

A Working VXLAN configuration with the CSR1000v under ESXi

Objective

The objective of the configuration below it to have direct L2 communication between two separate nodes / segments where the interconnect is a routed VXLAN. So without any routing in the Overlay network!

Pre-requisites:

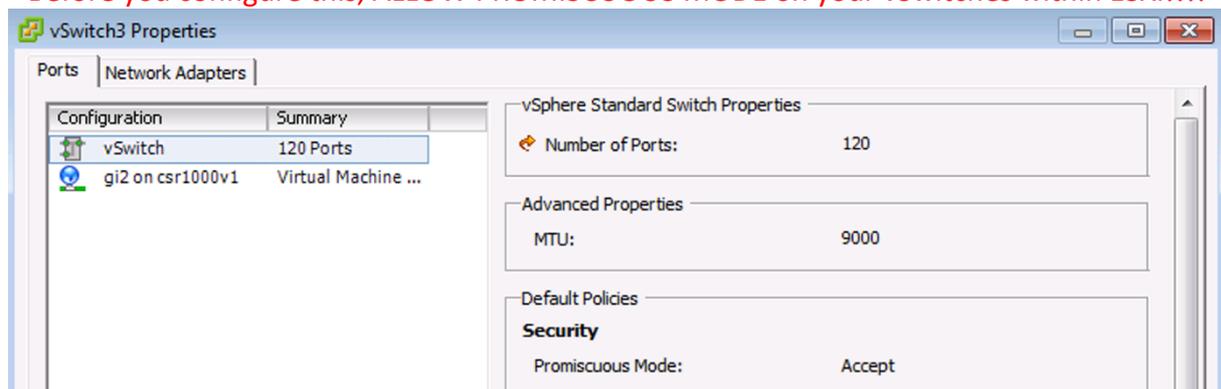
1. A Vmware ESXi server (5.5 used here) with the Cisco CSR routers images and 2 VM's PC1 and 2 "to ping with". For the two CSR routes, about 4GB RAM per VM is sufficient.

2. The version of the CSR Router will have to support VXLANs. If it does NOT recognize the global command "vxlan", then get a more modern image. The version used here was:
Cisco IOS XE Software, Version 16.03.01

3. vSwitches

Very Very Very Very Very Very Very Very Very important:

Before you configure this, ALLOW PROMISCUOUS MODE on your vSwitches within ESXi!!!!



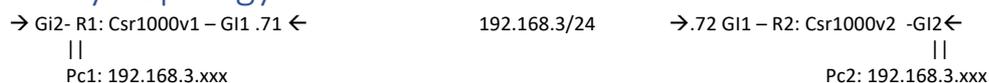
If you do not configure this (only costed me most of a day to troubleshoot.. ☹), then your broadcasts will work ("bum" traffic → is Multicast) but **your unicast traffic will NOT be tunneled**. (drives you spare..)

Credits

For the configuration below i stole from multiple websites and added a tiny bit. The majority of the commands came from this excellent website:

<https://networklessons.com/cisco/ccnp-encor-350-401/vxlan-flood-and-learn-with-multicast>

The Underlay Topology



The way the connections between PC1 and 2 have been isolated, is that within ESXi for the R1 → PC1 connection, a dedicated vSwitch has been created without containing any physical interfaces. (no tagging needed either) The same has been performed for the R1 → Pc2 connection. (see the screenshot above)


```

source-interface lo100
member vni 4096 mcast-group 239.1.1.1

int gi2
description "connected to VM1"
no sh
service instance 100 ethernet
encapsulation untagged

bridge-domain 100
member vni 4096
member GigabitEthernet2 service-instance 100

```

```

R2:
int lo100
ip address 11.11.11.11 255.255.255.255
ip ospf 1 area 0
ip pim sparse-mode

```

```

int nve1
no sh
source-interface lo100
member vni 4096 mcast-group 239.1.1.1

```

```

int gi2
description "connected to VM2"
no sh
service instance 100 ethernet
encapsulation untagged

```

```

bridge-domain 100
member vni 4096
member GigabitEthernet2 service-instance 100

```

That should be it..

Verification

```

#show nve interface nve 1 detail
Interface: nve1, State: Admin Up, Oper Up, Encapsulation: Vxlan,
BGP host reachability: Disable, VxLAN dport: 4789
VNI number: L3CP 0 L2DP 1
source-interface: Loopback100 (primary:10.10.10.10 vrf:0)
  Pkts In   Bytes In   Pkts Out   Bytes Out
    1056     98970      844       84662

```

```

# show nve peers
Interface  VNI      Type Peer-IP           Router-RMAC   eVNI   state flags UP time

```

→ nothing there yet.. but then there is no traffic. Please bare in mind that the VXLAN tunnels are STATELESS AND BUILD AND TORN DOWN ON DEMAND!! (UDP based)

Once there is traffic being tunneled:

```

Interface  VNI      Type Peer-IP           Router-RMAC   eVNI   state flags UP time
nve1      4096     L2DP 11.11.11.11

```

```

#show platform software vxlan F0 udp-port
VXLAN UDP Port: 4789

```

VXLAN GPE Tunnel UDP Port: 4790
VXLAN Dummy L2 Tunnel UDP Port: 4789

Oh and yes, the pings worked across the Overlay network! ☺

The beauty of this..

Is of course that you can have two completely separate L2 segments that are still reachable across a routable network. If for instance you would change **encapsulation untagged** for **encapsulation dot1q** then you can tunnel a whole tagged VLAN across a VXLAN, and that's just sweet.. You can e.g. move your VM to an entirely different network and things would work as always, without having to touch – and reconfigure your VM!

A point I did come across in my research was that you **might have to increase your MTU size due to the overhead of the tunnel.** But since this is a DataCenter technique anyway, that should not be a problem.

This “overlay” thing really seems to be wonderful..