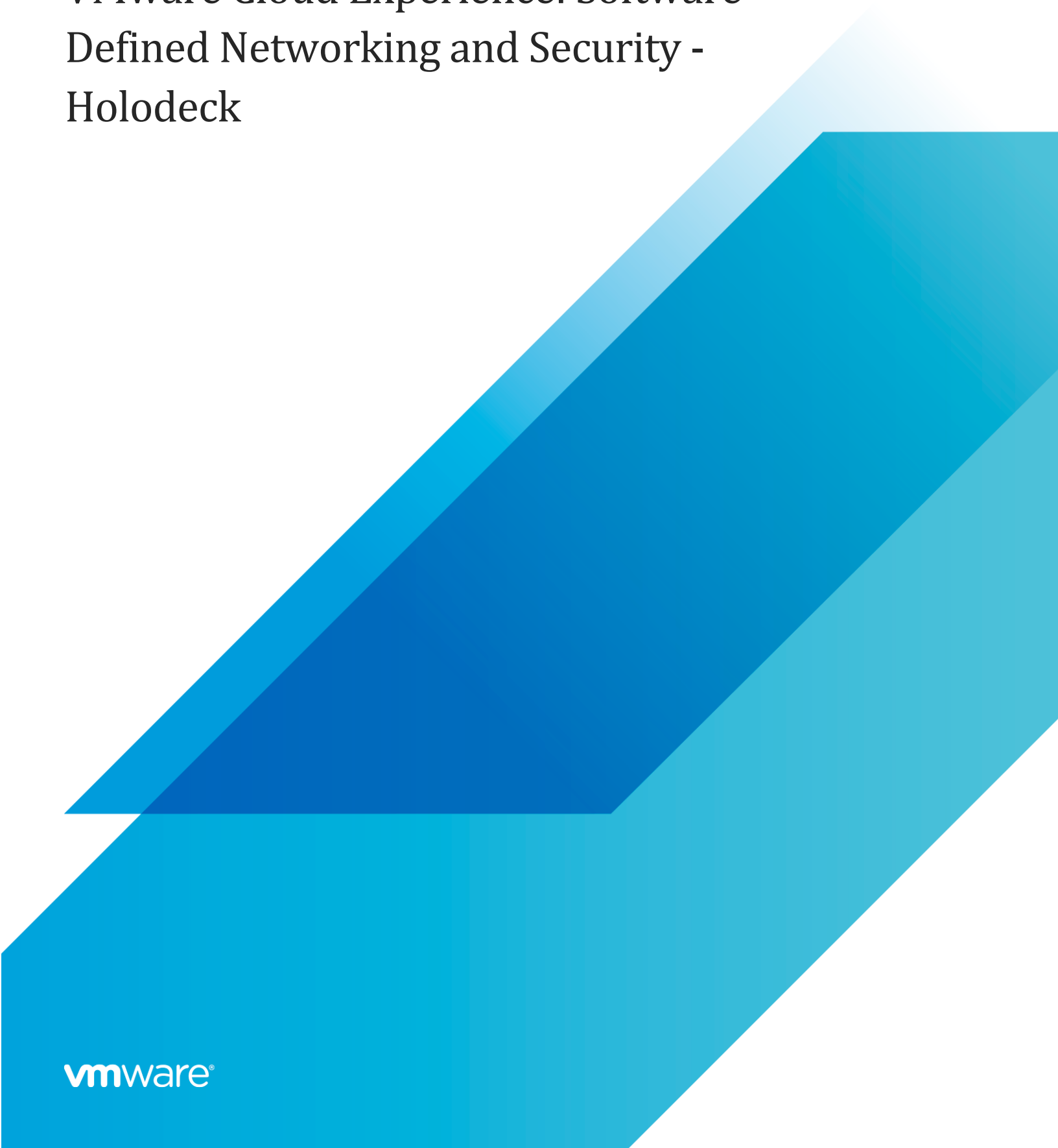
VMware Cloud Experience: Software Defined Networking and Security - Holodeck



vmware®

Table of contents

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 MqSQL 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.

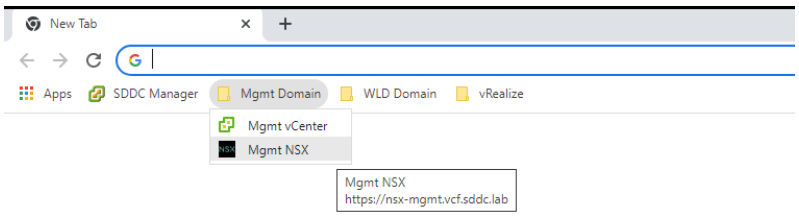
NSX enables customers to create elastic, logical networks that span physical network boundaries. NSX abstracts the physical network into a pool of capacity and separates the consumption of these services from the underlying physical infrastructure. This model is similar to the model vSphere uses to abstract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

Lab 1: Creating Network Segments

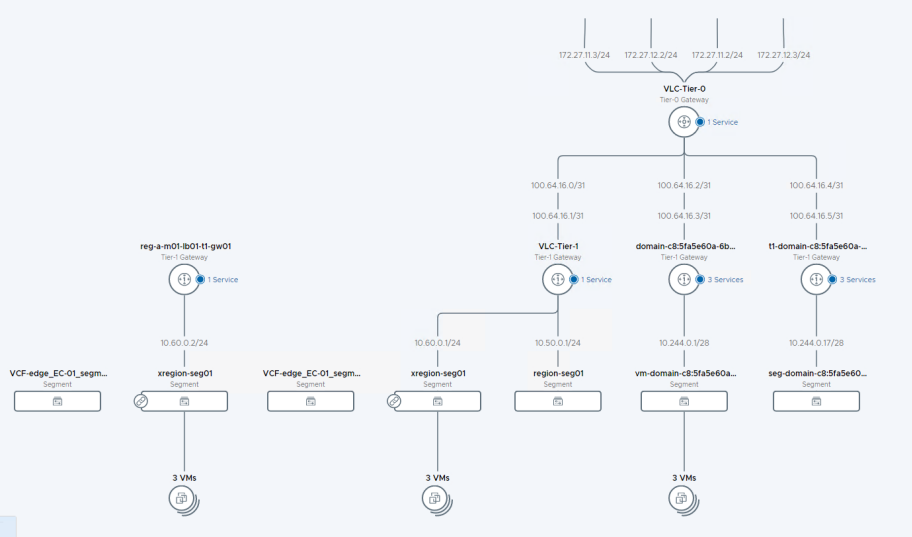
This exercise will deploy the necessary networking components to support a simple two-tier application called "Opencart". *Opencart* is an opensource E-Commerce platform that uses an Apache front end, and a MySQL backend. This lab uses preconfigured Apache and MySQL VM's that we will attach to newly 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**



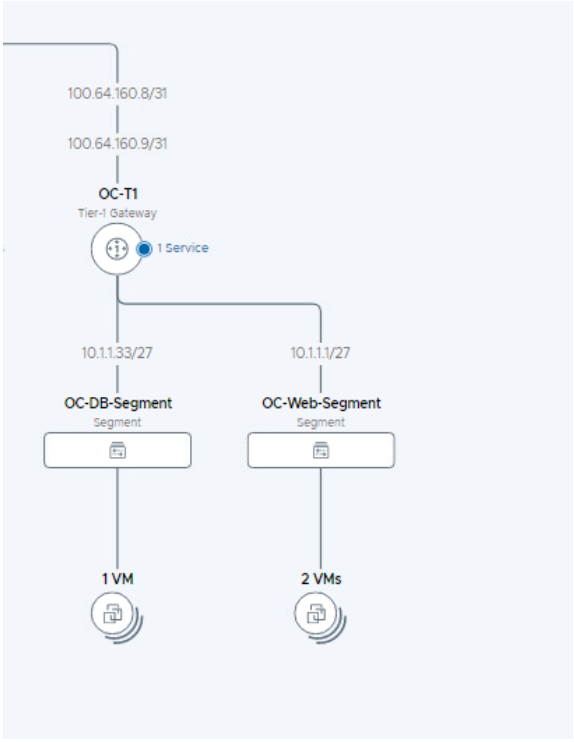
What you see are existing segments and routers that are used elsewhere in the lab environment. By the end of this lab exercise, you create the configuration shown below, with 2 new segments, connected to a Tier-1 gateway, with a load balancer and firewall configured.

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)

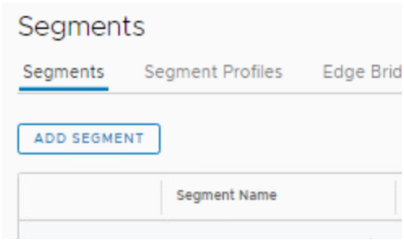
OC-DB-Segment

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

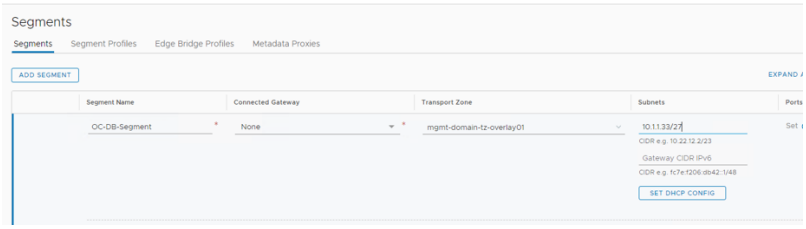


[Step 2a] Create the OC-DB-Segment

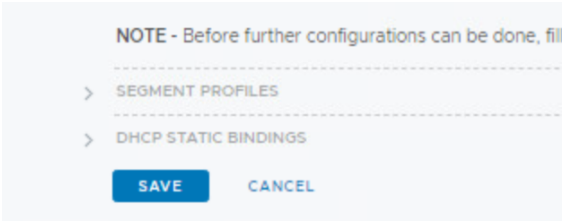
A. From the NSX-T Manager interface click the **Networking** tab at the top of the screen
B. Click **Segments** in the left pane.
C. Click **ADD SEGMENT** button



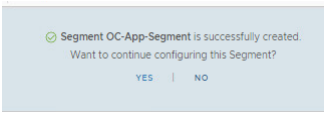
D. In the **Segment Name** field, enter **OC-DB-Segment**
E. Leave the **Connected Gateway** field blank
F. In the **Transport Zone** dropdown, select **mgmt-domain-tz-overlay01 | Overlay**
G. Add IPv4 gateway 10.1.1.33/27



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

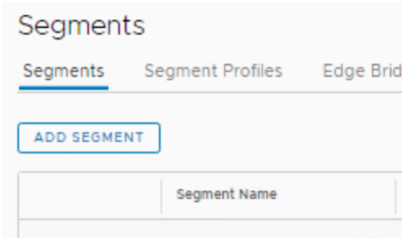


I. You will see your segment has been successfully created. Click **NO** on the Want to continue configuring this segment?

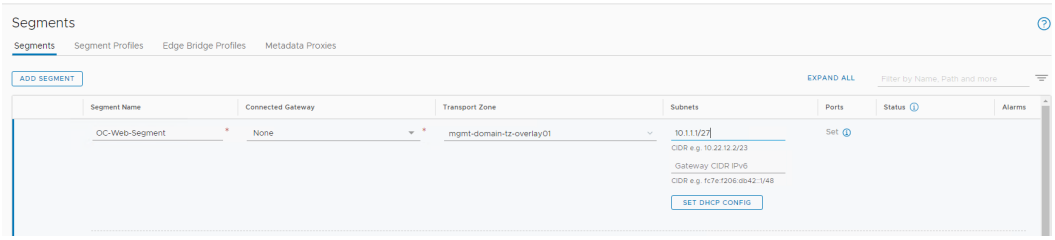


[Step 2b] Create the OC-Web-Segment

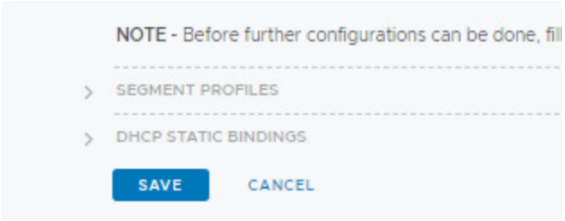
A. From the NSX-T Manager interface click the **Networking** tab at the top of the screen
B. Click **Segments** in the left pane.
C. Click **ADD SEGMENT** button



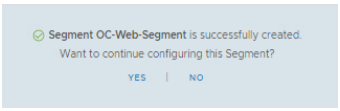
D. In the **Segment Name** field, enter **OC-Web-Segment**
E. Leave the **Connected Gateway** field blank
F. In the **Transport Zone** dropdown, select **mgmt-domain-tz-overlay01 | Overlay**
G. Add **IPv4 gateway 10.1.1.1/27**



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

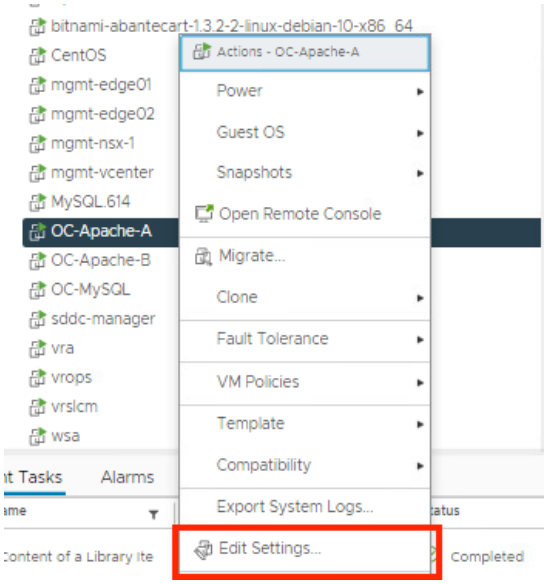
This step will attach our two Apache web server VM's to the OC-Web-Segment. Begin in the vCenter Server Web Interface:

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

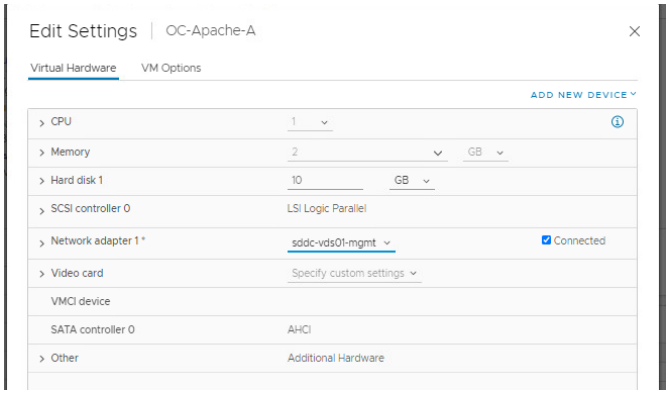
[Step 3.1] Attach OC-Apache-A to segment **OC-Web-Segment**

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

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

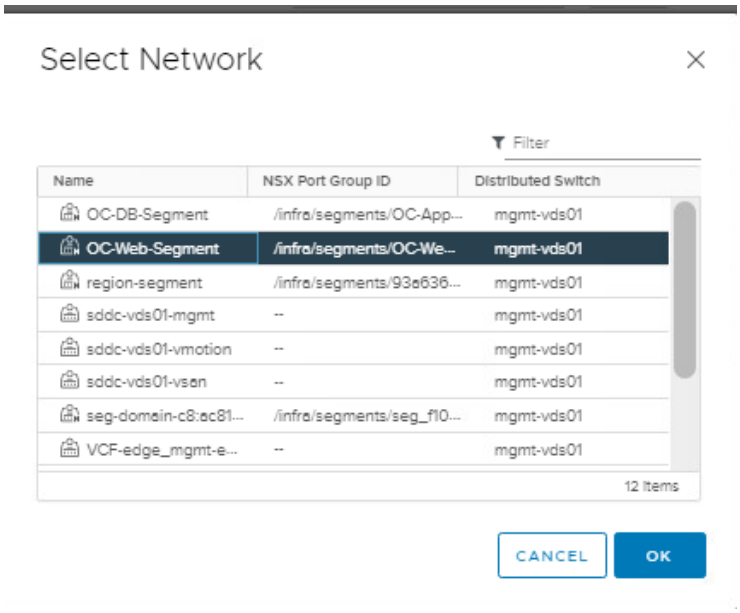


B. Click the dropdown next to **Network Adapter 1**



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.

A. Right click on **OC-Apache-B** and click **Edit Settings**
B. Click the dropdown next to **Network Adapter 1**
C. Click **Browse** click on **OC-Web-Segment**
D. 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

A. 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::250:56ff: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:~$ _
```

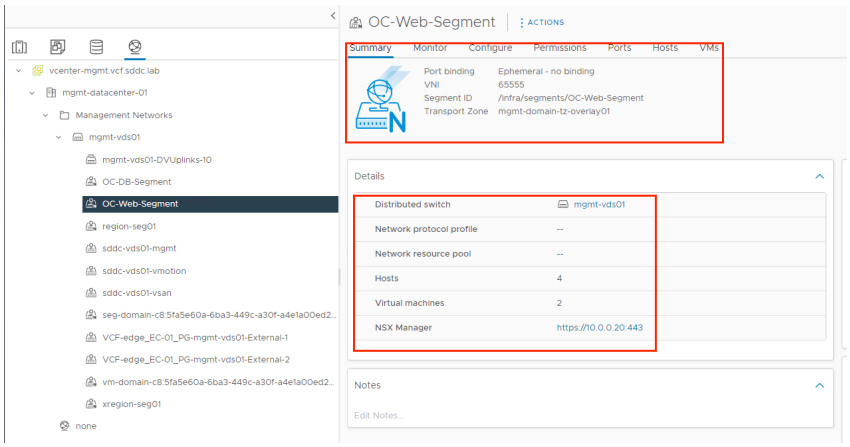
E. Notice we can communicate from OC-Apache-B to OC-Apache-A on a network we just created.

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

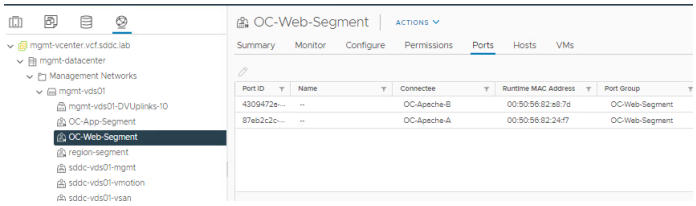
A. Click on the Management vCenter Server Tab
B. Click Menu -> Networking
C. Click on OC-Web-Segment
D. Notice the “N” showing this is an NSX segment versus a standard port group
E. The segment ID is shown along with the transport zone the segment is a part of
F. Lower on the screen you can see which 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*

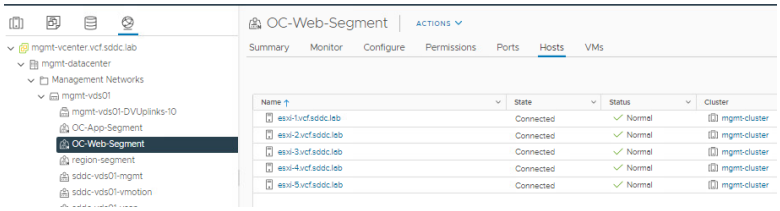
G. Finally notice the NSX Manager for this segment is hot linked



H. Click on the Ports tab
I. 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

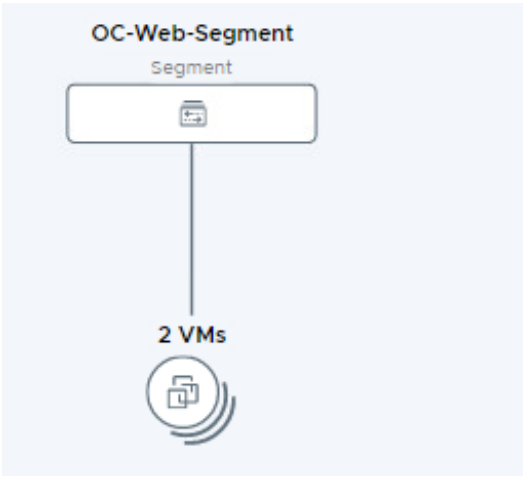


J. Click on the Hosts tab
K. Note that our OC-Web-Segment is connected on each ESXi host in the transport zone. When a segment is created, it is accessible to all hosts in the transport zone.

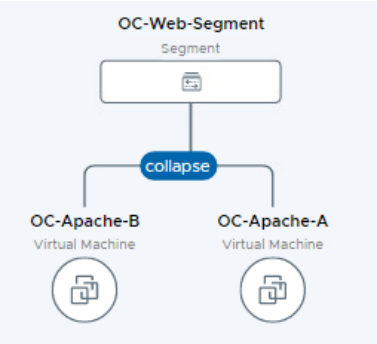


[Step 3.5] Discover the Network Topology

A. Click on the NSX Manager tab in the browser
B. If needed, Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
C. From the NSX-T Manager interface click the **Networking** tab
D. Select **Network Topology**
E. 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



G. Click on **2 vms** under OC-Web-Segment
H. 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

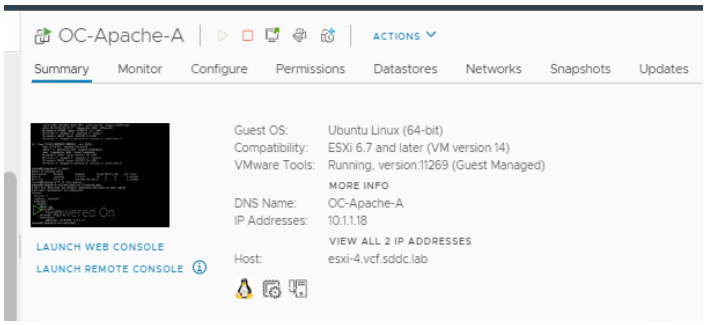
In this lab we will use the Traceflow capability in NSX to view traffic moving between virtual machines on the same host, on the same segment.

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

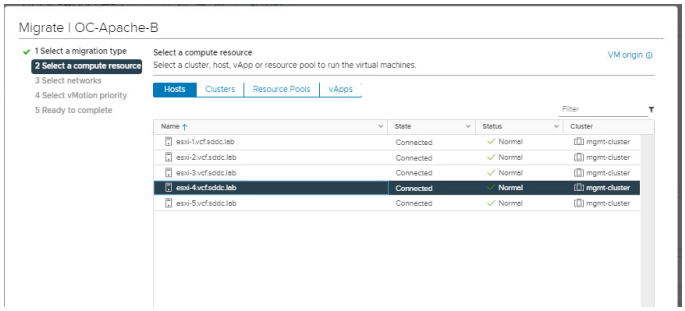
[Step 1] Setup VMs for test

A. Click on the Management vCenter Server tab in the browser
B. If session has timed out, login user: **administrator@vsphere.local** and password: **VMware123!**
C. 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:

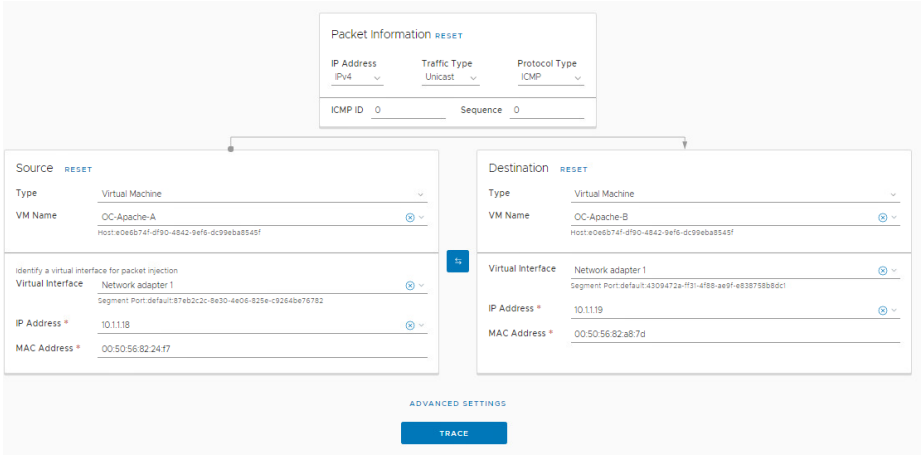


D. Next click on OC-Apache-B to determine its host. If it is running on the same host as OC-Apache-A, skip forward a step
E. If OC-Apache-B is not on the same host as OC-Apache-A, initiate a vMotion to move it to the same host

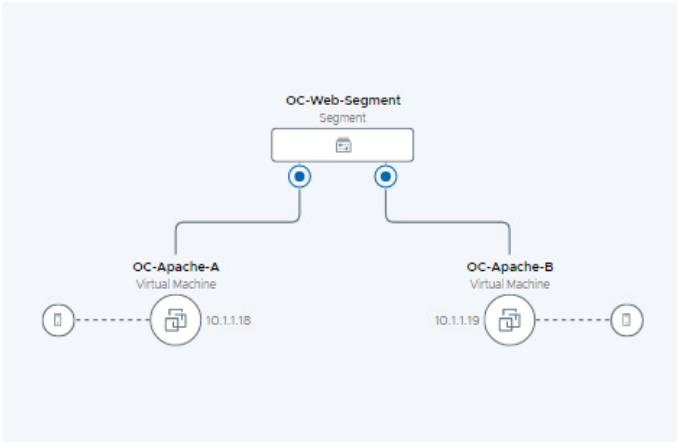


[Step 2] Test packet flow

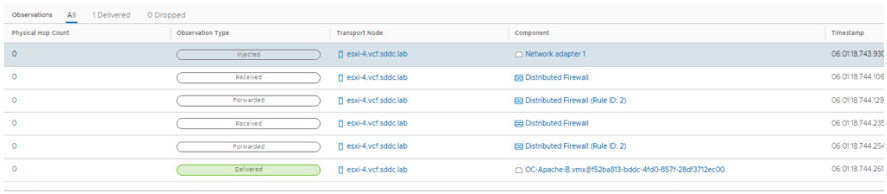
A. Click on the **Mgmt Domain** tab in the browser, and select **Mgmt NSX**.
B. If necessary, login as user: **admin** password: **VMware123!VMware123!**
C. Click **Plan and Troubleshoot** on the top menu bar
D. Click **Traceflow** on the left menu bar
E. Setup a Traceflow between OC-Apache-A and OC-Apache-B.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



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
   - We show 1 packet delivered
   - The physical hop count is 0, indicating that the packet did not leave the host
   - 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 is then received on the distributed firewall at the receiving VDS port for OC-Apache-B.
   - 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-Apache-B VM



[Lab 2 Summary]

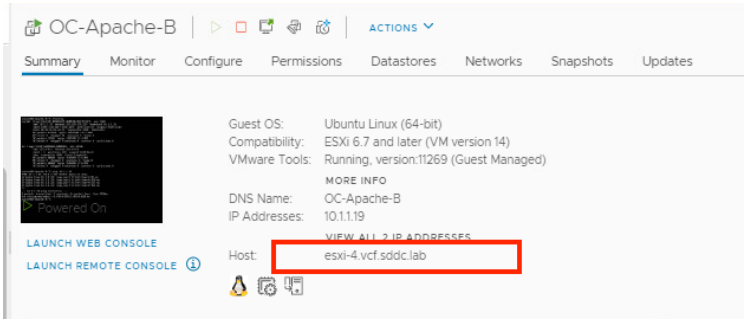
Lab 2 shows how ICMP packets travel between the VDS ports for 2 virtual machines running on the same ESXi host. You can see the packet pass from where it enters the VDS at the source, passes through the source side firewall, then get forwarded to the destination distributed firewall, and finally to the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

A. Click on the Management vCenter Server tab in the browser
B. If session has timed out, login user: **administrator@vsphere.local** password*:* **VMware123!**
C. 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



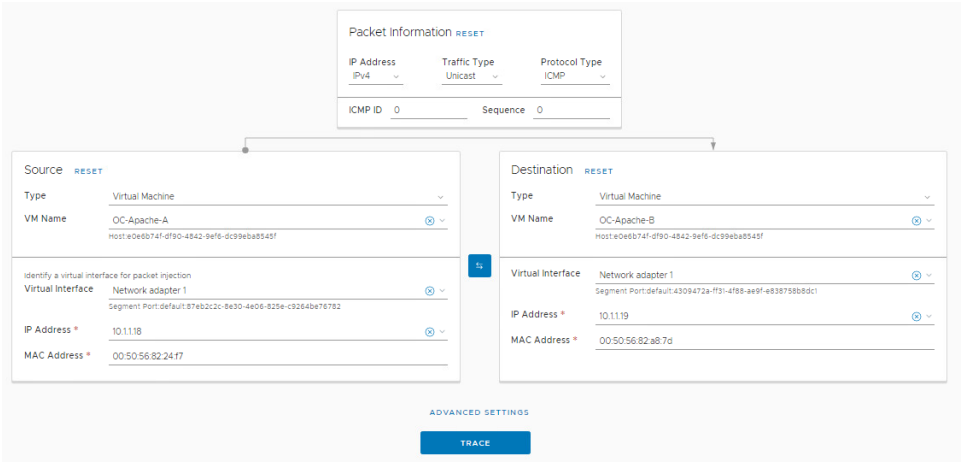
D. Next click on OC-Apache-B to determine its host. If OC-Apache-B is on the same host as OC-Apache-A, initiate a vMotion to move OC-Apache-B to the different host. As we are using vSAN storage, you only need to vMotion compute

[Step 2] View packet flow

A. Click on the NSX Manager tab in the browser
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

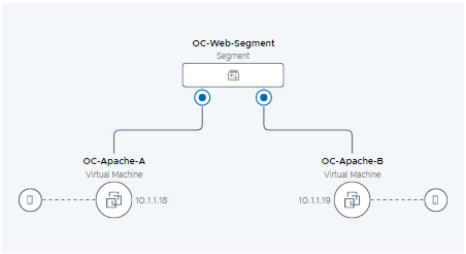
E. If your most recent Traceflow is still on screen between OC-Apache-A and OC-Apache-B, click **Retrace** and skip forward to step H

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

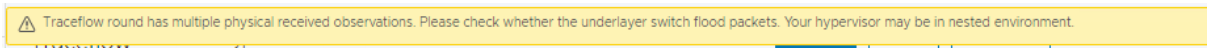


G. Click **Trace**

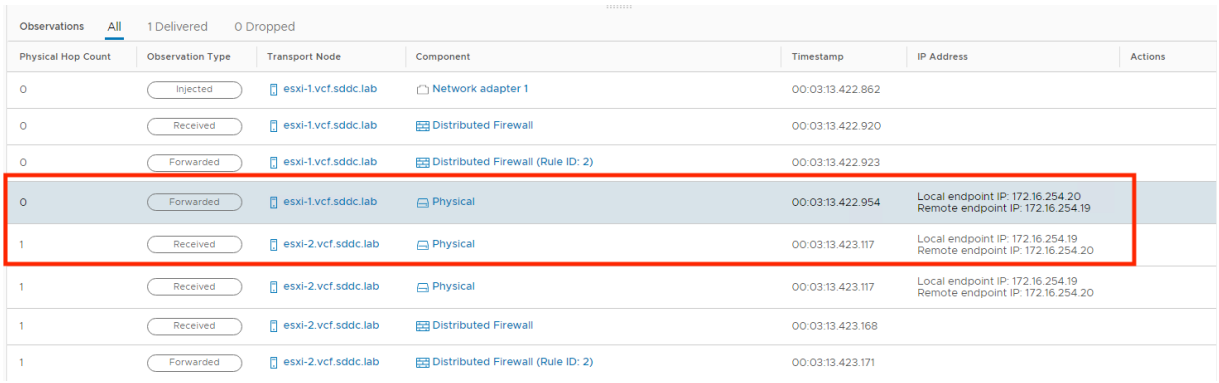
H. Notice the path the data packets take on the resulting topology view. We can see that packets move from app-01a to web-01a via the one-tier segment



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

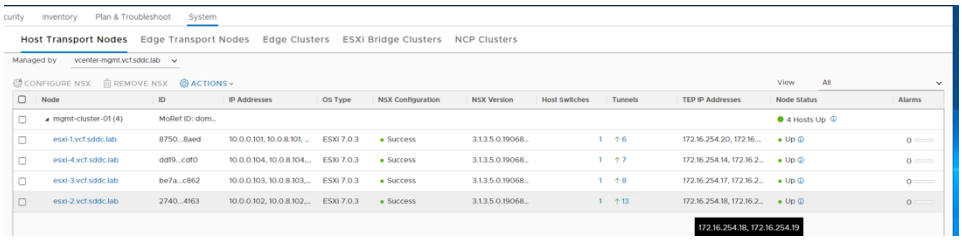


J. In the Observations panel, review the following
   - We show 1 packet delivered
   - The physical hop count increments to 1 part way through the flow, indicating that the packet left the host
   - The packet was injected at the network adapter for OC-Apache-A virtual machine
   - It is then received 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 the physical layer to transmit to the second host. Notice the local and remote endpoints are shown
   - 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
   - The packet is then received on the distributed firewall at the receiving VDS port for OC-Apache-B.
   - 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-Apache-B VM

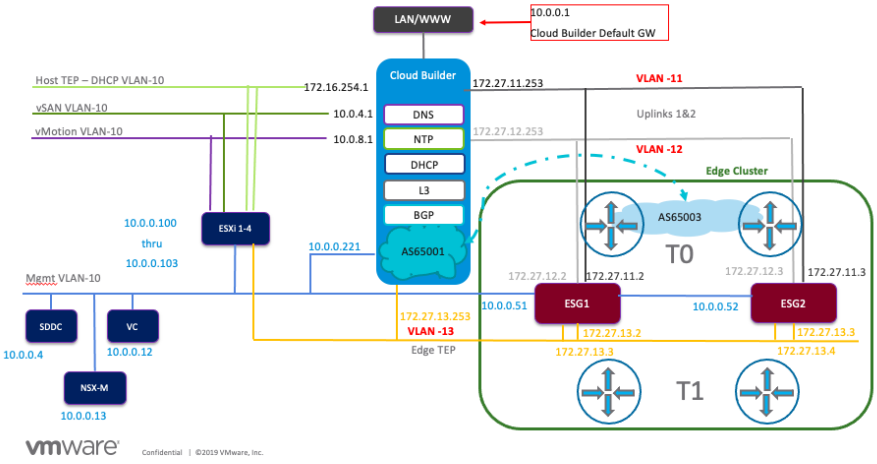


[Step 3] View Host TEP information

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



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 prepped 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]

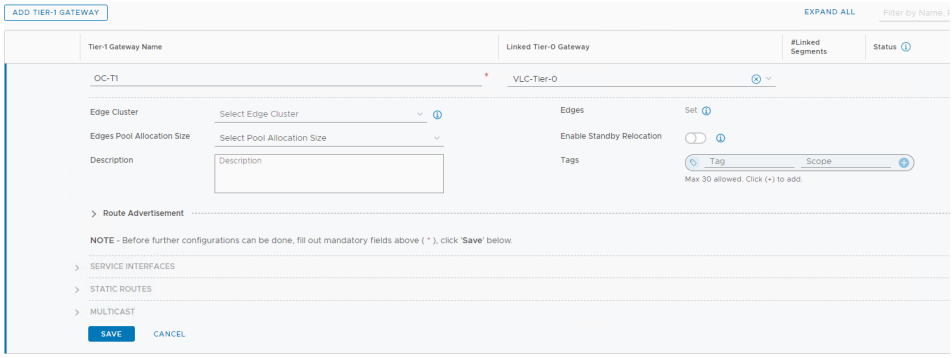
In Lab 3 we show how ICMP packets travel between the VDS ports for 2 virtual machines running on different ESXi hosts. Like lab 2, you first see the packet pass from where it enters the VDS at the source, through the source side firewall. The packet then gets encapsulated and placed on the segment (overlay 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

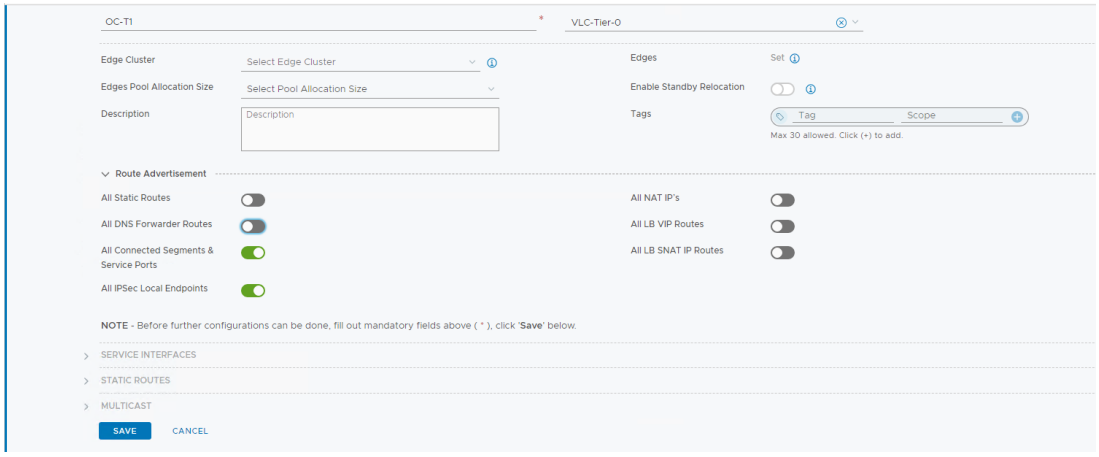
While we created two segments earlier in the lab, the segments are currently not connected together, or to other parts of the network. In this lab we will add an NSX Tier-1 router, connect OC-Web-Segment and OC-App-Segment to the T1 router, then connect to T1 router to the existing T0 router in the lab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

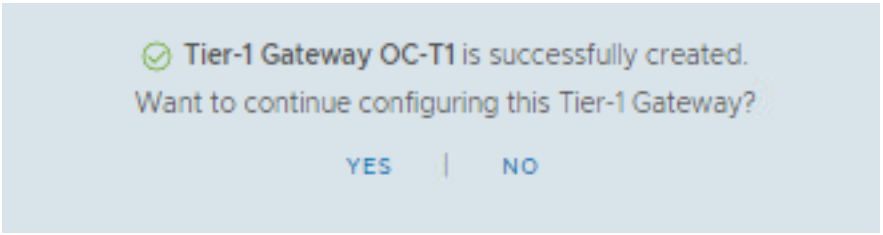


H. Expand Route Advertisement
I. Enable All Connected Segments & Service Ports



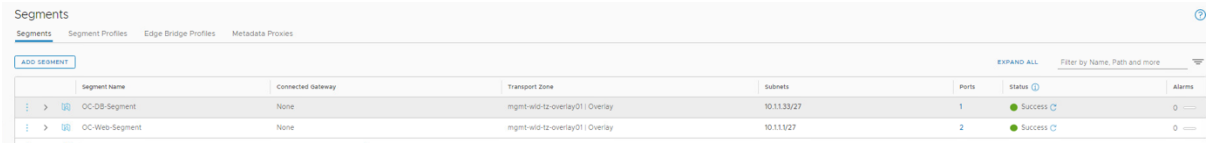
J. Click save

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

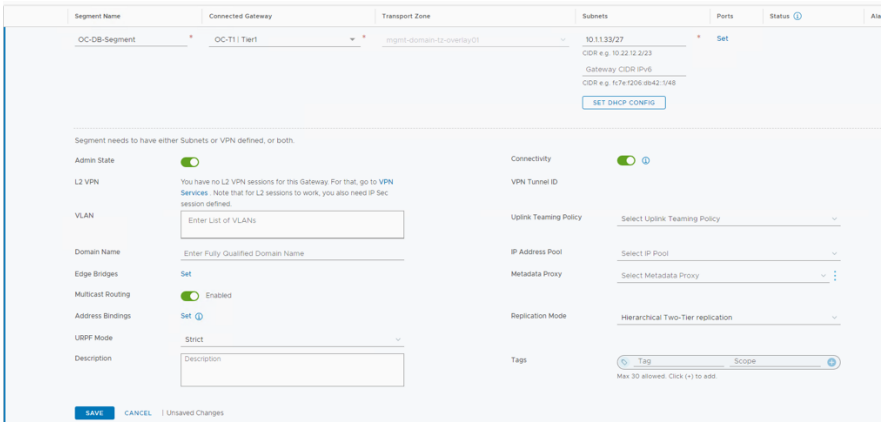


[Step 2] Connect segments to OC-T1

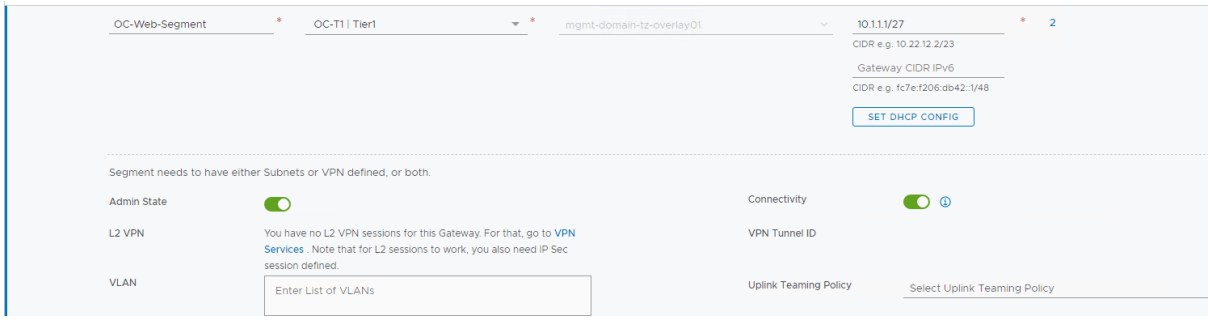
A. If necessary, open a new tab in the Chrome browser
B. Click the Management NSX-T shortcut in the bookmark bar (click advanced / proceed to nsx-mgmt.vcf.sddc.lab, if required to accept the certificate)
C. Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
D. From the 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



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



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

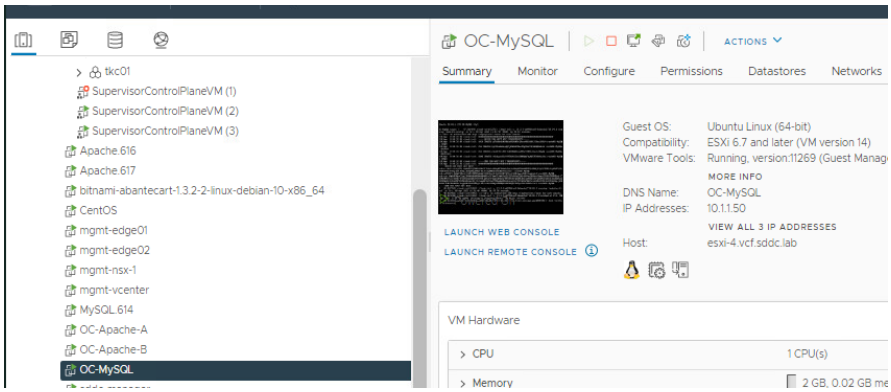


K. Click Save -> Close Editing

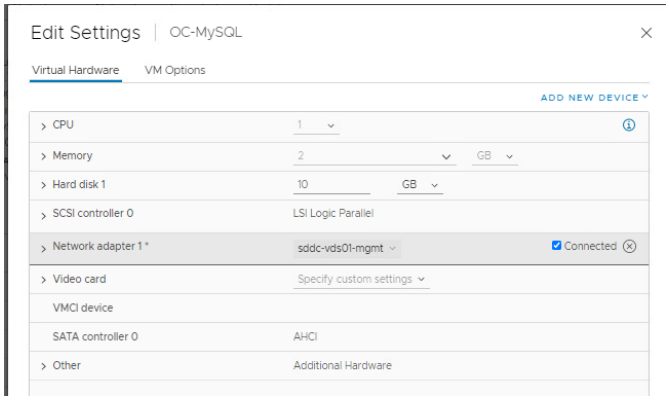
[Step 3] Attach OC-MySQL to segment **OC-DB-Segment**

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**

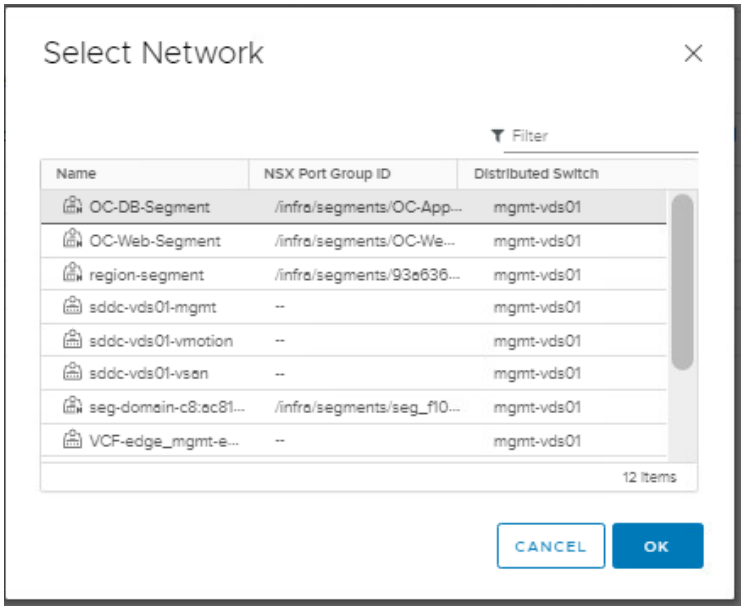


B. Click the dropdown next to **Network Adapter 1**



C. Click **Browse**

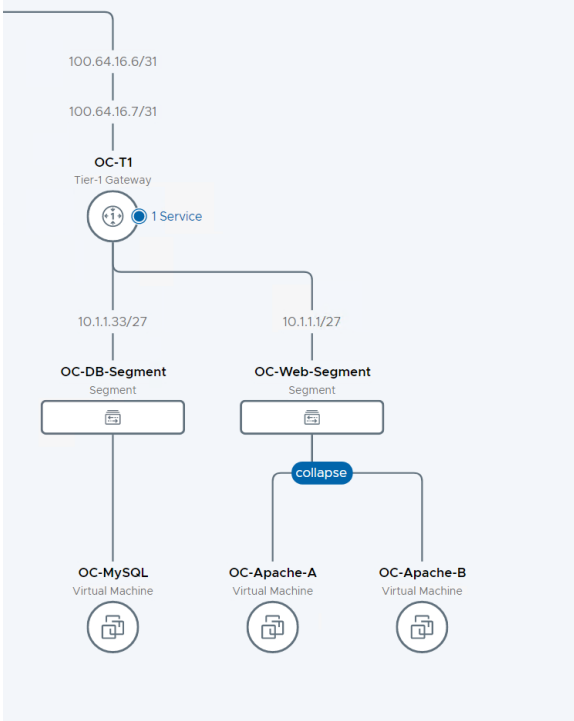
D. Click **OC-DB-Segment**



E. Click **OK**
F. Click **OK**

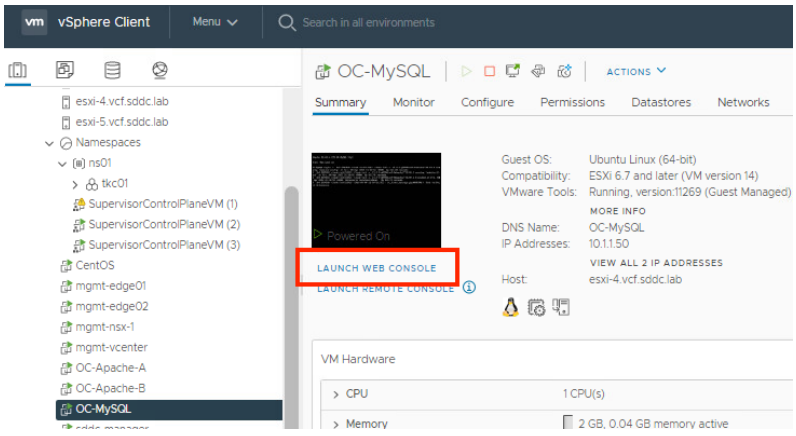
[Step 4] Review network topology

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 **Network Topology** in the left navigation panel
F. 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

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

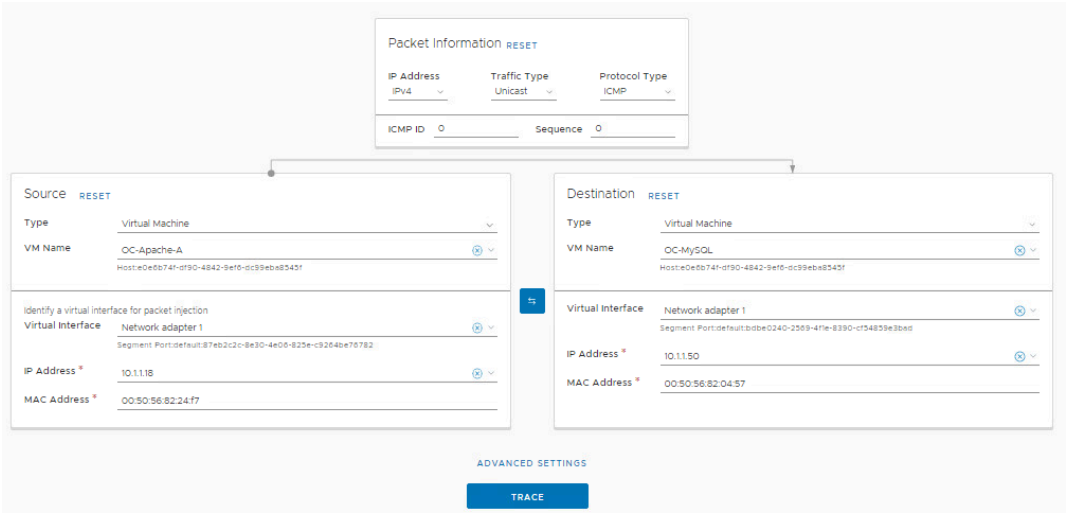


C. Login **ocuser** password **VMware123!**
D. Ping OC-Apache-A (10.1.1.18). Successful ping means that OC-MySQL can communicate with OC-Apache-A via the OC-T1 router

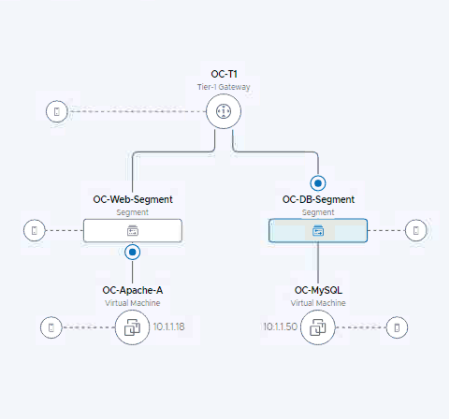
```
ocuser@OC-MySQL:~$ ping 10.1.1.18
PING 10.1.1.18 (10.1.1.18) 56(84) bytes of data.
64 bytes from 10.1.1.18: icmp_seq=1 ttl=63 time=2.10 ms
64 bytes from 10.1.1.18: icmp_seq=2 ttl=63 time=0.590 ms
64 bytes from 10.1.1.18: icmp_seq=3 ttl=63 time=0.601 ms
64 bytes from 10.1.1.18: icmp_seq=4 ttl=63 time=0.390 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

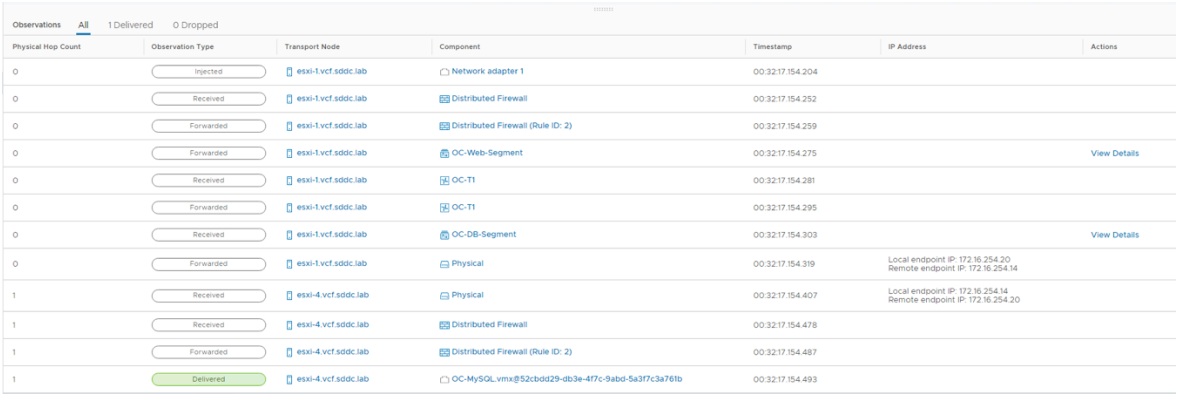
A. If necessary, open a new tab in the Chrome browser
B. Click the Management NSX-T shortcut in the bookmark bar (click advanced / proceed to nsx-mgmt.vcf.sddc.lab, if required to accept the certificate)
C. Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
D. From the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



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



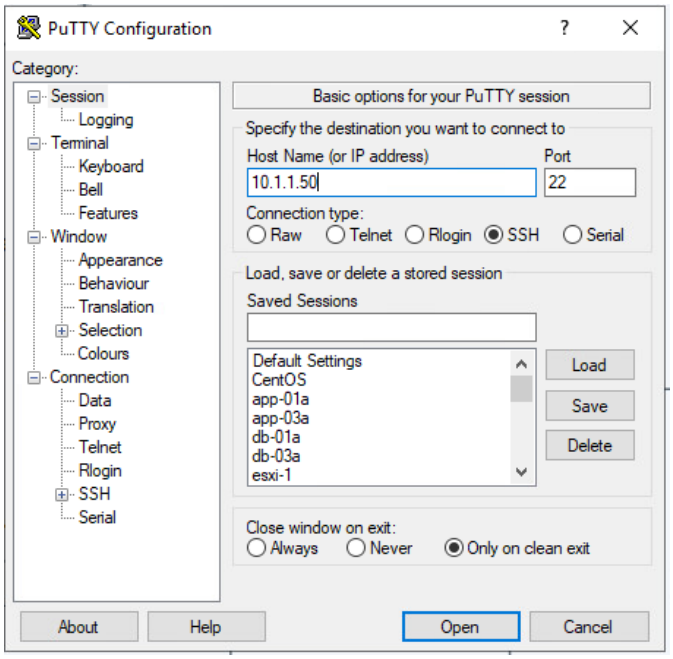
H. In the Observations panel, review the following
   - We show 1 packet delivered
   - The packet was injected at the network adapter for OC-Apache-A virtual machine
   - It is then received at the distributed firewall at the VDS port for OC-Apache-A
   - With no rule blocking, the packet is then forwarded 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 All 1 Delivered 0 Dropped

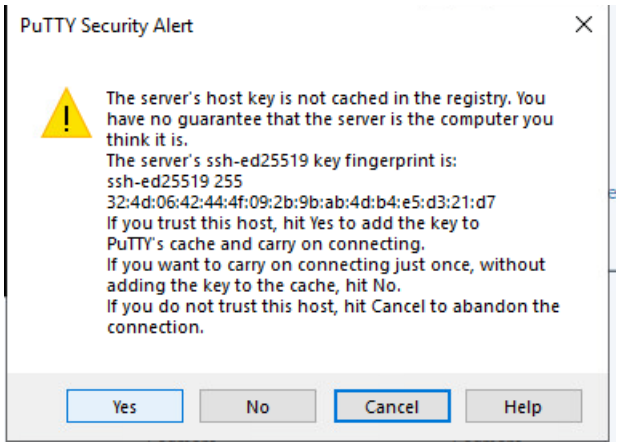
Physical Hop Count	Observation Type	Transport Node	Component	Timestamp	IP Address	Actions
0	Injected	esxi-1.vcf.sddc.lab	Network adapter 1	00:32:17.154.204		
0	Received	esxi-1.vcf.sddc.lab	Distributed Firewall	00:32:17.154.252		
0	Forwarded	esxi-1.vcf.sddc.lab	Distributed Firewall (Rule ID: 2)	00:32:17.154.259		
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@52cbdd29-db3e-4f7c-9abd-5a3f7c3a761b	00:32:17.154.493		

[Step 6] Test end to end communications

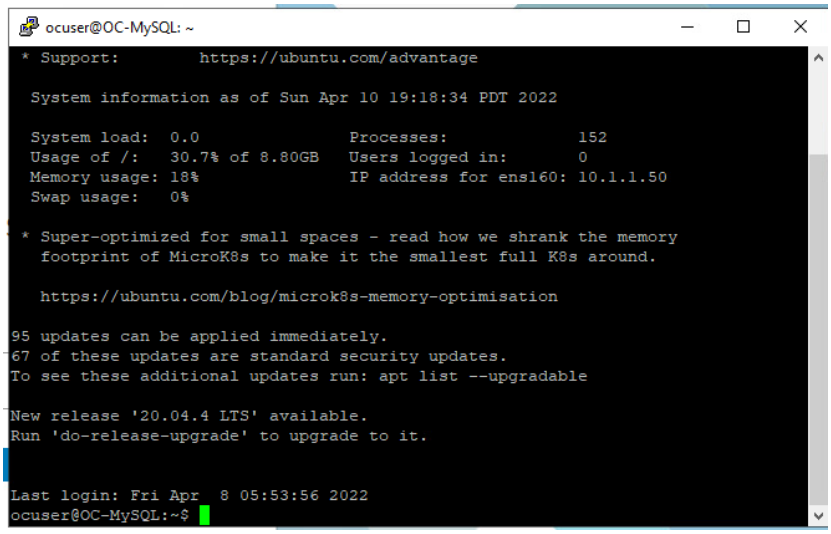
A. Click on the PuTTY icon
B. Connect to 10.1.1.50 (OC-MySQL).



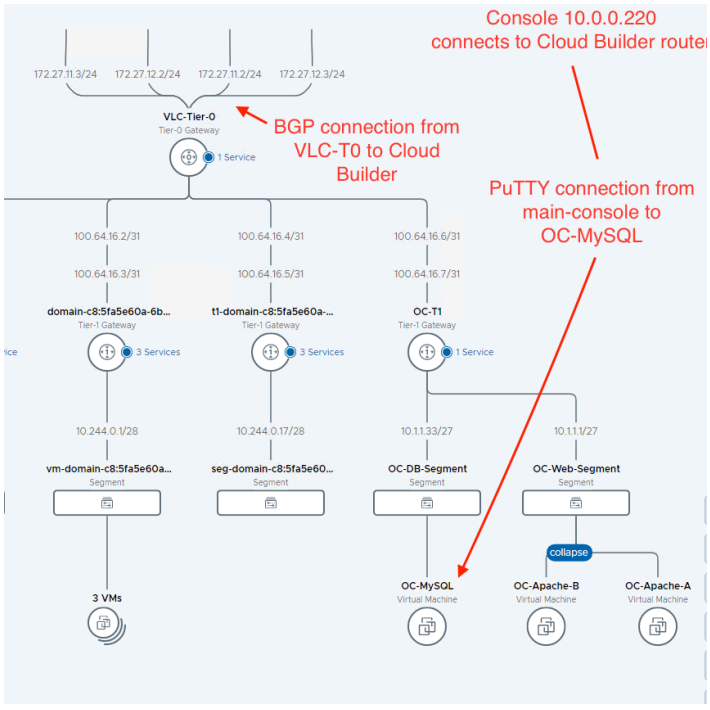
C. Click yes to accept ssh fingerprints if prompted



D. Login ocuser password VMware123!



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]

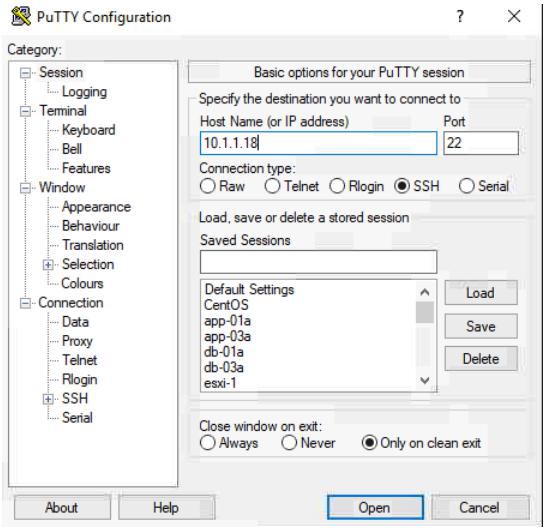
In Lab 5 we show packets travel between the VDS ports for 2 virtual machines running on different ESXi hosts across two segments connected by a Tier-1 router. The important distinction is this router functionality was distributed across all hosts versus a physical device cabled somewhere else in the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

Lab 5: Testing the application

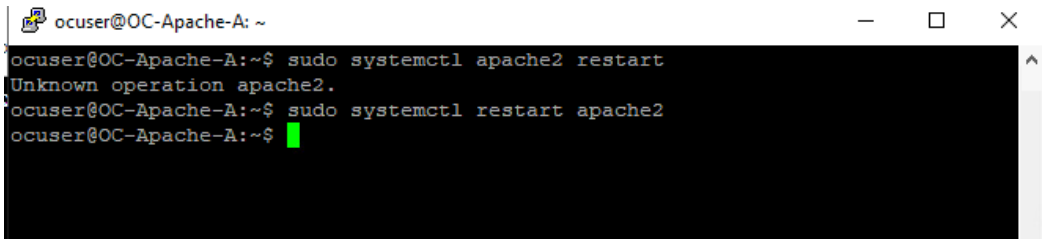
This lab will validate our web server VM's and database VM are working correctly prior to moving on to load balancing and security

[Step 1] Restart web servers

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



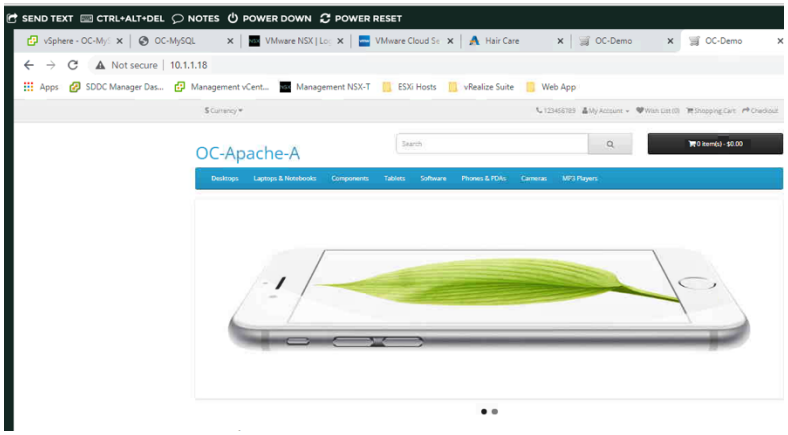
C. Login **ocuser** password **VMware123!**
D. Run the command **sudo systemctl restart apache2**



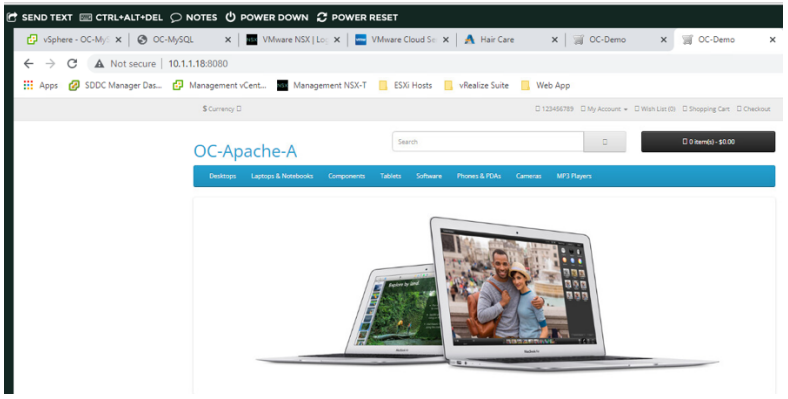
E. Repeat steps A-D for OC-Apache-B 10.1.1.19

[Step 2] Test Opencart app

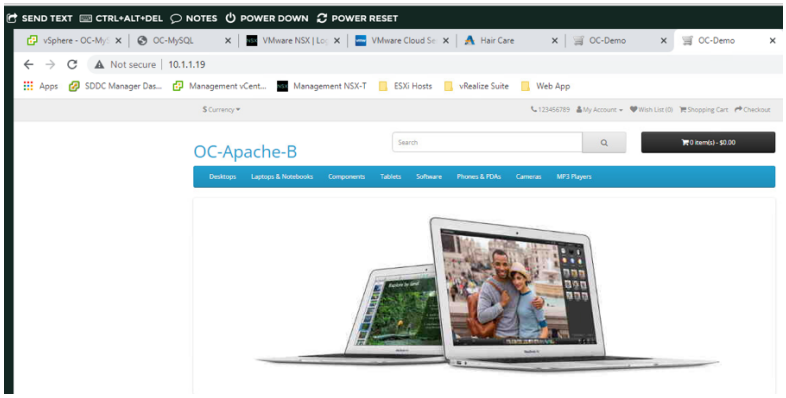
A. Open a new tab in the Chrome browser
B. Connect to OC-Apache-A **10.1.1.18**
C. You should see the OC-Apache-A web page (The webservers for this lab module were modified to show the name of the host you are connecting to for clarity)



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.



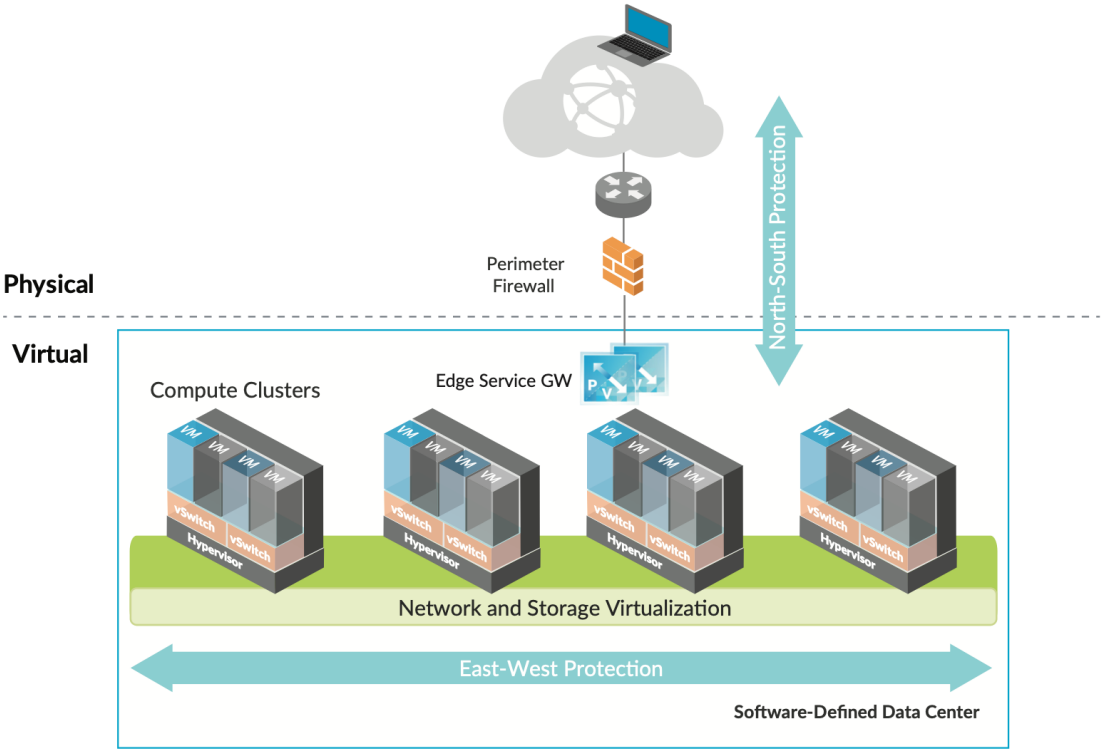
E. Repeat steps A-D for OC-Apache-B 10.1.1.19



[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.

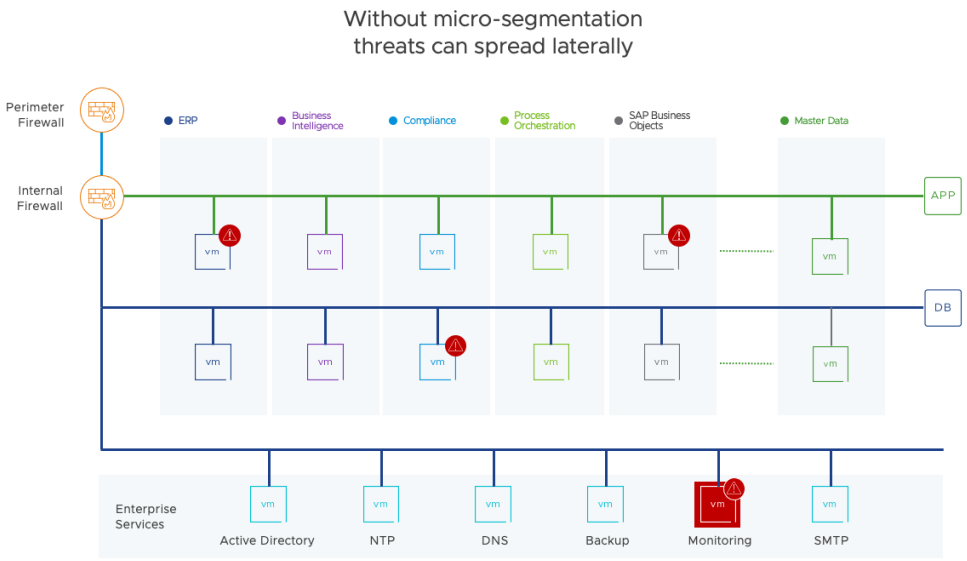
Module 2: Changing the Security Game with Microsegmentation



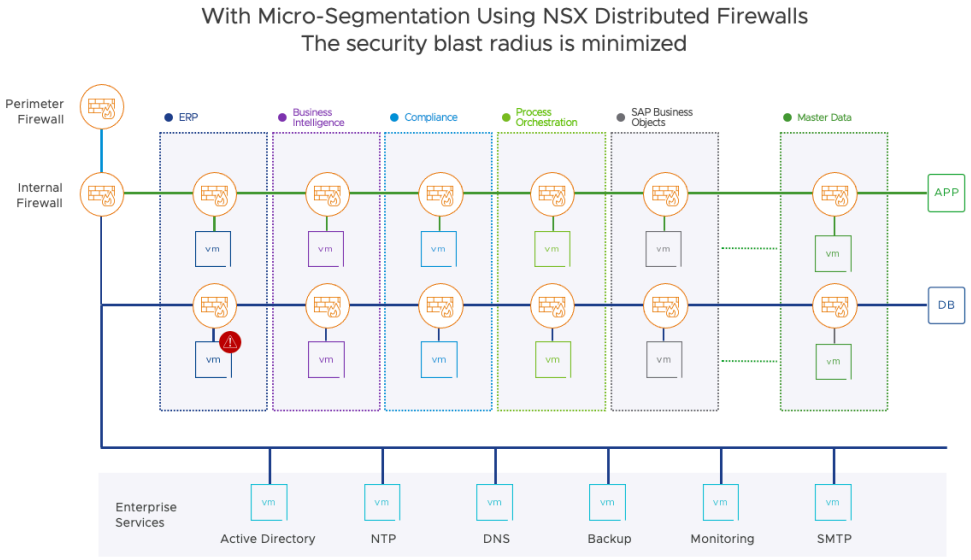
Over the past 10+ years traffic in a datacenter has changed. More and more traffic stays within the datacenter, moving between distributed application components. This traffic, known as “East-West”, is

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:

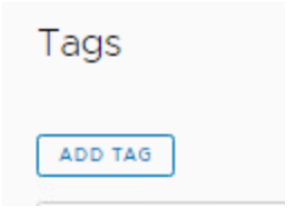
- This is a small sample 2-tier application. For applications leveraging micro-services, you'll be able to group more than one machine under one tag, and better leverage the security groups
- The advantage of using tags/groups is also an operational one. Once you create your infrastructure around Security 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

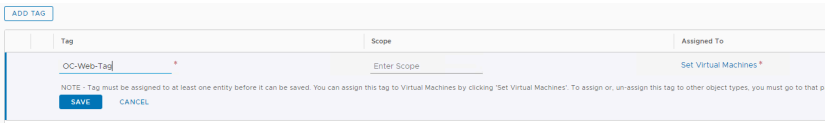
VM	Tag	Security Group
OC-MySQL 10.1.1.50	OC-DB-Tag	OC-DB-Group
OC-Apache-A 10.1.1.18	OC-Web-Tag	OC-Web-group
OC-Apache-B 10.1.1.19	*OC-Web-Tag*	*OC-Web-Group*

[Step 1] Create Tags

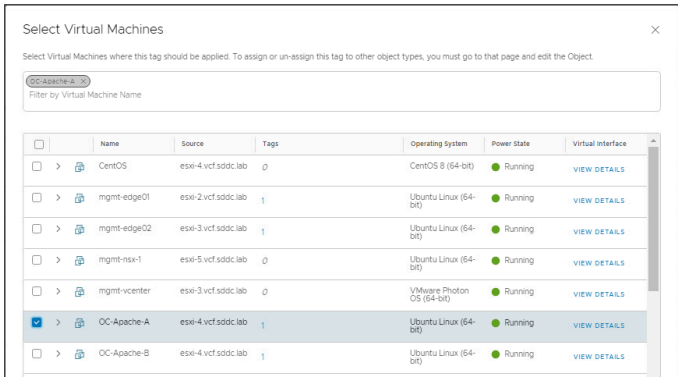
A. Open a new tab in the Chrome browser
B. Click the Management NSX-T shortcut in the bookmark bar (click advanced / proceed to nsx-mgmt.vcf.sddc.lab, if required to accept the certificate)
C. Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
D. Navigate to **Inventory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



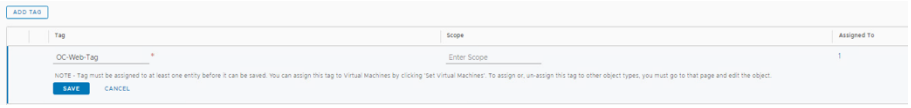
F. We will first create a new tag for the web servers. Add the name **OC-Web-Tag** in the Tag field. Don't hit save yet. As you can see, the note says the tag must be assigned to at least one object first.



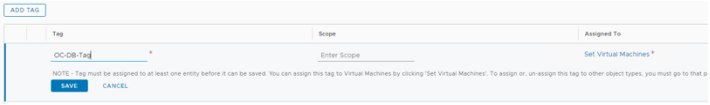
G. Assign this tag to the OC-Apache-A virtual machine. Click on the **Set Virtual Machines** link field and select the **OC-Apache-A** virtual machine (you may need to scroll). For now, do not select OC-Apache-B as it will added later



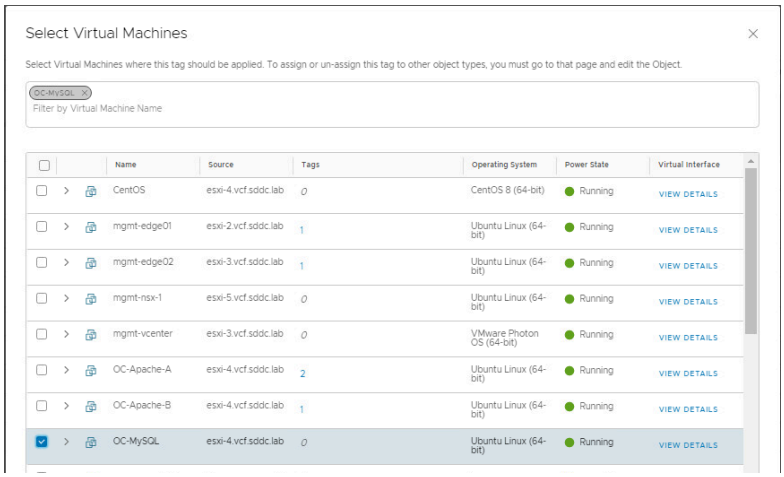
H. Click **APPLY**
I. Note the “Assigned To” value has incremented to 1
J. Click **SAVE** to save the new tag



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



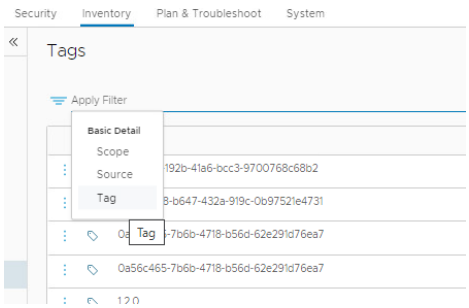
M. Click Set Virtual Machines then select OC-MySQL



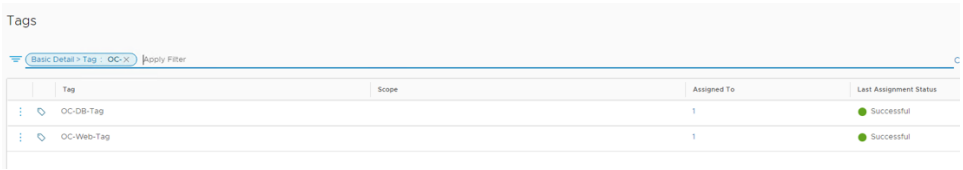
N. Click **Apply**
O. Click **Save**

Step 1A: Verify Tags

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

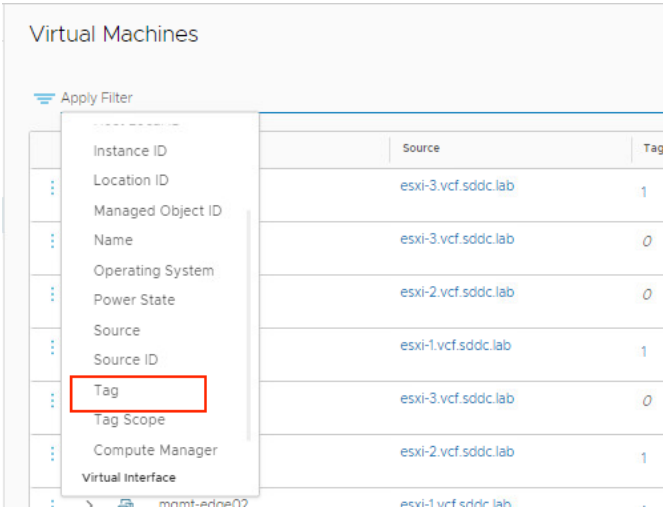


B. Search for Tags with the string "**OC**" in the name

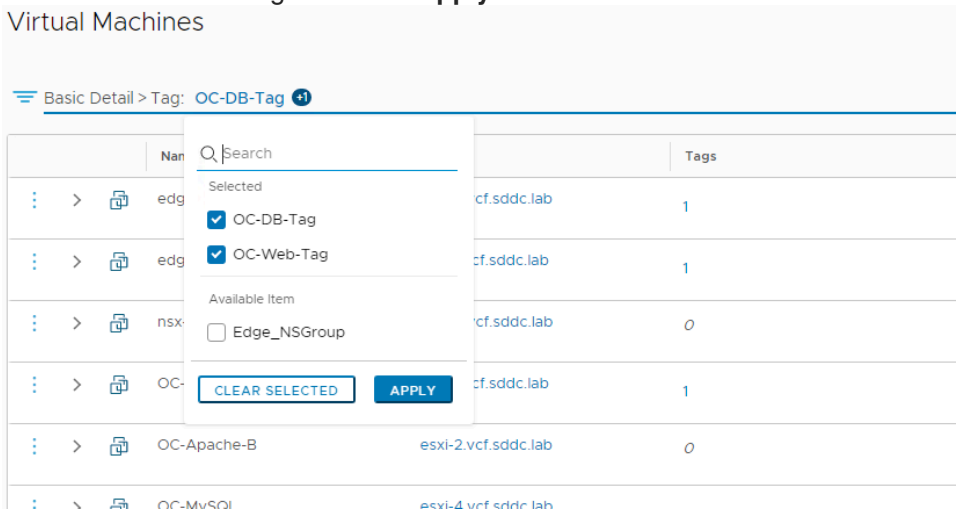


Step 1B: Verify virtual machines are mapped to tags.

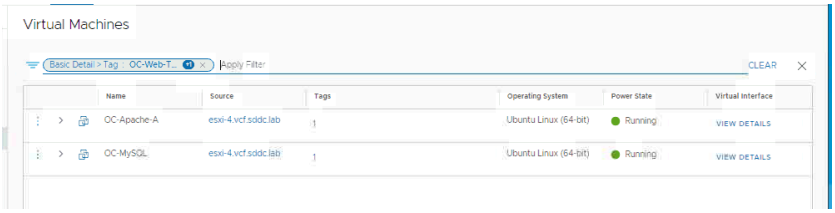
A. Select **Inventory** > **Virtual Machines** and click in the **Filter** area
B. Scroll down in Basic Detail and select **Tag**



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



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



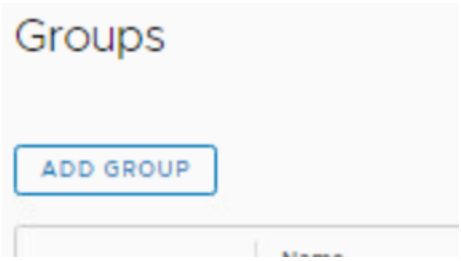
Step 2: Create Groups

Group mapping is as follows

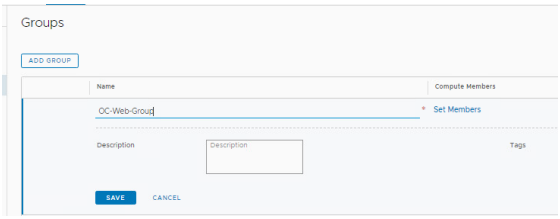
Tag	Security Group
OC-DB-Tag	OC-DB-Group
OC-Web-Tag	OC-Web-Group

Step 2.1: Create OC-Web-Group

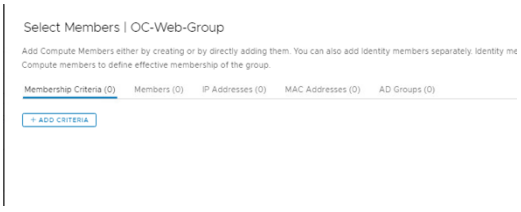
A. Click Inventory > **Groups** > **ADD GROUP**



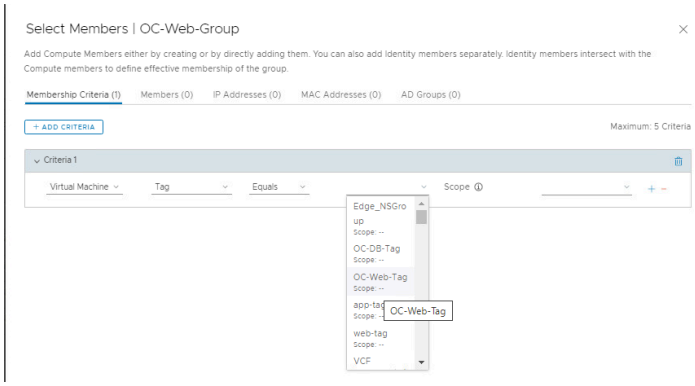
B. Add group name **OC-Web-Group**



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



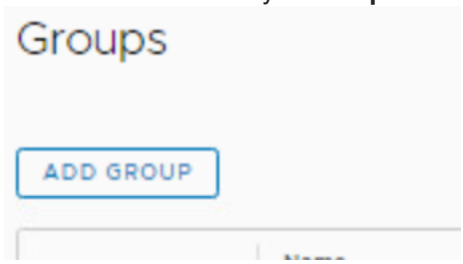
D. Click **Add Criteria**
E. Select **Virtual Machine**, that filters on **Tag Equals** "OC-Web-Tag"



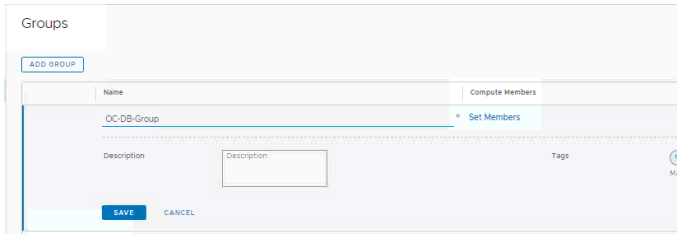
F. Click **Apply**
G. Click **Save**

Step 2.2: Create OC-DB-Group

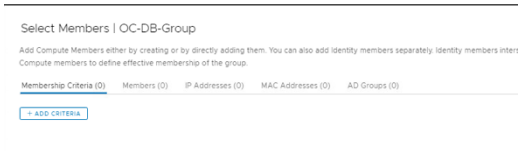
A. Click Inventory > **Groups** > **ADD GROUP**



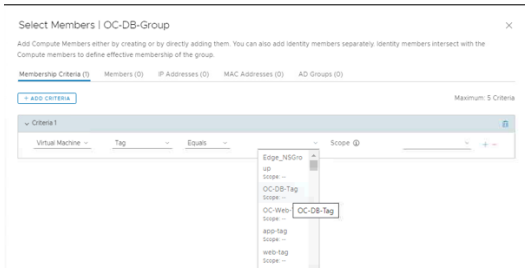
B. Add group name **OC-DB-Group**



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



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

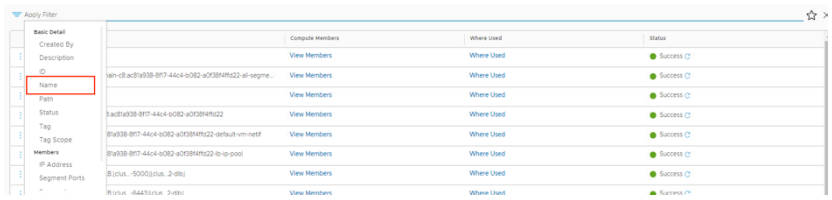


F. Click **Apply**
G. Click **Save**

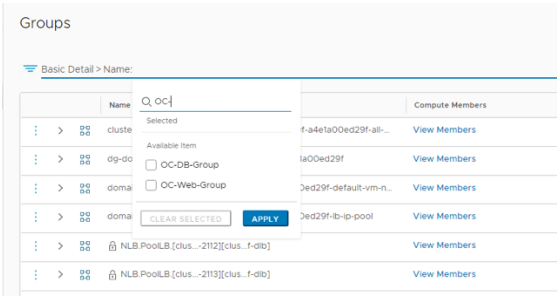
Step 2.1: Verify Groups

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

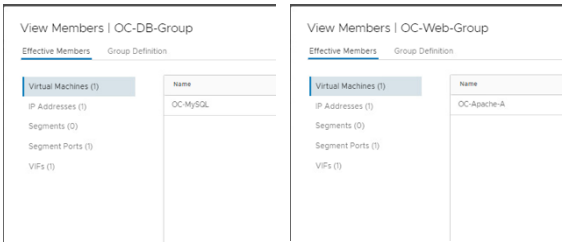
B. Click on Name in the Basic Detail column



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

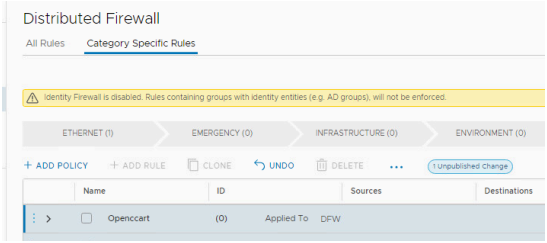
This lab will show configuring the distributed firewall to limit access in our Opencart Application. For the purposes of this lab, we will create the following rules.

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.

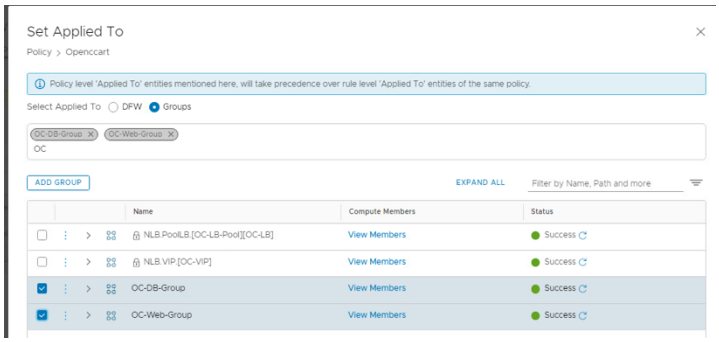
Step 1: Create New Policy

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.)
D. Log into NSX Manager as user: **admin** with the password: **VMware123!VMware123!**
E. Navigate to **Security > Distributed Firewall** in the NSX-T Console
F. Click on **Add Policy** > **New policy**, Name the policy "**Opencart**"



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

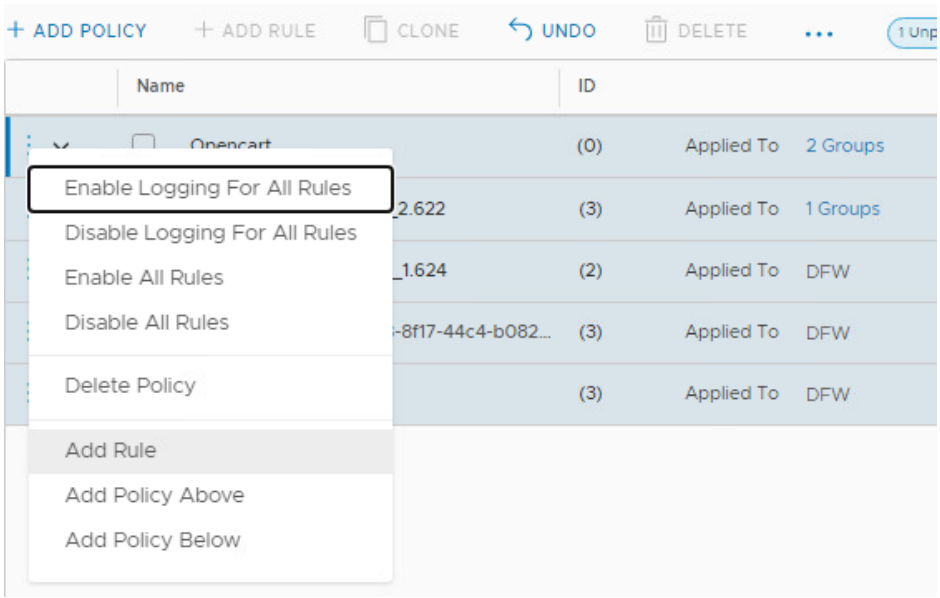
J. Search for OC then select OC-DB-Group and OC-Web-Group



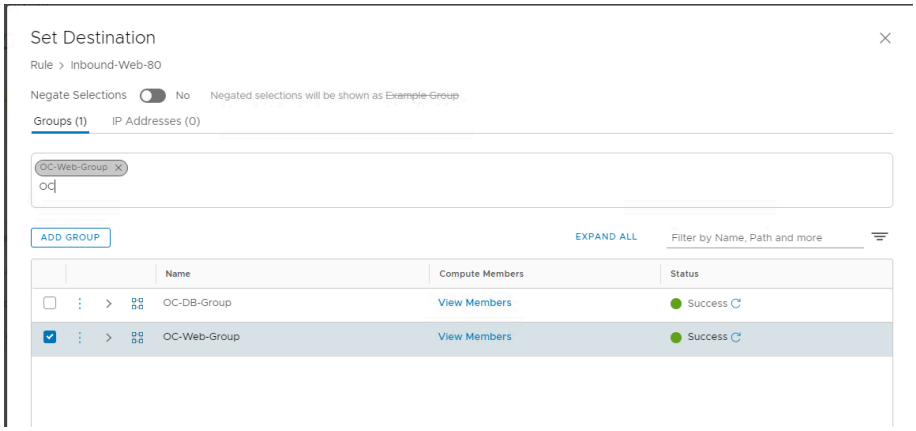
K. Click **Apply**

Step 2: Add inbound-web-80 rule

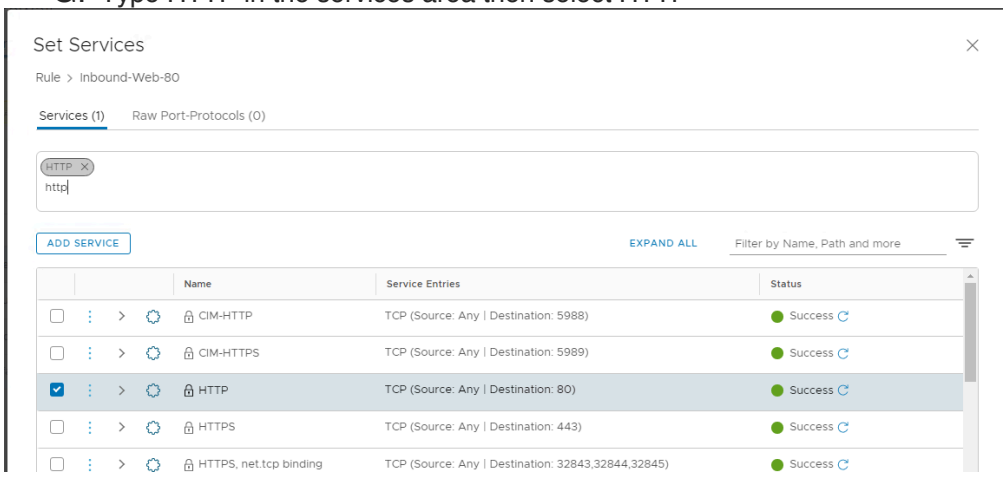
A. Click on the three vertical dots on the left of the **Opencart** policy.
B. Click **Add Rule**



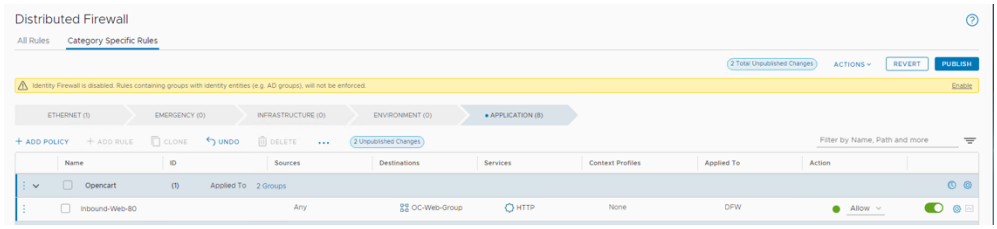
C. Click on **Add Rule** and change name to **inbound-web-80**
D. Hover over **Destinations**, then click the pencil icon
E. This brings up the Set Destination pop-up. Type OC in the search box then click on **OC-Web-Group** checkbox, then **apply**



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

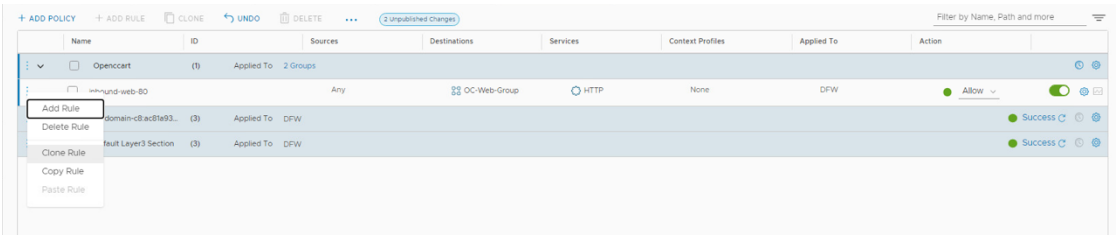


H. Click Apply
I. The result should be the following:

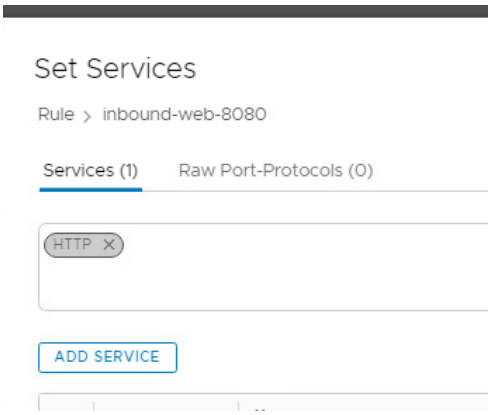


Step 2.1: Add inbound-web-8080 rule (clone inbound-web-80)

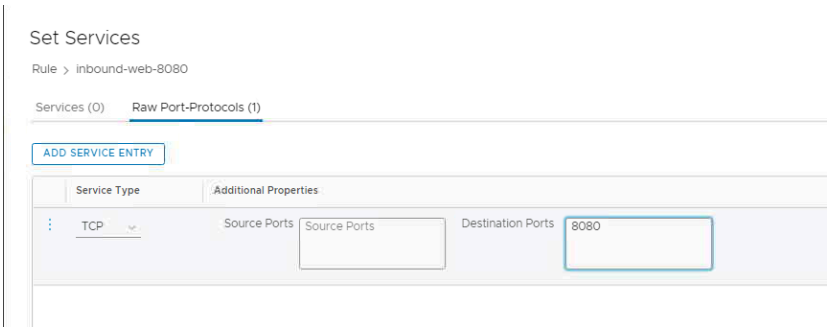
A. Click on the three vertical dots on the left of the **inbound-web-80** rule.
B. Click **Clone Rule**



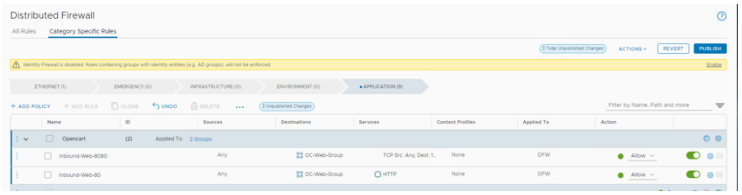
C. Click on **Copy of inbound-web-80** and change name to **inbound-web-8080**
D. Hover over **Services**, then click the pencil icon
E. Click the X on HTTP to delete the service



F. Click on **Raw Port Protocols**
G. Click **Add Service Entry**
H. Set Service Type **TCP** and Destinations Ports **8080**

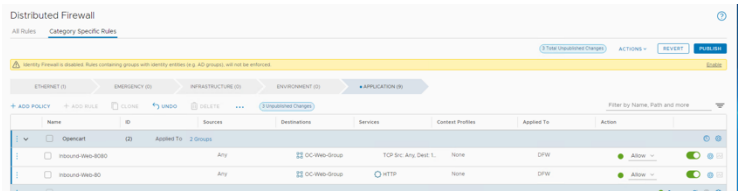


I. Click Apply
J. The result should be the following:

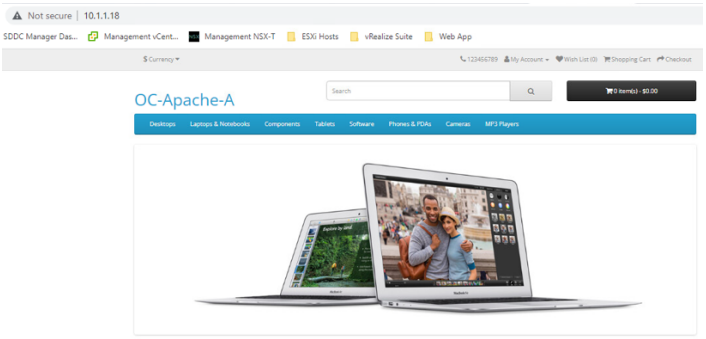


Step 3: Test inbound-web-XX rules

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

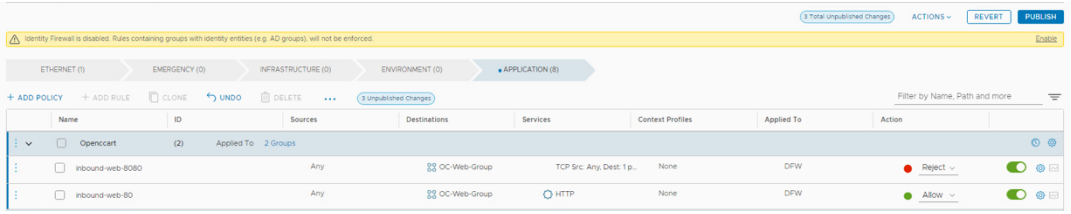


B. Open a tab on the browser and connect to OC-Apache-A 10.1.1.18



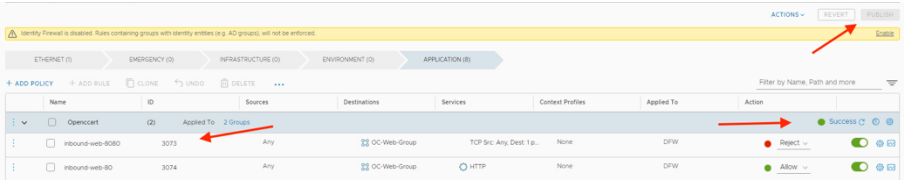
C. Open a second tab and connect to OC-Apache-A 10.1.1.18:8080

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

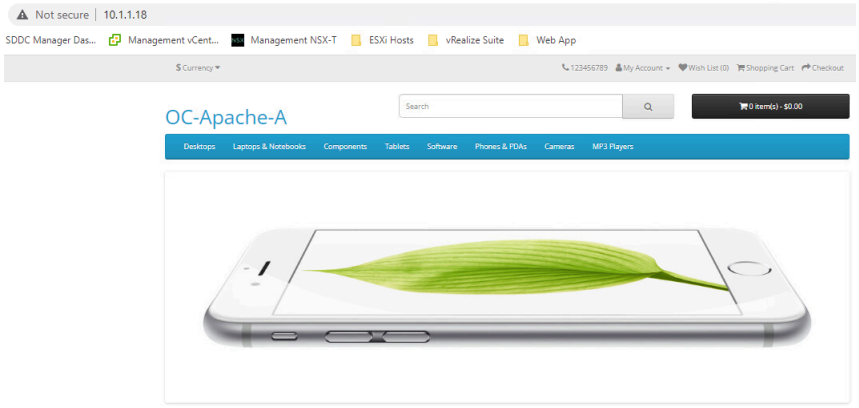


E. Click **PUBLISH** to make the rules active

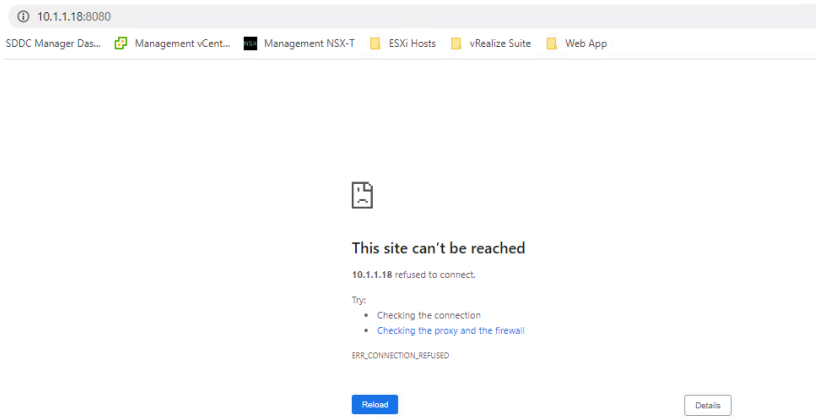
F. Notice that the Publish button is greyed out, showing there are no uncommitted changes. The green Success indicator is set at the policy level, and our two rules now have ID numbers showing they have been activated.



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

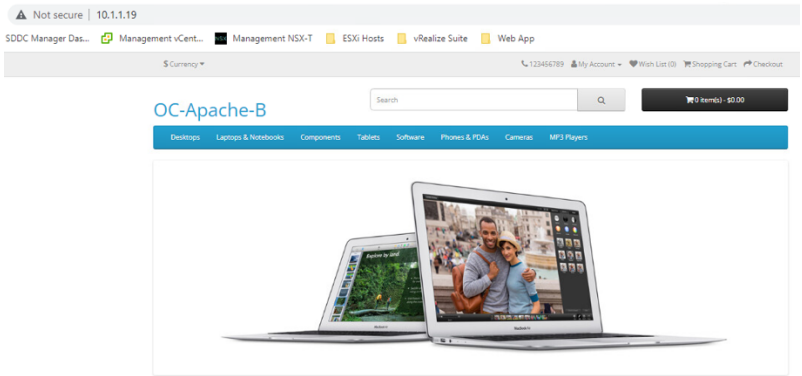


H. Go to the OC-Apache-A 10.1.1.18:8080 browser tab and refresh. The web page should fail to load

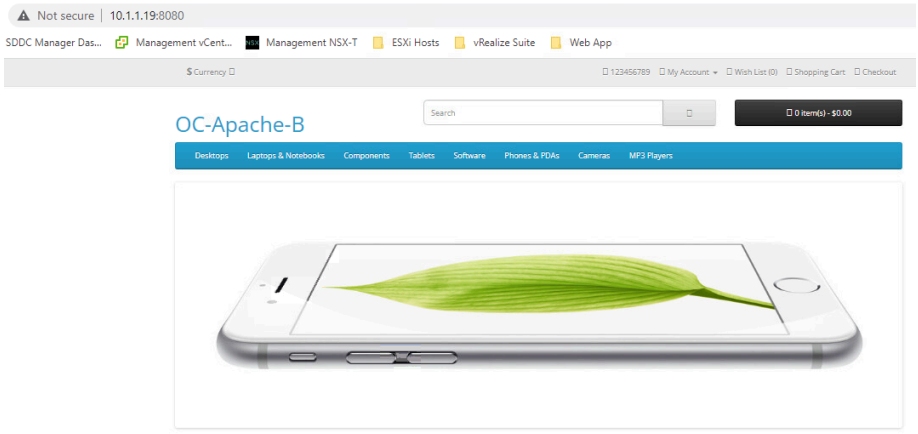


Step 4: Extend security policies to new VM based on tag

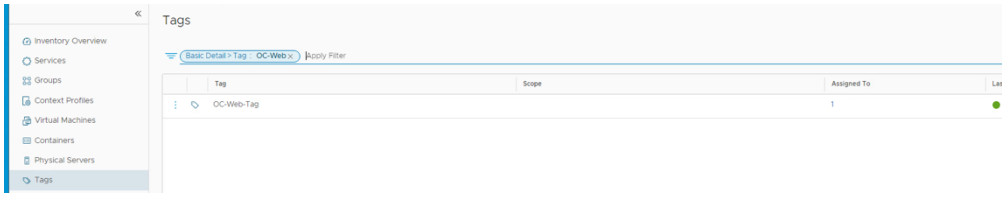
A. Test operations of the OC-Apache-B 10.1.1.19 webserver on port 80



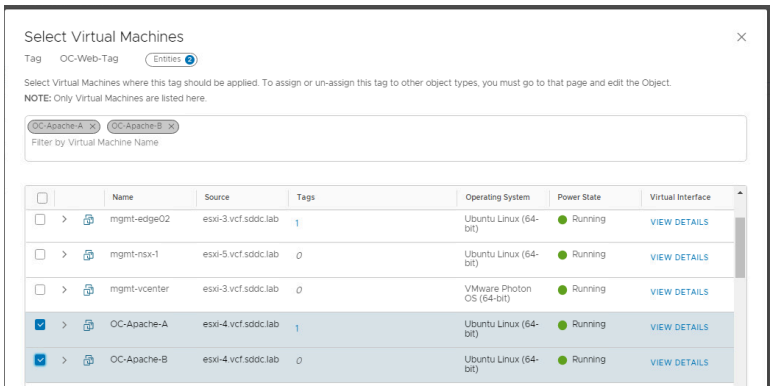
B. Test operations of the OC-Apache-B 10.1.1.19 webserver on port 8080



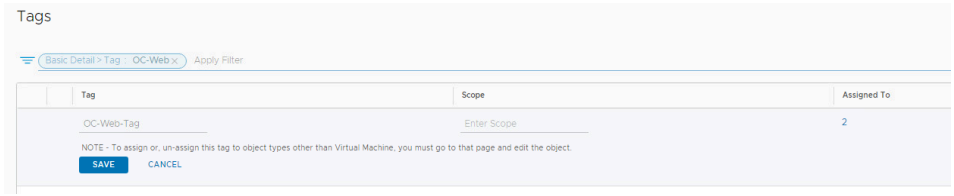
C. Observe that the Reject rule on port 8080 does not extend to OC-Apache-B.
D. Return to the NSX tab and click on **Inventory** then **Tags**
E. Click on Filter, select Tag and search for by Name and type OC-Web



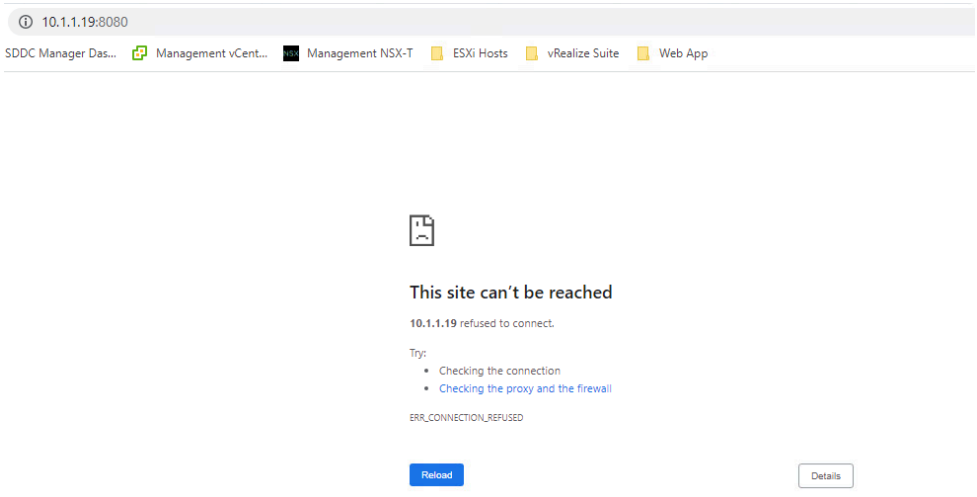
F. Click the 3 dots next to OC-Web-Tag, then click Edit
G. Click on the number 1 under Assigned To
H. Search for OC-Apache and select OC-Apache-B then click Apply



I. Click Save



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

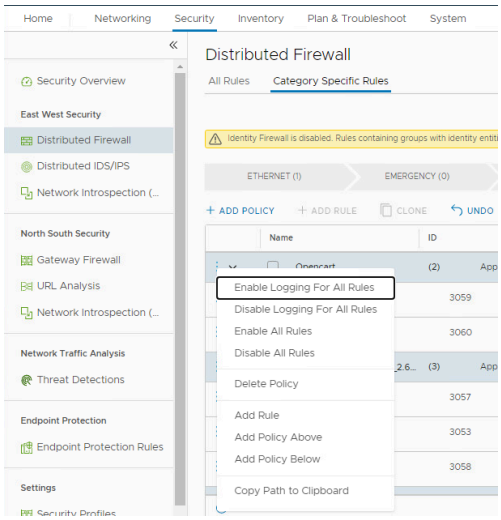


K. Notice that as soon as the tag was applied to OC-Apache-B VM, it immediately became a part of the Opencart Policy, because it became a member of the web-group that the rules applied to

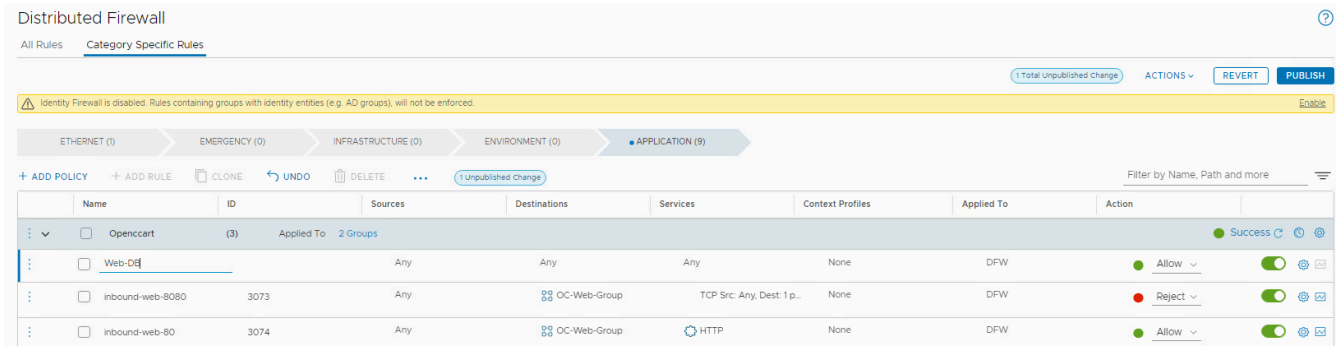
Step 5: Implement Web-DB rule

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

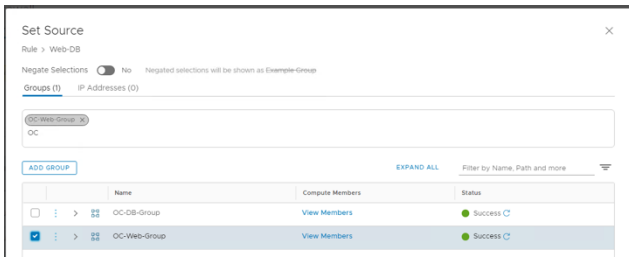
A. On the NSX Manager tab, go to Security -> Distributed Firewall
B. Click on the 3 dots next to Opencart and **Add Rule**



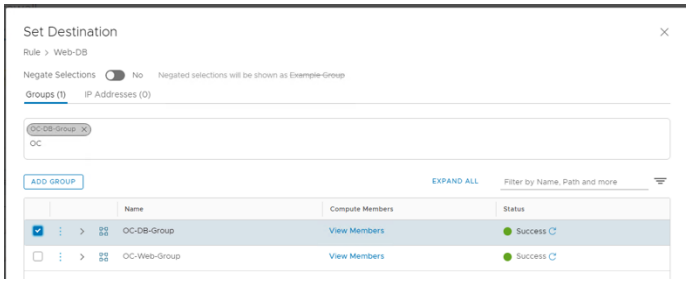
C. Name the rule **Web-DB**.



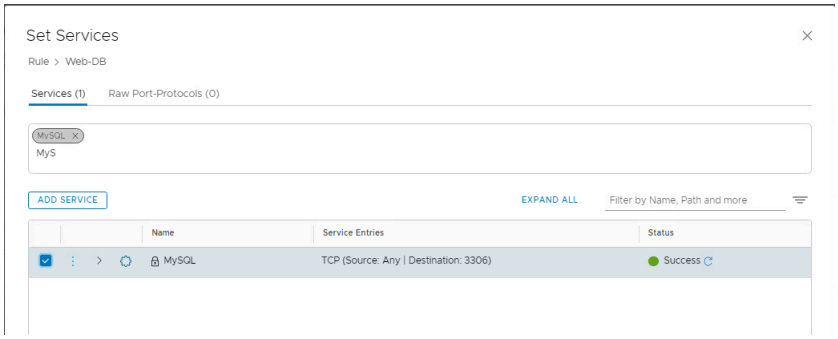
D. Set Sources to OC-Web-Group and click Apply



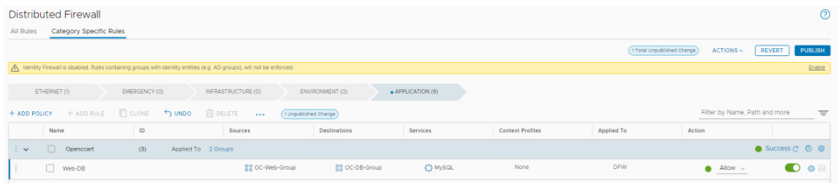
E. Set Destinations to OC-DB-Group and click Apply



F. Set Services to MySQL and click Apply

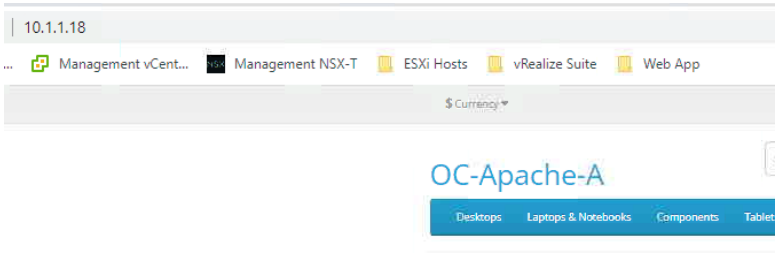


G. Your rule should look like this

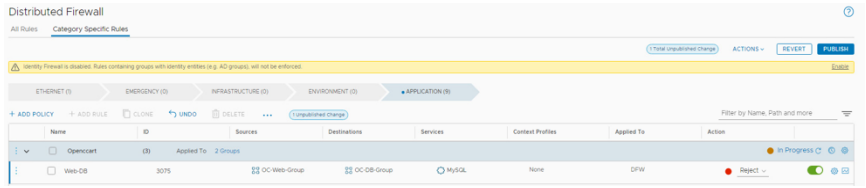


H. Click Publish

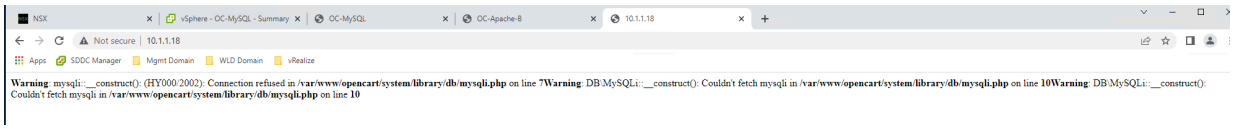
I. Test access to OC-Apache-A on 10.1.1.18. You should get a normal web page load



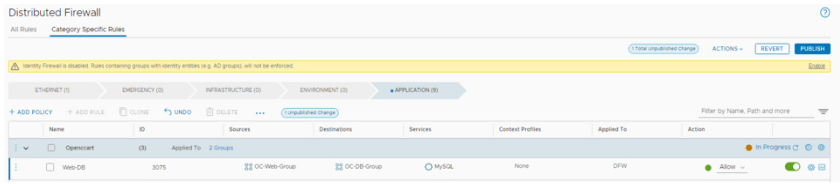
J. Return to the NSX tab and set the web-app rule to Reject and click Publish (this will allow us to see the impact of the firewall blocking access from Apache to MySQL which will be useful later in the lab)



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



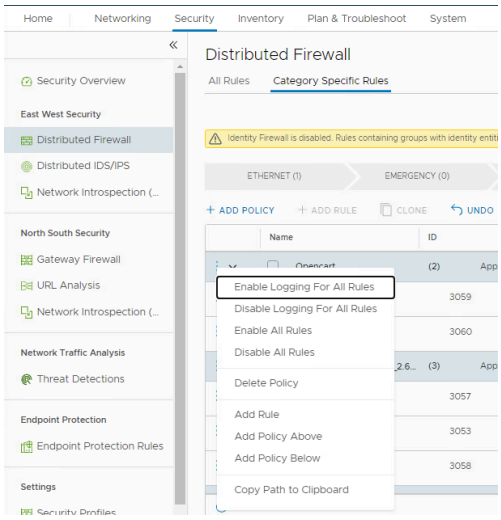
L. Reset the web-app rule to Allow and Publish before moving on to the next step



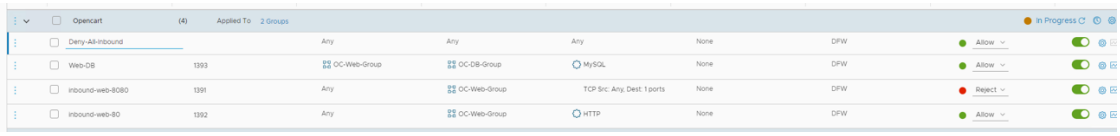
Step 5: Implement Deny All Inbound rule

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

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

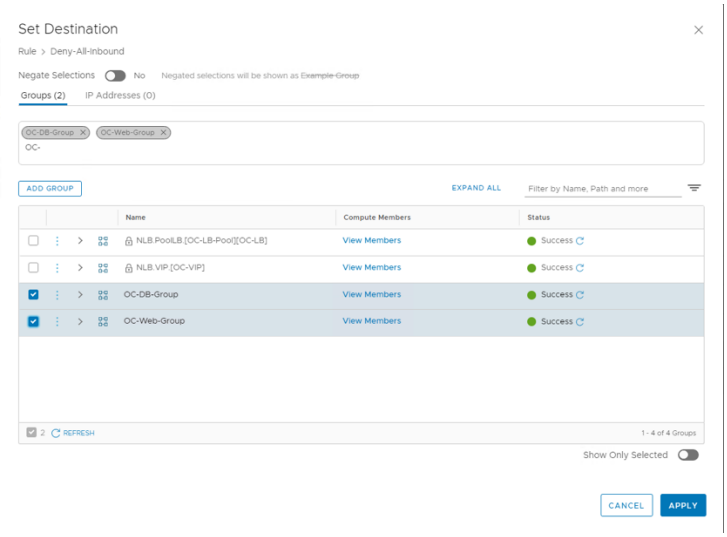


C. Name the rule Deny-All-Inbound.

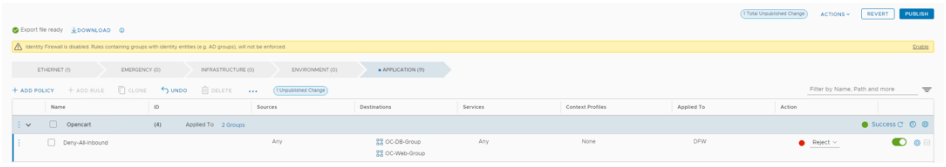


D. Leave Sources at Any

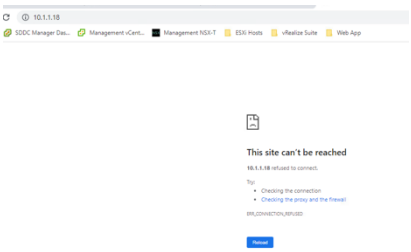
E. Set Destinations to groups OC-DB-Group and OC-Web-Group (Hint: Type OC- and enter to quickly show OC groups) Check the two groups and then Apply



F. Set Action to Reject, then click Publish

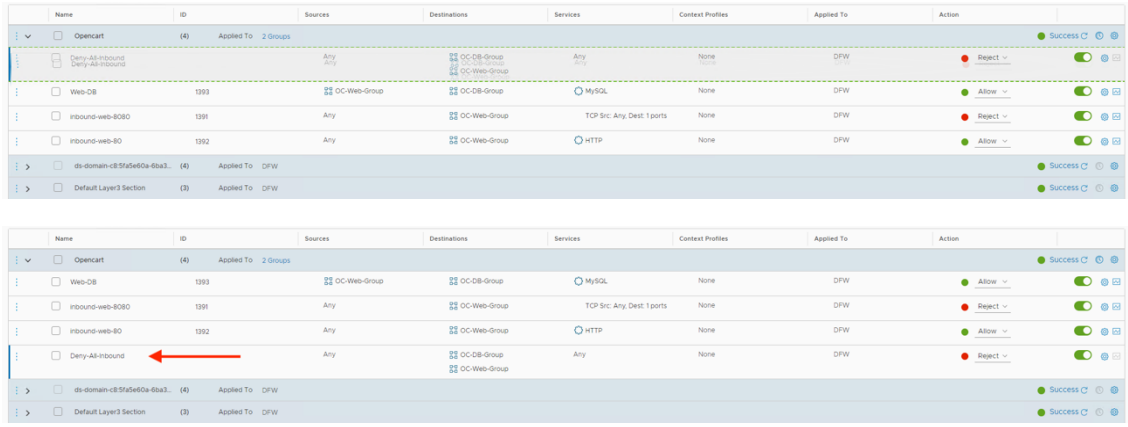


G. Test access to OC-Apache-A on 10.1.1.18. Your web page load should fail. This is because the Deny-All rule is evaluated prior to our Allow rules

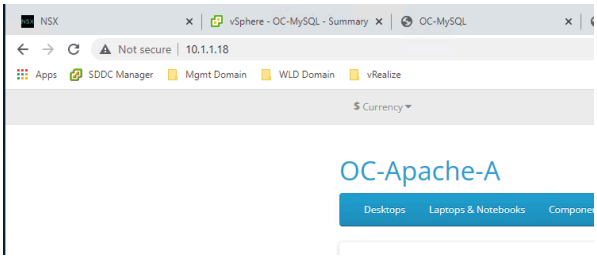


H. Return to the NSX tab

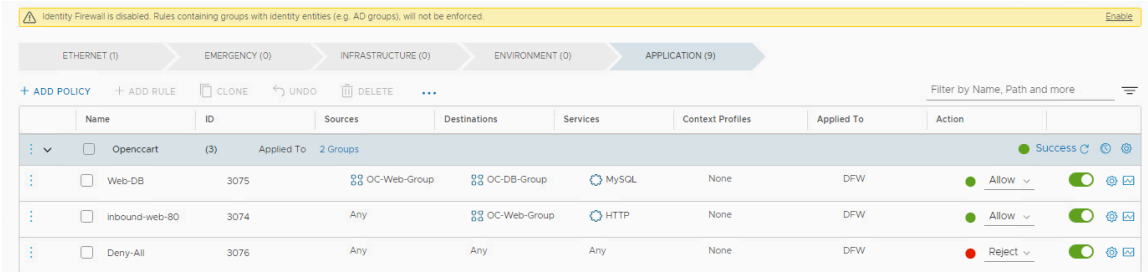
I. Move the deny-all rule down by clicking the mouse and holding down with the cursor anywhere on the Deny-All rule line and dragging the rule to below our inbound-web-80 rule then clicking Publish



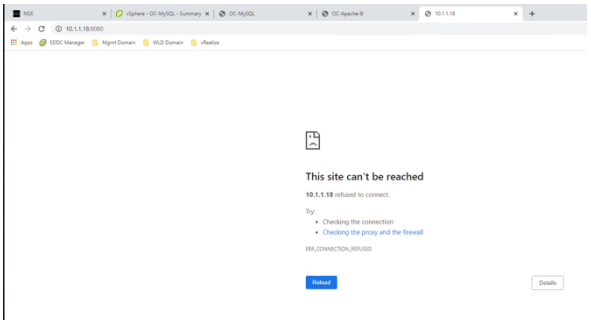
J. Test access to OC-Apache-A on 10.1.1.18. You should get a normal web page load



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



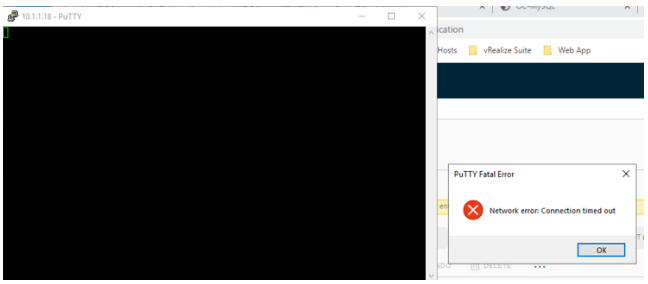
N. Go to the OC-Apache-A 10.1.1.18:8080 The web page should fail to load.



Step 5: Implement ssh rule

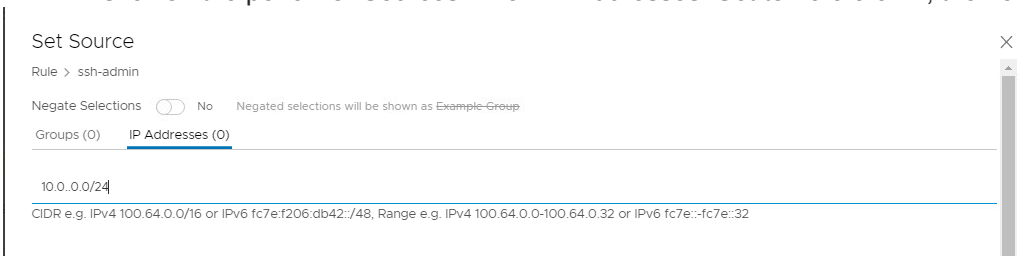
The step will implement a rule to allow ssh connections to our Apache and MySQL VM's, but only from our inside admin network (10.0.0.0/24)

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

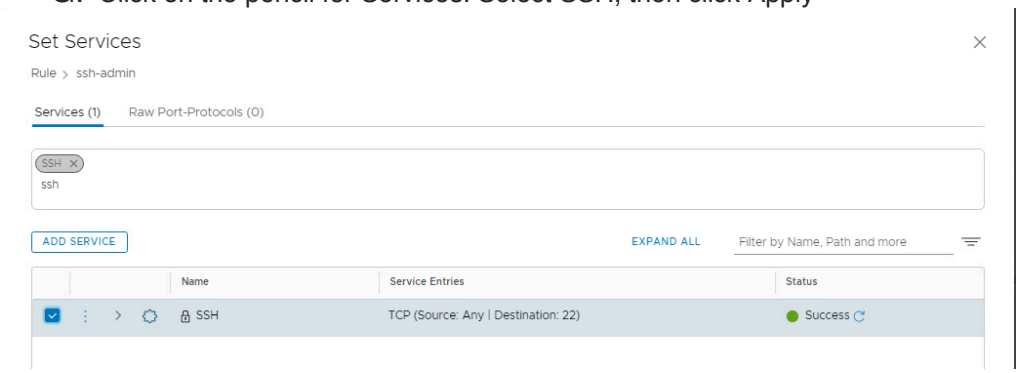


C. Return to the NSX tab
D. Click on the tree dots to the left of the Opencart and Add Rule
E. Name the rule ssh-admin

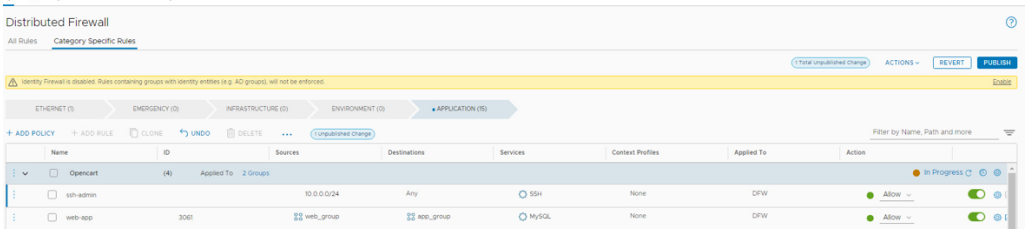
F. Click on the pencil for Sources. Then IP Addresses. Set to 10.0.0.0/24, then click Apply



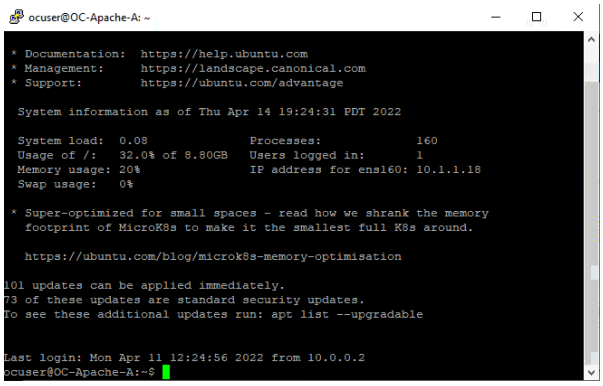
G. Click on the pencil for Services. Select SSH, then click Apply



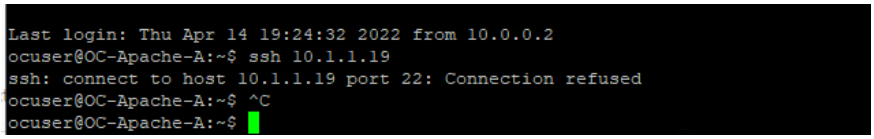
H. Publish your new rule



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!



L. Attempt to ssh from OC-Apache-A to OC-Apache-B 10.1.1.19
M. Your connection should be refused

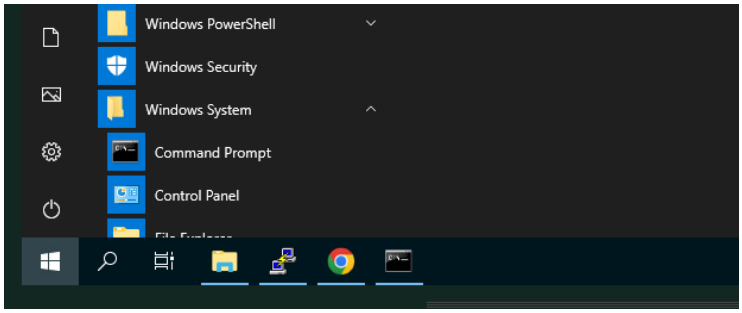


N. Observe that we have blocked ssh within the Web servers but allow between our admin network and the web servers.

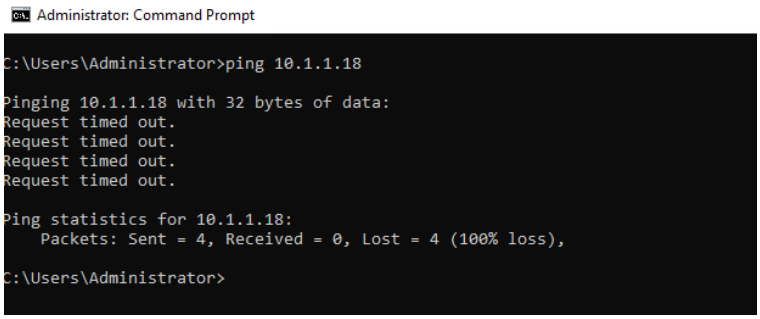
Step 5: Implement ICMP-Admin rule

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

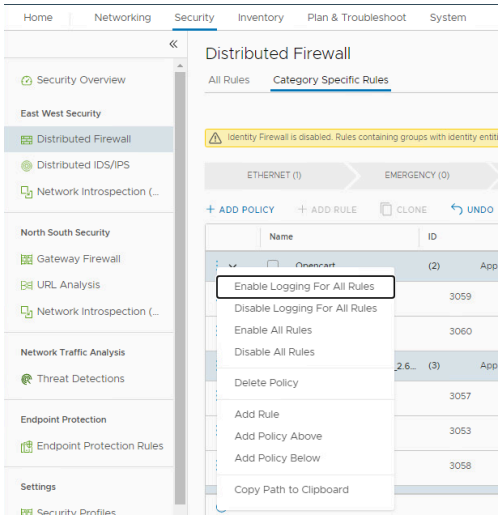


B. Attempt to ping to OC-Apache-A 10.1.1.18. Your connection should time out

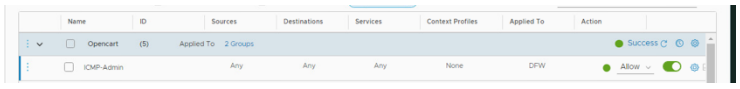


C. On the NSX Manager tab, go to Security -> Distributed Firewall

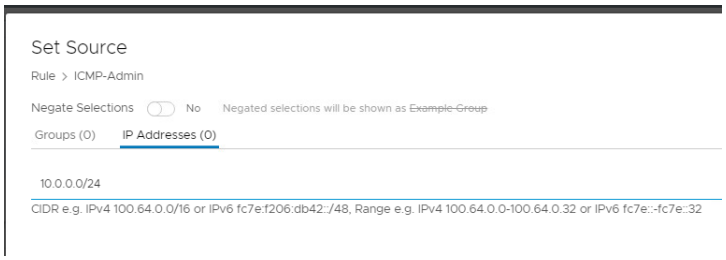
D. Click on the 3 dots next to Opencart and Add New Rule



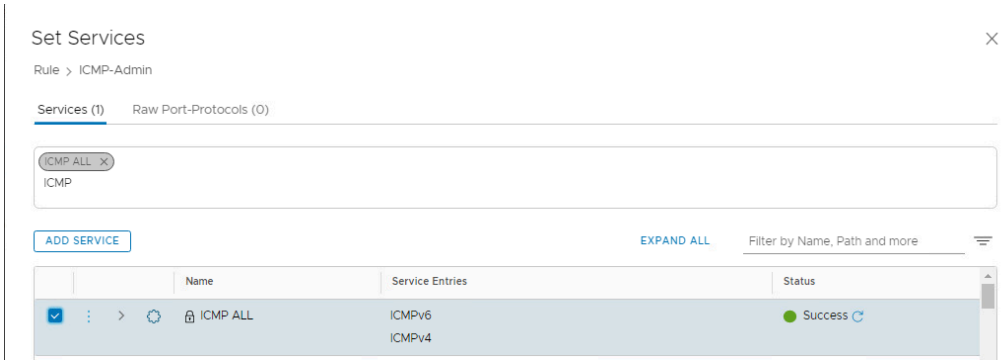
E. Name the rule ICMP-Admin



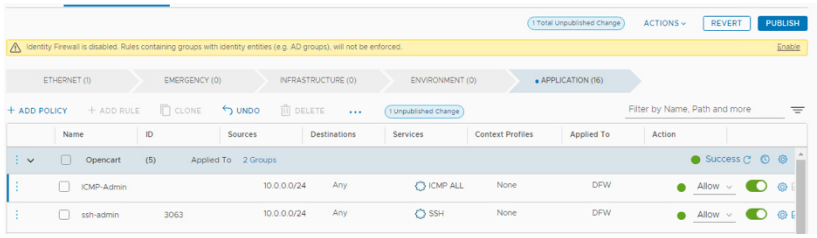
F. Click on the pencil for Sources. Then IP Addresses. Set to 10.0.0.0/24, then click Apply



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



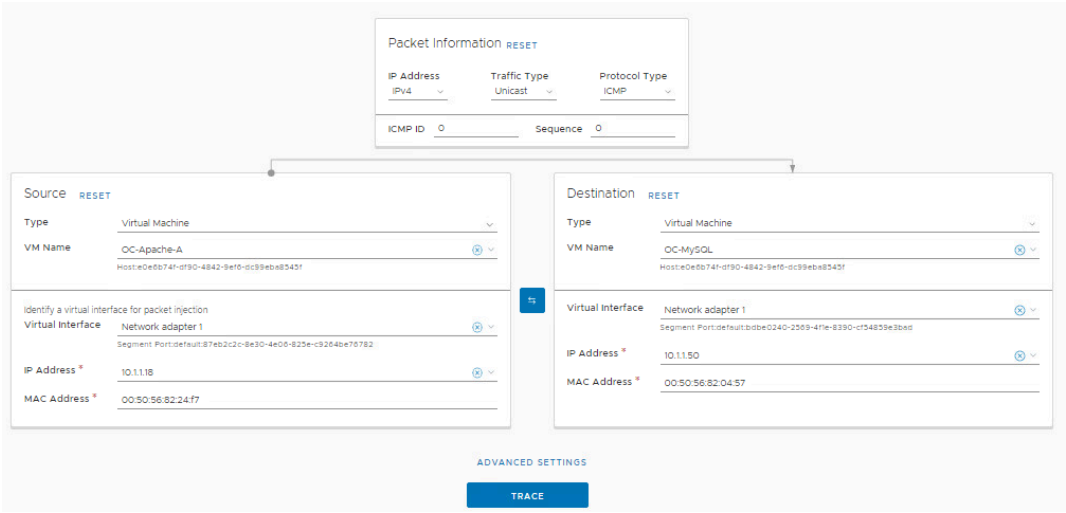
H. Publish your new rule



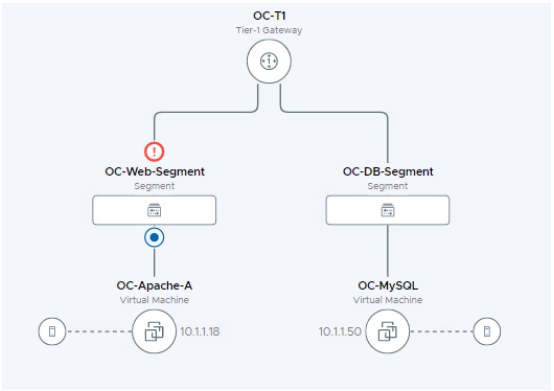
I. Return to your Windows command prompt
J. Attempt to ping to OC-Apache-A 10.1.1.18. Your connection should succeed

```
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
```

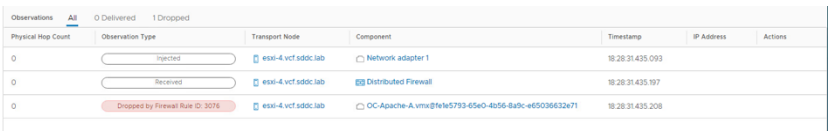
K. From the NSX-T Manager interface click the **Plan & Troubleshoot** tab
L. Click **Traceflow** in the left navigation panel
M. Configure Traceflow from OC-Apache-A to OC-MySQL and click Trace



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

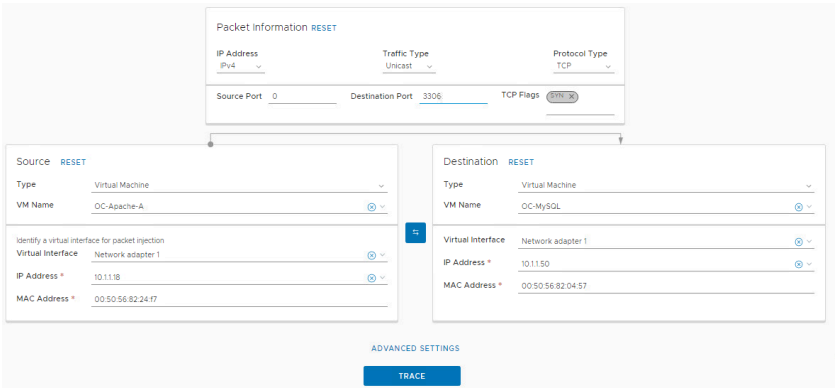


O. Review the observations panel and notice the packet was dropped at OC-Apache A between the VM NIC and OC-Web-Segment

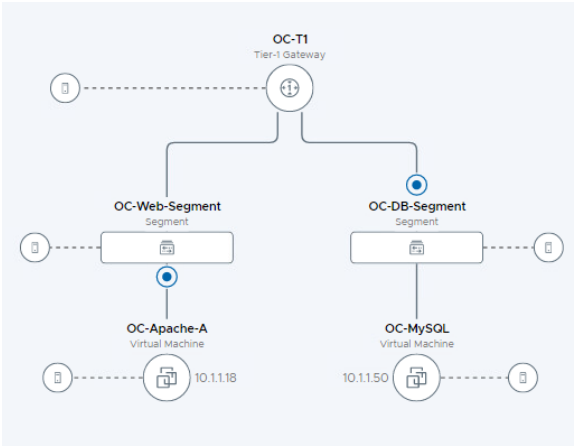


P. On the Traceflow screen, click EDIT to reconfigure the trace then click Proceed on the warning pop up

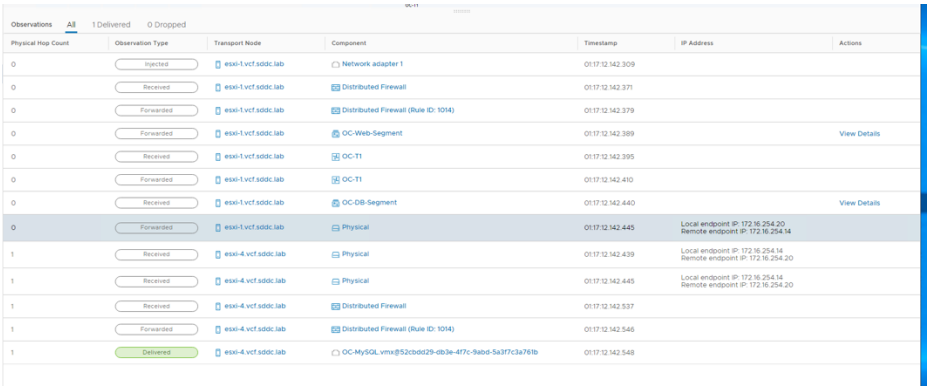
Q. Change Protocol Type from ICMP to TCP, with Destination Port 3306



R. Click Trace. Your Traceflow should succeed



S. In the Observations panel, review the following
- 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-DB-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



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

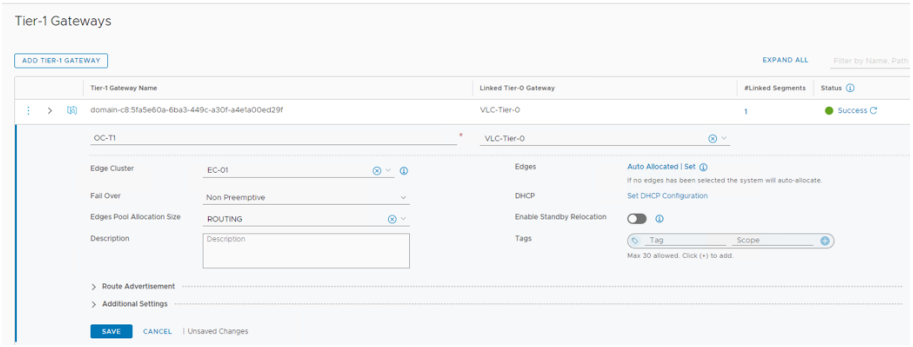
Lab 1: Configure load balancer

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

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 NSX Networking -> Tier-1 Gateways
E. Click the 3 dots next to OC-T1 and select edit
F. Click on Select Edge Cluster and select EC-01

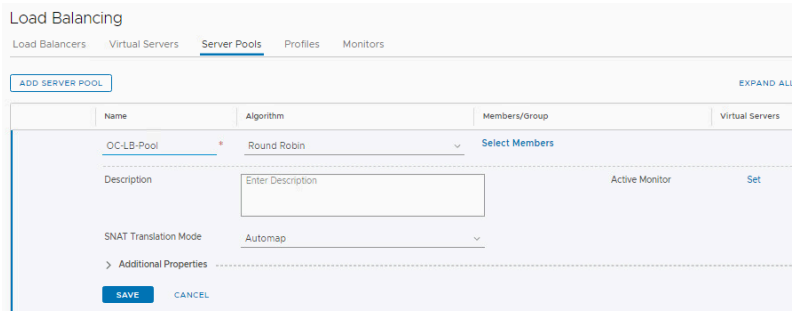


G. Click Save
H. Click Close Editing

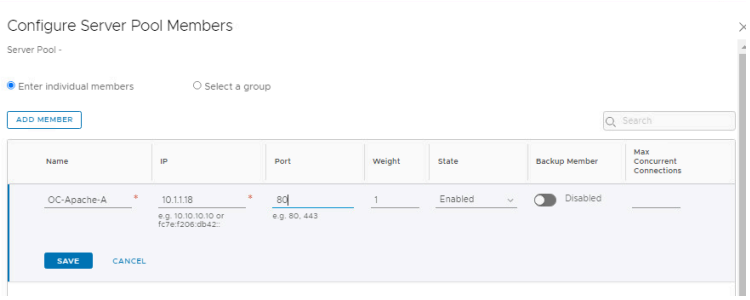
Step 2: Create Server Pool

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

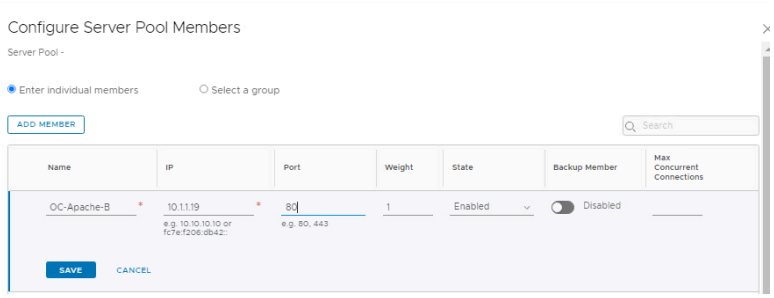
A. Navigate to NSX Networking -> Load Balancing -> Server Pools. Click **Add Server Pool**
B. Name the pool OC-LB-Pool



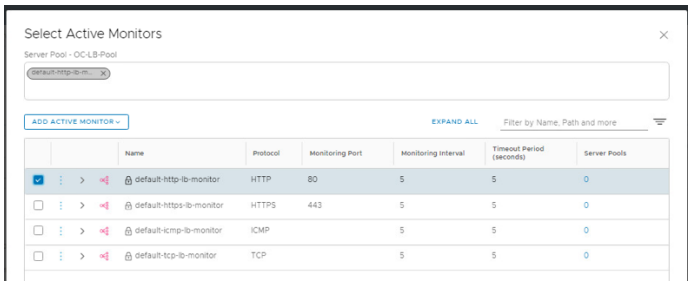
C. Click **Select Members**
D. Click **Add Member**
E. Name **OC-Apache-A**, IP **10.1.1.18** Port 80



F. Click SaveIP I
G. Repeat steps for OC-Apache-B, using:
    a. Name **OC-Apache-B**, IP **10.1.1.19** Port 80



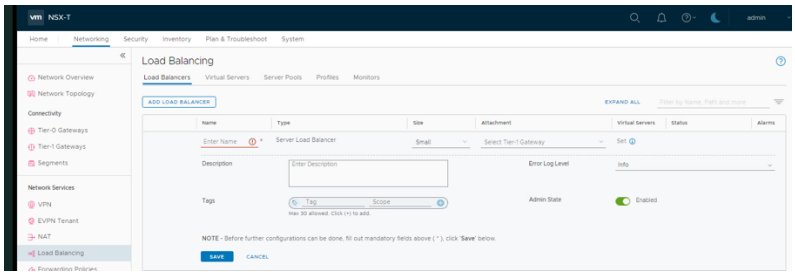
H. Click Set next to Active Monitor
I. Select the default HTTP Port 80 monitor, then click Apply



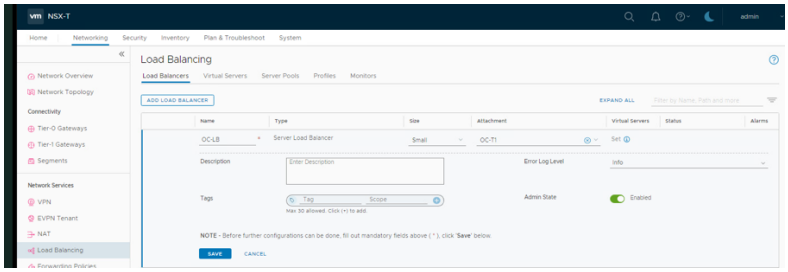
J. Click Save

Step 3: Create Load Balancer

A. Navigate to Networking -> Load Balancing



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

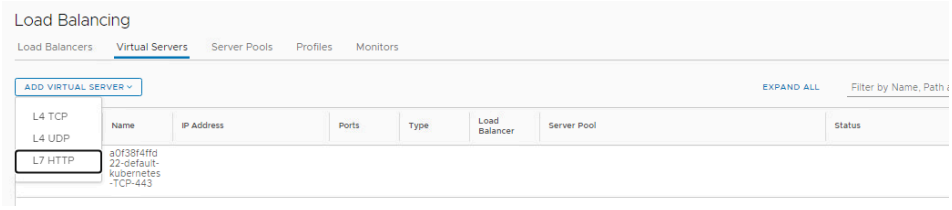


D. Click Save
E. When prompted Want to continue configuring this Load Balancer, select No

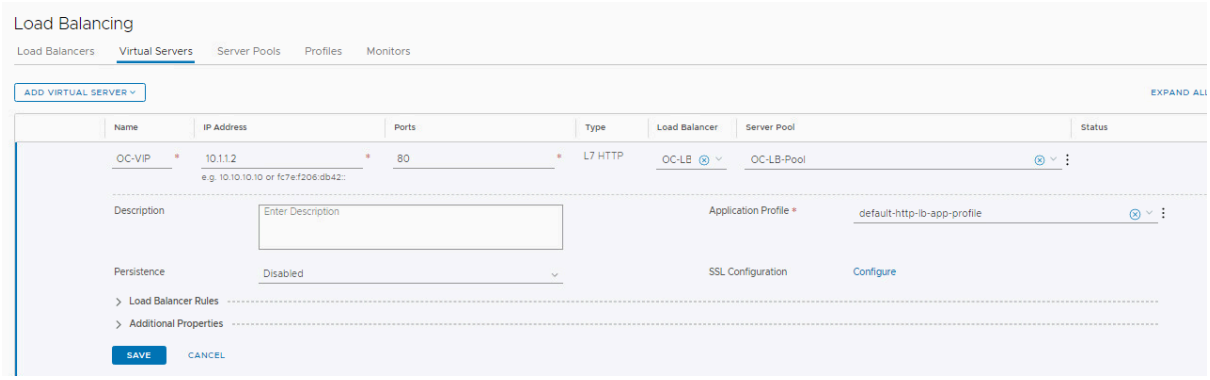
Step 4: Create Virtual Servers

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**

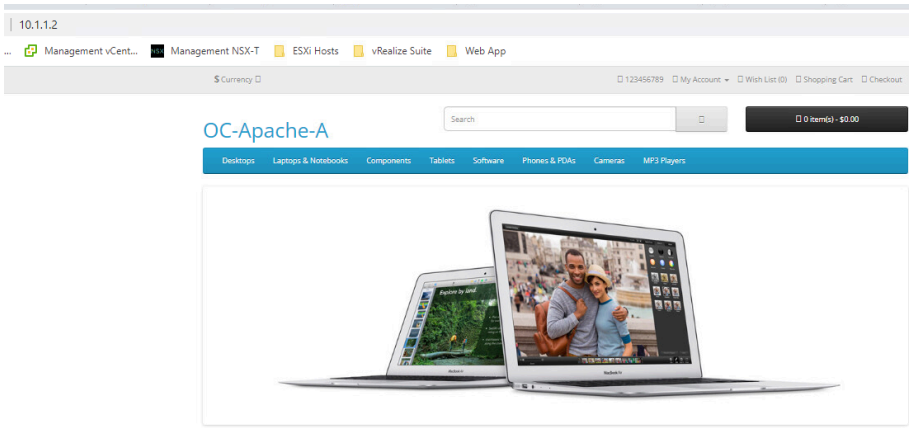


B. Name the Virtual Server **OC-VIP,** IP Address **10.1.1.2,** Port **80**
C. Type OC in the Load Balancer field and it will allow you to select OC-LB
D. Type OC in the Server Pool field and select OC-LB-Pool
E. Click on **SAVE**

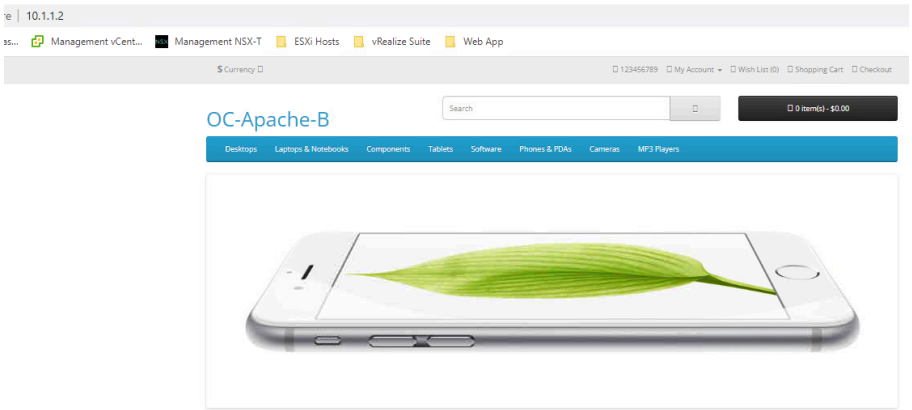


Step 5: Test Load Balancer

A. Open a new tab for 10.1.1.2

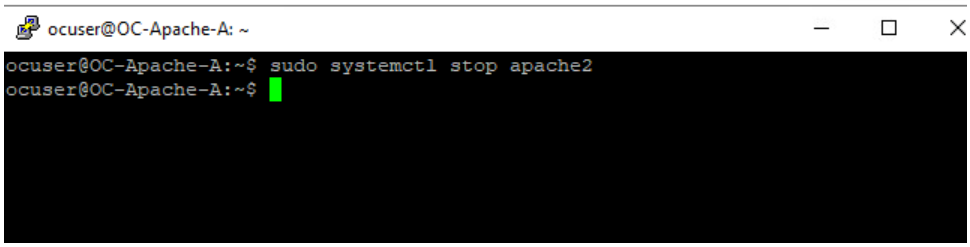


B. Refresh the browser for this tab. You should see the opposite web server



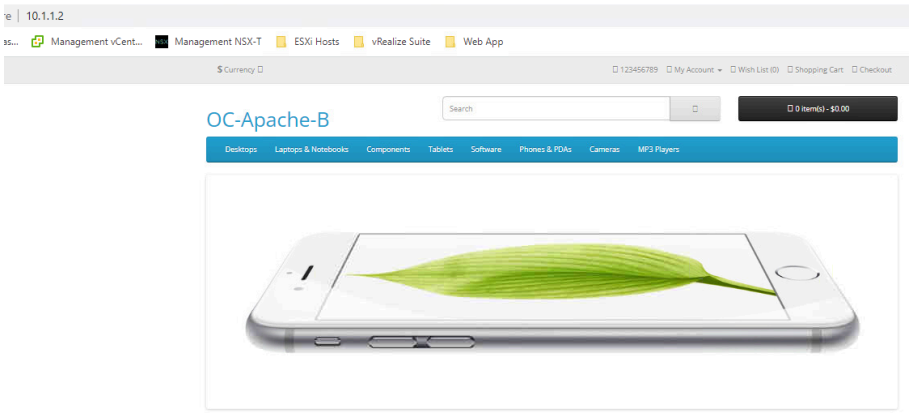
C. Open a PuTTY session to OC-Apache-A, login **ocuser**, password **VMware123!**

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



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