Tableau Breackout a envoyé + confirmation reboot brackout + maj du schéma + remote Leaf template

# Harware Spec

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Platform | Specifications | Access Leaf | Border Gateway | Border MPLS | Spine |
| Nexus 9364D-GX2A | 64 x 400G |  |  |  |  |
| Nexus 9332D-GX2B | 32 x 400G |  |  |  |  |
| Cisco Nexus 9316D-GX | 16 x 400G |  |  |  |  |
| Nexus 93600CD-GX | 28 x 100G & 8 x 400G |  |  |  |  |
| Nexus 10/1 G | TBD |  |  |  |  |

## Leaf to Spine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Leaf Platform | Specifications | 400G Interfaces | Planned uplink ports | Description |
| Nexus 93600CD-GX | 28 x 100G & 8 x 400G | E1/29-36 | e1/35-36 | Maximum 8 spines |
| Nexus 10/1 G | TBD | TBD | TBD | TBD |

## Leaf to Leaf (VPC Peer-link), Leaf to Servers, Border/service leaf to Firewalls and Gateways Connection

First 400G port to be used as peerlink

|  |  |  |  |
| --- | --- | --- | --- |
| Leaf Platform | Specifications | 400G Interfaces | Planned vPC ports |
| Nexus 93600CD-GX | 28 x 100G & 8 x 400G | e1/29-36 | e1/29-30 |
| Nexus 10/1 G | TBD | TBD | TBD |
| Nexus 93108TC-FX | 48 x 100M/1/10GBASE-T & 6 x 40/100 | e1/49 - 54 | e1/53 - 54 |
| Nexus 9336C-FX2 | 36 x 25/40/100G | e1/1 - 36 | e1/35 - 36 |

## Breakout for Nexus

<https://www.cisco.com/c/en/us/td/docs/dcn/nx-os/nexus9000/101x/configuration/interfaces/cisco-nexus-9000-nx-os-interfaces-configuration-guide-101x/m-overview.html#concept_855836F7B4C24198891FC6A5FACF6DE7>

Please note that we need to reboot the module (leaf) to apply the configuration

 The module is reloaded and the configuration for the interface is removed when the command is executed.

We need to read : Cisco Nexus N9K-C93600CD-GX breakout considerations:

# Underlay

# IP addressing :

We use /30 subnet in the whole fabric, another option is to use /31 if don’t have sufficient network space

***Configuration -* P2P Configuration**

**! Configuration on Leaf Interface**

LEAF1(config)#Interface e1/35

LEAF1(config-if)#Description \_Uplink\_To\_Spine

LEAF1(config-if)#ip address 10.10.10.1/30

**! Configuration on Spine Interface**

SPINE1(config)#Interface e1/1

SPINE1(config-if)#Description \_Link\_To\_Leaf

SPINE1(config-if)#ip address 10.10.10.2/30

# MTU :

For all Underlay traffic (VTEP to VTEP) interfaces need to be at MTU 9100

interface x/x

description UNDERLAY interface

mtu 9100

# MTU for External connections:

Depending on the device (PE MPLS, ESX, BG, Client switch…) MTU need to be set between 1500-9000

interface x/x

description OVERLAY interface toward servers/PE/Internet …

mtu *<1500-9000>*

!

interface vlan 1000  
description OVERLAY interface toward servers

mtu *9000*

!

# TCP MSS on all switches

In order to avoid fragmentation in BGP packet, it’s a best practice to rise the default MSS value on the switch from 536 to 8960

!  
configuration terminal

ip tcp mss 8960

!

# ISIS :

We use ISIS Level-1 topology as it is the only supported topology for Nexus VXLAN Fabric.  
The overload bit is used by Nexus to signal other devices not to use the switch as an intermediate hop in their shortest path first (SPF) calculations (on startup)

feature isis

router isis UNDERLAY

log-adjacency-changes

net 49.0001.0010.0100.1001.00

is-type level-1

set-overload-bit on-startup 60

For each interface from Leaf to Spine we use this sample to add isis in the routing interface

The same configuration should be implemented between elements:

* Access Leaves and Border Leaves (PE, Internet) to Spines
* BGW – Spines,

interface Ethernet 1/35

description Link to Spine S1

mtu 9100

ip address 10.10.10.1/31

ip router isis UNDERLAY

The loopback0 is used as router id for isis

interface loopback 0

ip address 10.20.10.1/32

ip router isis UNDERLAY

# ECMP

By default, isis load balance between 8 uplink, if we need in the future more then 8 spine we will use

router isis UNDERLAY

maximum-paths <number>

# Multi-destination traffic

We have two options for BUM traffic, Unicast mode (Ingress replication ) or Multicast mode, The solution is based on Multicast mode

In multicast mode each VNI is mapped to a Mcast GRP based on PIM ASM (AnySourceMcast)

feature pim

!

interface Ethernetx/y

description Link to Spine/Leaf/BGW

ip pim sparse-mode

# RP Placement

Need to discuss with NERIM the best way

|  |  |
| --- | --- |
| Fabric options | Anycast RPs Placement |
| Option 1 | 2 RPs configured on 2 Spines (one in each site) |
| Option 2 | 2 RPs configured on 2 Spines in each site |

Template for 1 site to be replacted (with another IP scheme) in the second site

**! Loopback Interface Configuration on each RP, enable PIM on Lo0 IGP Interface**

interface loopback0

description IGP Loopback Router\_ID

ip address 10.10.10.x/32

ip pim sparse-mode

**! Loopback Interface Configuration (Anycast RP) on all RPs (Spine)**

interface loopback 254

description Anycast RP

ip address 10.10.10.254/32

ip pim sparse-mode

**! Anycast-RP Configuration on all RPs (Spine / site)**

ip pim rp-address 10.10.10.254 group-list 224.0.0.0/8

ip pim anycast-rp 10.10.10.254 10.10.10.1 (Spine-1 IP)

ip pim anycast-rp 10.10.10.254 10.10.10.2 (Spine-2 IP)

**! Configure RP for mcast group on all Leafs**

ip pim rp-address 10.10.10.254 group-list 224.0.0.0/8

# Mulicast grouping for VXLAN:

There are three main approaches to map L2VNIs to Multicast groups:

— One-to-One L2VNI to Multicast Group mapping

— VRF-based L2VNI to Multicast Group mapping

— Odd & Even L2VNI to Multicast Group mapping

The less scale and most simple is One to One, we can make as well VRF based or ODD & Even in order to scale more and limit the number of MCast groupe spread over the Fabric

*One-to-One L2VNI to Multicast Group mapping*

interfcae nve 1

member vni 10011

mcast-group 225.1.1.11

member vni 10012

mcast-group 225.1.1.12

*VRF-based L2VNI to Multicast Group mapping*

interfcae nve 1

member vni 2000201

mcast-group 225.1.1.11

member vni 2000401

mcast-group 225.1.1.12

member vni 2000402

mcast-group 225.1.1.12

*Odd & Even L2VNI to Multicast Group mapping*

interfcae nve 1

member vni 2000201

mcast-group 225.1.1.11

member vni 2000401

mcast-group 225.1.1.11

member vni 2000402

mcast-group 225.1.1.12

We can have a mix of approach depending on the infra (Servers, ESX cluster, Switch client …)

# BFD

TBD, we do not recommend BFD for underlay (only direct back to back links) and we do it for Overlay (BGP connection for BGW inter-site)

# VXLAN BGP EVPN

RR interconnection to be discussed

Two route reflectors in each site (in all templates) will be configured. RRs will be on two spine switches in each site.

EVPN BGP neighborship (to be discussed)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BGP Peer A | BGP Peer B | Site | DC1 | DC2 |
| Leafs & BGWs | Spines | Same site |  |  |
| RR Spines Site 1 ? | RR Spines Site n ? | All sites |  |  |
| BGW Site 1 | BGW Site n | All sites |  |  |

# BGP EVPN on leaves

**Template for VxLAN and EVPN + VLAN and VXLAN VNI**

**! Enable VLAN-based VXLAN**

feature vn-segment-vlan-based

**! Enable VXLAN**

feature nv overlay

**! Enable the EVPN control plane for VXLAN**

nv overlay evpn

vlan <vlan-number>

vn-segment <vn-number>

!

evpn

vni <20000+VLANID> l2

rd *RID:L2VNI*

route-target both *ASN:L2VNI*

!

Note : VNI, RD and RT could be adapted to NERM needs

**Template for VRF for VXLAN Routing**

vrf context <context-name>

vni <*40000+vrf-id*>

rd *RID:L3VNI*

address-family ipv4 unicast

route-target both *ASN:L3VNI*

route-target both *ASN:L3VNI* evpn

address-family ipv6 unicast

route-target both *ASN:L3VNI*

route-target both *ASN:L3VNI* evpn

!

VNI, RD and RT could be adapted to NERIM needs

**Template for SVI hosts for VXLAN routing**

vlan <vlan-number>

vn-segment <vn-number>

!

interface vlan <vlan-number>

vrf member <vrf-name>

ip address <ip-address> <netmask>

!

Template for VRF Overlay VLAN for VXLAN routing

**! Create the VRF overlay VLAN and configure the vn-segment**

vlan <vlan-number>

vn-segment <vn-segment>

!

**! Configure VRF overlay VLAN/SVI for the VRF**

interface VlanX

no shutdown

vrf member <”overlay vrf name”>

ip forward

**!The system vlan nve-overlay id command should be used for a VRF or a Layer-3 VNI (L3VNI) only. Do not use this command for regular VLANs or Layer-2 VNIs (L2VNI).**

system vlan nve-overlay id <>

**! Create VRF and configure VNI**

vrf context CUSTOMER-VRF

vni <vni-number>

Anycast Gateway for VxLAN Routing: Each Fabric (site) should have individual MAC address. This MAC address is shared by the anycast gateway for all edge devices of the VxLAN fabric.

Template for configuring the anycast gateway

**! Configure distributed gateway virtual MAC address**

**! All VTEPs should have the same virtual MAC address**

fabric forwarding anycast-gateway-mac <mac-address>

!

interface vlan <vlan-number>

fabric forwarding mode anycast-gateway

!

Template for configuring NVE interface and VNIs

interface nve-interface

**! This defines BGP as the mechanism for host reachability advertisement**

host-reachability protocol bgp

**! Add Layer-3 VNIs, one per tenant VRF, to the overlay**

member vni <vni-number> associate-vrf

**! Add Layer 2 VNIs to the tunnel interface**

member vni <vni-number>

**! Configure the mcast group on a per-VNI basis (depending on MCAST strategy**

mcast-group <multicast-group-address>

**! advertise virtual rmac with advertise pip in bgp**

advertise virtual-rmac

**! use Loopback1 (new loopback for VTEP peering**

source-interface loopback1

!

Template for configuring BGP on the leaves and BGW

feature bgp

**! Need to discuss Public AS vs Private AS (use template-peer)**

router bgp <ASN-number>

log-neighbor-changes

address-family l2vpn evpn

**! Requires advertise virtual-rmac in NVE**

advertise-pip

**! Recommended to provision the same as IP address of interface loopback 0 for IGP**

router-id <address>

**! Provision Spines as RR**

**! ASN-number shuld remain the same and consistent within each site**

**! Using TemplateBGP**

template peer iBGP-Spines

remote-as <ASN-number>

**! Use loopback 0 as source interface for BGP peering**

update-source loopback0

**! Configure address family Layer 2 VPN EVPN under the BGP neighbor.**

address-family l2vpn evpn

send-community

send-community extended

**! Applying template to RR**

neighbor <RR1-address>

inherit peer iBGP-Spines

description Spine-1 Loopback0

neighbor <RR2-address>

inherit peer iBGP-Spines

description Spine-2 Loopback0

**!!! VRF client**

vrf <vrf-name>

address-family ipv4 unicast

**! Configure iBGP Multipath Load Sharing (Leaf will select multiple iBGP paths as the best paths to a destination.The best paths or multipaths are then installed in the IP routing table)**

maximum-paths ibgp 2

advertise l2vpn evpn

**! In case you're willing to advertise Type 5, you need network/redistribute statements. This is for Static routing (to be avoided)**

network <ip-network/subnet>

!

address-family ipv6 unicast

advertise l2vpn evpn

!

!

# BGP EVPN on Spines

Template for configuring BGP on the spines without VTEP functionality

feature bgp

**! Need to discuss Public AS vs Private AS (use template-peer)**

router bgp <ASN-number>

log-neighbor-changes

address-family l2vpn evpn

**! Recommended to provision the same as IP address of interface loopback 0 for IGP**

router-id <address>

**! Use template for RR client**

template peer iBGP-Leafs

remote-as <ASN-number>

update-source loopback0

address-family l2vpn evpn

send-community

send-community extended

route-reflector-client

**! Applying template to Leaf**

neighbor <LEAF1-address>

inherit peer iBGP-Leafs

description LEAF-1 Loopback0

!