

在Catalyst 9000系列交换机上为EVPN配置BGP VRF自动RD自动RT

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[术语](#)

[配置](#)

[全局VRF RD-auto](#)

[每个VRF rd-auto配置](#)

[混合静态RD和自动RD](#)

[BGP地址系列IPv4 Vrf和IPv6 Vrf](#)

[验证](#)

[枝叶](#)

[故障排除](#)

[调试](#)

[Catalyst和Nexus的互操作性](#)

[问题](#)

[补救](#)

[相关信息](#)

简介

本文档介绍Catalyst 9000系列交换机上EVPN中BGP VRF自动路由器和自动RT的EVPN简化CLI。

先决条件

要求

Cisco 建议您了解以下主题：

- 基本BGP配置
- 基本VRF配置
- 基本EVPN配置

使用的组件

本文档中的信息基于以下软件和硬件版本：

- Catalyst 9300
- Catalyst 9400
- Catalyst 9500
- Catalyst 9600
- Cisco IOS® XE 17.12.1及更高版本

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

第3层EVPN部署涉及具有许多配置选项的VRF配置，包括但不限于路由区分符(RD)和路由目标(RT)。

- 在引入BGP VRF自动RD自动RT功能之前，需要至少5条配置行（1条用于RD，4条用于RT）来设置特定的VRF以用于BGP EVPN。
- 使用BGP VRF自动RD自动RT，这只能通过2条线路来实现（如果启用全局VRF rd-auto，则每个VRF可能有一条线路）。
- 自动RD和静态RD在功能上没有任何区别。每个RD必须在一个给定的路由器或交换机内是唯一的。
- Auto RT和static RT之间的功能区别在于，Auto RT对于导入和导出、常规和拼接，只为一个且相同，而静态RT可以配置为零到多。
- 此外，Auto RT可以在任何特定VRF内与静态RT共存（除了此功能之前的现有静态RT外，您还可以配置Auto RT）。

自动RD包括BGP路由器ID加上内部生成的唯一编号，例如，如果BGP路由器ID是192.168.1.1，则自动RD将类似于“192.168.1.1:1”。

- 自动RT将包括BGP AS编号以及配置的vnid。例如：如果BGP AS编号为65000，并且vniid配置为123，则自动RT将为“65000：123”。
- 这适用于导入和导出、常规和拼接路由目标。
- 如果BGP AS为4字节，则改用AS_TRANS，即23456。

简化配置的功能对于部署的可行性是非常理想的（如果不是必要的话），并且已广泛用于BGP EVPN交换矩阵。此功能是EVPN的理想选择，因为它有助于避免在主干-枝叶拓扑中编写和维护广泛而复杂的配置，其中在特定枝叶中配置了许多VRF。

注意：此功能引入了新的CLI。

术语

VRF	虚拟路由转发	定义与其他VRF和全局IPv4/IPv6路由域分离的第3层路由域
AF	地址系列	定义BGP处理的前缀类型和路由信息
AS	自治系统	一组属于一个网络或一组网络的可路由IP前缀，全部由单个实体或组织管理、控制和监督
RD	路由区分符	允许BGP区分不同VRF中的前缀

RT	路由目标	路由目标用于限制路由更新。确定允许设备导入的前缀
EVPN	以太网虚拟专用网络	允许BGP传输第2层MAC和第3层IP信息的扩展是EVPN和 使用多协议边界网关协议(MP-BGP)作为协议，以分发有关VXLAN重叠网络的可达性信息。
VXLAN	虚拟可扩展LAN (局域网)	VXLAN旨在克服VLAN和STP的固有局限性。推荐的IETF标准[RFC 7348]与VLAN提供相同的以太网第2层网络服务，但灵活性更高。从功能上讲，它是MAC-in-UDP封装协议，在第3层底层网络上作为虚拟重叠运行。

配置

全局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 req
exit-address-family
!

```

每个VRF rd-auto配置

```

<#root>

Leaf-01#

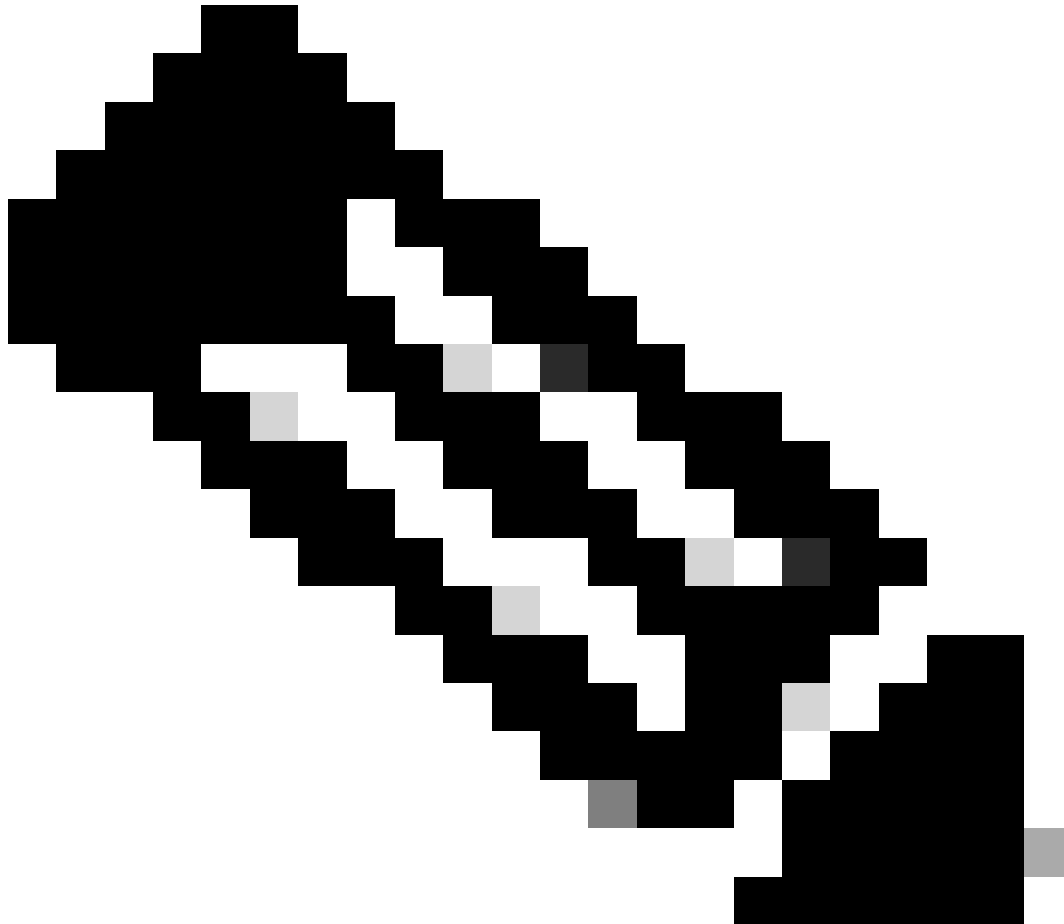
sh run | section vrf definition green

vrf definition green
rd-auto                                     <-- Enable Auto RD for this VRF green
vnid 35 evpn-instance                      <-- Enable Auto RT
!

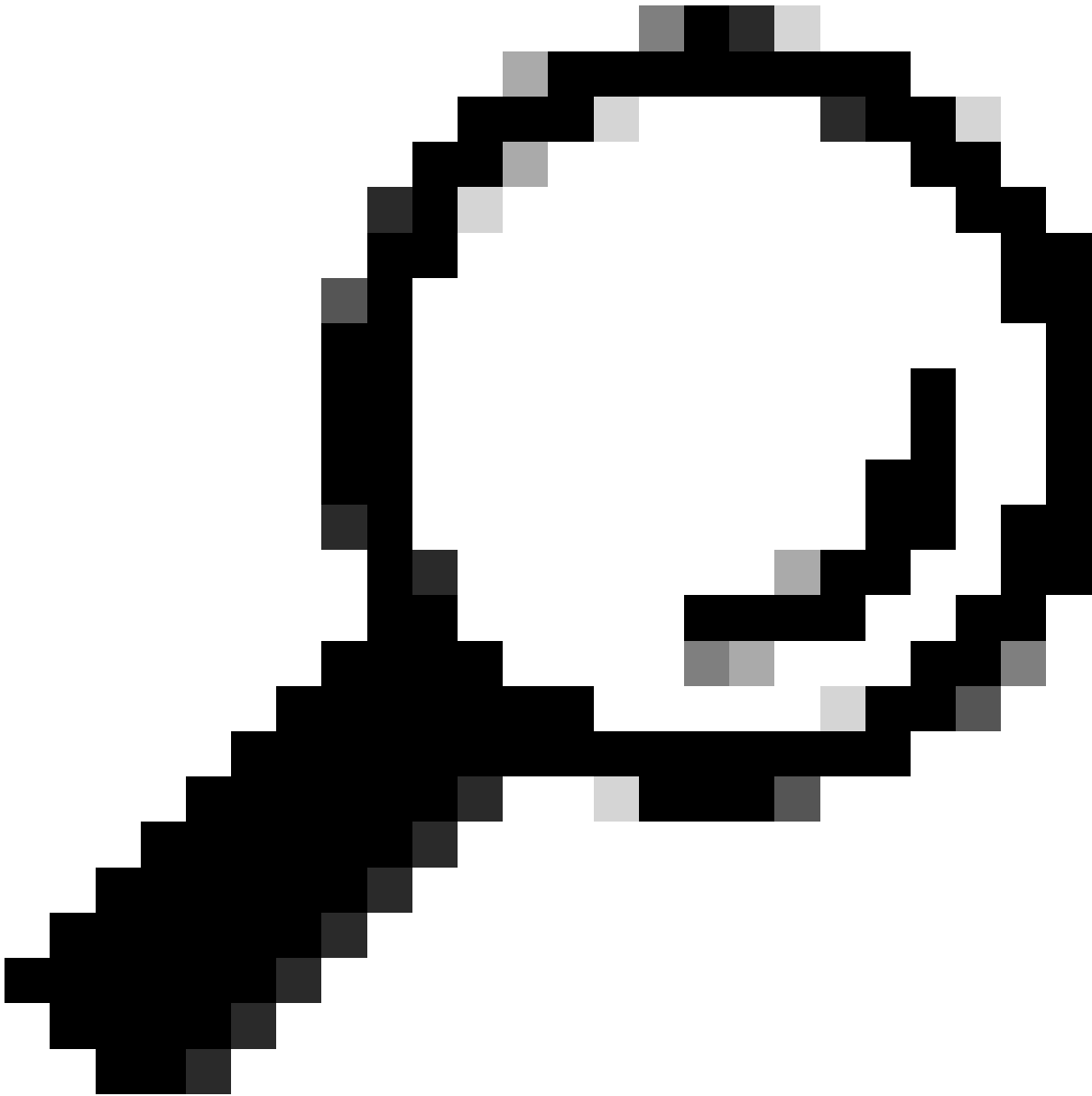
```

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

<-- address-family needs to be specified



注：可以针对不同的VRF使用静态和自动RD，但如果首先分配了自动RD，则静态RD的实际值不能与自动RD相同。



提示：当前删除静态RD将删除在VRF中配置的路由目标的配置，以及BGP IPv4和/或IPv6 VRF地址系列（以及下面的关联配置）。因此，删除自动RD将具有类似行为。除非绝对必要，否则建议不要触发删除RD。更改RD（即删除现有RD，静态或自动，然后添加新的RD，静态或自动，成本高昂，并且需要延迟时间才能让命令通过）

混合静态RD和自动RD

```
<#root>
```

```
vrf rd-auto
vrf definition green
  vnid 35 evpn-instance
!
address-family ipv6
exit-address-family
```

```
<-- This VRF green uses auto RD
```

```
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地址系列IPv4 Vrf和Ipv6 Vrf

(此配置示例是对现有功能的回顾)

```
<#root>
```

```
Leaf-01#
```

```
show run | sec r bgp
```

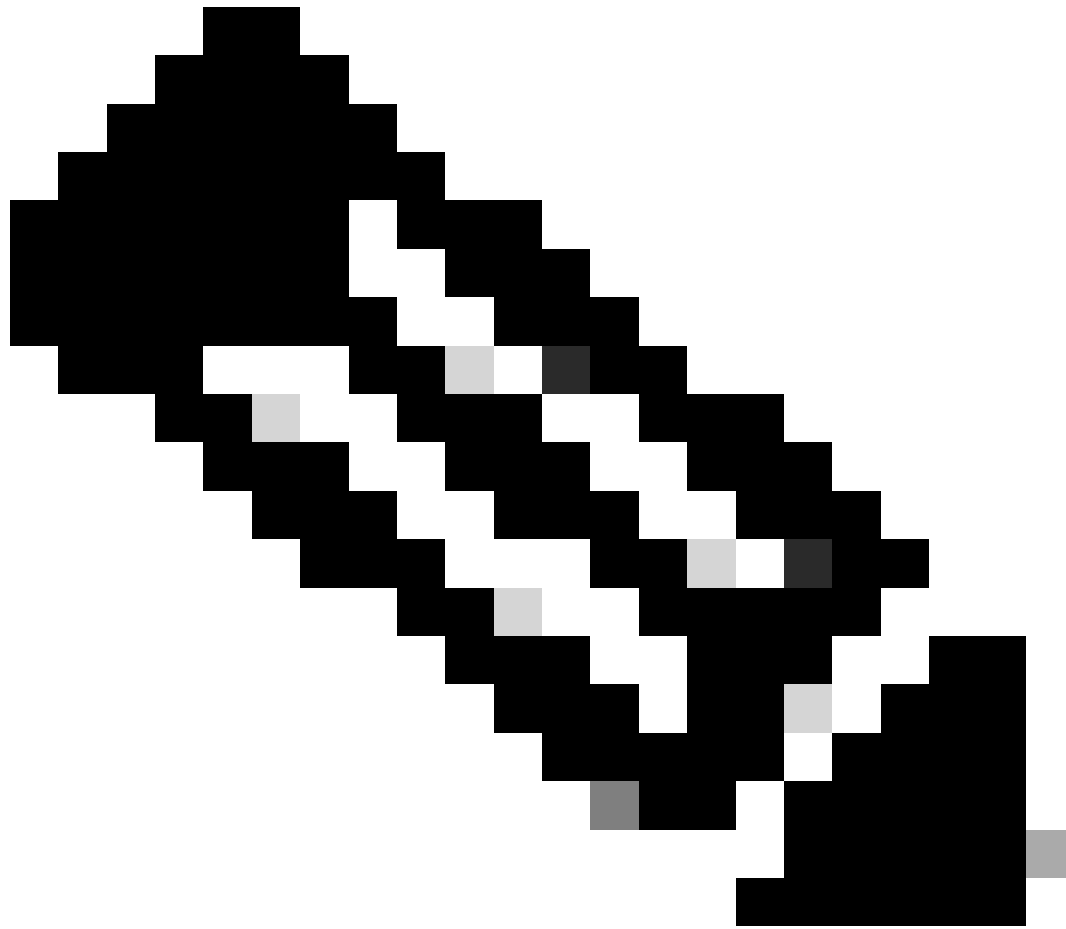
```
router bgp 65000
```

```
<-- Required for Auto RT
```

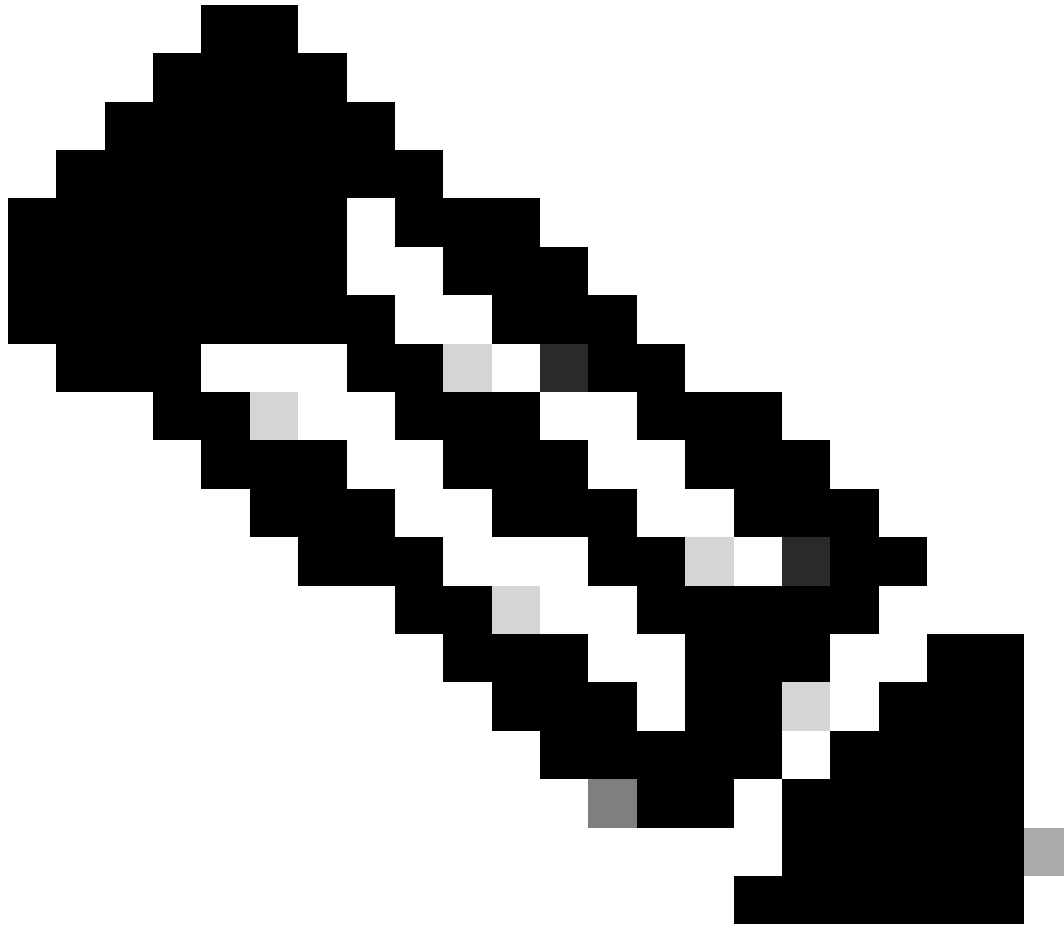
```
bgp router-id 192.168.1.1
```

```
<-- 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
```



注意：其他主干路由反射器的配置相同，因此在本部分不再重复



注意：其他EVPN枝叶可以使用静态RD或RT配置。只要RT匹配，EVPN前缀就能相互导入/导出。

验证

枝叶

验证枝叶，以自动进行rd

```
<#root>
```

```
VTEP1#
```

```
show vrf blue
```

Name	Default RD	Protocols	Interfaces
blue	192.168.1.1:1(auto)	ipv4	V134 Lo101

Et1/1
V14
V115

<#root>

VTEP1#

show vrf green

Name	Default RD	Protocols	Interfaces
green	192.168.1.1:2(auto)	ipv6	Lo102 Et1/2 V15 V113

<#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:

V134	Lo101	Et1/1
V14	V115	

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                V15
  V113
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
```

故障排除

调试

如果VRF自动RD自动RT出现问题，您可以使用调试来查看有关该问题的详细信息

启用相关调试

```
<#root>
Leaf-01#
debug ip bgp autordrt
Leaf-01#
debug vrf create
Leaf-01#
debug vrf delete
```

显示 调试信息

```
<#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
```

观察每个配置步骤中生成的调试

```
<#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和Nexus的互操作性

问题

默认情况下，Nexus分配基于vni的路由目标(ASN : VNI)，而Catalyst分配基于evi的路由目标(ASN : EVI)。

当route-target不匹配时，您可以观察到如下症状：

- L2VPN EVPN的BGP连接已建立，并且BGP表中可见第3类路由
- 未建立NVE对等
- 隧道邻接关系仍未完成

补救

有两个选项可用于解决此互操作性问题

1. 在一端配置手动路由目标，使其匹配
2. 配置C9500以使用“route-target auto vni”分配基于vni的路由目标

在l2vpn evpn部分下应用这些cli (用于选项2)

```
<#root>
```

```
address-family l2vpn evpn
```

```
rewrite-evpn-rt-asn <---
```

相关信息

- [BGP EVPN VXLAN配置指南，思科IOS XE都柏林17.11.x \(Catalyst 9500交换机 \)](#)
- [技术支持和文档 - Cisco Systems](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。