Configurer Easy Virtual Network

Table des matières

Introduction

Conditions préalables

- Exigences
- Composants utilisés

Informations générales

Configurer

- Diagramme du réseau
- Configurer le réseau EVN (Easy Virtual Network)
- Régler la ligne principale VNET
  - Liste des lignes principales
  - Attributs des lignes principales par VRF
  - Étiquettes VNET par lien

Vérifier

Dépannage

Informations connexes

Introduction

Ce document décrit la fonctionnalité Easy Virtual Network (EVN), qui fournit un mécanisme de virtualisation simple à configurer dans les réseaux de campus.

Conditions préalables

Exigences

Aucune exigence spécifique n'est associée à ce document.

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions matérielles et logicielles suivantes :

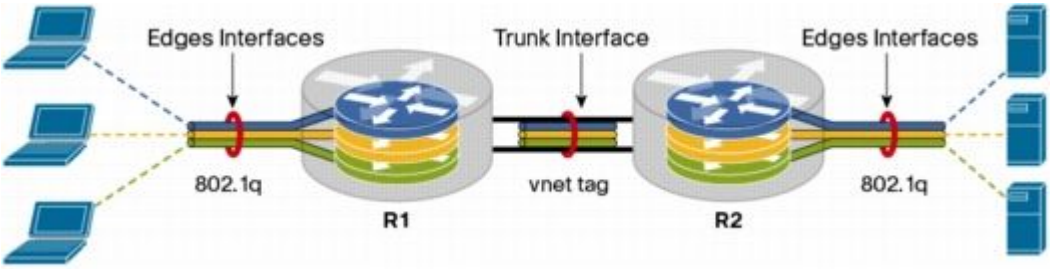
- Commutateurs de la gamme Cisco Catalyst 6000 (Cat6k) qui exécutent le logiciel, version

15.0(1)SY1
- Routeurs de services agrégés de la gamme Cisco 1000 (ASR1000) qui exécutent la version 3.2s du logiciel
- Routeurs ISR (Integrated Services Routers) Cisco 3925 et 3945 qui exécutent les versions 15.3(2)T et ultérieures de Cisco IOS®
- Commutateurs des gammes Cisco Catalyst 4500 (Cat4500) et 4900 (Cat4900) exécutant la version de logiciel 15.1(1)SG

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Informations générales

EVN tire parti des technologies actuelles telles que l'encapsulation VRF-Lite (Virtual Routing and Forwarding-Lite) et dot1q et n'introduit aucun nouveau protocole. Voici un aperçu de la fonctionnalité EVN :



- La fonctionnalité EVN utilise VRF-Lite pour créer plusieurs contextes de routage (jusqu'à 32).
- La connectivité dans le routage et le transfert virtuels (VRF) entre les appareils de couche 3 est assurée par des lignes principales de réseaux virtuels (VNET).
- Les lignes principales du réseau VNET sont des lignes principales normales du réseau dot1q.
- Configurez chaque VRF qui doit être transporté sur les agrégations VNET avec une étiquette VNET.
- Chaque étiquette VNET équivaut à une étiquette dot1q.
- Les sous-interfaces dot1q sont automatiquement créées et masquées.
- La configuration de l'interface principale provient des sous-interfaces (masquées).
- Utilisez des instances distinctes de protocoles de routage dans chaque VRF sur les agrégations VNET afin d'annoncer l'accessibilité du préfixe.
- La fuite du routage dynamique entre des VRF (contrairement au routage statique) est autorisée sans l'utilisation du protocole BGP (Border Gateway Protocol).
- Cette fonctionnalité est prise en charge pour IPv4 et IPv6.

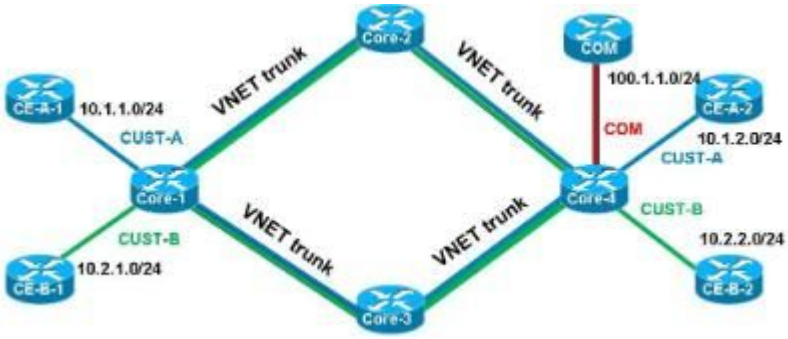
Configurer

Utilisez l’information figurant dans cette section pour configurer la fonctionnalité EVN.

Remarque : Utilisez l'outil Command Lookup Tool (clients enregistrés uniquement) afin d'obtenir plus d'informations sur les commandes utilisées dans cette section.

Diagramme du réseau

Cette configuration de réseau sert à illustrer la configuration de l’EVN et à afficher les commandes suivantes :



Voici des remarques importantes concernant cette configuration :

- Les deux VRF qui y sont définis (CUST-A et CUST-B) sont exécutés à partir du cœur du réseau par les lignes principales du réseau VNET.
- Le protocole OSPF (Open Shortest Path First) est utilisé dans les VRF afin d’annoncer l’accessibilité.
- Le VRF COM héberge un serveur commun (100.1.1.100), qui doit être accessible à partir des VRF CUST-A et CUST-B.
- L’image utilisée est i86bi_linux-adventerprisek9-ms.153-1.S.

Configurer le réseau EVN (Easy Virtual Network)

Voici la marche à suivre pour configurer l’EVN :

1. Configurez la définition du VRF :

```
vrf definition [name]
vnet tag [2-4094]
!
address-family ipv4|ipv6
exit-address-family
!
```

Remarques importantes sur cette configuration :

- Utilisez des balises comprises entre 2 et 1 000. N'utilisez pas les VLAN réservés 1 001 à 1 005. Utilisez les VLAN étendus 1 006 à 4 094 si nécessaire.
- N’autorisez pas l’utilisation de l’étiquette VNET par un VLAN actuel.
- Assurez-vous que les étiquettes VNET sont identiques sur tous les périphériques pour un VRF donné.
- Configurez la famille d'adresses ipv4|ipv6 pour activer le VRF dans l'AF associé.
- Il n’est pas nécessaire de définir une direction de routage (RD), car l’EVN n’utilise pas BGP.

Avec cette configuration, les VRF sont définis sur tous les routeurs principaux 4x. Par exemple, sur CORE-1 :

```
vrf definition CUST-A
   vnet tag 100
   !
   address-family ipv4
   exit-address-family
  vrf definition CUST-B
   vnet tag 200
   !
   address-family ipv4
   exit-address-family
```

Utilisez la même étiquette VNET sur tous les routeurs pour ces VRF. Sur CORE-4, VRF COM ne requiert aucune étiquette VNET. L’objectif est de conserver ce VRF local sur CORE-4 et de configurer la fuite et la redistribution afin de fournir un accès au serveur commun à partir de CUST-A et CUST-B.

Saisissez cette commande pour vérifier différents compteurs VNET :

<#root>

```
CORE-1#
```

show vnet counters

```
  Maximum number of VNETs supported: 32
  Current number of VNETs configured: 2
```

```
  Current number of VNET trunk interfaces: 2
  Current number of VNET subinterfaces: 4
  Current number of VNET forwarding interfaces: 6
  CORE-1#
```

2. Configurez la ligne principale VNET :

```
interface GigabitEthernetx/x
  vnet trunk
  ip address x.x.x.x y.y.y.y
  ...
```

Voici des remarques importantes concernant cette configuration :

- La commande vnet trunk crée autant de sous-interfaces dot1q que le nombre de VRF définis avec une étiquette VNET.
- La commande vnet trunk ne peut pas coexister avec certaines sous-interfaces configurées manuellement sur la même interface physique.
- Cette configuration est autorisée sur les interfaces routées (et non sur les ports de commutation) physiques et sur le port-channel.
- Les adresses IP (et autres commandes) qui sont appliquées à l'interface physique proviennent des sous-interfaces.
- Les sous-interfaces des VRF utilisent la même adresse IP.

Avec cette configuration, deux VRF VNET sont offerts, et donc deux sous-interfaces sont créées automatiquement sur l'interface configurée comme ligne principale VNET. Vous pouvez entrer la commande show derived-config pour voir la configuration cachée qui est créée automatiquement :

Voici la configuration en cours d'utilisation :

<#root>

```
CORE-1#
```

show run | s Ethernet0/0

```
interface Ethernet0/0
 vnet trunk
 ip address 192.168.1.1 255.255.255.252
 !
CORE-1#
```

Voici la configuration obtenue :

<#root>

CORE-1#

show derived-config | s Ethernet0/0

```
interface Ethernet0/0
 vnet trunk
 ip address 192.168.1.1 255.255.255.252
Interface Ethernet0/0.100
 description Subinterface for VNET CUST-A
 encapsulation dot1Q 100
 vrf forwarding CUST-A
 ip address 192.168.1.1 255.255.255.252
interface Ethernet0/0.200
 description Subinterface for VNET CUST-B
 encapsulation dot1Q 200
 vrf forwarding CUST-B
 ip address 192.168.1.1 255.255.255.252
CORE-1#
```

Comme il est indiqué, toutes les sous-interfaces proviennent de l'adresse IP de l'interface principale.

3. Affectez les [sous-]interfaces de périphérie aux VRF. Afin d'affecter une interface ou une sous-interface à un VRF VNET, utilisez la même procédure que celle utilisée pour affecter un VRF normalement :

```
interface GigabitEthernet x/x.y
 vrf forwarding [name]
  ip address x.x.x.x y.y.y.y
  ...
```

Avec cette configuration, la configuration est appliquée sur CORE-1 et CORE-4. Voici un exemple pour CORE-4 :

```
interface Ethernet2/0
   vrf forwarding CUST-A
   ip address 10.1.2.1 255.255.255.0
  !
  interface Ethernet3/0
   vrf forwarding CUST-B
   ip address 10.2.2.1 255.255.255.0
  !
  interface Ethernet4/0
   vrf forwarding COM
   ip address 100.1.1.1 255.255.255.0
```

4. Configurez les protocoles de routage pour chaque VRF (cette configuration n’est pas propre à EVN ou à VNET) :

<#root>

```
router ospf x vrf [name]
```

```
network x.x.x.x y.y.y.y area x
```

```
  ...
```


Remarque : Cette configuration doit inclure les adresses d’agrégation VNET ainsi que les adresses d’interface de périphérie.

Grâce à cette configuration, deux processus OSPF sont définis, soit un par VRF :

<#root>

```
CORE-1#
```

```
show run | s router os
```

```
  router ospf 1 vrf CUST-A
   network 10.1.1.0 0.0.0.255 area 0
   network 192.168.1.0 0.0.0.255 area 0
  router ospf 2 vrf CUST-B
   network 10.2.1.0 0.0.0.255 area 0
   network 192.168.1.0 0.0.0.255 area 0
  CORE-1#
```

Vous pouvez entrer le mode de contexte de routage afin de visualiser les informations relatives à un VRF en particulier, sans les spécifications du VRF dans chaque commande :

<#root>

```
CORE-1#
```

```
routing-context vrf CUST-A
```

```
  CORE-1%CUST-A#
  CORE-1%CUST-A#
```

```
show ip protocols
```

```
  *** IP Routing is NSF aware ***

  Routing Protocol is "ospf 1"
    Outgoing update filter list for all interfaces is not set
    Incoming update filter list for all interfaces is not set
    Router ID 192.168.1.13
    It is an area border router
    Number of areas in this router is 1. 1 normal 0 stub 0 nssa
    Maximum path: 4
    Routing for Networks:
      10.1.1.0 0.0.0.255 area 0
      192.168.1.0 0.0.0.255 area 0
    Routing Information Sources:
      Gateway         Distance      Last Update
      192.168.1.9          110      1d00h
      192.168.1.14         110      1d00h
    Distance: (default is 110)

  CORE-1%CUST-A#

  CORE-1%CUST-A#
```

show ip os neighbor

```
Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.14      1   FULL/DR         00:00:30    192.168.1.14    Ethernet1/0.100
192.168.1.5       1   FULL/BDR        00:00:37    192.168.1.2     Ethernet0/0.100
10.1.1.2          1   FULL/BDR        00:00:33    10.1.1.2        Ethernet2/0
CORE-1%CUST-A#
```

Remarque : La sortie de la commande show ip protocols affiche uniquement les informations relatives au VRF sélectionné.

Lorsque vous affichez la base d'informations de routage (RIB) pour les deux VRF, vous pouvez vérifier le sous-réseau distant au moyen des deux lignes principales VNET :

<#root>

CORE-1%CUST-A#

show ip route 10.1.2.0

```
  Routing Table: CUST-A
  Routing entry for 10.1.2.0/24
    Known via "ospf 1", distance 110, metric 30, type intra area
    Last update from 192.168.1.2 on Ethernet0/0.100, 1d00h ago
    Routing Descriptor Blocks:
    * 192.168.1.14, from 192.168.1.9, 1d00h ago, via Ethernet1/0.100
        Route metric is 30, traffic share count is 1
```

```
      192.168.1.2, from 192.168.1.9, 1d00h ago, via Ethernet0/0.100
        Route metric is 30, traffic share count is 1
  CORE-1%CUST-A#
  CORE-1%CUST-A#routing-context vrf CUST-B
  CORE-1%CUST-B#
  CORE-1%CUST-B#
```

show ip route 10.2.2.0

```
  Routing Table: CUST-B
  Routing entry for 10.2.2.0/24
    Known via "ospf 2", distance 110, metric 30, type intra area
    Last update from 192.168.1.2 on Ethernet0/0.200, 1d00h ago
    Routing Descriptor Blocks:
    * 192.168.1.14, from 192.168.1.6, 1d00h ago, via Ethernet1/0.200
        Route metric is 30, traffic share count is 1
      192.168.1.2, from 192.168.1.6, 1d00h ago, via Ethernet0/0.200
        Route metric is 30, traffic share count is 1
  CORE-1%CUST-B#
  CORE-1%CUST-B#exit
  CORE-1#
  CORE-1#
```

5. Déterminez la fuite de routage entre les VRF. La fuite de routage est réalisée par la réplication du routage. Par exemple, certaines routes dans un VRF peuvent être mises à disposition pour un autre VRF :

<#root>

vrf definition

VRF-X

```
  address-family ipv4|ipv6
   route-replicate from vrf
```

VRF-Y

unicast|

multicast

```
    [
```

route-origin

] [route-map [name]]

Voici des remarques importantes concernant cette configuration :

- La RIB pour VRF-X a accès aux routages sélectionnés en fonction des paramètres de commande de VRF-Y.
- Les routages répliqués dans VRF-X sont indiqués par un indicateur [+].

- L’option de multidiffusion (multicast) permet l’utilisation des routages d’un autre VRF pour le transfert RPF (Reverse Path Forwarding).
- L’option route-origin peut avoir une des valeurs suivantes :
    - tout
    - bgp
    - connected
    - eigrp
    - isis
    - mobile
    - odr
    - ospf
    - rip
    - static

Malgré le nom, la réplication de route ne réplique pas ou ne duplique pas les routes. Il s'agit d'une fuite normale via BGP common RT et elle ne consomme pas de mémoire supplémentaire.

Avec cette configuration, la fuite de routage est utilisée sur CORE-4 afin de fournir un accès de CUST-A et CUST-B à COM (et inversement) :

<#root>

```
vrf definition CUST-A
  address-family ipv4
     route-replicate from vrf COM unicast connected
!
vrf definition CUST-B
  address-family ipv4
     route-replicate from vrf COM unicast connected
!
vrf definition COM
  address-family ipv4
     route-replicate from vrf CUST-A unicast ospf 1 route-map USERS
     route-replicate from vrf CUST-B unicast ospf 2 route-map USERS
!
route-map USERS permit 10
  match ip address prefix-list USER-SUBNETS
!
ip prefix-list USER-SUBNETS seq 5 permit 10.0.0.0/8 le 32

CORE-4#
```

show ip route vrf CUST-A

```
Routing Table: COM
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area,* - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
```

```
       a - application route

+ - replicated route, % - next hop override


...
       10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O        10.1.1.0/24 [110/30] via 192.168.1.10, 3d19h, Ethernet1/0.100
           [110/30] via 192.168.1.5, 3d19h, Ethernet0/0.100
       100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C   +    100.1.1.0/24 is directly connected (COM), Ethernet4/0



CORE-4#
show ip route vrf CUST-B


...
       10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O        10.2.1.0/24 [110/30] via 192.168.1.10, 1d00h, Ethernet1/0.200
           [110/30] via 192.168.1.5, 1d00h, Ethernet0/0.200
       100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C   +    100.1.1.0/24 is directly connected (COM), Ethernet4/0



CORE-4#
show ip route vrf COM


...
       10.0.0.0/24 is subnetted, 2 subnets
O   +    10.1.1.0 [110/30] via 192.168.1.10 (CUST-A), 3d19h, Ethernet1/0.100
           [110/30] via 192.168.1.5 (CUST-A), 3d19h, Ethernet0/0.100
O   +    10.2.1.0 [110/30] via 192.168.1.10 (CUST-B), 1d00h, Ethernet1/0.200
           [110/30] via 192.168.1.5 (CUST-B), 1d00h, Ethernet0/0.200


       100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        100.1.1.0/24 is directly connected, Ethernet4/0
```

À ce stade, les routages répliqués ne sont pas envoyés dans le protocole IGP (Interior Gateway Protocol). Par conséquent, seuls CE-A-2 et CE-B-2 ont accès au service COM (100.1.1.100), et non CE-A-1 et CE-B-1.

Vous pouvez également utiliser une fuite de routage en provenance ou à destination du tableau global :

<#root>

```
vrf definition VRF-X
```

```
 address-family ipv4
  route-replicate from vrf >
```

global

```
 unicast|multicast [route-origin]
   [route-map [name]]
  exit-address-family
 !
 exit
!
```

global-address-family ipv4 unicast

```
 route-replicate from vrf [vrf-name] unicast|multicast [route-origin]
  [route-map [name]]
```

6. Définissez la propagation de la fuite de routage. Les fuites de routages ne sont pas dupliquées dans la RIB du VRF cible. Autrement dit, ils ne font pas partie de la RIB du VRF cible. La redistribution normale entre les processus du routeur ne fonctionne pas. Vous devez donc définir explicitement la connexion VRF de la RIB à laquelle le routage appartient :

<#root>

```
router ospf x vrf VRF-X
  redistribute
```

vrf VRF-Y

```
 [route-origin] [route-map [name]]
```

Les fuites de routage provenant de VRF-Y sont redistribuées dans le processus OSPF qui est utilisé dans VRF-X. Voici un exemple sur CORE-4 :

```
router ospf 1 vrf CUST-A
   redistribute vrf COM connected subnets route-map CON-2-OSPF
!
route-map CON-2-OSPF permit 10
 match ip address prefix-list COM
!
ip prefix-list COM seq 5 permit 100.1.1.0/24
```

La route-map n'est pas nécessaire dans ce cas, car il n'y a qu'une seule route connectée dans VRF COM. Il y a maintenant l'accessibilité au service COM (100.1.1.100) à partir de CE-A-1 et CE-B-1 :

```
<#root>

CE-A-1#

ping 100.1.1.100


  Type escape sequence to abort.
  Sending 5, 100-byte ICMP Echos to 100.1.1.100, timeout is 2 seconds:
  !!!!!
  Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
  CE-A-1#

CE-B-1#

ping 100.1.1.100


  Type escape sequence to abort.
  Sending 5, 100-byte ICMP Echos to 100.1.1.100, timeout is 2 seconds:
  !!!!!
  Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
  CE-B-1#
```

Régler la ligne principale VNET

Cette section fournit de l'information que vous pouvez utiliser pour régler la ligne principale VNET.

Liste des lignes principales

Par défaut, tous les VRF configurés avec une étiquette VNET sont autorisés sur l'ensemble des lignes principales VNET. Une liste des lignes principales vous permet de préciser la liste des VRF autorisés sur la ligne principale VNET :

```
<#root>

vrf list [list-name]


member [vrf-name]


!
interface GigabitEthernetx/x
  vnet trunk

list [list-name]
```

Remarque : Utilisez une ligne par VRF autorisé.

Par exemple, CORE-1 est réglé pour le VRF CUST-B sur la ligne principale VNET entre CORE-1 et CORE-2 :

```
vrf list TEST
 member CUST-A
!
interface ethernet0/0
 vnet trunk list TEST
```

Comme il est illustré, l'homologation OSPF pour le VRF CUST-B dans l'ensemble de la ligne principale est interrompue :

<#root>

```
%OSPF-5-ADJCHG: Process 2, Nbr 192.168.1.2 on Ethernet0/0.200 from FULL to DOWN,
 Neighbor Down: Interface down or detached
```

La sous-interface pour le VRF CUST-B est supprimée :

<#root>

```
CORE-1#

show derived-config | b Ethernet0/0


  interface Ethernet0/0
   vnet trunk list TEST
   ip address 192.168.1.1 255.255.255.252
  !
  interface Ethernet0/0.100
   description Subinterface for VNET CUST-A
   encapsulation dot1Q 100
   vrf forwarding CUST-A
   ip address 192.168.1.1 255.255.255.252
  !
```

Attributs des lignes principales par VRF

Par défaut, les sous-interfaces dot1q obtiennent les paramètres de l'interface physique, de sorte

que les sous-interfaces des VRF ont les mêmes attributs (p. ex., le coût et l'authentification). Vous pouvez régler les paramètres de la ligne principale par étiquette VNET :

```
interface GigaEthernetx/x
  vnet trunk
  vnet name VRF-X
   ip ospf cost 100
  vnet name VRF-Y
   ip ospf cost 15
```

Vous pouvez régler ces paramètres :

<#root>

```
CORE-1(config-if-vnet)#?
      Interface VNET instance override configuration commands:

bandwidth

     Set bandwidth informational parameter
        default       Set a command to its defaults

delay

         Specify interface throughput delay
        exit-if-vnet  Exit from VNET submode

ip

            Interface VNET submode Internet Protocol config commands
        no            Negate a command or set its defaults
        vnet          Configure protocol-independent VNET interface options
      CORE-1(config-if-vnet)#

      CORE-1(config-if-vnet)#ip ?
        authentication      authentication subcommands
        bandwidth-percent   Set EIGRP bandwidth limit
        dampening-change    Percent interface metric must change to cause update
        dampening-interval  Time in seconds to check interface metrics
        hello-interval      Configures EIGRP-IPv4 hello interval
        hold-time           Configures EIGRP-IPv4 hold time
        igmp                IGMP interface commands
        mfib                Interface Specific MFIB Control
        multicast           IP multicast interface commands
        next-hop-self       Configures EIGRP-IPv4 next-hop-self
        ospf                OSPF interface commands
        pim                 PIM interface commands
        split-horizon       Perform split horizon
        summary-address     Perform address summarization
```

```
        verify              Enable per packet validation

      CORE-1(config-if-vnet)#ip
```

Dans cet exemple, le coût OSPF par VRF pour CORE-1 est modifié, de sorte que le chemin CORE-2 est utilisé pour CUST-A et le chemin CORE-3 pour CUST-B (le coût par défaut est 10

<#root>

```
interface Ethernet0/0
  vnet name CUST-A
    ip ospf cost 8
  !
  vnet name CUST-B
    ip ospf cost 12
  !

CORE-1#

show ip route vrf CUST-A 10.1.2.0



Routing Table: CUST-A
Routing entry for 10.1.2.0/24
 Known via "ospf 1", distance 110, metric 28, type intra area
 Last update from 192.168.1.2 on Ethernet0/0.100, 00:05:42 ago
 Routing Descriptor Blocks:
 * 192.168.1.2, from 192.168.1.9, 00:05:42 ago, via Ethernet0/0.100
   Route metric is 28, traffic share count is 1
CORE-1#

CORE-1#

show ip route vrf CUST-B 10.2.2.0

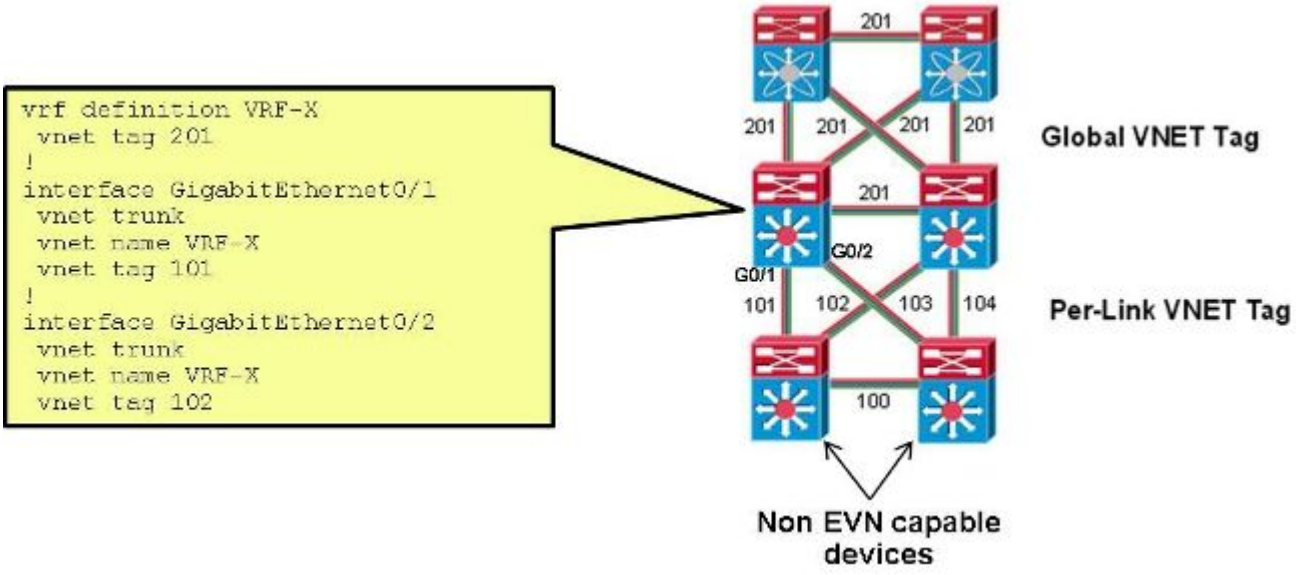


Routing Table: CUST-B
Routing entry for 10.2.2.0/24
 Known via "ospf 2", distance 110, metric 30, type intra area
 Last update from 192.168.1.14 on Ethernet1/0.200, 00:07:03 ago
 Routing Descriptor Blocks:
 * 192.168.1.14, from 192.168.1.6, 1d18h ago, via Ethernet1/0.200
   Route metric is 30, traffic share count is 1
CORE-1#
```

Étiquettes VNET par lien

Par défaut, l'étiquette VNET indiquée dans la définition du VRF est utilisée pour toutes les lignes principales. Cependant, vous pouvez utiliser une autre étiquette VNET par ligne principale.

Cet exemple décrit un scénario dans lequel vous êtes connecté à un périphérique non compatible avec EVN et utilisez VRF-Lite avec une ligne principale manuelle. Aussi, l'étiquette VNET globale est employée par un autre VLAN :



Avec cette configuration, l'étiquette VNET utilisée sur la ligne principale entre CORE-1 et CORE-2 pour CUST-A passe de 100 à 101 :

<#root>

```
interface Ethernet0/0
  vnet name CUST-A
   vnet tag
```

101

Une fois cette modification effectuée sur CORE-1, une nouvelle sous-interface est créée :

<#root>

```
CORE-1#
```

show derived-config | b Ethernet0/0

```
interface Ethernet0/0
 vnet trunk
 ip address 192.168.1.1 255.255.255.252
!
```

interface Ethernet0/0.101

```
 description Subinterface for VNET CUST-A
 encapsulation dot1Q 101
 vrf forwarding CUST-A
 ip address 192.168.1.1 255.255.255.252
!
interface Ethernet0/0.200
 description Subinterface for VNET CUST-B
 encapsulation dot1Q 200
 vrf forwarding CUST-B
 ip address 192.168.1.1 255.255.255.252
```

Si ce changement se produit seulement à une extrémité, la connectivité est alors perdue dans le VRF associé, et le protocole OSPF est interrompu :

<#root>

```
%OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.5 on Ethernet0/0.101 from FULL to DOWN,
 Neighbor Down: Dead timer expired
```

Lorsque la même étiquette VNET est utilisée sur CORE-2, la connectivité est restaurée, et l'étiquette dot1q 101 est utilisée sur la ligne principale, tandis que 100 est toujours utilisée sur la ligne principale CORE-1 à CORE-3 :

<#root>

```
%OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.5 on Ethernet0/0.101 from LOADING to
 FULL, Loading Done
```

Vérifier

Aucune procédure de vérification n'est disponible pour cette configuration.

Dépannage

Il n'existe actuellement aucune information de dépannage spécifique pour cette configuration.

Informations connexes

- Réseau virtuel simplifié : virtualisation de réseau de couche 3
- Assistance et documentation techniques - Cisco Systems

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.