レイヤ 2 外部接続の設定

レイヤ 2 外部接続の設定

レイヤ2外部接続とは、ACIリーフスイッチ（別名境界リーフ）と外部ルータ間のスイッチングネットワークのことを表します。外部L2ネットワークを表すVLANはファブリック内のブリッジドメインの1つにマッピングされ、ブリッジドメインにレイヤ2拡張を提供し、ブリッジドメインを使用するEPGを外部ネットワークと通信できるようにします。外部ネットワークはEPGにマッピングされるため、ノード全体でさまざまな内部アプリケーションとさまざまなL2外部VLAN間のコントラクトを認識するのに役立ちます。

レイヤ 2 外部接続の設定

注意 APIC でインターフェイスごとの設定を行う際に、GUI と CLI を混在させないでください。GUI で行われた設定が、NX-OS CLI では部分的にしか機能しない可能性があります。

たとえば、GUI の **[Tenants]** > ***[tenant-name]*** > **[Application Profiles]** > ***[application-profile-name]*** > **[Application EPGs]** > ***[EPG-name]*** > **[Static Ports]** > **[Deploy Static EPG on PC, VPC, or Interface]** でスイッチ ポートを設定したと仮定します。

次に NX-OS スタイルの CLI で show running-config コマンドを使用すると、以下のような出力を受信します。

```
leaf 102
interface ethernet 1/15
switchport trunk allowed vlan 201 tenant t1 application ap1 epg ep1
 exit
 exit
```

NX-OS スタイルの CLI でこれらのコマンドを使用してスタティック ポートを設定すると、次のエラーが発生します。

```
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 1/15
apic1(config-leaf-if)# switchport trunk allowed vlan 201 tenant t1 application ap1 epg
ep1
No vlan-domain associated to node 102 interface ethernet1/15 encap vlan-201
```

これは、CLI に APIC GUI では実行されない検証があることが原因です。show running-config コマンドによって出力されたコマンドが NX-OS CLI で機能するためには、VLAN ドメインが事前に設定されている必要があります。設定の順序は GUI に適用されません。

レイヤ 2 外部接続の設定はスタティック アプリケーション EPG と似ていて、ノード、ポートの VLAN を EPG にマッピングし、EPG をブリッジドメインにマッピングして、コントラクトを提供または使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	設定モードにアクセスします。 **例**： `apic1# configure`	
ステップ 2	テナント設定モードを開始します。 **例**： `apic1(config)# tenant exampleCorp`	
ステップ 3	**[no] external-l2 epg** *epg-name* **例**： `apic1(config-tenant)# external-l2 epg extendBD1`	外部レイヤ 2 EPG を作成（または削除）します。

レイヤ 2 外部接続の設定

	コマンドまたはアクション	目的
ステップ 4	EPG にブリッジ ドメインを割り当てます. 例： `apic1(config-tenant-extl2epg)# bridge-domain member bd1`	
ステップ 5	テナント設定モードに戻ります。 例： `apic1(config-tenant-extl2epg)# exit`	
ステップ 6	グローバル コンフィギュレーション モードに戻ります。 例： `apic1(config-tenant)# exit`	
ステップ 7	設定するリーフを指定します。 例： `apic1(config)# leaf 101`	
ステップ 8	外部 EPG のポートを指定します。 例： `apic1(config-leaf)# interface eth 1/2`	
ステップ 9	デフォルトでは、ポートはレイヤ 2 トランク モードです。ポートがレイヤ 3 モードである場合、このコマンドを使用してレイヤ 2 トランク モードに変換する必要があります。 例： `apic1(config-leaf-if)# switchport`	
ステップ 10	インターフェイスを VLAN ドメインに関連付けます。 例： `apic1(config-leaf-if)# vlan-domain member dom1`	
ステップ 11	Assigns a VLAN on the leaf port and maps the VLAN to a layer 2 external EPG, with the **switchport trunk allowed vlan** *vlan-id* **tenant** *tenant-name* **external-l2 epg** *epg-name* command. 例：	（注） インターフェイスが VLAN ドメインと関連付けられている必要があり、そうでない場合このコマンドは拒否されます。

レイヤ 2 外部接続の設定

	コマンドまたはアクション	目的
	`apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp external-l2 epg extendBD1`	
ステップ 12	リーフポートに vlan を割り当て、**switchport** {**trunk allowed** | を使用して vlan を外部 svi にマッピングします。**trunk native** |**vlan tenant external-svi** } vlan idテナント名コマンド。 **access** **例**： `apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp external-svi`	（注） インターフェイスがVLANドメインと関連付けられている必要があり、そうでない場合このコマンドは拒否されます。

例

次に、外部接続用にレイヤ 2 ポートを導入する例を示します。

```
apic1# configure
apic1(config)# tenant exampleCorp
apic1(config-tenant)# external-l2 epg extendBD1
apic1(config-tenant-extl2epg)# bridge-domain member bd1
apic1(config-tenant-extl2epg)# exit
apic1(config-tenant)# exit

apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/2
apic1(config-leaf-if)# switchport
apic1(config-leaf-if)# switchport mode trunk
apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp external-l2
epg extendBD1
```

次に、外部接続用にレイヤ 2 ポート チャネルまたは vPC を導入する例を示します。

```
...

apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp external-l2
epg extendBD1
```

次に、外部接続用にレイヤ 2 インターフェイスで SVI を設定する例を示します。

```
apic1(config)# leaf 101

pic1(config-leaf)# interface ethernet 1/5
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp external-svi
apic1(config-leaf-if)# no switchport trunk allowed vlan 10 tenant exampleCorp external-svi

apic1(config-leaf)# interface ethernet 1/37
```

```
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport access vlan 11 tenant exampleCorp external-svi
apic1(config-leaf-if)# no switchport access vlan 11 tenant exampleCorp external-svi

apic1(config-leaf)# interface port-channel po34
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk native vlan 12 tenant exampleCorp external-svi
apic1(config-leaf-if)# no switchport trunk native vlan 12 tenant exampleCorp external-svi
```

VLAN ドメインの設定

VLAN ドメインについて

ACI ファブリックは、4K VLAN のグループに分割することにより、複数のテナントで使用できる、ファブリック全体の大規模なレイヤ 2 ドメインを利用可能にします。VLAN ドメインはノードおよびポートのグループ上で設定できる一連の VLAN を表します。VLAN ドメインは、複数のテナントを共有し、ノード、ポートおよび VLAN などの共通のリソースを個別に管理できるようになりました。テナントは 1 つ以上の VLAN ドメインにアクセスできます。VLAN プールの詳細については、「*Cisco Application* セントリック インフラストラクチャ基礎」の「*ACI* ポリシー モード*l*」章の「エンドポイント グループ」を参照してください。

これらの VLAN ドメインは、スタティックまたはダイナミックに設定できます。スタティック VLAN ドメインは、スタティック VLAN プールをサポートしますが、ダイナミック VLAN ドメインは、スタティックとダイナミックの両方の VLAN プールをサポートできます。スタティック プールの VLAN は、ユーザによって管理され、ベア メタル ホストへの接続などのアプリケーションに使用されます。ダイナミック プールの VLAN は、ユーザの介入なしに APIC によって割り当てかつ管理され、VMM などのアプリケーションに使用されます。VLAN ドメインおよびドメイン内の VLAN プールのデフォルト タイプはスタティックです。

ファブリック管理者は、テナントが L2/L3 設定のファブリック リソースの使用を開始する前に、次のステップを実行します。

1. VLAN ドメインを作成し、各 VLAN ドメイン内の VLAN を割り当てます。
2. 1 つ以上の VLAN ドメインにリーフ スイッチの外部向けポートを割り当てます。
3. Convert a port to L2/L3 by using the **[no] switchport** command. ポートのデフォルト状態はトランク モードでは、L2（スイッチポート）です。
4. L2 ポートの場合、グローバルまたはローカル ポートになるポート上の VLAN の範囲を設定します。デフォルトは global です。

ファブリック管理者は、VLAN ドメインがテナントに割り当てられ、テナント アプリケーションによって使用された後でも、これらの手順の設定を更新することができます。

スパニング ツリー および VLAN ドメインに関する注意事項

スパニングツリーに ACI ファブリックが参加していない場合でも、アクセス ポリシー設定に基づきスパニング ツリー ドメインをパーティション化できます。ACI は、スパニングツリー ドメインを判断するのにブリッジ ドメインまたはその設定に依存しません。代わりに、VLAN プールが EPG ドメインに割り当てられている場合、リーフ スイッチは同じ VLAN のカプセル化内の BPDU をフラッディングします。EPG ドメインに割り当てられている VLAN プールは、最終的にスパニングツリー ドメインとして機能します。

異なる VLAN プールに関連付けられた複数の EPG ドメインを使用する場合、すべて同じ VLAN ID を使用していても、BPDU がエンドポイント プロパティ全体でフラッディングすることを許可しません。EPG ドメインのタイプ（物理またはレイヤ 2 の外部）では、この動作は変更されません。

ACI ファブリックはスパニング ツリー ドメイン内のすべてのデバイスからのすべての BPDU をフラッディングするため、これにより各インターフェイスの MAC アドレスなど、外部デバイスの BPDU 情報を確認する動作をトリガする可能性があります。アクティブになる機能の例は、IOS デバイスの「スパニング ツリー EtherChannel の不正な設定ガード」です。これらの機能は、レイヤ 2 トンネルとして ACI を利用する際に考慮する必要があります。

（注） ポート VLAN ごとの機能 (localPort 範囲に同じ VLAN ID を使用してリーフ スイッチで複数の EPG を設定する) で設定されたインターフェイスでは、マルチ スパニング ツリー (MST) はサポートされません。

基本的な VLAN ドメイン設定

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`	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	**[no] vlan-domain** *domain-name* **[dynamic]** 例： `apic1(config)# vlan-domain dom2 dynamic`	VLAN ドメインを作成するか、既存のドメインを編集します。Include the **dynamic** keyword to create a dynamic VLAN pool. デフォルトは static です。
ステップ 3	**[no] vlan** *range* **[dynamic]** 例： `apic1(config-vlan)# vlan 1000-1999,4001`	VLAN ドメインに VLAN の範囲またはカンマ区切りのリストを割り当てます。 VLAN は、スタティックまたはダイナミックのいずれかになります。スタティック VLAN は、たとえば、ホスト

	コマンドまたはアクション	目的
		から外部のスイッチド ネットワークへの接続を提供するなど、ユーザによって設定されますが、ダイナミック レンジの VLAN は、VMM または L4 ～ L7 サービスなどの APIC アプリケーションによって内部的に設定されます。デフォルト タイプはスタティックです。 （注） スタティック ドメインは、ダイナミック VLAN を含めることはできません。 特定のポートの VLAN は、1 つの VLAN ドメインのみにマッピングできます。これは、設定時に適用されます。

例

この例では、基本的な VLAN ドメインを設定する方法を示します。

```
apic1# configure
apic1(config)# vlan-domain dom1
apic1(config-vlan)# vlan 1000-1999,4001
apic1(config-vlan)# exit
apic1(config)# vlan-domain dom2 dynamic
apic1(config-vlan)# vlan 101-200
apic1(config-vlan)# vlan 301-400 dynamic
```

高度な VLAN ドメイン設定

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`	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	**[no] vlan-domain** *domain-name* **[dynamic]** [**type** {**phys** | **l2ext** | **l3ext**}] 例：	VLAN ドメインを作成するか、既存のドメインを編集します。Include the **dynamic** keyword to create a dynamic VLAN pool. デフォルトは static です。

	コマンドまたはアクション	目的
	`apic1(config)# vlan-domain dom1 type phys`	The **type** option is visible and mandatory if one or more of the following conditions exist: • 3 つのすべての VLAN ドメインのタイプが、このドメイン名に存在しない場合 • 3 つの VLAN ドメインのタイプが、異なる VLAN プールを持つ場合 • 3 つの VLAN ドメインのタイプが、同じ VLAN プールを共有するが、プール名は VLAN ドメインの名前と異なる場合
ステップ 3	**[no] vlan-pool** *vlan-pool-name* 例： `apic1(config-leaf)# vlan-pool myVlanPool3`	VLAN プールを作成します。This command is available only when the **type** option is present in the **vlan-domain** command. You must declare the VLAN pool before adding VLANs with the **vlan** command.
ステップ 4	**[no] vlan** *range* **[dynamic]** 例： `apic1(config-vlan-domain)# vlan 1000-1999,4001`	VLAN ドメインに VLAN の範囲またはカンマ区切りのリストを割り当てます。 VLAN は、スタティックまたはダイナミックのいずれかになります。スタティック VLAN は、たとえば、ホストから外部のスイッチド ネットワークへの接続を提供するなど、ユーザによって設定されますが、ダイナミック レンジの VLAN は、VMM または L4 ～ L7 サービスなどの APIC アプリケーションによって内部的に設定されます。デフォルト タイプはスタティックです。 （注） スタティック ドメインは、ダイナミック VLAN を含めることはできません。 特定のポートの VLAN は、1 つの VLAN ドメインのみにマッピングできます。これは、設定時に適用されます。

	コマンドまたはアクション	目的
ステップ 5	**show vlan-domain** [**name** *domain-name*] [**vlan** *vlan-id*] [**leaf** *leaf-id*] 例： `apic1(config-vlan-domain)# show vlan-domain name dom1 vlan 1002 leaf 102`	アプリケーション EPG、サブインターフェイス、外部 SVI および外部 L2 などのアプリケーションの VLAN ドメインの使用状況を表示します。

例

次の例では、VLAN プールを使用する VLAN ドメインを設定する方法を示します。

```
apic1# configure
(config)# vlan-domain dom1 type phys
(config-vlan-domain)# vlan-pool myVlanPool3
(config-vlan-domain)# vlan 1000-1999, 4001
```

ポートへの VLAN ドメインの関連付け

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`	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-id1-node-id2* 例： `apic1(config)# leaf 101-102`	設定するリーフのペアを指定します。
ステップ 3	**interface** *type* 例： `apic1(config-leaf)# int eth 1/1-24`	VLAN ドメインに関連付けられたポートまたはポートの範囲を指定します。
ステップ 4	**[no] vlan-domain member** *domain-name* 例： `apic1(config-leaf-if)# vlan-domain member dom1`	VLAN ドメインに指定したポートを割り当てます。
ステップ 5	**[no] switchport** 例： `apic1(config-leaf-if)# switchport`	デフォルトでは、ポートはレイヤ 2 トランク モードです。ポートがレイヤ 3 モードである場合、このコマンドを使用し

	コマンドまたはアクション	目的
		て、レイヤ 2 トランク モードに変換する必要があります。
ステップ 6	（任意） **[no] switchport vlan scope local** 例： `apic1(config-leaf-if)# switchport vlan scope local`	デフォルトでは、VLANの範囲はノードに対してグローバルです。1 つの VLAN は、ノードの 1 つだけの EPG にマッピングできます。VLANの範囲がポートに対してローカルの場合、VLAN からの EPG へのマッピングは、同じノード上の別のポートでは異なります。 To return the scope to global, use the **no** command prefix.
ステップ 7	**show vlan-domain** [**name** *domain-name*] [**vlan** *vlan-id*] [**leaf** *leaf-id*] 例： `apic1(config-leaf-if)# show vlan-domain name dom1 vlan 1002 leaf 102`	アプリケーション EPG、外部 SVI および外部 L2 などのアプリケーションの VLAN ドメインの使用状況を表示します。

例

この例では、ポートに VLAN ドメインを関連付ける方法を示します。

```
apic1# configure
(config) # leaf 101-102
(config-leaf) # int eth 1/1-24
(config-leaf-if) # vlan-domain member dom1

(config-leaf) # int eth 1/1-12
(config-leaf-if) # no switchport
(config-leaf) # int eth 1/13-24
(config-leaf-if) # switchport

(config) # leaf 101-102
(config-leaf) # int eth 1/1-12
(config-leaf-if) # switchport vlan scope local

(config-leaf) # int eth 1/13
(config-leaf-if) # no switchport vlan scope local
```

ポート チャネルへの VLAN ドメインの関連付け

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`	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-id1-node-id2* 例： `apic1(config)# leaf 101-102`	設定するリーフのペアを指定します。
ステップ 3	**interface port-channel** *port-channel-name* 例： `apic1(config-leaf)# int port-channel pc1`	VLAN ドメインに関連付けられるポート チャネルを指定します。
ステップ 4	**[no] vlan-domain member** *domain-name* 例： `apic1(config-leaf-if)# vlan-domain member dom1`	VLAN ドメインに指定したポート チャネルを割り当てます。

例

```
apic1# configure
apic1(config)# leaf 101-102
apic1(config-leaf)# int port-channel pc1
apic1(config-leaf-if)# vlan-domain member dom1
```

テンプレート ポリシー グループへの VLAN ドメインの関連付け

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`	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	**template policy-group** *policy-group-name* 例：	設定するテンプレート ポリシー グループを指定します。

	コマンドまたはアクション	目的
	apic1(config)# **template policy-group myPolGp5**	
ステップ 3	**[no] vlan-domain member** *domain-name* 例： apic1(config-pol-grp-if)# **vlan-domain member dom1**	VLAN ドメインに指定したテンプレート ポリシー グループを割り当てます。

例

```
apic1# configure
apic1(config)# template policy-group myPolGp5
apic1(config-pol-grp-if)# vlan-domain member dom1
```

テンプレート ポート チャネルへの VLAN ドメインの関連付け

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**	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	**template port-channel** *policy-group-name* 例： apic1(config)# **template port-channel myPC7**	設定するテンプレート ポート チャネルを指定します。
ステップ 3	**[no] vlan-domain member** *domain-name* 例： apic1(config-if)# **vlan-domain member dom1**	VLAN ドメインに指定したテンプレート ポート チャネルを割り当てます。

例

```
apic1# configure
apic1(config)# template port-channel myPC7
apic1(config-po-ch-if)# vlan-domain member dom1
```

バーチャル ポート チャネルへの VLAN ドメインの関連付け

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`	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	**vpccontext fex** id1 node-id2 [fex-id1 fex-id2] **leaf** 例： `apic1(config)# vpc context leaf 101 102`	設定する VPC およびリーフを指定します。
ステップ 3	**interface vpc** *vpc-name* [**fex** *fex-id1 fex-id2*] 例： `apic1(config-vpc)# int vpc vpc1`	VLAN ドメインに関連付けられるポート チャネルを指定します。
ステップ 4	**[no] vlan-domain member** *domain-name* 例： `apic1(config-vpc-if)# vlan-domain member dom1`	VLAN ドメインに指定した VPC を割り当てます。

例

```
apic1# configure
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# int vpc vpc1
apic1(config-vpc-if)# vlan-domain member dom1
```

EPG の Q-in-Q カプセル化マッピングの設定

EPG の Q-in-Q カプセル化マッピング

Cisco APIC を使用して、通常のインターフェイスまたは VPC で二重タグ付き VLAN トラフィック入力を EPG にマッピングできます。この機能が有効になっている場合、二重タグ付きトラフィックが EPG のネットワークに入るとき、両方のタグがファブリックで個別に処理され、ACI スイッチを入力するとき二重タグに復元されます。単一タグおよびタグなしのトラフィックの入力はドロップします。

この機能は、Nexus 9300-FX プラットフォーム スイッチでのみサポートされています。

外側と内側の両方のタグは、EtherType 0x8100 である必要があります。

MAC ラーニングおよびルーティングは、カプセル化のアクセスではなく、EPG ポート、sclass、VRF に基づいています。

QoS 優先度設定がサポートされ、入力の外側のタグから派生し、出力の両方のタグに書き換えられます。

EPG はリーフ スイッチの他のインターフェイスに同時に関連付けることができ、単一タグの VLAN に設定されます。

サービス グラフは、Q-in-Q カプセル化たインターフェイスにマッピングされているプロバイダとコンシューマ EPG をサポートしています。サービス ノードの入力および出力トラフィックが単一タグのカプセル化フレームにある限り、サービス グラフを挿入することができます。

この機能では、次の機能とオプションがサポートされていません。

- ポートごとの VLAN 機能
- FEX 接続
- 混合モードはサポートされていません。たとえば、Q-in-Q カプセル化モードのインターフェイスでは、通常の VLAN のカプセル化ではなく、二重タグ付きカプセルのみを持つ EPG にバインディングされている静的パスを有します。
- STP および「カプセル化でフラッディング」オプション
- タグなしおよび 802.1p モード
- マルチポッドと複数サイト
- レガシ ブリッジ ドメイン
- L2Out および L3Out 接続
- VMM の統合
- Q-in-Q カプセル化モードにルーティングされるポート モードの変更はサポートされていません
- Q-in-Q カプセル化モードのポートと通常のトランク モードのポート間では、各 Vlan MCP はサポートされていません。
- VPC ポートが Q-in-Q カプセル化モードを有効にしている場合、VLAN の整合性チェックは行われません。

NX-OS スタイル CLI を使用した Q-in-Q カプセル化リーフ インターフェイスへの EPG のマッピング

Q-in-Q カプセル化のインターフェイスを有効にし、EPG にインターフェイスを関連付けます。

始める前に

Q-in-Q モードに設定されているインターフェイスでマッピングされるテナント、アプリケーション プロファイル、アプリケーション EPG 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	**Configure** 例： `apic1# configure`	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** *number* 例： apic1(config)# leaf 101	設定するリーフを指定します。
ステップ 3	**interface ethernet***slot/port* 例： `apic1 (config-leaf)# interface ethernet 1/25`	設定するインターフェイスを指定します。
ステップ 4	**switchport mode dot1q-tunnel** doubleQtagPort 例： `apic1(config-leaf-if)# switchport mode dot1q-tunnel doubleQtagPort`	Q-in-Q カプセル化のインターフェイスを有効にします。
ステップ 5	**switchport trunkqinq outer-vlan***vlan-number* **inner-vlan** *vlan-number* **tenant** *tenant-name* **application** *application-name* **epg** *epg-name* 例： `apic1(config-leaf-if)# switchport trunk qinq outer-vlan 202 inner-vlan 203 tenant tenant64 application AP64 epg EPG64`	インターフェイスを EPG に関連付けます。

例

次の例では、リーフ インターフェイス 101/1/25 で Q-in-Q カプセル化を有効にして（VLAN ID 201 外部および VLAN ID 203 内部）、EPG64 にインターフェイスを関連付けます。

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/25
apic1(config-leaf-if)#switchport mode dot1q-tunnel doubleQtagPort
apic1(config-leaf-if)# switchport trunk qinq outer-vlan 202 inner-vlan 203 tenant tenant64
 application AP64 epg EPG64
```

ACI ファブリックでイーサネット トラフィックを介したファイバ チャネルのサポート

ACI ファブリック上のイーサネット トラフィックによるファイバ チャネルをサポートする

ACI を使用すると、設定し、(FCoE) トラフィック ACI ファブリックの Fibre Channel over Ethernet のサポートを管理できます。

FCoE は、ファイバ チャネル SAN からイーサネット ネットワークにシームレスに移動するストレージ トラフィックを有効になり、イーサネット パケット内のファイバ チャネル (FC) パケットをカプセル化するプロトコルです。

FCoE プロトコルのサポート ACI ファブリックでの一般的な実装には、FC ネットワーク上にある SAN ストレージ デバイスとの通信に ACI ファブリックのイーサネット ベースの上に存在するホストが有効になります。ホストは、ACI Leaf スイッチに導入仮想の F ポートを介して接続しています。仮想 F ポートが同じ ACI Leaf スイッチに導入する仮想 NP ポートを介して ACI ファブリックへの FCF ブリッジでは、SAN ストレージ デバイスおよび FC ネットワークが接続されています。仮想 NP ポートおよび仮想 F ポートも汎用的に仮想ファイバ チャネル (vFC) ポートに呼ばれます。

（注） リリース バージョン 2.0(1)、においては、FCoE サポートはハードウェアです N9K-C93180YC-EX と N9K-C93108TC-EX に制限されます。リリース バージョン 2.2(1) と N9K-C93180LC-EX 40 ギガビット イーサネット (GE) ポートは F または NP ポートとして使用できます。ただし、FCoE を許可されている場合は、40GE ポート ブレイク アウトを対応ことはできません。FCoE はブレイクアウト ポートではサポートされていません。

リリース バージョン 2.2(2) からは、N9K-C93180YC-FX および N9K-C93108TC-FX ハードウェアは、FCoE をサポートします。リリース 2.3(1) では、ハードウェア N9K-C93180YC-FX および N9K-C93108TC-FX Fex ポート上での FCoE サポートは使用できます。

リリース バージョン 2.2(x)、において FCoE は、次の FEX Nexus デバイスではサポートも。

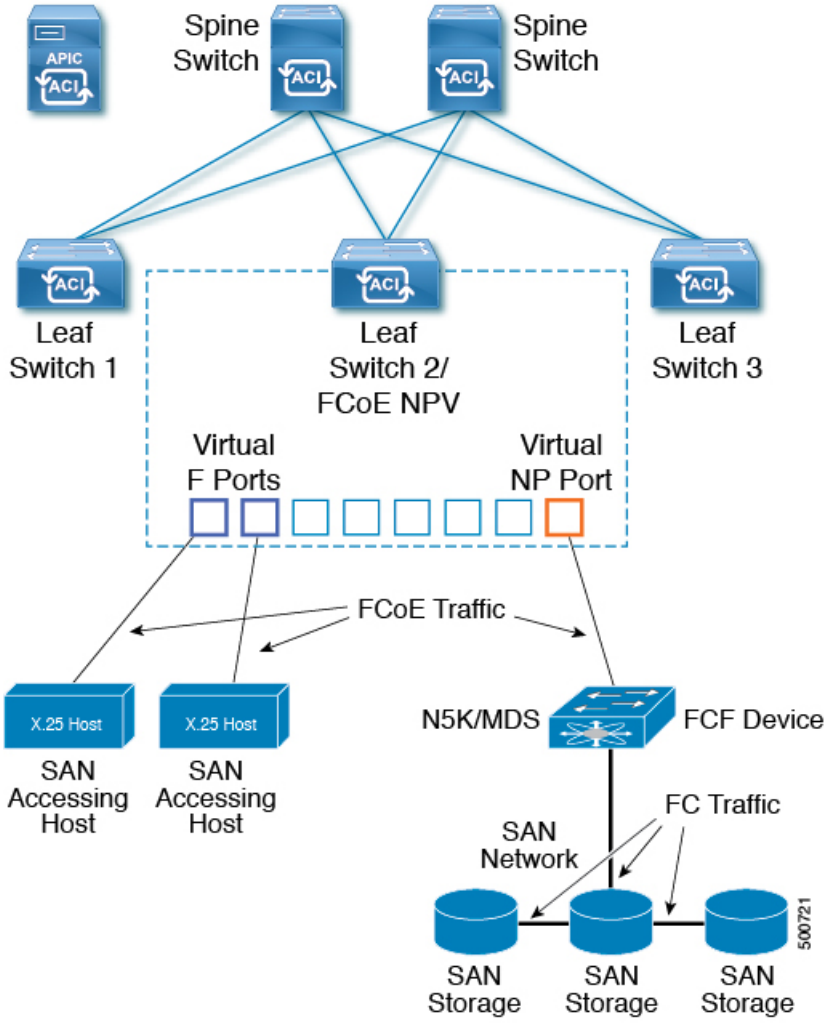
- 10 ギガ-ビット C2348UPQ N2K
- 10 ギガ-ビット C2348TQ N2K
- N2K-C2232PP-10GE
- N2K-B22DELL-P
- N2K-B22HP-P
- N2K-B22IBM-P
- N2K B22DELL P FI

（注） FCoE の使用されている vlan には、グローバルに設定 vlanScope 必要があります。FCoE の vlanScope portLocal に設定はサポートされていません。値は、L2 インターフェイス ポリシー l2IfPol 経由で設定されます。

FCoE ACI を通過するトラフィックをサポートするトポロジ

ACI ファブリック over FCoE トラフィックをサポートする一般的な設定のトポロジは、次のコンポーネントで構成されています。

図 *1 : ACI* トポロジ サポート *FCoE* トラフィック



- [FC SAN を介して NPV バックボーンとして機能するためのポリシーに設定された 1 つ以上の ACI Leaf スイッチ
- F ポートとして機能するように設定 NPV 設定のリーフ スイッチ上のインターフェイスを選択します。

F ポートでは、FCoE トラフィックと SAN 管理または SAN を消費しているアプリケーションを実行しているホストの対応します。

- NP ポートとして機能する NPV 設定のリーフ スイッチ上のインターフェイスを選択します。

  NP ポートでは、FCoE トラフィックと FCF ブリッジの対応します。

FCF ブリッジは、通常は SAN ストレージ デバイスを接続するファイバ チャネルのリンクから FC トラフィックを受信し、SAN 管理または SAN のデータを消費しているホストに ACI ファブリック上を伝送するための FCoE フレームに FC パケットをカプセル化します。FCoE トラフィックを受信し、ファイバ チャネル ネットワーク経由で伝送するための FC に戻る再します。

（注） 上記の ACI トポロジでは、FCoE トラフィックのサポートには、ホストと F ポートおよび FCF デバイスと NP ポート間の直接接続間の直接接続が必要です。

APIC サーバを設定し、APIC gui FCoE トラフィックをモニタ演算子を有効にする、APIC NX-OS スタイルの CLI、またはアプリケーションで、APIC REST API をコールします。

FCoE の初期化をサポートするトポロジ

FCoE トラフィック フローの説明に従ってを行われるするためには、する必要もありますSAN ホストが F ポートとして有効になっているインターフェイスを検出する、FCoE Initialization protocol (FIP) パケットがブロードキャストされる別の VLAN 接続を設定します。

vFC インターフェイス設定ルール

VFC のネットワークと APIC GUI、NX-OS スタイル CLI、または REST API を通じて EPG の導入を設定するかどうかは、プラットフォーム、次の一般的なルールが適用されます。

- F ポート モードは、vFC ポートのデフォルト モードです。NP ポート モードは、インターフェイス ポリシーで具体的に設定する必要があります。
- デフォルトのロード バランシング モードはリーフ スイッチ、またはインターフェイス レベル vFC 設定が src dst ox id。
- ブリッジ ドメインごとに 1 つの VSAN 割り当てがサポートされます。
- VSAN プールおよびVLAN プールの割り当てモードは、常にスタティックである必要があります。
- vFC のポートには、VSAN (ファイバ チャネル ドメインとも呼ばれる) が含まれるドメインの VLAN にマッピングされた VSAN との関連付けが必要です。

FCoE NX-OS スタイル CLI 設定

NX-OSスタイルCLIを使用したポリシーまたはプロファイルのないFCoE接続の設定

次の例の NX-OS スタイル CLI シーケンス EPG の FCoE 接続を設定する **e1** テナントで **t1** 設定またはスイッチ レベルとインターフェイス レベル ポリシーとプロファイルを適用せず。

手順

	コマンドまたはアクション	目的

ステップ 1	ターゲット テナントの下には、FCoE トラフィックをサポートするブリッジ ドメインを設定します。
	例：
	`apic1(config)# tenant t1`
	`apic1(config-tenant)# vrf context v1`
	`apic1(config-tenant-vrf)# exit`
	`apic1(config-tenant)# bridge-domain b1`
	`apic1(config-tenant-bd)# fc`
	`apic1(config-tenant-bd)# vrf member v1`
	`apic1(config-tenant-bd)# exit`
	`apic1(config-tenant)# exit`
		サンプル コマンド シーケンスはブリッジ ドメインを作成 **b1** テナントで **t1** FCoE 接続をサポートするように設定します。

ステップ 2	同じのテナントの下には、FCoE に設定されたブリッジ ドメインとターゲット EPG を関連付けます。
	例：
	`apic1(config)# tenant t1`
	`apic1(config-tenant)# application a1`
	`apic1(config-tenant-app)# epg e1`
	`apic1(config-tenant-app-epg)#`
	`bridge-domain member b1`
	`apic1(config-tenant-app-epg)# exit`
	`apic1(config-tenant-app)# exit`
	`apic1(config-tenant)# exit`
		サンプル コマンド シーケンス作成 EPG **e1** し、FCoE に設定されたブリッジ ドメインにその EPG を関連付けます **b1**
		。

ステップ 3	VLAN マッピングに VSAN ドメイン、VSAN プール、VLAN プール、VSAN を作成します。
	例：
	A
	`apic1(config)# vsan-domain dom1`
	`apic1(config-vsan)# vsan 1-10`
	`apic1(config-vsan)# vlan 1-10`
	`apic1(config-vsan)# fcoe vsan 1 vlan 1 loadbalancing src-dst-ox-id`
		例 A 、サンプル コマンド シーケンスは、VSAN ドメインを作成 **dom1** VSAN プールと VLAN プール、VSAN 1 を VLAN 1 にマッピングされ、VLAN 2 に VSAN 2 をマップ
		例 B 、代替サンプル コマンド シーケンスは再利用可能な VSAN 属性テンプレートを作成 **pol1** VSAN ドメインを作成し、 **dom1** 、そのテンプレートから属性とマッピングを継承します。

	コマンドまたはアクション	目的
	apic1(config-vsan)# fcoe vsan 2 vlan 2 例： **B** apic1(config)# template vsan-attribute pol1 apic1(config-vsan-attr)# fcoe vsan 2 vlan 12 loadbalancing src-dst-ox-id apic1(config-vsan-attr)# fcoe vsan 3 vlan 13 loadbalancing src-dst-ox-id apic1(config-vsan-attr)# exit apic1(config)# vsan-domain dom1 apic1(config-vsan)# vsan 1-10 apic1(config-vsan)# vlan 1-10 apic1(config-vsan)# inherit vsan-attribute pol1 apic1(config-vsan)# exit	
ステップ 4	FCoE Initialization (FIP) プロセスをサポートする物理ドメインを作成します。 例： apic1(config)# vlan-domain fipVlanDom apic1(config-vlan)# vlan 120 apic1(config-vlan)# exit	例では、コマンド シーケンスは、通常の VLAN ドメインを作成 **fipVlanDom** 、VLAN を含む **120** FIP プロセスをサポートします。
ステップ 5	ターゲット テナントの下には、定期的なブリッジ ドメインを設定します。 例： apic1(config)# tenant t1 apic1(config-tenant)# vrf context v2 apic1(config-tenant-vrf)# exit apic1(config-tenant)# bridge-domain fip-bd apic1(config-tenant-bd)# vrf member v2 apic1(config-tenant-bd)# exit apic1(config-tenant)# exit	コマンド シーケンスがブリッジ ドメインを作成例では、 **fip bd** 。
ステップ 6	同じのテナントの下には、設定されている定期的なブリッジドメインでこのEPGを関連付けます。 例： apic1(config)# tenant t1 apic1(config-tenant)# application a1 apic1(config-tenant-app)# epg epg-fip apic1(config-tenant-app-epg)# bridge-domain member fip-bd apic1(config-tenant-app-epg)# exit apic1(config-tenant-app)# exit apic1(config-tenant)# exit	例では、コマンド シーケンス関連付けます EPG **epg fip** ブリッジ ドメインを **fip bd** 。

	コマンドまたはアクション	目的

ステップ 7	VFC インターフェイスを F モードで設定します。
	例：
	A
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# interface ethernet 1/2`
	`apic1(config-leaf-if)# vlan-domain member fipVlanDom`
	`apic1(config-leaf-if)# switchport trunk native vlan 120 tenant t1 application a1 epg epg-fip`
	`apic1(config-leaf-if)# exit`
	
	`apic1(config-leaf)# exit`
	`apic1(config-leaf)# interface vfc 1/2`
	`apic1(config-leaf-if)# switchport mode f`
	`apic1(config-leaf-if)# vsan-domain member dom1`
	`apic1(config-leaf-if)# switchport vsan 2 tenant t1 application a1 epg e1`
	`apic1(config-leaf-if)# switchport trunk allowed vsan 3 tenant t1 application a1 epg e2`
	`apic1(config-leaf-if)# exit`
	例：
	B
	`apic1(config)# vpc context leaf 101 102`
	`apic1(config-vpc)# interface vpc vpc1`
	`apic1(config-vpc-if)# vlan-domain member vfdom100`
	`apic1(config-vpc-if)# vsan-domain member dom1`
	`apic1(config-vpc-if)# #For FIP discovery`
	`apic1(config-vpc-if)# switchport trunk native vlan 120 tenant t1 application a1 epg epg-fip`
	`apic1(config-vpc-if)# switchport vsan 2 tenant t1 application a1 epg e1`
	`apic1(config-vpc-if)# exit`
	`apic1(config-vpc)# exit`
	`apic1(config)# leaf 101-102`
	`apic1(config-leaf)# interface ethernet 1/3`
	`apic1(config-leaf-if)# channel-group vpc1 vpc`
	`apic1(config-leaf-if)# exit`
	`apic1(config-leaf)# exit`
	例：
	C
		例では **A** コマンド シーケンスは、インターフェイスを有効に **1/2** リーフ スイッチで **101** として機能する、 **F** ポートおよびインターフェイスの VSAN のドメインに関連 **dom1** 。
		ネイティブ モードで 1 つ (と 1 つだけ) の VSAN 対象のインターフェイスの各割り当てる必要があります。各インターフェイスには、通常モードで1つ以上の追加 Vsan を割り当てることができます。
		サンプル コマンド シーケンスは、対象のインターフェイスを関連付けます **1/2** と。
		• VLAN **120** FIP ディスカバリの EPG に関連付けます **epg fip** およびアプリケーション **a1** テナントで **t1** 。
		• VSAN **2** ネイティブ VSAN として、EPG に関連付けます **e1** およびアプリケーション **a1** テナントで **t1** 。
		• VSAN **3** 定期的な VSAN として。
		例では **B** 、コマンド シーケンスは、両方のレッグ上で同じ、VSAN で VPC over vFC を設定します。CLI からログごとに異なる Vsan を指定することはできません。代替設定は、GUI を高度な apic 内で実行できます。

	コマンドまたはアクション	目的
	apic1(config)# leaf 101 apic1(config-leaf)# interface vfc-po pc1 apic1(config-leaf-if)# vsan-domain member dom1 apic1(config-leaf-if)# switchport vsan 2 tenant t1 application a1 epg e1 apic1(config-leaf-if)# exit apic1(config-leaf)# interface ethernet 1/2 apic1(config-leaf-if)# channel-group pc1 apic1(config-leaf-if)# exit apic1(config-leaf)# exit	
ステップ 8	VFC インターフェイスを NP モードで設定します。 **例：** apic1(config)# leaf 101 apic1(config-leaf)# interface vfc 1/4 apic1(config-leaf-if)# switchport mode np apic1(config-leaf-if)# vsan-domain member dom1	サンプル コマンド シーケンスは、インターフェイスを有効に **1/4** リーフ スイッチで **101** として機能する、 **NP** ポートおよびインターフェイスの VSAN のドメインに関連 **dom1** 。
ステップ 9	VSAN を対象となる FCoE 対応インターフェイスに割り当てます。 **例：** apic1(config-leaf-if)# switchport trunk allowed vsan 1 tenant t1 application a1 epg e1 apic1(config-leaf-if)# switchport vsan 2 tenant t4 application a4 epg e4	ネイティブ モードで 1 つ (と 1 つだけ) の VSAN 対象のインターフェイスの各割り当てる必要があります。各インターフェイスには、通常モードで 1 つ以上の追加 Vsan を割り当てることができます。 サンプル コマンド シーケンスは、ターゲット インターフェイスを VSAN **1** に割り当て、それを EPG **e1** とアプリケーション **a1** にテナント **t1** の下で関連付けます。「trunk allowed」は、VSAN 1 に通常モードのステータスを割り当てます。コマンド シーケンスも割り当てます、インターフェイス、必要な **ネイティブ モード** VSAN **2** 。次の例に示すは、同一のインターフェイスを異なるテナント アクセスで実行されているさまざまな Epg を提供するためにさまざまな Vsan の動作を渡します。

NX-OSスタイルCLIを使用したポリシーまたはプロファイルがあるFCoE接続の設定

次の例 NX-OS スタイル CLI のシーケンスを作成し、EPG の FCoE 接続を設定するポリシーを使用して **e1** テナントで **t1** 。

手順

	コマンドまたはアクション	目的

ステップ 1	ターゲット テナントの下には、FCoE トラフィックをサポートするブリッジ ドメインを設定します。
	例：
	`apic1# configure`
	`apic1(config)# tenant t1`
	`apic1(config-tenant)# vrf context v1`
	`apic1(config-tenant-vrf)# exit`
	`apic1(config-tenant)# bridge-domain b1`
	`apic1(config-tenant-bd)# fc`
	`apic1(config-tenant-bd)# vrf member v1`
	`apic1(config-tenant-bd)# exit`
	`apic1(config-tenant)# exit`
	`apic1(config)#`
		サンプル コマンド シーケンスはブリッジ ドメインを作成 **b1** テナントで **t1** FCoE接続をサポートするように設定します。

ステップ 2	同じのテナントの下には、設定されている FCoE ブリッジ ドメインと、ターゲット EPG を関連付けます。
	例：
	`apic1(config)# tenant t1`
	`apic1(config-tenant)# application a1`
	`apic1(config-tenant-app)# epg e1`
	`apic1(config-tenant-app-epg)# bridge-domain member b1`
	`apic1(config-tenant-app-epg)# exit`
	`apic1(config-tenant-app)# exit`
	`apic1(config-tenant)# exit`
	`apic1(config)#`
		サンプル コマンド シーケンス作成EPG **e1** その EPG の FCoE に設定されたブリッジ ドメイン関連付け **b1** 。

ステップ 3	VLAN マッピングに VSAN ドメイン、VSAN プール、VLAN プール、VSAN を作成します。
	例：
	A
	`apic1(config)# vsan-domain dom1`
	`apic1(config-vsan)# vsan 1-10`
	`apic1(config-vsan)# vlan 1-10`
	`apic1(config-vsan)# fcoe vsan 1 vlan`
		例 A 、サンプル コマンド シーケンスは、VSAN ドメインを作成 **dom1** VSAN プールと VLAN プール、マップ VSAN 1 VLAN 1 と VLAN 2 に VSAN 2 をマップ
		例 B 、代替サンプル コマンド シーケンスは再利用可能な vsan 属性テンプレートを作成 **pol1** VSAN ドメインを

	コマンドまたはアクション	目的
	1 loadbalancing src-dst-ox-id apic1(config-vsan)# fcoe vsan 2 vlan 2 **例：** **B** apic1(config)# template vsan-attribute pol1 apic1(config-vsan-attr)# fcoe vsan 2 vlan 12 loadbalancing src-dst-ox-id apic1(config-vsan-attr)# fcoe vsan 3 vlan 13 loadbalancing src-dst-ox-id apic1(config-vsan-attr)# exit apic1(config)# vsan-domain dom1 apic1(config-vsan)# inherit vsan-attribute pol1 apic1(config-vsan)# exit	作成し、 **dom1** 、そのテンプレートから属性とマッピングを継承します。
ステップ 4	FCoE Initialization (FIP) プロセスをサポートする物理ドメインを作成します。 **例：** apic1(config)# vlan-domain fipVlanDom apic1(config)# vlan-pool fipVlanPool	
ステップ 5	ファイバ チャネル SAN ポリシーを設定します。 **例：** apic1# apic1# configure apic1(config)# template fc-fabric-policy ffp1 apic1(config-fc-fabric-policy)# fctimer e-d-tov 1111 apic1(config-fc-fabric-policy)# fctimer r-a-tov 2222 apic1(config-fc-fabric-policy)# fcoe fcmap 0E:FC:01 apic1(config-fc-fabric-policy)# exit	サンプル コマンド シーケンスは、SAN のファイバ チャネル ポリシーを作成 **ffp1** の組み合わせを指定するエラー検出タイムアウト値 (EDTOV)、resource allocation(リソース割り当て、リソースの割り当て) タイムアウト値 (RATOV)、およびターゲット リーフ上の FCoE 対応のインターフェイスのデフォルト FC マップ値スイッチです。
ステップ 6	ファイバ チャネル ノード ポリシーを作成します。 **例：** apic1(config)# template fc-leaf-policy flp1 apic1(config-fc-leaf-policy)# fcoe fka-adv-period 44 apic1(config-fc-leaf-policy)# exit	サンプル コマンド シーケンスは、ファイバ チャネル ノードのポリシーを作成 **flp1** を中断のロード バランシングの有効化と FIP キープア ライブ値の組み合わせを指定します。これらの値は、ターゲット リーフ スイッチ上のすべて

	コマンドまたはアクション	目的
		の FCoE 対応インターフェイスにも適用されます。

ステップ 7	ノード ポリシー グループを作成します。
	例：
	`apic1(config)# template leaf-policy-group lpg1`
	`apic1(config-leaf-policy-group)# inherit fc-fabric-policy ffp1`
	`apic1(config-leaf-policy-group)# inherit fc-leaf-policy flp1`
	`apic1(config-leaf-policy-group)# exit`
	`apic1(config)# exit`
	`apic1#`
		サンプル コマンド シーケンスはノード ポリシー グループを作成 **lpg1** 、SAN のファイバ チャネル ポリシーの値を結合する **ffp1** とファイバ チャネル ノードのポリシー、 **flp1** 。このノード ポリシー グループの合計値は、後で設定されているノードのプロファイルに適用できます。

ステップ 8	ノード プロファイルを作成します。
	例：
	`apic1(config)# leaf-profile lp1`
	`apic1(config-leaf-profile)# leaf-group lg1`
	`apic1(config-leaf-group)# leaf 101`
	`apic1(config-leaf-group)# leaf-policy-group lpg1`
		サンプル コマンド シーケンスがノードのプロファイルを作成 **lp1** ノード ポリシー グループと関連付けます **lpg1** 、ノード グループ **lg1** 、およびリーフ スイッチ **101** 。

ステップ 9	F ポート インターフェイスのインターフェイス ポリシー グループを作成します。
	例：
	`apic1(config)# template policy-group ipg1`
	`apic1(config-pol-grp-if)# priority-flow-control mode auto`
	`apic1(config-pol-grp-if)# switchport mode f`
	`apic1(config-pol-grp-if)# slow-drain pause timeout 111`
	`apic1(config-pol-grp-if)# slow-drain congestion-timeout count 55`
	`apic1(config-pol-grp-if)# slow-drain congestion-timeout action log`
		サンプル コマンド シーケンスは、インターフェイス グループのポリシーを作成 **ipg1** し、プライオリティ フロー制御の有効化、F ポートの有効化、およびこのポリシー グループに適用されているすべてのインターフェイスに対して低速ドレイン ポリシーの値を決定する値の組み合わせを割り当てます。

ステップ 10	NP ポート インターフェイスのインターフェイス ポリシー グループを作成します。
	例：
	`apic1(config)# template policy-group ipg2`
	`apic1(config-pol-grp-if)# priority-flow-control mode auto`
	`apic1(config-pol-grp-if)# switchport mode np`
		サンプル コマンド シーケンスは、インターフェイス グループ ポリシー **ipg2** を作成し、このポリシー グループに適用されているすべてのインターフェイスに対して、優先順位フロー制御の有効化、NP ポートの有効化、低速ドレイン ポリシーの値を決定する値の組み合わせを割り当てます。

	コマンドまたはアクション	目的
	apic1(config-pol-grp-if)# slow-drain pause timeout 111 apic1(config-pol-grp-if)# slow-drain congestion-timeout count 55 apic1(config-pol-grp-if)# slow-drain congestion-timeout action log	
ステップ 11	F ポート インターフェイスのインターフェイス プロファイルを作成します。 **例**： apic1# configure apic1(config)# leaf-interface-profile lip1 apic1(config-leaf-if-profile)# description 'test description lip1' apic1(config-leaf-if-profile)# leaf-interface-group lig1 apic1(config-leaf-if-group)# description 'test description lig1' apic1(config-leaf-if-group)# policy-group ipg1 apic1(config-leaf-if-group)# interface ethernet 1/2-6, 1/9-13	サンプル コマンド シーケンスは、インターフェイス プロファイルを作成 **lip1** F ポートのインターフェイスの F ポートの特定のインターフェイス ポリシー グループ プロファイルを関連付けます **ipg1** 、このインターフェイスを指定しプロファイルとその関連するポリシー。適用されます。
ステップ 12	NP ポート インターフェイスのインターフェイス プロファイルを作成します。 **例**： apic1# configure apic1(config)# leaf-interface-profile lip2 apic1(config-leaf-if-profile)# description 'test description lip2' apic1(config-leaf-if-profile)# leaf-interface-group lig2 apic1(config-leaf-if-group)# description 'test description lig2' apic1(config-leaf-if-group)# policy-group ipg2 apic1(config-leaf-if-group)# interface ethernet 1/14	サンプル コマンド シーケンスは、インターフェイス プロファイルを作成 **lip2** NP ポート インターフェイス、NP ポートの特定のインターフェイス ポリシー グループ プロファイルに関連付けます **ipg2** 、このインターフェイスを指定し、プロファイルとその関連するポリシー適用されます。
ステップ 13	レベル 1 の QoS クラス ポリシーを設定します。 **例**： apic1(config)# qos parameters level1 apic1(config-qos)# pause no-drop cos 3	サンプル コマンド シーケンスは、FCoE トラフィック プライオリティ フロー制御ポリシーを適用することがおよび非ドロップ パケットのクラスのサービス レベル 3 の処理を一時停止の QoS レベルを指定します。

NX-OS スタイル CLI を使用して FCoE オーバー FEX の設定

FEX ポートは、ポート Vsan として設定されます。

手順

ステップ 1 テナントと VSAN のドメインを設定します。

例：

```
apic1# configure
apic1(config)#  tenant t1
apic1(config-tenant)#   vrf context v1
apic1(config-tenant-vrf)#     exit
apic1(config-tenant)#   bridge-domain b1
apic1(config-tenant-bd)#       fc
apic1(config-tenant-bd)#       vrf member v1
apic1(config-tenant-bd)#       exit
apic1(config-tenant)#   application a1
apic1(config-tenant-app)#      epg e1
apic1(config-tenant-app-epg)# bridge-domain member b1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)#      exit
apic1(config-tenant)# exit

apic1(config)#  vsan-domain dom1
apic1(config-vsan)# vlan 1-100
apic1(config-vsan)# vsan 1-100
apic1(config-vsan)# fcoe vsan 2 vlan 2 loadbalancing src-dst-ox-id
apic1(config-vsan)# fcoe vsan 3 vlan 3 loadbalancing src-dst-ox-id
apic1(config-vsan)# fcoe vsan 5 vlan 5
apic1(config-vsan)# exit
```

ステップ 2 FEX をインターフェイスに関連付けます。

例：

```
apic1(config)#  leaf 101
apic1(config-leaf)#  interface ethernet 1/12
apic1(config-leaf-if)#  fex associate 111
apic1(config-leaf-if)#  exit
```

ステップ 3 ポート、ポート チャネル、および VPC あたり FEX を介して FCoE を設定します。

例：

```
apic1(config-leaf)#  interface vfc 111/1/2
apic1(config-leaf-if)#  vsan-domain member dom1
apic1(config-leaf-if)#  switchport vsan 2 tenant t1 application a1 epg e1
apic1(config-leaf-if)# exit

apic1(config-leaf)#  interface vfc-po pc1 fex 111
apic1(config-leaf-if)# vsan-domain member dom1
apic1(config-leaf-if)# switchport vsan 2 tenant t1 application a1 epg e1
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 111/1/3
apic1(config-leaf-if)# channel-group pc1
apic1(config-leaf-if# exit
apic1(config-leaf)# exit
```

```
apic1(config)# vpc domain explicit 12 leaf 101 102
apic1(config-vpc)# exit
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)#     interface vpc vpc1 fex 111 111
apic1(config-vpc-if)#       vsan-domain member dom1
apic1(config-vpc-if)#       switchport vsan 2 tenant t1 application a1 epg e1
apic1(config-vpc-if)#       exit
apic1(config-vpc)#     exit
apic1(config)# leaf 101-102
apic1(config-leaf)# interface ethernet 1/2
apic1(config-leaf-if)# fex associate 111
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 111/1/2
apic1(config-leaf-if)# channel-group vpc1 vpc
apic1(config-leaf-if)# exit
```

ステップ 4 設定を確認するには、次のコマンドを実行します。

例：

```
apic1(config-vpc)# show vsan-domain detail
vsan-domain : dom1

vsan : 1-100

vlan : 1-100

Leaf          Interface          Vsan  Vlan  Vsan-Mode    Port-Mode  Usage
 Operational State
------------  -----------------  ----  ----  -----------  ---------
---------------------------------------
101           vfc111/1/2       2       2    Native                  Tenant: t1
  Deployed
                                                                     App: a1
                                                                     Epg: e1

101           PC:pc1           5       5    Native                Tenant: t1
  Deployed
                                                                     App: a1
                                                                     Epg: e1

101           vfc111/1/3       3       3    Native       F          Tenant: t1
  Deployed
                                                                     App: a1
                                                                     Epg: e1
```

NX-OS スタイルの CLI を使用した FCoE 設定の検証

次 **show** コマンドは、リーフ スイッチ ポートで FCoE の設定を確認します。

手順

使用して、 **vsan ドメインを表示** コマンドをターゲット スイッチで FCoE が有効になっていることを確認します。

コマンドの例では、FCoE がリストされているリーフ スイッチおよび接続の詳細を FCF で有効になっていることを確認します。

例：

```
ifav-isim8-ifc1# show vsan-domain detail
vsan-domain : iPostfcoeDomP1

vsan : 1-20  51-52  100-102  104-110  200  1999  3100-3101  3133
       2000

vlan : 1-20  51-52  100-102  104-110  200  1999  3100-3101  3133
       2000

                                      Vsan    Port                    Operational
 Leaf  Interface       Vsan  Vlan     Mode    Mode  Usage             State
 ----  ---------       ----  ----     -------- ----  ----------------  ------------
 101   vfc1/11         1     1        Regular  F     Tenant: iPost101  Deployed

                                                    App: iPost1

                                                    Epg: iPost1


 101   vfc1/12         1     1        Regular  NP    Tenant: iPost101  Deployed

                                                    App: iPost1

                                                    Epg: iPost1


 101   PC:infraAccBndl 4     4        Regular  NP    Tenant: iPost101  Deployed

          Grp_pc01                                  App: iPost4

                                                    Epg: iPost4


 101   vfc1/30         2000           Native        Tenant: t1        Not deployed
                                                    App: a1           (invalid-path)

                                                    Epg: e1
```

NX-OS スタイル CLI を使用した FCoE 要素の展開解除

ACI ファブリックから FCoE 接続を導入解除に移動してもでは、いくつかのレベルで FCoE コンポーネントを削除することが必要です。

手順

ステップ 1 リーフ ポート インターフェイスの属性のリスト、そのモードの設定をデフォルトに設定し、その EPG の導入とドメインの関連付けを削除します。

インターフェイス vfc のポート モードの設定を設定する例 **1/2** のデフォルトに [EPG の導入を削除 **e1** と VSAN ドメインに関連付け **dom1** そのインターフェイスから。

例：

```
apic1(config)# leaf 101
apic1(config-leaf)# interface vfc 1/2
apic1(config-leaf-if)# show run
# Command: show running-config leaf 101 interface vfc 1 / 2
# Time: Tue Jul 26 09:41:11 2016
  leaf 101
    interface vfc 1/2
      vsan-domain member dom1
      switchport vsan 2 tenant t1 application a1 epg e1
      exit
    exit
apic1(config-leaf-if)# no switchport mode
apic1(config-leaf-if)# no switchport vsan 2 tenant t1 application a1 epg e1
apic1(config-leaf-if)# no vsan-domain member dom1
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

ステップ 2 一覧表示し、VSAN マッピングおよび VLAN と VSAN のプールを削除します。

例の VSAN の VLAN マッピングが削除されます **vsan 2** 、VLAN プール **1-10** 、および VSAN プール **1-10** VSAN ドメインから **dom1** 。

例：

```
apic1(config)# vsan-domain dom1
apic1(config-vsan)# show run
# Command: show running-config vsan-domain dom1
# Time: Tue Jul 26 09:43:47 2016
  vsan-domain dom1
    vsan 1-10
    vlan 1-10
    fcoe vsan 2 vlan 2
    exit
apic1(config-vsan)# no fcoe vsan 2
apic1(config-vsan)# no vlan 1-10
apic1(config-vsan)# no vsan 1-10
apic1(config-vsan)# exit

################################################################################
NOTE: To remove a template-based VSAN to VLAN mapping use an alternate sequence:
################################################################################

apic1(config)# template vsan-attribute <template_name>
apic1(config-vsan-attr)# no fcoe vsan 2
```

ステップ 3 VSAN ドメインを削除します。

例は、ドメインの VSAN を削除する **dom1** 。

例：

```
apic1(config)# no vsan-domain dom1
```

ステップ 4 必要はないかどうかは、関連付けられているテナント、EPG、およびセレクタを削除できます。

ファイバ チャネル NPV

ファイバ チャネル接続の概要

スイッチは、NPV を有効にした後は NPV モードになります。NPV モードはスイッチ全体に適用されます。NPV モードのスイッチに接続するすべてのエンド デバイスは、N ポートとしてログインし、この機能を使用する必要があります（ループ接続デバイスはサポートされていません）。（NPV モードの）エッジ スイッチから NPV コア スイッチへのすべてのリンクは、（E ポートではなく）NP ポートとして確立されます。このポートは、通常のスイッチ間リンクに使用されます。

FC NPV の利点

FC NPV では次の機能を提供します。

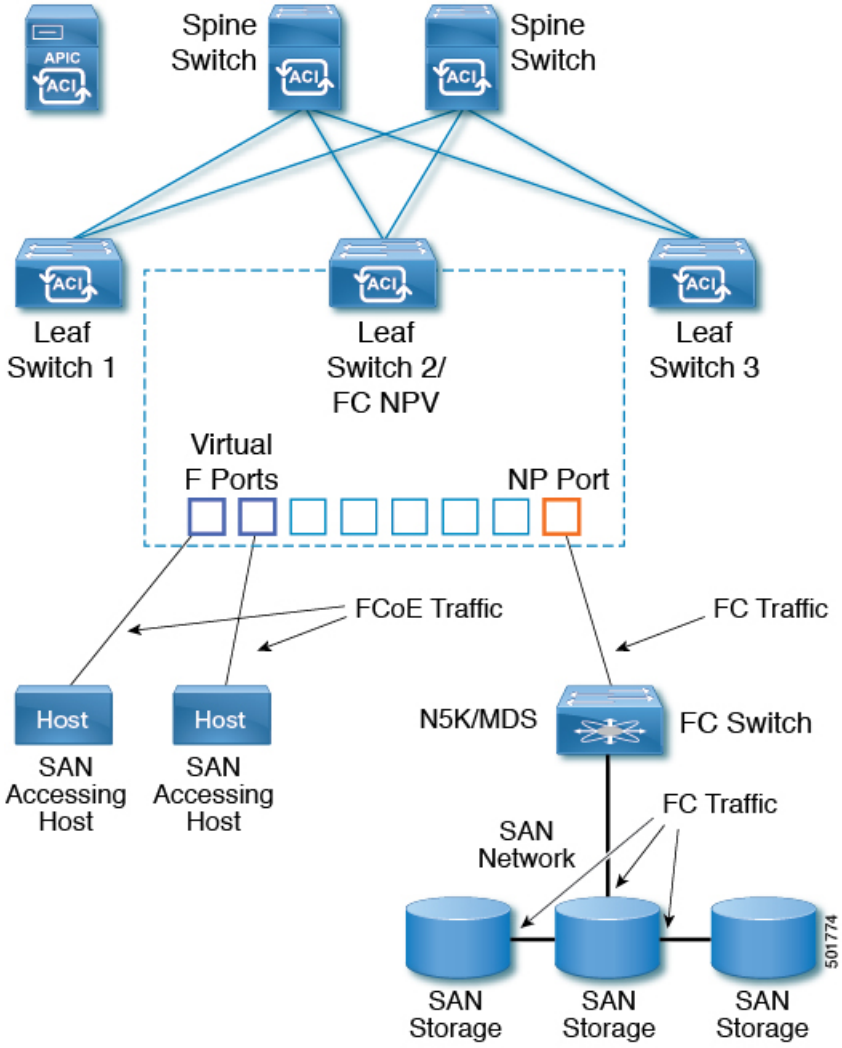
- ファブリックのドメイン ID を追加することがなく、ファブリックに接続しているホスト数の増加する
- FC および FCoE ホストとターゲットの FC インターフェイスを使用した SAN ファブリックへの接続
- 自動トラフィック マッピング
- スタティック トラフィック マッピング
- 自動ロード バランシングの中断

FC NPV モード

ACI の Feature-set fcoe-npv は、最初に FCoE/FC 設定がプッシュされるときに、デフォルトで自動的に有効なります。

FC トポロジ

ACI ファブリックで FC トラフィックをサポートする一般的な設定のトポロジは、次のコンポーネントで構成されています。



- リーフは、FCoE NP ポートまたはネイティブ FC NP ポートを使用して FC スイッチに接続できます。
- ACI リーフは、FCoE リンクを使用してサーバ/ストレージと直接接続することができます。
- FC/FCoE トラフィックは、ファブリック/スパインには送信されません。リーフ スイッチでは、FCoE トラフィックのローカル スイッチングは実行しません。スイッチングは、FC/FCoE NPV リンク経由でリーフ スイッチと接続されているコア スイッチによって行われます。
- Flogi に続く複数の FDISC は、FCoE ホストと FC/FCoE NP リンクによりサポートされます。

FC NPV の注意事項と制約事項

FC NPV を設定する場合、次の注意事項および制限事項に注意してください。

- FC 設定に使用できるポートの範囲は 1 ～ 48 です。ポート 49 ～ 54 までは FC ポートを変換できません。
- F ポートとしての FC ポート モードはサポートされていません。
- SAN ポート チャネルがサポートされていません。
- FC NPV リンクで FEX 経由の FCoE ホストはサポートされていません。
- FC 速度の自動化および 32 G はサポートされていません。
- ACI リーフ 93180YC-FX ポートが 8 G 速度で設定されているとき、Brocade Port Blade FC16-32 への FC Uplink（NP）接続はサポートされていません。
- FC トランク モード **ON** および **自動化** はサポートされていません。
- FC 塗りつぶしパターン ARBFF はサポートされていません。
- FC は 40 G およびブレイク アウト ポートではサポートされていません。
- FC ポートでは、FEX 起動はサポートされていません。
- FC NPV のサポートは、リリース 3.2(1) において N9K-C93180YC-FX に限られます。
- イーサネットから FC（またはその逆）のポート変換には、スイッチのリロードが必要です。現在 1 個のポートの連続範囲のみ FC ポートに変換可能で、この範囲はあり 4 の倍数で終わるポート番号である必要があります。例：1 ~ 4、1 ~ 8、または 21-24。

NX-OS CLI を使用したポリシーまたはプロファイルのない FC 接続の設定

次の例コマンド シーケンスは、FCoE 接続をサポートするように設定テナント t1 でブリッジ ドメイン b1 を作成します。

始める前に

- ターゲット テナントの下には、FCoE トラフィックをサポートするブリッジ ドメインを設定します。

手順

ステップ 1 FCoE 接続をブリッジ ドメインを作成します。

例：

```
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain b1
apic1(config-tenant-bd)# fc
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# exit
```

ステップ 2 同じのテナントの下には、FCoE に設定されたブリッジ ドメインとターゲット EPG を関連付けます。次のコマンド シーケンスの例では、EPG e1 を作成し、FCoE に設定されたブリッジ ドメイン b1 にその EPG を関連付けます。

例：

```
apic1(config)# tenant t1
apic1(config-tenant)# application a1
apic1(config-tenant-app)# epg e1
apic1(config-tenant-app-epg)# bridge-domain member b1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
```

ステップ 3 次の例では、1 ～ 10 の VSAN で VSAN ドメイン dom1 が作成されます。

例：

```
apic1(config)# vsan-domain dom1
apic1(config-vsan)# vsan 1-10
```

ステップ 4 ポートの範囲をイーサネットから FC モードに変換します。次の例では、スイッチのポート 1/1-4 を 101 から FC に変換します。

例：

```
apic1# config
apic1(config)# leaf 101
apic1(config-leaf)# slot 1
apic1(config-leaf-slot)# port 1 4 type fc
apic1(config-leaf-slot)# exit
apic1(config-leaf)# exit
```

（注） イーサネットから FC（およびその逆）へのポート変換には、スイッチのリロードが必要です。

ステップ 5 FC インターフェイスを NP モードに設定します。次の例では、インターフェイス fc 1/10 にさまざまなインターフェイス プロパティを設定し、VSAN ドメイン dom1 にそのインターフェイスを関連付けます。対象のインターフェイスのそれぞれが、ネイティブ モードで 1 個（そして 1 個のみ）の VSAN を割り当てる必要があります。サンプル コマンド シーケンスは、対象のインターフェイス 1/10 をネイティブ VSAN として VSAN 10 に関連付け、テナント t1 で EPG e1 およびアプリケーション a1 に関連付けます。

例：

```
apic1(config-leaf)# interface fc 1/10
apic1(config-leaf-fc-if)# switchport mode [f | np]
apic1(config-leaf-fc-if)# switchport rxbbcredit <16-64>
apic1(config-leaf-fc-if)# switchport speed [16G | 32G | 4G | 8G | auto | unknown]
apic1(config-leaf-fc-if)# vsan-domain member dom1
apic1(config-leaf-fc-if)# switchport vsan 10 tenant t1 application a1 epg e1
```

ステップ 6 サーバ ポートをアップリンク ポートにピンするためトラフィック マップを作成します。次の例では、FC 1/7 アップリンク インターフェイスにピンされている vFC 1/47 サーバインターフェイスにトラフィック マップを作成します。

例：

```
apic1# config
apic1(config)# leaf 101
apic1(config-leaf)# npv traffic-map server-interface vfc 1/47 label label1 tenant tenant1
 application app1 epg epg1
apic1(config-leaf)# npv traffic-map external-interface fc 1/7 tenant tenant1 label label1
```

ポリシーまたは NX-OS は、CLI を使用したプロファイルと FC 接続の設定

次の例コマンド シーケンスは、FCoE 接続をサポートするように設定テナント t1 でブリッジ ドメイン b1 を作成します。

始める前に

- ターゲット テナントの下には、FCoE トラフィックをサポートするブリッジ ドメインを設定します。

手順

ステップ 1 FCoE 接続をブリッジ ドメインを作成します。

例：

```
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain b1
apic1(config-tenant-bd)# fc
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# exit
```

ステップ 2 同じのテナントの下には、FCoE に設定されたブリッジ ドメインとターゲット EPG を関連付けます。次の例コマンド シーケンスは、EPG e1 を作成し、FCoE に設定されたブリッジ ドメイン b1 にその EPG を関連付けます。

例：

```
apic1(config)# tenant t1
apic1(config-tenant)# application a1
apic1(config-tenant-app)# epg e1
apic1(config-tenant-app-epg)# bridge-domain member b1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
```

ステップ 3 VSAN ドメインを作成します。次の例では、1 ~ 10 の vsan の vsan ドメイン dom1 が作成されます。

例：

```
apic1(config)# vsan-domain dom1
apic1(config-vsan)# vsan 1-10
```

ステップ 4 NP ポート インターフェイスのインターフェイス ポリシー グループを作成します。サンプル コマンド シーケンスは、FC インターフェイス ポリシー グループ ipg2 を作成し、このポリシー グループに適用されているインターフェイスの値を決定する値の組み合わせを割り当てます。

例：

```
apic1(config)# template fc-policy-group ipg1
apic1(config-fc-pol-grp-if)# switchport ?
 fill-pattern  Configure fill pattern for fc interface
 mode         Configure port mode for fc interface
 rxbbcredit   Configure rxBBCredit for fc interface
 speed        Configure speed for fc interface
 trunk-mode    Configure trunk-mode for fc interface
apic1(config-fc-pol-grp-if)# switchport fill-pattern [ARBFF | IDLE]
apic1(config-fc-pol-grp-if)# switchport mode [f | np]
apic1(config-fc-pol-grp-if)# switchport rxbbcredit <16-64>
apic1(config-fc-pol-grp-if)# switchport speed [16G | 32G | 4G | 8G | auto | unknown]
apic1(config-fc-pol-grp-if)# vsan-domain member dom1
```

ステップ 5 FC ポート インターフェイスのインターフェイス プロファイルを作成します。サンプル コマンド シーケンスの FC ポート インターフェイスのインターフェイス プロファイル lip1 を作成するには、プロファイルの関連付け FC インターフェイス ポリシー グループ ipg1、およびプロファイルとその関連するポリシーが適用このインターフェイスを指定します。

例：

```
apic1# configure
apic1(config)# leaf-interface-profile lip1
apic1(config-leaf-if-profile)# description 'test description lip1'
apic1(config-leaf-if-profile)# leaf-interface-group lig1
apic1(config-leaf-if-group)# description 'test description lig1'
apic1(config-leaf-if-group)# fc-policy-group ipg1
apic1(config-leaf-if-group)# interface fc 1/1-4
```

ステップ 6 リーフ プロファイルを作成し、そのリーフ プロファイルにリーフ インターフェイス プロファイルを割り当てて、そのプロファイルの適用先となるリーフ ID を割り当てます:

例：

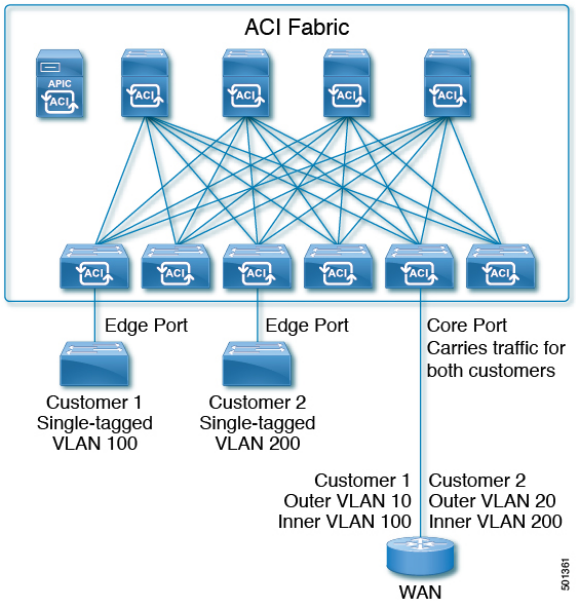
```
apic1(config)# leaf-profile lp103
apic1(config-leaf-profile)# leaf-interface-profile lip1
apic1(config-leaf-profile)# leaf-group range
apic1(config-leaf-group)# leaf 103
apic1(config-leaf-group)#
```

（注） リーフ インターフェイス プロファイル リーフに関連付けると、リーフのリロードが FC ポートとしてこれらのポートを起動する必要があります。

802.1Q トンネルの設定

ACI 802.1 q トンネルについて

図 *2 : ACI 802.1 q* トンネル



Cisco ACI および Cisco APIC リリース 2.2(1x) 以降では、エッジ（トンネル）ポートで 802.1Q トンネルを設定して、QoS（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、タグ付き、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。ACI 前面パネル ポートは、**Dot1q トンネル** の一部である可能性があります。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、第二世代以降のスイッチのモデル名の最後に「EX」がつく Cisco Nexus 9000 シリーズ スイッチでサポートされます。

Cisco ACI および Cisco APIC リリース 2.3(x) では、同じコア ポートで複数の 802.1 q トンネルを設定可能で、複数の顧客から二重タグ付きトラフィックを伝送し、これは 802.1 q 用に設定されたアクセスのカプセル化を各自識別できます。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。**Dot1q トンネル**のエッジ ポートとコア ポートの両方は、Cisco Nexus 9000 シリーズ スイッチの第三世代の、スイッチ モデル名の末尾に「FX」が付いている機種でサポートされます。

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。ポートチャネル（PC）または仮想ポート チャネル（vPC）ではサポートされません。
  - 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 802.1 q トンネル ブリッジ ドメインではサポートされません。
  - 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点滅し、ブリッジ ドメインでフラッディングして 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。
  - Cisco APIC リリース 2.3(x) 以降では、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 トンネル** の唯一のインターフェイスであり、**削除**され再度設定される場合、**Dot1q トンネル** に対する PC/VPC の関連付けを削除してから再設定します。
- Cisco APIC リリース 2.2(x) では、二重タグつきフレームのイーサタイプは 0x8100 の後に 0x9100 が続く必要があります。

  ただし、Cisco APIC リリース 2.3(x) 以降の場合、この制限は第三世代のスイッチ モデル名の最後に「FX」がつく Cisco Nexus switches のエッジ ポートには適用されません。

- 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（リーフ スイッチ上も）を **Dot1q トンネル** に含むことができます。
- 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。
- Cisco APIC リリース 2.3(x) 以降では、通常の EPG が 802.1 q 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** に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 は **Dot1q トンネル** のメンバとしてはサポートされていません。
- ブレークアウト ポートとして設定されているインターフェイスは、802.1 q をサポートしていません。
- 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。

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 チャネル、仮想ポート チャ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 チャネルと仮想ポート チャ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。	
ステップ 3	**leaf** *ID* 例： `apic1(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： `apic1(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：
	`apic1(config-leaf-if)# switchport mode dot1q-tunnel edgePort`
	`apic1(config-leaf-if)# exit`
	`apic1(config-leaf)# exit`
	`apic1(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： `apic1(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： `apic1(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。

	コマンドまたはアクション	目的

ステップ 9	**switchport tenant***tenant-name***dot1q-tunnel** *tunnel-name*
	例：
	`apic1(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_edgetunnel`
	`apic1(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

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 8021 q インターフェイスとして 2 つのポート チャネルにマークし、2 つ以上のポート チャネルをコア ポート 802.1 q 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
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel pc1
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/2-3
apic1(config-leaf-if)# channel-group pc1
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel pc1
apic1(config-leaf-if)# switchport mode dot1q-tunnel edgePort
apic1(config-leaf-if)# switchport tenant tenant64 dot1q-tunnel vrf64_tunnel
apic1(config-tenant-tunnel)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 102
apic1(config-leaf)# interface port-channel pc2
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/4-5
apic1(config-leaf-if)# channel-group pc2
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel pc2
apic1(config-leaf-if)# switchport mode dot1q-tunnel corePort
apic1(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 チャネルをトンネルに関連付けています。

```
apic1# configure
apic1(config)# vpc domain explicit 1 leaf 101 102
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# interface vpc vpc1
apic1(config-vpc-if)# switchport mode dot1q-tunnel edgePort
apic1(config-vpc-if)# exit
apic1(config-vpc)# exit
apic1(config)# vpc domain explicit 1 leaf 103 104
apic1(config)# vpc context leaf 103 104
apic1(config-vpc)# interface vpc vpc2
apic1(config-vpc-if)# switchport mode dot1q-tunnel corePort
apic1(config-vpc-if)# exit
apic1(config-vpc)# 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 103
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)# 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
```

ダイナミック ブレークアウト ポートの設定

ダイナミック ブレークアウト ポートの設定

ブレークアウト ケーブルは非常に短いリンクに適しており、コスト効率の良いラック内および隣接ラック間を接続する方法を提供します。

ブレークアウトでは、40 ギガビット（Gb）ポートを独立して 4 分割し、10Gb または 100Gb ポートを独立した状態で論理 25 Gb ポートに 4 分割できます。

ブレークアウト ポートを設定する前に、次のケーブルのいずれかを使用して 40 Gb ポートを 4 つの 10 Gb ポートまたは 100 Gb ポートを 4 つの 25 Gb ポートに接続します。

- Cisco QSFP-4SFP10G
- Cisco QSFP-4SFP25G

40 Gb から 10 Gb までのダイナミック ブレークアウト機能は、次のスイッチのポートが面しているアクセスでサポートされています。

- N9K-C9332PQ
- N9K-C93180LC-EX
- N9K-C9336C-FX

100 Gb から 25 Gb までの ブレークアウト機能は、次のスイッチのポートが面しているアクセスでサポートされています。

- N9K-C93180LC-EX
- N9K-C9336C-FX2

- N9K-C93180YC-FX

次の注意事項および制約事項を確認します。

- 一般に、ブレーアウトおよびポートプロファイル（アップリンクからダウンリンクへ変更されたポート）は、同一ポートではサポートされません。

  ただし Cisco APIC、リリース 3.2 から、ダイナミック ブレークアウト（100 Gb および 40 Gb の両方）は N9K-C93180YC-FX スイッチのプロファイリングされた QSFP ポートでサポートされます。

- ファスト リンク フェールオーバー ポリシーは、ダイナミック ブレークアウト機能と同一ポートではサポートされていません。
- ブレークアウトのサポートは、ポリシー モデルが使用されているその他のポート タイプと同じ方法で使用できます。
- ポートがダイナミック ブレークアウトに対して有効になっている場合、親ポートのその他のポート（モニタリング ポリシー以外）は無効になります。
- ポートがダイナミック ブレークアウトに対して有効になっている場合、親ポートのその他の EPG 展開が無効になります。
- ブレークアウト サブポートは、ブレークアウト ポリシー グループを使用してもこれ以上分割することはできません。

NX-OS スタイルの CLI を使用したダイナミック ブレイクアウト ポートの設定

ブレークアウトポートを設定、設定を確認および NX-OS スタイル CLI を使用してサブ ポートで、EPG を設定するには、次の手順を使用します。

始める前に

- ACI ファブリックが設置され、APIC コントローラがオンラインになっており、APIC クラスタが形成されて正常に動作していること。
- 必要なファブリック インフラストラクチャ設定を作成できる APIC ファブリック管理者アカウントが使用可能であること。
- ターゲット リーフ スイッチが ACI ファブリックに登録され、使用可能であること。
- 40GE または 100GE リーフ スイッチ ポートは、ダウンリンク ポートに Cisco ブレークアウト ケーブルを接続します。

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	**leaf** *ID* 例： `apic1(config)# leaf 101`	ブレークアウトポートが配置され、リーフ configuration mode(設定モード、コンフィギュレーション モード) を開始リーフ スイッチを選択します。
ステップ 3	**interface ethernet***slot/port* 例： `apic1(config-leaf)# interface ethernet 1/16`	40 ギガビット イーサネット (GE) ブレークアウトポートとして有効にするインターフェイスを識別します。
ステップ 4	**breakout10g-4x|25g-4x** 例： `apic1(config-leaf-if)# breakout 10g-4x`	ブレイク アウトを選択したインターフェイスを有効にします。 （注） ダイナミック ブレイク アウト ポート機能は、スイッチのサポートを参照してください。 ダイナミック ブレークアウト ポートの設定 （43 ページ） 。

ステップ 5	**show run**
	例：
	`apic1(config-leaf-if)# show run`
	`# Command: show running-config leaf 101 interface ethernet 1 / 16`
	`# Time: Fri Dec 2 18:13:39 2016`
	`leaf 101`
	`interface ethernet 1/16`
	`breakout 10g-4x`
	`apic1(config-leaf-if)# exit`
	`apic1(config-leaf)# exit`
		インターフェイスの実行コンフィギュレーションを表示することによって、設定を確認し、グローバル コンフィギュレーション モードに戻ります。

ステップ 6	**tenant** *tenant-name* 例： `apic1(config)# tenant tenant64`	選択またはブレイク アウト ポートで消費され、テナント configuration mode(設定モード、コンフィギュレーション モード) を開始するテナントを作成します。
ステップ 7	**vrf context** *vrf-name* 例：	作成またはテナントに関連付けられている Virtual Routing and Forwarding (VRF) インスタンスを識別し、

	コマンドまたはアクション	目的
	`apic1(config-tenant)# vrf context vrf64` `apic1(config-tenant-vrf)# exit`	configuration mode(設定モード、コンフィギュレーション モード) を終了します。
ステップ 8	**bridge-domain** *bridge-domain-name* 例： `apic1(config-tenant)# bridge-domain bd64`	作成またはテナントに関連付けられているブリッジ ドメインを識別し、BD configuration mode(設定モード、コンフィギュレーション モード) を開始します。

ステップ 9	**vrf member** *vrf-name*
	例：
	`apic1(config-tenant-bd)# vrf member vrf64`
	`apic1(config-tenant-bd)# exit`
		ブリッジ ドメイン、VRF の関連付け、configuration mode(設定モード、コンフィギュレーション モード) を終了します。

ステップ 10	**application** *application-profile-name* 例： `apic1(config-tenant)# application app64`	作成またはテナントと EPG に関連付けられているアプリケーションプロファイルを識別します。
ステップ 11	**epg** *epg-name* 例： `apic1(config-tenant)# epg epg64`	作成または EPG を識別し、EPG configuration mode(設定モード、コンフィギュレーション モード) に入力します。

ステップ 12	**bridge-domain member** *bridge-domain-name*
	例：
	`apic1(config-tenant-app-epg)# bridge-domain member bd64`
	`apic1(config-tenant-app-epg)# exit`
	`apic1(config-tenant-app)# exit`
	`apic1(config-tenant)# exit`
		EPG をブリッジ ドメインに関連付け、グローバル設定モードをに戻ります。
		たとえば、必要に応じて、サブ ポートを設定コマンドを使用して、速度リーフ インターフェイス モードでサブ ポートを設定します。

ステップ 13	**speed** *interface-speed*
	例：
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# interface ethernet 1/16/1`
	`apic1(config-leaf-if)# speed 10G`
	`apic1(config-leaf-if)# exit`
		リーフ インターフェイス モードを開始し、[インターフェイスの速度を設定 configuration mode(設定モード、コンフィギュレーション モード) を終了します。

ステップ 14	**show run** 例： `apic1(config-leaf)# show run`	サブ ポートを設定した後にリーフ 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

	コマンドまたはアクション	目的
		マンドを入力して、サブ ポートの詳細が表示されます。

サブ ポート 1/16/1、2/1/16、1/16/3 および 4/1/16 ブレイク アウトを有効になっているリーフ インターフェイス 1/16 で 101 上のポートを確認します。

例

この例では、ブレイク アウト ポートで設定します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/16
apic1(config-leaf-if)# breakout 10g-4x
```

この例では、サブインターフェイス ポートの EPG で設定します。

```
apic1(config)# tenant tenant64
apic1(config-tenant)# vrf context vrf64
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain bd64
apic1(config-tenant-bd)# vrf member vrf64
apic1(config-tenant-bd)# exit
apic1(config-tenant)# application app64
apic1(config-tenant-app)# epg epg64
apic1(config-tenant-app-epg)# bridge-domain member bd64
apic1(config-tenant-app-epg)# end
```

この例では、10 G に、ブレイク アウトの速度サブ ポートを設定します。

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/16/1
apic1(config-leaf-if)# speed 10G
apic1(config-leaf-if)# exit

apic1(config-leaf)# interface ethernet 1/16/2
apic1(config-leaf-if)# speed 10G
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/16/3
apic1(config-leaf-if)# speed 10G
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/16/4
apic1(config-leaf-if)# speed 10G
apic1(config-leaf-if)# exit
```

この例では、リーフ 101、インターフェイス 1/16 に接続されている、4 つのアシスタント的なポートを示します。

```
apic1#(config-leaf)# show run
# Command: show running-config leaf 101
# Time: Fri Dec  2 00:51:08 2016
  leaf 101
    interface ethernet 1/16/1
      speed 10G
      negotiate auto
      link debounce time 100
      exit
    interface ethernet 1/16/2
      speed 10G
```

```
  negotiate auto
  link debounce time 100
  exit
interface ethernet 1/16/3
  speed 10G
  negotiate auto
  link debounce time 100
  exit
interface ethernet 1/16/4
  speed 10G
  negotiate auto
  link debounce time 100
  exit
interface ethernet 1/16
  breakout 10g-4x
  exit
interface vfc 1/16
```

ポート プロファイルの設定

ポート プロファイルの設定

Cisco APIC リリース 3.1 (1) 以前、アップリンク ポートからダウンリンク ポート、あるいはダウンリンク ポートからアップリンク ポート（ポート プロファイル内）への変換は、Cisco ACI リーフ スイッチではサポートされていませんでした。Cisco APIC リリース 3.1 (1) から、アップリンクおよびダウンリンクの変換は、EX または FX およびそれ以降（たとえば、N9K-C9348GC-FXP）で終わる名前の Cisco Nexus 9000 シリーズ スイッチでサポートされています。変換後のダウンリンクに接続されている FEX もサポートされています。

この機能は次の Cisco スイッチでサポートされています。

- N9K-C9348GC-FXP
- N9K-C93180LC-EX および N9K-C93180YC-FX
- N9K-93180YC-EX、N9K-C93180YC-EX、N9K-C93180YC-EXU
- N9K-C93108TC-EX および N9K-C93108TC-FX
- N9K C9336C FX2（ダウンリンクからアップリンクへの変換のみサポート）

制約事項

FAST リンク フェールオーバー ポリシーとポート プロファイルは、同じポートではサポートされていません。ポート プロファイルが有効になっている場合、FAST リンク フェールオーバーを有効にすることはできません。その逆も同様です。

サポートされている TOR スイッチの最後の 2 つのアップリンク ポートは、ダウンリンク ポート（これらはアップリンク接続用に予約済み）に変換することはできません。

Cisco APIC リリース 3.2 までは、ポート プロファイルとブレイクアウト ポートは同じポートでサポートされていません。

Cisco APIC リリース 3.2 以降では、ダイナミック ブレイクアウト（100 Gb および 40 Gb の両方）が N9K-C93180YC-FX スイッチのプロファイリングされた QSFP ポートでサポートされます。ブレイクアウトおよびポートプロファイルでは、ポート 49-52 でアップリンクからダウンリンクへの変換が一緒にサポートされています。ブレイクアウト（**10 g 4 x** または **25 g 4 x** オプション）は、ダウンリンクのプロファイリングされたポートでサポートされています。

ガイドライン

アップリンクをダウンリンクに変換したり、ダウンリンクをアップリンクに変換したりする際は、次のガイドラインにご注意ください。

サブジェクト	ガイドライン
ポート プロファイルを使用したノードのデコミッション	デコミッションされたノードがポートプロファイル機能を展開している場合、ポート変換はノードのデコミッション後も削除されません。ポートをデフォルト状態に戻すには、デコミッション後に手動で設定を削除する必要があります。これを行うには、スイッチにログオンし、**setup-clean-config.sh** スクリプトを実行して、実行されるまで待ちます。それから、**リロード** コマンドを入力します。
FIPS	Cisco ACI ファブリックで Federal Information Processing Standards（FIPS）を有効または無効にしている場合、変更を有効にするためファブリック内のスイッチをそれぞれリロードする必要があります。FIPS 設定を変更した後に最初のリロードを発行すると、設定されているスケールプロファイル設定が失われます。スイッチは動作を継続しますが、デフォルトのスケール プロファイルを使用します。FIPS 設定が変更されていない場合は、この問題は後続のリロードでは発生しません。 FIPS は Cisco NX-OS リリース 13.1(1) またはそれ以降でさぽーとされています。 FIPS をサポートしているリリースから FIPS をサポートしていないリリースにファームウェアをダウングレードする必要がある場合、最初に Cisco ACI ファブリックで FIPS を無効にして、FIPS 設定の変更のためファブリック内のすべてのスイッチをリロードする必要があります。

サブジェクト	ガイドライン
最大アップリンク ポートの制限	最大アップリンク ポートの制限に達し、ポート 25 および 27 がアップリンクからダウンリンクへ返還されるとき、Cisco 93180LC EX スイッチのアップリンクに戻ります。 Cisco 93180LC EX スイッチでは、ポート 25 および 27 はネイティブのアップリンク ポートです。ポート プロファイルを使用してポート 25 および 27 をダウンリンク ポートに変換する場合は、ポート 29、30、31、32 は、4 つのネイティブ アップリンク ポートとしても使用できます。変換可能なポート数のしきい値のため（最大 12 ポート）、8 個以上のダウンリンク ポートをアップリンク ポートに変換できます。たとえば、ポート 1、3、5、7、9、13、15、17 はアップリンク ポートに変換され、ポート 29、30、31、32 ポートは 4 つのネイティブ アップリンク ポートとなります（Cisco 93180LC EX スイッチで最大のアップリンク ポートの制限）。 スイッチがこの状態でポート プロファイル設定がポート 25 および 27 で削除される場合、ポート 25 および 27 はアップリンク ポートへ再度変換されますが、前述したようにスイッチにはすでに 12 個のアップリンク ポートがあります。ポート 25 および 27 をアップリンク ポートとして適用するため、ポート範囲 1、3、5、7、9、13、15、17 からランダムで 2 個のポートがアップリンクへの変換を拒否されます。この状況はユーザにより制御することはできません。 そのため、リーフ ノードをリロードする前にすべての障害を消去し、ポート タイプに関する予期しない問題を回避することが必須です。ポートプロファイルの障害を消去せずにノードをリロードすると、特に制限超過に関する障害の場合、ポートは予想される動作状態になることに注意する必要があります。

ブレークアウト制限

スイッチ	リリース	制限事項
N9K-C9332PQ	Cisco APIC 2.2 (1n) 以降	• 4X10Gb ポートへの 40Gb ダイナミック ブレーク アウトがサポートされています。 • ポート 13 および 14 はブレークアウトをサポートしていません。 • ポート プロファイルおよびブレークアウトは、同じポートでサポートされていません。

スイッチ	リリース	制限事項
N9K-C93180LC-EX	Cisco APIC 3.1(1i) 以降	• 40 Gb と 100 Gb のダイナミック ブレークアウトは、ポート 1 ~ 24 の奇数ポート上でサポートされます。 • 上位ポート（奇数ポート）ブレークアウトされると、下部ポート（偶数ポート）はエラーが無効になります。 • ポート プロファイルおよびブレークアウトは、同じポートでサポートされていません。
N9K-C9336C-FX2	Cisco APIC 3.1(2m) 以降	• ポート 1 ~ 30 では、40 Gb と 100 Gb のダイナミック ブレークがサポートされています。 • ポート プロファイルおよびブレークアウトは、同じポートでサポートされていません。

スイッチ	リリース	制限事項
N9K-C93180YC-FX	Cisco APIC リリース 3.2(1x) 以降	• 40 Gb と 100 Gb のダイナミック ブレークは、52、上にあるときにプロファイリング QSFP ポートがポート 49 でサポートされます。ダイナミック ブレークアウトを使用するには、次の手順を実行します。 • ポート 49~52 を前面パネル ポート（ダウンリンク）に変換します。 • 次の方法のいずれかを使用して、ポート プロファイルのリロードを実行します。 • APIC GUI で、**[ファブリック]** > **[インベントリ]** > **[ポッド]** > **[リーフ]** に移動し、**[シャーシ]** クリックしてから **[リロード]** を選択します。 • NX-OS スタイル CLI で、**setup-clean-config.sh -k** スクリプトを入力し、実行を待機し、**reload** コマンドを入力します。 • プロファイルされたポート 49 - 52 のブレーク アウトを適用します。 • ポート 53 および 54 では、ポート プロファイルまたはブレークアウトをサポートしていません。

ポート プロファイルの設定のまとめ

次の表では、アップリンクからダウンリンク、ダウンリンクからアップリンクへのポートプロファイルの変換をサポートしているスイッチで、サポートされているアップリンクおよびダウンリンクをまとめています。

スイッチ モデル	デフォルト リンク	最大アップリンク (ファブリック ポート)	最大ダウンリンク (サーバのポート)	サポートされているリリース
N9K-C9348GC-FXP	48 x 100 M/1 G BASE-T ダウンリンク 4 x 10/25 Gbps SFP28 ダウンリンク 2 x 40/100 Gbps QSFP28 アップリンク	48 x 100 M/1 G BASE-T ダウンリンク 4 x 10/25 Gbps SFP28 アップリンク 2 x 40/100 Gbps QSFP28 アップリンク	デフォルトと同じ	3.1(1i)
N9K-C93180LC-EX	24 x 40 Gbps QSFP 28 ダウンリンク 6 x 40/100-Gbps QSFP 28 アップリンク または 12 x 100 Gbps QSFP 28 ダウンリンク 6 x 40/100-Gbps QSFP 28 アップリンク	12 x 40 Gbps QSFP 28 ダウンリンク 12 x 40/100-Gbps QSFP 28 アップリンク または 6 x 100 Gbps QSFP 28 ダウンリンク 12 x 40/100-Gbps QSFP 28 アップリンク	4 x 40 Gbps QSFP 28 ダウンリンク 2 x 40/100-Gbps QSFP 28 ダウンリンク 4 x 40/100-Gbps アップリンク（100 Gbps） または 12 x 100 Gbps QSFP 28 ダウンリンク 2 x 40/100-Gbps QSFP 28 ダウンリンク 4 x 40/100-Gbps アップリンク（100 Gbps）	3.1(1i)
N9K-C93180YC-EX N9K-C93180YC-FX	48 x 10/25 Gbps ファイバ ダウンリンク 6 x 40/100 Gbps QSFP28 アップリンク	48 x 10/25 Gbps ファイバ ダウンリンク 6 x 40/100 Gbps QSFP28 アップリンク	48 x 10/25 Gbps ファイバ ダウンリンク 4 x 40/100 Gbps QSFP28 ダウンリンク 2 x 40/100 Gbps QSFP28 アップリンク	3.1(1i)

スイッチ モデル	デフォルト リンク	最大アップリンク (ファブリック ポート)	最大ダウンリンク (サーバのポート)	サポートされているリリース
N9K-C93108TC-EX N9K-C93108TC-FX	48 x 10GBASE T ダウンリンク 6 x 40/100 Gbps QSFP28 アップリンク	デフォルトと同じ	48 x 10/25 Gbps ファイバ ダウンリンク 4 x 40/100 Gbps QSFP28 ダウンリンク 2 x 40/100 Gbps QSFP28 アップリンク	3.1(1i)
N9K-C9336C-FX2	30 x 40/100 Gbps QSFP28 ダウンリンク 6 x 40/100 Gbps QSFP28 アップリンク	18 x 40/100 Gbps QSFP28 ダウンリンク 18 x 40/100 Gbps QSFP28 アップリンク	デフォルトと同じ	3.1(2m)

NX-OS スタイル CLI を使用したポート プロファイルの設定

NX-OS スタイルの CLI を使用したポート プロファイルの設定をするには、次の手順を実行します。

始める前に

- ACI ファブリックが設置され、APIC コントローラがオンラインになっており、APIC クラスタが形成されて正常に動作していること。
- 必要なファブリック インフラストラクチャ設定を作成または変更できる APIC ファブリック管理者アカウントが使用可能であること。
- ターゲット リーフ スイッチが ACI ファブリックに登録され、使用可能であること。

手順

ステップ 1 **configure**

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。

例：

```
apic1# configure
```

ステップ 2 **leaf** *node-id*

設定するリーフまたはリーフ スイッチを指定します。

例：

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

ステップ 3 **interface** *type*

設定するインターフェイスを指定します。インターフェイス タイプと ID を指定できます。イーサネット ポートの場合は、ethernet *slot* / *port* を使用します。

例：

```
apic1(config-leaf)# interface ethernet 1/2
```

ステップ 4 **port-direction** {**uplink** | **downlink**}

ポートの方向を決定するか変更します。この例ではダウンリンクにポートを設定します。

（注） N9K-C9336C-FX スイッチでは、アップリンクからダウンリンクへの変更はサポートされていません。

例：

```
apic1(config-leaf-if)# port-direction downlink
```

ステップ 5 ポートがあるリーフ スイッチにログインし、**setup-clean-config.sh -k** コマンドを入力してから **reload** コマンドを入力します。

NX-OS スタイル CLI を使用したポート プロファイルの設定と変換の確認

show interface brief CLI コマンドを使用して、ポートの設定と変換を確認することができます。

（注） ポート プロファイルは、Cisco N9K-C93180LC EX スイッチのトップ ポートにのみ展開されます。たとえば、1、3、5、7、9、11、13、15、17、19、21、および 23 となります。ポート プロファイルを使用してトップ ポートを変換すると、ボトム ポートはハードウェア的に無効になります。たとえば、ポート プロファイルを使用して Eth 1/1 を変換すると、Eth 1/2 はハードウェア的に無効になります。

手順

ステップ 1 この例では、アップリンク ポートをダウンリンク ポートに変換する場合の出力を示しています。アップリンク ポートをダウンリンク ポートに変換変換する前に、この例での出力が表示

されます。**routed** というキーワードは、ポートがアップリンク ポートであることを示しています。

例：

```
switch# show interface brief
<snip>
Eth1/49          --      eth  routed  down    sfp-missing              100G(D)   --
Eth1/50          --      eth  routed  down    sfp-missing              100G(D)   --
<snip>
```

ステップ 2 ポートプロファイルを設定して、スイッチのリロード、後に、例では、出力が表示されます。キーワード **トランク** ダウンリンク ポートとしてポートを示します。

例：

```
switch# show interface brief
<snip>
Eth1/49          0       eth  trunk   down    sfp-missing              100G(D)   --
Eth1/50          0       eth  trunk   down    sfp-missing              100G(D)   --
<snip>
```

仮想スイッチ上のマイクロセグメンテーション

仮想スイッチでのマイクロセグメンテーションの設定

Cisco Application Centric Infrastructure（ACI）マイクロセグメンテーションは、さまざまなネットワークベースまたは仮想マシン（VM）ベース属性に基づき、エンドポイント グループ（EPG）と呼ばれるロジカル セキュリティ ゾーンにエンドポイントを自動的に割り当てることができます。このセクションでは、仮想スイッチのマイクロセグメント（uSeg）EPG を設定する方法について説明します。

Cisco ACI マイクロセグメンテーションでは、次に接続されている仮想エンドポイントのサポートを提供します。

- VMware vSphere Distributed Switch（vDS）
- Cisco Application Virtual Switch（AVS）
- Microsoft vSwitch

マイクロセグメンテーションと Cisco ACI の動作、前提条件、ガイドライン、およびシナリオについての詳細は「*Cisco ACI 仮想化ガイド*」を参照してください。

NX-OS スタイル CLI を使用した Cisco ACI でのマイクロセグメンテーションの設定

ここでは、アプリケーション EPG 内で VM ベースの属性を使用して Cisco ACI for Cisco ACI Virtual Edge、Cisco AVS、VMware VDS、または Microsoft vSwitch でマイクロセグメンテーションを設定する方法について説明します。

手順

ステップ 1 CLI で、コンフィギュレーション モードに入ります。

例：

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

ステップ 2 USeg EPG を作成します。

例：

この例は、アプリケーション EPG のためのものです。

（注） 次の例のマイクロセグメンテーションを許可するコマンドが VMware VDS にのみ必要です。

```
apic1(config)# tenant cli-ten1
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-baseEPG1
apic1(config-tenant-app-epg)# bridge-domain member cli-bd1
apic1(config-tenant-app-epg)# vmware-domain member cli-vmm1 allow-micro-segmentation
```

例：

（オプション）この例の設定は、uSeg EPG の EPG の優先順位と一致します。:

```
apic1(config)# tenant Coke
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented
apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1
apic1(config-tenant-app-uepg)# match-precedence 10
```

例：

この例では、属性 VM 名に基づいてフィルタを使用します。

```
apic1(config)# tenant cli-ten1
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented
apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1
apic1(config-tenant-app-uepg)# attribute-logical-expression 'vm-name contains <cos1>'
```

例：

この例では、IP アドレスに基づいてフィルタを使用します。

```
apic1(config)# tenant cli-ten1
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented
apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1
apic1(config-tenant-app-uepg)# attribute-logical-expression 'ip equals <FF:FF:FF:FF:FF:FF>'
```

例：

この例では、MAC アドレスに基づいてフィルタを使用します。

```
apic1(config)# tenant cli-ten1
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented
apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1
apic1(config-tenant-app-uepg)# attribute-logical-expression 'mac equals
<FF-FF-FF-FF-FF-FF>'
```

例：

この例では、演算子 AND を使用してすべての属性を一致させるか、演算子 OR を使用してすべての属性を一致させます。

```
apic1(config)# tenant cli-ten1
apic1(config-tenant)# application cli-a1
apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented
apic1(config-tenant-app-uepg)# attribute-logical-expression 'hv equals host-123 OR
(guest-os equals "Ubuntu Linux (64-bit)" AND domain contains fex)'
```

ステップ 3 (Cisco ACI Virtual Edge のみ）：uSeg EPG を Cisco ACI Virtual Edge VMM ドメインに接続し、スイッチングおよびカプセル化モードを指定します。

例：

```
vmware-domain member AVE-CISCO
    switching-mode AVE
    encap-mode vxlan
    exit
```

ステップ 4 USeg EPG の作成を確認します。

例：

次の例は、VM 名属性フィルタを持つ uSeg EPG のためのものです。

```
apic1(config-tenant-app-uepg)# show running-config
# Command: show running-config tenant cli-ten1 application cli-a1 epg cli-uepg1 type
micro-segmented # Time: Thu Oct 8 11:54:32 2015
  tenant cli-ten1
    application cli-a1
      epg cli-uepg1 type micro-segmented
        bridge-domain cli-bd1
        attribute-logical-expression 'vm-name contains cos1 force'
        {vmware-domain | microsoft-domain} member cli-vmm1
        exit
     exit
exit
```

ベア メタル上のマイクロセグメンテーションの設定

ベア メタルでのネットワークベースの属性によるマイクロセグメンテーションの使用

Cisco APIC を使用して Cisco ACI でのマイクロセグメンテーションを設定し、ネットワークベースの属性、MAC アドレス、または 1 つ以上の IP アドレスを使用した新しい属性ベースの EPG を作成できます。ネットワークベースの属性を使用して Cisco ACI でのマイクロセグメンテーションを設定し、単一のベース EPG または複数の EPG 内で VM または物理エンドポイントを分離できます。

IP ベースの属性の使用

IP ベースのフィルタを使用して、単一のマイクロセグメントで単一 IP アドレス、サブネット、または多様な非連続 IP アドレスを分離できます。ファイアウォールの使用と同様に、セキュリティ ゾーンを作成するための迅速かつ簡単な方法として、IP アドレスに基づいて物理エンドポイントを分離できます。

MAC ベースの属性の使用

MAC ベースのフィルタを使用して、単一 MAC アドレスまたは複数の MAC アドレスを分離できます。不適切なトラフィックをネットワークに送信するサーバがある場合はこの方法を推奨します。MAC ベースのフィルタを使用してマイクロセグメントを作成することで、このサーバを分離できます。

NX-OSスタイルのCLIを使用したベアメタル環境でのネットワークベースのマイクロセグメント EPG の設定

ここでは、ベアメタル環境のベース EPG 内で、ネットワークベースの属性（IP アドレスまたは MAC アドレス）を使用して Cisco ACI でマイクロセグメンテーションを設定する方法について説明します。

手順

	コマンドまたはアクション	目的

ステップ 1	CLI で、コンフィギュレーション モードに入ります。
	例：
	`apic1# configure`
	`apic1(config)#`	

ステップ 2	マイクロセグメントを作成します。	

	コマンドまたはアクション	目的

	例：
	この例では、IPアドレスに基づいてフィルタを使用します。
	`apic1(config)# tenant cli-ten1`
	`apic1(config-tenant)# application cli-a1`
	`apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented`
	`apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1`
	`apic1(config-tenant-app-uepg)# attribute cli-upg-att match ip <X.X.X.X>`
	`#Schemes to express the ip`
	`A.B.C.D IP Address`
	`A.B.C.D/LEN IP Address and mask`
	例：
	この例では、MAC アドレスに基づいてフィルタを使用します。
	`apic1(config)# tenant cli-ten1`
	`apic1(config-tenant)# application cli-a1`
	`apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented`
	`apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1`
	`apic1(config-tenant-app-uepg)# attribute cli-upg-att match mac <FF-FF-FF-FF-FF-FF>`
	`#Schemes to express the mac`
	`E.E.E MAC address (Option 1)`
	`EE-EE-EE-EE-EE-EE MAC address (Option 2)`
	`EE:EE:EE:EE:EE:EE MAC address (Option 3)`
	`EEEE.EEEE.EEEE MAC address (Option 4)`
	例：
	この例では、MAC アドレスに基づいてフィルタを使用し、この uSeg EPG のすべてのメンバー間に EPG 間分離を適用します。
	`apic1(config)# tenant cli-ten1`
	`apic1(config-tenant)# application cli-a1`
	`apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented`
	`apic1(config-tenant-app-uepg)# isolation enforced`
	`apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1`
	`apic1(config-tenant-app-uepg)# attribute cli-upg-att match mac`	

	コマンドまたはアクション	目的
	<FF-FF-FF-FF-FF-FF> #Schemes to express the mac E.E.E MAC address (Option 1) EE-EE-EE-EE-EE-EE MAC address (Option 2) EE:EE:EE:EE:EE:EE MAC address (Option 3) EEEE.EEEE.EEEE MAC address (Option 4)	
ステップ 3	EPG を導入します。 **例**： この例では、EPG を導入してリーフを指定します。 apic1(config)# leaf 101 apic1(config-leaf)# deploy-epg tenant cli-ten1 application cli-a1 epg cli-uepg1 type micro-segmented	
ステップ 4	マイクロセグメントの作成を確認します。 **例**： apic1(config-tenant-app-uepg)# show running-config # Command: show running-config tenant cli-ten1 application cli-app1 epg cli-uepg1 type micro-segmented # Time: Thu Oct 8 11:54:32 2015 tenant cli-ten1 application cli-app1 epg cli-esx1bu type micro-segmented bridge-domain cli-bd1 attribute cli-uepg-att match mac 00:11:22:33:44:55 exit exit exit	

レイヤ 2 IGMP スヌープ マルチキャストの設定

Cisco APIC と IGMP スヌーピングについて

IGMP スヌーピングは、Internet Group Management Protocol（IGMP）ネットワーク トラフィックをリスニングするプロセスです。機能は、ホストとルータおよびフィルタ マルチキャスト リンク、する必要はありませんがどのポートが特定のマルチキャスト トラフィックを受信を制御するための間での IGMP 対話リッスンするようにネットワーク スイッチを使用できます。

Cisco APIC は、N9000 スタンドアロンなどの従来のスイッチに含まれるフル IGMP スヌーピング機能のサポートを提供します。

- ポリシー ベース IGMP スヌーピングごとの設定ブリッジ ドメイン

  APIC を使用することを有効にする、無効にすると、またはブリッジ ドメイン単位で IGMP スヌーピングのプロパティをカスタマイズ ポリシーを設定することができます。1 つまたは複数のブリッジ ドメインへのポリシーを適用することができます。

- 静的ポート グループの実装

  IGMP スタティック ポートのグループ化を使用すると、事前プロビジョニング、すでに静的に - に割り当てられたポートを受信し、IGMP を処理するスイッチ ポートとしてアプリケーション EPG、マルチキャスト トラフィック。この事前のプロビジョニングには、通常は IGMP スヌーピング スタック ポートを動的に学習するときに参加遅延ができなくなります。

  静的ポートでのみスタティック グループ メンバーシップを事前にプロビジョニングされたできます (とも呼ばれる、 スタティック バインディング ポート) アプリケーション EPG に割り当てられます。

- アプリケーション Epg のアクセス グループの設定

  「アクセス-グループ」ができるストリームを制御するために使用任意ポート背後に参加します。

  実際に所属するするポートの設定を適用できることを確認するには、アプリケーション EPG に静的に割り当てられているインターフェイスでアクセス グループ設定を適用できる EPG。

  ルート マップ ベースのアクセス グループのみが許可されます。

（注） 使用することができます **vzAny** 、VRF 内のすべての Epg の IGMP スヌーピングなどのプロトコルをイネーブルにします。詳細については **vzAny** を参照してください VRF で通信ルールを自動的にすべての Epg に適用する vzAny を使用して 。

使用する **vzAny** に移動 **テナント** > **テナント名** > **ネットワー キング** > **Vrf** > ***vrf* 名** > **VRF の EPG コレクション** 。

静的ポート グループの IGMP スヌーピングを有効にする

IGMP 静的ポートのグループ化により以前アプリケーション EPG に静的に割り当てられた事前プロビジョニングを有効にして、スイッチ ポートが IGMP マルチキャスト トラフィックを受信および処理できます。この事前プロビジョニングは、通常 IGMP スヌーピング スタックがポートを動的に学習するときに発生する参加遅延を防止します。

静的グループ メンバーシップは、アプリケーション EPG に割り当てられている静的ポートでのみ事前プロビジョニングできます。

APIC GUI、CLI、および REST API インターフェイスを通じて、静的グループ メンバーシップを設定できます。

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	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。
	例：
	`apic1(config-tenant)# template ip igmp snooping policy cookieCut1`
	`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 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 stqrtup-query-count 2`
	`ip igmp snooping startup-query-interval 31`
	`no description`
	`exit`
	`exit`
	`apic1(config-tenant-template-ip-igmp-snooping)#`
		例の NX-OS スタイル CLI シーケンス：
		• 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。

ステップ 2	必要に応じてスヌーピング ポリシーを変更します。 **例：** `apic1(config-tenant-template-ip-igmp-snooping)#`	例の NX-OS スタイル CLI シーケンス： • 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。

	コマンドまたはアクション	目的
	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 stqrtup-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)#	• 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	ブリッジ ドメインにポリシーを割り当てます。 **例：** **apic1(config-tenant)# int bridge-domain bd3** **apic1(config-tenant-interface)# ip igmp snooping policy cookieCut1**	例の NX-OS スタイル CLI シーケンス： • 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：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

  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、epg_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、epg_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） スタティック ポートの割り当てに関する詳細は、「*Cisco APIC* レイヤ *3* 設定ガイド」の「*NX-OS* スタイル *CLI* を使用した *APIC* で特定のポートの *EPG* を展開する」を参照してください。
- 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	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マルチキャスティングを有効にします 例：	例のシーケンスでは次を有効にします。 • 静的にリンクされているターゲット インターフェイス 1/10 の IGMP スヌーピング、そしてマルチキャスト

	コマンドまたはアクション	目的
	apic1# conf t apic1(config)# tenant tenant_A apic1(config-tenant)# application application_A apic1(config-tenant-app)# epg epg_A 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	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、そしてマルチキャスト IP アドレス、227.1.1.1 に関連付けます

IGMP スヌープ アクセス グループの有効化

「アクセス-グループ」ができるストリームを制御するために使用任意ポート背後に参加します。

実際に所属するするポートの設定を適用できることを確認するには、アプリケーション EPG に静的に割り当てられているインターフェイスでアクセス グループ設定を適用できる EPG。

ルート マップ ベースのアクセス グループのみが許可されます。

APIC GUI、CLI、および REST API インターフェイスを通じて、IGMP スヌープ アクセス グループを設定できます。

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

  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、epg_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、epg_A に静的にリンクされています。

（注） スタティック ポートの割り当てに関する詳細は、「*Cisco APIC* レイヤ *2* 設定ガイド」の「*NX-OS* スタイル *CLI* を使用した *APIC* で特定のポートの *EPG* を展開する」を参照してください。

手順

	コマンドまたはアクション	目的

ステップ 1	route-map「アクセス グループ」を定義します。
	例：
	`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`
		例のシーケンスを設定します。
		• 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	ルート マップ設定を確認します。
	例：
	`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`	

	コマンドまたはアクション	目的

ステップ 3	アクセス グループ接続パスを指定します。
	例：
	`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`
		例のシーケンスを設定します。
		• 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	アクセスグループ接続を確認します。
	例：
	`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 tenent_A`
	`application application_A`
	`epg epg_A`
	`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`	

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

手順

ステップ 1 VLAN ドメインを設定します。

例：

```
apic1(config)# vlan-domain dom1
apic1(config-vlan)# vlan 10-100
```

ステップ 2 テナントを作成します。

例：

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

ステップ 3 プライベート ネットワーク/VRF を作成します。

例：

```
apic1(config-tenant)# vrf context ctx1
apic1(config-tenant-vrf)# exit
```

ステップ 4 ブリッジ ドメインを作成します。

例：

```
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member ctx1
apic1(config-tenant-bd)# exit
```

ステップ 5 アプリケーション プロファイルおよびアプリケーション EPG を作成します。

例：

```
apic1(config-tenant)# application AP1
apic1(config-tenant-app)# epg EPG1
apic1(config-tenant-app-epg)# bridge-domain member bd1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
```

ステップ 6 EPG を特定のポートに関連付けます。

例：

```
apic1(config)# leaf 1017
apic1(config-leaf)# interface ethernet 1/13
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 20 tenant t1 application AP1 epg
EPG1
```

（注） 上の例に示した vlan-domain コマンドと vlan-domain member コマンドは、ポートに EPG を導入するための前提条件です。

ポート セキュリティの設定

ポート セキュリティと ACI について

ポート セキュリティ機能は、ポートごとに取得される MAC アドレスの数を制限することによって、不明な MAC アドレスでフラッディングしないように ACI ファブリックを保護します。ポート セキュリティ機能のサポートは、物理ポート、ポート チャネル、および仮想ポート チャネルで使用できます。

ポート セキュリティに関するガイドラインと制約事項

次のようなガイドラインと制約事項があります。

- ポート セキュリティは、ポートごとに使用できます。
- ポートセキュリティは、物理ポート、ポートチャネル、および仮想ポートチャネル（vPC）でサポートされています。
- スタティック MAC アドレスとダイナミック MAC アドレスがサポートされています。
- セキュアなポートからセキュアでないポートへと、セキュアでないポートからセキュアなポートへの MAC アドレスの移動がサポートされています。
- MAC アドレスの制限は、MAC アドレスにのみ適用され、MAC と IP によるアドレスには実行されません。
- ポート セキュリティは、ファブリック エクステンダ（FEX）ではサポートされていません。

ポート レベルでのポート セキュリティ

APIC では、ユーザがスイッチ ポートのポート セキュリティを設定できます。ポート上で MAC が制限の最大設定値を超過すると、超過した MAC アドレスからすべてのトラフィックが転送されます。次の属性がサポートされます。

- **ポート セキュリティのタイムアウト** ：現在サポートされているタイムアウト値値は、60 ~ 3600 秒の範囲でサポートされています。
- **違反行為**：違反行為は保護モードで使用できます。保護モードでは、MAC の取得が無効になるため、MAC アドレスは CAM テーブルに追加されません。Mac ラーニングが設定されているタイムアウト値の後に再度有効になります。
- **最大エンドポイント** ：現在のサポートされている最大のエンドポイント設定値は、0 ～ 12000 の範囲でサポートされています。最大エンドポイント値が 0 の場合、そのポートではポート セキュリティ ポリシーが無効になります。

ポート セキュリティ ポリシー グループ テンプレートの設定

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	**[no] template policy-group** *policy-group-name* 例： `apic1(config)# template policy-group PortSecGrp1`	ポリシー グループ テンプレートを作成（または削除）します。
ステップ 3	**[no] switchport access vlan** *vlan-id* **tenant** *tenant-name* **application** *application-name* **epg** *epg-name* 例： `apic1(config-pol-grp-if)# switchport access vlan 4 tenant ExampleCorp application Web epg webEpg`	
ステップ 4	**[no] switchport port-security maximum** *number-of-addresses* 例： `apic1(config-pol-grp-if)# switchport port-security maximum 1`	ポートのセキュア MAC アドレスの最大数を設定します。範囲は 0 ～ 12000 アドレスです。デフォルトは 1 アドレスです。
ステップ 5	**[no] switchport port-security violation protect** 例： `apic1(config-pol-grp-if)# switchport port-security violation protect`	セキュリティ違反が検出された場合に実行するアクションを設定します。**protect** アクションは、十分な数のセキュア MAC アドレスを削除して最大値を下回るまで、不明な送信元アドレスのパケットをドロップします。
ステップ 6	**exit** 例： `apic1(config-pol-grp-if)# exit`	グローバル コンフィギュレーション モードに戻ります。

例

次に、ポートセキュリティポリシーグループテンプレートを作成する例を示します。

```
apic1# configure
apic1(config)# template policy-group PortSecGrp1
apic1(config-pol-grp-if)# switchport port-security maximum 20
apic1(config-pol-grp-if)# switchport port-security violation protect
apic1(config-pol-grp-if)# exit
```

次のタスク

インターフェイスにポート セキュリティ テンプレートを適用します。

テンプレートを使用したインターフェイスでのポートセキュリティの設定

始める前に

ポート セキュリティ ポリシー グループ テンプレートを作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**leaf** *node-id* 例： `apic1(config)# leaf 101`	設定するリーフを指定します。
ステップ 3	**interface** *type-or-range* 例： `apic1(config-leaf)# interface eth 1/2-4`	設定するポートまたはポートの範囲を指定します。
ステップ 4	**[no] policy-group** *policy-group-name* 例： `apic1(config-leaf-if)# policy-group PortSecGrp1`	ポートまたはポートの範囲にポリシー グループ テンプレートを適用します。

例

次に、イーサネット ポートの範囲にポート セキュリティ ポリシー グループ テンプレートを適用する例を示します。

```
apic1# configure
```

```
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/2-4
apic1(config-leaf-if)# policy-group PortSecGrp1
```

次に、テンプレートを使用してポートチャネルでポートセキュリティを設定する例を示します。

```
apic1# configure
apic1(config)# template port-channel po1
apic1(config-if)# switchport port-security maximum 10
apic1(config-if)# switchport port-security violation protect
apic1(config-if)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/3-4
apic1(config-leaf-if)# channel-group po1
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

オーバーライドを使用したインターフェイスでのポートセキュリティの設定

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	**leaf** *node-id* 例： `apic1(config)# leaf 101`	設定するリーフを指定します。
ステップ 3	**interface** *type-or-range* 例： `apic1(config-leaf)# interface eth 1/2-4`	設定するインターフェイスまたはインターフェイスの範囲を指定します。
ステップ 4	**[no] switchport port-security maximum** *number-of-addresses* 例： `apic1(config-leaf-if)# switchport port-security maximum 1`	インターフェイスのセキュアMACアドレスの最大数を設定します。範囲は0～12000アドレスです。デフォルトは1アドレスです。
ステップ 5	**[no] switchport port-security violation protect** 例：	セキュリティ違反が検出された場合に実行するアクションを設定します。**protect** アクションは、十分な数のセキュアMACアドレスを削除して最大値を下回

	コマンドまたはアクション	目的
	`apic1(config-leaf-if)# switchport port-security violation protect`	るまで、不明な送信元アドレスのパケットをドロップします。

例

次に、イーサネットインターフェイスでポートセキュリティを設定する方法を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/2
apic1(config-leaf-if)# switchport port-security maximum 10
apic1(config-leaf-if)# switchport port-security violation protect
```

次に、ポート チャネルでポート セキュリティを設定する例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po2
apic1(config-leaf-if)# switchport port-security maximum 10
apic1(config-leaf-if)# switchport port-security violation protect
```

次に、仮想ポートチャネル（VPC）でポートセキュリティを設定する例を示します。

```
apic1# configure
apic1(config)# vpc domain explicit 1 leaf 101 102
apic1(config-vpc)# exit
apic1(config)# template port-channel po4
apic1(config-if)# exit
apic1(config)# leaf 101-102
apic1(config-leaf)# interface eth 1/11-12
apic1(config-leaf-if)# channel-group po4 vpc
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# interface vpc po4
apic1(config-vpc-if)# switchport port-security maximum 10
apic1(config-vpc-if)# switchport port-security violation protect
```

802.1 x ポートおよびノードの認証

802.1 x ポートおよびノードの認証

IEEE 802.1x はポートベースの認証メカニズムで、不正なデバイスがネットワークにアクセスするのを防止します。NX-OS スタイル CLI を使用して、802.1 x ポートおよびノードの認証を設定することができます。

ポート認証 ポリシーの設定

手順

ステップ 1 CLI で、コンフィギュレーション モードに入ります。

例：

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

ステップ 2 ポリシー グループを作成します:

例：

```
apic1(config)# template policy-group mypol
```

ステップ 3 ポリシー グループでポート レベル認証ポリシーを設定します。

例：

```
apic1(config-pol-grp-if)# switchport port-authentication mydot1x
```

ステップ 4 ホストモードを設定します（2個のモードがサポートされます：マルチホストおよび単一ホスト：単一ホストがデフォルトでは設定されています）。

例：

```
apic1(config-port-authentication)# host-mode multi-host
```

ステップ 5 このポリシーを有効にします（ポリシーはデフォルトでは無効です）。

例：

```
apic1(config-port-authentication)# no shutdown
apic1(config-port-authentication)# exit
apic1(config-pol-grp-if)# exit
apic1(config)#
```

ステップ 6 リーフ インターフェイス ポリシーを設定します。

例：

```
apic1(config)#leaf-interface-profile myprofile
```

ステップ 7 リーフ スイッチ インターフェイス プロファイルのポリシー グループを設定します。

例：

```
apic1(config-leaf-if-profile)#leaf-interface-group mygroup
```

ステップ 8 インターフェイス グループのポートおよびインターフェイスを指定します。

例：

```
apic1(config-leaf-if-group)# interface ethernet 1/10-12
```

ステップ 9 インターフェイス グループにポリシーを適用します。

例：

```
apic1(config-leaf-if-group)# policy-group mypol
apic1(config-leaf-if-group)# exit
apic1(config-leaf-if-profile)# exit
```

ステップ 10 リーフ プロファイルを設定します。

例：

```
apic1(config)#
apic1(config)# leaf-profile myleafprofile
```

ステップ 11 リーフ ポリシー グループを設定し、グループのリーフ スイッチ ノードを指定します。

例：

```
apic1(config-leaf-profile)# leaf-group myleafgrp
apic1(config-leaf-group)# leaf 101
apic1(config-leaf-group)# exit
```

ステップ 12 リーフ スイッチ プロファイルにインターフェイス ポリシーを適用します。

例：

```
apic1(config-leaf-profile)# leaf-interface-profile myprofile
apic1(config-leaf-group)# exit
apic1(config)#
```

ノード認証ポリシーの設定

手順

ステップ 1 CLI で、コンフィギュレーション モードに入ります。

例：

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

ステップ 2 Radius 認証グループを設定します。

例：

```
apic1(config)# aaa group server radius myradiusgrp
 apic1(config-radius)#server 192.168.0.100 priority 1
apic1(config-radius)#exit
```

ステップ 3 ノード レベル ポート認証ポリシーを設定します。

例：

```
apic1(config)# policy-map type port-authentication mydot1x
apic1(config-pmap-port-authentication)#radius-provider-group myradiusgrp
```

ステップ 4 [オプション]認証が失敗した場合は、デフォルトの VLAN ID をオーバーライドします。:

例：

```
apic1(config-pmap-port-authentication)#fail-auth-vlan 2001
```

ステップ 5 [オプション]認証が失敗した場合は、デフォルト EPG をオーバーライドします。

例：

```
apic1(config-pmap-port-authentication)#fail-auth-epg tenant tn1 application ap1 epg
epg256
apic1(config)# exit
```

ステップ 6 ポリシー グループを設定し、グループ内でポート認証ポリシーを指定します。

例：

```
apic1(config)#template leaf-policy-group lpg2
apic1(config-leaf-policy-group)# port-authentication mydot1x
apic1(config-leaf-policy-group)#exit
```

ステップ 7 リーフ スイッチ プロファイルを設定します。

例：

```
apic1(config)# leaf-profile mylp2
```

ステップ 8 リーフ スイッチのプロファイルのグループを設定し、ポリシー グループを指定します。

例：

```
apic1(config-leaf-profile)#leaf-group mylg2
apic1(config-leaf-group)#  leaf-policy-group lpg2
apic1(config-leaf-group)#exit
```

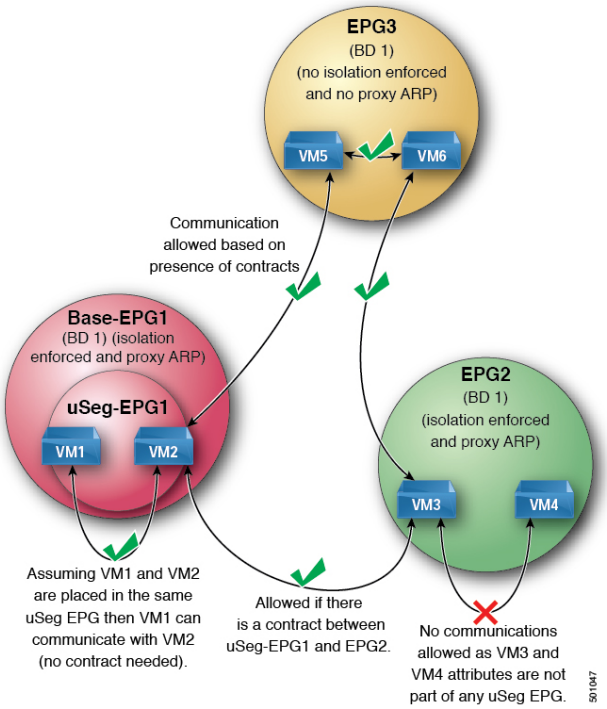
プロキシ ARP の設定

プロキシ ARP について

Cisco ACI のプロキシ ARP は、ネットワークまたはサブネット内のエンドポイントが、別のエンドポイントの MAC アドレスを知らなくても、そのエンドポイントと通信できるようにします。プロキシ ARP はトラフィックの宛先場所を知っており、代わりに、最終的な宛先として自身の MAC アドレスを提供します。

プロキシ ARP を有効にするには、EPG 内エンドポイント分離を EPG で有効にする必要があります。詳細については、次の図を参照してください。EPG 内エンドポイント分離と Cisco ACI の詳細については、「*Cisco ACI* 仮想化ガイド」を参照してください。

図 *3:* プロキシ *ARP* および *Cisco APIC*



Cisco ACI ファブリック内のプロキシ ARP は従来のプロキシ ARP とは異なります。通信プロセスの例として、プロキシ ARP が EPG で有効になっているとき、エンドポイント A が ARP 要求をエンドポイント B に送信し、エンドポイント B がファブリック内で学習される場合、エンドポイント A はブリッジ ドメイン（BD）MAC からプロキシ ARP 応答を受信します。エンドポイント A が B、エンドポイントの ARP 要求を送信し、エンドポイント B はすでに ACI ファブリック内で学習しない場合は、ファブリックはプロキシ ARP の BD 内で要求を送信します。エンドポイント B は、ファブリックに戻る要求、このプロキシ ARP に応答します。この時点では、ファブリックはプロキシ ARP エンドポイント A への応答を送信しませんが、エンドポイント B は、ファブリック内で学習します。エンドポイント A は、エンドポイント B に別の ARP 要求を送信する場合、ファブリックはプロキシ ARP 応答から送信 BD mac です。

次の例ではプロキシ ARP 解像度がクライアント VM1 と VM2 間の通信の手順します。

1. VM2 通信を VM1 が必要です。

図 *4 : VM2* 通信を *VM1* が必要です。

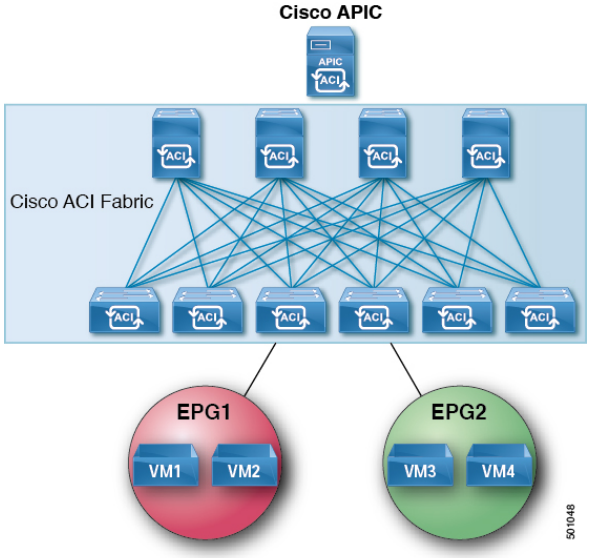


表 *2 : ARP* 表の説明

デバイス	状態
VM1	IP = * MAC = *
ACI ファブリック	IP = * MAC = *
VM2	IP = * MAC = *

2. VM1 は、ブロードキャスト MAC アドレスとともに ARP 要求を VM2 に送信します。

図 *5 : VM1* はブロードキャスト *MAC* アドレスとともに *ARP* 要求を *VM2* に送信します

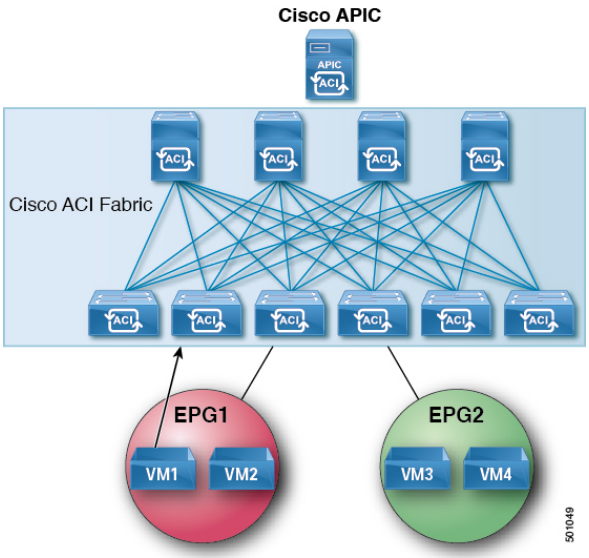


表 *3 : ARP* 表の説明

デバイス	状態
VM1	IP = VM2 IP; MAC = ?
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC
VM2	IP = * MAC = *

3. ACI ファブリックは、ブリッジ ドメイン（BD）内のプロキシ ARP 要求をフラッディングします。

図 *6 : ACI* ファブリックは *BD* 内のプロキシ *ARP* 要求をフラッディングします

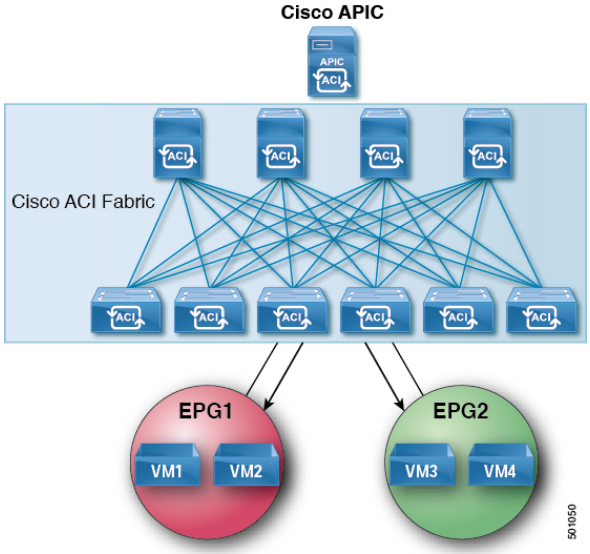


表 *4 : ARP* 表の説明

デバイス	状態
VM1	IP = VM2 IP; MAC = ?
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC
VM2	IP = VM1 IP; MAC = BD MAC

4. VM2 は、ARP 応答を ACI ファブリックに送信します。

図 *7 : VM2* は *ARP* 応答を *ACI* ファブリックに送信します

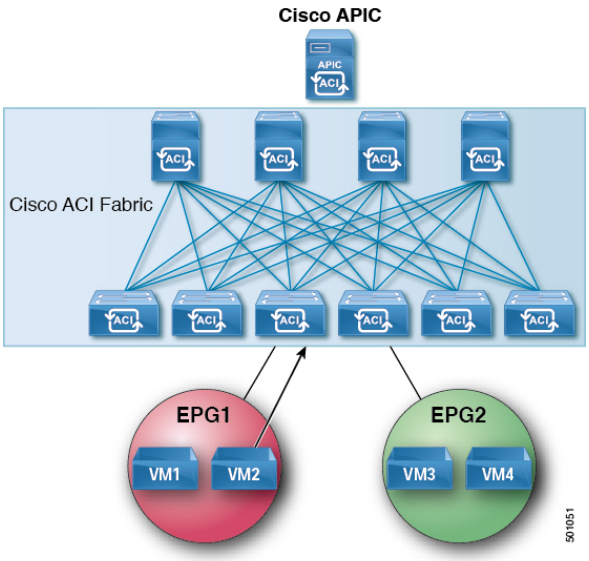


表 *5 : ARP* 表の説明

デバイス	状態
VM1	IP = VM2 IP; MAC = ?
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC
VM2	IP = VM1 IP; MAC = BD MAC

5. VM2 が学習されます。

図 *8 : VM2* が学習されます

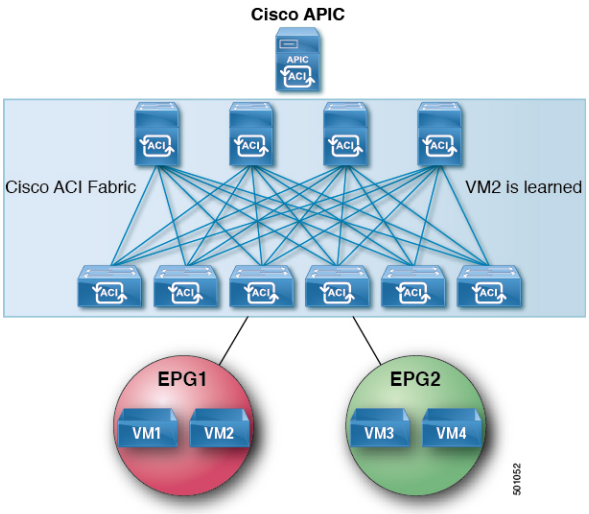


表 *6 : ARP* 表の説明

デバイス	状態
VM1	IP = VM2 IP; MAC = ?
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC IP = VM2 IP; MAC = VM2 MAC
VM2	IP = VM1 IP; MAC = BD MAC

6. VM1 は、ブロードキャスト MAC アドレスとともに ARP 要求を VM2 に送信します。

図 *9 : VM1* はブロードキャスト *MAC* アドレスとともに *ARP* 要求を *VM2* に送信します

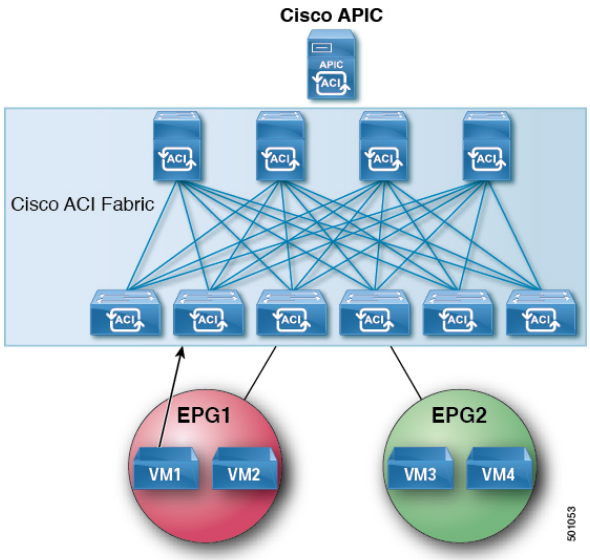


表 *7 : ARP* 表の説明

デバイス	状態
VM1	IP = VM2 IP MAC = ?
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC IP = VM2 IP; MAC = VM2 MAC
VM2	IP = VM1 IP; MAC = BD MAC

7. ACI ファブリックは、プロキシ ARP VM1 への応答を送信します。

図 *10 : ACI* ファブリック *VM1* にプロキシ *ARP* 応答を送信します。

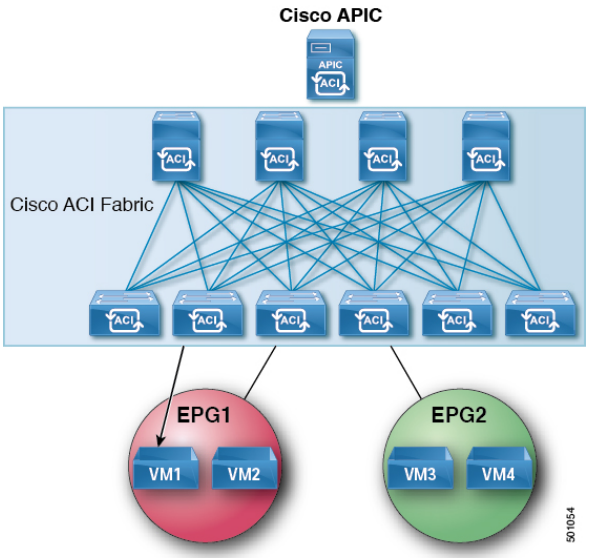


表 *8 : ARP* 表の説明

デバイス	状態
VM1	IP = VM2 IP;MAC = BD MAC
ACI ファブリック	IP = VM1 IP; MAC = VM1 MAC IP = VM2 IP; MAC = VM2 MAC
VM2	IP = VM1 IP; MAC = BD MAC

注意事項と制約事項

プロキシ ARP を使用すると、次のガイドラインと制限事項を考慮してください。

- プロキシ ARP は、隔離 Epg でのみサポートされます。EPG が隔離ではない場合、障害が発生します。プロキシ ARP が有効になっていると隔離 Epg 内で発生する通信では、uSeg Epg を設定する必要があります。たとえば、隔離の EPG 内で別の IP アドレスを持つ複数の Vm がある可能性があり、これらの Vm の IP address range(IP アドレス範囲、IP アドレスの範囲) に一致する IP の属性を持つ uSeg EPG を設定することができます。
- 隔離されたエンドポイントを通常のエンドポイントと、定期的なエンドポイントを隔離のエンドポイントからの ARP 要求には、プロキシ ARP は使用しないでください。このような場合は、エンドポイントは、接続先の Vm の実際の MAC アドレスを使用して通信します。

プロキシ ARP は、Cisco NX-OS スタイル CLI を使用しての設定

始める前に

- 適切なテナント、VRF、ブリッジ ドメイン、アプリケーションプロファイルおよび EPG を作成する必要があります。
- プロキシ ARP が有効にするのにが EPG で内通 EPG の分離を有効にする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure** 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	**tenant** *tenant-name* 例： `apic1(config)# tenant Tenant1`	テナント コンフィギュレーション モードを開始します。
ステップ 3	**application** *application-profile-name* 例： `apic1(config-tenant)# application Tenant1-App`	アプリケーション プロファイルを作成し、アプリケーション モードを開始します。
ステップ 4	**epg** *application-profile-EPG-name* 例： `apic1(config-tenant-app)# epg Tenant1-epg1`	EPG を作成し、EPG モードに入ります。
ステップ 5	**proxy-arp enable** 例： `apic1(config-tenant-app-epg)# proxy-arp enable`	プロキシ ARP を有効にします。 （注） プロキシ arp をディセーブルにできます、 **no プロキシ arp** コマンド。
ステップ 6	**exit** 例： `apic1(config-tenant-app-epg)# exit`	ポート アプリケーション モードに戻ります。
ステップ 7	**exit** 例： `apic1(config-tenant-app)# exit`	テナント コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 8	**exit** 例： `apic1(config-tenant)# exit`	グローバル コンフィギュレーション モードに戻ります。

例

次に、プロキシ ARP を設定する例を示します。

```
apic1# conf t
apic1(config)# tenant Tenant1
apic1(config-tenant)# application Tenant1-App
apic1(config-tenant-app)# epg Tenant1-epg1
apic1(config-tenant-app-epg)# proxy-arp enable
apic1(config-tenant-app-epg)#
apic1(config-tenant)#
```

カプセル化のフラッディングの設定

レイヤ 2 外部接続の設定はスタティック アプリケーション EPG と似ていて、ノード、ポートの VLAN を EPG にマッピングし、EPG をブリッジドメインにマッピングして、コントラクトを提供または使用します。

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	tenant *tenant-name* 例： `apic1(config)# tenant Tenant1`	テナント コンフィギュレーション モードを開始します。
ステップ 3	application *application-profile-name* 例： `apic1(config)# application Tenant1-App`	アプリケーション プロファイルを作成し、アプリケーション モードを開始します。
ステップ 4	epg *application-profile-EPG-name* 例： `apic1(config)# epg Tenant1-epg1`	EPG を作成し、EPG モードを開始します。
ステップ 5	flood-on-encapsulation enable 例：	Flood-on-encapsulation を有効にします。

	コマンドまたはアクション	目的
	apic1(config-tenant-app-epg)# **flood-on-encapsulation enable**	
ステップ 6	exit 例： apic1(config-tenant-app-epg)# **exit**	アプリケーション プロファイル モードに戻ります。

MACsec の設定

MACsec について

MACsec は、IEEE 802.1AE 規格ベースのレイヤ 2 ホップバイホップ暗号化であり、これにより、メディア アクセス非依存プロトコルに対してデータの機密性と完全性を確保できます。

MACsec は、暗号化キーにアウトオブバンド方式を使用して、有線ネットワーク上で MAC レイヤの暗号化を提供します。MACsec Key Agreement（MKA）プロトコルでは、必要なセッション キーを提供し、必要な暗号化キーを管理します。

802.1 ae MKA と暗号化はリンク、つまり、リンク (ネットワーク アクセス デバイスと、PC か IP 電話機などのエンドポイント デバイス間のリンク) が直面しているホストのすべてのタイプでサポートされますかにリンクが接続されている他のスイッチまたはルータ。

MACsec は、イーサネット パケットの送信元および宛先 MAC アドレスを除くすべてのデータを暗号化します。ユーザは、送信元と宛先の MAC アドレスの後に最大 50 バイトの暗号化をスキップするオプションもあります。

WAN またはメトロ イーサネット上に MACsec サービスを提供するために、サービス プロバイダーは、Ethernet over Multiprotocol Label Switching（EoMPLS）および L2TPv3 などのさまざまなトランスポート レイヤ プロトコルを使用して、E-Line や E-LAN などのレイヤ 2 透過サービスを提供しています。

EAP-over-LAN（EAPOL）プロトコルデータ ユニット（PDU）のパケット本体は、MACsec Key Agreement PDU（MKPDU）と呼ばれます。3 回のハートビート後（各ハートビートは 2 秒）に参加者から MKPDU を受信しなかった場合、ピアはライブ ピア リストから削除されます。たとえば、クライアントが接続を解除した場合、スイッチ上の参加者はクライアントから最後の MKPDU を受信した後、3 回のハートビートが経過するまで MKA の動作を継続します。

APIC ファブリック MACsec

APIC はまたは責任を負う MACsec キーチェーン ディストリビューションのポッド内のすべてのノードに特定のポートのノードになります。サポートされている MACsec キーチェーンし、apic 内でサポートされている MACsec ポリシー ディストリビューションのとおりです。

- 単一ユーザ提供キーチェーンと 1 ポッドあたりポリシー

- ユーザが提供されるキーチェーンとファブリック インターフェイスごとのユーザが提供されるポリシー
- 自動生成されたキーチェーンおよび 1 ポッドあたりのユーザが提供されるポリシー

ノードは、複数のポリシーは、複数のファブリック リンクの導入を持つことができます。これが発生すると、ファブリック インターフェイスごとキーチェーンおよびポリシーが優先して指定の影響を受けるインターフェイス。自動生成されたキーチェーンと関連付けられている MACsec ポリシーでは、最も優先度から提供されます。

APIC MACsec では、2 つのセキュリティ モードをサポートしています。MACsec **セキュリティで保護する必要があります** 中に、リンクの暗号化されたトラフィックのみを許可する **セキュリティで保護する必要があります** により、両方のクリアし、リンク上のトラフィックを暗号化します。MACsec を展開する前に **セキュリティで保護する必要があります** モードでのキーチェーンは影響を受けるリンクで展開する必要がありますまたはリンクがダウンします。たとえば、ポートをオンにできますで MACsec **セキュリティで保護する必要があります** モードがピアがしているリンクでのキーチェーンを受信する前にします。MACsec を導入することが推奨されて、この問題に対処する **セキュリティで保護する必要があります** モードとリンクの 1 回すべてにセキュリティ モードを変更 **セキュリティで保護する必要があります** 。

（注） MACsec インターフェイスの設定変更は、パケットのドロップになります。

MACsec ポリシー定義のキーチェーンの定義に固有の設定と機能の機能に関連する設定で構成されています。キーチェーン定義と機能の機能の定義は、別のポリシーに配置されます。MACsec 1 ポッドあたりまたはインターフェイスごとの有効化には、キーチェーン ポリシーおよび MACsec 機能のポリシーを組み合わせることが含まれます。

（注） 内部を使用して生成キーチェーンは、ユーザのキーチェーンを指定する必要はありません。

APIC アクセス MACsec

MACsec はリーフ スイッチ L3out インターフェイスと外部のデバイス間のリンクを保護するために使用します。APIC GUI および CLI のユーザを許可するで、MACsec キーとファブリック L3Out インターフェイスの設定を MacSec をプログラムを提供する物理/pc/vpc インターフェイスごと。ピアの外部デバイスが正しい MacSec 情報を使用してプログラムすることを確認するには、ユーザの責任です。

MACSec の注意事項と制約事項

次の注意事項と制約事項に従って MACSec を設定します。

- Fex ポートは MACsec ではサポートされていません。
- ポッド レベルでは、必須セキュア モードはサポートされていません。

- 「デフォルト」名の MACsec ポリシーはサポートされていません。
- 自動キー生成はファブリック ポートのポッド レベルでのみサポートされます。
- MACSEC はリモート リーフではサポートされていません。
- そのノードのファブリック ポートが **[必須セキュア]** モードの MACsec で実行されている場合、ノードの再起動をクリアしないでください。
- MACsec を実行しているポッド内のノードのポッドまたはステートレス再起動に新しいノードを追加することで、ノードがポッドに参加するために **[必須セキュア]** モードを [should-secure] モードに変更する必要があります。
- ファブリッ のリンクが **[should-secure]** モードの場合、アップグレー/ダウングレードのみを開始する必要があります。アップグレード/ダウン グレードが完了すると、次にモードを **[必須セキュア]** に変更できます。**[必須セキュア]** モードでのアップグレード/ダウングレードは、ノードからファブリックへの接続が失われます。接続の遮断から回復するには、APIC に表示されるノードのファブリック リンクを **[should-secure]** モードに設定する必要があります。ファブリックが MACsec をサポートしていないバージョンにダウングレードされた場合、ファブリック外のノードがクリーンリブートされる必要があります。
- PC/VPC インターフェイスでは、PC/VPC インターフェイスごとのポリシー グループによって MACsec を展開できます。ポート セレクタは、特定のポートのセットにポリシーを展開するために使用されます。したがって、L3Out インターフェイスに対応する正しいポート セレクタを作成することはユーザの責任です。
- 設定がエクスポートされる前に、MACsec ポリシーを **[should-secure]** モードに設定することをお勧めします。
- スパイン上のすべてのリンクは、ファブリック リンクと見なされます。ただし、スパイン リンクの IPN 接続を使用するとアクセス リンクとして次のリンクが処理されます。これは、MACsec アクセス ポリシーを次のリンクの MACsec を導入するために使用する必要があることを意味します。
- MACSEC セッションは、フォームに分をかかるまたは空のキーチェーンに新しいキーが追加されるか、アクティブ キーがキーチェーンから削除切断する可能性があります。

must-secure モードの展開

不正な導入手順に設定されているポリシーの **必須 secure** モードが接続の消失で発生することができます。そのような問題を避けるため次の手順に従う必要があります。

- MACsec **[必須セキュア]** モードを有効にする前に、各リンク ペアがそれぞれのキーチェーンを持っていることを確認する必要があります。これを確認するために推奨されることは、**[should-secure]** モードでポリシーを展開し、MACsec セッションが予想されるリンク上でアクティブになったら、モードを **[必須セキュア]** に変更することです。
- **[必須セキュア]** に設定されている MACsec ポリシーでキーチェーンの交換を試行すると、リンクがダウンする原因となる可能性があります。この場合、以下の推奨手順に従う必要があります。

- 新しいキーチェーンを使用している MACsec ポリシーを **[should-secure]** モードに変更します。
- 影響を受けるインターフェイスが [should-secure] モードを使用しているか確認します。
- 新しいキーチェーンを使用するように MACsec ポリシーを更新します。
- アクティブな MACsec セッションと関連するインターフェイスが新しいキーチェーンを使用していることを確認します。
- MACsec ポリシーを **[必須セキュア]** モードに変更します。

- [必須セキュア] モードに展開されている MACsec ポリシーを無効/削除するには、次の手順に従う必要があります。
  - MACsec ポリシーを **[should-secure]** に変更します。
  - 影響を受けるインターフェイスが **[should-secure]** モードを使用しているか確認します。
  - MACsec ポリシーを無効/削除します。

キーチェーンの定義

- 開始時刻が **[現在]** のキーチェーンに 1 個のキーが存在します。**must-secure** がすぐにアクティブになるキーを持たないキーチェーンで展開される場合、キーが現在時刻になり MACsec セッションが開始されるまでトラフィックはリンク上でブロックされます。**should-secure** モードが使用されている場合、キーが現在になり、MACsec セッションが開始されるまでトラフィックが暗号化されます。
- 終了時刻が **infinite** のキーチェーンに 1 個のキーが存在する必要があります。キーチェーンの期限が切れると、**must-secure** モードに設定されている影響を受けるインターフェイスでトラフィックがブロックされます。設定されたインターフェイス **はセキュア** モード暗号化されていないトラフィック送信します。
- 終了時刻のオーバーラップし、キーの間に移行すると、MACsecセッションを順番に使用されるキーの開始時刻が残っています。

NX-OS スタイルの CLI を使用したMACsecの設定

手順

ステップ 1 アクセス インターフェイスの MACsec セキュリティ ポリシーの設定

例：

```
apic1# configure
apic1(config)#   template macsec access security-policy accmacsecpol1
apic1(config-macsec-param)#     cipher-suite gcm-aes-128
```

```
apic1(config-macsec-param)#     conf-offset offset-30
apic1(config-macsec-param)#     description 'description for mac sec parameters'
apic1(config-macsec-param)#     key-server-priority 1
apic1(config-macsec-param)#     sak-expiry-time 110
apic1(config-macsec-param)#     security-mode must-secure
aapic1(config-macsec-param)#     window-size 1
apic1(config-macsec-param)#     exit
apic1(config)#
```

ステップ 2 アクセス インターフェイスの MACsec キー チェーンを設定します。

PSK は、2 通りの方法で設定できます:

（注）
- 下のキー 12ab に示すように、**psk-string** コマンドを使用してインラインで設定します。PSK は、ログに記録され、公開されるため、安全ではありません。
- キー ab12 で示すように、新しいコマンド **Enter PSK string** を **psk-string** コマンドの後で使用し、個別に入力して設定します。ローカルにエコーされるだけで、ログには記録されないため、PSK は安全です。

例：

```
apic1# configure
apic1(config)#   template macsec access keychain acckeychainpol1
apic1(config-macsec-keychain)#     description 'macsec key chain kc1'
apic1(config-macsec-keychain)#     key 12ab
apic1(config-macsec-keychain-key)#       life-time start 2017-09-19T12:03:15 end
2017-12-19T12:03:15
apic1(config-macsec-keychain-key)#       psk-string 123456789a223456789a323456789abc
apic1(config-macsec-keychain-key)#       exit
apic1(config-macsec-keychain)#     key ab12
apic1(config-macsec-keychain-key)#       life-time start now end infinite
apic1(config-macsec-keychain-key)#       life-time start now end infinite
apic1(config-macsec-keychain-key)# psk-string
Enter PSK string: 123456789a223456789a323456789abc
apic1(config-macsec-keychain-key)# exit
apic1(config-macsec-keychain)# exit
apic1(config)#
```

ステップ 3 アクセス インターフェイスの MACsec インターフェイス ポリシーを設定します:

例：

```
apic1# configure
apic1(config)#   template macsec access interface-policy accmacsecifpol1
apic1(config-macsec-if-policy)#     inherit macsec security-policy accmacsecpol1 keychain
 acckeychainpol1
apic1(config-macsec-if-policy)#     exit
apic1(config)#
```

ステップ 4 MACsec インターフェイス ポリシーをリーフ (またはスパイン) 上のアクセス ンターフェイスに関連付けます:

例：

```
apic1# configure
apic1(config)#   template macsec access interface-policy accmacsecifpol1
apic1(config-macsec-if-policy)#     inherit macsec security-policy accmacsecpol1 keychain
 acckeychainpol1
apic1(config-macsec-if-policy)#     exit
apic1(config)
```

ステップ 5 ファブリック インターフェイス用に MACsec セキュリティ ポリシーを設定します:

例：

```
apic1# configure
apic1(config)#   template macsec fabric security-policy fabmacsecpol1
apic1(config-macsec-param)#     cipher-suite gcm-aes-xpn-128
apic1(config-macsec-param)#     description 'description for mac sec parameters'
apic1(config-macsec-param)#     window-size 1
apic1(config-macsec-param)#     sak-expiry-time 100
apic1(config-macsec-param)#     security-mode must-secure
apic1(config-macsec-param)#     exit
apic1(config)#
```

ステップ 6 ファブリック インターフェイス用に MACsec キー チェーンを設定します:

PSK は、2 通りの方法で設定できます:

（注）
- 下のキー 12ab に示すように、**psk-string** コマンドを使用してインラインで設定します。PSK は、ログに記録され、公開されるため、安全ではありません。
- キー ab12 で示すように、新しいコマンド **Enter PSK string** を **psk-string** コマンドの後で使用し、個別に入力して設定します。ローカルにエコーされるだけで、ログには記録されないため、PSK は安全です。

例：

```
apic1# configure
apic1(config)#   template macsec fabric security-policy fabmacsecpol1
apic1(config-macsec-param)#     cipher-suite gcm-aes-xpn-128
apic1(config-macsec-param)#     description 'description for mac sec parameters'
apic1(config-macsec-param)#     window-size 1
apic1(config-macsec-param)#     sak-expiry-time 100
apic1(config-macsec-param)#     security-mode must-secure
apic1(config-macsec-param)#     exit
apic1(config)#   template macsec fabric keychain fabkeychainpol1
apic1(config-macsec-keychain)#     description 'macsec key chain kc1'
apic1(config-macsec-keychain)#     key 12ab
apic1(config-macsec-keychain-key)#       psk-string 123456789a223456789a323456789abc
apic1(config-macsec-keychain-key)#       life-time start 2016-09-19T12:03:15 end
2017-09-19T12:03:15
apic1(config-macsec-keychain-key)#       exit
apic1(config-macsec-keychain)#     key cd78
apic1(config-macsec-keychain-key)# psk-string
Enter PSK string: 123456789a223456789a323456789abc
apic1(config-macsec-keychain-key)# life-time start now end infinite
apic1(config-macsec-keychain-key)# exit
apic1(config-macsec-keychain)# exit
apic1(config)#
```

ステップ 7 MACsec インターフェイス ポリシーをリーフ (またはスパイン) 上のファブリック ンターフェイスに関連付けます:

例：

```
apic1# configure
apic1(config)#   leaf 101
apic1(config-leaf)#     fabric-interface ethernet 1/52-53
apic1(config-leaf-if)#       inherit macsec interface-policy fabmacsecifpol2
```

```
apic1(config-leaf-if)#       exit
apic1(config-leaf)#
```