



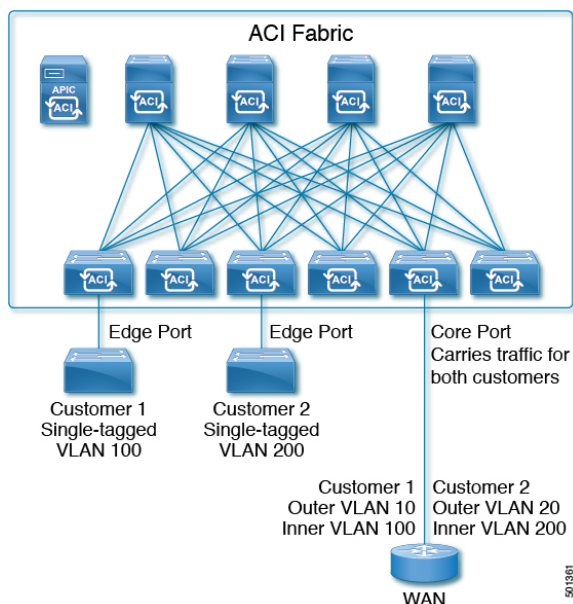
802.1 q トンネリング

この章は、次の内容で構成されています。

- [ACI 802.1 q トンネルについて \(1 ページ\)](#)
- [GUI を使用した 802.1Q トンネルの設定 \(3 ページ\)](#)
- [NX-OS スタイルの CLI を使用した 802.1Q トンネルの設定 \(6 ページ\)](#)
- [REST API を使用した 802.1Q トンネルの設定 \(10 ページ\)](#)

ACI 802.1 q トンネルについて

図 1: ACI 802.1 q トンネル



エッジ（トンネル）ポートで 802.1Q トンネルを設定して、Quality of Service (QoS) の優先順位設定とともに、ファブリックのイーサネットフレームの point-to-multi-point トンネリングを有効にできます。Dot1q トンネルは、タグなし、802.1Q タグ付き、802.1ad 二重タグ付きフレームを、ファブリックでそのまま送信します。各トンネルでは、単一の顧客からのトラフィック

を伝送し、単一のブリッジドメインに関連付けられています。Cisco Application Centric Infrastructure (ACI) の前面パネルポートは、Dot1q トンネルの一部とすることができます。レイヤ2 スイッチングは宛先 MAC (DMAC) に基づいて行われ、通常の MAC ラーニングはトンネルで行われます。エッジポート Dot1q トンネルは、スイッチモデル名の最後に「EX」またはそれ以降のサフィックスが付く、Cisco Nexus 9000 シリーズスイッチでサポートされません。

同じコアポートで複数の 802.1Q トンネルを設定することができ、複数の顧客からの二重タグ付きトラフィックを伝送できます。それぞれは、802.1Q トンネルごとに設定されたアクセスのカプセル化で識別されます。802.1Q トンネルでは、MAC アドレス学習を無効にすることもできます。エッジポートとコアポートの両方を、アクセスカプセル化が設定され、MAC アドレス学習が無効にされた 802.1Q トンネルに所属させることができます。エッジポートとコアポートの Dot1q トンネルは、スイッチモデル名の最後に「FX」またはそれ以降のサフィックスが付く、Cisco Nexus 9000 シリーズスイッチでサポートされます。

IGMP および MLD パケットは、802.1Q トンネルを介して転送できます。

このドキュメントで使用する用語は、Cisco Nexus 9000 シリーズのドキュメントとは異なっている場合があります。

表 1: 802.1Q トンネルの用語

ACI のドキュメント	Cisco Nexus 9000 シリーズのドキュメント
エッジポート	トンネルポート
コアポート	トランクポート

次の注意事項および制約事項が適用されます:

- VTP、CDP、LACP、LLDP、および STP プロトコルのレイヤ2 トンネリングは、次の制限付きでサポートされます。
 - リンク集約制御プロトコル (LACP) トンネリングは、個々のリーフインターフェイスを使用する、ポイントツーポイントトンネルでのみ、予想通りに機能します。ポートチャンネル (PC) または仮想ポートチャンネル (vPC) ではサポートされていません。
 - PC または vPC を持つ CDP および LLDP トンネリングは確定的ではありません。これは、トラフィックの宛先として選択するリンクによって異なります。
 - レイヤ2 プロトコル トンネリングに VTP を使用するには、CDP をトンネル上で有効にする必要があります。
 - レイヤ2 プロトコルのトンネリングが有効になっており、Dot1q トンネルのコアポートにブリッジドメインが展開されている場合、STP は 802.1Q トンネルブリッジドメインではサポートされません。
 - Cisco ACI リーフスイッチは、トンネルブリッジドメインのエンドポイントでフラッシングを行い、ブリッジドメインでフラッドイングすることにより、STP TCN パケットに反応します。

- 2 個上のインターフェイスを持つ CDP および LLDP トンネリングが、すべてのインターフェイスでパケットをフラッディングします。
- エッジポートからコアポートにトンネリングしているレイヤ 2 プロトコルパケットの宛先 MAC アドレスは、01-00-0c-cd-cd-d0 に書き換えられ、コアポートからエッジポートにトンネリングしているレイヤ 2 プロトコルパケットの宛先 MAC アドレスは、プロトコルに対して標準のデフォルト MAC アドレスに書き換えられます。
- PC または vPC が Dot1q Tunnel 内の唯一のインターフェイスであり、削除してから再設定した場合には、PC/VPC の Dot1q トンネルへの関連付けを削除して、再設定してください。
- 製品 ID に EX が含まれるスイッチに導入された 802.1Q トンネルでは、最初の 2 つの VLAN タグの 0x8100 + 0x8100、0x8100 + 0x88a8、0x88a8 + 0x88a8 の Ethertype の組み合わせはサポートされません。
トンネルが EX と FX またはそれ以降のスイッチの組み合わせに導入されている場合は、この制限が適用されます。
製品 ID に FX 以降が含まれるスイッチにのみトンネルが導入されている場合、この制限は適用されません。
- コアポートについては、二重タグつきフレームのイーサタイプは、0x8100 の後に 0x8100 が続く必要があります。
- 複数のエッジポートおよびコアポートを（リーフスイッチ上のものであっても）Dot1q トンネルに含めることができます。
- エッジポートは 1 つのトンネルの一部にのみ属することが可能ですが、コアポートは複数の Dot1q トンネルに属することができます。
- 通常の EPG を 802.1Q で使用されるコアポートに展開できます。
- L3Outs は、Dot1q トンネルで有効になっているインターフェイスではサポートされていません。
- FEX インターフェイスは Dot1q トンネルのメンバーとしてはサポートされていません。
- インターフェイスレベルの統計情報は Dot1q トンネルのインターフェイスでサポートされていますが、トンネルレベルの統計情報はサポートされていません。

GUI を使用した 802.1Q トンネルの設定

APIC GUI を使用した 802.1Q トンネル インターフェイスの設定

次の手順で、トンネルを使用するインターフェイスを設定します:

始める前に

トンネルを使用するテナントを作成します。

手順

-
- ステップ 1** メニューバーで、**[Fabric] > [Access Policies]** の順にクリックします。
- ステップ 2** [ナビゲーション] バーで、**[ポリシー] > [インターフェイス] > [L2 インターフェイス]** をクリックします。
- ステップ 3** **[L2 インターフェイス]** を右クリックし、**[L2 インターフェイス ポリシーの作成]** を選択して、次の操作を実行します。
- Name** フィールドに、レイヤ 2 インターフェイス ポリシーの名前を入力します。
 - オプション。ポリシーの説明を追加します。L2 インターフェイス ポリシーの目的を説明することをお勧めします。
 - Dot1q** トンネルで、エッジポートとして使用するインターフェイスを有効にするインターフェイス ポリシーを作成するために、**QinQ** フィールドで、**edgePort** をクリックします。
 - Dot1q** トンネルでコアポートとして使用するインターフェイスを有効にするインターフェイス ポリシーを作成するために、**QinQ** フィールドで、**corePort** をクリックします。
- ステップ 4** 次の手順で、L2 インターフェイス ポリシーをポリシー グループに適用します。
- [ファブリック] > [アクセス ポリシー] > [インターフェイス] > [リーフ インターフェイス]** をクリックして、**[ポリシー グループ]** を展開します。
 - [リーフ アクセス ポート]**、**[PC インターフェイス]** または **[VPC インターフェイス]** を右クリックし、トンネルに設定しているインターフェイスのタイプに応じて、次のいずれかを選択します。
 - **リーフ アクセス ポート ポリシー グループの作成**
 - **PC ポリシー グループの作成**
 - **VPC ポリシー グループの作成**
 - 表示されるダイアログボックスで、以下のアクションを実行します:
 - **Name** フィールドに、ポリシー グループの名前を入力します。
オプション。ポリシー グループについての説明を追加します。ポリシー グループの目的を説明することをお勧めします。
 - **L2 Interface Policy** フィールドで、下向き矢印をクリックし、前に作成した L2 インターフェイス ポリシーを選択します。
 - CDP レイヤ 2 トンネリング プロトコルでトンネルを作成する場合は、**[CDP Policy]** 下向き矢印をクリックし、ポリシー ダイアログボックスでポリシーの名前を追加し、管理状態を無効にして、**[Submit]** をクリックします。 .

- LLDP レイヤ2 トンネリングプロトコルでトンネルを作成する場合には、[LLDP Policy] 下向き矢印をクリックし、ポリシーダイアログボックスでポリシーの名前を追加し、送信状態を無効にして [submit] をクリックします。
- [Submit] をクリックします。

ステップ 5 次の手順に従ってリーフ インターフェイス プロファイルを作成します:

- a) **[Fabric] > [Access Policies] > [Interfaces] > [Leaf Interfaces] > [Profiles]** をクリックします。
- b) **Profiles** プロファイルを右クリックし、**Create Leaf Interface Profile** を選択し、次の手順に従います:
 - **Name** フィールドに、**Leaf Interface Profile** の名前を入力します。
オプション。説明を追加します。
 - **Interface Selectors** フィールドで、+ をクリックし、以下の情報を入力します:
 - **[名前]** フィールドに、インターフェイス セレクタの名前を入力します。
オプション。説明を追加します。
 - **Interface IDs** フィールドに、このトンネルに含まれる **Dot1q Tunnel** インターフェイス、または複数のインターフェイスの名前を入力します。
 - **Interface Policy Group** フィールドで、下向き矢印をクリックして、前に作成したインターフェイス ポリシー グループを選択します。

ステップ 6 トンネル設定のポートへのスタティック バインディングを作成するには、**[Tenant] > [Networking] > [Dot1Q Tunnels]** の順にクリックします。**[Dot1Q Tunnels]** を展開し、前に作成した **Dot1Q Tunnels <ポリシー名>** をクリックして、次の操作を実行します。

- a) **[Static Bindings]** テーブルを展開して **[Create Static Binding]** ダイアログボックスを開きます。
- b) **[Port]** フィールドで、ポートの種類を選択します。
- c) **[Node]** フィールドで、ドロップダウンリストからノードを選択します。
- d) **[Path]** フィールドで、ドロップダウンリストからインターフェイスパスを選択し、**[Submit]** をクリックします。

NX-OS スタイルの CLI を使用した 802.1Q トンネルの設定

NX-OS スタイル CLI を使用した 802.1Q トンネルの設定



- (注) **Dot1q** トンネルに含まれるインターフェイスのポート、ポートチャンネル、仮想ポートチャンネルを使用できます。手順の詳細にはポートの設定が含まれます。エッジおよびコアポートチャンネルと仮想ポートチャンネルを設定するコマンドについては、下の例を参照してください。

次の手順で、**Dot1q** トンネルを作成し、NX-OS スタイル CLI を使用してトンネルで使用するインターフェイスを設定します。



- (注) **Dot1q** トンネルには2個以上のインターフェイスを含める必要があります。手順を繰り返し（または2個のインターフェイスをまとめて設定）、**Dot1q** トンネルで使用する各インターフェイスをマークします。この例で、2個のインターフェイスは単一の顧客で使用されているエッジスイッチポートとして設定されます。

次の手順を使用して、設定を次の手順を使用して、NX-OS スタイル CLI を使用して **Dot1q** トンネルを設定します。

1. トンネルで使用するインターフェイスを最低2個設定します。
2. **Dot1q** トンネルを作成します。
3. トンネルとすべてのインターフェイスを関連付けます。

始める前に

Dot1q トンネルを使用するテナントを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	configure 例： apic1# 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	次の手順により 802.1Q で使用するための 2 個のインターフェイスを設定します。	

	コマンドまたはアクション	目的
ステップ 3	leaf ID 例 : apicl(config)# leaf 101	Dot1q トンネル のインターフェイスが配置されるリーフを特定します。
ステップ 4	interface ethernet slot/port 例 : apicl(config-leaf)# interface ethernet 1/13-14	トンネルのポートとしてマークされるインターフェイスを特定します。
ステップ 5	switchport mode dot1q-tunnel {edgePort corePort} 例 : apicl(config-leaf-if)# switchport mode dot1q-tunnel edgePort apicl(config-leaf-if)# exit apicl(config-leaf)# exit apicl(config)# exit	802.1Q トンネルで使用するインターフェイスをマークして、設定モードをそのままにします。 この例では、エッジポートを使用するためにいくつかのインターフェイス設定を示します。トンネルに複数のインターフェイスを設定するには、手順 3 ~ 5 を繰り返します。
ステップ 6	次の手順で 802.1q トンネルを作成します。	
ステップ 7	leaf ID 例 : apicl(config)# leaf 101	インターフェイスが配置されているリーフに戻ります。
ステップ 8	interface ethernet slot/port 例 : apicl(config-leaf)# interface ethernet 1/13-14	トンネルに含まれるインターフェイスに戻ります。
ステップ 9	switchport tenant tenant-namedot1q-tunnel tunnel-name 例 : apicl(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_edgetunnel apicl(config-leaf-if)# exit	トンネルにインターフェイスに関連付け、設定モードを終了します。
ステップ 10	トンネルとその他のインターフェイスを関連付けるには、ステップ 7 ~ 10 を繰り返します。	

例：NX-OS スタイル CLI でポートを使用する 802.1Q トンネルを設定する

例：NX-OS スタイル CLI でポートを使用する 802.1Q トンネルを設定する

この例では、2つのポートを **Dot1q** トンネルで使用されるエッジポートインターフェイスとしてマークし、さらに2つのポートをコアポートインターフェイスで使用されるものとしてマークし、トンネルを作成して、ポートをトンネルに関連付けます。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/13-14
apic1(config-leaf-if)# switchport mode dot1q-tunnel edgePort
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 1/10, 1/21
apic1(config-leaf-if)# switchport mode dot1q-tunnel corePort
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# tenant tenant64
apic1(config-tenant)# dot1q-tunnel vrf64_tunnel
apic1(config-tenant-tunnel)# l2protocol-tunnel cdp
apic1(config-tenant-tunnel)# l2protocol-tunnel lldp
apic1(config-tenant-tunnel)# access-encap 200
apic1(config-tenant-tunnel)# mac-learning disable
apic1(config-tenant-tunnel)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/13-14
apic1(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_tunnel
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 1/10, 1/21
apic1(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_tunnel
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

例：NX-OS スタイル CLI でポートチャンネルを使用する 802.1Q トンネルを設定する

例では、このエッジポート 802.1q インターフェイスとして2つのポートチャンネルにマークし、2つ以上のポートチャンネルをコアポート 802.1q インターフェイスとしてマークして、**Dotq** トンネルを作成し、トンネルとポートチャンネルを関連付けます。

```
apic1# configure
apic1(config)# tenant tenant64
apic1(config-tenant)# dot1q-tunnel vrf64_tunnel
apic1(config-tenant-tunnel)# l2protocol-tunnel cdp
apic1(config-tenant-tunnel)# l2protocol-tunnel lldp
apic1(config-tenant-tunnel)# access-encap 200
apic1(config-tenant-tunnel)# mac-learning disable
apic1(config-tenant-tunnel)# exit
```



```

apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# interface port-channel pc1
apicl(config-leaf-if)# exit
apicl(config-leaf)# interface ethernet 1/2-3
apicl(config-leaf-if)# channel-group pc1
apicl(config-leaf-if)# exit
apicl(config-leaf)# interface port-channel pc1
apicl(config-leaf-if)# switchport mode dot1q-tunnel edgePort
apicl(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_tunnel
apicl(config-tenant-tunnel)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 102
apicl(config-leaf)# interface port-channel pc2
apicl(config-leaf-if)# exit
apicl(config-leaf)# interface ethernet 1/4-5
apicl(config-leaf-if)# channel-group pc2
apicl(config-leaf-if)# exit
apicl(config-leaf)# interface port-channel pc2
apicl(config-leaf-if)# switchport mode dot1q-tunnel corePort
apicl(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_tunnel

```

例：NX-OS スタイル CLI で仮想ポート チャンネルを使用する 802.1Q トンネルを設定する

この例では、2つの仮想ポートチャンネル(vPC)をDot1qトンネルのエッジポート802.1Qインターフェイスとしてマークし、さらに2つのVPCをトンネルのためのコアポートインターフェイスとしてマークし、トンネルを作成して、仮想ポートチャンネルをトンネルに関連付けています。

```

apicl# configure
apicl(config)# vpc domain explicit 1 leaf 101 102
apicl(config)# vpc context leaf 101 102
apicl(config-vpc)# interface vpc vpc1
apicl(config-vpc-if)# switchport mode dot1q-tunnel edgePort
apicl(config-vpc-if)# exit
apicl(config-vpc)# exit
apicl(config)# vpc domain explicit 1 leaf 103 104
apicl(config)# vpc context leaf 103 104
apicl(config-vpc)# interface vpc vpc2
apicl(config-vpc-if)# switchport mode dot1q-tunnel corePort
apicl(config-vpc-if)# exit
apicl(config-vpc)# exit
apicl(config)# tenant tenant64
apicl(config-tenant)# dot1q-tunnel vrf64_tunnel
apicl(config-tenant-tunnel)# l2protocol-tunnel cdp
apicl(config-tenant-tunnel)# l2protocol-tunnel lldp
apicl(config-tenant-tunnel)# access-encap 200
apicl(config-tenant-tunnel)# mac-learning disable
apicl(config-tenant-tunnel)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 103
apicl(config-leaf)# interface ethernet 1/6
apicl(config-leaf-if)# channel-group vpc1 vpc
apicl(config-leaf-if)# exit
apicl(config-leaf)# exit
apicl(config)# leaf 104

```

```

apic1(config-leaf)# interface ethernet 1/6
apic1(config-leaf-if)# channel-group vpc1 vpc
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config-vpc)# interface vpc vpc1
apic1(config-vpc-if)# switchport tenant tenant64 dot1q-tunnel vrf64_tunnel
apic1(config-vpc-if)# exit

```

REST API を使用した 802.1Q トンネルの設定

REST API を使用してポートを持つトンネル 802.1 q の設定

作成、**Dot1q** トンネル、ポートを使用して、次の例などの手順でのインターフェイスを設定します。

始める前に

Dot1q トンネルを使用するテナントを設定します。

手順

ステップ 1 次の例のように XML で REST API を使用して **Dot1q** トンネルを作成します。

例では、LLDP レイヤ 2 トンネリング プロトコルでトンネルを設定し、アクセス カプセル化 VLAN を追加し、トンネルで MAC ラーニングを無効にします。

例：

```

<fvTnlEPg name="VRF64_dot1q_tunnel" qiqL2ProtTunMask="lldp" accEncap="vlan-10"
  fwdCtrl="mac-learn-disable" >
  <fvRsTnlpathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/13]"/>
</fvTnlEPg>

```

ステップ 2 次の例のように XML で静的にバインディングするレイヤ 2 インターフェイス ポリシーを設定します。

この例では、エッジスイッチ ポートにレイヤ 2 インターフェイス ポリシーを設定します。コアスイッチ ポートのポリシーを設定するには、**l2IfPol MO** で **edgePort** の代わりに **corePort** を使用します。

例：

```

<l2IfPol name="VRF64_L2_int_pol" qinq="edgePort" />

```

ステップ 3 次の例のように、XML でリーフ アクセス ポート ポリシー グループにレイヤ 2 インターフェイス ポリシーを適用します。

例：

```

<infraAccPortGrp name="VRF64_L2_Port_Pol_Group" >
  <infraRsL2IfPol tnL2IfPolName="VRF64_L2_int_pol"/>
</infraAccPortGrp>

```

ステップ 4 次の例のように、XML でインターフェイスセクタとともにリーフプロファイルを設定します。

例：

```
<infraAccPortP name="VRF64_dot1q_leaf_profile" >
  <infraHPortS name="vrf64_access_port_selector" type="range">
    <infraPortBlk name="block2" toPort="15" toCard="1" fromPort="13" fromCard="1"/>
    <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-VRF64_L2_Port_Pol_Group"
  />
</infraHPortS>
</infraAccPortP>
```

例

次の例のように、2 個の POST 上のエッジ ポートのポート設定を示します。

Post 1 の XML :

```
<polUni>
  <infraInfra>
    <l2IfPol name="testL2IfPol" qinq="edgePort"/>
    <infraNodeP name="Node_101_phys">
      <infraLeafS name="phys101" type="range">
        <infraNodeBlk name="test" from_="101" to_="101"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-phys21"/>
    </infraNodeP>
    <infraAccPortP name="phys21">
      <infraHPortS name="physHPortS" type="range">
        <infraPortBlk name="phys21" fromCard="1" toCard="1" fromPort="21" toPort="21"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-21"/>
      </infraHPortS>
    </infraAccPortP>
    <infraFuncP>
      <infraAccPortGrp name="21">
        <infraRsL2IfPol tnL2IfPolName="testL2IfPol"/>
        <infraRsAttEntP tDn="uni/infra/attentp-AttEntityProf1701"/>
      </infraAccPortGrp>
    </infraFuncP>
    <l2IfPol name='testL2IfPol' qinq='edgePort' />
    <infraAttEntityP name="AttEntityProf1701">
      <infraRsDomP tDn="uni/phys-dom1701"/>
    </infraAttEntityP>
  </infraInfra>
</polUni>
```

Post 2 の XML :

```
<polUni>
  <fvTenant dn="uni/tn-Coke" name="Coke">
    <fvTnLEPg name="WEB5" qiqL2ProtTunMask="lldp" accEncap="vlan-10"
    fwdCtrl="mac-learn-disable" >
      <fvRsTnLpathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/21]"/>
    </fvTnLEPg>
  </fvTenant>
</polUni>
```

REST API を使用した PC を持つトンネル 802.1Q の設定

PC を使用して **Dot1q** トンネル を作成して、次の例のような手順でインターフェイスを設定します。

始める前に

Dot1q トンネル を使用するテナントを設定します。

手順

ステップ 1 次の例のように XML で REST API を使用して **Dot1q** トンネル を作成します。

例では、LLDP レイヤ 2 トンネリング プロトコルでトンネルを設定し、アクセス カプセル化 VLAN を追加し、トンネルで MAC ラーニングを無効にします。

例：

```
<fvTnlEPg name="WEB" qiqL2ProtTunMask=lldp accEncap="vlan-10"
fwdCtrl="mac-learn-disable" >
  <fvRsTnlpathAtt tDn="topology/pod-1/paths-101/pathep-[po2]"/>
</fvTnlEPg>
```

ステップ 2 次の例のように XML で静的にバインディングするレイヤ 2 インターフェイス ポリシーを設定します。

この例では、エッジスイッチ ポートにレイヤ 2 インターフェイス ポリシーを設定します。コア スイッチ ポートのレイヤ 2 インターフェイス ポリシーを設定するには、`l2IfPol MO` の `edgePort` 代わりに `corePort` を使用します。

例：

```
<l2IfPol name="testL2IfPol" qinq="edgePort"/>
```

ステップ 3 次の例のように、XML で PC インターフェイス ポリシー グループにレイヤ 2 インターフェイス ポリシーを適用します。

例：

```
<infraAccBndlGrp name="po2" lagT="link">
  <infraRsL2IfPol tnL2IfPolName="testL2IfPol"/>
</infraAccBndlGrp>
```

ステップ 4 次の例のように、XML でインターフェイス セレクタを持つリーフ プロファイルを設定します。

例：

```
<infraAccPortP name="PC">
  <infraHPortS name="allow" type="range">
    <infraPortBlk name="block2" fromCard="1" toCard="1" fromPort="10" toPort="11" />
  <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-po2"/>
</infraHPortS>
</infraAccPortP>
```

例

次の例では、2 個のポストの PC 設定を示します。

この例では、エッジポートとして PC ポートを設定します。コアポートとして設定を使用するには、Post 1 にある l2IfPol MO の edgePort 代わりに corePort を使用します。

Post 1 の XML :

```
<infraInfra dn="uni/infra">
  <infraNodeP name="bLeaf3">
    <infraLeafS name="leafs3" type="range">
      <infraNodeBlk name="nblk3" from_="101" to_="101">
      </infraNodeBlk>
    </infraLeafS>
    <infraRsAccPortP tDn="uni/infra/accportprof-shipping3"/>
  </infraNodeP>
  <infraAccPortP name="shipping3">
  <infraHPortS name="pselc3" type="range">
    <infraPortBlk name="blk3" fromCard="1" toCard="1" fromPort="24" toPort="25"/>

    <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-accountingLag3" />
  </infraHPortS>
</infraAccPortP>
<infraFuncP>
  <infraAccBndlGrp name="accountingLag3" lagT='link'>
    <infraRsAttEntP tDn="uni/infra/attentp-default"/>
    <infraRsLacpPol tnLacpLagPolName='accountingLacp3' />
    <infraRsL2IfPol tnL2IfPolName="testL2IfPol3" />
  </infraAccBndlGrp>
</infraFuncP>
<lacpLagPol name='accountingLacp3' ctrl='15' descr='accounting' maxLinks='14' minLinks='1'
mode='active' />
<l2IfPol name='testL2IfPol3' qinq='edgePort' />
<infraAttEntityP name="default">
  </infraAttEntityP>
</infraInfra>
```

Post 2 の XML :

```
<polUni>
  <fvTenant dn="uni/tn-Coke" name="Coke">
    <!-- bridge domain -->
    <fvTnlEPg name="WEB6" qiql2ProtTunMask="lldp" accEncap="vlan-10"
fwdCtrl="mac-learn-disable" >
      <fvRsTnlpathAtt tDn="topology/pod-1/paths-101/pathep-[accountingLag1]"/>
    </fvTnlEPg>
  </fvTenant>
</polUni>
```

REST API を使用した vPC での 802.1 Q トンネルの設定

vPC を使用して Dot1q トンネルを作成し、次の例のような手順でインターフェイスを設定します。

始める前に

Dot1q トンネル を使用するテナントを設定します。

手順

ステップ 1 次の例のように、XML とともに REST API を使用して 802.1 q トンネルを作成します。

例ではレイヤ 2 トンネル プロトコルとともにトンネルを設定し、アクセス カプセル化 VLAN を追加し、トンネルで MAC 学習を無効にします。

例 :

```
<fvTnlEPg name="WEB" qiql2ProtTunMask=lldp accEncap="vlan-10" fwdCtrl="mac-learn-disable"
>
    <fvRsTnlpathAtt tDn="topology/pod-1/protopaths-101-102/pathep-[po4]" />
</fvTnlEPg>
```

ステップ 2 次の例のように、XML とともに静的バインディングでレイヤ 2 インターフェイス ポリシーを設定します。

この例では、エッジスイッチ ポートのレイヤ 2 インターフェイス ポリシーを設定します。コアスイッチ ポートのレイヤ 2 インターフェイス ポリシーを設定するには、qinq ="corePort" ポートタイプを使用します。

例 :

```
<l2IfPol name="testL2IfPol" qinq="edgePort"/>
```

ステップ 3 次の例のように、XML を持つ VPC インターフェイス ポリシー グループにレイヤ 2 インターフェイス ポリシーを適用します。

例 :

```
<infraAccBndlGrp name="po4" lagT="node">
    <infraRsL2IfPol tnL2IfPolName="testL2IfPol"/>
</infraAccBndlGrp>
```

ステップ 4 次の例のように、XML を持つインターフェイス セクタでリーフ プロファイルを設定します。

例 :

```
<infraAccPortP name="VPC">
    <infraHPortS name="allow" type="range">
        <infraPortBlk name="block2" fromCard="1" toCard="1" fromPort="10" toPort="11" />
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-po4"/>
    </infraHPortS>
</infraAccPortP>
```

例

次の例は、3つのポストでのvPC設定を示しています。

この例では、vPC ポートをエッジポートとして設定します。コアポートとして設定するには、ポスト 2 で corePort を edgePort の代わりに l2IfPol MO で使用します。

Post 1 の XML :

```

<polUni>
  <fabricInst>
    <fabricProtPol pairT="explicit">
      <fabricExplicitGEp name="101-102-vpc1" id="30">
        <fabricNodePEp id="101"/>
        <fabricNodePEp id="102"/>
      </fabricExplicitGEp>
    </fabricProtPol>
  </fabricInst>
</polUni>

```

Post 2 の XML :

```

<infraInfra dn="uni/infra">
  <infraNodeP name="bLeaf1">
    <infraLeafS name="leafs" type="range">
      <infraNodeBlk name="nblk" from_"101" to_"101">
        </infraNodeBlk>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-shipping1"/>
    </infraNodeP>

    <infraNodeP name="bLeaf2">
      <infraLeafS name="leafs" type="range">
        <infraNodeBlk name="nblk" from_"102" to_"102">
          </infraNodeBlk>
        </infraLeafS>
        <infraRsAccPortP tDn="uni/infra/accportprof-shipping2"/>
      </infraNodeP>

    <infraAccPortP name="shipping1">
      <infraHPortS name="pselc" type="range">
        <infraPortBlk name="blk" fromCard="1" toCard="1" fromPort="4" toPort="4"/>

        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-accountingLag1" />
      </infraHPortS>
    </infraAccPortP>

    <infraAccPortP name="shipping2">
      <infraHPortS name="pselc" type="range">
        <infraPortBlk name="blk" fromCard="1" toCard="1" fromPort="2" toPort="2"/>

        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-accountingLag2" />
      </infraHPortS>
    </infraAccPortP>

    <infraFuncP>
      <infraAccBndlGrp name="accountingLag1" lagT='node'>
        <infraRsAttEntP tDn="uni/infra/attentp-default"/>
        <infraRsLacpPol tnLacpLagPolName='accountingLacp1' />
        <infraRsL2IfPol tnL2IfPolName="testL2IfPol" />
      </infraAccBndlGrp>
      <infraAccBndlGrp name="accountingLag2" lagT='node'>
        <infraRsAttEntP tDn="uni/infra/attentp-default"/>
        <infraRsLacpPol tnLacpLagPolName='accountingLacp1' />
        <infraRsL2IfPol tnL2IfPolName="testL2IfPol" />
      </infraAccBndlGrp>
    </infraFuncP>
    <lacpLagPol name='accountingLacp1' ctrl='15' descr='accounting' maxLinks='14' minLinks='1'
    mode='active' />
    <l2IfPol name='testL2IfPol' qinq='edgePort' />

    <infraAttEntityP name="default">
      </infraAttEntityP>
  </infraInfra>

```

Post 3 の XML :

```
<polUni>
  <fvTenant dn="uni/tn-Coke" name="Coke">
    <!-- bridge domain -->
    <fvTnlEPg name="WEB6" qiqL2ProtTunMask="lldp" accEncap="vlan-10"
    fwdCtrl="mac-learn-disable" >
      <fvRsTnlpathAtt tDn="topology/pod-1/protopaths-101-102/pathep-[accountingLag2]"/>

    </fvTnlEPg>
  </fvTenant>
</polUni>
```


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。