VXLAN

- VXLAN
- Lab 1 - VXLAN
- Lab 2 - BGP EVPN VXLAN Fabric
- Lab 3 - Multi-Homing - Dual-Homed Device
- Lab 4 - Multi-Homing - Dual-Homed Network
- Lab 5 - Stack Leaf

VXLAN

Lab 1 - VXLAN

Lab1.1 - clab-pe-exp-7

Configuration complète: https://git.uttnetgroup.fr/PE-Redondance/ContainerLab/src/branch/main/clab-pe-exp-7

Objectif

Lab1.1 met en place un environnement VXLAN simple avec deux routeurs PE, deux switches L2 et quatre clients dans un seul segment de service. L'objectif est de valider la connectivite de bout en bout sur le VLAN 20 en s'appuyant sur un underlay commun entre `rt1` et `rt2`.

Topologie

- `rt1` et `rt2` sont des routeurs Cisco C8000V.
- `sw1` et `sw2` sont des switches L2 Cisco IOL.
- `client1`, `client2`, `client3` et `client4` sont des hotes Linux `network-multitool`.

Liens physiques

- `rt1` <-> `rt2` sur `eth1`
- `rt1` <-> `sw1` sur `eth2` / `Ethernet0/1`
- `sw1` <-> `client1` sur `Ethernet0/2`
- `sw1` <-> `client2` sur `Ethernet0/3`
- `rt2` <-> `sw2` sur `eth2` / `Ethernet0/1`
- `sw2` <-> `client3` sur `Ethernet0/2`
- `sw2` <-> `client4` sur `Ethernet0/3`

Plan d'adressage

Management Containerlab

- Sous-reseau IPv4: `172.20.20.0/24`

- Passerelle IPv4: `172.20.20.1`
- Sous-reseau IPv6: `3fff:172:20:20::/64`
- Passerelle IPv6: `3fff:172:20:20::1`

Underlay

- `rt1` Loopback0: `1.1.1.1/32`
- `rt2` Loopback0: `1.1.1.2/32`
- Lien inter-routeurs: `10.0.10.0/24`
- `rt1` Gi2.10: `10.0.10.1/24`
- `rt2` Gi2.10: `10.0.10.2/24`

Service L2

- VLAN 20 transporte les hotes clients.
- `rt1` et `rt2` utilisent `GigabitEthernet3` en `service instance 20` avec encapsulation `dot1q 20`.
- VNI associee: `10020`.
- Replication VXLAN: `rt1` pointe vers `1.1.1.2`, `rt2` vers `1.1.1.1`.

Clients

- `client1`: `10.0.20.1/24`
- `client2`: `10.0.20.2/24`
- `client3`: `10.0.20.3/24`
- `client4`: `10.0.20.4/24`

Points importants de configuration

- `rt1` et `rt2` utilisent `nve1` avec `Loopback0` comme source.
- Le transport inter-PE s'appuie sur le routage IPv4 entre les loopbacks via le lien `10.0.10.0/24`.
- Le segment utilisateur est limite au VLAN 20, ce qui fait de ce lab une base simple pour verifier la diffusion et l'atteignabilite L2.

Verification rapide

1. Verifier que les interfaces de management sont accessibles depuis Containerlab.
2. Verifier le voisinage IP entre `1.1.1.1` et `1.1.1.2`.
3. Tester la connectivite entre les clients du VLAN 20 sur les deux switches.

4. Confirmer que les deux PE annoncent le VNI `10020` sur `nve1`.

Lab1.2 - clab-pe-exp-8

Configuration complète: https://git.uttnetgroup.fr/PE-Redondance/ContainerLab/src/branch/main/clab-pe-exp-8

Objectif

Lab1.2 reprend la base de Lab1.1 et ajoute un second segment de service pour valider plusieurs domaines L2 sur la meme infrastructure VXLAN. Le premier segment reste le VLAN 20, et un nouveau segment VLAN 30 est ajoute.

Topologie

- `rt1` et `rt2` sont des routeurs Cisco C8000V.
- `sw1` et `sw2` sont des switches L2 Cisco IOL.
- `client20.1`, `client20.2`, `client20.3`, `client20.4` appartiennent au service VLAN 20.
- `client30.1` et `client30.2` appartiennent au service VLAN 30.

Liens physiques

- `rt1` <-> `rt2` sur `eth1`
- `rt1` <-> `sw1` sur `eth2` / `Ethernet0/1`
- `sw1` <-> `client20.1` sur `Ethernet0/2`
- `sw1` <-> `client20.2` sur `Ethernet0/3`
- `rt2` <-> `sw2` sur `eth2` / `Ethernet0/1`
- `sw2` <-> `client20.3` sur `Ethernet0/2`
- `sw2` <-> `client20.4` sur `Ethernet0/3`
- `sw1` <-> `client30.1` sur `Ethernet1/1`
- `sw2` <-> `client30.2` sur `Ethernet1/1`

Plan d'adressage

Management Containerlab

- Sous-reseau IPv4: `172.20.20.0/24`
- Passerelle IPv4: `172.20.20.1`
- Sous-reseau IPv6: `3fff:172:20:20::/64`
- Passerelle IPv6: `3fff:172:20:20::1`

Underlay

- `rt1` Loopback0: `1.1.1.1/32`
- `rt2` Loopback0: `1.1.1.2/32`
- Lien inter-routeurs: `10.0.10.0/24`
- `rt1` Gi2.10: `10.0.10.1/24`
- `rt2` Gi2.10: `10.0.10.2/24`

Services L2

- VLAN 20: VNI `10020`
- VLAN 30: VNI `10030`
- `rt1` et `rt2` utilisent `GigabitEthernet3` avec deux `service instance`:
  - `service instance 20` en `dot1q 20`
  - `service instance 30` en `dot1q 30`
- `nve1` transporte les deux VNIs avec replication vers le PE oppose.

Clients

- Service VLAN 20
  - `client20.1`: `10.0.20.1/24`
  - `client20.2`: `10.0.20.2/24`
  - `client20.3`: `10.0.20.3/24`
  - `client20.4`: `10.0.20.4/24`
- Service VLAN 30
  - `client30.1`: `10.0.30.1/24`
  - `client30.2`: `10.0.30.2/24`

Points importants de configuration

- La structure est identique a Lab1.1 pour l'underlay et le service VLAN 20.
- L'ajout principal est le second service VLAN 30, porte par le meme lien de service entre les PE.
- Cette topologie permet de valider que plusieurs VNIs peuvent coexister sur la meme paire de PE sans melange de trafic entre les segments.

Difference avec Lab1.1

- Lab1.1 expose un seul domaine L2 utilisateur, le VLAN 20.
- Lab1.2 expose deux domaines L2 utilisateur, VLAN 20 et VLAN 30.
- Lab1.2 est donc la version etendue de Lab1.1 pour tester la coexistence de plusieurs services VXLAN.

Verification rapide

1. Verifier le voisinage underlay entre `1.1.1.1` et `1.1.1.2`.
2. Verifier la presence des VNIs `10020` et `10030` sur `nve1`.
3. Tester la connectivite intra-VLAN pour `client20.*`.
4. Tester la connectivite intra-VLAN pour `client30.*`.
5. Confirmer qu'aucun trafic ne fuit entre les VLAN 20 et 30.

Lab 2 - BGP EVPN VXLAN Fabric

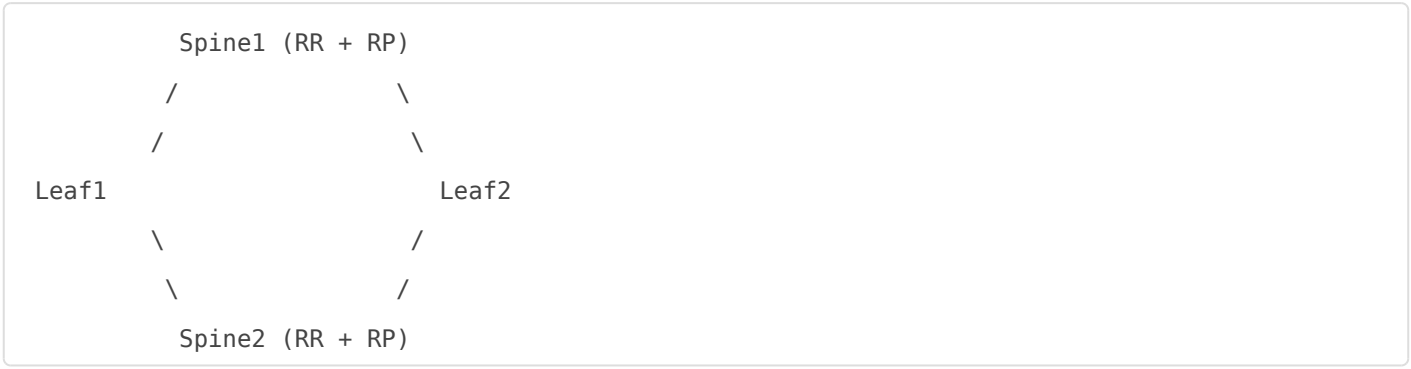
Fabric VXLAN EVPN

Configuration complète: https://git.uttnetgroup.fr/PE-Redondance/VXLAN-C9300L/src/branch/main/vxlan-lab2

Nous allons mettre en place une fabric VXLAN avec les switchs de la D206.

On se base sur la doc cisco: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9300/software/release/17-15/configuration_guide/vxlan/b_1715_bgp_evpn_vxlan_9300_cg/configuring_spine_switches_in_a_bgp_evpn_vxlan_fabric.html

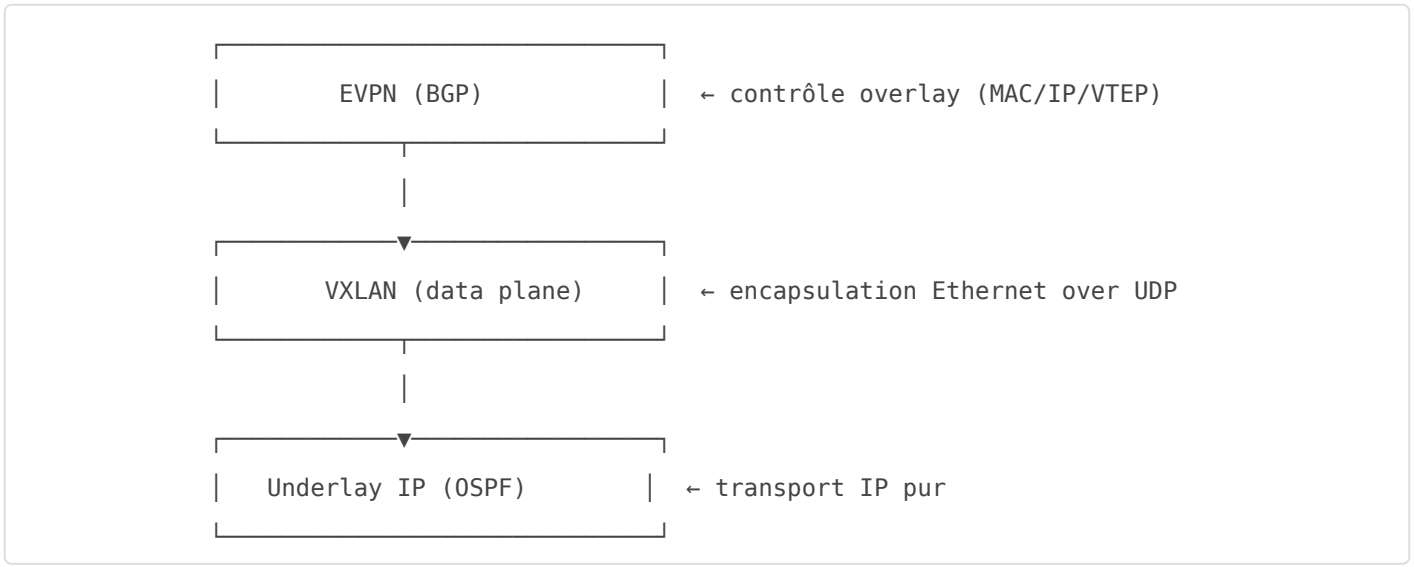
Structure



Dans cette structure nous avons les *spines* jouant le rôle intermediaire entre les *leaf* (qui seront les VTEP VXLAN).

Pour mettre à bien cette fabric nous allons utiliser différentes technologies:

- VXLAN qui agira comme **data plane**
- EVPN (avec MP-BGP) comme **control plane**
- OSPF pour l'underlay IP.
- Du **multicast** pour la gestion du traffic **BUM** (Broadcast; Unknown Unicast; Multicast)



Gestion du traffic BUM - Multicast Underlay

Une solution retenu pour gérer le traffic BUM est le Multicast Underlay et le Ingress Replication.

Leaf1 envoie un seul paquet vers un groupe multicast.

Exemple :

```
239.1.1.100
```

Le réseau PIM se charge de la duplication.

PIM (Le BUM Data Plane)

PIM est **Protocol Independent Multicast**.

PIM permet de construire les arbres multicast.

Multicast RP

Chaque Spine est un Anycast **RP** (Rendezvous Point) (Sparse Mode).

Exemple:

Quand Leaf2 veut recevoir :

```
239.1.1.100
```

il envoie :

```
PIM Join
```

vers le RP.

Quand Leaf1 émet :

```
239.1.1.100
```

il envoie aussi vers le RP.

MSDP

MSDP pour Multicast Source Discovery Protocol.

Permet à Spine1 de savoir ce que Spine2 sait.

Sinon:

```
Source connue par Spine1
Récepteur connecté à Spine2
```

Mais personne ne communique entre les deux.

Exemple:

```
VNI 10010

Leaf1
Leaf2
Leaf3
```

associés à :

```
239.1.1.10
```

Leaf2 et Leaf3 rejoignent :

```
239.1.1.10
```

via PIM.

Leaf1 reçoit un ARP Broadcast.

Il encapsule :

```
VXLAN
VNI 10010
Destination multicast :
239.1.1.10
```

L'underlay multicast :

```
Leaf1
  |
Spine
 / \
L2 L3
```

réplique automatiquement.

Leaf2 reçoit :

```
ARP Request
```

Leaf3 reçoit :

```
ARP Request
```

sans que Leaf1 ait dû envoyer deux copies.

Gestion du traffic BUM - Ingress Replication

Une autre méthode plus simple c'est de faire du **Ingress Replication** qui est de plus en plus utilisé (car plus simple, mais plus gourmand en ressources.)

Leaf1 crée lui-même N copies.

Exemple :

```
      Leaf1
        |
   +----+----+
   |    |    |
Leaf2 Leaf3 Leaf4
```

Pour un broadcast :

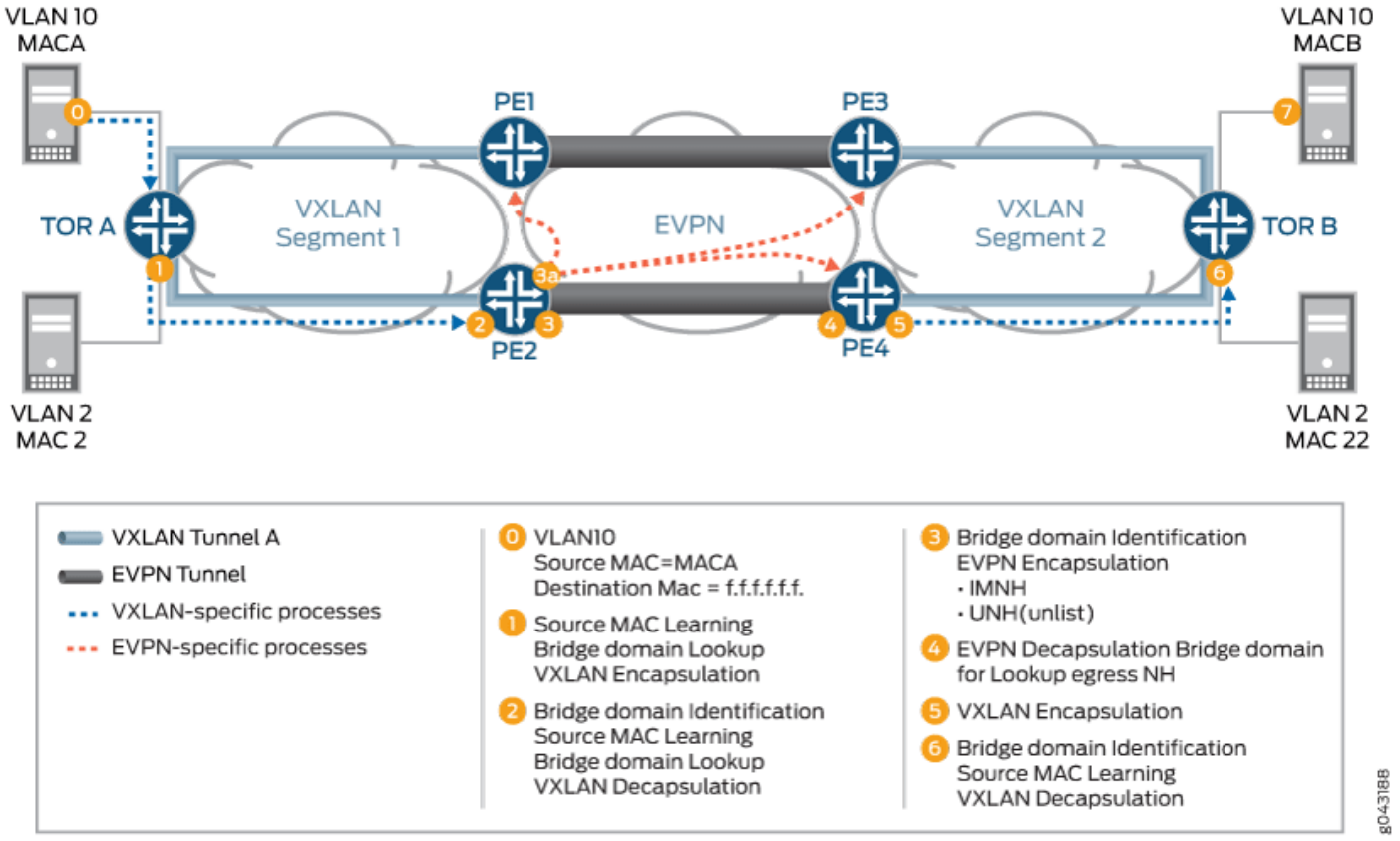
```
Copy 1 -> Leaf2
Copy 2 -> Leaf3
Copy 3 -> Leaf4
```

Le leaf source fait tout le travail.

C'est ce qu'on appelle :

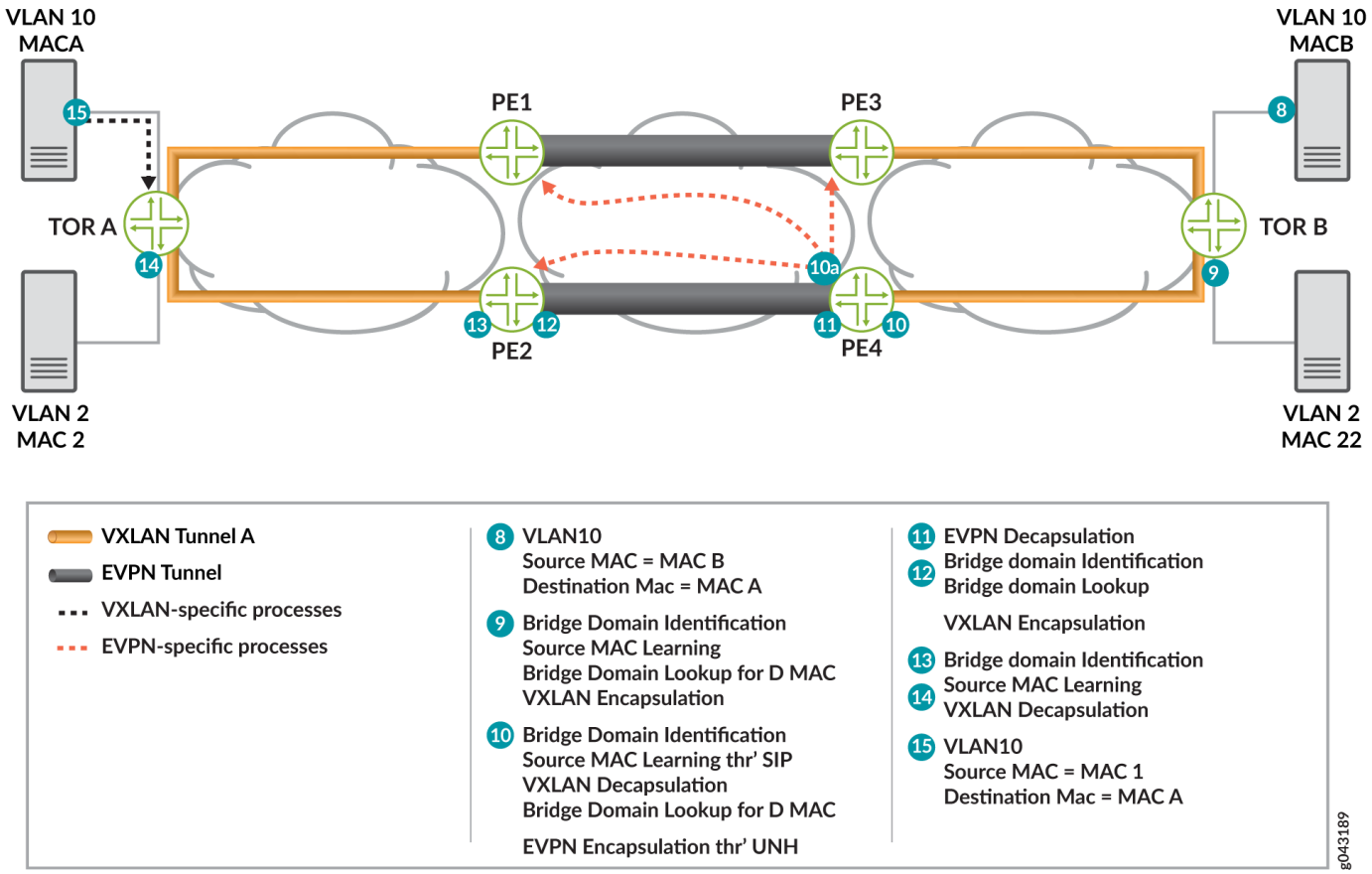
- **HER (Head End Replication)**
- Ingress Replication

Les routes EVPN Type 3 servent notamment à annoncer les VTEP participants afin que le leaf sache à qui envoyer ces copies.



(source: https://www.juniper.net/documentation/fr/fr/software/junos/evpn/topics/concept/vxlan-evpn-integration-overview.html)

Gestion du traffic Unicast



(source: https://www.juniper.net/documentation/fr/fr/software/junos/evpn/topics/concept/vxlan-evpn-integration-overview.html)

Chemin classique

Étape 1 — OSPF construit le transport

```
Leaf1 → OSPF → Leaf2 reachable via IP
```

donc Leaf1 peut envoyer des paquets IP à Leaf2 loopback

Étape 2 — BGP EVPN monte dessus

BGP utilise OSPF pour établir ses sessions :

```
BGP neighbor = Loopback IP
```

Ex :

```
neighbor 22.22.22.22 remote-as 65001
update-source loopback0
```

donc :

- OSPF = reachability
- BGP = overlay control plane

Étape 3 — EVPN distribue les infos VXLAN

BGP EVPN annonce :

```
MAC A → Leaf2
VNI 10010 members → Leaf1/2/3
```

Étape 4 — VXLAN utilise ces infos

Quand Leaf1 reçoit un paquet :

```
Host A → Host B (MAC inconnue ou connue)
```

Leaf1 regarde EVPN table :

```
MAC B → VTEP Leaf2
```

Étape 5 — VXLAN encapsule

```
Ethernet frame
↓
VXLAN tunnel
↓
IP via OSPF underlay
```

Configuration de base

La configuration ci-dessous correspond à la mise en place VXLAN/EVPN du lab2, qui sert ensuite de socle au lab3. On garde ici le modèle simple sans multi-homing : une fabric avec deux spines, deux leafs VTEP, un underlay OSPF et un overlay BGP EVPN.

L’idée est d’appliquer exactement les briques décrites plus haut :

- OSPF pour la reachability underlay,
- BGP EVPN pour distribuer les informations de contrôle,
- VXLAN pour encapsuler les VLANs du campus,
- multicast,
- des loopbacks stables pour le routage et le NVE.

Configuration commune des leafs

Les leafs sont les VTEP VXLAN (via les interfaces `NVE` qui s'occupe de désencapsuler les trames). Elles portent les VLANs du domaine utilisateur, construisent les tunnels VXLAN et annoncent leurs informations via EVPN.

```
vrf definition green
 rd 1:1
 !
 address-family ipv4
  route-target export 1:1
  route-target import 1:1
  route-target export 1:1 stitching
  route-target import 1:1 stitching
 exit-address-family
 !
 address-family ipv6
  route-target export 1:1
  route-target import 1:1
  route-target export 1:1 stitching
  route-target import 1:1 stitching
 exit-address-family

ip routing
ip multicast-routing

l2vpn evpn
 replication-type static
 router-id Loopback1
 default-gateway advertise

l2vpn evpn instance 101 vlan-based
 encapsulation vxlan
 replication-type static
```

```
l2vpn evpn instance 102 vlan-based
 encapsulation vxlan
 replication-type ingress

vlan configuration 101
 member evpn-instance 101 vni 10101
vlan configuration 102
 member evpn-instance 102 vni 10102
vlan configuration 901
 member vni 50901

interface Loopback0
 ip address 172.16.255.3 255.255.255.255
 ip ospf 1 area 0

interface Loopback1
 ip address 172.16.254.3 255.255.255.255
 ip pim sparse-mode
 ip ospf 1 area 0

interface nve1
 no ip address
 source-interface Loopback1
 host-reachability protocol bgp
 member vni 10101 mcast-group 225.0.0.101
 member vni 10102 ingress-replication
 member vni 50901 vrf green

router ospf 1
 router-id 172.16.255.3

router bgp 65001
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 172.16.255.1 remote-as 65001
 neighbor 172.16.255.1 update-source Loopback0
 neighbor 172.16.255.2 remote-as 65001
 neighbor 172.16.255.2 update-source Loopback0
 !
 address-family l2vpn evpn
```

```
  neighbor 172.16.255.1 activate
  neighbor 172.16.255.1 send-community both
  neighbor 172.16.255.2 activate
  neighbor 172.16.255.2 send-community both
 exit-address-family
 !
 address-family ipv4 vrf green
  advertise l2vpn evpn
  redistribute static
  redistribute connected
 exit-address-family
```

Dans cette configuration, `Loopback0` sert d'adresse de voisinage BGP et `Loopback1` sert de source VXLAN. Le `router-id` OSPF et le `router-id` EVPN sont choisis sur des loopbacks stables pour éviter qu'un changement d'interface physique ne casse le plan de contrôle.

Le VNI 10101 utilise le multicast underlay pour la diffusion BUM, ce qui colle au principe décrit plus haut. Le VNI 10102 est configuré en ingress replication, ce qui montre qu'on peut mélanger les mécanismes de réplication selon les besoins du service.

Configuration commune des spines

Les spines servent de transport underlay et de route reflectors BGP. Ils ne portent pas de VNIs, mais ils doivent rendre les loopbacks des leafs joignables et relayer les sessions EVPN.

```
ip routing
ip multicast-routing

interface Loopback0
 ip address 172.16.255.1 255.255.255.255
 ip ospf 1 area 0

interface Loopback1
 ip address 172.16.254.1 255.255.255.255
 ip ospf 1 area 0

interface Loopback2
 ip address 172.16.255.255 255.255.255.255
 ip pim sparse-mode
 ip ospf 1 area 0
```

```
interface GigabitEthernet2/0/1
 no switchport
 ip address 172.16.13.1 255.255.255.0
 ip pim sparse-mode
 ip ospf network point-to-point
 ip ospf 1 area 0

interface GigabitEthernet2/0/2
 no switchport
 ip address 172.16.14.1 255.255.255.0
 ip pim sparse-mode
 ip ospf network point-to-point
 ip ospf 1 area 0

interface GigabitEthernet2/0/3
 no switchport
 ip address 172.16.15.1 255.255.255.0
 ip pim sparse-mode
 ip ospf network point-to-point
 ip ospf 1 area 0

router ospf 1
 router-id 172.16.255.1

router bgp 65001
 template peer-policy RR-PP
  route-reflector-client
  send-community both
 exit-peer-policy
 !
 template peer-session RR-PS
  remote-as 65001
  update-source Loopback0
```

Les mêmes principes s’appliquent sur Spine-02, avec les adresses adaptées. L’idée est de garder un underlay simple : les leafs s’envoient leurs loopbacks via OSPF, puis BGP EVPN s’appuie dessus pour diffuser les routes MAC/IP et les informations VXLAN.

Configuration des ports d’accès

Le lab2 utilise ensuite des ports d’accès classiques sur les leafs pour rattacher les VLANs aux VTEP. C’est cette partie qui rend la fabric exploitable pour des machines ou des services locaux.

```
interface GigabitEthernet1/0/12
 switchport access vlan 101
 switchport mode access

interface GigabitEthernet1/0/24
 switchport access vlan 101
 switchport mode access
```

Avec cette configuration, le lab2 fournit le socle de base : un domaine L2 distribué par VXLAN EVPN, sans redondance d’accès avancée. Le lab3 reprendra ce socle en ajoutant seulement la couche multi-homing, sans devoir refaire l’underlay ni le mapping VXLAN.

Lab 3 - Multi-Homing - Dual-Homed Device

Multi-Homing - Dual-Homed Device

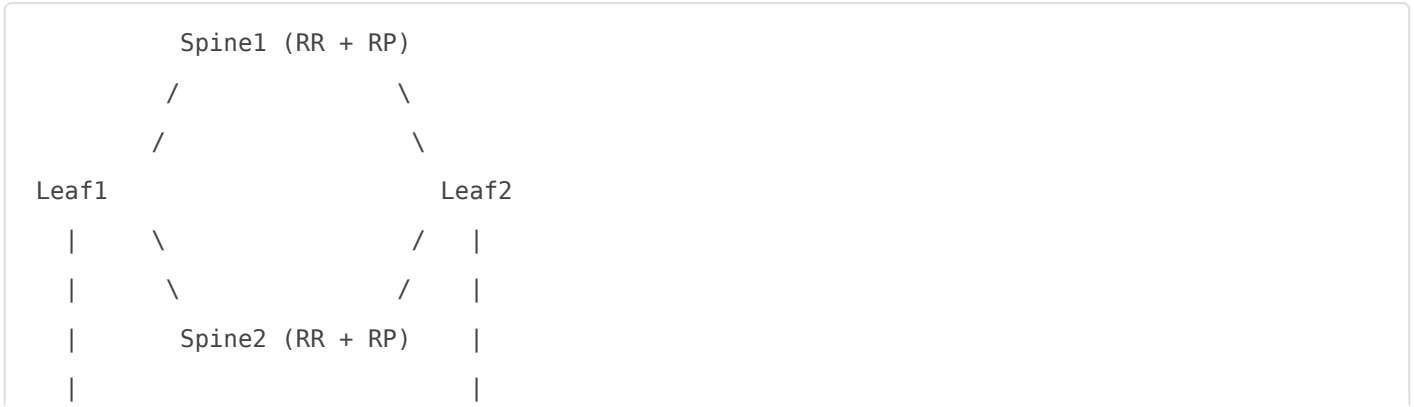
Configuration complète: https://git.uttnetgroup.fr/PE-Redondance/VXLAN-C9300L/src/branch/main/vxlan-lab3

Ce lab3 part de la fabric VXLAN EVPN du lab2 et n'en change pas le socle underlay/overlay. La différence importante est l'ajout du multi-homing EVPN pour un équipement terminal, afin de supprimer le point de défaillance unique que représentait un lien d'accès isolé.

L'objectif n'est donc pas de redécrire VXLAN ou EVPN, mais d'expliquer ce qui a été ajouté pour que deux leafs puissent présenter un même segment logique vers un serveur, avec un comportement all-active.

L'implémentation s'appuie sur les principes EVPN Multi-Homing décrits dans la documentation Cisco Catalyst 9300 pour les fabrics BGP EVPN VXLAN :
https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9300/software/release/17-15/configuration_guide/vxlan/b_1715_bgp_evpn_vxlan_9300_cg/configuring_multi_homing_in_bgp_evpn_vxlan_fabric.html

Structure



```
|                          |
|---- Port-Channel ---- Serveur
```

Le cœur de la topologie reste celui du lab2 : deux spines assurent le transport underlay et les leafs restent les VTEP. La nouveauté du lab3 se trouve au bord de la fabric, où le serveur est présenté comme un équipement dual-homed vers les deux leafs.

Ce montage permet de valider un cas plus réaliste qu'une simple liaison d'accès unique : le service continue d'exister si un lien physique, un port ou un leaf disparaît.

Multi-Homing en EVPN VXLAN

Le multi-homing EVPN consiste à faire apparaître deux VTEP comme un seul point logique d'attachement pour un même segment Ethernet. Dans le lab3, cela se traduit par une configuration identique sur Leaf-01 et Leaf-02, avec un même Ethernet Segment Identifier (ESI) et un même VLAN de service.

> L'idée est que EVPN permet de crée un segment réseau identique virtuellement à l'aide des Ethernet Segment Identifier (ESI). Ce qui permet donc de réaliser la topologie ci-dessus.

La partie importante n'est pas le sous-réseau de transport du lab2, qui reste inchangé, mais la façon dont le service VLAN est attaché à la fabric :

- le segment est identifié par un ESI commun,
- les deux leafs déclarent ce même segment,
- le serveur est raccordé via un agrégat de liens,
- EVPN arbitre le forwarding grâce au DF election.

Ce qui change par rapport à un VXLAN "basique" sans dual-homing

Dans un VXLAN simple, un port d'accès ou un VTEP unique suffit pour faire transiter le trafic. Ici, on ajoute les éléments suivants :

- un `l2vpn evpn ethernet-segment` sur les deux leafs,
- le même identifiant ESI sur les deux équipements,
- un `Port-channel` côté leaf associé à cet Ethernet Segment,
- un `Port-channel` côté serveur avec plusieurs interfaces physiques en `channel-group` actif,

- un délai de `df-election` pour laisser converger le control plane avant la désignation du forwarding actif.

Configuration appliquée sur les leafs

Sur Leaf-01 et Leaf-02, la configuration commune est la suivante :

```
l2vpn evpn ethernet-segment 1
 identifier type 0 00.00.00.00.00.00.00.00.01
 redundancy all-active
 df-election wait-time 1

interface Port-channel12
 switchport access vlan 101
 switchport mode access
 evpn ethernet-segment 1

interface GigabitEthernet1/0/12
 switchport access vlan 101
 switchport mode access
 channel-group 12 mode active
```

Le point clé est la cohérence entre les deux leafs. Le même ESI doit être déclaré des deux côtés, sinon EVPN ne comprend pas que les liens appartiennent à un même segment redondé.

`identifier type 0 ...` indique un ESI défini manuellement. C'est le choix le plus simple pour le lab : il garantit que les deux leafs annoncent exactement le même identifiant, sans dépendre d'un calcul automatique.

`redundancy all-active` place les deux leafs dans un mode où ils participent tous les deux au forwarding. Ce n'est pas seulement une redondance de secours : les deux chemins sont considérés comme utilisables par le segment.

`df-election wait-time 1` évite une élection trop précoce au moment du démarrage. On laisse le temps à EVPN de propager les routes et les informations de segment avant de figer le rôle de forwarding.

`Port-channel12` est l'interface logique sur laquelle on accroche le segment EVPN. Le `Port-channel` sert de point de rattachement stable, au lieu d'attacher le segment directement à une interface physique isolée.

Le VLAN 101 est le VLAN du service dual-homé dans ce lab. Il est associé au VNI 10101 côté fabric, ce qui permet à EVPN de transporter ce segment au travers du VXLAN sans changer son

comportement au bord de la fabric.

Configuration côté serveur

Le serveur de test est lui aussi agrégé, avec deux liens physiques dans un même bundle :

```
interface Port-channel10
 no switchport
 ip address 10.1.101.100 255.255.255.0

interface GigabitEthernet1/0/1
 description vers Leaf-01
 no switchport
 no ip address
 channel-group 10 mode active

interface GigabitEthernet1/0/2
 description vers Leaf-02
 no switchport
 no ip address
 channel-group 10 mode active
```

L’idée est simple : le serveur ne dépend plus d’un seul lien physique. Les deux interfaces sont placées dans le même `channel-group`, ce qui matérialise le double attachement vers la fabric.

Le fait que le port-channel soit routé (`no switchport`) montre que le lab met surtout en évidence la résilience de l’attachement et non un simple bridging L2 local. Le trafic du serveur repose donc sur un agrégat logique unique, tout en profitant de la redondance physique.

Modes de fonctionnement du multi-homing

EVPN définit deux grands modèles de multi-homing : `single-active` et `all-active`. Dans ce lab3, le choix retenu est `all-active`, car il correspond le mieux à l’objectif de validation de la redondance effective des deux leafs.

1. Single-Active (mode actif / standby)

Le mode single-active conserve une logique de secours : un seul leaf est réellement utilisé pour le forwarding d'un VLAN donné, l'autre reste en attente.

Dans un contexte de production, ce mode peut être intéressant lorsque l'on veut garder un comportement très conservateur. Pour le lab3, en revanche, il n'est pas le plus représentatif de l'objectif recherché, car il n'exploite pas simultanément les deux chemins.

On le cite surtout pour comparaison : il montre que le multi-homing EVPN ne se limite pas à faire de la redondance de secours, mais peut aussi distribuer le trafic.

2. All-Active (mode actif / actif)

Le mode all-active est celui utilisé dans le lab3.

Principe :

- les deux leafs participent au forwarding,
- le même Ethernet Segment est visible sur les deux VTEP,
- EVPN évite les boucles Layer 2 en s'appuyant sur l'élection DF,
- le trafic continue à circuler si un des liens ou un des leafs tombe.

Dans les faits, cela veut dire que l'infrastructure n'a plus besoin d'un chemin "principal" et d'un chemin "secours" pour le segment 101. La redondance devient active des deux côtés, ce qui améliore la disponibilité du service et évite de sous-utiliser un lien prêt à porter du trafic.

Designated Forwarder (DF)

Le Designated Forwarder est le rôle qui permet à EVPN de décider quel leaf doit gérer certains flux du segment, notamment pour le trafic BUM et la coordination du forwarding sur le VLAN concerné.

Dans le lab3, l'élection DF reste importante même en all-active, parce qu'elle sert à éviter que les deux leafs ne fassent exactement la même chose au même moment sur le même segment.

Le délai `df-election wait-time 1` est là pour stabiliser le comportement au démarrage. On laisse la fabric converger avant de figer la répartition des rôles.

En pratique, le DF ne remet pas en cause le fait que les deux leafs soient configurés en all-active. Il encadre simplement la manière dont EVPN orchestre le forwarding sur le segment commun.

Interaction avec STP (important)

Dans ce lab3, le mécanisme de protection du segment ne repose plus sur STP pour décider quel chemin est utilisable. La logique de redondance est portée par EVPN et par l'ESI associé au port-

channel.

STP reste un protocole de garde-fou pour les VLANs ou les domaines L2 classiques, mais pour le segment dual-homé du lab3, c’est EVPN qui coordonne le comportement de forwarding.

Autrement dit :

- STP n’est pas le mécanisme principal de résilience pour le service dual-homé,
- EVPN prend en charge la cohérence du segment,
- le port-channel sert de support physique logique,
- et le DF assure la coordination du forwarding au niveau de la fabric.

Lab 4 - Multi-Homing - Dual-Homed Network

Multi-Homing - Dual-Homed Network

Configuration complète: https://git.uttnetgroup.fr/PE-Redondance/VXLAN-C9300L/src/branch/main/vxlan-lab4

Le lab4 reprend la fabric VXLAN EVPN du lab2 et le premier cas de multi-homing du lab3, mais il déplace le point de redondance un cran plus bas dans l'architecture. On ne parle plus d'un serveur dual-homé, mais d'un réseau d'accès complet présenté comme un domaine L2 unique vers la fabric.

L'idée est de simuler un bloc d'accès de campus où plusieurs équipements en bordure de réseau sont eux-mêmes redondés vers les leafs. Le sujet n'est donc pas la mise en place du VXLAN de base, mais la manière dont EVPN étend son mécanisme de multi-homing à un segment d'accès agrégé.

L'implémentation s'appuie sur les principes EVPN Multi-Homing décrits dans la documentation Cisco Catalyst 9300 pour les fabrics BGP EVPN VXLAN :

https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9300/software/release/17-15/configuration_guide/vxlan/b_1715_bgp_evpn_vxlan_9300_cg/configuring_multi_homing_in_bgp_evpn_vxlan_fabric.html

Structure



```
      |    Spine2 (RR + RP)  |
      |                      |
    ------------------------------
    |                            |
Acces-SW1  ================  Acces-SW2
    |                            |
VLANs / hosts / endpoints / services L2
```

Le socle de la topologie reste celui du lab2 : les spines assurent l'underlay et les leafs sont les VTEP VXLAN. La différence importante se situe au bord de la fabric, où le réseau d'accès n'est plus vu comme un simple switch isolé, mais comme un segment logique redondé qui remonte vers les deux leafs.

Cette approche permet de valider un cas plus représentatif d'un réseau de campus : la disponibilité ne dépend plus d'un seul équipement d'accès, et le trafic peut continuer à circuler même si un switch d'accès, un uplink ou un chemin complet disparaît.

Principe du Dual-Homed Network

Le dual-homed network applique le multi-homing EVPN à un ensemble de switches L2 qui représentent un domaine d'accès. Dans le lab4, les deux switches d'accès sont présentés comme un bloc unique pour les VLANs de service, avec des uplinks redondés vers la fabric.

Concrètement, cela change le niveau où l'on place la redondance :

- le lab3 multi-homait un serveur,
- le lab4 multi-home un réseau d'accès entier,
- EVPN doit donc protéger un segment qui transporte potentiellement plusieurs VLANs, pas seulement un endpoint unique.

Le point essentiel est que les leafs ne voient pas deux accès indépendants, mais un même Ethernet Segment logique. C'est ce segment partagé qui permet à EVPN d'appliquer la logique de DF election et d'éviter les boucles Layer 2.

Modèle "stack logique" du lab

Dans ce lab, Access-SW1 et Access-SW2 simulent un domaine d'accès unique. Le but n'est pas de refaire un design de production complet avec tous les détails d'un chassis virtuel, mais de reproduire le comportement attendu d'un bloc d'accès redondé :

- un seul point logique d'attachement vers la fabric,
- plusieurs liens physiques derrière ce point logique,
- une continuité de service si un des membres tombe,

- une gestion EVPN du forwarding au lieu d’une simple logique locale de switch.

Ce qui est ajouté par rapport à un VXLAN basique sans dual-homing

Dans un VXLAN standard, un VLAN est simplement rattaché à un VTEP ou à un port d’accès classique. Ici, on ajoute les éléments nécessaires pour faire comprendre à EVPN que le réseau d’accès est redondé :

- un `l2vpn evpn ethernet-segment` dédié au segment d’accès,
- un ESI identique sur les deux leafs,
- un `Port-channel` côté leaf pour porter le segment logique,
- un `Port-channel` côté accès pour agréger les liens physiques,
- un trunk qui transporte plusieurs VLANs de service au lieu d’un seul VLAN d’endpoint,
- un `df-election wait-time` pour laisser le control plane converger avant de figer le rôle de forwarding.

Le point important est que la redondance n’est pas seulement physique. Elle est aussi décrite au niveau EVPN, ce qui permet à la fabric de comprendre que les liens appartiennent au même domaine d’accès.

Configuration appliquée sur les leafs

Sur Leaf-01 et Leaf-02, la partie commune au lab4 ressemble à ceci :

```
l2vpn evpn ethernet-segment 2
 identifier type 0 00.00.00.00.00.00.00.00.02
 redundancy all-active
 df-election wait-time 1

interface Port-channel14
 switchport trunk allowed vlan 101,102
 switchport mode trunk
 evpn ethernet-segment 2

interface GigabitEthernet1/0/13
 switchport trunk allowed vlan 101,102
 switchport mode trunk
 channel-group 14 mode active
```

```
interface GigabitEthernet1/0/14
 switchport trunk allowed vlan 101,102
 switchport mode trunk
 channel-group 14 mode active
```

Le `ethernet-segment 2` est l'élément central de cette configuration. Il identifie le second domaine redondé du lab, distinct du segment utilisé pour le cas précédent du serveur dual-homé. L'idée est de montrer que plusieurs Ethernet Segments peuvent coexister dans la même fabric, chacun avec son propre rôle.

`identifier type 0 ...02` fixe un ESI manuel. C'est un choix adapté au lab, car il garantit que les deux leafs annoncent exactement le même identifiant sans dépendre d'un calcul automatique. Si l'ESI n'est pas identique des deux côtés, EVPN ne peut pas comprendre que les uplinks appartiennent à un même segment redondé.

`redundancy all-active` signifie que les deux leafs participent au forwarding. Ce n'est pas un mode de secours passif : les deux chemins sont utilisables, ce qui colle bien à un réseau d'accès qui doit rester opérationnel même en cas de perte d'un lien ou d'un équipement.

`df-election wait-time 1` introduit un court délai avant l'élection du Designated Forwarder. Ce délai est utile pour éviter une décision trop précoce au démarrage, le temps que les annonces EVPN et la visibilité du segment soient stables.

`Port-channel14` matérialise l'attachement logique du réseau d'accès à la fabric. Le port-channel est la bonne abstraction ici, parce que le segment ne doit pas dépendre d'une interface physique unique. Les deux liens physiques `Gi1/0/13` et `Gi1/0/14` sont donc mis dans le même bundle LACP.

Le trunk autorise les VLANs 101 et 102, ce qui montre la vraie différence avec le lab3 : on ne transporte plus un seul VLAN de service, mais un petit domaine d'accès complet. Chaque VLAN reste ensuite mappé dans l'overlay EVPN/VXLAN comme dans le socle du lab2 ; le travail du lab4 porte surtout sur la manière dont ce trafic arrive sur la fabric.

Configuration côté accès

Du côté du réseau d'accès, le switch logique agrège aussi ses uplinks dans un port-channel unique :

```
interface Port-channel24
 switchport trunk allowed vlan 101,102
 switchport mode trunk

interface GigabitEthernet1/0/23
 switchport trunk allowed vlan 101,102
```

```
 switchport mode trunk
 channel-group 24 mode active

interface GigabitEthernet1/0/24
 switchport trunk allowed vlan 101,102
 switchport mode trunk
 channel-group 24 mode active
```

Le but est de représenter un domaine d’accès qui ne s’appuie pas sur un seul câble vers la fabric. Les deux interfaces physiques sont réunies dans un seul agrégat logique, ce qui donne au réseau d’accès un comportement stable vis-à-vis des leafs.

L’intérêt du trunk est également important : il permet de faire transiter plusieurs VLANs de service sans multiplier les liens. Le lab4 insiste donc sur une logique d’accès “campus” plutôt que sur un simple test de résilience point à point.

Modes de fonctionnement du multi-homing

EVPN distingue toujours deux grands modes de multi-homing : `single-active` et `all-active`. Pour ce lab4, le choix retenu est `all-active`, parce qu’il correspond le mieux à un réseau d’accès qui doit exploiter ses deux chemins de manière simultanée.

1. Single-Active

Le mode single-active conserve une logique de secours : un seul leaf participe réellement au forwarding pour le segment concerné, l’autre reste en attente.

Dans un design très conservateur, ce mode peut avoir un intérêt. Pour le lab4, il est moins pertinent, car il ne met pas en évidence le principe recherché ici : faire porter un domaine d’accès complet par deux leafs en parallèle.

2. All-Active

Le mode all-active est celui utilisé dans le lab4.

Principe :

- les deux leafs annoncent le même Ethernet Segment,
- les deux uplinks sont considérés comme membres du même domaine logique,
- EVPN coordonne le forwarding pour éviter les boucles,

- le réseau d’accès reste joignable si un des chemins tombe.

Dans ce mode, la redondance est réellement active des deux côtés. Le réseau d’accès n’attend pas qu’un chemin échoue pour être utilisé ; il s’appuie sur les deux leafs dès que la fabric est convergée.

Designated Forwarder (DF)

Le Designated Forwarder reste un point clé du lab4, même si le segment est en all-active. Son rôle est de décider quel leaf porte certains flux du segment, notamment pour le trafic BUM et la coordination du forwarding sur le VLAN concerné.

Ici, le DF ne remet pas en cause l’idée de dual-homing actif. Il évite simplement que les deux leafs traitent exactement les mêmes flux de la même manière au même instant sur le même segment.

Le paramètre `df-election wait-time 1` aide à stabiliser cette décision au démarrage. On laisse EVPN propager les informations du segment avant de figer les rôles.

Interaction avec STP

Dans le lab4, la résilience du segment d’accès ne repose pas sur STP. La logique principale de protection est portée par EVPN et par l’ESI partagé entre les leafs.

STP peut rester présent comme mécanisme de garde-fou dans le domaine L2 local, mais il ne pilote pas le comportement de redondance du segment dual-homé. Pour ce cas précis :

- EVPN coordonne le domaine redondé,
- le port-channel porte l’agrégation physique,
- le DF arbitre le forwarding sur le segment,
- STP n’est pas la brique qui décide quel chemin est actif pour le réseau d’accès.

Lab 5 - Stack Leaf

Multi-Homing - Stack Leaf

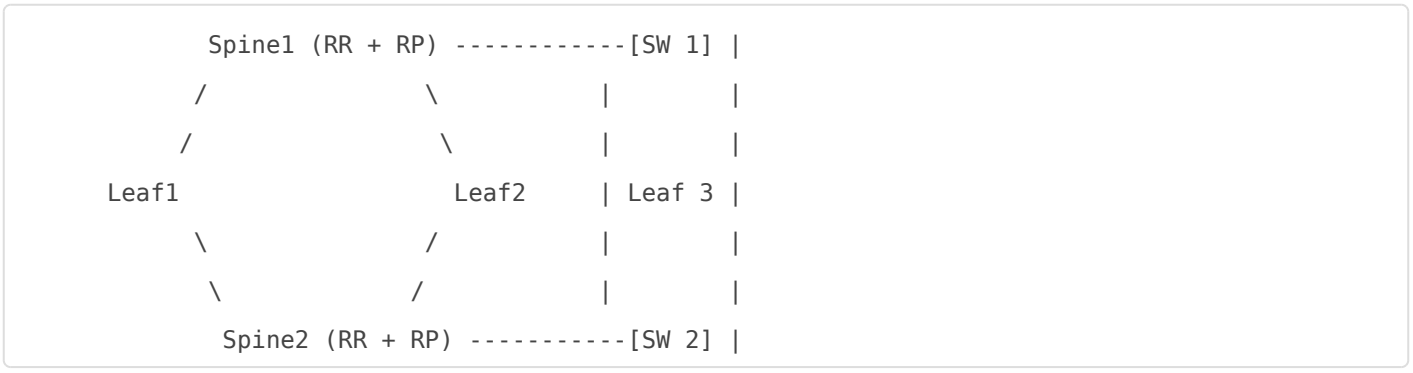
Configuration complète: https://git.uttnetgroup.fr/PE-Redondance/VXLAN-C9300L/src/branch/main/vxlan-lab5

Le lab5 reprend la fabric VXLAN EVPN des labs précédents, mais change le point d'intégration du domaine d'accès. Ici, la stack d'accès n'est plus un bloc séparé placé devant la fabric : elle devient directement une leaf EVPN à part entière.

L'idée est de faire de Leaf-03 une stack de deux switchs qui participe au même titre que Leaf-01 et Leaf-02 à la fabric. On garde donc le socle underlay/overlay du lab2, mais on rapproche la couche d'accès du rôle de leaf pour supprimer l'intermédiaire que représentait le design du lab4.

L'implémentation s'appuie sur les principes EVPN Multi-Homing décrits dans la documentation Cisco Catalyst 9300 pour les fabrics BGP EVPN VXLAN :
https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9300/software/release/17-15/configuration_guide/vxlan/b_1715_bgp_evpn_vxlan_9300_cg/configuring_multi_homing_in_bgp_evpn_vxlan_fabric.html

Structure



Le point important du lab5 est que Leaf-03 n'est pas un simple switch d'accès derrière la fabric. C'est une stack de deux membres qui joue le rôle de leaf VXLAN EVPN, avec ses propres uplinks sous-jacents vers les spines et ses propres ports de service pour les VLANs locaux.

Le lab conserve aussi les autres briques du scénario :

- Leaf-01 et Leaf-02 restent les leafs EVPN classiques,
- Spine-01 et Spine-02 assurent l'underlay et le rôle de RR,
- Server-01 reste un endpoint dual-homed pour valider la continuité du service,
- Acces-01 reste un bloc L2 de bordure avec un port-channel trunké sur les VLANs de service.

> On va crée une stack de switchs sans pourtant les connecter physiquement !

Principe du Stack Leaf

Le changement clé par rapport au lab4 est le suivant : on ne met plus la stack d'accès devant la leaf, on la fait devenir la leaf.

Dans le lab4, la redondance était portée par un segment d'accès dual-homé vers la fabric. Dans le lab5, la stack Leaf-03 participe directement à la fabric EVPN/VXLAN et porte elle-même les VLANs de service. La logique n'est donc plus celle d'un réseau d'accès redondé qui remonte vers une leaf, mais celle d'une leaf qui est elle-même construite comme une stack de deux switchs.

Cette approche permet de valider un cas plus proche d'un design de campus réaliste :

- la couche leaf peut être réalisée par une stack physique,
- les ports d'accès sont directement localisés sur cette leaf,
- le VXLAN EVPN transporte les VLANs du site sans couche intermédiaire,
- la fabric reste cohérente même si la leaf elle-même est composée de deux membres.

Ce qui change par rapport à une fabric VXLAN basique sans stack leaf

Par rapport au VXLAN simple du lab2, on conserve le même modèle EVPN/VXLAN, mais Leaf-03 ajoute une particularité matérielle :

- deux switchs sont provisionnés dans la même stack,
- le control plane de leaf est porté par cette stack,
- les liens underlay partent des deux membres de la stack vers les spines,
- les ports de service sont répartis sur les deux plans de la stack,
- le leaf local peut donc jouer à la fois le rôle de VTEP et le rôle de point d'accès.

Le lab5 ne cherche donc pas à introduire un nouveau mécanisme EVPN spécifique. Il montre plutôt qu'une leaf peut être matérialisée par une stack et que cette stack peut porter directement des VLANs de service dans l'overlay.

Configuration appliquée sur Leaf-03

La configuration de Leaf-03 est le cœur du lab5. Les deux membres de la stack sont provisionnés explicitement :

```
switch 1 provision c9300l-24t-4g
switch 2 provision c9300l-24t-4g
```

Ce point est essentiel, car il confirme que la leaf n’est pas un seul châssis logique classique, mais bien une stack de deux switchs. Le reste de la configuration suit ensuite le modèle leaf du lab2 :

```
ip routing
ip multicast-routing

l2vpn evpn
 replication-type static
 router-id Loopback1
 default-gateway advertise

l2vpn evpn instance 101 vlan-based
 encapsulation vxlan
 replication-type static

l2vpn evpn instance 102 vlan-based
 encapsulation vxlan
 replication-type ingress
```

La stack Leaf-03 participe ensuite au transport VXLAN comme les autres leafs :

- `Loopback0` sert d’adresse de voisinage BGP,
- `Loopback1` sert de source NVE,
- `router bgp 65001` forme les sessions vers les spines,
- `interface nve1` porte les VNIs associés aux VLANs de service.

Le sous-système VXLAN reste donc identique dans son principe. La différence est que ce sous-système repose sur une stack à deux membres au lieu d’un seul switch.

Ports d’accès de la stack leaf

Leaf-03 expose directement les VLANs de service sur ses ports locaux. Les interfaces du premier membre transportent principalement le VLAN 101, tandis que celles du second membre

transportent principalement le VLAN 102 :

```
interface GigabitEthernet1/0/2
 switchport access vlan 101
 switchport mode access

...

interface GigabitEthernet1/0/24
 switchport access vlan 101
 switchport mode access

interface GigabitEthernet2/0/2
 switchport access vlan 102
 switchport mode access

...

interface GigabitEthernet2/0/24
 switchport access vlan 102
 switchport mode access
```

Cette répartition est intéressante parce qu'elle montre concrètement le rôle de la stack : chaque membre fournit une partie du plan d'accès, mais l'ensemble reste piloté comme une seule leaf EVPN. On n'a donc pas une stack d'accès interposée entre la fabric et le service ; on a la leaf elle-même qui porte les VLANs.

Couche VLAN et NVE

Les VLANs 101, 102 et 901 sont toujours mappés vers les VNIs du lab :

```
vlan configuration 101
 member evpn-instance 101 vni 10101
vlan configuration 102
 member evpn-instance 102 vni 10102
vlan configuration 901
 member vni 50901
```

Le point à retenir est le même que dans les labs précédents :

- VLAN 101 et VLAN 102 transportent les services du lab,

- VNI 10101 et VNI 10102 font le relais dans l’overlay VXLAN,
- VNI 50901 sert la VRF green,
- `nve1` associe les VNIs à l’adresse de source de la leaf.

La différence avec le lab4 n’est pas dans le mapping VXLAN lui-même, mais dans le fait que ce mapping est directement porté par la stack Leaf-03.

Configuration côté fabric

Leaf-01 et Leaf-02 restent des leafs EVPN classiques. Leur configuration conserve les éléments déjà vus dans les labs précédents :

- `l2vpn evpn` avec le router-id de la loopback,
- les deux voisins BGP vers les spines,
- les VNIs 10101, 10102 et 50901,
- les interfaces `Port-channel12` et `Port-channel14` pour les segments EVPN,
- les ports de service en `channel-group` sur les bons VLANs.

Le lab5 ne modifie pas le rôle de ces leafs. Il ajoute simplement une leaf supplémentaire, Leaf-03, construite comme une stack de deux switchs.

Serveur et accès

Server-01 reste raccordé comme dans les labs précédents. Comme sa configuration ne change pas ici, je ne la redétaille pas dans cette partie ; il sert surtout de point de validation pour confirmer que la fabric reste cohérente pendant que Leaf-03 prend le rôle de leaf stack.

Acces-01 reste, lui, un bloc L2 simple avec un `Port-channel24` trunké sur les VLANs 101 et 102 :

```
interface Port-channel24
 switchport trunk allowed vlan 101,102
 switchport mode trunk
```

Le message de la configuration est simple : le bord de réseau peut toujours exister sous forme d’un switch ou d’une stack L2 classique, mais la leaf EVPN elle-même est désormais représentée par la stack Leaf-03.

Mode de fonctionnement

Le lab5 ne change pas le modèle EVPN de base : la fabric continue à utiliser VXLAN comme data plane et EVPN comme control plane.

Le vrai changement est architectural :

- le lab4 redondait un domaine d’accès devant la leaf,
- le lab5 fait de la stack d’accès la leaf elle-même,
- la stack Leaf-03 transporte donc directement les VLANs du site dans l’overlay.

En pratique, cela simplifie le design du point de vue EVPN : la leaf ne doit plus coordonner un segment intermédiaire de dual-homing comme dans le lab4, parce que c’est sa propre stack qui constitue le point d’attachement à la fabric.

Interaction avec STP

Comme dans les labs précédents, STP ne pilote pas le comportement de la fabric VXLAN EVPN. La résilience et le transport des VLANs passent par le contrôle-plane EVPN et par le comportement de la leaf stack.

STP reste utile comme filet de sécurité dans les domaines L2 locaux, mais il ne remplace pas le rôle de VXLAN/EVPN dans la propagation des services.