Cisco APIC Layer 4 to Layer 7 Services Deployment Guide, Release 5.2(x)

First Published: 2021-06-04

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Trademarks

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: http://www.cisco.com/go/softwareterms.Cisco product warranty information is available at http://www.cisco.com/go/warranty. US Federal Communications Commission Notices are found here http://www.cisco.com/c/en/us/products/us-fcc-notice.html.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com go trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

Trademarks

CONTENTS

PREFACE

Trademarks iii

CHAPTER 1

New and Changed Information 1

New and Changed Information 1

CHAPTER 2

Overview 3

About Deploying Application-Centric Infrastructure Layer 4 to Layer 7 Services 3

About Layer 4 to Layer 7 Service Devices 4

About Service Graph Templates 4

Configuring Layer 4 to Layer 7 Services Using the GUI 5

CHAPTER 3

Defining a Logical Device 7

About Device Clusters 7

About Concrete Devices 8

About Trunking 8

About Layer 4 to Layer 7 Services Endpoint Groups 9

Using Static Encapsulation for a Graph Connector 9

Configuring a Layer 4 to Layer 7 Services Device Using the GUI 9

Creating a Layer 4 to Layer 7 Device Using the NX-OS-Style CLI 12

Creating a High Availablity Cluster Using the NX-OS-Style CLI 16

Creating a Virtual Device Using the NX-OS-Style CLI 17

Example XML for Creating a Logical Device 18

Example XML of Creating an LDevVip Object 18

Example XML of Creating an AbsNode Object 18

Example XML of Associating a Layer 4 to Layer 7 Service Endpoint Group with a Connector 19

Example XML of Using Static Encapsulation with a Layer 4 to Layer 7 Service Endpoint Group 20

Modifying a Device Using the GUI 20
Enabling Trunking on a Layer 4 to Layer 7 Virtual ASA device Using the GUI 20
Enabling Trunking on a Layer 4 to Layer 7 Virtual ASA device Using the REST APIs 21
Using an Imported Device with the REST APIs 21
Importing a Device From Another Tenant Using the NX-OS-Style CLI 22
Verifying the Import of a Device Using the GUI 22

CHAPTER 4 **Service VM Orchestration** 23
Service VM Orchestration 23
Service VM Orchestration Guidelines and Limitations 24
Configuring Service VM Orchestration Using the Cisco APIC GUI 24
Creating a VM Instantiation Policy Using the Cisco APIC GUI 24
Creating a Layer 4 to Layer 7 services device and Associating the Device with the VM Instantiation Policy Using the GUI 25
Configuring Service VM Orchestration Using the NX-OS Style CLI 30
Configuring Service VM Orchestration Using REST API 30
Troubleshooting Service VM Orchestration 32
Service VM Templates Not Seen in VM Instantiation Policy 32
Port Groups Created in VMware vCenter Do Not Appear for CDev 33
Unable to Reach Service VM IP Address 33
Device State Shows Init 33
LIF Configuration Is Invalid 34

CHAPTER 5 **Selecting a Layer 4 to Layer 7 Device to Render a Graph** 35
About Device Selection Policies 35
Creating a Device Selection Policy Using the GUI 35
Configuring a Device Selection Policy Using REST APIs 38
Creating a Device Selection Policy Using the REST API 38
Adding a Logical Interface in a Device Using the REST APIs 39

CHAPTER 6 **Configuring a Service Graph** 41
About Service Graphs 41
About Function Nodes 43
About Function Node Connectors 43

About Service Graph Connections 44
About Terminal Nodes 44
Guidelines and Limitations for Configuring Service Graph 44
Configuring a Service Graph Template Using the GUI 44
Configuring a Service Graph Template Using the REST APIs 45
Applying a Service Graph Template to Endpoint Groups Using the GUI 46
Applying a Service Graph Template to an Endpoint Security Group Using the GUI 47
Applying a Service Graph Template with a Contract Using the NX-OS-Style CLI 48

CHAPTER 7 **Configuring Route Peering 53**
About Route Peering 53
Open Shortest Path First Policies 54
Border Gateway Protocol Policies 58
Selecting an L3extOut Policy for a Cluster 61
Route Peering End-to-End Flow 62
Cisco Application Centric Infrastructure Fabric Serving As a Transit Routing Domain 64
Configuring Route Peering Using the GUI 65
Creating a Static VLAN Pool Using the GUI 65
Creating an External Routed Domain Using the GUI 66
Creating an External Routed Network Using the GUI 66
Creating a Router Configuration Using the GUI 68
Creating a Service Graph Association Using the GUI 68
Configuring Route Peering Using the NX-OS-Style CLI 69
Troubleshooting Route Peering 71
Verifying the Leaf Switch Route Peering Functionality Using the CLI 72

CHAPTER 8 **Configuring Policy-Based Redirect 75**
About Policy-Based Redirect 75
Guidelines and Limitations for Configuring Policy-Based Redirect 77
Configuring Policy-Based Redirect Using the GUI 83
Configuring Policy-Based Redirect Using the NX-OS-Style CLI 85
Verifying a Policy-Based Redirect Configuration Using the NX-OS-Style CLI 88
About Multi-Node Policy-Based Redirect 89
About Symmetric Policy-Based Redirect 89

Policy Based Redirect and Hashing Algorithms 90
Policy-Based Redirect Resilient Hashing 90
Enabling Resilient Hashing in L4-L7 Policy-Based Redirect 92
About PBR Backup Policy 92
Creating a PBR Backup Policy 94
Enabling a PBR Backup Policy 95
About the Bypass Action 96
Configuring the Threshold Down Action in Policy-Based Redirect 98
Policy-Based Redirect with an L3Out 99
Guidelines and Limitations for Policy-Based Redirect with an L3Out 103
Configuring Policy-Based Redirect with an L3Out Using the GUI 105
PBR Support for Service Nodes in Consumer and Provider Bridge Domains 107
About Layer 1/Layer 2 Policy-Based Redirect 107
Layer 1/Layer 2 PBR Configuration Overview 107
Active/Standby Layer 1/Layer 2 PBR Design Overview 109
Active/Active Layer 1/Layer 2 Symmetric PBR Design Overview 110
Configuring a Layer 1/Layer 2 Device Using the GUI 111
Configuring Layer 1/ Layer 2 PBR Using the APIC GUI 112
Configuring ASA for Layer 1/ Layer 2 PBR Using CLI 113
Verifying Layer 1/ Layer 2 PBR Policy On The Leafs Using CLI 114
Configuring Layer 1/ Layer 2 PBR Using the REST API 116
Policy-Based Redirect and Tracking Service Nodes 117
Policy-Based Redirect and Tracking Service Nodes with a Health Group 117
Policy-Based Redirect and Threshold Settings for Tracking Service Nodes 117
Guidelines and Limitations for Policy-Based Redirect With Tracking Service Nodes 118
Configuring PBR and Tracking Service Nodes Using the GUI 119
Configuring a Redirect Health Group Using the GUI 120
Configuring Global GIPo for Remote Leaf Using the GUI 120
Configuring PBR to Support Tracking Service Nodes Using the REST API 121
About Location-Aware Policy Based Redirect 121
Guidelines for Location-Aware PBR 122
Configuring Location-Aware PBR Using the GUI 123
Configuring Location-Aware PBR Using the REST API 123

Policy-Based Redirect and Service Graphs to Redirect All EPG-to-EPG Traffic Within the Same VRF Instance 124

Guidelines and Limitations for Configuring a Policy-Based Redirect Policy with a Service Graph to Redirect All EPG-to-EPG Traffic Within the Same VRF Instance 127

Configuring a Policy-Based Redirect Policy with a Service Graph to Redirect All EPG-to-EPG Traffic Within the Same VRF Instance 127

Dynamic MAC Address Detection for a Layer 3 Policy-Based Redirect Destination 129

Guidelines and Limitations for Dynamic MAC Address Detection for a Layer 3 Policy-Based Redirect Destination 129

Configuring Dynamic MAC Address Detection for a Layer 3 Policy-Based Redirect Destination Using the GUI 130

Configuring Dynamic MAC Address Detection for a Layer 3 Policy-Based Redirect Destination Using the REST API 130

CHAPTER 9 **Configuring Direct Server Return** 131

About Direct Server Return 131

Layer 2 Direct Server Return 132

About Deploying Layer 2 Direct Server Return with Cisco Application Centric Infrastructure 133

Guidelines and Limitations for Configuring Direct Server Return 134

Supported Direct Server Return Configuration 135

Example XML POST of Direct Server Return for Static Service Deployment 135

Direct Server Return for Static Service Deployment 136

Direct Server Return for Static Service Deployment Logical Model 136

Direct Server Return for Service Graph Insertion 136

Direct Server Return Shared Layer 4 to Layer 7 Service Configuration 137

Configuring the Citrix Server Load Balancer for Direct Server Return 137

Configuring a Linux Server for Direct Server Return 137

CHAPTER 10 **Configuring Copy Services** 139

About Copy Services 139

Copy Services Limitations 140

Configuring Copy Services Using the GUI 140

Creating a Copy Device Using the GUI 141

Configuring Copy Services Using the NX-OS-Style CLI 142

Configuring Copy Services Using the REST API 144

CHAPTER 11 **Configuring Layer 4 to Layer 7 Resource Pools** 147

About Layer 4 to Layer 7 Resource Pools 147

About External and Public IP Address Pools 147

About External Layer 3 Routed Domains and the Associated VLAN Pools 148

About External Routed Networks 148

Creating an IP Address Pool for Layer 4 to Layer 7 Resource Pools Using the GUI 149

Creating a Dynamic VLAN Pool for Layer4 to Layer 7 Resource Pools Using the GUI 149

Creating an External Routed Domain for Layer 4 to Layer 7 Resource Pools Using the GUI 150

Preparing Layer 4 to Layer 7 Devices for Use in Layer 4 to Layer 7 Resource Pools 150

Validating the APIC Configuration of a Layer 4 to Layer 7 Device for Use in a Layer 4 to Layer 7 Resource Pool 151

Configuring the Device Management Network and Routes 151

Creating a Layer 4 to Layer 7 Resource Pool 152

Creating a Layer 4 to Layer 7 Resource Pool Using the GUI 152

Creating a Layer 4 to Layer 7 Resource Pool Using the NX-OS-Style CLI 152

Configuring a Layer 4 to Layer 7 Resource Pool Using the GUI 153

Configuring Layer 4 to Layer 7 Devices in a Resource Pool 153

Adding Layer 4 to Layer 7 Devices to a Layer 4 to Layer 7 Resouce Pool 153

Removing Layer 4 to Layer 7 Devices from a Layer 4 to Layer 7 Resource Pool 154

Configuring External IP Address Pools in a Resource Pool 154

Adding an External IP Address Pool to a Layer 4 to Layer 7 Resource Pool 154

Removing an External IP Address Pool from a Layer 4 to Layer 7 Resource Pool 155

Configuring Public IP Address Pools in a Resource Pool 156

Adding Public IP Address Pools to a Layer 4 to Layer 7 Resource Pool 156

Removing Public IP Address Pools from a Layer 4 to Layer 7 Resource Pool 156

Updating an External Routed Domain for a Layer 4 to Layer 7 Resource Pool 157

Updating External Routed Networks for a Layer 4 to Layer 7 Resource Pool 157

CHAPTER 12 **Monitoring a Service Graph** 159

Monitoring a Service Graph Instance Using the GUI 159

Monitoring Service Graph Faults Using the GUI 160

Resolving Service Graph Faults 160

Monitoring a Virtual Device Using the GUI 164

Monitoring Device Cluster and Service Graph Status Using the NX-OS-Style CLI 165

CHAPTER 13 Configuring Multi-Tier Application with Service Graph 169

About Multi-Tier Application with Service Graph 169

Creating a Multi-Tier Application Profile Using the GUI 169

CHAPTER 14 Configuring Administrator Roles for Managing a Service Configuration 173

About Privileges 173

Configuring a Role for Device Management 174

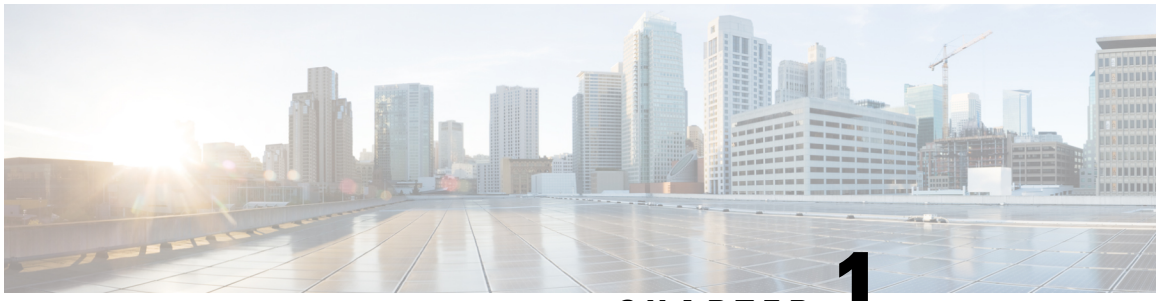
Configuring a Role for Service Graph Template Management 174

Configuring a Role for Exporting Devices 174

CHAPTER 15 Developing Automation 175

About the REST APIs 175

Examples of Automating Using the REST APIs 176



CHAPTER 1

New and Changed Information

• New and Changed Information, on page 1

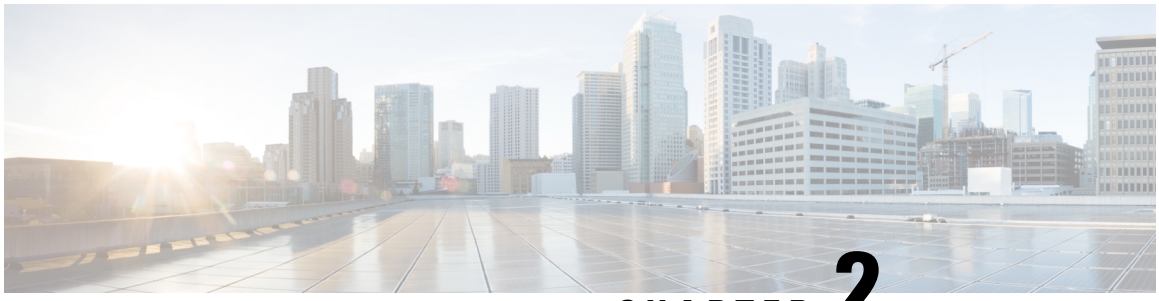
New and Changed Information

The following table provides an overview of the significant changes to this guide for this release. The table does not provide an exhaustive list of all changes made to the guide or of the new features up to this release.

Table 1: New Features and Changed Behavior for Cisco APIC Release 5.2(1)

Feature	Description	Where Documented
Dynamic MAC address detection for a Layer 3 policy-based redirect destination	You can configure any of the Layer 3 policy-based redirect (PBR) destinations without specifying a MAC address, which causes the leaf switches to use the Address Resolution Protocol (ARP) to determine the MAC address of the PBR next-hop. The benefit is that you do not need to check the MAC address of each PBR destination and an active-standby HA pair does not need to use a floating MAC address.	Dynamic MAC Address Detection for a Layer 3 Policy-Based Redirect Destination, on page 129
Policy-based redirect destination in an L3Out	A PBR destination can now be in an L3Out.	Policy-Based Redirect with an L3Out, on page 99
HTTP URI tracking	You can track service nodes using the HTTP URI.	Policy-Based Redirect and Tracking Service Nodes, on page 117
End of support for device packages	Device packages are no longer supported. There is no longer a managed mode for devices; all devices are effectively unmanaged.	N/A

New and Changed Information



CHAPTER 2

Overview

- About Deploying Application-Centric Infrastructure Layer 4 to Layer 7 Services, on page 3
- About Layer 4 to Layer 7 Service Devices, on page 4
- About Service Graph Templates, on page 4
- Configuring Layer 4 to Layer 7 Services Using the GUI, on page 5

About Deploying Application-Centric Infrastructure Layer 4 to Layer 7 Services

Traditionally, when you insert services into a network, you must perform a highly manual and complicated VLAN (Layer 2) or virtual routing and forwarding (VRF) instance (Layer 3) stitching between network elements and service appliances. This traditional model requires days or weeks to deploy new services for an application. The services are less flexible, operating errors are more likely, and troubleshooting is more difficult. When an application is retired, removing a service device configuration, such as firewall rules, is difficult. Scale out/scale down of services that is based on the load is also not feasible.

Although VLAN and virtual routing and forwarding (VRF) stitching is supported by traditional service insertion models, the Application Policy Infrastructure Controller (APIC) can automate service insertion while acting as a central point of policy control. The Cisco APIC policies manage both the network fabric and services appliances. The Cisco APIC can configure the network automatically so that traffic flows through the services. The Cisco APIC can also automatically configure the service according to the application's requirements, which allows organizations to automate service insertion and eliminate the challenge of managing the complex techniques of traditional service insertion.

Before you begin, the following Cisco APIC objects must be configured:

- The tenant that will provide/consume the Layer 4 to Layer 7 services
- A Layer 3 outside network for the tenant
- At least one bridge domain
- An application profile
- A physical domain or a VMM domain

  For a VMM domain, configure VMM domain credentials and configure a vCenter/vShield controller profile.
- A VLAN pool with an encapsulation block range

- At least one contract
- At least one EPG

You must perform the following tasks to deploy Layer 4 to Layer 7 services:

1. Register the device and the logical interfaces.

   This task also registers concrete devices and concrete interfaces.
2. Create a **Logical Device**.
3. Optional. If you are configuring an ASA firewall service, enable trunking on the device.
4. Configure a **Device Selection Policy**.
5. Configure a **Service Graph Template**.
6. Attach the service graph template to a contract.

Note Virtualized appliances can be deployed with VLANs as the transport between VMware ESX servers and leaf nodes, and can be deployed only with VMware ESX as the hypervisor.

About Layer 4 to Layer 7 Service Devices

A Layer 4 to Layer 7 service device is a functional component that is connected to a fabric, such as a firewall, Intrusion-Prevention System (IPS), or load balancer.

About Service Graph Templates

The Cisco Application Centric Infrastructure (ACI) allows you to define a sequence of meta-devices, such a firewall of a certain type followed by a load balancer of a certain make and version. This is called an service graph template, also known as an abstract graph. When a service graph template is referenced by a contract, the service graph template is instantiated by mapping it to concrete devices, such as the firewall and load balancers that are present in the fabric. The mapping happens with the concept of a *context*. The *device context* is the mapping configuration that allows Cisco ACI to identify which firewalls and which load balancers can be mapped to the service graph template. Another key concept is the *logical device*, which represents the cluster of concrete devices. The rendering of the service graph template is based on identifying the suitable logical devices that can be inserted in the path that is defined by a contract.

Cisco ACI treats services as an integral part of an application. Any services that are required are treated as a service graph that is instantiated on the Cisco ACI fabric from the Cisco Application Policy Infrastructure Controller (APIC). Users define the service for the application, while service graph templates identify the set of network or service functions that are needed by the application. Once the graph is configured in the Cisco APIC, the Cisco APIC automatically configures the services according to the service function requirements that are specified in the service graph template. The Cisco APIC also automatically configures the network according to the needs of the service function that is specified in the service graph template, which does not require any change in the service device.

Configuring Layer 4 to Layer 7 Services Using the GUI

The following list provides an overview of how to configure the Layer 4 to Layer 7 services using the GUI:

1. Configure a device.

   See Configuring a Layer 4 to Layer 7 Services Device Using the GUI, on page 9.

   (Optional) Modify a device.

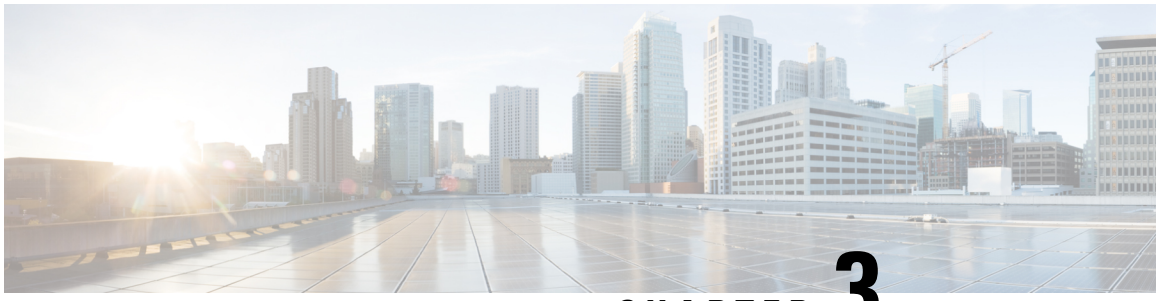
   See Modifying a Device Using the GUI, on page 20.

2. Configure a service graph template.

   See Configuring a Service Graph Template Using the GUI, on page 44.

3. Apply a service graph template to endpoint groups (EPGs).

   See Applying a Service Graph Template to Endpoint Groups Using the GUI, on page 46.



CHAPTER 3

Defining a Logical Device

- About Device Clusters, on page 7
- About Concrete Devices, on page 8
- About Trunking, on page 8
- About Layer 4 to Layer 7 Services Endpoint Groups, on page 9
- Using Static Encapsulation for a Graph Connector, on page 9
- Configuring a Layer 4 to Layer 7 Services Device Using the GUI, on page 9
- Creating a Layer 4 to Layer 7 Device Using the NX-OS-Style CLI, on page 12
- Creating a High Availablity Cluster Using the NX-OS-Style CLI, on page 16
- Creating a Virtual Device Using the NX-OS-Style CLI, on page 17
- Example XML for Creating a Logical Device, on page 18
- Modifying a Device Using the GUI, on page 20
- Enabling Trunking on a Layer 4 to Layer 7 Virtual ASA device Using the GUI, on page 20
- Enabling Trunking on a Layer 4 to Layer 7 Virtual ASA device Using the REST APIs, on page 21
- Using an Imported Device with the REST APIs, on page 21
- Importing a Device From Another Tenant Using the NX-OS-Style CLI, on page 22
- Verifying the Import of a Device Using the GUI, on page 22

About Device Clusters

A device cluster (also known as a logical device) is one or more concrete devices that act as a single device. A device cluster has cluster (logical) interfaces, which describe the interface information for the device cluster. During service graph template rendering, function node connectors are associated with cluster (logical) interfaces. The Application Policy Infrastructure Controller (APIC) allocates the network resources (VLAN or Virtual Extensible Local Area Network [VXLAN]) for a function node connector during service graph template instantiation and rendering and programs the network resources onto the cluster (logical) interfaces.

The Cisco APIC allocates only the network resources for the service graph and programs only on the fabric side during graph instantiation. This behavior is useful if your environment already has an existing orchestrator or a dev-op tool that programs the devices in a device cluster.

The Cisco APIC needs to know the topology information (logical interface and concrete interface) for the device cluster and devices. This information enables the Cisco APIC to program the appropriate ports on the leaf switch, and the Cisco APIC can also use this information for troubleshooting wizard purposes. The Cisco APIC also needs to know the relation to `DomP`, which is used for allocating the encapsulation.

A device cluster or logical device can be either physical or virtual. A device cluster is considered virtual when the virtual machines that are part of that cluster reside on an hypervisor that is integrated with Cisco APIC using VMM domains. If these virtual machines are not part of a VMM domain, then they are treated as physical devices even though they are virtual machine instances.

Note You can use only a VMware VMM domain or SCVMM VMM domain for a logical device

The following settings are required:

- Connectivity information for the logical device (`vnsLDevViP`) and devices (`CDev`)
- Information about supported function type (go-through, go-to, L1, L2)

The service graph template uses a specific device that is based on a device selection policy (called a *logical device context*) that an administrator defines.

An administrator can set up a maximum of two concrete devices in active-standby mode.

To set up a device cluster, you must perform the following tasks:

1. Connect the concrete devices to the fabric.
2. Configure the device cluster with the Cisco APIC.

Note The Cisco APIC does not validate a duplicate IP address that is assigned to two device clusters. The Cisco APIC can provision the wrong device cluster when two device clusters have the same management IP address. If you have duplicate IP addresses for your device clusters, delete the IP address configuration on one of the devices and ensure there are no duplicate IP addresses that are provisioned for the management IP address configuration.

About Concrete Devices

A concrete device can be either physical or virtual. If the device is virtual, you must choose the controller (vCenter or SCVMM controller) and the virtual machine name. A concrete device has concrete interfaces. When a concrete device is added to a logical device, the concrete interfaces are mapped to the logical interfaces. During service graph template instantiation, VLANs and VXLANs are programmed on concrete interfaces that are based on their association with logical interfaces.

About Trunking

You can enable trunking for a Layer 4 to Layer 7 virtual ASA device, which uses trunk port groups to aggregate the traffic of endpoint groups. Without trunking, a virtual service device can have only 1 VLAN per interface and up to 10 service graphs. With trunking enabled, the virtual service device can have an unlimited number of service graphs.

For more information about trunk port groups, see the *Cisco ACI Virtualization Guide*.

About Layer 4 to Layer 7 Services Endpoint Groups

The Application Policy Infrastructure Controller (APIC) enables you to specify an endpoint group to be used for the graph connector during graph instantiation. This enables you to better troubleshoot the graph deployment. The APIC uses the Layer 4 to Layer 7 services endpoint group that you specified to download encapsulation information to a leaf switch. The APIC also uses the endpoint group to create port groups on the distributed virtual switch for virtual devices. A Layer 4 to Layer 7 services endpoint group is also used to aggregate faults and statistics information for a graph connector.

In addition to the enhanced visibility into the deployed graph resources, a Layer 4 to Layer 7 services endpoint group can also be used to specify static encapsulation to be used for a specific graph instance. This encapsulation can also be shared across multiple graph instances by sharing the Layer 4 to Layer 7 services endpoint group across multiple graph instances.

For example XML code that shows how you can use a Layer 4 to Layer 7 services endpoint group with a graph connector, see Example XML of Associating a Layer 4 to Layer 7 Service Endpoint Group with a Connector, on page 19.

Using Static Encapsulation for a Graph Connector

The Application Policy Infrastructure Controller (APIC) allocates the encapsulation for various service graphs during processing. In some use cases, you want to be able explicitly to specify the encapsulation to be used for a specific connector in the service graph. This is known as static encapsulation. Static encapsulations are only supported for the service graph connector that has a service device cluster with physical services. Service device clusters with virtual service devices use the VLANs from the VMware or SCVMM domain that is associated with the service device cluster.

A static encapsulation can be used with a graph connector by specifying the encapsulation value as part of the Layer 4 to Layer 7 service endpoint group. For example XML code that shows how you can use a static encapsulation with a Layer 4 to Layer 7 services endpoint group, see Example XML of Using Static Encapsulation with a Layer 4 to Layer 7 Service Endpoint Group, on page 20.

Configuring a Layer 4 to Layer 7 Services Device Using the GUI

When you create a Layer 4 to Layer 7 services device, you can connect to either a physical device or a virtual machine. The fields are slightly different depending on the type to which you are connecting. When you connect to a physical device, you specify the physical interface. When you connect to a virtual machine, you specify the VMM domain, the virtual machine, and the virtual interfaces. Additionally, you can select an unknown model, which allows you to configure the connections manually.

Before you begin

- You must have configured a tenant.

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

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

Step 3 In the Navigation pane, choose **Tenant** *tenant_name* > **Services** > **L4-L7** > **Devices**.

Step 4 In the Work pane, choose **Actions** > **Create L4-L7 Devices**.

Step 5 In the **Create L4-L7 Devices** dialog box, in the **General** section, complete the following fields:

Name	Description
Name field	Enter a name for the device.
Service Type drop-down list	Choose the service type. The types are: • ADC • Firewall • Other **Note** For a Layer 1/Layer 2 firewall configuration, choose **Other**.
Device Type buttons	Choose the device type.
Physical Domain or **VMM Domain** drop-down list	Choose the physical domain or VMM domain.
Switching Mode (Cisco ACI Virtual Edge only)	For a Cisco ACI Virtual Edge virtual domain, choose one of the following modes: • **AVE**: Traffic is switched through the Cisco ACI Virtual Edge. • **native**: Traffic is switched through the VMware DVS.
View radio buttons	Choose the view for the device. The view can be: • **Single Node**: Only one node • **HA Node**: High availability nodes (two nodes) • **Cluster**: 3 or more nodes
Context Aware	The context awareness of the device. The awareness can be: • **Single**: The device cluster cannot be shared across multiple tenants of a given type that are hosted on the provider network. You must give the device cluster to a specific tenant for a given user. • **Multiple**: The device cluster can be shared across multiple tenants of a given type that you are hosting on the provider network. For example, there could be two hosting companies that share the same device. The default is **Single**. **Note** When you create a Layer 4 to Layer 7 services device that is a load balancer, the **Context Aware** parameter is not used and can be ignored. Beginning with the 5.2(1) release, this parameter is deprecated and the Cisco APIC ignores the value.

Name	Description
Function Type	Function types are: • **GoThrough**: Transparent mode • **GoTo**: Routed mode • **L1**: Layer 1 firewall mode • **L2**: Layer 2 firewall mode The default is **GoTo**. **Note** For Layer 1 or Layer 2 mode, check the check box to enable Active-Active mode. When enabled, active-active deployment/ECMP paths for Layer 1/Layer 2 PBR devices are supported.

Step 6 In the **Device 1** section, complete the following fields:

Name	Description
VM drop-down list	(Only for the virtual device type) Choose a virtual machine.

Step 7 In the **Device Interfaces** table, click the + button to add an interface and complete the following fields:

Name	Description
Name drop-down list	Choose the interface name.
VNIC drop-down list	(Only for the virtual device type) Choose a vNIC.
Path drop-down list	(Only for the physical device type or for an interface in L3Out) Choose a port, port channel, or virtual port channel to which the interface will connect.

Step 8 Click **Update**.

Step 9 (Only for an HA cluster) Complete the fields for each device.

Step 10 Complete the fields for the **Cluster Interfaces** section.

Click (+) to add a cluster interface and complete the following details:

Name	Description
Name drop-down list	Enter a name for the cluster interface.
Concrete Interfaces drop-down list	Select a concrete interface. The interfaces in the drop-down list are based on the device interfaces created in step 7.
Enhanced Lag Policy drop-down list	(Optional) Choose the Lag policy configured for the VMM domain of the device. This option is available, only if you have selected the **Device Type** (discussed in Step 5) as **Virtual**.

For an HA cluster, make sure that the cluster interfaces are mapped to the corresponding interfaces on both concrete devices in the cluster.

Step 11 Click **Finish**.

Creating a Layer 4 to Layer 7 Device Using the NX-OS-Style CLI

When you create a Layer 4 to Layer 7 device, you can connect to either a physical device or a virtual machine. When you connecting to a physical device, you specify the physical interface. When you connect to a virtual machine, you specify the VMM domain, the virtual machine, and the virtual interfaces.

Note When you configure a Layer 4 to Layer 7 device that is a load balancer, the context aware parameter is not used. The context aware parameter has a default value of `single context`, which can be ignored.

Before you begin

- You must have configured a tenant.

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.

```
tenant tenant_name
```

Example:

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

Step 3 Add a Layer 4 to Layer 7 device cluster.

```
l4l7 cluster name cluster_name type cluster_type vlan-domain domain_name
  [function function_type] [service service_type]
```

Parameter	Description
name	The name of the device cluster.
type	The type of the device cluster. Possible values are: • virtual • physical
vlan-domain	The domain to use for allocating the VLANs. The domain must be a VMM domain for virtual device and physical domain for physical device.

Parameter	Description
switching-mode (Cisco ACI Virtual Edge only)	(Optional) Choose one of the following modes: • **AVE**—Switches traffic through the Cisco ACI Virtual Edge. • **native**—Switches traffic through the VMware DVS. This is the default value.
function	(Optional) The function type. Possible values are: • go-to • go-through • L1 • L2
service	(Optional) The service type. This is used by the GUI to show the ADC- or firewall-specific icons and GUI. Possible values are: • ADC • FW • OTHERS

Example:

For a physical device, enter:

```
apic1(config-tenant)# l417 cluster name D1 type physical vlan-domain phys
  function go-through service ADC
```

For a virtual device, enter:

```
apic1(config-tenant)# l417 cluster name ADCCluster1 type virtual vlan-domain mininet
```

Step 4 Add one or more cluster devices in the device cluster.

```
cluster-device device_name [vcenter vcenter_name] [vm vm_name]
```

Parameter	Description
vcenter	(Only for a virtual device) The name of VCenter that hosts the virtual machine for the virtual device.
vm	(Only for a virtual device) The name of the virtual machine for the virtual device.

Example:

For a physical device, enter:

```
apic1(config-cluster)# cluster-device C1
apic1(config-cluster)# cluster-device C2
```

For a virtual device, enter:

```
apic1(config-cluster)# cluster-device C1 vcenter vcenter1 vm VM1
apic1(config-cluster)# cluster-device C2 vcenter vcenter1 vm VM2
```

Step 5 Add one or more cluster interfaces in the device cluster.

```
cluster-interface interface_name [vlan static_encap]
```

Parameter	Description
vlan	(Only for a physical device) The static encapsulation for the cluster interface. VLAN value must be between 1 to 4094.

Example:

For a physical device, enter:

```
apic1(config-cluster)# cluster-interface consumer vlan 1001
```

For a virtual device, enter:

```
apic1(config-cluster)# cluster-interface consumer
```

Step 6 Add one or more members in the cluster interface.

```
member device device_name device-interface interface_name
```

Parameter	Description
device	The name of the cluster device that must have been already added to this device cluster using **cluster-device** command.
device-interface	The name of the interface on the cluster device.

Example:

```
apic1(config-cluster-interface)# member device C1 device-interface 1.1
```

Step 7 Add an interface to a member.

```
interface {ethernet ethernet_port | port-channel port_channel_name [fex fex_ID] |
  vpc vpc_name [fex fex_ID]} leaf leaf_ID
```

If you want to add a vNIC instead of an interface, then skip this step.

Parameter	Description
ethernet	(Only for an Ethernet or FEX Ethernet interface) The Ethernet port on the leaf where the cluster device is connected to Cisco Application Centric Infrastructure (ACI) fabric. If you are adding a FEX Ethernet member, specify both the FEX ID and the FEX port in the following format: *FEX_ID/FEX_port* For example: 101/1/23 The FEX ID specifies where the cluster device is connected to Fabric extender.
port-channel	(Only for a port channel or FEX port channel interface) The port channel name where the cluster device is connected to ACI fabric.
vpc	(Only for a virtual port channel or FEX virtual port channel interface) The virtual port channel name where the cluster device is connected to ACI fabric.

Parameter	Description
fex	(Only for a port channel, FEX port channel, virtual port channel, or FEX virtual port channel interface) The FEX IDs in a space-separated list that are used to form the port channel or virtual port channel.
leaf	The leaf IDs in a space-separated list where the cluster device is connected.

Example:

For an Ethernet interface, enter:

```
apic1(config-member)# interface ethernet 1/23 leaf 101
apic1(config-member)# exit
```

For a FEX Ethernet interface, enter:

```
apic1(config-member)# interface ethernet 101/1/23 leaf 101
apic1(config-member)# exit
```

For a port channel interface, enter:

```
apic1(config-member)# interface port-channel pc1 leaf 101
apic1(config-member)# exit
```

For a FEX port channel interface, enter:

```
apic1(config-member)# interface port-channel pc1 leaf 101 fex 101
apic1(config-member)# exit
```

For a virtual port channel interface, enter:

```
apic1(config-member)# interface vpc vpc1 leaf 101 102
apic1(config-member)# exit
```

For a FEX virtual port channel interface, enter:

```
apic1(config-member)# interface vpc vpc1 leaf 101 102 fex 101 102
apic1(config-member)# exit
```

Step 8 Add a vNIC to a member.

```
vnic "vnic_name"
```

If you want to add an interface instead of a vNIC, then see the previous step.

Parameter	Description
vnic	The name of the vNIC adapter on the virtual machine for the cluster-device. Enclose the name in double quotes.

Example:

```
apic1(config-member)# vnic "Network adapter 2"
apic1(config-member)# exit
```

Step 9 If you are done creating the device, exit the configuration mode.

Example:

```
apic1(config-cluster-interface)# exit
apic1(config-cluster)# exit
```

```
apic1(config-tenant)# exit
apic1(config)# exit
```

Creating a High Availablity Cluster Using the NX-OS-Style CLI

This example procedure creates a high availability cluster using the NX-OS-style CLI.

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.

```
tenant tenant_name
```

Example:

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

Step 3 Create a cluster:

Example:

```
apic1(config-tenant)# l4l7 cluster name ifav108-asa type physical vlan-domain phyDom5 servicetype FW
```

Step 4 Add the cluster devices:

Example:

```
apic1(config-cluster)# cluster-device C1
apic1(config-cluster)# cluster-device C2
```

Step 5 Add a provider cluster interface:

Example:

```
apic1(config-cluster)# cluster-interface provider vlan 101
```

Step 6 Add member devices to the interface:

Example:

```
apic1(config-cluster-interface)# member device C1 device-interface Po1
apic1(config-member)# interface vpc VPCPolASA leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
apic1(config-cluster-interface)# member device C2 device-interface Po2
apic1(config-member)# interface vpc VPCPolASA-2 leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

Step 7 Add another provider cluster interface:

Example:

```
apic1(config-cluster)# cluster-interface provider vlan 102
```

Step 8 Add the same member devices from the first interface to this new interface:

Example:

```
apic1(config-cluster-interface)# member device C1 device-interface Po1
apic1(config-member)# interface vpc VPCPolASA leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
apic1(config-cluster-interface)# member device C2 device-interface Po2
apic1(config-member)# interface vpc VPCPolASA-2 leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

Step 9 Exit out of the cluster creation mode:

Example:

```
apic1(config-cluster)# exit
```

Creating a Virtual Device Using the NX-OS-Style CLI

This example procedure creates a virtual device using the NX-OS-style CLI.

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.

```
tenant tenant_name
```

Example:

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

Step 3 Create a cluster:

Example:

```
apic1(config-tenant)# l4l7 cluster name ifav108-citrix type virtual vlan-domain ACIVswitch servicetype
 ADC
```

Step 4 Add a cluster device:

Example:

```
apic1(config-cluster)# cluster-device D1 vcenter ifav108-vcenter vm NSVPX-ESX
```

Step 5 Add a consumer cluster interface:

Example:

```
apic1(config-cluster)# cluster-interface consumer
```

Step 6 Add a member device to the consumer interface:

Example:

```
apic1(config-cluster-interface)# member device D1 device-interface 1_1
apic1(config-member)# interface ethernet 1/45 leaf 102
ifav108-apic1(config-member)# vnic "Network adapter 2"
```

```
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

Step 7 Add a provider cluster interface:

Example:

```
apic1(config-cluster)# cluster-interface provider
```

Step 8 Add the same member device to the provider interface:

Example:

```
apic1(config-cluster-interface)# member device D1 device-interface 1_1
apic1(config-member)# interface ethernet 1/45 leaf 102
ifav108-apic1(config-member)# vnic "Network adapter 2"
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

Step 9 Exit out of the cluster creation mode:

Example:

```
apic1(config-cluster)# exit
```

Example XML for Creating a Logical Device

Example XML of Creating an LDevVip Object

The following example XML creates an `LDevVip` object:

```
<polUni>
    <fvTenant name="HA_Tenant1">
        <vnsLDevVip name="ADCCluster1" devtype="VIRTUAL" managed="no">
            <vnsRsALDevToDomP tDn="uni/vmmp-VMware/dom-mininet"/>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

For Cisco ACI Virtual Edge, the following example XML creates an `LDevVip` object associated to the Cisco ACI Virtual Edge VMM domain with `ave` as the switching mode:

```
<polUni>
    <fvTenant name="HA_Tenant1">
        <vnsLDevVip name="ADCCluster1" devtype="VIRTUAL" managed="no">
            <vnsRsALDevToDomP switchingMode="AVE" tDn="uni/vmmp-VMware/dom-mininet_ave"/>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

Example XML of Creating an AbsNode Object

The following example XML creates an `AbsNode` object:

```
<fvTenant name="HA_Tenant1">
    <vnsAbsGraph name="g1">
        <vnsAbsTermNodeProv name="Input1">
```

```
            <vnsAbsTermConn name="C1">
            </vnsAbsTermConn>
        </vnsAbsTermNodeProv>

        <!-- Node1 provides a service function -->
        <vnsAbsNode name="Node1" managed="no">
            <vnsAbsFuncConn name="outside" >
            </vnsAbsFuncConn>
            <vnsAbsFuncConn name="inside" >
            </vnsAbsFuncConn>
        </vnsAbsNode>

        <vnsAbsTermNodeCon name="Output1">
            <vnsAbsTermConn name="C6">
            </vnsAbsTermConn>
        </vnsAbsTermNodeCon>

        <vnsAbsConnection name="CON2" >
            <vnsRsAbsConnectionConns
              tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeCon-Output1/AbsTConn"/>
            <vnsRsAbsConnectionConns
              tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-outside"/>
        </vnsAbsConnection>

        <vnsAbsConnection name="CON1" >
            <vnsRsAbsConnectionConns
              tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-inside"/>
            <vnsRsAbsConnectionConns
              tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeProv-Input1/AbsTConn"/>
        </vnsAbsConnection>
    </vnsAbsGraph>
</fvTenant>
```

Example XML of Associating a Layer 4 to Layer 7 Service Endpoint Group with a Connector

The following example XML associates a Layer 4 to Layer 7 service endpoint group with a connector:

```
<fvTenant name="HA_Tenant1">
   <vnsLDevCtx ctrctNameOrLbl="any" descr="" dn="uni/tn-HA_Tenant1/ldevCtx-c-any-g-any-n-any"

      graphNameOrLbl="any" name="" nodeNameOrLbl="any">
        <vnsRsLDevCtxToLDev tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1"/>
        <vnsLIfCtx connNameOrLbl="inside" descr="" name="inside">
            <vnsRsLIfCtxToSvcEPg tDn="uni/tn-HA_Tenant1/ap-sap/SvcEPg-EPG1"/>
            <vnsRsLIfCtxToBD tDn="uni/tn-HA_Tenant1/BD-provBD1"/>
            <vnsRsLIfCtxToLIf tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1/lIf-inside"/>
        </vnsLIfCtx>
        <vnsLIfCtx connNameOrLbl="outside" descr="" name="outside">
            <vnsRsLIfCtxToSvcEPg tDn="uni/tn-HA_Tenant1/ap-sap/SvcEPg-EPG2"/>
            <vnsRsLIfCtxToBD tDn="uni/tn-HA_Tenant1/BD-consBD1"/>
            <vnsRsLIfCtxToLIf tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1/lIf-outside"/>
        </vnsLIfCtx>
    </vnsLDevCtx>
</fvTenant>
```

Example XML of Using Static Encapsulation with a Layer 4 to Layer 7 Service Endpoint Group

The following example XML uses static encapsulation with a Layer 4 to Layer 7 services endpoint group:

```
<polUni>
    <fvTenant name="HA_Tenant1">
        <fvAp name="sap">
            <vnsSvcEPg name="EPG1" encap="vlan-3510">
            </vnsSvcEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

Modifying a Device Using the GUI

After you create a device, you can modify the device.

Note To create a device or to add a device to an existing cluster, you must use the "Creating a Device" procedure.

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

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

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Devices** > ***device_name***.
The Work pane displays information about the device.

Step 4 You can change some of the parameters in the **General** section.

You can add interfaces or change the path for the existing interfaces in the **Device 1** section. To add an interface, click the + button. To change the path, double-click on the path you want to change.

Step 5 After you making any changes to the parameters, click **Submit**.

Enabling Trunking on a Layer 4 to Layer 7 Virtual ASA device Using the GUI

The following procedure enables trunking on a Layer 4 to Layer 7 virtual ASA device using the GUI.

Before you begin

- You must have configured a Layer 4 to Layer 7 virtual ASA device.

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

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

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Devices** > ***device_name***.

Step 4 In the Work pane, put a check in the **Trunking Port** check box.

Step 5 Click **Submit**.

Enabling Trunking on a Layer 4 to Layer 7 Virtual ASA device Using the REST APIs

The following procedure provides an example of enabling trunking on a Layer 4 to Layer 7 virtual ASA device using the REST APIs.

Before you begin

- You must have configured a Layer 4 to Layer 7 virtual ASA device.

Enable trunking on the Layer 4 to Layer 7 device named `InsiemeCluster`:

```
<polUni>
    <fvTenant name="tenant1">
        <vnsLDevVip name="InsiemeCluster" devtype="VIRTUAL" trunking="yes">
            ...
            ...
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

Using an Imported Device with the REST APIs

The following REST API uses an imported device:

```
<polUni>
  <fvTenant dn="uni/tn-tenant1" name="tenant1">
    <vnsLDevIf ldev="uni/tn-mgmt/lDevVip-ADCCluster1"/>
    <vnsLDevCtx ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any">
      <vnsRsLDevCtxToLDev tDn="uni/tn-tenant1/lDevIf-[uni/tn-mgmt/lDevVip-ADCCluster1]"/>
      <vnsLIfCtx connNameOrLbl="inside">
        <vnsRsLIfCtxToLIf
tDn="uni/tn-tenant1/lDevIf-[uni/tn-mgmt/lDevVip-ADCCluster1]/lDevIfLIf-inside"/>
        <fvSubnet ip="10.10.10.10/24"/>
        <vnsRsLIfCtxToBD tDn="uni/tn-tenant1/BD-tenant1BD1"/>
      </vnsLIfCtx>
      <vnsLIfCtx connNameOrLbl="outside">
        <vnsRsLIfCtxToLIf
tDn="uni/tn-tenant1/lDevIf-[uni/tn-mgmt/lDevVip-ADCCluster1]/lDevIfLIf-outside"/>
        <fvSubnet ip="70.70.70.70/24"/>
        <vnsRsLIfCtxToBD tDn="uni/tn-tenant1/BD-tenant1BD4"/>
      </vnsLIfCtx>
    </vnsLDevCtx>
  </fvTenant>
</polUni>
```

Importing a Device From Another Tenant Using the NX-OS-Style CLI

You can import a device from another tenant for a shared services scenario.

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.

```
tenant tenant_name
```

Example:

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

Step 3 Import the device.

```
l4l7 cluster import-from tenant_name device-cluster device_name
```

Parameter	Description
import-from	Name of the tenant from where to import the device.
device-cluster	Name of the device cluster to import from the specified tenant.

Example:

```
apic1(config-tenant)# l4l7 cluster import-from common device-cluster d1
apic1(config-import-from)# end
```

Verifying the Import of a Device Using the GUI

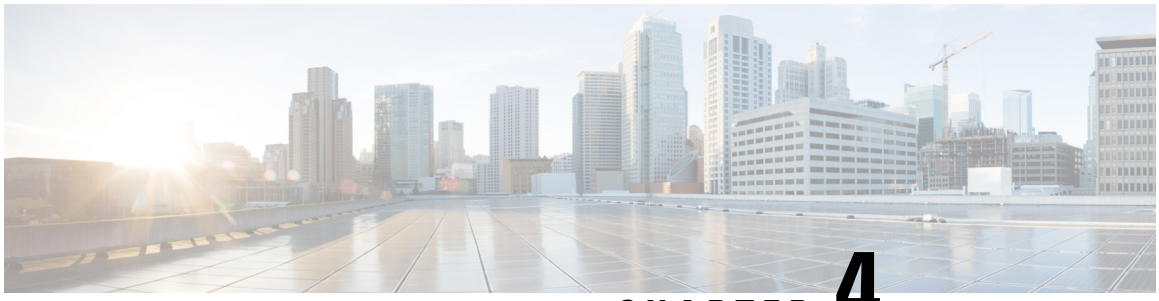
You can use the GUI to verify that a device was imported successfully.

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

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

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Imported Devices** > ***device_name***.

The device information appears in the **Work** pane.



CHAPTER 4

Service VM Orchestration

- Service VM Orchestration, on page 23
- Service VM Orchestration Guidelines and Limitations, on page 24
- Configuring Service VM Orchestration Using the Cisco APIC GUI, on page 24
- Configuring Service VM Orchestration Using the NX-OS Style CLI, on page 30
- Configuring Service VM Orchestration Using REST API, on page 30
- Troubleshooting Service VM Orchestration, on page 32

Service VM Orchestration

Service virtual machine (VM) orchestration is a policy-based feature that enables you to create and manage service VMs easily with Cisco Application Policy Infrastructure Controller (APIC). Service VM orchestration is a new feature for VMware vCenter environments in Cisco APIC 4.0(1).

Previously, you had to create a service VM in VMware vCenter, define the data center that it belonged to, and associate it with a data store. You also had to configure its management network settings and then attach it to Cisco APIC. However, service VM orchestration enables you to perform all these tasks in Cisco APIC.

Service VM orchestration streamlines the process of configuring the service VMs, also known as concrete devices (CDev). The CDevs are grouped into a device cluster, also known as a logical device (LDev). Configuration and policy that are applied to the LDev are applied to each CDev that it contains.

To use service VM orchestration, you create and upload a configuration file. You then configure a VM instantiation policy, create the Layer 4 to Layer 7 LDev, and then create CDevs associated with the LDev. Read and understand the section Service VM Orchestration Guidelines and Limitations, on page 24 before configuring service VM orchestration.

You can perform Service VM orchestration tasks using the Cisco APIC GUI, the NX-OS style CLI, or REST API. See the the following sections for instructions:

- Configuring Service VM Orchestration Using the Cisco APIC GUI, on page 24
- Configuring Service VM Orchestration Using the NX-OS Style CLI, on page 30
- Configuring Service VM Orchestration Using REST API, on page 30

Service VM Orchestration Guidelines and Limitations

Keep the following guidelines and limitations in mind when using service VM orchestration:

- Service VM orchestration is supported only for Cisco Adaptive Security Virtual Appliance (ASAv) and Palo Alto Networks devices.
- High-availability (HA) virtual machine (VM) deployment using service VM orchestration is supported only on shared storage. It is not supported on a local data store.
- Dynamic Host Configuration Protocol (DHCP) IP addressing is not supported for single or HA service VM deployments.
- Any port group or VM template created on VMware vCenter requires manual inventory sync on Cisco Application Policy Infrastructure Controller (APIC) before you use service VM orchestration. Check the configuration documentation on how to trigger inventory sync.
- Palo Alto deployment works only with the default username **admin** and the password **admin**.
- After a Palo Alto device is deployed, you see a `Script error: force config push is required` fault on Cisco APIC for 10 minutes. The message is due to an internal process running on the Palo Alto device; the fault will be cleared when the configuration is pushed successfully and the device becomes stable.
- Cisco APIC cannot reach a Cisco Adaptive Security Virtual Appliance (ASAv) device after deletion and redeployment. This issue occurs because the old MAC address is not cleared in the upstream switches. Clear the MAC entry of the IP address that is used for service VMs on the upstream switch and then redeploy the service VM using service VM orchestration.
- If you are cloning an existing policy, do not change a VM instantiation policy that is associated with a logical device before the cloning is completed.
- To deploy service VMs using service VM orchestration, enable additional VMware vCenter privileges. See the section "Custom User Account with Minimum VMware vCenter Privileges" in the chapter "Cisco ACI with VMware VDS Integration" in the *Cisco ACI Virtualization Guide*.

Configuring Service VM Orchestration Using the Cisco APIC GUI

You can perform several tasks in the Cisco Application Policy Infrastructure Controller (APIC) GUI to configure Service VM orchestration.

Creating a VM Instantiation Policy Using the Cisco APIC GUI

Creating a virtual machine (VM) instantiation file is the first task in the process of using service virtual machine (VM) orchestration to deploy and manage service VMs with the Cisco Application Policy Infrastructure Controller. The policy is created for a device cluster or logical device (LDev) and is then applied to concrete devices (CDev) that belong to the LDev.

Step 1 Log in to Cisco APIC.

Step 2 Go to **Tenants** > ***tenant*** > **Policies** > **VMM** > **VM Instantiation Policies**.

Step 3 In the upper-right corner of the work pane, click the icon of a hammer and wrench, and then choose **Create VM Instantiation Policy**.

Step 4 In the **Create VM Instantiation Policy** dialog box, complete the following steps:

a) In the **Name** field, enter the name of the policy.

b) From the **Controller** drop-down list, choose the controller.

c) From the **VM Template** drop-down list, choose the template for the service VM that you want to create.

The drop-down list shows you VM templates associated with the controller.

Note If you do not see the VM template created on VMware vCenter, complete the following steps:

1. Click the blue icon next to the controller drop-down list.
2. In the **Controller Instance** dialog box, click the wrench-and-hammer icon on the right, and then click **Trigger Inventory Sync**, and then click **Yes** to trigger the sync.
3. Close the **Controller Instance** dialog box to return to the **Create VM Instantiation Policy** dialog box.

d) From the **Host Name** drop-down list, choose the host where you want to deploy the service VM.

You can choose a VMware vSphere Distributed Resource Scheduler (DRS) cluster or an individual host.

e) From the **Data Store** drop-down list, choose the data store where you want to put the VM disk.

f) Click **Submit**.
The work pane shows the VM instantiation policies.

Creating a Layer 4 to Layer 7 services device and Associating the Device with the VM Instantiation Policy Using the GUI

In this procedure, you create a Layer 4 to Layer 7 services device and associate it with the virtual machine (VM) instantiation policy that you created earlier.

When you create a Layer 4 to Layer 7 services device, you can connect to either a physical device or a virtual machine. The fields are slightly different depending on the type to which you are connecting. When you connect to a physical device, you specify the physical interface. When you connect to a virtual machine, you specify the VMM domain, the virtual machine, and the virtual interfaces. Also, you can select an unknown model, which allows you to configure the connections manually.

When you create a Layer 4 to Layer 7 services device to be associated with the VM instantiation policy, you also specify the policy and create the new service VM.

Note When you configure a Layer 4 to Layer 7 services device that is a load balancer, the context aware parameter is not used. The context aware parameter has a default value of single context, which can be ignored.

Before you begin

- You must have configured a tenant.
- You must have created a VM instantiation policy. See the section Creating a VM Instantiation Policy Using the Cisco APIC GUI, on page 24.

Step 1 Log in to Cisco Application Policy Infrastructure Controller (APIC).

Step 2 Go to **Tenants** > ***tenant*** > **Services** > **L4-L7** > **Devices**.

Step 3 Right-click **Devices** and then choose **Create L4-L7 Devices**.

Alternatively, in the upper right of the work pane you can click the actions icon (crossed hammer and wrench) and then choose **Create L4-L7 Devices**.

Step 4 In the **Create L4-L7 Devices** dialog box, in the **General** section, complete the following fields:

Name	Description
Name	Enter a name for the Layer 4 to Layer 7 services device.
Service Type	Choose a service type from the drop-down list. You can choose one of the following service types: • **ADC** (Application Delivery Controller). **ADC** is the default service type. • **Firewall**: Choose routed or transparent deployment mode. • **Other**: any other mode. **Note** For a policy-based redirect configuration, choose **Firewall** or **ADC** as the service type.
Device Type	Choose **Virtual** (virtual Layer 4 to Layer 7 services device).
VMM Domain	Choose a VMM domain from the drop-down list.
VM Instantiation Policy	From the drop-down list, choose the VM instantiation policy that you created earlier. Choosing the policy associates it with the new Layer 4 to Layer 7 services device. It also helps creating the VM automatically on VMware vCenter.
Promiscuous Mode	Check the checkbox to enable promiscuous mode on Cisco Application Centric Infrastructure (ACI)-managed port groups that are generated after deploying a service graph. Enabling promiscuous mode allows all the traffic in a port group to reach a VM attached to a promiscuous port.

Name	Description
Context Aware	Choose **Single**, the default, or **Multiple**. If you choose **Single**, the device cluster cannot be shared across multiple tenants of a given type that are hosted on the provider network. You must give the device cluster to a specific tenant for a given user. If you choose **Multiple**, the device cluster can be shared across multiple tenants of a given type that you are hosting on the provider network. For example, there could be two hosting companies that share the same device.
Function Type	You can choose: • **GoThrough** (transparent mode) • **GoTo** (routed mode)

Step 5 In the **Devices** section, click the plus icon.

Step 6 In the **Create Device STEP 1 > Device** dialog box, complete the following fields to configure a concrete device (CDev) and associate it with the Layer 4 to Layer 7 services device:

Name	Description
Gateway IP	Enter the gateway IP address of the new service VM.
Subnet Mask	Enter the subnet mask for the new service VM.
Management vNIC	Choose the management vNIC for the new service VM from the drop-down list.
VM	Enter the VM name for the new service VM to appear in VMware vCenter.

Name	Description
Host (Optional)	From the drop-down list, choose the host for the new service VM. If you do not choose a host, the host that is chosen in VM instantiation policy will be used. For policy-based redirect (PBR) and direct server return (DSR) functionality, selection of particular host is important based on topology. In that case, choose the correct host. For DSR and PDR, compute VMs and service VMs cannot reside on the same top-of-rack (TOR) switch pair. So you need to choose the host for deploying service VMs for PBR or DSR topology. Oherwise, the feature could deploy the service VMs on the same host as the compute VMs. For devices to be connected on Cisco Application Centric Infrastructure (ACI) Virtual Edge, you cannot deploy high-availability Layer 4 to Layer 7 services devices on same host. Therefore, choose different hosts for primary and secondary VMs.
Port Group Name (Optional)	From the drop-down list, choose the port group for the new service VM to be deployed. If you do not choose one, the port group that is used in the VM template will be used.
HA EPG (Optional)	From the drop-down list, choose the high-availability (HA) endpoint group (EPG) or vSwitch or distributed virtual switch (DVS) port group for HA communication for the new service VM.
HA Network Adapter (Optional)	Choose an HA network adapter for the new service VM from the drop-down list.
Username	Enter the username for the new service VM.
Password	Enter the password for the new service VM.
Confirm Password	Re-enter the password.

Step 7 Click **Next**.

Step 8 In the **Create Device STEP 2 > Interfaces** dialog box, in the **Interfaces** section, click the plus icon.

Step 9 Complete the following fields in the dialog box to configure the interface for the CDev:

Name	Description
Name	Choose the name of the Layer 4 to Layer 7 services device interface from the drop-down list.
VNIC (Virtual device type only)	Choose the name of the VM network adapter from the drop-down list.

Name	Description
Path (Optional if the Layer 4 to Layer 7 services device is a virtual device)	Choose a port, port channel (PC), or virtual port channel (VPC) that the interface will connect to.

Step 10 In the **Interfaces** section, click the plus icon again and configure another interface.

Step 11 Click **Update**.

Step 12

Step 13 To add extra service VMs to the Layer 4 to Layer 7 services device, repeat Step 8 through Step 13.

Step 14 If you have multiple service VMs, in the **Create Device STEP 1 > Device** dialog box, in the **Cluster** section, complete the following fields for each device:

For an HA cluster, make sure that the cluster interfaces are mapped to the corresponding interfaces on both concrete devices in the cluster.

Name	Description

Cluster Interfaces area	Complete the following fields to configure outside connectivity for the Layer 4 to Layer 7 services device:
	• From the **Type** drop-down list, choose a cluster interface type. The type can be:
	  • **failover_link**
	  • **utility**
	  • **consumer**
	  • **provider**
	  • **mgmt**
	  • **cluster_ctrl_lk**
	  • **failover-lan**
	  • **consumer and provider**
	• From the **Name** drop-down list, choose the cluster interface name.
	• From the **Concrete Interfaces** drop-down list, choose the associated concrete interfaces.

Step 15 Click **Finish**.

What to do next

You can view creation of the new service VM in the VMware vCenter under **Recent Tasks**. It can take a while for it to appear.

Configuring Service VM Orchestration Using the NX-OS Style CLI

You can use the NX-OS style CLI to create the virtual machine (VM) instantiation policy and the Layer 4 to Layer 7 concrete device and map the device to the instantiation policy. You can then map the internal and external interfaces to the VM network adapter.

Step 1 Create the VM instantiation policy.

Example:

```
APIC1(config-tenant)# inst-pol VMPolName VMMname VcentercontrollerName VMtemplateName ClusterName
datastorename
```

Step 2 Create the Layer 4 to Layer 7 concrete device and associate it with the VM instantiation policy.

Example:

```
APIC1(config)# tenant T0
APIC1(config-tenant)# l4l7 cluster name ASA-Single type virtual vlan-domain ASAVMM switching-mode
AVE vm-instantiation-policy ASA-Template-Pol  service FW function go-to context single trunking
disable
```

Step 3 Map the internal and external interfaces to the VM network adapter.

Example:

```
APIC1(config-cluster)# cluster-interface external
APIC1(config-cluster-interface)# member device ASA-Cdev device-interface GigabitEthernet0/0
APIC1(config-member)# vnic "Network adapter 2"
APIC1(config-member)# exit
APIC1(config-cluster)# cluster-interface internal
APIC1(config-cluster-interface)# member device ASA-Cdev device-interface GigabitEthernet0/1
APIC1(config-member)# vnic "Network adapter 3"
APIC1(config-member)# exit
APIC1(config-cluster-interface)# exit
APIC1(config-cluster)#
```

Configuring Service VM Orchestration Using REST API

You can configure service VM orchestration using the REST API.

Configure service VM orchestration.

Example:

```
<vnsLDevVip contextAware="single-Context" devtype="VIRTUAL"
  dn="uni/tn-T0/lDevVip-NEW-HA-LDEV-20" funcType="GoTo" isCopy="no" mode="legacy-Mode"
  name="NEW-HA-LDEV-20" promMode="no" svcType="FW" trunking="no">
    <vnsLIf encap="unknown" name="client">
        <vnsRsMetaIf isConAndProv="no"
          tDn="uni/infra/mDev-CISCO-ASA-1.3/mIfLbl-external"/>
        <vnsRsCIfAttN
```

```
      tDn="uni/tn-T0/lDevVip-NEW-HA-LDEV-20/cDev-CDEV-HA-S1-NEW/cIf-[GigabitEthernet0/0]"/>
    <vnsRsCIfAttN
      tDn="uni/tn-T0/lDevVip-NEW-HA-LDEV-20/cDev-CDEV-HA-P1-NEW/cIf-[GigabitEthernet0/0]"/>
</vnsLIf>
<vnsLIf encap="unknown" name="server">
    <vnsRsMetaIf isConAndProv="no" tDn="uni/infra/mDev-CISCO-ASA-1.3/mIfLbl-internal"/>
    <vnsRsCIfAttN
      tDn="uni/tn-T0/lDevVip-NEW-HA-LDEV-20/cDev-CDEV-HA-S1-NEW/cIf-[GigabitEthernet0/1]"/>
    <vnsRsCIfAttN
      tDn="uni/tn-T0/lDevVip-NEW-HA-LDEV-20/cDev-CDEV-HA-P1-NEW/cIf-[GigabitEthernet0/1]"/>
</vnsLIf>
<vnsRsLDevVipToInstPol tDn="uni/tn-T0/svcCont/instPol-HA-POL"/>
<vnsRsALDevToDomP switchingMode="AVE" tDn="uni/vmmp-VMware/dom-mininet"/>
<vnsCDev cloneCount="0" host="10.197.146.188" isCloneOperation="no" isTemplate="no"
  name="CDEV-HA-S1-NEW" vcenterName="orionin103-vcenter1" vmName="ASA-S1-VM-20">
    <vnsHAPortGroup portGroupName="10.197.146.188 | VLAN2500-172-25"
      vnicName="Network adapter 10"/>
    <vnsDevFolder key="FailoverConfig" name="FailoverConfig">
        <vnsDevParam key="lan_unit" name="lan_unit" value="secondary"/>
        <vnsDevParam key="failover" name="failover" value="enable"/>
        <vnsDevFolder key="mgmt_standby_ip" name="mgmt_standby_ip">
            <vnsDevParam key="standby_ip" name="standby_ip" value="10.197.146.178"/>
        </vnsDevFolder>
        <vnsDevFolder key="polltime" name="polltime">
            <vnsDevParam key="interval_value" name="interval_value" value="1"/>
            <vnsDevParam key="interval_unit" name="interval_unit" value="second"/>
            <vnsDevParam key="holdtime_value" name="holdtime_value" value="3"/>
        </vnsDevFolder>
        <vnsDevFolder key="failover_link_interface" name="failover_link_interface">
            <vnsDevParam key="use_lan" name="use_lan" value="fover"/>
            <vnsDevParam key="interface_name" name="interface_name" value="fover"/>
            <vnsDevParam key="interface" name="interface" value="GigabitEthernet0/8"/>
        </vnsDevFolder>
        <vnsDevFolder key="failover_ip" name="failover_ip">
            <vnsDevParam key="interface_name" name="interface_name" value="fover"/>
            <vnsDevParam key="active_ip" name="active_ip" value="172.25.0.178"/>
            <vnsDevParam key="netmask" name="netmask" value="255.255.0.0"/>
            <vnsDevParam key="standby_ip" name="standby_ip" value="172.25.0.179"/>
        </vnsDevFolder>
        <vnsDevFolder key="failover_lan_interface" name="failover_lan_interface">
            <vnsDevParam key="interface_name" name="interface_name" value="fover"/>
            <vnsDevParam key="interface" name="interface" value="GigabitEthernet0/8"/>
        </vnsDevFolder>
    </vnsDevFolder>
    <vnsCIf name="GigabitEthernet0/1" vnicName="Network adapter 3"/>
    <vnsCIf name="GigabitEthernet0/0" vnicName="Network adapter 2"/>
</vnsCDev>
<vnsCDev cloneCount="0" host="10.197.146.187" isCloneOperation="no" isTemplate="no"
  name="CDEV-HA-P1-NEW" vcenterName="orionin103-vcenter1" vmName="ASA-P1-VM-20">
    <vnsHAPortGroup portGroupName="10.197.146.187 | VLAN2500-172-25"
      vnicName="Network adapter 10"/>
    <vnsDevFolder key="FailoverConfig" name="FailoverConfig">
        <vnsDevParam key="lan_unit" name="lan_unit" value="primary"/>
        <vnsDevParam key="failover" name="failover" value="enable"/>
        <vnsDevFolder key="failover_ip" name="failover_ip">
            <vnsDevParam key="interface_name" name="interface_name" value="fover"/>
            <vnsDevParam key="standby_ip" name="standby_ip" value="172.25.0.179"/>
            <vnsDevParam key="netmask" name="netmask" value="255.255.0.0"/>
            <vnsDevParam key="active_ip" name="active_ip" value="172.25.0.178"/>
        </vnsDevFolder>
        <vnsDevFolder key="failover_lan_interface" name="failover_lan_interface">
            <vnsDevParam key="interface_name" name="interface_name" value="fover"/>
            <vnsDevParam key="interface" name="interface" value="GigabitEthernet0/8"/>
        </vnsDevFolder>
```

```
            <vnsDevFolder key="mgmt_standby_ip" name="mgmt_standby_ip">
                <vnsDevParam key="standby_ip" name="standby_ip" value="10.197.146.179"/>
            </vnsDevFolder>
            <vnsDevFolder key="failover_link_interface" name="failover_link_interface">
                <vnsDevParam key="interface_name" name="interface_name" value="fover"/>
                <vnsDevParam key="use_lan" name="use_lan" value="fover"/>
                <vnsDevParam key="interface" name="interface" value="GigabitEthernet0/8"/>
            </vnsDevFolder>
            <vnsDevFolder key="polltime" name="polltime">
                <vnsDevParam key="holdtime_value" name="holdtime_value" value="3"/>
                <vnsDevParam key="interval_unit" name="interval_unit" value="second"/>
                <vnsDevParam key="interval_value" name="interval_value" value="1"/>
            </vnsDevFolder>
        </vnsDevFolder>
        <vnsCIf name="GigabitEthernet0/1" vnicName="Network adapter 3"/>
        <vnsCIf name="GigabitEthernet0/0" vnicName="Network adapter 2"/>
    </vnsCDev>
    <vnsRsMDevAtt tDn="uni/infra/mDev-CISCO-ASA-1.3"/>
</vnsLDevVip>
```

Troubleshooting Service VM Orchestration

This section contains known issues and limitations with service VM orchestration and instructions for troubleshooting issues if they occur.

Service VM Templates Not Seen in VM Instantiation Policy

Complete the following steps if you do not see the service VM templates that were created on VMware vCenter in the VM Instantiation policy.

Step 1 Check Visore using **vnsInstPol** and look for `vmTemplate`.

If there is no value for **vnsInstPol** field or if the value is null, go to the next step.

Step 2 Trigger an inventory synchronization:

a) In Cisco Application Policy Infrastructure Controller (APIC), go to **Virtual Networking** > **Inventory** and expand the **VMM Domains** and **VMware** folders.

b) Click the VMM domain.

c) In the central pane, double-click the controller.

d) In the **VMM Controller** dialog box, from the hammer-and-wrench drop-down list, choose **Trigger Inventory Sync** and when prompted, click **Yes**.

Step 3 Check the virtual machine (VM) instantiation policy: Choose the controller that is mapped to the VMM domain, and see if the VM template is present.

Port Groups Created in VMware vCenter Do Not Appear for CDev

Complete the following steps if port groups created in VMware vCenter do not appear for a concrete device (CDev).

Step 1 Trigger an inventory synchronization:

a) In Cisco Application Policy Infrastructure Controller (APIC), go to **Virtual Networking** > **Inventory** and expand the **VMM Domains** and **VMware** folders.
b) Click the VMM domain.
c) In the central pane, double-click the controller.
d) In the **VMM Controller** dialog box, from the hammer-and-wrench drop-down list, choose **Trigger Inventory Sync** and when prompted, click **Yes**.

Step 2 Check if the port group appears:

a) Go to **Tenants** > ***tenant*** > **Services** > **L4-L7** > **Devices** > ***device*** and then click the device.

Step 3 In the **Concrete Device** work pane, check the **Port Group Name** drop-down list to see if a port group is present.

Unable to Reach Service VM IP Address

Complete the following steps if you cannot reach the service virtual machine (VM) IP address after deploying the service virtual machine (VM).

Step 1 In Cisco Application Policy Infrastructure Controller (APIC), check the service VM connectivity.

Cisco APIC cannot reach a Cisco Adaptive Security Virtual Appliance (ASAv) device after deletion and redeployment. This issue occurs because the old MAC address is not cleared in the upstream switches. Clear the MAC entry of the IP address that is used for service VMs and then redeploy the service VM.

Step 2 If device management uses vSwitch port groups, check all intermediate switches and devices between Cisco APIC and the VMware vCenter to see if VLANs and routes are present.

Cisco APIC should be able to ping the device IP address if the service VM was deployed successfully.

Step 3 Make sure that the correct port group or EPG is chosen for the management interface for the concrete device (CDev).

Step 4 Check connectivity to make sure that the service VM can reach the upstream gateway.

Device State Shows Init

Complete the following steps if the device state shows init.

Step 1 From the NX-OS style CLI, ping the service device to verify reachability.

Step 2 Verify that the login credentials to the service device match the username and password that are supplied in the device configuration.

Step 3 Verify that the service device's virtual IP address and port are open.

Step 4 Verify that the username and password are correct in the Cisco Application Policy Infrastructure Controller (APIC) configuration.

LIF Configuration Is Invalid

Complete the following steps if you see an F0772 fault saying that the logical interface (LIF) configuration is invalid because of `lif-invalid-Clf` in the logical device.

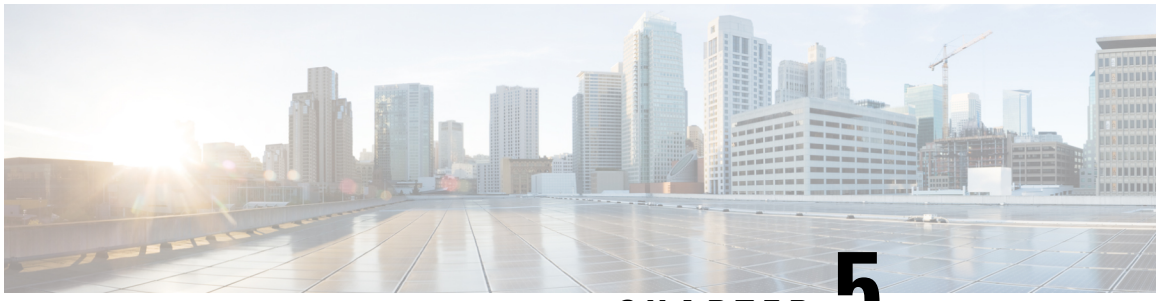
Step 1 Determine what items are called the LIF and the concrete interface (ClF).

With this particular fault, the LIF is the element that is not rendering properly. This is where the Function Node maps the LIF to the actual, or concrete, interface to form a relationship.

F0772 means one of the following problems:

- The LIF is not created.
- The LIF is not mapped to the correct concrete interface.

Step 2 For other Layer 4 to Layer 7 device state problems, see the troubleshooting content in this document.



CHAPTER 5

Selecting a Layer 4 to Layer 7 Device to Render a Graph

- About Device Selection Policies, on page 35
- Creating a Device Selection Policy Using the GUI, on page 35
- Configuring a Device Selection Policy Using REST APIs, on page 38

About Device Selection Policies

A device can be selected based on a contract name, a graph name, or the function node name inside the graph. After you create a device, you can create a device context, which provides a selection criteria policy for a device.

A device selection policy (also known as a device context) specifies the policy for selecting a device for a service graph template. This allows an administrator to have multiple device and then be able to use them for different service graph templates. For example, an administrator can have a device that has high-performance ADC appliances and another device that has lower-performance ADC appliances. Using two different device selection policies, one for the high-performance ADC device and the other for the low-performance ADC device, the administrator can select the high-performance ADC device for the applications that require higher performance and select the low-performance ADC devices for the applications that require lower performance.

Creating a Device Selection Policy Using the GUI

If you did not use the **Apply L4-L7 Service Graph Template To EPGs** wizard to apply the service graph template, you might need to configure a device selection policy (also known as a logical device context). The device selection policy instructs Cisco Application Centric Infrastructure (ACI) about which firewall or load balancer device to use to render a graph.

If you used the **Apply L4-L7 Service Graph Template To EPGs** wizard to apply the service graph template, then a device selection policy was configured automatically and you do not need to configure one manually.

The context name in device selection policy needs to be configured if the device cluster interface is used for intra-vrf and inter-vrf contract. The context name shall be identical for the same device shared by different deployed graph instances.

For example, when you have contract1 that is for intra-vrf and contract2 that is for inter-vrf traffic, if both the contracts have service graph, and you use same device cluster interface, you should configure same context name in device selection policy.

Note When using the NX-OS-style CLI, the device selection policy is configured automatically; there are no equivalent NX-OS-style CLI commands.

If you add copy devices to a service graph template that is already deployed, you must create a device selection policy to use for copy services.

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

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

Step 3 In the Navigation pane, choose **Tenant** *tenant_name* > **Services** > **L4-L7** > **Devices Selection Policies**.

Step 4 In the Work pane, choose **Actions** > **Create Logical Device Context**.

Step 5 In the **Create Logical Device Context** dialog box, fill in the fields as required, except as specified below:

a) In the **Contract Name** drop-down list, choose the contract for the device selection policy. If you do not want to use the contract name as part of the criteria for using a device, choose **any**.

b) In the **Graph Name** drop-down list, choose the graph for the device selection policy. If you do not want to use the graph name as part of the criteria for using a device, choose **any**.

c) In the **Node Name** drop-down list, choose the node for the device selection policy. If you do not want to use the node name as part of the criteria for using a device, choose **any**.

Step 6 In the **Cluster Interface Contexts** section, click **+** to add a cluster interface context.

Step 7 In the **Create A Cluster Interface Context** dialog, configure the following properties:

Property	Description
Connector Name	The connector name or label for the logical interface context. The default is **Any**.
Cluster Interface	The unique name of the target interface. This field is required.
Associated Network	Choose the associated network type. The possible choices are: • **Bridge Domain**: A service EPG will be newly created for the interface during the service graph deployment. • **L3Out**: The existing L3Out EPG is used for the interface.
Bridge Domain	Choose the bridge domain for the associated network of the target interface. This drop-down list only displays if you chose **Bridge Domain** for **Associated Network**. For Anycast, the bridge domain should be the same as that used for the node.

Property	Description
L3Out	Choose the L3Out EPG for the associated network of the target interface. This drop-down list only displays if you chose **L3Out** for **Associated Network**.
L3 Destination (VIP)	Indicates whether this logical interface is terminating the Layer 3 traffic in the service chain. The default for this parameter is enabled (checked). However, this setting is not considered if a policy-based redirect policy is configured on the logical interface context. **Note** For multi-node PBR, if this logical interface is a consumer construct on a load balancer terminated on a virtual IP address external network, put a check in this box and remove any association to a redirect policy in the next field (**L4-L7 Policy Based Redirect**). If this logical interface is a provider construct on a load balancer and it is performing SNAT, then put a check in this box and remove any association to a redirect policy in the next field (**L4-L7 Policy Based Redirect**).
L4-L7 Policy Based Redirect	Optional. Choose the policy-based redirect policy or **Create L4-L7 Policy Based Redirect**. **Note** For multi-node PBR, if this logical interface is a consumer construct on a load balancer terminated on a virtual IP address external network, remove this association to a redirect policy (if entered) and put a check in the **L3 Destination (VIP)** box.
L4-L7 Service EPG Policy	Choose to include or exclude the service EPG for the interface in the preferred group. By default, the service EPG is excluded.
Custom QoS Policy	Optional. Choose a Custom QoS Policy, the default policy, or **Create Custom QoS Policy**. This drop-down list only displays if you chose **Bridge Domain** for **Associated Network**.

Property	Description
Preferred Contract Group	The preferred group policy enforcement type. Valid types: • **Include**: EPGs or interfaces configured with this policy option are included in the subgroup and can communicate with others in the subgroup without a contract. • **Exclude**: EPGs or interfaces configured with this policy option are not included in the subgroup and cannot communicate with others in the subgroup without a contract.
Permit Logging	Put a check in the box to enable permit logging for the interface context. The default is disabled.
Subnets	Click + to add a subnet. Configure the gateway address, the network visibility of the subnet (scope), primary IP address (preferred subnet), and the subnet control state.
Virtual IP Addresses	Click + to add a Virtual IP Address (VIP) if this subnet is used for a Layer 3 virtual destination (**L3 Destination (VIP)** has a check in the box).

Step 8 Click **OK**.

Step 9 Click **Submit**.

Configuring a Device Selection Policy Using REST APIs

You can use the REST APIs to configure a device selection policy.

Creating a Device Selection Policy Using the REST API

The following REST API creates a device selection policy:

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vnsLDevCtx ctrctNameOrLbl="webCtrct" graphNameOrLbl="G1" nodeNameOrLbl="Node1">
            <vnsRsLDevCtxToLDev tDn="uni/tn-acme/lDevVip-ADCCluster1"/>

             <!-- The connector name C4, C5, etc.. should match the
                  Function connector name used in the service graph template -->

             <vnsLIfCtx connNameOrLbl="C4">
                <vnsRsLIfCtxToLIf tDn="uni/tn-acme/lDevVip-ADCCluster1/LIf-ext"/>
            </vnsLIfCtx>
            <vnsLIfCtx connNameOrLbl="C5">
                <vnsRsLIfCtxToLIf tDn="uni/tn-acme/lDevVip-ADCCluster1/LIf-int"/>
            </vnsLIfCtx>
```

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

Adding a Logical Interface in a Device Using the REST APIs

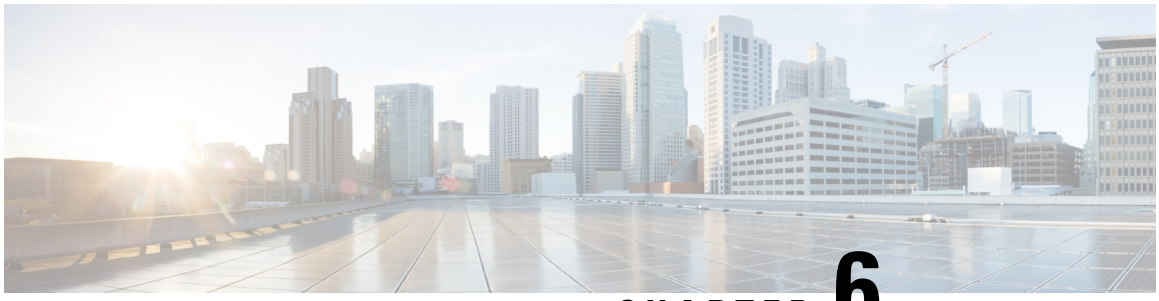
The following REST API adds a logical interface in a device:

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vnsLDevVip name="ADCCluster1">

            <!-- The LIF name defined here (such as e.g., ext, or int) should match the
                 vnsRsLIfCtxToLIf 'tDn' defined in LifCtx  -->

            <vnsLIf name="ext">

                <vnsRsMetaIf tDn="uni/infra/mDev-Acme-ADC-1.0/mIfLbl-outside"/>
                <vnsRsCIfAtt tDn="uni/tn-acme/lDevVip-ADCCluster1/cDev-ADC1/cIf-ext"/>
            </vnsLIf>
            <vnsLIf name="int">
                <vnsRsMetaIf tDn="uni/infra/mDev-Acme-ADC-1.0/mIfLbl-inside"/>
                <vnsRsCIfAtt tDn="uni/tn-acme/lDevVip-ADCCluster1/cDev-ADC1/cIf-int"/>
            </vnsLIf>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```



CHAPTER 6

Configuring a Service Graph

- About Service Graphs, on page 41
- About Function Nodes, on page 43
- About Function Node Connectors, on page 43
- About Service Graph Connections, on page 44
- About Terminal Nodes, on page 44
- Guidelines and Limitations for Configuring Service Graph, on page 44
- Configuring a Service Graph Template Using the GUI, on page 44
- Configuring a Service Graph Template Using the REST APIs, on page 45
- Applying a Service Graph Template to Endpoint Groups Using the GUI, on page 46
- Applying a Service Graph Template to an Endpoint Security Group Using the GUI, on page 47
- Applying a Service Graph Template with a Contract Using the NX-OS-Style CLI, on page 48

About Service Graphs

The Cisco Application Centric Infrastructure (ACI) treats services as an integral part of an application. Any services that are required are treated as a service graph that is instantiated on the Cisco ACI fabric from the Cisco Application Policy Infrastructure Controller (APIC). Users define the service for the application, while service graphs identify the set of network or service functions that are needed by the application.

A service graph represents the network using the following elements:

- Function node: A function node represents a function that is applied to the traffic, such as a transform (SSL termination, VPN gateway), filter (firewalls), or terminal (intrusion detection systems). A function within the service graph might require one or more parameters and have one or more connectors.
- Terminal node: A terminal node enables input and output from the service graph.
- Connector: A connector enables input and output from a node.
- Connection: A connection determines how traffic is forwarded through the network.

After the graph is configured in the Cisco APIC, the Cisco APIC automatically configures the services according to the service function requirements that are specified in the service graph. The Cisco APIC also automatically configures the network according to the needs of the service function that is specified in the service graph, which does not require any change in the service device.

A service graph is represented as two or more tiers of an application with the appropriate service function inserted between.

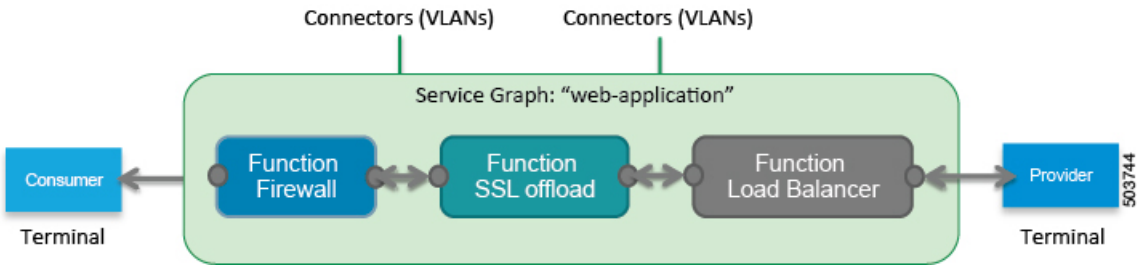
A service appliance (device) performs a service function within the graph. One or more service appliances might be required to render the services required by a graph. One or more service functions can be performed by a single-service device.

Service graphs and service functions have the following characteristics:

- Traffic sent or received by an endpoint group can be filtered based on a policy, and a subset of the traffic can be redirected to different edges in the graph.
- Service graph edges are directional.
- Taps (hardware-based packet copy service) can be attached to different points in the service graph.
- Logical functions can be rendered on the appropriate (physical or virtual) device, based on the policy.
- The service graph supports splits and joins of edges, and it does not restrict the administrator to linear service chains.
- Traffic can be reclassified again in the network after a service appliance emits it.
- Logical service functions can be scaled up or down or can be deployed in a cluster mode or 1:1 active-standby high-availability mode, depending on the requirements.

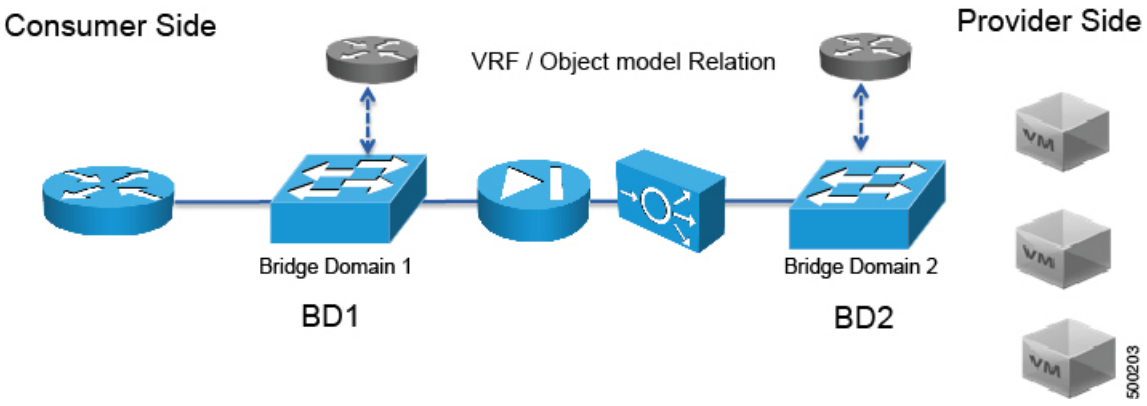
The following figure provides an example of a service graph deployment:

Figure 1: Example Service Graph Deployment



Deploying a service graph requires bridge domains and VRF instances, as shown in the following figure:

Figure 2: Bridge Domains and VRF instances of a Service Graph



Note If you have some of the legs of a service graph that are attached to endpoint groups in other tenants, when you use the **Remove Related Objects of Graph Template** function in the GUI, the Cisco APIC does not remove contracts that were imported from tenants other than where the service graph is located. The Cisco APIC also does not clean endpoint group contracts that are located in a different tenant than the service graph. You must manually remove these objects that are in different tenants.

About Function Nodes

A function node represents a single service function. A function node has function node connectors, which represent the network requirement of a service function.

The Cisco Application Policy Infrastructure Controller (APIC) only allocates the network resources and programs the VLAN/VXLAN on fabric side.

The following settings are not needed:

- `MFunc` relation
- Information about the supported function type (go-through, go-to)

The Cisco APIC needs to know the network information (LIF, CIF) for the function node. This information is needed so that the Cisco APIC can program the network appropriately on the leaf switch, and the Cisco APIC can also use this information for troubleshooting wizard purposes.

The following settings are still needed:

- `LDevCtx` to enable the selection of `LDevVip` during graph instantiation
- `LIfCtx` to enable the selection of `LIf` during graph instantiation
- Bridge domain in `LIfCtx`
- Route peering in `LIfCtx`
- Subnet in `LIfCtx`

Note For a Cisco ACI Multi-Site configuration, up to 2 nodes can be deployed in a service graph. For a non-Cisco ACI Multi-Site configuration, up to 5 nodes can be deployed in a service graph.

About Function Node Connectors

A function node connector connects a function node to the service graph and is associated with the appropriate bridge domain and connections based on the graph's connector's subset. Each connector is associated with a VLAN or Virtual Extensible LAN (VXLAN). Each side of a connector is treated as an endpoint group (EPG), and whitelists are downloaded to the switch to enable communication between the two function nodes.

About Service Graph Connections

A service graph connection connects one function node to another function node.

About Terminal Nodes

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

Guidelines and Limitations for Configuring Service Graph

The following are guidelines and limitations for configuring Service Graph.

- A service-graph related configuration such as
  - A bridge domain (if used with a service graph) and service graph template should not contain the string “C-“ as part of its name.
  - A logical device should not contain the string “N-“ as part of its name.

Configuring a Service Graph Template Using the GUI

A service graph template is a sequence of Layer 4 to Layer 7 services functions, Layer 4 to Layer 7 services devices, or copy devices and their associated configuration. The service graph template must be associated with a contract to be "rendered"—or configured—on the Layer 4 to Layer 7 services device or copy device, and on the fabric.

Before you begin

You must have configured a tenant.

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

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

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Service Graph Templates**.

Step 4 In the Navigation pane, right-click **Service Graph Templates** and choose **Create a L4-L7 Service Graph Template**.

The **Create L4-L7 Service Graph Template** dialog box appears.

Step 5 If necessary, create one or more Layer 4 to Layer 7 services devices or copy devices:

a) Click the drop-down arrow in the **Device Clusters** pane of the **Create L4-L7 Service Graph Template** dialog box and choose **Create L4-L7 Devices** or **Create Copy Devices**.
The corresponding dialog box appears.

b) Follow the dialog box by entering the appropriate values displayed in the dialog box and clicking **Next** until finished.

Note For an explanation of a field in a dialog box, click the help icon in the top-right corner to display the help file.

c) When finished, click **Finish**.

You return to the **Create L4-L7 Service Graph Template** dialog box.

Step 6 Enter the appropriate values in the fields of the **Create L4-L7 Service Graph Template** dialog box.

Note For an explanation of a field in a dialog box, click the help icon in the top-right corner to display the help file.

Step 7 (Optional) (Only for cloning an existing service graph template) If you want to remove any of the nodes from the cloned service graph template, right-click a node that you want to remove and choose **Remove Node**.

Step 8 To create a service node, drag a Layer 4 to Layer 7 services device from the **Device Clusters** section and drop it between the consumer endpoint group and provider endpoint group. To create a copy node, drag and drop a copy device. This step is optional if you cloned an existing service graph template and the service graph template has all of the nodes that you want to use.

You can drag and drop multiple devices to create multiple nodes. The maximum number of service nodes is 3, although you can drag and drop greater numbers of other devices.

The location where you drop a copy device becomes the point in the data flow from where the copy device copies the traffic.

Step 9 If you created one or more service nodes, in the ***device_name*** **Information** section for each Layer 4 to Layer 7 services device, complete the fields. The fields vary depending on the device type.

Note For an explanation of a field, click the help icon in the top-right corner to display the help file.

Step 10 When finished, click **Submit**.

Step 11 (Optional) In the **Navigation** pane, click the service graph template.
The work pane displays a graphic topology of the service graph template.

Configuring a Service Graph Template Using the REST APIs

You can configure a service graph template using the following REST API:

```
<polUni>
  <fvTenant name="acme">
    <vnsAbsGraph name="G1">
      <vnsAbsTermNodeCon name="Input1">
        <vnsAbsTermConn name="C1">
            </vnsAbsTermConn>
      </vnsAbsTermNodeCon>
      <vnsAbsNode name="Node" funcType="GoTo">
        <vnsRsDefaultScopeToTerm
          tDn="uni/tn-acme/AbsGraph-G1/AbsTermNodeProv-Output1/outtmnl"/>
        <vnsAbsFuncConn name="inside">
          <vnsRsMConnAtt
            tDn="uni/infra/mDev-Insieme-Generic-1.0/mFunc-SubnetFunc/mConn-external"/>
        </vnsAbsFuncConn>
        <vnsAbsFuncConn name="outside">
          <vnsRsMConnAtt
```

```
            tDn="uni/infra/mDev-Insieme-Generic-1.0/mFunc-SubnetFunc/mConn-internal"/>
        </vnsAbsFuncConn>
        <vnsAbsDevCfg>
          <vnsAbsFolder key="oneFolder" name="f1">
            <vnsAbsParam key="oneParam" name="p1" value="v1"/>
          </vnsAbsFolder>
        </vnsAbsDevCfg>
        <vnsAbsFuncCfg>
          <vnsAbsFolder key="folder" name="folder1" devCtxLbl="C1">
            <vnsAbsParam key="param" name="param" value="value"/>
          </vnsAbsFolder>
          <vnsAbsFolder key="folder" name="folder2" devCtxLbl="C2">
            <vnsAbsParam key="param" name="param" value="value"/>
          </vnsAbsFolder>
        </vnsAbsFuncCfg>
        <vnsRsNodeToMFunc tDn="uni/infra/mDev-Insieme-Generic-1.0/mFunc-SubnetFunc"/>
      </vnsAbsNode>
      <vnsAbsTermNodeProv name="Output1">
        <vnsAbsTermConn name="C6">
            </vnsAbsTermConn>
      </vnsAbsTermNodeProv>
      <vnsAbsConnection name="CON1">
        <vnsRsAbsConnectionConns
          tDn="uni/tn-acme/AbsGraph-G1/AbsTermNodeCon-Input1/AbsTConn"/>
       <vnsRsAbsConnectionConns tDn="uni/tn-acme/AbsGraph-G1/AbsNode-Node/AbsFConn-inside"/>

      </vnsAbsConnection>
      <vnsAbsConnection name="CON3">
       <vnsRsAbsConnectionConns tDn="uni/tn-acme/AbsGraph-G1/AbsNode-Node/AbsFConn-outside"/>

        <vnsRsAbsConnectionConns
          tDn="uni/tn-acme/AbsGraph-G1/AbsTermNodeProv-Output1/AbsTConn"/>
      </vnsAbsConnection>
    </vnsAbsGraph>
  </fvTenant>
</polUni>
```

Applying a Service Graph Template to Endpoint Groups Using the GUI

The following procedure explains how to apply a service graph template to endpoint groups:

Before you begin

You must have created the following things:

- Application endpoint groups
- A service graph template

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

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

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Service Graph Templates** > ***template_name***.

Step 4 In the Navigation pane, right-click on the ***template_name*** that you want to apply to EPGs and choose **Apply L4-L7 Service Graph Template**.

The **Apply L4-L7 Service Graph Template To EPGs** dialog box appears. You will be associating a Layer 4 to Layer 7 service graph template to your consumer and provider endpoint groups.

Step 5 Configure a contract in the **Apply L4-L7 Service Graph Template To EPGS STEP 1> Contract** dialog box by entering the appropriate values:

a) If you are configuring an intra-EPG contract, place a check in the **Configure an Intra-EPG Contract** check-box and choose the EPG and network combination from the **EPG / Network** drop-down list.

b) If you are configuring a standard contract, choose the consumer/provider EPGs and network combinations in the appropriate drop-down lists.

c) Create a new contract or choose an existing one by clicking the appropriate radio button in the **Contract** field. If you select **Create A New Contract** and want to configure the filters for it, remove the check from the **No Filter (Allow All Traffic)** check-box. Click **+** to add filter entries and **Update** when complete.

Note For copy service graphs, contracts can only be used multiple times if they are applied to L3Out EPGs. Internal EPGs require an unshared contract.

Step 6 Click **Next**.

The **STEP 2 > Graph** dialog appears.

Step 7 In the ***device_name*** **Information** section, configure the required fields represented with a red box.

Note To include the connector in a preferred group (endpoint to endpoint communication without a contract), choose a configured policy from the **Service EPG Policy** drop-down list.

Step 8 Click **Next**.

The **STEP 3 > device_name Information** dialog appears.

Step 9 Click **Finish**.

You now have an active service graph template.

Applying a Service Graph Template to an Endpoint Security Group Using the GUI

The following procedure explains how to apply a service graph template to an endpoint security group (ESG):

Before you begin

You must have created the following things:

- ESGs
- A service graph template

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

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

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Service Graph Templates** > ***template_name***.

Step 4 In the Navigation pane, right-click on the ***template_name*** that you want to apply to EPGs and choose **Apply L4-L7 Service Graph Template**.

The **Apply L4-L7 Service Graph Template To EPGs** dialog box appears. You will be associating a Layer 4 to Layer 7 service graph template to your consumer and provider endpoint security groups.

Step 5 Configure a contract in the **Apply L4-L7 Service Graph Template To EPGS STEP 1> Contract** dialog box by entering the appropriate values:

a) Select **Endpoint Security Group** as the enpdoint group type.
b) If you are configuring a standard contract, choose the consumer/provider ESGs and network combinations in the appropriate drop-down lists.
c) Create a new contract or choose an existing one by clicking the appropriate radio button in the **Contract** field. If you select **Create A New Contract** and want to configure the filters for it, remove the check from the **No Filter (Allow All Traffic)** check-box. Click **+** to add filter entries and **Update** when complete.

Step 6 Click **Next**.

The **STEP 2 > Graph** dialog appears.

Step 7 In the ***device_name*** **Information** section, configure the required fields represented with a red box.

Step 8 Click **Next**.

The **STEP 3 > device_name Information** dialog appears.

Step 9 Click **Finish**.
You now have an active service graph template.

Applying a Service Graph Template with a Contract Using the NX-OS-Style CLI

The following procedure applies a service graph template with a contract using the NX-OS-style CLI.

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.

```
tenant tenant_name
```

Example:

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

Step 3 Add a service graph.

```
l4l7 graph graph_name [contract contract_name]
```

Parameter	Description
graph	Name of the service graph.

Parameter	Description
contract	Name of the contract that is associated with this service graph instance. Specify the contract only if you want to create the service graph instance. You can simply configure a service graph (equivalent to the service graph template) without instantiating it.

Example:

```
apic1(config-tenant)# l4l7 graph G2 contract C2
```

Step 4 Add a node (service) in the service graph.

```
service node_name [device-cluster-tenant tenant_name] [device-cluster device_name] [mode
deployment_mode]
```

Parameter	Description
service	The name of the service node to add.
device-cluster-tenant	The tenant from which to import the device cluster. Specify this only if the device cluster is not in the same tenant in which the graph is being configured.
device-cluster	Name of the device cluster to use for this service node.
mode	The deployment mode. Possible values are: • ADC_ONE_ARM: Specifies one-arm mode. • ADC_TWO_ARM: Specifies two-arm mode. • FW_ROUTED: Specifies routed (GoTo) mode. • FW_TRANS: Specifies transparent (GoThrough) mode. • OTHERS: Specifies any other deployment mode. If the mode is not specified, then a deployment mode is not used.

Example:

The following example adds node N1 to the device cluster D4, which is from tenant t1:

```
apic1(config-graph)# service N1 device-cluster-tenant t1 device-cluster D4
```

The following example adds node N1 to the device cluster D4, which is from tenant t1, and uses the routed deployment mode:

```
apic1(config-graph)# service N1 device-cluster-tenant t1 device-cluster D4 mode FW_ROUTED
```

Step 5 Add the consumer connector.

```
connector connector_type [cluster-interface interface_type]
```

Parameter	Description
connector	The type of the connector in the service graph. Possible values are: • provider • consumer

Parameter	Description
cluster-interface	The type of the device cluster interface. Possible values are: • provider • consumer Do not specify this parameter if you are a service graph template in tenant `Common`.

Example:

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

Step 6 If the service interface is in a bridge domain, perform the following substeps:

a) Configure the bridge domain for the connectors by specifying the bridge domain information and tenant where the bridge domain is present.

```
bridge-domain tenant tenant_name name bridge_domain_name
```

Parameter	Description
tenant	Tenant that owns the bridge domain. You can only specify a bridge domain from same tenant or tenant `Common`. For example if you are in tenant `t1`, then you cannot specify the bridge domain from tenant `t2`.
name	Name of the bridge domain.

Example:

```
apic1(config-connector)# bridge-domain tenant t1 name bd2
```

b) Configure the direct server return (DSR) virtual IP address (VIP) for the connector.

```
dsr-vip ip_address
```

If you specify the DSR VIP, the Application Policy Infrastructure Controller (APIC) does not learn the VIP.

Parameter	Description
dsr-vip	The virtual IP address of the DSR for the connector.

Example:

```
apic1(config-connector)# dsr-vip 192.168.10.100
```

Step 7 If the service interface is in an L3Out, perform the following substeps:

a) Associate a tenant with the connector and then exit the connector configuration mode.

```
l4l7-peer tenant tenant_name out L3OutExternal epg epg_name
  redistribute redistribute_property
exit
```

Parameter	Description
tenant	The name of the tenant to associate with the connector.
out	The name of the Layer 3 outside.
epg	The name of the endpoint group.

Parameter	Description
redistribute	The properties of the redistribute protocol.

Example:

```
apic1(config-connector)# l4l7-peer tenant t1 out L3OutExternal epg L3ExtNet
  redistribute connected,ospf
apic1(config-connector)# exit
```

b) Repeat steps 5 and 7a for the provider.

Example:

```
apic1(config-service)# connector provider cluster-interface provider
apic1(config-connector)# l4l7-peer tenant t1 out L3OutInternal epg L3IntNet
  redistribute connected,ospf
apic1(config-connector)# exit
```

c) (Optional) Add a router and then exit the node configuration mode.

```
rtr-cfg router_ID
exit
```

Parameter	Description
rtr-cfg	The ID of the router.

Skip this step if you are creating a service graph template in tenant `Common`.

Example:

```
apic1(config-service)# rtr-cfg router-id1
apic1(config-service)# exit
```

Step 8 Configure connections for the consumer and provider and exit the service graph configuration mode.

```
connection connection_name {terminal terminal_type service node_name connector connector_type} |
  {intra_service service1 node_name connector1 connector_type service2 node_name connector2
connector_type}
exit
```

Parameter	Description
connection	The name of the connection.
terminal	Connects a service node to the terminal. Specifies the type of the terminal. Possible values are: • provider • consumer
service service1 service2	The name of the service node to add. service is used only with terminal. service1 and service2 are used only with intra_service.

Parameter	Description
connector connector1 connector2	The type of the connector. Possible values are: • provider • consumer connector is used only with terminal. connector1 and connector2 are used only with intra_service.
intra_service	Connects a service node to another node.

Example:

The following example configures the connections of a single node graph:

```
apic1(config-graph)# connection CON1 terminal consumer service N1 connector consumer
apic1(config-graph)# connection CON2 terminal provider service N2 connector provider
apic1(config-graph)# exit
```

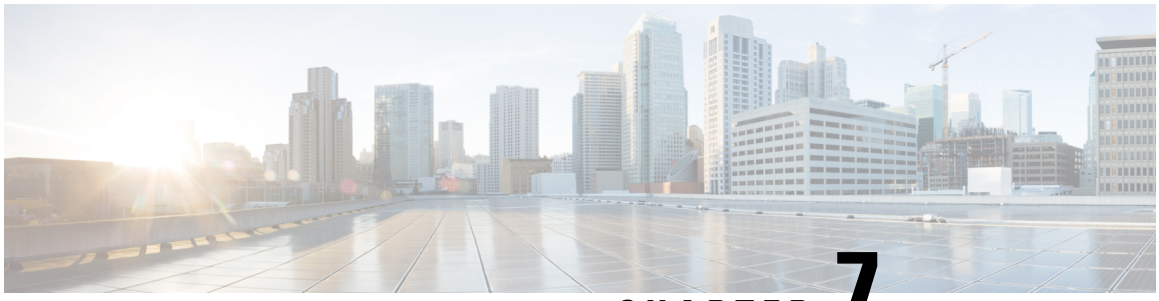
The following example configures the connections of a two node graph:

```
apic1(config-graph)# connection CON1 terminal consumer service N1 connector consumer
apic1(config-graph)# connection CON2 intra_service service1 N1 connector1 provider service2 N2
connector2 consumer
apic1(config-graph)# connection CON3 terminal provider service N2 connector provider
apic1(config-graph)# exit
```

Step 9 Exit the configuration mode.

Example:

```
apic1(config-tenant)# exit
apic1(config)# exit
```



CHAPTER 7

Configuring Route Peering

- About Route Peering, on page 53
- Open Shortest Path First Policies, on page 54
- Border Gateway Protocol Policies, on page 58
- Selecting an L3extOut Policy for a Cluster, on page 61
- Route Peering End-to-End Flow, on page 62
- Cisco Application Centric Infrastructure Fabric Serving As a Transit Routing Domain, on page 64
- Configuring Route Peering Using the GUI, on page 65
- Configuring Route Peering Using the NX-OS-Style CLI, on page 69
- Troubleshooting Route Peering, on page 71

About Route Peering

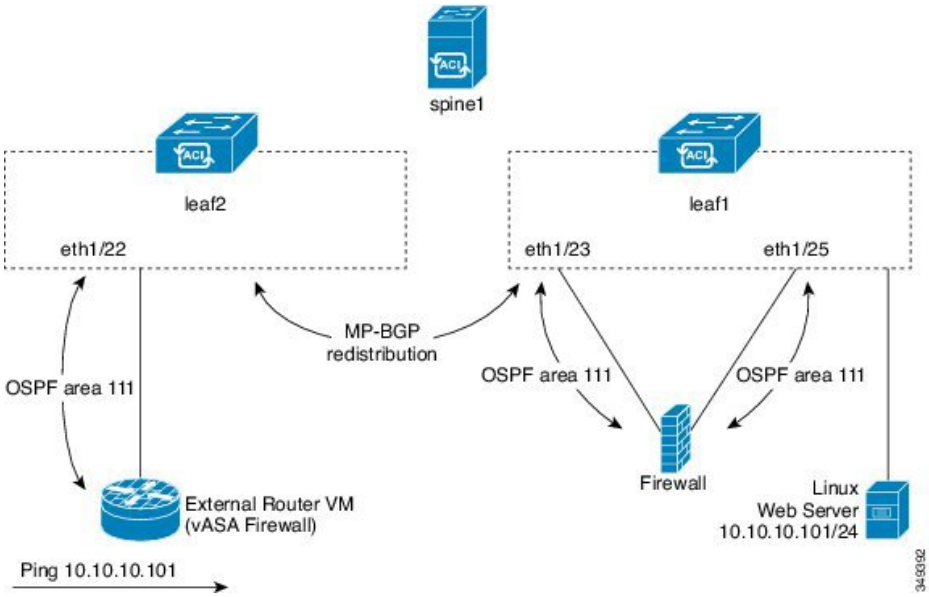
Route peering is a special case of the more generic Cisco Application Centric Infrastructure (ACI) fabric as a transit use case, in which route peering enables the ACI fabric to serve as a transit domain for Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP) protocols. A common use case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

The following protocols are supported for route peering:

- OSPF
- OSPFv3
- iBGPv4
- iBGPv6
- Static routes

The following figure shows how route peering is commonly deployed:

Figure 3: Common Route Peering Topology



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

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

For a more detailed example of the route peering topology, see Route Peering End-to-End Flow, on page 62.

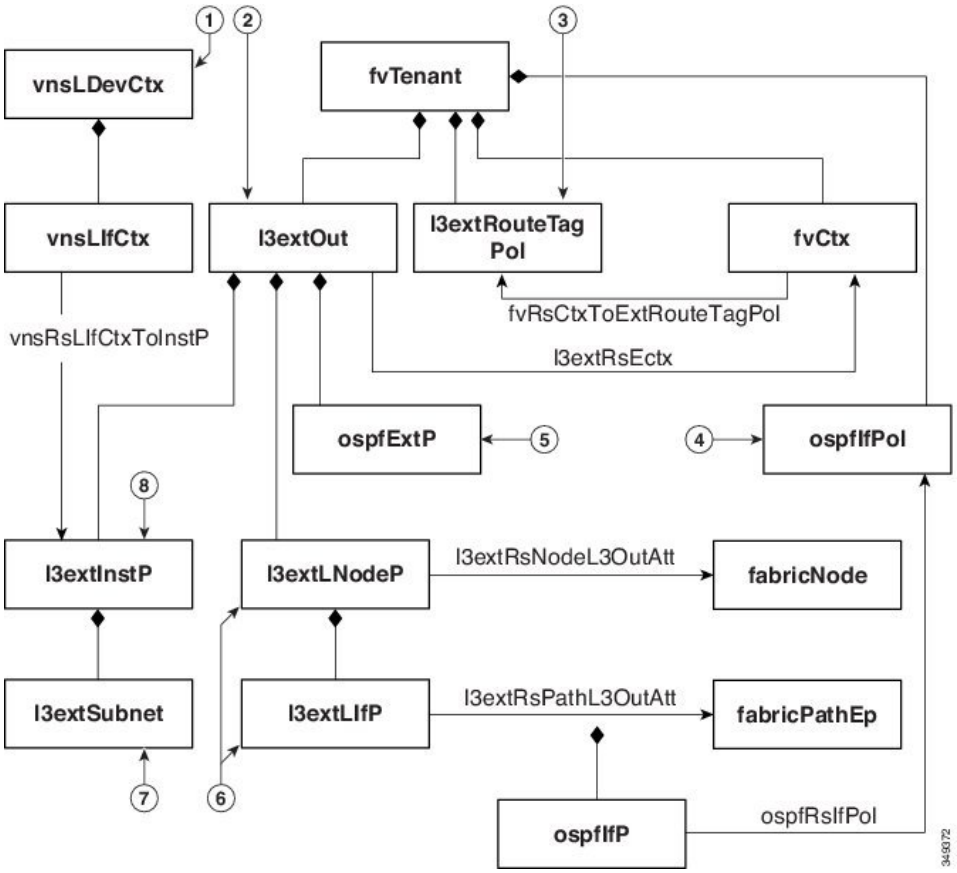
For more information about configuring `l3extOut` policies, see the *Cisco Application Centric Infrastructure Fundamentals Guide*.

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

Open Shortest Path First Policies

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

Figure 4: OSPF Route Peering Object Relations



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

Two example `l3extOut` policies, `OspfExternal` and `OspfInternal`, are shown below. These policies are deployed on the external and internal legs of the firewall device in Figure 3: Common Route Peering Topology, on page 54. The `l3extOut` policy specifies one or more prefix-based EPGs (`l3extInstP`), which control how traffic is classified by the fabric leaf and also how routes are imported from and exported to the service device. The `l3extOut` policy contains the OSPF per-area policy (`ospfExtP`) and one or more OSPF interface policies (`ospfIfPol`) that are specified under it.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`OspfExternal` Policy

`OspfInternal` Policy

Virtual Services

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

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

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

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

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

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

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

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

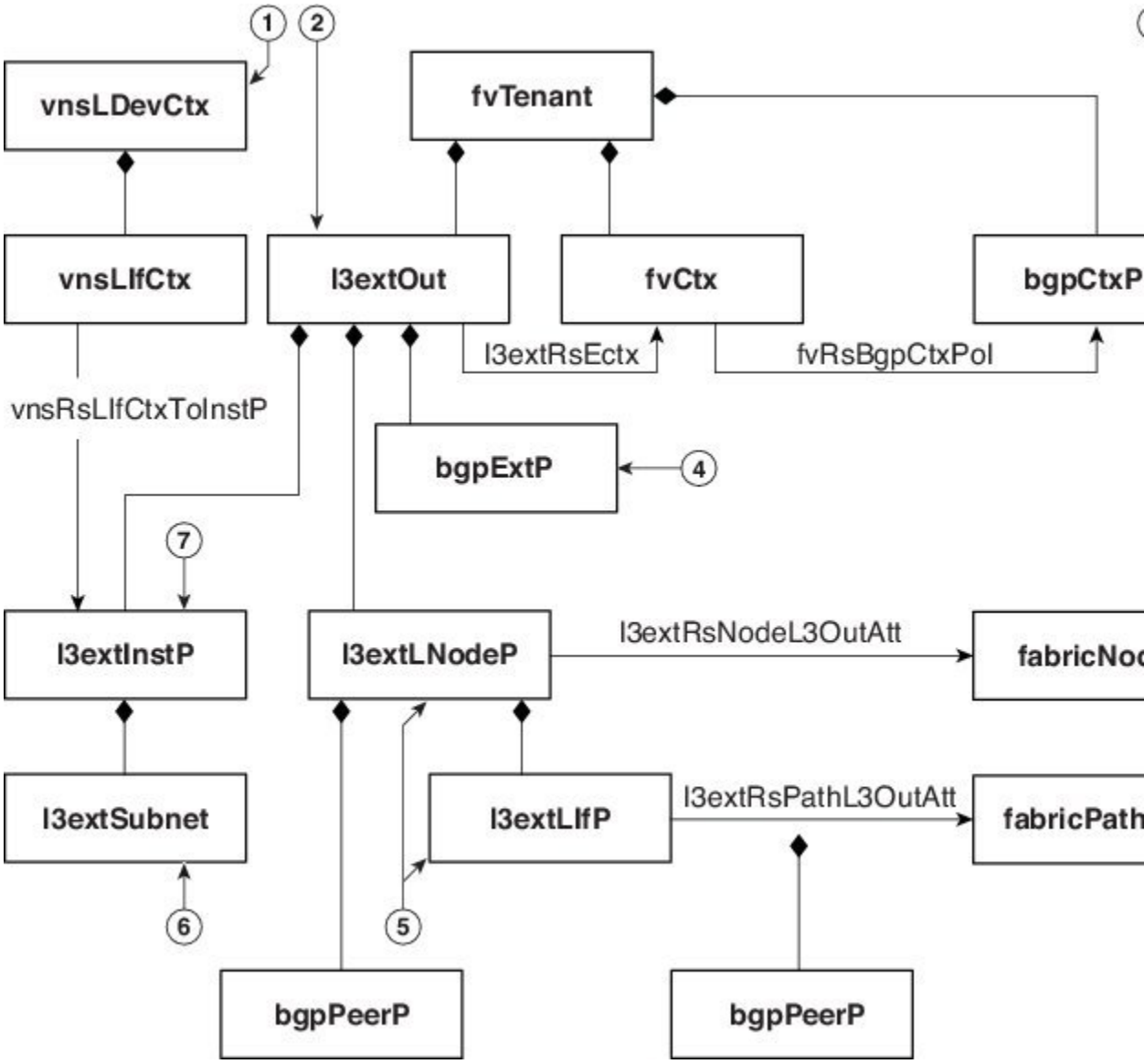
You can deploy virtual services with route peering, although the `l3extRsPathL3OutAtt` validation with the `vnsCIf` object is not performed. The datapath will work only if the `l3extOut` object is deployed to the correct leaf to which the virtual service device is connected.

Border Gateway Protocol Policies

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

The following figure illustrates the route peering object relations:

Figure 5: iBGP Route Peering Object Relations



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

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

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

    <l3extOut name="BgpExternal" status="created,modified">
      <l3extLNodeP name="bLeaf-101">
        <!-- <bgpPeerP addr="40.40.40.102/32 "ctrl="send-com"/> -->
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.8/28>"
          <l3extLoopBackIfP addr="50.50.50.100/32"/>
        </l3extRsNodeL3OutAtt>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
            ifInstT="ext-svi" encap="vlan-3843" addr="40.40.40.100/28 "mtu="1500">
            <bgpPeerP addr="40.40.40.102/32 "ctrl="send-com"/>
          </l3extRsPathL3OutAtt>
        </l3extLIfP>
      </l3extLNodeP>
      <bgpExtP/>
      <l3extInstP name="ExtInstP">
        <l3extSubnet ip="40.40.40.100/28 "scope="import-security"/>
        <l3extSubnet ip="10.10.10.0/24 "scope="import-security"/>
        <l3extSubnet ip="20.20.20.0/24 "scope="export-rtctrl"/>
      </l3extInstP>
      <l3extRsEctx tnFvCtxName="commonctx"/>
    </l3extOut>
  </fvTenant>
</polUni>
```

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

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

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

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

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

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

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

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

Selecting an L3extOut Policy for a Cluster

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

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

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

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

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

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

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

The following code is an example `vnsRtrCfg` policy:

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

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

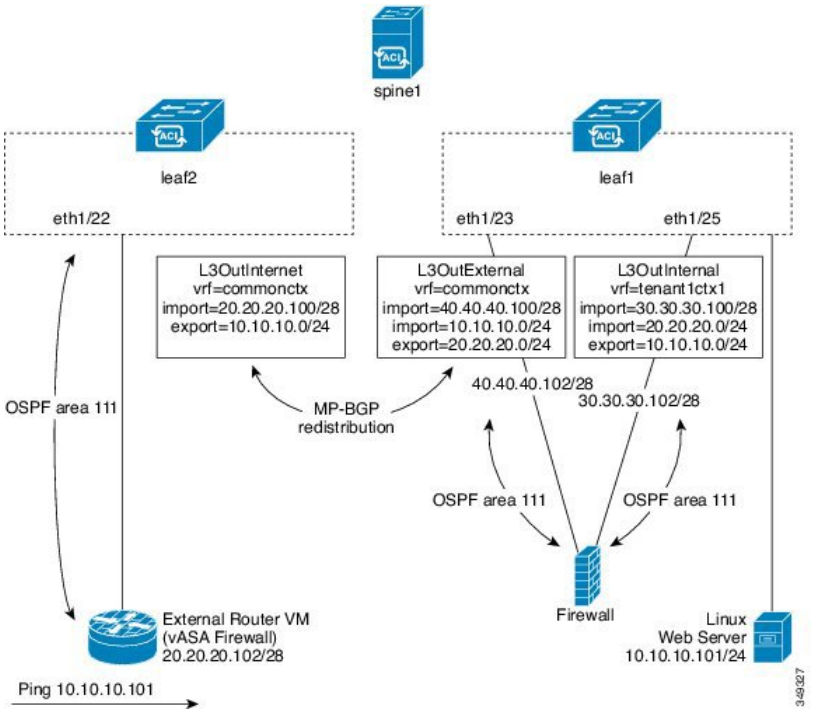
```
<vnsCDev name="ASA">
    <vnsCIf name="Gig0/0">
        <vnsRsCIfPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"/>
    </vnsCIf>
    <vnsCIf name="Gig0/1">
        <vnsRsCIfPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/25]"/>
    </vnsCIf>
    <vnsCMgmt name="devMgmt" host="{{asaIp}}"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

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

Route Peering End-to-End Flow

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

Figure 6: Route Peering End-to-End Flow



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

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

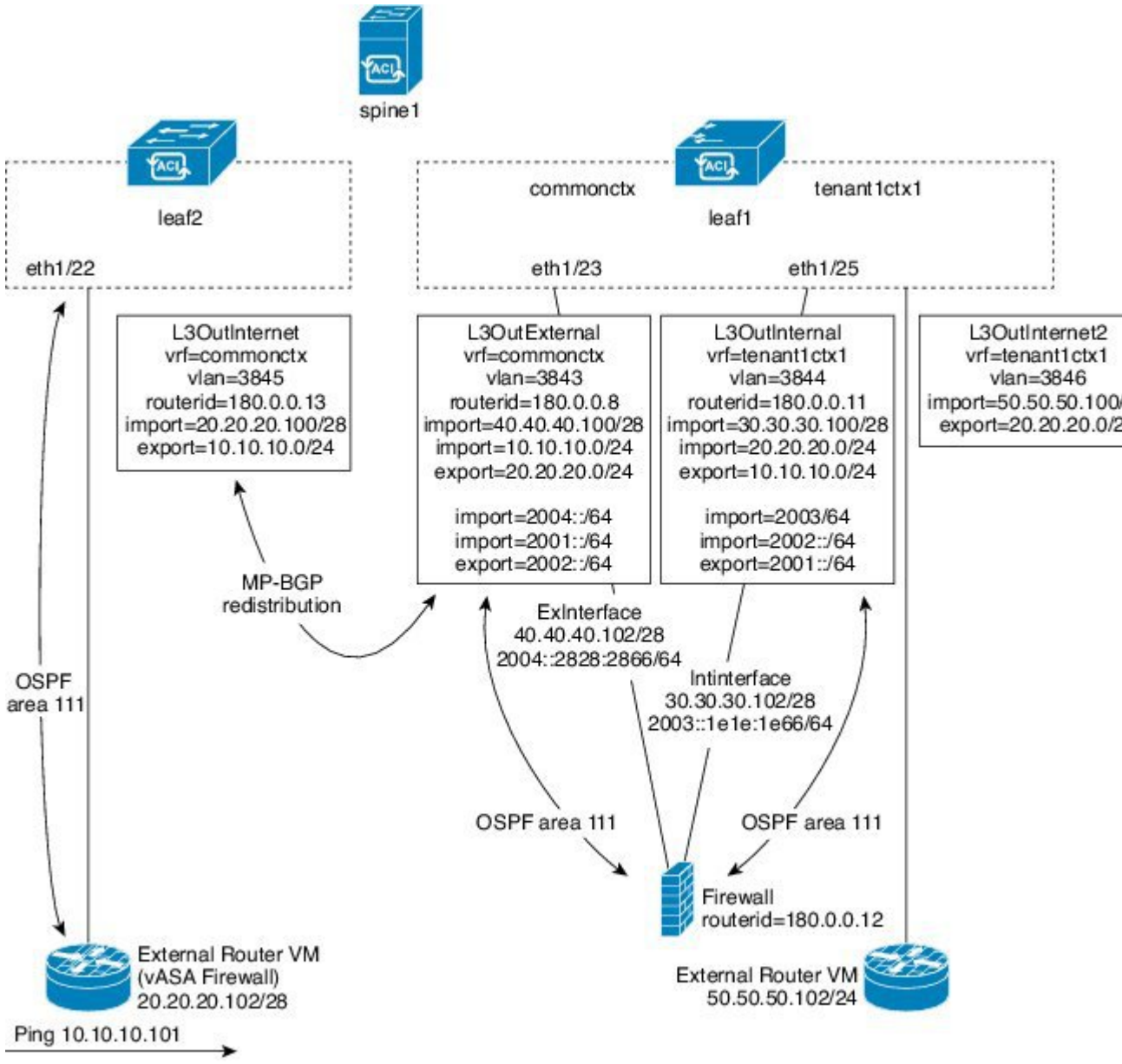
To publish the web server route to the external world, you deploy two `l3extOuts`—`L3OutExternal` and `L3OutInternal`—to the leaf switch ports to which the service device is connected. As a result, Open Shortest Path First (OSPF) peering sessions are established between the leaf switch and the firewall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

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

Cisco Application Centric Infrastructure Fabric Serving As a Transit Routing Domain

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

Figure 7: ACI Fabric Serving As a Transit Routing Domain



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

Configuring Route Peering Using the GUI

You must perform the following tasks to configure route peering:

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

   See Creating a Static VLAN Pool Using the GUI, on page 65.

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

   See Creating an External Routed Domain Using the GUI, on page 66.

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

   See Creating an External Routed Network Using the GUI, on page 66.

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

   See Creating a Router Configuration Using the GUI, on page 68.

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

   See Creating a Service Graph Association Using the GUI, on page 68.

Creating a Static VLAN Pool Using the GUI

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

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

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

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

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

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

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

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

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

Creating an External Routed Domain Using the GUI

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

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

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

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

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

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

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

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

Creating an External Routed Network Using the GUI

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

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

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

Step 3 In the Navigation pane, choose ***tenant_name*** > **Networking** > **External Routed Networks**.

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

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

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

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

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

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

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

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

b) In the **Router ID** field, enter a router ID that the ACI fabric will use for the routing protocol process.
c) If you are planning to use static routing between the ACI fabric and device, in the **Static Routes** section click +. Otherwise, go to step Step 10, on page 67.

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

Step 9 Click **OK**.

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

Step 11 If you are using BGP as the dynamic routing protocol with the device, in the **BGP Peer Connectivity Profiles** section, click +. Otherwise, go to step Step 14, on page 67.

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

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

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

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

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

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

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

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

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

Step 19 Click **OK**.

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

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

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

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

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

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

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

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

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

Step 26 Click **OK**.

Step 27 (Optional) Create additional subnets as needed.

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

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

Creating a Router Configuration Using the GUI

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

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

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

Step 3 In the Navigation pane, choose ***tenant_name*** > **Services** > **L4-L7** > **Router configurations**.

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

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

Step 6 Click **Update**.

Creating a Service Graph Association Using the GUI

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

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

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

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Device Selection Policies** > ***device_selection_policy***.

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

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

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

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

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

The following changes occur:

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

Configuring Route Peering Using the NX-OS-Style CLI

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

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.

Example:

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

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

Example:

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

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

Example:

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

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

Example:

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

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

Example:

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

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

Example:

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

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

Example:

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

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

Example:

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

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

Example:

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

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

Example:

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

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

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

Troubleshooting Route Peering

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

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

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

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

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

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

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

Verifying the Leaf Switch Route Peering Functionality Using the CLI

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

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

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

fab2-leaf3#
```

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

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

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

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

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

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

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

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

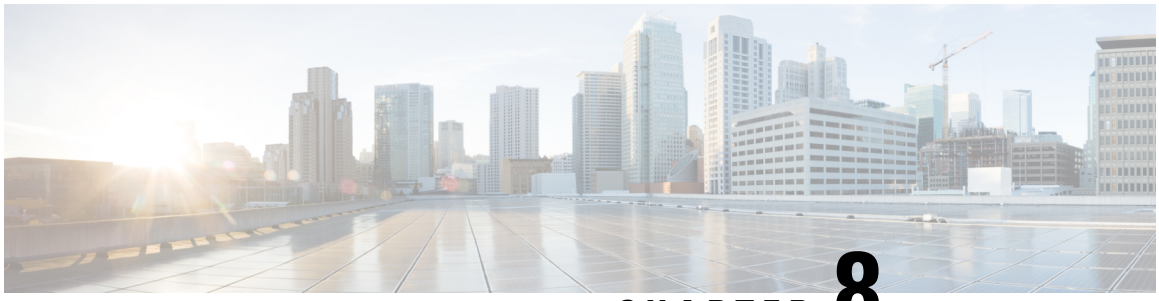
```
fab2-leaf3# show ip route vrf user1:global
IP Route Table for VRF "user1:global"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

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

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

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

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



CHAPTER 8

Configuring Policy-Based Redirect

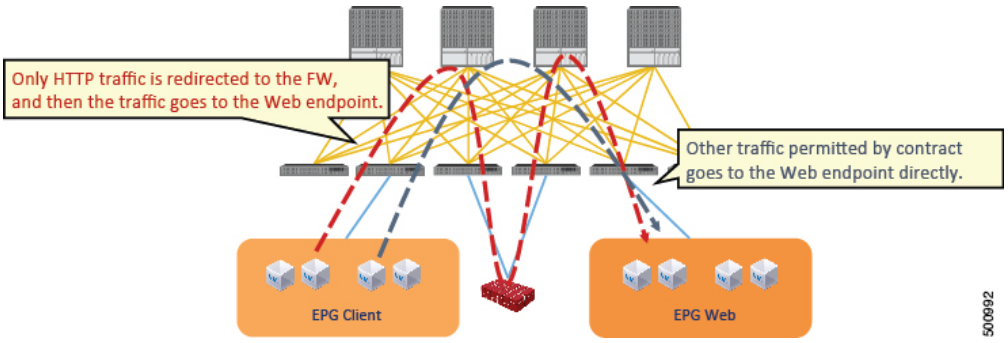
- About Policy-Based Redirect, on page 75
- About Multi-Node Policy-Based Redirect, on page 89
- About Symmetric Policy-Based Redirect, on page 89
- Policy Based Redirect and Hashing Algorithms, on page 90
- Policy-Based Redirect Resilient Hashing, on page 90
- About PBR Backup Policy, on page 92
- About the Bypass Action, on page 96
- Policy-Based Redirect with an L3Out, on page 99
- PBR Support for Service Nodes in Consumer and Provider Bridge Domains , on page 107
- About Layer 1/Layer 2 Policy-Based Redirect, on page 107
- Policy-Based Redirect and Tracking Service Nodes, on page 117
- About Location-Aware Policy Based Redirect, on page 121
- Policy-Based Redirect and Service Graphs to Redirect All EPG-to-EPG Traffic Within the Same VRF Instance, on page 124
- Dynamic MAC Address Detection for a Layer 3 Policy-Based Redirect Destination, on page 129

About Policy-Based Redirect

Cisco Application Centric Infrastructure (ACI) policy-based redirect (PBR) enables provisioning service appliances, such as firewalls or load balancers. Typical use cases include provisioning service appliances that can be pooled, tailored to application profiles, scaled easily, and have reduced exposure to service outages. PBR simplifies the deployment of service appliances by enabling the provisioning consumer and provider endpoint groups to be all in the same virtual routing and forwarding (VRF) instance. PBR deployment consists of configuring a route redirect policy and a cluster redirect policy, and creating a service graph template that uses the route and cluster redirect policies. After the service graph template is deployed, use the service appliance by enabling endpoint groups to consume the service graph provider endpoint group. This can be further simplified and automated by using vzAny. While performance requirements may dictate provisioning dedicated service appliances, virtual service appliances can also be deployed easily using PBR.

The following figure illustrates the use case of redirecting specific traffic to the firewall:

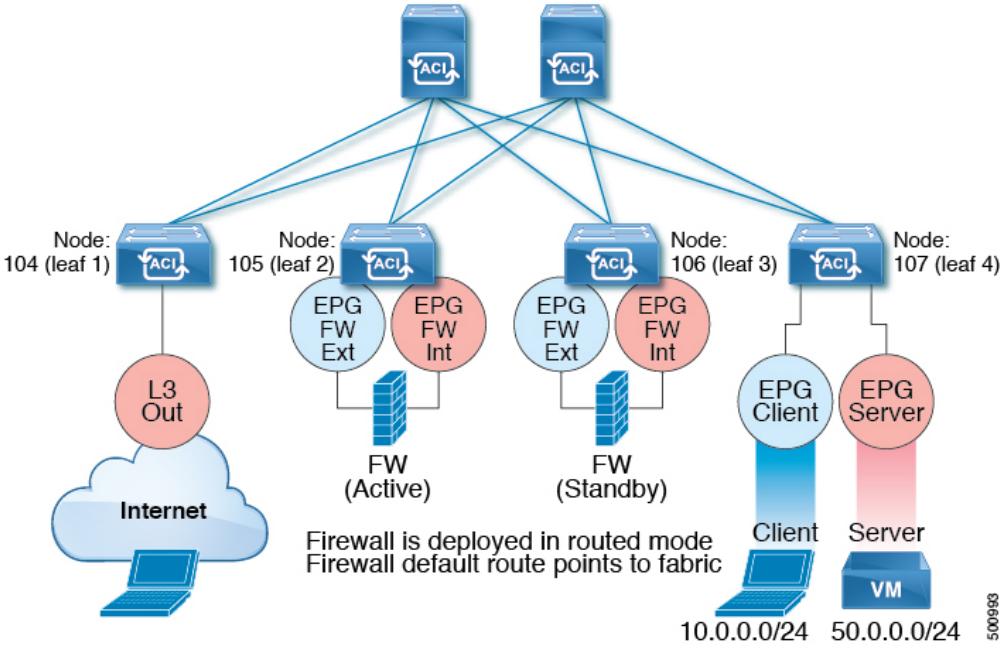
Figure 8: Use Case: Redirecting Specific Traffic to the Firewall



In this use case, you must create two subjects. The first subject permits HTTP traffic, which then gets redirected to the firewall. After the traffic passes through the firewall, it goes to the Web endpoint. The second subject permits all traffic, which captures traffic that is not redirected by the first subject. This traffic goes directly to the Web endpoint.

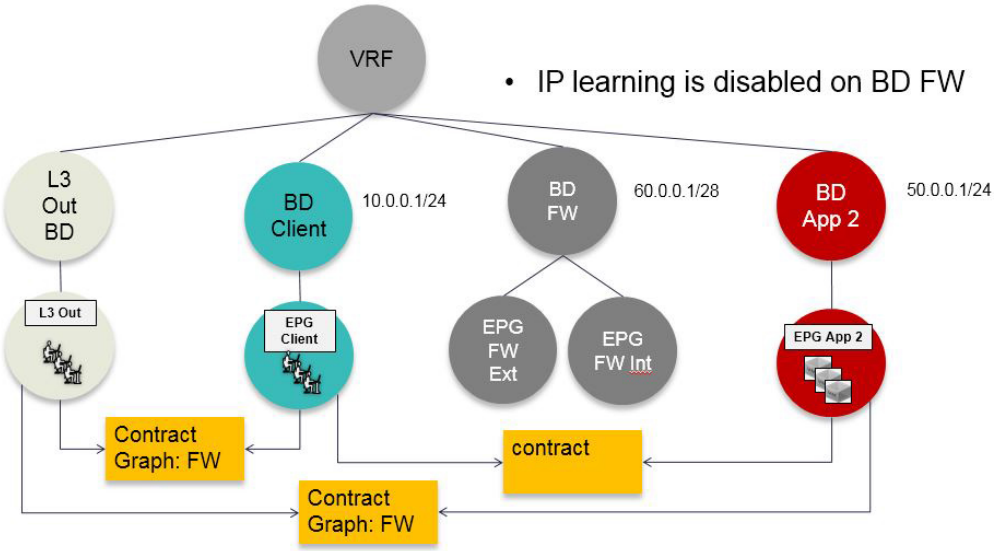
The following figure illustrates a sample ACI PBR physical topology:

Figure 9: Sample ACI PBR Physical Topology



The following figure illustrates a sample ACI PBR logical topology:

Figure 10: Sample ACI PBR Logical Topology



While these examples illustrate simple deployments, ACI PBR enables scaling up mixtures of both physical and virtual service appliances for multiple services, such as firewalls and server load balancers.

Guidelines and Limitations for Configuring Policy-Based Redirect

Observe the following guidelines and limitations when planning policy-based redirect (PBR) service nodes:

- A firewall (or a device that does not perform IP address translation) is inserted by using PBR for both directions.
- A load balancer (or a device that perorms IP address translation) is inserted by using unidirectional PBR. The destination IP address (VIP address or NAT'd IP address) for the other direction is owned by the device. The exception is Layer 2 direct server return, where the return traffic does not come back to the load balancer.
- The source MAC address of the packet can be rewritten because of the need to route the packet with PBR inside the fabric. The time-to-live (TTL) field in the IP address header will be decremented by as many times as the packet is routed within the fabric.
- Select the same action for both service legs. In other words, if you select the deny action for the internal service leg, you should also select the deny action for the external service leg.
- L3Out EPGs and regular EPGs can be consumer or provider EPGs.
- L2Out EPGs cannot be either consumer or provider EPGs.
- For a Cold Standby active/standby deployment, configure the service nodes with the MAC address of the active deployment. In a Cold Standby active/standby deployment, when the active node goes down, the standby node takes over the MAC address of active node.
- You must provide the next-hop service node IP address.

- Prior to the 5.2(1) release, you must provide the virtual MAC address. Beginning with the 5.2(1) release, you can optionally choose not to provide the virtual MAC address and instead let the Cisco Application Policy Infrastructure Controller (Cisco APIC) detect the address dynamically.
- If you provision service appliances in the same bridge domain, you must use Cisco Nexus 9300-EX and 9300-FX platform leaf switches.
- When downgrading from the Cisco APIC release 3.1, an internal code checks whether the policy-based redirect bridge domain uses the same bridge domain as a consumer or a provider. If it does, then the fault is disabled during the downgrade as such a configuration is not supported in earlier Cisco APIC versions.
- If you downgrade from the 5.2(1) or later release to a release earlier than 5.2(1), you must remove all PBR-related configurations that include PBR-related features from the 5.2 releases and you must remove the associated service graphs. For example:
  - Remove a device selection policy that uses a PBR destination in an L3Out.
  - Remove a Layer 4 to Layer 7 services device that uses the enhanced lag policy.
  - Remove an IP SLA monitoring policy that uses the HTTP SLA type.
  - Remove a PBR destination that does not have the destination MAC address configured.
- The service appliance, source, and bridge domain can be in the same VRF instance.
- For Cisco N9K-93128TX, N9K-9396PX, N9K-9396TX, N9K-9372PX, and N9K-9372TX switches, the service appliance must not be in the same leaf switch as either the source or destination endpoint group. For Cisco N9K-C93180YC-EX and N9K-93108TC-EX switches, the service appliance can be in the same leaf switch as either the source or destination endpoint group.
- PBR node interfaces are not supported on FEX host interfaces. A PBR node interface must be connected under leaf down link interface, not under FEX host interface. Consumer and provider endpoints can be connected under FEX host interfaces.
- The service appliance can only be in a bridge domain.
- The contract offered by the service appliance provider endpoint group can be configured to `allow-all`, but traffic should be routed by the Cisco Application Centric Infrastructure (Cisco ACI) fabric.
- If you use the Cisco Nexus 9300-EX and 9300-FX platform leaf switches, it is not necessary for you to have the endpoint dataplane learning disabled on policy-based redirect bridge domains. During service graph deployment, the endpoint dataplane learning will be automatically disabled only for policy-based redirect node EPG. If you use non-EX and non-FX platform leaf switches, you must have the endpoint dataplane learning disabled on policy-based redirect bridge domains. The policy-based redirect bridge domain must have the endpoint dataplane learning disabled.
- You can attach a service graph with PBR to a contract subject. The intra-EPG contract with the service graph cannot be used as an inter-EPG contract at the same time. You must use a separate contract for inter-EPG and intra-EPG communication when used with a service graph that has redirect enabled.
- You can use the filters-from-contract option in the service graph template to use the specific filter of the contract subject where the service graph is attached, instead of the default filter for zoning-rules that do not include consumer EPG class ID as source or destination. For zoning-rules that have consumer EPG class ID as source or destination, it uses the specific filter regardless the option.
- Multi-node policy-based redirect (multi-node PBR):

- Supports up to five function nodes in a service graph that can be configured for policy-based redirect.
- When using a multi-node PBR service chain, all the service devices have to be either in local leaf switch or they have to be connected to a remote leaf switch, but should not spread across both.
  - Supported topology:

    In this topology, *RL* means remote leaf switch and *LL* means local leaf switch that is under main location, and not under remote leaf switch.
    - N1(LL)--N2(LL)--N3(LL): All the devices are connected to local leaf switches not distributed across main location and remote leaf switch.
    - N1(RL)-N2(RL)--N3(RL): All the devices are connected to remote leaf switches.
  - Topology not supported:
    - N1(LL)--N2(RL)--N3(LL): Service devices are distributed across local leaf switches and remote leaf switches.
- Multi-node PBR Layer 3 destination guidelines for load balancers:
  - Layer 3 destination upgrade: The Layer 3 destination (VIP) parameter is enabled by default after the upgrade. No issues will occur from this because if the PBR policy was not configured on a specific service node (prior to the 3.2(1) release), the node connector was treated as an Layer 3 destination and will continue to be in the new Cisco APIC release.
  - Traffic does not always need to be destined to only consumer/provider.
  - In the forward direction, the traffic is destined to load balancer VIP address.
  - In the reverse direction, if SNAT is enabled, the traffic is destined to the load balancer's internal leg.
  - In both directions, enable (check) Layer 3 destination (VIP) on the Logical Interface Context.
  - Enable (check) Layer 3 destination (VIP) in both directions to allow you to switch from SNAT to No-SNAT on the load balancer internal by configuring the PBR policy on the internal side.
  - If SNAT is disabled:
    - Reverse direction traffic is destined to consumer but not to load balancer internal leg (enable PBR policy on the internal leg).
    - Layer 3 destination (VIP) is not applicable in this case because a PBR policy is applied.
- Multicast and broadcast traffic redirection is not supported.
- If you change a redirect policy's destination to a different group, the Cisco APIC raises a fault due to the change and the policy's operational status becomes disabled. You must clear the fault to re-enable the policy.
- An intra-EPG or intra-external EPG contract with PBR must not be used for the inter-EPG contract.
- When Migrating endpoints from a non-PBR EPG to a PBR EPG, the remote endpoints on the destination leaf switches do not clear their remote endpoints, which have the sclass details of the old non-PBR EPG. This issue occurs when the destination leaf switch with the remote endpoint is a switch with the -EX,

-FX, or -GX suffix in the product ID. This issue does not occur with switches that have -FX2, -GX2, or a later suffix in the product ID.

If you encounter this issue, you can manually clear the remote endpoint by using the following CLI command:

```
vsh -c "clear system internal epm endpoint key vrf vrf_name ip ip_name"
```

- Supported policy-based redirect configurations include the following:

Figure 11: Policy-based Redirect in the Same VRF Instance

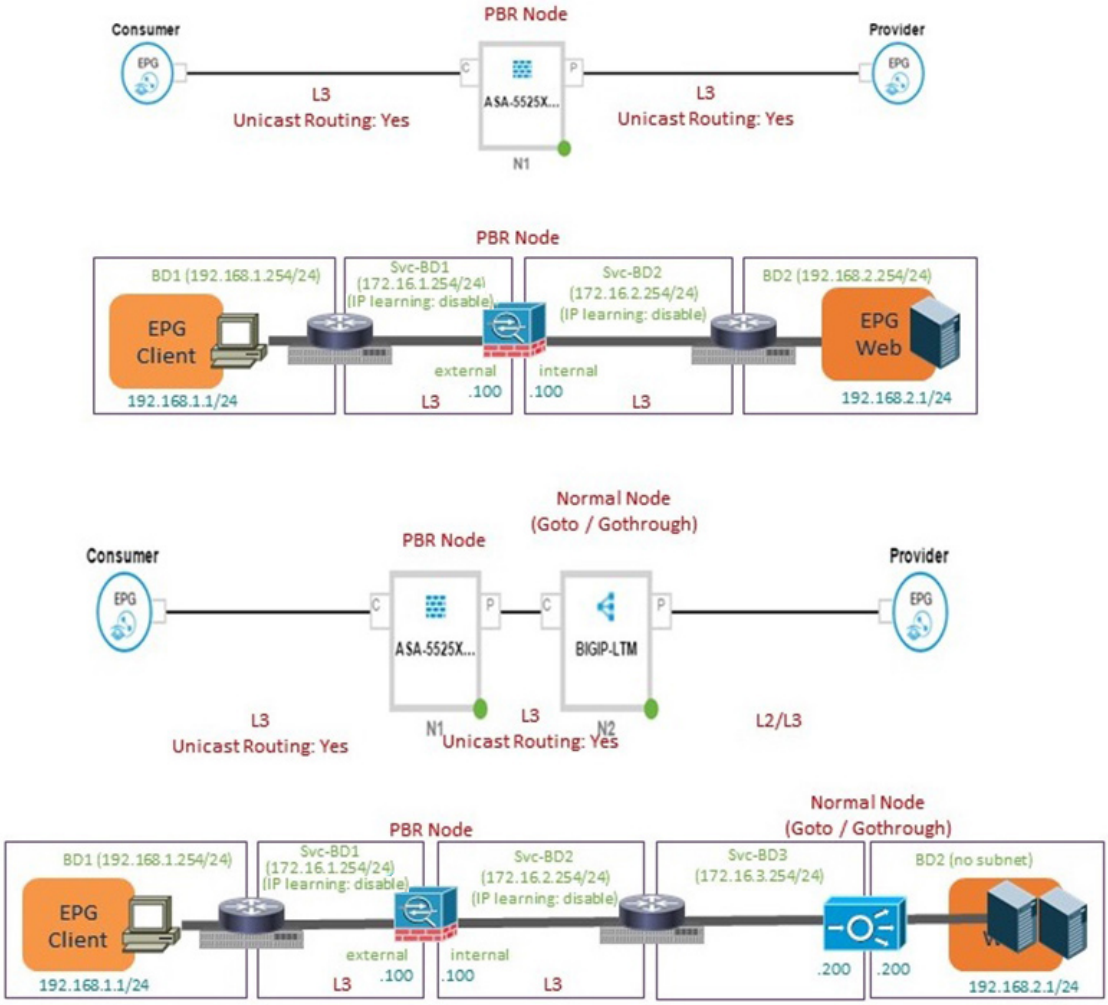


Figure 12: Policy-based Redirect With Different VRF Instances

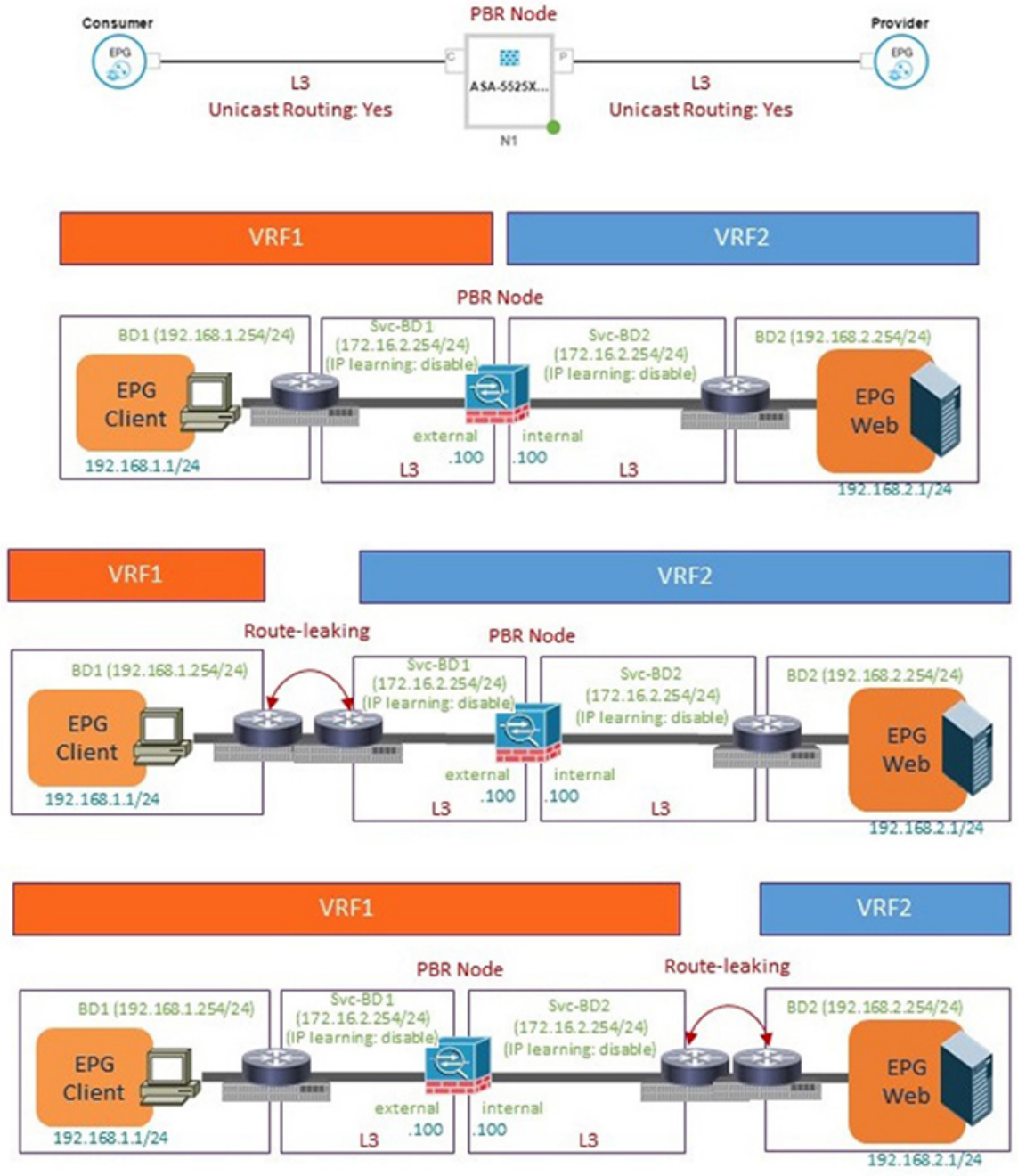
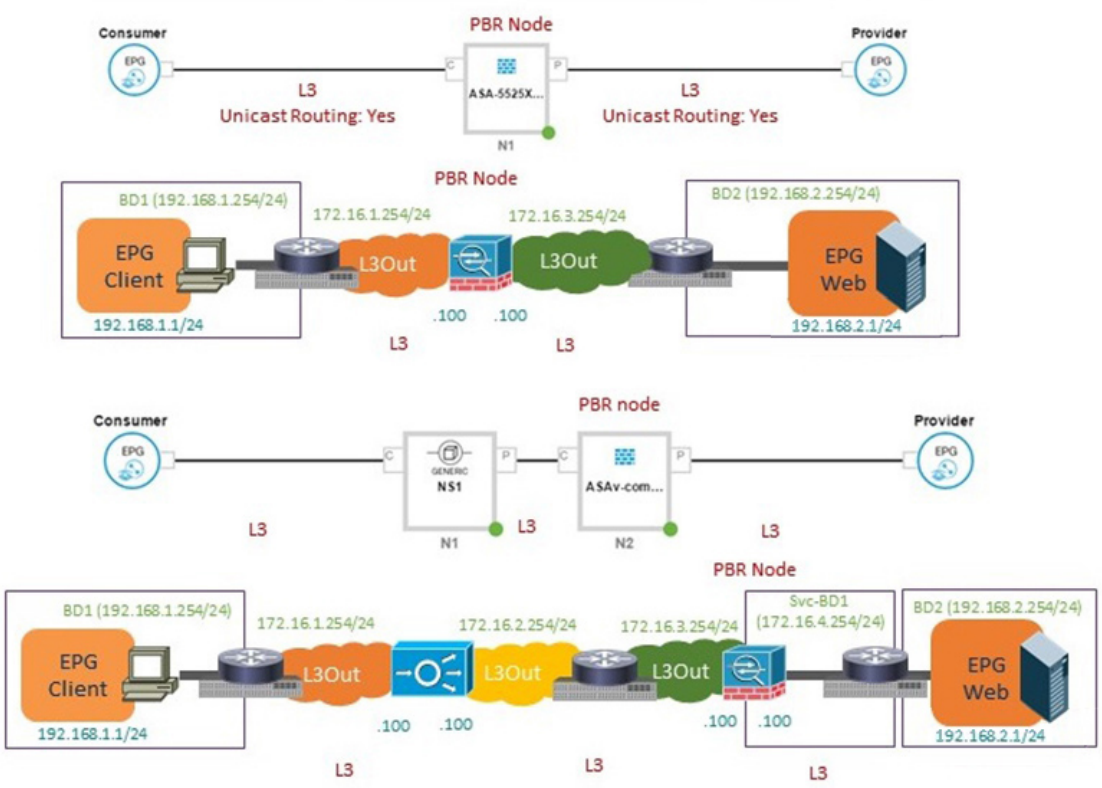
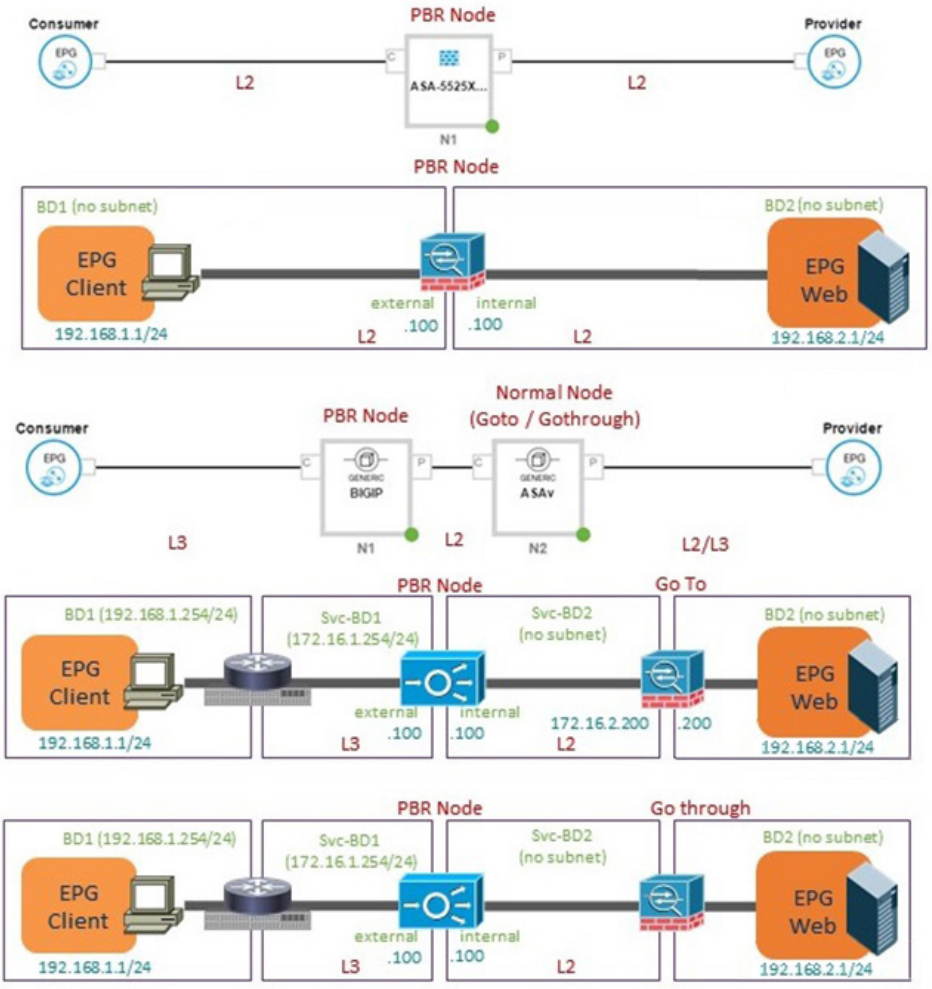


Figure 13: Policy-based Redirect With an L3Out Destination



- Unsupported policy-based redirect configurations include the following:

Figure 14: Unsupported Policy-based Redirect Configurations



Configuring Policy-Based Redirect Using the GUI

The following procedure configures policy-based redirect (PBR) using the GUI.

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

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

Step 3 In the Navigation pane, choose **Tenant** *tenant_name* > **Services** > **L4-L7** > **Devices**.

Step 4 In the Work pane, choose **Actions** > **Create L4-L7 Devices**.

Step 5 In the **Create L4-L7 Devices** dialog box, complete the fields as required.

In the **General** section, the **Service Type** can be **Firewall**, **ADC**, or Other.

Note For a Layer 1/Layer 2 PBR configuration, create the Layer 4 to Layer 7 service device, and perform the following steps:

a. Select the **Service Type** as **Other**.

b. Select the **Device Type Physical** (cloud/virtual is not supported).

c. Select a physical domain.

d. Select the **Function Type L1** or **L2** as required.

e. Create external and internal concrete interfaces and port connectivity on the corresponding leafs.

f. Create Cluster interfaces by selecting the previously created concrete interfaces. You must specify a VLAN encapsulation when creating this interface. The encapsulation is pushed to the service device.

Note For static VLAN configuration, ensure external and internal legs have a different VLAN for Layer 2, otherwise it is the same VLAN for Layer 1.

Step 6 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Service Graph Templates**.

Step 7 In the Work pane, choose **Action** > **Create L4-L7 Service Graph Template**.

Step 8 In the **Create L4-L7 Service Graph Template** dialog box, perform the following actions:

a) In the **Graph Name** field, enter a name for the service graph template.

b) For the **Graph Type** radio buttons, click **Create A New Graph**.

c) Drag and drop the device that you created from the **Device Clusters** pane to between the consumer endpoint group and provider endpoint group. This creates the service node.

As of Cisco Application Policy Infrastructure Controller (APIC) release 4.2(1), you can optionally repeat step c to include up to five (5) service nodes.

d) Select the following based on the service type of the device:

For Firewall, select **Routed** and continue with the steps below.

For ADC, select **One-Arm** or **Two-Arm** and continue with the steps below.

e) Select the **Route Redirect** checkbox.

f) Click **Submit**.

The new service graph template appears in the **Service Graph Templates** table.

Step 9 In the Navigation pane, choose **Tenant** *tenant_name* > **Policies** > **Protocol** > **L4-L7 Policy Based Redirect**.

Step 10 In the Work pane, choose **Action** > **Create L4-L7 Policy Based Redirect**.

Step 11 In the **Create L4-L7 Policy Based Redirect** dialog box, complete the fields as required. This policy-based redirect policy is for the consumer connector.

Step 12 Create another policy-based redirect policy for the provider connector.

Step 13 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Service Graph Templates** > ***service_graph_template_name***.

Choose the service graph template that you just created.

Step 14 Right click the service graph template and choose **Apply L4-L7 Service Graph Template**.

Step 15 In the **Apply L4-L7 Service Graph Template to EPGs** dialog box, perform the following actions:

a) In the **Consumer EPG/External Network** drop-down list, choose the consumer endpoint group.

b) In the **Provider EPG/External Network** drop-down list, choose the provider endpoint group.

c) For the **Contract** radio buttons, click **Create A New Contract**.
d) In the **Contract Name** field, enter a name for the contract.
e) Do not put a check in the **No Filter (Allow All Traffic)** check box.
f) On the **Filter Entries** table, click **+** to add an entry.
g) For the new filter entry, enter "IP" for the name, choose **IP** for the **Ether Type**, and click **Update**.
h) Click **Next**.
i) For the Consumer Connector **Redirect Policy** drop-down list, choose the redirect policy that you created for the consumer connector.
j) For the Consumer Connector **Cluster Interface** drop-down list, choose the consumer cluster interface.
k) For the Provider Connector **Redirect Policy** drop-down list, choose the redirect policy that you created for the provider connector.
l) For the Provider Connector **Cluster Interface** drop-down list, choose the provider cluster interface.
m) Cick **Finish**.

Configuring Policy-Based Redirect Using the NX-OS-Style CLI

The example commands in this procedure include the route redirect, the cluster redirect, and the graph deployment. The device is created under tenant T1.

Step 1 Create the device cluster.

Example:

```
l4l7 cluster name ifav-asa-vm-ha type virtual vlan-domain ACIVswitch service FW function go-to
    cluster-device Device2 vcenter ifav108-vcenter vm "ASAv_HA1"
    cluster-device Device1 vcenter ifav108-vcenter vm "ASAv_HA"
    cluster-interface provider
      member device Device1 device-interface GigabitEthernet0/1
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 3"
        exit
      member device Device2 device-interface GigabitEthernet0/1
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 3"
        exit
      exit
    cluster-interface failover_link
      member device Device1 device-interface GigabitEthernet0/8
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 10"
        exit
      member device Device2 device-interface GigabitEthernet0/8
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 10"
        exit
      exit
    cluster-interface consumer
      member device Device1 device-interface GigabitEthernet0/0
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 2"
        exit
      member device Device2 device-interface GigabitEthernet0/0
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 2"
```

```
        exit
      exit
    exit
exit
```

Step 2 Under tenant PBRv6_ASA_HA_Mode, deploy the PBR service graph instance.

Example:

```
tenant PBRv6_ASA_HA_Mode
  access-list Contract_PBRv6_ASA_HA_Mode_Filter
    match ip
    exit
```

Step 3 Create a contract for PBR with the filter match IP protocol. Under the subject, specify the Layer 4 to Layer 7 service graph name.

The contract offered by the service appliance provider endpoint group cannot be configured with the `allow-all` setting.

Example:

```
  contract Contract_PBRv6_ASA_HA_Mode
    scope tenant
    subject Subject
      access-group Contract_PBRv6_ASA_HA_Mode_Filter both
      l4l7 graph PBRv6_ASA_HA_Mode_Graph
      exit
    exit
  vrf context CTX1
    exit
  vrf context CTX2
    exit
```

Step 4 Create a bridge domain for the client and server endpoint group. Both the client and server are in the same VRF instance.

Example:

```
  bridge-domain BD1
    arp flooding
    l2-unknown-unicast flood
    vrf member CTX1
    exit
  bridge-domain BD2
    arp flooding
    l2-unknown-unicast flood
    vrf member CTX1
    exit
```

Step 5 Create a separate bridge domain for the external and internal leg of the firewall.

PBR requires the learning of the source VTEP on remote leaf switches to be disabled, which is done using the **no ip learning** command.

Example:

```
  bridge-domain External-BD3
    arp flooding
    no ip learning
    l2-unknown-unicast flood
    vrf member CTX1
    exit
  bridge-domain Internal-BD4
    arp flooding
    no ip learning
    l2-unknown-unicast flood
```

```
      vrf member CTX1
      exit
```

Step 6 Create the application profile and specify the endpoint groups.

Example:

```
    application AP1
      epg ClientEPG
        bridge-domain member BD1
        contract consumer Contract_PBRv6_ASA_HA_Mode
        exit
      epg ServerEPG
        bridge-domain member BD2
        contract provider Contract_PBRv6_ASA_HA_Mode
        exit
      exit
```

Step 7 Specify the default gateway for the bridge domains.

Example:

```
    interface bridge-domain BD1
      ipv6 address 89:1:1:1::64/64
      exit
    interface bridge-domain BD2
      ipv6 address 99:1:1:1::64/64
      exit

 interface bridge-domain External-BD3
      ipv6 address 10:1:1:1::64/64
      exit
    interface bridge-domain Internal-BD4
      ipv6 address 20:1:1:1::64/64
      exit
```

Step 8 Import the device from tenant T1.

Example:

```
l4l7 cluster import-from T1 device-cluster ifav-asa-vm-ha
```

Step 9 Create the service graph using the service redirect policy.

Example:

```
l4l7 graph PBRv6_ASA_HA_Mode_Graph contract Contract_PBRv6_ASA_HA_Mode
      service N2 device-cluster-tenant T1 device-cluster ifav-asa-vm-ha mode FW_ROUTED svcredir
enable
      connector consumer cluster-interface consumer_PBRv6
          bridge-domain tenant PBRv6_ASA_HA_Mode name External-BD3
          svcredir-pol tenant PBRv6_ASA_HA_Mode name External_leg
          exit
      connector provider cluster-interface provider_PBRv6
          bridge-domain tenant PBRv6_ASA_HA_Mode name Internal-BD4
          svcredir-pol tenant PBRv6_ASA_HA_Mode name Internal_leg
          exit
        exit
    connection C1 terminal consumer service N2 connector consumer
    connection C2 terminal provider service N2 connector provider
    exit
```

Step 10 Create the service redirect policy for the external and internal legs. IPv6 addresses are used in this example; you can also specify IPv4 addresses using the same command.

Example:

```
svcredir-pol Internal_leg
    redir-dest 20:1:1:1::1 00:00:AB:CD:00:11
    exit
svcredir-pol External_leg
    redir-dest 10:1:1:1::1 00:00:AB:CD:00:09
    exit
exit
```

Verifying a Policy-Based Redirect Configuration Using the NX-OS-Style CLI

After you have configured policy-based redirect, you can verify the configuration using the NX-OS-style CLI.

Step 1 Show the running configuration of the tenant.

Example:

```
apic1# show running-config tenant PBRv6_ASA_HA_Mode svcredir-pol
# Command: show running-config tenant PBRv6_ASA_HA_Mode svcredir-pol
# Time: Wed May 25 00:57:22 2016
  tenant PBRv6_ASA_HA_Mode
    svcredir-pol Internal_leg
      redir-dest 20:1:1:1::1/32 00:00:AB:CD:00:11
      exit
    svcredir-pol External_leg
      redir-dest 10:1:1:1::1/32 00:00:AB:CD:00:09
      exit
    exit
```

Step 2 Show the running configuration of the tenant and its service graph.

Example:

```
apic1# show running-config tenant PBRv6_ASA_HA_Mode l4l7 graph PBRv6_ASA_HA_Mode_Graph
# Command: show running-config tenant PBRv6_ASA_HA_Mode l4l7 graph PBRv6_ASA_HA_Mode_Graph
# Time: Wed May 25 00:55:09 2016
  tenant PBRv6_ASA_HA_Mode
    l4l7 graph PBRv6_ASA_HA_Mode_Graph contract Contract_PBRv6_ASA_HA_Mode
     service N2 device-cluster-tenant T1 device-cluster ifav-asa-vm-ha mode FW_ROUTED svcredir enable

        connector consumer cluster-interface consumer_PBRv6

          bridge-domain tenant PBRv6_ASA_HA_Mode name External-BD3

          svcredir-pol tenant PBRv6_ASA_HA_Mode name External_leg

          exit

        connector provider cluster-interface provider_PBRv6

          bridge-domain tenant PBRv6_ASA_HA_Mode name Internal-BD4
          svcredir-pol tenant PBRv6_ASA_HA_Mode name Internal_leg
          exit
        exit
      connection C1 terminal consumer service N2 connector consumer
      connection C2 terminal provider service N2 connector provider
      exit
    exit
```

Step 3 Show the service graph configuration.

Example:

```
apic1# show l4l7-graph graph PBRv6_ASA_HA_Mode_Graph
Graph           : PBRv6_ASA_HA_Mode-PBRv6_ASA_HA_Mode_Graph
Graph Instances : 1

Consumer EPg     : PBRv6_ASA_HA_Mode-ClientEPG
Provider EPg     : PBRv6_ASA_HA_Mode-ServerEPG
Contract Name    : PBRv6_ASA_HA_Mode-Contract_PBRv6_ASA_HA_Mode
Config status    : applied
Service Redirect : enabled

Function Node Name : N2
 Connector   Encap       Bridge-Domain  Device Interface      Service Redirect Policy
 ----------  ----------  ----------     --------------------  -----------------------
 consumer    vlan-241    PBRv6_ASA_HA_  consumer_PBRv6        External_leg
                         Mode-
                         External-BD3
 provider    vlan-105    PBRv6_ASA_HA_  provider_PBRv6        Internal_leg
                         Mode-
                         Internal-BD4
```

About Multi-Node Policy-Based Redirect

Multi-node policy-based redirect enhances PBR by supporting up to five function nodes in a service graph. You can configure which service node connector terminates the traffic and based on this configuration, the source and destination class IDs for the service chain are determined. In the multi-node PBR feature, policy-based redirection can be enabled on the consumer, provider, or both of the service node connectors. It can also be configured for the forward or reverse directions. If the PBR policy is configured on a service node connector, then that connector does not terminate traffic.

About Symmetric Policy-Based Redirect

Symmetric policy-based redirect (PBR) configurations enable provisioning a pool of service appliances so that the consumer and provider endpoint groups traffic is policy-based. The traffic is redirected to one of the service nodes in the pool, depending on the source and destination IP equal-cost multi-path routing (ECMP) prefix hashing.

Note Symmetric PBR configurations require 9300-EX hardware.

Sample symmetric PBR REST posts are listed below:

```
Under fvTenant svcCont

<vnsSvcRedirectPol name="LoadBalancer_pool">
          <vnsRedirectDest name="lb1" ip="1.1.1.1" mac="00:00:11:22:33:44"/>
          <vnsRedirectDest name="lb2" ip="2.2.2.2" mac="00:de:ad:be:ef:01"/>
          <vnsRedirectDest name="lb3" ip="3.3.3.3" mac="00:de:ad:be:ef:02"/>
</vnsSvcRedirectPol>

<vnsLIfCtx name="external">
```

```
          <vnsRsSvcRedirectPol tnVnsSvcRedirectPolName="LoadBalancer_pool"/>
          <vnsRsLIfCtxToBD tDn="uni/tn-solar/bd-fwBD">
</vnsLIfCtx>

<vnsAbsNode name="FW" routingMode="redirect">
```

Sample symmetric PBR NX-OS-style CLI commands are listed below.

The following commands under the tenant scope create a service redirect policy:

```
apic1(config-tenant)# svcredir-pol fw-external
apic1(svcredir-pol)# redir-dest 2.2.2.2 00:11:22:33:44:56
```

The following commands enable PBR:

```
apic1(config-tenant)# l4l7 graph FWOnly contract default
apic1(config-graph)# service FW svcredir enable
```

The following commands set the redirect policy under the device selection policy connector:

```
apic1(config-service)# connector external
apic1(config-connector)# svcredir-pol tenant solar name fw-external
```

Policy Based Redirect and Hashing Algorithms

Note This feature is available in the APIC Release 2.2(3x) release and going forward with APIC Release 3.1(1). It is not supported in APIC Release 3.0(x).

In Cisco APIC, Release 2.2(3x), Policy Based Redirect feature (PBR) supports the following hashing algorithms:

- Source IP address
- Destination IP address
- Source IP address, Destination IP address, and Protocol number (default configuration).

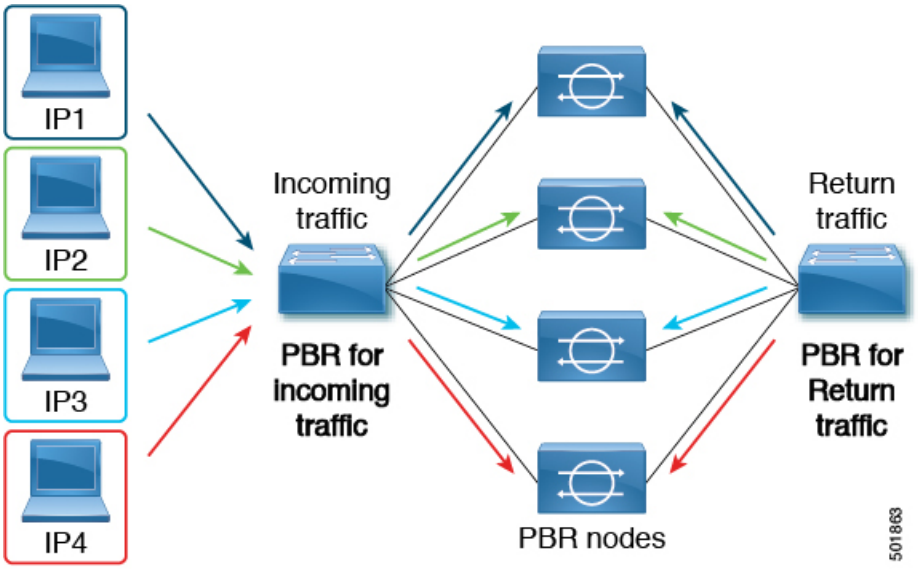
Policy-Based Redirect Resilient Hashing

In symmetric PBR, incoming and return user traffic uses the same PBR node in an ECMP group. If, however, one of the PBR nodes goes down/fails, the existing traffic flows are rehashed to another node. This can cause issues such as existing traffic on the functioning node being load balanced to other PBR nodes that do not have current connection information. If the traffic is traversing a stateful firewall, it can also lead to the connection being reset.

Resilient hashing is the process of mapping traffic flows to physical nodes and avoiding the rehashing of any traffic other than the flows from the failed node. The traffic from the failed node is remapped to a "backup" node. The existing traffic on the "backup" node is not moved.

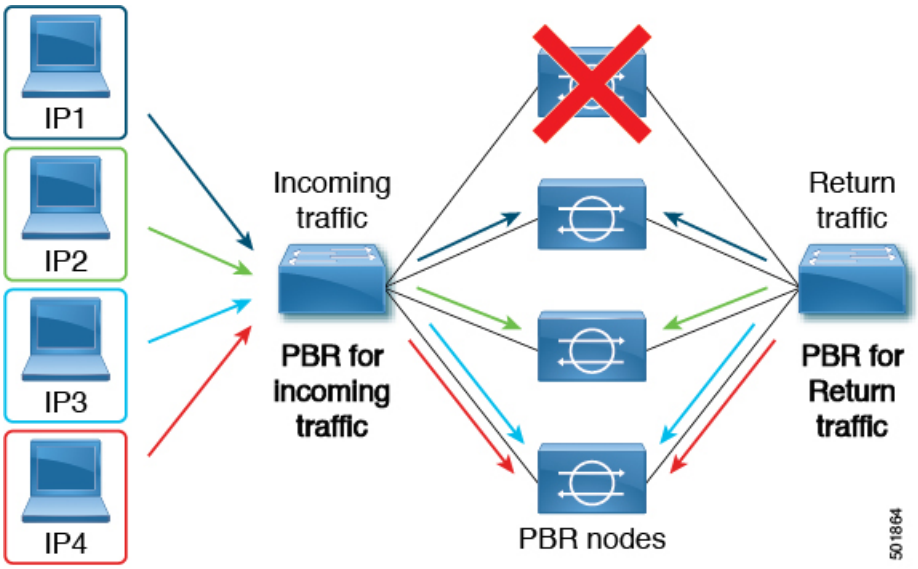
The image below shows the basic functionality of symmetric PBR with incoming and return user traffic using the same PBR nodes.

Figure 15: Symmetric PBR



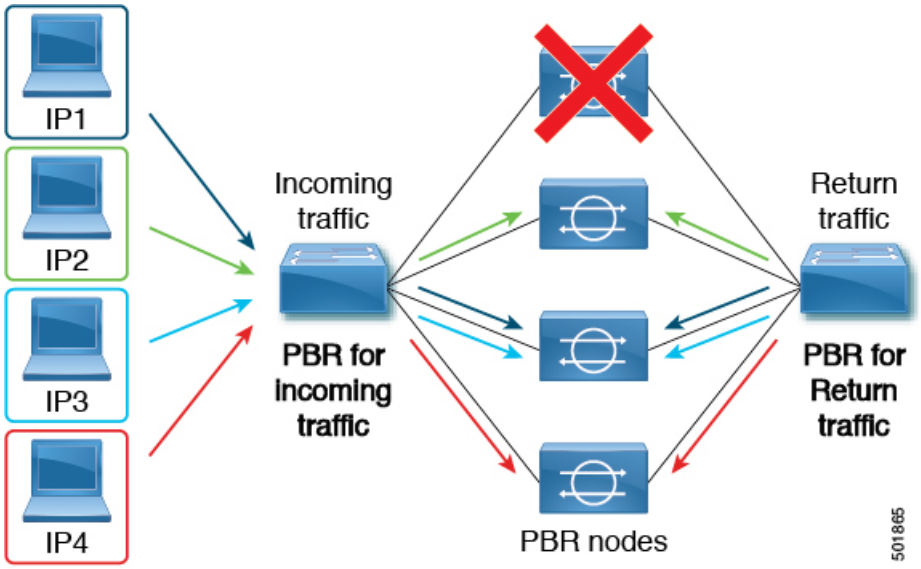
The next image shows what occurs when one of the PBR nodes is disabled or fails. The traffic for IP1 is rehashed to the next node and IP2 and IP3's traffic is load balanced to another PBR node. As stated earlier, this could lead to connectivity interruptions or delays if the other PBR nodes do not have the current connection information for IP2 and IP3 traffic.

Figure 16: Disabled/Failed PBR node without resilient hashing



The final image shows how this same use case is addressed when resilient hashing is enabled. Only the user traffic from the disabled/failed node is moved. All other user traffic remains on their respective PBR nodes.

Figure 17: Disabled/Failed PBR node with resilient hashing



If the node returns to service, the traffic flows rehashed from the failed node to the active node are returned to the reactivated node.

Note Adding or deleting PBR nodes from the ECMP group can cause all the traffic flows to be rehashed.

Enabling Resilient Hashing in L4-L7 Policy-Based Redirect

Before you begin

This task assumes that an L4-L7 Policy Based Redirect policy has been created.

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

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

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Policies** > **Protocol** > **L4-L7 Policy Based Redirect** > ***L4-L7_PBR_policy_name***.

Step 4 In the Work pane, check the **Resilient Hashing Enabled** check box.

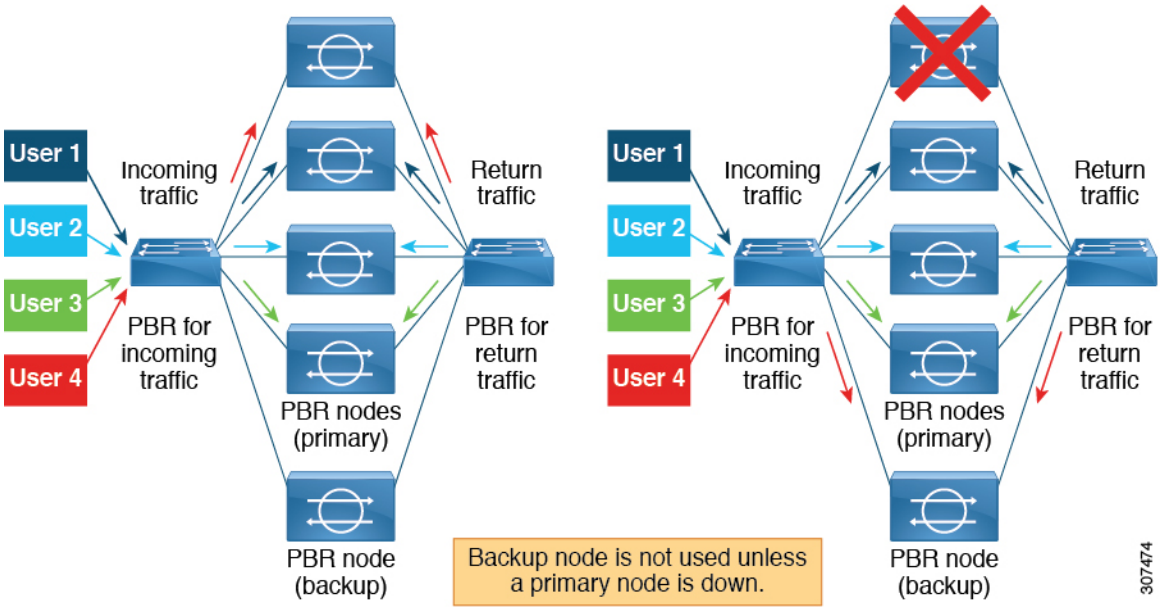
Step 5 Click **Submit**.

About PBR Backup Policy

Prior to Cisco APIC Release 4.2(1), all Policy-Based Redirect (PBR) destinations in a PBR policy are used as long as the PBR destination is functioning. If one of the PBR nodes fails, the existing traffic flows are rehashed. This could lead to the connection being reset if, for example, the data paths are traversing stateful

firewalls. With Resilient Hash PBR, only the traffics that went through failed node is directed to one of the available nodes, which could cause an overload of traffic on the newly shared node. Instead of sharing one of the available nodes, a backup node in the group can be configured and used to absorb the traffic load. You can configure multiple backup PBR destinations per PBR backup policy.

Beginning with Cisco APIC Release 4.2(1), a new PBR Backup Policy option is available:



With resilient hash, only the traffic that went through the failed nodes gets rerouted to one of the available nodes. With resilient hash and PBR backup policy, the traffic that went through the failed primary node gets rerouted to one of the available backup nodes.

Backup Policy Guidelines and Limitations

Follow these guidelines and limitations for the PBR Backup Policy option:

- The PBR backup policy option is supported only on new generation leaf switches, which are switch models with "-EX", "-FX" or "-FX2" at the end of the switch name.
- Resilient hashing must be enabled.
- Starting from Cisco APIC Release 5.0(1), Layer 1/ Layer 2 PBR also supports backup policy.
- A destination can be used as a PBR destination or backup PBR destination, not both (a primary PBR destination can’t be used as a backup PBR destination in the same or different PBR policy).
- One backup PBR policy can be used by only one PBR policy. If you attempt to add a second backup policy to a PBR policy, the configuration will be rejected.

  If you want to use same backup PBR destination for multiple PBR policies, create two different backup PBR policies using the same backup PBR destination. The destinations in both these policies must have the same health group configured.
- With resilient hash and PBR backup policy:
  - The traffic that went through the failed nodes goes to a backup node in the PBR backup policy, in the order of IP address from lowest to highest. When multiple primary nodes fail and all the backup

nodes are used then the traffic that went through the failed node is routed to one of the available nodes, including primary and backup nodes, in the order of IP address from lowest to highest. For example, assume there are four primary nodes (192.168.1.1 to 192.168.1.4) and two backup nodes (192.168.1.5 and 192.168.1.6):

- When a primary node with IP address 192.168.1.1 failed, the traffic that went through this node is routed to an available backup node with the lowest IP address 192.168.1.5.
- When two primary nodes with IP addresses 192.168.1.1 and 192.168.1.2 failed, the traffic that went through 192.168.1.1 is routed to the backup node 192.168.1.5 and the traffic that went through 192.168.1.2 is routed to the backup node 192.168.1.6.
- When three primary nodes with IP addresses 192.168.1.1, 198.168.1.2, and 192.168.1.3 failed and only one backup node 192.168.1.5 is available, the traffic that went through the first failed node 192.168.1.1 is routed to the backup node 192.168.1.5.
  - For second failed primary node 192.168.1.2, compare the IP address (192.168.1.1) that the backup node is used for and the IP address of available primary node 192.168.1.4, since 192.168.1.1 is smaller than the first available primary node 192.168.1.4, the traffic that went through the failed node 192.168.1.2 is routed again to the backup node 192.168.1.5.
  - For third failed node 192.168.1.3, since the backup node is already in use, the traffic that went through the third failed node is routed to the available primary node 192.168.1.4.

- When pod aware PBR is enabled, for a failed primary node, the traffic that went through the failed node first goes to an available local backup node. If a backup node is not available, then a local primary node is preferred. When all local primary nodes and local backup nodes failed and therefore no local node is available, then the traffic that went through the failed node goes to a remote primary node, and then to a remote backup node. For example:
  - When both primary nodes and backup nodes are in the same pod, and pod aware PBR is enabled, for a failed primary node in local pod, the traffic that went through the failed node goes to a backup node in the same local pod.
  - When there are local primary nodes and local backup nodes, and pod aware PBR is enabled, for a failed local primary node and failed local backup node, the traffic that went through the failed node goes to another primary node in different pod.

Creating a PBR Backup Policy

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

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

Step 3 In the Navigation pane, choose **Tenant** > ***tenant_name*** > **Policies** > **Protocol** > **L4-L7 Policy Based Redirect Backup**.

Step 4 Right-click **L4-L7 Policy Based Redirect Backup**, and then click **Create L4-L7 Policy-Based Redirect Backup**.

The **Create PBR Backup Policy** dialog appears.

Step 5 In the **Name** field, enter a unique name for the backup policy.

Step 6 In the **L3 Destinations** table, click +.

The **Create Destinations of Redirected Traffic** dialog appears.

a) In the **IP** field, enter the IP address of the Layer 3 destination node.
b) In the **MAC** field, enter the MAC address of the Layer 3 destination node.
c) Optional: In the **Second IP** field, enter a secondary IP address for the Layer 3 destination node.
d) In the **Redirect Health Group** field, select an existing health group or create a new one.

For more information on creating a new redirect health group, see Configuring a Redirect Health Group Using the GUI, on page 120.

e) Click **OK**.

Optional: Repeat steps a through e to add more Layer 3 destinations.

Step 7 Click **Submit**.

Enabling a PBR Backup Policy

Before you begin

This task assumes that a Layer 4 to Layer 7 policy-based redirect (PBR) policy has been created.

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

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

Step 3 In the Navigation pane, choose **Tenant** > ***tenant_name*** > **Policies** > **Protocol** > **L4-L7 Policy Based Redirect** > ***L4-L7_PBR_policy_name***.

Step 4 In the **Destination Type** field, choose **L3**.

Step 5 In the **IP SLA Monitoring Policy** field, select an existing policy or create a new IP SLA monitoring policy.

For more information on creating a new IP SLA monitoring policy, see the *Cisco APIC Layer 3 Networking Configuration Guide*.

Step 6 Check the **Resilient Hashing Enabled** box.

Step 7 In the **Backup Policy** field, choose an existing policy or create a new backup policy.

For more information on creating a new PBR backup policy, see Creating a PBR Backup Policy, on page 94.

Step 8 Make sure at least one active PBR destination appears in the **L3 Destinations** table and is configured with a redirect health group.

For more information on creating a new redirect health group, see Configuring a Redirect Health Group Using the GUI, on page 120.

Step 9 Click **Submit**.

About the Bypass Action

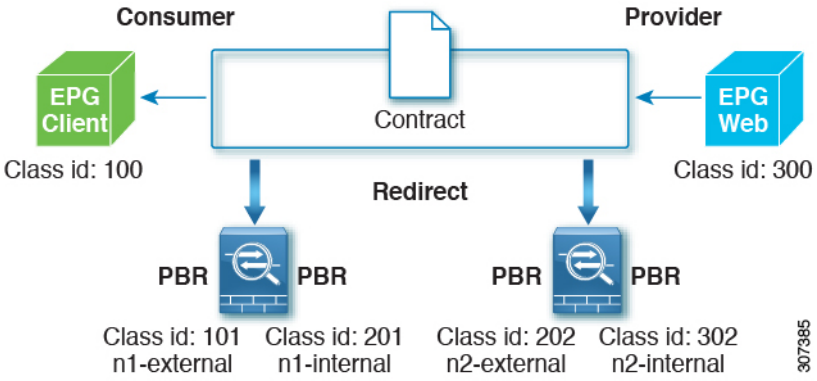
Prior to Cisco Application Policy Infrastructure Controller (APIC) release 4.1(2), when you choose Threshold Enable when creating a Layer 4 to Layer 7 services policy-based redirect, only two options were available: **deny action** or **permit action**.

With these two options, in a multi-node policy-based redirect graph, when one node crosses the low threshold, the following action would occur, depending on which of the two options you selected:

- **deny action**: Traffic is dropped at this node.
- **permit action**: Traffic is sent directly to the destination, and the rest of the service chain is skipped.

Beginning with Cisco APIC release 4.1(2), a new **bypass action** option is available. With this option, in a multi-node policy-based redirect graph, when one node crosses the low threshold, traffic is still able to proceed through the rest of the service chain that is either up or cannot be bypassed.

The following sections describe how traffic is handled for each of these three options using this example two-node policy-based redirect graph.



When both nodes are up, this two-node policy-based redirect behaves in the following manner:

Source EPG	Destination EPG	Action
100	300	PBR to n1-external
201	300	PBR to n2-external
302	300	permit
300	100	PBR to n2-internal
202	100	PBR to n1-internal
101	100	permit

The following sections describe how the two-node policy-based redirect behaves when the first node goes down, based on the option that you select in the **Threshold Down Action** field.

deny action

Using the example configuration described above, if you select **deny action** in the **Threshold Down Action** field and the first node goes down, the PBR policies that use the first node are updated to "Drop," and communication between the client EPG and the Web EPG will be dropped, as shown in the following table.

Source EPG	Destination EPG	Action
100	300	**Drop**
201	300	PBR to n2-external
302	300	permit
300	100	PBR to n2-internal
202	100	**Drop**
101	100	permit

permit action

Using the example configuration described above, if you select **permit action** in the **Threshold Down Action** field and the first node goes down, the PBR policies that use the first node are updated to "Permit." Traffic from the client EPG to the Web EPG (from 100 to 300) proceeds directly, without the service node. Return traffic from the Web EPG to the client EPG (from 300 to 100) is redirected to n2-internal, as shown in the following table; however, the second node might drop the packet because it is an asymmetric flow.

Source EPG	Destination EPG	Action
100	300	**Permit**
201	300	PBR to n2-external
302	300	permit
300	100	PBR to n2-internal
202	100	**Permit**
101	100	permit

bypass action

Beginning with Cisco APIC release 4.1(2), if you select the new **bypass action** option in the **Threshold Down Action** field and the first node goes down, the PBR policies that use the first node are updated to "PBR to next device". In this case, the following occurs:

- Traffic from the Client EPG to the Web EPG (from 100 to 300) is redirected to n2-external.
- Return traffic from the Web EPG to the Client EPG (from 300 to 100) is redirected to n2-internal.
- Return traffic from n2-external to consumer is set to "Permit."

Source EPG	Destination EPG	Action
100	300	**PBR to n2-external**
201	300	PBR to n2-external
302	300	permit
300	100	PBR to n2-internal
202	100	**Permit**
101	100	permit

Guidelines and Limitations

Following are the guidelines and limitations for the **bypass action** option:

- The **bypass action** option is supported only on new generation ToR switches, which are switch models with "EX," "FX," or "FX2" at the end of the switch name.
- The **bypass action** option is not needed on a one-node service graph. If bypass is configured in such a case, forwarding behavior is the same as permit action.
- L3Out EPGs and regular EPGs can be consumer or provider EPGs.
- A service node that has NAT enabled cannot be bypassed, as that will break the traffic flow.
- Beginning with the 5.0(1) release, Layer 1/Layer 2 PBR supports the bypass action.
- The **bypass action** option is not supported in the following cases:
  - Layer 4 to Layer 7 devices in one-arm mode.
  - Remote leaf switches.
- Do not use the same PBR policy in more than one service graph if bypass action is enabled. Cisco APIC will reject configurations if the same PBR policy with bypass action is used in multiple service graphs. To avoid this, configure different PBR policies that use the same PBR destination IP address, MAC address and Health Group.

Configuring the Threshold Down Action in Policy-Based Redirect

Before you begin

This task assumes that a Layer 4 to Layer 7 services policy-based redirect (PBR) policy has been created.

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

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

Step 3 In the Navigation pane, choose **Tenant** > ***tenant_name*** > **Policies** > **Protocol** > **L4-L7 Policy Based Redirect** > ***L4-L7_PBR_policy_name***.

Step 4 In the **Destination Type** field, select **L3**.

Step 5 In the **IP SLA Monitoring Policy** field, select an existing policy or create a new IP SLA monitoring policy.

For more information on creating a new IP SLA monitoring policy, see the *Cisco APIC Layer 3 Networking Configuration Guide*.

Step 6 Check the **Threshold Enable** check box.

The following fields appear:

- Min Threshold Percent (percentage)
- Max Threshold Percent (percentage)
- Threshold Down Action

Step 7 Select the minimum and maximum threshold percentage.

For more information on the minimum and maximum thresholds, see Policy-Based Redirect and Threshold Settings for Tracking Service Nodes, on page 117.

Step 8 In the **Threshold Down Action** area, select the threshold down action.

The options are:

- **bypass action**
- **deny action**
- **permit action**

Step 9 Click **Submit**.

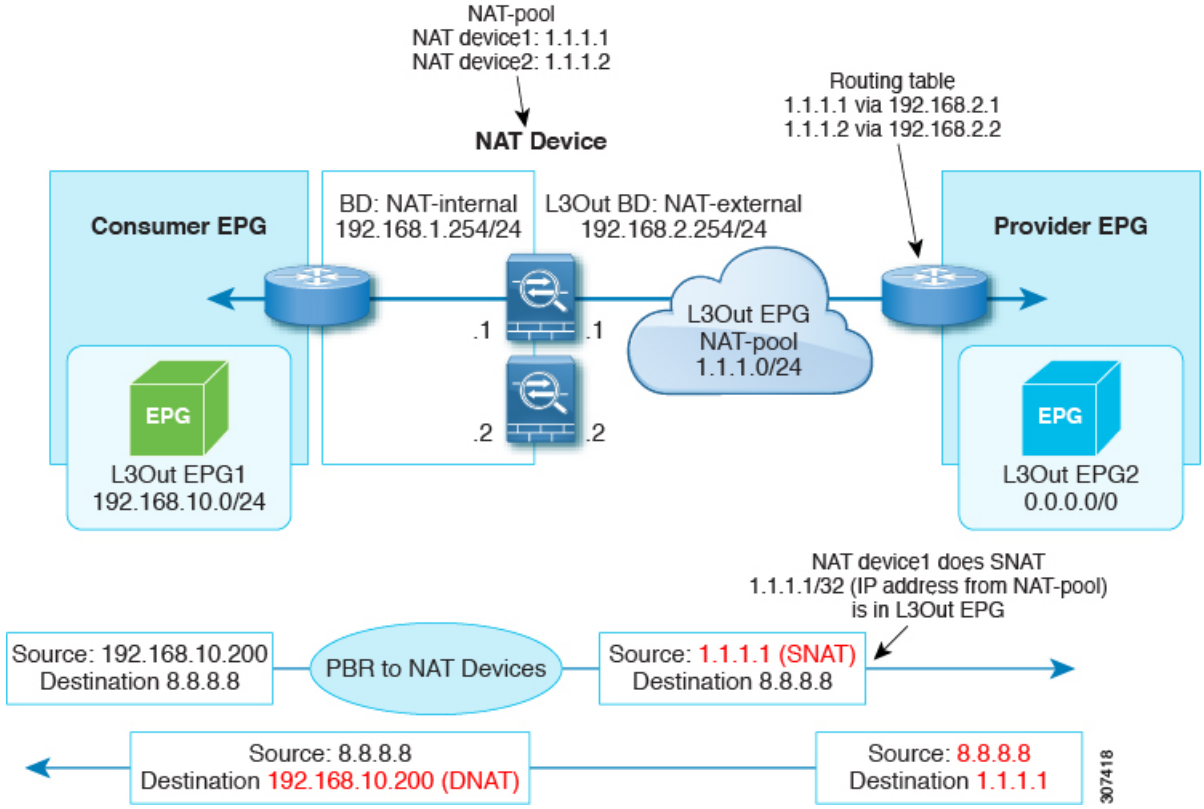
Policy-Based Redirect with an L3Out

Beginning with Cisco Application Policy Infrastructure Controller (APIC) release 4.1(2), you can use an L3Out to connect a Layer 4 to Layer 7 services device that is part of a service graph. There are multiple ways to use an L3Out as part of a policy-based redirect (PBR) service graph:

- Using PBR to redirect only to the consumer interface of the Layer 4 to Layer 7 services device, while the provider interface of the Layer 4 to Layer 7 services device is connected to an L3Out. This is referred to as "uni-directional" PBR, because PBR is done only for one direction of the traffic. This option was introduced in Cisco APIC release 4.1(2).
- Using PBR to redirect only to the provider interface of the Layer 4 to Layer 7 services device, while the consumer interface of the Layer 4 to Layer 7 services device is connected to an L3Out. This option was introduced in Cisco APIC release 5.0(1), which is also a uni-directional PBR design, and it is the symmetric design of the one described in the previous bullet.
- Using PBR to redirect a Layer 4 to Layer 7 services device interface that is connected to an L3Out. This option was introduced in Cisco APIC release 5.2(1).

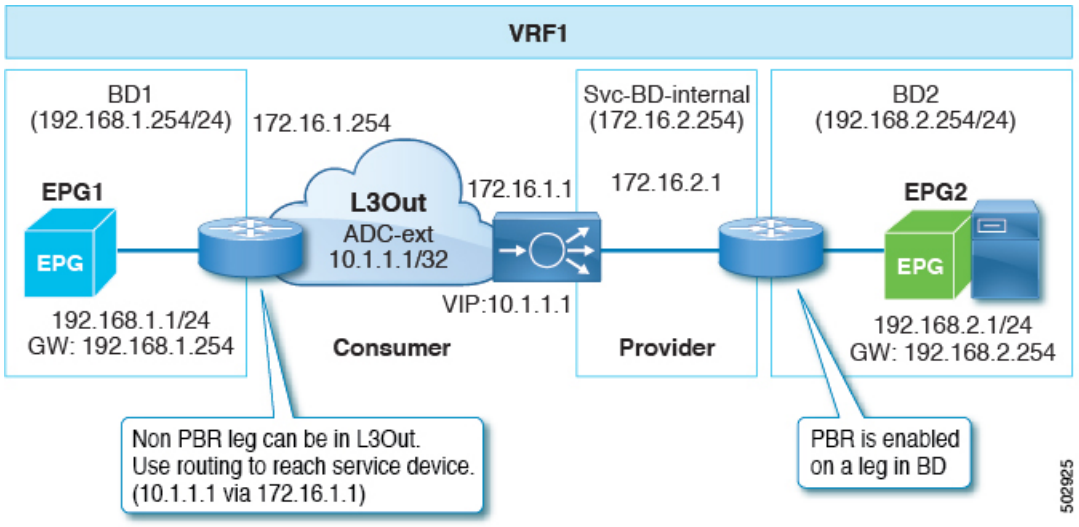
These use cases are described in greater detail in the text that follows.

As mentioned in the first bullet, beginning with Cisco APIC release 4.1(2), you can configure unidirectional PBR to a consumer interface, and can connect the provider interface to an L3Out as shown in the following illustration:

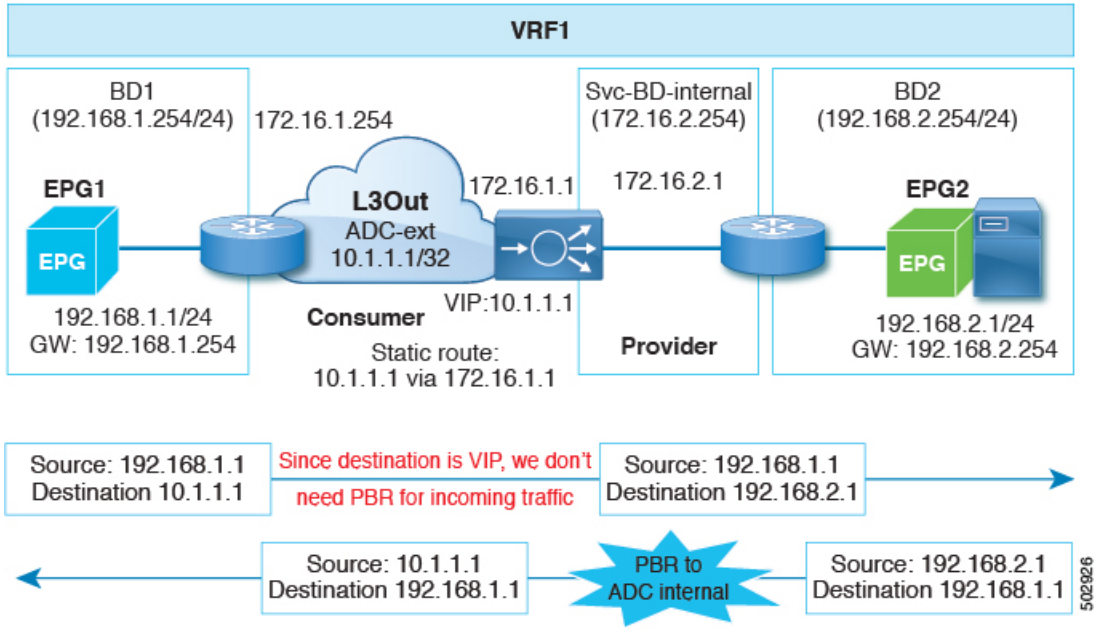


In the example, PBR is enabled in the consumer connector in the bridge domain, but PBR is not enabled on the provider connector in the L3Out. This design is supported only when L3Out is the provider connector of the last service node. Prior to the Cisco APIC 4.1(2) release, if PBR was configured to redirect traffic to a node of a service graph, both the consumer and provider connectors of the Layer 4 to Layer 7 services device had to be in a bridge domain, even in the case of uni-directional PBR.

Beginning with Cisco APIC release 5.0(1), uni-directional PBR is supported with the other connector in an L3Out, regardless if the L3Out is the provider or consumer connector and regardless if the L3Out is the last node or not. This includes the case where the load balancer has a VIP address outside of the local subnet on the consumer side of the service node, as shown in the following illustration.

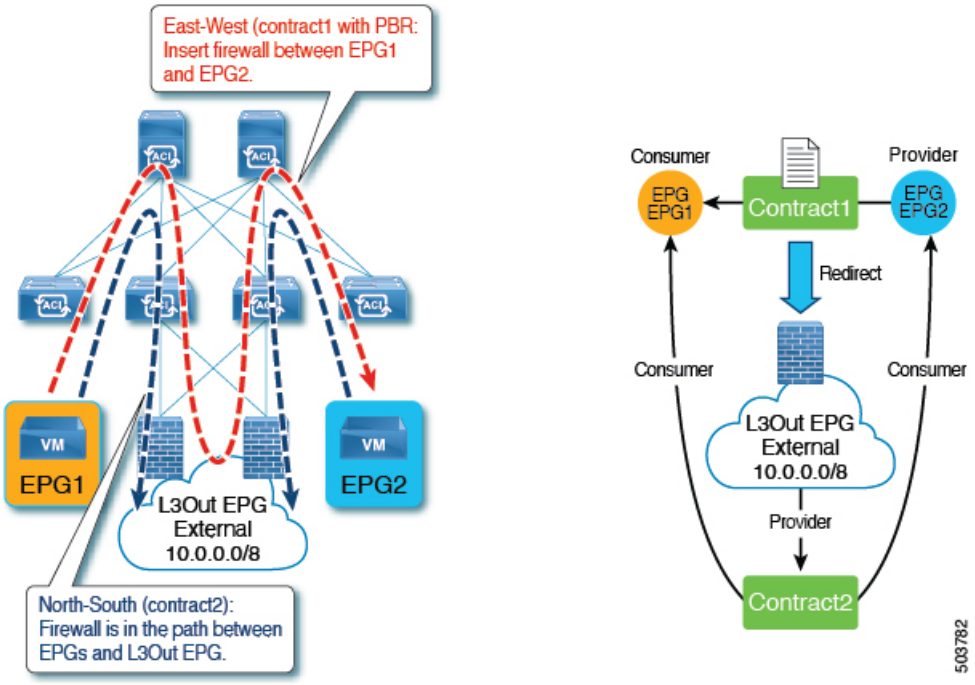


In the following illustration example, the incoming traffic from the consumer endpoint to the VIP address is forwarded to the load balancer that is connected to the L3Out, based on the routing table. Then, the traffic is forwarded to the provider endpoint. The return traffic from the provider endpoint to the consumer endpoint is redirected to the provider side of the service node because of PBR.

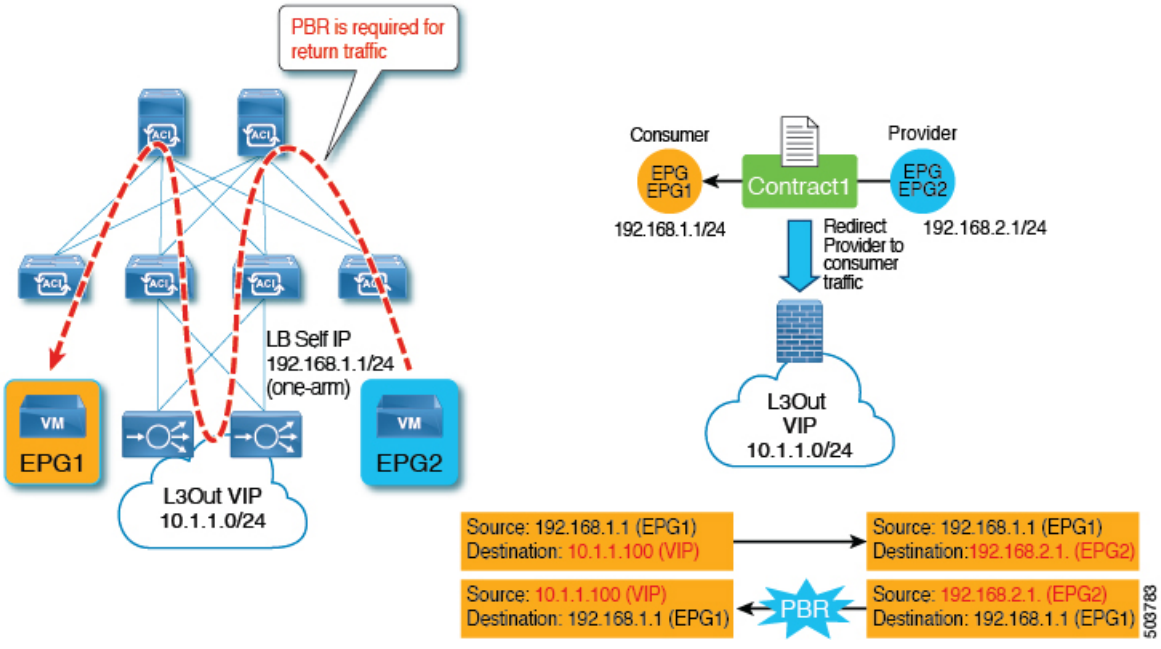


Beginning with Cisco APIC release 5.2(1), the Layer 4 to Layer 7 services device that is used as a destination of the PBR policy can have the interfaces in an L3Out. Prior to this release, a PBR policy's destination interface could only be in a bridge domain. Some common use cases include:

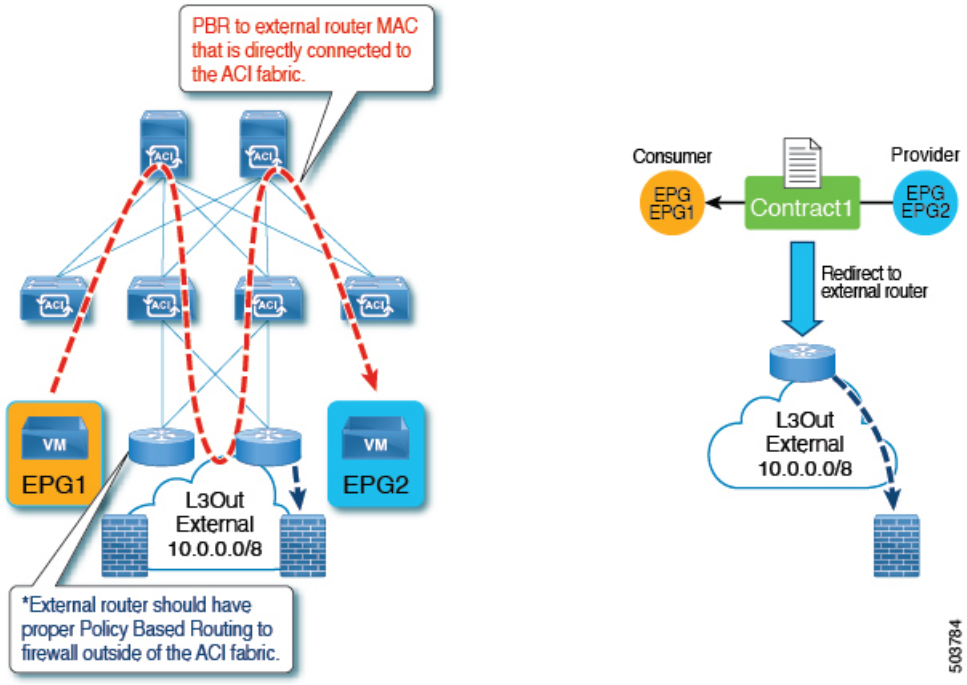
- You can use the same firewall for both East-West and North-South traffic. In this case, the firewall internal leg is connected to the Cisco Application Centric Infrastructure (ACI) fabric, while the firewall external leg is outside of the Cisco ACI fabric.



- You can have a one-arm load balancer with a VIP address that is outside of the local subnet. In this case, the VIP address is outside of the load balancer's self IP address subnet. The load balancer does not perform Source Network Address Translation (SNAT), and thus PBR is required for the return traffic.



- You can redirct traffic to a device, such as an external firewall, that is not directly connected to Cisco ACI.



Guidelines and Limitations for Policy-Based Redirect with an L3Out

The following guidelines and limitations are for policy-based redirect (PBR) with an L3Out:

- Both the one-arm and two-arm modes are supported.
- You cannot mix PBR on a bridge domain and PBR on an L3Out on the same function node in the service graph. For example:
  - You cannot configure the consumer connector of N1 in BD1 (PBR is enabled) and the provider connector of N1 in an L3Out1 (PBR is enabled).
  - But, you can configure the consumer connector of N1 in BD1 (PBR is not enabled) and the provider connector of N1 in an L3Out1 (PBR is enabled).
- An L3Out with a switch virtual interface (SVI), routed sub-interface, or routed interface is supported.
- You cannot use an infra L3Out, GOLF L3Out, SDA L3Out, or L3Out using a floating SVI for the PBR destination.
- Use a specific L3Out EPG subnet if there are other L3Out EPGs in the same VRF instance; otherwise, the other L3Outs might be used for EPG classification by mistake.
- An L3Out EPG with 0.0.0.0/0 or 0::0 cannot be used for the L3Out EPG for PBR destinations. This is because east-to-west traffic must be classified with the automatically created service-EPG. Hence, if the L3Out EPG is configured with 0.0.0.0/0, east-to-west traffic would be classified as coming from the outside.
- If the service device is in two-arm mode and one of the L3Outs for the service device connectors learns 0.0.0.0/0 or 0::0, both arms must be connected to the same leaf switch or the same vPC pair.

- If the consumer/provider EPG is an L3Out EPG, it cannot be under the service leaf switch where the L3Out for the PBR destination resides. This is a hardware limitation.
  - The leaf switch cannot figure out if a packet is coming from the consumer/provider L3Out EPG or back from the service device even if they use specific L3Out EPG subnets.

    If the consumer/provider EPG is a regular EPG, not an L3Out EPG, the consumer, provider, and service device L3Outs can be under the same leaf switch.

- When deploying PBR in the two-arm mode with the service device behind an L3Out and using the OSPF or EIGRP protocol for next-hop connectivity, deploying both arms on the same service leaf switch is not supported. Deploying each arm on different service leaf switch is supported.
- When deploying PBR in the two-arm mode and the service node L3Out with the OSPF, EIGRP, or BGP protocol, you must control next-hop for the service device properly on each arm.
- The following table shows the supported consumer/provider EPG type combinations:

Table 2: Supported consumer/provider EPG type combinations

Consumer/Provider	EPG	L3Out	ESG
EPG	Supported	Supported	Not supported[1]
L3Out	Supported	Supported	Supported
ESG	Not supported	Supported	Supported

[1] An EPG-to-ESG contract is not supported even without a service graph.

- An intra-EPG/ESG/L3Out EPG contract with PBR is supported.
  - Beginning with release 5.2(1), an intra-L3Out EPG contract is supported.
- When using a service graph with PBR with a bridge domain, Cisco ACI automatically creates a hidden EPG called a *service EPG*. Cisco ACI configures contracts between the service EPG and user-created EPGs to allow the traffic path as defined by the service graph. When connecting a Layer 4 to Layer 7 services device interface to an L3Out and using this interface as a PBR destination for a service graph, Cisco ACI creates a service EPG automatically, but the administrator must also create the L3Out EPG in addition to the service EPG. Some of the traffic is forwarded to the Layer 4 to Layer 7 interface using PBR, while other traffic, such as keepalives with load balancers, must be sent using regular traffic forwarding (routing). To enable the communication between the L3Out EPG used by an Layer 4 to Layer 7 services device and an the EPG where the endpoints are, as you need to do in the case of a load balancer keepalives, you must configure **Direct Connect** and you must also configure a contract between the L3Out EPG and the EPG where the servers are.
- Tracking is mandatory for PBR destinations in an L3Out for better convergence.
- The bypass feature is not supported for one-arm mode, which is also applicable to a PBR destination on a bridge domain.
- Multi-node PBR is supported.
- Active-active symmetric PBR is supported.
- Tracking, threshold down action is supported.

- Resilient hashing is supported.
- N+M redundancy is supported.
- A single pod, Cisco ACI Multi-Pod, or remote leaf switch is supported.
- Cisco ACI Multi-Site is not supported.
- For inter-VRF contracts without an endpoint security group (ESG), if the PBR L3Out destination is in the provider VRF instance:
  - You must leak the L3Out EPG subnet used by the service device to the consumer VRF instance. If you do not, the consumer VRF instance does not have the route to the PBR destination and the provider VRF instance does not have a permit rule for the traffic from the PBR destination in the provider VRF instance to the consumer EPG. If the PBR destination is in a bridge domain, you do not need to leak the service bridge domain for the PBR destination to the consumer VRF instance.
- For inter-VRF contracts with an ESG, for ESG-to-ESG or ESG-to-L3Out, with or without PBR:
  - You must leak the consumer ESG or L3Out subnet to the provider VRF instance, and you must leak the provider ESG or L3Out subnet to the consumer VRF instance. In addition, if you are using PBR:
    - If the PBR destination is in a bridge domain, you do not need to leak the service device subnet.
    - If the PBR destination in an L3Out, you must leak the L3Out EPG subnet used by the service device to the other VRF instance regardless if the L3Out EPG is in the consumer or provider VRF instance.
- To leak the L3Out EPG subnet, modify the subnet's properties and enable **Shared Route Control Subnet** and **Shared Security Import Subnet**. If needed, also enable **Aggregate Shared Routes**.
- Internal VRF instances will be created on a border leaf switch that has an L3Out toward the PBR destination (the VRF is created under the same tenant). An internal VRF instance is created per PBR destination in the PBR policy.
  - For example, if PBR-policy1 has 3 destinations, then 3 VRF instances are created in the PBR policy. If you reuse PBR-policy1 by multiple contracts, only 3 VRF instances are created.
  - There is no VRF scale impact on the consumer/provider leaf switches.
- Ensure that the L3Out belongs either to the consumer or provider VRF instance.

Configuring Policy-Based Redirect with an L3Out Using the GUI

The configuration steps for policy-based redirect (PBR) with an L3Out are mostly the same as a typical policy-based redirect configuration, except for a few differences.

Before you begin

Create the necessary tenant, VRF instance, EPGs, bridge domains for the EPGs, and service bridge domains.

Step 1 Create a Layer 4 to Layer 7 device. If the PBR destination is in an L3Out, then for the concrete interface, the path should match with the path used in the L3Out logical interface.

See Configuring a Layer 4 to Layer 7 Services Device Using the GUI, on page 9.

When using PBR with an L3Out in conjunction with a Layer 4 to Layer 7 services virtual appliance that peers with the L3Out, you must configure the path of the virtualized host intefaces as part of the concrete device configuration. The path used in the Layer 4 to Layer 7 services concrete device configuration and by the L3Out configuration must match. This is because the floating L3Out feature is not yet integrated with the service graph, hence Cisco Application Centric Infrastructure (ACI) must be configured with the path information.

Step 2 Create a service graph template.

See Configuring a Service Graph Template Using the GUI, on page 44.

Step 3 Configure an IP SLA monitoring policy.

See the chapter about IP SLAs in the *Cisco APIC Layer 3 Networking Configuration Guide* at the following site:

https://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

Step 4 Create a PBR policy.

See Configuring Policy-Based Redirect Using the GUI, on page 83.

To enable tracking, you must configure a redirect health group. See Configuring a Redirect Health Group Using the GUI, on page 120.

Step 5 Create an L3Out and an L3Out EPG (or external EPG).

Do not use 0.0.0.0/0 and make sure to include the subnet address of the firewall or the load balancer, as well as the subnets of the external traffic.

See the *Cisco APIC Layer 3 Networking Configuration Guide* at the following site:

https://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

Step 6 Create a device selection policy.

See Creating a Device Selection Policy Using the GUI, on page 35.

While following the procedure, substitute the following substeps as appropriate:

a) For the **Associated Network** buttons, choose **Bridge Domain** or **L3Out**.

If the destination of the PBR policy is an interface in an L3Out, you must choose **L3Out**.

b) If you chose **Bridge Domain**, then for the **Bridge Domain** drop-down list, choose the bridge domain of the target interface. If you chose **L3Out**, then for the **L3Out** drop-down list, choose the L3Out EPG of the target interface.

c) If necessary, for the **L4-L7 Policy-Based Redirect** drop-down list, choose the appropriate PBR policy.

If the destination of the PBR policy is an interface in an L3Out, you must choose a PBR policy.

d) Configure the remainder of the device selection policy as necessary.

Step 7 Apply the service graph where you attached the service graph to the contract.

See Applying a Service Graph Template to Endpoint Groups Using the GUI, on page 46.

PBR Support for Service Nodes in Consumer and Provider Bridge Domains

Starting with the Cisco APIC 3.1(1) release, bridge domains (BDs) that contain a consumer or provider also support service nodes. Therefore, you are not required to provision separate PBR bridge domains any longer.

The Cisco Nexus 9300-EX and 9300-FX platform leaf switches support this feature.

About Layer 1/Layer 2 Policy-Based Redirect

Using a Layer 1 device is typically referred to as *inline mode* or *wire mode* and is used for firewalls and intrusion prevention systems (IPS) if the service device is expected to perform security functions that are not participating in Layer 2 or Layer 3 forwarding.

Using a Layer 2 device is typically referred to as *transparent mode* or *bridged mode* and is used for firewalls and IPS.

Using a Layer 3 device is typically referred to as *routed mode* and is used for router firewalls and load balancers.

Prior to Cisco Application Policy Infrastructure Controller(APIC) release 4.1, PBR could be configured to redirect traffic to a Layer 4 to Layer 7 services device configured only in Layer 3 device (Go-To) mode. If the Layer 4 to Layer 7 services device is a Layer 1 or Layer 2 device, such as a transparent firewall, PBR could not be used. You could only deploy a Layer 4 to Layer 7 services device operating in Layer 1 or Layer 2 mode by using a service graph and defining the Layer 4 to Layer 7 services device in **Go-Through** mode.

Beginning with Cisco APIC release 4.1, PBR can be configured to redirect traffic to a Layer 4 to Layer 7 services device configured in the Layer 1/Layer 2 device mode as well. PBR can be used with inline IPS or a transparent firewall, in addition to a routed mode firewall.

As part of the Layer 1/Layer 2 PBR feature, the Cisco APIC can verify whether the Layer 4 to Layer 7 services device is forwarding traffic by using Layer 2 ping packets for link layer tracking.

Unlike the **Go-Through** mode, which can also forward non-IP address traffic, Layer 1/Layer 2 PBR is applicable only to IP address traffic.

Layer 1/Layer 2 PBR Configuration Overview

The following list summarizes some of the key Layer 1/Layer 2 PBR configuration concepts:

- Deploying a Layer 4 to Layer 7 services device with Layer 1/Layer 2 PBR requires the configuration of two service bridge domains, one for the consumer-side and one for the provider-side, unlike regular PBR, these bridge domains cannot be the same as the bridge domains where the endpoints (consumer or provider) are configured.
- The service bridge domains must be unicast routing enabled.

- The physical Layer 4 to Layer 7 services device can be connected with individual links or with VPC to the leaf switches.
- With a Layer 1 device, the consumer side VLAN and the provider side VLAN are the same but on different bridge domains, hence the consumer and provider-side of the Layer 4 to Layer 7 services device must be connected to different physical leafs.
- When the Layer 4 to Layer 7 services device is configured as a Layer 1 or Layer 2 device, it doesn't have an IP address on the interface where it receives and sends traffic, hence the redirect policy is defined by entering the leaf/port and VLAN where it is connected to.
- The redirect policy configuration requires only the definition of the leaf/port and VLAN, entering a MAC address is optional. If the MAC field is left empty, Cisco Application Centric Infrastructure (ACI) generates dynamically one MAC address that is used to rewrite the destination MAC address when sending to the Layer 4 to Layer 7 services device on the service bridge domain. These MAC addresses are not the Layer 4 to Layer 7 services device MAC addresses. They are virtual MAC addresses that Cisco ACI uses to rewrite the destination MAC address of the traffic.
- If the Layer 4 to Layer 7 services device is deployed in Layer 2 mode, it must be configured statically to forward the MAC addresses that PBR uses to forward traffic to the Layer 4 to Layer 7 services device. One MAC address identifies the consumer-to-provider destination MAC address used on the service bridge domain and the other MAC address defines the provider-to-consumer destination MAC address used on the other service bridge domain.

  These MAC addresses can be entered manually by the user in APIC as part of the redirect policy definition, or they are auto-generated if the field is left empty. The admin has to add these MAC addresses to the MAC address table of the Layer 4 to Layer 7 services device and be associated with the provider-side port for the MAC address used in the consumer-to-provider direction and with the consumer-side port for the provider-to-consumer direction.
- If an intermediate switch is in between leaf and the Layer 4 to Layer 7 services device deployed in Layer 1/Layer 2 mode, the intermediate switch also needs to forward the traffic destined to the rewritten destination MAC.
- Layer 1/Layer 2 PBR is based on forwarding to a leaf/port/VLAN, hence it can only be deployed with physical domains not with VMM domains. If you need to deploy Layer 1/Layer 2 PBR with a virtual appliance, that must be configured with a physical domain.
- From a high availability perspective, the Layer 4 to Layer 7 services device is deployed in active/standby mode and Cisco ACI has to perform tracking in order to verify which path (leaf/port) is active and which one is standby. Tracking is mandatory for multiple service devices in a Layer 4 to Layer 7 services logical device cluster.
- For Layer 1/Layer 2 PBR tracking, Layer 2 ping is used. The IP SLA type is Layer 2 ping.
- Layer 2 ping, `ethertype 0x0721` is exchanged between leaf nodes, which is going through the service device. Thus, the Layer 1/Layer 2 device needs to permit `ethertype 0x0721`.
- Layer 1/Layer 2 policy-based redirect is not supported from the CLI.
- Layer 1/Layer 2 PBR active-active PBR destinations cannot be connected to a remote leaf switch, as flood in encapsulation is not supported on a remote leaf switch. Provider and consumer service nodes can still be connected to remote leaf switch.
- Dynamic VLAN allocation is not supported.

Active/Standby Layer 1/Layer 2 PBR Design Overview

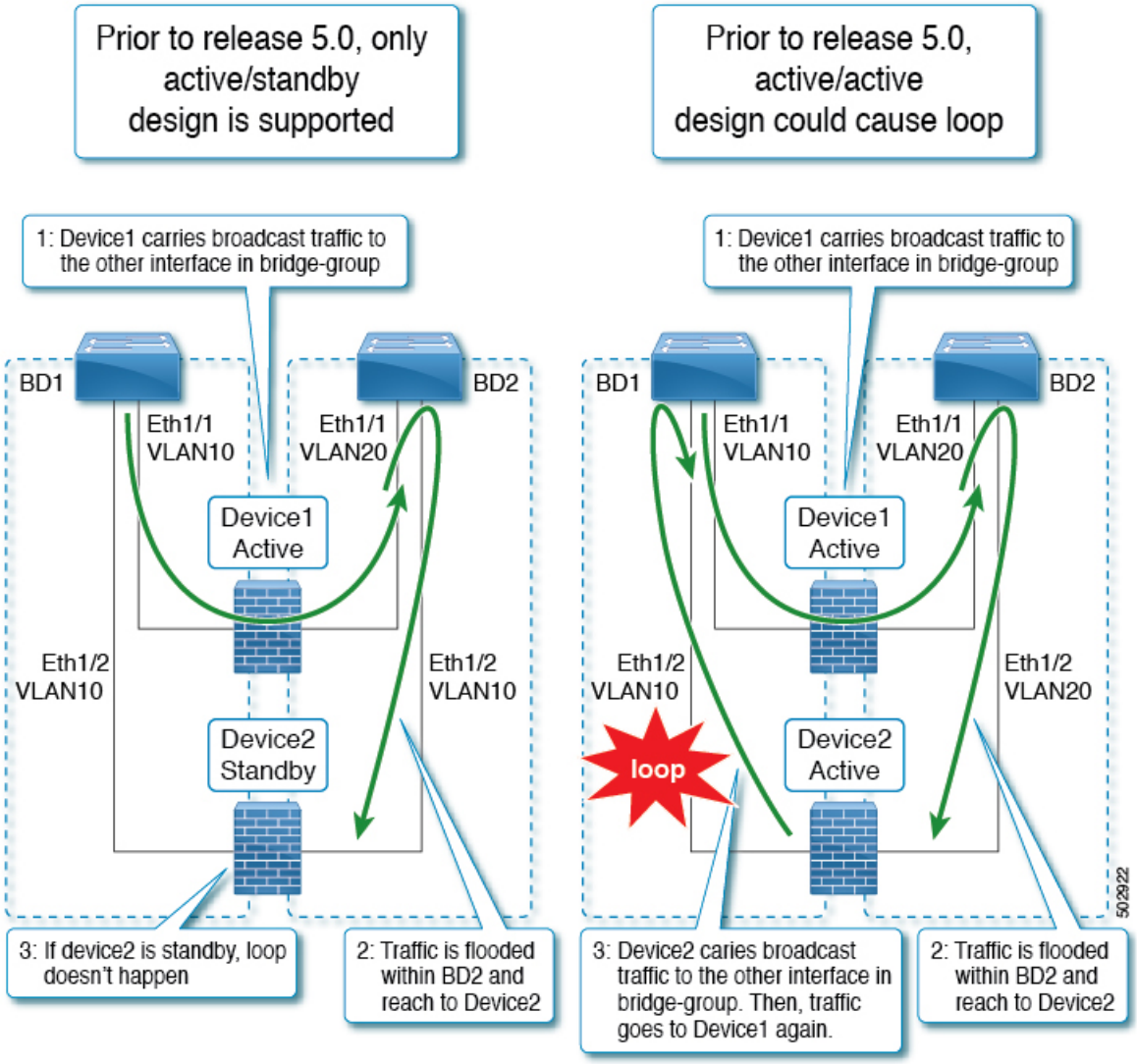
Beginning with Cisco Application Policy Infrastructure Controller (APIC) release 4.1, Layer 1/Layer 2 policy-based redirect (PBR) and active/standby PBR design is supported with tracking.

In the case of Layer 1/Layer 2 PBR, the source and destination MAC addresses of a Layer 2 ping are PBR destination MAC addresses. If the PBR node is up and carries the traffic, Layer 2 ping should successfully return to the Cisco Application Centric Infrastructure (ACI) fabric. Then, the Cisco ACI fabric knows that the PBR destination is available. If there are active and standby high availability Layer 1/Layer 2 service nodes that you want to insert using Layer 1/Layer 2 PBR and there are two PBR destinations where tracking is enabled, only one of the paths that is connected to an active node is up, because a standby device does not forward traffic. As a result, traffic is redirected to the interface that is connected to the active node.

If failover happens and standby takes over the active role, the tracking status changes and traffic gets redirected to the interface that is connected to the new active node.

Prior to Cisco APIC release 5.0(1), as shown in the following illustration, if there are multiple Layer 1/Layer 2 devices in the same service bridge domain pair with an active/standby design, even if traffic is flooded within a bridge domain and the traffic reaches the second Layer 4 to Layer 7 service device, a loop does not happen because this second Layer 4 to Layer 7 service device is in standby mode.

The reason why the active/active design is not supported with releases prior to the Cisco APIC release 5.0(1) is that if there are multiple Layer 1/Layer 2 devices in the same service bridge domain pair, with an active/active design, the second Layer 4 to Layer 7 service device would forward the traffic to the other interface in the other bridge domain and the traffic reaches the first device, thus causing a loop.



Active/Active Layer 1/Layer 2 Symmetric PBR Design Overview

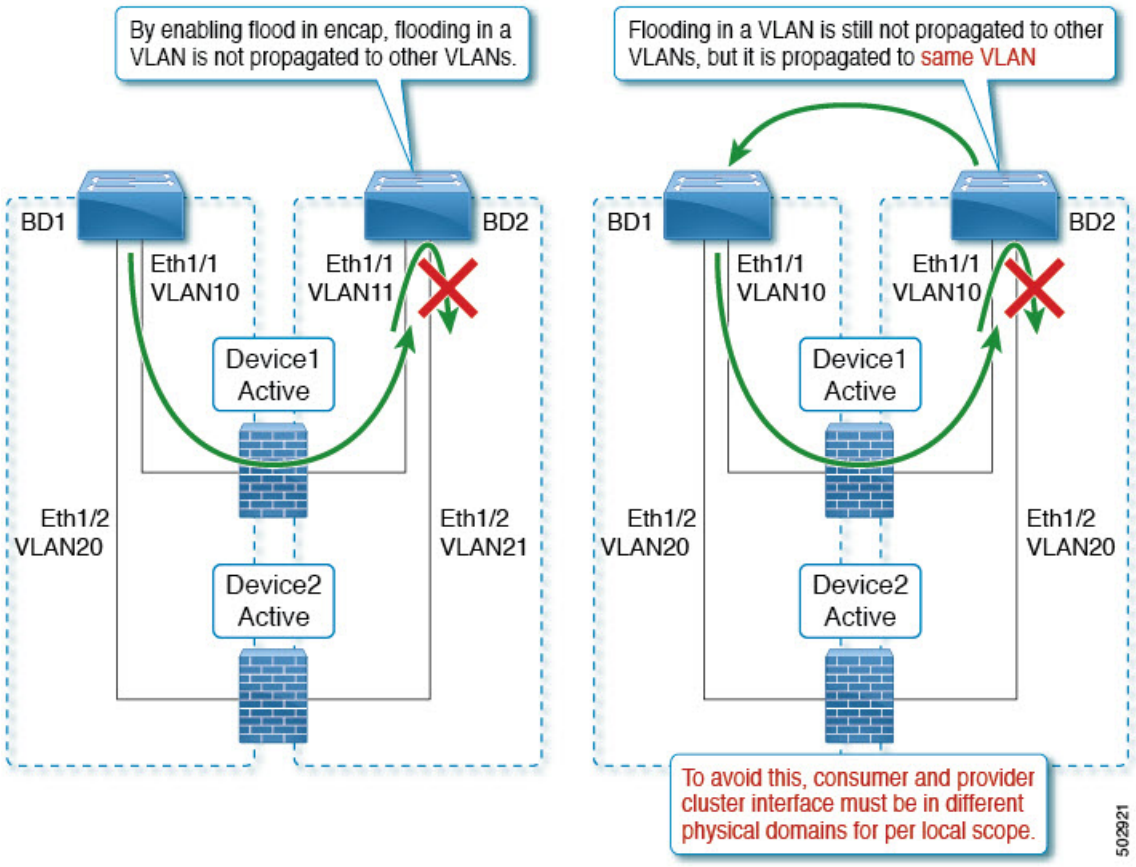
Starting from Cisco APIC release 5.0(1), the Layer 1/ Layer 2 devices in the service chain can operate in active/active symmetric PBR design. Symmetric PBR is used to load balance traffic to individual devices based on hash.

This mode provides high availability and efficient distribution of traffic flows. Symmetric PBR related features such as threshold, down action, and backup PBR policy(N+M high availability) are also supported in APIC Release 5.0(1). For Layer 1 PBR active/active mode, consumer and provider connectors must be in different physical domains.

In order to deploy Layer 1/Layer 2 active/active design, you need to enable the active-active mode in the Layer 4 to Layer 7 Logical Device cluster. You need to provide encapsulation for each concrete device interface in the cluster.

For example: In the same bridge domain pair, by using different VLANs with flood in encap enabled, flooding is propagated within a VLAN and not propagated to the other VLANs. So that you can connect multiple active devices in the same bridge domain.

For Layer 1 active-active mode, the external and internal connectors have the same encap. The use of different VLAN for each active node with flood in encap enabled is not enough to prevent loop. To prevent this issue, both legs of the device should be associated to different physical domain and different VLAN namespace (the actual VLAN range can remain the same). This generates a different `fabEncap` for each leg and prevents a traffic loop.



Configuring a Layer 1/Layer 2 Device Using the GUI

Before you begin

- Create a Layer 1/Layer 2 device using the Cisco APIC GUI and create the concrete device interface.

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

Step 2 In the Navigation pane, choose **Tenant** *tenant_name* > **Services** > **L4-L7**.

Step 3 Right click **Devices** > **Create L4-L7 Devices**

Step 4 In the **Create L4-L7 Devices** dialog box, complete the following fields:

a) In the **Name** field, provide a name for Layer 4 to Layer 7 device cluster.
b) In the **Service Type** choose **Other**.
c) In the **Device Type** choose **Physical**.
d) In the **Physical Domain** choose a physical domain name.
e) In the **Context Aware** choose **Single**.
f) In the **Function Type** choose **L1** or **L2**.
g) Put a check in the box to enable **Active-Active Mode**.

Step 5 Create the concrete device interface: For Layer 1 or Layer 2 **Active-Active Mode**, click **+** on the **Devices** mode in the right work pane. The **Create Concrete Deivce** dialog appears.

a) In the **Name** field, provide a device name.
b) Click **+** to create an encapsulation on the concrete device interface. Enter the name and concrete interface name.

   Since Layer 1/Layer 2 PBR supports two-arm design only, click **+** again to create another concrete interface. Enter the name, interface path, and encapsulation. Click **Update** > **OK**.

   Repeat Step 5a and Step 5b to add more active devices.

c) In the cluster, click **+** to create cluster interface for consumer, choose consumer concrete interfaces. For Layer 1 mode choose a physical domain.

   Click **+** again to create cluster interface for provider, select provider concrete interfaces. For Layer 1 mode, choose another physical domain.

   Note For Layer 1 Active-Active device, create two physical domain mapped to two different VLAN pools, but maintains same VLAN range. For a Layer 2 active-active device, the physical domain is chosen in Step 4e.

Step 6 Click **Finish**.

Configuring Layer 1/ Layer 2 PBR Using the APIC GUI

Before you begin

- Create a L4-L7 device and service graph using the Layer 1/ Layer 2 function type, see configuration steps in *Configuring Policy-Based Redirect Using the GUI*.

Step 1 On the menu bar, choose **Tenants** > **All Tenants**.
Step 2 In the Navigation pane, choose **Tenant** *tenant_name* > **Policies** > **Protocol** > **L4-L7 Policy Based Redirect**.
Step 3 In the Work pane, choose **Action** > **Create L4-L7 Policy Based Redirect**.
Step 4 In the **Create L4-L7 Policy Based Redirect** dialog box, complete the following fields:

a) In the **Name** field, provide a name.
b) In the **Destination Type** field, select **L1** or **L2**.
c) In the **IP SLA Monioring Policy** create the L2 ping monitoring policy.
   - In the **Name** field, provide a name.
   - In the **SLA Type** choose **L2Ping**.

The **SLA Frequency** is optional.

d) In the **L1-L2 Destination** click **+** to add a destination.

Enter the name, redirect health group, and concrete interface. MAC address is optional configuration.

e) Click **OK**.

Note Do not enter an actual interface MAC address. Either leave it blank so the APIC generates MAC automatically or enter a dummy MAC address for external PBR policy MAC A and internal PBR policy MAC B. Remember these MAC addresses are used in firewall configuration.

Step 5 Click **Submit**.

Step 6 In the Navigation pane, choose **Services** > **L4-L7** > **Device Selection Policies** > *Logical Device Context_name*.

Step 7 Expand the Logical Device and apply the Layer 1/ Layer 2 PBR policy in the **L4-L7 Policy-Based Redirect** field for the consumer or provider.

Step 8 Click **Submit**.

Configuring ASA for Layer 1/ Layer 2 PBR Using CLI

Before you begin

- In a general configuration, the service device must be able to forward the Layer 2 ping tracking packet.

  Layer 2 ping, `ethertype 0x0721` is used for tracking. Layer 2 ping is exchanged between leaf nodes, which is going through the service device. Thus, the Layer 1/Layer 2 device needs to permit `ethertype 0x0721`.

- The static MAC configuration is required.
- The following is an example for ASA configuration, where ASA is used as L4-L7 device in Layer 2 mode.

Step 1 The ASA interfaces (service legs) need to be configured in the same bridge-group.

Example:

```
interface GigabitEthernet0/0
nameif externalIf
brdige-group 1

interface GigabitEthernet0/1
nameif internalIf
bridge-group 1
```

Step 2 The ASA learns source MAC address of the layer 2 ping traffic. It's recommended to disable MAC learning to avoid conflicting entries getting created on the ASA as the layer 2 ping traffic uses the same source MAC to track consumer and provider directions.

In the following example, **`externalIf`** is the interface name on ASA, which is used as a consumer connector of the Layer 1 / Layer 2 service node. **`internalIf`** is the interface name on ASA, which is used as a provider connector of

the Layer 1 / Layer 2 service node. Disable MAC learning on **externalIf** and **internalIf**. L2 ping uses the same source MAC when it tries to track both the external and internal legs.

MAC learning is disabled to avoid conflicting entries getting created on the ASA as Layer 2 ping uses the same source MAC to track external and internal.

Example:

```
mac-learn externalIf disable
mac-learn internalIf disable
```

Step 3 Configure ASA rules to permit L2 ping custom ethertype.

Example:

```
access-list Permit-Eth ethertype permit any
access-group Permit-Eth in interface externalIf
access-group Permit-Eth in interface internalIf
```

Step 4 The redirected traffic and Layer2 ping packet use PBR destination MAC, while ASA bridges consumer and provider interfaces. The ASA transparent mode would commonly flood unknown destination MACs, but with L2 PBR, this method cannot be used as PBR destination MACs do not actually exist in the network. Therefore, static MAC entries are recommended to allow Layer 2 ping and PBR traffic to be properly bridged for all cases by the ASA.

Example:

```
mac-address-table static externalIf (MAC B)
mac-address-table static internalIf (MAC A)
```

Note Apart from the configuration of service device such as ASA, if there is an intermediate switch between leaf and the service device, the intermediate switch needs to be able to carry the traffic. You might need static MAC configuration or promiscuous mode configuration on the intermediate switch.

Verifying Layer 1/ Layer 2 PBR Policy On The Leafs Using CLI

The example commands in this procedure are for configuring Layer 1 and Layer 2 Policy-Based Redirect nodes.

Step 1 Check whether PBR group and destination information are configured on the switch:

Example:

```
sdk74-leaf4# show service redir info
GrpID Name                           destination                                   operSt
===== ====                           ===========                                   ======
1     destgrp-1                      dest-[50.50.50.1]-[vxlan-2719744]]            enabled
2     destgrp-2                      dest-[20.20.20.1]-[vxlan-2719744]]            enabled
Name                                 bdVnid          ip              vMac                 vrf
      vrfEncap        operSt
====                                 ======          ==              ====                 ===
      =
dest-[20.20.20.1]-[vxlan-2719744]   vxlan-16514958  20.20.20.1     00:00:14:00:00:01    coke1:cokectx1
  vxlan-2719744   enabled
dest-[50.50.50.1]-[vxlan-2719744]   vxlan-16711542  50.50.50.1     00:00:3C:00:00:01    coke1:cokectx1
  vxlan-2719744   enabled
```

Step 2 Check whether zoning-rule is configured with correct action and group information:

Example:

```
sdk74-leaf4# show zoning-rule | grep redir
4103            49155           49154           18              enabled         2719744
redir(destgrp-2)      fully_qual(6)
4106            49154           49155           17              enabled         2719744
redir(destgrp-1)      fully_qual(6)
```

Step 3 Aclqos subcommand for PBR:

Example:

```
module-1# show system internal aclqos services redir ?
  <CR>
  dest   Dest related info
  group  Group related info

module-1# show system internal aclqos services redir group 1

 Flag Legend :
 0x1: In SDK
 0x10: In local DB
 0x20: Delete pending
 0x40: Dummy adj

******* Service key redir-group(1) *******
 Service flags: 0x11
 Num of reference: 0x1
 Num of path: 1
 path 0 key: redir-dest-ipv4(vrf vnid vxlan-2719744 prefix-50.50.50.1)

module-1# show system internal aclqos services redir dest 2719744 50.50.50.1
 Flag Legend :
 0x1: In SDK
 0x10: In local DB
 0x20: Delete pending
 0x40: Dummy adj
******* Service key redir-dest-ipv4(vrf vnid vxlan-2719744 prefix-50.50.50.1) *******
 Service flags: 0x10
 Num of reference: 0x1
 Num of path: 1
 Ifindx: 0x18010007
 Bd_vnid: 16711542
 Vmac: 00:00:3c:00:00:01
```

Step 4 Zoning-rule command:

Example:

```
module-1# show system internal aclqos zoning-rules 4106
 ASIC type is Sug
==============================================================
Rule ID: 4106 Scope 3 Src EPG: 49154 Dst EPG: 49155 Filter 17
  Redir group: 1

  Curr TCAM resource:
  ============================
    unit_id: 0
    === Region priority: 1539 (rule prio: 6 entry: 3)===
        sw_index = 44 | hw_index = 44
    === SDK Info ===
        Result/Stats Idx: 81876
        30
```

```
        Tcam Total Entries: 1
        HW Stats: 0
```

Configuring Layer 1/ Layer 2 PBR Using the REST API

Layer 1/ Layer 2 Policy-Based Redirect configuration:

Example:

```
<polUni>
    <fvTenant name="coke" >

        <!—If L1/L2 device in active-active mode -- >
        <vnsLDevVip name="N1"  activeActive="yes" funcType="L1" managed="no">
        </vnsLDevVip>
        <!—If L1/L2 device in active-standby mode -- >
        <vnsLDevVip name="N1"  activeActive="no" funcType="L1" managed="no">
        </vnsLDevVip>

      <vnsAbsGraph descr="" dn="uni/tn-coke/AbsGraph-WebGraph" name="WebGraph" ownerKey="" ownerTag=""
 uiTemplateType="UNSPECIFIED">

        <!—For L2 device -- >
       <vnsAbsNode descr="" funcTemplateType="OTHER" funcType="L2" isCopy="no" managed="no" name="N1"
 ownerKey="" ownerTag="" routingMode="Redirect" sequenceNumber="0" shareEncap="no">
    </vnsAbsNode>

       <!—For L1 device -- >
     <vnsAbsNode descr="" funcTemplateType="OTHER" funcType="L1" isCopy="no" managed="no" name="N1"
 ownerKey="" ownerTag="" routingMode="Redirect" sequenceNumber="0" shareEncap="no">
    </vnsAbsNode>

    </vnsAbsGraph>

    <fvIPSLAMonitoringPol name="Pol2" slaType="l2ping"/>
<vnsSvcCont>
<vnsRedirectHealthGroup name="2" />
             <vnsSvcRedirectPol name="N1Ext"  destType="L2">
                   <vnsRsIPSLAMonitoringPol tDn="uni/tn-coke/ipslaMonitoringPol-Pol2"/>
     <vnsL1L2RedirectDest destName="1">
                <vnsRsL1L2RedirectHealthGroup tDn="uni/tn-coke/svcCont/redirectHealthGroup-2"/>
            <vnsRsToCIf tDn="uni/tn-coke/lDevVip-N1/cDev-ASA1/cIf-[Gig0/0]"/>
  </vnsL1L2RedirectDest>
             </vnsSvcRedirectPol>

             <vnsSvcRedirectPol name="N1Int" destType="L2">
                   <vnsRsIPSLAMonitoringPol tDn="uni/tn-coke/ipslaMonitoringPol-Pol2"/>
      <vnsL1L2RedirectDest destName="2">
                <vnsRsL1L2RedirectHealthGroup tDn="uni/tn-coke/svcCont/redirectHealthGroup-2"/>
           <vnsRsToCIf tDn="uni/tn-coke/lDevVip-N1/cDev-ASA1/cIf-[Gig0/1]"/>
     </vnsL1L2RedirectDest>
          </vnsSvcRedirectPol>
        </vnsSvcCont>
</fvTenant>
</polUni>
```

Policy-Based Redirect and Tracking Service Nodes

Beginning with the Cisco Application Policy Infrastructure Controller (APIC) 2.2(3) and 3.1(1) releases (but, excluding the 3.0 releases), the policy-based redirect feature (PBR) supports the ability to track service nodes. Tracking enables you to prevent redirection of traffic to a service node that is down. If a service node (PBR destination) is down, the PBR hashing can begin selecting an available PBR destination in a policy. This feature requires Cisco Nexus 9300-EX, -FX, or later platform leaf switches.

Service nodes can support dual IP address stacking. Therefore, this feature has the capability to track both IPv4 and IPv6 addresses at the same time. When both IPv4 and IPv6 addresses are "up," the PBR destination is marked as "up."

Switches internally use the Cisco IP SLA monitoring feature to support PBR tracking. The tracking feature marks a redirect destination node as "down" if the service node is not reachable. The tracking feature marks a redirect destination as node "up" if the service node resumes connectivity. When a service node is marked as "down," it will not be used to send or hash the traffic. Instead, the traffic will be sent or hashed to a different service node in the cluster of redirection destination nodes.

To avoid black holing of the traffic in one direction, you can associate a service node's ingress and egress redirect destination nodes with a redirection health policy. Doing so ensures that if either an ingress or egress redirection destination node is down, the other redirection destination node will also be marked as "down." Hence, both ingress and egress traffic gets hashed to a different service node in the cluster of the redirect destination nodes.

You can use the following protocols for tracking:

- ICMP (for Layer 3 PBR)
- TCP (for Layer 3 PBR)
- L2ping (for Layer 1/2 PBR)

Policy-Based Redirect and Tracking Service Nodes with a Health Group

Policy-based redirect (PBR) service node tracking enables you to prevent the redirection of traffic to a failed PBR node. If the consumer or the provider connector of the PBR node is down, the traffic that went through the failed node may get black holed. To prevent the traffic from being black holed, Cisco Application Centric Infrastructure (ACI) avoids the use of the PBR node for traffic in both directions. Some Layer 4 to Layer services devices can bring down an interface if another interface is down, which you can use to prevent traffic from being black holed. If the PBR node does not have this capability, you should use the health group feature to disable PBR for the node if either the consumer or provider connector is down.

Each PBR destination IP and MAC address can be in a health group. For example, assume that you have two PBR node destinations. One has 172.16.1.1 as the consumer connector and 172.16.2.1 as the provider connector, and these are in Health-group1. The other has 172.16.1.2 as the consumer connector and 172.16.2.2 as the provider connector, and these are in Health-group2. If either of the PBR destinations in the same health group is down, that node will not be used for PBR.

Policy-Based Redirect and Threshold Settings for Tracking Service Nodes

The following threshold settings are available when configuring a policy-based redirect (PBR) policy for tracking service nodes:

- Threshold enabled or disabled: When the threshold is enabled, you an specify the minimum and maximum threshold percentages. Threshold enabled is required when you want to disable the redirect destination group completely and prevent any redirection. When there is no redirection, the traffic is directly sent between the consumer and the provider.
- Minimum threshold: The minimum threshold percentage specified. If the traffic goes below the minimum percentage, the packet is permitted instead of being redirected. The default value is 0.
- Maximum threshold: The maximum threshold percentage specified. Once the minimum threshold is reached, to get back to operational state, the maximum percentage must first be reached. The default value is 0.

Let us assume as an example that there are three redirect destinations in a policy. The minimum threshold is specified at 70% and the maximum threshold is specified at 80%. If one of the three redirect destination policies goes down, the percentage of availability goes down by one of three (or 33%), which is less than the minimum threshold. As a result, the minimum threshold percentage of the redirect destination group is brought down and traffic begins to get permitted instead of being redirected. Continuing with the same example, if the maximum threshold is 80%, to bring the redirect policy destination group back to the operational state, a percentage greater than the maximum threshold percentage must be reached.

Guidelines and Limitations for Policy-Based Redirect With Tracking Service Nodes

Follow these guidelines and limitations when using policy-based redirect (PBR) tracking with service nodes:

- Destination groups that share destinations must have same health group and IP SLA monitoring policies configured.
- Beginning in release 4.0(1), remote leaf switch configurations support PBR tracking, but only if system-level global GIPo is enabled. See *Configuring Global GIPo for Remote Leaf Using the GUI*.
- Beginning in release 4.0(1), remote leaf switch configurations support PBR resilient hashing.
- A Cisco ACI Multi-Pod fabric setup is supported.
- A Cisco ACI Multi-Site setup supported, but the PBR destinations cannot be in a different site.
- An L3Out is supported for the consumer and provider EPGs.
- PBR supports up to 100 trackable IP addresses in leaf switches and 1500 trackable IP addresses in the Cisco Application Centric Infrastructure (ACI) fabric.
- For the maximum number of service graph instances in the Cisco ACI fabric, see the Verified Scalability Guide for Cisco APIC for your specific release.
- For the maximum number of service graph instances per device, see the Verified Scalability Guide for Cisco APIC for your specific release.
- You can configure up to 40 service nodes per PBR policy.
- You can configure up to 5 service nodes per service chain.
- Shared services are supported with PBR tracking.
- The following threshold down actions are supported:

- bypass action
- deny action
- permit action

- If multiple PBR policies have the same PBR destination IP address in the same VRF instance, the policies must use the same IP SLA policy and health group for the PBR destination.
- When using HTTP URI tracking, the following guidelines and limitations apply:
  - Tracking supports both IPv4 and IPv6.
  - Tracking supports only HTTP, not HTTPS.
  - Tracking supports only HTTP version 1.0 and 1.1.
  - The destination port must be 80.
  - You must configure the URI (not the URL). Domain resolution is not supported.
  - The URI must not be empty and must start with "/".
  - Tracking supports 100 probes per leaf switch and 1500 per fabric. The values are the total of ICMP, L2ping, TCP, and HTTP probes.
  - The minimum frequency should be 5 seconds. In standalone NX-OS, the minimum frequency is 60 seconds.
- The Cisco Application Policy Infrastructure Controller (APIC) generates a fault when the same destination is tracked with two different tracking protocols. The fault is similar to the following example:

```
Fault delegate: PBR service source on nodeid 106 fabric hostname apic1-leaf6 is in
failed state. reason multiple tracking types configured.
```

Configuring PBR and Tracking Service Nodes Using the GUI

Step 1 On the menu bar, click **Tenant** > ***tenant_name***. In the navigation pane, click **Policies** > **Protocol** > **L4-L7 Policy Based Redirect** .

Step 2 Right-click **L4 –L7 Policy Based Redirect**, and click **Create L4–L7 Policy Based Redirect**.

Step 3 In the **Create L4–L7 Policy Based Redirect** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the policy-based redirect (PBR) policy.

b) In the dialog box, choose the appropriate settings to configure the hashing algorithm, IP SLA monitoring policy, and other required values.

Note Destination groups that share destinations must have same IP SLA monitoring policy configured.

c) In the threshold setting fields, specify the settings as appropriate and if desired.

d) For **L3 Destinations**, click **+** to display **Create Destination of Redirected Traffic**.

e) In the **Create Destination of Redirected Traffic** dialog box, enter the appropriate values.

The **IP** and **Additional IPv4/IPv6** fields are provided where you can specify IPv4 or IPv6 addresses.

Note The **Additional IPv4/IPv6** field is not mandatory. Use the field if the Layer 4 to Layer 7 services device has multiple IP addresses and you want Cisco Application Centric Infrastructure (ACI) to verify both of them.

If both the **IP** and **Additional IPv4/IPv6** parameters are configured, both must be up in order to mark the PBR destination as "UP".

f) In the **Redirect Health Group** field, associate an existing health group or create a new health group, as appropriate. Click **OK**.

Note Destination groups that share destinations must have same health group configured.

g) In the **Create L4–L7 Policy Based Redirect** dialog box, click **Submit**.

The Layer 4 to Layer 7 PBR and tracking of service nodes is configured after binding the redirect health group policy to the PBR policy and the settings to track the redirect destination group are enabled.

Configuring a Redirect Health Group Using the GUI

Step 1 On the menu bar, click **Tenant** > **tenant_name**. In the navigation pane, click **Policies** > **Protocol** > **L4-L7 Redirect Health Groups**.

Step 2 Right-click **L4 –L7 Redirect Health Groups**, and choose **Create L4–L7 Redirect Health Group**.

Step 3 In the **Create L4–L7 Redirect Health Group** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the Redirect Health Group policy.

b) In the **Description** field, enter additional information if appropriate, and click **Submit**.

The Layer 4 to Layer 7 services redirect health policy is configured.

Configuring Global GIPo for Remote Leaf Using the GUI

Performing this task allows PBR tracking to function in remote leaf configurations.

Note This configuration must be performed for PBR tracking to function on a remote leaf. Without this configuration, PBR tracking will not work on the remote leaf, even when the main data center is reachable.

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

Step 2 In the **System Settings** navigation pane, click **System Global GIPo**.

Step 3 In the **System Global GIPo Policy** work pane, click **Enabled**.

Step 4 In the **Policy Usage Warning** dialog, review the nodes and policies that may be using the GIPo policy and, if appropriate, click **Submit Changes**.

Configuring PBR to Support Tracking Service Nodes Using the REST API

Configure PBR to support tracking service nodes.

Example:

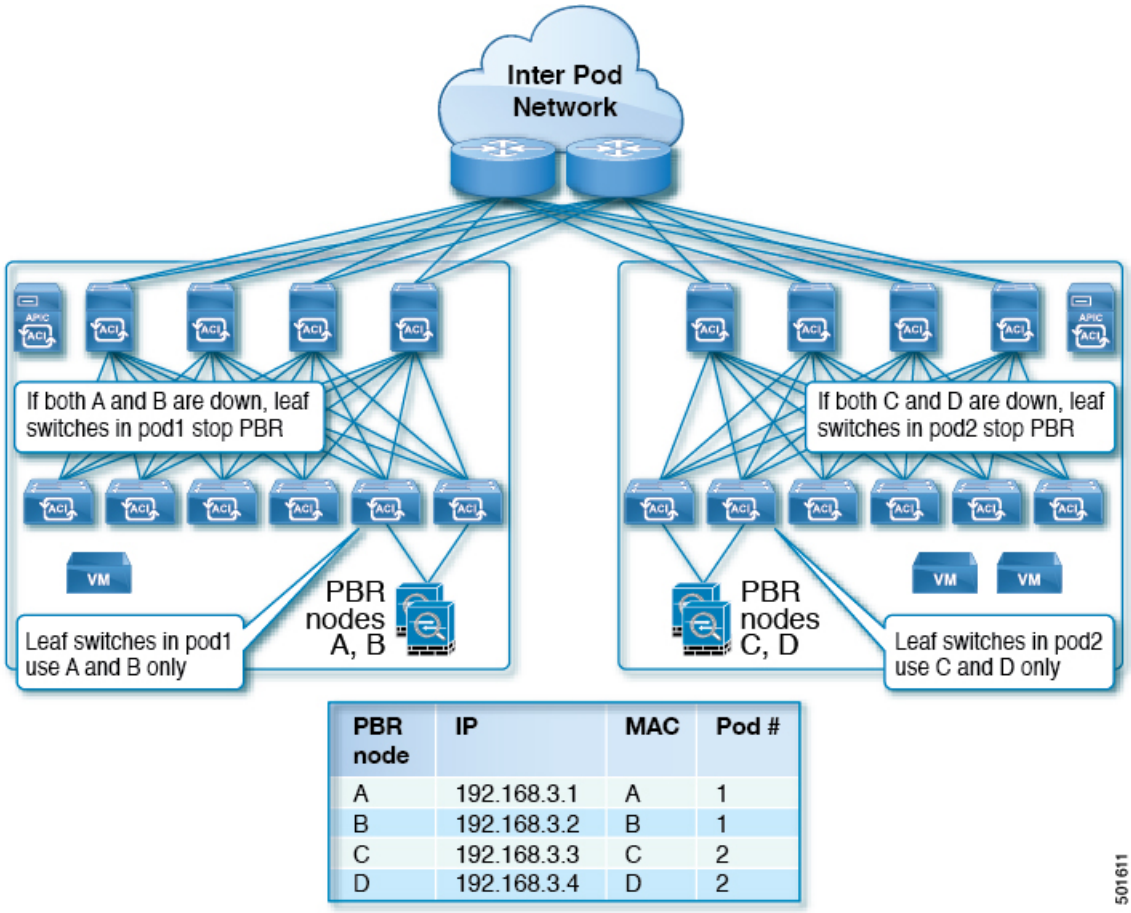
```
<polUni>
    <fvTenant name="t1" >
    <fvIPSLAMonitoringPol name="tcp_Freq60_Pol1" slaType="tcp" slaFrequency="60" slaPort="2222" />
    <vnsSvcCont>
        <vnsRedirectHealthGroup name="fwService1"/>
          <vnsSvcRedirectPol name="fwExt" hashingAlgorithm="sip" thresholdEnable="yes"
            minThresholdPercent="20" maxThresholdPercent="80">
            <vnsRedirectDest ip="40.40.40.100" mac="00:00:00:00:00:01">
               <vnsRsRedirectHealthGroup tDn="uni/tn-t1/svcCont/redirectHealthGroup-fwService1"/>
            </vnsRedirectDest>
            <vnsRsIPSLAMonitoringPol tDn="uni/tn-t1/ipslaMonitoringPol-tcp_Freq60_Pol1"/>
          </vnsSvcRedirectPol>
         <vnsSvcRedirectPol name="fwInt" hashingAlgorithm="sip" thresholdEnable="yes"
           minThresholdPercent="20" maxThresholdPercent="80">
             <vnsRedirectDest ip="30.30.30.100" mac="00:00:00:00:00:02">
                <vnsRsRedirectHealthGroup tDn="uni/tn-t1/svcCont/redirectHealthGroup-fwService1"/>
             </vnsRedirectDest>
             <vnsRsIPSLAMonitoringPol tDn="uni/tn-t1/ipslaMonitoringPol-tcp_Freq60_Pol1"/>
          </vnsSvcRedirectPol>
        </vnsSvcCont>
    </fvTenant>
</polUni>
```

About Location-Aware Policy Based Redirect

Location-Aware Policy Based Redirect (PBR) is now supported. This feature is useful in a multipod configuration scenario. Now there is pod-awareness support, and you can specify the preferred local PBR node. When you enable location-aware redirection, and Pod IDs are specified, all the redirect destinations in the Layer 4-Layer 7 PBR policy will have pod awareness. The redirect destination is programmed only in the leaf switches located in a specific pod.

The following image displays an example with two pods. PBR nodes A and B are in Pod 1 and PBR nodes C and D are in Pod 2. When you enable the location-aware PBR configuration, the leaf switches in Pod 1 prefer to use PBR nodes A and B, and the leaf switches in Pod 2 use PBR nodes in C and D. If PBR nodes A and B in Pod 1 are down, then the leaf switches in Pod 1 will start to use PBR nodes C and D. Similarly, if PBR nodes C and D in Pod 2 are down, the leaf switches in Pod 2 will start to use PBR nodes A and B.

Figure 18: An Example of Location Aware PBR Configuration with Two Pods



PBR node	IP	MAC	Pod #
A	192.168.3.1	A	1
B	192.168.3.2	B	1
C	192.168.3.3	C	2
D	192.168.3.4	D	2

Guidelines for Location-Aware PBR

Follow these guidelines when using location-aware PBR:

- The Cisco Nexus 9300 (except Cisco Nexus 9300–EX and 9300–FX) platform switches do not support the location-aware PBR feature.
- Use location-aware PBR for north-south firewall integration with GOLF host advertisement.

  Use location-aware PBR for a contract that is enforced on the same leaf nodes for incoming and returning traffic, such as an intra-VRF contract for external-EPG-to-EPG and an inter-VRF contract for EPG-to-EPG traffic. Otherwise, there can be a loss of traffic symmetry.
- If multiple PBR policies have the same PBR destination IP address in the same VRF, then all of the policies must either have Pod ID aware redirection enabled or Pod ID aware redirection disabled. The same (VRF, IP address) pair cannot be used in Pod ID aware redirection enabled and Pod ID aware redirection disabled policies at the same time. For example, the following configuration is not supported:
  - PBR-policy1 has PBR destination 192.168.1.1 in VRF A, Pod ID aware redirection enabled, and 192.168.1.1 is set to POD 1.
  - PBR-policy2 has PBR destination 192.168.1.1 in VRF A and Pod ID aware redirection disabled.

Configuring Location-Aware PBR Using the GUI

You must program two items for this feature to be enabled. Enable pod ID aware redirection and associate the Pod IDs with the preferred PBR nodes to program redirect destinations in the leaf switches located in the specific pods.

Step 1 On the menu bar, click **Tenant** > ***tenant_name***. In the **Navigation** pane, click **Policies** > **Protocol** > **L4-L7 Policy Based Redirect** .

Step 2 Right-click **L4 –L7 Policy Based Redirect**, and click **Create L4–L7 Policy Based Redirect**.

Step 3 In the **Create L4–L7 Policy Based Redirect** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the PBR policy.
b) In the **Enable Pod ID Aware Redirection** check the check box.
c) In the dialog box, choose the appropriate settings to configure the hashing algorithm, IP SLA Monitoring Policy, and other required values.
d) In the threshold setting fields, specify the settings as appropriate and if desired.
e) Expand **Destinations** to display **Create Destination of Redirected Traffic**.
f) In the **Create Destination of Redirected Traffic** dialog box, enter the appropriate details including the **IP** address and the **MAC address** fields.

   The fields for IP address and Second IP address are provided where you can specify an IPv4 address and an IPv6 address.

g) In the **Pod ID** field, enter the pod identification value.
h) In the **Redirect Health Group** field, associate an existing health group or create a new health group, as appropriate. Click **OK**.

   Create additional destinations of redirected traffic with different Pod IDs as required.

i) In the **Create L4–L7 Policy Based Redirect** dialog box, click **Submit**.

The L4-L7 location-aware PBR is configured.

Configuring Location-Aware PBR Using the REST API

You must configure two items to enable location-aware PBR and to program redirect destinations in the leaf switches located in the specific pods. The attributes that are configured to enable location-aware PBR in the following example are: programLocalPodOnly and podId.

Configure location-aware PBR.

Example:

```
<polUni>
    <fvTenant name="coke" >
    <fvIPSLAMonitoringPol name="icmp_Freq60_Pol1" slaType="icmp" slaFrequency="60"/>
    <vnsSvcCont>
        <vnsRedirectHealthGroup name="fwService1"/>
          <vnsSvcRedirectPol name="fwExt" hashingAlgorithm="sip" thresholdEnable="yes"
minThresholdPercent="20" maxThresholdPercent="80" programLocalPodOnly="yes">
            <vnsRedirectDest ip="40.40.40.100" mac="00:00:00:00:00:01" podId="2">
```

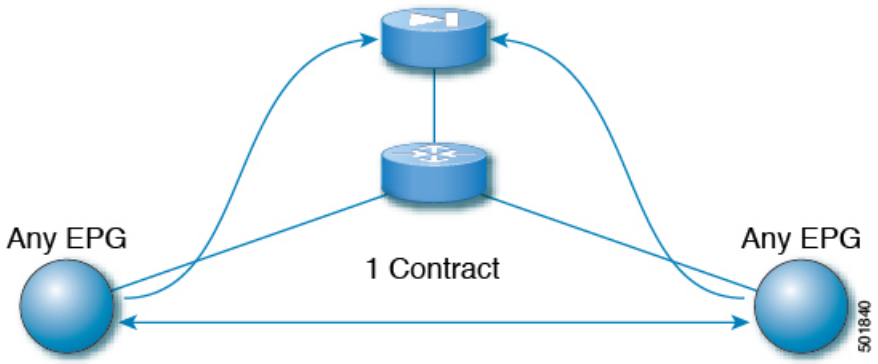
```
                <vnsRsRedirectHealthGroup tDn="uni/tn-coke/svcCont/redirectHealthGroup-fwService1"/>
              </vnsRedirectDest>
              <vnsRsIPSLAMonitoringPol tDn="uni/tn-coke/ipslaMonitoringPol-icmp_Freq60_Pol1"/>
            </vnsSvcRedirectPol>
            <vnsSvcRedirectPol name="fwInt" hashingAlgorithm="dip" thresholdEnable="yes"
minThresholdPercent="20" maxThresholdPercent="80">
              <vnsRedirectDest ip="30.30.30.100" mac="00:00:00:00:00:02">
                <vnsRsRedirectHealthGroup tDn="uni/tn-coke/svcCont/redirectHealthGroup-fwService1"/>

              </vnsRedirectDest>
              <vnsRsIPSLAMonitoringPol tDn="uni/tn-coke/ipslaMonitoringPol-icmp_Freq60_Pol1"/>
            </vnsSvcRedirectPol>
          </vnsSvcCont>
      </fvTenant>
</polUni>
```

Policy-Based Redirect and Service Graphs to Redirect All EPG-to-EPG Traffic Within the Same VRF Instance

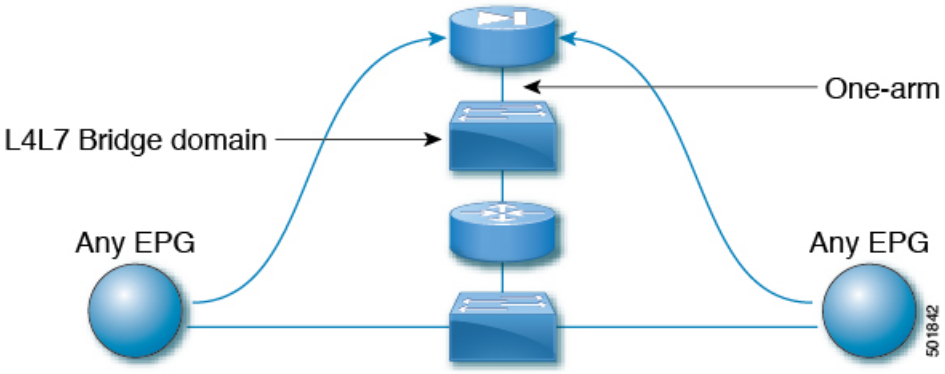
You can configure Cisco Application Centric Infrastructure (Cisco ACI) to forward all traffic from any endpoint group to any other endpoint group in the same VRF instance through a Layer 4 to Layer 7 device by configuring vzAny with service graph redirect. vzAny is a construct that represents all the endpoint groups under the same VRF instance. vzAny is sometimes referred to as "any EPG."

Figure 19: vzAny topology



Traffic between any endpoint group pair that is under the same VRF instance can be redirected to a Layer 4 to Layer 7 device, such as a firewall. You can also redirect traffic within the same bridge domain to a firewall. The firewall can filter traffic between any pair of endpoint groups, as illustrated in the following figure:

Figure 20: A firewall filtering traffic between any pair of EPGs

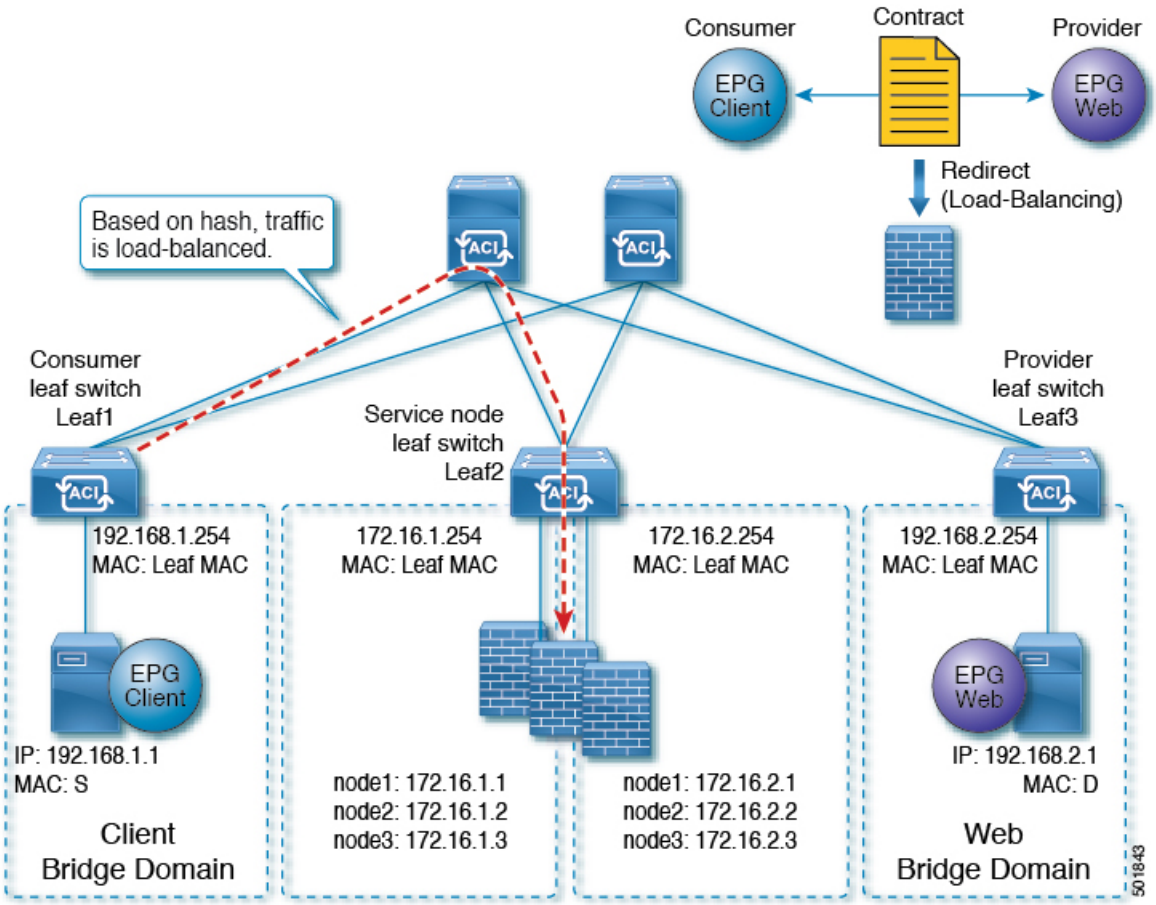


One use case of this functionality is to use Cisco ACI as a default gateway, but filter traffic through a firewall. With vzAny and a policy-based redirect policy, the security administrator manages the ACL rules and the network administrator manages routing and switching. Some of the benefits of this configuration include being able to use the Cisco Application Policy Infrastructure Controller's (Cisco APIC's) tools, such as endpoint tracking, first hop security with ARP inspection, or IP address source guard.

Applying a service graph with a policy-based redirect policy also enables the following functionality:

- Firewall clustering
- Firewall health tracking
- Location-aware redirection

Figure 21: Firewall clustering



Prior to the Cisco APIC 3.2 release, you could use vzAny as the consumer of a contract. Starting in the Cisco APIC 3.2 release, you can also use vzAny as the provider of a contract. This enhancement enables the following configurations:

- vzAny as the provider and vzAny as the consumer (policy-based redirect with one-arm only)
- vzAny as the provider and a regular endpoint group as the consumer (policy-based redirect and non-policy-based redirect case)

After you have applied a service graph with a policy-based redirect policy that redirects traffic using vzAny, if you want some traffic to bypass the firewall, such as for data backup traffic between two servers, you can create a more specific contract between the endpoint groups. For example, two endpoint groups can transmit traffic to one another directly over a given port. More specific rules win over the "any EPG to any EPG" redirect rule.

Guidelines and Limitations for Configuring a Policy-Based Redirect Policy with a Service Graph to Redirect All EPG-to-EPG Traffic Within the Same VRF Instance

The following guidelines and limitations apply when configuring a policy-based redirect policy with a service graph to redirect all EPG-to-EPG traffic within the same VRF instance:

- The Layer 4 to Layer 7 services device and vzAny must belong to the same VRF instance.
- You must deploy the Layer 4 to Layer 7 services device in one-arm mode.
- We generally recommend that you use a vzAny contract to enable PBR for many EPGs to many EPGs traffic instead of many EPGs consuming and providing the same contract. However, do not have a contract that has service graph attached as both the consumer and provider contract on the same EPG.

  This recommendation is due to a possible impact on changing a configuration on a contract that has many provider and consumer EPGs. If one configuration change on the Cisco Application Policy Infrastructure Controller (APIC) is related to multiple zoning-rule changes at the same time, the Cisco APIC needs time to finish programming the hardware of a given leaf node.
- vzAny configured with a multinode service graph might work, but this configuration has not been tested and is unsupported; use at your own risk.
- The use in conjunction with VRF leaking is not implemented. You cannot have vzAny of a VRF instance providing or consuming a vzAny contract of another VRF instance.
- You can have a contract between endpoint groups and vzAny in different tenants as long as they belong to the same VRF instance, such as if the VRF instance is in tenant **Common**.
- In a multipod environment, you can use vzAny as a provider and consumer.
- In a Cisco ACI Multi-Site environment, you cannot use vzAny as a provider and consumer across sites.

Configuring a Policy-Based Redirect Policy with a Service Graph to Redirect All EPG-to-EPG Traffic Within the Same VRF Instance

The following procedure configures a policy-based redirect policy with service graphs to redirect all EPG-to-EPG traffic within the same VRF instance:

Step 1 Create the service bridge domain that you will dedicate to the connectivity of the Layer 4 to Layer 7 device.

For information about creating a bridge domain, see the *Cisco APIC Basic Configuration Guide*.

On the **STEP 1 > Main** screen:

a) In the **VRF** drop-down list, choose the VRF instance that contains the endpoint groups.
b) In the **Forwarding** drop-down list, if you choose **Custom**, then in the **L2 Unknown Unicast** drop-down list, you can choose **Flood** if desired.

On the **STEP 2 > L3 Configurations** screen:

a) Ensure that there is a check in the **Unicast Routing** check box.
b) In the **Subnets** table, create a subnet.

The **Gateway IP** address must be in the same subnet as the IP address that you will give to the Layer 4 to Layer 7 device interface.

c) Remove the check from the **Endpoint Dataplane Learning** check box.

Step 2 Create the redirect policy.

a) In the **Navigation** pane, choose **Tenant tenant_name** > **Policies** > **Protocol** > **L4-L7 Policy Based Redirect**.
b) Right-click **L4-L7 Policy Based Redirect** and choose **Create L4-L7 Policy Based Redirect**.
c) In the **Name** field, enter a name for the policy.
d) In the **L3 Destinations** table, click **+**.
e) In the **Create Destination of Redirected Traffic** dialog, enter the following information:

- **IP**—Enter the IP address that you will assign to the Layer 4 to Layer 7 device. The IP address must be in the same subnet as the IP address that you have given to the bridge domain.
- **MAC**—Optional. Enter the MAC address that you will assign to the Layer 4 to Layer 7 device. You should use a MAC address that is valid also upon failover of the Layer 4 to Layer 7 device. For example, in the case of a ASA firewall, this is called a *virtual MAC*. If you do not specify the MAC address, the address will be dynamically detected.

f) Enter any other desired values, then click **OK**.
g) In the **Create L4-L7 Policy Based Redirect** dialog, enter any other desired values, then click **Submit**.

Step 3 Create the Layer 4 to Layer 7 device with one concrete interface and one logical interface.

For information about creating a Layer 4 to Layer 7 device, see Configuring a Layer 4 to Layer 7 Services Device Using the GUI, on page 9.

Step 4 Create the service graph template with route redirect enabled.

a) In the **Navigation** pane, choose **Tenant tenant_name** > **Services** > **L4-L7** > **Service Graph Template**.
b) Right-click **Service Graph Template** and choose **Create Service Graph Template**.
c) In the **Name** field, enter a name for the service graph.
d) If you did not previously create the Layer 4 to Layer 7 device, in the **Device Clusters** pane, create the device.
e) Drag and drop the Layer 4 to Layer 7 device from the **Device Clusters** pane to in-between the consumer EPG and provider EPG.
f) For the **L4L7** radio buttons, click **Routed**.
g) Put a check in the **Routed Redirect** check box.
h) Click **Submit**.

Step 5 Apply the service graph to the vzAny (AnyEPG) endpoint group.

On the **STEP 1 > Contract** screen:

a) In the **Navigation** pane, choose **Tenant tenant_name** > **Services** > **L4-L7** > **Service Graph Template** > ***service_graph_name***.

service_graph_name is the service graph template that you just created.

b) Right-click the service graph template and choose **Apply L4-L7 Service Graph Template**.
c) In the **Consumer EPG / External Network** drop-down list, choose the **AnyEPG** list item that corresponds to the tenant and VRF instance that you want to use for this use case.

For example, if the tenant is "tenant1" and the VRF instance is "vrf1," choose **tenant1/vrf1/AnyEPG**.

d) In the **Provider EPG / Internal Network** drop-down list, choose the same **AnyEPG** list item that you chose for the consumer EPG.
e) In the **Contract Name** field, enter a name for the contract.
f) Click **Next**.

On the **STEP 2 > Graph** screen:

a) For both **BD** drop-down lists, choose the Layer 4 to Layer 7 service bridge domain that you created in step 1.
b) For both **Redirect Policy** drop-down lists, choose the redirect policy that you created for this use case.
c) For the Consumer Connector **Cluster Interface** drop-down list, choose the cluster interface (logical interface) that you created in step 3.
d) For the Provider Connector **Cluster Interface** drop-down list, choose the same cluster interface (logical interface) that you created in step 3.
e) Click **Finish**.

Dynamic MAC Address Detection for a Layer 3 Policy-Based Redirect Destination

Beginning in the Cisco Application Policy Infrastructure Controller (APIC) 5.2(1) release, you can configure any of the Layer 3 policy-based redirect (PBR) destinations without specifying a MAC address. An example of a PBR destination is a Layer 4 to Layer 7 device that is part of a service graph. By configuring this feature, the leaf switches use the Address Resolution Protocol (ARP) to determine the MAC address of the PBR next-hop. The benefit is that you do not need to check the MAC address of each PBR destination and an active-standby HA pair does not need to use a floating MAC address.

Guidelines and Limitations for Dynamic MAC Address Detection for a Layer 3 Policy-Based Redirect Destination

The following are guidelines and limitations for configuring dynamic MAC address detection for a Layer 3 policy-based redirect (PBR) destination:

- You must enable tracking on the destinations for which you did not specify a MAC address.
- You can use all Layer 3 PBR equal cost multipath (ECMP) features and IPv4 and IPv6 destinations.
- In the same PBR policy, you can have destinations for which you did not configure the MAC address together with destinations for which you configured the MAC address.
- When a MAC address is changed, detecting the new MAC address and updating the PBR destination MAC address on the consumer and provider leaf switches takes time, depending on the tracking interval.
- You can have 100 destinations per leaf switch and 1500 destinations per fabric.

Configuring Dynamic MAC Address Detection for a Layer 3 Policy-Based Redirect Destination Using the GUI

The following procedure configures dynamic MAC address detection for a Layer 3 policy-based redirect (PBR) destination.

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

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

Step 3 In the Navigation pane, choose **Tenant** *tenant_name* > **Policies** > **Protocol** > **L4-L7 Policy-Based Redirect**.

Step 4 Right-click **L4-L7 Policy-Based Redirect** and choose **Create L4-L7 Policy-Based Redirect**.

Step 5 In the **Create L4-L7 Policy-Based Redirect** dialog box, complete the fields as required, except as specified below:

a) For **Destination Type**, choose **L3** if it is not already chosen.

b) In the **IP SLA Monitoring Policy** drop-down list, choose an existing IP SLA monitoring policy or create a new policy.

c) In the **L3 Destinations** section, click **+**.

d) In the **Create Destination of redirected traffic** dialog box, for the **MAC** field, enter `00:00:00:00:00:00` or leave the value empty.

Either way enables dynamic MAC address detection. If you leave the value empty, the value becomes `00:00:00:00:00:00` when you finish creating the policy.

e) For **Redirect Health Group**, choose an existing health group or create a new health group, as appropriate.

f) Complete the remaining fields as required.

g) Click **OK**.

h) Click **Submit**.

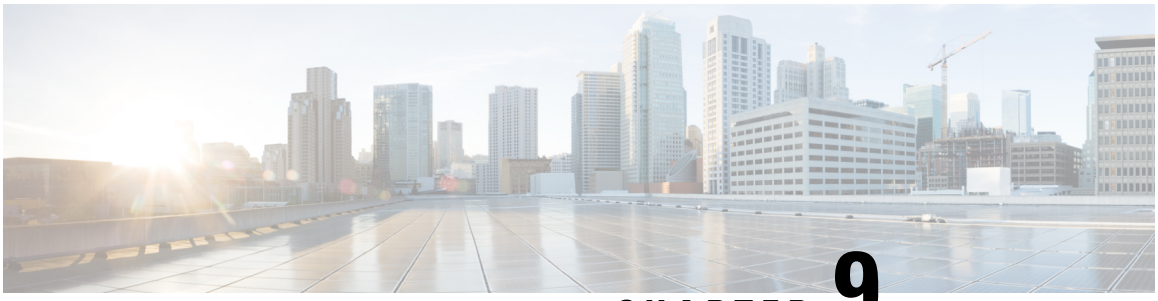
Configuring Dynamic MAC Address Detection for a Layer 3 Policy-Based Redirect Destination Using the REST API

The following REST API example enables dynamic MAC address detection for a Layer 3 policy-based redirect destination by specifying `00:00:00:00:00:00` for the MAC address:

```
<vnsSvcRedirectPol AnycastEnabled="no" destType="L3"
  dn="uni/tn-t0/svcCont/svcRedirectPol-TEST-PBR-POL" hashingAlgorithm="sip-dip-prototype"
  maxThresholdPercent="0" minThresholdPercent="0" name="TEST-PBR-POL"
  programLocalPodOnly="no" resilientHashEnabled="no" srcMacRewriteEnabled="no"
  thresholdDownAction="permit" thresholdEnable="no" userdom=":all:common:">
    <vnsRsIPSLAMonitoringPol tDn="uni/tn-t0/ipslaMonitoringPol-l3ping"
      userdom=":all:common:"/>
    <vnsRedirectDest ip="11.2.2.100" ip2="0.0.0.0" mac="00:00:00:00:00:00" podId="1"
      userdom=":all:common:">
        <vnsRsRedirectHealthGroup tDn="uni/tn-t0/svcCont/redirectHealthGroup-Test-HG"
          userdom=":all:common:"/>
    </vnsRedirectDest>
</vnsSvcRedirectPol>
```

Alternately, you can specify an empty value for `mac`:

```
    <vnsRedirectDest ip="11.2.2.100" ip2="0.0.0.0" mac="" podId="1" userdom=":all:common:">
```



CHAPTER 9

Configuring Direct Server Return

- About Direct Server Return, on page 131
- Example XML POST of Direct Server Return for Static Service Deployment, on page 135
- Direct Server Return for Static Service Deployment, on page 136
- Direct Server Return for Service Graph Insertion, on page 136
- Configuring the Citrix Server Load Balancer for Direct Server Return, on page 137
- Configuring a Linux Server for Direct Server Return, on page 137

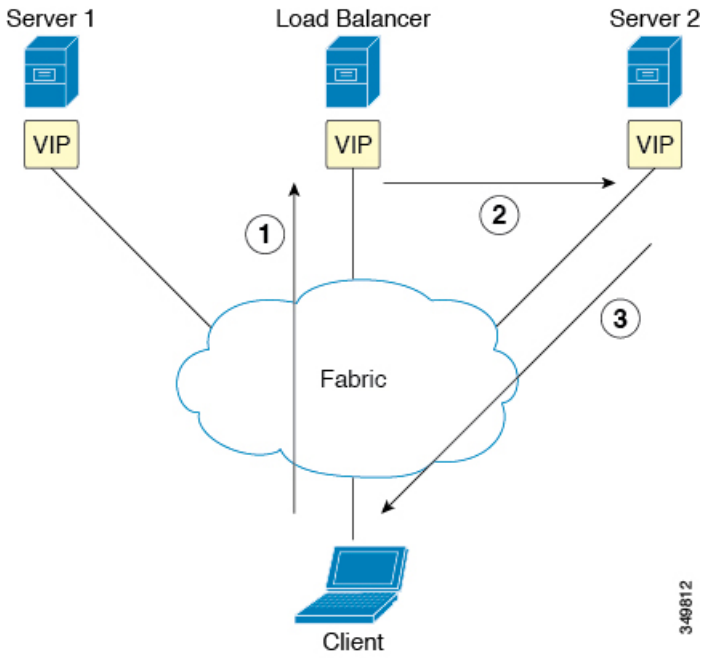
About Direct Server Return

The direct server return feature enables a server to respond directly to clients without having to go through the load balancer, which eliminates a bottleneck in the server-to-client path. In traditional load balancer deployments, the load balancer is in the path of the client-to-server communication: both the client-to-server request path and the server-to-client response path. While the amount of data in the requests from the client-to-server direction are relatively small, the server-to-client response traffic is much higher: approximately 10 times that of client-to-server request data. The load balancer in the path of this high volume response traffic becomes a bottleneck and adversely affects the communication.

For direct server return deployments, a virtual IP address is shared by the load balancer and server. Clients always address their request to the virtual IP address that is intended to reach the load balancer, and the direct response from the server-to-client use this virtual IP address as the source address. Cisco Application Centric Infrastructure (ACI) enabled with data-path learning of the IP source address poses problems when it learns the virtual IP address from the server-to-client traffic, leading to the disruption of Client-to-load balancer request traffic. To allow for the proper operation of a direct server return deployment, the ACI fabric must ensure that the request-response traffic between the communicating endpoints are delivered to their intended destination correctly. This requires that the data-path IP address learning on the leaf switches must be controlled in such a way that there is no interruption to client-to-load balancer, load balancer-to-server, and server-to-client traffic.

The following figure illustrates the data path in a direct server return deployment:

Figure 22: Direct Server Return High-Level Flow



1. The load balancer and all of the back-end servers are configured with the virtual IP address. The load balancer alone responds to Address Resolution Protocol (ARP) requests for this virtual IP address. After load balancing the client request, the load balancer re-writes the destination MAC address in the packet and forwards the MAC address to one of the back-end servers.
2. The virtual IP address is configured on the back-end server, but ARP is disabled to prevent back-end servers from responding to ARP requests for this virtual IP address.
3. The server sends the return traffic directly to the client, by-passing the load-balancer.

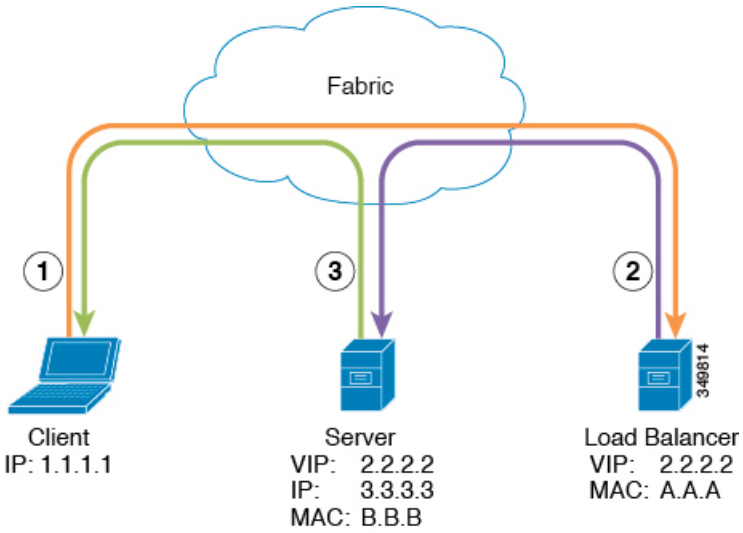
Layer 2 Direct Server Return

Layer 2 direct server return is the common or traditional deployment, also known as direct routing, SwitchBack, or nPath. In this deployment, the virtual IP address is shared by the load balancer and server. The load balancers and servers must be layer 2 adjacent. A layer 2 direct server return deployment has the following limitations:

- You lose flexibility in server placement
- You need an extra server configuration to suppress Address Resolution Protocol (ARP) responses to client virtual IP address requests
- Port selection is layer 3 and protocol dependent; port selection cannot happen at layer 2 (load balancer to server communication)

A layer 2 direct server return deployment has the following traffic flow:

Figure 23: Layer 2 Direct Server Return Traffic Flow



1. Client to load balancer

Source IP Address	1.1.1.1
Destination IP Address	2.2.2.2
Destination MAC Address	A.A.A

2. Load balancer to server

Source IP Address	1.1.1.1
Destination IP Address	2.2.2.2
Destination MAC Address	B.B.B

3. Server to client

Source IP Address	2.2.2.2
Destination IP Address	1.1.1.1
Destination MAC Address	MAC address of the default gateway

About Deploying Layer 2 Direct Server Return with Cisco Application Centric Infrastructure

The following information applies to deploying layer 2 direct server return with Cisco Application Centric Infrastructure (ACI):

- The virtual IP address (2.2.2.2) moves within the ACI fabric

  - The load balancer-to-server and server-to-client traffic with the same source virtual IP address (2.2.2.2)
  - The server-to-client traffic is routed; the traffic is addressed to the gateway MAC address in the fabric
  - The data-path learning of the source IP address from the server moves to the virtual IP address within the fabric
- There are no issues for the client IP address (1.1.1.1) appearing from difference sources
  - The client IP address appears as the source IP address from both the client and the load balancer in the fabric
  - The load balancer and server are layer 2 adjacent and the load balancer-to-server traffic is layer 2 forwarded
  - There is no data-path IP address learning from layer 2 forwarded traffic in the fabric
  - Even if the client IP address appears as the source IP address from the load balancer in the fabric, the client IP address is not learned

Guidelines and Limitations for Configuring Direct Server Return

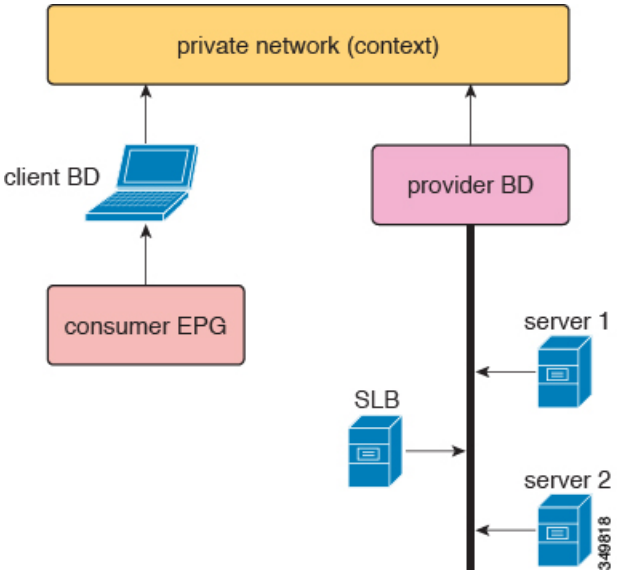
Follow these guidelines and limitations when deploying Direct Server Return:

- The VRF where the VIP is deployed must be set to "enforced" mode.
- The VRF must be set to "ingress" enforcement.
- Shared services are not supported for this configuration.
- EP Move Detection Mode: GARP-based Detection must be enabled for the bridge domain.
- Unicast routing must be enabled for the bridge domain.
- The EPG where the VIP is located must have a contract associated with it (the contract drives the configuration in hardware).
- The Layer 4 to Layer 7 VIP option is configurable only under EPG but not under the EPG collection for VRF (also known as vzAny).
- Client to VIP traffic must always go through the proxy spine.
- The load balancer must be in one-armed mode.
- The server and load balancer EPG must be the same device or the load balancer EPG must be deployed on all server EPG ToRs.
- The server EPG and load balancer EPG must be in the same bridge domain.
- Configuring a Layer 4 to Layer 7 virtual IP (VIP) address under microsegmented EPGs or their corresponding Base EPGs is not supported.

Supported Direct Server Return Configuration

The following figure illustrates the supported direct server return configuration:

Figure 24: Supported Direct Server Return Configuration



The following information applies to the supported configuration:

- The server load balancer and servers are in the same subnet and bridge domain
- The server load balancer should operate in 1 ARM mode; the inside and outside legs of server load balancer should point to the same bridge domain
- The consumer and provider endpoint groups should be under the same private network; no shared service configuration is supported

Example XML POST of Direct Server Return for Static Service Deployment

The following XML POST is an example of a direct server return (DSR) static service deployment:

```
<fvAp name="dev">
      <fvAEPg name="loadbalancer">
        <fvRsDomAtt tDn="uni/phys-{{tenantName}}"/>
        <fvRsBd tnFvBDName="lab"/>
        <fvVip addr="121.0.0.{{net}}" />
        <fvRsPathAtt tDn="topology/pod-1/paths-104/pathep-[eth1/1]" encap="vlan-33" />
        <fvRsProv tnVzBrCPName="loadBalancer"/>
        <fvRsCons tnVzBrCPName="webServer"/>
      </fvAEPg>
        <fvAEPg name="webServer">
          <fvRsDomAtt tDn="uni/phys-{{tenantName}}"/>
            <fvRsBd tnFvBDName="lab"/>
          <fvRsPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/1]" encap="vlan-34"/>
            <fvRsProv tnVzBrCPName="webServer"/>
        </fvAEPg>
      <fvAEPg name="client">
```

```
        <fvRsDomAtt tDn="uni/phys-{{tenantName}}"/>
        <fvRsBd tnFvBDName="lab"/>
        <fvRsPathAtt tDn="topology/pod-1/paths-103/pathep-[eth1/4]" encap="vlan-1114"/>
        <fvRsCons tnVzBrCPName="loadBalancer"/>
      </fvAEPg>
</fvAp>
```

The DSR configuration is downloaded to all top-of-rack switches (ToRs) where the EPG with a Layer 4 to Layer 7 virtual IP address is deployed or an EPG that has a contract with the EPG with a Layer 4 to Layer 7 virtual IP address is deployed, regardless of the contract direction. In the example, the DSR virtual IP address configuration is downloaded to the ToR nodes 101, 103, and 104. Node 104 has a load balancer EPG with a Layer 4 to Layer 7 virtual IP address configured. Nodes 101 and 103 have a webserver or client EPG, which has a contract to the load balancer EPG.

All ToRs that downloaded the DSR configuration do not learn the Layer 4 to Layer 7 virtual IP address from the datapath. Such ToRs also do not learn the Layer 4 to Layer 7 virtual IP address from the other EPGs. This is true even if you use Address Resolution Protocol (ARP), Gratuitous Address Resolution Protocol (GARP), or IPv6 Neighbor Discovery (ND). For example, the ToRs only learn the Layer 4 to Layer 7 virtual IP address from a load balancer EPG by way of the control plane. This restriction helps to prevent the erroneous learning of the Layer 4 to Layer 7 virtual IP address from a web server EPG, such as if you forgot to suppress ARP on a web server.

Direct Server Return for Static Service Deployment

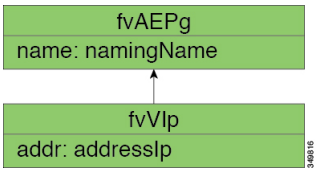
In the static service deployment mode, you configure the service flow by creating the appropriate application endpoint groups and contracts on a hop-by-hop basis.

Direct Server Return for Static Service Deployment Logical Model

You can configure the virtual IP addresses that are used by the load balancers by using the `fvVip` object under an application endpoint group (`fvAEPg`).

The following figure illustrates the logical model for static service deployment:

Figure 25: Static Service Deployment Logical Model



Direct Server Return for Service Graph Insertion

The Cisco Application Centric Infrastructure (ACI) provides automated service insertion by using vendor packages and service graphs. In this mode, the endpoint groups that are created for the service device legs, such as inside and outside endpoint groups, are created by the ACI without the operator configuration.

For service graph insertion, you must configure the virtual IP addresses under the appropriate logical interface context for the service device, as shown in the following example XML POST:

```
<vnsLDevCtx ctrctNameOrLbl="webCtrct"
  graphNameOrLbl="G1"
  nodeNameOrLbl="SLB">
    <vnsRsLDevCtxToLDev tDn="uni/tn-t1/lDevVip-InsiemeCluster"/>
    <vnsLIfCtx connNameOrLbl="inside">
        <vnsRsLIfCtxToBD tDn="uni/tn-t1/BD-t1BD1"/>
        <vnsRsLIfCtxToLIf tDn="uni/tn-t1/lDevVip-InsiemeCluster/lIf-inside"/>
    </vnsLIfCtx>
    <vnsLIfCtx connNameOrLbl="outside">
        <vnsRsLIfCtxToBD tDn="uni/tn-t1/BD-t1BD1"/>
        <vnsRsLIfCtxToLIf tDn="uni/tn-t1/lDevVip-InsiemeCluster/lIf-outside"/>
        <vnsSvcVip addr="9.9.9.9" />
        <vnsSvcVip addr="11.11.11.11" />
    </vnsLIfCtx>
</vnsLDevCtx>
```

The sample request configures two virtual IP addresses (9.9.9.9 and 11.11.11.11) on the outside leg of the server load balancer. The virtual IP address definition is under `LIfCtx` instead of being under an endpoint group as it is with a static direct server return configuration. This is because in the service graph case, operators do not have direct access to an endpoint group for the device legs, unlike with a static service deployment.

Direct Server Return Shared Layer 4 to Layer 7 Service Configuration

When the service device is configured in the common tenant or management tenant, the implicit model differs slightly. Instead of `vnsEPpInfo`, the service virtual IP address update managed object is created as a child of `vnsREPpInfo`. One `vnsSvcEpgCont` managed object is created per `vnsRsEPpInfo` to keep track of shared `SvcVip`s across tenants.

Configuring the Citrix Server Load Balancer for Direct Server Return

The following procedure provides an overview of how to configure the Citrix server load balancer for direct server return.

Step 1 Configure the virtual IP address on the backend server's loopback so that the backend server accepts the packets.

Step 2 Disable Address Resolution Protocol (ARP) reply for the virtual IP address on backend server.

Step 3 If necessary, disable the proxy port on services that are bound to the load balancing virtual server. The proxy port is disabled by default.

Step 4 Set the m parameter to "MAC" on the load balancing virtual server.

Step 5 Enable the USIP mode either globally or for each service.

Step 6 Enable the "L3", "USNIP", and "MBF" modes.

Step 7 Configure a route on the backend servers so that they can reach the Internet directly.

Configuring a Linux Server for Direct Server Return

The following procedure provides an overview of how to configure a Linux server for direct server return.

Step 1 Configure the virtual IP addresses on the loopback interfaces by creating the `/etc/sysconfig/network-scripts/ifcfg-lo` file in Centos with the following contents:

```
DEVICE=lo:1
IPADDRESS=10.10.10.99
NETMASK=255.255.255.255
NETWORK=10.10.10.99
BROADCAST=10.10.10.99
ONBOOT=yes
NAME=loopback
```

In this example, 10.10.10.99 is the virtual IP address.

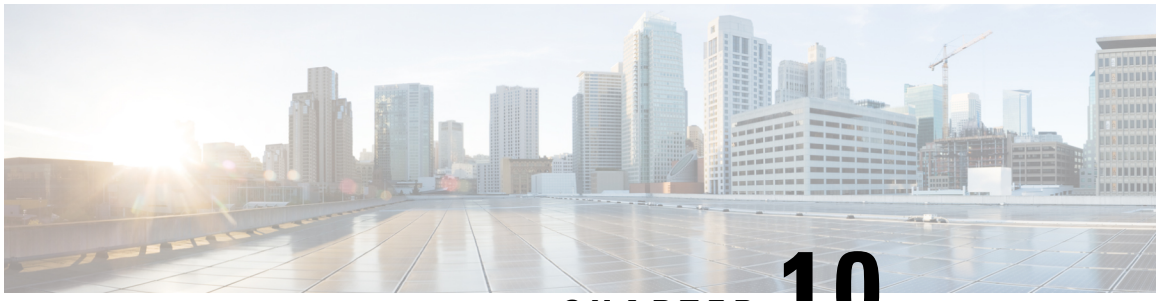
Step 2 Set the arp_ignore and arp_announce settings in the server interface that is used to reply to client request:

```
echo 1 > /proc/sys/net/ipv4/conf/eth1/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/eth1/arp_announce
```

In this example, `eth1` is the server interface used to respond to client requests.

For more information about the ARP settings, see the following Linux Virtual Server wiki page:

http://kb.linuxvirtualserver.org/wiki/Using_arp_announce/arp_ignore_to_disable_ARP



CHAPTER 10

Configuring Copy Services

- About Copy Services, on page 139
- Copy Services Limitations, on page 140
- Configuring Copy Services Using the GUI, on page 140
- Configuring Copy Services Using the NX-OS-Style CLI, on page 142
- Configuring Copy Services Using the REST API, on page 144

About Copy Services

Unlike SPAN that duplicates all of the traffic, the Cisco Application Centric Infrastructure (ACI) copy services feature enables selectively copying portions of the traffic between endpoint groups, according to the specifications of the contract. Broadcast, unknown unicast and multicast (BUM), and control plane traffic that are not covered by the contract are not copied. In contrast, SPAN copies everything out of endpoint groups, access ports or uplink ports. Unlike SPAN, copy services do not add headers to the copied traffic. Copy service traffic is managed internally in the switch to minimize impact on normal traffic forwarding.

A copy service is configured as part of a Layer 4 to Layer 7 service graph template that specifies a copy cluster as the destination for the copied traffic. A copy service can tap into different hops within a service graph. For example, a copy service could select traffic between a consumer endpoint group and a firewall provider endpoint group, or between a server load balancer and a firewall. Copy clusters can be shared across tenants.

Copy services require you to do the following tasks:

- Identify the source and destination endpoint groups.
- Configure the contract that specifies what to copy according to the subject and what is allowed in the contract filter.
- Configure Layer 4 to Layer 7 copy devices that identify the target devices and specify the ports where they attach.
- Use the copy service as part of a Layer 4 to Layer 7 service graph template.
- Configure a device selection policy that specifies which device will receive the traffic from the service graph. When you configure the device selection policy, you specify the contract, service graph, copy cluster, and cluster logical interface that is in copy device.

Copy Services Limitations

The following limitations apply when using the copy services feature:

- Copy services are only supported only on Cisco Nexus 9000-series switches with names that end in "-EX" or later, such as N9K-C93180LC-EX, N9K-C93108TC-FX, or N9K-93240YC-FX2.
- For data path traffic that is copied to the local and remote analyzer port, the Class of Service (CoS) and Differentiated Services Code Point (DSCP) values are not preserved in the copied traffic. This is because the contract with the copy action can be hit on either the ingress or egress leaf switch before or after the actual COS or DSCP value gets modified.

  When policing the data path traffic at a given endpoint ingress direction, the traffic that is copied is the actual incoming traffic before the traffic is policed. This is due to an ASIC limitation in the N9K-C93108TC-EX and N9K-C93180YC-EX switches.
- Copy services support only one device per copy cluster.
- A copy cluster supports only one logical interface.
- You can configure copy analyzers in the consumer endpoint or provider endpoint only in N9K-C93108TC-EX and N9K-C93180YC-EX switches.
- The `tn-common/ctx-copy` VRF instance, also known as the copy VRF instance, is a system-reserved context for a copy service. The copy VRF instance is auto-configured by the system during the boot up sequence. The copy VRF instance cannot be configured nor deleted by the user.
- Copy services with a vzAny contract is not supported.
- Copy service is not supported when deployed on local leaf and when source or destination is on the remote leaf. In this scenario routable TEP IP address is not allocated for local leaf switch.
- When using a separate copy device for each direction of a flow, you must have two different unidirectional filters.

Configuring Copy Services Using the GUI

This procedure uses the GUI to configure copy services.

Note When you configure a copy device, the context aware parameter is not used. The context aware parameter has a default value of `single context`, which can be ignored.

Step 1 Create one or more copy devices.

For information about creating a copy device, see Creating a Copy Device Using the GUI, on page 141.

Step 2 Create a service graph template to use for copy services.

For information about creating a service graph template, see Configuring a Service Graph Template Using the GUI, on page 44.

a) If you want to create one or more service nodes, drag Layer 4 to Layer 7 service devices from the **Device Clusters** section to in-between the consumer endpoint group and provider endpoint group.
b) Create one or more copy nodes by dragging copy devices from the **Device Clusters** section to in-between any two objects.

   The location where you drop the copy device becomes the point in the data flow from where the copy device will copy the traffic.

Step 3 Apply the Layer 4 to Layer 7 service graph template.

For information about applying a service graph template, see Applying a Service Graph Template to Endpoint Groups Using the GUI, on page 46.

Creating a Copy Device Using the GUI

A copy device is used as part of the copy services feature to create a copy node. A copy node specifies at which point of the data flow between endpoint groups to copy traffic.

This procedure only creates a copy device and does not configure anything else that is required to use the copy services feature. For information about configuring copy services, see Configuring Copy Services Using the GUI, on page 140.

Before you begin

You must have configured a tenant.

Step 1 On the menu bar, choose **Tenants** > **All Tenants**.
Step 2 In the Work pane, double-click the tenant's name.
Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Devices**.
Step 4 In the Work pane, choose **Actions** > **Create Copy Devices**.
Step 5 In the **Create Copy Devices** dialog box, in the **General** section, complete the following fields:

Name	Description
Name field	Enter a name for the copy device.
Device Type buttons	The device type. A copy device can only be a physical device.
Physical Domain drop-down list	Choose the physical domain for the device.

Step 6 In the **Device 1** section, click + to add a device interface, complete the following fields, and then click **Update**:

Name	Description
Name field	Enter a name for the device interface.
Path drop-down list	Choose a port, port channel, or virtual port channel for the device interface to use. The copy device connects to that port, port channel, or virtual port channel and copies traffic from it.

Step 7 In the **Cluster** section, click **+** to add a cluster interface, complete the following fields, and then click **Update**:

Name	Description
Name field	Enter a name for the cluster interface.
Concrete Interfaces drop-down list	Choose one or more concrete interfaces for the cluster interface to use.

Encap field	Enter a VLAN to use for encapsulation. The VLAN name format is as follows:
	`vlan-#`
	# is the VLAN's ID. For example:
	`vlan-12`

Step 8 Click **Submit**.

Configuring Copy Services Using the NX-OS-Style CLI

This procedure provides examples of using the CLI to configure copy services.

Note When you configure a copy device, the context aware parameter is not used. The context aware parameter has a default value of `single context`, which can be ignored.

Step 1 Create a copy cluster.

Example:

```
l4l7 cluster name Copy_1 type physical vlan-domain phys_scale_copy service COPY function none
  cluster-device Copy_1_Device_1
  cluster-interface Tap_copy vlan 3644
    member device Copy_1_Device_1 device-interface int1
      interface ethernet 1/15 leaf 104
      exit
    member device Copy_1_Device_1 device-interface int2
      interface ethernet 1/15 leaf 105
      exit
    member device Copy_1_Device_1 device-interface int3
      interface ethernet 1/20 leaf 105
      exit
    exit
  exit
```

Step 2 Create an abstract graph and device context, and then apply the graph.

Example:

```
l4l7 graph g5 contract c5
  service CP1 device-cluster-tenant t1 device-cluster Copy_1 mode OTHER service COPY
    connector copy cluster-interface Tap_copy
      exit
    exit
```

```
connection C1 terminal consumer terminal provider copyservice CP1 connector copy
Exit
```

Step 3 Attach the contract to the graph.

Example:

```
contract c5
  scope tenant
  subject Subject
    access-group default both
    l4l7 graph g5
    exit
  Exit
```

Step 4 Attach the endpoint groups to the contract.

Example:

```
epg epg2210
  bridge-domain member bd5
  contract consumer c5
  exit
epg epg2211
  bridge-domain member bd5
  contract provider c5
  Exit
```

Example

The following example creates a firewall service graph with a copy device on both sides:

```
tenant tenant_cmd_line
  l4l7 graph graph_fire contract fire
    service Fire device-cluster-tenant tenant_cmd_line device-cluster Fire mode FW_ROUTED

      connector consumer cluster-interface Outside_cmdline
        bridge-domain tenant tenant_cmd_line name Consumer_BD_1
        exit
      connector provider cluster-interface Inside_cmdline
        bridge-domain tenant tenant_cmd_line name Provider_BD1
        exit
      exit
     service CP2 device-cluster-tenant tenant_cmd_line device-cluster copy1 mode OTHER
      service COPY
       connector copy cluster-interface int1
        exit
       exit
     service CP3 device-cluster-tenant tenant_cmd_line device-cluster copy1 mode OTHER
      service COPY
       connector copy cluster-interface int1
        exit
       exit
     connection C1 terminal consumer service Fire connector consumer copyservice CP2
      connector copy
     connection C2 terminal provider service Fire connector provider copyservice CP3
      connector copy
     exit
   Exit
```

The following example creates a firewall and load balance in one-arm mode with copy devices attached in all the links:

```
l4l7 graph Graph_LB_Firewall contract c1_firewall
  service Fire device-cluster-tenant Tenant_Firewall_LB device-cluster Firewall_1 mode
   FW_ROUTED
    connector consumer cluster-interface Outside_Firewall
      bridge-domain tenant Tenant_Firewall_LB name BD1_Consumer
      exit
    connector provider cluster-interface Inside_Firewall
      bridge-domain tenant Tenant_Firewall_LB name BD2_Provider
      exit
    exit
 service LB device-cluster-tenant Tenant_Firewall_LB device-cluster LB_1 mode ADC_ONE_ARM

    connector consumer cluster-interface LB_Inside
      bridge-domain tenant Tenant_Firewall_LB name BD2_Provider
      exit
    connector provider cluster-interface LB_Inside
      bridge-domain tenant Tenant_Firewall_LB name BD2_Provider
      exit
    Exit
  service CP6 device-cluster-tenant Tenant_Pass2 device-cluster Copy_pass2 mode OTHER
   service-type COPY
    connector copy cluster-interface tap_copy
      exit
    Exit
  service CP7 device-cluster-tenant Tenant_Pass2 device-cluster Copy_pass2 mode OTHER
   service-type COPY
    connector copy cluster-interface tap_copy
      exit
    Exit
  service CP8 device-cluster-tenant Tenant_Pass2 device-cluster Copy_pass2 mode OTHER
   service-type COPY
    connector copy cluster-interface tap_copy
      exit
    exit
  connection C1 terminal consumer service Fire connector consumer copyservice CP6
   connector copy
  connection C2 intra-service service1 Fire connector1 provider service2 LB connector2
   consumer copyservice CP7 connector copy
  connection C3 terminal provider service LB connector provider copyservice CP8
   connector copy
  exit
exit
```

Configuring Copy Services Using the REST API

A copy device is used as part of the copy services feature to create a copy node. A copy node specifies at which point of the data flow between endpoint groups to copy traffic.

This procedure provides examples of using the REST API to configure copy services.

Note When you configure a copy device, the context aware parameter is not used. The context aware parameter has a default value of `single context`, which can be ignored.

Before you begin

You must have configured a tenant.

Step 1 Create a copy device.

Example:

```
<vnsLDevVip contextAware="single-Context" devtype="PHYSICAL" funcType="None" isCopy="yes"
  managed="no" mode="legacy-Mode" name="copy0" svcType="COPY" trunking="no">
    <vnsRsALDevToPhysDomP tDn="uni/phys-phys_scale_copy"/>
    <vnsCDev devCtxLbl="" name="copy_Dyn_Device_0" vcenterName="" vmName="">
        <vnsCIf name="int1" vnicName="">
            <vnsRsCIfPathAtt tDn="topology/pod-1/paths-104/pathep-[eth1/15]"/>
        </vnsCIf>
        <vnsCIf name="int2" vnicName="">
            <vnsRsCIfPathAtt tDn="topology/pod-1/paths-105/pathep-[eth1/15]"/>
        </vnsCIf>
    </vnsCDev>
    <vnsLIf encap="vlan-3540" name="TAP">
        <vnsRsCIfAttN tDn="uni/tn-t22/lDevVip-copy0/cDev-copy_Dyn_Device_0/cIf-[int2]"/>
        <vnsRsCIfAttN tDn="uni/tn-t22/lDevVip-copy0/cDev-copy_Dyn_Device_0/cIf-[int1]"/>
    </vnsLIf>
</vnsLDevVip>
```

Step 2 Create a logical device context (also known as a device selection policy).

Example:

```
<vnsLDevCtx ctrctNameOrLbl="c0" descr="" graphNameOrLbl="g0" name="" nodeNameOrLbl="CP1">
    <vnsRsLDevCtxToLDev tDn="uni/tn-t22/lDevVip-copy0"/>
    <vnsLIfCtx connNameOrLbl="copy" descr="" name="">
        <vnsRsLIfCtxToLIf tDn="uni/tn-t22/lDevVip-copy0/lIf-TAP"/>
    </vnsLIfCtx>
</vnsLDevCtx>
```

Step 3 Create and apply the copy graph template.

Example:

```
<vnsAbsGraph descr="" name="g0" ownerKey="" ownerTag="" uiTemplateType="UNSPECIFIED">
    <vnsAbsTermNodeCon descr="" name="T1" ownerKey="" ownerTag="">
        <vnsAbsTermConn attNotify="no" descr="" name="1" ownerKey="" ownerTag=""/>
        <vnsInTerm descr="" name=""/>
        <vnsOutTerm descr="" name=""/>
    </vnsAbsTermNodeCon>
    <vnsAbsTermNodeProv descr="" name="T2" ownerKey="" ownerTag="">
        <vnsAbsTermConn attNotify="no" descr="" name="1" ownerKey="" ownerTag=""/>
        <vnsInTerm descr="" name=""/>
        <vnsOutTerm descr="" name=""/>
    </vnsAbsTermNodeProv>
    <vnsAbsConnection adjType="L2" connDir="provider" connType="external" descr="" name="C1"
      ownerKey="" ownerTag="" unicastRoute="yes">
        <vnsRsAbsConnectionConns tDn="uni/tn-t22/AbsGraph-g0/AbsTermNodeCon-T1/AbsTConn"/>
        <vnsRsAbsConnectionConns tDn="uni/tn-t22/AbsGraph-g0/AbsTermNodeProv-T2/AbsTConn"/>
        <vnsRsAbsCopyConnection tDn="uni/tn-t22/AbsGraph-g0/AbsNode-CP1/AbsFConn-copy"/>
    </vnsAbsConnection>
    <vnsAbsNode descr="" funcTemplateType="OTHER" funcType="None" isCopy="yes" managed="no"
      name="CP1" ownerKey="" ownerTag="" routingMode="unspecified" sequenceNumber="0"
      shareEncap="no">
        <vnsAbsFuncConn attNotify="no" descr="" name="copy" ownerKey="" ownerTag=""/>
        <vnsRsNodeToLDev tDn="uni/tn-t22/lDevVip-copy0"/>
    </vnsAbsNode>
</vnsAbsGraph>
```

Step 4 Define the relation to the copy graph in the contract that is associated with the endpoint groups.

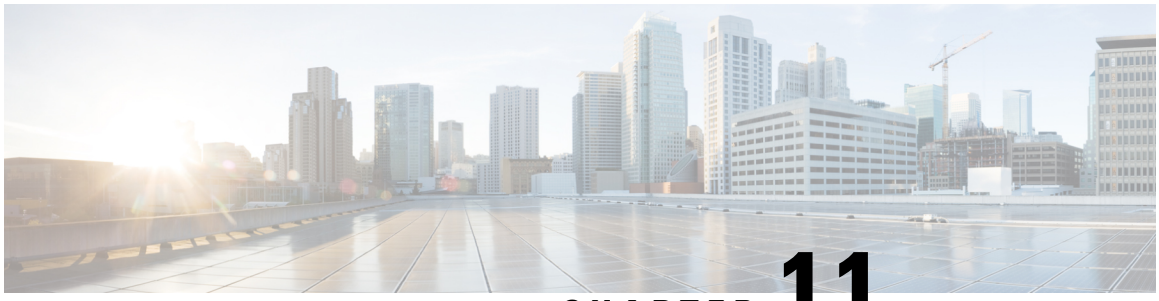
Example:

```
<vzBrCP descr="" name="c0" ownerKey="" ownerTag="" prio="unspecified" scope="tenant"
  targetDscp="unspecified">
    <vzSubj consMatchT="AtleastOne" descr="" name="Subject" prio="unspecified"
      provMatchT="AtleastOne" revFltPorts="yes" targetDscp="unspecified">
        <vzRsSubjFiltAtt directives="" tnVzFilterName="default"/>
        <vzRsSubjGraphAtt directives="" tnVnsAbsGraphName="g0"/>
    </vzSubj>
</vzBrCP>
```

Step 5 Attach the contract to the endpoint group.

Example:

```
<fvAEPg name="epg2860">
    <fvRsCons tnVzBrCPName="c0"/>
    <fvRsBd tnFvBDName="bd0"/>
    <fvRsDomAtt tDn="uni/phys-phys_scale_SB"/>
    <fvRsPathAtt tDn="topology/pod-1/paths-104/pathep-[PC_int2_g1]" encap="vlan-2860"
      instrImedcy="immediate"/>
</fvAEPg>
<fvAEPg name="epg2861">
    <fvRsProv tnVzBrCPName="c0"/>
    <fvRsBd tnFvBDName="bd0"/>
    <fvRsDomAtt tDn="uni/phys-phys_scale_SB"/>
    <fvRsPathAtt tDn="topology/pod-1/paths-105/pathep-[PC_policy]" encap="vlan-2861"
      instrImedcy="immediate"/>
</fvAEPg>
```



CHAPTER 11

Configuring Layer 4 to Layer 7 Resource Pools

- About Layer 4 to Layer 7 Resource Pools, on page 147
- About External and Public IP Address Pools, on page 147
- About External Layer 3 Routed Domains and the Associated VLAN Pools, on page 148
- About External Routed Networks, on page 148
- Creating an IP Address Pool for Layer 4 to Layer 7 Resource Pools Using the GUI, on page 149
- Creating a Dynamic VLAN Pool for Layer4 to Layer 7 Resource Pools Using the GUI, on page 149
- Creating an External Routed Domain for Layer 4 to Layer 7 Resource Pools Using the GUI, on page 150
- Preparing Layer 4 to Layer 7 Devices for Use in Layer 4 to Layer 7 Resource Pools, on page 150
- Validating the APIC Configuration of a Layer 4 to Layer 7 Device for Use in a Layer 4 to Layer 7 Resource Pool, on page 151
- Configuring the Device Management Network and Routes, on page 151
- Creating a Layer 4 to Layer 7 Resource Pool, on page 152
- Configuring a Layer 4 to Layer 7 Resource Pool Using the GUI, on page 153

About Layer 4 to Layer 7 Resource Pools

Layer 4 to Layer 7 resource pools bring together related configurations with regard to deploying Layer 4 to Layer 7 service devices. The related configuration is packaged together so that it can be used by orchestration layers such as Cisco Application Centric Infrastructure (Cisco ACI) Windows Azure Pack integration to deploy Layer 4 to Layer 7 service devices.

About External and Public IP Address Pools

For Layer 4 to Layer 7 resource pools created in Cisco APIC Release 3.0(x) and earlier, the public and external IP address pools were one and the same and were simply marked as external. For Layer 4 to Layer 7 resource pools created in Cisco APIC Release 3.1(x) and later, there is a separation and distinction between these two types of address pools. External IP address pools are used for the external interface of the Layer 4 to Layer 7 device, and L3Out SVI IP allocation. For Layer 4 to Layer 7 devices that are connected through a VPC into the fabric, 3 IP addresses are consumed by the L3Out configuration (side A primary IP address, side B primary IP address, and secondary IP address) while port channel and single interface connections consume 2 IP addresses (primary IP address and secondary IP address).

Public IP address pools are used to allocate dynamic NAT IP addresses (1 per tenant VRF), load balancers, virtual IP addresses (1 per tenant EPG), and additional public NAT IP addresses.

By separating the two IP address types, a Cisco APIC administrator is able to achieve the following:

- Export only the IP addresses in the IP pool marked as public - hiding the device-level interface IP addresses
- Incrementally add to the public IP address pool's varying blocks of IP addresses as they are acquired and available to the common tenant L3Out

About External Layer 3 Routed Domains and the Associated VLAN Pools

The external L3Out routed domain is used to provision the L3Out for both the internal and external connectors of the Layer 4 to Layer 7 devices. These L3Outs allow for traffic to originate from outside of the Cisco Application Centric Infrastructure (Cisco ACI) fabric and be able to reach the resources that are inside of the Cisco ACI fabric. The L3Outs also allow for traffic to originate from within the Cisco ACI fabric and be able to reach outside of the Cisco ACI fabric. The VLANs within the VLAN pool that are associated with the Layer 3 routed domain must be unique for a given leaf or VPC leaf switch pair where the Layer 4 to Layer 7 service devices are connected. If the Layer 4 to Layer 7 service devices span across multiple leaf or VPC leaf switch pairs, then the limitation also extends to these leaf and VPC leaf switch pairs.

Note VLAN blocks should not be reconfigured or removed from the VLAN pools once the Layer 4 to Layer 7 resource pool is in use. You can add VLAN blocks to the current VLAN block if required for expansion.

The following VLAN pool sizing considerations apply:

- 1 VLAN is dynamically allocated per external IP address pool
- 1 VLAN is dynamically allocated per tenant virtual forwarding and routing (VRF) that is accessing the Layer 4 to Layer 7 resource pool
- The external routed domain and the associated VLAN pool can be used across Layer 4 to Layer 7 resource pools

About External Routed Networks

For information about configuring external routed networks, see the *Cisco APIC Layer 3 Outside for Tenant Networks* document at the following URL:

http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

Creating an IP Address Pool for Layer 4 to Layer 7 Resource Pools Using the GUI

The following procedure creates an IP address pool for Layer 4 to Layer 7 resource pools using either GUI mode.

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

Step 2 In the **Navigation** pane, choose **Tenant Common** > **IP Address Pools**.

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

Step 4 In the **Create IP Address Pool** dialog box, fill in the fields as required.

Do not include the gateway address in the **Address Ranges**. The gateway address will be used as the secondary IP address of the Layer 4 to Layer 7 device external L3Out, which will act as a pervasive gateway.

Example:

- **Name**—`ExtIPPool1`
- **Gateway Address**— `132.121.101.1/24`
- **Address Block**
  - **From**—`132.121.101.2`
  - **To**—`132.121.101.200`

Step 5 Click **Submit**.

Creating a Dynamic VLAN Pool for Layer4 to Layer 7 Resource Pools Using the GUI

The following procedure creates a dynamic VLAN pool for Layer 4 to Layer 7 resource pools using the GUI mode.

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

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

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

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

a) For the **Allocation Mode** buttons, click **Dynamic Allocation**.

b) On the **Encap Blocks** table, click +.

c) In the **Create Ranges** dialog box, fill in the fields as specified below:

- In the **Range** fields, enter the desired VLAN range.

- For the **Allocation Mode** buttons, click **Inherit alloc mode from parent**.

d) Click **OK**.

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

Creating an External Routed Domain for Layer 4 to Layer 7 Resource Pools Using the GUI

The following procedure creates a dynamic VLAN pool for Layer 4 to Layer 7 resource pools using the GUI mode.

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

Step 2 In the **Navigation** pane, choose **Physical and External Domains** > **External Routed Domains**.

Step 3 In the **Work** pane, choose **Actions** > **Create Layer 3 Domain**.

Step 4 In the **Create Layer 3 Domain** dialog box, fill in the fields as required, except as specified below:

a) For the **Associated Attachable Entity Profile** drop-down list, choose the attachable entity profile to which all of the Layer 4 to Layer 7 service devices are connected.

b) For the **VLAN Pool** drop-down list, choose the dynamic VLAN pool that you created for Layer 4 to Layer 7 resource pools.

c) On the **Security Domains** table, add any required security domains.

Step 5 Click **Submit**.

Preparing Layer 4 to Layer 7 Devices for Use in Layer 4 to Layer 7 Resource Pools

To configure the physical connectivity of the Layer 4 to Layer 7 devices, see the appropriate configuration guide for each respective device regarding port channel or VPC configuration within the device.

Note For ASA55xx firewall devices that are context aware, the path configuration must be consistent across all the ASA contexts for a given physical ASA55xx. Configuring ASA contexts using different interfaces is not allowed in this configuration.

Validating the APIC Configuration of a Layer 4 to Layer 7 Device for Use in a Layer 4 to Layer 7 Resource Pool

The following procedure validates the Cisco Application Policy Infrastructure Controller (Cisco APIC) configuration of a Layer 4 to Layer 7 services device for use in Layer 4 to Layer 7 resource pools using the GUI mode.

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

Step 2 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Devices** > ***ASA_or_NetScaler_logical_device_name*** > ***concrete_device_name***.

Step 3 In the **Work** pane, choose the **Policy** tab.

Step 4 In the **Interfaces** table, verify that there are at least 2 interfaces, with each one mapping to a validate path (port, port channel, or vPC) in the fabric.

Step 5 For each ASA or NetScaler, verify that there is both a **Cluster > consumer** interface and a **Cluster > provider** interface defined. Even if the NetScalers will be used for internal load balancing, having such a configuration allows the tenant to use the NetScaler in both private and public IP address load balancing.

Step 6 For HA configurations, verify that there are 2 concrete interfaces for each cluster interface. Doing so will ensure that each port, port channel, or vPC will be configured correctly.

Configuring the Device Management Network and Routes

You must configure the management routes and remove the default route out of band directly on the Layer 4 to Layer 7 device.

The following example uses the Cisco Application Policy Infrastructure Controller (Cisco APIC) NX-OS-style CLI to configure the management route of an ASA firewall:

```
apic1(config)# route management 10.24.24.0 255.255.255.0 172.0.0.1
```

The following example uses the Cisco APIC NX-OS-style CLI to remove the default route:

```
apic1(config)# no route 0.0.0.0 0.0.0.0 172.0.0.1
```

The following example uses the Citrix NetScaler CLI to configure the management route of a NetScaler Application Delivery Controller (ADC) load balancer:

```
> add route 10.24.24.0 255.255.255.0 172.0.0.1
```

The following example uses the Citrix NetScaler CLI to remove the default route:

```
> rm route 0.0.0.0 0.0.0.0 172.0.0.1
```

Creating a Layer 4 to Layer 7 Resource Pool

Creating a Layer 4 to Layer 7 Resource Pool Using the GUI

The following procedure creates a Layer 4 to Layer 7 resource pool using the GUI mode. Once the resource pool has allocated various components for use by the tenants, you cannot modify to the resource pool. You can perform maintenance tasks such as adding IP address blocks, adding VLAN blocks, and adding logical devices, such as an ASA firewall or Citrix NetScaler load balancer.

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

Step 2 In the **Navigation** pane, choose **Tenant Common** > **Services** > **L4-L7** > **L4-L7 Resource Pools**.

Step 3 In the **Work** pane, choose **Actions** > **Create L4-L7 Resource Pool**.

Step 4 In the **Create L4-L7 Resource Pool** dialog box, fill in the fields as required, except as specified below:

a) In the **Private IP Address Subnet** field, enter the subnet that is used for internal device interface IP addresses, internal VIP addresses, and internal L3Out IP addresses.

b) For the **External IP Address Pool** drop-down list, choose the IP address pool that is used for the dynamic allocation of IP addresses used throughout the service graph and devices. You can create a new IP address pool if necessary. For **Connect Type**, choose **L3 External Network**.

c) For the **Public IP Address Pool** table, choose the IP address pool that is used for the dynamic allocation of IP addresses used for NAT IP addressing and VIP addressing. You can create a new IP address pool, if necessary. For **Connect Type**, choose **L3 External Network**.

d) For the **External Routed Domain** drop-down list, choose the external routed domain that you created for use in this Layer 4 to Layer 7 resource pool. You can create a new external routed domain if necessary.

e) In the **External Routed Networks** table, add the external routed networks that the tenants can consume.

The first external routed network will automatically be marked as `Default`. Only the default routed network is currently used.

f) In the **L4-L7 Devices** table, add the Layer 4 to Layer 7 devices that will be part of this Layer 4 to Layer 7 resource pool.

Step 5 Click **Submit**.

Creating a Layer 4 to Layer 7 Resource Pool Using the NX-OS-Style CLI

This section provides example commands for using the NX-OS-style CLI to configure Layer 4 to Layer 7 resource pools.

Step 1 Enter the configure mode.

```
apic1# configure
```

Step 2 Enter the configure mode for tenant common.

```
apic1(config)# tenant common
```

Step 3 Specify the Layer 4 to Layer 7 resource pool.

```
apic1(config)# l4l7 resource-pool <resource pool name>
```

Step 4 Set the resource pool version.

```
apic1(config-resource-pool)# version normalized
```

Note The version can be:

- **classic**: For resource pools created before Cisco Application Policy Infrastructure Controller (APIC) release 3.1(1).
- **normalized**: For resource pools created in or after Cisco APIC release 3.1(1).

Step 5 Associate an Layer 4 to Layer 7 devices to a resource pool.

```
apic1(config-resource-pool)# l4l7-cluster Dev-ASA-4
apic1(config-resource-pool)# l4l7-cluster Dev-MPX-4
```

Step 6 Associate an IP address pool as an external IP address pool to a resource pool.

```
apic1(config-resource-pool)# address-pool mininetExtPoolL3Ext l3-external
```

Step 7 (For normalized resource pools) Associate an IP address pool as a public IP address pool to a resource pool.

```
apic1(config-resource-pool)# public-address-pool mininetPubPoolL3Ext l3-external
```

Step 8 Associate to an external routed domain.

```
apic1(config-resource-pool)# external-routed-domain L3ServicesDom
```

Step 9 Configure the private IP address subnet for a resource pool.

```
apic1(config-resource-pool)# subnet 192.168.254.1/24
```

Step 10 Associate to an L3Out EPG in tenant common.

```
apic1(config-resource-pool)# l3out vpcDefaultInstP default
```

Configuring a Layer 4 to Layer 7 Resource Pool Using the GUI

Configuring Layer 4 to Layer 7 Devices in a Resource Pool

Adding Layer 4 to Layer 7 Devices to a Layer 4 to Layer 7 Resouce Pool

Note A dedicated VLAN will be consumed for each for each L3Out created for the tenant in their private VRF. The dynamic VLAN pool associated with the Layer 3 domain may need additional VLANs added to accommodate the additional devices to the resource pool.

You can add new Layer 4 to Layer 7 devices to the resource pool at any time.

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

Step 2 In the **Navigation** pane, choose **Tenant Common** > **Services** > **L4-L7** > **L4-L7 Resource Pools**.

The resource pools appear in the **Navigation** pane as a drop-down list under **L4-L7 Resource Pools**.

Step 3 Click the Layer 4 to Layer 7 resource pool to which you want to add a device.

Step 4 From the Work pane, click the **L4-L7 Devices** tab.

Step 5 From the **L4-L7 Devices** table, click the plus icon (+).

The **Create An L4-L7 Device** dialog appears.

Step 6 Click the **Device** drop-down arrow and choose a Layer 4 to Layer 7 device.

Step 7 Click **Submit**.

Removing Layer 4 to Layer 7 Devices from a Layer 4 to Layer 7 Resource Pool

The resource pool is unusable by any tenants without available Layer 4 to Layer 7 devices configured. If the L4-L7 Device is not allocated and exported to any tenants, perform the following:

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

Step 2 In the **Navigation** pane, choose **Tenant Common** > **Services** > **L4-L7** > **L4-L7 Resource Pools**.

The resource pools appear in the **Navigation** pane as a drop-down list under **L4-L7 Resource Pools**.

Step 3 Click the Layer 4 to Layer 7 resource pool with the device you want to remove.

Step 4 From the Work pane, click the **L4-L7 Devices** tab.

Step 5 Click to highlight the Layer 4 to Layer 7 device you want to remove then click the **trashcan** icon.

A confirmation dialog appears.

Step 6 Click **Yes** to confirm the deletion.

Configuring External IP Address Pools in a Resource Pool

Adding an External IP Address Pool to a Layer 4 to Layer 7 Resource Pool

If the resource pool is in use, do not remove or update the external IP address pool as it is in use by tenants.

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

Step 2 In the **Navigation** pane, choose **Tenant Common** > **Services** > **L4-L7** > **L4-L7 Resource Pools**.

The resource pools appear in the **Navigation** pane as a drop-down list under **L4-L7 Resource Pools**.

Step 3 Click the Layer 4 to Layer 7 resource pool to which you want to add an external IP address pool.

Step 4 From the Work pane, click the **Basic** tab.

Step 5 From the **External IP Address Pool** table, click the plus icon (**+**) .

The **External IP Address Pool** fields appear.

Step 6 Click the **Connect Type** drop-down arrow and choose **L3 External Network** then enter the appropriate values in the remaining **External IP Address Pool** fields.

Note For a description of a field, click the help icon (**?**) in the top-right corner.

Step 7 Click **Update**.

Removing an External IP Address Pool from a Layer 4 to Layer 7 Resource Pool

Note

- If the resource pool is in use, do not remove or update the external IP address pool as it is in use by tenants.
- If removing, adding, or updating the external IP address pool to handle IP address pool exhaustion, do not remove and add a larger IP address pool. In these situations, create a new Layer 4 to Layer 7 resource pool with a similar configuration such as Layer 3 domain and an L3Out but with a new external IP address pool.
- The resource pool is unusable by any tenants without an external IP address pool configured.

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

Step 2 In the **Navigation** pane, choose **Tenant Common** > **Services** > **L4-L7** > **L4-L7 Resource Pools**.

The resource pools appear in the **Navigation** pane as a drop-down list under **L4-L7 Resource Pools**.

Step 3 Click the Layer 4 to Layer 7 resource pool with the external IP address pool you want to remove.

Step 4 From the Work pane, click the **Basic** tab.

Step 5 From the **External IP Address Pool** table, click to highlight the external IP address pool you want to remove then click the **trashcan** icon.

A confirmation dialog appears.

Step 6 Click **Yes** to confirm the deletion.

Configuring Public IP Address Pools in a Resource Pool

Adding Public IP Address Pools to a Layer 4 to Layer 7 Resource Pool

Note
- For Layer 4 to Layer 7 resource pools created in Cisco APIC Release 3.0(x) and earlier, the external IP address pool is used as the public IP address pool and should not be modified once in use by any tenants.
- For Layer 4 to Layer 7 resource pools created in Cisco APIC Release 3.1(x) and later, you can add new public IP address pools to the resource pool at any time.
- The resource pool is unusable by any tenant without public IP address pools configured.

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

Step 2 In the **Navigation** pane, choose **Tenant Common** > **Services** > **L4-L7** > **L4-L7 Resource Pools**.

The resource pools appear in the **Navigation** pane as a drop-down list under **L4-L7 Resource Pools**.

Step 3 Click the Layer 4 to Layer 7 resource pool to which you want to add a public IP address pool.

Step 4 From the Work pane, click the **Basic** tab.

Step 5 From the **Public IP Address Pool** table, click the plus icon (**+**).

The **Public IP Address Pool** fields appear.

Step 6 Click the **Connect Type** drop-down arrow and choose **L3 External Network** then enter the appropriate values in the remaining **External IP Address Pool** fields.

Note For a description of a field, click the help icon (**?**) in the top-right corner.

Step 7 Click **Update**.

Removing Public IP Address Pools from a Layer 4 to Layer 7 Resource Pool

Note
- For Layer 4 to Layer 7 resource pools created in Cisco APIC Release 3.0(x) and earlier, the external IP address pool is used as the public IP address pool and should not be modified once in use by any tenants.
- For Layer 4 to Layer 7 resource pools created in Cisco APIC Release 3.1(x) and later, removing IP address pools from the resource pool should not be performed if any tenants are currently utilizing the IP address pool.
- The resource pool is unusable by any tenant if no public IP address pools configured.

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

Step 2 In the **Navigation** pane, choose **Tenant Common** > **Services** > **L4-L7** > **L4-L7 Resource Pools**.

The resource pools appear in the **Navigation** pane as a drop-down list under **L4-L7 Resource Pools**.

Step 3 Click the Layer 4 to Layer 7 resource pool with the public IP address pool you want to remove.

Step 4 From the Work pane, click the **Basic** tab.

Step 5 From the **Public IP Address Pool** table, click to highlight the public IP address pool you want to remove then click the **trashcan** icon.

A confirmation dialog appears.

Step 6 Click **Yes** to confirm the deletion.

Updating an External Routed Domain for a Layer 4 to Layer 7 Resource Pool

The resource pool is unusable by any tenant if no external routed domain is configured.

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

Step 2 In the **Navigation** pane, choose **Tenant Common** > **Services** > **L4-L7** > **L4-L7 Resource Pools**.

The resource pools appear in the **Navigation** pane as a drop-down list under **L4-L7 Resource Pools**.

Step 3 Click the Layer 4 to Layer 7 resource pool with the external routed domain you want to update.

Step 4 From the Work pane, click the **External** tab.

Step 5 Click the **External Routed Domain** drop-down arrow and choose a Layer 3 domain.

Step 6 Click **Submit**.

Updating External Routed Networks for a Layer 4 to Layer 7 Resource Pool

The resource pool is unusable by any tenant if no external routed networks are configured.

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

Step 2 In the **Navigation** pane, choose **Tenant Common** > **Services** > **L4-L7** > **L4-L7 Resource Pools**.

The resource pools appear in the **Navigation** pane as a drop-down list under **L4-L7 Resource Pools**.

Step 3 Click the Layer 4 to Layer 7 resource pool with the external routed network you want to update.

Step 4 From the Work pane, click the **External** tab.

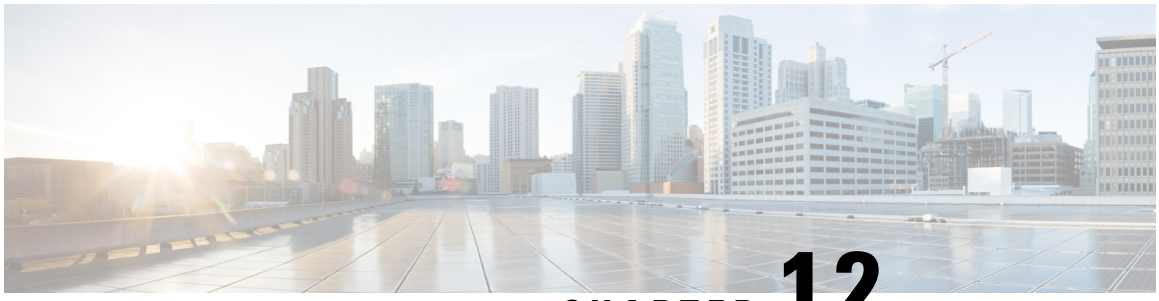
Step 5 From the **External Routed Networks** table, click the plus icon (+).

The **External Routed Networks** fields appear.

Step 6 Enter the appropriate value in the **External Routed Networks** fields.

Note For a description of a field, click the help icon (**?**) in the top-right corner.

Step 7 Click **Update**.



CHAPTER 12

Monitoring a Service Graph

- Monitoring a Service Graph Instance Using the GUI, on page 159
- Monitoring Service Graph Faults Using the GUI, on page 160
- Resolving Service Graph Faults, on page 160
- Monitoring a Virtual Device Using the GUI, on page 164
- Monitoring Device Cluster and Service Graph Status Using the NX-OS-Style CLI, on page 165

Monitoring a Service Graph Instance Using the GUI

After you configure a service graph template and attach the graph to an endpoint group (EPG) and a contract, you can monitor the service graph instance. Monitoring includes viewing the state of the graph instances, functions of a graph instance, resources allocated to a function, and parameters specified for a function.

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

Step 2 In the Work pane, double click the tenant's name for which you want to monitor its service graph.

Step 3 In the **Navigation** pane, choose **Tenant** *tenant_name* > **Services** > **L4-L7** > **Deployed Graph Instances**. The **Work** pane displays the following information about the active service graph instances:

Name	Description
Service Graph column	The name of the service graph template.
Contract column	The name of the contract that is shown in the service graph template.
Contained By column	The name of the network that contains the service graph template.
State column	The state of the service graph template. A state of **applied** means that the graph has been applied, and the graph policy is active within the fabric and the service device.
Description column	The description of the service graph.

Step 4 Expand the **Deployed Service Graphs** branch. The active service graph instances are listed under the branch.

Step 5 Click a service graph instance to view additional information about that instance in the **Work** pane. The default view is the topology of the graph. You can click one of the tabs in the **Work** pane to change the view for that graph.

Step 6 Expand the branch for one of the graph instances. The functions of the graph instance are listed under the instance.

Step 7 Click one of the functions to view additional information about that function in the **Work** pane. The default view is the policy of that function. You can click one of the tabs in the **Work** pane to change the view for that function. The **Work** pane displays the following information about the policy:

Name	Description
POLICY tab	The function's properties, resources allocated to the function, and the parameters of the function.
FAULTS tab	The issues that are happening on the function node.
HISTORY tab	The history of events that occurred on the function node.

Step 8 In the **Navigation** pane, click **Deployed Device**. The **Work** pane displays information about the device instances.

Monitoring Service Graph Faults Using the GUI

After you configure a service graph template and attach the graph to an endpoint group (EPG) and a contract, you can monitor a service graph template's faults.

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

Step 2 In the Work pane, double click the tenant's name for which you want to monitor its service graph.

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Deployed Graph Instances**.

Step 4 Expand the branch for a graph instance for which you want to view its faults. The functions of the graph instance are listed under the instance.

Step 5 Click on one of the functions. By default, the **Work** pane shows the policy of that function.

Step 6 Click the **FAULTS** tab in the **Work** pane. The **Work** pane displays the faults of the function node.

Resolving Service Graph Faults

After you have observed one or more service graph template faults, resolving the issue depends on the fault. The following tables describe the faults and provide how to resolve faults.

Table 3: Connector Faults

Fault	CLI Label	Description and Resolution
missing-connection	connection associated with a connector not found	The configuration for a graph connector is invalid. The associated connection for the connector could not be found.

Fault	CLI Label	Description and Resolution
missing-nodeinst	NodeInst associated with a connector not found	The configuration for a graph connector is invalid. The associated NodeInst for the connector could not be found.
conn-nonrenderable	Graph connector could not be rendered.	The configuration for a graph connector is invalid. The graph could not be rendered.
invalid-bd	BD associated with a connector is not valid	The configuration for a graph connector is invalid. The associated bridge domain for the connector is not valid.
invalid-ctx	Ctx associated with a connector is not valid.	The configuration for a graph connector is invalid. The associated Ctx for the connector is not valid.
missing-peer-conn	Peer connector associated with a connector not found.	Configuration for a graph connector is invalid. The peer connector for the connection could not be found.

Table 4: AbsGraph and GraphInst Faults

Fault	CLI Label	Description and Resolution
invalid-abstract-graph-config	invalid abstract graph config	The abstract graph configuration is invalid.
epp-download-failure	epp download failure	Graph policies failed to download to the switch.
param-duplicate-name-failure	duplicate param name	Multiple identical copies of a parameter were found with the same name.
id-allocation-failure	id allocation failure	A unique network resource (either VLAN or VXLAN) could not be allocated.
missing-ldev	No cluster found	A cluster could not be found.
context-cardinality-violation-failure	invalid cluster context cardinality	The cluster does not support the required tenancy(multi-tenant or single tenant).
function-type-mismatch-failure	invalid function type	The function type is not supported for the selected device. Check if the AbsNode functype and resolved LDevVip function type match.
missing-mparam	No parameter definition found	A required parameter definition could not be found.

Fault	CLI Label	Description and Resolution
missing-abs-graph	no abs graph found	The abstract graph configuration is missing for the graph instance.
invalid-param-config	invalid param config	The parameter configuration is invalid.
invalid-param-scope	invalid parameter scope	The parameter scope is invalid. Check the vnsRsScopeToTerm parameter in the AbsGraph to see if parameter is correct.
invalid-ldev	Invalid cluster	The cluster configuration is invalid. Check the status of the resolved LDevVip and correct the fault.
missing-tenant	no tenant found	The tenant could not be found for the graph.
internal-error	internal error	An internal error occurred during graph processing.
resource-allocation-failure	resource allocation failure	A required resource could not be allocated during graph processing.
missing-abs-function	no abstract function found	The abstract function definition is missing.
missing-mconn	No connector found	A required connector could not be found.
invalid-graphinst	invalid graphinst config	The graph instance is invalid.
missing-interface	no interface found	An interface could not be found.
missing-bd	no bd found	A bridge domain could not be found.
missing-terminal	Terminal node is missing a terminal	Terminal node is missing a terminal. Check the terminal node settings.
missing-namespace	no vlan/vxlan namespace found	The namespace that is needed to allocate the VLAN or VXLAN is missing. Verify that the resolved vnsLDevVip has the phyDomp parameter or the vmmDomp parameter configured that has a relation to the resolved fvnsVlanInstp.
missing-lif	no cluster interface found	A required cluster interface could not be found. Verify that the vnsLIf parameter in vnsLDevVip is configured correctly.

Fault	CLI Label	Description and Resolution
missing-cdev	No device found	The concrete device could not be found in the cluster. Verify that a valid vnsCDev is present under the resolved vnsLDevVip.
insufficient-devctx	Folder must have one value for each associated CDev	The folder is concrete device specific. The folder must have at least one value for each concrete device.
cdev-missing-cif	No interface defined	A concrete device must have at least one interface defined.
cdev-missing-pathinfo	Missing path for interface	For a physical service appliance, we must know to which leaf ports the interface is connected. Verify that the vnsCifPathAtt parameter is present for all vnsCIf under the resolved vnsCDev.
missing-cif	Device interfaces does not match cluster	The device interfaces should match the interfaces configured for their cluster. Verify that the vnsCIf parameter and the vnsLIf parameter are present under the resolved vnsLDevVip.
lif-invalid-CIf	LIf has an invalid CIf	The CIf contained by LIf is not present. Check the concrete device and CIf settings.
missing-function-node	Abstract graph missing function node	An abstract graph must have at least one function node.
graph-loop-detected	Abstract graph config has a loop	The abstract graph configuration is invalid. The configuration has a loop.
gothrough-routing-enabled-both	Both the legs of go through node has routing enabled	Both the legs of the go through node have routing enabled.
invalid-terminal-nodes	Abstract graph has invalid number of terminal nodes	An abstract graph must have at least two terminal nodes.
missing-ldev-ctx	No device context found for LDev	The device context for the device could not be found. Verify that vnsLDevCtx has values that match the contract, graph and node.
arp-flood-enabled	ARP flood is enabled on the management end point group	ARP flood must be disabled for the management endpoint group.
folderinst-validation-failed	FolderInst has key, that is not found in MFolder	The FolderInst's key and value should honor MFolder specifications.

Fault	CLI Label	Description and Resolution
paraminst-validation-failed	ParamInst has key and/or value, that are not found in MParam	ParamInst's key and value should honor MParam specifications.
invalid-mfolder	FolderInst points to an invalid MFolder	FolderInst must point to a valid MFolder.
invalid-mparam	ParamInst points to an invalid MParam	ParamInst must point to a valid MParam.
devfolder-validation-failed	DevFolder has key, that is not found in MFolder	DevFolders key and value should honor MFolder specifications.
devparam-validation-failed	DevParam has key and/or value, that are not found in MParam	DevParam's key and value should honor MParam specifications
cdev-missing-virtual-info	Virtual Object Info is missing in CDev	Virtual object information must be provided if LDevVip is of type Virtual.
invalid-rsmconnatt	Relationship to metaconnector is invalid	Correct the metaconnector DN and ensure it binds to the correct MDev hierarchy.

Monitoring a Virtual Device Using the GUI

After you configure a service graph template and attach the graph to an endpoint group (EPG) and a contract, you can monitor the virtual devices of a tenant. Monitoring the virtual devices tells you what devices are in use, which VLANs are configured for a device, the parameters passed to the devices, the statistics of the devices, and the health of the devices.

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

Step 2 In the Work pane, double click the tenant's name for which you want to monitor its service graph.

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Deployed Devices**.

Step 4 Click on one of the deployed devices. By default, the **Work** pane shows the policy of that deployed device. You can click the tabs in the **Work** pane to change the view. The tabs display the following information about the virtual device:

Tab	Description
POLICY tab	The device that is in use, the VLANs that are configured within the device, and the parameters that have been passed to the devices.
OPERATIONAL tab	The statistics that are being received from the various devices.
HEALTH tab	The health of the devices.

Monitoring Device Cluster and Service Graph Status Using the NX-OS-Style CLI

The commands in this section provide examples of how to monitor device cluster and service graph status using the NX-OS-style CLI.

Showing the Operation Information of a Device Cluster

The following command shows the operational information of a device cluster:

```
show l4l7-cluster tenant tenant_name cluster device_cluster_name
```

Example:

```
apic1# show l4l7-cluster tenant HA_Tenant1 cluster Firewall
tenant-graph : HA_Tenant1-g2,HA_Tenant1-g1


Device Cluster    : Firewall
Cluster Interface : consumer1
Encap             : vlan-501
Pctag             : 32773
Devices           : FW2(int),FW1(int)
Graphs            : HA_Tenant1-g1
Contracts         : HA_Tenant1-c1


Device Cluster    : Firewall
Cluster Interface : provider1
Encap             : vlan-502
Pctag             : 32774
Devices           : FW2(ext),FW1(ext)
Graphs            : HA_Tenant1-g1
Contracts         : HA_Tenant1-c1
```

Showing the Operation Status of a Device Cluster

The following command shows the operation status of a device cluster:

```
apic1# show l4l7-graph tenant tenant_name [graph graph_name]
```

Examples:

The following example gives high-level output of the status of the `HA_Tenant1` tenant:

```
apic1# show l4l7-graph tenant HA_Tenant1
Graph           : g1
Total Instances : 1
Encaps Used     : vlan-501,vlan-502,vlan-503,vlan-504
Device Used     : uni/tn-HA_Tenant1/lDevVip-Firewall

Graph           : g2
Total Instances : 1
Encaps Used     : vlan-501,vlan-502,vlan-503,vlan-504
Device Used     : uni/tn-HA_Tenant1/lDevVip-Firewall
```

The following example gives detailed output of the status of the `g1` service graph that is associated with the `HA_Tenant1` tenant:

```
apic1# show l4l7-graph tenant HA_Tenant1 graph g1
Graph           : HA_Tenant1-g1
Graph Instances : 1

Consumer EPg  : HA_Tenant1-consEPG1
Provider EPg  : HA_Tenant1-provEPG1
Contract Name : HA_Tenant1-c1
Config status : applied

Function Node Name : Node1
 Connector   Encap       Bridge-Domain  Device Interface
 ----------  ----------  ----------     --------------------
 consumer    vlan-3001   provBD1        consumer
 provider    vlan-3335   consBD1        provider
```

Showing the Faults of a Device Cluster

The following command shows the faults of a device cluster:

```
show faults l4l7-cluster
```

Example:

```
apic1# show faults l4l7-cluster
Code            : F0772
Severity        : minor
Last Transition : 2015-09-01T01:41:13.767+00:00
Lifecycle       : soaking-clearing
Affected object : uni/tn-ts1/lDevVip-d1/lIf-ext/fault-F0772
Description     : LIf configuration ext for L4-L7 Devices d1 for tenant ts1
                  is invalid.

Code            : F1085
Severity        : cleared
Last Transition : 2015-09-01T01:39:04.696+00:00
Lifecycle       : retaining
Affected object : uni/tn-ts1/lDevVip-d1/rsmDevAtt/fault-F1085
Description     : Failed to form relation to MO uni/infra/mDev-CiscoInternal-
                  NetworkOnly-1.0 of class vnsMDev

Code            : F1690
Severity        : minor
Last Transition : 2015-09-01T01:39:04.676+00:00
Lifecycle       : soaking
Affected object : uni/tn-ts1/lDevVip-d1/vnsConfIssue-missing-
                  namespace/fault-F1690
Description     : Configuration is invalid due to no vlan/vxlan namespace
                  found
```

Showing the Faults of a Service Graph

The following command shows the faults of a service graph:

```
show faults l4l7-graph
```

Example:

```
apic1# show faults l4l7-graph
Code            : F1690
Severity        : minor
Last Transition : 2015-11-25T20:07:33.635+00:00
Lifecycle       : raised
DN              : uni/tn-HA_Tenant1/AbsGraph-WebGraph/vnsConfIssue-invalid-
                  abstract-graph-config-param/fault-F1690
```

```
Description     : Configuration is invalid due to invalid abstract graph
                  config param
```

Showing the Running Configuration of a Device Cluster

The following command shows the running configuration of a device cluster:

```
show running-config tenant tenant_name l4l7 cluster
```

Example:

```
apic1# show running-config tenant common l4l7 cluster
# Command: show running-config tenant common l4l7 cluster
# Time: Thu Nov 26 00:35:59 2015
  tenant common
    l4l7 cluster name ifav108-asa type physical vlan-domain phyDom5 service FW function
go-through
      cluster-device C1
      cluster-interface consumer_1
        member device C1 device-interface port-channel1
          interface vpc VPCPolASA leaf 103 104
          exit
        exit
      cluster-interface provider_1
        member device C1 device-interface port-channel1
          interface vpc VPCPolASA leaf 103 104
          exit
        exit
      exit
```

Showing the Running Configuration of a Service Graph

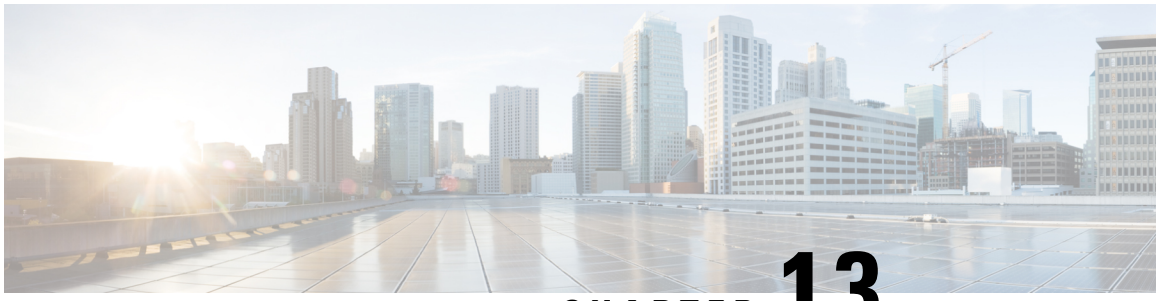
The following command shows the running configuration of a service graph:

```
show running-config tenant tenant_name l4l7 graph
```

Example:

```
apic1# show running-config tenant common l4l7 graph
# Command: show running-config tenant common l4l7 graph
# Time: Thu Nov 26 00:35:59 2015
  tenant T1
    l4l7 graph Graph-Citrix contract Contract-Citrix
     service N1 device-cluster-tenant common device-cluster ifav108-citrix mode ADC_ONE_ARM

        connector provider cluster-interface pro
          bridge-domain tenant common name BD4-Common
          exit
        connector consumer cluster-interface pro
          bridge-domain tenant common name BD4-Common
          exit
        exit
      connection C1 terminal consumer service N1 connector consumer
      connection C2 terminal provider service N1 connector provider
      exit
```



CHAPTER 13

Configuring Multi-Tier Application with Service Graph

- About Multi-Tier Application with Service Graph, on page 169
- Creating a Multi-Tier Application Profile Using the GUI, on page 169

About Multi-Tier Application with Service Graph

The Multi-Tier Application with Service Graph Quick Start dialog provides a consolidated method of configuring service graph components such as bridge domains, EPGs, VRFs, services, and contracts. As opposed to configuring each object in different locations in the Cisco APIC, the Quick Start dialog gathers the necessary configurations and combines them into a simple, organized step-by-step process.

Creating a Multi-Tier Application Profile Using the GUI

Before you begin

Configure the following objects before or, if available, while performing the procedure:

- **Tenants:** Configure at least one tenant before performing the procedure.
- **VMM Domain Profile:** If you will use virtual service devices, configure a Virtual Machine Manager (VMM) domain profile and a VM in the Layer 4 to Layer 7 device cluster (on which the device is hosted).
- **External Routed Network:** If you will connect a service device to an external routed network, configure a Layer 3 outside (L3Out) network.

Step 1 Access the Quick Start **Multi-Tier Application** dialog:

a) On the menu bar, click **Tenant** > **All Tenants**.
b) In the All Tenants Work pane, double-click the tenant's name.
c) In the Navigation pane, choose **Tenant** *tenant_name* > **Quick Start** > **Multi-tier Application**.
d) In the work pane, click **Configure Multi-tier Application**.
The **Create Application Profile** dialog appears.
e) Click **Start**.

Step 2 In the **STEP 2 > EPGs** dialog, configure the basics of the profile and design your Bridge Domain and EPGs:

a) In the **Application Profile** field, enter a unique name for the profile.

b) (Optional) If one or more devices in this profile are to be virtual, choose a Virtual Machine Manager (VMM) domain profile from the **VMM Domain Profile** drop-down list.

Note A VMM domain profile must be created (**Virtual Networking** > **VMM Domains**) prior to attempting this step in order for it to appear and be selected in the **VMM Domain Profile** drop-down list.

c) (Optional) If the consumer or provider EPG belongs to an external routed network, choose the network from the drop-down list for the **Consumer L3 Outside** and/or the **Provider L3 Outside** field(s).

Note An external routed network must be created (**Tenants** > *tenant* > **Networking** > **External Routed Networks**) prior to attempting this step in order for it to appear and be selected in the **L3 Outside** drop-down lists.

d) For the Bridge Domain buttons, determine if the EPG gateway IP address will be a single shared subnet or will be configured per EPG.

If you chose **Shared**, the **Shared Gateway IP** field appears. If you chose **Per EPG**, continue with step f.

e) If you chose **Shared** from the **Bridge Domain** buttons, enter the IPv4 address of the gateway to be shared by the EPGs in the **Shared Gateway IP** field.

f) In the Application Tiers (EPGs) **Name** field, enter a name for the EPG.

g) If you chose **Per EPG** from the **Bridge Domain** buttons, enter the IPv4 address of the gateway to be used by the EPG. If you chose **Shared** from the **Bridge Domain** buttons, the IP address that you entered in the **Shared Gateway IP** field is displayed.

h) (Optional) Click **+** to add another EPG and configure the EPG according to step g. Repeat this step if a third EPG is required.

i) Click **Next**.

Step 3 In the **STEP 3 > Services** dialog, optionally configure the inclusion of services adjacent to your EPGs:

a) (Optional) Put a check in the **Share same device** box to share the firewall or load balancer devices across all EPGs.

b) (Optional) Between each EPG, select the firewall (**FW**) or load balancer (**ADC**) device to include in this profile.

c) (Optional) If you add more than one device between an EPG, click **< Toggle >** to reposition the devices.

d) Click **Next**.

Step 4 (Firewall and Load Balancer) In the **STEP 4 >** dialog and the Firewall or Load Balancer Configuration section, configure service devices:

a) For the **Device Type** buttons, choose **Physical** or **Virtual**.

b) If you chose **Physical** for the **Device Type**, choose a domain from the **Physical Domain** drop-down list. If you chose **Virtul** for the **Device Type**, choose a domain from the **VMM Domain** drop-down list and the virtual machine (VM) on which the device is hosted from the **Device 1 VM** drop-down list.

c) For the **Node Type** buttons, chhoose **One-Arm** or **Two-Arm**. This determines if the device has only a consumer connector (one-arm) or consumer and provider connectors (two-arm).

d) For the **View** buttons, choose **Single Node** or **HA Node**. If you chose **HA Node**, a second interface (physical devices) or a second VNIC (virtual devices) is included in the connector configuration. for virtual devices, you must also choose a second virtual machine.

Step 5 (Firewall only) In the **STEP 4 >** dialog and the Consumer and Provider section, configure the firewall consumer and provider connectors:

a) In the **IP** field, for a physical device, enter the consumer/provider interface IP address of the Layer 4 to Layer 7 policy based redirect policy for firewall devices. For a virtual device, enter the consumer/provider interface IP address.

b) In the **MAC** field, enter MAC address of the LLayer 4 to Layer 7 policy based redirect policy for firewall devices.

c) In the **Gateway IP** field, enter the route gateway IP address.
d) For a physical device, in the **Device 1 Interface** drop-down list, choose an interface. For a virtual device, in the **Device 1 VNIC** drop-down list, choose a vNIC. If you chose **HA Node** for from the **View** buttons, you must choose a second vNIC in the **Device 2 VNIC** drop-down list.
e) (Physical device only) In the **Encap** field, enter the port encapsulation for the interface.

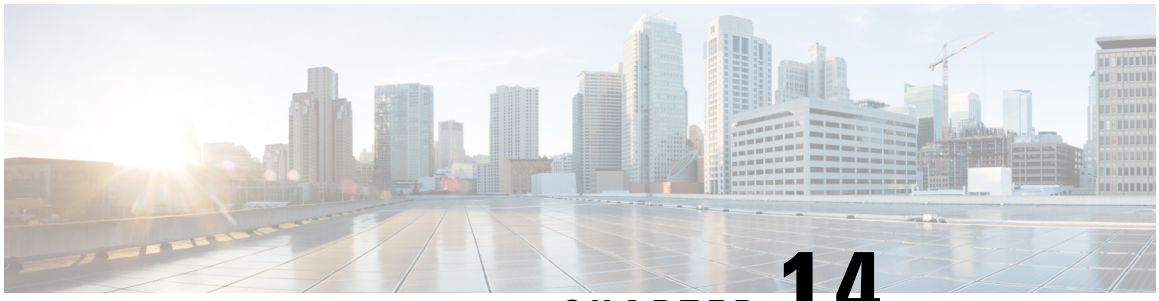
Step 6 (Load Balancer only) In the **STEP 4 >** dialog and the Consumer and Provider section, configure load balancer consumer and provider connectors:

a) In the **Gateway IP** field, enter the route gateway IP address.
b) For a physical device, in the **Device 1 Interface** drop-down list, choose an interface. For a virtual device, in the **Device 1 VNIC** drop-down list, choose a vNIC. If you chose **HA Node** for from the **View** buttons, you must choose a second vNIC in the **Device 2 VNIC** drop-down list.
c) (Physical device only) In the **Encap** field, enter the port encapsulation for the interface.
d) Leave the check in the **L3 Destination (VIP)** box to terminate L3 traffic on the connector. Remove the check if the connector is not an L3 destination.

Note The default for this parameter is enabled (checked). However, this setting is not considered if policy-based redirect is configured on the interface.

Step 7 If you have any additional devices to configure, click **Next** and repeat steps 4 through 6 for each device.

Step 8 Click **Finish**.



CHAPTER 14

Configuring Administrator Roles for Managing a Service Configuration

- About Privileges, on page 173
- Configuring a Role for Device Management, on page 174
- Configuring a Role for Service Graph Template Management, on page 174
- Configuring a Role for Exporting Devices, on page 174

About Privileges

You can grant privileges to the roles that you configure in the Application Policy Infrastructure Controller (APIC). Privileges determine what tasks a role is allowed to perform. You can grant the following privileges to the administrator roles:

Privilege	Description
nw-svc-policy	The network service policy privilege enables you to do the following: • Create a service graph template • Attach a service graph template to an application endpoint group (EPG) and a contract • Monitor a service graph
nw-svc-device	The network service device privilege enables you to do the following: • Create a device • Create a concrete device • Create a device context

Configuring a Role for Device Management

To enable a role to manage devices, you must grant the following privilege to that role:

- nw-svc-device

Configuring a Role for Service Graph Template Management

To enable a role to manage service graph templatess, you must grant the following privilege to that role:

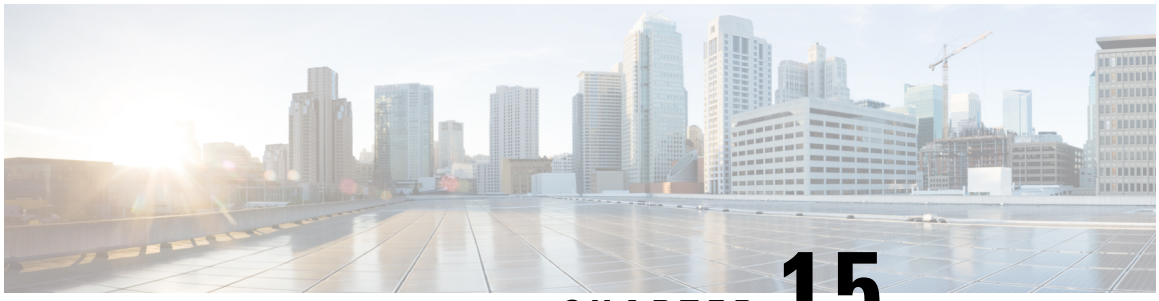
- nw-svc-policy

Configuring a Role for Exporting Devices

Devices can be exported to enable sharing of devices among tenants. A tenant with the role **nw-device** can create a device. If the tenant that owns the device wants to share these with another tenant, the sharing requires the **nw-svc-devshare** privilege.

The **nw-svc-devshare** privilege allows a tenant to be able to export devices.

Note To be able to use imported devices, other tenants that have imported devices need to have the **nw-svc-policy** privilege.



CHAPTER 15

Developing Automation

- About the REST APIs, on page 175
- Examples of Automating Using the REST APIs, on page 176

About the REST APIs

Automation relies on the Application Policy Infrastructure Controller (APIC) northbound Representational State Transfer (REST) APIs. Anything that can be done through the Cisco APIC GUI can also be done using XML-based REST POSTs using the northbound APIs. For example, you can monitor events through those APIs, dynamically enable EPGs, and add policies.

You can also use the northbound REST APIs to monitor for notifications that a device has been brought onboard, and to monitor faults. In both cases, you can monitor events that trigger specific actions. For example, if you see faults that occur on a specific application tier and determine that there is a loss of connectivity and a leaf node is going down, you can trigger an action to redeploy those applications somewhere else. If you have certain contracts on which you detect packet drops occurring, you could enable some copies of those contracts on the particular application. You can also use a statistics monitoring policy, where you monitor certain counters because of issues that have been reported.

For information on how to construct the XML files submitted to the Cisco APIC northbound API, see *Cisco APIC Layer 4 to Layer 7 Device Package Development Guide*.

The following Python APIs, defined in the *Cisco APIC Management Information Model Reference* can be used to submit REST POST calls using the northbound API:

- `vns:LDevVip`: Upload a device cluster
- `vns:CDev`: Upload a device
- `vns:LIf`: Create logical interfaces
- `vns:AbsGraph`: Create a graph
- `vz:BrCP`: Attach a graph to a contract

Note For endpoint security groups (ESGs), you can use the same service graph deployment REST APIs that are available for endpoint groups. However, you must associate the contract to the ESGs.

Examples of Automating Using the REST APIs

This section contains examples of using the REST APIs to automate tasks.

The following REST request creates a tenant with a broadcast domain, a Layer 3 network, application endpoint groups, and an application profile:

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">

        <!—L3 Network-->
        <fvCtx name="MyNetwork"/>

        <!-- Bridge Domain for MySrvr EPG -->
        <fvBD name="MySrvrBD">
            <fvRsCtx tnFvCtxName="MyNetwork"/>
            <fvSubnet ip="10.10.10.10/24">
            </fvSubnet>
        </fvBD>

        <!-- Bridge Domain for MyClnt EPG -->
        <fvBD name="MyClntBD">
            <fvRsCtx tnFvCtxName="MyNetwork"/>
            <fvSubnet ip="20.20.20.20/24">
            </fvSubnet>
        </fvBD>

        <fvAp dn="uni/tn-acme/ap-MyAP" name="MyAP">

            <fvAEPg dn="uni/tn-acme/ap-MyAP/epg-MyClnt" name="MyClnt">
                <fvRsBd tnFvBDName="MySrvrBD"/>
                <fvRsDomAtt tDn="uni/vmmp-Vendor1/dom-MyVMs"/>
                <fvRsProv tnVzBrCPName="webCtrct"> </fvRsProv>
                <fvRsPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/21]"
                  encap="vlan-202"/>
                <fvRsPathAtt tDn="topology/pod-1/paths-18/pathep-[eth1/21]"
                  encap="vlan-202"/>
           </fvAEPg>

            <fvAEPg dn="uni/tn-acme/ap-MyAP/epg-MySRVR" name="MySRVR">
                <fvRsBd tnFvBDName="MyClntBD"/>
                <fvRsDomAtt tDn="uni/vmmp-Vendor1/dom-MyVMs"/>
                <fvRsCons tnVzBrCPName="webCtrct"> </fvRsCons>
                <fvRsPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/21]"
                  encap="vlan-203"/>
                <fvRsPathAtt tDn="topology/pod-1/paths-18/pathep-[eth1/21]"
                  encap="vlan-203"/>
            </fvAEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

The following REST request creates a VLAN namespace:

```
<polUni>
    <infraInfra>
        <fvnsVlanInstP name="MyNS" allocMode="dynamic">
            <fvnsEncapBlk name="encap" from="vlan-201" to="vlan-300"/>
        </fvnsVlanInstP>
    </infraInfra>
</polUni>
```

The following REST request creates a VMM domain:

```
<polUni>
    <vmmProvP vendor="Vendor1">
        <vmmDomP name="MyVMs">
            <infraRsVlanNs tDn="uni/infra/vlanns-MyNS-dynamic"/>
            <vmmUsrAccP name="admin" usr="administrator" pwd="in$1eme"/>
            <vmmCtrlrP name="vcenter1" hostOrIp="192.168.64.186">
                <vmmRsAcc tDn="uni/vmmp-Vendor1/dom-MyVMs/usracc-admin"/>
            </vmmCtrlrP>
        </vmmDomP>
    </vmmProvP>
</polUni>
```

The following REST request creates a physical domain:

```
<polUni>
    <physDomP name="phys">
        <infraRsVlanNs tDn="uni/infra/vlanns-MyNS-dynamic"/>
    </physDomP>
</polUni>
```

The following REST request creates a device cluster:

```
<polUni>
    <fvTenant name="HA_Tenant1">

        <vnsLDevVip name="ADCCluster1" devtype="VIRTUAL" managed="no">
            <vnsRsALDevToDomP tDn="uni/vmmp-VMware/dom-mininet"/>
        </vnsLDevVip>

    </fvTenant>
</polUni>
```

The following REST request creates a device cluster context:

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vnsLDevCtx ctrctNameOrLbl="webCtrct" graphNameOrLbl="G1" nodeNameOrLbl="Node1">
            <vnsRsLDevCtxToLDev tDn="uni/tn-acme/lDevVip-ADCCluster1"/>
            <vnsLIfCtx connNameOrLbl="ssl-inside">
                <vnsRsLIfCtxToLIf tDn="uni/tn-acme/lDevVip-ADCCluster1/lIf-int"/>
            </vnsLIfCtx>
            <vnsLIfCtx connNameOrLbl="any">
                <vnsRsLIfCtxToLIf tDn="uni/tn-acme/lDevVip-ADCCluster1/lIf-ext"/>
            </vnsLIfCtx>
        </vnsLDevCtx>
    </fvTenant>
</polUni>
```

The following REST request creates a device cluster context used in route peering:

```
<polUni>
    <fvTenant dn="uni/tn-coke{{tenantId}}" name="coke{{tenantId}}">
        <vnsLDevCtx ctrctNameOrLbl="webCtrct1" graphNameOrLbl="WebGraph"
          nodeNameOrLbl="FW">
            <vnsRsLDevCtxToLDev tDn="uni/tn-tenant1/lDevVip-Firewall"/>
            <vnsLIfCtx connNameOrLbl="internal">
                <vnsRsLIfCtxToInstP tDn="uni/tn-tenant1/out-OspfInternal/instP-IntInstP"
                  status="created,modified"/>
                <vnsRsLIfCtxToLIf tDn="uni/tn-tenant1/lDevVip-Firewall/lIf-internal"/>
            </vnsLIfCtx>
            <vnsLIfCtx connNameOrLbl="external">
                <vnsRsLIfCtxToInstP tDn="uni/tn-common/out-OspfExternal/instP-ExtInstP"
                  status="created,modified"/>
                <vnsRsLIfCtxToLIf tDn="uni/tn-tenant1/lDevVip-Firewall/lIf-external"/>
```

```
            </vnsLIfCtx>
        </vnsLDevCtx>
    </fvTenant>
</polUni>
```

Note For information about configuring external connectivity for tenants (a Layer 3 outside), see the *Cisco APIC Basic Configuration Guide*.

The following REST request adds a logical interface in a device cluster:

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vnsLDevVip name="ADCCluster1">
            <vnsLIf name="C5">
                <vnsRsMetaIf tDn="uni/infra/mDev-Acme-ADC-1.0/mIfLbl-outside"/>
                <vnsRsCIfAtt tDn="uni/tn-acme/lDevVip-ADCCluster1/cDev-ADC1/cIf-int"/>
            </vnsLIf>
            <vnsLIf name="C4">
                <vnsRsMetaIf tDn="uni/infra/mDev-Acme-ADC-1.0/mIfLbl-inside"/>
                <vnsRsCIfAtt tDn="uni/tn-acme/lDevVip-ADCCluster1/cDev-ADC1/cIf-ext"/>
            </vnsLIf>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

The following REST request adds a concrete device in a physical device cluster:

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vnsLDevVip name="ADCCluster1">
            <vnsCDev name="ADC1" devCtxLbl="C1">
                <vnsCIf name="int">
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/22]"/>
                </vnsCIf>
                <vnsCIf name="ext">
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/21]"/>
                </vnsCIf>
                <vnsCIf name="mgmt">
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/20]"/>
                </vnsCIf>
            </vnsCDev>
            <vnsCDev name="ADC2" devCtxLbl="C2">
                <vnsCIf name="int">
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/23]"/>
                </vnsCIf>
                <vnsCIf name="ext">
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/24]"/>
                </vnsCIf>
                <vnsCIf name="mgmt">
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/30]"/>
                </vnsCIf>
            </vnsCDev>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

The following REST request adds a concrete device in a virtual device cluster:

```
<polUni>
    <fvTenant dn="uni/tn-coke5" name="coke5">
        <vnsLDevVip name="Firewall5" devtype="VIRTUAL">
            <vnsCDev name="ASA5" vcenterName="vcenter1" vmName="ifav16-ASAv-scale-05">
```

```
                <vnsCIf name="Gig0/0" vnicName="Network adapter 2"/>
                <vnsCIf name="Gig0/1" vnicName="Network adapter 3"/>
                <vnsCIf name="Gig0/2" vnicName="Network adapter 4"/>
                <vnsCIf name="Gig0/3" vnicName="Network adapter 5"/>
                <vnsCIf name="Gig0/4" vnicName="Network adapter 6"/>
                <vnsCIf name="Gig0/5" vnicName="Network adapter 7"/>
                <vnsCIf name="Gig0/6" vnicName="Network adapter 8"/>
                <vnsCIf name="Gig0/7" vnicName="Network adapter 9"/>
            </vnsCDev>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

The following REST request creates a service graph:

```
<polUni>
    <fvTenant name="HA_Tenant1">
        <vnsAbsGraph name="g1">

            <vnsAbsTermNodeProv name="Input1">
                <vnsAbsTermConn name="C1">
                </vnsAbsTermConn>
            </vnsAbsTermNodeProv>

            <!-- Node1 Provides LoadBalancing functionality -->
            <vnsAbsNode name="Node1" managed="no">
                <vnsRsDefaultScopeToTerm
                  tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeProv-Input1/outtmnl"/>
                <vnsAbsFuncConn name="outside" attNotify="true">
                </vnsAbsFuncConn>
                <vnsAbsFuncConn name="inside" attNotify="true">
                </vnsAbsFuncConn>
            </vnsAbsNode>

            <vnsAbsTermNodeCon name="Output1">
                <vnsAbsTermConn name="C6">
                </vnsAbsTermConn>
            </vnsAbsTermNodeCon>

            <vnsAbsConnection name="CON2" adjType="L3" unicastRoute="yes">
                <vnsRsAbsConnectionConns
                  tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeCon-Output1/AbsTConn"/>
                <vnsRsAbsConnectionConns
                  tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-outside"/>
            </vnsAbsConnection>

            <vnsAbsConnection name="CON1" adjType="L2" unicastRoute="no">
                <vnsRsAbsConnectionConns
                  tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-inside"/>
                <vnsRsAbsConnectionConns
                  tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeProv-Input1/AbsTConn"/>
            </vnsAbsConnection>

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

The following REST request creates a filter and a security policy (contract):

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vzFilter name="HttpIn">
            <vzEntry name="e1" prot="6" dToPort="80"/>
        </vzFilter>
```

```
        <vzBrCP name="webCtrct">
            <vzSubj name="http">
                <vzRsSubjFiltAtt tnVzFilterName="HttpIn"/>
            </vzSubj>
        </vzBrCP>
    </fvTenant>
</polUni>
```

The following REST request attaches a service graph to a contract:

```
<polUni>
    <fvTenant name="acme">
        <vzBrCP name="webCtrct">
            <vzSubj name="http">
                <vzRsSubjGraphAtt graphName="G1" termNodeName="Input1"/>
            </vzSubj>
        </vzBrCP>
    </fvTenant>
</polUni>
```