



# Deploying Layer 4 to Layer 7 Services

- [Overview, on page 1](#)
- [Deploying a Service Graph, on page 5](#)

## Overview

The Cisco Cloud Network Controller enables you to deploy Layer 4 to Layer 7 service devices to the public cloud. This initial release supports application load balancer (ALB) deployments in Amazon Web Services (AWS).

## About Application Load Balancers

An application load balancer (ALB) is a Layer 7 load balancer that inspects packets and creates access points to HTTP and HTTPS headers. It also identifies the load and spreads it out to the targets with higher efficiency. You deploy an ALB using a service graph, which enables you to define how you want traffic to come into the network, the devices that the traffic passes through, and how the traffic leaves the network. You specify these actions by configuring one or more listeners.

Listeners enable you to specify the ports and protocols (HTTP or HTTPS) that the ALB accepts traffic on. When specifying HTTPS, you also choose a security policy and an SSL certificate.



---

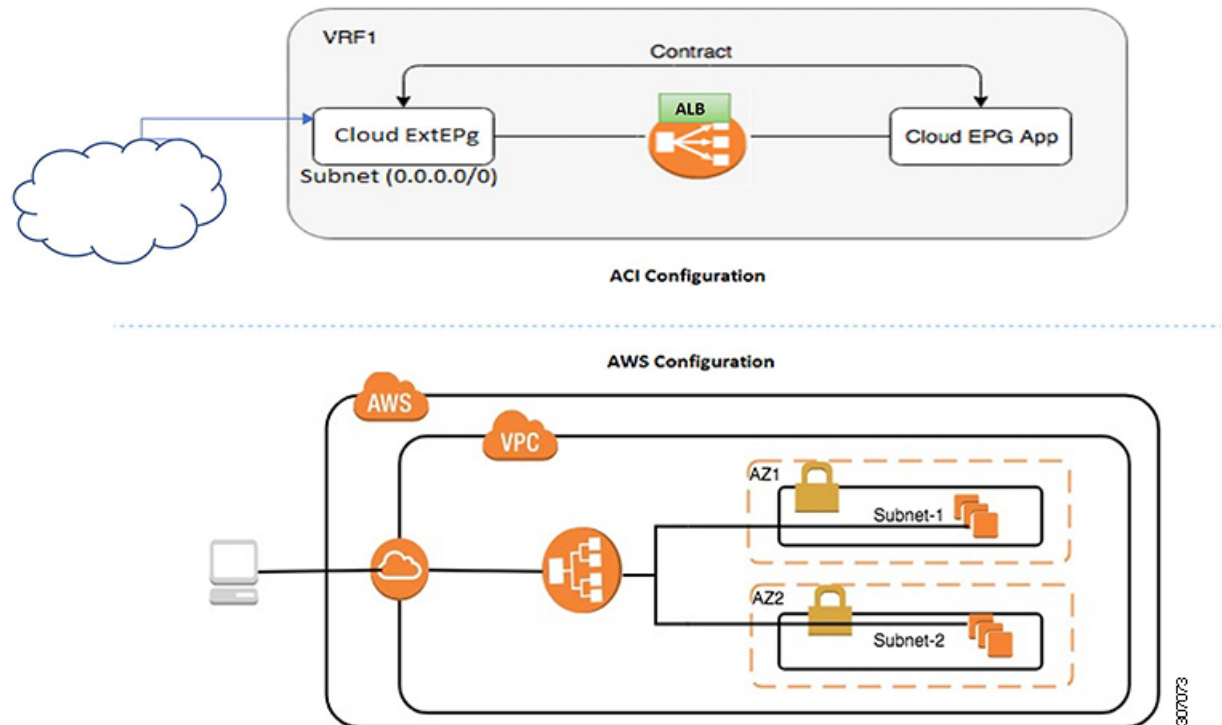
**Note** A listener can have multiple certificates.

---

All listeners require you to configure at least one rule (a default rule, which does not have a condition). Rules enable you to specify the action that the load balancer takes when a condition is met. For example, you can create a rule that redirects traffic to a specified URL when a request is made to a specified hostname or path.

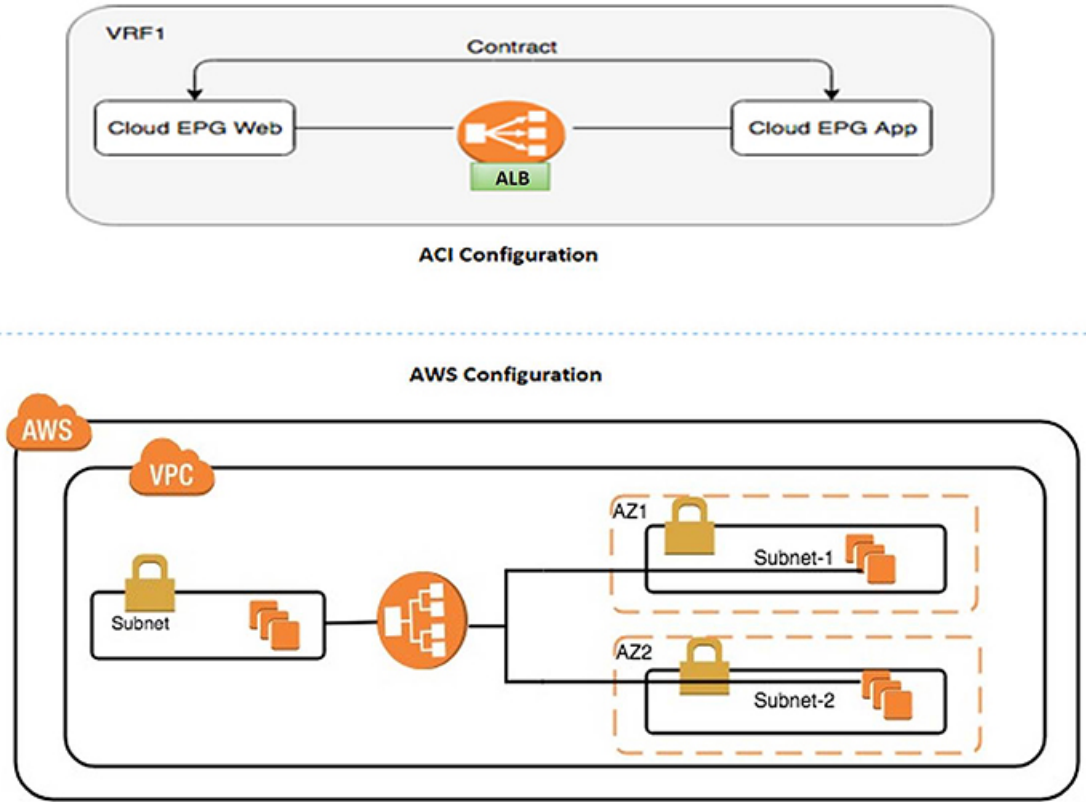
There are two deployment types: internet-facing and internal-facing. An internet-facing deployment inserts the ALB as a service between the consumer external EPG and the provider cloud EPG. The following figure shows the contract configuration within the VRF and the ALB as a service inserted between the consumer external EPG and the provider cloud EPG.

Figure 1: Internet-Facing Deployment



An internal-facing deployment inserts the ALB as a service between the consumer cloud EPG and the provider cloud EPG. The following figure shows the contract configuration within the VRF and the ALB as a service inserted between the consumer cloud EPG and provider cloud EPG.

Figure 2: Internal-Facing Deployment



**Note** You can find more information about ALBs in the documentation on the AWS website.

## Dynamic Server Attachment to Server Pool

Servers in the server pool or target group are dynamically added. You do not need to specify the IP addresses or instance Ids for the targets. The relation from a listener rule to a provider cloud EPG is used for the dynamic selection of endpoints. The relation is also used for adding the endpoints to the target group. By default, the endpoints are registered with the port number 80.

Based on the target group-to-security group association that is provided in the ALB, and the EPG (security group) of the endpoint, the EC2 instance (server) is associated to the target group dynamically on the target group's default port. Alternatively, instead of registering the EC2 instance on the target group port, you can attach the custom port by specifying the ports in the following table:

Table 1: Custom Port-Based Attachment

Provider EPG	Ports
EPGMap:<Epg1DN>	9090

Provider EPG	Ports
EPGMap:<Epg2DN>	9091, 9099

You can specify EPGMap:<EpgDN> as the tag and the list of ports to be registered on the target group as a list separated by commas.

## About Service Graphs

The Cisco Application Centric Infrastructure (ACI) treats services as a 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 APIC. You define the service for the application while service graphs identify the set of network or service functions that the application needs.

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 load balancer. 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.

After the graph is configured, 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 them.

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. A single-service device can perform one or more service functions.

Service graphs and service functions have the following characteristics:

- Traffic sent from specific endpoint groups can be redirected based on a policy.
- Service graph redirection is directional. In other words, redirection can be applied to both traffic directions or either one of them.
- Logical functions can be rendered on the appropriate 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.

By using a service graph, you can install a service, a load balancer, once and deploy it multiple times in different logical topologies. Each time the graph is deployed, Cisco ACI takes care of changing the configuration on the service device to enable the forwarding in the new logical topology.

## 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.

A function node within a service graph requires the following parameters:

- A tenant
- A cloud context profile with subnets in two availability zones

Function parameters can be specified when the service graph is rendered. For example, if the function node is a load balancer, the listener and its rule can be specified for the function node at the time the graph is rendered.

## About Terminal Nodes

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

## Deploying a Service Graph

The service graph enables you to define how traffic flows between devices, how the traffic comes into the network, which devices the traffic passes through, and how the traffic leaves the network.

Before you can configure a service graph, you must first configure the following:

1. A tenant
2. A cloud context profile
3. Subnets
4. An application profile
5. A consumer EPG
6. A provider EPG
7. A contract

## Deploying the Service Graph Using the Cisco Cloud Network Controller GUI

### Creating a Load Balancer Using the Cisco Cloud Network Controller GUI

This section explains how to create a load balancer using the Cisco Cloud Network Controller GUI.

- 
- Step 1** Click **Application Management > Services**.  
The **Services** page appears.

**Step 2** Click the Devices tab, then click **Actions > Create Device**.

The **Create Device** page appears.

**Step 3** Enter the appropriate values in each field as listed in the following *Create Device Dialog Box Fields* table then continue.

**Table 2: Create Device Dialog Box Fields**

Properties	Description
<b>General</b>	
<b>Name</b>	Enter the name of the load balancer.
<b>Tenant</b>	To choose a tenant: <ol style="list-style-type: none"> <li>Click <b>Select Tenant</b>. The <b>Select Tenant</b> dialog appears.</li> <li>From the column on the left, click to choose a tenant.</li> <li>Click <b>Select</b>. You return to the <b>Create Device</b> dialog box.</li> </ol>
<b>Settings</b>	
<b>Service Type</b>	Choose <b>Application Load Balancer</b> .
<b>Scheme</b>	Choose <b>Internal</b> or <b>Internet Facing</b> .
<b>Subnets</b>	You can specify only one subnet per availability zone. You must specify subnets from at least two availability zones to increase the availability of your load balancer. <ol style="list-style-type: none"> <li>Click <b>Add Subnet</b>. The <b>Add Subnet</b> dialog box appears.</li> <li>In the <b>Add Subnet</b> dialog box, click <b>Select Cloud Context Profile</b>. The <b>Select Cloud Context Profile</b> dialog box appears.</li> <li>In the <b>Select Cloud Context Profile</b> dialog box, select a cloud context profile, then click <b>Select</b>. You are returned to the <b>Add Subnet</b> dialog box.</li> <li>In the <b>Add Subnet</b> dialog box, click <b>Select Subnet</b>. The <b>Select Subnet</b> dialog box appears.</li> <li>In the <b>Select Subnet</b> dialog box, select a subnet, then click <b>Select</b>. You are returned to the <b>Add Subnet</b> dialog box.</li> <li>In the <b>Add Subnet</b> dialog box, click <b>Add</b>. You are returned to the <b>Create Device</b> page.</li> </ol>

**Step 4** Click **Save** when finished.

## Creating a Service Graph Template Using the Cisco Cloud Network Controller GUI

This section explains how to configure a service graph template using the Cisco Cloud Network Controller GUI.

### Before you begin

You have already created a device.

**Step 1** Click **Application Management > Services**.

The **Services** page appears.

**Step 2** Click the **Service Graphs** tab, then click **Actions > Create Service Graph**.

The **Create Service Graph** page appears.

**Step 3** Enter the appropriate values in each field as listed in the following *Create Service Graph Dialog Box Fields* table then continue.

**Table 3: Create Service Graph Dialog Box Fields**

Properties	Description
<b>General</b>	
<b>Name</b>	Enter the name of service graph template.
<b>Tenant</b>	To choose a tenant: <ol style="list-style-type: none"> <li>a. Click <b>Select Tenant</b>. The <b>Select Tenant</b> dialog appears.</li> <li>b. From the column on the left, click to choose a tenant.</li> <li>c. Click <b>Select</b>. You return to the <b>Create Service Graph</b> dialog box.</li> </ol>
<b>Description</b>	Enter a description of the service graph template.
<b>Settings</b>	

Properties	Description
Select a Device	<p>To choose a device:</p> <ol style="list-style-type: none"> <li>Drag and drop the Application Load Balancer icon to the <b>Drop Device</b> area in the service graph. The <b>Service Node</b> dialog box appears.</li> <li>Click <b>Select Application Load Balancer</b>. The <b>Select Application Load Balancer</b> dialog appears.</li> <li>From the column on the left, click to choose a device.</li> <li>Click <b>Select</b>. You return to the <b>Service Node</b> dialog box.</li> <li>Click <b>Add</b>. You return to the <b>Create Service Graph</b> window.</li> </ol>

**Step 4** Click **Save** when finished.

## Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI

This section explains how to deploy Layer 4 to Layer 7 services.

### Before you begin

- You have configured a device.
- You have configured a service graph.

**Step 1** Click the **Intent** icon. The **Intent** menu appears.

**Step 2** Click the drop-down arrow below the **Intent** search box and choose **Configuration**.

A list of **Configuration** options appear in the **Intent** menu.

**Step 3** From the **Configuration** list in the **Intent** menu, click **EPG Communication**. The **EPG Communication** dialog box appears with the **Consumer EPGs**, **Contract**, and **Provider EPGs** information.

**Step 4** To choose a contract:

- Click **Select Contract**. The **Select Contract** dialog appears.
- In the pane on the left side of the **Select Contract** dialog, click to choose a contract then click **Select**. The **Select Contract** dialog box closes.

**Step 5** To add a consumer EPG:

- Click **Add Consumer EPGs**. The **Select Consumer EPGs** dialog appears.
- In the pane on the left side of the **Select Consumer EPGs** dialog, click to place a check in a check box to choose a cloud EPG (for an internal facing load balancer) or a cloud external EPG (for an internet facing load balancer) then click **Select**. The **Select Consumer EPGs** dialog box closes.



- Step 6** To add a provider EPG:
- Click **Add Provider EPGs**. The **Select Provider EPGs** dialog appears.
  - In the pane on the left side of the **Select Provider EPGs** dialog, click to place a check in a check box to choose a provider EPG then click **Select**. The **Select Provider EPGs** dialog box closes.

- Step 7** To choose a service graph:
- From the **EPG Communication Configuration** dialog, click **Select Service Graph**. The **Select Service Graph** dialog box appears.
  - In the pane on the left side of the **Select Service Graph** dialog, click to choose a service graph then click **Select**. The **Select Service Graph** dialog box closes.

- Step 8** Under **Service Graph Preview**, click **Add Cloud Load Balancer Listener**. The **Add Cloud Load Balancer Listener** dialog appears that enables you to add listeners.

Listeners are the ports and protocols that the device will work on.

- Step 9** Enter the appropriate values in each field as listed in the following *Add Cloud Load Balancer Listener Dialog Box Fields* table then continue.

**Table 4: Add Cloud Load Balancer Listener Dialog Box Fields**

Properties	Description
<b>Name</b>	Enter the name of the listener.
<b>Port</b>	Enter the port that the device will accept traffic on.
<b>Protocol</b>	Click to choose <b>HTTP</b> or <b>HTTPS</b> .
<b>Security Policy</b>	Click the drop-down list and choose a security policy (only available when <b>HTTPS</b> is chosen).
<b>SSL Certificate</b>	<p>To choose an SSL certificate(only available when <b>HTTPS</b> is chosen):</p> <ol style="list-style-type: none"> <li>Click <b>Add SSL Certificates</b>.</li> <li>Click to place a check mark in the check box of the certificates you want to add.</li> <li>Choose a key ring: <ol style="list-style-type: none"> <li>Click <b>Select Key Ring</b>. The <b>Select Key Ring</b> dialog appears.</li> <li>From the <b>Select Key Ring</b> dialog, click to choose a key ring in the left column then click <b>Select</b>. The <b>Select Key Ring</b> dialog box closes.</li> </ol> </li> <li>Click the <b>Certificate Store</b> drop-down list and choose a certificate.</li> </ol> <p><b>Note</b> A listener can have multiple certificates.</p>

Properties	Description
<b>Add Rule</b>	To add rule settings to the device listener, click <b>Add Rule</b> . A new row appears in the <b>Rules</b> list and the <b>Rules Settings</b> fields are enabled.

Properties	Description
Rule Settings	

Properties	Description
	<p>The <b>Rule Settings</b> pane contains the following options:</p> <ul style="list-style-type: none"> <li>• <b>Name</b>—Enter a name for the rule.</li> <li>• <b>Host</b>—Enter a hostname to create a host-based condition. When a request is made for this hostname, the action you specify is taken.</li> <li>• <b>Path</b>—Enter a path to create a path-based condition. When a request is made for this path, the action you specify is taken.</li> <li>• <b>Type</b>—The action type tells the device which action to take. The action type options: <ul style="list-style-type: none"> <li>• <b>Return fixed response</b>—Returns a response using the following options: <ul style="list-style-type: none"> <li>• <b>Fixed Response Body</b>—Enter a response message.</li> <li>• <b>Fixed Response Code</b>—Enter a response code.</li> <li>• <b>Fixed response Content-Type</b>—Choose a content type.</li> </ul> </li> <li>• <b>Forward</b>—Forwards traffic using the following options: <ul style="list-style-type: none"> <li>• <b>Port</b>—Enter the port that the device will accept traffic on.</li> <li>• <b>Protocol</b>—Click to choose <b>HTTP</b> or <b>HTTPS</b>.</li> <li>• <b>Provider EPG</b>—The EPG with the web server that handles the traffic.</li> <li>• <b>EPG</b>—To choose an EPG: <ol style="list-style-type: none"> <li>a. Click <b>Select EPG</b>. The <b>Select EPG</b> dialog box appears.</li> <li>b. From the <b>Select EPG</b> dialog ox, click to choose an EPG in the left column then click <b>Select</b>. The <b>Select EPG</b> dialog box closes.</li> </ol> </li> </ul> </li> <li>• <b>Redirect</b>—Redirects requests to another location using the following options: <ul style="list-style-type: none"> <li>• <b>Redirect Code</b>—Click the <b>Redirect Code</b> drop-down list and choose a code.</li> </ul> </li> </ul> </li> </ul>

Properties	Description
	<ul style="list-style-type: none"> <li>• <b>Redirect Hostname</b>—Enter a hostname for the redirect.</li> <li>• <b>Redirect Path</b>—Enter a redirect path.</li> <li>• <b>Redirect Port</b>—Enter the port that the device will accept traffic on.</li> <li>• <b>Redirect Protocol</b>—Click to the <b>Redirect Protocol</b> drop-down list and choose <b>HTTP</b>, <b>HTTPS</b>, or <b>Inherit</b>.</li> <li>• <b>Redirect Query</b>—Enter a redirect query.</li> </ul> <p>Click <b>Add Rule</b> when finished.</p>

**Step 10** Click **Add** when finished.  
The service graph is deployed.

## Deploying a Service Graph Using the REST API

### Creating an Internal-Facing Load Balancer Using the REST API

This example demonstrates how to create an internal-facing load balancer using the REST API.

To create an internal-facing load balancer:

**Example:**

```
<polUni>
  <fvTenant name="t2" status="">
    <cloudLB name="ALB1" type="application" scheme="internal" status="">
      <cloudRsLDevToCloudSubnet tDn="uni/tn-t2/ctxprofile-cl/cidr-[10.33.0.0/16]/subnet-[10.33.7.0/24]"
status=""/>
      <cloudRsLDevToCloudSubnet tDn="uni/tn-t2/ctxprofile-cl/cidr-[10.33.0.0/16]/subnet-[10.33.8.0/24]"
status=""/>
    </cloudLB>
  </fvTenant>
</polUni>
```

### Configuring an Internet-Facing Load Balancer Using the REST API

This example demonstrates how to create an internet-facing load balancer using the REST API.

To create an internet-facing load balancer:

**Example:**

```
<polUni>
  <fvTenant name="t2" status="">
    <cloudLB name="ALB1" type="application" scheme="internet" status="">
      <cloudRsLDevToCloudSubnet tDn="uni/tn-t2/ctxprofile-c1/cidr-[10.33.0.0/16]/subnet-[10.33.5.0/24]"
      status=""/>
      <cloudRsLDevToCloudSubnet tDn="uni/tn-t2/ctxprofile-c1/cidr-[10.33.0.0/16]/subnet-[10.33.6.0/24]"
      status=""/>
    </cloudLB>
  </fvTenant>
</polUni>
```

**Creating a Service Graph Using the REST API**

This example demonstrates how to create a service graph using the REST API.

To create a service graph:

```
<polUni>
  <fvTenant name="t2">
    <vnsAbsGraph name="CloudGraph" type="cloud" status="">
      <vnsAbsTermNodeProv name="Input1">
        <vnsAbsTermConn name="C1"/>
      </vnsAbsTermNodeProv>
      <vnsAbsTermNodeCon name="Output1">
        <vnsAbsTermConn name="C2"/>
      </vnsAbsTermNodeCon>
      <vnsAbsNode funcType="GoTo" name="N1" managed="yes">
        <vnsRsNodeToCloudLDev tDn="uni/tn-t2/clb-ALB1" status=""/>
        <vnsAbsFuncConn name="provider"/>
        <vnsAbsFuncConn name="consumer"/>
      </vnsAbsNode>
      <vnsAbsConnection connDir="consumer" connType="external" name="CON2">
        <vnsRsAbsConnectionConns tDn="uni/tn-t2/AbsGraph-CloudGraph/AbsTermNodeCon-Output1/AbsTConn"/>
        <vnsRsAbsConnectionConns tDn="uni/tn-t2/AbsGraph-CloudGraph/AbsNode-N1/AbsFConn-consumer"/>
      </vnsAbsConnection>
      <vnsAbsConnection connDir="provider" connType="internal" name="CON1">
        <vnsRsAbsConnectionConns tDn="uni/tn-t2/AbsGraph-CloudGraph/AbsTermNodeProv-Input1/AbsTConn"/>
        <vnsRsAbsConnectionConns tDn="uni/tn-t2/AbsGraph-CloudGraph/AbsNode-N1/AbsFConn-provider"/>
      </vnsAbsConnection>
    </vnsAbsGraph>
  </fvTenant>
</polUni>
```

**Attaching a Service Graph Using the REST API**

This example demonstrates how to attach a service graph using the REST API.

To attach a service graph:

```

<polUni>
  <fvTenant name="t2">
    <vzBrCP name="httpFamily">
      <vzSubj name="default" revFltPorts="yes" targetDscp="unspecified">
        <vzRsSubjGraphAtt tnVnsAbsGraphName="CloudGraph"/>
      </vzSubj>
    </vzBrCP>
  </fvTenant>
</polUni>

```

## Configuring an HTTP Service Policy Using the REST API

This example demonstrates how to create an HTTP service policy using the REST API.

To create an HTTP service policy:

```

<polUni>
  <fvTenant name="t2">
    <vnsAbsGraph name="CloudGraph" type="cloud" status="">
      <vnsAbsNode funcType="GoTo" name="N1" managed="yes">
        <cloudSvcPolicy tenantName="t2" contractName="httpFamily" subjectName="consubj">
          <cloudListener name="http_listener1" port="80" protocol="http" status="">
            <cloudListenerRule name="rule1" priority="10" default="yes" status="">
              <cloudRuleAction type="forward" port="80" protocol="http"
epgdn="uni/tn-t2/cloudapp-ap/cloudepg-provEPG"/>
            </cloudListenerRule>
            <cloudListenerRule name="redirectRule" priority="20">
              <cloudRuleCondition type="path" value="/img/*"/>
              <cloudRuleAction type="redirect" RedirectPort="8080"/>
            </cloudListenerRule>
            <cloudListenerRule name="FixedRspRule" priority="30">
              <cloudRuleCondition type="host" value="example.com"/>
              <cloudRuleAction type="fixedResponse" FixedResponseCode="200"/>
            </cloudListenerRule>
            <cloudListenerRule name="redirectHPRule" priority="40" status="">
              <cloudRuleCondition type="host" value="example.com"/>
              <cloudRuleCondition type="path" value="/img/*"/>
              <cloudRuleAction type="forward" port="80" protocol="http"
epgdn="uni/tn-t2/cloudapp-ap/cloudepg-provEPG"/>
            </cloudListenerRule>
          </cloudListener>
        </cloudSvcPolicy>
      </vnsAbsNode>
    </vnsAbsGraph>
  </fvTenant>
</polUni>

```

## Configuring a Key Ring Using the REST API

This example demonstrates how to configure a key ring using the REST API. For more information about key ring configuration, see the *Cisco APIC Basic Configuration Guide*.

To configure a key ring:

```
<polUni>
  <fvTenant name="t2">
    <cloudCertStore>
      <pkiKeyRing status="" name="lbCert" tp="lbTP" key="-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAACAQEA4DGxaK+RHv/nToHLnmDBq2BfLimgX/zNJQC9bGuzr8Mj7dm0
XuHfQYGv0h1PtL4Pdx5f5qjB0NbHjAVB1Gw8cDiErEgAXy9Km27ySo2foKryNqCRE
Ginn/CgF75QPied568eScNDZPt/eMeHAuRX/PykKUatWWncGanJvHqc+SOLPF6TD
gQ5nwOHFFvYm2DY8bfdYWrWmGsO7JqZzbPMptA2QWblLsSoIrdkI Igf6ZfYy/EN
bH+nYN2rJT8lzYsxxz0YmR0oRQHTiN2NiDY/ZV63yxCXfLg9qpNZCuD8KOfdCZPEq
8takiWBxiR5/HRPscWAdWQsoiKgG1k4NEbFA9QIDAQABAoIBAQQDQqA9Is1YrdtqN
q6mZ3s2BNfF/4kgb7gn0Dws+9EJJLCJNZVhFEo2ZxxyfPp6HRnjYS50W83/ElanD
+GD1bSucTuxqFWIQVh7r1ebYZIWK+NYSjr5yNVxux8U2hCNNV8WWVqkJjKcUqICB
Bm47FKj53LV46zE0qyCaibFrYxZJ9+farGneyBdnoV+3thmez7534KCioT3J3Eri
lgSY3ql6hPXB2ZXAP4jdAoLgWDU4I1M6OqOiWopZM/QYIE/WtPYyJ0QzNCXObtc5
FboDcvedsgd4x5Glfv2A4xTBQMCTZUZJ9fyAcFogTZXD+UVqxorh47tf/mz+1fjq
fLXphEDL1AoGBAPVlvkfgW46qqRnYovfryxxx4OMlsvSgcJpQTQtBQi2koJ8OwEzJ
2s+CX0r+oDqwP23go/QEVYVkcic9RGkJBNGel+dm/bTjzgmQYtqSCNctetsZD5JN
yljkciiizznDkjcyjReS22kh3dGXlBriYk7ezp2z7EKfDrHe5x5ouGmgCnAoGBAOnh
buDEohv8KJaB+DiUfhtoa3aKNPBO+zWPCHp0HFGjPXshJcIYZc1GcycmuDKVnNdD
MxhE/yOnQHowi4T9FMLpz5yh5zuCUVqOBgB1P6Mzbc5t5MtLrEYr/AqFN11CqyXQ
cVcT6iCW1OAFJRW3c/OiESwLMzchsl8RnbwOi6kDaOGBANV1zmPb07zB3eGTUOt
KGiqwFLncUkVaDZRFZYPPnwiRkoe73j9brkNbgCqxW+Nlp5UjoeFry0N6y106q/
ZA4I7FnXryLBw2HYuw41vixl+XOZ/HeO3RmFN1z717dGmaGbv43aKIB9x+X5n8wF
6z1NtBHmBk7yNwomlIRaglSbAoGAX0p4cJ/tJNXSe7AswHDQCL68uimJdDfZ5nKG
k83nE+Qc0qQozDJAmCiSfmuSNRnSep3FiafjBFXK0X4h+mdbJcC7bagRnI92Mh0X
mOwsp4P2GdywZwdbuHQ6UBp1Ferf9aztzt+nas6xKOUATEezy9DK9zMWzQhhtAY
m9yZTp0CgYEALUtcpWjAzQbXODJGmxGdAAakPpeiKw/Da3MccrTdGJt88ezM1OeJ
Pdoab0G2PcGfJZoTSGk7N4KArVKeq7pgZ0kwcYAsh06A2Hal+D1z/bGoZP+kmD/x
Ny82phxYOXCncE5Vv921U59+j7e067UFLAYJe6fu+oFImvofRnP4DIQ= -----END RSA PRIVATE KEY-----" cert="-----BEGIN
CERTIFICATE----- MIIElTCCA32gAwIBAgIJAKWNjp//arBsMA0GCSqGSIb3DQEBCwUAMIGNMQswCQYD
VQQGEwJVUzELMAkGA1UECBMCQ0ExETAPBgNVBACTCFhbiBk3N1MRlweAYDVQQK
Ew1NeUNvbXBhbHbnkxDjAMBGNVBAStBU15T3JnMRgwFgYDVQDDFA8qLmFtYXpvcjF3
cy5jb20xIDAeBgkqhkiG9w0BCQEWEWEXJhbXNoYWhAY21zY28uY29tMB4XDTE4MTAw
MjIwNTMwNV0xMTE4MTAwMjIwNTMwNV0wY0xzc2EwIBAgIwIBAgIwIBAgIwIBAgIwIBAgI
EwJDQTERMA8GA1UEBxMIU2FuIEpvc2UxZjAqBGNVBAoTCU15Q29tcGFueTEOMAwG
A1UECXMFTX1PcmcxGDAWBgNVBAMUDyouYW1hem9uYXZzLmNvbTEgMB4GCCqGSIb3
DQEJARYRcmFtc2hhaEBjaXNjby5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQQDgMbFor5Ee/+dOgcueYMGryF8uKaBf/M01AL1sa7OvwyPt2bRe4d9E
ga/SHU+0vg93F/mqMHQ1seMBUHUbDxwOISsSABfL0qbbvJKjZ+gqvI2oJF4aKef8
KAXv1A8h53nrx5Jw0Nk+394x4c5Ff8/KQpRq1ZadwZqe08epz5I4s8XpMOBDMfA
4ccW/IzYNjxt91hataYaw7smpnNs8ym0DZBZuUguxKgIt2QgiB/pl9jL8Q1sf6dg
3as1PyXNizHPRiZHSHFADoi3Y2INj9lXrflEJd8ud2qk1kK4Pwo590Jk8Sry1qSj
YHGJHn8de+xxYB1ZCyilQAbWTg0RsUD1AgMBAAGjgUwgfIwHQYDVR0OBBYEFBYq
K3b39+1oOr4IBSsePwOpML7MIHCBGnVHSMEgbowgbeAFBYqK3b39+1oOr4IBSse
PwOpML7oYGTpIGQMIGNMQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExETAPBgNV
BACTCFhbiBk3N1MRlweAYDVQQKEw1NeUNvbXBhbHbnkxDjAMBGNVBAStBU15T3Jn
MRgwFgYDVQDDFA8qLmFtYXpvcjF3cy5jb20xIDAeBgkqhkiG9w0BCQEWEWEXJhbXNo
YWhAY21zY28uY29tggkAPY20n/9qsGwwDAYDVR0TBAAUwAwEB/zANBgkqhkiG9w0B
AQsFAAOCAQEAE/RuzCheLiBhbrurGet6eaVx9DPYydNiKVBSAKO+5iuR84mqzhoT
nx5CN109xu5ml5baCYZsSnn6D7usC092bPA/kRCGxt29gkjpWA74tJHqIhVWgbM
mOrLiSHoelew+wR1oOvrCh1tFkT068Tuk6vrqpw76hKfOHIa7b2h1IIMdq6VA/
+A5FQ0xqYfKdVd2RaINpzI8mqZisZqw+7E6j1PL5k4tftWEaYpfGp1VesFEyJEL
gHBUIPt8TibaMYI8gUqMb/emnLXeKQ5PRxdRnleA3h8jfq3D1CQRTLjmdL3tpFwg qopM6et5ZKqShX4T87BsgZIoiquzXqsuHg==
-----END CERTIFICATE-----">
    </pkiKeyRing>
  </fvTenant>
</polUni>
<pkiTP status="" name="lbTP" certChain="-----BEGIN CERTIFICATE-----
MIIElTCCA32gAwIBAgIJAKWNjp//arBsMA0GCSqGSIb3DQEBCwUAMIGNMQswCQYD
VQQGEwJVUzELMAkGA1UECBMCQ0ExETAPBgNVBACTCFhbiBk3N1MRlweAYDVQQK
Ew1NeUNvbXBhbHbnkxDjAMBGNVBAStBU15T3JnMRgwFgYDVQDDFA8qLmFtYXpvcjF3
cy5jb20xIDAeBgkqhkiG9w0BCQEWEWEXJhbXNoYWhAY21zY28uY29tMB4XDTE4MTAw
MjIwNTMwNV0xMTE4MTAwMjIwNTMwNV0wY0xzc2EwIBAgIwIBAgIwIBAgIwIBAgIwIBAgI
EwJDQTERMA8GA1UEBxMIU2FuIEpvc2UxZjAqBGNVBAoTCU15Q29tcGFueTEOMAwG
A1UECXMFTX1PcmcxGDAWBgNVBAMUDyouYW1hem9uYXZzLmNvbTEgMB4GCCqGSIb3
DQEJARYRcmFtc2hhaEBjaXNjby5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQQDgMbFor5Ee/+dOgcueYMGryF8uKaBf/M01AL1sa7OvwyPt2bRe4d9E
ga/SHU+0vg93F/mqMHQ1seMBUHUbDxwOISsSABfL0qbbvJKjZ+gqvI2oJF4aKef8
KAXv1A8h53nrx5Jw0Nk+394x4c5Ff8/KQpRq1ZadwZqe08epz5I4s8XpMOBDMfA
4ccW/IzYNjxt91hataYaw7smpnNs8ym0DZBZuUguxKgIt2QgiB/pl9jL8Q1sf6dg
3as1PyXNizHPRiZHSHFADoi3Y2INj9lXrflEJd8ud2qk1kK4Pwo590Jk8Sry1qSj
YHGJHn8de+xxYB1ZCyilQAbWTg0RsUD1AgMBAAGjgUwgfIwHQYDVR0OBBYEFBYq
K3b39+1oOr4IBSsePwOpML7MIHCBGnVHSMEgbowgbeAFBYqK3b39+1oOr4IBSse
PwOpML7oYGTpIGQMIGNMQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExETAPBgNV
BACTCFhbiBk3N1MRlweAYDVQQKEw1NeUNvbXBhbHbnkxDjAMBGNVBAStBU15T3Jn
MRgwFgYDVQDDFA8qLmFtYXpvcjF3cy5jb20xIDAeBgkqhkiG9w0BCQEWEWEXJhbXNo
YWhAY21zY28uY29tggkAPY20n/9qsGwwDAYDVR0TBAAUwAwEB/zANBgkqhkiG9w0B
AQsFAAOCAQEAE/RuzCheLiBhbrurGet6eaVx9DPYydNiKVBSAKO+5iuR84mqzhoT
nx5CN109xu5ml5baCYZsSnn6D7usC092bPA/kRCGxt29gkjpWA74tJHqIhVWgbM
mOrLiSHoelew+wR1oOvrCh1tFkT068Tuk6vrqpw76hKfOHIa7b2h1IIMdq6VA/
+A5FQ0xqYfKdVd2RaINpzI8mqZisZqw+7E6j1PL5k4tftWEaYpfGp1VesFEyJEL
gHBUIPt8TibaMYI8gUqMb/emnLXeKQ5PRxdRnleA3h8jfq3D1CQRTLjmdL3tpFwg qopM6et5ZKqShX4T87BsgZIoiquzXqsuHg==
-----END CERTIFICATE-----"
    </pkiKeyRing>
  </pkiTP>
</polUni>
```



```

MjIwNTMwNVoXDTExMTAwMjIwNTMwNVowgY0xCzAJBgNVBAYTA1VTMQswCQYDVQQLI
EwJDRTERMA8GA1UEBxMIU2FuIEpvc2UxEjAQBgNVBAoTCU15Q29tcGFueTEOMAwG
A1UECmFTX1PcmcxGDAWBGNVBAUDyouYw1hem9uYXdzLmNvbTEgMB4GCSqGSIb3
DQEJARYRcmFtc2hhaEBjaXNjby5jb20wgEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQQdMgFor5Ee/+dOgcueYMGryF8uKaBf/M0lAL1sa7OvwyPt2bRe4d9B
ga/SHU+0vg93F/mqMHQ1seMBUHUBDxwOISsSABfL0qbbvJKjZ+qqvI2oJF4aKef8
KAXv1A8h53nrx5Jw0Nk+394x4cC5Ff8/KQpRq1ZadwZqeO8epz5I4s8XpMOBDMfA
4ccW/IzYNjxt9lhataYaw7smpnNs8ym0DZBZuUguxKgit2QgiB/pl9jL8Q1sf6dg
3aslPyXNizHPRIzHShFadOI3Y2INj9lXrfLEJd8uD2qk1kk4Pwo590Jk8Sry1qSJ
YHGJHn8de+xxYBlZCyIqAbWTg0RsUD1AgMBAAGjgfUwgfIwHQYDVR0OBBYEFBYq
K3b39+1oOr4IBSsePwcOpML7MIHCBgNVHSMGbowgbeAFBYqK3b39+1oOr4IBSse
PwcOpML7oYGTpIGQMIGNMQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0EExETAPBgNV
BACTFNhbIBKb3NlMRIwEAYDVQQKEw1NeUNvbXBhbnkxZjAMBGNVBAStBU15T3Jn
MRgwFgYDVQDDFA8qLmFtYXpvc2UxEjAQBgNVBAoTCU15Q29tcGFueTEOMAwG
YWhAY2lzy28uY29tggkApY2On/9qsGwwDAYDVR0TBAAUwAwEB/zANBgkqhkiG9w0B
AQsFAAOCAQEAE/RuzCheLIbHbrurGet6eaVx9DPYydNiKVBSAKO+5iuR84mQzhoT
nx5CN109xu5ml5baCYZsSnn6D7usC092bPA/kRCGxt29gkjpWA74tJHqIhVWgbM
mOrLiShoelewv+wRl0oVRChlTfKtXO68TUK6vrqpW76hKfOHIA7b2h1IIMdq6VA/
+A5FQ0xqYfGkDvd2RaINpzI8mqZiszqw+7E6j1PL5k4tftWEaYpfGPLVesFEyJEL
gHBUIPt8TIbaMYI8qUQmB/emnLXeKQ5PRxdRnleA3h8jfq3D1CQRTLjmdL3tpFwg
qopM6et5ZKqShX4T87BsgZIoiquzXqsuHg==
-----END CERTIFICATE-----"
  </pkITP>
  </cloudCertStore>
</fvTenant>
</polUni>

```

## Creating an HTTPS Service Policy Using the REST API

This section demonstrates how to create an HTTPS service policy using the REST API.



**Note** A listener can have multiple certificates. The certificate options are:

- ELBSecurityPolicy-2016-08 – The default when no security policy is chosen.
- ELBSecurityPolicy-FS-2018-06
- ELBSecurityPolicy-TLS-1-2-2017-01
- ELBSecurityPolicy-TLS-1-2-Ext-2018-06
- ELBSecurityPolicy-TLS-1-1-2017-01
- ELBSecurityPolicy-2015-05
- ELBSecurityPolicy-TLS-1-0-2015-04

If you use multiple certificates, you must specify the default certificate. The default is specified using the **defaultCert** property in **cloudRsListenerToCert**.

### Before you begin

You have already configured a key ring certificate.

To create an HTTPS service policy:

```

<polUni>
  <fvTenant name="t2">
    <vnsAbsGraph name="CloudGraph" type="cloud" status="">
      <vnsAbsNode funcType="GoTo" name="N1" managed="yes">
        <cloudSvcPolicy tenantName="t2" contractName="httpFamily" subjectName="consubj">
          <cloudListener name="https_listener" port="443" protocol="https"
secPolicy="eLBSecurityPolicy-2016-08" status="">
            <cloudRsListenerToCert defaultCert="yes" certStore="iam"
tDn="uni/tn-t2/certstore/keyring-lbCert" status=""/>
              <cloudListenerRule name="defaultRule" default="yes" priority="100" status="">
                <cloudRuleAction type="forward" port="80" protocol="http"
epgdn="uni/tn-t1/cloudapp-ap/cloudepg-ep1">
                  </cloudRuleAction>
                </cloudListenerRule>
              </cloudListenerRule>
            </cloudListener>
          </cloudSvcPolicy>
        </vnsAbsNode>
      </vnsAbsGraph>
    </fvTenant>
  </polUni>

```

---