

VMware Cloud Experience: Software Defined Networking and Security - Holodeck

Table of contents

VCF Experience Program Lab Overview.....	3
VCF Experience Program: Software Defined Networking and Security.....	3
Module 1: Segments and distributed routing.....	4
Lab 1: Creating Network Segments	4
Lab 2: View packet flow within a host	14
Lab 3: View packet flow between hosts	17
Lab 4: Adding router connectivity	21
Lab 5: Testing the application	30
Module 2: Changing the Security Game with Microsegmentation	32
Lab 1 Tagging VMs and Grouping Workloads based on Tags	34
Lab 2: Applying Distributed Firewall Rules Based on Tagging on a segment	42
Module 3: Load Balancing.....	64
Lab 1: Configure load balancer	64

VCF Experience Program Lab Overview

The VMware Cloud Foundation (VCF) Experience Program is designed to provide a hands-on experience highlighting how VCF delivers a *Cloud Operating Model* for customer managed on-premises environments, capable of hosting traditional and modern applications. This Experience Program guide is intended for use with a VCF Lab Constructor (VLC) based nested environment built using the Automated Holodeck config.

Credentials

The following credentials are used in this lab. For your convenience, links to all management interfaces are in the bookmark bar or in the MGMT Domain folder of Google Chrome in your lab environment.

- **SDDC Manager**
 - Username: administrator@vsphere.local
 - Password: VMware123!
- **vCenter Server Admin Console**
 - Username: root
 - Password: VMware123!
- **vSphere Web Client**
 - Username: administrator@vsphere.local
 - Password: VMware123!
- **VMware NSX Manager**
 - Username: admin
 - Password NSX-T: VMware123!VMware123!
- **vRealize Operations Manager**
 - Username: admin
 - Password: VMware123!
- **vRealize Automation Cloud Assembly**
 - Username: configadmin
 - Password: VMware123!
- **Windows Console (Jump Host)**
 - Username: administrator
 - Password: VMware1!
- **Opencart Apache and MySQL VMs**
 - Username: ocuser
 - Password: VMware123!

VCF Experience Program: Software Defined Networking and Security

Overview

This session provides an understanding of the fundamentals of Software Defined Networking and Security provided by VMWare NSX. The modules in this lab focus on the simplicity of virtualizing network functions in software versus traditional hardware approaches. Network and security functions, once decoupled from their hardware counterparts (switches, routers, and firewalls), can be leveraged on a per-application basis rather than depend on their physical location.

What will you learn in this session? Module breakdown:

- Module 1 – Introducing Software Defined Networking: Segments and Distributed Routing
- Module 2 – Changing the Security Game – Distributed Firewall
- Module 3 – Load Balancing

Module 1: Segments and distributed routing

Software Defined Networking in VMware Cloud Foundation is provided by VMware NSX. NSX operates as an “Overlay Network”, where the networking capabilities are delivered in software, and “encapsulated” within standard TCP/IP packets transported by a standard IP “underlay” network.

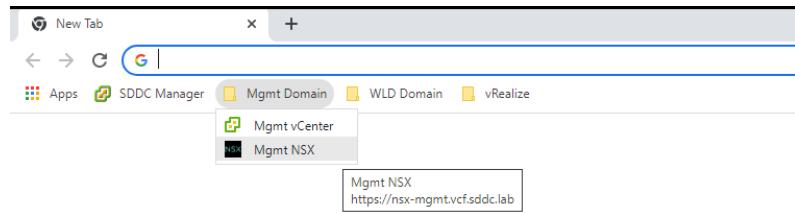
NSX enables customers to create elastic, logical networks that span physical network boundaries. NSX abstracts the physical network into a pool of capacity and separates the consumption of these services from the underlying physical infrastructure. This model is similar to the model vSphere uses to abstract compute capacity from the server hardware to create virtual pools of resources that can be consumed as a service.

Lab 1: Creating Network Segments

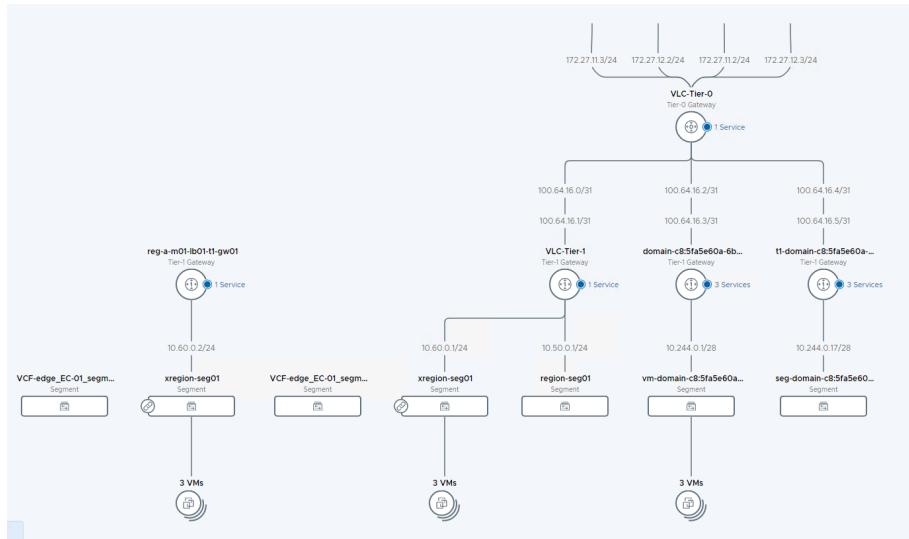
This exercise will deploy the necessary networking components to support a simple two-tier application called “Opencart”. [Opencart](#) is an opensource E-Commerce platform that uses an Apache front end, and a MySQL backend. This lab uses preconfigured Apache and MySQL VM’s that we will attach to newly created SDN segments.

[Step 1] Logging in to the environment

- A. Open a new tab in the Chrome browser
- B. Click the Mgmt Domain folder in the bookmark bar then select Mgmt NSX



- C. Click advanced / proceed to nsx-mgmt.vcf.sddc.lab, if required to accept the certificate
- D. Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
- E. From the NSX-T Manager interface click the **Networking** tab
- F. Select **Network Topology**



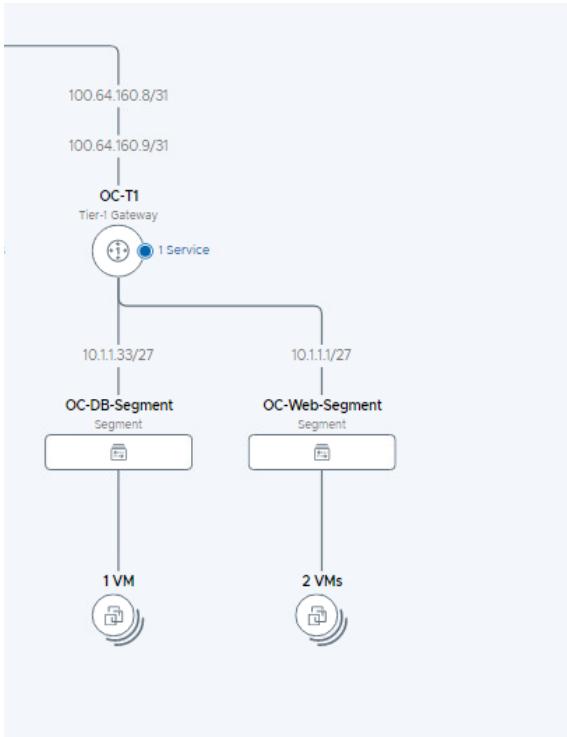
What you see are existing segments and routers that are used elsewhere in the lab environment. By the end of this lab exercise, you create the configuration shown below, with 2 new segments, connected to a Tier-1 gateway, with a load balancer and firewall configured.

OC-Web-Segment

- 10.1.1.1/27
- Gateway 10.1.1.1/27
- OC-Apache-A 10.1.1.18
- OC-Apache-B 10.1.1.19
- OC-LB 10.1.1.2 (Load balancer)

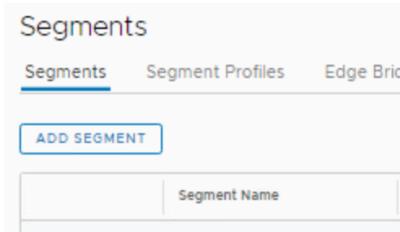
OC-DB-Segment

- 10.1.1.32/27
- Gateway 10.1.1.33/27
- OC-MySQL 10.1.1.50



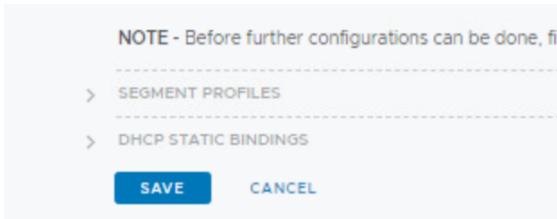
[Step 2a] Create the OC-DB-Segment

- From the NSX-T Manager interface click the **Networking** tab at the top of the screen
- Click **Segments** in the left pane.
- Click **ADD SEGMENT** button

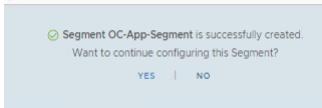


- In the **Segment Name** field, enter **OC-DB-Segment**
- Leave the **Connected Gateway** field blank
- In the **Transport Zone** dropdown, select **mgmt-domain-tz-overlay01 | Overlay**
- Add IPv4 gateway **10.1.1.33/27**

- H. All other settings should remain default – scroll to bottom, click **SAVE**



- I. You will see your segment has been successfully created. Click **NO** on the Want to continue configuring this segment?

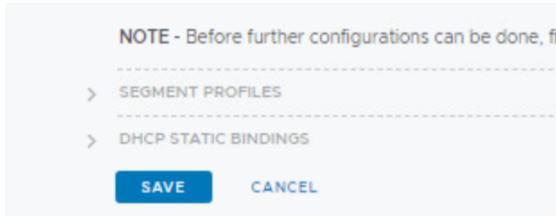


[Step 2b] Create the OC-Web-Segment

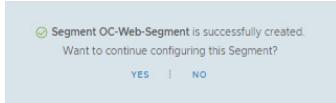
- From the NSX-T Manager interface click the **Networking** tab at the top of the screen
- Click **Segments** in the left pane.
- Click **ADD SEGMENT** button

- In the **Segment Name** field, enter **OC-Web-Segment**
- Leave the **Connected Gateway** field blank
- In the **Transport Zone** dropdown, select **mgmt-domain-tz-overlay01 | Overlay**
- Add **IPv4 gateway 10.1.1.1/27**

- H. All other settings should remain default – scroll to bottom, click **SAVE**



- I. You will see your segment has been successfully created. Click **NO** on the Want to continue configuring this segment?



[Step 3] Connect Opencart web server VMs

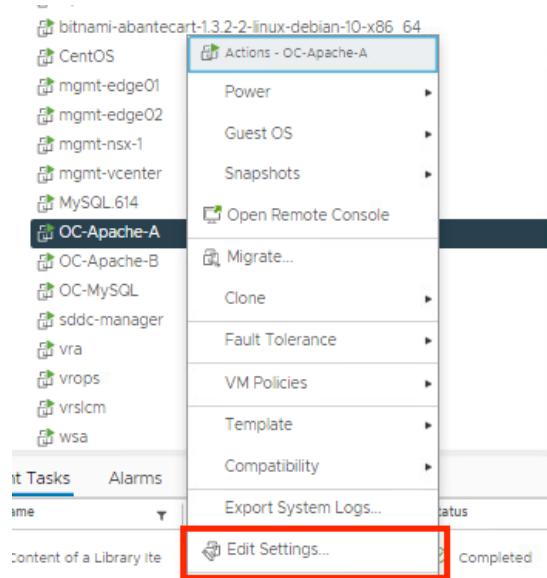
This step will attach our two Apache web server VM's to the OC-Web-Segment. Begin in the vCenter Server Web Interface:

- Click the **Management vCenter on** the bookmark bar in the “Mgmt Domain” folder
- Click **Launch vSphere Client** if prompted
- Log onto vSphere Client as the user: **administrator@vsphere.local** with the password: **VMware123!**

[Step 3.1] Attach OC-Apache-A to segment OC-Web-Segment

From the vCenter Server Hosts and Clusters view, find **OC-Apache-A** in the left-side scroll-list.

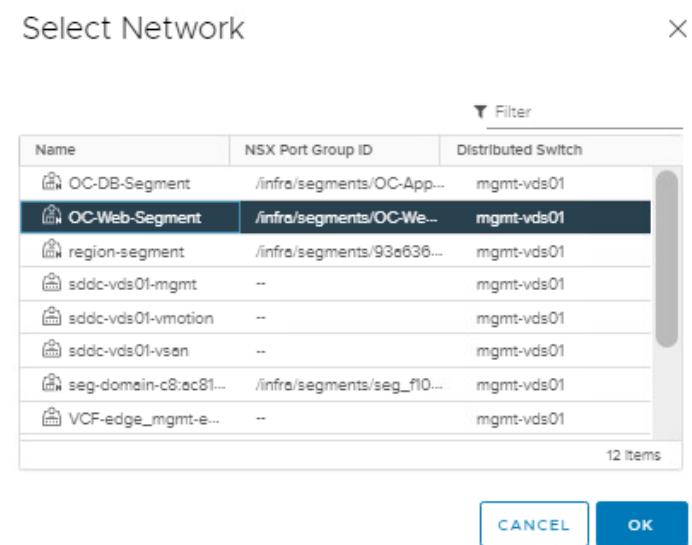
- Right click on **OC-Apache-A** and click **Edit Settings**



B. Click the dropdown next to Network Adapter 1

The screenshot shows the 'Edit Settings' dialog for the VM 'OC-Apache-A'. The 'Virtual Hardware' tab is selected. Under 'CPU', there are dropdowns for '1' and 'GB'. 'Memory' is set to '2 GB'. 'Hard disk 1' is set to '10 GB'. 'SCSI controller 0' is set to 'LSI Logic Parallel'. The 'Network adapter 1' dropdown is set to 'sddc-vds01-mgmt' and has a checked 'Connected' checkbox. Other options like 'Video card' and 'SATA controller 0' are also visible. A red box highlights the 'Network adapter 1' dropdown.

- C. Click **Browse** (Notice the network segment that you just created in NSX is now visible in vSphere)**
D. Click **OC-Web-Segment**



E. Click **OK**

F. Click **OK**

[Step 3.2] Attach OC-Apache-B to OC-Web-Segment.

- Right click on **OC-Apache-B** and click **Edit Settings**
- Click the dropdown next to **Network Adapter 1**
- Click **Browse** click on **OC-Web-Segment**
- Click **OK** to close the Menu and **OK** again to close the Settings.

[Step 3.3] Test basic connectivity – OC-Apache-B

With 2 VM's on the segment we can test connectivity between the VM's. IP Assignment is as follows:

- OC-Apache-A – 10.1.1.18
 - OC-Apache-B – 10.1.1.19
- Open a web console on OC-Apache-B. You may need to hit enter in the window to get a login prompt



B. Login as **ocuser** password **VMware123!**

```
Ubuntu 18.04.6 LTS OC-Apache-B tty1
Hint: Num Lock on
OC-Apache-B login: ocuser
Password:
```

C. Check the interface configuration with **ifconfig**

```
ocuser@OC-Apache-B:~$ ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.1.1.19 netmask 255.255.255.224 broadcast 10.1.1.31
                inet6 fe80::2a0:3ff:fe82:a87d prefixlen 64 scopeid 0x20<link>
                      ether 00:50:56:82:a8:7d txqueuelen 1000 (Ethernet)
                        RX packets 193156 bytes 12331580 (12.3 MB)
                        RX errors 0 dropped 18 overruns 0 frame 0
                        TX packets 32942 bytes 1402498 (1.4 MB)
                        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                inet6 ::1 prefixlen 128 scopeid 0x10<host>
                      loop txqueuelen 1000 (Local Loopback)
                        RX packets 88068 bytes 7605858 (7.6 MB)
                        RX errors 0 dropped 0 overruns 0 frame 0
                        TX packets 88068 bytes 7605858 (7.6 MB)
                        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ocuser@OC-Apache-B:~$ _
```

D. Test connectivity with OC-Apache-A (ping 10.1.1.18).

```
ocuser@OC-Apache-B:~$ ping 10.1.1.18
PING 10.1.1.18 (10.1.1.18) 56(84) bytes of data.
64 bytes from 10.1.1.18: icmp_seq=1 ttl=64 time=2.05 ms
64 bytes from 10.1.1.18: icmp_seq=2 ttl=64 time=0.439 ms
64 bytes from 10.1.1.18: icmp_seq=3 ttl=64 time=0.557 ms
64 bytes from 10.1.1.18: icmp_seq=4 ttl=64 time=0.452 ms
^C
--- 10.1.1.18 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3030ms
rtt min/avg/max/mdev = 0.439/0.875/2.055/0.683 ms
ocuser@OC-Apache-B:~$ -
```

- E. Notice we can communicate from OC-Apache-B to OC-Apache-A on a network we just created.

[Step 3.4] View NSX OC-Web-Segment in vCenter Server

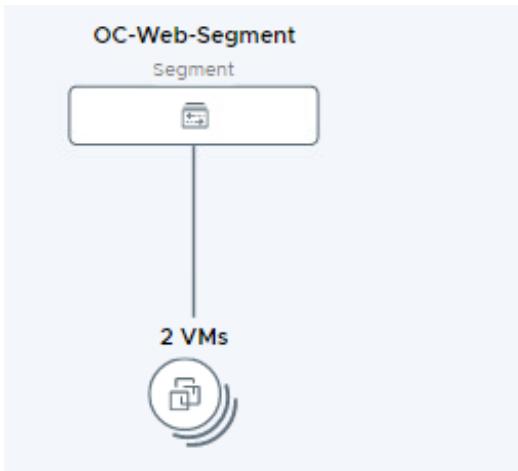
- Click on the Management vCenter Server Tab
 - Click Menu -> Networking
 - Click on OC-Web-Segment
 - Notice the "N" showing this is an NSX segment versus a standard port group
 - The segment ID is shown along with the transport zone the segment is a part of
 - Lower on the screen you can see which VDS this segment is configured on,
- Note: With vSphere 7 and NSX-T 3.1 and higher versions, NSX segments are an extension of the vSphere Distributed Switch and are completely visible to the vSphere team*
- Finally notice the NSX Manager for this segment is hot linked

- Click on the Ports tab
- Notice there is a port per virtual machine we attached to the segment, along with mac addresses for the interfaces on the segment, and other port specific data

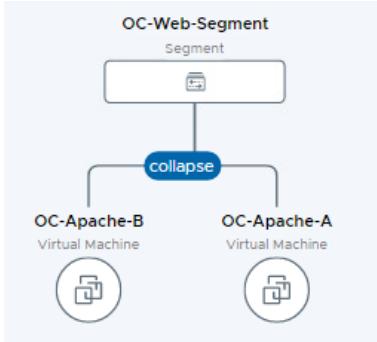
- J. Click on the Hosts tab
 K. Note that our OC-Web-Segment is connected on each ESXi host in the transport zone. When a segment is created, it is accessible to all hosts in the transport zone.

[Step 3.5] Discover the Network Topology

- Click on the NSX Manager tab in the browser
- If needed, Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
- From the NSX-T Manager interface click the **Networking** tab
- Select **Network Topology**
- Locate our new **OC-Web-Segment**. You may need to drag the screen to the left and zoom in to see the right side of the topology display. Look for a segment that has 2 VMs connected to it



- Click on **2 vms** under OC-Web-Segment
- Notice the 2 vms you configured on the OC-Web-Segment in the topology map



[Lab 1 Summary]

Lab 1 shows how simple it is to create an overlay network using NSX Manager. In this example we created a fully functional IP subnet on an overlay network segment in just a few steps. Unlike traditional VLANs in vSphere, segments do not require any underlying VLAN configuration.

Lab 2: View packet flow within a host

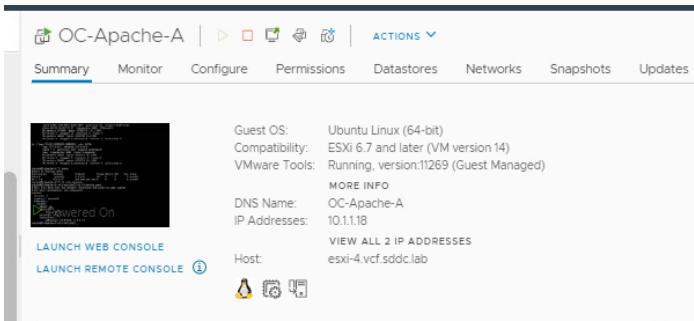
In this lab we will use the Traceflow capability in NSX to view traffic moving between virtual machines on the same host, on the same segment.

Traceflow injects packets into a vSphere distributed switch (VDS) port and provides observation points along the packet's path as it traverses the overlay and underlay network. Observation points include entry and exit of distributed firewalls, host and edge TEP, logical routers, etc. This allows you to identify the path (or paths) a packet takes to reach its destination or, conversely, where a packet is dropped along the way. Each entity reports the packet handling on input and output, so you can determine whether issues occur when receiving a packet or when forwarding the packet. Traceflow is not the same as a ping request/response that goes from guest-VM stack to guest-VM stack. With the NSX prepped VDS in VCF, NSX Traceflow can inject and monitor packets at the point where a VM vNIC connects to the VDS switch port. This means that a Traceflow can be successful even when the guest VM is powered down. Note: If the VM has not been powered on since attaching the NSX segment, the NSX control plane cannot know which host to use to inject packets from that VM as source and that test will fail.

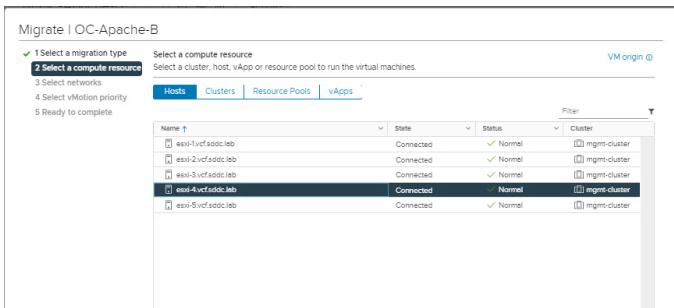
[Step 1] Setup VMs for test

- Click on the Management vCenter Server tab in the browser
- If session has timed out, login user: **administrator@vsphere.local** and password: **VMware123!**
- From the **Hosts and Clusters** view click on **OC-Apache-A** to determine on which ESXi is the VM running

In this example, **OC-Apache-A** is running on host esxi-4:



- D. Next click on OC-Apache-B to determine its host. If it is running on the same host as OC-Apache-A, skip forward a step
- E. If OC-Apache-B is not on the same host as OC-Apache-A, initiate a vMotion to move it to the same host



[Step 2] Test packet flow

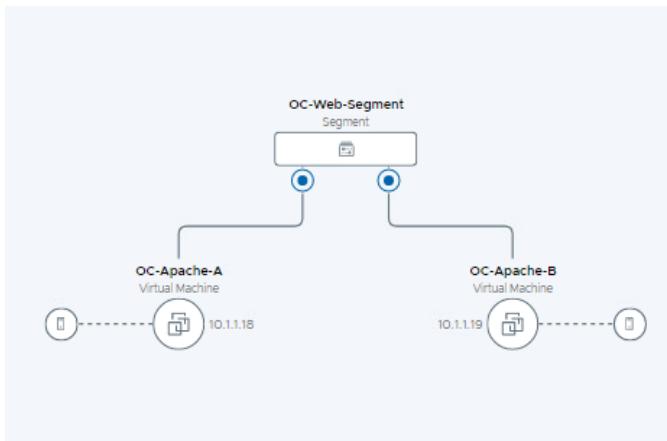
- A. Click on the **Mgmt Domain** tab in the browser, and select **Mgmt NSX**.
- B. If necessary, login as user: **admin** password: **VMware123!VMware123!**
- C. Click **Plan and Troubleshoot** on the top menu bar
- D. Click **Traceflow** on the left menu bar
- E. Setup a Traceflow between OC-Apache-A and OC-Apache-B. All values other than VM name can remain at defaults

The screenshot shows the NetworkMiner interface with the following configuration:

- Packet Information:**
 - IP Address: IPv4
 - Traffic Type: Unicast
 - Protocol Type: ICMP
 - ICMP ID: 0
 - Sequence: 0
- Source:**
 - Type: Virtual Machine
 - VM Name: OC-Apache-A
 - Host: a0eb7af-d90-4842-9ef6-dc99eba8545f
 - Virtual Interface: Network adapter 1
 - Segment Port: default-87eb2c2c-8e30-4e06-825e-c9264be76782
 - IP Address: 10.1.1.18
 - MAC Address: 00:50:56:82:24:17
- Destination:**
 - Type: Virtual Machine
 - VM Name: OC-Apache-B
 - Host: a0eb7af-d90-4842-9ef6-dc99eba8545f
 - Virtual Interface: Network adapter 1
 - Segment Port: default-4309472a-f31-4f98-a49f-e830759b8dc1
 - IP Address: 10.1.1.19
 - MAC Address: 00:50:56:82:a8:7d
- Buttons:**
 - ADVANCED SETTINGS
 - TRACE (highlighted in blue)

F. Click **Trace**

- G. Notice the path the data packets take on the resulting topology view. We can see that packets move from OC-Apache-A to OC-Apache-B via the OC-Web-Segment



H. In the Observations panel, review the following

- o We show 1 packet delivered
- o The physical hop count is 0, indicating that the packet did not leave the host
- o The packet was injected at the network adapter for OC-Apache-A virtual machine
- o It is then received at the distributed firewall at the VDS port for OC-Apache-A
- o With no rule blocking, the packet is then forwarded on from the sending VDS port
- o The packet is then received on the distributed firewall at the receiving VDS port for OC-Apache-B
- o With no rule blocking forwarding, the packet is then forwarded to the destination
- o The last step shows the packet being delivered to the network adapter for the OC-Apache-B VM

Observations	All	1 Delivered	0 Dropped	
Physical Hop Count	Observation Type	Transport Node	Component	Timestamp
0	Received	esxi-4.vcf.sddc.lab	Network adapter 1	06/01/18 743.95
0	Received	esxi-4.vcf.sddc.lab	Distributed Firewall	06/01/18 744.06
0	Forwarded	esxi-4.vcf.sddc.lab	Distributed Firewall (Rule ID: 2)	06/01/18 744.29
0	Received	esxi-4.vcf.sddc.lab	Distributed Firewall	06/01/18 744.23
0	Forwarded	esxi-4.vcf.sddc.lab	Distributed Firewall (Rule ID: 2)	06/01/18 744.25
0	Delivered	esxi-4.vcf.sddc.lab	OC-Apache-B.vmx#f52ba813-bd3c-4fd0-8577-2bd3712ec00	06/01/18 744.26

[Lab 2 Summary]

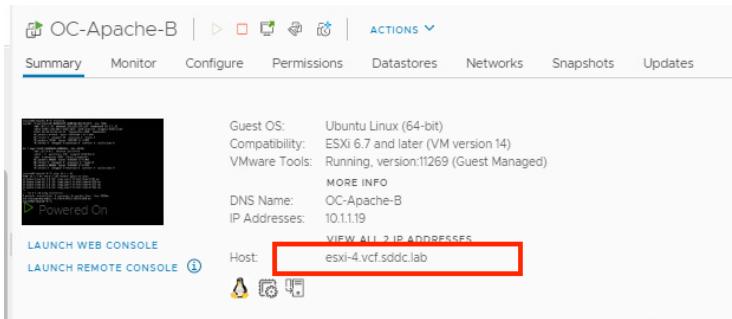
Lab 2 shows how ICMP packets travel between the VDS ports for 2 virtual machines running on the same ESXi host. You can see the packet pass from where it enters the VDS at the source, passes through the source side firewall, then get forwarded to the destination distributed firewall, and finally to the destination VDS port.

Lab 3: View packet flow between hosts

In this lab we will use the Traceflow capability in NSX to view traffic moving between virtual machines on different hosts on the same segment.

[Step 1] Setup VMs for test

- Click on the Management vCenter Server tab in the browser
- If session has timed out, login user: **administrator@vsphere.local** password: **VMware123!**
- From the hosts and clusters view, click on OC-Apache-A to determine which ESXi host currently has the VM running. In this example, OC-Apache-A is running on host esxi-4

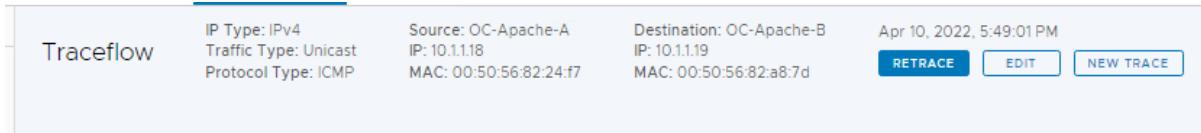


- Next click on OC-Apache-B to determine its host. If OC-Apache-B is on the same host as OC-Apache-A, initiate a vMotion to move OC-Apache-B to the different host. As we are using vSAN storage, you only need to vMotion compute

[Step 2] View packet flow

- Click on the NSX Manager tab in the browser
- If necessary, login as user: **admin** password: **VMware123!VMware123!**
- Click **Plan and Troubleshoot** on the top menu bar
- Click **Traceflow** on the left menu bar

- E. If your most recent Traceflow is still on screen between OC-Apache-A and OC-Apache-B, click **Retrace** and skip forward to step H



- F. If the previous Traceflow is not correct, Click **New Trace** and setup a Traceflow OC-Apache-A and OC-Apache-B. All values other than VMname can remain at defaults

The screenshot shows the "New Trace" dialog with the following settings:

Packet Information

- IP Address: IPv4
- Traffic Type: Unicast
- Protocol Type: ICMP
- ICMP ID: 0
- Sequence: 0

Source

- Type: Virtual Machine
- VM Name: OC-Apache-A
- Host: e0e6b74f-df90-4842-9ef6-dc99ebab545f
- Virtual Interface: Network adapter 1
- Segment Port: default 87eb2c2c-8e30-4e06-b25e-c9264be76782
- IP Address: 10.1.1.18
- MAC Address: 00:50:56:82:24:f7

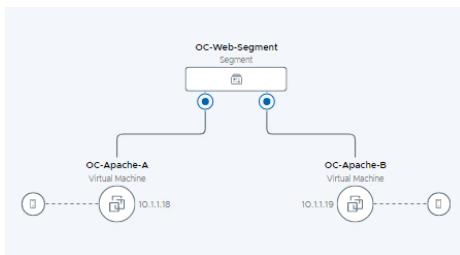
Destination

- Type: Virtual Machine
- VM Name: OC-Apache-B
- Host: e0e6b74f-df90-4842-9ef6-dc99ebab545f
- Virtual Interface: Network adapter 1
- Segment Port: default 4309472a-ff31-4f88-aef1-e818758bb8dc1
- IP Address: 10.1.1.19
- MAC Address: 00:50:56:82:a8:7d

Buttons: ADVANCED SETTINGS, TRACE

- G. Click **Trace**

- H. Notice the path the data packets take on the resulting topology view. We can see that packets move from app-01a to web-01a via the one-tier segment



- I. Ignore the “multiple physical received observations” banner if it shows up, as this is a nested lab environment

⚠ Traceflow round has multiple physical received observations. Please check whether the underlayer switch flood packets. Your hypervisor may be in nested environment.

- J. In the Observations panel, review the following
- o We show 1 packet delivered
 - o The physical hop count increments to 1 part way through the flow, indicating that the packet left the host
 - o The packet was injected at the network adapter for OC-Apache-A virtual machine
 - o It is then received at the distributed firewall at the VDS port for OC-Apache-A
 - o With no rule blocking, the packet is then forwarded on from the sending VDS port
 - o The packet then hits the physical layer to transmit to the second host. Notice the local and remote endpoints are shown
 - o The packet is then received on the second host. Notice the inverse local and remote endpoint IPs. The local and remote endpoints are the “Tunneling End Points” (TEP). When the OC-Apache-B virtual machine was migrated to another host, the NSX manager updated all hosts in the transport zone with the new TEP for the virtual machine
 - o The packet is then received on the distributed firewall at the receiving VDS port for OC-Apache-B.
 - o With no rule blocking forwarding, the packet is then forwarded to the destination
 - o The last step shows the packet being delivered to the network adapter for the OC-Apache-B VM

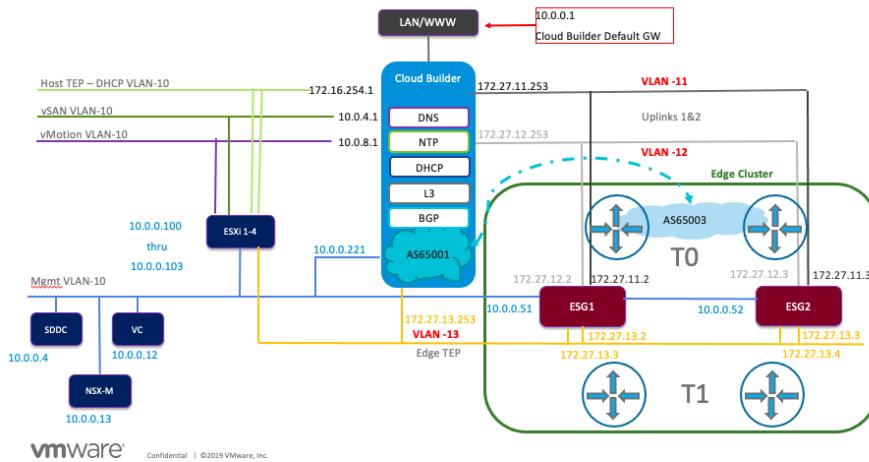
Observations	All	1 Delivered	0 Dropped			
Physical Hop Count	Observation Type	Transport Node	Component	Timestamp	IP Address	Actions
0	Injected	esxi-1.vcf.sddc.lab	Network adapter 1	00:03:13.422.862		
0	Received	esxi-1.vcf.sddc.lab	Distributed Firewall	00:03:13.422.920		
0	Forwarded	esxi-1.vcf.sddc.lab	Distributed Firewall (Rule ID: 2)	00:03:13.422.923		
0	Forwarded	esxi-1.vcf.sddc.lab	Physical	00:03:13.422.954	Local endpoint IP: 172.16.254.20 Remote endpoint IP: 172.16.254.19	
1	Received	esxi-2.vcf.sddc.lab	Physical	00:03:13.423.117	Local endpoint IP: 172.16.254.19 Remote endpoint IP: 172.16.254.20	
1	Received	esxi-2.vcf.sddc.lab	Distributed Firewall	00:03:13.423.168		
1	Forwarded	esxi-2.vcf.sddc.lab	Distributed Firewall (Rule ID: 2)	00:03:13.423.171		

[Step 3] View Host TEP information

- A. Click on the NSX Manager tab in the browser
- B. On the top menu bar click **System**
- C. In the left menu click **Fabric**
- D. Click **Nodes**
- E. On the Managed By field select **vcenter.mgmt.vcf.sddc.lab** from the pulldown
- F. Expand the **mgmt-cluster** if necessary
- G. Notice the TEP IP Addresses column. Each host has 2 TEP interfaces in the Host TEP VLAN. In the Holodeck lab configuration, Host TEP addresses are DHCP assigned on the 172.16.254.1/24 network with Cloud Builder acting as the DHCP server for ESXi hosts connecting to the Host TEP Network

Host Transport Nodes										
Node	ID	IP Addresses	Os Type	NSX Configuration	NSX Version	Host Switches	Tunnels	TEP IP Addresses	Node Status	Alarms
mgmt-cluster-01 (4)	McRef ID: dom...								4 Hosts Up	
esxi-1.vcf.sddc.lab	8750...Baed	10.0.0.101, 10.0.8.101, ...	ESXi 7.0.3	Success	3.13.5.0.19068...	1 ↑ 6	172.16.254.20, 172.16...	• Up ⓘ	0	
esxi-4.vcf.sddc.lab	dd9...c0f0	10.0.0.104, 10.0.8.104, ...	ESXi 7.0.3	Success	3.13.5.0.19068...	1 ↑ 7	172.16.254.14, 172.16.2...	• Up ⓘ	0	
esxi-3.vcf.sddc.lab	be7a...c862	10.0.0.103, 10.0.8.103, ...	ESXi 7.0.3	Success	3.13.5.0.19068...	1 ↑ 8	172.16.254.17, 172.16.2...	• Up ⓘ	0	
esxi-2.vcf.sddc.lab	2740...4163	10.0.0.102, 10.0.8.102, ...	ESXi 7.0.3	Success	3.13.5.0.19068...	1 ↑ 13	172.16.254.18, 172.16.2...	• Up ⓘ	0	
										172.16.254.19, 172.16.254.19

- H. Compare this to the logical layout of the Holodeck environment. Notice each host has two TEP interfaces on the DHCP based Host TEP Network



- I. The NSX Manager, interfaced with a vCenter Server instance, is responsible for updating all transport nodes in the transport zone any time a VM powers on or is migrated. This provides mapping of VM to TEP addresses to send overlay traffic for a specific VM. As a “Tunnel End Point” the NSX prepended vSphere Distributed switch is responsible to de-encapsulate overlay traffic to a VM and encapsulate traffic to communicate on the overlay. This is transparent to the VM and the underlay network.

[Lab 3 Summary]

In Lab 3 we show how ICMP packets travel between the VDS ports for 2 virtual machines running on different ESXi hosts. Like lab 2, you first see the packet pass from where it enters the VDS at the source, through the source side firewall. The packet then gets encapsulated and placed on the segment (overlay network) via the source side TEP and forwarded to the destination TEP. The destination TEP de-encapsulates the packet and passes to the destination side distributed firewall, and finally to the VDS port. This is a very simple example of the power of overlay networks in VCF. The source and destination physical machines do not have to be in the same subnet, as would be common in a multi rack configuration with Spine/Leaf physical networking.

Lab 4: Adding router connectivity

While we created two segments earlier in the lab, the segments are currently not connected together, or to other parts of the network. In this lab we will add an NSX Tier-1 router, connect OC-Web-Segment and OC-App-Segment to the T1 router, then connect to T1 router to the existing T0 router in the lab config

[Step 1] Create a T1 Router

- A. If necessary, open a new tab in the Chrome browser
- B. Click the Management NSX-T shortcut in the bookmark bar (click advanced / proceed to nsx-mgmt.vcf.sddc.lab, if required to accept the certificate)
- C. Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
- D. From the NSX-T Manager interface click the **Networking** tab
- E. Click **Tier-1 Gateways** in the left navigation panel
- F. Click on Add Tier-1 Gateway
- G. Name the gateway **OC-T1**, and select VLC-Tier-0 for the Linked Tier-0 gate

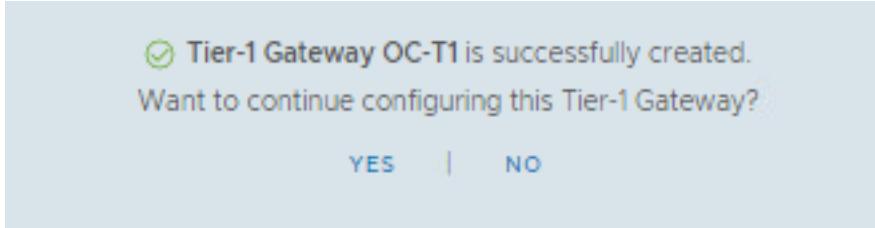
ADD TIER-1 GATEWAY		EXPAND ALL		Filter by Name, I
Tier-1 Gateway Name	Linked Tier-0 Gateway	#Linked Segments	Status	
OC-T1	VLC-Tier-0			
Edge Cluster	Select Edge Cluster	Edges	Set	
Edges Pool Allocation Size	Select Pool Allocation Size	Enable Standby Relocation	<input checked="" type="checkbox"/>	
Description	Description	Tags	<input checked="" type="checkbox"/> Tag <input checked="" type="checkbox"/> Scope	
Route Advertisement <small>NOTE - Before further configurations can be done, fill out mandatory fields above (*), click 'Save' below.</small>				
<input checked="" type="checkbox"/> SERVICE INTERFACES <input checked="" type="checkbox"/> STATIC ROUTES <input checked="" type="checkbox"/> MULTICAST				
<input type="button" value="SAVE"/> <input type="button" value="CANCEL"/>				

- H. Expand Route Advertisement
- I. Enable All Connected Segments & Service Ports

ADD TIER-1 GATEWAY		EXPAND ALL		Filter by Name, I
Tier-1 Gateway Name	Linked Tier-0 Gateway	#Linked Segments	Status	
OC-T1	VLC-Tier-0			
Edge Cluster	Select Edge Cluster	Edges	Set	
Edges Pool Allocation Size	Select Pool Allocation Size	Enable Standby Relocation	<input checked="" type="checkbox"/>	
Description	Description	Tags	<input checked="" type="checkbox"/> Tag <input checked="" type="checkbox"/> Scope	
Route Advertisement <small>NOTE - Before further configurations can be done, fill out mandatory fields above (*), click 'Save' below.</small>				
<input checked="" type="checkbox"/> All Static Routes <input checked="" type="checkbox"/> All DNS Forwarder Routes <input checked="" type="checkbox"/> All Connected Segments & Service Ports <input checked="" type="checkbox"/> All IPSec Local Endpoints				
<input checked="" type="checkbox"/> All NAT IP's <input checked="" type="checkbox"/> All LB VIP Routes <input checked="" type="checkbox"/> All LB SNAT IP Routes				
<input type="button" value="SAVE"/> <input type="button" value="CANCEL"/>				

- J. Click save

- K. Click no when asked if you want to continue configuring
- L. Status will be “in Progress” until you refresh it



[Step 2] Connect segments to OC-T1

- A. If necessary, open a new tab in the Chrome browser
- B. Click the Management NSX-T shortcut in the bookmark bar (click advanced / proceed to nsx-mgmt.vcf.sddc.lab, if required to accept the certificate)
- C. Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
- D. From the NSX-T Manager interface click the **Networking** tab
- E. Click **Segments** in the left navigation panel
- F. Click the three dots to the left of OC-DB-Segment then select Edit

Segments						
Segments	Segment Profiles	Edge Bridge Profiles	Metadata Proxies			
ADD SEGMENT						
Segment Name	Connected Gateway	Transport Zone	Subnets	Ports	Status	Alarms
> OC-DB-Segment	None	mgmt-wild-tz-overlay01 Overlay	10.11.33/27	1	Success	0
> OC-Web-Segment	None	mgmt-wild-tz-overlay01 Overlay	10.11.1/27	2	Success	0

- G. Under Connected Gateway, scroll to find OC-T1 and select

Segment Name	Connected Gateway	Transport Zone	Subnets	Ports	Status	Alarms
OC-DB-Segment	OC-T1 Tier	mgmt-domain-0-overlay01	10.11.33/27	1	Success	0

Segment needs to have either Subnets or VPN defined, or both.

Admin State: **ON**

L2 VPN: You have no L2 VPN sessions for this Gateway. For that, go to VPN Services. Note that for L2 sessions to work, you also need IPsec session defined.

VLAN: Enter List of VLANs

Domain Name: Enter Fully Qualified Domain Name

Edge Bridges: Set

Multicast Routing: Enabled

Address Bindings: Set

URPF Mode: Strict

Description: Description

Connectivity: **ON**

VPN Tunnel ID: **Set**

Uplink Teaming Policy: Select Uplink Teaming Policy

IP Address Pool: Select IP Pool

Metadata Proxy: Select Metadata Proxy

Replication Mode: Hierarchical Two-Tier replication

Tags: Tag Scope

SAVE CANCEL | Unsaved Changes

- H. Click Save -> Close Editing
- I. Click the three dots to the left of OC-Web-Segment then select Edit
- J. Under Connected Gateway, scroll to find OC-T1 and select

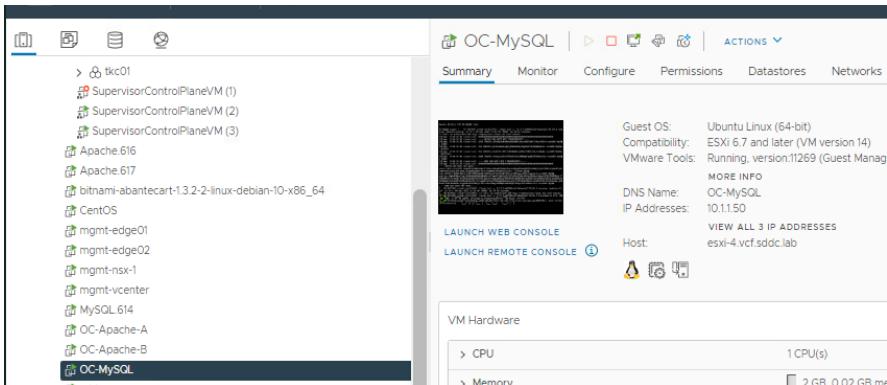


K. Click Save -> Close Editing

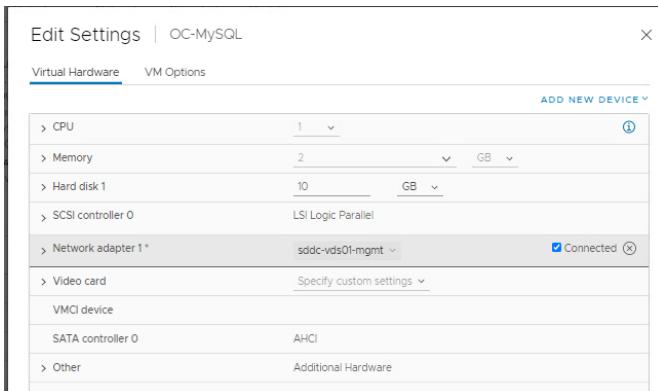
[Step 3] Attach OC-MySQL to segment OC-DB-Segment

From the vCenter Server Hosts and Clusters view, find **OC-MySQL** in the left-side scroll-list.

A. Right click on **OC-MySQL** and click **Edit Settings**

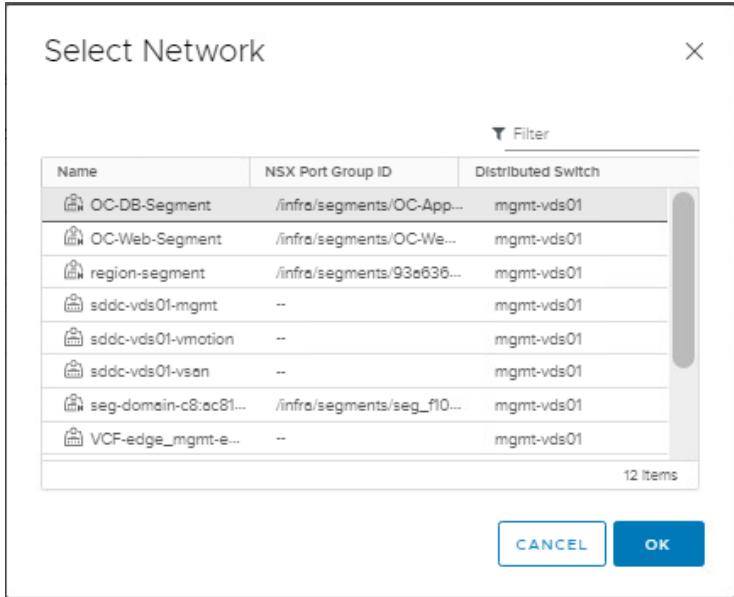


B. Click the dropdown next to **Network Adapter 1**



C. Click **Browse**

D. Click OC-DB-Segment

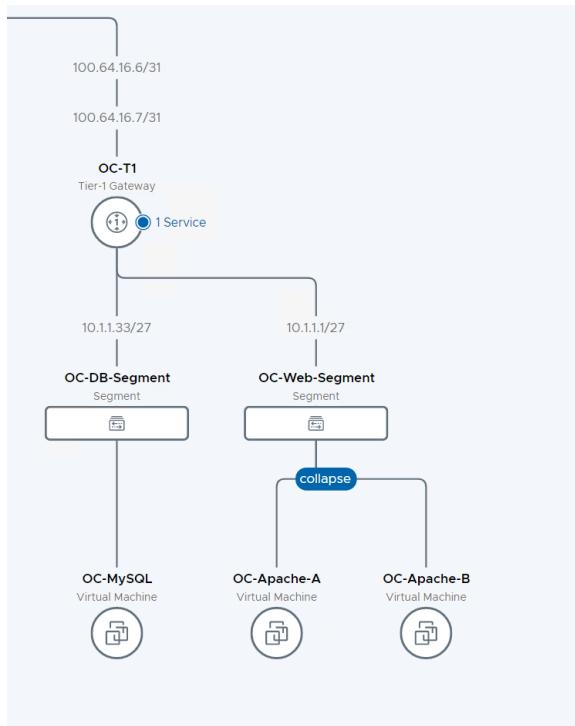


E. Click OK

F. Click OK

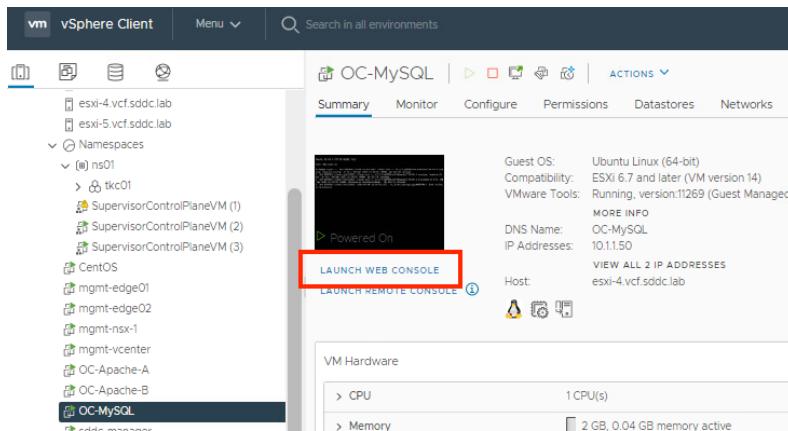
[Step 4] Review network topology

- If necessary, open a new tab in the Chrome browser
- Click the Management NSX-T shortcut in the bookmark bar (click advanced / proceed to nsx-mgmt.vcf.sddc.lab, if required to accept the certificate)
- Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
- From the NSX-T Manager interface click the **Networking** tab
- Click **Network Topology** in the left navigation panel
- Your newly created Tier-1 Router and connected segments should be on the right-hand side of the topology map



[Step 5] Test Web to App communications

- From the vCenter Server Hosts and Clusters view, find **OC-MySQL** in the left-side scroll-list.
- Click on **OC-MySQL** then click Open Web Console. You may need to click enter to bring up a login prompt



- Login **ocuser** password **VMware123!**
- Ping OC-Apache-A (10.1.1.18). Successful ping means that OC-MySQL can communicate with OC-Apache-A via the OC-T1 router

```
ocuser@OC-MySQL:~$ ping 10.1.1.18
PING 10.1.1.18 (10.1.1.18) 56(84) bytes of data.
64 bytes from 10.1.1.18: icmp_seq=1 ttl=63 time=2.10 ms
64 bytes from 10.1.1.18: icmp_seq=2 ttl=63 time=0.590 ms
64 bytes from 10.1.1.18: icmp_seq=3 ttl=63 time=0.601 ms
64 bytes from 10.1.1.18: icmp_seq=4 ttl=63 time=0.390 ms
64 bytes from 10.1.1.18: icmp_seq=5 ttl=63 time=0.366 ms
^C
--- 10.1.1.18 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 0.366/0.810/2.103/0.653 ms
ocuser@OC-MySQL:~$
```

[Step 5] View Layer 3 communications in NSX Traceflow

- If necessary, open a new tab in the Chrome browser
- Click the Management NSX-T shortcut in the bookmark bar (click advanced / proceed to nsx-mgmt.vcf.sddc.lab, if required to accept the certificate)
- Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
- From the NSX-T Manager interface click the **Plan & Troubleshoot** tab
- Click **Traceflow** in the left navigation panel
- Configure Traceflow from OC-Apache-A to OC-MySQL and click Trace

Packet Information [RESET](#)

IP Address IPv4	Traffic Type Unicast	Protocol Type ICMP
ICMP ID <input type="text" value="0"/>	Sequence <input type="text" value="0"/>	

Source [RESET](#)

Type Virtual Machine
VM Name OC-Apache-A
HostedOedb74f-0f90-4842-9ef6-dc89eb8d545f

Identify a virtual interface for packet injection

Virtual Interface Network adapter 1
Segment Port default:87eb2c2c-8e30-4e00-825e-c9264be70782
IP Address 10.1.1.18
MAC Address 00:50:56:82:04:57

Destination [RESET](#)

Type Virtual Machine
VM Name OC-MySQL
HostedOedb74f-0f90-4842-9ef6-dc89eb8d545f

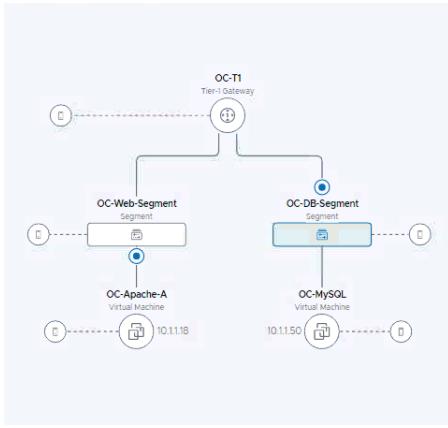
Virtual Interface [Network adapter 1](#)

Segment Port default:bdbe0240-2569-4ff4-8390-cf54859e3bed
IP Address 10.1.1.50
MAC Address 00:50:56:82:04:57

[ADVANCED SETTINGS](#)

TRACE

G. Your output should look like this. Notice the communications go via the OC-T1 router



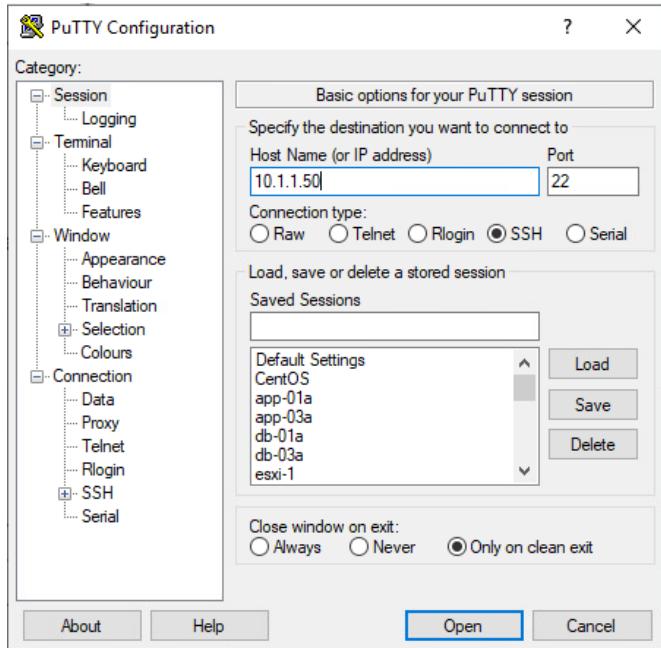
H. In the Observations panel, review the following

- We show 1 packet delivered
- The packet was injected at the network adapter for OC-Apache-A virtual machine
- It is then received at the distributed firewall at the VDS port for OC-Apache-A
- With no rule blocking, the packet is then forwarded on from the sending VDS port
- The packet then hits OC-T1 router and gets forwarded to the OC-Web-Segment
- Since OC-Apache-A and OC-MySQL are running on different ESXi hosts, you notice the physical hop between TEP's
- The packet is then received on the distributed firewall at the receiving VDS port for OC-MySQL
- With no rule blocking forwarding, the packet is then forwarded to the destination, the last step shows the packet being delivered to the network adapter for the OC-MySQL VM

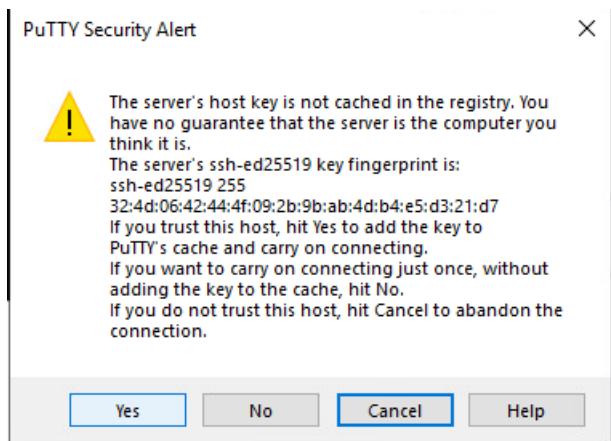
Observations						
Physical Hop Count	Observation Type	Transport Node	Component	Timestamp	IP Address	Actions
0	Injected	[esxi-1.vcf.sddc.lab]	Network adapter 1	00:32:17.154.204		
0	Received	[esxi-1.vcf.sddc.lab]	Distributed Firewall	00:32:17.154.252		
0	Forwarded	[esxi-1.vcf.sddc.lab]	Distributed Firewall (Rule ID: 2)	00:32:17.154.259		
0	Forwarded	[esxi-1.vcf.sddc.lab]	OC-Web-Segment	00:32:17.154.275		View Details
0	Received	[esxi-1.vcf.sddc.lab]	OC-T1	00:32:17.154.281		
0	Forwarded	[esxi-1.vcf.sddc.lab]	OC-T1	00:32:17.154.295		
0	Received	[esxi-1.vcf.sddc.lab]	OC-DB-Segment	00:32:17.154.303		View Details
0	Forwarded	[esxi-1.vcf.sddc.lab]	Physical	00:32:17.154.319	Local endpoint IP: 172.16.254.20 Remote endpoint IP: 172.16.254.14	
1	Received	[esxi-4.vcf.sddc.lab]	Physical	00:32:17.154.407	Local endpoint IP: 172.16.254.14 Remote endpoint IP: 172.16.254.20	
1	Received	[esxi-4.vcf.sddc.lab]	Distributed Firewall	00:32:17.154.478		
1	Forwarded	[esxi-4.vcf.sddc.lab]	Distributed Firewall (Rule ID: 2)	00:32:17.154.487		
1	Delivered	[esxi-4.vcf.sddc.lab]	OC-MySQL vmx@52cbd29-db3e-4f7c-9abd-5a3f7c3a761b	00:32:17.154.493		

[Step 6] Test end to end communications

- A. Click on the PuTTY icon
- B. Connect to 10.1.1.50 (OC-MySQL).



C. Click yes to accept ssh fingerprints if prompted



D. Login ocuser password VMware123!

```

Puuser@OC-MySQL: ~
* Support: https://ubuntu.com/advantage

System information as of Sun Apr 10 19:18:34 PDT 2022

System load: 0.0          Processes: 152
Usage of /: 30.7% of 8.80GB Users logged in: 0
Memory usage: 18%         IP address for ens160: 10.1.1.50
Swap usage: 0%

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

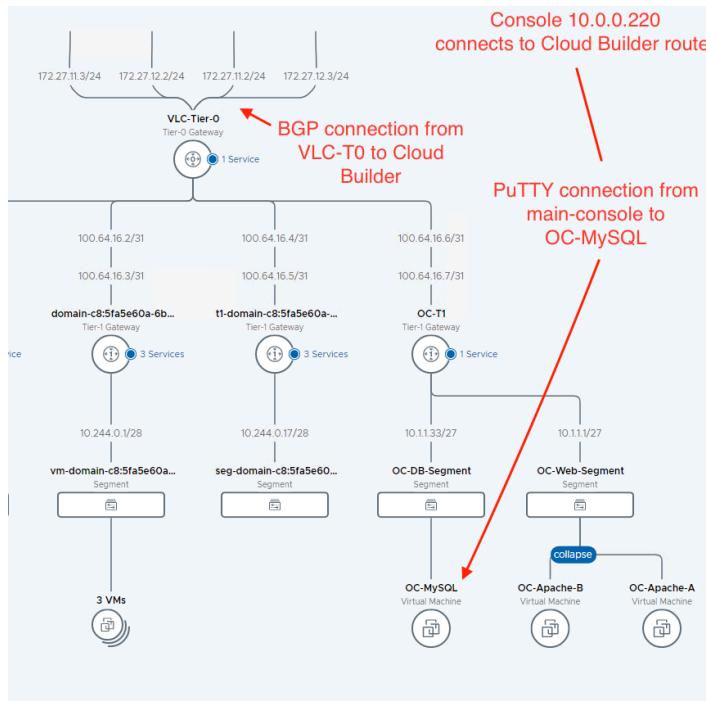
95 updates can be applied immediately.
67 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Apr  8 05:53:56 2022
Puuser@OC-MySQL:~$ 

```

- E. By successfully connecting PuTTY from the lab console to OC-MySQL we have tested the entire SDN connection. In this lab, the NSX Edge Cluster connects via BGP to the pod router where our lab console is connected. SSH traffic flows from our Windows console to the pod router, over BGP links to the Tier-0 router, to the OC-T1 router and finally to the OC-MySQL VM on OC-App-Segment, and back.



[Lab 5 Summary]

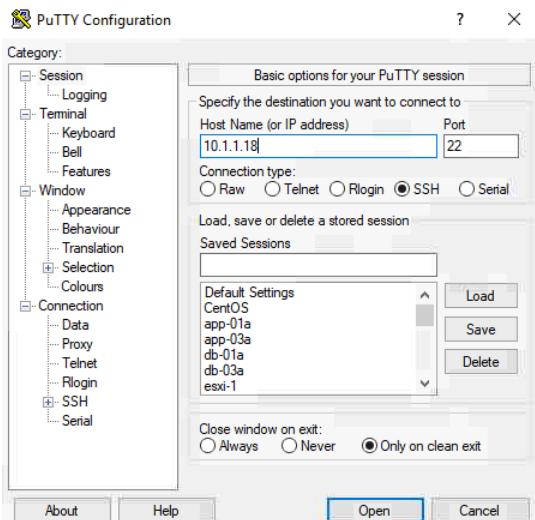
In Lab 5 we show packets travel between the VDS ports for 2 virtual machines running on different ESXi hosts across two segments connected by a Tier-1 router. The important distinction is this router functionality was distributed across all hosts versus a physical device cabled somewhere else in the datacenter.

Lab 5: Testing the application

This lab will validate our web server VM's and database VM are working correctly prior to moving on to load balancing and security

[Step 1] Restart web servers

- Click on the PuTTY icon
- Connect to 10.1.1.18 (OC-Apache-A). (Accept SSH warning if necessary)



- Login **ocuser** password **VMware123!**
- Run the command **sudo systemctl restart apache2**

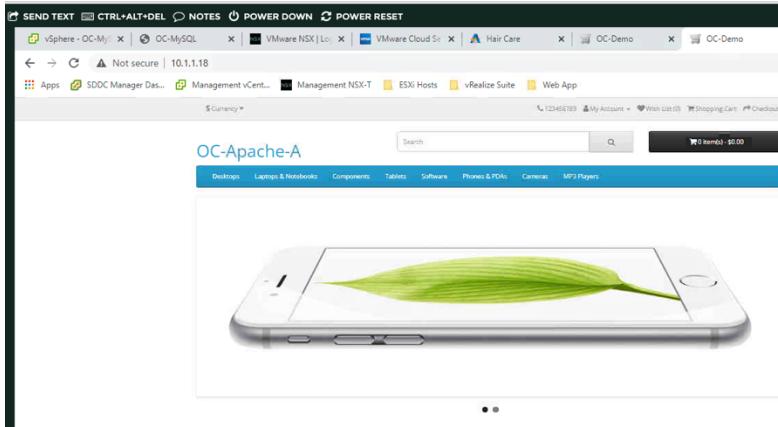
The screenshot shows a terminal window with the following command history:

```
ocuser@OC-Apache-A: ~
ocuser@OC-Apache-A:~$ sudo systemctl apache2 restart
Unknown operation apache2.
ocuser@OC-Apache-A:~$ sudo systemctl restart apache2
ocuser@OC-Apache-A:~$
```

- Repeat steps A-D for OC-Apache-B 10.1.1.19

[Step 2] Test Opencart app

- A. Open a new tab in the Chrome browser
- B. Connect to OC-Apache-A 10.1.1.18
- C. You should see the OC-Apache-A web page (The webservers for this lab module were modified to show the name of the host you are connecting to for clarity)



- D. Connect to the alternate port on OC-Apache-A 10.1.1.18:8080. You should see identical output on port 8080. This port will be used later in the security lab.



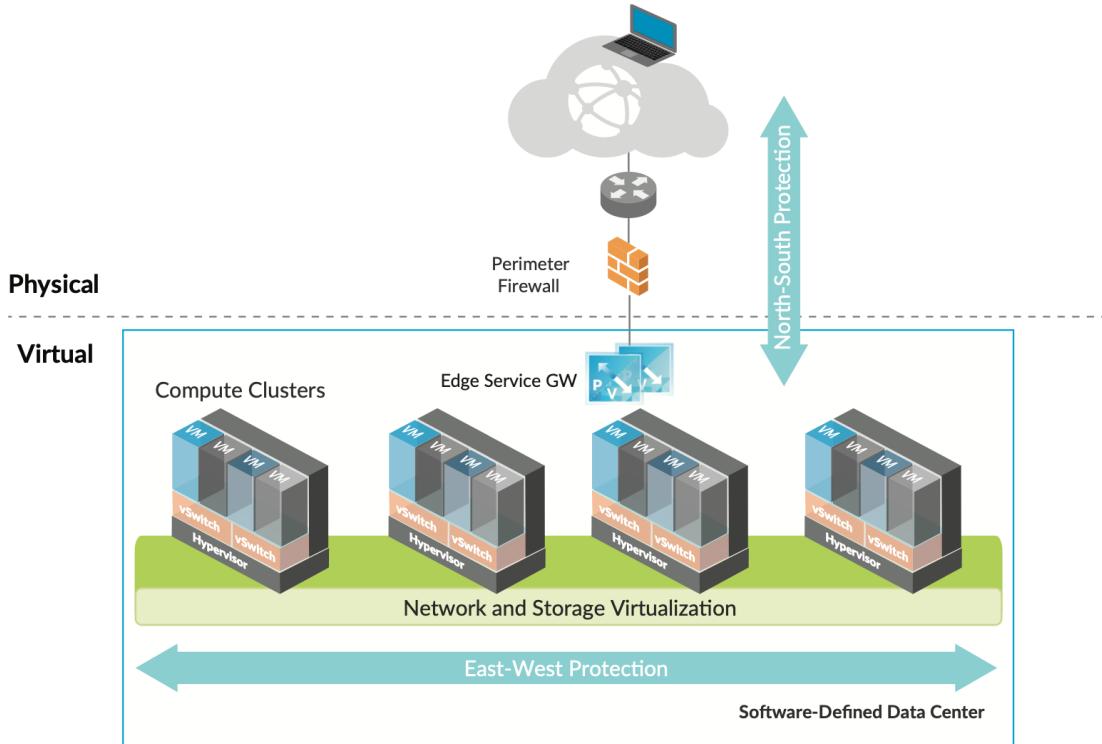
- E. Repeat steps A-D for OC-Apache-B 10.1.1.19



[Lab 5 Summary]

Congratulations. In just the time it took to get this far in the lab, you have deployed 2 brand new overlay network segments, a software defined router, and connected all so you can access applications from outside.

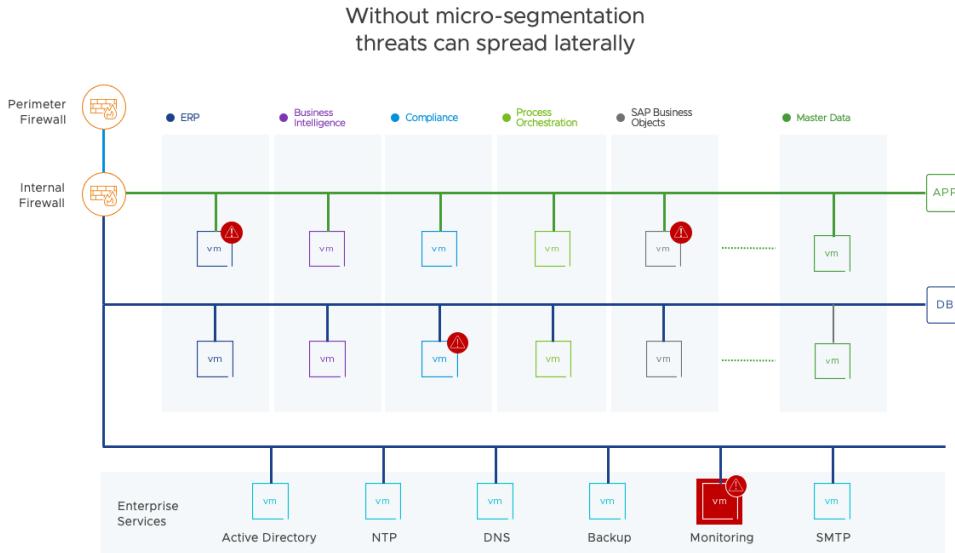
Module 2: Changing the Security Game with Microsegmentation



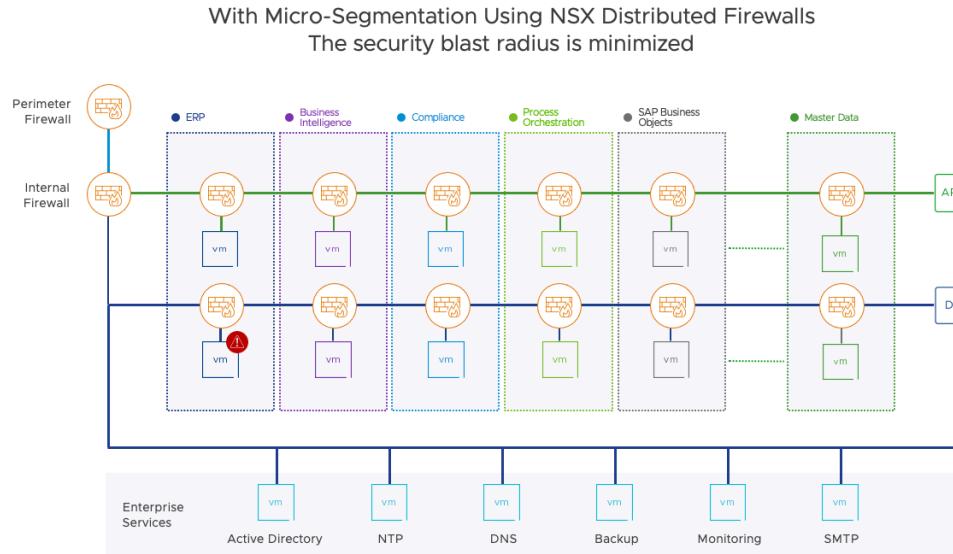
Over the past 10+ years traffic in a datacenter has changed. More and more traffic stays within the datacenter, moving between distributed application components. This traffic, known as "East-West", is

difficult to secure using traditional perimeter firewalls, which were predominantly designed for traditional “North-South” traffic.

Microsegmentation enables administrators to increase the agility and efficiency of the datacenter while maintaining an acceptable security posture. Microsegmentation decreases the level of risk and increases the security posture of the modern data center.



Microsegmentation with NSX in VCF is applied at the vNIC to VDS interface. Packets are inspected as they enter and leave each virtual machine. Microsegmentation is effectively a centralized packet filtering solution that acts on every machine.



Lab 1 Tagging VMs and Grouping Workloads based on Tags

This lab will explore the use of tagging to create groups of VM's to apply specific distributed firewall rules to. In small environments, creating groups based on VM name may suffice. However, as your environment grows, tagging may be a better alternate.

Terminology & definitions:

- **Tags** – A virtual machine is not directly managed by NSX however, NSX allows attachment of tags to a virtual machine. This tagging enables tag-based grouping of objects (e.g., you can apply a Tag called “AppServer” to all application servers).
- **Security Groups** – Security Groups enable you to assign security policy, such as distributed firewall rules, to a group of objects, such as virtual machines. In addition to Tags, you can also create groups based on VM attributes such as VM name, OS, IP, Ports, etc.
- **Security Policies** – Each firewall rule contains policies that act as instructions that determine whether a packet should be allow or blocked, which protocols it is allowed to use, which ports it is allowed to use, etc. Policies can be stateful or stateless.

Note: Tagging in NSX is distinct from tagging in vCenter Server, and at this time vCenter Server tags cannot be used to create grouping in NSX. In larger, more automated environments customers would use a solution such as vRealize Automation to deploy virtual machines and containers with security tagging set at time of creation.

Given that the demo Opencart app only has 2 web servers and 1 database server, we're going to create 2 tags as criteria for 2 groups. This might seem somewhat redundant, creating one tag per group, however it's essential to remember:

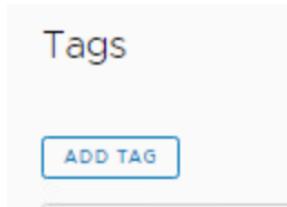
- This is a small sample 2-tier application. For applications leveraging micro-services, you'll be able to group more than one machine under one tag, and better leverage the security groups
- The advantage of using tags/groups is also an operational one. Once you create your infrastructure around Security Groups that contain tags, the moment you tag a machine with a specific tag, it immediately inherits the specific Security Group, Firewall rules and so on. This brings us closer to the cloud delivery mode.
- The downside is that a certain level of caution needs to be implemented when working with tags and Security Groups, meaning that it's just as easy to add a machine to an existing Security Group and avoid the complication that comes with setting up the firewall rules, but it is also just as easy to evade good security by giving the new machine too many permissions due to old tags/security group configurations.

To show the capability of tags we will set up **OC-Apache-A** with the appropriate Tags and Security Group. And then we'll have OC-Apache-B inherit the web tag and see how easy it is to apply all the appropriate rules to "a new machine". VM->Tag->Group mapping is as follows

VM	Tag	Security Group
OC-MySQL 10.1.1.50	OC-DB-Tag	OC-DB-Group
OC-Apache-A 10.1.1.18	OC-Web-Tag	OC-Web-group
OC-Apache-B 10.1.1.19	OC-Web-Tag	OC-Web-Group

[Step 1] Create Tags

- A. Open a new tab in the Chrome browser
- B. Click the Management NSX-T shortcut in the bookmark bar (click advanced / proceed to nsx-mgmt.vcf.sddc.lab, if required to accept the certificate)
- C. Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
- D. Navigate to **Inventory > Tags**
- E. Click on **ADD TAG**



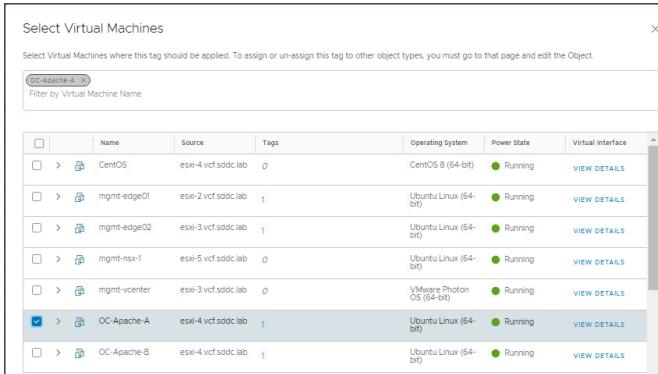
- F. We will first create a new tag for the web servers. Add the name **OC-Web-Tag** in the Tag field. Don't hit save yet. As you can see, the note says the tag must be assigned to at least one object first.

ADD TAG		
Tag	Scope	Assigned To
OC-Web-Tag *	Enter Scope	Set Virtual Machines *

NOTE - Tag must be assigned to at least one entry before it can be saved. You can assign this tag to Virtual Machines by clicking 'Set Virtual Machines'. To assign or un-assign this tag to other object types, you must go to that page and edit the object.

SAVE **CANCEL**

- G. Assign this tag to the OC-Apache-A virtual machine. Click on the **Set Virtual Machines** link field and select the **OC-Apache-A** virtual machine (you may need to scroll). For now, do not select OC-Apache-B as it will be added later



- H. Click **APPLY**
 I. Note the “Assigned To” value has incremented to 1
 J. Click **SAVE** to save the new tag

ADD TAG		
Tag	Scope	Assigned To
OC-Web-Tag *	Enter Scope	1

NOTE - Tag must be assigned to at least one entry before it can be saved. You can assign this tag to Virtual Machines by clicking 'Set Virtual Machines'. To assign or un-assign this tag to other object types, you must go to that page and edit the object.

SAVE **CANCEL**

- K. Click **Add Tag**
 L. Add the name “OC-DB-Tag”

ADD TAG		
Tag	Scope	Assigned To
OC-DB-Tag *	Enter Scope	Set Virtual Machines *

NOTE - Tag must be assigned to at least one entry before it can be saved. You can assign this tag to Virtual Machines by clicking 'Set Virtual Machines'. To assign or un-assign this tag to other object types, you must go to that page and edit the object.

SAVE **CANCEL**

- M. Click Set Virtual Machines then select OC-MySQL

Name	Source	Tags	Operating system	Power State	Virtual Interface
CentOS	esxi-4.vcf.sddc.lab	0	CentOS 8 (64-bit)	Running	VIEW DETAILS
mgmt-edge01	esxi-2.vcf.sddc.lab	1	Ubuntu Linux (64-bit)	Running	VIEW DETAILS
mgmt-edge02	esxi-3.vcf.sddc.lab	1	Ubuntu Linux (64-bit)	Running	VIEW DETAILS
mgmt-nsx-1	esxi-5.vcf.sddc.lab	0	Ubuntu Linux (64-bit)	Running	VIEW DETAILS
mgmt-vcenter	esxi-3.vcf.sddc.lab	0	VMware Photon OS (64-bit)	Running	VIEW DETAILS
OC-Apache-A	esxi-4.vcf.sddc.lab	2	Ubuntu Linux (64-bit)	Running	VIEW DETAILS
OC-Apache-B	esxi-4.vcf.sddc.lab	1	Ubuntu Linux (64-bit)	Running	VIEW DETAILS
OC-MySQL	esxi-4.vcf.sddc.lab	0	Ubuntu Linux (64-bit)	Running	VIEW DETAILS

N. Click **Apply**

O. Click **Save**

Step 1A: Verify Tags

A. Click on **Inventory > Tags > Filter by Name, Path and more**

B. Search for Tags with the string “OC” in the name

Tag	Scope	Assigned To	Last Assignment Status
OC-DB-Tag	1	1	Successful
OC-Web-Tag	1	1	Successful

Step 1B: Verify virtual machines are mapped to tags.

- A. Select **Inventory > Virtual Machines** and click in the **Filter** area
- B. Scroll down in Basic Detail and select **Tag**

Virtual Machines

Apply Filter

	Source	Tag
Instance ID	esxi-3.vcf.sddc.lab	1
Location ID	esxi-3.vcf.sddc.lab	0
Managed Object ID	esxi-3.vcf.sddc.lab	0
Name	esxi-3.vcf.sddc.lab	0
Operating System	esxi-2.vcf.sddc.lab	0
Power State	esxi-2.vcf.sddc.lab	0
Source	esxi-1.vcf.sddc.lab	1
Source ID	esxi-3.vcf.sddc.lab	0
Tag	esxi-2.vcf.sddc.lab	1
Tag Scope	esxi-1.vcf.sddc.lab	0
Compute Manager	esxi-2.vcf.sddc.lab	1
Virtual Interface	esxi-1.vcf.sddc.lab	0

- C. Select our 2 tags and click **Apply**

Virtual Machines

Basic Detail > Tag: OC-DB-Tag +1

Name	Tags
Edge-NSGroup	vcf.sddc.lab 1
Edge-NSGroup	vcf.sddc.lab 1
OC-Apache-B	vcf.sddc.lab 0
OC-MySQL	esxi-2.vcf.sddc.lab 0

Selected: OC-DB-Tag, OC-Web-Tag

Available Item: Edge_NSGroup

Buttons: CLEAR SELECTED, APPLY

- D. Verify OC-Apache-A and OC-MySQL are present

Virtual Machines

Basic Detail > Tag: OC-Web-Tag +1 Apply Filter

Name	Source	Tags	Operating System	Power State	Virtual Interface
OC Apache-A	esxi-4.vcf.sddc.lab	1	Ubuntu Linux (64-bit)	Running	VIEW DETAILS
OC MySQL	esxi-4.vcf.sddc.lab	1	Ubuntu Linux (64-bit)	Running	VIEW DETAILS

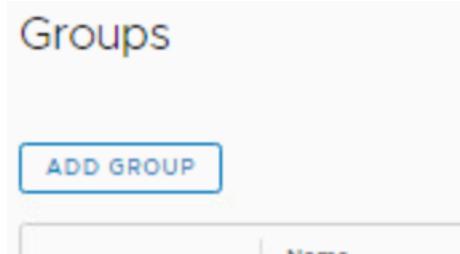
Step 2: Create Groups

Group mapping is as follows

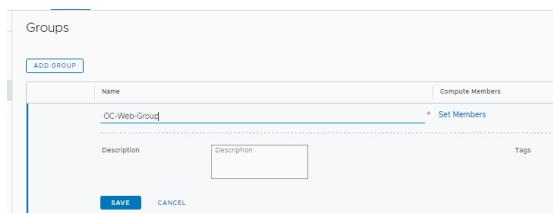
Tag	Security Group
OC-DB-Tag	OC-DB-Group
OC-Web-Tag	OC-Web-Group

Step 2.1: Create OC-Web-Group

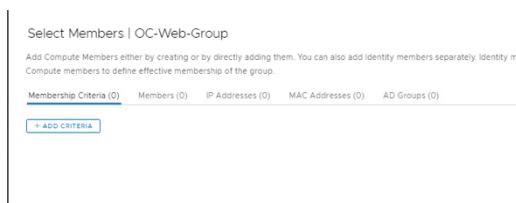
- A. Click Inventory > **Groups** > ADD GROUP



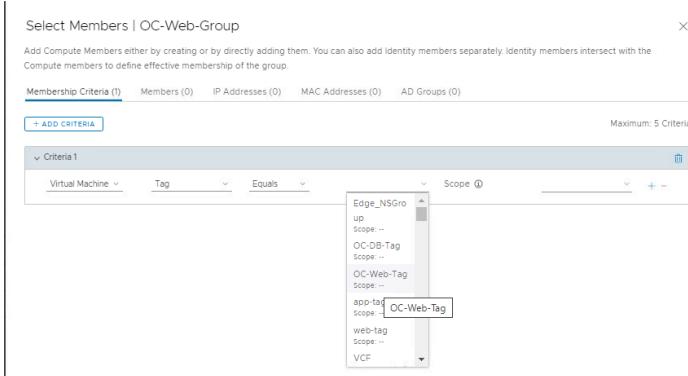
- B. Add group name **OC-Web-Group**



- C. Click on “**Select Members**” to add group members. In this example we will use the tags we just created to populate the group.



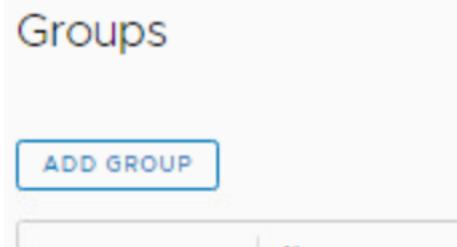
- D. Click **Add Criteria**
- E. Select **Virtual Machine**, that filters on **Tag Equals** “OC-Web-Tag”



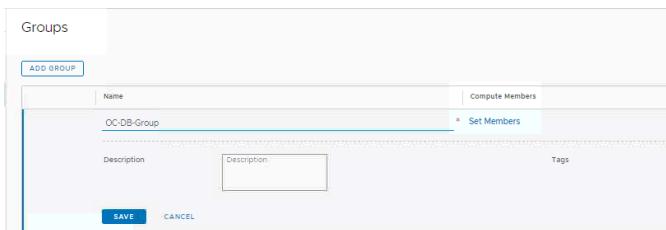
- F. Click **Apply**
- G. Click **Save**

Step 2.2: Create OC-DB-Group

- A. Click Inventory > **Groups** > **ADD GROUP**



- B. Add group name **OC-DB-Group**



- C. Click on **Select Members** to add group members. In this example we will use the tags we just created to populate the group.

Select Members | OC-DB-Group

Add Compute Members either by creating or by directly adding them. You can also add identity members separately. Identity members intersect with Compute members to define effective membership of the group.

Membership Criteria (0) Members (0) IP Addresses (0) MAC Addresses (0) AD Groups (0)

+ ADD CRITERIA

- D. Click **Add Criteria**
 E. Select **Virtual Machine**, that filters on **Tag Equals** OC-DB-Tag

Select Members | OC-DB-Group

Add Compute Members either by creating or by directly adding them. You can also add identity members separately. Identity members intersect with Compute members to define effective membership of the group.

Membership Criteria (0) Members (0) IP Addresses (0) MAC Addresses (0) AD Groups (0)

+ ADD CRITERIA

Criteria

Virtual Machine Tag Equals

OC-DB-Tag

- F. Click **Apply**
 G. Click **Save**

Step 2.1: Verify Groups

- A. On the **Inventory > Groups** panel click in the **Filter by Name, Path and More** field

Groups

ADD GROUP

EXPAND ALL Filter by Name, Path and more

Name	Compute Members	Where Used	Status

- B. Click on Name in the Basic Detail column

Basic Detail	Compute Members	Where Used	Status
Created By	View Members	Where Used	Success
Description	View Members	Where Used	Success
ID	View Members	Where Used	Success
Name	View Members	Where Used	Success
Path	View Members	Where Used	Success
Status	View Members	Where Used	Success
Tag	View Members	Where Used	Success
Tag Scope	View Members	Where Used	Success
Members	View Members	Where Used	Success
IP Address	View Members	Where Used	Success
Segment Ports	View Members	Where Used	Success
-	View Members	Where Used	Success

- C. Type OC to filter for our group names

- D. Select the **OC-Web-Group** and **OC-DB-Group** groups
- E. Click **Apply**
- F. Click **View Members** for each group
- G. Each group should have a single VM as a member (we can ignore IP addresses, Ports and VIF for now)

- H. At this point we have implemented the following:

VM	Tag	Group	Configured
OC-Apache-A 10.1.1.18	OC-Web-Tag	OC-Web-Group	Yes
OC-Apache-B 10.1.1.19			No
OC-MySQL 10.1.1.50	OC-DB-Tag	OC-DB-Group	Yes

[Lab 1 Summary]

Lab 1 shows how to create tagging and grouping in NSX. This capability allows creation and management of a scalable set of distributed firewall rules.

Lab 2: Applying Distributed Firewall Rules Based on Tagging on a segment

This lab will show configuring the distributed firewall to limit access in our Opencart Application. For the purposes of this lab, we will create the following rules.

Name	Source	Destination	Port/Protocol	Allowed	Notes
Inbound-web-80	Any	OC-Web-Group	HTTP (80)	Allow	Outside to web port 80
Inbound-web-8080	Any	OC-Web-Group	HTTP (8080)	Reject	Outside to web port 8080
Web-DB	OC-Web-Group	OC-DB-Group	3306 (MySQL)	Allow	Web to DB comms
ssh-admin	10.0.0.0/24	OC-DB-Group OC-Web-Group	SSH	Allow	SSH from lab console only
ICMP-Admin	10.0.0.0/24	OC-DB-Group OC-Web-Group	ICMP ALL	Allow	Allow ICMP only from lab console
ssh	Any	OC-DB-Group OC-Web-Group	SSH		SSH from any
Deny-All-Inbound	Any	OC-DB-Group OC-Web-Group	Any	Reject	Reject all else inbound

Keep in mind that this all happening at the distributed firewall level, where firewall rules are implemented at the VM switch port versus needing the services of a routed (perimeter) firewall to implement. Since we have created groups in the previous lab, now we can create access rules based on these groups.

Step 1: Create New Policy

- If necessary, open a new tab in the Chrome browser
- Click the Management NSX-T shortcut in the bookmark bar (click advanced / proceed to nsx-mgmt.vcf.sddc.lab, if required to accept the certificate
-)
- Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
- Navigate to **Security > Distributed Firewall** in the NSX-T Console
- Click on **Add Policy > New policy**, Name the policy “**Opencart**”



- Hover over **DFW**, then click the pencil.
- Note the default is the entire distributed firewall, however we want this rule to apply to the groups we created in the previous labs.
- Click the **Groups** radio button

J. Search for OC then select OC-DB-Group and OC-Web-Group

Name	Compute Members	Status
NLB PoolLB [OC-LB-Pool][OC-LB]	View Members	Success
NLB VIP [OC-VIP]	View Members	Success
OC-DB-Group	View Members	Success
OC-Web-Group	View Members	Success

K. Click **Apply**

Step 2: Add inbound-web-80 rule

- Click on the three vertical dots on the left of the **Opencart** policy.
- Click **Add Rule**

Name	ID	Applied To	Groups
Enable Logging For All Rules	(0)	Applied To	2 Groups
Disable Logging For All Rules	2.622	Applied To	1 Groups
Enable All Rules	1.624	Applied To	DFW
Disable All Rules	-8f17-44c4-b082...	Applied To	DFW
Delete Policy	(3)	Applied To	DFW

- Click on **Add Rule** and change name to **inbound-web-80**
- Hover over **Destinations**, then click the pencil icon
- This brings up the Set Destination pop-up. Type OC in the search box then click on **OC-Web-Group** checkbox, then **apply**

Set Destination

Rule > Inbound-Web-80

Negate Selections No Negated selections will be shown as Example-Group

Groups (1) IP Addresses (0)

(OC-Web-Group X) oc

ADD GROUP

	Name	Compute Members	Status
<input type="checkbox"/>	OC-DB-Group	View Members	Success
<input checked="" type="checkbox"/>	OC-Web-Group	View Members	Success

- F. Hover over **Services**, then click the pencil icon
 G. Type HTTP in the services area then select HTTP

Set Services

Rule > Inbound-Web-80

Services (1) Raw Port-Protocols (0)

HTTP X http

ADD SERVICE

	Name	Service Entries	Status
<input type="checkbox"/>	CIM-HTTP	TCP (Source: Any Destination: 5988)	Success
<input type="checkbox"/>	CIM-HTTPS	TCP (Source: Any Destination: 5989)	Success
<input checked="" type="checkbox"/>	HTTP	TCP (Source: Any Destination: 80)	Success
<input type="checkbox"/>	HTTPS	TCP (Source: Any Destination: 443)	Success
<input type="checkbox"/>	HTTPS, net.tcp binding	TCP (Source: Any Destination: 32843,32844,32845)	Success

- H. Click Apply
 I. The result should be the following:

Distributed Firewall

All Rules **Category Specific Rules**

ETHERNET (0) **EMERGENCY (0)** **INFRASTRUCTURE (0)** **ENVIRONMENT (0)** **APPLICATION (8)**

Unpublished Changes **PUBLISH**

Opencart (1) Applied To 2 Groups

Inbound-Web-80 Any DC-Web-Group HTTP None DFW Allow

Step 2.1: Add inbound-web-8080 rule (clone inbound-web-80)

- Click on the three vertical dots on the left of the **inbound-web-80** rule.
- Click **Clone Rule**

The screenshot shows the 'Rules' section of the VCF Policy interface. A context menu is open over the 'inbound-web-80' rule, with 'Clone Rule' highlighted. Other options visible in the menu include 'Add Rule', 'Delete Rule', 'Copy Rule', and 'Paste Rule'. The main table lists three rules: 'Openccart' (1), 'inbound-web-80' (2), and 'fault Layer3 Section' (3). The 'inbound-web-80' rule is selected and has its details shown in the right panel: 'Applied To' is 'DFW', 'Services' is 'HTTP', 'Action' is 'Allow', and there are two audit logs: 'Success' and 'Success'.

- Click on **Copy of inbound-web-80** and change name to **inbound-web-8080**
- Hover over **Services**, then click the pencil icon
- Click the X on HTTP to delete the service

The screenshot shows the 'Set Services' configuration for the 'inbound-web-8080' rule. Under the 'Services (1)' tab, the 'HTTP' service is listed with a delete icon. Below the list is an 'ADD SERVICE' button. The 'Raw Port-Protocols (0)' tab is also present.

- Click on **Raw Port Protocols**
- Click **Add Service Entry**
- Set Service Type **TCP** and Destinations Ports **8080**

The screenshot shows the 'Raw Port-Protocols' configuration for the 'inbound-web-8080' rule. Under the 'Raw Port-Protocols (1)' tab, an 'ADD SERVICE ENTRY' button is visible. The configuration table shows a single entry: 'Service Type' is 'TCP', 'Source Ports' is empty, and 'Destination Ports' is set to '8080'.

- I. Click Apply
- J. The result should be the following:

The screenshot shows the 'Distributed Firewall' interface under the 'Category Specific Rules' tab. It displays two rules applied to the 'OpenCart' profile:

- Inbound-Web-8080:** Any source, TCP Src Any Dest 1, DC-Web-Group, None, DPW, Allow.
- Inbound-Web-80:** Any source, DC-Web-Group, HTTP, None, DPW, Allow.

Step 3: Test inbound-web-XX rules

- A. Notice at this point we have two rules in place that are defaulted to Allow, and we have not yet published the rule changes. Leave this as is for the moment. Next, we will test that both ports are currently active on our web server

This screenshot is identical to the one above, showing the 'Distributed Firewall' interface with the same two inbound web rules applied to the 'OpenCart' profile.

- B. Open a tab on the browser and connect to OC-Apache-A 10.1.1.18

The screenshot shows a web browser window for 'OC-Apache-A' at the URL '10.1.1.18'. The page displays a search bar and a navigation menu with categories like Desktops, Laptops & Notebooks, Components, Tablets, Software, Phones & PDAs, Cameras, and MP3 Players. Below the menu, there is a large image of two laptops.

- C. Open a second tab and connect to OC-Apache-A 10.1.1.18:8080



- D. Return to the NSX Manager tab. Click the arrow next to **Allow** on inbound-web-8080 and select **Reject**

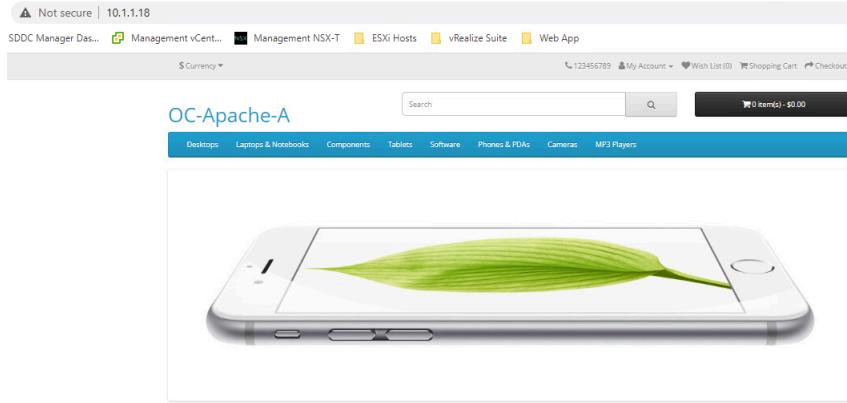
Name	ID	Sources	Destinations	Services	Context Profiles	Applied To	Action
Opencart	(2)	Applied To: 2 Groups					
inbound-web-8080		Any	OC-Web-Group	TCP Src: Any, Dest: 1 p...	None	DFW	
inbound-web-80		Any	OC-Web-Group	HTTP	None	DFW	

- E. Click **PUBLISH** to make the rules active

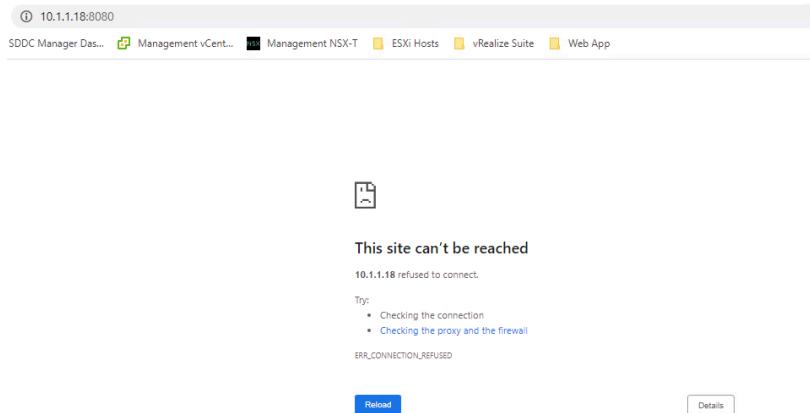
- F. Notice that the Publish button is greyed out, showing there are no uncommitted changes. The green Success indicator is set at the policy level, and our two rules now have ID numbers showing they have been activated.

Name	ID	Sources	Destinations	Services	Context Profiles	Applied To	Action
Opencart	(2)	Applied To: 2 Groups					
inbound-web-8080	3073	Any	OC-Web-Group	TCP Src: Any, Dest: 1 p...	None	DFW	
inbound-web-80	3074	Any	OC-Web-Group	HTTP	None	DFW	

- G. Go to the OC-Apache-A 10.1.1.18 browser tab and refresh. The web page should load normally



- H. Go to the OC-Apache-A 10.1.1.18:8080 browser tab and refresh. The web page should fail to load

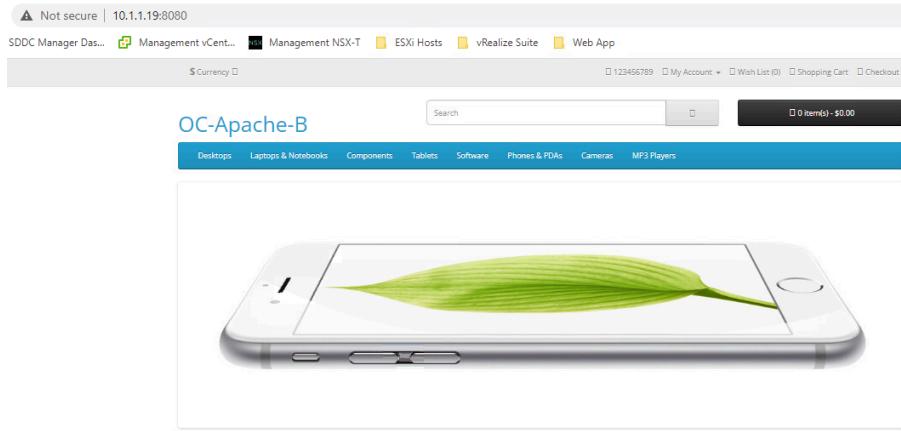


Step 4: Extend security policies to new VM based on tag

- A. Test operations of the OC-Apache-B 10.1.1.19 webserver on port 80



B. Test operations of the OC-Apache-B 10.1.1.19 webserver on port 8080



- C. Observe that the Reject rule on port 8080 does not extend to OC-Apache-B.
- D. Return to the NSX tab and click on **Inventory** then **Tags**
- E. Click on Filter, select Tag and search for by Name and type OC-Web

Tag	Scope	Assigned To
OC-Web-Tag		1

- F. Click the 3 dots next to OC-Web-Tag, then click Edit
- G. Click on the number 1 under Assigned To
- H. Search for OC-Apache and select OC-Apache-B then click Apply

	Name	Source	Tags	Operating System	Power State	Virtual Interface
<input type="checkbox"/>	mgmt-edge02	esxi-3.vcf.sddc.lab	1	Ubuntu Linux (64-bit)	Running	VIEW DETAILS
<input type="checkbox"/>	mgmt-nsx-1	esxi-5.vcf.sddc.lab	0	Ubuntu Linux (64-bit)	Running	VIEW DETAILS
<input type="checkbox"/>	mgmt-vcenter	esxi-3.vcf.sddc.lab	0	VMware Photon OS (64-bit)	Running	VIEW DETAILS
<input checked="" type="checkbox"/>	OC-Apache-A	esxi-4.vcf.sddc.lab	1	Ubuntu Linux (64-bit)	Running	VIEW DETAILS
<input checked="" type="checkbox"/>	OC-Apache-B	esxi-4.vcf.sddc.lab	0	Ubuntu Linux (64-bit)	Running	VIEW DETAILS

I. Click Save

Tag	Scope	Assigned To
OC-Web-Tag	Enter Scope	2

J. Test operations of the OC-Apache-B 10.1.1.19 webserver on port 8080

This site can't be reached
10.1.1.19 refused to connect.
Try:
• Checking the connection
• Checking the proxy and the firewall
ERR_CONNECTION_REFUSED

[Reload](#) [Details](#)

K. Notice that as soon as the tag was applied to OC-Apache-B VM, it immediately became a part of the Opencart Policy, because it became a member of the web-group that the rules applied to

Step 5: Implement Web-DB rule

The step will Allow communications from the Apache web servers to MySQL

- On the NSX Manager tab, go to Security -> Distributed Firewall
- Click on the 3 dots next to Opencart and Add Rule

- Name the rule **Web-DB**.

- Set Sources to OC-Web-Group and click Apply

The screenshot shows the 'Set Source' dialog for a rule named 'Web-DB'. Under 'Groups (1)', 'OC-Web-Group' is selected. In the main pane, a table lists groups with their names, compute members, and status. 'OC-DB-Group' and 'OC-Web-Group' both have a status of 'Success'.

Name	Compute Members	Status
OC-DB-Group	View Members	Success
OC-Web-Group	View Members	Success

E. Set Destinations to OC-DB-Group and click Apply

The screenshot shows the 'Set Destination' dialog for the same rule. Under 'Groups (1)', 'OC-DB-Group' is selected. The table in the main pane shows 'OC-DB-Group' with a status of 'Success'.

Name	Compute Members	Status
OC-DB-Group	View Members	Success
OC-Web-Group	View Members	Success

F. Set Services to MySQL and click Apply

The screenshot shows the 'Set Services' dialog. Under 'Services (1)', 'MySQL' is selected. The table in the main pane shows 'MySQL' with a service entry 'TCP (Source: Any | Destination: 3306)' and a status of 'Success'.

Name	Service Entries	Status
MySQL	TCP (Source: Any Destination: 3306)	Success

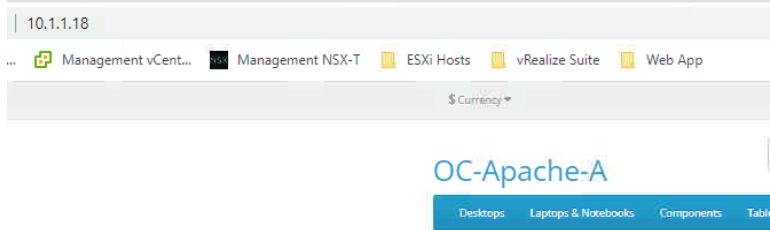
G. Your rule should look like this

The screenshot shows the 'Distributed Firewall' rules list. It displays a single rule named 'OpenCart' with the following details:

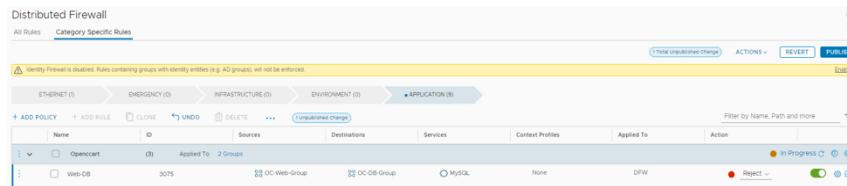
- Applied To:** 2 DFWs
- Sources:** Web-DB
- Destinations:** OC-DB-Group
- Services:** MySQL
- Action:** Allow
- Status:** Success

H. Click Publish

I. Test access to OC-Apache-A on 10.1.1.18. You should get a normal web page load



J. Return to the NSX tab and set the web-app rule to Reject and click Publish (this will allow us to see the impact of the firewall blocking access from Apache to MySQL which will be useful later in the lab)



K. Test access to OC-Apache-A on 10.1.1.18. Your web page load should fail



L. Reset the web-app rule to Allow and Publish before moving on to the next step



Step 5: Implement Deny All Inbound rule

The step will implement a Deny All Inbound rule which will deny all inbound traffic we have not explicitly allowed. This step will also show the order of rule evaluation within a security policy

- On the NSX Manager tab, go to Security -> Distributed Firewall
- Click on the 3 dots next to Opencart and Add New Rule

The screenshot shows the 'Distributed Firewall' section of the VCF interface. On the left, there's a navigation sidebar with categories like 'East West Security', 'North South Security', 'Network Traffic Analysis', 'Threat Detections', 'Endpoint Protection', and 'Settings'. The main area displays a table of rules under the heading 'All Rules'. One rule, 'Deny-All-Inbound', is highlighted with a red box. The table columns include 'Name', 'ID', and 'Action'. The 'Action' column shows 'Allow' for most rules and 'Reject' for the 'Deny-All-Inbound' rule.

Name	ID	Action
Enable Logging For All Rules	3059	Allow
Disable Logging For All Rules	3060	Allow
Enable All Rules	2.6... (3)	Allow
Disable All Rules		Reject
Delete Policy	3057	Allow
Add Rule	3053	Allow
Add Policy Above	3058	Allow
Add Policy Below		Allow
Copy Path to Clipboard		Allow

C. Name the rule Deny-All-Inbound.

This screenshot shows the detailed configuration for the 'Deny-All-Inbound' rule. It lists four specific rules under the 'Opencart' group. The 'Weo-DB' rule has 'MySQL' as its destination group. The 'inbound-web-B0B0' and 'inbound-web-B0' rules both have 'HTTP' as their destination group. The 'Weo-DB' rule has 'Allow' action, while the other three have 'Reject' action.

Rule	Source	Destination	Action
Weo-DB	Any	MySQL	Allow
inbound-web-B0B0	Any	TOP Src Any; Dest 1 ports	Reject
inbound-web-B0	Any	HTTP	Reject

D. Leave Sources at Any

E. Set Destinations to groups OC-DB-Group and OC-Web-Group (Hint: Type OC- and enter to quickly show OC groups) Check the two groups and then Apply

This screenshot shows the 'Set Destination' dialog for the 'Deny-All-Inbound' rule. It lists 'Groups (2)' and 'IP Addresses (0)'. Under 'Groups (2)', 'OC-DB-Group' and 'OC-Web-Group' are selected. The 'ADD GROUP' button is visible. Below the list, a table shows the selected groups with their names, compute members, and status. Both groups have a status of 'Success'.

Name	Compute Members	Status
NLB PoolLB [OC-LB-Pool][OC-LB]	View Members	Success
NLB VIP [OC-VIP]	View Members	Success
OC-DB-Group	View Members	Success
OC-Web-Group	View Members	Success

F. Set Action to Reject, then click Publish

The screenshot shows the NSX Firewall Policy interface. A warning message at the top states: "Identity Firewall is disabled. Rules containing groups with identity entities (e.g. AD groups), will not be enforced." The policy path is highlighted as "APPLICATION (1)". A table lists rules under the "OpenCart" section. One rule, "Deny-All-Inbound", is selected and has a status of "Success".

Name	ID	Sources	Destinations	Services	Context Profiles	Applied To	Action
Deny-All-Inbound	(4)	Any	OC-DB-Group OC-Web-Group	Any	None	DPW	Reject

- G. Test access to OC-Apache-A on 10.1.1.18. Your web page load should fail. This is because the Deny-All rule is evaluated prior to our Allow rules

The screenshot shows a browser window with the URL "10.1.1.18". The error message is "This site can't be reached" and includes the sub-message "10.1.1.18 refused to connect". It also lists troubleshooting steps: "Try:" followed by "Checking the connection" and "Checking the proxy and the firewall".

H. Return to the NSX tab

- I. Move the deny-all rule down by clicking the mouse and holding down with the cursor anywhere on the Deny-All rule line and dragging the rule to below our inbound-web-80 rule then clicking Publish

The first screenshot shows the "OpenCart" section with the "Deny-All-Inbound" rule at the top. The second screenshot shows the same section after the rule has been moved to the bottom, positioned between the "Weo-DB" and "inbound-web-80" rules.

Name	ID	Sources	Destinations	Services	Context Profiles	Applied To	Action
Deny-All-Inbound	(4)	Any	OC-DB-Group OC-Web-Group	Any	None	DPW	Reject
Weo-DB	1393	OC-Web-Group	OC-DB-Group	MySQL	None	DPW	Allow
inbound-web-80	1391	Any	OC-Web-Group	TCP Src Any; Dest 1 ports	None	DPW	Reject
inbound-web-80	1392	Any	OC-Web-Group	HTTP	None	DPW	Allow
Deny-All-Inbound	(4)	Any	OC-DB-Group OC-Web-Group	Any	None	DPW	Reject

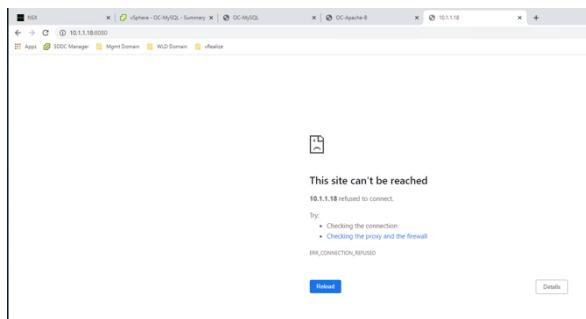
- J. Test access to OC-Apache-A on 10.1.1.18. You should get a normal web page load

The screenshot shows a browser window with the URL "10.1.1.18". The title bar says "NSX" and "vSphere - OC-MySQL - Summary". The main content area displays the Apache welcome page with the text "OC-Apache-A".

- K. Return to the NSX tab
- L. Click on the tree dots to the left of the inbound-web-8080 line and Delete Rule (since we have a hard Deny-All, unless port 8080 is explicitly allowed, it will be blocked, rendering this rule no longer needed.)
- M. Publish the rules. You may need to click Refresh on the bottom of the policy screen

Name	ID	Sources	Destinations	Services	Context Profiles	Applied To	Action
Opencart	(3)	Applied To 2 Groups					Success
Web-DB	3075	OC-Web-Group	OC-DB-Group	MySQL	None	DFW	Allow
inbound-web-80	3074	Any	OC-Web-Group	HTTP	None	DFW	Allow
Deny-All	3076	Any	Any	Any	None	DFW	Reject

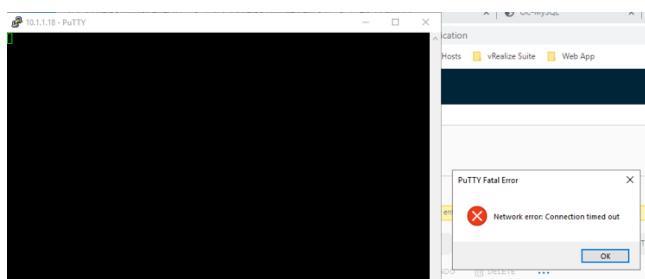
- N. Go to the OC-Apache-A 10.1.1.18:8080 The web page should fail to load.



Step 5: Implement ssh rule

The step will implement a rule to allow ssh connections to our Apache and MySQL VM's, but only from our inside admin network (10.0.0.0/24)

- A. Attempt to SSH to OC-Apache-A. Click on PuTTY and connect to 10.1.1.18
- B. Your connection should time out



- C. Return to the NSX tab
- D. Click on the tree dots to the left of the Opencart and Add Rule
- E. Name the rule ssh-admin

Name	ID	Sources	Destinations	Services	Context Profiles	Applied To	Action
Opencart	(4)	Applied To 2 Groups					In Progress Revert Clone Details
ssh-admin		Any	Any	Any	None	DFW	Allow Block Details

- F. Click on the pencil for Sources. Then IP Addresses. Set to 10.0.0.0/24, then click Apply

Set Source

Rule > ssh-admin

Negate Selections No Negated selections will be shown as Example-Group

Groups (0) IP Addresses (0)

10.0.0.0/24

CIDR e.g. IPv4 100.64.0.0/16 or IPv6 fc7e:f206:db42::/48, Range e.g. IPv4 100.64.0.0-100.64.0.32 or IPv6 fc7e::fc7e::32

- G. Click on the pencil for Services. Select SSH, then click Apply

Set Services

Rule > ssh-admin

Services (1) Raw Port-Protocols (0)

SSH	X
ssh	

ADD SERVICE EXPAND ALL Filter by Name, Path and more

	Name	Service Entries	Status
SSH		TCP (Source: Any Destination: 22)	Success

- H. Publish your new rule

Distributed Firewall							
All Rules		Category Specific Rules					
Identity Firewall is disabled. Rules containing groups with identity entities (e.g. AD groups), will not be enforced.							
+ ADD POLICY	+ ADD RULE	<input type="checkbox"/> CLONE	UNDO	DELETE	...	Unpublished Change	ACTIONS REVERT PUBLISH
ETHERNET (0)	EMERGENCY (0)	INFRASTRUCTURE (0)	ENVIRONMENT (0)	APPLICATION (5)			
Name	ID	Sources	Destinations	Services	Context Profiles	Applied To	Action
Opencart	(4)	Applied To 2 Groups					In Progress Revert Clone Details
ssh-admin		10.0.0.0/24	Any	<input checked="" type="radio"/> SSH	None	DFW	Allow Block Details
web-app	3061	<input checked="" type="checkbox"/> web_group	<input checked="" type="checkbox"/> app_group	<input checked="" type="radio"/> MySQL	None	DFW	Allow Block Details

- I. Attempt to SSH to OC-Apache-A. Click on PuTTY and connect to 10.1.1.18
- J. Your connection should succeed.
- K. Login as ocuser, password VMware123!

```

ocuser@OC-Apache-A: ~
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Thu Apr 14 19:24:31 PDT 2022

System load: 0.08      Processes:          160
Usage of /: 32.0% of 8.80GB   Users logged in:    1
Memory usage: 20%           IP address for ens160: 10.1.1.18
Swap usage:  0%

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

  https://ubuntu.com/blog/microk8s-memory-optimisation

101 updates can be applied immediately.
73 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Mon Apr 11 12:24:56 2022 from 10.0.0.2
ocuser@OC-Apache-A:~$ 

```

L. Attempt to ssh from OC-Apache-A to OC-Apache-B 10.1.1.19

M. Your connection should be refused

```

Last login: Thu Apr 14 19:24:32 2022 from 10.0.0.2
ocuser@OC-Apache-A:~$ ssh 10.1.1.19
ssh: connect to host 10.1.1.19 port 22: Connection refused
ocuser@OC-Apache-A:~$ ^C
ocuser@OC-Apache-A:~$ 

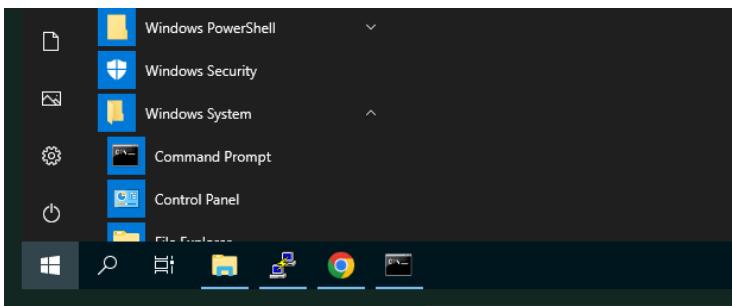
```

N. Observe that we have blocked ssh within the Web servers but allow between our admin network and the web servers.

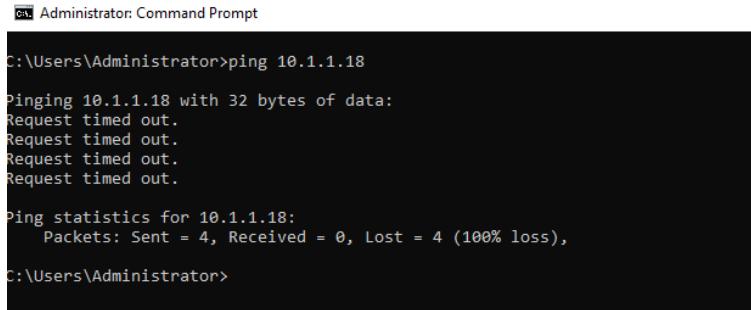
Step 5: Implement ICMP-Admin rule

The step will implement a rule to allow ICMP (Ping) to our Apache and MySQL VM's, but only from our inside admin network (10.0.0.0/24), as Ping is used to determine host accessibility in many security threat situations.

A. Open a command prompt on the Windows desktop by clicking the window icon -> Windows System -> Command Prompt



B. Attempt to ping to OC-Apache-A 10.1.1.18. Your connection should time out



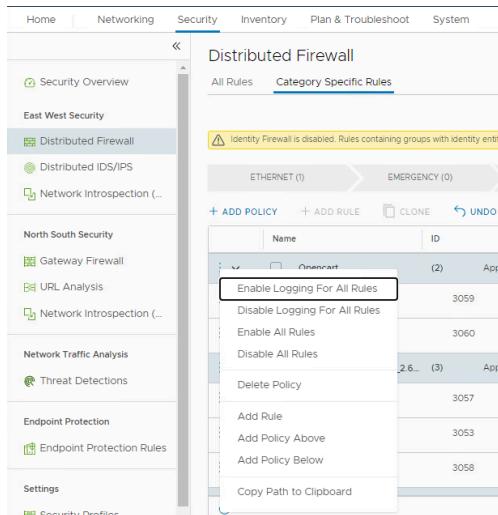
```
C:\> Administrator: Command Prompt
C:\Users\Administrator>ping 10.1.1.18

Pinging 10.1.1.18 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.1.18:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\Users\Administrator>
```

C. On the NSX Manager tab, go to Security -> Distributed Firewall

D. Click on the 3 dots next to Opencart and Add New Rule



E. Name the rule ICMP-Admin

Name	ID	Sources	Destinations	Services	Context Profiles	Applied To	Action
Opencart (5)	Applied To 2 Groups						Success
ICMP-Admin		Any	Any	Any	None	DFW	Allow

F. Click on the pencil for Sources. Then IP Addresses. Set to 10.0.0.0/24, then click Apply

G. Click on the pencil for Services. Select ICMP-ALL, then click Apply

Name	Service Entries	Status
ICMP ALL	ICMPv6 ICMPv4	Success

H. Publish your new rule

Name	ID	Sources	Destinations	Services	Context Profiles	Applied To	Action	Status
OpenCart	(5)	Applied To	2 Groups					Success
ICMP-Admin			Any	ICMP ALL	None	DFW	Allow	
ssh-admin	3063		Any	SSH	None	DFW	Allow	

I. Return to your Windows command prompt

J. Attempt to ping to OC-Apache-A 10.1.1.18. Your connection should succeed

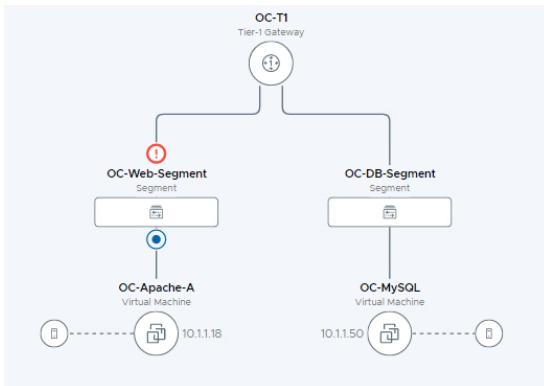
```
C:\Users\Administrator>ping 10.1.1.18

Pinging 10.1.1.18 with 32 bytes of data:
Reply from 10.1.1.18: bytes=32 time=9ms TTL=61
Reply from 10.1.1.18: bytes=32 time=2ms TTL=61
Reply from 10.1.1.18: bytes=32 time=2ms TTL=61
Reply from 10.1.1.18: bytes=32 time=2ms TTL=61

Ping statistics for 10.1.1.18:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 9ms, Average = 3ms
```

- K. From the NSX-T Manager interface click the **Plan & Troubleshoot** tab
- L. Click **Traceflow** in the left navigation panel
- M. Configure Traceflow from OC-Apache-A to OC-MySQL and click Trace

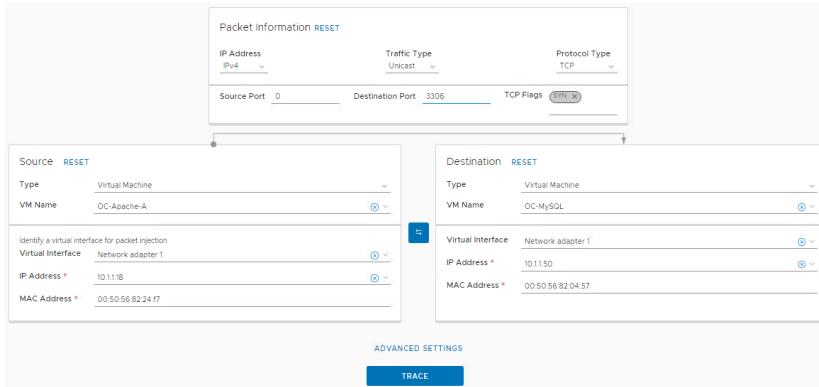
- N. Traceflow should show the ICMP packet dropped at the first firewall point (OC-Apache-A) before being placed on the segment



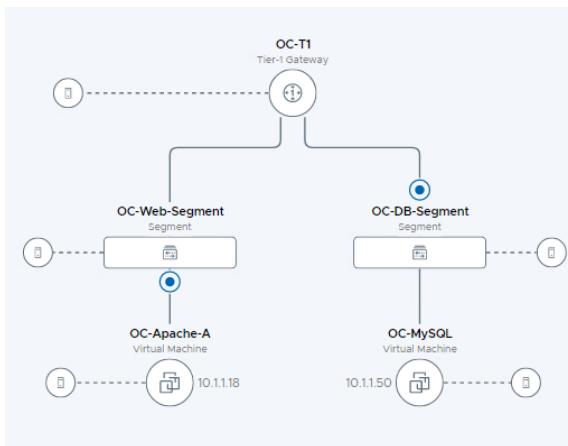
- O. Review the observations panel and notice the packet was dropped at OC-Apache A between the VM NIC and OC-Web-Segment

Observations	All	0 Delivered	1 Dropped			
Physical Hop Count	Observation Type	Transport Node	Component	Timestamp	IP Address	Actions
0	Injected	essi-4.vcf.sddc.lab	Network adapter 1	18:28:31.435.093		
0	Received	essi-4.vcf.sddc.lab	Distributed Firewall	18:28:31.435.197		
0	Dropped by Firewall Rule ID: 3076	essi-4.vcf.sddc.lab	OC-Apache-A.vmx@fele5793-65e0-4b56-8a9c-e65036632e71	18:28:31.435.208		

- P. On the Traceflow screen, click EDIT to reconfigure the trace then click Proceed on the warning pop up
 Q. Change Protocol Type from ICMP to TCP, with Destination Port 3306



- R. Click Trace. Your Traceflow should succeed



- S. In the Observations panel, review the following
- o We show 1 packet delivered
 - o The packet was injected at the network adapter for OC-Apache-A virtual machine
 - o It is then received at the distributed firewall at the VDS port for OC-Apache-A
 - o With no rule blocking, the packet is then forwarded on from the sending VDS port
 - o The packet then hits OC-T1 router and gets forwarded to the OC-DB-Segment
 - o Since OC-Apache-A and OC-MySQL are running on different ESXi hosts, you notice the physical hop between TEP's
 - o The packet is then received on the distributed firewall at the receiving VDS port for OC-MySQL
 - o With no rule blocking forwarding, the packet is then forwarded to the destination, the last step shows the packet being delivered to the network adapter for the OC-MySQL VM

Observations	All	1 Delivered	0 Dropped			
Physical Hop Count	Observation Type	Transport Node	Component	Timestamp	IP Address	Actions
0	Injected	esxi-1.vcf.sddc.lab	Network adapter 1	01/17/12 14:2:309		
0	Received	esxi-1.vcf.sddc.lab	Distributed Firewall	01/17/12 14:2:371		
0	Forwarded	esxi-1.vcf.sddc.lab	Distributed Firewall (Rule ID: 1014)	01/17/12 14:2:379		
0	Forwarded	esxi-1.vcf.sddc.lab	OC-Web-Segment	01/17/12 14:2:389		View Details
0	Received	esxi-1.vcf.sddc.lab	DC-T1	01/17/12 14:2:395		
0	Forwarded	esxi-1.vcf.sddc.lab	DC-T1	01/17/12 14:2:410		
0	Received	esxi-1.vcf.sddc.lab	OC-DB-Segment	01/17/12 14:2:440		View Details
0	Forwarded	esxi-1.vcf.sddc.lab	Physical	01/17/12 14:2:445	Local endpoint IP: 172.16.254.20 Remote endpoint IP: 172.16.254.14	
1	Received	esxi-4.vcf.sddc.lab	Physical	01/17/12 14:2:439	Local endpoint IP: 172.16.254.14 Remote endpoint IP: 172.16.254.20	
1	Received	esxi-4.vcf.sddc.lab	Physical	01/17/12 14:2:445	Local endpoint IP: 172.16.254.14 Remote endpoint IP: 172.16.254.20	
1	Received	esxi-4.vcf.sddc.lab	Distributed Firewall	01/17/12 14:2:537		
1	Forwarded	esxi-4.vcf.sddc.lab	Distributed Firewall (Rule ID: 1014)	01/17/12 14:2:546		
1	Delivered	esxi-4.vcf.sddc.lab	OC-MySQL.vmx@52c0d029-0b3e-4f7c-9abd-5a3f7ca76fb	01/17/12 14:2:548		

- T. Notice the flexibility of Traceflow to allow us to troubleshoot our distributed firewall and distributed routing using appropriate communications protocols.

[Lab 2 Summary]

Lab 2 shows the power of the distributed firewall capability in NSX. Using tagging a grouping, we were able to create a scalable set of rules for our Opencart application that only allow necessary communications for application operation, while blocking all other traffic. This was all done directly at the vSphere VDS switch port level, versus a piece of hardware elsewhere in the datacenter.

Module 3: Load Balancing

This module will add a Load Balancer for HTTP traffic

Lab 1: Configure load balancer

This lab will configure an L3-L7 load balancer on the NSX Tier-1 Router created in a Module 1. (

Step 1: Configure OC-T1 to run on an Edge Cluster

To support stateful services, such as Layer 3-7 Firewall we need to configure OC-T1 as a Services Router (SR). This simply means associating OC-T1 with our existing NSX Edge cluster in management domain

- Open a new tab in the Chrome browser
- Click the Management NSX-T shortcut in the bookmark bar (click advanced / proceed to nsx-mgmt.vcf.sddc.lab, if required to accept the certificate)
- Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
- Navigate to NSX Networking -> Tier-1 Gateways
- Click the 3 dots next to OC-T1 and select edit
- Click on Select Edge Cluster and select EC-01

Tier-1 Gateways

ADD TIER-1 GATEWAY

Tier-1 Gateway Name	Linked Tier-0 Gateway	#Linked Segments	Status
domain<8-5fa5e60a-6ba3-449c-a30f-a4ef0Ded29f	VLC-Tier-0	1	Success

OC-T1

Edge Cluster: EC-01

Fall Over: Non Preemptive

Edges Pool Allocation Size: ROUTING

Description: Description

Edges: VLC-Tier-0

DHCP: Auto Allocated | Set If no edges has been selected the system will auto-allocate. Set DHCP Configuration.

Enable Standby Relocation:

Tags: Tag Scope Max 30 allowed. Click (+) to add.

SAVE | **CANCEL** | **Unsaved Changes**

G. Click Save

H. Click Close Editing

Step 2: Create Server Pool

A server pool is a set of servers that can share the same content.

- A. Navigate to NSX Networking -> Load Balancing -> Server Pools. Click **Add Server Pool**
- B. Name the pool OC-LB-Pool

Load Balancing

Server Pools

ADD SERVER POOL

Name	Algorithm	Members/group	Virtual Servers
OC-LB-Pool *	Round Robin	Select Members	

Description: Enter Description

Active Monitor: Set

SNAT Translation Mode: Automap

SAVE | **CANCEL**

C. Click **Select Members**

D. Click **Add Member**

E. Name **OC-Apache-A**, IP **10.1.1.18** Port **80**

Configure Server Pool Members

Server Pool -

Enter individual members Select a group

Name	IP	Port	Weight	State	Backup Member	Max Concurrent Connections
OC-Apache-A	10.1.1.18 e.g. 10.10.10.10 or fc7ef206-db42::	80 e.g. 80, 443	1	Enabled	<input checked="" type="checkbox"/>	Disabled

SAVE **CANCEL**

F. Click SaveIP I

G. Repeat steps for OC-Apache-B, using:

- Name **OC-Apache-B**, IP **10.1.1.19** Port **80**

Configure Server Pool Members

Server Pool -

Enter individual members Select a group

Name	IP	Port	Weight	State	Backup Member	Max Concurrent Connections
OC-Apache-B	10.1.1.19 e.g. 10.10.10.10 or fc7ef206-db42::	80 e.g. 80, 443	1	Enabled	<input checked="" type="checkbox"/>	Disabled

SAVE **CANCEL**

H. Click Set next to Active Monitor

I. Select the default HTTP Port 80 monitor, then click Apply

Select Active Monitors

Server Pool - OC-LB-Pool

default-ntp-lb-monitor

ADD ACTIVE MONITOR

	Name	Protocol	Monitoring Port	Monitoring Interval	Timeout Period (seconds)	Server Pools
<input checked="" type="checkbox"/>	> default-ntp-lb-monitor	HTTP	80	5	5	0
<input type="checkbox"/>	> default-https-lb-monitor	HTTPS	443	5	5	0
<input type="checkbox"/>	> default-icmp-lb-monitor	ICMP		5	5	0
<input type="checkbox"/>	> default-tcp-lb-monitor	TCP		5	5	0

J. Click Save

Name	Algorithm	Members/Group
OC-LB-Pool	Round Robin	2
Description	Enter Description	Active Monitor 1
SNAT Translation Mode	Automap	
> Additional Properties		
SAVE		CANCEL

Step 3: Create Load Balancer

- Navigate to Networking -> Load Balancing

ADD LOAD BALANCER						
Name	Type	Size	Attachment	Virtual Servers	Status	Alarms
Enter Name	Server Load Balancer	Small	Select Tier-1 Gateway			
Description	Enter Description	Error Log Level		Info		
Tags	Tag Scope	Admin State		Enabled		
NOTE: Before further configurations can be done, fill out mandatory fields above (*), click 'Save' below.						
SAVE		CANCEL				

- Click Add Load Balancer
- Name the Load Balancer OC-LB. On Attachment select OC-T1

ADD LOAD BALANCER						
Name	Type	Size	Attachment	Virtual Servers	Status	Alarms
OC-LB	Server Load Balancer	Small	OC-T1			
Description	Enter Description	Error Log Level		Info		
Tags	Tag Scope	Admin State		Enabled		
NOTE: Before further configurations can be done, fill out mandatory fields above (*), click 'Save' below.						
SAVE		CANCEL				

- Click Save
- When prompted Want to continue configuring this Load Balancer, select No

Step 4: Create Virtual Servers

A Virtual Server is an IP address that acts as the front end for a Server Pool

- A. Navigate to NSX Networking -> Load Balancing -> Virtual Servers. Click **Add Virtual Server, L7 HTTP**

Name	IP Address	Ports	Type	Load Balancer	Server Pool	Status
ap0384fffd-22-default-kubernetes-TCP-443						

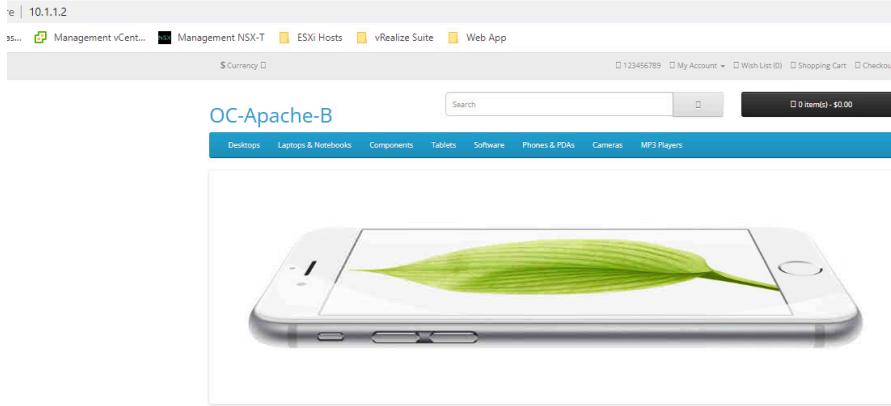
- B. Name the Virtual Server **OC-VIP**, IP Address **10.1.1.2**, Port **80**
 C. Type OC in the Load Balancer field and it will allow you to select OC-LB
 D. Type OC in the Server Pool field and select OC-LB-Pool
 E. Click on **SAVE**

Step 5: Test Load Balancer

- A. Open a new tab for 10.1.1.2



B. Refresh the browser for this tab. You should see the opposite web server



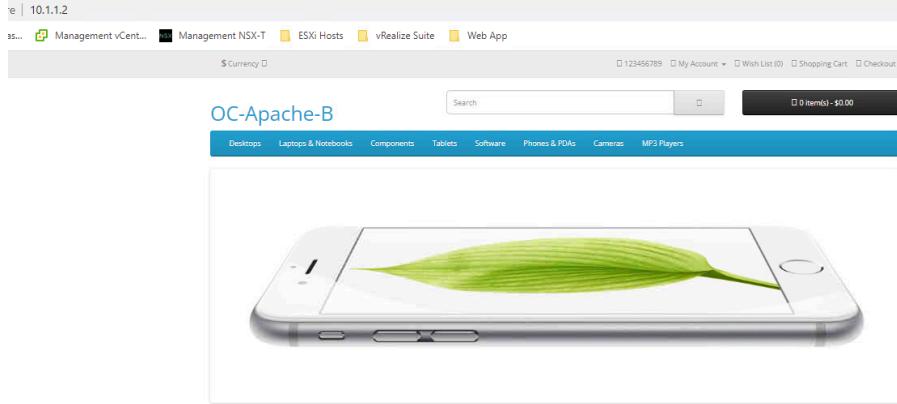
C. Open a PuTTY session to OC-Apache-A, login **ocuser**, password **VMware123!**

D. Run the command **sudo systemctl stop apache2**

A screenshot of a terminal window titled 'ocuser@OC-Apache-A: ~'. The user runs the command 'sudo systemctl stop apache2'. The terminal shows the command being typed and then executed, with the output indicating success.

E. Wait approximately 30 seconds

F. Refresh the browser for the 10.1.1.2 tab several times in a row. You should see the OC-Apache-B web page only, as the Active Monitor would detect the failure of OC-Apache-A quickly



- G. Return to the OC-Apache-A PUTTY session and run the command **sudo systemctl start apache2**
- H. Wait approx. 30 seconds
- I. Refresh the browser for the 10.1.1.2 tab several times in a row. You should see both OC-Apache-A and OC-Apache-B web pages, as the Active Monitor would detect the return of OC-Apache-A quickly

[Lab 1 Summary]

Lab 1 shows how quickly a load balancer can be instantiated on the NSX Tier-1 router



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 vmware.com Copyright © 2020 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 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. Item No: vmw-wp-temp-uslet-word-101-proof 6/20