Deploying the VXLAN Feature in Cisco Nexus 1000V Series Switches

Deployment Guide

May, 2012

For further information, questions and comments please contact ccbu-pricing@cisco.com

Contents

Overview

This document provides guidelines for deploying Virtual Extensible LAN (VXLAN) on Cisco Nexus® 1000V Series Switches. For detailed configuration documentation, please refer to the configuration guides on http://www.cisco.com.

Audience

This document is intended for network architects, network engineers, virtualization administrators, and server administrators interested in understanding and deploying Cisco Nexus 1000V VXLAN technology on the Cisco Nexus 1000V Series.

Introduction

Many customers are building private or public clouds. Intrinsic to cloud computing is having multiple tenants with numerous applications using the cloud infrastructure. Each of these tenants and applications needs to be logically isolated, even at the networking level. For example, a three-tier application can have multiple virtual machines requiring logically isolated networks between the virtual machines. Traditional network isolation techniques such as IEEE 802.1Q VLAN provide 4096 LAN segments (through a 12-bit VLAN identifier) and may not provide enough segments for large cloud deployments.

Cisco and a group of industry vendors are working together to address new requirements for scalable LAN segmentation as well as for transporting virtual machines across a broader diameter. The underlying technology, referred to as Virtual eXtensible LAN (or VXLAN), defines a 24-bit LAN segment identifier to provide segmentation at cloud-deployment scale. More details can be found in the IETF draft: http://www.ietf.org/mail-archive/web/i-d-announce/current/msg39532.html.

In addition, VXLAN provides an architecture for customers to grow their cloud deployments with repeatable pods in different subnets. With Cisco Nexus 1000V Series Switches supporting VXLAN, customers can quickly and confidently deploy their applications to the cloud.

The Cisco Nexus 1000V Series supports VXLAN and provides significant benefits beyond VXLAN's baseline capabilities:

- **Fully supports VMware vCloud Director 1.5:** The Cisco Nexus 1000V Series version 1.5 [4.2(1)SV1(5.1)] is fully integrated into VMware vCloud Director, providing on-demand provisioning of the network.
- **Extends existing operational model to the cloud:** The Cisco Nexus 1000V Series offers a nondisruptive operational model for network and server administrators. With the Cisco Nexus 1000V Series supporting VXLAN, the same operational model can now be extended to the cloud without disrupting the existing operational model, accelerating cloud deployment.

This document focuses on the steps for deploying VXLAN technology on Cisco Nexus 1000V Series Switches. We will not go over the details or best practices for deploying the 1000V Series. For that information, refer to the Cisco Nexus 1000V Series Switches Deployment Guide at: http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9902/guide_c07-556626.html.

Deployment Steps

Here is the summary of steps required to deploy VXLAN on the Cisco Nexus 1000V Series:

1. Install Cisco Nexus 1000V Series.
2. Turn on VXLAN feature on Cisco Nexus 1000V Series.
3. Create a new port-profile with capability VXLAN.
4. Create a new vmkernel interface to each ESX host and assign the new port-profile.
5. Turn on multicast on uplink physical Layer 3 switch or router.
6. Turn on Proxy Address Resolution Protocol (ARP) on upstream Layer 3 switch or router.
7. Increase the MTU on the 1000V Series uplink interfaces and uplink physical interfaces.

After performing these steps, you are ready to create port-profile backed by VXLAN.

Background

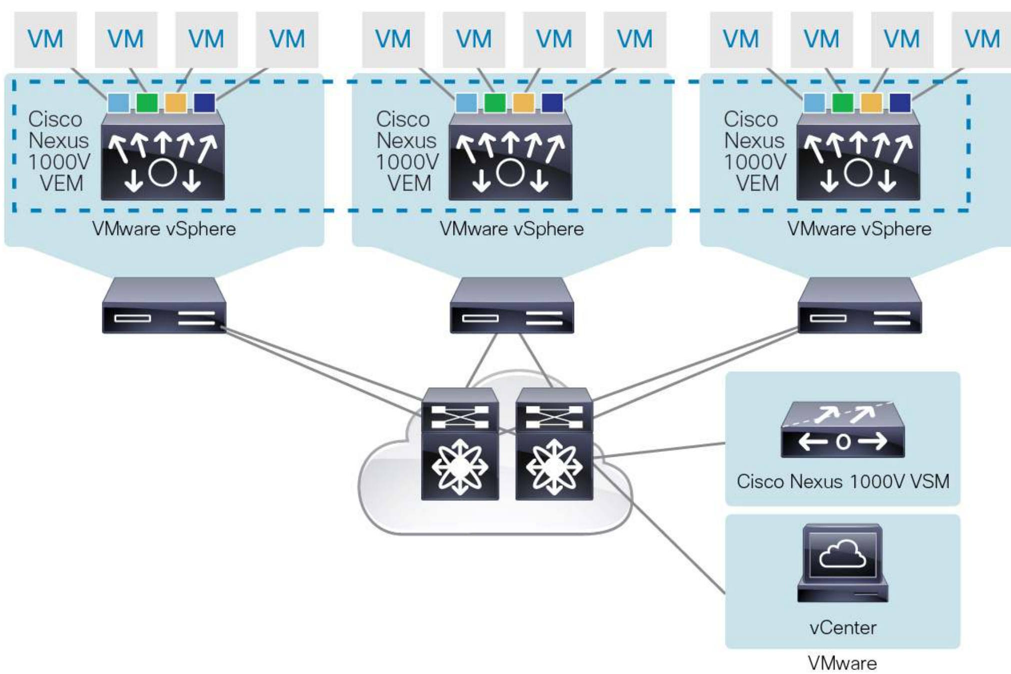
Cisco Nexus 1000V Networking

The Cisco Nexus 1000V Series provides Layer 2 switching, advanced networking functions, and a common network management model in a virtualized server environment by replacing the virtual switch within the VMware vSphere. As Figure 1 shows, the Cisco Nexus 1000V Series Switch manages a data center as defined in VMware vCenter Server. Each server in the data center is represented as a line card in the Cisco Nexus 1000V Series Switch and can be managed as if it were a line card in a physical Cisco® switch.

The Cisco Nexus 1000V Series implementation has two main components:

- Virtual Supervisor Module (VSM)
- Virtual Ethernet module (VEM)

Figure 1. Cisco Nexus 1000V Series Switches Managing VMware ESX Servers



Benefits

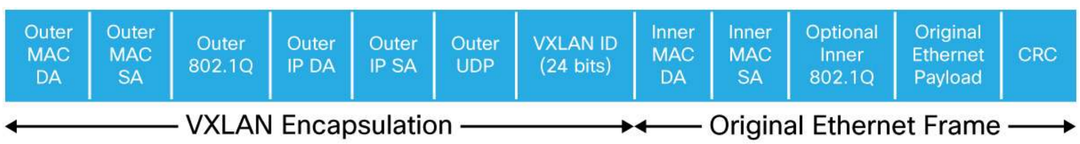
The benefits of deploying the Cisco Nexus 1000V Series include:

- Advanced networking capabilities, such as quality of service, network statistics gathering with Cisco NetFlow Collector, packet mirroring with Cisco ERSPAN, and many others
- Nondisruptive operational model, with Cisco Nexus 1000V Series fully integrated into vCloud Director and VMware vCenter Server
- Easier regulatory compliance of applications in the cloud since there is complete transparency in both the physical and virtual networks

Overview of Cisco Nexus 1000V VXLAN

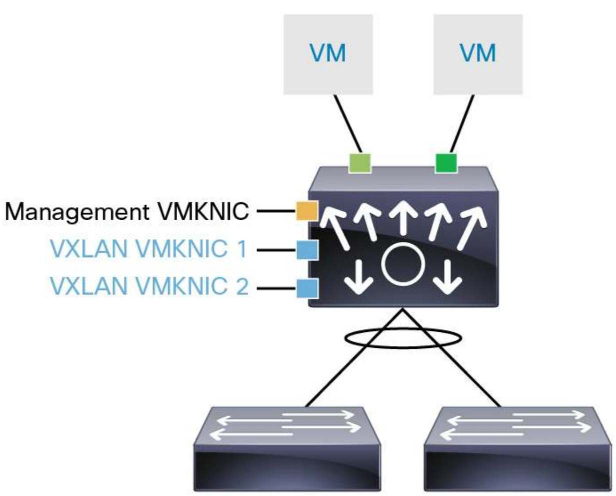
Cisco VXLAN is a Layer 2 network isolation technology that uses a 24-bit segment identifier to scale beyond the 4K limitations of VLANs. VXLAN technology creates LAN segments by using an overlay approach with MAC in IP encapsulation. The Virtual Ethernet Module (VEM) encapsulates the original Layer 2 frame from the virtual machine (VM) (Figure 2).

Figure 2. VXLAN encapsulated frame format



Each VEM is assigned an IP address, which is used as the source IP address when encapsulating MAC frames to be sent on the network. This is accomplished by creating virtual network adaptors (VMKNICs) on each VEM (Figure 3). You can have multiple VMKNICs per VEM and use them for this encapsulated traffic. The encapsulation carries the VXLAN identifier, which is used to scope the MAC address of the payload frame.

Figure 3. VEM VMKNIC Interface with VXLAN Capability



The connected VXLAN is specified within the port profile configuration of the vNIC and is applied when the VM connects. Each VXLAN uses an assigned IP multicast group to carry broadcast traffic within the VXLAN segment.

When a VM attaches to a VEM, if it is the first to join the particular VXLAN segment on the VEM, an Internet Group Management Protocol (IGMP) join is issued for the VXLAN's assigned multicast group. When the VM transmits a packet on the network segment, a lookup is made in the Layer 2 table using the destination MAC of the frame and the VXLAN identifier. If the result is a hit, the Layer 2 table entry will contain the remote IP address to use to encapsulate the frame, and the frame will be transmitted within an IP packet destined to the remote IP address. If the result is a miss (broadcast/multicast/unknown unicasts fall into this bucket), the frame is encapsulated with the destination IP address set to be the VXLAN segment's assigned IP multicast group.

Deployment Considerations

Multicast

In a typical Layer 2 network using VLANs, if a frame is received with an unknown destination MAC address, it is flooded out to of every interface (except the one it came from). In VXLAN, multicast/broadcast (including unknown unicast) frames will be sent encapsulated with a multicast destination IP address. Ideally, each VXLAN should use a different IP multicast group address to minimize flooding frames to VEMs that do not need them. When using VXLAN encapsulation, a multicast IP address must be assigned to each VXLAN.

If the VXLAN VMKNICs on different VEMs belong to the same subnet, you only need to enable IGMP snooping on the VLAN on upstream switching to provide Layer 2 optimization for multicast traffic.

If the VXLAN VMKNICs on different VEMs are in different subnets, Layer 3 multicast routing must be configured on the upstream routers. This will be a scenario where you have two or more N1KV switches on different physical subnets and a VXLAN segment is span across more than one N1KV switch. The recommended Multicast protocol to deploy for this scenario is Bidirectional Protocol Independent Multicast (PIM), since the VEMs act as both multicast speakers and receivers at the same time. For more information on deploying multicast on Cisco switches and routers, visit: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6552/ps6592/whitepaper_c11-474791.html

Proxy ARP

VXLAN VMkernel interface IP address is used to encapsulate and transport VXLAN frames in a UDP tunnel. The VMware host routing table is ignored. The VMKNIC's netmask is also ignored. The VEM will initiate an ARP for all remote VEM IP addresses, regardless of whether they are on the same subnet or not. If they are across a Layer 3 switch or router, you need to enable the Proxy ARP feature on the Layer 3 gateway so that it can respond to off-subnet ARPs.

Communication outside the VXLAN

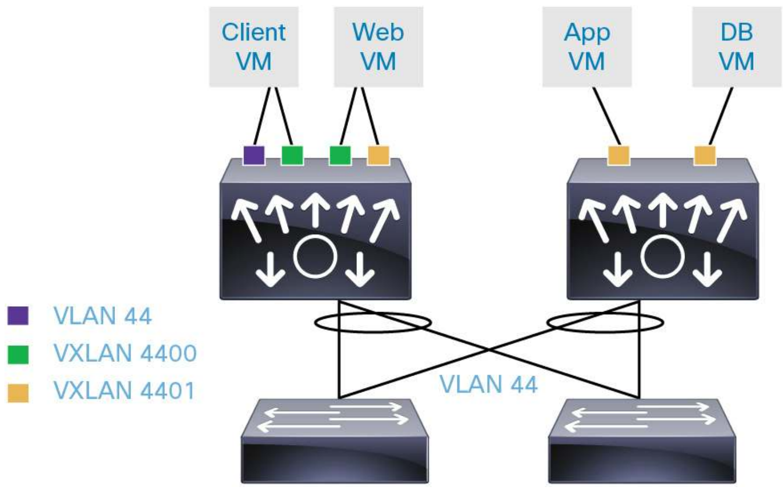
Today, the VXLAN format is only supported by the Cisco Nexus 1000V Series Switch. If communication has to be established outside the VXLAN segment, there are two options available today.

The first option is a multihomed VM with one interface in VXLAN and one interface in a VLAN. Any communication to be established to VXLAN from outside has to traverse by mean of the multihomed VM on the VLAN interface. For example, suppose you have a vApp with following VMs:

- Dual-homed client machine with one interface in VXLAN 4400 and one interface in VLAN 44
- Web server with one interface in VXLAN 4400 and one interface in VXLAN 4401
- Database server with one interface in VXLAN 4401

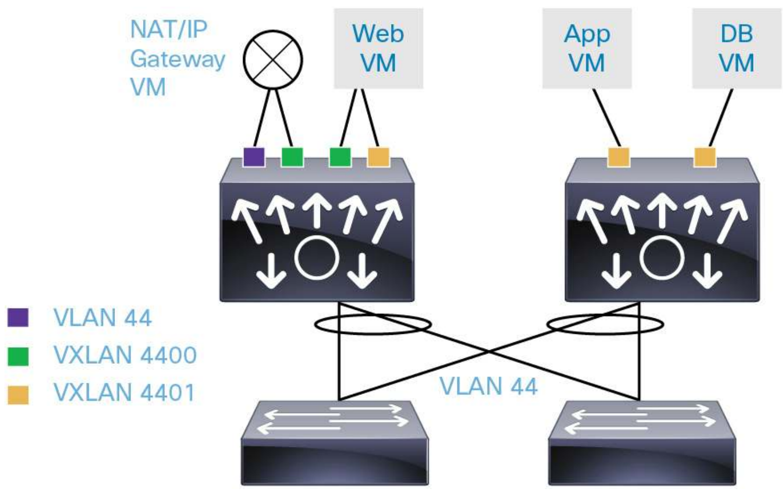
In this scenario, you can do a remote desktop to a client machine on the interface, which is on VLAN 44, and then browse to the webserver in VXLAN 4400. Now web server can communicate to the database server on VXLAN 4401, which is totally isolated i.e. the client machine has no access to the database server directly (Figure 4).

Figure 4. External Connectivity with Dual-Homed VM



Another option is to have a VM (for example, a firewall VM) o provide NAT and IP Gateway functions to connect VMs on VXLAN to outside network. Let's take the same example as in Figure 4 and see how it will connect to external network using firewall VM. In this case, the external interface of the firewall VM will be on VLAN 44, and internal interface on VXLAN 4400 (Figure 5).

Figure 5. External Connectivity via IP Gateway VM



Same VXLAN on Multiple Cisco Nexus 1000V Series Switches

The same VXLAN can be configured on more than one Cisco Nexus 1000V Series Switch. VMs connected to one Cisco Nexus 1000V Series Switch are part of the same Layer 2 domain as VMs connected to the other Cisco Nexus 1000V Series Switch on the same VXLAN segment. The only requirement is that when you are creating the segment ID on these switches, they should have same multicast group.

For example, suppose you create a VXLAN on 1000V Series by defining a segment ID and multicast group as follows:

```
bridge-domain Tenant1
  segment id 5000
  group 225.1.1.1
```

If the same segment ID is being defined on multiple Cisco Nexus 1000V Series Switches, it should be associated with the same multicast group as configured on the other Nexus 1000V switch.

VXLAN Working with OTV/LISP

VXLANs are intended for creating a large number of logical networks in a cloud environment. Overlay Transport Virtualization (OTV) is a data center interconnect technology extending VLANs to different data centers over Layer 3. Unlike VXLAN, OTV has simpler deployment requirements since it does not mandate a multicast-enabled transport network. Locator ID Separation Protocol (LISP) goes a step further by providing IP address mobility between data centers with dynamic routing updates. While VXLAN, OTV, and LISP, all use User Datagram Protocol/Internet Protocol (UDP/IP) encapsulation, they serve very different networking purposes and are hence complementary to each other.

Scalability with VXLAN

Today, a single Cisco Nexus 1000V Series Switch supports up to 2000 Layer 2 logical networks consisting of VLAN and VXLAN. In order to scale beyond 2000 Layer 2 logical networks, you need to deploy additional Cisco Nexus 1000V Series Switches.

Securing VXLAN in Physical Network

Since VXLAN is transported over IP in a physical network, some best practice recommendations should be implemented when setting up the transport network for VXLAN.

The preferred option is to have all the VXLAN VMkernel interfaces on the VEM in the same subnet. In this scenario, you can make them part of the same VLAN and keep that VLAN a strict Layer 2 VLAN. Only the VMKNICs used for VXLAN encapsulation should attach to this VLAN. This approach will provide natural protection and limit unwanted exposure to external communications.

In a scenario, where the number of VEMs has exceeded the available IPs in the subnet, VMKNICs for VXLAN encapsulations may need to be assigned IP addresses in multiple subnets. In this scenario, where VXLAN VMkernel interfaces belong to two different VLANs, the communications between the multiple subnets has to take place through a Layer 3 switch or router. Both VLANs must have switch virtual interface (SVI) interfaces. To make sure that VXLAN traffic cannot be attacked or snooped from unauthorized endpoints, use one of following options:

- Use access control lists (ACLs) to prevent unauthorized injection of VXLAN encapsulated traffic to VEM VMKNICs from outside sources.
- Use a VRF to segregate the VLANs and SVIs on which VXLAN VMKNICs are assigned IP addresses. For specific configurations of ACLs or VRFs, please refer to the configuration guides for your physical Layer 3 switch or router.

The recommended scenarios just described not only reduce external security threats, but also keep the multicast deployment simpler in the physical network

Port Channel

Port channels use different load-sharing algorithms for dividing outgoing traffic among different physical interfaces. IP encapsulation results in all outgoing VXLAN traffic carrying an outer header that has the source MAC/IP address of the VEM's VMKNIC. For optimal load balancing, users must configure a 5-tuple-based hash as the load-sharing algorithm. The use case section of the document will cover how to configure 5-tuple-based hashes.

MTU Size

VXLAN traffic is encapsulated in a UDP packet when sent out to the physical network. This encapsulation imposes the following overhead on each packet:

Outer Ethernet Header (14) + UDP header (8) + IP header (20) + VXLAN header (8) = 50 bytes

To avoid fragmentation and possible performance degradation, all the physical network devices transporting the VXLAN traffic need to handle 50 bytes greater than the maximum transmission unit (MTU) size expected for the frame. Therefore, adjust the MTU settings for all these devices, which will transport the VXLAN traffic. This includes the uplink port-profiles of Cisco Nexus 1000V Series Switch carrying the VXLAN traffic.

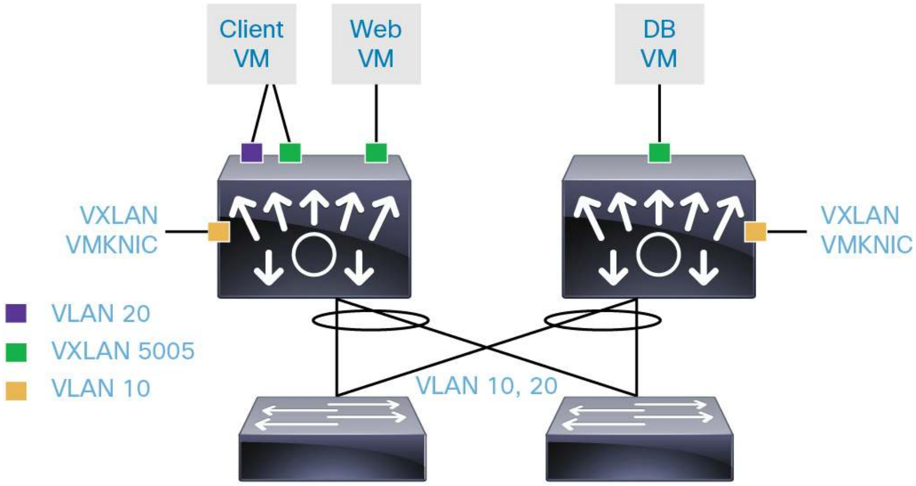
Some switches take a global setting for MTU and others must be set per port. Refer to the system configuration guides of your upstream switches to increase the MTU of the physical interfaces of all the transit switches and routers.

VXLAN Deployment Use Cases

Deploying a Two-Tier Web Development vApp

Figure 6 shows this simple use case.

Figure 6. Two-Tier Web Development App Deployed on VXLAN for Isolation



In this simple use case, we are deploying a two-tier web development application. Inter-VM communications between the web and database servers are isolated by using a VXLAN. Only the northbound communication is taking place on VLAN 20. Remote users can connect to the Win7 (dual-homed) on this VLAN and have access to this web development environment, which resides on the VXLAN. The goal is to provide the web developer with an isolated environment to test by remotely connecting to Win 7 client and accessing the web application servers that reside in the VXLAN. Only the Win7 client has northbound connectivity through the external network.

Setting up the Cisco Nexus 1000V for VXLAN

Step 1. Turn on the NSM and VXLAN feature on Cisco Nexus 1000V.

```
N1KV-VSM(config)# feature segmentation
```

Verify that the feature is enabled on Cisco Nexus 1000V

```
N1KV-VSM(config)# show feature
Feature Name           Instance     State
--------------------   --------     --------
dhcp-snooping             1         disabled
http-server               1         enabled
lacp                      1         disabled
netflow                   1         disabled
network-segmentation      1         disabled
port-profile-roles        1         disabled
private-vlan              1         disabled
segmentation              1         enabled
sshServer                 1         enabled
tacacs                    1         disabled
telnetServer              1         disabled
```

Step 2. Create a port-profile with capability VXLAN.

```
port-profile type vethernet VMK-FI-A
   vmware port-group
  switchport access vlan 10
  capability vxlan
  no shutdown
  state enabled
```

You can verify that the VXLAN is enabled on this interface by issuing the command:

```
N1KV-VSM(config)# show port-profile name VMK-FI-A

port-profile VMK-FI-A
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  switchport access vlan 10
```



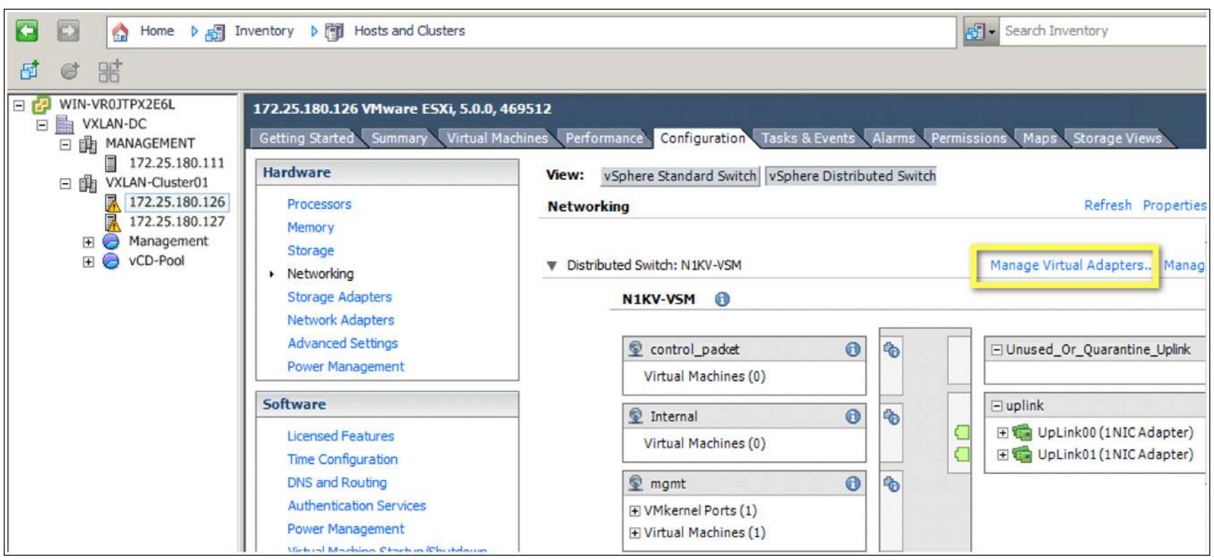
```
 capability vxlan
 no shutdown
evaluated config attributes:
 switchport access vlan 10
 capability vxlan
 no shutdown
assigned interfaces:
port-group: VMK-FI-A
system vlans: none
capability l3control: no
capability iscsi-multipath: no
capability vxlan: yes
port-profile role: none
port-binding: static
```

Step 3. Create VMkernel interface on ESX host.

Attach a VMkernel interface to each ESX host of the cluster in vCenter.

First, navigate to Home > Inventory > Host and Clusters in the vCenter. Next, under Configuration >Networking > vSphere Distributed Switch, select Manage Virtual Adapter as shown in Figure 7:

Figure 7. Selecting Manage Virtual Adapters



Now add a new VMkernel interface, as shown in Figures 8 through 10:

Figure 8. Selecting Add in Manage Virtual Adapters

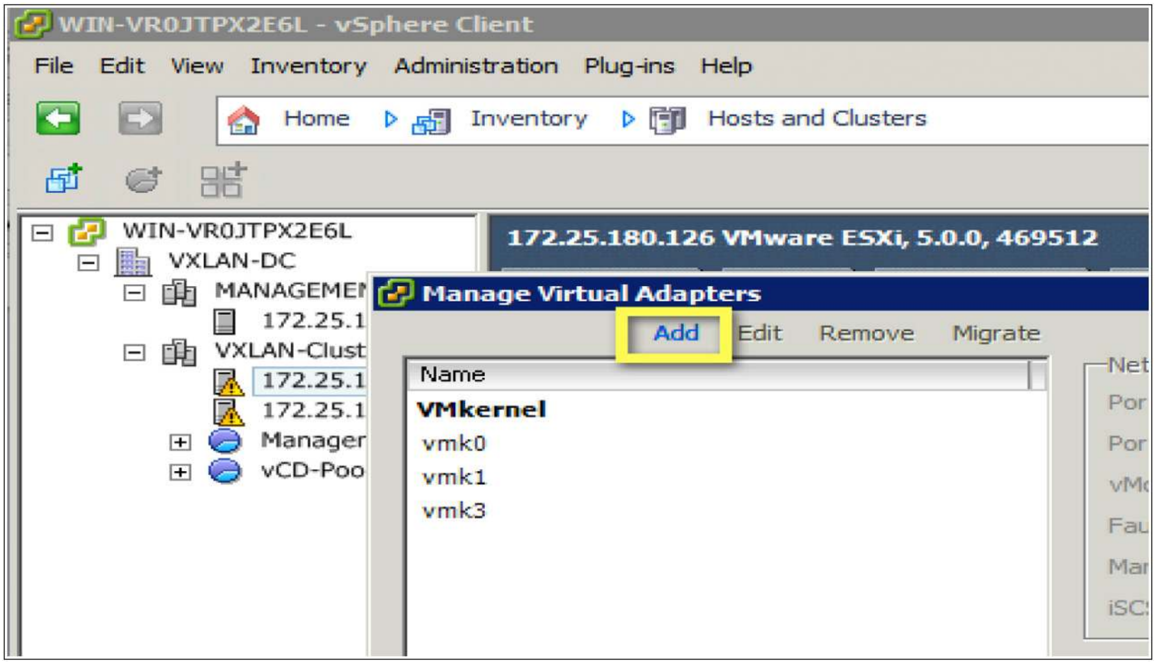


Figure 9. Selecting a New Virtual Adapter

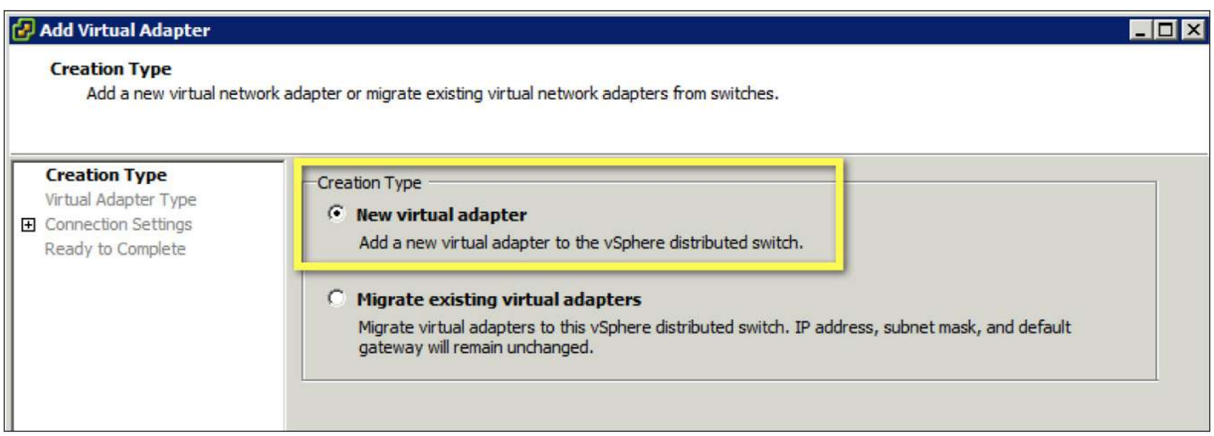
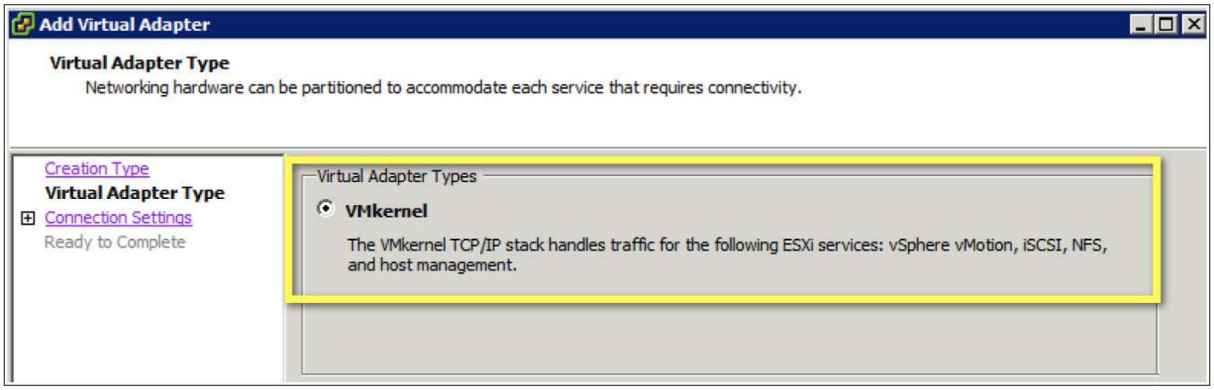


Figure 10. Selecting a New VMkernel Interface



Finally, select the VXLAN-enabled port-profile and configure the IP-to-VMkernel interface used to encapsulate the VXLAN, as shown in Figures 11 and 12.

Figure 11. Selecting VXLAN-Enabled Port-Profile

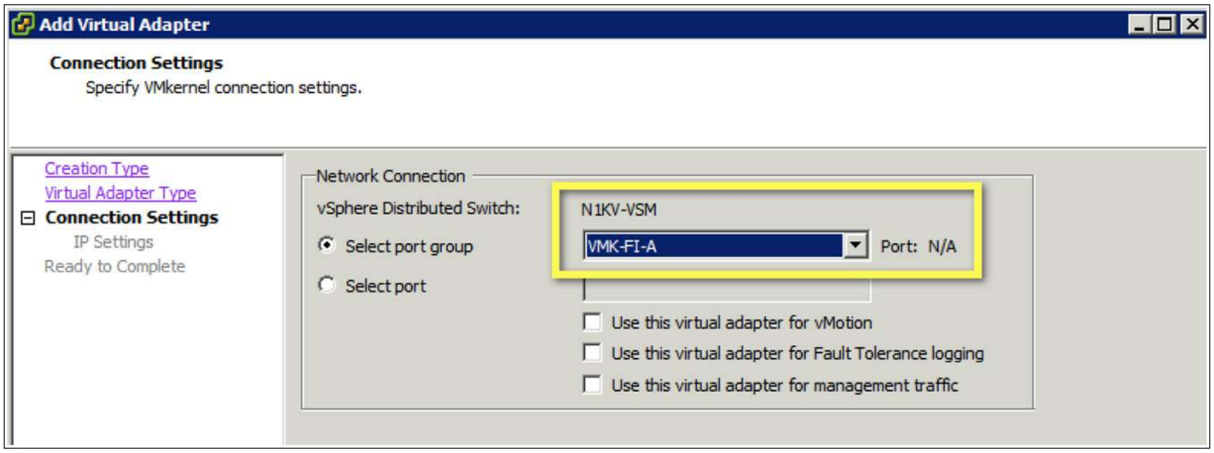


Figure 12. Configuring an IP-to-VMkernel Interface Used to Encapsulate VXLAN

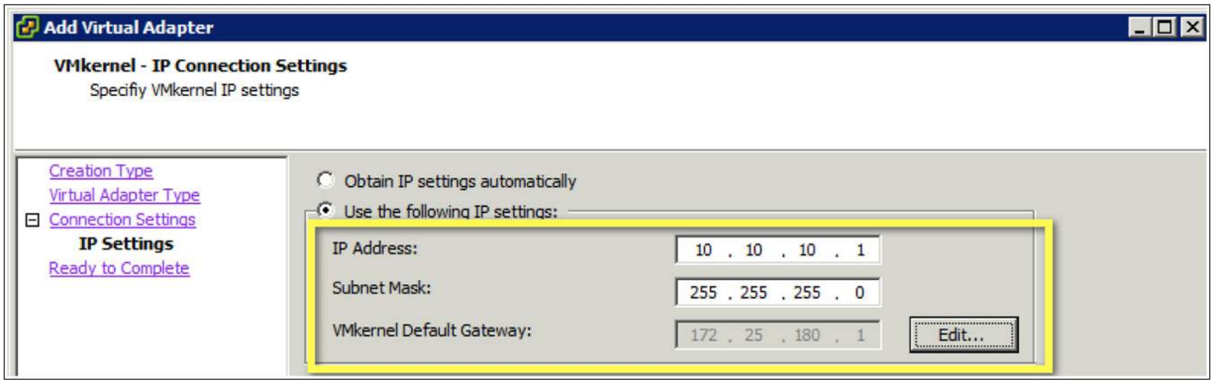
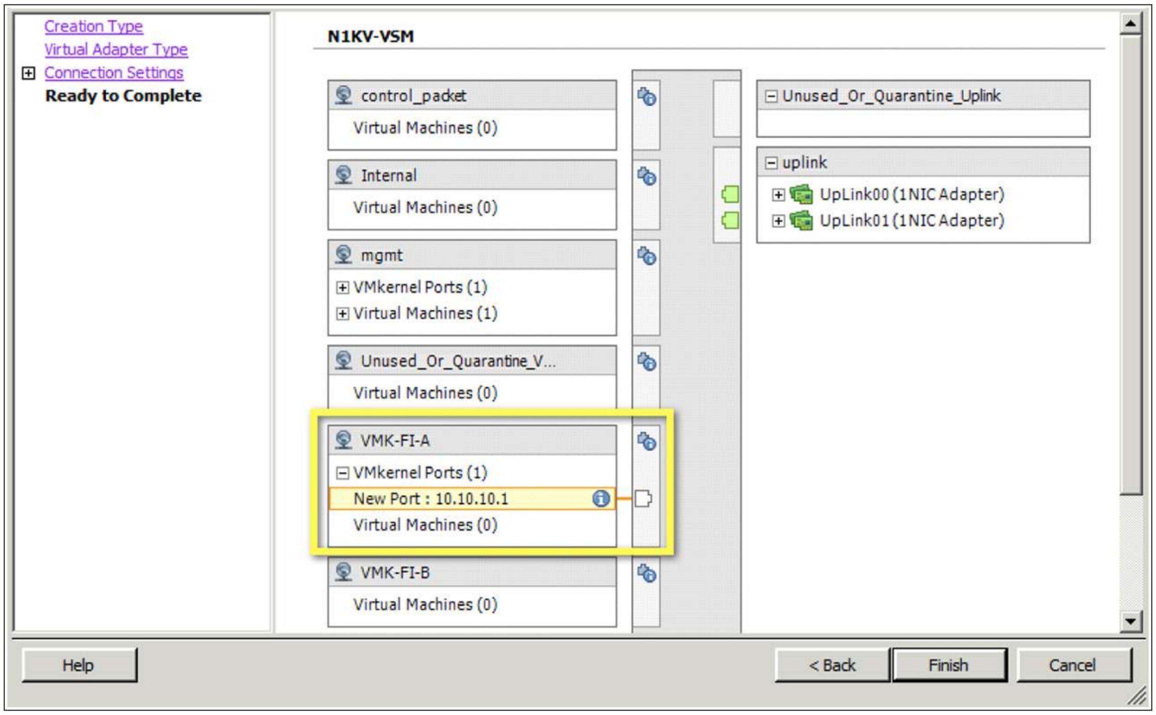


Figure 13 shows a summary of the new VMkernel interface.

Figure 13. Summary of VXLAN VMkernel Interface



Repeat the steps for other ESX hosts. The only difference is that you need to assign a unique IP address for each interface created on the host.

On the VSM, you can verify the interfaces are up on that Layer 3 VMkernel interface by issuing the following command:

```
N1KV-VSM(config)# sh port-profile name VMK-FI-A

port-profile VMK-FI-A
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  switchport access vlan 10
  capability vxlan
  no shutdown
 evaluated config attributes:
  switchport access vlan 10
  capability vxlan
  no shutdown
```



```
 assigned interfaces:
  Vethernet4
  Vethernet5
 port-group: VMK-FI-A
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: yes
 port-profile role: none
 port-binding: static
```

The two virtual VMkernel interfaces (vEthernet 4 and vEthernet 5) belong to two different ESX hosts in the example.

Step 4. Change the MTU on the uplink interface.

To avoid fragmentation, it is highly recommended to increase the MTU of the uplink interfaces of Cisco Nexus 1000V and the physical interfaces of the upstream switches, which are connected the Layer 2 domain of the vSphere environment.

The following command needs to be configured on the uplink port-profile to increase the MTU:

```
port-profile type ethernet uplink
  vmware port-group
  switchport trunk allowed vlan 10, 20,180
  switchport mode trunk
  switchport trunk native vlan 180
  mtu 1550
  no shutdown
  system vlan 10,180
  state enabled
```

Refer to the system configuration guides for your upstream switches to increase the MTU of the physical interfaces of all the transit switches and routers.

Step 5. Enable multicast on upstream physical switch.

In this example, all the VEM VXLANs are in the same VLAN. We are enabling the IGMP snooping querier on the VLAN.

```
vlan 10
  ip igmp snooping querier 10.45.46.45

5K-B# show ip igmp snooping querier

Vlan IP Address       Version   Expires     Port
10    10.45.45.45      v3        00:02:45    Ethernet1/8
```

Step 6. Create the VXLAN (bridge domain) in VSM.

Now you are ready to create VXLAN IDs in VSM and place the VMs on the VXLAN. On VSM, configure a new bridge-domain as follows:

```
config t
bridge-domain vxlan_5005
  segment id 5005
  group 225.1.1.5
```

In the example, `segment ID` is the VXLAN ID and `group` is the multicast group. You can verify bridge domain status using the following `show` command:

```
N1KV-VSM# show bridge-domain vxlan_5005

Bridge-domain vxlan_5005 (0 ports in all)
Segment ID: 5005 (Manual/Active)
Group IP: 225.1.1.5
State: UP               Mac learning: Enabled
```

Step 7. Create VXLAN-backed port-profile.

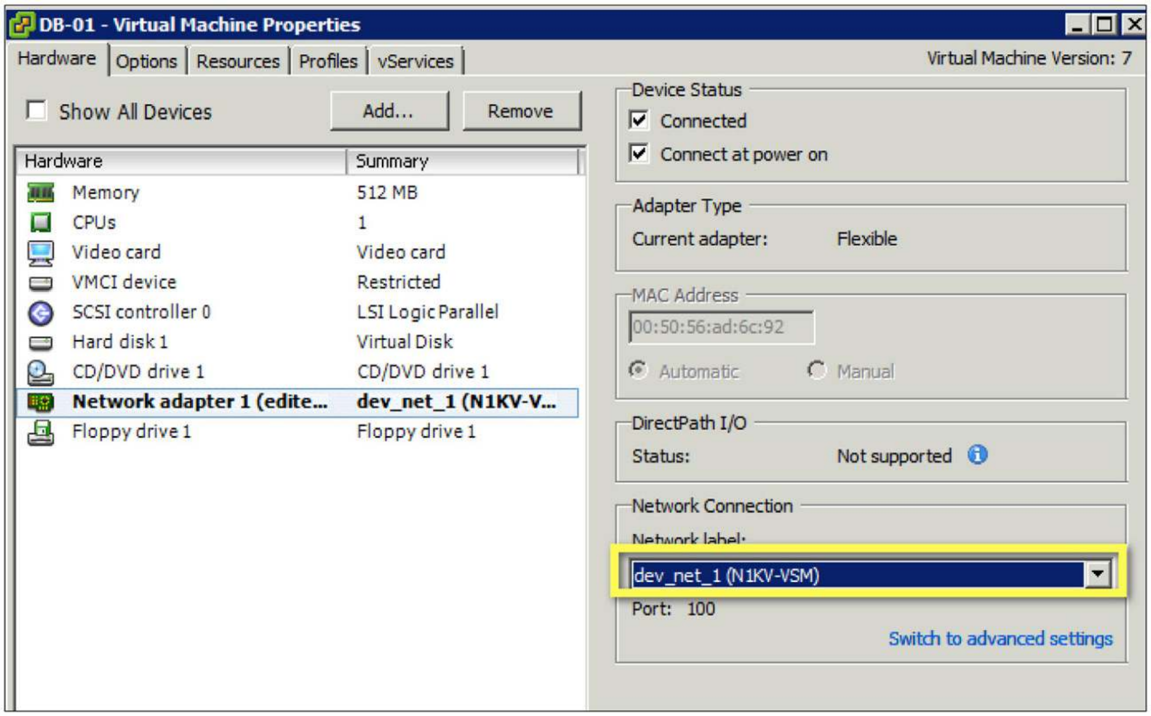
```
port-profile type vethernet dev_net_1
  vmware port-group
  switchport mode access
  switchport access bridge-domain vxlan_5005
  no shutdown
  state enabled
```

Step 8. Assign the VM to newly created port-profile.

Navigate to Home > Inventory > Host and Clusters in the vCenter

Select the VM and right-click for the properties. Assign the VM to the newly created port-profile, as shown in Figure 14.

Figure 14. Assigning the VM to the New Port-Profile



Similarly, the other two VMs are attached to the network using the same port-group.

On the VSM, you can verify that all those VMs are connected to the port-profile using the following command:

```
N1KV-VSM# show port-profile usage name dev_net_1

port-profile dev_net_1
 Vethernet1
 Vethernet2
 Vethernet17
```

The following command verifies that the vEthernets ports belonging to the VXLAN:

```
N1KV-VSM# show bridge-domain vxlan_5005

Bridge-domain vxlan_5005 (3 ports in all)
Segment ID: 5005 (Manual/Active)
Group IP: 225.1.1.5
State: UP               Mac learning: Enabled
Veth1, Veth2, Veth17
```

On the VEM command-line interface, additional commands are available to verify the multicast and VXLAN encapsulation statistics:


```
~ # vemcmd show port
  LTL   VSM Port  Admin Link  State  PC-LTL  SGID  Vem Port  Type
   17     Eth4/1     UP   UP    FWD     305     0    vmnic0
   18     Eth4/2     UP   UP    FWD     305     1    vmnic1
   49     Veth17     UP   UP    FWD       0          DB-01.eth0
   50      Veth2     UP   UP    FWD       0          Web-01.eth0
   58     Veth10     UP   UP    FWD       0     0    vmk0
   59      Veth4     UP   UP    FWD       0     0    vmk1  VXLAN
   60     Veth14     UP   UP    FWD       0     1    vmk3
  305        Po2     UP   UP    FWD       0
```

```
~ # vemcmd show vxlan interfaces
LTL           IP       Seconds since Last
                       IGMP Query Received
(* Interface on which IGMP Joins are sent)
------------------------------------------
 59      10.10.10.1        63          *
~ #
```

```
~ # vemcmd show vxlan-stats
  LTL  Ucast   Mcast   Ucast   Mcast    Total
       Encaps  Encaps  Decaps  Decaps   Drops
   49     136      42     131      55       0
   50       0      13       0      55       0
   59    1106     110    1100     160     150
```

Also, IGMP details for the VXLAN VMKNIC can be verified on the VEM command-line interface by issuing the following command:

```
~ # vemcmd show igmp 10 detail

IGMP is ENABLED on VLAN 10

Multicast Group Table:

 Group */*, Multicast LTL: 4409
    Members: 305
```

Summary

The Cisco VXLAN solution enables scalable cloud architecture with replicated server pods in different subnets. Because of the Layer 3 approach of UDP, virtual machine migration extends even to different subnets. Cisco Nexus 1000V Series Switch with VXLAN support provides numerous advantages for customers, enabling customers to use LAN segments in a robust and customizable way without disrupting existing operational mode.

For More Information

For more information about the Cisco Nexus 1000V Series, please refer to the following URLs:

- Cisco Nexus 1000V Series product information: http://www.cisco.com/go/1000v
- Cisco Nexus 1000V Series technical documentation: http://www.cisco.com/go/1000vdocs
- Cisco Nexus 1000V Series community: http://www.cisco.com/go/1000vcommunity

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Printed in USA C07-702975-01 06/12