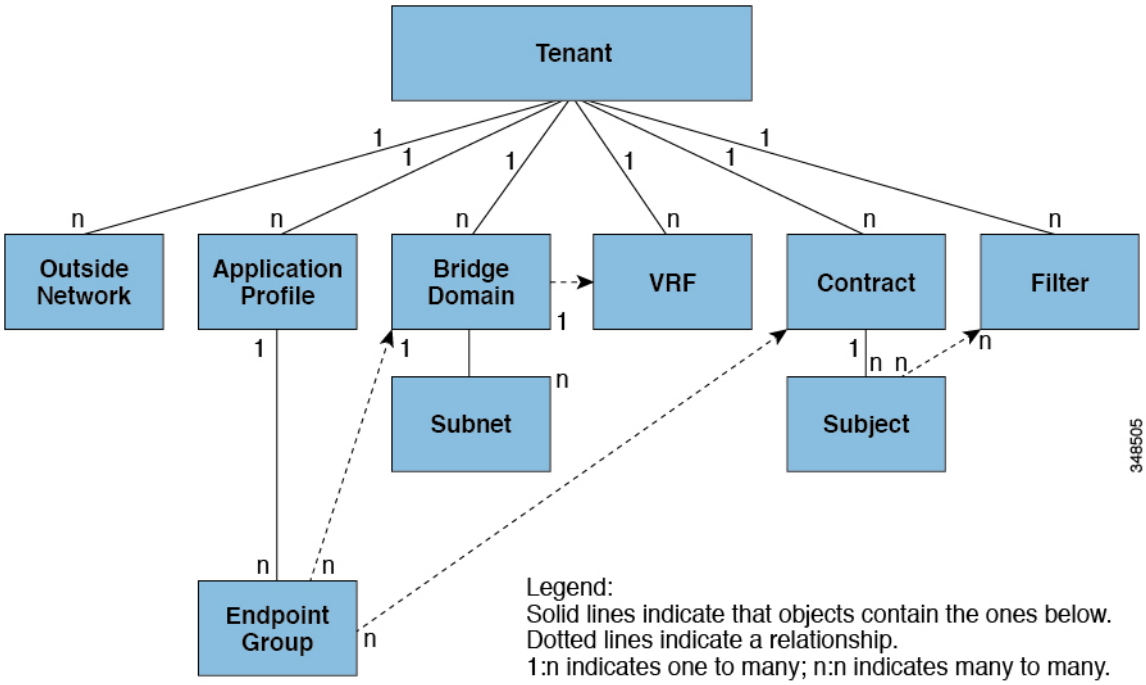
Basic User Tenant Configuration

This chapter contains the following sections:

Tenants

A tenant (`fvTenant`) is a logical container for application policies that enable an administrator to exercise domain-based access control. A tenant represents a unit of isolation from a policy perspective, but it does not represent a private network. Tenants can represent a customer in a service provider setting, an organization or domain in an enterprise setting, or just a convenient grouping of policies. The following figure provides an overview of the tenant portion of the management information tree (MIT).

Figure 1: Tenants



Tenants can be isolated from one another or can share resources. The primary elements that the tenant contains are filters, contracts, outside networks, bridge domains, Virtual Routing and Forwarding (VRF) instances, and application profiles that contain endpoint groups (EPGs). Entities in the tenant inherit its policies. VRFs are also known as contexts; each VRF can be associated with multiple bridge domains.

Note In the APIC GUI under the tenant navigation path, a VRF (context) is called a private network.

Tenants are logical containers for application policies. The fabric can contain multiple tenants. You must configure a tenant before you can deploy any Layer 4 to Layer 7 services. The ACI fabric supports IPv4, IPv6, and dual-stack configurations for tenant networking.

Routing Within the Tenant

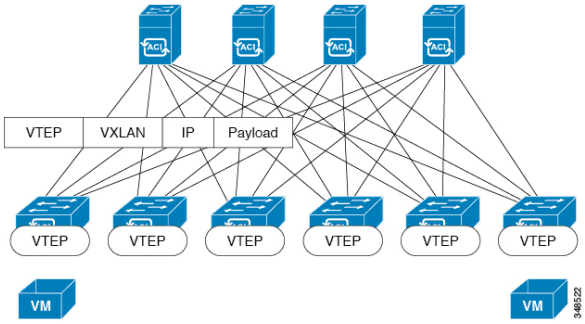
The Application Centric Infrastructure (ACI) fabric provides tenant default gateway functionality and routes between the fabric virtual extensible local area (VXLAN) networks. For each tenant, the fabric provides a virtual default gateway or Switched Virtual Interface (SVI) whenever a subnet is created on the APIC. This spans any switch that has a connected endpoint for that tenant subnet. Each ingress interface supports the default gateway interface and all of the ingress interfaces across the fabric share the same router IP address and MAC address for a given tenant subnet.

Layer 3 VNIDs Facilitate Transporting Inter-subnet Tenant Traffic

The ACI fabric provides tenant default gateway functionality that routes between the ACI fabric VXLAN networks. For each tenant, the fabric provides a virtual default gateway that spans all of the leaf switches assigned to the tenant. It does this at the ingress interface of the first leaf switch connected to the endpoint. Each ingress interface supports the default gateway interface. All of the ingress interfaces across the fabric share the same router IP address and MAC address for a given tenant subnet.

The ACI fabric decouples the tenant endpoint address, its identifier, from the location of the endpoint that is defined by its locator or VXLAN tunnel endpoint (VTEP) address. Forwarding within the fabric is between VTEPs. The following figure shows decoupled identity and location in ACI.

Figure 2: ACI Decouples Identity and Location



VXLAN uses VTEP devices to map tenant end devices to VXLAN segments and to perform VXLAN encapsulation and de-encapsulation. Each VTEP function has two interfaces:

- A switch interface on the local LAN segment to support local endpoint communication through bridging
- An IP interface to the transport IP network

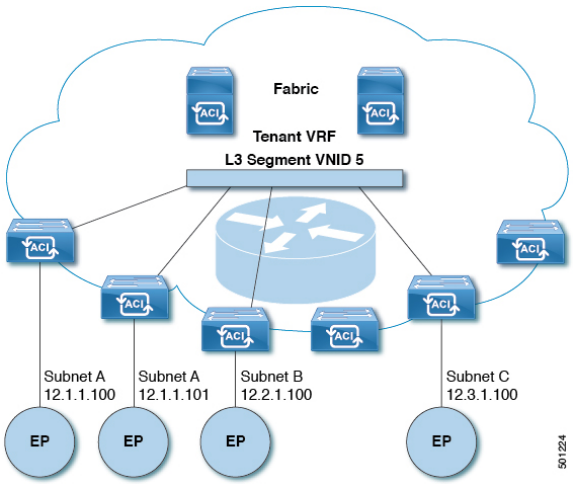
The IP interface has a unique IP address that identifies the VTEP device on the transport IP network known as the infrastructure VLAN. The VTEP device uses this IP address to encapsulate Ethernet frames and transmit the encapsulated packets to the transport network through the IP interface. A VTEP device also discovers the remote VTEPs for its VXLAN segments and learns remote MAC Address-to-VTEP mappings through its IP interface.

The VTEP in ACI maps the internal tenant MAC or IP address to a location using a distributed mapping database. After the VTEP completes a lookup, the VTEP sends the original data packet encapsulated in VXLAN with the destination address of the VTEP on the destination leaf switch. The destination leaf switch de-encapsulates the packet and sends it to the receiving host. With this model, ACI uses a full mesh, single hop, loop-free topology without the need to use the spanning-tree protocol to prevent loops.

The VXLAN segments are independent of the underlying network topology; conversely, the underlying IP network between VTEPs is independent of the VXLAN overlay. It routes the encapsulated packets based on the outer IP address header, which has the initiating VTEP as the source IP address and the terminating VTEP as the destination IP address.

The following figure shows how routing within the tenant is done.

Figure 3: Layer 3 VNIDs Transport ACI Inter-subnet Tenant Traffic



For each tenant VRF in the fabric, ACI assigns a single L3 VNID. ACI transports traffic across the fabric according to the L3 VNID. At the egress leaf switch, ACI routes the packet from the L3 VNID to the VNID of the egress subnet.

Traffic arriving at the fabric ingress that is sent to the ACI fabric default gateway is routed into the Layer 3 VNID. This provides very efficient forwarding in the fabric for traffic routed within the tenant. For example, with this model, traffic between 2 VMs belonging to the same tenant, on the same physical host, but on different subnets, only needs to travel to the ingress switch interface before being routed (using the minimal path cost) to the correct destination.

To distribute external routes within the fabric, ACI route reflectors use multiprotocol BGP (MP-BGP). The fabric administrator provides the autonomous system (AS) number and specifies the spine switches that become route reflectors.

Note Cisco ACI does not support IP fragmentation. Therefore, when you configure Layer 3 Outside (L3Out) connections to external routers, or Multi-Pod connections through an Inter-Pod Network (IPN), it is recommended that the interface MTU is set appropriately on both ends of a link.

IGP Protocol Packets (EIGRP, OSPFv3) are constructed by components based on the Interface MTU size. In Cisco ACI, if the CPU MTU size is less than the Interface MTU size and if the constructed packet size is greater than the CPU MTU, then the packet is dropped by the kernal, especially in IPv6. To avoid such control packet drops always configure the same MTU values on both the control plane and on the interface.

On some platforms, such as Cisco ACI, Cisco NX-OS, and Cisco IOS, the configurable MTU value does not take into account the Ethernet headers (matching IP MTU, and excluding the 14-18 Ethernet header size), while other platforms, such as IOS-XR, include the Ethernet header in the configured MTU value. A configured value of 9000 results in a max IP packet size of 9000 bytes in Cisco ACI, Cisco NX-OS, and Cisco IOS, but results in a max IP packet size of 8986 bytes for an IOS-XR untagged interface.

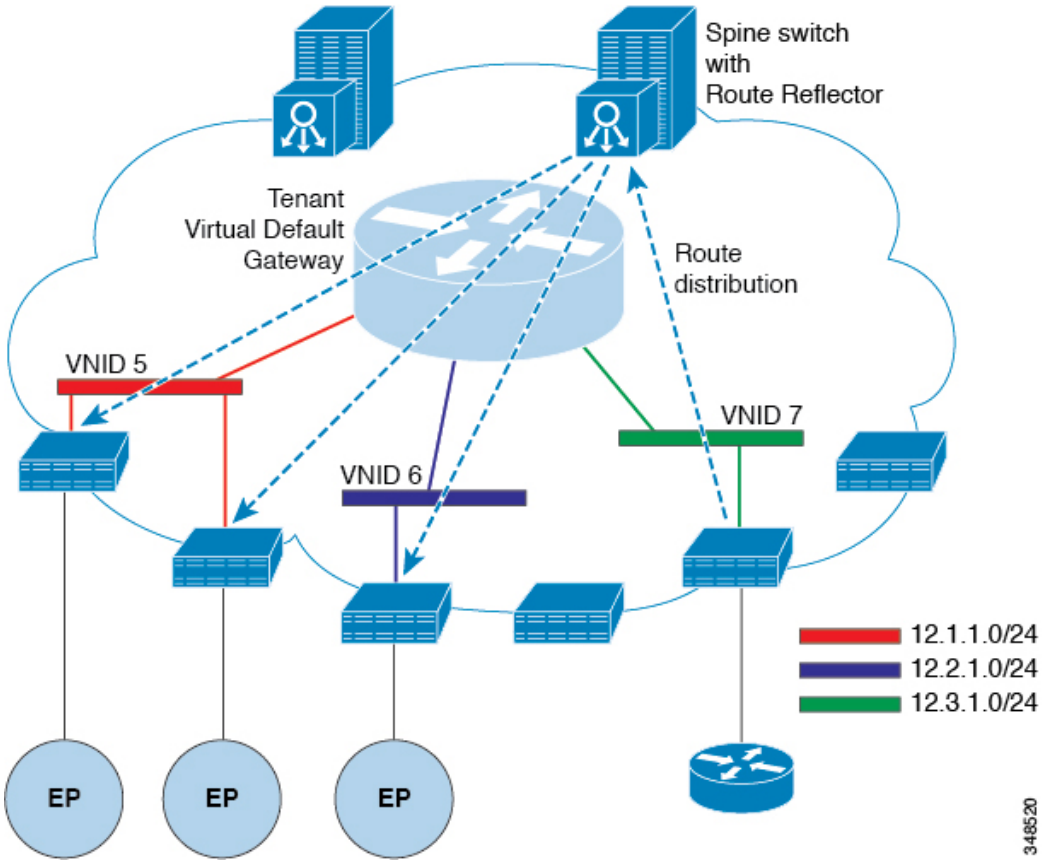
For the appropriate MTU values for each platform, see the relevant configuration guides.

We highly recommend that you test the MTU using CLI-based commands. For example, on the Cisco NX-OS CLI, use a command such as ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1.

Router Peering and Route Distribution

As shown in the figure below, when the routing peer model is used, the leaf switch interface is statically configured to peer with the external router's routing protocol.

Figure 4: Router Peering

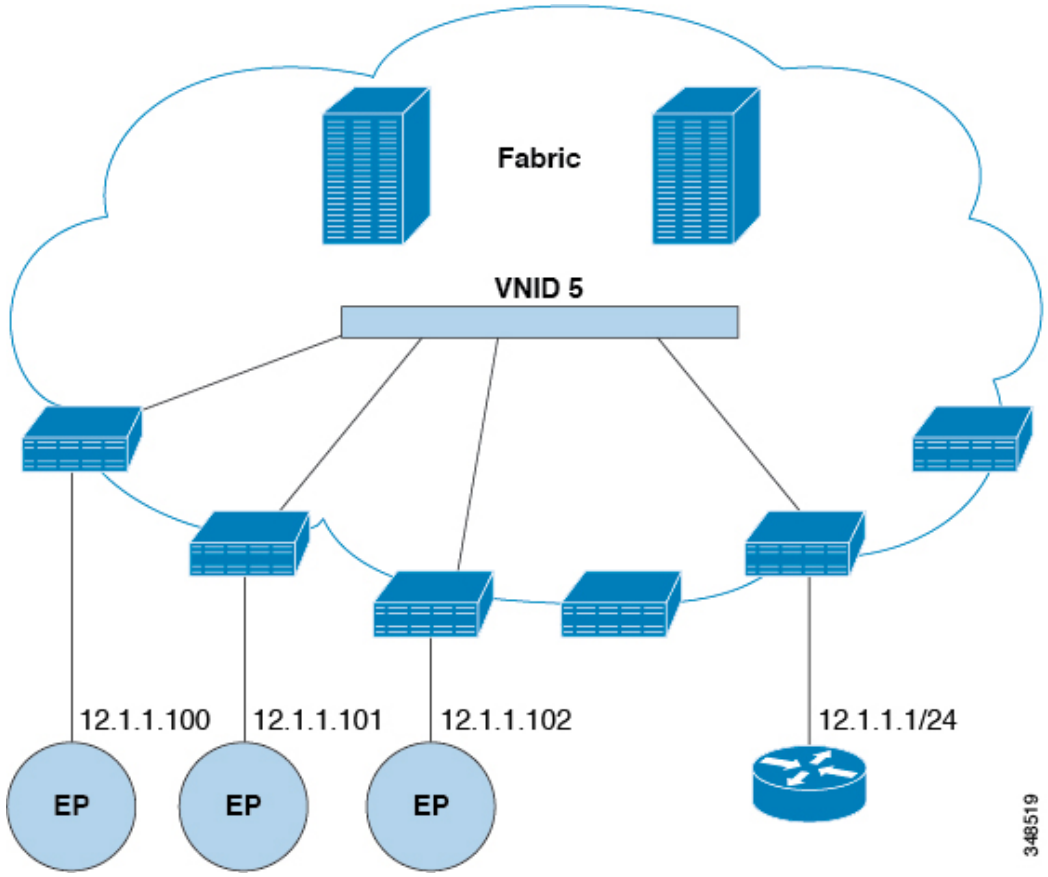


The routes that are learned through peering are sent to the spine switches. The spine switches act as route reflectors and distribute the external routes to all of the leaf switches that have interfaces that belong to the same tenant. These routes are longest prefix match (LPM) summarized addresses and are placed in the leaf switch's forwarding table with the VTEP IP address of the remote leaf switch where the external router is connected. WAN routes have no forwarding proxy. If the WAN routes do not fit in the leaf switch's forwarding table, the traffic is dropped. Because the external router is not the default gateway, packets from the tenant endpoints (EPs) are sent to the default gateway in the ACI fabric.

Bridged Interface to an External Router

As shown in the figure below, when the leaf switch interface is configured as a bridged interface, the default gateway for the tenant VNID is the external router.

Figure 5: Bridged External Router



The ACI fabric is unaware of the presence of the external router and the APIC statically assigns the leaf switch interface to its EPG.

Configuring Route Reflectors

ACI fabric route reflectors use multiprotocol BGP (MP-BGP) to distribute external routes within the fabric. To enable route reflectors in the ACI fabric, the fabric administrator must select the spine switches that will be the route reflectors, and provide the autonomous system (AS) number. It is recommended to configure at least two spine nodes per pod as MP-BGP route reflectors for redundancy.

After route reflectors are enabled in the ACI fabric, administrators can configure connectivity to external networks through leaf nodes using a component called Layer 3 Out (L3Out). A leaf node configured with an L3Out is called a border leaf. The border leaf exchanges routes with a connected external device via a routing protocol specified in the L3Out. You can also configure static routes via L3Outs.

After both L3Outs and spine route reflectors are deployed, border leaf nodes learn external routes via L3Outs, and those external routes are distributed to all leaf nodes in the fabric via spine MP-BGP route reflectors.

Check the *Verified Scalability Guide for Cisco APIC* for your release to find the maximum number of routes supported by a leaf.

Configuring External Connectivity Using a Layer 3 Out

This section provides a step-by-step configuration required for the ACI fabric to connect to an external routed network through L3Outs and MP-BGP route reflectors.

This example uses Open Shortest Path First (OSPF) as the routing protocol in an L3Out under the 'mgmt' tenant.

Configuring an MP-BGP Route Reflector Using the GUI

Procedure

Step 1 On the menu bar, choose **System** > **System Settings**.

Step 2 In the **Navigation** pane, right-click **BGP Route Reflector**, and click **Create Route Reflector Node**.

Step 3 In the **Create Route Reflector Node** dialog box, from the **Spine Node** drop-down list, choose the appropriate spine node. Click **Submit**.

Note
Repeat the above steps to add additional spine nodes as required.

The spine switch is marked as the route reflector node.

Step 4 In the **BGP Route Reflector** properties area, in the **Autonomous System Number** field, choose the appropriate number. Click **Submit**.

Note
The autonomous system number must match the leaf connected router configuration if Border Gateway Protocol (BGP) is configured on the router. If you are using routes learned using static or Open Shortest Path First (OSPF), the autonomous system number value can be any valid value.

Step 5 On the menu bar, choose **Fabric** > **Fabric Policies** > **Pods** > **Policy Groups**.

Step 6 In the **Navigation** pane, expand and right-click **Policy Groups**, and click **Create Pod Policy Group**.

Step 7 In the **Create Pod Policy Group** dialog box, in the **Name** field, enter the name of a pod policy group.

Step 8 In the **BGP Route Reflector Policy** drop-down list, choose the appropriate policy (default). Click **Submit**.
The BGP route reflector policy is associated with the route reflector pod policy group, and the BGP process is enabled on the leaf switches.

Step 9 On the menu bar, choose **Fabric** > **Fabric Policies** > **Profiles** > **Pod Profile default** > **default**.

Step 10 In the **Work** pane, from the **Fabric Policy Group** drop-down list, choose the pod policy that was created earlier. Click **Submit**.
The pod policy group is now applied to the fabric policy group.

Configuring an MP-BGP Route Reflector for the ACI Fabric

To distribute routes within the ACI fabric, an MP-BGP process must first be operating, and the spine switches must be configured as BGP route reflectors.

The following is an example of an MP-BGP route reflector configuration:

Note In this example, the BGP fabric ASN is 100. Spine switches 104 and 105 are chosen as MP-BGP route-reflectors.

```
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 100
apic1(config-bgp-fabric)# route-reflector spine 104,105
```

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>
```

Verifying the MP-BGP Route Reflector Configuration

Procedure

Step 1 Verify the configuration by performing the following actions:

a) Use secure shell (SSH) to log in as an administrator to each leaf switch as required.

b) Enter the **show processes | grep bgp** command to verify the state is S.

If the state is NR (not running), the configuration was not successful.

Step 2 Verify that the autonomous system number is configured in the spine switches by performing the following actions:

a) Use the SSH to log in as an administrator to each spine switch as required.

b) Execute the following commands from the shell window

Example:

cd /mit/sys/bgp/inst

Example:

grep asn summary

The configured autonomous system number must be displayed. If the autonomous system number value displays as 0, the configuration was not successful.

Creating an OSPF L3Out for Management Tenant Using the GUI

- 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 L3Out for a management tenant. To create an OSPF L3Out 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

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

Step 2 In the **Navigation** pane, expand **Networking** > **L3Outs**.

Step 3 Right-click **L3Outs**, and click **Create L3Out**.

The **Create L3Out** wizard appears.

Step 4 In the **Identity** window in the **Create L3Out** wizard, perform the following actions:

a) In the **Name** field, enter a name (RtdOut).

b) In the **VRF** field, from the drop-down list, choose the VRF (inb).

Note

This step associates the routed outside with the in-band VRF.

c) From the **L3 Domain** drop-down list, choose the appropriate domain.
d) Check the **OSPF** check box.
e) In the **OSPF Area ID** field, enter an area ID.
f) In the **OSPF Area Control** field, check the appropriate check box.
g) In the **OSPF Area Type** field, choose the appropriate area type.
h) In the **OSPF Area Cost** field, choose the appropriate value.
i) Click **Next**.

The **Nodes and Interfaces** window appears.

Step 5 In the **Nodes and Interfaces** window, perform the following actions:

a) Uncheck the **Use Defaults** box.

This allows you to edit the **Node Profile Name** field.

b) In the **Node Profile Name** field, enter a name for the node profile. (borderLeaf).
c) In the **Node ID** field, from the drop-down list, choose the first node. (leaf1).
d) In the **Router ID** field, enter a unique router ID.
e) In the **Loopback Address** field, use a different IP address or leave this field empty if you do not want to use the router ID for the loopback address.

Note
The **Loopback Address** field is automatically populated with the same entry that you provide in the **Router ID** field. This is the equivalent of the **Use Router ID for Loopback Address** option in previous builds. Use a different IP address or leave this field empty if you do not want to use the router ID for the loopback address.

f) Enter the appropriate information in the **Interface**, **IP Address**, **Interface Profile Name** and **MTU** fields for this node, if necessary.
g) In the **Nodes** field, click + icon to add a second set of fields for another node.

Note
You are adding a second node ID.

h) In the **Node ID** field, from the drop-down list, choose the first node. (leaf1).
i) In the **Router ID** field, enter a unique router ID.
j) In the **Loopback Address** field, use a different IP address or leave this field empty if you do not want to use the router ID for the loopback address.

Note
The **Loopback Address** field is automatically populated with the same entry that you provide in the **Router ID** field. This is the equivalent of the **Use Router ID for Loopback Address** option in previous builds. Use a different IP address or leave this field empty if you do not want to use the router ID for the loopback address.

k) Enter the appropriate information in the **Interface**, **IP Address**, **Interface Profile Name** and **MTU** fields for this node, if necessary.
l) Click **Next**.

The **Protocols** window appears.

Step 6 In the **Protocols** window, in the **Policy** area, click **default**, then click **Next**.

The **External EPG** window appears.

Step 7 In the **External EPG** window, perform the following actions:

a) In the **Name** field, enter a name for the external network (extMgmt).
b) Uncheck the **Default EPG for all external networks** field.

The **Subnets** area appears.

c) Click **+** to access the **Create Subnet** dialog box.
d) In the **Create Subnet** dialog box, in the **IP address** field, enter an IP address and mask for the subnet.
e) In the **Scope** field, check the desired check boxes. Click **OK**.
f) In the **External EPG** dialog box, click **Finish**.

Note
In the **Work** pane, in the **L3Outs** area, the L3Out icon (RtdOut) is now displayed.

Creating an OSPF External Routed Network for a Tenant Using the NX-OS CLI

Configuring external routed network connectivity involves the following steps:

1. Create a VRF under Tenant.
2. Configure L3 networking configuration for the VRF on the border leaf switches, which are connected to the external routed network. This configuration includes interfaces, routing protocols (BGP, OSPF, EIGRP), protocol parameters, route-maps.
3. Configure policies by creating external-L3 EPGs under tenant and deploy these EPGs on the border leaf switches. External routed subnets on a VRF which share the same policy within the ACI fabric form one "External L3 EPG" or one "prefix EPG".

Configuration is realized in two modes:

- Tenant mode: VRF creation and external-L3 EPG configuration
- Leaf mode: L3 networking configuration and external-L3 EPG deployment

The following steps are for creating an OSPF external routed network for a tenant. To create an OSPF external routed network for a tenant, you must choose a tenant and then create a VRF for the tenant.

Note The examples in this section show how to provide external routed connectivity to the "web" epg in the "OnlineStore" application for tenant "exampleCorp".

Procedure

Step 1 Configure the VLAN domain.

Example:

```
apic1(config)# vlan-domain dom_exampleCorp
apic1(config-vlan)# vlan 5-1000
apic1(config-vlan)# exit
```

Step 2 Configure the tenant VRF and enable policy enforcement on the VRF.

Example:

```
apic1(config)# tenant exampleCorp
apic1(config-tenant)# vrf context
 exampleCorp_v1
apic1(config-tenant-vrf)# contract enforce
apic1(config-tenant-vrf)# exit
```

Step 3 Configure the tenant BD and mark the gateway IP as “public”. The entry "scope public" makes this gateway address available for advertisement through the routing protocol for external-L3 network.

Example:

```
apic1(config-tenant)# bridge-domain exampleCorp_b1
apic1(config-tenant-bd)# vrf member exampleCorp_v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# interface bridge-domain exampleCorp_b1
apic1(config-tenant-interface)# ip address 172.1.1.1/24 scope public
apic1(config-tenant-interface)# exit
```

Step 4 Configure the VRF on a leaf.

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf exampleCorp_v1
```

Step 5 Configure the OSPF area and add the route map.

Example:

```
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant exampleCorp vrf exampleCorp_v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.1 route-map map100 out
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
```

Step 6 Assign the VRF to the interface (sub-interface in this example) and enable the OSPF area.

Example:

Note
For the sub-interface configuration, the main interface (ethernet 1/11 in this example) must be converted to an L3 port through “no switchport” and assigned a vlan-domain (dom_exampleCorp in this example) that contains the encapsulation VLAN used by the sub-interface. In the sub-interface ethernet1/11.500, 500 is the encapsulation VLAN.

```
apic1(config-leaf)# interface ethernet 1/11
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vlan-domain member dom_exampleCorp
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/11.500
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf exampleCorp_v1
```

```
apic1(config-leaf-if)# ip address 157.10.1.1/24
apic1(config-leaf-if)# ip router ospf default area 0.0.0.1
```

Step 7 Configure the external-L3 EPG policy. This includes the subnet to match for identifying the external subnet and consuming the contract to connect with the epg "web".

Example:

```
apic1(config)# tenant t100
apic1(config-tenant)# external-l3 epg  l3epg100
apic1(config-tenant-l3ext-epg)# vrf member v100
apic1(config-tenant-l3ext-epg)# match ip 145.10.1.0/24
apic1(config-tenant-l3ext-epg)# contract consumer web
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)#exit
```

Step 8 Deploy the external-L3 EPG on the leaf switch.

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t100 vrf v100
apic1(config-leaf-vrf)# external-l3 epg l3epg100
```

Creating Tenants, VRFs, and Bridge Domains

Tenants Overview

- A tenant contains policies that enable qualified users domain-based access control. Qualified users can access privileges such as tenant administration and networking administration.
- A user requires read/write privileges for accessing and configuring policies in a domain. A tenant user can have specific privileges into one or more domains.
- In a multitenancy environment, a tenant provides group user access privileges so that resources are isolated from one another (such as for endpoint groups and networking). These privileges also enable different users to manage different tenants.

Tenant Creation

A tenant contains primary elements such as filters, contracts, bridge domains, and application profiles that you can create after you first create a tenant.

VRF and Bridge Domains

You can create and specify a VRF and a bridge domain for the tenant. The defined bridge domain element subnets reference a corresponding Layer 3 context.

For details about enabling IPv6 Neighbor Discovery see*IPv6 and Neighbor Discovery* in *Cisco APIC Layer 3 Networking Guide*.

Creating a Tenant, VRF, and Bridge Domain Using the GUI

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

Procedure

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

Step 2 In the **Create Tenant** dialog box, perform the following tasks:

a) In the **Name** field, enter a name.
b) In the **Security Domains** section, click the **+** to open the **Create Security Domain** dialog box.
c) In the **Name** field, enter a name for the security domain, then click **Submit**.
d) In the **Create Tenant** dialog box, click **Update** for the security domain that you created.
e) Fill in the other fields as necessary.
f) Click **Submit**.

The ***tenant_name*** > **Networking** screen displays.

Step 3 In the **Work** pane, drag the **VRF** icon to the canvas to open the **Create VRF** dialog box, and perform the following tasks:

a) In the **Name** field, enter a name.
b) Fill in the other fields as necessary.
c) Click **Submit** to complete the VRF instance configuration.

Step 4 In the **Work** pane, drag the **Brdige Domain** icon to the canvas within the circle around the VRF instance to connect the two. In the **Create Bridge Domain** dialog box that displays, perform the following tasks:

a) In the **Name** field, enter a name.
b) Fill in the other fields as necessary.
c) Click **Next**.
d) In the **Subnets** section, click the **+** to open the **Create Subnet** dialog box.
e) In the **Gateway IP** field, enter the IP address and subnet mask.
f) Fill in the other fields as necessary.
g) Click **OK**.
h) Back in the **Create Bridge Domain** dialog box, fill in the other fields as necessary.
i) Click **Next**.
j) Fill in the fields as necessary.
k) Click **OK** to complete bridge domain configuration.

Step 5 In the **Work** pane, drag the **L3** icon to the canvas within the circle around the VRF instance to connect the two. In the **Create Routed Outside** dialog box that displays, perform the following tasks:

a) In the **Name** field, enter a name.
b) In the **Nodes And Interfaces Protocol Profiles** section, click the **+** to open the **Create Node Profile** dialog box.
c) In the **Name** field, enter a name.
d) In the **Nodes** section, click the **+** to open the **Select Node** dialog box.
e) In the **Node ID** drop-down list, choose a node.
f) In the **Router ID** field, enter the router ID.

g) In the **Static Routes** section, click the **+** to open the **Create Static Route** dialog box.
h) In the **Prefix** field, enter the IPv4 or IPv6 address.
i) In the **Next Hop Addresses** section, click the **+** to open the **Create Next Hop** dialog box.
j) In the **Next Hop Address** field, enter the IPv4 or IPv6 address.
k) In the **Preference** field, enter a number.
l) Fill in the other fields as necessary.
m) Click **OK**.
n) In the **Create Static Route** dialog box, fill in the other fields as necessary.
o) Click **OK**.
p) In the **Select Node** dialog box, fill in the other fields as necessary.
q) Click **OK**.
r) In the **Create Node Profile** dialog box, fill in the other fields as necessary.
s) Click **OK**.
t) Put a check in the **BGP**, **OSPF**, or **EIGRP** check boxes if desired.
u) Fill in the other fields as necessary.
v) Click **Next**.
w) Fill in the fields as necessary.
x) Click **OK** to complete the Layer 3 configuration.

To confirm the Layer 3 configuration, in the **Navigation** pane, expand **Networking** > **VRFs**.

Deploying EPGs

Statically Deploying an EPG on a Specific Port

This topic provides a typical example of how to statically deploy an EPG on a specific port when using Cisco APIC.

Deploying an EPG on a Specific Node or Port Using the GUI

Before you begin

The tenant where you deploy the EPG is already created.

You can create an EPG on a specific node or a specific port on a node.

Procedure

Step 1 Log in to the Cisco APIC.

Step 2 Choose **Tenants** > ***tenant***.

Step 3 In the left navigation pane, expand ***tenant***, **Application Profiles**, and the ***application profile***.

Step 4 Right-click **Application EPGs** and choose **Create Application EPG**.

Step 5 In the **Create Application EPG STEP 1 > Identity** dialog box, complete the following steps:

a) In the **Name** field, enter a name for the EPG.

b) From the **Bridge Domain** drop-down list, choose a bridge domain.
c) Check the **Statically Link with Leaves/Paths** check box.

This check box allows you to specify on which port you want to deploy the EPG.

d) Click **Next**.
e) From the **Path** drop-down list, choose the static path to the destination EPG.

Step 6 In the **Create Application EPG STEP 2 > Leaves/Paths** dialog box, from the **Physical Domain** drop-down list, choose a physical domain.

Step 7 Complete one of the following sets of steps:

Option	Description
If you want to deploy the EPG on...	Then
A node	**a.** Expand the **Leaves** area. **b.** From the **Node** drop-down list, choose a node. **c.** In the **Encap** field, enter the appropriate VLAN. **d.** (Optional) From the **Deployment Immediacy** drop-down list, accept the default **On Demand** or choose **Immediate**. **e.** (Optional) From the Mode drop-down list, accept the default **Trunk** or choose another mode.
A port on the node	**a.** Expand the **Paths** area. **b.** From the **Path** drop-down list, choose the appropriate node and port. **c.** (Optional) In the **Deployment Immediacy** field drop-down list, accept the default **On Demand** or choose **Immediate**. **d.** (Optional) From the Mode drop-down list, accept the default **Trunk** or choose another mode. **e.** In the **Port Encap** field, enter the secondary VLAN to be deployed. **f.** (Optional) In the **Primary Encap** field, enter the primary VLAN to be deployed.

Step 8 Click **Update** and click **Finish**.

Step 9 In the left navigation pane, expand the EPG that you created.

Step 10 Complete one of the following actions:

- If you created the EPG on a node, click **Static Leafs**, and in the work pane view details of the static binding paths.
- If you created the EPG on a port of the node, click **Static Ports**, and in the work pane view details of the static binding paths.

Creating Domains, Attach Entity Profiles, and VLANs to Deploy an EPG on a Specific Port

This topic provides a typical example of how to create physical domains, Attach Entity Profiles (AEP), and VLANs that are mandatory to deploy an EPG on a specific port.

All endpoint groups (EPGs) require a domain. Interface policy groups must also be associated with Attach Entity Profile (AEP), and the AEP must be associated with a domain, if the AEP and EPG have to be in same domain. Based on the association of EPGs to domains and of interface policy groups to domains, the ports and VLANs that the EPG uses are validated. The following domain types associate with EPGs:

- Application EPGs
- Layer 3 external outside network instance EPGs
- Layer 2 external outside network instance EPGs
- Management EPGs for out-of-band and in-band access

The APIC checks if an EPG is associated with one or more of these types of domains. If the EPG is not associated, the system accepts the configuration but raises a fault. The deployed configuration may not function properly if the domain association is not valid. For example, if the VLAN encapsulation is not valid for use with the EPG, the deployed configuration may not function properly.

Note EPG association with the AEP without static binding does not work in a scenario when you configure the EPG as **Trunk** under the AEP with one end point under the same EPG supporting Tagging and the other end point in the same EPG does not support VLAN tagging. While associating AEP under the EPG, you can configure it as Trunk, Access (Tagged) or Access (Untagged).

Creating Domains, and VLANS to Deploy an EPG on a Specific Port Using the GUI

Before you begin

- The tenant where you deploy the EPG is already created.
- An EPG is statically deployed on a specific port.

Procedure

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

Step 2 In the **Navigation** pane, click **Quick Start**.

Step 3 In the **Work** pane, click **Configure an Interface, PC, and vPC**.

Step 4 In the **Configure an Interface, PC, and vPC** dialog box, click the + icon to select switches and perform the following actions:

a) From the **Switches** drop-down list, check the check box for the desired switch.

b) In the **Switch Profile Name** field, a switch name is automatically populated.

Note

Optionally, you can enter a modified name.

c) Click the + icon to configure the switch interfaces.
d) In the **Interface Type** field, click the **Individual** radio button.
e) In the **Interfaces** field, enter the range of desired interfaces.
f) In the **Interface Selector Name** field, an interface name is automatically populated.

Note
Optionally, you can enter a modified name.

g) In the **Interface Policy Group** field, choose the **Create One** radio button.
h) From the **Link Level Policy** drop-down list, choose the appropriate link level policy.

Note
Create additional policies as desired, otherwise the default policy settings are available.

i) From the **Attached Device Type** field, choose the appropriate device type.
j) In the **Domain** field, click the **Create One** radio button.
k) In the **Domain Name** field, enter a domain name.
l) In the **VLAN** field, click the **Create One** radio button.
m) In the **VLAN Range** field, enter the desired VLAN range. Click **Save**, and click **Save** again.
n) Click **Submit**.

Step 5 On the menu bar, click **Tenants**. In the **Navigation** pane, expand the appropriate ***Tenant_name*** > **Application Profiles** > **Application EPGs** > ***EPG_name*** and perform the following actions:

a) Right-click **Domains (VMs and Bare-Metals)**, and click **Add Physical Domain Association**.
b) In the **Add Physical Domain Association** dialog box, from the **Physical Domain Profile** drop-down list, choose the appropriate domain.
c) Click **Submit**.
The AEP is associated with a specific port on a node and with a domain. The physical domain is associated with the VLAN pool and the Tenant is associated with this physical domain.

The switch profile and the interface profile are created. The policy group is created in the port block under the interface profile. The AEP is automatically created, and it is associated with the port block and with the domain. The domain is associated with the VLAN pool and the Tenant is associated with the domain.

Deploying an Application EPG through an AEP or Interface Policy Group to Multiple Ports

Through the APIC Advanced GUI and REST API, you can associate attached entity profiles directly with application EPGs. By doing so, you deploy the associated application EPGs to all those ports associated with the attached entity profile in a single configuration.

Through the APIC REST API or the NX-OS style CLI, you can deploy an application EPG to multiple ports through an Interface Policy Group.

Deploying an EPG through an AEP to Multiple Interfaces Using the APIC GUI

You can quickly associate an application with an attached entity profile to quickly deploy that EPG over all the ports associated with that attached entity profile.

Before you begin

- The target application EPG is created.
- The VLAN pools has been created containing the range of VLANs you wish to use for EPG Deployment on the AEP.
- The physical domain has been created and linked to the VLAN Pool and AEP.
- The target attached entity profile is created and is associated with the ports on which you want to deploy the application EPG.

Procedure

Step 1 Navigate to the target attached entity profile.

a) Open the page for the attached entity profile to use. In the GUI, click **Fabric** > **Access Policies** > **Policies** > **Global** > **Attachable Access Entity Profiles**.

b) Click the target attached entity profile to open its Attachable Access Entity Profile window.

Step 2 Click the **Show Usage** button to view the leaf switches and interfaces associated with this attached entity profile.

the application EPGs associated with this attached entity profile are deployed to all the ports on all the switches associated with this attached entity profile.

Step 3 Use the **Application EPGs** table to associate the target application EPG with this attached entity profile. Click + to add an application EPG entry. Each entry contains the following fields:

Field	Action
Application EPGs	Use the drop down to choose the associated Tenant, Application Profile, and target application EPG.
Encap	Enter the name of the VLAN over which the target application EPG will communicate.
Primary Encap	If the application EPG requires a primary VLAN, enter the name of the primary VLAN.
Mode	Use the drop down to specify the mode in which data is transmitted: • **Trunk** -- Choose if traffic from the host is tagged with a VLAN ID. • **Access** -- Choose if traffic from the host is tagged with an 802.1p tag. • **Access Untagged** -- Choose if the traffic from the host is untagged.

Step 4 Click **Submit**.

the application EPGs associated with this attached entity profile are deployed to all the ports on all the switches associated with this attached entity profile.

Microsegmented EPGs

Using Microsegmentation with Network-based Attributes on Bare Metal

You can use Cisco APIC to configure Microsegmentation with Cisco ACI to create a new, attribute-based EPG using a network-based attribute, a MAC address or one or more IP addresses. You can configure Microsegmentation with Cisco ACI using network-based attributes to isolate VMs or physical endpoints within a single base EPG or VMs or physical endpoints in different EPGs.

Using an IP-based Attribute

You can use an IP-based filter to isolate a single IP address, a subnet, or multiple of noncontiguous IP addresses in a single microsegment. You might want to isolate physical endpoints based on IP addresses as a quick and simply way to create a security zone, similar to using a firewall.

Using a MAC-based Attribute

You can use a MAC-based filter to isolate a single MAC address or multiple MAC addresses. You might want to do this if you have a server sending bad traffic int he network. By creating a microsegment with a MAC-based filter, you can isolate the server.

Configuring Network-based Microsegmented EPGs in a Bare-Metal environment Using the GUI

You can use Cisco APIC to configure microsegmentation to put physical endpoint devices that belong to different base EPGs or the same EPG into a new attribute-based EPG.

Procedure

Step 1 Log into the Cisco APIC.

Step 2 Choose **Tenants** and then choose the tenant within which you want to create a microsegment.

Step 3 In the tenant navigation pane, expand the tenant folder, the **Application Profiles** folder, the ***profile*** folder, and the **Application EPGs** folder.

Step 4 Take one of the following actions:

- If you want to put physical endpoint devices from the same base EPG into a new, attribute-based EPG, click the base EPG containing the physical endpoint devices.
- If you want to put physical endpoint devices from different base EPGs into a new, attribute-based EPG, click one of the base EPG containing the physical endpoint devices.

The properties for the base EPG appear in the work pane.

Step 5 In the work pane, click the **Operational** tab at the top right of the screen.

Step 6 Below the **Operational** tab, ensure that the **Client End-Points** tab is active.
The work pane displays all the physical endpoints that belong to the base EPG.

Step 7 Note the IP address or MAC address for the endpoint device or endpoint devices that you want to put into a new microsegment.

Step 8 If you want to put endpoint devices from different base EPGs into a new attribute-based EPG, repeat Step 4 through Step 7 for each of the base EPGs.

Step 9 In the tenant navigation pane, right-click the **uSeg EPGs** folder, and then choose **Create uSeg EPG**.

Step 10 Complete the following series of steps to begin creation of an attribute-based EPG for one of the groups of endpoint devices:

a) In the **Create uSeg EPG** dialog box, in the **Name** field, enter a name.

We recommend that you choose a name that indicates that the new attribute-based EPG is a microsegment.

b) In the intra-EPG isolation field, select **enforced** or **unenforced**.

If you select **enforced**, ACI prevents all communication between the endpoint devices within this uSeg EPG.

c) In the **Bridge Domain** area, choose a bridge domain from the drop-down list.

d) In the **uSeg Attributes** area, choose **IP Address Filter** or **MAC Address Filter** from the + drop-down list on the right side of the dialog box.

Step 11 Complete one of the following series of steps to configure the filter.

If you want to use...	Then...
An IP-based attribute	**a.** In the **Create IP Attribute** dialog box, in the **Name** field, enter a name. We recommend that you choose a name that reflects the filter's function. **b.** In the **IP Address** field, enter an IP address or a subnet with the appropriate subnet mask. **c.** Click **OK**. **d.** (Optional) Create a second IP Address filter by repeating Step 10 c through Step 11 c. You might want to do this to include discontinuous IP addresses in the microsegment. **e.** In the **Create uSeg EPG** dialog box, click **Submit**.
A MAC-based attribute	**a.** In the **Create MAC Attribute** dialog box, in the **Name** field, enter a name. We recommend that you choose a name that reflects the filter's function. **b.** In the **MAC Address** field, enter a MAC address. **c.** Click **OK**. **d.** In the **Create uSeg EPG** dialog box, click **Submit**.

Step 12 Complete the following steps to associate the uSeg EPG with a physical domain.

a) In the navigation pane, ensure that the uSeg EPG folder is open and then open the container for the microsegment that you just created.

b) Click the folder **Domains (VMs and Bare-Metals)**.

c) On the right side of the work pane, click **Actions** and then choose **Add Physical Domain Association** from the drop-down list.

d) In the **Add Physical Domain Association** dialog box, choose a profile from the **Physical Domain Profile** drop-down list.
e) In the **Deploy Immediacy** area, accept the default **On Demand**.
f) In the **Resolution Immediacy** area, accept the default **Immediate**.
g) Click **Submit**.

Step 13 Associate the uSeg EPG with the appropriate leaf switch.

a) In the navigation pane, ensure the uSeg EPG folder is open then click **Static Leafs**.
b) In the Static Leafs window, click **Actions** > **Statically Link with Node**
c) In the Statically Link With Node dialog, select the leaf node and mode.
d) Click **Submit**.

Step 14 Repeat Step 9 through Step 13 for any other network attribute-based EPGs that you want to create.

What to do next

Verify that the attribute-based EPG was created correctly.

If you configured an IP-based or MAC-based attribute, make sure that traffic is running on the end pont devices that you put into the new microsegments.

IP Address-Based Microsegmented EPG as a Shared Resource

You can configure an IP address-based microsegemented EPG as a resource that can be accessed from both within and without the VRF on which it is located. The method of doing so is to configure an existing IP address-based microsegmented EPG with a subnet (assigned a unicast IP address) and enable that subnet for being advertised and shared by devices located on VRFs other than the one on which this EPG is native. Then you define an IP attribute with an option enabled that associates the EPG with the IP address of the shared subnet.

Configuring an IP-based Microsegmented EPG as a Shared Resource Using the GUI

You can configure a microsegmented EPG with an IP-Address with 32 bit mask as a shared service, accessible by clients outside of the VRF and the current fabric.

Before you begin

The following GUI description of configuring assumes the preconfiguration of an IP address-based microsegmented EPG configured whose subnet mask is /32.

Note

- For directions on configuring an IP address based EPG in a physical environment, see Using Microsegmentation with Network-based Attributes on Bare Metal, on page 20
- For directions on configuring an IP address based EPG in a virtual environment, see *Configuring Microsegmentation with Cisco ACI* in the *Cisco ACI Virtualization Guide*.

Procedure

Step 1 Navigate to the target IP-address-based EPG.

a) In the APIC GUI, click **Tenant** > **tenant_name** > **uSeg EPGs** > **uSeg_epg_name** to display the EPG's **Properties** dialog.

Step 2 For the target EPG, configure an IP attribute to match the EPG subnet address.

a) In the **Properties** dialog, locate the **uSeg Attributes** table, and click **+**. When prompted, choose **IP Address Filter** to display the **Create IP Attribute** dialog.

b) Enter a name in the Name field

c) Check the box for **Use FV Subnet**.

Enabling this option, indicates that the IP attribute value matches the IP address of a shared subnet.

d) Click **Submit**.

Step 3 Create a shared subnet for the target EPG.

a) With the folder for the target IP address-based uSeg EPG still open in the APIC navigation pane, right-click the **Subnets** folder and select **Create EPG Subnets**.

b) In the **Default Gateway** field, enter the IP address/mask of the IP address-based microsegmented EPG.

Note

- In all cases the subnet mask must be /32.
- In the context of an IP address-based EPG, you are not actually entering the default address for a gateway, rather you are entering the IP address for the shared EPG subnet.

c) Select **Treat as a virtual IP address**.

d) Under Scope select **Advertised Externally** and **Shared between VRFs**.

e) Click **Submit**.

Unconfiguring an IP-based Microsegmented EPG as a Shared Resource Using the GUI

When you unconfigure an IP address-Based microsegmented EPG as a shared service, you must remove the shared subnet and also disable the option to use that subnet as a shared resource.

Before you begin

Before you unconfigure an IP address-based microsegmented EPG as a shared service, you should know the following:

- Know which subnet is configured as a shared service address for the IP address-based microsegmented EPG.
- Know which IP attribute is configured with the **Use FV Subnet** option enabled.

Procedure

Step 1 Remove subnet from the IP addressed-based microsegmented EPG.

a) In the APIC GUI, click **Tenant** > **tenant_name** > **Application Profiles** > **epg_name** > **uSeg EPGs** > **uSeg EPGs** > **uSeg_epg_name**.

b) With the folder for the target IP address-based uSeg EPG still open in the APIC navigation pane, click the **Subnets** folder.

c) In the **Subnets** window, select the subnet that is advertised and shared with other VRFs and click **Actions** > **Delete**. then

d) Click **Yes** to confirm the deletion.

Step 2 Disable the **Use FV Subnet** option.

a) With the folder for the target IP address-based uSeg EPG still open in the APIC navigation pane, click the name of the micro-segmented EPG to display the to display the EPG's **Properties** dialog.

b) In the **Properties** dialog, locate the **uSeg Attributes** table, and locate the IP attribute item with the **Use FV Subnet** option enabled.

c) Double-click that item to display the **Edit IP Attribute** dialog.

d) In the **`Edit IP Attribute`** dialog, deselect the **Use FV Subnet** option.

e) Assign another IP address attribute in the IP Address field.

Note
This address must be a unicast address with a 32 bit mask (for example: 124.124.124.123/32).

f) Click **Submit**.

Deploying Application Profiles and Contracts

Security Policy Enforcement

As traffic enters the leaf switch from the front panel interfaces, the packets are marked with the EPG of the source EPG. The leaf switch then performs a forwarding lookup on the packet destination IP address within the tenant space. A hit can result in any of the following scenarios:

1. A unicast (/32) hit provides the EPG of the destination endpoint and either the local interface or the remote leaf switch VTEP IP address where the destination endpoint is present.
2. A unicast hit of a subnet prefix (not /32) provides the EPG of the destination subnet prefix and either the local interface or the remote leaf switch VTEP IP address where the destination subnet prefix is present.
3. A multicast hit provides the local interfaces of local receivers and the outer destination IP address to use in the VXLAN encapsulation across the fabric and the EPG of the multicast group.

Note Multicast and external router subnets always result in a hit on the ingress leaf switch. Security policy enforcement occurs as soon as the destination EPG is known by the ingress leaf switch.

A miss result in the forwarding table causes the packet to be sent to the forwarding proxy in the spine switch. The forwarding proxy then performs a forwarding table lookup. If it is a miss, the packet is dropped. If it is a hit, the packet is sent to the egress leaf switch that contains the destination endpoint. Because the egress leaf switch knows the EPG of the destination, it performs the security policy enforcement. The egress leaf switch must also know the EPG of the packet source. The fabric header enables this process because it carries the EPG from the ingress leaf switch to the egress leaf switch. The spine switch preserves the original EPG in the packet when it performs the forwarding proxy function.

On the egress leaf switch, the source IP address, source VTEP, and source EPG information are stored in the local forwarding table through learning. Because most flows are bidirectional, a return packet populates the forwarding table on both sides of the flow, which enables the traffic to be ingress filtered in both directions.

Contracts Contain Security Policy Specifications

In the ACI security model, contracts contain the policies that govern the communication between EPGs. The contract specifies what can be communicated and the EPGs specify the source and destination of the communications. Contracts link EPGs, as shown below.

EPG 1 ---------- CONTRACT ---------- EPG 2

Endpoints in EPG 1 can communicate with endpoints in EPG 2 and vice versa if the contract allows it. This policy construct is very flexible. There can be many contracts between EPG 1 and EPG 2, there can be more than two EPGs that use a contract, and contracts can be reused across multiple sets of EPGs, and more.

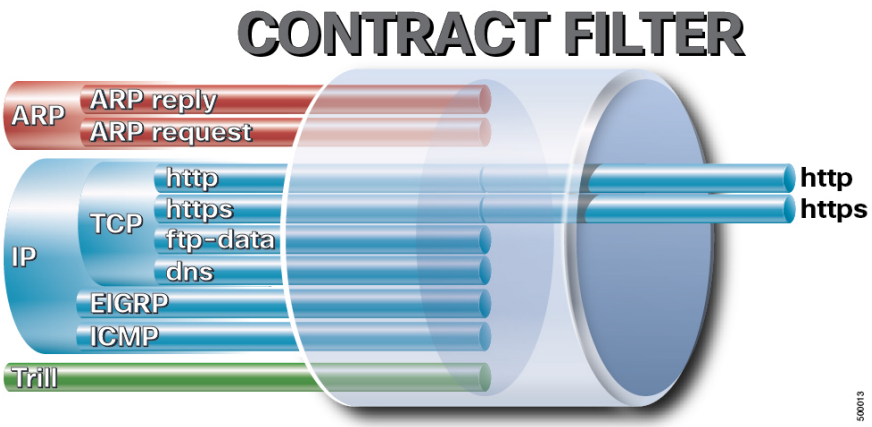
There is also directionality in the relationship between EPGs and contracts. EPGs can either provide or consume a contract. An EPG that provides a contract is typically a set of endpoints that provide a service to a set of client devices. The protocols used by that service are defined in the contract. An EPG that consumes a contract is typically a set of endpoints that are clients of that service. When the client endpoint (consumer) tries to connect to a server endpoint (provider), the contract checks to see if that connection is allowed. Unless otherwise specified, that contract would not allow a server to initiate a connection to a client. However, another contract between the EPGs could easily allow a connection in that direction.

This providing/consuming relationship is typically shown graphically with arrows between the EPGs and the contract. Note the direction of the arrows shown below.

EPG 1 <-------consumes-------- CONTRACT <-------provides-------- EPG 2

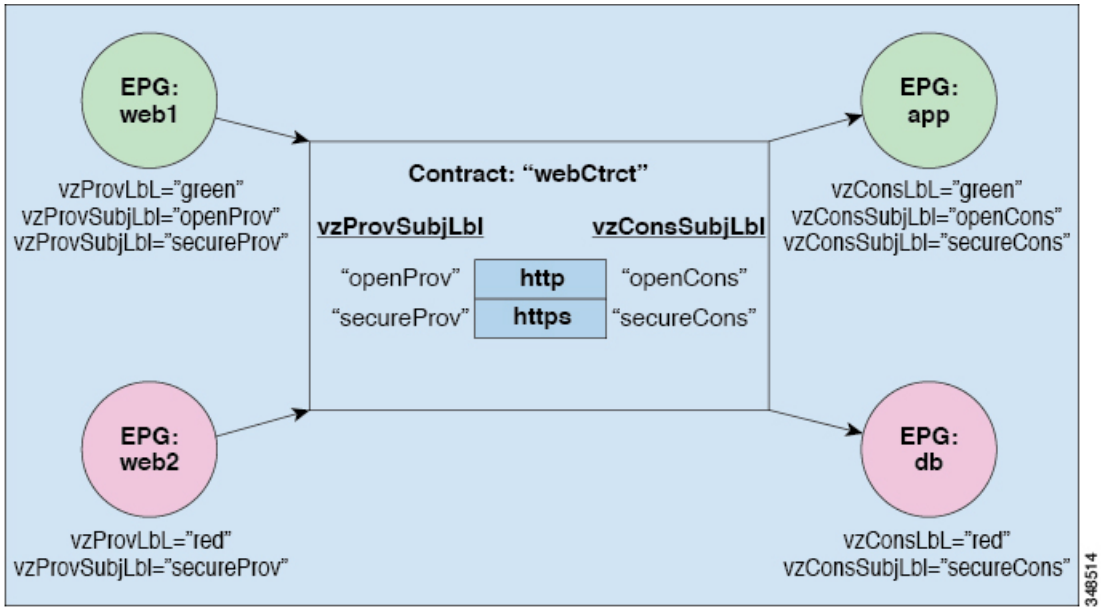
The contract is constructed in a hierarchical manner. It consists of one or more subjects, each subject contains one or more filters, and each filter can define one or more protocols.

Figure 6: Contract Filters



The following figure shows how contracts govern EPG communications.

Figure 7: Contracts Determine EPG to EPG Communications



For example, you may define a filter called HTTP that specifies TCP port 80 and port 8080 and another filter called HTTPS that specifies TCP port 443. You might then create a contract called webCtrct that has two sets of subjects. openProv and openCons are the subjects that contain the HTTP filter. secureProv and secureCons are the subjects that contain the HTTPS filter. This webCtrct contract can be used to allow both secure and non-secure web traffic between EPGs that provide the web service and EPGs that contain endpoints that want to consume that service.

These same constructs also apply for policies that govern virtual machine hypervisors. When an EPG is placed in a virtual machine manager (VMM) domain, the APIC downloads all of the policies that are associated with the EPG to the leaf switches with interfaces connecting to the VMM domain. For a full explanation of VMM domains, see the *Virtual Machine Manager Domains* chapter of *Application Centric Infrastructure Fundamentals*. When this policy is created, the APIC pushes it (pre-populates it) to a VMM domain that specifies which switches allow connectivity for the endpoints in the EPGs. The VMM domain defines the set

of switches and ports that allow endpoints in an EPG to connect to. When an endpoint comes on-line, it is associated with the appropriate EPGs. When it sends a packet, the source EPG and destination EPG are derived from the packet and the policy defined by the corresponding contract is checked to see if the packet is allowed. If yes, the packet is forwarded. If no, the packet is dropped.

Contracts consist of 1 or more subjects. Each subject contains 1 or more filters. Each filter contains 1 or more entries. Each entry is equivalent to a line in an Access Control List (ACL) that is applied on the Leaf switch to which the endpoint within the endpoint group is attached.

In detail, contracts are comprised of the following items:

- Name—All contracts that are consumed by a tenant must have different names (including contracts created under the common tenant or the tenant itself).
- Subjects—A group of filters for a specific application or service.
- Filters—Used to classify traffic based upon layer 2 to layer 4 attributes (such as Ethernet type, protocol type, TCP flags and ports).
- Actions—Action to be taken on the filtered traffic. The following actions are supported:
  - Permit the traffic (regular contracts, only)
  - Mark the traffic (DSCP/CoS) (regular contracts, only)
  - Redirect the traffic (regular contracts, only, through a service graph)
  - Copy the traffic (regular contracts, only, through a service graph or SPAN)
  - Block the traffic (taboo contracts)

    With Cisco APIC Release 3.2(x) and switches with names that end in EX or FX, you can alternatively use a subject Deny action or Contract or Subject Exception in a standard contract to block traffic with specified patterns.
  - Log the traffic (taboo contracts and regular contracts)
- Aliases—(Optional) A changeable name for an object. Although the name of an object, once created, cannot be changed, the Alias is a property that can be changed.

Thus, the contract allows more complex actions than just allow or deny. The contract can specify that traffic that matches a given subject can be re-directed to a service, can be copied, or can have its QoS level modified. With pre-population of the access policy in the concrete model, endpoints can move, new ones can come on-line, and communication can occur even if the APIC is off-line or otherwise inaccessible. The APIC is removed from being a single point of failure for the network. Upon packet ingress to the ACI fabric, security policies are enforced by the concrete model running in the switch.

Three-Tier Application Deployment

A filter specifies the data protocols to be allowed or denied by a contract that contains the filter. A contract can contain multiple subjects. A subject can be used to realize uni- or bidirectional filters. A unidirectional filter is a filter that is used in one direction, either from consumer-to-provider (IN) or from provider-to-consumer (OUT) filter. A bidirectional filter is the same filter that is used in both directions. It is not reflexive.

Contracts are policies that enable inter-End Point Group (inter-EPG) communication. These policies are the rules that specify communication between application tiers. If no contract is attached to the EPG, inter-EPG

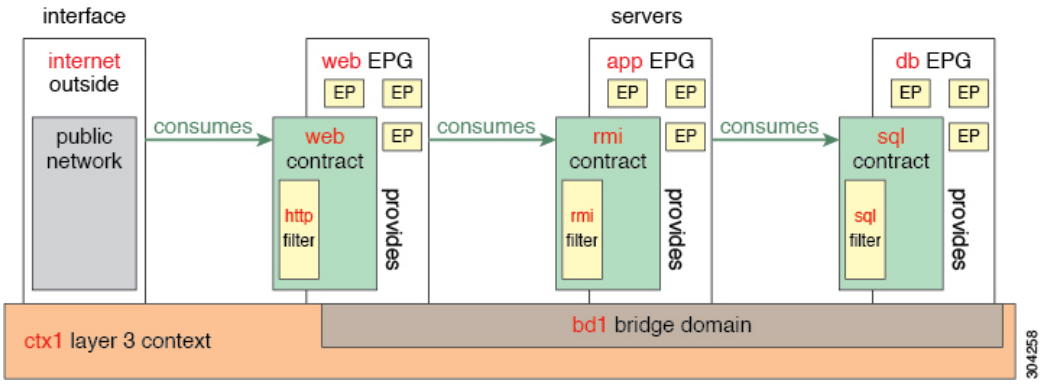
communication is disabled by default. No contract is required for intra-EPG communication because intra-EPG communication is always allowed.

Application profiles enable you to model application requirements that the APIC then automatically renders in the network and data center infrastructure. The application profiles enable administrators to approach the resource pool in terms of applications rather than infrastructure building blocks. The application profile is a container that holds EPGs that are logically related to one another. EPGs can communicate with other EPGs in the same application profile and with EPGs in other application profiles.

To deploy an application policy, you must create the required application profiles, filters, and contracts. Typically, the APIC fabric hosts a three-tier application within a tenant network. In this example, the application is implemented by using three servers (a web server, an application server, and a database server). See the following figure for an example of a three-tier application.

The web server has the HTTP filter, the application server has the Remote Method Invocation (RMI) filter, and the database server has the Structured Query Language (SQL) filter. The application server consumes the SQL contract to communicate with the database server. The web server consumes the RMI contract to communicate with the application server. The traffic enters from the web server and communicates with the application server. The application server then communicates with the database server, and the traffic can also communicate externally.

Figure 8: Three-Tier Application Diagram



Parameters to Create a Filter for http

The parameters to create a filter for http in this example is as follows:

Parameter Name	Filter for http
Name	http
Number of Entries	2
Entry Name	Dport-80 Dport-443
Ethertype	IP
Protocol	tcp tcp

Parameter Name	Filter for http
Destination Port	http https

Parameters to Create Filters for rmi and sql

The parameters to create filters for rmi and sql in this example are as follows:

Parameter Name	Filter for rmi	Filter for sql
Name	rmi	sql
Number of Entries	1	1
Entry Name	Dport-1099	Dport-1521
Ethertype	IP	IP
Protocol	tcp	tcp
Destination Port	1099	1521

Example Application Profile Database

The application profile database in this example is as follows:

EPG	Provided Contracts	Consumed Contracts
web	web	rmi
app	rmi	sql
db	sql	--

Creating an Application Profile Using the GUI

Procedure

Step 1 On the menu bar, choose **TENANTS**. In the **Navigation** pane, expand the tenant, right-click **Application Profiles**, and click **Create Application Profile**.

Step 2 In the **Create Application Profile** dialog box, in the **Name** field, add the application profile name (OnlineStore).

Creating EPGs Using the GUI

The port the EPG uses must belong to one of the VM Managers (VMM) or physical domains associated with the EPG.

Procedure

Step 1 On the menu bar, choose **Tenants** and the tenant where you want to create an EPG.

Step 2 In the navigation pane, expand the folder for the tenant, the **Application Profiles** folder, and the folder for the application profile.

Step 3 Right-click the **Application EPG** folder, and in the **Create Application EPG** dialog box, perform the following actions:

a) In the **Name** field, add the EPG name (db).

b) In the **Bridge Domain** field, choose the bridge domain from the drop-down list (bd1).

c) Check the **Associate to VM Domain Profiles** check box. Click **Next**.

d) In the **STEP 2 > Domains** area, expand **Associate VM Domain Profiles** and from the drop-down list, choose the desired VMM domain.

e) From the **Deployment Immediacy** drop-down list, accept the default or choose when policies are deployed from Cisco APIC to the physical leaf switch.

f) From the **Resolution Immediacy** drop-down list, choose when policies are deployed from the physical leaf switch to the virtual leaf.

If you have Cisco AVS, choose **Immediate** or **On Demand**; if you have Cisco ACI Virtual Edge or VMware VDS, choose **Immediate**, **On Demand**, or **Pre-provision**.

g) (Optional) In the **Delimiter** field, enter one of the following symbols: |, ~, !, @, ^, +, or =.

If you do not enter a symbol, the system uses the default | delimiter in the VMware portgroup name.

h) If you have Cisco ACI Virtual Edge or Cisco AVS, from the **Encap Mode** drop-down list, choose an encapsulation mode.

You can choose one of the following encapsulation modes:

- **VXLAN**:This overrides the domain's VLAN configuration, and the EPG uses VXLAN encapsulation. However, a fault is for the EPG if a multicast pool is not configured on the domain.
- **VLAN**: This overrides the domain's VXLAN configuration, and the EPG uses VLAN encapsulation. However, a fault is triggered for the EPG if a VLAN pool is not configured on the domain.
- **Auto**: This causes the EPG to use the same encapsulation mode as the VMM domain. This is the default configuration.

i) If you have Cisco ACI Virtual Edge, from the **Switching Mode** drop-down list, choose **native** or **AVE**.

If you choose **native**, the EPG is switched through the VMware VDS; if you choose **AVE**, the EPG is switched through the Cisco ACI Virtual Edge. The default is **native**.

j) Click **Update** and then click **Finish**.

Step 4 In the **Create Application Profile** dialog box, create two more EPGs. Create the three EPGs—db, app, and web—in the same bridge domain and data center.

Configuring Contracts Using the APIC GUI

Guidelines and Limitations for Contracts and Filters

If your fabric consists of first-generation Cisco Nexus 9300 leaf switches, such as Cisco Nexus 93128TX, 93120TX, 9396TX, 9396PX, 9372PX, 9372PX-E, 9372TX and 9372TX-E, only **IP** as an **EtherType** match is supported with contract filters. The capability to match more granular options, such as **IPv4** or **IPv6**, in the **EtherType** field for contract filters is supported only on leaf switch models with -EX, -FX, or -FX2 at the end of the switch name.

Creating a Filter Using the GUI

Create three separate filters. In this example they are HTTP, RMI, SQL. This task shows how to create the HTTP filter. The task is identical for creating the other filters.

Before you begin

Verify that the tenant, network, and bridge domain have been created.

Procedure

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

Step 2 In the **Navigation** pane, expand the ***tenant-name*** > **Contracts**, right-click **Filters**, and choose **Create Filter**.

Step 3 In the **Create Filter** dialog box, perform the following actions:

a) In the **Name** field, enter the filter name (http).
b) In the **Entries** table, click +.and in the **Name** field, enter the name (Dport-80).
c) From the **EtherType** drop-down list, choose the EtherType (IP).
d) From the **IP Protocol** drop-down list, choose the protocol (tcp).
e) From the **Destination Port/Range** drop-down lists, choose **http** in the **From** and **To** fields. (http)
f) Click **Update**, and click **Submit**.
The newly added filter appears in the **Navigation** pane and in the **Work** pane.

Note
In the **Entries** table, **ARP Flag** has no functionality and you cannot configure it. Ignore this field.

Step 4 Expand **Entries** in the **Name** field. Follow the same process to add another entry with HTTPS as the **Destination** port, and click **Update**.

This new filter rule is added.

Step 5 Follow the same process in the earlier steps to create two more filters (rmi and sql) and use the parameters provided in Parameters to Create Filters for rmi and sql, on page 29.

Creating a Contract Using the GUI

Procedure

Step 1 On the menu bar, choose **Tenants** and the tenant name on which you want to operate. In the **Navigation** pane, expand the ***tenant-name*** > **Contracts**.

Step 2 Right-click **Standard** > **Create Contract**.

Step 3 In the **Create Contract** dialog box, perform the following tasks:

a) In the **Name** field, enter the contract name (web).

b) Click the + sign next to **Subjects** to add a new subject.

c) In the **Create Contract Subject** dialog box, enter a subject name in the **Name** field. (web)

d) **Note**
This step associates the filters created that were earlier with the contract subject.

In the **Filter Chain** area, click the + sign next to **Filters**.

e) In the dialog box, from the drop-down menu, choose the filter name (http), and click **Update**.

Step 4 In the **Create Contract Subject** dialog box, click **OK**.

Step 5 Create two more contracts for rmi and for sql following the same steps in this procedure. For the rmi contract, choose the rmi subject and for sql, choose the sql subject.

Consuming and Providing Contracts Using the GUI

You can associate contracts that were created earlier to create policy relationships between the EPGs.

When you name the provided and consumed contracts, verify that you give the same name for both provided and consumed contracts.

Procedure

Step 1 **Note**
The db, app, and web EPGs are displayed as icons.

Click and drag across the APIC GUI window from the db EPG to the app EPG.
The **Add Consumed Contract** dialog box is displayed.

Step 2 In the **Name** field, from the drop-down list, choose **sql** contract. Click **OK**.
This step enables the db EPG to provide the sql contract and the app EPG to consume the sql contract.

Step 3 Click and drag across the APIC GUI screen from the app ePG to the web EPG.
The **Add Consumed Contract** dialog box is displayed.

Step 4 In the **Name** field, from the drop-down list, choose **rmi** contract. Click **OK**.
This step enables the app EPG to provide the rmi contract and the web EPG to consume the rmi contract.

Step 5 Click the web EPG icon, and click the + sign in the **Provided Contracts** area.
The **Add Provided Contract** dialog box is displayed.

Step 6 In the **Name** field, from the drop-down list, choose **web** contract. Click **OK**. Click **Submit**.
You have created a three-tier application profile called OnlineStore.

Step 7 To verify, in the **Navigation** pane, navigate to and click **OnlineStore** under **Application Profiles**.
In the **Work** pane, you can see the three EPGs app, db, and web are displayed.

Step 8 In the **Work** pane, choose **Operational** > **Contracts**.
You can see the EPGs and contracts displayed in the order that they are consumed and provided.

Optimize Contract Performance

Optimize Contract Performance

Starting with Cisco APIC, Release 3.2, you can configure bidirectional contracts that support more efficient hardware TCAM storage of contract data. With optimization enabled, contract statistics for both directions are aggregated.

TCAM Optimization is supported on the second generation Cisco Nexus 9000 Series top of rack (TOR) switches, which are those with suffixes of EX, FX, and FX2, and later (for example, N9K-C93180LC-EX or N9K-C93180YC-FX).

To configure efficient TCAM contract data storage, you enable the following options:

- Mark the contracts to be applied in both directions between the provider and consumer.
- For filters with IP TCP or UDP protocols, enable the reverse port option.
- When configuring the contract subjects, select the **Enable Policy Compression** directive, which adds the `no_stats` option to the `action` attribute of the `actrl:Rule` managed object.

Limitations

With the **Enable Policy Compression** (`no_stats`) option selected, per-rule statistics are lost. However, combined rule statistics for both directions are present in the hardware statistics.

After upgrading to Cisco APIC 3.2(1), to add the `no_stats` option to a pre-upgrade contract subject (with filters or filter entries), you must delete the contract subject and reconfigure it with the **Enable Policy Compression** directive. Otherwise, compression does not occur.

For each contract with a bi-directional subject filter, Cisco NX-OS creates 2 rules:

- A rule with an `sPcTag` and `dPcTag` that is marked `direction=bi-dir`, which is programmed in hardware
- A rule marked with `direction=uni-dir-ignore` which is not programmed

Rules with the following settings are not compressed:

- Rules with priority other than `fully_qual`
- Opposite rules (`bi-dir` and `uni-dir-ignore` marked) with non-identical properties, such as **action** including **directives**, **prio, qos** or **markDscp**
- Rule with `Implicit` or `implarp` filters
- Rules with the actions `Deny`, `Redir`, `Copy`, or `Deny-log`

The following MO query output shows the two rules for a contract, that is considered for compression:

```
apic1# moquery -c actrlRule
Total Objects shown: 2

# actrl.Rule
scopeId            : 2588677
sPcTag             : 16388
dPcTag             : 49156
fltId              : 67
action             : no_stats,permit
actrlCfgFailedBmp  :
actrlCfgFailedTs   : 00:00:00:00.000
actrlCfgState      : 0
childAction        :
ctrctName          :
descr              :
direction          : bi-dir
dn                 : sys/actrl/scope-2588677/rule-2588677-s-16388-d-49156-f-67
id                 : 4112
lcOwn              : implicit
markDscp           : unspecified
modTs              : 2019-04-27T09:01:33.152-07:00
monPolDn           : uni/tn-common/monepg-default
name               :
nameAlias          :
operSt             : enabled
operStQual         :
prio               : fully_qual
qosGrp             : unspecified
rn                 : rule-2588677-s-16388-d-49156-f-67
status             :
type               : tenant


# actrl.Rule
scopeId            : 2588677
sPcTag             : 49156
dPcTag             : 16388
fltId              : 64
action             : no_stats,permit
actrlCfgFailedBmp  :
actrlCfgFailedTs   : 00:00:00:00.000
actrlCfgState      : 0
childAction        :
ctrctName          :
descr              :
direction          : uni-dir-ignore
dn                 : sys/actrl/scope-2588677/rule-2588677-s-49156-d-16388-f-64
id                 : 4126
lcOwn              : implicit
markDscp           : unspecified
modTs              : 2019-04-27T09:01:33.152-07:00
monPolDn           : uni/tn-common/monepg-default
name               :
nameAlias          :
operSt             : enabled
operStQual         :
prio               : fully_qual
qosGrp             : unspecified
rn                 : rule-2588677-s-49156-d-16388-f-64
status             :
type               : tenant
```

Table 1: Compression Matrix

Reverse Filter Port Enabled	TCP or UDP Source Port	TCP or UCP Destination Port	Compressed
Yes	Port A	Port B	Yes
Yes	Unspecified	Port B	Yes
Yes	Port A	Unspecified	Yes
Yes	Unspecified	Unspecified	Yes
No	Port A	Port B	No
No	Unspecified	Port B	No
No	Port A	Unspecified	No
No	Unspecified	Unspecified	Yes

Configure a Contract with Optimized TCAM Usage Using the GUI

This procedure describes how to configure a contract that optimizes TCAM storage of contract data on hardware.

Before you begin

- Create the tenant, VRF, and EPGs that will provide and consume the contract.
- Create one or more filters that define the traffic to be permitted or denied by this contract.

Procedure

Step 1 On the menu bar, choose **Tenants** and the tenant name on which you want to operate. In the **Navigation** pane, expand the *tenant-name* and **Contracts**.

Step 2 Right-click **Standard** > **Create Contract**.

Step 3 In the **Create Contract** dialog box, perform the following tasks:

a) In the **Name** field, enter the contract name.
b) Click the + icon next to **Subjects** to add a new subject.
c) In the **Create Contract Subject** dialog box, enter a subject name in the **Name** field.

Note
This step associates filters with the contract subject.

d) To enable the TCAM-contract usage optimization feature, ensure that **Apply Both Directions** and **Reverse Filter Ports** are enabled.
e) Click the + icon to expand **Filters**.

f) In the dialog box, from the drop-down menu, choose a default filter, a previously configured filter, or **Create Filter**.
g) In the **Directives** field, choose **Enable Policy Compression**
h) In the **Action** field, choose **Permit** or **Deny**.

Note
Currently, the **Deny** action is not supported. Optimization only occurs for the **Permit** action.

i) (Optional) In the **Priority** field, choose the priority level.
j) Click **Update**.

Step 4 In the **Create Contract Subject** dialog box, click **OK**.

Step 5 In the **Create Contract** dialog box, click **Submit**.

Contract and Subject Exceptions

Configuring Contract or Subject Exceptions for Contracts

In Cisco APIC Release 3.2(1), contracts between EPGs are enhanced to enable denying a subset of contract providers or consumers from participating in the contract. Inter-EPG contracts and Intra-EPG contracts are supported with this feature.

You can enable a provider EPG to communicate with all consumer EPGs except those that match criteria configured in a subject or contract exception. For example, if you want to enable an EPG to provide services to all EPGs for a tenant, except a subset, you can enable those EPGs to be excluded. To configure this, you create an exception in the contract or one of the subjects in the contract. The subset is then denied access to providing or consuming the contract.

Labels, counters, and permit and deny logs are supported with contracts and subject exceptions.

To apply an exception to all subjects in a contract, add the exception to the contract. To apply an exception only to a single subject in the contract, add the exception to the subject.

When adding filters to subjects, you can set the action of the filter (to permit or deny objects that match the filter criteria). Also for **Deny** filters, you can set the priority of the filter. **Permit** filters always have the default priority. Marking the subject-to-filter relation to deny automatically applies to each pair of EPGs where there is a match for the subject. Contracts and subjects can include multiple subject-to-filter relationships that can be independently set to permit or deny the objects that match the filters.

Exception Types

Contract and subject exceptions can be based on the following types and include regular expressions, such as the * wildcard:

Exception criteria exclude these objects as defined in the Consumer Regex and Provider Regex fields	Example	Description
Tenant	`<vzException consRegex= "common" field= "Tenant" name= "excep03" provRegex= "t1" />`	This example, excludes EPGs using the `common` tenant from consuming contracts provided by the `t1` tenant.
VRF	`<vzException consRegex= "ctx1" field= "Ctx" name= "excep05" provRegex= "ctx1" />`	This example excludes members of `ctx1` from consuming the services provided by the same VRF.
EPG	`<vzException consRegex= "EPgPa.*" field= "EPg" name= "excep03" provRegex= "EPg03" />`	The example assumes that multiple EPGs exist, with names starting with `EPGPa`, and they should all be denied as consumers for the contract provided by `EPg03`
Dn	`<vzException consRegex= "uni/tn-t36/ap-customer/epg-epg193" field= "Dn" name="excep04" provRegex= "uni/tn-t36/ap-customer/epg-epg200" />`	This example excludes `epg193` from consuming the contract provided by `epg200`.
Tag	`<vzException consRegex= "red" field= "Tag" name= "excep01" provRegex= "green" />`	The example excludes objects marked with the `red` tag from consuming and those marked with the `green` tag from participating in the contract.

Configure a Contract or Subject Exception Using the GUI

In this task, you configure a contract that will allow most of the EPGs to communicate, but deny access to a subset of them.

Before you begin

Configure the tenant, VRF, application profile, and EPGs that provide and consume the contract.

Procedure

Step 1 Click **Tenants** > **All Tenants** on the menu bar.

Step 2 Double-click the tenant in which you are creating the contract.

Step 3 On the navigation bar, expand **Contracts**, right-click **Filter**, and choose **Create Filter**.

A filter is essentially an Access Control List (ACL) that defines the traffic that is permitted or denied access through the contract. You can create multiple filters that define objects that can be permitted or denied.

Step 4 Enter the filter name and add the criteria that define the traffic to permit or deny, then click **Submit**.

Step 5 Right-click **Standard**, and choose **Create Contract**.

Step 6 Enter the contract name, set the scope, and click the + icon to add a subject.

Step 7 Repeat to add another subject.

Step 8 Click **Submit**

Step 9 To add an exception to all subjects in the contract, perform the following steps:

a) Click the contract, then click **Contract Exception**.
b) Add subjects and set them to be permitted or denied.
c) Click the + icon to add a contract exception.
d) Enter the exception name and type.
e) Add regular expressions in the **Consumer Regex** and **Provider Regex** fields to define the EPGs to be excluded from all subjects in the contract.

Step 10 To add an exception to one subject in the contract, perform the following steps:

a) Click the subject, then click **Subject Exception**.
b) Click the + icon to add a contract exception.
c) Enter the exception name and type.
d) Add regular expressions in the **Consumer Regex** and **Provider Regex** to define the EPGs to be excluded from all subjects in the contract.

Intra-EPG Contracts

Intra-EPG Contracts

You can configure contracts to control communication between EPGs. Beginning in Cisco APIC Release 3.0(1), you can also configure contracts within an EPG.

Without intra-EPG contracts, communication between endpoints in an EPG is all-or-nothing. Communication is unrestricted by default, or you can configure intra-EPG isolation to bar any communication between endpoints.

However, with intra-EPG contracts, you can control communication between endpoints in the same EPG, allowing some traffic and barring the rest. For example, you may want to allow web traffic but block the rest. Or you can allow all ICMP traffic and TCP port 22 traffic while blocking all other traffic.

Guidelines and Limitations for Intra-EPG Contracts

Observe the following guidelines and limitations when planning intra-EPG contracts:

- Intra-EPG contracts can be configured for application EPGs and microsegment EPGs (uSegs) on VMware VDS, Open vSwitch (OVS), and baremetal servers.

Note OVS is available in the Kubernetes integration with Cisco Application Centric Infrastructure (ACI) feature. In Kubernetes, you can create EPGs and assign namespaces to them. You can then apply intra-EPG policies to the EPGs in Cisco Application Policy Infrastructure Controller (APIC) as you would for VMware VDS or baremetal servers.

- Intra-EPG contracts require that the leaf switch support proxy Address Resolution Protocol (ARP). Intra-EPG contracts are supported on Cisco Nexus 9000 Series switches with EX or FX at the end of their model name or later models.
- Intra-EPG Contracts are not supported in Cisco Application Virtual Switch, Cisco ACI Virtual Edge, and Microsoft domains. Attempting to set intra-EPG contracts to be enforced in these domains may cause ports to go into a blocked state.
- Intra-EPG contracts in service graphs:
  - A service graph cannot be associated with a subject of an intra-EPG contract that has an action of deny.
  - Support for intra-EPG contracts in service graphs is limited to single node one-arm mode policy-based redirect and copy service.
- Beginning in Cisco APIC release 5.2(1), intra-EPG contracts are supported on L3Out EPGs.
  - The action can be **permit**, **deny**, or **redirect**. The **redirect** action requires a service graph with policy-based redirect (PBR).
  - An L3Out EPG with an IP address and subnet of 0.0.0.0/0 or 0::0 cannot use an intra-EPG contract nor intra-EPG isolation. The Cisco APIC raises a fault in these cases. However, you can instead use an IP address and subnet of 0.0.0.0/1 and 128.0.0.0/1 for the L3Out EPG to catch all traffic.
  - Unlike an intra-EPG contract on an EPG, the implicit deny rule is not automatically added for an intra-EPG contract on an L3Out EPG. You must enable intra-EPG isolation to deny other traffic. Intra-EPG isolation on an L3Out EPG works only when the VRF instance is in the enforced mode.
  - Cisco ACI cannot control how traffic reaches the Cisco ACI border leaf switch for intra L3Out enforcement.

Adding an Intra-EPG Contract to an Application EPG Using the GUI

After you configure a contract, you can add the contract to an EPG as an intra-EPG contract. The procedure is the same for VMware VDS, OVS, and baremetal servers.

Before you begin

- You must have an application EPG configured.
- You must have a contract with filters configured for this application. See Creating a Contract Using the GUI, on page 32.

Procedure

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

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

Step 3 Complete one of the following sets of steps, depending on the type of EPG:

If you want to apply an intra-EPG contract to...	Then...
An application EPG	**a.** In the Navigation pane, expand ***tenant_name*** > **Application Profiles** > ***application profile*** > **Application EPGs** > ***epg***. **b.** Right-click the **Contracts** folder and then choose **Add Intra-EPG Contract**. **c.** In the **Add Intra Ext-EPG Contract** dialog box, from the **Contract** drop-down list, choose an existing contract or create a new contract. **d.** Click **Submit**.
A uSeg EPG	**a.** In the Navigation pane, expand ***tenant_name*** > **Application Profiles** > ***application profile*** > **uSeg EPGs** > ***epg***. **b.** Right-click the **Contracts** folder and then choose **Add Intra-EPG Contract**. **c.** In the **Add Intra Ext-EPG Contract** dialog box, from the **Contract** drop-down list, choose an existing contract or create a new contract. **d.** Click **Submit**.
An L3Out EPG	**a.** In the Navigation pane, choose ***tenant_name*** > **Networking** > **L3Outs** > ***L3Out_name*** > **External EPGs** > ***ext_epg_name***. **b.** In the Work pane, for **Intra Ext-EPG Isolation**, choose **Enforced**. **c.** Click **Submit**. **d.** In the Work pane, choose the **Policy** > **Contracts** tab. **e.** In the action menu, choose **Add Intra Ext-EPG Contract**. **f.** In the **Add Intra Ext-EPG Contract** dialog box, from the **Contract** drop-down list, choose an existing contract or create a new contract. **g.** Click **Submit**. The chosen contract appears in the **Contract Type: Intra EPG Contract** section of the Work pane.

Adding an Intra-EPG Contract to an Application EPG Using the NX-OS-Style CLI

After you configure a contract, you can configure the contract as an intra-EPG contract. The procedure is the same for VMware VDS, OVS, and baremetal servers.

Before you begin

- You must have an EPG configured.
- You must have a contract with filters configured.

Procedure

Step 1 Enter the configuration mode.

Example:

```
apic1# configure
```

Step 2 Create or choose a tenant.

Example:

```
apic1(config)# tenant Tenant-l3out
```

Step 3 Create or choose an external Layer 3 EPG.

Example:

```
apic1(config-tenant)# external-l3 epg ext-epg
```

Step 4 Bind the external EPG to a VRF instance.

Example:

```
apic1(config-tenant-l3ext-epg)# vrf member vrf1
```

Step 5 Enable intra-EPG isolation.

Example:

```
(config-tenant-l3ext-epg)# isolation enforce
```

Afterward, if necessary, you can disable intra-EPG isolation by preceding the command with `no`.

Example:

```
(config-tenant-l3ext-epg)# no isolation enforce
```

Step 6 Assign a contract to the intra-external EPG to allow the desired traffic between the endpoints.

Example:

```
apic1(config-tenant-l3ext-epg)# contract intra-epg contr-intra
```

Adding an Intra-EPG Contract to an Application EPG Using the REST API

After you configure a contract, you can add the contract to an EPG as an intra-EPG contract. The procedure is the same for VMware VDS, OVS, and baremetal servers.

Before you begin

- You must have configured an EPG.
- You must have configured a contract with filters.

Procedure

Step 1 Configure the selectors using an XML POST request similar to the following example:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<polUni>
  <infraInfra>
    <infraAccPortP name="Ports-1-12" status="deleted"/>

    <!-- VMM VLAN range -->
    <fvnsVlanInstP name="test" allocMode="dynamic">
      <fvnsEncapBlk name="encap" from="vlan-10" to="vlan-100"/>
    </fvnsVlanInstP>

    <!-- Static VLAN range -->
    <fvnsVlanInstP name="test" allocMode="static">
      <fvnsEncapBlk name="default" from="vlan-101" to="vlan-4095"/>
    </fvnsVlanInstP>

    <infraAttEntityP name="test">
      <infraRsDomP tDn="uni/phys-test"/>
      <infraRsDomP tDn="uni/l3dom-test"/>
      <infraRsDomP tDn="uni/vmmp-VMware/dom-test"/>
    </infraAttEntityP>

    <!-- Node profile -->
    <infraNodeP name="test">
      <infraLeafS name="test" type="range">
        <infraNodeBlk name="default" from_="101" to_="102"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-test"/>
    </infraNodeP>

    <!-- Port profile -->
    <infraAccPortP name="test">
      <!-- 12 regular ports -->
      <infraHPortS name="ports1Through12" type="range">
        <infraPortBlk name="default" fromCard="1" toCard="1" fromPort="1" toPort="12"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-test"/>
      </infraHPortS>

      <!-- 2 ports in PC -->
      <infraHPortS name="portsForPc1" type="range">
        <infraPortBlk name="default" fromCard="1" toCard="1" fromPort="13" toPort="14"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-testPc"/>
      </infraHPortS>
```

```
  <!-- 2 ports in PC -->
  <infraHPortS name="portsForPc2" type="range">
    <infraPortBlk name="blk1" fromCard="1" toCard="1" fromPort="15" toPort="16"/>
    <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-pc"/>
  </infraHPortS>

  <!-- 2 ports in PC for FEX -->
  <infraHPortS name="portsForFex" type="range">
    <infraPortBlk name="blk1" fromCard="1" toCard="1" fromPort="17" toPort="18"/>
    <infraRsAccBaseGrp tDn="uni/infra/fexprof-default/fexbundle-test" fexId="111"/>
  </infraHPortS>

  <!-- 2 ports in PC for VPC -->
  <infraHPortS name="portsForVpc" type="range">
    <infraPortBlk name="blk1" fromCard="1" toCard="1" fromPort="19" toPort="20"/>
    <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-testVpc"/>
  </infraHPortS>
</infraAccPortP>

<!-- FEX profile -->
<infraFexP name="default">
  <infraFexBndlGrp name="default"/>

  <!-- 12 FEX ports -->
  <infraHPortS name="ports1Through12" type="range">
    <infraPortBlk name="default" fromCard="1" toCard="1" fromPort="1" toPort="12"/>
    <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-test"/>
  </infraHPortS>

  <!-- 3 ports in FEX PC -->
  <infraHPortS name="portsForPc" type="range">
    <infraPortBlk name="blk1" fromCard="1" toCard="1" fromPort="13" toPort="16"/>
    <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-testPcOnFex"/>
  </infraHPortS>

  <!-- 3 ports in FEX VPC -->
  <infraHPortS name="portsForVpc" type="range">
    <infraPortBlk name="blk1" fromCard="1" toCard="1" fromPort="17" toPort="19"/>
    <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-testVpcOnFex"/>
  </infraHPortS>
</infraFexP>

<!-- Functional profile -->
<infraFuncP>
  <!-- Regular port group -->
  <infraAccPortGrp name="test">
    <infraRsAttEntP tDn="uni/infra/attentp-test"/>
  </infraAccPortGrp>

  <!-- PC -->
  <infraAccBndlGrp name="testPc" lagT="link">
    <infraRsLacpPol tnLacpLagPolName="testPc"/>
    <infraRsAttEntP tDn="uni/infra/attentp-test"/>
  </infraAccBndlGrp>

  <!-- VPC -->
  <infraAccBndlGrp name="testVpc" lagT="node">
    <infraRsLacpPol tnLacpLagPolName="testVpc"/>
    <infraRsAttEntP tDn="uni/infra/attentp-test"/>
  </infraAccBndlGrp>

  <!-- PC on FEX -->
  <infraAccBndlGrp name="testPcOnFex" lagT="link">
```

```
        <infraRsLacpPol tnLacpLagPolName="testPcOnFex"/>
        <infraRsAttEntP tDn="uni/infra/attentp-test"/>
      </infraAccBndlGrp>

      <!-- VPC on FEX -->
      <infraAccBndlGrp name="testVpcOnFex" lagT="node">
        <infraRsLacpPol tnLacpLagPolName="testVpcOnFex"/>
        <infraRsAttEntP tDn="uni/infra/attentp-test"/>
      </infraAccBndlGrp>
    </infraFuncP>

    <!-- Link aggregation policies -->
    <lacpLagPol name="testPc" minLinks='1' maxLinks='10'/>
    <lacpLagPol name="testVpc" minLinks='1' maxLinks='10'/>
    <lacpLagPol name="testPcOnFex" minLinks='2' maxLinks='5'/>
    <lacpLagPol name="testVpcOnFex" minLinks='2' maxLinks='10'/>
  </infraInfra>

  <fabricInst>
    <fabricProtPol name="testVpc">
      <fabricExplicitGEp name="testVpc" id="101">
        <fabricNodePEp id="101"/>
        <fabricNodePEp id="102"/>
      </fabricExplicitGEp>
    </fabricProtPol>
  </fabricInst>

  <physDomP name="test">
    <infraRsVlanNs tDn="uni/infra/vlanns-test-static"/>
  </physDomP>

  <l3extDomP name="test">
    <infraRsVlanNs tDn="uni/infra/vlanns-test-static"/>
  </l3extDomP>
</polUni>
```

Step 2 Configure the tenant using an XML POST request similar to the following example:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<polUni>
   <fvTenant name="Tenant-l3out">
      <vzBrCP intent="install" name="contr-intra" scope="context">
         <vzSubj consMatchT="AtleastOne" name="subj" revFltPorts="yes">
            <vzRsSubjFiltAtt action="permit" priorityOverride="default"
              tnVzFilterName="flt-ssh" />
         </vzSubj>
      </vzBrCP>
      <vzBrCP intent="install" name="contr2" scope="context">
         <vzSubj consMatchT="AtleastOne" name="contr2-subj" revFltPorts="yes">
            <vzRsSubjFiltAtt action="permit" priorityOverride="default"
              tnVzFilterName="flt-ftp" />
         </vzSubj>
      </vzBrCP>
      <vzBrCP intent="install" name="contr1" scope="context">
         <vzSubj consMatchT="AtleastOne" name="subj-http" revFltPorts="yes">
            <vzRsSubjFiltAtt action="deny" priorityOverride="default"
              tnVzFilterName="flt-http" />
         </vzSubj>
      </vzBrCP>
      <l3extOut enforceRtctrl="export" mplsEnabled="no" name="l3out1">
         <l3extRsL3DomAtt tDn="uni/l3dom-test" />
         <l3extRsEctx tnFvCtxName="vrf1" />
         <l3extLNodeP name="l3out1_nodeProfile" tag="yellow-green">
```

```
      <l3extRsNodeL3OutAtt rtrId="172.16.0.1" rtrIdLoopBack="yes"
        tDn="topology/pod-1/node-101" />
      <l3extLIfP name="l3out1_interfaceProfile" tag="yellow-green">
         <l3extRsPathL3OutAtt addr="192.168.15.1/24" autostate="disabled"
           encap="unknown" encapScope="local" ifInstT="l3-port" ipv6Dad="enabled"
           isMultiPodDirect="no" llAddr="::" mac="00:22:BD:F8:19:FF"
           mode="regular" mtu="inherit"
           tDn="topology/pod-1/paths-101/pathep-[eth1/10]" />
      </l3extLIfP>
   </l3extLNodeP>

   <!--
     Set pcEnfPref to "enforced" to enable intra-Ext-EPG isolation.
     Set pcEnfPref to "unenforced" to disable intra-Ext-EPG isolation.
   -->
   <l3extInstP floodOnEncap="disabled" matchT="AtleastOne"
     name="l3epg1" pcEnfPref="unenforced" prefGrMemb="exclude">
      <l3extSubnet ip="172.16.0.0/16" scope="import-security" />
      <fvRsCons tnVzBrCPName="contr2" />
      <fvRsIntraEpg tnVzBrCPName="contr-intra" />
   </l3extInstP>
</l3extOut>
<fvCtx bdEnforcedEnable="no" ipDataPlaneLearning="enabled" knwMcastAct="permit"
  name="vrf1" pcEnfDir="egress" pcEnfPref="unenforced" vrfIndex="0">
   <fvRsVrfValidationPol />
   <vzAny matchT="AtleastOne" prefGrMemb="disabled" />
</fvCtx>
<fvBD OptimizeWanBandwidth="no" arpFlood="yes" epClear="no" hostBasedRouting="no"
  intersiteBumTrafficAllow="no" intersiteL2Stretch="no" ipLearning="yes"
  ipv6McastAllow="no" limitIpLearnToSubnets="yes" llAddr="::"
  mac="00:22:BD:F8:19:FF" mcastAllow="no" multiDstPktAct="bd-flood" name="bd-web"
  type="regular" unicastRoute="yes" unkMacUcastAct="proxy" unkMcastAct="flood"
  v6unkMcastAct="flood" vmac="not-applicable">
   <fvSubnet ip="192.168.1.254/24" ipDPLearning="enabled" preferred="no"
     scope="private" virtual="no" />
   <fvRsCtx tnFvCtxName="vrf1" />
   <fvRsBdToEpRet resolveAct="resolve" />
</fvBD>
<fvBD OptimizeWanBandwidth="no" arpFlood="yes" epClear="no" hostBasedRouting="no"
  intersiteBumTrafficAllow="no" intersiteL2Stretch="no" ipLearning="yes"
  ipv6McastAllow="no" limitIpLearnToSubnets="yes" llAddr="::"
  mac="00:22:BD:F8:19:FF" mcastAllow="no" multiDstPktAct="bd-flood" name="bd-app"
  type="regular" unicastRoute="yes" unkMacUcastAct="proxy" unkMcastAct="flood"
  v6unkMcastAct="flood" vmac="not-applicable">
   <fvSubnet ip="192.168.2.254/24" ipDPLearning="enabled" preferred="no"
     scope="private" virtual="no" />
   <fvRsCtx tnFvCtxName="vrf1" />
   <fvRsBdToEpRet resolveAct="resolve" />
</fvBD>
<vzFilter name="flt-ftp">
   <vzEntry applyToFrag="no" arpOpc="unspecified" dFromPort="ftpData"
     dToPort="ftpData" etherT="ipv4" icmpv4T="unspecified" icmpv6T="unspecified"
     matchDscp="unspecified" name="ftp" prot="tcp" sFromPort="unspecified"
     sToPort="unspecified" stateful="no" />
</vzFilter>
<vzFilter name="flt-ssh">
   <vzEntry applyToFrag="no" arpOpc="unspecified" dFromPort="ssh" dToPort="ssh"
     etherT="ipv4" icmpv4T="unspecified" icmpv6T="unspecified"
     matchDscp="unspecified" name="ssh" prot="tcp" sFromPort="unspecified"
     sToPort="unspecified" stateful="no" />
</vzFilter>
<vzFilter name="flt-http">
   <vzEntry applyToFrag="no" arpOpc="unspecified" dFromPort="http" dToPort="http"
     etherT="ipv4" icmpv4T="unspecified" icmpv6T="unspecified"
```

```
            matchDscp="unspecified" name="flt1" prot="tcp" sFromPort="unspecified"
            sToPort="unspecified" stateful="no" />
        </vzFilter>
        <fvAp name="ap-app1">
           <fvAEPg floodOnEncap="disabled" hasMcastSource="no" isAttrBasedEPg="no"
             matchT="AtleastOne" name="epg-app" pcEnfPref="unenforced"
             prefGrMemb="exclude" shutdown="no">
              <fvRsProv intent="install" matchT="AtleastOne" tnVzBrCPName="contr2" />
              <fvRsProv intent="install" matchT="AtleastOne" tnVzBrCPName="contr1" />
              <fvRsPathAtt encap="vlan-103" instrImedcy="immediate" mode="native"
                primaryEncap="unknown" tDn="topology/pod-1/paths-101/pathep-[eth1/3]" />
              <fvRsDomAtt bindingType="none" classPref="encap" encap="unknown"
                encapMode="auto" epgCos="Cos0" epgCosPref="disabled" instrImedcy="lazy"
                netflowDir="both" netflowPref="disabled" numPorts="0" portAllocation="none"
                primaryEncap="unknown" primaryEncapInner="unknown" resImedcy="immediate"
                secondaryEncapInner="unknown" switchingMode="native" tDn="uni/phys-test"
                untagged="no" vnetOnly="no" />
              <fvRsBd tnFvBDName="bd-app" />
           </fvAEPg>
           <fvAEPg floodOnEncap="disabled" hasMcastSource="no"
             isAttrBasedEPg="no" matchT="AtleastOne" name="epg-web" pcEnfPref="unenforced"
             prefGrMemb="exclude" shutdown="no">
              <fvRsPathAtt encap="vlan-104" instrImedcy="immediate" mode="native"
                primaryEncap="unknown" tDn="topology/pod-1/paths-101/pathep-[eth1/4]" />
              <fvRsDomAtt bindingType="none" classPref="encap" encap="unknown"
                encapMode="auto" epgCos="Cos0" epgCosPref="disabled" instrImedcy="lazy"
                netflowDir="both" netflowPref="disabled" numPorts="0" portAllocation="none"
                primaryEncap="unknown" primaryEncapInner="unknown" resImedcy="immediate"
                secondaryEncapInner="unknown" switchingMode="native" tDn="uni/phys-test"
                untagged="no" vnetOnly="no" />
              <fvRsCons intent="install" tnVzBrCPName="contr1" />
              <fvRsBd tnFvBDName="bd-web" />
           </fvAEPg>
        </fvAp>
     </fvTenant>
</polUni>
```

EPG Contract Inheritance

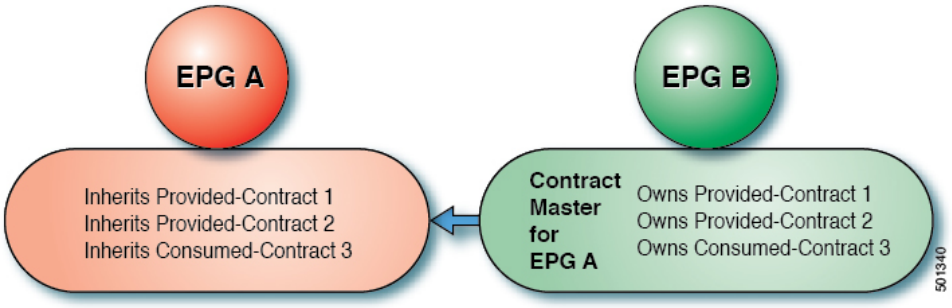
About Contract Inheritance

To streamline associating contracts to new EPGs, you can now enable an EPG to inherit all the (provided and consumed) contracts associated directly to another EPG in the same tenant. Contract inheritance can be configured for application, microsegmented, L2Out, and L3Out EPGs.

With Release 3.x, you can also configure contract inheritance for Inter-EPG contracts, both provided and consumed. Inter-EPG contracts are supported on Cisco Nexus 9000 Series switches with EX or FX at the end of their model name or later models.

You can enable an EPG to inherit all the contracts associated directly to another EPG, using the APIC GUI, NX-OS style CLI, and the REST API.

Figure 9: Contract Inheritance



In the diagram above, EPG A is configured to inherit Provided-Contract 1 and 2 and Consumed-Contract 3 from EPG B (contract master for EPG A).

Use the following guidelines when configuring contract inheritance:

- Contract inheritance can be configured for application, microsegmented (uSeg), external L2Out EPGs, and external L3Out EPGs. The relationships must be between EPGs of the same type.
- Both provided and consumed contracts are inherited from the contract master when the relationship is established.
- Contract masters and the EPGs inheriting contracts must be within the same tenant.
- Changes to the masters' contracts are propagated to all the inheritors. If a new contract is added to the master, it is also added to the inheritors.
- An EPG can inherit contracts from multiple contract masters.
- Contract inheritance is only supported to a single level (cannot be chained) and a contract master cannot inherit contracts.
- Labels with contract inheritance is supported. When EPG A inherits a contract from EPG B, if different subject labels are configured under EPG A and EPG B, APIC uses the label configured under EPG B for the contract inherited from EPG B. APIC uses the label configured under EPG A for the contract where EPG A is directly involved.
- Whether an EPG is directly associated to a contract or inherits a contract, it consumes entries in TCAM. So contract scale guidelines still apply. For more information, see the *Verified Scalability Guide* for your release.
- vzAny security contracts and taboo contracts are not supported.
- Beginning in Cisco APIC releases 5.0(1) and 4.2(6), contract inheritance with a service graph is supported if the contract and EPGs are in the same tenant.

For information about configuring Contract Inheritance and viewing inherited and standalone contracts, see *Cisco APIC Basic Configuration Guide*.

Configuring EPG Contract Inheritance Using the GUI

Configuring Application EPG Contract Inheritance Using the GUI

To configure contract inheritance for an application EPG, in the APIC Basic or Advanced mode GUI, use the following steps.

Before you begin

Configure the tenant and application profile to be used by the EPGs.

Optional. Configure the bridge domain to be used by the EPG that will inherit contracts.

Configure at least one application EPG, to serve as the **EPG Contract Master**.

Configure the contracts to be shared, and associate them to the contract master.

Procedure

Step 1 Navigate to **Tenants** > ***tenant-name*** > **Application Profiles**, and expand ***AP-name***

Step 2 Right-click **Application EPGs** and select **Create Application EPG**.

Step 3 Type the name of the EPG that will inherit contracts from the **EPG Contract Master**.

Step 4 On the **Bridge Domain** field, select the common/default bridge domain or a previously created bridge domain, or create a bridge domain for this EPG.

Step 5 On the **EPG Contract Master** field, click the + symbol, select the previously configured Application Profile and EPG, and click **Update**.

Step 6 Click **Finish**.

Step 7 To view information about the EPG, including the contract master, navigate to **Tenants** > ***tenant-name*** > **Application Profiles** > ***AP-name*** > **Application EPGs** > ***EPG-name***. To view the **EPG Contract Master**, click **General**.

Step 8 To view information about the inherited contracts, expand ***EPG-name*** and click **Contracts.**

Configuring uSeg EPG Contract Inheritance Using the GUI

To configure contract inheritance for a uSeg EPG, in the APIC Basic or Advanced mode GUI, use the following steps.

Before you begin

Configure the tenant and application profile to be used by the EPGs.

Optional. Configure the bridge domain to be used by the EPG that will inherit contracts.

Configure the uSeg EPG, to serve as the **EPG Contract Master**.

Configure the contracts to be shared, and associate them to the contract master.

Procedure

Step 1 Navigate to **Tenants** > ***tenant-name*** > **Application Profiles**, expand ***AP-name***.

Step 2 Right-click **uSeg EPGs** and select **Create uSeg EPG**.

Step 3 Type the name of the EPG that will inherit contracts from the contract master.

Step 4 On the **Bridge Domain** field, select the common/default bridge domain or a previously created bridge domain, or create a bridge domain for this EPG.

Step 5 Click ***uSeg-EPG-name***. In the **EPG Contract Master** field, click the + symbol, select the Application Profile and EPG (to serve as contract master), and click **Update**.

Step 6 Click **Finish**.

Step 7 To view information about the contracts, navigate to **Tenants** > ***tenant-name*** > **Application Profiles** > ***AP-name*** > **uSeg EPGs** > , expand the ***EPG-name*** and click **Contracts.**.

Configuring L2Out EPG Contract Inheritance Using the GUI

To configure contract inheritance for an external L2Out EPG, in the Cisco Application Policy Infrastructure Controller (APIC) GUI, perform the following steps.

Before you begin

Configure the tenant and application profile to be used by the EPGs.

Configure a Layer 2 Outside (L2Out) and the external L2Out EPG (L2extInstP) that will serve as the **L2Out Contract Master**.

Configure the contracts to be shared, and associate them to the contract master.

Procedure

Step 1 Navigate to **Tenants** > ***tenant-name*** > **Networking** > **L2Outs**.

Step 2 Expand the ***L2Out-name***.

Step 3 Right-click **External EPGs** and choose **Create External EPG.**

Step 4 Type the name of the external network and optionally add other attributes.

Step 5 Click **Submit**.

Step 6 Expand **External EPGs**.

Step 7 Click the ***external-epg-name***.

Step 8 In the **External EPG** panel, click the + symbol on the **L2Out Contract Masters** field.

Step 9 Select the L2Out and the L2Out contract master for this external L2Out EPG.

Step 10 Click **Update**.

Step 11 To view the contracts inherited by this external L2Out EPG, click on the external EPG name and click **Contracts** > **Inherited Contracts**.

Configuring External L3Out EPG Contract Inheritance Using the GUI

To configure contract inheritance for an external L3Out EPG, in the Cisco Application Policy Infrastructure Controller (APIC) GUI, use the following steps.

Before you begin

Configure the tenant and application profile to be used by the EPGs.

Configure an external routed network (L3Out) and the external L3Out EPG (L3extInstP) that will serve as the **L3Out Contract Master**.

Configure the contracts to be shared, and associate them to the contract master.

Procedure

Step 1 To configure contract inheritance for an external L3Out EPG, navigate to **Tenants** > ***tenant-name*** > **Networking** > **L3Outs**.

Step 2 Expand the ***L3Out-name*** leading to the external L3Out EPG.

Step 3 Right-click **External EPGs** and select **Create External EPG.**

Step 4 Type the name of the external EPG and optionally add subnets and other attributes.

Step 5 Click **Submit**.

Step 6 Expand **Networks**.

Step 7 Click the ***network-name***.

Step 8 In the **External EPG** panel, click the + symbol on the **L3Out Contract Masters** field.

Step 9 Choose the L3Out and external EPG to serve as L3Out contract master for this external L3Out EPG.

Step 10 Click **Update**.

Step 11 To view the contracts inherited by this external L3Out EPG, click on the external EPG name and click **Contracts** > **Inherited Contracts**.

Contract Preferred Groups

About Contract Preferred Groups

There are two types of policy enforcements available for EPGs in a VRF with a contract preferred group configured:

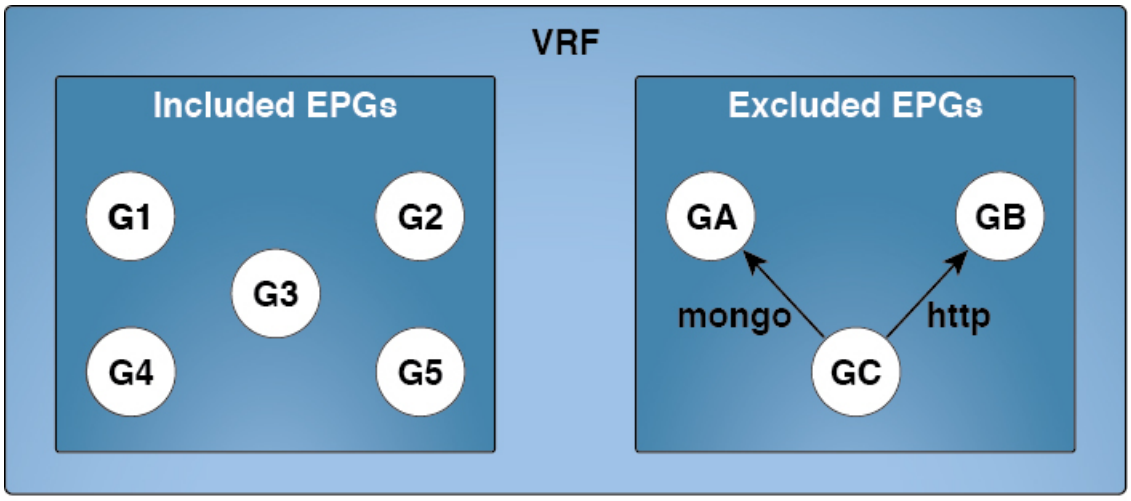
- Included EPGs: EPGs can freely communicate with each other without contracts, if they have membership in a contract preferred group. This is based on the source-any-destination-any-permit default rule.
- Excluded EPGs: EPGs that are not members of preferred groups require contracts to communicate with each other. Otherwise, the default source-any-destination-any-deny rule applies.

The contract preferred group feature enables greater control of communication between EPGs in a VRF. If most of the EPGs in the VRF should have open communication, but a few should only have limited

communication with the other EPGs, you can configure a combination of a contract preferred group and contracts with filters to control inter-EPG communication precisely.

EPGs that are excluded from the preferred group can only communicate with other EPGs if there is a contract in place to override the source-any-destination-any-deny default rule.

Figure 10: Contract Preferred Group Overview



Source	Destination	Filter	Action
any	any	implicit	permit
GA	any	implicit	deny
any	GA	implicit	deny
GB	any	implicit	deny
any	GB	implicit	deny
GC	any	implicit	deny
any	GC	implicit	deny

Source	Destination	Filter	Action
GC	GA	mongo	permit
GA	GC	mongo	permit
GC	GB	http	permit
GB	GC	http	permit

Limitations

The following limitations apply to contract preferred groups:

- In topologies where an L3Out and application EPG are configured in a Contract Preferred Group, and the EPG is deployed only on a VPC, you may find that only one leaf switch in the VPC has the prefix entry for the L3Out. In this situation, the other leaf switch in the VPC does not have the entry, and therefore drops the traffic.

  To workaround this issue, you can do one of the following:

  - Disable and reenable the contract group in the VRF
  - Delete and recreate the prefix entries for the L3Out EPG

- Also, where the provider or consumer EPG in a service graph contract is included in a contract group, the shadow EPG can not be excluded from the contract group. The shadow EPG will be permitted in the contract group, but it does not trigger contract group policy deployment on the node where the shadow

EPG is deployed. To download the contract group policy to the node, you deploy a dummy EPG within the contract group .

Due to CSCvm63145, an EPG in a Contract Preferred Group can consume a shared service contract, but cannot be a provider for a shared service contract with an L3Out EPG as consumer.

Guidelines for Contract Preferred Groups

When configuring contract preferred groups, refer to the following guidelines:

- If the (s, g) entry is installed on a border leaf switch, you might see drops in unicast traffic that comes from the fabric to this source outside the fabric when the following conditions are met:
  - Preferred group is used on the L3Out EPG
  - Unicast routing table for the source is using the default route 0.0.0.0/0

  This behavior is expected.
- Contract Preferred Group-included EPGs are not supported with a 0/0 prefix in external EPG (InstP). If, for the external EPG (InstP) to Tenant EPG, a 0/0 prefix is required with the use of Contract Preferred Group, then 0/0 can be split to 0/1 and 128/1.
- Contract Preferred Group-EPGs are not supported with the GOLF feature. Communication between an application EPG and the L3Out EPG for GOLF must be governed by explicit contracts.

Configuring Contract Preferred Groups Using the GUI

Before you begin

Create the tenants and VRF, and EPGs that will consume the contract preferred group.

Procedure

Step 1 On the menu bar, click **Tenants** > ***tenant_name***.

Step 2 In the **Navigation** pane, expand the tenant, **Networking**, and **VRFs**.

Step 3 Click the VRF name for which you are configuring the contract preferred group.

Step 4 In the **Preferred Group Member** field, click **Enabled**.

Step 5 Click **Submit**.

Step 6 In the **Navigation** pane, expand **Application Profiles** and create or expand an application profile for the tenant VRF.

Step 7 Expand **Application EPGs** and click the EPG that will consume the contract preferred group.

Step 8 Select the **Policy** and **General** tab.

Step 9 In the **Preferred Group Member** field, click **Include**.

Step 10 Click **Submit**.

What to do next

Enable membership in the preferred group for other EPGs that should have unlimited communication with this EPG. You can also configure appropriate contracts to control communication between the EPGs in the preferred group and other EPGs that may not be members.

Note If you want to support preferred group members through L4-L7 service graphs, you must create a L4-L7 service EPG policy. For more information regarding creating an L4-L7 Service EPG Policy, see Creating an L4-L7 Service EPG Policy Using the GUI, on page 53.

Creating an L4-L7 Service EPG Policy Using the GUI

This task creates a policy that defines if EPGs are to be included in, or excluded from, a preferred group. Preferred groups membership allows endpoints to communicate with each other without requiring a contract. After the policy is created, it can be selected during the application of a service graph template to the EPGs.

Before you begin

You must have configured a tenant.

Procedure

Step 1 On the Menu bar, choose **Tenants** > ***tenant_name***.

Step 2 In the Navigation pane, choose **Policies** > **Protocol** > **L4-L7 Service EPG Policy**.

Step 3 In the Navigation pane, right-click **L4-L7 Service EPG Policy** and choose **Create L4-L7 Service EPG Policy**.

The Create L4-L7 Service EPG Policy dialog box appears.

Step 4 Enter a unique name for the policy in the **Name** field.

Step 5 Optional. Enter a description of the policy in the **Description** field.

Step 6 Choose whether to exclude or include EPGs as preferred members in the **Preferred Group Member** field.

Step 7 Click **Submit**.

The newly created policy appears in the L4-L7 Service EPG Policy Work pane list. To edit a policy in the Work pane, double-click the list line containing the policy.

What to do next

The new L4-L7 service EPG policy can now be selected in a service graph template when applying the graph to EPGs. Refer to Applying a Service Graph Template to Endpoint Groups Using the GUI in the Using the GUI chapter of the *Cisco APIC Layer 4 to Layer 7 Services Deployment Guide*.

Contracts with Permit and Deny Rules

About Contracts with Permit and Deny Rules

Starting with the Cisco Application Policy Infrastructure Controller (Cisco APIC) release 3.2, You can configure contracts with both permit and deny actions, instead of just permit. You can configure the deny action with different priorities: default, highest, medium and lowest.

Rule conflicts are resolved as follows:

- The implicit deny has the lowest priority of all rules.
- Contracts between vzAny have higher priority than the implicit deny.
- Contracts between specific EPG pairs win over contracts with vzAny, because EPG-to-EPG contract rules have higher priority than vzAny-to-vzAny rules.
- Deny rules with the default priority for a contract between a specific EPG pair have the same level of priority as the permit rules for that EPG pair. When traffic matches both a permit and a deny rule with the same priority, the deny rule wins.
- Deny rules with the default priority for a contract between vzAny has the same level of priority as the permit rules for the vzAny pair. When traffic matches both a permit and a deny rule with the same priority, the deny rule wins.
- Deny rules with the highest priority are handled at the same level as EPG-to-EPG contracts.
- Deny rules with medium priority are handled at the same level as vzAny-to-EPG contracts.
- Deny rules with the lowest priority are handled at the same level as vzAny-to-vzAny contracts.
- If the deny priority is lowered in a contract between EPGs, a permit rule match between EPGs would win over deny.

Enabling ACL Contract Permit and Deny Logging Using the GUI

The following steps show how to enable contract permit and deny logging using the GUI:

Note The tenant that contains the permit logging is the tenant that contains the VRF that the EPG is associated to. This will not necessarily be the same tenant as the EPG or its associated contracts.

Procedure

Step 1 On the menu bar, choose **Tenants** > **<tenant name>**.

Step 2 In the **Navigation** pane, expand **Contracts**, right-click **Standard**, and choose **Create Contract**.

Step 3 In the **Create Contract** dialog box, perform the following actions:

a) In the **Name** field, type the name for the contract.
b) In the **Scope** field, choose the scope for it (VRF, Tenant, or Global).
c) Optional. Set the target DSCP or QoS class to be applied to the contract.
d) Click the + icon to expand **Subjects**.

Step 4 In the Create Contract Subject dialog box, perform the following actions:

Step 5 Enter the name of the subject and an optional description.

Step 6 Optional. From the drop-down list for the target DSCP, select the DSCP to be applied to the subject.

Step 7 Leave **Apply Both Directions** checked, unless you want the contract to only be applied from the consumer to the provider, instead of in both directions.

Step 8 Leave **Reverse Filter Ports** checked if you unchecked **Apply Both Directions** to swap the Layer 4 source and destination ports so that the rule is applied from the provider to the consumer.

Step 9 Click the + icon to expand **Filters**.

Step 10 In the **Name** drop-down list, choose an option; for example, click **arp**, **default**, **est**, or **icmp**, or choose a previously configured filter.

Step 11 In the **Directives** drop-down list, click **log**.

Step 12 (Optional) Change the Action to be taken with this subject to **Deny** (or leave the action to the default, **Permit**.

With Directive: log enabled, if the action for this subject is **Permit**, ACL permit logs track the flows and packets that are controlled by the subject and contract. If the action for this subject is **Deny**, ACL deny logs track the flows and packets.

Step 13 (Optional) Set the priority for the subject.

Step 14 Click **Update**.

Step 15 Click **OK**.

Step 16 Click **Submit**.
Logging is enabled for this contract.

Enabling ACL Contract Permit Logging Using the NX-OS CLI

The following example shows how to enable Contract permit logging using the NX-OS CLI.

Procedure

Step 1 To enable logging of packets or flows that were allowed to be sent because of Contract permit rules, use the following commands:

```
configure
tenant <tenantName>
contract <contractName> type <permit>
subject <subject Name>
access-group <access-list> <in/out/both> log
```

Example:

For example:

```
apic1# configure
apic1(config)# tenant BDMode1
```

```
apic1(config-tenant)# contract Logicmp type permit
apic1(config-tenant-contract)# subject icmp
apic1(config-tenant-contract-subj)# access-group arp both log
```

Step 2 To disable the permit logging use the **no** form of the access-group command; for example, use the `no access-group arp both log` command.

Enabling ACL Contract Permit Logging Using the REST API

The following example shows you how to enable permit and deny logging using the REST API. This example configures ACL permit and deny logging for a contract with subjects that have Permit and Deny actions configured.

Procedure

For this configuration, send a post with XML similar to the following example:

Example:

```
<vzBrCP dn="uni/tn-Tenant64/brc-C64" name="C64" scope="context">
    <vzSubj consMatchT="AtleastOne" name="HTTPSsbj" provMatchT="AtleastOne" revFltPorts="yes"
 rn="subj-HTTPSsbj">
         <vzRsSubjFiltAtt action="permit" directives="log" forceResolve="yes"
priorityOverride="default"
rn="rssubjFiltAtt-PerHTTPS" tDn="uni/tn-Tenant64/flt-PerHTTPS" tRn="flt-PerHTTPS"
tnVzFilterName="PerHTTPS"/>
    </vzSubj>
    <vzSubj consMatchT="AtleastOne" name="httpSbj" provMatchT="AtleastOne" revFltPorts="yes"
 rn="subj-httpSbj">
        <vzRsSubjFiltAtt action="deny" directives="log" forceResolve="yes"
priorityOverride="default"
rn="rssubjFiltAtt-httpFilter" tDn="uni/tn-Tenant64/flt-httpFilter" tRn="flt-httpFilter"
tnVzFilterName="httpFilter"/>
    </vzSubj>
    <vzSubj consMatchT="AtleastOne" name="subj64" provMatchT="AtleastOne" revFltPorts="yes"
 rn="subj-subj64">
        <vzRsSubjFiltAtt action="permit" directives="log" forceResolve="yes"
priorityOverride="default"
rn="rssubjFiltAtt-icmp" tDn="uni/tn-common/flt-icmp" tRn="flt-icmp" tnVzFilterName="icmp"/>

    </vzSubj>
</vzBrCP>
```

Enabling Taboo Contract Deny Logging Using the GUI

The following steps show how to enable Taboo Contract deny logging using the GUI.

Procedure

Step 1 On the menu bar, choose **Tenants** > **<tenant name>**.

Step 2 In the **Navigation** pane, expand **Contracts**.

Step 3 Right-click **Taboos** and choose **Create Taboo Contract**.

Step 4 In the Create Taboo Contract dialog box, perform the following actions to specify the Taboo contract:

a) In the **Name** field, type the name for the contract.
b) Optional. In the **Description** field, type a description of the Taboo contract.
c) Click the + icon to expand **Subjects**.

Step 5 In the **Create Taboo Contract Subject** dialog box, perform the following actions:

a) In the Specify Identity of Subject area, type a name and optional description.
b) Click the + icon to expand **Filters**.
c) From the **Name** drop-down list, choose one of the default values, such as**<tenant_name>/arp**, **<tenant_name>/default**, **<tenant_name>/est**, **<tenant_name>/icmp**, choose a previously created filter, or**Create Filter**.

Note
If you chose **Create Filter**, in the Specify Filter Identity Area, perform the following actions to specify criteria for the ACL Deny rule:

a. Type a name and optional description.

b. Expand **Entries**, type a name for the rule, and choose the criteria to define the traffic you want to deny.

c. In theDirectives drop-down list, choose **log**.

d. Click **Update**.

e. Click **OK**.

Step 6 Click **Submit**.
Logging is enabled for this Taboo contract.

Enabling Taboo Contract Deny Logging Using the NX-OS CLI

The following example shows how to enable Taboo Contract deny logging using the NX-OS CLI.

Procedure

Step 1 To enable logging of packets or flows dropped because of Taboo Contract deny rules, use the following commands:

```
configure
tenant <tenantName>
contract <contractName> type <deny>
subject <subject Name>
access-group <access-list> <both> log
```

Example:

For example:

```
apic1# configure
apic1(config)# tenant BDMode1
```

```
apic1(config-tenant)# contract dropFTP type deny
apic1(config-tenant-contract)# subject dropftp
apic1(config-tenant-contract-subj)# access-group ftp both log
```

Step 2 To disable the deny logging use the **no** form of the access-group command; for example, use the `no access-group https both log` command.

Enabling Taboo Contract Deny Logging Using the REST API

The following example shows you how to enable Taboo Contract deny logging using the REST API.

Procedure

To configure taboo contract deny logging, send a post with XML similar to the following example.

Example:

```
<vzTaboo dn="uni/tn-Tenant64/taboo-TCtrctPrefix" name="TCtrctPrefix" scope="context">
    <vzTSubj name="PrefSubj" rn="tsubj-PrefSubj"">
        <vzRsDenyRule directives="log" forceResolve="yes" rn="rsdenyRule-default"
tCl="vzFilter"
tDn="uni/tn-common/flt-default" tRn="flt-default"/>
    </vzTSubj>
</vzTaboo>
```

Viewing ACL Permit and Deny Logs Using the GUI

The following steps show how to view ACL permit and deny logs (if they are enabled) for traffic flows, using the GUI:

Procedure

Step 1 On the menu bar, choose **Tenants** > **<tenant name>**.

Step 2 In the **Navigation** pane, click on **Tenant <tenant name>**.

Step 3 In the **Tenants <tenant name> Work** pane, click the **Operational** tab.

Step 4 Under the **Operational** tab, click the **Flows** tab.
Under the **Flows** tab, click one of the tabs to view log data for Layer 2 permit logs (**L2 Permit**) Layer 3 permit logs (**L3 Permit**, Layer 2 deny logs (**L2 Drop**), or Layer 3 deny logs (**L3 Drop**). On each tab, you can view ACL logging data, if traffic is flowing. The data points differ according to the log type and ACL rule; for example, the following data points are included for **L3 Permit** and **L3 Deny** logs:

- VRF
- Alias
- Source IP address

- Destination IP address
- Protocol
- Source port
- Destination port
- Source MAC address
- Destination MAC address
- Node
- Source interface
- VRF Encap
- Source EPG
- Destination EPG
- Source PC Tag
- Destination PC Tag

Note
You can also use the **Packets** tab (next to the **Flows** tab) to access ACL logs for groups of packets (up to 10) with the same signature, source and destination. You can see what type of packets are being sent and which are being dropped.

Viewing ACL Permit and Deny Logs Using the REST API

The following example shows how to view Layer 2 deny log data for traffic flows, using the REST API. You can send queries using the following MOs:

- acllogDropL2Flow
- acllogPermitL2Flow
- acllogDropL3Flow
- acllogPermitL3Flow
- acllogDropL2Pkt
- acllogPermitL2Pkt
- acllogDropL3Pkt
- acllogPermitL3Pkt

Before you begin

You must enable permit or deny logging, before you can view ACL contract permit and deny log data.

Procedure

To view Layer 3 drop log data, send the following query using the REST API:

```
GET https://apic-ip-address/api/class/acllogDropL3Flow
```

Example:

The following example shows sample output:

```
<?xml version="1.0" encoding="UTF-8"?>
<imdata totalCount="2">
    <acllogPermitL3Flow childAction=""
dn="topology/pod-1/node-101/ndbgs/acllog/tn-common/ctx-inb

/permitl3flow-spctag-333-dpctag-444-sepgname-unknown-depgname-unknown-sip-[100:c000:a00:700:b00:0:f00:0]


-dip-[19.0.2.10]-proto-udp-sport-17459-dport-8721-smac-00:00:15:00:00:28-dmac-00:00:12:00:00:25-sintf-

        [port-channel5]-vrfencap-VXLAN: 2097153" dstEpgName="unknown" dstIp="19.0.2.10"
dstMacAddr="00:00:12:00:00:25"
       dstPcTag="444" dstPort="8721" lcOwn="local" modTs="never" monPolDn="" protocol="udp"
 srcEpgName="unknown"
        srcIntf="port-channel5" srcIp="100:c000:a00:700:b00:0:f00:0"
srcMacAddr="00:00:15:00:00:28" srcPcTag="333"
        srcPort="17459" status="" vrfEncap="VXLAN: 2097153"/>
    <acllogPermitL3Flow childAction=""
dn="topology/pod-1/node-102/ndbgs/acllog/tn-common/ctx-inb

/permitl3flow-spctag-333-dpctag-444-sepgname-unknown-depgname-unknown-sip-[100:c000:a00:700:b00:0:f00:0]-dip-


[19.0.2.10]-proto-udp-sport-17459-dport-8721-smac-00:00:15:00:00:28-dmac-00:00:12:00:00:25-sintf-

        [port-channel5]-vrfencap-VXLAN: 2097153" dstEpgName="unknown" dstIp="19.0.2.10"
dstMacAddr="00:00:12:00:00:25"
       dstPcTag="444" dstPort="8721" lcOwn="local" modTs="never" monPolDn="" protocol="udp"
 srcEpgName="unknown"
        srcIntf="port-channel5" srcIp="100:c000:a00:700:b00:0:f00:0"
srcMacAddr="00:00:15:00:00:28" srcPcTag="333"
        srcPort="17459" status="" vrfEncap="VXLAN: 2097153"/>
</imdata>
```

Viewing ACL Permit and Deny Logs Using the NX-OS CLI

The following steps show how to view ACL log details using the NX-OS-style CLI **show acllog** command.

The syntax for the Layer 3 command is **show acllog {permit | deny} l3 {pkt | flow} tenant <tenant_name> vrf <vrf_name> srcip <source_ip> dstip <destination_ip> srcport <source_port> dstport <destination_port> protocol <protocol> srcintf <source_interface> start-time <startTime> end-time <endTime> detail**

The syntax for the Layer 2 command is **show acllog {permit | deny} l2 {flow | pkt} tenant <tenant_name> vrf <VRF_name> srcintf <source_interface> vlan <VLAN_number> detail**

Note The full syntax of the **show acllog** command is only available on Generation 2 Cisco Nexus 9000 series switches (with names that end in EX or FX or later, such as N9K-C93180LC-EX) and Cisco APIC Release 3.2 or later. With Generation 1 switches (with names that do not end in EX or FX) or Cisco APIC releases before 3.2, the available syntax is as above.

In Cisco APIC 3.2 and later, additional keywords are added to both versions of the command, with the **detail** keyword:**[dstEpgName <destination_EPG_name>| dstmac <destination_MAC_address> | dstpctag <destination_PCTag> | srcEpgName <source_EPG_name> | srcmac <source_MAC_address> | srcpctag <source_PCTag>]**

Procedure

Step 1 The following example shows how to use the **show acllog drop l3 flow tenant common vrf default detail** command to display detailed information about Layer 3 deny logs for the common tenant:

Example:

```
apic1# show acllog deny l3 flow tenant common vrf default detail
SrcPcTag  : 49153
DstPcTag  : 32773
SrcEPG    : uni/tn-TSW_Tenant0/ap-tsw0AP0/epg-tsw0ctx0BD0epg6
DstEPG    : uni/tn-TSW_Tenant0/ap-tsw0AP0/epg-tsw0ctx0BD0epg5
SrcIp     : 16.0.2.10
DstIp     : 19.0.2.10
Protocol  : udp
SrcPort   : 17459
DstPort   : 8721
SrcMAC    : 00:00:15:00:00:28
DstMAC    : 00:00:12:00:00:25
Node      : 101
SrcIntf   : port-channel5
VrfEncap  : VXLAN: 2097153
```

This example shows the output on Generation 2 switches, with Cisco APIC Release 3.2 or later.

Step 2 The following example shows how to use the **show acllog deny l2 flow tenant common vrf tsw0connctx0 detail** command to display detailed information about Layer 3 deny logs for the common tenant:

Example:

```
apic1# show acllog deny l2 flow tenant common vrf tsw0connctx0 detail
SrcPcTag DstPcTag   SrcEPG           DstEPG          SrcMAC            DstMAC          Node
   SrcIntf   vlan
--------- --------- ----------------  --------------  ------------------  ------------------
   -----   ---------  ------
 32773    49153   uni/tn-TSW       uni/tn-TSW  00:00:11:00:00:11  11:00:32:00:00:33   101
    port-     2
                   _Tenant0/ap-     _Tenant0/ap-
    channel8
                   tsw0AP0/epg-     tsw0AP0/epg-
                   tsw0ctx0BD0epg5  tsw0ctx0BD0epg6
```

This example shows the output on Generation 2 switches, with Cisco APIC Release 3.2 or later.

Step 3 The following example shows how to use the **show acllog permit l3 pkt tenant <tenant name> vrf <vrf name> [detail]** command to display detailed information about the common VRF ACL Layer 3 permit packets that were sent:

```
apic1# show acllog permit l3 pkt tenant common vrf default detail acllog permit l3 packets
 detail:
srcIp      : 10.2.0.19
dstIp      : 10.2.0.16
protocol   : udp
srcPort    : 13124
dstPort    : 4386
srcIntf    : port-channel5
vrfEncap   : VXLAN: 2097153
pktLen     : 112
srcMacAddr : 00:00:15:00:00:28
dstMacAddr : 00:00:12:00:00:25
timeStamp  : 2015-03-17T21:31:14.383+00:00
```

This example shows the output on Generation 1 switches, or with Cisco APIC releases before 3.2.

Step 4 The following example shows how to use the **show acllog permit l2 pkt tenant <tenant name> vrf <vrf name> srcintf <s interface>** command to view information about default VRF Layer 2 packets sent from interface port-channel15:

```
apic1# show acllog permit l2 pkt tenant common vrf default srcintf port-channel5
acllog permit L2 Packets
      Node          srcIntf       pktLen     timeStamp
 --------------  --------------  --------  --------------
                 port-channel5      1      2015-03-17T21:
                                           31:14.383+00:00
```

This example shows the output on Generation 1 switches, or with Cisco APIC releases before 3.2.