



## VRF-Lite の設定

---

- [VRF-Lite について \(1 ページ\)](#)
- [VRF-Lite の設定に関するガイドライン \(3 ページ\)](#)
- [VRF-Lite の設定方法 \(4 ページ\)](#)
- [VRF-Lite に関する追加情報 \(26 ページ\)](#)
- [VRF-Lite 設定の確認 \(26 ページ\)](#)
- [VRF-Lite の設定例 \(27 ページ\)](#)
- [VRF-Lite に関するその他の参考資料 \(31 ページ\)](#)
- [マルチキャスト VRF-Lite の機能履歴と情報 \(31 ページ\)](#)

## VRF-Lite について

VRF-Lite の機能によって、サービスプロバイダーは、VPN 間で重複した IP アドレスを使用できる複数の VPN をサポートできます。VRF-Lite は入力インターフェイスを使用して異なる VPN のルートを区別し、各 VRF に 1 つまたは複数のレイヤ 3 インターフェイスを対応付けて仮想パケット転送テーブルを形成します。VRF のインターフェイスは、イーサネットポートなどの物理インターフェイス、または VLAN SVI などの論理インターフェイスにすることができますが、レイヤ 3 インターフェイスは、一度に複数の VRF に属することはできません。



---

(注) VRF-Lite インターフェイスは、レイヤ 3 インターフェイスである必要があります。

---

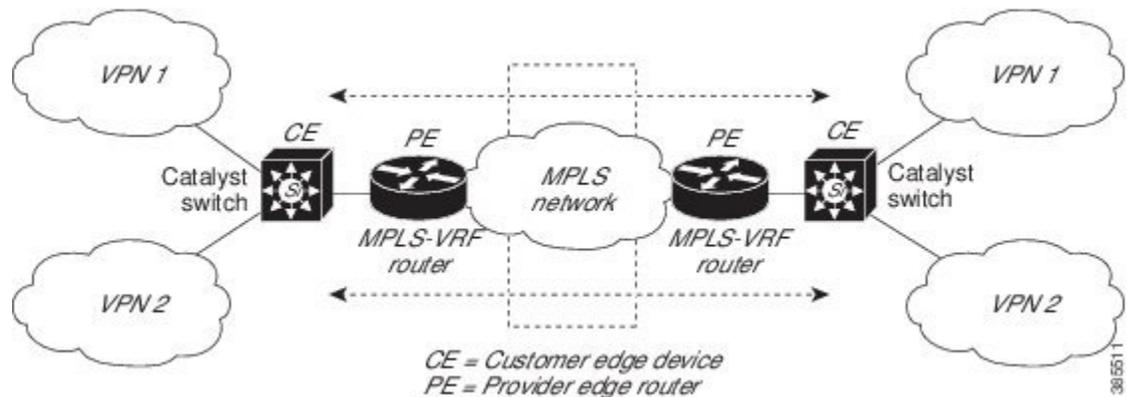
VRF-Lite には次のデバイスが含まれます。

- カスタマーエッジ (CE) デバイスにおいて、カスタマーは、1 つまたは複数のプロバイダーエッジ (PE) ルータへのデータリンクを介してサービスプロバイダーネットワークにアクセスできます。CE デバイスは、サイトのローカルルートをプロバイダーエッジルータにアドバタイズし、そこからリモート VPN ルートを学習します。Cisco Catalyst スイッチは、CE にすることができます。
- プロバイダールータ (またはコアルータ) とは、サービスプロバイダーネットワーク内にあり、CE デバイスに接続していないすべてのルータです。

VRF-lite を使用すると、複数のカスタマーが 1 つの CE を共有できます。また、1 つの物理リンクのみが CE と PE 間に使用されます。共有 CE は、お客様ごとに別々の VRF テーブルを維持し、独自のルーティングテーブルに基づいて、カスタマーごとにパケットをスイッチングまたはルーティングします。VRF-lite は限定された PE の機能を CE デバイスに拡張して、個別の VRF テーブルを保守する機能を付与し、VPN のプライバシーおよびセキュリティをブランチオフィスまで拡張します。

次の図に、各 Cisco Catalyst スイッチが複数の仮想 CE として機能する設定を示します。VRF-Lite はレイヤ 3 機能であるため、VRF の各インターフェイスはレイヤ 3 インターフェイスである必要があります。

図 1: 複数の仮想 CE として機能する Cisco Catalyst スイッチ



次の図に、VRF-Lite の CE 対応ネットワークでのパケット転送プロセスを示します。

- CE が VPN からパケットを受信すると、CE は入力インターフェイスに基づいたルーティングテーブルを検索します。ルートが見つかったら、CE はパケットを PE に転送します。
- 入力 PE は、CE からパケットを受信すると、VRF 検索を実行します。ルートが見つかったら、ルータは対応する MPLS ラベルをパケットに追加し、MPLS ネットワークに送信します。
- 出力 PE は、ネットワークからパケットを受信すると、ラベルを除去してそのラベルを使用し、正しい VPN ルーティングテーブルを識別します。次に、出力 PE が通常のルート検索を行います。ルートが見つかったら、パケットを正しい隣接デバイスに転送します。
- CE が出力 PE からパケットを受信すると、CE は入力インターフェイスを使用して正しい VPN ルーティングテーブルを検索します。ルートが見つかったら、CE はパケットを VPN 内に転送します。

VRF を設定するには、VRF テーブルを作成し、VRF に対応付けられたレイヤ 3 インターフェイスを指定します。次に、VPN および CE と PE 間でルーティングプロトコルを設定します。プロバイダーのバックボーンで VPN ルーティング情報を配信する場合は、BGP が優先ルーティングプロトコルです。VRF-Lite ネットワークには、次の 3 つの主要なコンポーネントがあります。

- VPN ルート ターゲット コミュニティ：VPN コミュニティの他のすべてのメンバをリストします。VPN コミュニティメンバーごとに VPN ルート ターゲットを設定する必要があります。
- VPN コミュニティ PE ルータのマルチプロトコル BGP ピアリング：VPN コミュニティのすべてのメンバーに VRF 到達可能性情報を伝播します。VPN コミュニティのすべての PE ルータで BGP ピアリングを設定する必要があります。
- VPN 転送：VPN サービスプロバイダー ネットワークのすべての VPN コミュニティ メンバ間のすべてのトラフィックを転送します。

## VRF-Lite の設定に関するガイドライン

### IPv4 と IPv6

- VRF-Lite が設定されたスイッチは複数のカスタマーで共有され、すべてのカスタマーが独自のルーティング テーブルを持ちます。
- カスタマーは別々の VRF テーブルを使用するので、同じ IP アドレスを再利用できます。別々の VPN では IP アドレスの重複が許可されます。
- VRF-Lite では、複数のカスタマーが PE と CE の間で同一の物理リンクを共有できます。複数の VLAN を持つトランク ポートでは、パケットがお客様間で分離されます。すべてのカスタマーが独自の VLAN を持ちます。
- PE ルータでは、VRF-Lite の使用と複数の CE の使用には違いがありません。[VRF-Lite について \(1 ページ\)](#) では、複数の仮想レイヤ 3 インターフェイスが VRF-Lite デバイスに接続されています。
- Cisco Catalyst スイッチでは、物理ポートか VLAN SVI、またはその両方の組み合わせを使用して、VRF を設定できます。アクセス ポートまたはトランク ポート経由で SVI を接続できます。
- カスタマーは、別のカスタマーと重複しないかぎり、複数の VLAN を使用できます。カスタマーの VLAN は、スイッチに保存されている適切なルーティング テーブルの識別に使用される特定のルーティング テーブル ID にマッピングされます。
- レイヤ 3 TCAM リソースは、すべての VRF 間で共有されます。各 VRF が十分な CAM 領域を持つようにするには、**maximum routes** コマンドを使用します。
- VRF を使用した Cisco Catalyst スイッチは、1つのグローバル ネットワークと複数の VRF をサポートできます。サポートされるルートの総数は、TCAM のサイズに制限されます。
- 1つの VRF を IPv4 と IPv6 の両方に設定できます。
- 着信パケットの宛先アドレスが VRF テーブルにない場合、そのパケットはドロップされます。また、VRF ルートに TCAM 領域が十分でない場合、その VRF のハードウェアス

イッチングは無効になり、対応するデータパケットがソフトウェアに送信されて処理されます。

### IPv4 固有

- CE と PE 間のほとんどのルーティングプロトコル（BGP、OSPF、EIGRP、RIP、およびスタティックルーティング）を使用できます。ただし、次の理由から External BGP（EBGP）を使用することを推奨します。
  - BGP では、複数の CE とのやり取りに複数のアルゴリズムを必要としません。
  - BGP は、さまざまな管理者によって稼働するシステム間でルーティング情報を渡すように設計されています。
  - BGP は、ルートの属性の CE への引き渡しを単純化します。
- Cisco Catalyst スイッチでは、PIM-SM プロトコルと PIM-SSM プロトコルがサポートされます。
- `router ospf` の `capability vrf-lite` サブコマンドは、PE と CE 間のルーティングプロトコルとして OSPF が設定されている場合に使用する必要があります。

### IPv6 固有

- VRF 認識 OSPFv3、BGPv6、EIGRPv6、および IPv6 スタティックルーティングがサポートされます。
- VRF 認識 IPv6 ルートアプリケーションには、ping、telnet、ssh、tftp、ftp、およびトレースルートが含まれています（このリストには管理インターフェイスは含まれていません。これは、その下に IPv4 も IPv6 も設定できますが、別々に処理されます）。

## VRF-Lite の設定方法

ここでは、VRF-Lite の設定について説明します。

### IPv4 用の VRF-Lite の設定

ここでは、IPv4 用の VRF-Lite の設定について説明します。

#### VRF 認識サービスの設定

IP サービスは、グローバルなインターフェイス上と、グローバルなルーティングインスタンス内で設定できます。IP サービスは複数のルーティングインスタンス上で稼働するように拡張されます。これが、VRF 認識です。システム内の任意の設定済み VRF であればいずれも、VRF 認識サービス用に指定できます。

VRF 認識サービスは、プラットフォームから独立したモジュールに実装されています。VRF は、Cisco IOS 内の複数のルーティングインスタンスを提供します。各プラットフォームには、サポートする VRF 数に関して独自の制限があります。

VRF 認識サービスには、次の特性があります。

- ユーザーは、ユーザー指定の VRF 内のホストに ping を実行できます。
- ARP エントリは、個別の VRF で学習されます。ユーザーは、特定の VRF の ARP エントリを表示できます。

## ARP のユーザインターフェイスの設定

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>show ip arp vrf vrf-name</b> 例： Device# show ip arp vrf vrf-name	指定された VRF で、ARP テーブル（スタティック エントリおよびダイナミック エントリ）を表示します。
ステップ 2	<b>arp vrf vrf-name ip-address mac-address ARPA</b> 例： Device(config)# arp vrf vrf-name ip-address mac-address ARPA	指定された VRF でスタティック ARP エントリを作成します。

## TACACS+ サーバ用の Per-VRF の設定

TACACS+ サーバ機能の per-VRF は TACACS+ サーバの per- 仮想単位ルート転送（per-VRF）の認証、認可、アカウントिंग（AAA）を設定することができます。

VRF ルーティング テーブル（ステップ 3 および 4 で示すように）を作成し、インターフェイスを設定する（ステップ 6、7、および 8）ことができます。TACACS+ サーバの per-VRF 単位の実際の設定は、ステップ 10～13 で行われます。

### 始める前に

TACACS+ サーバの per-VRF を設定する前に、AAA およびサーバグループを設定しておく必要があります。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>enable</b> 例： Device> enable	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	<b>configure terminal</b> 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	<b>vrf definition vrf-name</b> 例： Device(config)# vrf definition vrf-name	VRF テーブルを設定し、VRF コンフィギュレーション モードを開始します。
ステップ 4	<b>rd route-distinguisher</b> 例： Device(config-vrf)# rd route-distinguisher	VRF インスタンスに対するルーティングおよびフローディング テーブルを作成します。
ステップ 5	<b>exit</b> 例： Device(config-vrf)# exit	VRF コンフィギュレーション モードを終了します。
ステップ 6	<b>interface interface-name</b> 例： Device(config)# interface interface-name	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 7	<b>vrf forwarding vrf-name</b> 例： Device(config-if)# vrf forwarding vrf-name	インターフェイスに VRF を設定します。
ステップ 8	<b>ip address ip-address mask [secondary]</b> 例： Device(config-if)# ip address ip-address mask [secondary]	インターフェイスに対するプライマリ IP アドレスまたはセカンダリ IP アドレスを設定します。
ステップ 9	<b>exit</b> 例： Device(config-vrf)# exit	インターフェイス コンフィギュレーション モードを終了します。
ステップ 10	<b>aaa group server tacacs+ group-name</b> 例： Device(config)# aaa group server tacacs+ tacacs1	異なる TACACS+ サーバホストを別々のリストと方式にグループ化し、server-group コンフィギュレーション モードを開始します。
ステップ 11	<b>server-private {ip-address   name} [nat] [single-connection] [port port-number] [timeout seconds] [key [0   7] string]</b> 例： Device(config-sg-tacacs+)# server-private 10.1.1.1 port 19 key cisco	グループ サーバに対するプライベート TACACS+ サーバの IP アドレスを設定します。

	コマンドまたはアクション	目的
ステップ 12	<b>vrf forwarding</b> <i>vrf-name</i> 例： Device(config-sg-tacacs+)# vrf forwarding vrf-name	AAA TACACS+ サーバグループの VRF リファレンスを設定します。
ステップ 13	<b>ip tacacs source-interface</b> <i>subinterface-name</i> 例： Device(config-sg-tacacs+)# ip tacacs source-interface subinterface-name	すべての発信 TACACS+ パケットに対して、指定されたインターフェイスの IP アドレスを使用します。
ステップ 14	<b>exit</b> 例： Device(config-sg-tacacs)# exit	server-group コンフィギュレーションモードを終了します。

### 例

次の例で、per-VRF TACACS+ の設定に必要なすべての手順をリストします。

```
Device> enable
Device# configure terminal
Device(config)# vrf definition cisco
Device(config-vrf)# rd 100:1
Device(config-vrf)# exit
Device(config)# interface Loopback0
Device(config-if)# vrf forwarding cisco
Device(config-if)# ip address 10.0.0.2 255.0.0.0
Device(config-if)# exit
Device(config-sg-tacacs+)# vrf forwarding cisco
Device(config-sg-tacacs+)# ip tacacs source-interface Loopback0
Device(config-sg-tacacs)# exit
```

## マルチキャスト VRF の設定

### 手順の概要

1. **configure terminal**
2. **ip routing**
3. **vrf definition** *vrf-name*
4. **ip multicast-routing vrf** *vrf-name*
5. **rd** *route-distinguisher*
6. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
7. **import map** ルート マップ
8. **interface** *interface-id*
9. **vrf forwarding** *vrf-name*
10. **ip address** *ip-address*/*mask*
11. **ip pim sparse-mode**
12. **end**

13. **show vrf definition** [brief | detail | interfaces] [vrf-name]
14. **copy running-config startup-config**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Device# configure terminal	グローバル設定モードを開始します。
ステップ 2	<b>ip routing</b> 例： Device(config)# ip routing	IP ルーティングを有効にします。
ステップ 3	<b>vrf definition vrf-name</b> 例： Device(config)# vrf definition vrf-name	VRF テーブルを設定し、VRF コンフィギュレーションモードを開始します。
ステップ 4	<b>ip multicast-routing vrf vrf-name</b> 例： Device(config-vrf)# ip multicast-routing vrf vrf-name	(任意) VRF テーブルでグローバルマルチキャストルーティングをイネーブルにします。
ステップ 5	<b>rd route-distinguisher</b> 例： Device(config-vrf)# rd route-distinguisher	ルート識別子を指定して VRF テーブルを作成します。自律システム (AS) 番号および任意の数 (xxx:y) または IP アドレスおよび任意の数 (A.B.C.D:y) のどちらかを入力します。
ステップ 6	<b>route-target {export   import   both} route-target-ext-community</b> 例： Device(config-vrf)# route-target {export   import   both} route-target-ext-community	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポートルートターゲットコミュニティのリストを作成します。AS システム番号と任意の番号 (xxx:y) または IP アドレスと任意の番号 (A.B.C.D:y) を入力します。  ルートターゲット ext コミュニティ値は、ステップ 4 で入力した route-distinguisher 値と同じです。
ステップ 7	<b>import map</b> ルートマップ 例： Device(config-vrf)# import map route-map	(任意) VRF にルートマップを対応付けます。
ステップ 8	<b>interface interface-id</b> 例： Device(config)# interface interface-id	インターフェイス コンフィギュレーションモードを開始して、VRF に対応付けるレイヤ 3 インターフェイスを指定します。有効なインターフェイスは、ルーテッドポートまたは SVI です。

	コマンドまたはアクション	目的
ステップ 9	<b>vrf forwarding vrf-name</b> 例： Device(config-if)# vrf forwarding vrf-name	VRF をレイヤ3 インターフェイスに対応付けます。
ステップ 10	<b>ip address ip-addressmask</b> 例： Device(config-if)# ip address ip-address mask	レイヤ3 インターフェイスの IP アドレスを設定します。
ステップ 11	<b>ip pim sparse-mode</b> 例： Device(config-if)# ip pim sparse-mode	VRF に関連付けられているレイヤ3 インターフェイス上で、PIM をイネーブルにします。
ステップ 12	<b>end</b> 例： Device(config-if)# end	特権 EXEC モードに戻ります。
ステップ 13	<b>show vrf definition [brief   detail   interfaces] [vrf-name]</b> 例： Device# show vrf definition brief	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 14	<b>copy running-config startup-config</b> 例： Device# copy running-config startup-config	(任意) コンフィギュレーション ファイルに設定を保存します。

### 例

次に、VRF テーブル内にマルチキャストを設定する例を示します。

```
Device(config)# ip routing
Device(config)# vrf definition multiVrfA
Device(config-vrf)# ip multicast-routing vrf multiVrfA
Device(config-vrf)# interface GigabitEthernet3/1/0
Device(config-if)# vrf forwarding multiVrfA
Device(config-if)# ip address 172.21.200.203 255.255.255.0
Device(config-if)# ip pim sparse-mode
```

## VPN ルーティング セッションの設定

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例：	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	Device# configure terminal	
ステップ 2	<b>router ospf process-id vrf vrf-name</b> 例： Device(config)# router ospf process-id vrf vrf-name	OSPF ルーティングを有効にし、VPN 転送テーブルを指定して、ルータ コンフィギュレーション モードを開始します。
ステップ 3	<b>log-adjacency-changes</b> 例： Device(config-router)# log-adjacency-changes	(任意) 隣接状態 (デフォルト) の変更を記録します。
ステップ 4	<b>redistribute bgp autonomous-system-number subnets</b> 例： Device(config-router)# redistribute bgp autonomous-system-number subnets	BGP ネットワークから OSPF ネットワークに情報を再配布するようにスイッチを設定します。
ステップ 5	<b>network network-number area area-id</b> 例： Device(config-router)# network network-number area area-id	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 6	<b>end</b> 例： Device(config-router)# end	特権 EXEC モードに戻ります。
ステップ 7	<b>show ip ospf process-id</b> 例： Device# show ip ospf process-id	OSPF ネットワークの設定を確認します。
ステップ 8	<b>copy running-config startup-config</b> 例： Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。  VPN 転送テーブルと OSPF ルーティングプロセスの関連付けを解除するには、 <b>no router ospf process-id vrf vrf-name</b> グローバル コンフィギュレーション コマンドを使用します。

## 例

```

Device(config)# vrf definition VRF-RED
Device(config-vrf)# rd 1:1
Device(config-vrf)# exit
Device(config)# router eigrp virtual-name
Device(config-router)# address-family ipv4 vrf VRF-RED autonomous-system 1
Device(config-router-af)# network 10.0.0.0 0.0.0.255
Device(config-router-af)# topology base
Device(config-router-topology)# default-metric 10000 100 255 1 1500

```

```
Device(config-router-topology)# exit-af-topology
Device(config-router-af)# exit-address-family
```

## BGP PE/CE ルーティング セッションの設定

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>router bgp <i>autonomous-system-number</i></b> 例： Device(config)# router bgp <i>autonomous-system-number</i>	その他の BGP ルータに渡された AS 番号で BGP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。
ステップ 3	<b>network <i>network-number</i> mask <i>network-mask</i></b> 例： Device(config-router)# network <i>network-number</i> mask <i>network-mask</i>	BGP を使用してアナウンスするネットワークおよびマスクを指定します。
ステップ 4	<b>redistribute ospf <i>process-id</i> match <i>internal</i></b> 例： Device(config-router)# redistribute ospf <i>process-id</i> match <i>internal</i>	OSPF 内部ルートを再配布するようにスイッチを設定します。
ステップ 5	<b>network <i>network-number</i> area <i>area-id</i></b> 例： Device(config-router)# network <i>network-number</i> area <i>area-id</i>	OSPF が動作するネットワークアドレスとマスク、およびそのネットワークアドレスのエリア ID を定義します。
ステップ 6	<b>address-family ipv4 vrf <i>vrf-name</i></b> 例： Device(config-router-af)# address-family ipv4 vrf <i>vrf-name</i>	PE から CE のルーティングセッションの BGP パラメータを定義し、VRF アドレスファミリ モードを開始します。
ステップ 7	<b>neighbor <i>address</i> remote-as <i>as-number</i></b> 例： Device(config-router-af)# neighbor <i>address</i> remote-as <i>as-number</i>	PE と CE ルータの間の BGP セッションを定義します。
ステップ 8	<b>neighbor <i>address</i> activate</b> 例： Device(config-router-af)# neighbor <i>address</i> activate	IPv4 アドレスファミリのアドバタイズメントをアクティブ化します。

	コマンドまたはアクション	目的
ステップ 9	<b>end</b> 例： Device(config-router-af)# end	特権 EXEC モードに戻ります。
ステップ 10	<b>show ip bgp [ipv4] [neighbors]</b> 例： Device# show ip bgp [ipv4] [neighbors]	BGP 設定を確認します。  BGP ルーティングプロセスを削除するには、 <b>no router bgp autonomous-system-number</b> グローバル コンフィギュレーションコマンドを使用します。ルーティング特性を削除するには、コマンドにキーワードを指定してこのコマンドを使用します。

## IPv4 VRF の設定

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>ip routing</b> 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	<b>vrf definition vrf-name</b> 例： Device(config)# vrf definition vrf-name	VRF 名を指定し、VRF コンフィギュレーション モードを開始します。
ステップ 4	<b>rd route-distinguisher</b> 例： Device(config-vrf)# rd route-distinguisher	ルート識別子を指定して VRF テーブルを作成します。自律システム番号と任意の数値 (xxx:y)、または IP アドレスと任意の数値 (A.B.C.D:y) のいずれかを入力します。
ステップ 5	<b>route-target {export   import   both} route-target-ext-community</b> 例： Device(config-vrf)# route-target {export   import   both} route-target-ext-community	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポートルートターゲットコミュニティのリストを作成します。AS システム番号と任意の番号 (xxx:y) または IP アドレスと任意の番号 (A.B.C.D:y) を入力します。  (注) このコマンドは、BGP が動作している場合にのみ有効です。

	コマンドまたはアクション	目的
ステップ 6	<b>import map</b> ルート マップ 例： Device(config-vrf)# import map route-map	(任意) VRF にルート マップを対応付けます。
ステップ 7	<b>interface interface-id</b> 例： Device(config-vrf)# interface interface-id	インターフェイス コンフィギュレーション モードを開始して、VRF に対応付けるレイヤ 3 インターフェイスを指定します。インターフェイスにはルーテッドポートまたは SVI を設定できます。
ステップ 8	<b>vrf forwarding vrf-name</b> 例： Device(config-if)# vrf forwarding vrf-name	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 9	<b>end</b> 例： Device(config-if)# end	特権 EXEC モードに戻ります。
ステップ 10	<b>show vrf definition [brief   detail   interfaces] [vrf-name]</b> 例： Device# show vrf definition [brief   detail   interfaces] [vrf-name]	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 11	<b>copy running-config startup-config</b> 例： Device# copy running-config startup-config	(任意) コンフィギュレーション ファイルに設定を保存します。  VRF とそのすべてのインターフェイスを削除するには、 <b>no vrf definition vrf-name</b> グローバル コンフィギュレーション コマンドを使用します。VRF からインターフェイスを削除するには、 <b>no vrf forwarding</b> インターフェイス コンフィギュレーション コマンドを使用します。

## IPv6 用の VRF-Lite の設定

ここでは、IPv6 用の VRF-Lite の設定について説明します。

### VRF 認識サービスの設定

IPv6 サービスは、グローバルなインターフェイス上と、グローバルなルーティング インスタンス内で設定できます。IPv6 サービスは複数のルーティング インスタンス上で稼働するように拡張されます。これが、VRF 認識です。システム内の任意の設定済み VRF であればいずれも、VRF 認識サービス用に指定できます。

VRF 認識サービスは、プラットフォームから独立したモジュールに実装されています。VRF は、Cisco IOS 内の複数のルーティングインスタンスを提供します。各プラットフォームには、サポートする VRF 数に関して独自の制限があります。

VRF 認識サービスには、次の特性があります。

- ユーザーは、ユーザー指定の VRF 内のホストに ping を実行できます。
- ネイバー探索エントリは、個別の VRF で学習されます。ユーザは、特定の VRF のネイバー探索 (ND) エントリを表示できます。

次のサービスは VRF 認識です。

- Ping
- ユニキャスト RPF (uRPF)
- traceroute
- FTP および TFTP
- [Telnet および SSH (Telnet and SSH) ]
- NTP

## PING のユーザ インターフェイスの設定

VRF 認識 ping を設定するには、次の作業を実行します。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<p><b>ping vrf vrf-name ipv6-host</b></p> <p>例 :</p> <pre>Device# ping vrf vrf-name ipv6-host</pre>	指定された VRF で、IPv6 ホストまたはアドレスに対して ping を実行します。

## uRPF のユーザ インターフェイスの設定

VRF に割り当てられているインターフェイス上で、uRPF を設定できます。送信元の検索が VRF テーブルで実行されます。

### 手順の概要

1. **configure terminal**
2. **interface interface-id**
3. **no switchport**
4. **vrf forwarding vrf-name**
5. **ipv6 address ip-addresssubnet-mask**
6. **ipv6 verify unicast source reachable-via rx allow-default**
7. **end**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>interface interface-id</b> 例： Device(config)# interface interface-id	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 3	<b>no switchport</b> 例： Device(config-if)# no switchport	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 4	<b>vrf forwarding vrf-name</b> 例： Device(config-if)# vrf forwarding vrf-name	インターフェイス上で VRF を設定します。
ステップ 5	<b>ipv6 address ip-address subnet-mask</b> 例： Device(config-if)# ip address ip-address mask	インターフェイスの IPv6 アドレスを入力します。
ステップ 6	<b>ipv6 verify unicast source reachable-via rx allow-default</b> 例： Device(config-if)# ipv6 verify unicast source reachable-via rx allow-default	インターフェイス上で uRPF を有効にします。
ステップ 7	<b>end</b> 例： Device(config-if)# end	特権 EXEC モードに戻ります。

## Traceroute のユーザ インターフェイスの設定

## 手順の概要

## 1. traceroute vrf vrf-name ipv6address

## 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>traceroute vrf vrf-name ipv6address</b> 例： Device# traceroute vrf vrf-name ipv6address	宛先アドレスを取得する VPN VRF の名前を指定します。

## Telnet および SSH のユーザインターフェイスの設定

## 手順の概要

1. **telnet ipv6-address/ vrf vrf-name**
2. **ssh -l username -vrf vrf-name ipv6-host**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>telnet ipv6-address/ vrf vrf-name</b> 例： Device# telnet ipv6-address/vrf vrf-name	指定された VRF で、IPv6 ホストまたはアドレスに Telnet 経由で接続します。
ステップ 2	<b>ssh -l username -vrf vrf-name ipv6-host</b> 例： Device# ssh -l username -vrf vrf-name ipv6-host	指定された VRF で、IPv6 ホストまたはアドレスに SSH 経由で接続します。

## NTP のユーザインターフェイスの設定

## 手順の概要

1. **configure terminal**
2. **ntp server vrf vrf-name ipv6-host**
3. **ntp peer vrf vrf-name ipv6-host**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Device# <b>configure terminal</b>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>ntp server vrf vrf-name ipv6-host</b> 例： Device(config)# <b>ntp server vrf vrf-name ipv6-host</b>	指定された VRF で NTP サーバを設定します。
ステップ 3	<b>ntp peer vrf vrf-name ipv6-host</b> 例： Device(config)# <b>ntp peer vrf vrf-name ipv6-host</b>	指定された VRF で NTP ピアを設定します。

## IPv6 VRF の設定

### 手順の概要

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **rd** *route-distinguisher*
4. **address-family** *ipv4* | *ipv6*
5. **route-target** {*export* | *import* | *both*} *route-target-ext-community*
6. **exit-address-family**
7. **vrf definition** *vrf-name*
8. **ipv6 multicast multitopology**
9. **address-family** *ipv6* **multicast**
10. **end**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Device# <code>configure terminal</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>vrf definition</b> <i>vrf-name</i> 例： Device(config)# <code>vrf definition vrf-name</code>	VRF 名を指定し、VRF コンフィギュレーション モードを開始します。
ステップ 3	<b>rd</b> <i>route-distinguisher</i> 例： Device(config-vrf)# <code>rd route-distinguisher</code>	(任意) ルート識別子を指定して VRF テーブルを作成します。自律システム番号および任意の数 (xxx:y)、または IP アドレスおよび任意の数 (A.B.C.D:y) のいずれかを入力します。
ステップ 4	<b>address-family</b> <i>ipv4</i>   <i>ipv6</i> 例： Device(config-vrf)# <code>address-family ipv4   ipv6</code>	(任意) デフォルトは IPv4 です。IPv6 の必須設定。
ステップ 5	<b>route-target</b> { <i>export</i>   <i>import</i>   <i>both</i> } <i>route-target-ext-community</i> 例： Device(config-vrf)# <code>route-target {export   import   both} route-target-ext-community</code>	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポートルートターゲットコミュニティのリストを作成します。AS システム番号と任意の番号 (xxx:y) または IP アドレスと任意の番号 (A.B.C.D:y) を入力します。  (注) このコマンドは、BGP が動作している場合にのみ有効です。

## 定義済み VRF へのインターフェイスの関連付け

	コマンドまたはアクション	目的
ステップ 6	<b>exit-address-family</b> 例： Device(config-vrf)# exit-address-family	VRF アドレス ファミリ コンフィギュレーション モードを終了し、VRF コンフィギュレーション モードに戻ります。
ステップ 7	<b>vrf definition vrf-name</b> 例： Device(config)# vrf definition vrf-name	VRF コンフィギュレーションモードを開始します。
ステップ 8	<b>ipv6 multicast multitopology</b> 例： Device(config-vrf-af)# ipv6 multicast multitopology	マルチキャスト固有の RPF トポロジを有効にします。
ステップ 9	<b>address-family ipv6 multicast</b> 例： Device(config-vrf)# address-family ipv6 multicast	マルチキャスト IPv6 アドレス ファミリを入力します。
ステップ 10	<b>end</b> 例： Device(config-vrf-af)# end	特権 EXEC モードに戻ります。

## 例

次に、VRF を設定する例を示します。

```
Device(config)# vrf definition red
Device(config-vrf)# rd 100:1
Device(config-vrf)# address family ipv6
Device(config-vrf-af)# route-target both 200:1
Device(config-vrf)# exit-address-family
Device(config-vrf)# vrf definition red
Device(config-vrf)# ipv6 multicast multitopology
Device(config-vrf)# address-family ipv6 multicast
Device(config-vrf-af)# end
```

## 定義済み VRF へのインターフェイスの関連付け

## 手順の概要

1. **interface** *interface-id*
2. **no switchport**
3. **vrf forwarding** *vrf-name*
4. **ipv6 enable**
5. **ipv6 address** *ip-address subnet-mask*
6. **show ipv6 vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*]

## 7. copy running-config startup-config

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>interface interface-id</b> 例： Device(config-vrf)# interface interface-id	インターフェイス コンフィギュレーション モードを開始して、VRF に対応付けるレイヤ 3 インターフェイスを指定します。インターフェイスにはルーテッドポートまたは SVI を設定できます。
ステップ 2	<b>no switchport</b> 例： Device(config-if)# no switchport	コンフィギュレーションモードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 3	<b>vrf forwarding vrf-name</b> 例： Device(config-if)# vrf forwarding vrf-name	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 4	<b>ipv6 enable</b> 例： Device(config-if)# ipv6 enable	インターフェイスで IPv6 を有効にします。
ステップ 5	<b>ipv6 address ip-address subnet-mask</b> 例： Device(config-if)# ipv6 address ip-address subnet-mask	インターフェイスの IPv6 アドレスを入力します。
ステップ 6	<b>show ipv6 vrf [brief   detail   interfaces] [vrf-name]</b> 例： Device# show ipv6 vrf [brief   detail   interfaces] [vrf-name]	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 7	<b>copy running-config startup-config</b> 例： Device# copy running-config startup-config	（任意）コンフィギュレーションファイルに設定を保存します。

### 例

次に、インターフェイスを VRF に関連付ける例を示します。

```
Switch(config-vrf)# interface ethernet0/1
Switch(config-if)# vrf forwarding red
Switch(config-if)# ipv6 enable
Switch(config-if)# ipv6 address 5000::72B/64
```

## ルーティング プロトコル経由での VRF へのルートの入力

ここでは、ルーティングプロトコル経由での VRF へのルートの入力について説明します。

### VRF スタティック ルートの設定

#### 手順の概要

1. **configure terminal**
2. **ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address | interface-type interface-number [ipv6-address]}**

#### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address   interface-type interface-number [ipv6-address]}</b> 例： Device(config)# ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address   interface-type interface-number [ipv6-address]}	VRF に固有のスタティック ルートを設定します。

#### 例

```
Device(config)# ipv6 route vrf v6a 7000::/64 TenGigabitEthernet32 4000::2
```

### OSPFv3 ルータ プロセスの設定

#### 手順の概要

1. **configure terminal**
2. **router ospfv3 process-id**
3. **area area-ID [default-cot | nssa | stub]**
4. **router-id router-id**
5. **address-family ipv6 unicast vrf vrf-name**
6. **redistribute source-protocol [process-id] options**
7. **end**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>router ospfv3 process-id</b> 例： Device(config)# router ospfv3 process-id	IPv6 アドレス ファミリの OSPFv3 ルータ コンフィギュレーション モードを有効にします。
ステップ 3	<b>area area-ID [default-cot   nssa   stub]</b> 例： Device(config-router)# area area-ID [default-cot   nssa   stub]	OSPFv3 エリアを設定します。
ステップ 4	<b>router-id router-id</b> 例： Device(config-router)# router-id router-id	固定ルータ ID を使用します。
ステップ 5	<b>address-family ipv6 unicast vrf vrf-name</b> 例： Device(config-router)# address-family ipv6 unicast vrf vrf-name	vrf vrf-name の OSPFv3 の IPv6 アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 6	<b>redistribute source-protocol [process-id] options</b> 例： Device(config-router)# redistribute source-protocol [process-id] options	あるルーティング ドメインから別のルーティング ドメインへ IPv6 ルートを再配布します。
ステップ 7	<b>end</b> 例： Device(config-router)# end	特権 EXEC モードに戻ります。

## 例

次に、OSPFv3 ルータ プロセスを設定する例を示します。

```
Device(config-router)# router ospfv3 1
Device(config-router)# router-id 1.1.1.1
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# exit-address-family
```

## インターフェイス上での OSPFv3 の有効化

### 手順の概要

1. **configure terminal**
2. **interface** *type-number*
3. **ospfv3** *process-id* **area** *area-id* **ipv6** [**instance** *instance-id*]
4. **end**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>interface</b> <i>type-number</i> 例： Device(config-vrf)# interface <i>type-number</i>	インターフェイスのタイプと番号を指定し、スイッチをインターフェイスコンフィギュレーションモードにします。
ステップ 3	<b>ospfv3</b> <i>process-id</i> <b>area</b> <i>area-id</i> <b>ipv6</b> [ <b>instance</b> <i>instance-id</i> ] 例： Device(config-if)# ospfv3 <i>process-id</i> <b>area</b> <i>area-ID</i> <b>ipv6</b> [ <i>instance instance-id</i> ]	IPv6 AF を設定したインターフェイスで OSPFv3 を有効にします。
ステップ 4	<b>end</b> 例： Device(config-if)# end	特権 EXEC モードに戻ります。

### 例

次に、インターフェイス上で OSPFv3 を有効にする例を示します。

```
Device(config)# interface GigabitEthernet2/1
Device(config-if)# no switchport
Device(config-if)# ipv6 address 4000::2/64
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 ospf 1 area 0
Device(config-if)# end
```

## EIGRPv6 ルーティング プロセスの設定

### 手順の概要

1. **configure terminal**
2. **router eigrp** *virtual-instance-name*
3. **address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number*

4. **topology {base | topology-name tid number**
5. **exit-aftopology**
6. **eigrp router-id ip-address**
7. **end**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>router eigrp virtual-instance-name</b> 例： Device(config)# router eigrp virtual-instance-name	EIGRP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。
ステップ 3	<b>address-family ipv6 vrf vrf-name autonomous-system autonomous-system-number</b> 例： Device(config-router)# address-family ipv6 vrf vrf-name autonomous-system autonomous-system-number	EIGRP IPv6 VRF-Lite を有効にし、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 4	<b>topology {base   topology-name tid number</b> 例： Device(config-router-af)# topology {base   topology-name tid number	指定されたトポロジインスタンスで IP トラフィックをルーティングするよう EIGRP プロセスを設定し、アドレス ファミリ トポロジ コンフィギュレーション モードを開始します。
ステップ 5	<b>exit-aftopology</b> 例： Device(config-router-af-topology)# exit-aftopology	アドレス ファミリ トポロジ コンフィギュレーション モードを終了します。
ステップ 6	<b>eigrp router-id ip-address</b> 例： Device(config-router)# eigrp router-id ip-address	固定ルータ ID の使用を有効にします。
ステップ 7	<b>end</b> 例： Device(config-router)# end	ルータ コンフィギュレーション モードを終了します。

## 例

次に、EIGRP ルーティング プロセスを設定する例を示します。

```
Device(config)# router eigrp test
```

```

Device(config-router)# address-family ipv6 unicast vrf b1 autonomous-system 10
Device(config-router-af)# topology base
Device(config-router-af-topology)# exit-af-topology
Device(config-router)# eigrp router-id 2.3.4.5
Device(config-router)# exit-address-family

```

## EBGPv6 ルーティング プロセスの設定

### 手順の概要

1. **configure terminal**
2. **router bgp as-number**
3. **neighbor peer-group-name peer-group**
4. **neighbor {ip-address | ipv6-address[%] | peer-group-name} remote-as autonomous-system-number [ alternate-as autonomous-system-number ...]**
5. **address-family ipv6 [ vrf vrf-name] [unicast | multicast | vpnv6]**
6. **neighbor ipv6-address peer-group peer-group-name**
7. **neighbor {ip-address | peer-group-name | ipv6-address[%]} route-map map-name {in | out}**
8. **exit**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	<b>router bgp as-number</b> 例： Device(config)# router bgp as-number	指定したルーティング プロセスのルータ コンフィギュレーション モードを開始します。
ステップ 3	<b>neighbor peer-group-name peer-group</b> 例： Device(config-router)# neighbor peer-group-name peer-group	マルチプロトコル BGP ピア グループを作成します。
ステップ 4	<b>neighbor {ip-address   ipv6-address[%]   peer-group-name} remote-as autonomous-system-number [ alternate-as autonomous-system-number ...]</b> 例： Device(config-router)# neighbor {ip-address   ipv6-address[%]   peer-group-name} remote-as autonomous-system-number [alternate-as autonomous-system-number ...]	指定した自律システム内のネイバーの IPv6 アドレスを、ローカル ルータの IPv6 マルチプロトコル BGP ネイバーテーブルに追加します。
ステップ 5	<b>address-family ipv6 [ vrf vrf-name] [unicast   multicast   vpnv6]</b> 例：	IPv6 アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	Device(config-router)# address-family ipv6 [vrf vrf-name] [unicast   multicast   vpnv6]	<ul style="list-style-type: none"> <li>• <b>unicast</b> キーワードは、IPv6 ユニキャストアドレスファミリを指定します。デフォルトでは、<b>address-family ipv6</b> コマンドでユニキャストキーワードが指定されていない場合、スイッチは IPv6 ユニキャストアドレスファミリのコンフィギュレーションモードになります。</li> <li>• <b>multicast</b> キーワードは、IPv6 マルチキャストアドレスプレフィックスを指定します。</li> </ul>
ステップ 6	<b>neighbor ipv6-address peer-group peer-group-name</b> 例 : Device(config-router-af)# neighbor ipv6-address peer-group peer-group-name	BGP ネイバーの IPv6 アドレスをピア グループに割り当てます。
ステップ 7	<b>neighbor {ip-address   peer-group-name   ipv6-address[%]} route-map map-name {in   out}</b> 例 : Device(config-router-af)# neighbor {ip-address   peer-group-name   ipv6-address[%]} route-map map-name {in   out}	着信ルートまたは発信ルートにルートマップを適用します。ルートマップへの変更は、ピアリングがリセットされるまで、またはソフトリセットが実行されるまで、現在のピアでは有効になりません。soft キーワードと in キーワードを指定して <b>clear bgp ipv6</b> コマンドを使用すると、ソフトリセットが実行されます。
ステップ 8	<b>exit</b> 例 : Device(config-router-af)# exit	アドレスファミリ コンフィギュレーション モードを終了し、ルータをルータコンフィギュレーションモードに戻します。

## 例

次に、EBGPv6 を設定する例を示します。

```

Device(config)# router bgp 2
Device(config-router)# bgp router-id 2.2.2.2
Device(config-router)# bgp log-neighbor-changes
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# neighbor 2500::1 remote-as 1
Device(config-router)# neighbor 4000::2 remote-as 3
Device(config-router)# address-family ipv6 vrf b1
Device(config-router-af)# network 2500::/64
Device(config-router-af)# network 4000::/64
Device(config-router-af)# neighbor 2500::1 remote-as 1
Device(config-router-af)# neighbor 2500::1 activate
Device(config-router-af)# neighbor 4000::2 remote-as 3
Device(config-router-af)# neighbor 4000::2 activate
Device(config-router-af)# exit-address-family

```

## VRF-Lite に関する追加情報

ここでは、VRF-Lite に関する追加情報を提供します。

### IPv4 と IPv6 間での VPN の共存

IPv4 を設定するための「以前の」CLI と、IPv6 用の「新しい」CLI 間には下位互換性があります。つまり、設定に両方の CLI を含めることができます。IPv4 CLI は、同じインターフェイス上で、VRF 内で定義されている IP アドレスとともにグローバルルーティングテーブルで定義されている IPv6 アドレスも備える機能を保持しています。

次に例を示します。

```
vrf definition red
 rd 100:1
 address family ipv6
 route-target both 200:1
 exit-address-family
!
vrf definition blue
 rd 200:1
 route-target both 200:1
!
interface Ethernet0/0
 vrf forwarding red
 ip address 50.1.1.2 255.255.255.0
 ipv6 address 4000::72B/64
!
interface Ethernet0/1
 vrf forwarding blue
 ip address 60.1.1.2 255.255.255.0
 ipv6 address 5000::72B/64
```

この例では、Ethernet0/0 用に定義されたすべてのアドレス（v4 と v6）が VRF red を参照します。Ethernet0/1 については、IP アドレスは VRF blue を参照しますが、ipv6 アドレスはグローバル IPv6 アドレス ルーティング テーブルを参照します。

## VRF-Lite 設定の確認

ここでは、VRF-Lite 設定を確認する手順について説明します。

### IPv4 VRF-Lite ステータスの表示

VRF-Lite の設定およびステータスに関する情報を表示するには、次の作業のいずれかを行います。

コマンド	目的
Device# <b>show ip protocols vrf</b> <i>vrf-name</i>	VRF に対応付けられたルーティングプロトコル情報を表示します。

コマンド	目的
Device# <b>show ip route vrf</b> <i>vrf-name</i> [ <b>connected</b> ] [ <i>protocol</i> ] [ <i>as-number</i> ] [ <b>list</b> ] [ <b>mobile</b> ] [ <b>odr</b> ] [ <b>profile</b> ] [ <b>static</b> ] [ <b>summary</b> ] [ <b>supernets-only</b> ]	VRF に対応付けられた IP ルーティング テーブル情報を表示します。
Device# <b>show vrf definition</b> [ <b>brief</b>   <b>detail</b>   <b>interfaces</b> ] [ <i>vrf-name</i> ]	定義された VRF インスタンスに関する情報を表示します。
Device# <b>bidir vrf</b> <i>instance-name a.b.c.d</i>   <b>active</b>   <b>bidirectional</b>   <b>count</b>   <b>interface</b>   <b>proxy</b>   <b>pruned</b>   <b>sparse</b>   <b>ssm</b>   <b>static</b>   <b>summary</b>	定義された VRF インスタンスに関する情報を表示します。

次に、VRF インスタンス内のマルチキャスト ルート テーブル情報を表示する例を示します。

```
Switch# show ip mroute 226.0.0.2
IP Multicast Routing Table
Flags: S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 226.0.0.2), 00:01:17/stopped, RP 1.1.1.1, flags: SJCF
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan100, Forward/Sparse, 00:01:17/00:02:36

(5.0.0.11, 226.0.0.2), 00:01:17/00:01:42, flags: FT
  Incoming interface: Vlan5, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan100, Forward/Sparse, 00:01:17/00:02:36
```

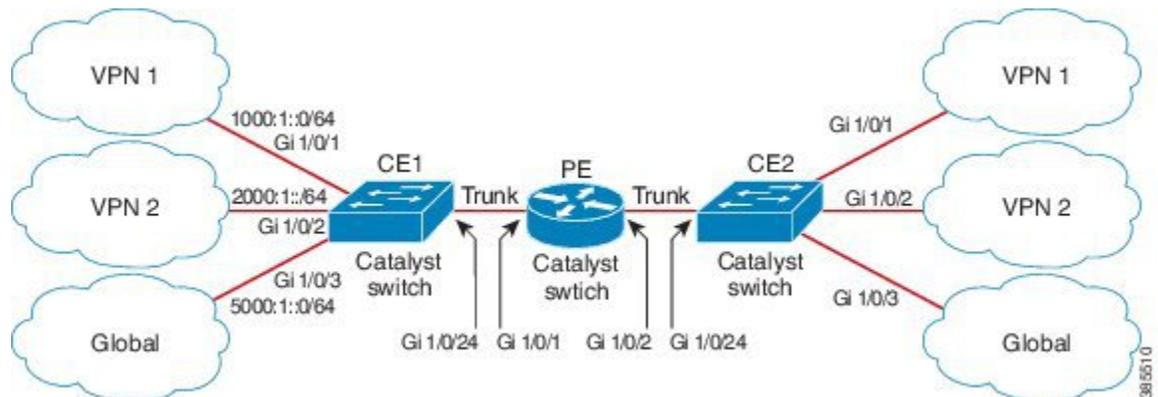
## VRF-Lite の設定例

ここでは、VRF-Lite の設定例を示します。

### IPv6 VRF-Lite の設定例

次に、CE-PE ルーティングに OSPFv3 を使用するトポロジを示します。

図 2: VRF-Lite の設定例



### CE1 スイッチの設定

```

ipv6 unicast-routing
vrf definition v1
 rd 100:1
 !
address-family ipv6
 exit-address-family
!

vrf definition v2
 rd 200:1
 !
address-family ipv6
 exit-address-family
!

interface Vlan100
 vrf forwarding v1
 ipv6 address 1000:1::1/64
 ospfv3 100 ipv6 area 0
!

interface Vlan200
 vrf forwarding v2
 ipv6 address 2000:1::1/64
 ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
 switchport access vlan 100
end

interface GigabitEthernet 1/0/2
 switchport access vlan 200
end

interface GigabitEthernet 1/0/24
 switchport trunk encapsulation dot1q

switchport mode trunk
end

router ospfv3 100
 router-id 10.10.10.10

```

```
!
address-family ipv6 unicast vrf v1
 redistribute connected
 area 0 normal
 exit-address-family
!

router ospfv3 200
 router-id 20.20.20.20
 !
 address-family ipv6 unicast vrf v2
  redistribute connected
  area 0 normal
 exit-address-family
!
```

### PE スイッチの設定

```
ipv6 unicast-routing

vrf definition v1
 rd 100:1
 !
 address-family ipv6
  exit-address-family
 !

vrf definition v2
 rd 200:1
 !
 address-family ipv6
  exit-address-family
 !

interface Vlan600
 vrf forwarding v1
 no ipv6 address
 ipv6 address 1000:1::2/64
 ospfv3 100 ipv6 area 0
 !

interface Vlan700
 vrf forwarding v2
 no ipv6 address
 ipv6 address 2000:1::2/64
 ospfv3 200 ipv6 area 0
 !

interface Vlan800
 vrf forwarding v1
 ipv6 address 3000:1::7/64
 ospfv3 100 ipv6 area 0
 !

interface Vlan900
 vrf forwarding v2
 ipv6 address 4000:1::7/64
 ospfv3 200 ipv6 area 0
 !

interface GigabitEthernet 1/0/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
 exit

interface GigabitEthernet 1/0/2
```

```
switchport trunk encapsulation dot1q

switchport mode trunk
exit

router ospfv3 100
router-id 30.30.30.30
!
address-family ipv6 unicast vrf v1
redistribute connected
area 0 normal
exit-address-family
!
address-family ipv6 unicast vrf v2
redistribute connected
area 0 normal
exit-address-family
!
```

### CE2 スイッチの設定

```
ipv6 unicast-routing

vrf definition v1
rd 100:1
!
address-family ipv6
exit-address-family
!

vrf definition v2
rd 200:1
!
address-family ipv6
exit-address-family
!

interface Vlan100
vrf forwarding v1

ipv6 address 1000:1::3/64
ospfv3 100 ipv6 area 0
!

interface Vlan200
vrf forwarding v2
ipv6 address 2000:1::3/64
ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
switchport access vlan 100
end

interface GigabitEthernet 1/0/2
switchport access vlan 200
end

interface GigabitEthernet 1/0/24
switchport trunk encapsulation dot1q
switchport mode trunk
end

router ospfv3 100
```

```

router-id 40.40.40.40
!
address-family ipv6 unicast vrf v1
 redistribute connected
 area 0 normal
exit-address-family
!

router ospfv3 200
router-id 50.50.50.50
!
address-family ipv6 unicast vrf v2
 redistribute connected

area 0 normal
exit-address-family
!

```

## VRF-Lite に関するその他の参考資料

### 関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	の「IP マルチキャストルーティングのコマンド」の項を参照してください。 <i>Command Reference (Catalyst 9300 Series Switches)</i>

### 標準および RFC

標準/RFC	タイトル
RFC 6763	『DNS-Based Service Discovery』
マルチキャスト DNS インターネット (ドラフト)	<a href="#">マルチキャスト</a>

## マルチキャスト VRF-Lite の機能履歴と情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、[www.cisco.com/go/cfn](http://www.cisco.com/go/cfn) に移動します。Cisco.com のアカウントは必要ありません。

機能名	リリース	機能情報
VRF-Lite を使用した IPv6 マルチキャストのサポート	Cisco IOS XE Everest 16.6.1	IPv6 VRF-Lite によって、サービスプロバイダーは1つのインターフェイスを使用して、重複する IP アドレスを持つ複数の VPN をサポートできます。

## 翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。