Cisco APIC configuration

Configure the recommended Cisco Application Policy Infrastructure Controller settings

We recommend the you enable certain Cisco Application Policy Infrastructure Controller (APIC) settings.

Enforce Subnet Check

The enforce subnet check feature enforces subnet checks at the VRF instance level, which is when Cisco Application Centric Infrastructure (ACI) learns the IP address as an endpoint from the data plane. Although the subnet check scope is the VRF instance, you can enable and disable this feature only globally under the fabric-wide setting policy. If you enable 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.

Guidelines and limitations for the enforce subnet check

You cannot enable this option in only a single VRF instance. If you enable the option, it gets enabled in all VRF instances.

Enable the enforce subnet check

1. Navigate to **System > System Settings > Fabric Wide Setting.**
2. Put a check in the **Enforce Subnet Check** box.

IP aging policy

The IP aging policy tracks and ages unused IP addresses on an endpoint. The Cisco APIC performs tracking 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 the Cisco APIC does not receive a response from an IP address, that IP address is aged out.

Enable the IP aging policy

1. Navigate to **System > System Settings > Endpoint Controls**.
2. Click the **IP Aging** tab.
3. For **Administrative State**, choose **Enabled**.

Mis-cabling protocol

The mis-cabling protocol (MCP) detects loops that can be caused by various issues, such as misconfiguration, that the link layer discovery protocol (LLDP) and spanning tree protocol (STP) cannot discover. This option enables MCP to send packets on a per-EPG basis.

Enable the mis-cabling protocol

1. Navigate to **Fabric > Access Policies > Policies > Global > MCP Instance Policy default**.
2. For **Admin State**, choose **Enabled**.
3. Put a check in the **Enable MCP PDU per VLAN** box.

Cisco Application Policy Infrastructure Controller interfaces

You can access or configure all functions of Cisco Application Policy Infrastructure Controller (APIC) through the application programming interface (API) by using the following interfaces:

GUI

The Cisco APIC GUI is a browser-based graphical interface to the Cisco APIC that communicates internally with the Cisco APIC engine by exchanging REST API messages. Use the GUI for large-scale configurations, deployments, and operations. The GUI 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.

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

NX-OS-style CLI

The NX-OS style command-line interface (CLI) can be used for Cisco APIC configuration, deployment, and operation. The CLI is organized in a hierarchy of command modes with the EXEC mode as the root and contains a tree of configuration sub-modes that begin with the 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 Cisco APIC GUI to configure Cisco APIC, see Limitations with mixing the user interfaces, on page 3.

REST API

The REST API accepts configuration changes and provides access to management functions for the controller. This interface is a crucial component for the GUI and CLI, and provides a touch point for automation tools, provisioning scripts, and third-party monitoring and management tools.

The Cisco 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 managed object descriptions.

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

Limitations with mixing the user interfaces

Because Cisco 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. For example, configurations done through the NX-OS-style CLI are rendered in the Cisco APIC GUI. They can be seen, but sometimes might not be editable in the GUI. Similarly, changes made in the Cisco APIC GUI can be seen in the NX-OS-style CLI, but might only partially work.

Limitation with mixing the NX-OS-style CLI and the Cisco APIC GUI when you configure per-interface

When you configure per-interface, configurations you perform in the Cisco APIC GUI might only partially work in the NX-OS-style CLI.

For example, say that 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**. When 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.

Limitation with mixing the user interfaces for the Layer 3 external connectivity configuration modes

This section describes considerations for configuring Layer 3 external connectivity with the NX-OS style CLI when you may also be using other 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 that is not compatible with the Cisco APIC GUI nor the REST API.
- **Named (or Explicit) mode:** Compatible with the Cisco APIC GUI and the REST API.

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

Differences between the Layer 3 external connectivity configuration modes

In both modes, the configuration settings are defined within an internal container object, the "Layer 3 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:** You provide the naming of the container. CLI commands in the named mode have an additional l3Out field. To configure the named L3Out correctly and avoid faults, you must 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 limitations for the Layer 3 external connectivity configuration modes

The following guidelines and limitations apply to the Layer 3 external connectivity configuration modes:

Using implicit mode and named mode together

In the same Cisco APIC instance, you can use both modes together for configuring Layer 3 external connectivity. However, the Layer 3 external connectivity configuration for a given combination of tenant, VRF instance, and leaf switch can be done only through one mode.

Use one mode for a tenant VRF instance that is deployed for Layer 3 external connectivity

For a given tenant VRF instance, the policy domain where the external 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 instance combination across all the nodes where the given tenant VRF instance is deployed for Layer 3 external connectivity. The modes can be different across different tenants or different VRF instances and no restrictions apply.

Configurations might be validated against inconsistencies

In some cases, an incoming configuration to a Cisco APIC cluster will be validated against inconsistencies, where the validations involve externally-visible configurations (northbound traffic through the L3Outs). An invalid configuration error message will appear for those situations where the configuration is invalid.

Supported configuration modes for external Layer 3 features

The external Layer 3 features are supported in both configuration modes, with the following exception: route peering and route health injection (RHI) with a Layer 4 to Layer 7 service appliance is supported only in the named mode. Use the named mode across all border leaf switches for the tenant VRF instance where route-peering is involved.

L3Outs created using the implicit mode CLI are read-only in the GUI

Layer 3 external network objects (also known as an L3Out) created using the implicit mode CLI procedures are identified by names starting with “_ui_" and are marked as read-only in the GUI.

L3Outs created using the implicit mode CLI can become unmodifiable in the CLI if modified using the REST API

The CLI partitions L3Outs by function, such as interfaces, protocols, route map, and EPG. Modifying an L3Out's configuration using the REST API can break this structure, preventing you from modifying the L3Out using the CLI.

To remove the unmodifiable L3Outs, see the Troubleshooting Unwanted _ui_ Objects section in the *Cisco APIC Troubleshooting Guide*.

Configuration Validation

When the administrator enters a configuration in the Cisco Application Policy Infrastructure Controller (APIC), the Cisco APIC performs checks to make sure that the configuration is valid, which is known as validation. If the Cisco APIC accepts the configuration, but the configuration conflicts with existing configurations, Cisco APIC or the leaf switches might raise faults. The number 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 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.

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 you see when you use the following command:

```
show snapshot jobs import_job
```