



Configuring a Service Graph

- [Configuring a Service Graph, page 1](#)

Configuring a Service Graph

About Service Graphs

A service graph is an ordered set of function nodes between a set of terminals, which identifies a set of network service functions that are required by an application. Service functions within a graph are automatically provisioned on a service device that is based on an application's requirements.

For more information about function nodes, see .

You can define a service graph by using the GUI, CLI, or the Application Policy Infrastructure Controller (APIC). Configuring a service device through the APIC does not require changes on a service device.

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. For more information about function node connectors, see .

A function node within a service graph can require one or more parameters. The parameters can be specified by an endpoint group (EPG), an application profile, or a tenant context. Parameters can also be assigned at the time that you define a service graph. The parameter values can be locked to prevent any additional changes.

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.

About Service Graph Configuration Parameters

A service graph can have configuration parameters, which are specified by the device package. Configuration parameters can also be specified by an EPG, application profile, or tenant context. A function node within a service graph can require one or more configuration parameters. The parameter values can be locked to prevent any additional changes.

When you configure a service graph and specify the values of the configuration parameters, the Application Policy Infrastructure Controller (APIC) passes the parameters to the device script that is within the device package. The device script converts the parameter data to the configuration that is downloaded onto the device.

Configuring a Service Graph

After you register a device, you can create service graphs using that device and all the functions that device has exposed. The service graph can be created under tenant common or can be tenant specific. This can be done by the provider administrator or by the tenant administrator within its own tenancy.

Step 1 In the **Navigation** pane, click **L4-L7 Services > Service Graphs**.

Step 2 Choose **Action > Create L4-L7 Service Graph**. The **CREATE L4-L7 SERVICE GRAPH** dialog box appears. The left pane lists the service devices that the Application Policy Infrastructure Controller (APIC) knows about and lists the service functions that are provided by the devices. The APIC knows about these devices from the device package that you imported.

Step 3 Complete the following fields:

Name	Description
Name field	The name of the service graph.

Name	Description
Description field	The description of the service graph.

- Step 4** Drag and drop a service function from the left pane to the right pane to add that function to the service graph.
- Step 5** Based on the device package, click on the function connectors, int and ext, to map them respectively to the meta connectors, Inside and Outside. the meta connectors, Inside and Outside, are device specific.
- Step 6** Connect the terminal nodes to the connectors of the service function by clicking a terminal node and dragging a connection to a connector of the function. Repeat this action for each terminal node that you want to connect to the function. Terminal nodes represent the connectivity to the application EPG.
- Step 7** Click the function. The **Config Profile** dialog box appears, prompting you to choose a profile for the service node. You can configure the service function parameters at this stage, or you can do so at the graph attachment time when a graph is associated with the contract.
- Step 8** Once you have completed configuring the service function parameters, return to the **CREATE L4-L7 SERVICE GRAPH** dialog box and click **Submit** to create the graph. The **Service Graph** dialog box lists the new graph that you created.

Configuring Using REST APIs to Create a Service Graph

```

<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>

```

Configuring Using CLI Commands

```

# l4-l7-service-graph

cd '/aci/tenants/asa_tenant1/l4-l7-services/service-graphs'
mocreate 'g1'
moconfig commit

# consumer-terminal-node

cd '/aci/tenants/asa_tenant1/l4-l7-services/service-graphs/g1/consumer-terminal-nodes'
mocreate 'Input1'
moconfig commit

# terminal-connector

cd '/aci/tenants/asa_tenant1/l4-l7-services/service-graphs/g1/consumer-terminal-nodes/Input1'
mocreate terminal-connector
cd 'terminal-connector'
moset name 'C1'
moconfig commit

# provider-terminal-node

cd '/aci/tenants/asa_tenant1/l4-l7-services/service-graphs/g1/provider-terminal-nodes'
mocreate 'Output1'
moconfig commit

# terminal-connector

cd '/aci/tenants/asa_tenant1/l4-l7-services/service-graphs/g1/provider-terminal-nodes/Output1'
mocreate terminal-connector
cd 'terminal-connector'
moset name 'C6'
moconfig commit

# function-node

cd '/aci/tenants/asa_tenant1/l4-l7-services/service-graphs/g1/function-nodes'
mocreate 'ASA_FW'
cd 'ASA_FW'
moset function-name 'l4-l7-services/packages/CISCO-ASA-1.0.1.37/functions/Firewall'
moconfig commit

# device-config

cd '/aci/tenants/asa_tenant1/l4-l7-services/service-graphs/g1/function-nodes/ASA_FW'
mocreate device-config

```

```

moconfig commit

# function-connector

cd '/aci/tenants/asa_tenant1/14-17-services/service-graphs/g1/function-nodes/ASA_FW'
mcreate function-connector 'external'
cd 'function-connector-external'
moset meta-connector 'uni/infra/mDev-CISCO-ASA-1.0.1.37/mFunc-Firewall/mConn-external'
moconfig commit

# function-connector

cd '/aci/tenants/asa_tenant1/14-17-services/service-graphs/g1/function-nodes/ASA_FW'
mcreate function-connector 'internal'
cd 'function-connector-internal'
moset meta-connector 'uni/infra/mDev-CISCO-ASA-1.0.1.37/mFunc-Firewall/mConn-internal'
moconfig commit

# default-scope-to-term

cd '/aci/tenants/asa_tenant1/14-17-services/service-graphs/g1/function-nodes/ASA_FW'
mcreate default-scope-to-term
cd 'default-scope-to-term'
moset target-dn 'uni/tn-asa_tenant1/AbsGraph-g1/AbsTermNodeProv-Output1/outtmnl'
moconfig commit

# connection

cd '/aci/tenants/asa_tenant1/14-17-services/service-graphs/g1/connections'
mcreate 'CON2'
moconfig commit

# relation-from-abstract-connection-to-abstract-connectors

cd '/aci/tenants/asa_tenant1/14-17-services/service-graphs/g1/connections/CON2'
mcreate
'tenants/asa_tenant1/14-17-services/service-graphs/g1/function-nodes/ASA_FW/function-connector-internal'
moconfig commit

# relation-from-abstract-connection-to-abstract-connectors

cd '/aci/tenants/asa_tenant1/14-17-services/service-graphs/g1/connections/CON2'
mcreate
'tenants/asa_tenant1/14-17-services/service-graphs/g1/provider-terminal-nodes/Output1/terminal-connector'
moconfig commit

# connection

cd '/aci/tenants/asa_tenant1/14-17-services/service-graphs/g1/connections'
mcreate 'CON1'
moconfig commit

# relation-from-abstract-connection-to-abstract-connectors

cd '/aci/tenants/asa_tenant1/14-17-services/service-graphs/g1/connections/CON1'
mcreate
'tenants/asa_tenant1/14-17-services/service-graphs/g1/consumer-terminal-nodes/Input1/terminal-connector'
moconfig commit

# relation-from-abstract-connection-to-abstract-connectors

cd '/aci/tenants/asa_tenant1/14-17-services/service-graphs/g1/connections/CON1'
mcreate
'tenants/asa_tenant1/14-17-services/service-graphs/g1/function-nodes/ASA_FW/function-connector-external'
moconfig commit
admin@ifav15-ifc2:g1>

```

