

Exemple virtuel facile de configuration réseau

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Informations générales](#)

[Configurez](#)

[Diagramme du réseau](#)

[Configurez EVN](#)

[Accordez le joncteur réseau VNET](#)

[Liste de joncteur réseau](#)

[Attributs de joncteur réseau de Par-VRF](#)

[Balises du Par-lien VNET](#)

[Vérifiez](#)

[Dépannez](#)

[Informations connexes](#)

Introduction

Ce document décrit la caractéristique facile du réseau virtuel (EVN), qui est conçue afin de fournir un facile, simple-à-configurer le mécanisme de virtualisation dans les réseaux campus. Il accroît des Technologies en cours, telles que le routage virtuel et l'Expédition-Lite (Vrf-Lite) et l'encapsulation dot1q, et n'introduit pas n'importe quel nouveau protocole.

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

[Composants utilisés](#)

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Cisco Catalyst 6000 gammes (Cat6k) commute cette version de logiciel 15.0(1)SY1 de

passage

- La gamme Cisco 1000 a agrégé les Routeurs de services (ASR1000) cette version de logiciel 3.2s de passage
- Integrated Services Router de gammes Cisco 3925 et 3945 qui exécutent des versions 15.3(2)T et ultérieures de Cisco IOS®
- Cisco Catalyst 4500 (Cat4500) et 4900 Commutateurs de la gamme (Cat4900) qui exécutent la version de logiciel 15.1(1)SG

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Informations générales

Voici un aperçu de la caractéristique EVN :

- La caractéristique EVN emploie Vrf-Lite afin de créer plusieurs contextes de routage (jusqu'à de 32).
- La Connectivité dans le Virtual Routing and Forwarding (VRF) entre les périphériques Layer3 est assurée par l'intermédiaire des joncteurs réseau du réseau virtuel (VNET).
- Les joncteurs réseau VNET sont les joncteurs réseau réguliers dot1q.
- Chaque VRF qui doit être transporté à travers les joncteurs réseau VNET devrait être configuré avec une balise VNET.
- Chaque balise VNET égale une balise dot1q.
- **Les sous-interfaces dot1q sont automatiquement créées et masquées.**
- **La configuration de l'interface principale est héritée par toutes les sous-interfaces (masquées).**
- Des exemples distincts des protocoles de routage devraient être utilisés dans chaque VRF au-dessus des joncteurs réseau VNET afin d'annoncer l'accessibilité de préfixe.
- On permet l'artère dynamique coulant entre les vrf (opposés aux artères statiques) sans utilisation de Protocole BGP (Border Gateway Protocol).
- La caractéristique est prise en charge pour l'ipv4 et l'IPv6.

Configurez

Utilisez les informations qui sont décrites dans cette section afin de configurer la caractéristique EVN.

Remarque: Utilisez l'[Outil de recherche de commande](#) (clients [enregistrés](#) seulement) pour obtenir plus d'informations sur les commandes utilisées dans cette section.

Diagramme du réseau

Cette configuration réseau est utilisée afin d'illustrer la configuration et des commandes show EVN :

Voici quelques informations importantes au sujet de cette installation :

- Deux vrf sont définis (**CUST-A** et **CUST-B**) qui sont effectués du coeur du réseau par des joncteurs réseau VNET.
- Le Protocole OSPF (Open Shortest Path First) est utilisé dans les vrf afin d'annoncer l'accessibilité.
- COM de VRF héberge un serveur commun (**100.1.1.100**) qui doit être accessible du VRF CUST-A et du CUST-B.
- L'image qui est utilisée est **i86bi_linux-adventerprisek9-ms.153-1.S**.

Conseil : Le Cisco IOS sur le Linux (IOL) a installé qui est utilisé est disponible [ici](#).

Configurez EVN

Terminez-vous ces étapes afin de configurer la caractéristique EVN :

1. Configurez le vrf definition :

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

!Voici quelques informations importantes au sujet de cette configuration :

Cisco recommande que vous utilisiez des balises de l'ordre de 2 à 1,000. N'utilisez pas les VLAN réservés 1,001 à 1,005. Les VLAN étendus 1,006 à 4,094 peuvent être utilisés, si nécessaires.

La balise VNET ne devrait pas être utilisée par un courant VLAN.

Les balises VNET devraient être identiques sur tous les périphériques pour n'importe quel VRF donné.

L'adresse-family ipv4|ipv6 devrait être configurée afin de lancer le VRF dans l'AF relatif.

Il n'y a aucun besoin de définir une direction d'artère (RD) parce qu'EVN n'utilise pas le BGP. Avec cette installation, les vrf devraient être définis sur tous les Routeurs du noyau 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 balise VNET sur tous les Routeurs pour ces vrf. Sur CORE-4, COM de VRF n'a pas besoin d'une balise VNET. Le but est de garder que des gens du pays de VRF sur CORE-4 et configure la fuite et la redistribution afin de permettre d'accéder au serveur commun de CUST-A et de CUST-B.

Sélectionnez cette commande afin de vérifier de divers compteurs VNET :

```
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 le joncteur réseau VNET :

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

... Voici quelques informations importantes au sujet de cette configuration :

La commande de **joncteur réseau de vnet** crée autant de sous-interfaces dot1q comme nombre de vrf qui sont définis avec une balise VNET.

La commande de **joncteur réseau de vnet** ne peut pas coexister avec quelques sous-interfaces manuel-configurées sur la même interface physique.

On permet cette configuration sur les interfaces conduites (pas ports de commutateur), l'examen médical et le portchannel.

Les adresses IP (et d'autres commandes) qui sont appliqués sur l'interface physique sont héritées par les sous-interfaces.

Les sous-interfaces pour tous les vrf utilisent la même adresse IP.

Avec cette installation, il y a deux vrf VNET, ainsi deux sous-interfaces sont automatiquement créées sur l'interface qui est configurée comme joncteur réseau VNET. Vous pouvez sélectionner la commande de **show derived-config** afin de visualiser la configuration masquée qui est automatiquement créée :

Voici la configuration qui fonctionne actuellement :

```
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 dérivée :

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 affiché, toutes les sous-interfaces héritent de l'adresse IP de l'interface principale.

3. Assignez les sous) interfaces de périphérie (aux vrf. Afin d'assigner une interface ou une sous-interface à un VRF VNET, utilisez la même procédure que cela utilisée afin d'assigner un VRF normalement :

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

...Avec cette installation, 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 (ce n'est pas spécifique à EVN ou à VNET) :

```
router ospf x vrf [name]
  network x.x.x.x y.y.y.y area x
```

...Remarque: Cette configuration devrait inclure les adresses de joncteur réseau VNET aussi bien que les adresses d'interface de périphérie.

Avec cette installation, deux processus OSPF sont définis, un par VRF :

```
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 routage-contexte afin de visualiser les informations qui sont liées à un VRF spécifique sans caractéristiques de VRF dans chaque commande :

```
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 commande de **show ip protocols** affiche seulement les informations qui sont liées au VRF sélectionné.

Quand vous visualisez le Routing Information Base (NERVURE) pour les deux vrf, vous pouvez vérifier le sous-réseau distant par l'intermédiaire des deux joncteurs réseau VNET :

```

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 l'artère coulant entre les vrf. La fuite d'artère est exécutée par l'intermédiaire de la réplication d'artère. Par exemple, quelques artères dans un VRF pourraient être rendues disponibles pour un autre VRF :

```

vrf definition VRF-X
  address-family ipv4|ipv6
    route-replicate from vrf VRF-Y unicast|multicast

```

[route-origin] [route-map [name]] Voici quelques informations importantes au sujet de cette configuration :

La NERVURE pour **VRF-X** a accès aux routes sélectionnées, basées sur des paramètres de commande de **VRF-Y**.

Les artères répliquées dans **VRF-X** sont identifiées par l'indicateur a [+].

L'option de **Multidiffusion** permet l'utilisation des artères d'un autre VRF pour le Reverse Path Forwarding (RPF).

L'artère-origine peut avoir une de ces valeurs :

tousBGPconnectéeigrpISISmobileodrOSPFdéchirurestatique

À la différence du nom indique, les artères ne sont pas répliquées ou sont reproduites ; c'est le cas avec la fuite normale par BGP droite commune, qui ne consomme pas la mémoire supplémentaire.

Avec cette installation, la fuite d'artère est utilisée sur CORE-4 afin de fournir l'accès de CUST-A et de CUST-B à COM (et vice-versa) :

```
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

```

En ce moment, les artères répliquées ne sont propagées dans le Protocole IGP (Interior Gateway Protocol), ainsi seulement le CE-A-2 et le CE-B-2 ont accès au service COM (100.1.1.100), pas au CE-A-1 et au CE-B-1.

Vous pouvez également utiliser l'artère coulant ou derrière une table globale :

```

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 disjointe d'artère. Les artères coulées ne sont pas reproduites dans la NERVURE de VRF de cible. En d'autres termes, ils ne sont pas une partie de la NERVURE de VRF de cible. La redistribution normale entre les processus de routeur ne fonctionne pas, ainsi vous devez explicitement définir la connexion de VRF de la NERVURE à laquelle l'artère appartient :

```

router ospf x vrf VRF-X

```

redistribute **vrf VRF-Y** [route-origin] [route-map [name]] Les artères coulées de VRF-Y sont redistribuées dans le processus OSPF qui fonctionne 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 Le route-map n'est pas nécessaire dans ce cas, puisqu'il y a seulement une route connectée dans COM de VRF. Il y a maintenant d'accessibilité au service COM (100.1.1.100) de CE-A-1 et de CE-B-1 :

```

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#
```

Accordez le joncteur réseau VNET

Cette section fournit les informations que vous pouvez employer afin d'accorder le joncteur réseau VNET.

Liste de joncteur réseau

Par défaut, on permet tous les vrf qui sont configurés avec une balise VNET sur tous les joncteurs réseau VNET. Une liste de joncteur réseau te permet pour spécifier la liste de vrf autorisés sur le joncteur réseau VNET :

```
vrf list [list-name]
  member [vrf-name]
!
interface GigabitEthernetx/x
  vnet trunk list [list-name]
```

Remarque: Il devrait y avoir une ligne par VRF permis.

Comme exemple, CORE-1 est accordé pour le VRF CUST-B sur le joncteur réseau VNET entre CORE-1 et CORE-2 :

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

Comme affiché, l'OSPF scrutant pour le VRF CUST-B à travers le joncteur réseau descend :

```
%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 retirée :

```
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 de joncteur réseau de Par-VRF

Par défaut, les sous-interfaces dot1q héritent des paramètres de l'interface physique de sorte que les sous-interfaces pour tous les vrf aient les mêmes attributs (tels que le coût et l'authentification). Vous pouvez accorder les paramètres de jonction par balise VNET :

```

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

```

Vous pouvez accorder ces paramètres :

```

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 le coût OSPF par VRF pour CORE-1 est changé, ainsi le chemin CORE-2 est utilisé pour CUST-A et le chemin CORE-3 pour CUST-B (le coût par défaut est 10) :

```

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#

Balises du Par-lien VNET

Par défaut, la balise VNET qui est définie dans le vrf definition est utilisée pour tous les joncteurs réseau. Cependant, vous pouvez utiliser une balise différente VNET par joncteur réseau.

Cet exemple décrit un scénario où vous êtes connecté à un périphérique capable de non-EVN et vous utilisez Vrf-Lite avec un joncteur réseau manuel, et la balise globale VNET est utilisée par un autre VLAN :

Avec cette installation, la balise VNET qui est utilisée sur le joncteur réseau entre CORE-1 et CORE-2 pour CUST-A est changée de **100** à **101** :

```
interface Ethernet0/0
vnet name CUST-A
  vnet tag 101
```

Après que cette modification se produise sur CORE-1, une nouvelle sous-interface est créée :

```
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 cette modification se produit seulement sur une extrémité, alors la Connectivité est perdue dans le VRF associé et l'OSPF descend :

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

Une fois que la même balise VNET est utilisée sur CORE-2, la Connectivité est restaurée et la balise **101** dot1q est utilisée sur ce joncteur réseau tandis que **100** est encore utilisés sur le CORE-1 au joncteur réseau CORE-3 :

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

Vérifiez

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

Dépannez

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

Informations connexes

- [Réseau virtuel facile - Simplification de la virtualisation de réseau de la couche 3](#)
- [Support et documentation techniques - Cisco Systems](#)