

Catalyst 9000シリーズスイッチでのEVPNに対するBGP VRF Auto RD Auto RTの設定

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[用語](#)

[設定](#)

[グローバルVRF RD-auto](#)

[VRFごとのrd-auto設定](#)

[スタティックRDと自動RDの混在](#)

[BGPアドレスファミリIPv4 VrfおよびIpv6 Vrf](#)

[確認](#)

[リーフ](#)

[トラブルシューティング](#)

[デバッグ](#)

[CatalystとNexusの相互運用性](#)

[問題](#)

[修復方法](#)

[関連情報](#)

はじめに

このドキュメントでは、Catalyst 9000シリーズスイッチ上のEVPNにおけるBGP VRF Auto RDおよびAuto RTのEVPN簡素化CLIについて説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- 基本的なBGP設定
- 基本的なVRF設定
- 基本的なEVPN設定

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

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

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

レイヤ3 EVPNの導入には、ルート識別子(RD)やルートターゲット(RT)など、多くの設定オプションを備えたVRF設定が含まれます。

- BGP VRF Auto RD Auto RT機能を導入する前に、BGP EVPNを使用するために特定のVRFを設定するには、少なくとも5つの設定行（RD用に1つ、RT用に4つ）が必要です。
- BGP VRF Auto RD Auto RTでは、これを2回線だけで実現できます（グローバルVRF rd-autoが有効な場合、VRFごとに1回線の可能性があります）。
- 自動RDと静的RDの機能上の違いはありません。各RDは、特定の1台のルータまたはスイッチ内で一意である必要があります。
- Auto RTとStatic RTの機能的な違いは、Auto RTはインポートとエクスポート、通常のステイッチングとステイッチングに対して1つだけであり、Static RTは0対多で設定できることです。
- また、Auto RTは、特定のVRF内でstatic RTと共存できます（この機能の前に既存のstatic RTに追加してAuto RTを設定できます）。

自動RDは、BGPルータIDと内部生成された一意の番号で構成されます。たとえば、BGPルータIDが192.168.1.1の場合、自動RDは「192.168.1.1:1」のようになります。

- 自動RTは、BGP AS番号と設定されているvnidで構成されます。たとえば、BGP AS番号が65000で、vniが123に設定されている場合、自動RTは「65000:123」になります。
- これはインポートとエクスポート、通常のルートターゲットとステッチルータターゲットの両方に使用されます。
- BGP ASが4バイトの場合は、代わりにAS_TRANSが使用されます。これは23456です。

設定を簡素化する機能は、導入が可能であるためには非常に望ましく（不要な場合）、BGP EVPNファブリックではすでに広く採用されています。この機能は、多くのVRFが特定のリーフで設定されるスパインリーフトポロジでの広範で複雑な設定の書き込みとメンテナンスを回避できるため、EVPNに適しています。

注：この機能では、新しいCLIが導入されています。

用語

VRF	仮想ルーティング転送	他のVRFおよびグローバルIPv4/IPv6ルーティングドメインから分離されたレイヤ3ルーティングドメインを定義する
AF	アドレスファミリー	BGPが処理するプレフィックスとルーティング情報のタイプを定義します。
AS	自律システム	ネットワークまたはネットワークの集合に属し、単一のエンティティまたは組織によってすべて管理、制御、および監視される一連のインターネットルーティング可能なIPプレフィックス

RD	ルート識別子	BGPが異なるVRFで1つのプレフィックスを別のプレフィックスと区別できるようにする
RTの後に引き継ぐ	ルートターゲット	ルートターゲットは、ルーティング更新を制限するために使用されます。デバイスでインポートできるプレフィックスを決定します
EVPN	イーサネット仮想プライベートネットワーク	BGPによるレイヤ2 MACおよびレイヤ3 IP情報の転送を可能にする拡張はEVPNであり、は、VXLANオーバーレイネットワークに関連する到達可能性情報を配布するプロトコルとして、マルチプロトコルボーダーゲートウェイプロトコル(MP-BGP)を使用します。
VXLAN	仮想拡張LAN (ローカルエリアネットワーク)	VXLANは、VLANとSTPに固有の制限を克服するように設計されています。VLANと同じイーサネットレイヤ2ネットワークサービスを提供するIETF標準[RFC 7348]として提案されていますが、柔軟性に優れています。機能的には、レイヤ3アンダーレイネットワーク上で仮想オーバーレイとして動作するMAC-in-UDPカプセル化プロトコルです。

設定

グローバル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 required
```

```
exit-address-family
```

```
!
```

VRFごとのrd-auto設定

```
<#root>
```

```
Leaf-01#
```

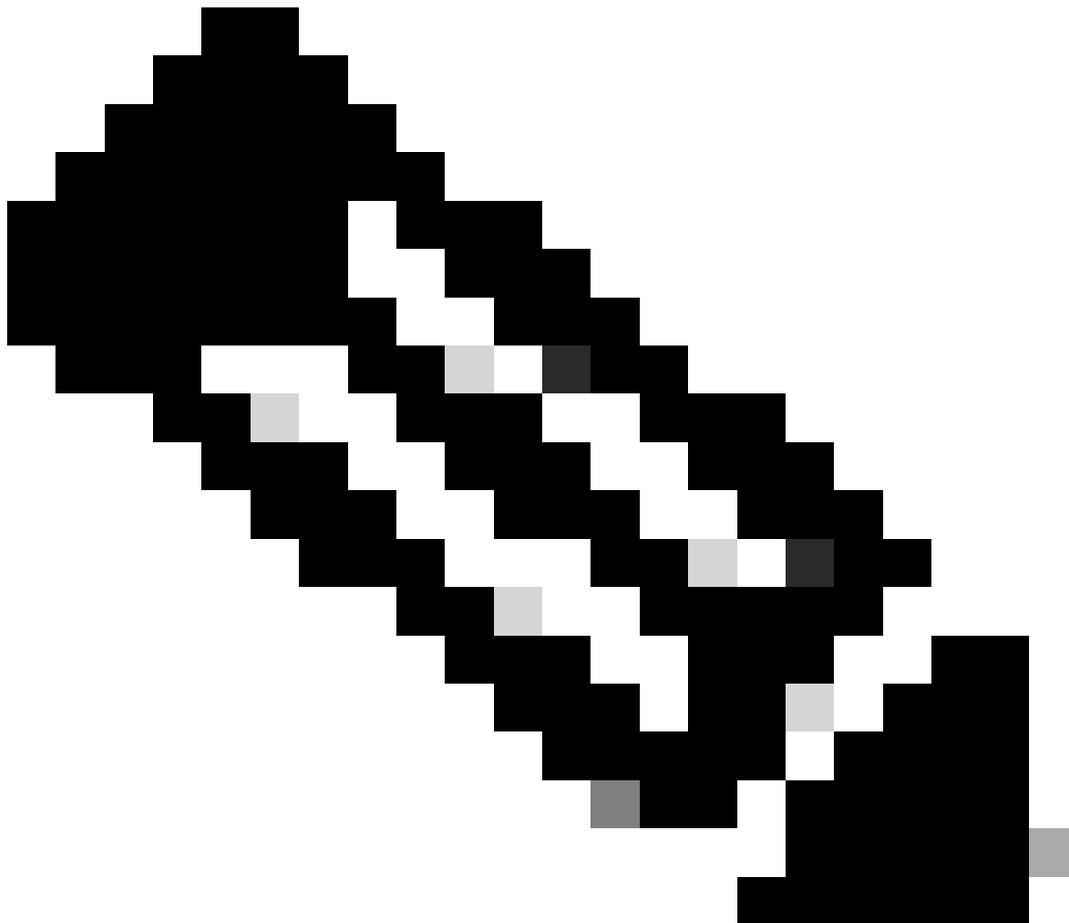
```
sh run | section vrf definition green
```

```
vrf definition green  
rd-auto  
vrid 35 evpn-instance  
!  
address-family ipv4  
exit-address-family  
!  
address-family ipv6  
exit-address-family
```

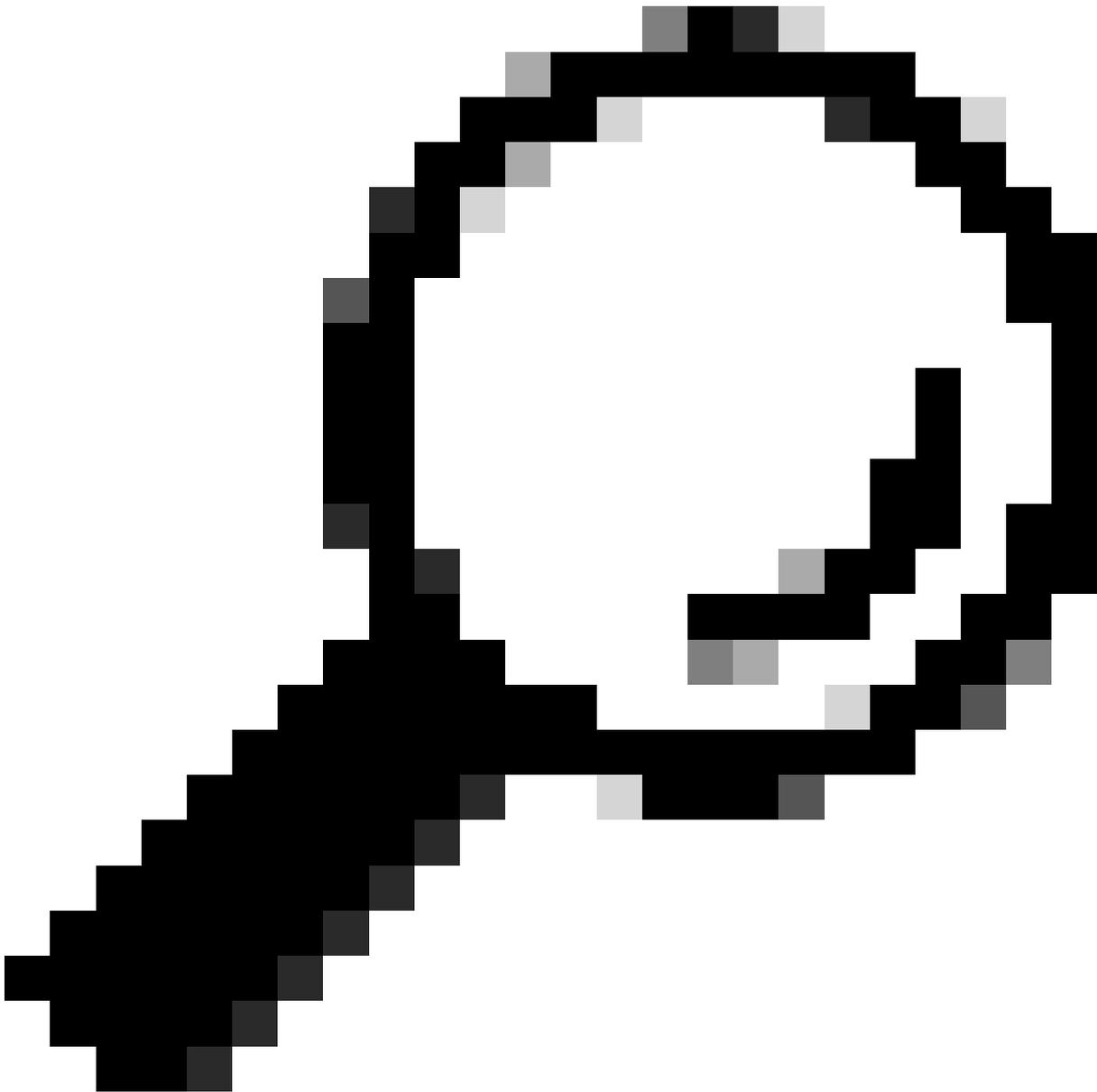
```
<-- Enable Auto RD for this VRF green
```

```
<-- Enable Auto RT
```

```
<-- address-family needs to be specified
```



注：異なるVRFに対してスタティックRDと自動RDを設定することは可能ですが、自動RDが最初に割り当てられた場合は、スタティックRDと自動RDの間に実際のRDを設定することはできません。



ヒント：現在、スタティックRDを削除すると、VRFで設定されているルートターゲットの設定だけでなく、BGP IPv4およびIPv6 VRFアドレスファミリ（およびその下の関連設定）の設定も削除されます。したがって、自動RDを削除しても同様の動作になります。絶対に必要でない限り、RDの削除をトリガーしないことをお勧めします。RDの変更（つまり、既存のRDを静的または自動で削除し、新しいRDを静的または自動で追加する操作は、コストがかかり、コマンドの処理に遅延が発生する可能性があります）

```

<#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アドレスファミリIPv4 VrfおよびIpv6 Vrf

(この設定例は既存の機能の要約です)

```

<#root>

Leaf-01#

show run | sec r bgp

router bgp 65000

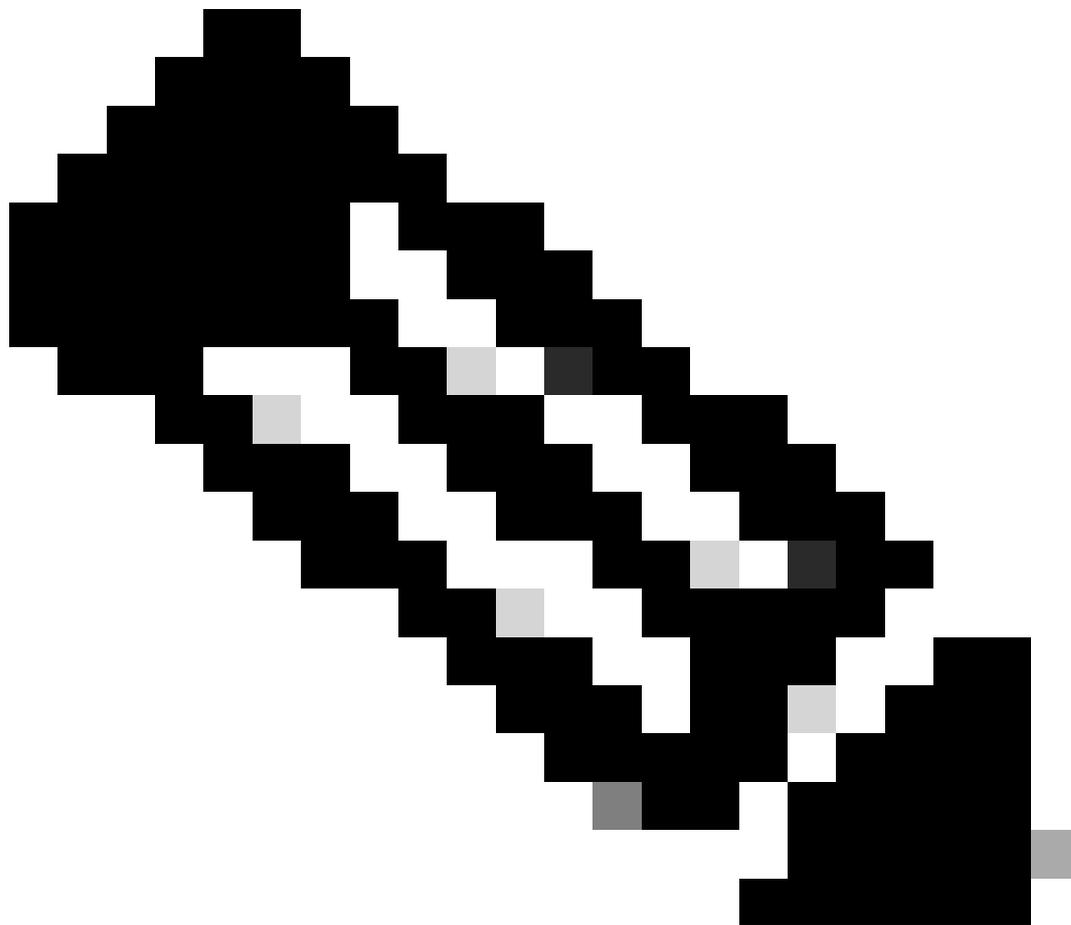
<-- Required for Auto RT

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：他のスパインルートリフレクタの設定は同じであるため、このセクションでは繰り返しません

注：他のEVPNリーフでは、スタティックRDまたはRT設定を使用できます。RTが一致する限り、EVPNプレフィックスは相互にインポート/エクスポートできます。

確認

リーフ

自動RDを有効にするためにリーフを確認します

```
<#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
```

トラブルシューティング

デバッグ

VRF auto RD auto RTに問題がある場合は、デバッグを使用して問題の詳細を確認できます

関連デバッグを有効にする

```
<#root>
```

```
Leaf-01#
```

```
debug ip bgp autordrt
```

```
Leaf-01#
```

```
debug vrf create
```

```
Leaf-01#
```

```
debug vrf delete
```

Display デバッグ情報

```
<#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
```

各設定手順で生成されるデバッグを確認します

```
<#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とNexusの相互運用性

問題

デフォルトでは、Nexusはvniベースのルートターゲット(ASN:VNI)を割り当て、Catalystはeviベースのルートターゲット(ASN:EVI)を割り当てます。

ルートターゲットが一致しない場合は、次のような症状が見られます。

- L2VPN EVPNのBGP接続が確立され、タイプ3ルートがBGPテーブルに表示されます
- NVEピアリングが確立されない
- トンネル隣接関係が不完全なままである

修復方法

この相互運用性の問題を修正するオプションがいくつかあります

1. 一方の側で手動ルートターゲットを設定し、一致させる
2. 「route-target auto vni」を使用してvniベースのルートターゲットを割り当てるようにC9500を設定します。

l2vpn evpnセクションでこれらのcli (オプション2用) を適用します

```
<#root>
```

```
address-family l2vpn evpn
```

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

関連情報

- [BGP EVPN VXLANコンフィギュレーションガイド、Cisco IOS XE Dublin 17.11.x \(Catalyst 9500スイッチ \)](#)
- [テクニカル サポートとドキュメント - Cisco Systems](#)

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。