



サービス グラフ テンプレートの使用

- [GUI を使用したサービス グラフ テンプレートとコントラクトおよび EPG の関連付け](#) (1 ページ)
- [NX-OS スタイルの CLI を使用したサービス グラフ テンプレートの作成](#) (1 ページ)
- [REST API を使用したサービス グラフ テンプレートの設定](#) (4 ページ)

GUI を使用したサービス グラフ テンプレートとコントラクトおよび EPG の関連付け

GUI を使用して、サービス グラフ テンプレートをコントラクトとエンドポイント グループ (EPG) に関連付ける必要があります。



(注) サービス グラフ テンプレートをコントラクトと EPG に関連付けるには、GUI のみを使用できます。

サービス グラフ テンプレートをコントラクトと EPG に関連付ける手順については、[GUI の使用方法](#) を参照してください。

NX-OS スタイルの CLI を使用したサービス グラフ テンプレートの作成

次に、サービス グラフ テンプレートの作成手順を示します。

ステップ 1 コンフィギュレーション モードを開始します。

例 :

```
apic1# configure
```

ステップ 2 テナントのコンフィギュレーション モードを開始します。

```
tenant tenant_name
```

例 :

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

ステップ 3 サービス グラフをテンプレートに関連付けます。

```
1417 graph graph_name contract contract_name
```

パラメータ	説明
グラフ	サービス グラフ テンプレートの名前。
contract	サービス グラフ テンプレートで使用するコントラクトの名前。

例 :

```
apicl(config-tenant)# 1417 graph GraphL3asa contract ContractL3ASA
```

ステップ 4 機能ノードを追加します。

```
service node_name [device-cluster-tenant tenant_name] [device-cluster device_name] [mode deployment_mode]
```

パラメータ	説明
service	追加するサービス ノードの名前。
device-cluster-tenant	デバイスクラスタのインポート元のテナント。グラフを設定するテナントと同じテナントにデバイスクラスタがない場合にのみ、このパラメータを指定します。
device-cluster	このサービス ノードに使用するデバイス クラスタの名前。
mode	導入モード。値は次のとおりです。 <ul style="list-style-type: none"> • ADC_ONE_ARM : ワンアーム モードを指定します。 • ADC_TWO_ARM : ツーアーム モードを指定します。 • FW_ROUTED : ルーテッド (GoTo) モードを指定します。 • FW_TRANS : トランスペアレント (GoThrough) モードを指定します。 • OTHERS モードを指定しないと、導入モードは使用されません。

例 :

```
apicl(config-graph)# service Node1 device-cluster-tenant common device-cluster ifav108-asa-2 mode FW_ROUTED
```

ステップ 5 コンシューマ コネクタを追加します。

```
connector connector_type [cluster-interface interface_type]
```

パラメータ	説明
コネクタ	サービス グラフ内のコネクタのタイプ。値は次のとおりです。 <ul style="list-style-type: none"> • provider • consumer
cluster-interface	デバイス クラスター インターフェイスのタイプ。値は次のとおりです。 <ul style="list-style-type: none"> • provider • consumer テナント <code>Common</code> 内のサービス グラフ テンプレートの場合は、このパラメータを指定しないでください。

例：

```
apic1(config-service)# connector consumer cluster-interface consumer
```

ステップ 6 テナントをコネクタに関連付け、コネクタ コンフィギュレーション モードを終了します。

```
1417-peer tenant tenant_name out L3OutExternal epg epg_name
  redistribute redistribute_property
exit
```

パラメータ	説明
テナント	コネクタに関連付けるテナントの名前。
out	レイヤ 3 Outside の名前。
epg	エンドポイント グループの名前。
redistribute	再配布プロトコルのプロパティ。

例：

```
apic1(config-connector)# 1417-peer tenant t1 out L3OutExternal epg L3ExtNet
  redistribute connected,ospf
apic1(config-connector)# exit
```

ステップ 7 プロバイダーに対してステップ 5 と 6 を繰り返します。

例：

```
apic1(config-service)# connector provider cluster-interface provider
apic1(config-connector)# 1417-peer tenant t1 out L3OutInternal epg L3IntNet
  redistribute connected,ospf
apic1(config-connector)# exit
```

ステップ 8 (任意) ルータを追加し、ノード コンフィギュレーション モードを終了します。

```
rtr-cfg router_ID
exit
```

パラメータ	説明
rtr-cfg	ルータの ID。

テナント `Common` でサービス グラフ テンプレートを作成する場合は、この手順をスキップします。

例：

```
apicl(config-service)# rtr-cfg router-id1
apicl(config-service)# exit
```

ステップ 9 1つの接続をコンシューマ コネクタに、もう1つをプロバイダー コネクタに関連付けてから、サービス グラフ コンフィギュレーション モードを終了します。

```
connection connection_name terminal terminal_type service node_name
connector connector_type
exit
```

パラメータ	説明
connection	コネクタに関連付ける接続の名前。
terminal	端末のタイプ。値は次のとおりです。 <ul style="list-style-type: none"> • provider • consumer
service	サービス グラフのノードの名前。
コネクタ	コネクタのタイプ。値は次のとおりです。 <ul style="list-style-type: none"> • provider • consumer

例：

```
apicl(config-graph)# connection C1 terminal consumer service Node1 connector consumer
apicl(config-graph)# connection C2 terminal provider service Node1 connector provider
apicl(config-graph)# exit
```

ステップ 10 コンフィギュレーション モードを終了します。

例：

```
apicl(config-tenant)# exit
apicl(config)# exit
```

REST API を使用したサービス グラフ テンプレートの設定

次の REST API を使用してサービス グラフ テンプレートを設定できます。

```

<polUni>
  <fvTenant dn="uni/tn-acme" name="acme">
    <!--L3 Network-->
    <fvCtx name="MyNetwork"/>
    <!-- Bridge Domain for MySrvr EPG -->
    <fvBD name="MySrvrBD">
      <fvRsCtx tnFvCtxName="MyNetwork" />
      <fvSubnet ip="10.10.10.10/24">
        </fvSubnet>
      </fvBD>
    <!-- Bridge Domain for MyClnt EPG -->
    <fvBD name="MyClntBD">
      <fvRsCtx tnFvCtxName="MyNetwork" />
      <fvSubnet ip="20.20.20.20/24">
        </fvSubnet>
      </fvBD>
    <fvAp dn="uni/tn-acme/ap-MyAP" name="MyAP">
      <fvAEPg dn="uni/tn-acme/ap-MyAP/epg-MyClnt" name="MyClnt">
        <fvRsBd tnFvBDName="MySrvrBD" />
        <fvRsDomAtt tDn="uni/vmmp-Vendor1/dom-MyVMs" />
        <fvRsProv tnVzBrCPName="webCtrct">
          </fvRsProv>
        <fvRsPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/21]"
encap="vlan-202"/>
        <fvRsPathAtt tDn="topology/pod-1/paths-18/pathep-[eth1/21]"
encap="vlan-202"/>
        </fvAEPg>
      <fvAEPg dn="uni/tn-acme/ap-MyAP/epg-MySRVR" name="MySRVR">
        <fvRsBd tnFvBDName="MyClntBD" />
        <fvRsDomAtt tDn="uni/vmmp-Vendor1/dom-MyVMs" />
        <fvRsCons tnVzBrCPName="webCtrct">
          </fvRsCons>
        <fvRsPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/21]"
encap="vlan-203"/>
        <fvRsPathAtt tDn="topology/pod-1/paths-18/pathep-[eth1/21]"
encap="vlan-203"/>
        </fvAEPg>
      </fvAp>
    </fvTenant>
  </polUni>

```

REST API を使用したセキュリティ ポリシーの作成

次の REST API を使用してセキュリティ ポリシーを作成することができます。

```

<polUni>
  <fvTenant dn="uni/tn-acme" name="acme">
    <vzFilter name="HttpIn">
      <vzEntry name="e1" prot="6" dToPort="80"/>
    </vzFilter>
    <vzBrCP name="webCtrct">
      <vzSubj name="http">
        <vzRsSubjFiltAtt tnVzFilterName="HttpIn"/>
      </vzSubj>
    </vzBrCP>
  </fvTenant>
</polUni>

```

