

# Esempio di configurazione semplice della rete virtuale

## Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Configurazione](#)

[Esempio di rete](#)

[Configurazione EVN](#)

[Ottimizzare il trunk della rete virtuale](#)

[Elenco trunk](#)

[Attributi per-VRF Trunk](#)

[Tag rete virtuale per collegamento](#)

[Verifica](#)

[Risoluzione dei problemi](#)

[Informazioni correlate](#)

## Introduzione

Questo documento descrive la funzione Easy Virtual Network (EVN), progettata per fornire un meccanismo di virtualizzazione semplice da configurare nelle reti del campus. Sfrutta le tecnologie correnti, quali Virtual Routing and Forwarding-Lite (VRF-Lite) e incapsulamento dot1q, e non introduce alcun nuovo protocollo.

## Prerequisiti

### Requisiti

Nessun requisito specifico previsto per questo documento.

### Componenti usati

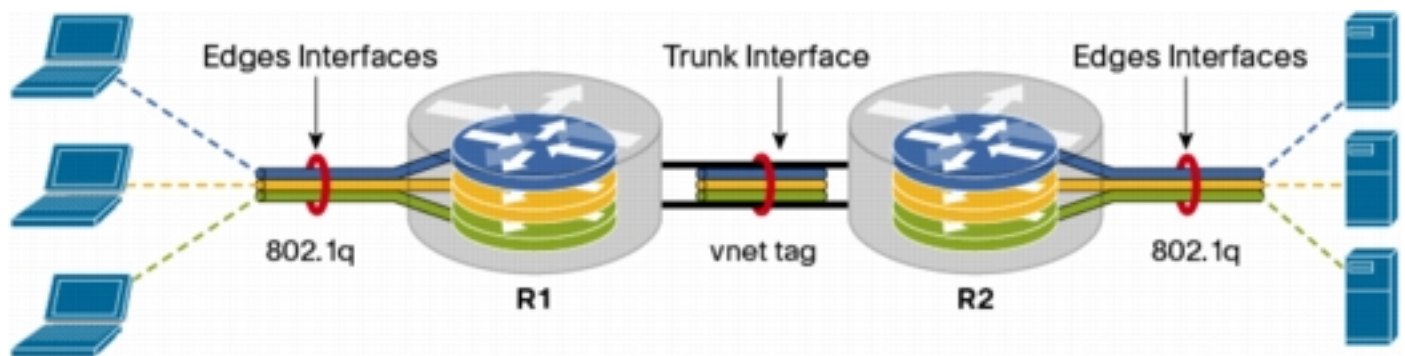
Le informazioni di questo documento si basano sulle seguenti versioni hardware e software:

- Switch Cisco Catalyst serie 6000 (Cat6k) con software versione 15.0(1)SY1
- Cisco serie 1000 Aggregated Services Router (ASR1000) con software versione 3.2s
- Router di servizi integrati Cisco serie 3925 e 3945 con Cisco IOS® versione 15.3(2)T e successive
- Switch Cisco Catalyst serie 4500 (Cat4500) e 4900 (Cat4900) con software versione 15.1(1)SG

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

## Premesse

Ecco una panoramica della funzione EVN:



- La funzione EVN utilizza VRF-Lite per creare diversi contesti di routing (fino a 32).
- La connettività all'interno del VRF (Virtual Routing and Forwarding) tra i dispositivi di layer 3 è garantita dai trunk della rete virtuale (VNET).
- I trunk VNET sono trunk standard dot1q.
- Ogni VRF che deve essere trasportato attraverso i trunk della rete virtuale deve essere configurato con un tag VNET.
- Ogni tag VNET è uguale a un tag dot1q.
- **Le sottointerfacce dot1q vengono create e nascoste automaticamente.**
- **La configurazione dell'interfaccia principale è ereditata da tutte le sottointerfacce (nascoste).**
- Per annunciare la raggiungibilità del prefisso, è necessario utilizzare istanze separate dei protocolli di routing in ogni VRF sui trunk VNET.
- È consentita la perdita dinamica del percorso tra VRF (in contrapposizione alle route statiche)

senza l'utilizzo del protocollo BGP (Border Gateway Protocol).

- Questa funzionalità è supportata per IPv4 e IPv6.

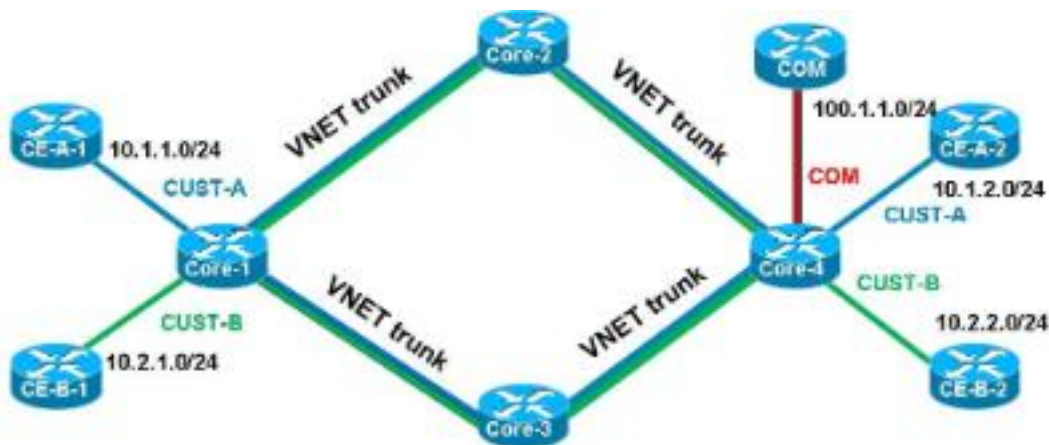
## Configurazione

Per configurare la funzione EVN, attenersi alle informazioni descritte in questa sezione.

**Nota:** per ulteriori informazioni sui comandi menzionati in questa sezione, usare lo strumento di ricerca dei comandi (solo utenti registrati).

## Esempio di rete

Questa configurazione di rete viene utilizzata per illustrare la configurazione EVN e i comandi show:



Di seguito sono riportate alcune note importanti su questa configurazione:

- Vengono definiti due VRF (**CUST-A** e **CUST-B**) che vengono eseguiti dal nucleo della rete tramite i trunk VNET.
- Open Shortest Path First (OSPF) viene utilizzato nei VRF per pubblicizzare la raggiungibilità.
- Il VRF COM ospita un server comune (100.1.1.100) che deve essere raggiungibile sia da VRF CUST-A che da CUST-B.
- L'immagine utilizzata è `i86bi_linux-adventerprisek9-ms.153-1.S`.

**Suggerimento:** Il programma di installazione di Cisco IOS su Linux (IOL) in uso è disponibile [qui](#).

## Configurazione EVN

Per configurare la funzione EVN, completare i seguenti passaggi:

## 1. Configurare la definizione VRF:

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

Ecco alcune note importanti su questa configurazione:

Cisco consiglia di utilizzare tag compresi tra 2 e 1000. Non utilizzare le VLAN riservate da 1001 a 1005. Se necessario, è possibile usare le VLAN estese da 1006 a 4094.

Il tag VNET non deve essere utilizzato da una VLAN corrente.

I tag VNET devono essere gli stessi su tutti i dispositivi per ogni VRF specificato.

È necessario configurare **address-family ipv4|ipv6** per attivare il VRF nell'AF correlato.

Non è necessario definire una direzione di route (RD) perché EVN non utilizza BGP. Con questa configurazione, i VRF devono essere definiti su tutti i router core 4x. Ad esempio, in 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
```

Utilizzare lo stesso tag VNET su tutti i router per questi VRF. In CORE-4, VRF COM non richiede un tag VNET. L'obiettivo è mantenere tale VRF locale sul CORE-4 e configurare la perdita e la redistribuzione in modo da fornire l'accesso al server comune da CUST-A e CUST-B.

Immettere questo comando per controllare i vari contatori della rete virtuale:

```
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. Configurare il trunk della rete virtuale:

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

Ecco alcune note importanti su questa configurazione:

Il comando **vnet trunk** crea tante sottointerfacce dot1q quante sono le VRF definite con un tag VNET.

Il comando **vnet trunk** non può coesistere con alcune sottointerfacce configurate manualmente sulla stessa interfaccia fisica.

Questa configurazione è consentita su interfacce indirizzate (non porte dello switch), su porte fisiche e su portchannel.

Gli indirizzi IP e gli altri comandi applicati all'interfaccia fisica vengono ereditati dalle sottointerfacce.

Le sottointerfacce di tutti i VRF utilizzano lo stesso indirizzo IP.

Con questa configurazione, sono disponibili due VRF VNET, in modo che due sottointerfacce vengano create automaticamente sull'interfaccia configurata come trunk VNET. È possibile immettere il comando **show derive-config** per visualizzare la configurazione nascosta creata automaticamente:

La configurazione attualmente in esecuzione è la seguente:

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

Di seguito è riportata la configurazione derivata:

```
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#
```

Come mostrato, tutte le sottointerfacce ereditano l'indirizzo IP dell'interfaccia principale.

3. Assegnare le interfacce (secondarie) ai VRF. Per assegnare un'interfaccia o una sottointerfaccia a un VRF VNET, utilizzare la stessa procedura utilizzata per assegnare un VRF:

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

Con questa configurazione, la configurazione viene applicata a CORE-1 e CORE-4. Di seguito è riportato un esempio per 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. Configurare i protocolli di routing per ogni VRF (opzione non specifica di EVN o VNET):

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

**Nota:** Questa configurazione deve includere gli indirizzi del trunk della rete virtuale e gli indirizzi dell'interfaccia perimetrale.

Con questa configurazione vengono definiti due processi OSPF, uno per ogni 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#
```

È possibile accedere alla modalità contesto di routing per visualizzare le informazioni correlate a un VRF specifico senza le specifiche VRF in ciascun 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:** L'output del comando **show ip protocols** visualizza solo le informazioni correlate al VRF selezionato.

Quando si visualizza la base RIB (Routing Information Base) per entrambi i VRF, è possibile verificare la subnet remota tramite i due trunk 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. Determinare il percorso di perdita tra i VRF. La perdita di route viene eseguita tramite la replica route. Ad esempio, alcune route di un VRF potrebbero essere rese disponibili per un altro VRF:

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

Ecco alcune note importanti su questa configurazione:

Il RIB per **VRF-X** ha accesso ai percorsi selezionati, in base ai parametri di comando del **VRF-Y**.

Le route replicate in **VRF-X** sono contrassegnate con un flag [+].

L'opzione **multicast** consente di utilizzare route da un altro VRF per l'inoltro di percorso inverso (RPF).

L'origine della route può avere uno dei valori seguenti:

**tuttbogpconnessoeigrpismobileodrospfripstatico**

A differenza del nome indicato, le route non vengono replicate o duplicate; questo è il caso di una normale perdita attraverso BGP common RT, che non consuma memoria aggiuntiva.

Con questa configurazione, le perdite di percorso vengono utilizzate sul CORE-4 per fornire l'accesso da CUST-A e da CUST-B a COM (e 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
```

A questo punto, le route replicate non vengono propagate nel protocollo IGP (Interior Gateway Protocol), quindi solo CE-A-2 e CE-B-2 hanno accesso al servizio COM (100.1.1.100), non CE-A-1 e CE-B-1.

È inoltre possibile utilizzare la route di perdite da o verso una tabella 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. Definire la propagazione delle perdite di route. Le route perse non vengono duplicate nel RIB VRF di destinazione. In altre parole, non fanno parte del VRF RIB di destinazione. La ridistribuzione normale tra i processi del router non funziona, quindi è necessario definire esplicitamente la connessione VRF del RIB a cui appartiene la route:

```

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

```

Le route trafugate da VRF-Y vengono ridistribuite nel processo OSPF eseguito in VRF-X. Di seguito è riportato un esempio di 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

```

In questo caso non è necessaria la mappa dei percorsi, poiché in VRF COM è presente un solo percorso connesso. È ora possibile raggiungere il servizio COM (100.1.1.100) da CE-A-1 e 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#

```

## Ottimizzare il trunk della rete virtuale

In questa sezione vengono fornite informazioni che è possibile utilizzare per ottimizzare il trunk della rete virtuale.

### Elenco trunk

Per impostazione predefinita, tutti i VRF configurati con un tag VNET sono consentiti su tutti i trunk

VNET. Un elenco di trunk consente di specificare l'elenco di VRF autorizzate nel trunk della rete virtuale:

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

**Nota:** Deve esistere una linea per ogni VRF consentito.

Ad esempio, CORE-1 è ottimizzato per il VRF CUST-B sul trunk VNET tra CORE-1 e CORE-2:

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

Come mostrato, il peer OSPF per VRF CUST-B attraverso il trunk si interrompe:

```
%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 sottointerfaccia per VRF CUST-B viene rimossa:

```
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
!
```

## Attributi per-VRF Trunk

Per impostazione predefinita, le sottointerfacce dot1q ereditano i parametri dell'interfaccia fisica in modo che le sottointerfacce di tutti i VRF abbiano gli stessi attributi (ad esempio, costo e autenticazione). È possibile regolare i parametri del trunk per tag di rete virtuale:

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

È possibile ottimizzare i seguenti parametri:

```
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

In questo esempio, il costo OSPF per VRF per CORE-1 viene modificato, quindi il percorso CORE-2 viene utilizzato per CUST-A e il percorso CORE-3 per CUST-B (il costo predefinito è 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#
```

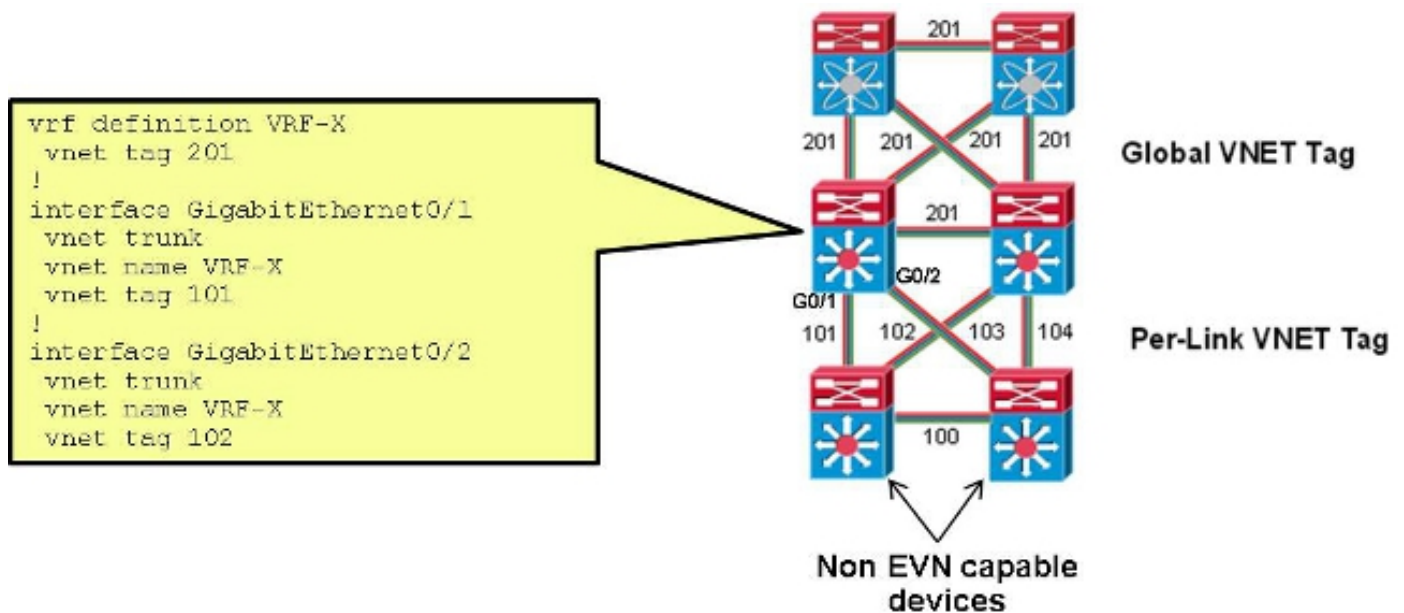
CORE-1#show ip route vrf CUST-B 10.2.2.0

```
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#
```

**Tag rete virtuale per collegamento**

Per impostazione predefinita, il tag VNET definito nella definizione VRF viene utilizzato per tutti i trunk. Tuttavia, è possibile utilizzare un tag VNET diverso per ogni trunk.

Nell'esempio viene descritto uno scenario in cui si è connessi a un dispositivo non compatibile con EVN e si utilizza VRF-Lite con un trunk manuale e il tag VNET globale viene utilizzato da un'altra VLAN:



Con questa configurazione, il tag VNET utilizzato sul trunk tra CORE-1 e CORE-2 per CUST-A viene modificato da **100** a **101**:

```

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

Dopo che questa modifica si è verificata sul CORE-1, viene creata una nuova sottointerfaccia:

```

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 la modifica si verifica solo su un'estremità, la connettività viene interrotta nel VRF associato e l'OSPF diventa inattivo:

```

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

Se lo stesso tag VNET viene utilizzato su CORE-2, la connettività viene ripristinata e il tag dot1q 101 viene utilizzato su tale trunk, mentre 100 viene ancora utilizzato sul trunk da CORE-1 a CORE-3:

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

## Verifica

Attualmente non è disponibile una procedura di verifica per questa configurazione.

## Risoluzione dei problemi

Al momento non sono disponibili informazioni specifiche per la risoluzione dei problemi di questa configurazione.

## Informazioni correlate

- [Easy Virtual Network - Semplificazione della virtualizzazione della rete di layer 3](#)
- [Documentazione e supporto tecnico – Cisco Systems](#)