

# Exemplo de configuração fácil da rede virtual

## Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Configurar](#)

[Diagrama de Rede](#)

[Configurar EVN](#)

[Ajuste o tronco VNET](#)

[Lista do tronco](#)

[Atributos do tronco Por-VRF](#)

[Etiquetas do Por-link VNET](#)

[Verificar](#)

[Troubleshooting](#)

[Informações Relacionadas](#)

## Introdução

Este documento descreve a característica fácil da rede virtual (EVN), que é projetada a fim fornecer um fácil, simples-à-configurar o mecanismo da virtualização nas redes do campus. Leverages tecnologias atual, tais como o encapsulamento do roteamento virtual e do Transmissão-Lite (VRF-Lite) e do dot1q, e não introduz nenhum protocolo novo.

## Pré-requisitos

### Requisitos

Não existem requisitos específicos para este documento.

### [Componentes Utilizados](#)

As informações neste documento são baseadas nas seguintes versões de hardware e software:

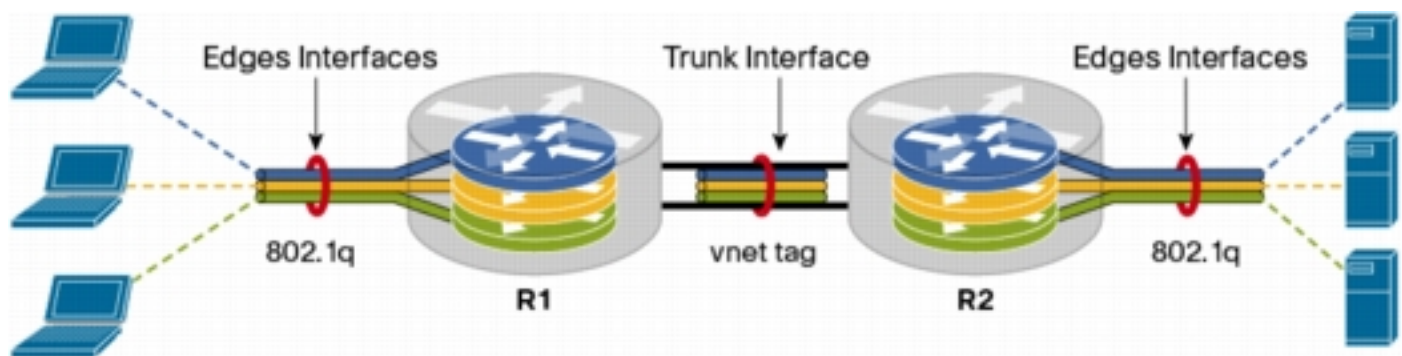
- Series Switch do Cisco catalyst 6000 (Cat6K) que executam a versão de software 15.0(1)SY1

- O Cisco 1000 Series agregou o Roteadores dos serviços (ASR1000) essa versão de software 3.2s da corrida
- Roteadores dos Serviços integrados do Cisco e Series que executa versões 15.3(2)T e mais recente do <sup>®</sup> do Cisco IOS
- Series Switch (Cat4900) do Cisco catalyst 4500 (Cat4500) e 4900 que executam a versão de software 15.1(1)SG

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

## Informações de Apoio

Está aqui uma vista geral da característica EVN:



- A característica EVN usa VRF-Lite a fim criar diversos (contextos de roteamento de até 32).
- A Conectividade dentro do roteamento virtual e da transmissão (VRF) entre os dispositivos Layer3 é assegurada através dos troncos da rede virtual (VNET).
- Os troncos VNET são troncos regulares do dot1q.
- Cada VRF que deve ser transportado através dos troncos VNET deve ser configurado com uma etiqueta VNET.
- Cada etiqueta VNET iguala uma etiqueta do dot1q.
- **As subinterfaces do dot1q são criadas automaticamente e hidden.**
- **A configuração da interface principal é herdada por todas as (hidden) subinterfaces.**
- As instâncias separadas dos protocolos de roteamento devem ser usadas em cada VRF sobre os troncos VNET a fim anunciar a alcançabilidade do prefixo.
- A rota dinâmica que escapa entre os VRF (opostos às rotas estáticas) é permitida sem o uso do Border Gateway Protocol (BGP).

- A característica é apoiada para o IPv4 e o IPv6.

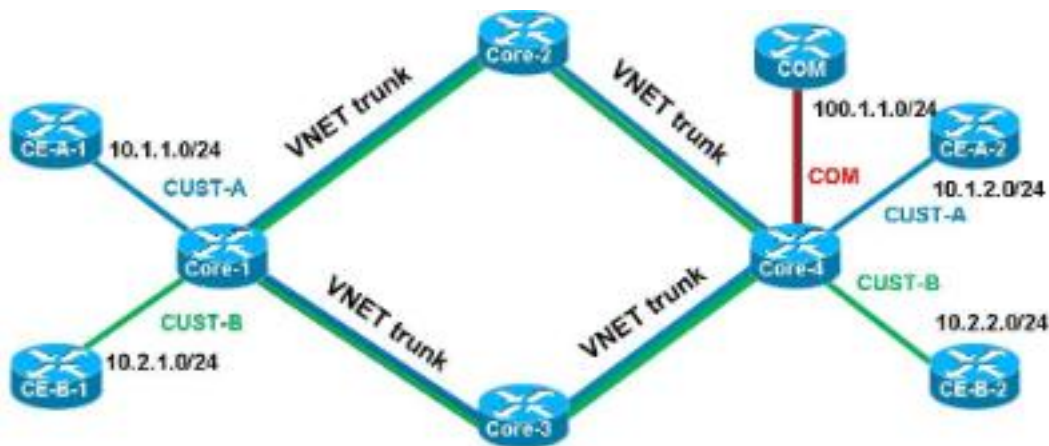
## Configurar

Use a informação que é descrita nesta seção a fim configurar a característica EVN.

Nota: Use a [Command Lookup Tool](#) ( [somente clientes registrados](#)) para obter mais informações sobre os comandos usados nesta seção.

## Diagrama de Rede

Esta instalação de rede é usada a fim ilustrar a configuração e comandos show EVN:



Estão aqui algumas observações importantes sobre esta instalação:

- Dois VRF são definidos (**CUST-A** e **CUST-B**) que são realizados do núcleo da rede através dos troncos VNET.
- O Open Shortest Path First (OSPF) é usado nos VRF a fim anunciar a alcançabilidade.
- O VRF COM hospeda um servidor comum (**100.1.1.100**) que deva ser alcançável de VRF CUST-A e de CUST-B.
- A imagem que é usada é **i86bi\_linux-adventerprisek9-ms.153-1.S**.

Dica: O Cisco IOS em Linux (IOL) setup que é usado está disponível [aqui](#).

## Configurar EVN

Termine estas etapas a fim configurar a característica EVN:

### 1. Configurar a definição VRF:

```
vrf definition [name]
vnet tag [2-4094]
```

```
!  
address-family ipv4|ipv6  
exit-address-family
```

!Estão aqui algumas observações importantes sobre esta configuração:

Cisco recomenda que você usa etiquetas na escala de 2 a 1,000. Não use os VLAN reservados 1,001 a 1,005. Os VLAN prolongados 1,006 a 4,094 podem ser usados, se necessários.

A etiqueta VNET não deve ser usada por um VLAN atual.

As etiquetas VNET devem ser as mesmas em todos os dispositivos para todo o VRF dado.

O endereço-family ipv4|ipv6 deve ser configurado a fim ativar o VRF no AF relacionado.

Não há nenhuma necessidade de definir um sentido da rota (RD) porque EVN não usa o BGP.

Com esta instalação, os VRF devem ser definidos em todos os roteadores centrais 4x. Por exemplo, em 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 Use a mesma etiqueta VNET em todo o Roteadores para estes VRF. Em CORE-4, o VRF COM não exige uma etiqueta VNET. O objetivo é manter-se que VRF local em CORE-4 e configura o escape e a redistribuição a fim fornecer o acesso ao servidor comum de CUST-A e de CUST-B.

Incorpore este comando a fim verificar vários 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. Configurar o tronco 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#
```

Estão aqui algumas observações importantes sobre esta configuração:

O comando trunk do vnet cria tantas como subinterfaces do dot1q como o número de VRF que são definidos com uma etiqueta VNET.

O comando trunk do vnet não pode coexistir com algumas subinterfaces manual-

configuradas na mesma interface física.

Esta configuração é permitida em interfaces roteada (não portas de switch), em exame e em portchannel.

Os endereços IP de Um ou Mais Servidores Cisco ICM NT (e outros comandos) que é aplicado na interface física são herdados pelas subinterfaces.

As subinterfaces para todos os VRF usam o mesmo endereço IP de Um ou Mais Servidores Cisco ICM NT.

Com esta instalação, há dois VNET VRF, assim que duas subinterfaces são criadas automaticamente na relação que é configurada como o tronco VNET. Você pode incorporar o comando da derivar-**configuração da mostra** a fim ver hidden a configuração que é criada automaticamente:

Está aqui a configuração que é executado atualmente:

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

CORE-1#**Está aqui a configuração 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
```

CORE-1#**Como mostrado, todas as subinterfaces herdam o endereço IP de Um ou Mais Servidores Cisco ICM NT da interface principal.**

3. Atribua relações (secundárias) da borda aos VRF. A fim atribuir uma relação ou uma subinterface a um VNET VRF, use o mesmo procedimento que aquele usado a fim atribuir normalmente um VRF:

```
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#Com esta instalação, a configuração é aplicada em CORE-1 e em CORE-4. Está aqui um exemplo 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. Configurar protocolos de roteamento para cada VRF (isto não é específico a EVN ou a VNET):

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

...Nota: Esta configuração deve incluir os endereços do tronco VNET assim como os endereços da relação da borda.

Com esta instalação, dois processos de OSPF são definidos, um pelo 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#Você pode entrar no modo do contexto de roteamento a fim ver a informação que é relacionada a um VRF específico sem as especificações VRF em 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: Os indicadores da saída do comando **show ip protocols** somente a informação que é relacionada ao VRF selecionado.

Quando você vê o Routing Information Base (RIB) para ambos os VRF, você pode verificar a sub-rede remota através dos dois troncos 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 o vazamento de rota entre os VRF. O vazamento de rota é executado através da replicação da rota. Por exemplo, algumas rotas em um VRF puderam ser feitas disponíveis para um outro VRF:

```
vrf definition VRF-X
  address-family ipv4|ipv6
    route-replicate from vrf VRF-Y unicast|multicast
[route-origin] [route-map [name]]
```

Estão aqui algumas observações importantes sobre esta configuração:

O RIB para **VRF-X** tem o acesso às rotas selecionadas, com base em parâmetros de comando de **VRF-Y**.

As rotas replicated em **VRF-X** são identificadas por meio de uma bandeira do **[+]**.

A opção do **Multicast** permite o uso das rotas de um outro VRF para o encaminhamento de caminho reverso (RPF).

A rota-origem pode ter um destes valores:

**tudobgpconectadoeigrpismóvelodrOSPFripestático**

Ao contrário do nome indica, as rotas não replicated nem são duplicadas; este é o caso com escape normal com o BGP RT comum, que não consome a memória extra.

Com esta instalação, o vazamento de rota é usado em CORE-4 a fim fornecer o acesso de CUST-A e de CUST-B ao COM (e 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

```

Neste momento, as rotas replicated não são propagadas no Interior Gateway Protocol (IGP), tão somente no CE-A-2 e no CE-B-2 têm o acesso ao serviço COM (100.1.1.100), não ao CE-A-1 e ao CE-B-1.

Você pode igualmente usar o vazamento de rota ou a uma tabela 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 a propagação do vazamento de rota. As rotas escapadas não são duplicadas no RIB do alvo VRF. Ou seja não são parte do RIB do alvo VRF. A redistribuição normal entre processos de roteador não trabalha, assim que você deve explicitamente definir a conexão VRF do RIB a que a rota pertence:

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

redistribute vrf VRF-Y [route-origin] [route-map [name]]

As rotas escapadas de VRF-Y são redistribuídas no processo de OSPF que é executado em VRF-X. Está aqui um exemplo em CORE-4:

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

redistribute vrf VRF-Y [route-origin] [route-map [name]]

O mapa de rotas não é precisado neste caso, desde que há somente uma rota conectada em VRF COM. Há agora alcançabilidade ao serviço COM (100.1.1.100) de CE-A-1 e 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 o tronco VNET

Esta seção fornece a informação que você pode usar a fim ajustar o tronco VNET.

### Lista do tronco

Àrevelia, todos os VRF que são configurados com uma etiqueta VNET são permitidos em todos os troncos VNET. Uma lista do tronco permite que você especifique a lista de VRF autorizados no tronco VNET:

```
vrf list [list-name]
```

```
  member [vrf-name]
```

```
!
```

```
interface GigabitEthernetx/x
```

```
  vnet trunk list [list-name]
```

Nota: Deve haver uma linha pelo VRF permitido.

Como um exemplo, CORE-1 é ajustado para o VRF CUST-B no tronco VNET entre CORE-1 e CORE-2:

```
vrf list [list-name]
```

```
  member [vrf-name]
```

```
!
```

```
interface GigabitEthernetx/x
```

```
  vnet trunk list [list-name]
```

Como mostrado, o OSPF que espera para VRF CUST-B através do tronco vai para baixo:

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

A subinterface para VRF CUST-B é removida:

```
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 do tronco Por-VRF

À revelia, as subinterfaces do dot1q herdam os parâmetros da interface física de modo que as subinterfaces para todos os VRF tenham os mesmos atributos (tais como o custo e a autenticação). Você pode ajustar os parâmetros de tronco pela 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
!
```

Você pode ajustar estes parâmetros:

```
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
```

Neste exemplo, o os custos de OSPF pelo VRF para CORE-1 são mudados, assim que o trajeto CORE-2 é usado para CUST-A e o trajeto CORE-3 para CUST-B (os custos padrão são 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 do Por-link VNET

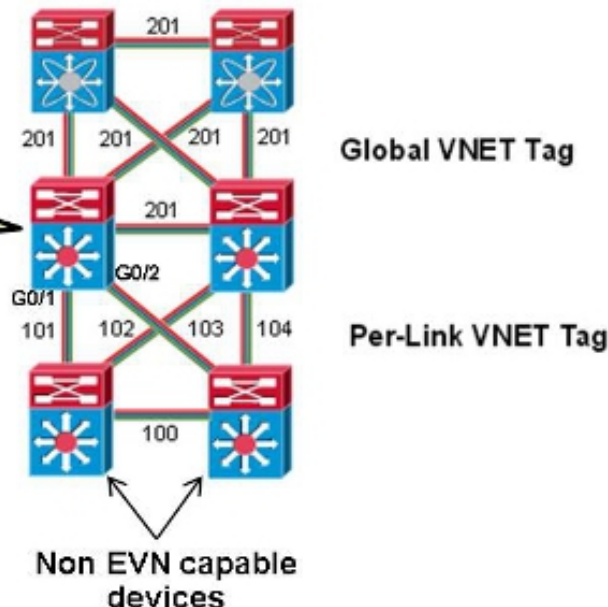
À revelia, a etiqueta VNET que é definida na definição VRF é usada para todos os troncos. Contudo, você pode usar uma etiqueta diferente VNET pelo tronco.

Este exemplo descreve uma encenação onde você seja conectado a um dispositivo NON-EVN capaz e você use VRF-Lite com um tronco manual, e a etiqueta global VNET é usada por um outro VLAN:

```

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

```



Com esta instalação, a etiqueta VNET que é usada no tronco entre CORE-1 e CORE-2 para CUST-A é mudada de 100 a 101:

```

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

```

Depois que esta mudança ocorre em CORE-1, uma subinterface nova está criada:

```

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

```

Se esta mudança ocorre somente em uma extremidade, a seguir a Conectividade está perdida no VRF associado e o OSPF vai para baixo:

```

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

```

Uma vez que a mesma etiqueta VNET é usada em CORE-2, a Conectividade está restaurada e a etiqueta 101 do dot1q está usada nesse tronco quando 100 forem usados ainda no CORE-1 ao tronco CORE-3:

```

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

```

## Verificar

No momento, não há procedimento de verificação disponível para esta configuração.

## Troubleshooting

Atualmente, não existem informações disponíveis específicas sobre Troubleshooting para esta configuração.

## Informações Relacionadas

- [Rede virtual fácil - Simplificando a virtualização da rede da camada 3](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)