Catalyst 9000 Series 스위치에서 EVPN에 대한 BGP VRF Auto RD Auto RT 구성

목차

소개

이 문서에서는 Catalyst 9000 Series 스위치의 EVPN에서 BGP VRF Auto RD 및 Auto RT를 위한 EVPN 간소화 CLI에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- 기본 BGP 컨피그레이션
- 기본 VRF 컨피그레이션
- 기본 EVPN 컨피그레이션

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Catalyst 9300
- Catalyst 9400
- Catalyst 9500
- Catalyst 9600
- Cisco IOS® XE 17.12.1 이상

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

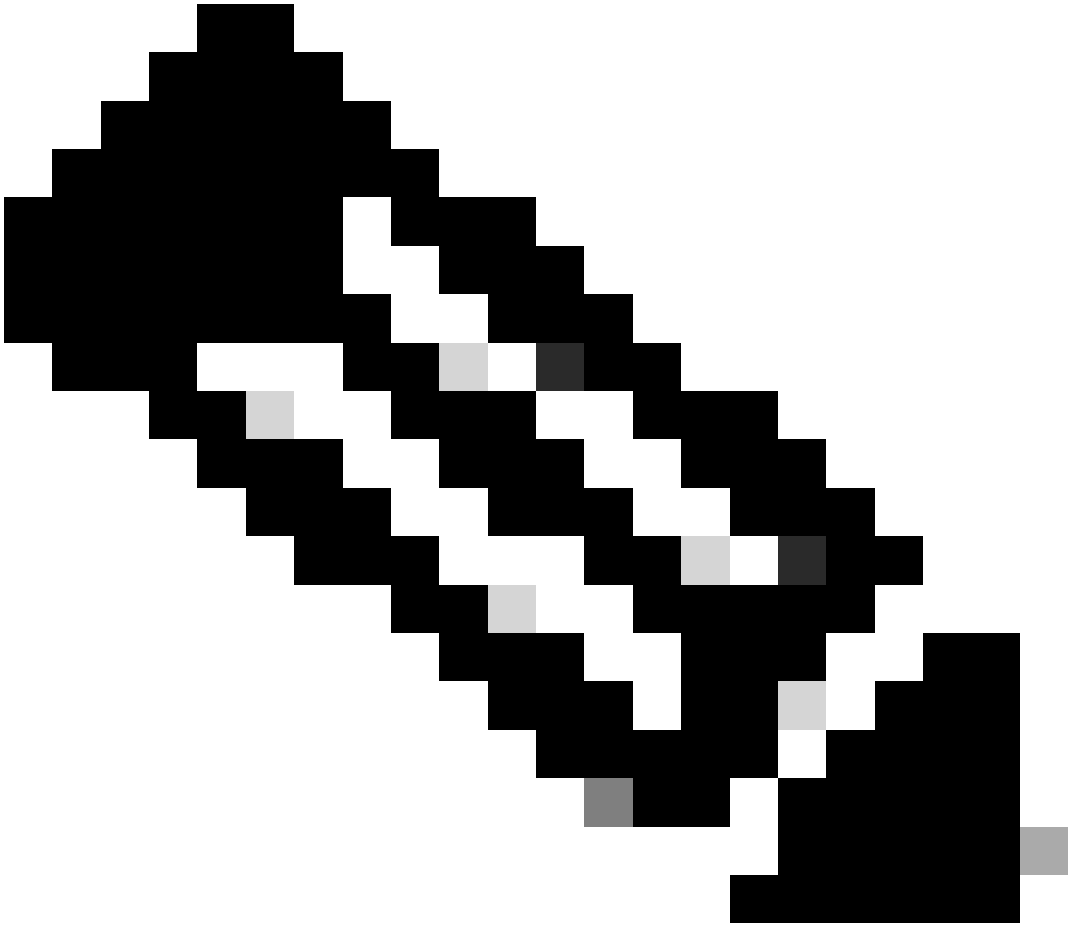
레이어 3 EVPN 구축에는 RD(Route Distinguisher) 및 RT(Route-Targets)를 비롯한 여러 컨피그레이션 옵션이 포함된 VRF 컨피그레이션이 포함됩니다.

- BGP VRF Auto RD Auto RT 기능을 도입하기 전에 BGP EVPN 사용을 위한 특정 VRF를 설정하려면 최소 5개의 컨피그레이션 라인(RD의 경우 1, RT의 경우 4)이 필요합니다.
- BGP VRF Auto RD Auto RT를 사용하면 2개의 라인으로만 이 작업을 수행할 수 있습니다(글로벌 VRF rd-auto가 활성화된 경우 VRF당 1개의 라인).
- Auto RD와 Static RD 사이에는 기능적 차이가 없습니다. 각 RD는 하나의 지정된 라우터 또는 스위치 내에서 고유해야 합니다.
- Auto RT와 Static RT의 기능적 차이점은 Auto RT는 가져오기 및 내보내기, 일반 및 스티칭에 대해 하나뿐이며 동일하다는 것입니다. 반면 Static RT는 0에서 다수로 구성할 수 있습니다.
- 또한 Auto RT는 특정 VRF 내에서 고정 RT와 공존할 수 있습니다(이 기능 이전에 기존 고정 RT에 추가하여 Auto RT를 구성할 수 있음).

자동 RD는 BGP 라우터 ID와 내부 생성 고유 번호로 구성됩니다. 예를 들어, BGP 라우터 ID가 192.168.1.1이면 자동 RD는 "192.168.1.1:1"과 같습니다.

- 자동 RT는 BGP AS 번호와 구성 중인 vnid로 구성됩니다. 예를 들어 BGP AS 번호가 65000이고 vnid가 123으로 구성된 경우 자동 RT는 "65000:123"이 됩니다.
- 이는 가져오기 및 내보내기, 일반 및 스티칭 경로 대상 모두에 사용됩니다.
- BGP AS가 4바이트인 경우 AS_TRANS가 대신 사용됩니다(23456).

컨피그레이션을 간소화하는 기능은 구축이 실현 가능하기 위해 매우 바람직하며(필요 없는 경우), 이미 BGP EVPN 패브릭에 널리 채택되었습니다. 이 기능은 EVPN에 유용합니다. 많은 VRF가 특정 leaf에 구성된 Spine-Leaf 토폴로지에서 광범위하고 복잡한 컨피그레이션의 쓰기 및 유지 보수를 피할 수 있기 때문입니다.



참고: 이 기능은 새 CLI를 도입합니다.

용어

VRF	가상 라우팅 전달	다른 VRF 및 전역 IPv4/IPv6 라우팅 도메인과 구분되는 레이어 3 라우팅 도메인을 정의합니다.
AF	주소군	어떤 유형 접두사 및 라우팅 정보 BGP가 처리되는지 정의합니다.
AS	자동 시스템	단일 엔터티 또는 조직에서 모두 관리, 제어 및 감독하는 네트워크 또는 네트워크 컬렉션에 속하는 인터넷 라우팅 가능 IP 접두사 집합입니다
RD	경로 구분자	BGP가 서로 다른 VRF에서 접두사를 구별할 수 있도록 허용

RT	경로 대상	경로 대상은 라우팅 업데이트를 제한하는 데 사용됩니다. 디바이스에서 가져올 수 있는 접두사를 결정합니다.
EVPN	이더넷 가상 사설망	BGP가 레이어 2 MAC 및 레이어 3 IP 정보를 전송하도록 허용하는 확장은 EVPN이며 에서는 VXLAN 오버레이 네트워크와 관련된 연결 정보를 배포하기 위한 프로토콜로 MP-BGP(Multi-Protocol Border Gateway Protocol)를 사용합니다.
VXLAN	가상 확장 LAN(Local Area Network)	VXLAN은 VLAN과 STP의 내재적 한계를 극복하기 위해 설계되었습니다. 이는 VLAN과 동일한 이더넷 레이어 2 네트워크 서비스를 제공하되 더 높은 유연성을 제공하는 IETF 표준[RFC 7348]입니다. 기능적으로 레이어 3 언더레이 네트워크에서 가상 오버레이로 실행되는 MAC-in-UDP 캡슐화 프로토콜입니다.

구성

글로벌 VRF RD-auto

<#root>

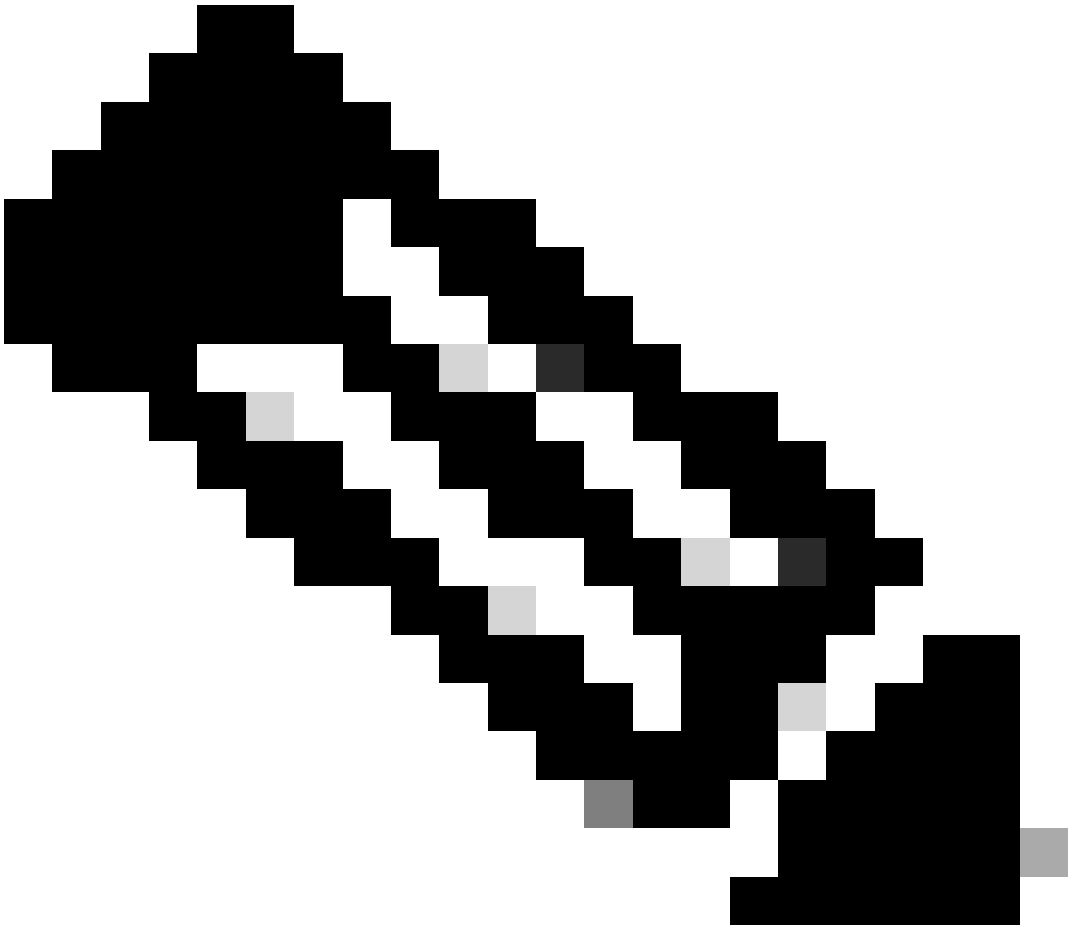
```
Leaf-01#
sh run | include vrf rd-auto
vrf rd-auto                                          <-- Enable Auto RD for all the VRFs
Leaf-01#
sh run | section vrf definition blue

vrf definition blue
vnid 123 evpn-instance                               <-- Enable Auto RT
!
address-family ipv4                                  <-- address-family needs to be specified
 route-target 100:123                                <-- Optionally can have static route-target as requ
exit-address-family
!
```

VRF rd-auto 컨피그레이션당

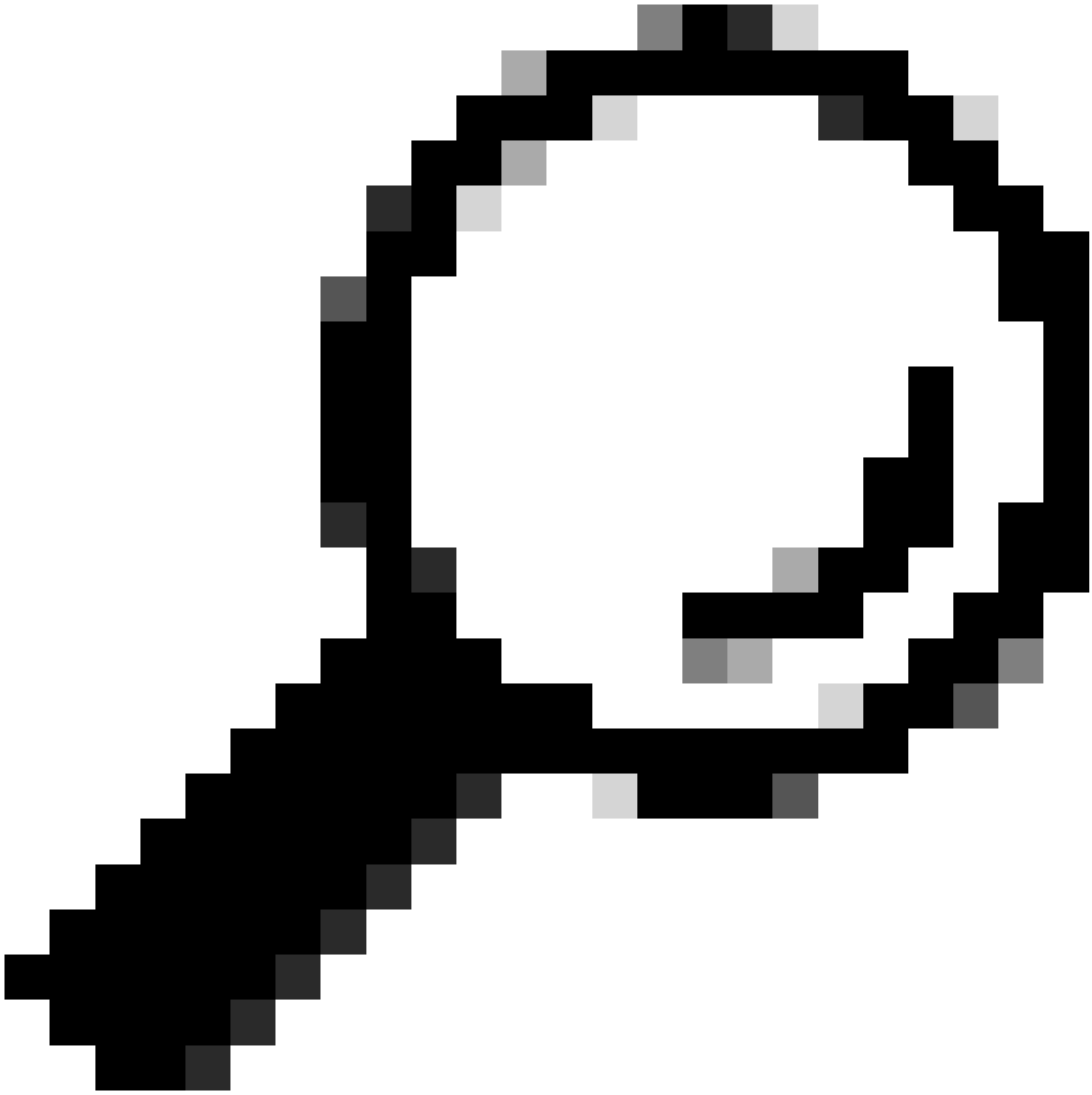
<#root>

```
Leaf-01#
sh run | section vrf definition green
```

```
vrf definition green
rd-auto                                   <-- Enable Auto RD for this VRF green
vnid 35 evpn-instance                     <-- Enable Auto RT
!
address-family ipv4                       <-- address-family needs to be specified
exit-address-family
!
address-family ipv6
exit-address-family
```



참고: 서로 다른 VRF에 대해 고정 및 자동 RD를 가질 수 있지만, 자동 RD가 먼저 할당된 경우 고정 RD에 자동 RD와 동일한 실제 RD가 없어야 합니다.



팁: 현재 고정 RD를 삭제하면 VRF에 구성된 경로 대상의 컨피그레이션은 물론 BGP IPv4 및/또는 IPv6 VRF 주소군(및 그 아래의 관련 컨피그레이션)도 삭제됩니다. 따라서 자동 RD를 삭제하면 유사한 동작이 발생합니다. 반드시 필요한 경우가 아니면 RD의 삭제를 트리거하지 않는 것이 좋습니다. RD를 변경하면(즉, 기존 RD를 정적 또는 자동에서 삭제한 다음 새 RD를 정적 또는 자동에서 추가하면 많은 비용이 소요되며 명령이 통과하기 위해 지연 시간이 필요합니다)

혼합 정적 RD 및 자동 RD

<#root>

```
vrf rd-auto
vrf definition green                                    <-- This VRF green uses auto RD
 vnid 35 evpn-instance
```

```
!
 address-family ipv6
 exit-address-family
vrf definition red                                          <-- This VRF red uses static RD
 rd-auto disable
 rd 100:1
 !
 address-family ipv4
  route-target export 100:1
  route-target import 100:1
  route-target export 100:1 stitching
  route-target import 100:1 stitching
 exit-address-family
```

BGP 주소군 IPv4 Vrf 및 Ipv6 Vrf

(이 컨피그레이션 예는 기존 기능의 요약)

<#root>

Leaf-01#

show run | sec r bgp

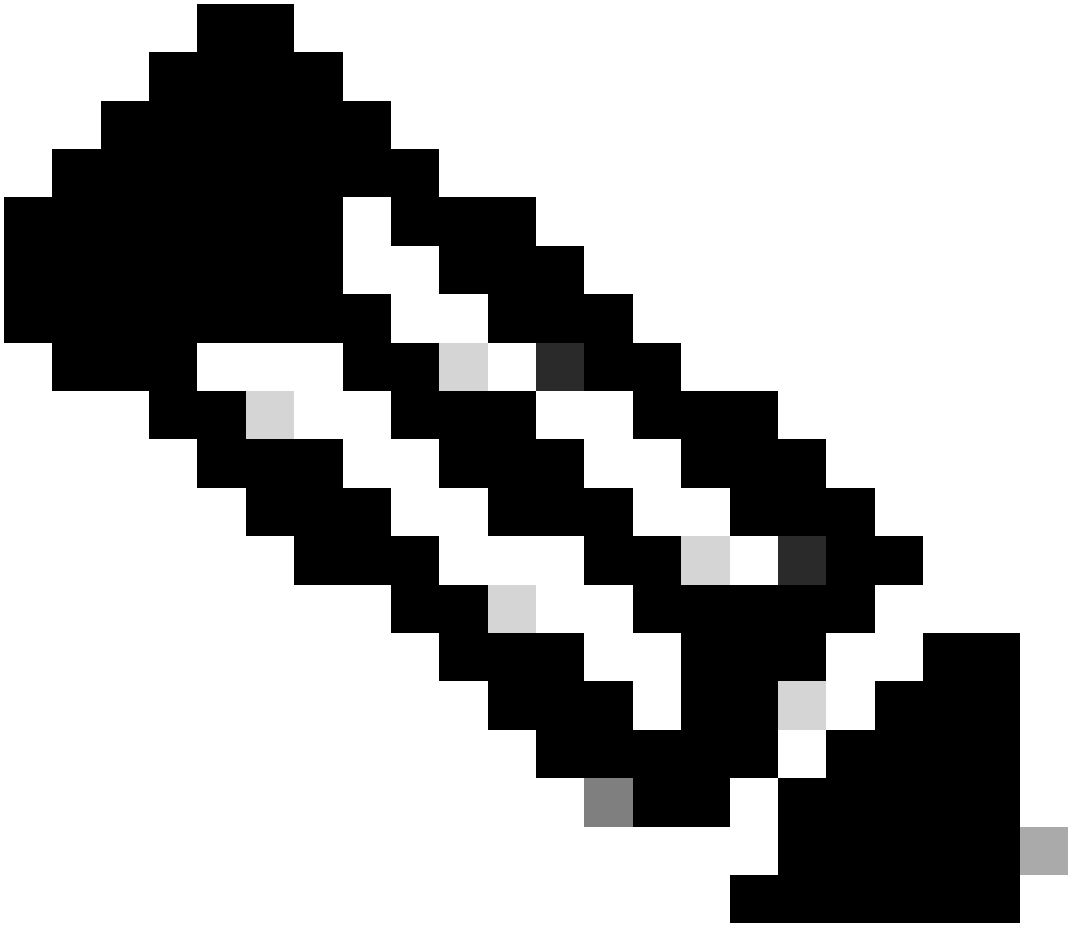
router bgp 65000

<-- Required for Auto RT

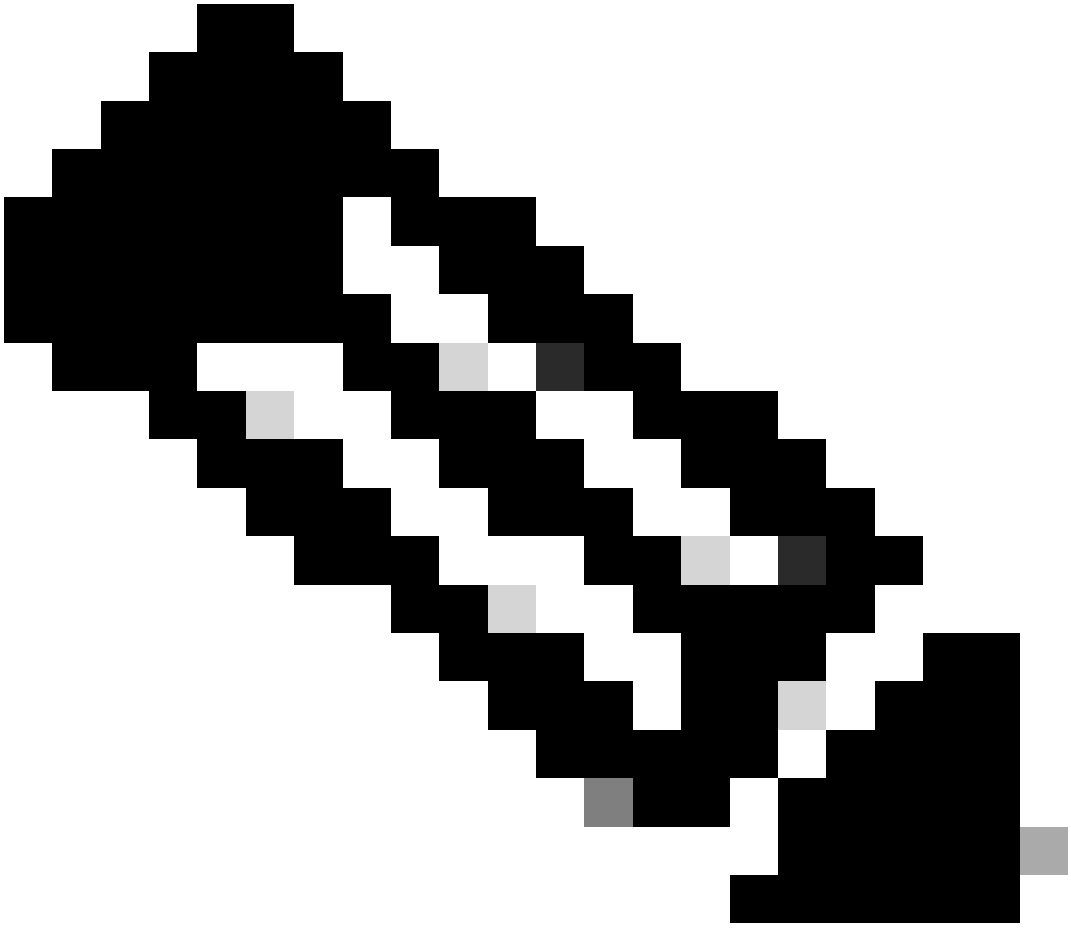
bgp router-id 192.168.1.1

<-- Required for Auto RD

```
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 192.168.1.2 remote-as 65000
 neighbor 192.168.1.2 update-source Loopback0
 neighbor 192.168.1.3 remote-as 65001
 neighbor 192.168.1.3 update-source Loopback0
 !
 address-family ipv4 vrf green
  advertise l2vpn evpn
  redistributed connected
 exit-address-family
 !
 address-family ipv6 vrf green
  advertise l2vpn evpn
  redistribute connected
 exit-address-family
```



참고: 다른 Spine Route Reflector의 컨피그레이션은 동일하므로 이 섹션에서는 반복하지 않습니다



참고: 다른 EVPN leaf에서는 정적 RD 또는 RT 컨피그레이션을 사용할 수 있습니다. RT가 일치하는 경우 EVPN 접두사는 서로 가져오거나 내보낼 수 있습니다.

다음을 확인합니다.

리프

Leaf를 확인하여 auto RD를 보유합니다.

<#root>

VTEP1#

show vrf blue

```
  Name                             Default RD            Protocols   Interfaces
  blue                             192.168.1.1:1(auto)   ipv4        Vl34
                                                                     Lo101
```

```
                                                                  Et1/1
                                                                  Vl4
                                                                  Vl15
```

<#root>

```
VTEP1#

show vrf green

  Name                              Default RD            Protocols   Interfaces
  green                             192.168.1.1:2(auto)   ipv6        Lo102
                                                                      Et1/2
                                                                      Vl5
                                                                      Vl13
```

<#root>

```
VTEP1#

show vrf detail blue

VRF blue (VRF Id = 2); default RD 192.168.1.1:1(auto); default VPNID
  New CLI format, supports multiple address-families
  vnid: 123 evpn-instance vni 35000 core-vlan 34
  Flags: 0x180C
  Interfaces:
    Vl34                    Lo101                   Et1/1
    Vl4                     Vl15
Address family ipv4 unicast (Table ID = 0x2):
  Flags: 0x0
  Export VPN route-target communities
    RT:100:123              RT:65000:123 (auto)
  Import VPN route-target communities
    RT:100:123              RT:65000:123 (auto)
  Export VPN route-target stitching communities
    RT:65000:123 (auto)
  Import VPN route-target stitching communities
    RT:65000:123 (auto)
  No import route-map
  No global export route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
Address family ipv6 unicast not active
Address family ipv4 multicast not active
Address family ipv6 multicast not active
```

<#root>

```
VTEP1#

show vrf detail green

VRF green (VRF Id = 4); default RD 192.168.1.1:2(auto); default VPNID
  New CLI format, supports multiple address-families
```

```
  vnid: 35 evpn-instance
  Flags: 0x380C
  Interfaces:
    Lo102                   Et1/2                   Vl5
    Vl13
Address family ipv4 unicast not active
Address family ipv6 unicast (Table ID = 0x1E000002):
  Flags: 0x0
  Export VPN route-target communities
    RT:65000:35 (auto)
  Import VPN route-target communities
    RT:65000:35 (auto)
  Export VPN route-target stitching communities
    RT:65000:35 (auto)
  Import VPN route-target stitching communities
    RT:65000:35 (auto)
  No import route-map
  No global export route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
Address family ipv4 multicast not active
Address family ipv6 multicast not active
```

문제 해결

디버그

VRF auto RD auto RT에 문제가 있는 경우 디버그를 사용하여 문제에 대한 자세한 내용을 확인할 수 있습니다

관련 디버깅 활성화

<#root>

Leaf-01#

debug ip bgp autordrt

Leaf-01#

debug vrf create

Leaf-01#

debug vrf delete

표시 디버그 정보

<#root>

VTEP1#

show debug

VRF Manager:

VRF creation debugging is on
VRF deletion debugging is on

Packet Infra debugs:
Ip Address Port
--|----------
IP routing:

BGP auto rd rt debugging is on

각 컨피그레이션 단계에서 생성된 디버그를 관찰합니다

<#root>

Leaf-01(config)#

vrf definition test

*Jun 26 08:19:44.173: LID: Get id @0x7F4414FE4A18 - current A [1..2705] (checking enabled)
*Jun 26 08:19:44.173: LID: AVAIL (verified) - id A
*Jun 26 08:19:44.173: vrfmgr: VRF test: Created vrf_rec with vrfid 0xA
*Jun 26 08:19:44.173: BGP: VRF config event of

rd-auto change for vrf test

*Jun 26 08:19:44.173: BGP-VPN: bgp vpn global

rd-auto for vrf test assigns rd of 192.168.1.1:6

*Jun 26 08:19:44.173: BGP: VRF config event of

vnid change for vrf test

Leaf-01(config-vrf)#

vnid 246 evpn-instance

% vnid 246 evpn-instance auto (vni 0 core-vlan 0) is configured in "vrf test"
*Jun 26 08:20:03.466: BGP: VRF config event of

vnid change for vrf test

Leaf-01(config-vrf)#

address-family ipv4

*Jun 26 08:20:12.276: vrfmgr: VRF test ipv4 unicast: Received topology create notification
*Jun 26 08:20:12.276: vrfmgr: VRF test ipv4 multicast: Received topology create notification
*Jun 26 08:20:12.276: vrfmgr: VRF test ipv4 unicast:

Created vrf_sub_rec with vrfid 0xA, tableid 0xA

*Jun 26 08:20:12.276: BGP: VRF config event of vnid change for vrf test
*Jun 26 08:20:12.276: BGP: afi 0 vrf

test vnid 246 RT assign

```
*Jun 26 08:20:12.276: BGP: vrf assign auto import stitching rt for VRF test
*Jun 26 08:20:12.276: BGP: vrf assign auto export stitching rt for VRF test

Leaf-01(config-vrf-af)#
address-family ipv6

*Jun 26 08:20:20.949: vrfmgr: VRF test ipv6 unicast: Received topology create notification
*Jun 26 08:20:20.949: vrfmgr: VRF test ipv6 multicast: Received topology create notification
*Jun 26 08:20:20.949: vrfmgr: VRF test ipv6 unicast:
Created vrf_sub_rec with vrfid 0xA, tableid 0x1E000004

*Jun 26 08:20:20.949: BGP: VRF config event of vnid change for vrf test
*Jun 26 08:20:20.949: BGP:
afi 0 vrf test vnid 246 RT assign

*Jun 26 08:20:20.949: BGP: vrf assign auto import stitching rt for VRF test
*Jun 26 08:20:20.949: BGP: vrf assign auto export stitching rt for VRF test
*Jun 26 08:20:20.949: BGP:
afi 1 vrf test vnid 246 RT assign

*Jun 26 08:20:20.949: BGP: vrf assign auto import stitching rt for VRF test
*Jun 26 08:20:20.949: BGP: vrf assign auto export stitching rt for VRF test

Leaf-01(config-vrf-af)#
do sh vrf detail test

VRF test (VRF Id = 10)

; default
RD 192.168.1.1:6(auto)

; default VPNID
<-- VRF ID = 10 (hex 0xA) | auto RD assigned matches debug "assigns rd of 192.168.1.1:6"

  New CLI format, supports multiple address-families


vnid: 246

 evpn-instance
  Flags: 0x180C
  No interfaces
Address family ipv4 unicast (Table ID = 0xA):
  Flags: 0x0
  Export VPN route-target communities
    RT:65000:246 (auto)
  Import VPN route-target communities
    RT:65000:246 (auto)
  Export VPN route-target stitching communities
    RT:65000:246 (auto)
  Import VPN route-target stitching communities
    RT:65000:246 (auto)
  No import route-map
  No global export route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
Address family ipv6 unicast
(Table ID = 0x1E000004)
```

```
:

<-- ID matches debug

 "

Created vrf_sub_rec with vrfid 0xA, tableid 0x1E000004"

  Flags: 0x0
  Export VPN route-target communities
    RT:65000:246 (auto)
  Import VPN route-target communities
    RT:65000:246 (auto)
  Export VPN route-target stitching communities
    RT:65000:246 (auto)
  Import VPN route-target stitching communities
    RT:65000:246 (auto)
  No import route-map
  No global export route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
Address family ipv4 multicast not active
Address family ipv6 multicast not active

Leaf-01(config-vrf-af)#

do sh run vrf test

Building configuration...

Current configuration : 145 bytes
vrf definition test
 vnid 246 evpn-instance
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
```

Catalyst 및 Nexus 상호 운용성

문제

기본적으로 Nexus는 vni 기반 경로 대상(ASN:VNI)을 할당하고, Catalyst는 evi 기반 경로 대상(ASN:EVI)을 할당합니다.

경로-대상이 일치하지 않을 경우 다음과 같은 증상을 관찰할 수 있습니다.

- L2VPN EVPN에 대한 BGP 연결은 BGP 테이블에서 설정되며 유형 3 경로가 표시됩니다
- NVE 피어링이 설정되지 않음
- 터널 인접성이 완료되지 않음

교정

이 interop 문제를 해결하는 몇 가지 옵션이 있습니다.

1. 한쪽에서 수동 경로 대상을 구성하여 일치시킵니다.
2. 'route-target auto vni'를 사용하여 vni 기반 경로 대상을 할당하도록 C9500 구성

l2vpn evpn 섹션에서 이러한 cli(옵션 2의 경우)를 적용합니다

<#root>

```
address-family l2vpn evpn
```

rewrite-evpn-rt-asn <---

관련 정보

- BGP EVPN VXLAN 컨피그레이션 가이드, Cisco IOS XE Dublin 17.11.x(Catalyst 9500 스위치)
- 기술 지원 및 문서 – Cisco Systems

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.