『Configuring MPLS Layer 3 VPNs』

この章では、Cisco Nexus 9508 スイッチでマルチプロトコル ラベル スイッチング（MPLS）レイヤ 3 仮想プライベート ネットワーク（VPN）を構成する方法について説明します。

MPLS レイヤ 3 VPNs の概要

MPLS レイヤ 3 VPN は、MPLS プロバイダー コア ネットワークにより相互接続されている一連のサイトから構成されます。各カスタマーサイトでは、1つ以上のカスタマーエッジ（CE）ルータまたはレイヤ2スイッチが、1つ以上のプロバイダーエッジ（PE）ルータに接続されます。ここでは次の項目について説明します。

- MPLS レイヤ 3 VPN の定義
- MPLS レイヤ 3 VPN の動作方法
- MPLS レイヤ 3 VPN のコンポーネント
- ハブ アンド スポーク トポロジ
- MPLS VPN のための OSPF 模造リンクのサポート

MPLS レイヤ 3 VPN の定義

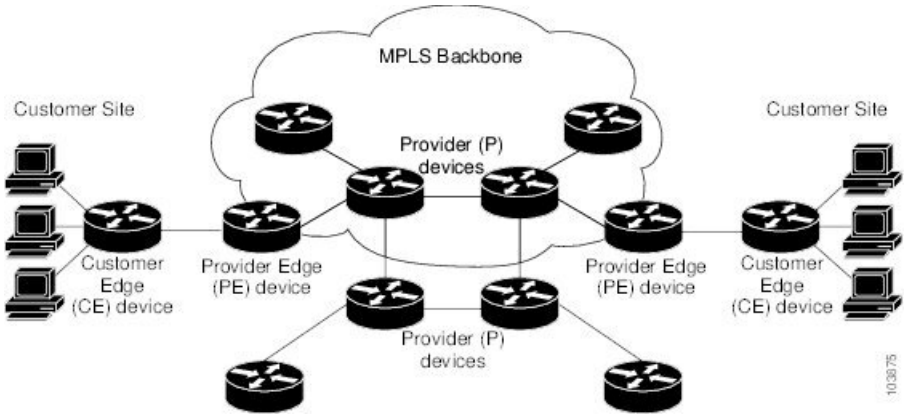
MPLS レイヤ 3 VPN はピア モデルに基づいており、これにより、サービス プロバイダーおよびカスタマーは、レイヤ3のルーティング情報を交換できます。プロバイダーは、カスタマーサイト間でデータをリレーします。このとき、カスタマーが直接何かを行う必要はありません。

新しいサイトが MPLS VPN に追加された場合、更新する必要があるのは、カスタマー サイトにサービスを提供するサービス プロバイダーのエッジ ルータだけです。

MPLS レイヤー 3 VPN には、以下のコンポーネントが含まれています。

- プロバイダー（P）ルータ：プロバイダー ネットワークのコア内のルータ。P ルータは MPLS スイッチングを実行しますが、ルーティングされるパケットに VPN ラベル（PE ルータによって割り当てられた、各ルート内の MPLS ラベル）を付加しません。
- プロバイダー エッジ（PE）ルータ：着信パケットが受信されるインターフェイスまたはサブインターフェイスに基づいて、着信パケットに VPN ラベルを付加するルータ。PE ルータは、CE ルータに直接接続します。
- カスタマーエッジ（CE）ルータ：ネットワーク上の PE ルータに接続するプロバイダーのネットワーク上のエッジルータ。CE ルータは、PE ルータとインターフェイスする必要があります。

図 *1 : MPLS* レイヤ *3 VPN* の基本用語



MPLS レイヤ 3 VPN の動作方法

MPLS レイヤ 3 VPN 機能は、MPLS ネットワークのエッジで有効になっています。PE ルータは、次のタスクを実行します。

- CE ルータとルーティング アップデートを交換する。
- CE ルーティング情報を VPN ルートに変換する。
- 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）を介して、他の PE ルータとレイヤ 3 VPN ルートを交換する。

MPLS レイヤ 3 VPN のコンポーネント

MPLS ベースの VPN ネットワークには、次の 3 つの主要コンポーネントがあります。

1. VPN ルート ターゲット コミュニティ：VPN ルート ターゲット コミュニティは、レイヤ 3 VPN コミュニティのすべてのメンバのリストです。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。
2. 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：マルチプロトコル BGP は、VPN コミュニティのすべてのメンバに VRF の到達可能情報を伝播します。VPN コミュニティ内のすべての PE ルータにマルチプロトコル BGP ピアリングを設定する必要があります。
3. MPLS 転送：MPLS は、VPN エンタープライズまたはサービス プロバイダー ネットワーク上のすべての 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。

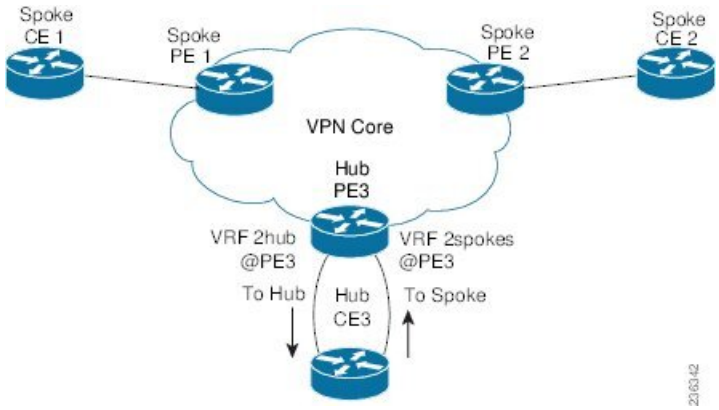
1 対 1 の関係は、カスタマー サイトと VPNs 間に必ずしも存在する必要はありません。1 つのサイトを複数の VPNs のメンバにできます。ただし、サイトは、1 つの VRF とだけ関連付けることができます。カスタマー サイトの VRF には、そのサイトがメンバとなっている VPNs からサイトへの、利用できるすべてのルートが含まれています。

ハブ アンド スポーク トポロジ

ハブ アンド スポーク トポロジは、スポーク プロバイダー エッジ（PE）ルータでの加入者間のローカル接続を禁止し、加入者がハブ サイトに常に接続されるようにします。同じ PE ルータに接続しているすべてのサイトは、ハブ サイトを使用して、サイト間のトラフィックを転送する必要があります。このトポロジより、スポーク サイトでのルーティングは、常にアクセス側インターフェイスからネットワーク側インターフェイスに対して、またはネットワーク側インターフェイスからアクセス側インターフェイスに対して実行されます。アクセス側インターフェイスからアクセス側インターフェイスへのルーティングは発生しません。ハブ アンド スポーク トポロジにより、サイト間のアクセス制限を維持できます。

ハブ アンド スポーク トポロジを使用すると、PE ルータが、トラフィックをハブ サイトを介して渡さずに、スポークをローカルに切り替えるという状況が回避されます。このトポロジにより、加入者が互いに直接接続することがなくなります。ハブアンドスポーク トポロジでは、スポークごとに 1 つの VRF は必要ありません。

図 *2:* ハブ アンド スポーク トポロジ



図に示すように、ハブ アンド スポーク トポロジは通常、2 つの VRF で設定されたハブ PE で設定されます。

- 専用リンクが設定された VRF 2hub がハブのカスタマー エッジ（CE）に接続されます。
- VRF 2spoke は、ハブ CE に接続された別の専用リンクを使用します。

内部ゲートウェイ プロトコル（IGP）または外部 BGP（eBGP）セッションは、通常、ハブ PE-CE リンクを介してセットアップされます。VRF 2hub は、すべてのスポーク PE からエクスポートされたすべてのルート ターゲットをインポートします。ハブ CE はスポーク サイトからのすべてのルートを学習し、それらをハブ PE の VRF 2spoke に再アドバタイズして戻します。VRF 2spoke は、これらすべてのルートをスポーク PE にエクスポートします。

ハブ PE とハブ CE の間の eBGP を使用する場合は、通常は禁止されているパスで自律システム（AS）番号を複製できるようにする必要があります。ハブ PE の VRF 2spoke のネイバー、およびすべてのスポーク PE の VPN アドレス ファミリ ネイバーでこの重複 AS 番号を許可するようにルータを設定できます。さらに、ハブ PE の VRF 2spoke でネイバーにルートを配布する場合は、ハブ CE でピア AS 番号チェックを無効にする必要があります。

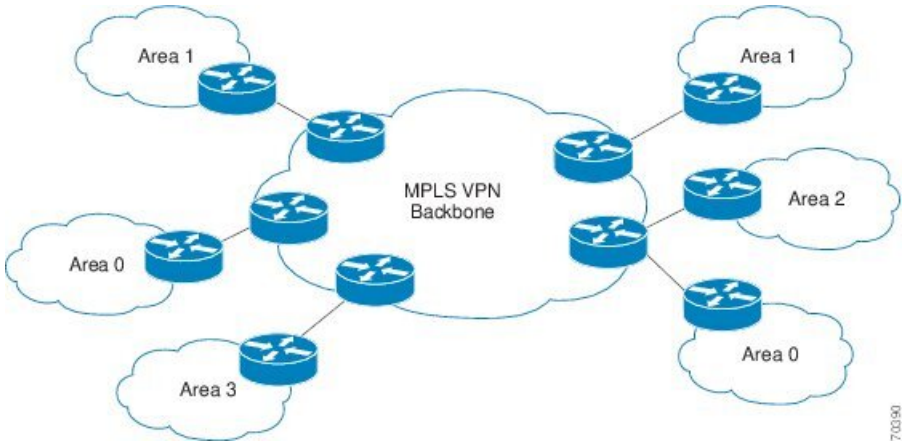
MPLS VPN のための OSPF 模造リンクのサポート

マルチプロトコル ラベル スイッチング（MPLS）VPN 構成では、Open Shortest Path First（OSPF）プロトコルを使用して、VPN バックボーン内のカスタマー エッジ（CE）デバイスをサービス プロバイダーエッジ（PE）デバイスに接続できます。多くのカスタマーは、OSPF をサイト内ルーティング プロトコルとして実行し、VPN サービスにサブスクライブし、MPLS VPN バックボーンで OSPF を（移行時または常時）使用してサイト間でルーティング情報を交換することを望んでいます。

MPLS VPN の OSPF 模造リンク サポートの利点は次のとおりです。

- MPLS VPN バックボーン全体でのクライアント サイトの接続：模造リンクによって、バックドア リンクを共有する OSPF クライアント サイトが、MPLS VPN バックボーンを介して通信を行い、VPN サービスに参加するようになります。
- MPLS VPN 設定での柔軟なルーティング：MPLS VPN 設定で模造リンクに対して設定する OSPF コストを使用して、OSPF クライアント サイトのトラフィックを、バックドア リンク経由にするか、または VPN バックボーン経由にするかを指定できます。

下の図に、OSPF を実行する各 VPN クライアント サイトを、MPLS VPN バックボーンで接続する例を示します。



OSPF を使用して PE デバイスと CE デバイスを接続するには、VPN サイトから学習したすべてのルーティング情報を、着信インターフェイスに関連付けられた VPN ルーティングおよび転送（VRF）インスタンスに格納します。VPN に接続された PE デバイス間では、ボーダーゲートウェイ プロトコル（BGP）を使用して、VPN ルートが交換されます。CE デバイスはこの VPN 内の他のサイトへのルートを、自分が接続された PE デバイスとのピアリングによって学習します。MPLS VPN スーパーバックボーンは、OSPF を実行する各 VPN サイトを内部接続するための追加のルーティング階層レベルを提供します。

OSPF ルートが MPLS VPN バックボーン全体に伝播されると、プレフィックスに関する追加情報が、BGP 拡張コミュニティ形式（ルート タイプ、ドメイン ID 拡張コミュニティ）で BGP アップデートに付加されます。このコミュニティ情報を使用して、受信した PE デバイスは、BGP ルートを OSPF PE-CE プロセスに再配布するときに生成するリンクステート アドバタイズメント（LSA）のタイプを決定します。このようにして、同じ VPN に属し、VPN バックボーン全体にアドバタイズされる内部 OSPF ルートが、リモート サイト上でエリア内ルートとして認識されます。

MPLS レイヤ 3 VPNs の前提条件

MPLS レイヤ 3 VPNs には次の前提条件があります。

- ネットワークに MPLS およびラベル配布プロトコル（LDP）を設定する必要があります。PE ルータを含む、コア内のすべてのルータは、MPLS 転送をサポートできる必要があります。
- MPLS の正しいライセンスおよび MPLS で使用する他の機能をインストールすることが必要です。

MPLS レイヤ 3 VPNs に関する注意事項と制限事項

MPLS レイヤ 3 VPNs 設定時の注意事項と制限事項は次のとおりです。

- Cisco Nexus 3600-R プラットフォーム スイッチおよび N9K-X9636C-RX、N9K-X9636C-R、N9K-X96136YC-R、および N9K-X9636Q-R ライン カードを搭載した および Cisco Nexus 9504 および 9508 プラットフォーム スイッチで、MPLS レイヤ 3 VPN（LDP）を設定できます。
- MPLS IP 転送はサポートされていないため、トンネルエンドポイントを終端するインターフェイスで有効になっていないことを確認してください。
- 着信パケットのラベルに基づいて転送の決定が行われるインターフェイスでは、MPLS IP 転送を有効にする必要があります。VPN ラベルがプレフィックス モードごとに割り当てられている場合は、PE と CE 間のリンクで MPLS IP 転送を有効にする必要があります。
- N9K-X9636C-R および N9K-X9636Q-R ラインカードを搭載した Cisco Nexus 9508 プラットフォーム スイッチのトラップ解決のハードウェア制限のため、インバンド経由でのスーパーバイザ バウンド パケットに uRPF が適用されない場合があります。
- -R シリーズ ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチでは、ブリッジ トラフィックが RACL にヒットしないように、RACL はルーティングされたトラフィックにのみ適用されます。これは、すべてのマルチキャスト OSPF 制御トラフィックに適用されます。
- -R シリーズ ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチでは、SUP への送信時に、明示的 NULL ラベルを持つ制御パケットは優先されません。これにより、明示的に NULL が設定されている場合、制御プロトコルのフラッピングが発生する可能性があります。
- 500K の規模でのラベルごとの統計は、ハードウェアの制限のため、-R シリーズ ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチではサポートされていません。
- -R シリーズ ラインカードを備えた Cisco Nexus 9500 プラットフォーム スイッチでの ARP スケーリングは、すべての 64K MAC が異なる場合、64K に制限されます。この制限は、インターフェイスに複数の等コストマルチパス（ECMP）が構成されている場合にも適用されます。
- MPLS の明示的 NULL のパケットは、デフォルトのラインカードプロファイルでは正しく解析されない場合があります。
- MPLS レイヤ 3 VPN は、次の CE-PE ルーティング プロトコルをサポートします。
  - BGP（IPv4 および IPv6）
  - 拡張内部ゲートウェイ プロトコル（EIGRP）（IPv4）
  - Open Shortest Path First（OSPFv2）
  - ルーティング情報プロトコル（RIPv2）
- インポート ルート マップの set ステートメントは無視されます。

- すべての iBGP および eBGP セッションの BGP 最小ルート アドバタイズメント インターバル（MRAI）値はゼロであり、設定できません。
- EIGRP に多数の BGP ルートが再配布されるハイ スケールなセットアップでは、EIGRP のコンバージェンス時間が BGP のコンバージェンス時間よりも長くなるように EIGRP シグナル タイマーの設定を変更する必要があります。このプロセスにより、EIGRP シグナルのコンバージェンス前にすべての BGP ルートを EIGRP に再配布することができます。
- MPLS レイヤ 3 VPN は、M3 シリーズ モジュールでサポートされています。
- PE と CE デバイス間のプロトコルとして OSPF を使用する場合、VPN バックボーン全体にルートがアドバタイズされる際、OSPF メトリックは保持されます。このメトリックは、リモート PE デバイスで適切なルートを選択するために使用されます。OSPF から BGP への再配布、および、BGP から OSPF への再配布において、メトリック値を変更しないでください。メトリック値を変更すると、ルーティング ループが発生する可能性があります。
- MPLS トラフィック エンジニアリング（RSVP）は、N9K-X9636C-R および N9K-X9636Q-R ラインカードを備えた Cisco Nexus 9508 プラットフォーム スイッチではサポートされていません。
- Cisco NX-OS リリース 9.3(1) 以降、BGP プレベスト パス挿入ポイント (POI) の動作が変更されました。このリリースでは、NX-OS RPM、BGP、および HMM ソフトウェアは単一のコスト コミュニティ ID（内部ルートの場合は 128、外部ルートの場合は 129）を使用して、BGP VPNv4 ルートを EIGRP 発信ルートとして識別します。コスト コミュニティ ID 128 または 129 に設定されたプレベスト パス値を持つルートのみが、コスト外部コミュニティとともに URIB にインストールされます。上記のコスト コミュニティ ID を伝える非 EIGRP 発信ルートは、プレベスト パス コスト コミュニティとともに URIB にインストールされます。その結果、URIB はこのコストを使用して、管理的距離とは異なる、iBGP を介して学習したルートとバックドア EIGRP の間のより適切なルートを識別します。

  コスト コミュニティ ID 128 または 129 に設定されたプレベスト パス値を持つルートのみが、コスト外部コミュニティとともに URIB にインストールされます。
- 出力 RACL（e-RACL）TCAM 機能と MPLS 拡張 ECMP 機能は相互に排他的です。Cisco Nexus N9K-X9636C-RX ライン カードで MPLS 拡張 ECMP（**hardware profile mpls extended-ecmp**） を有効にするには、e-RACL TCAM カービングを 0 に設定します。

  Cisco NX-OS リリース 10.3(1) 以降では、RX ベースのプラットフォームの **hardware profile mpls extended-ecmp** プロファイルの 21504 を超える RACL の TCAM リージョンをカービングする必要があります。 **hardware access-list tcam region racl** コマンドを使用して RACL TCAM リージョンを設定し、MPLS 拡張 ECMP を有効にすることができます。

MPLS レイヤ 3 VPNs のデフォルト設定

表 *1*: デフォルトの *MPLS* レイヤ *3 VPN* パラメータ

パラメータ	デフォルト
L3VPN 機能	無効
L3VPN SNMP 通知	無効
allowas-in（ハブアンドスポーク トポロジの場合）	0
disable-peer-as-check（ハブアンドスポーク トポロジの場合）	無効化

『Configuring MPLS Layer 3 VPNs』

OSPF ドメイン ID とタグについて

VRF 内の OSPF ルータ インスタンスの domain_ID を設定できます。OSPF では、Cisco NX-OS は domain_ID とドメイン タグを使用して、プロバイダー エッジ（PE）またはカスタマー エッジ（CE）での BGP ルート再配布の側面を制御します。

- 再配布される OSPF ルートのプライマリおよびセカンダリ domain_ID を設定できます。
- OSPF は、ドメイン タグを使用して OSPF プロセス ID を識別します。

ドメイン ID とドメイン タグの Cisco NX-OS 実装は、RFC 4577 に準拠しています。

（注） OSPF のプライマリとセカンダリの domain_ID とドメイン タグは、MPLS L3VPN 機能が有効になっている場合にのみ使用できます。

PE および CE 境界での OSPF の設定

ドメイン ID とドメイン タグを使用することで、NX-OS を設定して OSPF ルートを BGP ネットワークに再配布できます。また、BGP 再配布ルートを PE と CE の境界で OSPF に受信させることができます。次の項を参照してください。

- OSPF ドメイン ID とタグについて （8 ページ）
- OSPF ドメイン ID の構成 （9 ページ）

• セカンダリ ドメイン ID の構成 （10 ページ）
• OSPF ドメイン タグの設定 （9 ページ）

OSPF ドメイン タグの設定

ドメイン タグは、NX-OS が PE または CE で BGP に再配布する OSPF プロセス インスタンス番号を指定します。

始める前に

MPLS と OSPFv2 が有効になっていることを確認します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch-1# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		端末の構成を開始します

ステップ 2	**router ospf** *process-tag*
	例：
	`switch-1(config)# router ospf 101`
	`switch-1(config-router)#`
		ルータ構成モードを開始して、OSPF ルータ インスタンスを構成します。プロセス タグは、ルータを識別する 1 ～ 20 文字の英数字文字列です。

ステップ 3	**vrf** *vrf-name*
	例：
	`switch-1(config-router)# vrf pubstest`
	`switch-1(config-router-vrf)#`
		OSPF の特定の VRF インスタンスを入力します。VRF 名は、VRF を識別する 1 ～ 32 文字の英数字文字列です。

ステップ 4	**ospf domain-tag** *as-number*
	例：
	`switch-1(config-router-vrf)# domain-tag 9999`
	`nxosv2(config-router-vrf)#`
		ドメイン タグを設定します。ドメイン タグは、AS 番号を識別する 0 ～ 2147483647 の英数字の文字列です。

OSPF ドメイン ID の構成

VRF 内の OSPF ルータ インスタンスの domain_ID を設定して、CE または PE での OSPF への BGP ルートの再配布を制御できます。

この機能を削除するには、このコマンドの **no domain-id** 形式を使用します。

始める前に

OSPF domain_ID 機能を使用するには、MPLS L3VPN 機能と OSPFv2 機能の両方を有効にする必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch-1# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		端末の構成を開始します

ステップ 2	**router ospf** *process-tag*
	例：
	`switch-1(config)# router ospf 101`
	`switch-1(config-router)#`
		ルータ構成モードを開始して、OSPF ルータ インスタンスを構成します。プロセス タグは、ルータを識別する 1 ～ 20 文字の英数字文字列です。

ステップ 3	**vrf** *vrf-name*
	例：
	`switch-1(config-router)# vrf pubstest`
	`switch-1(config-router-vrf)#`
		OSPF の特定の VRF インスタンスを入力します。VRF 名は、VRF を識別する 1 ～ 32 文字の英数字文字列です。

ステップ 4	**domain-id** { *id* | **type** *domain-type* **value** *value* | **Null** } 例： `switch-1(config-router-vrf)# domain-id 19.0.2.0`	domain_ID と追加のパラメータを設定します。 • *id* は、ドメイン ID をドット付き 10 進表記で指定します (例: 1.2.3.4)。 • *type* は、0005 などの 4 バイト表記でドメイン タイプを指定します。 • *value* は、ドメイン値を 6 バイトの 16 進表記で指定します (例: 0x0005)。 Null 引数を使用して、domain_ID をクリアすることができます。

セカンダリ ドメイン ID の構成

VRF 内の OSPF ルータ インスタンスにセカンダリ domain_ID を設定して、CE または PE での OSPF への BGP ルートの再配布を制御できます。

domain-id Null コマンドを使用して、domain_ID を構成解除します。

始める前に

OSPFv2 および MPLS 機能が有効になっていることを確認します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch-1# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		端末の構成を開始します

ステップ 2	**router ospf** *process-tag*
	例：
	`switch-1(config)# router ospf 101`
	`switch-1(config-router)#`
		ルータ構成モードを開始して、OSPF ルータ インスタンスを構成します。プロセス タグは、ルータを識別する 1 ～ 20 文字の英数字文字列です。

ステップ 3	**vrf** *vrf-name*
	例：
	`switch-1(config-router)# vrf pubstest`
	`switch-1(config-router-vrf)#`
		OSPF の特定の VRF インスタンスを入力します。VRF 名は、VRF を識別する 1 ～ 32 文字の英数字文字列です。

ステップ 4	**domain-id** { *id* | **type** *domain-type* **value** *value* | **Null** } 例： `switch-1(config-router-vrf)# domain-id 19.0.2.0`	自律システムの domain_ID を設定します。

コア ネットワークの設定

MPLS レイヤ 3 VPN カスタマーのニーズの評価

MPLS レイヤ 3 VPN のカスタマーに最善のサービスを提供できるように、コア ネットワークトポロジを識別することができます。

- ネットワークのサイズを識別します。
  - 必要となるルータとポートの数を決定するために、次の内容を識別します。
  - サポートする必要があるカスタマーの数
  - カスタマーごとに必要となる VPN の数
  - 各 VPN に存在する、仮想ルーティングおよび転送インスタンスの数
- コア ネットワークで必要なルーティング プロトコルを決定します。

- MPLS VPN ハイ アベイラビリティのサポートが必要であるかどうかを判断します。

（注） MPLS VPN ノンストップ フォワーディングおよびグレースフル リスタートは、選択ルータおよび Cisco NX-OS リリースでサポートされています。BGP および LDP のグレースフル リスタートが有効であることを確認する必要があります。

- コア ネットワークのルーティング プロトコルを設定します。
- MPLS レイヤ 3 VPN コアで BGP 負荷共有および冗長パスが必要であるかどうかを決定します。

コアにおける MPLS の設定

コアのすべてのルータで MPLS をイネーブルにするには、ラベル配布プロトコルを設定する必要があります。次のいずれかをラベル配布プロトコルとして使用できます。

- MPLS ラベル配布プロトコル（LDP）。
- MPLS トラフィック エンジニアリング リソース予約プロトコル（RSVP）。

PE ルータおよびルート リフレクタでのマルチプロトコル BGP の設定

PE ルータおよびルート リフレクタでマルチプロトコル BGP 接続を設定できます。

始める前に

- BGP および LDP のすべてのルータでグレースフル リスタートがイネーブルになっていることを確認します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

	コマンドまたはアクション	目的

ステップ 3	**install feature-set mpls**
	例：
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをインストールします。

ステップ 4	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 5	**feature mpls l3vpn**
	例：
	`switch(config)# feature mpls l3vpn`
	`switch(config)#`
		MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 6	**router bgp** *as - number* 例： `switch(config)# router bgp 1.1`	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。as-number 引数は、ルータを他の BGP ルータに対して識別し、ルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
ステップ 7	**router-id** *ip-address* 例： `switch(config-router)# router-id 192.0.2.255`	（任意）BGP ルータ ID を設定します。この IP アドレスによって、この BGP スピーカを特定します。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。

ステップ 8	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.1`
	
	`switch(config-router-neighbor)#`
		エントリを iBGP ネイバー テーブルに追加します。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。

ステップ 9	**address-family { vpnv4 | vpnv6 } unicast** 例： `switch(config-router-neighbor)# address-family vpnv4 unicast`	アドレス ファミリ コンフィギュレーション モードを開始して、標準 VPNv4 または VPNv6 アドレス プレフィックスを使用する、BGP などのルーティング セッションを設定します。

	コマンドまたはアクション	目的
	`switch(config-router-neighbor-af)#`	
ステップ 10	**send-community extended** 例： `switch(config-router-neighbor-af)# send-community extended`	コミュニティ属性がBGPネイバーに送信されるように指定します。
ステップ 11	**show bgp { vpnv4 | vpnv6 } unicast neighbors** 例： `switch(config-router-neighbor-af)# show bgp vpnv4 unicast neighbors`	（任意）BGPネイバーに関する情報を表示します。
ステップ 12	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

MPLS VPN カスタマーの接続

カスタマーの接続を可能にするための、PE ルータでの VRF の定義

カスタマーの接続をイネーブルにするためPEルータにVRFを作成する必要があります。ルートターゲットを設定し、カスタマーのVPNサイトへのIPプレフィックスのインポート、およびBGPネットワークへのIPプレフィックスのエクスポートを制御します。必要に応じて、インポートまたはエクスポートルートマップを使用して、カスタマーVPNサイトにインポートされる、またはVPNサイトからエクスポートされるIPプレフィックスを、より詳細に制御できます。ルートマップを使用して、ルートのルートターゲット拡張コミュニティ属性に基づいて、VRFでのインポートまたはエクスポートに適したルートをフィルタリングできます。たとえば、ルートマップにより、インポートルートターゲットリスト上のコミュニティから、選択したルートへのアクセスが拒否される場合があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal switch(config)#`	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	**install feature-set mpls** 例：	MPLS フィーチャ セットをインストールします。

	コマンドまたはアクション	目的
	`switch(config)# install feature-set mpls` `switch(config)#`	

ステップ 3	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 4	**feature-set mpls l3vpn**
	例：
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 5	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context vpn1`
	`switch(config-vrf)#`
		VRF 名を割り当て、VRF コンフィギュレーション モードを開始することにより、VPN ルーティング インスタンスを定義します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 6	**rd** *route-distinguisher*
	例：
	`switch(config-vrf)# rd 1.2:1`
	`switch(config-vrf)#`
		ルート識別子を設定します。route-distinguisher 引数によって、8 バイトの値が IPv4 プレフィックスに追加され、VPN IPv4 プレフィックスが作成されます。RD は、次のいずれかの形式で入力できます。
		• 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。
		• 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		IPv4 アドレス ファミリ タイプを指定し、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。

ステップ 8	**route-target** { **import** | **export** } *route-target-ext-community* **}** 例： `switch(config-vrf-af-ipv4)# route-target import 1.0:1`	次のように VRF 用にルート ターゲット拡張コミュニティを指定します。 • import キーワードを使用すると、ターゲット VPN 拡張コミュニティからルーティング情報がインポートされます。

	コマンドまたはアクション	目的
		• export キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。 • route-target-ext-community 引数により、ルートターゲット拡張コミュニティ属性が、インポート、またはエクスポートのルートターゲット拡張コミュニティの VRF リストに追加されます。 route-target-ext-community 引数は、次のいずれかの形式で入力できます。 • 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。 • 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。
ステップ 9	**maximum routes** *max-routes* [**threshold** *value*] [**reinstall**] 例： `switch(config-vrf-af-ipv4)# maximum routes 10000`	（任意）VRF ルートテーブルに格納できる最大ルート数を設定します。max-routes の範囲は 1 ～ 4294967295 です。しきい値の値の範囲は 1 ～ 100 です。
ステップ 10	**import** [**vrf default** *max-prefix*] **map** *route-map* 例： `switch(config-vrf-af-ipv4)# import vrf default map vpn1-route-map`	（任意）デフォルト VRF からプレフィックスをインポートするための VRF のインポートポリシーを次のように設定します。 • max-prefix の範囲は 1 ～ 2147483647 です。デフォルトは 1000 プレフィックスです。 • route-map 引数は VRF のインポートルートマップとして使用されるルートマップを最大 63 文字の英数字文字列（大文字と小文字を区別）で指定します。

	コマンドまたはアクション	目的
ステップ 11	**show vrf** *vrf-name* 例： `switch(config-vrf-af-ipv4)# show vrf vpn1`	（任意）VRF の情報を表示します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 12	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

各 VPN カスタマー用の PE ルータでの VRF インスタンスの設定

PE ルータのインターフェイスまたはサブインターフェイスに仮想ルーティングおよび転送（VRF）インスタンスを関連付けることができます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**interface** *type number*
	例：
	`switch(config)# interface Ethernet 5/0`
	`switch(config-if)#`
		設定するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始する方法は次のとおりです。
		• type 引数で、設定するインターフェイスのタイプを指定します。
		• number 引数には、ポート、コネクタ、またはインターフェイス カード番号を指定します。

ステップ 3	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member vpn1`	指定したインターフェイスまたはサブインターフェイスに VRF を関連付けます。vrf-name 引数は、VRF に割り当てる名前です。
ステップ 4	**show vrf** *vrf-name* **interface** 例： `switch(config-if)# show vrf vpn1 interface`	（任意）VRF に関連付けられるインターフェイスの情報を表示します。vrf-name 引数には最大 32 文字の英数字文字列を

	コマンドまたはアクション	目的
		指定します。大文字と小文字は区別されます。
ステップ 5	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

PE ルータと CE ルータ間でのルーティング プロトコルの設定

PE ルータと CE ルータ間でスタティックまたは直接接続されたルートの設定

スタティック ルートを使用する PE-to-CE ルーティング セッション用の PE ルータを設定することができます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context vpn1`
	`switch(config-vrf)#`
		VRF 名を割り当て、VRF コンフィギュレーションモードを開始することにより、VPN ルーティング インスタンスを定義します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 3	{ **ip ipv6** } **route** *prefix nexthop* 例： `switch(config-vrf)# ip route 192.0.2.1/28 ethernet 2/1`	PE から CE への各セッション用のスタティック ルート パラメータを定義します。prefix および nexthop は次のとおりです。 • IPv4：ドット付き 10 進表記 • IPv6：16 進形式

ステップ 4	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		IPv4 アドレス ファミリ タイプを指定し、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

ステップ 5	**feature bgp** *as - number*
	例：
	`switch(config-vrf-af)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 6	**router bgp** *as - number* 例： `switch(config)# router bgp 1.1`	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。as-number 引数は、ルータを他の BGP ルータに対して識別し、ルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 7	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		BGP プロセスを VRF に関連付けます。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 8	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		IPv4 アドレス ファミリ タイプを指定し、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。

ステップ 9	**redistribute static route-map** *map-name* 例： `switch(config-router-vrf-af)# redistribute static route-map StaticMap`	スタティック ルートを BGP に再配布します。 マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。
ステップ 10	**redistribute direct route-map** *map-name* 例： `switch(config-router-vrf-af)# redistribute direct route-map StaticMap`	直接接続されたルートを BGP に再配布します。 マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。
ステップ 11	**show { ipv4 | ipv6 } route vrf** *vrf-name* 例：	（任意）ルートに関する情報を表示します。

	コマンドまたはアクション	目的
	`switch(config-router-vrf-af)# show ip ipv4 route vrf vpn1`	vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 12	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

BGP を PE ルータと CE ルータ間のルーティング プロトコルに設定

eBGP を使用して PE-to-CE ルーティング セッション用の PE ルータを設定できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 3	**router bgp** *as - number*
	例：
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		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。
		as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 4	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		BGP プロセスを VRF に関連付けます。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

	コマンドまたはアクション	目的

ステップ 5	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.1`
	`switch(config-router-neighbor)#`
		エントリを iBGP ネイバー テーブルに追加します。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 6	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		アドレス ファミリ コンフィギュレーション モードを開始して、標準 IPv4 または IPv6 アドレス プレフィックスを使用する、BGP などのルーティング セッションを設定します。

ステップ 7	**show bgp { vpnv4 | vpnv6 } unicast neighbors vrf** *vrf-name* 例： `switch(config-router-neighbor-af)# show bgp vpnv4 unicast neighbors`	（任意）BGP ネイバーに関する情報を表示します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 8	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

PE ルータと CE ルータ間での RIPv2 の設定

RIP を使用して PE-to-CE ルーティング セッション用の PE ルータを設定できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**feature rip**
	例：
	`switch(config)# feature rip`
	`switch(config)#`
		RIP 機能を有効にします。

ステップ 3	**router rip** *instance-tag* 例：	RIP をイネーブルにし、ルータ 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
	`switch(config)# router rip Test1`	instance-tag には最大 20 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 4	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		RIP プロセスを VRF に関連付けます。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 5	**address-family ipv4 unicast**
	例：
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		アドレス ファミリ タイプを指定し、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。

ステップ 6	**redistribute** { **bgp** as | **direct** | { **egrip** | **ospf** | **rip** } *instance-tag* | static } **route-map** *map-name vrf-name* 例： `switch(config-router-vrf-af)# show ip rip vrf vpn1`	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。 as 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。instance-tag は、大文字と小文字が区別される 20 文字以下の任意の英数字文字列にできます。
ステップ 7	**show ip rip vrf** *vrf-name* 例： `switch(config-router-vrf-af)# show ip rip vrf vpn1`	（任意）RIP に関する情報を表示します。 vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 8	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

PE ルータと CE ルータ間での OSPF の設定

OSPFv2 を使用して PE-to-CE ルーティング セッション用の PE ルータを設定できます。MPLS ネットワークの一部ではない OSPF バックドア リンクがある場合は、オプションで OSPF 模造リンクを作成できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**feature ospf**
	例：
	`switch(config)# feature ospf`
	`switch(config)#`
		OSPF 機能をイネーブルにします。

ステップ 3	**router ospf** *instance-tag* 例： `switch(config)# router ospf Test1`	OSPF をイネーブルにし、ルータ コンフィギュレーション モードを開始します。 instance-tag には最大 20 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 4	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		ルータ VRF 設定モードを開始します。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 5	**area** area-id **sham-link** *source-address destination-address* 例： `switch(config-router-vrf)# area 1 sham-link 10.2.1.1 10.2.1.2`	（任意）PE インターフェイス上の模造リンクを、指定した OSPF エリア内に設定します。エンドポイントとして各ループバック インターフェイスを IP アドレスで指定します。 PE の両エンドポイントで模造リンクを設定する必要があります。

ステップ 6	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		アドレス ファミリ タイプを指定し、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。

ステップ 7	**redistribute** { **bgp** as | **direct** | { **egrip** | **ospf** | **rip** } *instance-tag* | static } **route-map** *map-name* 例：	BGP を EIGRP に再配布します。 BGP ネットワークの自律システム番号は、このステップで設定されます。BGP

	コマンドまたはアクション	目的
	`switch(config-router-vrf-af)# redistribute bgp 1.0 route-map BGPMap`	を CE サイトの EIGRP に再配布して、EIGRP 情報を伝送する BGP ルートを受け入れるようにする必要があります。また、BGP ネットワークにメトリックを指定する必要があります。 マップ-名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

ステップ 8	**autonomous-system** *as-number*
	例：
	`switch(config-router-vrf-af)#`
	`autonomous-system 1.3`
		（任意）自律システム番号を、カスタマーサイトのこのアドレスファミリに指定します。
		as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 9	**show ip egrip vrf** *vrf-name* 例： `switch(config-router-vrf-af)# show ipv4 eigrp vrf vpn1`	（任意）この VRF の EIGRP に関する情報を表示します。 vrf-name には最大 32 文字の英数字文字列を指定します。大文字と-小文字は区別されます。
ステップ 10	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

PE ルータと CE ルータ間での EIGRP の設定

PE ルータと CE ルータ間で Enhanced Interior Gateway Routing Protocol（EIGRP）を使用して MPLS 対応 BGP コア ネットワーク経由で EIGRP カスタマー ネットワークがトランスペアレントに接続されるように PE ルータを設定できます。これにより、EIGRP ルートが BGP ネットワークの VPN を経由して内部 BGP（iBGP）ルートとして再配布されます。

始める前に

ネットワーク コアで BGP を設定する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**feature eigrp**
	例：
	`switch(config)# feature eigrp`
	`switch(config)#`
		EIGRP 機能を有効にします。

ステップ 3	**router eigrp** *instance-tag* 例： `switch(config)# router eigrp Test1`	EIGRP インスタンスを設定し、ルータ コンフィギュレーション モードを開始します。 instance-tag には最大 20 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 4	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf vpn1`
	`switch(config-router-vrf)#`
		ルータ VRF 設定モードを開始します。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 5	**address-family ipv4 unicast**
	例：
	`switch(config-router-vrf)#`
	`address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		（任意）標準 IPv4 アドレス プレフィックスを使用するルーティング セッションを設定するために、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	**redistribute bgp** *as-number* **route-map** *map-name*
	例：
	`switch(config-router-vrf-af)#`
	`redistribute bgp 235354 route-map`
	`mtest1`
		ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。
		AS 番号としては、16 ビット整数または 32 ビット整数があり得ます。後者の場合、上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。instance-tag には最大 20 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 7	**show ip ospf** *instance-tag* **vrf** *vrf-name* 例：	（任意）OSPF に関する情報を表示します。

	コマンドまたはアクション	目的
	`switch(config-router-vrf-af)# show ip rip vrf vpn1`	
ステップ 8	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

MPLS VPN での BGP の PE-CE 再配布の設定

PE-CE プロトコルが BGP ではない場合は、MPLS レイヤ 3 VPN サービスを提供するすべての PE ルータで、PE-CE ルーティング プロトコルが配布されるように BGP を設定する必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 3	**router bgp** *instance-tag*
	例：
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		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。as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 4	**router id** *ip-address*
	例：
	`switch(config-router)# router-id 192.0.2.255 1`
	`switch(config-router)#`
		（任意）BGP ルータ ID を設定します。この IP アドレスによって、この BGP スピーカを特定します。このコマンドによって、BGP ネイバーセッションの自動通知およびセッション リセットが開始されます。

	コマンドまたはアクション	目的

ステップ 5	**router id** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2`
	`switch(config-router-neighbor)#`
		BGP ネイバーテーブルまたはマルチプロトコル BGP ネイバー テーブルにエントリを追加します。ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 6	**update-source loopback [0 | 1]** 例： `switch(config-router-neighbor)# update-source loopback 0#`	BGP セッションの送信元アドレスを指定します。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-neighbor)# address-family vpnv4`
	`switch(config-router-neighbor-af)#`
		アドレス ファミリ コンフィギュレーション モードを開始して、標準 VPNv4 または VPNv6 アドレス プレフィックスを使用する、BGP などのルーティング セッションを設定します。unicast キーワード（任意）では、VPNv4 または VPNv6 ユニキャスト アドレス プレフィックスを指定します。

ステップ 8	**send-community extended** 例： `switch(config-router-neighbor-af)# send-community extended`	コミュニティ属性が BGP ネイバーに送信されるように指定します。

ステップ 9	**vrf** *vrf-name*
	例：
	`switch(config-router-neighbor-af)# vrf vpn1`
	`switch(config-router-vrf)#`
		ルータ VRF 設定モードを開始します。
		vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 10	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		標準 IPv4 または VPNv6 アドレス プレフィックスを使用するルーティング セッションを設定するために、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 11	**redistribute** { **direct** | { **egrip** | **ospfv3** | **ospfv3** |**rip** } *instance-tag* | static } **route-map** *map-name* 例： `switch(config-router-af-vrf)# redistribute eigrp Test2 route-map EigrpMap`	ルートを 1 つのルーティング ドメインから他のルーティング ドメインに再配布します。as 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

	コマンドまたはアクション	目的
		instance-tag には最大 20 文字の英数字文字列を指定します。大文字と小文字は区別されます。map-name には最大 63 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 12	**show bgp { ipv4 | ipv6 } unicast vrf** *vrf-name* 例： `switch(config-router--vrf-af)# show bgp ipv4 unicast vrf vpn1vpn1`	（任意）BGP に関する情報を表示します。vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。
ステップ 13	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ハブ アンド スポーク トポロジの設定

ハブ PE ルータにおける VRF の設定

ハブ PE ルータ上でハブ アンド スポーク VRF を設定できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**install feature-set mpls**
	例：
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをインストールします。

ステップ 3	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 4	**feature-set mpls l3vpn** 例：	MPLS レイヤ 3 VPN 機能をイネーブルにします。

	コマンドまたはアクション	目的
	`switch(config)# feature-set mpls l3vpn` `switch(config)#`	

ステップ 5	**vrf context** *vrf-hub*
	例：
	`switch(config)# vrf context 2hub`
	`switch(config-vrf)#`
		VRF 名を割り当て、VRF コンフィギュレーションモードを開始することにより、PE ハブの VPN ルーティング インスタンスを定義します。vrf-hub 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 6	**rd** *route-distinguisher*
	例：
	`switch(config-vrf)# rd 1.2:1`
	`switch(config-vrf)#`
		ルート識別子を設定します。route-distinguisher 引数によって、8 バイトの値が IPv4 プレフィックスに追加され、VPN IPv4 プレフィックスが作成されます。RD は、次のいずれかの形式で入力できます。
		• 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。
		• 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		IPv4 アドレス ファミリ タイプを指定し、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。

ステップ 8	**route-target** { **import** | **export** } route-target-ext-community **}** 例： `switch(config-vrf-af-ipv4)# route-target import 1.0:1`	次のように VRF 用にルート ターゲット拡張コミュニティを指定します。 • **import** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティからインポートされます。 • **export** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。 • route-target-ext-community 引数により、ルート ターゲット拡張コミュニティ属性が、インポート、またはエクスポートのルート ターゲッ

	コマンドまたはアクション	目的
		ト拡張コミュニティの VRF リストに追加されます。 route-target-ext-community 引数は、次のいずれかの形式で入力できます。 • 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。 • 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。

ステップ 9	**vrf context** *vrf-spoke*
	例：
	`switch(config-vrf-af-ipv4)# vrf context 2spokes`
	`switch(config-vrf)#`
		VRF 名を割り当て、VRF コンフィギュレーションモードを開始することにより、PE スポークの VPN ルーティングインスタンスを定義します。vrf-spoke 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 10	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		IPv4 アドレス ファミリ タイプを指定し、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。

ステップ 11	**route-target** { **import** | **export** } *route-target-ext-community* **}** 例： `switch(config-vrf-af-ipv4)# route-target export 1:100`	次のように VRF 用にルート ターゲット拡張コミュニティを指定します。 • VRF 用にルート ターゲット拡張コミュニティを作成します。**import** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティからインポートされます。**export** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。 route-target-ext-community 引数により、ルート ターゲット拡張コミュニティ属性が、インポート、またはエクスポートのルート ターゲット拡張コミュニティの VRF リスト

	コマンドまたはアクション	目的
		に追加されます。 route-target-ext-community 引数は、次のいずれかの形式で入力できます。 • 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。 • 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。
ステップ 12	**show running-config vrf** *vrf-name* 例： `switch(config-vrf-af-ipv4)# show running-config vrf 2spokes`	（オプション）VRF の実行コンフィギュレーションを表示します。 vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。 。
ステップ 13	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ハブ PE ルータにおける eBGP の設定

eBGP を使用して PE-to-CE ハブ ルーティング セッションを設定できます。

（注） すべての CE サイトが同じ BGP AS 番号を使用している場合は、次のタスクを実行する必要があります。

- PE（ハブ）で BGP **as-override** コマンドを設定するか、受信 CE ルータで **allowas-in** コマンドを設定します。
- ある ASN から学習した BGP ルートを同じ ASN に戻してアドバタイズするには、ループバックを防止するために、PE ルータで **disable-peer-as-check** コマンドを設定します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**feature-set mpls** 例： `switch(config)# feature-set mpls`	MPLS フィーチャ セットをイネーブルにします。
ステップ 3	**feature mpls l3vpn** 例： `switch(config)# feature mpls l3vpn`	MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 4	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 5	**router bgp** *as - number*
	例：
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		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。
		as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 6	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2`
	`switch(config-router-neighbor)#`
		エントリを iBGP ネイバー テーブルに追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

	コマンドまたはアクション	目的

ステップ 7	**address-family** { **ipv4** | **ipv6** } **unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、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。

ステップ 8	**send-community extended** 例： `switch(config-router-neighbor-af)# send-community extended`	（任意）BGP を設定し、拡張コミュニティ リストをアドバタイズします。

ステップ 9	**vrf** *vrf-hub*
	例：
	`switch(config-router-neighbor-af)# vrf 2hub`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。*vrf-hub* 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 10	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		BGP またはマルチプロトコル BGP ネイバー テーブルに、この VRF のためのエントリを追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 11	**address-family** { **ipv4** | **ipv6** } **unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、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。

ステップ 12	**as-override** 例： `switch(config-router-vrf-neighbor-af)# as-override`	（オプション）更新を送信するときに AS 番号を上書きします。すべての BGP サイトが同じ AS 番号を使用している場合、次のコマンドのいずれか： • PE（ハブ）で BGP as-override コマンドを設定します または • 受信 CE ルータで allowas-in コマンドを設定します。

	コマンドまたはアクション	目的

ステップ 13	**vrf** *vrf-spoke*
	例：
	`switch(config-router-vrf-neighbor-af)# vrf 2spokes`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。
		vrf-spoke 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 14	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		BGP またはマルチプロトコル BGP ネイバー テーブルに、この VRF のためのエントリを追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 15	**address-family** { **ipv4** | **ipv6** } **unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、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。

ステップ 16	**allowas-in** [*number*] 例： `switch(config-router-vrf-neighbor-af)# allowas-in 3`	(オプション) AS パスでの AS 番号の重複を許可します。 VPN アドレス ファミリ コンフィギュレーション モード（PE スポーク）およびネイバー モード（PE ハブ）で、このパラメータを設定します。
ステップ 17	**show running-config bgp** *vrf-name* 例： `switch(config-router-vrf-neighbor-af)# show running-config bgp`	（任意）BGP の実行コンフィギュレーションを表示します。
ステップ 18	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ハブ CE ルータにおける eBGP の設定

eBGP を使用して PE-to-CE ハブ ルーティング セッションを設定できます。

（注） すべての CE サイトが同じ BGP AS 番号を使用している場合は、次のタスクを実行する必要があります。

- PE（ハブ）で as-override コマンドを設定するか、受信 CE ルータで allowas-in コマンドを設定します。
- CE ルータで disable-peer-as-check コマンドを設定します。
- ある ASN から学習した BGP ルートを同じ ASN に戻しアドバタイズするには、ループバックを防止するために、PE ルータで disable-peer-as-check コマンドを設定します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**feature-set mpls** 例： `switch(config)# feature-set mpls`	MPLS フィーチャ セットをイネーブルにします。
ステップ 3	**feature mpls l3vpn** 例： `switch(config)# feature mpls l3vpn`	MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 4	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 5	**router bgp** *as - number*
	例：
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		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。
		as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

	コマンドまたはアクション	目的

ステップ 6	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2`
	`switch(config-router-neighbor)#`
		エントリを iBGP ネイバー テーブルに追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、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。

ステップ 8	**send-community extended** 例： `switch(config-router-neighbor-af)# send-community extended`	（任意）BGP を設定し、拡張コミュニティ リストをアドバタイズします。

ステップ 9	**vrf** *vrf-hub*
	例：
	`switch(config-router-neighbor-af)# vrf 2hub`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。*vrf-hub* 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 10	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		BGP またはマルチプロトコル BGP ネイバー テーブルに、この VRF のためのエントリを追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 11	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、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
ステップ 12	**as-override** 例： `switch(config-router-vrf-neighbor-af)# as-override`	（オプション）更新を送信するときにAS番号を上書きします。すべてのBGPサイトが同じ AS 番号を使用している場合、次のコマンドのいずれか： • PE（ハブ）で BGP **as-override** コマンドを設定します または • 受信 CE ルータで **allowas-in** コマンドを設定します。

ステップ 13	**vrf** *vrf-spoke*
	例：
	`switch(config-router-vrf-neighbor-af)# vrf 2spokes`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。
		vrf-spoke 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

ステップ 14	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		BGP またはマルチプロトコル BGP ネイバー テーブルに、この VRF のためのエントリを追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 15	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		IP アドレス ファミリ タイプを指定し、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。

ステップ 16	**allowas-in** [*number*] 例： `switch(config-router-vrf-neighbor-af)# allowas-in 3`	(オプション) AS パスでの AS 番号の重複を許可します。 VPN アドレス ファミリ コンフィギュレーション モード（PE スポーク）およびネイバー モード（PE ハブ）で、このパラメータを設定します。
ステップ 17	**show running-config bgp** *vrf-name* 例：	（任意）BGP の実行コンフィギュレーションを表示します。

	コマンドまたはアクション	目的
	`switch(config-router-vrf-neighbor-af)# show running-config bgp`	
ステップ 18	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

スポーク PE ルータにおける VRF の設定

スポーク PE ルータ上でハブ アンド スポーク VRFs を設定できます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**install feature-set mpls**
	例：
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		MPLS 機能セットを有効化します。

ステップ 3	**feature-set mpls**
	例：
	`switch(config)# feature-set mpls`
	`switch(config)#`
		MPLS フィーチャ セットをイネーブルにします。

ステップ 4	**feature-set mpls l3vpn**
	例：
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		MPLS レイヤ 3 VPN 機能をイネーブルにします。

ステップ 5	**vrf context** *vrf-spoke*
	例：
	`switch(config)# vrf context spoke`
	`switch(config-vrf)#`
		VRF 名を割り当て、VRF コンフィギュレーションモードを開始することにより、PE スポークの VPN ルーティング インスタンスを定義します。vrf-spoke 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。

	コマンドまたはアクション	目的

ステップ 6	**rd** *route-distinguisher*
	例：
	`switch(config-vrf)# rd 1.101`
	`switch(config-vrf)#`
		ルート識別子を設定します。
		route-distinguisher 引数によって、8 バイトの値が IPv4 プレフィックスに追加され、VPN IPv4 プレフィックスが作成されます。RD は、次のいずれかの形式で入力できます。
		• 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。
		• 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		IPv4 アドレス ファミリ タイプを指定し、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。

ステップ 8	**route-target** { **import** | **export** } route-target-ext-community **}** 例： `switch(config-vrf-af-ipv4)# route-target import 1.0:1`	次のように VRF 用にルート ターゲット拡張コミュニティを指定します。 • **import** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティからインポートされます。 • **export** キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。 • route-target-ext-community 引数により、ルート ターゲット拡張コミュニティ属性が、インポート、またはエクスポートのルート ターゲット拡張コミュニティの VRF リストに追加されます。 route-target-ext-community 引数は、次のいずれかの形式で入力できます。 • 16 ビットまたは 32 ビットの AS 番号:32 ビットの番号。1.2:3 など。

	コマンドまたはアクション	目的
		• 32 ビットの IP アドレス:16 ビットの番号。192.0.2.1:1 など。
ステップ 9	**show running-config vrf** *vrf-name* 例： `switch(config-vrf-af-ipv4)# show running-config vrf 2spokes`	（オプション）VRF の実行コンフィギュレーションを表示します。 vrf-name 引数には最大 32 文字の英数字文字列を指定します。大文字と小文字は区別されます。 。
ステップ 10	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

スポーク PE ルータにおける eBGP の設定

eBGP を使用して PEスポーク ルーティング セッションを設定できます。

（注） すべての CE サイトが同じ BGP AS 番号を使用している場合は、次のタスクを実行する必要があります。

- 認識しているスポーク ルータで allowas-in コマンドを設定します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**feature-set mpls** 例： `switch(config)# feature-set mpls`	MPLS フィーチャ セットをイネーブルにします。
ステップ 3	**feature mpls l3vpn** 例： `switch(config)# feature mpls l3vpn`	MPLS レイヤ 3 VPN 機能をイネーブルにします。

	コマンドまたはアクション	目的

ステップ 4	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 5	**router bgp** *as - number*
	例：
	`switch(config)# router bgp 100`
	`switch(config-router)#`
		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。
		as-number 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 6	**neighbor** *ip-address***remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 63.63.0.63 remote-as 100`
	`switch(config-router-neighbor)#`
		エントリを iBGP ネイバー テーブルに追加します。
		• ip-address 引数には、ドット付き 10 進表記でネイバーの IP アドレスを指定します。
		• as-number 引数には、ネイバーが属している自律システムを指定します。

ステップ 7	**address-family { ipv4 | ipv6 } unicast**
	例：
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IPv4 または IPv6 アドレス ファミリ タイプを指定し、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。

ステップ 8	**allowas-in** *number* 例： `switch(config-router-vrf-neighbor-af)# allowas-in 3`	（任意）指定した回数だけ、PE ASN が設定された AS パスを許可します。 • 値の範囲は 1 ～ 10 です。 • すべての BGP サイトが同じ AS 番号を使用している場合は、次のコマンドを構成します。 （注）

	コマンドまたはアクション	目的
		PE（ハブ）で **BGP as-override** コマンドを設定するか、受信 CE ルータで **allowas-in** コマンドを設定します。 *as-number* 引数は、ルータを他の BGP ルータに対して識別し、転送するルーティング情報にタグを設定する自律システムの番号を示します。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。
ステップ 9	**send-community extended** 例： `switch(config-router-neighbor)# send-community extended`	（任意）BGP を設定し、拡張コミュニティ リストをアドバタイズします。
ステップ 10	**show running-config bgp** 例： `switch(config-router-vrf-neighbor-af)# show running-config bgp`	（任意）BGP の実行コンフィギュレーションを表示します。
ステップ 11	**copy running-config startup-config** 例： `switch(config-router-vrf)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ハードウェア プロファイル コマンドを使用した MPLS の設定

リリース 7.0(3)F3(3) 以降、N9K-X9636C-R、N9K-X9636C-RX、および N9K-X9636Q-R ラインカードを備えた Cisco Nexus 9508 スイッチは、複数のハードウェア プロファイルをサポートします。スイッチでハードウェア プロファイル コンフィギュレーション コマンドを使用して、MPLS および/または VXLAN を設定できます。ハードウェア プロファイル コンフィギュレーション コマンドは、スイッチで使用可能な適切なコンフィギュレーション ファイルを呼び出します。VXLAN はデフォルトで有効になっています。

始める前に

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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	**feature bgp**
	例：
	`switch(config)# feature bgp`
	`switch(config)#`
		BGP 機能をイネーブルにします。

ステップ 3	**hardware profile** [**vxlan** | **mpls**] **module all** 例： `switch(config)# hardware profile mpls module all`	すべてのスイッチ モジュールで MPLS を有効にします。。

ステップ 4	**show hardware profile module** [**all** | *number*]
	例：
	`switch(config)# show hardware profile module all`
	`switch(config)#`
		すべてのモジュールまたは特定のモジュールのハードウェア プロファイルを表示します。

ステップ 5	**show module internal sw info** | [**i** | **mpls**] 例： `switch(config)# show module internal sw info`	スイッチのソフトウェア情報を表示します。
ステップ 6	**show running configuration** | [**i** | **mpls**] 例： `switch(config)# show module internal sw info`	実行設定を表示します。

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。