

Exemplo de configuração de rede virtual fácil

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Configurar](#)

[Diagrama de Rede](#)

[Configurar EVN](#)

[Ajustar o tronco VNET](#)

[Lista de Troncos](#)

[Atributos de tronco por VRF](#)

[Marcas de VNET por link](#)

[Verificar](#)

[Troubleshoot](#)

[Informações Relacionadas](#)

Introduction

Este documento descreve o recurso Easy Virtual Network (EVN), projetado para fornecer um mecanismo de virtualização fácil e simples de configurar em redes de campus. Ele aproveita as tecnologias atuais, como Virtual Routing and Forwarding-Lite (VRF-Lite) e o encapsulamento dot1q, e não introduz nenhum novo protocolo.

Prerequisites

Requirements

Não existem requisitos específicos para este documento.

Componentes Utilizados

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

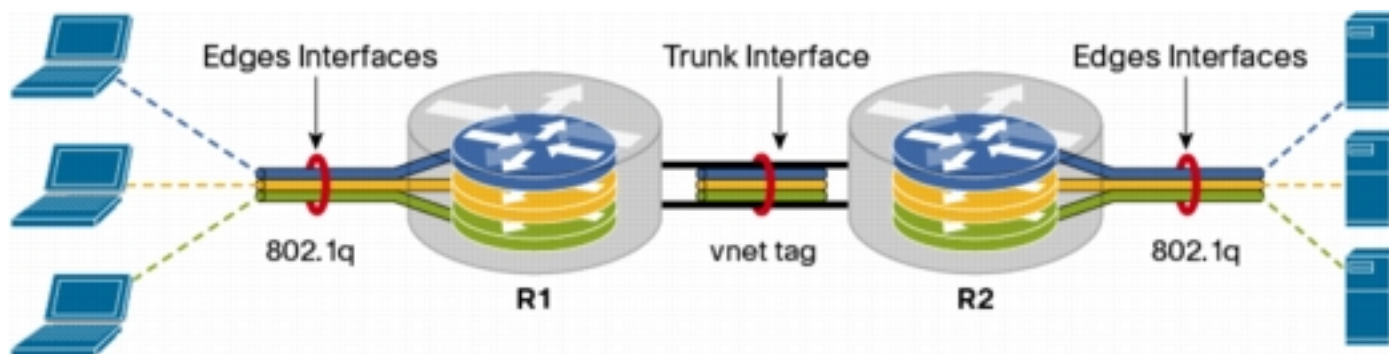
- Switches Cisco Catalyst 6000 (Cat6k) Series que executam a versão de software 15.0(1)SY1

- Roteadores de serviços agregados Cisco 1000 Series (ASR1000) que executam a versão de software 3.2s
- Roteadores de serviços integrados Cisco 3925 e 3945 Series que executam o Cisco IOS® versões 15.3(2)T e posteriores
- Switches Cisco Catalyst 4500 (Cat4500) e 4900 (Cat4900) Series que executam o software versão 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. If your network is live, make sure that you understand the potential impact of any command.

Informações de Apoio

Aqui está uma visão geral do recurso EVN:



- O recurso EVN usa VRF-Lite para criar vários (até 32) contextos de roteamento.
- A conectividade dentro do Virtual Routing and Forwarding (VRF) entre dispositivos de Camada 3 é garantida através de troncos de Rede Virtual (VNET).
- Os troncos VNET são troncos dot1q regulares.
- Cada VRF que deve ser transportado através dos troncos VNET deve ser configurado com uma marca VNET.
- Cada marca VNET é igual a uma marca dot1q.
- **As subinterfaces dot1q são criadas e ocultadas automaticamente.**
- **A configuração da interface principal é herdada por todas as subinterfaces (ocultas).**
- Instâncias separadas de protocolos de roteamento devem ser usadas em cada VRF sobre troncos VNET para anunciar a acessibilidade do prefixo.
- O vazamento de rota dinâmica entre VRFs (ao contrário das rotas estáticas) é permitido sem o uso do Border Gateway Protocol (BGP).

- O recurso é suportado para IPv4 e IPv6.

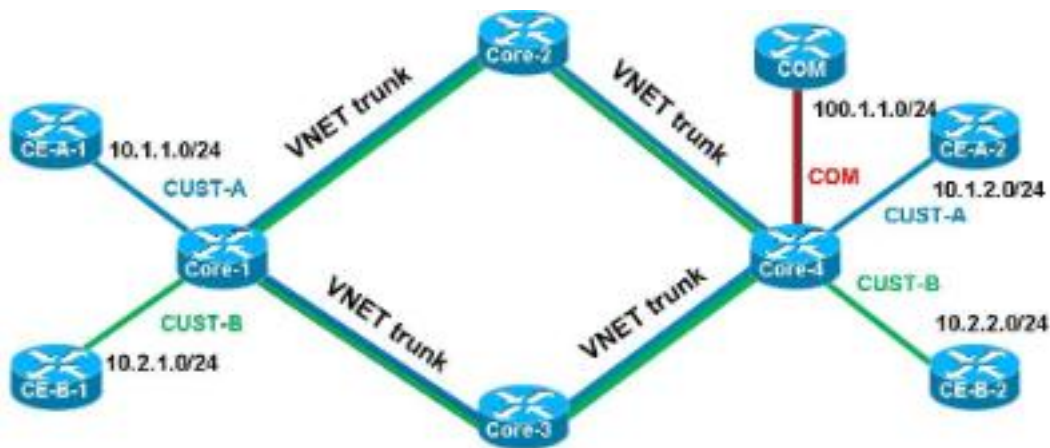
Configurar

Use as informações descritas nesta seção para configurar o recurso EVN.

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

Diagrama de Rede

Essa configuração de rede é usada para ilustrar a configuração do EVN e os comandos show:



Aqui estão algumas notas importantes sobre esta configuração:

- Dois VRFs são definidos (**CUST-A** e **CUST-B**) que são executados a partir do núcleo da rede através de troncos VNET.
- O OSPF (Open Shortest Path First) é usado nos VRFs para anunciar a acessibilidade.
- O VRF COM hospeda um servidor comum (100.1.1.100) que deve ser alcançável do VRF CUST-A e CUST-B.
- A imagem usada é **i86bi_linux-adventerprise9-ms.153-1.S**.

Tip: A configuração do Cisco IOS no Linux (IOL) usada está disponível [aqui](#).

Configurar EVN

Conclua estes passos para configurar o recurso EVN:

1. Configure a definição de VRF:

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

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

Aqui estão algumas observações importantes sobre esta configuração:

A Cisco recomenda que você use tags no intervalo de 2 a 1.000. Não use as VLANs 1.001 a 1.005 reservadas. As VLANs estendidas de 1.006 a 4.094 podem ser usadas, se necessário.

A marca VNET não deve ser usada por uma VLAN atual.

As marcas VNET devem ser as mesmas em todos os dispositivos para qualquer VRF especificado.

O endereço-família **ipv4|ipv6** deve ser configurado para ativar o VRF no AF relacionado.

Não há necessidade de definir uma Direção de Rota (RD) porque o EVN não usa BGP. Com essa configuração, os VRFs devem ser definidos em todos os roteadores de núcleo 4x. Por exemplo, no 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 marca VNET em todos os roteadores para esses VRFs. No CORE-4, VRF COM não requer uma marca VNET. O objetivo é manter esse VRF local no CORE-4 e configurar o vazamento e a redistribuição para fornecer acesso ao servidor comum do CUST-A e do CUST-B.

Insira este comando para 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. Configure o tronco VNET:

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

Aqui estão algumas observações importantes sobre esta configuração:

O comando **vnet trunk** cria tantas subinterfaces dot1q como o número de VRFs definidos com uma marca VNET.

O comando **vnet trunk** não pode coexistir com algumas subinterfaces configuradas manualmente na mesma interface física.

Essa configuração é permitida em interfaces roteadas (não portas de switch), físicas e de canal de porta.

Os endereços IP (e outros comandos) aplicados na interface física são herdados pelas subinterfaces.

As subinterfaces de todos os VRFs usam o mesmo endereço IP. Com essa configuração, há dois VRFs VNET, de modo que duas subinterfaces são criadas automaticamente na interface configurada como o tronco VNET. Você pode inserir o comando **show derivado-config** para exibir a configuração oculta que é criada automaticamente:

Esta é a configuração que é executada 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#
```

Esta é 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 da interface principal.

3. Atribuir (sub)interfaces de borda aos VRFs. Para atribuir uma interface ou subinterface a um VRF VNET, use o mesmo procedimento usado para atribuir um VRF normalmente:

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

Com essa configuração, a configuração é aplicada no CORE-1 e no CORE-4. Aqui está um exemplo para o 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. Configurar protocolos de roteamento para cada VRF (isso não é específico para EVN ou VNET):

```

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

```

Note: Essa configuração deve incluir os endereços de tronco VNET, bem como os endereços da interface de borda.

Com essa configuração, dois processos OSPF são definidos, um por 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 de contexto de roteamento para visualizar as informações relacionadas a um VRF específico sem as especificações de 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#

```

Note: A saída do comando **show ip protocols** exibe apenas as informações relacionadas ao VRF selecionado.

Quando você visualiza o Routing Information Base (RIB) para ambos os VRFs, é possível 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 da rota entre os VRFs. O vazamento de rota é realizado por meio da replicação de rota. Por exemplo, algumas rotas em um VRF podem ser disponibilizadas para outro VRF:

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

Aqui estão algumas observações importantes sobre esta configuração:

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

As rotas replicadas no **VRF-X** são marcadas com um flag **[+]**.

A opção **multicast** permite o uso de rotas de outro VRF para RPF (Reverse Path Forwarding).

A **origem da rota** pode ter um destes valores:

todos osbgpconectadoeigrpismoívelodrospfripestático

Ao contrário do nome indica, as rotas não são replicadas ou duplicadas; esse é o caso com vazamento normal através de RT comum de BGP, que não consome memória extra.

Com essa configuração, o vazamento de rota é usado no CORE-4 para fornecer acesso de

CUST-A e CUST-B a 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
```

Nesse ponto, as rotas replicadas não são propagadas no Interior Gateway Protocol (IGP), portanto, somente CE-A-2 e CE-B-2 têm acesso ao serviço COM (100.1.1.100), não CE-A-1 e CE-B-1.

Você também pode usar o vazamento de rota de ou para 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 de vazamento de rota. As rotas vazadas não são duplicadas no RIB do VRF de destino. Em outras palavras, eles não fazem parte do RIB do VRF de destino. A redistribuição normal entre os processos do roteador não funciona, então você deve definir explicitamente a conexão VRF do RIB à qual a rota pertence:

```

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

```

As rotas vazadas do VRF-Y são redistribuídas no processo OSPF executado no VRF-X. Veja um exemplo no 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

```

O mapa de rota não é necessário nesse caso, pois há apenas uma rota conectada no VRF COM. Agora há acessibilidade para o serviço COM (100.1.1.100) do CE-A-1 e do 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#

```

Ajustar o tronco VNET

Esta seção fornece informações que você pode usar para ajustar o tronco VNET.

Lista de Troncos

Por padrão, todos os VRFs configurados com uma marca VNET são permitidos em todos os troncos VNET. Uma lista de troncos permite especificar a lista de VRFs autorizados no tronco VNET:

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

Note: Deve haver uma linha por VRF permitido.

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

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

Como mostrado, o peering OSPF para VRF CUST-B através do tronco cai:

```
%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 do 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 de tronco por VRF

Por padrão, as subinterfaces dot1q herdam os parâmetros da interface física para que as subinterfaces de todos os VRFs tenham os mesmos atributos (como custo e autenticação). Você pode ajustar os parâmetros de tronco por tag VNET:

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

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 custo OSPF por VRF para CORE-1 é alterado, de modo que o caminho CORE-2 é usado para CUST-A e o caminho CORE-3 para CUST-B (o custo padrão é 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#

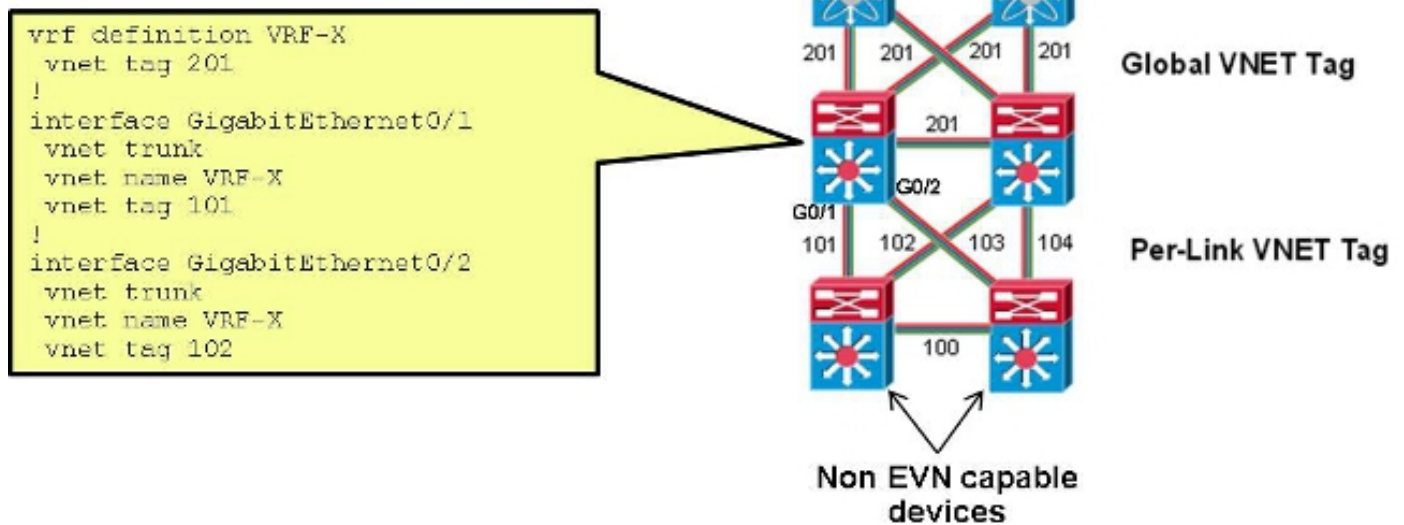
```

Marcas de VNET por link

Por padrão, a marca VNET definida na definição de VRF é usada para todos os troncos. No entanto, você pode usar uma marca VNET diferente por tronco.

Este exemplo descreve um cenário em que você está conectado a um dispositivo sem

capacidade de EVN e usa VRF-Lite com um tronco manual, e a marca global VNET é usada por outra VLAN:



Com essa configuração, a marca VNET usada no tronco entre CORE-1 e CORE-2 para CUST-A é alterada de **100** para **101**:

```

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

Após essa alteração ocorrer no CORE-1, uma nova subinterface será 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 essa alteração ocorrer somente em uma extremidade, a conectividade será perdida no VRF associado e o OSPF será desativado:

```

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

Quando a mesma marca VNET é usada no CORE-2, a conectividade é restaurada e a marca dot1q **101** é usada nesse tronco enquanto **100** ainda é usada no tronco CORE-1 para 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.

Troubleshoot

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

Informações Relacionadas

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