Cisco APIC Basic Configuration Guide, Release 2.x

First Published: 2016-06-29

Last Modified: 2018-08-08

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

CONTENTS

Preface

This preface includes the following sections:

Audience

This guide is intended primarily for data center administrators with responsibilities and expertise in one or more of the following:

- Virtual machine installation and administration
- Server administration
- Switch and network administration

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.

Convention	Description
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Warning IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Related Documentation

Application Policy Infrastructure Controller (APIC) Documentation

The following companion guides provide documentation for APIC:

- *Cisco APIC Getting Started Guide*
- *Cisco APIC Basic Configuration Guide*
- *Cisco ACI Fundamentals*
- *Cisco APIC Layer 2 Networking Configuration Guide*
- *Cisco APIC Layer 3 Networking Configuration Guide*
- *Cisco APIC NX-OS Style Command-Line Interface Configuration Guide*
- *Cisco APIC REST API Configuration Guide*
- *Cisco APIC Layer 4 to Layer 7 Services Deployment Guide*
- *Cisco ACI Virtualization Guide*
- *Cisco Application Centric Infrastructure Best Practices Guide*

All these documents are available 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

Cisco Application Centric Infrastructure (ACI) Documentation

The broader ACI documentation is available 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.

Cisco Application Centric Infrastructure (ACI) Simulator Documentation

The Cisco ACI Simulator documentation is available at http://www.cisco.com/c/en/us/support/cloud-systems-management/application-centric-infrastructure-simulator/tsd-products-support-series-home.html.

Cisco Nexus 9000 Series Switches Documentation

The Cisco Nexus 9000 Series Switches documentation is available at http://www.cisco.com/c/en/us/support/switches/nexus-9000-series-switches/tsd-products-support-series-home.html.

Cisco Application Virtual Switch Documentation

The Cisco Application Virtual Switch (AVS) documentation is available at http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/tsd-products-support-series-home.html.

Cisco Application Centric Infrastructure (ACI) Integration with OpenStack Documentation

Cisco ACI integration with OpenStack documentation is available at http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html.

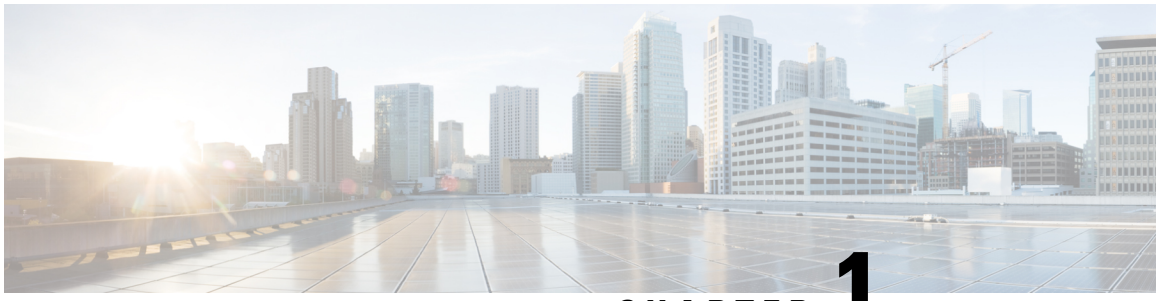
Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to apic-docfeedback@cisco.com. We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at: http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation as an RSS feed and delivers content directly to your desktop using a reader application. The RSS feeds are a free service.



CHAPTER 1

New and Changed Information

This chapter contains the following sections:

New and Changed Information

The following tables provide an overview of the significant changes to this guide up to this current 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 in Cisco APIC 2.3(1e) Release

Changed Feature	Description	Where Documented
Contract Inheritance	To streamline associating contracts to new EPGs, you can now enable an EPG to inherit all the (provided/consumed) contracts associated directly to another EPG in the same tenant. Contract inheritance can be configured for application, microsegmented, L2Out, and L3Out EPGs. Any changes you make to the EPG contract master's contracts, are received by the inheriting EPG.	*Basic User Tenant Configuration*
DHCP Relay Support for Consumer on Layer 3 Outside	You can make a Layer 3 Port a relay interface.	*Provisioning Core ACI Fabric Services*

Table 2: Changed Features in this Document

Changed Feature	Description	Where Documented
Chapter moved to new guide	*ACI Fabric Access Layer 2 Connectivity* chapter moved to new configuration guide	*Cisco APIC Layer 2 Configuration Guide*

Changed Feature	Description	Where Documented
Chapter moved to new guide	*ACI Fabric Access Layer 3 Outside Connectivity* chapter moved to new configuration guide	*Cisco APIC Layer 3 Configuration Guide*

Table 3: New Features and Changed Behavior in Cisco APIC 2.2(2e) Release

Feature or Change	Description	Where Documented
Name Change	Changed name of "Layer 3 EVPN Services for Fabric WAN" to "Cisco ACI GOLF	*Cisco ACI GOLF* in *ACI Fabric Layer 3 Outside Connectivity*

Table 4: New Features and Changed Behavior in Cisco APIC 2.2 (1n) Release

Feature	Description	Where Documented
FCoE over FEX	You can now configure FCoE over FEX ports.	Supporting Fibre Channel over Ethernet Traffic on the ACI Fabric

Table 5: New Features and Changed Behavior in Cisco APIC 2.1(1h) Release

Feature	Description	Where Documented
Distribute EVPN Type-2 Host Routes	In this release, for optimal traffic forwarding in an EVPN topology, you can enable fabric spines to advertise host routes using EVPN type-2 (MAC-IP) routes to the DCIG along with public BD subnets in the form of BGP EVPN type-5 (IP Prefix) routes.	*Distributing BGP EVPN Type-2 Host Routes* in *Configuring Layer 3 EVPN Services over Fabric WAN*
Configuring network-based microsegmented EPGs in a bare-metal environment	In this release you can configure microsegmented EPGs with IP address attributes or MAC address attributes for physical endpoint devices.	*Using Microsegmentation with Network-based Attributes on Bare-Metal*
Configuring IP address-based EPGs as shared resources	In this release you can configure a IP address-based microsegmented EPG as a resource that can be access and shared by devices on VRFs other than the one on which the EPG is native.	*Configuring IP Address-based Microsegmented EPGs as a shared resource*

Feature	Description	Where Documented
Global in-band/out-of-band default management connectivity toggle	In this release, you can toggle between In-band and out-of-band as the default management connectivity mode between the APIC server and external management devices.	*Configuring In-Band Management Access Using the Advanced GUI*, *Configuring In-Band Management Access Using the NX-OS Style CLI*, and *Configuring In-Band Management Access Using the REST API*

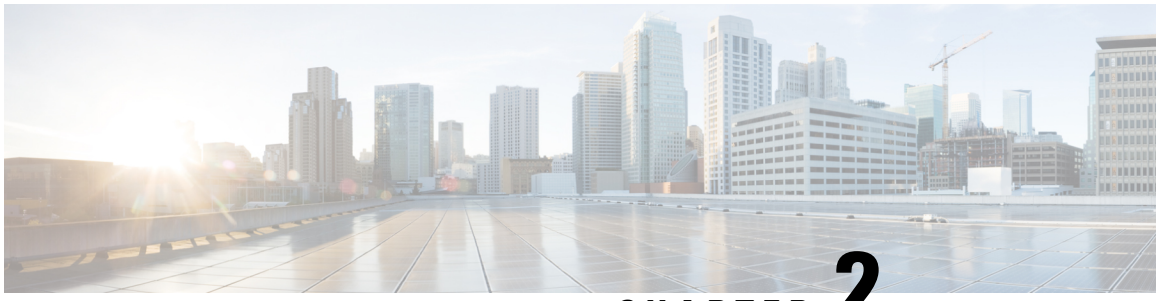
Table 6: New Features and Changed Behavior in Cisco APIC 2.0(2f) Release

Feature	Description	Where Documented
No significant changes occurred in the release.		

Table 7: New Features and Changed Behavior in Cisco APIC and Document Reorganization

Cisco APIC Release Version	Feature	Description	Where Documented
Release 2.0(1m)	Import control policy support -- for OSPF available for inbound filtering.	Notice that enabling import and export controls now applies to OSPF as well as BGP.	OSPF support for import and export controls is inserted in the following topics: • Configuring a Layer 3 Outside for Tenant Networks Using the GUI 200 • Configuring Layer 3 Outside for Tenant Networks Using the REST API 202 • Configuring a Route Control Protocol to Use Import and Export Controls, With the GUI 225 • Configuring a Route Control Protocol to Use Import and Export Controls, With the REST API 227 • Configuring a Route Control Protocol to Use Import and Export Controls, With the NX-OS Style CLI 228 • Transit Route Control 241

Cisco APIC Release Version	Feature	Description	Where Documented
Release 2.0(1m)	–	The contents of this guide was reorganized. Several GUI, REST API, and CLI tasks that were in the *Cisco APIC Getting Started Guide* in earlier releases are now migrated in this guide.	–
Release 2.0(1m)	Fibre Channel over Ethernet (FCoE) support	An overview and configuration topics for implementing FCoE connectivity over the ACI fabric.	FCoE concepts and APIC configuration are described in the **ACI Fabric Layer 2 Connectivity** chapter in the following topics. • FCoE Basic GUI Configuration 122 • FCoE Advanced GUI Configuration 129 • Configuring FCoE Connectivity Using the NX-OS Style CLI 147 • Configuring FCoE Connectivity Using the REST API 155
Release 2.0(1m)	Layer 3 EVPN Services Over Fabric WAN	An overview and configuration topics for implementing Layer 3 EVPN Services over the Fabric WAN	Layer 3 Services Over Fabric WAN concepts and APIC configuration are described in the **ACI Fabric Layer 3 Outside Connectivity** chapter in the following topics. • Layer 3 EVPN Services Over Fabric WAN 191 • Configuring Layer 3 EVPN for WAN Services Using the GUI 208 • Configuring Layer 3 EVPN for WAN Services Using the NX-OS Style CLI 210 • Configuring Layer 3 EVPN for WAN Services Using the REST API 211



CHAPTER 2

About Cisco ACI/APIC Configuration

Recommended Settings for the Cisco Application Policy Infrastructure Controller

We recommend the following settings for the Cisco Application Policy Infrastructure Controller (Cisco APIC):

Table 8: Recommended Settings for the Cisco APIC

Navigation Path	Property	Value	Description
System > **System Settings** > **Fabric Wide Setting**	Disable Remote EP Learning	Put a check in the box.	When this knob is enabled, IP remote learning on border leaf switches is disabled on VRF instances that are configured in ingress policy enforcement mode that contain external interfaces, except for remote IP learning that is generated as a result of IP multicast packets.
	Enforce Subnet Check	Put a check in the box.	This feature enforces subnet checks at the VRF instance level, when the Cisco Application Centric Infrastructure (Cisco ACI) learns the IP address as an endpoint from the data plane. Although the subnet check scope is the VRF instance, this feature can be enabled and disabled only globally under the fabric-wide setting policy. You cannot enable this option only in one VRF instance. If you put a check in the box for this option, the fabric will not learn IP addresses from a subnet other than the one configured on the bridge domain. This feature prevents the fabric from learning endpoint information in this scenario.

Navigation Path	Property	Value	Description
System > **System Settings** > **Endpoint Controls**	IP Aging Policy	Enabled	The IP aging policy tracks and ages unused IP addresses on an endpoint. Tracking is performed by using the endpoint retention policy, which is configured for the bridge domain to send ARP requests (for IPv4) and neighbor solicitations (for IPv6) at 75% of the local endpoint aging interval. When no response is received from an IP address, that IP address is aged out.
Fabric > **External Access Policies** > **Policies** > **Global** > **MCP Instance Policy default**	Admin State	Enabled	This enables the Mis-cabling Protocol (MCP)
	Controls: Enable MCP PDU per VLAN	Put a check in the box.	MCP detects other types of loops that can be caused by various issues, such as misconfiguration, that LLDP and STP cannot discover. This option enables MCP to send packets on a per-EPG basis.

About ACI/APIC Interfaces

The single point of management within the Cisco Application Centric Infrastructure (ACI) architecture is known as the Application Policy Infrastructure Controller (APIC). This controller provides access to all configuration, management, monitoring, and health functions. Having a centralized controller with an application programming interface (API) means that all functions configured or accessed through the fabric can be approached through the following interfaces:

- APIC GUI

  The APIC GUI is a browser-based graphical interface to the APIC that communicates internally with the APIC engine by exchanging REST API messages. It includes two modes:

  - Formerly called Advanced Mode, now simply the APIC GUI—Used for large scale configurations, deployments, and operations; enables granular policy controls such as in switch profiles, interface profiles, policy groups, or access entity profiles (AEPs) for automating mass fabric configuration and deployment.

- Formerly Basic Mode—Up to release 3.1(x), but now removed, this was a simple interface to enable common workflows, the GUI operational mode enables administrators to get started easily with ACI with a minimal knowledge of the object model. The simplified GUI allows the configuration of leaf ports and tenants without the need to configure advanced policies.

For more information about the APIC GUI, see *Cisco APIC Getting Started Guide, Release 3.x* and *Cisco APIC Basic Configuration Guide, Release 3.x*.

- NX-OS Style CLI—The NX-OS style Command-Line Interface (CLI) can be used for APIC configuration, deployment, and operation. It is organized in a hierarchy of command modes with EXEC mode as the root, containing a tree of configuration submodes beginning with global configuration mode. The commands available to you depend on the mode you are in.

  For important guidelines to use both the NX-OS style CLI and the APIC GUI to configure Cisco APIC, see Mixing the NX-OS Style CLI and the APIC GUI, on page 8.

  For more information about the NX-OS style CLI, see *Cisco APIC NX-OS Style Command-Line Interface Configuration Guide*.

- APIC REST API—The REST API is responsible for accepting configuration, as well as providing access to management functions for the controller. This interface is a crucial component for the GUI and CLI, and also provides a touch point for automation tools, provisioning scripts and third party monitoring and management tools.

  The APIC REST API is a programmatic interface that uses REST architecture. The API accepts and returns HTTP (not enabled by default) or HTTPS messages that contain JavaScript Object Notation (JSON) or Extensible Markup Language (XML) documents. You can use any programming language to generate the messages and the JSON or XML documents that contain the API methods or MO descriptions.

  For more information about the REST API, see the *Cisco APIC REST API Configuration Guide*.

Mixing the NX-OS Style CLI and the APIC GUI

Basic mode is deprecated since Cisco APIC Release 3.0(1). There is only one GUI as of that release.

Caution Configurations done through the NX-OS style CLI are rendered in the APIC GUI. They can be seen, but sometimes may not be editable in the GUI. Also changes made in the APIC GUI may be seen in the NX-OS style CLI, but may only partially work. See the following examples:

- Do not mix the GUI and the CLI, when doing per-interface configuration on APIC. Configurations performed in the GUI, may only partially work in the NX-OS CLI.

  For example, if you configure a switch port in the GUI at **Tenants** > ***tenant-name*** > **Application Profiles** > ***application-profile-name*** > **Application EPGs** > ***EPG-name*** > **Static Ports** > **Deploy Static EPG on PC, VPC, or Interface**

  Then you use the show running-config command in the NX-OS style CLI, you receive output such as:

  ```
  leaf 102
  interface ethernet 1/15
  switchport trunk allowed vlan 201 tenant t1 application ap1 epg ep1
   exit
   exit
  ```

  If you use these commands to configure a static port in the NX-OS style CLI, the following error occurs:

  ```
  apic1(config)# leaf 102
  apic1(config-leaf)# interface ethernet 1/15
  apic1(config-leaf-if)# switchport trunk allowed vlan 201 tenant t1 application ap1 epg
   ep1
  No vlan-domain associated to node 102 interface ethernet1/15 encap vlan-201
  ```

  This occurs because the CLI has validations that are not performed by the APIC GUI. For the commands from the show running-config command to function in the NX-OS CLI, a vlan-domain must have been previously configured. The order of configuration is not enforced in the GUI.

For the steps to remove such objects, see *Troubleshooting Unwanted _ui_ Objects* in the *APIC Troubleshooting Guide*.

About the Modes of Configuring Layer 3 External Connectivity

Because APIC supports multiple user interfaces (UIs) for configuration, the potential exists for unintended interactions when you create a configuration with one UI and later modify the configuration with another UI. This section describes considerations for configuring Layer 3 external connectivity with the APIC NX-OS style CLI, when you may also be using other APIC user interfaces.

When you configure Layer 3 external connectivity with the APIC NX-OS style CLI, you have the choice of two modes:

- Implicit mode, a simpler mode, is not compatible with the APIC GUI or the REST API.
- Named (or Explicit) mode is compatible with the APIC GUI and the REST API.

In either case, the configuration should be considered read-only in the incompatible UI.

How the Modes Differ

In both modes, the configuration settings are defined within an internal container object, the "L3 Outside" (or "L3Out"), which is an instance of the **l3extOut** class in the API. The main difference between the two modes is in the naming of this container object instance:

- Implicit mode—the naming of the container is implicit and does not appear in the CLI commands. The CLI creates and maintains these objects internally.
- Named mode—the naming is provided by the user. CLI commands in the Named Mode have an additional **l3Out** field. To configure the named L3Out correctly and avoid faults, the user is expected to understand the API object model for external Layer 3 configuration.

Note Except for the procedures in the *Configuring Layer 3 External Connectivity Using the Named Mode* section, this guide describes Implicit mode procedures.

Guidelines and Restrictions

- In the same APIC instance, both modes can be used together for configuring Layer 3 external connectivity with the following restriction: The Layer 3 external connectivity configuration for a given combination of tenant, VRF, and leaf can be done only through one mode.
- For a given tenant VRF, the policy domain where the External-l3 EPG can be placed can be in either the Named mode or in the Implicit mode. The recommended configuration method is to use only one mode for a given tenant VRF combination across all the nodes where the given tenant VRF is deployed for Layer 3 external connectivity. The modes can be different across different tenants or different VRFs and no restrictions apply.
-
- The external Layer 3 features are supported in both configuration modes, with the following exception:
  - Route-peering and Route Health Injection (RHI) with a L4-L7 Service Appliance is supported only in the Named mode. The Named mode should be used across all border leaf switches for the tenant VRF where route-peering is involved.
- Layer 3 external network objects (l3extOut) created using the Implicit mode CLI procedures are identified by names starting with "__ui_" and are marked as read-only in the GUI. The CLI partitions these external-l3 networks by function, such as interfaces, protocols, route-map, and EPG. Configuration modifications performed through the REST API can break this structure, preventing further modification through the CLI.

For the steps to remove such objects, see *Troubleshooting Unwanted _ui_ Objects* in the *APIC Troubleshooting Guide*.

Configuration Validation

When the administrator enters a configuration in the Cisco Application Policy Infrastructure Controller (Cisco APIC), the Cisco APIC performs checks to make sure that the configuration is valid, which is known as validation. If the configuration is accepted, but it conflicts with other previous configurations, Cisco APIC or the leaf switches might raise faults. The amount of checks performed by the Cisco APIC before accepting a configuration varies depending on the release. Newer releases have been enhanced to perform more checks before the configuration is accepted instead of only raising faults asynchronously.

The release with the greatest amount of changes in terms of additional validations is the Cisco APIC release 2.3. Cisco APIC release 3.0 further enhances validations at the VRF instance level. As an example, in Cisco

APIC release 2.3, for the same VRF instance and the same L3Out, you can define multiple Switch Virtual Interface (SVI) logical interface profiles for the same SVI (encap) with different IP addresses. You can define IP address 10.10.10.1/24 on path node1, port 1/41, VLAN (encap) 10, and IP address 10.10.10.2/24 for path node1, port 1/43, VLAN (encap) 10.

This results in only one IP address being used for SVI 10 on the leaf switch despite the fact that you configured multiple IP addresses, and depending on which IP address is used as the next hop for routing or whether you have IGP configured, the configuration might function properly.

Starting with Cisco APIC release 3.0, the above configuration would not be accepted, because even if in the Cisco Application Centric Infrastructure (Cisco ACI) object model the SVI is defined per path (logical interface profile), a given VRF instance on a given leaf switch can only have one IP address for an SVI and potentially a secondary IP address. Several other validations were also introduced in Cisco APIC release 3.0.

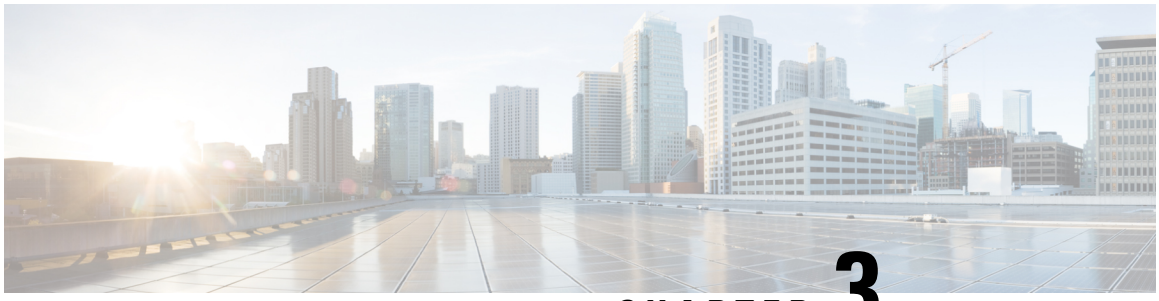
The objective of these validations is to reduce or eliminate configuration errors by informing the user of the errors at the configuration time instead of accepting the configuration and raising faults asynchronously.

As a result of these improvements, if you POST a configuration that was incorrect, but was considered valid prior to the 2.3 release, this POST would not result in the configuration being posted and the Cisco APIC will return an error message.

There might be existing Cisco APIC deployments that are functioning correctly with versions prior to Cisco APIC release 2.3 despite the fact that the configurations might not be valid. To reduce the impact of a firmware upgrade in such scenarios, after you upgrade to the 2.3 release or later, the Cisco APIC relaxes the validation checks on existing configurations.

Cisco APIC also offers the option to import an existing configuration with the "Best Effort" mode instead of the "Atomic" mode. This option offers the ability to accept a configuration even if there are portions that are not valid. The Cisco APIC pushes the valid portions of the configuration and ignores the portions that are not consistent with the validation. For the inconsistent portions, the Cisco APIC issues an error message that is visible when you use the following command:

```
show snapshot jobs import_job
```



CHAPTER 3

User Access, Authentication, and Accounting

This chapter contains the following sections:

Access Rights Workflow Dependencies

The Cisco Application Centric Infrastructure (ACI) RBAC rules enable or restrict access to some or all of the fabric. For example, in order to configure a leaf switch for bare metal server access, the logged in administrator must have rights to the `infra` domain. By default, a tenant administrator does not have rights to the `infra` domain. In this case, a tenant administrator who plans to use a bare metal server connected to a leaf switch could not complete all the necessary steps to do so. The tenant administrator would have to coordinate with a fabric administrator who has rights to the `infra` domain. The fabric administrator would set up the switch configuration policies that the tenant administrator would use to deploy an application policy that uses the bare metal server attached to an ACI leaf switch.

User Access, Authorization, and Accounting

Application Policy Infrastructure Controller (APIC) policies manage the authentication, authorization, and accounting (AAA) functions of the Cisco Application Centric Infrastructure (ACI) fabric. The combination of user privileges, roles, and domains with access rights inheritance enables administrators to configure AAA functions at the managed object level in a granular fashion. These configurations can be implemented using the REST API, the CLI, or the GUI.

Multiple Tenant Support

A core Application Policy Infrastructure Controller (APIC) internal data access control system provides multitenant isolation and prevents information privacy from being compromised across tenants. Read/write restrictions prevent any tenant from seeing any other tenant's configuration, statistics, faults, or event data. Unless the administrator assigns permissions to do so, tenants are restricted from reading fabric configuration, policies, statistics, faults, or events.

User Access: Roles, Privileges, and Security Domains

The APIC provides access according to a user's role through role-based access control (RBAC). An Cisco Application Centric Infrastructure (ACI) fabric user is associated with the following:

- A set of roles
- For each role, a privilege type: no access, read-only, or read-write
- One or more security domain tags that identify the portions of the management information tree (MIT) that a user can access

The ACI fabric manages access privileges at the managed object (MO) level. A privilege is an MO that enables or restricts access to a particular function within the system. For example, fabric-equipment is a privilege bit. This bit is set by the Application Policy Infrastructure Controller (APIC) on all objects that correspond to equipment in the physical fabric.

A role is a collection of privilege bits. For example, because an "admin" role is configured with privilege bits for "fabric-equipment" and "tenant-security," the "admin" role has access to all objects that correspond to equipment of the fabric and tenant security.

A security domain is a tag associated with a certain subtree in the ACI MIT object hierarchy. For example, the default tenant "common" has a domain tag `common`. Similarly, the special domain tag `all` includes the entire MIT object tree. An administrator can assign custom domain tags to the MIT object hierarchy. For example, an administrator could assign the "solar" domain tag to the tenant named solar. Within the MIT, only certain objects can be tagged as security domains. For example, a tenant can be tagged as a security domain but objects within a tenant cannot.

Note Security Domain password strength parameters can be configured by creating **Custom Conditions** or by selecting **Any Three Conditions** that are provided.

Creating a user and assigning a role to that user does not enable access rights. It is necessary to also assign the user to one or more security domains. By default, the ACI fabric includes two special pre-created domains:

- `All`—allows access to the entire MIT
- `Infra`— allows access to fabric infrastructure objects/subtrees, such as fabric access policies

Note For read operations to the managed objects that a user's credentials do not allow, a "DN/Class Not Found" error is returned, not "DN/Class Unauthorized to read." For write operations to a managed object that a user's credentials do not allow, an HTTP 401 Unauthorized error is returned. In the GUI, actions that a user's credentials do not allow, either they are not presented, or they are grayed out.

A set of predefined managed object classes can be associated with domains. These classes should not have overlapping containment. Examples of classes that support domain association are as follows:

- Layer 2 and Layer 3 network managed objects
- Network profiles (such as physical, Layer 2, Layer 3, management)
- QoS policies

When an object that can be associated with a domain is created, the user must assign domain(s) to the object within the limits of the user's access rights. Domain assignment can be modified at any time.

If a virtual machine management (VMM) domain is tagged as a security domain, the users contained in the security domain can access the correspondingly tagged VMM domain. For example, if a tenant named solar is tagged with the security domain called sun and a VMM domain is also tagged with the security domain called sun, then users in the solar tenant can access the VMM domain according to their access rights.

Configuring a Local User

In the initial configuration script, the admin account is configured and the admin is the only user when the system starts. The APIC supports a granular, role-based access control system where user accounts can be created with various roles including non-admin users with fewer privileges.

Configuring a Local User Using the GUI

Before you begin

- The ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.
- As appropriate, the security domain(s) that the user will access are defined. For example, if the new use account will be restricted to accessing a tenant, the tenant domain is tagged accordingly.
- An APIC user account is available that will enable the following:
  - Creating the TACACS+ and TACACS+ provider group.
  - Creating the local user account in the target security domain(s). If the target domain is `all`, the login account used to create the new local user must be a fabric-wide administrator that has access to `all`. If the target domain is a tenant, the login account used to create the new local user must be a tenant administrator that has full read write access rights to the target tenant domain.

Procedure

Step 1 On the menu bar, choose **ADMIN** > **AAA**.

Step 2 In the **Navigation** pane, click **Users** and **Local Users** in the **Work** pane.

Step 3 In the **Work** pane, verify that you in the **Local Users** tab.

The admin user is present by default

Step 4 In the **Work** pane, click on task icon drop-down list and select **Create Local User**.

Step 5 In the **User Identity** dialog box, enter a **Login ID** and **Password** for the user, and click **Next**.

Step 6 In the **Security** dialog box, choose the desired security domain for the user, and click **Next**.

Step 7 In the **Roles** dialog box, click the radio buttons to choose the roles for your user, and click **Next**.

You can provide read-only or read/write privileges.

Step 8 In the **User Identity** dialog box, perform the following actions:

a) In the **Login ID** field, add an ID.

b) In the **Password** field, enter the password.

At the time a user sets their password, the APIC validates it against the following criteria:

- Minimum password length is 8 characters.
- Maximum password length is 64 characters.
- Has fewer than three consecutive repeated characters.
- Must have characters from at least three of the following characters types: lowercase, uppercase, digit, symbol.
- Does not use easily guessed passwords.
- Cannot be the username or the reverse of the username.
- Cannot be any variation of cisco, isco or any permutation of these characters or variants obtained by changing the capitalization of letters therein.

c) In the **Confirm Password** field, confirm the password.

d) (Optional) For client-based authentication, in the **User Certificate Attribute** field, enter the user identity from the authentication certificate.

e) Click **Finish**.

Step 9 In the **Navigation** pane, click the name of the user that you created. In the **Work** pane, expand the + sign next to your user in the **Security Domains** area.
The access privileges for your user are displayed.

Configuring SSH Public Key Authentication Using the GUI

Before you begin

- Create a local user account in the target security domain(s). If the target domain is `all`, the login account used to create the new local user must be a fabric-wide administrator that has access to `all`. If the target domain is a tenant, the login account used to create the new local user must be a tenant administrator that has full read write access rights to the target tenant domain.
- Generate a public key using the Unix command **ssh-keygen**.

  The default login domain must be set to **local**

Procedure

Step 1 On the menu bar, choose **ADMIN** > **Security Management** > **Local Users**.

Step 2 In the **Navigation** pane, click the name of the user that you previously created.

Step 3 In the **Work** pane, expand the **SSH Keys** table, and insert the following information:

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

b) In the **Key** field, insert the public key previously created. Click **Update**.

Note To create the SSH Private Key File for downloading to a remote location then in the menu bar, expand **Firmware** > **Download Tasks**.

Configuring a Local User Using the NX-OS Style CLI

Procedure

Step 1 In the NX-OS CLI, start in configuration mode, shown as follows:

Example:

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

Step 2 Create a new user, shown as follows:

Example:

```
apic1(config)# username
 WORD       User name (Max Size 28)
 admin
 cli-user
 jigarshah
 test1
 testUser

apic1(config)# username test
```

```
apic1(config-username)#
 account-status       Set The status of the locally-authenticated user account.
 certificate          Create AAA user certificate in X.509 format.
 clear-pwd-history    Clears the password history of a locally-authenticated user
 domain               Create the AAA domain to which the user belongs.
 email                Set The email address of the locally-authenticated user.
 exit                 Exit from current mode
 expiration           If expires enabled, Set expiration date of locally-authenticated user
 account.
 expires              Enable expiry for locally-authenticated user account
 fabric               show fabric related information
 first-name           Set the first name of the locally-authenticated user.
 last-name            Set The last name of the locally-authenticated user.
 no                   Negate a command or set its defaults
 password             Set The system user password.
 phone                Set The phone number of the locally-authenticated user.
 pwd-lifetime         Set The lifetime of the locally-authenticated user password.
 pwd-strength-check   Enforces the strength of the user password
 show                 Show running system information
 ssh-key              Update ssh key for the user for ssh authentication
 where                show the current mode

apic1(config-username)# exit
```

Configuring a Local User Using the REST API

Procedure

Create a local user.

Example:

```
URL: https://apic-ip-address/api/policymgr/mo/uni/userext.xml
POST CONTENT:
        <aaaUser name="operations" phone="" pwd="<strong_password>" >
           <aaaUserDomain childAction="" descr="" name="all" rn="userdomain-all" status="">

                <aaaUserRole childAction="" descr="" name="Ops" privType="writePriv"/>
            </aaaUserDomain>
      </aaaUser>
```

Configuring a Remote User

Instead of configuring local users, you can point the APIC at the centralized enterprise credential datacenter. The APIC supports Lightweight Directory Access Protocol (LDAP), active directory, RADIUS, and TACACS+.

Note When an APIC is in minority (disconnected from the cluster), remote logins can fail because the ACI is a distributed system and the user information is distributed across APICS. Local logins, however, continue to work because they are local to the APIC.

Starting with the 3.1(1) release, **Server Monitoring** can be configured through RADIUS, TACACS+, LDAP, and RSA to determine whether the respective AAA servers are alive or not. Server monitoring feature uses the respective protocol login to check for server aliveness. For example, a LDAP server will use ldap login and a Radius server will use radius login with server monitoring to determine server aliveness.

To configure a remote user authenticated through an external authentication provider, you must meet the following prerequisites:

- The DNS configuration should have already been resolved with the hostname of the RADIUS server.
- You must configure the management subnet.

AV Pair on the External Authentication Server

The Cisco APIC requires that an administrator configure a Cisco AV Pair on an external authentication server. The Cisco AV pair specifies the APIC required RBAC roles and privileges for the user. The Cisco AV Pair format is the same for RADIUS, LDAP, or TACACS+.

To configure a Cisco AV Pair on an external authentication server, an administrator adds a Cisco AV pair to the existing user record. The Cisco AV pair format is as follows:

```
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,
domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,
domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2(16003)
```

Starting with Cisco APIC release 2.1, if no UNIX ID is provided in AV Pair, the APIC allocates the unique UNIX user ID internally.

Note The APIC Cisco AV-pair format is compatible and can co-exist with other Cisco AV-pair formats. APIC will pick up the first matching AV-pair from all the AV-pairs.

The APIC supports the following regexes:

```
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})$
```

Examples:

- Example 1: A Cisco AV Pair that contains a single Login domain with only writeRoles:

```
shell:domains=domainA/writeRole1|writeRole2/
```

- Example 2: A Cisco AV Pair that contains a single Login domain with only readRoles:

```
shell:domains=domainA//readRole1|readRole2
```

Note The "/" character is a separator between writeRoles and readRoles per Login domain and is required even if only one type of role is to be used.

The Cisco AVpair string is case sensitive. Although a fault may not be seen, using mismatching cases for the domain name or roles could lead to unexpected privileges being given.

An example configuration for an open RADIUS server (/etc/raddb/users) is as follows:

```
aaa-network-admin Cleartext-Password := "<password>"
Cisco-avpair = "shell:domains = all/aaa/read-all(16001)"
```

Best Practice for Assigning AV Pairs

As best practice,

Cisco recommends that you assign unique UNIX user ids in the range of 16000 to 23999 for the AV Pairs that are assigned to users when in bash shell (using SSH, Telnet or Serial/KVM consoles). If a situation arises when the Cisco AV Pair does not provide a UNIX user id, the user is assigned a user id of 23999 or similar number from the range that also enables the user's home directories, files, and processes accessible to remote users with a UNIX ID of 23999.

To ensure that your remote authentication server does NOT explicitly assign a UNIX ID in its cisco-av-pair response, open an SSH session to the APIC and login as an administrator (using a remote user account). Once logged in, run the following commands (replace “userid” with the username you logged in with):

```
admin@apic1:remoteuser-userid> cd /mit/uni/userext/remoteuser-userid
admin@apic1:remoteuser-userid> cat summary
```

The Cisco AVpair string is case sensitive. Although a fault may not be seen, using mismatching cases for the domain name or roles could lead to unexpected privileges being given.

Configuring an AV Pair on the External Authentication Server

The numerical value within the parentheses in the attribute/value (AV) pair string is used as the UNIX user ID of the user who is logged in using Secure Shell (SSH) or Telnet.

Procedure

Configure an AV pair on the external authentication server.

The Cisco AV pair definition is as follows (Cisco supports AV pairs with and without UNIX user IDs specified):

Example:

```
* shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2

        * shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2(8101)

These are the boost regexes supported by APIC:
uid_regex("shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?),\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$");
regex("shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?),\\S+?/\\S*?/\\S*?){0,31})$");
```

The following is an example:

```
shell:domains = coke/tenant-admin/read-all,pepsi//read-all(16001)
```

Configuring APIC for TACACS+ Access

Before you begin

- The Cisco Application Centric Infrastructure (ACI) fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The TACACS+ server host name or IP address, port, and key are available.
- The APIC management endpoint group is available.

Procedure

Step 1 In the APIC, create the **TACACS+ Provider**.

a) On the menu bar, choose **Admin** > **AAA**.
b) In the **Navigation** pane, choose **TACACS+ Managment** > **TACACS+ Providers**.
c) In the **Work** pane, choose **Actions** > **Create TACACS+ Provider**.
d) Specify the TACACS+ host name (or IP address), port, authorization protocol, key, and management endpoint group.

Note If the APIC is configured for in-band management connectivity, out-of-band management does not work for authentication. With the APIC release2.1(1x), you can set a global toggle between In-band and out-of-band as the default management connectivity between the APIC server and other external management devices.

For toggling in-band or out-of-band management in the APIC GUI:

- Prior to Release 2.2(1x): In the **Navigation** pane, choose **Fabric** > **Fabric Policies** > **Global Policies** > **Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.
- For Release 2.2(x) and 2.3(x): In the **Navigation** pane, choose **Fabric** > **Fabric Policies** > **Global Policies** > **APIC Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.
- For Release 3.0(1x) or later: In the **Navigation** pane, choose **System** > **System Settings** > **APIC Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.

Step 2 Create the **TACACS+ Provider Group**.

a) In the **Navigation** pane, choose **TACACS+ Managment** > **TACACS+ Provider Groups**.
b) In the **Work** pane, choose **Actions** > **Create TACACS+ Provider Group**.
c) Specify the TACACS+ provider group name, description, and providers as appropriate.

Step 3 Create the **Login Domain** for TACACS+.

a) In the **Navigation** pane, choose **AAA Authentication** > **Login Domains**.
b) In the **Work** pane, choose **Actions** > **Create Login Domain**.

c) Specify the login domain name, description, realm, and provider group as appropriate.

What to do next

This completes the APIC TACACS+ configuration steps. Next, if a RAIDUS server will also be used, configure the APIC for RADIUS. If only a TACACS+ server will be used, go to the ACS server configuration topic below.

Configuring APIC for RADIUS Access

Before you begin

- The ACI fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The RADIUS server host name or IP address, port, authorization protocol, and key are available.
- The APIC management endpoint group is available.

Procedure

Step 1 In the APIC, create the RADIUS provider.

a) On the menu bar, choose **Admin** > **AAA**.
b) In the **Navigation** pane, choose **RADIUS Managment** > **RADIUS Providers**.
c) In the **Work** pane, choose **Actions** > **Create RADIUS Provider**.
d) Specify the RADIUS host name (or IP address), port, protocol, and management endpoint group.

Note If the APIC is configured for in-band management connectivity, out-of-band management does not work for authentication. With the APIC release2.1(1x), you can set a global toggle between In-band and out-of-band as the default management connectivity between the APIC server and other external management devices.

For toggling in-band or out-of-band management in the APIC GUI:

- Prior to Release 2.2(1x): In the **Navigation** pane, choose **Fabric** > **Fabric Policies** > **Global Policies** > **Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.
- For Release 2.2(x) and 2.3(x): In the **Navigation** pane, choose **Fabric** > **Fabric Policies** > **Global Policies** > **APIC Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.
- For Release 3.0(1x) or later: In the **Navigation** pane, choose **System** > **System Settings** > **APIC Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.

Step 2 Create the RADIUS provider group.

a) In the **Navigation** pane, choose **RADIUS Managment** > **RADIUS Provider Groups**.
b) In the **Work** pane, choose **Actions** > **Create RADIUS Provider Group**.

c) Specify the RADIUS Provider Group name, description, and providers as appropriate.

Step 3 Create the login domain for RADIUS.

a) In the **Navigation** pane, choose **AAA Authentication** > **Login Domains**.
b) In the **Work** pane, choose **Actions** > **Create Login Domain**.
c) Specify the login domain name, description, realm, and provider group as appropriate.

What to do next

This completes the APIC RADIUS configuration steps. Next, configure the RADIUS server.

Configuring a Cisco Secure Access Control Server for RADIUS and TACACS+ Access to the APIC

Before you begin

- The Cisco Secure Access Control Server (ACS) version 5.5 is installed and online.

Note ACS v5.5 was used to document these steps. Other versions of ACS might support this task but the GUI procedures might vary accordingly.

- The Cisco Application Policy Infrastructure Controller (Cisco APIC) RADIUS or TACACS+ keys are available (or keys for both if both will be configured).
- The Cisco APICs are installed and online; the Cisco APIC cluster is formed and healthy.
- The RADIUS or TACACS+ port, authorization protocol, and key are available.

Procedure

Step 1 Log in to the ACS server to configure the Cisco APIC as a client.

a) Navigate to **Network Resources** > **Network Devices Groups** > **Network Devices and AAA Clients**.
b) Specify the client name, the Cisco APIC in-band IP address, select the TACACS+ or RADIUS (or both) authentication options.

Note If the only RADIUS or TACACS+ authentication is needed, select only the needed option.

c) Specify the authentication details such as Shared Secret (key), and port as appropriate for the authentication option(s).

Note The **Shared Secret**(s) must match the Cisco APIC **Provider** key(s).

Step 2 Create the Identity Group.

a) Navigate to **Users and Identity Stores** > **Internal Groups** option.
b) Specify the **Name**, and **Parent Group** as appropriate.

Step 3 Map users to the Identity Group.

a) In the **Navigation** pane, click the **Users and Identity Stores** > **Internal Identity Stores** > **Users** option.
b) Specify the user **Name**, and **Identity Group** as appropriate.

Step 4 Create the Policy Element.

a) Navigate to the **Policy Elements** option.
b) For RADIUS, specify the Authorization and Permissions > Network Access > Authorization Profiles **Name**. For TACACS+, specify the Authorization and Permissions > Device Administration > Shell Profile **Name** as appropriate.
c) For RADIUS, specify the **Attribute** as `cisco-av-pair`, **Type** as string, and the **Value** as **`shell:domains = <domain>/<role>/,<domain>// role`** as appropriate. For TACACS+, specify the **Attribute** as `cisco-av-pair`, **Requirement** as Mandatory, and the **Value** as **`shell:domains = <domain>/<role>/,<domain>// role`** as appropriate.

   For example, if the *cisco-av-pair* has a value of `shell:domains = solar/admin/,common// read-all(16001)`, then `solar` is the security domain, `admin` is the role for this user that gives write privileges to this user in the security domain called `solar`, `common` is the Cisco Application Centric Infrastructure (Cisco ACI) tenant common, and `read-all(16001)` is the role with read privileges that gives this user read privileges to all of the Cisco ACI tenant common.

Step 5 Create a service selection rule.

a) For RADIUS, create a service selection rule to associate the Identity Group with the Policy Element by navigating to **Access Policies** > **Default Device Network Access Identity** > **Authorization** and specifying the rule **Name**, **Status**, and **Conditions** as appropriate, and **Add** the `Internal Users:UserIdentityGroup in ALL Groups:<identity group name>`.
b) For TACACS+, create a service selection rule to associate the Identity Group with the Shell Profile by navigating to **Access Policies** > **Default Device Admin Identity** > **Authorization**. Specify the rule **Name**, **Conditions**, and **Select** the **Shell Profile** as appropriate.

What to do next

Use the newly created RADIUS and TACACS+ users to log in to the Cisco APIC. Verify that the users have access to the correct Cisco APIC security domain according to the assigned RBAC roles and privileges. The users should not have access to items that have not been explicitly permitted. Read and write access rights should match those configured for that user.

Configuring Windows Server 2008 LDAP for APIC Access with Cisco AVPair

Before you begin

- First, configure the LDAP server, then configure the Cisco Application Policy Infrastructure Controller (Cisco APIC) for LDAP access.
- The Microsoft Windows Server 2008 is installed and online.

- The Microsoft Windows Server 2008 Server Manager ADSI Edit tool is installed. To install ADSI Edit, follow the instructions in the Windows Server 2008 Server Manager help.
- `CiscoAVPair` attribute specifications: Common Name = **`CiscoAVPair`**, LDAP Display Name = **`CiscoAVPair`**, Unique X500 Object ID = 1.3.6.1.4.1.9.22.1, Description = **`CiscoAVPair`**, Syntax = **`Case Sensitive String`**.

Note For LDAP configurations, best practice is to use **`CiscoAVPair`** as the attribute string. If customer faces the issue using Object ID 1.3.6.1.4.1.9.22.1, an additional Object ID 1.3.6.1.4.1.9.2742.1-5 can also be used in the LDAP server.

- A Microsoft Windows Server 2008 user account is available that will enable the following:
  - Running ADSI Edit to add the `CiscoAVPair` attribute to the Active Directory (AD) Schema.
  - Configuring an Active Directory LDAP user to have `CiscoAVPair` attribute permissions.
- Port 636 is required for configuring LDAP integration with SSL/TLS.

Procedure

Step 1 Log in to an Active Directory (AD) server as a domain administrator.

Step 2 Add the `CiscoAVPair` attribute to the AD schema.

a) Navigate to **Start** > **Run**, type **`mmc`** and press **Enter**.
The Microsoft Management Console (MMC) opens.
b) Navigate to **File** > **Add/Remove Sanp-in** > **Add**.
c) In the **Add Standalonee Snap-in** dialog box, select the **Active Directory Schema** and click **Add**.
The MMC **Console** opens.
d) Right-click the **Attributes** folder, select the **Create Attribute** option.
The **Create New Attribute** dialog box opens.
e) Enter **`CiscoAVPair`** for the **Common Name** , **`CiscoAVPair`** for the **LDAP Display Name**, **`1.3.6.1.4.1.9.22.1`** for the **Unique X500 Object ID**, and select **`Case Sensitive String`** for the **Syntax**.
f) Click **OK** to save the attribute.

Step 3 Update the **User Properties** class to include the **CiscoAVPair** attribute.

a) In the MMC **Console**, expand the **Classes** folder, right-click the **user** class, and choose **Properties**.
The **user Properties** dialog box opens.
b) Click the **Attributes** tab, and click **Add** to open the **Select Schema Object** window.
c) In the **Select a schema object:** list, choose **`CiscoAVPair`**, and click **Apply**.
d) In the MMC **Console**, right-click the **Active Directory Schema**, and select **Reload the Schema**.

Step 4 Configure the **`CiscoAVPair`** attribute permissions.

Now that the LDAP includes the **`CiscoAVPair`** attributes, LDAP users need to be granted Cisco APIC permission by assigning them Cisco APIC RBAC roles.

a) In the ADSI Edit dialog box, locate a user who needs access to the Cisco APIC.
b) Right-click on the user name, and choose **Properties**.

The **<user> Properties** dialog box opens.

c) Click the **Attribute Editor** tab, select the *CiscoAVPair* attribute, and enter the *Value* as **`shell:domains = <domain>/<role>/,<domain>// role`**.

For example, if the *CiscoAVPair* has a value of `shell:domains = solar/admin/,common// read-all(16001)`, then `solar` is the security domain, `admin` is the role for this user that gives write privileges to this user in the security domain called `solar`, `common` is the Cisco Application Centric Infrastructure (Cisco ACI) tenant common, and `read-all(16001)` is the role with read privileges that gives this user read privileges to all of the Cisco ACI tenant common.

d) Click **OK** to save the changes and close the **<user> Properties** dialog box.

The LDAP server is configured to access the Cisco APIC.

What to do next

Configure the Cisco APIC for LDAP access.

Configuring APIC for LDAP Access

Before you begin

- The Cisco Application Centric Infrastructure (ACI) fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The LDAP server host name or IP address, port, bind DN, Base DN, and password are available.
- The APIC management endpoint group is available.

Procedure

Step 1 In the APIC, configure the LDAP Provider.

a) On the menu bar, choose **Admin** > **AAA**.
b) In the **Navigation** pane, choose **LDAP Managment** > **LDAP Providers**.
c) In the **Work** pane, choose **Actions** > **Create LDAP Provider**.
d) Specify the LDAP host name (or IP address), port, bind DN, base DN, password, attribute, and management endpoint group.

Note

- The bind DN is the string that the APIC uses to log in to the LDAP server. The APIC uses this account to validate the remote user attempting to log in. The base DN is the container name and path in the LDAP server where the APIC searches for the remote user account. This is where the password is validated. Filter is used to locate the attribute that the APIC requests to use for the *cisco-av-pair*. This contains the user authorization and assigned RBAC roles for use on the APIC. The APIC requests the attribute from the LDAP server.
- **Attribute** field—Enter one of the following:
  - For LDAP server configurations with a Cisco AVPair, enter **`CiscoAVPair`**.
  - For LDAP server configurations with an LDAP group map, enter **`memberOf`**.
- If the APIC is configured for in-band management connectivity, choosing an out-of-band management endpoint group for LDAP access does not take effect. Alternatively, an out-of-band over an in-band management endpoint group can connect a LDAP server, but requires configuring a static route for the LDAP server. The sample configuration procedures in this document use an APIC in-band management endpoint group.

Step 2 In the APIC, configure the LDAP Provider Group.

a) In the **Navigation** pane, choose **LDAP Managment** > **LDAP Provider Groups**.
b) Right-click **LDAP Provider Groups** and click **Create LDAP Provider Group**.
c) Specify the following information as appropriate.
   - **Name**—Enter a name for the LDAP provider group
   - **Description**—(Optional) Enter a description of the LDAP provider group
   - **Auth Choice**—Choose the authorization option, which can be **CiscoAVPair** (the default) or **LdapGroupMap**.

     Note The **LdapGroupMap** option requires creating an LDAP group map.

   - **LDAP Group Map**—(For the **LdapGroupMap** option) Click the drop-down arrow and choose the LDAP group map.
   - **Providers**—Click the + to enter an LDAP provider name, priority, and description.

Step 3 On the APIC, configure the login domain for LDAP.

a) In the **Navigation** pane, choose **AAA Authentication** > **Login Domains**.
b) In the **Work** pane, choose **Actions** > **Create Login Domain**.
c) Specify the login domain name, description, realm, and provider group as appropriate.

What to do next

This completes the APIC LDAP configuration steps. Next, test the APIC LDAP login access.

Changing the Default Behavior for Remote Users with Missing or Bad Cisco AV Pairs

Procedure

Step 1 On the menu bar, click **ADMIN** > **AAA**.

Step 2 In the **Navigation** pane, click **AAA Authentication**.

Step 3 In the **Work** pane, in the **Properties** area, from the **Remote user login policy** drop-down list, choose **Assign Default Role**.

The default value is **No Login**. The **Assign Default Role** option assigns the minimal read-only privileges to users that have missing or bad Cisco AV Pairs. Bad AV Pairs are those AV Pairs that fail the parsing rules.

Changing Default Behavior for Remote Users with Missing or Bad Cisco AV Pairs Using the NX-OS Style CLI

The Cisco APIC requires that an administrator configure a Cisco AV Pair on an external authentication server. To do so, an administrator adds a Cisco AV pair to the existing user record. The Cisco AV pair specifies the APIC required RBAC roles and privileges for the user. The Cisco AV Pair format is the same for RADIUS, LDAP, or TACACS+. One AV pair format contains a Cisco UNIX user ID and one does not. Both are correct if all remote users have the same role and mutual file access is acceptable. If the UNIX user ID is not specified, ID 23999 is applied by the APIC system, and more than one role/read privilege is specified to any AV Pair user. This can cause users to have higher or lower permissions than configured through the group settings. This topic explains how to change the bahavior if that is not acceptable.

To change the default behavior for remote users with missing or bad Cisco AV pairs using the NX-OS CLI:

Procedure

Step 1 In the NX-OS CLI, start in Configuration mode.

Example:

```
apic1#
apic1# configure
```

Step 2 Configure the aaa user default role.

Example:

```
apic1(config)# aaa user default-role
 assign-default-role  assign-default-role
 no-login             no-login
```

Step 3 Configure the aaa authentication login methods.

Example:

```
apic1(config)# aaa authentication
 login  Configure methods for login

apic1(config)# aaa authentication login
 console  Configure console methods
 default  Configure default methods
 domain   Configure domain methods

apic1(config)# aaa authentication login console
 <CR>

apic1(config)# aaa authentication login domain
 WORD      Login domain name
 fallback
```

About Signature-Based Transactions

The APIC controllers in a Cisco ACI fabric offer different methods to authenticate users.

The primary authentication method uses a username and password and the APIC REST API returns an authentication token that can be used for future access to the APIC. This may be considered insecure in a situation where HTTPS is not available or enabled.

Another form of authentication that is offered utilizes a signature that is calculated for every transaction. The calculation of that signature uses a private key that must be kept secret in a secure location. When the APIC receives a request with a signature rather than a token, the APIC utilizes an X.509 certificate to verify the signature. In signature-based authentication, every transaction to the APIC must have a newly calculated signature. This is not a task that a user should do manually for each transaction. Ideally this function should be utilized by a script or an application that communicates with the APIC. This method is the most secure as it requires an attacker to crack the RSA/DSA key to forge or impersonate the user credentials.

Note Additionally, you must use HTTPS to prevent replay attacks.

Before you can use X.509 certificate-based signatures for authentication, verify that the following pre-requisite tasks are completed:

1. Create an X.509 certificate and private key using OpenSSL or a similar tool.
2. Create a local user on the APIC. (If a local user is already available, this task is optional).
3. Add the X.509 certificate to the local user on the APIC.

Guidelines and Limitations

Follow these guidelines and limitations:

- Local users are supported. Remote AAA users are not supported.

- The APIC GUI does not support the certificate authentication method.
- WebSockets and eventchannels do not work for X.509 requests.
- Certificates signed by a third party are not supported. Use a self-signed certificate.

Generating an X.509 Certificate and a Private Key

Procedure

Step 1 Enter an OpenSSL command to generate an X.509 certificate and private key.

Example:

```
$ openssl req -new -newkey rsa:1024 -days 36500 -nodes -x509 -keyout userabc.key -out
userabc.crt -subj '/CN=User ABC/O=Cisco Systems/C=US'
```

Note
- Once the X.509 certificate is generated, it will be added to the users profile on the APIC, and it is used to verify signatures. The private key is used by the client to generate the signatures.
- The certificate contains a public key but not the private key. The public key is the primary information used by the APIC to verify the calculated signature. The private key is never stored on the APIC. You must keep it secret.

Step 2 Display the fields in the certificate using OpenSSL.

Example:

```
$ openssl x509 -text -in userabc.crt
        Certificate:
            Data:
                Version: 3 (0x2)
                Serial Number:
                    c4:27:6c:4d:69:7c:d2:b6
                Signature Algorithm: sha1WithRSAEncryption
                Issuer: CN=User ABC, O=Cisco Systems, C=US
                Validity
                    Not Before: Jan 12 16:36:14 2015 GMT
                    Not After : Dec 19 16:36:14 2114 GMT
                Subject: CN=User ABC, O=Cisco Systems, C=US
                Subject Public Key Info:
                    Public Key Algorithm: rsaEncryption
                    RSA Public Key: (1024 bit)
                        Modulus (1024 bit):
                            00:92:35:12:cd:2b:78:ef:9d:ca:0e:11:77:77:3a:
                            99:d3:25:42:94:b5:3e:8a:32:55:ce:e9:21:2a:ff:
                            e0:e4:22:58:6d:40:98:b1:0d:42:21:db:cd:44:26:
                            50:77:e5:fa:b6:10:57:d1:ec:95:e9:86:d7:3c:99:
                            ce:c4:7f:61:1d:3c:9e:ae:d8:88:be:80:a0:4a:90:
                            d2:22:e9:1b:25:27:cd:7d:f3:a5:8f:cf:16:a8:e1:
                            3a:3f:68:0b:9c:7c:cb:70:b9:c7:3f:e8:db:85:d8:
                            98:f6:e3:70:4e:47:e2:59:03:49:01:83:8e:50:4a:
                            5f:bc:35:d2:b1:07:be:ec:e1
                        Exponent: 65537 (0x10001)
                X509v3 extensions:
                    X509v3 Subject Key Identifier:
                        0B:E4:11:C7:23:46:10:4F:D1:10:4C:C1:58:C2:1E:18:E8:6D:85:34
                    X509v3 Authority Key Identifier:
```

```
                    keyid:0B:E4:11:C7:23:46:10:4F:D1:10:4C:C1:58:C2:1E:18:E8:6D:85:34
                    DirName:/CN=User ABC/O=Cisco Systems/C=US
                    serial:C4:27:6C:4D:69:7C:D2:B6

                X509v3 Basic Constraints:
                    CA:TRUE
        Signature Algorithm: sha1WithRSAEncryption
            8f:c4:9f:84:06:30:59:0c:d2:8a:09:96:a2:69:3d:cf:ef:79:
            91:ea:cd:ae:80:16:df:16:31:3b:69:89:f7:5a:24:1f:fd:9f:
            d1:d9:b2:02:41:01:b9:e9:8d:da:a8:4c:1e:e5:9b:3e:1d:65:
            84:ff:e8:ad:55:3e:90:a0:a2:fb:3e:3e:ef:c2:11:3d:1b:e6:
            f4:5e:d2:92:e8:24:61:43:59:ec:ea:d2:bb:c9:9a:7a:04:91:
            8e:91:bb:9d:33:d4:28:b5:13:ce:dc:fe:c3:e5:33:97:5d:37:
            cc:5f:ad:af:5a:aa:f4:a3:a8:50:66:7d:f4:fb:78:72:9d:56:
            91:2c
[snip]
```

Configuring a Local User

Creating a Local User and Adding a User Certificate Using the GUI

Procedure

Step 1 On the menu bar, choose **ADMIN** > **AAA**.

Step 2 In the **Navigation** pane, click **Users** and **Local Users** in the **Work** pane.

Step 3 In the **Work** pane, verify that you in the **Local Users** tab.

The admin user is present by default

Step 4 In the **Work** pane, click on task icon drop-down list and select **Create Local User**.

Step 5 In the **Security** dialog box, choose the desired security domain for the user, and click **Next**.

Step 6 In the **Roles** dialog box, click the radio buttons to choose the roles for your user, and click **Next**.

You can provide read-only or read/write privileges.

Step 7 In the **User Identity** dialog box, perform the following actions:

a) In the **Login ID** field, add an ID.
b) In the **Password** field, enter the password.
c) In the **Confirm Password** field, confirm the password.
d) (Optional) For client-based authentication, in the **User Certificate Attribute** field, enter the user identity from the authentication certificate.
e) Click **Finish**.

Step 8 In the **Navigation** pane, click the name of the user that you created. In the **Work** pane, expand the + sign next to your user in the **Security Domains** area.
The access privileges for your user are displayed.

Step 9 In the **Work** pane, in the **User Certificates** area, click the user certificates + sign, and in the **Create X509 Certificate** dialog box, perform the following actions:

a) In the **Name** field, enter a certificate name.

b) In the **Data** field, enter the user certificate details.
c) Click **Submit**.

The X509 certificate is created for the local user.

Creating a Local User and Adding a User Certificate Using the REST API

Procedure

Create a local user and add a user certificate.

Example:

```
method: POST
        url: http://apic/api/node/mo/uni/userext/user-userabc.json
        payload:
        {
            "aaaUser": {
                "attributes": {
                    "name": "userabc",
                    "firstName": "Adam",
                    "lastName": "BC",
                    "phone": "408-525-4766",
                    "email": "userabc@cisco.com",
                },
                "children": [{
                    "aaaUserCert": {
                        "attributes": {
                            "name": "userabc.crt",
                          "data": "-----BEGIN CERTIFICATE-----\nMIICjjCCAfegAwIBAgIJAMQnbE
 <snipped content> ==\n-----END CERTIFICATE-----",
                        },
                        "children": []
                    },
                    "aaaUserDomain": {
                        "attributes": {
                            "name": "all",
                        },
                        "children": [{
                            "aaaUserRole": {
                                "attributes": {
                                    "name": "aaa",
                                    "privType": "writePriv",
                                },
                                "children": []
                            }
                        }, {
                            "aaaUserRole": {
                                "attributes": {
                                    "name": "access-admin",
                                    "privType": "writePriv",
                                },
                                "children": []
                            }
                        }, {
                            "aaaUserRole": {
                                "attributes": {
                                    "name": "admin",
                                    "privType": "writePriv",
```

```
                        },
                        "children": []
                    }
                }, {
                    "aaaUserRole": {
                        "attributes": {
                            "name": "fabric-admin",
                            "privType": "writePriv",
                        },
                        "children": []
                    }
                }, {
                    "aaaUserRole": {
                        "attributes": {
                            "name": "nw-svc-admin",
                            "privType": "writePriv",
                        },
                        "children": []
                    }
                }, {
                    "aaaUserRole": {
                        "attributes": {
                            "name": "ops",
                            "privType": "writePriv",
                        },
                        "children": []
                    }
                }, {
                    "aaaUserRole": {
                        "attributes": {
                            "name": "read-all",
                            "privType": "writePriv",
                        },
                        "children": []
                    }
                }, {
                    "aaaUserRole": {
                        "attributes": {
                            "name": "tenant-admin",
                            "privType": "writePriv",
                        },
                        "children": []
                    }
                }, {
                    "aaaUserRole": {
                        "attributes": {
                            "name": "tenant-ext-admin",
                            "privType": "writePriv",
                        },
                        "children": []
                    }
                }, {
                    "aaaUserRole": {
                        "attributes": {
                            "name": "vmm-admin",
                            "privType": "writePriv",
                        },
                        "children": []
                    }
                }]
            }
        }]
}
```

```
        }
```

Creating a Local User Using Python SDK

Procedure

Create a local user.

Example:

```
#!/usr/bin/env python
from cobra.model.pol import Uni as PolUni
from cobra.model.aaa import UserEp as AaaUserEp
from cobra.model.aaa import User as AaaUser
from cobra.model.aaa import UserCert as AaaUserCert
from cobra.model.aaa import UserDomain as AaaUserDomain
from cobra.model.aaa import UserRole as AaaUserRole
from cobra.mit.access import MoDirectory
from cobra.mit.session import LoginSession
from cobra.internal.codec.jsoncodec import toJSONStr

APIC = 'http://10.10.10.1'
username = 'admin'
password = 'p@$$w0rd'

session = LoginSession(APIC, username, password)
modir = MoDirectory(session)
modir.login()

def readFile(fileName=None, mode="r"):
   if fileName is None:
      return ""
   fileData = ""
   with open(fileName, mode) as aFile:
      fileData = aFile.read()
   return fileData

# Use a dictionary to define the domain and a list of tuples to define
# our aaaUserRoles (roleName, privType)
# This can further be abstracted by doing a query to get the valid
# roles, that is what the GUI does

userRoles = {'all': [
               ('aaa', 'writePriv'),
               ('access-admin', 'writePriv'),
               ('admin', 'writePriv'),
               ('fabric-admin', 'writePriv'),
               ('nw-svc-admin', 'writePriv'),
               ('ops', 'writePriv'),
               ('read-all', 'writePriv'),
               ('tenant-admin', 'writePriv'),
               ('tenant-ext-admin', 'writePriv'),
               ('vmm-admin', 'writePriv'),
           ],
}

uni = PolUni('') # '' is the Dn string for topRoot
```

```
aaaUserEp = AaaUserEp(uni)
aaaUser = AaaUser(aaaUserEp, 'userabc', firstName='Adam',
                  email='userabc@cisco.com')

aaaUser.lastName = 'BC'
aaaUser.phone = '555-111-2222'
aaaUserCert = AaaUserCert(aaaUser, 'userabc.crt')
aaaUserCert.data = readFile("/tmp/userabc.crt")
# Now add each aaaUserRole to the aaaUserDomains which are added to the
# aaaUserCert
for domain,roles in userRoles.items():
   aaaUserDomain = AaaUserDomain(aaaUser, domain)
   for roleName, privType in roles:
       aaaUserRole = AaaUserRole(aaaUserDomain, roleName,
                                 privType=privType)
print toJSONStr(aaaUser, prettyPrint=True)

cr = ConfigRequest()
cr.addMo(aaaUser)
modir.commit(cr)
# End of Script to create a user
```

Using a Private Key to Calculate a Signature

Before you begin

You must have the following information available:

- HTTP method - GET, POST, DELETE
- REST API URI being requested, including any query options
- For POST requests, the actual payload being sent to the APIC
- The private key used to generate the X.509 certificate for the user
- The distinguished name for the user X.509 certificate on the APIC

Procedure

Step 1 Concatenate the HTTP method, REST API URI, and payload together in this order and save them to a file.

This concatenated data must be saved to a file for OpenSSL to calculate the signature. In this example, we use a filename of payload.txt. Remember that the private key is in a file called userabc.key.

Example:

GET example:

```
GET http://10.10.10.1/api/class/fvTenant.json?rsp-subtree=children
```

POST example:

```
POST http://10.10.10.1/api/mo/tn-test.json{"fvTenant": {"attributes": {"status": "deleted",
 "name": "test"}}}
```

Step 2 Verify that the payload.txt file contains the correct information.

For example, using the GET example shown in the previous step:

```
GET http://10.10.10.1/api/class/fvTenant.json?rsp-subtree=children
```

Your payload.txt file should contain only the following information:

```
GET/api/class/fvTenant.json?rsp-subtree=children
```

Step 3 Verify that you didn't inadvertently create a new line when you created the payload file.

Example:

```
# cat -e payload.txt
```

Determine if there is a **$** symbol at the end of the output, similar to the following:

```
GET/api/class/fvTenant.json?rsp=subtree=children$
```

If so, then that means that a new line was created when you created the payload file. To prevent creating a new line when generating the payload file, use a command similar to the following:

```
echo -n "GET/api/class/fvTenant.json?rsp-subtree=children" >payload.txt
```

Step 4 Calculate a signature using the private key and the payload file using OpenSSL.

Example:

```
openssl dgst -sha256 -sign userabc.key payload.txt > payload_sig.bin
```

The resulting file has the signature printed on multiple lines.

Step 5 Convert the signature to base64 format:

Example:

```
openssl base64 -A -in payload_sig.bin -out payload_sig.base64
```

Step 6 Strip the signature of the new lines using Bash.

Example:

```
$ tr -d '\n' < payload_sig.base64
P+OTqK0CeAZjl7+Gute2R1Ww8OGgtzE0wsLlx8fIXXl4V79Zl7
Ou8IdJH9CB4W6CEvdICXqkv3KaQszCIC0+Bn07o3qF//BsIplZmYChD6gCX3f7q
IcjGX+R6HAqGeK7k97cNhXlWEoobFPe/oajtPjOu3tdOjhf/9ujG6Jv6Ro=
```

Note This is the signature that will be sent to the APIC for this specific request. Other requests will require to have their own signatures calculated.

Step 7 Place the signature inside a string to enable the APIC to verify the signature against the payload.

This complete signature is sent to the APIC as a cookie in the header of the request.

Example:

```
APIC-Request-Signature=P+OTqK0CeAZjl7+Gute2R1Ww8OGgtzE0wsLlx8f
IXXl4V79Zl7Ou8IdJH9CB4W6CEvdICXqkv3KaQszCIC0+Bn07o3qF//BsIplZmYChD6gCX3f
7qIcjGX+R6HAqGeK7k97cNhXlWEoobFPe/oajtPjOu3tdOjhf/9ujG6Jv6Ro=;
APIC-Certificate-Algorithm=v1.0; APIC-Certificate-Fingerprint=fingerprint;
APIC-Certificate-DN=uni/userext/user-userabc/usercert-userabc.crt
```

Note The DN used here must match the DN of the user certified object containing the x509 certificate in the next step.

Step 8 Use the CertSession class in the Python SDK to communicate with an APIC using signatures.

The following script is an example of how to use the CertSession class in the ACI Python SDK to make requests to an APIC using signatures.

Example:

```
#!/usr/bin/env python
# It is assumed the user has the X.509 certificate already added to
# their local user configuration on the APIC
from cobra.mit.session import CertSession
from cobra.mit.access import MoDirectory

def readFile(fileName=None, mode="r"):
    if fileName is None:
        return ""
    fileData = ""
    with open(fileName, mode) as aFile:
        fileData = aFile.read()
    return fileData

pkey = readFile("/tmp/userabc.key")
csession = CertSession("https://ApicIPOrHostname/",
                       "uni/userext/user-userabc/usercert-userabc", pkey)

modir = MoDirectory(csession)
resp = modir.lookupByDn('uni/fabric')
pring resp.dn
# End of script
```

Note The DN used in the earlier step must match the DN of the user certified object containing the x509 certificate in this step.

Accounting

ACI fabric accounting is handled by these two managed objects (MO) that are processed by the same mechanism as faults and events:

- The `aaaSessionLR` MO tracks user account login and logout sessions on the APIC and switches, and token refresh. The ACI fabric session alert feature stores information such as the following:
  - Username
  - IP address initiating the session
  - Type (telnet, https, REST etc.)
  - Session time and length
  - Token refresh – a user account login event generates a valid active token which is required in order for the user account to exercise its rights in the ACI fabric.

Note Token expiration is independent of login; a user could log out but the token expires according to the duration of the timer value it contains.

- The `aaaModLR` MO tracks the changes users make to objects and when the changes occurred.
- If the AAA server is not pingable, it is marked unavailable and a fault is seen.

Both the `aaaSessionLR` and `aaaModLR` event logs are stored in APIC shards. Once the data exceeds the pre-set storage allocation size, it overwrites records on a first-in first-out basis.

Note In the event of a destructive event such as a disk crash or a fire that destroys an APIC cluster node, the event logs are lost; event logs are not replicated across the cluster.

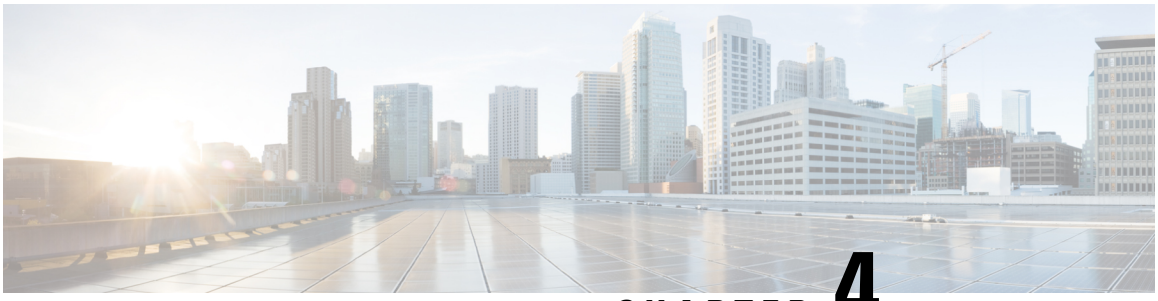
The `aaaModLR` and `aaaSessionLR` MOs can be queried by class or by distinguished name (DN). A class query provides all the log records for the whole fabric. All `aaaModLR` records for the whole fabric are available from the GUI at the **Fabric** > **Inventory** > **POD** > **History** > **Audit Log** section, The APIC GUI **History** > **Audit Log** options enable viewing event logs for a specific object identified in the GUI.

The standard syslog, callhome, REST query, and CLI export mechanisms are fully supported for `aaaModLR` and `aaaSessionLR` MO query data. There is no default policy to export this data.

There are no pre-configured queries in the APIC that report on aggregations of data across a set of objects or for the entire system. A fabric administrator can configure export policies that periodically export `aaaModLR` and `aaaSessionLR` query data to a syslog server. Exported data can be archived periodically and used to generate custom reports from portions of the system or across the entire set of system logs.

Routed Connectivity to External Networks as a Shared Service Billing and Statistics

The APIC can be configured to collect byte count and packet count billing statistics from a port configured for routed connectivity to external networks (an `l3extInstP` EPG) as a shared service. Any EPG in any tenant can share an `l3extInstP` EPG for routed connectivity to external networks. Billing statistics can be collected for each EPG in any tenant that uses an `l3extInstP` EPG as a shared service. The leaf switch where the `l3extInstP` is provisioned forwards the billing statistics to the APIC where they are aggregated. Accounting policies can be configured to periodically export these billing statics to a server.



CHAPTER 4

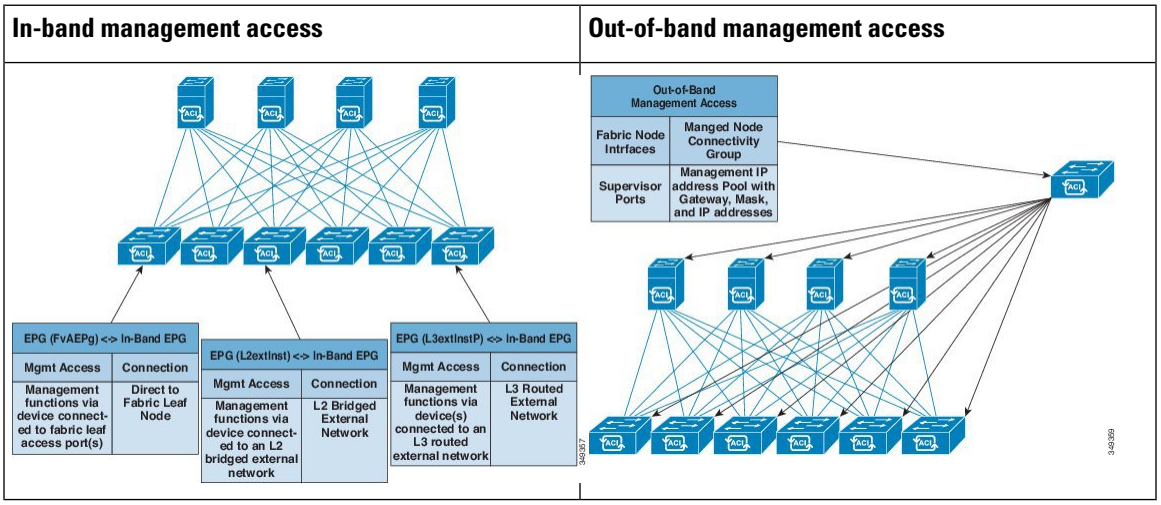
Management

This chapter contains the following sections:

Management Workflows

ACI Management Access Workflows

This workflow provides an overview of the steps required to configure management connectivity to switches in the ACI fabric.



1. Prerequisites

- Ensure that you have read/write access privileges to the infra security domain.
- Ensure that the target leaf switches with the necessary interfaces are available.

2. Configure the ACI Leaf Switch Access Ports

Choose which of these management access scenarios you will use:

- For **in-band** management, follow the suggested topics for in-band configuration in the *ACI Configuration Guide*.
- For **out-of-band** management, follow the suggested topics for out-of-band configuration in the *ACI Configuration Guide*.

Suggested topics

For additional information, see the following topics in the ACI Basic Configuration Guide:

- Configuring In-Band Management Access Using the Advanced GUI
- Configuring In-Band Management Access Using the NX-OS Style CLI
- Configuring In-Band Management Access Using the REST API
- Configuring Out-of-Band Management Access Using the Advanced GUI
- Configuring Out-of-Band Management Access Using the NX-OS Style CLI
- Configuring Out-of-Band Management Access Using the REST API

Adding Management Access

An APIC controller has two routes to reach the management network, one is by using the in-band management interface and the other is by using the out-of-band management interface.

- In-band management access—You can configure in-band management connectivity to the APIC and the ACI fabric. You first configure the VLANs that will be used by APIC when the APIC is communicating with the leaf switches, and then you configure the VLANs that the VMM servers will use to communicate with the leaf switches.
- Out-of-band management access—You can configure out-of-band management connectivity to the APIC and the ACI fabric. You configure an out-of-band contract that is associated with an out-of-band endpoint group (EPG), and attach the contract to the external network profile.

Note The APIC out-of-band management connection link must be 1 Gbps.

The APIC controller always selects the in-band management interface over the out-of-band management interface, if the in-band management interface is configured. The out-of-band management interface is used only when the in-band management interface is not configured, or if the destination address is on the same subnet as the out-of-band management subnet of the APIC.

The APIC management interface does not support an IPv6 address and cannot connect to an external IPv6 server through this interface.

Configuring the external management instance profile under the management tenant for in-band or out-of-band has no effect on the protocols that are configured under the fabric-wide communication policies. The subnets and contracts specified under the external management instance profile do not affect HTTP/HTTPS or SSH/Telnet.

Adding Management Access in the GUI

An APIC controller has two routes to reach the management network, one is by using the in-band management interface and the other is by using the out-of-band management interface.

The in-band management network allows APIC to communicate with the leaf switches and with the outside using the ACI fabric, and it makes it possible for external management devices to communicate with the APIC or the leaf switches and spine switches using the fabric itself.

The out-of-band management network configuration defines the configuration of the management port on the controllers, the leaf switches and the spine switches.

The APIC controller always selects the in-band management interface over the out-of-band management interface, if the in-band management interface is configured. The out-of-band management interface is used only when the in-band management interface is not configured or if the destination address is on the same subnet as the out-of-band management subnet of the APIC. This behavior cannot be changed or reconfigured. The APIC management interface does not support an IPv6 address and cannot connect to an external IPv6 server through this interface.

The APIC out-of-band management connection link must be 1 Gbps.

IPv4/IPv6 Addresses and In-Band Policies

In-band management addresses can be provisioned on the APIC controller only through a policy (Postman REST API, NX-OS Style CLI, or GUI). Additionally, the in-band management addresses must be configured statically on each node.

IPv4/IPv6 Addresses in Out-of-Band Policies

Out-of-band management addresses can be provisioned on the APIC controller either at the time of bootstrap or by using a policy (Postman REST API, NX-OS Style CLI, GUI). Additionally, the out-of-band management addresses must be configured statically on each node or by specifying a range of addresses (IPv4/IPv6) to the entire cluster. IP addresses are randomly assigned from a range to the nodes in the cluster.

IPv6 Table Modifications to Mirror the Existing IP Tables Functionality

All IPv6 tables mirror the existing IP tables functionality, except for Network Address Translation (NAT).

Existing IP Tables

1. Earlier, every rule in the IPv6 tables were executed one at a time and a system call was made for every rule addition or deletion.
2. Whenever a new policy was added, rules were appended to the existing IP tables file and no extra modifications were done to the file.
3. When a new source port was configured in the out-of-band policy, it added source and destination rules with the same port number.

Modifications to IP Tables

1. When IP tables are created, they are first written into hash maps that are then written into intermediate file IP tables-new which are restored. When saved, a new IP tables file is created in the /etc/sysconfig/ folder. You can find both these files at the same location. Instead of making a system call for every rule, you must make a system call only while restoring and saving the file.
2. When a new policy is added instead of appending it to the file, an IP table is created from scratch, that is by loading default policies into the hashmaps, checking for new policies, and adding them to hashmaps. Later, they are written to the intermediate file (/etc/sysconfig/iptables-new) and saved.
3. It is not possible to configure source ports alone for a rule in out-of-band policy. Either destination port or source port along with a destination port can be added to the rules.
4. When a new policy is added, a new rule will be added to the IP tables file. This rule changes the access flow of IP tables default rules.

   ```
   -A INPUT -s <OOB Address Ipv4/Ipv6> -j apic-default
   ```

5. When a new rule is added, it presents in the IP tables-new file and not in the IP tables file, and it signifies that there is some error in the IP tables-new file. Only if the restoration is successful, the file is saved and new rules are seen in the IP tables file.

Note

- If only IPv4 is enabled, do not configure an IPv6 policy.
- If only IPv6 is enabled, do not configure an IPv4 policy.
- If both IPv4 and IPv6 are enabled and a policy is added, it will be configured to both the versions . So when you add an IPv4 subnet, it will be added to IP tables and similarly an IPv6 subnet is added to IPv6 tables.

Configuring In-Band and Out-of-Band Management Access with Wizards

In APIC, release 3.1(x), wizards were added to simplify configuring management access. You can still use the other methods of configuring management access included in this document.

Procedure

Step 1 To configure **In-Band Management Access**, perform the following steps:

a) On the menu bar, click **Tenants** > **mgmt**.
b) Expand **Quick Start**.
c) Click **In-Band Management Access** > **Configure In-Band Management Access** > **Start**.
d) Follow the instructions to add the **Nodes** in the management network, the **IP addresses** for the nodes, communication filters for the **Connected Devices**, and communication filters for **Remote Attached Devices**.

Step 2 To configure **Out-of-Band Management Access**, perform the following steps:

a) On the menu bar, click **Tenants** > **mgmt**.
b) Expand **Quick Start**.
c) Click **Out-of-Band Management Access** > **Configure Out-of-Band Management Access** > **Start**.
d) Follow the instructions to add the **Nodes** in the out-of-band management network, the **IP addresses** for the nodes, subnets allowed for the **External Hosts**, and communication filters that will determine communication for **Access**.

Configuring In-Band Management Access Using the Cisco APIC GUI

Note IPv4 and IPv6 addresses are supported for in-band management access. IPv6 configurations are supported using static configurations (for both in-band and out-of-band). IPv4 and IPv6 dual in-band and out-of-band configurations are supported only through static configuration. For more information, see the KB article,*Configuring Static Management Access in Cisco APIC*.

Procedure

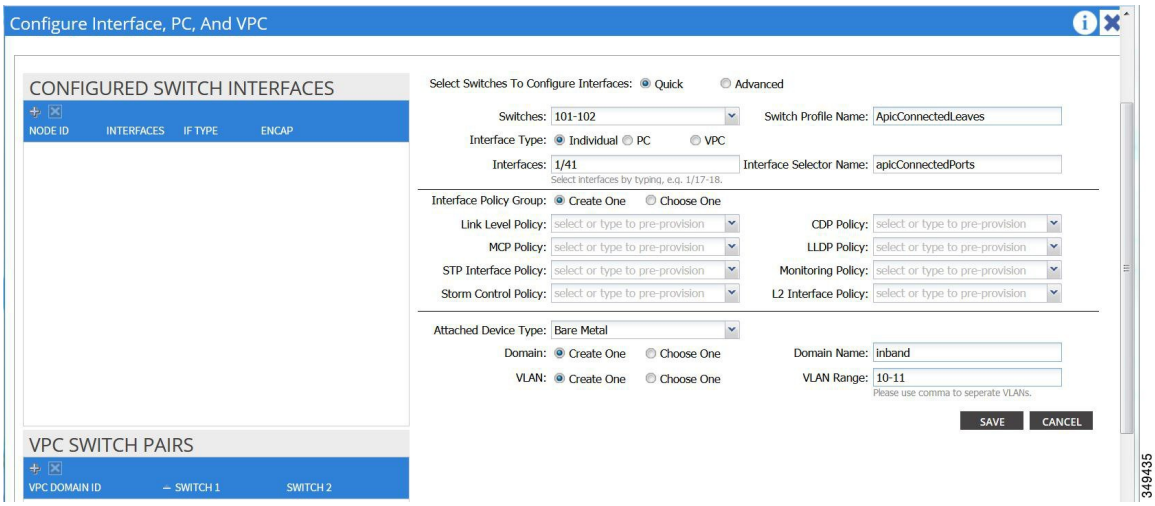
Step 1 On the menu bar, choose **FABRIC** > **External Access Policies**.

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

Step 3 In the **Configure Interface, PC, and VPC** dialog box, to configure switch ports connected to 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. (leaf1 and leaf2).
c) In the **Switch Profile Name** field, enter a name for the profile (apicConnectedLeaves).
d) Click the + icon to configure the ports.

A dialog box similar to the following image is displayed for the user to enter the content:



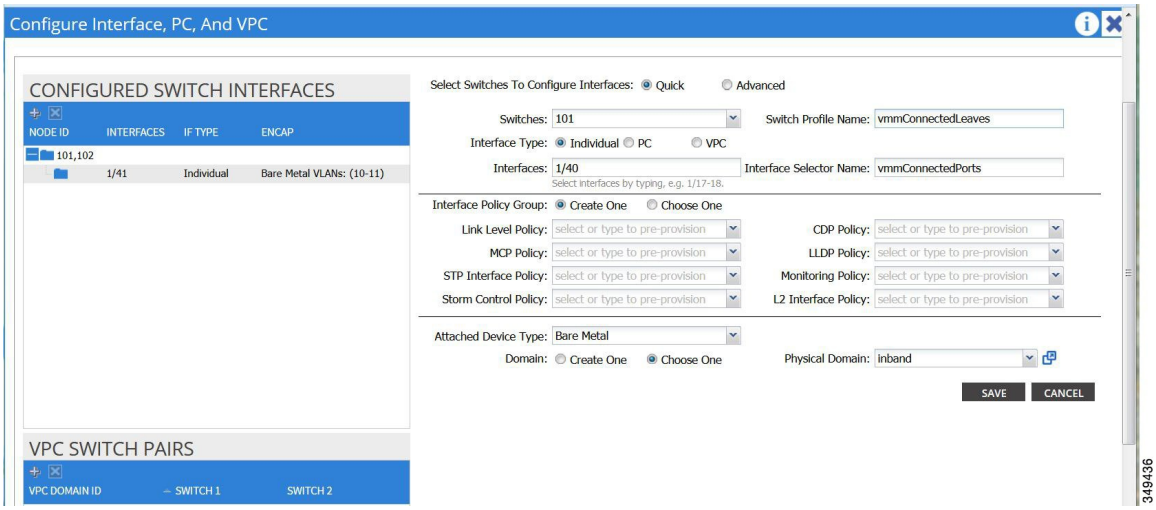
e) Verify that in the **Interface Type** area, the **Individual** radio button is selected.
f) In the **Interfaces** field, enter the ports to which APICs are connected.
g) In the **Interface Selector Name** field, enter the name of the port profile (apicConnectedPorts).
h) In the **Interface Policy Group** field, click the **Create One** radio button.
i) In the **Attached Device Type** field, choose the appropriate device type to configure the domain (Bare Metal).
j) In the **Domain** field, click the **Create One** radio button.
k) In the **Domain Name** field, enter the domain name. (**inband**)
l) In the **VLAN** field, choose the **Create One** radio button.
m) In the **VLAN Range** field, enter the VLAN range. Click **Save**, and click **Save** again. Click **Submit**.

Step 4 In the **Navigation** pane, right-click **Switch Policies** and choose **Configure Interface, PC and VPC**.

Step 5 In the **Configure Interface, PC, and VPC** dialog box, perform the following actions:

a) Click the large + icon next to the switch diagram to create a new profile and configure VLANs for the server.
b) In the **Switches** field, from drop-down list, check the check boxes for the switches to which the servers are connected. (leaf1).
c) In the **Switch Profile Name** field, enter a name for the profile (vmmConnectedLeaves).
d) Click the + icon to configure the ports.

A dialog box similar to the following image is displayed for the user to enter the content:



e) Verify that in the **Interface Type** area, the **Individual** radio button is selected.
f) In the **Interfaces** field, enter the ports to which the servers are connected (1/40).
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** field, choose the appropriate device type to configure the domain (Bare Metal).
j) In the **Domain** field, from the drop-down list click the **Choose One** radio button
k) From the **Physical Domain** drop-down list, choose the domain created earlier.
l) In the **Domain Name** field, enter the domain name.
m) Click **Save**, and click **Save** again.

Step 6 In the **Configure Interface, PC, and VPC** dialog box, click **Submit**.

Step 7 On the menu bar, click **TENANTS** > **mgmt**. In the **Navigation** pane, expand **Tenant mgmt** > **Networking** > **Bridge Domains** to configure the bridge domain on the in-band connection.

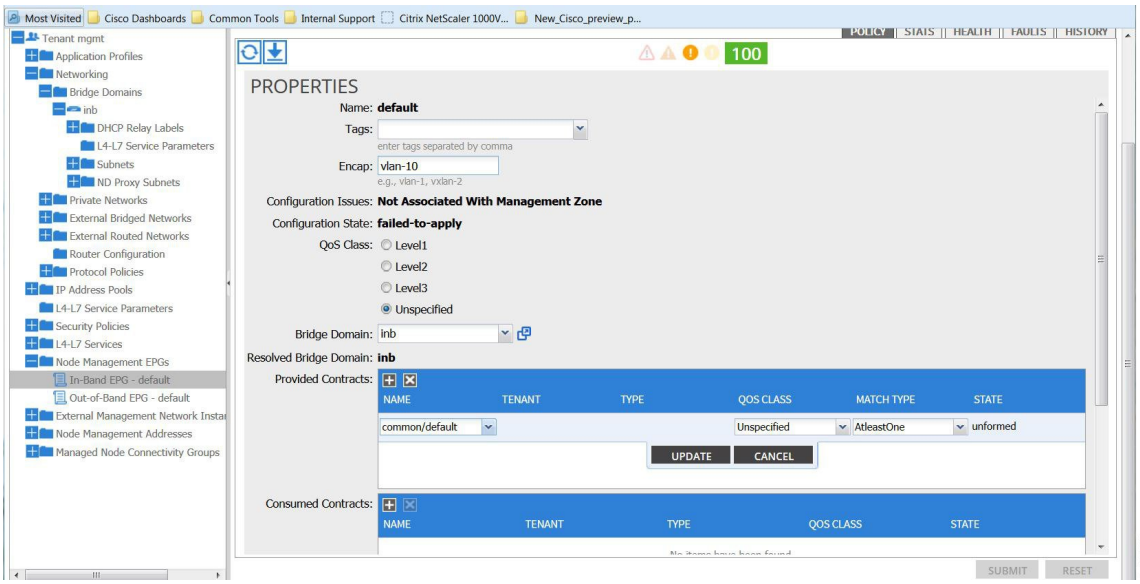
Step 8 Expand the in-band bridge domain (inb). Right-click **Subnets**. Click **Create Subnet** and perform the following actions to configure the in-band gateway:

a) In the **Create Subnet** dialog box, in the **Gateway IP** field, enter the in-band management gateway IP address and mask.
b) Click **Submit**.

Step 9 In the **Navigation** pane, expand **Tenant mgmt** > **Node Management EPGs**. Right-click **Node Management EPGs** and choose **Create In-Band Management EPG**. Perform the following actions to set the VLAN on the in-band EPG used to communicate with the APIC:

a) In the Name field, enter the in-band management EPG name.
b) In the **Encap** field, enter the VLAN (vlan-10).
c) From the **Bridge Domain** drop-down field, choose the bridge domain. Click **Submit**.
d) In the **Navigation** pane, choose the newly created in-band EPG.
e) Expand **Provided Contracts**. In the **Name** field, from the drop-down list, choose the default contract to enable EPG to provide the default contract that will be consumed by the EPGs on which the VMM servers are located.
f) Click **Update**, and click **Submit**.

A dialog box similar to the following image is displayed:



Step 10 In the **Navigation** pane, right-click **Node Management Addresses** and click **Create Node Management Addresses**, and perform the following actions to configure the IP addresses to be assigned to APIC controllers in the fabric:

a) In the **Create Node Management Addresses** dialog box, in the **Policy Name** field, enter the policy name (apicInb).
b) In the **Nodes** field, **Select** column, check the check boxes for the nodes that will be part of this fabric (apic1, apic2, apic3).
c) In the **Config** field, check the **In-Band Addresses** check box.
d) In the **Node Range** fields, enter the range.
e) In the **In-Band IP Addresses** area, in the **In-Band Management EPG** field, from the drop-down list, choose default. This associates the default in-band Management EPG.
f) In the **In-Band IP Addresses** and **Gateway** fields, enter the IPv4 or IPv6 addresses as desired.
g) Click **Submit**. The IP addresses for the APICs are now configured.

Step 11 In the **Navigation** pane, right-click **Node Management Addresses**. Click **Create Node Management Addresses**, and perform the following actions to configure the IP addresses for the leaf and spine switches in the fabric:

a) In the **Create Node Management Addresses** dialog box, in the **Policy Name** field, enter the policy name (switchInb).
b) In the **Nodes** field, **Select** column, check the check boxes next to the nodes that will be part of this fabric (leaf1, leaf2, spine1, spine2).
c) In the **Config** field, click the **In-Band Addresses** checkbox.
d) In the **Node Range** fields, enter the range.
e) In the **In-Band IP Addresses** area, in the **In-Band Management EPG** field, from the drop-down list, choose default. The default in-band management EPG is now associated.
f) In the **In-Band IP Addresses** and **Gateway** fields, enter the IPv4 or IPv6 addresses as desired.
g) Click **Submit**. In the **Confirm** dialog box, click **Yes**. The IP addresses for the leaf and spine switches are now configured.

Step 12 In the **Navigation** pane, under **Node Management Addresses**, click the APIC policy name (apicInb) to verify the configurations. In the **Work** pane, the IP addresses assigned to various nodes are displayed.

Step 13 In the **Navigation** pane, under **Node Management Addresses**, click the switches policy name (switchInb). In the **Work** pane, the IP addresses that are assigned to switches and the gateway addresses they are using are displayed.

Note You can make out-of-band management access the default management connectivity mode for the APIC server by clicking **System** > **System Settings** > **APIC Connectivity Preferences**. Then on the **Connectivity Preferences** page, click **inband**.

Configuring In-Band Management Access Using the NX-OS Style CLI

Procedure

Step 1 Assign a VLAN for the APIC inband management, as shown in the following example:

Example:

```
apic1(config)#
apic1(config)# vlan-domain inband-mgmt
apic1(config-vlan) vlan 10
apic1(config-vlan) exit
```

Step 2 Provide external connectivity to the inband management ports, as shown in the following example:

Example:

Note In this step, the controller is connected to a port on a leaf switch. You must add a VLAN domain member on that port. In this example, in leaf 101, the port ethernet 1/2 is connected to controller 1. You are configuring the VLAN domain member "inband management". This is one part of the connection. The other part is that the management station is connected to leaf 102, interface ethernet 1/3. A controller is one machine connected one port on the leaf switch, which in this case is leaf 102. The machine is trying to connect to the controller from the outside (ethernet 1/3).

```
apic1(config)#
apic1(config)# leaf 101
apic1(config-leaf) internet ethernet 1/2
apic1(config-leaf-if)# vlan-domain member inband-mgmt
apic1(config-leaf-if)# exit
apic1(config)# leaf 102
apic1(config-leaf) internet ethernet 1/3
apic1(config-leaf-if)# vlan-domain member inband-mgmt
apic1(config-leaf-if)# switchport trunk allowed vlan
apic1(config-leaf-if)# exit
```

Note You can make in-band management access the default management connectivity mode for the APIC server using the following CLI command sequence:

```
apic1# configure
apic1(config)# mgmt_connectivity pref inband
```

Configuring In-Band Management Access Using the REST API

IPv4 and IPv6 addresses are supported for in-band management access. IPv6 configurations are supported using static configurations (for both in-band and out-of-band). IPv4 and IPv6 dual in-band and out-of-band configurations are supported only through static configuration. For more information, see the KB article,*Configuring Static Management Access in Cisco APIC*.

Procedure

Step 1 Create a VLAN namespace.

Example:

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/uni.xml -->
<polUni>
  <infraInfra>
    <!-- Static VLAN range -->
    <fvnsVlanInstP name="inband" allocMode="static">
      <fvnsEncapBlk name="encap" from="vlan-10" to="vlan-11"/>
    </fvnsVlanInstP>
  </infraInfra>
</polUni>
```

Step 2 Create a physical domain.

Example:

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/uni.xml -->
<polUni>
  <physDomP name="inband">
    <infraRsVlanNs tDn="uni/infra/vlanns-inband-static"/>
  </physDomP>
</polUni>
```

Step 3 Create selectors for the in-band management.

Example:

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <infraInfra>
    <infraNodeP name="vmmNodes">
      <infraLeafS name="leafS" type="range">
        <infraNodeBlk name="single0" from_="101" to_="101"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-vmmPorts"/>
    </infraNodeP>

    <!-- Assumption is that VMM host is reachable via eth1/40. -->
    <infraAccPortP name="vmmPorts">
```

```
      <infraHPortS name="portS" type="range">
        <infraPortBlk name="block1"
                      fromCard="1" toCard="1"
                      fromPort="40" toPort="40"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-inband" />
      </infraHPortS>
    </infraAccPortP>


    <infraNodeP name="apicConnectedNodes">
      <infraLeafS name="leafS" type="range">
        <infraNodeBlk name="single0" from_="101" to_="102"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-apicConnectedPorts"/>
    </infraNodeP>

    <!-- Assumption is that APIC is connected to eth1/1. -->
    <infraAccPortP name="apicConnectedPorts">
      <infraHPortS name="portS" type="range">
        <infraPortBlk name="block1"
                      fromCard="1" toCard="1"
                      fromPort="1" toPort="3"/>
        <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-inband" />
      </infraHPortS>
    </infraAccPortP>

    <infraFuncP>
      <infraAccPortGrp name="inband">
        <infraRsAttEntP tDn="uni/infra/attentp-inband"/>
      </infraAccPortGrp>
    </infraFuncP>

    <infraAttEntityP name="inband">
      <infraRsDomP tDn="uni/phys-inband"/>
    </infraAttEntityP>
  </infraInfra>
</polUni>
```

Step 4 Configure an in-band bridge domain and endpoint group (EPG).

Example:

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="mgmt">
    <!-- Configure the in-band management gateway address on the
         in-band BD. -->
    <fvBD name="inb">
      <fvSubnet ip="10.13.1.254/24"/>
    </fvBD>

    <mgmtMgmtP name="default">
      <!-- Configure the encap on which APICs will communicate on the
           in-band network. -->
      <mgmtInB name="default" encap="vlan-10">
        <fvRsProv tnVzBrCPName="default"/>
      </mgmtInB>
    </mgmtMgmtP>
  </fvTenant>
</polUni>
```

Step 5 Create an address pool.

Example:

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="mgmt">
    <!-- Adresses for APIC in-band management network -->
    <fvnsAddrInst name="apicInb" addr="10.13.1.254/24">
      <fvnsUcastAddrBlk from="10.13.1.1" to="10.13.1.10"/>
    </fvnsAddrInst>

    <!-- Adresses for switch in-band management network -->
    <fvnsAddrInst name="switchInb" addr="10.13.1.254/24">
      <fvnsUcastAddrBlk from="10.13.1.101" to="10.13.1.120"/>
    </fvnsAddrInst>
  </fvTenant>
</polUni>
```

Note Dynamic address pools for IPv6 is not supported.

Step 6 Create management groups.

Example:

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <infraInfra>
    <!-- Management node group for APICs -->
    <mgmtNodeGrp name="apic">
      <infraNodeBlk name="all" from_="1" to_="3"/>
      <mgmtRsGrp tDn="uni/infra/funcprof/grp-apic"/>
    </mgmtNodeGrp>

    <!-- Management node group for switches-->
    <mgmtNodeGrp name="switch">
      <infraNodeBlk name="all" from_="101" to_="104"/>
      <mgmtRsGrp tDn="uni/infra/funcprof/grp-switch"/>
    </mgmtNodeGrp>

    <!-- Functional profile -->
    <infraFuncP>
      <!-- Management group for APICs -->
      <mgmtGrp name="apic">
        <!-- In-band management zone -->
        <mgmtInBZone name="default">
          <mgmtRsInbEpg tDn="uni/tn-mgmt/mgmtp-default/inb-default"/>
          <mgmtRsAddrInst tDn="uni/tn-mgmt/addrinst-apicInb"/>
        </mgmtInBZone>
      </mgmtGrp>

      <!-- Management group for switches -->
      <mgmtGrp name="switch">
        <!-- In-band management zone -->
        <mgmtInBZone name="default">
          <mgmtRsInbEpg tDn="uni/tn-mgmt/mgmtp-default/inb-default"/>
          <mgmtRsAddrInst tDn="uni/tn-mgmt/addrinst-switchInb"/>
        </mgmtInBZone>
      </mgmtGrp>
```

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

Note Dynamic address pools for IPv6 is not supported.

Configuring Out-of-Band Management Access Using the Cisco APIC GUI

Note IPv4 and IPv6 addresses are supported for out-of-band management access.

You must configure out-of-band management access addresses for the leaf and spine switches as well as for APIC

Before you begin

The APIC out-of-band management connection link must be 1 Gbps.

Procedure

Step 1 On the menu bar, choose **Tenants** > **mgmt**. In the **Navigation** pane, expand **Tenant mgmt**.

Step 2 Right-click **Node Management Addresses**, and click **Create Node Management Addresses**.

Step 3 In the **Create Node Management Addresses** dialog box, perform the following actions:

a) In the **Policy Name** field, enter a policy name (switchOob).

b) In the **Nodes** field, check the check boxes next to the appropriate leaf and spine switches (leaf1, leaf2, spine1).

c) In the **Config** field, check the check box for **Out of-Band Addresses**.

Note The **Out-of-Band IP addresses** area is displayed.

d) In the **Out-of-Band Management EPG** field, choose the EPG from the drop-down list (default).

e) In the **Out-Of-Band Gateway** field, enter the IP address and network mask for the external out-of-band management network.

f) In the **Out-of-Band IP Addresses** field, enter the range of desired IPv4 or IPv6 addresses that will be assigned to the switches. Click **Submit**.

The node management IP addresses are configured.

Step 4 In the **Navigation** pane, expand **Node Management Addresses**, and click the policy that you created. In the **Work** pane, the out-of-band management addresses are displayed against the switches.

Step 5 In the **Navigation** pane, expand **Contracts** > **Out-of-Band Contracts**.

Step 6 Right-click **Out-of-Band Contracts**, and click **Create Out-of-Band Contract**.

Step 7 In the **Create Out-of-Band Contract** dialog box, perform the following tasks:

a) In the **Name** field, enter a name for the contract (oob-default).

b) Expand **Subjects**. In the **Create Contract Subject** dialog box, in the **Name** field, enter a subject name (oob-default).

c) Expand **Filter Chain**, and in the **Name** field, from the drop-down list, choose the name of the filter (default). Click **Update**, and click **OK**.
d) In the **Create Out-of-Band Contract** dialog box, click **Submit**.

An out-of-band contract that can be applied to the out-of-band EPG is created.

Step 8 In the **Navigation** pane, expand **Node Management EPGs** > **Out-of-Band EPG - default**.

Step 9 In the **Work** pane, expand **Provided Out-of-Band Contracts**.

Step 10 In the **OOB Contract** column, from the drop-down list, choose the out-of-band contract that you created (oob-default). Click **Update**, and click **Submit**.
The contract is associated with the node management EPG.

Step 11 In the **Navigation** pane, right-click **External Network Instance Profile**, and click **Create External Management Entity Instance**.

Step 12 In the **Create External Management Entity Instance** dialog box, perform the following actions:

a) In the **Name** field, enter a name (oob-mgmt-ext).
b) Expand the **Consumed Out-of-Band Contracts** field. From the **Out-of-Band Contract** drop-down list, choose the contract that you created (oob-default). Click **Update**.

   Choose the same contract that was provided by the out-of-band management.

c) In the **Subnets** field, enter the subnet address. Click **Submit**.

   Only the subnet addresses you choose here will be used to manage the switches. The subnet addresses that are not included cannot be used to manage the switches.

The node management EPG is attached to the external network instance profile. The out-of-band management connectivity is configured.

Note You can make out-of-band management access the default management connectivity mode for the APIC server by clicking **System** > **System Settings** > **APIC Connectivity Preferences**. Then on the **Connectivity Preferences** page, click **ooband**.

Configuring Out-of-Band Management Access Using the NX-OS Style CLI

Before you begin

The APIC out-of-band management connection link must be 1 Gbps.

Procedure

Provide access control for out-of-band management interface to external management subnets as follows:

Example:

```
apic1(config-tenant)# external-l3 epg default oob-mgmt
apic1(config-tenant-l3ext-epg)#match ip 10.0.0.0/8
apic1(config-tenant-l3ext-epg)# exit
apic1(config)# exit
```

Note You can make out-of-band management access the default management connectivity mode for the APIC server using the following CLI command sequence:

```
apic1 # configure
apic1(config)# mgmt_connectivity pref ooband
```

Configuring Out-of-Band Management Access Using the REST API

IPv4 and IPv6 addresses are supported for out-of-band management access.

Before you begin

The APIC out-of-band management connection link must be 1 Gbps.

Procedure

Step 1 Create an out-of-band contract.

Example:

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

<polUni>
    <fvTenant name="mgmt">
        <!-- Contract -->
        <vzOOBBrCP name="oob-default">
            <vzSubj name="oob-default">
                <vzRsSubjFiltAtt tnVzFilterName="default" />
            </vzSubj>
        </vzOOBBrCP>
    </fvTenant>
</polUni>
```

Step 2 Associate the out-of-band contract with an out-of-band EPG.

Example:

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

<polUni>
    <fvTenant name="mgmt">
        <mgmtMgmtP name="default">
            <mgmtOoB name="default">
                <mgmtRsOoBProv tnVzOOBBrCPName="oob-default" />
            </mgmtOoB>
        </mgmtMgmtP>
    </fvTenant>
</polUni>
```

Step 3 Associate the out-of-band contract with an external management EPG.

Example:

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

<polUni>
```

```
    <fvTenant name="mgmt">
        <mgmtExtMgmtEntity name="default">
            <mgmtInstP name="oob-mgmt-ext">
                <mgmtRsOoBCons tnVzOOBBrCPName="oob-default" />
                <!-- SUBNET from where switches are managed -->
                <mgmtSubnet ip="10.0.0.0/8" />
            </mgmtInstP>
        </mgmtExtMgmtEntity>
    </fvTenant>
</polUni>
```

Step 4 Create a management address pool.

Example:

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

<polUni>
    <fvTenant name="mgmt">
        <fvnsAddrInst name="switchOoboobaddr" addr="172.23.48.1/21">
            <fvnsUcastAddrBlk from="172.23.49.240" to="172.23.49.244"/>
        </fvnsAddrInst>
    </fvTenant>
</polUni>
```

Step 5 Create node management groups.

Example:

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

<polUni>
    <infraInfra>
        <infraFuncP>
            <mgmtGrp name="switchOob">
                <mgmtOoBZone name="default">
                    <mgmtRsAddrInst tDn="uni/tn-mgmt/addrinst-switchOoboobaddr" />
                    <mgmtRsOobEpg tDn="uni/tn-mgmt/mgmtp-default/oob-default" />
                </mgmtOoBZone>
            </mgmtGrp>
        </infraFuncP>
        <mgmtNodeGrp name="switchOob">
            <mgmtRsGrp tDn="uni/infra/funcprof/grp-switchOob" />
            <infraNodeBlk name="default" from_="101" to_="103" />
        </mgmtNodeGrp>
    </infraInfra>
</polUni>
```

Note You can configure the APIC server to use out-of-band management connectivity as the default connectivity mode.

```
POST https://apic-ip-address/api/node/mo/.xml
<polUni>
<fabricInst>
    <mgmtConnectivityPrefs interfacePref="ooband"/>
</fabricInst>
</polUni>
```

Exporting Tech Support, Statistics, and Core Files

About Exporting Files

An administrator can configure export policies in the APIC to export statistics, technical support collections, faults and events, to process core files and debug data from the fabric (the APIC as well as the switch) to any external host. The exports can be in a variety of formats, including XML, JSON, web sockets, secure copy protocol (SCP), or HTTP. You can subscribe to exports in streaming, periodic, or on-demand formats.

An administrator can configure policy details such as the transfer protocol, compression algorithm, and frequency of transfer. Policies can be configured by users who are authenticated using AAA. A security mechanism for the actual transfer is based on a username and password. Internally, a policy element handles the triggering of data.

File Export Guidelines and Restrictions

- HTTP export and the streaming API format is supported only with statistics information. Core and **Tech Support** data are not supported.
- The destination IP for exported files cannot be an IPv6 address.

Note Do not trigger **Tech Support** from more than five nodes simultaneously, especially if they are to be exported into the APIC or to an external server with insufficient bandwidth and compute resources.

In order to collect **Tech Support** from all the nodes in the fabric periodically, you must create multiple policies. Each policy must cover a subset of the nodes and should be scheduled to trigger in a staggered way (at least 30 minutes apart).

Creating a Remote Location for Exporting Files

This procedure configures the host information and file transfer settings for a remote host that will receive exported files.

Procedure

Step 1 In the menu bar, click **Admin**.

Step 2 In the submenu bar, click **Import/Export**.

Step 3 In the **Navigation** pane, expand **Export Policies**.

Step 4 Right-click **Remote Locations** and choose **Create Remote Path of a File**.

Step 5 In the **Create Remote Path of a File** dialog box, perform the following actions:

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

b) In the **Host Name/IP** field, enter an IP address or a fully qualified domain name for the destination host.

c) In the **Protocol** field, click the radio button for the desired file transfer protocol.

d) In the **Remote Path** field, type the path where the file will be stored on the remote host.
e) Enter a username and password for logging in to the remote host and confirm the **Password**.
f) From the **Management EPG** drop-down list, choose the management EPG.
g) Click **Submit**.

Sending an On-Demand Techsupport File Using the GUI

Procedure

Step 1 In the menu bar, click **Admin**.

Step 2 In the submenu bar, click **Import/Export**.

Step 3 In the **Navigation** pane, expand **Export Policies**.

Step 4 Right-click **On-demand TechSupport** and choose **Create On-demand TechSupport**.

The **Create On-demand TechSupport** dialog box appears.

Step 5 Enter the appropriate values in the fields of the **Create On-demand TechSupport** dialog box.

Note For an explanation of a field, click the help icon in the **Create On-demand TechSupport** dialog box. The help file opens to a properties description page.

Step 6 Click **Submit** to send the techsupport file.

Note On-demand tech support files can be saved to another APIC to balance storage and CPU requirements. To verify the location, click on the On-demand TechSupport policy in the **Navigation** pane, then click the **OPERATIONAL** tab in the **Work** pane. The controller is displayed in the **EXPORT LOCATION** field.

Step 7 Right-click the policy name and choose **Collect Tech Support**.

Step 8 Choose **Yes** to begin collecting tech support information.

Sending an On-Demand Techsupport File Using the NX-OS Style CLI

Note Do not trigger techsupport file collection from more than five nodes simultaneously, especially if they are to be exported into the APIC or to an external server with insufficient bandwidth and compute resources.

To avoid excessive storage usage in APIC, remove locally-stored techsupport files promptly.

Before you begin

Configure a remote path for exporting the techsupport file.

Procedure

	Command or Action	Purpose
Step 1	**trigger techsupport** {**all** | **controllers switch** *node-id*} [**remotename** *remote-path-name*] **Example:** `apic1# trigger techsupport switch 101,103 remotename remote5`	Triggers the export of a techsupport file from the controllers, switches, or all to the remote path. For switches, you can specify a range or a comma-separated list. If no remote host is specified, the file is collected in the controller itself.
Step 2	**trigger techsupport host** *host-id* **Example:** `apic1# trigger techsupport host`	Triggers the export of a techsupport file from the specified host to the remote host. If no remote host is specified, the file is collected in the controller itself.
Step 3	**trigger techsupport local** **Example:** `apic1# trigger techsupport local`	Triggers the export of a local techsupport file to the remote host. If no remote host is specified, the file is collected in the controller itself.
Step 4	**show techsupport** {**all** | **controllers switch** *node-id*} **status** **Example:** `apic1# show techsupport switch 101 status`	After a techsupport file is triggered, this command shows the status of the techsupport report.

Examples

This example shows how to trigger a techsupport file for switch 101, to be stored locally on the apic1 controller.

```
apic1# trigger techsupport switch 101

Triggering techsupport for Switch 101 using policy supNode101, setting filters to default
value

Triggered on demand tech support successfully for Switch 101, will be available at:
/data/techsupport on
the controller. Use 'show techsupport' with your options to check techsupport status.
```

Sending an On-Demand TechSupport File Using the REST API

Procedure

Step 1 Set the remote destination for a technical support file using the REST API, by sending a POST with XML such as the following example:

Example:

```
<fileRemotePath userName="" remotePort="22" remotePath="" protocol="sftp" name="ToSupport"
 host="192.168.200.2"
dn="uni/fabric/path-ToSupport" descr="">

<fileRsARemoteHostToEpg tDn="uni/tn-mgmt/mgmtp-default/oob-default"/>

</fileRemotePath>
```

Step 2 Generate an on-demand technical support file using the REST API by sending a POST with XML such as the following:

Example:

```
<dbgexpTechSupOnD upgradeLogs="no" startTime="unspecified" name="Tech_Support_9-20-16"
exportToController="no"
endTime="unspecified" dn="uni/fabric/tsod-Tech_Support_9-20-16" descr="" compression="gzip"

category="forwarding" adminSt="untriggered">

<dbgexpRsExportDest tDn="uni/fabric/path-ToSupport"/>

<dbgexpRsTsSrc tDn="topology/pod-1/node-102/sys"/>

<dbgexpRsTsSrc tDn="topology/pod-1/node-103/sys"/>

<dbgexpRsTsSrc tDn="topology/pod-1/node-101/sys"/>

<dbgexpRsData tDn="uni/fabric/tscont"/>

</dbgexpTechSupOnD>
```

Overview

This topic provides information on:

- How to use configuration Import and Export to recover configuration states to the last known good state using the Cisco APIC
- How to encrypt secure properties of Cisco APIC configuration files

You can do both scheduled and on-demand backups of user configuration. Recovering configuration states (also known as "roll-back") allows you to go back to a known state that was good before. The option for that is called an Atomic Replace. The configuration import policy (configImportP) supports atomic + replace (importMode=atomic, importType=replace). When set to these values, the imported configuration overwrites the existing configuration, and any existing configuration that is not present in the imported file is deleted. As long as you do periodic configuration backups and exports, or explicitly trigger export with a known good configuration, then you can later restore back to this configuration using the following procedures for the CLI, REST API, and GUI.

For more detailed conceptual information about recovering configuration states using the Cisco APIC, please refer to the *Cisco Application Centric Infrastructure Fundamentals Guide.*

The following section provides conceptual information about encrypting secure properties of configuration files:

Configuration File Encryption

As of release 1.1(2), the secure properties of APIC configuration files can be encrypted by enabling AES-256 encryption. AES encryption is a global configuration option; all secure properties conform to the AES configuration setting. It is not possible to export a subset of the ACI fabric configuration such as a tenant configuration with AES encryption while not encrypting the remainder of the fabric configuration. See the *Cisco Application Centric Infrastructure Fundamentals*, "Secure Properties" chapter for the list of secure properties.

The APIC uses a 16 to 32 character passphrase to generate the AES-256 keys. The APIC GUI displays a hash of the AES passphrase. This hash can be used to see if the same passphrases was used on two ACI fabrics. This hash can be copied to a client computer where it can be compared to the passphrase hash of another ACI fabric to see if they were generated with the same passphrase. The hash cannot be used to reconstruct the original passphrase or the AES-256 keys.

Observe the following guidelines when working with encrypted configuration files:

- Backward compatibility is supported for importing old ACI configurations into ACI fabrics that use the AES encryption configuration option.

  Note Reverse compatibility is not supported; configurations exported from ACI fabrics that have enabled AES encryption cannot be imported into older versions of the APIC software.

- Always enable AES encryption when performing fabric backup configuration exports. Doing so will assure that all the secure properties of the configuration will be successfully imported when restoring the fabric.

  Note If a fabric backup configuration is exported without AES encryption enabled, none of the secure properties will be included in the export. Since such an unencrypted backup would not include any of the secure properties, it is possible that importing such a file to restore a system could result in the administrator along with all users of the fabric being locked out of the system.

- The AES passphrase that generates the encryption keys cannot be recovered or read by an ACI administrator or any other user. The AES passphrase is not stored. The APIC uses the AES passphrase to generate the AES keys, then discards the passphrase. The AES keys are not exported. The AES keys cannot be recovered since they are not exported and cannot be retrieved via the REST API.
- The same AES-256 passphrase always generates the same AES-256 keys. Configuration export files can be imported into other ACI fabrics that use the same AES passphrase.
- For troubleshooting purposes, export a configuration file that does not contain the encrypted data of the secure properties. Temporarily turning off encryption before performing the configuration export removes the values of all secure properties from the exported configuration. To import such a configuration file that has all secure properties removed, use the import merge mode; do not use the import replace mode. Using the import merge mode will preserve the existing secure properties in the ACI fabric.
- By default, the APIC rejects configuration imports of files that contain fields that cannot be decrypted. Use caution when turning off this setting. Performing a configuration import inappropriately when this

default setting is turned off could result in all the passwords of the ACI fabric to be removed upon the import of a configuration file that does not match the AES encryption settings of the fabric.

Note Failure to observe this guideline could result in all users, including fabric administrations, being locked out of the system.

Configuring a Remote Location Using the GUI

This procedure explains how to create a remote location using the APIC GUI.

Procedure

Step 1 On the menu bar, choose **ADMIN > Import/Export**.

Step 2 In the navigation pane, right-click **Remote Locations** and choose **Create Remote Location**. The **Create Remote Location** dialog appears.

Step 3 Enter the appropriate values in the **Create Remote Location** dialog fields.

Note For an explanation of a field, click the 'i' icon to display the help file.

Step 4 When finished entering values in the **Create Remote Location** dialog fields, click **Submit**. You have now created a remote location for backing up your data.

Configuring a Remote Location Using the NX-OS Style CLI

In the ACI fabric, you can configure one or more remote destinations for exporting techsupport or configuration files.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	**[no] remote path** *remote-path-name* **Example:** `apic1(config)# remote path myFiles`	Enters configuration mode for a remote path.
Step 3	**user** *username* **Example:** `apic1(config-remote)# user admin5`	Sets the user name for logging in to the remote server. You are prompted for a password.

	Command or Action	Purpose
Step 4	**path** {**ftp** | **scp** | **sftp**} *host*[*:port*] [**remote-directory**] **Example:** `apic1(config-remote)# path sftp filehost.example.com:21 remote-directory /reports/apic`	Sets the path and protocol to the remote server. You are prompted for a password.

Examples

This example shows how to configure a remote path for exporting files.

```
apic1# configure
apic1(config)# remote path myFiles
apic1(config-remote)# user admin5
You must reset the password when modifying the path:
Password:
Retype password:
apic1(config-remote)# path sftp filehost.example.com:21 remote-directory /reports/apic
You must reset the password when modifying the path:
Password:
Retype password:
```

Configuring a Remote Location Using the REST API

This procedure explains how to create a remote location using the REST API.

```
<fileRemotePath name="local" host="host or ip" protocol="ftp|scp|sftp" remotePath="path to
 folder" userName="uname" userPasswd="pwd" />
```

Configuring an Export Policy Using the GUI

This procedure explains how to configure an Export policy using the APIC GUI. Follow these steps to trigger a backup of your data.

Procedure

Step 1 On the menu bar, choose **Admin > Import/Export**.

Step 2 In the navigation pane, right-click **Export Policies** and choose **Create Configuration Export Policy**. The **Create Configuration Export Policy** dialog appears.

Step 3 Enter the appropriate values in the **Create Configuration Export Policy** dialog fields.

Note For an explanation of a field, click the help (**?**) icon to display the help file.

Step 4 When finished entering values in the **Create Configuration Export Policy** dialog fields, click **Submit**. You have now created a backup. You can view this under the **Configuration** tab (The backup file will appear in the **Configuration** pane on the right). There's an **Operational** tab where you can see if it's running, successful, or failed. If you didn't trigger it yet, it is empty. If you created a backup, it creates a file that is

shown in the **Operational** view of the backup file that was created. If you want to then import that data, you must create an Import policy.

Configuring an Export Policy Using the NX-OS Style CLI

Before you begin

If you want to export snapshots according to a schedule, configure a scheduler before configuring the export policy.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	**[no] snapshot export** *policy-name* **Example:** `apic1(config)# snapshot export myExportPolicy`	Creates a policy for exporting snapshots.
Step 3	**format** {**xml** | **json**} **Example:** `apic1(config-export)# format json`	Specifies the data format for the exported configuration file.
Step 4	(Optional) **[no] schedule** *schedule-name* **Example:** `apic1(config-export)# schedule EveryEightHours`	Specifies an existing scheduler for exporting snapshots.
Step 5	(Optional) **[no] target** [**infra** | **fabric** | *tenant-name*] **Example:** `apic1(config-export)# target tenantExampleCorp`	Assigns the target of the export, which can be fabric, infra, a specific tenant, or none. If no target is specified, all configuration information is exported. The default is no target.
Step 6	(Optional) **[no] remote path** *remote-path-name* **Example:** `apic1(config-export)# remote path myBackupServer`	Specifies the name of a configured remote path to which the file will be sent. If no remote path is specified, the file is exported locally to a folder in the controller. The default is no remote path.
Step 7	**end** **Example:** `apic1(config-export)# end`	Returns to EXEC mode.

	Command or Action	Purpose
Step 8	Required: **trigger snapshot export** *policy-name* **Example:** `apic1# trigger snapshot export myExportPolicy`	Executes the snapshot export task. If the export policy is configured with a scheduler, this step is unnecessary unless you want an immediate export.

Examples

This example shows how to configure the periodic export of a JSON-format snapshot file for a specific tenant configuration.

```
apic1# configure
apic1(config)# snapshot export myExportPolicy
apic1(config-export)# format json
apic1(config-export)# target tenantExampleCorp
apic1(config-export)# schedule EveryEightHours
```

Configuring an Export Policy Using the REST API

To configure an export policy using the REST API:

```
POST
https://<ip-of-apic>/api/mo/uni/fabric.xml
<fabricInst dn="uni/fabric">
<configExportP name="export" format="xml" adminSt="triggered">
<configRsExportDestination tnFileRemotePathName="backup" />
</configExportP>
<fileRemotePath name="backup" host="10.10.10.1" protocol="scp"
remotePath="/home/user" userName="user" userPasswd="pass" />
</fabricInst>
```

Configuring an Import Policy Using the GUI

This procedure explains how to configure an Import policy using the APIC GUI. Follow these steps to import your backed up data:

Procedure

Step 1 On the menu bar, choose **ADMIN > Import/Export**.

Step 2 In the navigation pane, right-click **Import Policies** and click **Create Configuration Import Policy**. The **Create Configuration Import Policy** dialog appears.

Step 3 Enter the appropriate values in the **Create Configuration Import Policy** dialog fields.

Note For an explanation of a field, click the 'i' icon to display the help file. For more detailed information on import types and modes including (**Replace, Merge, Best Effort,** and **Atomic**), refer to the *Cisco Application Centric Infrastructure Fundamentals Guide* .

Step 4 When finished entering values in the **Create Configuration Import Policy** dialog fields, click **Submit**.

Configuring an Import Policy Using the NX-OS Style CLI

To configure an import policy using the NX-OS Style CLI, enter the following:

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	**[no] snapshot import** *policy-name* **Example:** `apic1(config)# snapshot import myImportPolicy`	Creates a policy for importing snapshots.
Step 3	**file** *filename* **Example:** `apic1(config-import)# file ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz`	Specifies the name of the file to be imported.
Step 4	**action** {**merge** | **replace**} **Example:** `apic1(config-import)# action replace`	Specifies whether the imported configuration settings will be merged with the current settings or whether the imported configuration will completely replace the current configuration.
Step 5	**[no] mode** {**atomic** | **best-effort**} **Example:** `apic1(config-import)# mode atomic`	Specifies how the import process handles configuration errors when applying the imported settings. The best-effort import mode allows skipping individual configuration errors in the archive, while atomic mode cancels the import upon any configuration error.
Step 6	(Optional) **[no] remote path** *remote-path-name* **Example:** `apic1(config-import)# remote path myBackupServer`	Specifies the name of a configured remote path from which the file will be imported. If no remote path is specified, the file is imported locally from a folder in the controller. The default is no remote path.
Step 7	**end** **Example:** `apic1(config-import)# end`	Returns to EXEC mode.
Step 8	Required: **trigger snapshot import** *policy-name*	Executes the snapshot import task.

	Command or Action	Purpose
	Example: `apic1# trigger snapshot import myImportPolicy`	

Examples

This example shows how to configure and execute the importing of a snapshot file to replace the current configuration.

```
apic1# show snapshot files
File    : ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
Created : 2015-11-21T01:00:21.167+00:00
Root    :
Size    : 22926

apic1# configure
apic1(config)# snapshot import myImportPolicy
apic1(config-import)# file ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
apic1(config-import)# action replace
apic1(config-import)# mode atomic
apic1(config-import)# end
apic1# trigger snapshot import myImportPolicy
```

Configuring an Import Policy Using the REST API

To configure an import policy using the REST API:

```
POST
https://<ip-of-apic>/api/mo/uni/fabric.xml
<fabricInst dn="uni/fabric">
<configImportP name="imp" fileName="aa.tar.gz" adminSt="triggered" importType="replace"
importMode="best-effort">
<configRsImportSource tnFileRemotePathName="backup" />
</configImportP>
<fileRemotePath name="backup" host="10.10.10.1" protocol="scp"
remotePath="/home/user" userName="user" userPasswd="pass" />
</fabricInst>
```

Encrypting Configuration Files Using the GUI

AES-256 encryption is a global configuration option. When enabled, all secure properties conform to the AES configuration setting. A portion of the ACI fabric configuration can be exported using configuration export with a specific targetDn. However, it is not possible to use REST API to export just a portion of the ACI fabric such as a tenant configuration with secure properties and AES encryption. The secure properties do not get included during REST API requests.

This section explains how to enable AES-256 encryption.

Procedure

Step 1 On the menu bar, choose **ADMIN > AAA**.

Step 2 In the navigation pane, click **AES Encryption Passphrase and Keys for Config Export (and Import)**. The **Global AES Encryption Settings for all Configurations Import and Export** window appears in the right pane.

Step 3 Create a passphrase, which can be between 16 and 32 characters long. There are no restrictions on the type of characters used.

Step 4 Click **SUBMIT**.

Note Once you have created and posted the passphrase, the keys are then generated in the back-end and the passphrase is not recoverable. Therefore, your passphrase is not visible to anyone because the key is automatically generated then deleted. Your backup only works if you know the passphrase (no one else can open it).

The **Key Configured** field now shows **yes**. You now see an encrypted hash (which is not the actual passphrase, but just a hash of it) in the **Encrpyted Passphrase** field.

Step 5 After setting and confirming your passphrase, check the check box next to **Enable Encryption** to turn the AES encryption feature on (checked).

The **Global AES Encryption Settings** field in your export and import policies will now be enabled by default.

Note

- Be sure that the **Fail Import if secure fields cannot be decrypted** check box is checked (which is the default selection) in your import and export policies. We highly recommend that you do not uncheck this box when you import configurations. If you uncheck this box, the system attempts to import all the fields However, any fields that it cannot encrypt are blank/missing. As a result, you could lock yourself out of the system because the admin passwords could go blank/missing (if you lock yourself out of the system, refer to *Cisco APIC Troubleshooting Guide*). Unchecking the box launches a warning message. If the box is checked, there are security checks that prevent lockouts and the configuration does not import.
- When the **Enable Encryption** check box is unchecked (off), encryption is disabled and all exported configurations (exports) are missing the secure fields (such as passwords and certificates). When this box is checked (on), encryption is enabled and all exports show the secure fields.
- After enabling encryption, you cannot configure a passphrase when creating a new import or export policy. The passphrase you previously set is now global across all configurations in this box and across all tenants. If you export a configuration from this tab (you have configured a passphrase and enabled encryption) you get a complete backup file. If encryption is not enabled, you get a backup file with the secure properties removed. These backup files are useful when exporting to TAC support engineers, for example, because all the secure fields are missing. This is true for any secure properties in the configuration. There is also a clear option that clears the encryption key.

Note the list of the configuration import behaviors and associated results in the following table:

Configuration Import Behavior Scenario	Result
Old configuration from previous release	Import of configurations from old releases is fully supported and successfully imports all secure fields stored in old configurations.
Configuration import when AES encryption is not configured	If the import is for a configuration without secure fields, it is successful with the behavior previously described. If the imported configuration has secure fields, it is rejected.
Configuration import when AES passphrases do not match	If the import is for a configuration without secure fields, it is successful with the behavior previously described. If the imported configuration has secure fields, it is rejected.
Configuration import when AES passphrases match	Import is successful
Configuration import when AES passphrases do not match for copy/pasted fields	This specific case occurs when you have copied and pasted secure fields from other configurations that were exported with a different passphrase. During the first pass parsing of the imported backup file, if any property fails to decrypt correctly, the import

Configuration Import Behavior Scenario	Result
	fails without importing any shards. Therefore, if a shard fails to decrypt all properties, all shards are rejected.

Encrypting Configuration Files Using the NX-OS Style CLI

To encrypt a configuration file using the NX-OS Style CLI:

```
apic1# configure
apic1(config)# crypto aes
 <CR>
apic1(config)# crypto aes
apic1(config-aes)#
 clear-encryption-key  Clears AES encryption key
encryption            Enable AES Encryption
no                    Negate a command or set its defaults
passphrase            Configure passphrase for AES encryption

bash                  bash shell for unix commands
end                   Exit to the exec mode
exit                  Exit from current mode
fabric                show fabric related information
show                  Show running system information
where                 show the current mode
apic1(config-aes)# encryption
 <CR>
apic1(config-aes)# encryption
apic1(config-aes)#
 clear-encryption-key  Clears AES encryption key
encryption            Enable AES Encryption
no                    Negate a command or set its defaults
passphrase            Configure passphrase for AES encryption

bash                  bash shell for unix commands
end                   Exit to the exec mode
exit                  Exit from current mode
fabric                show fabric related information
show                  Show running system information
where                 show the current mode
apic1(config-aes)# passphrase
 WORD  Passphrase for AES encryption (Range of chars: 16-32) in quotes
apic1(config-aes)# passphrase "abcdefghijklmnopqrstuvwxyz"
apic1(config-aes)#
```

Encrypting Configuration Files Using the REST API

Procedure

To encrypt a configuration file using the REST API, send a post with XML such as the following example:

Example:

```
https://apic-ip-address/api/mo/uni/fabric.xml
<pkiExportEncryptionKey passphrase="abcdefghijklmnopqrstuvwxyz"
strongEncryptionEnabled="true"/>
```

Backing up, Restoring, and Rolling Back Controller Configuration

This section describes the set of features for backing up (creating snapshots), restoring, and rolling back a controller configuration.

Backing Up, Restoring, and Rolling Back Configuration Files Workflow

This section describes the workflow of the features for backing up, restoring, and rolling back configuration files. All of the features described in this document follow the same workflow pattern. Once the corresponding policy is configured, **admintSt** must be set to **triggered** in order to trigger the job.

Once triggered, an object of type **configJob** (representing that run) is created under a container object of type **configJobCont**. (The naming property value is set to the policy DN.) The container's **lastJobName** field can be used to determine the last job that was triggered for that policy.

Note Up to five **configJob** objects are kept under a single job container at a time, with each new job triggered. The oldest job is removed to ensure this.

The **configJob** object contains the following information:

- Execution time
- Name of the file being processed/generated
- Status, as follows:
  - Pending
  - Running
  - Failed
  - Fail-no-data
  - Success
  - Success-with-warnings
- Details string (failure messages and warnings)
- Progress percentage = 100 * lastStepIndex/totalStepCount
- Field lastStepDescr indicating what was being done last

About the fileRemotePath Object

The fileRemotePath object holds the following remote location-path parameters:

- Hostname or IP
- Port
- Protocol: FTP, SCP, and others
- Remote directory (not file path)
- Username
- Password

Note The password must be resubmitted every time changes are made.

Sample Configuration

The following is a sample configuration:

Under **fabricInst** (uni/fabric), enter:

```
<fileRemotePath name="path-name" host="host name or ip" protocol="scp"
remotePath="path/to/some/folder" userName="user-name" userpasswd="password" />
```

Configuration Export to Controller

The configuration export extracts user-configurable managed object (MO) trees from all thirty-two shards in the cluster, writes them into separate files, then compresses them into a tar gzip. The configuration export then uploads the tar gzip to a pre-configured remote location (configured via **configRsRemotePath** pointing to a **fileRemotePath** object) or stores it as a **snapshot** on the controller(s).

Note See the Snapshots section for more details.

The **configExportP** policy is configured as follows:

- **name** - policy name
- **format** - format in which the data is stored inside the exported archive (xml or json)
- **targetDn** - the domain name (DN) of the specific object you want to export (empty means everything)
- **snapshot** - when true, the file is stored on the controller, no remote location configuration is needed
- **includeSecureFields** - Set to true by default, indicates whether the encrypted fields (passwords, etc.) should be included in the export archive.

Note The **configSnapshot** object is created holding the information about this snapshot (see the Snapshots section).

Scheduling Exports

An export policy can be linked with a scheduler, which triggers the export automatically based on a pre-configured schedule. This is done via the **configRsExportScheduler** relation from the policy to a **trigSchedP** object (see the following Sample Configuration section).

Note A scheduler is optional. A policy can be triggered at any time by setting the adminSt to **triggered**.

Troubleshooting

If you get an error message indicating that the generated archive could not be uploaded to the remote location, refer to the Connectivity Issues section.

Sample Configuration Using the NX-OS Style CLI

The following is a sample configuration using the NX-OS Style CLI:

```
apic1(config)# snapshot
 download  Configuration snapshot download setup mode
export    Configuration export setup mode
import    Configuration import setup mode
rollback  Configuration rollback setup mode
upload    Configuration snapshot upload setup mode
apic1(config)# snapshot export policy-name
apic1(config-export)#
 format    Snapshot format: xml or json
no        Negate a command or set its defaults
remote    Set the remote path configuration will get exported to
schedule  Schedule snapshot export
target    Snapshot target

bash      bash shell for unix commands
end       Exit to the exec mode
exit      Exit from current mode
fabric    show fabric related information
show      Show running system information
where     show the current mode
apic1(config-export)# format xml
apic1(config-export)# no remote path             [If no remote path is specified, the file
 is exported locally to a folder in the controller]
apic1(config-export)# target                     [Assigns the target of the export, which
can be fabric, infra, a specific tenant, or none. If no target is specified, all configuration
 information is exported.]
WORD  infra, fabric or tenant-x
apic1(config-export)#
apic1# trigger snapshot export policy-name       [Executes the snapshot export task]
apic1# ls /data2                                 [If no remote path is specified, the
configuration export file is saved locally to the controller under the folder data2]
ce_Dailybackup.tgz
```

Sample Configuration Using the GUI

The following is a sample configuration using the GUI:

1. On the menu bar, click the **ADMIN** tab.
2. Select **IMPORT/EXPORT**.
3. Under **Export Policies**, select **Configuration**.
4. Under Configuration, click the configuration that you would like to roll back to. For example, you can click **defaultOneTime**, which is the default.
5. Next to **Format**, select a button for either JSON or XML format.
6. Next to **Start Now**, select a button for either **No** or **Yes** to indicate whether you want to trigger now or trigger based on a schedule. (The easiest method is to choose to trigger immediately.)
7. For the **Target DN** field, enter the name of the tenant configuration you are exporting.
8. If you want to store the configuration on the controller itself, check the **Snapshot** option. If you want to configure a remote location, uncheck this option.
9. For the **Scheduler** field, you have the option to create a scheduler instructing when and how often to export the configuration.
10. For the **Encryption** field, you have the option to enable or disable the encryption of your configuration file.
11. When you have finished your configuration, click **Start Now**.
12. Click **SUBMIT** to trigger your configuration export.

Sample Configuration Using REST API

The following is a sample configuration using the REST API:

```
<configExportP name="policy-name" format="xml" targetDn="/some/dn or empty which means
everything"
snapshot="false" adminSt="triggered">
<configRsRemotePath tnFileRemotePathName="some remote path name" />
<configRsExportScheduler tnTrigSchedPName="some scheduler name" />
</configExportP>
```

Note When providing a remote location, if you set the snapshot to True, the backup ignores the remote path and stores the file on the controller.

Configuration Import to Controller

Configuration import downloads, extracts, parses, analyzes and applies the specified, previously exported archive one shard at a time in the following order: infra, fabric, tn-common, then everything else. The fileRemotePath configuration is performed the same way as for export (via configRsRemotePath). Importing snapshots is also supported.

The **configImportP** policy is configured as follows:

- **name** - policy name
- **fileName** - name of the archive file (not the path file) to be imported
- **importMode**

- Best-effort mode: each MO is applied individually, and errors only cause the invalid MOs to be skipped.

Note If the object is not present on the controller, none of the children of the object get configured. Best-effort mode attempts to configure the children of the object.

- Atomic mode: configuration is applied by whole shards. A single error causes whole shard to be rolled back to its original state.

- **importType**
  - replace - Current system configuration is replaced with the contents or the archive being imported (only atomic mode is supported)
  - merge - Nothing is deleted, archive content is applied on top the existing system configuration.
- **snapshot** - when true, the file is taken from the controller and no remote location configuration is needed.
- **failOnDecryptErrors** - (true by default) the file fails to import if the archive was encrypted with a different key than the one that is currently set up in the system.

Troubleshooting

The following scenarios may need troubleshooting:

- If the generated archive could not be downloaded from the remote location, refer to the Connectivity Issues section.
- If the import succeeded with warnings, check the details.
- If a file could not be parsed, refer to the following scenarios:
  - If the file is not a valid XML or JSON file, check whether or not the files from the exported archive were manually modified.
  - If an object property has an unknown property or property value, it may be because:
    - The property was removed or an unknown property value was manually entered
    - The model type range was modified (non-backward compatible model change)
    - The naming property list was modified
- If an MO could not be configured, note the following:
  - Best-effort mode logs the error and skips the MO
  - Atomic mode logs the error and skips the shard

Sample Configuration Using the NX-OS Style CLI

The following is a sample configuration using the NX-OS Style CLI:

```
apic1# configure
apic1(config)# snapshot
 download  Configuration snapshot download setup mode
export    Configuration export setup mode
import    Configuration import setup mode
rollback  Configuration rollback setup mode
upload    Configuration snapshot upload setup mode
apic1(config)# snapshot import
 WORD       Import configuration name
default
rest-user
apic1(config)# snapshot import policy-name
apic1(config-import)#
 action  Snapshot import action merge|replace
file    Snapshot file name
mode    Snapshot import mode atomic|best-effort
no      Negate a command or set its defaults
remote  Set the remote path configuration will get imported from

bash    bash shell for unix commands
end     Exit to the exec mode
exit    Exit from current mode
fabric  show fabric related information
show    Show running system information
where   show the current mode
apic1(config-import)# file < from "show snapshot files" >
apic1(config-import)# no remote path
apic1(config-import)#
apic1# trigger snapshot import policy-name       [Executes the snapshot import task]
```

Sample Configuration Using the GUI

The following is a sample configuration using the GUI:

1. On the menu bar, click the **ADMIN** tab.
2. Select **IMPORT/EXPORT**.
3. Under **Import Policies**, select **Configuration**.
4. Under **Configuration**, select **Create Configuration Import Policy**. The **CREATE CONFIGURATION IMPORT POLICY** window appears.
5. In the **Name** field, the file name must match whatever was backed up and will have a very specific format. The file name is known to whoever did the backup.
6. The next two options relate to recovering configuration states (also known as "roll-back"). The options are **Input Type** and **Input Mode**. When you recover a configuration state, you want to roll back to a known state that was good before. The option for that is an **Atomic Replace**.
7. If you want to store the configuration on the controller itself, check the **Snapshot** option. If you want to configure a remote location, uncheck this option.
8. In the **Import Source** field, specify the same remote location that you already created.
9. For the **Encryption** field, you have the option to enable or disable the encryption

   of your configuration file.
10. Click **SUBMIT** to trigger your configuration import.

Sample Configuration Using the REST API

The following shows a sample configuration using the REST API:

```
<configImportP name="policy-name" fileName="someexportfile.tgz" importMode="atomic"
importType="replace" snapshot="false" adminSt="triggered">
<configRsRemotePath tnFileRemotePathName="some remote path name" />
</configImportP>
```

Snapshots

Snapshots are configuration backup archives, stored (and replicated) in a controller managed folder. To create one, an export can be performed with the **snapshot** property set to true. In this case, no remote path configuration is needed. An object of **configSnapshot** type is created to expose the snapshot to the user.

You can create recurring snapshots, which are saved to **Admin** > **Import/Export** > **Export Policies** > **Configuration** > **defaultAuto**.

configSnapshot objects provide the following:

- file name
- file size
- creation date
- root DN indicating what the snapshot is of (fabric, infra, specific tenant, and so on)
- ability to remove a snapshot (by setting the retire field to true)

To import a snapshot, first create an import policy. Navigate to **Admin** > **Import/Export** and click **Import Policies**. Right click and choose **Create Configuration Import Policy** to set the import policy attributes.

Snapshot Manager Policy

The **configSnapshotManagerP** policy allows you to create snapshots from remotely stored export archives. You can attach a remote path to the policy, provide the file name (same as with configImportP), set the mode to download, and trigger. The manager downloads the file, analyzes it to make sure the archive is valid, stores it on the controller, and creates the corresponding configSnapshot object.

You can also create a recurring snapshot.

Note When enabled, recurring snapshots are saved to **Admin** > **Import/Export** > **Export Policies** > **Configuration** > **defaultAuto**.

The snapshot manager also allows you to upload a snapshot archive to a remote location. In this case, the mode must be set to upload.

Troubleshooting

For troubleshooting, refer to the Connectivity Issues section.

Snapshot Upload from Controller to Remote Path Using the NX-OS CLI

```
apic1(config)# snapshot upload policy-name
apic1(config-upload)#
 file    Snapshot file name
no      Negate a command or set its defaults
remote  Set the remote path configuration will get uploaded to

bash    bash shell for unix commands
end     Exit to the exec mode
exit    Exit from current mode
fabric  show fabric related information
show    Show running system information
where   show the current mode
apic1(config-upload)# file <file name from "show snapshot files">
apic1(config-upload)# remote path remote-path-name
apic1# trigger snapshot upload policy-name       [Executes the snapshot upload task]
```

Snapshot Download from Controller to Remote Path Using the NX-OS CLI

```
apic1(config)# snapshot download policy-name
apic1(config-download)#
 file    Snapshot file name
no      Negate a command or set its defaults
remote  Set the remote path configuration will get downloaded from

bash    bash shell for unix commands
end     Exit to the exec mode
exit    Exit from current mode
fabric  show fabric related information
show    Show running system information
where   show the current mode
apic1(config-download)# file < file from remote path>
apic1(config-download)# remote path remote-path-name
apic1# trigger snapshot download policy-name       [Executes the snapshot download task]
```

Snapshot Upload and Download Using the GUI

To upload a snapshot file to a remote location:

1. Right-click on the snapshot file listed in the **Config Rollbacks** pane, and select the **Upload to Remote Location option**. The **Upload snapshot to remote location** box appears.
2. Click **SUBMIT**.

To download a snapshot file from a remote location:

1. Click the import icon on the upper right side of the screen. The **Import remotely stored export archive to snapshot** box appears.
2. Enter the file name in the **File Name** field.
3. Select a remote location from the Import Source pull-down, or check the box next to **Or create a new one** to create a new remote location.
4. Click **SUBMIT**.

Snapshot Upload and Download Using the REST API

```
<configSnapshotManagerP name="policy-name" fileName="someexportfile.tgz"
mode="upload|download" adminSt="triggered">
```

```
<configRsRemotePath tnFileRemotePathName="some remote path name" />
</configSnapshotManagerP>
```

Rollback

The **configRollbackP** policy is used to undo the changes made between two snapshots. Objects are processed as follows:

- Deleted MOs are recreated
- Created MOs are deleted
- Modified MOs are reverted

Note The rollback feature only operates on snapshots. Remote archives are not supported. To use one, the snapshot manager can be used to create a snapshot from it for the rollback. The policy does not require a remote path configuration. If one is provided, it will be ignored.

Rollback Workflow

The policy snapshotOneDN and snapshotTwoDn fields must be set and the first snapshot (S1) must precede snapshot two (S2). Once triggered, snapshots are extracted and analyzed, and the difference between them is calculated and applied.

MOs are located that are:

- Present in S1 but not present in S2 - these MOs are deleted and rollback re-creates them
- Not present in S1 but not present in S2 - these MOs are created after S1 and rollback deletes them if:
  - These MOs are not modified after S2 is taken
  - None of the MO's descendants are created or modified after S2 is taken
- Present in both S1 and S2, but with different property values - these MO properties are reverted to S1, unless the property was modified to a diffferent value after S2 is taken. In this case, it is left as is.

  The rollback feature also generates a diff file that contains the confiuration generated as a result of these calculations. Applying this configuration is the last step of the rollback process. The content of this file can be retrieved via a special REST API called readiff: apichost/mqapi2/snapshots.readiff.xml?jobdn=SNAPSHOT_JOB_DN.

  Rollback (which is difficult to predict) also has a preview mode (set preview to true), which prevents rollback from making any actual changes. It calculates and generates the diff file, allowing you to preview what exactly is going to happen once the rollback is actually performed.

Diff Tool

Another special REST API is available, which provides diff functionality between two snapshots: apichost/mqapi2/snapshots.diff.xml?s1dn=SNAPSHOT_ONE_DN&s2dn=SNAPSHOT_TWO_DN.

Sample Configuration Using the NX-OS Style CLI

This example shows how to configure and execute a rollback using the NX-OS Style CLI:

```
apic1# show snapshot files
File    : ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
Created : 2015-11-21T01:00:21.167+00:00
Root    :
Size    : 22926

File    : ce2_DailyAutoBackup-2015-11-21T09-00-21.tar.gz
Created : 2015-11-21T09:00:24.025+00:00
Root    :
Size    : 23588

apic1# configure
apic1(config)# snapshot rollback myRollbackPolicy
apic1(config-rollback)# first-file ce2_DailyAutoBackup-2015-11-21T01-00-17.tar.gz
apic1(config-rollback)# second-file ce2_DailyAutoBackup-2015-11-21T09-00-21.tar.gz
apic1(config-rollback)# preview
apic1(config-rollback)# end
apic1# trigger snapshot rollback myRollbackPolicy
```

Sample Configuration Using the GUI

This example shows how to configure and execute a rollback using the GUI:

1. On the menu bar, click the **Admin** tab.
2. Click **Config Rollbacks**, located under the Admin tab.
3. Select the first configuration file from the **Config Rollbacks** list (in the left-side pane).
4. Select the second configuration file in the **Configuration for selected snapshot** pane (in the right-side pane).
5. Click the **Compare with previous snapshot** drop-down menu (at the bottom of the right-side pane), then select the second configuration file from that list. A diff file is then generated so that you can compare the differences between the two snapshots.

Note After the file generates, there is an option to undo these changes.

Sample Configuration Using the REST API

This example shows how to configure and execute a rollback using the REST API:

```
<configRollbackP name="policy-name" snapshotOneDn="dn/of/snapshot/one"
snapshotOneDn="dn/of/snapshot/two" preview="false" adminSt="triggered" />
```

Using Syslog

About Syslog

During operation, a fault or event in the Cisco Application Centric Infrastructure (ACI) system can trigger the sending of a system log (syslog) message to the console, to a local file, and to a logging server on another

system. A system log message typically contains a subset of information about the fault or event. A system log message can also contain audit log and session log entries.

Note For a list of syslog messages that the APIC and the fabric nodes can generate, see http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/syslog/guide/aci_syslog/ACI_SysMsg.html.

Many system log messages are specific to the action that a user is performing or the object that a user is configuring or administering. These messages can be the following:

- Informational messages, providing assistance and tips about the action being performed
- Warning messages, providing information about system errors related to an object, such as a user account or service profile, that the user is configuring or administering

In order to receive and monitor system log messages, you must specify a syslog destination, which can be the console, a local file, or one or more remote hosts running a syslog server. In addition, you can specify the minimum severity level of messages to be displayed on the console or captured by the file or host. The local file for receiving syslog messages is `/var/log/external/messages`.

A syslog source can be any object for which an object monitoring policy can be applied. You can specify the minimum severity level of messages to be sent, the items to be included in the syslog messages, and the syslog destination.

You can change the display format for the Syslogs to NX-OS style format.

Additional details about the faults or events that generate these system messages are described in the *Cisco APIC Faults, Events, and System Messages Management Guide*, and system log messages are listed in the *Cisco ACI System Messages Reference Guide*.

Note Not all system log messages indicate problems with your system. Some messages are purely informational, while others may help diagnose problems with communications lines, internal hardware, or the system software.

Creating a Syslog Destination and Destination Group

This procedure configures syslog data destinations for logging and evaluation. You can export syslog data to the console, to a local file, or to one or more syslog servers in a destination group.

Procedure

Step 1 In the menu bar, click **Admin**.

Step 2 In the submenu bar, click **External Data Collectors**.

Step 3 In the **Navigation** pane, expand **Monitoring Destinations**.

Step 4 Right-click **Syslog** and choose **Create Syslog Monitoring Destination Group**.

Step 5 In the **Create Syslog Monitoring Destination Group** dialog box, perform the following actions:

a) In the group and profile **Name** field, enter a name for the monitoring destination group and profile.

b) In the group and profile **Format** field, choose the format for Syslog messages.

The default is **aci**, or the RFC 5424 compliant message format, but you can choose to set it to the NX-OS style format instead.

c) In the group and profile **Admin State** drop-down list, choose **enabled**.
d) To enable sending of syslog messages to a local file, choose **enabled** from the Local File Destination **Admin State** drop-down list and choose a minimum severity from the Local File Destination **Severity** drop-down list.

The local file for receiving syslog messages is `/var/log/external/messages`.

e) To enable sending of syslog messages to the console, choose **enabled** from the Console Destination **Admin State** drop-down list and choose a minimum severity from the Console Destination **Severity** drop-down list.
f) Click **Next**.
g) In the **Create Remote Destinations** area, click + to add a remote destination.

Caution Risk of hostname resolution failure for remote Syslog destinations, if the DNS server used is configured to be reachable over in-band connectivity. To avoid the issue, configure the Syslog server using the IP address, or if you use a hostname, ensure that the DNS server is reachable over an out-of-band interface.

Step 6 In the **Create Syslog Remote Destination** dialog box, perform the following actions:
a) In the **Host** field, enter an IP address or a fully qualified domain name for the destination host.
b) (Optional) In the **Name** field, enter a name for the destination host.
c) In the **Admin State** field, click the **enabled** radio button.
d) (Optional) Choose a minimum severity **Severity**, a **Port** number, and a syslog **Forwarding Facility**.
e) From the **Management EPG** drop-down list, choose the management endpoint group.
f) Click **OK**.

Step 7 (Optional) To add more remote destinations to the remote destination group, click + again and repeat the steps in the **Create Syslog Remote Destination** dialog box

Step 8 Click **Finish**.

Creating a Syslog Source

A syslog source can be any object for which an object monitoring policy can be applied.

Before you begin

Create a syslog monitoring destination group.

Procedure

Step 1 From the menu bar and the navigation frame, navigate to a **Monitoring Policies** menu for the area of interest.

You can configure monitoring policies for tenants, fabric, and access.

Step 2 Expand **Monitoring Policies**, then select and expand a monitoring policy.

Under **Fabric > Fabric Policies > Monitoring Policies > Common Policy** is a basic monitoring policy that applies to all faults and events and is automatically deployed to all nodes and controllers in the fabric. Alternatively, you can specify an existing policy with a more limited scope.

Step 3 Under the monitoring policy, click **Callhome/SNMP/Syslog**.

Step 4 In the **Work** pane, choose **Syslog** from the **Source Type** drop-down list.

Step 5 From the **Monitoring Object** list, choose a managed object to be monitored.

If the desired object does not appear in the list, follow these steps:

a) Click the Edit icon to the right of the **Monitoring Object** drop-down list.
b) From the **Select Monitoring Package** drop-down list, choose an object class package.
c) Select the checkbox for each object that you want to monitor.
d) Click **Submit**.

Step 6 In a tenant monitoring policy, if you select a specific object instead of **All**, a **Scope** selection appears.

In the **Scope** field, select a radio button to specify the system log messages to send for this object:

- **all**—Send all events and faults related to this object
- **specific event**—Send only the specified event related to this object. From the **Event** drop-down list, choose the event policy.
- **specific fault**—Send only the specified fault related to this object. From the **Fault** drop-down list, choose the fault policy.

Step 7 Click + to create a syslog source.

Step 8 In the **Create Syslog Source** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the syslog source.
b) From the **Min Severity** drop-down list, choose the minimum severity of system log messages to be sent.
c) In the **Include** field, check the checkboxes for the type of messages to be sent.
d) From the **Dest Group** drop-down list, choose the syslog destination group to which the system log messages will be sent.
e) Click **Submit**.

Step 9 (Optional) To add more syslog sources, click + again and repeat the steps in the **Create Syslog Source** dialog box

Enabling Syslog to Display in NX-OS CLI Format, Using the REST API

By default the Syslog format is RFC 5424 compliant. You can change the default display of Syslogs to NX-OS type format, similar to the following example:

```
apic1# moquery -c "syslogRemoteDest"
Total Objects shown: 1

# syslog.RemoteDest
host               : 172.23.49.77
adminState         : enabled
childAction        :
descr              :
```

```
dn                  : uni/fabric/slgroup-syslog-mpod/rdst-172.23.49.77
epgDn               :
format              : nxos
forwardingFacility  : local7
ip                  :
lcOwn               : local
modTs               : 2016-05-17T16:51:57.231-07:00
monPolDn            : uni/fabric/monfab-default
name                : syslog-dest
operState           : unknown
port                : 514
rn                  : rdst-172.23.49.77
severity            : information
status              :
uid                 : 15374
vrfId               : 0
vrfName             :
```

To enable the Syslogs to display in NX-OS type format, perform the following steps, using the REST API.

Procedure

Step 1 Enable the Syslogs to display in NX-OS type format, as in the following example:

```
POST https://192.168.20.123/api/node/mo/uni/fabric.xml
<syslogGroup name="DestGrp77" format="nxos">
<syslogRemoteDest name="slRmtDest77" host="172.31.138.20" severity="debugging"/>
</syslogGroup>
```

The **syslogGroup** is the Syslog monitoring destination group, the **sysLogRemoteDest** is the name you previously configured for your Syslog server, and the **host** is the IP address for the previously configured Syslog server.

Step 2 Set the Syslog format back to the default RFC 5424 format, as in the following example:

```
POST https://192.168.20.123/api/node/mo/uni/fabric.xml
<syslogGroup name="DestGrp77" format="aci">
<syslogRemoteDest name="slRmtDest77" host="172.31.138.20" severity="debugging"/>
</syslogGroup>
```

Using Atomic Counters

About Atomic Counters

Atomic counters allow you to gather statistics about traffic between flows. Using atomic counters, you can detect drops and misrouting in the fabric, enabling quick debugging and isolation of application connectivity issues. For example, an administrator can enable atomic counters on all leaf switches to trace packets from endpoint 1 to endpoint 2. If any leaf switches have nonzero counters, other than the source and destination leaf switches, an administrator can drill down to those leafs.

In conventional settings, it is nearly impossible to monitor the amount of traffic from a bare metal NIC to a specific IP address (an endpoint) or to any IP address. Atomic counters allow an administrator to count the number of packets that are received from a bare metal endpoint without any interference to its data path. In addition, atomic counters can monitor per-protocol traffic that is sent to and from an endpoint or an application group.

Leaf-to-leaf (TEP-to-TEP) atomic counters can provide the following:

- Counts of sent, received, dropped, and excess packets
  - Sent packets: The sent number reflects how many packets were sent from the source TEP (tunnel endpoint) to the destination TEP.
  - Received packets: The received number reflects how many packets the destination TEP received from the source TEP.
  - Dropped packets: The dropped number reflects how many packets were dropped during transmission. This number is the difference in the amount of packets sent and the amount of packets received.
  - Excess packets: The excess number reflects how many extra packets were received during transmission. This number is the amount of packets that were unexpectedly received due to a forwarding mismatch or a misrouting to the wrong place.
- Short-term data collection such as the last 30 seconds, and long-term data collection such as 5 minutes, 15 minutes, or more
- A breakdown of per-spine traffic (available when the number of TEPs, leaf or VPC, is less than 64)
- Ongoing monitoring

Note Leaf-to-leaf (TEP to TEP) atomic counters are cumulative and cannot be cleared. However, because 30-second atomic counters reset at 30-second intervals, they can be used to isolate intermittent or recurring problems. Atomic counters require an active fabric Network Time Protocol (NTP) policy.

Tenant atomic counters can provide the following:

- Application-specific counters for traffic across the fabric, including sent, received, dropped, and excess packets
- Modes include the following:
  - EPtoEP (endpoint to endpoint)
  - EPGtoEPG (endpoint group to endpoint group)

    Note For EPGtoEPG, the options include ipv4 only, ipv6 only, and ipv4, ipv6. Any time there is an ipv6 option, you use twice the TCAM entries, which means the scale numbers may be less than expected for pure ipv4 policies.
  - EPGtoEP (endpoint group to endpoint)
  - EPtoAny (endpoint to any)
  - AnytoEP (any to endpoint)

- EPGtoIP (endpoint group to IP, used only for external IP address)
- EPtoExternalIP (endpoint to external IP address)

Atomic Counters Guidelines and Restrictions

- Use of atomic counters is not supported when the endpoints are in different tenants or in different contexts (VRFs) within the same tenant.
- In Cisco APIC Release 3.1(2m) (and later), if no statistics have been generated on a path in the lifetime of the fabric, no atomic counters are generated for the path. Also the **Traffic Map** in the **Visualization** tab (**Operations** > **Visualization** in the APIC GUI) does not show all paths, only the active paths (paths that had traffic at some point in the fabric lifetime).
- In pure Layer 2 configurations where the IP address is not learned (the IP address is 0.0.0.0), endpoint-to-EPG and EPG-to-endpoint atomic counter policies are not supported. In these cases, endpoint-to-endpoint and EPG-to-EPG policies are supported. External policies are virtual routing and forwarding (VRF)-based, requiring learned IP addresses, and are supported.
- When the atomic counter source or destination is an endpoint, the endpoint must be dynamic and not static. Unlike a dynamic endpoint (fv:CEp), a static endpoint (fv:StCEp) does not have a child object (fv:RsCEpToPathEp) that is required by the atomic counter.
- In a transit topology, where leaf switches are not in full mesh with all spine switches, then leaf-to-leaf (TEP to TEP) counters do not work as expected.
- For leaf-to-leaf (TEP to TEP) atomic counters, once the number of tunnels increases the hardware limit, the system changes the mode from trail mode to path mode and the user is no longer presented with per-spine traffic.
- The atomic counter does not count spine proxy traffic.
- Packets dropped before entering the fabric or before being forwarded to a leaf port are ignored by atomic counters.
- Packets that are switched in the hypervisor (same Port Group and Host) are not counted.
- Atomic counters require an active fabric Network Time Protocol (NTP) policy.
- Atomic counters work for IPv6 sources and destinations but configuring source and destination IP addresses across IPv4 and IPv6 addresses is not allowed.
- An atomic counter policy configured with fvCEp as the source and/or destination counts only the traffic that is from/to the MAC and IP addresses that are present in the fvCEp managed objects (MOs). If the fvCEp MO has an empty IP address field, then all traffic to/from that MAC address would be counted regardless of the IP address. If the APIC has learned multiple IP addresses for an fvCEp, then traffic from only the one IP address in the fvCEp MO itself is counted as previously stated. In order to configure an atomic counter policy to/from a specific IP address, use the fvIp MO as the source and/or destination.
- If there is an fvIp behind an fvCEp, you must add fvIP-based policies and not fvCEp-based policies.

Configuring Atomic Counters

Procedure

Step 1 In the menu bar, click **Tenants**.

Step 2 In the submenu bar, click the desired tenant.

Step 3 In the **Navigation** pane, expand the tenant and expand **Policies** and then expand **Troubleshoot**.

Step 4 Under **Troubleshoot**, expand **Atomic Counter Policy** and choose a traffic topology.

You can measure traffic between a combination of endpoints, endpoint groups, external interfaces, and IP addresses.

Step 5 Right-click the desired topology and choose **Add** *topology* **Policy** to open an **Add Policy** dialog box.

Step 6 In the **Add Policy** dialog box, perform the following actions:

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

b) choose or enter the identifying information for the traffic source.

The required identifying information differs depending on the type of source (endpoint, endpoint group, external interface, or IP address).

c) choose or enter the identifying information for the traffic destination.

d) (Optional) (Optional) In the **Filters** table, click the + icon to specify filtering of the traffic to be counted.

In the resulting **Create Atomic Counter Filter** dialog box, you can specify filtering by the IP protocol number (TCP=6, for example) and by source and destination IP port numbers.

e) Click **Submit** to save the atomic counter policy.

Step 7 In the **Navigation** pane, under the selected topology, choose the new atomic counter policy.

The policy configuration is displayed in the **Work** pane.

Step 8 In the **Work** pane, click the **Operational** tab and click the **Traffic** subtab to view the atomic counter statistics.

Using SNMP

About SNMP

The Cisco Application Centric Infrastructure (ACI) provides extensive SNMPv1, v2, and v3 support, including Management Information Bases (MIBs) and notifications (traps). The SNMP standard allows any third-party applications that support the different MIBs to manage and monitor the ACI fabric.

SNMPv3 provides extended security. Each SNMPv3 device can be selectively enabled or disabled for SNMP service. In addition, each device can be configured with a method of handling SNMPv1 and v2 requests.

For more information about using SNMP, see the *Cisco ACI MIB Quick Reference*.

SNMP Access Support in ACI

SNMP support in ACI is as follows:

- SNMP read queries (Get, Next, Bulk, Walk) are supported by leaf and spine switches and by APIC.
- SNMP write commands (Set) are not supported by leaf and spine switches or by APIC.
- SNMP traps (v1, v2c, and v3) are supported by leaf and spine switches and by APIC.

Note ACI supports a maximum of 10 trap receivers.

- SNMPv3 is supported by leaf and spine switches and by APIC.

Table 9: SNMP Support Changes by Cisco APIC Release

Release	Description
1.2(2)	IPv6 support is added for SNMP trap destinations.
1.2(1)	SNMP support for the APIC controller is added. Previous releases support SNMP only for leaf and spine switches.

For the complete list of MIBs supported in ACI, see http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/mib/list/mib-support.html.

SNMP Trap Aggregation

The SNMP Trap Aggregation feature allows SNMP traps from the fabric nodes to be delivered to one of the APICs in the cluster and allows the forwarding of SNMP traps received from the fabric nodes to the external destination.

In order to handle the partition tolerance of APIC cluster, the SNMP trap aggregation must be configured on more than one APIC. You can configure multiple trap destinations in the SNMP policy. See Configuring the SNMP Policy Using the GUI, on page 87

The SNMP Trap Aggregation feature was introduced in the APIC release 3.1(1). In this release, SNMPV2 trap aggregation and forwarding is supported.

If an APIC is decommissioned, the user is expected to clean reboot the decommissioned APIC. Since SNMP Trap Aggregation functionality is active on decommissioned APICs, the user could receive duplicate traps on the trap destination if the decommissioned APIC is not clean rebooted.

Configuring SNMP

Configuring the SNMP Policy Using the GUI

This procedure configures and enables the SNMP policy on ACI switches.

Before you begin

To allow SNMP communications, you must configure the following:

- Configure an out-of-band contract allowing SNMP traffic. SNMP traffic typically uses UDP port 161 for SNMP requests.

- Configure the APIC out-of-band IP addresses in the 'mgmt' tenant. Although the out-of-band addresses are configured during APIC setup, the addresses must be explicitly configured in the 'mgmt' tenant before the out-of-band contract will take effect.

Procedure

Step 1 In the menu bar, click **Fabric**.

Step 2 In the submenu bar, click **Fabric Policies**.

Step 3 In the **Navigation** pane, expand **Pod Policies**.

Step 4 Under **Pod Policies**, expand **Policies**.

Step 5 Right-click **SNMP** and choose **Create SNMP Policy**.

As an alternative to creating a new SNMP policy, you can edit the **default** policy fields in the same manner as described in the following steps.

Step 6 In the SNMP policy dialog box, perform the following actions:

a) In the **Name** field, enter an SNMP policy name.
b) In the **Admin State** field, select **Enabled**.
c) (Optional) In the **SNMP v3 Users** table, click the + icon, enter a **Name**, enter the user's authentication data, and click **Update**.

This step is needed only if SNMPv3 access is required.

d) In the **Community Policies** table, click the + icon, enter a **Name** (include only alphanumeric characters and do not include the @ symbol) and click **Update**.
e) In the **Trap Forward Servers** table, click the + icon, enter the **IP Address** of the external server and click **Update**.

Step 7 Required: To configure allowed SNMP management stations, perform the following actions in the SNMP policy dialog box:

a) In the **Client Group Policies** table, click the + icon to open the **Create SNMP Client Group Profile** dialog box.
b) In the **Name** field, enter an SNMP client group profile name.
c) From the **Associated Management EPG** drop-down list, choose the management EPG.
d) In the **Client Entries** table, click the + icon.
e) Enter a client's name in the **Name** field, enter the client's IP address in the **Address** field, and click **Update**.

Note When an SNMP management station connects with APIC using SNMPv3, APIC does not enforce the client IP address specified in the SNMP client group profile. For SNMPv3, the management station must exist in the **Client Entries** list, but the IP address need not match, as the SNMPv3 credentials alone are sufficient for access.

Step 8 Click **OK**.

Step 9 Click **Submit**.

Step 10 Under **Pod Policies**, expand **Policy Groups** and choose a policy group or right-click **Policy Groups** and choose **Create POD Policy Group**.

You can create a new pod policy group or you can use an existing group. The pod policy group can contain other pod policies in addition to the SNMP policy.

Step 11 In the pod policy group dialog box, perform the following actions:
a) In the **Name** field, enter a pod policy group name.
b) From the **SNMP Policy** drop-down list, choose the SNMP policy that you configured and click **Submit**.

Step 12 Under **Pod Policies**, expand **Profiles** and click **default**.

Step 13 In the **Work pane**, from the **Fabric Policy Group** drop-down list, choose the pod policy group that you created.

Step 14 Click **Submit**.

Step 15 Click **OK**.

Configuring an SNMP Trap Destination Using the GUI

This procedure configures the host information for an SNMP manager that will receive SNMP trap notifications.

Note ACI supports a maximum of 10 trap receivers. If you configure more than 10, some will not receive notifications.

Procedure

Step 1 In the menu bar, click **Admin**.

Step 2 In the submenu bar, click **External Data Collectors**.

Step 3 In the **Navigation** pane, expand **Monitoring Destinations**.

Step 4 Right-click **SNMP** and choose **Create SNMP Monitoring Destination Group**.

Step 5 In the **Create SNMP Monitoring Destination Group** dialog box, perform the following actions:
a) In the **Name** field, enter an SNMP destination name and click **Next**.
b) In the **Create Destinations** table, click the + icon to open the **Create SNMP Trap Destination** dialog box.
c) In the **Host Name/IP** field, enter an IP address or a fully qualified domain name for the destination host.

Note Cisco APIC Release 1.2(2) and later releases support IPv6 SNMP trap destinations.

d) Choose the **Port** number and SNMP **Version** for the destination.
e) For SNMP v1 or v2c destinations, enter one of the configured community names as the **Security Name** and choose **noauth** as **v3 Security Level**.

SNMP community names cannot contain the @ symbol.

f) For SNMP v3 destinations, enter one of the configured SNMP v3 user names as **Security Name** and choose the desired **v3 Security Level**.
g) From the **Management EPG** drop-down list, choose the management EPG.
h) Click **OK**.
i) Click **Finish**.

Configuring an SNMP Trap Source Using the GUI

This procedure selects and enables a source object within the fabric to generate SNMP trap notifications.

Procedure

Step 1 In the menu bar, click **Fabric**.

Step 2 In the submenu bar, click **Fabric Policies**.

Step 3 In the **Navigation** pane, expand **Monitoring Policies**.

You can create an SNMP source in the **Common Policy**, the **default** policy, or you can create a new monitoring policy.

Step 4 Expand the desired monitoring policy and choose **Callhome/SNMP/Syslog**.

If you chose the **Common Policy**, right-click **Common Policy**, choose **Create SNMP Source**, and follow the instructions below for that dialog box.

Step 5 In the **Work** pane, from the **Monitoring Object** drop-down list, choose **ALL**.

Step 6 From the **Source Type** drop-down list, choose **SNMP**.

Step 7 In the table, click the + icon to open the **Create SNMP Source** dialog box.

Step 8 In the **Create SNMP Source** dialog box, perform the following actions:

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

b) From the **Dest Group** drop-down list, choose an existing destination for sending notifications or choose **Create SNMP Monitoring Destination Group** to create a new destination.

The steps for creating an SNMP destination group are described in a separate procedure.

c) Click **Submit**.

Monitoring the System Using SNMP

You can remotely monitor individual hosts (APIC or another host) and find out the state of any particular node.

You can check the system's CPU and memory usage using SNMP to find out if the CPU is spiking or not. The SNMP, a network management system, uses an SNMP client and accesses information over the APIC and retrieves information back from it.

You can remotely access the system to figure out if the information is in the context of the network management system and you can learn whether or not it is taking too much CPU or memory, or if there are any system or performance issues. Once you learn the source of the issue, you can check the system health and verify whether or not it is using too much memory or CPU.

Refer to the *Cisco ACI MIB Quick Reference Manual* for additional information.

Configuring SNMP Policy Using CLI

Use this procedure to configure SNMP policy.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**template snmp-fabric default** **Example:** `apic1(config)# template snmp-fabric default`	Creates a SNMP policy.
Step 3	**snmp-server clientgroup**	Configures SNMP client group. A client group is a group of client IP addresses that allows SNMP access to routers or switches.
Step 4	**snmp-server community** **Example:** `apic1(config-template-snmp-fabric)# snmp-server community abc`	Configures SNMP community. The SNMP community profile enables access to the router or switch statistics for monitoring.
Step 5	**snmp-server contact**	Configures SNMP contact information.

Step 6	**snmp-server host**
	Example:
	`apic1(config-template-snmp-fabric)# snmp-server host 2001:420:28e:2020::10 traps-version 2c abc`
	`apic1(config-template-snmp-fabric)# snmp-server host 2001:420:28e:2020::2 traps-version 2c abc`
	`apic1(config-template-snmp-fabric)# snmp-server host 2001:420:28e:2020::11 traps-version 2c abc`
		Configures SNMP trap host.

Step 7	**snmp-server location**	Configures SNMP location.
Step 8	**snmp-server protocol** **Example:** `apic1(config-template-snmp-fabric)# snmp-server protocol enable`	Configures SNMP protocol.
Step 9	**snmp-server trap-fwd-server** **Example:** `apic1(config-template-snmp-fabric)# snmp-server trap-fwd-server 172.31.128.199`	Configures SNMP trap forwarding server.

	Command or Action	Purpose
Step 10	**snmp-server user** **Example:** `apic1(config-template-snmp-fabric)# snmp-server user test_user auth hmac-md5-96 '' priv none privacy-passphrase ''`	Configures SNMP user. The SNMP user profile is used to associate users with SNMP policies for monitoring devices in a network.
Step 11	**show running-config**	Verifies the configuration.

Using SPAN

About SPAN

You can use the Switched Port Analyzer (SPAN) utility to perform detailed troubleshooting or to take a sample of traffic from a particular application host for proactive monitoring and analysis.

SPAN copies traffic from one or more ports, VLANs, or endpoint groups (EPGs) and sends the copied traffic to one or more destinations for analysis by a network analyzer. The process is nondisruptive to any connected devices and is facilitated in the hardware, which prevents any unnecessary CPU load.

You can configure SPAN sessions to monitor traffic received by the source (ingress traffic), traffic transmitted from the source (egress traffic), or both. By default, SPAN monitors all traffic, but you can configure filters to monitor only selected traffic.

Multinode SPAN

APIC traffic monitoring policies can SPAN policies at the appropriate places to track members of each application group and where they are connected. If any member moves, APIC automatically pushes the policy to the new leaf switch. For example, when an endpoint VMotions to a new leaf switch, the SPAN configuration automatically adjusts.

SPAN Guidelines and Restrictions

- Use SPAN only for troubleshooting. SPAN traffic competes with user traffic for switch resources. To minimize the load, configure SPAN to copy only the specific traffic that you want to analyze.
- You cannot specify an l3extLIfP layer 3 subinterface as a SPAN source. You must use the entire port for monitoring traffic from external sources.
- In local SPAN for FEX interfaces—the FEX interfaces can only be used as SPAN sources, not SPAN destinations. On Generation 1 switches (Cisco Nexus 9000 Series switches without EX or FX on the switch name), Tx SPAN does not work for any Layer 3 switched traffic. On Generation 2 switches (with EX or FX on the switch name), Tx SPAN does not work whether traffic is Layer 2 or Layer 3 switched. There are no limitations for Rx SPAN.

- Tenant and access SPAN use the encapsulated remote extension of SPAN (ERSPAN) type I, while fabric SPAN uses ERSPAN type II. For information regarding ERSPAN headers, refer to the IETF Internet Draft at this URL: https://tools.ietf.org/html/draft-foschiano-erspan-00.
- ERSPAN destination IPs must be learned in the fabric as an endpoint.
- SPAN supports IPv6 traffic but the destination IP for the ERSPAN cannot be an IPv6 address.
- See the *Verified Scalability Guide for Cisco ACI* document for SPAN-related limits, such as the maximum number of active SPAN sessions.

Configuring a SPAN Session

This procedure shows how to configure a SPAN policy to forward replicated source packets to a remote traffic analyzer.

Procedure

Step 1 In the menu bar, click **Tenants**.

Step 2 In the submenu bar, click the tenant that contains the source endpoint.

Step 3 In the **Navigation** pane, expand the tenant, expand **Policies** > **Troubleshoot**, and expand **SPAN**.

Step 4 Under **SPAN**, right-click **SPAN Destination Groups** and choose **Create SPAN Destination Group**.
The **Create SPAN Destination Group** dialog appears.

Step 5 Enter the appropriate values in the required fields of the **Create SPAN Destination Group** dialog box then click **OK** and **Submit**.

Note For a description of a field, click the information icon (i) at the top-right corner of the dialog box to display the help file.

Step 6 Under **SPAN**, right-click **SPAN Source Groups** and choose **Create SPAN Source Group**.
The **Create SPAN Source Group** dialog appears.

Step 7 Enter the appropriate values in the required fields of the **Create SPAN Source Group** dialog box then click **OK** and **Submit**.

Note For a description of a field, click the information icon (i) at the top-right corner of the dialog box to display the help file.

What to do next

Using a traffic analyzer at the SPAN destination, you can observe the data packets from the SPAN source EPG to verify the packet format, addresses, protocols, and other information.

Using Traceroute

About Traceroute

The traceroute tool is used to discover the routes that packets actually take when traveling to their destination. Traceroute identifies the path taken on a hop-by-hop basis and includes a time stamp at each hop in both directions. You can use traceroute to test the connectivity of ports along the path between the generating device and the device closest to the destination. If the destination cannot be reached, the path discovery traces the path up to the point of failure.

A traceroute that is initiated from the tenant endpoints shows the default gateway as an intermediate hop that appears at the ingress leaf switch.

Traceroute supports a variety of modes, including:

- Endpoint-to-endpoint, and leaf-to-leaf (tunnel endpoint, or TEP to TEP)
- Endpoint-to-external-IP
- External-IP-to-endpoint
- External-IP-to-external-IP

Traceroute discovers all paths across the fabric, discovers point of exits for external endpoints, and helps to detect if any path is blocked.

Traceroute Guidelines and Restrictions

- When the traceroute source or destination is an endpoint, the endpoint must be dynamic and not static. Unlike a dynamic endpoint (fv:CEp), a static endpoint (fv:StCEp) does not have a child object (fv:RsCEpToPathEp) that is required for traceroute.
- Traceroute works for IPv6 source and destinations but configuring source and destination IP addresses across IPv4 and IPv6 addresses is not allowed.
- See the *Verified Scalability Guide for Cisco ACI* document for traceroute-related limits.
- When an endpoint moves from one ToR switch to a different ToR switch that has a new MAC address (one that is different than the MAC address that you specified while configuring the traceroute policy), the traceroute policy shows "missing-target" for the endpoint. In this scenario you must configure a new traceroute policy with the new MAC address.

Performing a Traceroute Between Endpoints

Procedure

Step 1 In the menu bar, click **Tenants**.

Step 2 In the submenu bar, click the tenant that contains the source endpoint.

Step 3 In the **Navigation** pane, expand the tenant and expand **Policies** > **Troubleshoot**.

Step 4 Under **Troubleshoot**, right-click on one of the following traceroute policies:

- **Endpoint-to-Endpoint Traceroute Policies** and choose **Create Endpoint-to-Endpoint Traceroute Policy**
- **Endpoint-to-External-IP Traceroute Policies** and choose **Create Endpoint-to-External-IP Traceroute Policy**
- **External-IP-to-Endpoint Traceroute Policies** and choose **Create External-IP-to-Endpoint Traceroute Policy**
- **External-IP-to-External-IP Traceroute Policies** and choose **Create External-IP-to-External-IP Traceroute Policy**

Step 5 Enter the appropriate values in the dialog box fields and click **Submit**.

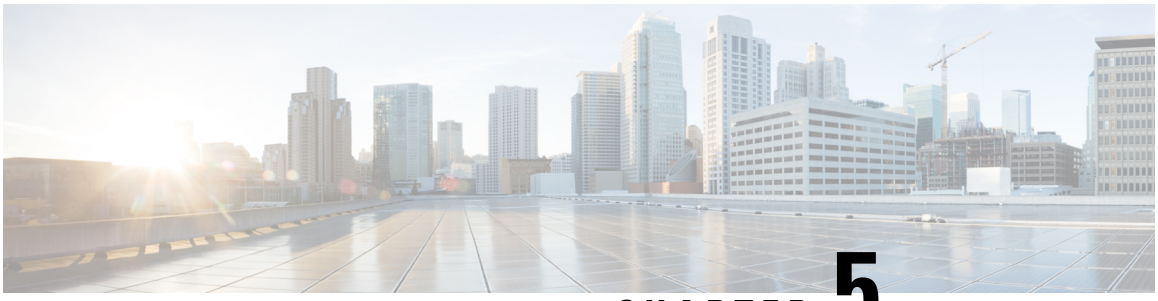
Note For the description of a field, click the help icon (**?**) in the top-right corner of the dialog box.

Step 6 In the **Navigation** pane or the **Traceroute Policies** table, click the traceroute policy.
The traceroute policy is displayed in the **Work** pane.

Step 7 In the **Work** pane, click the **Operational** tab, click the **Source Endpoints** tab, and click the **Results** tab.

Step 8 In the **Traceroute Results** table, verify the path or paths that were used in the trace.

Note
- More than one path might have been traversed from the source node to the destination node.
- For readability, you can increase the width of one or more columns, such as the **Name** column.



CHAPTER 5

Provisioning Core ACI Fabric Services

This chapter contains the following sections:

Time Synchronization and NTP

Within the Cisco Application Centric Infrastructure (ACI) fabric, time synchronization is a crucial capability upon which many of the monitoring, operational, and troubleshooting tasks depend. Clock synchronization is important for proper analysis of traffic flows as well as for correlating debug and fault time stamps across multiple fabric nodes.

An offset present on one or more devices can hamper the ability to properly diagnose and resolve many common operational issues. In addition, clock synchronization allows for the full utilization of the atomic counter capability that is built into the ACI upon which the application health scores depend. Nonexistent or improper configuration of time synchronization does not necessarily trigger a fault or a low health score. You should configure time synchronization before deploying a full fabric or applications so as to enable proper usage of these features. The most widely adapted method for synchronizing a device clock is to use Network Time Protocol (NTP).

Prior to configuring NTP, consider what management IP address scheme is in place within the ACI fabric. There are two options for configuring management of all ACI nodes and Application Policy Infrastructure Controllers (APICs), in-band management and/or out-of-band management. Depending upon which management option is chosen for the fabric, configuration of NTP will vary. Another consideration in deploying time synchronization is where the time source is located. The reliability of the source must be carefully considered when determining if you will use a private internal clock or an external public clock.

In-Band and Out-of-Band Management NTP

Note See the Adding Management Access section in this guide for information about in-band management access and out-of-band management access.

- Out-of-band management NTP—When an ACI fabric is deployed with out-of-band management, each node of the fabric, inclusive of spines, leaves, and all members of the APIC cluster, is managed from outside the ACI fabric. This IP reachability will be leveraged so that each node can individually query the same NTP server as a consistent clock source. To configure NTP, a Date and Time policy must be created that references an out-of-band management endpoint group. Date and Time policies are confined to a single pod and must be deployed across all pods provisioned in the ACI fabric.
- In-Band Management NTP—When an ACI fabric is deployed with in-band management, consider the reachability of the NTP server from within the ACI in-band management network. In-band IP addressing used within the ACI fabric is not reachable from anywhere outside the fabric. To leverage an NTP server external to the fabric with in-band management, construct a policy to enable this communication. The steps used to configure in-band management policies are identical to those used to establish an out-of-band management policy. The distinction is around how to allow the fabric to connect to the NTP server.

NTP over IPv6

NTP over IPv6 addresses is supported in hostnames and peer addresses. The gai.conf can also be set up to prefer the IPv6 address of a provider or a peer over an IPv4 address. The user can provide a hostname that can be resolved by providing an IP address (both IPv4 or IPv6, depending on the installation or preference).

Configuring NTP Using the GUI

Procedure

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

Step 2 In the **Navigation** pane, choose **Pod Policies** > **Policies**.

Step 3 In the **Work** pane, choose **Actions** > **Create Date and Time Policy**.

Step 4 In the **Create Date and Time Policy** dialog box, perform the following actions:

a) Enter a name for the policy to distinguish between the different NTP configurations in your environment..

b) Click **enabled** for the **Authentication State** field and expand the **NTP Client Authentication Keys** table and enter the key information. Click **Update** and **Next**.

c) Click the + sign to specify the NTP server information (provider) to be used.

d) In the **Create Providers** dialog box, enter all relevant information, including the following fields: **Name**, **Description**, **Minimum Polling Intervals**, and **Maximum Polling Intervals**.

- If you are creating multiple providers, check the **Preferred** check box for the most reliable NTP source.

- In the Management EPG drop-down list, if the NTP server is reachable by all nodes on the fabric through out-of-band management, choose Out-of-Band. If you have deployed in-band management, see the details about In-Band Management NTP. Click **OK**.

Repeat the steps for each provider that you want to create.

Step 5 In the **Navigation** pane, choose **Pod Policies** > **Policy Groups**.

Step 6 In the **Work** pane, choose **Actions** > **Create Pod Policy Group**.

Step 7 In the **Create Pod Policy Group** dialog box, perform the following actions:

a) Enter a name for the policy group.

b) In the **Date Time Policy** field, from the drop down list, choose the NTP policy that you created earlier. Click **Submit**.
The pod policy group is created. Alternatively, you can use the default pod policy group.

Step 8 In the **Navigation** pane, choose **Pod Policies** > **Profiles**.

Step 9 In the **Work** pane, double-click the desired pod selector name.

Step 10 In the Properties area, from the **Fabric Policy Group** drop down list, choose the pod policy group you created. Click **Submit**.

Configuring NTP Using the NX-OS Style CLI

When an ACI fabric is deployed with out-of-band management, each node of the fabric is managed from outside the ACI fabric. You can configure an out-of-band management NTP server so that each node can individually query the same NTP server as a consistent clock source.

Procedure

Step 1 **configure**

Enters configuration mode.

Example:

```
apic1# configure
```

Step 2 **template ntp-fabric** *ntp-fabric-template-name*

Specifies the NTP template (policy) for the fabric.

Example:

```
apic1(config)# template ntp-fabric pol1
```

Step 3 **[no] server** *dns-name-or-ipaddress* **[prefer]** [**use-vrf** {**inb-mgmt** | **oob-mgmt**}] [**key** *key-value*]

Configures an NTP server for the active NTP policy. To make this server the preferred server for the active NTP policy, include the **prefer** keyword. If NTP authentication is enabled, specify a reference key ID. To specify the tenant in-band or out-of-band management access VRF, include the **use-vrf** keyword with the **inb-mgmt** or **oob-mgmt** keyword.

Example:

```
apic1(config-template-ntp-fabric)# server 192.0.20.123 prefer use-vrf oob-mgmt
```

Step 4 **[no] authenticate**

Enables (or disables) NTP authentication.

Example:

```
apic1(config-template-ntp-fabric)# no authenticate
```

Step 5 **[no] authentication-key** *key-value*

Configures an authentication NTP authentication. The range is 1 to 65535.

Example:

```
apic1(config-template-ntp-fabric)# authentication-key 12345 md5 "key_value"
```

Step 6 **[no] trusted-key** *key-value*

Configures a trusted NTP authentication. The range is 1 to 65535.

Example:

```
apic1(config-template-ntp-fabric)# trusted-key 54321
```

Step 7 **exit**

Returns to global configuration mode

Example:

```
apic1(config-template-ntp-fabric)# exit
```

Step 8 **template pod-group** *pod-group-template-name*

Configures a pod-group template (policy).

Example:

```
apic1(config)# template pod-group allPods
```

Step 9 **inherit ntp-fabric** *ntp-fabric-template-name*

Configures the NTP fabric pod-group to use the previously configured NTP fabric template (policy).

Example:

```
apic1(config-pod-group)# inherit ntp-fabric pol1
```

Step 10 **exit**

Returns to global configuration mode

Example:

```
apic1(config-template-pod-group)# exit
```

Step 11 **pod-profile** *pod-profile-name*

Configures a pod profile.

Example:

```
apic1(config)# pod-profile all
```

Step 12 **pods** {*pod-range-1-255* | **all**}

Configures a set of pods.

Example:

```
apic1(config-pod-profile)# pods all
```

Step 13 **inherit pod-group** *pod-group-name*

Associates the pod-profile with the previously configured pod group.

Example:

```
apic1(config-pod-profile-pods)# inherit pod-group allPods
```

Step 14 **end**

Returns to EXEC mode.

Example:

```
apic1(config-pod-profile-pods)# end
```

Examples

This example shows how to configure a preferred out-of-band NTP server and how to verify the configuration and deployment.

```
apic1# configure t
apic1(config)# template ntp-fabric pol1
apic1(config-template-ntp-fabric)# server 192.0.20.123 use-vrf oob-default
apic1(config-template-ntp-fabric)# no authenticate
apic1(config-template-ntp-fabric)# authentication-key 12345 md5 abcdef1235
apic1(config-template-ntp-fabric)# trusted-key 12345
apic1(config-template-ntp-fabric)# exit
apic1(config)# template pod-group allPods
apic1(config-pod-group)# inherit ntp-fabric pol1
apic1(config-pod-group)# exit
apic1(config)# pod-profile all
apic1(config-pod-profile)# pods all
apic1(config-pod-profile-pods)# inherit pod-group allPods
apic1(config-pod-profile-pods)# end
apic1#
```

```
apic1# show ntpq
 nodeid     remote         refid   st    t   when   poll   reach  delay    offset   jitter

 ------  -  ------------  ------  ----  --  -----  -----  -----  ------   ------   ------
 1       *  192.0.20.123  .GPS.         u   27     64     377    76.427   0.087    0.067

 2       *  192.0.20.123  .GPS.         u   3      64     377    75.932   0.001    0.021
 3       *  192.0.20.123  .GPS.         u   3      64     377    75.932   0.001    0.021
```

Configuring NTP Using the REST API

Procedure

Step 1 Configure NTP.

Example:

```
POST url: https://APIC-IP/api/node/mo/uni/fabric/time-test.xml

<imdata totalCount="1">
    <datetimePol adminSt="enabled" authSt="disabled" descr="" dn="uni/fabric/time-CiscoNTPPol"
 name="CiscoNTPPol" ownerKey="" ownerTag="">
        <datetimeNtpProv descr="" keyId="0" maxPoll="6" minPoll="4" name="10.10.10.11"
preferred="yes">
            <datetimeRsNtpProvToEpg tDn="uni/tn-mgmt/mgmtp-default/inb-default"/>
        </datetimeNtpProv>
    </datetimePol>
</imdata>
```

Step 2 Add the default Date Time Policy to the pod policy group.

Example:

```
POST url: https://APIC-IP/api/node/mo/uni/fabric/funcprof/podpgrp-calo1/rsTimePol.xml

POST payload: <imdata totalCount="1">
<fabricRsTimePol tnDatetimePolName="CiscoNTPPol">
</fabricRsTimePol>
</imdata>
```

Step 3 Add the pod policy group to the default pod profile.

Example:

```
POST url:
https://APIC-IP/api/node/mo/uni/fabric/podprof-default/pods-default-typ-ALL/rspodPGrp.xml

payload: <imdata totalCount="1">
<fabricRsPodPGrp tDn="uni/fabric/funcprof/podpgrp-calo1" status="created">
</fabricRsPodPGrp>
</imdata>
```

Verifying NTP Operation Using the GUI

Procedure

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

Step 2 In the **Navigation** pane, choose **Pod Policies** > **Policies** > **Date and Time** > ***ntp_policy*** > ***server_name***.

The *ntp_policy* is the previously created policy. An IPv6 address is supported in the Host Name/IP address field. If you enter a hostname and it has an IPv6 address set, you must implement the priority of IPv6 address over IPv4 address.

Step 3 In the **Work** pane, verify the details of the server.

Verifying NTP Policy Deployed to Each Node Using the NX-OS Style CLI

Procedure

Step 1 Log onto an APIC controller in the fabric using the SSH protocol.

Step 2 Attach to a node and check the NTP peer status, shown as follows:

```
apic1# fabric  node_name show ntp peer-status
```

Step 3 Repeat step 2 for different nodes in the fabric.

NTP Server

The NTP server enables client switches to also act as NTP servers to provide NTP time information to downstream clients. When the NTP server is enabled, the NTP daemon on the switch responds with time information to all unicast (IPv4/IPv6) requests from NTP clients. NTP server implementation is complaint to NTP RFCv3. As per NTP RFC, server will not maintain any state related to clients.

- NTP Server enables the in-band/out-of-band management IP of the switches to serve NTP client requests.
- NTP Server, like existing NTP Client functionality works only with In-band & Out-of-band Management VRFs.
- NTP Server responds to incoming NTP requests on both Management VRFs, and responds back using the same VRF.
- NTP Server supports both IPv4/IPv6.
- Switches can sync as IPv4 Client and serve as IPv6 server and vice versa.
- Switches can sync as NTP client via out-of-band management VRF and serve through in-band management VRF and vice versa.
- No additional Contracts or IP Table Configurations are required.
- If the switch is synced to the upstream server, then the server will send time info with stratum number, an increment to its system peer's stratum.
- If the switch clock is undisciplined (not synced to upstream server), then the server will send time information with stratum 16. Clients will not be able to sync to this server.

By default, NTP server functionality is disabled. It needs to be enabled explicitly by config policy.

Note Clients can use the in-band, out-of-band IP of the leaf as the NTP server IP. Clients can also use the BD SVI of the EPG which they are part of, also as NTP server IP.

Note Fabric switches should not sync to other switches of the same fabric. The Fabric switches should always sync to external NTP servers.

Enabling the NTP Server Using the GUI

This section explains how to enable an NTP server when configuring NTP in the APIC GUI.

Procedure

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

Step 2 In the **Navigation** pane, choose **Pod Policies** > **Policies** .

The **Date and Time** option appears in the **Navigation** pane.

Step 3 From the **Navigation** pane, right-click on **Date and Time** and choose **Create Date and Time Policy**.

The **Create Date and Time Policy** dialog appears in the **Work** pane.

Step 4 In the **Create Date and Time Policy** dialog box, perform the following actions:

a) Enter a name for the policy to distinguish between the different NTP configurations in your environment.

b) For the **Server State** option, click **enabled**.

Server State enables switches to act as NTP servers to provide NTP time information to downstream clients.

Note To support the server functionality, it is always recommended to have a peer setup for the server. This enables the server to have a consistent time to provide to the clients.

When **Server State** is enabled:

- The NTP server sends time info with a stratum number, an increment to the system peer's stratum number, to switches that are synched to the upstream server.
- The server sends time info with stratum 16 if the switch clock is not synched to the upstream server. Clients are not able to sync to this server.

Note To support the server functionality, it is always recommended to have a peer setup for the server. The peer setup allows for a consistent time to provide to the clients.

c) For the **Master Mode** option, click **enabled**.

Master Mode enables the designated NTP server to provide undisciplined local clock time to downstream clients with a configured stratum number. For example, a leaf switch that is acting as the NTP server can provide undisciplined local clock time to leaf switches acting as clients.

Note
- **Master Mode** is only applicable when the server clock is undisciplined.
- The default master mode **Stratum Value** is 8.

d) For the **Stratum Value** field, specify the stratum level from which NTP clients will get their time synchronized. The range is from 1 to 14.

e) Click **Next**.
f) Click the + sign to specify the NTP server information (provider) to be used.
g) In the **Create Providers** dialog box, enter all relevant information, including the following fields: **Name**, **Description**, **Minimum Polling Intervals**, and **Maximum Polling Intervals**.

- If you are creating multiple providers, check the **Preferred** check box for the most reliable NTP source.
- In the Management EPG drop-down list, if the NTP server is reachable by all nodes on the fabric through out-of-band management, choose Out-of-Band. If you have deployed in-band management, see the details about In-Band Management NTP. Click **OK**.

Repeat the steps for each provider that you want to create.

Step 5 In the **Navigation** pane, choose **Pod Policies** then right-click on **Policy Groups**.

The **Create Pod Policy Group** dialog appears.

Step 6 In the **Work** pane, choose **Actions** > **Create Pod Policy Group**.

Step 7 In the **Create Pod Policy Group** dialog box, perform the following actions:

a) Enter a name for the policy group.
b) In the **Date Time Policy** field, from the drop down list, choose the NTP policy that you created earlier. Click **Submit**.
The pod policy group is created. Alternatively, you can use the default pod policy group.

Step 8 In the **Navigation** pane, choose **Pod Policies** > **Profiles**.

Step 9 In the **Work** pane, double-click the desired pod selector name.

Step 10 In the Properties area, from the **Fabric Policy Group** drop down list, choose the pod policy group you created.

Step 11 Click **Submit**.

Enabling the NTP Server Using the CLI

This section explains how to enable the NTP server feature using CLI commands.

Before you begin

Procedure

Step 1 Enter the global configure mode:

Example:

```
apic1#configure t
```

Step 2 Configure an NTP server for the active NTP policy.

Example:

```
apic1(config)#template ntp-fabric default
```

Step 3 Specify the NTP server.

Example:

```
apic1(config-template-ntp-fabric)#server 10.81.254.201 prefer use-vrf oob-default
```

Step 4 Enable the switches to act as NTP servers.

Example:

```
apic1(config-template-ntp-fabric)#server-mode
```

Step 5 Enable the switches to act in NTP mastermode with a stratum value of 10.

Example:

```
apic1(config-template-ntp-fabric)#master stratum 10
```

Step 6 Return to global configuration

Example:

```
aic1(config-template-ntp-fabric)#exit
```

Enabling the NTP Server Using the REST API

This example demonstrates how to configure the NTP server using the REST API.

Procedure

Enable `serverState` and `masterMode` and specify the `StratumValue` (the `StratumValue` can be from 1-14.

Example:

```
POST url: https://APIC-IP/api/node/mo/uni/fabric/time-test.xml
<datetimePol name="testdatetime" adminSt="enabled" authSt="enabled" serverState="enabled"
masterMode="enabled" StratumValue="10" >
```

Configuring a DHCP Relay Policy

A DHCP relay policy may be used when the DHCP client and server are in different subnets. If the client is on an ESX hypervisor with a deployed vShield Domain profile, then the use of a DHCP relay policy configuration is mandatory.

When a vShield controller deploys a Virtual Extensible Local Area Network (VXLAN), the hypervisor hosts create a kernel (vmkN, virtual tunnel end-point [VTEP]) interface. These interfaces need an IP address in the infrastructure tenant that uses DHCP. Therefore, you must configure a DHCP relay policy so that the APIC can act as the DHCP server and provide these IP addresses.

When an ACI fabric acts as a DHCP relay, it inserts the DHCP Option 82 (the DHCP Relay Agent Information Option) in DHCP requests that it proxies on behalf of clients. If a response (DHCP offer) comes back from a DHCP server without Option 82, it is silently dropped by the fabric. Therefore, when the ACI fabric acts as a DHCP relay, DHCP servers providing IP addresses to compute nodes attached to the ACI fabric must support Option 82.

Configuring a DHCP Server Policy for the APIC Infrastructure Using the GUI

- The port and the encapsulation used by the application Endpoint Group must belong to a physical or VM Manager (VMM) domain. If no such association with a domain is established, the APIC continues to deploy the EPG but raises a fault.
- Cisco APIC supports DHCP relay for both IPv4 and IPv6 tenant subnets. DHCP server addresses can be IPv4 or IPv6. DHCPv6 relay will occur only if IPv6 is enabled on the fabric interface and one or more DHCPv6 relay servers are configured.

Deploying DHCP Relay Policy for an Endpoint Group

Before you begin

Make sure that Layer 2 or Layer 3 management connectivity is configured.

Procedure

Step 1 On the menu bar, choose **TENANTS** > **infra**. In the **Navigation** pane, under **Tenant infra**, expand **Networking** > **Protocol Policies** > **DHCP** > **Relay Policies**.

Step 2 Right-click **Relay Policies** and click **Create DHCP Relay Policy**.

Step 3 In the **Create DHCP Relay Policy** dialog box, perform the following actions:

a) In the **Name** field, enter the DHCP relay profile name (DhcpRelayP).

b) Expand **Providers**. In the **Create DHCP Provider** dialog box, in the **EPG Type** field, click the appropriate radio button depending upon where the DHCP server is connected.

c) In the **Application EPG** area, in the **Tenant** field, from the drop-down list, choose the tenant. (infra)

d) In the **Application Profile** field, from the drop-down list, choose the application. (access)

e) In the **EPG** field, from the drop-down list, choose the EPG. (default)

f) In the **DHCP Server Address** field, enter the IP address for the infra DHCP server. Click **Update**.

Note The infra DHCP IP address is the infra IP address of APIC1. You must enter the default IP address of 10.0.0.1 if deploying for vShield controller configuration.

g) Click **Submit**.

The DHCP relay policy is created.

Step 4 In the **Navigation** pane, expand **Networking** > **Bridge Domains** > **default** > **DHCP Relay Labels**.

Step 5 Right-click **DHCP Relay Labels**, and click **Create DHCP Relay Label**.

Step 6 In the **Create DHCP Relay Label** dialog box, perform the following actions:

a) In the **Scope** field, click the tenant radio button.
This action displays, in the **Name** field drop-down list, the DHCP relay policy created earlier.

b) In the **Name** field, from the drop-down list, choose the name of the DHCP policy created (DhcpRelayP) or create a new relay policy by choosing **Create DHCP Relay Policy**.

c) In the **DHCP Option Policy**, select an existing option policy, or create a new one by choosing **Create DHCP Option Policy**.

d) Click **Submit**.

The DHCP server is associated with the bridge domain.

Step 7 In the **Navigation** pane, expand **Networking** > **Bridge Domains** > **default** > **DHCP Relay Labels** to view the DHCP server created.

Configuring a DHCP Server Policy for the APIC Infrastructure Using the NX-OS Style CLI

- The port and the encapsulation used by the application Endpoint Group must belong to a physical or VM Manager (VMM) domain. If no such association with a domain is established, the APIC continues to deploy the EPG but raises a fault.
- Cisco APIC supports DHCP relay for both IPv4 and IPv6 tenant subnets. DHCP server addresses can be IPv4 or IPv6. DHCPv6 relay will occur only if IPv6 is enabled on the fabric interface and one or more DHCPv6 relay servers are configured.

Before you begin

Ensure that Layer 2 or Layer 3 connectivity is configured to reach the DHCP server address.

Procedure

Configure DHCP server policy settings for the APIC infrastructure traffic.

Example:

DHCP Relay Policy for an Endpoint Group

```
apic1(config)# tenant infra
apic1(config-tenant)# template dhcp relay policy DhcpRelayP
apic1(config-tenant-template-dhcp-relay)# ip address 10.0.0.1 tenant infra application access epg default
apic1(config-tenant-template-dhcp-relay)# exit
apic1(config-tenant)# interface bridge-domain default
apic1(config-tenant-interface)# dhcp relay policy tenant DhcpRelayP
apic1(config-tenant-interface)# exit
```

Example:

DHCP Relay Policy for Layer 3 Outside

```
ifav28-ifc2(config)# tenant dhcpTn
ifav28-ifc2(config-tenant)# template dhcp relay policy DhcpRelayPol
ifav28-ifc2(config-tenant-template-dhcp-relay)# ip address 11.1.1.11 tenant dhcpTn application ap epg serverEpg
ifav28-ifc2(config-tenant-template-dhcp-relay)# exit
ifav28-ifc2(config-tenant)# exit
ifav28-ifc2(config)# leaf 2001
ifav28-ifc2(config-leaf)# interface ethernet 1/4
ifav28-ifc2(config-leaf-if)# no switchport
ifav28-ifc2(config-leaf-if)# vrf member tenant dhcpTn vrf v1
ifav28-ifc2(config-leaf-if)# dhcp relay policy tenant DhcpRelayPol
ifav28-ifc2(config-leaf-if)# exit
```

Configuring a DHCP Server Policy for the APIC Infrastructure Using the REST API

- This task is a prerequisite for users who want to create a vShield Domain Profile.
- The port and the encapsulation used by the application Endpoint Group must belong to a physical or VM Manager (VMM) domain. If no such association with a domain is established, the APIC continues to deploy the EPG but raises a fault.
- Cisco APIC supports DHCP relay for both IPv4 and IPv6 tenant subnets. DHCP server addresses can be IPv4 or IPv6. DHCPv6 relay will occur only if IPv6 is enabled on the fabric interface and one or more DHCPv6 relay servers are configured.

Before you begin

Make sure that Layer 2 or Layer 3 management connectivity is configured.

Procedure

Configure the APIC as the DHCP server policy for the infrastructure tenant.

Note This relay policy will be pushed to all the leaf ports that are connected hypervisors using the attach entity profile configuration. For details about configuring with attach entity profile, see the examples related to creating VMM domain profiles.

Example:

DHCP Relay Policy for EPG

```
<!-- api/policymgr/mo/.xml -->
<polUni>

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


  <fvTenant name="infra">

    <dhcpRelayP name="DhcpRelayP"  owner="tenant">
      <dhcpRsProv tDn="uni/tn-infra/ap-access/epg-default" addr="10.0.0.1" />
    </dhcpRelayP>

    <fvBD name="default">
      <dhcpLbl name="DhcpRelayP" owner="tenant"/>
    </fvBD>

  </fvTenant>
</polUni>
```

Example:

DHCP Relay Policy for Layer 3 Outside

Note You must specify DHCP Relay label under **l3extLIfP** with an appropriate name and owner.

```
<polUni>
  <fvTenant name="dhcpTn">
    <l3extOut name="Out1" >
```

```
      <l3extLNodeP name="NodeP" >
        <l3extLIfP name="Intf1">
           <dhcpLbl name="DhcpRelayPol" owner="tenant" />
        </l3extLIfP>
      </l3extLNodeP>
    </l3extOut>
  </fvTenant>
<polUni>

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

Configuring a DNS Service Policy

A DNS policy is required to connect to external servers, for example AAA, RADIUS, vCenter, and services by hostname. A DNS service policy is a shared policy, so any tenant and VRF that uses this service must be configured with the specific DNS profile label. To configure a DNS policy for the ACI fabric, you must complete the following tasks:

- Ensure that the management EPG is configured for the DNS policy, otherwise this policy will not take into effect on the switches.

Note For the management EPG, only the default DNS policy is supported.

- Create a DNS profile (default) that contains the information about DNS providers and DNS domains.
- Associate the DNS profile (default or another DNS profile) name to a DNS label under the required tenant.

It is possible to configure a per-tenant, per-VRF DNS profile configuration. Additional DNS profiles can be created and applied to specific VRFs of specific tenants using the appropriate DNS label. For example, if you create a DNS profile with a name of acme, you can add a DNS label of acme to the appropriate **Networking** > **VRF** policy configuration in the tenants configuration.

Configuring External Destinations with an In-Band DNS Service Policy

Configure the external destinations for the services as follows:

Source	In-Band Management	Out-of-Band Management	External Server Location
APIC	IP address or Fully Qualified domain name (FQDN)	IP address or FQDN	Anywhere

Source	In-Band Management	Out-of-Band Management	External Server Location
Leaf switches	IP address	IP address or FQDN **Note** The DNS policy must specify the out-of-band management EPG for reachability of the DNS server.	Anywhere
Spine switches	IP address	IP address or FQDN **Note** The DNS policy must specify the out-of-band management EPG for reachability of the DNS server.	Directly connected to a leaf switch

The following is a list of external servers:

- Call Home SMTP server
- Syslog server
- SNMP Trap destination
- Statistics Export destination
- Configuration Export destination
- Techsupport Export destination
- Core Export destination

The recommended guidelines are as follows:

- The external servers must be atatched to the leaf access ports.
- Use in-band connectivity for the leaf switches to avoid extra cabling for the management port.
- Use out-of-band management connectivity for the spine switches. Connect this out-of-band network for spine switches to one of the leaf ports with in-band management virtual routing and forwarding (VRF) so that the spine switches and the leaf switches can reach the same set of external servers.
- Use IP addresses for the external servers.

Dual Stack IPv4 and IPv6 DNS Servers

DNS servers have primary DNS records which can be A records (IPV4) or AAAA records (IPV6). Both A and AAAA records associate domain name with a specific IP address (IPv4 or IPv6).

The ACI fabric can be configured to use reputable public DNS servers that run on IPv4. These servers are able to resolve and respond with A record (IPv4) or AAAA record (IPv6).

In a pure IPv6 environment, the system administrators must use IPv6 DNS servers. The IPv6 DNS servers are enabled by adding them to /etc/resolv.conf.

A more common environment is to have dual-stack IPv4 and IPv6 DNS servers. In the dual-stack case, both IPv4 and IPv6 name servers are listed in /etc/resolv.conf. However, in a dual-stack environment, simply appending the IPv6 DNS servers to the list may cause a large delay in DNS resolutions. This is because the IPv6 protocol takes precedence by default, and it is unable to connect to the IPv4 DNS servers (if they are listed first in /etc/resolv.conf). The solution is to list IPv6 DNS servers ahead of IPv4 DNS servers. Also add “options single-request-reopen” to enable the same socket to be used for both IPv4 and IPv6 lookups.

Here is an example of resolv.conf in dual-stack IPv4 and IPv6 DNS servers where the IPv6 DNS servers are listed first. Also note the “single-request-reopen” option:

```
options single-request-reopen
nameserver 2001:4860:4680::8888
nameserver 2001:4860:4680::8844
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Dual-Stack IPv4 and IPv6 Environment

If the management network in the ACI fabric supports both IPv4 and IPv6, the Linux system application (glibc) will use the IPv6 network by default because getaddrinfo() will return IPv6 first.

Under certain conditions however, an IPv4 address may be preferred over an IPv6 address. The Linux IPv6 stack has a feature which allows an IPv4 address mapped as an IPv6 address using IPv6 mapped IPv4 address (::ffff/96). This allows an IPv6 capable application to use only a single socket to accept or connect both IPv4 and IPv6. This is controlled by the glibc IPv6 selection preference for getaddrinfo() in /etc/gai.conf.

In order to allow glibc to return multiple addresses when using /etc/hosts, “multi on” should be added to the /etc/hosts file. Otherwise, it may return only the first match.

If an application is not aware whether both IPv4 and IPv6 exist, it may not perform fallback attempts using different address families. Such applications may require a fallback implementation.

Policy for Priority of IPv4 or IPv6 in a DNS Profile

The DNS profile supports version preference choices between IPv4 and IPv6. Using the user interface, you can enable your preference. IPv4 is the default.

The following is an example of a policy based configuration using Postman REST API:

```
<?xml version="1.0" encoding="UTF-8”?>
<!— api/node/mo/uni/fabric/dnsp-default.xml —>
<dnsProfile dn="uni/fabric/dnsp-default" IPVerPreference="IPv6" childAction="" descr="" >
</dnsProfile>
```

The gai.conf settings control destination address selection. The file has a label table, precedence table, and an IPv4 scopes table. The changes for prioritizing IPv4 or IPv6 over the other need to go into the precedence

table entries. Given below are sample contents of the standard file as it is used in Linux systems for many flavors. A single line of precedence label in the file overrides any default settings.

The following is an example of a gai.conf to prioritize IPv4 over IPv6:

```
# Generated by APIC
label ::1/128       0
label ::/0          1
label 2002::/16     2
label ::/96         3
label ::ffff:0:0/96 4
precedence  ::1/128       50
precedence ::/0           40
precedence 2002::/16      30
precedence ::/96          20
# For APICs prefering IPv4 connections, change the value to 100.
precedence ::ffff:0:0/96  10
```

Configuring a DNS Service Policy to Connect with DNS Providers Using the GUI

Before you begin

Make sure that Layer 2 or Layer 3 management connectivity is configured.

Procedure

Step 1 On the menu bar, choose **FABRIC** > **Fabric Policies**. In the **Navigation** pane, expand **Global Policies** > **DNS Profiles**, and click the default DNS profile.

Step 2 In the **Work** pane, in the **Management EPG** field, from the drop-down list, choose the appropriate management EPG (default (Out-of-Band)).

Step 3 Expand **DNS Providers**, and perform the following actions:

a) In the **Address** field, enter the provider address.

b) In the **Preferred** column, check the check box if you want to have this address as the preferred provider.

You can have only one preferred provider.

c) Click **Update**.

d) (Optional) To add a secondary DNS provider, expand **DNS Providers**, and in the **Address** field, type the provider address. Click **Update**.

Step 4 Expand **DNS Domains**, and perform the following actions:

a) In the **Name** field, enter the domain name (cisco.com).

b) In the **Default** column, check the check box to make this domain the default domain.

You can have only one domain name as the default.

c) Click **Update**.

d) (Optional) To add a secondary DNS domain, expand **DNS Domains**. In the **Address** field, enter the secondary domain name. Click **Update**.

Step 5 Click **Submit**.
The DNS server is configured.

Step 6 On the menu bar, click **TENANTS** > **mgmt**.

Step 7 In the **Navigation** pane, expand **Networking** > **VRF** > **oob**, and click **oob**.

Step 8 In the **Work** pane, under **Properties**, in the **DNS labels** field, enter the appropriate DNS label (default). Click **Submit**.
The DNS profile label is now configured on the tenant and VRF.

Configuring a DNS Service Policy to Connect with DNS Providers Using the NX-OS Style CLI

Procedure

Step 1 In the NX-OS CLI, get into configuration mode, shown as follows:

Example:

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

Step 2 Configure a DNS server policy.

Example:

```
apic1(config)# dns
apic1(config-dns)# address 172.21.157.5 preferred
apic1(config-dns)# address 172.21.157.6
apic1(config-dns)# domain company.local default
apic1(config-dns)# use-vrf oob-default
```

Step 3 Configure a DNS profile label on any VRF where you want to use the DNS profile.

Example:

```
apic1(config)# tenant mgmt
apic1(config-tenant)# vrf context oob
apic1(config-tenant-vrf)# dns label default
```

Configuring a DNS Service Policy to Connect with DNS Providers Using the REST API

Before you begin

Make sure that Layer 2 or Layer 3 management connectivity is configured.

Procedure

Step 1 Configure the DNS service policy.

Example:

```
POST URL :
https://apic-IP-address/api/node/mo/uni/fabric.xml

<dnsProfile name="default">

    <dnsProv addr="172.21.157.5" preferred="yes"/>
    <dnsProv addr="172.21.157.6"/>

    <dnsDomain name="cisco.com" isDefault="yes"/>

    <dnsRsProfileToEpg tDn="uni/tn-mgmt/mgmtp-default/oob-default"/>

</dnsProfile>
```

Step 2 Configure the DNS label under the out-of-band management tenant.

Example:

```
POST URL: https://apic-IP-address/api/node/mo/uni/tn-mgmt/ctx-oob.xml
<dnsLbl name="default" tag="yellow-green"/>
```

Verifying that the DNS Profile is Configured and Applied to the Fabric Controller Switches Using the NX-OS Style CLI

Procedure

Step 1 Verify the configuration for the default DNS profile.

Example:

```
apic1# show running-config dns

# Command: show running-config dns
# Time: Sat Oct  3 00:23:52 2015
  dns
    address 172.21.157.5 preferred
    address 172.21.157.6
    domain company.local default
    use-vrf oob-default
    exit
```

Step 2 Verify the configurations for the DNS labels.

Example:

```
apic1# show running-config tenant mgmt vrf context oob

# Command: show running-config tenant mgmt vrf context oob
# Time: Sat Oct  3 00:24:36 2015
  tenant mgmt
    vrf context oob
      dns label default
      exit
    exit
```

Step 3 Verify that the applied configuration is operating on the fabric controllers.

Example:

```
apic1# cat /etc/resolv.conf
# Generated by IFC

nameserver 172.21.157.5
nameserver 172.21.157.6
```

Configuring Custom Certificates

Configuring Custom Certificate Guidelines

- Wildcard certificates (such as *.cisco.com, which is used across multiple devices) and its associated private key generated elsewhere are not supported on the APIC as there is no support to input the private key or password in the APIC. Also, exporting private keys for any certificates, including wildcard certificates, is not supported.
- You must download and install the public intermediate and root CA certificates before generating a Certificate Signing Request (CSR). Although a root CA Certificate is not technically required to generate a CSR, Cisco requires the root CA certificate before generating the CSR to prevent mismatches between the intended CA authority and the actual one used to sign the CSR. The APIC verifies that the certificate submitted is signed by the configured CA.
- To use the same public and private keys for a renewed certificate generation, you must satisfy the following guidelines:
  - You must preserve the originating CSR as it contains the public key that pairs with the private key in the key ring.
  - The same CSR used for the originating certificate must be resubmitted for the renewed certificate if you want to re-use the public and private keys on the APIC.
  - Do not delete the original key ring when using the same public and private keys for the renewed certificate. Deleting the key ring will automatically delete the associated private key used with CSRs.

Configuring a Custom Certificate for Cisco ACI HTTPS Access Using the GUI

CAUTION: PERFORM THIS TASK ONLY DURING A MAINTENANCE WINDOW AS THERE IS A POTENTIAL FOR DOWNTIME. The downtime affects access to the APIC cluster and switches from external users or systems and not the APIC to switch connectivity. The NGINX process on the switches will also be impacted but that will be only for external connectivity and not for the fabric data plane. Access to the APIC, configuration, management, troubleshooting and such will be impacted. Expect a restart of all web servers in the fabric during this operation.

Before you begin

Determine from which authority you will obtain the trusted certification so that you can create the appropriate Certificate Authority.

Procedure

Step 1 On the menu bar, choose **Admin** > **AAA**.

Step 2 In the **Navigation** pane, choose **Security**.

Step 3 In the **Work** pane, choose **Public Key Management** > **Certificate Authorities** > **Create Certificate Authority**.

Step 4 In the **Create Certificate Authority** dialog box, in the **Name** field, enter a name for the certificate authority.

Step 5 In the **Certificate Chain** field, copy the intermediate and root certificates for the certificate authority that will sign the Certificate Signing Request (CSR) for the Application Policy Infrastructure Controller (APIC).

The certificate should be in Base64 encoded X.509 (CER) format. The intermediate certificate is placed before the root CA certificate. It should look similar to the following example:

```
-----BEGIN CERTIFICATE-----
<Intermediate Certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Root CA Certificate>
-----END CERTIFICATE-----
```

Step 6 Click **Submit**.

Step 7 In the **Navigation** pane, choose **Public Key Management** > **Key Rings**.

Step 8 In the **Work** pane, choose **Actions** > **Create Key Ring**.

Step 9 In the **Create Key Ring** dialog box, in the **Name** field, enter a name.

Step 10 In the **Certificate** field, do not add any content.

Step 11 In the **Modulus** field, click the radio button for the desired key strength.

Step 12 In the **Certificate Authority** field, from the drop-down list, choose the certificate authority that you created earlier. Click **Submit**.

Note Do not delete the key ring. Deleting the key ring will automatically delete the associated private key used with CSRs.

In the **Work** pane, in the **Key Rings** area, the **Admin State** for the key ring created displays **Started**.

Step 13 In the **Navigation** pane, choose **Public Key Management** > **Key Rings** > ***key_ring_name***.

Step 14 In the **Work** pane, choose **Actions** > **Create Certificate Request**.

Step 15 In the **Subject** field, enter the fully qualified domain name (FQDN) of the APIC.

Step 16 Fill in the remaining fields as appropriate.

Note Check the online help information available in the **Create Certificate Request** dialog box for a description of the available parameters.

Step 17 Click **Submit**.
The object is created and displayed in the **Navigation** pane under the key ring you created earlier. In the **Navigation** pane, click the object and in the **Work** pane, in the **Properties** area, in the **Request** field the CSR is displayed. Copy the contents from the field to submit to the **Certificate Authority**.

Step 18 In the **Navigation** pane, choose **Public Key Management** > **Key Rings** > ***key_ring_name***.

Step 19 In the **Work** pane, in the **Certificate** field, paste the signed certificate that you received from the certificate authority.

Step 20 Click **Submit**.

Note If the CSR was not signed by the Certificate Authority indicated in the key ring, or if the certificate has MS-DOS line endings, an error message is displayed and the certificate is not accepted. Remove the MS-DOS line endings.

The key is verified, and in the **Work** pane, the **Admin State** changes to **Completed** and is now ready for use in the HTTP policy.

Step 21 On the menu bar, choose **Fabric** > **Fabric Policies**.

Step 22 In the Navigation pane, choose **Pod Policies** > **Policies** > **Management Access** > **default**.

Step 23 In the **Work** pane, in the **Admin Key Ring** drop-down list, choose the desired key ring.

Step 24 (Optional) For Client-based authentication, in the **Client Certificate TP** drop-down list, choose the previously created Local User policy and click **Enabled** for **Client Certificate Authentication state**.

Step 25 Click **Submit**.
All web servers restart. The certificate is activated, and the non-default key ring is associated with HTTPS access.

What to do next

You must remain aware of the expiration date of the certificate and take action before it expires. To preserve the same key pair for the renewed certificate, you must preserve the CSR as it contains the public key that pairs with the private key in the key ring. Before the certificate expires, the same CSR must be resubmitted. Do not delete or create a new key ring as deleting the key ring will delete the private key stored internally on the APIC.

Provisioning Fabric Wide System Settings

Configuring APIC In-Band or Out-of-Band Connectivity Preferences

This topic describes how to toggle between in-band and out-of-band connectivity on the APIC server for management access to devices such as authentication servers or SNMP servers external to the ACI fabric. Enabling **inband** executes in-band management connectivity between the APIC server to external devices through leaf switches on the ACI fabric. Enabling **ooband** executes out-of-band management connectivity between the APIC server to external devices through connections external to the ACI fabric.

Before you begin

Configure in-band and out-of-band management networks. For more information, see *Management* in the *Cisco APIC Basic Configuration Guide, Release 3.x*.

Procedure

Step 1 On the menu bar, click **System** > **System Settings**.
Step 2 On the Navigation bar, click **APIC Connectivity Preferences**.
Step 3 To enable the policy, click **inband** or **ooband**.
Step 4 Click **Submit**.

Configure Quota Management Policies

Starting in the Cisco Application Policy Infrastructure Controller (APIC) Release 2.3(1), there are limits on number of objects a tenant admin can configure. This enables the admin to limit the number of managed objects that can be added globally across tenants.

This feature is useful when you want to limit any tenant or group of tenants from exceeding ACI maximums per leaf or per fabric or unfairly consuming a majority of available resources, potentially affecting other tenants on the same fabric.

Procedure

Step 1 On the menu bar, click **System** > **System Settings**.
Step 2 Right-click **Quota** and choose **Create Quota Configuration**..
Step 3 In the **Class** field, choose the object type to limit with the quota.
Step 4 In the **Container Dn** field, enter the distinguished name (DN) that describes the class.
Step 5 In the **Exceed Action** field, choose either **Fail Transaction Action** or **Raise Fault Action**.
Step 6 In the **Max Number** field, enter the maximum number of the managed objects that can be created after which the exceed action will be applied.
Step 7 Click **Submit**.

Create an Enforced BD Exception List

This topic describes how to create a global exception list of subnets which are not subject to an enforced bridge domain. With the Enforced BD feature configured, the endpoints in a subject endpoint group (EPG) can only ping subnet gateways within the associated bridge domain.

The exception IP addresses can ping all of the BD gateways across all of your VRFs.

A loopback interface configured for an L3Out does not enforce reachability to the IP address that is configured for the subject loopback interface.

When an eBGP peer IP address exists in a different subnet than the subnet of the L3Out interface, the peer subnet must be added to the allowed exception subnets. Otherwise, eBGP traffic is blocked because the source IP address exists in a different subnet than the L3Out interface subnet.

Before you begin

Create an enforced bridge domain (BD).

Procedure

Step 1 On the menu bar, click **System** > **System Settings**.
Step 2 Click **BD Enforced Exception List**.
Step 3 Click the + on **Exception List**.
Step 4 Add the IP address and network mask for the subnet that can ping any subnet gateway.
Step 5 Repeat to add more subnets that are exceptions to the enforced bridge domain.
Step 6 Click **Submit**.

Create a BGP Route Reflector Policy and Route Reflector Node Endpoints

This topic describes how to create ACI fabric route reflectors, which use multiprotocol BGP (MP-BGP) to distribute external routes within the fabric. To enable route reflectors in the ACI fabric, the fabric administrator must select the spine switches that will be the route reflectors, and provide the autonomous system (AS) number. Once route reflectors are enabled in the ACI fabric, administrators can configure connectivity to external networks.

Before you begin

Required:

- To connect external routers to the ACI fabric, the fabric infrastructure administrator must configure spine nodes as Border Gateway Protocol (BGP) route reflectors.
- For redundancy purposes, more than one spine is configured as a router reflector node (one primary and one secondary reflector).

Procedure

Step 1 To create a BGP Route Reflector policy, perform the following steps:
a) On the menu bar, click **System** > **System Settings**.
b) Click **BGP Route Reflector**.
c) Enter the Autonomous System Number.
d) Click the + on **Route Reflector Nodes**.
e) Enter the spine route reflector node ID endpoint, and click **Submit**.,

Step 2 To create external route reflector node endpoints, perform the following steps:
a) Click the + on **External Route Reflector Nodes**.
b) Choose the spine to serve as external route reflector node endpoint.
c) If this is a site managed by Multi-Site, you can also specify an intersite spine route reflector.

d) Click **Submit**.

Configure a Fabric Wide Control Plane MTU Policy

This topic describes how to create a fabric-wide Control Plane (CP) MTU policy, that sets the global MTU size for control plane packets sent by the nodes (APIC and the switches) in the fabric.

In a multipod topology, the MTU setting for the fabric external ports must be greater than or equal to the CP MTU value set. Otherwise, the fabric external ports might drop the CP MTU packets.

Note If you set the L3Out Interface Profile to inherit the MTU from the IPN, it will be 9150. If you want the MTU to be used across the IPN to be 9216, you must explicitly configure it in the L3Out Interface Profile (at **Tenants** > ***tenant-name*** > **Networking** > **External Routed Networks** > **Create Routed Outside** > **Nodes and Interface Protocol Profiles** > **Create Node Profile** > **Create Interface Profile**).

If you change the IPN or CP MTU, Cisco recommends changing the CP MTU value first, then changing the MTU value on the spine of the remote pod. This reduces the risk of losing connectivity between the pods due to MTU mismatch.

Procedure

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

Step 2 Click **Control Plane MTU**.

Step 3 Enter the MTU for fabric ports.

Step 4 Click **Submit**.

Create a COOP Group Policy

This topic describes how to create a Council of Oracle Protocol (COOP) Group Policy, which is used to communicate the mapping information (location and identity) to the spine proxy. A leaf switch forwards endpoint address information to the spine switch 'Oracle' using Zero Message Queue (ZMQ). COOP running on the spine nodes will ensure all spine nodes maintain a consistent copy of endpoint address and location information and additionally maintain the distributed hash table (DHT) repository of endpoint identity to location mapping database.

Procedure

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

Step 2 Click **COOP Group**.

Step 3 Choose the policy property type. The type can be **Compatible Type** or **Strict Type**.

The Oracle Nodes are the spines in the fabric, automatically populated by the system.

Step 4 Click **Submit**

Configure Endpoint Loop Protection

The endpoint loop protection policy specifies how loops detected by frequent MAC moves are handled. To configure EP loop protection perform the following steps:

Procedure

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

Step 2 Click **Endpoint Controls**.

Step 3 Click the **Ep Loop Protection** tab.

Step 4 To enable the policy, click **Enabled** in the **Administrative State** field.

Step 5 Optional. Set the loop detection interval, which specifies the time to detect a loop. The interval range is from 30 to 300 seconds. The default setting is 60 seconds.

Step 6 Set the loop detection multiplication factor, which is the number of times a single EP moves between ports within the loop detection interval. The range is from 1 to 255. The default is 4.

Step 7 Choose the action to take when detecting a loop.

The action can be:

- **BD Learn Disable**
- **Port Disable**

The default is **Port Disable**.

Step 8 Click **Submit**.

About the Rogue Endpoint Control Policy

A rogue endpoint attacks top of rack (ToR) switches through frequently, repeatedly injecting packets on different ToR ports and changing 802.1Q tags (thus, emulating endpoint moves) causing learned class and EPG port changes. Misconfigurations can also cause frequent IP and MAC address changes (moves).

Such rapid movement in the fabric causes significant network instability, high CPU usage, and in rare instances, endpoint mapper (EPM) and EPM client (EPMC) crashes due to significant and prolonged messaging and transaction service (MTS) buffer consumption. Also, such frequent moves may result in the EPM and EPMC logs rolling over very quickly, hampering debugging for unrelated endpoints.

The rogue endpoint control feature addresses this vulnerability by quickly:

- Identifying such rapidly moving MAC and IP endpoints
- Stopping the movement by temporarily making endpoints static (thus, quarantining the endpoint)
- Keeping the endpoint static for the **Rogue EP Detection Interval** and dropping the traffic to and from the rogue endpoint. After this time expires, deleting the unauthorized MAC or IP address

- Generating a host tracking packet to enable the system to re-learn the impacted MAC or IP address
- Raising a fault, to enable corrective action

The rogue endpoint control policy is configured globally and, unlike other loop prevention methods, functions at the level of individual endpoints (IP and MAC addresses). It does not distinguish between local or remote moves; any type of interface change is considered a move in determining if an endpoint should be quarantined.

The rogue endpoint control feature is disabled by default.

Limitations of the Rogue Endpoint Control Policy

The following limitations apply when using a rogue endpoint control policy:

- Changing rogue endpoint control policy parameters will not affect existing rogue endpoints.
- If a rogue endpoint is enabled, loop detection and bridge domain move frequency will not take effect.
- Disabling the rogue endpoint feature clears all rogue endpoints.
- You must disable the rogue endpoint feature prior to upgrading or downgrading the Cisco Application Policy Infrastructure Controller (Cisco APIC).
- The endpoint mapper (EPM) has value limits for rogue endpoint parameters. If you set the parameter values outside of this range, the Cisco APIC raises a fault for each mismatched parameter.
- The rogue endpoint feature is not supported on remote leaf switches or Cisco ACI Multi-Site.

Configure the Rogue Endpoint Control Policy Using the GUI

You can configure the **Rogue EP Control** policy for the fabric, to detect and delete unauthorized endpoints, using the Cisco Application Policy Infrastructure Controller (Cisco APIC) GUI. This topic also includes the steps to clear rogue endpoints on a TOR switch, ad-hoc.

The policy options have the following valid and supported values:

- **Rogue EP Detection Interval**—Sets the rogue endpoint detection interval, which specifies the time to detect rogue endpoints. Valid values are from 0 to 65535 seconds. The default is 60.
- **Hold Interval (sec)**—Interval in seconds after the endpoint is declared rogue, where it is kept static so learning is prevented and the traffic to and from the Rogue endpoint is dropped. After this interval, the endpoint is deleted. Valid values are from 1800 to 3600. The default is 1800.
- **Rogue EP Detection Multiplication Factor**—Sets the rogue endpoint detection multiplication factor for determining if an endpoint is unauthorized. If the endpoint moves more times than this number, within the EP detection interval, the endpoint is declared rogue. Valid values are from 2 to 10. The default is 6.

Procedure

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

Step 2 On the navigation bar, click **Endpoint Controls** and click the **Rogue EP Control** tab.

Step 3 Set the **Administrative State** to **Enabled**.

Step 4 Optional. Reset the **Rogue EP Detection Interval (sec)**, **Rogue EP Detection Multiplication Factor**, or the **Hold Interval (sec)**.

Step 5 (Optional) To clear rogue endpoints on a TOR switch, perform the following steps:

a) On the Cisco APIC menu bar, click **Fabric** > **Inventory**.
b) On the Navigation bar, expand the Pod and click the leaf switch where you want to clear rogue endpoints.
c) When the leaf switch summary appears in the work pane, right-click the leaf switch name in the Navigation bar, and choose **Clear Rogue Endpoints**.
d) Click **Yes**.

Configure Rogue Endpoint Control Using the NX-OS Style CLI

You can configure the **Rogue EP Control** policy for the fabric, to detect and delete unauthorized endpoints, using the NX-OS style CLI.

Procedure

Step 1 **configure**

Enters global configuration mode.

Example:

```
apic1# configure
```

Step 2 **endpoint rogue-detect enable**

Enables the global Rogue Endpoint Control policy.

Example:

```
apic1(config)# endpoint rogue-detect enable
```

Step 3 **endpoint rogue-detect hold-interval** *hold_interval*

Sets the hold interval in seconds after the endpoint is declared rogue, where it is kept static so learning is prevented, and the traffic to and from the rogue endpoint is dropped. After this interval, the endpoint is deleted. Valid values are from 1800 to 3600 seconds. The default is 1800.

Example:

```
apic1(config)# endpoint rogue-detect hold-interval 1800
```

Step 4 **endpoint rogue-detect interval** *interval*

Sets the rogue detection interval in seconds, which specifies the time to detect rogue endpoints. Valid values are from 0 to 65535 seconds. The default is 60.

Example:

```
apic1(config)# endpoint rogue-detect interval 60
```

Step 5 **endpoint rogue-detect factor** *factor*

Specifies the multiplication factor for determining if an endpoint is unauthorized. If the endpoint moves more times during the interval, the EP is declared rogue. Valid values are from 2 to 10. The default is 6.

Example:

```
apic1# endpoint rogue-detect factor 6
```

Step 6 This example configures a Rogue Endpoint Control policy.

Example:

```
apic1# cconfigure
apic1(config)# endpoint rogue-detect enable
apic1(config)# endpoint rogue-detect hold-interval 1800
apic1(config)# endpoint rogue-detect interval 60
apic1(config)# endpoint rogue-detect factor 6
```

Configure the Rogue Endpoint Control Policy Using the REST API

You can configure the **Rogue EP Control** policy for the fabric, to detect and delete unauthorized endpoints, using the REST API.

Procedure

To configure the Rogue EP Control policy, send a post with XML similar to the following:

Example:

```
<polUni>
    <infraInfra>
        <epControlP name="default" adminSt="enabled"  holdIntvl="1800"
rogueEpDetectIntvl="60" rogueEpDetectMult="6"/>
    </infraInfra>
</polUni>
```

Configure IP Aging

This topic describes how to enable an IP Aging policy. When enabled, the IP aging policy ages unused IPs on an endpoint.

When the Administrative State is enabled, the IP aging policy sends ARP requests (for IPv4) and neighbor solicitations (for IPv6) to track IPs on endpoints. If no response is given, the policy ages the unused IPs.

Procedure

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

Step 2 Click **Endpoint Controls**.

Step 3 Click the **Ip Aging** tab.

Step 4 To enable the policy, click **Enabled** in the **Administrative State** field.

What to do next

Create an End Point Retention policy, which is required, to specify the timer used for tracking IPs on endpoints. Navigate to **Tenants** > *tenant-name* > > **Policies** > **Protocol** > **End Point Retention**.

Disable Remote Endpoint Learning

This topic describes how to enable or disable IP end point learning.

The scope of this policy is fabric-wide. After configuration, a policy is pushed to each leaf switch as it comes up.

You should enable this policy in fabrics which include the Cisco Nexus 9000 series switches, 93128 TX, 9396 PX, or 9396 TX switches with the N9K-M12PQ uplink module, after all the nodes have been successfully upgraded to APIC Release 2.2(2x) or higher.

After any of the following configuration changes, you may need to manually flush previously learned IP endpoints:

- Remote IP endpoint learning is disabled
- The VRF is configured for ingress policy enforcement
- At least one Layer 3 interface exists in the VRF

To manually flush previously learned IP endpoints, enter the following command on both VPC peers: `vsh -c "clear system internal epm endpoint vrf <vrf-name> remote"`

To enable or disable IP end point learning, perform the following steps:

Procedure

Step 1 On the menu bar, click **System** > **System Settings**.
Step 2 Click **Fabric Wide Setting**.
Step 3 Click the check box on **Disable Remote EP Learn**.
Step 4 Click **Submit**.

Globally Enforce Subnet Checks

This topic describes how to enable or disable subnet checking. When enabled, IP address learning is disabled outside of subnets configured in a VRF, for all other VRFs.

The scope of this policy is fabric-wide. After configuration, a policy is pushed to each leaf switch as it comes up.

Procedure

Step 1 On the menu bar, click **System** > **System Settings**.
Step 2 Click **Fabric Wide Setting**.
Step 3 Click the check box on **Enforce Subnet Check**.
Step 4 Click **Submit**.

Reallocate a GIPo

This topic describes how to enable reallocating GIPos on non-stretched BDs to make room for stretched BDs.

The scope of this policy is fabric-wide. After configuration, a policy is pushed to each leaf switch as it comes up.

Procedure

Step 1 On the menu bar, click **System** > **System Settings**.
Step 2 Click **Fabric Wide Setting**.
Step 3 Click the check box on **Reallocate Gipo**.
Step 4 Click **Submit**.

Globally Enforce Domain Validation

This topic describes how to enforce domain validation. When enabled, a validation check is performed when a static path is added, to determine if no domain is associated with an EPG.

The scope of this policy is fabric-wide. After configuration, a policy is pushed to each leaf switch as it comes up.

Procedure

Step 1 On the menu bar, click **System** > **System Settings**.
Step 2 Click **Fabric Wide Setting**.
Step 3 Click the check box on **Enforce Domain Validation**.
Step 4 Click **Submit**.

Enable OpFlex Client Authentication

This topic describes how to enable OpFlex client authentication for GOLF and Linux.

To deploy GOLF or Linux Opflex clients in an environment where the identity of the client cannot be guaranteed by the network, you can dynamically validate the client's identity based on a client certificate.

Note When you enable certificate enforcement, connectivity with any GOLF or Linux Opflex client that does not support client authentication is disabled.

The scope of this policy is fabric-wide. After configuration, a policy is pushed to each leaf switch as it comes up.

Procedure

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

Step 2 Click **Fabric Wide Setting**.

Step 3 Click the check box on **OpFlex Client Authentication** to enable or disable enforcing client certificate authentication for GOLF and Linux Opflex clients.

Step 4 Click **Submit**.

Create a Load Balancer Policy

This topic describes how to configure the default Load Balancer policy.

The load balancing policy options balance traffic among the available uplink ports. Static hash load balancing is the traditional load balancing mechanism used in networks where each flow is allocated to an uplink based on a hash of its 5-tuple. This load balancing gives a distribution of flows across the available links that is roughly even. Usually, with a large number of flows, the even distribution of flows results in an even distribution of bandwidth as well. However, if a few flows are much larger than the rest, static load balancing might give suboptimal results.

Procedure

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

Step 2 Click **Load Balancer**.

Step 3 Choose the **Dynamic Load Balancing Mode**.

The dynamic load balancer (DLB) mode adjusts the traffic allocations according to congestion levels. It measures the congestion across the available paths and places the flows on the least congested paths, which results in an optimal or near optimal placement of the data. DLB can be configured to place traffic on the available uplinks using the granularity of flows or of flowlets. Flowlets are bursts of packets from a flow that are separated by intervals. The mode can be **Aggressive**, **Conservative**, or **Off** (the default).

Step 4 Enable or disable **Dynamic Packet Prioritization** by choosing **On** or **Off** (the default.

Dynamic Packet Prioritization (DPP) prioritizes short flows higher than long flows; a short flow is less than approximately 15 packets. Short flows are more sensitive to latency than long ones. DPP can improve overall application performance.

Step 5 Choose the Load Balancing Mode. The mode can be **Link Failure** or **Traditional** (the default).

The load balancer administrative state. In all modes of load balancing, static or dynamic, the traffic is sent only on those uplinks or paths that meet the criteria for equal cost multipath (ECMP); these paths are equal and the lowest cost from a routing perspective.

Step 6 Click **Submit**.

Enable a Time Precision Policy

This topic describes how to enable Precision Time Protocol (PTP), a time synchronization protocol for nodes distributed across a network. Its hardware timestamp feature provides greater accuracy than other time synchronization protocols such as Network Time Protocol (NTP).

PTP is a distributed protocol that specifies how real-time PTP clocks in the system synchronize with each other. These clocks are organized into a master-member synchronization hierarchy with the grandmaster clock, the clock at the top of the hierarchy, determining the reference time for the entire system. Synchronization is achieved by exchanging PTP timing messages, with the members using the timing information to adjust their clocks to the time of their master in the hierarchy. PTP operates within a logical scope called a PTP domain.

Procedure

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

Step 2 Click **Precision Time Protocol**.

Step 3 Choose **Enabled** or **Disabled**.

If you choose disable PTP, NTP time is used to sync the fabric. If you enable PTP, a spine is automatically chosen as a master to which the entire site gets synced.

Step 4 Click **Submit**.

Enable a Global System GIPo Policy

This topic describes how to use the infra tenant GIPo as the system GIPo.

An ACI multipod deployment requires the 239.255.255.240 system Global IP Outside (GIPo) to be configured on the inter-pod network (IPN) as a PIM BIDIR range. This 239.255.255.240 PIM BIDIR range configuration on the IPN devices can be avoided by using the infra GIPo as System GIPo.

Before you begin

Upgrade all of the switches in the ACI fabric, including the leaf switches and spine switches, to the latest APIC release.

Procedure

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

Step 2 Choose **Enabled** or **Disabled** (the default) on **Use Infra GIPo as System GIPo**

Step 3 Click **Submit**.

Provisioning Global Fabric Access Policies

Create a Global Attachable Access Entity Profile

An Attachable Entity Profile (AEP) represents a group of external entities with similar infrastructure policy requirements. The infrastructure policies consist of physical interface policies that configure various protocol options, such as Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol (LLDP), or Link Aggregation Control Protocol (LACP).

An AEP is required to deploy VLAN pools on leaf switches. Encapsulation blocks (and associated VLANs) are reusable across leaf switches. An AEP implicitly provides the scope of the VLAN pool to the physical infrastructure.

The following AEP requirements and dependencies must be accounted for in various configuration scenarios, including network connectivity, VMM domains, and multipod configuration:

- The AEP defines the range of allowed VLANS but it does not provision them. No traffic flows unless an EPG is deployed on the port. Without defining a VLAN pool in an AEP, a VLAN is not enabled on the leaf port even if an EPG is provisioned.
- A particular VLAN is provisioned or enabled on the leaf port that is based on EPG events either statically binding on a leaf port or based on VM events from external controllers such as VMware vCenter or Microsoft Azure Service Center Virtual Machine Manager (SCVMM).
- Attached entity profiles can be associated directly with application EPGs, which deploy the associated application EPGs to all those ports associated with the attached entity profile. The AEP has a configurable generic function (infraGeneric), which contains a relation to an EPG (infraRsFuncToEpg) that is deployed on all interfaces that are part of the selectors that are associated with the attachable entity profile.

A virtual machine manager (VMM) domain automatically derives physical interface policies from the interface policy groups of an AEP.

Before you begin

Create the tenant, VRF, application profiles, and EPGs to associate to the attached entity profile.

Procedure

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

Step 2 On the navigation bar, expand **Policies** and **Global**.

Step 3 Right-click **Attachable Access Entity Profile** and choose **Create Attachable Access Entity Profile**.

Step 4 Enter a name for the policy.

Step 5 Click the + icon on **Domains** table.

Step 6 Enter a physical domain, a previously created physical, Layer 2, Layer 3, or Fibre Channel domain, or create one.

Step 7 Enter the encapsulation for the domain and click **Update**.

Step 8 Click the + icon on the **EPG DEPLOYMENT** table.

Step 9 Enter the tenant, application profile, EPG, encapsulation (such as vlan-1), primary encapsulation (primary encapsulation number) and interface mode (trunk, Access (802.1P, or Access (Untagged).

Step 10 Click **Update**.

Step 11 Click **Next**.

Step 12 Choose the interfaces to associate to the attachable entity profile.

Step 13 Click **Finish**.

Configure the Global QoS Class Policy

The global QoS Class policy can be used to:

- Preserve the CoS priority level, to guarantee that the CoS value in 802.1P packets which enter and transit the ACI fabric is preserved. 802.1P CoS preservation is supported in single pod and multipod topologies. In multipod topologies, CoS Preservation can be used where you want to preserve the QoS priority settings of 802.1P traffic entering POD 1 and egressing out of POD 2, but you are not concerned with preserving the CoS/DSCP settings in interpod network (IPN) traffic between the pods. To preserve CoS/DSCP settings when multipod traffic is transitting an IPN, use a DSCP policy (configured at **Tenants** > **infra** > > **Policies** > **Protocol** > **DSCP class-cos translation policy for L3 traffic**)
- Reset the properties for the default QoS class levels, such as the **MTU**, **Queue Limit**, or **Scheduling Algorithm**.

Procedure

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

Step 2 On the navigation bar, expand **Policies** and **Global**.

Step 3 Click **QOS Class**.

Step 4 To enable 802.1P CoS preservation, click the **Preserve COS** check box.

Step 5 To change the default settings for a QoS class, double-click on it. Enter the new settings and click **Submit**.

Create a Global DHCP Relay Policy

The global DHCP Relay policy identifies the DHCP Server for the fabric.

Procedure

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

Step 2 On the navigation bar, expand **Policies** and **Global**.

Step 3 Right-click **DHCP Relay** and choose **Create DHCP Relay Policy**.

Step 4 Enter a name for the policy.

Step 5 Click the + icon on **Providers**.

Step 6 Choose the EPG type, and for an application EPG, choose the tenant, application profile, and the EPG to be the provider.

Step 7 In the **DHCP Server Address** field, enter the IP address for the server.

Step 8 Click **OK**.

Enable a Global MCP Instance Policy

Enable a global Mis-Cabling Protocol (MCP) instance policy. In the current implementation, only one instance of MCP runs in the system.

Procedure

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

Step 2 On the navigation bar, expand **Policies** and **Global**.

Step 3 Click **MCP Instance Policy default**.

Step 4 Change the **Admin State** to **Enabled**.

Step 5 Set other properties as needed for your fabric.

Step 6 Click **Submit**.

What to do next

Create an Error Disabled Recovery Policy

The error disabled recovery policy specifies the policy for re-enabling a port that was disabled due to one or more pre-defined error conditions.

Procedure

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

Step 2 On the navigation bar, expand **Policies** and **Global**.

Step 3 Click **Error Disabled Recovery Policy**..

Step 4 Double-click on an event to enable it for the recovery policy.

Step 5 Click the check box and click **Update**.

Step 6 Optional. Repeat steps 4 and 5 for more events.

Step 7 Optional. Reset the **Error disable recovery interval (sec)**.

Step 8 Click **Submit.**

Configure a Global Port Tracking Policy

Uplink failure detection can be enabled in the fabric access global port tracking policy. The port tracking policy monitors the status of links between leaf switches and spine switches. When an enabled port tracking policy is triggered, the leaf switches take down all access interfaces on the switch that have EPGs deployed on them.

Procedure

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

Step 2 On the navigation bar, expand **Policies** and **Global**.

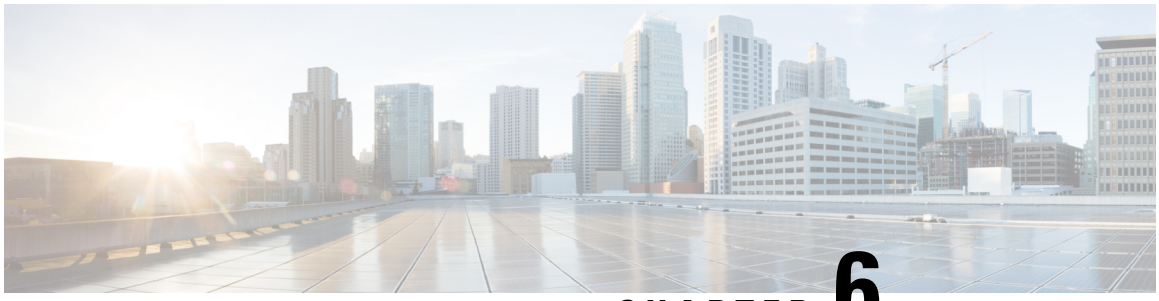
Step 3 Click **Port Tracking**.

Step 4 Enable port tracking by setting the **Port tracking state** to **on**.

Step 5 Optional. Change the **Daily restore timer**.

Step 6 Enter the **Number of active spine links that triggers port tracking**

Step 7 Click **Submit**



CHAPTER 6

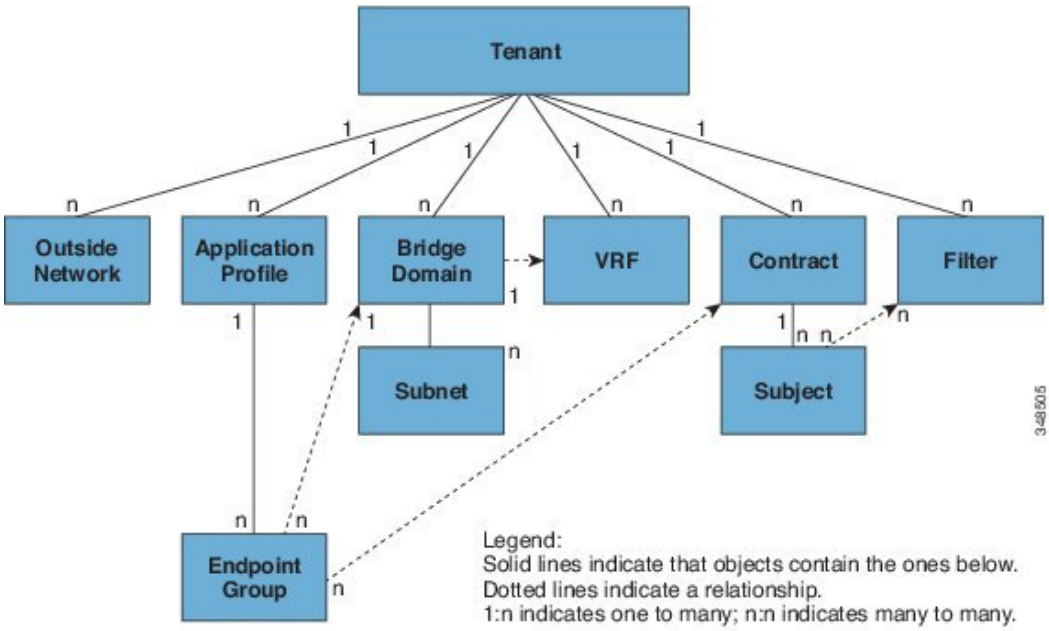
Basic User Tenant Configuration

This chapter contains the following sections:

Tenants

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

Figure 1: Tenants



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

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

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

Routing Within the Tenant

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

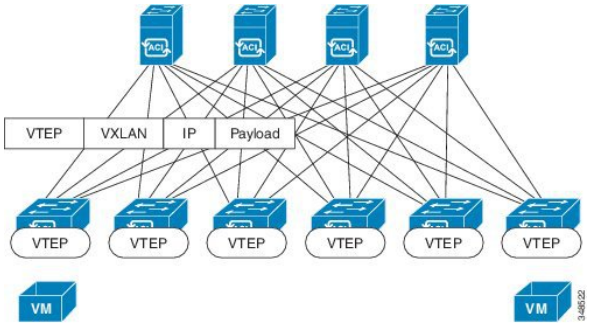
Layer 3 VNIDs Facilitate Transporting Inter-subnet Tenant Traffic

The ACI fabric provides tenant default gateway functionality that routes between the ACI fabric VXLAN networks. For each tenant, the fabric provides a virtual default gateway that spans all of the leaf switches

assigned to the tenant. It does this at the ingress interface of the first leaf switch connected to the endpoint. Each ingress interface supports the default gateway interface. All of the ingress interfaces across the fabric share the same router IP address and MAC address for a given tenant subnet.

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

Figure 2: ACI Decouples Identity and Location



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

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

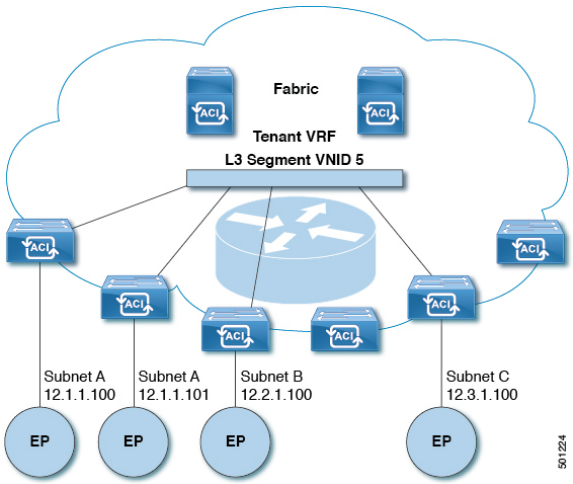
The IP interface has a unique IP address that identifies the VTEP device on the transport IP network known as the infrastructure VLAN. The VTEP device uses this IP address to encapsulate Ethernet frames and transmit the encapsulated packets to the transport network through the IP interface. A VTEP device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

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

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

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

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



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

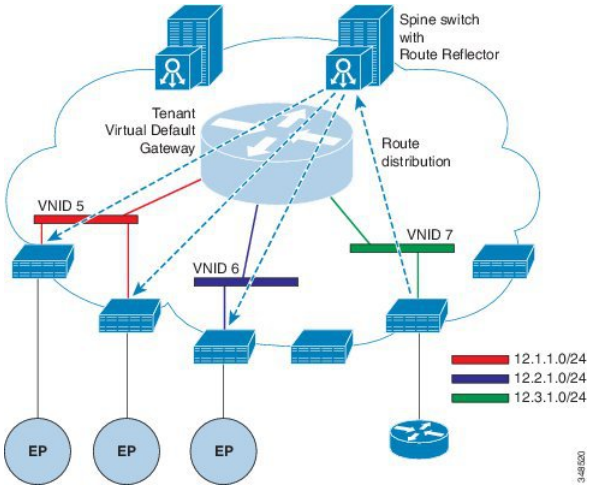
Traffic arriving at the fabric ingress that is sent to the ACI fabric default gateway is routed into the Layer 3 VNID. This provides very efficient forwarding in the fabric for traffic routed within the tenant. For example, with this model, traffic between 2 VMs belonging to the same tenant, on the same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

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

Router Peering and Route Distribution

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

Figure 4: Router Peering

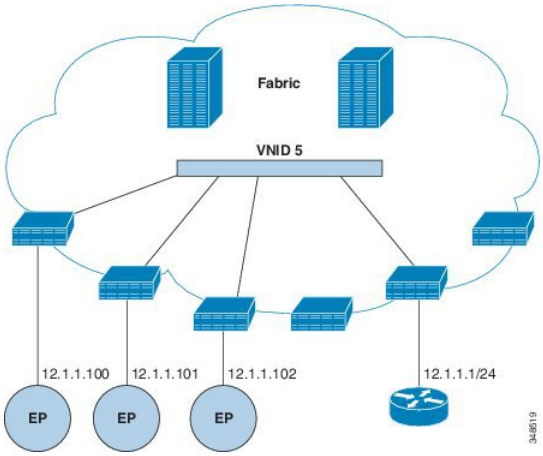


The routes that are learned through peering are sent to the spine switches. The spine switches act as route reflectors and distribute the external routes to all of the leaf switches that have interfaces that belong to the same tenant. These routes are longest prefix match (LPM) summarized addresses and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

Bridged Interface to an External Router

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

Figure 5: Bridged External Router



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

Configuring Route Reflectors

The ACI fabric route reflectors use multiprotocol BGP (MP-BGP) to distribute external routes within the fabric. To enable route reflectors in the ACI fabric, the fabric administrator must select the spine switches that will be the route reflectors, and provide the autonomous system (AS) number. Once route reflectors are enabled in the ACI fabric, administrators can configure connectivity to external networks as described in the following sections.

To connect external routers to the ACI fabric, the fabric infrastructure administrator configures spine nodes as Border Gateway Protocol (BGP) route reflectors. For redundancy purposes, more than one spine is configured as a router reflector node (one primary and one secondary reflector).

When a tenant needs to attach a WAN router to the ACI fabric, the infrastructure administrator configures the leaf node (as described below) to which the WAN router is being connected as WAN top of rack (ToR) and pairs this WAN ToR with one of the route reflector nodes as a BGP peer. When route reflectors are configured on the WAN ToR, they are able to advertise the tenant routes into the fabric.

Each leaf node can store up to 4000 routes. If a WAN router has to advertise more than 4000 routes, it should peer with multiple leaf nodes. The infrastructure administrator configures each of the paired leaf nodes with the routes (or route prefixes) that it can advertise.

The infrastructure administrator must configure an external WAN router connected to the fabric as follows:

1. Configure up to two spine nodes as route reflectors. For redundancy, configure primary and secondary route reflectors.
2. On WAN ToRs, configure the primary and secondary route reflector nodes.
3. On WAN ToRs, configure the routes that the ToR is responsible for advertising. This is optional and needs to be done only when the tenant router is known to advertise more than 4000 routes.

Configuring External Connectivity for Tenants

Before you can distribute the static route to the other leaf switches on the Application Centric Infrastructure (ACI) fabric, a multiprotocol BGP (MP-BGP) process must first be operating, and the spine switches must be configured as BGP route reflectors.

To integrate the ACI fabric into an external routed network, you can configure Open Shortest Path First (OSPF) for management tenant Layer 3 connectivity.

Configuring an MP-BGP Route Reflector Using the GUI

Procedure

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

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

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

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

The spine switch is marked as the route reflector node.

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

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

Step 5 On the menu bar, choose **Fabric** > **Fabric Policies** > **POD Policies**.

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

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

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

Step 9 In the **Navigation** pane, choose **Pod Policies** > **Profiles** > **default**. In the **Work** pane, from the **Fabric Policy Group** drop-down list, choose the pod policy that was created earlier. Click **Submit**.
The pod policy group is now applied to the fabric policy group.

Configuring an MP-BGP Route Reflector for the ACI Fabric

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

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

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

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

Configuring an MP-BGP Route Reflector Using the REST API

Procedure

Step 1 Mark the spine switches as route reflectors.

Example:

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

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

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

Example:

For the FuncP setup—

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

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

Example:

For the PodP setup—

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

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

Verifying the MP-BGP Route Reflector Configuration

Procedure

Step 1 Verify the configuration by performing the following actions:

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

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

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

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

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

b) Execute the following commands from the shell window

Example:

cd /mit/sys/bgp/inst

Example:

grep asn summary

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

Creating an OSPF External Routed Network for Management Tenant Using the GUI

- You must verify that the router ID and the logical interface profile IP address are different and do not overlap.

- The following steps are for creating an OSPF external routed network for a management tenant. To create an OSPF external routed network for a tenant, you must choose a tenant and create a VRF for the tenant.
- For more details, see *Cisco APIC and Transit Routing*.

Procedure

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

Step 2 In the **Navigation** pane, expand **Networking** > **External Routed Networks**.

Step 3 Right-click **External Routed Networks**, and click **Create Routed Outside**.

Step 4 In the **Create Routed Outside** dialog box, perform the following actions:

a) In the **Name** field, enter a name (RtdOut).
b) Check the **OSPF** check box.
c) In the **OSPF Area ID** field, enter an area ID.
d) In the **OSPF Area Control** field, check the appropriate check box.
e) In the **OSPF Area Type** field, choose the appropriate area type.
f) In the **OSPF Area Cost** field, choose the appropriate value.
g) In the **VRF** field, from the drop-down list, choose the VRF (inb).

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

h) From the **External Routed Domain** drop-down list, choose the appropriate domain.
i) Click the + icon for **Nodes and Interfaces Protocol Profiles** area.

Step 5 In the **Create Node Profile** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the node profile. (borderLeaf).
b) In the **Nodes** field, click the + icon to display the **Select Node** dialog box.
c) In the **Node ID** field, from the drop-down list, choose the first node. (leaf1).
d) In the **Router ID** field, enter a unique router ID.
e) Uncheck the **Use Router ID as Loopback Address** field.

Note By default, the router ID is used as a loopback address. If you want them to be different, uncheck the **Use Router ID as Loopback Address** check box.

f) Expand **Loopback Addresses**, and enter the IP address in the **IP** field. Click **Update**, and click **OK**.

Enter the desired IPv4 or IPv6 IP address.

g) In the **Nodes** field, expand the + icon to display the **Select Node** dialog box.

Note You are adding a second node ID.

h) In the **Node ID** field, from the drop-down list, choose the next node. (leaf2).
i) In the **Router ID** field, enter a unique router ID.
j) Uncheck the **Use Router ID as Loopback Address** field.

Note By default, the router ID is used as a loopback address. If you want them to be different, uncheck the **Use Router ID as Loopback Address** check box.

k) Expand **Loopback Addresses**, and enter the IP address in the **IP** field. Click **Update**, and click **OK**. Click **OK**.

Enter the desired IPv4 or IPv6 IP address.

Step 6 In the **Create Node Profile** dialog box, in the **OSPF Interface Profiles** area, click the + icon.

Step 7 In the **Create Interface Profile** dialog box, perform the following tasks:

a) In the **Name** field, enter the name of the profile (portProf).
b) In the **Interfaces** area, click the **Routed Interfaces** tab, and click the + icon.
c) In the **Select Routed Interfaces** dialog box, in the **Path** field, from the drop-down list, choose the first port (leaf1, port 1/40).
d) In the **IP Address** field, enter an IP address and mask. Click **OK**.
e) In the **Interfaces** area, click the **Routed Interfaces** tab, and click the + icon.
f) In the **Select Routed Interfaces** dialog box, in the **Path** field, from the drop-down list, choose the second port (leaf2, port 1/40).
g) In the **IP Address** field, enter an IP address and mask. Click **OK**.

Note This IP address should be different from the IP address you entered for leaf1 earlier.

h) In the **Create Interface Profile** dialog box, click **OK**.

The interfaces are configured along with the OSPF interface.

Step 8 In the **Create Node Profile** dialog box, click **OK**.

Step 9 In the **Create Routed Outside** dialog box, click **Next**.
The **Step 2 External EPG Networks** area is displayed.

Step 10 In the **External EPG Networks** area, click the + icon.

Step 11 In the **Create External Network** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the external network (extMgmt).
b) Expand **Subnet** and in the **Create Subnet** dialog box, in the **IP address** field, enter an IP address and mask for the subnet.
c) In the **Scope** field, check the desired check boxes. Click **OK**.
d) In the **Create External Network** dialog box, click **OK**.
e) In the **Create Routed Outside** dialog box, click **Finish**.

Note In the **Work** pane, in the **External Routed Networks** area, the external routed network icon (RtdOut) is now displayed.

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

Configuring external routed network connectivity involves the following steps:

1. Create a VRF under Tenant.
2. Configure L3 networking configuration for the VRF on the border leaf switches, which are connected to the external routed network. This configuration includes interfaces, routing protocols (BGP, OSPF, EIGRP), protocol parameters, route-maps.

3. Configure policies by creating external-L3 EPGs under tenant and deploy these EPGs on the border leaf switches. External routed subnets on a VRF which share the same policy within the ACI fabric form one "External L3 EPG" or one "prefix EPG".

Configuration is realized in two modes:

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

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

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

Procedure

Step 1 Configure the VLAN domain.

Example:

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

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

Example:

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

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

Example:

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

Step 4 Configure the VRF on a leaf.

Example:

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

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

Example:

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

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

Example:

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

```
apic1(config-leaf)# interface ethernet 1/11
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vlan-domain member dom_exampleCorp
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/11.500
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf exampleCorp_v1
apic1(config-leaf-if)# ip address 157.10.1.1/24
apic1(config-leaf-if)# ip router ospf default area 0.0.0.1
```

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

Example:

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

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

Example:

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

Creating Tenants, VRFs, and Bridge Domains

Tenants Overview

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

Tenant Creation

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

VRF and Bridge Domains

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

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

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

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

Procedure

Step 1 On the menu bar, click **TENANT** > **Add Tenant**.

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

a) In the **Name** field, enter a name.
b) Click the **Security Domains** + icon to open the **Create Security Domain** dialog box.
c) In the **Name** field, enter a name for the security domain. Click **Submit**.
d) In the **Create Tenant** dialog box, check the check box for the security domain that you created, and click **Submit**.

Step 3 In the **Navigation** pane, expand **Tenant-name** > **Networking**, and in the **Work** pane, drag the **VRF** icon to the canvas to open the **Create VRF** dialog box, and perform the following tasks:

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

b) Click **Submit** to complete the VRF configuration.

Step 4 In the **Networking** pane, drag the **BD** icon to the canvas while connecting it to the **VRF** icon. In the **Create Bridge Domain** dialog box that displays, perform the following tasks:

a) In the **Name** field, enter a name.
b) Click the **L3 Configurations** tab.
c) Expand **Subnets** to open the **Create Subnet** dialog box, enter the subnet mask in the **Gateway IP** field and click **OK**.
d) Click **Submit** to complete bridge domain configuration.

Step 5 In the **Networks** pane, drag the **L3** icon down to the canvas while connecting it to the **VRF** icon. In the **Create Routed Outside** dialog box that displays, perform the following tasks:

a) In the **Name** field, enter a name.
b) Expand **Nodes And Interfaces Protocol Profiles** to open the **Create Node Profile** dialog box.
c) In the **Name** field, enter a name.
d) Expand **Nodes** to open the **Select Node** dialog box.
e) In the **Node ID** field, choose a node from the drop-down list.
f) In the **Router ID** field, enter the router ID.
g) Expand **Static Routes** to open the **Create Static Route** dialog box.
h) In the **Prefix** field, enter the IPv4 or IPv6 address.
i) Expand **Next Hop Addresses** and in the **Next Hop IP** field, enter the IPv4 or IPv6 address.
j) In the **Preference** field, enter a number, then click **UPDATE** and then **OK**.
k) In the **Select Node** dialog box, click **OK**.
l) In the **Create Node Profile** dialog box, click **OK**.
m) Check the **BGP**, **OSPF**, or **EIGRP** check boxes if desired, and click **NEXT**. Click **OK** to complete the Layer 3 configuration.

To confirm L3 configuration, in the **Navigation** pane, expand **Networking** > **VRFs**.

Deploying EPGs

Statically Deploying an EPG on a Specific Port

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

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

Before you begin

The tenant where you deploy the EPG is already created.

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

Procedure

Step 1 Log in to the Cisco APIC.

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

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

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

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

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

b) From the **Bridge Domain** drop-down list, choose a bridge domain.

c) Check the **Statically Link with Leaves/Paths** check box.

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

d) Click **Next**.

e)

f) From the **Path** drop-down list, choose the static path to the destination EPG.

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

Step 7 Complete one of the following sets of steps:

Option	Description
If you want to deploy the EPG on...	Then
A node	**1.** Expand the **Leaves** area. **2.** From the **Node** drop-down list, choose a node. **3.** In the **Encap** field, enter the appropriate VLAN. **4.** (Optional) From the **Deployment Immediacy** drop-down list, accept the default **On Demand** or choose **Immediate**. **5.** (Optional) From the Mode drop-down list, accept the default **Trunk** or choose another mode.
A port on the node	**1.** Expand the **Paths** area. **2.** From the **Path** drop-down list, choose the appropriate node and port. **3.** (Optional) In the **Deployment Immediacy** field drop-down list, accept the default **On Demand** or choose **Immediate**. **4.** (Optional) From the Mode drop-down list, accept the default **Trunk** or choose another mode. **5.** In the **Port Encap** field, enter the secondary VLAN to be deployed. **6.** (Optional) In the **Primary Encap** field, enter the primary VLAN to be deployed.

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

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

Step 10 Complete one of the following actions:

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

Deploying an EPG on a Specific Port with APIC Using the NX-OS Style CLI

Procedure

Step 1 Configure a VLAN domain:

Example:

```
apic1(config)# vlan-domain dom1
apic1(config-vlan)# vlan 10-100
```

Step 2 Create a tenant:

Example:

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

Step 3 Create a private network/VRF:

Example:

```
apic1(config-tenant)# vrf context ctx1
apic1(config-tenant-vrf)# exit
```

Step 4 Create a bridge domain:

Example:

```
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member ctx1
apic1(config-tenant-bd)# exit
```

Step 5 Create an application profile and an application EPG:

Example:

```
apic1(config-tenant)# application AP1
apic1(config-tenant-app)# epg EPG1
apic1(config-tenant-app-epg)# bridge-domain member bd1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
```

Step 6 Associate the EPG with a specific port:

Example:

```
apic1(config)# leaf 1017
apic1(config-leaf)# interface ethernet 1/13
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 20 tenant t1 application AP1 epg EPG1
```

Note The vlan-domain and vlan-domain member commands mentioned in the above example are a pre-requisite for deploying an EPG on a port.

Deploying an EPG on a Specific Port with APIC Using the REST API

Before you begin

The tenant where you deploy the EPG is created.

Procedure

Deploy an EPG on a specific port.

Example:

```
<fvTenant name="<tenant_name>" dn="uni/tn-test1" >
    <fvCtx name="<network_name>" pcEnfPref="enforced" knwMcastAct="permit"/>
    <fvBD name="<bridge_domain_name>" unkMcastAct="flood" >
        <fvRsCtx tnFvCtxName="<network_name>"/>
    </fvBD>
    <fvAp name="<application_profile>" >
        <fvAEPg name="<epg_name>" >
            <fvRsPathAtt tDn="topology/pod-1/paths-1017/pathep-[eth1/13]" mode="regular"
instrImedcy="immediate" encap="vlan-20"/>
        </fvAEPg>
    </fvAp>
</fvTenant>
```

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

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

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

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

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

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

Before you begin

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

Procedure

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

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

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

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

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

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

Note Optionally, you can enter a modified name.

c) Click the + icon to configure the switch interfaces.

d) In the **Interface Type** field, click the **Individual** radio button.

e) In the **Interfaces** field, enter the range of desired interfaces.

f) In the **Interface Selector Name** field, an interface name is automatically populated.

Note Optionally, you can enter a modified name.

g) In the **Interface Policy Group** field, choose the **Create One** radio button.

h) From the **Link Level Policy** drop-down list, choose the appropriate link level policy.

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

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

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

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

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

Creating AEP, Domains, and VLANs to Deploy an EPG on a Specific Port Using the NX-OS Style CLI

Before you begin

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

Procedure

Step 1 Create a VLAN domain and assign VLAN ranges:

Example:

```
apic1(config)# vlan-domain domP
apic1(config-vlan)# vlan 10
apic1(config-vlan)# vlan 25
apic1(config-vlan)# vlan 50-60
apic1(config-vlan)# exit
```

Step 2 Create an interface policy group and assign a VLAN domain to the policy group:

Example:

```
apic1(config)# template policy-group PortGroup
apic1(config-pol-grp-if)# vlan-domain member domP
```

Step 3 Create a leaf interface profile, assign an interface policy group to the profile, and assign the interface IDs on which the profile will be applied:

Example:

```
apic1(config)# leaf-interface-profile InterfaceProfile1
apic1(config-leaf-if-profile)# leaf-interface-group range
apic1(config-leaf-if-group)# policy-group PortGroup
apic1(config-leaf-if-group)# interface ethernet 1/11-13
apic1(config-leaf-if-profile)# exit
```

Step 4 Create a leaf profile, assign the leaf interface profile to the leaf profile, and assign the leaf IDs on which the profile will be applied:

Example:

```
apic1(config)# leaf-profile SwitchProfile-1019
apic1(config-leaf-profile)# leaf-interface-profile InterfaceProfile1
apic1(config-leaf-profile)# leaf-group range
apic1(config-leaf-group)# leaf 1019
apic1(config-leaf-group)#
```

Creating AEP, Domains, and VLANs to Deploy an EPG on a Specific Port Using the REST API

Before you begin

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

Procedure

Step 1 Create the interface profile, switch profile and the Attach Entity Profile (AEP).

Example:

```
   <infraInfra>

      <infraNodeP  name="<switch_profile_name>" dn="uni/infra/nprof-<switch_profile_name>"
 >
           <infraLeafS  name="SwitchSeletor" descr="" type="range">
                <infraNodeBlk name="nodeBlk1" descr="" to_="1019" from_="1019"/>
           </infraLeafS>
          <infraRsAccPortP tDn="uni/infra/accportprof-<interface_profile_name>"/>
      </infraNodeP>

      <infraAccPortP name="<interface_profile_name>"
dn="uni/infra/accportprof-<interface_profile_name>" >
           <infraHPortS  name="portSelector"  type="range">
                <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-<port_group_name>"
fexId="101"/>
              <infraPortBlk name="block2"  toPort="13" toCard="1" fromPort="11"
fromCard="1"/>
         </infraHPortS>
     </infraAccPortP>
```

```
    <infraAccPortGrp  name="<port_group_name>"
dn="uni/infra/funcprof/accportgrp-<port_group_name>" >
          <infraRsAttEntP tDn="uni/infra/attentp-<attach_entity_profile_name>"/>
          <infraRsHIfPol tnFabricHIfPolName="1GHifPol"/>
    </infraAccPortGrp>

    <infraAttEntityP  name="<attach_entity_profile_name>"
dn="uni/infra/attentp-<attach_entity_profile_name>" >
         <infraRsDomP tDn="uni/phys-<physical_domain_name>"/>
    </infraAttEntityP>

<infraInfra>
```

Step 2 Create a domain.

Example:

```
<physDomP  name="<physical_domain_name>" dn="uni/phys-<physical_domain_name>">
    <infraRsVlanNs tDn="uni/infra/vlanns-[<vlan_pool_name>]-static"/>
</physDomP>
```

Step 3 Create a VLAN range.

Example:

```
<fvnsVlanInstP  name="<vlan_pool_name>" dn="uni/infra/vlanns-[<vlan_pool_name>]-static"
allocMode="static">
    <fvnsEncapBlk name="" descr="" to="vlan-25" from="vlan-10"/>
</fvnsVlanInstP>
```

Step 4 Associate the EPG with the domain.

Example:

```
<fvTenant  name="<tenant_name>" dn="uni/tn-" >
    <fvAEPg prio="unspecified" name="<epg_name>" matchT="AtleastOne"
dn="uni/tn-test1/ap-AP1/epg-<epg_name>" descr="">
        <fvRsDomAtt tDn="uni/phys-<physical_domain_name>" instrImedcy="immediate"
resImedcy="immediate"/>
    </fvAEPg>
</fvTenant>
```

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

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

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

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

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

Before you begin

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

Procedure

Step 1 Navigate to the target attached entity profile.

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

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

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

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

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

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

Step 4 Click **Submit**.
the application EPGs associated with this attached entity profile are deployed to all the ports on all the switches associated with this attached entity profile.

Deploying an EPG through an Interface Policy Group to Multiple Interfaces Using the NX-OS Style CLI

In the NX-OS CLI, an attached entity profile is not explicitly defined to associate with an EPG for rapid deployment; instead the interface policy group is defined, assigned a domain, applied to all the ports associated with a VLAN and configured to include the application EPG to be deployed over that VLAN.

Before you begin

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

Procedure

Step 1 Associate the target EPG with the interface policy group.

The sample command sequence specifies an interface policy group **pg3** associated with VLAN domain, **domain1**, and with VLAN **1261**. The application EPG, **epg47** is deployed to all interfaces associated with this policy group.

Example:

```
 apic1# configure terminal
apic1(config)# template policy-group pg3
apic1(config-pol-grp-if)# vlan-domain member domain1
apic1(config-pol-grp-if)# switchport trunk allowed vlan 1261 tenant tn10 application pod1-AP

      epg epg47
```

Step 2 Check the target ports to ensure deployment of the policies of the interface policy group associated with application EPG.

The output of the sample **show** command sequence indicates that policy group **pg3** is deployed on Ethernet port **1/20** on leaf switch **1017**.

Example:

```
 apic1# show run leaf 1017 int eth 1/20
# Command: show running-config leaf 1017 int eth 1/20
# Time: Mon Jun 27 22:12:10 2016
  leaf 1017
    interface ethernet 1/20
      policy-group pg3
      exit
    exit
ifav28-ifc1#
```

Deploying an EPG through an AEP to Multiple Interfaces Using the REST API

The interface selectors in the AEP enable you to configure multiple paths for an AEPg. The following can be selected:

1. A node or a group of nodes
2. An interface or a group of interfaces

   The interfaces consume an interface policy group (and so an `infra:AttEntityP`).
3. The infra:AttEntityP is associated to the AEPg, thus specifying the VLANs to use.

   An infra:AttEntityP can be associated with multiple AEPgs with different VLANs.

When you associate the infra:AttEntityP with the AEPg, as in 3, this deploys the AEPg on the nodes selected in 1, on the interfaces in 2, with the VLAN provided by 3.

In this example, the AEPg `uni/tn-Coke/ap-AP/epg-EPG1` is deployed on interfaces 1/10, 1/11, and 1/12 of nodes 101 and 102, with vlan-102.

Before you begin

- Create the target application EPG (`AEPg`).
- Create the VLAN pool containing the range of VLANs you wish to use for EPG deployment with the Attached Entity Profile (AEP).
- Create the physical domain and link it to the VLAN pool and AEP.

Procedure

To deploy an AEPg on selected nodes and interfaces, send a post with XML such as the following:

Example:

```
 <infraInfra dn="uni/infra">
   <infraNodeP name="NodeProfile">
       <infraLeafS name="NodeSelector" type="range">
          <infraNodeBlk name="NodeBlok" from_="101" to_="102"/>
          <infraRsAccPortP tDn="uni/infra/accportprof-InterfaceProfile"/>
        </infraLeafS>
   </<infraNodeP>

   <infraAccPortP name="InterfaceProfile">
      <infraHPortS name="InterfaceSelector" type="range">
          <infraPortBlk name=" InterfaceBlock" fromCard="1" toCard="1" fromPort="10"
toPort="12"/>
          <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-PortGrp" />
      </infraHPortS>
   </infraAccPortP>

   <infraFuncP>
       <infraAccPortGrp name="PortGrp">
           <infraRsAttEntP  tDn="uni/infra/attentp-AttEntityProfile"/>
       </infraAccPortGrp>
   </infraFuncP>

   <infraAttEntityP name="AttEntityProfile" >
```

```
        <infraGeneric name="default" >
           <infraRsFuncToEpg tDn="uni/tn-Coke/ap-AP/epg-EPG1" encap="vlan-102"/>
        </infraGeneric>
    </infraAttEntityP>
</infraInfra>
```

Microsegmented EPGs

Using Microsegmentation with Network-based Attributes on Bare Metal

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

Using an IP-based Attribute

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

Using a MAC-based Attribute

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

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

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

Procedure

Step 1 Log into the Cisco APIC.

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

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

Step 4 Take one of the following actions:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

If you want to use...	Then...
An IP-based attribute	**1.** In the **Create IP Attribute** dialog box, in the **Name** field, enter a name. We recommend that you choose a name that reflects the filter's function. **2.** In the **IP Address** field, enter an IP address or a subnet with the appropriate subnet mask. **3.** Click **OK**. **4.** (Optional) Create a second IP Address filter by repeating Step 10 c through Step 11 c. You might want to do this to include discontinuous IP addresses in the microsegment. **5.** In the **Create uSeg EPG** dialog box, click **SUBMIT**.
A MAC-based attribute	**1.** In the **Create MAC Attribute** dialog box, in the **Name** field, enter a name. We recommend that you choose a name that reflects the filter's function. **2.** In the **MAC Address** field, enter a MAC address. **3.** Click **OK**. **4.** In the **Create uSeg EPG** dialog box, click **SUBMIT**.

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

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

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

c) On the right side of the work pane, click **ACTIONS** and then choose **Add Physical Domain Association** from the drop-down list.
d) In the **Add Physical Domain Association** dialog box, choose a profile from the **Physical Domain Profile** drop-down list.
e) In the **Deploy Immediacy** area, accept the default **On Demand**.
f) In the **Resolution Immediacy** area, accept the default **On Demand**.
g) Click **SUBMIT**.

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

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

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

What to do next

Verify that the attribute-based EPG was created correctly.

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

Configuring a Network-Based Microsegmented EPG in a Bare-Metal Environment Using the NX-OS Style CLI

This section describes how to configure microsegmentation with Cisco ACI using network-based attributes (IP address or MAC address) within a base EPG in a bare-metal environment.

Procedure

	Command or Action	Purpose

Step 1	In the CLI, enter configuration mode:
	Example:
	`apic1# configure`
	`apic1(config)#`	

Step 2	Create the microsegment:
	Example:
	This example uses a filter based on an IP address.
	`apic1(config)# tenant cli-ten1`
	`apic1(config-tenant)# application cli-a1`
	`apic1(config-tenant-app)# epg cli-uepg1`
	`type micro-segmented`
	`apic1(config-tenant-app-uepg)#`
	`bridge-domain member cli-bd1`
	`apic1(config-tenant-app-uepg)# attribute`
	`cli-upg-att match ip <X.X.X.X>`
	`#Schemes to express the ip`	

	Command or Action	Purpose

	A.B.C.D IP Address
	A.B.C.D/LEN IP Address and mask
	Example:
	This example uses a filter based on a MAC address.
	`apic1(config)# tenant cli-ten1`
	`apic1(config-tenant)# application cli-a1`
	`apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented`
	`apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1`
	`apic1(config-tenant-app-uepg)# attribute cli-upg-att match mac`
	`<FF-FF-FF-FF-FF-FF>`
	`#Schemes to express the mac`
	`E.E.E MAC address (Option 1)`
	`EE-EE-EE-EE-EE-EE MAC address (Option 2)`
	`EE:EE:EE:EE:EE:EE MAC address (Option 3)`
	`EEEE.EEEE.EEEE MAC address (Option 4)`
	Example:
	This example uses a filter based on a MAC address and enforces intra-EPG isolation between all members of this uSeg EPG:
	`apic1(config)# tenant cli-ten1`
	`apic1(config-tenant)# application cli-a1`
	`apic1(config-tenant-app)# epg cli-uepg1 type micro-segmented`
	`apic1(config-tenant-app-uepg)# isolation enforced`
	`apic1(config-tenant-app-uepg)# bridge-domain member cli-bd1`
	`apic1(config-tenant-app-uepg)# attribute cli-upg-att match mac`
	`<FF-FF-FF-FF-FF-FF>`
	`#Schemes to express the mac`
	`E.E.E MAC address (Option 1)`
	`EE-EE-EE-EE-EE-EE MAC address (Option 2)`
	`EE:EE:EE:EE:EE:EE MAC address (Option 3)`
	`EEEE.EEEE.EEEE MAC address (Option 4)`	

Step 3	Deploy the EPG.
	Example:
	This example deploys the EPG and bids to the leaf.
	`apic1(config)# leaf 101`
	`apic1(config-leaf)# deploy-epg tenant cli-ten1 application cli-a1 epg cli-uepg1 type micro-segmented`	

Step 4	Verify the microsegment creation: **Example:**	

	Command or Action	Purpose
	apic1(config-tenant-app-uepg)# show running-config # Command: show running-config tenant cli-ten1 application cli-app1 epg cli-uepg1 type micro-segmented # Time: Thu Oct 8 11:54:32 2015 tenant cli-ten1 application cli-app1 epg cli-esx1bu type micro-segmented bridge-domain cli-bd1 attribute cli-uepg-att match mac 00:11:22:33:44:55 exit exit exit	

Configuring a Network-Based Microsegmented EPG in a Bare-Metal Environment Using the REST API

This section describes how to configure network attribute microsegmentation with Cisco ACI in a bare-metal environment using the REST API.

Procedure

Step 1 Log in to the Cisco APIC.

Step 2 Post the policy to https://*apic-ip-address*/api/node/mo/.xml.

Example:

A: The following example configures a microsegment named 41-subnet using an IP-based attribute.

```
<polUni>
  <fvTenant dn="uni/tn-User-T1" name="User-T1">
      <fvAp dn="uni/tn-User-T1/ap-Base-EPG" name="Base-EPG">
            <fvAEPg dn="uni/tn-User-T1/ap-Base-EPG/epg-41-subnet" name="41-subnet"
pcEnfPref="enforced” isAttrBasedEPg="yes" >
               <fvRsBd tnFvBDName="BD1" />
               <fvCrtrn name="Security1">
                  <fvIpAttr name="41-filter" ip="12.41.0.0/16"/>
               </fvCrtrn>
          </fvAEPg>
       </fvAp>
   </fvTenant>
</polUni>
```

Example:

This example is for base EPG for Example A: .

```
<polUni>
  <fvTenant dn="uni/tn-User-T1" name="User-T1">
      <fvAp dn="uni/tn-User-T1/ap-Base-EPG" name="Base-EPG">
         <fvAEPg dn="uni/tn-User-T1/ap-Base-EPG/baseEPG” name=“baseEPG” pcEnfPref="enforced”
 >
               <fvRsBd tnFvBDName="BD1" />
          </fvAEPg>
       </fvAp>
```

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

Example:

B: The following example configures a microsegment named useg-epg using a MAC-based attribute.

```
<polUni>
  <fvTenant name="User-T1">
        <fvAp name="customer">
            <fvAEPg name="useg-epg" isAttrBasedEPg="true">
              <fvRsBd tnFvBDName="BD1"/>
             <fvRsDomAtt  instrImedcy="immediate" resImedcy="immediate" tDn="uni/phys-phys"
 />
            <fvRsNodeAtt tDn="topology/pod-1/node-101" instrImedcy="immediate" />
            <fvCrtrn name="default">
                <fvMacAttr name="mac" mac="00:11:22:33:44:55" />
              </fvCrtrn>
            </fvAEPg>
        </fvAp>
  </fvTenant>
</polUni>
```

IP Address-Based Microsegmented EPG as a Shared Resource

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

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

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

Before you begin

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

Note

- For directions on configuring an IP address based EPG in a physical environment, see Using Microsegmentation with Network-based Attributes on Bare Metal, on page 159
- For directions on configuring an IP address based EPG in a virtual environment, see *Configuring Microsegmentation with Cisco ACI* in the *Cisco ACI Virtualization Guide*.

Procedure

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

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

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

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

b) Enter a name in the Name field

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

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

d) Click **Submit**.

Step 3 Create a shared subnet for the target EPG.

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

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

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

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

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

e) Click **Submit**.

Configuring an IP-based Microsegmented EPG as a Shared Resource Using the NX-OS CLI

Before you begin

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

Procedure

	Command or Action	Purpose

Step 1	Enable the IP address microsegmented EPG for shared service by associating the EPG with the IP address of its subnet.
	Example:
	`apic-1(config)# tenant t0`
	`apic-1(config-tenant-app)# epg cli-epg type micro-segmented`
	`apic-1(config-tenant-app-uepg)# bridge-domain member b0`
		In this example, microsegmented EPG, cli-epg, is configured with the **`ip-use-epg-subnet`** option (useFvSubnet), thus associating the EPG with the IP address of its subnet. APIC then advertises that subnet address, thus making the EPG accessible as a service to devices on VRFs other than the one on which the EPG is native.

	Command or Action	Purpose
	apic-1(config-tenant-app-uepg)# **attribute ip match ip-use-epg-subnet** apic-1(config-tenant-app-uepg)# show run # Command: show running-config tenant t0 application a0 epg cli-epg type micro-segmented # Time: Thu Sep 22 00:17:07 2016 tenant t0 application a0 epg cli-epg type micro-segmented bridge-domain member b0 **attribute ip match ip-use-epg-subnet** exit exit Exit	
Step 2	Deploy the EPG to a leaf.	In this example, microsegmented EPG, cli-epg, is deployed to leaf 102. apic-1(config)# leaf 102 apic-1(config-leaf)# deploy-epg tenant t0 application a0 epg cli-epg type micro-segmented apic-1(config-leaf)# show run # Command: show running-config leaf 102 # Time: Thu Sep 22 00:18:46 2016 leaf 102 deploy-epg tenant t0 application a0 epg cli-epg type micro-segmented

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

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

Procedure

To configure an IP address-attribute microsegmented EPG `epg3` with a shared subnet, with an IP address and 32-bit mask, send a post with XML such as the following example. In the IP attributes, the attribute **usefvSubnet** is set to "yes."

Example:

```
<fvAEPg descr="" dn="uni/tn-t0/ap-a0/epg-epg3" fwdCtrl=""
          isAttrBasedEPg="yes" matchT="AtleastOne" name="epg3" pcEnfPref="unenforced"
prefGrMemb="exclude"prio="unspecified">
        <fvRsCons prio="unspecified" tnVzBrCPName="ip-epg"/>
        <fvRsNodeAtt descr="" encap="unknown" instrImedcy="immediate" mode="regular"
tDn="topology/pod-2/node-106"/>
        <fvSubnet ctrl="" descr="" ip="56.4.0.2/32" name="" preferred="no"
           scope="public,shared" virtual="no"/>
        <fvRsDomAtt classPref="encap" delimiter="" encap="unknown" encapMode="auto"
instrImedcy="immediate"
           primaryEncap="unknown" resImedcy="immediate" tDn="uni/phys-vpc"/>
```

```
        <fvRsCustQosPol tnQosCustomPolName=""/>
        <fvRsBd tnFvBDName="b2"/>
        <fvCrtrn descr="" match="any" name="default" ownerKey="" ownerTag="" prec="0">
           <fvIpAttr descr="" ip="1.1.1.3" name="ipv4" ownerKey="" ownerTag=""
usefvSubnet="yes"/>
        </fvCrtrn>
        <fvRsProv matchT="AtleastOne" prio="unspecified" tnVzBrCPName="ip-epg"/>
        <fvRsProv matchT="AtleastOne" prio="unspecified" tnVzBrCPName="shared-svc"/>
</fvAEPg>
```

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

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

Before you begin

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

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

Procedure

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

a) In the APIC GUI, click **Tenant** > **tenant_name** > **Application Profiles** > **epg_name** > **uSeg EPGs** > **uSeg EPGs** > **uSeg_epg_name**.
b) With the folder for the target IP address-based uSeg EPG still open in the APIC navigation pane, click the **Subnets** folder.
c) In the **Subnets** window, select the subnet that is advertised and shared with other VRFs and click **Actions** > **Delete**. then
d) Click **Yes** to confirm the deletion.

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

a) With the folder for the target IP address-based uSeg EPG still open in the APIC navigation pane, click the name of the micro-segmented EPG to display the to display the EPG's **Properties** dialog.
b) In the **Properties** dialog, locate the **uSeg Attributes** table, and locate the IP attribute item with the **Use FV Subnet** option enabled.
c) Double-click that item to display the **Edit IP Attribute** dialog.
d) In the **Edit IP Attribute** dialog, deselect the **Use FV Subnet** option.
e) Assign another IP address attribute in the IP Address field.

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

f) Click **Submit**.

Unconfiguring an IP-based Microsegmented EPG as a Shared Resource Using the NX-OS Style CLI

To unconfigure an IP address-based microsegmented EPG as a shared service, disable the ip-use-epg-subnet option for that EPG.

Before you begin

Procedure

	Command or Action	Purpose

Step 1	Disable the ip-use-epg-subnet option.
	Example:
	`apic-1(config)# tenant t0`
	`apic-1(config-tenant-app)# epg cli-epg type micro-segmented`
	`apic-1(config-tenant-app-uepg)# no attribute ip match ip-use-epg-subnet`
	`apic-1(config-tenant-app-uepg)# exit`
	`apic-1(config-tenant-app)# exit`
		The example code disables the ip-use-epg-subnet option for the microsegmented EPG cli-epg.

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

You can disable an IP address-based microsegmented EPG by setting the **usefvSubnet** property to "no."

Procedure

In the API structure for the microsegmented EPG currently configured as a shared service, change the value of the usefvSubnet property from "yes" to "no."

In the example, the IP address-based microsegmented EPG, epg3, is disabled as a shared service.

Example:

```
<fvAEPg descr="" dn="uni/tn-t0/ap-a0/epg-epg3" fwdCtrl="" isAttrBasedEPg="yes"
matchT="AtleastOne" name="epg3" pcEnfPref="unenforced" prefGrMemb="exclude"prio="unspecified">

   <fvRsCons prio="unspecified" tnVzBrCPName="ip-epg"/>
   <fvRsNodeAtt descr="" encap="unknown" instrImedcy="immediate" mode="regular"
tDn="topology/pod-2/node-106"/>
  <fvSubnet ctrl="" descr="" ip="56.4.0.2/32" name="" preferred="no" scope="public,shared"
 virtual="no"/>
   <fvRsDomAtt classPref="encap" delimiter="" encap="unknown" encapMode="auto"
instrImedcy="immediate" primaryEncap="unknown" resImedcy="immediate" tDn="uni/phys-vpc"/>
   <fvRsCustQosPol tnQosCustomPolName=""/>
   <fvRsBd tnFvBDName="b2"/>
   <fvCrtrn descr="" match="any" name="default" ownerKey="" ownerTag="" prec="0">
     <fvIpAttr descr="" ip="1.1.1.3" name="ipv4" ownerKey="" ownerTag="" usefvSubnet="no"/>

   </fvCrtrn>
   <fvRsProv matchT="AtleastOne" prio="unspecified" tnVzBrCPName="ip-epg"/>
   <fvRsProv matchT="AtleastOne" prio="unspecified" tnVzBrCPName="shared-svc"/>
```

```
</fvAEPg>
```

Deploying Application Profiles and Contracts

Security Policy Enforcement

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

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

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

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

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

Contracts Contain Security Policy Specifications

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

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

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

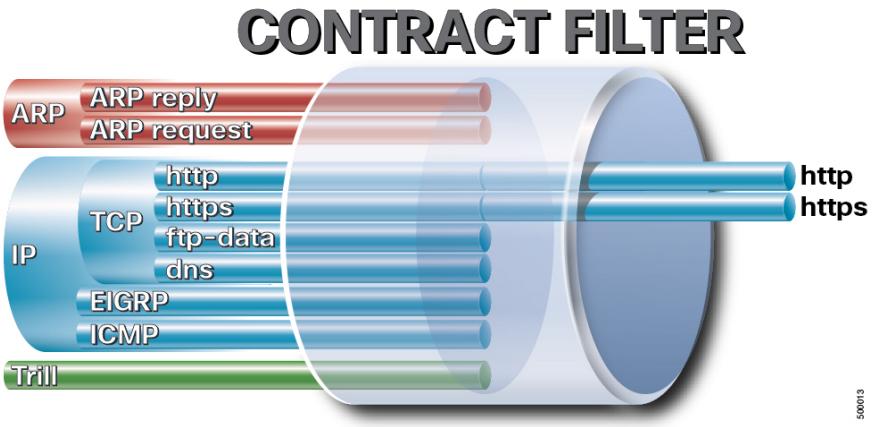
There is also directionality in the relationship between EPGs and contracts. EPGs can either provide or consume a contract. An EPG that provides a contract is typically a set of endpoints that provide a service to a set of client devices. The protocols used by that service are defined in the contract.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

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

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

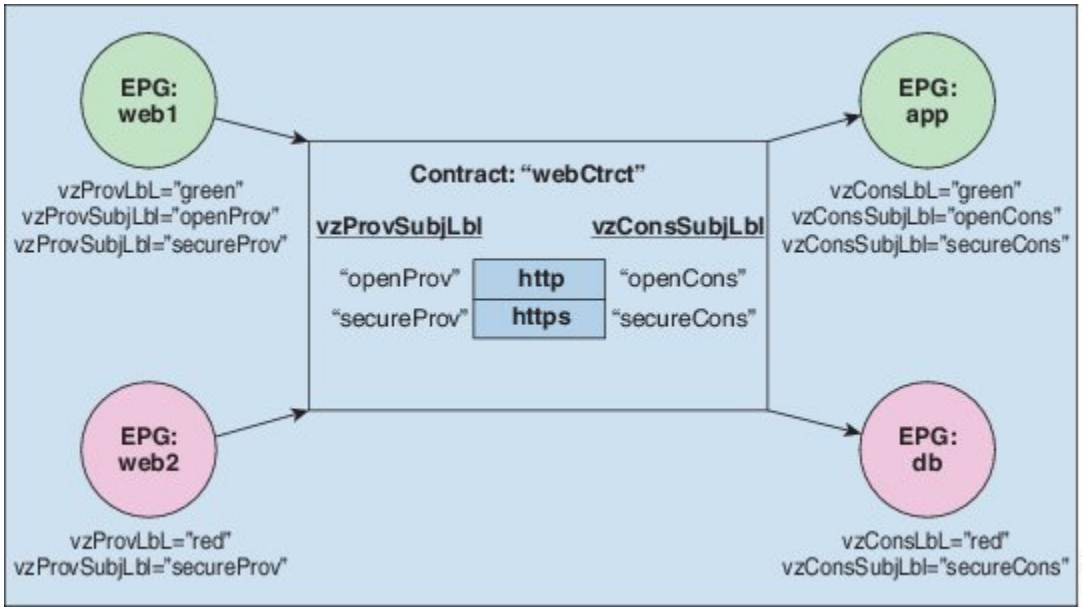
The contract is constructed in a hierarchical manner. It consists of one or more subjects, each subject contains one or more filters, and each filter can define one or more protocols.

Figure 6: Contract Filters



The following figure shows how contracts govern EPG communications.

Figure 7: Contracts Determine EPG to EPG Communications



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

These same constructs also apply for policies that govern virtual machine hypervisors. When an EPG is placed in a virtual machine manager (VMM) domain, the APIC downloads all of the policies that are associated with the EPG to the leaf switches with interfaces connecting to the VMM domain. For a full explanation of VMM domains, see the *Virtual Machine Manager Domains* chapter of *Application Centric Infrastructure Fundamentals*. When this policy is created, the APIC pushes it (pre-populates it) to a VMM domain that specifies which switches allow connectivity for the endpoints in the EPGs. The VMM domain defines the set of switches and ports that allow endpoints in an EPG to connect to. When an endpoint comes on-line, it is associated with the appropriate EPGs. When it sends a packet, the source EPG and destination EPG are derived from the packet and the policy defined by the corresponding contract is checked to see if the packet is allowed. If yes, the packet is forwarded. If no, the packet is dropped.

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

In detail, contracts are comprised of the following items:

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

- Permit the traffic (regular contracts, only)
- Mark the traffic (DSCP/CoS) (regular contracts, only)
- Redirect the traffic (regular contracts, only, through a service graph)
- Copy the traffic (regular contracts, only, through a service graph or SPAN)
- Block the traffic (taboo contracts)

  With Cisco APIC Release 3.2(x) and switches with names that end in EX or FX, you can alternatively use a subject Deny action or Contract or Subject Exception in a standard contract to block traffic with specified patterns.
- Log the traffic (taboo contracts and regular contracts)

- Aliases—(Optional) A changeable name for an object. Although the name of an object, once created, cannot be changed, the Alias is a property that can be changed.

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

Three-Tier Application Deployment

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

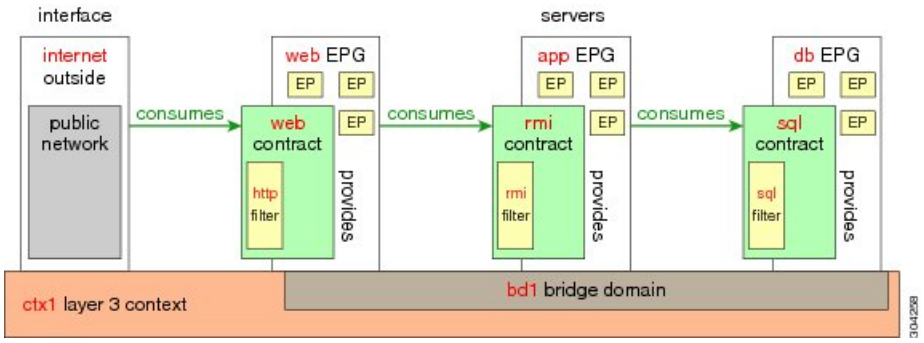
Contracts are policies that enable inter-End Point Group (inter-EPG) communication. These policies are the rules that specify communication between application tiers. If no contract is attached to the EPG, inter-EPG communication is disabled by default. No contract is required for intra-EPG communication because intra-EPG communication is always allowed.

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

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

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

Figure 8: Three-Tier Application Diagram



Parameters to Create a Filter for http

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

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

Parameters to Create Filters for rmi and sql

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

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

Example Application Profile Database

The application profile database in this example is as follows:

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

Creating an Application Profile Using the GUI

Procedure

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

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

Creating EPGs Using the GUI

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

Procedure

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

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

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

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

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

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

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

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

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

f) If you have Cisco AVS, from the **Encap Mode** drop-down list, choose an encapsulation mode.

You can choose one of the following encap modes:

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

g) Click **Update** and then click **FINISH**.

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

Configuring Contracts Using the APIC GUI

Creating a Filter Using the GUI

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

Before you begin

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

Procedure

Step 1 On the menu bar, choose **Tenants**. In the **Navigation** pane, expand the ***tenant-name*** > **Contracts**, right-click **Filters**, and click **Create Filter**.

Note In the **Navigation** pane, you expand the tenant where you want to add filters.

Step 2 In the **Create Filter** dialog box, perform the following actions:

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

Step 3 Expand **Entries** in the **Name** field. Follow the same process to add another entry with HTTPS as the **Destination** port, and click **Update**.

This new filter rule is added.

Step 4 Follow the same process in the earlier steps to create two more filters (rmi and sql) and use the parameters provided in Parameters to Create Filters for rmi and sql, on page 173.

Creating a Contract Using the GUI

Procedure

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

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

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

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

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

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

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

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

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

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

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

Consuming and Providing Contracts Using the GUI

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

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

Procedure

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

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

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

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

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

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

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

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

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

Configuring Contracts Using the NX-OS Style CLI

Configuring Contracts

Contracts are configured under a tenant with the following tasks:

- Define filters as access lists
- Define the contract and subjects
- Link the contract to an EPG

The tasks need not follow this order. For example, you can link a contract name to an EPG before you have defined the contract.

Note Filters (ACLs) in APIC use **match** instead of **permit | deny** as in the traditional NX-OS ACL. The purpose of a filter entry is only to match a given traffic flow. The traffic will be permitted or denied when the ACL is applied on a contract or on a taboo contract.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `tenant exampleCorp`	Creates a tenant if it does not exist and enters the tenant configuration mode.
Step 3	**access-list** *acl-name* **Example:** `apic1(config-tenant)# access-list http_acl`	Creates an access list (filter) that can be used in a contract.
Step 4	(Optional) **match** {**arp** | **icmp** | **ip**} **Example:** `apic1(config-tenant-acl)# match arp`	Creates a rule to match traffic of the selected protocol.

	Command or Action	Purpose

Step 5	(Optional) **match** {**tcp** | **udp**} [**src** *from*[*-to*]] [**dest** *from*[*-to*]]
	Example:
	`apic1(config-tenant-acl)# match tcp dest 80`
	`apic1(config-tenant-acl)# match tcp dest 443`
		Creates a rule to match TCP or UDP traffic.

Step 6	(Optional) **match raw** *options* **Example:** `apic1(config-tenant-acl)#`	Creates a rule to match a raw vzEntry.
Step 7	**exit** **Example:** `apic1(config-tenant-acl)# exit`	Returns to the tenant configuration mode.
Step 8	**contract** *contract-name* **Example:** `apic1(config-tenant)# contract web80`	Creates a contract and enters the contract configuration mode.
Step 9	**subject** *subject-name* **Example:** `apic1(config-tenant-contract)# subject web80`	Creates a contract subject and enters the subject configuration mode.
Step 10	(Optional) **[no] access-group** *acl-name* [**in** | **out** | **both**] **Example:** `apic1(config-tenant-contract-subj)# access-group http_acl both`	Adds (removes) an access list from the contract, specifying the direction of the traffic to be matched.
Step 11	(Optional) **[no] label name** *label-name* {**provider** | **consumer**} **Example:** `apic1(config-tenant-contract-subj)#`	Adds (removes) a provider or consumer label to the subject.
Step 12	(Optional) **[no] label match** {**provider** | **consumer**} [**any** | **one** | **all** | **none**] **Example:** `apic1(config-tenant-contract-subj)#`	Specifies the match type for the provider or consumer label: • **any**—Match if any label is found in the contract relation. • **one**—Match if exactly one label is found in the contract relation. • **all**—Match if all labels are found in the contract relation.

	Command or Action	Purpose
		• **none**—Match if no labels are found in the contract relation.
Step 13	**exit** **Example:** `apic1(config-tenant-contract-subj)# exit`	Returns to the contract configuration mode.
Step 14	**exit** **Example:** `apic1(config-tenant-contract)# exit`	Returns to the tenant configuration mode.
Step 15	**application** *app-name* **Example:** `apic1(config-tenant)# application OnlineStore`	Enters application configuration mode.
Step 16	**epg** *epg-name* **Example:** `apic1(config-tenant-app)# epg exampleCorp_webepg1`	Enters configuration mode for the EPG to be linked to the contract.
Step 17	**bridge-domain member** *bd-name* **Example:** `apic1(config-tenant-app-epg)# bridge-domain member exampleCorp_bd1`	Specifies the bridge domain for this EPG.
Step 18	**contract provider** *provider-contract-name* **Example:** `apic1(config-tenant-app-epg)# contract provider web80`	Specifies the provider contract for this EPG. Communication with this EPG can be initiated from other EPGs as long as the communication complies with this provider contract.
Step 19	**contract consumer** *consumer-contract-name* **Example:** `apic1(config-tenant-app-epg)# contract consumer rmi99`	Specifies the consumer contract for this EPG. The endpoints in this EPG may initiate communication with any endpoint in an EPG that is providing this contract.

Examples

This example shows how to create and apply contracts to an EPG.

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

    # CREATE FILTERS
apic1(config-tenant)# access-list http_acl
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# match tcp dest 443
```

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

    # CREATE CONTRACT WITH FILTERS
apic1(config-tenant)# contract web80
apic1(config-tenant-contract)# subject web80
apic1(config-tenant-contract-subj)# access-group http_acl both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit

    # ASSOCIATE CONTRACTS TO EPG
apic1(config-tenant)# application OnlineStore
apic1(config-tenant-app)# epg exampleCorp_webepg1
apic1(config-tenant-app-epg)# bridge-domain member exampleCorp_bd1
apic1(config-tenant-app-epg)# contract consumer rmi99
apic1(config-tenant-app-epg)# contract provider web80
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)#exit
apic1(config-tenant)#exit

    # ASSOCIATE PORT AND VLAN TO EPG
apic1(config)#leaf 101
apic1(config-leaf)# interface ethernet 1/4
apic1(config-leaf-if)# switchport trunk allowed vlan 102 tenant exampleCorp application
 OnlineStore epg exampleCorp_webepg1
```

This example shows a simpler method for defining a contract by declaring the filters inline in the contract itself.

```
apic1# configure
apic1(config)# tenant exampleCorp
apic1(config-tenant)# contract web80
apic1(config-tenant-contract)# match tcp 80
apic1(config-tenant-contract)# match tcp 443
```

Exporting a Contract to Another Tenant

You can export a contract from one tenant and import it to another. In the tenant that imports the contract, the contract can be applied only as a consumer contract. The contract can be renamed during the export.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1(config)# tenant RedCorp`	Enters the tenant configuration mode for the exporting tenant.
Step 3	**contract** *contract-name* **Example:**	Enters the contract configuration mode for the contract to be exported.

	Command or Action	Purpose
	`apic1(config-tenant)# contract web80`	
Step 4	**scope** {**application** | **exportable** | **tenant** | **vrf**} **Example:** `apic1(config-tenant-contract)# scope exportable`	Configures how the contract can be shared. The scope can be: • **application**—Can be shared among the EPGs of the same application. • **exportable**—Can be shared across tenants. • **tenant**—Can be shared among the EPGs of the same tenant. • **vrf**—Can be shared among the EPGs of the same VRF.
Step 5	**export to tenant** *other-tenant-name* **as** *new-contract-name* **Example:** `apic1(config-tenant-contract)# export to tenant BlueCorp as webContract1`	Exports the contract to the other tenant. You can use the same contract name or you can rename it.
Step 6	**exit** **Example:** `apic1(config-tenant-contract)# exit`	Returns to the tenant configuration mode.
Step 7	**exit** **Example:** `apic1(config-tenant)# exit`	Returns to the global configuration mode.
Step 8	**tenant** *tenant-name* **Example:** `tenant BlueCorp`	Enters the tenant configuration mode for the importing tenant.
Step 9	**application** *app-name* **Example:** `apic1(config-tenant)# application BlueStore`	Enters application configuration mode.
Step 10	**epg** *epg-name* **Example:** `apic1(config-tenant-app)# epg BlueWeb`	Enters configuration mode for the EPG to be linked to the contract.
Step 11	**contract consumer** *consumer-contract-name* **imported** **Example:**	Specifies the imported consumer contract for this EPG. The endpoints in this EPG may initiate communication with any endpoint in an EPG that is providing this contract.

	Command or Action	Purpose
	apic1(config-tenant-app-epg)# **contract consumer webContract1 imported**	

Examples

This example shows how to export a contract from the tenant RedCorp to the tenant BlueCorp, where it will be a consumer contract.

```
apic# configure
apic1(config)# tenant RedCorp
apic1(config-tenant)# contract web80
apic1(config-tenant-contract)# scope exportable
apic1(config-tenant-contract)# export to tenant BlueCorp as webContract1
apic1(config-tenant-contract)# exit
apic1(config-tenant)# exit
apic1(config)# tenant BlueCorp
apic1(config-tenant)# application BlueStore
apic1(config-tenant-application)# epg BlueWeb
apic1(config-tenant-application-epg)# contract consumer webContract1 imported
```

Configuring Contracts Using the REST API

Configuring a Contract Using the REST API

Procedure

Configure a contract using an XML POST request similar to the following example:

Example:

```
<vzBrCP name="webCtrct">
          <vzSubj name="http" revFltPorts="true" provmatchT="All">
              <vzRsSubjFiltAtt tnVzFilterName="Http"/>
              <vzRsSubjGraphAtt graphName="G1" termNodeName="TProv"/>
              <vzProvSubjLbl name="openProv"/>
              <vzConsSubjLbl name="openCons"/>
          </vzSubj>
          <vzSubj name="https" revFltPorts="true" provmatchT="All">
              <vzProvSubjLbl name="secureProv"/>
              <vzConsSubjLbl name="secureCons"/>
              < vzRsSubjFiltAtt tnVzFilterName="Https"/>
              <vzRsOutTermGraphAtt graphName="G2" termNodeName="TProv"/>
          </vzSubj>
      </vzBrCP>
```

Configuring a Taboo Contract Using the REST API

Before you begin

The following objects must be created:

- The tenant that will be associated with this **Taboo Contract**
- An application profile for the tenant
- At least one EPG for the tenant

Procedure

To create a taboo contract with the REST API, use XML such as in the following example:

Example:

```
<vzTaboo ownerTag="" ownerKey="" name="VRF64_Taboo_Contract"
dn="uni/tn-Tenant64/taboo-VRF64_Taboo_Contract" descr=""><vzTSubj
name="EPG_subject" descr=""><vzRsDenyRule tnVzFilterName="default"
directives="log"/>
</vzTSubj>
</vzTaboo>
```

Verifying Contracts, Taboo Contracts, and Filters Using the REST API

This topic provides the REST API XML to verify contracts, taboo contracts, and filters.

Procedure

Step 1 Verify a contract for an EPG or an external network with XML such as the following example for a provider:

Example:

```
QUERY https://apic-ip-address/api/node/class/fvRsProv.xml
```

Step 2 Verify a contract on an EPG with XML such as the following example for a consumer:

Example:

```
QUERY https://apic-ip-address/api/node/class/fvRsCons.xml
```

Step 3 Verify exported contracts using XML such as the following example:

Example:

```
QUERY https://apic-ip-address/api/node/class/vzCPif.xml
```

Step 4 Verify contracts for a VRF with XML such as the following example:

Example:

```
QUERY https://apic-ip-address/api/node/class/vzBrCP.xml
```

Step 5 Verify taboo contracts with XML such as the following example:

Example:

```
QUERY https://apic-ip-address/api/node/class/vzTaboo.xml
```

For taboo contracts for an EPG, use the same query as for contracts for EPGs.

Step 6 Verify filters using XML such as the following example:

Example:

```
QUERY https://apic-ip-address/api/node/class/vzFilter.xml
```

Optimize Contract Performance

Optimize Contract Performance

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

TCAM Optimization is supported on the Cisco Nexus 9000 Series top of rack (TOR) switches with names ending with EX and FX, and later (for example, N9K-C93180LC-EX or N9K-C93180YC-FX).

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

- Mark the contracts to be applied in both directions between the provider and consumer
- For filters with IP TCP or UDP protocols, enable the reverse port option
- When configuring the contract subjects, enable the **no stats** directive.

Limitations

With the no_stats option enabled, per rule statistics are lost. However combined rule statistics for both directions are there in the hardware statistics.

After upgrading to Cisco APIC 3.2(1), to add the `no stats` option to a pre-upgrade contract subject (with filters or filter entries), you must delete the subject and reconfigure it with the `no stats` option. Otherwise, compression does not occur.

For each contract with a bi-directional subject filter, Cisco NX-OS creates 2 rules, one rule with an `sPcTag` and `dPcTag` that is marked `direction=bi-dir`, which is programmed in hardware, and another rule marked with `direction=uni-dir-ignore` which is not programmed.

Rules with the following settings are not compressed:

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

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

```
# actrl.Rule
scopeId            : 2588677
sPcTag             : 16388
dPcTag             : 49156
fltId              : 67
action             : no_stats,permit
actrlCfgFailedBmp  :
actrlCfgFailedTs   : 00:00:00:00.000
actrlCfgState      : 0
childAction        :
descr              :
direction          : bi-dir
dn                 : sys/actrl/scope-2588677/rule-2588677-s-16388-d-49156-f-67
id                 : 4112
lcOwn              : implicit
markDscp           : unspecified
modTs              : 2018-04-27T09:01:33.152-07:00
monPolDn           : uni/tn-common/monepg-default
name               :
nameAlias          :
operSt             : enabled
operStQual         :
prio               : fully_qual
qosGrp             : unspecified
rn                 : rule-2588677-s-16388-d-49156-f-67
status             :
type               : tenant


# actrl.Rule
scopeId            : 2588677
sPcTag             : 49156
dPcTag             : 16388
fltId              : 64
action             : no_stats,permit
actrlCfgFailedBmp  :
actrlCfgFailedTs   : 00:00:00:00.000
actrlCfgState      : 0
childAction        :
descr              :
direction          : uni-dir-ignore
dn                 : sys/actrl/scope-2588677/rule-2588677-s-49156-d-16388-f-64
id                 : 4126
lcOwn              : implicit
markDscp           : unspecified
modTs              : 2018-04-27T09:01:33.152-07:00
monPolDn           : uni/tn-common/monepg-default
name               :
nameAlias          :
operSt             : enabled
operStQual         :
prio               : fully_qual
qosGrp             : unspecified
rn                 : rule-2588677-s-49156-d-16388-f-64
status             :
type               : tenant
```

Table 10: Compression Matrix

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

Configure a Contract with Optimized TCAM Usage Using the GUI

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

Before you begin

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

Procedure

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

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

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

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

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

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

Note This step associates filters with the contract subject.

d) To enable the TCAM-contract usage optimization feature, ensure that **Apply Both Directions** and **Reverse Filter Ports** are enabled.

e) Click the + icon to expand **Filters**.

f) In the dialog box, from the drop-down menu, choose a default filter, a previously configured filter, or **Create Filter**.

g) In the **Directives** field, choose **no stats**

h) In the **Action** field, choose **Permit** or **Deny**.

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

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

j) Click **Update**.

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

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

Configure a Contract with Optimized TCAM Usage Using the REST API

Before you begin

Create the tenant, VRF, and EPGs that will provide and consume the contract.

Procedure

To configure a filter and contract that optimizes TCAM storage of contract data on hardware, send a post with XML similar to the following example:

Example:

```
<vzFilter dn="uni/tn-Tenant64/flt-webFilter" name="webFilter">
    <vzEntry applyToFrag="no" dFromPort="https" dToPort="https"
        dn="uni/tn-Tenant64/flt-webFilter/e-https" etherT="ip" name="https" prot="tcp"
stateful="no"/>
    </vzFilter>
<vzBrCP dn="uni/tn-Tenant64/brc-OptimizedContract" name="OptimizedContract"
provMatchT="AtleastOne" revFltPorts="yes">
   <vzSubj consMatchT="AtleastOne" dn="uni/tn-Tenant64/brc-OptimizedContract/subj-WebSubj"
 lcOwn="local" name="WebSubj"
          provMatchT="AtleastOne" revFltPorts="yes">
         <vzRsSubjFiltAtt action="permit" directives="no_stats" forceResolve="yes"
lcOwn="local" tCl="vzFilter"
          tDn="uni/tn-Tenant64/flt-webFilter" tRn="flt-webFilter" tType="name"
tnVzFilterName="webFilter"/>
    </vzSubj>
</vzBrCP>
```

Contract and Subject Exceptions

Configuring Contract or Subject Exceptions for Contracts

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

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

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

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

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

Exception Types

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

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

Configure a Contract or Subject Exception Using the GUI

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

Before you begin

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

Procedure

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

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

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

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

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

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

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

Step 7 Repeat to add another subject.

Step 8 Click **Submit**

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

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

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

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

Configure a Contract or Subject Exception Using the NX-OS Style CLI

In this task, you configure a contract that will allow most of the EPGs to communicate, but deny access to a subset of them. Multiple exceptions can be added to a contract or a subject.

Before you begin

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

Procedure

Step 1 Configure filters for HTTP and HTTPS, using commands as in the following example:

Example:

```
apic1(config)# tenant t2
apic1(config-tenant)# access-list ac1
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# exit
apic1(config-tenant)# access-list ac2
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 443
```

Step 2 Configure a contract that excludes `EPg01` from consuming it and `EPg03` from providing it.

Example:

```
apic1(config-tenant)# contract webCtrct
apic1(config-tenant-contract)# subject https-subject
apic1(config-tenant-contract-subj)# exception name EPG consumer-regexp EPg01 field EPg
provider-regexp EPg03
apic1(config-tenant-contract-subj)# access-group ac1 in blacklist
apic1(config-tenant-contract-subj)# access-group ac2 in whitelist
```

Configure a Contract or Subject Exception Using the REST API

In this task, you configure a contract that will allow most of the EPGs to communicate, but deny access to a subset of them. Multiple exceptions can be added to a contract or a subject.

Before you begin

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

Procedure

Step 1 Create a filter by sending a post with XML, such as the following example:

Example:

```
<vzFilter name='http-filter'>
    <vzEntry name='http-e' etherT='ip' prot='tcp'/>
    <vzEntry name='https-e' etherT='ip' prot='tcp'/>
</vzFilter>
```

Step 2 Create a contract that excludes `EPg01` from consuming the subject and `EPg03` from providing it, by sending a post with XML, such as the following example:

The vzException MO can be contained by the vzBrCP or vzSubj MOs.

Example:

```
<vzBrCP name="httpCtrct" scope="context">
    <vzSubj name="subj1"
```

```
        <vzException consRegex="EPg01" field="EPg" name="excep01" provRegex=EPg03"/>
    <vzSubj/>
         <vzRsSubjFiltAtt tnVzFilterName="http-filter" Action="deny"/>
         <vzRsSubjFiltAtt tnVzFilterName="https-filter" Action="permit"/>
     </vzSubj>
</vzBrCP>
```

Intra-EPG Contracts

Intra-EPG Contracts

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

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

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

Support

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

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

Intra-EPG contracts require that the leaf switch support proxy Address Resolution Protocol (ARP). They are supported on Cisco Nexus 9000 Series switches with EX or FX at the end of their model name or later models.

Configuring an Intra-EPG Contract Using the GUI

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

Before you begin

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

Procedure

Step 1 Log in to the APIC GUI.

Step 2 Go to **TENANTS** > ***tenant***.

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

If you want to apply an intra-EPG contract to...	Then...
An application EPG	Then... 1. In the left navigation pane, expand **Application Profiles** > ***application profile*** > **Application EPGs** > ***epg***. 2. Right-click the **Contracts** folder and then choose **Add Intra-EPG Contract**. 3. In the **Add Intra-EPG Contract** dialog box, from the **Contract** drop-down list, choose a contract. 4. Click **SUBMIT**.
A uSeg EPG	1. In the left navigation pane, expand **Application Profiles** > ***application profile*** > **uSeg EPGs** > ***epg***. 2. Right-click the **Contracts** folder and then choose **Add Intra-EPG Contract**. 3. In the **Add Intra-EPG Contract** dialog box, from the **Contract** drop-down list, choose a contract. 4. Click **SUBMIT**.

Configuring an Intra-EPG Contract Using the NX-OS Style CLI

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

Before you begin

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

Procedure

Step 1 In the NX-OS CLI, start in Configuration mode.

Example:

```
apic #
apic # configure
```

Step 2 Choose the tenant.

Example:

```
apic (config) # tenant t001
```

Step 3 Choose the application profile.

Example:

```
apic (config-tenant) application ap3
```

Step 4 Choose the EPG.

Example:

```
apic (config-tenant-app) epg ep3
```

Step 5 Configure an intra-EPG contract for the EPG.

Example:

```
apic (config-tenant-app-epg) contract intra-epg ct1
```

Configuring an Intra-EPG Contract Using the REST API

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

Before you begin

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

Procedure

Configure an intra-EPG contract using an XML POST request similar to the following example:

Example:

```
<fvTenant name="t001">
  <fvAp name="ap3">
    <fvAEPg name="ep3">
      <fvRsIntraEpg tnVzBrCPName="ct1"/>
    </fvAEPg>
  </fvAp>
</fvTenant>
```

EPG Contract Inheritance

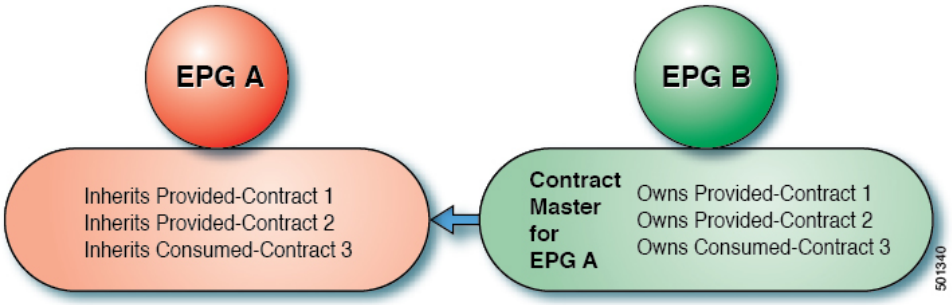
About Contract Inheritance

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

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

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

Figure 9: Contract Inheritance



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

Use the following guidelines when configuring contract inheritance:

- Contract inheritance can be configured for application, microsegmented (uSeg), external L2Out EPGs, and external L3Out EPGs. The relationships must be between EPGs of the same type.
- Both provided and consumed contracts are inherited from the contract master when the relationship is established.
- Contract masters and the EPGs inheriting contracts must be within the same tenant.
- Changes to the masters' contracts are propagated to all the inheritors. If a new contract is added to the master, it is also added to the inheritors.
- An EPG can inherit contracts from multiple contract masters.
- Contract inheritance is only supported to a single level (cannot be chained) and a contract master cannot inherit contracts.
- Contract subject label and EPG label inheritance is supported. When EPG A inherits a contract from EPG B, if different subject labels are configured under EPG A and EPG B, APIC only uses the subject label configured under EPG B and not a collection of labels from both EPGs.

- Whether an EPG is directly associated to a contract or inherits a contract, it consumes entries in TCAM. So contract scale guidelines still apply. For more information, see the *Verified Scalability Guide* for your release.
- vzAny security contracts and taboo contracts are not supported.

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

Configuring EPG Contract Inheritance Using the GUI

Configuring Application EPG Contract Inheritance Using the GUI

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

Before you begin

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

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

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

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

Procedure

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

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

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

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

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

Step 6 Click **Finish**.

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

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

Configuring uSeg EPG Contract Inheritance Using the GUI

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

Before you begin

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

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

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

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

Procedure

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

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

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

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

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

Step 6 Click **Finish**.

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

Configuring L2Out EPG Contract Inheritance Using the GUI

To configure contract inheritance for an external L2Out EPG, in the APIC Advanced mode GUI, use the following steps.

Before you begin

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

Configure an external bridged network (L2Out) and the External Network Instance Profile (L2extInstP) that will serve as the **L2Out Contract Master**.

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

Procedure

Step 1 To configure contract inheritance for an external L2Out EPG, navigate to **Tenants** > ***tenant-name*** > **Networking** > **External Bridged Networks**, and perform the following steps:

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

Step 3 Right-click **Networks** and select **Create External Network.**

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

Step 5 Click **Submit**.

Step 6 Expand **Networks**.

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

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

Step 9 Select the L2Out and the L2Out Contract Master for this external L2Out EPG.

Step 10 Click **Update**.

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

Configuring External L3Out EPG Contract Inheritance Using the Advanced GUI

To configure contract inheritance for an external L3Out EPG, in the APIC Advanced mode GUI, use the following steps.

Before you begin

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

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

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

Procedure

Step 1 To configure contract inheritance for an external L3Out EPG, navigate to **Tenants** > ***tenant-name*** > **Networking** > **External Routed Networks**, and perform the following steps:

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

Step 3 Right-click **Networks** and select **Create External Network.**

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

Step 5 Click **Submit**.

Step 6 Expand **Networks**.

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

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

Step 9 Select the L3Out and Interface Profile to serve as L3Out contract master for this external L3Out EPG.

Step 10 Click **Update**.

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

Configuring Contract Inheritance Using the NX-OS Style CLI

Configuring Application or uSeg EPG Contract Inheritance Using the NX-OS Style CLI

To configure contract inheritance for application or uSeg EPGs, use the following commands:

Before you begin

Configure the tenant, application profile, and bridge-domain to be used by the EPGs.

Configure the contracts to be shared by the EPGs at the VRF level.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1# (config) tenant Tn1`	Creates or specifies the tenant to be configured; and enters into tenant configuration mode.
Step 3	**application** *application-name* **Example:** `apic1(config-tenant)# application AP1`	Creates or specifies an application and enters into application mode.
Step 4	**epg** *epg-name* [**type micro-segmented**] **Example:** `apic1(config-tenant-app)# epg AEPg403`	Creates or specifies the application or uSeg EPG to be configured and enters into EPG configuration mode. For uSeg EPGs add the type. In this example, this is the application EPG contract master.
Step 5	**bridge-domain member** *bd-name* **Example:** `apic1(config-tenant-app-epg)# bridge-domain member T1BD1`	Associates the EPG with the bridge domain.
Step 6	**contract consumer** *contract-name* **Example:** `apic1(config-tenant-app-epg)# contract consumer cctr5`	Adds a contract to be consumed by this EPG.
Step 7	**contract provider** [**label** *label*] **Example:** `apic1(config-tenant-app-epg)# contract provider T1ctrl_cif`	Adds a contract to be provided by this EPG, including an optional list of subject or EPG labels (must be previously configured).
Step 8	**exit** **Example:** `apic1(config-tenant-app-epg)# exit`	Exits the configuration mode

	Command or Action	Purpose
Step 9	**epg** *epg-name* [**type micro-segmented**] **Example:** `apic1(config-tenant-app)# epg AEPg404`	Creates or specifies the application or uSeg EPG to be configured and enters into EPG configuration mode. For uSeg EPGs add the type. In this example, this is the EPG inheriting contracts.
Step 10	**bridge-domain member** *bd-name* **Example:** `apic1(config-tenant-app-epg)# bridge-domain member T1BD1`	Associates the EPG with the bridge domain.
Step 11	**inherit-from-epg application** *application-name***epg** *EPG-contract-master-name*] **Example:** `apic1(config-tenant-app-epg)# inherit-from-epg application AP1 epg AEPg403`	Configures this EPG to inherit contracts from the EPG contract master.
Step 12	**exit** **Example:** `apic1(config-tenant-app-epg)# exit`	Exits the configuration mode
Step 13	**epg** *epg-name* [**type micro-segmented**] **Example:** `apic1(config-tenant-app)# epg uSeg1_403_10 type micro-segmented`	Creates or specifies the application or uSeg EPG to be configured and enters into EPG configuration mode. In this example, this is the uSeg EPG contract master.
Step 14	**bridge-domain member** *bd-name* **Example:** `apic1(config-tenant-app-epg)# bridge-domain member T1BD1`	Associates the EPG with the bridge domain.
Step 15	**contract provider** [**label** *label*] **Example:** `apic1(config-tenant-app-epg)# contract provider T1ctrl_uSeg_l3out`	Adds a contract to be provided by this EPG, including an optional list of subject or EPG labels (must be previously configured).
Step 16	**attribute-logical-expression** *logical-expression* **Example:** `apic1(config-tenant-app-epg)# attribute-logical-expression 'ip equals 192.168.103.10 force'`	Adds a logical expression to the uSeg EPG as matching criteria.

	Command or Action	Purpose
Step 17	**exit** **Example:** `apic1(config-tenant-app-epg)# exit`	Exits the configuration mode
Step 18	**epg** *epg-name* [**type micro-segmented**] **Example:** `apic1(config-tenant-app)# epg uSeg1_403_30 type micro-segmented`	Creates or specifies the application or uSeg EPG to be configured and enters into EPG configuration mode. In this example, this is the uSeg EPG that inherits contracts from the EPG contract master.
Step 19	**bridge-domain member** *bd-name* **Example:** `apic1(config-tenant-app-epg)# bridge-domain member T1BD1`	Associates the EPG with the bridge domain.
Step 20	**attribute-logical-expression** *logical-expression* **Example:** `apic1(config-tenant-app-epg)# attribute-logical-expression 'ip equals 192.168.103.30 force'`	Adds a logical expression to the uSeg EPG as criteria.
Step 21	**inherit-from-epg application** *application-name* **epg** *EPG-contract-master-name* **Example:** `apic1(config-tenant-app-epg)# inherit-from-epg application AP1 epg uSeg1_403_10`	Configures this EPG to inherit contracts from the EPG contract master.
Step 22	**exit** **Example:** `apic1(config-tenant-app-epg)# exit`	Exits the configuration mode
Step 23	**exit** **Example:** `apic1(config-tenant-app)# exit`	Exits the configuration mode
Step 24	**exit** **Example:** `apic1(config-tenant)# exit`	Exits the configuration mode
Step 25	**exit** **Example:** `apic1(config)# exit`	Exits the configuration mode

Example

```
ifav90-ifc1# show running-config tenant Tn1 application AP1
# Command: show running-config tenant Tn1 application AP1
# Time: Fri Apr 28 17:28:32 2017
  tenant Tn1
    application AP1
      epg AEPg403
        bridge-domain member T1BD1
        contract consumer cctr5 imported
        contract provider T1ctr1_cif
        exit
      epg AEPg404
        bridge-domain member T1BD1
        inherit-from-epg application AP1 epg AEPg403
        exit
      epg uSeg1_403_10 type micro-segmented
        bridge-domain member T1BD1
        contract provider T1Ctr1_uSeg_l3out
        attribute-logical-expression 'ip equals 192.168.103.10 force'
        exit
      epg uSeg1_403_30 type micro-segmented
        bridge-domain member T1BD1
        attribute-logical-expression 'ip equals 192.168.103.30 force'
        inherit-from-epg application AP1 epg uSeg1_403_10
        exit
      exit
    exit
```

Configuring L2Out EPG Contract Inheritance Using the NX-OS Style CLI

To configure contract inheritance for an external L2Out EPG, use the following commands:

Before you begin

Configure the tenant, VRF, and bridge-domain to be used by the EPGs.

Configure the Layer 2 outside network (L2Out) that the EPGs will use.

Configure the contracts to be shared by the EPGs, at the VRF level.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1(config)# tenant Tn1`	Creates or specifies the tenant to be configured; and enters into tenant configuration mode.

	Command or Action	Purpose
Step 3	**external-l2 epg** *external-l2-epg-name* **Example:** `apic1(config-tenant)# external-l2 epg l2out1:l2Ext1`	Configures or specifies an external L2Out EPG. In this example, this is the L2out contract master.
Step 4	**bridge-domain member** *bd-name* **Example:** `apic1(config-tenant-l2ext-epg)# bridge-domain member T1BD1`	Associates the L2Out EPG with a bridge domain.
Step 5	**contract provider** *contract-name* [**label** *label*] **Example:** `apic1(config-tenant-l2ext-epg)# contract provider T1ctr_tcp`	Adds a contract to be provided by this EPG.
Step 6	**exit** **Example:** `apic1(config-tenant-l2ext-epg)# exit`	Exits the configuration mode
Step 7	**external-l2 epg** *external-l2-epg-name* **Example:** `apic1(config-tenant)# external-l2 epg L2out12:l2Ext12`	Configures an external L2Out EPG. In this example, this is the EPG that inherits contracts from the L2out contract master.
Step 8	**bridge-domain member** *bd-name* **Example:** `apic1(config-tenant-l2ext-epg)# bridge-domain member T1BD1`	Associates the L2out EPG with the bridge domain.
Step 9	**inherit-from-epg** *L2Out-contract-master-name* **Example:** `apic1(config-tenant-l2ext-epg)# inherit-from-epg epg l2out1:l2Ext1`	Configures this EPG to inherit contracts from the L2Out contract master.
Step 10	**exit** **Example:** `apic1(config-tenant-l2ext-epg)# exit`	Exits the configuration mode

Example

The steps above are taken from the following example:

```
apic1# show running-config tenant Tn1 external-l2
# Command: show running-config tenant Tn1 external-l2
# Time: Thu May 11 13:10:14 2017
```

```
tenant Tn1
  external-l2 epg l2out1:l2Ext1
    bridge-domain member T1BD1
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out10:l2Ext10
    bridge-domain member T1BD10
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out11:l2Ext11
    bridge-domain member T1BD11
    contract provider T1ctr_udp
    exit
  external-l2 epg l2out12:l2Ext12
    bridge-domain member T1BD12
    inherit-from-epg epg l2out1:l2Ext1
    inherit-from-epg epg l2out10:l2Ext10
    inherit-from-epg epg l2out11:l2Ext11
    inherit-from-epg epg l2out2:l2Ext2
    inherit-from-epg epg l2out3:l2Ext3
    inherit-from-epg epg l2out4:l2Ext4
    inherit-from-epg epg l2out5:l2Ext5
    inherit-from-epg epg l2out6:l2Ext6
    inherit-from-epg epg l2out7:l2Ext7
    inherit-from-epg epg l2out8:l2Ext8
    inherit-from-epg epg l2out9:l2Ext9
    exit
  external-l2 epg l2out2:l2Ext2
    bridge-domain member T1BD2
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out3:l2Ext3
    bridge-domain member T1BD3
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out4:l2Ext4
    bridge-domain member T1BD4
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out5:l2Ext5
    bridge-domain member T1BD5
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out6:l2Ext6
    bridge-domain member T1BD6
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out7:l2Ext7
    bridge-domain member T1BD7
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out8:l2Ext8
    bridge-domain member T1BD8
    contract provider T1ctr_tcp
    exit
  external-l2 epg l2out9:l2Ext9
    bridge-domain member T1BD9
    contract provider T1ctr_tcp
    exit
  exit
```

Configuring External L3Out EPG Contract Inheritance Using the NX-OS Style CLI

To configure contract inheritance for an external L3Out EPG, use the following commands:

Before you begin

Configure the tenant, VRF, and bridge-domain to be used by the EPGs.

Configure the Layer 3 outside network (L3Out) that the EPGs will use.

Configure the contracts to be shared by the EPGs, at the VRF level.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.
Step 2	**tenant** *tenant-name* **Example:** `apic1(config)# tenant Tn1`	Creates or specifies the tenant to be configured; and enters into tenant configuration mode.
Step 3	**external-l3 epg** *external-l3-epg-name* **l3out** *l3out-name* **Example:** `apic1(config-tenant-app)# external-l3 epg l3Ext108 l3out T1L3out1`	Configures an external L3Out EPG. In this example, this is the L3out contract master.
Step 4	**vrf member** *vrf-name* **Example:** `apic1(tenant-l3out)# vrf member T1ctx1`	Associates the L3out with the VRF.
Step 5	**match ip** *ip-address-and-mask* **Example:** `apic1(config-tenant-l3ext-epg)# match ip 192.168.110.0/24 shared`	Adds a subnet that identifies hosts as part of the EPG and adds the optional shared scope for the subnet.
Step 6	**contract provider** *contract-name* [**label** *label*] **Example:** `apic1(config-tenant-l3ext-epg)# contract provider T1ctrl-L3out`	Adds a contract to be provided by this EPG.
Step 7	**exit** **Example:** `apic1(config-tenant-l3ext-epg)# exit`	Exits the configuration mode

	Command or Action	Purpose
Step 8	**external-l3 epg** *external-l3-epg-name* **l3out** *l3out-name* **Example:** `apic1(config-tenant-app)# external-l3 epg l3Ext110 l3out T1L3out1`	Configures an external L3Out EPG. In this example, this is the EPG that inherits contracts from the L3out contract master.
Step 9	**vrf member** *vrf-name* **Example:** `apic1(tenant-l3out)# vrf member T1ctx1`	Associates the L3out with the VRF.
Step 10	**match ip** *ip-address-and-mask* **Example:** `apic1(config-tenant-l3ext-epg)# match ip 192.168.112.0/24 shared`	Adds a subnet that identifies hosts as part of the EPG and adds the optional shared scope for the subnet.
Step 11	**inherit-from-epg** *L3Out-contract-master-name* **Example:** `apic1(config-tenant-l3ext-epg)# inherit-from-epg l3Ext108`	Configures this EPG to inherit contracts from the L3Out contract master.
Step 12	**exit** **Example:** `apic1(config-tenant-l3ext-epg)# exit`	Exits the configuration mode

Example

```
ifav90-ifc1# show running-config tenant Tn1 external-l3 epg l3Ext110
# Command: show running-config tenant Tn1 external-l3 epg l3Ext110
# Time: Fri Apr 28 17:36:15 2017
  tenant Tn1
    external-l3 epg l3Ext108 l3out T1L3out1
      vrf member T1ctx1
      match ip 192.168.110.0/24 shared
      contract provider T1ctrl-L3out
      exit
    external-l3 epg l3Ext110 l3out T1L3out1
      vrf member T1ctx1
      match ip 192.168.112.0/24 shared
      inherit-from-epg epg l3Ext108
      exit
    exit
```

Configuring EPG Contract Inheritance Using the REST API

Configuring Application EPG Contract Inheritance Using the REST API

Before you begin

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

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

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

Procedure

To configure contract inheritance using the REST API, send a post with XML such as the following XML and JSON examples, with a URL directed to the EPG that will inherit the contracts:

Example:

XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- /api/node/mo/uni/tn-coke/ap-AP/epg-EPg_B.xml -->
<polUni>

  <fvEPg>

    <fvRsSecInherited tDn="uni/tn-coke/ap-AP/epg-EPg_B/>
  </fvEPg>
</polUni>
```

JSON Example

```
https://192.168.200.10/api/node/mo/uni/tn-coke/ap-AP/epg-EPg_B.json
fvAEPg":{"attributes":{
   "dn":"uni/tn-coke/ap-AP/epg-EPg_B","name":"EPg_C",
   "rn":"epg-EPg_C",
   "status":"created"},
   "children":[{
     "fvRsBd":{"attributes":{
        "tnFvBDName":"default",
        "status":"created,modified"},
        "children":[]}},{
     "fvRsSecInherited":{"attributes":{
        "tDn":"uni/tn-coke/ap-AP/epg-EPg_B",
        "status":"created"},
        "children":[]}}]}}
```

Configuring uSeg EPG Contract Inheritance Using the REST API

Before you begin

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

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

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

Procedure

To configure uSeg contract inheritance using the REST API, send a post with XML such as the following example:

Example:

```
<polUni>
    <fvTenant name="Tn1" >
        <fvAEPg descr="" dn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_120" fwdCtrl=""
isAttrBasedEPg="yes" matchT="AtleastOne" name="uSeg1_301_120" pcEnfPref="unenforced"
prefGrMemb="exclude" prio="unspecified">
            <fvRsSecInherited tDn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_100" />
            <fvRsSecInherited tDn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_110" />
            <fvRsSecInherited tDn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_50" />
            <fvRsSecInherited tDn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_60" />
            <fvRsSecInherited tDn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_30" />
            <fvRsSecInherited tDn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_10" />
            <fvRsSecInherited tDn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_40" />
            <fvRsSecInherited tDn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_70" />
            <fvRsSecInherited tDn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_90" />
            <fvRsSecInherited tDn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_20" />
            <fvRsSecInherited tDn="uni/tn-Tn1/ap-AP1/epg-uSeg1_301_80" />
            <fvRsNodeAtt descr="" encap="unknown" instrImedcy="immediate" mode="regular"
tDn="topology/pod-1/node-108" />
            <fvRsNodeAtt descr="" encap="unknown" instrImedcy="immediate" mode="regular"
tDn="topology/pod-1/node-109" />
            <fvRsDomAtt classPref="encap" delimiter="" encap="vlan-301" encapMode="auto"
instrImedcy="immediate" netflowPref="disabled" primaryEncap="unknown" resImedcy="immediate"
 tDn="uni/phys-PhysDom1" />
            <fvRsCustQosPol tnQosCustomPolName="" />
            <fvRsBd tnFvBDName="T1BD21" />
          <fvCrtrn descr="" match="any" name="default" nameAlias="" ownerKey="" ownerTag=""
 prec="0">
                <fvIpAttr descr="" ip="192.14.1.120" name="0" nameAlias="" ownerKey=""
ownerTag="" usefvSubnet="no" />
            </fvCrtrn>
        </fvAEPg>
    </fvTenant>
</polUni>
```

What to do next

Configuring L2Out EPG Contract Inheritance Using the REST API

Before you begin

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

Configure the L2Out EPG, to serve as the **L2Out Contract Master**.

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

Procedure

To configure L2Out EPG contract inheritance using the REST API, send a post with XML such as the following example:

Example:

```
<polUni>
  <fvTenant name="Tn1" >
     <l2extOut name="l2out1">
        <l2extRsEBd encap="vlan-51" tnFvBDName="T1BD1" />
        <l2extRsL2DomAtt tDn="uni/l2dom-l2Dom1" />
        <l2extLNodeP name="default" >
           <l2extLIfP name="default" >
              <l2extRsPathL2OutAtt tDn="topology/pod-1/protpaths-108-109/pathep-[VPC83]"
/>
           </l2extLIfP>
        </l2extLNodeP>
        <l2extInstP matchT="AtleastOne" name="l2Ext1">
           <fvSubnet ctrl="nd" ip="192.13.1.10/24" preferred="no" scope="public,shared"
virtual="no" />
           <fvRsProv tnVzBrCPName="T1ctr_tcp" />
        </l2extInstP>
     </l2extOut>

     <l2extOut name="l2out2">
        <l2extRsEBd encap="vlan-53" tnFvBDName="T1BD3" />
        <l2extRsL2DomAtt tDn="uni/l2dom-l2Dom1" />
        <l2extLNodeP name="default" >
           <l2extLIfP name="default" >
              <l2extRsPathL2OutAtt tDn="topology/pod-1/protpaths-108-109/pathep-[VPC84]"
/>
           </l2extLIfP>
        </l2extLNodeP>
        <l2extInstP matchT="AtleastOne" name="l2Ext3" prefGrMemb="exclude">
           <fvSubnet ctrl="nd" ip="192.13.2.10/24" preferred="no" scope="public,shared"
virtual="no" />
           <fvRsSecInherited tDn="uni/tn-Tn1/l2out-l2out1/instP-l2Ext1" />
        </l2extInstP>
     </l2extOut>

  </fvTenant>
</polUni>
```

Configuring L3Out EPG Contract Inheritance Using the REST API

Before you begin

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

Configure the L3Out EPG, to serve as the **L3Out Contract Master**.

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

Procedure

To configure L3Out EPG contract inheritance using the REST API, send a post with XML such as the following example:

Example:

```
<polUni>
  <fvTenant name="Tn6" >

  <!— L3out creation -->
     <ospfIfPol deadIntvl="40" helloIntvl="10" name="ospf1" pfxSuppress="inherit" prio="1"
 rexmitIntvl="5" xmitDelay="1" />
     <l3extOut enforceRtctrl="export" name="T6L3out821">
        <ospfExtP areaCost="1" areaCtrl="redistribute,summary" areaId="0.0.0.1"
areaType="regular" />
        <l3extRsL3DomAtt tDn="uni/l3dom-L3Dom1" />
        <l3extRsEctx tnFvCtxName="T6ctx21" />
        <l3extLNodeP name="l3out_vpc82_prof" >
           <l3extRsNodeL3OutAtt rtrId="1.1.1.8" rtrIdLoopBack="yes"
tDn="topology/pod-1/node-108">
              <l3extInfraNodeP fabricExtCtrlPeering="no" />
           </l3extRsNodeL3OutAtt>
           <l3extRsNodeL3OutAtt rtrId="1.1.1.9" rtrIdLoopBack="yes"
tDn="topology/pod-1/node-109">
              <l3extInfraNodeP fabricExtCtrlPeering="no" />
           </l3extRsNodeL3OutAtt>
           <l3extLIfP name="ospf1" >
              <ospfIfP authKeyId="1" authType="none" >
                 <ospfRsIfPol tnOspfIfPolName="ospf1" />
              </ospfIfP>
              <l3extRsPathL3OutAtt encap="vlan-551" ifInstT="ext-svi" mode="regular"
mtu="1500" tDn="topology/pod-1/protpaths-108-109/pathep-[VPC82]" >
                 <l3extMember addr="192.16.51.1/24" llAddr="0.0.0.0" side="B" />
                 <l3extMember addr="192.16.51.2/24" llAddr="0.0.0.0" side="A" />
              </l3extRsPathL3OutAtt>
              <l3extRsNdIfPol tnNdIfPolName="" />
           </l3extLIfP>
        </l3extLNodeP>

        <l3extInstP matchT="AtleastOne" name="T6l3Ext821">
              <fvRsProv tnVzBrCPName="T6ctr_UDP_TCP2" />
              <fvRsCons tnVzBrCPName="T6ctr_UDP_TCP1" />
           <l3extSubnet ip="192.16.51.0/24"
scope="import-security,shared-rtctrl,shared-security" />
              <l3extSubnet ip="192.16.61.0/24"
scope="import-security,shared-rtctrl,shared-security" />
              <vzConsSubjLbl name="tcp" tag="green" />
           <vzProvSubjLbl name="tcp" tag="green" />
        </l3extInstP>

        <l3extInstP matchT="AtleastOne" name="T6l3Ext823">
           <fvRsSecInherited tDn="uni/tn-Tn6/out-T6L3out821/instP-T6l3Ext821" />
              <l3extSubnet ip="192.16.63.0/24"
scope="import-security,shared-rtctrl,shared-security" />
        </l3extInstP>
     </l3extOut>

  </fvTenant>
```

```
</polUni>
```

Contract Preferred Groups

About Contract Preferred Groups

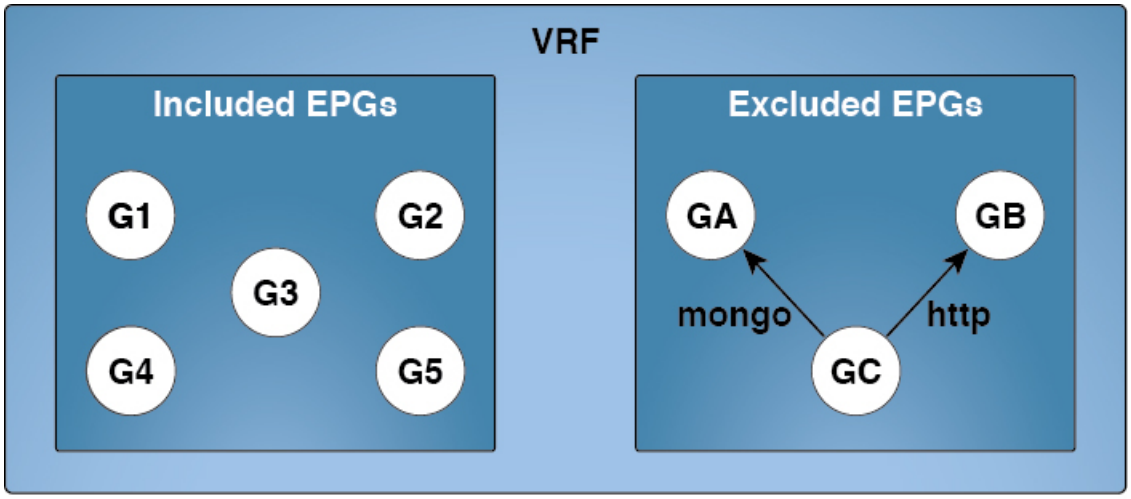
There are two types of policy enforcements available for EPGs in a VRF with a contract preferred group configured:

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

The contract preferred group feature enables greater control of communication between EPGs in a VRF. If most of the EPGs in the VRF should have open communication, but a few should only have limited communication with the other EPGs, you can configure a combination of a contract preferred group and contracts with filters to control inter-EPG communication precisely.

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

Figure 10: Contract Preferred Group Overview



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

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

Limitations

The following limitations apply to contract preferred groups:

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

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

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

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

Guidelines for Contract Preferred Groups

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

- Contract Preferred Group-included EPGs are not supported with a 0/0 prefix in external EPG (InstP). If, for the external EPG (InstP) to Tenant EPG, a 0/0 prefix is required with the use of Contract Preferred Group, then 0/0 can be split to 0/1 and 128/1.
- When configuring policy enforcement between external EPGs (transit routing case), you must configure the second external EPG (InstP) with the default prefix 0/0 for export route control, aggregate export, and external security. In addition, the preferred group must be excluded, and you must use an any any contract (or desired contract) between the transit InstPs.
- Contract Preferred Group-EPGs are not supported with the GOLF feature. Communication between an application EPG and the L3Out EPG for GOLF must be governed by explicit contracts.

Configuring Contract Preferred Groups Using the GUI

Before you begin

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

Procedure

Step 1 On the menu bar, click **Tenants** > ***tenant-name***.

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

Step 3 Expand the VRF for which you are configuring the contract preferred group, and click **`EPG Collection for VRF`**.

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

Step 5 Click **`Submit`**.

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

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

Step 8 In the **Preferred Group Member** field, click **`Include`**.

Step 9 Click **`Submit`**.

What to do next

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

Configuring Contract Preferred Groups Using the NX-OS Style CLI

You can use the APIC NX-OS style CLI to configure a contract preferred group. In this example, a contract preferred group is configured for a VRF. One of the EPGs using the VRF is included in the preferred group.

Before you begin

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

Procedure

	Command or Action	Purpose

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

Step 2	**tenant***tenant-name* **Example:** `apic1(config)# tenant tenant64`	Creates a tenant or enters into tenant configuration mode
Step 3	**vrf context** *vrf-name* **Example:** `apic1(config-tenant)# vrf context vrf64`	Creates a VRF or enters into VRF configuration mode.

Step 4	**whitelist-blacklist-mix**
	Example:
	`apic1(config-tenant-vrf)# whitelist-blacklist-mix`
	`apic1(config-tenant-vrf)# exit`
		Enables a contract preferred group for the VRF and then returns to tenant configuration mode.

Step 5	**bridge-domain** *bd-name* **Example:** `apic1(config-tenant)# bridge-domain bd64`	Creates a bridge-domain for the VRF or enters into BD configuration mode.

Step 6	**vrf member** *vrf-name*
	Example:
	`apic1(config-tenant-bd)# vrf member vrf64`
	`apic1(config-tenant-bd)# exit`
		Associates the VRF with the bridge-domain and returns to teanant configuration mode.

Step 7	**application** *app-name* **Example:** `apic1(config-tenant)# application app-ldap`	Creates an application or enters into application configuration mode.
Step 8	**epg** *epg-name* **Example:**	Creates an EPG or enters into EPG tenant-app EPG configuration mode.

	Command or Action	Purpose
	`apic1(config-tenant-app)# epg epg-ldap`	
Step 9	**bridge-domain member** *bd-name* **Example:** `apic1(config-tenant-app-epg)# bridge-domain member bd64`	Associates the EPG with the bridge-domain .
Step 10	**vrf-blacklist-mode** **Example:** `apic1(config-tenant-app-epg)# vrf-blacklist-mode`	Configures this EPG to be included in the contract preferred group.

Example

The following example creates a contract preferred group for `vrf64` and includes `epg-ldap` in it.

```
apic1# configure
apic1(config)# tenant tenant64
apic1(config-tenant)# vrf context vrf64
apic1(config-tenant-vrf)# whitelist-blacklist-mix
apic1(config-tenant-vrf)# exit

apic1(config-tenant)# bridge-domain bd64
apic1(config-tenant-bd)# vrf member vrf64
apic1(config-tenant-bd)# exit

apic1(config-tenant)# application app-ldap
apic1(config-tenant-app)# epg epg-ldap
apic1(config-tenant-app-epg)# bridge-domain member bd64
apic1(config-tenant-app-epg)# vrf-blacklist-mode
```

Configuring Contract Preferred Groups Using the REST API

The following example creates a contract preferred group in `vrf64`, and creates three EPGs in the VRF:

- `epg-ldap`—Included in the preferred group
- `mail`—Included in the preferred group
- `radius`—Excluded from the preferred group

Before you begin

Create the tenants, VRFs, and the EPGs in the VRF.

Procedure

Create a contract preferred group by sending a post, with XML such as the example:

Example:

```
<polUni>
  <fvTenant name="tenant64">
    <fvCtx name="vrf64"> <vzAny prefGrMemb="enabled"/> </fvCtx>
    <fvBD name="bd64"> <fvRsCtx tnFvCtxName="vrf64"/> </fvBD>
    <fvAp name="app-lldp">
      <fvAEPg name="epg-ldap" prefGrMemb="include">
        <fvRsBd tnFvBDName="bd64"/>
        <fvRsPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/3]" encap="vlan-113"
instrImedcy="immediate"/>
      </fvAEPg>
      <fvAEPg name="mail" prefGrMemb="include">
        <fvRsBd tnFvBDName="bd64"/>
        <fvRsPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/4]" encap="vlan-114"
instrImedcy="immediate"/>
      </fvAEPg>
      <fvAEPg name="radius" prefGrMemb="exclude">
        <fvRsBd tnFvBDName="bd64"/>
        <fvRsPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/5]" encap="vlan-115"
instrImedcy="immediate"/>
      </fvAEPg>
    </fvAp>
  </fvTenant>
</polUni>
```

What to do next

Create a contract governing the communication of the `radius` EPG with other EPGs.

Contracts with Permit and Deny Rules

About Contracts with Permit and Deny Rules

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

Rule conflicts are resolved as follows:

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

- Deny rules with medium priority are handled at the same level as vzAny-to-EPG contracts.
- Deny rules with the lowest priority are handled at the same level as vzAny-to-vzAny contracts.
- If the deny priority is lowered in a contract between EPGs, a permit rule match between EPGs would win over deny.

INDEX