

HOW TO BUILD A NESTED NSX-T 2.3 LAB

Explore the features and capabilities of
VMware NSX-T

Jim Streit, VCIX-NV

NSX Senior Technical Account Specialist (TAS)

VMware Professional Services

Table of Contents

INTRO: WHY BUILD A NESTED NSX-T LAB?	3
WHAT WE ARE GOING TO DEPLOY.....	3
1. GETTING STARTED.....	4
2. DEPLOY AND CONFIGURE VCENTER SERVER APPLIANCE 6.5U1	5
3. DEPLOY AND CONFIGURE 3 X NESTED ESXI 6.5U2 VIRTUAL APPLIANCE VMS	5
4. DEPLOY NSX-T MANAGER, 1 X CONTROLLER & 1 X EDGE.....	7
5. DEPLOY THE NSX-T CONTROLLER.....	8
6. DEPLOY AN NSX-T EDGE.....	9
7. CONFIGURE NSX-T	11
8. VIRTUAL NETWORKING	18
9. SECURITY, THE DISTRIBUTED FIREWALL (DFW).....	23

Intro: Why build a nested NSX-T Lab?

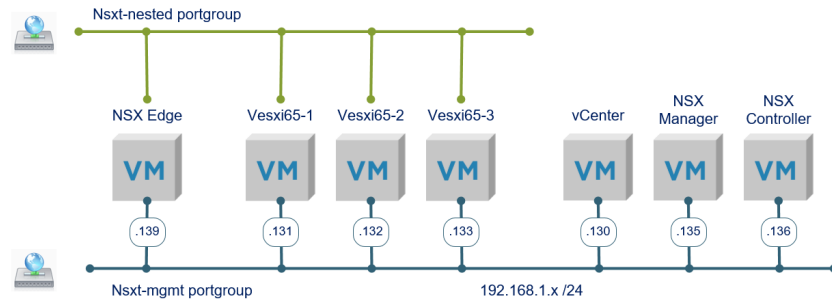
This technical document covers how to build a nested lab with NSX-T 2.3 running in it, so that you can easily explore the VMware NSX platform. A nested hypervisor is the process of running a hypervisor inside another hypervisor... or an ESXi host in a virtual machine, running on an ESXi host. A nested lab is like mini-datacenter that is a collection of virtual machines which provides a small area that won't impact your production environment and doesn't require a lot of expensive hardware or space, hence is extremely useful and cost-effective.

Quick disclaimer: an ESXi host running as a VM is not officially supported by VMware, but there is great community support for building and running this type of environment.

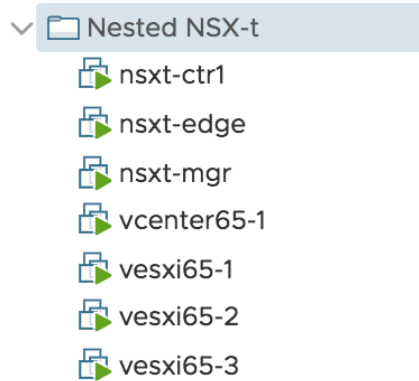
What we are going to deploy

1. Building the nested environment
 - Deploy and configure vCenter Server Appliance 6.5u1
 - Deploy and configure 3 x Nested ESXi 6.5u1 Virtual Appliance VMs and attaching it to vCenter Server
2. Deploying NSX-T
 - NSX-T Manager, 1 x Controller & 1 x Edge and setup both the Management and Control Cluster Plane
 - Configure NSX-T with IP Pool, Transport Zone, add vCenter Server as Compute Manager, Create Logical Switch, Prepare ESXi hosts, Create Uplink Profile & Add configure ESXi hosts as a Transport Node
3. Creating the NSX-T virtual network
4. Taking a look at some of the capabilities of the distributed firewall

Here is a logical picture of how our nested lab is going to look:



Here is what all the VM's will look like from our vCenter:




1. Getting started

I like to start with a list of DNS names and IP addresses for each item that I'm going to create and add these names and IP in my DNS server, so I know they are ready as I deploy them.

Name	Type	Data
vcenter65-1	Host (A)	192.168.1.130
vesxi65-1	Host (A)	192.168.1.131
vesxi65-2	Host (A)	192.168.1.132
vesxi65-3	Host (A)	192.168.1.133
nsxt-mgr	Host (A)	192.168.1.135
nsxt-ctrl	Host (A)	192.168.1.136
nsxt-edge	Host (A)	192.168.1.139

I'm also going to make two portgroups: nsxt-mgmt and nsxt-nested, to help keep things organized. In both of these portgroups I need to Set Promiscuous Mode = Accept.

 nsxt-mgmt

 nsxt-nested

2. Deploy and configure vCenter Server Appliance 6.5u1

I'm not going to show all the details for installing the vCSA because it's a standard installation and instructions are well documented, but I will call out a few settings that we'll want to use.

- Mount the vCSA ISO and run the installer.exe located in the vcsa-ui-installer folder.
I'm using *VMware-VCSA-all-6.5.0-8815520.iso*
- Select the **Tiny** deployment size.
- When selecting storage, I like to Enable Thin Disk Mode to save on space in the lab
- Select **nsxt-mgmt** for the vCSA network
- Finish

Now is a good time to go grab something to drink as this will take several minutes.

- Proceed with stage 2
- Enter your NTP server IP
- Select the option to create a new SSO domain
- Finish

At this point I have a fully functioning vCenter appliance for my nested lab. Next, I'll deploy my nested ESXi hosts as VM's and add them to my new vCenter.

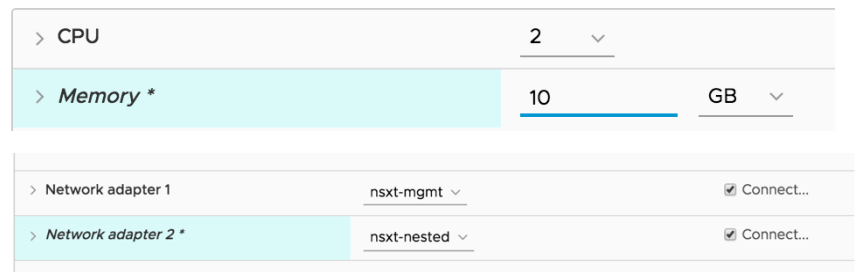
3. Deploy and configure 3 x Nested ESXi 6.5u2 Virtual Appliance VMs

When deploying nested ESXi hosts, I like to use the preconfigured OVA's that William Lam creates. The appliance allows anyone to quickly stand up a fully functional nested ESXi VM which includes guest customization such as networking, NTP, syslog, passwords, etc. You can download William's appliances [here](#).

- Select your physical ESXi host, then select Deploy OVF Template
I'm using
Nested_ESXi6.5u2_Appliance_Template_v1.ova
- Follow the prompts to fill in the appropriate information.
- When I get to the part on networks, select **nsxt-mgmt** as my destination network.
- Enter my IP information
- Finish

After the ESXi appliance VM has been deployed, I'm going to edit a couple settings

- Increase the memory to 10 GB
- Change the Network Adapter 2 to **nsxt-nested**
- Save and power on.

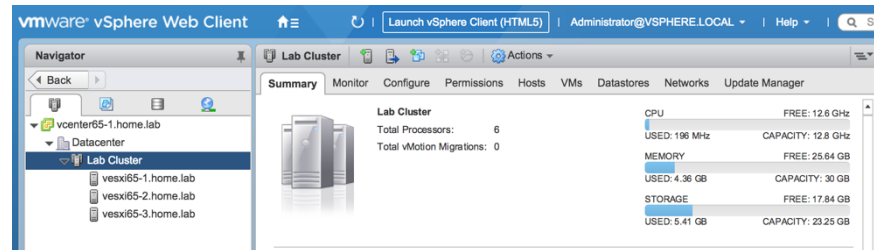


I follow the same steps two more times to deploy nested hosts **vesxi65-2** and **vesxi65-3**.

When I have my nested ESXi VM's deployed, I can login to my nested vCenter to add the hosts.

- Create a Datacenter
- Create Cluster
- Add hosts to the cluster

This is what my nested lab vCenter looks at this point:



4. Deploy NSX-T Manager, 1 x Controller & 1 x Edge

The NSX Manager is the centralized network management component of NSX, and is installed as a virtual appliance on an ESXi host. It provides an aggregated system view.

The NSX-T Manager is bundled as an OVA, so the initial deployment is like a typical VM.

- Select my physical ESXi host, then select Deploy OVF Template
I'm using *nsx-unified-appliance-2.3.0.0.0.10085405.ova*
- Select the **Small** deployment configuration
- I select **nsxt-mgmt** as my destination network.
- I follow the prompts to fill in the appropriate passwords and IP information.
- Finish

After a few minutes, I have a deployed NSX-T Manager. From here, the rest of the instructions will be done in the NSX-T Manager. Let's login to the NSX Manager.

Https://<NSX-t-Manager-IP> For me, my Manager IP is 192.168.1.135

5. Deploy the NSX-T controller

The controller is responsible for providing configuration to other NSX Controller components such as the logical switches, logical routers, and edge configuration. Traffic doesn't pass through the controller. Stability and reliability of data transport are central concerns in networking. To further enhance high availability and scalability, the NSX Controller is deployed in a cluster of three instances, but for my lab I'm only deploying a single controller which works fine.

First, I'll add a my main vCenter that is holding my nested ESXi hosts as a compute manager.

- From the main menu, select **Fabric**, then **Compute Managers**, then **Add**.
- Provide a name for this connect, vCenter IP, username and password, click **Add**.

New Compute Manager ⓘ ×

Name*

Description

Domain Name/IP Address*

Type*

Username*

Password*

SHA-256 Thumbprint

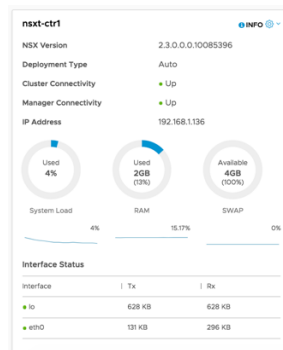
- When I see the Registration Status is Registered and the Connection Status is Up I can move on to deploying the controller.

<input type="checkbox"/>	Compute Manager ↑	ID	Domain Name/IP Addr	Type	Registration Status	Version	Connection Status
<input type="checkbox"/>	Main vCenter	e664_e6ec	192.168.1.246	vCenter	Registered	6.7.0	Up

Deploy the controller.

- From the main menu, select **System**, then **Components**, Select **Add Controllers**
- Select your Compute Manager that you just added.
- Enter passwords for all of the user accounts.
- Enter DNS and NTP settings
- Select **Medium** for the deployment size.
- Next
- I enter **nsxt-ctr1** as the hostname for my Controller.
- I select the cluster and datastore where I would like to deploy the Controller.
- Select the **nsxt-mgmt** network.
- I enter the Management IP/Netmask: **192.168.1.136/24**
- Enter the Management Gateway: **192.168.1.1**
- Finish

After a couple minutes your Controller should be deployed. Check to ensure the Cluster and Manager Connectivity shows as Up. *Note; after the Controller is deployed, to use less resources in my lab, I shut it down and change the CPU from 4 to 2 and Memory from 16 GB to 4 GB.*



6. Deploy an NSX-T Edge

The NSX Edge provides routing services and connectivity to networks that are external to the NSX-T deployment. In this example, the NSX Edge will route between my nested network and my regular network to allow connectivity for VM's running inside the nested lab.

To deploy an Edge:

- From the main menu, select **Fabric**, then select **Nodes**.
- Select **Add Edge VM**
- Provide a name, hostname and form factor of **small**
- Next
- Provide CLI and root password credentials
- Next
- I select the same **Compute Manager** that I used to deploy the Controller
- Select the Cluster and Datastore
- Next
- For IP Assignment, select **Static** and provide the **Management IP** with the netmask and the **Default Gateway**. This interface will only be used for managing the Edge and not used for network routing. In this example it's **192.168.1.139/24**
- The Management Interface is **nsxt-mgmt**.
- The data paths are additional Edge interfaces, which will be used for network routing.
 - #1 set to **nsxt-mgmt**
 - #2 set to **nsxt-mgmt**
 - #3 set to **nsxt-nested**

Add Edge VM

- 1 Name and Description
- 2 Credentials
- 3 Configure Deployment
- 4 Configure Ports

Configure Ports

IP Assignment* ⓘ ×

DHCP

Static

Management IP* ⓘ 192.168.1.139/24

Default Gateway ⓘ 192.168.1.1

Management Interface* nsxt-mgmt ▼

Did not find expected? Try refresh to fetch latest interfaces from System. ↻

Datapath Interfaces

#1* nsxt-mgmt ▼

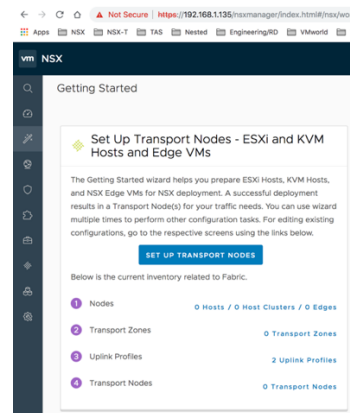
#2* nsxt-mgmt ▼

#3* nsxt-nested ▼

CANCEL
PREVIOUS
FINISH

7. Configure NSX-T

I'm going to use the Getting Started Setup Wizard to deploy NSX-t to my nested hosts. The Getting Started wizard helps you prepare ESXi Hosts, KVM Hosts, and NSX Edge VMs for NSX deployment. A successful deployment results in a Transport Node(s) for your traffic needs. You can use wizard multiple times to perform other configuration tasks.



My unused network interface will be the one that is connected to the **nsxt-nested** portgroup.

- Click Set Up Transport Node to start the process

We are presented with a couple options. I can select the host cluster option if I would like to prepare an entire cluster at one time which is the easiest. But I can also select the host option which allows me to deploy NSX-t without having to connect to a vCenter. I'm going to use the Host option.

IMPORTANT FOR HOST PREPARATION:

Each host, either standalone or in a cluster, must have at least one available and unused Physical NIC.

Select option below:

Host Cluster

Prepare a host cluster and link it to an Overlay Transport Zone

Host

Prepare a host and link it to an Overlay Transport Zone

NSX Edge VM (with an option to add it to an NSX Edge Cluster)

Prepare an NSX Edge VM and add it to an Overlay Transport Zone for East-West network traffic connectivity. If you want an external North-South network traffic connection, the NSX Edge VM must be added to the VLAN Transport Zone.

Add my first host.

- Name: **vesxi65-1**
- IP Address: **192.168.1.131**
- Operating System: **ESXi**
- Username: **root**
- I enter my Password
- Click Add

Add Host



Name *	vesxi65-1
IP Addresses *	192.168.1.131 <input type="text" value="Enter IP Addresses"/>
Operating System *	ESXi
Username *	root
Password *
SHA-256 Thumbprint	<div style="border: 1px solid #ccc; height: 40px;"></div>

CANCEL

ADD

A transport zone defines the potential reach of transport nodes.

The next step is to create an Overlay Transport Zone for East-West traffic.

- For simplicity I'm going to call the overlay **Overlay-TZ**
- Name the N-VDS **tswitch1**

Add Transport Zone

Name *	Overlay-TZ
Description	<input type="text"/>
N-VDS Name *	tswitch1
Traffic Type	<input checked="" type="radio"/> Overlay

Next, I need to create an uplink profile. I'm going to call it **Overlay-Uplink-Profile**

- I leave the teaming policy at **Failover Order**
- MTU at **1600**.
- In my particular deployment, my nested ESXi hosts only have 1 unused network interface card, so I've only defined one an Active Uplink and no Standby Uplinks.


The Active Uplink I called **Uplink1**.

Add Uplink Profile ⊗ ×

Name *	Overlay-Uplink-Profile
Description	<input type="text"/>
Teaming Policy *	Failover Order

LAGs

+ ADD DELETED

<input type="checkbox"/>	Name *	LACP Mode	LACP Load Balancing *	Uplinks	LACP Time Out
 <p>No LAGs Added Yet!</p>					

Active Uplinks *	Uplink1
Standby Uplinks	Enter Standby Uplinks (Example: link1, link2)
Transport VLAN	0
MTU *	1600

The next thing I do is define my new N-VDS switch to provide connectivity for my Transport Nodes. When I add the N-VDS, NSX-t will create Tunnel Endpoints (TEP) on each of the ESXi hosts. The TEP is a Kernel interface and needs to have an IP. I have the option to use DHCP or use an IP Pool. I like to know exactly which IP's are going to be used for my TEP's, so I'm going to create a new IP Pool.

- Select Create IP Pool.
- Name: **ESXi-TEP-Pool**
- Provide IP Ranges: **192.168.1.80 – 84**
- CIDR format, **192.168.1.0/24**
- My DNS server is **192.168.1.205**
- My DNS suffix is **home.lab**

Add IP Pool ⓘ ×

Name *

Description

Subnets

+ ADD DELETE

<input type="checkbox"/>	IP Ranges *	Gateway	CIDR *	DNS Servers	DNS Suffix
<input type="checkbox"/>	192.168.1.80- 192.168.1.84	192.168.1.1	192.168.1.0/24	192.168.1.205	home.lab

I'll use my new Uplink profile and assign my Active Uplink **Uplink1** to the host physical network interface card (PNIC).

It should look something like this:

Link To Transport Zone East-West

1 IP Assignment

Assignment IP Address Use IP Pool

IP Pool ESXI-TEP-Pool

[CREATE IP POOL](#)

Details

ESXI-TEP-Pool

IP Ranges	Gateway	CIDR	DNS Servers	DNS Suffix
192.168.1.80-192.168.1.84	192.168.1.1	192.168.1.0/24	192.168.1.205	home.lab

2 Host NIC Connections

Host NIC connection defines how the uplink ports on the N-VDS connect to the physical uplink profiles on the host.

Host PNIC	Host Uplink
vmnic1	<u>Uplink1</u>

After my first host is completed, I will need to do the same thing for hosts **vesxi65-2** and **vesxi65-3**. Note; If I had more than a couple hosts it would be faster to add the nested vCenter as a Compute Manager and use the cluster option to deploy to all of the hosts within a cluster.

When all of my hosts are completed, I can check the status. Notice the Controller and Manager Connectivity for all hosts is Up.

- From the main menu, select **Fabric**, then **Nodes**.

Host	ID	IP Addresses	OS Type	OS Version	Deployment Status	NSX Version	Controller Connectivity	Manager Connectivity	Transport Node (TN)
Lab Cluster (3)	MoRef ID: domain...								
vesxi65-1	2600...4cb6	192.168.1.131	ESXi	6.5.0	NSX Installed	2.3.0.0.0.10085378	Up	Up	2600be17-35a0-4b54-921...
vesxi65-2	4d5d...aaa3	192.168.1.132	ESXi	6.5.0	NSX Installed	2.3.0.0.0.10085378	Up	Up	4d5d98af-3d82-4c29-a845...
vesxi65-3	83e7...b6c9	192.168.1.133	ESXi	6.5.0	NSX Installed	2.3.0.0.0.10085378	Up	Up	83e7b5c0-17bc-4771-96fd...

Note: I don't like the Transport Node names being the auto generated GUID, so I changed them by selecting the Transport Node ID name, then selecting Edit, and change the

Name. I changed mine to ESXi1-TN, ESXi2-TN and ESXi3-TN so that when I saw the name on other screens I would know what they were.

I'm going to add my Edge as a Transport Node. An NSX Edge transport node can belong to multiple transport zones: One overlay transport zone and multiple VLAN transport zones. VLAN transport zones are for the VLAN uplinks to the outside world.

I need to configure a Transport Zone for my Edge.

- From the main menu, select **Fabric**, then **Transport Zones**
- Select **Add**
- Provide the new transport zone name: **VLAN-TZ**
- Provide a name for the N-VDS switch: **tswtich2**
- I make sure to select the traffic type as **VLAN**.

New Transport Zone ? ×

Name*

Description

N-VDS Name*

N-VDS Mode

Standard

Enhanced Datapath

Traffic Type

Overlay

VLAN

Uplink Teaming Policy Names

Now I'll add the Edge to my Transport Zones

- From the main menu, select **Fabric**, then select **Nodes**.
- I select my **Edge-01** Edge VM
- From the **Actions** pull down, select **Configure as Transport Node**.

Nodes

Hosts **Edges** Edge Clusters ESXi Bridge Clusters Transport No

Edge	ID	IP
Edge-01	136e...7eab	139

- Provide a name, and select both the **Overlay-TZ** and the **VLAN-TZ**
- Select N-VDS at the top to configure the Edge connections
- Edge Switch Name, select **tswitch1**
- Uplink Profile, select **Overlay-Uplink-Profile**
- IP Assignment, Select **Use IP Pool**
- IP Pool, Select **ESXi-TEP-Pool**
- Virtual NIC's, select **fp-eth2** (make sure the MAC address matches the interface that you have connected to nsxt-nested) and **Uplink1**
- Do not hit Add yet, select **Add N-VDS**
- Edge Switch Name, select **tswitch2**
- Uplink Profile, select **Overlay-Uplink-Profile**
- Virtual NIC's, select **fp-eth0** and **Uplink1**
- Click Add

When I've completed the Edge configuration I want to check the status. I can check the status of all of the Host and Edge Transport Nodes.

- From the main menu, select **Fabric**, then **Nodes**. I see Controller and Manager Connectivity as Up. Also notice the Edge has two N-VDS and Transport Zones

Nodes

Hosts Edges Edge Clusters ESXi Bridge Clusters **Transport Nodes**

+ ADD EDIT DELETE ACTIONS View All

Transport Node	ID	N-VDS	Configuration State	Status	IP Addresses	Fabric Node Type	Transport Zones	NSX Version
ESXi1-TN	2600...4cb6	1	Success	Up	192.168.1.131	Host - ESXi 6.5.0	Overlay-TZ	2.3.0.0.0.1008...
ESXi2-TN	4d5d...aaa3	1	Success	Up	192.168.1.132	Host - ESXi 6.5.0	Overlay-TZ	2.3.0.0.0.1008...
ESXi3-TN	83e7...b6c9	1	Success	Up	192.168.1.133	Host - ESXi 6.5.0	Overlay-TZ	2.3.0.0.0.1008...
Edge-01	136e...7eab	2	Success	Up	192.168.1.139	Edge - Virtual Machine	Overlay-TZ VLAN-TZ	2.3.0.0.0.1008...

There is one more thing I need to complete before I can start playing around with the NSX. I need to add the Edge Node into an Edge Cluster. The NSX-T Edge cluster is a logical grouping of NSX-T Edge virtual machines.

- From the main menu, select **Fabric**, then **Nodes**
- Select **Edge Cluster**
- Select **Add**
- Provide a cluster Name, **Edge-Cluster**
- Under Transport Nodes, Select **Edge-01**
- Click Add.

Nodes

Hosts Edges **Edge Clusters** ESXi Bridge Clusters Transport Nodes

+ ADD EDIT DELETE ACTIONS Search

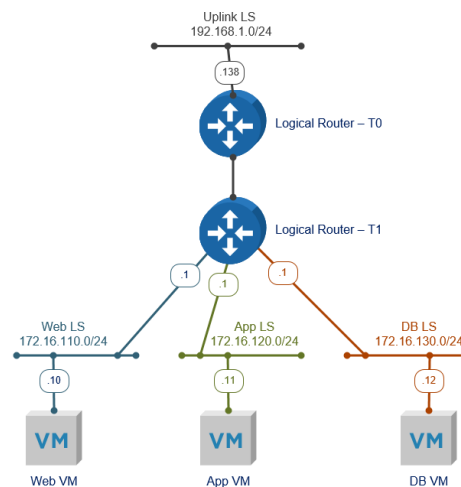
Edge Cluster	ID	Member Type	Cluster Profile	Transport Nodes
Edge-Cluster	d465...003d	Edge Node	nsx-default-edge-high...	1

This completes the installation of NSX-T in my nested lab. NSX is security platform and it's also a networking platform. You can use it for one or the other or both. From here out I'm going to look at examples of both capabilities. I'll create some logical switching and routing setup for a basic 3-tier application, and provide some firewall examples.

8. Virtual Networking

Now let's take a look at the software defined networking capabilities.

Here is what my 3-tier application is going to look like.



Create my Logical Switches

A logical switch provides a representation of Layer 2 switched connectivity across many hosts

- From the main menu, select **Networking**, then **Switches**
- Select **Add**
- Provide a name for the logical switch, **Web-LS**
- For the Transport Zone, select **Overlay-TZ**
- Click **Add**
- Do the same thing for the **App-LS** and **DB-LS**

When it comes to doing the Uplink logical switch, there is one small difference, the Transport Zone.

- Select **Add**
- Provide a name for the logical switch, **Uplink-LS**
- For the Transport Zone, select **VLAN-TZ**
- Provide the VLAN of my **nsxt-mgmt** portgroup
- Click **Add**

When I'm complete it looks like this. Notice the Uplink Transport Zone is VLAN-TZ and not Overlay-TZ like the others.

+ ADD EDIT DELETE ACTIONS ▾							
Search							
<input type="checkbox"/>	Logical Switch ↑	ID	Admin Status	Logical Ports	Traffic Type	Config State	Transport Zone
<input type="checkbox"/>	App-LS	c8d9...df63	● Up	0	Overlay : 71681	Success	Overlay-TZ
<input type="checkbox"/>	DB-LS	820d...d449	● Up	0	Overlay : 71682	Success	Overlay-TZ
<input type="checkbox"/>	Uplink-LS	2b67...cd8a	● Up	0	VLAN : 0	Success	VLAN-TZ
<input type="checkbox"/>	Web-LS	2b07...44ad	● Up	0	Overlay : 71680	Success	Overlay-TZ

Create the Logical Router - TO

The north end of Tier-0 interfaces with the physical network, and is where dynamic routing protocols can be configured to exchange routing information with physical routers. The south end of Tier-0 connects to multiple Tier-1 routing layer(s) and receives routing information from them.

- From the main menu, select **Networking**, then **Routers**
- Select **Add**, then **Tier-0 Router**
- Name the router, **TO-Router**
- The Edge Cluster, select our **Edge-Cluster**
- Click Add.

Configure the TO-Router.

- Select the **TO-Router**, then select **Configuration**, then **Router Ports**.
- Select **Add**
- Provide a router port name. I'm going to use **Net-192** because it connects to my 192.168 network.
- Type is **Uplink**
- MTU **1500**
- Transport Node, select **Edge-01**
- URPF Mode, **None**
- Logical Switch, select **Uplink-LS**
- Logical Switch Port, attach to new switch port, I gave it the name **Net-192-port**
- IP Address / mask, **192.168.1.138/24**

- Click Add

When Completed it looks like this, and if all worked correctly, I should be able to ping the 192.168.1.138 interface IP. Yep, the ping works.

Logical Rout	ID	Type	IP Address/mask	Connected To	Transport Node	Relay Service	Statistics
LinkedP...	15e0...8...	Linked ...	100.64.128.0/31	T1-Router			
Net-192	a5d6...0...	Uplink	192.168.1.138/24	Uplink-LS (Net-192-port)	Edge-01		

Create the Logical Router – T1

Southbound, the Tier-1 routing layer interfaces with the logical switches defined, and provides one-hop routing function between them.

- From the main menu, select **Networking**, then **Routers**
- Select **Add**, then **Tier-1 Router**
- Name the router, **T1-Router**
- Tier-0 Router, Select **T0-Router**
- The Edge Cluster, select our **Edge-Cluster**
- Edge Cluster Members, Select **Edge-01**
- Click **Add**.

Configure the T1-Router.

- Select the **T1-Router**, then select **Configuration**, then **Router Ports**.
- Select **Add**
- Provide a router port name. I'm going to use **Net-110** because it connects to my 172.16.110.0 network.
- Type is **Downlink**
- URPF Mode, **None**
- Logical Switch, select **Web-LS**
- Logical Switch Port, attach to new switch port, **Net-110-port**

- IP Address / mask, **172.16.110.1/24**
- Click Add
- Do the same thing for the other logical switches, **App-LS** with the IP **172.16.120.0/24** and **DB-LS** with the IP **172.16.130.0/24**.

When completed it should look like this:

Logical Rout	ID	Type	IP Address/mask	Connected To	Transport Node	Relay Service	Statistics
LinkedP...	868a...a...	Linked ...	100.64.128.1/31	T0-Router (LinkedPort_T1-Rout...	Edge-01		
Net-110	6de7...e...	Downlink	172.16.110.1/24	Web-LS (Net-110-port)			
Net-120	a732...0...	Downlink	172.16.120.1/24	App-LS (Net-120-port)			
Net-130	75df...e...	Downlink	172.16.130.1/24	DB-LS (Net-130-port)			

The last thing I need to do on the T1 Router is to enable route advertisement for the logical switches that are connected to it.

- Select the **T1-Router**, then select **Routing**, then **Route Advertisement**.
- Select **Edit**
- Set Status to **Enabled**
- Advertise All NSX Connected Routes should be turned to **Yes**.

T1-Router			
Overview	Configuration ▾	Routing ▾	Services ▾
Route Advertisement EDIT			
Status		● Enabled	
Advertise All NSX Connected Routes		● Yes	
Advertise All NAT Routes		● No	
Advertise All Static Routes		● No	
Advertise All LB VIP Routes		● No	
Advertise All LB SNAT IP Routes		● No	

At this point if everything is successful I should be able to ping the interfaces for each of the logical switches.

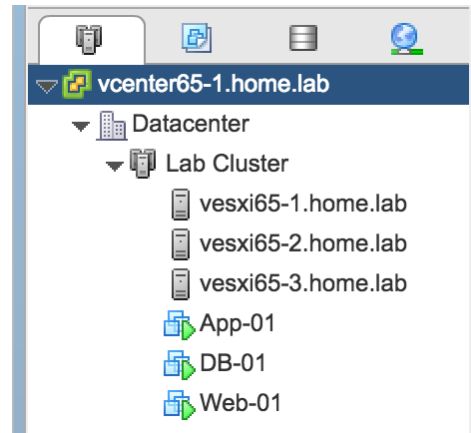
- Ping 172.16.110.1
- Ping 172.16.120.1
- Ping 172.16.130.1

9. Security, The Distributed Firewall (DFW)

The NSX Distributed Firewall is a hypervisor kernel-embedded firewall that delivers close to line rate throughput and provides a scale-out architecture that automatically extends firewall capacity when additional hosts are added. The Distributed Firewall provides each VM network interface with their own firewall. Micro-segmentation via the DFW is important because it decreases the network attack surface. If a breach occurs, micro-segmentation reduces the potential impact and lateral movement of a hacker or malware.

For this section I've added 3 VM's to create firewalls with.

- Web-01 = 172.16.110.10
- App-01 = 172.16.120.11
- DB-01 = 172.16.130.12



I've done a ping test to confirm each VM is responding on as it should.

```

Command Prompt

C:\Users\admin>ping 172.16.110.10

Pinging 172.16.110.10 with 32 bytes of data:
Reply from 172.16.110.10: bytes=32 time=3ms TTL=62
Reply from 172.16.110.10: bytes=32 time=2ms TTL=62
Reply from 172.16.110.10: bytes=32 time=2ms TTL=62
Reply from 172.16.110.10: bytes=32 time=2ms TTL=62

Ping statistics for 172.16.110.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 3ms, Average = 2ms

C:\Users\admin>ping 172.16.120.11

Pinging 172.16.120.11 with 32 bytes of data:
Reply from 172.16.120.11: bytes=32 time=3ms TTL=62
Reply from 172.16.120.11: bytes=32 time=3ms TTL=62
Reply from 172.16.120.11: bytes=32 time=3ms TTL=62
Reply from 172.16.120.11: bytes=32 time=2ms TTL=62

Ping statistics for 172.16.120.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 3ms, Average = 2ms

C:\Users\admin>ping 172.16.130.12

Pinging 172.16.130.12 with 32 bytes of data:
Reply from 172.16.130.12: bytes=32 time=3ms TTL=62
Reply from 172.16.130.12: bytes=32 time=2ms TTL=62
Reply from 172.16.130.12: bytes=32 time=2ms TTL=62
Reply from 172.16.130.12: bytes=32 time=3ms TTL=62

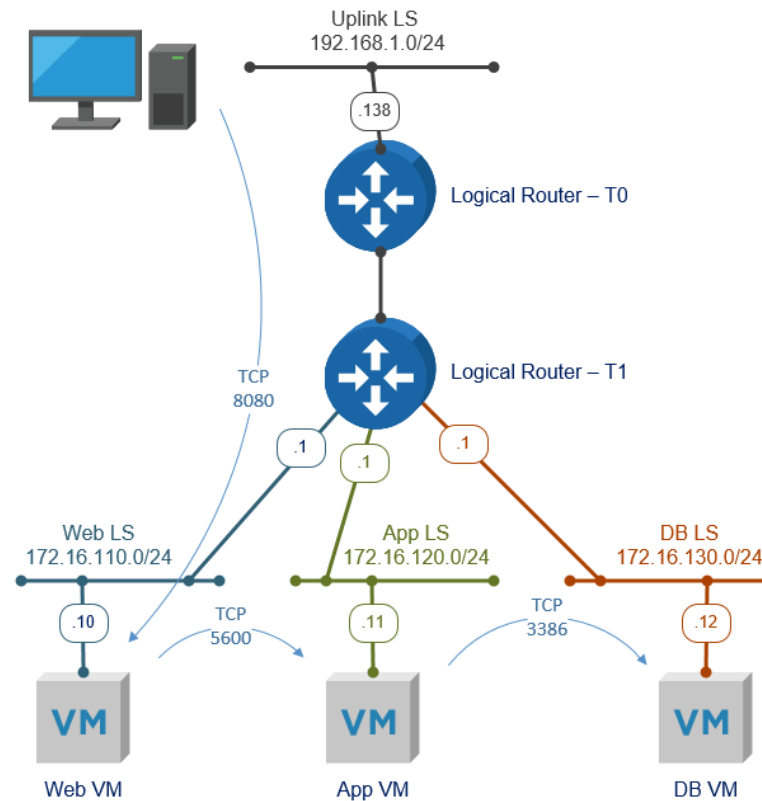
Ping statistics for 172.16.130.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 3ms, Average = 2ms

C:\Users\admin>

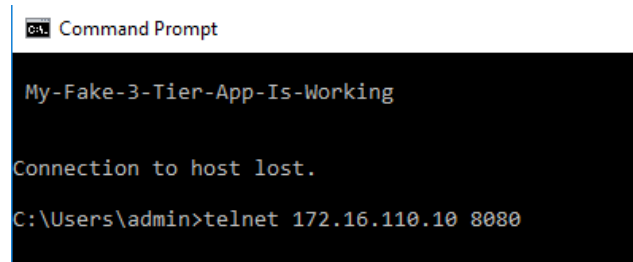
```

I've also setup a small application that can be easily configured to communicate on multiple ports. It's not a very sophisticated application but it allows me to verify communication sessions between VM's on different ports.

- My desktop talks to Web-01 on TCP 8080
- Web-01 talks to App-01 on TCP 5600
- App-01 talks to DB-01 on TCP 3386



Without the firewall running, when I Telnet to Web-01 on port 8080, I get a response back of “My-Fake-3-Tier-App-Is-Working”



```
Command Prompt
My-Fake-3-Tier-App-Is-Working
Connection to host lost.
C:\Users\admin>telnet 172.16.110.10 8080
```

Let's take a look at how the distributed firewall works. Each individual firewall rule contains instructions that determine whether a packet should be allowed or blocked; which protocols it is allowed to use; which ports it is allowed to use and so forth.

NSX has multiple ways to include VM's in firewall rules. I'm going to use multiple ways for the purpose of demonstration rather than suggesting this is the best way to create rules.

Firewall Section

Firewall sections are used to group a set of firewall rules. A firewall section can be made up of one or more individual firewall rules. Let's create a section for our application.

- From the main menu, select **Security**, then **Distributed Firewall**
- Select the **Default Layer3 Section**, then select **Add Section**, and **Add Section Above**.
- Provide a name for the section, **3 Tier App**
- Click **OK**

Now that I have a new section. Let's create some rules.

- Select our new section, **3 Tier App**, then select the **Add Rule**. We are going to create three rules, so click Add Rule three times.

First rule

- Name, **Any to Web**
- Source, **Any**
- Destination, select to **Edit Rule Destination**
 - Object Type, select **Logical Switch**
 - Highlight **Web-LS**
 - Click **OK**

Specify Destination | Any to Web

Specify destination for the rule. You can provide container objects or IP addresses to which the communication is targeting.

Negate Destination: Off

Container Objects (1) IP Addresses (0)

Select one or more objects for the destination field to the firewall rule

Object Type: Logical Switch

Available Objects

Filter

<input type="checkbox"/>	Name
<input type="checkbox"/>	↔ App-LS
<input type="checkbox"/>	↔ DB-LS
<input type="checkbox"/>	↔ Uplink-LS
<input type="checkbox"/>	↔ Web-LS

1 - 4 of 4 objects

Selected Objects

<input type="checkbox"/>	Name	Object Type
<input type="checkbox"/>	↔ Web-LS	Logical Switch

Max limit: 128 1 Objects

OK

CANCEL

- Service, select **Edit Rule Service**
- Select either an existing Service or create a New Raw Protocol. In my example I'm going to create **New Raw Protocol**.
 - Type of Service: **L4 PortSet**
 - Protocol: **TCP**
 - Source Ports: **8080**
 - Click **Add**
- Make sure the Action is set to **Allow**

Second Rule

- Name, **Web to App**
- Source, Select the **Web-LS** logical switch
- Destination, select to **Edit Rule Destination**
- Object type: **IP Set**
- At the bottom, I select **Create New IP Set**
 - Name: **App IPs**
 - Address, select **Add**, Enter our App-01 VM IP, **172.16.120.11**
 - Click **OK**
- Click **OK**

Add New IP Set

Name *

Description

Address

[+ ADD](#) | [DELETE](#)

<input type="checkbox"/>	IP Addresses
<input type="checkbox"/>	<u>172.16.120.11</u>

- Service, select **Edit Rule Service**
- Select either an existing Service or create a New Raw Protocol. In my example I'm going to create **New Raw Protocol**.
 - Type of Service: **L4 PortSet**
 - Protocol: **TCP**
 - Source Ports: **5600**
 - Click **Add**
- Make sure the Action is set to **Allow**

Third Rule

- Name, **App to DB**
- Source, Select the **App IPs** IP Set

- Destination, select to Edit Rule Destination
- Object type: **NSGroup**
- At the bottom, lets select Create New NSGroup
 - Name: **DB VMs**
 - Next
 - Select **Add Membership Criteria**
 - Virtual Machine, Name, Contains, **DB**
 - Next
 - Don't select anything on the Members page, Click **Finish**
- Click **OK**
- Service, select **Edit Rule Service**
- Select either an existing Service or create a New Raw Protocol. Again, in my example I'm going to create **New Raw Protocol**.
 - Type of Service: **L4 PortSet**
 - Protocol: **TCP**
 - Source Ports: **3386**
 - Click **Add**
- Make sure the Action is set to **Allow**

When I'm completed with all the rules, it should look like this:

#	Name	Source	Destination	Service	Applied To	Log	Action	Popularity Index
1	Any to Web ID: 1026	Any	Web-LS	TCP	Distributed Firew...	Off	Allow	
2	Web to App ID: 1027	Web-LS	App IPs	TCP	Distributed Firew...	Off	Allow	
3	App to DB ID: 1028	App IPs	DB VMs	TCP	Distributed Firew...	Off	Allow	
4	Default Layer3 Rule ID: 2	Any	Any	Any	Distributed Firewall	Off	Drop	

- Select **Publish** to write the rules to the distributed firewalls.

As soon as you hit publish changes, only the traffic that was defined will be allowed to access the VM.

Very cool.

As you can see, with only minimal investments and resources you can get a nested lab stood up in your environment. Performance of a nested lab probably won't match that of

running on physical hardware but that is not the objective here, being able to experiment with products is the goal. As you gain experience you can grow or shrink the nested lab as your requirements evolve. There are many advantages in having a safe place to test and learn features, and the overall return on value for a company is phenomenal.

Happy computing!

About the author:

Jim is an experienced NSX, vSAN and virtualization specialist, having spent over 25 years working with several companies in technology roles from operations to leadership. As an NSX Technical Account Specialist (TAS) at VMware Jim has collaborated on deploying multiple software-defined data centers at various sized companies across several industries. Prior to joining VMware, he was a Virtualization Architect for a global news, media and information organization.

For more information on how a VMware NSX Technical Account Specialist can help drive business outcomes from your VMware NSX deployment, please contact your VMware account representative.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Copyright © 2018 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. and its subsidiaries in the United States and other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

