

Ejemplo de configuración fácil de la red virtual

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Configurar](#)

[Diagrama de la red](#)

[Configuración EVN](#)

[Ajuste el trunk VNET](#)

[Lista del trunk](#)

[Atributos del trunk Por-VRF](#)

[Etiquetas del Por-link VNET](#)

[Verificación](#)

[Troubleshooting](#)

[Información Relacionada](#)

Introducción

Este documento describe la característica fácil de la red virtual (EVN), que se diseña para proporcionar un fácil, mecanismo de la virtualización de la simple-a-configuración en las redes de oficinas centrales. Leverages las tecnologías actuales, tales como encapsulación del ruteo virtual y de Expedición-Lite (VRF-Lite) y del dot1q, y no introduce ningún nuevo protocolo.

Prerrequisitos

Requisitos

No hay requisitos específicos para este documento.

Componentes Utilizados

La información que contiene este documento se basa en estas versiones de software y hardware.

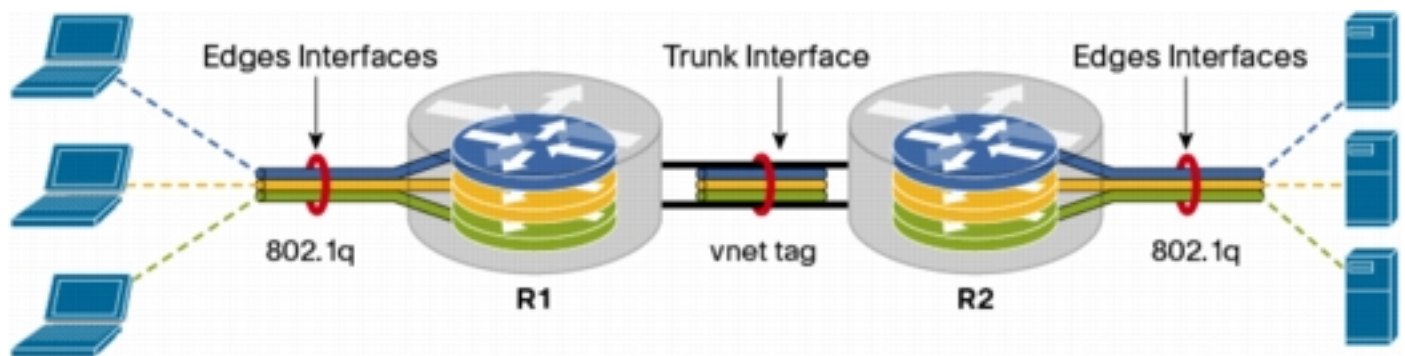
- Cisco Catalyst 6000 switches de la serie (Cat6k) que funcionan con la versión de software 15.0(1)SY1

- Las Cisco 1000 Series agregaron al Routers de los servicios (ASR1000) esa versión de software 3.2s del funcionamiento
- Routers de los Servicios integrados de las Cisco y Series que funciona con las versiones 15.3(2)T del [®] del Cisco IOS y posterior
- Switches de la serie (Cat4900) del Cisco Catalyst 4500 (Cat4500) y 4900 que funcionan con la versión de software 15.1(1)SG

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Antecedentes

Aquí está una descripción de la característica EVN:



- La característica EVN utiliza VRF-Lite para crear varios contextos de ruteo (de hasta 32).
- La Conectividad dentro del ruteo virtual y de la expedición (VRF) entre los dispositivos Layer3 se asegura vía los trunks de la red virtual (VNET).
- Los trunks VNET son trunks regulares del dot1q.
- Cada VRF que se debe transportar a través de los trunks VNET se debe configurar con una etiqueta VNET.
- Cada etiqueta VNET iguala una etiqueta del dot1q.
- **Las subinterfaces del dot1q se crean y se ocultan automáticamente.**
- **La configuración de la interfaz principal es heredada por todas las subinterfaces del (hidden).**
- Los casos distintos de Routing Protocol se deben utilizar en cada VRF sobre los trunks VNET para hacer publicidad del accesibilidad del prefijo.
- La ruta dinámico que se escapa entre los VRF (opuestos a las Static rutas) se permite sin el

uso del Border Gateway Protocol (BGP).

- La característica se soporta para el IPv4 y el IPv6.

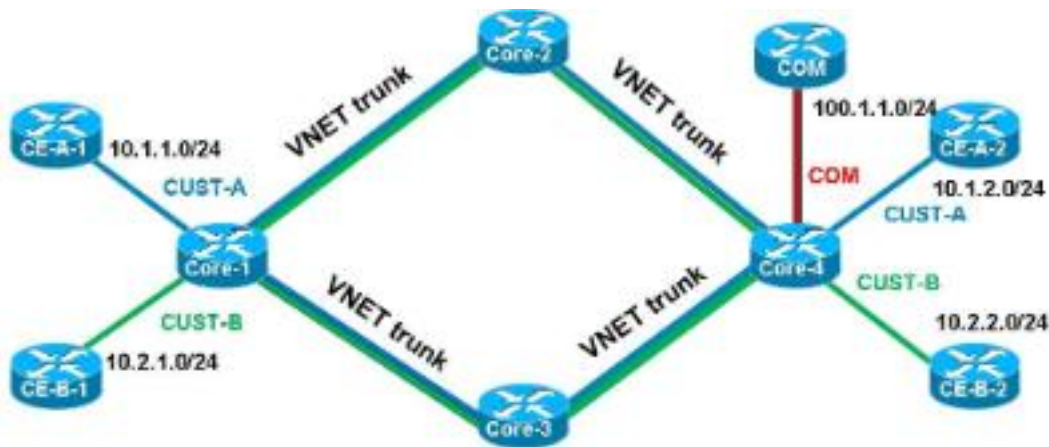
Configurar

Utilice la información que se describe en esta sección para configurar la característica EVN.

Nota: Use la [Command Lookup Tool](#) ([clientes registrados solamente](#)) para obtener más información sobre los comandos usados en esta sección.

Diagrama de la red

Esta configuración de la red se utiliza para ilustrar la configuración y los comandos show EVN:



Aquí están algunas NOTAS IMPORTANTES sobre esta configuración:

- Se definen dos VRF (**CUST-A** y **CUST-B**) que se realizan de la base de la red a través de los trunks VNET.
- El Open Shortest Path First (OSPF) se utiliza en los VRF para hacer publicidad del accesibilidad.
- El servidor común de los host A VRF COM (100.1.1.100) que debe ser accesible de VRF CUST-A y de CUST-B.
- La imagen se utiliza que es `i86bi_linux-adventerprisek9-ms.153-1.S`.

Consejo: El Cisco IOS en Linux (IOL) puso se utiliza que está disponible [aquí](#).

Configuración EVN

Complete estos pasos para configurar la característica EVN:

1. Configure la definición VRF:

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

!Aquí están algunas NOTAS IMPORTANTES sobre esta configuración:

Cisco recomienda que usted utiliza las etiquetas en el rango de 2 a 1,000. No utilice los VLA N reservados 1,001 a 1,005. Los VLA N extendidos 1,006 a 4,094 se pueden utilizar, si es necesario.

La etiqueta VNET no se debe utilizar por un VLA N actual.

Las etiquetas VNET deben ser lo mismo en todos los dispositivos para cualquier VRF dado.

El direccionamiento-family ipv4|ipv6 se debe configurar para activar el VRF en el AF relacionado.

No hay necesidad de definir una dirección de la ruta (RD) porque EVN no utiliza el BGP. Con esta configuración, los VRF se deben definir en todos los routers del núcleo 4x. Por ejemplo, en 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 Utilice la misma etiqueta VNET en todo el Routers para estos VRF. En CORE-4, el VRF COM no requiere una etiqueta VNET. La meta es guardar que VRF local en CORE-4 y configura escaparse y la redistribución para proporcionar el acceso al servidor común de CUST-A y de CUST-B.

Ingrese este comando para marcar los diversos contadores 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. Configure el trunk 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#
```

Aquí están algunas NOTAS IMPORTANTES sobre esta configuración:

El comando trunk del vnet crea tantas subinterfaces del dot1q como el número de VRF que

se definan con una etiqueta VNET.

El comando trunk del vnet no puede coexistir con algunas subinterfaces manual-configuradas en la misma interfaz física.

Esta configuración se permite en las interfaces ruteadas (no puertos del switch), la comprobación y el portchannel.

Los IP Addresses (y otros comandos) que se aplica en la interfaz física son heredados por las subinterfaces.

Las subinterfaces para todos los VRF utilizan la misma dirección IP.

Con esta configuración, hay dos VNET VRF, así que dos subinterfaces se crean automáticamente en la interfaz que se configura como el trunk VNET. Usted puede ingresar el comando de los derivar-config de la demostración para ver la configuración oculta que se crea automáticamente:

Aquí está la configuración que se ejecuta actualmente:

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

Aquí está la configuración derivada:

```
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
```

Como se muestra, todas las subinterfaces heredan la dirección IP de la interfaz principal.

3. Asigne las interfaces (sub) del borde a los VRF. Para asignar una interfaz o una subinterfaz a un VNET VRF, utilice el mismo procedimiento que ése usado para asignar un VRF normalmente:

```
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#Con esta configuración, la configuración se aplica en CORE-1 y CORE-4. Aquí está un ejemplo para CORE-4:

```
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#
```

4. Routing Protocol de la configuración para cada VRF (esto no es específico a EVN o a VNET):

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

...Nota: Esta configuración debe incluir los direccionamientos del trunk VNET así como los direccionamientos de la interfaz del borde.

Con esta configuración, se definen dos procesos OSPF, uno por el 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#Usted puede ingresar el modo del contexto de ruteo para ver la información que se relaciona con un VRF específico sin las especificaciones VRF en cada comando:

```
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**#**Nota:** Las visualizaciones de la salida del **comando show ip protocols** solamente la información que se relaciona con el VRF seleccionado.

Cuando usted ve el Routing Information Base (RIB) para ambos VRF, usted puede verificar la subred remota vía los dos trunks 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. Determine Route Leaking entre los VRF. Route Leaking se realiza vía la replicación de la ruta. Por ejemplo, algunas rutas en un VRF se pudieron hacer disponibles para otro VRF:

```
vrf definition VRF-X
```

```
address-family ipv4|ipv6
```

```
route-replicate from vrf VRF-Y unicast|multicast
```

[**route-origin**] [route-map [name]] Aquí están algunas NOTAS IMPORTANTES sobre esta configuración:

El RIB para **VRF-X** tiene acceso a las rutas seleccionadas, sobre la base de los parámetros de comando de **VRF-Y**.

Las rutas replicadas en **VRF-X** se marcan con un indicador del [+].

La opción del **Multicast** permite el uso de las rutas de otro VRF para el reenvío de trayecto inverso (RPF).

El ruta-origen puede tener uno de estos valores:

todosbgpconectadoeigrpISISmóvilodrOSPFripestática

A diferencia del nombre indica, las rutas no se replican ni se duplican; éste es el caso con escaparse normal con el BGP RT común, que no consume memoria extra.

Con esta configuración, Route Leaking se utiliza en CORE-4 para proporcionar el acceso de CUST-A y de CUST-B al COM (y viceversa):

```

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 este momento, las rutas replicadas no se propagan en el Interior Gateway Protocol (IGP), tan solamente el CE-A-2 y el CE-B-2 tienen acceso al servicio COM (100.1.1.100), no a CE-A-1 y a CE-B-1.

Usted puede también utilizar Route Leaking o a una tabla global:

```

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. Defina la propagación de Route Leaking. Las rutas escapadas no se duplican en el RIB de la blanco VRF. Es decir no son una parte del RIB de la blanco VRF. La redistribución normal entre los procesos del router no trabaja, así que usted debe definir explícitamente la conexión VRF del RIB al cual la ruta pertenece:

```

router ospf x vrf VRF-X
  redistribute vrf VRF-Y [route-origin] [route-map [name]]

```

Las rutas escapadas de VRF-Y se redistribuyen en el proceso OSPF que se ejecuta en VRF-X. Aquí está un ejemplo en CORE-4:

```

router ospf x vrf VRF-X
  redistribute vrf VRF-Y [route-origin] [route-map [name]]

```

El route-map no se necesita en este caso, puesto que hay solamente un Routeconectad en VRF COM. Ahora hay accesibilidad al servicio COM (100.1.1.100) de CE-A-1 y 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#

```

Ajuste el trunk VNET

Esta sección proporciona la información que usted puede utilizar para ajustar el trunk VNET.

Lista del trunk

Por abandono, todos los VRF que se configuran con una etiqueta VNET se permiten en todos los trunks VNET. Una lista del trunk permite que usted especifique la lista de VRF autorizados en el trunk VNET:

```

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

```

Nota: Debe haber una línea por el VRF permitido.

Como un ejemplo, CORE-1 está ajustado para el VRF CUST-B en el trunk VNET entre CORE-1 y CORE-2:

```

vrf list [list-name]
  member [vrf-name]
!

```

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

Como se muestra, el peering OSPF para VRF CUST-B a través del trunk va abajo:

```
%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 subinterfaz para VRF CUST-B se quita:

```
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
!
```

Atributos del trunk Por-VRF

Por abandono, las subinterfaces del dot1q heredan los parámetros de la interfaz física de modo que las subinterfaces para todos los VRF tengan los mismos atributos (tales como coste y autenticación). Usted puede ajustar los parámetros de tronco por la etiqueta VNET:

```
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
!
```

Usted puede ajustar estos parámetros:

```
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
```

En este ejemplo, el costo de OSPF por el VRF para CORE-1 se cambia, así que la trayectoria CORE-2 se utiliza para CUST-A y la trayectoria CORE-3 para CUST-B (el costo predeterminado es 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#
```

Etiquetas del Por-link VNET

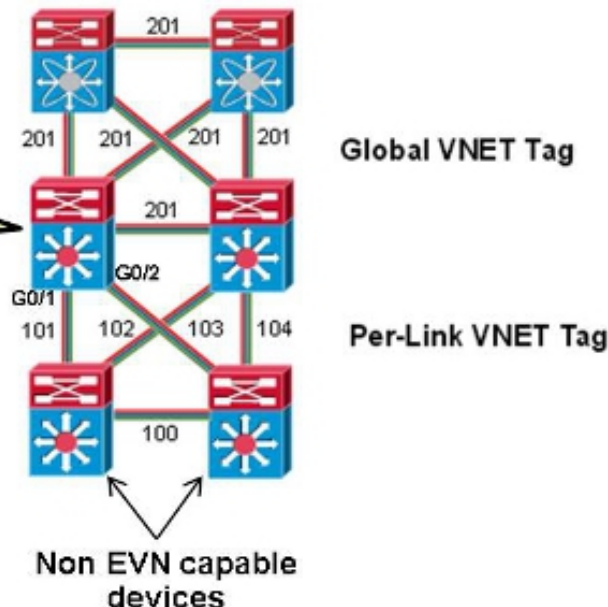
Por abandono, la etiqueta VNET que se define en la definición VRF se utiliza para todos los trunks. Sin embargo, usted puede utilizar una diversa etiqueta VNET por el trunk.

Este ejemplo describe un escenario donde usted está conectado con un dispositivo con capacidad NON-EVN y usted utiliza VRF-Lite con un trunk manual, y la etiqueta global VNET es utilizada por otro VLA N:

```

vrf definition VRF-X
 vnet tag 201
 !
interface GigabitEthernet0/1
 vnet trunk
 vnet name VRF-X
 vnet tag 101
 !
interface GigabitEthernet0/2
 vnet trunk
 vnet name VRF-X
 vnet tag 102

```



Con esta configuración, la etiqueta VNET que se utiliza en el trunk entre CORE-1 y CORE-2 para CUST-A se cambia a partir del 100 a 101:

```

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

```

Después de que este cambio ocurra en CORE-1, se crea una nueva subinterfaz:

```

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 este cambio ocurre solamente en un extremo, después la Conectividad se pierde en el VRF asociado y el OSPF va abajo:

```

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

```

Una vez que la misma etiqueta VNET se utiliza en CORE-2, se restablece la Conectividad y la etiqueta 101 del dot1q se utiliza en ese trunk mientras que 100 todavía se utiliza en el CORE-1 al trunk CORE-3:

```

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

```

Verificación

Actualmente, no hay un procedimiento de verificación disponible para esta configuración.

Troubleshooting

Actualmente, no hay información específica de troubleshooting disponible para esta configuración.

Información Relacionada

- [Red virtual fácil - Simplificación de la virtualización de la red de la capa 3](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)