Configuring Route Peering

About Route Peering

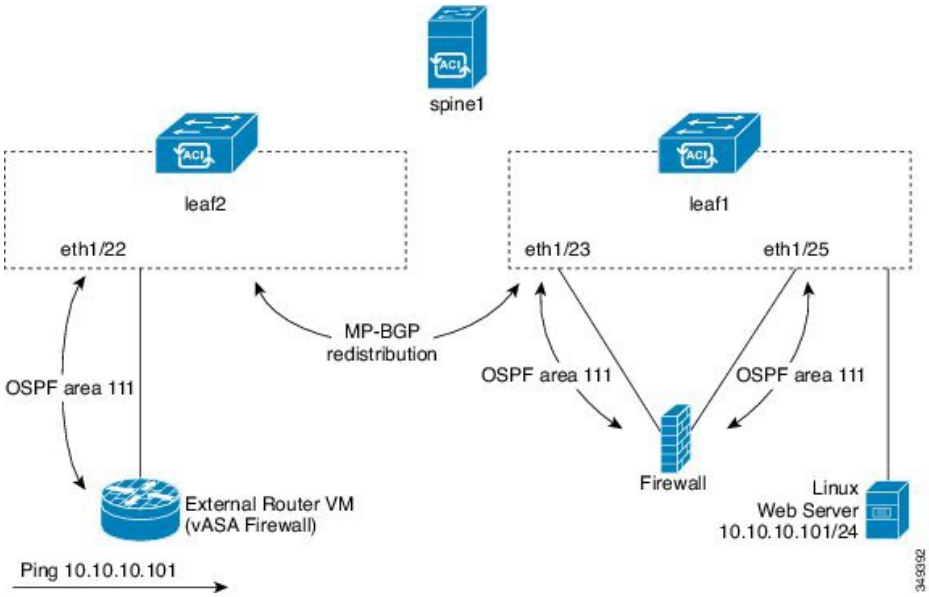
Route peering is a special case of the more generic Cisco Application Centric Infrastructure (ACI) fabric as a transit use case, in which route peering enables the ACI fabric to serve as a transit domain for Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP) protocols. A common use case for route peering is route health injection, in which the server load balancing virtual IP is advertised over OSPF or internal BGP (iBGP) to clients that are outside of the ACI fabric. You can use route peering to configure OSPF or BGP peering on a service device so that the device can peer and exchange routes with the ACI leaf switch to which it is connected.

The following protocols are supported for route peering:

- OSPF
- OSPFv3
- iBGPv4
- iBGPv6
- Static routes

The following figure shows how route peering is commonly deployed:

Figure 1: Common Route Peering Topology



As shown in the figure, a Web server's public IP address is advertised to an external router through a firewall by deploying a service graph with route peering configured. You must deploy OSPF routing policies on each leg of the firewall. This is typically done by deploying `l3extOut` policies. This enables the Web server reachability information to be advertised over OSPF through the firewall to the border leaf switch and to the external router.

Route distribution between leaf switches in the fabric is internally accomplished over Multi-Protocol Border Gateway Protocol (MP-BGP).

For a more detailed example of the route peering topology, see Route Peering End-to-End Flow, on page 10.

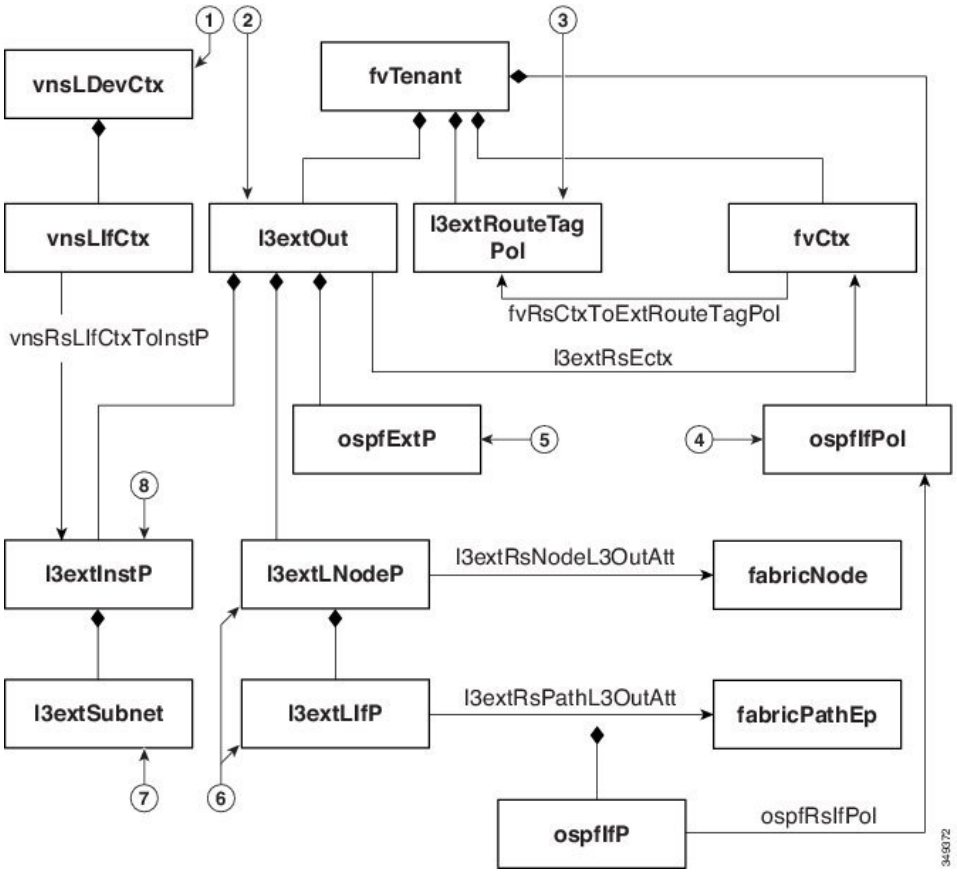
For more information about configuring `l3extOut` policies, see the *Cisco Application Centric Infrastructure Fundamentals Guide*.

Note Point-to-point non-broadcast mode is not supported on an Adaptive Security Appliance (ASA). You must remove the point-to-point non-broadcast mode configuration from the Application Policy Infrastructure Controller (APIC) if the configuration exists.

Open Shortest Path First Policies

To configure route peering, you must first create one or more `l3extOut` policies and deploy them on the fabric leaf nodes where the service device is connected. These `l3extOut` policies specify the Open Shortest Path First (OSPF) parameters that you must enable on the fabric leaf. The policies are very similar to the `l3extOut` policies that are used for external communication. The following figure illustrates the route peering object relations.

Figure 2: OSPF Route Peering Object Relations



1. `vnsLDevCtx`—Device selection policy.
2. `l3extOut`—Contains all OSPF policies for a single area.
3. `l3extRouteTagPol`—Every context used by route peering needs a unique route tag to avoid OSPF loops. The OSPF routes that are learned from one leg will not be learned on the other leg unless the route tags are different.
4. `ospfIfPol`—OSPF per interface policy.
5. `ospfExtP`—OSPF per area policy.
6. `l3extLNodeP/l3extLIfP`—The nodes or ports on which this `l3extOut` is deployed.
7. `l3extSubnet`—Subnets to export from or import into the fabric.
8. `l3extInstP`—Prefix-based EPG.

Two example `l3extOut` policies, `OspfExternal` and `OspfInternal`, are shown below. These policies are deployed on the external and internal legs of the firewall device in Figure 1: Common Route Peering Topology, on page 2. The `l3extOut` policy specifies one or more prefix-based EPGs (`l3extInstP`), which control how traffic is classified by the fabric leaf and also how routes are imported from and exported to the service device. The `l3extOut` policy contains the OSPF per-area policy (`ospfExtP`) and one or more OSPF interface policies (`ospfIfPol`) that are specified under it.

The following example shows an OSPF area with area-Id being configured with a value of "100":

```
<ospfExtP areaId="100" areaType="regular" areaCtrl="redistribute"/>
```

The area type is set to "regular" and the area control attribute is set to "redistribute".

The OSPF interface policy specifies one or more OSPF interface timers:

```
<ospfIfPol name="ospfIfPol" ctrl="mtu-ignore" nwT="bcast" xmitDelay="1" helloIntvl="10"
  deadIntvl="40" status="created,modified"/>
```

If default timers are fine, then you do not need to specify this policy. This policy allows certain timers to be modified from default values and is associated with one or more interfaces by using the following relation:

```
<l3extRsPathL3OutAtt tDn="topology/pod-l/paths-101/pathep-[eth1/25]" ifInstT="ext-svi"
  encap="vlan-3844" addr="30.30.30.100/28" mtu="1500"/>
```

The attributes of the `l3extRsPathL3OutAtt` relation are as follows:

- `ifInstT`—The logical interface type, which is typically "ext-svi".
- `encap`—You must specify a VLAN encapsulation when creating this interface. The encapsulation is pushed to the service device.
- `addr`— The IP address of the SVI interface that was created on the fabric leaf where this `l3extOut` is deployed.

The following policy controls where the `l3extOut` policy is deployed:

```
<l3extNodeP name="bLeaf-101">
  <l3extRsNodeL30utAtt tDn="topology/pod-l/node-101" rtrId="180.0.0.11"/>
  <l3extLIfP name="portlf">
    <l3extRsPathL30utAtt tDn="topology/pod-1/paths-101/pathep-teth1/251"
      ifInstT="ext-svi" encap="vlan-3844" addr="30.30.30.100/28" mtu="1500"/>
    <ospfIfP authKey="tecom" authType="md5" authKeyId='1'>
      <ospfRslfPol tnOspfIfPolName="ospfIfPol"/>
    </ospflfP>
  </l3extLIfP>
</l3extLNodeP>
```

The `l3extOut` policy must be deployed to the same leaf ports to which the service device is connected.

The `scope=import-security` attribute does the following things:

- Controls the flow of traffic in the data plane
- Acts as a directive to the external device to advertise this route

Note For route peering to work correctly, the `l3extRsPathL3OutAtt` relation must point to the same fabric destination as the `RsCIfPathAtt` relation under the `vnsCDev` that represents the device.

`OspfExternal` Policy

`OspfInternal` Policy

Virtual Services

```
<polUni>
  <fvTenant name="common">
    <fvCtx name="commonctx">
      <fvRsCtxToExtRouteTagPol tnL3extRouteTagPolName="myTagPol"/>
    </fvCtx>
    <l3extRouteTagPol tag="212" name="myTagPol"/>
    <l3extOut name="OspfExternal" status="created,modified">
      <l3extLNodeP name="bLeaf-101">
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.8/28"/>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
            ifInstT="ext-svi" encap="vlan-3843" addr="40.40.40.100/28" mtu="1500"/>
          <ospfIfP authKey="tecom" authType="md5" authKeyId='1'>
            <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
          </ospfIfP>
        </l3extLIfP>
      </l3extLNodeP>
      <ospfExtP areaId="100" areaType="regular" areaCtrl="redistribute"/>
      <l3extInstP name="ExtInstP">
        <l3extSubnet ip="40.40.40.100/28" scope="import-security"/>
        <l3extSubnet ip="10.10.10.0/24" scope="import-security"/>
      </l3extInstP>
      <l3extRsEctx tnFvCtxName="commonctx"/>
    </l3extOut>
   <ospfIfPol name="ospfIfPol" ctrl="mtu-ignore" nwT="bcast" xmitDelay="1" helloIntvl="10"

      deadIntvl="40" status="created,modified"/>
  </fvTenant>
</polUni>
```

```
<polUni>
  <fvTenant name="tenant1">
    <l3extRouteTagPol tag="213" name="myTagPol"/>
    <fvCtx name="tenant1ctx1">
      <fvRsCtxToExtRouteTagPol tnL3extRouteTagPolName="myTagPol"/>
    </fvCtx>
    <l3extOut name="OspfInternal" status="created,modified">
      <l3extLNodeP name="bLeaf-101">
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.11"/>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/25]"
            ifInstT="ext-svi" encap="vlan-3844" addr="30.30.30.100/28" mtu="1500"/>
          <ospfIfP authKey="tecom" authType="md5" authKeyId='1'>
            <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
          </ospfIfP>
        </l3extLIfP>
      </l3extLNodeP>
      <ospfExtP areaId="100" areaType="regular" areaCtrl="redistribute"/>
      <l3extInstP name="IntInstP">
        <l3extSubnet ip="30.30.30.100/28" scope="import-security"/>
        <l3extSubnet ip="20.20.20.0/24" scope="import-security"/>
      </l3extInstP>
      <l3extRsEctx tnFvCtxName="tenant1ctx1"/>
    </l3extOut>
   <ospfIfPol name="ospfIfPol" ctrl="mtu-ignore" nwT="bcast" xmitDelay="1" helloIntvl="10"
```

```
      deadIntvl="40" status="created,modified"/>
  </fvTenant>
</polUni>
```

The `OspfExternalInstP` policy specifies that prefixes 40.40.40.100/28 and 10.10.10.0/24 must be used for prefix-based endpoint association. The policy also instructs the fabric to export prefix 20.20.20.0/24 to the service device.

```
<l3extInstP name="OspfExternalInstP">
  <l3extSubnet ip="40.40.40.100/28" scope="import-security"/>
  <l3extSubnet ip="10.10.10.0/24" scope="import-security"/>
  <l3extSubnet ip="20.20.20.0/24" scope="export"/>
</l3extInstP>
```

The `bleaf-101` policy controls where this `l3extOut` policy is deployed.

```
<l3extLNodeP name="bLeaf-101">
  <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.8/28"/>
  <l3extLIfP name="portIf">
    <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
      ifInstT="ext-svi" encap="vlan-3843" addr="40.40.40.100/28" mtu="1500"/>
    <!-- <ospfIfP authKey="tecom" authType="md5" authKeyId='1'> -->
    <ospfIfP>
      <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
    </ospfIfP>
  </l3extLIfP>
</l3extLNodeP>
```

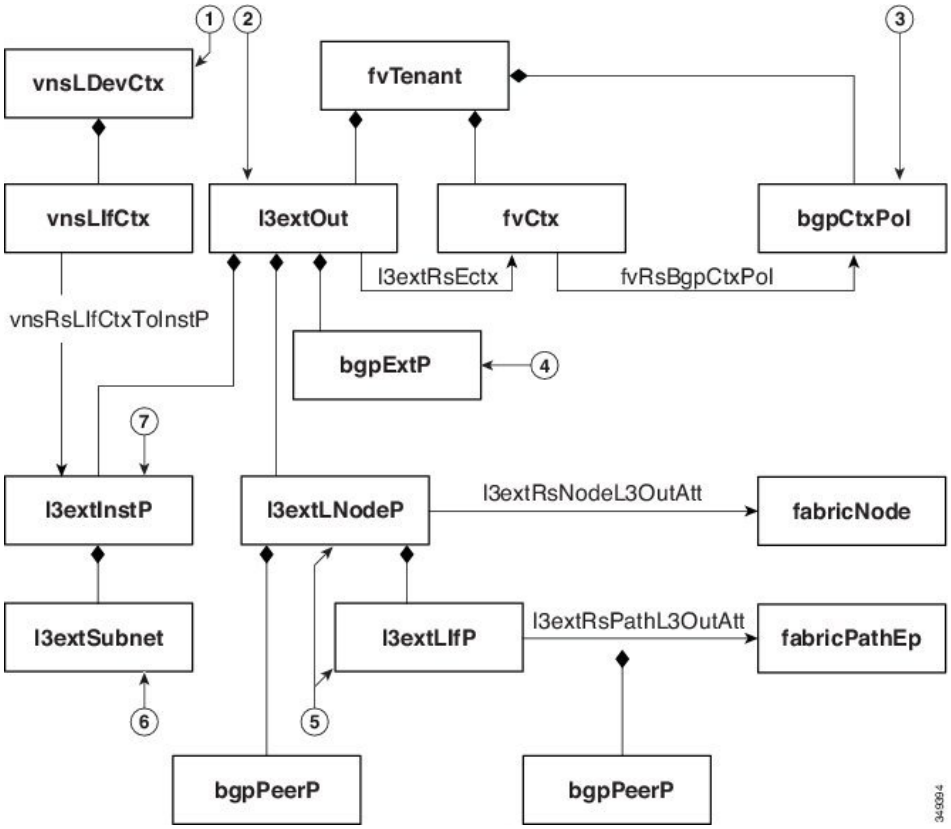
You can deploy virtual services with route peering, although the `l3extRsPathL3OutAtt` validation with the `vnsCIf` object is not performed. The datapath will work only if the `l3extOut` object is deployed to the correct leaf to which the virtual service device is connected.

Border Gateway Protocol Policies

You can configure route peering using internal Border Gateway Protocol (iBGP) on the device's external interface and static routes on the internal interface. You cannot configure iBGP on both the internal and external interfaces of the device without extra configuration, as the interfaces must be in different autonomous systems and inter-autonomous system redistribute policies do not get pushed down.

The following figure illustrates the route peering object relations:

Figure 3: iBGP Route Peering Object Relations



1. `vnsLDevCtx`—Device selection policy.
2. `l3extOut`—Contains all BGP policies for a single autonomous system.
3. `bgpCtxPol`—Per-context BGP timers.
4. `bgpExtP`—BGP per ASN policy.
5. `l3extLIfP/l3extLNodeP`—Controls to which nodes or ports these endpoint groups (EPGs) are deployed.
6. `l3extSubnet`—Subnets to export from and import into the fabric.
7. `l3extInstP`—Prefix-based EPG.

The following policy configures iBGPv4/v6 on the external interface:

```
<polUni>
  <fvTenant name="common">
    <fvCtx name="commonctx">
      <fvRsBgpCtxPol tnBgpCtxPolName="timer-3-9"/>
      <fvRsCtxToExtRouteTagPol tnL3extRouteTagPolName="myTagPol"/>
    </fvCtx>
    <l3extRouteTagPol tag="212" name="myTagPol"/>
   <bgpCtxPol grCtrl="helper" holdIntvl="9" kaIntvl="3" name="timer-3-9" staleIntvl="30"/>

    <l3extOut name="BgpExternal" status="created,modified">
      <l3extLNodeP name="bLeaf-101">
        <!-- <bgpPeerP addr="40.40.40.102/32 "ctrl="send-com"/> -->
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.8/28>"
```

```
          <l3extLoopBackIfP addr="50.50.50.100/32"/>
        </l3extRsNodeL3OutAtt>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
            ifInstT="ext-svi" encap="vlan-3843" addr="40.40.40.100/28 "mtu="1500">
            <bgpPeerP addr="40.40.40.102/32 "ctrl="send-com"/>
          </l3extRsPathL3OutAtt>
        </l3extLIfP>
      </l3extLNodeP>
      <bgpExtP/>
      <l3extInstP name="ExtInstP">
        <l3extSubnet ip="40.40.40.100/28 "scope="import-security"/>
        <l3extSubnet ip="10.10.10.0/24 "scope="import-security"/>
        <l3extSubnet ip="20.20.20.0/24 "scope="export-rtctrl"/>
      </l3extInstP>
      <l3extRsEctx tnFvCtxName="commonctx"/>
    </l3extOut>
  </fvTenant>
</polUni>
```

iBGP peers can be configured at the physical interface level or the loopback level. The following example shows a iBGP peer configured at the physical interface level:

```
<l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
            ifInstT="ext-svi" encap="vlan-3843" addr="40.40.40.100/28 "mtu="1500">
            <bgpPeerP addr="40.40.40.102/32 "ctrl="send-com"/>
          </l3extRsPathL3OutAtt>
</l3extLIfP>
```

In this case, the iBGP process that is running on the fabric uses the switch virtual interface (SVI) IP address 40.40.40.100/28 to peer with its neighbor. The neighbor is the service device at IP address 40.40.40.102/32.

In the following example, the iBGP peer definition has been moved to the logical node level (under `l3extLNodeP`) and a loopback interface has been configured:

```
      <l3extLNodeP name="bLeaf-101">
        <bgpPeerP addr="40.40.40.102/32 "ctrl="send-com"/>
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.8/28>"
          <l3extLoopBackIfP addr="50.50.50.100/32"/>
        </l3extRsNodeL3OutAtt>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
            ifInstT="ext-svi" encap="vlan-3843" addr="40.40.40.100/28 "mtu="1500">
          </l3extRsPathL3OutAtt>
        </l3extLIfP>
      </l3extLNodeP>
```

In this case, the iBGP process uses the loopback address to peer with its neighbor. If no loopback is configured, then the fabric uses the IP address that is specified by rtrId to peer with the neighbor.

In such cases, the device needs a route to reach the SVI. This is typically configured using graph parameters, as shown by the following example for ASA, where IP address 50.50.50.0 is reachable from IP address 40.40.40.100:

```
<vnsAbsFolder name="ExtRouteCfg" key="StaticRoute">
              <vnsAbsFolder name="route1" key="route">
                <vnsAbsParam name="network" key="network" value="50.50.50.0"/>
                <vnsAbsParam name="netmask" key="netmask" value="255.255.255.0"/>
                <vnsAbsParam name="gateway" key="gateway" value="40.40.40.100"/>
              </vnsAbsFolder>
              <vnsAbsFolder name="route2" key="ipv6_route">
                <vnsAbsParam name="prefix" key="prefix" value="2005::/64"/>
                <vnsAbsParam name="gateway" key="gateway" value="2004::2828:2866"/>
```

```
            </vnsAbsFolder>
</vnsAbsFolder>
```

The following example shows static route configuration on the fabric for the internal interface of the device:

```
<polUni>
  <fvTenant name="tenant11">
    <l3extOut name="StaticInternal" status="created,modified">
      <l3extLNodeP name="bLeaf-201">
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.11>"
          <ipRouteP ip="20.20.20.0/24>"
            <ipNexthopP nhAddr="30.30.30.102/32"/>
          </ipRouteP>
        </l3extRsNodeL3OutAtt>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/25]"
            ifInstT="ext-svi" encap="vlan-3844" addr="30.30.30.100/28 "mtu="1500"/>
        </l3extLIfP>
      </l3extLNodeP>
      <l3extInstP name="IntInstP">
        <l3extSubnet ip="20.20.20.0/24 "scope="import-security"/>
      </l3extInstP>
      <l3extRsEctx tnFvCtxName="tenant1ctx1"/>
    </l3extOut>
  </fvTenant>
</polUni>
```

Selecting an L3extOut Policy for a Cluster

A specific `l3extOut` policy can be associated with a logical device's interface using its selection policy `vnsLIfCtx`. The following example shows how this is achieved:

```
<vnsLDevCtx ctrctNameOrLbl="webCtrct1" graphNameOrLbl="WebGraph" nodeNameOrLbl="FW">
    <vnsRsLDevCtxToLDev tDn="uni/tn-tenant1/lDevVip-Firewall"/>
    <vnsRsLDevCtxToRtrCfg tnVnsRtrCfgName="FwRtrCfg"/>
    <vnsLIfCtx connNameOrLbl="internal">
        <vnsRsLIfCtxToInstP tDn="uni/tn-tenant1/out-OspfInternal/instP-IntInstP"
          status="created,modified"/>
        <vnsRsLIfCtxToLIf tDn="uni/tn-tenant1/lDevVip-Firewall/lIf-internal"/>
    </vnsLIfCtx>
    <vnsLIfCtx connNameOrLbl="external">
        <vnsRsLIfCtxToInstP tDn="uni/tn-common/out-OspfExternal/instP-ExtInstP"
          status="created,modified"/>
        <vnsRsLIfCtxToLIf tDn="uni/tn-tenant1/lDevVip-Firewall/lIf-external"/>
    </vnsLIfCtx>
</vnsLDevCtx>
```

The `vnsRsLIfCtxToInstP` relation is used to select a particular prefix-based EPG that (`l3extInstP`) is associated with this leg of the service device. You can specify the `redistribute` protocol redistribute property on this relation. The default value for the `redistribute` property is "ospf,bgp". Leaving `redistribute` at the default value causes the Application Policy Infrastructure Controller (APIC) to auto-detect the routing protocols that are configured on each leg and push the appropriate redistribute settings. The automatic settings always redistribute from an interior gateway protocol (OSPF) to an exterior gateway protocol (BGP).

If you want to use a specific redistribute setting, such as static or connected, then you can add those settings to this relation. For example, `redistribute="ospf,bgp,static"` causes the auto-detected settings and `redistribute-static` to be pushed to the service device.

Setting this property to a specific value that does not include the defaults, such as `redistribute="ospf,static,connected"`, causes those exact settings to be pushed to the service device. This is useful in scenarios in which you want to override the defaults that are chosen by the APIC.

Note The relation points to an EPG (`l3extInstP`) and not to the `l3extOut` itself, as there can be multiple such EPGs under an `l3extOut` policy, and different device selection policies could point to those EPGs. This allows for fine control of which prefixes are imported or exported by different service graphs.

The `vnsRsLDevCtxToRtrCfg` relation is used to select a particular `vnsRtrCfg` policy for this device selector. `vnsRtrCfg` policies are needed to specify the router ID that is used by routing protocols, such as Open Shortest Path First (OSPF) or internal Border Gateway Protocol (iBGP), and must be supplied by the user. This router ID is sent to the device.

The following code is an example `vnsRtrCfg` policy:

```
<vnsRtrCfg name="FwRtrCfg" rtrId="180.0.0.10”/>
```

The associated concrete device must have a `vnsRsCIfPathAtt` object, which deploys the device to the same fabric leaf as shown below:

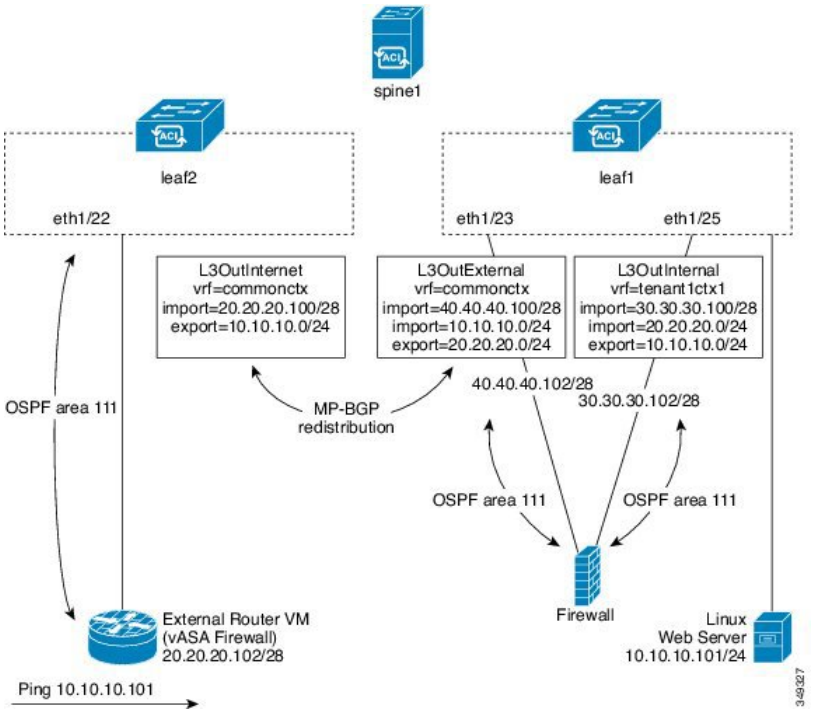
```
<vnsCDev name="ASA">
    <vnsCIf name="Gig0/0">
        <vnsRsCIfPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"/>
    </vnsCIf>
    <vnsCIf name="Gig0/1">
        <vnsRsCIfPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/25]"/>
    </vnsCIf>
    <vnsCMgmt name="devMgmt" host="{{asaIp}}" port="443"/>
    <vnsCCred name="username" value="admin"/>
    <vnsCCredSecret name="password" value="insieme"/>
</vnsCDev>
```

Note When route peering is configured, you do not need to configure bridge domains on the `vnsLIfCtx` selectors. Attempting to configure both bridge domain relations (`vnsRsLIfCtxToBD`) and `l3extInstP` relations (`vnsRsLIfCtxToInstP`) will result in a fault.

Route Peering End-to-End Flow

The following figure shows how route peering works end-to-end.

Figure 4: Route Peering End-to-End Flow



The figure shows an example two leaf switch, single spine switch topology where a Linux web server's IP address is advertised to an external router using route peering. The Linux web server is at IP address 10.10.10.101/24 and is hosted on an ESX server that is connected to `leaf1`. A regular bridge domain-based endpoint group (EPG) is deployed to represent traffic that originates from the web server.

You deploy a service graph that comprises a two-arm routable firewall, with both arms being connected to `leaf1`. There is a virtual routing and forwarding (VRF)-split on the firewall device, meaning that each arm of the firewall is connected to the leaf switch in a different VRF (context). The VRF-split is necessary to ensure that traffic is routed through the service device, rather than being short-circuited by the leaf switch. The external traffic is represented by an `l3extOut` (`L3OutInternet`) that is deployed on `leaf2`. `leaf2` can be viewed as a fabric border-leaf switch in this scenario. You deploy a contract between `L3OutInternet` and the web server EPG. This contract is associated with a service graph that encompasses the firewall device.

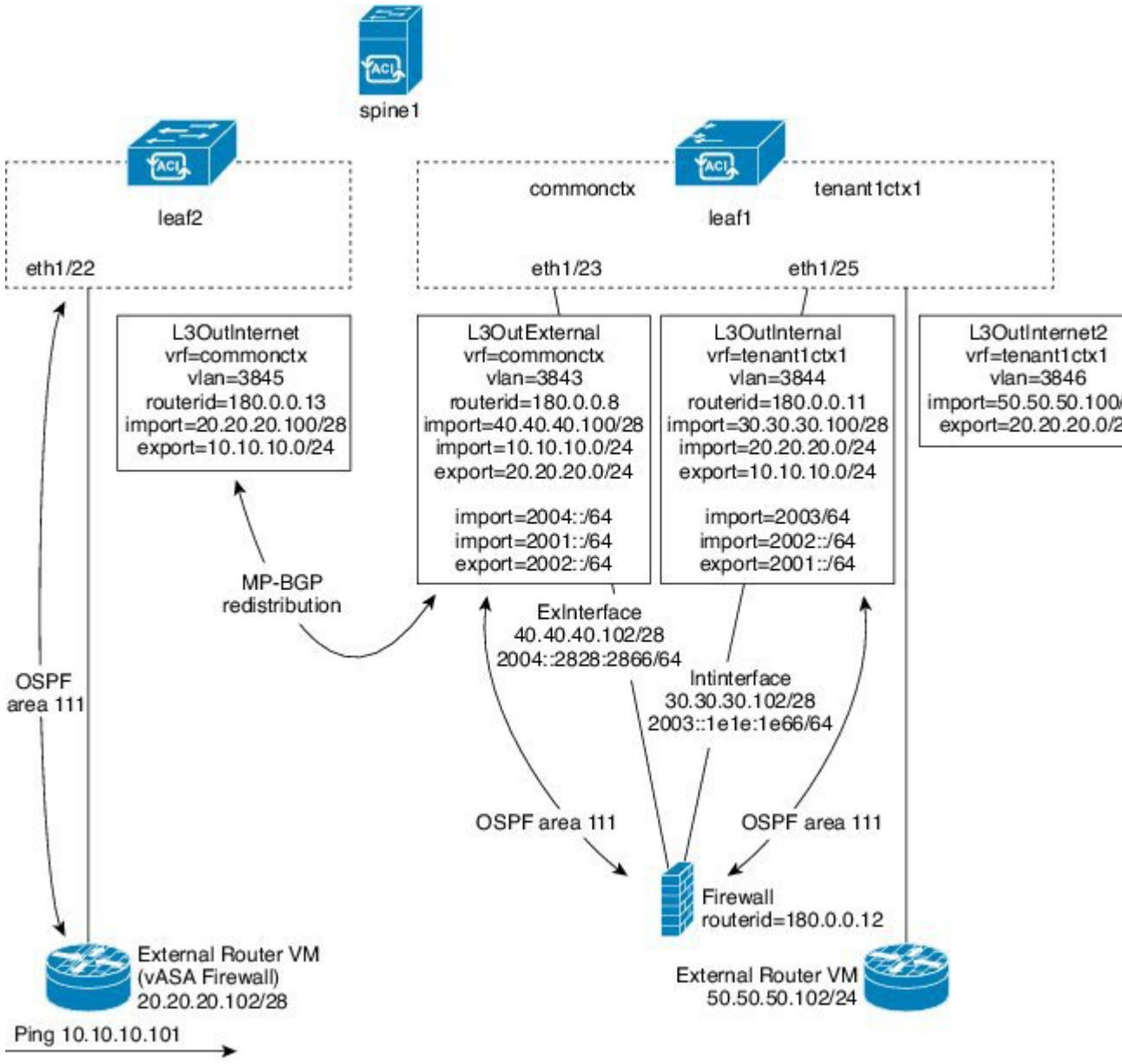
To publish the web server route to the external world, you deploy two `l3extOuts`—`L3OutExternal` and `L3OutInternal`—to the leaf switch ports to which the service device is connected. As a result, Open Shortest Path First (OSPF) peering sessions are established between the leaf switch and the firewall in both of the contexts (`commonctx` and `tenant1ctx1`). The export attribute on these `l3extOuts` control how the routing information is advertised to the border leaf switch. Routes are exchanged internally between the fabric leaf switches using Multiprotocol Border Gateway Protocol (MP-BGP) redistribution.

Ultimately, the web server route is advertised to the external router (IP address 20.20.20.102) using a separate OSPF session. This results in the external router being able to ping the web server without any manual static route configuration.

Cisco Application Centric Infrastructure Fabric Serving As a Transit Routing Domain

You can deploy the Cisco Application Centric Infrastructure (ACI) fabric as a transit routing domain, which is useful when the ACI point of delivery (POD) serves as a transit routing domain between other PODs. In the following illustration, two external `l3extOut`s—`L3OutInternet` and `L3OutInternet2`—are deployed on two border leaf switches. There is a contract associated between these `l3extOut`s, and the contract is attached to a single node service graph containing a firewall service device.

Figure 5: ACI Fabric Serving As a Transit Routing Domain



Two additional `l3extOut`s are deployed on the external and internal legs of the firewall device to establish Open Shortest Path First (OSPF) peering sessions between them. By appropriately configuring the import security control (the `import-security` attribute), you can control which routes are allowed to transit the ACI fabric to the border leaf switches.

Configuring Route Peering Using the GUI

You must perform the following tasks to configure route peering:

1. Create a static VLAN pool that will be used for the encapsulation VLAN between the device and the Cisco Application Centric Infrastructure (ACI) fabric.

   See Creating a Static VLAN Pool Using the GUI, on page 13.

2. Create an external routed domain that will tie together the location (leaf node/path) of the device and the VLAN pool.

   See Creating an External Routed Domain Using the GUI, on page 14.

3. Create an external routed network, which is used to specify the routing configuration in the ACI fabric for route peering.

   See Creating an External Routed Network Using the GUI, on page 14.

4. Create a new router configuration to specify the router ID that will be used on the device.

   See Creating a Router Configuration Using the GUI, on page 16.

5. Create a service graph association, which involves associating the external routed network policy and router configuration with a device selection policy.

   See Creating a Service Graph Association Using the GUI, on page 16.

Creating a Static VLAN Pool Using the GUI

Before creating an external routed network configuration, you must create a static VLAN pool that will be used for the encapsulation VLAN between the device and the fabric.

Step 1 On the menu bar, choose **Fabric** > **Access Policies**.

Step 2 In the Navigation pane, choose **Pools** > **VLAN**.

Step 3 In the Work pane, choose **Actions** > **Create VLAN Pool**.

Step 4 In the **Create VLAN Pool** dialog box, fill in the fields as required, except as specified below:

a) For the **Allocation Mode** radio buttons, choose **Static Allocation**.

b) In **Encap Blocks** section, click **+**.

Step 5 In the **Create Ranges** dialog box, enter a unique range of VLANs and click **OK**.

Step 6 In the **Create VLAN Pool** dialog box, click **Submit**.

Creating an External Routed Domain Using the GUI

You must create an external routed domain that ties together the location (leaf node/path) of the device and the static VLAN pool that you created for route peering.

Step 1 On the menu bar, choose **FABRIC** > **Access Policies**.

Step 2 In the Navigation pane, right-click **Switch Policies** and choose **Configure Interface, PC, and VPC**.

Step 3 In the **Configure Interface, PC, and VPC** dialog box, to configure switch ports connected to Application Policy Infrastructure Controllers (APICs), perform the following actions:

a) Click the large **+** icon next to the switch diagram to create a new profile and configure VLANs for the APIC.
b) From the **Switches** field drop-down list, check the check boxes for the switches to which the APICs are connected.
c) In the **Switch Profile Name** field, enter a name for the profile.
d) Click the **+** icon to configure the ports.
e) Verify that in the **Interface Type** area, the **Individual** radio button is selected.
f) In the **Interfaces** field, enter the ports to which APICs are connected.
g) In the **Interface Selector Name** field, enter the name of the port profile.
h) In the **Interface Policy Group** field, click the **Create One** radio button.
i) In the **Attached Device Type** drop-down list, choose **External Routed Devices**.
j) For the **Domain** radio buttons, click the **Create One** radio button.
k) In the **Domain Name** field, enter the domain name.
l) If you have previously created a VLAN pool, then for the **VLAN** radio buttons, click the **Choose One** radio button. Otherwise, click the **Create One** radio button.

   If you are choosing an existing VLAN pool, in the **VLAN Pool** drop-down list, choose the VLAN pool.

   If you are creating a VLAN pool, in the **VLAN Range** field, enter the VLAN range.

m) Click **Save**, and click **Save** again.
n) Click **Submit**.

Creating an External Routed Network Using the GUI

The external routed network specifies the routing configuration in the Cisco Application Centric Infrastructure (ACI) fabric for route peering.

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_name*** > **Networking** > **External Routed Networks**.

Step 4 In the Work pane, choose **Actions** > **Create Routed Outside**.

Step 5 In the **Create Routed Outside** dialog box, fill in the fields as required, except as specified below:

a) For dynamic routing, put a check in either the **BGP** or **OSPF** check box.

   For open shortest path first (OSPF), fill in the additional OSPF-specific fields.

b) In the **Private Network** drop-down list, choose the private network with which the device will exchange routes.

c) In the **External Routed Domain** drop-down list, choose the external routed domain that you created for route peering.
d) In the **Nodes and Interfaces Protocol Profiles** section, click +.

Step 6 In the **Create Node Profile** dialog box, fill in the fields as required, except as specified below:
a) In the **Nodes** section, click +.

Step 7 In the **Select Node** dialog box, fill in the fields as required, except as specified below:
a) In the **Node ID** drop-down list, choose the node ID where the device is connected.
- For physical devices, the ID should be the node where the physical device is connected to the fabric.
- For virtual devices, the ID should be the node where the server hosting the virtual machine is connected.

b) In the **Router ID** field, enter a router ID that the ACI fabric will use for the routing protocol process.
c) If you are planning to use static routing between the ACI fabric and device, in the **Static Routes** section click +. Otherwise, go to step Step 10, on page 15.

Step 8 In the **Create Static Route** dialog box, fill in the fields as required, except as specified below:
a) In the **Prefix** section, enter a prefix for the static route.
b) In the **Next Hop Addresses** section, click +.
c) Enter the next hop IP address for the static route.
d) Click **Update**.

Step 9 Click **OK**.

Step 10 In the **Select Node** dialog box, click **OK**.

Step 11 If you are using BGP as the dynamic routing protocol with the device, in the **BGP Peer Connectivity Profiles** section, click +. Otherwise, go to step Step 14, on page 15.

Step 12 In the **Create Peer Connectivity Profile** dialog box, fill in the fields as required, except as specified below:
a) In the **Peer Address** field, enter a peer address, which should be an IP address on the device with which the BGP session will be established.

Step 13 In the **Create Peer Connectivity Profile** dialog box, click **OK**.

Step 14 In the **Interface Profiles** section, click +.

Step 15 In the **Create Interface Profile** dialog box, fill in the fields as required.
a) If you are using OSPF as the dynamic routing protocol, enter the OSPF profile information.

Step 16 In the **Interface** section, choose the **SVI** tab.

Step 17 In the **Interface** section, click +.

Step 18 In the **Select SVI Interface** dialog box, fill in the fields as required, except as specified below:
a) For the **Path Type** radio buttons, choose the type that matches how the device is connected to the fabric.
b) In the **Path** drop-down list, choose the path where the device is connected to the fabric.
- For physical devices, this is the path where the physical device is connected to the fabric.
- For virtual devices, this is the path where the server that is hosting the virtual machine is connected.

c) In the **Encap** field, specify the encapsulation VLAN.
d) In the **IP Address** field, specify the IP address to use on the fabric SVI interface.
e) In the **MTU (bytes)** field, specify the maximum transmission unit size, in bytes.

The default value is "inherit", which uses a default value of "9000" on the ACI and typically a default value of "1500" on the remote device. Having different MTU values can cause issues when peering between the ACI and the remote device. If the remote device's MTU value is set to "1500", then set the MTU value on the remote device's `L3Out` object to "9000" to match the ACI's MTU value.

Step 19 Click **OK**.

Step 20 In the **Create Interface Profile** dialog box, click **OK**.

Step 21 In the **Create Node Profile** dialog box, click **OK**.

Step 22 In the **Create Routed Outside** dialog box, click **Next**.

Step 23 In the **External EPG Networks** section, click **+**.

Step 24 In the **Create External Network** dialog box, fill in the fields as required, except as specified below:

a) In the **Subnet** section, click **+**.

Step 25 In the **Create Subnet** dialog box, fill in the fields as required, except as specified below:

a) In the **IP Address** field, enter the IP address or subnet mask.

The subnet mask is equivalent to a network statement that is defined in a traditional routing protocol configuration.

Step 26 Click **OK**.

Step 27 (Optional) Create additional subnets as needed.

Step 28 In the **Create External Network** dialog box, click **OK**.

Step 29 In the **Create Routed Outside** dialog box, click **Finish**.

Creating a Router Configuration Using the GUI

As part of the routing protocol configuration, you must specify the router ID that will be used on the device.

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_name*** > **Services** > **L4-L7** > **Router configurations**.

Step 4 In the Work pane, in the **Router Configurations** table, click **+**.

Step 5 Enter an IP address to use as the router ID on the device.

Step 6 Click **Update**.

Creating a Service Graph Association Using the GUI

You must create a service graph association, which involves associating the external routed network policy and router configuration with a device selection policy.

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** > **Device Selection Policies** > ***device_selection_policy***.

Step 4 In the Navigation pane, choose ***tenant_name*** > **L4-L7 Services** > **Device Selection Policies** > ***device_selection_policy***. *device_selection_policy* is the device selection policy with which you want to perform route peering with the Cisco Application Centric Infrastructure (ACI) fabric.

Step 5 In the Work pane, in the properties section, in the **Router Config** drop-down list, choose the router configuration that you created for route peering.

Step 6 In the Navigation pane, expand the chosen device selection policy and choose the interface that will peer with the ACI fabric.

Step 7 In the Work pane, in the properties section, for the **Associated Network** radio buttons, choose **L3 External Network**.

Step 8 In the **L3 External Network** drop-down list, choose the external routed network that you created for route peering.

The following changes occur:

- The encapsulation VLAN for the interface that is associated with the external routed network is reprogrammed to match the VLAN that is configured as part of the external routed network interface profile
- The external routed network interface and routing protocol configuration is pushed to the leaf switch
- The routing protocol configuration is pushed to the device using the device package

Configuring Route Peering Using the NX-OS-Style CLI

This section provides example commands of using the NX-OS-style CLI to configure route peering.

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.

Example:

```
apic1(config)# tenant 101
```

Step 3 Add a service graph and associate it with a contract.

Example:

```
apic1(config-tenant)# l4l7 graph g1 contract c1
```

Step 4 Add a node (service) that is associated with the device cluster.

Example:

```
apic1(config-graph)# service ASA_FW device-cluster-tenant 101 device-cluster ASA_FW1
```

Step 5 Under the service function, configure the consumer connector and provider cluster-interface.

Example:

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

Step 6 Under the cluster-interface, specify the Layer 3 outside (l3extOut) and the endpoint group (l3extInstP) to be used for route peering with the service device, then exit the connector configuration mode.

Example:

```
apic1(config-connector)# l4l7-peer tenant 101 out l101 epg e101 redistribute bgp
apic1(config-connector)# exit
```

Step 7 Repeat step 5 and step 6 for the provider connector and consumer cluster-interface.

Example:

```
apic1(config-service)# connector provider cluster-interface consumer
apic1(config-connector)# l4l7-peer tenant 101 out l101 epg e101 redistribute bgp
apic1(config-connector)# exit
```

Step 8 (Optional) If you want to disassociate the endpoint group from the connector, use the **no l4l7-peer** command.

Example:

```
apic1(config-connector)# no l4l7-peer tenant 101 out l101 epg e101 redistribute bgp
```

Step 9 Create a router configuration policy under a tenant, supply a router ID for the peer Layer 4 to Layer 7 device, and exit back to the configuration mode.

Example:

```
apic1(config)# tenant 102
apic1(config-tenant)# rtr-cfg bgp1
apic1(config-router)# router-id 1.2.3.5
apic1(config-router)# exit
```

Step 10 Associate the router configuration policy with a particular service device and exit back to the tenant configuration mode.

Example:

```
apic1(config-tenant)# l4l7 graph g2 contract c2 subject http
apic1(config-graph)# service ASA_FW device-cluster-tenant 102 device-cluster ASA_FW2
apic1(config-service)# rtr-cfg bgp1
apic1(config-service)# exit
apic1(config-graph)# exit
```

Step 11 Associate a Layer 3 outside with a leaf interface and a VRF:

Example:

```
apic1(config-tenant)# external-l3 epg e101 l3out l101
apic1(config-tenant-l3ext-epg)# vrf member v101
apic1(config-tenant-l3ext-epg)# match ip 101.101.1.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant 101 vrf v101 l3out l101
apic1(config-leaf-vrf)# ip route 101.101.1.0/24 99.1.1.2
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# vrf member tenant 101 vrf v101 l3out l101
apic1(config-leaf-if)# vlan-domain member dom101
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# ip address 99.1.1.1/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

For the complete configuration for Layer 3 external connectivity (Layer 3 outside) using the named mode, including routing protocols (BGP, OSPF) and route maps, see the *Cisco APIC NX-OS Style CLI Command Reference* document.

Note The external Layer 3 configuration in the CLI is available in two modes: basic mode and named mode. For a given tenant or VRF, user only one of these modes for all external Layer 3 configuration. Route peering is supported only in the named mode.

Troubleshooting Route Peering

If your Cisco Application Centric Infrastructure (ACI) fabric has a route peering or data traffic issue, there are several commands that you can run on ACI fabric leaf switches to troubleshoot the issue.

The following table provides troubleshooting commands that you can run in the switch shell on the fabric leaf switch.

Command	Description
show ip route vrf all	Displays all of the routes in a particular context, including dynamically learned routes.
show ip ospf neighbor vrf all	Displays Open Shortest Path First (OSPF) peering sessions with neighboring devices.
show ip ospf vrf all	Displays the run-time OSPF configuration in each context.
show ip ospf traffic vrf all	Examines OSPF traffic on each virtual routing and forwarding (VRF) context.
show system internal policymgr stats	Displays the contract filter rules on a particular leaf switch and examines the packet hit counts on the rules.

The following table provides a troubleshooting command that you can run in the **vsh_lc** shell.

Command	Description
show system internal aclqos prefix	Examines the IPv4 prefix association rules on a particular leaf switch and the traffic hit counts on the rules.

In addition to the shell commands, you can check the following things to help with troubleshooting:

- Health count on the device
- All of the faults and `NwIssues` under a particular tenant

Verifying the Leaf Switch Route Peering Functionality Using the CLI

You can use switch shell commands on the fabric leaf to verify the leaf switch configuration and route peering functionality.

Step 1 On the fabric leaf switch where the device is connected, verify that the SVI interface is configured:

```
fab2-leaf3# show ip interface vrf user1:global
IP Interface Status for VRF "user1:global"
vlan30, Interface status: protocol-up/link-up/admin-up, iod: 134,
  IP address: 1.1.1.1, IP subnet: 1.1.1.0/30
  IP broadcast address: 255.255.255.255
  IP primary address route-preference: 1, tag: 0
lo3, Interface status: protocol-up/link-up/admin-up, iod: 133,
  IP address: 10.10.10.1, IP subnet: 10.10.10.1/32
  IP broadcast address: 255.255.255.255
  IP primary address route-preference: 1, tag: 0

fab2-leaf3#
```

Interface vlan30 contains the SVI interface configuration and Interface lo3 contains the router ID specified in the external routed network configuration.

Step 2 Verify the Open Shortest Path First (OSPF) configuration on the fabric leaf switch:

```
fab2-leaf3# show ip ospf vrf user1:global

 Routing Process default with ID 10.10.10.1 VRF user1:global
 Stateful High Availability enabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Table-map using route-map exp-ctx-2949120-deny-external-tag
 Redistributing External Routes from
   static route-map exp-ctx-st-2949120
   bgp route-map exp-ctx-proto-2949120
   eigrp route-map exp-ctx-proto-2949120
 Maximum number of non self-generated LSA allowed 100000
    (feature configured but inactive)
    Current number of non self-generated LSA 1
    Threshold for warning message 75%
    Ignore-time 5 minutes, reset-time 10 minutes
    Ignore-count allowed 5, current ignore-count 0
 Administrative distance 110
 Reference Bandwidth is 40000 Mbps
 SPF throttling delay time of 200.000 msecs,
   SPF throttling hold time of 1000.000 msecs,
   SPF throttling maximum wait time of 5000.000 msecs
 LSA throttling start time of 0.000 msecs,
   LSA throttling hold interval of 5000.000 msecs,
   LSA throttling maximum wait time of 5000.000 msecs
 Minimum LSA arrival 1000.000 msec
 LSA group pacing timer 10 secs
 Maximum paths to destination 8
 Number of external LSAs 0, checksum sum 0x0
 Number of opaque AS LSAs 0, checksum sum 0x0
 Number of areas is 1, 1 normal, 0 stub, 0 nssa
 Number of active areas is 1, 1 normal, 0 stub, 0 nssa
   Area (0.0.0.200)
        Area has existed for 00:17:55
        Interfaces in this area: 1 Active interfaces: 1
        Passive interfaces: 0  Loopback interfaces: 0
        SPF calculation has run 4 times
```

```
         Last SPF ran for 0.000273s
        Area ranges are
        Area-filter in 'exp-ctx-proto-2949120'
        Number of LSAs: 3, checksum sum 0x0
fab2-leaf3#
```

Step 3 Verify the OSPF neighbor relationship on the fabric leaf switch:

```
fab2-leaf3# show ip ospf neighbors vrf user1:global
 OSPF Process ID default VRF user1:global
 Total number of neighbors: 1
 Neighbor ID     Pri State            Up Time  Address         Interface
 10.10.10.2        1 FULL/BDR         00:03:02 1.1.1.2         Vlan30
fab2-leaf3#
```

Step 4 Verify that the routes are being learned by the fabric leaf switch:

```
fab2-leaf3# show ip route vrf user1:global
IP Route Table for VRF "user1:global"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

1.1.1.0/30, ubest/mbest: 1/0, attached, direct
    *via 1.1.1.1, vlan30, [1/0], 00:26:50, direct
1.1.1.1/32, ubest/mbest: 1/0, attached
    *via 1.1.1.1, vlan30, [1/0], 00:26:50, local, local
2.2.2.0/24, ubest/mbest: 1/0
    *via 1.1.1.2, vlan30, [110/20], 00:06:19, ospf-default, type-2
10.10.10.1/32, ubest/mbest: 2/0, attached, direct
    *via 10.10.10.1, lo3, [1/0], 00:26:50, local, local
    *via 10.10.10.1, lo3, [1/0], 00:26:50, direct
10.122.254.0/24, ubest/mbest: 1/0
    *via 1.1.1.2, vlan30, [110/20], 00:06:19, ospf-default, type-2
fab2-leaf3#
```

Step 5 Verify that OSPF has been configured on the device, which is a Cisco ASAv in this example:

```
ciscoasa# show running-config
: Saved
:
: Serial Number: 9AGRM5NBEXG
: Hardware:   ASAv, 2048 MB RAM, CPU Xeon 5500 series 2133 MHz
:
ASA Version 9.3(1)
!
hostname ciscoasa
enable password 8Ry2YjIyt7RRXU24 encrypted
names
!
interface GigabitEthernet0/0
 nameif internalIf
 security-level 100
 ip address 2.2.2.1 255.255.255.0
!
interface GigabitEthernet0/1
 nameif externalIf
 security-level 50
 ip address 1.1.1.2 255.255.255.252
!
<<..>>
router ospf 1
 router-id 10.10.10.2
 network 1.1.1.0 255.255.255.252 area 200
```

```
 area 200
 log-adj-changes
 redistribute connected
 redistribute static
!
```