.0 solution. It includes the following sections:

Configuring VLANS for the VCF

CLI Quick Configuration

To quickly configure VLANs for the VCF, enter the following configuration statements on the device acting in the master role:

```
[edit]
set vlans NSX_UNDERLAY vlan-id 15
set vlans NSX_UNDERLAY description “Default VLAN for VMware ESXi hosts and Synology
storage”
set vlans NSX_UNDERLAY l3-interface irb.15
```

Step-by-Step Procedure

To configure VLANs:

1. Assign VLAN ID 15 to the NSX_UNDERLAY VLAN.

```
[edit vlans]
user@vcf# set NSX_UNDERLAY vlan-id 15
```

2. Add a description for the NSX_UNDERLAY VLAN.

```
[edit vlans]
user@vcf# set vlans NSX_UNDERLAY description “Default VLAN for VMware ESXi hosts
and Synology storage”
```

3. Add interface irb.15 as the Layer 3 IRB interface for the NSX_UNDERLAY VLAN.

```
[edit vlans]
user@vcf# set vlans NSX_UNDERLAY l3-interface irb.15
```

Configuring Interfaces for the VCF

CLI Quick Configuration

To quickly configure interfaces for the VCF, enter the following configuration statements on the device acting in the master role:

```
[edit]
set interfaces irb.15 family inet address 10.0.1.1/24
set interfaces irb.15 mtu 9000
set interfaces lo0.0 family inet address 10.0.0.1/24
set interfaces xe-6/0/4.0 family ethernet-switching interface-mode access
set interfaces xe-6/0/4.0 family ethernet-switching vlan members NSX_UNDERLAY
set interfaces xe-6/0/4.0 mtu 9216
set interfaces xe-7/0/4.0 family ethernet-switching interface-mode access
set interfaces xe-7/0/4.0 family ethernet-switching vlan members NSX_UNDERLAY
set interfaces xe-7/0/4.0 mtu 9216
set interfaces xe-7/0/5.0 family ethernet-switching interface-mode access
set interfaces xe-7/0/5.0 family ethernet-switching vlan members NSX_UNDERLAY
```

To configure the interfaces:

1. Configure interface irb.15 as the Layer 3 integrated routing and bridging (IRB) interface for the NSX_UNDERLAY VLAN.

   It acts as the default gateway for all hosts and storage devices.

```
[edit interfaces]
user@vcf# set irb.15 family inet address 10.0.1.1/24
```

2. Configure loopback interface lo0.

```
[edit interfaces]
user@vcf# set lo0.0 family inet address 10.0.0.1/24
```

3. Configure three interfaces as access ports.

```
[edit interfaces]
user@vcf# set xe-6/0/4.0 family ethernet-switching interface-mode access
user@vcf# set xe-7/0/4.0 family ethernet-switching interface-mode access
user@vcf# set xe-7/0/5.0 family ethernet-switching interface-mode access
```

4. Assign the three interfaces to the NSX_UNDERLAY VLAN.

```
[edit interfaces]
user@vcf# set xe-6/0/4.0 family ethernet-switching vlan members NSX_UNDERLAY
user@vcf# set xe-7/0/4.0 family ethernet-switching vlan members NSX_UNDERLAY
user@vcf# set xe-7/0/5.0 family ethernet-switching vlan members NSX_UNDERLAY
```

5. Increase the maximum transmission unit (MTU) beyond the default value of 1,500 bytes.

   Because there will be VXLAN encapsulated traffic flowing between VMware ESXi servers, you must select a larger MTU to accommodate for the outer MAC address, UDP header, IP header, and VXLAN header. VCF supports Jumbo Frames, so set the MTU over 9,000 bytes.

```
[edit interfaces]
user@vcf# set irb.15 mtu 9000
user@vcf# set xe-6/0/4.0 mtu 9216
user@vcf# set xe-7/0/4.0 mtu 9216
```

Configuring IGMP for the VCF

CLI Quick Configuration

VMware NSX uses multicast for flooding broadcast, unknown unicast, and multicast traffic. As a result, you must configure Internet Group Management Protocol (IGMP) when integrating physical servers with the VMware NSX virtual networks, so that the flooding of traffic can extend into the VCF.

To quickly configure IGMP for the VCF, enter the following configuration statements on the device acting in the master role:

```
[edit]
set protocols igmp interface xe-6/0/4.0
set protocols igmp interface xe-7/0/4.0
set protocols igmp interface irb.15
```

To configure IGMP:

1. Configure IGMP on selected interfaces so that the hosts can signal their interest in multicast groups.

```
[edit protocols igmp]
user@vcf# set interface xe-6/0/4.0
user@vcf# set interface xe-7/0/4.0
user@vcf# set interface irb.15
```

Configuring OSPF for the VCF

CLI Quick Configuration

To quickly configure OSPF for the VCF, enter the following configuration statements on the device acting in the master role:

```
[edit]
set protocols ospf area 0.0.0.0 interface irb.15
set protocols ospf area 0.0.0.0 interface lo0.0
```

To configure OSPF:

1. Configure OSPF on the loopback and IRB interfaces so that the VMs and servers can communicate across the VCF at Layer 3.

   ```
   [edit protocols ospf]
   user@vcf# set area 0.0.0.0 interface irb.15
   user@vcf# set area 0.0.0.0 interface lo0.0
   ```

Configuring VMware NSX for MetaFabric Architecture 2.0

This portion of the example explains the components required to install and configure VMware NSX to work with the MetaFabric Architecture 2.0 solution. These components include:

- Integrating the VMware NSX Manager into the VMware vCenter Server. This step provides connectivity so the VMware NSX can be managed through the VMware vCenter web client.
- Setting up the basic logical switches, transport zones, and segment IDs for VXLAN.
- Configuring the VMware NSX Edge Gateway and Logical Distributed Router (LDR) to provide virtual connectivity between the VMware ESXi hosts and the physical network.

This example includes the following sections:

Configuring the ESXi Hosts

Step-by-Step Procedure

Configure the following ESXi hosts:

1. **esxi-01**—A Supermicro server that is compatible with VMware software. Configure the vKernel management IP address for esxi-01 as 10.0.1.140. When you install the VMware NSX components, place the NSX Manager and NSX Edge on this host. When all components have been configured, create an example application on this host with a Web server, an application server, and a database server. All of the servers are deployed in pairs, with one VM per host.

2. **esxi-02**—A host that is exactly the same as the esxi-01 host running on Supermicro hardware. Deploy the VMware NSX Controller and Edge Gateway on this host to balance your network. The other half of the example servers run on this host as well. Configure the vKernel management IP address for esxi-02 as 10.0.1.141.

3. **vcenter**—A separate VMware vCenter server that is used to manage esxi-01 and esxi-02. Although you can run a nested VMware vCenter server on the same hosts that are being managed, it is best to keep them separate to avoid any confusion and reduce troubleshooting in the future. Configure the VMware vCenter server with an IP address of 10.0.1.110.

4. **storage-01**—A Synology NAS device. The ESXi hosts esxi-01 and esxi-02 use iSCSI to mount storage remotely on this device. Configure the IP address 10.0.1.40 on this device to provide management and iSCSI connectivity.

Results

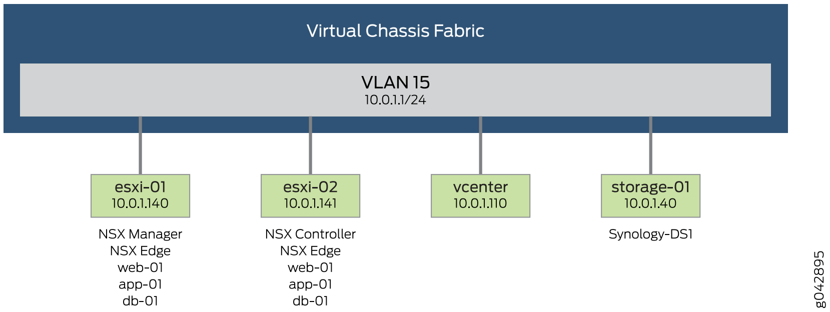
In summary, the physical IP address assignments for servers and storage in this example are shown in Table 5 on page 25.

Table 5: IP Address Assignments

Device	IP Address
esxi-01	10.0.1.140
esxi-02	10.0.1.141
vcenter	10.0.1.110
storage-01	10.0.1.40

A graphical representation of the hosts and appliances are shown in Figure 16 on page 26.

Figure 16: Virtual Chassis Fabric and VMware NSX IP Addressing



Installing VMware NSX

GUI Step-by-Step Procedure

To install VMware NSX:

1. Deploy the VMware-NSX-Manager-6.1.0-2107742.ova as a new template by logging in to VMware vCenter Web client, clicking **Deploy OVT template**, and specifying the VMware-NSX-Manager-6.1.0-2107742.ova file.
2. Go through the installation steps to accept the EULA, set a password, and specify a hostname.
3. For the network settings, configure an IP address of 10.0.1.111 for the VMware NSX Manager.

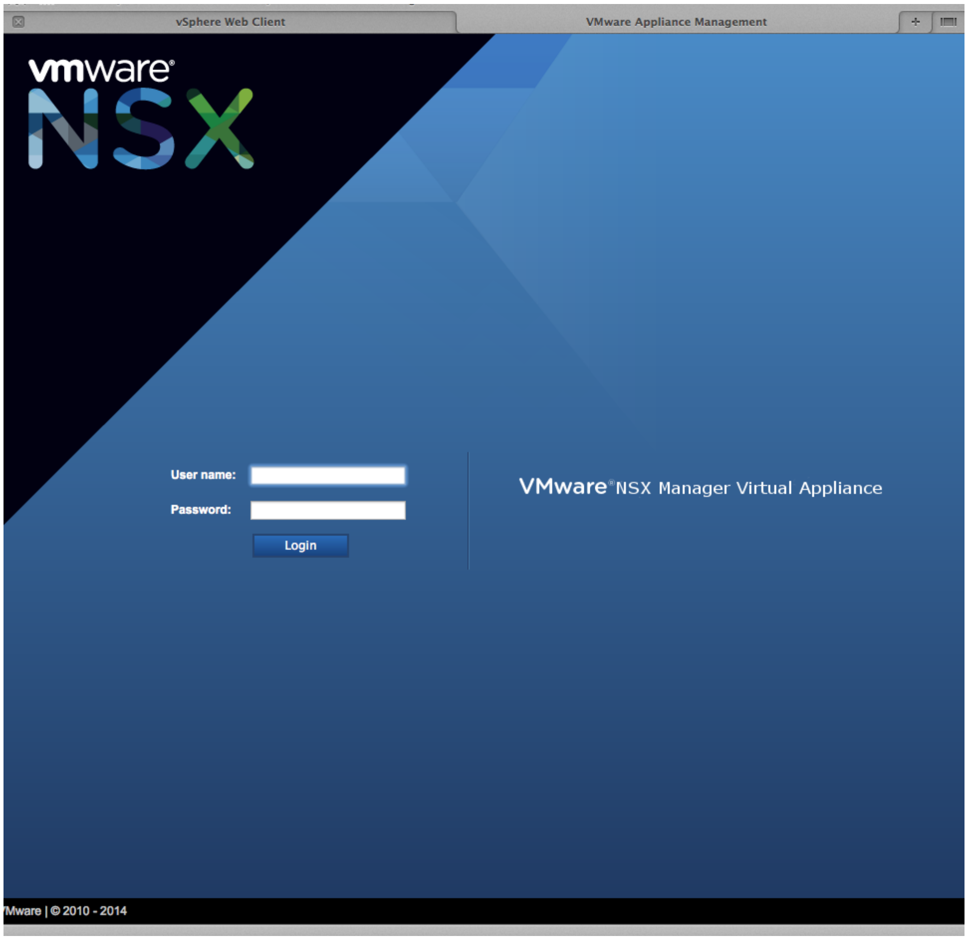
Integrating VMware NSX Manager

GUI Step-by-Step Procedure

After you deploy the OVT template successfully, the VMware NSX Manager starts automatically. To integrate VMware NSX Manager into your network:

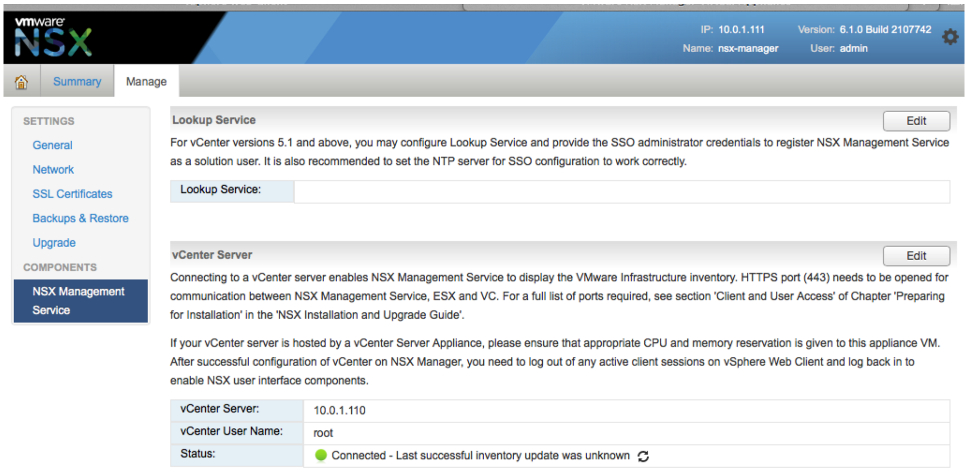
1. Log in to the Web client at http://10.0.1.111 as shown in Figure 17 on page 27.

Figure 17: Network Director 1.6 — VCF Autoprovisioning



2. Configure a username of **admin**, and enter the same password that you specified during the creation of the OVT template.
3. Log in to the VMware Manager Appliance and integrate it with the VMware vCenter Server.
4. After you log in, click **Manage Application Settings**, then select **NSX Management Service**, and click **Configure**.
5. Type the IP address of the VMware vCenter Server, which in this example is 10.0.1.110, and click **OK** to make sure that the status appears as **Connected** as shown in Figure 18 on page 28.

Figure 18: Integrating VMware NSX Manager with VMware vCenter Server



IP: 10.0.1.111 Version: 6.1.0 Build 2107742
Name: nsx-manager User: admin

Lookup Service

For vCenter versions 5.1 and above, you may configure Lookup Service and provide the SSO administrator credentials to register NSX Management Service as a solution user. It is also recommended to set the NTP server for SSO configuration to work correctly.

Lookup Service:	

vCenter Server

Connecting to a vCenter server enables NSX Management Service to display the VMware Infrastructure inventory. HTTPS port (443) needs to be opened for communication between NSX Management Service, ESX and VC. For a full list of ports required, see section 'Client and User Access' of Chapter 'Preparing for Installation' in the 'NSX Installation and Upgrade Guide'.

If your vCenter server is hosted by a vCenter Server Appliance, please ensure that appropriate CPU and memory reservation is given to this appliance VM. After successful configuration of vCenter on NSX Manager, you need to log out of any active client sessions on vSphere Web Client and log back in to enable NSX user interface components.

vCenter Server:	10.0.1.110
vCenter User Name:	root
Status:	● Connected - Last successful inventory update was unknown

Installing the VMware NSX Controller

GUI Step-by-Step Procedure

To install the VMware NSX Controller:

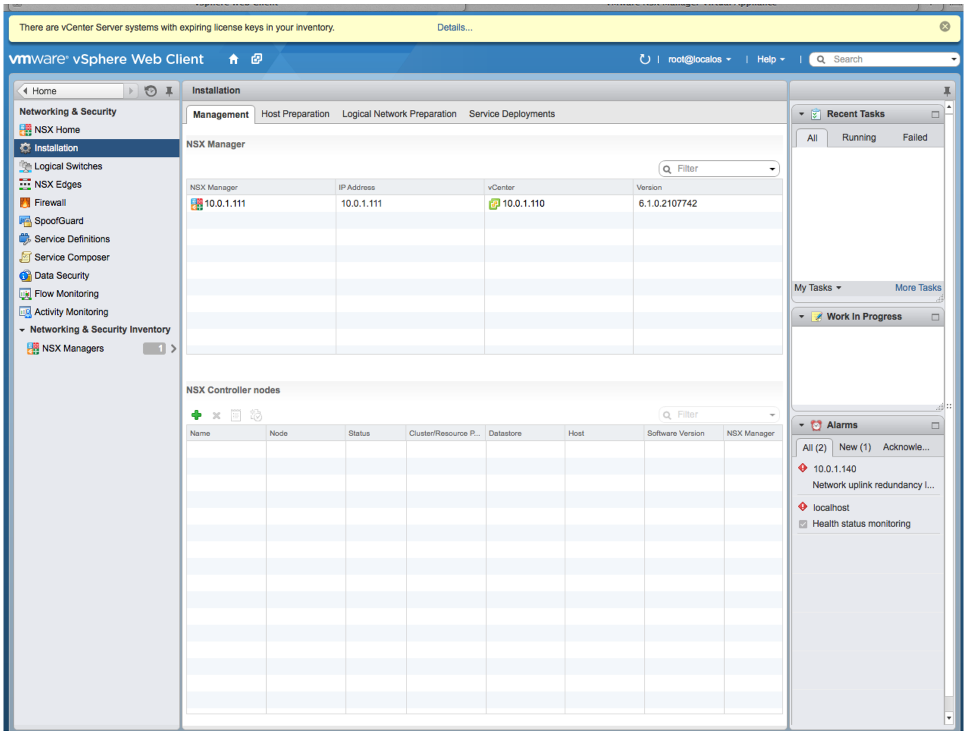
1. Log In to the VMware vCenter Web client.

   You should see a new management pane on the left called **Networking & Security**. This pane is where you provision and manage all VMware NSX tasks.

2. Install the VMware NSX Controller.

   By default, no controllers are installed as shown in Figure 19 on page 29.

Figure 19: VMware NSX Controller Nodes



There are vCenter Server systems with expiring license keys in your inventory. Details...

NSX Manager

NSX Manager	IP Address	vCenter	Version
10.0.1.111	10.0.1.111	10.0.1.110	6.1.0.2107742

NSX Controller nodes

Name	Node	Status	Cluster/Resource P...	Datastore	Host	Software Version	NSX Manager

10.0.1.140
Network uplink redundancy l...

localhost
Health status monitoring

3. Install a new VMware NSX Controller by clicking the green **+** symbol, select a cluster and data store for the new VMware NSX Controller appliance, and click **Next**.

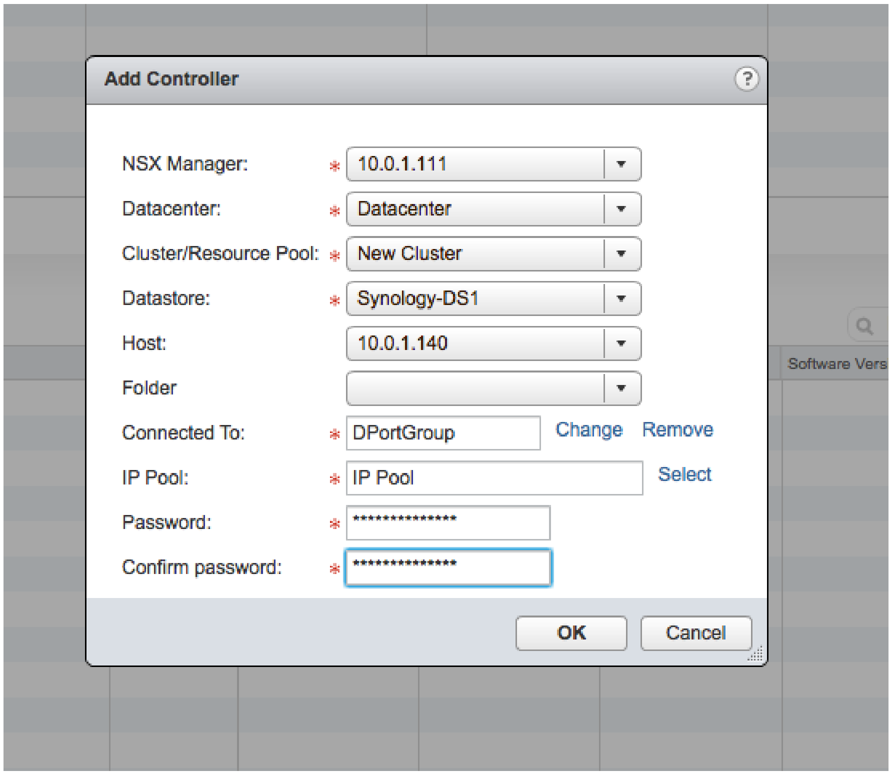
4. Set up an IP address pool to be used for VMware NSX IP address assignments.

   In this case, use the IP range of 10.0.1.200 - 10.0.1.219.

5. Select the virtual switch that the VMware NSX Controller will use for connectivity.

   This example uses the new distributed virtual switch **DPortGroup** as shown in Figure 20 on page 30.

Figure 20: Adding the VMware NSX Controller



Add Controller

NSX Manager: * 10.0.1.111

Datacenter: * Datacenter

Cluster/Resource Pool: * New Cluster

Datastore: * Synology-DS1

Host: 10.0.1.140

Folder

Connected To: * DPortGroup

IP Pool: * IP Pool

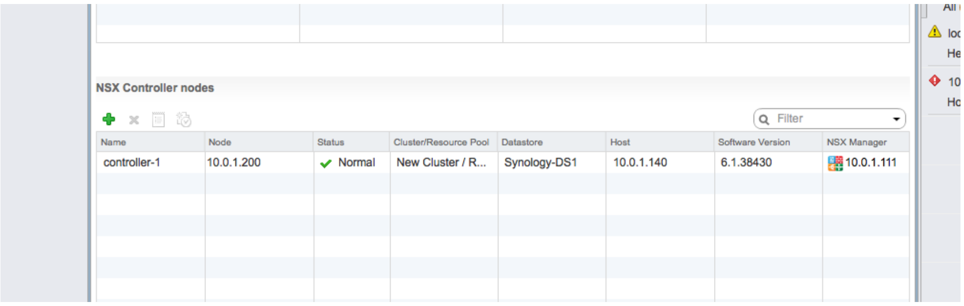
Password: * ***************

Confirm password: * ***************

6. When you have completed entering the resource selection, virtual switch, IP pool, and password, click **OK**.

   When the VMware NSX Controller is installed correctly, you should see it listed in the **NSX Controller nodes** section as shown in Figure 21 on page 30.

Figure 21: VMware NSX Controller Installed Successfully



NSX Controller nodes

Name	Node	Status	Cluster/Resource Pool	Datastore	Host	Software Version	NSX Manager
controller-1	10.0.1.200	✓ Normal	New Cluster / R...	Synology-DS1	10.0.1.140	6.1.38430	10.0.1.111

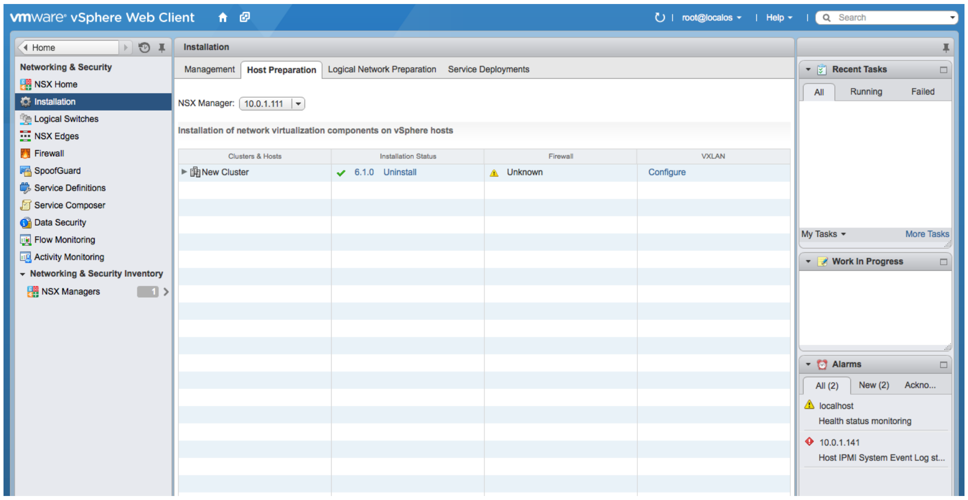
Configuring VXLAN Transport

GUI Step-by-Step Procedure

To configure VXLAN transport:

1. Navigate back to the **Network & Security** page, click **Installation**, look for the **Host Preparation** tab, click the **Configure** button for the **New Cluster**, and begin the VXLAN transport configuration as shown in Figure 22 on page 31.

Figure 22: Host Preparation and Installation



NSX Manager: 10.0.1.111

Installation of network virtualization components on vSphere hosts

Clusters & Hosts	Installation Status	Firewall	VXLAN
New Cluster	✓ 6.1.0 Uninstall	⚠ Unknown	Configure

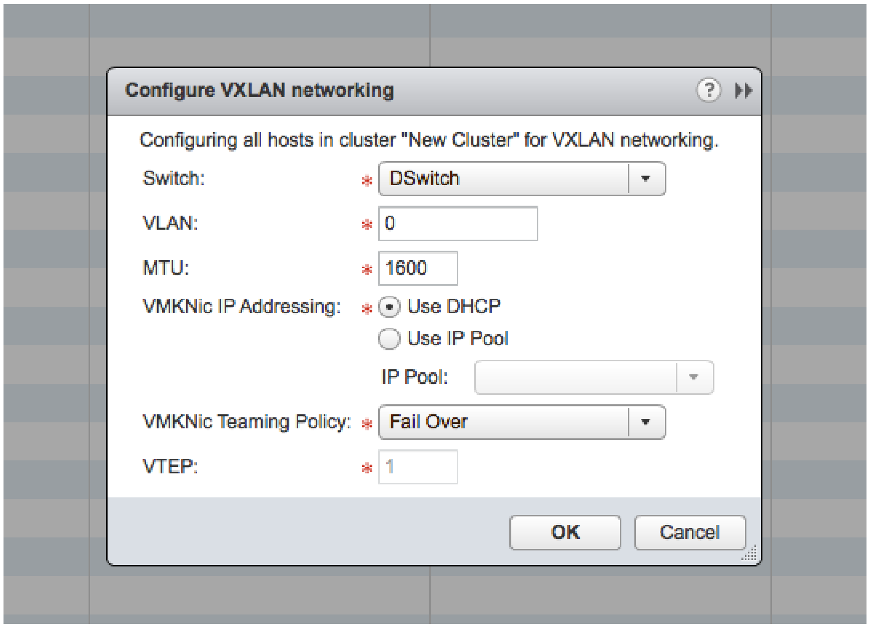
⚠ localhost
Health status monitoring

10.0.1.141
Host IPMI System Event Log st...

2. Define which virtual switch the cluster uses for VXLAN networking.

   In this example, select the default distributed virtual switch **Dswitch** as shown in Figure 23 on page 31.

Figure 23: Configure VXLAN Networking



Configure VXLAN networking

Configuring all hosts in cluster "New Cluster" for VXLAN networking.

Switch: *	DSwitch
VLAN: *	0
MTU: *	1600
VMKNic IP Addressing: *	● Use DHCP ○ Use IP Pool
IP Pool:	
VMKNic Teaming Policy: *	Fail Over
VTEP: *	1

Set the MTU to at least 1600 to account for the additional 50 bytes for VXLAN. Use the same previous IP pool that you created earlier to configure VXLAN networking as well. When you have finished entering these values, click **OK**.

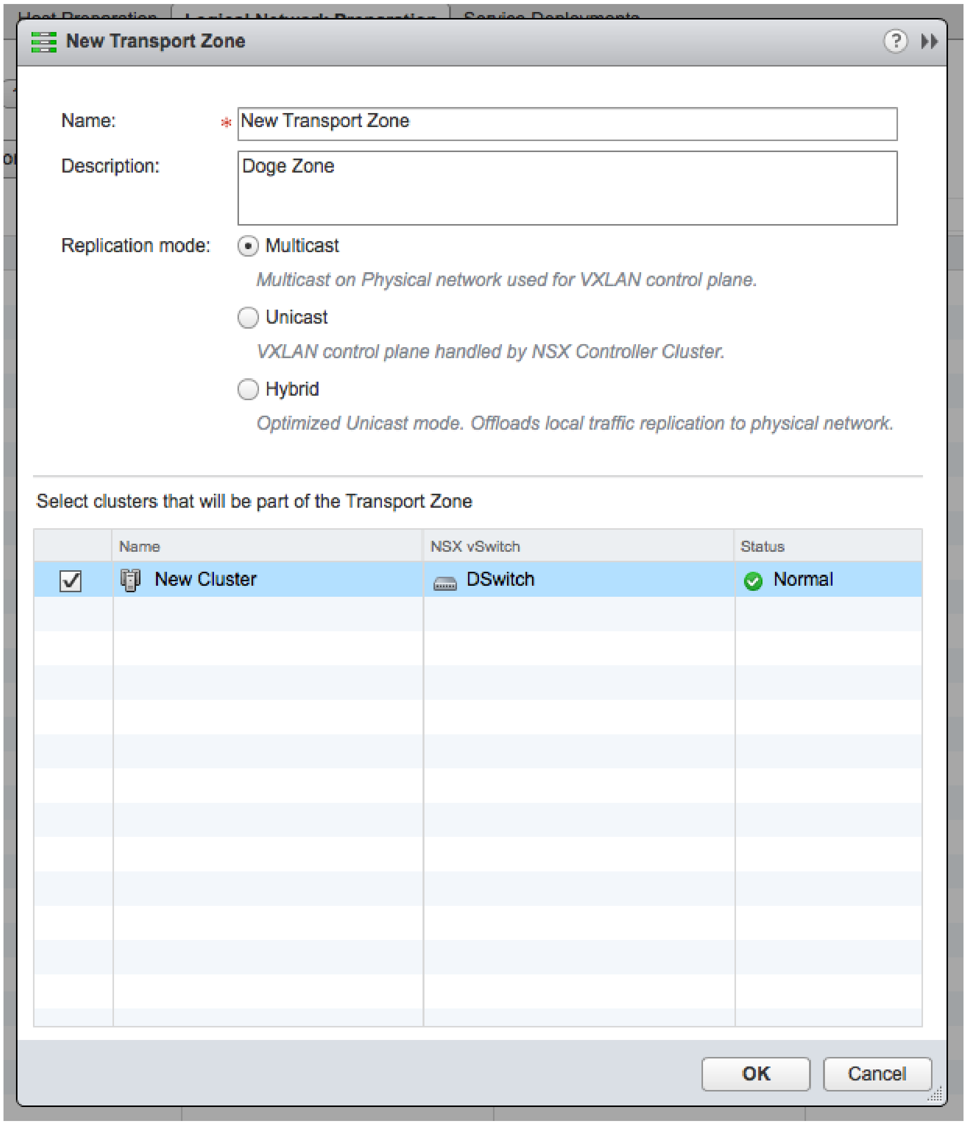
3. Add a new transport zone for VXLAN by going back to the **Networking & Security** page and clicking **Logical Network Preparation**.

   You should see a tab called **Transport Zones**.

4. Click the **New Transport Zone** button.

   As shown in Figure 24 on page 32, use the **Multicast** option for **Replication mode** so that the VCF can handle the replication and MAC address learning tasks.

Figure 24: New Transport Zone



New Transport Zone

Name: * New Transport Zone

Description: Doge Zone

Replication mode:
● Multicast
Multicast on Physical network used for VXLAN control plane.
○ Unicast
VXLAN control plane handled by NSX Controller Cluster.
○ Hybrid
Optimized Unicast mode. Offloads local traffic replication to physical network.

Select clusters that will be part of the Transport Zone

	Name	NSX vSwitch	Status
☑	New Cluster	DSwitch	✓ Normal

NOTE: A transport zone is nothing but an abstract zone that defines how VMware NSX handles MAC address learning. Generally, a single transport zone is sufficient for a small or medium enterprise private cloud. However, if you want to build a scale-out architecture, it is a good idea to create one transport zone per POD.

Configuring a Segment ID

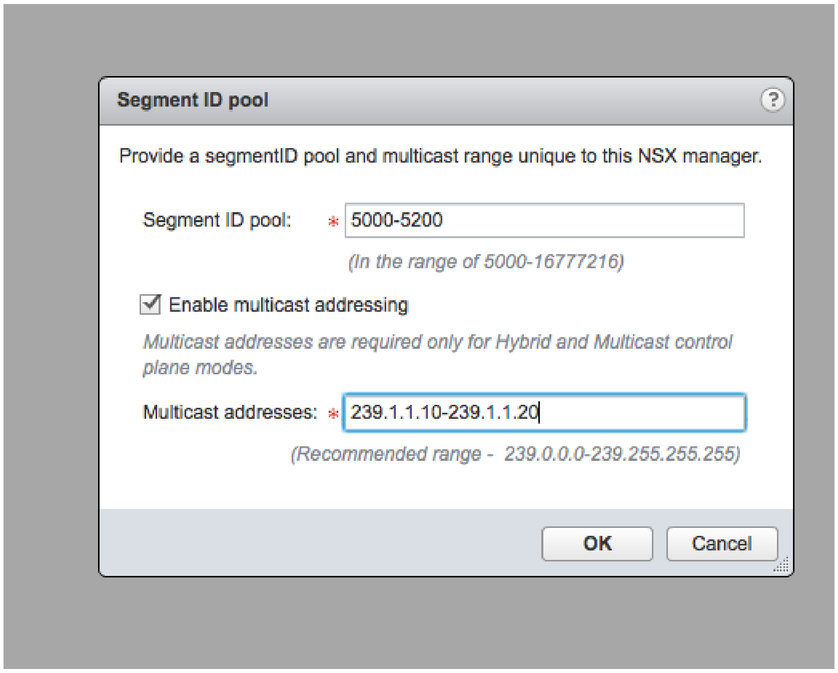
GUI Step-by-Step Procedure

To configure a segment ID:

1. Add a VXLAN Segment ID and Multicast Address pool.

   As you create new logical switches (VXLANs), the segment ID (VNI) and multicast address are assigned automatically from a pool as shown in Figure 25 on page 33.

Figure 25: Segment ID Pool



Segment ID pool

Provide a segmentID pool and multicast range unique to this NSX manager.

Segment ID pool: * 5000-5200

(In the range of 5000-16777216)

☑ Enable multicast addressing

Multicast addresses are required only for Hybrid and Multicast control plane modes.

Multicast addresses: * 239.1.1.10-239.1.1.20

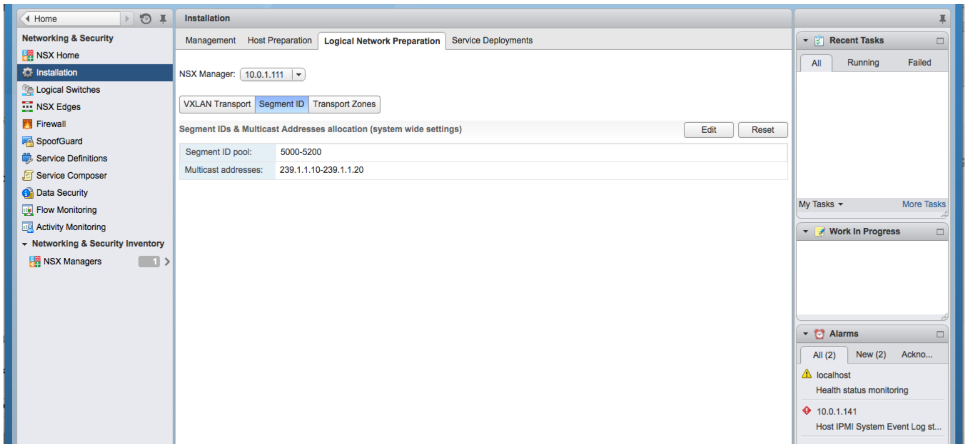
(Recommended range - 239.0.0.0-239.255.255.255)

In this example, create a segment ID pool in the range of 5000-5200. Also, check the box to enable multicast addressing. The multicast addresses in our example are in the range of 239.1.1.10 to 239.1.1.20.

NOTE: If you plan to implement this feature in a production environment, you need to create a larger multicast address pool than the one shown in this example.

2. After you create the segment ID and multicast address pool, you should see a summary as shown in Figure 26 on page 34.

Figure 26: VXLAN Segment ID and Multicast Address Allocation



NSX Manager: 10.0.1.111

Segment IDs & Multicast Addresses allocation (system wide settings)

Segment ID pool: 5000-5200

Multicast addresses: 239.1.1.10-239.1.1.20

localhost
Health status monitoring

10.0.1.141
Host IPMI System Event Log st...

Configuring Logical Switches

GUI Step-by-Step Procedure

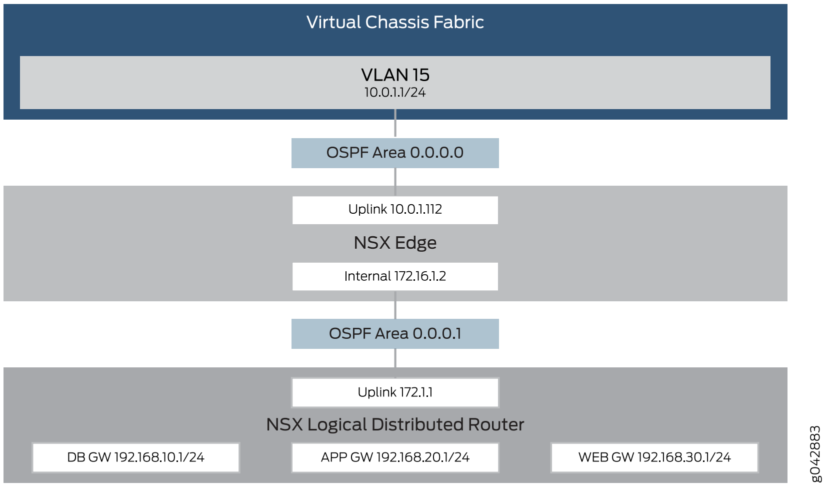
Before you create the VMware NSX Edge Gateway and LDR, you need to create the logical switches that the appliances use. You must configure four logical switches as shown in Table 6 on page 34.

Table 6: Logical Switch Settings

Name	VNI	Multicast Group	Transport Zone
Uplink Logical Switch	5000	239.1.1.10	Transport Zone 1
Database Switch	5001	239.1.1.11	Transport Zone 1
Application Switch	5002	239.1.1.12	Transport Zone 1
Web Switch	5003	239.1.1.13	Transport Zone 1

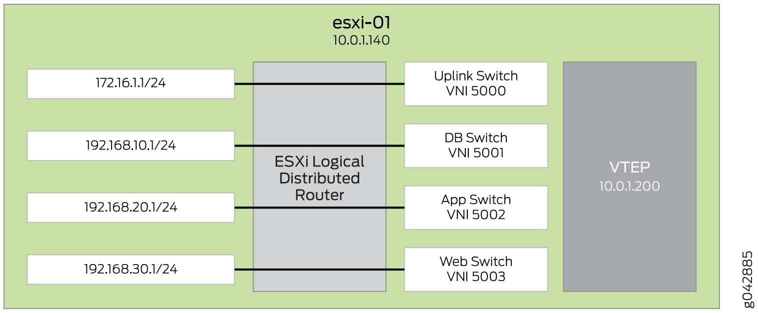
These four logical switches enable you to create the logical topology shown in Figure 27 on page 35. The Uplink Logical Switch is used between the VMware NSX Edge Gateway and VMware NSX LDR. The database, application, and web logical switches are used by the VMware NSX LDR for our example application. This enables you to create a 3-tier application with network segmentation easily.

Figure 27: Logical Topology of Juniper Networks and VMware Components



All of the VMware NSX virtual switches are associated with a VNI as shown in Figure 28 on page 35. Each hypervisor has a virtual tunnel end-point (VTEP) which is responsible for encapsulating VM traffic inside of a VXLAN header and routing the packet to a destination VTEP for further processing.

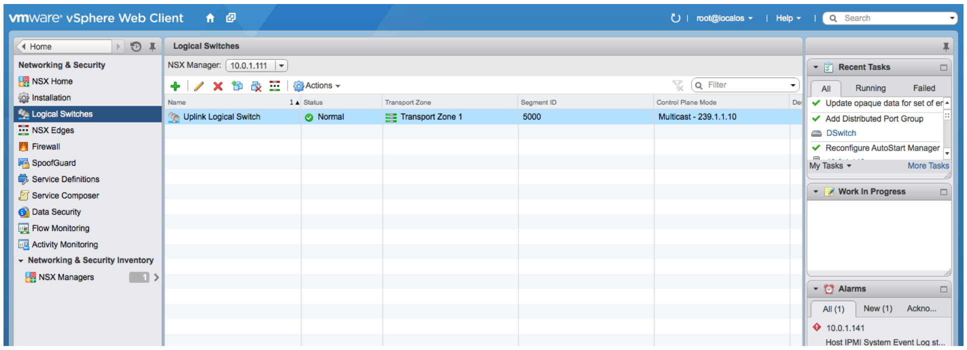
Figure 28: VMware NSX Logical Switches and VTEPs



To configure logical switches:

1. Navigate back to the **Networking & Security** page and click **Logical Switches** as shown in Figure 29 on page 36.

Figure 29: Adding new Logical Switches



NSX Manager: 10.0.1.111

Name	Status	Transport Zone	Segment ID	Control Plane Mode	De
Uplink Logical Switch	Normal	Transport Zone 1	5000	Multicast - 239.1.1.10	

2. Add and configure each logical switch as shown in Table 6 on page 34.

   Do not assign the segment ID or multicast group, as the segment ID and multicast group pool automatically assigns these values for each new logical switch. However, to keep the values the same as shown in Table 6 on page 34, create the following logical switches in order:

   1. Uplink Logical Switch
   2. Database Logical Switch
   3. Application Logical Switch
   4. Web Logical Switch

   When you finish this task, you can create the VMware NSX Edge Gateway and LDR using the newly created logical switches.

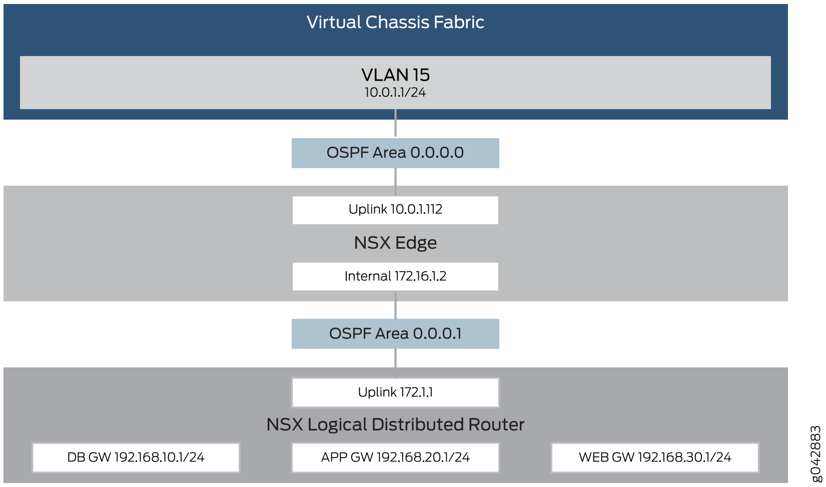
Configuring the VMware NSX Edge Gateway

GUI Step-by-Step Procedure

Because the physical topology and addressing have been resolved, you can begin to implement the logical topology and integrate the VCF with VMware NSX for vSphere. You need a logical gateway between the physical network and the logical networks in this example. The gateway acts as a logical edge router to provide a routing and security policy between the physical and virtual resources.

The VMware NSX Edge Gateway requires two interfaces. The first interface is an **Uplink** with an IP address of 10.0.1.112 as shown in Figure 30 on page 37.

Figure 30: Logical Topology of Juniper Networks and VMware Components



Any traffic that needs to enter or leave the virtual networks created by VMware NSX must transit through the VMware NSX Edge Gateway **Uplink** interface and security policies. The **Uplink** interface also enables the OSPF routing protocol so that any virtual networks created by the NSX Logical Distributed Router (LDR) can be advertised to the physical network. For the purposes of this example, use the standard OSPF backbone Area 0 between the irb.15 interface of the VCF and the VMware NSX Edge Gateway **Uplink** interface.

The second VMware NSX Edge Gateway interface is the **Internal** interface that connects to the VMware NSX LDR. Configure the **Internal** interface for OSPF Area 1. Any virtual networks created by the VMware NSX LDR are advertised directly to the **Internal** interface, and then sent to the VCF.

Table 7 on page 37 shows the associated values for both the **Uplink** and **Internal** interfaces.

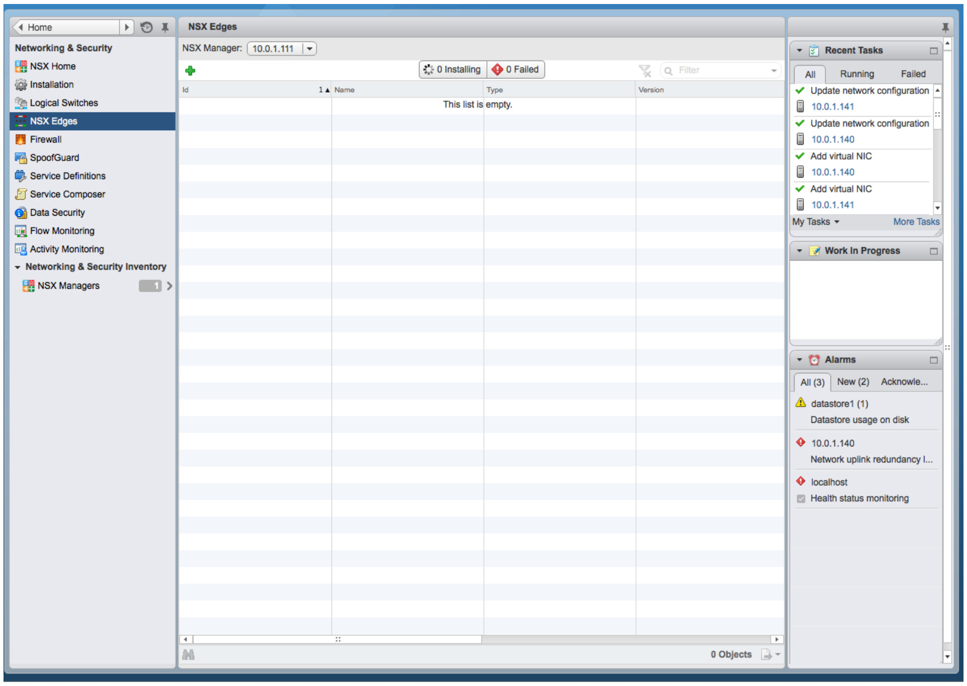
Table 7: VMware NSX Edge Gateway Virtual Switches

Interface	Virtual Switch	IP Address	VNI	Multicast Group
Uplink	DPortGroup	10.0.1.112/24	–	–
Internal	Uplink Logical Switch	172.16.1.2/24	5000	239.1.1.10

To configure the VMware NSX Edge Gateway:

1. Return to the **Networking & Security** page and click **NSX Edges** as shown in Figure 31 on page 38.

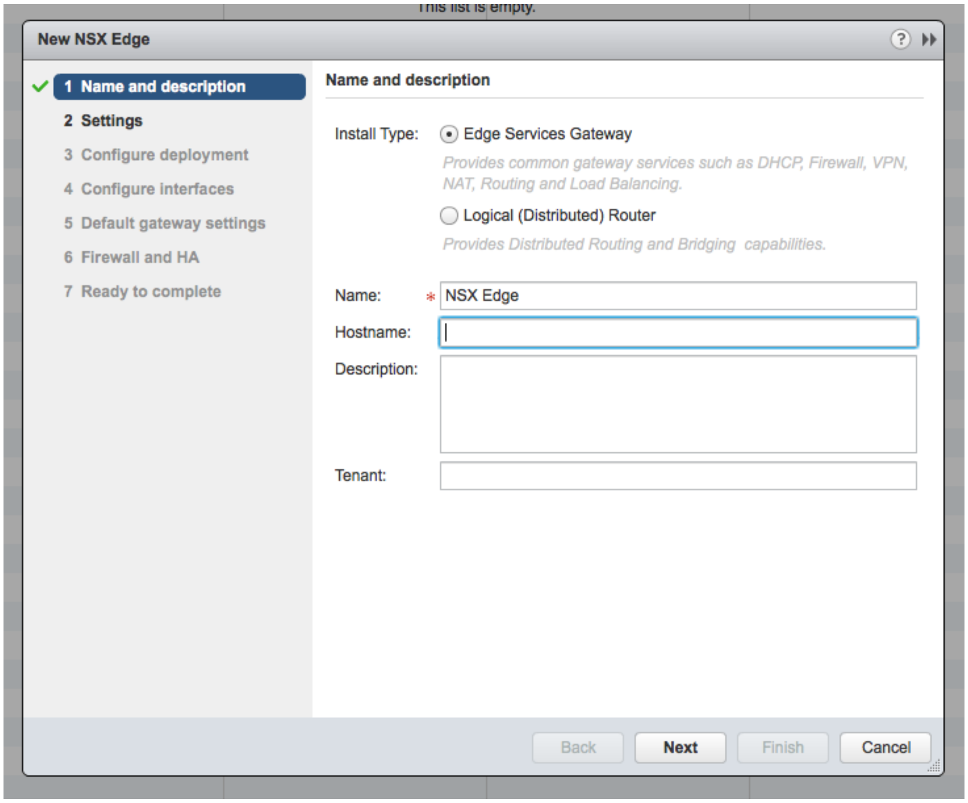
Figure 31: - VMware NSX Edges


NSX Manager: 10.0.1.111

This list is empty.

2. Click the green **+** icon to create a new VMware NSX Edge Gateway as shown in Figure 32 on page 39, give the new appliance a name, and click **Next**.

Figure 32: New NSX Edge



Name and description

Install Type: ● Edge Services Gateway

Provides common gateway services such as DHCP, Firewall, VPN, NAT, Routing and Load Balancing.

○ Logical (Distributed) Router

Provides Distributed Routing and Bridging capabilities.

Name: * NSX Edge

Hostname:

Description:

Tenant:

3. Configure the deployment options.

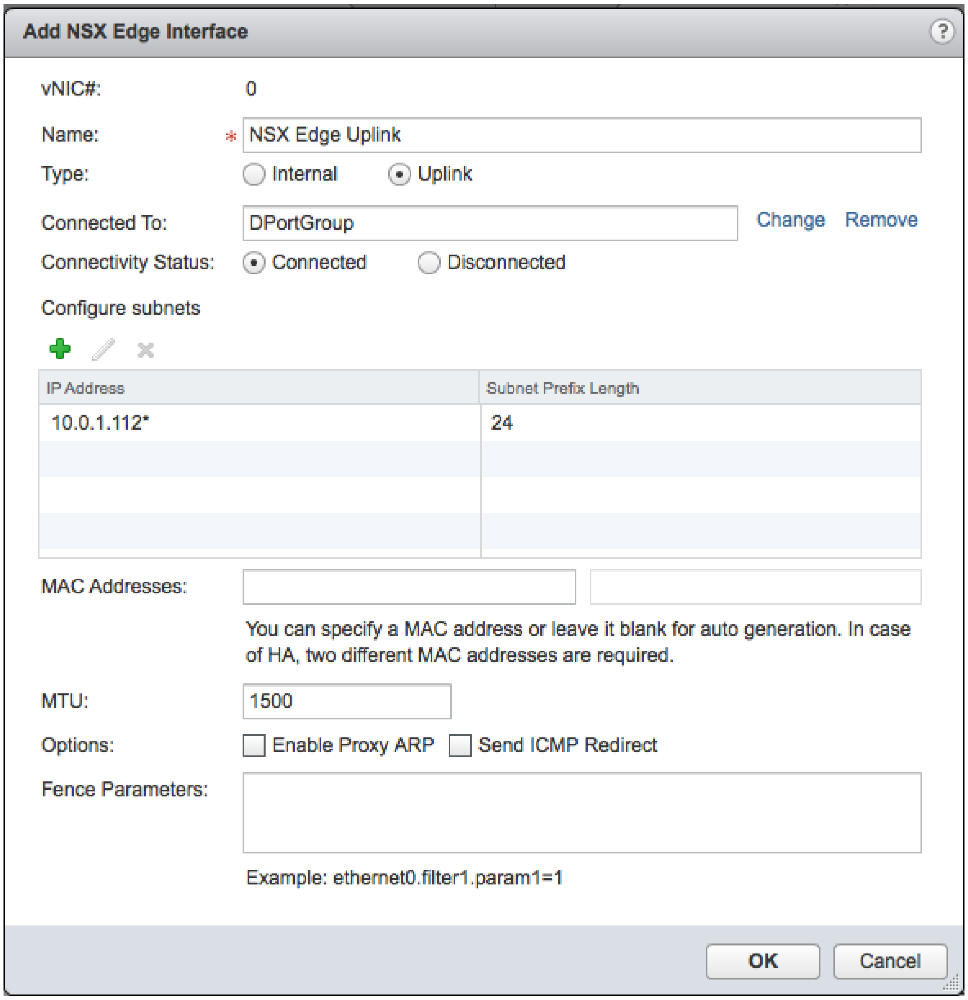
   In this example, use a compact appliance size.

NOTE: Check the VMware NSX documentation to see which appliance size suites your production data center depending on the scale and performance.

4. Configure the uplink interface – the first of two interfaces for VMware NSX Edge Gateway – by placing this interface into the **DPortGroup** as shown in Figure 33 on page 40.

   The **NSX Edge Uplink** interface communicates with the VCF.

Figure 33: Add VMware NSX Edge Interfaces



Add NSX Edge Interface

vNIC#: 0

Name: * NSX Edge Uplink

Type: ○ Internal ● Uplink

Connected To: DPortGroup

Connectivity Status: ● Connected ○ Disconnected

Configure subnets

IP Address	Subnet Prefix Length
10.0.1.112*	24

MAC Addresses:

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

MTU: 1500

Options: □ Enable Proxy ARP □ Send ICMP Redirect

Fence Parameters:

Example: ethernet0.filter1.param1=1

5. Click the green **+** symbol to add new interfaces, name the first interface as **NSX Edge Uplink**, and click the next green **+** symbol to add a new subnet.

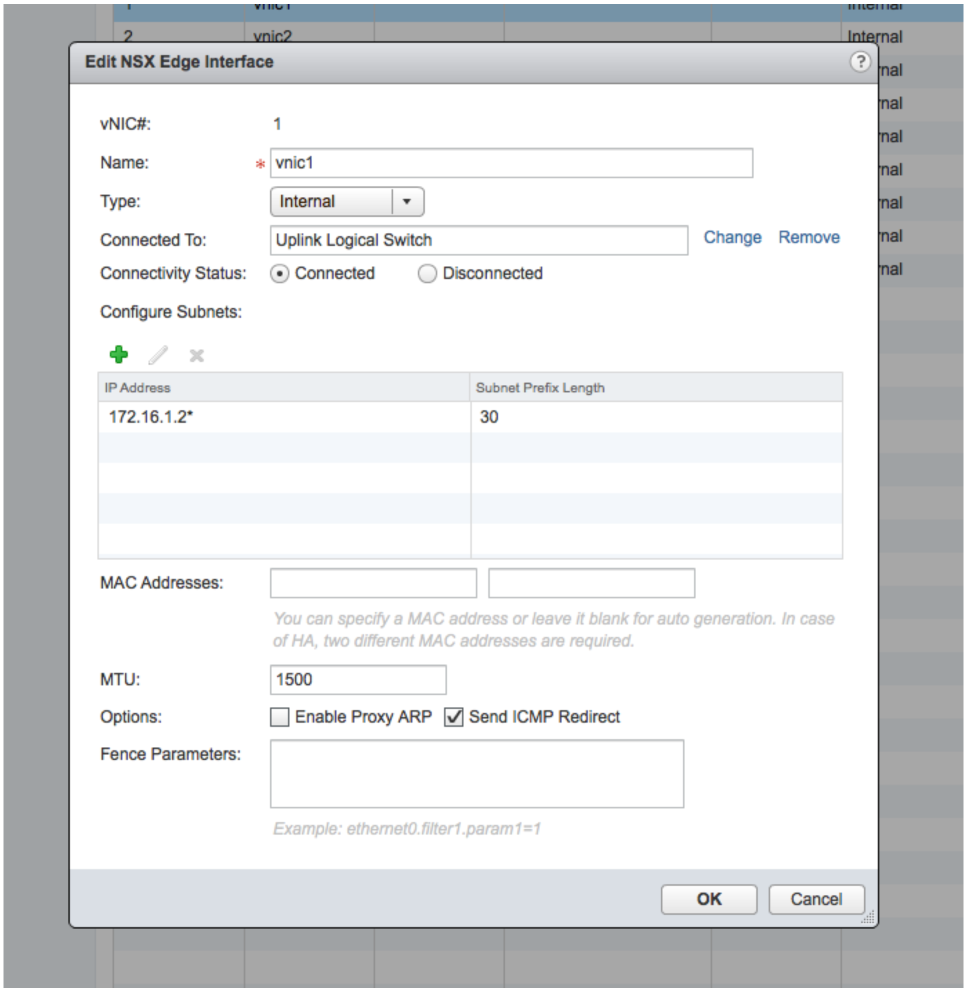
   For this example, you need the uplink interface to use OSPF to connect with the VCF.

6. To establish base IP connectivity, assign an IP address of 10.0.1.112/24.

7. Perform the same actions you did in Step 4 to create a second VMware NSX Edge Gateway interface that connects with the south-bound VMware NSX LDR, and call this the **Internal** interface.

   It must connect to the **Uplink Logical Switch** that you created earlier, and is shown in Figure 34 on page 41.

Figure 34: Internal VMware NSX Edge Interface



Edit NSX Edge Interface

vNIC#: 1

Name: * vnic1

Type: Internal

Connected To: Uplink Logical Switch

Connectivity Status: ● Connected ○ Disconnected

Configure Subnets:

IP Address	Subnet Prefix Length
172.16.1.2*	30

MAC Addresses:

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

MTU: 1500

Options: □ Enable Proxy ARP ☑ Send ICMP Redirect

Fence Parameters:

Example: ethernet0.filter1.param1=1

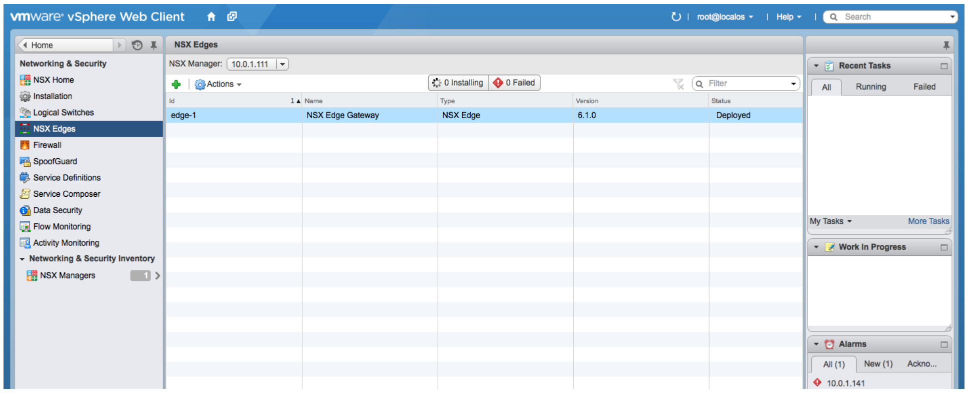
8. Click the green **+** symbol to create a new subnet and configure the IP address as 172.16.1.2/24 (per the VMware NSX logical design in Figure 30 on page 37).

   This address connects to the VMware NSX LDR, which you will configure in the next procedure.

9. Deploy the new VMware NSX Edge Gateway.

   After installation it should be deployed as shown in Figure 35 on page 42.

Figure 35: VMware NSX Edge Deployed



NSX Manager: 10.0.1.111

Id	Name	Type	Version	Status
edge-1	NSX Edge Gateway	NSX Edge	6.1.0	Deployed

Configuring the VMware NSX Logical Distributed Router

GUI Step-by-Step Procedure

To configure the VMware NSX Logical Distributed Router (LDR):

1. Use the same procedure you used to install the VMware NSX Edge by returning to the **Network & Security** page, clicking **NSX Edges**, and clicking the green **+** symbol to create a new VMware NSX Edge for the VMware NSX LDR.
2. Add the interfaces according to the information in Table 8 on page 42 and Table 9 on page 42.

Table 8: VMware NSX LDR Virtual Switches

Interface	Virtual Switch	IP Address	VNI	Multicast Group
Uplink	Uplink Logical Switch	172.16.1.1/24	5000	239.1.1.10
vnic10	Database Switch	192.168.10.1/24	5001	239.1.1.11
vnic11	Application Switch	192.168.20.1/24	5002	239.1.1.12
vnic12	Web Switch	192.168.30.1/24	5003	239.1.1.13

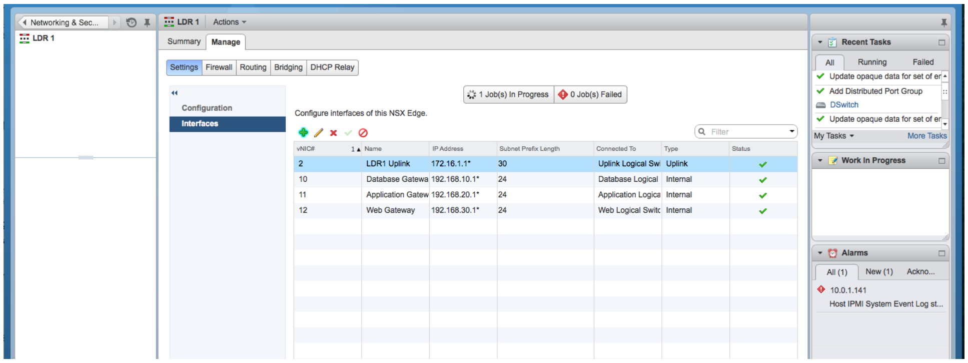
Table 9: VMware NSX LDR Interface Settings

Name	IP Address	Subnet Mask	Virtual Switch	Type
LDR1 Uplink	172.16.1.1	24	Uplink Logical Switch	Uplink
Database Gateway	192.168.10.1	24	Database Logical Switch	Internal
Application Gateway	192.168.20.1	24	Application Logical Switch	Internal
Web Gateway	192.168.30.1	24	Web Gateway	Internal

The database, application, and web gateways are the default gateway addresses for the VMs. The **LDR1 Uplink** acts as a transit interface to the VMware NSX Edge Gateway for connectivity outside of the VMware NSX environment.

3. After the interfaces are configured, you should see the interface summary on the **Manage** tab as shown in Figure 36 on page 43.

Figure 36: VMware NSX LDR Interfaces



Configure interfaces of this NSX Edge.

vNIC#	Name	IP Address	Subnet Prefix Length	Connected To	Type	Status
2	LDR1 Uplink	172.16.1.1*	30	Uplink Logical Swi	Uplink	✓
10	Database Gatewa	192.168.10.1*	24	Database Logical	Internal	✓
11	Application Gatew	192.168.20.1*	24	Application Logica	Internal	✓
12	Web Gateway	192.168.30.1*	24	Web Logical Switc	Internal	✓

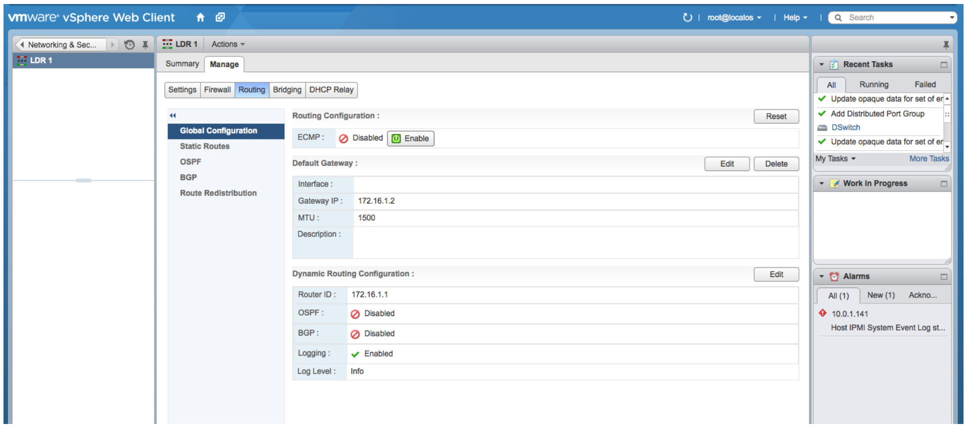
Configuring Routing Protocols

GUI Step-by-Step Procedure

To configure routing protocols for the VMware NSX network:

1. Return to the **Networking & Security** page, click **NSX Edges**, click each of the VMware NSX edge appliances, go to **Manage > Routing**, and set a router ID of 172.16.1.1 in the **Global Configuration** section as shown in Figure 37 on page 43.

Figure 37: Setting the Router ID



Routing Configuration :

ECMP : Disabled

Default Gateway :

Interface :
Gateway IP : 172.16.1.2
MTU : 1500
Description :

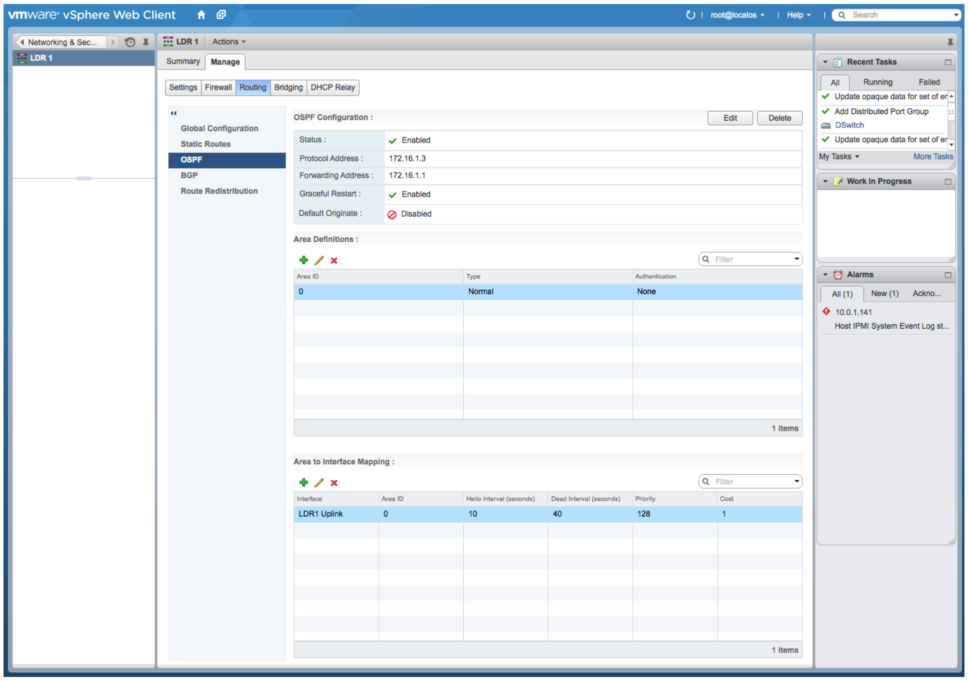
Dynamic Routing Configuration :

Router ID : 172.16.1.1
OSPF : Disabled
BGP : Disabled
Logging : Enabled
Log Level : Info

This step configures the router ID for the VMware NSX Edge Gateway and VMware NSX LDR.

2. While at the **Manage > Routing** section, click **OSPF** in the navigation bar as shown in Figure 38 on page 44.

Figure 38: Configuring OSPF in VMware NSX



OSPF Configuration :

Status :	✓ Enabled
Protocol Address :	172.16.1.3
Forwarding Address :	172.16.1.1
Graceful Restart :	✓ Enabled
Default Originate :	⊘ Disabled

Area Definitions :

Area ID	Type	Authentication
0	Normal	None

1 items

Area to Interface Mapping :

Interface	Area ID	Hello Interval (seconds)	Dead Interval (seconds)	Priority	Cost
LDR1 Uplink	0	10	40	128	1

1 items

Per the logical design in Figure 30 on page 37, the OSPF area between the VMware NSX Edge Gateway and the VCF is Area 0 (0.0.0.0). The OSPF area between the VMware NSX Edge Gateway and VMware NSX LDR is Area 1 (0.0.0.1).

3. For each VMware NSX Edge appliance, click the green **+** symbol to create an area definition, and assign the appropriate interface to the corresponding OSPF area as shown in Table 10 on page 44.

Table 10: VMware NSX OSPF Areas and Interfaces

VMware NSX Appliance	OSPF Area	OSPF Interface
VMware NSX Edge Gateway	0.0.0.0	Uplink
VMware NSX Edge Gateway	0.0.0.1	Internal
VMware NSX LDR	0.0.0.1	Uplink

Configuring Example Applications

GUI Step-by-Step Procedure

Now that you have configured all the VMware NSX components and the VCF, the final step is to create an example allocation and integrate it into VMware NSX.

To configure example applications to interact with VMware NSX:

1. Create six servers and place them into the three logical switches: database, application, and web.

   This example application consists of Debian 7 Linux servers. Simply create new VMs with the settings shown in Table 11 on page 45.

Table 11: Example Application VM Settings

Name	IP Address	Virtual Switch	Host
db-01	192.168.10.100	Database Logical Switch	esxi-01
db-02	192.168.10.101	Database Logical Switch	esxi-02
app-01	192.168.20.100	Application Logical Switch	esxi-01
app-02	192.168.20.101	Application Logical Switch	esxi-02
web-01	192.168.30.100	Web Logical Switch	esxi-01
web-02	192.168.30.101	Web Logical Switch	esxi-02

   Different VMs are placed on different VMware ESXi hosts on purpose. This design ensures that VXLAN works between the VMware ESXi hosts and that multicast MAC address learning occurs on the VCF.

Verification

Confirm that the MetaFabric Architecture 2.0 configuration is working properly.

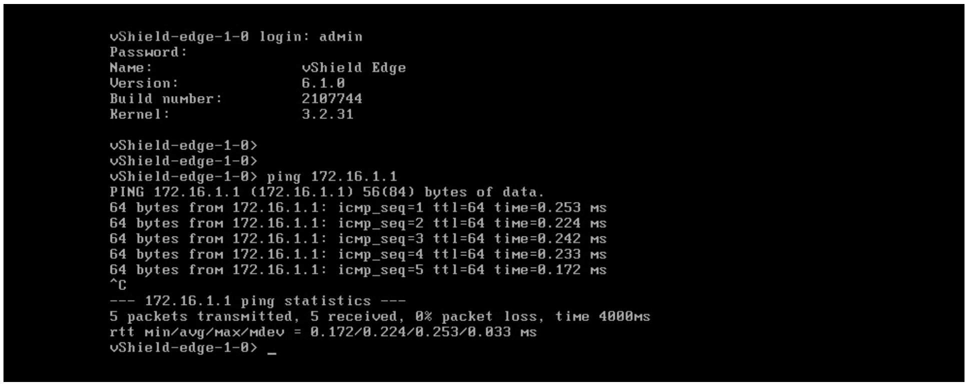
- Verifying Connectivity Between the VMware NSX Edge Gateway and the VMware NSX LDR on page 45
- Verifying OSPF on page 46
- Verifying Connectivity Between the VCF and the VMWare NSX Components on page 46

Verifying Connectivity Between the VMware NSX Edge Gateway and the VMware NSX LDR

Purpose Confirm that the VMware NSX Edge Gateway and the VMware NSX LDR can reach each other.

Action After you configure the VMware NSX OSPF settings, test the connectivity by logging in to the VMware vSphere Controller of the VMware NSX Edge Gateway appliance. Use the **admin** username and the password you specified during the creation of the appliance. Verify connectivity between the VMware NSX Edge Gateway and the VMware NSX LDR by issuing the **ping** command as shown in Figure 39 on page 46.

Figure 39: Test Connectivity Between VMware NSX Edge Appliances



```
vShield-edge-1-0 login: admin
Password:
Name:                  vShield Edge
Version:               6.1.0
Build number:          2107744
Kernel:                3.2.31

vShield-edge-1-0>
vShield-edge-1-0>
vShield-edge-1-0> ping 172.16.1.1
PING 172.16.1.1 (172.16.1.1) 56(84) bytes of data.
64 bytes from 172.16.1.1: icmp_seq=1 ttl=64 time=0.253 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=64 time=0.224 ms
64 bytes from 172.16.1.1: icmp_seq=3 ttl=64 time=0.242 ms
64 bytes from 172.16.1.1: icmp_seq=4 ttl=64 time=0.233 ms
64 bytes from 172.16.1.1: icmp_seq=5 ttl=64 time=0.172 ms
^C
--- 172.16.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.172/0.224/0.253/0.033 ms
vShield-edge-1-0> _
```

Meaning If the **ping** command is successful, connectivity between the VMware NSX Edge Gateway and the VMware NSX LDR is working properly.

Verifying OSPF

Purpose Confirm that the OSPF configuration is working.

Action On the VCF, issue the **show ospf neighbor** command:

```
user@vcf> show ospf neighbor

 Address          Interface              State     ID               Pri  Dead
10.1.12.2        irb.15                  Full     10.0.1.1          128   36
```

On both VMware NSX Edge appliances, issue the **show ip ospf neighbor** command to verify that the OSPF state is **Full/DR**.

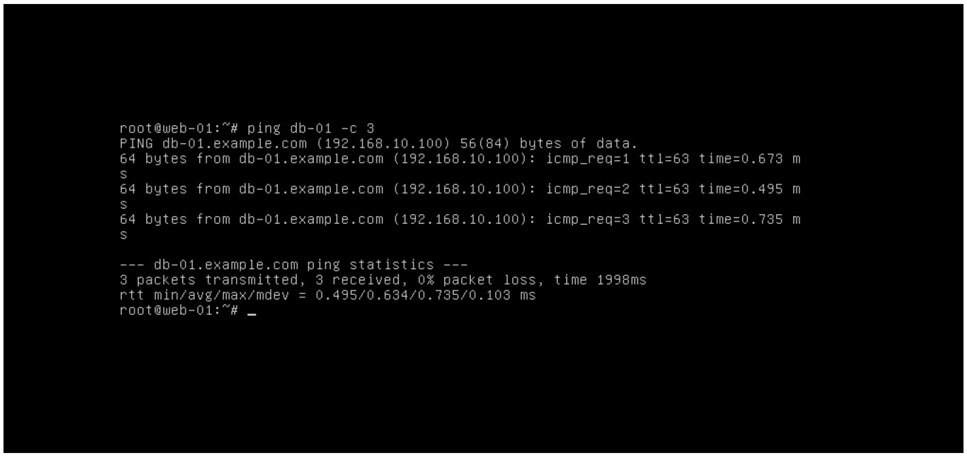
Meaning If the OSPF state is **Full** in both the VCF and the VMware NSX Edge appliances, connectivity between the virtual and physical components is working properly.

Verifying Connectivity Between the VCF and the VMWare NSX Components

Purpose Confirm that your VCF and VMware NSX configuration is working.

Action To verify connectivity between web-01 and db-01, issue the **ping** command on a client for web-01 as shown in Figure 40 on page 47.

Figure 40: Ping Between web-01 and db-01

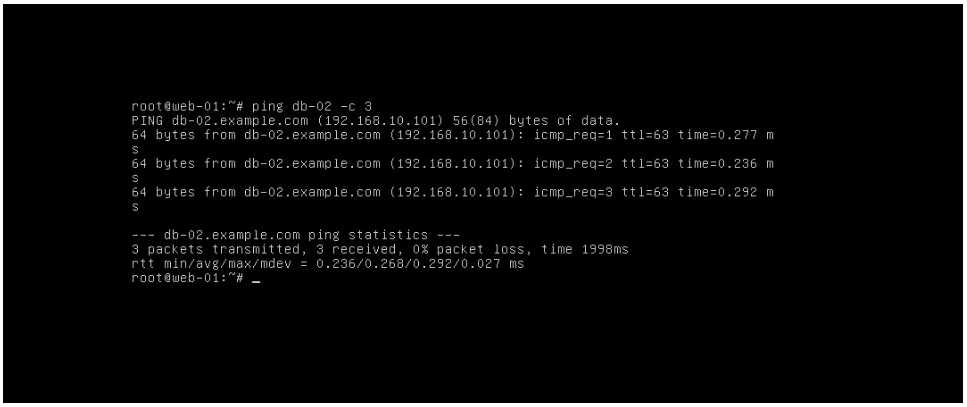


```
root@web-01:~# ping db-01 -c 3
PING db-01.example.com (192.168.10.100) 56(84) bytes of data.
64 bytes from db-01.example.com (192.168.10.100): icmp_req=1 ttl=63 time=0.673 m
s
64 bytes from db-01.example.com (192.168.10.100): icmp_req=2 ttl=63 time=0.495 m
s
64 bytes from db-01.example.com (192.168.10.100): icmp_req=3 ttl=63 time=0.735 m
s

--- db-01.example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.495/0.634/0.735/0.103 ms
root@web-01:~# _
```

The VMs have full connectivity, but only through the local VMware LDR on the local VMware ESXi host. The next step is to verify connectivity through VXLAN and multicast MAC address learning. To verify connectivity between web-01 and db-02, issue the **ping** command on a client for web-01 as shown in Figure 41 on page 47.

Figure 41: Ping Between web-01 and db-02



```
root@web-01:~# ping db-02 -c 3
PING db-02.example.com (192.168.10.101) 56(84) bytes of data.
64 bytes from db-02.example.com (192.168.10.101): icmp_req=1 ttl=63 time=0.277 m
s
64 bytes from db-02.example.com (192.168.10.101): icmp_req=2 ttl=63 time=0.236 m
s
64 bytes from db-02.example.com (192.168.10.101): icmp_req=3 ttl=63 time=0.292 m
s

--- db-02.example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.236/0.268/0.292/0.027 ms
root@web-01:~# _
```

Meaning When web-01 pings db-02, the traffic is encapsulated in VXLAN and transmitted across the VCF. MAC address learning happens through multicast, and all subsequent unicast traffic is sent directly to the VTEP on the VMware ESXi host esxi-02. Because the pings between web-01 and db-01 were successful, and the pings between web-01 and db-02 were successful, connectivity between the VCF and the VMWare NSX components is working properly.

Related Documentation

- Understanding MetaFabric Architecture 2.0 on page 5
- Understanding Network Virtualization with VMware NSX on page 13