



## テナントの設定

- テナント、VRF、およびブリッジドメインの作成 (1 ページ)
- ブリッジドメイン設定の追加 (4 ページ)
- 適用されるブリッジドメインの設定 (6 ページ)
- アプリケーションエンドポイントグループの作成 (8 ページ)
- ブリッジドメインのレガシー転送モードの設定 (11 ページ)
- コントラクトの設定 (13 ページ)
- コントラクトの継承 (16 ページ)
- コントラクト優先グループの設定 (26 ページ)
- 他のテナントへのコントラクトのエクスポート (29 ページ)
- コントラクトまたは件名除外の設定 (31 ページ)
- クォータ管理の作成 (34 ページ)

## テナント、VRF、およびブリッジドメインの作成

このトピックでは、新しいテナントの基本的なプロビジョニングにおける次の手順について説明します。

1. テナントを作成します
2. テナントをセキュリティドメインと関連付けます
3. テナントの VRF を作成します
4. テナント内のエンドポイントグループに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	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。

	コマンドまたはアクション	目的
ステップ 2	<b>tenant</b> <i>tenant-name</i> 例： apic1(config)# <b>tenant exampleCorp</b>	テナント (存在しない場合) を作成し、テナントコンフィギュレーションモードを開始します。
ステップ 3	<b>security domain</b> <i>domain-name</i> 例： apic1(config-tenant)# <b>security domain exampleCorp_dom1</b>	1つ以上のセキュリティドメインとテナントを関連付けます。
ステップ 4	<b>[no] vrf context</b> <i>vrf-name</i> 例： apic1(config-tenant)# <b>vrf context exampleCorp_v1</b>	テナントのプライベートネットワーク (VRF) を作成します。テナントには1つ以上のVRFを設定できます。
ステップ 5	<b>[no] contract {provider   consumer}</b> <i>contract-name</i> 例： apic1(config-tenant-vrf)# <b>contract provider web</b>	VRF下ですべてのEPGのコントラクトを提供または使用します。
ステップ 6	<b>exit</b> 例： apic1(config-tenant-vrf)# <b>exit</b>	テナントコンフィギュレーションモードに戻ります。
ステップ 7	<b>[no] bridge-domain</b> <i>bd-name</i> 例： apic1(config-tenant)# <b>bridge-domain exampleCorp_b1</b>	テナント下でブリッジドメインを作成または削除します。ブリッジドメインコンフィギュレーションモードを開始します。
ステップ 8	<b>[no] vrf member</b> <i>vrf-name</i> 例： apic1(config-tenant-bd)# <b>vrf member exampleCorp_v1</b>	VRFにブリッジドメインを割り当てます。
ステップ 9	<b>exit</b> 例： apic1(config-tenant-bd)# <b>exit</b>	テナントコンフィギュレーションモードに戻ります。
ステップ 10	<b>interface bridge-domain</b> <i>bd-name</i> 例： apic1(config-tenant)# <b>interface bridge-domain exampleCorp_b1</b>	テナントインターフェイスコンフィギュレーションモードを開始して、ブリッジドメインにインターフェイスを適用します。
ステップ 11	<b>[no]{ip  ipv6} address</b> アドレス/マスク 長[scope {private  public}] [secondary]	ブリッジドメインのゲートウェイIPアドレスを割り当てまたは削除し、IP

	コマンドまたはアクション	目的
	<p>例 :</p> <pre> apicl(config-tenant-if)# ip address 172.1.1.1/24 apicl(config-tenant-if)# ipv6 address 2001:1:1::1/64                     </pre>	<p>アドレスモードを開始し、オプションの IP アドレス プロパティを設定します。</p> <p>ゲートウェイアドレスの範囲は、次のいずれかです。</p> <ul style="list-style-type: none"> <li>• <b>Public</b> : ルーティング プロトコル (BGP、OSPF、EIGRP) により外部レイヤ 3 ネットワークにアドバタイズできます。</li> <li>• <b>Private</b> : 外部レイヤ 3 ネットワークにアドバタイズされません。</li> </ul> <p>The optional <b>secondary</b> keyword allows you to configure a secondary gateway address.</p>

例

次に、セキュリティドメインへの割り当て、コントラクトと VRF の作成、ブリッジドメインの作成を含むテナントの基本設定の例を示します。

```

apicl# configure
apicl(config)# tenant exampleCorp
apicl(config-tenant)# security domain exampleCorp_dom1
apicl(config-tenant)# vrf context exampleCorp_v1
apicl(config-tenant-vrf)# contract enforce
apicl(config-tenant-vrf)# contract provider web
apicl(config-tenant-vrf)# contract consumer db
apicl(config-tenant-vrf)# contract provider icmp
apicl(config-tenant-vrf)# contract consumer icmp
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# bridge-domain exampleCorp_b1
apicl(config-tenant-bd)# vrf member exampleCorp_v1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# interface bridge-domain exampleCorp_b1
apicl(config-tenant-interface)# ip address 172.1.1.1/24
apicl(config-tenant-interface)# ipv6 address 2001:1:1::1/64
apicl(config-tenant-interface)# exit
                    
```

次に、リーフに特有の VRF 設定の例を示します。

```

apicl# configure
apicl(config)# leaf 101
apicl(config-leaf)# vrf context exampleCorp_v1 tenant exampleCorp
apicl(config-leaf-vrf)# ip route 1.2.3.4 5.6.7.8
                    
```

次に、リーフ インターフェイスに特有の VRF 設定の例を示します。

```

apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# int eth 1/1
apic1(config-leaf-if)# vrf member exampleCorp_v1 tenant exampleCorp

```

### 次のタスク

アプリケーションプロファイルを追加し、アプリケーションエンドポイントグループ (EPG) を作成し、ブリッジドメインに EPG を関連付けます。

## ブリッジドメイン設定の追加

この項では、ブリッジドメインの次の設定について説明します。

- MAC アドレスの設定
- DHCP リレー アドレスの設定
- 共有サービスのルート漏出の設定

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： apic(config-tenant)# <b>tenant exampleCorp</b>	テナント コンフィギュレーション モードを開始します。
ステップ 3	<b>interface bridge-domain bd-name</b> 例： apic(config-tenant)# <b>interface bridge-domain exampleCorp_bd1</b>	テナント インターフェイス コンフィギュレーションモードを開始してブリッジドメインを設定します。
ステップ 4	(任意) <b>mac-address mac-address</b> 例： apic(config-tenant-interface)# <b>mac-address 1234.5678.abcd</b>	広く使用されているゲートウェイ機能の ARP 応答で使用される MAC アドレスを設定します。
ステップ 5	(任意) <b>no mac-address</b> 例： apic(config-tenant-interface)# <b>no mac-address</b>	MAC アドレスをデフォルトに変更します。

	コマンドまたはアクション	目的
ステップ 6	<p>(任意) <b>[no] ip dhcp relay address tenant</b>  <i>tenant-name dhcp-address {application app-name epg epg-name   external-l2 l2-epg-name   external-l3 l3-epg-name}</i></p> <p>例 :</p> <pre>apic(config-tenant-interface)# ip dhcp relay address 192.0.20.1 tenant exampleCorp application app1 epg epg1</pre>	<p>サポートされているすべてのオプションとともに、ブリッジドメインの DHCP リレー アドレスを設定または削除します。</p>
ステップ 7	<p>(任意) <b>[no]{ip  ipv6} shared address</b>            アドレス/マスク長 <b>provider application</b>            アプリケーション名 <b>epg epg</b></p> <p>例 :</p> <pre>apic(config-tenant-interface)# ip shared address 7.8.9.1/24 provider application app2 epg epg2</pre>	<p>ルート漏出は、DHCP、複数テナント VRF の DNS のような共通サービスを提供するために VRF 全体に許可されます。共有サービスは、サブネットをプロバイダーまたはコンシューマのサブネットとしてマークし、共有サービスを提供する EPG を指定することによって実現されます。</p>
ステップ 8	<p>(任意) <b>[no]{ip  ipv6} shared epg</b>  <b>consumer application</b> アドレス/マスク <b>any</b>            長 <b>addressany</b></p> <p>例 :</p> <pre>apic(config-tenant-interface)# ip shared address 3.2.3.4/24 consumer application any epg any</pre>	<p>上のステップを参照してください。</p>

例

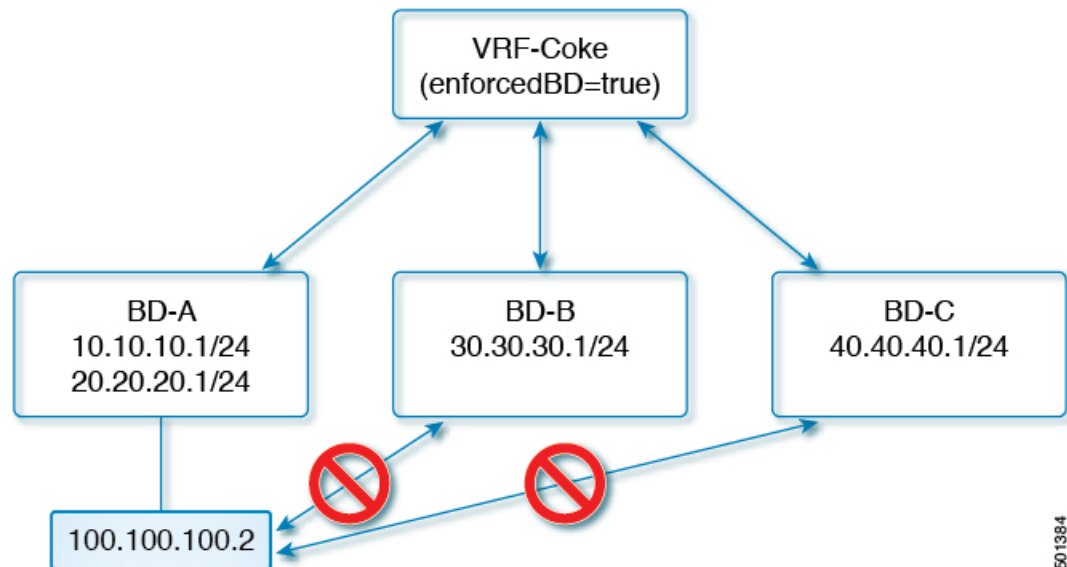
```
apic1# configure
apic1(config)# tenant exampleCorp
apic1(config-tenant)# interface bridge-domain exampleCorp_bd1
apic1(config-tenant-interface)# mac-address 1234.5678.abcd
apic1(config-tenant-interface)# ip dhcp relay address 192.0.20.1 tenant exampleCorp
application app1 epg epg1
apic1(config-tenant-interface)# ip shared address 1.2.3.4/24 provider application any
apic1(config-tenant-interface)# ip shared address 3.2.3.4/24 consumer application any
epg any
apic1(config-tenant-interface)# exit
apic1(config-tenant)# exit
apic1(config)# tenant my_dhcp_provider
apic1(config-tenant)# interface bridge-domain bd_dhcp
apic1(config-tenant-interface)# ip shared address 7.8.9.1/24 provider application app2
epg epg2
```

## 適用されるブリッジドメインの設定

適用されるブリッジドメイン (BD) 設定では、関連付けられているブリッジドメイン内のサブネットゲートウェイを ping できるサブジェクトエンドポイントグループ (EPG) で、エンドポイントを作成する必要があります。

この設定では、任意のサブネットゲートウェイに ping を送信できる IP アドレスのグローバル例外リストを作成できます。

図 1: 適用されるブリッジドメイン



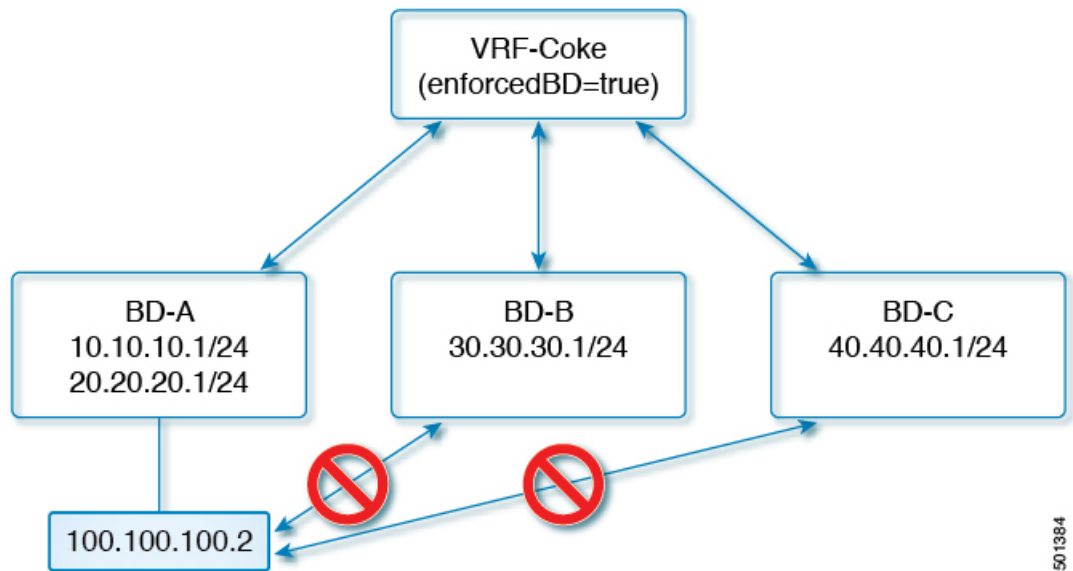
- (注)
- 例外の IP アドレスは、すべての VRF 上ですべての BD ゲートウェイの ping を実行できません。
  - L3 アウトに設定されているループバック インターフェイスは、対象とするループバック インターフェイスに設定されている IP アドレスへの到達可能性を強制しません。
  - eBGP ピア IP アドレスが、L3out インターフェイスのサブネットとは異なるサブネットに存在している場合、許容される例外サブネットにピアサブネットを追加する必要があります。
- そうしないと、送信元 IP アドレスが L3out インターフェイスのサブネットとは異なるサブネットに存在するため、eBGP トラフィックがブロックされます。

## 適用されるブリッジドメインの設定

適用されるブリッジドメイン (BD) 設定では、関連付けられているブリッジドメイン内のサブネットゲートウェイを ping できるサブジェクトエンドポイントグループ (EPG) で、エンドポイントを作成する必要があります。

この設定では、任意のサブネットゲートウェイに ping を送信できる IP アドレスのグローバル例外リストを作成できます。

図 2: 適用されるブリッジドメイン



- (注)
- 例外の IP アドレスは、すべての VRF 上ですべての BD ゲートウェイの ping を実行できません。
  - L3 アウトに設定されているループバック インターフェイスは、対象とするループバック インターフェイスに設定されている IP アドレスへの到達可能性を強制しません。
  - eBGP ピア IP アドレスが、L3out インターフェイスのサブネットとは異なるサブネットに存在している場合、許容される例外サブネットにピアサブネットを追加する必要があります。

そうしないと、送信元 IP アドレスが L3out インターフェイスのサブネットとは異なるサブネットに存在するため、eBGP トラフィックがブロックされます。

## NX-OS スタイル CLI を使用した適用されるブリッジドメインの設定

このセクションでは、NX-OS スタイル コマンドライン インターフェイス (CLI) を使用して、適用されるブリッジドメインを設定する方法について説明します。

## 手順

---

**ステップ1** テナントを作成し有効にします。

例：

次の例 ( 「`cokeVrf`」 ) が作成され有効になっています。

```
apic1(config-tenant)# vrf context cokeVrf
apic1(config-tenant-vrf)# bd-enforce enable
apic1(config-tenant-vrf)# exit
apic1(config-tenant)#exit
```

**ステップ2** 例外リストに、サブネットを追加します。

例：

```
apic1(config)#bd-enf-exp-ip add1.2.3.4/24
apic1(config)#exit
```

---

適用されるブリッジドメインは次のようなコマンドを使用して動作可能かどうかを確認できます。

```
apic1# show running-config all | grep bd-enf
bd-enforce enable
bd-enf-exp-ip add 1.2.3.4/24
```

例

次のコマンドでは、除外リストからサブネットを削除します。

```
apic1(config)# no bd-enf-exp-ip 1.2.3.4/24
apic1(config)#tenant coke
apic1(config-tenant)#vrf context cokeVrf
```

### 次のタスク

適用されるブリッジドメインを無効にするには、次のコマンドを実行します。

```
apic1(config-tenant-vrf)# no bd-enforce enable
```

## アプリケーションエンドポイントグループの作成

このトピックでは、スタティックアプリケーション EPG の基本的なプロビジョニングにおける次の手順について説明します。

1. テナント内のアプリケーションプロファイルを作成します



2. アプリケーションプロファイルに EPG を作成します
3. EPG にブリッジドメインを割り当てます
4. EPG をレイヤ 2 インターフェイスに導入します

始める前に

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エンドポイントグループ (EPG) を作成する前に、VLANドメイン、テナント、VRF、およびブリッジドメインを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apicl# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： apicl (config)# <b>tenant exampleCorp</b>	テナント コンフィギュレーションモードを開始します。
ステップ 3	<b>[no] application app-name</b> 例： apicl (config-tenant)# <b>application OnlineStore</b>	アプリケーションプロファイルを作成し、アプリケーションプロファイ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 4	<b>[no] epg epg-name</b> 例： apicl (config-tenant-app)# <b>epg exampleCorp_webepg1</b>	アプリケーションプロファイルの EPG を作成 (または削除) し、EPG コンフィギュレーションモードを開始します。
ステップ 5	<b>[no] bridge-domain member epg-name</b> 例： apicl (config-tenant-app-epg)# <b>bridge-domain member exampleCorp_b1</b>	ブリッジドメインに EPG を関連付けます。EPG はすべて BD に属している必要があります。
ステップ 6	<b>exit</b> 例： apicl (config-tenant-app-epg)# <b>exit</b>	テナントアプリケーションコンフィギュレーションモードに戻ります。
ステップ 7	<b>exit</b> 例： apicl (config-tenant-app)# <b>exit</b>	テナントコンフィギュレーションモードに戻ります。

	コマンドまたはアクション	目的
ステップ 8	<b>exit</b> 例： apic1(config-tenant)# <b>exit</b>	グローバル コンフィギュレーション モードに戻ります。
ステップ 9	<b>leaf node-id</b> 例： apic1(config)# <b>leaf 101</b>	設定するリーフを指定します。
ステップ 10	<b>interface type</b> 例： apic1(config-leaf)# <b>interface eth 1/2</b>	設定するインターフェイスを指定します。イーサネット ポートの場合は、「ethernet slot / port」を使用します。
ステップ 11	(任意) <b>switchport</b> 例： apic1(config-leaf-if)# <b>switchport</b>	レイヤ 2 はポートのデフォルトの状態であるため、このコマンドはポートをレイヤ 3 設定から変換する必要があるときにのみ必要です。
ステップ 12	<b>vlan-domain member domain-name</b> 例： apic1(config-leaf-if)# <b>vlan-domain member dom1</b>	インターフェイスを VLAN ドメインに関連付けます。
ステップ 13	<b>switchport trunk allowed vlan vlan-id tenant tenant-name app app-name epg epg-name</b> 例： apic1(config-leaf-if)# <b>switchport trunk allowed vlan 10 tenant exampleCorp application OnlineStore epg exampleCorp_webepg1</b>	インターフェイスに EPG を導入し、EPG と VLAN のマッピングを通じて EPG を特定します。この設定は、静的な EPG の導入にのみ適用されます。VLAN が別の EPG または外部 SVI で使用されている場合、この EPG に使用する前に VLAN 設定を削除する必要があります。  (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ます、そうでない場合このコマンドは拒否されます。

## 例

次に、レイヤ 2 ポートに導入されるアプリケーション EPG を作成する例を示します。

```
apic1# configure
apic1(config)# tenant exampleCorp
apic1(config-tenant)# application OnlineStore
```

```

apicl(config-tenant-app)# epg exampleCorp_webepg1
apicl(config-tenant-app-epg)# bridge-domain member exampleCorp_b1
apicl(config-tenant-app-epg)# exit
apicl(config-tenant-app)# exit
apicl(config-tenant)# exit

apicl(config)# leaf 101
apicl(config-leaf)# interface eth 1/2
apicl(config-leaf-if)# switchport
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp application
OnlineStore epg exampleCorp_webepg1
    
```

次に、ポートチャンネルに EPG を導入する例を示します。

```

apicl(config)# leaf 101
apicl(config-leaf)# interface port-channel po1
apicl(config-leaf-if)# switchport
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# switchport trunk allowed vlan 10 tenant exampleCorp application
OnlineStore epg exampleCorp_webepg1
    
```

### 次のタスク

ポートの VLAN を EPG にマッピングします。

## ブリッジドメインのレガシー転送モードの設定

レガシーの転送モードでは、コントラクトまたは EPG を使用せずにスイッチングとルーティングを行うことができます。このモードでは、ポートの VLAN はブリッジドメインに直接マッピングされます。The **legacy forwarding vlan** command automatically creates all necessary objects so that no EPG-related configuration is required.

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： apicl(config)# <b>tenant exampleCorp</b>	テナントコンフィギュレーションモードを開始します。
ステップ 3	<b>bridge-domain bd-name</b> 例： apicl(config-tenant)# <b>bridge-domain exampleCorp_b1</b>	テナントインターフェイスコンフィギュレーションモードを開始してブリッジドメインを設定します。

	コマンドまたはアクション	目的
ステップ 4	<b>[no] legacy forwarding vlan <i>vlan-id</i> vlan-domain <i>vlan-domain-name</i></b>  例： apicl(config-tenant-bd) # <b>legacy-forwarding vlan 50 vlan-domain dom1</b>	ブリッジドメインに VLAN をマッピングします。
ステップ 5	<b>exit</b>  例： apicl(config-tenant-bd) # <b>exit</b>	テナント コンフィギュレーション モードに戻ります。
ステップ 6	<b>exit</b>  例： apicl(config-tenant) # <b>exit</b>	グローバル コンフィギュレーション モードに戻ります。
ステップ 7	<b>leaf <i>node-id</i></b>  例： apicl(config) # <b>leaf 101</b>	設定するリーフを指定します。
ステップ 8	<b>interface <i>type</i></b>  例： apicl(config-leaf) # <b>interface eth 1/1</b>	設定するインターフェイスを指定します。For an Ethernet port, use <b>ethernet slot/port</b> .
ステップ 9	<b>[no] switchport trunk allowed vlan <i>vlan-id</i> tenant <i>tenant-name</i> legacy-forwarding</b>  例： apicl(config-leaf-if) # <b>switchport trunk allowed vlan 50 tenant exampleCorp legacy-forwarding</b>	インターフェイスで VLAN を有効にして、レガシーの転送モードで VLAN を使用するテナントブリッジドメインにこれを関連付けます。

## 例

次に、ブリッジドメイン間の転送にレガシーの転送モードを設定する例を示します。

```
apicl# configure
apicl(config)# tenant exampleCorp
apicl(config-tenant)# bridge-domain exampleCorp_b1
apicl(config-tenant-bd)# legacy-forwarding vlan 50 vlan-domain dom1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# bridge-domain exampleCorp_b2
apicl(config-tenant-bd)# legacy-forwarding vlan 60 vlan-domain dom1
apicl(config-tenant-bd)# exit
apicl(config-tenant)# exit

apicl(config)# leaf 101
apicl(config-leaf)# interface eth 1/1
apicl(config-leaf-if)# vlan-domain member dom1
```

```
apicl(config-leaf-if)# switchport trunk allowed vlan 50 tenant exampleCorp
legacy-forwarding
apicl(config-leaf-if)# exit
apicl(config-leaf)# interface eth 1/2
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# switchport trunk allowed vlan 60 tenant exampleCorp
legacy-forwarding
```

## コントラクトの設定

コントラクトは次のタスクでテナントの下に設定します。

- アクセスリストとしてフィルタを定義します
- コントラクトおよびサブジェクトを定義します
- EPG にコントラクトをリンクします

タスクは、この順序に従う必要はありません。たとえば、コントラクトを定義する前に、EPG にコントラクト名をリンクすることができます。



(注) APIC のフィルタ (ACL) では、従来の NX-OS ACL の **permit | deny** の代わりに **match** が使用されます。フィルタ エントリの目的は、特定のトラフィック フローを一致させることです。トラフィックは、ACL にコントラクトまたはタブー コントラクトが適用されると、許可または拒否されます。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apicl# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： <b>tenant exampleCorp</b>	テナント (存在しない場合) を作成し、テナント コンフィギュレーション モードを開始します。
ステップ 3	<b>access-list acl-name</b> 例： apicl(config-tenant)# <b>access-list http_acl</b>	コントラクトで利用できるアクセスリスト (フィルタ) を作成します。
ステップ 4	(任意) <b>match {arp   icmp   ip}</b> 例：	選択したプロトコルのトラフィックに一致するルールを作成します。

	コマンドまたはアクション	目的
	<code>apicl (config-tenant-acl) # match arp</code>	
ステップ 5	<p>(任意) <code>match {tcp   udp} [ src from[-to]] [ dest from[-to]]</code></p> <p>例 :</p> <pre>apicl (config-tenant-acl) # match tcp dest 80 apicl (config-tenant-acl) # match tcp dest 443</pre>	TCP または UDP トラフィックに一致するルールを作成します。
ステップ 6	<p>(任意) <code>match raw options</code></p> <p>例 :</p> <pre>apicl (config-tenant-acl) #</pre>	Raw vEntry に一致するルールを作成します。
ステップ 7	<p><code>exit</code></p> <p>例 :</p> <pre>apicl (config-tenant-acl) # exit</pre>	テナントコンフィギュレーションモードに戻ります。
ステップ 8	<p><code>contract contract-name</code></p> <p>例 :</p> <pre>apicl (config-tenant) # contract web80</pre>	コントラクトを作成し、コントラク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 9	<p><code>subject subject-name</code></p> <p>例 :</p> <pre>apicl (config-tenant-contract) # subject web80</pre>	コントラクトサブジェクトを作成し、サブジェク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 10	<p>(任意) <code>[no] access-group acl-name [in   out   both]</code></p> <p>例 :</p> <pre>apicl (config-tenant-contract-subj) # access-group http_acl both</pre>	一致するトラフィックの方向を指定し、コントラクトからアクセスリストを追加 (削除) します。
ステップ 11	<p>(任意) <code>[no] label name label-name {provider   consumer}</code></p> <p>例 :</p> <pre>apicl (config-tenant-contract-subj) #</pre>	サブジェクトにプロバイダーまたはコンシューマのラベルを追加 (削除) します。
ステップ 12	<p>(任意) <code>[no] label match {provider   consumer} [any   one   all   none]</code></p> <p>例 :</p> <pre>apicl (config-tenant-contract-subj) #</pre>	<p>次のプロバイダーまたはコンシューマのラベルの一致タイプを指定します。</p> <ul style="list-style-type: none"> <li>• <b>any</b> : 任意のラベルにコントラクト関係がある場合の一致のこと。</li> <li>• <b>one</b> : 1つのラベルにコントラクト関係がある場合の一致のこと。</li> </ul>

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> <li>• <b>all</b> : すべてのラベルにコントラクト関係がある場合の一致のこと。</li> <li>• <b>none</b> : ラベルにコントラクト関係がない場合の一致のこと。</li> </ul>
ステップ 13	<b>exit</b> 例 : apicl (config-tenant-contract-subj) # <b>exit</b>	コントラクトコンフィギュレーションモードに戻ります。
ステップ 14	<b>exit</b> 例 : apicl (config-tenant-contract) # <b>exit</b>	テナントコンフィギュレーションモードに戻ります。
ステップ 15	<b>application app-name</b> 例 : apicl (config-tenant) # <b>application OnlineStore</b>	アプリケーションコンフィギュレーションモードを開始します。
ステップ 16	<b>epg epg-name</b> 例 : apicl (config-tenant-app) # <b>epg exampleCorp_webepg1</b>	コントラクトにリンクする EPG のコンフィギュレーションモードを開始します。
ステップ 17	<b>bridge-domain member bd-name</b> 例 : apicl (config-tenant-app-epg) # <b>bridge-domain member exampleCorp_bd1</b>	この EPG のブリッジドメインを指定します。
ステップ 18	<b>contract provider provider-contract-name</b> 例 : apicl (config-tenant-app-epg) # <b>contract provider web80</b>	この EPG のプロバイダーコントラクトを指定します。この EPG との通信は、このプロバイダーコントラクトに従う通信である限り、その他の EPG から開始することができます。
ステップ 19	<b>contract consumer consumer-contract-name</b> 例 : apicl (config-tenant-app-epg) # <b>contract consumer rmi99</b>	この EPG のコンシューマコントラクトを指定します。この EPG のエンドポイントは、このコントラクトを提供する EPG の任意のエンドポイントとの通信を開始することができます。

## 例

この例では、EPGにコントラクトを作成し適用する方法を示します。

```

apic1# configure
apic1(config)# tenant exampleCorp

# CREATE FILTERS
apic1(config-tenant)# access-list http_acl
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# match tcp dest 443
apic1(config-tenant-acl)# exit

# CREATE CONTRACT WITH FILTERS
apic1(config-tenant)# contract web80
apic1(config-tenant-contract)# subject web80
apic1(config-tenant-contract-subj)# access-group http_acl both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit

# ASSOCIATE CONTRACTS TO EPG
apic1(config-tenant)# application OnlineStore
apic1(config-tenant-app)# epg exampleCorp_webepg1
apic1(config-tenant-app-epg)# bridge-domain member exampleCorp_bd1
apic1(config-tenant-app-epg)# contract consumer rmi99
apic1(config-tenant-app-epg)# contract provider web80
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)#exit
apic1(config-tenant)#exit

# ASSOCIATE PORT AND VLAN TO EPG
apic1(config)#leaf 101
apic1(config-leaf)# interface ethernet 1/4
apic1(config-leaf-if)# switchport trunk allowed vlan 102 tenant exampleCorp application
OnlineStore epg exampleCorp_webepg1

```

この例では、コントラクト自体のフィルタインラインを宣言してコントラクトを定義するためのシンプルな方法を示します。

```

apic1# configure
apic1(config)# tenant exampleCorp
apic1(config-tenant)# contract web80
apic1(config-tenant-contract)# match tcp 80
apic1(config-tenant-contract)# match tcp 443

```

# コントラクトの継承

## コントラクト継承について

関連する契約を新しいEPGに統合するため、EPGを有効にして同じテナントの別のEPGに直接関連する契約すべて（提供済み/消費済み）を継承できます。コントラクトの継承は、アプ

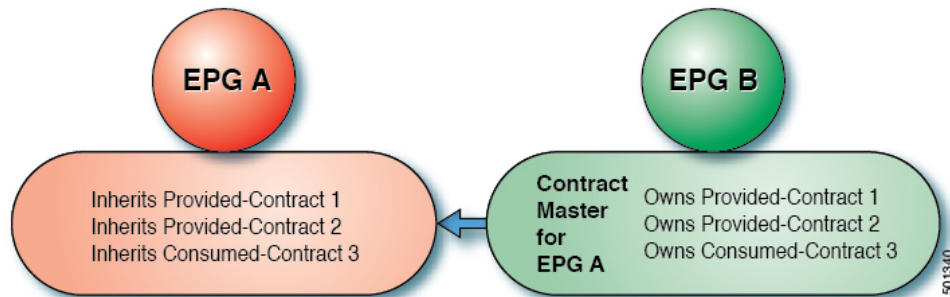


リケーション EPG、マイクロセグメント EPG、L2Out EPG、および L3Out EPG に設定できます。

リリース 3.x では、EPG 間の提供済み/消費済みの両方の契約に、契約を継承する設定も可能です。EPG 間契約が、モデル名や後発のモデルの最後に EX または EX が付く、Cisco Nexus 9000 シリーズ スイッチでサポートされています。

EPG を有効にし、APIC GUI、NX-OS スタイル CLI、REST API を使用して、別の EPG に直接関連する契約すべてを継承できます。

図 3: コントラクトの継承



上の図で、EPG A は EPG B から (EPG A の契約マスター) 提供済みの契約 1 および 2、消費済みの契約 3 を継承するように設定されています。

コントラクト継承を設定する際は、次のガイドラインに従ってください。

- コントラクト継承は、アプリケーション EPG、マイクロセグメント (uSeg) EPG、外部 L2Out EPG、および外部 L3Out EPG 用に設定できます。コントラクト関係は同じタイプの EPG 間で確立する必要があります。
- 関係が確立されると、提供するコントラクトと消費するコントラクトの両方がコントラクトマスターから継承されます。
- コントラクトマスターとコントラクトを継承する EPG は同じテナント内にある必要があります。
- マスター契約への変更は、すべての継承に伝播されます。新しい契約がマスターに追加される場合、継承先にも追加されます。
- EPG は、複数のコントラクトマスターからコントラクトを継承することができます。
- コントラクト継承は単一のレベルでのみサポートされ (連結できない)、コントラクトマスターがコントラクトを継承することはできません。
- コントラクトサブジェクトラベルおよび EPG ラベルの継承がサポートされています。EPG A が EPG B から契約を継承する場合、EPG A と EPG B で異なるサブジェクトラベルが設定されていると、APIC は EPG B で設定されているサブジェクトラベルのみを使用し、どちらの EPG からもラベルを収集しません。
- EPG が契約に直接関連付けられている、または契約を継承しているかどうかに関わらず、TCAM 内のエントリが消費されます。したがって契約スケールガイドラインが引き続き

適用されます。詳細については、お使いのリリースの「検証されたスケーラビリティガイド」を参照してください。

- vzAny セキュリティ コントラクトとタブー コントラクトはサポートされません。

契約の継承設定および継承済みおよびスタンドアロン契約を表示することに関する詳細は、「Cisco APIC の基本設定ガイドを参照してください。

## NX-OS スタイルの CLI を使用したアプリケーションまたは uSeg EPG のコントラクト継承の設定

アプリケーション EPG または uSeg EPG のコントラクト継承を設定するには、次のコマンドを使用します。

### 始める前に

EPG が使用するテナント、アプリケーション プロファイル、およびブリッジ ドメインを設定します。

VRF レベルで EPG が共有するコントラクトを設定します。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 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	<b>tenant tenant-name</b> 例： apic1# (config) tenant Tn1	設定するテナントを作成または指定し、テナントコンフィギュレーションモードを開始します。
ステップ 3	<b>application application-name</b> 例： apic1(config-tenant)# application AP1	アプリケーションを作成または指定し、アプリケーションモードを開始します。
ステップ 4	<b>epg epg-name [type micro-segmented]</b> 例： apic1(config-tenant-app)# epg AEPg403	設定するアプリケーション EPG または uSeg EPG を作成または指定し、EPG コンフィギュレーションモードを開始します。uSeg EPG の場合はタイプを追加します。  この例では、アプリケーション EPG のコントラクト マスターです。

	コマンドまたはアクション	目的
ステップ 5	<b>bridge-domain member <i>bd-name</i></b> 例 : apicl (config-tenant-app-epg) # bridge-domain member T1BD1	ブリッジ ドメインに EPG を関連付けます。
ステップ 6	<b>contract consumer <i>contract-name</i></b> 例 : apicl (config-tenant-app-epg) # contract consumer cctr5	この EPG が消費するコントラクトを追加します。
ステップ 7	<b>contract provider [ <i>label label</i> ]</b> 例 : apicl (config-tenant-app-epg) # contract provider T1ctrl_cif	サブジェクトまたは EPG ラベルのオプションリストなど (事前に設定済みである必要があります)、この EPG が提供するコントラクトを追加します。
ステップ 8	<b>exit</b> 例 : apicl (config-tenant-app-epg) # exit	コンフィギュレーションモードを終了します。
ステップ 9	<b>epg <i>epg-name</i> [type <i>micro-segmented</i> ]</b> 例 : apicl (config-tenant-app) # epg AEPg404	設定するアプリケーション EPG または uSeg EPG を作成または指定し、EPG コンフィギュレーションモードを開始します。uSeg EPG の場合はタイプを追加します。  この例では、コントラクトを継承する EPG です。
ステップ 10	<b>bridge-domain member <i>bd-name</i></b> 例 : apicl (config-tenant-app-epg) # bridge-domain member T1BD1	ブリッジ ドメインに EPG を関連付けます。
ステップ 11	<b>inherit-from-epg application <i>application-name</i>epg <i>EPG-contract-master-name</i> ]</b> 例 : apicl (config-tenant-app-epg) # inherit-from-epg application AP1 epg AEPg403	この EPG が EPG コントラクトマスターからコントラクトを継承するように設定します。
ステップ 12	<b>exit</b> 例 : apicl (config-tenant-app-epg) # exit	コンフィギュレーションモードを終了します。

	コマンドまたはアクション	目的
ステップ 13	<b>epg epg-name [type micro-segmented]</b> 例 : <pre>apic1(config-tenant-app)# epg uSeg1_403_10 type micro-segmented</pre>	設定するアプリケーション EPG または uSeg EPG を作成または指定し、EPG コンフィギュレーションモードを開始します。  この例では、uSeg EPG のコントラクトマスターです。
ステップ 14	<b>bridge-domain member bd-name</b> 例 : <pre>apic1(config-tenant-app-epg)# bridge-domain member T1BD1</pre>	ブリッジドメインに EPG を関連付けます。
ステップ 15	<b>contract provider [ label label]</b> 例 : <pre>apic1(config-tenant-app-epg)# contract provider T1ctrl_uSeg_l3out</pre>	サブジェクトまたは EPG ラベルのオプションリストなど（事前に設定済みである必要があります）、この EPG が提供するコントラクトを追加します。
ステップ 16	<b>attribute-logical-expression logical-expression</b> 例 : <pre>apic1(config-tenant-app-epg)# attribute-logical-expression 'ip equals 192.168.103.10 force'</pre>	一致基準として論理式を uSeg EPG に追加します。
ステップ 17	<b>exit</b> 例 : <pre>apic1(config-tenant-app-epg)# exit</pre>	コンフィギュレーションモードを終了します。
ステップ 18	<b>epg epg-name [type micro-segmented]</b> 例 : <pre>apic1(config-tenant-app)# epg uSeg1_403_30 type micro-segmented</pre>	設定するアプリケーション EPG または uSeg EPG を作成または指定し、EPG コンフィギュレーションモードを開始します。  この例では、EPG コントラクトマスターからコントラクトを継承する uSeg EPG です。
ステップ 19	<b>bridge-domain member bd-name</b> 例 : <pre>apic1(config-tenant-app-epg)# bridge-domain member T1BD1</pre>	ブリッジドメインに EPG を関連付けます。
ステップ 20	<b>attribute-logical-expression logical-expression</b> 例 :	基準として論理式を uSeg EPG に追加します。

	コマンドまたはアクション	目的
	apicl(config-tenant-app-epg)# attribute-logical-expression 'ip equals 192.168.103.30 force'	
ステップ 21	<b>inherit-from-epg application</b> <i>application-name epg</i> <i>EPG-contract-master-name</i>  例： apicl(config-tenant-app-epg)# inherit-from-epg application AP1 epg uSeg1_403_10	この EPG が EPG コントラクト マスターからコントラクトを継承するように設定します。
ステップ 22	<b>exit</b>  例： apicl(config-tenant-app-epg)# exit	コンフィギュレーションモードを終了します。
ステップ 23	<b>exit</b>  例： apicl(config-tenant-app)# exit	コンフィギュレーションモードを終了します。
ステップ 24	<b>exit</b>  例： apicl(config-tenant)# exit	コンフィギュレーションモードを終了します。
ステップ 25	<b>exit</b>  例： apicl(config)# exit	コンフィギュレーションモードを終了します。

例

```
ifav90-ifcl1# show running-config tenant Tn1 application AP1
# Command: show running-config tenant Tn1 application AP1
# Time: Fri Apr 28 17:28:32 2017
tenant Tn1
  application AP1
    epg AEPg403
      bridge-domain member T1BD1
      contract consumer cctr5 imported
      contract provider T1ctrl1_cif
    exit
    epg AEPg404
      bridge-domain member T1BD1
      inherit-from-epg application AP1 epg AEPg403
    exit
    epg uSeg1_403_10 type micro-segmented
      bridge-domain member T1BD1
      contract provider T1Ctrl1_uSeg_l3out
      attribute-logical-expression 'ip equals 192.168.103.10 force'
    exit
    epg uSeg1_403_30 type micro-segmented
      bridge-domain member T1BD1
```

```

attribute-logical-expression 'ip equals 192.168.103.30 force'
inherit-from-epg application AP1 epg uSeg1_403_10
exit
exit
exit

```

## NX-OS スタイルの CLI を使用した L2Out EPG のコントラクト継承の設定

外部 L2Out EPG のコントラクト継承を設定するには、次のコマンドを使用します。

### 始める前に

EPG が使用するテナント、VRF、およびブリッジドメインを設定します。

EPG が使用するレイヤ 2 外部ネットワーク (L2Out) を設定します。

VRF レベルで EPG が共有するコントラクトを設定します。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 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	<b>tenant tenant-name</b> 例： apic1(config)# tenant Tn1	設定するテナントを作成または指定し、テナントコンフィギュレーションモードを開始します。
ステップ 3	<b>external-l2 epg external-l2-epg-name</b> 例： apic1(config-tenant)# external-l2 epg l2out1:l2Ext1	外部 L2Out EPG を設定または指定します。この例では、L2out コントラクトマスターです。
ステップ 4	<b>bridge-domain member bd-name</b> 例： apic1(config-tenant-l2ext-epg)# bridge-domain member T1BD1	ブリッジドメインに L2Out EPG を関連付けます。
ステップ 5	<b>contract provider contract-name [ label label]</b> 例： apic1(config-tenant-l2ext-epg)# contract provider T1ctr_tcp	この EPG が提供するコントラクトを追加します。
ステップ 6	<b>exit</b> 例：	コンフィギュレーションモードを終了します。

	コマンドまたはアクション	目的
	<code>apicl(config-tenant-l2ext-epg)# exit</code>	
ステップ 7	<b>external-l2 epg <i>external-l2-epg-name</i></b> 例： <code>apicl(config-tenant)# external-l2 epg L2out12:l2Ext12</code>	外部 L2Out EPG を設定します。この例では、L2out コントラクト マスターからコントラクトを継承する EPG です。
ステップ 8	<b>bridge-domain member <i>bd-name</i></b> 例： <code>apicl(config-tenant-l2ext-epg)# bridge-domain member T1BD1</code>	ブリッジ ドメインに L2out EPG を関連付けます。
ステップ 9	<b>inherit-from-epg <i>L2Out-contract-master-name</i></b> 例： <code>apicl(config-tenant-l2ext-epg)# inherit-from-epg epg l2out1:l2Ext1</code>	この EPG が L2Out コントラクト マスターからコントラクトを継承するように設定します。
ステップ 10	<b>exit</b> 例： <code>apicl(config-tenant-l2ext-epg)# exit</code>	コンフィギュレーションモードを終了します。

### 例

上記の手順は次の例からの抜粋です。

```
apicl# show running-config tenant Tn1 external-l2
# Command: show running-config tenant Tn1 external-l2
# Time: Thu May 11 13:10:14 2017
tenant Tn1
  external-l2 epg l2out1:l2Ext1
    bridge-domain member T1BD1
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out10:l2Ext10
    bridge-domain member T1BD10
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out11:l2Ext11
    bridge-domain member T1BD11
    contract provider T1ctr_udp
    exit
  external-l2 epg l2out12:l2Ext12
    bridge-domain member T1BD12
    inherit-from-epg epg l2out1:l2Ext1
    inherit-from-epg epg l2out10:l2Ext10
    inherit-from-epg epg l2out11:l2Ext11
    inherit-from-epg epg l2out2:l2Ext2
    inherit-from-epg epg l2out3:l2Ext3
    inherit-from-epg epg l2out4:l2Ext4
    inherit-from-epg epg l2out5:l2Ext5
    inherit-from-epg epg l2out6:l2Ext6
    inherit-from-epg epg l2out7:l2Ext7
```

```

inherit-from-epg epg l2out8:l2Ext8
inherit-from-epg epg l2out9:l2Ext9
exit
external-l2 epg l2out2:l2Ext2
  bridge-domain member T1BD2
  contract provider T1ctr_tcp
exit
external-l2 epg l2out3:l2Ext3
  bridge-domain member T1BD3
  contract provider T1ctr_tcp
exit
external-l2 epg l2out4:l2Ext4
  bridge-domain member T1BD4
  contract provider T1ctr_tcp
exit
external-l2 epg l2out5:l2Ext5
  bridge-domain member T1BD5
  contract provider T1ctr_tcp
exit
external-l2 epg l2out6:l2Ext6
  bridge-domain member T1BD6
  contract provider T1ctr_tcp
exit
external-l2 epg l2out7:l2Ext7
  bridge-domain member T1BD7
  contract provider T1ctr_tcp
exit
external-l2 epg l2out8:l2Ext8
  bridge-domain member T1BD8
  contract provider T1ctr_tcp
exit
external-l2 epg l2out9:l2Ext9
  bridge-domain member T1BD9
  contract provider T1ctr_tcp
exit
exit

```

## NX-OS スタイルの CLI を使用した外部 L3Out EPG のコントラクト継承の設定

外部 L3Out EPG のコントラクト継承を設定するには、次のコマンドを使用します。

### 始める前に

EPG が使用するテナント、VRF、およびブリッジドメインを設定します。

EPG が使用するレイヤ 3 外部ネットワーク (L3Out) を設定します。

VRF レベルで EPG が共有するコントラクトを設定します。

### 手順

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



	コマンドまたはアクション	目的
	apicl# configure	
ステップ 2	<b>tenant</b> <i>tenant-name</i> 例 : apicl(config)# tenant Tn1	設定するテナントを作成または指定し、テナントコンフィギュレーションモードを開始します。
ステップ 3	<b>external-l3 epg</b> <i>external-l3-epg-name l3out l3out-name</i> 例 : apicl(config-tenant-app)# external-l3 epg l3Ext108 l3out T1L3out1	外部 L3Out EPG を設定します。この例では、L3out コントラクトマスターです。
ステップ 4	<b>vrf member</b> <i>vrf-name</i> 例 : apicl(tenant-l3out)# vrf member T1ctx1	L3out を VRF に関連付けます。
ステップ 5	<b>match ip</b> <i>ip-address-and-mask</i> 例 : apicl(config-tenant-l3ext-epg)# match ip 192.168.110.0/24 shared	EPG の一部としてホストを識別するサブネットを追加し、そのサブネットのオプションの共有範囲を追加します。
ステップ 6	<b>contract provider</b> <i>contract-name [ label label]</i> 例 : apicl(config-tenant-l3ext-epg)# contract provider T1ctrl-L3out	この EPG が提供するコントラクトを追加します。
ステップ 7	<b>exit</b> 例 : apicl(config-tenant-l3ext-epg)# exit	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	<b>external-l3 epg</b> <i>external-l3-epg-name l3out l3out-name</i> 例 : apicl(config-tenant-app)# external-l3 epg l3Ext110 l3out T1L3out1	外部 L3Out EPG を設定します。この例では、L3out コントラクトマスターからコントラクトを継承する EPG です。
ステップ 9	<b>vrf member</b> <i>vrf-name</i> 例 : apicl(tenant-l3out)# vrf member T1ctx1	L3out を VRF に関連付けます。
ステップ 10	<b>match ip</b> <i>ip-address-and-mask</i> 例 : apicl(config-tenant-l3ext-epg)# match ip 192.168.112.0/24 shared	EPG の一部としてホストを識別するサブネットを追加し、そのサブネットのオプションの共有範囲を追加します。

	コマンドまたはアクション	目的
ステップ 11	<b>inherit-from-epg</b> <i>L3Out-contract-master-name</i> 例： <pre>apic1(config-tenant-l3ext-epg)# inherit-from-epg l3Ext108</pre>	この EPG が L3Out コントラクトマスターからコントラクトを継承するように設定します。
ステップ 12	<b>exit</b> 例： <pre>apic1(config-tenant-l3ext-epg)# exit</pre>	コンフィギュレーションモードを終了します。

### 例

```
ifav90-ifc1# show running-config tenant Tn1 external-l3 epg l3Ext110
# Command: show running-config tenant Tn1 external-l3 epg l3Ext110
# Time: Fri Apr 28 17:36:15 2017
tenant Tn1
  external-l3 epg l3Ext108 l3out T1L3out1
    vrf member T1ctx1
    match ip 192.168.110.0/24 shared
    contract provider T1ctrl-L3out
    exit
  external-l3 epg l3Ext110 l3out T1L3out1
    vrf member T1ctx1
    match ip 192.168.112.0/24 shared
    inherit-from-epg epg l3Ext108
    exit
exit
```

## コントラクト優先グループの設定

### 契約優先グループについて

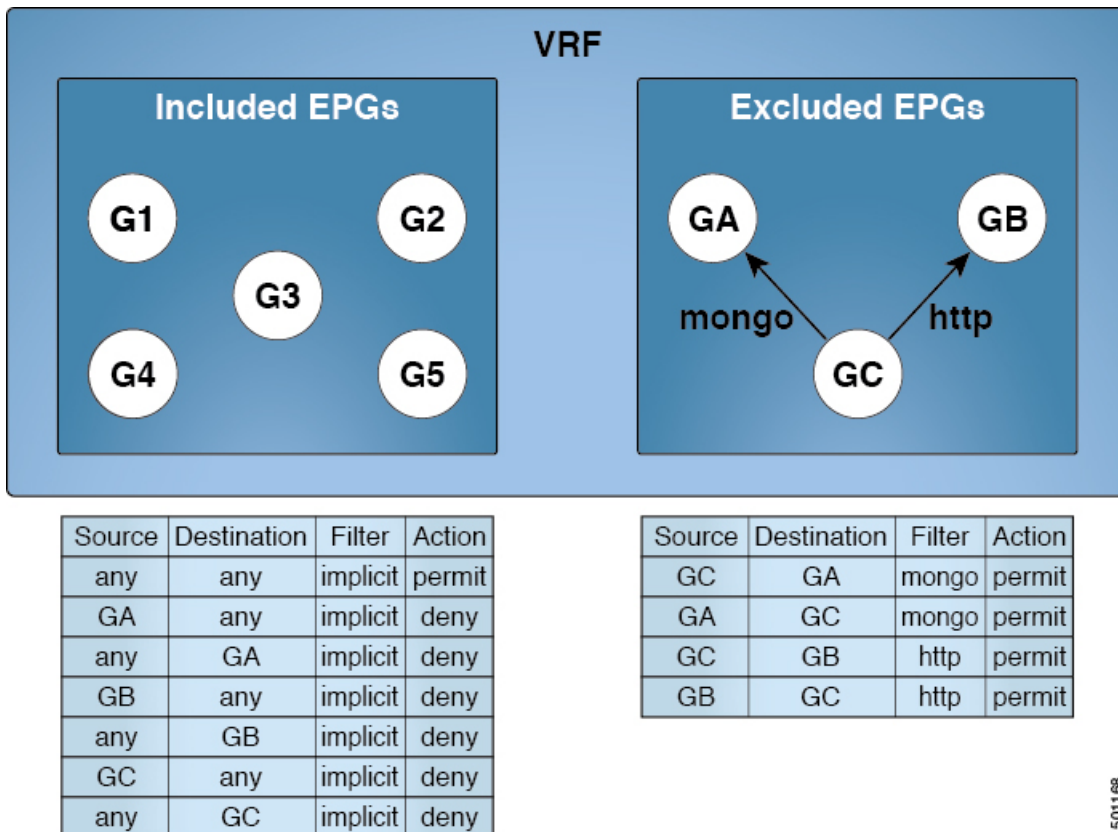
契約優先グループが設定されている VRF で、EPG に利用可能なポリシー適用には 2 種類あります。

- EPG を含む：EPG が契約優先グループのメンバーシップを持っている場合、EPG は契約をせずにお互いに自由に通信できます。これは、`source-any-destination-any-permit` デフォルトルールに基づくものです。
- EPG を除外：優先グループのメンバーではない EPG は、相互に通信するために契約が必要です。そうしない場合、デフォルトの `source-any-destination-any-deny` ルールが適用されます。

契約優先グループ機能では、VRF で EPG 間のより高度な通信の制御が可能です。VRF の EPG のほとんどはオープン通信ですが、一部には他の EPG との制限がある場合、契約優先グループとフィルタ付きの契約の組み合わせを設定し、EPG 内の通信を正確に制御できます。

優先グループから除外されている EPG は、source-any-destination-any-deny デフォルトルールを上書きする契約がある場合にのみ、他 EPG と通信できます。

図 4: 契約優先グループの概要



501168

**制限事項**

以下の制限が契約優先グループに適用されます。

- L3Out およびアプリケーション EPG が契約優先グループで設定されており、EPG が VPC でのみ展開されているトポロジで、VPC の 1 つのリーフ スイッチのみに L3Out のプレフィックス エントリがあることがわかります。この場合、VPC の他のリーフ スイッチにはエントリがなく、そのためトラフィックをドロップします。

この問題を回避するには、次のいずれかを行います。

- VRF の契約グループを無効および再度有効にします。
- L3Out EPG のプレフィックス エントリを削除し再度作成します。
- また、サービス グラフ契約のプロバイダまたはコンシューマ EPG が契約グループに含まれる場合、シャドウ EPG は契約グループから除外できません。シャドウ EPG は契約グループで許可されますが、シャドウ EPG が展開されているノードで契約グループポリシーの展開をトリガしません。ノードに契約グループポリシーをダウンロードするには、契約グループ内にダミー EPG を展開します。

## NX-OS スタイル CLI を使用したコントラクト優先グループの設定

APIC NX-OS スタイル CLI を使用して、コントラクト優先グループを設定することができます。この例では、VRF のコントラクト優先グループが設定されています。VRF を使用する EPG のひとつは、優先グループに含まれます。

### 始める前に

コントラクト優先グループで消費されるテナント、VRF、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	<code>configure</code> 例： <code>apic1# configure</code> <code>apic1(config)#</code>	設定モードを開始します。
ステップ 2	<code>tenant tenant-name</code> 例： <code>apic1(config)# tenant tenant64</code>	テナントを作成するか、テナント設定モードを開始します
ステップ 3	<code>vrf context vrf-name</code> 例： <code>apic1(config-tenant)# vrf context vrf64</code>	VRF を作成するか、VRF 設定モードを開始します
ステップ 4	<code>whitelist-blacklist-mix</code> 例： <code>apic1(config-tenant-vrf)# whitelist-blacklist-mix</code> <code>apic1(config-tenant-vrf)# exit</code>	VRF のコントラクト優先グループを有効にし、テナント設定モードに戻ります。
ステップ 5	<code>bridge-domain bd-name</code> 例： <code>apic1(config-tenant)# bridge-domain bd64</code>	VRF のブリッジドメインを作成するか、BD 設定モードを開始します。
ステップ 6	<code>vrf member vrf-name</code> 例： <code>apic1(config-tenant-bd)# vrf member vrf64</code> <code>apic1(config-tenant-bd)# exit</code>	ブリッジドメインと VRF を関連付け、テナント設定モードに戻ります。

	コマンドまたはアクション	目的
ステップ 7	<b>application <i>app-name</i></b> 例： apicl(config-tenant)# application app-ldap	アプリケーションを作成するか、アプリケーション設定モードを開始します。
ステップ 8	<b>epg <i>epg-name</i></b> 例： apicl(config-tenant-app)# epg epg-ldap	EPGを作成するか、EPGテナントアプリケーションEPG設定モードを開始します。
ステップ 9	<b>bridge-domain member <i>bd-name</i></b> 例： apicl(config-tenant-app-epg)# bridge-domain member bd64	ブリッジ-ドメインに EPG を関連付けます。
ステップ 10	<b>vrf-blacklist-mode</b> 例： apicl(config-tenant-app-epg)# vrf-blacklist-mode	コントラクト優先グループに含まれるこの EPG を設定します。

### 例

次の例では、vrf64 のコントラクト優先グループを作成し、epg-ldap を含めます。

```
apicl# configure
apicl(config)# tenant tenant64
apicl(config-tenant)# vrf context vrf64
apicl(config-tenant-vrf)# whitelist-blacklist-mix
apicl(config-tenant-vrf)# exit

apicl(config-tenant)# bridge-domain bd64
apicl(config-tenant-bd)# vrf member vrf64
apicl(config-tenant-bd)# exit

apicl(config-tenant)# application app-ldap
apicl(config-tenant-app)# epg epg-ldap
apicl(config-tenant-app-epg)# bridge-domain member bd64
apicl(config-tenant-app-epg)# vrf-blacklist-mode
```

## 他のテナントへのコントラクトのエクスポート

1つのテナントからコントラクトをエクスポートし、別のテナントにインポートできます。コントラクトをインポートするテナントでは、コントラクトはコンシューマコントラクトとしてのみ適用できます。コントラクトはエクスポート時に名前を変更できます。

## 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configure</b> 例： apic1# <b>configure</b>	コンフィギュレーションモードに入ります。
ステップ 2	<b>tenant tenant-name</b> 例： apic1(config)# <b>tenant RedCorp</b>	エクスポートするテナントのテナントコンフィギュレーションモードを開始します。
ステップ 3	<b>contract contract-name</b> 例： apic1(config-tenant)# <b>contract web80</b>	エクスポートするコントラクトのコントラク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 4	<b>scope {application   exportable   tenant   vrf}</b> 例： apic1(config-tenant-contract)# <b>scope exportable</b>	<p>コントラクトの共有方法を設定します。スコープは次のようになります。</p> <ul style="list-style-type: none"> <li>• <b>application</b>—同じアプリケーションの EPG で共有可能</li> <li>• <b>exportable</b>—テナントで共有可能</li> <li>• <b>tenant</b>—同じテナントの EPG で共有可能</li> <li>• <b>vrf</b>—同じ VRF の EPG で共有可能</li> </ul>
ステップ 5	<b>export to tenant other-tenant-name as new-contract-name</b> 例： apic1(config-tenant-contract)# <b>export to tenant BlueCorp as webContract1</b>	他のテナントにコントラクトをエクスポートします。同じコントラクト名を使用することも、名前を変更することもできます。
ステップ 6	<b>exit</b> 例： apic1(config-tenant-contract)# <b>exit</b>	テナントコンフィギュレーションモードに戻ります。
ステップ 7	<b>exit</b> 例： apic1(config-tenant)# <b>exit</b>	グローバルコンフィギュレーションモードに戻ります。
ステップ 8	<b>tenant tenant-name</b> 例： <b>tenant BlueCorp</b>	インポートするテナントのテナントコンフィギュレーションモードを開始します。

	コマンドまたはアクション	目的
ステップ 9	<b>application <i>app-name</i></b> 例： apicl(config-tenant)# <b>application BlueStore</b>	アプリケーション コンフィギュレーション モードを開始します。
ステップ 10	<b>epg <i>epg-name</i></b> 例： apicl(config-tenant-app)# <b>epg BlueWeb</b>	コントラクトにリンクする EPG のコンフィギュレーションモードを開始します。
ステップ 11	<b>contract consumer <i>consumer-contract-name</i> imported</b> 例： apicl(config-tenant-app-epg)# <b>contract consumer webContract1 imported</b>	この EPG にインポートされたコンシューマコントラクトを指定します。この EPG のエンドポイントは、このコントラクトを提供する EPG の任意のエンドポイントとの通信を開始することができます。

### 例

次に、テナント RedCorp から、コンシューマ コントラクトになるテナント BlueCorp にコントラクトをエクスポートする例を示します。

```
apic# configure
apicl(config)# tenant RedCorp
apicl(config-tenant)# contract web80
apicl(config-tenant-contract)# scope exportable
apicl(config-tenant-contract)# export to tenant BlueCorp as webContract1
apicl(config-tenant-contract)# exit
apicl(config-tenant)# exit
apicl(config)# tenant BlueCorp
apicl(config-tenant)# application BlueStore
apicl(config-tenant-application)# epg BlueWeb
apicl(config-tenant-application-epg)# contract consumer webContract1 imported
```

## コントラクトまたは件名除外の設定

### コントラクトまたはコントラクトの件名の除外の設定

Cisco APIC リリース 3.2(1) では、EPG 間のコントラクトが拡張され、コントラクトに参加しているコントラクトプロバイダまたはコンシューマのサブネットを拒否できます。インター EPG コントラクトおよび内部 EPG コントラクトは、この機能でサポートされます。

プロバイダ EPG の件名を有効にして、件名またはコントラクトの例外で一致基準が設定されているものを除くすべてのコンシューマ EPG との通信が可能になります。たとえば、サブセッ

トを除く、テナントのすべての EPG にサービスを提供するために EPG を有効にする場合、これら EPG を除外できます。これを設定するには、コントラクトまたはそのコントラクトの件名のいずれかで例外を作成します。サブセットがコントラクトの提供または消費のアクセスを拒否します。

ラベル、カウンタ、許可および拒否ログは、コントラクトおよび件名の例外でサポートされています。

コントラクトのすべての件名に例外を適用するには、コントラクトに例外を追加します。コントラクトの単一の件名にのみ例外を適用する場合、件名に例外を追加します。

件名にフィルタを追加する場合、フィルタのアクションを設定できます（フィルタ条件に一致するオブジェクトを許可または拒否する）。また、**[拒否]** フィルタについては、フィルタの優先順位を設定することができます。**[許可]** フィルタは常にデフォルトの優先順位があります。自動拒否の件名-フィルタ関係をマーキングすると、件名に一致している場合、各 EPG のペアに適用されます。コントラクトと件名には、複数の件名-フィルタ関係を含むことができます。これは、フィルタに一致するオブジェクトを許可または拒否するように独自に設定できます。

### 例外タイプ

コントラクトと件名の例外は次のタイプに基づき、\* ワイルドカードなどの正規表現を含むことができます。

例外の条件は、[コンシューマ正規表現] および [プロバイダ正規表現] のフィールドで定義されているように、これらのオブジェクトを除外します。	例	説明
<b>テナント</b>	<pre>&lt;vzException consRegex="common" field="Tenant" name="excep03" provRegex="t1" /&gt;</pre>	この例では、common テナントを使用して、EPG が t1 テナントにより提供されるコントラクトを消費しないように除外します。
<b>VRF</b>	<pre>&lt;vzException consRegex="ctx1" field="Ctx" name="excep05" provRegex="ctx1" /&gt;</pre>	この例では、ctx1 のメンバーが同じ VRF から提供されるサービスを使用しないように除外します。
<b>EPG</b>	<pre>&lt;vzException consRegex="EPgPa*" field="EPg" name="excep03" provRegex="EPg03" /&gt;</pre>	この例では、名前が EPgPa から始まる複数の EPG が存在すると仮定し、EPg03 により提供されているコントラクトのコンシューマとしてすべて拒否される必要があります。



例外の条件は、[コンシューマ正規表現] および [プロバイダ正規表現] のフィールドで定義されているように、これらのオブジェクトを除外します。	例	説明
<b>Dn</b>	<pre>&lt;vzException consRegex= "uni/tn-t36/ap-customer/epg-epg193" field= "Dn" name="excep04" provRegex= "uni/tn-t36/ap-customer/epg-epg200" /&gt;</pre>	この例では、epg193 が epg200 により提供されたコントラクトを消費しないように除外します。
<b>タグ</b>	<pre>&lt;vzException consRegex= "red" field= "Tag" name= "excep01" provRegex= "green" /&gt;</pre>	例では、red タグでマークされているオブジェクトが消費することと、green タグでマークされているオブジェクトがコントラクトに参加しないように除外します。

## NX-OS スタイルの CLI を使用したコントラクトまたはコントラクトの件名除外の設定

このタスクでは、ほとんどの EPG の通信を許可するコントラクトを設定しますが、それらのサブネットへのアクセスを拒否します。契約またはサブジェクトには、複数の例外を追加することができます。

### 始める前に

テナント、VRF、アプリケーションプロファイル、EPG を設定して、コントラクトを提供し消費します。

### 手順

**ステップ 1** 次の例のようにコマンドを使用して、HTTP および HTTPS のフィルタを設定します。

例：

```
apicl(config)# tenant t2
apicl(config-tenant)# access-list acl
apicl(config-tenant-acl)# match ip
apicl(config-tenant-acl)# match tcp dest 80
apicl(config-tenant-acl)# exit
apicl(config-tenant)# access-list ac2
apicl(config-tenant-acl)# match ip
apicl(config-tenant-acl)# match tcp dest 443
```

**ステップ 2** EPg01 の消費と EPg03 の提供を除外するコントラクトを設定します。

例：

```
apicl(config-tenant)# contract webCtrct
apicl(config-tenant-contract)# subject https-subject
apicl(config-tenant-contract-subj)# exception name EPG consumer-regex EPg01 field EPg
provider-regex EPg03
apicl(config-tenant-contract-subj)# access-group ac1 in blacklist
apicl(config-tenant-contract-subj)# access-group ac2 in whitelist
```

## クォータ管理の作成

### APIC クォータ管理の設定について

Cisco で開始 Application Policy Infrastructure Controller ( APIC ) 2.3(1) のリリースでは、テナント管理者を設定できるオブジェクトの数に制限します。これにより、特定のテナントでまたはテナント間でグローバルに追加できるどのような管理対象オブジェクトを制限する管理できます。APIC ) 2.3(1) のリリースでは、テナント管理者を設定できるオブジェクトの数に制限します。これにより、特定のテナントでまたはテナント間でグローバルに追加できるどのような管理対象オブジェクトを制限する管理できます。

この機能は、任意のテナントまたは承認からテナントのグループを制限するときに便利な ACI リーフ単位またはまたは不当可能性がある他のテナントで同じファブリックに影響する、使用可能なリソースの大部分を使用するファブリックあたりの最大値。

### NX-OS スタイル CLI を使用したクォータ設定の作成

この手順では、NX-OS スタイル CLI を使用してクォータ管理構成を作成する方法について説明します。

手順

NX-OS CLI を使用して、クォータ管理設定を作成します。

例：

```
apicl# conf t
apicl(config)# quota fvBD max 100 scope uni/tn-green exceed-action fault
apicl(config)# quota fvBD max 1000 scope uni exceed-action fail
apicl(config)# quota fvBD max 34 tenant red
```

構文：

```
[no] quota <className> max <maxValue> [exceed-action {fail|fault}] \
[scope <containerDn>| tenant <tenantName> \
[bridge-domain <bd>|application <ap> [epg <epgName>]]}]
```

<className> は、**fvBD** や **fvCtx** などの管理対象オブジェクトクラス名です。モデルのクォータフラグの存在に従ってすべての対象となるクラスは受け入れられます。

<maxValue> は、<exceed-action> が適用された後の値です。

<maxValue> が超過した後 <exceed-action> がアクションを起こす場合、次のいずれかになります。

- **失敗**：制限を超えたトランザクションが失敗することを希望する場合。
- **障害**：障害が発生します。

ここで、<containerDn> は、制限が適用されるツリーです。「uni」はポリシーモデル全体ACIにわたって行われ、「tenant green」は、グリーンテナントに対して行われます。

---

