



NX-OS スタイル CLI を使用したタスクの実行

- [Part I : レイヤ 3 の設定 \(1 ページ\)](#)
- [パートII : 外部ルーティング \(L3Out\) の設定 \(32 ページ\)](#)

Part I : レイヤ 3 の設定

NX-OS スタイルの CLI を使用した共通パーベシブ ゲートウェイの設定

NX-OS スタイルの CLI を使用した共通パーベシブ ゲートウェイの設定

始める前に

- テナント、VRF、およびブリッジ ドメインが作成されていること。

手順

共通パーベシブ ゲートウェイを設定します。

例 :

```
apicl#configure
apicl(config)#tenant demo
apicl(config-tenant)#bridge-domain test
apicl(config-tenant-bd)#l2-unknown-unicast flood
apicl(config-tenant-bd)#arp flooding
apicl(config-tenant-bd)#exit

apicl(config-tenant)#interface bridge-domain test
apicl(config-tenant-interface)#multi-site-mac-address 12:34:56:78:9a:bc
apicl(config-tenant-interface)#mac-address 00:CC:CC:CC:C1:01 (Should be unique for each ACI fabric)
```

```
apic1(config-tenant-interface)#ip address 192.168.10.1/24 multi-site
apic1(config-tenant-interface)#ip address 192.168.10.254/24 (Should be unique for each ACI fabric)
```

NX-OS Style CLI を使用した IP エージングの設定

NX-OS スタイル CLI を使用した IP エージング ポリシーの設定

このセクションでは、CLI を使用した IP エージング ポリシーを有効および無効にする方法を説明します。

手順

ステップ 1 IP エージング ポリシーを有効にするには：

例：

```
ifc1(config)# endpoint ip aging
```

ステップ 2 IP エージング ポリシーを無効にするには：

例：

```
ifav9-ifc1(config)# no endpoint ip aging
```

次のタスク

エンドポイントの IP アドレスをトラッキングするために使用される間隔を指定するには、エンドポイント保持ポリシーを作成します。

NX-OS スタイル CLI を使用したブリッジ ドメイン上のスタティック ルートの設定

NX-OS スタイル CLI を使用したブリッジ ドメイン上のスタティック ルートの構成

パーベイスブブリッジドメイン (BD) でスタティック ルートを構成するには、NX-OS スタイルの次の CLI コマンドを使用します：

始める前に

テナント、VRF、BD および EPG が構成されています。

- スタティック ルートのサブネットを作成するには、epg (fvAEPg で fvSubnet オブジェクト)、普及 BD (fvBD) 自体 BD しないに関連付けられているように構成されます。

- サブネットマスクが/32 にする必要があります (128/for IPv6) 1 つの IP アドレスまたは 1 つのエンドポイントをポイントします。これは、EPG に関連付けられている普及 BD で含まれています。

手順の概要

1. **configure**
2. **tenant** *tenant-name*
3. **application** *ap-name*
4. **epg** *epg-name*
5. **endpoint** *ipA.B.C.D/LEN next-hop A.B.C.D [scope scope]*

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	構成モードに入ります。
ステップ 2	tenant <i>tenant-name</i> 例 : apic1(config)# tenant t1	テナントを作成するか、テナント構成モードに入ります。
ステップ 3	application <i>ap-name</i> 例 : apic1(config-tenant)# application ap1	アプリケーションプロファイルを作成するか、アプリケーションプロファイル モードに入ります。
ステップ 4	epg <i>epg-name</i> 例 : apic1(config-tenant-app)# epg ep1 <> <A.B.C.D> [scope <scope>]	EPG を作成するか、EPG 構成モードに入ります。
ステップ 5	endpoint <i>ipA.B.C.D/LEN next-hop A.B.C.D [scope scope]</i> 例 : apic1(config-tenant-app-epg)# endpoint ip 125.12.1.1/32 next-hop 26.0.14.101	EPG の背後にエンドポイントを作成します。サブネットマスクは /32 で (IPv6 の場合は /128)、1 つの IP アドレスまたは 1 つのエンドポイントをポイントする必要があります。

例

次の例は、EPG の背後にあるエンドポイントを構成するコマンドを示しています。

```
apic1# config
apic1(config)# tenant t1
apic1(config-tenant)# application ap1
apic1(config-tenant-app)# epg ep1
apic1(config-tenant-app-epg)# endpoint ip 125.12.1.1/32 next-hop 26.0.14.101
```

NX-OS Style CLI を使用した VRF ごとのデータプレーン IP ラーニングの設定

NX-OS-Style CLI を使用したデータプレーン IP ラーニングの設定

このセクションでは、NX-OS-Style CLI を使用してデータプレーン IP ラーニングを無効にする方法について説明します。

特定の VRF のデータプレーン IP ラーニングを無効にするには：

手順

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：

```
apic1# config
```

ステップ 2 特定のテナントのテナント モードに入ります。

例：

```
apic1(config)# tenant name
```

ステップ 3 VRF のコンテキスト モードに入ります。

例：

```
apic1(config-tenant)# vrf context name
```

ステップ 4 VRF のデータプレーン IP ラーニングを無効にします。

例：

```
apic1(config-tenant-vrf)# ipdataplanelearning disabled
```

NX-OS Style CLI を使用した IPv6 ネイバー探索の設定

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
```

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	configure 例 : <pre>apic1# 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	tenant tenant_name 例 : <pre>apic1(config)# tenant ExampleCorp apic1(config-tenant)#</pre>	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	template ipv6 nd policy policy_name 例 : <pre>apic1(config-tenant)# template ipv6 nd policy NDPo1001</pre>	IPv6 ND ポリシーを作成します。
ステップ 4	ipv6 nd mtu mtu value 例 : <pre>apic1(config-tenant-template-ipv6-nd)# ipv6 nd mtu 1500 apic1(config-tenant-template-ipv6)# exit apic1(config-tenant-template)# exit apic1(config-tenant)#</pre>	IPv6 ND ポリシーに MTU 値を割り当てます。
ステップ 5	vrf context VRF_name 例 : <pre>apic1(config-tenant)# vrf context pvn1 apic1(config-tenant-vrf)# exit</pre>	VRF を作成します。
ステップ 6	l3out VRF_name 例 :	レイヤ 3 アウトを作成します。

	コマンドまたはアクション	目的
	<code>apicl(config-tenant)# l3out l3extOut001</code>	
ステップ 7	vrf member <i>VRF_name</i> 例 : <pre> apicl(config-tenant-l3out)# vrf member pvn1 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	external-l3 epg instp l3out l3extOut001 例 : <pre> apicl(config-tenant)# external-l3 epg instp l3out l3extOut001 apicl(config-tenant-l3ext-epg)# vrf member pvn1 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	leaf 2011 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	vrf context tenant <i>ExampleCorp</i> vrf pvn1 l3out l3extOut001 例 : <pre> apicl(config-leaf)# vrf context tenant ExampleCorp vrf pvn1 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	int eth 1/1 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。
ステップ 12	vrf member tenant <i>ExampleCorp</i> vrf pvn1 l3out l3extOut001 例 : <pre> apicl(config-leaf-if)# vrf member tenant ExampleCorp vrf pvn1 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。

	コマンドまたはアクション	目的
ステップ 13	ipv6 address 2001:20:21:22::2/64 preferred 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	ipv6 nd prefix 2001:20:21:22::2/64 1000 1000 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	inherit ipv6 nd NDPol001 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。

NX-OS Style CLI を使用した Microsoft NLB の設定

NX-OS Style CLI を使用したユニキャスト モードでの Microsoft NLB の構成

このタスクは、ブリッジドメインのすべてのポートに Microsoft NLB がフラッドするように構成します。

始める前に

これらの手順を進める前に次の使用可能な情報を準備してください。

- Microsoft NLB クラスタ VIP
- Microsoft NLB クラスタ MAC アドレス

手順の概要

1. **configure**
2. **tenant tenant-name**
3. **application app-profile-name**
4. **epg epg-name**
5. **[no] endpoint {ip | ipv6} ip-address epnlb mode mode-uc mac mac-address**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	構成モードに入ります。
ステップ 2	tenant <i>tenant-name</i> 例 : <code>apic1 (config)# tenant tenant1</code>	存在しない場合はテナントを作成します。または、テナント構成モードを開始します。
ステップ 3	application <i>app-profile-name</i> 例 : <code>apic1 (config-tenant)# application appl1</code>	存在しない場合はアプリケーションプロファイルを作成します。または、アプリケーションプロファイル構成モードを開始します。
ステップ 4	epg <i>epg-name</i> 例 : <code>apic1 (config-tenant-app)# epg epg1</code>	存在しない場合はEPGを作成します。または、EPG構成モードを開始します。
ステップ 5	[no] endpoint {ip ipv6} ip-address egnlb mode mode-uc mac mac-address 例 : <code>apic1 (config-tenant-app-epg)# endpoint ip 192.0.2.2/32 egnlb mode mode-uc mac 03:BF:01:02:03:04</code>	Microsoft NLB をユニキャストモードで構成します : <ul style="list-style-type: none">• <i>ip-address</i> は Microsoft NLB クラスタ VIP です。• <i>mac-address</i> Microsoft NLB クラスタ MAC アドレスです。

NX-OS Style CLI を使用したマルチキャスト モードでの Microsoft NLB の構成

このタスクは、ブリッジドメインの特定のポートでのみ Microsoft NLB がフラッドするように構成します。

始める前に

これらの手順を進める前に次の使用可能な情報を準備してください。

- Microsoft NLB クラスタ VIP
- Microsoft NLB クラスタ MAC アドレス

手順の概要

1. **configure**
2. **tenant** *tenant-name*
3. **application** *app-profile-name*

4. **epg** *epg-name*
5. **[no] endpoint {ip | ipv6} ip-address eplnb mode mode-mcast--static mac mac-address**
6. **[no] nlb static-group mac-address leaf leaf-num interface {ethernet slot/port | port-channel port-channel-name} vlan portEncapVlan**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	構成モードに入ります。
ステップ 2	tenant <i>tenant-name</i> 例 : apicl (config)# tenant <i>tenant1</i>	存在しない場合はテナントを作成します。または、テナント構成モードを開始します。
ステップ 3	application <i>app-profile-name</i> 例 : apicl (config-tenant)# application <i>appl</i>	存在しない場合はアプリケーションプロファイルを作成します。または、アプリケーションプロファイル構成モードを開始します。
ステップ 4	epg <i>epg-name</i> 例 : apicl (config-tenant-app)# epg <i>epg1</i>	EPG 構成モードを開始します。まだ存在しない場合は EPG を作成します。
ステップ 5	[no] endpoint {ip ipv6} ip-address eplnb mode mode-mcast--static mac mac-address 例 : apicl (config-tenant-app-epg)# endpoint <i>ip</i> <i>192.0.2.2/32</i> eplnb mode mode-mcast--static mac <i>03:BF:01:02:03:04</i>	静的マルチキャスト モードで Microsoft NLB を構成します : <ul style="list-style-type: none"> • <i>ip-address</i> は Microsoft NLB クラスタ VIP です。 • <i>mac-address</i> Microsoft NLB クラスタ MAC アドレスです。
ステップ 6	[no] nlb static-group mac-address leaf leaf-num interface {ethernet slot/port port-channel port-channel-name} vlan portEncapVlan 例 : apicl (config-tenant-app-epg)# nlb static-group <i>03:BF:01:02:03:04</i> leaf <i>102</i> interface <i>ethernet 1/12</i> vlan <i>19</i>	Microsoft NLB マルチキャスト VMAC を、Microsoft NLB サーバが接続されている EPG ポートに追加します。 <ul style="list-style-type: none"> • <i>mac-address</i> は、入力した Microsoft NLB クラスタの MAC アドレスです。 ステップ 5 (10 ページ)。 • <i>leaf-num</i> は、追加または削除するインターフェイスを含むリーフ スイッチです。 • <i>port-channel-name</i> は、port-channel オプションを使用する場合のポートチャネルの名前です。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • <i>portEncapVlan</i> は、アプリケーション EPG の静的メンバーのカプセル化 VLAN です。

NX-OS Style CLI を使用した IGMP モードでの Microsoft NLB の構成

このタスクは、ブリッジドメインの特定のポートでのみ Microsoft NLB がフラッドするように構成します。

始める前に

これらの手順を進める前に次の使用可能な情報を準備してください。

- Microsoft NLB クラスタ VIP
- Microsoft NLB クラスタ MAC アドレス

手順の概要

1. **configure**
2. **tenant** *tenant-name*
3. **application** *app-profile-name*
4. **epg** *epg-name*
5. **[no] endpoint {ip | ipv6} ip-address eplb mode mode-mcast-igmp group multicast-IP-address**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apic1# configure	構成モードに入ります。
ステップ 2	tenant <i>tenant-name</i> 例 : apic1 (config)# tenant <i>tenant1</i>	存在しない場合はテナントを作成します。または、テナント構成モードを開始します。
ステップ 3	application <i>app-profile-name</i> 例 : apic1 (config-tenant)# application <i>app1</i>	存在しない場合はアプリケーションプロファイルを作成します。または、アプリケーションプロファイル構成モードを開始します。
ステップ 4	epg <i>epg-name</i> 例 :	存在しない場合はEPGを作成します。または、EPG構成モードを開始します。

	コマンドまたはアクション	目的
	<code>apic1 (config-tenant-app) # epg epg1</code>	
ステップ 5	<p>[no] endpoint {ip ipv6} ip-address eplb mode mode-mcast-igmp group multicast-IP-address</p> <p>例 :</p> <pre>apic1 (config-tenant-app-epg) # endpoint ip 192.0.2.2/32 eplb mode mode-mcast-igmp group 1.3.5.7</pre>	<p>Microsoft NLB を IGMP モードで構成します :</p> <ul style="list-style-type: none"> • <i>ip-address</i> は Microsoft NLB クラスター VIP です。 • <i>multicast-IP-address</i> は、NLB エンドポイントグループのマルチキャスト IP です。

NX-OS Style CLI を使用した IGMP スヌーピングの設定

NX-OS スタイル CLI を使用した IGMP スヌーピング ポリシーの設定とブリッジドメインへの割り当て

始める前に

- IGMP スヌーピングのポリシーを消費するテナントを作成します。
- IGMP スヌーピング ポリシーを接続するテナントのブリッジドメインを作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	<p>デフォルト値に基づいてスヌーピングポリシーを作成します。</p> <p>例 :</p> <pre>apic1(config-tenant)# template ip igmp snooping policy cookieCut1 apic1(config-tenant-template-ip-igmp-snooping)# show run all</pre> <pre># Command: show running -config all tenant foo template ip igmp snooping policy cookieCut1 # Time: Thu Oct 13 18:26:03 2016 tenant t_10 template ip igmp snooping policy cookieCut1 ip igmp snooping no ip igmp snooping fast-leave ip igmp snooping last-member-query-interval 1 no ip igmp snooping querier ip igmp snooping query-interval 125 ip igmp snooping query-max-response-time 10 ip igmp snooping stqrup-query-count 2 ip igmp snooping startup-query-interval 31 no description</pre>	<p>例の NX-OS スタイル CLI シーケンス :</p> <ul style="list-style-type: none"> • デフォルト値を持つ cookieCut1 という名前の IGMP スヌーピング ポリシーを作成します。 • ポリシー cookieCut1 のデフォルト IGMP スヌーピングの値が表示されます。

	コマンドまたはアクション	目的
	<pre> exit exit apic1(config-tenant-template-ip-igmp-snooping)# </pre>	
ステップ 2	<p>必要に応じてスヌーピングポリシーを変更します。</p> <p>例 :</p> <pre> apic1(config-tenant-template-ip-igmp-snooping)# ip igmp snooping query-interval 300 apic1(config-tenant-template-ip-igmp-snooping)# show run all # Command: show running -config all tenant foo template ip igmp snooping policy cookieCut1 #Time: Thu Oct 13 18:26:03 2016 tenant foo template ip igmp snooping policy cookieCut1 ip igmp snooping no ip igmp snooping fast-leave ip igmp snooping last-member-query-interval 1 no ip igmp snooping querier ip igmp snooping query-interval 300 ip igmp snooping query-max-response-time 10 ip igmp snooping stqrtpup-query-count 2 ip igmp snooping startup-query-interval 31 no description exit exit apic1(config-tenant-template-ip-igmp-snooping)# exit apic1(config--tenant)# </pre>	<p>例の NX-OS スタイル CLI シーケンス :</p> <ul style="list-style-type: none"> • cookieCut1 という名前の IGMP スヌーピングポリシーのクエリ間隔値のカスタム値を指定します。 • ポリシー cookieCut1 の変更された IGMP スヌーピング値を確認します。
ステップ 3	<p>必要に応じてスヌーピングポリシーを変更します。</p> <p>例 :</p> <pre> apic1(config-tenant-template-ip-igmp-snooping)# ip igmp snooping ? <CR> fast-leave Enable IP IGMP Snooping fast leave processing last-member-query-interval Change the IP IGMP snooping last member query interval param querier Enable IP IGMP Snooping querier processing query-interval Change the IP IGMP snooping query interval param query-max-response-time Change the IP IGMP snooping max query response time startup-query-count Change the IP IGMP snooping number of initial queries to send startup-query-interval Change the IP IGMP snooping time for sending initial queries version Change the IP IGMP snooping version param </pre>	<p>例の NX-OS スタイル CLI シーケンス :</p> <ul style="list-style-type: none"> • IGMP スヌーピングポリシーのクエリバージョンのカスタム値を指定します。 • ポリシーの変更された IGMP スヌーピングバージョンを確認します。

	コマンドまたはアクション	目的
	<pre> apic1(config-tenant-template-ip-igmp-snooping)# ip igmp snooping version ? v2 version-2 v3 version-3 apic1(config-tenant)# show run # Command: show running-config tenant tenant1 # Time: Mon Jun 1 01:53:53 2020 tenant tenant1 <snipped> interface bridge-domain amit_bd ip address 10.175.31.30/24 secondary ip address 100.175.31.1/32 secondary snooping-querier ip igmp snooping policy igmp_snoop_policy exit template ip igmp snooping policy igmp_snoop_policy ip igmp snooping fast-leave ip igmp snooping last-member-query-interval 2 ip igmp snooping querier v3 ip igmp snooping query-interval 100 ip igmp snooping startup-query-count 5 ip igmp snooping version v3 exit exit </pre>	
ステップ 4	<p>ブリッジ ドメインにポリシーを割り当てます。</p> <p>例 :</p> <pre> apic1(config-tenant)# int bridge-domain bd3 apic1(config-tenant-interface)# ip igmp snooping policy cookieCut1 </pre>	<p>例の NX-OS スタイル CLI シーケンス :</p> <ul style="list-style-type: none"> ブリッジドメインの BD3 に移動します。IGMP スヌーピング ポリシーのクエリ間隔値は cookieCut1 という名前です。 ポリシー cookieCut1 の変更された IGMP スヌーピングの値を持つ IGMP スヌーピングのポリシーを割り当てます。

次のタスク

複数のブリッジ ドメインに IGMP スヌーピングのポリシーを割り当てることができます。

NX-OS スタイル CLI によりスタティック ポートで IGMP スヌーピングおよびマルチキャストの有効化

EPG に静的に割り当てられたポートで IGMP スヌーピングおよびマルチキャストをイネーブルにできます。それらのポートで有効な IGMP スヌーピングおよびマルチキャストトラフィックへのアクセスを許可または拒否するアクセスユーザーのグループを作成および割り当てることができます。

このタスクで説明されている手順には、次のエンティティの事前設定を前提とします。

- テナント : tenant_A

- アプリケーション : application_A
- EPG : epq_A
- ブリッジ ドメイン : bridge_domain_A
- vrf : vrf_A -- a member of bridge_domain_A
- VLAN ドメイン : vd_A (300 ~ 310 の範囲で設定される)
- リーフ スイッチ : 101 およびインターフェイス 1/10

スイッチ 101 のターゲット インターフェイス 1/10 が VLAN 305 に関連付けられており、enant_A、application_A、epq_A に静的にリンクされています。

- リーフ スイッチ : 101 およびインターフェイス 1/11

スイッチ 101 のターゲット インターフェイス 1/11 が VLAN 309 に関連付けられており、enant_A、application_A、epq_A に静的にリンクされています。

始める前に

EPG に IGMP スヌーピングおよびマルチキャストを有効にする前に、次のタスクを実行します。

- この機能を有効にして静的に EPG に割り当てるインターフェイスを特定する



(注) 静的割り当てポートの詳細については、*[NX-OS スタイルの CLI を使用した APIC の特定のポートへの EPG の導入 (Deploying an EPG on a Specific Port with APIC Using the NX-OS Style CLI)]* を参照します。これは、*[Cisco APIC レイヤ 2 ネットワーク構成ガイド (Cisco APIC Layer 2 Networking Configuration Guide)]*にあります。

- IGMP スヌーピング マルチキャスト トラフィックの受信者の IP アドレスを特定します。

手順

	コマンドまたはアクション	目的
ステップ 1	<p>ターゲット インターフェイスで IGMP スヌーピング およびレイヤ 2 マルチキャストリングを有効にします</p> <p>例 :</p> <pre>apic1# conf t apic1(config)# tenant tenant_A apic1(config-tenant)# application application_A apic1(config-tenant-app)# epq epq_A</pre>	<p>例のシーケンスでは次を有効にします。</p> <ul style="list-style-type: none"> • 静的にリンクされているターゲット インターフェイス 1/10 の IGMP スヌーピング、そしてマルチキャスト IP アドレス、225.1.1.1 に関連付けます • 静的にリンクされているターゲット インターフェイス 1/11 の IGMP スヌーピング、そしてマ

コマンドまたはアクション	目的
<pre> apic1(config-tenant-app-epg)# ip igmp snooping static-group 225.1.1.1 leaf 101 interface ethernet 1/10 vlan 305 apic1(config-tenant-app-epg)# end apic1# conf t apic1(config)# tenant tenant_A; application application_A; epg epg_A apic1(config-tenant-app-epg)# ip igmp snooping static-group 227.1.1.1 leaf 101 interface ethernet 1/11 vlan 309 apic1(config-tenant-app-epg)# exit apic1(config-tenant-app)# exit </pre>	ルチキャスト IP アドレス、227.1.1.1 に関連付けます

NX-OS スタイル CLI を使用した IGMP スヌーピングおよびマルチキャスト グループへのアクセスの有効化

EPG に静的に割り当てられたポートで IGMP スヌーピングおよびマルチキャストを有効にした後、それらのポートで有効な IGMP スヌーピングおよびマルチキャストトラフィックへのアクセスを許可または拒否するユーザーのアクセス グループを作成および割り当てできます。

このタスクで説明されている手順には、次のエンティティの事前設定を前提とします。

- テナント : tenant_A
- アプリケーション : application_A
- EPG : epg_A
- ブリッジ ドメイン : bridge_domain_A
- vrf : vrf_A -- a member of bridge_domain_A
- VLAN ドメイン : vd_A (300 ~ 310 の範囲で設定される)
- リーフ スイッチ : 101 およびインターフェイス 1/10
- リーフ スイッチ : 101 およびインターフェイス 1/11

スイッチ 101 のターゲット インターフェイス 1/10 が VLAN 305 に関連付けられており、tenant_A、application_A、epg_A に静的にリンクされています。

スイッチ 101 のターゲット インターフェイス 1/11 が VLAN 309 に関連付けられており、tenant_A、application_A、epg_A に静的にリンクされています。



(注) 静的割り当てポートの詳細については、*[NX-OS スタイルの CLI を使用した APIC の特定のポートへの EPG の導入 (Deploying an EPG on a Specific Port with APIC Using the NX-OS Style CLI)]* を参照します。これは、*[Cisco APIC レイヤ 2 ネットワーク構成ガイド (Cisco APIC Layer 2 Networking Configuration Guide)]*にあります。

手順

	コマンドまたはアクション	目的
ステップ 1	<p>route-map 「アクセス グループ」 を定義します。</p> <p>例 :</p> <pre>apic1# conf t apic1(config)# tenant tenant_A; application application_A; epg epg_A apic1(config-tenant)# route-map fooBroker permit apic1(config-tenant-rtmap)# match ip multicast group 225.1.1.1/24 apic1(config-tenant-rtmap)# exit apic1(config-tenant)# route-map fooBroker deny apic1(config-tenant-rtmap)# match ip multicast group 227.1.1.1/24 apic1(config-tenant-rtmap)# exit</pre>	<p>例のシーケンスを構成します :</p> <ul style="list-style-type: none"> マルチキャスト グループ 225.1.1.1/24 にリンクされる Route-map-access グループ 「foobroker」 のアクセスが許可されています。 マルチキャスト グループ 225.1.1.1/24 にリンクされる Route-map-access グループ 「foobroker」 のアクセスが拒否されています。
ステップ 2	<p>ルート マップ設定を確認します。</p> <p>例 :</p> <pre>apic1(config-tenant)# show running-config tenant test route-map fooBroker # Command: show running-config tenant test route-map fooBroker # Time: Mon Aug 29 14:34:30 2016 tenant test route-map fooBroker permit 10 match ip multicast group 225.1.1.1/24 exit route-map fooBroker deny 20 match ip multicast group 227.1.1.1/24 exit exit</pre>	
ステップ 3	<p>アクセス グループ接続パスを指定します。</p> <p>例 :</p> <pre>apic1(config-tenant)# application application_A apic1(config-tenant-app)# epg epg_A apic1(config-tenant-app-epg)# ip igmp snooping access-group route-map fooBroker leaf 101 interface ethernet 1/10 vlan 305 apic1(config-tenant-app-epg)# ip igmp snooping access-group route-map newBroker leaf 101 interface ethernet 1/10 vlan 305</pre>	<p>例のシーケンスを設定します。</p> <ul style="list-style-type: none"> リーフ スイッチ 101、インターフェイス 1/10、VLAN 305 で接続されている Route-map-access グループ 「foobroker」 。 リーフ スイッチ 101、インターフェイス 1/10、VLAN 305 で接続されている Route-map-access グループ 「newbroker」 。
ステップ 4	<p>アクセスグループ接続を確認します。</p> <p>例 :</p> <pre>apic1(config-tenant-app-epg)# show run # Command: show running-config tenant tenant_A application application_A epg epg_A # Time: Mon Aug 29 14:43:02 2016 tenant tenant_A application application_A epg epg_A</pre>	

	コマンドまたはアクション	目的
	<pre> bridge-domain member bridge_domain_A ip igmp snooping access-group route-map fooBroker leaf 101 interface ethernet 1/10 vlan 305 ip igmp snooping access-group route-map fooBroker leaf 101 interface ethernet 1/11 vlan 309 ip igmp snooping access-group route-map newBroker leaf 101 interface ethernet 1/10 vlan 305 ip igmp snooping static-group 225.1.1.1 leaf 101 interface ethernet 1/10 vlan 305 ip igmp snooping static-group 225.1.1.1 leaf 101 interface ethernet 1/11 vlan 309 exit exit exit </pre>	

NX-OS Style CLI を使用した MLD スヌーピングの設定

NX-OS Style CLI を使用したブリッジ ドメインに対する MLD スヌーピング ポリシーの構成と割り当て

始める前に

- MLD スヌーピングのポリシーを消費するテナントを作成します。
- MLD スヌーピング ポリシーを接続するテナントのブリッジ ドメインを作成します。

手順の概要

1. **configure terminal**
2. **tenant *tenant-name***
3. **template ipv6 mld snooping policy *policy-name***
4. **[no] ipv6 mld snooping**
5. **[no] ipv6 mld snooping fast-leave**
6. **[no] ipv6 mld snooping querier**
7. **ipv6 mld snooping last-member-query-interval *parameter***
8. **ipv6 mld snooping query-interval *parameter***
9. **ipv6 mld snooping query-max-response-time *parameter***
10. **ipv6 mld snooping startup-query-count *parameter***
11. **ipv6 mld snooping startup-query-interval *parameter***
12. **exit**
13. **interface bridge-domain *bridge-domain-name***
14. **ipv6 address *sub-bits/prefix-length* snooping-querier**
15. **ipv6 mld snooping policy *policy-name***
16. **exit**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : <pre>apicl# configure terminal apicl(config)#</pre>	構成モードに入ります。
ステップ 2	tenant tenant-name 例 : <pre>apicl(config)# tenant tn1 apicl(config-tenant)#</pre>	テナントを作成するか、テナント構成モードに入ります。
ステップ 3	template ipv6 mld snooping policy policy-name 例 : <pre>apicl(config-tenant)# template ipv6 mld snooping policy mldPolicy1 apicl(config-tenant-template-ip-mld-snooping)#</pre>	MLD スヌーピング ポリシーを作成します。例の NX-OS スタイルの CLI シーケンスは、mldPolicy1 という名前の MLD スヌーピング ポリシーを作成します。
ステップ 4	[no] ipv6 mld snooping 例 : <pre>apicl(config-tenant-template-ip-mld-snooping)# ipv6 mld snooping apicl(config-tenant-template-ip-mld-snooping)# no ipv6 mld snooping</pre>	MLD スヌープ ポリシーの管理状態を有効または無効にします。デフォルトのステートはディセーブルです。
ステップ 5	[no] ipv6 mld snooping fast-leave 例 : <pre>apicl(config-tenant-template-ip-mld-snooping)# ipv6 mld snooping fast-leave apicl(config-tenant-template-ip-mld-snooping)# no ipv6 mld snooping fast-leave</pre>	IPv6 MLD スヌーピング ファストリーブ処理を有効または無効にします。
ステップ 6	[no] ipv6 mld snooping querier 例 : <pre>apicl(config-tenant-template-ip-mld-snooping)# ipv6 mld snooping querier apicl(config-tenant-template-ip-mld-snooping)# no ipv6 mld snooping querier</pre>	IPv6 MLD スヌーピング クエリア処理を有効または無効にします。有効にするクエリア オプションを割り当て済みのポリシーで効果的に有効にするには、 ステップ 14 (21 ページ) で説明されているように、ポリシーを適用するブリッジ ドメインに割り当てられるサブネットでもクエリア オプションを有効にする必要があります。

	コマンドまたはアクション	目的
ステップ 7	ipv6 mld snooping last-member-query-interval <i>parameter</i> 例 : <pre>apic1(config-tenant-template-ip-mld-snooping)# ipv6 mld snooping last-member-query-interval 25</pre>	IPv6 MLD スヌーピングの最終メンバー クエリー間隔パラメータを変更します。NX-OS スタイルの CLI シーケンスの例では、IPv6 MLD スヌーピングの最後のメンバーのクエリー間隔パラメータが 25 秒に変更されます。有効なオプションは 1 ～ 25 です。デフォルト値は 1 秒です。
ステップ 8	ipv6 mld snooping query-interval <i>parameter</i> 例 : <pre>apic1(config-tenant-template-ip-mld-snooping)# ipv6 mld snooping query-interval 300</pre>	IPv6 MLD スヌーピングクエリー間隔パラメータを変更します。NX-OS スタイルの CLI シーケンス例では、IPv6 MLD スヌーピングクエリー間隔パラメータを 300 秒に変更します。有効なオプションは 1 ～ 18000 です。デフォルト値は 125 秒です。
ステップ 9	ipv6 mld snooping query-max-response-time <i>parameter</i> 例 : <pre>apic1(config-tenant-template-ip-mld-snooping)# ipv6 mld snooping query-max-response-time 25</pre>	IPv6 MLD スヌーピングの最大クエリー応答時間を変更します。NX-OS スタイルの CLI シーケンスの例では、IPv6 MLD スヌーピングの最大クエリー応答時間が 25 秒に変更されます。有効なオプションは 1 ～ 25 です。デフォルトは 10 秒です。
ステップ 10	ipv6 mld snooping startup-query-count <i>parameter</i> 例 : <pre>apic1(config-tenant-template-ip-mld-snooping)# ipv6 mld snooping startup-query-count 10</pre>	送信する初期クエリーの IPv6 MLD スヌーピング数を変更します。NX-OS スタイルの CLI シーケンスの例では、最初のクエリーの IPv6 MLD スヌーピング数を 10 に変更します。有効なオプションは 1 ～ 10 です。デフォルトは 2 です。
ステップ 11	ipv6 mld snooping startup-query-interval <i>parameter</i> 例 : <pre>apic1(config-tenant-template-ip-mld-snooping)# ipv6 mld snooping startup-query-interval 300</pre>	初期クエリーを送信するための IPv6 MLD スヌーピング時間を変更します。NX-OS スタイルの CLI シーケンスの例では、最初のクエリーを送信するための IPv6 MLD スヌーピング時間が 300 秒に変更されます。有効なオプションは 1 ～ 18000 です。デフォルト値は 31 秒です。
ステップ 12	exit 例 : <pre>apic1(config-tenant-template-ip-mld-snooping)# exit apic1(config-tenant)#</pre>	構成モードに戻ります。
ステップ 13	interface bridge-domain <i>bridge-domain-name</i> 例 : <pre>apic1(config-tenant)# interface bridge-domain bd1 apic1(config-tenant-interface)#</pre>	インターフェイスブリッジドメインを構成します。例の NX-OS スタイルの CLI シーケンスは、bd1 という名前のインターフェイスブリッジドメインを構成します。

	コマンドまたはアクション	目的
ステップ 14	ipv6 address sub-bits/prefix-length snooping-querier 例 : <pre>apicl(config-tenant-interface)# ipv6 address 2000::5/64 snooping-querier</pre>	ブリッジドメインをスイッチ クエリアとして構成します。これにより、ポリシーが適用されるブリッジドメインに割り当てられたサブネットでクエリア オプションが有効になります。
ステップ 15	ipv6 mld snooping policy policy-name 例 : <pre>apicl(config-tenant-interface)# ipv6 mld snooping policy mldPolicy1</pre>	ブリッジドメインを MLD スヌーピング ポリシーに関連付けます。例の NX-OS スタイルの CLI シーケンスは、mldPolicy1 という名前の MLD スヌーピングポリシーにブリッジドメインに関連付けます。
ステップ 16	exit 例 : <pre>apicl(config-tenant-interface)# exit apicl(config-tenant)#</pre>	構成モードに戻ります。

NX-OS Style CLI を使用した IP マルチキャストの設定

NX-OS スタイルの CLI を使用したレイヤ 3 マルチキャストの設定

手順

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 :

```
apicl# configure
```

ステップ 2 テナントの設定モード、VRF の設定モードは、および PIM オプションの設定モードに入ります。

例 :

```
apicl(config)# tenant tenant1
apicl(config-tenant)# vrf context tenant1_vrf
apicl(config-tenant-vrf)# ip pim
apicl(config-tenant-vrf)# ip pim fast-convergence
apicl(config-tenant-vrf)# ip pim bsr forward
```

ステップ 3 IGMP を設定し、VRF に適切な IGMP オプションを設定します。

例 :

```
apicl(config-tenant-vrf)# ip igmp
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# interface bridge-domain tenant1_bd
apicl(config-tenant-interface)# ip multicast
apicl(config-tenant-interface)# ip igmp allow-v3-asm
```

```

apic1(config-tenant-interface)# ip igmp fast-leave
apic1(config-tenant-interface)# ip igmp inherit interface-policy igmp_intpoll
apic1(config-tenant-interface)# exit

```

ステップ 4 テナントの L3 Out モードに入り、PIM を有効にし、リーフ インターフェイス モードに入ります。このインターフェイスの PIM を設定します。

例 :

```

apic1(config-tenant)# l3out tenant1_l3out
apic1(config-tenant-l3out)# ip pim
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# exit
apic1(config)#
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/125
apic1(config-leaf-if) ip pim inherit interface-policy pim_intpoll

```

ステップ 5 IGMP コマンドを使用して、インターフェイスの IGMP を設定します。

例 :

```

apic1(config-leaf-if)# ip igmp fast-leave
apic1(config-leaf-if)# ip igmp inherit interface-policy igmp_intpoll
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit

```

ステップ 6 ファブリック RP を設定します。

例 :

```

apic1(config)# tenant tenant1
apic1(config-tenant)# vrf context tenant1_vrf
apic1(config-tenant-vrf)# ip pim fabric-rp-address 20.1.15.1 route-map intervrf-ctx2
apic1(config-tenant-vrf)# ip pim fabric-rp-address 20.1.15.2 route-map intervrf-ctx1
apic1(config-tenant-vrf)# exit

```

ステップ 7 Inter-VRF マルチキャストを設定します。

例 :

```

apic1(config-tenant)# vrf context tenant1_vrf
apic1(config-tenant-vrf)# ip pim inter-vrf-src ctx2 route-map intervrf-ctx2
apic1(config-tenant-vrf)# route-map intervrf-ctx2 permit 1
apic1(config-tenant-vrf)# match ip multicast group 226.20.0.0/24
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)#

```

これにより、APIC のレイヤ 3 マルチキャストの設定を完了します。

NX-OS Style CLI を使用したレイヤ 3 IPv6 の設定

始める前に

- 目的の VRF、ブリッジドメイン、IPv6 アドレスを持つレイヤ 3 Out インターフェイスは、PIM6 が有効になるように設定する必要があります。レイヤ 3 Out の場合、IPv6 マルチキャストが機能するために、論理ノードプロファイルのノードに IPv6 ループバック アドレスが設定されます。
- 基本的なユニキャスト ネットワークを設定する必要があります。

手順

ステップ 1 VRF で PIM6 を有効にし、ランデブー ポイント (RP) を設定します。

例 :

```
apicl(config)# tenant tenant1
apicl(config-tenant)# vrf context tenant1_vrf
apicl(config-tenant-vrf)# ipv6 pim
apicl(config-tenant-vrf)# ipv6 rp-address 2018::100:100:100:100 route-map ipv6_pim_routemap
```

ステップ 2 PIM6 インターフェイス ポリシーを設定し、レイヤ 3 Out に適用します。

例 :

```
apicl(config-tenant)# l3out tenant1_l3out
apicl(config-tenant-l3out)# ipv6 pim
apicl(config-tenant-l3out)# exit
apicl(config-tenant)# exit
apicl(config)#
apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/125
apicl(config-leaf-if) ipv6 pim inherit interface-policy pim6_intpoll
```

ステップ 3 BD で PIM6 を有効にします。

例 :

```
apicl(config-tenant)# interface bridge-domain tenant1_bd
apicl(config-tenant-interface)# ipv6 multicast
apicl(config-tenant)# exit
apicl(config)#
```

PIM6 を使用したレイヤ 3 IPv6 マルチキャストが有効になります。

NX-OS スタイルの CLI を使用したマルチキャスト フィルタリングの構成

ブリッジ ドメイン レベルでマルチキャスト フィルタリングを構成します。このトピックの手順を使用して、ブリッジ ドメイン レベルで送信元フィルタリングまたは受信者フィルタリング、あるいはその両方を構成します。

始める前に

- マルチキャストフィルタリングを構成するブリッジドメインはすでに作成されています。
- ブリッジドメインは PIM 対応ブリッジドメインです。
- レイヤ 3 マルチキャストは VRF レベルで有効になります。

手順

ステップ 1 構成モードを開始します。

```
apic1# configure  
apic1(config)#
```

ステップ 2 テナントにアクセスし、PIM を有効にします。

```
apic1(config)# tenant tenant-name  
apic1(config-tenant)# vrf context VRF-name  
apic1(config-tenant-vrf)# ip pim  
apic1(config-tenant-vrf)# exit  
apic1(config-tenant)#
```

例 :

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

ステップ 3 マルチキャスト フィルタリングを構成するブリッジ ドメインにアクセスします。

```
apic1(config-tenant)# bridge-domain BD-name  
apic1(config-tenant-bd)#
```

例 :

```
apic1(config-tenant)# bridge-domain bd1  
apic1(config-tenant-bd)#
```

ステップ4 このブリッジ ドメインにマルチキャスト 送信元 または 受信者 のフィルタリングを有効にするかどうかを決定します。

(注)

送信元フィルタリングと受信先フィルタリングの両方を同じブリッジ ドメインで有効にできます。

- このブリッジ ドメインでマルチキャスト送信元フィルタリングを有効にする場合は、次の例のように入力します。

```
apicl(config-tenant-bd) # src-filter source-route-map-policy
```

次に例を示します。

```
apicl(config-tenant-bd) # src-filter routemap-Mcast-src
```

- このブリッジ ドメインでマルチキャスト送信元フィルタリングを有効にする場合は、次の例のように入力します。

```
apicl(config-tenant-bd) # dst-filter destination-route-map-policy
```

次に例を示します。

```
apicl(config-tenant-bd) # dst-filter routemap-Mcast-dst
```

ステップ5 IPv4 のマルチキャストを有効にします。

```
apicl(config-tenant-bd) # mcast-allow  
apicl(config-tenant-bd) #
```

ステップ6 VRF にブリッジ ドメインを関連付けます。

```
apicl(config-tenant-bd) # vrf member VRF-name  
apicl(config-tenant-bd) # exit  
apicl(config-tenant) #
```

例：

```
apicl(config-tenant-bd) # vrf member v1  
apicl(config-tenant-bd) # exit  
apicl(config-tenant) #
```

ステップ7 ブリッジ ドメインでマルチキャストを有効にします。

```
apicl(config-tenant) # interface bridge-domain BD-name  
apicl(config-tenant-interface) # ip multicast  
apicl(config-tenant-interface) # exit  
apicl(config-tenant) #
```

例：

```
apicl(config-tenant) # interface bridge-domain bd1
```

```
apic1(config-tenant-interface)# ip multicast
apic1(config-tenant-interface)# exit
apic1(config-tenant)#
```

ステップ 8 ルート マップを構成します。

```
apic1(config-tenant)# route-map destination-route-map-policy <permit/deny> sequence_number
apic1(config-tenant-rtmap)# match ip multicast <source/group> IP_address_subnet <source/group>
IP_address_subnet
apic1(config-tenant-rtmap)# exit
apic1(config-tenant)# exit
apic1(config)#
```

例 :

```
apic1(config-tenant)# route-map routemap-Mcast-src permit 1
apic1(config-tenant-rtmap)# match ip multicast source 10.10.1.1/24 group 192.1.1.1/32
apic1(config-tenant-rtmap)# exit
apic1(config-tenant)# route-map routemap-Mcast-dst permit 1
apic1(config-tenant-rtmap)# match ip multicast group 192.2.2.2/32
apic1(config-tenant-rtmap)# exit
apic1(config-tenant)# exit
apic1(config)#
```

NX-OS Style CLI を使用したマルチポッドの設定

NX-OS CLI を使用したマルチポッド ファブリックのセットアップ

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

ステップ 1 次の例に示すように、マルチポッドを設定します。

例 :

```
ifav4-ifc1# show run system
# Command: show running-config system
# Time: Mon Aug 1 21:32:03 2016
system cluster-size 3
system switch-id FOX2016G9DW 204 ifav4-spine4 pod 2
system switch-id SAL1748H56D 201 ifav4-spine1 pod 1
system switch-id SAL1803L25H 102 ifav4-leaf2 pod 1
system switch-id SAL1819RXP4 101 ifav4-leaf1 pod 1
system switch-id SAL1931LA3B 203 ifav4-spine2 pod 2
system switch-id SAL1934MNY0 103 ifav4-leaf3 pod 1
system switch-id SAL1934MNY3 104 ifav4-leaf4 pod 1
```

```

system switch-id SAL1938P7A6 202 ifav4-spine3 pod 1
system switch-id SAL1938PHBB 105 ifav4-leaf5 pod 2
system switch-id SAL1942R857 106 ifav4-leaf6 pod 2
system pod 1 tep-pool 10.0.0.0/16
system pod 2 tep-pool 10.1.0.0/16
ifav4-ifc1#

```

ステップ2 次の例のよ、VLAN ドメインを設定します。

例：

```

ifav4-ifc1# show running-config vlan-domain l3Dom
# Command: show running-config vlan-domain l3Dom
# Time: Mon Aug 1 21:32:31 2016
vlan-domain l3Dom
vlan 4
exit
ifav4-ifc1#

```

ステップ3 次の例のよ、ファブリックの外部接続を設定します。

例：

```

ifav4-ifc1# show running-config fabric-external
# Command: show running-config fabric-external
# Time: Mon Aug 1 21:34:17 2016
fabric-external 1
bgp evpn peering
pod 1
interpod data hardware-proxy 100.11.1.1/32
bgp evpn peering
exit
pod 2
interpod data hardware-proxy 200.11.1.1/32
bgp evpn peering
exit
route-map interpod-import
ip prefix-list default permit 0.0.0.0/0
exit
route-target extended 5:16
exit
ifav4-ifc1#

```

ステップ4 スパイン スイッチ インターフェイスと次の例のよの OSPF 設定を構成します。

例：

```

# Command: show running-config spine
# Time: Mon Aug 1 21:34:41 2016
spine 201
vrf context tenant infra vrf overlay-1
router-id 201.201.201.201
exit
interface ethernet 1/1
vlan-domain member l3Dom
exit
interface ethernet 1/1.4
vrf member tenant infra vrf overlay-1
ip address 201.1.1.1/30
ip router ospf default area 1.1.1.1
ip ospf cost 1
exit
interface ethernet 1/2
vlan-domain member l3Dom
exit

```

```
interface ethernet 1/2.4
  vrf member tenant infra vrf overlay-1
  ip address 201.2.1.1/30
  ip router ospf default area 1.1.1.1
  ip ospf cost 1
  exit
router ospf default
  vrf member tenant infra vrf overlay-1
  area 1.1.1.1 loopback 201.201.201.201
  area 1.1.1.1 interpod peering
  exit
exit
spine 202
  vrf context tenant infra vrf overlay-1
  router-id 202.202.202.202
  exit
interface ethernet 1/2
  vlan-domain member l3Dom
  exit
interface ethernet 1/2.4
  vrf member tenant infra vrf overlay-1
  ip address 202.1.1.1/30
  ip router ospf default area 1.1.1.1
  exit
router ospf default
  vrf member tenant infra vrf overlay-1
  area 1.1.1.1 loopback 202.202.202.202
  area 1.1.1.1 interpod peering
  exit
exit
exit
spine 203
  vrf context tenant infra vrf overlay-1
  router-id 203.203.203.203
  exit
interface ethernet 1/1
  vlan-domain member l3Dom
  exit
interface ethernet 1/1.4
  vrf member tenant infra vrf overlay-1
  ip address 203.1.1.1/30
  ip router ospf default area 0.0.0.0
  ip ospf cost 1
  exit
interface ethernet 1/2
  vlan-domain member l3Dom
  exit
interface ethernet 1/2.4
  vrf member tenant infra vrf overlay-1
  ip address 203.2.1.1/30
  ip router ospf default area 0.0.0.0
  ip ospf cost 1
  exit
router ospf default
  vrf member tenant infra vrf overlay-1
  area 0.0.0.0 loopback 203.203.203.203
  area 0.0.0.0 interpod peering
  exit
exit
exit
spine 204
  vrf context tenant infra vrf overlay-1
  router-id 204.204.204.204
```

```

exit
interface ethernet 1/31
vlan-domain member l3Dom
exit
interface ethernet 1/31.4
vrf member tenant infra vrf overlay-1
ip address 204.1.1.1/30
ip router ospf default area 0.0.0.0
ip ospf cost 1
exit
router ospf default
vrf member tenant infra vrf overlay-1
area 0.0.0.0 loopback 204.204.204.204
area 0.0.0.0 interpod peering
exit
exit
exit
ifav4-ifc1#

```

NX-OS Style CLI を使用したリモート リーフ スイッチの設定

NX-OS スタイル CLI を使用したリモート リーフ スイッチの設定

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 リーフ 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 スイッチ設定の注意事項](#)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:

```
apic1(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-name pod pod-id remote-leaf-site remote-leaf-site-id node-type remote-leaf-wan**

例：

```
apic1(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 ドメインを構成します。

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 を構成します。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構成します。

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
```

```
apicl(config-spine-if)# ip router ospf multipod-internal area 5
apicl(config-spine-if)# exit
apicl(config-spine)# exit
apicl(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構成します。

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構成を示しています :

```
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
```

```

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)#
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

```

パートII：外部ルーティング（L3Out）の設定

外部ネットワークへのルーテッド接続

NX-OS Style CLI を使用した MP-BGP ルート リフレクタの設定

ACI ファブリックの MP-BGP ルート リフレクタの設定

ACI ファブリック内のルートを配布するために、MP-BGP プロセスを最初に実行し、スパインスイッチを BGP ルート リフレクタとして設定する必要があります。

次に、MP-BGP ルート リフレクタの設定例を示します。



(注) この例では、BGP ファブリック ASN は 100 です。スパインスイッチ 104 と 105 が MP-BGP ルート リフレクタとして選択されます。

```

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

```

L3Out のノードとインターフェイス

NX-OS Style CLI を使用したレイヤ 3 ルーテッド ポート チャンネルとサブインターフェイス ポート チャンネルの設定

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 の構成

この手順では、レイヤ 3 ルーテッドポート チャンネルを構成します。

手順の概要

1. **configure**
2. **leaf node-id**
3. **interface port-channel channel-name**
4. **no switchport**
5. **vrf member vrf-name tenant tenant-name**
6. **vlan-domain member vlan-domain-name**
7. **ip address ip-address/subnet-mask**
8. **ipv6 address sub-bits/prefix-length preferred**
9. **ipv6 link-local ipv6-link-local-address**
10. **mac-address mac-address**
11. **mtu mtu-value**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	グローバル構成モードを開始します。
ステップ 2	leaf node-id 例 : apicl(config)# leaf 101	リーフ スイッチまたはリーフ スイッチの構成を指定します。 <i>node-id</i> は、1 つの ID または ID の範囲になります。 <i>node-id1-node-id2</i> 形式に次の構成が適用されます：
ステップ 3	interface port-channel channel-name 例 : apicl(config-leaf)# interface port-channel po1	指定したポート チャンネルのインターフェイス構成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 4	no switchport 例 : <pre>apic1(config-leaf-if)# no switchport</pre>	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	vrf member vrf-name tenant tenant-name 例 : <pre>apic1(config-leaf-if)# vrf member v1 tenant t1</pre>	<p>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。</p> <ul style="list-style-type: none"> • vrf-name 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。 • tenant-name 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。
ステップ 6	vlan-domain member vlan-domain-name 例 : <pre>apic1(config-leaf-if)# vlan-domain member dom1</pre>	以前に構成された 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	ip address ip-address/subnet-mask 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 8	ipv6 address sub-bits/prefix-length preferred 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 アドレスを構成し、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"> • sub-bits 引数は、prefix-name 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。 • prefix-length 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。
ステップ 9	ipv6 link-local ipv6-link-local-address 例 :	インターフェイスに IPv6 リンクローカルアドレスを構成します。

	コマンドまたはアクション	目的
	<code>apic1(config-leaf-if) # ipv6 link-local fe80::1</code>	
ステップ 10	mac-address mac-address 例 : <code>apic1(config-leaf-if) # mac-address 00:44:55:66:55::01</code>	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	mtu mtu-value 例 : <code>apic1(config-leaf-if) # mtu 1500</code>	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 チャンネルを設定する方法を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member v1 tenant t1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
apic1(config-leaf-if)# ipv6 link-local fe80::1
apic1(config-leaf-if)# mac-address 00:44:55:66:55::01
apic1(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 チャンネルの設定

この手順では、レイヤ 3 サブインターフェイス ポート チャンネルを設定します。

手順の概要

1. **configure**
2. **leaf node-id**
3. **vrf member vrf-name tenant tenant-name**
4. **vlan-domain member vlan-domain-name**
5. **ip address ip-address / subnet-mask**
6. **ipv6 address sub-bits / prefix-length preferred**
7. **ipv6 link-local ipv6-link-local-address**
8. **mac-address mac-address**
9. **mtu mtu-value**
10. **exit**
11. **interface port-channel channel-name**
12. **vlan-domain member vlan-domain-name**

13. `exit`
14. `interface port-channel channel-name.number`
15. `vrf member vrf-name tenant tenant-name`
16. `exit`

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	leaf node-id 例 : <code>apic1(config)# leaf 101</code>	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、構成が適用される <i>node-id1-node-id2</i> 形式で 1 つの ID または ID の範囲になります。
ステップ 3	vrf member vrf-name tenant tenant-name 例 : <code>apic1(config-leaf-if)# vrf member v1 tenant t1</code>	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"> • <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。 • <i>tenant-name</i> 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。
ステップ 4	vlan-domain member vlan-domain-name 例 : <code>apic1(config-leaf-if)# vlan-domain member dom1</code>	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	ip address ip-address / subnet-mask 例 : <code>apic1(config-leaf-if)# ip address 10.1.1.1/24</code>	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	ipv6 address sub-bits / prefix-length preferred 例 : <code>apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred</code>	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"> • <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

	コマンドまたはアクション	目的
		<p>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。</p> <ul style="list-style-type: none"> • <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。
ステップ 7	ipv6 link-local <i>ipv6-link-local-address</i> 例 : apicl(config-leaf-if) # ipv6 link-local fe80::1	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	mac-address <i>mac-address</i> 例 : apicl(config-leaf-if) # mac-address 00:44:55:66:55::01	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 9	mtu <i>mtu-value</i> 例 : apicl(config-leaf-if) # mtu 1500	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) # exit	設定モードに戻ります。
ステップ 11	interface port-channel <i>channel-name</i> 例 : apicl(config-leaf) # interface port-channel po1	指定したポート チャン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	vlan-domain member <i>vlan-domain-name</i> 例 : apicl(config-leaf-if) # vlan-domain member dom1	以前に設定された VLAN ドメインには、ポート チャン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) # exit	設定モードに戻ります。
ステップ 14	interface port-channel <i>channel-name.number</i> 例 :	指定したサブインターフェイス ポート チャンネルのインターフェイス設定モードを開始します。

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
	apic1(config-leaf)# interface port-channel po1.2001	
ステップ 15	vrf member vrf-name tenant tenant-name 例 : apic1(config-leaf-if)# vrf member v1 tenant t1	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 チャンネルを関連付けます。場所 : <ul style="list-style-type: none"> • vrf-name 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。 • tenant-name 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。
ステップ 16	exit 例 : apic1(config-leaf-if)# exit	設定モードに戻ります。

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。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 2001
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member v1 tenant t1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
apic1(config-leaf-if)# ipv6 link-local fe80::1
apic1(config-leaf-if)# mac-address 00:44:55:66:55::01
apic1(config-leaf-if)# mtu 1500
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel po1.2001
apic1(config-leaf-if)# vrf member v1 tenant t1
apic1(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、以前に構成したレイヤ 3 ポート チャンネルにポートを追加します。

手順の概要

1. **configure**
2. **leaf** *node-id*
3. **interface Ethernet** *slot/port*
4. **channel-group** *channel-name*

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <code>apic1# configure</code>	グローバル構成モードを開始します。
ステップ 2	leaf <i>node-id</i> 例 : <code>apic1(config)# leaf 101</code>	リーフ スイッチまたはリーフ スイッチの構成を指定します。 <i>node-id</i> は、構成が適用される <i>node-id1-node-id2</i> 形式で 1 つの ID または ID の範囲になります。
ステップ 3	interface Ethernet <i>slot/port</i> 例 : <code>apic1(config-leaf)# interface Ethernet 1/1-2</code>	構成するインターフェイスのインターフェイス構成モードを開始します。
ステップ 4	channel-group <i>channel-name</i> 例 : <code>apic1(config-leaf-if)# channel-group p01</code>	チャンネル グループでポートを構成します。

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
```

NX-OS Style CLI を使用したスイッチ仮想インターフェイスの設定

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。

手順の概要

1. コンフィギュレーション モードを開始します。
2. スイッチ モードを開始します。
3. VLAN インターフェイスを作成します。
4. カプセル化の範囲を指定します。
5. インターフェイス モードを終了します。

手順の詳細

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	カプセル化の範囲を指定します。 例： <code>apicl(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>apicl(config-leaf-if)# exit</code>	インターフェイス モードを終了します。

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

- テナントと 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。

手順の概要

1. コンフィギュレーション モードを開始します。
2. スイッチ モードを開始します。
3. VLAN インターフェイスを作成します。
4. SVI 自動状態を有効にします。
5. インターフェイス モードを終了します。

手順の詳細

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 : <code>apic1(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>apic1(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>apic1(config-leaf-if)# exit</code>	インターフェイス モードを終了します。

NX-OS Style CLI を使用したルーティング プロトコルの設定

NX-OS Style CLI を使用した BFD サポート付き BGP 外部ルーテッド ネットワークの設定

NX-OS スタイルの CLI を使用した BGP 外部ルーテッド ネットワークの設定

手順

ここでは、NX-OS CLI を使用して BGP 外部ルーテッド ネットワークを設定する方法を示します。

例：

```
apic1(config-leaf)# template route-profile damp_rp tenant t1
This template will be available on all leaves where tenant t1 has a VRF deployment
apic1(config-leaf-template-route-profile)# set dampening 15 750 2000 60
apic1(config-leaf-template-route-profile)# exit
apic1(config-leaf)#
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant t1 vrf ctx3
apic1(config-leaf-bgp-vrf)# neighbor 32.0.1.0/24 l3out l3out-bgp
apic1(config-leaf-bgp-vrf-neighbor)# update-source ethernet 1/16.401
apic1(config-leaf-bgp-vrf-neighbor)# address-family ipv4 unicast
apic1(config-leaf-bgp-vrf-neighbor-af)# weight 400
apic1(config-leaf-bgp-vrf-neighbor-af)# exit
apic1(config-leaf-bgp-vrf-neighbor)# remote-as 65001
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control remove-exclusive
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control remove-exclusive-all
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control remove-exclusive-all-replace-as
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# address-family ipv4 unicast
apic1(config-leaf-bgp-vrf-af)# inherit bgp dampening damp_rp
This template will be inherited on all leaves where VRF ctx3 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 dampening damp_rp
This template will be inherited on all leaves where VRF ctx3 has been deployed
apic1(config-leaf-bgp-vrf-af)# exit
```

NX-OS スタイルの CLI を使用した BGP 最大パスの設定

始める前に

次のフィールドの許容値は、『*Verified Scalability Guide for Cisco APIC*』を参照します。これは、[Cisco APIC ドキュメンテーション ページ](#)にあります。

適切なテナントと BGP 外部ルーテッド ネットワークが作成され、使用可能になっています。

BGP にログインして、次のコマンドを使用します：

- eBGP パスのマルチパスを構成するためのコマンド：

```
maximum-paths <value>
```

```
no maximum-paths <value>
```

- iBGP パスのマルチパスを構成するためのコマンド：

```
maximum-paths ibgp <value>
no maximum-paths ibgp <value>
```

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 ?
<1-64> Number of parallel paths
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#
```

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

手順の概要

1. 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 コマンドには、
apicl(config-leaf-vrf-template-route-profile)# set
as-path {'prepend as-num [,... as-num] | prepend-last-as num}の形式です。

手順の詳細

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 コマンドには、apicl(config-leaf-vrf-template-route-profile)# set as-path {'prepend as-num [,... as-num] | prepend-last-as num}の形式です。

例：

```
apicl(config)# leaf 103
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# template route-profile rpl
apicl(config-leaf-vrf-template-route-profile)# set as-path ?
prepend Prepend to the AS-Path
```

NX-OS Style CLI を使用した BGP ネイバー シャットダウンの設定

```
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 }
```

NX-OS Style CLI を使用した BGP ネイバー シャットダウンの設定

NX-OS Style CLI を使用した BGP ネイバー シャットダウンの構成

次の手順では、NX-OS CLI を使用して BGP ネイバー シャットダウン機能を使用する方法について説明します。

手順

ステップ 1 L3Out のノードとインターフェイスを構成します。

この例では、ノード103（境界リーフスイッチ）の VRF v1 を構成します。これは、nodep1と名付けられています。ルータ ID 11.11.11.103付きです。また、インターフェイス eth1/3 をルーテッドインターフェイス（レイヤ3ポート）として構成します。これは、IP アドレス 12.12.12.3/24 およびレイヤ3 ドメイン dom1 と構成します。

例：

```
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)# exit
```

ステップ 2 BGP ルーティング プロトコルを構成します。

この例では 15.15.15.2、および ASN 100 の BGP ピア 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 を構成します。

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
```

ステップ3 BGPネイバーシャットダウン機能を使用します。

例：

```
apic1(config-leaf-bgp-vrf-neighbor)# shutdown
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# 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

手順の概要

1. 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。
2. タイマー ポリシーを作成します。
3. 設定された BGP ポリシーを表示します。
4. ノードで特定のポリシーを参照します。
5. 特定の BGP のタイマー ポリシーのノードが表示されます。

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ1	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。 例： <pre>apic1(config)# apic1(config)# bgp-fabric apic1(config-bgp-fabric)# route-reflector spine 102 apic1(config-bgp-fabric)# asn 42 apic1(config-bgp-fabric)# exit apic1(config)# exit apic1#</pre>	
ステップ2	タイマー ポリシーを作成します。 例： <pre>apic1# config apic1(config)# leaf 101 apic1(config-leaf)# template bgp timers pol7 tenant tn1 This template will be available on all nodes where</pre>	特定の値は、例としてのみ提供されます。

	コマンドまたはアクション	目的
	<pre> tenant tn1 has a VRF deployment apic1(config-bgp-timers)# timers bgp 120 240 apic1(config-bgp-timers)# graceful-restart stalepath-time 500 apic1(config-bgp-timers)# maxas-limit 300 apic1(config-bgp-timers)# exit apic1(config-leaf)# exit apic1(config)# exit apic1# </pre>	
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# show run leaf 101 template bgp timers pol7 # Command: show running-config leaf 101 template bgp timers pol7 leaf 101 template bgp timers pol7 tenant tn1 timers bgp 120 240 graceful-restart stalepath-time 500 maxas-limit 300 exit exit </pre>	
ステップ 4	<p>ノードで特定のポリシーを参照します。</p> <p>例 :</p> <pre> apic1# config apic1(config)# leaf 101 apic1(config-leaf)# router bgp 42 apic1(config-leaf-bgp)# vrf member tenant tn1 vrf ctx1 apic1(config-leaf-bgp-vrf)# inherit node-only bgp timer pol7 apic1(config-leaf-bgp-vrf)# exit apic1(config-leaf-bgp)# exit apic1(config-leaf)# exit apic1(config)# exit apic1# </pre>	
ステップ 5	<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> apic1# 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 </pre>	

	コマンドまたはアクション	目的
	<code>exit</code> <code>apicl#</code>	

NX-OS スタイルの CLI を使用したセカンダリ IP アドレスでの双方向フォワーディング検出の設定

この手順では、NX-OS スタイルの CLI を使用して、セカンダリ IP アドレスに双方向転送検出 (BFD) を設定します。この例ではノード 103 (border リーフ スイッチ) で、ルータ ID を 11.11.11.103 で VRF v1 を構成します。また、インターフェイス eth1/3 をルーテッド インターフェイス (レイヤ 3 のポート) として構成し、IP アドレス 12.12.12.3/24 をプライマリ アドレスとして、6.11.1.224/24 をレイヤー 3 ドメイン dom1 のセカンダリ アドレスとして構成します。BFD は 99.99.99.14/32 で有効になっており、セカンダリ サブネット 6.11.1.0/24 を使用して到達可能です。

手順

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 :

```
apicl# configure terminal
```

ステップ 2 リーフ スイッチ 103 の構成モードを開始します。

例 :

```
apicl(config)# leaf 103
```

ステップ 3 VRF インスタンスの構成モードを開始します。

例 :

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

ステップ 4 セカンダリ IP アドレスを構成します。

例 :

```
apicl(config-leaf-vrf)# router-id 1.1.24.24
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)# ip address 6.11.1.224/24 secondary
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
```

ステップ 5 BFD を有効にします。

例 :

```
apicl(config-leaf)# vrf context tenant t1 vrf v1 l3out Routed
apicl(config-leaf-vrf)# router-id 1.1.24.24
```

```
apic1(config-leaf-vrf)#ip route 95.95.95.95/32 12.12.12.4 bfd
apic1(config-leaf-vrf)#ip route 99.99.99.14/32 6.11.1.100 bfd
```

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 :

```
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 :

```
apic1# configure
apic1(config)# template leaf-policy-group test_leaf_policy_group
apic1(config-leaf-policy-group)# [no] inherit bfd ip bfd_ipv4_global_policy
apic1(config-leaf-policy-group)# [no] inherit bfd ipv6 bfd_ipv6_global_policy
apic1(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 :

```
apic1(config)# leaf-profile test_leaf_profile
apic1(config-leaf-profile)# leaf-group test_leaf_group
apic1(config-leaf-group)# leaf-policy-group test_leaf_policy_group
```

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

```
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：

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

NX-OS スタイルの CLI を使用して BFD インターフェイスのオーバーライドを設定する

```
apic1(config)# spine-profile test_spine_profile
apic1(config-spine-profile)# spine-group test_spine_group
apic1(config-spine-group)# spine-policy-group test_spine_policy_group
apic1(config-spine-group)# spine 103-104
apic1(config-leaf-group)# exit
```

NX-OS スタイルの CLI を使用して BFD インターフェイスのオーバーライドを設定する

手順

ステップ1 NX-OS CLI を使用して BFD インターフェイス ポリシー (bfdIfPol) を設定するには:

例:

```
apic1# configure
apic1(config)# tenant t0
apic1(config-tenant)# vrf context v0
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t0 vrf v0
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface Ethernet 1/18
apic1(config-leaf-if)# vrf member tenant t0 vrf v0
apic1(config-leaf-if)# exit
apic1(config-leaf)# template bfd bfdIfPol1 tenant t0
apic1(config-template-bfd-pol)# [no] echo-mode enable
apic1(config-template-bfd-pol)# [no] echo-rx-interval 500
apic1(config-template-bfd-pol)# [no] min-rx 70
apic1(config-template-bfd-pol)# [no] min-tx 100
apic1(config-template-bfd-pol)# [no] multiplier 5
apic1(config-template-bfd-pol)# [no] optimize subinterface
apic1(config-template-bfd-pol)# exit
```

ステップ2 NX-OS CLI を使用して、以前に作成した BFD インターフェイス ポリシーを、IPv4 アドレスを持つ L3 インターフェイスに継承させるには:

例:

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface Ethernet 1/15
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdPol1
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password
```

ステップ3 NX-OS CLI を使用して、以前に作成した BFD インターフェイス ポリシーを、IPv6 アドレスを持つ L3 インターフェイスに継承させるには:

例:

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface Ethernet 1/15
apic1(config-leaf-if)# ipv6 address 2001::10:1/64 preferred
apic1(config-leaf-if)# bfd ipv6 tenant mode
```

```
apic1(config-leaf-if)# bfd ipv6 inherit interface-policy bfdPoll1
apic1(config-leaf-if)# bfd ipv6 authentication keyed-sha1 key 10 key password
```

ステップ 4 NX-OS CLI を使用して、IPv4 アドレスを持つ VLAN インターフェイス上の BFD を設定するには:

例 :

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 15
apic1(config-leaf-if)# vrf member tenant t0 vrf v0
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdPoll1
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password
```

ステップ 5 NX-OS CLI を使用して、IPv6 アドレスを持つ VLAN インターフェイス上の BFD を設定するには:

例 :

```
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 t0 vrf v0
apic1(config-leaf-if)# bfd ipv6 tenant mode
apic1(config-leaf-if)# bfd ipv6 inherit interface-policy bfdPoll1
apic1(config-leaf-if)# bfd ipv6 authentication keyed-sha1 key 10 key password
```

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 :

```

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

apic1# configure
apic1(config)# spine 103
apic1(config-spine)# interface ethernet 5/3.4
apic1(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 :

```

apic1(config-leaf-vrf)# [no] ip route 10.0.0.1/16 10.0.0.5 bfd

apic1(config)# spine 103
apic1(config-spine)# vrf context tenant infra vrf overlay-1
apic1(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 :

```

apic1(config)# leaf 101
apic1(config-spine)# interface ethernet 1/49
apic1(config-spine-if)# isis bfd enabled
apic1(config-spine-if)# exit
apic1(config-spine)# exit

apic1(config)# spine 103
apic1(config-spine)# interface ethernet 5/2
apic1(config-spine-if)# isis bfd enabled
apic1(config-spine-if)# exit
apic1(config-spine)# exit

```

NX-OS Style CLI を使用した OSPF 外部ルーテッド ネットワークの設定

NX-OS CLI を使用したテナントの OSPF 外部ルーテッド ネットワークの作成

外部ルーテッド ネットワーク接続の設定には、次のステップがあります。

1. テナントの下に VRF を作成します。
2. 外部ルーテッド ネットワークに接続された境界リーフ スイッチの VRF の L3 ネットワーキング構成を設定します。この設定には、インターフェイス、ルーティング プロトコル (BGP、OSPF、EIGRP)、プロトコル パラメータ、ルートマップが含まれています。
3. テナントの下に外部 L3 EPG を作成してポリシーを設定し、これらの EPG を境界リーフ スイッチに導入します。ACI ファブリック内で同じポリシーを共有する VRF の外部ルーテッド サブネットが、1 つの「外部 L3 EPG」または 1 つの「プレフィクス EPG」を形成します。

設定は、2 つのモードで実現されます。

- テナント モード：VRF の作成および外部 L3 EPG 設定
- リーフ モード：L3 ネットワーキング構成と外部 L3 EPG の導入

次の手順は、テナントの OSPF 外部ルーテッドネットワークを作成するためのものです。テナントの OSPF 外部ルーテッドネットワークを作成するには、テナントを選択してからテナント用の VRF を作成する必要があります。



(注) この項の例では、テナント「exampleCorp」の「OnlineStore」アプリケーションの「web」epg に外部ルーテッド接続を提供する方法について説明します。

手順

ステップ 1 VLAN ドメインを設定します。

例：

```
apicl(config)# vlan-domain dom_exampleCorp
apicl(config-vlan)# vlan 5-1000
apicl(config-vlan)# exit
```

ステップ 2 テナント VRF を設定し、VRF のポリシーの適用を有効にします。

例：

```
apicl(config)# tenant exampleCorp
apicl(config-tenant)# vrf context
exampleCorp_v1
apicl(config-tenant-vrf)# contract enforce
apicl(config-tenant-vrf)# exit
```

ステップ 3 テナント BD を設定し、ゲートウェイ IP を「public」としてマークします。エントリ「scope public」は、このゲートウェイ アドレスを外部 L3 ネットワークのルーティングプロトコルによるアドバタイズに使用できるようにします。

例：

```
apicl(config-tenant)# bridge-domain exampleCorp_b1
apicl(config-tenant-bd)# vrf member exampleCorp_v1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# interface bridge-domain exampleCorp_b1
apicl(config-tenant-interface)# ip address 172.1.1.1/24 scope public
apicl(config-tenant-interface)# exit
```

ステップ 4 リーフの VRF を設定します。

例：

```
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant exampleCorp vrf exampleCorp_v1
```

ステップ 5 OSPF エリアを設定し、ルート マップを追加します。

例 :

```

apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant exampleCorp vrf exampleCorp_v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.1 route-map map100 out
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit

```

ステップ 6 VRF をインターフェイス (この例ではサブインターフェイス) に割り当て、OSPF エリアを有効にします。

例 :

(注)

サブインターフェイスの構成では、メインインターフェイス (この例では、`ethernet 1/11`) は、「no switchport」によって L3 ポートに変換し、サブインターフェイスが使用するカプセル化 VLAN を含む `vlan` ドメイン (この例では `dom_exampleCorp`) を割り当てる必要があります。サブインターフェイス `ethernet1/11.500` で、500 はカプセル化 VLAN です。

```

apic1(config-leaf)# interface ethernet 1/11
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vlan-domain member dom_exampleCorp
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/11.500
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf exampleCorp_v1
apic1(config-leaf-if)# ip address 157.10.1.1/24
apic1(config-leaf-if)# ip router ospf default area 0.0.0.1

```

ステップ 7 外部 L3 EPG ポリシーを設定します。これは、外部サブネットを特定し、epg 「web」と接続する契約を消費するために一致させるサブネットが含まれます。

例 :

```

apic1(config)# tenant t100
apic1(config-tenant)# external-l3 epg l3epg100
apic1(config-tenant-l3ext-epg)# vrf member v100
apic1(config-tenant-l3ext-epg)# match ip 145.10.1.0/24
apic1(config-tenant-l3ext-epg)# contract consumer web
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)#exit

```

ステップ 8 リーフ スイッチの外部 L3 EPG を導入します。

例 :

```

apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t100 vrf v100
apic1(config-leaf-vrf)# external-l3 epg l3epg100

```

NX-OS Style CLI を使用した EIGRP 外部ルーテッドネットワークの設定

NX-OS スタイルの CLI を使用した EIGRP の構成

手順

ステップ 1 SSH を [Application Policy Infrastructure Controller] ([APIC]) へファブリック :

例 :

```
# ssh admin@node_name
```

ステップ 2 構成モードを開始します :

例 :

```
apic1# configure
```

ステップ 3 テナントの構成モードを入力します。

例 :

```
apic1(config)# tenant tenant1
```

ステップ 4 テナントでレイヤ 3 Outside を構成します :

例 :

```
apic1(config-tenant)# show run
# Command: show running-config tenant tenant1
# Time: Tue Feb 16 09:44:09 2016
tenant tenant1
  vrf context l3out
    exit
  l3out l3out-L1
    vrf member l3out
    exit
  l3out l3out-L3
    vrf member l3out
    exit
  external-l3 epg tenant1 l3out l3out-L3
    vrf member l3out
    match ip 0.0.0.0/0
    match ip 3.100.0.0/16
    match ipv6 43:101::/48
    contract consumer default
    exit
  external-l3 epg tenant1 l3out l3out-L1
    vrf member l3out
    match ipv6 23:101::/48
    match ipv6 13:101::/48
    contract provider default
    exit
  exit
```

ステップ 5 リーフで EIGRP の VRF を構成します :

例 :

```
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant tenant1 vrf l3out l3out l3out-L1
```

```

apicl(config-leaf-vrf)# show run
# Command: show running-config leaf 101 vrf context tenant tenant1 vrf l3out l3out l3out-L1
# Time: Tue Feb 16 09:44:45 2016
leaf 101
  vrf context tenant tenant1 vrf l3out l3out l3out-L1
    router-id 3.1.1.1
    route-map l3out-L1_in
      scope global
      ip prefix-list tenant1 permit 1:102::/48
      match prefix-list tenant1
    exit
  exit
  route-map l3out-L1_out
    scope global
    ip prefix-list tenant1 permit 3.102.10.0/23
    ip prefix-list tenant1 permit 3.102.100.0/31
    ip prefix-list tenant1 permit 3.102.20.0/24
    ip prefix-list tenant1 permit 3.102.30.0/25
    ip prefix-list tenant1 permit 3.102.40.0/26
    ip prefix-list tenant1 permit 3.102.50.0/27
    ip prefix-list tenant1 permit 3.102.60.0/28
    ip prefix-list tenant1 permit 3.102.70.0/29
    ip prefix-list tenant1 permit 3.102.80.0/30
    ip prefix-list tenant1 permit 3.102.90.0/32
    <OUTPUT TRUNCATED>
    ip prefix-list tenant1 permit ::/0
    match prefix-list tenant1
  exit
  exit
  route-map l3out-L1_shared
    scope global
  exit
  exit
  exit
exit

```

ステップ6 EIGRP インターフェイス ポリシーを構成します：

例：

```

apicl(config-leaf)# template eigrp interface-policy tenant1 tenant tenant1
This template will be available on all leaves where tenant tenant1 has a VRF deployment
apicl(config-template-eigrp-if-pol)# show run
# Command: show running-config leaf 101 template eigrp interface-policy tenant1 tenant tenant1
# Time: Tue Feb 16 09:45:50 2016
leaf 101
  template eigrp interface-policy tenant1 tenant tenant1
    ip hello-interval eigrp default 10
    ip hold-interval eigrp default 30
    ip throughput-delay eigrp default 20 tens-of-micro
    ip bandwidth eigrp default 20
  exit
exit

```

ステップ7 EIGRP の VRF ポリシーを構成します：

例：

```

apicl(config-leaf)# template eigrp vrf-policy tenant1 tenant tenant1
This template will be available on all leaves where tenant tenant1 has a VRF deployment
apicl(config-template-eigrp-vrf-pol)# show run
# Command: show running-config leaf 101 template eigrp vrf-policy tenant1 tenant tenant1
# Time: Tue Feb 16 09:46:31 2016
leaf 101
  template eigrp vrf-policy tenant1 tenant tenant1
    metric version 64bit

```

```
exit
exit
```

ステップ8 EIGRP VLAN インターフェイスを構成し、インターフェイスで EIGRP を有効にします：

例：

```
apic1(config-leaf)# interface vlan 1013
apic1(config-leaf-if)# show run
# Command: show running-config leaf 101 interface vlan 1013
# Time: Tue Feb 16 09:46:59 2016
leaf 101
  interface vlan 1013
    vrf member tenant tenant1 vrf l3out
    ip address 101.13.1.2/24
    ip router eigrp default
    ipv6 address 101:13::1:2/112 preferred
    ipv6 router eigrp default
    ipv6 link-local fe80::101:13:1:2
    inherit eigrp ip interface-policy tenant1
    inherit eigrp ipv6 interface-policy tenant1
  exit
exit
apic1(config-leaf-if)# ip summary-address ?
  eigrp  Configure route summarization for EIGRP
apic1(config-leaf-if)# ip summary-address eigrp default 11.11.0.0/16 ?
  <CR>
apic1(config-leaf-if)# ip summary-address eigrp default 11.11.0.0/16
apic1(config-leaf-if)# ip summary-address eigrp default 11:11:1::/48
apic1(config-leaf-if)# show run
# Command: show running-config leaf 101 interface vlan 1013
# Time: Tue Feb 16 09:47:34 2016
leaf 101
  interface vlan 1013
    vrf member tenant tenant1 vrf l3out
    ip address 101.13.1.2/24
    ip router eigrp default
    ip summary-address eigrp default 11.11.0.0/16
    ip summary-address eigrp default 11:11:1::/48
    ipv6 address 101:13::1:2/112 preferred
    ipv6 router eigrp default
    ipv6 link-local fe80::101:13:1:2
    inherit eigrp ip interface-policy tenant1
    inherit eigrp ipv6 interface-policy tenant1
  exit
exit
```

ステップ9 物理インターフェイスに VLAN を適用します：

例：

```
apic1(config-leaf)# interface ethernet 1/5
apic1(config-leaf-if)# show run
# Command: show running-config leaf 101 interface ethernet 1 / 5
# Time: Tue Feb 16 09:48:05 2016
leaf 101
  interface ethernet 1/5
    vlan-domain member cli
    switchport trunk allowed vlan 1213 tenant tenant13 external-svi l3out l3out-L1
    switchport trunk allowed vlan 1613 tenant tenant17 external-svi l3out l3out-L1
    switchport trunk allowed vlan 1013 tenant tenant1 external-svi l3out l3out-L1
    switchport trunk allowed vlan 666 tenant ten_v6_cli external-svi l3out l3out_cli_L1
    switchport trunk allowed vlan 1513 tenant tenant16 external-svi l3out l3out-L1
    switchport trunk allowed vlan 1313 tenant tenant14 external-svi l3out l3out-L1
```

```

switchport trunk allowed vlan 1413 tenant tenant15 external-svi l3out l3out-L1
switchport trunk allowed vlan 1113 tenant tenant12 external-svi l3out l3out-L1
switchport trunk allowed vlan 712 tenant mgmt external-svi l3out inband_l1
switchport trunk allowed vlan 1913 tenant tenant10 external-svi l3out l3out-L1
switchport trunk allowed vlan 300 tenant tenant1 external-svi l3out l3out-L1
exit
exit

```

ステップ 10 ルータ EIGRP を有効にします :

例 :

```

apic1(config-eigrp-vrf)# show run
# Command: show running-config leaf 101 router eigrp default vrf member tenant tenant1 vrf l3out
# Time: Tue Feb 16 09:49:05 2016
leaf 101
  router eigrp default
  exit
  router eigrp default
  exit
  router eigrp default
  exit
  router eigrp default
  vrf member tenant tenant1 vrf l3out
  autonomous-system 1001 l3out l3out-L1
  address-family ipv6 unicast
    inherit eigrp vrf-policy tenant1
  exit
  address-family ipv4 unicast
    inherit eigrp vrf-policy tenant1
  exit
  exit
exit

```

NX-OS スタイル CLI を使用したルート集約の設定

NX-OS スタイル CLI を使用した BGP、OSPF、および EIGRP のルート集約の設定

手順

ステップ 1 NX-OS CLI を使用して次のように BGP ルート集約を設定します:

- a) 次のように BGP を有効にします:

例 :

```

apic1(config)# pod 1
apic1(config-pod)# bgp fabric
apic1(config-pod-bgp)# asn 10
apic1(config-pod)# exit
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 10

```

- b) 次のように 要約ルートを設定します:

例 :

```
apicl(config-bgp)# vrf member tenant common vrf vrf1
apicl(config-leaf-bgp-vrf)# aggregate-address 10.0.0.0/8
```

ステップ 2 NX-OS CLI を使用して次のように OSPF 外部集約を設定します。

例 :

```
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant common vrf vrf1
apicl(config-leaf-ospf-vrf)# summary-address 10.0.0.0/8
```

ステップ 3 NX-OS CLI を使用して次のように OSPF エリア間集約を設定します。

```
apicl(config-leaf)# router ospf default
apicl(config-leaf-ospf)# vrf member tenant common vrf vrf1
apicl(config-leaf-ospf-vrf)# area 0.0.0.2 range 10.0.0.0/8 cost 20
```

ステップ 4 NX-OS CLI を使用して次のように EIGRP 集約を設定します。

例 :

```
apicl(config)# leaf 101
apicl(config-leaf)# interface ethernet 1/31 (Or interface vlan <vlan-id>)
apicl(config-leaf-if)# ip summary-address eigrp default 10.0.0.0/8
```

(注)

EIGRP を設定するルート集約ポリシーはありません。EIGRP の集約を有効にするために必要なだけの設定では、サマリー サブネット、InstP です。

NX-OS スタイルの CLI を使用したルート マップとルート プロファイルによるルート制御の構成

NX-OS Style CLI を使用した BGP ピアごとのルート制御の設定

次の手順では、NX-OS CLI を使用して BGP ピア単位のルート制御を設定する方法について説明します。

手順

ステップ 1 ルート グループ テンプレートを作成し、ルート グループに IP プレフィックスを追加します。

この例では、ルート グループ `match-rule1` を作成します。これは、テナント `t1` 用です。そしてルート グループへ `200.3.2.0/24` の IP プレフィックスを追加します。

例 :

```

apic1(config)# leaf 103
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)#

```

ステップ2 ノードのテナント VRF モードを開始します。

この例では、VRF v1 のテナント VRF モードを開始します。これは、テナント t1 用です。

例：

```

apic1(config-leaf)# vrf context tenant t1 vrf v1

```

ステップ3 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モードを開始します。すでに作成されているルートグループとマッチし、マッチモードを開始してルートプロファイルを設定します。

この例では、ルートマップ rp1 を作成します。そしてルートグループ match-rule1 を注文番号 0 に一致させます。

例：

```

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

```

ステップ4 BGP ルーティング プロトコルを構成します。

この例では、15.15.15.2、および ASN 100 の BGP ピア 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 を構成します。

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

```

ステップ5 BGP ピアごとのルート制御機能を構成します。

場所：

- **in** は、ルート インポート ポリシー（ファブリックに許可されるルート）です。
- **out** は、ルート エクスポート ポリシー（外部ネットワークからアドバタイズされるルート）です。

例：

```

apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

```

NX-OS スタイル CLI を使用して、明示的なプレフィックス リストでルート マップ/プロファイルの設定

始める前に

- テナントと VRF は、NX-OS CLI を介して設定する必要があります。
- NX-OS CLI を介してリーフ スイッチで VRF をイネーブルにする必要があります。

手順の概要

1. **configure**
2. **leaf node-id**
3. **template route group group-name tenant tenant-name**
4. **ip prefix permit prefix/masklen [le {32 | 128 }]**
5. **community-list [standard | expanded] community-list-name expression**
6. **exit**
7. **vrf context tenant tenant-name vrf vrf-name [l3out {BGP | EIGRP | OSPF | STATIC }]**
8. **template route-profile profile-name [route-control-context-name order-value]**
9. **set attribute value**
10. **exit**
11. **route-map map-name**
12. **match route group group-name [order number] [deny]**
13. **inherit route-profile profile-name**
14. **exit**
15. **exit**
16. **exit**
17. **router bgp fabric-asn**
18. **tl v1 vrf member tenant vrf**
19. **neighbor IP-address-of-neighbor**
20. **route-map map-name {in | out }**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : apicl# configure	コンフィギュレーション モードに入ります。
ステップ 2	leaf node-id 例 : apicl(config)# leaf 101	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	template route group <i>group-name</i> tenant <i>tenant-name</i> 例 : <pre>apic1(config-leaf)# template route group g1 tenant exampleCorp</pre>	ルート グループ テンプレートを作成します。 (注) ルート グループ (マッチ ルール) は、1 つ以上の IP プレフィックスと 1 つ以上のマッチ コミュニティ タームを持つことができます。マッチ タイプ全体では、AND フィルタがサポートされているため、ルート マッチ ルールが受け入れられるようにするために、ルート グループ内のすべての条件がマッチしている必要があります。ルート グループに複数の IP プレフィックスがある場合は、OR フィルタがサポートされます。マッチする場合は、いずれかのプレフィックスがルート タイプとして受け入れられます。
ステップ 4	ip prefix permit <i>prefix/masklen</i> [le { 32 128 }] 例 : <pre>apic1(config-route-group)# ip prefix permit 15.15.15.0/24</pre>	ルート グループに IP プレフィックスを追加します。 (注) IP プレフィックスは、BD サブネットまたは外部 ネットワークを示すことができます。集約プレフィックスが必要な場合は、IPv4 にはオプションの le 32 を、IPv6 には le 128 を使用してください。
ステップ 5	community-list [standard expanded] <i>community-list-name expression</i> 例 : <pre>apic1(config-route-group)# community-list standard com1 65535:20</pre>	これは任意のコマンドです。コミュニティも IP プレフィックスと照合する必要がある場合は、コミュニティのマッチ基準を追加します。
ステップ 6	exit 例 : <pre>apic1(config-route-group)# exit apic1(config-leaf)#</pre>	テンプレート モードを終了します。
ステップ 7	vrf context tenant <i>tenant-name</i> vrf <i>vrf-name</i> [l3out { BGP EIGRP OSPF STATIC }] 例 : <pre>apic1(config-leaf)# vrf context tenant exampleCorp vrf v1</pre>	ノードのテナント VRF モードを開始します。 (注) オプションの l3out 文字列を入力する場合、L3Out は NX-OS CLI で構成した L3Out である必要があります。
ステップ 8	template route-profile <i>profile-name</i> <i>[route-control-context-name order-value]</i> 例 :	マッチするルートに適用する必要があるセット アクションを含む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
	<code>apicl(config-leaf-vrf) # template route-profile rp1 ctx1 1</code>	
ステップ 9	set attribute value 例 : <code>apicl(config-leaf-vrf-template-route-profile) # set metric 128</code>	必要な属性 (アクションの設定) をテンプレートに追加します。
ステップ 10	exit 例 : <code>apicl(config-leaf-vrf-template-route-profile) # exit apicl(config-leaf-vrf) #</code>	テンプレート モードを終了します。
ステップ 11	route-map map-name 例 : <code>apicl(config-leaf-vrf) # route-map bgpMap</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 モードを開始します。
ステップ 12	match route group group-name [order number] [deny] 例 : <code>apicl(config-leaf-vrf-route-map) # match route group g1 order 1</code>	すでに作成されているルートグループとマッチし、マッチ モードを開始してルートプロファイルを設定します。さらに、ルートグループで定義されているマッチ基準にマッチするルートを拒否する必要がある場合は、キーワード 拒否 を選択します。デフォルトは次のとおりです。 許可 。
ステップ 13	inherit route-profile profile-name 例 : <code>apicl(config-leaf-vrf-route-map-match) # inherit route-profile rp1</code>	ルートプロファイルを継承します (アクションを設定します)。 (注) これらのアクションは、マッチしたルートに適用されます。または、ルートプロファイルを継承する代わりに、インラインで設定されたアクションを設定することもできます。
ステップ 14	exit 例 : <code>apicl(config-leaf-vrf-route-map-match) # exit apicl(config-leaf-vrf-route-map) #</code>	一致モードを終了します。
ステップ 15	exit 例 : <code>apicl(config-leaf-vrf-route-map) # exit apicl(config-leaf-vrf) #</code>	ルートマップ コンフィギュレーション モードを終了します。
ステップ 16	exit 例 :	VRF コンフィギュレーションモードを終了します。

	コマンドまたはアクション	目的
	<pre>apic1(config-leaf-vrf)# exit apic1(config-leaf)#</pre>	
ステップ 17	router bgp fabric-asn 例 : <pre>apic1(config-leaf)# router bgp 100</pre>	リーフ ノードを設定します。
ステップ 18	t1 vl vrf member tenant vrf 例 : <pre>apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1</pre>	BGP ポリシーの BGP の VRF メンバーシップとテナントを設定します。
ステップ 19	neighbor IP-address-of-neighbor 例 : <pre>apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2</pre>	BGP ネイバーを設定します。
ステップ 20	route-map map-name {in out } 例 : <pre>apic1(config-leaf-bgp-vrf-neighbor)# route-map bgpMap out</pre>	BGP ネイバのルート マップを設定します。

NX-OS スタイルの CLI を使用した、インポート制御とエクスポート制御を使用するルート制御プロトコルの設定

この例では、ネットワーク接続 BGP を使用して外部レイヤ 3 が設定されていることを前提としています。OSPF を使用するように設定されたネットワークに対してもこれらのタスクを実行することができます。

ここでは、NX-OS CLI を使用してルート マップを作成する方法を説明します。

始める前に

- テナント、プライベートネットワーク、およびブリッジドメインが作成されていること。
- レイヤ 3 Outside テナント ネットワークが設定されていること。

手順

ステップ 1 一致コミュニティ、一致プレフィックス リストを使用したインポート ルート制御

例 :

```
apic1# configure
apic1(config)# leaf 101
# Create community-list
apic1(config-leaf)# template community-list standard CL_1 65536:20 tenant exampleCorp
```

```

apicl(config-leaf)# vrf context tenant exampleCorp vrf v1

#Create Route-map and use it for BGP import control.
apicl(config-leaf-vrf)# route-map bgpMap
# Match prefix-list and set route-profile actions for the match.
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
apicl(config-leaf-vrf-route-map)# match prefix-list list1
apicl(config-leaf-vrf-route-map-match)# set tag 200
apicl(config-leaf-vrf-route-map-match)# set local-preference 64
apicl(config-leaf)# router bgp 100
apicl(config-bgp)# vrf member tenant exampleCorp vrf v1
apicl(config-leaf-bgp-vrf)# neighbor 3.3.3.3
apicl(config-leaf-bgp-vrf-neighbor)# route-map bgpMap in

```

ステップ2 一致 BD、デフォルトのエクスポート ルート プロファイルを使用したエクスポート ルート制御

例：

```

# Create custom and "default-export" route-profiles
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 256
apicl(config-leaf-vrf)# template route-profile bd-rtctrl
apicl(config-leaf-vrf-template-route-profile)# set metric 128

#Create a Route-map and match on BD, prefix-list
apicl(config-leaf-vrf)# route-map bgpMap
apicl(config-leaf-vrf-route-map)# match bridge-domain bd1
apicl(config-leaf-vrf-route-map-match)#exit
apicl(config-leaf-vrf-route-map)# match prefix-list p1
apicl(config-leaf-vrf-route-map-match)#exit
apicl(config-leaf-vrf-route-map)# match bridge-domain bd2
apicl(config-leaf-vrf-route-map-match)# inherit route-profile bd-rtctrl

```

(注)

この場合、bd1 のパブリック サブネットとプレフィックスリスト p1 を照合するプレフィックスが、ルート プロファイルの「default-export」を使用してエクスポートされ、bd2 のパブリック サブネットはルート プロファイルの「bd-rtctrl」を使用してエクスポートされます。

NX-OS Style CLI を使用したインターリーク再配布の構成

次の手順では、NX-OS スタイルの CLI を使用してインターリーク再配布を構成する方法について説明します。

始める前に

テナント、VRF および L3Out を作成します。

手順

ステップ 1 境界リーフ ノードのインターリーク再配布のルート マップを構成します。

例：

次の例では、ルート マップ `CLI_RP` を構成します。これは、IP プレフィックスリスト `CLI_PFX1` 付きです。この、IP プレフィックスリストは、テナント `CLI_TEST` および VRF `VRF1` 用です：

```
apic1# conf t
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant CLI_TEST vrf VRF1
apic1(config-leaf-vrf)# route-map CLI_RP
apic1(config-leaf-vrf-route-map)# ip prefix-list CLI_PFX1 permit 192.168.1.0/24
apic1(config-leaf-vrf-route-map)# match prefix-list CLI_PFX1 [deny]
```

ステップ 2 構成されたルート マップを使用して、インターリーク再配布を構成します。

例：

次に、構成されたルート マップ [`CLI_RP`] を使用して OSPF ルートの再配布を構成する例を示します `CLI_RP`：

```
apic1# conf t
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 65001
apic1(config-leaf-bgp)# vrf member tenant CLI_TEST vrf VRF1
apic1(config-leaf-bgp-vrf)# redistribute ospf route-map CLI_RP
```

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

NX-OS スタイル CLI を使用したトランジット ルーティングの構成

次の手順では、テナントのトランジットルーティングを構成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 ドメインを構成します。
- `vlan-domain domain` および `vlan vlan-range` コマンドを使用して VLAN ドメインを構成します。
- BGP ルート リフレクタ ポリシーを構成し、ファブリック内でルーテッドを伝達します。

手順

ステップ1 テナントおよび VRF を構成します。

この例は、テナント t1 VRF v1付きで構成します。VRF 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
```

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 を構成します。

- 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 で構成されます。ルーテッド インターフェイス ifp1 が eth1/3 にあります。これは、IP アドレス 12.12.12.3/24 付きです。
- 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 に構成されています。これは、ルーテッド インターフェイス ifp2 が eth1/3 にあります。これは、IP アドレス 23.23.23.1/24 付きです。

例：

```
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)# exit
apicl(config)# leaf 102
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# router-id 22.22.22.203
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 23.23.23.3/24
apicl(config-leaf-if)# exit
apicl(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 プロトコルを構成します。

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 を構成します。ノード 101 を BGP ピア 15.15.15.2 およびノード 102 を BGP ピア 25.25.25.2 と共に構成します。

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
apic1(config)# leaf 102
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)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit
```

ステップ4 接続ルーティング プロトコルを構成します。

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 を構成します。

例：

```
apic1(config)# leaf 101
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 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)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 60.60.60.1
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit
```

ステップ5 外部 EPG を構成します。

この例は、ネットワーク 192.168.1.0/24 を外部ネットワーク extnw1 としてノード 101 を構成します。そして、ネットワーク 192.168.2.0/24 を外部ネットワーク extnw2 としてノード 102 を構成します。

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 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
```

```

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 マップを構成します。

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 マップを構成します。

例：

例：

```

apicl(config)# leaf 101
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 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-rule2 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-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 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

```

例：中継ルーティング

```
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：

```
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)# exit
```

ステップ8 契約を構成し、Epg に関連付けます。

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)# contract provider httpCtrct
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)# contract consumer httpCtrct
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(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。

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

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)# 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 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)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# router-id 22.22.22.203
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 23.23.23.3/24
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 25.25.25.2/24
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 60.60.60.3
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 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

apicl(config)# leaf 101
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
```

```

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-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 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

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 httpCtct
apic1(config-tenant-contract)# scope vrf
apic1(config-tenant-contract)# subject http-subj
apic1(config-tenant-contract-subj)# access-group http-filter both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
apic1(config-tenant)# 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)# contract provider httpCtct
apic1(config-tenant-l3ext-epg)# exit
apic1(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)#

```

NX-OS Style CLI を使用した共有サービスの設定

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

手順の概要

1. コンフィギュレーション モードを開始します。
2. プロバイダー レイヤ 3 を設定します。
3. レイヤ 3 Out コンシューマを設定します。

手順の詳細

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# configure </pre>	
ステップ 2	<p>プロバイダー レイヤ 3 を設定します。</p> <p>例 :</p> <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 t1_provider vrf VRF1 l3out T0-o1-L3OUT-1 apicl(config-leaf-vrf)# route-map T0-o1-L3OUT-1_shared apicl(config-leaf-vrf-route-map)# ip prefix-list l3extInstP-1 permit 192.168.2.0/24 apicl(config-leaf-vrf-route-map)# match prefix-list l3extInstP-1 apicl(config-leaf-vrf-route-map-match)# exit apicl(config-leaf-vrf-route-map)# exit </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

	コマンドまたはアクション	目的
	<pre>apicl(config-leaf-vrf)# exit apicl(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>apicl(config)# tenant t1_consumer apicl(config-tenant)# external-l3 epg l3extInstP-2 l3out T0-o1-L3OUT-1 apicl(config-tenant-l3ext-epg)# vrf member VRF2 apicl(config-tenant-l3ext-epg)# match ip 199.16.2.0/24 shared apicl(config-tenant-l3ext-epg)# contract consumer vzBrCP-1 imported apicl(config-tenant-l3ext-epg)# exit apicl(config-tenant)# exit apicl(config)# leaf 101 apicl(config-leaf)# vrf context tenant t1_consumer vrf VRF2 l3out T0-o1-L3OUT-1 apicl(config-leaf-vrf)# route-map T0-o1-L3OUT-1_shared apicl(config-leaf-vrf-route-map)# ip prefix-list l3extInstP-2 permit 199.16.2.0/24 apicl(config-leaf-vrf-route-map)# match prefix-list l3extInstP-2 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)#</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

手順の概要

1. コンフィギュレーションモードを開始します。
2. プロバイダ テナントおよび VRF の設定
3. コンシューマ テナントおよび VRF の設定
4. コントラクトの設定
5. プロバイダ外部レイヤ 3 EPG の設定
6. プロバイダ エクスポート マップの設定
7. コンシューマ外部レイヤ 3 EPG の設定
8. コンシューマ エクスポート マップの設定

手順の詳細

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>apic1# configure</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>apic1(config)# tenant t1_provider apic1(config-tenant)# vrf context VRF1 apic1(config-tenant-vrf)# exit apic1(config-tenant)# exit</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>apic1(config)# tenant t1_consumer apic1(config-tenant)# vrf context VRF2 apic1(config-tenant-vrf)# exit apic1(config-tenant)# exit</pre>	
ステップ 4	<p>コントラクトの設定</p> <p>例 :</p> <pre>apic1(config)# tenant t1_provider apic1(config-tenant)# contract vzBrCP-1 type permit apic1(config-tenant-contract)# scope exportable apic1(config-tenant-contract)# export to tenant t1_consumer apic1(config-tenant-contract)# exit</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>apic1(config-tenant)# external-l3 epg l3extInstP-1 apic1(config-tenant-l3ext-epg)# vrf member VRF1 apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24 shared apic1(config-tenant-l3ext-epg)# contract provider vzBrCP-1 apic1(config-tenant-l3ext-epg)# exit apic1(config-tenant)# exit</pre>	
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</pre>	

	コマンドまたはアクション	目的
	<pre> apicl(config-leaf-vrf)# route-map map1 apicl(config-leaf-vrf-route-map)# ip prefix-list p1 permit 192.168.2.0/24 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 apicl(config-leaf-vrf)# export map map1 apicl(config-leaf-vrf)# exit apicl(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> apicl(config)# tenant t1_consumer apicl(config-tenant)# external-l3 epg l3extInstP-2 apicl(config-tenant-l3ext-epg)# vrf member VRF2 apicl(config-tenant-l3ext-epg)# match ip 199.16.2.0/24 shared apicl(config-tenant-l3ext-epg)# contract consumer vzBrCP-1 imported apicl(config-tenant-l3ext-epg)# exit apicl(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> apicl(config)# leaf 101 apicl(config-leaf)# vrf context tenant t1_consumer vrf VRF2 apicl(config-leaf-vrf)# route-map map2 apicl(config-leaf-vrf-route-map)# ip prefix-list p2 permit 199.16.2.0/24 apicl(config-leaf-vrf-route-map)# match prefix-list p2 apicl(config-leaf-vrf-route-map-match)# exit apicl(config-leaf-vrf-route-map)# exit apicl(config-leaf-vrf)# export map map2 apicl(config-leaf-vrf)# exit apicl(config-leaf)# exit apicl(config)# </pre>	

NX-OS スタイルの CLI を使用した L3Out の QoS の設定

CLI を使用した L3Out での QoS の直接構成

この章では L3Out で QoS ディレクトリを構成する方法について説明します。これは、[Cisco APIC] リリース 4.0 (1) 以降の L3Out QoS の推奨構成方法です。

次のオブジェクトの内の 1 つで L3Out の QoS を構成できます :

- スイッチ仮想インターフェイス (SVI)
- サブ インターフェイス

- 外部ルーテッド

手順

ステップ 1 L3Out SVI に QoS プライオリティを構成します。

例：

```
interface vlan 19
  vrf member tenant DT vrf dt-vrf
  ip address 107.2.1.252/24
  description 'SVI19'
  service-policy type qos VrfQos006 // for custom QoS attachment
  set qos-class level6 // for set QoS priority
  exit
```

ステップ 2 サブインターフェイスに QoS プライオリティを構成します。

例：

```
interface ethernet 1/48.10
  vrf member tenant DT vrf inter-tenant-ctx2 l3out L4_E48_inter_tenant
  ip address 210.2.0.254/16
  service-policy type qos vrfQos002
  set qos-class level5
```

ステップ 3 外部ルーテッドに QoS プライオリティを構成します。

例：

```
interface ethernet 1/37
  no switchport
  vrf member tenant DT vrf dt-vrf l3out L2E37
  ip address 30.1.1.1/24
  service-policy type qos vrfQos002
  set qos-class level5
  exit
```

CLI を使用した L3Out の QoS コントラクトの設定

この項では、コントラクトを使用して L3Out の QoS を設定する方法について説明します。



(注) リリース 4.0 (1) 以降では、[CLI を使用した L3Out での QoS の直接構成 \(76 ページ\)](#) で説明されているように L3OutQoS 用にカスタム QoS ポリシーを使用することを推奨しています。

手順

ステップ 1 L3Out で QoS 優先順位の適用をサポートするために、出力モードの VRF を構成し、ポリシー適用を有効化します。

```

apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# contract enforce egress
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)#

```

ステップ2 QoS を構成します。

フィルタを作成する場合（access-list）、**match dscp** コマンドをターゲット DSCP レベルで含めます。コントラクトを設定するとき、L3Out でのトラフィック出力の QoS クラスを含めます。または、ターゲット DSCP の値を定義することもできます。QoS ポリシーは、コントラクトまたはサブジェクトのいずれかでサポートされます。

L3out インターフェイスでの QoS またはカスタム QoS では VRF の適用は入力である必要があります。VRF の適用を出力にする必要があるのは、QoS 分類が EPG と L3out の間、または L3out から L3out へのトラフィックのコントラクトで実行される場合に限りです。

(注)

QoS 分類がコントラクトで設定され、VRF の適用が出力である場合、コントラクト QoS 分類は L3out インターフェイス QoS またはカスタム QoS 分類をオーバーライドします。

```

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)# match dscp EF
apic1(config-tenant-acl)# exit
apic1(config-tenant)# contract httpCtct
apic1(config-tenant-contract)# scope vrf
apic1(config-tenant-contract)# qos-class level1
apic1(config-tenant-contract)# subject http-subject
apic1(config-tenant-contract-subj)# access-group http-filter both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
apic1(config-tenant)# exit
apic1(config)#

```

NX-OS Style CLI を使用した ACI IP SLA の設定

NX-OS Style CLI を使用した IP SLA モニタリング ポリシーの構成

NX-OS スタイル CLI を使用して特定の SLA タイプのモニタリング プローブを送信するように [Cisco Application Policy Infrastructure Controller] ([APIC]) を構成するには、次の手順を実行します：

始める前に

テナントが構成されていることを確認します。

手順

ステップ 1 構成モードを開始します。

例：
`apicl# configure`

ステップ 2 テナントを作成してテナント構成モードを開始するか、既存のテナントのテナント構成モードを開始します。

例：
`apicl(config)# tenant t1`

ステップ 3 IP SLA モニタリング ポリシーを作成し、IP SLA ポリシー構成モードを開始します。

例：
`apicl(config-tenant)# ipsla-pol ipsla-policy-3`

ステップ 4 モニタリング頻度を秒単位で構成します。これはプローブの送信間隔です。

例：
`apicl(config-ipsla-pol)# sla-frequency 40`

ステップ 5 モニタリング プローブ タイプを構成します。

タイプに指定できる値は次のとおりです：

- icmp
- l2ping
- tcp sla-port *number*

スタティック ルートの IP SLA には ICMP と TCP のみが有効です。

例：
`apicl(config-ipsla-pol)# sla-type tcp sla-port 90`

次のタスク

作成した IP SLA モニタリング ポリシーを表示するには、次のように入力します：

show running-config all tenant *tenant-name* ipsla-pol

次の出力が表示されます：

```
# Command: show running-config all tenant 99 ipsla-pol
# Time: Tue Mar 19 19:01:06 2019
tenant t1
  ipsla-pol ipsla-policy-3
    sla-detectmultiplier 3
    sla-frequency 40
    sla-type tcp sla-port 90
```

```
sla-port 90
exit
exit
exit
```

NX-OS Style CLI を使用した IP-SLA トラック メンバーの設定

NX-OS スタイルの CLI を使用して IP SLA トラック メンバーを設定するには、次の手順を実行します。

始める前に

テナントおよびテナントの下の IP SLA モニタリング ポリシーが設定されていることを確認します。

手順

ステップ 1 **configure**

コンフィギュレーション モードに入ります。

例：

```
apic1# configure
```

ステップ 2 **tenant *tenant-name***

テナントを作成するか、テナント設定モードに入ります。

例：

```
apic1(config)# tenant t1
```

ステップ 3 **name ipv4-or-ipv6-address name **track-member** **dst-IPAddr** **l3-out****

宛先 IP アドレスを持つトラック メンバーを作成し、トラック メンバー コンフィギュレーション モードを開始します。

例：

```
apic1(config-tenant)# )# track-member tm-1 dst-IPAddr 10.10.10.1 l3-out ext-l3-1
```

ステップ 4 **ipsla-monpol *name***

トラック メンバーに IP SLA モニタリング ポリシーを割り当てます。

例：

```
apic1(config-track-member)# ipsla-monpol ipsla-policy-3
```

例

次の例は、IP SLA トラック メンバーを設定するコマンドを示しています。

```
apicl# configure
apicl(config)# tenant t1
apicl(config-tenant)# )# track-member tm-1 dst-IPAddr 10.10.10.1 l3-out ext-l3-1
apicl(config-track-member)# ipsla-monopol ipsla-policy-3
```

次のタスク

作成したトラック メンバー設定を表示するには、次のように入力します。

show running-config all tenant *tenant-name* track-member *name*

次の出力が表示されます。

```
# Command: show running-config all tenant 99 track-member tm-1
# Time: Tue Mar 19 19:01:06 2019
tenant t1
  track-member tm-1 10.10.10.1 l3-out ext-l3-1
    ipsla-monopol slaICMPProbe
  exit
exit
```

NX-OS Style CLI を使用した IP-SLA トラック リストの設定

NX-OS スタイルの CLI を使用して IP SLA トラック リストを設定するには、次の手順を実行します。

始める前に

テナント、IP SLA モニタリング ポリシー、およびテナント下の少なくとも 1 つのトラック メンバーが設定されていることを確認します。

手順

ステップ 1 **configure**

コンフィギュレーション モードに入ります。

例：

```
apicl# configure
```

ステップ 2 **tenant *tenant-name***

テナントを作成するか、テナント設定モードに入ります。

例：

```
apicl(config)# tenant t1
```

ステップ 3 **track-list *name* { percentage [percentage-down | percentage-up] *number* | weight [weight-down | weight-up *number*] }**

パーセンテージまたは重みしきい値の設定でトラックリストを作成し、トラック リスト コンフィギュレーション モードを開始します。

例：

```
apic1(config-tenant)# )# track-list tl-1 percentage percentage-down 50 percentage-up 100
```

ステップ 4 track-member name

既存のトラック メンバーをトラック リストに割り当てます。

例 :

```
apic1(config-track-list)# track-member tm-1
```

例

次の例は、IP SLA トラック リストを設定するコマンドを示しています。

```
apic1# configure
  apic1(config)# tenant t1
  apic1(config-tenant)# )# track-list tl-1 percentage percentage-down 50 percentage-up
100
  apic1(config-track-list)# track-member tml
```

次のタスク

作成したトラック メンバー設定を表示するには、次のように入力します。

show running-config all tenant tenant-name track-member name

次の出力が表示されます。

```
# Command: show running-config all tenant 99 track-list tl-1
# Time: Tue Mar 19 19:01:06 2019
tenant t1
  track-list tl-1 percentage percentage-down 50 percentage-up 100
  track-member tm-1 weight 10
  exit
exit
```

NX-OS Style CLI を使用したスタティック ルートとトラック リストの関連付け

NX-OS スタイル CLI を使用して IP SLA トラック リストをスタティック ルートに関連付けるには、次の手順を実行します。

始める前に

テナント、VRF およびテナントの下にあるトラック リストが設定されていることを確認してください。

手順

ステップ 1 configure

コンフィギュレーション モードに入ります。

例 :

```
apicl# configure
```

ステップ 2 leaf id または leaf-name

リーフ スイッチを選択し、リーフ スイッチ コンフィギュレーション モードを開始します。

例：

```
apicl(config)# leaf 102
```

ステップ 3 vrf context tenant name vrf name

VRF コンテキストを選択し、VRF コンフィギュレーション モードを開始します。

例：

```
apicl(config-leaf)# )# vrf context tenant 99 vrf default
```

ステップ 4 ip route ip-address next-hop-ip-address route-prefix bfd ip-trackList name

既存のトラック リストをスタティック ルートに割り当てます。

例：

```
apicl(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd ip-trackList tl-1
```

例

次に、IP SLA トラック リストをスタティック ルートに関連付けるコマンドの例を示します。

```
apicl# configure
apicl(config)# leaf 102
apicl(config-leaf)# )# vrf context tenant 99 vrf default
apicl(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd ip-trackList tl-1
```

NX-OS Style CLI を使用したトラック リストとネクスト ホップ プロファイルの関連付け

NX-OS スタイルの CLI を使用して IP SLA トラック リストをネクスト ホップ プロファイルに関連付けるには、次の手順を実行します。

始める前に

テナント、VRF およびテナントの下にあるトラック リストが設定されていることを確認してください。

手順

ステップ 1 configure

コンフィギュレーション モードに入ります。

例：

CLI を使用したトラック リストおよびトラック メンバー ステータスの表示

```
apic1# configure
```

ステップ 2 leaf id または leaf-name

リーフ スイッチを選択し、リーフ スイッチ コンフィギュレーション モードを開始します。

例 :

```
apic1(config)# leaf 102
```

ステップ 3 vrf context tenant name vrf name

VRF コンテキストを選択し、VRF コンフィギュレーション モードを開始します。

例 :

```
apic1(config-leaf)# )# vrf context tenant 99 vrf default
```

ステップ 4 ip route ip-address next-hop-ip-address route-prefix bfd nh-ip-trackList name

既存のトラック リストをネクスト ホップに割り当てます。

例 :

```
apic1(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd nh-trackList tl-1
```

例

次に、IP SLA トラック リストをネクスト ホップ プロファイルに関連付けるコマンドの例を示します。

```
apic1# configure
apic1(config)# leaf 102
apic1(config-leaf)# )# vrf context tenant 99 vrf default
apic1(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd nh-ip-trackList
tl-1
```

CLI を使用したトラック リストおよびトラック メンバー ステータスの表示

IP SLA トラック リストおよびトラック メンバー ステータスを表示できます。

手順

	コマンドまたはアクション	目的
ステップ 1	show track brief 例 : switch# show track brief	すべてのトラック リストおよびトラック メンバーのステータスを表示します。

例

```
switch# show track brief
TrackId  Type      Instance  Parameter  State  Last Change
```

97	IP SLA	2034	reachability	up	2019-03-20T14:08:34.127-07:00
98	IP SLA	2160	reachability	up	2019-03-20T14:08:34.252-07:00
99	List	---	percentage	up	2019-03-20T14:08:45.494-07:00
100	List	---	percentage	down	2019-03-20T14:08:45.039-07:00
101	List	---	percentage	down	2019-03-20T14:08:45.040-07:00
102	List	---	percentage	up	2019-03-20T14:08:45.495-07:00
103	IP SLA	2040	reachability	up	2019-03-20T14:08:45.493-07:00
104	IP SLA	2887	reachability	down	2019-03-20T14:08:45.104-07:00
105	IP SLA	2821	reachability	up	2019-03-20T14:08:45.494-07:00
1	List	---	percentage	up	2019-03-20T14:08:39.224-07:00
2	List	---	weight	down	2019-03-20T14:08:33.521-07:00
3	IP SLA	2412	reachability	up	2019-03-20T14:08:33.983-07:00
26	IP SLA	2320	reachability	up	2019-03-20T14:08:33.988-07:00
27	IP SLA	2567	reachability	up	2019-03-20T14:08:33.987-07:00
28	IP SLA	2598	reachability	up	2019-03-20T14:08:33.990-07:00
29	IP SLA	2940	reachability	up	2019-03-20T14:08:33.986-07:00
30	IP SLA	2505	reachability	up	2019-03-20T14:08:38.915-07:00
31	IP SLA	2908	reachability	up	2019-03-20T14:08:33.990-07:00
32	IP SLA	2722	reachability	up	2019-03-20T14:08:33.992-07:00
33	IP SLA	2753	reachability	up	2019-03-20T14:08:38.941-07:00
34	IP SLA	2257	reachability	up	2019-03-20T14:08:33.993-07:00

CLI を使用したトラック リストとトラック メンバーの詳細の表示

IP SLA トラック リストおよびトラック メンバーの詳細を表示できます。

手順

	コマンドまたはアクション	目的
ステップ 1	show track [<i>number</i>] more 例 : switch# show track more	すべてのトラック リストおよびトラック メンバーの詳細を表示します。

例

```
switch# show track | more
Track 4
  IP SLA 2758
  reachability is down
  1 changes, last change 2019-03-12T21:41:34.729+00:00
  Tracked by:
    Track List 3
    Track List 5

Track 3
  List Threshold percentage
  Threshold percentage is down
  1 changes, last change 2019-03-12T21:41:34.700+00:00
  Threshold percentage up 1% down 0%
  Tracked List Members:
    Object 4 (50)% down
    Object 6 (50)% down
  Attached to:
    Route prefix 172.16.13.0/24

Track 5
```

```

List Threshold percentage
Threshold percentage is down
1 changes, last change 2019-03-12T21:41:34.710+00:00
Threshold percentage up 1% down 0%
Tracked List Members:
  Object 4 (100)% down
Attached to:
  Nexthop Addr 12.12.12.2/32

Track 6
  IP SLA 2788
  reachability is down
  1 changes, last change 2019-03-14T21:34:26.398+00:00
  Tracked by:
    Track List 3
    Track List 7

Track 20
  List Threshold percentage
  Threshold percentage is up
  4 changes, last change 2019-02-21T14:04:21.920-08:00
  Threshold percentage up 100% down 32%
  Tracked List Members:
    Object 4 (20)% up
    Object 5 (20)% up
    Object 6 (20)% up
    Object 3 (20)% up
    Object 9 (20)% up
  Attached to:
    Route prefix 88.88.88.0/24
    Route prefix 5000:8:1:14::/64
    Route prefix 5000:8:1:2::/64
    Route prefix 5000:8:1:1::/64

```

この例では、Track 4 は IP SLA ID と **Tracked by:** フィールドのトラック リストによって識別されるトラック メンバーです。

Track 3 は、しきい値情報と [トラック リスト メンバー (**Track List Members**)] フィールドのトラック メンバーによって識別されるトラック リストです。

トラック 20 は、現在到達可能（アップ）で、関連付けられているスタティック ルートを示すトラック リストです。

NX-OS Style CLI を使用した HSRP の設定

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。

手順の概要

1. **configure**
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。

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>apic1# 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	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 : <pre>apic1(config)# leaf 101 apic1(config-leaf)# interface ethernet 1/17 apic1(config-leaf-if)# hsrp version 1 apic1(config-leaf-if)# hsrp use-bia apic1(config-leaf-if)# hsrp delay minimum 30 apic1(config-leaf-if)# hsrp delay reload 30 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)# timers 5 18 apic1(config-if-hsrp)# priority 100 apic1(config-if-hsrp)# preempt apic1(config-if-hsrp)# preempt delay minimum 60 apic1(config-if-hsrp)# preempt delay reload 60 apic1(config-if-hsrp)# preempt delay sync 60 apic1(config-if-hsrp)# authentication none apic1(config-if-hsrp)# authentication simple apic1(config-if-hsrp)# authentication md5 apic1(config-if-hsrp)# authentication-key <mypassword> apic1(config-if-hsrp)# authentication-key-timeout <timeout></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。

手順の概要

1. **configure**
2. HSRP ポリシー テンプレートを設定します。
3. 設定されているポリシー テンプレートを使用します。

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例 : <pre>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	HSRP ポリシー テンプレートを設定します。 例 : <pre> 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 apicl(config)# leaf 101 apicl(config-leaf)# template hsrp group-policy hsrp-groupPol1 tenant t9 apicl(config-template-hsrp-group-pol)# timers 5 </pre>	

	コマンドまたはアクション	目的
	<pre> 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>	

NX-OS Style CLI を使用した Cisco ACI GOLF の設定

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
```

[Cisco ACI GOLF] NX-OS スタイル CLI を使用した 構成の例

次の例を構成する CLI コマンドの [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。

BGP EVPN のテナント インフラの構成

次の例を構成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 の構成

次の例では、BGP EVPN をサポートする BGP を構成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 のテナントの構成

次の例では、BGPEVPN、BGPEVPNセッションで提供されるゲートウェイ サブネットを含むのテナントを構成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 の構成

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 マップを構成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

```

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	<p>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。</p> <p>例 :</p> <pre>apic1(config)# leaf 101 apic1(config-leaf)# template bgp address-family bgpAf1 tenant bgp_t1 apic1(config-bgp-af)# distance 250 240 230 apic1(config-bgp-af)# host-rt-enable apic1(config-bgp-af)# exit</pre>	<p>このテンプレートは、テナント bgp_t1 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、no host-rt-enable コマンドを使用します。</p>

翻訳について

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