Using a Service Graph Template

Associating Service Graph Templates with Contracts and EPGs Using the GUI

You must associate the service graph templates with contracts and endpoint groups (EPGs) using the GUI.

Note You can use only the GUI to associate a service graph template with contracts and EPGs.

See Using the GUI for the procedure for associating service graph templates with contracts and EPGs.

Creating a Service Graph Template Using the NX-OS-Style CLI

The following procedure creates a service graph template.

Step 1 Enter the configure mode.

Example:
```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.
```
tenant tenant_name
```

Example:

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

Step 3 Associate a service graph to the template.

```
l4l7 graph graph_name contract contract_name
```

Parameter	Description
graph	The name of the service graph template.
contract	The name of the contract to use with the service graph template.

Example:

```
apic1(config-tenant)# l4l7 graph GraphL3asa contract ContractL3ASA
```

Step 4 Add a function node.

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

Parameter	Description
service	The name of the service node to add.
device-cluster-tenant	The tenant from which to import the device cluster. Specify this only if the device-cluster is not in the same tenant in which the graph is being configured.
device-cluster	Name of the device cluster to use for this service node.
mode	The deployment mode. Possible values are: • ADC_ONE_ARM—Specifies one-arm mode. • ADC_TWO_ARM—Specifies two-arm mode. • FW_ROUTED—Specifies routed (GoTo) mode. • FW_TRANS—Specifies transparent (GoThrough) mode. • OTHERS If the mode is not specified, then a deployment mode is not used.

Example:

```
apic1(config-graph)# service Node1 device-cluster-tenant common device-cluster ifav108-asa-2 mode
FW_ROUTED
```

Step 5 Add the consumer connector.

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

Parameter	Description
connector	The type of the connector in the service graph. Possible values are: • provider • consumer
cluster-interface	The type of the device cluster interface. Possible values are: • provider • consumer Do not specify this parameter if you are a service graph template in tenant `Common`.

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

Step 6 Associate a tenant with the connector and then exit the connector configuration mode.
```
l4l7-peer tenant tenant_name out L3OutExternal epg epg_name
  redistribute redistribute_property
exit
```

Parameter	Description
tenant	The name of the tenant to associate with the connector.
out	The name of the Layer 3 outside.
epg	The name of the endpoint group.
redistribute	The properties of the redistribute protocol.

Example:
```
apic1(config-connector)# l4l7-peer tenant t1 out L3OutExternal epg L3ExtNet
  redistribute connected,ospf
apic1(config-connector)# exit
```

Step 7 Repeat steps 5 and 6 for the provider.

Example:
```
apic1(config-service)# connector provider cluster-interface provider
apic1(config-connector)# l4l7-peer tenant t1 out L3OutInternal epg L3IntNet
```

```
  redistribute connected,ospf
apic1(config-connector)# exit
```

Step 8 (Optional) Add a router and then exit the node configuration mode.

```
rtr-cfg router_ID
exit
```

Parameter	Description
rtr-cfg	The ID of the router.

Skip this step if you are creating a service graph template in tenant `Common`.

Example:

```
apic1(config-service)# rtr-cfg router-id1
apic1(config-service)# exit
```

Step 9 Associate a connection with a consumer connector and another with a provider connector, and then exit the service graph configuration mode.

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

Parameter	Description
connection	The name of the connection to associate with the connector.
terminal	The type of the terminal. Possible values are: • provider • consumer
service	The name of the node of the service graph.
connector	The type of the connector. Possible values are: • provider • consumer

Example:

```
apic1(config-graph)# connection C1 terminal consumer service Node1 connector consumer
apic1(config-graph)# connection C2 terminal provider service Node1 connector provider
apic1(config-graph)# exit
```

Step 10 Exit the configuration mode.

Example:
```
apic1(config-tenant)# exit
apic1(config)# exit
```

Configuring a Service Graph Template Using the REST APIs

You can configure a service graph template using the following REST API:
```
<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>
```

Creating a Security Policy Using the REST APIs

You can create a security policy using the following REST API:
```
<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>
```