



## レイヤ 3 外部接続の設定

- [レイヤ 3 外部接続の設定のモードについて \(1 ページ\)](#)
- [レイヤ 3 外部接続の設定 \(3 ページ\)](#)
- [外部ネットワークへのルーテッド接続 \(4 ページ\)](#)
- [レイヤ 3 ルーティングとサブインターフェイス ポート チャネル \(16 ページ\)](#)
- [レイヤ 3 アウトからレイヤ 3 アウト内部 VRF への漏洩 \(23 ページ\)](#)
- [SVI 外部カプセル化の範囲について \(28 ページ\)](#)
- [SVI 自動状態について \(31 ページ\)](#)
- [インターフェイスとスタティック ルーティングの設定 \(33 ページ\)](#)
- [OSPF の設定 \(36 ページ\)](#)
- [BGP の設定 \(44 ページ\)](#)
- [EIGRP の設定 \(62 ページ\)](#)
- [ルート マップの設定 \(70 ページ\)](#)
- [双方向ルート転送 \(BFD\) の設定 \(81 ページ\)](#)
- [レイヤ 3 マルチキャストの設定 \(97 ページ\)](#)
- [外部 L3 EPG の設定 \(113 ページ\)](#)
- [名前付きモードを使用したレイヤ 3 外部接続の設定 \(116 ページ\)](#)
- [IPv6 ネイバー探索 \(131 ページ\)](#)
- [『Configuring HSRP』 \(136 ページ\)](#)
- [Cisco ACI GOLF \(139 ページ\)](#)
- [Multipod\\_Fabric \(159 ページ\)](#)
- [リモート リーフ スイッチ \(166 ページ\)](#)
- [トランジット ルーティング \(174 ページ\)](#)

## レイヤ 3 外部接続の設定のモードについて

APIC は設定のための複数のユーザ インターフェイス (UI) をサポートしているため、1 つの UI を使用して設定を作成し、その後、別の UI を使用して設定を変更する場合は、予期しないインタラクションが潜んでいます。ここでは、さらに他の APIC のユーザ インターフェイスを使

用した可能性がある場合、APIC NX-OS スタイルの CLI を使用してレイヤ3外部接続を設定するための考慮事項を説明します。

APIC NX-OS スタイルの CLI を使用してレイヤ3外部接続を設定する場合、次の2つのモードを選択することができます。

- よりシンプルな暗黙モードは、APIC GUI または REST API と互換性がありません。
- 名前付き（または明示）モードは、APIC GUI および REST API と互換性があります。

いずれの場合も、設定は互換性がない UI では読み取り専用であると考えてください。

### モードの違いについて

どちらのモードでも、構成設定は API の **l3extOut** クラスのインスタンスである内部コンテナオブジェクト「L3 Outside」（または「L3Out」）内で定義されます。2つのモード間の主な違いは、このコンテナオブジェクトインスタンスの命名にあります。

- 暗黙モード：コンテナのネーミングは潜在的であり、CLI コマンドには表示されません。CLI は、これらのオブジェクトを内部的に作成し保持します。
- 名前付きモード：名前はユーザーが決定します。名前付きモードの CLI コマンドには、追加の **l3Out** フィールドがあります。名前付き L3Out がを正常に設定され障害を回避するためには、ユーザーが外部レイヤ3用の API オブジェクトモデルを理解する必要があります。



(注) 「名前付きモードセクションを使用したレイヤ3外部接続の設定」セクションの手順を除き、このガイドでは、暗黙モードの手順を説明します。

### 注意事項および制約事項

- 同じ APIC インターフェイスでは、両方のモードを、次の制限でレイヤ3外部接続を設定するために一緒に使用することができます。テナント VRF、およびリーフの特定の組み合わせのレイヤ3外部接続設定は、1つのモードを介してのみ実行できます。
- 特定のテナント VRF の場合、外部 L3 EPG を配置できるポリシー ドメインは、名前付きモードまたは暗黙モードのいずれかになります。推奨する設定方式は、特定のテナント VRF が、レイヤ3外部接続用に展開されたすべてのノード全体で、特定のテナント VRF の組み合わせに対して1つのモードだけを使用することです。モードは、異なるテナントまたは異なる VRF 全体で変えることができ、制限は適用されません。
- 外部レイヤ3機能は、次の例外を除いて、両方の設定モードでサポートされます
  - L4～L7 サービス アプライアンスを使用したルーティング ピアリングとルートヘルスインジェクション(RHI)は、名前付きモードでのみをサポートされます。名前付きモードは、ルーティングピアリングが含まれるテナント VRF のすべての境界リーフスイッチ全体で使用する必要があります。

- 暗黙モード CLI 手順を使用して作成されたレイヤ3 外部ネットワーク オブジェクト (l3extOut) は、「\_ui\_」で始まる名前でも識別され、GUI で読み取り専用としてマークされます。CLI は、インターフェイス、プロトコル、ルートマップ、EPG などの機能で、これらの外部 L3 ネットワークを分割します。REST API を介して実行される設定変更は、この構造を破棄することができ、CLI を介してさらなる変更を防ぐことができます。

このようなオブジェクトを削除する手順については、『*APIC Troubleshooting Guide*』の「*Troubleshooting Unwanted \_ui\_ Objects*」を参照してください。

## レイヤ3 外部接続の設定

外部ネットワークへのレイヤ3 (L3) ルーティング接続の設定は、次のコンポーネントで構成されています。

- インターフェイス：レイヤ3 ポートの設定インターフェイス、サブインターフェイス、外部ルータに接続するために使用される外部 SVI。
- ルーティングプロトコルの設定：CLI は静的ルート、BGP、OSPF、EIGRP プロトコル設定をサポートします。
- ルートマップ制御：ルートマップは、prefixes/BD パブリックサブネットと一致させ、ルーティング制御ポリシーを適用するために使用されます。ルートマップが作成されると、「in」(BGP または OSPF)、「out」(BGP、OSPF、EIGRP) などの、方向のルーティングプロトコルと関連付けることができます。

インターフェイス、ルートプロトコル、およびルートマップに関する設定は、構成リーフの設定モードでリーフスイッチ単位で保持されます。

- L3 外部 EPG：コントラクトと QoS ポリシーを適用するためのエンドポイントグループに分類されるテナント VRF の外部サブネットのリスト。L3 外部 EPG (プレフィックス EPG ともいいます) には、他の外部 L3 EPG およびアプリケーション EPG とコントラクトを持つことができます。L3 外部 EPG 設定はテナント設定で保持されます。L3 外部 EPG は、VRF が設定されているノードのサブセットに展開できます。

レイヤ3 外部接続を設定するための手順は、次のように要約できます。

1. テナントの下に VRF を作成します。
2. 境界リーフスイッチ上の VRF を設定して展開します。
3. 境界リーフインターフェイス上のレイヤ3 インターフェイスを設定します。
4. リーフスイッチ上のルートマップを設定します。
5. リーフおよびリーフインターフェイスの下にルーティングプロトコル (BGP、OSPF、EIGRP) を設定します。
6. テナントの下に L3 外部 EPG を作成して設定します。
7. 境界リーフスイッチの外部 L3 EPG を展開します。

# 外部ネットワークへのルーテッド接続

## 外部ネットワークへルートされた接続について

ネットワーク構成 (L3Out) 外部レイヤ 3 では、ファブリック以外のトラフィックを転送する方法を定義します。レイヤ 3 はし、他のノードのアドレスを見つける、ルートを選択して、サービスの品質を選択して、入力して、終了、およびファブリックを移動する際は、トラフィックを転送に使用されます。



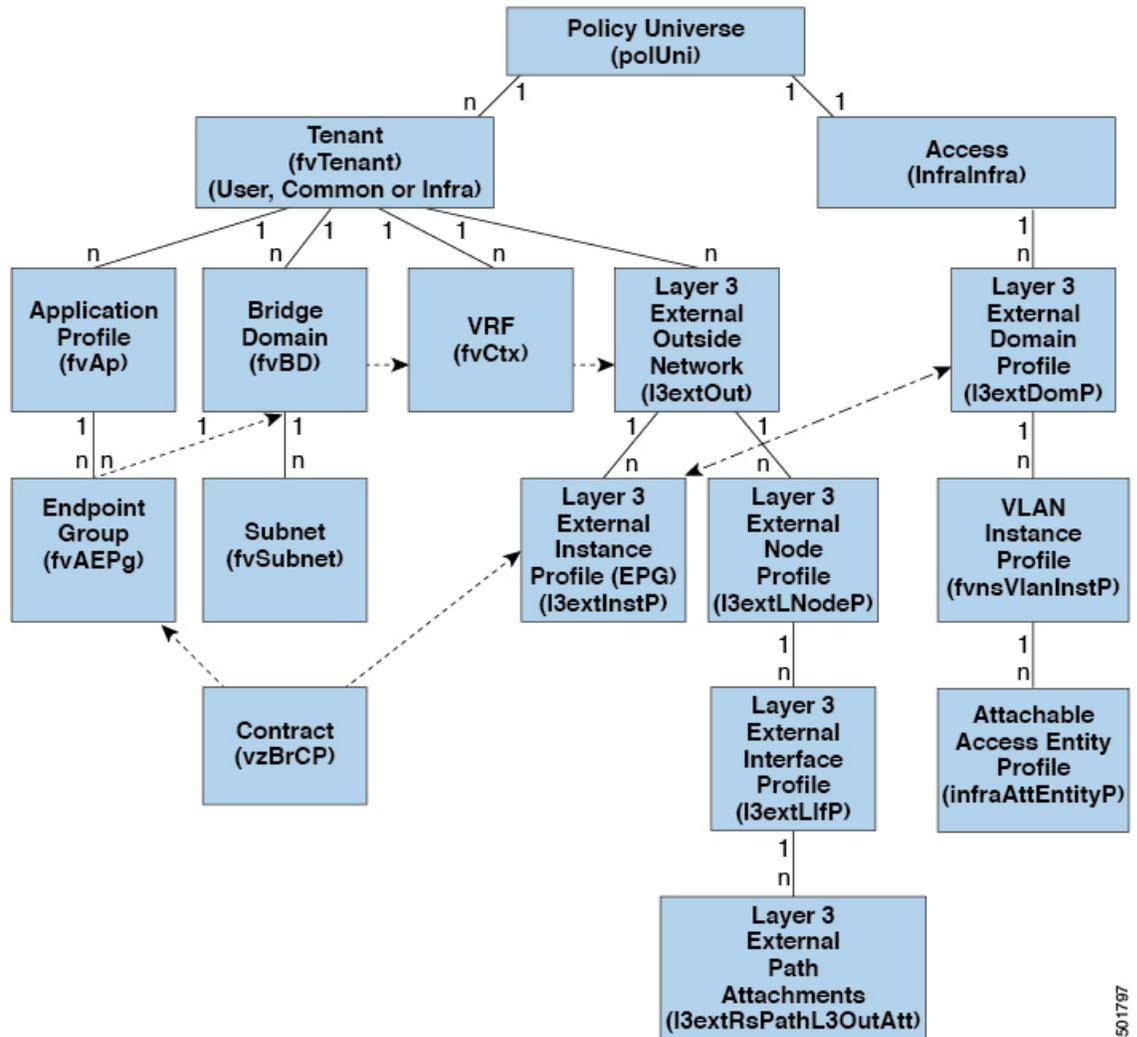
(注) ガイドラインとの設定と接続の外部レイヤ 3 を維持するための注意事項は、次を参照してください。 [外部ネットワークへの接続をルーティングするための注意事項 \(6 ページ\)](#)。

L3Outs の種類についての詳細は、[外部レイヤ 3 Outside 接続タイプ \(8 ページ\)](#) を参照してください。

## 外部ネットワークへのルーテッド接続のためのレイヤ 3 Out

外部ネットワークへのルーテッド接続は、次の図の階層で示すようにファブリック アクセス (infraInfra) 外部ルーテッドドメイン (l3extDomP) をレイヤ 3 外部外側ネットワーク (l3extOut) のテナント レイヤ 3 外部インスタンス プロファイル (l3extInstP または外部 EPG) に関連付けることによって有効になります。

図 1: レイヤ3外部接続のポリシーモデル



501797

レイヤ3外部アウトサイドネットワーク (l3extOut オブジェクト) には、ルーティングプロトコルのオプション (BGP、OSPF、または EIGRP またはサポートされている組み合わせ) およびスイッチとインターフェイス固有の設定が含まれています。l3extOut にルーティングプロトコル (たとえば、関連する仮想ルーティングおよび転送 (VRF) およびエリア ID を含む OSPF) が含まれる一方で、レイヤ3外部インターフェイスのプロファイルには必要な OSPF インターフェイスの詳細が含まれます。いずれも OSPF のイネーブル化に必要です。

l3extInstP EPG は、コントラクトを通してテナント EPG に外部ネットワークを公開します。たとえば、Web サーバのグループを含むテナント EPG は、l3extOut に含まれるネットワーク設定に応じてコントラクトを介して l3extInstP EPG と通信できます。外部ネットワーク設定は、ノードを L3 外部ノードプロファイルに関連付けることで複数のノードに容易に再利用できます。同じプロファイルを使用する複数のノードをフェールオーバーやロードバランシングのために設定できます。ノードを複数の l3extOuts に追加することで、l3extOuts に関連付けられている VRF がノードでも展開されます。拡張性に関する情報については、現行の「*Verified Scalability Guide for Cisco ACI*」を参照してください。

## 外部ネットワークへの接続をルーティングするための注意事項

レイヤ3外部接続を作成し、維持する際には、次のガイドラインを使用してください。

トピック	注意またはガイドライン
入力ベース ポリシーの適用	Cisco APIC リリース 1.2(1) 以降、入力ベース ポリシーの適用により、入出力両方向でレイヤ3アウトサイド (L3Out) トラフィックにポリシー適用を定義できます。デフォルトでは入力になっています。リリース 1.2(1) 以降にアップグレード中、既存の L3Out 設定が出力に設定され、動作が既存の設定と一致します。特別なアップグレードのシーケンスは必要ありません。アップグレード後、グローバルプロパティ値を入力に変更します。変更されると、システムがルールとプレフィックスエントリを再プログラミングします。規則は出力リーフから削除され、入力リーフ上に既存の規則がない場合は、入力リーフ上にインストールされます。既存の設定がない場合、Actrl プレフィックス エントリが入力リーフ上にインストールされます。ダイレクト サーバリターン (DSR) および属性 EPG には入力ベースのポリシー適用が必要です。vzAny と禁止コントラクトは、入力ベースのポリシー適用を契約無視します。入力には中継規則が適用されます。
L3Outs によるブリッジドメイン	テナントのブリッジドメインには、共通テナントでプロビジョニングされている l3extOut によってアドバタイズされたパブリック サブネットを含めることができます。
BGP 最大プレフィックス制限	Cisco APIC リリース 1.2 (1x) 以降、BGP l3extOut 接続のテナントポリシーは、最大プレフィックス制限を使用して設定できます。これにより、ピアから受信されるルートプレフィックスの数をモニタし、制限することができます。最大プレフィックス制限を超えると、ログエントリが記録され、さらにプレフィックスが拒否されます。カウントが一定の間隔でしきい値を下回る場合、接続を再起動することができますが、そうしない場合接続がシャットダウンします。一度に1つのオプションだけを使用できます。デフォルト設定では20,000プレフィックスに制限され、その後は新しいプレフィックスは拒否されます。拒否オプションが導入されると、APIC でエラーが発生する前に BGP は設定されている制限よりも1つ多くプレフィックスを受け入れます。

トピック	注意またはガイドライン
MTU	<p>Cisco ACI は、IP フラグメンテーションをサポートしていません。したがって、外部ルータへのレイヤ3 Outside (L3Out) 接続、または Inter-Pod Network (IPN) を介した multipod 接続を設定する場合は、MTU が両側で適切に設定されていることが重要です。ACI、Cisco NX-OS、Cisco IOS などの一部のプラットフォームでは、設定された MTU 値は IP ヘッダーを考慮に入れていますが (結果として、最大パケットサイズは、ACI で 9216 バイト、NX-OS および IOS で 9000 バイトに設定されます)。ただし、IOS XR などの他のプラットフォームは、パケットヘッダーのを除く MTU 値を設定します (結果として最大パケットサイズは 8986 バイトになります)。</p> <p>各プラットフォームの適切な MTU 値については、それぞれの設定ガイドを参照してください。</p> <p>CLI ベースのコマンドを使用して MTU をテストすることを強く推奨します。たとえば、Cisco NX-OS CLI で <code>ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1</code> などのコマンドを使用します。</p>
レイヤ4から7	<p>マルチノードサービス グラフを使用する際、異なる VRF で2つの EPG が必須です。これらの機能では、システムはレイヤ3 検索を実行する必要があるため、EPG が異なる VRF を分離する必要があります。この制限は、レイヤ2 およびレイヤ3 の検索に基づいてレガシサービスの挿入に続きます。</p>
L3Outs の QoS	<p>L3Out 用の QoS ポリシーを設定し、L3Out が存在する BL スイッチで適用されるポリシーを有効にするには、次の注意事項に従ってください。</p> <ul style="list-style-type: none"> <li>• VRF ポリシー制御の適用方向を <b>出力</b> に設定する必要があります。</li> <li>• VRF ポリシー制御適用の優先度設定を <b>有効</b> に設定する必要があります。</li> <li>• L3Out を使用して EPG 間の通信を制御するコントラクトを設定する際に、コントラクトまたはコントラクトの件名に QoS クラスまたはターゲット DSCP を含めません。</li> </ul>

## 外部レイヤ 3 Outside 接続タイプ

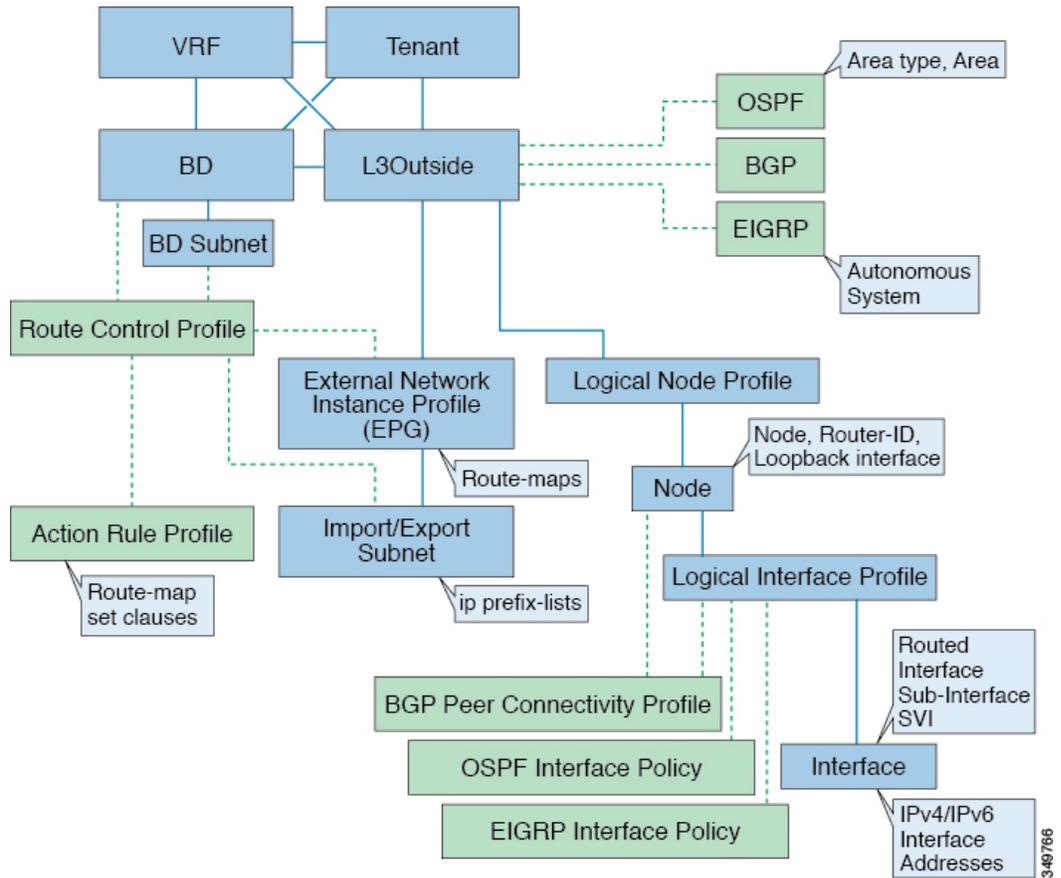
ACI は、以下の外部レイヤ 3 Outside 接続オプションをサポートします。

- スタティック ルーティング (IPv4 および IPv6 でサポート)
- 標準および NSSA エリアの OSPFv2 (IPv4)
- 標準および NSSA エリアの OSPFv3 (IPv6)
- iBGP (IPv4 および IPv6)
- eBGP (IPv4 および IPv6)
- BGP (IPv4 および IPv6)

外部レイヤ 3 Outside 接続は、以下のインターフェイスでサポートされます。

- レイヤ 3 ルーテッド インターフェイス
- 802.1Q タギング対応のサブインターフェイス：サブインターフェイスを使用すると、複数のプライベート ネットワークに対するレイヤ 2 外部接続を提供できます。
- スイッチ仮想インターフェイス (SVI)：SVI インターフェイスを使用すると、レイヤ 2 とレイヤ 3 をサポートする同じ物理インターフェイスをレイヤ 2 外部接続とレイヤ 3 外部接続に使用できます。

図 2: ACI レイヤ3 管理対象オブジェクト



L3Outside 接続に使用される管理対象オブジェクトは、次のとおりです。

- 外部レイヤ3 Outside (L3ext) : ルーティングプロトコルオプション (OSPF エリアタイプ、エリア、EIGRP AS、BGP)、プライベートネットワーク、外部物理ドメイン。
- 論理ノードプロファイル : 外部レイヤ3 Outside 接続に対して1つ以上のノードが定義されたプロファイル。ルータ ID とループバック インターフェイス設定はプロファイルで定義されます。



(注) 複数の外部レイヤ3 Outside 接続間の同じノードには同じルータ ID を使用してください。

- 論理インターフェイスプロファイル : IPv4 および IPv6 インターフェイスの IP インターフェイス設定。これは、ルートインターフェイス、ルーテッドサブインターフェイス、および SVI でサポートされます。SVI は、物理ポート、ポートチャネルまたは VPC で設定できます。
- OSPF インターフェイスポリシー : OSPF のネットワークタイプ、優先順位などの詳細が含まれています。

- EIGRP インターフェイス ポリシー：タイマー、スプリット ホライズン タイマーなどの詳細が含まれています。
- BGP ピア接続プロファイル：ほとんどの BGP ピア設定、リモート AS、ローカル AS、および BGP ピア接続オプションが設定されるプロファイル。BGP ピア接続プロファイルは、ノードプロファイルの下の論理インターフェイス プロファイルまたはループバック インターフェイスに関連付けることができます。これは、BGP ピアリングセッションの update-source 設定を決定します。
- 外部ネットワーク インスタンス プロファイル (EPG) (l3extInstP)：外部 EPG はプレフィックススペースの EPG または InstP とも呼ばれます。インポートおよびエクスポートのルート制御ポリシー、セキュリティ インポート ポリシー、およびコントラクトの関連付けは、このプロファイルで定義されます。単一 L3Out に複数の外部 EPG を設定できます。単一外部レイヤ 3 Outside 接続で別のルートまたはセキュリティ ポリシーが定義されている場合、複数の外部 EPG を使用できます。1 つの外部 EPG または複数の外部 EGP がルートマップにまとめられます。外部 EPG で定義されるインポート/エクスポートサブネットは、ルートマップの IP プレフィックスリストの match 句と関連しています。外部 EPG は、インポートセキュリティサブネットとコントラクトが関連付けられる場所でもあります。これは、この L3out のトラフィックの許可またはドロップに使用されます。
- アクションルールプロファイル：アクションルールプロファイルは、L3Out のルートマップの set 句を定義するために使用されます。サポートされる set 句は、BGP communities (standard および extended)、Tags、Preference、Metric、および Metric type です。
- ルート制御プロファイル：ルート制御プロファイルは、アクションルールプロファイルを参照するために使用されます。これは、アクションルールプロファイルの順序付きプロファイルにすることができます。ルート制御プロファイルは、テナント BD、BD サブネット、外部 EPG、または外部 EPG サブネットで参照できます。

BGP、OSPF、および EIGRP L3Out 用の追加のプロトコル設定が存在します。これらの設定は、GUI の [ACI Protocol Policies] セクションでテナントごとに設定されます。



- (注) 外部 EPG (中継ルーティング ケース) の間でポリシーの適用を設定する際には、エクスポートルート制御、集約エクスポート、および外部のセキュリティのために、デフォルトプレフィックスである 0/0 で 2 番目の外部 EPG (InstP) を設定する必要があります。さらに、優先グループを除外し、中継 InstPs 間で任意の契約 (または希望の契約) を使用する必要があります。

## NX-OS スタイルの CLI を使用したテナント ネットワークのレイヤ 3 Outside の設定

次の手順では、テナントネットワークのネットワークの外部レイヤ 3 を設定する方法について説明します。次に、NX-OS CLI を使用してテナント VRF 外部 L3 接続にノードと L3 ポートを配備する例を示します。

この例は、わかりやすくするための手順に分割されます。マージされた例については、[NX-OS スタイル CLI の例 : L3Out \(14 ページ\)](#) を参照してください。

#### 始める前に

- ノード、ポート、機能プロファイル、AEP、レイヤ3 ドメインを設定します。
- 使用して VLAN ドメイン設定、`vlan ドメイン ドメイン` および `vlan vlan 範囲 コマンド`。
- BGP ルート リフレクタ ポリシーを設定し、ファブリック内でルーテッドを伝達します。

コマンドを使用して、これらの前提条件の例を参照してください。[NX-OS スタイル CLI の例: L3Out の前提条件 \(14 ページ\)](#)。

#### 手順

##### ステップ1 テナントおよび VRF を設定します。

この例では VRF `v1` でテナント `t1` を設定します。これらはまだ展開されていません。

例 :

```
apicl# configure
apicl(config)# tenant t1
apicl(config-tenant)# vrf context v1
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# exit
apicl(config)#
```

##### ステップ2 L3Out のノードとインターフェイスを設定します。

この例では設定 VRF `v1` ノード `103` (border リーフ スイッチ) と呼ばれるで `nodep1`、ルータ ID を `11.11.11.103`。インターフェイスの設定も `eth1/3` ルーテッドインターフェイス (レイヤ3 のポート)、IP アドレスとして `12.12.12.3/24` とレイヤ3 ドメイン `dom1`。

例 :

```
apicl(config)# leaf 103
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# router-id 11.11.11.103
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# interface ethernet 1/3
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vrf member tenant t1 vrf v1
apicl(config-leaf-if)# ip address 12.12.12.3/24
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
```

##### ステップ3 ルーティング プロトコルを設定します。

この例では、BGP ピアのアドレスを使用して、プライマリのルーティング プロトコルとして BGP を設定 `15.15.15.2 ASN 100` ドルとします。

例 :

```

apic1(config)# leaf 103
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

```

**ステップ 4** オプション。接続ルーティングプロトコルを設定します。

この例では、定期的なエリア ID と、通信プロトコルとして OSPF を設定 0.0.0.0 、ループバックアドレスと 30.30.30.0 。

例：

```

apic1(config)# leaf 103
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 30.30.30.0
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit

```

**ステップ 5** ノード 103 上に外部 EPG を設定します。

この例では、ネットワークで 20.20.20.0/24 外部ネットワークとして設定されている extnw1 。

例：

```

apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 20.20.20.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw1
apic1(config-leaf-vrf)# exit

```

**ステップ 6** オプション。ルートマップを設定します。

この例では、ルートマップ設定 rp1 アウトバウンド方向に BGP ピアの。ルートマップがルートの宛先に一致するのに適用される 200.3.2.0/24 。

また、正常な一致で(ルートには、この範囲が一致する)ルート AS パスアトリビュートが更新され、 200 および 100 。

例：

```

apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 200.3.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config)# leaf 103
apic1(config-leaf)# router bgp 100

```

```
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)#exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# exit
```

### ステップ 7 ブリッジドメインを追加します。

例：

```
apicl(config)# tenant t1
apicl(config-tenant)# bridge-domain bd1
apicl(config-tenant-bd)# vrf member v1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# interface bridge-domain bd1
apicl(config-tenant-interface)# ip address 44.44.44.1/24 scope public
apicl(config-tenant-interface)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# route-map rp1
apicl(config-leaf-vrf-route-map)# match bridge-domain bd1 tenant t1
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# exit
```

### ステップ 8 ノード 101 で EPG アプリケーションを作成します。

例：

```
apicl(config)# tenant t1
apicl(config-tenant)# application appl
apicl(config-tenant-app)# epg epg1
apicl(config-tenant-app-epg)# bridge-domain member bd1
apicl(config-tenant-app-epg)# exit
apicl(config-tenant-app)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/3
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# switchport trunk allowed vlan 2011 tenant t1 application appl epg
epg1
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
apicl(config)#
```

### ステップ 9 フィルタ (アクセス リスト) と契約を作成します。

例：

```
apicl(config)# tenant t1
apicl(config-tenant)# access-list http-filter
apicl(config-tenant-acl)# match ip
apicl(config-tenant-acl)# match tcp dest 80
apicl(config-tenant-acl)# exit
apicl(config-tenant)# contract httpCtrct
apicl(config-tenant-contract)# scope vrf
apicl(config-tenant-contract)# subject subj1
apicl(config-tenant-contract-subj)# access-group http-filter both
apicl(config-tenant-contract-subj)# exit
apicl(config-tenant-contract)# exit
```

ステップ 10 契約を設定し、Epg に関連付けます。

例 :

```
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# contract provider httpCtrct
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# application appl
apic1(config-tenant-app)# epg epg1
apic1(config-tenant-app-epg)# contract consumer httpCtrct
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
apic1(config)#
```

## NX-OS スタイル CLI の例: L3Out の前提条件

L3Out を設定する前に、次の手順を実行します。

1. VLAN ドメインを設定します。

```
apic1# configure
apic1(config)# vlan-domain dom1
apic1(config-vlan)# vlan 1024-2048
apic1(config-vlan)# exit
```

2. BGP ルート リフレクタを設定します:

```
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 100
apic1(config-bgp-fabric)# route-reflector spine 104,105
```

## NX-OS スタイル CLI の例 : L3Out

次の例は、L3Out を設定する手順のマージバージョン NX-OS スタイル CLI を使用します。  
L3Out を設定する前に、次の前提条件を設定します。

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 11.11.11.103
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 12.12.12.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
```

```
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant t1 vrf v1
apicl(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 30.30.30.0
apicl(config-leaf-ospf-vrf)# exit
apicl(config-leaf-ospf)# exit
apicl(config-leaf)# exit
apicl(config)# tenant t1
apicl(config-tenant)# external-l3 epg extnw1
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# match ip 20.20.20.0/24
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 103
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# external-l3 epg extnw1
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# template route group match-rule1 tenant t1
apicl(config-route-group)# ip prefix permit 200.3.2.0/24
apicl(config-route-group)# exit
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# route-map rp1
apicl(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# router bgp 100
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apicl(config-leaf-bgp-vrf-neighbor)#exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# exit
apicl(config)# tenant t1
apicl(config-tenant)# bridge-domain bd1
apicl(config-tenant-bd)# vrf member v1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# interface bridge-domain bd1
apicl(config-tenant-interface)# ip address 44.44.44.1/24 scope public
apicl(config-tenant-interface)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# route-map map1
apicl(config-leaf-vrf-route-map)# match bridge-domain bd1 tenant t1
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# exit
apicl(config)# tenant t1
apicl(config-tenant)# application app1
apicl(config-tenant-app)# epg epg1
apicl(config-tenant-app-epg)# bridge-domain member bd1
apicl(config-tenant-app-epg)# exit
apicl(config-tenant-app)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/3
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# switchport trunk allowed vlan 2011 tenant t1 application app1 epg
epg1
```

```

apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# tenant t1
apic1(config-tenant)# access-list http-filter
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# exit
apic1(config-tenant)# contract httpCtrct
apic1(config-tenant-contract)# scope vrf
apic1(config-tenant-contract)# subject subj1
apic1(config-tenant-contract-subj)# access-group http-filter both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# contract provider httpCtrct
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# application appl
apic1(config-tenant-app)# epg epg1
apic1(config-tenant-app-epg)# contract consumer httpCtrct
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
apic1(config)#

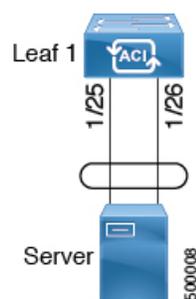
```

## レイヤ3ルーティングとサブインターフェイス ポート チャンネル

### レイヤ3ポートチャンネルについて

以前、Cisco APICではレイヤ2ポートチャンネルのみサポートしていましたが。リリース3.2(1)より、Cisco APICではレイヤ3ポートチャンネルもサポートしています。

図3:スイッチポートチャンネル設定



### ポートチャンネルのNX-OSは、CLIを使用してをルーテッドレイヤ3の設定

この手順では、レイヤ3ルーテッドポートチャンネルを設定します。

## 手順

	コマンドまたはアクション	目的
ステップ1	<b>configure</b> 例： apicl# <b>configure</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>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	リーフスイッチまたはリーフスイッチの設定を指定します。 <i>Node-id</i> は形式 <i>node-id1-node-id2</i> の単一ノードIDまたはIDの範囲となる可能性があり、設定が適用されます。
ステップ3	<b>interface port-channel channel-name</b> 例： apicl(config-leaf)# <b>interface port-channel po1</b>	指定したポートチャネルの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4	<b>no switchport</b> 例： apicl(config-leaf-if)# <b>no switchport</b>	レイヤ3インターフェイスを可能になります。
ステップ5	<b>vrf member vrf-name tenant tenant-name</b> 例： apicl(config-leaf-if)# <b>vrf member v1 tenant t1</b>	この仮想ルーティングおよび転送(VRF)インスタンスとL3ポリシー、外部には、このポートチャネルを関連付けます場所。  <ul style="list-style-type: none"> <li>• <i>Vrf-name</i> は VRF 名です。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。</li> <li>• テナント名は、テナント名です。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。</li> </ul>
ステップ6	<b>vlan-domain member vlan-domain-name</b> 例： apicl(config-leaf-if)# <b>vlan-domain member dom1</b>	以前に設定された VLAN ドメインには、ポートチャネルのテンプレートを関連付けます。
ステップ7	<b>ip address ip-address / subnet-mask</b> 例： apicl(config-leaf-if)# <b>ip address 10.1.1.1/24</b>	指定したインターフェイスの IP アドレスとサブネットマスクを設定します。

	コマンドまたはアクション	目的
ステップ 8	<b>ipv6 address <i>sub-bits/prefix-length</i> preferred</b> 例 : <pre>apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred</pre>	IPv6 の一般的なプレフィックスに基づいて IPv6 アドレスを設定し、インターフェイスにおける IPv6 処理をイネーブルにします。場所 : <ul style="list-style-type: none"> <li>• <i>sub-bits</i> i 引数は、<i>prefix-name</i> 引数で指定された一般的なプレフィックスによって提供されるプレフィックスに連結する、アドレスのサブプレフィックスビットおよびホストビットです。sub-bits 引数は、RFC 2373 に記載された形式で指定する必要があります。この形式では、アドレスは、16 進数値を 16 ビット単位でコロンで区切って指定します。</li> <li>• <i>Prefix-length</i> は IPv6 プレフィックスの長さです。プレフィックス (アドレスのネットワーク部分) を構成するアドレスの上位連続ビット数を示す 10 進値です。10 進数値の前にスラッシュ記号が必要です。</li> </ul>
ステップ 9	<b>ipv6 link-local <i>ipv6-link-local-address</i></b> 例 : <pre>apic1(config-leaf-if)# ipv6 link-local fe80::1</pre>	インターフェイスに IPv6 リンクローカルアドレスを設定します。
ステップ 10	<b>mac-address <i>mac-address</i></b> 例 : <pre>apic1(config-leaf-if)# mac-address 00:44:55:66:55::01</pre>	インターフェイス MAC アドレスを手動で設定します。
ステップ 11	<b>mtu <i>mtu-value</i></b> 例 : <pre>apic1(config-leaf-if)# mtu 1500</pre>	このサービスクラスの MTU を設定します

## 例

この例では、基本レイヤ 3 ポートチャネルを設定する方法を示します。

```
apic1# configure
```

```

apicl(config)# leaf 101
apicl(config-leaf)# interface port-channel po1
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vrf member v1 tenant t1
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# ip address 10.1.1.1/24
apicl(config-leaf-if)# ipv6 address 2001::1/64 preferred
apicl(config-leaf-if)# ipv6 link-local fe80::1
apicl(config-leaf-if)# mac-address 00:44:55:66:55::01
apicl(config-leaf-if)# mtu 1500

```

## NX-OS CLI を使用したレイヤ3サブインターフェイスポートチャネルの設定

この手順では、レイヤ3サブインターフェイスポートチャネルを設定します。

### 手順

	コマンドまたはアクション	目的
ステップ1	<b>configure</b> 例： apicl# <b>configure</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>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	リーフスイッチまたはリーフスイッチの設定を指定します。 <i>Node-id</i> は形式 <i>node-id1-node-id2</i> の単一ノードIDまたはIDの範囲となる可能性があり、設定が適用されます。
ステップ3	<b>vrf member vrf-name tenant tenant-name</b> 例： apicl(config-leaf-if)# <b>vrf member v1 tenant t1</b>	この仮想ルーティングおよび転送(VRF)インスタンスとL3アウトサイドポリシーにポートチャネルを関連付けます。場所： <ul style="list-style-type: none"> <li>• <i>Vrf-name</i> はVRF名です。32文字以内の英数字のストリング（大文字と小文字を区別）で指定します。</li> <li>• テナント名は、テナント名です。32文字以内の英数字のストリング（大文字と小文字を区別）で指定します。</li> </ul>

	コマンドまたはアクション	目的
ステップ 4	<b>vlan-domain member <i>vlan-domain-name</i></b> 例 : <pre>apic1(config-leaf-if)# vlan-domain member dom1</pre>	以前に設定された VLAN ドメインには、ポートチャネルのテンプレートを関連付けます。
ステップ 5	<b>ip address <i>ip-address / subnet-mask</i></b> 例 : <pre>apic1(config-leaf-if)# ip address 10.1.1.1/24</pre>	指定した インターフェイスの IP アドレスとサブネット マスクを設定します。
ステップ 6	<b>ipv6 address <i>sub-bits/prefix-length preferred</i></b> 例 : <pre>apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred</pre>	<p>IPv6 の一般的なプレフィックスに基づいて IPv6 アドレスを設定し、インターフェイスにおける IPv6 処理をイネーブルにします。場所 :</p> <ul style="list-style-type: none"> <li>• <i>sub-bits</i> 引数は、<i>prefix-name</i> 引数で指定された一般的なプレフィックスによって提供されるプレフィックスに連結する、アドレスのサブプレフィックスビットおよびホスト ビットです。<i>sub-bits</i> 引数は、RFC 2373 に記載された形式で指定する必要があります。この形式では、アドレスは、16 進数値を 16 ビット単位でコロンで区切って指定します。</li> <li>• <i>Prefix-length</i> は IPv6 プレフィックスの長さです。プレフィックス (アドレスのネットワーク部分) を構成するアドレスの上位連続ビット数を示す 10 進値です。10 進数値の前にスラッシュ記号が必要です。</li> </ul>
ステップ 7	<b>ipv6 link-local <i>ipv6-link-local-address</i></b> 例 : <pre>apic1(config-leaf-if)# ipv6 link-local fe80::1</pre>	インターフェイスに IPv6 リンクローカルアドレスを設定します。
ステップ 8	<b>mac-address <i>mac-address</i></b> 例 : <pre>apic1(config-leaf-if)# mac-address 00:44:55:66:55::01</pre>	インターフェイス MAC アドレスを手動で設定します。

	コマンドまたはアクション	目的
ステップ 9	<b>mtu</b> <i>mtu-value</i> 例： apicl(config-leaf-if)# <b>mtu 1500</b>	このサービス クラスの MTU を設定します
ステップ 10	exit 例： apicl(config-leaf-if)# <b>exit</b>	設定モードに戻ります。
ステップ 11	<b>interface port-channel</b> <i>channel-name</i> 例： apicl(config-leaf)# <b>interface port-channel po1</b>	指定したポート チャネルのインターフェイス コンフィギュレーションモードを開始します。
ステップ 12	<b>vlan-domain member</b> <i>vlan-domain-name</i> 例： apicl(config-leaf-if)# <b>vlan-domain member dom1</b>	以前に設定された VLAN ドメインには、ポートチャネルのテンプレートを関連付けます。
ステップ 13	exit 例： apicl(config-leaf-if)# <b>exit</b>	設定モードに戻ります。
ステップ 14	<b>interface port-channel</b> <i>channel-name.number</i> 例： apicl(config-leaf)# <b>interface port-channel po1.2001</b>	指定したサブインターフェイスポートチャネルのインターフェイス設定モードを開始します。
ステップ 15	<b>vrf member</b> <i>vrf-name tenant tenant-name</i> 例： apicl(config-leaf-if)# <b>vrf member v1 tenant t1</b>	この仮想ルーティングおよび転送(VRF)インスタンスと L3 アウトサイドポリシーにポートチャネルを関連付けます。場所： <ul style="list-style-type: none"><li>• <i>Vrf-name</i> は VRF 名です。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。</li><li>• テナント名は、テナント名です。32 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。</li></ul>
ステップ 16	exit 例：	設定モードに戻ります。

## NX-OS CLI を使用したレイヤ 3 ポート チャンネルにポートを追加する

	コマンドまたはアクション	目的
	apicl(config-leaf-if) # <b>exit</b>	

### 例

この例では、基本的なレイヤ 3 サブインターフェイス ポートチャンネルを設定する方法を示します。

```

apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# interface vlan 2001
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vrf member v1 tenant t1
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# ip address 10.1.1.1/24
apicl(config-leaf-if)# ipv6 address 2001::1/64 preferred
apicl(config-leaf-if)# ipv6 link-local fe80::1
apicl(config-leaf-if)# mac-address 00:44:55:66:55::01
apicl(config-leaf-if)# mtu 1500
apicl(config-leaf-if)# exit
apicl(config-leaf)# interface port-channel po1
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# exit
apicl(config-leaf)# interface port-channel po1.2001
apicl(config-leaf-if)# vrf member v1 tenant t1
apicl(config-leaf-if)# exit

```

## NX-OS CLI を使用したレイヤ 3 ポート チャンネルにポートを追加する

この手順では、以前に設定したレイヤ 3 ポート チャンネルにポートを追加します。

### 手順

	コマンドまたはアクション	目的
<b>ステップ 1</b>	<b>configure</b> 例： apicl# <b>configure</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。
<b>ステップ 2</b>	<b>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	リーフ スイッチまたはリーフ スイッチの設定を指定します。 <i>Node-id</i> は形式 <i>node-id1-node-id2</i> の単一ノード ID または ID の範囲となる可能性があり、設定が適用されます。

	コマンドまたはアクション	目的
ステップ3	<b>interface Ethernet slot/port</b> 例： apic1(config-leaf)# <b>interface Ethernet 1/1-2</b>	設定するインターフェイスのインターフェイス コンフィギュレーション モードを開始します。
ステップ4	<b>channel-group</b> チャンネル名 例： apic1(config-leaf-if)# <b>channel-group p01</b>	チャンネル グループでポートを設定します。

### 例

この例では、ポートをレイヤ3にポートチャンネルを追加する方法を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface Ethernet 1/1-2
apic1(config-leaf-if)# channel-group p01
```

## レイヤ3アウトからレイヤ3アウト内部VRFへの漏洩

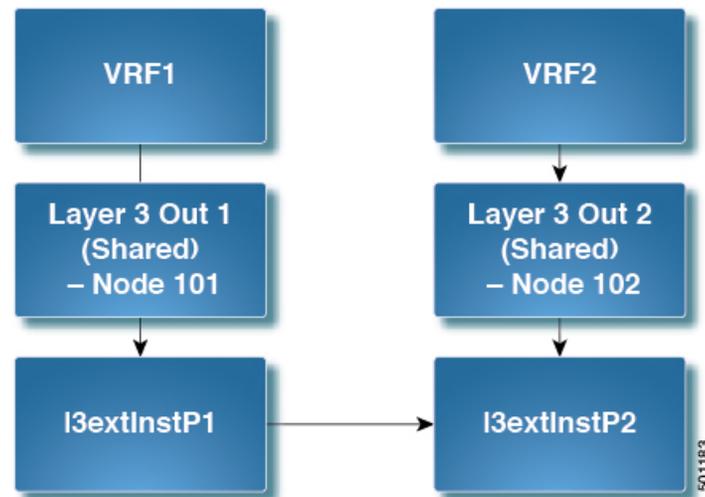
Cisco APIC リリース 2.2(2e) から、2つの異なる VRF に2個のレイヤ3アウトがある場合、VRF 内部の漏洩がサポートされています。

この機能を稼働するには、次の条件を満たす必要があります。

- 2個のレイヤ3アウト間にはコントラクトが必要です。
- レイヤ3アウトの接続したり移行したりするサブネットのルートは、コントラクトを適用し（L3Out-L3OutおよびL3Out-EPG）、VRF間の動的または静的ルートを漏洩させることなく漏洩します。
- 動的または静的ルートは、コントラクトを適用し（L3Out-L3OutおよびL3Out-EPG）、VRF間で直接接続したり移行したりするルートをアドバタイズすることなく漏洩します。
- 異なるVRFの共有のレイヤ3アウトは相互に通信できます。
- 2個のレイヤ3アウトは異なる2個のVRFに存在し、正常にルートを交換できます。
- この強化は、アプリケーション EPG およびレイヤ3アウト内部VRF間の通信と同じです。唯一の違いは、アプリケーション EPGではなく別のレイヤ3アウトが存在します。したがってこの状況では、コントラクトは2個のレイヤ3アウト間で記録されます。

次の図では、共有サブネットによる2個のレイヤ3アウトが存在します。両方のVRFでレイヤ3外部インスタンスプロファイル (l3extInstP) 間のコントラクトがあります。この場合、VRF1の共有レイヤ3アウトはVRF2の共有レイヤ3と通信できます。

図4:2個のVRF間で通信する共有レイヤ3アウト



## NX-OS スタイル CLI を使用して共有 レイヤ3 VRF 内リークを設定する - 名前が付けられた例

### 手順

	コマンドまたはアクション	目的
ステップ1	コンフィギュレーションモードを開始します。  例： <pre>apicl# configure</pre>	
ステップ2	プロバイダーレイヤ3を設定します。  例： <pre>apicl(config)# tenant t1_provider apicl(config-tenant)# external-l3 epg l3extInstP-1 l3out T0-o1-L3OUT-1 apicl(config-tenant-l3ext-epg)# vrf member VRF1 apicl(config-tenant-l3ext-epg)# match ip 192.168.2.0/24 shared apicl(config-tenant-l3ext-epg)# contract provider vzBrCP-1 apicl(config-tenant-l3ext-epg)# exit apicl(config-tenant)# exit apicl(config)# leaf 101 apicl(config-leaf)# vrf context tenant</pre>	

	コマンドまたはアクション	目的
	<pre> t1_provider vrf VRF1 l3out T0-o1-L3OUT-1 apic1(config-leaf-vrf)# route-map T0-o1-L3OUT-1_shared apic1(config-leaf-vrf-route-map)# ip prefix-list l3extInstP-1 permit 192.168.2.0/24 apic1(config-leaf-vrf-route-map)# match prefix-list l3extInstP-1 apic1(config-leaf-vrf-route-map-match)# exit apic1(config-leaf-vrf-route-map)# exit apic1(config-leaf-vrf)# exit apic1(config-leaf)# exit </pre>	
ステップ3	<p>レイヤ3 Out コンシューマを設定します。</p> <p>例：</p> <pre> apic1(config)# tenant t1_consumer apic1(config-tenant)# external-l3 epg l3extInstP-2 l3out T0-o1-L3OUT-1 apic1(config-tenant-l3ext-epg)# vrf member VRF2 apic1(config-tenant-l3ext-epg)# match ip 199.16.2.0/24 shared apic1(config-tenant-l3ext-epg)# contract consumer vzBrCP-1 imported apic1(config-tenant-l3ext-epg)# exit apic1(config-tenant)# exit apic1(config)# leaf 101 apic1(config-leaf)# vrf context tenant t1_consumer vrf VRF2 l3out T0-o1-L3OUT-1 apic1(config-leaf-vrf)# route-map T0-o1-L3OUT-1_shared apic1(config-leaf-vrf-route-map)# ip prefix-list l3extInstP-2 permit 199.16.2.0/24 apic1(config-leaf-vrf-route-map)# match prefix-list l3extInstP-2 apic1(config-leaf-vrf-route-map-match)# exit apic1(config-leaf-vrf-route-map)# exit apic1(config-leaf-vrf)# exit apic1(config-leaf)# exit apic1(config)# </pre>	

## NX-OS Style CLI を使用した共有レイヤ 3 VRF 間リークの設定 : 名前を付けた例

### 手順

	コマンドまたはアクション	目的
ステップ 1	<p>コンフィギュレーション モードを開始します。</p> <p>例 :</p> <pre>apicl# <b>configure</b></pre>	
ステップ 2	<p>プロバイダテナントおよび VRF の設定</p> <p>例 :</p> <pre>apicl (config)# <b>tenant t1_provider</b> apicl (config-tenant)# <b>vrf context VRF1</b> apicl (config-tenant-vrf)# <b>exit</b> apicl (config-tenant)# <b>exit</b></pre>	
ステップ 3	<p>コンシューマテナントおよび VRF の設定</p> <p>例 :</p> <pre>apicl (config)# <b>tenant t1_consumer</b> apicl (config-tenant)# <b>vrf context VRF2</b> apicl (config-tenant-vrf)# <b>exit</b> apicl (config-tenant)# <b>exit</b></pre>	
ステップ 4	<p>コントラクトの設定</p> <p>例 :</p> <pre>apicl (config)# <b>tenant t1_provider</b> apicl (config-tenant)# <b>contract vzBrCP-1</b> <b>type permit</b> apicl (config-tenant-contract)# <b>scope</b> <b>exportable</b> apicl (config-tenant-contract)# <b>export</b> <b>to tenant t1_consumer</b> apicl (config-tenant-contract)# <b>exit</b></pre>	
ステップ 5	<p>プロバイダ外部レイヤ 3 EPG の設定</p> <p>例 :</p> <pre>apicl (config-tenant)# <b>external-l3 ep</b> <b>l3extInstP-1</b> apicl (config-tenant-l3ext-epg)# <b>vrf</b> <b>member VRF1</b> apicl (config-tenant-l3ext-epg)# <b>match</b> <b>ip 192.168.2.0/24 shared</b> apicl (config-tenant-l3ext-epg)# <b>contract provider vzBrCP-1</b> apicl (config-tenant-l3ext-epg)# <b>exit</b> apicl (config-tenant)# <b>exit</b></pre>	

	コマンドまたはアクション	目的
ステップ6	<p>プロバイダ エクスポート マップの設定</p> <p>例 :</p> <pre> apic1(config)# leaf 101 apic1(config-leaf)# vrf context tenant t1_provider vrf VRF1 apic1(config-leaf-vrf)# route-map map1 apic1(config-leaf-vrf-route-map)# ip prefix-list p1 permit 192.168.2.0/24 apic1(config-leaf-vrf-route-map)# match prefix-list p1 apic1(config-leaf-vrf-route-map-match)# exit apic1(config-leaf-vrf-route-map)# exit apic1(config-leaf-vrf)# export map map1 apic1(config-leaf-vrf)# exit apic1(config-leaf)# exit </pre>	
ステップ7	<p>コンシューマ外部レイヤ3 EPG の設定</p> <p>例 :</p> <pre> apic1(config)# tenant t1_consumer apic1(config-tenant)# external-l3 epg l3extInstP-2 apic1(config-tenant-l3ext-epg)# vrf member VRF2 apic1(config-tenant-l3ext-epg)# match ip 199.16.2.0/24 shared apic1(config-tenant-l3ext-epg)# contract consumer vzBrCP-1 imported apic1(config-tenant-l3ext-epg)# exit apic1(config-tenant)# exit </pre>	
ステップ8	<p>コンシューマ エクスポート マップの設 定</p> <p>例 :</p> <pre> apic1(config)# leaf 101 apic1(config-leaf)# vrf context tenant t1_consumer vrf VRF2 apic1(config-leaf-vrf)# route-map map2 apic1(config-leaf-vrf-route-map)# ip prefix-list p2 permit 199.16.2.0/24 apic1(config-leaf-vrf-route-map)# match prefix-list p2 apic1(config-leaf-vrf-route-map-match)# exit apic1(config-leaf-vrf-route-map)# exit apic1(config-leaf-vrf)# export map map2 apic1(config-leaf-vrf)# exit apic1(config-leaf)# exit apic1(config)# </pre>	

## SVI 外部カプセル化の範囲について

レイヤ3アウト設定のコンテキストでは、スイッチ仮想インターフェイス (SVI) は ACI リーフスイッチとルータ間に接続性を提供するように設定されます。

デフォルトで単一のレイヤ3アウトが SVI インターフェイスで設定されている場合、VLAN のカプセル化はファブリック内の複数のノードに範囲が及びます。これは、図で示されるように SVI インターフェイスが同じ外部カプセル化 (SVI) を使用する限り、レイヤ3アウト SVI が展開されているファブリックで、ACI ファブリックがすべてのノード上に同じブリッジドメイン (VXLAN VN) を設定するため発生します。

ただし、異なるレイヤ3アウトが展開されている場合、同じ外部カプセル化 (SVI) を使用している場合でも ACI ファブリックは異なるブリッジドメインを使用します。

図 5: ローカル範囲のカプセル化と 1 個のレイヤ3アウト

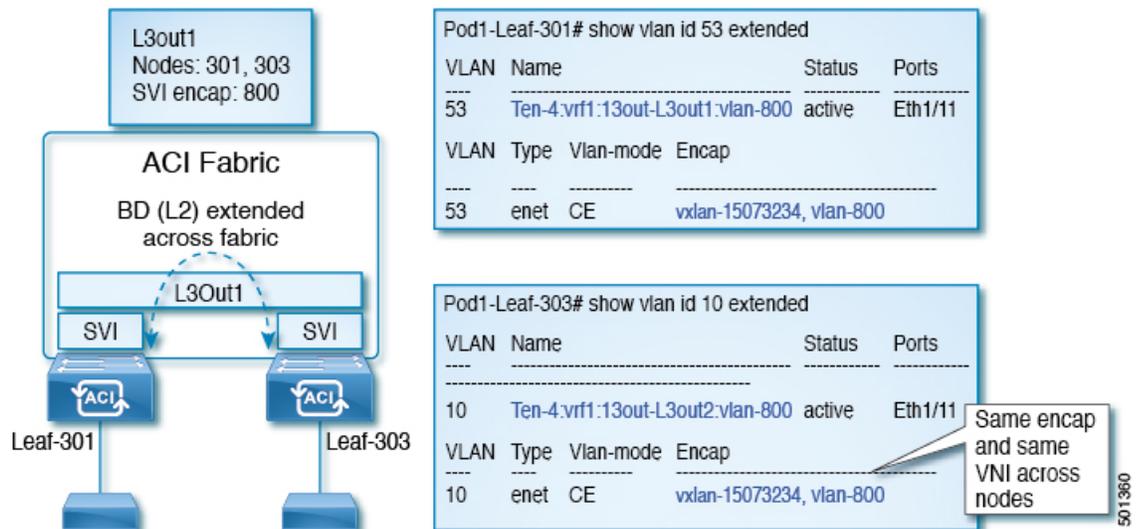
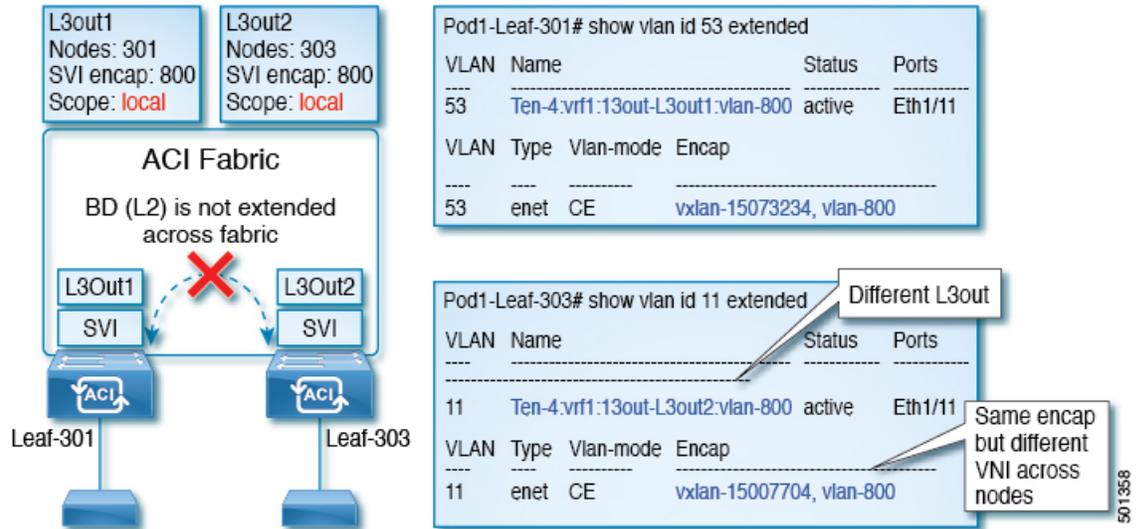


図 6: ローカル範囲のカプセル化と 2 個のレイヤ 3 アウト

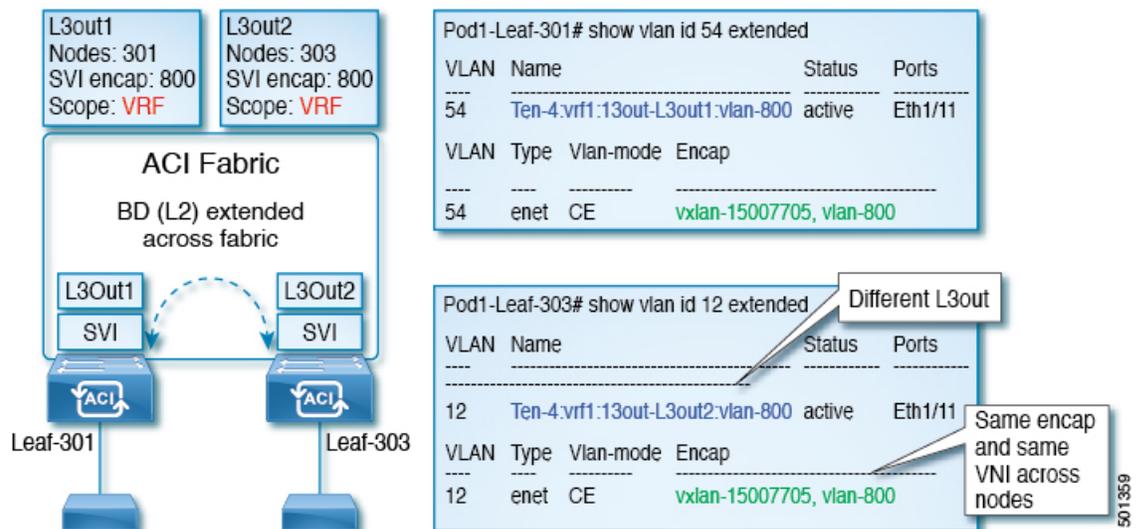


Cisco APIC リリース 2.3 以降、同じ外部カプセル化 (SVI) を使用して、2 個以上のレイヤ 3 アウトを展開する場合の動作を選択できるようになりました。

カプセル化の範囲は、ローカルまたは VRF として設定できます。

- ローカル範囲 (デフォルト) : 例の動作が「ローカル範囲のカプセル化および 2 個のレイヤ 3 アウト」というタイトルの図に表示されます。
- VRF 範囲 : ACI ファブリックが、同じ外部カプセル化 (SVI) が展開されているすべてのノードとレイヤ 3 アウト上で同じブリッジドメイン (VXLAN VNI) を設定します。「VRF 範囲のカプセル化および 2 個のレイヤ 3 アウト」というタイトルの図の例を参照してください。

図 7: VRF 範囲のカプセル化および 2 個のレイヤ 3 アウト



## カプセル化スコープ構文

レイヤ 3 Out プロファイルで使用されるカプセル化の範囲を設定するためのオプションは次のとおりです。

- **Ctx ]**: 特定の VLAN のカプセル化の同じ VRF に、すべてのレイヤ 3 が記録されるで同じ外部 SVI。これはグローバル値です。
- **ローカル** : レイヤ 3 Out ごとの一意の外部 SVI。これはデフォルト値です。

CLI、API、および GUI 構文間のマッピングは次のとおりです。

表 1:カプセル化スコープ構文

CLI	API	GUI
l3out	local	local
vrf	ctx	VRF



(注) カプセル化の範囲を設定する CLI コマンドでは、名前付きのレイヤ 3 アウト設定、VRF が設定されている場合にのみサポートされます。

## NX-OS スタイル CLI を使用して、SVI インターフェイスのカプセル化スコープの設定

SVI インターフェイスカプセル化のスコープ設定を次の例表示する手順では、名前付きのレイヤ 3 アウト設定です。

### 手順

	コマンドまたはアクション	目的
ステップ 1	コンフィギュレーション モードを開始します。 例： <code>apic1# configure</code>	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	スイッチ モードを開始します。 例： <code>apic1(config)# leaf 104</code>	スイッチ モードを開始します。
ステップ 3	VLAN インターフェイスを作成します。 例：	VLAN インターフェイスを作成します。 VLAN の範囲は 1 ~ 4094 です。

	コマンドまたはアクション	目的
	<code>apic1(config-leaf)# interface vlan 2001</code>	
ステップ4	カプセル化の範囲を指定します。 例： <code>apic1(config-leaf-if)# encaps scope vrf context</code>	カプセル化の範囲を指定します。
ステップ5	インターフェイスモードを終了します。 例： <code>apic1(config-leaf-if)# exit</code>	インターフェイスモードを終了します。

## SVI 自動状態について



- (注) この機能は、APIC リリース 2.2(3x) リリースおよび APIC リリース 3.1 (1) で使用できます。APIC リリース 3.0(x) ではサポートされていません。

スイッチ仮想インターフェイス (SVI) は、デバイスの VLAN のブリッジング機能とルーティング機能間の論理インターフェイスを表します。SVIは、物理ポート、直接ポートチャネル、仮想ポートチャネルのメンバーを有することができます。SVI論理インターフェイスはVLANに関連付けられ、VLAN ポートメンバシップを有します。

SVIの状態はメンバーに依存しません。Cisco APICのSVIのデフォルトの自動状態動作は、自動状態の値が無効になっているときに最新の状態になっていることを意味します。これは、インターフェイスが対応するVLANで動作していない場合、SVIがアクティブであることを意味します。

SVI自動状態の値を有効に変更する場合、関連するVLANのポートメンバに依存します。VLANインターフェイスがVLANで複数のポートを有する場合、SVIはVLANのすべてのポートがダウンするとダウン状態になります。

表 2: SVI 自動状態

SVI 自動状態	SVI 状態の説明
ディセーブル	インターフェイスが対応するVLANで動作していない場合、SVIがアップ状態であることを意味します。 無効がデフォルトのSVI自動状態の値です。
イネーブル	SVIは、関連付けられているVLANのポートメンバによって異なります。VLANインターフェイスに複数のポートを含む場合、SVIはVLANのすべてのポートがダウンするとダウン状態になります。

## SVI 自動状態の動作のガイドラインと制限事項

次のガイドラインをお読みください。

- SVI の自動状態の動作を有効化または無効化にすると、SVI あたりの自動状態の動作を設定します。これらはグローバル コマンドではありません。

## NX-OS スタイル CLI を使用した SVI 自動状態の設定

始める前に

- テナントおよび VRF が設定されています。
- レイヤ3アウトが設定されており、レイヤ3アウトの論理ノードプロファイルと論理インターフェイス プロファイルが設定されています。

手順

	コマンドまたはアクション	目的
ステップ 1	コンフィギュレーション モードを開始します。 例： <code>apicl# configure</code>	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	スイッチ モードを開始します。 例： <code>apicl (config)# leaf 104</code>	スイッチ モードを開始します。
ステップ 3	VLAN インターフェイスを作成します。 例： <code>apicl (config-leaf)# interface vlan 2001</code>	VLAN インターフェイスを作成します。 VLAN の範囲は 1 ~ 4094 です。
ステップ 4	SVI 自動状態を有効にします。 例： <code>apicl (config-leaf-if)# autostate</code>	SVI 自動状態を有効にします。 デフォルトで、SVI 自動状態の値は有効ではありません。
ステップ 5	インターフェイスモードを終了します。 例： <code>apicl (config-leaf-if)# exit</code>	インターフェイスモードを終了します。

# インターフェイスとスタティックルーティングの設定

始める前に

テナントおよびVRF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	<b>configure</b> 例： apicl# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ2	<b>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ3	<b>[no] vrf context tenant tenant-name vrf vrf-name</b> 例： apicl(config-leaf)# <b>vrf context tenant exampleCorp vrf v1</b>	ノードのテナントVRFを設定します。
ステップ4	(任意) <b>[no] router-id ipv4-address</b> 例： apicl(config-leaf-vrf)# <b>router-id 1.2.3.4</b>	VRFで実行されているルーティングプロトコルに対してルータIDを割り当てます。ルータIDを割り当てない場合、各リーフスイッチに固有のIDが内部で生成されます。
ステップ5	<b>[no] {ip   ipv6} route ip-prefix/masklen next-hop-address [preferred]</b> 例： apicl(config-leaf-vrf)# <b>ip route 21.1.1.1/32 32.1.1.1</b> apicl(config-leaf-vrf)# <b>ipv6 route 5001::1/128 6002::1</b>	VRFの静的ルートを設定します。
ステップ6	<b>exit</b> 例： apicl(config-leaf-vrf)# <b>exit</b>	リーフ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>interface type</b> 例： apicl(config-leaf)# <b>interface eth 1/1</b>	外部インターフェイスのポートを指定します。

	コマンドまたはアクション	目的
ステップ 8	<b>vlan-domain member <i>domain-name</i></b> 例 : <pre>apic1(config-leaf-if)# vlan-domain member dom1</pre>	インターフェイスに VLAN ドメインを割り当てます。The VLAN domain must have already been created using the <b>vlan-domain</b> command in the global configuration mode.
ステップ 9	<b>no switchport</b> 例 : <pre>apic1(config-leaf-if)# no switchport</pre>	インターフェイスをレイヤ 3 インターフェイスとして設定し、設定オプションでレイヤ 3 コマンドを公開します。
ステップ 10	<b>vrf member tenant <i>tenant-name</i> vrf <i>vrf-name</i></b> 例 : <pre>apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1</pre>	テナント VRF にインターフェイスを接続します。
ステップ 11	<b>[no] {ip   ipv6} address <i>ip-prefix/masklen</i> [eui64] [secondary] [preferred]</b> 例 : <pre>apic1(config-leaf-if)# ip address 10.1.1.1/24 apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred</pre>	インターフェイスに IP アドレスを設定します。指定されたアドレスは次のいずれかとして宣言できます。 <ul style="list-style-type: none"> <li>• <b>preferred</b>—インターフェイスからトラフィックのデフォルト送信元アドレス。</li> <li>• <b>secondary</b>—インターフェイスのセカンダリアドレス。</li> </ul> <p>With the optional <b>eui64</b> keyword, the host can assign itself a 64-bit Extended Unique Identifier (EUI).</p> <p>In this mode, you can also configure <b>ipv6 link-local</b>, <b>mac address</b>, <b>mtu</b>, and other layer 3 properties on the interface.</p>
ステップ 12	<b>[[no]]ip dhcprelayaddress <i>tenant</i><i>tenant-name</i> <i>dhcp-address</i> {<b>application</b><i>app-name</i> <i>epg</i><i>epg-name</i>   <i>external</i><i>extn</i><i>12/2-qg-name</i>   <i>external</i><i>extn</i><i>13/3-qg-name</i>}</b> 例 : <pre>apic(config-leaf-if)# ip dhcp relay address 192.0.20.1 tenant exampleCorp application app1 epg epg1</pre>	サポートされているすべてのオプションとともに、外部インターフェイスの DHCP リレー アドレスを設定または削除します。

## 例

次に、外部接続用にレイヤ3ポートを導入する例を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf v1
apicl(config-leaf-vrf)# router-id 1.2.3.4
apicl(config-leaf-vrf)# ip route 21.1.1.1/32 32.1.1.1
apicl(config-leaf-vrf)# ipv6 route 5001::1/128 6002::1 preferred
apicl(config-leaf-vrf)# exit

apicl(config-leaf)# interface eth 1/1
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vrf member tenant exampleCorp vrf v1
apicl(config-leaf-if)# ip address 10.1.1.1/24
apicl(config-leaf-if)# ip address 11.1.1.1/24 secondary
apicl(config-leaf-if)# ipv6 address 2001::1/64 preferred
apicl(config-leaf-if)# ipv6 link-local fe80::1
apicl(config-leaf-if)# mac-address 00:44:55:66:55::01
apicl(config-leaf-if)# mtu 4470
```

次に、外部接続用にレイヤ3サブインターフェイスポートを設定する例を示します。この例では、サブインターフェイスID (1/2.100の「100」) は実際にはIDではなくVLANのカプセル化です。サブインターフェイスでは、レイヤ3ポートでサポートされるすべてのプロパティを同様に利用できます。

```
apicl# configure
apicl(config)# leaf 101
    # SAME VRF CONTEXT CONFIGURATION AS PREVIOUS EXAMPLE

apicl(config-leaf)# interface eth 1/2.100
apicl(config-leaf-if)# vrf member tenant exampleCorp vrf v1
    # SAME L3 PROPERTIES CONFIGURATION AS PREVIOUS EXAMPLE
```

次に、外部接続用にスイッチ仮想インターフェイス (SVI) を設定する方法を示します。各外部SVIはSVIIDで示されるカプセル化VLANによって一意に識別されます。

```
apicl# configure
apicl(config)# leaf 101
    # SAME VRF CONTEXT CONFIGURATION AS PREVIOUS EXAMPLE

apicl(config-leaf)# interface vlan 200
apicl(config-leaf-if)# vrf member tenant exampleCorp vrf v1
apicl(config-leaf-if)# ip address 13.1.1.1/24

    # HOW TO ATTACH A PORT TO THE EXTERNAL SVI:
apicl(config)# leaf 101
apicl(config-leaf)# interface eth 1/4
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp external-svi

    # HOW TO ATTACH A PORT CHANNEL TO THE EXTERNAL SVI:
apicl(config)# leaf 102
```

```

apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf)# switchport trunk allowed vlan 10 tenant exampleCorp external-svi

# HOW TO ATTACH A VIRTUAL PORT CHANNEL (vPC) TO THE EXTERNAL SVI:
apic1(config)# vpc context leaf 101 102
apic1(config-leaf)# interface vpc vpc103
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf)# switchport trunk allowed vlan 10 tenant exampleCorp external-svi

```



- (注) 外部 SVI は参加ノードごとに設定する必要があります。これにより、同じ SVI でノードごとに異なる IP アドレスを設定することができます。vPC が外部 SVI の一部である場合、参加している vPC ピアごとに SVI を個別に作成する必要があり、SVI ごとに異なる IP アドレスを指定することができます。

## OSPF の設定

### 『Configuring OSPF』

OSPF はエリアで次のいずれかのモードで動作できます。

- OSPF がノードでテナント VRF の主要なルーティングプロトコルとして使用される場合、OSPF は OSPF エリアで設定されたルートマップに定義されているルートをインポートおよびエクスポートします。ルートマップにはエクスポート ルートが含まれます。
- OSPF が BGP の接続プロトコルとして使用される場合、OSPF は BGP セッションの送信元として使用されるループバック アドレスをアドバタイズします。ループバック ID ではなくループバック IP アドレスが使用されることに注意してください。この場合、OSPF に依存している BGP セッションは `update-source` コマンドで同じループバック IP アドレスを使用します。

OSPF と OSPFv3 に個別の設定は必要ありません。ルータ OSPF モードは OSPF で実行されているエリアでは暗黙的に OSPFv2 と OSPFv3 の両方を処理します。

OSPF セッションは、次に示す、リーフのすべてのタイプのレイヤ3 インターフェイスでサポートされます。

- レイヤ3 ポート
- サブインターフェイス
- 外部 SVI

## 手順

	コマンドまたはアクション	目的
ステップ1	<b>configure</b> 例： apicl# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ2	<b>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ3	<b>router ospf default</b> 例： apicl(config-leaf)# <b>router ospf default</b>	OSPFルーティングプロセスを作成し、OSPFポリシー設定を開始します。
ステップ4	<b>vrf member tenant tenant-name vrf vrf-name</b> 例： apicl(config-leaf-ospf)# <b>vrf member tenant exampleCorp vrf v100</b>	OSPFセッションでVRFを有効にします。
ステップ5	(任意) <b>default-information originate [always]</b> 例： apicl(config-leaf-ospf-vrf)# <b>default-information originate</b>	スイッチでデフォルトルートが生成されます。
ステップ6	<b>area area-id nssa [no-redistribution] [default-information-originate]</b> 例： apicl(config-leaf-ospf-vrf)# <b>area 0 nssa</b>	Not-So-Stubby Area (NSSA) を定義します。
ステップ7	<b>area area-id stub</b> 例： apicl(config-leaf-ospf-vrf)# <b>area 17 stub</b>	エリアをスタブエリアとして定義します。
ステップ8	<b>area エリア id default-cost コスト</b> 例： apicl(config-leaf-ospf-vrf)# <b>area 17 default-cost 20</b>	OSPFのデフォルトエリアコストを0～16777215の値に設定します。
ステップ9	<b>area area-id route-map map-name out</b> 例：	アウトバウンドフィルタリングのルートマップを指定します。

	コマンドまたはアクション	目的
	<code>apic1(config-leaf-ospf-vrf)# area 17 route-map ospf-to-eigrp out</code>	
ステップ 10	<b>area area-id loopback loopback-address</b> 例： <code>apic1(config-leaf-ospf-vrf)# area 17 loopback 192.0.20.11/32</code>	OSPF が BGP の接続プロトコルとして使用される場合、OSPF は BGP セッションの送信元として使用されるループバックアドレスをアドバタイズしません。ループバック ID ではなくループバック IP アドレスが使用されることに注意してください。この場合、OSPF に依存している BGP セッションは <code>update-source</code> コマンドで同じループバック IP アドレスを使用します。
ステップ 11	<b>inherit {ipv4   ipv6} ospf vrf-policy policy-name</b> 例： <code>apic1(config-leaf-ospf-vrf)# inherit ipv4 ospf vrf-policy vrfTemplate2</code>	この VRF で OSPF テンプレート ポリシーを継承します。
ステップ 12	<b>summary-address ip-address</b> 例： <code>apic1(config-leaf-ospf-vrf)# summary-address 182.1.20.0/24</code>	外部ルート集約を設定します。他のプロトコルから学習した外部ルートのサマリーアドレスを入力します。
ステップ 13	<b>area area-id range address-range cost cost</b> 例： <code>apic1(config-leaf-ospf-vrf)# area 17 range 192.0.20.0/24 cost 20</code>	エリア間のネットワークを集約するエリア間ルート集約を設定します。  The <b>range</b> is the summary route to be advertised in areas. The <b>cost</b> is a value between 0 and 16777215.
ステップ 14	<b>exit</b> 例： <code>apic1(config-leaf-ospf-vrf)# exit</code>	OSPF コンフィギュレーションモードに戻ります。
ステップ 15	<b>exit</b> 例： <code>apic1(config-leaf-ospf)# exit</code>	リーフ コンフィギュレーションモードに戻ります。
ステップ 16	<b>interface slot/port</b> 例： <code>apic1(config-leaf)# interface eth 1/2</code>	OSPF インターフェイスのポートを指定します。The interface could also be specified as <b>interface slot/port.vlan-id</b> or <b>interface vlanvlan-id</b> .
ステップ 17	<b>{ip   ipv6} router ospf default area area-id</b> 例：	OSPF ルーティングプロセスを作成し、OSPF ポリシー設定を開始します。

	コマンドまたはアクション	目的
	<code>apicl(config-leaf-if)# ip router ospf default area 17</code>	
ステップ 18	<b>{ip   ipv6} ospf inherit interface-policy if-policy-name tenant tenant-name</b> 例： <code>apicl(config-leaf-if)# ip ospf inherit interface-policy ifPolicy3 tenant exampleCorp</code>	このテナントで OSPF インターフェイス テンプレート ポリシーを継承します。
ステップ 19	<b>[no] {ip   ipv6} ospf prefix-suppression {enable   disable   inherit}</b> 例： <code>apicl(config-leaf-if)# ip ospf prefix-suppression enable</code>	セカンダリ IP アドレスに関連付けられているプレフィックスを除き、特定のインターフェイスに属するすべての IP プレフィックスを OSPF がアドバタイズするのを防ぎます。
ステップ 20	<b>[no] {ip   ipv6} ospf passive-interface</b> 例： <code>apicl(config-leaf-if)# ip ospf passive-interface</code>	インターフェイス上でルーティングが更新されないようにします。
ステップ 21	<b>[no] ip ospf authentication {md5   none   simple}</b> 例： <code>apicl(config-leaf-if)# ip ospf authentication md5</code>	認証タイプを指定します。
ステップ 22	<b>ip ospf authentication-key key</b> 例： <code>apicl(config-leaf-if)# ip ospf authentication-key c1\$c0123</code>	認証キーを指定します

## 例

```

apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant exampleCorp vrf v100
apicl(config-leaf-ospf-vrf)# area 0 nssa
apicl(config-leaf-ospf-vrf)# area 17 stub
apicl(config-leaf-ospf-vrf)# area 17 default-cost 20
apicl(config-leaf-ospf-vrf)# area 17 route-map ospf-to-eigrp out
apicl(config-leaf-ospf-vrf)# area 17 loopback 192.0.20.11/32
apicl(config-leaf-ospf-vrf)# inherit ipv4 ospf vrf-policy vrfTemplate2
apicl(config-leaf-ospf-vrf)# summary-address 182.1.20.0/24
apicl(config-leaf-ospf-vrf)# area 17 range 192.0.20.0/24 cost 20
apicl(config-leaf-ospf-vrf)# exit
apicl(config-leaf-ospf)# exit
apicl(config-leaf)# interface eth 1/3

```

```

apic1(config-leaf-if)# ip router ospf default area 17
apic1(config-leaf-if)# ip ospf inherit interface-policy ifPolicy3 tenant exampleCorp
apic1(config-leaf-if)# ip ospf prefix-suppression enable
apic1(config-leaf-if)# ip ospf passive-interface
apic1(config-leaf-if)# ip ospf authentication md5
apic1(config-leaf-if)# ip ospf authentication-key c1$c0123

```

## OSPF VRF とインターフェイス テンプレートの作成

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>template ospf vrf-policy vrf-policy-name</b> <b>tenant tenant-name</b> 例： apic1(config-leaf)# <b>template ospf vrf-policy vrfTemplate3 tenant exampleCorp</b>	指定したテナント下に OSPF VRF ポリシー テンプレートを作成します。
ステップ 4	<b>timers throttle lsa start-time hold-interval max-time</b> 例： apic1(config-vrf-policy)# <b>timers throttle lsa 200 10000 45000</b>	リンクステートアドバタイズメント (LSA) の開始間隔、ホールド間隔、および最大間隔を設定します。
ステップ 5	<b>timers lsa-group-pacing seconds</b> 例： apic1(config-vrf-policy)# <b>timers lsa-group-pacing 240</b>	LSA のグループ化、リフレッシュ、チェックサム、またはエージングを行う間隔を設定します。
ステップ 6	<b>timers lsa-arrival milliseconds</b> 例： apic1(config-vrf-policy)# <b>timers lsa-arrival 1000</b>	各 LSA の到着間の最小間隔を設定します。
ステップ 7	<b>timers throttle spf spf-start spf-hold spf-max-wait</b> 例：	LSA の SPF 開始間隔、ホールド間隔、および最大間隔を設定します。

	コマンドまたはアクション	目的
	<code>apicl(config-vrf-policy)# timers throttle spf 5 1000 90000</code>	
ステップ 8	<b>auto-cost reference-bandwidth <i>bandwidth</i></b> 例： <code>apicl(config-vrf-policy)# auto-cost reference-bandwidth 1000</code>	OSPF ポリシー帯域幅参照を設定します (Mbps)。
ステップ 9	<b>distance <i>distance</i></b> 例： <code>apicl(config-vrf-policy)# distance 200</code>	OSPF ポリシー優先アドミニストレーティブ ディスタンスを設定します。
ステップ 10	<b>maximum-paths <i>max-paths</i></b> 例： <code>apicl(config-vrf-policy)# maximum-paths 8</code>	OSPF がルーティング テーブル内にインストールできるパラレルルートの最大数を設定します。範囲は1～16ルートです。
ステップ 11	<b>graceful-restart helper-disable</b> 例： <code>apicl(config-vrf-policy)# graceful-restart helper-disable</code>	グレースフルリスタートヘルパーモードをディセーブルにします。
ステップ 12	<b>prefix-suppression</b> 例： <code>apicl(config-vrf-policy)# prefix-suppression</code>	ループバック、セカンダリ IP アドレスおよびパッシブインターフェイスに関連付けられているプレフィックスを除き、すべての IP プレフィックスを OSPF がアドバタイズするのを防ぎます。
ステップ 13	<b>name-lookup</b> 例： <code>apicl(config-vrf-policy)# name-lookup</code>	DNS 名をルックアップするように OSPF を設定します。
ステップ 14	<b>exit</b> 例： <code>apicl(config-vrf-policy)# exit</code>	リーフコンフィギュレーションモードに戻ります。
ステップ 15	<b>template ospf interface-policy <i>if-policy-name tenant tenant-name</i></b> 例： <code>apicl(config-leaf)# template ospf interface-policy ifTemplate5 tenant exampleCorp</code>	指定したテナント下に OSPF インターフェイスポリシーテンプレートを作成します。

	コマンドまたはアクション	目的
ステップ 16	<b>[no] advertise-subnet</b> 例： apic1(config-interface-policy)# <b>advertise-subnet</b>	/32 の代わりにプライマリ IP アドレスサブネットマスクをアドバタイズします。
ステップ 17	<b>[no] cost if-cost</b> 例： apic1(config-interface-policy)# <b>cost</b> <b>300</b>	インターフェイスの OSPF コストを設定します。指定できる範囲は 0 ～ 65535 です。
ステップ 18	<b>[no] dead-interval seconds</b> 例： apic1(config-interface-policy)# <b>dead-interval 60</b>	ルータがダウンしたとネイバーが宣言するまでの間、hello パケットを非表示にしなければならない間隔を秒単位で設定します。指定できる範囲は 1 ～ 65535 秒です。
ステップ 19	<b>[no] hello-interval seconds</b> 例： apic1(config-interface-policy)# <b>hello-interval 10</b>	hello パケット間の間隔を秒単位で指定します。指定できる範囲は 1 ～ 65535 秒です。
ステップ 20	<b>[no] mtu-ignore</b> 例： apic1(config-interface-policy)# <b>mtu-ignore</b>	インターフェイスで MTU 不一致の検出を無効にします。
ステップ 21	<b>[no] network {bcast   p2p   unspecified}</b> 例： apic1(config-interface-policy)# <b>network p2p</b>	OSPF インターフェイスポリシーのネットワークタイプをブロードキャストまたはポイントツーポイントのいずれかに設定します。
ステップ 22	<b>[no] passive-interface</b> 例： apic1(config-interface-policy)# <b>passive-interface</b>	インターフェイス上で OSPF ルーティングが更新されないようにします。
ステップ 23	<b>[no] priority priority</b> 例： apic1(config-interface-policy)# <b>priority 4</b>	特定のネットワークの代表ルータ (DR) の決定に使用される OSPF インターフェイスプライオリティを設定します。指定できる範囲は 0 ～ 255 です。
ステップ 24	<b>[no] retransmit-interval seconds</b> 例： apic1(config-interface-policy)# <b>retransmit-interval 5</b>	インターフェイスに属する隣接に対するリンクステートアドバタイズメント (LSA) の再送信間隔を指定します。指定できる範囲は 1 ～ 65535 秒です。

	コマンドまたはアクション	目的
ステップ 25	<b>[no] transmit-delay seconds</b> 例 : apicl(config-interface-policy) # <b>transmit-delay 2</b>	インターフェイス上でリンクステート更新パケットを送信するために必要とされる時間を設定します。範囲は 1 ~ 450 秒です。

## 例

次に、VRFテンプレートおよびインターフェイステンプレートを設定する例を示します。

```

apicl# configure
apicl(config)# leaf 101

      # CONFIGURING THE VRF TEMPLATE:
apicl(config-leaf)# template ospf vrf-policy vrfTemplate3 tenant exampleCorp
apicl(config-vrf-policy)# timers throttle lsa 200 10000 45000
apicl(config-vrf-policy)# timers lsa-group-pacing 240
apicl(config-vrf-policy)# timers lsa-arrival 1000
apicl(config-vrf-policy)# timers throttle spf 5 1000 90000
apicl(config-vrf-policy)# auto-cost reference-bandwidth 1000
apicl(config-vrf-policy)# distance 200
apicl(config-vrf-policy)# maximum-paths 8
apicl(config-vrf-policy)# graceful-restart helper-disable
apicl(config-vrf-policy)# prefix-suppression
apicl(config-vrf-policy)# name-lookup
apicl(config-vrf-policy)# exit

      # CONFIGURING THE INTERFACE TEMPLATE:
apicl(config-leaf)# template ospf interface-policy ifTemplate5 tenant exampleCorp
apicl(config-ospf-if-policy)# advertise-subnet
apicl(config-ospf-if-policy)# cost 300
apicl(config-ospf-if-policy)# dead-interval 60
apicl(config-ospf-if-policy)# hello-interval 10
apicl(config-ospf-if-policy)# mtu-ignore
apicl(config-ospf-if-policy)# network p2p
apicl(config-ospf-if-policy)# passive-interface
apicl(config-ospf-if-policy)# priority 4
apicl(config-ospf-if-policy)# retransmit-interval 5
apicl(config-ospf-if-policy)# transmit-delay 2

```

# BGP の設定

## 『Configuring BGP』

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apicl# <b>configure</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>bgp-fabric</b> 例： apicl(config)# <b>bgp-fabric</b>	ファブリックの BGP 設定モードを開始します。
ステップ 3	<b>asn asn-number</b> 例： apicl(config-bgp-fabric)# <b>asn 100</b>	BGP 自律システム番号 (ASN) を指定します。
ステップ 4	<b>route-reflector spine spine-id</b> 例： apicl(config-bgp-fabric)# <b>route-reflector spine 105</b>	指定したスパインスイッチを BGP ルートリフレクタに設定します。

### 例

```
apicl# configure
apicl(config)# bgp-fabric
apicl(config-bgp-fabric)# asn 100
apicl(config-bgp-fabric)# route-reflector spine 105
```

### 次のタスク

BGP アドレス ファミリとカウンタを設定します。

## BGP アドレス ファミリとタイマー テンプレートの作成

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apicl# <b>configure</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>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>template bgp timers timer-policy-name tenant tenant-name</b> 例： apicl(config-leaf)# <b>template bgp timers bgpTimers tenant exampleCorp</b> This template will be available on all leaves where tenant exampleCorp has a VRF deployment	指定したテナント下に BGP タイマー ポリシーテンプレートを作成します。
ステップ 4	<b>graceful-restart-helper</b>	BGP ポリシーのグレースフルリスタート ヘルパーを設定します。  apicl(config-bgp-timers)# <b>graceful-restart-helper</b>
ステップ 5	<b>graceful-restart stalepath-time seconds</b>	BGP が再起動中の BGP ピアからの古いルートを維持する最大時間を設定します。指定できる範囲は1～3600秒です。  apicl(config-bgp-timers)# <b>graceful-restart stalepath-time 3600</b>
ステップ 6	<b>timers bgp keep-alive-seconds hold-seconds</b>	キープアライブタイマーおよびホールドタイマーの値を設定します。両方の値の範囲は1～3600秒です。  apicl(config-bgp-timers)# <b>timers bgp 10 20</b>
ステップ 7	<b>exit</b>	apicl(config-bgp-timers)# <b>exit</b>
ステップ 8	<b>template bgp address-family family-name tenant tenant-name</b> 例： apicl(config-leaf)# <b>template bgp address-family bgpAfl tenant</b>	指定したテナント下に BGP アドレス ファミリテンプレートを作成します。

	コマンドまたはアクション	目的
	<b>exampleCorp</b> This template will be available on all leaves where tenant exampleCorp has a VRF deployment	
ステップ 9	<b>distance ebgp-distance ibgp-distance local-distance</b>	eBGP ルート、iBGP ルート、およびローカルルートのアドミニストレーティブディスタンスを設定します。指定できる範囲は 1 ~ 255 です。  apic1(config-bgp-af)# <b>distance 250 240 230</b>
ステップ 10	<b>exit</b>	リーフ コンフィギュレーションモードに戻ります。  apic1(config-bgp-af)# <b>exit</b>

### 例

次に、BGP タイマー テンプレートとアドレス ファミリ テンプレートを作成する例を示します。

```
apic1# configure
apic1(config)# leaf 101

# CREATE A TIMER TEMPLATE
apic1(config-leaf)# template bgp timers bgpTimers tenant exampleCorp
This template will be available on all leaves where tenant exampleCorp has a VRF deployment
apic1(config-bgp-timers)# timers bgp 10 20
apic1(config-bgp-timers)# graceful-restart stalepath-time 3600
apic1(config-bgp-timers)# exit

# CREATE AN ADDRESS FAMILY TEMPLATE
apic1(config-leaf)# template bgp address-family bgpAf1 tenant bgp_t1
This template will be available on all leaves where tenant exampleCorp has a VRF deployment
apic1(config-bgp-af)# distance 250 240 230
apic1(config-bgp-af)# exit
apic1(config-leaf)# exit
```

## BGP アドレス ファミリとタイマーの設定

### 始める前に

BGP アドレス ファミリのテンプレートとタイマーのテンプレートを作成します。

## 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーション モードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>router bgp asn-number</b> 例： apic1(config-leaf)# <b>router bgp 100</b>	BGP ポリシー設定を入力します。
ステップ 4	<b>vrf member tenant tenant-name vrf vrf-name</b> 例： apic1(config-bgp)# <b>vrf member tenant exampleCorp vrf v100</b>	後続のアドレス ファミリ コンフィギュレーション モード コマンドと関連付ける VRF インスタンスを指定します。
ステップ 5	<b>inherit bgp timer timer-name</b> 例： apic1(config-leaf-bgp-vrf)# <b>inherit bgp timer bgpTimers</b> This template will be inherited on all leaves where VRF v100 has been deployed	既存のタイマー設定が適用されます。
ステップ 6	<b>address-family {ipv4   ipv6} unicast</b> 例： apic1(config-leaf-bgp-vrf)# <b>address-family ipv4 unicast</b>	正常な IPv4 コニキャスト ルートを交換するネイバーを宣言します。
ステップ 7	<b>inherit bgp address-family family-name</b> 例： apic1(config-leaf-bgp-vrf-af)# <b>inherit bgp address-family ipv4-af-pol</b> This template will be inherited on all leaves where VRF v100 has been deployed	このアドレス ファミリに指定したアドレス ファミリを追加します。
ステップ 8	<b>exit</b> 例： apic1(config-leaf-bgp-vrf-af)# <b>exit</b>	

## 例

この例では、BGP タイマー設定、IPv4 および IPv6 アドレス ファミリを継承する方法を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant exampleCorp vrf v100
apic1(config-leaf-bgp-vrf)# inherit bgp timer bgpTimers
This template will be inherited on all leaves where VRF v100 has been deployed
apic1(config-leaf-bgp-vrf)# address-family ipv4 unicast
apic1(config-leaf-bgp-vrf-af)# inherit bgp address-family ipv4-af-pol
This template will be inherited on all leaves where VRF v100 has been deployed
apic1(config-leaf-bgp-vrf-af)# exit
apic1(config-leaf-bgp-vrf)# address-family ipv6 unicast
apic1(config-leaf-bgp-vrf-af)# inherit bgp address-family ipv6-af-pol
This template will be inherited on all leaves where VRF v100 has been deployed
apic1(config-leaf-bgp-vrf-af)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf)# exit

```

## BGP ネイバーの設定

## 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>router bgp asn-number</b> 例： apic1(config-leaf)# <b>router bgp 100</b>	BGP ポリシー設定を入力します。
ステップ 4	<b>vrf member tenant tenant-name vrf vrf-name</b> 例： apic1(config-bgp)# <b>vrf member tenant exampleCorp vrf v100</b>	後続のポリシー設定モードコマンドと関連付ける VRF インスタンスを指定します。

	コマンドまたはアクション	目的
ステップ 5	<p>(任意) <b>aggregate-address</b> <i>ip-address/masklength</i> [<b>as-set</b>]</p> <p>例 :</p> <pre>apicl (config-leaf-bgp-vrf) # aggregate-address 192.0.10.0/24 as-set</pre>	<p>アドレスの範囲のサマリーアドレスを設定し、BGP データベースに集約エントリを作成します。アドレスは IPv4 または IPv6 を使用できます。The <b>as-set</b> option generates autonomous system set path information.</p>
ステップ 6	<p><b>neighbor</b> <i>ip-address</i>[/<i>masklength</i>]</p> <p>例 :</p> <pre>apicl (config-leaf-bgp-vrf) # neighbor 192.0.2.229/32</pre>	<p>ネイバーの IP アドレスを指定します。</p>
ステップ 7	<p><b>address-family</b> {<i>ipv4</i>   <i>ipv6</i>} <b>unicast</b></p> <p>例 :</p> <pre>apicl (config-leaf-bgp-vrf-neighbor) # address-family ipv4 unicast</pre>	<p>正常な IPv4 ユニキャストルートを交換するネイバーを宣言します。</p>
ステップ 8	<p>[<b>no</b>] <b>maximum-prefix</b> <i>count</i> [<b>action</b> {<b>log</b>   <b>shut</b>   <b>restart</b> [<b>restart-time</b> <i>minutes</i>]}] [<b>threshold</b> <i>percent</i>]</p> <p>例 :</p> <pre>apicl (config-leaf-bgp-vrf-neighbor-af) # maximum-prefix 10 threshold 10 action restart restart-time 10</pre>	<p>このネイバーからのプレフィックスの最大数を設定します。範囲は 1 ~ 300000 プレフィックスです。その他のオプションの設定は次のとおりです。</p> <ul style="list-style-type: none"> <li>• <b>action</b> : 最大プレフィックスの制限に達したときに実行するアクションです。If the <b>action</b> is <b>restart</b>, you can optionally specify the <b>restart-time</b>, which is the period of time in minutes before restarting the peer when the maximum prefix limit is reached. 指定できる範囲は 1 ~ 65535 分です。</li> <li>• <b>threshold</b> : 警告が発行されるまでのプレフィックスの最大数のしきい値の割合。指定できる範囲は 1 ~ 100% です。</li> </ul>
ステップ 9	<p><b>exit</b></p> <p>例 :</p> <pre>apicl (config-leaf-bgp-vrf-neighbor-af) # exit</pre>	
ステップ 10	<p><b>update-source</b> {<i>loopback ip-address</i>   <i>ethernet ip-address</i>   <i>vlan vlan-id</i>}</p> <p>例 :</p> <pre>apicl (config-leaf-bgp-vrf-neighbor) # update-source loopback 192.0.2.230</pre>	<p>ネイバー アドレスが OSPF を通じて学習される場合、OSPF で使用されるのと同じループバックアドレスを指定します。</p>

	コマンドまたはアクション	目的
ステップ 11	<b>weight number</b> 例 : <pre>apic1(config-leaf-bgp-vrf-neighbor) # weight 2000</pre>	最適なパスを選択するために、重み属性を指定します。重みは、0 ~ 65,535 で指定できます。同じ宛先に対して複数のルートがある場合、より高い重み値のルートが優先されます。
ステップ 12	<b>private-as-control {remove-exclusive   remove-exclusive-all   remove-exclusive-all-replace-as}</b> 例 : <pre>apic1(config-leaf-bgp-vrf-neighbor) # private-as-control</pre>	自律システムパスからプライベート自律システム番号を削除します。次の 3 つの可能なバリエーションに従って、プライベート AS 番号をピアごとの AS パスから削除し、eBGP ピアのみで使用することができます。 <ul style="list-style-type: none"> <li>• <b>remove-exclusive</b> : AS パスがプライベート AS 番号のみ場合、削除します。</li> <li>• <b>remove-exclusive-all</b> : AS パスがプライベートおよびパブリック AS 番号の両方の場合、削除します。</li> <li>• <b>remove-exclusive-all-replace-as</b> : ルータのローカル AS 番号とプライベート AS 番号を置き換えます。</li> </ul> このコマンドは例として表示されています。この時点で、次の表に示すネイバー設定のいずれかを設定できます。

次の表に、この時点で設定できるインターフェイス設定を示します。

コマンド	目的
allow-self-as	as-path を、その中に自身の AS を含めて受け入れます
allowed-self-as-count <i>count</i>	ローカルアクセスサービスネットワークの発生件数を指定します
disable-connected-check	直接接続されたピアのチェックをディセーブルにします
disable-peer-as-check	アドバタイジング中のピア AS 番号のチェックをディセーブルにします
ebgp-multihop <i>count</i>	リモートピアにマルチホップ TTL を指定します

コマンド	目的
<code>local-as <i>asn</i></code>	BGP ピアのローカルの自律システム設定をします
<code>next-hop-self</code>	ピアリングアドレスをネクストホップとして設定します
<code>password <i>password</i></code>	ネイバーのパスワードを設定します
<code>private-as-control</code>	AS パスからプライベート ASN を削除します
<code>remote-as <i>asn</i></code>	ネイバーの自律システム番号を指定します
<code>route-map <i>name</i> {in   out}</code>	ネイバーにルートマップ適用します
<code>send-community [extended]</code>	コミュニティ属性をこのネイバーに送信します
<code>update-source vlan <i>vlan-id</i></code>	送信元 VLAN インターフェイスを指定します
<code>update-source ethernet <i>slot/port</i></code>	送信元イーサネット インターフェイスを指定します
<code>update-source loopback <i>ip-address</i></code>	送信元ルックバック インターフェイスを指定します
<code>weight <i>number</i></code>	ルーティングテーブルの BGP 重みを指定します

## 例

この例では、IPv4 BGP ネイバーを設定する方法を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# router bgp 100
apicl(config-bgp)# vrf member tenant exampleCorp vrf v100
apicl(config-leaf-bgp-vrf)# aggregate-address 192.0.10.0/24 as-set
apicl(config-leaf-bgp-vrf)# neighbor 192.0.2.229/32
apicl(config-leaf-bgp-vrf-neighbor)# address-family ipv4 unicast
apicl(config-leaf-bgp-vrf-neighbor-af)# maximum-prefix 10 threshold 10 action restart
restart-time 10
apicl(config-leaf-bgp-vrf-neighbor-af)# exit
apicl(config-leaf-bgp-vrf-neighbor)# allow-self-as
apicl(config-leaf-bgp-vrf-neighbor)# allowed-self-as-count 2
apicl(config-leaf-bgp-vrf-neighbor)# disable-connected-check
apicl(config-leaf-bgp-vrf-neighbor)# disable-peer-as-check
apicl(config-leaf-bgp-vrf-neighbor)# ebgp-multihop 4
apicl(config-leaf-bgp-vrf-neighbor)# local-as 100
apicl(config-leaf-bgp-vrf-neighbor)# next-hop-self
apicl(config-leaf-bgp-vrf-neighbor)# password abcdef
apicl(config-leaf-bgp-vrf-neighbor)# remote-as 200
```

```

apic1(config-leaf-bgp-vrf-neighbor)# send-community extended
apic1(config-leaf-bgp-vrf-neighbor)# update-source vlan 601
apic1(config-leaf-bgp-vrf-neighbor)# update-source ethernet 1/15
apic1(config-leaf-bgp-vrf-neighbor)# update-source loopback 192.0.2.230
Warning: BGP Configuration changed. Please re-configure BGP Password if it was enabled
apic1(config-leaf-bgp-vrf-neighbor)# local-as 100 no-prepend replace-as dual-as
apic1(config-leaf-bgp-vrf-neighbor)# route-map rMapT3 out
apic1(config-leaf-bgp-vrf-neighbor)# weight 2000
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf)# exit

```

この例では、IPv6 BGP ネイバーを設定する方法を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant exampleCorp vrf v100
apic1(config-leaf-bgp-vrf)# neighbor 2001:80:1:2::229
apic1(config-leaf-bgp-vrf-neighbor)# address-family ipv6 unicast
apic1(config-leaf-bgp-vrf-neighbor-af)# maximum-prefix 100
apic1(config-leaf-bgp-vrf-neighbor-af)# exit
apic1(config-leaf-bgp-vrf-neighbor)# allow-self-as
apic1(config-leaf-bgp-vrf-neighbor)# allowed-self-as-count 2
apic1(config-leaf-bgp-vrf-neighbor)# disable-connected-check
apic1(config-leaf-bgp-vrf-neighbor)# disable-peer-as-check
apic1(config-leaf-bgp-vrf-neighbor)# ebgp-multihop 4
apic1(config-leaf-bgp-vrf-neighbor)# local-as 100
apic1(config-leaf-bgp-vrf-neighbor)# next-hop-self
apic1(config-leaf-bgp-vrf-neighbor)# password abcdef
apic1(config-leaf-bgp-vrf-neighbor)# remote-as 200
apic1(config-leaf-bgp-vrf-neighbor)# send-community extended
apic1(config-leaf-bgp-vrf-neighbor)# update-source vlan 601
apic1(config-leaf-bgp-vrf-neighbor)# update-source ethernet 1/15
apic1(config-leaf-bgp-vrf-neighbor)# update-source loopback 2001:80:1:2::230/128
Warning: BGP Configuration changed. Please re-configure BGP Password if it was enabled
apic1(config-leaf-bgp-vrf-neighbor)# local-as 100 no-prepend replace-as dual-as
apic1(config-leaf-bgp-vrf-neighbor)# route-map rMapT3 out
apic1(config-leaf-bgp-vrf-neighbor)# weight 2000
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf-af)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf)# exit

```

## NX-OS スタイル CLI を使用してノード BGP タイマー ポリシーあたりの VRF あたりを設定する

### 手順

	コマンドまたはアクション	目的
ステップ1	<p>タイマーポリシーを作成する前に、BGP ASN およびルートリフレクタを設定します。</p> <p>例：</p> <pre> apic1(config)# apic1(config)# <b>bgp-fabric</b> apic1(config-bgp-fabric)# <b>route-reflector spine 102</b> apic1(config-bgp-fabric)# <b>asn 42</b> apic1(config-bgp-fabric)# <b>exit</b> apic1(config)# <b>exit</b> apic1# </pre>	
ステップ2	<p>タイマーポリシーを作成します。</p> <p>例：</p> <pre> apic1# <b>config</b> apic1(config)# <b>leaf 101</b> apic1(config-leaf)# <b>template bgp timers</b> <b>pol7 tenant tnl</b> This template will be available on all nodes where tenant tnl has a VRF deployment apic1(config-bgp-timers)# <b>timers bgp</b> <b>120 240</b> apic1(config-bgp-timers)# <b>graceful-restart stalepath-time 500</b> apic1(config-bgp-timers)# <b>maxas-limit</b> <b>300</b> apic1(config-bgp-timers)# <b>exit</b> apic1(config-leaf)# <b>exit</b> apic1(config)# <b>exit</b> apic1# </pre>	特定の値は、例としてのみ提供されます。
ステップ3	<p>設定された BGP ポリシーを表示します。</p> <p>例：</p> <pre> apic1# <b>show run leaf 101 template bgp</b> <b>timers pol7</b> # Command: show running-config leaf 101 template bgp timers pol7 leaf 101 template bgp timers pol7 tenant tnl timers bgp 120 240 graceful-restart stalepath-time </pre>	

	コマンドまたはアクション	目的
	<pre>500 maxas-limit 300 exit exit</pre>	
<b>ステップ4</b>	<p>ノードで特定のポリシーを参照します。</p> <p>例：</p> <pre>apicl# config apicl(config)# leaf 101 apicl(config-leaf)# router bgp 42 apicl(config-leaf-bgp)# vrf member tenant tn1 vrf ctx1 apicl(config-leaf-bgp-vrf)# inherit node-only bgp timer pol7 apicl(config-leaf-bgp-vrf)# exit apicl(config-leaf-bgp)# exit apicl(config-leaf)# exit apicl(config)# exit apicl#</pre>	
<b>ステップ5</b>	<p>特定のBGPのタイマーポリシーのノードが表示されます。</p> <p>例：</p> <pre>apicl# show run leaf 101 router bgp 42 vrf member tenant tn1 vrf ctx1 # Command: show running-config leaf 101 router bgp 42 vrf member tenant tn1 vrf ctx1 leaf 101 router bgp 42 vrf member tenant tn1 vrf ctx1 inherit node-only bgp timer pol7 exit exit exit apicl#</pre>	

## BGP 最長パスの設定

### 始める前に

適切なテナントおよびBGP外部ルーテッドネットワークが作成され使用可能です。

次の機能を使用すると、ルート表にパスの最大数を追加し、コストマルチパスのロードバランシングを有効にできます。

さらに多くのパスを設定できるようにする2つのプロパティは、bgpCtxAfPol オブジェクトの maxEcmp と maxEcmpIbgp です。これら2つのプロパティを設定した後、実装の残り部分に反映されます。

BGP にログインして、次のコマンドを使用します:

```
maximum-paths [ibgp]
no maximum-paths [ibgp]
```

設定例:

手順

例:

```
apicl(config)# leaf 101
apicl(config-leaf)# template bgp address-family newAf tenant t1
This template will be available on all nodes where tenant t1 has a VRF deployment
apicl(config-bgp-af)# maximum-paths ?
<1-16> Maximum number of equal-cost paths for load sharing. The default is 16.
ibgp Configure multipath for IBGP paths
apicl(config-bgp-af)# maximum-paths 10
apicl(config-bgp-af)# maximum-paths ibgp 8
apicl(config-bgp-af)# end
apicl#
no maximum-paths [ibgp]
```

## ASパスプリペンドの設定

BGP ピアは、ASパスアトリビュートの長さを増やすことで、リモートピアでベストパス選択の影響を与えることができます。番号として指定桁の前に付加してASパスアトリビュートの長さを向上するために使用するメカニズムを提供するASパスPrepend。

ASパス前に付加は、ルートマップを使用してアウトバウンド方向にのみ適用できます。パスとして前に付加が機能しないiBGPセッションで。

ASパスPrepend機能は、次のように変更を有効に。

プリペンド	<p>ルートマップと一致するルートのASパスに、指定したAS番号を付加します。</p> <p>(注)</p> <ul style="list-style-type: none"> <li>• 1個以上のAS番号を設定できます。</li> <li>• 4バイト番号がサポートされています。</li> <li>• 合計をprependは32のAS番号。AS番号は、ASパスアトリビュートに挿入されます順序を指定する必要があります。</li> </ul>
Prepend-最後-として	最後の前に付加ASパス1から10までの範囲に番号として。

次の表では、ASパスPrependの実装の選択基準について説明します。

プリペンド	1	指定されたAS番号を追加します。
-------	---	------------------

Prepend-最後-として	2	最後の AS 番号を AS パスに付加します。
デフォルト	Prepend(1)	指定された AS 番号を追加します。

## NX-OS スタイルの CLI を使用した AS パスのプリペンド

このセクションでは、NX-OS スタイル コマンドライン インターフェイス (CLI) を使用して、AS パスのプリペンド機能を実現する方法について説明します。

### 始める前に

構成済みのテナント

### 手順

境界ゲートウェイ プロトコル (BGP) ルートの自動システムパス (AS パス) を変更するには、`set as-path` コマンドを使用します。`set as-path` コマンドは、`apic1(config-leaf-vrf-template-route-profile)# set as-path {'prepend as-num [ ,... as-num ] | prepend-last-as num}` の形式で実行します。

#### 例 :

```
apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# template route-profile rp1
apic1(config-leaf-vrf-template-route-profile)# set as-path ?
prepend Prepend to the AS-Path
prepend-last-as Prepend last AS to the as-path
apic1(config-leaf-vrf-template-route-profile)# set as-path prepend 100, 101, 102, 103
apic1(config-leaf-vrf-template-route-profile)# set as-path prepend-last-as 8
apic1(config-leaf-vrf-template-route-profile)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
```

### 次のタスク

AS パスのプリペンドを無効にするには、示されているコマンドの `no` 形式を使用します:

```
apic1(config-leaf-vrf-template-route-profile)# [no] set
as-path { prepend as-num [ ,... as-num ] | prepend-last-as num}
```

## BGP にディストリビューションをルーティング

### テナントの範囲によるルート プロファイルの設定

#### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</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>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>template route-profile profile-name tenant tenant-name</b> 例： apic1(config-leaf)# <b>template route-profile map_eigrp tenant exampleCorp</b>	BGP ダンプニングとルート再配布のテナントの下にルート プロファイル テンプレートを作成します。
ステップ 4	必須: <b>[no] set tag name</b> 例： apic1(config-leaf-template-route-profile)# <b>set tag 200</b>	タグ値を設定します。name パラメータは符号なし整数です。
ステップ 5	必須: <b>exit</b> 例： apic1(config-leaf-template-route-profile)# <b>exit</b>	リーフ コンフィギュレーション モードに戻ります。
ステップ 6	<b>template route-profile profile-name tenant tenant-name</b> 例： apic1(config-leaf)# <b>template route-profile map_ospf tenant exampleCorp</b>	BGP ダンプニングとルート再配布のテナントの下にルート プロファイル テンプレートを作成します。
ステップ 7	必須: <b>[no] set tag name</b> 例： apic1(config-leaf-template-route-profile)# <b>set tag 100</b>	タグ値を設定します。name パラメータは符号なし整数です。

## 例

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# template route-profile map_eigrp tenant exampleCorp
apic1(config-leaf-template-route-profile)# set tag 200
apic1(config-leaf-template-route-profile)# exit
apic1(config-leaf)# template route-profile map_ospf tenant exampleCorp
apic1(config-leaf-template-route-profile)# set tag 100
apic1(config-leaf-template-route-profile)# exit

```

## 次のタスク

この手順で作成されたルートプロファイルのいずれかを使用して、OSPFおよびEIGRPのBGPの下に再配布ルートプロファイルを設定します。

## 再配布ルート プロファイルの設定

## 始める前に

ルート再配布のテナントの下にルートプロファイルテンプレートを作成します。

## 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</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>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>router bgp asn-number</b> 例： apic1(config-leaf)# <b>router bgp 100</b>	BGP ポリシー設定を入力します。
ステップ 4	<b>vrf member tenant tenant-name vrf vrf-name</b> 例： apic1(config-bgp)# <b>vrf member tenant exampleCorp vrf v100</b>	後続のポリシー設定モードコマンドと関連付ける VRF インスタンスを指定します。
ステップ 5	必須: <b>redistribute {ospf eigrp} route-map map-name</b>	

	コマンドまたはアクション	目的
	例 : <pre>apicl(config-leaf-bgp-vrf)# redistribute ospf route-map map_ospf</pre>	

### 例

この例では、テナントの範囲によるルートプロファイルの作成の例で作成したルートプロファイルを使用して、OSPF および EIGRP の BGP の下に再配布ルートプロファイルを設定します。再配布ルートマップは、すべてのルートを可能（許可）し、ルート制御アクションに関するルートプロファイルを適用します。この例では、すべての EIGRP 学習ルートは、タグ 200 を使用して BGP に再配布され、OSPF ルートは、タグ 100 を使用して BGP に再配布されます。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# router bgp 100
apicl(config-bgp)# vrf member tenant exampleCorp vrf v1
apicl(config-leaf-bgp-vrf)# redistribute eigrp route-map map_eigrp
apicl(config-leaf-bgp-vrf)# redistribute ospf route-map map_ospf
```

## BGP ルート ダンプニングの設定

BGP ルート ダンプニングでは、境界リーフ スイッチ (BLs) に接続されている外部ルータから受信したフラッピング eBGP ルートのファブリックへの伝搬を最小限に抑えます。外部ルータから頻繁にフラッピングするルートは設定された条件に基づいて BLs で抑制され、iBGP ピア (ACI スパインスイッチ) への再配布が禁止されます。抑制されたルートは設定された時間条件の後で再使用されます。各フラップは eBGP ルートに 1000 のペナルティを科します。フラップのペナルティが定義された抑制限度のしきい値 (デフォルトは 2000) に達すると、eBGP ルートは抑制としてマーキングされます。抑制されたルートは他の BGP ピアにはアドバタイズされません。ペナルティは、半減期 (デフォルトは 15 分) ごとに半分に減少します。抑制されたルートは、ペナルティが指定された再利用の限度 (デフォルトは 750) を下回ると再利用されます。抑制されたルートは、指定された最大抑制時間 (最大 45 分間) の間最大限に抑制されます。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例 : <pre>apicl# configure</pre>	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
ステップ 2	<b>leaf node-id</b> 例 : <pre>apic1(config)# leaf 101</pre>	設定するリーフを指定します。
ステップ 3	<b>template route-profile profile-name tenant tenant-name</b> 例 : <pre>apic1(config-leaf)# template route-profile damp_rp tenant exampleCorp</pre>	BGP ダンプニングとルート再配布のテナントの下にルートプロファイルテンプレートを作成します。
ステップ 4	必須: <b>[no] set dampening half-life reuse suppress max-suppress-time</b> 例 : <pre>apic1(config-leaf-template-route-profile)# set dampening 15 750 2000 60</pre>	ルートフラップダンプニングの動作を設定します。パラメータは次のとおりです。 <ul style="list-style-type: none"> <li>• <i>half-life</i> : 時間経過後にペナルティが減少する分単位の半減期。ルートにペナルティを割り当てると、ペナルティは半減期で半分まで小さくなります。指定できる範囲は 1 ~ 60 分で、デフォルト値は 15 分です。</li> <li>• <i>reuse</i> : フラッピングルートのペナルティがこの値を下回るまで減少すると、ルートの抑制が中止（再利用）されます。指定できる値の範囲は 1 ~ 20000 です。デフォルトは 750 です。</li> <li>• <i>suppress</i> : ペナルティがこの制限を超えるとルートは抑制されます。指定できる値の範囲は 1 ~ 20000 です。デフォルトは 2000 です。</li> <li>• <i>max-suppress-time</i> : 安定したルートを抑制できる分単位の最大時間。指定できる範囲は 1 ~ 255 です。</li> </ul>
ステップ 5	必須: <b>exit</b> 例 : <pre>apic1(config-leaf-template-route-profile)# exit</pre>	リーフコンフィギュレーションモードに戻ります。
ステップ 6	<b>router bgp asn-number</b> 例 :	BGP ポリシー設定を入力します。

	コマンドまたはアクション	目的
	<code>apicl(config-leaf)# router bgp 100</code>	
ステップ 7	<b>vrf member tenant</b> <i>tenant-name</i> <b>vrf</b> <i>vrf-name</i> 例： <code>apicl(config-bgp)# vrf member tenant exampleCorp vrf v100</code>	後続のポリシー設定モードコマンドと関連付ける VRF インスタンスを指定します。
ステップ 8	<b>neighbor</b> <i>ip-address[/masklength]</i> 例： <code>apicl(config-leaf-bgp-vrf)# neighbor 192.0.2.229/32</code>	ネイバーの IP アドレスを指定します。マスク長は 32 です。
ステップ 9	<b>address-family</b> { <i>ipv4</i>   <i>ipv6</i> } <b>unicast</b> 例： <code>apicl(config-leaf-bgp-vrf-neighbor)# address-family ipv4 unicast</code>	正常な IPv4 ユニキャストルートを交換するネイバーを宣言します。
ステップ 10	<b>inherit bgp dampening</b> <i>profile-name</i> 例： <code>apicl(config-leaf-bgp-vrf-neighbor-af)# inherit bgp dampening damp_rp</code>	
ステップ 11	<b>exit</b> 例： <code>apicl(config-leaf-bgp-vrf-neighbor-af)# exit</code>	
ステップ 12	<b>exit</b> 例： <code>apicl(config-leaf-bgp-vrf-neighbor)# exit</code>	
ステップ 13	<b>address-family</b> { <i>ipv4</i>   <i>ipv6</i> } <b>unicast</b> 例： <code>apicl(config-leaf-bgp-vrf)# address-family ipv4 unicast</code>	正常な IPv4 ユニキャストルートを交換するネイバーを宣言します。
ステップ 14	<b>inherit bgp dampening</b> <i>profile-name</i> 例： <code>apicl(config-leaf-bgp-vrf-af)# inherit bgp dampening damp_rp</code>	
ステップ 15	<b>exit</b> 例： <code>apicl(config-leaf-bgp-vrf-af)# exit</code>	

## 例

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# template route-profile damp_rp tenant exampleCorp
apic1(config-leaf-template-route-profile)# set dampening 15 750 2000 60
apic1(config-leaf-template-route-profile)# exit
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant exampleCorp vrf v100
apic1(config-leaf-bgp-vrf)# neighbor 192.0.2.229/32
apic1(config-leaf-bgp-vrf-neighbor)# address-family ipv4 unicast
apic1(config-leaf-bgp-vrf-neighbor-af)# inherit bgp dampening damp_rp
apic1(config-leaf-bgp-vrf-neighbor-af)# exit
apic1(config-leaf-bgp-vrf)# address-family ipv6 unicast
apic1(config-leaf-bgp-vrf-af)# inherit bgp dampening damp_rp
apic1(config-leaf-bgp-vrf-af)# exit

```

## EIGRP の設定

### EIGRP VRF とインターフェイス テンプレートの作成

## 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</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>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>template eigrp vrf-policy vrf-policy-name</b> <b>tenant tenant-name</b> 例： apic1(config-leaf)# <b>template eigrp vrf-policy vrfTemplate3 tenant exampleCorp</b> This template will be available on all leaves where tenant exampleCorp has a VRF deployment	指定したテナント下に EIGRP VRF ポリシー テンプレートを作成します。
ステップ 4	<b>distance internal external</b> 例：	EIGRP 内部ルートと外部ルートの管理 デスタンスプリファレンスを設定し

	コマンドまたはアクション	目的
	<code>apicl(config-template-eigrp-vrf-pol)# distance 2 5</code>	ます。ディスタンスは 1 ~ 255 で指定 できます。
ステップ 5	<b>maximum-paths limit</b> 例 : <code>apicl(config-template-eigrp-vrf-pol)# maximum-paths 8</code>	VRF ポリシー テンプレートの EIGRP の最大パスの制限を設定します。制限 は 1 ~ 32 で指定できます。
ステップ 6	<b>metric version 64bit</b> 例 : <code>apicl(config-template-eigrp-vrf-pol)# metric version 64bit</code>	広範なメトリック (64 ビット) に EIGRP メトリック形式を設定します。
ステップ 7	<b>timers active-time minutes</b> 例 : <code>apicl(config-template-eigrp-vrf-pol)# timers active-time 1</code>	EIGRP のアクティブ タイマーの間隔を 設定します。指定できる範囲は 1 ~ 65535 分です。
ステップ 8	<b>template eigrp interface-policy if-policy-name tenant tenant-name</b> 例 : <code>apicl(config-leaf)# template eigrp interface-policy ifTemplate5 tenant exampleCorp</code> This template will be available on all leaves where tenant exampleCorp has a VRF deployment	指定したテナント下に EIGRP インター フェイスポリシーテンプレートを作成 します。
ステップ 9	<b>ip hello-interval eigrp default seconds</b> 例 : <code>apicl(config-template-eigrp-if-pol)# ip hello-interval eigrp default 10</code>	EIGRP Hello の間隔時間を設定します。 指定できる範囲は 1 ~ 65535 秒です。
ステップ 10	<b>ip hold-interval eigrp default seconds</b> 例 : <code>apicl(config-template-eigrp-if-pol)# ip hold-interval eigrp default 10</code>	EIGRP ホールドの間隔時間を設定しま す。指定できる範囲は 1 ~ 65535 秒で す。
ステップ 11	<b>ip next-hop-self eigrp default</b> 例 : <code>apicl(config-template-eigrp-if-pol)# ip next-hop-self eigrp default</code>	EIGRP の next-hop-self フラグを設定し ます。
ステップ 12	<b>ip passive-interface eigrp default</b> 例 : <code>apicl(config-template-eigrp-if-pol)# ip passive-interface eigrp default</code>	EIGRP の passive-interface フラグを設定 します。

	コマンドまたはアクション	目的
ステップ 13	<b>ip split-horizon eigrp default</b> 例 : apic1(config-template-eigrp-if-pol)# <b>ip split-horizon eigrp default</b>	EIGRP の split-horizon フラグを設定します。
ステップ 14	<b>exit</b> 例 : apic1(config-template-eigrp-if-pol)# <b>exit</b>	リーフ コンフィギュレーションモードに戻ります。

### 例

```

apic1# configure
apic1(config)# leaf 101

      # CONFIGURING THE VRF TEMPLATE:
apic1(config-leaf)# template eigrp vrf-policy vrfTemplate3 tenant exampleCorp
This template will be available on all leaves where tenant exampleCorp has a VRF deployment
apic1(config-template-eigrp-vrf-pol)# distance 2 5
apic1(config-template-eigrp-vrf-pol)# maximum-paths 8
apic1(config-template-eigrp-vrf-pol)# metric version 64bit
apic1(config-template-eigrp-vrf-pol)# timers active-time 1
apic1(config-template-eigrp-vrf-pol)# exit

      # CONFIGURING THE INTERFACE TEMPLATE:
apic1(config-leaf)# template eigrp interface-policy ifTemplate5 tenant exampleCorp
This template will be available on all leaves where tenant exampleCorp has a VRF deployment
apic1(config-template-eigrp-if-pol)# ip hello-interval eigrp default 5
apic1(config-template-eigrp-if-pol)# ip hold-interval eigrp default 10
apic1(config-template-eigrp-if-pol)# ip next-hop-self eigrp default
apic1(config-template-eigrp-if-pol)# ip passive-interface eigrp default
apic1(config-template-eigrp-if-pol)# ip split-horizon eigrp default
apic1(config-template-eigrp-if-pol)# exit
apic1(config-leaf)# exit
apic1(config)# exit

```

### 次のタスク

EIGRP アドレス ファミリとカウンタを設定します。

## EIGRP アドレス ファミリとカウンタの設定

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apicl# <b>configure</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>leaf node-id</b> 例： apicl (config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>router eigrp default</b> 例： apicl (config-leaf)# <b>router eigrp default</b>	EIGRP ポリシー設定を入力します。
ステップ 4	<b>vrf member tenant tenant-name vrf vrf-name</b> 例： apicl (config-eigrp)# <b>vrf member tenant exampleCorp vrf v100</b>	後続のアドレスファミリ コンフィギュレーションモードコマンドと関連付ける VRF インスタンスを指定します。
ステップ 5	<b>autonomous-system asn</b> 例： apicl (config-eigrp-vrf)# <b>autonomous-system 300</b>	EIGRP の自律システム設定を入力します。
ステップ 6	<b>address-family {ipv4   ipv6} unicast</b> 例： apicl (config-eigrp-vrf)# <b>address-family ipv4 unicast</b>	EIGRP ポリシー アドレス ファミリを設定します。
ステップ 7	<b>distance internal external</b> 例： apicl (config-address-family)# <b>distance 2 5</b>	EIGRP 内部ルートと外部ルートの管理ディスタンスプリファレンスを設定します。ディスタンスは 1 ~ 255 で指定できます。
ステップ 8	<b>maximum-paths limit</b> 例： apicl (config-address-family)# <b>maximum-paths 8</b>	VRF ポリシー テンプレートの EIGRP の最大パスの制限を設定します。制限は 1 ~ 32 で指定できます。

	コマンドまたはアクション	目的
ステップ 9	<b>metric version 64bit</b> 例： apic1(config-address-family)# <b>metric version 64bit</b>	広範なメトリック (64 ビット) に EIGRP メトリック形式を設定します。
ステップ 10	<b>timers active-time minutes</b> 例： apic1(config-address-family)# <b>timers active-time 1</b>	EIGRP のアクティブタイマーの間隔を設定します。指定できる範囲は 1 ～ 65535 分です。
ステップ 11	<b>inherit eigrp vrf-policy vrf-policy-name</b> 例： apic1(config-address-family)# <b>inherit eigrp vrf-policy vrfTemplate3</b>	このアドレス ファミリに EIGRP VRF ポリシーを適用します。

### 例

この例では、EIGRP アドレス ファミリを設定する方法および EIGRP VRF ポリシーを継承する方法を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router eigrp default
apic1(config-eigrp)# vrf member tenant exampleCorp vrf v100
apic1(config-eigrp-vrf)# autonomous-system 300
apic1(config-eigrp-vrf)# address-family ipv4 unicast
This configuration will affect all leaves where VRF v100 has been deployed
apic1(config-address-family)# distance 2 5
This configuration will affect all leaves where VRF v100 has been deployed
apic1(config-address-family)# maximum-paths 8
This configuration will affect all leaves where VRF v100 has been deployed
apic1(config-address-family)# metric version 64bit
This configuration will affect all leaves where VRF v100 has been deployed
apic1(config-address-family)# timers active-time 1
This configuration will affect all leaves where VRF v100 has been deployed
apic1(config-address-family)# inherit eigrp vrf-policy vrfTemplate3
This template will be inherited on all leaves where VRF v100 has been deployed
apic1(config-address-family)# exit
apic1(config-eigrp-vrf)# exit
apic1(config-eigrp)# exit

```

## EIGRP 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>configure</b> 例： apicl# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>interface ethernet slot/port</b> 例： apicl(config-leaf)# <b>interface ethernet 1/21</b>	設定するインターフェイスを指定します。
ステップ 4	<b>[no] switchport slot/port</b> 例： apicl(config-leaf-if)# <b>no switchport</b>	デフォルトでは、ポートはレイヤ2 トランク モードです。ポートがレイヤ3 モードである場合、このコマンドを使用して、レイヤ2 トランク モードに変換する必要があります。
ステップ 5	<b>[no] vlan-domain member vlan-id</b> 例： apicl(config-leaf-if)# <b>vlan-domain member dom1</b>	VLAN ドメインのコンフィギュレーション モードを作成および開始します。
ステップ 6	<b>[no] vrf member tenant exampleCorp vrf vrf-name</b> 例： apicl(config-leaf-if)# <b>vrf member tenant exampleCorp vrf v100</b>	インターフェイスを VRF に関連付けます。
ステップ 7	<b>[no] {ip   ipv6} address ip-address/mask-length</b> 例： apicl(config-leaf-if)# <b>ip address 181.12.12.1/24</b>	インターフェイスの IP アドレスを設定します。
ステップ 8	<b>[no] {ip   ipv6} router eigrp default</b> 例： apicl(config-leaf-if)# <b>ip router eigrp default</b>	EIGRP ルータのポリシーをデフォルトに設定します。

	コマンドまたはアクション	目的
ステップ 9	<p><b>[no] {ip   ipv6} distribute-list eigrp default route-map map-name out</b></p> <p>例 :</p> <pre>apic1(config-leaf-if)# ip distribute-list eigrp default route-map rMapT5 out</pre>	<p>EIGRP は、distribute-list コマンドで指定されたルートマップで一致するルートをアドバタイズします。ルートマップのプレフィックスリストに記載されているルーティングプレフィックスは、BGP、OSPF、Static、Connected などの他のプロトコルソースから学習できます。再配布ルートマップは、distribute-list コマンドに基づいて自動的に作成されます。同じスイッチ上の別のインターフェイス上で実行されている EIGRP セッションから学んだプレフィックスが、配布リストによってフィルタリングされず、常に外にアドバタイズされることに注意してください。</p>
ステップ 10	<p><b>[no] {ip   ipv6} hello-interval eigrp default seconds</b></p> <p>例 :</p> <pre>apic1(config-leaf-if)# ip hello-interval eigrp default 10</pre>	<p>EIGRP Hello の間隔時間を設定します。指定できる範囲は 1 ~ 65535 秒です。</p>
ステップ 11	<p><b>[no] {ip   ipv6} hold-interval eigrp default seconds</b></p> <p>例 :</p> <pre>apic1(config-leaf-if)# ip hold-interval eigrp default 10</pre>	<p>EIGRP ホールドの間隔時間を設定します。指定できる範囲は 1 ~ 65535 秒です。</p>
ステップ 12	<p><b>[no] {ip   ipv6} next-hop-self eigrp default</b></p> <p>例 :</p> <pre>apic1(config-leaf-if)# ip next-hop-self eigrp default</pre>	<p>EIGRP の next-hop-self フラグを設定します。</p>
ステップ 13	<p><b>[no] {ip   ipv6} passive-interface eigrp default</b></p> <p>例 :</p> <pre>apic1(config-leaf-if)# ip passive-interface eigrp default</pre>	<p>EIGRP の passive-interface フラグを設定します。</p>
ステップ 14	<p><b>[no] {ip   ipv6} split-horizon eigrp default</b></p> <p>例 :</p> <pre>apic1(config-leaf-if)# ip split-horizon eigrp default</pre>	<p>EIGRP の split-horizon フラグを設定します。</p>

	コマンドまたはアクション	目的
ステップ 15	<p><b>[no] inherit eigrp ip interface-policy if-policy-name</b></p> <p>例 :</p> <pre>apicl(config-leaf-if)# inherit eigrp ip interface-policy ifTemplate5</pre>	このインターフェイスに EIGRP インターフェイスポリシーを適用します。
ステップ 16	<p><b>[no] ip summary-address eigrp default ip-prefix</b></p> <p>例 :</p> <pre>apicl(config-leaf-if)# ip summary-address eigrp default 172.10.1.0/24</pre> <pre>apicl(config-leaf-if)# ip summary-address eigrp default 2001::/64</pre>	<p>EIGRP のルート集約を設定します。サマリーアドレスは、EIGRP セッション上に集約されたプレフィックスをアドバタイズするように設定できます。</p> <p>(注) 1つのインターフェイスでネーブルになっているサマリーアドレスは、スイッチ上の同じ VRF 上の他の EIGRP 対応インターフェイスに適用されます。</p>

## 例

この例では、EIGRP インターフェイスを設定する方法を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/21
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# vrf member tenant exampleCorp vrf v100
apicl(config-leaf-if)# ip address 181.12.12.1/24
apicl(config-leaf-if)# ip router eigrp default
apicl(config-leaf-if)# ip distribute-list eigrp default route-map rMapT5 out
distribute list will be updated on all EIGRP interfaces on node 1021 VRF exampleCorp/v100
apicl(config-leaf-if)# ip hello-interval eigrp default 5
apicl(config-leaf-if)# ip hold-interval eigrp default 10
apicl(config-leaf-if)# ip next-hop-self eigrp default
apicl(config-leaf-if)# ip passive-interface eigrp default
apicl(config-leaf-if)# ip split-horizon eigrp default
apicl(config-leaf-if)# inherit eigrp ip interface-policy ifTemplate5
apicl(config-leaf-if)# ip summary-address eigrp default 172.10.1.0/24
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
apicl(config)# exit
```

# ルートマップの設定

## テンプレートの設定

### ルート プロファイルについて

ルートプロファイルは、インポート、エクスポートに使用されるルーティング制御設定のアクションを指定し、ルートマップを再配布します。ルートプロファイルテンプレートは、テナントまたはテナント VRF で定義できます。

### テナントスコープルート プロファイルの設定

この手順では、BGP ダンプニングとルート再配布の設定に使用されるテナントスコープのルートプロファイルを作成します。

#### 始める前に

- テナントおよび VRF を設定します。
- リーフで VRF を有効にします。

#### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>[no] template route-profile profile-name tenant tenant-name</b> 例： apic1(config-leaf)# <b>template route-profile rp1 tenant exampleCorp</b>	テナントスコープのルートプロファイルを作成します。
ステップ 4	必須: <b>[no] set community {regular  extended} 値 {none  replace  additive}</b> 例： apic1(config-leaf-template-route-profile)# <b>set community extended 20:22 additive</b>	BGP コミュニティ属性を設定します。

	コマンドまたはアクション	目的
ステップ5	<p>必須: <b>[no] set dampening half-life reuse suppress max-suppress-time</b></p> <p>例:</p> <pre>apic1(config-leaf-template-route-profile)# set dampening 15 750 2000 60</pre>	<p>ルートフラップダンプニングの動作を設定します。パラメータは次のとおりです。</p> <ul style="list-style-type: none"> <li>• <b>half-life</b>: 時間経過後にペナルティが減少する分単位の半減期。ルートにペナルティを割り当てると、ペナルティは半減期で半分まで小さくなります。指定できる範囲は1～60分です。</li> <li>• <b>reuse</b>: フラッピングルートのペナルティがこの値を下回るまで減少すると、ルートの抑制が中止（再利用）されます。指定できる範囲は1～20000です。</li> <li>• <b>suppress</b>: ペナルティがこの制限を超えるとルートは抑制されます。指定できる範囲は1～20000です。</li> <li>• <b>max-suppress-time</b>: 安定したルートを抑制できる分単位の最大時間。指定できる範囲は1～255です。</li> </ul>
ステップ6	<p>必須: <b>[no] set local-preference value</b></p> <p>例:</p> <pre>apic1(config-leaf-template-route-profile)# set local-preference 64</pre>	<p>BGP ローカルプリファレンス値を設定します。範囲は0～4294967295です。</p>
ステップ7	<p>必須: <b>[no] set metric value</b></p> <p>例:</p> <pre>apic1(config-leaf-template-route-profile)# set metric 128</pre>	<p>宛先ルーティングプロトコルのメトリックを設定します。</p>
ステップ8	<p>必須: <b>[no] set metric-type {type-1   type-2}</b></p> <p>例:</p> <pre>apic1(config-leaf-template-route-profile)# set metric-type type-2</pre>	<p>オプションは次のとおりです。</p> <ul style="list-style-type: none"> <li>• <b>type-1</b>: OSPF 外部タイプ1メトリック</li> <li>• <b>type-2</b>: OSPF 外部タイプ2メトリック</li> </ul>

	コマンドまたはアクション	目的
ステップ 9	必須: <b>[no] set tag name</b> 例: <pre>apic1(config-leaf-template-route-profile)# set tag 1111</pre>	宛先ルーティングプロトコルのタグ値を設定します。 <i>name</i> パラメータは符号なし整数です。
ステップ 10	必須: <b>[no] set weight weight</b> 例: <pre>apic1(config-leaf-template-route-profile)# set weight 20</pre>	宛先ルーティングプロトコルのタグ値を設定します。 <i>weight</i> パラメータは符号なし整数です。

### 例

次に、テナントスコープのルートプロファイルを設定する例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# template route-profile rp1 tenant exampleCorp
This template will be available on all leaves where tenant exampleCorp has a VRF deployment
apic1(config-leaf-template-route-profile)# set community extended 20:22 additive
apic1(config-leaf-template-route-profile)# set dampening 15 750 2000 60
apic1(config-leaf-template-route-profile)# set local-preference 64
apic1(config-leaf-template-route-profile)# set metric 128
apic1(config-leaf-template-route-profile)# set metric-type type-2
apic1(config-leaf-template-route-profile)# set tag 1111
apic1(config-leaf-template-route-profile)# set weight 20
```

## VRF スコープ ルート プロファイルの設定

この手順では、「default-export」および「default-import」を含む VRF スコープのルートプロファイルを作成します。This route profile can be attached to a bridge domain (BD) while 'matching' a bridge-domain inside a route map through the **inherit route-profile** command.



(注) VRF-scoped route profiles name **default-export** and **default-import** values, which are automatically applied on the match statements on the respective export/import route-maps used in the same tenant VRF.

### 始める前に

- テナントおよび VRF を設定します。
- リーフで VRF を有効にします。

## 手順

	コマンドまたはアクション	目的
ステップ1	<b>configure</b> 例： apicl# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ2	<b>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ3	<b>[no] vrf context tenant tenant-name vrf vrf-name</b> 例： apicl(config-leaf)# <b>vrf context tenant exampleCorp vrf vrf1</b>	リーフで VRF を有効にし、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>[no] template route-profile profile-name</b> 例： apicl(config-leaf-vrf)# <b>template route-profile default-export</b>	VRF スコープのルートプロファイルを作成します。
ステップ5	必須: <b>[no] set community {regular  extended} {no-advertise no-export 値 {none  replace  additive}}</b> 例： apicl(config-leaf-vrf-template-route-profile)# <b>set community extended 20:22 additive</b>	BGP コミュニティ属性を設定します。
ステップ6	必須: <b>[no] set local-preference value</b> 例： apicl(config-tenant-vrf-route-profile)# <b>set local-preference 64</b>	BGP ローカルプリファレンス値を設定します。範囲は0～4294967295です。
ステップ7	必須: <b>[no] set metric value</b> 例： apicl(config-tenant-vrf-route-profile)# <b>set metric 128</b>	宛先ルーティングプロトコルのメトリックを設定します。
ステップ8	必須: <b>[no] set metric-type {type-1   type-2}</b> 例： apicl(config-tenant-vrf-route-profile)# <b>set metric-type type-2</b>	オプションは次のとおりです。  <ul style="list-style-type: none"> <li>• <b>type-1</b> : OSPF 外部タイプ1メトリック</li> <li>• <b>type-2</b> : OSPF 外部タイプ2メトリック</li> </ul>

	コマンドまたはアクション	目的
ステップ 9	必須: <b>[no] set tag name</b> 例: <pre>apicl(config-tenant-vrf-route-profile)# set tag 1111</pre>	宛先ルーティングプロトコルのタグ値を設定します。 <i>name</i> パラメータは符号なし整数です。
ステップ 10	必須: <b>[no] set weight weight</b> 例: <pre>apicl(config-tenant-vrf-route-profile)# set weight 20</pre>	宛先ルーティングプロトコルのタグ値を設定します。 <i>weight</i> パラメータは符号なし整数です。

### 例

次に、VRF スコープのルートプロファイルを設定する例を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf vrf1
apicl(config-leaf-vrf)# template route-profile default-export
apicl(config-leaf-vrf-template-route-profile)# set community extended 20:22 additive
apicl(config-leaf-vrf-template-route-profile)# set local-preference 64
apicl(config-leaf-vrf-template-route-profile)# set metric 128
apicl(config-leaf-vrf-template-route-profile)# set metric-type type-2
apicl(config-leaf-vrf-template-route-profile)# set tag 1111
apicl(config-leaf-vrf-template-route-profile)# set weight 20
```

## ルートマップの作成

ルートマップは、外部ルータにアドバタイズされるブリッジドメインのパブリックサブネットを示すため、テナント単位のプレフィックスリストとともに作成されます。また、プレフィックスリストはすべての中継ルートが外部ルータにアドバタイズされるように作成する必要があります。中継ルートのプレフィックスリストは管理者によって設定されます。デフォルトの動作では、外部ルータへの中継ルートのアドバタイズメントはすべて拒否されます。

### 始める前に

テナントおよび VRF 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	<b>configure</b> 例: <pre>apicl# configure</pre>	コンフィギュレーションモードに入ります。

	コマンドまたはアクション	目的
ステップ 2	<b>leaf node-id</b> 例： apicl(config)# leaf 101	設定するリーフを指定します。
ステップ 3	<b>[no] vrf context tenant tenant-name vrf vrf-name</b> 例： apicl(config-leaf)# vrf context tenant exampleCorp vrf v1	ノードのテナント VRF を設定します。
ステップ 4	(任意) <b>[no] router-id ipv4-address</b> 例： apicl(config-leaf-vrf)# router-id 1.2.3.4	VRF で実行されているルーティングプロトコルに対してルータ ID を割り当てます。ルータ ID を割り当てない場合、各リーフ スイッチに固有の ID が内部で生成されます。
ステップ 5	必須: <b>[no] route-map name</b> 例： apicl(config-leaf-vrf)# route-map bgpMap	ルートマップを作成し、ルートマップ コンフィギュレーションを開始します。
ステップ 6	必須: <b>[no] ip permit le 32</b> リスト名プレフィックス/マスク長[ <b>{ prefix-list128}</b> ] 例： apicl(config-leaf-vrf-route-map)# ip prefix-list list1 permit 13.13.13.0/24	ルートマップにプレフィックスリストを作成します。
ステップ 7	必須: <b>[no] match prefix-list list-name</b> 例： apicl(config-leaf-vrf-route-map)# match prefix-list list1	作成済みのプレフィックスリストと照合し、一致モードを開始して、プレフィックスリストのルート制御プロファイルを設定します。
ステップ 8	必須: <b>[no] set metric value</b> 例： apicl(config-leaf-vrf-route-map-match)# set metric 128	宛先ルーティングプロトコルのメトリックを設定します。
ステップ 9	必須: <b>[no] set metric-type {type-1   type-2}</b> 例： apicl(config-leaf-vrf-route-map-match)# set metric-type type-2	オプションは次のとおりです。  <ul style="list-style-type: none"> <li>• <b>type-1</b> : OSPF 外部タイプ 1 メトリック</li> <li>• <b>type-2</b> : OSPF 外部タイプ 2 メトリック</li> </ul>

	コマンドまたはアクション	目的
ステップ 10	必須: <b>[no] set local-preference value</b> 例: <pre>apic1(config-leaf-vrf-route-map-match)# set local-preference 64</pre>	BGP ローカルプリファレンス値を設定します。範囲は0～4294967295です。
ステップ 11	必須: <b>[no] set community {regular   extended} value {none   replace   additive}</b> 例: <pre>apic1(config-leaf-vrf-route-map-match)# set community extended 20:22 additive</pre>	BGP ルートアップデートのコミュニティ属性を設定します。 <i>community-value</i> を <i>aa:nn</i> 形式で指定します。次のいずれかのアクションを指定します。 <ul style="list-style-type: none"> <li>• <b>additive</b>—既存のコミュニティに追加します</li> <li>• <b>replace</b>—既存のコミュニティを置き換えます</li> <li>• <b>none</b>—コミュニティを変更しません</li> </ul>
ステップ 12	必須: <b>[no] set tag name</b> 例: <pre>apic1(config-leaf-vrf-route-map-match)# set tag 1111</pre>	宛先ルーティングプロトコルのタグ値を設定します。 <i>name</i> パラメータは符号なし整数です。
ステップ 13	必須: <b>[no] set weight value</b> 例: <pre>apic1(config-leaf-vrf-route-map-match)# set weight 32</pre>	ルーティングプロトコルの BGP 重みを指定します。
ステップ 14	必須: <b>[no] contract {provider   consumer} contract-name [imported]</b> 例: <pre>apic1(config-leaf-vrf-route-map-match)# contract provider prov 1</pre>	この VRF から（このプレフィックスリストと一致する）ルートをリークするために必要なコントラクトを追加します。
ステップ 15	必須: <b>[no] match route group group-name [order number]</b> 例: <pre>apic1(config-leaf-vrf-route-map)# match route group g1 order 1</pre>	すでに作成されたルートグループを一致させ、一致モードを開始してルートマップを設定します。 手順 8-13 または手順 18 のみを繰り返して、ルートグループのルートマップを設定します。手順 17 を参照して、インラインセットアクションの代わりにルートマップを継承します。

	コマンドまたはアクション	目的
ステップ 16	必須: <b>[no] match bridge-domain <i>bd-name</i></b> 例: <pre>apicl(config-leaf-vrf-route-map) # bridge-domain bd1</pre>	プロトコルによってパブリック サブネットをエクスポートするため、ブリッジドメインと照合します。
ステップ 17	必須: <b>[no] inherit route-profile <i>profile-name</i></b> 例: <pre>apicl(config-leaf-vrf-route-map-match) # inherit route-profile rpl</pre>	ブリッジドメインのルートマップを設定します。  (注) ルートマップはすでに <b>template route-profile</b> コマンドを使用してすでに作成されています。
ステップ 18	必須: <b>[no] bridge-domain-match</b> 例: <pre>apicl(config-leaf-vrf-route-map) # no bridge-domain-match</pre>	ブリッジドメインのルートマップを設定します。  (注) ルートマップでブリッジドメイン (BD) 一致を無効にして、ルートマップから BD 設定を削除する必要性をなくします。これは、BD がルートマップで一致して、ルートマップがルートグループ/明示プレフィックスリストを使用して BD サブネットをフィルタするために使用される場合に必要です。

## 例

次に、ルートマップを作成し、プレフィックスリスト、コミュニティリスト、およびブリッジドメインの追加/一致を行う例を示します。

```
# CREATE A ROUTE-MAP
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf v1
apicl(config-leaf-vrf)# route-map bgpMap

# CREATE A PREFIX-LIST
apicl(config-leaf-vrf-route-map)# ip prefix-list list1 permit 13.13.13.0/24
apicl(config-leaf-vrf-route-map)# ip prefix-list list1 permit 14.14.14.0/24

# MATCH THE PREFIX-LIST
apicl(config-leaf-vrf-route-map)# match prefix-list list1

# CONFIGURE A ROUTE-PROFILE FOR THE PREFIX-LIST
```

```

apic1(config-leaf-vrf-route-map-match)# set metric 128
apic1(config-leaf-vrf-route-map-match)# set metric-type type-2
apic1(config-leaf-vrf-route-map-match)# set local-preference 64
apic1(config-leaf-vrf-route-map-match)# set community extended 20:22 additive
apic1(config-leaf-vrf-route-map-match)# set tag 1111
apic1(config-leaf-vrf-route-map-match)# set weight 32
apic1(config-leaf-vrf-route-map-match)# contract provider prov 1

# CREATE COMMUNITY LIST
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# template community-list standard CL_1 65536:20 tenant exampleCorp

# CREATE ROUTE GROUP
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# template route group g1 tenant exampleCorp
apic1(config-route-group)# ip prefix permit 15.15.15.0/24
apic1(config-route-group)# community-list standard 65535:20

# MATCH ROUTE GROUP
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf)# route-map bgpMap
apic1(config-leaf-vrf-route-map)# match route group g1 order 1

# CONFIGURE ROUTE PROFILE FOR COMMUNITY-LIST
apic1(config-leaf-vrf-route-map-match)# set metric 128
apic1(config-leaf-vrf-route-map-match)# set metric-type type-2
apic1(config-leaf-vrf-route-map-match)# set local-preference 64
apic1(config-leaf-vrf-route-map-match)# set community extended 20:22 additive
apic1(config-leaf-vrf-route-map-match)# set tag 1111
apic1(config-leaf-vrf-route-map-match)# set weight 32

# CONFIGURE ROUTE PROFILE ROUTE GROUP
apic1(config-leaf-vrf-route-map-match)# set metric 128
apic1(config-leaf-vrf-route-map-match)# set metric-type type-2
apic1(config-leaf-vrf-route-map-match)# set local-preference 64
apic1(config-leaf-vrf-route-map-match)# set community extended 20:22 additive
apic1(config-leaf-vrf-route-map-match)# set tag 1111
apic1(config-leaf-vrf-route-map-match)# set weight 32

# Or CREATE A ROUTE PROFILE TEMPLATE AND INHERIT IT FOR ROUTE GROUP
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf)# template route-profile rp1
apic1(config-leaf-vrf-template-route-profile)# set metric 128
apic1(config-leaf-vrf-template-route-profile)# set metric-type type-2
apic1(config-leaf-vrf-template-route-profile)# set local-preference 64
apic1(config-leaf-vrf-template-route-profile)# set community extended 20:22 additive
apic1(config-leaf-vrf-template-route-profile)# set tag 1111
apic1(config-leaf-vrf-template-route-profile)# set weight 32
apic1(config-leaf-vrf-template-route-profile)# exit

apic1(config-leaf-vrf)# route-map bgpMap
apic1(config-leaf-vrf-route-map)# match route group g1 order 1
apic1(config-leaf-vrf-route-map-match)# inherit route-profile rp1

# CREATE A BRIDGE-DOMAIN
apic1# configure
apic1(config)# tenant exampleCorp

```

```

apicl(config-tenant)# vrf context v1
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# bridge-domain bd1
apicl(config-tenant-bd)# vrf member v1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# interface bridge-domain bd1
apicl(config-tenant-interface)# ip address 13.13.13.1/24 scope public
apicl(config-tenant-interface)# exit
apicl(config-tenant)# exit

# CREATE A ROUTE-PROFILE FOR THE BRIDGE-DOMAIN
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf v1
apicl(config-leaf-vrf)# template route-profile default-export
apicl(config-leaf-vrf-template-route-profile)# set metric 128
apicl(config-leaf-vrf-template-route-profile)# set metric-type type-2
apicl(config-leaf-vrf-template-route-profile)# set local-preference 64
apicl(config-leaf-vrf-template-route-profile)# set community extended 20:22 additive
apicl(config-leaf-vrf-template-route-profile)# set tag 1111
apicl(config-leaf-vrf-template-route-profile)# set weight 20
apicl(config-leaf-vrf-template-route-profile)# exit

# MATCH THE BRIDGE-DOMAIN
apicl(config-leaf-vrf)# route-map bgpMap
apicl(config-leaf-vrf-route-map)# match bridge-domain bd1

# CONFIGURE A ROUTE-PROFILE FOR THE BRIDGE-DOMAIN
apicl(config-leaf-vrf-route-map-match)# inherit route-profile default-export

```

## ルーティングプロトコルのルートマップの設定

OSPF、BGP、およびEIGRPルーティングプロトコルでは、インポートおよびエクスポートするルートフィルタ処理にルートマップを使用します。これらのプロトコルを設定するために必要な一般的な手順についてはそれぞれのドキュメントセクションを参照してください。これらのプロトコルでルートマップを設定するには、次のコマンドを使用し、例を参照してください。

プロトコル	ルートマップコマンド
BGP	<b>[no] route-map map-name {in   out}</b>
OSPF	<b>[no] area エリア id route-map マップ名 {in   out}</b>
EIGRP	<b>[no] ip distribute list default route-map map-name out</b>

### 例

次に、BGP、OSPF、およびEIGRPでルートマップを設定する例を示します。

```

# BGP
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# router bgp 100

```

```

apic1(config-bgp)# vrf member tenant exampleCorp vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 3.3.3.3
apic1(config-leaf-bgp-vrf-neighbor)# route-map map1 out
apic1(config-leaf-bgp-vrf-neighbor)# route-map map2 in
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-bgp)# exit
apic1(config-leaf)# exit

# OSPF
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant exampleCorp vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.1 route-map map1 out
apic1(config-leaf-ospf-vrf)# area 0.0.0.1 route-map map2 in
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit

#EIGRP
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1
apic1(config-leaf-if)# ip address 13.13.13.13/24
apic1(config-leaf-if)# ip router eigrp default
apic1(config-leaf-if)# ip distribute-list eigrp default route-map map1 out
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit

```

## エクスポートマップ（インターVRF ルートリーク）の設定

### 始める前に

- ルートマップを作成します。
- リークさせる必要があるルートと一致するプレフィックスを含むプレフィックスリストをルートマップに追加します。
- プレフィックスリストを照合し、ルートリークを有効にするコントラクトを追加します。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。

	コマンドまたはアクション	目的
ステップ 2	<b>leaf node-id</b>  例 : apicl(config)# leaf 101	設定するリーフを指定します。
ステップ 3	<b>[no] vrf context tenant tenant-name vrf vrf-name</b>  例 : apicl(config-leaf)# vrf context tenant exampleCorp vrf v1	ノードのテナント VRF を設定します。
ステップ 4	<b>[no] export map map-name</b>  例 : apicl(config-leaf-vrf)# export map shared-route-map1	この VRF からコンシューマ VRF に (リーク) ルートをエクスポートするようこの VRF でルートマップを設定します。

## 例

次に、ルートマップを作成およびエクスポートする例を示します。

```
# CREATE A ROUTE-MAP
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf v1
apicl(config-leaf-vrf)# router-id 1.2.3.4
apicl(config-leaf-vrf)# route-map shared-route-map1
apicl(config-leaf-vrf-route-map)# ip prefix-list list1 permit 13.13.13.0/24
apicl(config-leaf-vrf-route-map)# match prefix-list list1
apicl(config-leaf-vrf-route-map-match)# contract provider prov1
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# exit

# EXPORT THE ROUTE-MAP
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf v1
apicl(config-leaf-vrf)# export map shared-route-map1
```

# 双方向ルート転送 (BFD) の設定

## BFD について

双方向フォワーディング検出 (BFD) は、メディア タイプ、カプセル化、トポロジ、およびルーティングプロトコルのために短時間での転送パス障害検出を提供するために設計された検

出プロトコルです。BFD を使用することで、さまざまなプロトコルの Hello メカニズムにより、変動速度ではなく一定速度で転送パス障害を検出できます。BFD はプロファイリングおよびプランニングを簡単にし、再コンバージェンス時間の一貫性を保ち、予測可能にします。

双方向フォワーディング検出 (BFD) を使用して、ピアリングルータの接続をサポートするように設定された ACI ファブリック境界リーフ スイッチ間の転送パスのサブセカンド障害検出時間を提供します。

## BFD グローバルの設定

デバイスのすべての BFD セッションの BFD セッションパラメータを設定できます。BFD セッションパラメータは、スリーウェイ ハンドシェイクの BFD ピア間でネゴシエートされます。

グローバル BFD を設定するには、次の手順を実行します。

- グローバル BFD 構成設定を設定します
- アクセスリーフ ポリシー グループを設定し、以前作成したグローバル BFD ポリシーを継承します
- 以前作成したリーフ ポリシー グループをリーフ スイッチまたはリーフ スイッチのグループに関連付けます

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>[no] template bfd {ip   ipv6} global-policy-name</b> 例： apic1(config)# <b>template bfd ip bfd_global</b>	BFD ポリシーテンプレートを作成します。
ステップ 3	<b>[no] echo-address ip-address</b> 例： apic1(config-bfd)# <b>echo-address 192.0.20.123</b> apic1(config-bfd)# <b>echo-address 34::1/64</b>	BFD エコーパケットの送信元アドレスとして使用する IP アドレスを指定します。
ステップ 4	<b>[no] slow-timer milliseconds</b> 例： apic1(config-bfd)# <b>slow-timer 2000</b>	エコー機能で使用する slow timer を設定します。この値はエコー機能がイネーブルの場合、BFD が新しいセッションを開始する速度および非同期

	コマンドまたはアクション	目的
		セッションがBFD制御パケットに使用する速度を決定します。 <b>slow-timer</b> 値は新しい制御パケット間隔として使用されますが、エコーパケットは設定されたBFD間隔を使用します。エコーパケットはリンク障害検出に使用されますが、低速の制御パケットはBFDセッションを維持します。指定できる範囲は1000～30000ミリ秒です。
ステップ5	<b>[no] min-tx milliseconds</b> 例： apicl (config-bfd) # <b>min-tx 100</b>	このデバイスがBFD Helloメッセージを送信する間隔を指定します。有効値は50～999ミリ秒です。
ステップ6	<b>[no] min-rx milliseconds</b> 例： apicl (config-bfd) # <b>min-rx 70</b>	このデバイスが別のBFDデバイスからのBFD Helloメッセージを受け付ける最小間隔を指定します。有効値は50～999ミリ秒です。
ステップ7	<b>[no] multiplier policy-name</b> 例： apicl (config-bfd) # <b>multiplier 3</b>	転送パスの障害を検出するまでに喪失した、別のBFDデバイスからのBFD Helloメッセージの数を指定します。指定できる範囲は1～50です。
ステップ8	<b>[no] echo-rx-interval policy-name</b> 例： apicl (config-bfd) # <b>echo-rx-interval 500</b>	このシステムがサポートできる受信BFDエコーパケット間の最小間隔を指定します。有効値は50～999ミリ秒です。
ステップ9	<b>exit</b> 例： apicl (config-bfd) # <b>exit</b>	グローバルコンフィギュレーションモードに戻ります。
ステップ10	<b>[no] template leaf-policy-group leaf-policy-name</b> 例： apicl (config) # <b>template leaf-policy-group leaf_pg1</b>	アクセススリーフポリシーグループを設定します。
ステップ11	<b>[no] inherit bfd {ip   ipv6} global-policy-name</b> 例： apicl (config-leaf-policy-group) # <b>inherit bfd ip bfd_global</b>	以前作成したグローバルポリシーを継承します。

	コマンドまたはアクション	目的
ステップ 12	<b>exit</b> 例： apic1(config-leaf-policy-group)# <b>exit</b>	グローバル コンフィギュレーション モードに戻ります。
ステップ 13	<b>[no] leaf-profile leaf-profile-name</b> 例： apic1(config)# <b>leaf-profile leaf_profile1</b>	リーフ プロファイルを設定します。
ステップ 14	<b>[no] leaf-group leaf-group-name</b> 例： apic1(config-leaf-profile)# <b>leaf-group leaf_group1</b>	リーフ スイッチのグループを作成または指定します。
ステップ 15	<b>[no] leaf-policy-group leaf-policy-name</b> 例： apic1(config-leaf-group)# <b>leaf-policy-group leaf_pg1</b>	リーフ スイッチに関連付けられる、以前作成したリーフ ポリシーグループを指定します。
ステップ 16	<b>[no] leaf leaf-range</b> 例： apic1(config-leaf-group)# <b>leaf 101-102</b>	リーフ スイッチグループに1つ以上のリーフ スイッチを追加します。

### 例

この例では、グローバル BFD を設定し、リーフ スイッチのグループに適用する方法を示します。

```
# CONFIGURE BFD GLOBAL POLICIES
apic1# configure
apic1(config)# template bfd ip bfd_global
apic1(config-bfd)# echo-address 192.0.20.123
apic1(config-bfd)# slow-timer 2000
apic1(config-bfd)# min-tx 100
apic1(config-bfd)# min-rx 70
apic1(config-bfd)# multiplier 3
apic1(config-bfd)# echo-rx-interval 500
apic1(config-bfd)# exit

# CONFIGURE AN ACCESS LEAF POLICY GROUP AND INHERIT BFD GLOBAL POLICIES
apic1(config)# template leaf-policy-group leaf_pg1
apic1(config-leaf-policy-group)# inherit bfd ip bfd_global
apic1(config-leaf-policy-group)# exit

# CONFIGURE A LEAF GROUP AND ASSOCIATE THE LEAF POLICY GROUP
apic1(config)# leaf-profile leaf_profile1
apic1(config-leaf-profile)# leaf-group leaf_group1
apic1(config-leaf-group)# leaf-policy-group leaf_pg1
```

```
apicl(config-leaf-group)# leaf 101-102
```

## NX-OS スタイル CLI を使用したリーフスイッチでの BFD のグローバルな設定

### 手順

**ステップ 1** NX-OS CLI を使用して BFD IPV4 グローバル設定 (bfdIpv4InstPol) を設定するには :

例 :

```
apicl# configure
apicl(config)# template bfd ip bfd_ipv4_global_policy
apicl(config-bfd)# [no] echo-address 1.2.3.4
apicl(config-bfd)# [no] slow-timer 2500
apicl(config-bfd)# [no] min-tx 100
apicl(config-bfd)# [no] min-rx 70
apicl(config-bfd)# [no] multiplier 3
apicl(config-bfd)# [no] echo-rx-interval 500
apicl(config-bfd)# exit
```

**ステップ 2** NX-OS CLI を使用して BFD IPV6 グローバル設定 (bfdIpv6InstPol) を設定するには :

例 :

```
apicl# configure
apicl(config)# template bfd ipv6 bfd_ipv6_global_policy
apicl(config-bfd)# [no] echo-address 34::1/64
apicl(config-bfd)# [no] slow-timer 2500
apicl(config-bfd)# [no] min-tx 100
apicl(config-bfd)# [no] min-rx 70
apicl(config-bfd)# [no] multiplier 3
apicl(config-bfd)# [no] echo-rx-interval 500
apicl(config-bfd)# exit
```

**ステップ 3** NX-OS CLI を使用してアクセス リーフ ポリシー グループ (infraAccNodePGrp) を設定し、以前に作成した BFD グローバル ポリシーを継承するには:

例 :

```
apicl# configure
apicl(config)# template leaf-policy-group test_leaf_policy_group
apicl(config-leaf-policy-group)# [no] inherit bfd ip bfd_ipv4_global_policy
apicl(config-leaf-policy-group)# [no] inherit bfd ipv6 bfd_ipv6_global_policy
apicl(config-leaf-policy-group)# exit
```

**ステップ 4** NX-OS CLI を使用して以前に作成したリーフ ポリシー グループをリーフに関連付けるには:

例 :

```
apicl(config)# leaf-profile test_leaf_profile
apicl(config-leaf-profile)# leaf-group test_leaf_group
apicl(config-leaf-group)# leaf-policy-group test_leaf_policy_group
```

```
apic1(config-leaf-group)# leaf 101-102
apic1(config-leaf-group)# exit
```

## NX-OS スタイル CLI を使用したスパインスイッチ上の BFD のグローバル設定

次の手順を使用して、NX-OS スタイル CLI を使用してスパインスイッチの BFD をグローバルに設定します。

### 手順

**ステップ 1** NX-OS CLI を使用して BFD IPV4 グローバル設定 (bfdIpv4InstPol) を設定するには :

例 :

```
apic1# configure
apic1(config)# template bfd ip bfd_ipv4_global_policy
apic1(config-bfd)# [no] echo-address 1.2.3.4
apic1(config-bfd)# [no] slow-timer 2500
apic1(config-bfd)# [no] min-tx 100
apic1(config-bfd)# [no] min-rx 70
apic1(config-bfd)# [no] multiplier 3
apic1(config-bfd)# [no] echo-rx-interval 500
apic1(config-bfd)# exit
```

**ステップ 2** NX-OS CLI を使用して BFD IPV6 グローバル設定 (bfdIpv6InstPol) を設定するには :

例 :

```
apic1# configure
apic1(config)# template bfd ipv6 bfd_ipv6_global_policy
apic1(config-bfd)# [no] echo-address 34::1/64
apic1(config-bfd)# [no] slow-timer 2500
apic1(config-bfd)# [no] min-tx 100
apic1(config-bfd)# [no] min-rx 70
apic1(config-bfd)# [no] multiplier 3
apic1(config-bfd)# [no] echo-rx-interval 500
apic1(config-bfd)# exit
```

**ステップ 3** NX-OS CLI を使用してスパイン ポリシー グループを設定し以前作成した BFD グローバル ポリシーを継承するには :

例 :

```
apic1# configure
apic1(config)# template spine-policy-group test_spine_policy_group
apic1(config-spine-policy-group)# [no] inherit bfd ip bfd_ipv4_global_policy
apic1(config-spine-policy-group)# [no] inherit bfd ipv6 bfd_ipv6_global_policy
apic1(config-spine-policy-group)# exit
```

**ステップ 4** NX-OS を使用して以前作成したスパインポリシーグループをスパインスイッチに関連付けるには :

例：

```
apicl# configure
apicl(config)# spine-profile test_spine_profile
apicl(config-spine-profile)# spine-group test_spine_group
apicl(config-spine-group)# spine-policy-group test_spine_policy_group
apicl(config-spine-group)# spine 103-104
apicl(config-leaf-group)# exit
```

## グローバル BFD 設定のオーバーライド

### BFD インターフェイス オーバーライド ポリシーの設定

明示的な BFD を設定できる、3つのサポート対象のインターフェイス（ルーテッド L3 インターフェイス、外部インターフェイス SVI とルーテッドサブインターフェイス）があります。グローバルコンフィギュレーションを使用しないで、さらに特定のインターフェイスの明示的な設定をしたい場合、特定のスイッチまたは一連のすべてのインターフェイスに適用される独自のグローバルコンフィギュレーションを作成できます。特定のインターフェイス上の特定のスイッチの粒度がさらに必要な場合、このインターフェイスオーバーライド設定を使用する必要があります。

#### 始める前に

テナントはすでに作成されています。

#### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apicl# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： apicl(config)# <b>tenant exampleCorp</b>	設定するテナントを指定します。
ステップ 3	<b>vrf context vrf-name</b> 例： apicl(config-tenant)# <b>vrf context vrf1</b>	テナントと VRF を関連付けます。
ステップ 4	<b>exit</b> 例： apicl(config-tenant-vrf)# <b>exit</b>	テナントコンフィギュレーションモードに戻ります。

	コマンドまたはアクション	目的
ステップ 5	<b>exit</b> 例： apic1(config-tenant)# <b>exit</b>	グローバル コンフィギュレーション モードに戻ります。
ステップ 6	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 7	<b>[no] vrf context tenant tenant-name vrf vrf-name</b> 例： apic1(config-leaf)# <b>vrf context tenant exampleCorp vrf vrf1</b>	ノードのテナント VRF を設定します。
ステップ 8	<b>exit</b> 例： apic1(config-leaf-vrf)# <b>exit</b>	リーフ コンフィギュレーション モードに戻ります。
ステップ 9	<b>[no] interface type</b> 例： apic1(config-leaf)# <b>interface eth 1/18</b>	インターフェイス設定モードを開始します。
ステップ 10	<b>[no] vrf member tenant tenant-name vrf vrf-name</b> 例： apic1(config-leaf-if)# <b>vrf member tenant exampleCorp vrf vrf1</b>	
ステップ 11	<b>exit</b> 例： apic1(config-leaf-if)# <b>exit</b>	リーフ コンフィギュレーション モードに戻ります。
ステップ 12	<b>[no] template bfd template-name tenant tenant-name</b> 例： apic1(config-leaf)# <b>template bfd bfdIfPol1 tenant exampleCorp</b>	BFD インターフェイス ポリシーを設定します。
ステップ 13	<b>[no] echo-mode enable</b> 例： apic1(config-template-bfd-pol)# <b>echo-mode enable</b>	BFD 制御パケットに加えて BFD エコーパケットを送信をイネーブルまたはディセーブルにします。

	コマンドまたはアクション	目的
ステップ 14	<b>[no] echo-rx-interval <i>policy-name</i></b> 例： apicl (config-template-bfd-pol) # <b>echo-rx-interval 500</b>	このシステムがサポートできる受信 BFD エコー パケット間の最小間隔を指定します。有効値は 50 ～ 999 ミリ秒です。
ステップ 15	<b>[no] min-tx milliseconds</b> 例： apicl (config-template-bfd-pol) # <b>min-tx 100</b>	このデバイスが、BFD hello メッセージを送信する間隔を指定します。有効値は 50 ～ 999 ミリ秒です。
ステップ 16	<b>[no] min-rx milliseconds</b> 例： apicl (config-template-bfd-pol) # <b>min-rx 70</b>	このデバイスが別の BFD デバイスからの BFD Hello メッセージを受け付ける最小間隔を指定します。有効値は 50 ～ 999 ミリ秒です。
ステップ 17	<b>[no] multiplier <i>policy-name</i></b> 例： apicl (config-template-bfd-pol) # <b>multiplier 5</b>	転送パスの障害を検出するまでに喪失した、別の BFD デバイスからの BFD Hello メッセージの数を指定します。指定できる範囲は 1 ～ 50 です。
ステップ 18	<b>[no] optimize subinterface</b> 例： apicl (config-template-bfd-pol) # <b>optimize subinterface</b>	サブインターフェイスの最適化をイネーブまたはディセーブにします。BFD により、設定されているすべてのサブインターフェイスのセッションが作成されます。BFD により、設定されている最小の VLAN ID を持つサブインターフェイスがマスターサブインターフェイスとして設定され、そのサブインターフェイスは親インターフェイスの BFD セッションパラメータを使用します。残りのサブインターフェイスは slow timer を使用します。最適化されたサブインターフェイスセッションでエラーが検出されると、BFD により、その物理インターフェイスのすべてのサブインターフェイスがダウンとマークされます。

### 例

この例では、BFD オーバーライド ポリシーを作成し、インターフェイスに適用する方法を示します。

```
apicl# configure
```

```

apic1(config)# tenant exampleCorp
apic1(config-tenant)# vrf context vrf1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf vrf1
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface eth 1/18
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf vrf1
apic1(config-leaf-if)# exit

# CONFIGURE BFD INTERFACE OVERRIDE POLICY
apic1(config-leaf)# template bfd bfdIfPol1 tenant exampleCorp
apic1(config-template-bfd-pol)# echo-mode enable
apic1(config-template-bfd-pol)# echo-rx-interval 500
apic1(config-template-bfd-pol)# min-tx 100
apic1(config-template-bfd-pol)# min-rx 70
apic1(config-template-bfd-pol)# multiplier 5
apic1(config-template-bfd-pol)# optimize subinterface

```

## インターフェイスへの BFD インターフェイス オーバーライド ポリシーの適用

ルーテッド L3 インターフェイス、外部 SVI インターフェイスとルーテッドサブインターフェイスに、BFD インターフェイス オーバーライド ポリシーを適用できます。

### 始める前に

BFD インターフェイス オーバーライド ポリシーがすでに作成されています。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーション モードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>[no] interface type</b> 例： apic1(config-leaf)# <b>interface Ethernet 1/15</b>	インターフェイス設定モードを開始します。サポートされているインターフェイスは、ルーテッド L3 インターフェイス、外部 SVI インターフェイスとルーテッドサブインターフェイスです。
ステップ 4	<b>[no] ipv6 address ipv6-address [preferred]</b> 例： apic1(config-leaf-if)# <b>ipv6 address 2001::10:1/64 preferred</b>	インターフェイスからのトラフィックのデフォルトの発信元アドレスである IP アドレスを指定します。

	コマンドまたはアクション	目的
		(注) このコマンドは、インターフェイスが IPv6 インターフェイスである場合にのみ使用されます。
ステップ 5	<b>[no] vrf member tenant <i>tenant-name</i> vrf <i>vrf-name</i></b>  例： <code>apic1(config-leaf-if)# vrf member tenant exampleCorp vrf vrf1</code>	テナント VRF にインターフェイスを接続します。  (注) このコマンドは、インターフェイスが VLAN インターフェイスである場合にのみ使用されます。
ステップ 6	<b>bfd {ip   ipv6} tenant mode</b>  例： <code>apic1(config-leaf-if)# bfd ip tenant mode</code>	BFD テナント モードをイネーブルにします。
ステップ 7	<b>bfd {ip   ipv6} inherit interface-policy <i>policy-name</i></b>  例： <code>apic1(config-leaf-if)# bfd ip inherit interface-policy bfdIfPoll</code>	指定された BFD インターフェイス テンプレート ポリシーを継承します。
ステップ 8	<b>bfd {ip   ipv6} authentication keyed-sha1 keyid <i>keyid</i> key <i>key</i></b>  例： <code>apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password</code>	キー付き SHA-1 として BFD 認証を設定します。

### 例

この例では、IPv4 アドレスを持つ L3 インターフェイス上に以前作成した BFD インターフェイス ポリシーを継承する方法を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/15
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdIfPoll
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password
```

この例では、IPv6 アドレスを持つ L3 インターフェイス上に以前作成した BFD インターフェイス ポリシーを継承する方法を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/15
apic1(config-leaf-if)# ipv6 address 2001::10:1/64 preferred
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdIfPoll
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password

```

この例では、IPv4 アドレスを持つ VLAN インターフェイス上で BFD を設定する方法を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 15
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf vrf1
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdIfPoll
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password

```

この例では、IPv6 アドレスを持つ VLAN インターフェイス上で BFD を設定する方法を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 15
apic1(config-leaf-if)# ipv6 address 2001::10:1/64 preferred
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf vrf1
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdIfPoll
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password

```

## コンシューマ プロトコル上の BFD の有効化

これらの手順は、BFD 機能を利用する4つのコンシューマプロトコル（BGP、EIGRP、OSPF、静的ルート）で BFD を有効にする手順を紹介します。

### BGP コンシューマ プロトコルの BFD の有効化

始める前に

テナントはすでに作成されています。

手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例 : apic1# <b>configure</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。

	コマンドまたはアクション	目的
ステップ 2	<b>bgp-fabric</b> 例： apicl(config-bgp-fabric)# <b>bgp-fabric</b>	ファブリックの BGP 設定モードを開始します。
ステップ 3	<b>asn asn-number</b> 例： apicl(config-bgp-fabric)# <b>asn 200</b>	BGP 自律システム番号 (ASN) を指定します。
ステップ 4	<b>exit</b> 例： apicl(config-bgp-fabric)# <b>exit</b>	グローバル コンフィギュレーション モードに戻ります。
ステップ 5	<b>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 6	<b>router bgp asn-number</b> 例： apicl(config-leaf)# <b>router bgp 200</b>	BGP ポリシー設定を入力します。
ステップ 7	<b>vrf member tenant tenant-name vrf vrf-name</b> 例： apicl(config-bgp)# <b>vrf member tenant exampleCorp vrf v100</b>	後続のポリシー設定モード コマンドと関連付ける VRF インスタンスを指定します。
ステップ 8	<b>neighbor ip-address[/masklength]</b> 例： apicl(config-leaf-bgp-vrf)# <b>neighbor 1.2.3.4</b>	ネイバーの IP アドレスを指定します。マスク長は 32 です。
ステップ 9	<b>[no] bfd enable</b> 例： apicl(config-leaf-bgp-vrf-neighbor)# <b>bfd enable</b>	BGP コンシューマプロトコルで BFD を有効または無効にします。

### 例

次に、BGP コンシューマプロトコルで BFD を有効にする例を示します。

```
apicl# configure
apicl(config)# bgp-fabric
apicl(config-bgp-fabric)# asn 200
apicl(config-bgp-fabric)# exit
```

```

apic1(config)# leaf 101
apic1(config-leaf)# router bgp 200
apic1(config-bgp)# vrf member tenant exampleCorp vrf v100
apic1(config-leaf-bgp-vrf)# neighbor 1.2.3.4
apic1(config-leaf-bgp-vrf-neighbor)# bfd enable

```

## EIGRP コンシューマ プロトコルの BFD の有効化

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>[no] interface type</b> 例： apic1(config-leaf)# <b>interface Ethernet 1/15</b>	インターフェイス設定モードを開始します。
ステップ 4	<b>[no] {ip   ipv6} bfd eigrp enable</b> 例： apic1(config-leaf-if)# <b>ip bfd eigrp enable</b>	EIGRP コンシューマ プロトコルで BFD を有効または無効にします。

### 例

次に、EIGRP コンシューマ プロトコルで BFD を有効にする例を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/15
apic1(config-leaf-if)# ip bfd eigrp enable

```

## OSPF コンシューマ プロトコルの BFD の有効化

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>[no] interface type</b> 例： apic1(config-leaf)# <b>interface vlan 123</b>	インターフェイス設定モードを開始します。
ステップ 4	<b>[no] ip ospf bfd enable</b> 例： apic1(config-leaf-if)# <b>ip ospf bfd enable</b>	OSPF コンシューマ プロトコルで BFD を有効または無効にします。

### 例

次に、OSPF コンシューマ プロトコルで BFD を有効にする例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 123
apic1(config-leaf-if)# ip ospf bfd enable
```

## 静的ルート コンシューマ プロトコルの BFD の有効化

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。

	コマンドまたはアクション	目的
ステップ 3	<b>[no] vrf context tenant <i>tenant-name</i> vrf <i>vrf-name</i></b>  例 : <pre>apic1(config-leaf)# vrf context tenant exampleCorp vrf vrf1</pre>	ノードのテナント VRF を設定します。
ステップ 4	<b>[no] {ip   ipv6} route <i>ip-prefix/masklen</i> <i>next-hop-address</i> bfd</b>  例 : <pre>apic1(config-leaf-vrf)# ip route 10.0.0.1/16 10.0.0.5 bfd</pre>	静的ルートのコンシューマ プロトコルで BFD を有効または無効にします。

### 例

次に、静的ルートのコンシューマ プロトコルで BFD を有効にする例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf vrf1
apic1(config-leaf-vrf)# ip route 10.0.0.1/16 10.0.0.5 bfd
```

## NX-OS スタイルの CLI を使用した BFD コンシューマ プロトコルの設定

### 手順

**ステップ 1** NX-OS は、CLI を使用して、BGP コンシューマ プロトコルを BFD をイネーブルにします。

例 :

```
apic1# configure
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 200
apic1(config-bgp-fabric)# exit
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 200
apic1(config-bgp)# vrf member tenant t0 vrf v0
apic1(config-leaf-bgp-vrf)# neighbor 1.2.3.4
apic1(config-leaf-bgp-vrf-neighbor)# [no] bfd enable
```

**ステップ 2** NX-OS は、CLI を使用して、EIGRP コンシューマ プロトコルを BFD をイネーブルにします。

例 :

```
apic1(config-leaf-if)# [no] ip bfd eigrp enable
```

**ステップ 3** NX-OS は、CLI を使用して、OSPF コンシューマ プロトコルを BFD をイネーブルにします。

例：

```
apicl(config-leaf-if)# [no] ip ospf bfd enable

apicl# configure
apicl(config)# spine 103
apicl(config-spine)# interface ethernet 5/3.4
apicl(config-spine-if)# [no] ip ospf bfd enable
```

**ステップ4** NX-OS は、CLI を使用して、スタティック ルート コンシューマ プロトコルを BFD をイネーブルにします。

例：

```
apicl(config-leaf-vrf)# [no] ip route 10.0.0.1/16 10.0.0.5 bfd

apicl(config)# spine 103
apicl(config-spine)# vrf context tenant infra vrf overlay-1
apicl(config-spine-vrf)# [no] ip route 21.1.1.1/32 32.1.1.1 bfd
```

**ステップ5** NX-OS は、CLI を使用して、IS-IS コンシューマ プロトコルを BFD をイネーブルにします。

例：

```
apicl(config)# leaf 101
apicl(config-spine)# interface ethernet 1/49
apicl(config-spine-if)# isis bfd enabled
apicl(config-spine-if)# exit
apicl(config-spine)# exit

apicl(config)# spine 103
apicl(config-spine)# interface ethernet 5/2
apicl(config-spine-if)# isis bfd enabled
apicl(config-spine-if)# exit
apicl(config-spine)# exit
```

## レイヤ3 マルチキャストの設定

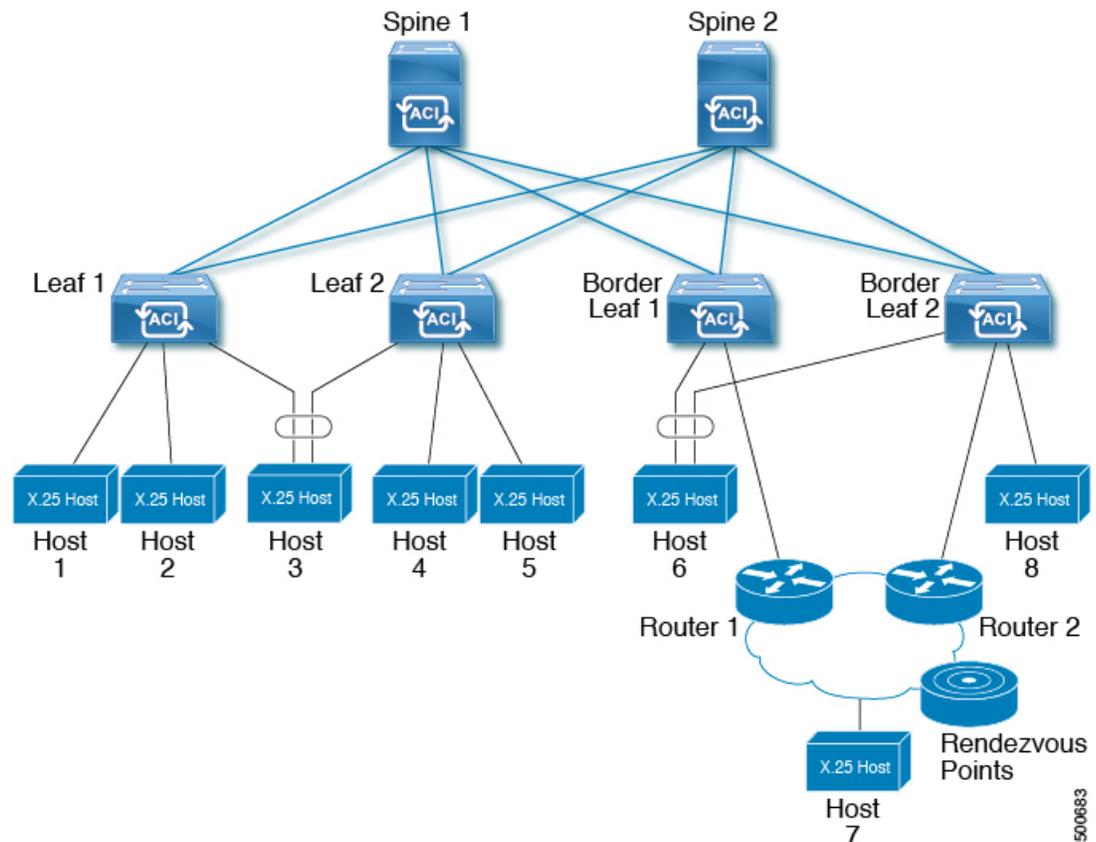
### レイヤ3 マルチキャスト

ACI ファブリックでは、ほとんどのユニキャストとマルチキャスト ルーティングが同じ境界リーフスイッチで稼働しており、ユニキャスト ルーティング プロトコル上でマルチキャスト プロトコルが稼働しています。

このアーキテクチャでは、ボーダーリーフスイッチのみが完全な Protocol Independent Multicast (PIM) プロトコルを実行します。非ボーダーリーフスイッチは、インターフェイス上でパッシブモードの PIM を実行します。これらは、その他の PIM ルータとピアリングしません。ボーダーリーフスイッチは、L3 Out を介してそれらの接続された他の PIM ルータとピアリングし、またそれら相互にもピアリングします。

次の図に、マルチキャストクラウド内のルータ（R1 と R2）に接続しているボーダーリーフ（BL）スイッチを示します。マルチキャストルーティングを必要とするファブリック内の各 Virtual Routing and Forwarding（VRF）は、それぞれ別に外部マルチキャストルータとピアリングします。

図 8: マルチキャストクラウドの概要



## レイヤ3 マルチキャストの設定に関するガイドライン

次のガイドラインを参照してください。

- レイヤ3 マルチキャストの設定は VRF レベルで実行されます。そのため、VRF 内とマルチキャスト内のプロトコル機能が VRF で有効になり、各マルチキャスト VRF を個別にオンまたはオフにすることができます。
- マルチキャストで VRF が有効になると、有効になった VRF の個別のブリッジドメイン (BD) と L3 Out を有効にしてマルチキャストを設定できます。デフォルトでは、マルチキャストはすべての BD およびレイヤ3 Out で無効になっています。
- 現時点では、レイヤ3 マルチキャストは、共有 L3 Out で設定された VRF ではサポートされていません。
- Any Source Multicast (ASM) と Source-Specific Multicast (SSM) はサポートされています。

- 現時点では、双方向 PIM、ACI ファブリック内のランデブー ポイント (RP)、および PIM IPv6 はサポートされていません。
- IGMP スヌーピングは、マルチキャストルーティングが有効になっているパーペイシブブリッジドメインでは無効にできません。
- マルチキャストルータは、パーペイシブブリッジドメインではサポートされていません。
- 次の -EX でレイヤ3のマルチキャスト機能がサポートされているリーフスイッチのモデルします。
  - N9K-93180YC-EX
  - N9K-93108TC-EX
  - N9K-93180LC-EX
- レイヤ3ポートとサブインターフェイスはサポートされていますが、外部 SVI はサポートされていません。外部 SVI がサポートされていないため、PIM を L3-VPC で有効にできません。
- マルチポッドのレイヤ3マルチキャストサポートについて、イングレスリーフスイッチはマルチキャストルーティングに対応しているブリッジドメインに接続されたソースからパケットを受信するとき、イングレスリーフスイッチはファブリックにルート済みVRFコピーのみ送信します（ルート済みとは TTL が1に減少し、source-mac が拡散型サブネットMACに再度書き込みされます）。また、出力リーフスイッチも、関連するすべてのブリッジドメイン内の受信者へパケットをルーティングします。そのため、受信者のブリッジドメインが送信元と同じで、リーフスイッチが送信元とは異なる場合、その受信者は同じブリッジドメイン内であっても、ルーティングされたコピーを受け取り続けます。

詳細については、次のリンクで、既存のレイヤ2設計を活用する multipod サポートレイヤ3マルチキャストに関する詳細情報を参照してください。 [追加ポッド](#)。
- リリース 3.1(1x) で始まる、FEX にマルチキャストのレイヤ3はサポートされています。マルチキャストのソースまたは FEX ポートに接続されているレシーバがサポートされません。詳細については、テスト環境で FEX を追加する方法について、設定、次の URL をアプリケーションセントリックインフラストラクチャとファブリックエクステンダを参照してください: <https://www.cisco.com/c/en/us/support/docs/cloud-systems-management/application-policy-infrastructure-controller-apic/200529-Configure-a-Fabric-Extender-with-Applica.html>。リリース 3.1(1x) 以降のレイヤ3マルチキャストでは FEX がサポートされていません。マルチキャストのソースまたは FEX ポートに接続されているレシーバはサポートされていません。



(注) Cisco ACI は、IP フラグメンテーションをサポートしていません。したがって、外部ルータへのレイヤ3 Outside (L3Out) 接続、または Inter-Pod Network (IPN) を介した multipod 接続を設定する場合は、MTU が両側で適切に設定されていることが重要です。ACI、Cisco NX-OS、Cisco IOS などの一部のプラットフォームでは、設定された MTU 値は IP ヘッダーを考慮に入れています（結果として、最大パケットサイズは、ACI で 9216 バイト、NX-OS および IOS で 9000 バイトに設定されます）。ただし、IOS XR などの他のプラットフォームは、パケットヘッダーのを除く MTU 値を設定します（結果として最大パケットサイズは 8986 バイトになります）。

各プラットフォームの適切な MTU 値については、それぞれの設定ガイドを参照してください。

CLI ベースのコマンドを使用して MTU をテストすることを強く推奨します。たとえば、Cisco NX-OS CLI で `ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1` などのコマンドを使用します。

## レイヤ3 マルチキャストの設定手順

ここでは、レイヤ3 マルチキャストの設定手順を示します。手順は次のとおりです。

1. テナント VRF の PIM オプションを設定します。
2. VRF の IGMP オプションを設定します。
3. テナントに L3 Out 設定し、PIM を有効にしてリーフ インターフェイスを設定します。
4. 目的のブリッジドメイン内で PIM を有効にします。

## レイヤ3 マルチキャストの PIM オプションの設定

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： <code>apic1# configure</code>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： <code>apic1(config)# tenant exampleCorp</code>	設定するテナントを指定します。
ステップ 3	<b>vrf context vrf-name</b> 例： <code>apic1(config-tenant)# vrf context exampleCorp_vrf1</code>	テナントと VRF を関連付けます。

	コマンドまたはアクション	目的
ステップ4	<b>[no] ip pim</b> 例： apicl(config-tenant-vrf)# ip pim	Protocol Independent Multicast (PIM) を設定します。
ステップ5	(任意) <b>[no] ip pim auto-rp {forward [listen]  listen  mapping-agent-policy マッピングエージェントポリシー名}</b> 例： apicl(config-tenant-vrf)# ip pim auto-rp forward listen	PIM 自動 RP (ランデブーポイント) オプションを設定します。Auto-RPは、PIM ネットワークにおけるグループから RP へのマッピングの配信を自動化します。自動 RP メッセージの転送、自動 RP メッセージのリッスン、またはマッピングエージェントメッセージのフィルタリングに対するルートマップポリシーの関連付けを選択できます。
ステップ6	(任意) <b>[no] ip pim bsr {forward [listen]  listen   bsr-policy mapping-agent-policy-name}</b> 例： apicl(config-tenant-vrf)# ip pim bsr forward listen	PIM ブートストラップルータ (BSR) オプションを設定します。BSRは、RP 機能およびグループの RP 情報のリレーに候補ルータを使用するという点において自動 RP と同様に動作します。RP 情報は、PIM メッセージ内で伝送される BSR メッセージを通じて配信されます。ブートストラップ/RP 候補メッセージの転送、ブートストラップ/RP 候補メッセージのリッスン、または BSR メッセージのフィルタリングに対するルートマップポリシーの関連付けを選択できます。
ステップ7	(任意) <b>[no] ip pim fast-convergence</b> 例： apicl(config-tenant-vrf)# ip pim fast-convergence	PIM 高速コンバージェンス機能を有効にして、スイッチが応答しないネイバーを迅速に検出できるようにします。
ステップ8	(任意) <b>[no] ip pim mtu mtu-size</b> 例： apicl(config-tenant-vrf)# ip pim mtu 1500	PIM メッセージの最大サイズを設定します。有効な範囲は 1500 ~ 65536 バイトです。
ステップ9	(任意) <b>[no] ip pim register-policy register-policy-name</b> 例： apicl(config-tenant-vrf)# ip pim register-policy regPolicy1	登録メッセージのフィルタリングにポリシーの名前を指定します。

	コマンドまたはアクション	目的
ステップ 10	<p>(任意) <b>[no] ip pim register-rate-limit mtu-size</b></p> <p>例 :</p> <pre>apic1(config-tenant-vrf)# ip pim register-rate-limit 1024</pre>	PIM データレジスタにレート制限を指定します。範囲は毎秒 0 ~ 65535 パケットです。
ステップ 11	<p>(任意) <b>[no] ip pim register-source ip-address</b></p> <p>例 :</p> <pre>apic1(config-tenant-vrf)# ip pim register-source 192.0.20.123</pre>	PIM メッセージの送信元 IP アドレスを設定します。
ステップ 12	<p>(任意) <b>[no] ip pim rp-address ip-address [route-map route-map-name]</b></p> <p>例 :</p> <pre>apic1(config-tenant-vrf)# ip pim rp-address 192.0.20.99</pre>	マルチキャストグループ範囲に、静的ルートプロセッサ (RP) アドレスを設定します。
ステップ 13	<p>(任意) <b>[no] ip pim sg-expiry-timer ip-address [sg-list route-map-name]</b></p> <p>例 :</p> <pre>apic1(config-tenant-vrf)# ip pim sg-expiry-timer 4096</pre>	PIM スパースモード (PIM-SM) (S,G) マルチキャストルートに (S,G) 期限切れタイマーの間隔を設定します。値の範囲は 180 ~ 604801 秒です。The optional <b>sg-list</b> parameter specifies S,G values to which the timer applies. デフォルトは 4096 です。
ステップ 14	<p>(任意) <b>[no] ip pim ssm route-map route-map-name</b></p> <p>例 :</p> <pre>apic1(config-tenant-vrf)# ip pim ssm route-map SSMRtMap</pre>	IP マルチキャストの拡張機能である Source Specific Multicast (SSM) を設定します。この機能を使用すると、受信者に転送されるデータグラムトラフィックは、その受信者が明示的に加入しているマルチキャスト送信元からのトラフィックだけになります。ルートマップポリシーは、グループのプレフィックスのリストを表示します。
ステップ 15	<p>(任意) <b>[no] ip pim state-limit max-entries [reserved route-map-name [maximum-reserve-state-entries]]</b></p> <p>例 :</p> <pre>apic1(config-tenant-vrf)# ip pim state-limit 100000 reserved myReservedPolicy 40000</pre>	現在の VRF インスタンスの PIM の状態エントリの最大数を設定します。最大状態エントリ数の範囲は 0 ~ 4294967295 です。オプションで、ポリシーマップに指定されるルート用に予約する状態エントリ数を指定することができ、また、この VRF で許可される最大予約済み (*,G) と (S,G) エントリを指定できます。子の数は、許可

	コマンドまたはアクション	目的
		される最大状態数以下である必要があります。範囲は1～4294967295です。
ステップ 16	(任意) <b>[no] ip pim use-shared-tree-only group-list policy-name</b> 例： apicl(config-tenant-vrf)# <b>ip pim use-shared-tree-only group-list myGroup1</b>	PIM (*,G) 状態のみを作成します (送信元の状態を作成しない場合)。このポリシーで、この機能を適用するグループプレフィックスを定義します。
ステップ 17	<b>exit</b> 例： apicl(config-tenant-vrf)# <b>exit</b>	テナントコンフィギュレーションモードに戻ります。

## 次のタスク

VRF の IGMP オプションを設定します。

## レイヤ3 マルチキャストの VRF での IGMP オプションの設定

## 始める前に

テナント VRF の PIM オプションを設定します。

## 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apicl# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： apicl(config)# <b>tenant exampleCorp</b>	設定するテナントを指定します。
ステップ 3	<b>vrf context vrf-name</b> 例： apicl(config-tenant)# <b>vrf context vrf1</b>	テナントと VRF を関連付けます。
ステップ 4	<b>[no] ip igmp</b> 例： apicl(config-tenant-vrf)# <b>ip igmp</b>	インターネットグループ管理プロトコル (IGMP) を有効にします。

	コマンドまたはアクション	目的
ステップ 5	<b>exit</b> 例： apic1(config-tenant-vrf)# <b>exit</b>	テナントコンフィギュレーションモードに戻ります。
ステップ 6	<b>interface bridge-domain <i>bd-name</i></b> 例： apic1(config-tenant)# <b>interface bridge-domain exampleCorp_bd1</b>	テナント インターフェイス コンフィギュレーション モードを開始してブリッジドメインを設定します。
ステップ 7	<b>[no] ip multicast</b> 例： apic1(config-tenant-interface)# <b>ip multicast</b>	インターフェイスでIPマルチキャストルーティングを有効にします。
ステップ 8	<b>[no] ip igmp allow-v3-asm</b> 例： apic1(config-tenant-interface)# <b>ip igmp allow-v3-asm</b>	Any Source Multicast (ASM) グループに関する IGMPv3 レポートの送信元アドレスのフィルタリングを許可します。
ステップ 9	<b>[no] ip igmp fast-leave</b> 例： apic1(config-tenant-interface)# <b>ip igmp fast-leave</b>	IP IGMP スヌーピングの高速脱退処理を有効にします。この機能は、IGMPv2 プロトコルのホストレポート抑制メカニズムのために明示的に追跡できない IGMPv2 ホストをサポートします。高速脱退が有効になっている場合、IGMP ソフトウェアは、各ポートに接続されたホストが 1 つだけであると見なします。
ステップ 10	<b>[no] ip igmp group-timeout <i>seconds</i></b> 例： apic1(config-tenant-interface)# <b>ip igmp group-timeout 260</b>	IGMPv2 のグループメンバーシップ タイムアウトを設定します。値の範囲は 3～65535 秒です。デフォルト値は 260 秒です。
ステップ 11	<b>[no] ip igmp inherit interface-policy <i>policy-name</i></b> 例： apic1(config-tenant-interface)# <b>ip igmp inherit interface-policy MyIfPolicy</b>	このインターフェイスに IGMP インターフェイス ポリシーを関連付けます。
ステップ 12	<b>[no] ip igmp join-group route-map <i>route-map-name</i></b> 例：	1 つ以上のマルチキャスト グループをインターフェイスに静的にバインドします。ルート マップ ポリシーは、グループのプレフィクス、グループの範

	コマンドまたはアクション	目的
	<code>apicl(config-tenant-interface)# ip igmp join-group route-map MyGroupsRMap</code>	囲、および送信元プレフィックスのリストを表示します。
ステップ 13	<b>[no] ip igmp last-member-query-count <i>count</i></b>  例： <code>apicl(config-tenant-interface)# ip igmp last-member-query-count 2</code>	ホストの Leave メッセージを受信してから、IGMP クエリーが送信される回数を設定します。範囲は 1～5 クエリーです。デフォルトは 2 クエリーです。
ステップ 14	<b>[no] ip igmp last-member-query-response-time <i>seconds</i></b>  例： <code>apicl(config-tenant-interface)# ip igmp last-member-query-response-time 1</code>	メンバーシップレポートを送信してから、ソフトウェアがグループステートを解除するまでのクエリーインターバルを設定します。値の範囲は 1～25 秒です。デフォルト値は 1 秒です。
ステップ 15	<b>[no] ip igmp querier-timeout <i>seconds</i></b>  例： <code>apicl(config-tenant-interface)# ip igmp querier-timeout 255</code>	前クエリアがクエリーを停止してから、自身がクエリアとして処理を引き継ぐまで、ソフトウェアが待機する秒数を設定します。指定できる範囲は 1～65535 秒です。デフォルト値は 255 秒です。
ステップ 16	<b>[no] ip igmp query-interval <i>seconds</i></b>  例： <code>apicl(config-tenant-interface)# ip igmp query-interval 125</code>	IGMP ホストクエリーメッセージの送信頻度を設定します。大きな値を設定すると、ソフトウェアによる IGMP クエリーの送信頻度が低くなるため、ネットワーク上の IGMP メッセージ数を調整できます。指定できる範囲は 1～18000 秒です。デフォルト値は 125 秒です。
ステップ 17	<b>[no] ip igmp query-max-response-time <i>seconds</i></b>  例： <code>apicl(config-tenant-interface)# ip igmp query-max-response-time 10</code>	IGMP クエリーでアドバタイズされる応答時間を設定します。大きな値を設定すると、ホストの応答時間が延長されるため、ネットワークの IGMP メッセージのバースト性を調整できます。この値は、クエリーインターバルよりも短く設定する必要があります。値の範囲は 1～25 秒です。デフォルトは 10 秒です。
ステップ 18	<b>[no] ip igmp report-link-local-groups</b>  例： <code>apicl(config-tenant-interface)# ip igmp report-link-local-groups</code>	224.0.0.0/24 に含まれるグループに対して、レポート送信をイネーブルにします。リンクローカルアドレスは、ローカルネットワークプロトコルだけで使

	コマンドまたはアクション	目的
		用されます。非リンクローカルグループには、常にレポートが送信されます。デフォルトでは、リンクローカルグループにレポートは送信されません。
ステップ 19	<b>[no] ip igmp report-policy <i>policy-name</i></b> 例： apic1(config-tenant-interface)# ip igmp report-policy MyReportPolicy	ルートマップポリシーに基づく、IGMP レポートのアクセスポリシーを設定します。
ステップ 20	<b>[no] ip igmp robustness-variable <i>value</i></b> 例： apic1(config-tenant-interface)# ip igmp robustness-variable 2	輻輳ネットワーク上のパケット損失を補うようにロバストネス変数を設定します。ロバストネス値は、IGMP ソフトウェアがメッセージ送信回数を確認するために使用されます。ネットワークのパケット損失が多い場合は、この値を大きくします。指定できる範囲は 1～7 です。デフォルトは 2 です。
ステップ 21	<b>[no] ip igmp snooping</b> 例： apic1(config-tenant-interface)# ip igmp snooping	インターフェイスに対して IGMP スヌーピングを有効にします。
ステップ 22	<b>[no] ip igmp snooping fast-leave</b> 例： apic1(config-tenant-interface)# ip igmp snooping fast-leave	ソフトウェアが IGMP Leave レポートを受信した場合に、IGMP クエリーメッセージを送信することなく、グループステートを解除できるようにします。このパラメータは、IGMPv2 ホストに関して、各ポート上のホストが 1 つしか存在しない場合に使用されます。
ステップ 23	<b>[no] ip igmp snooping last-member-query-interval</b> 例： apic1(config-tenant-interface)# ip igmp snooping last-member-query-interval 5	時間間隔を秒単位で設定します。この時間が経過しても IGMP クエリーメッセージにホストが 1 つも応答しない場合は、関連するポートからこのグループが削除されます。値の範囲は 1～25 秒です。デフォルトは 5 秒です。
ステップ 24	<b>[no] ip igmp snooping policy <i>policy-name</i></b> 例： apic1(config-tenant-interface)# ip igmp snooping policy MySnoopingPolicy	IGMP スヌーピングポリシーとブリッジドメインを関連付けます。

	コマンドまたはアクション	目的
ステップ 25	<b>[no] ip igmp snooping querier</b> 例 : <pre>apicl(config-tenant-interface) # ip igmp snooping querier</pre>	IP IGMP スヌーピング クエリアを有効にして、IP マルチキャストトラフィックを受信するホストから IGMP レポートメッセージをトリガーする IGMP クエリーを定期的送信します。IGMP スヌーピングはこれらの IGMP レポートを待ち受けて、適切な転送を確立します。
ステップ 26	<b>[no] ip igmp snooping query-interval seconds</b> 例 : <pre>apicl(config-tenant-interface) # ip igmp snooping query-interval 125</pre>	マルチキャストトラフィックをルーティングする必要がないため、PIM をイネーブルにしていない場合に、スヌーピングクエリーインターバルを設定します。指定できる範囲は 1～18000 秒です。デフォルト値は 125 秒です。
ステップ 27	<b>[no] ip igmp snooping query-max-response-time seconds</b> 例 : <pre>apicl(config-tenant-interface) # ip igmp snooping query-max-response-time 10</pre>	マルチキャストトラフィックをルーティングする必要がないため PIM を無効にしている場合に対して、クエリーメッセージのスヌーピング最大応答時間を設定します。値の範囲は 1～25 秒です。デフォルトは 10 秒です。
ステップ 28	<b>[no] ip igmp snooping startup-query-count count</b> 例 : <pre>apicl(config-tenant-interface) # ip igmp snooping startup-query-count 5</pre>	マルチキャストトラフィックをルーティングする必要がないため、PIM をイネーブルにしていない場合に、起動時に送信されるクエリー数に対してスヌーピングを設定します。範囲は 1～10 クエリーです。デフォルトは 5 クエリーです。
ステップ 29	<b>[no] ip igmp snooping startup-query-interval seconds</b> 例 : <pre>apicl(config-tenant-interface) # ip igmp snooping startup-query-interval 15000</pre>	マルチキャストトラフィックをルーティングする必要がないため、PIM をイネーブルにしていない場合に、起動時のスヌーピングクエリーインターバルを設定します。指定できる範囲は 1～18000 秒です。デフォルト値は 15000 秒です。
ステップ 30	<b>[no] ip igmp startup-query-count count</b> 例 : <pre>apicl(config-tenant-interface) # ip igmp startup-query-count 2</pre>	スタートアップクエリーインターバル中に送信される起動時のクエリー数を設定します。範囲は 1～10 クエリーです。デフォルトは 2 クエリーです。

	コマンドまたはアクション	目的
ステップ 31	<p><b>[no] ip igmp startup-query-interval seconds</b></p> <p>例 :</p> <pre>apic1(config-tenant-interface)# ip igmp startup-query-interval 31</pre>	<p>ソフトウェアの起動時に使用されるクエリーインターバルを設定します。デフォルトでは、ソフトウェアができるだけ迅速にグループステートを確立できるように、このインターバルはクエリーインターバルより短く設定されています。指定できる範囲は 1 ~ 18000 秒です。デフォルト値は 260 秒です。デフォルト値は 31 秒です。</p>
ステップ 32	<p><b>[no] ip igmp state-limit max-states [reserved route-map-name [max-reserved-gsg-entries]]</b></p> <p>例 :</p> <pre>apic1(config-tenant-interface)# ip igmp state-limit 100000 reserved myReservedPolicy 40000</pre>	<p>IGMP メンバーシップレポート (IGMP 加入) の結果として作成される mroute ステートの数に対するインターフェイスごとの制限を設定します。許可される状態の範囲は 1 ~ 4294967295 です。オプションで、ポリシーマップに指定されるルート用に予約する状態エントリの数を指定することができ、また、インターフェイスで許可される最大予約済み (*, G) と (S, G) エントリを指定できます。予約済み状態の数は、許可される最大状態数以下である必要があります。範囲は 1 ~ 4294967295 です。</p>
ステップ 33	<p><b>[no] ip igmp static-oif route-map route-map-name</b></p> <p>例 :</p> <pre>apic1(config-tenant-interface)# ip igmp static-oif route-map MyOifMap</pre>	<p>マルチキャストグループを発信インターフェイス (OIF) に静的にバインドし、デバイスハードウェアで処理します。ルートマップは、この機能を適用するグループプレフィックスを定義します。</p>
ステップ 34	<p><b>[no] ip igmp version {v1   v2   v3}</b></p> <p>例 :</p> <pre>apic1(config-tenant-interface)# ip igmp version v3</pre>	<p>インターフェイスの IGMP バージョン番号を設定します。デフォルトのバージョンは v2 です。</p>
ステップ 35	<p><b>exit</b></p> <p>例 :</p> <pre>apic1(config-tenant-interface)# exit</pre>	<p>テナントコンフィギュレーションモードに戻ります。</p>

### 次のタスク

テナントに L3 Out を設定し、PIM を有効にしてリーフインターフェイスを設定します。

## レイヤ3マルチキャストのL3 Outの設定

### 始める前に

- テナント VRF の PIM オプションを設定します。
- テナント VRF の IGMP 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	<b>configure</b> 例： apicl# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： apicl(config)# <b>tenant exampleCorp</b>	設定するテナントを指定します。
ステップ 3	<b>l3out l3out-name</b> 例： apicl(config-tenant)# <b>l3out exampleCorp_l3out</b>	テナントにL3 Outのインターフェイスを設定します。
ステップ 4	<b>ip pim</b> 例： apicl(config-tenant-l3out)# <b>ip pim</b>	インターフェイスでPIMをイネーブルにします。
ステップ 5	<b>exit</b> 例： apicl(config-tenant-l3out)#	テナントコンフィギュレーションモードに戻ります。
ステップ 6	<b>exit</b> 例： apicl(config-tenant)# <b>exit</b>	グローバル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>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	リーフ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>interface ethernet slot/port</b> 例： apicl(config-leaf)# <b>interface ethernet 1/3</b>	設定するインターフェイスを指定します。

	コマンドまたはアクション	目的
ステップ 9	<b>[no] ip igmp allow-v3-asm</b> 例： apic1(config-leaf-if)# ip igmp allow-v3-asm	Any Source Multicast (ASM) グループに関する IGMPv3 レポートの送信元アドレスのフィルタリングを許可します。
ステップ 10	<b>[no] ip igmp fast-leave</b> 例： apic1(config-leaf-if)# ip igmp fast-leave	IP IGMP スヌーピングの高速脱退処理を有効にします。この機能は、IGMPv2 プロトコルのホストレポート抑制メカニズムのために明示的に追跡できない IGMPv2 ホストをサポートします。高速脱退が有効になっている場合、IGMP ソフトウェアは、各ポートに接続されたホストが 1 つだけであると見なします。
ステップ 11	<b>[no] ip igmp group-timeout seconds</b> 例： apic1(config-leaf-if)# ip igmp group-timeout 260	IGMPv2 のグループメンバーシップタイムアウトを設定します。値の範囲は 3 ~ 65535 秒です。デフォルト値は 260 秒です。
ステップ 12	<b>[no] ip igmp inherit interface-policy policy-name</b> 例： apic1(config-leaf-if)# ip igmp inherit interface-policy MyIfPolicy	このインターフェイスに IGMP インターフェイス ポリシーを関連付けます。
ステップ 13	<b>[no] ip igmp join-group route-map route-map-name</b> 例： apic1(config-leaf-if)# ip igmp join-group route-map MyGroupsRMap	1 つ以上のマルチキャスト グループをインターフェイスに静的にバインドします。ルート マップ ポリシーは、グループのプレフィクス、グループの範囲、および送信元プレフィクスのリストを表示します。
ステップ 14	<b>[no] ip igmp last-member-query-count count</b> 例： apic1(config-leaf-if)# ip igmp last-member-query-count 2	ホストの Leave メッセージを受信してから、IGMP クエリーが送信される回数を設定します。範囲は 1 ~ 5 クエリーです。デフォルトは 2 クエリーです。
ステップ 15	<b>[no] ip igmp last-member-query-response-time seconds</b> 例： apic1(config-leaf-if)# ip igmp last-member-query-response-time 1	メンバーシップレポートを送信してから、ソフトウェアがグループステートを解除するまでのクエリーインターバルを設定します。値の範囲は 1 ~ 25 秒です。デフォルト値は 1 秒です。

	コマンドまたはアクション	目的
ステップ 16	<b>[no] ip igmp querier-timeout <i>seconds</i></b> 例： <pre>apicl(config-leaf-if)# ip igmp querier-timeout 255</pre>	前クエリアがクエリーを停止してから、自身がクエリアとして処理を引き継ぐまで、ソフトウェアが待機する秒数を設定します。指定できる範囲は1～65535秒です。デフォルト値は255秒です。
ステップ 17	<b>[no] ip igmp query-interval <i>seconds</i></b> 例： <pre>apicl(config-leaf-if)# ip igmp query-interval 125</pre>	IGMP ホストクエリーメッセージの送信頻度を設定します。大きな値を設定すると、ソフトウェアによるIGMPクエリーの送信頻度が低くなるため、ネットワーク上のIGMPメッセージ数を調整できます。指定できる範囲は1～18000秒です。デフォルト値は125秒です。
ステップ 18	<b>[no] ip igmp query-max-response-time <i>seconds</i></b> 例： <pre>apicl(config-leaf-if)# ip igmp query-max-response-time 10</pre>	IGMPクエリーでアドバタイズされる応答時間を設定します。大きな値を設定すると、ホストの応答時間が延長されるため、ネットワークのIGMPメッセージのバースト性を調整できます。この値は、クエリーインターバルよりも短く設定する必要があります。値の範囲は1～25秒です。デフォルトは10秒です。
ステップ 19	<b>[no] ip igmp report-link-local-groups</b> 例： <pre>apicl(config-leaf-if)# ip igmp report-link-local-groups</pre>	224.0.0.0/24に含まれるグループに対して、レポート送信をイネーブルにします。リンクローカルアドレスは、ローカルネットワークプロトコルだけで使用されます。非リンクローカルグループには、常にレポートが送信されます。デフォルトでは、リンクローカルグループにレポートは送信されません。
ステップ 20	<b>[no] ip igmp report-policy <i>policy-name</i></b> 例： <pre>apicl(config-leaf-if)# ip igmp report-policy MyReportPolicy</pre>	ルートマップポリシーに基づく、IGMPレポートのアクセスポリシーを設定します。
ステップ 21	<b>[no] ip igmp robustness-variable <i>value</i></b> 例： <pre>apicl(config-leaf-if)# ip igmp robustness-variable 2</pre>	輻輳ネットワーク上のパケット損失を補うようにロバストネス変数を設定します。ロバストネス値は、IGMPソフトウェアがメッセージ送信回数を確認

	コマンドまたはアクション	目的
		するために使用されます。ネットワークの packets 損失が多い場合は、この値を大きくします。指定できる範囲は 1～7 です。デフォルトは 2 です。
ステップ 22	<b>[no] ip igmp startup-query-count <i>count</i></b> 例 : <pre>apic1(config-leaf-if)# ip igmp startup-query-count 2</pre>	スタートアップクエリーインターバル中に送信される起動時のクエリー数を設定します。範囲は 1～10 クエリーです。デフォルトは 2 クエリーです。
ステップ 23	<b>[no] ip igmp startup-query-interval <i>seconds</i></b> 例 : <pre>apic1(config-leaf-if)# ip igmp startup-query-interval 31</pre>	ソフトウェアの起動時に使用されるクエリーインターバルを設定します。デフォルトでは、ソフトウェアができるだけ迅速にグループステートを確立できるように、このインターバルはクエリーインターバルより短く設定されています。指定できる範囲は 1～18000 秒です。デフォルト値は 260 秒です。デフォルト値は 31 秒です。
ステップ 24	<b>[no] ip igmp state-limit <i>max-states</i> [<i>reserved route-map-name</i> [<i>max-reserved-gsg-entries</i>]]</b> 例 : <pre>apic1(config-leaf-if)# ip igmp state-limit 100000 reserved myReservedPolicy 40000</pre>	IGMP メンバーシップレポート (IGMP 加入) の結果として作成される mroute ステートの数に対するインターフェイスごとの制限を設定します。許可される状態の範囲は 1～4294967295 です。オプションで、ポリシーマップに指定されるルート用に予約する状態エントリの数を指定することができ、また、インターフェイスで許可される最大予約済み (*, G) と (S, G) エントリを指定できます。予約済み状態の数は、許可される最大状態数以下である必要があります。範囲は 1～4294967295 です。
ステップ 25	<b>[no] ip igmp static-oif route-map <i>route-map-name</i></b> 例 : <pre>apic1(config-leaf-if)# ip igmp static-oif route-map MyOifMap</pre>	マルチキャストグループを発信インターフェイス (OIF) に静的にバインドし、デバイスハードウェアで処理します。ルートマップは、この機能を適用するグループプレフィックスを定義します。
ステップ 26	<b>[no] ip igmp version {v1   v2   v3}</b> 例 : <pre>apic1(config-leaf-if)# ip igmp version v3</pre>	インターフェイスの IGMP バージョン番号を設定します。デフォルトのバージョンは v2 です。

	コマンドまたはアクション	目的
ステップ 27	<b>exit</b> 例： apicl(config-leaf-if)# <b>exit</b>	テナントコンフィギュレーションモードに戻ります。

## 例：レイヤ3マルチキャストの設定

```
# CONFIGURE PIM OPTIONS ON A TENANT VRF

apicl# configure
apicl(config)# tenant exampleCorp
apicl(config-tenant)# vrf context exampleCorp_vrf1
apicl(config-tenant-vrf)# ip pim
apicl(config-tenant-vrf)# ip pim fast-convergence
apicl(config-tenant-vrf)# ip pim bsr forward

# ENABLE AND CONFIGURE IGMP ON THE TENANT VRF AND BRIDGE DOMAIN

apicl(config-tenant-vrf)# ip igmp
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# interface bridge-domain exampleCorp_bd
apicl(config-tenant-interface)# ip multicast
apicl(config-tenant-interface)# ip igmp allow-v3-asm
apicl(config-tenant-interface)# ip igmp fast-leave
apicl(config-tenant-interface)# exit

# CREATE AN L3OUT AND CONFIGURE PIM

apicl(config-tenant)# l3out exampleCorp_l3out
apicl(config-tenant-l3out)# ip pim
apicl(config-tenant-l3out)# exit
apicl(config-tenant)# exit

# CONFIGURE AN EXTERNAL INTERFACE AND CONFIGURE IGMP ON THE INTERFACE

apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/125
apicl(config-leaf-if)# ip igmp fast-leave
apicl(config-leaf-if)# ip-igmp join-group
```

## 外部 L3 EPG の設定

外部 L3 EPG はテナント VRF 下に分類されます。CLI で、外部 L3 EPG はテナントモードで定義され、個々のノードに展開されます。VRF のすべてのノードではなく一部のノードに外部 L3 EPG を配置することができます。

外部 L3 EPG はそれぞれ複数のコントラクトのプロデューサまたはコンシューマになることができ、外部 L3 EPG にはそれぞれファブリック内の DSCP マーキングおよびキューイングプライオリティに関する独自の QoS ポリシーがあります。

## 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： apic1(config)# <b>tenant exampleCorp</b>	テナント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>external-l3 epg epg-name</b> 例： apic1(config-tenant)# <b>external-l3 epg epgExtern1</b>	外部 L3 EPG 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>vrf member vrf-name</b> 例： apic1(config-tenant-l3ext-epg)# <b>vrf member v1</b>	EPG を VRF に関連付けます。
ステップ 5	<b>match {ip   ipv6} ip-address/masklength</b> 例： apic1(config-tenant-l3ext-epg)# <b>match ip 192.0.20.0/24</b> apic1(config-tenant-l3ext-epg)# <b>match ipv6 2001::1/64</b>	サブネットに一致するルールを作成します。
ステップ 6	<b>set qos-class class</b> 例： apic1(config-tenant-l3ext-epg)# <b>set qos-class level1</b>	EPG の QoS レベルを指定します。
ステップ 7	<b>set dscp dscp-value</b> 例： apic1(config-tenant-l3ext-epg)# <b>set dscp af31</b>	EPG の DSCP 値を指定します。
ステップ 8	<b>contract consumer contract-name</b> 例： apic1(config-tenant-l3ext-epg)# <b>contract consumer cConsumer1</b>	EPG のコンシューマコントラクトを指定します。
ステップ 9	<b>contract provider contract-name</b> 例：	EPG のプロバイダーコントラクトを指定します。

	コマンドまたはアクション	目的
	<code>apicl(config-tenant-l3ext-epg)# contract provider cProvider1</code>	
ステップ 10	<b>contract deny contract-name</b> 例： <code>apicl(config-tenant-l3ext-epg)# contract deny cDeny1</code>	EPG の拒否コントラクトを指定します。
ステップ 11	<b>exit</b> 例： <code>apicl(config-tenant-l3ext-epg)# exit</code>	
ステップ 12	<b>exit</b> 例： <code>apicl(config-tenant)# exit</code>	
ステップ 13	<b>leaf node-id</b> 例： <code>apicl(config)# leaf 101</code>	設定するリーフを指定します。
ステップ 14	<b>vrf context tenant tenant-name vrf vrf-name</b> 例： <code>apicl(config-leaf)# vrf context tenant exampleCorp vrf v1</code>	ノードのテナント VRF を設定します。
ステップ 15	<b>external-l3 epg epg-name</b> 例： <code>apicl(config-leaf-vrf)# external-l3 epg epgExtern1</code>	VRF の外部レイヤ 3 EPG を関連付けます。

## 例

次に、外部レイヤ 3 EPG を設定し、リーフに EPG を展開する例を示します。

```
apicl# configure
apicl(config)# tenant exampleCorp

# CONFIGURE EXTERNAL L3 EPG
apicl(config-tenant)# external-l3 epg epgExtern1
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# match ip 192.0.20.0/24
apicl(config-tenant-l3ext-epg)# match ipv6 2001::1/64
apicl(config-tenant-l3ext-epg)# set qos-class level1
apicl(config-tenant-l3ext-epg)# set dscp af31
apicl(config-tenant-l3ext-epg)# contract consumer cConsumer1
apicl(config-tenant-l3ext-epg)# contract provider cProvider1
apicl(config-tenant-l3ext-epg)# contract deny cDeny1
```

```

apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit

# DEPLOY EXTERNAL L3 EPG ON A LEAF
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf)# external-l3 epg epgExtern1

```

## 名前付きモードを使用したレイヤ 3 外部接続の設定

### 名前付き L3Out の作成

#### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： apic1(config)# <b>tenant exampleCorp</b>	テナント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 context vrf-name</b> 例： apic1(config-tenant)# <b>vrf context v1</b>	テナントを VRF に関連付けます。
ステップ 4	<b>l3out l3out-name</b> 例： apic1(config-tenant)# <b>l3out out1</b>	名前付き L3Out を作成します。
ステップ 5	<b>vrf member vrf-name</b> 例： apic1(config-tenant-l3out)# <b>vrf member v1</b>	L3Out をテナント VRF と関連付けます。
ステップ 6	<b>exit</b> 例： apic1(config-tenant-l3out)# <b>exit</b>	テナント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>exit</b> 例： apic1(config-tenant)# <b>exit</b>	グローバルコンフィギュレーションモードに戻ります。

	コマンドまたはアクション	目的
ステップ 8	<b>leaf node-id</b> 例： apicl(config)# leaf 101	ノード
ステップ 9	<b>vrf context tenant tenant-name vrf vrf-name l3out l3out-name</b> 例： apicl(config-leaf)# vrf context tenant exampleCorp vrf v1 l3out out1	ノードのテナント VRF を設定します。
ステップ 10	必須: <b>[no] router-id ipv4-address</b> 例： apicl(config-leaf-vrf)# router-id 1.2.3.4	VRF で実行されているルーティングプロトコルに対してルータ ID を割り当てます。
ステップ 11	<b>[no] {ip   ipv6} route ip-prefix/masklen next-hop-address [preferred]</b> 例： apicl(config-leaf-vrf)# ip route 21.1.1.1/32 32.1.1.1 apicl(config-leaf-vrf)# ipv6 route 5001::1/128 6002::1	VRF の静的ルートの情報を設定します。

## 例

次に、テナントの下に名前付き L3Out を作成し、テナント VRF に割り当て、境界リーフスイッチに導入する例を示します。

```
apicl# configure
apicl(config)# tenant exampleCorp
apicl(config-tenant)# vrf context v1
apicl(config-tenant)# l3out out1
apicl(config-tenant-l3out)# vrf member v1
apicl(config-tenant-l3out)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf v1 l3out out1
apicl(config-leaf-vrf)# router-id 1.2.3.4
apicl(config-leaf-vrf)# ip route 21.1.1.1/32 32.1.1.1
```

## 次のタスク

名前付き L3Out のレイヤ 3 インターフェイスを設定します。

## 名前付き L3Out のレイヤ 3 インターフェイスの設定

この手順では、名前付き L3Out にレイヤ 3 ポート インターフェイスを設定する方法を示します。例では、名前付き L3Out にサブインターフェイスまたは SVI を設定する方法を示します。

- A given interface can be added to multiple L3Outs by providing multiple L3Out names after the **l3out** keyword.
- An SVI can be configured using the **switchport trunk allowed vlan** command under any of the following interface types:
  - **interface Ethernet**
  - **interface port-channel**
  - **interface vpc**

始める前に

名前付き L3Out を作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーション モードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>interface type</b> 例： apic1(config-leaf)# <b>interface eth 1/20</b>	外部インターフェイスのポートを指定します。
ステップ 4	<b>no switchport</b> 例： apic1(config-leaf-if)# <b>no switchport</b>	インターフェイスをレイヤ 3 インターフェイスとして設定し、設定オプションでレイヤ 3 コマンドを公開します。
ステップ 5	<b>vrf member tenant tenant-name vrf vrf-name l3out l3out-name</b> 例： apic1(config-leaf-if)# <b>vrf member tenant exampleCorp vrf v1 l3out out1</b>	テナント VRF にインターフェイスを接続します。

	コマンドまたはアクション	目的
ステップ 6	<p><b>[no] {ip   ipv6} address ip-prefix/masklen [eui64] [secondary] [preferred]</b></p> <p>例 :</p> <pre>apicl(config-leaf-if)# ip address 10.1.1.1/24 apicl(config-leaf-if)# ipv6 address 2001::1/64 preferred</pre>	<p>インターフェイスに IP アドレスを設定します。指定されたアドレスは次のいずれかとして宣言できます。</p> <ul style="list-style-type: none"> <li>• <b>preferred</b>—インターフェイスからトラフィックのデフォルト送信元アドレス。</li> <li>• <b>secondary</b>—インターフェイスのセカンダリアドレス。</li> </ul> <p>With the optional <b>eui64</b> keyword, the host can assign itself a 64-bit Extended Unique Identifier (EUI).</p> <p>In this mode, you can also configure <b>ipv6 link-local</b>, <b>mac address</b>, <b>mtu</b>, and other layer 3 properties on the interface.</p>

## 例

次に、名前付き L3Out にレイヤ3 ポートを割り当てる例を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# interface eth 1/20
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1
apicl(config-leaf-if)# ip address 10.1.1.1/24
apicl(config-leaf-if)# ipv6 address 2001::1/64 preferred
```

次に、名前付き L3Out にレイヤ3 サブインターフェイスを割り当てる例を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# interface eth 1/5
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vlan-domain member d1
apicl(config-leaf-if)# exit
apicl(config-leaf)# interface ethernet 1/5.1000
apicl(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1
apicl(config-leaf-if)# ip address 10.1.1.1/24
apicl(config-leaf-if)# ipv6 address 2001::1/64 preferred
```

次に、名前付き L3Out にレイヤ3 SVI を割り当てる例を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# interface vlan 200
apicl(config-leaf-if)# vrf member tenant exampleCorp vrf v1
```

```

apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/4
apic1(config-leaf-if)# vlan-domain member d1
apic1(config-leaf-if)# switchport trunk allowed vlan 200 tenant t1 external-svi l3out
out1

```

## 名前付き L3Out のルート マップの設定

- ルートマップはリーフ、VRF モードで設定します。
- 次のルートマップは、すべての名前付き L3Out に対して作成されます。
  - エクスポート：L3Out で有効なルーティングプロトコルからアドバタイズされるルートのルートマップ。By default, no routes are exported out until you explicitly enable them in the route-map through one or more of **match bridge-domain**, **match prefix-list** and **match community-list** statements.
  - インポート：L3Out のルーティングプロトコルにインポートされるルートのルートマップ。デフォルトでは、すべてのルートがインポートされます。You can control specific routes to be imported by using one or more **match prefix-list** or **match community-list** statements.
  - 共有：ルートと、この VRF からコントラクトの関連付けがあるその他の VRF へのルートをリークするために使用されるコントラクトプロバイダー/コンシューマ ポリシーを含むルートマップ。

These route-maps are created when you associate a leaf to the L3Out through the **vrf context tenant tenant-name vrf vrf-name l3out l3out-name** command.

- 名前付き L3Out 下のルートマップの範囲は常にグローバルであり、名前付き L3Out が展開されているすべてのノードに適用できます。
- All commands under the route-map (such as **match prefix-list**, **match community-list**, **match bridge-domain**) are the same as the route-map configuration for the Basic Mode discussed in the previous sections.

### 始める前に

名前付き L3Out を作成します。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。

	コマンドまたはアクション	目的
ステップ 2	<b>leaf node-id</b> 例： apicl(config)# leaf 101	設定するリーフを指定します。
ステップ 3	<b>[no] vrf context tenant tenant-name vrf vrf-name l3out l3out-name</b> 例： apicl(config-leaf)# vrf context tenant exampleCorp vrf v1 l3out out1	ノードのテナント VRF を設定します。
ステップ 4	必須: <b>[no] route-map name</b> 例： apicl(config-leaf-vrf)# route-map out1_in	ルートマップを作成し、ルートマップコンフィギュレーションを開始します。これがインポートルートマップになります。
ステップ 5	必須: <b>[no] ip prefix-list list-name permit prefix/masklen [le {32   128}]</b> 例： apicl(config-leaf-vrf-route-map)# ip prefix-list p1 permit 15.1.1.0/24	ルートマップにプレフィックスリストを作成します。
ステップ 6	必須: <b>[no] match prefix-list list-name</b> 例： apicl(config-leaf-vrf-route-map)# match prefix-list p1	作成済みのプレフィックスリストと照合し、一致モードを開始して、プレフィックスリストのルート制御プロファイルを設定します。
ステップ 7	必須: <b>exit</b> 例： apicl(config-leaf-vrf-route-map-match)# exit	ルートマップコンフィギュレーションモードに戻ります。
ステップ 8	必須: <b>exit</b> 例： apicl(config-leaf-vrf-route-map)# exit	リーフ VRF コンフィギュレーションモードに戻ります。
ステップ 9	必須: <b>[no] route-map name</b> 例： apicl(config-leaf-vrf)# route-map out1_out	ルートマップを作成し、ルートマップコンフィギュレーションを開始します。これがエクスポートルートマップになります。
ステップ 10	必須: <b>[no] ip prefix-list list-name permit prefix/masklen [le {32   128}]</b> 例：	ルートマップにプレフィックスリストを作成します。

	コマンドまたはアクション	目的
	<code>apicl(config-leaf-vrf-route-map)# ip prefix-list p2 permit 16.1.1.0/24</code>	
ステップ 11	必須: <code>[no] match prefix-list list-name</code> 例: <code>apicl(config-leaf-vrf-route-map)# match prefix-list p2</code>	作成済みのプレフィックスリストと照合し、一致モードを開始して、プレフィックスリストのルート制御プロファイルを設定します。
ステップ 12	必須: <code>set tag name</code> 例: <code>apicl(config-leaf-vrf-route-map-match)# set tag 100</code>	タグ値を設定します。name パラメータは符号なし整数です。
ステップ 13	必須: <code>exit</code> 例: <code>apicl(config-leaf-vrf-route-map-match)# exit</code>	ルートマップコンフィギュレーションモードに戻ります。
ステップ 14	必須: <code>[no] match bridge-domain list-name</code> 例: <code>apicl(config-leaf-vrf-route-map)# match bridge-domain bdl</code>	プロトコルによってパブリック サブ ネットをエクスポートするため、ブリッジドメインと照合します。
ステップ 15	必須: <code>exit</code> 例: <code>apicl(config-leaf-vrf-route-map-match)# exit</code>	ルートマップコンフィギュレーションモードに戻ります。
ステップ 16	必須: <code>[no] route-map name</code> 例: <code>apicl(config-leaf-vrf)# route-map out1_shared</code>	ルートマップを作成し、ルートマップコンフィギュレーションを開始します。これが共有ルートマップになります。
ステップ 17	必須: <code>[no] ip prefix-list list-name permit prefix/masklen [le {32   128}]</code> 例: <code>apicl(config-leaf-vrf-route-map)# ip prefix-list p3 permit 16.10.1.0/24</code>	ルートマップにプレフィックスリストを作成します。
ステップ 18	必須: <code>[no] match prefix-list list-name</code> 例: <code>apicl(config-leaf-vrf-route-map)# match prefix-list p3</code>	作成済みのプレフィックスリストと照合し、一致モードを開始して、プレフィックスリストのルート制御プロファイルを設定します。

	コマンドまたはアクション	目的
ステップ 19	必須: <b>contract provider name</b> 例: <pre>apicl(config-leaf-vrf-route-map-match)# <b>contract provider default</b></pre>	この VRF から (このプレフィックスリストと一致する) ルートをリークするために必要なコントラクトを追加します。

### 例

次に、名前付き L3Out のルート マップを設定する例を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf v1 l3out out1

# CREATE IMPORT ROUTE-MAP
apicl(config-leaf-vrf)# route-map out1_in
apicl(config-leaf-vrf-route-map)# ip prefix-list p1 permit 15.1.1.0/24
apicl(config-leaf-vrf-route-map)# match prefix-list p1
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit

# CREATE EXPORT ROUTE-MAP
apicl(config-leaf-vrf)# route-map out1_out
apicl(config-leaf-vrf-route-map)# ip prefix-list p2 permit 16.1.1.0/24
apicl(config-leaf-vrf-route-map)# match prefix-list p2
apicl(config-leaf-vrf-route-map-match)# set tag 100
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# match bridge-domain bd1
apicl(config-leaf-vrf-route-map-match)# exit

# CREATE SHARED ROUTE-MAP
apicl(config-leaf-vrf)# route-map out1_shared
apicl(config-leaf-vrf-route-map)# ip prefix-list p3 permit 16.10.1.0/24
apicl(config-leaf-vrf-route-map)# match prefix-list p3
apicl(config-leaf-vrf-route-map-match)# contract provider default
```

## 名前付き L3Out のルーティング プロトコルの設定

### 名前付き L3Out の BGP の設定

- All commands under the BGP neighbor with the exception of **route-map** are identical to those in the Basic Mode of L3Out configuration. BGP テンプレート設定およびテンプレートの継承は、基本モードと同じです。
- L3Out 設定の名前付きモードでは、**route-map** は L3Out レベルで適用されます。L3Out とネイバーを関連付けることにより、**route-map** は L3Out のプロトコルに自動的に適用されます。したがって、**route-map** オプションは適用できず、BGP ネイバーでは使用できません。同じ理由で、**route-map** オプションは OSPF エリアで使用できず、**distribute-list EIGRP** オプションはインターフェイスで使用できません。

## 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apicl# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>router bgp asn-number</b> 例： apicl(config-leaf)# <b>router bgp 100</b>	BGP ポリシー設定を入力します。
ステップ 4	<b>vrf member tenant tenant-name vrf vrf-name</b> 例： apicl(config-bgp)# <b>vrf member tenant exampleCorp vrf v100</b>	後続のポリシー設定モードコマンドと関連付ける VRF インスタンスを指定します。
ステップ 5	<b>neighbor ip-address[/masklength] l3out l3out-name</b> 例： apicl(config-leaf-bgp-vrf)# <b>neighbor 192.0.2.229 l3out out1</b>	ネイバーの IP アドレスを指定します。
ステップ 6	<b>remote-as asn</b> 例： apicl(config-leaf-bgp-vrf-neighbor)# <b>remote-as 300</b>	ネイバーの自律システム番号を指定します。
ステップ 7	<b>allow-self-as-count count</b> 例： apicl(config-leaf-bgp-vrf-neighbor)# <b>allow-self-as-count 5</b>	指定できる数は 1 ~ 10 です。デフォルトは 3 です。
ステップ 8	<b>update-source ethernet interface-range</b> 例： apicl(config-leaf-bgp-vrf-neighbor)# <b>update-source ethernet 1/3</b>	BGP パケットの送信元 IP を、ループバック、物理、サブインターフェイス、または SVI インターフェイスのいずれかに更新します。

## 例

次に、名前付き L3Out の BGP ルーティング プロトコルを設定する例を示します。

```

apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# router bgp 100
apicl(config-leaf-bgp)# vrf member tenant exampleCorp vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 192.0.2.229 l3out out1
apicl(config-leaf-bgp-vrf-neighbor)# remote-as 300
apicl(config-leaf-bgp-vrf-neighbor)# allow-self-as-count 5
apicl(config-leaf-bgp-vrf-neighbor)# update-source ethernet 1/3

```

## 名前付き L3Out の OSPF の設定

All commands under the **router ospf default** command, with the exception of **area area-id route-map map-name out**, are identical to those in the Basic Mode of L3Out configuration. インターフェイス下の OSPF コマンドおよび OSPF テンプレート継承コマンドも、基本モードと同じです。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apicl# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apicl(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>router ospf default</b> 例： apicl(config-leaf)# <b>router ospf default</b>	OSPF ルーティングプロセスを作成し、OSPF ポリシー設定を開始します。
ステップ 4	<b>vrf member tenant tenant-name vrf vrf-name</b> 例： apicl(config-leaf-ospf)# <b>vrf member tenant exampleCorp vrf v100</b>	OSPF セッションで VRF を有効にします。
ステップ 5	<b>area</b> エリア <i>id</i> <b>l3out l3out</b> : 名前 例： apicl(config-leaf-ospf-vrf)# <b>area 0.0.0.1 l3out out1</b>	L3Out で OSPF を有効にします。
ステップ 6	<b>area area-id loopback loopback-address</b> 例： apicl(config-leaf-ospf-vrf)# <b>area 0.0.0.1 loopback 192.0.20.11</b>	OSPF が BGP の接続プロトコルとして使用される場合、OSPF は BGP セッションの送信元として使用されるループバックアドレスをアドバタイズしま

	コマンドまたはアクション	目的
		す。ループバック ID ではなくループバック IP アドレスが使用されることに注意してください。この場合、OSPF に依存している BGP セッションは <code>update-source</code> コマンドで同じループバック IP アドレスを使用します。
ステップ 7	<b>area <i>area-id</i> nssa [no-redistribution] [default-information-originate]</b>  例： <code>apic1(config-leaf-ospf-vrf)# area 0.0.0.1 nssa</code>	Not-So-Stubby Area (NSSA) を定義します。
ステップ 8	<b>exit</b>  例： <code>apic1(config-leaf-ospf-vrf)# exit</code>	OSPF コンフィギュレーションモードに戻ります。
ステップ 9	<b>exit</b>  例： <code>apic1(config-leaf-ospf)# exit</code>	リーフコンフィギュレーションモードに戻ります。
ステップ 10	<b>interface <i>type</i></b>  例： <code>apic1(config-leaf)# interface eth 1/20</code>	外部インターフェイスのポートを指定します。
ステップ 11	<b>vlan-domain member <i>domain-name</i></b>  例： <code>apic1(config-leaf-if)# vlan-domain member dom1</code>	インターフェイスに VLAN ドメインを割り当てます。The VLAN domain must have already been created using the <b>vlan-domain</b> command in the global configuration mode.
ステップ 12	<b>no switchport</b>  例： <code>apic1(config-leaf-if)# no switchport</code>	インターフェイスをレイヤ 3 インターフェイスとして設定し、設定オプションでレイヤ 3 コマンドを公開します。
ステップ 13	<b>vrf member tenant <i>tenant-name</i> vrf <i>vrf-name</i> l3out <i>l3out-name</i></b>  例： <code>apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1</code>	テナント VRF にインターフェイスを接続します。
ステップ 14	<b>[no] {ip   ipv6} address <i>ip-prefix/masklen</i> [eui64] [secondary] [preferred]</b>  例：	インターフェイスに IP アドレスを設定します。指定されたアドレスは次のいずれかとして宣言できます。

	コマンドまたはアクション	目的
	<pre>apicl(config-leaf-if)# ip address 10.1.1.1/24 apicl(config-leaf-if)# ipv6 address 2001::1/64 preferred</pre>	<ul style="list-style-type: none"> <li>• <b>preferred</b>—インターフェイスからトラフィックのデフォルト送信元アドレス。</li> <li>• <b>secondary</b>—インターフェイスのセカンダリ アドレス。</li> </ul> <p>With the optional <b>eui64</b> keyword, the host can assign itself a 64-bit Extended Unique Identifier (EUI).</p> <p>In this mode, you can also configure <b>ipv6 link-local</b>, <b>mac address</b>, <b>mtu</b>, and other layer 3 properties on the interface.</p>
ステップ 15	<pre>{ip   ipv6} router ospf default area area-id</pre> <p>例 :</p> <pre>apicl(config-leaf-if)# ip router ospf default area 0.0.0.1</pre>	OSPF ルーティングプロセスを作成し、OSPF ポリシー設定を開始します。

## 例

次に、名前付き L3Out の OSPF ルーティング プロトコルを設定する例を示します。

```
apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant exampleCorp vrf v1
apicl(config-leaf-ospf-vrf)# area 0.0.0.1 l3out out1
apicl(config-leaf-ospf-vrf)# area 0.0.0.1 loopback 192.0.20.11
apicl(config-leaf-ospf-vrf)# area 0.0.0.1 nssa
apicl(config-leaf-ospf-vrf)# exit
apicl(config-leaf-ospf)# exit
apicl(config-leaf)# interface eth 1/20
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1
apicl(config-leaf-if)# ip address 10.1.1.1/24
apicl(config-leaf-if)# ip router ospf default area 0.0.0.1
```

## 名前付き L3Out の EIGRP の設定

All EIGRP commands under vrf mode and interface mode, with the exception of **ip distribute-list**, are identical to those in the Basic Mode of L3Out configuration. This includes the EIGRP **template** and **inherit** commands. The **ip distribute-list** commands are not applicable to the Named Mode of L3Out configuration, as the route-maps are defined at the L3Out level and by associating an interface with the L3Out, the route-map distribute-list is automatically associated. For this reason, **ip distribute-list** is not available in the CLI as an option.

## 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 3	<b>router eigrp default</b> 例： apic1(config-leaf)# <b>router eigrp default</b>	EIGRP ポリシー設定を入力します。
ステップ 4	<b>vrf member tenant tenant-name vrf vrf-name</b> 例： apic1(config-eigrp)# <b>vrf member tenant exampleCorp vrf v100</b>	後続のコンフィギュレーションモードコマンドと関連付ける VRF インスタンスを指定します。
ステップ 5	<b>autonomous-systemasn l3out l3out: 名前</b> 例： apic1(config-eigrp-vrf)# <b>autonomous-system 500 l3out out1</b>	EIGRP の自律システム設定を入力します。
ステップ 6	<b>exit</b> 例： apic1(config-eigrp-vrf)# <b>exit</b>	EIGRP 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>exit</b> 例： apic1(config-eigrp)# <b>exit</b>	リーフ コンフィギュレーションモードに戻ります。
ステップ 8	<b>interface type</b> 例： apic1(config-leaf)# <b>interface eth 1/5</b>	外部インターフェイスのポートを指定します。
ステップ 9	<b>vlan-domain member domain-name</b> 例： apic1(config-leaf-if)# <b>vlan-domain member dom1</b>	インターフェイスに VLAN ドメインを割り当てます。The VLAN domain must have already been created using the <b>vlan-domain</b> command in the global configuration mode.

	コマンドまたはアクション	目的
ステップ 10	<b>no switchport</b> 例： apicl(config-leaf-if)# <b>no switchport</b>	インターフェイスをレイヤ 3 インターフェイスとして設定し、設定オプションでレイヤ 3 コマンドを公開します。
ステップ 11	<b>vrf member tenant tenant-name vrf vrf-name l3out l3out-name</b> 例： apicl(config-leaf-if)# <b>vrf member tenant exampleCorp vrf v1 l3out out1</b>	テナント VRF にインターフェイスを接続します。
ステップ 12	<b>[no] {ip   ipv6} address ip-prefix/masklen [eui64] [secondary] [preferred]</b> 例： apicl(config-leaf-if)# <b>ip address 10.1.1.1/24</b> apicl(config-leaf-if)# <b>ipv6 address 2001::1/64 preferred</b>	インターフェイスに IP アドレスを設定します。指定されたアドレスは次のいずれかとして宣言できます。  <ul style="list-style-type: none"> <li>• <b>preferred</b>—インターフェイスからトラフィックのデフォルト送信元アドレス。</li> <li>• <b>secondary</b>—インターフェイスのセカンダリ アドレス。</li> </ul> <p>With the optional <b>eui64</b> keyword, the host can assign itself a 64-bit Extended Unique Identifier (EUI).</p> <p>In this mode, you can also configure <b>ipv6 link-local, mac address, mtu,</b> and other layer 3 properties on the interface.</p>
ステップ 13	<b>{ip   ipv6} router eigrp default</b> 例： apicl(config-leaf-if)# <b>ip router eigrp default</b>	EIGRP ポリシーをデフォルトに設定します。

### 例

次に、名前付き L3Out の EIGRP ルーティング プロトコルを設定する例を示します。

```

apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# router eigrp default
apicl(config-eigrp)# vrf member tenant exampleCorp vrf v1
apicl(config-eigrp-vrf)# autonomous-system 500 l3out out1
apicl(config-eigrp-vrf)# exit
apicl(config-eigrp)# exit
apicl(config-leaf)# interface eth 1/5
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# no switchport

```

```

apic1(config-leaf-if)# vrf member tenant exampleCorp vrf v1 l3out out1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ip router eigrp default

```

## 名前付き L3Out の外部 L3 EPG の設定

外部 L3 EPG はテナント VRF 下に分類されます。

All commands under the **config-tenant-l3ext-epg** mode are identical to those in the Basic Mode of L3Out configuration with the following differences:

- VRF は自動的に EPG と関連付けられます。L3Out は VRF に関連付けられ、EPG は L3Out に関連付けられます。
- The **external-l3 epg** command is not available under the leaf **vrf context tenant tenant-name vrf vrf-name l3out l3out-name** command, as this configuration is not applicable for Named L3Outs. The **external-l3 epg** is automatically deployed on the leaf, when the **external-l3 epg** is created within a named L3Out and a leaf is associated with the same L3Out through the **vrf context tenant tenant-name vrf vrf-name l3out l3out-name** command.

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーション モードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： apic1(config)# <b>tenant exampleCorp</b>	テナント 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>external-l3 epg epg: l3out名前l3out</b> 例： apic1(config-tenant)# <b>external-l3 epg epg1 l3out out1</b>	外部 L3 EPG 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>match {ip   ipv6} ip-address/masklength</b> 例： apic1(config-tenant-l3ext-epg)# <b>match ip 192.0.20.0/24</b> apic1(config-tenant-l3ext-epg)# <b>match ipv6 2001::1/64</b>	サブネットに一致するルールを作成します。
ステップ 5	<b>contract consumer contract-name</b> 例：	EPG のコンシューマ コントラクトを指定します。

	コマンドまたはアクション	目的
	<code>apicl (config-tenant-l3ext-epg) # contract consumer cConsumer1</code>	
ステップ 6	<b>contract provider <i>contract-name</i></b>  例 : <code>apicl (config-tenant-l3ext-epg) # contract provider cProvider1</code>	EPG のプロバイダー コントラクトを指定します。

### 例

次に、名前付き L3Out に外部レイヤ 3 EPG を設定する例を示します。

```
apicl# configure
apicl(config)# tenant exampleCorp
apicl (config-tenant) # external-l3 epg epg1 l3out out1
apicl (config-tenant-l3ext-epg) # match ip 192.0.20.0/24
apicl (config-tenant-l3ext-epg) # match ipv6 2001::1/64
apicl (config-tenant-l3ext-epg) # contract consumer cConsumer1
apicl (config-tenant-l3ext-epg) # contract provider cProvider1
```

## IPv6 ネイバー探索

### ネイバー探索

IPv6 ネイバー探索 (ND) は、ノードのアドレスの自動設定、リンク上の他のノードの探索、他のノードのリンク層アドレスの判別、重複アドレスの検出、使用可能なルータと DNS サーバの検出、アドレスプレフィックスの探索、および他のアクティブなネイバー ノードへのパスに関する到達可能性情報の維持を担当します。

ND 固有のネイバー要求/ネイバー アドバタイズメント (NS/NA) およびルータ要求/ルータ アドバタイズメント (RS/RA) パケットタイプは、物理、層3 サブ インターフェイス、および SVI (外部およびパーベイス) を含むすべての ACI ファブリックのレイヤ 3 インターフェイスでサポートされます。APIC リリース 3.1(1x) まで、RS/RA パケットはすべてのレイヤ 3 インターフェイスの自動設定のために使用されますが、拡散型 SVI の設定のみ可能です。

APIC リリース 3.1(2x) より、RS/RA パケットは自動設定のため使用され、ルーテッドインターフェイス、レイヤ 3 サブ インターフェイス、SVI (外部および拡散) を含むレイヤ 3 インターフェイスで設定できます。

ACI のブリッジ ドメイン ND は常にフラッド モードで動作します。ユニキャスト モードはサポートされません。

ACI ファブリック ND サポートに含まれるもの :

- インターフェイス ポリシー (nd:IfPol) は、NS/NA メッセージに関する ND タイマーと動作を制御します。
- ND プレフィックス ポリシー (nd:PrefixPol) コントロール RA メッセージ。
- ND の IPv6 サブネット (fv:Subnet) の設定。
- 外部ネットワークの ND インターフェイス ポリシー。
- 外部ネットワークの設定可能 ND サブネットおよびパーベイシブ ブリッジ ドメインの任意サブネット設定はサポートされません。

設定可能なオプションは次のとおりです。

- 隣接関係
  - 設定可能な静的 Adjacencies : (<vrf、L3Iface < ipv6 address> --> mac address)
  - 動的 Adjacencies : NS/NA パケットの交換経由で学習
- インターフェイス単位
  - ND パケットの制御 (NS/NA)
    - ネイバー要求間隔
    - ネイバー要求再試行回数
  - RA パケットの制御
    - RA の抑制
    - RA MTU の抑制
    - RA 間隔、RA 最小間隔、再送信時間
- プレフィックス単位 (RA でアドバタイズ) の制御
  - ライフタイム、優先ライフタイム
  - プレフィックス コントロール (自動設定、リンク上)
- ネイバー検索重複アドレスの検出 (DAD)

## NX-OS スタイル CLI を使用したブリッジドメイン上のIPv6 ネイバー検索によるテナント、VRF、ブリッジドメインの設定

### 手順

**ステップ1** IPv6 ネイバー検索インターフェイス ポリシーを設定し、ブリッジドメインに割り当てます。

- a) IPv6 ネイバー検索インターフェイス ポリシーを作成します。

例：

```
apicl(config)# tenant ExampleCorp
apicl(config-tenant)# template ipv6 nd policy NDPol001
apicl(config-tenant-template-ipv6-nd)# ipv6 nd mtu 1500
```

- b) VRF およびブリッジドメインを作成します：

例：

```
apicl(config-tenant)# vrf context pvn1
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# bridge-domain bd1
apicl(config-tenant-bd)# vrf member pvn1
apicl(config-tenant-bd)# exit
```

- c) IPv6 ネイバー検索ポリシーをブリッジドメインに割り当てます。

例：

```
apicl(config-tenant)# interface bridge-domain bd1
apicl(config-tenant-interface)# ipv6 nd policy NDPol001
apicl(config-tenant-interface)#exit
```

**ステップ2** サブネット上でIPv6ブリッジドメインサブネットおよびネイバー検索プレフィックスポリシーを作成します。

例：

```
apicl(config-tenant)# interface bridge-domain bd1
apicl(config-tenant-interface)# ipv6 address 34::1/64
apicl(config-tenant-interface)# ipv6 address 33::1/64
apicl(config-tenant-interface)# ipv6 nd prefix 34::1/64 1000 1000
apicl(config-tenant-interface)# ipv6 nd prefix 33::1/64 4294967295 4294967295
```

## 注意事項と制約事項

次のガイドラインと制限事項に適用ネイバー探索ルータ アドバタイズメント (ND RA) のプレフィックスのレイヤ3インターフェイス。

- NDRA 設定は、IPv6 プレフィクスにのみ適用されます。IPv4 プレフィクスで ND ポリシーを設定しようとするは適用に失敗します。

## NX-OS スタイル CLI を使用したレイヤ3 インターフェイス上の RA による IPv6 ネイバー探索インターフェイス ポリシーの設定

この例では、IPv6 ネイバー探索インターフェイス ポリシーを設定し、レイヤ3 インターフェイスに割り当てます。次に、IPv6 レイヤ3 アウトインターフェイス、ネイバー探索プレフィックス ポリシーを設定し、インターフェイスにネイバー探索ポリシーを関連付けます。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant_name</b> 例： apic1(config)# <b>tenant ExampleCorp</b> apic1(config-tenant)#	テナントを作成し、テナントモードを開始します。
ステップ 3	<b>template ipv6 nd policy policy_name</b> 例： apic1(config-tenant)# <b>template ipv6 nd policy NDPol1001</b>	IPv6 ND ポリシーを作成します。
ステップ 4	<b>ipv6 nd mtu mtu value</b> 例： apic1(config-tenant-template-ipv6-nd)# <b>ipv6 nd mtu 1500</b> apic1(config-tenant-template-ipv6)# <b>exit</b> apic1(config-tenant-template)# <b>exit</b> apic1(config-tenant)#	IPv6 ND ポリシーに MTU 値を割り当てます。
ステップ 5	<b>vrf context VRF_name</b> 例： apic1(config-tenant)# <b>vrf context pvn1</b>	VRF を作成します。

	コマンドまたはアクション	目的
	<code>apicl(config-tenant-vrf)# exit</code>	
ステップ 6	<b>l3out VRF_name</b> 例 : <pre>apicl(config-tenant)# l3out l3extOut001</pre>	レイヤ3 アウトを作成します。
ステップ 7	<b>vrf member VRF_name</b> 例 : <pre>apicl(config-tenant-l3out)# vrf member pvnl apicl(config-tenant-l3out)# exit</pre>	VRF をレイヤ3 アウトインターフェイスに関連付けます。
ステップ 8	<b>external-l3 epg instp l3out l3extOut001</b> 例 : <pre>apicl(config-tenant)# external-l3 epg instp l3out l3extOut001 apicl(config-tenant-l3ext-epg)# vrf member pvnl apicl(config-tenant-l3ext-epg)# exit</pre>	レイヤ3 アウトおよび VRF をレイヤ3 インターフェイスに割り当てます。
ステップ 9	<b>leaf 2011</b> 例 : <pre>apicl(config)# leaf 2011</pre>	リーフスイッチモードを開始します。
ステップ 10	<b>vrf context tenant ExampleCorp vrf pvnl l3out l3extOut001</b> 例 : <pre>apicl(config-leaf)# vrf context tenant ExampleCorp vrf pvnl l3out l3extOut001 apicl(config-leaf-vrf)# exit</pre>	VRF をリーフスイッチに関連付けます。
ステップ 11	<b>int eth 1/1</b> 例 : <pre>apicl(config-leaf)# int eth 1/1 apicl(config-leaf-if)#</pre>	インターフェイスモードに入ります。

	コマンドまたはアクション	目的
ステップ 12	<b>vrf member tenant ExampleCorp vrf pvnl l3out l3extOut001</b>  例 :  <pre>apic1(config-leaf-if)# vrf member tenant ExampleCorp vrf pvnl l3out l3extOut001</pre>	インターフェイスで関連付けられているテナント、VRF、レイヤ 3 Out を指定します。
ステップ 13	<b>ipv6 address 2001:20:21:22::2/64 preferred</b>  例 :  <pre>apic1(config-leaf-if)# ipv6 address 2001:20:21:22::2/64 preferred</pre>	プライマリまたは優先 IPv6 アドレスを指定します。
ステップ 14	<b>ipv6 nd prefix 2001:20:21:22::2/64 1000 1000</b>  例 :  <pre>apic1(config-leaf-if)# ipv6 nd prefix 2001:20:21:22::2/64 1000 1000</pre>	レイヤ 3 インターフェイス下で IPv6 ND プレフィックス ポリシーを設定します。
ステップ 15	<b>inherit ipv6 nd NDPol001</b>  例 :  <pre>apic1(config-leaf-if)# inherit ipv6 nd NDPol001 apic1(config-leaf-if)# exit apic1(config-leaf)# exit</pre>	レイヤ 3 インターフェイス下で ND ポリシーを設定します。

設定が完了します。

## 『Configuring HSRP』

### NX-OS スタイル CLI での Cisco APIC を使用してインラインパラメータで HSRP の設定

リーフスイッチが設定されている場合、HSRP が有効になっています。

## 始める前に

- テナントと VRF が設定されています。
- VLAN プールは、適切な VLAN 範囲が定義され、レイヤ3 ドメインが作成されて VLAN プールに接続されている状態で設定される必要があります。
- エンティティプロファイルの接続も、レイヤ3 ドメインに関連付けられている必要があります。
- リーフスイッチのインターフェイスプロファイルは必要に応じて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	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ2	インラインパラメータを作成することにより、HSRP を設定します。 例： apic1(config)# <b>leaf 101</b> apic1(config-leaf)# <b>interface ethernet 1/17</b> apic1(config-leaf-if)# <b>hsrp version 1</b> apic1(config-leaf-if)# <b>hsrp use-bia</b> apic1(config-leaf-if)# <b>hsrp delay minimum 30</b> apic1(config-leaf-if)# <b>hsrp delay reload 30</b> apic1(config-leaf-if)# <b>hsrp 10 ipv4</b> apic1(config-if-hsrp)# <b>ip 182.16.1.2</b> apic1(config-if-hsrp)# <b>ip 182.16.1.3 secondary</b> apic1(config-if-hsrp)# <b>ip 182.16.1.4 secondary</b> apic1(config-if-hsrp)# <b>mac-address 5000.1000.1060</b> apic1(config-if-hsrp)# <b>timers 5 18</b> apic1(config-if-hsrp)# <b>priority 100</b> apic1(config-if-hsrp)# <b>preempt</b> apic1(config-if-hsrp)# <b>preempt delay minimum 60</b> apic1(config-if-hsrp)# <b>preempt delay reload 60</b> apic1(config-if-hsrp)# <b>preempt delay sync 60</b> apic1(config-if-hsrp)# <b>authentication none</b> apic1(config-if-hsrp)# <b>authentication simple</b> apic1(config-if-hsrp)# <b>authentication</b>	

	コマンドまたはアクション	目的
	<pre>md5 apicl(config-if-hsrp)# authentication-key &lt;mypassword&gt; apicl(config-if-hsrp)# authentication-key-timeout &lt;timeout&gt;</pre>	

## NX-OS スタイル CLI のテンプレートとポリシーを使用した Cisco APIC の HSRP の設定

リーフスイッチが設定されている場合、HSRP が有効になっています。

### 始める前に

- テナントと VRF が設定されています。
- VLAN プールは、適切な VLAN 範囲が定義され、レイヤ 3 ドメインが作成されて VLAN プールに接続されている状態で設定される必要があります。
- エンティティプロファイルの接続も、レイヤ 3 ドメインに関連付けられている必要があります。
- リーフスイッチのインターフェイスプロファイルは必要に応じて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	<pre>configure 例： apicl# configure</pre>	コンフィギュレーションモードに入ります。
ステップ 2	<pre>HSRP ポリシーテンプレートを設定します。 例：  apicl(config)# leaf 101 apicl(config-leaf)# template hsrp interface-policy hsrp-intfPol1 tenant t9 apicl(config-template-hsrp-if-pol)# hsrp use-bia apicl(config-template-hsrp-if-pol)# hsrp delay minimum 30 apicl(config-template-hsrp-if-pol)# hsrp delay reload 30</pre>	

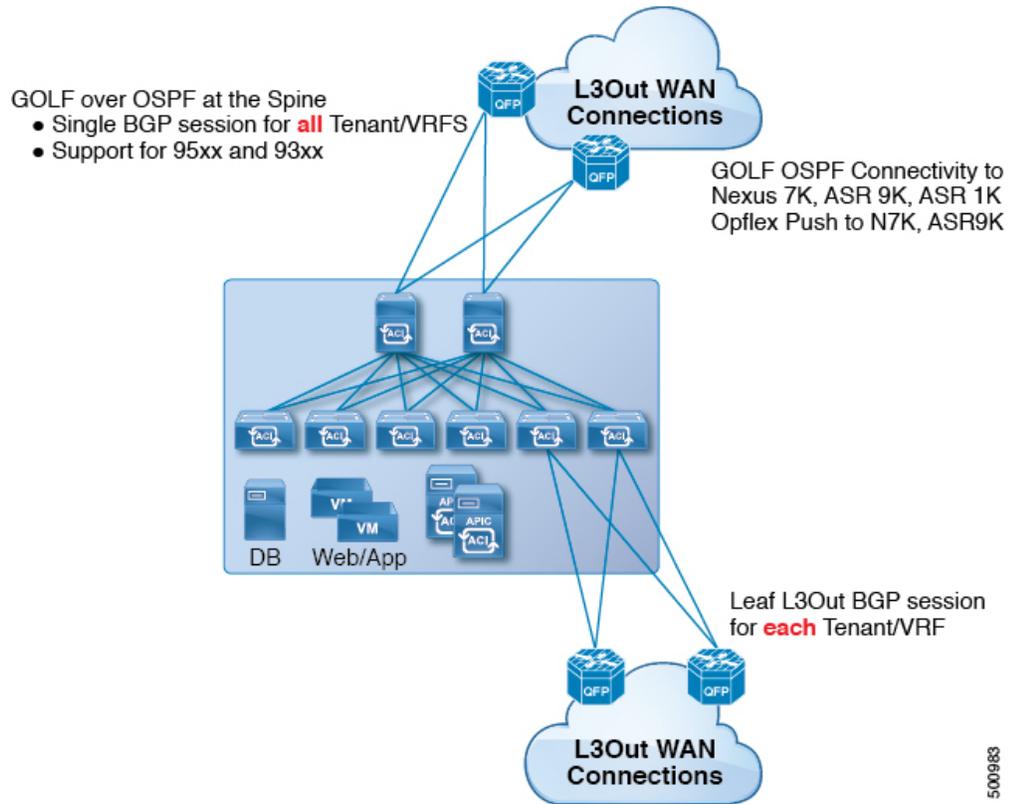
	コマンドまたはアクション	目的
	<pre> apic1(config)# leaf 101 apic1(config-leaf)# template hsrp group-policy hsrp-groupPol1 tenant t9 apic1(config-template-hsrp-group-pol)# timers 5 18 apic1(config-template-hsrp-group-pol)# priority 100 apic1(config-template-hsrp-group-pol)# preempt apic1(config-template-hsrp-group-pol)# preempt delay minimum 60 apic1(config-template-hsrp-group-pol)# preempt delay reload 60 apic1(config-template-hsrp-group-pol)# preempt delay sync 60 </pre>	
ステップ3	<p>設定されているポリシー テンプレートを使用します。</p> <p>例 :</p> <pre> apic1(config)# leaf 101 apic1(config-leaf)# interface ethernet 1/17 apic1(config-leaf-if)# hsrp version 1 apic1(config-leaf-if)# inherit hsrp interface-policy hsrp-intfPol1 apic1(config-leaf-if)# hsrp 10 ipv4 apic1(config-if-hsrp)# ip 182.16.1.2 apic1(config-if-hsrp)# ip 182.16.1.3 secondary apic1(config-if-hsrp)# ip 182.16.1.4 secondary apic1(config-if-hsrp)# mac-address 5000.1000.1060 apic1(config-if-hsrp)# inherit hsrp group-policy hsrp-groupPol1 </pre>	

## Cisco ACI GOLF

### Cisco ACI GOLF

Cisco ACI GOLF 機能 (ファブリック WAN のレイヤ3 EVPN サービス機能とも呼ばれる) では、より効率的かつスケーラブルな ACI ファブリック WAN 接続が可能になります。スパインスイッチに接続されている WAN に OSPF 経由で BGP EVPN プロトコルが使用されます。

図 9: Cisco ACI GOLF のトポロジ



すべてのテナント WAN 接続が、WAN ルータが接続されたスパインスイッチ上で単一のセッションを使用します。データセンター相互接続ゲートウェイ (DCIG) へのテナント BGP セッションのこの集約では、テナント BGP セッションの数と、それらすべてに必要な設定の量を低減することによって、コントロールプレーンのスケールが向上します。ネットワークは、スパインファブリックポートに設定されたレイヤ 3 サブインターフェイスを使用して拡張されます。GOLFを使用した、共有サービスを伴うトランジットルーティングはサポートされていません。

スパインスイッチでの GOLF 物理接続のためのレイヤ 3 外部外側ネットワーク (L3extOut) は、infra テナントの下で指定され、次のものを含みます:

- LNodeP (infra テナントの L3Out では、L3extInstP は必要ありません)。
- infra テナントの GOLF 用の L3extOut のプロバイダラベル。
- OSPF プロトコル ポリシー
- BGP プロトコル ポリシー

すべての通常テナントが、上記で定義した物理接続を使用します。通常のテナントで定義した L3extOut では、次が必要です:

- サブネットとコントラクトを持つ `L3extInstP` (EPG)。サブネットの範囲を使用して、ルート制御ポリシーとセキュリティポリシーのインポートまたはエクスポートを制御します。ブリッジドメインサブネットは外部的にアダプタイズするように設定される必要があります。アプリケーション EPG および GOLF L3Out EPG と同じ VRF に存在する必要があります。
- アプリケーション EPG と GOLF L3Out EPG の間の通信は、(契約優先グループではなく) 明示的な契約によって制御されます。
- `L3extConsLbl` コンシューマ ラベル。これは `infra` テナントの GOLF 用の L3Out の同じプロバイダラベルと一致している必要があります。ラベルを一致させることにより、他のテナント内のアプリケーション EPG が `LNodeP` 外部 L3Out EPG を利用することが可能になります。
- `infra` テナント内のマッチングプロバイダ `L3extOut` の BGP EVPN セッションは、この `L3out` で定義されたテナント ルートをアダプタイズします。

次に示す GOLF のガイドラインおよび制限事項に従ってください。

- すべての Cisco Nexus 9000 シリーズ ACI モードのスイッチと、すべての Cisco Nexus 9500 プラットフォーム ACI モード スイッチライン カードおよびファブリック モジュールが GOLF をサポートします。Cisco APIC、リリース 3.1(x) 以降では、これに N9K-C9364C スイッチが含まれます。
- 現時点では、ファブリック全体のスパインスイッチインターフェイスに展開できるのは、単一の GOLF プロバイダ ポリシーだけです。
- APIC リリース 2.0(2) までは、GOLF はマルチポッドでサポートされていません。リリース 2.0 (2) では、同じファブリックでの 2 つの機能を、スイッチ名の末尾に「EX」のない Cisco Nexus N9000K スイッチ上でのみサポートしています。たとえば N9K-9312TX です。2.1(1) リリース以降では、2 つの機能を、マルチポッドおよび EVPN トポロジで使用されているすべてのスイッチとともに展開できるようになりました。
- スパイン スイッチで GOLF を設定する場合、コントロールプレーンがコンバージするまでは、別のスパイン スイッチで GOLF の設定を行わないでください。
- スパイン スイッチは複数のプロバイダの GOLF 外側ネットワーク (GOLF L3Outs) に追加できますが、GOLF L3Out ごとのプロバイダ ラベルは異なっている必要があります。また、この例では、OSPF エリアも `L3extOut` ごとに異なっていて、異なるループバックアドレスを使用する必要があります。
- `infra` テナント内のマッチングプロバイダ `L3out` の BGPEVPN セッションは、この `L3extOut` で定義されたテナント ルートをアダプタイズします。
- 3 つの GOLF Outs を展開する場合、1 つだけが GOLF, and 0/0 エクスポート集約のプロバイダ/コンシューマラベルを持っているなら、APIC はすべてのルートをエクスポートします。これは、テナントのリーフ スイッチ上の既存の `L3extOut` と同じです。
- スパインスイッチとデータセンター相互接続 (DCI) ルータ間に直接ピアリングがある場合、リーフ スイッチから ASR へのトランジットルートには、リーフ スイッチの PTEP として次のホップが存在することになります。この場合、その ACI ポッドの TEP 範囲に対

して ASR の静的ルートを定義します。また、DCI が同じポッドにデュアルホーム接続されている場合は、静的ルートの優先順位（管理距離）は、他のリンクを通じて受信するルートと同じである必要があります。

- デフォルトの `bgpPeerPfxPol` ポリシーは、ルートを 20,000 に制限します。ACI WAN インターコネクトピアの場合には、必要に応じてこれを増やしてください。
- 1 つのスパインスイッチ上に 2 つの `L3extOut` が存在し、そのうちの一方のプロバイダラベルが `prov1` で DCI 1 とピアリングしており、もう一方の `L3extOut` のプロバイダラベルが `prov2` で DCI 2 とピアリングしているという、展開シナリオを考えます。テナント VRF に、プロバイダラベルのいずれか一方 (`prov1` または `prov2`) をポイントしているコンシューマラベルがある場合、テナントルートは DCI 1 と DCI 2 の両方に送信されます。
- GOLF OpFlex Vrf を集約する場合、ACI ファブリックまたは GOLF OpFlex VRF とシステム内のその他の VRF 間の GOLF デバイスでは、ルートのリーキングは発生しません。VRF リーキングのためには、(GOLF ルータではなく) 外部デバイスを使用する必要があります。



- (注) Cisco ACI は、IP フラグメンテーションをサポートしていません。したがって、外部ルータへのレイヤ 3 Outside (L3Out) 接続、または Inter-Pod Network (IPN) を介した multipod 接続を設定する場合は、MTU が両側で適切に設定されていることが重要です。ACI、Cisco NX-OS、Cisco IOS などの一部のプラットフォームでは、設定された MTU 値は IP ヘッダーを考慮に入れています（結果として、最大パケットサイズは、ACI で 9216 バイト、NX-OS および IOS で 9000 バイトに設定されます）。ただし、IOS XR などの他のプラットフォームは、パケットヘッダーのを除く MTU 値を設定します（結果として最大パケットサイズは 8986 バイトになります）。

各プラットフォームの適切な MTU 値については、それぞれの設定ガイドを参照してください。

CLI ベースのコマンドを使用して MTU をテストすることを強く推奨します。たとえば、Cisco NX-OS CLI で `ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1` などのコマンドを使用します。

## NX-OS スタイル CLI を使用して Cisco ACI GOLF サービスを設定する設定タスク

次のタスクを実行して、NX-OS スタイル CLI で GOLF サービスを設定します（BGP EVPN プロトコルを使用）。

- VLAN ドメイン、VRF、インターフェイス IP アドレッシング、OSPF を含む BGP EVPN の内部テナントを設定します。
- スパインの BPG を設定して BGP EVPN をサポートします。
- BGP EVPN のテナントを設定します。

- テナントの BGP EVPN ルートターゲット、ルートマップ、プレフィックス EPG を設定します。
- BGP アドレスファミリーを設定して、**host-rt-enable** コマンドで DCIG に BGP EVPN type-2 (MAC-IP) ホストルートを配布することを有効にします。

## スパインを設定し、NX-OS スタイル CLI を使用して、BGP EVPN のテナントインフラ

このタスクを設定する方法を説明する、次の手順で VLAN ドメイン、VRF、インターフェイスの IP アドレッシングを含む、BGP EVPN および OSPF のテナントインフラ。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</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。  apic1# <b>configure</b>
ステップ 2	<b>vlan-domain <i>vlan-domain-name</i> dynamic</b>	Creates a VLAN domain.  apic1(config)# <b>vlan-domain evpn-dom dynamic</b>
ステップ 3	<b>spine <i>spine-name</i></b>	Creates the spine or enters spine configuration mode.  apic1(config)# <b>spine 111</b>
ステップ 4	<b>vrf context <i>tenant</i><i>tenant-name</i> vrf <i>vrf-name</i></b>	テナントに、VRF を関連付けます。  apic1(config-spine)# <b>vrf context tenant infra vrf overlay-1</b>
ステップ 5	<b>router-id <i>A.B.C.D</i></b>	VRF のルータ ID を設定します。  apic1(config-spine-vrf)# <b>router-id 10.10.3.3</b>
ステップ 6	<b>exit</b>	Returns to spine configuration mode.  apic1(config-spine-vrf)# <b>exit</b>
ステップ 7	<b>interface ethernet <i>slot/port</i></b>	スパインノードのインターフェイスを設定します。  apic1(config-spine)# <b>interface ethernet 1/33</b>
ステップ 8	<b>vlan-domain member <i>vlan-domain-name</i></b>	Associates the interface with the VLAN domain.  apic1(config-spine-if)# <b>vlan-domain member evpn-dom</b>

■ スパインを設定し、NX-OS スタイル CLI を使用して、BGP EVPN のテナントインフラ

	コマンドまたはアクション	目的
ステップ 9	<b>exit</b>	Returns to spine configuration mode. <code>apic1(config-spine-if)# <b>exit</b></code>
ステップ 10	<b>interface ethernet sub-interface-id</b>	Creates a sub-interface. <code>apic1(config-spine)# <b>interface ethernet 1/33.4</b></code>
ステップ 11	<b>vrf member tenant tenant-name vrf vrf-name</b>	インターフェイスをオーバーレイ1 VRF に関連付けると、テナントインフラ。 <code>apic1(config-spine-if)# <b>vrf member tenant infra vrf overlay-1</b></code>
ステップ 12	<b>mtu mtu-value</b>	Sets the maximum transmission unit (MTU) for the interface. <code>apic1(config-spine-if)# <b>mtu 1500</b></code>
ステップ 13	<b>ip address A.B.C.D/LEN</b>	インターフェイスの IP アドレスを設定します。 <code>apic1(config-spine-if)# <b>ip address 5.0.0.1/24</b></code>
ステップ 14	<b>ip router ospf default area ospf エリア id</b>	インターフェイスのデフォルトの OSPF エリア ID を設定します。 <code>apic1(config-spine-if)# <b>ip router ospf default area 0.0.0.150</b></code>
ステップ 15	<b>exit</b>	Returns to spine configuration mode. <code>apic1(config-spine-if)# <b>exit</b></code>
ステップ 16	<b>interface ethernet slot/port</b>	スパインノードのインターフェイスを設定します。 <code>apic1(config-spine)# <b>interface ethernet 1/34</b></code>
ステップ 17	<b>vlan-domain member vlan-domain-name</b>	Associates the interface with the VLAN domain. <code>apic1(config-spine-if)# <b>vlan-domain member evpn-dom</b></code>
ステップ 18	<b>exit</b>	Returns to spine configuration mode. <code>apic1(config-spine-if)# <b>exit</b></code>
ステップ 19	<b>interface ethernet sub-interface-id</b>	Creates a sub-interface. <code>apic1(config-spine)# <b>interface ethernet 1/34.4</b></code>
ステップ 20	<b>vrf member tenant tenant-name vrf vrf-name</b>	インターフェイスをオーバーレイ1 VRF に関連付けると、テナントインフラ。

	コマンドまたはアクション	目的
		<code>apic1(config-spine-if)# vrf member tenant infra vrf overlay-1</code>
ステップ 21	<code>mtu mtu-value</code>	Sets the maximum transmission unit (MTU) for the interface.  <code>apic1(config-spine-if)# mtu 1500</code>
ステップ 22	<code>ip address A.B.C.D/LEN</code>	インターフェイスの IP アドレスを設定します。  <code>apic1(config-spine-if)# ip address 2.0.0.1/24</code>
ステップ 23	<code>ip router ospf default areaospf エリア id</code>	インターフェイスのデフォルトの OSPF エリア ID を設定します。  <code>apic1(config-spine-if)# ip router ospf default area 0.0.0.200</code>
ステップ 24	<code>exit</code>	Returns to spine configuration mode.  <code>apic1(config-spine-if)# exit</code>
ステップ 25	<code>router ospf default</code>	スパインの OSPF を設定します。  <code>apic1(config-spine)# router ospf default</code>
ステップ 26	<code>vrf member tenant tenant-name vrf vrf-name</code>	テナントをオーバーレイ 1 の VRF とインフラ Router OSPF ポリシーに関連付けます。  <code>apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1</code>
ステップ 27	<code>area area-id loopback loopback-ip-address</code>	Configure an OSPF area for the OSPF policy.  <code>apic1(config-spine-ospf-vrf)# area 0.0.0.150 loopback 10.10.5.3</code>
ステップ 28	<code>area area-id loopback loopback-ip-address</code>	OSPF ポリシーのもう 1 つの OSPF エリアを設定します。  <code>apic1(config-spine-ospf-vrf)# area 0.0.0.200 loopback 10.10.4.3</code>
ステップ 29	<code>exit</code>	Returns to spine OSPF configuration mode.  <code>apic1(config-spine-ospf-vrf)# exit</code>
ステップ 30	<code>exit</code>	Returns to spine configuration mode.  <code>apic1(config-spine-ospf)# exit</code>

## 複数のサイトで共有 APIC ゴルフ接続

トポロジでは、複数のサイト、APIC サイトの拡大 Vrf は、ゴルフ接続を共有している場合、リスクのクロス VRF トラフィックの問題を回避する次のガイドラインに従います。

### スパインスイッチと、DCI の間でルートターゲットの設定

ゴルフ Vrf の EVPN ルートターゲット (RTs) を設定する 2 つの方法があります: 手動 RT と自動 RT. ルートターゲットは、ACI 背表紙と OpFlex を介して DCIs の間で同期されます。ゴルフ Vrf の自動 RT は、形式に組み込まれて Fabric ID: - ASN : [ FabricID ] VNID

2 つのサイトには、次の図のように導入の Vrf がある、Vrf 間のトラフィックを混在させることができます。

サイト 1	サイト 2
ASN: 100、ファブリック ID: 1	ASN: 100、ファブリック ID: 1
VRF A : VNID 1000 インポート/エクスポートルートターゲット : 100 : [1] 1000	VRF A : VNID 2000 インポート/エクスポートルートターゲット : 100 : [1] 2000
VRF B : VNID 2000 インポート/エクスポートルートターゲット : 100 : [1] 2000	VRF B : VNID 1000 インポート/エクスポートルートターゲット : 100 : [1] 1000

### Dci のために必要なルート マップ

トンネルは、中継ルートは、[DCI を介してリークとサイト間では作成されません、ため、コントロールプレーンの手間をも削減する必要があります。もう 1 つのサイトでゴルフ スパインに、DCI への 1 つのサイトでゴルフ スパインから送信される EVPN タイプ 5 およびタイプ 2 ルートを送信できませんする必要があります。これが発生スパイン スイッチに dci のために次のタイプの BGP セッションが必要がある場合。

Site1: IBGP--DCI--EBGP--サイト 2

Site1: EBGP--DCI--IBGP--サイト 2

Site1:--DCI--EBGP EBGP--サイト 2

Site1: IBGP RR クライアント--DCI (RR)---IBGP サイト 2

Dci のためにこの問題を避けるためには、ルートマップは、インバウンドおよびアウトバウンドのピア ポリシーのさまざまな BGP コミュニティで使用されます。

ルートを 1 つのサイト、もう 1 つのサイト フィルタ着信ピア ポリシーでコミュニティに基づくルートでゴルフ スパインへのアウトバウンドピア ポリシー ゴルフ スパインから受信します。別のアウトバウンドピア ポリシーは、WAN へコミュニティを取り除き。すべてのルートマップは、ピアのレベルです。

## NX-OS スタイル CLI を使用した推奨される共有 GOLF 設定

マルチサイトで管理されている複数の APIC サイト間で、DCI による GOLF 接続を共有する場合、ルートマップと BPG を設定し VRF 間のトラフィックの問題を避けるために次の手順を使用します。

### 手順

#### ステップ1 インバウンドルート マップ

例：

```
Inbound peer policy to attach community:  
|  
| route-map multi-site-in permit 10  
|  
| set community 1:1 additive
```

#### ステップ2 アウトバウンドピア ポリシーを設定し、インバウンドピア ポリシーのコミュニティに基づいてルートをフィルタします。

例：

```
ip community-list standard test-com permit 1:1  
  
route-map multi-site-out deny 10  
  
    match community test-com exact-match  
  
route-map multi-site-out permit 11
```

#### ステップ3 アウトバウンドピア ポリシーを設定し、WAN へのコミュニティをフィルタします。

例：

```
ip community-list standard test-com permit 1:1  
  
route-map multi-site-wan-out permit 11  
  
    set comm-list test-com delete
```

#### ステップ4 BGP を設定します。

例：

```
router bgp 1  
  
    address-family l2vpn evpn  
  
        neighbor 11.11.11.11 remote-as 1  
  
            update-source loopback0  
  
        address-family l2vpn evpn  
  
            send-community both  
  
        route-map multi-site-in in  
  
        neighbor 13.0.0.2 remote-as 2
```

```

address-family l2vpn evpn
    send-community both
    route-map multi-site-out out

```

## NX-OS スタイル CLI を使用して、スパインで BGP EVPN をサポートする BGP の設定

このタスクは、次の手順で BGP EVPN をサポートするスパインで BGP を設定する方法を示しています。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</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。 <code>apic1# <b>configure</b></code>
ステップ 2	<b>spine spine-name</b>	Creates the spine or enters spine configuration mode. <code>apic1(config)# <b>spine 111</b></code>
ステップ 3	<b>router bgp AS-number</b>	スパインノードの BGP を設定します。 <code>apic1(config-spine)# <b>router bgp 100</b></code>
ステップ 4	<b>vrf context tenant tenant-name vrf vrf-name</b>	Router BGP ポリシーに関連付けます、インフラ テナントとオーバーレイ 1 VRF。 <code>apic1(config-spine-bgp)# <b>vrf context tenant infra vrf overlay-1</b></code>
ステップ 5	<b>vrf context tenant tenant-name vrf vrf-name</b>	Router BGP ポリシーに関連付けます、インフラ テナントとオーバーレイ 1 VRF。 <code>apic1(config-spine-bgp-vrf)# <b>vrf context tenant infra vrf overlay-1</b></code>
ステップ 6	<b>neighbor neighbor-ip-address evpn</b>	Configures the IP address for an EVPN BGP neighbor. <code>apic1(config-spine-bgp-vrf)# <b>neighbor 10.10.4.1 evpn</b></code>
ステップ 7	<b>label label-name</b>	ネイバーにラベルを割り当てます。

	コマンドまたはアクション	目的
		apic1(config-spine-bgp-vrf-neighbor)# <b>label evpn-aci</b>
ステップ 8	<b>update-source loopback</b> ループバック IP アドレス <b>vrf vrf</b>	ネイバーループバック IP アドレスを更新の送信元を設定します。  apic1(config-spine-bgp-vrf-neighbor)# <b>update-source loopback 10.10.4.3</b>
ステップ 9	<b>remote-as AS-number</b>	Specifies the autonomous system (AS) number of the neighbor. The valid value can be from 1 to 4294967295.  apic1(config-spine-bgp-vrf-neighbor)# <b>remote-as 100</b>
ステップ 10	<b>exit</b>	Returns to BGP VRF configuration mode.  apic1(config-spine-bgp-vrf-neighbor)# <b>exit</b>
ステップ 11	<b>neighbor neighbor-ip-address evpn</b>	Configures the IP address for an EVPN BGP neighbor.  apic1(config-spine-bgp-vrf)# <b>neighbor 10.10.5.1 evpn</b>
ステップ 12	<b>label label-name</b>	ネイバーにラベルを割り当てます。  apic1(config-spine-bgp-vrf-neighbor)# <b>label evpn-aci2</b>
ステップ 13	<b>update-source loopback</b> ループバック IP アドレス <b>vrf vrf-name</b>	ネイバーループバック IP アドレスを更新の送信元を設定します。  apic1(config-spine-bgp-vrf-neighbor)# <b>update-source loopback 10.10.5.3</b>
ステップ 14	<b>remote-as AS-number</b>	Specifies the autonomous system (AS) number of the neighbor. The valid value can be from 1 to 4294967295.  apic1(config-spine-bgp-vrf-neighbor)# <b>remote-as 100</b>

## NX-OS スタイル CLI を使用した BG EVPN のテナントの設定

このタスクは、次の手順で BGP EVPN のテナントを設定する方法を示しています。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</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。

	コマンドまたはアクション	目的
		apic1# <b>configure</b>
ステップ 2	<b>tenant</b> <i>tenant-name</i>	テナントを作成するか、テナント設定モードを開始します。  apic1(config)# <b>tenant sky</b>
ステップ 3	<b>vrf context</b> <i>vrf-name</i>	テナントの VRF を作成します。  apic1(config-tenant)# <b>vrf context vrf-sky</b>
ステップ 4	<b>exit</b>	テナントコンフィギュレーションモードに戻ります。  apic1(config-tenant-vrf)# <b>exit</b>
ステップ 5	<b>bridge-domain</b> <i>bd-name</i>	ブリッジドメインを作成します  apic1(config-tenant)# <b>bridge-domain bd-sky</b>
ステップ 6	<b>vrf member</b> <i>vrf-name</i>	VRF およびテナントにブリッジドメインを関連付けます。  apic1(config-tenant-bd)# <b>vrf member vrf-sky</b>
ステップ 7	<b>exit</b>	テナントコンフィギュレーションモードに戻ります。  apic1(config-tenant-bd)# <b>exit</b>
ステップ 8	<b>interface bridge-domain</b> <i>bd-name</i>	ブリッジドメインにインターフェイスを作成します。  apic1(config-tenant)# <b>interface bridge-domain bd_sky</b>
ステップ 9	<b>ip address</b> <i>A.B.C.D/LEN</i>	IPアドレスと長さをブリッジドメインインターフェイスに割り当てます。  apic1(config-tenant-interface)# <b>ip address 59.10.1.1/24</b>
ステップ 10	<b>exit</b>	テナントコンフィギュレーションモードに戻ります。  apic1(config-tenant-interface)# <b>exit</b>
ステップ 11	<b>bridge-domain</b> <i>bd-name</i>	ブリッジドメインを作成します  apic1(config-tenant)# <b>bridge-domain bd-sky2</b>
ステップ 12	<b>vrf member</b> <i>vrf-name</i>	VRF およびテナントにブリッジドメインを関連付けます。

	コマンドまたはアクション	目的
		<code>apic1(config-tenant-bd)# vrf member vrf-sky</code>
ステップ 13	<code>exit</code>	テナントコンフィギュレーションモードに戻ります。 <code>apic1(config-tenant-bd)# exit</code>
ステップ 14	<code>interface bridge-domain bd-name</code>	ブリッジドメインにインターフェイスを作成します。 <code>apic1(config-tenant)# interface bridge-domain bd_sky2</code>
ステップ 15	<code>ip address A.B.C.D/LEN</code>	IPアドレスと長さをブリッジドメインインターフェイスに割り当てます。 <code>apic1(config-tenant-interface)# ip address 59.11.1.1/24</code>
ステップ 16	<code>exit</code>	テナントコンフィギュレーションモードに戻ります。 <code>apic1(config-tenant-interface)# exit</code>

## ルートマップの設定

このタスクは、BGP EVPN を介してブリッジドメインサブネットをアドバタイズするルートマップを設定する方法を示します。各ブリッジドメインは、一意のプロバイダラベルで、スパイン上のさまざまな BGP EVPN セッションを通してアドバタイズされます。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<code>configure</code>	コンフィギュレーションモードに入ります。 <code>apic1# configure</code>
ステップ 2	<code>spine spine-name</code>	スパインを作成するか、スパイン設定モードを開始します。 <code>apic1(config)# spine 111</code>
ステップ 3	<code>vrf context tenanttenant-name vrf vrf-name</code>	VRFを作成するか、VRF設定モードを開始します。 <code>apic1(config-spine)# vrf context tenant sky vrf vrf_sky</code>
ステップ 4	<code>address-family { ipv4   ipv6 } unicast</code>	VRFのIPv4またはIPv6ユニキャストアドレスファミリを設定します。

	コマンドまたはアクション	目的
		apicl(config-spine-vrf)# <b>address-family ipv4 unicast</b>
ステップ 5	<b>route-target mode extended-community-number</b>	アドレスファミリにエクスポートルートターゲットを割り当てます。  apicl(config-spine-vrf-af)# <b>route-target export 100:1</b>
ステップ 6	<b>route-target mode extended-community-number</b>	アドレスファミリにインポートルートターゲットを割り当てます。  apicl(config-spine-vrf-af)# <b>route-target import 100:1</b>
ステップ 7	<b>exit</b>	スパイン VRF 設定モードに戻ります。  apicl(config-spine-vrf-af)# <b>exit</b>
ステップ 8	<b>route-map route-map-name</b>	EVPN にルートマップを作成します (トランジットネットワークから学習したプレフィックス)。  apicl(config-spine-vrf)# <b>route map rmap</b>
ステップ 9	<b>ip prefix-list ip-pl-name permit A.B.C.D/LEN</b>	ルートマップに IP プレフィックスリストを追加して、指定されたサブネットからのトラフィックを許可します。  apicl(config-spine-vrf-route-map)# <b>ip prefix-list pl permit 11.10.10.0/24</b>
ステップ 10	<b>match bridge-domain bd-name</b>	ブリッジドメインに属するトラフィックを照合するルートマップを設定します。  apicl(config-spine-vrf-route-map)# <b>match bridge-domain bd_sky</b>
ステップ 11	<b>exit</b>	スパイン VRF ルートマップ設定モードに戻ります。  apicl(config-spine-vrf-route-map-match)# <b>exit</b>
ステップ 12	<b>match prefix-list pl-name</b>	指定されたプレフィックスリストを照合するルートマップを設定します。  apicl(config-spine-vrf-route-map)# <b>match prefix-list pl</b>
ステップ 13	<b>exit</b>	スパイン VRF ルートマップ設定モードに戻ります。  apicl(config-spine-vrf-route-map-match)# <b>exit</b>

	コマンドまたはアクション	目的
ステップ 14	<b>exit</b>	スパイン VRF 設定モードに戻ります。 apic1(config-spine-vrf-route-map)# <b>exit</b>
ステップ 15	<b>evpn export map</b> <i>route-map-name label consumer-label-name</i>	VRF にコンシューマ ラベルを割り当てます。 apic1(config-spine-vrf)# <b>evpn export map rmap label evpn-aci</b>
ステップ 16	<b>route-map</b> <i>route-map-name</i>	EVPN にルート マップを作成します (トランジットネットワークから学習したプレフィックス)。 apic1(config-spine-vrf)# <b>route map rmap2</b>
ステップ 17	<b>match bridge-domain</b> <i>bd-name</i>	ブリッジドメインに属するトラフィックを照合するルートマップを設定します。 apic1(config-spine-vrf-route-map)# <b>match bridge-domain bd_sky</b>
ステップ 18	<b>exit</b>	スパイン VRF ルート マップ設定モードに戻ります。 apic1(config-spine-vrf-route-map-match)# <b>exit</b>
ステップ 19	<b>match prefix-list</b> <i>pl-name</i>	指定されたプレフィックスリストを照合するルートマップを設定します。 apic1(config-spine-vrf-route-map)# <b>match prefix-list pl</b>
ステップ 20	<b>exit</b>	スパイン VRF ルート マップ設定モードに戻ります。 apic1(config-spine-vrf-route-map-match)# <b>exit</b>
ステップ 21	<b>exit</b>	スパイン VRF 設定モードに戻ります。 apic1(config-spine-vrf-route-map)# <b>exit</b>
ステップ 22	<b>evpn export map</b> <i>route-map-name label consumer-label-name</i>	VRF にコンシューマ ラベルを割り当てます。 apic1(config-spine-vrf)# <b>evpn export map rmap label evpn-aci2</b>
ステップ 23	<b>external-l3</b> <i>epgepg-name</i>	apic1(config-spine-vrf)# <b>external-l3 epg 13_sky</b>

	コマンドまたはアクション	目的
ステップ 24	<code>vrf member vrf-name</code>	<code>apic1(config-spine-vrf-l3ext-epg)# vrf member vrf_sky</code>
ステップ 25	<code>match ip A.B.C.D/LEN</code>	EPG の一部であるホストを識別するサブネットを設定します。  <code>apic1(config-spine-vrf-l3ext-epg)# match ip 80.10.1.0/24</code>

## NX-OS スタイル CLI を使用して DCIG への配布の BGP EVPN タイプ2のホストルートの有効化

### 手順

	コマンドまたはアクション	目的
ステップ 1	BGP アドレス ファミリ configuration mode(設定モード、コンフィギュレーションモード)で、次のコマンドを DCIG に配布 EVPN タイプ2のホストルートを設定します。  例：  <code>apic1(config)# leaf 101</code> <code>apic1(config-leaf)# template bgp address-family bgpAf1 tenant bgp_t1</code> <code>apic1(config-bgp-af)# distance 250 240 230</code> <code>apic1(config-bgp-af)# host-rt-enable</code> <code>apic1(config-bgp-af)# exit</code>	このテンプレートは、テナント <code>bgp_t1</code> は VRF の導入を持つすべてのノードで利用可能になります。配布 EVPN タイプ2のホストルートを無効にするには、次のように入力します。、 <code>no ホスト-rt-enable</code> コマンド。

## NX-OS スタイル CLI を使用した Cisco ACI GOLF 設定の例:

次の例を設定する CLI コマンドの show GOLF サービスで、OSPF over スパイン スイッチに接続されている WAN ルータの BGP EVPN プロトコルを使用します。

### 設定、BGP EVPN のテナントインフラ

次の例を設定する方法を示しています、インフラ VLAN ドメイン、VRF、インターフェイスの IP アドレッシングを含む、BGP EVPN および OSPF のテナントします。

```
configure
  vlan-domain evpn-dom dynamic
  exit
```

```
spine 111
  # Configure Tenant Infra VRF overlay-1 on the spine.
  vrf context tenant infra vrf overlay-1
  router-id 10.10.3.3
  exit

interface ethernet 1/33
  vlan-domain member golf_dom
  exit
interface ethernet 1/33.4
  vrf member tenant infra vrf overlay-1
  mtu 1500
  ip address 5.0.0.1/24
  ip router ospf default area 0.0.0.150
  exit
interface ethernet 1/34
  vlan-domain member golf_dom
  exit
interface ethernet 1/34.4
  vrf member tenant infra vrf overlay-1
  mtu 1500
  ip address 2.0.0.1/24
  ip router ospf default area 0.0.0.200
  exit

router ospf default
  vrf member tenant infra vrf overlay-1
  area 0.0.0.150 loopback 10.10.5.3
  area 0.0.0.200 loopback 10.10.4.3
  exit
exit
```

### スパインノード上の BGP の設定

次の例では、BGP EVPN をサポートする BGP を設定する方法を示します。

```
Configure
spine 111
router bgp 100
  vrf member tenant infra vrf overlay- 1
  neighbor 10.10.4.1 evpn
  label golf_aci
  update-source loopback 10.10.4.3
  remote-as 100
  exit
  neighbor 10.10.5.1 evpn
  label golf_aci2
  update-source loopback 10.10.5.3
  remote-as 100
  exit
exit
exit
```

### BGP EVPN のテナントの設定

次の例では、BGPEVPN、BGPEVPNセッションで提供されるゲートウェイサブネットを含むテナントを設定する方法を示します。

```
configure
tenant sky
  vrf context vrf_sky
```

```

exit
bridge-domain bd_sky
  vrf member vrf_sky
exit
interface bridge-domain bd_sky
  ip address 59.10.1.1/24
exit
bridge-domain bd_sky2
  vrf member vrf_sky
exit
interface bridge-domain bd_sky2
  ip address 59.11.1.1/24
exit
exit

```

### BGP EVPN ルート ターゲット、ルートマップと、テナントのプレフィックス EPG の設定

次の例では、BGP EVPN を介してブリッジ ドメイン サブネットをアドバタイズするルートマップを設定する方法を示します。

```

configure
spine 111
  vrf context tenant sky vrf vrf_sky
    address-family ipv4 unicast
      route-target export 100:1
      route-target import 100:1
    exit

  route-map rmap
    ip prefix-list p1 permit 11.10.10.0/24
    match bridge-domain bd_sky
    exit
    match prefix-list p1
    exit

  evpn export map rmap label golf_aci

  route-map rmap2
    match bridge-domain bd_sky
    exit
    match prefix-list p1
    exit
  exit

  evpn export map rmap label golf_aci2

external-l3 epg l3_sky
  vrf member vrf_sky
  match ip 80.10.1.0/24
exit

```

## DCIG への EVPN タイプ2 ルート配信のトラブルシューティング

EVPN トポロジで最適なトラフィックの転送を行うには、ファブリック スパインを有効にし、EVPN タイプ2 (MAC-IP) ルートを使用して、BGP EVPN タイプ5 (IPプレフィックス) ルートの形式で公的BDサブネットとともにホストルートを Data Center Interconnect Gateway (DCIG) に配信できます。HostLeak オブジェクトを使用して有効になります。ルート配信に問題が発生した場合、このトピックでのトラブルシューティングの手順を使用します。

## 手順

- ステップ1** スパインスイッチCLIで次のようなコマンドを入力して、問題のVRF AFでHostLeakオブジェクトが有効になっていることを確認します。

例：

```
spine1# ls /mit/sys/bgp/inst/dom-apple/af-ipv4-ucast/
ctrl-l2vpn-evpn ctrl-vpnv4-ucast hostleak summary
```

- ステップ2** スパインスイッチCLIで次のようなコマンドを入力して、設定MOがBGPにより正常に処理されていることを確認します。

例：

```
spine1# show bgp process vrf apple
```

次と同様の出力を検索します。

```
Information for address family IPv4 Unicast in VRF apple
Table Id           : 0
Table state        : UP
Table refcount     : 3
Peers              Active-peers  Routes    Paths    Networks  Aggregates
0                  0              0         0        0         0

Redistribution
None

Wait for IGP convergence is not configured
GOLF EVPN MAC-IP route is enabled
EVPN network next-hop 192.41.1.1
EVPN network route-map map_pfxleakctrl_v4
Import route-map rtctrlmap-apple-v4
EVPN import route-map rtctrlmap-evpn-apple-v4
```

- ステップ3** 公的BDサブネットがEVPNタイプ5ルートとしてDCIGに提供されたことを確認します。

例：

```
spine1# show bgp l2vpn evpn 10.6.0.0 vrf overlay-1
Route Distinguisher: 192.41.1.5:4123 (L3VNI 2097154)
BGP routing table entry for [5]:[0]:[0]:[16]:[10.6.0.0]:[0.0.0.0]/224, version 2088
Paths: (1 available, best #1)
Flags: (0x000002 00000000) on xmit-list, is not in rib/evpn
Multipath: eBGP iBGP

Advertised path-id 1
Path type: local 0x4000008c 0x0 ref 1, path is valid, is best path
AS-Path: NONE, path locally originated
 192.41.1.1 (metric 0) from 0.0.0.0 (192.41.1.5)
  Origin IGP, MED not set, localpref 100, weight 32768
  Received label 2097154
  Community: 1234:444
  Extcommunity:
    RT:1234:5101
    4BYTEAS-GENERIC:T:1234:444

Path-id 1 advertised to peers:
50.41.50.1
```

[パス タイプ] エントリで、**ref 1** は1個のルートが送信されたことその1つのルートが送信されたを示します。

**ステップ4** EVPN ピアにアドバタイズされたホストルートが EVPN タイプ2 MAC IP ルートであることを確認します。

例：

```
spine1# show bgp l2vpn evpn 10.6.41.1 vrf overlay-1
Route Distinguisher: 10.10.41.2:100 (L2VNI 100)
BGP routing table entry for [2]:[0]:[2097154]:[48]:[0200.0000.0002]:[32]:[10.6.41.1]/272, version 1146
Shared RD: 192.41.1.5:4123 (L3VNI 2097154)
Paths: (1 available, best #1)
Flags: (0x00010a 00000000) on xmit-list, is not in rib/evpn
Multipath: eBGP iBGP

  Advertised path-id 1
  Path type: local 0x4000008c 0x0 ref 0, path is valid, is best path
  AS-Path: NONE, path locally originated
EVPN network: [5]:[0]:[0]:[16]:[10.6.0.0]:[0.0.0.0] (VRF apple)
  10.10.41.2 (metric 0) from 0.0.0.0 (192.41.1.5)
  Origin IGP, MED not set, localpref 100, weight 32768
  Received label 2097154 2097154
  Extcommunity:
    RT:1234:16777216
```

```
Path-id 1 advertised to peers:
50.41.50.1
```

[共有 RD] ラインは、EVPN タイプ2 ルートと BD サブネットとで共有された RD/VNI を示します。

[EVPN ネットワーク] ラインは、BD サブネットの EVPN タイプ5 のルートを表示します。

[ピアにアドバタイズされたパス ID] は、EVPN ピアにパスがアドバタイズされたことを示します。

**ステップ5** DCIG デバイスで次のようなコマンドを入力して（DCIG が下の例で Cisco ASR 9000 スイッチであることを仮定します）、EVPN ピア（DCIG）が正常なタイプ2 MAC IP ルートを受信し、ホストルートが正常に特定の VRF にインポートされたことを確認します。

例：

```
RP/0/RSP0/CPU0:asr9k#show bgp vrf apple-2887482362-8-1 10.6.41.1
Tue Sep  6 23:38:50.034 UTC
BGP routing table entry for 10.6.41.1/32, Route Distinguisher: 44.55.66.77:51
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          2088      2088
Last Modified: Feb 21 08:30:36.850 for 28w2d
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
192.41.1.1 (metric 42) from 10.10.41.1 (192.41.1.5)
  Received Label 2097154
  Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
  imported
  Received Path ID 0, Local Path ID 1, version 2088
```

```

Community: 1234:444
Extended community: 0x0204:1234:444 Encapsulation Type:8 Router
MAC:0200.c029.0101 RT:1234:5101
RIB RNH: table_id 0xe0000190, Encap 8, VNI 2097154, MAC Address: 0200.c029.0101,
IP Address: 192.41.1.1, IP table_id 0x00000000
Source AFI: L2VPN EVPN, Source VRF: default,
Source Route Distinguisher: 192.41.1.5:4123

```

この出力では、受信した RD、ネクストホップ、属性はタイプ2ルートおよびBDサブネットと同じです。

## Multipod\_Fabric

### マルチポッド ファブリックについて

マルチポッドは、隔離されたコントロールプレーンプロトコルを持つ複数のポッドで構成された、障害耐性の高いファブリックのプロビジョニングを可能にします。また、マルチポッドでは、さらに柔軟にリーフとスパインスイッチ間のフルメッシュ配線を行うことができます。たとえば、リーフスイッチが異なるフロアや異なる建物にまたがって分散している場合、マルチポッドでは、フロアごと、または建物ごとに複数のポッドをプロビジョニングし、スパインスイッチを通じてポッド間を接続することができます。

マルチポッドは、異なるポッドの ACI スパイン間のコントロールプレーン通信プロトコルとして MP-BGP EVPN を使用します。

WAN ルータは、スパインスイッチに直接接続されるか、またはボーダーリーフスイッチに接続された IPN 内でプロビジョニングできます。マルチポッドはすべてのポッドに単一の APIC クラスタを使用します。そのため、すべてのポッドが単一のファブリックとして機能します。ポッド全体にわたって個々の APIC コントローラが配置されますが、それらはすべて単一の APIC クラスタの一部です。

### マルチポッド ファブリックでのスイッチの割り当て

#### 始める前に

ノードグループポリシーと L3Out ポリシーがすでに作成されています。

#### 手順

	コマンドまたはアクション	目的
ステップ1	<b>configure</b> 例： apic1# <b>configure</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。

	コマンドまたはアクション	目的
ステップ 2	<p><b>[no] system switch-id serial-number switch-id switch-name [pod pod-id] [role {leaf   spine}]</b></p> <p>例 :</p> <pre>apic1(config)# system switch-id SAL1748H56D 201 ifav4-spine1 pod 1 role spine</pre>	マルチポッドファブリック内の各スイッチに対して、スイッチの関連するポッドとロール（リーフ/スパイン）を宣言します。マルチポッドファブリック内の各リーフとスパインスイッチにこのコマンドを繰り返します。
ステップ 3	<p><b>[no] system pod pod-id tep-pool ip-prefix/length</b></p> <p>例 :</p> <pre>apic1(config)# system pod 1 tep-pool 10.0.0.0/16</pre>	ポッドにトンネルエンドポイント IP アドレスプールを設定します。マルチポッドファブリック内の各ポッドにこのコマンドを繰り返します。

### 例

次に、2 ポッドファブリックでスパインとリーフスイッチを割り当てる例を示します。

```
apic1# configure
apic1(config)# system switch-id SAL1748H56D 201 ifav4-spine1 pod 1 role spine
apic1(config)# system switch-id SAL1938P7A6 202 ifav4-spine3 pod 1 role spine
apic1(config)# system switch-id SAL1819RXP4 101 ifav4-leaf1 pod 1 role leaf
apic1(config)# system switch-id SAL1803L25H 102 ifav4-leaf2 pod 1 role leaf
apic1(config)# system switch-id SAL1934MNY0 103 ifav4-leaf3 pod 1 role leaf
apic1(config)# system switch-id SAL1934MNY3 104 ifav4-leaf4 pod 1 role leaf
apic1(config)# system switch-id SAL1931LA3B 203 ifav4-spine2 pod 2 role spine
apic1(config)# system switch-id FGE173400A9 204 ifav4-spine4 pod 2 role spine
apic1(config)# system switch-id SAL1938PHBB 105 ifav4-leaf5 pod 2 role leaf
apic1(config)# system switch-id SAL1942R857 106 ifav4-leaf6 pod 2 role leaf
apic1(config)# system pod 1 tep-pool 10.0.0.0/16
apic1(config)# system pod 2 tep-pool 10.1.0.0/16
```

### 次のタスク

ファブリックの外部接続を設定します。

## マルチポッドファブリックのファブリックの外部接続の設定

### 始める前に

- ノードグループポリシーと L3Out ポリシーがすでに作成されています。
- スイッチがポッドに割り当てられています。

## 手順

	コマンドまたはアクション	目的
ステップ1	<b>configure</b> 例： apicl# <b>configure</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>[no] fabric-external controller-number</b> 例： apicl(config)# <b>fabric-external 1</b>	
ステップ3	<b>[no]bgp password type automatic_with_full_mesh</b> [ピアリング-パスワード] [{  <b>evpn peeringautomatic_with_rr</b> }] 例： apicl(config-fabric-external)# <b>bgp evpn peering</b>	BGP EVPN ピアリング プロファイルを設定します。ピアリングパスワードを設定できます。また、タイプはフルメッシュに設定するか、ルータリフレクタで設定できます。
ステップ4	<b>[no] pod pod-id</b> 例： apicl(config-fabric-external)# <b>pod 1</b>	設定用のポッドを選択します。
ステップ5	<b>[no] interpod data hardware-proxy ip-addr/mask</b> 例： apicl(config-fabric-external-pod)# <b>interpod data hardware-proxy 100.11.1.1/32</b>	ポッド間トラフィックの各ポートにユニキャストハードウェアプロキシ IP アドレスを設定します。
ステップ6	<b>[no] bgp evpn peering [password peering-password] [type {automatic_with_full_mesh   automatic_with_rr}]</b> 例： apicl(config-fabric-external-pod)# <b>bgp evpn peering</b>	
ステップ7	<b>exit</b> 例： apicl(config-fabric-external-pod)# <b>exit</b>	BGP EVPN ピアリング プロファイルの設定に戻ります。
ステップ8	マルチポッドファブリック内の各ポッドにステップ4～7を繰り返します。	

	コマンドまたはアクション	目的
ステップ 9	<b>[no] route-map interpod-import</b> 例： apic1(config-fabric-external)# <b>route-map interpod-import</b>	OSPF プロトコルを介してファブリック内への参加を許可されるポッド間ネットワーク (IPN) のサブネットを含むルート マップを設定します。
ステップ 10	<b>[no] ip prefix-list prefix-list-name [permit ip-address/len]</b> 例： apic1(config-fabric-external-route-map)# <b>ip prefix-list default permit 0.0.0.0/0</b>	
ステップ 11	<b>exit</b> 例： apic1(config-fabric-external-route-map)# <b>exit</b>	ファブリック外コンフィギュレーションモードに戻ります。
ステップ 12	<b>[no] route-target extended ASN4:NN</b> 例： apic1(config-fabric-external)# <b>route-target extended 5:16</b>	ルートターゲットは拡張コミュニティ属性として伝送されます。AA4:NN2形式でコミュニティ番号を入力します (1-4294967295: 1-65535)。
ステップ 13	<b>exit</b>	

## 例

次に、マルチポッドファブリックにファブリック外部接続を設定する例を示します。

```
apic1# configure
apic1(config)# fabric-external 1
apic1(config-fabric-external)# bgp evpn peering
apic1(config-fabric-external)# pod 1
apic1(config-fabric-external-pod)# interpod data hardware-proxy 100.11.1.1/32
apic1(config-fabric-external-pod)# bgp evpn peering
apic1(config-fabric-external-pod)# exit
apic1(config-fabric-external)# pod 2
apic1(config-fabric-external-pod)# interpod data hardware-proxy 200.11.1.1/32
apic1(config-fabric-external-pod)# bgp evpn peering
apic1(config-fabric-external-pod)# exit
apic1(config-fabric-external)# route-map interpod-import
apic1(config-fabric-external-route-map)# ip prefix-list default permit 0.0.0.0/0
apic1(config-fabric-external-route-map)# exit
apic1(config-fabric-external)# route-target extended 5:16
apic1(config-fabric-external)# exit
```

## 次のタスク

スパイン インターフェイスおよび OSPF を設定します。

# マルチポッドファブリックのスパインインターフェイスと OSPF の設定

## 始める前に

- スイッチがポッドに割り当てられています。
- VLAN ドメインが存在している必要があります。

## 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apicl# <b>configure</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>spine spine-id</b> 例： apicl(config)# <b>spine 104</b>	スパインスイッチは、101～4000 の範囲内の ID 番号か、または「spine1」などの名前指定できます。
ステップ 3	<b>[no] vrf context tenant infra vrf vrf-name</b> 例： apicl(config-spine)# <b>vrf context tenant infra vrf overlay-1</b>	
ステップ 4	<b>[no] router-id A.B.C.D</b> 例： apicl(config-spine-vrf)# <b>router-id 201.201.201.201</b>	ルータ識別子 (ID) を設定します。
ステップ 5	<b>exit</b> 例： apicl(config-spine-vrf)# <b>exit</b>	スパインコンフィギュレーションモードに戻ります。
ステップ 6	<b>[no] interface ethernet slot/port</b> 例： apicl(config-spine)# <b>interface ethernet 1/1</b>	
ステップ 7	<b>[no] vlan-domain member domain-name</b> 例： apicl(config-spine)# <b>vlan-domain member 13Dom</b>	The VLAN domain must already exist, having been created using the <b>vlan-domain domain-name</b> command in the global configuration mode.

	コマンドまたはアクション	目的
ステップ 8	<b>exit</b> 例： apic1(config-spine-if)# <b>exit</b>	スパインコンフィギュレーションモードに戻ります。
ステップ 9	<b>[no] interface ethernet type/slot.subinterface</b> 例： apic1(config-spine)# <b>interface ethernet 1/1.4</b>	サブインターフェイスのカプセル化は 4 である必要があります。
ステップ 10	<b>[no] vrf member tenant infra vrf vrf-name</b> 例： apic1(config-spine-if)# <b>vrf member tenant infra vrf overlay-1</b>	テナント VRF のメンバーとしてインターフェイスを設定します。
ステップ 11	<b>[no] ip address ip-address</b> 例： apic1(config-spine-if)# <b>ip address 201.1.1.1/30</b>	
ステップ 12	<b>[no] ip router ospf default area 0.0.0.0</b> 例： apic1(config-spine-if)# <b>ip router ospf default area 0.0.0.0</b>	
ステップ 13	<b>[no] ip ospf cost cost</b> 例： apic1(config-spine-if)# <b>ip ospf cost 1</b>	
ステップ 14	<b>exit</b> 例： apic1(config-spine-if)# <b>exit</b>	スパインコンフィギュレーションモードに戻ります。
ステップ 15	ステップ 6 (163 ページ) ~ ステップ 14 (164 ページ) を繰り返し、他のインターフェイスを追加します。	
ステップ 16	<b>[no] router ospf default</b> 例： apic1(config-spine)# <b>router ospf default</b>	
ステップ 17	<b>[no] vrf member tenant infra vrf vrf-name</b> 例：	

	コマンドまたはアクション	目的
	<code>apicl(config-spine-ospf)# vrf member tenant infra vrf overlay-1</code>	
ステップ 18	<b>[no] area area loopback ip-address</b> 例： <code>apicl(config-spine-ospf-vrf)# area 0.0.0.0 loopback 201.201.201.201</code>	OSPF を介してループバック アドレスをアドバタイズします。このアドレスは、ピアリングの場合に BGP EVPN セッションで使用されます。
ステップ 19	<b>[no] area area interpod peering</b> 例： <code>apicl(config-spine-ospf-vrf)# area 0.0.0.0 interpod peering</code>	OSPF エリアでポッド間ピアリングを有効にし、OSPF によってアドバタイズされたループバックアドレスを使用して BGP EVPN セッションを自動的に設定します。
ステップ 20	<b>exit</b> 例： <code>apicl(config-spine-ospf-vrf)# exit</code>	OSPF コンフィギュレーション モードに戻ります。
ステップ 21	<b>exit</b> 例： <code>apicl(config-spine-ospf)# exit</code>	スパインコンフィギュレーションモードに戻ります。
ステップ 22	<b>exit</b> 例： <code>apicl(config-spine)# exit</code>	グローバル コンフィギュレーションモードに戻ります。
ステップ 23	ステップ 2 (163 ページ) ~ ステップ 22 (165 ページ) を繰り返し、他のスパインスイッチを設定します。	

## 例

```

apicl# configure

# CONFIGURE FIRST SPINE

apicl(config)# spine 201
apicl(config-spine)# vrf context tenant infra vrf overlay-1
apicl(config-spine-vrf)# router-id 201.201.201.201
apicl(config-spine-vrf)# exit

apicl(config-spine)# interface ethernet 1/1
apicl(config-spine-if)# vlan-domain member l3Dom
apicl(config-spine-if)# exit
apicl(config-spine)# interface ethernet 1/1.4
apicl(config-spine-if)# vrf member tenant infra vrf overlay-1
apicl(config-spine-if)# ip address 201.1.1.1/30
apicl(config-spine-if)# ip router ospf default area 0.0.0.0

```

```

apic1(config-spine-if)# ip ospf cost 1
apic1(config-spine-if)# exit

apic1(config-spine)# interface ethernet 1/2
apic1(config-spine-if)# vlan-domain member l3Dom
apic1(config-spine-if)# exit
apic1(config-spine)# interface ethernet 1/2.4
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1
apic1(config-spine-if)# ip address 201.2.1.1/30
apic1(config-spine-if)# ip router ospf default area 0.0.0.0
apic1(config-spine-if)# ip ospf cost 1
apic1(config-spine-if)# exit

apic1(config-spine)# router ospf default
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 0.0.0.0 loopback 201.201.201.201
apic1(config-spine-ospf-vrf)# area 0.0.0.0 interpod peering
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)# exit

# CONFIGURE SECOND SPINE

apic1(config)# spine 202
apic1(config-spine)# vrf context tenant infra vrf overlay-1
apic1(config-spine-vrf)# router-id 202.202.202.202
apic1(config-spine-vrf)# exit

apic1(config-spine)# interface ethernet 1/2
apic1(config-spine-if)# vlan-domain member l3Dom
apic1(config-spine-if)# exit
apic1(config-spine)# interface ethernet 1/2.4
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1
apic1(config-spine-if)# ip address 202.1.1.1/30
apic1(config-spine-if)# ip router ospf default area 0.0.0.0
apic1(config-spine-if)# ip ospf cost 1
apic1(config-spine-if)# exit

apic1(config-spine)# router ospf default
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 0.0.0.0 loopback 202.202.202.202
apic1(config-spine-ospf-vrf)# area 0.0.0.0 interpod peering
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)# exit

# CONFIGURE ADDITIONAL SPINES

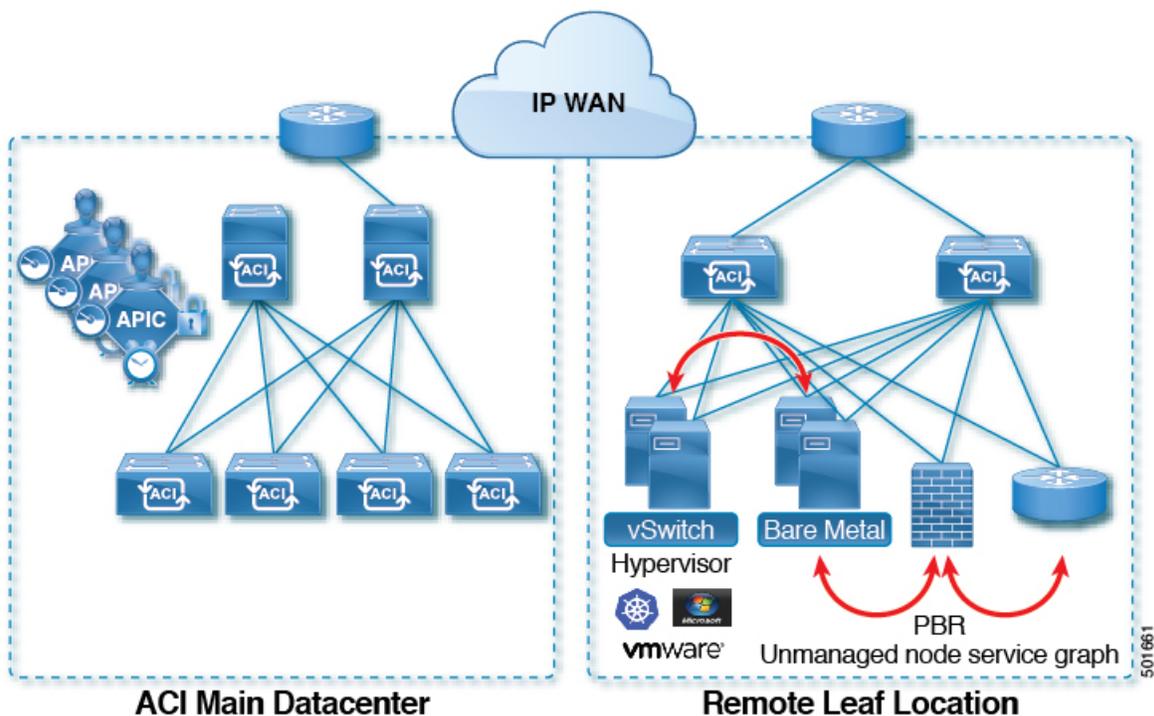
```

## リモート リーフ スイッチ

### ACI ファブリックのリモート リーフ スイッチについて

ACI ファブリックの展開では、ローカル スパイン スイッチまたは APIC が接続されていない Cisco ACI リーフ スイッチのリモート データセンタに、ACI サービスと APIC 管理を拡張できます。

図 10: リモートリーフトポロジ



リモートリーフスイッチがファブリックの既存のポッドに追加されます。メインデータセンタに展開されるすべてのポリシーはリモートスイッチで展開され、ポッドに属するローカルリーフスイッチのように動作します。このトポロジでは、すべてのユニキャストトラフィックはレイヤ3上のVXLANを経由します。レイヤ2ブロードキャスト、不明なユニキャスト、マルチキャスト（BUM）メッセージは、マルチキャストを使用することなく、Head End Replication（HER）トンネルを使用して送信されます。リモートサイトのすべてのローカルトラフィックは、物理または仮想にかかわらずエンドポイント間で直接切り替えられます。スパインスイッチプロキシを使用する必要があるすべてのトラフィックは、メインデータセンタに転送されます。

APICシステムは、起動時にリモートリーフスイッチを検出します。その時点から、ファブリックの一部としてAPICで管理できます。



- (注)
- VRF間のすべてのトラフィックは、転送される前にスパインスイッチに移動します。
  - リモートリーフを解除する前に、vPCを最初に削除する必要があります。

ウィザードを使用するか（使用しない場合も）、REST APIまたはNX-OSスタイルCLIを使用して、APIC GUIのリモートリーフを設定できます。

## リモートのリーフハードウェアの要件

リモートのリーフスイッチの機能には、次のスイッチがサポートされています。

### ファブリック スパインスイッチ

WAN ルータに接続された ACI メイン データ センターにスパイン スイッチでの次のスパイン スイッチがサポートされています。

- 固定スパイン スイッチの Cisco Nexus 9000 シリーズ N9K C9364C  
N9K-X9732C-EX または N9K-X9736C-FX ラインカードをモジュラ スパイン スイッチ
- 古い生成スパイン スイッチは、固定スパイン スイッチ N9K C9336PQ または N9K X9736PQ ラインカードでモジュラ スパイン スイッチなどのメイン データ センターではサポートが次世代のみのスパイン スイッチは、WAN への接続をサポートします。

### リモート リーフ スイッチ

- リモートのリーフ スイッチ、後で (たとえば N9K-C93180LC-EX) EX で終了する名前と Cisco Nexus 9000 シリーズ スイッチのみがサポートされています。



(注) Cisco Nexus 9000 N9K-C9336C-FX スイッチは、リモートのリーフ スイッチのサポートされていません。

- リモートのリーフ スイッチする必要がありますにイメージを実行する、スイッチ 13.1.x 以降 (aci n9000 dk9.13.1.x.x.bin) 検出できる前にします。これにより、リーフ スイッチでの手動アップグレードが必要があります。

## 制約事項と制限



- (注) Cisco APIC のリリースでは、以前にサポートされていませんでした 3.2(x) が、次の機能がサポートされます。
- リモートのリーフ スイッチに接続されている FEX デバイス
  - リモートのリーフ スイッチまたはリモート リーフ スイッチとローカル リーフ スイッチ間でという原子カウンタという
  - Cisco VXLAN での VLAN とシスコの AV で AV
  - VXLAN での VLAN と ACI の仮想エッジで Cisco ACI 仮想エッジ

リモート リーフ機能では、次の導入と設定がサポートされていません。

- APIC コントローラは、リモートのリーフ スイッチに直接接続

- VPC ドメインでのリモートリーフスイッチで孤立ポートチャンネルまたは物理ポート
- コンシューマ、プロバイダー、およびサービスノードがすべてスイッチvPCモードでは、リモートのリーフに接続されている場合に、リモートロケーション内で転送ローカルトラフィックがサポートのみとサービスノードの統合がなければ、

このリリースでは、次の機能を除くで、リモートのリーフスイッチでは、ファブリックおよびテナントの完全なポリシーがサポートされています。

- ACI マルチサイト
- レイヤ2 (スタティック Epg) を除く接続外部
- 802.1 q トンネリング
- EPG のための Q-in-Q カプセル化マッピング
- VzAny 契約とサービスをコピーします。
- リモートのリーフスイッチの FCoE 接続
- ブリッジドメインまたは Epg のカプセル化をフラグディングします。
- 高速リンク フェールオーバー ポリシー
- 遠隔地での管理対象のサービス グラフに接続されたデバイス
- NetFlow
- トラフィック ストーム制御
- クラウド秒および MacSec 暗号化
- ファーストホップセキュリティ
- PTP
- レイヤ3 マルチキャスト リモートリーフスイッチ上のルーティング
- リモートのリーフスイッチでの PBR トラッキング
- Openstack および Kubernetes VMM ドメイン
- メンテナンス モード
- ウィザードのトラブルシューティング
- 遠隔地での中継 L3Out
- 同じリモートデータセンターで同じポッドおよびポッド全体の2つのリモートリーフスイッチ間で直接トラフィックの転送

## WAN ルータとリモートリーフ設定の注意事項

リモートリーフが検出され APIC 管理に組み込まれる前に、WAN ルータとリモートリーフスイッチを設定する必要があります。

次の要件に従い、ファブリック スパインスイッチの外部インターフェイスとリモートリーフスイッチポートに接続する WAN ルータを接続します。

### WAN ルータ

- エリア ID、タイプ、コストなど、同じ詳細を有するインターフェイスで OSPF を有効にします。
- メインファブリックの各 APIC の IP アドレスにつながるインターフェイスで DHCP リレーを設定します。
- スパインスイッチで VLAN 5 インターフェイスに接続する WAN ルータのインターフェイスは、通常のマルチポッドネットワークに接続するインターフェイス以外に、異なる VRF に存在する必要があります。

### リモートリーフスイッチ

- ファブリックポートの 1 つから直接接続して、アップストリームルータにリモートリーフスイッチを接続します。アップストリームルータへの次の接続がサポートされています。
  - 40 Gbps 以上の接続
  - QSFP-SFP アダプタでは、1/10 G SFP がサポートされています

WAN の帯域幅は最小で 100 Mbps、最大。サポートされている遅延は 300 ミリ秒です。

- 上記が推奨されますが、vPC とリモートリーフスイッチのペアを接続する必要はありません。vPC の両端にあるスイッチは、同じリモートデータセンターのリモートリーフスイッチである必要があります。
- 一意の IP アドレスを持つ VLAN 4 でレイヤ 3 サブインターフェイスとしてノースバウンドインターフェイスを設定します。

リモートのリーフスイッチからルータに 1 個以上のインターフェイスを接続する場合、一意の IP アドレスで各インターフェイスを設定します。
- インターフェイスで OSPF を有効にします。
- リモートリーフスイッチ内の TEP プールサブネットの IP アドレスは、ポッド TEP サブネットプールと重複しないようにする必要があります。使用されるサブネットは /24 以下である必要があります。
- マルチポッドがサポートされますが、リモートリーフ機能は必要ありません。
- 単一ポッドファブリックのポッドをリモートリーフスイッチに接続するとき、スパインスイッチから WAN ルータへ、リモートリーフスイッチから WAN ルータへ L3Out を設定し、これは両方ともスイッチインターフェイスで VLAN-4 を使用します。

- マルチポッドファブリックのポッドをリモートリーフスイッチに接続するとき、スパインスイッチから WAN ルータへ、リモートリーフスイッチから WAN ルータへ L3Out を設定し、これは両方ともスイッチインターフェイスで VLAN-4 を使用します。また、VLAN-5 を使用してマルチポッド内部 L3Out を設定し、リモートリーフスイッチを宛先としてポッドを通過するトラフィックをサポートします。VLAN 4 および VLAN 5 を使用する限り、通常のマルチポッドおよびマルチポッド内部接続は、同じ物理インターフェイスで設定できます。
- マルチポッド内部 L3Out を設定している場合、通常のマルチポッド L3Out として同じルータ ID を使用しますが、ルータ ID の [ループバックアドレスとしてルータ ID を使用する] オプションを選択解除して、異なるループバック IP アドレスを設定します。これで ECMP が機能します。

## NX-OS スタイル CLI を使用したリモートリーフの設定

この例では、リーフスイッチがメインのファブリックポッドと通信できるようにするため、スパインスイッチとリモートリーフスイッチを設定しています。

### 始める前に

- IPN ルータとリモートスイッチはアクティブで設定されています。 [WAN ルータとリモートリーフ設定の注意事項 \(170 ページ\)](#) を参照してください。
- リモートリーフスイッチは、13.1.x 以降 (aci n9000 dk9.13.1.x.x.bin) のスイッチイメージを実行しています。
- リモートリーフスイッチを追加する予定のポッドが作成され、設定されています。

### 手順

**ステップ 1** ポッド 2 のリモートロケーション 5 で TEP プールを定義します。

ネットワークマスクは /24 以下である必要があります。

次の新しいコマンドを使用します: **system remote-leaf-site site-id pod pod-id tep-pool ip-address-and-netmask**

例:

```
apicl(config)# system remote-leaf-site 5 pod 2 tep-pool 192.0.0.0/16
```

**ステップ 2** ポッド 2 の、リモートリーフサイト 5 にリモートリーフスイッチを追加します。

次のコマンドを使用します: **system switch-id serial-number node-id leaf-switch-namepod pod-id remote-leaf-site remote-leaf-site-id node-type remote-leaf-wan**

例:

```
apicl(config)# system switch-id FDO210805SKD 109 ifav4-leaf9 pod 2
remote-leaf-site 5 node-type remote-leaf-wan
```

**ステップ3** VLAN 4 を含む VLAN で VLAN ドメインを設定します。

例：

```
apic1(config)# vlan-domain ospfDom
apic1(config-vlan)# vlan 4-5
apic1(config-vlan)# exit
```

**ステップ4** インフラ テナントに2つの L3Out を設定します。1つはリモート リーフ接続のために、もう1つはマルチポッド IPN のためです。

例：

```
apic1(config)# tenant infra
apic1(config-tenant)# l3out rl-wan
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# l3out ipn-multipodInternal
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# exit
apic1(config)#
```

**ステップ5** L3Out が使用する、スパイン スイッチ インターフェイスとサブインターフェイスを設定します。

例：

```
apic1(config)# spine 201
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-vrf)# exit
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-vrf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36
apic1(config-spine-if)# vlan-domain member ospfDom
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf default
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.4
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-if)# ip router ospf default area 5
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf multipod-internal
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out ipn-multipodInternal
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.5
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-if)# ip router ospf multipod-internal area 5
apic1(config-spine-if)# exit
apic1(config-spine)# exit
apic1(config)#
```

**ステップ6** メインのファブリックポッドと通信するために使用するリモートのリーフスイッチインターフェイスとサブインターフェイスを設定します。

例：

```
(config)# leaf 101
apicl(config-leaf)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apicl(config-leaf-vrf)# exit
apicl(config-leaf)#
apicl(config-leaf)# interface ethernet 1/49
apicl(config-leaf-if)# vlan-domain member ospfDom
apicl(config-leaf-if)# exit
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant infra vrf overlay-1
apicl(config-leaf-ospf-vrf)# area 5 l3out rl-wan-test
apicl(config-leaf-ospf-vrf)# exit
apicl(config-leaf-ospf)# exit
apicl(config-leaf)#
apicl(config-leaf)# interface ethernet 1/49.4
apicl(config-leaf-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apicl(config-leaf-if)# ip router ospf default area 5
apicl(config-leaf-if)# exit
```

例

次の例は、ダウンロード可能な設定を示しています：

```
apicl# configure
apicl(config)# system remote-leaf-site 5 pod 2 tep-pool 192.0.0.0/16
apicl(config)# system switch-id FDO210805SKD 109 ifav4-leaf9 pod 2
apicl(config)# remote-leaf-site 5 node-type remote-leaf-wan
apicl(config)# vlan-domain ospfDom
apicl(config-vlan)# vlan 4-5
apicl(config-vlan)# exit
apicl(config)# tenant infra
apicl(config-tenant)# l3out rl-wan-test
apicl(config-tenant-l3out)# vrf member overlay-1
apicl(config-tenant-l3out)# exit
apicl(config-tenant)# l3out ipn-multipodInternal
apicl(config-tenant-l3out)# vrf member overlay-1
apicl(config-tenant-l3out)# exit
apicl(config-tenant)# exit
apicl(config)#
apicl(config)# spine 201
apicl(config-spine)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apicl(config-spine-vrf)# exit
apicl(config-spine)# vrf context tenant infra vrf overlay-1 l3out ipn-multipodInternal
apicl(config-spine-vrf)# exit
apicl(config-spine)#
apicl(config-spine)# interface ethernet 8/36
apicl(config-spine-if)# vlan-domain member ospfDom
apicl(config-spine-if)# exit
apicl(config-spine)# router ospf default
apicl(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apicl(config-spine-ospf-vrf)# area 5 l3out rl-wan-test
apicl(config-spine-ospf-vrf)# exit
apicl(config-spine-ospf)# exit
apicl(config-spine)#
apicl(config-spine)# interface ethernet 8/36.4
apicl(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
```

```

apic1(config-spine-if)# ip router ospf default area 5
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf multipod-internal
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out ipn-multipodInternal
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.5
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-if)# ip router ospf multipod-internal area 5
apic1(config-spine-if)# exit
apic1(config-spine)# exit
apic1(config)#
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-vrf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49
apic1(config-leaf-if)# vlan-domain member ospfDom
apic1(config-leaf-if)# exit
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-leaf-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49.4
apic1(config-leaf-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-if)# ip router ospf default area 5
apic1(config-leaf-if)# exit

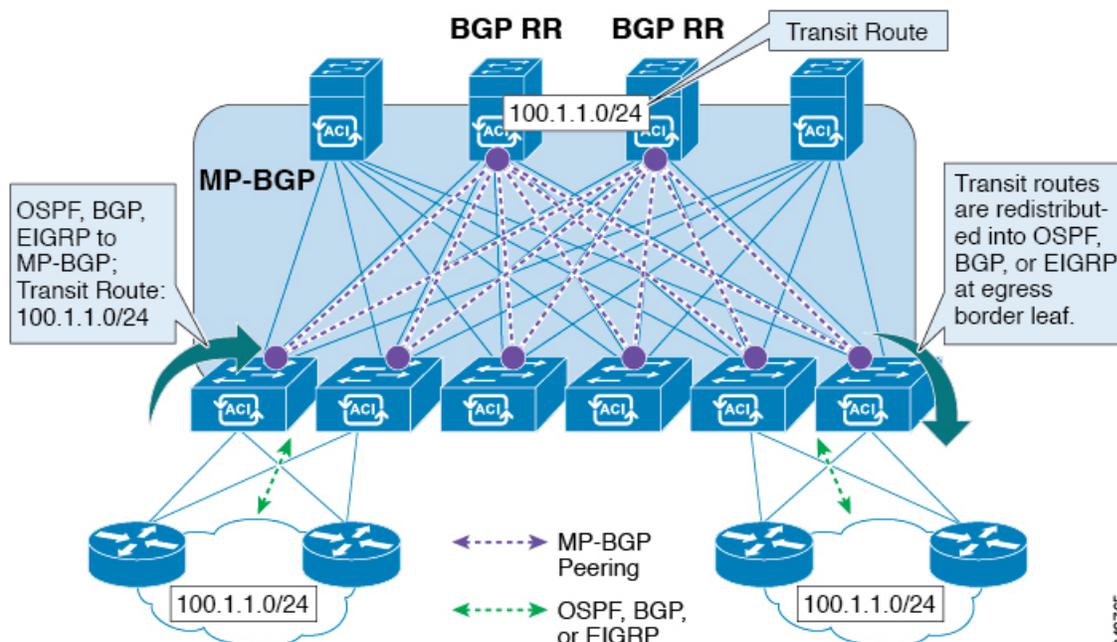
```

## トランジットルーティング

### 中継 ACI ファブリックのルーティング

Cisco APIC ソフトウェアは、OSPF (NSSA) および iBGP を使用した外部レイヤ3 接続をサポートします。ファブリックは、外部レイヤ3 アウトサイド (l3out) 接続の外部ルータにテナントブリッジドメインのサブネットをアドバタイズします。外部ルータから学習されたルートは、他の外部ルータにアドバタイズされません。ファブリックはスタブネットワークと同じように動作し、外部レイヤ3 ドメイン間のトラフィックの伝送に使用できます。

図 11: ファブリックでルーティング中継



3-497/65

中継のルーティングで1つのテナントとVRF内の複数のL3Out接続がサポートされているし、APICは別のL3Out接続を1つのL3Out接続から学習したルートを実バタイズします。外部レイヤ3ドメインは、境界リーフスイッチのファブリックとピアリングします。ファブリックはピア間の Multiprotocol-Border Gateway Protocol (MP-BGP) 中継ドメインです。

外部L3Out接続の設定は、テナントとVRFレベルで実行されます。外部ピアから学習したルートは、VRFごとに入力リーフのMP-BGPにインポートされます。L3Out接続から学習したプレフィックスは、テナントVRFが存在するリーフスイッチにのみエクスポートされます。



(注) 注意事項と中継ルーティングの設定のガイドラインは、次を参照してください。 [中継ルーティングのガイドライン \(177 ページ\)](#)

## トピックに関連するトランジットルーティング

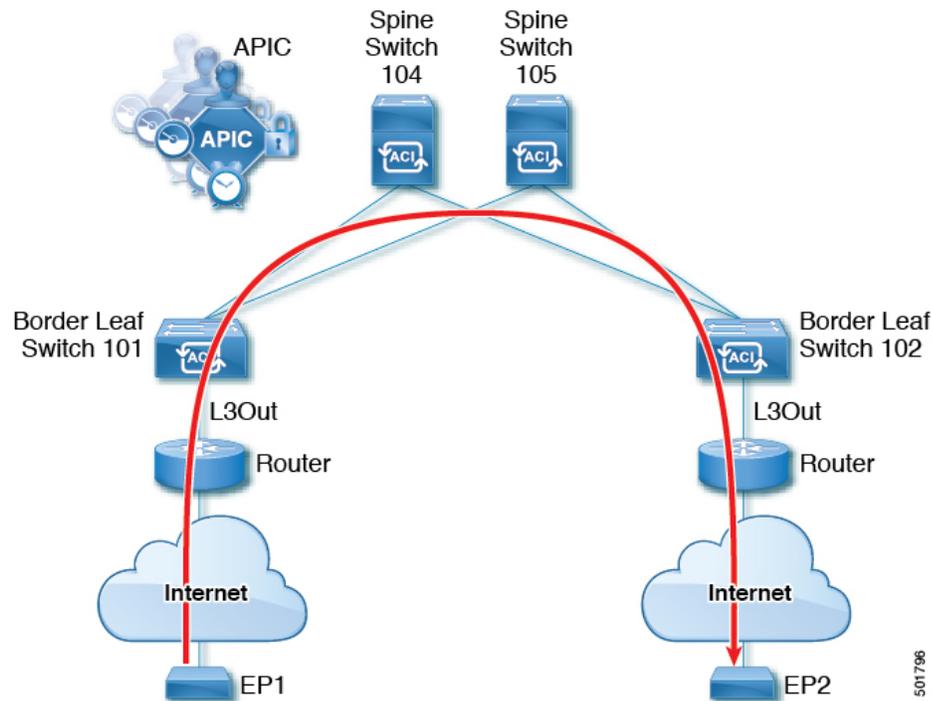
トランジットルーティングの概要、使用例、注意事項、トランジットルーティングのルート制御情報に関する詳細は、「Cisco APIC レイヤ3 ネットワーキング 設定ガイド」を参照してください。

## トランジットルーティングの概要

このトピックでは、Cisco APIC を使用する際のトランジットルーティングを設定する方法の一般的な例を説明します。

この章にある例では、次のトポロジを使用します。

図 12:



この章の例では、Cisco ACI ファブリックには APIC クラスタによって制御される 2 個のリーフスイッチと 2 個のスパインスイッチがあります。境界リーフスイッチ 101 と 102 には L3Out があり、2 つのルータに接続することでインターネットにも接続しています。この例の目標は、2 つの L3Out を通して、インターネット上の EP1 から EP2 へ、ファブリック内外をトラフィックが行き来できるようにすることです。

この例では、両方の L3Out に関連付けられているテナントは、t1 と、VRF がつく v1 です。

L3Out を設定する前に、ノード、ポート、機能プロファイル、AEP、レイヤ 3 ドメインを設定します。BGP ルートリフレクタとして 104 と 105 スパインスイッチを設定する必要があります。

トランジットルーティングの設定には、次のコンポーネントの定義が含まれます。

1. テナントおよび VRF
2. リーフ 101 と 102 上のノードおよびインターフェイス
3. 各 L3Out のプライマリルーティングプロトコル（境界リーフスイッチと外部ルータ間のルートの交換に使用。この例では BGP）
4. 各 L3Out のルーティングプロトコルの接続性（プライマリプロトコルへの到達可能性情報の提供。この例では、OSPF）
5. 2 個の外部 EPG
6. 1 個のルートマップ

7. 少なくとも 1 つのフィルタと 1 つのコントラクト
8. 外部 EPG とコントラクトを関連付ける



(注) トランジットルーティングの注意事項については、[中継ルーティングのガイドライン \(177 ページ\)](#) を参照してください。

次の表では、この章で使用される名前を一覧にしています。

プロパティ	ノード 101 の L3Out1 の名前	ノード 102 の L3Out2 の名前
テナント	t1	t1
VRF	v1	v1
ノード	ルータ ID 11.11.11.103 を持つ nodep1	ルータ ID 22.22.22.203 を持つ nodep2
OSPF インターフェイス	Eth/1/3 の ifp1	Eth/1/3 の ifp2
BPG ピア アドレス	15.15.15.2/24	25.25.25.2/24
外部 EPG	192.168.1.0/24 の extnw1	192.168.2.0/24 の extnw2
ルート マップ	Ctx1 を持つ rp1 とルートの宛先 192.168.1.0/24	Ctx2 を持つ rp2 とルートの宛先 192.168.2.0/24
フィルタ	http-filter	http-filter
コントラクト	extnw1 によって提供される httpCtrct	extnw2 によって消費される httpCtrct

## 中継ルーティングのガイドライン

作成し、中継ルーティング接続を維持する場合は、次のガイドラインを使用します。

トピック	注意またはガイドライン
中継が1つのL3Outプロファイルを使用してルーティング	

トピック	注意またはガイドライン
	<p>前に APIC2.3(1f) のリリースでは、ルーティングを通過、1 つの L3Out プロファイル内ではサポートされていませんでした。APIC2.3(1f) のリリース、および以降では、中継が単一 L3Out プロファイルで、次の制限でルーティングを設定できます。</p> <ul style="list-style-type: none"> <li>• VRF が適用されるではない場合は、同じ L3EPG を共有ルータ間のトラフィックを許可する 0.0.0.0/0 の外部のサブネット (l3extSubnet) を使用できます。</li> <li>• 外部デフォルトサブネット (0.0.0.0/0) の場合は、VRF を適用すると、<b>できません</b> 同一のレイヤ 3 EPG 内のトラフィックの送信元と宛先の両方のプレフィックスに一致するために使用します。同一のレイヤ 3 EPG 内のすべてのトラフィックを一致するには、次のプレフィックスがサポートされています。 <ul style="list-style-type: none"> <li>• <b>IPv4</b> <ul style="list-style-type: none"> <li>• 外部 EPG の 0.0.0.0/1—with 外部サブネット</li> <li>• 外部 EPG の 128.0.0.0/1—with 外部サブネット</li> <li>• 0.0.0.0/0—with インポートルート制御サブネット、集約のインポート</li> </ul> </li> <li>• <b>IPv6</b> <ul style="list-style-type: none"> <li>• 0::0/1: 外部 EPG の外部のサブネットを持つ</li> <li>• 8000::0/1: 外部 EPG の外部のサブネットを持つ</li> <li>• 0:0/0: インポートルート制御のサブネットで集約のインポート</li> </ul> </li> </ul> </li> <li>• また、VzAny 契約と組み合わせると、1 つのデフォルトのサブネット (0.0.0.0/0) を使用できます。次に例を示します。 <ul style="list-style-type: none"> <li>• 使用契約とレイヤ 3 EPG の提供 VzAny 消費契約 (一致する 0.0.0.0/0)、または、VzAny には、契約とレイヤ 3 EPG が提供される契約 (0.0.0.0/0 に一致する) が使用された行数。</li> <li>• サブネット 0.0.0.0/0—with インポート/エクスポートルート制御サブネット、集約のインポートおよび集約エクスポートを使用します。</li> </ul> </li> </ul>

トピック	注意またはガイドライン
ハードウェア サポートの違いを共有ルート:	<p>第2世代のスイッチの VRF 機能間で正常にルートが共有されます (N9K-93108TC-EX など、スイッチモデル名の最後やその後に「EX」や「FX」がつく Cisco Nexus N9K)。第1世代のスイッチですが、ルートを保存する物理的な3進コンテンツ対応メモリ (TCAM) にルートの解析を完全にサポートするだけの容量がないため、この設定のパケットは失敗する可能性があります。</p>
背面に戻る設定で EIGRP や OSPF	<p>Cisco APIC では、中継が、L3Out に設定されているエクスポートルート制御ポリシーのルーティングをサポートします。ルート(プレフィックス)を通過するこれらのポリシー制御は、L3Out でルーティングプロトコルに再配送されます。これらの中継ルートは、EIGRP や OSPF に再配布されたが、これらはルーティングループを防ぐためにタグ付けされた 4294967295 です。Cisco ACI ファブリックは、OSPF または EIGRP L3Out で学習すると、このタグに一致するルートを受け入れません。ただし、この場合、この動作をオーバーライドする必要があります。</p> <ul style="list-style-type: none"> <li>• EIGRP や OSPF を使用して、2つの Cisco ACI ファブリックを接続します。</li> <li>• EIGRP や OSPF を使用して、同じ Cisco ACI ファブリックで2つの異なる Vrf を接続します。</li> </ul> <p>APIC GUI の次の場所にある別のタグのポリシーを使用して、VRF を設定する必要がありますをオーバーライドする必要がある場所: テナント &gt; <i>Tenant_name</i> &gt; ネットワーキング &gt; プロトコル ポリシー &gt; ルート タグ . 異なるタグを適用します。</p> <p>新しいルート タグ ポリシーを作成するには、だけでなく、APIC GUI の次の場所でこのポリシーを使用する VRF を更新: テナント &gt; <i>Tenant_name</i> &gt; ネットワーキング &gt; Vrf &gt; <i>Tenant_VRF</i> . VRF を作成したルート タグ ポリシーを適用します。</p> <p>(注) 複数 L3Outs または同じ L3Out で複数のインターフェイスは同じリーフスイッチに導入し、中継ルーティングに使用、(、IGP に再配布されません) IGP 内で、ルートをアドバタイズします。この状況では、ルート タグ ポリシーは適用されません。</p>

トピック	注意またはガイドライン
BD サブネットをファブリック外にアドバタイズする	<p>インポートおよびエクスポートのルート制御ポリシーは、中継ルート（他の外部ピアから学習したルート）およびスタティックルートのみ適用されます。テナント BD サブネット上に設定されているファブリック内部のサブネットは、エクスポート ポリシー サブネットを使用して外部にアドバタイズされません。IP プレフィックスリストおよびルート マップを使用すると IP テナントサブネットは許可されますが、これらは別の設定手順を使用して実装されます。テナントサブネットをファブリックの外部にアドバタイズする場合は、次の設定手順を参照してください。</p> <ol style="list-style-type: none"> <li>1. [subnet properties] ウィンドウで、テナント サブネットの範囲を [Public Subnet] として設定します。</li> <li>2. オプション。[subnet properties] ウィンドウで、[Subnet Control] を [ND RA Prefix] として設定します。</li> <li>3. テナントブリッジドメイン (BD) を外部レイヤ 3 Outside に関連付けます (L3Out)。</li> <li>4. テナント EPG と外部 EPG 間にコントラクト (プロバイダ/コンシューマ) の関連付けを作成します。</li> </ol> <p>BD サブネットを Public 範囲に設定し、BD をレイヤ 3 Out に関連付けると、BD サブネットプレフィックスの境界リーフに IP プレフィックスおよびルートマップの連続エントリが作成されます。</p>

トピック	注意またはガイドライン
デフォルト ルートのアドバタイズ	<p>デフォルトルートのみを必要とするファブリックへの外部接続の場合、OSPF、EIGRP、およびBGPのL3Out接続をデフォルトルートの起点とすることがサポートされます。外部ピアからデフォルトルートが受信されると、この文書で説明されている中継エクスポートルート制御に従って、このルートを別のピアに再配布できます。</p> <p>デフォルトルートは、デフォルトルートリークポリシーを使用してアドバタイズすることもできます。このポリシーは、デフォルトルートがルーティングテーブル内にあるか、または常にデフォルトルートをアドバタイズすることがサポートされている場合、デフォルトルートのアドバタイズをサポートします。デフォルトルートリークポリシーは、L3Out接続で設定されます。</p> <p>デフォルトルートリークポリシーを作成するときは、以下のガイドラインに従ってください:</p> <ul style="list-style-type: none"> <li>• BGPの場合、<b>Always</b>プロパティは適用されません。</li> <li>• BGPの場合、<b>Scope</b>プロパティを設定するときには、<b>Outside</b>を選択します。</li> <li>• OSPFの場合、範囲の値が<b>Context</b>だとタイプ5LSAが作成されるのに対し、<b>Outside</b>だとタイプ7LSAが作成されます。選択したは、L3Outで設定されたエリアのタイプによって異なります。エリアタイプが場合 <b>定期的な</b>、範囲を設定します <b>コンテキスト</b>。エリアタイプが場合 <b>NSSA</b>、範囲を設定します <b>外部</b>。</li> <li>• EIGRPで、<b>Scope</b>プロパティを選択する場合には、<b>Context</b>を選択する必要があります。</li> </ul>

トピック	注意またはガイドライン
MTU	<p>Cisco ACI は、IP フラグメンテーションをサポートしていません。したがって、外部ルータへのレイヤ3 Outside (L3Out) 接続、または Inter-Pod Network (IPN) を介した multipod 接続を設定する場合は、MTU が両側で適切に設定されていることが重要です。ACI、Cisco NX-OS、Cisco IOS などの一部のプラットフォームでは、設定された MTU 値は IP ヘッダーを考慮に入れています（結果として、最大パケットサイズは、ACI で 9216 バイト、NX-OS および IOS で 9000 バイトに設定されます）。ただし、IOS XR などの他のプラットフォームは、パケットヘッダーのを除く MTU 値を設定します（結果として最大パケットサイズは 8986 バイトになります）。</p> <p>各プラットフォームの適切な MTU 値については、それぞれの設定ガイドを参照してください。</p> <p>CLI ベースのコマンドを使用して MTU をテストすることを強く推奨します。たとえば、Cisco NX-OS CLI で <code>ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1</code> などのコマンドを使用します。</p>

## NX-OS スタイル CLI を使用したトランジットルーティングの設定

次の手順では、テナントのトランジットルーティングを設定する方法を説明します。この例では、別々にルータに接続された2つの境界リーフスイッチ上の1個のVRFで、2個のL3Outsを展開します。

### 始める前に

- ノード、ポート、AEP、機能プロファイル、レイヤ3ドメインを設定します。
- 使用してVLANドメイン設定、`vlan`ドメインドメイン および `vlan` `vlan` 範囲 コマンド。
- BGP ルート リフレクタ ポリシーを設定し、ファブリック内でルーテッドを伝達します。

これらの前提条件のコマンド例については、[NX-OS スタイル CLI の例: L3Out の前提条件 \(14 ページ\)](#) を参照してください。

### 手順

**ステップ1** テナントおよびVRFを設定します。

この例ではVRF `v1` でテナント `t1` を設定します。VRFはまだ展開されていません。

例：

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
```

**ステップ2** ノードおよびインターフェイスを設定します。

この例では、2つの境界リーフスイッチでテナント t1 の2つの L3Outs を設定します。

- 最初の L3Out はノード 101 上にあり、nodep1 という名前です。ノード 101 はルータ ID 11.11.11.103 で設定されます。ルーテッドインターフェイス ifp1 が eth1/3 にあり、IP アドレス 12.12.12.3/24 です。
- 2番目の L3Out がノード 102 上にあり、nodep2 という名前です。ノード 102 はルータ ID 22.22.22.203 で設定されます。ルーテッドインターフェイス ifp2 が eth1/3 に存在し、IP アドレスは 23.23.23.1/24 です。

例：

```
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 11.11.11.103
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 12.12.12.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 22.22.22.203
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 23.23.23.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

**ステップ3** 両方のリーフスイッチのルーティングプロトコルを設定します。

この例では、両方の境界リーフスイッチに対して、ASN 100 でプライマリ ルーティングプロトコルとして BGP を設定します。BGP ピア 15.15.15.2 を持つノード 101 と BGP ピア 25.25.25.2 を持つノード 102 を設定します。

例：

```
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit
```

```
apicl(config)# leaf 102
apicl(config-leaf)# router bgp 100
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 25.25.25.2
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# exit
```

#### ステップ4 接続ルーティングプロトコルを設定します。

この例では、定期的なエリアID 0.0.0.0 で両方の L3Outs に対して通信プロトコルとして OSPF を設定します。

例：

```
apicl(config)# leaf 101
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant t1 vrf v1
apicl(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 40.40.40.1
apicl(config-leaf-ospf-vrf)# exit
apicl(config-leaf-ospf)# exit
apicl(config-leaf)# exit
apicl(config)# leaf 102
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant t1 vrf v1
apicl(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 60.60.60.1
apicl(config-leaf-ospf-vrf)# exit
apicl(config-leaf-ospf)# exit
apicl(config-leaf)# exit
```

#### ステップ5 外部 EPG を設定します。

この例では、ネットワーク 192.168.1.0/24 をノード 101 上の外部ネットワーク extnw1 として、ネットワーク 192.168.2.0/24 をノード 102 上の外部ネットワーク extnw2 として設定します。

例：

```
apicl(config)# tenant t1
apicl(config-tenant)# external-l3 epg extnw1
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# match ip 192.168.1.0/24
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# external-l3 epg extnw2
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# match ip 192.168.2.0/24
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# external-l3 epg extnw1
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# exit
apicl(config)# leaf 102
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# external-l3 epg extnw2
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# exit
```

#### ステップ6 オプション。ルートマップを設定します。

この例では、インバウンドおよびアウトバウンド方向で各 BGP ピアのルートマップを設定します。

例：

例：

```

apic1(config)# leaf 101
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp2 out
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 102
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 25.25.25.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp2 in
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 out
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

```

**ステップ 7** フィルタ（アクセスリスト）およびコントラクトを作成し、EPGが通信できるようにします。

例：

```

apicl(config)# tenant t1
apicl(config-tenant)# access-list http-filter
apicl(config-tenant-acl)# match ip
apicl(config-tenant-acl)# match tcp dest 80
apicl(config-tenant-acl)# exit
apicl(config-tenant)# contract httpCtrct
apicl(config-tenant-contract)# scope vrf
apicl(config-tenant-contract)# subject subj1
apicl(config-tenant-contract-subj)# access-group http-filter both
apicl(config-tenant-contract-subj)# exit
apicl(config-tenant-contract)# exit
apicl(config-tenant)# exit

```

**ステップ8** コントラクトを設定し、EPGに関連付けます。

例：

```

apicl(config)# tenant t1
apicl(config-tenant)# external-l3 epg extnw1
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# contract provider httpCtrct
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# external-l3 epg extnw2
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# contract consumer httpCtrct
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# exit
apicl(config)#

```

## 例：中継ルーティング

この例では、中継ルーティングのマージされた設定を提供します。設定は別々のルータに接続されている2個の障壁リーフスイッチで、2つのL3Outsを持つ単一のテナントとVRFのためにあります。

```

apicl# configure
apicl(config)# tenant t1
apicl(config-tenant)# vrf context v1
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# exit

apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# router-id 11.11.11.103
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# interface ethernet 1/3
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vrf member tenant t1 vrf v1
apicl(config-leaf-if)# ip address 12.12.12.3/24
apicl(config-leaf-if)# exit
apicl(config-leaf)# router bgp 100
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant t1 vrf v1

```

```

apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 40.40.40.1
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 22.22.22.203
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 23.23.23.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 25.25.25.2/24
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 60.60.60.3
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit

apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 192.168.1.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# external-l3 epg extnw2
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit

apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw1
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw2
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 101
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0

```

```
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# router bgp 100
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp2 out
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# exit

apicl(config)# leaf 102
apicl(config-leaf)# template route group match-rule1 tenant t1
apicl(config-route-group)# ip prefix permit 192.168.1.0/24
apicl(config-route-group)# exit
apicl(config-leaf)# template route group match-rule2 tenant t1
apicl(config-route-group)# ip prefix permit 192.168.2.0/24
apicl(config-route-group)# exit
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# route-map rp1
apicl(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# route-map rp2
apicl(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apicl(config-leaf-vrf-route-map-match)# exit
apicl(config-leaf-vrf-route-map)# exit
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# router bgp 100
apicl(config-leaf-bgp)# vrf member tenant t1 vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 25.25.25.2
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp2 in
apicl(config-leaf-bgp-vrf-neighbor)# route-map rp1 out
apicl(config-leaf-bgp-vrf-neighbor)# exit
apicl(config-leaf-bgp-vrf)# exit
apicl(config-leaf-bgp)# exit
apicl(config-leaf)# exit

apicl(config)# tenant t1
apicl(config-tenant)# access-list http-filter
apicl(config-tenant-acl)# match ip
apicl(config-tenant-acl)# match tcp dest 80
apicl(config-tenant-acl)# exit
apicl(config-tenant)# contract httpCtrct
apicl(config-tenant-contract)# scope vrf
apicl(config-tenant-contract)# subject http-subj
apicl(config-tenant-contract-subj)# access-group http-filter both
apicl(config-tenant-contract-subj)# exit
apicl(config-tenant-contract)# exit
apicl(config-tenant)# exit

apicl(config)# tenant t1
apicl(config-tenant)# external-l3 epg extnw1
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# contract provider httpCtrct
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# external-l3 epg extnw2
apicl(config-tenant-l3ext-epg)# vrf member v1
apicl(config-tenant-l3ext-epg)# contract consumer httpCtrct
apicl(config-tenant-l3ext-epg)# exit
apicl(config-tenant)# exit
apicl(config)#
```

■ 例：中継ルーティング