Performing Tasks Using the NX-OS Style CLI

Part I: Layer 3 Configuration

Configuring Common Pervasive Gateway Using the NX-OS Style CLI

Configuring Common Pervasive Gateway Using the NX-OS Style CLI

Before you begin

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

Procedure

Configure Common Pervasive Gateway.

Example:

```
apic1#configure
apic1(config)#tenant demo
apic1(config-tenant)#bridge-domain test
apic1(config-tenant-bd)#l2-unknown-unicast flood
apic1(config-tenant-bd)#arp flooding
apic1(config-tenant-bd)#exit

apic1(config-tenant)#interface bridge-domain test
apic1(config-tenant-interface)#multi-site-mac-address 12:34:56:78:9a:bc
apic1(config-tenant-interface)#mac-address 00:CC:CC:CC:C1:01 (Should be unique for each ACI fabric)
apic1(config-tenant-interface)#ip address 192.168.10.1/24 multi-site
apic1(config-tenant-interface)#ip address 192.168.10.254/24 (Should be unique for each ACI fabric)
```

Configuring IP Aging Using the NX-OS Style CLI

Configuring the IP Aging Policy Using the NX-OS-Style CLI

This section explains how to enable and disable the IP Aging policy using the CLI.

Procedure

Step 1 To enable the IP aging policy:

Example:

```
ifc1(config)# endpoint ip aging
```

Step 2 To disable the IP aging policy:

Example:

```
ifav9-ifc1(config)# no endpoint ip aging
```

What to do next

To specify the interval used for tracking IP addresses on endpoints, create an Endpoint Retention policy.

Configuring a Static Route on a Bridge Domain Using the NX-OS Style CLI

Configuring a Static Route on a Bridge Domain Using the NX-OS Style CLI

To configure a static route in a pervasive bridge domain (BD), use the following NX-OS style CLI commands:

Before you begin

The tenant, VRF, BD and EPG are configured.

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

SUMMARY STEPS

1. **configure**
2. **tenant** *tenant-name*
3. **application** *ap-name*
4. **epg** *epg-name*
5. **endpoint ip***A.B.C.D/LEN* **next-hop** *A.B.C.D* [scope *scope*]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1(config)# tenant t1`	Creates a tenant or enters tenant configuration mode.
Step 3	**application** *ap-name* **Example:** `apic1(config-tenant)# application ap1`	Creates an application profile or enters application profile mode.
Step 4	**epg** *epg-name* **Example:** `apic1(config-tenant-app)# epg ep1` <> <A.B.C.D> [scope <scope>]	Creates an EPG or enters EPG configuration mode.
Step 5	**endpoint ip***A.B.C.D/LEN* **next-hop** *A.B.C.D* [scope *scope*] **Example:** `apic1(config-tenant-app-epg)# endpoint ip 125.12.1.1/32 next-hop 26.0.14.101`	Creates an endpoint behind the EPG. The subnet mask must be /32 (/128 for IPv6) pointing to one IP address or one endpoint.

Example

The following example shows the commands to configure an endpoint behind an EPG.

```
apic1# config
      apic1(config)# tenant t1
      apic1(config-tenant)# application ap1
      apic1(config-tenant-app)# epg ep1
      apic1(config-tenant-app-epg)# endpoint ip 125.12.1.1/32 next-hop 26.0.14.101
```

Configuring Dataplane IP Learning per VRF Using the NX-OS Style CLI

Configuring Dataplane IP Learning Using the NX-OS-Style CLI

This section explains how to disable dataplane IP learning using the NX-OS-style CLI.

To disable dataplane IP learning for a specific VRF:

Procedure

Step 1 Enter the configuration mode.

Example:

```
apic1# config
```

Step 2 Enter the tenant mode for the specific tenant.

Example:

```
apic1(config)# tenant name
```

Step 3 Enter the VRF context mode.

Example:

```
apic1(config-tenant)# vrf context name
```

Step 4 Disable dataplane IP learning for the VRF.

Example:

```
apic1(config-tenant-vrf)# ipdataplanelearning disabled
```

Configuring IPv6 Neighbor Discovery Using the NX-OS Style CLI

Configuring a Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery on the Bridge Domain Using the NX-OS Style CLI

Procedure

Step 1 Configure an IPv6 neighbor discovery interface policy and assign it to a bridge domain:

a) Create an IPv6 neighbor discovery interface policy:

Example:

```
apic1(config)# tenant ExampleCorp
apic1(config-tenant)# template ipv6 nd policy NDPol001
apic1(config-tenant-template-ipv6-nd)# ipv6 nd mtu 1500
```

b) Create a VRF and bridge domain:

Example:

```
apic1(config-tenant)# vrf context pvn1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member pvn1
apic1(config-tenant-bd)# exit
```

c) Assign an IPv6 neighbor discovery policy to the bridge domain:

Example:

```
apic1(config-tenant)# interface bridge-domain bd1
apic1(config-tenant-interface)# ipv6 nd policy NDPol001
apic1(config-tenant-interface)#exit
```

Step 2 Configure an IPV6 bridge domain subnet and neighbor discovery prefix policy on the subnet:

Example:

```
apic1(config-tenant)# interface bridge-domain bd1
apic1(config-tenant-interface)# ipv6 address 34::1/64
apic1(config-tenant-interface)# ipv6 address 33::1/64
apic1(config-tenant-interface)# ipv6 nd prefix 34::1/64 1000 1000
apic1(config-tenant-interface)# ipv6 nd prefix 33::1/64 4294967295 4294967295
```

Configuring an IPv6 Neighbor Discovery Interface Policy with RA on a Layer 3 Interface Using the NX-OS Style CLI

This example configures an IPv6 neighbor discovery interface policy, and assigns it to a Layer 3 interface. Next, it configures an IPv6 Layer 3 Out interface, neighbor discovery prefix policy, and associates the neighbor discovery policy to the interface.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.

Step 2	**tenant** *tenant_name*
	Example:
	`apic1(config)# tenant ExampleCorp`
	`apic1(config-tenant)#`
		Creates a tenant and enters the tenant mode.

Step 3	**template ipv6 nd policy** *policy_name* **Example:** `apic1(config-tenant)# template ipv6 nd policy NDPol001`	Creates an IPv6 ND policy.
Step 4	**ipv6 nd mtu** *mtu value* **Example:**	Assigns an MTU value to the IPv6 ND policy.

	Command or Action	Purpose

	`apic1(config-tenant-template-ipv6-nd)# ipv6 nd mtu 1500`
	`apic1(config-tenant-template-ipv6)# exit`
	`apic1(config-tenant-template)# exit`
	`apic1(config-tenant)#`	

Step 5	**vrf context** *VRF_name*
	Example:
	`apic1(config-tenant)# vrf context pvn1`
	`apic1(config-tenant-vrf)# exit`
		Creates a VRF.

Step 6	**l3out** *VRF_name* **Example:** `apic1(config-tenant)# l3out l3extOut001`	Creates a Layer 3 Out.

Step 7	**vrf member** *VRF_name*
	Example:
	`apic1(config-tenant-l3out)# vrf member pvn1`
	`apic1(config-tenant-l3out)# exit`
		Associates the VRF with the Layer 3 Out.

Step 8	**external-l3 epg instp l3out** *l3extOut001*
	Example:
	`apic1(config-tenant)# external-l3 epg instp l3out l3extOut001`
	`apic1(config-tenant-l3ext-epg)# vrf member pvn1`
	`apic1(config-tenant-l3ext-epg)# exit`
		Assigns the Layer 3 Out and the VRF to a Layer 3 interface.

Step 9	**leaf** *2011* **Example:** `apic1(config)# leaf 2011`	Enters the leaf switch mode.

Step 10	**vrf context tenant** *ExampleCorp* **vrf** *pvn1* **l3out** *l3extOut001*
	Example:
	`apic1(config-leaf)# vrf context tenant ExampleCorp vrf pvn1 l3out l3extOut001`
	`apic1(config-leaf-vrf)# exit`
		Associates the VRF to the leaf switch.

	Command or Action	Purpose

Step 11	**int** *eth 1/1*
	Example:
	`apic1(config-leaf)# int eth 1/1`
	`apic1(config-leaf-if)#`
		Enters the interface mode.

Step 12	**vrf member tenant** *ExampleCorp* **vrf** *pvn1* **l3out** *l3extOut001* **Example:** `apic1(config-leaf-if)# vrf member tenant ExampleCorp vrf pvn1 l3out l3extOut001`	Specifies the associated Tenant, VRF, Layer 3 Out in the interface.
Step 13	**ipv6 address** *2001:20:21:22::2/64* **preferred** **Example:** `apic1(config-leaf-if)# ipv6 address 2001:20:21:22::2/64 preferred`	Specifies the primary or preferred IPv6 address.
Step 14	**ipv6 nd prefix** *2001:20:21:22::2/64 1000 1000* **Example:** `apic1(config-leaf-if)# ipv6 nd prefix 2001:20:21:22::2/64 1000 1000`	Configures the IPv6 ND prefix policy under the Layer 3 interface.

Step 15	**inherit ipv6 nd** *NDPol001*
	Example:
	`apic1(config-leaf-if)# inherit ipv6 nd NDPol001`
	`apic1(config-leaf-if)# exit`
	`apic1(config-leaf)# exit`
		Configures the ND policy under the Layer 3 interface.

The configuration is complete.

Configuring Microsoft NLB Using the NX-OS Style CLI

Configuring Microsoft NLB in Unicast Mode Using the NX-OS Style CLI

This task configures Microsoft NLB to flood all of the ports in the bridge domain.

Before you begin

Have the following information available before proceeding with these procedures:

- Microsoft NLB cluster VIP
- Microsoft NLB cluster MAC address

SUMMARY STEPS

1. **configure**
2. **tenant** *tenant-name*
3. **application** *app-profile-name*
4. **epg** *epg-name*
5. **[no] endpoint {ip | ipv6}** *ip-address* **epnlb mode mode-uc mac** *mac-address*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1 (config)# tenant tenant1`	Creates a tenant if it does not exist or enters tenant configuration mode.
Step 3	**application** *app-profile-name* **Example:** `apic1 (config-tenant)# application app1`	Creates an application profile if it doesn't exist or enters application profile configuration mode.
Step 4	**epg** *epg-name* **Example:** `apic1 (config-tenant-app)# epg epg1`	Creates an EPG if it doesn't exist or enters EPG configuration mode.
Step 5	**[no] endpoint {ip | ipv6}** *ip-address* **epnlb mode mode-uc mac** *mac-address* **Example:** `apic1 (config-tenant-app-epg)# endpoint ip 192.0.2.2/32 epnlb mode mode-uc mac 03:BF:01:02:03:04`	Configures Microsoft NLB in unicast mode, where: • *ip-address* is the Microsoft NLB cluster VIP. • *mac-address* is the Microsoft NLB cluster MAC address.

Configuring Microsoft NLB in Multicast Mode Using the NX-OS Style CLI

This task configures Microsoft NLB to flood only on certain ports in the bridge domain.

Before you begin

Have the following information available before proceeding with these procedures:

- Microsoft NLB cluster VIP
- Microsoft NLB cluster MAC address

SUMMARY STEPS

1. **configure**
2. **tenant** *tenant-name*
3. **application** *app-profile-name*
4. **epg** *epg-name*
5. **[no] endpoint {ip | ipv6}** *ip-address* **epnlb mode mode-mcast--static mac** *mac-address*
6. **[no] nlb static-group** *mac-address* **leaf** *leaf-num* **interface {ethernet** *slot/port* **| port-channel** *port-channel-name***} vlan** *portEncapVlan*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1 (config)# tenant tenant1`	Creates a tenant if it does not exist or enters tenant configuration mode.
Step 3	**application** *app-profile-name* **Example:** `apic1 (config-tenant)# application app1`	Creates an application profile if it doesn't exist or enters application profile configuration mode.
Step 4	**epg** *epg-name* **Example:** `apic1 (config-tenant-app)# epg epg1`	Creates an EPG if it does not exist or enters EPG configuration mode.
Step 5	**[no] endpoint {ip | ipv6}** *ip-address* **epnlb mode mode-mcast--static mac** *mac-address* **Example:** `apic1 (config-tenant-app-epg)# endpoint ip 192.0.2.2/32 epnlb mode mode-mcast--static mac 03:BF:01:02:03:04`	Configures Microsoft NLB in static multicast mode, where: • *ip-address* is the Microsoft NLB cluster VIP. • *mac-address* is the Microsoft NLB cluster MAC address.
Step 6	**[no] nlb static-group** *mac-address* **leaf** *leaf-num* **interface {ethernet** *slot/port* **| port-channel** *port-channel-name***} vlan** *portEncapVlan* **Example:**	Adds Microsoft NLB multicast VMAC to the EPG ports where the Microsoft NLB servers are connected, where: • *mac-address* is the Microsoft NLB cluster MAC address that you entered in Step 5, on page 9.

	Command or Action	Purpose
	`apic1 (config-tenant-app-epg)# nlb static-group 03:BF:01:02:03:04 leaf 102 interface ethernet 1/12 vlan 19`	• *leaf-num* is the leaf switch that contains the interface to be added or removed. • *port-channel-name* is the name of the port-channel, when the port-channel option is used. • *portEncapVlan* is the encapsulation VLAN for the static member of the application EPG.

Configuring Microsoft NLB in IGMP Mode Using the NX-OS Style CLI

This task configures Microsoft NLB to flood only on certain ports in the bridge domain.

Before you begin

Have the following information available before proceeding with these procedures:

- Microsoft NLB cluster VIP
- Microsoft NLB cluster MAC address

SUMMARY STEPS

1. **configure**
2. **tenant** *tenant-name*
3. **application** *app-profile-name*
4. **epg** *epg-name*
5. **[no] endpoint {ip | ipv6}** *ip-address* **epnlb mode mode-mcast-igmp group** *multicast-IP-address*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1 (config)# tenant tenant1`	Creates a tenant if it does not exist or enters tenant configuration mode.
Step 3	**application** *app-profile-name* **Example:** `apic1 (config-tenant)# application app1`	Creates an application profile if it doesn't exist or enters application profile configuration mode.

	Command or Action	Purpose
Step 4	**epg** *epg-name* **Example:** `apic1 (config-tenant-app)# epg epg1`	Creates an EPG if it doesn't exist or enters EPG configuration mode.
Step 5	**[no] endpoint {ip | ipv6}** *ip-address* **epnlb mode mode-mcast-igmp group** *multicast-IP-address* **Example:** `apic1 (config-tenant-app-epg)# endpoint ip 192.0.2.2/32 epnlb mode mode-mcast-igmp group 1.3.5.7`	Configures Microsoft NLB in IGMP mode, where: • *ip-address* is the Microsoft NLB cluster VIP. • *multicast-IP-address* is the multicast IP for the NLB endpoint group.

Configuring IGMP Snooping Using the NX-OS Style CLI

Configuring and Assigning an IGMP Snooping Policy to a Bridge Domain using the NX-OS Style CLI

Before you begin

- Create the tenant that will consume the IGMP Snooping policy.
- Create the bridge domain for the tenant, where you will attach he IGMP Snooping policy.

Procedure

	Command or Action	Purpose

Step 1	Create a snooping policy based on default values.
	Example:
	`apic1(config-tenant)# template ip igmp snooping policy cookieCut1`
	`apic1(config-tenant-template-ip-igmp-snooping)# show run all`
	
	`# Command: show running -config all tenant foo template ip igmp snooping policy cookieCut1`
	`# Time: Thu Oct 13 18:26:03 2016`
	`  tenant t_10`
	`    template ip igmp snooping policy cookieCut1`
	`      ip igmp snooping`
	`      no ip igmp snooping fast-leave`
	`      ip igmp snooping last-member-query-interval 1`
	`      no ip igmp snooping querier`
	`      ip igmp snooping query-interval 125`
	`      ip igmp snooping query-max-response-time 10`
	`      ip igmp snooping stqrtup-query-count 2`
	`      ip igmp snooping startup-query-interval 31`
	`      no description`
	`    exit`
		The example NX-OS style CLI sequence:
		• Creates an IGMP Snooping policy named cookieCut1 with default values.
		• Displays the default IGMP Snooping values for the policy cookieCut1.

Performing Tasks Using the NX-OS Style CLI

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td></td><td><pre> exit
apic1(config-tenant-template-ip-igmp-snooping)#</pre></td><td></td></tr>
<tr><td>Step 2</td><td>Modify the snooping policy as necessary.
Example:
<pre>apic1(config-tenant-template-ip-igmp-snooping)# ip
 igmp snooping query-interval 300
apic1(config-tenant-template-ip-igmp-snooping)#
show run all

Command: show running -config all tenant foo
template ip igmp snooping policy cookieCut1
#Time: Thu Oct 13 18:26:03 2016
  tenant foo
    template ip igmp snooping policy cookieCut1
      ip igmp snooping
      no ip igmp snooping fast-leave
      ip igmp snooping last-member-query-interval
 1
      no ip igmp snooping querier
      ip igmp snooping query-interval 300
      ip igmp snooping query-max-response-time 10
      ip igmp snooping stqrtup-query-count 2
      ip igmp snooping startup-query-interval 31
      no description
    exit
  exit
apic1(config-tenant-template-ip-igmp-snooping)#
exit
apic1(config--tenant)#</pre></td><td>The example NX-OS style CLI sequence:
• Specifies a custom value for the query-interval value in the IGMP Snooping policy named cookieCut1.
• Confirms the modified IGMP Snooping value for the policy cookieCut1.</td></tr>
<tr><td>Step 3</td><td>Modify the snooping policy as necessary.
Example:
<pre>apic1(config-tenant-template-ip-igmp-snooping)# ip
 igmp snooping ?
 <CR>
 fast-leave   Enable IP IGMP
Snooping fast leave processing
 last-member-query-interval Change the IP IGMP
snooping last member query interval param
 querier   Enable IP IGMP
Snooping querier processing
 query-interval   Change the IP IGMP
snooping query interval param
 query-max-response-time   Change the IP IGMP
snooping max query response time
 startup-query-count   Change the IP IGMP
snooping number of initial queries to send
 startup-query-interval   Change the IP IGMP
snooping time for sending initial queries
 version   Change the IP IGMP
snooping version param

apic1(config-tenant-template-ip-igmp-snooping)# ip
 igmp snooping version ?</pre></td><td>The example NX-OS style CLI sequence:
• Specifies a custom value for the query version of the IGMP Snooping policy.
• Confirms the modified IGMP Snooping version for the policy.</td></tr>
</table>

	Command or Action	Purpose

	`v2  version-2`
	`v3  version-3`
	
	`apic1(config-tenant)# show run`
	`# Command: show running-config tenant tenant1`
	`# Time: Mon Jun  1 01:53:53 2020`
	`  tenant tenant1`
	`    <snipped>`
	`    interface bridge-domain amit_bd`
	`      ip address 10.175.31.30/24 secondary`
	`      ip address 100.175.31.1/32 secondary`
	`snooping-querier`
	`      ip igmp snooping policy igmp_snoop_policy`
	`      exit`
	`    template ip igmp snooping policy`
	`igmp_snoop_policy`
	`      ip igmp snooping fast-leave`
	`      ip igmp snooping last-member-query-interval`
	` 2`
	`      ip igmp snooping querier v3`
	`      ip igmp snooping query-interval 100`
	`      ip igmp snooping startup-query-count 5`
	`      ip igmp snooping version v3`
	`      exit`
	`    exit`	

Step 4	Assign the policy to a bridge domain. **Example:** **`apic1(config-tenant)# int bridge-domain`** `bd3` **`apic1(config-tenant-interface)# ip igmp snooping policy cookieCut1`**	The example NX-OS style CLI sequence: • Navigates to bridge domain, BD3. for the query-interval value in the IGMP Snooping policy named cookieCut1. • Assigns the IGMP Snooping policy with a modified IGMP Snooping value for the policy cookieCut1.

What to do next

You can assign the IGMP Snooping policy to multiple bridge domains.

Enabling IGMP Snooping and Multicast on Static Ports in the NX-OS Style CLI

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

The steps described in this task assume the pre-configuration of the following entities:

- Tenant: tenant_A
- Application: application_A
- EPG: epg_A
- Bridge Domain: bridge_domain_A
- vrf: vrf_A -- a member of bridge_domain_A
- VLAN Domain: vd_A (configured with a range of 300-310)

- Leaf switch: 101 and interface 1/10

  The target interface 1/10 on switch 101 is associated with VLAN 305 and statically linked with tenant_A, application_A, epg_A

- Leaf switch: 101 and interface 1/11

  The target interface 1/11 on switch 101 is associated with VLAN 309 and statically linked with tenant_A, application_A, epg_A

Before you begin

Before you begin to enable IGMP snooping and multicasting for an EPG, complete the following tasks.

- Identify the interfaces to enable this function and statically assign them to that EPG

> **Note** For details on static port assignment, see *Deploying an EPG on a Specific Port with APIC Using the NX-OS Style CLI* in the *Cisco APIC Layer 2 Networking Configuration Guide*.

- Identify the IP addresses that you want to be recipients of IGMP snooping multicast traffic.

Procedure

	Command or Action	Purpose

Step 1	On the target interfaces enable IGMP snooping and layer 2 multicasting
	
	Example:
	`apic1# conf t`
	`apic1(config)# tenant tenant_A`
	`apic1(config-tenant)# application application_A`
	`apic1(config-tenant-app)# epg epg_A`
	`apic1(config-tenant-app-epg)# ip igmp snooping static-group 225.1.1.1 leaf 101 interface ethernet 1/10 vlan 305`
	`apic1(config-tenant-app-epg)# end`
	
	`apic1# conf t`
	`apic1(config)# tenant tenant_A; application application_A; epg epg_A`
	`apic1(config-tenant-app-epg)# ip igmp snooping static-group 227.1.1.1 leaf 101 interface ethernet 1/11 vlan 309`
	`apic1(config-tenant-app-epg)# exit`
	`apic1(config-tenant-app)# exit`
		The example sequences enable:
		• IGMP snooping on the statically-linked target interface 1/10 and associates it with a multicast IP address, 225.1.1.1
		• IGMP snooping on the statically-linked target interface 1/11 and associates it with a multicast IP address, 227.1.1.1

Enabling Group Access to IGMP Snooping and Multicast using the NX-OS Style CLI

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

The steps described in this task assume the pre-configuration of the following entities:

- Tenant: tenant_A
- Application: application_A
- EPG: epg_A
- Bridge Domain: bridge_domain_A
- vrf: vrf_A -- a member of bridge_domain_A
- VLAN Domain: vd_A (configured with a range of 300-310)
- Leaf switch: 101 and interface 1/10

  The target interface 1/10 on switch 101 is associated with VLAN 305 and statically linked with tenant_A, application_A, epg_A
- Leaf switch: 101 and interface 1/11

  The target interface 1/11 on switch 101 is associated with VLAN 309 and statically linked with tenant_A, application_A, epg_A

Note For details on static port assignment, see *Deploying an EPG on a Specific Port with APIC Using the NX-OS Style CLI* in the *Cisco APIC Layer 2 Networking Configuration Guide*.

Procedure

	Command or Action	Purpose

Step 1	Define the route-map "access groups."
	Example:
	`apic1# conf t`
	`apic1(config)# tenant tenant_A; application application_A; epg epg_A`
	`apic1(config-tenant)# route-map fooBroker permit`
	`apic1(config-tenant-rtmap)# match ip multicast group 225.1.1.1/24`
	`apic1(config-tenant-rtmap)# exit`
	
	`apic1(config-tenant)# route-map fooBroker deny`
	`apic1(config-tenant-rtmap)# match ip multicast group 227.1.1.1/24`
	`apic1(config-tenant-rtmap)# exit`
		The example sequences configure:
		• Route-map-access group "foobroker" linked to multicast group 225.1.1.1/24, access permited
		• Route-map-access group "foobroker" linked to multicast group 227.1.1.1/24, access denied

Step 2	Verify route map configurations.
	Example:
	`apic1(config-tenant)# show running-config tenant test route-map fooBroker`
	`# Command: show running-config tenant test route-map fooBroker`
	`# Time: Mon Aug 29 14:34:30 2016`
	`  tenant test`
	`    route-map fooBroker permit 10`	

	Command or Action	Purpose

	` match ip multicast group 225.1.1.1/24`
	` exit`
	` route-map fooBroker deny 20`
	` match ip multicast group 227.1.1.1/24`
	` exit`
	` exit`	

Step 3	Specify the access group connection path.
	Example:
	`apic1(config-tenant)# application application_A`
	`apic1(config-tenant-app)# epg epg_A`
	`apic1(config-tenant-app-epg)# ip igmp snooping access-group route-map fooBroker leaf 101 interface ethernet 1/10 vlan 305`
	`apic1(config-tenant-app-epg)# ip igmp snooping access-group route-map newBroker leaf 101 interface ethernet 1/10 vlan 305`
		The example sequences configure:
		• Route-map-access group "foobroker" connected through leaf switch 101, interface 1/10, and VLAN 305.
		• Route-map-access group "newbroker" connected through leaf switch 101, interface 1/10, and VLAN 305.

Step 4	Verify the access group connections.
	Example:
	`apic1(config-tenant-app-epg)# show run`
	`# Command: show running-config tenant tenant_A application application_A epg epg_A`
	`# Time: Mon Aug 29 14:43:02 2016`
	` tenant tenent_A`
	` application application_A`
	` epg epg_A`
	` bridge-domain member bridge_domain_A`
	
	` ip igmp snooping access-group route-map fooBroker leaf 101 interface ethernet 1/10 vlan 305`
	` ip igmp snooping access-group route-map fooBroker leaf 101 interface ethernet 1/11 vlan 309`
	` ip igmp snooping access-group route-map newBroker leaf 101 interface ethernet 1/10 vlan 305`
	` ip igmp snooping static-group 225.1.1.1 leaf 101 interface ethernet 1/10 vlan 305`
	` ip igmp snooping static-group 225.1.1.1 leaf 101 interface ethernet 1/11 vlan 309`
	` exit`
	` exit`
	` exit`	

Configuring MLD Snooping Using the NX-OS Style CLI

Configuring and Assigning an MLD Snooping Policy to a Bridge Domain using the NX-OS Style CLI

Before you begin

- Create the tenant that will consume the MLD Snooping policy.

- Create the bridge domain for the tenant, where you will attach the MLD Snooping policy.

SUMMARY STEPS

1. **configure terminal**
2. **tenant** *tenant-name*
3. **template ipv6 mld snooping policy** *policy-name*
4. **[no] ipv6 mld snooping**
5. **[no] ipv6 mld snooping fast-leave**
6. **[no] ipv6 mld snooping querier**
7. **ipv6 mld snooping last-member-query-interval** *parameter*
8. **ipv6 mld snooping query-interval** *parameter*
9. **ipv6 mld snooping query-max-response-time** *parameter*
10. **ipv6 mld snooping startup-query-count** *parameter*
11. **ipv6 mld snooping startup-query-interval** *parameter*
12. **exit**
13. **interface bridge-domain** *bridge-domain-name*
14. **ipv6 address** *sub-bits/prefix-length* **snooping-querier**
15. **ipv6 mld snooping policy** *policy-name*
16. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`apic1# configure terminal`
	`apic1(config)#`
		Enters configuration mode.

Step 2	**tenant** *tenant-name*
	Example:
	`apic1(config)# tenant tn1`
	`apic1(config-tenant)#`
		Creates a tenant or enters tenant configuration mode.

Step 3	**template ipv6 mld snooping policy** *policy-name*
	Example:
	`apic1(config-tenant)# template ipv6 mld snooping policy mldPolicy1`
	`apic1(config-tenant-template-ip-mld-snooping)#`
		Creates an MLD snooping policy. The example NX-OS style CLI sequence creates an MLD snooping policy named mldPolicy1.

Step 4	**[no] ipv6 mld snooping** **Example:**	Enables or disables the admin state of the MLD snoop policy. The default state is disabled.

	Command or Action	Purpose
	`apic1(config-tenant-template-ip-mld-snooping)#` **`ipv6 mld snooping`** `apic1(config-tenant-template-ip-mld-snooping)#` **`no ipv6 mld snooping`**	
Step 5	**[no] ipv6 mld snooping fast-leave** **Example:** `apic1(config-tenant-template-ip-mld-snooping)#` **`ipv6 mld snooping fast-leave`** `apic1(config-tenant-template-ip-mld-snooping)#` **`no ipv6 mld snooping fast-leave`**	Enables or disables IPv6 MLD snooping fast-leave processing.
Step 6	**[no] ipv6 mld snooping querier** **Example:** `apic1(config-tenant-template-ip-mld-snooping)#` **`ipv6 mld snooping querier`** `apic1(config-tenant-template-ip-mld-snooping)#` **`no ipv6 mld snooping querier`**	Enables or disables IPv6 MLD snooping querier processing. For the enabling querier option to be effectively enabled on the assigned policy, you must also enable the querier option in the subnets assigned to the bridge domains to which the policy is applied, as described in Step 14, on page 19.
Step 7	**ipv6 mld snooping last-member-query-interval** *parameter* **Example:** `apic1(config-tenant-template-ip-mld-snooping)#` **`ipv6 mld snooping last-member-query-interval 25`**	Changes the IPv6 MLD snooping last member query interval parameter. The example NX-OS style CLI sequence changes the IPv6 MLD snooping last member query interval parameter to 25 seconds. Valid options are 1-25. The default is 1 second.
Step 8	**ipv6 mld snooping query-interval** *parameter* **Example:** `apic1(config-tenant-template-ip-mld-snooping)#` **`ipv6 mld snooping query-interval 300`**	Changes the IPv6 MLD snooping query interval parameter. The example NX-OS style CLI sequence changes the IPv6 MLD snooping query interval parameter to 300 seconds. Valid options are 1-18000. The default is 125 seconds.
Step 9	**ipv6 mld snooping query-max-response-time** *parameter* **Example:** `apic1(config-tenant-template-ip-mld-snooping)#` **`ipv6 mld snooping query-max-response-time 25`**	Changes the IPv6 MLD snooping max query response time. The example NX-OS style CLI sequence changes the IPv6 MLD snooping max query response time to 25 seconds. Valid options are 1-25. The default is 10 seconds.
Step 10	**ipv6 mld snooping startup-query-count** *parameter* **Example:** `apic1(config-tenant-template-ip-mld-snooping)#` **`ipv6 mld snooping startup-query-count 10`**	Changes the IPv6 MLD snooping number of initial queries to send. The example NX-OS style CLI sequence changes the IPv6 MLD snooping number of initial queries to send to 10. Valid options are 1-10. The default is 2.
Step 11	**ipv6 mld snooping startup-query-interval** *parameter* **Example:**	Changes the IPv6 MLD snooping time for sending initial queries. The example NX-OS style CLI sequence changes the IPv6 MLD snooping time for sending initial queries

	Command or Action	Purpose
	`apic1(config-tenant-template-ip-mld-snooping)# ipv6 mld snooping startup-query-interval 300`	to 300 seconds. Valid options are 1-18000. The default is 31 seconds.

Step 12	**exit**
	Example:
	`apic1(config-tenant-template-ip-mld-snooping)# exit`
	`apic1(config-tenant)#`
		Returns to configure mode.

Step 13	**interface bridge-domain** *bridge-domain-name*
	Example:
	`apic1(config-tenant)# interface bridge-domain bd1`
	`apic1(config-tenant-interface)#`
		Configures the interface bridge-domain. The example NX-OS style CLI sequence configures the interface bridge-domain named bd1.

Step 14	**ipv6 address** *sub-bits/prefix-length* **snooping-querier** **Example:** `apic1(config-tenant-interface)# ipv6 address 2000::5/64 snooping-querier`	Configures the bridge domain as switch-querier. This will enable the querier option in the subnet assigned to the bridge domain where the policy is applied.
Step 15	**ipv6 mld snooping policy** *policy-name* **Example:** `apic1(config-tenant-interface)# ipv6 mld snooping policy mldPolicy1`	Associates the bridge domain with an MLD snooping policy. The example NX-OS style CLI sequence associates the bridge domain with an MLD snooping policy named mldPolicy1.

Step 16	**exit**
	Example:
	`apic1(config-tenant-interface)# exit`
	`apic1(config-tenant)#`
		Returns to configure mode.

Configuring IP Multicast Using the NX-OS Style CLI

Configuring Layer 3 Multicast Using the NX-OS Style CLI

Procedure

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant, the configure mode for the VRF, and configure PIM options.

Example:

```
apic1(config)# tenant tenant1
apic1(config-tenant)# vrf context tenant1_vrf
apic1(config-tenant-vrf)# ip pim
apic1(config-tenant-vrf)# ip pim fast-convergence
apic1(config-tenant-vrf)# ip pim bsr forward
```

Step 3 Configure IGMP and the desired IGMP options for the VRF.

Example:

```
apic1(config-tenant-vrf)# ip igmp
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# interface bridge-domain tenant1_bd
apic1(config-tenant-interface)# ip multicast
apic1(config-tenant-interface)# ip igmp allow-v3-asm
apic1(config-tenant-interface)# ip igmp fast-leave
apic1(config-tenant-interface)# ip igmp inherit interface-policy igmp_intpol1
apic1(config-tenant-interface)# exit
```

Step 4 Enter the L3 Out mode for the tenant, enable PIM, and enter the leaf interface mode. Then configure PIM for this interface.

Example:

```
apic1(config-tenant)# l3out tenant1_l3out
apic1(config-tenant-l3out)# ip pim
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# exit
apic1(config)#
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/125
apic1(config-leaf-if) ip pim inherit interface-policy pim_intpol1
```

Step 5 Configure IGMP for the interface using the IGMP commands.

Example:

```
apic1(config-leaf-if)# ip igmp fast-leave
apic1(config-leaf-if)# ip igmp inherit interface-policy igmp_intpol1
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

Step 6 Configure a fabric RP.

Example:

```
apic1(config)# tenant tenant1
apic1(config-tenant)# vrf context tenant1_vrf
apic1(config-tenant-vrf)# ip pim fabric-rp-address 20.1.15.1 route-map intervrf-ctx2
apic1(config-tenant-vrf)# ip pim fabric-rp-address 20.1.15.2 route-map intervrf-ctx1
apic1(config-tenant-vrf)# exit
```

Step 7 Configure a inter-VRF multicast.

Example:

```
apic1(config-tenant)# vrf context tenant1_vrf
apic1(config-tenant-vrf)# ip pim inter-vrf-src ctx2 route-map intervrf-ctx2
apic1(config-tenant-vrf)# route-map intervrf-ctx2 permit 1
```

```
apic1(config-tenant-vrf)# match ip multicast group 226.20.0.0/24
apic1(config-tenant-vrf)#  exit
apic1(config-tenant)# exit
apic1(config)#
```

This completes the APIC Layer 3 multicast configuration.

Configuring Layer 3 IPv6 Multicast Using the NX-OS Style CLI

Before you begin

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

Procedure

Step 1 Enable PIM6 on the VRF and configure the Rendezvous Point (RP).

Example:

```
apic1(config)# tenant tenant1
apic1(config-tenant)# vrf context tenant1_vrf
apic1(config-tenant-vrf)# ipv6 pim
apic1(config-tenant-vrf)# ipv6 rp-address 2018::100:100:100:100 route-map ipv6_pim_routemap
```

Step 2 Configure a PIM6 interface policy and apply it on the Layer 3 Out.

Example:

```
apic1(config-tenant)# l3out tenant1_l3out
apic1(config-tenant-l3out)# ipv6 pim
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# exit
apic1(config)#
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/125
apic1(config-leaf-if) ipv6 pim inherit interface-policy pim6_intpol1
```

Step 3 Enable PIM6 on the BD.

Example:

```
apic1(config-tenant)# interface bridge-domain tenant1_bd
apic1(config-tenant-interface)# ipv6 multicast
apic1(config-tenant)# exit
apic1(config)#
```

Layer 3 IPv6 multicast with PIM6 is enabled.

Configuring Multicast Filtering Using the NX-OS Style CLI

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

Before you begin

- The bridge domain where you will be configuring multicast filtering is already created.
- The bridge domain is a PIM-enabled bridge domain.
- Layer 3 multicast is enabled at the VRF level.

Procedure

Step 1 Enter the configuration mode.

```
apic1# configure
apic1(config)#
```

Step 2 Access the tenant and enable PIM.

```
apic1(config)# tenant tenant-name
apic1(config-tenant)# vrf context VRF-name
apic1(config-tenant-vrf)# ip pim
apic1(config-tenant-vrf)# exit
apic1(config-tenant)#
```

Example:

```
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# ip pim
apic1(config-tenant-vrf)# exit
apic1(config-tenant)#
```

Step 3 Access the bridge domain where you want to configure multicast filtering.

```
apic1(config-tenant)# bridge-domain BD-name
apic1(config-tenant-bd)#
```

Example:

```
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)#
```

Step 4 Determine whether you want to enable multicast *source* or *receiver* filtering on this bridge domain.

Note
You can also enable both source and receiver filtering on the same bridge domain.

- If you want to enable multicast source filtering on this bridge domain, enter the following:

  ```
  apic1(config-tenant-bd)# src-filter source-route-map-policy
  ```

  For example:

  ```
  apic1(config-tenant-bd)# src-filter routemap-Mcast-src
  ```

- If you want to enable multicast receiver filtering on this bridge domain, enter the following:

  ```
  apic1(config-tenant-bd)# dst-filter destination-route-map-policy
  ```

  For example:

  ```
  apic1(config-tenant-bd)# dst-filter routemap-Mcast-dst
  ```

Step 5 Enable multicasting for IPv4:

```
apic1(config-tenant-bd)# mcast-allow
apic1(config-tenant-bd)#
```

Step 6 Associate the bridge domain with the VRF.

```
apic1(config-tenant-bd)# vrf member VRF-name
apic1(config-tenant-bd)# exit
apic1(config-tenant)#
```

Example:

```
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)#
```

Step 7 Enable multicast on the bridge domain.

```
apic1(config-tenant)# interface bridge-domain BD-name
apic1(config-tenant-interface)# ip multicast
apic1(config-tenant-interface)# exit
apic1(config-tenant)#
```

Example:

```
apic1(config-tenant)# interface bridge-domain bd1
apic1(config-tenant-interface)# ip multicast
apic1(config-tenant-interface)# exit
apic1(config-tenant)#
```

Step 8 Configure the route map.

```
apic1(config-tenant)# route-map destination-route-map-policy <permit/deny> sequence_number
apic1(config-tenant-rtmap)# match ip multicast <source/group> IP_address_subnet <source/group>
IP_address_subnet
apic1(config-tenant-rtmap)# exit
apic1(config-tenant)# exit
apic1(config)#
```

Example:

```
apic1(config-tenant)# route-map routemap-Mcast-src permit 1
apic1(config-tenant-rtmap)# match ip multicast source 10.10.1.1/24 group 192.1.1.1/32
apic1(config-tenant-rtmap)# exit
apic1(config-tenant)# route-map routemap-Mcast-dst permit 1
apic1(config-tenant-rtmap)# match ip multicast group 192.2.2.2/32
apic1(config-tenant-rtmap)# exit
apic1(config-tenant)# exit
apic1(config)#
```

Configuring Multi-Pod Using the NX-OS Style CLI

Setting Up Multi-Pod Fabric Using the NX-OS CLI

Before you begin

- The node group and L3Out policies have already been created.

Procedure

Step 1 Set up the multi-pod, as in the following example:

Example:

```
ifav4-ifc1#  show run system
# Command: show running-config system
# Time: Mon Aug  1 21:32:03 2016
  system cluster-size 3
  system switch-id FOX2016G9DW 204 ifav4-spine4 pod 2
  system switch-id SAL1748H56D 201 ifav4-spine1 pod 1
  system switch-id SAL1803L25H 102 ifav4-leaf2 pod 1
  system switch-id SAL1819RXP4 101 ifav4-leaf1 pod 1
  system switch-id SAL1931LA3B 203 ifav4-spine2 pod 2
  system switch-id SAL1934MNY0 103 ifav4-leaf3 pod 1
  system switch-id SAL1934MNY3 104 ifav4-leaf4 pod 1
  system switch-id SAL1938P7A6 202 ifav4-spine3 pod 1
  system switch-id SAL1938PHBB 105 ifav4-leaf5 pod 2
  system switch-id SAL1942R857 106 ifav4-leaf6 pod 2
  system pod 1 tep-pool 10.0.0.0/16
  system pod 2 tep-pool 10.1.0.0/16
ifav4-ifc1#
```

Step 2 Configure a VLAN domain, as in the following example:

Example:

```
ifav4-ifc1# show running-config vlan-domain l3Dom
# Command: show running-config vlan-domain l3Dom
# Time: Mon Aug  1 21:32:31 2016
  vlan-domain l3Dom
    vlan 4
    exit
ifav4-ifc1#
```

Step 3 Configure the fabric external connectivity, as in the following example:

Example:

```
ifav4-ifc1# show running-config fabric-external
# Command: show running-config fabric-external
# Time: Mon Aug  1 21:34:17 2016
  fabric-external 1
    bgp evpn peering
    pod 1
      interpod data hardware-proxy 100.11.1.1/32
      bgp evpn peering
      exit
    pod 2
      interpod data hardware-proxy 200.11.1.1/32
      bgp evpn peering
      exit
    route-map interpod-import
      ip prefix-list default permit 0.0.0.0/0
      exit
    route-target extended 5:16
    exit
ifav4-ifc1#
```

Step 4 Configure the spine switch interface and OSPF configuration as in the following example:

Example:

```
# Command: show running-config spine
# Time: Mon Aug  1 21:34:41 2016
  spine 201
    vrf context tenant infra vrf overlay-1
      router-id 201.201.201.201
      exit
    interface ethernet 1/1
      vlan-domain member l3Dom
      exit
    interface ethernet 1/1.4
      vrf member tenant infra vrf overlay-1
      ip address 201.1.1.1/30
      ip router ospf default area 1.1.1.1
      ip ospf cost 1
      exit
    interface ethernet 1/2
      vlan-domain member l3Dom
      exit
    interface ethernet 1/2.4
      vrf member tenant infra vrf overlay-1
      ip address 201.2.1.1/30
      ip router ospf default area 1.1.1.1
      ip ospf cost 1
      exit
    router ospf default
      vrf member tenant infra vrf overlay-1
        area 1.1.1.1 loopback 201.201.201.201
```

```
      area 1.1.1.1 interpod peering
      exit
    exit
  exit
spine 202
  vrf context tenant infra vrf overlay-1
    router-id 202.202.202.202
    exit
  interface ethernet 1/2
    vlan-domain member l3Dom
    exit
  interface ethernet 1/2.4
        vrf member tenant infra vrf overlay-1
    ip address 202.1.1.1/30
    ip router ospf default area 1.1.1.1
    exit
  router ospf default
    vrf member tenant infra vrf overlay-1
      area 1.1.1.1 loopback 202.202.202.202
      area 1.1.1.1 interpod peering
      exit
    exit
  exit
spine 203
  vrf context tenant infra vrf overlay-1
    router-id 203.203.203.203
    exit
  interface ethernet 1/1
    vlan-domain member l3Dom
    exit
  interface ethernet 1/1.4
    vrf member tenant infra vrf overlay-1
    ip address 203.1.1.1/30
    ip router ospf default area 0.0.0.0
    ip ospf cost 1
    exit
  interface ethernet 1/2
    vlan-domain member l3Dom
    exit
  interface ethernet 1/2.4
    vrf member tenant infra vrf overlay-1
    ip address 203.2.1.1/30
    ip router ospf default area 0.0.0.0
    ip ospf cost 1
    exit
  router ospf default
    vrf member tenant infra vrf overlay-1
      area 0.0.0.0 loopback 203.203.203.203
      area 0.0.0.0 interpod peering
      exit
    exit
  exit
spine 204
  vrf context tenant infra vrf overlay-1
    router-id 204.204.204.204
    exit
  interface ethernet 1/31
    vlan-domain member l3Dom
    exit
  interface ethernet 1/31.4
    vrf member tenant infra vrf overlay-1
    ip address 204.1.1.1/30
    ip router ospf default area 0.0.0.0
    ip ospf cost 1
```

```
      exit
    router ospf default
      vrf member tenant infra vrf overlay-1
        area 0.0.0.0 loopback 204.204.204.204
        area 0.0.0.0 interpod peering
        exit
      exit
    exit
ifav4-ifc1#
```

Configuring Remote Leaf Switches Using the NX-OS Style CLI

Configure Remote Leaf Switches Using the NX-OS Style CLI

This example configures a spine switch and a remote leaf switch to enable the leaf switch to communicate with the main fabric pod.

Before you begin

- The IPN router and remote leaf switches are active and configured; see WAN Router and Remote Leaf Switch Configuration Guidelines.
- The remote leaf switches are running a switch image of 13.1.x or later (aci-n9000-dk9.13.1.x.x.bin).
- The pod in which you plan to add the remote leaf switches is created and configured.

Procedure

Step 1 Define the TEP pool for a remote location 5, in pod 2.

The network mask must be /24 or lower.

Use the following new command: **system remote-leaf-site** *site-id* **pod** *pod-id* **tep-pool** *ip-address-and-netmask*

Example:

```
apic1(config)# system remote-leaf-site 5 pod 2 tep-pool 192.0.0.0/16
```

Step 2 Add a remote leaf switch to pod 2, remote-leaf-site 5.

Use the following command: **system switch-id** *serial-number node-id leaf-switch-name***pod** *pod-id* **remote-leaf-site** *remote-leaf-site-id* **node-type remote-leaf-wan**

Example:

```
apic1(config)# system switch-id FDO210805SKD 109 ifav4-leaf9 pod 2
remote-leaf-site 5 node-type remote-leaf-wan
```

Step 3 Configure a VLAN domain with a VLAN that includes VLAN 4.

Example:

```
apic1(config)# vlan-domain ospfDom
apic1(config-vlan)# vlan 4-5
apic1(config-vlan)# exit
```

Step 4 Configure two L3Outs for the infra tenant, one for the remote leaf connections and one for the multipod IPN.

Example:

```
apic1(config)# tenant infra
apic1(config-tenant)# l3out rl-wan
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# l3out ipn-multipodInternal
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# exit
apic1(config)#
```

Step 5 Configure the spine switch interfaces and sub-interfaces to be used by the L3Outs.

Example:

```
apic1(config)# spine 201
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-vrf)# exit
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-vrf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36
apic1(config-spine-if)# vlan-domain member ospfDom
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf default
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.4
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-if)# ip router ospf default area 5
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf multipod-internal
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out ipn-multipodInternal
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.5
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-if)# ip router ospf multipod-internal area 5
apic1(config-spine-if)# exit
apic1(config-spine)# exit
apic1(config)#
```

Step 6 Configure the remote leaf switch interface and sub-interface used for communicating with the main fabric pod.

Example:

```
(config)# leaf 101
apic1(config-leaf)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-vrf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49
apic1(config-leaf-if)# vlan-domain member ospfDom
apic1(config-leaf-if)# exit
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-leaf-ospf-vrf)# area 5 l3out rl-wan-test
```

```
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49.4
apic1(config-leaf-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-if)# ip router ospf default area 5
apic1(config-leaf-if)# exit
```

Example

The following example provides a downloadable configuration:

```
apic1# configure
apic1(config)# system remote-leaf-site 5 pod 2 tep-pool 192.0.0.0/16
apic1(config)# system switch-id FDO210805SKD 109 ifav4-leaf9 pod 2
remote-leaf-site 5 node-type remote-leaf-wan
apic1(config)# vlan-domain ospfDom
apic1(config-vlan)# vlan 4-5
apic1(config-vlan)# exit
apic1(config)# tenant infra
apic1(config-tenant)# l3out rl-wan-test
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# l3out ipn-multipodInternal
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# exit
apic1(config)#
apic1(config)# spine 201
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-vrf)# exit
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-vrf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36
apic1(config-spine-if)# vlan-domain member ospfDom
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf default
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.4
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-if)# ip router ospf default area 5
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf multipod-internal
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out ipn-multipodInternal
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.5
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-if)# ip router ospf multipod-internal area 5
apic1(config-spine-if)# exit
apic1(config-spine)# exit
apic1(config)#
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
```

```
apic1(config-leaf-vrf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49
apic1(config-leaf-if)# vlan-domain member ospfDom
apic1(config-leaf-if)# exit
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-leaf-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49.4
apic1(config-leaf-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-if)# ip router ospf default area 5
apic1(config-leaf-if)# exit
```

Part II: External Routing (L3Out) Configuration

Routed Connectivity to External Networks

Configuring an MP-BGP Route Reflector Using the NX-OS Style CLI

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
```

Node and Interface for L3Out

Configuring Layer 3 Routed and Sub-Interface Port Channels Using the NX-OS Style CLI

Configuring a Layer 3 Routed Port-Channel Using the NX-OS CLI

This procedure configures a Layer 3 routed port channel.

SUMMARY STEPS

1. **configure**
2. **leaf** *node-id*

3. **interface port-channel** *channel-name*
4. **no switchport**
5. **vrf member** *vrf-name* **tenant** *tenant-name*
6. **vlan-domain member** *vlan-domain-name*
7. **ip address** *ip-address/subnet-mask*
8. **ipv6 address** *sub-bits/prefix-length* **preferred**
9. **ipv6 link-local** *ipv6-link-local-address*
10. **mac-address** *mac-address*
11. **mtu** *mtu-value*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	**leaf** *node-id* **Example:** `apic1(config)# leaf 101`	Specifies the leaf switch or leaf switches to be configured. The *node-id* can be a single node ID or a range of IDs, in the form *node-id1-node-id2*, to which the configuration will be applied.
Step 3	**interface port-channel** *channel-name* **Example:** `apic1(config-leaf)# interface port-channel po1`	Enters the interface configuration mode for the specified port channel.
Step 4	**no switchport** **Example:** `apic1(config-leaf-if)# no switchport`	Makes the interface Layer 3 capable.
Step 5	**vrf member** *vrf-name* **tenant** *tenant-name* **Example:** `apic1(config-leaf-if)# vrf member v1 tenant t1`	Associates this port channel to this virtual routing and forwarding (VRF) instance and L3 outside policy, where: • *vrf-name* is the VRF name. The name can be any case-sensitive, alphanumeric string up to 32 characters. • *tenant-name* is the tenant name. The name can be any case-sensitive, alphanumeric string up to 32 characters.
Step 6	**vlan-domain member** *vlan-domain-name* **Example:** `apic1(config-leaf-if)# vlan-domain member dom1`	Associates the port channel template with the previously configured VLAN domain.

	Command or Action	Purpose
Step 7	**ip address** *ip-address*/*subnet-mask* **Example:** `apic1(config-leaf-if)# ip address 10.1.1.1/24`	Sets the IP address and subnet mask for the specified interface.
Step 8	**ipv6 address** *sub-bits*/*prefix-length* **preferred** **Example:** `apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface, where: • *sub-bits* is the subprefix bits and host bits of the address to be concatenated with the prefixes provided by the general prefix specified with the prefix-name argument. The sub-bits argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. • *prefix-length* is the length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
Step 9	**ipv6 link-local** *ipv6-link-local-address* **Example:** `apic1(config-leaf-if)# ipv6 link-local fe80::1`	Configures an IPv6 link-local address for an interface.
Step 10	**mac-address** *mac-address* **Example:** `apic1(config-leaf-if)# mac-address 00:44:55:66:55::01`	Manually sets the interface MAC address.
Step 11	**mtu** *mtu-value* **Example:** `apic1(config-leaf-if)# mtu 1500`	Sets the MTU for this class of service.

Example

This example shows how to configure a basic Layer 3 port channel.

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member v1 tenant t1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
apic1(config-leaf-if)# ipv6 link-local fe80::1
apic1(config-leaf-if)# mac-address 00:44:55:66:55::01
apic1(config-leaf-if)# mtu 1500
```

Configuring a Layer 3 Sub-Interface Port-Channel Using the NX-OS CLI

This procedure configures a Layer 3 sub-interface port channel.

SUMMARY STEPS

1. **configure**
2. **leaf** *node-id*
3. **vrf member** *vrf-name* **tenant** *tenant-name*
4. **vlan-domain member** *vlan-domain-name*
5. **ip address** *ip-address* / *subnet-mask*
6. **ipv6 address** *sub-bits* / *prefix-length* **preferred**
7. **ipv6 link-local** *ipv6-link-local-address*
8. **mac-address** *mac-address*
9. **mtu** *mtu-value*
10. exit
11. **interface port-channel** *channel-name*
12. **vlan-domain member** *vlan-domain-name*
13. exit
14. **interface port-channel** *channel-name.number*
15. **vrf member** *vrf-name* **tenant** *tenant-name*
16. exit

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	**leaf** *node-id* **Example:** `apic1(config)# leaf 101`	Specifies the leaf switch or leaf switches to be configured. The *node-id* can be a single node ID or a range of IDs, in the form *node-id1-node-id2*, to which the configuration will be applied.
Step 3	**vrf member** *vrf-name* **tenant** *tenant-name* **Example:** `apic1(config-leaf-if)# vrf member v1 tenant t1`	Associates this port channel to this virtual routing and forwarding (VRF) instance and L3 outside policy, where:, where: • *vrf-name* is the VRF name. The name can be any case-sensitive, alphanumeric string up to 32 characters.

	Command or Action	Purpose
		• *tenant-name* is the tenant name. The name can be any case-sensitive, alphanumeric string up to 32 characters.
Step 4	**vlan-domain member** *vlan-domain-name* **Example:** `apic1(config-leaf-if)# vlan-domain member dom1`	Associates the port channel template with the previously configured VLAN domain.
Step 5	**ip address** *ip-address* / *subnet-mask* **Example:** `apic1(config-leaf-if)# ip address 10.1.1.1/24`	Sets the IP address and subnet mask for the specified interface.
Step 6	**ipv6 address** *sub-bits* / *prefix-length* **preferred** **Example:** `apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface, where: • *sub-bits* is the subprefix bits and host bits of the address to be concatenated with the prefixes provided by the general prefix specified with the prefix-name argument. The sub-bits argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. • *prefix-length* is the length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
Step 7	**ipv6 link-local** *ipv6-link-local-address* **Example:** `apic1(config-leaf-if)# ipv6 link-local fe80::1`	Configures an IPv6 link-local address for an interface.
Step 8	**mac-address** *mac-address* **Example:** `apic1(config-leaf-if)# mac-address 00:44:55:66:55::01`	Manually sets the interface MAC address.
Step 9	**mtu** *mtu-value* **Example:** `apic1(config-leaf-if)# mtu 1500`	Sets the MTU for this class of service.
Step 10	exit **Example:** `apic1(config-leaf-if)# exit`	Returns to configure mode.

	Command or Action	Purpose
Step 11	**interface port-channel** *channel-name* **Example:** `apic1(config-leaf)# interface port-channel po1`	Enters the interface configuration mode for the specified port channel.
Step 12	**vlan-domain member** *vlan-domain-name* **Example:** `apic1(config-leaf-if)# vlan-domain member dom1`	Associates the port channel template with the previously configured VLAN domain.
Step 13	exit **Example:** `apic1(config-leaf-if)# exit`	Returns to configure mode.
Step 14	**interface port-channel** *channel-name.number* **Example:** `apic1(config-leaf)# interface port-channel po1.2001`	Enters the interface configuration mode for the specified sub-interface port channel.
Step 15	**vrf member** *vrf-name* **tenant** *tenant-name* **Example:** `apic1(config-leaf-if)# vrf member v1 tenant t1`	Associates this port channel to this virtual routing and forwarding (VRF) instance and L3 outside policy, where:, where: • *vrf-name* is the VRF name. The name can be any case-sensitive, alphanumeric string up to 32 characters. • *tenant-name* is the tenant name. The name can be any case-sensitive, alphanumeric string up to 32 characters.
Step 16	exit **Example:** `apic1(config-leaf-if)# exit`	Returns to configure mode.

Example

This example shows how to configure a basic Layer 3 sub-interface port-channel.

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 2001
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member v1 tenant t1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
apic1(config-leaf-if)# ipv6 link-local fe80::1
apic1(config-leaf-if)# mac-address 00:44:55:66:55::01
apic1(config-leaf-if)# mtu 1500
```

```
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface port-channel po1.2001
apic1(config-leaf-if)# vrf member v1 tenant t1
apic1(config-leaf-if)# exit
```

Adding Ports to the Layer 3 Port-Channel Using the NX-OS CLI

This procedure adds ports to a Layer 3 port channel that you configured previously.

SUMMARY STEPS

1. **configure**
2. **leaf** *node-id*
3. **interface Ethernet** *slot/port*
4. **channel-group** *channel-name*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	**leaf** *node-id* **Example:** `apic1(config)# leaf 101`	Specifies the leaf switch or leaf switches to be configured. The *node-id* can be a single node ID or a range of IDs, in the form *node-id1-node-id2*, to which the configuration will be applied.
Step 3	**interface Ethernet** *slot/port* **Example:** `apic1(config-leaf)# interface Ethernet 1/1-2`	Enters interface configuration mode for the interface you want to configure.
Step 4	**channel-group** *channel-name* **Example:** `apic1(config-leaf-if)# channel-group p01`	Configures the port in a channel group.

Example

This example shows how to add ports to a Layer 3 port-channel.

```
apic1# configure
apic1(config)# leaf 101
```

```
apic1(config-leaf)# interface Ethernet 1/1-2
apic1(config-leaf-if)# channel-group p01
```

Configuring a Switch Virtual Interface Using the NX-OS Style CLI

Configuring SVI Interface Encapsulation Scope Using NX-OS Style CLI

The following example displaying steps for an SVI interface encapsulation scope setting is through a named Layer 3 Out configuration.

SUMMARY STEPS

1. Enter the configure mode.
2. Enter the switch mode.
3. Create the VLAN interface.
4. Specify the encapsulation scope.
5. Exit the interface mode.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	Enter the configure mode. **Example:** `apic1# configure`	Enters the configuration mode.
Step 2	Enter the switch mode. **Example:** `apic1(config)# leaf 104`	Enters the switch mode.
Step 3	Create the VLAN interface. **Example:** `apic1(config-leaf)# interface vlan 2001`	Creates the VLAN interface. The VLAN range is 1-4094.
Step 4	Specify the encapsulation scope. **Example:** `apic1(config-leaf-if)# encap scope vrf context`	Specifies the encapsulation scope.
Step 5	Exit the interface mode. **Example:** `apic1(config-leaf-if)# exit`	Exits the interface mode.

Configuring SVI Auto State Using NX-OS Style CLI

Before you begin

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

SUMMARY STEPS

1. Enter the configure mode.
2. Enter the switch mode.
3. Create the VLAN interface.
4. Enable SVI auto state.
5. Exit the interface mode.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	Enter the configure mode. **Example:** `apic1# configure`	Enters the configuration mode.
Step 2	Enter the switch mode. **Example:** `apic1(config)# leaf 104`	Enters the switch mode.
Step 3	Create the VLAN interface. **Example:** `apic1(config-leaf)# interface vlan 2001`	Creates the VLAN interface. The VLAN range is 1-4094.
Step 4	Enable SVI auto state. **Example:** `apic1(config-leaf-if)# autostate`	Enables SVI auto state. By default, the SVI auto state value is not enabled.
Step 5	Exit the interface mode. **Example:** `apic1(config-leaf-if)# exit`	Exits the interface mode.

Configuring Routing Protocols Using the NX-OS Style CLI

Configuring BGP External Routed Networks with BFD Support Using the NX-OS Style CLI

Configuring BGP External Routed Network Using the NX-OS Style CLI

Procedure

The following shows how to configure the BGP external routed network using the NX-OS CLI:

Example:

```
apic1(config-leaf)# template route-profile damp_rp tenant t1
This template will be available on all leaves where tenant t1 has a VRF deployment
apic1(config-leaf-template-route-profile)# set dampening 15 750 2000 60
apic1(config-leaf-template-route-profile)# exit
apic1(config-leaf)#
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant t1 vrf ctx3
apic1(config-leaf-bgp-vrf)# neighbor 32.0.1.0/24 l3out l3out-bgp
apic1(config-leaf-bgp-vrf-neighbor)# update-source ethernet 1/16.401
apic1(config-leaf-bgp-vrf-neighbor)# address-family ipv4 unicast
apic1(config-leaf-bgp-vrf-neighbor-af)# weight 400
apic1(config-leaf-bgp-vrf-neighbor-af)# exit
apic1(config-leaf-bgp-vrf-neighbor)# remote-as 65001
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control remove-exclusive
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control remove-exclusive-all
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control remove-exclusive-all-replace-as
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# address-family ipv4 unicast
apic1(config-leaf-bgp-vrf-af)# inherit bgp dampening damp_rp
This template will be inherited on all leaves where VRF ctx3 has been deployed
apic1(config-leaf-bgp-vrf-af)# exit
apic1(config-leaf-bgp-vrf)# address-family ipv6 unicast
apic1(config-leaf-bgp-vrf-af)# inherit bgp dampening damp_rp
This template will be inherited on all leaves where VRF ctx3 has been deployed
apic1(config-leaf-bgp-vrf-af)# exit
```

Configuring BGP Max Path Using the NX-OS Style CLI

Before you begin:

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

The appropriate tenant and the BGP external routed network are created and available.

Use the following commands when logged in to BGP:

- Commands for configuring multipath for eBGP paths:

```
maximum-paths <value>
```

```
no maximum-paths <value>
```

- Commands for configuring multipath for iBGP paths:

  `maximum-paths ibgp <value>`

  `no maximum-paths ibgp <value>`

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# template bgp address-family newAf tenant t1
This template will be available on all nodes where tenant t1 has a VRF deployment
apic1(config-bgp-af)# maximum-paths ?
<1-64>  Number of parallel paths
ibgp  Configure multipath for IBGP paths
apic1(config-bgp-af)# maximum-paths 10
apic1(config-bgp-af)# maximum-paths ibgp 8
apic1(config-bgp-af)# end
apic1#
```

Configuring AS Path Prepend Using the NX-OS Style CLI

This section provides information on how to configure the AS Path Prepend feature using the NX-OS style command line interface (CLI).

Before you begin

A configured tenant.

SUMMARY STEPS

1. To modify the autonomous system path (AS Path) for Border Gateway Protocol (BGP) routes, you can use the `set as-path` command. The set as-path command takes the form of `apic1(config-leaf-vrf-template-route-profile)# set as-path {'prepend as-num [ ,... as-num ] | prepend-last-as num}`

DETAILED STEPS

Procedure

To modify the autonomous system path (AS Path) for Border Gateway Protocol (BGP) routes, you can use the `set as-path` command. The set as-path command takes the form of `apic1(config-leaf-vrf-template-route-profile)# set as-path {'prepend as-num [ ,... as-num ] | prepend-last-as num}`

Example:

```
apic1(config)# leaf 103

apic1(config-leaf)# vrf context tenant t1 vrf v1

apic1(config-leaf-vrf)# template route-profile rp1

apic1(config-leaf-vrf-template-route-profile)# set as-path ?

prepend Prepend to the AS-Path

prepend-last-as Prepend last AS to the as-path

apic1(config-leaf-vrf-template-route-profile)# set as-path prepend 100, 101, 102, 103

apic1(config-leaf-vrf-template-route-profile)# set as-path prepend-last-as 8
```

```
apic1(config-leaf-vrf-template-route-profile)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
```

What to do next

To disable AS Path prepend, use the no form of the shown command:

```
apic1(config-leaf-vrf-template-route-profile)# [no] set
as-path { prepend as-num [ ,... as-num ] | prepend-last-as num}
```

Configuring BGP Neighbor Shutdown Using the NX-OS Style CLI

Configuring BGP Neighbor Shutdown Using the NX-OS Style CLI

The following procedure describes how to use the BGP neighbor shutdown feature using the NX-OS CLI.

Procedure

Step 1 Configure the node and interface for the L3Out.

This example configures VRF `v1` on node 103 (the border leaf switch), which is named `nodep1`, with router ID `11.11.11.103`. It also configures interface `eth1/3` as a routed interface (Layer 3 port), with IP address `12.12.12.3/24` and Layer 3 domain `dom1`.

Example:

```
apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 11.11.11.103
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 12.12.12.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

Step 2 Configure the BGP routing protocol.

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

Example:

```
apic1(config)# leaf 103
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
```

Step 3 Use the BGP neighbor shutdown feature.

Example:

```
apic1(config-leaf-bgp-vrf-neighbor)# shutdown
```

```
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit
```

Configuring a Per VRF Per Node BGP Timer Policy Using the NX-OS Style CLI

SUMMARY STEPS

1. Configure BGP ASN and the route reflector before creating a timer policy.
2. Create a timer policy.
3. Display the configured BGP policy.
4. Refer to a specific policy at a node.
5. Display the node specific BGP timer policy.

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	Configure BGP ASN and the route reflector before creating a timer policy.
	Example:
	`apic1(config)#`
	`apic1(config)# bgp-fabric`
	`apic1(config-bgp-fabric)# route-reflector spine 102`
	`apic1(config-bgp-fabric)# asn 42`
	`apic1(config-bgp-fabric)# exit`
	`apic1(config)# exit`
	`apic1#`	

Step 2	Create a timer policy.
	Example:
	`apic1# config`
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# template bgp timers pol7 tenant tn1`
	`This template will be available on all nodes where tenant tn1 has a VRF deployment`
	`apic1(config-bgp-timers)# timers bgp 120 240`
	`apic1(config-bgp-timers)# graceful-restart stalepath-time 500`
	`apic1(config-bgp-timers)# maxas-limit 300`
	`apic1(config-bgp-timers)# exit`
	`apic1(config-leaf)# exit`
	`apic1(config)# exit`
	`apic1#`
		The specific values are provided as examples only.

Step 3	Display the configured BGP policy. **Example:**	

	Command or Action	Purpose

	`apic1# show run leaf 101 template bgp timers pol7`
	
	`# Command: show running-config leaf 101 template bgp timers pol7`
	`  leaf 101`
	`    template bgp timers pol7 tenant tn1`
	`      timers bgp 120 240`
	`      graceful-restart stalepath-time 500`
	`      maxas-limit 300`
	`      exit`
	`    exit`	

Step 4	Refer to a specific policy at a node.
	Example:
	`apic1# config`
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# router bgp 42`
	`apic1(config-leaf-bgp)# vrf member tenant tn1 vrf ctx1`
	`apic1(config-leaf-bgp-vrf)# inherit node-only bgp timer pol7`
	`apic1(config-leaf-bgp-vrf)# exit`
	`apic1(config-leaf-bgp)# exit`
	`apic1(config-leaf)# exit`
	`apic1(config)# exit`
	`apic1#`	

Step 5	Display the node specific BGP timer policy.
	Example:
	`apic1# show run leaf 101 router bgp 42 vrf member tenant tn1 vrf ctx1`
	`# Command: show running-config leaf 101 router bgp 42 vrf member tenant tn1 vrf ctx1`
	`  leaf 101`
	`    router bgp 42`
	`      vrf member tenant tn1 vrf ctx1`
	`        inherit node-only bgp timer pol7`
	`        exit`
	`      exit`
	`    exit`
	`apic1#`	

Configuring Bidirectional Forwarding Detection on a Secondary IP Address Using the NX-OS-Style CLI

This procedure configures bidirectional forwarding detection (BFD) on a secondary IP address using the NX-OS-style CLI. This example configures VRF v1 on node 103 (the border leaf switch), with router ID 1.1.24.24. It also configures interface eth1/3 as a routed interface (Layer 3 port), with IP address 12.12.12.3/24 as primary and 6.11.1.224/24 as secondary address in Layer 3 domain dom1. BFD is enabled on 99.99.99.14/32, which is reachable using the secondary subnet 6.11.1.0/24.

Procedure

Step 1 Enter the configure mode.

Example:

```
apic1# configure terminal
```

Step 2 Enter the configure mode for leaf switch 103.

Example:

```
apic1(config)# leaf 103
```

Step 3 Enter the configure mode for a VRF instance.

Example:

```
apic1(config-leaf)# vrf context tenant t1 vrf v1
```

Step 4 Configure a secondary IP address.

Example:

```
apic1(config-leaf-vrf)# router-id 1.1.24.24
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 12.12.12.3/24
apic1(config-leaf-if)# ip address 6.11.1.224/24 secondary
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

Step 5 Enable BFD.

Example:

```
apic1(config-leaf)# vrf context tenant t1 vrf v1 l3out Routed
apic1(config-leaf-vrf)#router-id 1.1.24.24
apic1(config-leaf-vrf)#ip route 95.95.95.95/32 12.12.12.4 bfd
apic1(config-leaf-vrf)#ip route 99.99.99.14/32 6.11.1.100 bfd
```

Configuring BFD Globally on Leaf Switch Using the NX-OS Style CLI

Procedure

Step 1 To configure the BFD IPV4 global configuration (bfdIpv4InstPol) using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# template bfd ip bfd_ipv4_global_policy
apic1(config-bfd)# [no] echo-address 1.2.3.4
apic1(config-bfd)# [no] slow-timer 2500
apic1(config-bfd)# [no] min-tx 100
```

```
apic1(config-bfd)# [no] min-rx 70
apic1(config-bfd)# [no] multiplier 3
apic1(config-bfd)# [no] echo-rx-interval 500
apic1(config-bfd)# exit
```

Step 2 To configure the BFD IPV6 global configuration (bfdIpv6InstPol) using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# template bfd ipv6 bfd_ipv6_global_policy
apic1(config-bfd)# [no] echo-address 34::1/64
apic1(config-bfd)# [no] slow-timer 2500
apic1(config-bfd)# [no] min-tx 100
apic1(config-bfd)# [no] min-rx 70
apic1(config-bfd)# [no] multiplier 3
apic1(config-bfd)# [no] echo-rx-interval 500
apic1(config-bfd)# exit
```

Step 3 To configure access leaf policy group (infraAccNodePGrp) and inherit the previously created BFD global policies using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# template leaf-policy-group test_leaf_policy_group
apic1(config-leaf-policy-group)# [no] inherit bfd ip bfd_ipv4_global_policy
apic1(config-leaf-policy-group)# [no] inherit bfd ipv6 bfd_ipv6_global_policy
apic1(config-leaf-policy-group)# exit
```

Step 4 To associate the previously created leaf policy group onto a leaf using the NX-OS CLI:

Example:

```
apic1(config)# leaf-profile test_leaf_profile
apic1(config-leaf-profile)# leaf-group test_leaf_group
apic1(config-leaf-group)# leaf-policy-group test_leaf_policy_group
apic1(config-leaf-group)# leaf 101-102
apic1(config-leaf-group)# exit
```

Configuring BFD Globally on Spine Switch Using the NX-OS Style CLI

Use this procedure to configure BFD globally on spine switch using the NX-OS style CLI.

Procedure

Step 1 To configure the BFD IPV4 global configuration (bfdIpv4InstPol) using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# template bfd ip bfd_ipv4_global_policy
apic1(config-bfd)# [no] echo-address 1.2.3.4
apic1(config-bfd)# [no] slow-timer 2500
apic1(config-bfd)# [no] min-tx 100
apic1(config-bfd)# [no] min-rx 70
apic1(config-bfd)# [no] multiplier 3
```

```
apic1(config-bfd)# [no] echo-rx-interval 500
apic1(config-bfd)# exit
```

Step 2 To configure the BFD IPV6 global configuration (bfdIpv6InstPol) using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# template bfd ipv6 bfd_ipv6_global_policy
apic1(config-bfd)# [no] echo-address 34::1/64
apic1(config-bfd)# [no] slow-timer 2500
apic1(config-bfd)# [no] min-tx 100
apic1(config-bfd)# [no] min-rx 70
apic1(config-bfd)# [no] multiplier 3
apic1(config-bfd)# [no] echo-rx-interval 500
apic1(config-bfd)# exit
```

Step 3 To configure spine policy group and inherit the previously created BFD global policies using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# template spine-policy-group test_spine_policy_group
apic1(config-spine-policy-group)# [no] inherit bfd ip bfd_ipv4_global_policy
apic1(config-spine-policy-group)# [no] inherit bfd ipv6 bfd_ipv6_global_policy
apic1(config-spine-policy-group)# exit
```

Step 4 To associate the previously created spine policy group onto a spine switch using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# spine-profile test_spine_profile
apic1(config-spine-profile)# spine-group test_spine_group
apic1(config-spine-group)# spine-policy-group test_spine_policy_group
apic1(config-spine-group)# spine 103-104
apic1(config-leaf-group)# exit
```

Configuring BFD Interface Override Using the NX-OS Style CLI

Procedure

Step 1 To configure BFD Interface Policy (bfdIfPol) using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# tenant t0
apic1(config-tenant)# vrf context v0
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t0 vrf v0
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface Ethernet 1/18
apic1(config-leaf-if)# vrf  member tenant t0 vrf v0
apic1(config-leaf-if)# exit
```

```
apic1(config-leaf)# template bfd bfdIfPol1 tenant t0
apic1(config-template-bfd-pol)# [no] echo-mode enable
apic1(config-template-bfd-pol)# [no] echo-rx-interval 500
apic1(config-template-bfd-pol)# [no] min-rx 70
apic1(config-template-bfd-pol)# [no] min-tx 100
apic1(config-template-bfd-pol)# [no] multiplier 5
apic1(config-template-bfd-pol)# [no] optimize subinterface
apic1(config-template-bfd-pol)# exit
```

Step 2 To inherit the previously created BFD interface policy onto a L3 interface with IPv4 address using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface Ethernet 1/15
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdPol1
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password
```

Step 3 To inherit the previously created BFD interface policy onto an L3 interface with IPv6 address using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface Ethernet 1/15
apic1(config-leaf-if)# ipv6 address 2001::10:1/64 preferred
apic1(config-leaf-if)# bfd ipv6 tenant mode
apic1(config-leaf-if)# bfd ipv6 inherit interface-policy bfdPol1
apic1(config-leaf-if)# bfd ipv6 authentication keyed-sha1 key 10 key password
```

Step 4 To configure BFD on a VLAN interface with IPv4 address using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 15
apic1(config-leaf-if)# vrf member tenant t0 vrf v0
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdPol1
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password
```

Step 5 To configure BFD on a VLAN interface with IPv6 address using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 15
apic1(config-leaf-if)# ipv6 address 2001::10:1/64 preferred
apic1(config-leaf-if)# vrf member tenant t0 vrf v0
apic1(config-leaf-if)# bfd ipv6 tenant mode
apic1(config-leaf-if)# bfd ipv6 inherit interface-policy bfdPol1
apic1(config-leaf-if)# bfd ipv6 authentication keyed-sha1 key 10 key password
```

Configuring BFD Consumer Protocols Using the NX-OS Style CLI

Procedure

Step 1 To enable BFD on the BGP consumer protocol using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 200
apic1(config-bgp-fabric)# exit
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 200
apic1(config-bgp)# vrf member tenant t0 vrf v0
apic1(config-leaf-bgp-vrf)# neighbor 1.2.3.4
apic1(config-leaf-bgp-vrf-neighbor)# [no] bfd enable
```

Step 2 To enable BFD on the EIGRP consumer protocol using the NX-OS CLI:

Example:

```
apic1(config-leaf-if)# [no] ip bfd eigrp enable
```

Step 3 To enable BFD on the OSPF consumer protocol using the NX-OS CLI:

Example:

```
apic1(config-leaf-if)# [no] ip ospf bfd enable
```

```
apic1# configure
apic1(config)# spine 103
apic1(config-spine)# interface ethernet 5/3.4
apic1(config-spine-if)# [no] ip ospf bfd enable
```

Step 4 To enable BFD on the Static Route consumer protocol using the NX-OS CLI:

Example:

```
apic1(config-leaf-vrf)# [no] ip route 10.0.0.1/16 10.0.0.5 bfd
```

```
apic1(config)# spine 103
apic1(config-spine)# vrf context tenant infra vrf overlay-1
apic1(config-spine-vrf)# [no] ip route 21.1.1.1/32 32.1.1.1 bfd
```

Step 5 To enable BFD on IS-IS consumer protocol using the NX-OS CLI:

Example:

```
apic1(config)# leaf 101
apic1(config-spine)# interface ethernet 1/49
apic1(config-spine-if)# isis bfd enabled
apic1(config-spine-if)# exit
apic1(config-spine)# exit
```

```
apic1(config)# spine 103
apic1(config-spine)# interface ethernet 5/2
```

```
apic1(config-spine-if)# isis bfd enabled
apic1(config-spine-if)# exit
apic1(config-spine)# exit
```

Configuring OSPF External Routed Networks Using the NX-OS Style CLI

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
```

Configuring EIGRP External Routed Networks Using the NX-OS Style CLI

Configuring EIGRP Using the NX-OS-Style CLI

Procedure

Step 1 SSH to an Application Policy Infrastructure Controller (APIC) in the fabric:

Example:

```
# ssh admin@node_name
```

Step 2 Enter the configure mode:

Example:

```
apic1# configure
```

Step 3 Enter the configure mode for a tenant:

Example:

```
apic1(config)# tenant tenant1
```

Step 4 Configure the Layer 3 Outside on the tenant:

Example:

```
apic1(config-tenant)# show run
# Command: show running-config tenant tenant1
# Time: Tue Feb 16 09:44:09 2016
  tenant tenant1
    vrf context l3out
      exit
    l3out l3out-L1
      vrf member l3out
      exit
    l3out l3out-L3
      vrf member l3out
      exit
    external-l3 epg tenant1 l3out l3out-L3
      vrf member l3out
      match ip 0.0.0.0/0
      match ip 3.100.0.0/16
      match ipv6 43:101::/48
      contract consumer default
      exit
    external-l3 epg tenant1 l3out l3out-L1
      vrf member l3out
      match ipv6 23:101::/48
      match ipv6 13:101::/48
```

```
      contract provider default
      exit
    exit
```

Step 5 Configure a VRF for EIGRP on a leaf:

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant tenant1 vrf l3out l3out l3out-L1
apic1(config-leaf-vrf)# show run
# Command: show running-config leaf 101 vrf context tenant tenant1 vrf l3out l3out l3out-L1
# Time: Tue Feb 16 09:44:45 2016
  leaf 101
    vrf context tenant tenant1 vrf l3out l3out l3out-L1
      router-id 3.1.1.1
      route-map l3out-L1_in
        scope global
        ip prefix-list tenant1 permit 1:102::/48
        match prefix-list tenant1
          exit
        exit
      route-map l3out-L1_out
        scope global
        ip prefix-list tenant1 permit 3.102.10.0/23
        ip prefix-list tenant1 permit 3.102.100.0/31
        ip prefix-list tenant1 permit 3.102.20.0/24
        ip prefix-list tenant1 permit 3.102.30.0/25
        ip prefix-list tenant1 permit 3.102.40.0/26
        ip prefix-list tenant1 permit 3.102.50.0/27
        ip prefix-list tenant1 permit 3.102.60.0/28
        ip prefix-list tenant1 permit 3.102.70.0/29
        ip prefix-list tenant1 permit 3.102.80.0/30
        ip prefix-list tenant1 permit 3.102.90.0/32
        <OUTPUT TRUNCATED>
        ip prefix-list tenant1 permit ::/0
        match prefix-list tenant1
          exit
        exit
      route-map l3out-L1_shared
        scope global
        exit
      exit
    exit
```

Step 6 Configure the EIGRP interface policy:

Example:

```
apic1(config-leaf)# template eigrp interface-policy tenant1 tenant tenant1
This template will be available on all leaves where tenant tenant1 has a VRF deployment
apic1(config-template-eigrp-if-pol)# show run
# Command: show running-config leaf 101 template eigrp interface-policy tenant1 tenant tenant1
# Time: Tue Feb 16 09:45:50 2016
  leaf 101
    template eigrp interface-policy tenant1 tenant tenant1
      ip hello-interval eigrp default 10
      ip hold-interval eigrp default 30
      ip throughput-delay eigrp default 20 tens-of-micro
      ip bandwidth eigrp default 20
      exit
    exit
```

Step 7 Configure the EIGRP VRF policy:

Example:

```
apic1(config-leaf)# template eigrp vrf-policy tenant1 tenant tenant1
This template will be available on all leaves where tenant tenant1 has a VRF deployment
apic1(config-template-eigrp-vrf-pol)# show run
# Command: show running-config leaf 101 template eigrp vrf-policy tenant1 tenant tenant1
# Time: Tue Feb 16 09:46:31 2016
  leaf 101
    template eigrp vrf-policy tenant1 tenant tenant1
      metric version 64bit
      exit
    exit
```

Step 8 Configure the EIGRP VLAN interface and enable EIGRP in the interface:

Example:

```
apic1(config-leaf)# interface vlan 1013
apic1(config-leaf-if)# show run
# Command: show running-config leaf 101 interface vlan 1013
# Time: Tue Feb 16 09:46:59 2016
  leaf 101
    interface vlan 1013
      vrf member tenant tenant1 vrf l3out
      ip address 101.13.1.2/24
      ip router eigrp default
      ipv6 address 101:13::1:2/112 preferred
      ipv6 router eigrp default
      ipv6 link-local fe80::101:13:1:2
      inherit eigrp ip interface-policy tenant1
      inherit eigrp ipv6 interface-policy tenant1
      exit
    exit
apic1(config-leaf-if)# ip summary-address ?
 eigrp  Configure route summarization for EIGRP
apic1(config-leaf-if)# ip summary-address eigrp default 11.11.0.0/16 ?
 <CR>
apic1(config-leaf-if)# ip summary-address eigrp default 11.11.0.0/16
apic1(config-leaf-if)# ip summary-address eigrp default 11:11:1::/48
apic1(config-leaf-if)# show run
# Command: show running-config leaf 101 interface vlan 1013
# Time: Tue Feb 16 09:47:34 2016
  leaf 101
    interface vlan 1013
      vrf member tenant tenant1 vrf l3out
      ip address 101.13.1.2/24
      ip router eigrp default
      ip summary-address eigrp default 11.11.0.0/16
      ip summary-address eigrp default 11:11:1::/48
      ipv6 address 101:13::1:2/112 preferred
      ipv6 router eigrp default
      ipv6 link-local fe80::101:13:1:2
      inherit eigrp ip interface-policy tenant1
      inherit eigrp ipv6 interface-policy tenant1
      exit
    exit
```

Step 9 Apply the VLAN on the physical interface:

Example:

```
apic1(config-leaf)# interface ethernet 1/5
apic1(config-leaf-if)# show run
# Command: show running-config leaf 101 interface ethernet 1 / 5
# Time: Tue Feb 16 09:48:05 2016
```

```
  leaf 101
    interface ethernet 1/5
      vlan-domain member cli
      switchport trunk allowed vlan 1213 tenant tenant13 external-svi l3out l3out-L1
      switchport trunk allowed vlan 1613 tenant tenant17 external-svi l3out l3out-L1
      switchport trunk allowed vlan 1013 tenant tenant1 external-svi l3out l3out-L1
      switchport trunk allowed vlan 666 tenant ten_v6_cli external-svi l3out l3out_cli_L1
      switchport trunk allowed vlan 1513 tenant tenant16 external-svi l3out l3out-L1
      switchport trunk allowed vlan 1313 tenant tenant14 external-svi l3out l3out-L1
      switchport trunk allowed vlan 1413 tenant tenant15 external-svi l3out l3out-L1
      switchport trunk allowed vlan 1113 tenant tenant12 external-svi l3out l3out-L1
      switchport trunk allowed vlan 712 tenant mgmt external-svi l3out inband_l1
      switchport trunk allowed vlan 1913 tenant tenant10 external-svi l3out l3out-L1
      switchport trunk allowed vlan 300 tenant tenant1 external-svi l3out l3out-L1
      exit
    exit
```

Step 10 Enable router EIGRP:

Example:

```
apic1(config-eigrp-vrf)# show run
# Command: show running-config leaf 101 router eigrp default vrf member tenant tenant1 vrf l3out
# Time: Tue Feb 16 09:49:05 2016
  leaf 101
    router eigrp default
      exit
    router eigrp default
      exit
    router eigrp default
      exit
    router eigrp default
      vrf member tenant tenant1 vrf l3out
        autonomous-system 1001 l3out l3out-L1
        address-family ipv6 unicast
          inherit eigrp vrf-policy tenant1
          exit
        address-family ipv4 unicast
          inherit eigrp vrf-policy tenant1
          exit
        exit
      exit
```

Configuring Route Summarization Using the NX-OS Style CLI

Configuring Route Summarization for BGP, OSPF, and EIGRP Using the NX-OS Style CLI

Procedure

Step 1 Configure BGP route summarization using the NX-OS CLI as follows:

a) Enable BGP as follows:

Example:

```
apic1(config)# pod 1
```

```
apic1(config-pod)# bgp fabric
apic1(config-pod-bgp)# asn 10
apic1(config-pod)# exit
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 10
```

b) Configure the summary route as follows:

Example:

```
apic1(config-bgp)# vrf member tenant common vrf vrf1
apic1(config-leaf-bgp-vrf)# aggregate-address 10.0.0.0/8
```

Step 2 Configure OSPF external summarization using the NX-OS CLI as follows:

Example:

```
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant common vrf vrf1
apic1(config-leaf-ospf-vrf)# summary-address 10.0.0.0/8
```

Step 3 Configure OSPF inter-area summarization using the NX-OS CLI as follows:

```
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant common vrf vrf1
apic1(config-leaf-ospf-vrf)# area 0.0.0.2 range 10.0.0.0/8 cost 20
```

Step 4 Configure EIGRP summarization using the NX-OS CLI as follows:

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/31 (Or interface vlan <vlan-id>)
apic1(config-leaf-if)# ip summary-address eigrp default 10.0.0.0/8
```

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

Configuring Route Control with Route Maps and Route Profile Using NX-OS Style CLI

Configuring Route Control Per BGP Peer Using the NX-OS Style CLI

The following procedure describes how to configure the route control per BGP peer feature using the NX-OS CLI.

Procedure

Step 1 Create a route group template and add IP prefix to the route group.

This example creates a route group `match-rule1` for tenant `t1`, and adds the IP prefix of `200.3.2.0/24` to the route group.

Example:

```
apic1(config)# leaf 103
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 200.3.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)#
```

Step 2 Enter a tenant VRF mode for the node.

This example enters a tenant VRF mode for VRF `v1` for tenant `t1`.

Example:

```
apic1(config-leaf)# vrf context tenant t1 vrf v1
```

Step 3 Create a route-map and enter the route-map configuration mode, then match a route group that has already been created and enter the match mode to configure the route-profile.

This example creates a route-map `rp1`, and matches route group `match-rule1` with an order number `0`.

Example:

```
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
```

Step 4 Configure the BGP routing protocol.

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

Example:

```
apic1(config)# leaf 103
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
```

Step 5 Configure the route control per BGP peer feature.

Where:

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

Example:

```
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit
```

Configuring Route Map/Profile with Explicit Prefix List Using NX-OS Style CLI

Before you begin

- Tenant and VRF must be configured through the NX-OS CLI.
- The VRF must be enabled on the leaf switch through the NX-OS CLI.

SUMMARY STEPS

1. **configure**
2. **leaf** *node-id*
3. **template route group** *group-name* **tenant** *tenant-name*
4. **ip prefix permit** *prefix/masklen* [**le**{**32** | **128** }]
5. **community-list** [**standard** | **expanded**] *community-list-name expression*
6. **exit**
7. **vrf context tenant** *tenant-name* **vrf** *vrf-name* [**l3out** {**BGP** | **EIGRP** | **OSPF** | **STATIC** }]
8. **template route-profile** *profile-name* [*route-control-context-name order-value*]
9. **set** *attribute value*
10. **exit**
11. **route-map** *map-name*
12. **match route group** *group-name* [**order** *number*] [**deny**]
13. **inherit route-profile** *profile-name*
14. **exit**
15. **exit**
16. **exit**
17. **router bgp** *fabric-asn*
18. **vrf member tenant** *t1* **vrf** *v1*
19. **neighbor** *IP-address-of-neighbor*
20. **route-map** *map-name* {**in** | **out** }

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**leaf** *node-id* **Example:** `apic1(config)# leaf 101`	Specifies the leaf to be configured.
Step 3	**template route group** *group-name* **tenant** *tenant-name* **Example:**	Creates a route group template. **Note**

	Command or Action	Purpose
	`apic1(config-leaf)# template route group g1 tenant exampleCorp`	The route group (match rule) can have one or more of the IP prefixes and one or more match community terms. Across the match types, the AND filter is supported, so all conditions in the route group must match for the route match rule to be accepted. When there are multiple IP prefixes in route group, the OR filter is supported. Any one match prefix is accepted as a route type if it matches.
Step 4	**ip prefix permit** *prefix/masklen* [**le**{**32** | **128** }] **Example:** `apic1(config-route-group)# ip prefix permit 15.15.15.0/24`	Add IP prefix to the route group. **Note** The IP prefix can denote a BD subnet or an external network. Use optional argument le 32 for IPv4 and le 128 for IPv6 if you desire an aggregate prefix.
Step 5	**community-list** [**standard** | **expanded**] *community-list-name expression* **Example:** `apic1(config-route-group)# community-list standard com1 65535:20`	This is an optional command. Add match criteria for community if community also needs to be matched along with IP prefix.

Step 6	**exit**
	Example:
	`apic1(config-route-group)# exit`
	`apic1(config-leaf)#`
		Exit template mode.

Step 7	**vrf context tenant** *tenant-name* **vrf** *vrf-name* [**l3out** {**BGP** | **EIGRP** | **OSPF** | **STATIC** }] **Example:** `apic1(config-leaf)# vrf context tenant exampleCorp vrf v1`	Enters a tenant VRF mode for the node. **Note** If you enter the optional **l3out** string, the L3Out must be an L3Out that you configured through the NX-OS CLI.
Step 8	**template route-profile** *profile-name* [*route-control-context-name order-value*] **Example:** `apic1(config-leaf-vrf)# template route-profile rp1 ctxl 1`	Creates a template containing set actions that should be applied to the matched routes.
Step 9	**set** *attribute value* **Example:** `apic1(config-leaf-vrf-template-route-profile)# set metric 128`	Add desired attributes (set actions) to the template.

Step 10	**exit**
	Example:
	`apic1(config-leaf-vrf-template-route-profile)# exit`
	`apic1(config-leaf-vrf)#`
		Exit template mode.

	Command or Action	Purpose
Step 11	**route-map** *map-name* **Example:** `apic1(config-leaf-vrf)# route-map bgpMap`	Create a route-map and enter the route-map configuration mode.
Step 12	**match route group** *group-name* [**order** *number*] [**deny**] **Example:** `apic1(config-leaf-vrf-route-map)# match route group g1 order 1`	Match a route group that has already been created, and enter the match mode to configure the route- profile. Additionally choose the keyword **Deny** if routes matching the match criteria defined in route group needs to be denied. The default is **Permit**.
Step 13	**inherit route-profile** *profile-name* **Example:** `apic1(config-leaf-vrf-route-map-match)# inherit route-profile rp1`	Inherit a route-profile (set actions). **Note** These actions will be applied to the matched routes. Alternatively, the set actions can be configured inline instead of inheriting a route-profile.

Step 14	**exit**
	Example:
	`apic1(config-leaf-vrf-route-map-match)# exit`
	`apic1(config-leaf-vrf-route-map)#`
		Exit match mode.

Step 15	**exit**
	Example:
	`apic1(config-leaf-vrf-route-map)# exit`
	`apic1(config-leaf-vrf)#`
		Exit route map configuration mode.

Step 16	**exit**
	Example:
	`apic1(config-leaf-vrf)# exit`
	`apic1(config-leaf)#`
		Exit VRF configuration mode.

Step 17	**router bgp** *fabric-asn* **Example:** `apic1(config-leaf)# router bgp 100`	Configure the leaf node.
Step 18	**vrf member tenant** *t1* **vrf** *v1* **Example:** `apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1`	Set the BGP's VRF membership and the tenant for the BGP policy.
Step 19	**neighbor** *IP-address-of-neighbor* **Example:** `apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2`	Configure a BGP neighbor.
Step 20	**route-map** *map-name* {**in** | **out** } **Example:**	Configure the route map for a BGP neighbor.

	Command or Action	Purpose
	apic1(config-leaf-bgp-vrf-neighbor)# **route-map bgpMap out**	

Configuring a Route Control Protocol to Use Import and Export Controls, With the NX-OS Style CLI

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

This section describes how to create a route map using the NX-OS CLI:

Before you begin

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

Procedure

Step 1 Import Route control using match community, match prefix-list

Example:

```
apic1# configure
apic1(config)# leaf 101
    # Create community-list
apic1(config-leaf)# template community-list standard CL_1 65536:20 tenant exampleCorp
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1

     #Create Route-map and use it for BGP import control.
apic1(config-leaf-vrf)# route-map bgpMap
    # Match prefix-list and set route-profile actions for the match.
apic1(config-leaf-vrf-route-map)# ip prefix-list list1 permit 13.13.13.0/24
apic1(config-leaf-vrf-route-map)# ip prefix-list list1 permit 14.14.14.0/24
apic1(config-leaf-vrf-route-map)# match prefix-list list1
apic1(config-leaf-vrf-route-map-match)# set tag 200
apic1(config-leaf-vrf-route-map-match)# set local-preference 64
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant exampleCorp vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 3.3.3.3
apic1(config-leaf-bgp-vrf-neighbor)# route-map bgpMap in
```

Step 2 Export Route Control using match BD, default-export route-profile

Example:

```
# Create custom and "default-export" route-profiles
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf)# template route-profile default-export
apic1(config-leaf-vrf-template-route-profile)# set metric 256
apic1(config-leaf-vrf)# template route-profile bd-rtctrl
apic1(config-leaf-vrf-template-route-profile)# set metric 128

#Create a Route-map and match on BD, prefix-list
```

```
apic1(config-leaf-vrf)# route-map bgpMap
apic1(config-leaf-vrf-route-map)# match bridge-domain bd1
apic1(config-leaf-vrf-route-map-match)#exit
apic1(config-leaf-vrf-route-map)# match prefix-list p1
apic1(config-leaf-vrf-route-map-match)#exit
apic1(config-leaf-vrf-route-map)# match bridge-domain bd2
apic1(config-leaf-vrf-route-map-match)# inherit route-profile bd-rtctrl
```

Note

In this case, public-subnets from bd1 and prefixes matching prefix-list p1 are exported out using route-profile “default-export”, while public-subnets from bd2 are exported out using route-profile “bd-rtctrl”.

Configuring Interleak Redistribution Using the NX-OS-Style CLI

The following procedure describes how to configure the interleak redistribution using the NX-OS-style CLI.

Before you begin

Create the tenant, VRF, and L3Out.

Procedure

Step 1 Configure the route map for interleak redistribution for the border leaf node.

Example:

The following example configures the route map `CLI_RP` with an IP prefix-list `CLI_PFX1` for tenant `CLI_TEST` and VRF `VRF1`:

```
apic1# conf t
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant CLI_TEST vrf VRF1
apic1(config-leaf-vrf)# route-map CLI_RP
apic1(config-leaf-vrf-route-map)# ip prefix-list CLI_PFX1 permit 192.168.1.0/24
apic1(config-leaf-vrf-route-map)# match prefix-list CLI_PFX1 [deny]
```

Step 2 Configure the interleak redistribution using the configured route-map.

Example:

The following example configures the redistribution of OSPF routes with the configured route map `CLI_RP`:

```
apic1# conf t
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 65001
apic1(config-leaf-bgp)# vrf member tenant CLI_TEST vrf VRF1
apic1(config-leaf-bgp-vrf)# redistribute ospf route-map CLI_RP
```

Configuring Transit Routing Using the NX-OS Style CLI

Configure Transit Routing Using the NX-OS Style CLI

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

Before you begin

- Configure the node, port, functional profile, AEP, and Layer 3 domain.
- Configure a VLAN domain using the **vlan-domain** *domain* and **vlan** *vlan-range* commands.
- Configure a BGP route reflector policy to propagate the routed within the fabric.

Procedure

Step 1 Configure the tenant and VRF.

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

Example:

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
```

Step 2 Configure the nodes and interfaces.

This example configures two L3Outs for the tenant t1, on two border leaf switches:

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

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 11.11.11.103
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 12.12.12.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 22.22.22.203
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
```

```
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 23.23.23.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

Step 3 Configure the routing protocol for both leaf switches.

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

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 25.25.25.2
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit
```

Step 4 Configure a connectivity routing protocol.

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

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 40.40.40.1
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 60.60.60.1
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit
```

Step 5 Configure the external EPGs.

This example configures the network `192.168.1.0/24` as external network `extnw1` on node 101 and the network `192.168.2.0/24` as external network `extnw2` on node 102.

Example:

```
apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 192.168.1.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# external-l3 epg extnw2
apic1(config-tenant-l3ext-epg)# vrf member v1
```

```
apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw1
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw2
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
```

Step 6 Optional. Configure the route maps.

This example configures a route map for each BGP peer in the inbound and outbound directions.

Example:

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp2 out
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 102
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
```

```
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 25.25.25.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp2 in
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 out
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit
```

Step 7 Create filters (access lists) and contracts to enable the EPGs to communicate.

Example:

```
apic1(config)# tenant t1
apic1(config-tenant)# access-list http-filter
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# exit
apic1(config-tenant)# contract httpCtrct
apic1(config-tenant-contract)# scope vrf
apic1(config-tenant-contract)# subject subj1
apic1(config-tenant-contract-subj)# access-group http-filter both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
apic1(config-tenant)# exit
```

Step 8 Configure contracts and associate them with EPGs.

Example:

```
apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# contract provider httpCtrct
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# external-l3 epg extnw2
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# contract consumer httpCtrct
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(config)#
```

Example: Transit Routing

This example provides a merged configuration for transit routing. The configuration is for a single tenant and VRF, with two L3Outs, on two border leaf switches, that are each connected to separate routers.

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit

apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 11.11.11.103
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
```

```
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 12.12.12.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 40.40.40.1
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 22.22.22.203
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 23.23.23.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 25.25.25.2/24
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 60.60.60.3
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit

apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 192.168.1.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# external-l3 epg extnw2
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit

apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw1
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw2
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 101
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
```

```
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp2 out
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 102
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 25.25.25.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp2 in
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 out
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

apic1(config)# tenant t1
apic1(config-tenant)# access-list http-filter
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# exit
apic1(config-tenant)# contract httpCtrct
apic1(config-tenant-contract)# scope vrf
apic1(config-tenant-contract)# subject http-subj
apic1(config-tenant-contract-subj)# access-group http-filter both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
apic1(config-tenant)# exit
```

```
apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# contract provider httpCtrct
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# external-l3 epg extnw2
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# contract consumer httpCtrct
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(config)#
```

Configuring Shared Services Using the NX-OS Style CLI

Configuring Shared Layer 3 Out Inter-VRF Leaking Using the NX-OS Style CLI - Named Example

SUMMARY STEPS

1. Enter the configure mode.
2. Configure the provider Layer 3 Out.
3. Configure the consumer Layer 3 Out.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	Enter the configure mode. **Example:** `apic1# configure`	

Step 2	Configure the provider Layer 3 Out.
	Example:
	`apic1(config)# tenant t1_provider`
	`apic1(config-tenant)# external-l3 epg l3extInstP-1 l3out T0-o1-L3OUT-1`
	`apic1(config-tenant-l3ext-epg)# vrf member VRF1`
	`apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24 shared`
	`apic1(config-tenant-l3ext-epg)# contract provider vzBrCP-1`
	`apic1(config-tenant-l3ext-epg)# exit`
	`apic1(config-tenant)# exit`
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# vrf context tenant t1_provider vrf VRF1 l3out T0-o1-L3OUT-1`
	`apic1(config-leaf-vrf)# route-map T0-o1-L3OUT-1_shared`
	`apic1(config-leaf-vrf-route-map)# ip prefix-list l3extInstP-1 permit 192.168.2.0/24`
	`apic1(config-leaf-vrf-route-map)# match prefix-list l3extInstP-1`
	`apic1(config-leaf-vrf-route-map-match)# exit`	

	Command or Action	Purpose

	`apic1(config-leaf-vrf-route-map)# exit`
	`apic1(config-leaf-vrf)# exit`
	`apic1(config-leaf)# exit`	

Step 3	Configure the consumer Layer 3 Out.
	Example:
	`apic1(config)# tenant t1_consumer`
	`apic1(config-tenant)# external-l3 epg l3extInstP-2 l3out T0-o1-L3OUT-1`
	`apic1(config-tenant-l3ext-epg)# vrf member VRF2`
	`apic1(config-tenant-l3ext-epg)# match ip 199.16.2.0/24 shared`
	`apic1(config-tenant-l3ext-epg)# contract consumer vzBrCP-1 imported`
	`apic1(config-tenant-l3ext-epg)# exit`
	`apic1(config-tenant)# exit`
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# vrf context tenant t1_consumer vrf VRF2 l3out T0-o1-L3OUT-1`
	`apic1(config-leaf-vrf)# route-map T0-o1-L3OUT-1_shared`
	`apic1(config-leaf-vrf-route-map)# ip prefix-list l3extInstP-2 permit 199.16.2.0/24`
	`apic1(config-leaf-vrf-route-map)# match prefix-list l3extInstP-2`
	`apic1(config-leaf-vrf-route-map-match)# exit`
	`apic1(config-leaf-vrf-route-map)# exit`
	`apic1(config-leaf-vrf)# exit`
	`apic1(config-leaf)# exit`
	`apic1(config)#`	

Configuring Shared Layer 3 Out Inter-VRF Leaking Using the NX-OS Style CLI - Implicit Example

SUMMARY STEPS

1. Enter the configure mode.
2. Configure the provider tenant and VRF.
3. Configure the consumer tenant and VRF.
4. Configure the contract.
5. Configure the provider External Layer 3 EPG.
6. Configure the provider export map.
7. Configure the consumer external Layer 3 EPG.
8. Configure the consumer export map.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	Enter the configure mode.	

	Command or Action	Purpose
	Example: `apic1# configure`	

Step 2	Configure the provider tenant and VRF.
	Example:
	`apic1(config)# tenant t1_provider`
	`apic1(config-tenant)# vrf context VRF1`
	`apic1(config-tenant-vrf)# exit`
	`apic1(config-tenant)# exit`	

Step 3	Configure the consumer tenant and VRF.
	Example:
	`apic1(config)# tenant t1_consumer`
	`apic1(config-tenant)# vrf context VRF2`
	`apic1(config-tenant-vrf)# exit`
	`apic1(config-tenant)# exit`	

Step 4	Configure the contract.
	Example:
	`apic1(config)# tenant t1_provider`
	`apic1(config-tenant)# contract vzBrCP-1 type permit`
	`apic1(config-tenant-contract)# scope exportable`
	`apic1(config-tenant-contract)# export to tenant t1_consumer`
	`apic1(config-tenant-contract)# exit`	

Step 5	Configure the provider External Layer 3 EPG.
	Example:
	`apic1(config-tenant)# external-l3 epg l3extInstP-1`
	`apic1(config-tenant-l3ext-epg)# vrf member VRF1`
	`apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24 shared`
	`apic1(config-tenant-l3ext-epg)# contract provider vzBrCP-1`
	`apic1(config-tenant-l3ext-epg)# exit`
	`apic1(config-tenant)# exit`	

Step 6	Configure the provider export map.
	Example:
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# vrf context tenant t1_provider vrf VRF1`
	`apic1(config-leaf-vrf)# route-map map1`
	`apic1(config-leaf-vrf-route-map)# ip prefix-list p1 permit 192.168.2.0/24`
	`apic1(config-leaf-vrf-route-map)# match prefix-list p1`
	`apic1(config-leaf-vrf-route-map-match)# exit`
	`apic1(config-leaf-vrf-route-map)# exit`
	`apic1(config-leaf-vrf)# export map map1`
	`apic1(config-leaf-vrf)# exit`
	`apic1(config-leaf)# exit`	

	Command or Action	Purpose

Step 7	Configure the consumer external Layer 3 EPG.
	Example:
	`apic1(config)# tenant t1_consumer`
	`apic1(config-tenant)# external-l3 epg l3extInstP-2`
	`apic1(config-tenant-l3ext-epg)# vrf member VRF2`
	`apic1(config-tenant-l3ext-epg)# match ip 199.16.2.0/24 shared`
	`apic1(config-tenant-l3ext-epg)# contract consumer vzBrCP-1 imported`
	`apic1(config-tenant-l3ext-epg)# exit`
	`apic1(config-tenant)# exit`	

Step 8	Configure the consumer export map.
	Example:
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# vrf context tenant t1_consumer vrf VRF2`
	`apic1(config-leaf-vrf)# route-map map2`
	`apic1(config-leaf-vrf-route-map)# ip prefix-list p2 permit 199.16.2.0/24`
	`apic1(config-leaf-vrf-route-map)# match prefix-list p2`
	`apic1(config-leaf-vrf-route-map-match)# exit`
	`apic1(config-leaf-vrf-route-map)# exit`
	`apic1(config-leaf-vrf)# export map map2`
	`apic1(config-leaf-vrf)# exit`
	`apic1(config-leaf)# exit`
	`apic1(config)#`	

Configuring QoS for L3Outs Using the NX-OS Style CLI

Configuring QoS Directly on L3Out Using CLI

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

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

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

Procedure

Step 1 Configure QoS priorities for a L3Out SVI.

Example:

```
interface vlan 19
    vrf member tenant DT vrf dt-vrf
```

```
    ip address 107.2.1.252/24
    description  'SVI19'
    service-policy type qos VrfQos006   // for custom QoS attachment
    set qos-class level6                // for set QoS priority
    exit
```

Step 2 Configure QoS priorities for a sub-interface.

Example:

```
interface ethernet 1/48.10
    vrf member tenant DT vrf inter-tentant-ctx2 l3out L4_E48_inter_tennant
    ip address 210.2.0.254/16
    service-policy type qos vrfQos002
    set qos-class level5
```

Step 3 Configure QoS priorities for a routed outside.

Example:

```
interface ethernet 1/37
    no switchport
    vrf member tenant DT vrf dt-vrf l3out L2E37
    ip address 30.1.1.1/24
    service-policy type qos vrfQos002
    set qos-class level5
    exit
```

Configuring QoS Contract for L3Out Using CLI

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

> **Note** Starting with Release 4.0(1), we recommend using custom QoS policies for L3Out QoS as described in Configuring QoS Directly on L3Out Using CLI, on page 71 instead.

Procedure

Step 1 Configure the VRF for egress mode and enable policy enforcement to support QoS priority enforcement on the L3Out.

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# contract enforce egress
apic1(config-tenant-vrf)# exit
apic1(congig-tenant)# exit
apic1(config)#
```

Step 2 Configure QoS.

When creating filters (`access-list`), include the **match dscp** command with target DSCP level.

When configuring contracts, include the QoS class for traffic ingressing on the L3Out. Alternatively, you can define a target DSCP value. QoS policies are supported on either the contract or the subject

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

Note
If QoS classification is set in the contract and VRF enforcement is egress, then contract QoS classification would override the L3Out interface QoS or Custom QoS classification.

```
apic1(config)# tenant t1
apic1(config-tenant)# access-list http-filter
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# match dscp EF
apic1(config-tenant-acl)# exit
apic1(config-tenant)# contract httpCtrct
apic1(config-tenant-contract)# scope vrf
apic1(config-tenant-contract)# qos-class level1
apic1(config-tenant-contract)# subject http-subject
apic1(config-tenant-contract-subj)# access-group http-filter both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
apic1(config-tenant)# exit
apic1(config)#
```

Configuring ACI IP SLAs Using the NX-OS Style CLI

Configuring an IP SLA Monitoring Policy Using the NX-OS-Style CLI

To configure Cisco Application Policy Infrastructure Controller (APIC) to send monitoring probes for a specific SLA type using the NX-OS-style CLI, perform the following steps:

Before you begin

Make sure a tenant is configured.

Procedure

Step 1 Enter the configuration mode.

Example:

```
apic1# configure
```

Step 2 Create a tenant and enter tenant configuration mode, or enter tenant configuration mode for an existing tenant.

Example:

```
apic1(config)# tenant t1
```

Step 3 Create an IP SLA monitoring policy and enter IP SLA policy configuration mode.

Example:

```
apic1(config-tenant)# ipsla-pol ipsla-policy-3
```

Step 4 Configure the monitoring frequency in seconds, which is the interval between sending probes.

Example:

```
apic1(config-ipsla-pol)# sla-frequency 40
```

Step 5 Configure the monitoring probe type.

The possible values for the type are:

- `icmp`
- `l2ping`
- `tcp sla-port` *number*

Only ICMP and TCP are valid for IP SLA in static routes.

Example:

```
apic1(config-ipsla-pol)# sla-type tcp sla-port 90
```

What to do next

To view the IP SLA monitoring policy you just created, enter:

show running-config all tenant *tenant-name* ipsla-pol

The following output appears:

```
# Command: show running-config all tenant 99 ipsla-pol
# Time: Tue Mar 19 19:01:06 2019
  tenant t1
    ipsla-pol ipsla-policy-3
      sla-detectmultiplier 3
      sla-frequency 40
      sla-type tcp sla-port 90
        sla-port 90
        exit
      exit
    exit
```

Configuring an IP-SLA Track Member Using the NX-OS Style CLI

To configure an IP SLA track member using the NX-OS style CLI, perform the following steps:

Before you begin

Make sure a tenant and an IP SLA monitoring policy under the tenant is configured.

Procedure

Step 1 **configure**

Enters configuration mode.

Example:

```
apic1# configure
```

Step 2 **tenant** *tenant-name*

Creates a tenant or enters tenant configuration mode.

Example:

```
apic1(config)# tenant t1
```

Step 3 **track-member** *name* **dst-IpAddr** *ipv4-or-ipv6-address* **l3-out** *name*

Creates a track member with a destination IP address and enters track member configuration mode.

Example:

```
apic1(config-tenant)# )# track-member tm-1 dst-IpAddr 10.10.10.1 l3-out ext-l3-1
```

Step 4 **ipsla-monpol** *name*

Assigns an IP SLA monitoring policy to the track member.

Example:

```
apic1(config-track-member)# ipsla-monpol ipsla-policy-3
```

Example

The following example shows the commands to configure an IP SLA track member.

```
apic1# configure
      apic1(config)# tenant t1
      apic1(config-tenant)# )# track-member tm-1 dst-IpAddr 10.10.10.1 l3-out ext-l3-1
      apic1(config-track-member)# ipsla-monpol ipsla-policy-3
```

What to do next

To view the track member configuration you just created, enter:

`show running-config all tenant` *`tenant-name`* **`track-member`** *`name`*

The following output appears:

```
# Command: show running-config all tenant 99 track-member tm-1
# Time: Tue Mar 19 19:01:06 2019
  tenant t1
    track-member tm-1 10.10.10.1 l3-out ext-l3-1
      ipsla-monpol slaICMPProbe
      exit
    exit
```

Configuring an IP-SLA Track List Using the NX-OS Style CLI

To configure an IP SLA track list using the NX-OS style CLI, perform the following steps:

Before you begin

Make sure a tenant, an IP SLA monitoring policy, and at least one track member under the tenant is configured.

Procedure

Step 1 **configure**

Enters configuration mode.

Example:

```
apic1# configure
```

Step 2 **tenant** *tenant-name*

Creates a tenant or enters tenant configuration mode.

Example:

```
apic1(config)# tenant t1
```

Step 3 **track-list** *name* **{ percentage [percentage-down | percentage-up]** *number* **| weight [weight-down | weight-up** *number* **}**

Creates a track list with percentage or weight threshold settings and enters track list configuration mode.

Example:

```
apic1(config-tenant)# )# track-list tl-1 percentage percentage-down 50 percentage-up 100
```

Step 4 **track-member** *name*

Assigns an existing track member to the track list.

Example:

```
apic1(config-track-list)# track-member tm-1
```

Example

The following example shows the commands to configure an IP SLA track list.

```
apic1# configure
      apic1(config)# tenant t1
      apic1(config-tenant)# )# track-list tl-1 percentage percentage-down 50 percentage-up
 100
      apic1(config-track-list)# track-member tm1
```

What to do next

To view the track member configuration you just created, enter:

show running-config all tenant *tenant-name* **track-member** *name*

The following output appears:

```
# Command: show running-config all tenant 99 track-list tl-1
# Time: Tue Mar 19 19:01:06 2019
  tenant t1
    track-list tl-1 percentage percentage-down 50 percentage-up 100
      track-member tm-1 weight 10
```

```
  exit
exit
```

Associating a Track List with a Static Route Using the NX-OS Style CLI

To associate an IP SLA track list with a static route using the NX-OS style CLI, perform the following steps:

Before you begin

Make sure a tenant, a VRF, and a track list under the tenant is configured.

Procedure

Step 1 **configure**

Enters configuration mode.

Example:

```
apic1# configure
```

Step 2 **leaf** *id or leaf-name*

Selects a leaf switch and enter the leaf switch configuration mode.

Example:

```
apic1(config)# leaf 102
```

Step 3 **vrf context tenant** *name* **vrf** *name*

Selects a VRF context and enters the VRF configuration mode.

Example:

```
apic1(config-leaf)# )# vrf context tenant 99 vrf default
```

Step 4 **ip route** *ip-address next-hop-ip-address route-prefix* **bfd ip-trackList** *name*

Assigns an existing track list to the static route.

Example:

```
apic1(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd ip-trackList tl-1
```

Example

The following example shows the commands to associate an IP SLA track list with a static route.

```
apic1# configure
      apic1(config)# leaf 102
      apic1(config-leaf)# )# vrf context tenant 99 vrf default
      apic1(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd ip-trackList tl-1
```

Associating a Track List with a Next Hop Profile Using the NX-OS Style CLI

To associate an IP SLA track list with a next hop profile using the NX-OS style CLI, perform the following steps:

Before you begin

Make sure a tenant, a VRF, and a track list under the tenant is configured.

Procedure

Step 1 **configure**

Enters configuration mode.

Example:

```
apic1# configure
```

Step 2 **leaf** *id or leaf-name*

Selects a leaf switch and enter the leaf switch configuration mode.

Example:

```
apic1(config)# leaf 102
```

Step 3 **vrf context tenant** *name* **vrf** *name*

Selects a VRF context and enters the VRF configuration mode.

Example:

```
apic1(config-leaf)# )# vrf context tenant 99 vrf default
```

Step 4 **ip route** *ip-address next-hop-ip-address route-prefix* **bfd nh-ip-trackList** *name*

Assigns an existing track list to the next hop.

Example:

```
apic1(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd nh-trackList tl-1
```

Example

The following example shows the commands to associate an IP SLA track list with a next hop profile.

```
apic1# configure
      apic1(config)# leaf 102
      apic1(config-leaf)# )# vrf context tenant 99 vrf default
      apic1(config-leaf-vrf)# ip route 10.10.10.1/4 20.20.20.8 10 bfd nh-ip-trackList tl-1
```

Viewing Track List and Track Member Status Using the CLI

You can display IP SLA track list and track member status.

Procedure

	Command or Action	Purpose
Step 1	show track brief **Example:** `switch# show track brief`	Displays the status of all track lists and track members.

Example

```
switch# show track brief
TrackId  Type     Instance    Parameter        State    Last Change
97       IP SLA   2034        reachability     up       2019-03-20T14:08:34.127-07:00
98       IP SLA   2160        reachability     up       2019-03-20T14:08:34.252-07:00
99       List     ---         percentage       up       2019-03-20T14:08:45.494-07:00
100      List     ---         percentage       down     2019-03-20T14:08:45.039-07:00
101      List     ---         percentage       down     2019-03-20T14:08:45.040-07:00
102      List     ---         percentage       up       2019-03-20T14:08:45.495-07:00
103      IP SLA   2040        reachability     up       2019-03-20T14:08:45.493-07:00
104      IP SLA   2887        reachability     down     2019-03-20T14:08:45.104-07:00
105      IP SLA   2821        reachability     up       2019-03-20T14:08:45.494-07:00
1        List     ---         percentage       up       2019-03-20T14:08:39.224-07:00
2        List     ---         weight           down     2019-03-20T14:08:33.521-07:00
3        IP SLA   2412        reachability     up       2019-03-20T14:08:33.983-07:00
26       IP SLA   2320        reachability     up       2019-03-20T14:08:33.988-07:00
27       IP SLA   2567        reachability     up       2019-03-20T14:08:33.987-07:00
28       IP SLA   2598        reachability     up       2019-03-20T14:08:33.990-07:00
29       IP SLA   2940        reachability     up       2019-03-20T14:08:33.986-07:00
30       IP SLA   2505        reachability     up       2019-03-20T14:08:38.915-07:00
31       IP SLA   2908        reachability     up       2019-03-20T14:08:33.990-07:00
32       IP SLA   2722        reachability     up       2019-03-20T14:08:33.992-07:00
33       IP SLA   2753        reachability     up       2019-03-20T14:08:38.941-07:00
34       IP SLA   2257        reachability     up       2019-03-20T14:08:33.993-07:00
```

Viewing Track List and Track Member Detail Using the CLI

You can display IP SLA track list and track member detail.

Procedure

	Command or Action	Purpose
Step 1	show track [*number*] | more **Example:** `switch# show track | more`	Displays the detail of all track lists and track members.

Example

```
switch# show track | more
Track 4
    IP SLA 2758
    reachability is down
```

```
    1 changes, last change 2019-03-12T21:41:34.729+00:00
    Tracked by:
        Track List 3
        Track List 5

Track 3
    List Threshold percentage
    Threshold percentage is down
    1 changes, last change 2019-03-12T21:41:34.700+00:00
    Threshold percentage up 1% down 0%
    Tracked List Members:
        Object 4 (50)% down
        Object 6 (50)% down
    Attached to:
        Route prefix 172.16.13.0/24

Track 5
    List Threshold percentage
    Threshold percentage is down
    1 changes, last change 2019-03-12T21:41:34.710+00:00
    Threshold percentage up 1% down 0%
    Tracked List Members:
        Object 4 (100)% down
    Attached to:
        Nexthop Addr 12.12.12.2/32

Track 6
    IP SLA 2788
    reachability is down
    1 changes, last change 2019-03-14T21:34:26.398+00:00
    Tracked by:
        Track List 3
        Track List 7

Track 20
    List Threshold percentage
    Threshold percentage is up
    4 changes, last change 2019-02-21T14:04:21.920-08:00
    Threshold percentage up 100% down 32%
    Tracked List Members:
    Object 4 (20)% up
    Object 5 (20)% up
    Object 6 (20)% up
    Object 3 (20)% up
    Object 9 (20)% up
    Attached to:
    Route prefix 88.88.88.0/24
    Route prefix 5000:8:1:14::/64
    Route prefix 5000:8:1:2::/64
    Route prefix 5000:8:1:1::/64
```

In this example, Track 4 is a track member identified by the IP SLA ID and by the track lists in the **Tracked by:** field.

Track 3 is a track list identified by the threshold information and the track member in the **Track List Members** field.

Track 20 is a track list that is currently reachable (up) and shows the static routes to which it is associated.

Configuring HSRP Using the NX-OS Style CLI

Configuring HSRP in Cisco APIC Using Inline Parameters in NX-OS Style CLI

HSRP is enabled when the leaf switch is configured.

Before you begin

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

SUMMARY STEPS

1. **configure**
2. Configure HSRP by creating inline parameters.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.

Step 2	Configure HSRP by creating inline parameters.
	Example:
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# interface ethernet 1/17`
	`apic1(config-leaf-if)# hsrp version 1`
	`apic1(config-leaf-if)# hsrp use-bia`
	`apic1(config-leaf-if)# hsrp delay minimum 30`
	`apic1(config-leaf-if)# hsrp delay reload 30`
	`apic1(config-leaf-if)# hsrp 10 ipv4`
	`apic1(config-if-hsrp)# ip 182.16.1.2`
	`apic1(config-if-hsrp)# ip 182.16.1.3 secondary`
	`apic1(config-if-hsrp)# ip 182.16.1.4 secondary`
	`apic1(config-if-hsrp)# mac-address 5000.1000.1060`
	`apic1(config-if-hsrp)# timers 5 18`
	`apic1(config-if-hsrp)# priority 100`
	`apic1(config-if-hsrp)# preempt`
	`apic1(config-if-hsrp)# preempt delay minimum 60`
	`apic1(config-if-hsrp)# preempt delay reload 60`
	`apic1(config-if-hsrp)# preempt delay sync 60`
	`apic1(config-if-hsrp)# authentication none`
	`apic1(config-if-hsrp)# authentication simple`	

	Command or Action	Purpose

	`apic1(config-if-hsrp)# authentication md5`
	`apic1(config-if-hsrp)# authentication-key <mypassword>`
	`apic1(config-if-hsrp)# authentication-key-timeout <timeout>`	

Configuring HSRP in Cisco APIC Using Template and Policy in NX-OS Style CLI

HSRP is enabled when the leaf switch is configured.

Before you begin

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

SUMMARY STEPS

1. **configure**
2. Configure HSRP policy templates.
3. Use the configured policy templates

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.

Step 2	Configure HSRP policy templates.
	Example:
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# template hsrp interface-policy hsrp-intfPol1 tenant t9`
	`apic1(config-template-hsrp-if-pol)# hsrp use-bia`
	`apic1(config-template-hsrp-if-pol)# hsrp delay minimum 30`
	`apic1(config-template-hsrp-if-pol)# hsrp delay reload 30`
	
	`apic1(config)# leaf 101`	

	Command or Action	Purpose

	`apic1(config-leaf)# template hsrp group-policy hsrp-groupPol1 tenant t9`
	`apic1(config-template-hsrp-group-pol)# timers 5 18`
	`apic1(config-template-hsrp-group-pol)# priority 100`
	`apic1(config-template-hsrp-group-pol)# preempt`
	`apic1(config-template-hsrp-group-pol)# preempt delay minimum 60`
	`apic1(config-template-hsrp-group-pol)# preempt delay reload 60`
	`apic1(config-template-hsrp-group-pol)# preempt delay sync 60`	

Step 3	Use the configured policy templates
	Example:
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# interface ethernet 1/17`
	`apic1(config-leaf-if)# hsrp version 1`
	`apic1(config-leaf-if)# inherit hsrp interface-policy hsrp-intfPol1`
	`apic1(config-leaf-if)# hsrp 10 ipv4`
	`apic1(config-if-hsrp)# ip 182.16.1.2`
	`apic1(config-if-hsrp)# ip 182.16.1.3 secondary`
	`apic1(config-if-hsrp)# ip 182.16.1.4 secondary`
	`apic1(config-if-hsrp)# mac-address 5000.1000.1060`
	`apic1(config-if-hsrp)# inherit hsrp group-policy hsrp-groupPol1`	

Configuring Cisco ACI GOLF Using the NX-OS Style CLI

Recommended Shared GOLF Configuration Using the NX-OS Style CLI

Use the following steps to configure route maps and BGP to avoid cross-VRF traffic issues when sharing GOLF connections with a DCI between multiple APIC sites that are managed by Multi-Site.

Procedure

Step 1 Configure the inbound route map

Example:

```
Inbound peer policy to attach community:

 route-map multi-site-in permit 10

  set community 1:1 additive
```

Step 2 Configure the outbound peer policy to filter routes based on the community in the inbound peer policy.

Example:

```
ip community-list standard test-com permit 1:1

route-map multi-site-out deny 10

  match community test-com exact-match

route-map multi-site-out permit 11
```

Step 3 Configure the outbound peer policy to filter the community towards the WAN.

Example:

```
ip community-list standard test-com permit 1:1

route-map multi-site-wan-out permit 11

  set comm-list test-com  delete
```

Step 4 Configure BGP.

Example:

```
router bgp 1

  address-family l2vpn evpn

  neighbor 11.11.11.11 remote-as 1

    update-source loopback0

    address-family l2vpn evpn

      send-community both

      route-map multi-site-in in

  neighbor 13.0.0.2 remote-as 2

    address-family l2vpn evpn

      send-community both

      route-map multi-site-out out
```

Cisco ACI GOLF Configuration Example, Using the NX-OS Style CLI

These examples show the CLI commands to configure GOLF Services, which uses the BGP EVPN protocol over OSPF for WAN routers that are connected to spine switches.

Configuring the infra Tenant for BGP EVPN

The following example shows how to configure the infra tenant for BGP EVPN, including the VLAN domain, VRF, Interface IP addressing, and OSPF:

```
configure
  vlan-domain evpn-dom dynamic
  exit
  spine 111
       # Configure  Tenant Infra VRF overlay-1 on the spine.
    vrf context tenant infra vrf overlay-1
```

```
        router-id 10.10.3.3
        exit

    interface ethernet 1/33
         vlan-domain member golf_dom
         exit
    interface ethernet 1/33.4
         vrf member tenant infra vrf overlay-1
         mtu 1500
         ip address 5.0.0.1/24
         ip router ospf default area 0.0.0.150
         exit
    interface ethernet 1/34
         vlan-domain member golf_dom
        exit
    interface ethernet 1/34.4
        vrf member tenant infra vrf overlay-1
        mtu 1500
        ip address 2.0.0.1/24
        ip router ospf default area 0.0.0.200
       exit

    router ospf default
       vrf member tenant infra vrf overlay-1
           area 0.0.0.150 loopback 10.10.5.3
           area 0.0.0.200 loopback 10.10.4.3
           exit
       exit
```

Configuring BGP on the Spine Node

The following example shows how to configure BGP to support BGP EVPN:

```
Configure
spine 111
router bgp 100
    vrf member tenant infra vrf overlay- 1
         neighbor 10.10.4.1 evpn
             label golf_aci
             update-source loopback 10.10.4.3
             remote-as 100
             exit
         neighbor 10.10.5.1 evpn
             label golf_aci2
             update-source loopback 10.10.5.3
             remote-as 100
             exit
    exit
exit
```

Configuring a Tenant for BGP EVPN

The following example shows how to configure a tenant for BGP EVPN, including a gateway subnet which will be advertised through a BGP EVPN session:

```
configure
  tenant sky
    vrf context vrf_sky
      exit
    bridge-domain bd_sky
      vrf member vrf_sky
```

```
      exit
    interface bridge-domain bd_sky
      ip address 59.10.1.1/24
      exit
    bridge-domain bd_sky2
      vrf member vrf_sky
      exit
    interface bridge-domain bd_sky2
      ip address 59.11.1.1/24
      exit
    exit
```

Configuring the BGP EVPN Route Target, Route Map, and Prefix EPG for the Tenant

The following example shows how to configure a route map to advertise bridge-domain subnets through BGP EVPN.

```
configure
spine 111
    vrf context tenant sky vrf vrf_sky
        address-family ipv4 unicast
            route-target export 100:1
            route-target import 100:1
             exit

        route-map rmap
            ip prefix-list p1 permit 11.10.10.0/24
            match bridge-domain bd_sky
                exit
            match prefix-list p1
                exit

         evpn export map rmap label golf_aci

          route-map rmap2
           match bridge-domain bd_sky
               exit
           match prefix-list p1
              exit
          exit

         evpn export map rmap label golf_aci2

    external-l3 epg l3_sky
      vrf member vrf_sky
      match ip 80.10.1.0/24
      exit
```

Enabling Distributing BGP EVPN Type-2 Host Routes to a DCIG Using the NX-OS Style CLI

Procedure

	Command or Action	Purpose
Step 1	Configure distributing EVPN type-2 host routes to a DCIG with the following commands in the BGP address family configuration mode. **Example:**	This template will be available on all nodes where tenant bgp_t1 has a VRF deployment. To disable distributing EVPN type-2 host routes, enter the **no host-rt-enable** command.

	Command or Action	Purpose
	apic1(config)# **leaf 101** apic1(config-leaf)# **template bgp address-family bgpAf1 tenant bgp_t1** apic1(config-bgp-af)# **distance 250 240 230** apic1(config-bgp-af)# **host-rt-enable** apic1(config-bgp-af)# **exit**	

Performing Tasks Using the NX-OS Style CLI