Configuring a Service Graph

About Service Graphs

The Cisco Application Centric Infrastructure (ACI) treats services as an integral part of an application. Any services that are required are treated as a service graph that is instantiated on the Cisco ACI fabric from the Cisco Application Policy Infrastructure Controller (APIC). Users define the service for the application, while service graphs identify the set of network or service functions that are needed by the application.

A service graph represents the network using the following elements:

- Function node: A function node represents a function that is applied to the traffic, such as a transform (SSL termination, VPN gateway), filter (firewalls), or terminal (intrusion detection systems). A function within the service graph might require one or more parameters and have one or more connectors.
- Terminal node: A terminal node enables input and output from the service graph.
- Connector: A connector enables input and output from a node.
- Connection: A connection determines how traffic is forwarded through the network.

After the graph is configured in the Cisco APIC, the Cisco APIC automatically configures the services according to the service function requirements that are specified in the service graph. The Cisco APIC also automatically configures the network according to the needs of the service function that is specified in the service graph, which does not require any change in the service device.

A service graph is represented as two or more tiers of an application with the appropriate service function inserted between.

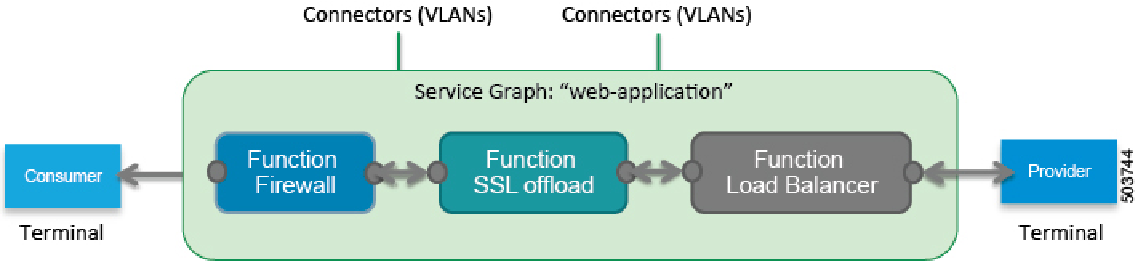
A service appliance (device) performs a service function within the graph. One or more service appliances might be required to render the services required by a graph. One or more service functions can be performed by a single-service device.

Service graphs and service functions have the following characteristics:

- Traffic sent or received by an endpoint group can be filtered based on a policy, and a subset of the traffic can be redirected to different edges in the graph.
- Service graph edges are directional.
- Taps (hardware-based packet copy service) can be attached to different points in the service graph.
- Logical functions can be rendered on the appropriate (physical or virtual) device, based on the policy.
- The service graph supports splits and joins of edges, and it does not restrict the administrator to linear service chains.
- Traffic can be reclassified again in the network after a service appliance emits it.
- Logical service functions can be scaled up or down or can be deployed in a cluster mode or 1:1 active-standby high-availability mode, depending on the requirements.

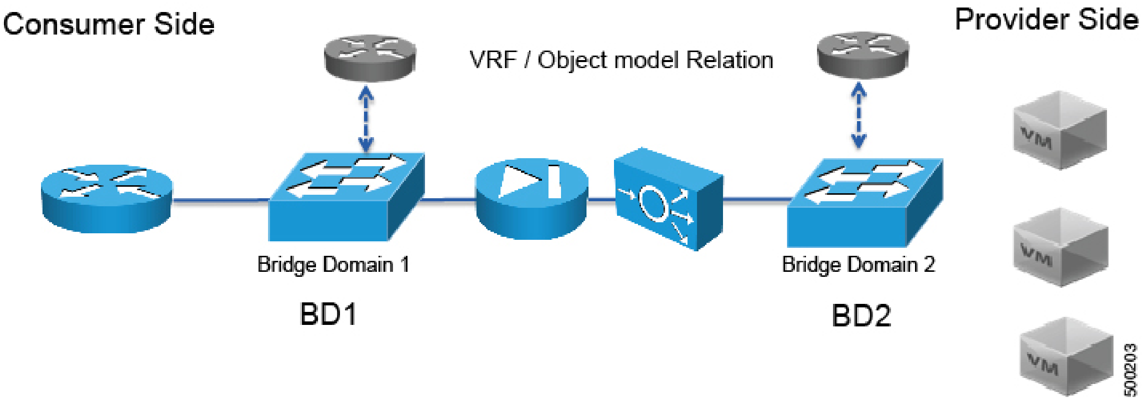
The following figure provides an example of a service graph deployment:

Figure 1: Example Service Graph Deployment



Deploying a service graph requires bridge domains and VRF instances, as shown in the following figure:

Figure 2: Bridge Domains and VRF instances of a Service Graph



Note If you have some of the legs of a service graph that are attached to endpoint groups in other tenants, when you use the **Remove Related Objects of Graph Template** function in the GUI, the Cisco APIC does not remove contracts that were imported from tenants other than where the service graph is located. The Cisco APIC also does not clean endpoint group contracts that are located in a different tenant than the service graph. You must manually remove these objects that are in different tenants.

About Function Nodes

A function node represents a single service function. A function node has function node connectors, which represent the network requirement of a service function.

The Cisco Application Policy Infrastructure Controller (APIC) only allocates the network resources and programs the VLAN/VXLAN on fabric side.

The following settings are not needed:

- `MFunc` relation
- Information about the supported function type (go-through, go-to)

The Cisco APIC needs to know the network information (LIF, CIF) for the function node. This information is needed so that the Cisco APIC can program the network appropriately on the leaf switch, and the Cisco APIC can also use this information for troubleshooting wizard purposes.

The following settings are still needed:

- `LDevCtx` to enable the selection of `LDevVip` during graph instantiation
- `LIfCtx` to enable the selection of `LIf` during graph instantiation
- Bridge domain in `LIfCtx`
- Route peering in `LIfCtx`
- Subnet in `LIfCtx`

Note For a Cisco ACI Multi-Site configuration, up to 2 nodes can be deployed in a service graph. For a non-Cisco ACI Multi-Site configuration, up to 5 nodes can be deployed in a service graph.

About Function Node Connectors

A function node connector connects a function node to the service graph and is associated with the appropriate bridge domain and connections based on the graph's connector's subset. Each connector is associated with a VLAN or Virtual Extensible LAN (VXLAN). Each side of a connector is treated as an endpoint group (EPG), and whitelists are downloaded to the switch to enable communication between the two function nodes.

About Service Graph Connections

A service graph connection connects one function node to another function node.

About Terminal Nodes

Terminal nodes connect a service graph with the contracts. You can insert a service graph for the traffic between two application endpoint groups (EPGs) by connecting the terminal node to a contract. Once connected, traffic between the consumer EPG and provider EPG of the contract is redirected to the service graph.

Guidelines and Limitations for Configuring Service Graph

The following are guidelines and limitations for configuring Service Graph.

- A service-graph related configuration such as
  - A bridge domain (if used with a service graph) and service graph template should not contain the string “C-“ as part of its name.
  - A logical device should not contain the string “N-“ as part of its name.

Configuring a Service Graph Template Using the GUI

A service graph template is a sequence of Layer 4 to Layer 7 services functions, Layer 4 to Layer 7 services devices, or copy devices and their associated configuration. The service graph template must be associated with a contract to be "rendered"—or configured—on the Layer 4 to Layer 7 services device or copy device, and on the fabric.

Before you begin

You must have configured a tenant.

Step 1 On the menu bar, choose **Tenants** > **All Tenants**.

Step 2 In the Work pane, double-click the tenant's name.

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Service Graph Templates**.

Step 4 In the Navigation pane, right-click **Service Graph Templates** and choose **Create a L4-L7 Service Graph Template**.

The **Create L4-L7 Service Graph Template** dialog box appears.

Step 5 If necessary, create one or more Layer 4 to Layer 7 services devices or copy devices:

a) Click the drop-down arrow in the **Device Clusters** pane of the **Create L4-L7 Service Graph Template** dialog box and choose **Create L4-L7 Devices** or **Create Copy Devices**.
The corresponding dialog box appears.

b) Follow the dialog box by entering the appropriate values displayed in the dialog box and clicking **Next** until finished.

Note For an explanation of a field in a dialog box, click the help icon in the top-right corner to display the help file.

c) When finished, click **Finish**.

You return to the **Create L4-L7 Service Graph Template** dialog box.

Step 6 Enter the appropriate values in the fields of the **Create L4-L7 Service Graph Template** dialog box.

Note For an explanation of a field in a dialog box, click the help icon in the top-right corner to display the help file.

Step 7 (Optional) (Only for cloning an existing service graph template) If you want to remove any of the nodes from the cloned service graph template, right-click a node that you want to remove and choose **Remove Node**.

Step 8 To create a service node, drag a Layer 4 to Layer 7 services device from the **Device Clusters** section and drop it between the consumer endpoint group and provider endpoint group. To create a copy node, drag and drop a copy device. This step is optional if you cloned an existing service graph template and the service graph template has all of the nodes that you want to use.

You can drag and drop multiple devices to create multiple nodes. The maximum number of service nodes is 3, although you can drag and drop greater numbers of other devices.

The location where you drop a copy device becomes the point in the data flow from where the copy device copies the traffic.

Step 9 If you created one or more service nodes, in the ***device_name*** **Information** section for each Layer 4 to Layer 7 services device, complete the fields. The fields vary depending on the device type.

Note For an explanation of a field, click the help icon in the top-right corner to display the help file.

Step 10 When finished, click **Submit**.

Step 11 (Optional) In the **Navigation** pane, click the service graph template.
The work pane displays a graphic topology of the service graph template.

Configuring a Service Graph Template Using the REST APIs

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

```
<polUni>
  <fvTenant name="acme">
    <vnsAbsGraph name="G1">
      <vnsAbsTermNodeCon name="Input1">
        <vnsAbsTermConn name="C1">
            </vnsAbsTermConn>
      </vnsAbsTermNodeCon>
      <vnsAbsNode name="Node" funcType="GoTo">
        <vnsRsDefaultScopeToTerm
          tDn="uni/tn-acme/AbsGraph-G1/AbsTermNodeProv-Output1/outtmnl"/>
        <vnsAbsFuncConn name="inside">
          <vnsRsMConnAtt
            tDn="uni/infra/mDev-Insieme-Generic-1.0/mFunc-SubnetFunc/mConn-external"/>
        </vnsAbsFuncConn>
        <vnsAbsFuncConn name="outside">
          <vnsRsMConnAtt
```

```
            tDn="uni/infra/mDev-Insieme-Generic-1.0/mFunc-SubnetFunc/mConn-internal"/>
        </vnsAbsFuncConn>
        <vnsAbsDevCfg>
          <vnsAbsFolder key="oneFolder" name="f1">
            <vnsAbsParam key="oneParam" name="p1" value="v1"/>
          </vnsAbsFolder>
        </vnsAbsDevCfg>
        <vnsAbsFuncCfg>
          <vnsAbsFolder key="folder" name="folder1" devCtxLbl="C1">
            <vnsAbsParam key="param" name="param" value="value"/>
          </vnsAbsFolder>
          <vnsAbsFolder key="folder" name="folder2" devCtxLbl="C2">
            <vnsAbsParam key="param" name="param" value="value"/>
          </vnsAbsFolder>
        </vnsAbsFuncCfg>
        <vnsRsNodeToMFunc tDn="uni/infra/mDev-Insieme-Generic-1.0/mFunc-SubnetFunc"/>
      </vnsAbsNode>
      <vnsAbsTermNodeProv name="Output1">
        <vnsAbsTermConn name="C6">
            </vnsAbsTermConn>
      </vnsAbsTermNodeProv>
      <vnsAbsConnection name="CON1">
        <vnsRsAbsConnectionConns
          tDn="uni/tn-acme/AbsGraph-G1/AbsTermNodeCon-Input1/AbsTConn"/>
       <vnsRsAbsConnectionConns tDn="uni/tn-acme/AbsGraph-G1/AbsNode-Node/AbsFConn-inside"/>

      </vnsAbsConnection>
      <vnsAbsConnection name="CON3">
       <vnsRsAbsConnectionConns tDn="uni/tn-acme/AbsGraph-G1/AbsNode-Node/AbsFConn-outside"/>

        <vnsRsAbsConnectionConns
          tDn="uni/tn-acme/AbsGraph-G1/AbsTermNodeProv-Output1/AbsTConn"/>
      </vnsAbsConnection>
    </vnsAbsGraph>
  </fvTenant>
</polUni>
```

Applying a Service Graph Template to Endpoint Groups Using the GUI

The following procedure explains how to apply a service graph template to endpoint groups:

Before you begin

You must have created the following things:

- Application endpoint groups
- A service graph template

Step 1 On the menu bar, choose **Tenants** > **All Tenants**.

Step 2 In the Work pane, double click the tenant's name.

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Service Graph Templates** > ***template_name***.

Step 4 In the Navigation pane, right-click on the ***template_name*** that you want to apply to EPGs and choose **Apply L4-L7 Service Graph Template**.

The **Apply L4-L7 Service Graph Template To EPGs** dialog box appears. You will be associating a Layer 4 to Layer 7 service graph template to your consumer and provider endpoint groups.

Step 5 Configure a contract in the **Apply L4-L7 Service Graph Template To EPGS STEP 1> Contract** dialog box by entering the appropriate values:

a) If you are configuring an intra-EPG contract, place a check in the **Configure an Intra-EPG Contract** check-box and choose the EPG and network combination from the **EPG / Network** drop-down list.

b) If you are configuring a standard contract, choose the consumer/provider EPGs and network combinations in the appropriate drop-down lists.

c) Create a new contract or choose an existing one by clicking the appropriate radio button in the **Contract** field. If you select **Create A New Contract** and want to configure the filters for it, remove the check from the **No Filter (Allow All Traffic)** check-box. Click **+** to add filter entries and **Update** when complete.

Note For copy service graphs, contracts can only be used multiple times if they are applied to L3Out EPGs. Internal EPGs require an unshared contract.

Step 6 Click **Next**.

The **STEP 2 > Graph** dialog appears.

Step 7 In the ***device_name*** **Information** section, configure the required fields represented with a red box.

Note To include the connector in a preferred group (endpoint to endpoint communication without a contract), choose a configured policy from the **Service EPG Policy** drop-down list.

Step 8 Click **Next**.

The **STEP 3 > device_name Information** dialog appears.

Step 9 Click **Finish**.
You now have an active service graph template.

Applying a Service Graph Template to an Endpoint Security Group Using the GUI

The following procedure explains how to apply a service graph template to an endpoint security group (ESG):

Before you begin

You must have created the following things:

- ESGs
- A service graph template

Step 1 On the menu bar, choose **Tenants** > **All Tenants**.

Step 2 In the Work pane, double click the tenant's name.

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Service Graph Templates** > ***template_name***.

Step 4 In the Navigation pane, right-click on the ***template_name*** that you want to apply to EPGs and choose **Apply L4-L7 Service Graph Template**.

The **Apply L4-L7 Service Graph Template To EPGs** dialog box appears. You will be associating a Layer 4 to Layer 7 service graph template to your consumer and provider endpoint security groups.

Step 5 Configure a contract in the **Apply L4-L7 Service Graph Template To EPGS STEP 1> Contract** dialog box by entering the appropriate values:

a) Select **Endpoint Security Group** as the enpdoint group type.
b) If you are configuring a standard contract, choose the consumer/provider ESGs and network combinations in the appropriate drop-down lists.
c) Create a new contract or choose an existing one by clicking the appropriate radio button in the **Contract** field. If you select **Create A New Contract** and want to configure the filters for it, remove the check from the **No Filter (Allow All Traffic)** check-box. Click **+** to add filter entries and **Update** when complete.

Step 6 Click **Next**.

The **STEP 2 > Graph** dialog appears.

Step 7 In the ***device_name*** **Information** section, configure the required fields represented with a red box.

Step 8 Click **Next**.

The **STEP 3 > device_name Information** dialog appears.

Step 9 Click **Finish**.
You now have an active service graph template.

Applying a Service Graph Template with a Contract Using the NX-OS-Style CLI

The following procedure applies a service graph template with a contract using the NX-OS-style CLI.

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 Add a service graph.

```
l4l7 graph graph_name [contract contract_name]
```

Parameter	Description
graph	Name of the service graph.

Parameter	Description
contract	Name of the contract that is associated with this service graph instance. Specify the contract only if you want to create the service graph instance. You can simply configure a service graph (equivalent to the service graph template) without instantiating it.

Example:

```
apic1(config-tenant)# l417 graph G2 contract C2
```

Step 4 Add a node (service) in the service graph.

```
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`: Specifies any other deployment mode. If the mode is not specified, then a deployment mode is not used.

Example:

The following example adds node `N1` to the device cluster `D4`, which is from tenant `t1`:

```
apic1(config-graph)# service N1 device-cluster-tenant t1 device-cluster D4
```

The following example adds node `N1` to the device cluster `D4`, which is from tenant `t1`, and uses the routed deployment mode:

```
apic1(config-graph)# service N1 device-cluster-tenant t1 device-cluster D4 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

Parameter	Description
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 If the service interface is in a bridge domain, perform the following substeps:

a) Configure the bridge domain for the connectors by specifying the bridge domain information and tenant where the bridge domain is present.

```
bridge-domain tenant tenant_name name bridge_domain_name
```

Parameter	Description
tenant	Tenant that owns the bridge domain. You can only specify a bridge domain from same tenant or tenant `Common`. For example if you are in tenant `t1`, then you cannot specify the bridge domain from tenant `t2`.
name	Name of the bridge domain.

Example:

```
apic1(config-connector)# bridge-domain tenant t1 name bd2
```

b) Configure the direct server return (DSR) virtual IP address (VIP) for the connector.

```
dsr-vip ip_address
```

If you specify the DSR VIP, the Application Policy Infrastructure Controller (APIC) does not learn the VIP.

Parameter	Description
dsr-vip	The virtual IP address of the DSR for the connector.

Example:

```
apic1(config-connector)# dsr-vip 192.168.10.100
```

Step 7 If the service interface is in an L3Out, perform the following substeps:

a) 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.

Parameter	Description
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
```

b) Repeat steps 5 and 7a 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
```

c) (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 8 Configure connections for the consumer and provider and exit the service graph configuration mode.

```
connection connection_name {terminal terminal_type service node_name connector connector_type} |
  {intra_service service1 node_name connector1 connector_type service2 node_name connector2
connector_type}
exit
```

Parameter	Description
connection	The name of the connection.
terminal	Connects a service node to the terminal. Specifies the type of the terminal. Possible values are: • provider • consumer
service service1 service2	The name of the service node to add. service is used only with terminal. service1 and service2 are used only with intra_service.

Parameter	Description
connector connector1 connector2	The type of the connector. Possible values are: • provider • consumer connector is used only with terminal. connector1 and connector2 are used only with intra_service.
intra_service	Connects a service node to another node.

Example:

The following example configures the connections of a single node graph:

```
apic1(config-graph)# connection CON1 terminal consumer service N1 connector consumer
apic1(config-graph)# connection CON2 terminal provider service N2 connector provider
apic1(config-graph)# exit
```

The following example configures the connections of a two node graph:

```
apic1(config-graph)# connection CON1 terminal consumer service N1 connector consumer
apic1(config-graph)# connection CON2 intra_service service1 N1 connector1 provider service2 N2
connector2 consumer
apic1(config-graph)# connection CON3 terminal provider service N2 connector provider
apic1(config-graph)# exit
```

Step 9 Exit the configuration mode.

Example:

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