Performing Tasks Using REST API

- Part I: Layer 3 Configuration, on page 1
- Part II: External Routing (L3Out) Configuration, on page 26

Part I: Layer 3 Configuration

Configuring Common Pervasive Gateway Using REST API

Configuring Common Pervasive Gateway Using the REST API

Before you begin

- The tenant, VRF, and bridge domain are created.

Procedure

Configure common pervasive gateway.

In the following example REST API XML, the bolded text is relevant to configuring a common pervasive gateway.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="test">
    <fvCtx name="test"/>

    <fvBD name="test" vmac="12:34:56:78:9a:bc">
      <fvRsCtx tnFvCtxName="test"/>
      <!-- Primary address -->
      <fvSubnet ip="192.168.15.254/24" preferred="yes"/>
      <!-- Virtual address -->
      <fvSubnet ip="192.168.15.1/24" virtual="yes"/>
    </fvBD>

    <fvAp name="test">
```

```
      <fvAEPg name="web">
        <fvRsBd tnFvBDName="test"/>
        <fvRsPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/3]" encap="vlan-1002"/>
      </fvAEPg>
    </fvAp>
  </fvTenant>
</polUni>
```

Configuring IP Aging Using REST API

Configuring IP Aging Using the REST API

This section explains how to enable and disable the IP aging policy using the REST API.

Procedure

Step 1 To enable the IP aging policy:

Example:

```
<epIpAgingP adminSt="enabled" descr="" dn="uni/infra/ipAgingP-default" name="default" ownerKey=""
ownerTag=""/>
```

Step 2 To disable the IP aging policy:

Example:

```
<epIpAgingP adminSt="disabled" descr="" dn="uni/infra/ipAgingP-default" name="default" ownerKey=""
ownerTag=""/>
```

What to do next

To specify the interval used for tracking IP addresses on endpoints, create an Endpoint Retention policy by sending a post with XML such as the following example:

```
<fvEpRetPol bounceAgeIntvl="630" bounceTrig="protocol"
holdIntvl="350" lcOwn="local" localEpAgeIntvl="900" moveFreq="256"
name="EndpointPol1" remoteEpAgeIntvl="350"/>
```

Configuring a Static Route on a Bridge Domain Using REST API

Configuring a Static Route on a Bridge Domain Using the REST API

- When creating the subnet for the static route, it is configured under the EPG (fvSubnet object under fvAEPg), associated with the pervasive BD (fvBD), not the BD itself.
- The subnet mask must be /32 (/128 for IPv6) pointing to one IP address or one endpoint. It is contained in the EPG associated with the pervasive BD.

Before you begin

The tenant, VRF, BD, and EPG have been created.

Procedure

To configure a static route for the BD used in a pervasive gateway, enter a post such as the following example:

Example:

```
      <fvAEPg name="ep1">
    <fvRsBd tnFvBDName="bd1"/>
          <fvSubnet ip="2002:0db8:85a3:0000:0000:8a2e:0370:7344/128" ctrl="no-default-gateway"   >
              <fvEpReachability>
                  <ipNexthopEpP  nhAddr="2001:0db8:85a3:0000:0000:8a2e:0370:7343/128" />
              </fvEpReachability>
          </fvSubnet>
      </fvAEPg>
```

Configuring IPv6 Neighbor Discovery Using REST API

Creating the Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery on the Bridge Domain Using the REST API

Procedure

Create a tenant, VRF, bridge domain with a neighbor discovery interface policy and a neighbor discovery prefix policy.

Example:

```
<fvTenant descr="" dn="uni/tn-ExampleCorp" name="ExampleCorp" ownerKey="" ownerTag="">
    <ndIfPol name="NDPol001" ctrl="managed-cfg”  descr="" hopLimit="64" mtu="1500" nsIntvl="1000"
nsRetries="3" ownerKey="" ownerTag="" raIntvl="600" raLifetime="1800" reachableTime="0"
retransTimer="0"/>
    <fvCtx descr="" knwMcastAct="permit" name="pvn1" ownerKey="" ownerTag="" pcEnfPref="enforced">
    </fvCtx>
    <fvBD arpFlood="no" descr="" mac="00:22:BD:F8:19:FF" multiDstPktAct="bd-flood" name="bd1"
ownerKey="" ownerTag="" unicastRoute="yes" unkMacUcastAct="proxy" unkMcastAct="flood">
        <fvRsBDToNdP tnNdIfPolName="NDPol001"/>
        <fvRsCtx tnFvCtxName="pvn1"/>
        <fvSubnet ctrl="nd" descr="" ip="34::1/64" name="" preferred="no" scope="private">
            <fvRsNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
        </fvSubnet>
        <fvSubnet ctrl="nd" descr="" ip="33::1/64" name="" preferred="no" scope="private">
            <fvRsNdPfxPol tnNdPfxPolName="NDPfxPol002"/>
        </fvSubnet>
    </fvBD>
    <ndPfxPol ctrl="auto-cfg,on-link" descr="" lifetime="1000" name="NDPfxPol001" ownerKey=""
ownerTag="" prefLifetime="1000"/>
    <ndPfxPol ctrl="auto-cfg,on-link" descr="" lifetime="4294967295" name="NDPfxPol002" ownerKey=""
 ownerTag="" prefLifetime="4294967295"/>
</fvTenant>
```

Note
If you have a public subnet when you configure the routed outside, you must associate the bridge domain with the outside configuration.

Configuring an IPv6 Neighbor Discovery Interface Policy with RA on a Layer 3 Interface Using the REST API

Procedure

Configure an IPv6 neighbor discovery interface policy and associate it with a Layer 3 interface:

The following example displays the configuration in a non-VPC set up.

Example:

```
<fvTenant dn="uni/tn-ExampleCorp" name="ExampleCorp">
  <ndIfPol name="NDPol001" ctrl="managed-cfg" hopLimit="64" mtu="1500" nsIntvl="1000" nsRetries="3"
 raIntvl="600" raLifetime="1800" reachableTime="0" retransTimer="0"/>
  <fvCtx name="pvn1" pcEnfPref="enforced">
                  </fvCtx>
  <l3extOut enforceRtctrl="export" name="l3extOut001">
    <l3extRsEctx tnFvCtxName="pvn1"/>
    <l3extLNodeP name="lnodeP001">
      <l3extRsNodeL3OutAtt rtrId="11.11.205.1" rtrIdLoopBack="yes" tDn="topology/pod-2/node-2011"/>
      <l3extLIfP name="lifP001">
        <l3extRsPathL3OutAtt addr="2001:20:21:22::2/64" ifInstT="l3-port" llAddr="::"
mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit" tDn="topology/pod-2/paths-2011/pathep-[eth1/1]">

          <ndPfxP>
            <ndRsPfxPToNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
          </ndPfxP>
        </l3extRsPathL3OutAtt>
        <l3extRsNdIfPol tnNdIfPolName="NDPol001"/>
      </l3extLIfP>
    </l3extLNodeP>
    <l3extInstP name="instp"/>
  </l3extOut>
  <ndPfxPol ctrl="auto-cfg,on-link" descr="" lifetime="1000" name="NDPfxPol001" ownerKey="" ownerTag=""
 prefLifetime="1000"/>
</fvTenant>
```

Note
For VPC ports, ndPfxP must be a child of l3extMember instead of l3extRsNodeL3OutAtt. The following code snippet shows the configuration in a VPC setup.

```
<l3extLNodeP name="lnodeP001">
<l3extRsNodeL3OutAtt rtrId="11.11.205.1" rtrIdLoopBack="yes" tDn="topology/pod-2/node-2011"/>
<l3extRsNodeL3OutAtt rtrId="12.12.205.1" rtrIdLoopBack="yes" tDn="topology/pod-2/node-2012"/>
  <l3extLIfP name="lifP002">
    <l3extRsPathL3OutAtt addr="0.0.0.0" encap="vlan-205" ifInstT="ext-svi" llAddr="::"
mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
tDn="topology/pod-2/protpaths-2011-2012/pathep-[vpc7]" >
      <l3extMember addr="2001:20:25:1::1/64" descr="" llAddr="::" name="" nameAlias="" side="A">
        <ndPfxP >
```

```
        <ndRsPfxPToNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
      </ndPfxP>
    </l3extMember>
    <l3extMember addr="2001:20:25:1::2/64" descr="" llAddr="::" name="" nameAlias="" side="B">
      <ndPfxP >
        <ndRsPfxPToNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
      </ndPfxP>
    </l3extMember>
  </l3extRsPathL3OutAtt>
  <l3extRsNdIfPol tnNdIfPolName="NDPol001"/>    </l3extLIfP>
</l3extLNodeP>
```

Configuring Neighbor Discovery Duplicate Address Detection Using the REST API

Procedure

Step 1 Disable the Neighbor Discovery Duplicate Address Detection process for a subnet by changing the value of the ipv6Dad entry for that subnet to **disabled**.

The following example shows how to set the Neighbor Discovery Duplicate Address Detection entry for the 2001:DB8:A::11/64 subnet to **disabled**:

Note
In the following REST API example, long single lines of text are broken up with the \ character to improve readability.

Example:

```
                <l3extRsPathL3OutAtt addr="2001:DB8:A::2/64" autostate="enabled" \
                 childAction="" descr="" encap="vlan-1035" encapScope="local" \
                 ifInstT="ext-svi" ipv6Dad="enabled" llAddr=": :" \
                 mac="00:22:BD:F8:19:DD"  mtu="inherit"  \
                 rn="rspathL3OutAtt-[topology/pod-1/paths-105/pathep-[eth1/1]]"  \
                 status=""  tDn="topology/pod-1/paths-105/pathep-[eth1/1]"  >
                    <l3extIp addr="2001:DB8:A::11/64" childAction="" descr="" \
                     ipv6Dad="disabled"  name="" nameAlias="" \
                     rn="addr-[2001:DB8:A::11/64]" status=""/>

                </l3extRsPathL3OutAtt>
            </l3extLIfP>
        </l3extLNodeP>
```

Step 2 Enter the **show ipv6 int** command on the leaf switch to verify that the configuration was pushed out correctly to the leaf switch. For example:

```
swtb23-leaf5# show ipv6 int vrf icmpv6:v1
IPv6 Interface Status for VRF "icmpv6:v1"(9)

vlan2, Interface status: protocol-up/link-up/admin-up, iod: 73
if_mode: ext
```

```
  IPv6 address:
    2001:DB8:A::2/64 [VALID] [PREFERRED]
    2001:DB8:A::11/64 [VALID] [dad-disabled]
  IPv6 subnet:  2001:DB8:A::/64
  IPv6 link-local address: fe80::863d:c6ff:fe9f:eb8b/10 (Default) [VALID]
```

Configuring Microsoft NLB Using REST API

Configuring Microsoft NLB in Unicast Mode Using the REST API

Procedure

To configure Microsoft NLB in unicast mode, send a post with XML such as the following example:

Example:

```
https://apic-ip-address/api/node/mo/uni/.xml
<polUni>
    <fvTenant name="tn2" >
        <fvCtx name="ctx1"/>
        <fvBD name="bd2">
            <fvRsCtx tnFvCtxName="ctx1" />
        </fvBD>
        <fvAp name = "ap1">
            <fvAEPg name = "ep1">
                <fvRsBd tnFvBDName = "bd2"/>
                <fvSubnet ip="10.0.1.1/32" scope="public" ctrl="no-default-gateway">
                    <fvEpNlb mac="12:21:21:35" mode="mode-uc"/>
                </fvSubnet>
            </fvAEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

Configuring Microsoft NLB in Multicast Mode Using the REST API

Procedure

To configure Microsoft NLB in multicast mode, send a post with XML such as the following example:

Example:

```
https://apic-ip-address/api/node/mo/uni/.xml
<polUni>
    <fvTenant name="tn2" >
        <fvCtx name="ctx1"/>
        <fvBD name="bd2">
            <fvRsCtx tnFvCtxName="ctx1" />
        </fvBD>
        <fvAp name = "ap1">
```

```
            <fvAEPg name = "ep1">
                <fvRsBd tnFvBDName = "bd2"/>
                <fvSubnet ip="2001:0db8:85a3:0000:0000:8a2e:0370:7344/128" scope="public"
ctrl="no-default-gateway">
                    <fvEpNlb mac="03:21:21:35" mode="mode-mcast--static"/>
                </fvSubnet>
                <fvRsPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/6]" encap="vlan-911" >
                    <fvNlbStaticGroup mac = "03:21:21:35"  />
                </fvRsPathAtt>
            </fvAEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

Configuring Microsoft NLB in IGMP Mode Using the REST API

Procedure

To configure Microsoft NLB in IGMP mode, send a post with XML such as the following example:

Example:

```
https://apic-ip-address/api/node/mo/uni/.xml
<polUni>
    <fvTenant name="tn2" >
        <fvCtx name="ctx1"/>
        <fvBD name="bd2">
            <fvRsCtx tnFvCtxName="ctx1" />
        </fvBD>
        <fvAp name = "ap1">
            <fvAEPg name = "ep1">
                <fvRsBd tnFvBDName = "bd2"/>
                <fvSubnet ip="10.0.1.3/32" scope="public" ctrl="no-default-gateway">
                    <fvEpNlb group ="224.132.18.17" mode="mode-mcast-igmp" />
                </fvSubnet>
            </fvAEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

Configuring IGMP Snooping Using REST API

Configuring and Assigning an IGMP Snooping Policy to a Bridge Domain using the REST API

SUMMARY STEPS

1. To configure an IGMP Snooping policy and assign it to a bridge domain, send a post with XML such as the following example:

DETAILED STEPS

Procedure

To configure an IGMP Snooping policy and assign it to a bridge domain, send a post with XML such as the following example:

Example:

```
https://apic-ip-address/api/node/mo/uni/.xml
<fvTenant name="mcast_tenant1">
<!-- Create an IGMP snooping template, and provide the options -->
<igmpSnoopPol name="igmp_snp_bd_21"
ver="v2"
adminSt="enabled"
lastMbrIntvl="1"
queryIntvl="125"
rspIntvl="10"
startQueryCnt="2"
startQueryIntvl="31"
/>
<fvCtx name="ip_video"/>
<fvBD name="bd_21">
<fvRsCtx tnFvCtxName="ip_video"/>
<!-- Bind IGMP snooping to a BD -->
<fvRsIgmpsn tnIgmpSnoopPolName="igmp_snp_bd_21"/>
</fvBD></fvTenant>
```

This example creates and configures the IGMP Snooping policy, igmp_snp_bd_12 with the following properties, and binds the IGMP policy, igmp_snp_bd_21, to bridge domain, bd_21:

- Administrative state is enabled
- Last Member Query Interval is the default 1 second.
- Query Interval is the default 125.
- Query Response interval is the default 10 seconds
- The Start Query Count is the default 2 messages
- The Start Query interval is 31 seconds.
- Setting the Querier Version to v2.

Enabling IGMP Snooping and Multicast on Static Ports Using the REST API

You can enable IGMP snooping and multicast processing on ports that have been statically assigned to an EPG. You can create and assign access groups of users that are permitted or denied access to the IGMP snoop and multicast traffic enabled on those ports.

SUMMARY STEPS

1. To configure application EPGs with static ports, enable those ports to receive and process IGMP snooping and multicast traffic, and assign groups to access or be denied access to that traffic, send a post with XML such as the following example.

DETAILED STEPS

Procedure

To configure application EPGs with static ports, enable those ports to receive and process IGMP snooping and multicast traffic, and assign groups to access or be denied access to that traffic, send a post with XML such as the following example.

In the following example, IGMP snooping is enabled on `leaf 102` interface `1/10` on VLAN `202`. Multicast IP addresses `224.1.1.1` and `225.1.1.1` are associated with this port.

Example:

```
https://apic-ip-address/api/node/mo/uni/.xml
<fvTenant name="tenant_A">
  <fvAp name="application">
    <fvAEPg name="epg_A">
      <fvRsPathAtt encap="vlan-202" instrImedcy="immediate" mode="regular"
tDn="topology/pod-1/paths-102/pathep-[eth1/10]">
        <!-- IGMP snooping static group case -->
        <igmpSnoopStaticGroup group="224.1.1.1" source="0.0.0.0"/>
        <igmpSnoopStaticGroup group="225.1.1.1" source="2.2.2.2"/>
      </fvRsPathAtt>
    </fvAEPg>
  </fvAp>
</fvTenant>
```

Enabling Group Access to IGMP Snooping and Multicast using the REST API

After you have enabled IGMP snooping and multicast on ports that have been statically assigned to an EPG, you can then create and assign access groups of users that are permitted or denied access to the IGMP snooping and multicast traffic enabled on those ports.

Procedure

To define the access group, `F23broker`, send a post with XML such as in the following example.

The example configures access group `F23broker`, associated with tenant_A, Rmap_A, application_A, epg_A, on leaf 102, interface 1/10, VLAN 202. By association with Rmap_A, the access group `F23broker` has access to multicast traffic received at multicast address 226.1.1.1/24 and is denied access to traffic received at multicast address 227.1.1.1/24.

Example:

<!-- api/node/mo/uni/.xml --> <fvTenant name="tenant_A"> <pimRouteMapPol name="Rmap_A"> <pimRouteMapEntry action="permit" grp="226.1.1.1/24" order="10"/> <pimRouteMapEntry action="deny" grp="227.1.1.1/24" order="20"/> </pimRouteMapPol> <fvAp name="application_A"> <fvAEPg name="epg_A"> <fvRsPathAtt encap="vlan-202"

instrImedcy="immediate" mode="regular" tDn="topology/pod-1/paths-102/pathep-[eth1/10]"> <!-- IGMP snooping access group case --> <igmpSnoopAccessGroup name="F23broker"> <igmpRsSnoopAccessGroupFilterRMap tnPimRouteMapPolName="Rmap_A"/> </igmpSnoopAccessGroup> </fvRsPathAtt> </fvAEPg> </fvAp> </fvTenant>

Configuring MLD Snooping Using REST API

Configuring and Assigning an MLD Snooping Policy to a Bridge Domain using the REST API

Procedure

To configure an MLD Snooping policy and assign it to a bridge domain, send a post with XML such as the following example:

Example:

```
https://apic-ip-address/api/node/mo/uni/.xml
<fvTenant name="mldsn">
    <mldSnoopPol adminSt="enabled" ctrl="fast-leave,querier" name="mldsn-it-fabric-querier-policy"
queryIntvl="125"
        rspIntvl="10" startQueryCnt="2" startQueryIntvl="31" status=""/>
    <fvBD name="mldsn-bd3">
        <fvRsMldsn status="" tnMldSnoopPolName="mldsn-it-policy"/>
    </fvBD>
</fvTenant>
```

This example creates and configures the MLD Snooping policy `mldsn` with the following properties, and binds the MLD policy `mldsn-it-fabric-querier-policy` to bridge domain `mldsn-bd3`:

- Fast leave processing is enabled
- Querier processing is enabled
- Query Interval is set at 125
- Max query response time is set at 10
- Number of initial queries to send is set at 2
- Time for sending initial queries is set at 31

Configuring IP Multicast Using REST API

Configuring Layer 3 Multicast Using REST API

Procedure

Step 1 Configure a tenant and VRF and enable multicast on a VRF.

Example:

```
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
   <fvCtx knwMcastAct="permit" name="ctx1">
       <pimCtxP mtu="1500">
       </pimCtxP>
   </fvCtx>
  </fvTenant>
```

Step 2 Configure L3 Out and enable multicast (PIM, IGMP) on the L3 Out.

Example:

```
<l3extOut enforceRtctrl="export" name="l3out-pim_l3out1">
       <l3extRsEctx tnFvCtxName="ctx1"/>
       <l3extLNodeP configIssues="" name="bLeaf-CTX1-101">
         <l3extRsNodeL3OutAtt rtrId="200.0.0.1" rtrIdLoopBack="yes" tDn="topology/pod-1/node-101"/>
         <l3extLIfP name="if-PIM_Tenant-CTX1" tag="yellow-green">
           <igmpIfP/>
           <pimIfP>
             <pimRsIfPol tDn="uni/tn-PIM_Tenant/pimifpol-pim_pol1"/>
           </pimIfP>
           <l3extRsPathL3OutAtt addr="131.1.1.1/24" ifInstT="l3-port" mode="regular" mtu="1500"
tDn="topology/pod-1/paths-101/pathep-[eth1/46]"/>
         </l3extLIfP>
       </l3extLNodeP>
       <l3extRsL3DomAtt tDn="uni/l3dom-l3outDom"/>
       <l3extInstP name="l3out-PIM_Tenant-CTX1-1topo" >
       </l3extInstP>
       <pimExtP enabledAf="ipv4-mcast" name="pim"/>
     </l3extOut>
```

Step 3 Configure a BD under the tenant and enable multicast and IGMP on the BD.

Example:

```
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
   <fvBD arpFlood="yes" mcastAllow="yes" multiDstPktAct="bd-flood" name="bd2" type="regular"
unicastRoute="yes" unkMacUcastAct="flood" unkMcastAct="flood">
       <igmpIfP/>
       <fvRsBDToOut tnL3extOutName="l3out-pim_l3out1"/>
       <fvRsCtx tnFvCtxName="ctx1"/>
       <fvRsIgmpsn/>
       <fvSubnet ctrl="" ip="41.1.1.254/24" preferred="no" scope="private" virtual="no"/>
     </fvBD>
 </fvTenant>
```

Step 4 Configure an IGMP policy and assign it to the BD.

Example:

```
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
    <igmpIfPol grpTimeout="260" lastMbrCnt="2" lastMbrRespTime="1" name="igmp_pol" querierTimeout="255"
 queryIntvl="125" robustFac="2" rspIntvl="10" startQueryCnt="2" startQueryIntvl="125" ver="v2">
     </igmpIfPol>
      <fvBD arpFlood="yes" mcastAllow="yes" name="bd2">
      <igmpIfP>
         <igmpRsIfPol tDn="uni/tn-PIM_Tenant/igmpIfPol-igmp_pol"/>
       </igmpIfP>
      </fvBD>
 </fvTenant>
```

Step 5 Configure a route map, PIM, and RP policy on the VRF.

Note
When configuring a fabric RP using the REST API, first configure a static RP.

Example:

Configuring a static RP:

```
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
     <pimRouteMapPol name="rootMap">
       <pimRouteMapEntry action="permit" grp="224.0.0.0/4" order="10" rp="0.0.0.0" src="0.0.0.0/0"/>

     </pimRouteMapPol>
     <fvCtx knwMcastAct="permit" name="ctx1">
       <pimCtxP ctrl="" mtu="1500">
         <pimStaticRPPol>
           <pimStaticRPEntryPol rpIp="131.1.1.2">
             <pimRPGrpRangePol>
               <rtdmcRsFilterToRtMapPol tDn="uni/tn-PIM_Tenant/rtmap-rootMap"/>
             </pimRPGrpRangePol>
           </pimStaticRPEntryPol>
         </pimStaticRPPol>
       </pimCtxP>
   </fvCtx>
</fvTenant>
```

Configuring a fabric RP:

```
<fvTenant name="t0">
            <pimRouteMapPol name="fabricrp-rtmap">
               <pimRouteMapEntry grp="226.20.0.0/24" order="1"  />
            </pimRouteMapPol>
  <fvCtx name="ctx1">
         <pimCtxP ctrl="">
               <pimFabricRPPol status="">
                   <pimStaticRPEntryPol rpIp="6.6.6.6">
                       <pimRPGrpRangePol>
                          <rtdmcRsFilterToRtMapPol tDn="uni/tn-t0/rtmap-fabricrp-rtmap" />
                       </pimRPGrpRangePol>
                   </pimStaticRPEntryPol>
               </pimFabricRPPol>
            </pimCtxP>
  </fvCtx>
</fvTenant>
```

Step 6 Configure a PIM interface policy and apply it on the L3 Out.

Example:

```
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
  <pimIfPol authKey="" authT="none" ctrl="" drDelay="60" drPrio="1" helloItvl="30000" itvl="60"
name="pim_pol1"/>
  <l3extOut enforceRtctrl="export" name="l3out-pim_l3out1" targetDscp="unspecified">
       <l3extRsEctx tnFvCtxName="ctx1"/>
       <l3extLNodeP  name="bLeaf-CTX1-101">
         <l3extRsNodeL3OutAtt rtrId="200.0.0.1" rtrIdLoopBack="yes" tDn="topology/pod-1/node-101"/>
         <l3extLIfP name="if-SIRI_VPC_src_recv-CTX1" tag="yellow-green">
           <pimIfP>
             <pimRsIfPol tDn="uni/tn-tn-PIM_Tenant/pimifpol-pim_pol1"/>
           </pimIfP>
         </l3extLIfP>
       </l3extLNodeP>
     </l3extOut>
 </fvTenant>
```

Step 7 Configure inter-VRF multicast.

Example:

```
<fvTenant name="t0">
            <pimRouteMapPol name="intervrf" status="">
               <pimRouteMapEntry grp="225.0.0.0/24" order="1" status=""/>
               <pimRouteMapEntry grp="226.0.0.0/24" order="2" status=""/>
               <pimRouteMapEntry grp="228.0.0.0/24" order="3" status="deleted"/>
            </pimRouteMapPol>
        <fvCtx name="ctx1">
            <pimCtxP ctrl="">
               <pimInterVRFPol status="">
                   <pimInterVRFEntryPol srcVrfDn="uni/tn-t0/ctx-stig_r_ctx" >
                    <rtdmcRsFilterToRtMapPol tDn="uni/tn-t0/rtmap-intervrf" />
                   </pimInterVRFEntryPol>
               </pimInterVRFPol>
            </pimCtxP>
        </fvCtx>
</fvTenant>
```

Configuring Layer 3 IPv6 Multicast Using REST API

Before you begin

- The desired VRF, bridge domains, Layer 3 Out interfaces with IPv6 addresses must be configured to enable PIM6. For Layer 3 Out, for IPv6 multicast to work, an IPv6 loopback address is configured for the node in the logical node profile.
- Basic unicast network must be configured.

Procedure

Step 1 Enable PIM6 on the VRF.

Example:

```
<fvTenant name="t0">
  <fvCtx name="ctx1" pcEnfPref="unenforced" >
    <pimIPV6CtxP ctrl=""  mtu="1500" />
  </fvCtx>
</fvTenant>
```

Step 2 Enable PIM6 on the Layer 3 Out.

Example:

```
<fvTenant dn="uni/tn-t0" name="t0">
  <l3extOut enforceRtctrl="export" name="bl_l3out_1">
    <pimExtP enabledAf="ipv6-mcast" name="pim"/>
  </l3extOut>
</fvTenant>
```

Step 3 Enable PIM6 on the BD.

Example:

```
<fvTenant name="t0" >
```

```
  <fvBD name="BD_VPC5" ipv6McastAllow="yes" >
    <fvRsCtx tnFvCtxName="ctx1" />
    <fvSubnet ip="124:1::ffff:ffff:ffff:0/64" scope="public"/>
  </fvBD>
</fvTenant>
```

Step 4 Configure Static Rendezvous Point.

Example:

```
<fvTenant name="t0">
  <pimRouteMapPol dn="uni/tn-t0/rtmap-static_101_ipv6" name="static_101_ipv6">
    <pimRouteMapEntry action="permit" grp="ff00::/8" order="1" rp="2001:0:2001:2001:1:1:1:1/128"
src="::"/>
  </pimRouteMapPol>
  <fvCtx name="ctx1" pcEnfPref="unenforced">
    <pimIPV6CtxP ctrl="" mtu="1500">
      <pimStaticRPPol>
        <pimStaticRPEntryPol rpIp="2001:0:2001:2001:1:1:1:1">
          <pimRPGrpRangePol>
            <rtdmcRsFilterToRtMapPol tDn="uni/tn-t0/rtmap-static_101_ipv6"/>
          </pimRPGrpRangePol>
        </pimStaticRPEntryPol>
      </pimStaticRPPol>
    </pimIPV6CtxP>
  </fvCtx>
</fvTenant>
```

Step 5 Configure a PIM6 interface policy and apply it on the Layer 3 Out.

Example:

```
<fvTenant dn="uni/tn-t0" name="t0">
  <l3extOut enforceRtctrl="export" name="bl_l3out_1">
    <l3extLNodeP annotation="" configIssues="" descr="" name="common_np1" nameAlias="" ownerKey=""
ownerTag="" tag="yellow-green" targetDscp="unspecified">
       <l3extLIfP annotation="" descr="" name="common_intp1_v6" nameAlias="" ownerKey="" ownerTag=""
 prio="unspecified" tag="yellow-green">
            <pimIPV6IfP annotation="" descr="" name="" nameAlias="">
            <pimRsV6IfPol annotation="" tDn="uni/tn-common/pimifpol-pimv6_policy"/>
            </pimIPV6IfP>
        </l3extLIfP>
    </l3extLNodeP>
  </l3extOut>
</fvTenant>
```

Layer 3 IPv6 multicast with PIM6 is enabled.

Configuring Multicast Filtering Using the REST API

You will be configuring multicast filtering at the bridge domain level. Use the procedures in this topic to configure either source filtering or receiver filtering, or both, at the bridge domain level.

Before you begin

- The bridge domain where you will be configuring multicast filtering is already created.
- The bridge domain is a PIM-enabled bridge domain.

- Layer 3 multicast is enabled at the VRF level.

Procedure

Step 1 If you want to enable multicast *source* filtering on the bridge domain, send a post with XML such as the following example:

Example:

```
<fvBD dn="uni/tn-filter/BD-BD1520" ipv6McastAllow="no" mcastAllow="yes">
 <pimBDP annotation="" descr="" name="" nameAlias="" ownerKey="" ownerTag="">
  <pimBDFilterPol annotation="" descr="" name="" nameAlias="">
   <pimBDSrcFilterPol annotation="" descr="" name="" nameAlias="">
    <rtdmcRsFilterToRtMapPol tDn="uni/tn-filter/rtmap-test_src_filter"/>
   </pimBDSrcFilterPol>
  </pimBDFilterPol>
 </pimBDP>
</fvBD>
```

Step 2 If you want to enable multicast *receiver* filtering on the bridge domain, send a post with XML such as the following example:

Example:

```
<fvBD dn="uni/tn-filter/BD-BD1520" ipv6McastAllow="no" mcastAllow="yes">
 <pimBDP annotation="" descr="" name="" nameAlias="" ownerKey="" ownerTag="">
  <pimBDFilterPol annotation="" descr="" name="" nameAlias="">
   <pimBDDestFilterPol annotation="" descr="" name="" nameAlias="">
    <rtdmcRsFilterToRtMapPol tDn="uni/tn-filter/rtmap-Recv_filter"/>
   </pimBDDestFilterPol>
  </pimBDFilterPol>
 </pimBDP>
</fvBD>
```

Note
You can also enable both source and receiver filtering on the same bridge domain by sending a post with XML such as the following example:

```
<fvBD dn="uni/tn-filter/BD-BD1520" ipv6McastAllow="no" mcastAllow="yes">
 <pimBDP annotation="" descr="" name="" nameAlias="" ownerKey="" ownerTag="">
  <pimBDFilterPol annotation="" descr="" name="" nameAlias="">
   <pimBDSrcFilterPol annotation="" descr="" name="" nameAlias="">
    <rtdmcRsFilterToRtMapPol tDn="uni/tn-filter/rtmap-test_src_filter"/>
   </pimBDSrcFilterPol>
   <pimBDDestFilterPol annotation="" descr="" name="" nameAlias="">
    <rtdmcRsFilterToRtMapPol tDn="uni/tn-filter/rtmap-Recv_filter"/>
   </pimBDDestFilterPol>
  </pimBDFilterPol>
 </pimBDP>
</fvBD>
```

Configuring Multi-Pod Using REST API

Setting Up Multi-Pod Fabric Using the REST API

Procedure

Step 1 Login to Cisco APIC:

Example:

```
http://<apic-name/ip>:80/api/aaaLogin.xml

data: <aaaUser name="admin" pwd="ins3965!”/>
```

Step 2 Configure the TEP pool:

Example:

```
http://<apic-name/ip>:80/api/policymgr/mo/uni/controller.xml

<fabricSetupPol status=''>
    <fabricSetupP podId="1" tepPool="10.0.0.0/16" />
    <fabricSetupP podId="2" tepPool="10.1.0.0/16" status='' />
</fabricSetupPol>
```

Step 3 Configure the node ID policy:

Example:

```
http://<apic-name/ip>:80/api/node/mo/uni/controller.xml

<fabricNodeIdentPol>
<fabricNodeIdentP serial="SAL1819RXP4" name="ifav4-leaf1" nodeId="101" podId="1"/>
<fabricNodeIdentP serial="SAL1803L25H" name="ifav4-leaf2" nodeId="102" podId="1"/>
<fabricNodeIdentP serial="SAL1934MNY0" name="ifav4-leaf3" nodeId="103" podId="1"/>
<fabricNodeIdentP serial="SAL1934MNY3" name="ifav4-leaf4" nodeId="104" podId="1"/>
<fabricNodeIdentP serial="SAL1748H56D" name="ifav4-spine1" nodeId="201" podId="1"/>
<fabricNodeIdentP serial="SAL1938P7A6" name="ifav4-spine3" nodeId="202" podId="1"/>
<fabricNodeIdentP serial="SAL1938PHBB" name="ifav4-leaf5" nodeId="105" podId="2"/>
<fabricNodeIdentP serial="SAL1942R857" name="ifav4-leaf6" nodeId="106" podId="2"/>
<fabricNodeIdentP serial="SAL1931LA3B" name="ifav4-spine2" nodeId="203" podId="2"/>
<fabricNodeIdentP serial="FGE173400A9" name="ifav4-spine4" nodeId="204" podId="2"/>
</fabricNodeIdentPol>
```

Step 4 Configure infra L3Out and external connectivity profile:

Example:

```
http://<apic-name/ip>:80/api/node/mo/uni.xml

<polUni>

<fvTenant descr="" dn="uni/tn-infra" name="infra" ownerKey="" ownerTag="">

   <l3extOut descr="" enforceRtctrl="export" name="multipod" ownerKey="" ownerTag=""
targetDscp="unspecified" status=''>
      <ospfExtP areaId='0' areaType='regular' status=''/>
      <l3extRsEctx tnFvCtxName="overlay-1"/>
      <l3extProvLbl descr="" name="prov_mp1" ownerKey="" ownerTag="" tag="yellow-green"/>

      <l3extLNodeP name="bSpine">
```

```
        <l3extRsNodeL3OutAtt rtrId="201.201.201.201" rtrIdLoopBack="no" tDn="topology/pod-1/node-201">

            <l3extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
            <l3extLoopBackIfP addr="201::201/128" descr="" name=""/>
            <l3extLoopBackIfP addr="201.201.201.201/32" descr="" name=""/>
         </l3extRsNodeL3OutAtt>

        <l3extRsNodeL3OutAtt rtrId="202.202.202.202" rtrIdLoopBack="no" tDn="topology/pod-1/node-202">

            <l3extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
            <l3extLoopBackIfP addr="202::202/128" descr="" name=""/>
            <l3extLoopBackIfP addr="202.202.202.202/32" descr="" name=""/>
         </l3extRsNodeL3OutAtt>

        <l3extRsNodeL3OutAtt rtrId="203.203.203.203" rtrIdLoopBack="no" tDn="topology/pod-2/node-203">

            <l3extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
            <l3extLoopBackIfP addr="203::203/128" descr="" name=""/>
            <l3extLoopBackIfP addr="203.203.203.203/32" descr="" name=""/>
         </l3extRsNodeL3OutAtt>

        <l3extRsNodeL3OutAtt rtrId="204.204.204.204" rtrIdLoopBack="no" tDn="topology/pod-2/node-204">

            <l3extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
            <l3extLoopBackIfP addr="204::204/128" descr="" name=""/>
            <l3extLoopBackIfP addr="204.204.204.204/32" descr="" name=""/>
         </l3extRsNodeL3OutAtt>

         <l3extLIfP name='portIf'>
            <l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-1/paths-201/pathep-[eth1/1]"
encap='vlan-4'  ifInstT='sub-interface' addr="201.1.1.1/30" />
            <l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-1/paths-201/pathep-[eth1/2]"
encap='vlan-4'  ifInstT='sub-interface' addr="201.2.1.1/30" />
            <l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-1/paths-202/pathep-[eth1/2]"
encap='vlan-4'  ifInstT='sub-interface' addr="202.1.1.1/30" />
            <l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-2/paths-203/pathep-[eth1/1]"
encap='vlan-4'  ifInstT='sub-interface' addr="203.1.1.1/30" />
            <l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-2/paths-203/pathep-[eth1/2]"
encap='vlan-4'  ifInstT='sub-interface' addr="203.2.1.1/30" />
            <l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-2/paths-204/pathep-[eth4/31]"
encap='vlan-4'  ifInstT='sub-interface' addr="204.1.1.1/30" />

           <ospfIfP>
               <ospfRsIfPol tnOspfIfPolName='ospfIfPol'/>
           </ospfIfP>

         </l3extLIfP>
      </l3extLNodeP>

      <l3extInstP descr="" matchT="AtleastOne" name="instp1" prio="unspecified"
targetDscp="unspecified">
          <fvRsCustQosPol tnQosCustomPolName=""/>
      </l3extInstP>
   </l3extOut>

  <fvFabricExtConnP descr="" id="1" name="Fabric_Ext_Conn_Pol1" rt="extended:as2-nn4:5:16" status=''>

      <fvPodConnP descr="" id="1" name="">
         <fvIp addr="100.11.1.1/32"/>
      </fvPodConnP>
      <fvPodConnP descr="" id="2" name="">
         <fvIp addr="200.11.1.1/32"/>
      </fvPodConnP>
      <fvPeeringP descr="" name="" ownerKey="" ownerTag="" type="automatic_with_full_mesh"/>
```

```
      <l3extFabricExtRoutingP descr="" name="ext_routing_prof_1" ownerKey="" ownerTag="">
         <l3extSubnet aggregate="" descr="" ip="100.0.0.0/8" name="" scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="200.0.0.0/8" name="" scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="201.1.0.0/16" name="" scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="201.2.0.0/16" name="" scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="202.1.0.0/16" name="" scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="203.1.0.0/16" name="" scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="203.2.0.0/16" name="" scope="import-security"/>
         <l3extSubnet aggregate="" descr="" ip="204.1.0.0/16" name="" scope="import-security"/>
      </l3extFabricExtRoutingP>
   </fvFabricExtConnP>
</fvTenant>
</polUni>
```

Configuring Remote Leaf Switches Using REST API

Configure Remote Leaf Switches Using the REST API

To enable Cisco APIC to discover and connect the IPN router and remote leaf switches, perform the steps in this topic.

This example assumes that the remote leaf switches are connected to a pod in a multipod topology. It includes two L3Outs configured in the infra tenant, with VRF overlay-1:

- One is configured on VLAN-4, that is required for both the remote leaf switches and the spine switch connected to the WAN router.
- One is the multipod-internal L3Out configured on VLAN-5, that is required for the multipod and Remote Leaf features, when they are deployed together.

Procedure

Step 1 To define the TEP pool for two remote leaf switches to be connected to a pod, send a post with XML such as the following example:

Example:

```
<fabricSetupPol>
      <fabricSetupP tepPool="10.0.0.0/16" podId="1" >
           <fabricExtSetupP tepPool="30.0.128.0/20" extPoolId="1"/>
     </fabricSetupP>
     <fabricSetupP tepPool="10.1.0.0/16" podId="2" >
          <fabricExtSetupP tepPool="30.1.128.0/20" extPoolId="1"/>
     </fabricSetupP>
</fabricSetupPol>
```

Step 2 To define the node identity policy, send a post with XML, such as the following example:

Example:

```
<fabricNodeIdentPol>
     <fabricNodeIdentP serial="SAL17267Z7W" name="leaf1" nodeId="101" podId="1"
extPoolId="1" nodeType="remote-leaf-wan"/>
     <fabricNodeIdentP serial="SAL17267Z7X" name="leaf2" nodeId="102" podId="1"
extPoolId="1" nodeType="remote-leaf-wan"/>
```

```
     <fabricNodeIdentP serial="SAL17267Z7Y" name="leaf3" nodeId="201" podId="1"
extPoolId="1" nodeType="remote-leaf-wan"/>
     <fabricNodeIdentP serial="SAL17267Z7Z" name="leaf4" nodeId="201" podId="1"
extPoolId="1" nodeType="remote-leaf-wan"/>
</fabricNodeIdentPol>
```

Step 3 To configure the Fabric External Connection Profile, send a post with XML such as the following example:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<imdata totalCount="1">
     <fvFabricExtConnP dn="uni/tn-infra/fabricExtConnP-1" id="1" name="Fabric_Ext_Conn_Pol1"
rt="extended:as2-nn4:5:16" siteId="0">
           <l3extFabricExtRoutingP name="test">
                <l3extSubnet ip="150.1.0.0/16" scope="import-security"/>
           </l3extFabricExtRoutingP>
          <l3extFabricExtRoutingP name="ext_routing_prof_1">
               <l3extSubnet ip="204.1.0.0/16" scope="import-security"/>
               <l3extSubnet ip="209.2.0.0/16" scope="import-security"/>
               <l3extSubnet ip="202.1.0.0/16" scope="import-security"/>
               <l3extSubnet ip="207.1.0.0/16" scope="import-security"/>
               <l3extSubnet ip="200.0.0.0/8" scope="import-security"/>
               <l3extSubnet ip="201.2.0.0/16" scope="import-security"/>
               <l3extSubnet ip="210.2.0.0/16" scope="import-security"/>
               <l3extSubnet ip="209.1.0.0/16" scope="import-security"/>
               <l3extSubnet ip="203.2.0.0/16" scope="import-security"/>
               <l3extSubnet ip="208.1.0.0/16" scope="import-security"/>
               <l3extSubnet ip="207.2.0.0/16" scope="import-security"/>
               <l3extSubnet ip="100.0.0.0/8" scope="import-security"/>
               <l3extSubnet ip="201.1.0.0/16" scope="import-security"/>
               <l3extSubnet ip="210.1.0.0/16" scope="import-security"/>
               <l3extSubnet ip="203.1.0.0/16" scope="import-security"/>
               <l3extSubnet ip="208.2.0.0/16" scope="import-security"/>
           </l3extFabricExtRoutingP>
           <fvPodConnP id="1">
                 <fvIp addr="100.11.1.1/32"/>
          </fvPodConnP>
          <fvPodConnP id="2">
                <fvIp addr="200.11.1.1/32"/>
          </fvPodConnP>
          <fvPeeringP type="automatic_with_full_mesh"/>
     </fvFabricExtConnP>
</imdata>
```

Step 4 To configure an L3Out on VLAN-4, that is required for both the remote leaf switches and the spine switch connected to the WAN router, enter XML such as the following example:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<polUni>
<fvTenant name="infra">
  <l3extOut name="rleaf-wan-test">
    <ospfExtP areaId="0.0.0.5"/>
    <bgpExtP/>
    <l3extRsEctx tnFvCtxName="overlay-1"/>
    <l3extRsL3DomAtt tDn="uni/l3dom-l3extDom1"/>
    <l3extProvLbl descr="" name="prov_mp1" ownerKey="" ownerTag="" tag="yellow-green"/>
    <l3extLNodeP name="rleaf-101">
      <l3extRsNodeL3OutAtt rtrId="202.202.202.202" tDn="topology/pod-1/node-101">
      </l3extRsNodeL3OutAtt>
      <l3extLIfP name="portIf">
        <l3extRsPathL3OutAtt ifInstT="sub-interface" tDn="topology/pod-1/paths-101/pathep-[eth1/49]"
 addr="202.1.1.2/30" mac="AA:11:22:33:44:66" encap='vlan-4'/>
```

Performing Tasks Using REST API

```
        <ospfIfP>
          <ospfRsIfPol tnOspfIfPolName='ospfIfPol'/>
        </ospfIfP>
      </l3extLIfP>
    </l3extLNodeP>
    <l3extLNodeP name="rlSpine-201">
      <l3extRsNodeL3OutAtt rtrId="201.201.201.201" rtrIdLoopBack="no" tDn="topology/pod-1/node-201">

        <!--
        <l3extLoopBackIfP addr="201::201/128" descr="" name=""/>
        <l3extLoopBackIfP addr="201.201.201.201/32" descr="" name=""/>
        -->
        <l3extLoopBackIfP addr="::" />
      </l3extRsNodeL3OutAtt>
      <l3extLIfP name="portIf">
        <l3extRsPathL3OutAtt ifInstT="sub-interface" tDn="topology/pod-1/paths-201/pathep-[eth8/36]"
 addr="201.1.1.1/30" mac="00:11:22:33:77:55" encap='vlan-4'/>
        <ospfIfP>
          <ospfRsIfPol tnOspfIfPolName='ospfIfPol'/>
        </ospfIfP>
      </l3extLIfP>
    </l3extLNodeP>
   <l3extInstP descr="" matchT="AtleastOne" name="instp1" prio="unspecified" targetDscp="unspecified">

      <fvRsCustQosPol tnQosCustomPolName=""/>
    </l3extInstP>
  </l3extOut>
  <ospfIfPol name="ospfIfPol" nwT="bcast"/>
</fvTenant>
</polUni>
```

Step 5 To configure the multipod L3Out on VLAN-5, that is required for both multipod and the remote leaf topology, send a post such as the following example:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<polUni>

  <fvTenant  name="infra" >
    <l3extOut name="ipn-multipodInternal">
     <ospfExtP areaCtrl="inherit-ipsec,redistribute,summary" areaId="0.0.0.5" multipodInternal="yes"
 />
      <l3extRsEctx tnFvCtxName="overlay-1" />
      <l3extLNodeP name="bLeaf">
       <l3extRsNodeL3OutAtt rtrId="202.202.202.202" rtrIdLoopBack="no" tDn="topology/pod-2/node-202">

            <l3extLoopBackIfP addr="202.202.202.212"/>
       </l3extRsNodeL3OutAtt>
       <l3extRsNodeL3OutAtt rtrId="102.102.102.102" rtrIdLoopBack="no" tDn="topology/pod-1/node-102">

            <l3extLoopBackIfP addr="102.102.102.112"/>
       </l3extRsNodeL3OutAtt>
        <l3extLIfP name="portIf">
          <ospfIfP authKeyId="1" authType="none">
            <ospfRsIfPol tnOspfIfPolName="ospfIfPol" />
          </ospfIfP>
          <l3extRsPathL3OutAtt addr="10.0.254.233/30" encap="vlan-5" ifInstT="sub-interface"
tDn="topology/pod-2/paths-202/pathep-[eth5/2]"/>
          <l3extRsPathL3OutAtt addr="10.0.255.229/30" encap="vlan-5" ifInstT="sub-interface"
tDn="topology/pod-1/paths-102/pathep-[eth5/2]"/>
        </l3extLIfP>
      </l3extLNodeP>
      <l3extInstP matchT="AtleastOne" name="ipnInstP" />
    </l3extOut>
```

```
  </fvTenant>
</polUni>
```

Configuring SR-MPLS Handoff Using REST API

Configuring an SR-MPLS Infra L3Out Using the REST API

- The SR-MPLS infra L3Out is configured on the border leaf switch, which is used to set up the underlay BGP-LU and overlay MP-BGP EVPN sessions that are needed for the SR-MPLS handoff.
- An SR-MPLS infra L3Out will be scoped to a pod or a remote leaf switch site.
- Border leaf switches or remote leaf switches in one SR-MPLS infra L3Out can connect to one or more provider edge (PE) routers in one or more routing domains.
- A pod or remote leaf switch site can have one or more SR-MPLS infra L3Outs.
- Each SR-MPLS infra L3Out should have a unique provider label and one provider label only. Each SR-MPLS infra L3Out is identified by the provider label.

You will configure the following pieces when configuring the SR-MPLS infra L3Out:

- **Nodes**
  - Only leaf switches are allowed to be configured as nodes in the SR-MPLS infra L3Out (border leaf switches and remote leaf switches).
  - Each SR-MPLS infra L3Out can have border leaf switches from one pod or remote leaf switch from the same site.
  - Each border leaf switch or remote leaf switch can be configured in multiple SR-MPLS infra L3Outs if it connects to multiple SR-MPLS domains.
  - You will also configure the loopback interface underneath the node, and a node SID policy underneath the loopback interface.

- **Interfaces**
  - Supported types of interfaces are:
    - Routed interface or sub-interface
    - Routed port channel or port channel sub-interface

    For sub-interfaces, any VLAN tag is supported.
  - You will also configure the underlay BGP peer policy underneath the interfaces area in the SR-MPLS infra L3Out.

- **QoS rules**
  - You can configure the MPLS ingress rule and MPLS egress rule through the MPLS QoS policy in the SR-MPLS infra L3Out.

Performing Tasks Using REST API

- If you do not create an MPLS QoS policy, any ingressing MPLS traffic is assigned the default QoS level.

You will also configure the underlay and overlay through the SR-MPLS infra L3Out:

- **Underlay**: BGP peer IP (BGP LU peer) configuration as part of the interface configuration.
- **Overlay**: MP-BGP EVPN remote IPv4 address (MP-BGP EVPN peer) configuration as part of the logical node profile configuration.

Before you begin

- Review the SR-MPLS guidelines and limitations provided in Guidelines and Limitations for SR-MPLS, especially the guidelines and limitations provided in Guidelines and Limitations for the SR-MPLS Infra L3Out.
- (Optional) If necessary, configure an MPLS custom QoS policy using the procedures provided in Creating SR-MPLS Custom QoS Policy Using REST API, on page 24.

Procedure

Post with information similar to the following:

```
<polUni>
<fvTenant name="infra">
    <mplsIfPol name="default"/>
    <mplsLabelPol name="default" >
       <mplsSrgbLabelPol minSrgbLabel="16000" maxSrgbLabel="17000" localId="1" status=""/>
    </mplsLabelPol>

<l3extOut name="mplsOut" status="" descr="bl" mplsEnabled="yes">
    <l3extRsEctx tnFvCtxName="overlay-1"/>
    <l3extProvLbl name="mpls" />
    <mplsExtP status="" >
        <mplsRsLabelPol tDn="uni/tn-infra/mplslabelpol-default"/>
    </mplsExtP>
    <l3extLNodeP name="mplsLNP" status="">
        <l3extRsNodeL3OutAtt rtrId="100.1.1.1" rtrIdLoopBack="no" tDn="topology/pod-1/node-101"
status="">
            <l3extLoopBackIfP addr="10.10.10.11" status="">
                <mplsNodeSidP sidoffset="2" loopbackAddr="10.1.3.11" status=""/>
            </l3extLoopBackIfP>
        </l3extRsNodeL3OutAtt>

        <l3extLIfP name="mplsLIfP1" status="">
            <mplsIfP status="">
                <mplsRsIfPol tnMplsIfPolName="default" />
            </mplsIfP>
            <l3extRsPathL3OutAtt addr="34.1.2.3/30" ifInstT="l3-port"
tDn="topology/pod-1/paths-101/pathep-[eth1/8]">
                <bgpPeerP addr="9.9.9.7" addrTCtrl="af-ucast,af-label-ucast" ctrl="send-ext-com"
ttl="1" status="">
                    <bgpAsP asn="100"/>
                </bgpPeerP>
            </l3extRsPathL3OutAtt>
        </l3extLIfP>
```

```
       <bgpInfraPeerP addr="20.1.1.1" ctrl="send-com,send-ext-com" peerT="sr-mpls" ttl="3" status=""
 >
           <bgpAsP asn="100"/>
        </bgpInfraPeerP>
    </l3extLNodeP>


    <l3extInstP name="mplsInstP">
        <l3extSubnet aggregate="" descr="" ip="11.11.11.0/24" name="" scope="import-security"/>
    </l3extInstP>
    <bgpExtP/>
    <l3extRsL3DomAtt tDn="uni/l3dom-l3extDom1" />
</l3extOut>

</fvTenant>
</polUni>
```

Configuring an SR-MPLS VRF L3Out Using the REST API

Using the procedures in this section, you will configure a SR-MPLS VRF L3Out, which will be used to forward traffic from the SR-MPLS infra L3Out that you configured in the previous set of procedures.

- User tenant VRFs are mapped to the SR-MPLS infra L3Outs to advertise tenant bridge domain subnets to the DC-PE routers and import the MPLS VPN routes received from the DC-PE.
- You must specify routing and security policies in the SR-MPLS VRF L3Out for each VRF. These policies point to one or more SR-MPLS infra L3Outs.
- One SR-MPLS VRF L3Out is supported for each VRF.
- You can configure multiple consumer labels in one SR-MPLS VRF L3Out, with each consumer label identifying one SR-MPLS infra L3Out. A consumer label identifies the entry and exit point for traffic to and from one SR-MPLS VRF L3Out, which is a particular MPLS domain for a particular pod or remote leaf switch.

Before you begin

- Review the SR-MPLS guidelines and limitations provided in Guidelines and Limitations for SR-MPLS, especially the guidelines and limitations provided in Guidelines and Limitations for the SR-MPLS VRF L3Out.
- Configure an SR-MPLS infra L3Out using the procedures provided in Configuring an SR-MPLS Infra L3Out Using the REST API, on page 21.

Procedure

Post with information similar to the following:

```
<polUni>
<fvTenant name="t1">
  <fvCtx name="v1">
    <!-- specify bgp evpn route-target -->
    <bgpRtTargetP af="ipv4-ucast">
```

```
      <bgpRtTarget rt="route-target:as4-nn2:100:1259" type="import"/>
      <bgpRtTarget rt="route-target:as4-nn2:100:1259" type="export"/>
    </bgpRtTargetP>
  </fvCtx>

  <!-- MPLS L3out -->
  <l3extOut name="out1" mplsEnabled="yes">
    <l3extRsEctx tnFvCtxName="v1" />

    <!-- MPLS consumer label -->
    <l3extConsLbl name="mpls1">
      <!-- route profile association -->
      <l3extRsLblToProfile tDn="uni/tn-t1/prof-rp1" direction="export" />
      <!-- InstP association -->
      <l3extRsLblToInstP tDn="uni/tn-t1/out-out1/instP-epgMpls1" />
    </l3extConsLbl>

    <!-- External-EPG -->
    <l3extInstP name="epgMpls1">
      <fvRsProv tnVzBrCPName="cp1"/>
      <l3extSubnet ip="55.1.1.1/28"/>
    </l3extInstP>
    <bgpExtP/>
  </l3extOut>

  <!-- route control profile -->
  <rtctrlProfile descr="" name="rp1" type="global" status="">
    <rtctrlCtxP action="permit" descr="" name="ctx1" order="0">
      <rtctrlRsCtxPToSubjP status="" tnRtctrlSubjPName="subj1"/>
    </rtctrlCtxP>
  </rtctrlProfile>
  <rtctrlSubjP descr="" name="subj1" status="" >
    <rtctrlMatchRtDest ip="101.1.1.1/32"/>
    <rtctrlMatchRtDest ip="102.1.1.0/24" aggregate="yes"/>
  </rtctrlSubjP>

  <!-- Filter and Contract (global) -->
  <vzBrCP name="cp1" scope="global">
    <vzSubj name="allow-all">
      <vzRsSubjFiltAtt action="permit" tnVzFilterName="default" />
    </vzSubj>
  </vzBrCP>
</fvTenant>
</polUni>
```

Creating SR-MPLS Custom QoS Policy Using REST API

SR-MPLS Custom QoS policy defines the priority of the packets coming from an SR-MPLS network while they are inside the ACI fabric based on the incoming MPLS EXP values defined in the MPLS QoS ingress policy. It also marks the CoS and MPLS EXP values of the packets leaving the ACI fabric through an MPLS interface based on IPv4 DSCP values defined in MPLS QoS egress policy.

If no custom ingress policy is defined, the default QoS Level (`Level3`) is assigned to packets inside the fabric. If no custom egress policy is defined, the default EXP value of `0` will be marked on packets leaving the fabric.

Procedure

Step 1 Create SR-MPLS QoS policy.

In the following POST:

- Replace *customqos1* with the name of the SR-MPLS QoS policy you want to create.
- For the `qosMplsIngressRule`:
  - Replace `from="2" to="3"` with the EXP range you want the policy to match.
  - Replace `prio="level5"` with the ACI QoS Level for the packet while it's inside the ACI fabric.
  - Replace `target="CS5"` with the DSCP value you want to set on the packet when it's matched.
  - Replace `targetCos="4"` with the CoS value you want to set on the packet when it's matched.
- For the `qosMplsEgressRule`:
  - Replace `from="CS2" to="CS4"` with the DSCP range you want the policy to match.
  - Replace `targetExp="5"` with the EXP value you want to set on the packet when it's leaving the fabric.
  - Replace `targetCos="3"` with the CoS value you want to set on the packet when it's leaving the fabric.

```
<polUni>
  <fvTenant name="infra">
    <qosMplsCustomPol descr="" dn="uni/tn-infra/qosmplscustom-customqos1" name="customqos1"  status=""
 >
        <qosMplsIngressRule from="2" to="3" prio="level5" target="CS5" targetCos="4" status="" />
        <qosMplsEgressRule from="CS2" to="CS4" targetExp="5" targetCos="3" status=""/>
    </qosMplsCustomPol>
  </fvTenant>
</polUni>
```

Step 2 Applying SR-MPLS QoS policy.

In the following POST, replace *customqos1* with the name of the SR-MPLS QoS policy you created in the previous step.

```
<polUni>
    <fvTenant name="infra">
        <l3extOut name="mplsOut" status="" descr="bl">
            <l3extLNodeP name="mplsLNP" status="">
                <l3extRsLNodePMplsCustQosPol  tDn="uni/tn-infra/qosmplscustom-customqos1"/>
            </l3extLNodeP>
        </l3extOut>
    </fvTenant>
</polUni>
```

Part II: External Routing (L3Out) Configuration

Routed Connectivity to External Networks

Configuring an MP-BGP Route Reflector Using REST API

Configuring an MP-BGP Route Reflector Using the REST API

Procedure

Step 1 Mark the spine switches as route reflectors.

Example:

```
POST https://apic-ip-address/api/policymgr/mo/uni/fabric.xml

<bgpInstPol name="default">
  <bgpAsP asn="1" />
  <bgpRRP>
     <bgpRRNodePEp id="<spine_id1>"/>
     <bgpRRNodePEp id="<spine_id2>"/>
  </bgpRRP>
</bgpInstPol>
```

Step 2 Set up the pod selector using the following post.

Example:

For the FuncP setup—

```
POST https://apic-ip-address/api/policymgr/mo/uni.xml

<fabricFuncP>
  <fabricPodPGrp name="bgpRRPodGrp">
    <fabricRsPodPGrpBGPRRP tnBgpInstPolName="default" />
  </fabricPodPGrp>
</fabricFuncP>
```

Example:

For the PodP setup—

```
POST https://apic-ip-address/api/policymgr/mo/uni.xml

<fabricPodP name="default">
  <fabricPodS name="default" type="ALL">
    <fabricRsPodPGrp tDn="uni/fabric/funcprof/podpgrp-bgpRRPodGrp"/>
  </fabricPodS>
</fabricPodP>
```

Configuring the BGP Domain-Path Feature for Loop Prevention Using the REST API

Before you begin

Become familiar with the BGP Domain-Path feature using the information provided in About the BGP Domain-Path Feature for Loop Prevention.

Procedure

Step 1 If you want to use the BGP Domain-Path feature for loop prevention, set the global `DomainIdBase`.

```
<polUni>
  <fabricInst>
   <bgpInstPol name="default">
    <bgpDomainIdBase  domainIdBase="12346" />
   </bgpInstPol>
  </fabricInst>
 </polUni>
```

Step 2 Enable `send-domain-path` in the appropriate L3Out.

```
<bgpPeerP addr="22.22.3.5" addrTCtrl="af-ucast" allowedSelfAsCnt="3" ttl="2"
 ctrlExt="send-domain-path" ctrl="send-ext-com">
</bgpPeerP>
```

Node and Interface for L3Out

Configuring Layer 3 Routed and Sub-Interface Port Channels Using REST API

Configuring a Layer 3 Routed Port Channel Using the REST API

Before you begin

- The ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.
- An APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.
- The target leaf switches are registered in the ACI fabric and available.
- Port channels are configured when port channels are used for L3Out interfaces.

Note In the following REST API example, long single lines of text are broken up with the \ character to improve readability.

Procedure

To configure a Layer 3 route to the port channels that you created previously using the REST API, send a post with XML such as the following:

Example:

```
<polUni>
<fvTenant name=pep9>
    <l3extOut descr="" dn="uni/tn-pep9/out-routAccounting" enforceRtctrl="export" \
     name="routAccounting" nameAlias="" ownerKey="" ownerTag="" \
     targetDscp="unspecified">
        <l3extRsL3DomAtt tDn="uni/l3dom-Dom1"/>
        <l3extRsEctx tnFvCtxName="ctx9"/>
        <l3extLNodeP configIssues="" descr="" name="node101" nameAlias="" ownerKey="" \
         ownerTag="" tag="yellow-green" targetDscp="unspecified">
            <l3extRsNodeL3OutAtt rtrId="10.1.0.101" rtrIdLoopBack="yes" \
             tDn="topology/pod-1/node-101">
                <l3extInfraNodeP descr="" fabricExtCtrlPeering="no" \
                 fabricExtIntersiteCtrlPeering="no" name="" nameAlias="" spineRole=""/>
            </l3extRsNodeL3OutAtt>
            <l3extLIfP descr="" name="lifp17" nameAlias="" ownerKey="" ownerTag="" \
             tag="yellow-green">
                <ospfIfP authKeyId="1" authType="none" descr="" name="" nameAlias="">
                    <ospfRsIfPol tnOspfIfPolName=""/>
                </ospfIfP>
                <l3extRsPathL3OutAtt addr="10.1.5.3/24" autostate="disabled" descr="" \
                 encap="unknown" encapScope="local" ifInstT="l3-port" llAddr="::" \
                 mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit" \
                 tDn="topology/pod-1/paths-101/pathep-[po17_PolGrp]" \
                 targetDscp="unspecified"/>
                <l3extRsNdIfPol tnNdIfPolName=""/>
                <l3extRsIngressQosDppPol tnQosDppPolName=""/>
                <l3extRsEgressQosDppPol tnQosDppPolName=""/>
            </l3extLIfP>
        </l3extLNodeP>
        <l3extInstP descr="" floodOnEncap="disabled" matchT="AtleastOne" \
         name="accountingInst" nameAlias="" prefGrMemb="exclude" prio="unspecified" \
         targetDscp="unspecified">
            <fvRsProv matchT="AtleastOne" prio="unspecified" tnVzBrCPName="webCtrct"/>
            <l3extSubnet aggregate="export-rtctrl,import-rtctrl" descr="" ip="0.0.0.0/0" \
             name="" nameAlias="" scope="export-rtctrl,import-rtctrl,import-security"/>
            <l3extSubnet aggregate="export-rtctrl,import-rtctrl" descr="" ip="::/0" \
             name="" nameAlias="" scope="export-rtctrl,import-rtctrl,import-security"/>
            <fvRsCustQosPol tnQosCustomPolName=""/>
        </l3extInstP>
        <l3extConsLbl descr="" name="golf" nameAlias="" owner="infra" ownerKey="" \
         ownerTag="" tag="yellow-green"/>
    </l3extOut>
</fvTenant>
</polUni>
```

Configuring a Layer 3 Sub-Interface Port Channel Using the REST API

Before you begin

- The ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.

- An APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.
- The target leaf switches are registered in the ACI fabric and available.
- Port channels are configured using the procedures in "Configuring Port Channels Using the REST API".

Note In the following REST API example, long single lines of text are broken up with the \ character to improve readability.

Procedure

To configure a Layer 3 sub-interface route to the port channels that you created previously using the REST API, send a post with XML such as the following:

Example:

```
<polUni>
<fvTenant name=pep9>
    <l3extOut descr="" dn="uni/tn-pep9/out-routAccounting" enforceRtctrl="export" \
     name="routAccounting" nameAlias="" ownerKey="" ownerTag="" targetDscp="unspecified">
        <l3extRsL3DomAtt tDn="uni/l3dom-Dom1"/>
        <l3extRsEctx tnFvCtxName="ctx9"/>
        <l3extLNodeP configIssues="" descr="" name="node101" nameAlias="" ownerKey="" \
         ownerTag="" tag="yellow-green" targetDscp="unspecified">
            <l3extRsNodeL3OutAtt rtrId="10.1.0.101" rtrIdLoopBack="yes" \
             tDn="topology/pod-1/node-101">
                <l3extInfraNodeP descr="" fabricExtCtrlPeering="no" \
                 fabricExtIntersiteCtrlPeering="no" name="" nameAlias="" spineRole=""/>
            </l3extRsNodeL3OutAtt>
            <l3extLIfP descr="" name="lifp27" nameAlias="" ownerKey="" ownerTag="" \
             tag="yellow-green">
                <ospfIfP authKeyId="1" authType="none" descr="" name="" nameAlias="">
                    <ospfRsIfPol tnOspfIfPolName=""/>
                </ospfIfP>
                <l3extRsPathL3OutAtt addr="11.1.5.3/24" autostate="disabled" descr="" \
                 encap="vlan-2001" encapScope="local" ifInstT="sub-interface" \
                 llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit" \
                 tDn="topology/pod-1/paths-101/pathep-[po27_PolGrp]" \
                 targetDscp="unspecified"/>
                <l3extRsNdIfPol tnNdIfPolName=""/>
                <l3extRsIngressQosDppPol tnQosDppPolName=""/>
                <l3extRsEgressQosDppPol tnQosDppPolName=""/>
            </l3extLIfP>
        </l3extLNodeP>
        <l3extInstP descr="" floodOnEncap="disabled" matchT="AtleastOne" \
         name="accountingInst" nameAlias="" prefGrMemb="exclude" prio="unspecified" \
         targetDscp="unspecified">
            <fvRsProv matchT="AtleastOne" prio="unspecified" tnVzBrCPName="webCtrct"/>
            <l3extSubnet aggregate="export-rtctrl,import-rtctrl" descr="" ip="0.0.0.0/0" \
             name="" nameAlias="" scope="export-rtctrl,import-rtctrl,import-security"/>
            <l3extSubnet aggregate="export-rtctrl,import-rtctrl" descr="" ip="::/0" \
             name="" nameAlias="" scope="export-rtctrl,import-rtctrl,import-security"/>
            <fvRsCustQosPol tnQosCustomPolName=""/>
        </l3extInstP>
        <l3extConsLbl descr="" name="golf" nameAlias="" owner="infra" ownerKey="" \
         ownerTag="" tag="yellow-green"/>
```

```
    </l3extOut>
</fvTenant>
</polUni>
```

Configuring a Switch Virtual Interface Using REST API

Configuring SVI Interface Encapsulation Scope Using the REST API

Before you begin

The interface selector is configured.

Procedure

Configure the SVI interface encapsulation scope.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- /api/node/mo/.xml -->
<polUni>
 <fvTenant name="coke">
  <l3extOut descr="" dn="uni/tn-coke/out-l3out1" enforceRtctrl="export" name="l3out1" nameAlias=""
ownerKey="" ownerTag="" targetDscp="unspecified">
   <l3extRsL3DomAtt tDn="uni/l3dom-Dom1"/>
   <l3extRsEctx tnFvCtxName="vrf0"/>
   <l3extLNodeP configIssues="" descr="" name="__ui_node_101" nameAlias="" ownerKey="" ownerTag=""
tag="yellow-green" targetDscp="unspecified">
    <l3extRsNodeL3OutAtt rtrId="1.1.1.1" rtrIdLoopBack="no" tDn="topology/pod-1/node-101"/>
    <l3extLIfP descr="" name="int1_11" nameAlias="" ownerKey="" ownerTag="" tag="yellow-green">
     <l3extRsPathL3OutAtt addr="1.2.3.4/24" descr="" encap="vlan-2001" encapScope="ctx"
ifInstT="ext-svi" llAddr="0.0.0.0" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
tDn="topology/pod-1/paths-101/pathep-[eth1/5]" targetDscp="unspecified"/>
     <l3extRsNdIfPol tnNdIfPolName=""/>
     <l3extRsIngressQosDppPol tnQosDppPolName=""/>
     <l3extRsEgressQosDppPol tnQosDppPolName=""/>
    </l3extLIfP>
   </l3extLNodeP>
   <l3extInstP descr="" matchT="AtleastOne" name="epg1" nameAlias="" prefGrMemb="exclude"
prio="unspecified" targetDscp="unspecified">
    <l3extSubnet aggregate="" descr="" ip="101.10.10.1/24" name="" nameAlias=""
scope="import-security"/>
    <fvRsCustQosPol tnQosCustomPolName=""/>
   </l3extInstP>
  </l3extOut>
 </fvTenant>
</polUni>
```

Configuring SVI Auto State Using the REST API

Before you begin

- The tenant and VRF configured.

- A Layer 3 Out is configured and a logical node profile and a logical interface profile under the Layer 3 Out is configured.

Procedure

Enable the SVI auto state value.

Example:

```
<fvTenant name="t1" >
   <l3extOut name="out1">
      <l3extLNodeP name="__ui_node_101" >
         <l3extLIfP descr="" name="__ui_eth1_10_vlan_99_af_ipv4" >
            <l3extRsPathL3OutAtt addr="19.1.1.1/24" autostate="enabled" descr="" encap="vlan-100"
encapScope="local" ifInstT="ext-svi" llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
 tDn="topology/pod-1/paths-101/pathep-[eth1/10]" targetDscp="unspecified" />
         </l3extLIfP>
      </l3extLNodeP>
   </l3extOut>
</fvTenant>
```

To disable the autostate, you must change the value to disabled in the above example. For example, `autostate="disabled"`.

Configuring Routing Protocols Using REST API

Configuring BGP External Routed Networks with BFD Support Using REST API

Configuring BGP External Routed Network Using the REST API

Before you begin

The tenant where you configure the BGP external routed network is already created.

The following shows how to configure the BGP external routed network using the REST API:

For Example:

Procedure

Example:

```
<l3extOut descr="" dn="uni/tn-t1/out-l3out-bgp" enforceRtctrl="export" name="l3out-bgp" ownerKey=""
 ownerTag="" targetDscp="unspecified">
 <l3extRsEctx tnFvCtxName="ctx3"/>
 <l3extLNodeP configIssues="" descr="" name="l3extLNodeP_1" ownerKey="" ownerTag="" tag="yellow-green"
 targetDscp="unspecified">
  <l3extRsNodeL3OutAtt rtrId="1.1.1.1" rtrIdLoopBack="no" tDn="topology/pod-1/node-101"/>
  <l3extLIfP descr="" name="l3extLIfP_2" ownerKey="" ownerTag="" tag="yellow-green">
   <l3extRsNdIfPol tnNdIfPolName=""/>
   <l3extRsIngressQosDppPol tnQosDppPolName=""/>
```

```
   <l3extRsEgressQosDppPol tnQosDppPolName=""/>
   <l3extRsPathL3OutAtt addr="3001::31:0:1:2/120" descr="" encap="vlan-3001" encapScope="local"
ifInstT="sub-interface" llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
tDn="topology/pod-1/paths-101/pathep-[eth1/8]" targetDscp="unspecified">
    <bgpPeerP addr="3001::31:0:1:0/120" allowedSelfAsCnt="3" ctrl="send-com,send-ext-com" descr=""
name="" peerCtrl="bfd" privateASctrl="remove-all,remove-exclusive,replace-as" ttl="1" weight="1000">

     <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
     <bgpAsP asn="3001" descr="" name=""/>
    </bgpPeerP>
   </l3extRsPathL3OutAtt>
  </l3extLIfP>
  <l3extLIfP descr="" name="l3extLIfP_1" ownerKey="" ownerTag="" tag="yellow-green">
   <l3extRsNdIfPol tnNdIfPolName=""/>
   <l3extRsIngressQosDppPol tnQosDppPolName=""/>
   <l3extRsEgressQosDppPol tnQosDppPolName=""/>
   <l3extRsPathL3OutAtt addr="31.0.1.2/24" descr="" encap="vlan-3001" encapScope="local"
ifInstT="sub-interface" llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
tDn="topology/pod-1/paths-101/pathep-[eth1/8]" targetDscp="unspecified">
    <bgpPeerP addr="31.0.1.0/24" allowedSelfAsCnt="3" ctrl="send-com,send-ext-com" descr="" name=""
 peerCtrl="" privateASctrl="remove-all,remove-exclusive,replace-as" ttl="1" weight="100">
     <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
     <bgpLocalAsnP asnPropagate="none" descr="" localAsn="200" name=""/>
     <bgpAsP asn="3001" descr="" name=""/>
    </bgpPeerP>
   </l3extRsPathL3OutAtt>
  </l3extLIfP>
 </l3extLNodeP>
 <l3extRsL3DomAtt tDn="uni/l3dom-l3-dom"/>
 <l3extRsDampeningPol af="ipv6-ucast" tnRtctrlProfileName="damp_rp"/>
 <l3extRsDampeningPol af="ipv4-ucast" tnRtctrlProfileName="damp_rp"/>
 <l3extInstP descr="" matchT="AtleastOne" name="l3extInstP_1" prio="unspecified"
targetDscp="unspecified">
  <l3extSubnet aggregate="" descr="" ip="130.130.130.0/24" name="" scope="import-rtctrl"></l3extSubnet>

  <l3extSubnet aggregate="" descr="" ip="130.130.131.0/24" name="" scope="import-rtctrl"/>
  <l3extSubnet aggregate="" descr="" ip="120.120.120.120/32" name=""
scope="export-rtctrl,import-security"/>
  <l3extSubnet aggregate="" descr="" ip="3001::130:130:130:100/120" name="" scope="import-rtctrl"/>
 </l3extInstP>
 <bgpExtP descr=""/>
</l3extOut>
<rtctrlProfile descr="" dn="uni/tn-t1/prof-damp_rp" name="damp_rp" ownerKey="" ownerTag=""
type="combinable">
 <rtctrlCtxP descr="" name="ipv4_rpc" order="0">
  <rtctrlScope descr="" name="">
   <rtctrlRsScopeToAttrP tnRtctrlAttrPName="act_rule"/>
  </rtctrlScope>
 </rtctrlCtxP>
</rtctrlProfile>
<rtctrlAttrP descr="" dn="uni/tn-t1/attr-act_rule" name="act_rule">
 <rtctrlSetDamp descr="" halfLife="15" maxSuppressTime="60" name="" reuse="750" suppress="2000"
type="dampening-pol"/>
</rtctrlAttrP>
```

Configuring BGP Max Path Using the REST API

Refer to the *Verified Scalability Guide for Cisco APIC* on the Cisco APIC documentation page for the acceptable values for the following fields.

The two properties that enable you to configure more paths are `maxEcmp` and `maxEcmpIbgp` in the `bgpCtxAfPol` object. After you configure these two properties, they are propagated to the rest of your implementation. The ECMP policy is applied at the VRF level.

The following example provides information on how to configure the BGP Max Path feature using the REST API:

```
<fvTenant descr="" dn="uni/tn-t1" name="t1">
    <fvCtx name="v1">
        <fvRsCtxToBgpCtxAfPol af="ipv4-ucast" tnBgpCtxAfPolName="bgpCtxPol1"/>
    </fvCtx>
    <bgpCtxAfPol name="bgpCtxPol1" maxEcmp="64" maxEcmpIbgp="64"/>
</fvTenant>
```

Configuring AS Path Prepend Using the REST API

This following example provides information on how to configure the AS Path Prepend feature using the REST API:

```
<?xml version="1.0" encoding="UTF-8"?>
<fvTenant name="coke">
    <rtctrlAttrP name="attrp1">
        <rtctrlSetASPath criteria="prepend">
            <rtctrlSetASPathASN asn="100" order="1"/>
            <rtctrlSetASPathASN asn="200" order="10"/>
            <rtctrlSetASPathASN asn="300" order="5"/>
        <rtctrlSetASPath/>
        <rtctrlSetASPath criteria="prepend-last-as" lastnum=”9" />
    </rtctrlAttrP>

    <l3extOut name="out1">
        <rtctrlProfile name="rp1">
            <rtctrlCtxP name="ctxp1" order="1">
                <rtctrlScope>
                    <rtctrlRsScopeToAttrP tnRtctrlAttrPName="attrp1"/>
                </rtctrlScope>
            </rtctrlCtxP>
        </rtctrlProfile>
    </l3extOut>
</fvTenant>
```

Configuring BGP External Routed Network with Autonomous System Override Enabled Using the REST API

SUMMARY STEPS

1. Configure the BGP External Routed Network with Autonomous override enabled.

DETAILED STEPS

Procedure

Configure the BGP External Routed Network with Autonomous override enabled.

Note

The line of code that is in bold displays the BGP AS override portion of the configuration. This feature was introduced in the Cisco APIC Release 3.1(2m).

Example:

```
<fvTenant name="coke">

   <fvCtx name="coke" status="">
       <bgpRtTargetP af="ipv4-ucast">
           <bgpRtTarget type="import" rt="route-target:as4-nn2:1234:1300" />
           <bgpRtTarget type="export" rt="route-target:as4-nn2:1234:1300" />
       </bgpRtTargetP>
       <bgpRtTargetP af="ipv6-ucast">
           <bgpRtTarget type="import" rt="route-target:as4-nn2:1234:1300" />
           <bgpRtTarget type="export" rt="route-target:as4-nn2:1234:1300" />
       </bgpRtTargetP>
   </fvCtx>

   <fvBD name="cokeBD">
       <!-- Association from Bridge Doamin to Private Network -->
       <fvRsCtx tnFvCtxName="coke" />
       <fvRsBDToOut tnL3extOutName="routAccounting" />
       <!-- Subnet behind the bridge domain-->
       <fvSubnet ip="20.1.1.1/16" scope="public"/>
       <fvSubnet ip="2000:1::1/64" scope="public"/>
   </fvBD>
   <fvBD name="cokeBD2">
       <!-- Association from Bridge Doamin to Private Network -->
       <fvRsCtx tnFvCtxName="coke" />
       <fvRsBDToOut tnL3extOutName="routAccounting" />
       <!-- Subnet behind the bridge domain-->
       <fvSubnet ip="30.1.1.1/16" scope="public"/>

   </fvBD>
   <vzBrCP name="webCtrct" scope="global">
       <vzSubj name="http">
           <vzRsSubjFiltAtt tnVzFilterName="default"/>
       </vzSubj>
   </vzBrCP>

   <!-- GOLF L3Out -->
   <l3extOut name="routAccounting">
      <l3extConsLbl name="golf_transit" owner="infra" status=""/>
      <bgpExtP/>
      <l3extInstP name="accountingInst">
          <!--
          <l3extSubnet ip="192.2.2.0/24" scope="import-security,import-rtctrl" />
          <l3extSubnet ip="192.3.2.0/24" scope="export-rtctrl"/>
          <l3extSubnet ip="192.5.2.0/24" scope="export-rtctrl"/>
          <l3extSubnet ip="64:ff9b::c007:200/120" scope="export-rtctrl" />
          -->
          <l3extSubnet ip="0.0.0.0/0"
                               scope="export-rtctrl,import-security"
                               aggregate="export-rtctrl"

          />
          <fvRsProv tnVzBrCPName="webCtrct"/>
      </l3extInstP>

      <l3extRsEctx tnFvCtxName="coke"/>
   </l3extOut>
```

```
    <fvAp name="cokeAp">
      <fvAEPg name="cokeEPg" >
          <fvRsBd tnFvBDName="cokeBD" />
            <fvRsPathAtt tDn="topology/pod-1/paths-103/pathep-[eth1/20]" encap="vlan-100"
instrImedcy="immediate" mode="regular"/>
          <fvRsCons tnVzBrCPName="webCtrct"/>
      </fvAEPg>
      <fvAEPg name="cokeEPg2" >
          <fvRsBd tnFvBDName="cokeBD2" />
            <fvRsPathAtt tDn="topology/pod-1/paths-103/pathep-[eth1/20]" encap="vlan-110"
instrImedcy="immediate" mode="regular"/>
          <fvRsCons tnVzBrCPName="webCtrct"/>
      </fvAEPg>
    </fvAp>

    <!-- Non GOLF L3Out-->
    <l3extOut name="NonGolfOut">
       <bgpExtP/>
       <l3extLNodeP name="bLeaf">
           <!--
           <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="20.1.13.1"/>
           -->
           <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="20.1.13.1">
           <l3extLoopBackIfP addr="1.1.1.1"/>

           <ipRouteP ip="2.2.2.2/32"  >
             <ipNexthopP nhAddr="20.1.12.3"/>
      </ipRouteP>


        </l3extRsNodeL3OutAtt>
           <l3extLIfP name='portIfV4'>
              <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/17]" encap='vlan-1010'
 ifInstT='sub-interface' addr="20.1.12.2/24">

               </l3extRsPathL3OutAtt>
           </l3extLIfP>
           <l3extLIfP name='portIfV6'>
              <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/17]" encap='vlan-1010'
 ifInstT='sub-interface' addr="64:ff9b::1401:302/120">
                    <bgpPeerP addr="64:ff9b::1401:d03" ctrl="send-com,send-ext-com" />
                </l3extRsPathL3OutAtt>
           </l3extLIfP>
           <bgpPeerP addr="2.2.2.2" ctrl="as-override,disable-peer-as-check, send-com,send-ext-com"
 status=""/>
       </l3extLNodeP>
       <!--
        <bgpPeerP addr="2.2.2.2" ctrl="send-com,send-ext-com" status=""/>
        -->
       <l3extInstP name="accountingInst">
           <l3extSubnet ip="192.10.0.0/16" scope="import-security,import-rtctrl" />
           <l3extSubnet ip="192.3.3.0/24" scope="import-security,import-rtctrl" />
           <l3extSubnet ip="192.4.2.0/24" scope="import-security,import-rtctrl" />
           <l3extSubnet ip="64:ff9b::c007:200/120" scope="import-security,import-rtctrl" />
           <l3extSubnet ip="192.2.2.0/24" scope="export-rtctrl" />
           <l3extSubnet ip="0.0.0.0/0"
                        scope="export-rtctrl,import-rtctrl,import-security"
                        aggregate="export-rtctrl,import-rtctrl"

         />
       </l3extInstP>
      <l3extRsEctx tnFvCtxName="coke"/>
  </l3extOut>
```

```
</fvTenant>
```

Configuring BGP Neighbor Shutdown and Soft Reset Using the REST API

Configuring BGP Neighbor Shutdown Using the REST API

The following procedure describes how to use the BGP neighbor shutdown feature using the REST API.

Procedure

Step 1 Configure the node and interface.

This example configures VRF `v1` on node 103 (the border leaf switch), with the node profile, `nodep1`, and router ID `11.11.11.103`. It also configures interface `eth1/3` as a routed interface (Layer 3 port), with IP address `12.12.12.1/24` and Layer 3 domain `dom1`.

Example:

```
<l3extOut name="l3out1">
    <l3extRsEctx tnFvCtxName="v1"/>
    <l3extLNodeP name="nodep1">
        <l3extRsNodeL3OutAtt rtrId="11.11.11.103" tDn="topology/pod-1/node-103"/>
        <l3extLIfP name="ifp1"/>
         <l3extRsPathL3OutAtt addr="12.12.12.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-103/pathep-[eth1/3]"/>
        </l3extLIfP>
    </l3extLNodeP>
    <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
</l3extOut>
```

Step 2 Configure the BGP routing protocol and configure the BGP neighbor shutdown feature.

This example configures BGP as the primary routing protocol, with a BGP peer with the IP address, `15.15.15.2` and ASN 100.

The **adminSt** variable can be set to one of the following:

- **enabled**: Enables the BGP neighbor shutdown feature.
- **disabled**: Disables the BGP neighbor shutdown feature.

In the following example, the BGP neighbor shutdown feature is enabled.

Example:

```
<l3extOut name="l3out1">
    <l3extLNodeP name="nodep1">
        <bgpPeerP addr="15.15.15.2"> adminSt="enabled"
            <bgpAsP asn="100"/>
        </bgpPeerP>
    </l3extLNodeP>
    <bgpExtP/>
</l3extOut>
```

Configuring BGP Neighbor Soft Reset Using the REST API

The following procedure describes how to use the BGP neighbor soft reset feature using the REST API.

Procedure

Step 1 Configure the node and interface.

This example configures VRF `v1` on node 103 (the border leaf switch), with the node profile, `nodep1`, and router ID `11.11.11.103`. It also configures interface `eth1/3` as a routed interface (Layer 3 port), with IP address `12.12.12.1/24` and Layer 3 domain `dom1`.

Example:

```
<l3extOut name="l3out1">
    <l3extRsEctx tnFvCtxName="v1"/>
    <l3extLNodeP name="nodep1">
        <l3extRsNodeL3OutAtt rtrId="11.11.11.103" tDn="topology/pod-1/node-103"/>
        <l3extLIfP name="ifp1"/>
         <l3extRsPathL3OutAtt addr="12.12.12.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-103/pathep-[eth1/3]"/>
        </l3extLIfP>
    </l3extLNodeP>
    <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
</l3extOut>
```

Step 2 Configure the BGP routing protocol and configure the BGP neighbor soft reset feature.

This example configures BGP as the primary routing protocol, with a BGP peer with the IP address, `15.15.15.2` and ASN 100.

The **dir** variable can be set to one of the following:

- **in**: Enables the soft dynamic inbound reset.
- **out**: Enables the soft outbound reset.

In the following example, the soft dynamic inbound reset is enabled.

Example:

```
<l3extOut name="l3out1">
    <l3extLNodeP name="nodep1">
        <bgpPeerP addr="15.15.15.2">
            <bgpAsP asn="100"/>
            <bgpPeerEntryClearPeerLTask>
                <attributes>
                    <mode>soft</mode>
                    <dir>in</dir>
                    <adminSt>start</adminSt>
                </attributes>
                <children/>
            </bgpPeerEntryClearPeerLTask>
        </bgpPeerP>
    </l3extLNodeP>
    <bgpExtP/>
</l3extOut>
```

Configuring a Per VRF Per Node BGP Timer Using the REST API

The following example shows how to configure Per VRF Per node BGP timer in a node. Configure `bgpProtP` under `l3extLNodeP` configuration. Under `bgpProtP`, configure a relation (`bgpRsBgpNodeCtxPol`) to the desired BGP Context Policy (`bgpCtxPol`).

Procedure

Configure a node specific BGP timer policy on `node1`, and configure `node2` with a BGP timer policy that is not node specific.

Example:

```
POST https://apic-ip-address/mo.xml

<fvTenant name="tn1" >
    <bgpCtxPol name="pol1" staleIntvl="25" />
    <bgpCtxPol name="pol2" staleIntvl="35" />
    <fvCtx name="ctx1" >
      <fvRsBgpCtxPol tnBgpCtxPolName="pol1"/>
    </fvCtx>
     <l3extout name="out1" >
      <l3extRsEctx toFvCtxName="ctx1" />
      <l3extLNodeP name="node1" >
        <bgpProtP name="protp1" >
            <bgpRsBgpNodeCtxPol tnBgpCtxPolName="pol2" />
        </bgpProtP>
      </l3extLNodeP>
      <l3extLNodeP name="node2" >
      </l3extLNodeP>
```

In this example, `node1` gets BGP timer values from policy `pol2`, and `node2` gets BGP timer values from `pol1`. The timer values are applied to the `bgpDom` corresponding to VRF `tn1:ctx1`. This is based upon the BGP timer policy that is chosen following the algorithm described in the *Per VRF Per Node BPG Timer Values* section.

Deleting a Per VRF Per Node BGP Timer Using the REST API

The following example shows how to delete an existing Per VRF Per node BGP timer in a node.

Procedure

Delete the node specific BGP timer policy on `node1`.

Example:

```
POST https://apic-ip-address/mo.xml

<fvTenant name="tn1" >
    <bgpCtxPol name="pol1" staleIntvl="25" />
    <bgpCtxPol name="pol2" staleIntvl="35" />
    <fvCtx name="ctx1" >
      <fvRsBgpCtxPol tnBgpCtxPolName="pol1"/>
    </fvCtx>
     <l3extout name="out1" >
      <l3extRsEctx toFvCtxName="ctx1" />
      <l3extLNodeP name="node1" >
```

```
  <bgpProtP name="protp1" status="deleted" >
      <bgpRsBgpNodeCtxPol tnBgpCtxPolName="pol2" />
  </bgpProtP>
</l3extLNodeP>
<l3extLNodeP name="node2" >
</l3extLNodeP>
```

The code phrase `<bgpProtP name="protp1" status="deleted" >` in the example above, deletes the BGP timer policy. After the deletion, `node1` defaults to the BGP timer policy for the VRF with which `node1` is associated, which is `pol1` in the above example.

Configuring Bidirectional Forwarding Detection on a Secondary IP Address Using the REST API

The following example configures bidirectional forwarding detection (BFD) on a secondary IP address using the REST API:

```
<l3extLIfP
  dn="uni/tn-sec-ip-bfd/out-secip-bfd-l3out/lnodep-secip-bfd-l3out_nodeProfile/
  lifp-secip-bfd-l3out_interfaceProfile" name="secip-bfd-l3out_interfaceProfile"
  prio="unspecified" tag="yellow-green" userdom=":all:">
    <l3extRsPathL3OutAtt addr="50.50.50.200/24" autostate="disabled"
      encap="vlan-2" encapScope="local" ifInstT="ext-svi" ipv6Dad="enabled"
      isMultiPodDirect="no" llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular"
      mtu="inherit" tDn="topology/pod-1/paths-101/pathep-[eth1/3]"
      targetDscp="unspecified" userdom=":all:">
        <l3extIp addr="9.9.9.1/24" ipv6Dad="enabled" userdom=":all:"/>
        <l3extIp addr="6.6.6.1/24" ipv6Dad="enabled" userdom=":all:"/>
    </l3extRsPathL3OutAtt>
    <l3extRsNdIfPol userdom="all"/>
    <l3extRsLIfPCustQosPol userdom="all"/>
    <l3extRsIngressQosDppPol userdom="all"/>
    <l3extRsEgressQosDppPol userdom="all"/>
    <l3extRsArpIfPol userdom="all"/>
</l3extLIfP>
<ipRouteP aggregate="no"
  dn="uni/tn-sec-ip-bfd/out-secip-bfd-l3out/lnodep-secip-bfd-l3out_nodeProfile/
  rsnodeL3OutAtt-[topology/pod-1/node-101]/rt-[6.0.0.1/24]"
  fromPfxLen="0" ip="6.0.0.1/24" pref="1" rtCtrl="bfd" toPfxLen="0" userdom=":all:">
    <ipNexthopP nhAddr="6.6.6.2" pref="unspecified" type="prefix" userdom=":all:"/>
</ipRouteP>
```

Configuring BFD Globally Using the REST API

Procedure

The following REST API shows the global configuration for bidirectional forwarding detection (BFD):

Example:

```
<polUni>
 <infraInfra>
    <bfdIpv4InstPol name="default" echoSrcAddr="1.2.3.4" slowIntvl="1000" minTxIntvl="150"
minRxIntvl="250" detectMult="5" echoRxIntvl="200"/>
    <bfdIpv6InstPol name="default" echoSrcAddr="34::1/64" slowIntvl="1000" minTxIntvl="150"
minRxIntvl="250" detectMult="5" echoRxIntvl="200"/>
```

```
 </infraInfra>
</polUni>
```

Configuring BFD Interface Override Using the REST API

Procedure

The following REST API shows the interface override configuration for bidirectional forwarding detection (BFD):

Example:

```
<fvTenant name="ExampleCorp">
  <bfdIfPol name="bfdIfPol" minTxIntvl="400" minRxIntvl="400" detectMult="5" echoRxIntvl="400"
echoAdminSt="disabled"/>
    <l3extOut name="l3-out">
        <l3extLNodeP name="leaf1">
            <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="2.2.2.2"/>
            <l3extLIfP name='portIpv4'>
              <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/11]" ifInstT='l3-port'
 addr="10.0.0.1/24" mtu="1500"/>
                <bfdIfP type="sha1" key="password">
                    <bfdRsIfPol tnBfdIfPolName='bfdIfPol'/>
                </bfdIfP>
            </l3extLIfP>

        </l3extLNodeP>
    </l3extOut>
</fvTenant>
```

Configuring BFD Consumer Protocols Using the REST API

Procedure

Step 1 The following example shows the interface configuration for bidirectional forwarding detection (BFD):

Example:

```
<fvTenant name="ExampleCorp">
  <bfdIfPol name="bfdIfPol" minTxIntvl="400" minRxIntvl="400" detectMult="5" echoRxIntvl="400"
echoAdminSt="disabled"/>
    <l3extOut name="l3-out">
        <l3extLNodeP name="leaf1">
            <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="2.2.2.2"/>
            <l3extLIfP name='portIpv4'>
              <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/11]" ifInstT='l3-port'
 addr="10.0.0.1/24" mtu="1500"/>
                <bfdIfP type="sha1" key="password">
                    <bfdRsIfPol tnBfdIfPolName='bfdIfPol'/>
                </bfdIfP>
            </l3extLIfP>

        </l3extLNodeP>
```

```
    </l3extOut>
</fvTenant>
```

Step 2 The following example shows the interface configuration for enabling BFD on OSPF and EIGRP:

Example:

BFD on leaf switch

```
<fvTenant name="ExampleCorp">
      <ospfIfPol  name="ospf_intf_pol" cost="10" ctrl="bfd"/>
      <eigrpIfPol ctrl="nh-self,split-horizon,bfd" dn="uni/tn-Coke/eigrpIfPol-eigrp_if_default"
</fvTenant>
```

Example:

BFD on spine switch

```
<l3extLNodeP name="bSpine">

             <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-103" rtrId="192.3.1.8">
                 <l3extLoopBackIfP addr="10.10.3.1" />
                 <l3extInfraNodeP fabricExtCtrlPeering="false" />
             </l3extRsNodeL3OutAtt>

             <l3extLIfP name='portIf'>
                <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-103/pathep-[eth5/10]" encap='vlan-4'
 ifInstT='sub-interface' addr="20.3.10.1/24"/>
                 <ospfIfP>
                     <ospfRsIfPol tnOspfIfPolName='ospf_intf_pol'/>
                 </ospfIfP>
                 <bfdIfP name="test" type="sha1" key="hello" status="created,modified">
                    <bfdRsIfPol tnBfdIfPolName='default' status="created,modified"/>
                </bfdIfP>
             </l3extLIfP>

         </l3extLNodeP>
```

Step 3 The following example shows the interface configuration for enabling BFD on BGP:

Example:

```
<fvTenant name="ExampleCorp">
    <l3extOut name="l3-out">
        <l3extLNodeP name="leaf1">
            <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="2.2.2.2"/>
            <l3extLIfP name='portIpv4'>
              <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/11]" ifInstT='l3-port'
 addr="10.0.0.1/24" mtu="1500">
                  <bgpPeerP addr="4.4.4.4/24" allowedSelfAsCnt="3" ctrl="bfd" descr="" name=""
peerCtrl="" ttl="1">
                      <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
                      <bgpAsP asn="3" descr="" name=""/>
                  </bgpPeerP>
               </l3extRsPathL3OutAtt>
            </l3extLIfP>

        </l3extLNodeP>
    </l3extOut>
</fvTenant>
```

Step 4 The following example shows the interface configuration for enabling BFD on Static Routes:

Performing Tasks Using REST API

Example:

BFD on leaf switch

```
<fvTenant name="ExampleCorp">
    <l3extOut name="l3-out">
        <l3extLNodeP name="leaf1">
            <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="2.2.2.2">
              <ipRouteP ip="192.168.3.4" rtCtrl="bfd">
                <ipNexthopP nhAddr="192.168.62.2"/>
              </ipRouteP>
            </l3extRsNodeL3OutAtt>
            <l3extLIfP name='portIpv4'>
               <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/3]" ifInstT='l3-port'
 addr="10.10.10.2/24" mtu="1500" status="created,modified" />
            </l3extLIfP>

        </l3extLNodeP>

    </l3extOut>
</fvTenant>
```

Example:

BFD on spine switch

```
<l3extLNodeP name="bSpine">

             <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-103" rtrId="192.3.1.8">
              <ipRouteP ip="0.0.0.0" rtCtrl="bfd">
                <ipNexthopP nhAddr="192.168.62.2"/>
              </ipRouteP>
             </l3extRsNodeL3OutAtt>

             <l3extLIfP name='portIf'>
                <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-103/pathep-[eth5/10]" encap='vlan-4'
 ifInstT='sub-interface' addr="20.3.10.1/24"/>

                 <bfdIfP name="test" type="sha1" key="hello" status="created,modified">
                    <bfdRsIfPol tnBfdIfPolName='default' status="created,modified"/>
                 </bfdIfP>
             </l3extLIfP>

         </l3extLNodeP>
```

Step 5 The following example shows the interface configuration for enabling BFD on IS-IS:

Example:

```
<fabricInst>
          <l3IfPol name="testL3IfPol" bfdIsis="enabled"/>
              <fabricLeafP name="LeNode" >
  <fabricRsLePortP tDn="uni/fabric/leportp-leaf_profile" />
 <fabricLeafS name="spsw" type="range">
 <fabricNodeBlk name="node101" to_="102" from_="101" />
 </fabricLeafS>
            </fabricLeafP>

           <fabricSpineP name="SpNode" >
 <fabricRsSpPortP tDn="uni/fabric/spportp-spine_profile" />
 <fabricSpineS name="spsw" type="range">
     <fabricNodeBlk name="node103" to_="103" from_="103" />
 </fabricSpineS>
```

```
        </fabricSpineP>

         <fabricLePortP name="leaf_profile">
 <fabricLFPortS name="leafIf" type="range">
<fabricPortBlk name="spBlk" fromCard="1" fromPort="49" toCard="1" toPort="49" />
       <fabricRsLePortPGrp tDn="uni/fabric/funcprof/leportgrp-LeTestPGrp" />
 </fabricLFPortS>
        </fabricLePortP>

       <fabricSpPortP name="spine_profile">
 <fabricSFPortS name="spineIf" type="range">
      <fabricPortBlk name="spBlk" fromCard="5" fromPort="1" toCard="5" toPort="2" />
      <fabricRsSpPortPGrp tDn="uni/fabric/funcprof/spportgrp-SpTestPGrp" />
 </fabricSFPortS>
     </fabricSpPortP>

 <fabricFuncP>
               <fabricLePortPGrp name = "LeTestPGrp">
 <fabricRsL3IfPol tnL3IfPolName="testL3IfPol"/>
              </fabricLePortPGrp>

           <fabricSpPortPGrp name = "SpTestPGrp">
 <fabricRsL3IfPol tnL3IfPolName="testL3IfPol"/>
          </fabricSpPortPGrp>

</fabricFuncP>

</fabricInst>
```

Configuring OSPF External Routed Networks Using REST API

Creating OSPF External Routed Network for Management Tenant Using REST API

- You must verify that the router ID and the logical interface profile IP address are different and do not overlap.
- The following steps are for creating an OSPF external routed network for a management tenant. To create an OSPF external routed network for a tenant, you must choose a tenant and create a VRF for the tenant.
- For more details, see *Cisco APIC and Transit Routing*.

Procedure

Create an OSPF external routed network for management tenant.

Example:

```
POST: https://apic-ip-address/api/mo/uni/tn-mgmt.xml

<fvTenant name="mgmt">
   <fvBD name="bd1">
      <fvRsBDToOut tnL3extOutName="RtdOut" />
      <fvSubnet ip="1.1.1.1/16" />
      <fvSubnet ip="1.2.1.1/16" />
      <fvSubnet ip="40.1.1.1/24" scope="public" />
      <fvRsCtx tnFvCtxName="inb" />
   </fvBD>
```

```
   <fvCtx name="inb" />

   <l3extOut name="RtdOut">
      <l3extRsL3DomAtt tDn="uni/l3dom-extdom"/>
      <l3extInstP name="extMgmt">
      </l3extInstP>
      <l3extLNodeP name="borderLeaf">
         <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="10.10.10.10"/>
         <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-102" rtrId="10.10.10.11"/>
         <l3extLIfP name='portProfile'>
           <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/40]" ifInstT='l3-port'
addr="192.168.62.1/24"/>
           <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-102/pathep-[eth1/40]" ifInstT='l3-port'
addr="192.168.62.5/24"/>
            <ospfIfP/>
                               </l3extLIfP>
      </l3extLNodeP>
      <l3extRsEctx tnFvCtxName="inb"/>
      <ospfExtP areaId="57" />
   </l3extOut>
</fvTenant>
```

Configuring EIGRP External Routed Networks Using REST API

Configuring EIGRP Using the REST API

Procedure

Step 1 Configure an EIGRP context policy.

Example:

```
<polUni>
    <fvTenant name="cisco_6">
        <eigrpCtxAfPol actIntvl="3" descr="" dn="uni/tn-cisco_6/eigrpCtxAfP-eigrp_default_pol"
extDist="170"
          intDist="90" maxPaths="8" metricStyle="narrow" name="eigrp_default_pol" ownerKey=""
ownerTag=""/>
    </fvTenant>
</polUni>
```

Step 2 Configure an EIGRP interface policy.

Example:

```
<polUni>
    <fvTenant name="cisco_6">
       <eigrpIfPol bw="10" ctrl="nh-self,split-horizon" delay="10" delayUnit="tens-of-micro" descr=""
 dn="uni/tn-cisco_6/eigrpIfPol-eigrp_if_default"
          helloIntvl="5" holdIntvl="15" name="eigrp_if_default" ownerKey="" ownerTag=""/>
    </fvTenant>
</polUni>
```

Step 3 Configure an EIGRP VRF.

Example:

IPv4:

```
<polUni>
    <fvTenant name="cisco_6">
        <fvCtx name="dev">
          <fvRsCtxToEigrpCtxAfPol tnEigrpCtxAfPolName="eigrp_ctx_pol_v4" af="1"/>
        </fvCtx>
    </fvTenant>
</polUni>
```

IPv6:

```
<polUni>
    <fvTenant name="cisco_6">
        <fvCtx name="dev">
          <fvRsCtxToEigrpCtxAfPol tnEigrpCtxAfPolName="eigrp_ctx_pol_v6" af="ipv6-ucast"/>
        </fvCtx>
    </fvTenant>
</polUni>
```

Step 4 Configure an EIGRP Layer3 Outside.

Example:

IPv4

```
<polUni>
    <fvTenant name="cisco_6">
        <l3extOut name="ext">
            <eigrpExtP asn="4001"/>
            <l3extLNodeP name="node1">
                <l3extLIfP name="intf_v4">
                    <l3extRsPathL3OutAtt addr="201.1.1.1/24" ifInstT="l3-port"
                      tDn="topology/pod-1/paths-101/pathep-[eth1/4]"/>
                    <eigrpIfP name="eigrp_ifp_v4">
                        <eigrpRsIfPol tnEigrpIfPolName="eigrp_if_pol_v4"/>
                    </eigrpIfP>
                </l3extLIfP>
            </l3extLNodeP>
        </l3extOut>
    </fvTenant>
</polUni>
```

IPv6

```
<polUni>
    <fvTenant name="cisco_6">
        <l3extOut name="ext">
            <eigrpExtP asn="4001"/>
            <l3extLNodeP name="node1">
                <l3extLIfP name="intf_v6">
                    <l3extRsPathL3OutAtt addr="2001::1/64" ifInstT="l3-port"
                      tDn="topology/pod-1/paths-101/pathep-[eth1/4]"/>
                    <eigrpIfP name="eigrp_ifp_v6">
                        <eigrpRsIfPol tnEigrpIfPolName="eigrp_if_pol_v6"/>
                    </eigrpIfP>
                </l3extLIfP>
            </l3extLNodeP>
        </l3extOut>
    </fvTenant>
</polUni>
```

IPv4 and IPv6

```
<polUni>
    <fvTenant name="cisco_6">
        <l3extOut name="ext">
            <eigrpExtP asn="4001"/>
```

```
            <l3extLNodeP name="node1">
                <l3extLIfP name="intf_v4">
                    <l3extRsPathL3OutAtt addr="201.1.1.1/24" ifInstT="l3-port"
                      tDn="topology/pod-1/paths-101/pathep-[eth1/4]"/>
                    <eigrpIfP name="eigrp_ifp_v4">
                        <eigrpRsIfPol tnEigrpIfPolName="eigrp_if_pol_v4"/>
                    </eigrpIfP>
                </l3extLIfP>

                <l3extLIfP name="intf_v6">
                    <l3extRsPathL3OutAtt addr="2001::1/64" ifInstT="l3-port"
                      tDn="topology/pod-1/paths-101/pathep-[eth1/4]"/>
                    <eigrpIfP name="eigrp_ifp_v6">
                        <eigrpRsIfPol tnEigrpIfPolName="eigrp_if_pol_v6"/>
                    </eigrpIfP>
                </l3extLIfP>
            </l3extLNodeP>
        </l3extOut>
    </fvTenant>
</polUni>
```

Step 5 (Optional) Configure the interface policy knobs.

Example:

```
<polUni>
    <fvTenant name="cisco_6">
        <eigrpIfPol bw="1000000" ctrl="nh-self,split-horizon" delay="10"
          delayUnit="tens-of-micro" helloIntvl="5" holdIntvl="15" name="default"/>
    </fvTenant>
</polUni>
```

The `bandwidth` (bw) attribute is defined in Kbps. The `delayUnit` attribute can be "tens of micro" or "pico".

Configuring Route Summarization Using REST API

Configuring Route Summarization for BGP, OSPF, and EIGRP Using the REST API

Procedure

Step 1 Configure BGP route summarization using the REST API as follows:

Example:

```
<fvTenant name="common">
           <fvCtx name="vrf1"/>
   <bgpRtSummPol name="bgp_rt_summ" cntrl='as-set'/>
   <l3extOut name="l3_ext_pol" >
             <l3extLNodeP name="bLeaf">
                <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="20.10.1.1"/>
              <l3extLIfP name='portIf'>
              <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/31]" ifInstT='l3-port'
 addr="10.20.1.3/24/>
               </l3extLIfP>
           </l3extLNodeP>
         <bgpExtP />
```

```
          <l3extInstP name="InstP" >
         <l3extSubnet ip="10.0.0.0/8" scope="export-rtctrl">
          <l3extRsSubnetToRtSumm tDn="uni/tn-common/bgprtsum-bgp_rt_summ"/>
          <l3extRsSubnetToProfile tnRtctrlProfileName="rtprof"/>
         </l3extSubnet>
       </l3extInstP>
       <l3extRsEctx tnFvCtxName="vrf1"/>
   </l3extOut>
</fvTenant>
```

Step 2 Configure OSPF inter-area and external summarization using the following REST API:

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<fvTenant name="t20">
  <!--Ospf Inter External route summarization Policy-->
  <ospfRtSummPol cost="unspecified" interAreaEnabled="no" name="ospfext"/>
  <!--Ospf Inter Area route summarization Policy-->
  <ospfRtSummPol cost="16777215" interAreaEnabled="yes" name="interArea"/>
  <fvCtx name="ctx0" pcEnfDir="ingress" pcEnfPref="enforced"/>
  <!-- L3OUT backbone Area-->
  <l3extOut enforceRtctrl="export" name="l3_1" ownerKey="" ownerTag="" targetDscp="unspecified">
    <l3extRsEctx tnFvCtxName="ctx0"/>
    <l3extLNodeP name="node-101">
      <l3extRsNodeL3OutAtt rtrId="20.1.3.2" rtrIdLoopBack="no" tDn="topology/pod-1/node-101"/>
      <l3extLIfP name="intf-1">
        <l3extRsPathL3OutAtt addr="20.1.5.2/24" encap="vlan-1001" ifInstT="sub-interface"
tDn="topology/pod-1/paths-101/pathep-[eth1/33]"/>
      </l3extLIfP>
    </l3extLNodeP>
    <l3extInstP name="l3InstP1">
      <fvRsProv tnVzBrCPName="default"/>
      <!--Ospf External Area route summarization-->
      <l3extSubnet aggregate="" ip="193.0.0.0/8" name="" scope="export-rtctrl">
        <l3extRsSubnetToRtSumm tDn="uni/tn-t20/ospfrtsumm-ospfext"/>
      </l3extSubnet>
    </l3extInstP>
    <ospfExtP areaCost="1" areaCtrl="redistribute,summary" areaId="backbone" areaType="regular"/>
  </l3extOut>
  <!-- L3OUT Regular Area-->
  <l3extOut enforceRtctrl="export" name="l3_2">
    <l3extRsEctx tnFvCtxName="ctx0"/>
    <l3extLNodeP name="node-101">
      <l3extRsNodeL3OutAtt rtrId="20.1.3.2" rtrIdLoopBack="no" tDn="topology/pod-1/node-101"/>
      <l3extLIfP name="intf-2">
        <l3extRsPathL3OutAtt addr="20.1.2.2/24" encap="vlan-1014" ifInstT="sub-interface"
tDn="topology/pod-1/paths-101/pathep-[eth1/11]"/>
      </l3extLIfP>
    </l3extLNodeP>
    <l3extInstP matchT="AtleastOne" name="l3InstP2">
      <fvRsCons tnVzBrCPName="default"/>
      <!--Ospf Inter Area route summarization-->
      <l3extSubnet aggregate="" ip="197.0.0.0/8" name="" scope="export-rtctrl">
        <l3extRsSubnetToRtSumm tDn="uni/tn-t20/ospfrtsumm-interArea"/>
      </l3extSubnet>
    </l3extInstP>
    <ospfExtP areaCost="1" areaCtrl="redistribute,summary" areaId="0.0.0.57" areaType="regular"/>
  </l3extOut>
</fvTenant>
```

Step 3 Configure EIGRP summarization using the following REST API:

Example:

```
<fvTenant name="exampleCorp">
  <l3extOut name="out1">
    <l3extInstP  name="eigrpSummInstp" >
      <l3extSubnet aggregate="" descr="" ip="197.0.0.0/8" name="" scope="export-rtctrl">
        <l3extRsSubnetToRtSumm/>
      </l3extSubnet>
    </l3extInstP>
  </l3extOut>
  <eigrpRtSummPol name="pol1" />
```

Note
There is no route summarization policy to be configured for EIGRP. The only configuration needed for enabling EIGRP summarization is the summary subnet under the InstP.

Configuring Route Control with Route Maps and Route Profile Using REST API

Configuring Route Control Per BGP Peer Using the REST API

The following procedure describes how to configure the route control per BGP peer feature using the REST API.

Procedure

Configure the route control per BGP peer feature.

Where:

- **direction="import"** is the route import policy (routes allowed into the fabric)
- **direction="export"** is the route export policy (routes advertised out the external network)

Example:

```
<polUni>
   <fvTenant name="t1">
       <fvCtx name="v1"/>
       <l3extOut name="l3out1">
           <l3extRsEctx tnFvCtxName="v1"/>
           <l3extLNodeP name="nodep1">
               <l3extRsNodeL3OutAtt rtrId="11.11.11.103" tDn="topology/pod-1/node-103"/>
               <l3extLIfP name="ifp1">
                   <l3extRsPathL3OutAtt addr="12.12.12.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-103/pathep-[eth1/3]"/>
               </l3extLIfP>
               <bgpPeerP addr="15.15.15.2">
                   <bgpAsP asn="100"/>
                   <bgpRsPeerToProfile direction="export" tnRtctrlProfileName="rp1"/>
               </bgpPeerP>
           </l3extLNodeP>
           <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
           <bgpExtP/>
```

```
            <ospfExtP areaId="0.0.0.0" areaType="regular"/>
            <l3extInstP name="extnw1" >
                <l3extSubnet ip="20.20.20.0/24" scope="import-security"/>
            </l3extInstP>
        </l3extOut>
        <rtctrlProfile name="rp1">
            <rtctrlCtxP name="ctxp1" action="permit" order="0">
                <rtctrlScope>
                    <rtctrlRsScopeToAttrP tnRtctrlAttrPName="attrp1"/>
                </rtctrlScope>
                <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1"/>
            </rtctrlCtxP>
        </rtctrlProfile>
        <rtctrlSubjP name="match-rule1">
            <rtctrlMatchRtDest ip="200.3.2.0/24"/>
        </rtctrlSubjP>
        <rtctrlAttrP name="attrp1">
            <rtctrlSetASPath criteria="prepend">
                <rtctrlSetASPathASN asn="100" order="2"/>
                <rtctrlSetASPathASN asn="200" order="1"/>
            </rtctrlSetASPath>
        </rtctrlAttrP>
</fvTenant>
</polUni>
```

Configuring Route Map/Profile with Explicit Prefix List Using REST API

Before you begin

- Tenant and VRF must be configured.

Procedure

Configure the route map/profile using explicit prefix list.

Note
The entries shown in **bold** below are enhancements for match prefix that are available for APIC releases 4.2(3) and later. For more information on these fields, see Enhancements for Match Prefix.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<fvTenant name="PM" status="">
   <rtctrlAttrP name="set_dest">
      <rtctrlSetComm community="regular:as2-nn2:5:24" />
   </rtctrlAttrP>
   <rtctrlSubjP name="allow_dest">
      <rtctrlMatchRtDest ip="192.169.0.0/24" aggregate="yes" fromPfxLen="26" toPfxLen="30" />
      <rtctrlMatchCommTerm name="term1">
         <rtctrlMatchCommFactor community="regular:as2-nn2:5:24" status="" />
         <rtctrlMatchCommFactor community="regular:as2-nn2:5:25" status="" />
      </rtctrlMatchCommTerm>
      <rtctrlMatchCommRegexTerm commType="regular" regex="200:*" status="" />
   </rtctrlSubjP>
   <rtctrlSubjP name="deny_dest">
      <rtctrlMatchRtDest ip="192.168.0.0/24" />
```

```
  </rtctrlSubjP>
  <fvCtx name="ctx" />
  <l3extOut name="L3Out_1" enforceRtctrl="import,export" status="">
     <l3extRsEctx tnFvCtxName="ctx" />
     <l3extLNodeP name="bLeaf">
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="1.2.3.4" />
        <l3extLIfP name="portIf">
         <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/25]" ifInstT="sub-interface"
encap="vlan-1503" addr="10.11.12.11/24" />
           <ospfIfP />
        </l3extLIfP>
        <bgpPeerP addr="5.16.57.18/32" ctrl="send-com" />
        <bgpPeerP addr="6.16.57.18/32" ctrl="send-com" />
     </l3extLNodeP>
     <bgpExtP />
     <ospfExtP areaId="0.0.0.59" areaType="nssa" status="" />
     <l3extInstP name="l3extInstP_1" status="">
        <l3extSubnet ip="17.11.1.11/24" scope="import-security" />
     </l3extInstP>
     <rtctrlProfile name="default-export" type="global" status="">
        <rtctrlCtxP name="ctx_deny" action="deny" order="1">
           <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="deny_dest" status="" />
        </rtctrlCtxP>
        <rtctrlCtxP name="ctx_allow" order="2">
           <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="allow_dest" status="" />
        </rtctrlCtxP>
        <rtctrlScope name="scope" status="">
           <rtctrlRsScopeToAttrP tnRtctrlAttrPName="set_dest" status="" />
        </rtctrlScope>
     </rtctrlProfile>
  </l3extOut>
  <fvBD name="testBD">
     <fvRsBDToOut tnL3extOutName="L3Out_1" />
     <fvRsCtx tnFvCtxName="ctx" />
     <fvSubnet ip="40.1.1.12/24" scope="public" />
     <fvSubnet ip="40.1.1.2/24" scope="private" />
     <fvSubnet ip="2003::4/64" scope="public" />
  </fvBD>
</fvTenant>
```

Configuring a Route Control Protocol to Use Import and Export Controls, With the REST API

This example assumes that you have configured the Layer 3 outside network connections using BGP. It is also possible to perform these tasks for a network using OSPF.

Before you begin

- The tenant, private network, and bridge domain are created.
- The Layer 3 outside tenant network is configured.

Procedure

Configure the route control protocol using import and export controls.

Example:

```
<l3extOut descr="" dn="uni/tn-Ten_ND/out-L3Out1" enforceRtctrl="export" name="L3Out1" ownerKey=""
ownerTag="" targetDscp="unspecified">
        <l3extLNodeP descr="" name="LNodeP1" ownerKey="" ownerTag="" tag="yellow-green"
targetDscp="unspecified">
            <l3extRsNodeL3OutAtt rtrId="1.2.3.4" rtrIdLoopBack="yes" tDn="topology/pod-1/node-101">
                <l3extLoopBackIfP addr="2000::3" descr="" name=""/>
            </l3extRsNodeL3OutAtt>
            <l3extLIfP descr="" name="IFP1" ownerKey="" ownerTag="" tag="yellow-green">
                <ospfIfP authKeyId="1" authType="none" descr="" name="">
                    <ospfRsIfPol tnOspfIfPolName=""/>
                </ospfIfP>
                <l3extRsNdIfPol tnNdIfPolName=""/>
               <l3extRsPathL3OutAtt addr="10.11.12.10/24" descr="" encap="unknown" ifInstT="l3-port"

llAddr="::" mac="00:22:BD:F8:19:FF" mtu="1500" tDn="topology/pod-1/paths-101/pathep-[eth1/17]"
targetDscp="unspecified"/>
            </l3extLIfP>
        </l3extLNodeP>
        <l3extRsEctx tnFvCtxName="PVN1"/>
        <l3extInstP descr="" matchT="AtleastOne" name="InstP1" prio="unspecified"
targetDscp="unspecified">
            <fvRsCustQosPol tnQosCustomPolName=""/>
            <l3extSubnet aggregate="" descr="" ip="192.168.1.0/24" name="" scope=""/>
        </l3extInstP>
        <ospfExtP areaCost="1" areaCtrl="redistribute,summary" areaId="0.0.0.1" areaType="nssa"
descr=""/>
        <rtctrlProfile descr="" name="default-export" ownerKey="" ownerTag="">
            <rtctrlCtxP descr="" name="routecontrolpvtnw" order="3">
                <rtctrlScope descr="" name="">
                    <rtctrlRsScopeToAttrP tnRtctrlAttrPName="actionruleprofile2"/>
                </rtctrlScope>
            </rtctrlCtxP>
        </rtctrlProfile>
    </l3extOut>
```

Configuring Interleak Redistribution Using the REST API

The following procedure describes how to configure the interleak redistribution using the REST API.

Before you begin

Create the tenant, VRF, and L3Out.

Procedure

Step 1 Configure the route-map for interleak redistribution.

Example:

The following example configures a route map `INTERLEAK_RP` with two contexts (`ROUTES_A` and `ROUTES_ALL`). The first context `ROUTES_A` matches with an IP prefix-list 10.0.0.0/24 le 32 to set a community attribute via set rule `COM_A`. The second context matches with all routes.

```
POST: https://<APIC IP>/api/mo/uni.xml
BODY:
<fvTenant dn="uni/tn-SAMPLE">
```

```
    <!-- route map with two contexts (ROUTES_A and ROUTES_ALL)-->
    <rtctrlProfile type="global" name="INTERLEAK_RP">
        <rtctrlCtxP name="ROUTES_A" order="0" action="permit">
            <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="PFX_10-0-0-0_24"/>
            <rtctrlScope>
                <rtctrlRsScopeToAttrP tnRtctrlAttrPName="COM_A"/>
            </rtctrlScope>
        </rtctrlCtxP>
        <rtctrlCtxP name="ROUTES_ALL" order="9" action="permit">
            <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="ALL_PREFIX"/>
        </rtctrlCtxP>
    </rtctrlProfile>

    <!-- match rule with an IP prefix-list -->
    <rtctrlSubjP name="ALL_PREFIX">
        <rtctrlMatchRtDest ip="0.0.0.0/0" aggregate="yes"/>
    </rtctrlSubjP>

    <!-- match rule with an IP prefix-list -->
    <rtctrlSubjP name="PFX_10-0-0-0_24">
        <rtctrlMatchRtDest ip="10.0.0.0/24" aggregate="yes"/>
    </rtctrlSubjP>

    <!-- setu rule for community attribute -->
    <rtctrlAttrP name="COM_A">
        <rtctrlSetComm type="community" setCriteria="append" community="regular:as2-nn2:100:200"/>
    </rtctrlAttrP>
</fvTenant>
```

Step 2 Apply the configured route map to an L3Out.

The following example applies the route map from Step 1 to L3Out `l3out1` to customize interleak redistribution of routes from the given L3Out.

`L3extRsInterleakPol` is applied for dynamic routing protocol (OSPF/EIGRP) routes used by the given L3Out. `L3extRsRedistributePol` is applied for static routes, as specified by the `src` attribute (`static`).

Example:

```
POST: https://<APIC IP>/api/mo/uni.xml
BODY:
<fvTenant dn="uni/tn-SAMPLE">
    <l3extOut name="l3out1">
        <!-- interleak redistribution for OSPF/EIGRP routes -->
        <l3extRsInterleakPol tnRtctrlProfileName="INTERLEAK_RP"/>
        <!-- interleak redistribution for static routes -->
        <l3extRsRedistributePol tnRtctrlProfileName="INTERLEAK_RP" src="static"/>
    </l3extOut>
</fvTenant>
```

Configuring Transit Routing Using REST API

Configuring Transit Routing Using the REST API

These steps describe how to configure transit routing for a tenant. This example deploys two L3Outs, in one VRF, on two border leaf switches, that are each connected to a separate router.

Before you begin

- Configure the node, port, functional profile, AEP, and Layer 3 domain.
- Create the external routed domain and associate it to the interface for the L3Out.
- Configure a BGP route reflector policy to propagate the routes within the fabric.

Procedure

Step 1 Configure the tenant and VRF.

This example configures tenant `t1` and VRF `v1`. The VRF is not yet deployed.

Example:

```
<fvTenant  name="t1">
    <fvCtx name="v1"/>
</fvTenant>
```

Step 2 Configure the nodes and interfaces.

This example configures two L3Outs for the tenant t1 and VRF v1, on two border leaf switches. The VRF has a Layer 3 domain, `dom1`.

- The first L3Out is on node 101, which is named `nodep1`. Node 101 is configured with router ID `11.11.11.103`. It has a routed interface `ifp1` at `eth1/3`, with the IP address `12.12.12.3/24`.
- The second L3Out is on node 102, which is named `nodep2`. Node 102 is configured with router ID`22.22.22.203`. It has a routed interface `ifp2` at `eth1/3`, with the IP address, `23.23.23.1/24`.

Example:

```
<l3extOut name="l3out1">
    <l3extRsEctx tnFvCtxName="v1"/>
    <l3extLNodeP name="nodep1">
        <l3extRsNodeL3OutAtt rtrId="11.11.11.103" tDn="topology/pod-1/node-101"/>
        <l3extLIfP name="ifp1"/>
        <l3extRsPathL3OutAtt addr="12.12.12.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-101/pathep-[eth1/3]"/>
        </l3extLIfP>
    </l3extLNodeP>
    <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
</l3extOut>

<l3extOut name="l3out2">
    <l3extRsEctx tnFvCtxName="v1"/>
    <l3extLNodeP name="nodep2">
        <l3extRsNodeL3OutAtt rtrId="22.22.22.203" tDn="topology/pod-1/node-102"/>
        <l3extLIfP name="ifp2"/>
        <l3extRsPathL3OutAtt addr="23.23.23.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-102/pathep-[eth1/3]"/>
        </l3extLIfP>
    </l3extLNodeP>
    <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
</l3extOut>
```

Step 3 Configure the routing protocol for both border leaf switches.

Performing Tasks Using REST API

This example configures BGP as the primary routing protocol for both the border leaf switches, both with ASN 100. It also configures Node 101 with BGP peer `15.15.15.2` and node 102 with BGP peer `25.25.25.2`.

Example:

```
<l3extOut name="l3out1">
    <l3extLNodeP name="nodep1">
        <bgpPeerP addr="15.15.15.2/24"
            <bgpAsP asn="100"/>
        </bgpPeerP>
    </l3extLNodeP>
</l3extOut>

<l3extOut name="l3out2">
    <l3extLNodeP name="nodep2">
        <bgpPeerP addr="25.25.25.2/24"
            <bgpAsP asn="100"/>
        </bgpPeerP>
    </l3extLNodeP>
</l3extOut>
```

Step 4 Configure a connectivity routing protocol.

This example configures OSPF as the communication protocol, for both L3Outs, with regular area ID `0.0.0.0`.

Example:

```
<l3extOut name="l3out1">
    <ospfExtP areaId="0.0.0.0" areaType="regular"/>
    <l3extLNodeP name="nodep1">
        <l3extLIfP name="ifp1">
            <ospfIfP/>
        <l3extIfP>
    <l3extLNodeP>
</l3extOut>
<l3extOut name="l3out2">
    <ospfExtP areaId="0.0.0.0" areaType="regular"/>
    <l3extLNodeP name="nodep2">
        <l3extLIfP name="ifp2">
            <ospfIfP/>
        <l3extIfP>
    <l3extLNodeP>
</l3extOut>
```

Step 5 Configure the external EPGs.

This example configures the network `192.168.1.0/24` as external network `extnw1` on node 101 and `192.168.2.0/24` as external network `extnw2` on node 102. It also associates the external EPGs with the route control profiles `rp1` and `rp2`.

Example:

```
<l3extOut name="l3out1">
     <l3extInstP name="extnw1">
          <l3extSubnet ip="192.168.1.0/24" scope="import-security"/>
          <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp1"/>
    </l3extInstP>
</l3extOut>
<l3extOut name="l3out2">
     <l3extInstP name="extnw2">
          <l3extSubnet ip="192.168.2.0/24" scope="import-security"/>
          <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp2"/>
    </l3extInstP>
</l3extOut>
```

Step 6 Optional. Configure a route map.

This example configures a route map for each BGP peer in the inbound and outbound directions. For `l3out1`, the route map `rp1` is applied for routes that match an import destination of `192.168.1.0/24` and the route map `rp2` is applied for routes that match an export destination of `192.168.2.0/24`. For `l3out2`, the direction of the route maps is reversed.

Example:

```
<fvTenant name="t1">
    <rtctrlSubjP name="match-rule1">
        <rtctrlMatchRtDest ip="192.168.1.0/24" />
    </rtctrlSubjP>
    <rtctrlSubjP name="match-rule2">
        <rtctrlMatchRtDest ip="192.168.2.0/24" />
    </rtctrlSubjP>
    <l3extOut name="l3out1">
        <rtctrlProfile name="rp1">
            <rtctrlCtxP name="ctxp1" action="permit" order="0">
                <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1" />
            </rtctrlCtxP>
        </rtctrlProfile>
        <rtctrlProfile name="rp2">
            <rtctrlCtxP name="ctxp1" action="permit" order="0">
                <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule2" />
            </rtctrlCtxP>
        </rtctrlProfile>
        <l3extInstP name="extnw1">
            <l3extRsInstPToProfile direction="import" tnRtctrlProfileName="rp1" />
            <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp2" />
        </l3extInstP>
    </l3extOut>
    <l3extOut name="l3out2">
        <rtctrlProfile name="rp1">
            <rtctrlCtxP name="ctxp1" action="permit" order="0">
                <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1" />
            </rtctrlCtxP>
        </rtctrlProfile>
        <rtctrlProfile name="rp2">
            <rtctrlCtxP name="ctxp1" action="permit" order="0">
                <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule2" />
            </rtctrlCtxP>
        </rtctrlProfile>
        <l3extInstP name="extnw2">
            <l3extRsInstPToProfile direction="import" tnRtctrlProfileName="rp2" />
            <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp1" />
        </l3extInstP>
    </l3extOut>
</fvTenant>
```

Step 7 Create the filter and contract to enable the EPGs to communicate.

This example configures the filter `http-filter` and the contract `httpCtrct`. The external EPGs and the application EPGs are already associated with the contract `httpCtrct` as providers and consumers respectively.

Example:

```
<vzFilter name="http-filter">
    <vzEntry  name="http-e" etherT="ip" prot="tcp"/>
</vzFilter>
<vzBrCP name="httpCtrct" scope="context">
    <vzSubj name="subj1">
        <vzRsSubjFiltAtt tnVzFilterName="http-filter"/>
    </vzSubj>
</vzBrCP>
```

Step 8 Associate the external EPGs with the contract.

This example associates the external EPG `extnw1` as provider and external EPG `extnw2` as consumer of the contract `httpCtrct`.

```
 <l3extOut name="l3out1">
    <l3extInstP name="extnw1">
        <fvRsProv tnVzBrCPName="httpCtrct"/>
    </l3extInstP>
</l3extOut>
<l3extOut name="l3out2">
    <l3extInstP name="extnw2">
        <fvRsCons tnVzBrCPName="httpCtrct"/>
    </l3extInstP>
</l3extOut>
```

REST API Example: Transit Routing

The following example configures two L3Outs on two border leaf switches, using the REST API.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
    <fvTenant name="t1">
        <fvCtx name="v1"/>
        <l3extOut name="l3out1">
            <l3extRsEctx tnFvCtxName="v1"/>
            <l3extLNodeP name="nodep1">
                <bgpPeerP addr="15.15.15.2/24">
                    <bgpAsP asn="100"/>
                </bgpPeerP>
                <l3extRsNodeL3OutAtt rtrId="11.11.11.103" tDn="topology/pod-1/node-101"/>
                <l3extLIfP name="ifp1">
                    <l3extRsPathL3OutAtt addr="12.12.12.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-101/pathep-[eth1/3]" />
                    <ospfIfP/>
                </l3extLIfP>
            </l3extLNodeP>
            <l3extInstP name="extnw1">
                <l3extSubnet ip="192.168.1.0/24" scope="import-security"/>
                <l3extRsInstPToProfile direction="import" tnRtctrlProfileName="rp1"/>
                <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp2"/>
                <fvRsProv tnVzBrCPName="httpCtrct"/>
            </l3extInstP>
            <bgpExtP/>
            <ospfExtP areaId="0.0.0.0" areaType="regular"/>
            <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
            <rtctrlProfile name="rp1">
                <rtctrlCtxP name="ctxp1" action="permit" order="0">
                    <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1"/>
                </rtctrlCtxP>
            </rtctrlProfile>
            <rtctrlProfile name="rp2">
                <rtctrlCtxP name="ctxp1" action="permit" order="0">
                    <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule2"/>
                </rtctrlCtxP>
            </rtctrlProfile>
        </l3extOut>
        <l3extOut name="l3out2">
            <l3extRsEctx tnFvCtxName="v1"/>
            <l3extLNodeP name="nodep2">
```

```
                <bgpPeerP addr="25.25.25.2/24">
                    <bgpAsP asn="100"/>
                </bgpPeerP>
                <l3extRsNodeL3OutAtt rtrId="22.22.22.203" tDn="topology/pod-1/node-102" />
                <l3extLIfP name="ifp2">
                    <l3extRsPathL3OutAtt addr="23.23.23.3/24" ifInstT="l3-port"
tDn="topology/pod-1/paths-102/pathep-[eth1/3]" />
                    <ospfIfP/>
                </l3extLIfP>
            </l3extLNodeP>
            <l3extInstP name="extnw2">
                <l3extSubnet ip="192.168.2.0/24" scope="import-security"/>
                <l3extRsInstPToProfile direction="import" tnRtctrlProfileName="rp2"/>
                <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp1"/>
                <fvRsCons tnVzBrCPName="httpCtrct"/>
            </l3extInstP>
            <bgpExtP/>
            <ospfExtP areaId="0.0.0.0" areaType="regular"/>
            <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
            <rtctrlProfile name="rp1">
                <rtctrlCtxP name="ctxp1" action="permit" order="0">
                    <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1"/>
                </rtctrlCtxP>
            </rtctrlProfile>
            <rtctrlProfile name="rp2">
                <rtctrlCtxP name="ctxp1" action="permit" order="0">
                    <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule2"/>
                </rtctrlCtxP>
            </rtctrlProfile>
        </l3extOut>
        <rtctrlSubjP name="match-rule1">
            <rtctrlMatchRtDest ip="192.168.1.0/24"/>
        </rtctrlSubjP>
        <rtctrlSubjP name="match-rule2">
            <rtctrlMatchRtDest ip="192.168.2.0/24"/>
        </rtctrlSubjP>
        <vzFilter name="http-filter">
            <vzEntry name="http-e" etherT="ip" prot="tcp"/>
        </vzFilter>
        <vzBrCP name="httpCtrct" scope="context">
            <vzSubj name="subj1">
                <vzRsSubjFiltAtt tnVzFilterName="http-filter"/>
            </vzSubj>
        </vzBrCP>
    </fvTenant>
</polUni>
```

Shared L3Out

Configuring Shared Services Using REST API

Configuring Two Shared Layer 3 Outs in Two VRFs Using REST API

The following REST API configuration example that displays how two shared Layer 3 Outs in two VRFs communicate.

Procedure

Step 1 Configure the provider Layer 3 Out.

Example:

```
<tenant name="t1_provider">
<fvCtx name="VRF1">
<l3extOut name="T0-o1-L3OUT-1">
                <l3extRsEctx tnFvCtxName="o1"/>
                <ospfExtP areaId='60'/>
                <l3extInstP name="l3extInstP-1">
                <fvRsProv tnVzBrCPName="vzBrCP-1">
                </fvRsProv>
              <l3extSubnet ip="192.168.2.0/24" scope="shared-rtctrl, shared-security" aggregate=""/>
                </l3extInstP>
</l3extOut>
</tenant>
```

Step 2 Configure the consumer Layer 3 Out.

Example:

```
<tenant name="t1_consumer">
<fvCtx name="VRF2">
<l3extOut name="T0-o1-L3OUT-1">
                <l3extRsEctx tnFvCtxName="o1"/>
                <ospfExtP areaId='70'/>
                <l3extInstP name="l3extInstP-2">
                <fvRsCons tnVzBrCPName="vzBrCP-1">
                </fvRsCons>
                <l3extSubnet ip="199.16.2.0/24" scope="shared-rtctrl, shared-security"
aggregate=""/>
                </l3extInstP>
</l3extOut>
</tenant>
```

Configuring QoS for L3Outs Using REST API

Configuring QoS Directly on L3Out Using REST API

This section describes how to configure QoS directly on an L3Out. This is the preferred way of configuring L3Out QoS starting with Cisco APIC Release 4.0(1).

You can configure QoS for L3Out on one of the following objects:

- Switch Virtual Interface (SVI)
- Sub Interface
- Routed Outside

Procedure

Step 1 Configure QoS priorities for a L3Out SVI.

Example:

```
<l3extLIfP descr="" dn="uni/tn-DT/out-L3_4_2_24_SVI17/lnodep-L3_4_E2_24/lifp-L3_4_E2_24_SVI_19"
           name="L3_4_E2_24_SVI_19" prio="level6" tag="yellow-green">
    <l3extRsPathL3OutAtt addr="0.0.0.0" autostate="disabled" descr="SVI19" encap="vlan-19"
                         encapScope="local" ifInstT="ext-svi" ipv6Dad="enabled" llAddr="::"
                         mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
                         tDn="topology/pod-1/protpaths-103-104/pathep-[V_L3_l4_2-24]"
                         targetDscp="unspecified">
        <l3extMember addr="107.2.1.253/24" ipv6Dad="enabled" llAddr="::" side="B"/>
        <l3extMember addr="107.2.1.252/24" ipv6Dad="enabled" llAddr="::" side="A"/>
    </l3extRsPathL3OutAtt>
    <l3extRsLIfPCustQosPol tnQosCustomPolName="VrfQos006"/>
</l3extLIfP>
```

Step 2 Configure QoS priorities for a sub-interface.

Example:

```
<l3extLIfP dn="uni/tn-DT/out-L4E48_inter_tenant/lnodep-L4E48_inter_tenant/lifp-L4E48"
           name="L4E48" prio="level4" tag="yellow-green">
    <l3extRsPathL3OutAtt addr="210.1.0.254/16" autostate="disabled" encap="vlan-20"
                         encapScope="local" ifInstT="sub-interface" ipv6Dad="enabled" llAddr="::"
                         mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
                         tDn="topology/pod-1/paths-104/pathep-[eth1/48]" targetDscp="unspecified"/>
    <l3extRsNdIfPol annotation="" tnNdIfPolName=""/>
    <l3extRsLIfPCustQosPol annotation="" tnQosCustomPolName=" vrfQos002"/>
</l3extLIfP>
```

Step 3 Configure QoS priorities for a routed outside.

Example:

```
<l3extLIfP dn="uni/tn-DT/out-L2E37/lnodep-L2E37/lifp-L2E37OUT"
           name="L2E37OUT" prio="level5" tag="yellow-green">
    <l3extRsPathL3OutAtt addr="30.1.1.1/24" autostate="disabled" encap="unknown"
                         encapScope="local" ifInstT="l3-port" ipv6Dad="enabled"
                         llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular"
                         mtu="inherit" targetDscp="unspecified"
                         tDn="topology/pod-1/paths-102/pathep-[eth1/37]"/>
    <l3extRsNdIfPol annotation="" tnNdIfPolName=""/>
    <l3extRsLIfPCustQosPol tnQosCustomPolName="vrfQos002"/>
</l3extLIfP>
```

Configuring QoS Contract for L3Out Using REST API

This section describes how to configure QoS for L3Outs using Contracts.

Note Starting with Release 4.0(1), we recommend using custom QoS policies for L3Out QoS as described in Configuring QoS Directly on L3Out Using REST API, on page 58 instead.

Procedure

Step 1 When configuring the tenant, VRF, and bridge domain, configure the VRF for egress mode (`pcEnfDir="egress"`) with policy enforcement enabled (`pcEnfPref="enforced"`). Send a post with XML similar to the following example:

Example:

```
<fvTenant  name="t1">
    <fvCtx name="v1" pcEnfPref="enforced" pcEnfDir="egress"/>
    <fvBD name="bd1">
        <fvRsCtx tnFvCtxName="v1"/>
        <fvSubnet ip="44.44.44.1/24" scope="public"/>
        <fvRsBDToOut tnL3extOutName="l3out1"/>
    </fvBD>"/>
</fvTenant>
```

Step 2 When creating the filters and contracts to enable the EPGs participating in the L3Out to communicate, configure the QoS priority.

The contract in this example includes the QoS priority, `level1`, for traffic ingressing on the L3Out. Alternatively, it could define a target DSCP value. QoS policies are supported on either the contract or the subject.

The filter also has the `matchDscp="EF"` criteria, so that traffic with this specific TAG received by the L3out processes through the queue specified in the contract subject.

Note
VRF enforcement should be ingress, for QOS or custom QOS on L3out interface, VRF enforcement need be egress, only when the QOS classification is going to be done in the contract for traffic between EPG and L3out or L3out to L3out.

Note
If QOS classification is set in the contract and VRF enforcement is egress, then contract QOS classification would override the L3out interface QOS or Custom QOS classification, So either we need to configure this one or the new one.

Example:

```
<vzFilter name="http-filter">
    <vzEntry  name="http-e" etherT="ip" prot="tcp" matchDscp="EF"/>
</vzFilter>
<vzBrCP name="httpCtrct" prio="level1" scope="context">
    <vzSubj name="subj1">
        <vzRsSubjFiltAtt tnVzFilterName="http-filter"/>
    </vzSubj>
</vzBrCP>
```

Creating SR-MPLS Custom QoS Policy Using REST API

SR-MPLS Custom QoS policy defines the priority of the packets coming from an SR-MPLS network while they are inside the ACI fabric based on the incoming MPLS EXP values defined in the MPLS QoS ingress policy. It also marks the CoS and MPLS EXP values of the packets leaving the ACI fabric through an MPLS interface based on IPv4 DSCP values defined in MPLS QoS egress policy.

If no custom ingress policy is defined, the default QoS Level (`Level3`) is assigned to packets inside the fabric. If no custom egress policy is defined, the default EXP value of `0` will be marked on packets leaving the fabric.

Procedure

Step 1 Create SR-MPLS QoS policy.

In the following POST:

- Replace *customqos1* with the name of the SR-MPLS QoS policy you want to create.
- For the `qosMplsIngressRule`:
  - Replace `from="2" to="3"` with the EXP range you want the policy to match.
  - Replace `prio="level5"` with the ACI QoS Level for the packet while it's inside the ACI fabric.
  - Replace `target="CS5"` with the DSCP value you want to set on the packet when it's matched.
  - Replace `targetCos="4"` with the CoS value you want to set on the packet when it's matched.
- For the `qosMplsEgressRule`:
  - Replace `from="CS2" to="CS4"` with the DSCP range you want the policy to match.
  - Replace `targetExp="5"` with the EXP value you want to set on the packet when it's leaving the fabric.
  - Replace `targetCos="3"` with the CoS value you want to set on the packet when it's leaving the fabric.

```
<polUni>
  <fvTenant name="infra">
    <qosMplsCustomPol descr="" dn="uni/tn-infra/qosmplscustom-customqos1" name="customqos1"  status=""
 >
        <qosMplsIngressRule from="2" to="3" prio="level5" target="CS5" targetCos="4" status="" />
        <qosMplsEgressRule from="CS2" to="CS4" targetExp="5" targetCos="3" status=""/>
    </qosMplsCustomPol>
  </fvTenant>
</polUni>
```

Step 2 Applying SR-MPLS QoS policy.

In the following POST, replace *customqos1* with the name of the SR-MPLS QoS policy you created in the previous step.

```
<polUni>
    <fvTenant name="infra">
        <l3extOut name="mplsOut" status="" descr="bl">
            <l3extLNodeP name="mplsLNP" status="">
                <l3extRsLNodePMplsCustQosPol  tDn="uni/tn-infra/qosmplscustom-customqos1"/>
            </l3extLNodeP>
        </l3extOut>
    </fvTenant>
</polUni>
```

Configuring ACI IP SLAs Using REST API

Configuring an IP SLA Monitoring Policy Using the REST API

To enable Cisco APIC to send monitoring probes for a specific SLA type using REST API, perform the following steps:

Procedure

Configure an IP SLA monitoring policy.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<imdata totalCount="1">
                <fvIPSLAMonitoringPol annotation="" descr=""
dn="uni/tn-t8/ipslaMonitoringPol-ICMP-Probe"
                 name="ICMP-Probe" nameAlias="" ownerKey="" ownerTag="" slaDetectMultiplier="3"
slaFrequency="5"
                 slaPort="0" slaType="icmp"/>
</imdata>
```

Configuring an IP-SLA Track Member Using the REST API

To configure an IP SLA track member using REST API, perform the following steps:

Procedure

Configure an IP SLA track member.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<imdata totalCount="1">
                <fvTrackMember annotation="" descr="" dn="uni/tn-t8/trackmember-TM_pc_sub"
                dstIpAddr="52.52.52.1" name="TM_pc_sub" nameAlias="" ownerKey="" ownerTag=""
                scopeDn="uni/tn-t8/out-t8_l3">
                      <fvRsIpslaMonPol annotation="" tDn="uni/tn-t8/ipslaMonitoringPol-TCP-Telnet"/>

                </fvTrackMember>
</imdata>
```

Configuring an IP-SLA Track List Using the REST API

To configure an IP SLA track list using REST API, perform the following steps:

Procedure

Configure an IP SLA track list.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<imdata totalCount="1">
                <fvTrackList annotation="" descr="" dn="uni/tn-t8/tracklist-T8_pc_sub1"
                name="T8_pc_sub1" nameAlias="" ownerKey="" ownerTag="" percentageDown="0"
                percentageUp="1" type="weight" weightDown="5" weightUp="10">
                         <fvRsOtmListMember annotation="" tDn="uni/tn-t8/trackmember-TM_pc_sub"
                         weight="10"/>
                </fvTrackList>
</imdata>
```

Associating a Track List with a Static Route Using the REST API

To associate an IP SLA track list with a static route using REST API, perform the following steps:

Procedure

Associate an IP SLA track list with a static route.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<imdata totalCount="1">
                <ipRouteP aggregate="no" annotation="" descr=""

dn="uni/tn-t8/out-t8_l3/lnodep-t8_l3_vpc1/rsnodeL3OutAtt-[topology/pod-2/node-108]/rt-[88.88.88.2/24]"

                ip="88.88.88.2/24" name="" nameAlias="" pref="1" rtCtrl="">
                         <ipRsRouteTrack annotation="" tDn="uni/tn-t8/tracklist-T8_TL1_Static"/>
                         <ipNexthopP annotation="" descr="" name="" nameAlias="" nhAddr="23.23.2.3"

                         pref="1" type="prefix"/>
                </ipRouteP>
</imdata>
```

Associating a Track List with a Next Hop Profile Using the REST API

To associate an IP SLA track list with a next hop profile using REST API, perform the following steps:

Procedure

Associate an IP SLA track list with a next hop profile.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<imdata totalCount="1">
                <ipRouteP aggregate="no" annotation="" descr=""

dn="uni/tn-t8/out-t8_l3/lnodep-t8_l3_vpc1/rsnodeL3OutAtt-[topology/pod-2/node-109]/rt-[86.86.86.2/24]"

                ip="86.86.86.2/24" name="" nameAlias="" pref="1" rtCtrl="">
                        <ipNexthopP annotation="" descr="" name="" nameAlias="" nhAddr="25.25.25.3"
 pref="1" type="prefix">
                                <ipRsNexthopRouteTrack annotation=""
tDn="uni/tn-t8/tracklist-ctx0_25.25.25.3"/>
                                <ipRsNHTrackMember annotation=""
tDn="uni/tn-t8/trackmember-ctx0_25.25.25.3"/>
                        </ipNexthopP>
                </ipRouteP>
</imdata>
```

Configuring HSRP Using REST API

Configuring HSRP in APIC Using REST API

HSRP is enabled when the leaf switch is configured.

Before you begin

- The tenant and VRF must be configured.
- VLAN pools must be configured with the appropriate VLAN range defined and the appropriate Layer 3 domain created and attached to the VLAN pool.
- The Attach Entity Profile must also be associated with the Layer 3 domain.
- The interface profile for the leaf switches must be configured as required.

Procedure

Step 1 Create port selectors.

Example:

```
<polUni>
  <infraInfra dn="uni/infra">
    <infraNodeP name="TenantNode_101">
      <infraLeafS name="leafselector" type="range">
        <infraNodeBlk name="nodeblk" from_="101" to_="101">
            </infraNodeBlk>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-TenantPorts_101"/>
    </infraNodeP>
    <infraAccPortP name="TenantPorts_101">
      <infraHPortS name="portselector" type="range">
        <infraPortBlk name="portblk" fromCard="1" toCard="1" fromPort="41" toPort="41">
            </infraPortBlk>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-TenantPortGrp_101"/>
      </infraHPortS>
```

```
    </infraAccPortP>
    <infraFuncP>
      <infraAccPortGrp name="TenantPortGrp_101">
        <infraRsAttEntP tDn="uni/infra/attentp-AttEntityProfTenant"/>
        <infraRsHIfPol tnFabricHIfPolName="default"/>
      </infraAccPortGrp>
    </infraFuncP>
  </infraInfra>
</polUni>
```

Step 2 Create a tenant policy.

Example:

```
<polUni>
  <fvTenant name="t9" dn="uni/tn-t9" descr="">
    <fvCtx name="t9_ctx1" pcEnfPref="unenforced">
    </fvCtx>
    <fvBD name="t9_bd1" unkMacUcastAct="flood" arpFlood="yes">
      <fvRsCtx tnFvCtxName="t9_ctx1"/>
      <fvSubnet ip="101.9.1.1/24" scope="shared"/>
    </fvBD>
    <l3extOut dn="uni/tn-t9/out-l3extOut1" enforceRtctrl="export" name="l3extOut1">
      <l3extLNodeP name="Node101">
       <l3extRsNodeL3OutAtt rtrId="210.210.121.121" rtrIdLoopBack="no" tDn="topology/pod-1/node-101"/>

      </l3extLNodeP>
      <l3extRsEctx tnFvCtxName="t9_ctx1"/>
      <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
      <l3extInstP matchT="AtleastOne" name="extEpg" prio="unspecified" targetDscp="unspecified">
        <l3extSubnet aggregate="" descr="" ip="176.21.21.21/21" name="" scope="import-security"/>
      </l3extInstP>
    </l3extOut>
  </fvTenant>
</polUni>
```

Step 3 Create an HSRP interface policy.

Example:

```
<polUni>
  <fvTenant name="t9" dn="uni/tn-t9" descr="">
    <hsrpIfPol name="hsrpIfPol" ctrl="bfd" delay="4" reloadDelay="11"/>
  </fvTenant>
</polUni>
```

Step 4 Create an HSRP group policy.

Example:

```
<polUni>
  <fvTenant name="t9" dn="uni/tn-t9" descr="">
    <hsrpIfPol name="hsrpIfPol" ctrl="bfd" delay="4" reloadDelay="11"/>
  </fvTenant>
</polUni>
```

Step 5 Create an HSRP interface profile and an HSRP group profile.

Example:

```
<polUni>
  <fvTenant name="t9" dn="uni/tn-t9" descr="">
    <l3extOut dn="uni/tn-t9/out-l3extOut1" enforceRtctrl="export" name="l3extOut1">
      <l3extLNodeP name="Node101">
```

```
        <l3extLIfP name="eth1-41-v6" ownerKey="" ownerTag="" tag="yellow-green">
          <hsrpIfP name="eth1-41-v6" version="v2">
            <hsrpRsIfPol tnHsrpIfPolName="hsrpIfPol"/>
            <hsrpGroupP descr="" name="HSRPV6-2" groupId="330" groupAf="ipv6" ip="fe80::3"
mac="00:00:0C:18:AC:01" ipObtainMode="admin">
              <hsrpRsGroupPol tnHsrpGroupPolName="G1"/>
            </hsrpGroupP>
          </hsrpIfP>
          <l3extRsPathL3OutAtt addr="2002::100/64" descr="" encap="unknown" encapScope="local"
ifInstT="l3-port" llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
tDn="topology/pod-1/paths-101/pathep-[eth1/41]" targetDscp="unspecified">
            <l3extIp addr="2004::100/64"/>
          </l3extRsPathL3OutAtt>
        </l3extLIfP>
        <l3extLIfP name="eth1-41-v4" ownerKey="" ownerTag="" tag="yellow-green">
          <hsrpIfP name="eth1-41-v4" version="v1">
            <hsrpRsIfPol tnHsrpIfPolName="hsrpIfPol"/>
            <hsrpGroupP descr="" name="HSRPV4-2" groupId="51" groupAf="ipv4" ip="177.21.21.21"
mac="00:00:0C:18:AC:01" ipObtainMode="admin">
              <hsrpRsGroupPol tnHsrpGroupPolName="G1"/>
            </hsrpGroupP>
          </hsrpIfP>
          <l3extRsPathL3OutAtt addr="177.21.21.11/24" descr="" encap="unknown" encapScope="local"
ifInstT="l3-port" llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
tDn="topology/pod-1/paths-101/pathep-[eth1/41]" targetDscp="unspecified">
            <l3extIp addr="177.21.23.11/24"/>
          </l3extRsPathL3OutAtt>
        </l3extLIfP>
      </l3extLNodeP>
    </l3extOut>
  </fvTenant>
</polUni>
```

Configuring Cisco ACI GOLF Using REST API

Configuring GOLF Using the REST API

SUMMARY STEPS

1. The following example shows how to deploy nodes and spine switch interfaces for GOLF, using the REST API:
2. The XML below configures the spine switch interfaces and infra tenant provider of the GOLF service. Include this XML structure in the body of the POST message.
3. The XML below configures the tenant consumer of the infra part of the GOLF service. Include this XML structure in the body of the POST message.

DETAILED STEPS

Procedure

Step 1 The following example shows how to deploy nodes and spine switch interfaces for GOLF, using the REST API:

Example:

```
POST
https://192.0.20.123/api/mo/uni/golf.xml
```

Step 2 The XML below configures the spine switch interfaces and infra tenant provider of the GOLF service. Include this XML structure in the body of the POST message.

Example:

```
<l3extOut descr="" dn="uni/tn-infra/out-golf" enforceRtctrl="export,import"
    name="golf"
    ownerKey="" ownerTag="" targetDscp="unspecified">
    <l3extRsEctx tnFvCtxName="overlay-1"/>
    <l3extProvLbl descr="" name="golf"
         ownerKey="" ownerTag="" tag="yellow-green"/>
    <l3extLNodeP configIssues="" descr=""
         name="bLeaf" ownerKey="" ownerTag=""
         tag="yellow-green" targetDscp="unspecified">
         <l3extRsNodeL3OutAtt rtrId="10.10.3.3" rtrIdLoopBack="no"
             tDn="topology/pod-1/node-111">
             <l3extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
             <l3extLoopBackIfP addr="10.10.3.3" descr="" name=""/>
         </l3extRsNodeL3OutAtt>
         <l3extRsNodeL3OutAtt rtrId="10.10.3.4" rtrIdLoopBack="no"
             tDn="topology/pod-1/node-112">
         <l3extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
         <l3extLoopBackIfP addr="10.10.3.4" descr="" name=""/>
         </l3extRsNodeL3OutAtt>
         <l3extLIfP descr="" name="portIf-spine1-3"
             ownerKey="" ownerTag="" tag="yellow-green">
             <ospfIfP authKeyId="1" authType="none" descr="" name="">
               <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
             </ospfIfP>
             <l3extRsNdIfPol tnNdIfPolName=""/>
             <l3extRsIngressQosDppPol tnQosDppPolName=""/>
             <l3extRsEgressQosDppPol tnQosDppPolName=""/>
             <l3extRsPathL3OutAtt addr="7.2.1.1/24" descr=""
                encap="vlan-4"
                encapScope="local"
                ifInstT="sub-interface"
                llAddr="::" mac="00:22:BD:F8:19:FF"
                mode="regular"
                mtu="1500"
                tDn="topology/pod-1/paths-111/pathep-[eth1/12]"
                targetDscp="unspecified"/>
          </l3extLIfP>
          <l3extLIfP descr="" name="portIf-spine2-1"
              ownerKey=""
              ownerTag=""
              tag="yellow-green">
              <ospfIfP authKeyId="1"
                   authType="none"
                   descr=""
                   name="">
                   <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
              </ospfIfP>
              <l3extRsNdIfPol tnNdIfPolName=""/>
              <l3extRsIngressQosDppPol tnQosDppPolName=""/>
              <l3extRsEgressQosDppPol tnQosDppPolName=""/>
              <l3extRsPathL3OutAtt addr="7.1.0.1/24" descr=""
                   encap="vlan-4"
                   encapScope="local"
                   ifInstT="sub-interface"
                   llAddr="::" mac="00:22:BD:F8:19:FF"
                   mode="regular"
```

Performing Tasks Using REST API

```
        mtu="9000"
        tDn="topology/pod-1/paths-112/pathep-[eth1/11]"
        targetDscp="unspecified"/>
</l3extLIfP>
<l3extLIfP descr="" name="portif-spine2-2"
   ownerKey=""
   ownerTag=""
   tag="yellow-green">
   <ospfIfP authKeyId="1"
        authType="none" descr=""
        name="">
        <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
  </ospfIfP>
  <l3extRsNdIfPol tnNdIfPolName=""/>
  <l3extRsIngressQosDppPol tnQosDppPolName=""/>
  <l3extRsEgressQosDppPol tnQosDppPolName=""/>
  <l3extRsPathL3OutAtt addr="7.2.2.1/24" descr=""
        encap="vlan-4"
        encapScope="local"
        ifInstT="sub-interface"
               llAddr="::" mac="00:22:BD:F8:19:FF"
               mode="regular"
               mtu="1500"
               tDn="topology/pod-1/paths-112/pathep-[eth1/12]"
               targetDscp="unspecified"/>
  </l3extLIfP>
  <l3extLIfP descr="" name="portIf-spine1-2"
       ownerKey="" ownerTag="" tag="yellow-green">
       <ospfIfP authKeyId="1" authType="none" descr="" name="">
            <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
       </ospfIfP>
       <l3extRsNdIfPol tnNdIfPolName=""/>
       <l3extRsIngressQosDppPol tnQosDppPolName=""/>
       <l3extRsEgressQosDppPol tnQosDppPolName=""/>
       <l3extRsPathL3OutAtt addr="9.0.0.1/24" descr=""
        encap="vlan-4"
        encapScope="local"
        ifInstT="sub-interface"
             llAddr="::" mac="00:22:BD:F8:19:FF"
             mode="regular"
             mtu="9000"
             tDn="topology/pod-1/paths-111/pathep-[eth1/11]"
             targetDscp="unspecified"/>
  </l3extLIfP>
  <l3extLIfP descr="" name="portIf-spine1-1"
        ownerKey="" ownerTag="" tag="yellow-green">
        <ospfIfP authKeyId="1" authType="none" descr="" name="">
             <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
        </ospfIfP>
        <l3extRsNdIfPol tnNdIfPolName=""/>
        <l3extRsIngressQosDppPol tnQosDppPolName=""/>
        <l3extRsEgressQosDppPol tnQosDppPolName=""/>
        <l3extRsPathL3OutAtt addr="7.0.0.1/24" descr=""
          encap="vlan-4"
          encapScope="local"
          ifInstT="sub-interface"
          llAddr="::" mac="00:22:BD:F8:19:FF"
          mode="regular"
          mtu="1500"
          tDn="topology/pod-1/paths-111/pathep-[eth1/10]"
               targetDscp="unspecified"/>
  </l3extLIfP>
  <bgpInfraPeerP addr="10.10.3.2"
     allowedSelfAsCnt="3"
```

```
                ctrl="send-com,send-ext-com"
                descr="" name="" peerCtrl=""
                peerT="wan"
                privateASctrl="" ttl="2" weight="0">
                <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
                <bgpAsP asn="150" descr="" name="aspn"/>
             </bgpInfraPeerP>
             <bgpInfraPeerP addr="10.10.4.1"
                allowedSelfAsCnt="3"
                ctrl="send-com,send-ext-com" descr="" name="" peerCtrl=""
                peerT="wan"
                privateASctrl="" ttl="1" weight="0">
                <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
                <bgpAsP asn="100" descr="" name=""/>
              </bgpInfraPeerP>
              <bgpInfraPeerP addr="10.10.3.1"
                allowedSelfAsCnt="3"
                ctrl="send-com,send-ext-com" descr="" name="" peerCtrl=""
                peerT="wan"
                privateASctrl="" ttl="1" weight="0">
                <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
                <bgpAsP asn="100" descr="" name=""/>
             </bgpInfraPeerP>
       </l3extLNodeP>
       <bgpRtTargetInstrP descr="" name="" ownerKey="" ownerTag="" rtTargetT="explicit"/>
       <l3extRsL3DomAtt tDn="uni/l3dom-l3dom"/>
       <l3extInstP descr="" matchT="AtleastOne" name="golfInstP"
                 prio="unspecified"
                 targetDscp="unspecified">
                 <fvRsCustQosPol tnQosCustomPolName=""/>
        </l3extInstP>
        <bgpExtP descr=""/>
        <ospfExtP areaCost="1"
               areaCtrl="redistribute,summary"
               areaId="0.0.0.1"
               areaType="regular" descr=""/>
</l3extOut>
```

Step 3 The XML below configures the tenant consumer of the infra part of the GOLF service. Include this XML structure in the body of the POST message.

Example:

```
<fvTenant descr="" dn="uni/tn-pep6" name="pep6" ownerKey="" ownerTag="">
     <vzBrCP descr="" name="webCtrct"
          ownerKey="" ownerTag="" prio="unspecified"
          scope="global" targetDscp="unspecified">
          <vzSubj consMatchT="AtleastOne" descr=""
               name="http" prio="unspecified" provMatchT="AtleastOne"
               revFltPorts="yes" targetDscp="unspecified">
               <vzRsSubjFiltAtt directives="" tnVzFilterName="default"/>
          </vzSubj>
      </vzBrCP>
      <vzBrCP descr="" name="webCtrct-pod2"
           ownerKey="" ownerTag="" prio="unspecified"
           scope="global" targetDscp="unspecified">
           <vzSubj consMatchT="AtleastOne" descr=""
                name="http" prio="unspecified"
                provMatchT="AtleastOne" revFltPorts="yes"
                targetDscp="unspecified">
                <vzRsSubjFiltAtt directives=""
                      tnVzFilterName="default"/>
          </vzSubj>
```

Performing Tasks Using REST API

```
</vzBrCP>
<fvCtx descr="" knwMcastAct="permit"
     name="ctx6" ownerKey="" ownerTag=""
     pcEnfDir="ingress" pcEnfPref="enforced">
     <bgpRtTargetP af="ipv6-ucast"
          descr="" name="" ownerKey="" ownerTag="">
          <bgpRtTarget descr="" name="" ownerKey="" ownerTag=""
          rt="route-target:as4-nn2:100:1256"
          type="export"/>
          <bgpRtTarget descr="" name="" ownerKey="" ownerTag=""
               rt="route-target:as4-nn2:100:1256"
               type="import"/>
     </bgpRtTargetP>
     <bgpRtTargetP af="ipv4-ucast"
          descr="" name="" ownerKey="" ownerTag="">
          <bgpRtTarget descr="" name="" ownerKey="" ownerTag=""
               rt="route-target:as4-nn2:100:1256"
               type="export"/>
          <bgpRtTarget descr="" name="" ownerKey="" ownerTag=""
               rt="route-target:as4-nn2:100:1256"
               type="import"/>
     </bgpRtTargetP>
     <fvRsCtxToExtRouteTagPol tnL3extRouteTagPolName=""/>
     <fvRsBgpCtxPol tnBgpCtxPolName=""/>
     <vzAny descr="" matchT="AtleastOne" name=""/>
     <fvRsOspfCtxPol tnOspfCtxPolName=""/>
     <fvRsCtxToEpRet tnFvEpRetPolName=""/>
     <l3extGlobalCtxName descr="" name="dci-pep6"/>
</fvCtx>
<fvBD arpFlood="no" descr="" epMoveDetectMode=""
     ipLearning="yes"
     limitIpLearnToSubnets="no"
     llAddr="::" mac="00:22:BD:F8:19:FF"
     mcastAllow="no"
     multiDstPktAct="bd-flood"
     name="bd107" ownerKey="" ownerTag="" type="regular"
     unicastRoute="yes"
     unkMacUcastAct="proxy"
     unkMcastAct="flood"
     vmac="not-applicable">
     <fvRsBDToNdP tnNdIfPolName=""/>
     <fvRsBDToOut tnL3extOutName="routAccounting-pod2"/>
     <fvRsCtx tnFvCtxName="ctx6"/>
     <fvRsIgmpsn tnIgmpSnoopPolName=""/>
     <fvSubnet ctrl="" descr="" ip="27.6.1.1/24"
          name="" preferred="no"
          scope="public"
          virtual="no"/>
          <fvSubnet ctrl="nd" descr="" ip="2001:27:6:1::1/64"
               name="" preferred="no"
               scope="public"
               virtual="no">
               <fvRsNdPfxPol tnNdPfxPolName=""/>
          </fvSubnet>
          <fvRsBdToEpRet resolveAct="resolve" tnFvEpRetPolName=""/>
</fvBD>
<fvBD arpFlood="no" descr="" epMoveDetectMode=""
     ipLearning="yes"
     limitIpLearnToSubnets="no"
     llAddr="::" mac="00:22:BD:F8:19:FF"
     mcastAllow="no"
     multiDstPktAct="bd-flood"
     name="bd103" ownerKey="" ownerTag="" type="regular"
     unicastRoute="yes"
```

Performing Tasks Using REST API

```
     unkMacUcastAct="proxy"
     unkMcastAct="flood"
     vmac="not-applicable">
     <fvRsBDToNdP tnNdIfPolName=""/>
     <fvRsBDToOut tnL3extOutName="routAccounting"/>
     <fvRsCtx tnFvCtxName="ctx6"/>
     <fvRsIgmpsn tnIgmpSnoopPolName=""/>
     <fvSubnet ctrl="" descr="" ip="23.6.1.1/24"
          name="" preferred="no"
          scope="public"
          virtual="no"/>
     <fvSubnet ctrl="nd" descr="" ip="2001:23:6:1::1/64"
          name="" preferred="no"
          scope="public" virtual="no">
          <fvRsNdPfxPol tnNdPfxPolName=""/>
     </fvSubnet>
     <fvRsBdToEpRet resolveAct="resolve" tnFvEpRetPolName=""/>
</fvBD>
<vnsSvcCont/>
<fvRsTenantMonPol tnMonEPGPolName=""/>
<fvAp descr="" name="AP1"
     ownerKey="" ownerTag="" prio="unspecified">
     <fvAEPg descr=""
          isAttrBasedEPg="no"
          matchT="AtleastOne"
          name="epg107"
          pcEnfPref="unenforced" prio="unspecified">
          <fvRsCons prio="unspecified"
               tnVzBrCPName="webCtrct-pod2"/>
          <fvRsPathAtt descr=""
               encap="vlan-1256"
               instrImedcy="immediate"
               mode="regular" primaryEncap="unknown"
               tDn="topology/pod-2/paths-107/pathep-[eth1/48]"/>
          <fvRsDomAtt classPref="encap" delimiter=""
               encap="unknown"
               instrImedcy="immediate"
               primaryEncap="unknown"
               resImedcy="lazy" tDn="uni/phys-phys"/>
          <fvRsCustQosPol tnQosCustomPolName=""/>
          <fvRsBd tnFvBDName="bd107"/>
          <fvRsProv matchT="AtleastOne"
               prio="unspecified"
               tnVzBrCPName="default"/>
     </fvAEPg>
     <fvAEPg descr=""
          isAttrBasedEPg="no"
          matchT="AtleastOne"
          name="epg103"
          pcEnfPref="unenforced" prio="unspecified">
          <fvRsCons prio="unspecified" tnVzBrCPName="default"/>
          <fvRsCons prio="unspecified" tnVzBrCPName="webCtrct"/>
          <fvRsPathAtt descr="" encap="vlan-1256"
               instrImedcy="immediate"
               mode="regular" primaryEncap="unknown"
               tDn="topology/pod-1/paths-103/pathep-[eth1/48]"/>
               <fvRsDomAtt classPref="encap" delimiter=""
                    encap="unknown"
                    instrImedcy="immediate"
                    primaryEncap="unknown"
                    resImedcy="lazy" tDn="uni/phys-phys"/>
               <fvRsCustQosPol tnQosCustomPolName=""/>
               <fvRsBd tnFvBDName="bd103"/>
     </fvAEPg>
```

```
        </fvAp>
        <l3extOut descr=""
             enforceRtctrl="export"
             name="routAccounting-pod2"
             ownerKey="" ownerTag="" targetDscp="unspecified">
             <l3extRsEctx tnFvCtxName="ctx6"/>
             <l3extInstP descr=""
                  matchT="AtleastOne"
                  name="accountingInst-pod2"
                  prio="unspecified" targetDscp="unspecified">
             <l3extSubnet aggregate="export-rtctrl,import-rtctrl"
                  descr="" ip="::/0" name=""
                  scope="export-rtctrl,import-rtctrl,import-security"/>
             <l3extSubnet aggregate="export-rtctrl,import-rtctrl"
                  descr=""
                  ip="0.0.0.0/0" name=""
                  scope="export-rtctrl,import-rtctrl,import-security"/>
             <fvRsCustQosPol tnQosCustomPolName=""/>
             <fvRsProv matchT="AtleastOne"
                  prio="unspecified" tnVzBrCPName="webCtrct-pod2"/>
             </l3extInstP>
             <l3extConsLbl descr=""
                  name="golf2"
                  owner="infra"
                  ownerKey="" ownerTag="" tag="yellow-green"/>
        </l3extOut>
        <l3extOut descr=""
             enforceRtctrl="export"
             name="routAccounting"
             ownerKey="" ownerTag="" targetDscp="unspecified">
             <l3extRsEctx tnFvCtxName="ctx6"/>
             <l3extInstP descr=""
                  matchT="AtleastOne"
                  name="accountingInst"
                  prio="unspecified" targetDscp="unspecified">
             <l3extSubnet aggregate="export-rtctrl,import-rtctrl" descr=""
                  ip="0.0.0.0/0" name=""
                  scope="export-rtctrl,import-rtctrl,import-security"/>
             <fvRsCustQosPol tnQosCustomPolName=""/>
             <fvRsProv matchT="AtleastOne" prio="unspecified" tnVzBrCPName="webCtrct"/>
             </l3extInstP>
             <l3extConsLbl descr=""
                  name="golf"
                  owner="infra"
                  ownerKey="" ownerTag="" tag="yellow-green"/>
        </l3extOut>
</fvTenant>
```

Enabling Distributing BGP EVPN Type-2 Host Routes to a DCIG Using the REST API

Enable distributing BGP EVPN type-2 host routes using the REST API, as follows:

Before you begin

EVPN services must be configured.

Procedure

Step 1 Configure the Host Route Leak policy, with a POST containing XML such as in the following example:

Example:

```
<bgpCtxAfPol descr="" ctrl="host-rt-leak" name="bgpCtxPol_0 status=""/>
```

Step 2 Apply the policy to the VRF BGP Address Family Context Policy for one or both of the address families using a POST containing XML such as in the following example:

Example:

```
<fvCtx name="vni-10001">
<fvRsCtxToBgpCtxAfPol af="ipv4-ucast" tnBgpCtxAfPolName="bgpCtxPol_0"/>
<fvRsCtxToBgpCtxAfPol af="ipv6-ucast" tnBgpCtxAfPolName="bgpCtxPol_0"/>
</fvCtx>
```

Performing Tasks Using REST API