Design Notebook: 3-Stage Clos Network with Static VXLAN

Inputs: Bhavani Parise, Deepti Chandra and Developed by: Sarah Samuel

Modern cloud-scale data center networks require increased server-to-server communication over a network that is resilient despite the increased number of devices involved. In this notebook, we will explore how to bring up the 3-stage clos network as the underlay and then configure static VXLANs over it to increase the network's scalability and meet this growing demand.

A 3-stage clos network is a system of interconnecting data center network switches where each spine node connects to all leaf nodes. And each leaf node connects to a server in the data center. Any server in the data center is just 3 hops away from another server. The first hop is from the server to the directly connected leaf node, the second hop is across the spine nodes to the destination leaf node and the third hop is between the destination leaf node to the destination server. This kind of network architecture is highly scalable and irrespective of the number of devices in the data center, the number of hops between the servers or the end-hosts is always 3. This ensures consistent latency in the data center network.

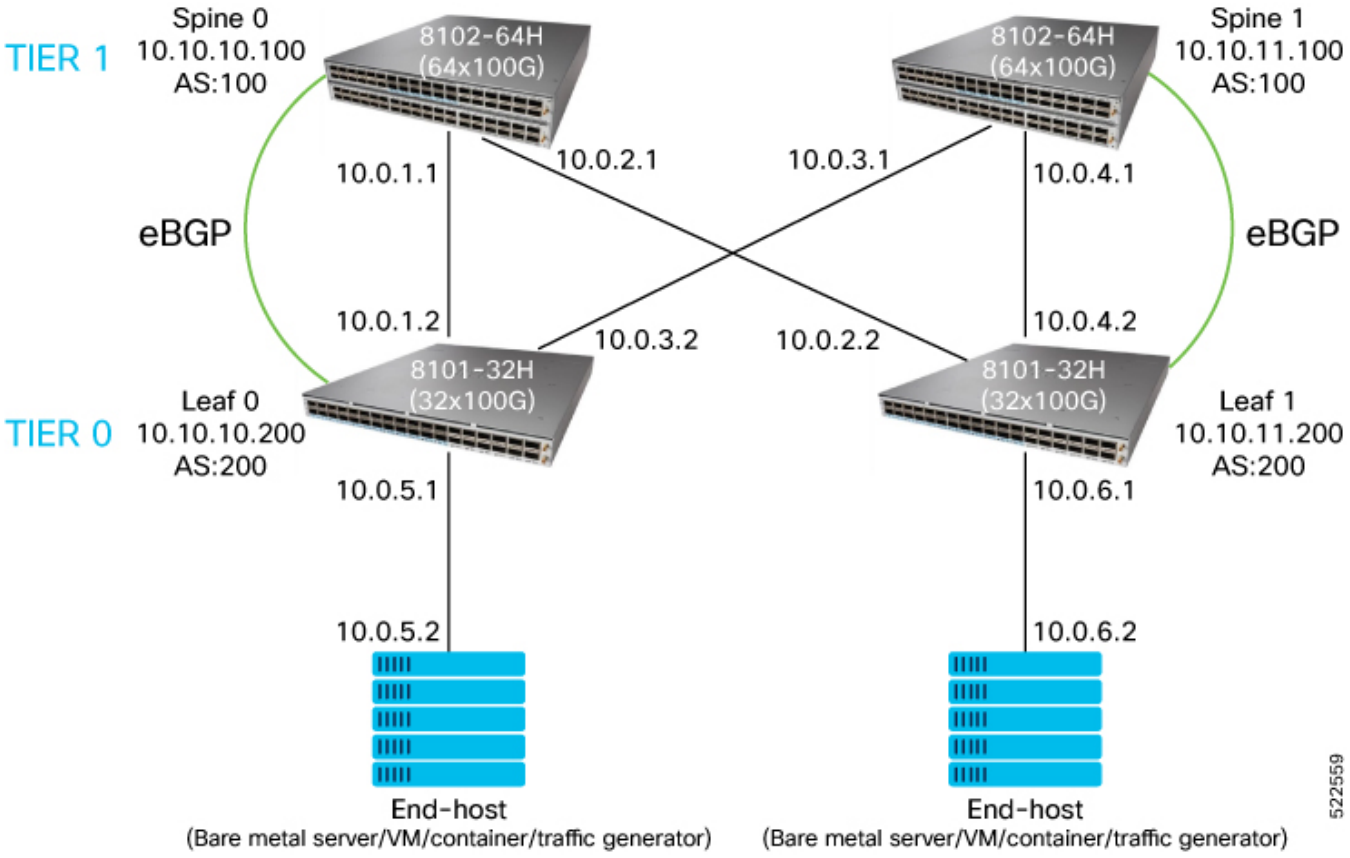
The 3-stage clos network is a robust IP-BGP underlay network for the datacenter. Over this network, you can configure overlay features such as VXLANs.

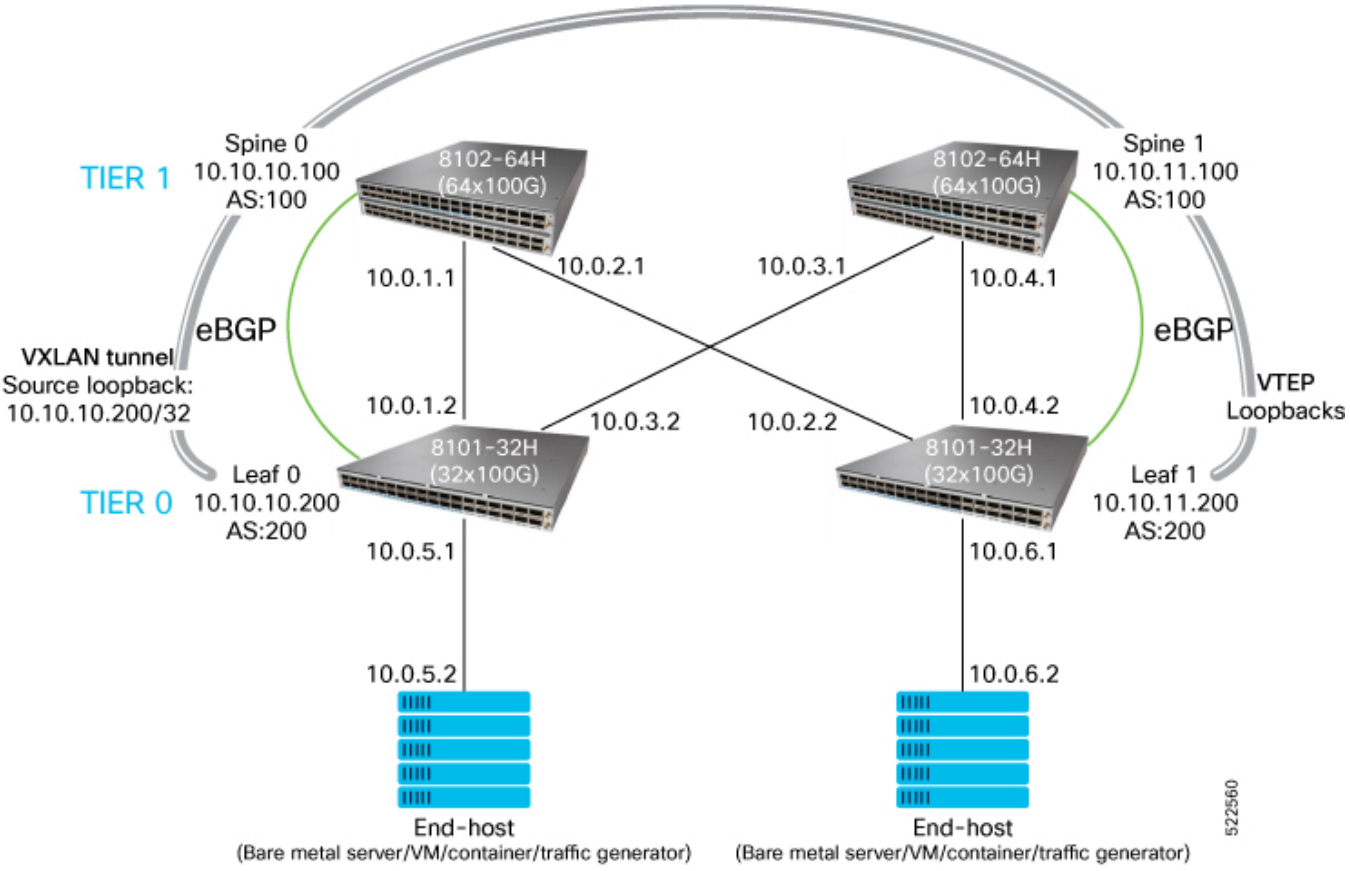
Virtual Extensible Local Area Network (VXLAN) is a tunnelling protocol that stretches layer 2 networks over an underlying layer 3 IP network. It does this by encapsulating layer 2 Ethernet frames within layer 4 User Datagram Protocol (UDP) and then transports the encapsulated frames over a layer 3 network. You can create upto 16 million VXLANs as opposed to the traditional VLAN, which allows only 4096 VLANs in a network. And so VXLAN allows you to build highly scalable networks with physically distant layer 2 network segments. There are 2 types of VXLANS:

1. Ethernet virtual private network (EVPN) VXLANs - EVPN provides the control plane functionality for these VXLANs.
2. Static VXLANs - There is no control plane for static VXLANs. So you should manually configure virtual tunnel end points and routes.

The following topology diagram depicts a simple 3-stage clos network with 2 leaf nodes in Tier-0 and 2 spine nodes in Tier- 1.

This topology shows the static VXLAN overlay over the 3-stage clos network:





After topology bring up, play through the following steps to configure 3-stage clos network:

Configure Host-Names Assign IP-Addresses Configure eBGP
Verify BGP Route Exchange Send Traffic from TREX Verify Traffic Statistics

After the 3-stage clos network is up, configure static VXLAN as the overlay, by playing through these steps:

Set Up and Apply Static VXLAN Configurations Set Up and Load VNET Route Tables
Verify Static VXLAN Tunnels
Send Traffic Across VXLAN Tunnel

Tested Static VXLAN Scale

Bring up topology and access ssh console of each device in the topology.

Configure 3-Stage Clos Network

To configure a 3-stage clos network, continue playing through the following steps:

Configure Host-Names

Configure host names for the spine and leaf nodes for easy identification.

```
out = nodes['S0'].execute('sudo config hostname SPINE0') out =
nodes['S0'].execute('sudo config save -y')
out = nodes['S1'].execute('sudo config hostname SPINE1') out =
nodes['S1'].execute('sudo config save -y')

out = nodes['L0'].execute('sudo config hostname LEAF0') out =
nodes['L0'].execute('sudo config save -y')
out = nodes['L1'].execute('sudo config hostname LEAF1') out =
nodes['L1'].execute('sudo config save -y')
```

In [2]:

```
2022-05-04 12:51:43,554: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config hostname
SPINE0' +++
sudo config hostname SPINE0
Running command: service hostname-config restart
Reloading Monit configuration ...
Reinitializing monit daemon
Please note loaded setting will be lost after system reboot. To preserve setting,
run `config save`.
cisco@sonic:~$
2022-05-04 12:51:44,776: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config save -y'
+++
sudo config save -y
Running command: /usr/local/bin/sonic-cfggen -d --print-data >
/etc/sonic/config_db.json
cisco@sonic:~$
2022-05-04 12:51:45,888: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config hostname
SPINE1' +++
sudo config hostname SPINE1
Running command: service hostname-config restart
Reloading Monit configuration ...
Reinitializing monit daemon
Please note loaded setting will be lost after system reboot. To preserve setting,
run `config save`.
cisco@sonic:~$
2022-05-04 12:51:47,160: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config save -y'
+++
sudo config save -y
Running command: /usr/local/bin/sonic-cfggen -d --print-data >
/etc/sonic/config_db.json
cisco@sonic:~$
2022-05-04 12:51:48,311: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config hostname
LEAF0' +++
sudo config hostname LEAF0
Running command: service hostname-config restart
Reloading Monit configuration ...
Reinitializing monit daemon
```

```
Please note loaded setting will be lost after system reboot. To preserve setting,
run `config save`.
cisco@sonic:~$
2022-05-04 12:51:49,398: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config save -y'
+++
sudo config save -y
Running command: /usr/local/bin/sonic-cfggen -d --print-data >
/etc/sonic/config_db.json
cisco@sonic:~$
2022-05-04 12:51:50,316: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config hostname
LEAF1' +++
sudo config hostname LEAF1
Running command: service hostname-config restart
Reloading Monit configuration ...
Reinitializing monit daemon
Please note loaded setting will be lost after system reboot. To preserve setting,
run `config save`.
cisco@sonic:~$
2022-05-04 12:51:51,257: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config save -y'
+++
sudo config save -y
Running command: /usr/local/bin/sonic-cfggen -d --print-data >
/etc/sonic/config_db.json
cisco@sonic:~$
```

Assign IP Addresses

This step assigns IP addresses on the interfaces of all devices as per the topology diagram. And then saves the configurations.

```
out = nodes['S0'].execute('sudo config interface ip add Ethernet0 10.0.1.1/24')
out = nodes['S0'].execute('sudo config interface ip add Ethernet1 10.0.2.1/24')
out = nodes['S0'].execute('sudo config interface ip add Loopback0 10.10.10.100/32')
out = nodes['S0'].execute('sudo config save -y')
out = nodes['S1'].execute('sudo config interface ip add Ethernet0 10.0.3.1/24')
out = nodes['S1'].execute('sudo config interface ip add Ethernet1 10.0.4.1/24')
out = nodes['S1'].execute('sudo config interface ip add Loopback0 10.10.11.100/32')
out = nodes['S1'].execute('sudo config save -y')
out = nodes['L0'].execute('sudo config interface ip add Ethernet0 10.0.1.2/24')
out = nodes['L0'].execute('sudo config interface ip add Ethernet1 10.0.3.2/24')
out = nodes['L0'].execute('sudo config interface ip add Ethernet2 10.0.5.1/24')
out = nodes['L0'].execute('sudo config interface ip add Loopback0 10.10.10.200/32')
out = nodes['L0'].execute('sudo config save -y')
out = nodes['L1'].execute('sudo config interface ip add Ethernet0 10.0.2.2/24')
out = nodes['L1'].execute('sudo config interface ip add Ethernet1 10.0.4.2/24')
out = nodes['L1'].execute('sudo config interface ip add Ethernet2 10.0.6.1/24')
out = nodes['L1'].execute('sudo config interface ip add Loopback0 10.10.11.200/32')
out = nodes['L1'].execute('sudo config save -y')
out = nodes['trex'].execute('ifconfig eth1 10.0.5.2 netmask 255.255.255.0 up')
out = nodes['trex'].execute('ifconfig eth2 10.0.6.2 netmask 255.255.255.0 up')
```

In [3]:

```
2022-05-04 12:52:02,330: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Ethernet0 10.0.1.1/24' +++
sudo config interface ip add Ethernet0 10.0.1.1/24
cisco@sonic:~$
2022-05-04 12:52:03,349: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Ethernet1 10.0.2.1/24' +++
sudo config interface ip add Ethernet1 10.0.2.1/24
cisco@sonic:~$
2022-05-04 12:52:04,093: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Loopback0 10.10.10.100/32' +++
sudo config interface ip add Loopback0 10.10.10.100/32
cisco@sonic:~$
2022-05-04 12:52:04,844: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config save -y'
+++
sudo config save -y
Running command: /usr/local/bin/sonic-cfggen -d --print-data >
/etc/sonic/config_db.json
cisco@sonic:~$
2022-05-04 12:52:05,918: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Ethernet0 10.0.3.1/24' +++
sudo config interface ip add Ethernet0 10.0.3.1/24
cisco@sonic:~$
2022-05-04 12:52:06,665: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Ethernet1 10.0.4.1/24' +++
sudo config interface ip add Ethernet1 10.0.4.1/24
cisco@sonic:~$
2022-05-04 12:52:07,474: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Loopback0 10.10.11.100/32' +++
sudo config interface ip add Loopback0 10.10.11.100/32
cisco@sonic:~$
2022-05-04 12:52:08,245: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config save -y'
+++
sudo config save -y
Running command: /usr/local/bin/sonic-cfggen -d --print-data >
/etc/sonic/config_db.json
cisco@sonic:~$
2022-05-04 12:52:09,331: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Ethernet0 10.0.1.2/24' +++
sudo config interface ip add Ethernet0 10.0.1.2/24
cisco@sonic:~$
2022-05-04 12:52:09,995: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Ethernet1 10.0.3.2/24' +++
sudo config interface ip add Ethernet1 10.0.3.2/24
cisco@sonic:~$
2022-05-04 12:52:10,656: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Ethernet2 10.0.5.1/24' +++
```

```
sudo config interface ip add Ethernet2 10.0.5.1/24
cisco@sonic:~$
2022-05-04 12:52:11,317: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Loopback0 10.10.10.200/32' +++
sudo config interface ip add Loopback0 10.10.10.200/32
cisco@sonic:~$
2022-05-04 12:52:12,019: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config save -y'
+++
sudo config save -y
Running command: /usr/local/bin/sonic-cfggen -d --print-data >
/etc/sonic/config_db.json
cisco@sonic:~$
2022-05-04 12:52:12,994: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Ethernet0 10.0.2.2/24' +++
sudo config interface ip add Ethernet0 10.0.2.2/24
cisco@sonic:~$
2022-05-04 12:52:13,656: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Ethernet1 10.0.4.2/24' +++
sudo config interface ip add Ethernet1 10.0.4.2/24
cisco@sonic:~$
2022-05-04 12:52:14,279: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Ethernet2 10.0.6.1/24' +++
sudo config interface ip add Ethernet2 10.0.6.1/24
cisco@sonic:~$
2022-05-04 12:52:14,924: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config interfac
e ip add Loopback0 10.10.11.200/32' +++
sudo config interface ip add Loopback0 10.10.11.200/32
cisco@sonic:~$
2022-05-04 12:52:15,586: %UNICON-INFO: +++ sonic with via 'cli': executing command
'sudo config save -y'
+++
sudo config save -y
Running command: /usr/local/bin/sonic-cfggen -d --print-data >
/etc/sonic/config_db.json
cisco@sonic:~$
2022-05-04 12:52:16,440: %UNICON-INFO: +++ localhost with via 'cli': executing
command 'ifconfig eth1 10
.0.5.2 netmask 255.255.255.0 up' +++
ifconfig eth1 10.0.5.2 netmask 255.255.255.0 up
[root@localhost ~]#
2022-05-04 12:52:16,583: %UNICON-INFO: +++ localhost with via 'cli': executing
command 'ifconfig eth2 10
.0.6.2 netmask 255.255.255.0 up' +++
ifconfig eth2 10.0.6.2 netmask 255.255.255.0 up
[root@localhost ~]#
```

Verify the IP addresses configured.

```
for n in nodes:
if (n == 'trex'):
```

```
out = nodes[n].execute('ifconfig -a eth1')

out = nodes[n].execute('ifconfig -a eth2') else:

out = nodes[n].execute('show ip interfaces')
```