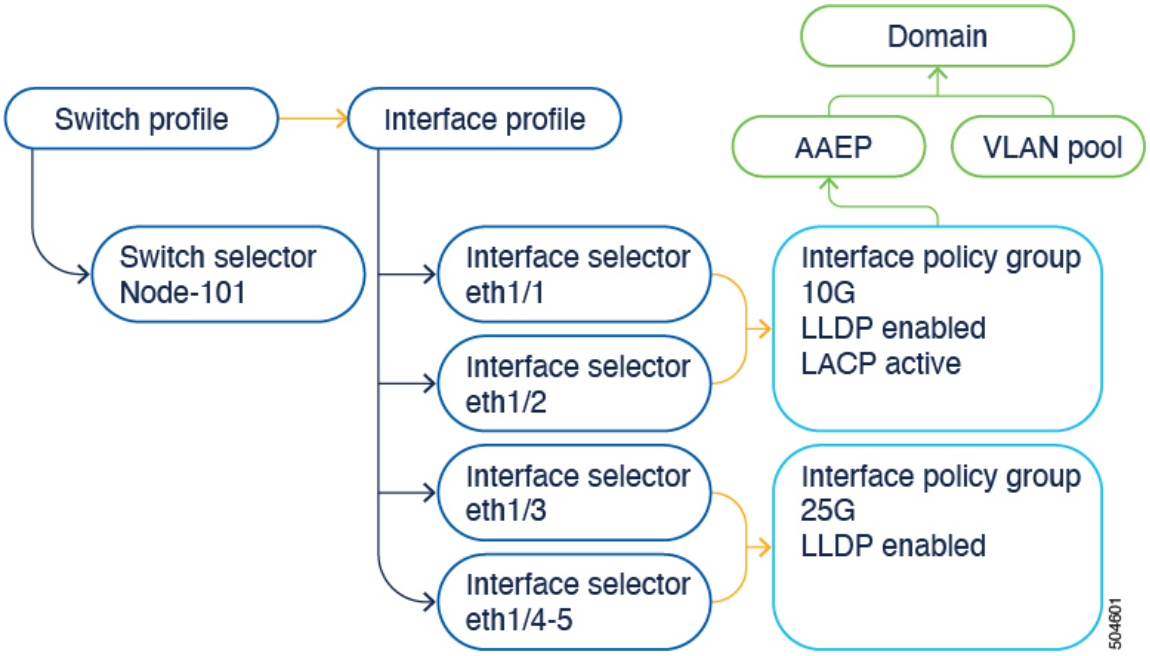
Access Interfaces

About Access Interfaces

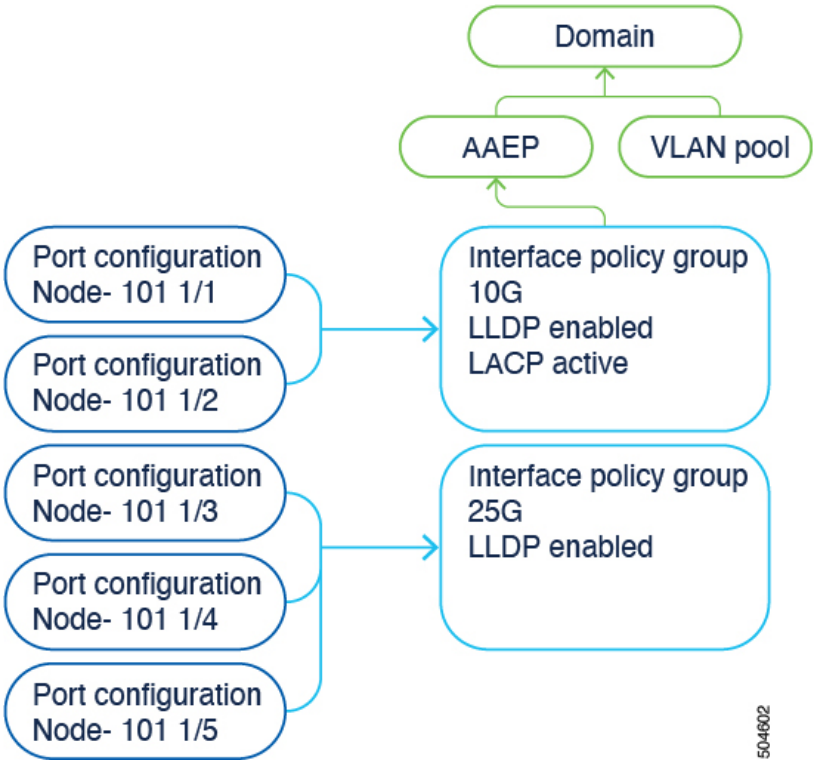
In Cisco Application Centric Infrastructure (ACI), interface configurations are performed by associating an interface policy group, which is a group of interface policies such as interface speed or link layer discovery protocol (LLDP), to an interface on a switch node. Cisco ACI uses four objects (switch profile, switch selector, interface profile, and interface selector) to select a certain interface on a certain switch node. This document refers to this mode of operations as the "profiles and selectors configuration." The following figure illustrates this configuration:

Figure 1: Interface configuration based on profiles and selectors



The Cisco ACI 6.0(1) release adds the "per-port configuration" configuration option (also known as the "interface configuration" or `infraPortConfig`, which is the name of the object for this configuration) that simplifies the interface configuration. This option presents the four objects as a single object and has the object specify an interface on a switch node. As a result, you do not need to use nor maintain switch profiles, switch selectors, interface profiles, and interface selectors.

Figure 2: Per-port interface configuration



You can access the per-port configuration in the following ways in the Cisco Application Policy Infrastructure Controller (APIC) GUI:

- **Fabric** > **Access Policies** > **Interface Configurations**
- **Fabric** > **Access Policies** > **Quick Start** > **Configure Interfaces**
- **Fabric** > **Inventory** > ***pod_ID*** > ***switch_name*** > **Interface tab** > **Configure Interfaces**

You can configure a switch using switch profiles and selectors and interface profiles and selectors at the same locations as before in the Cisco APIC GUI:

- **Fabric** > **Access Policies** > **Switches** > **Leaf Switches** > **Profiles**
- **Fabric** > **Access Policies** > **Switches** > **Spine Switches** > **Profiles**
- **Fabric** > **Access Policies** > **Interfaces** > **Leaf Interfaces** > **Profiles**
- **Fabric** > **Access Policies** > **Interfaces** > **Spine Interfaces** > **Profiles**

However, we recommend that you use the per-port configuration.

When using the interface configuration option, the Cisco APIC creates and maintains switch profiles and selectors and interface profiles and selectors as read-only with as few objects as possible. For example, if you configure two contiguous ports identically, the Cisco APIC automatically creates a range in the configuration. You configure the ports individually and you do not have to worry about these optimizations; the Cisco APIC takes care of them. These objects that the Cisco APIC creates automatically are called "system-generated profiles" and you do not need to maintain them.

The system-generated profiles are still visible under **Fabric** > **Access Policies** > **Interfaces** > **{Leaf | Spine} {Switches | Interfaces}** > **Profiles** in the GUI along with any user-defined profiles.

If you configure an interface using the interface configuration option and you previously configured the interface with profiles and selectors, the Cisco APIC automatically removes the interface from the existing profiles and moves the interface to the system-generated profiles seamlessly. If the pre-existing switch and interface profiles contain other interfaces, the Cisco APIC does not delete them; you can keep using them in the traditional way. If the pre-existing profiles no longer contain any interfaces, the Cisco APIC automatically removes those profiles because they are no longer needed.

If you already configured an interface using a multinode selector, meaning that you assigned the port selector to a profile with multiple leaf switches, you must simultaneously configure the same interface for each node that belongs to the multinode selector for the Cisco APIC to remove those nodes automatically from the existing profile. Otherwise, a validation failure blocks the migration.

Physical Ports Configuration

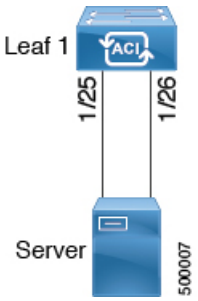
There are multiple ways to configure Cisco Application Centric Infrastructure (ACI) leaf switch interfaces:

- By using the selector and profile-based configuration model. From **Fabric** > **Access Policies** > **Switches** > **Leaf Switches** > **Profiles**, you can configure a switch profile that selects a leaf node with a leaf selector and the associated interface profiles to select the interface profiles (**Fabric** > **Access Policies** > **Interfaces** > **Leaf Interfaces** > **Profiles**), which in turn select one or more interfaces and associate them to interface policy groups.
- By using the interface configuration beginning with the Cisco Application Policy Infrastructure Controller (APIC) 6.0(1) release. Go to **Fabric** > **Access Policies** > **Interface Configurations**. This configuration option simplifies the configuration workflow by reducing the number of configuration steps from four to one.
- By using the inventory view from **Fabric** > **Inventory** > ***pod_ID*** > ***switch_name***. Beginning with the Cisco APIC 6.0(1) release, the inventory view configuration also uses the interface configuration.
- By using the **Fabric** > **Access Policies** > **Quick Start** wizard. Beginning with the Cisco APIC 6.0(1) release, the inventory view configuration also uses the interface configuration.

Configuring Leaf Switch Physical Ports Using the Interface Configuration Model Using the GUI

This procedure uses either the **Fabric** > **Access Policies** > **Quick Start** > **Configure Interfaces** or the **Fabric** > **Access Policies** > **Interface Configuration** page to attach a server to a Cisco Application Centric Infrastructure (ACI) leaf switch interface. The steps would be the same for attaching other kinds of devices to a Cisco ACI leaf switch interface.

Figure 3: Switch Interface Configuration for Bare Metal Server



Before you begin

- The Cisco ACI fabric is installed, Cisco Application Policy Infrastructure Controllers (APICs) are online, and the Cisco APIC cluster is formed and healthy.
- A Cisco APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.
- The target leaf switches are registered in the Cisco ACI fabric and available.

Procedure

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

Step 2 In the **Navigation** pane, choose **Quick Start** or **Interface Configuration**.

Step 3 In the **Work** pane, click **Configure Interfaces**. of the Quick Start wizard, click **Configure Interfaces**, or in the **Work** pane of **Interface Configuration** choose **Actions** > **Configure Interfaces**.

Step 4 In the **Configure Interfaces** dialog, perform the following actions:

a) For **Node Type**, click **Leaf**.
b) For **Port Type**, click **Access**.
c) For **Interface Type**, choose the desired type.
d) For **Interface Aggregation Type**, choose **Individual**.
e) For **Node**, click **Select Node**, put a check in the box for the desired switch (node), then click **OK**. You can select multiple switches.
f) For **Interfaces For All Switches**, enter the range of desired interfaces.
g) For **Leaf Access Port Policy Group**, click **Select Leaf Access Port Policy Group**.
h) In the **Select Leaf Access Port Policy Group** dialog, click **Create Leaf Access Port Policy Group**.

The interface policy group is a named policy that specifies the group of interface policies you will apply to the selected interfaces of the switch. Examples of interface policies include link level policy (for example, 1gbit port speed) and storm control interface policy.

i) In the **Create Leaf Access Port Policy Group** dialog, choose or create the desired policies.
j) Click **Save**.

What to do next

This completes the basic leaf switch interface configuration steps.

Note While this configuration enables hardware connectivity, no data traffic can flow without a valid application profile, EPG, and contract that is associated with this hardware configuration.

Migrating Interfaces From the Selector and Profile to Interface Configuration Using the GUI

You can convert the configuration of an existing interface from the selector- and profile-based model to the interface configuration model by using this procedure.

Note The Cisco Application Policy Infrastructure Controller (APIC) does not automatically migrate interfaces with an active policy group override. You must migrate these ports manually.

Procedure

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

Step 2 In the **Navigation** pane, choose **Interface Configuration**.

Step 3 In the table, select the interface that you want to migrate and click the three dots at the right.

Step 4 In the pop-up menu, choose **Edit Interface Configuration**.

The following a message displays:

```
This interface is configured using interface selectors. We recommend migrating it to newer way of
configuring interfaces. Clicking on Save will migrate this interface.
```

Step 5 Click **Save**.

The Cisco APIC converts the interface to the new configuration model.

Step 6 Perform one of the following sets of substeps depending on your Cisco APIC release and what you want to do:

To migrate a single interface:

a) In the table, select the interface that you want to migrate and click the three dots at the right.
b) In the pop-up menu, choose **Edit Interface Configuration**.

The following a message displays:

```
This interface is configured using interface selectors. We recommend migrating it to newer way of
configuring interfaces. Clicking on Save will migrate this interface.
```

c) Click **Save**.

The Cisco APIC converts the interface to the new configuration model.

In the 6.0(2) release and later, Cisco APIC simplifies the task of migrating existing configurations based on the selector- and profile-based model to the interface configuration model. You can migrate the selector-based configuration for all the ports of a node by selecting multiple nodes. This capability is useful if a selector spans across multiple nodes. To migrate multiple interfaces:

a) In the table, select the interfaces that you want to migrate.
b) Choose **Actions** > **Configure Interfaces**.

   The following a message displays:

   ```
   This interface is configured using interface selectors. We recommend migrating it to newer way of configuring interfaces. Clicking on Save will migrate this interface.
   ```

c) Click **Save**.

   The Cisco APIC converts the interfaces to the new configuration model.

Modifying the Interface Configuration Using the GUI

You can modify the configuration of an interface as follows:

Procedure

Step 1 On the menu bar, choose **Fabric** > **Access Policies**.
Step 2 In the **Navigation** pane, choose **Interface Configuration**.
Step 3 In the table, select the interface that you want to migrate and click the three dots at the right.
Step 4 In the pop-up menu, choose **Edit Interface Configuration**.

A window appears with the policy group associated to this interface.

Step 5 If there is an existing policy group, you can remove it by clicking the **x** next to the group.
Step 6 Click **Select Leaf Access Port Policy Group** to assign a new policy group.
Step 7 Choose an existing policy group or click **Create Leaf Access Port Policy Group** to create a new one.
Step 8 Click **Save**.

Viewing the Interface Configuration Using the GUI

The Cisco Application Policy Infrastructure Controller (APIC) GUI offers a unified view of the interface configuration regardless of whether the interfaces have been configured using the selector and profile model or the interface configuration model.

Choose **Fabric** > **Access Policies** > **Interface Configuration** and navigate the table on the right to see all the leaf nodes and interfaces.

Click a leaf node to view the leaf node's information, such as the admin state, the IP address of the TEP, the ID number, the hardware model, the serial number, and the software version.

Click an interface to view the interface's information. This view is called the "Infra Port Summary." Click the middle icon on the top right for a full screen view of the interface's information. The full screen view contains the following tabs that show additional information: **Overview**, **Operational**, **Deployed EPGs**, **VLANs**, **Statistics**, **QoS stats**, and **Event Analytics**. Click the **x** in the top right to close this full screen view.

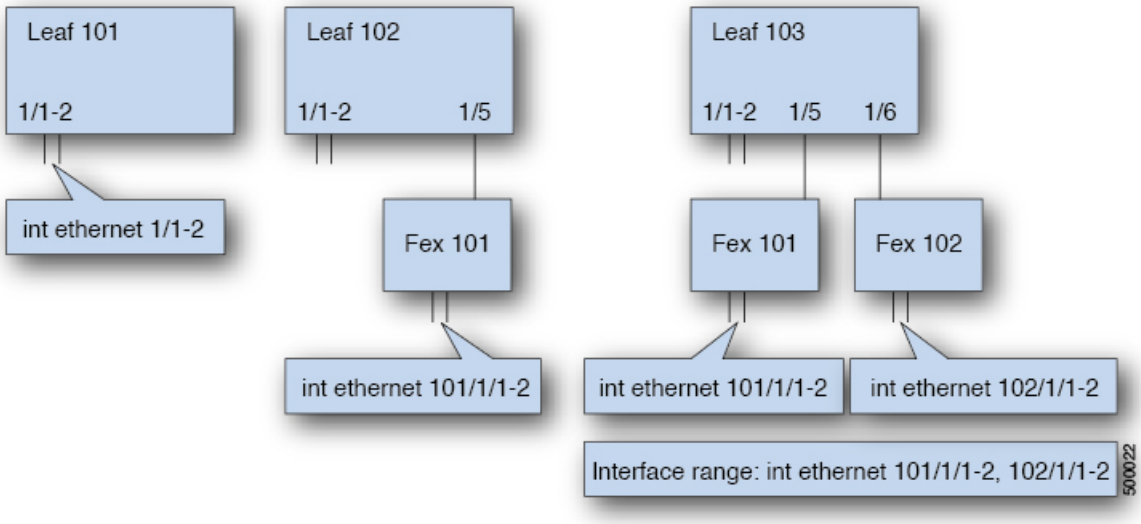
Click the policy group name of a given interface to view information about the policy group, such as the 802.1X configuration, attachable entity profile, CDP configuration, and LLDP configuration.

Configuring Physical Ports in Leaf Nodes and FEX Devices Using the NX-OS CLI

The commands in the following examples create many managed objects in the Cisco Application Centric Infrastructure (ACI) policy model that are fully compatible with the REST API/SDK and GUI. However, the CLI user can focus on the intended network configuration instead of Cisco ACI model internals.

Figure 4: Example of leaf node ports and FEX ports in Cisco ACI, on page 8 shows examples of Ethernet ports directly on leaf nodes or FEX modules attached to leaf nodes and how each is represented in the CLI. For FEX ports, the *fex-id* is included in the naming of the port itself as in **ethernet 101/1/1**. While describing an interface range, the **ethernet** keyword need not be repeated as in NX-OS. Example: **interface ethernet 101/1/1-2, 102/1/1-2**.

Figure 4: Example of leaf node ports and FEX ports in Cisco ACI



- Leaf node ID numbers are global.
- The *fex-id* numbers are local to each leaf node.
- Note the space after the keyword **ethernet**.

Procedure

Step 1 configure

Enters global configuration mode.

Example:

```
apic1# configure
```

Step 2 **leaf** *node-id*

Specifies the leaf nodes to be configured. The *node-id* can be a single node ID or a range of IDs, in the form *node-id1−node-id2*, to which the configuration will be applied.

Example:

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

Step 3 **interface** *type*

Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use "ethernet slot / port."

Example:

```
apic1(config-leaf)# interface ethernet 1/2
```

Step 4 (Optional) **fex associate** *node-id*

If the interface or interfaces to be configured are FEX interfaces, you must use this command to attach the FEX module to a leaf node before configuration.

Note
This step is required before creating a port channel using FEX ports.

Example:

```
apic1(config-leaf-if)# fex associate 101
```

Step 5 **speed** *speed*

The speed setting is shown as an example. At this point you can configure any of the interface settings shown in the table below.

Example:

```
apic1(config-leaf-if)# speed 10G
```

The following table shows the interface settings that can be configured at this point:

Command	Purpose
[no] shut	Shut down physical interface
[no] speed *speedValue*	Set the speed for physical interface
[no] link debounce time *time*	Set link debounce
[no] negotiate auto	Configure negotiate
[no] cdp enable	Disable/enable Cisco Discovery Protocol (CDP)
[no] mcp enable	Disable/enable Mis-cabling Protocol (MCP)

Command	Purpose
[no] lldp transmit	Set the transmit for physical interface
[no] lldp receive	Set the LLDP receive for physical interface
spanning-tree {bpduguard | bpdufilter} {enable | disable}	Configure spanning tree BPDU
[no] storm-control level *percentage* [burst-rate *percentage*]	Storm-control configuration (percentage)
[no] storm-control pps *packets-per-second* burst-rate *packets-per-second*	Storm-control configuration (packets-per-second)

Examples

Configure one port in a leaf node. The following example shows how to configure the interface eth1/2 in leaf 101 for the following properties: speed, cdp, and admin state.

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/2
apic1(config-leaf-if)# speed 10G
apic1(config-leaf-if)# cdp enable
apic1(config-leaf-if)# no shut
```

Configure multiple ports in multiple leaf nodes. The following example shows the configuration of speed for interfaces eth1/1-10 for each of the leaf nodes 101-103.

```
apic1(config)# leaf 101-103
apic1(config-leaf)# interface eth 1/1-10
apic1(config-leaf-if)# speed 10G
```

Attach a FEX to a leaf node. The following example shows how to attach a FEX module to a leaf node. Unlike in NX-OS, the leaf node port Eth1/5 is implicitly configured as fabric port and a FEX fabric port channel is created internally with the FEX uplink port(s). In Cisco ACI, the FEX fabric port channels use default configuration and no user configuration is allowed.

Note This step is required before creating a port channel using FEX ports, as described in the next example.

```
apic1(config)# leaf 102
apic1(config-leaf)# interface eth 1/5
apic1(config-leaf-if)# fex associate 101
```

Configure FEX ports attached to leaf nodes. This example shows configuration of speed for interfaces eth1/1-10 in FEX module 101 attached to each of the leaf nodes 102-103. The FEX ID 101 is included in the port identifier. FEX IDs start with 101 and are local to a leaf node.

```
apic1(config)# leaf 102-103
apic1(config-leaf)# interface eth 101/1/1-10
apic1(config-leaf-if)# speed 1G
```

Port Channels

PC Host Load Balancing Algorithms

The following table provides the default hash algorithm and symmetric hash algorithm options used in port channel load balancing across Cisco Application Centric Infrastructure (ACI) leaf node downlinks. The symmetric hash algorithm options were introduced in Cisco Application Policy Infrastructure Controller (APIC) release 2.3(1e).

Table 1: PC Host Load Balancing Algorithms

Traffic Type	Hashing Data Points
End Host PC (default)	**For Layer 2 traffic**: • Source MAC address • Destination MAC address • Segment ID (VXLAN VNID) or VLAN ID **For IP Traffic**: • Source MAC address • Destination MAC address • Source IP address • Destination IP address • Protocol type • Source Layer 4 port • Destination Layer 4 port • Segment ID (VXLAN VNID) or VLAN ID
PC symmetric hash (configurable)	**Choose one option**: • Source IP address • Destination IP address • Source Layer 4 port • Destination Layer 4 port

When there is more than one port channel on a leaf switch, such as Po1 and Po2, then the following scenario is supported:

- Po1: Enable symmetric hash with SIP only.
- Po2: Do not enable symmetric hash. Use default hashing.

However, the following scenario is not supported because the second port channel Po2 has a different hash parameter:

- Po1: Enable symmetric hash with SIP only.
- Po2: Enable symmetric hash with DIP only.

That is, on a single leaf switch, all port channels that require symmetric hashing should use the same hash policy/parameter or use the default hashing.

Symmetric hashing is not supported on the following switches:

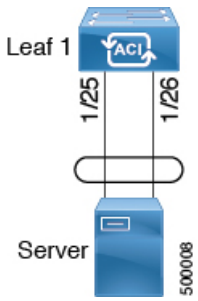
- Cisco Nexus 93128TX
- Cisco Nexus 9372PX
- Cisco Nexus 9372PX-E
- Cisco Nexus 9372TX
- Cisco Nexus 9372TX-E
- Cisco Nexus 9396PX
- Cisco Nexus 9396TX

Note Port channel hash algorithms are applied at each individual leaf node independently. The algorithms do not have influence on load balancing within the fabric, such as load balancing to leaf nodes in a vPC pair. Thus, symmetrical hashing is not supported on a vPC.

Cisco ACI Leaf Switch Port Channel Configuration Using the GUI

This procedure uses either the **Fabric** > **Access Policies** > **Quick Start** > **Configure Interfaces** or the **Fabric** > **Access Policies** > **Interface Configuration** page to attach a server to a Cisco Application Centric Infrastructure (ACI) leaf switch interface with a port channel. The steps would be the same for attaching other kinds of devices to a Cisco ACI leaf switch interface.

Figure 5: Switch Port Channel Configuration



Before you begin

- The Cisco ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.

- An APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.
- The target leaf switches are registered in the Cisco ACI fabric and available.

Procedure

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

Step 2 Use one of the following methods to open the **Configure Interfaces** dialog:

Method 1:

a) In the **Navigation** pane, choose **Quick Start**.
b) In the **Work** pane, click **Configure Interfaces**.

Method 2:

a) In the **Navigation** pane, choose **Interface Configuration**.
b) In the **Work** pane, choose **Actions** > **Configure Interfaces**.

Step 3 In the **Configure Interfaces** dialog, perform the following actions:

a) For **Node Type**, click **Leaf**.
b) For **Port Type**, click **Access**.
c) For **Interface Type**, choose the desired type.
d) For **Interface Aggregation Type**, choose **PC**.
e) For **Node**, click **Select Node**, put a check in the box for the desired switch (node), then click **OK**.
f) For **Interfaces For All Switches**, enter the range of desired interfaces.
g) For **PC/vPC Interface Policy Group**, click **Select PC/vPC Interface Policy Group**, then select an existing port channel policy group or create new one.
h) For **Port Channel Member Policy**, click **Select Port Channel Member Policy**, then select an existing port channel member policy or create new one.
i) In the **Select PC/vPC Interface Policy Group** dialog, choose an existing policy group or click **Create PC/vPC Interface Policy Group** to create a new one.
j) Click **Save**.

What to do next

This completes the port channel configuration steps.

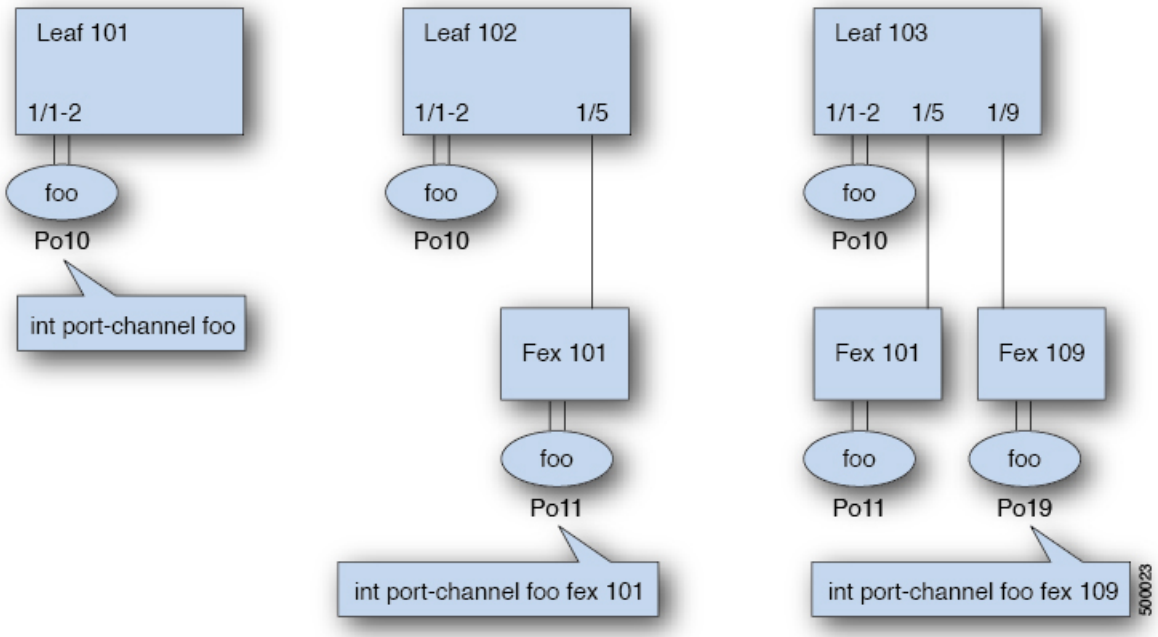
Note While this configuration enables hardware connectivity, no data traffic can flow without a valid application profile, EPG, and contract that is associated with this hardware configuration.

Configuring Port Channels in Leaf Nodes and FEX Devices Using the NX-OS CLI

Port channels are logical interfaces in NX-OS used to aggregate bandwidth for multiple physical ports and also for providing redundancy in case of link failures. In NX-OS, port channel interfaces are identified by user-specified numbers in the range 1 to 4096 unique within a node. Port channel interfaces are either configured explicitly (using the **interface port-channel** command) or created implicitly (using the **channel-group** command). The configuration of the port channel interface is applied to all the member ports of the port channel. There are certain compatibility parameters (speed, for example) that cannot be configured on the member ports.

In the ACI model, port channels are configured as logical entities identified by a name to represent a collection of policies that can be assigned to set of ports in one or more leaf nodes. Such assignment creates one port channel interface in each of the leaf nodes identified by an auto-generated number in the range 1 to 4096 within the leaf node, which may be same or different among the nodes for the same port channel name. The membership of these port channels may be same or different as well. When a port channel is created on the FEX ports, the same port channel name can be used to create one port channel interface in each of the FEX devices attached to the leaf node. Thus, it is possible to create up to N+1 unique port channel interfaces (identified by the auto-generated port channel numbers) for each leaf node attached to N FEX modules. This is illustrated with the examples below. Port channels on the FEX ports are identified by specifying the *fex-id* along with the port channel name (**interface port-channel foo fex 101**, for example).

Figure 6: Example with port channels on leaf switches and FEX ports



- N+1 instances per leaf of port channel foo are possible when each leaf is connected to N FEX nodes.
- Leaf ports and FEX ports cannot be part of the same port channel instance.
- Each FEX node can have only one instance of port channel foo.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters global configuration mode.
Step 2	**template port-channel** *channel-name* **Example:** `apic1(config)# template port-channel foo`	Creates a new port channel or configures an existing port channel (global configuration).
Step 3	[**no**] **switchport access vlan** *vlan-id* **tenant** *tenant-name* **application** *application-name* **epg** *epg-name* **Example:** `apic1(config-po-ch-if)# switchport access vlan 4 tenant ExampleCorp application Web epg webEpg`	Deploys the EPG with the VLAN on all ports with which the port channel is associated.

Step 4	**channel-mode active**
	Example:
	`apic1(config-po-ch-if)# channel-mode active`
	Note
	To enable symmetric hashing, enter the **lacp symmetric-hash** command:
	`apic1(config-po-ch-if)# lacp symmetric-hash`
		Note
		The **channel-mode** command is equivalent to the mode option in the channel-group command in NX-OS. In ACI, however, this is supported for the port channel (not on a member port).
		Symmetric hashing is not supported on the following switches:
		• Cisco Nexus 93128TX
		• Cisco Nexus 9372PX
		• Cisco Nexus 9372PX-E
		• Cisco Nexus 9372TX
		• Cisco Nexus 9372TX-E
		• Cisco Nexus 9396PX
		• Cisco Nexus 9396TX

Step 5	**exit** **Example:** `apic1(config-po-ch-if)# exit`	Returns to configure mode.
Step 6	**leaf** *node-id* **Example:** `apic1(config)# leaf 101`	Specifies the leaf switches to be configured. The *node-id* can be a single node ID or a range of IDs, in the form *node-id1−node-id2*, to which the configuration will be applied.

	Command or Action	Purpose
Step 7	**interface** *type* **Example:** `apic1(config-leaf)# interface ethernet 1/1-2`	Specifies the interface or range of interfaces that you are configuring to the port channel.
Step 8	[**no**] **channel-group** *channel-name* **Example:** `apic1(config-leaf-if)# channel-group foo`	Assigns the interface or range of interfaces to the port channel. Use the keyword **no** to remove the interface from the port channel. To change the port channel assignment on an interface, you can enter the **channel-group** command without first removing the interface from the previous port channel.

Step 9	(Optional) **lacp port-priority** *priority*
	Example:
	`apic1(config-leaf-if)# lacp port-priority 1000`
	`apic1(config-leaf-if)# lacp rate fast`
		This setting and other per-port LACP properties can be applied to member ports of a port channel at this point.
		Note
		In the ACI model, these commands are allowed only after the ports are member of a port channel. If a port is removed from a port channel, configuration of these per-port properties are removed as well.

The following table shows various commands for global configurations of port channel properties in the ACI model. These commands can also be used for configuring overrides for port channels in a specific leaf in the (config-leaf-if) CLI mode. The configuration made on the port channel is applied to all member ports.

CLI Syntax	Feature
[no] speed <speedValue>	Set the speed for port channel
[no] link debounce time <time>	Set Link Debounce for port channel
[no] negotiate auto	Configure Negotiate for port channel
[no] cdp enable	Disable/Enable CDP for port channel
[no] mcp enable	Disable/Enable MCP for port channel
[no] lldp transmit	Set the transmit for port channel
[no] lldp receive	Set the lldp receive for port channel
spanning-tree <bpduguard | bpdufilter> <enable | disable>	Configure spanning tree BPDU
[no] storm-control level <percentage> [burst-rate <percentage>]	Storm-control configuration (percentage)
[no] storm-control pps <packet-per-second> burst-rate <packets-per-second>	Storm-control configuration (packets-per-second)
[no] channel-mode { active | passive | on| mac-pinning }	LACP mode for the link in port channel l

CLI Syntax	Feature
[no] lacp min-links <value>	Set minimum number of links
[no] lacp max-links <value>	Set maximum number of links
[no] lacp fast-select-hot-standby	LACP fast select for hot standby ports
[no] lacp graceful-convergence	LACP graceful convergence
[no] lacp load-defer	LACP load defer member ports
[no] lacp suspend-individual	LACP individual Port suspension
[no] lacp port-priority	LACP port priority
[no] lacp rate	LACP rate

Examples

Configure a port channel (global configuration). A logical entity foo is created that represents a collection of policies with two configurations: speed and channel mode. More properties can be configured as required.

Note The channel mode command is equivalent to the mode option in the channel group command in NX-OS. In ACI, however, this supported for the port channel (not on member port).

```
apic1(config)# template port-channel foo
apic1(config-po-ch-if)# switchport access vlan 4 tenant ExampleCorp application Web epg
webEpg
apic1(config-po-ch-if)# speed 10G
apic1(config-po-ch-if)# channel-mode active
```

Configure ports to a port channel in a FEX. In this example, port channel foo is assigned to ports Ethernet 1/1-2 in FEX 101 attached to leaf node 102 to create an instance of port channel foo. The leaf node will auto-generate a number, say 1002 to identify the port channel in the switch. This port channel number would be unique to the leaf node 102 regardless of how many instance of port channel foo are created.

Note The configuration to attach the FEX module to the leaf node must be done before creating port channels using FEX ports.

```
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 101/1/1-2
apic1(config-leaf-if)# channel-group foo
```

In Leaf 102, this port channel interface can be referred to as interface port channel foo FEX 101.

```
apic1(config)# leaf 102
apic1(config-leaf)# interface port-channel foo fex 101
apic1(config-leaf)# shut
```

Configure ports to a port channel in multiple leaf nodes. In this example, port channel foo is assigned to ports Ethernet 1/1-2 in each of the leaf nodes 101-103. The leaf nodes will auto generate a number unique in each node (which may be same or different among nodes) to represent the port channel interfaces.

```
apic1(config)# leaf 101-103
apic1(config-leaf)# interface ethernet 1/1-2
apic1(config-leaf-if)# channel-group foo
```

Add members to port channels. This example would add two members eth1/3-4 to the port channel in each leaf node, so that port channel foo in each node would have members eth 1/1-4.

```
apic1(config)# leaf 101-103
apic1(config-leaf)# interface ethernet 1/3-4
apic1(config-leaf-if)# channel-group foo
```

Remove members from port channels. This example would remove two members eth1/2, eth1/4 from the port channel foo in each leaf node, so that port channel foo in each node would have members eth 1/1, eth1/3.

```
apic1(config)# leaf 101-103
apic1(config-leaf)# interface eth 1/2,1/4
apic1(config-leaf-if)# no channel-group foo
```

Configure port channel with different members in multiple leaf nodes. This example shows how to use the same port channel foo policies to create a port channel interface in multiple leaf nodes with different member ports in each leaf. The port channel numbers in the leaf nodes may be same or different for the same port channel foo. In the CLI, however, the configuration will be referred as interface port channel foo. If the port channel is configured for the FEX ports, it would be referred to as interface port channel foo fex <fex-id>.

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/1-2
apic1(config-leaf-if)# channel-group foo
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 1/3-4
apic1(config-leaf-if)# channel-group foo
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 103
apic1(config-leaf)# interface ethernet 1/5-8
apic1(config-leaf-if)# channel-group foo
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 101/1/1-2
apic1(config-leaf-if)# channel-group foo
```

Configure per port properties for LACP. This example shows how to configure member ports of a port channel for per-port properties for LACP.

Note In ACI model, these commands are allowed only after the ports are member of a port channel. If a port is removed from a port channel, configuration of these per-port properties would be removed as well.

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/1-2
apic1(config-leaf-if)# channel-group foo
```

```
apic1(config-leaf-if)# lacp port-priority 1000
apic1(config-leaf-if)# lacp rate fast
```

Configure admin state for port channels. In this example, a port channel foo is configured in each of the leaf nodes 101-103 using the channel-group command. The admin state of port channel(s) can be configured in each leaf using the port channel interface. In ACI model, the admin state of the port channel cannot be configured in the global scope.

```
// create port-channel foo in each leaf
apic1(config)# leaf 101-103
apic1(config-leaf)# interface ethernet 1/3-4
apic1(config-leaf-if)# channel-group foo

// configure admin state in specific leaf
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel foo
apic1(config-leaf-if)# shut
```

Override config is very helpful to assign specific vlan-domain, for example, to the port channel interfaces in each leaf while sharing other properties.

```
// configure a port channel global config
apic1(config)# interface port-channel foo
apic1(config-if)# speed 1G
apic1(config-if)# channel-mode active

// create port-channel foo in each leaf
apic1(config)# leaf 101-103
apic1(config-leaf)# interface ethernet 1/1-2
apic1(config-leaf-if)# channel-group foo

// override port-channel foo in leaf 102
apic1(config)# leaf 102
apic1(config-leaf)# interface port-channel foo
apic1(config-leaf-if)#  speed 10G
apic1(config-leaf-if)#  channel-mode on
apic1(config-leaf-if)#  vlan-domain dom-foo
```

This example shows how to change port channel assignment for ports using the channel-group command. There is no need to remove port channel membership before assigning to other port channel.

```
apic1(config)# leaf 101-103
apic1(config-leaf)# interface ethernet 1/3-4
apic1(config-leaf-if)# channel-group foo
apic1(config-leaf-if)# channel-group bar
```

Port channel dynamic load balancing

Port channel static load balancing

When a switch forwards a packet to a server or another switch through a port channel with multiple bundled links, it uses a hashing algorithm to choose the link for that packet. The hashing algorithm considers parameters such as the source and destination IP addresses, source and destination port numbers, and occasionally protocol type.

In port channel static load balancing, traffic is consistently hashed to the same link in the port channel, irrespective of the load on each link. The egress member link is statically selected based on a 5-tuple hash.

Disadvantage of port channel static load balancing

Port channel static load balancing approach can result in uneven traffic distribution on the links of a port channel, with some links being highly utilized while others remain underutilized.

Port channel dynamic load balancing

Port channel dynamic load balancing (DLB) is a networking technique that distributes traffic across multiple links in a port channel based on the load of each link. Port channel DLB adjusts traffic distribution on the links of a port channel based on the current load of the links. The switch monitors the egress traffic load on each link and selects the link with the least utilization to distribute the traffic.

Flowlet

A flowlet is a group of consecutive packets, or a burst, within the same flow, separated by an idle interval. Each flowlet can be forwarded independently without causing packet reordering.

Flowlet aging time

The flowlet table maintains information about flowlets and uses aging to identify flowlet gaps and prevent packet reordering. For example, the flowlet aging time is set to x microseconds. If traffic is not received for an existing flowlet for x microseconds, the flowlet entry will be marked for deletion. After one additional x microseconds, this flowlet entry will be permanently removed.

When the burst gap is less than the configured flowlet aging time, packets are sent on the same link as the flowlet entry still exists in the flowlet table. When the burst gap exceeds the configured flowlet aging time, packets are sent on different links as the flowlet entry for these packets is removed.

Benefits of port channel dynamic load balancing

Port channel DLB has these benefits.

- Efficient traffic distribution: balances traffic based on real-time link utilization, preventing network congestion.
- Improved network performance: reduces the impact of long-lived bursty flows on network congestion.
- Faster convergence: adapts to changes in port status without requiring software intervention.

Limitations of port channel dynamic load balancing

Port channel DLB has these limitations.

- Port channel DLB can be applied only to the front panel ports configured as Layer 2 port channels.
- Port channel DLB on a Layer 2 port channel for an L3Out SVI VLAN is not qualified, and therefore it is not supported.
- FEX HIF, NIF, and FC ports are not supported.
- SAN and fibre channel port channels are not supported.
- The number of port channel member ports with DLB supported is 32 in each slice and 64 in each switch.

Note The number of fabric links with **Uplink Port Fast Link Failover** enabled is also included in the maximum port capacity in each switch, which is 64.

Note A switch consists of one or more slices. Each slice is a self-contained switching subsystem. The ports in the switch are distributed among the slices.

- BUM (Broadcast, Unknown Unicast, and Multicast traffic) is not supported.
- LEM-based chassis such as Cisco Nexus 9400 is not supported.

Create port channel dynamic load balancing policy using the GUI

Follow these steps to create a port channel dynamic load balancing policy using the GUI.

Procedure

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

Step 2 In the Navigation pane, expand **Policies** > **Switch** , right-click **Dynamic Load Balancing**, and choose **Create Dynamic Load Balancing Policy**.

The **Create Dynamic Load Balancing Policy** dialog box appears.

Step 3 Enter the name and description of the policy.

Step 4 Enter the aging time, in microseconds, in the **Flowlet Aging Time** field.

The range of **Flowlet Aging Time** is from 1 to 2000000 microseconds. The default value is 500 microseconds. The recommended value is twice the maximum Round Trip Time (RTT) observed in the traffic flows.

Step 5 Click **Submit**.

Associate port channel dynamic load balancing policy to a switch policy group using the GUI

Follow these steps to associate a port channel dynamic load balancing policy to a switch policy group using the GUI.

Procedure

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

Step 2 In the Navigation pane, expand **Switches** > **Leaf Switches** > **Policy Groups** and choose a switch policy group.

The **Access Switch Policy Group** pane appears.

Step 3 In the **Dynamic Load Balancing Policy** drop-down list, choose a DLB policy.

Step 4 Click **Submit**.

Enable dynamic load balancing on the port channel using the GUI

Follow these steps to enable dynamic load balancing on the port channel using the GUI.

Procedure

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

Step 2 In the Navigation pane, expand **Interfaces** > **Leaf Interfaces** > **Profiles** and expand an access profile.

Step 3 Choose a port channel from the access profile.

The **Access Port Selector** pane appears.

Step 4 In the **Policy Group** drop-down list, perform one of the following:

- Choose an exisitng policy group and click the icon next to the **Policy Group** drop-down list. The **PC/vPC Interface Policy Group** dialog box appears.
- Choose **Create PC Interface Policy Group**. The **Create PC Interface Policy Group** dialog box appears. Enter the name of the policy group.
- Choose **Create VPC Interface Policy Group**. The **Create VPC Interface Policy Group** appears. Enter the name of the policy group.

Step 5 In the **Port Channel Policy** drop-down list, perform one of the following:

- Choose an exisitng port channel policy and click the icon next to the **Port Channel Policy** drop-down list. The **Port Channel Policy** dialog box appears.
- Choose **Create Port Channel Policy**. The **Create Port Channel Policy** dialog box appears. Enter the name of the port channel policy.

Step 6 In the **Port Channel Policy** dialog box or **Create Port Channel Policy** dialog box, click the **Dynamic** toggle button to enable dynamic load balancing on the port channel.

Step 7 Click **Submit**.

View port channel dynamic load balancing configurations using the GUI

Follow these steps to view the port channel dynamic load balancing configurations using the GUI.

Procedure

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

Step 2 In the Navigation pane, expand **Pod** > **Leaf**.

Step 3 Choose **Dynamic Load Balancing and Fast Link Failover**.

The **Port-Channel Dynamic Load Balancing and Uplink Port Fast Link Failover** pane appears.

Step 4 Click the **Policy** tab to view the flowlet aging time of the port channel DLB policy.

Step 5 Click the **Operational** > **Per Switch Usage** tabs to view the number of ports that currently use DLB or Uplink Port Fast Link Failover and the maximum number of ports supported at the switch level.

The number of ports activated with DLB or Uplink Port Fast Link Failover must not exceed the switch capacity.

Step 6 Click the **Operational** > **Per Slice Usage (DLB Only)** tabs to view the number of ports that currently use DLB and the maximum number of ports supported at the slice level.

The number of ports in each slice activated with DLB must not exceed the slice capacity.

Step 7 In the Navigation pane, expand **Pod** > **Node** and click the **Summary** > **DLB Interfaces** pane to view the number of ports that currently use DLB or Uplink Port Fast Link Failover and the maximum number of ports supported at the node level.

View port channel dynamic load balancing statistics using the GUI

Follow these steps to view the port channel dynamic load balancing statistics using the GUI.

Procedure

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

Step 2 In the Navigation pane, expand **Pod** > **Leaf**.

Step 3 Choose **Dynamic Load Balancing and Fast Link Failover**.

The **Port-Channel Dynamic Load Balancing and Uplink Port Fast Link Failover** pane appears.

Step 4 Click the **Stats** tab to view the DLB statistics information.

Table 2: Description of DLB Statistics

DLB Statistics	Description
Total DLB Ingress Packets	Total packets received in DLB
Total Flowlets with Collision	Number of occurrences of flowlet collisions
Total Flowlets Created	Number of new flowlet entries created
Total Flowlets Hit	Number of times an incoming packet matched an existing flowlet entry

Verify port channel dynamic load balancing configurations using the CLI

Follow these steps to verify the port channel dynamic load balancing configurations using the CLI.

Procedure

Step 1 Run the **show port-channel dlb usage** command to verify the DLB resources used by all the port channels.

Example:

```
node# show port-channel dlb usage
Dynamic load balancing resource usage for port-channel1

                   Interface         Configured                Operational               State
    Port-channel:  port-channel1     dynamic                   dynamic                   (SU)

    Members:       Ethernet1/31      dynamic                   dynamic                   (up)

                   Ethernet1/32      dynamic                   dynamic                   (up)

                   Ethernet1/33      dynamic                   dynamic                   (up)

                   Ethernet1/34      dynamic                   dynamic                   (up)

                   Ethernet1/35      dynamic                   dynamic                   (up)


DLB resources utilized by port-channel1: 5

Dynamic load balancing resource usage for port-channel2

                   Interface         Configured                Operational               State
    Port-channel:  port-channel2     static                    static                    (SU)

    Members:       Ethernet1/1       static                    static                    (up)


DLB resources utilized by port-channel2: 0

Dynamic load balancing resource usage for port-channel3

                   Interface         Configured                Operational               State
    Port-channel:  port-channel3     static                    static                    (SU)

    Members:       Ethernet1/3       static                    static                    (up)


DLB resources utilized by port-channel3: 0

Dynamic load balancing resource usage for port-channel4

                   Interface         Configured                Operational               State
    Port-channel:  port-channel4     dynamic                   dynamic                   (SU)

    Members:       Ethernet1/7       dynamic                   dynamic                   (up)

                   Ethernet1/8       dynamic                   dynamic                   (up)

                   Ethernet1/9       dynamic                   dynamic                   (up)

                   Ethernet1/10      dynamic                   dynamic                   (up)
```

```
DLB resources utilized by port-channel4: 4

Dynamic load balancing resource usage for port-channel5

                   Interface         Configured                Operational               State
    Port-channel:  port-channel5     static                    static                    (RU)

    Members:       Ethernet1/4       static                    static                    (up)

                   Ethernet1/5       static                    static                    (up)


DLB resources utilized by port-channel5: 0

Dynamic Load Balance Resource Summary for above port-channels

Total DLB configured port-channels:               2
Total DLB operational port-channels:              2
Total DLB non operational port-channels:          0
Total DLB not supported port-channels:            0
Total DLB resources utilized by port-channels:    9
```

Step 2 Run the **show port-channel dlb usage interface port-channel** command to verify the DLB resources used by a specific port channel.

Example:

```
node# show port-channel dlb usage interface port-channel 4
Dynamic load balancing resource usage for port-channel4

                   Interface         Configured                Operational               State
    Port-channel:  port-channel4     dynamic                   dynamic                   (SU)

    Members:       Ethernet1/7       dynamic                   dynamic                   (up)

                   Ethernet1/8       dynamic                   dynamic                   (up)

                   Ethernet1/9       dynamic                   dynamic                   (up)

                   Ethernet1/10      dynamic                   dynamic                   (up)


DLB resources utilized by port-channel4: 4

Dynamic Load Balance Resource Summary for above port-channels

Total DLB configured port-channels:               1
Total DLB operational port-channels:              1
Total DLB non operational port-channels:          0
Total DLB not supported port-channels:            0
Total DLB resources utilized by port-channels:    4
```

Step 3 Run the **show port-channel internal info** command to verify the DLB status of ports of the port channel.

Example:

```
node# show port-channel internal info interface port-channel 4

port-channel4
channel    : 4
bundle     : 65535
ifindex    : 0x16000003
```

```
admin mode : active
oper mode  : active
nports     : 4
active     : 4
pre cfg    : 0
ltl        : 0x2006 (8198)
lif        : 0x0
iod        : 0x150 (336)
global id  : 4
flag       : 0
lock count : 0
num. of SIs: 0
ac mbrs    : 0 0
lacp graceful conv disable   : 0
lacp suspend indiv disable   : 0
pc min-links                 : 1
pc max-bundle                : 16
pc max active members        : 16
pc is-suspend-minlinks       : 0
port load defer enable       : 0
port-channel bfd config enabled     : 0
port-channel bfd config complete: 0
port-channel bfd destination: null
port-channel bfd start timeout: 0
port-channel bfd distinguished name (dn):

port-channel dlb admin mode: dynamic
port-channel dlb operational mode: dynamic
lacp fast-select-hot-standby disable   : 0
port-channel port hash-distribution    : adaptive
ethpm bundle lock count : 0
Members:
Ethernet1/7 [bundle_no = 0]     is_ltl_programmed = 1
    is_pixm_ltl_programmed = 1
Port BFD session state: 5 (none)
dlb operational mode: dynamic
Ethernet1/8 [bundle_no = 1]     is_ltl_programmed = 1
    is_pixm_ltl_programmed = 1
Port BFD session state: 5 (none)

dlb operational mode: dynamic
Ethernet1/9 [bundle_no = 2]     is_ltl_programmed = 1
    is_pixm_ltl_programmed = 1
Port BFD session state: 5 (none)

dlb operational mode: dynamic

Ethernet1/10 [bundle_no = 3]     is_ltl_programmed = 1
    is_pixm_ltl_programmed = 1
Port BFD session state: 5 (none)

dlb operational mode: dynamic
port-channel external lock:
Lock Info: resource [eth-port-channel 4]
  type[0] p_gwrap[(nil)]
      FREE @ 2024-10-14T17:07:22.364638000+05:30
  type[1] p_gwrap[(nil)]
      FREE @ 2024-10-14T17:10:05.142954000+05:30
  type[2] p_gwrap[(nil)]
      FREE @ 2024-10-14T17:10:04.493453000+05:30
0x16000003
internal (ethpm bundle) lock:
Lock Info: resource [eth-port-channel 4]
  type[0] p_gwrap[(nil)]
```

```
      FREE @ 2024-10-14T17:07:22.364616000+05:30
  type[1] p_gwrap[(nil)]
      FREE @ 2024-10-14T17:10:11.300577000+05:30
  type[2] p_gwrap[(nil)]
      FREE @ 2024-10-14T17:10:11.300369000+05:30
0x16000003
```

Step 4 Run the **show dlb statistics** command to verify the DLB statistics.

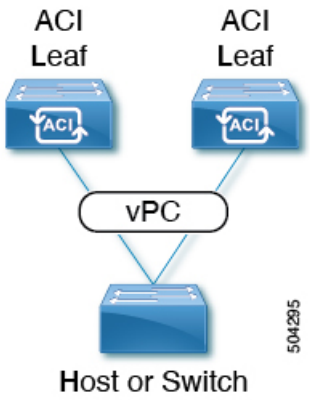
Example:

```
node# show dlb statistics
dlb statistics:
    Flowlet Hit          = 0
    Flowlet Create       = 156
    Flowlet Collision    = 0
    Flowlet DLB Rx pkts  = 3734297752
```

Virtual Port Channels in Cisco ACI

A virtual port channel (vPC) allows links that are physically connected to two different Cisco Application Centric Infrastructure (ACI) leaf nodes to appear as a single port channel (PC) to a third device, such as a network switch, server, any other networking device that supports link aggregation technology. vPCs consist of two Cisco ACI leaf switches designated as vPC peer switches. Of the vPC peers, one is primary and one is secondary. The system formed by the switches is referred to as a vPC domain.

Figure 7: vPC Domain



The following behavior is specific to the Cisco ACI vPC implementation:

- No dedicated peer-link between the vPC peers. Instead, the fabric itself serves as the Multi-Chassis Trunking (MCT).
- Peer reachability protocol: Cisco ACI uses the Zero Message Queue (ZMQ) instead of Cisco Fabric Services (CFS).
  - ZMQ is an open-source, high-performance messaging library that uses TCP as the transport.
  - This library is packaged as libzmq on the switch and linked into each application that needs to communicate with a vPC peer.

- Peer reachability is not handled using a physical peer link. Instead, routing triggers are used to detect peer reachability.
  - The vPC manager registers with Unicast Routing Information Base (URIB) for peer route notifications.
  - When IS-IS discovers a route to the peer, URIB notifies the vPC manager, which in turn attempts to open a ZMQ socket with the peer.
  - When the peer route is withdrawn by IS-IS, the vPC manager is again notified by URIB, and the vPC manager brings down the MCT link.
- When creating a vPC domain between two leaf switches, the following hardware model limitations apply:
  - Generation 1 switches are compatible only with other generation 1 switches. These switch models can be identified by the lack of "EX," "FX, "FX2," "GX," or later suffix at the end of the switch name. For example, N9K-9312TX.
  - Generation 2 and later switches can be mixed together in a vPC domain. These switch models can be identified by the "EX," "FX, "FX2," "GX," or later suffix at the end of the switch name. For example N9K-93108TC-EX or N9K-9348GC-FXP.

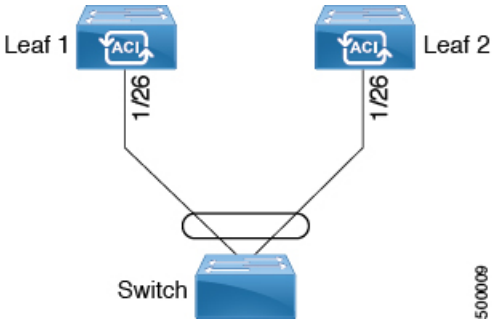
  Examples of compatible vPC switch pairs:
  - N9K-C9312TX and N9K-C9312TX
  - N9K-C93108TC-EX and N9K-C9348GC-FXP
  - N9K-C93180TC-FX and N9K-C93180YC-FX
  - N9K-C93180YC-FX and N9K-C93180YC-FX

  Examples of incompatible vPC switch pairs:
  - N9K-C9312TX and N9K-C93108TC-EX
  - N9K-C9312TX and N9K-C93180YC-FX
- Port channels and virtual port channels can be configured with or without LACP.

  If you configure a virtual port channel with LACP, LACP sets a port to the suspended state if it does not receive an LACP PDU from the peer. This can cause some servers to fail to boot up as they require LACP to bring up the port logically. You can tune the behavior to individual use by disabling **LACP suspend individual**. To do so, create a port channel policy in your vPC policy group, and after setting the mode to LACP active, remove **Suspend Individual Port**. Afterward, the ports in the vPC will stay active and continue to send LACP packets.
- Adaptive load balancing (ALB), based on ARP negotiation, across virtual port channels is not supported in Cisco ACI.

Cisco ACI Virtual Port Channel Workflow

Figure 8: Virtual port channel configuration



The configuration workflow for virtual port channels (vPCs) is as follows:

Before you begin

- 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.
- Ensure that the hardware of the two leaf switches that are going to be part of the same vPC pair is compatible. For more information, see Virtual Port Channels in Cisco ACI, on page 27.

Procedure

Step 1 Configure the VLAN pools, domain, AAEP, access leaf port policy group of type vPC.

Step 2 Configure the vPC switch pairs.

Step 3 Configure the vPC interfaces.

Step 4 Configure the application profile.

a) On the menu bar, choose **Tenants** > **All Tenants**.
b) In the Work pane, double-click a tenant.
c) In the Navigation pane, choose ***tenant_name*** > **Quick Start**.
d) Configure the endpoint groups (EPGs), contracts, bridge domain, subnet, and context.
e) Associate the application profile EPGs with the virtual port channel switch profile that you created previously.

Defining a vPC Using the GUI

This procedure defines a vPC using the GUI. We recommend that you keep the leaf switch peer group names simple as shown in the following example:

- Leaf201_202
- Leaf203_204
- Leaf205_206

For naming and numbering best practices, see the *Cisco ACI Object Naming and Numbering: Best Practices* document:

https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/kb/b-Cisco-ACI-Naming-and-Numbering.html

Before you begin

Ensure that the hardware of the two leaf switches that are going to be part of the same vPC pair is compatible. For more information, see Virtual Port Channels in Cisco ACI, on page 27.

Procedure

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

Step 2 In the Navigation pane, choose **Policies** > **Switch** > **Virtual Port Channel default**.

Step 3 In the **Explicit vPC Protection Groups** table, click **+** and fill out the fields as follows:

a) In the **Name** field, enter the vPC pair name.

Example name: `Leaf201_202`. A name similar to the example easily identifies which two fabric nodes are vPC peers.

b) In the **ID** field, enter the vPC pair ID (logical peer ID).

Example ID: `201`. The example uses the first node ID number of the pair to make it easier to correlate the ID with the vPC pair.

c) In the **Switch 1** and **Switch 2** fields, choose the leaf switches for the vPC switch pair.

d) Click **Submit**.

The vPC pair gets added to the **Explicit vPC Protection Groups** table. The **Virtual IP** value is an auto-generated IP address from the system tunnel endpoint (TEP) pool, and represents the virtual shared (Anycast) TEP of the vPC switch pair. That is, packets destined to vPC-connected endpoints of the vPC pair will use this Anycast VTEP to send the packets.

Configuring Virtual Port Channels in Leaf Nodes and FEX Devices Using Profiles and Selectors

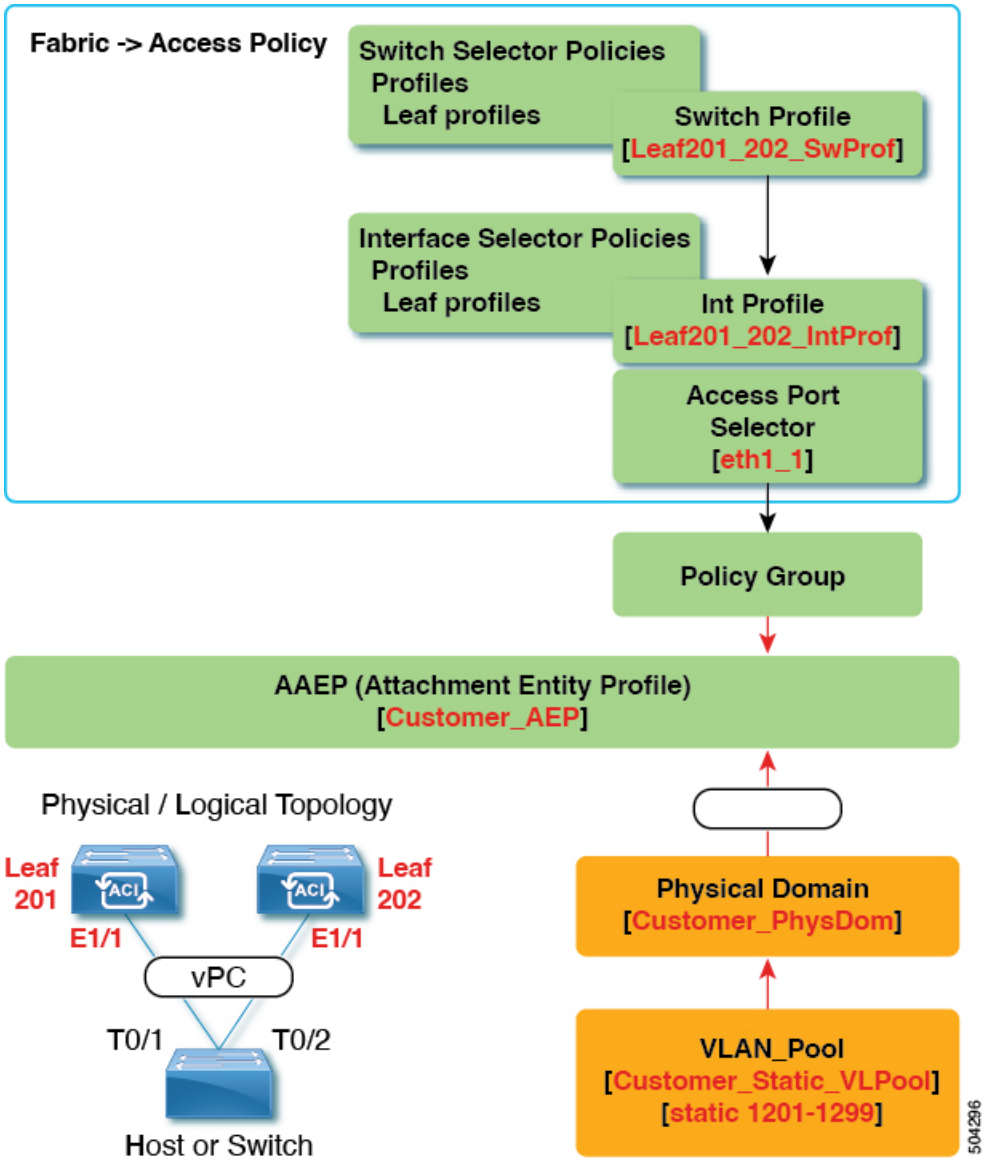
vPC With the Same Leaf Switch Interfaces Across Two Leaf Switches With Combined Profiles

For the example of this use case, you define the following things:

- A combined switch profile called `Leaf201_202_SwProf` (node 201 and node 202).
- A combined interface profile called `Leaf201_202_IntProf` (node 201 and node 202).
- An access port selector called `Eth1_1` (under the `Leaf201_202` interface profile) is pointing toward a vPC interface policy group.
- The vPC interface policy group is pointing toward an AAEP called `Customer_AEP`.
- The AEP (`Customer_AEP`) has an association with the `Customer_PhysDom`.

- The `Customer_PhysDom` has an association with a VLAN pool called `Customer_Static_VLPool`.

Figure 9: vPC With the Same Leaf Switch Interfaces Across Two Leaf Switches With Combined Profiles



What This Configuration Does

On switches `Leaf201` and `Leaf202`, configure port Eth1/1 to be part of a vPC. This vPC interface will have access to VLANs 1201 through 1299. Depending on the interface policy group, you can enable LACP Active and other interface specific policy configurations.

When to Use This Configuration

If you have dedicated pairs of compute leaf switches with nothing but vPC-connected servers, for example, this would be a solid use case for using combined-switch/interface profiles under your fabric access policies

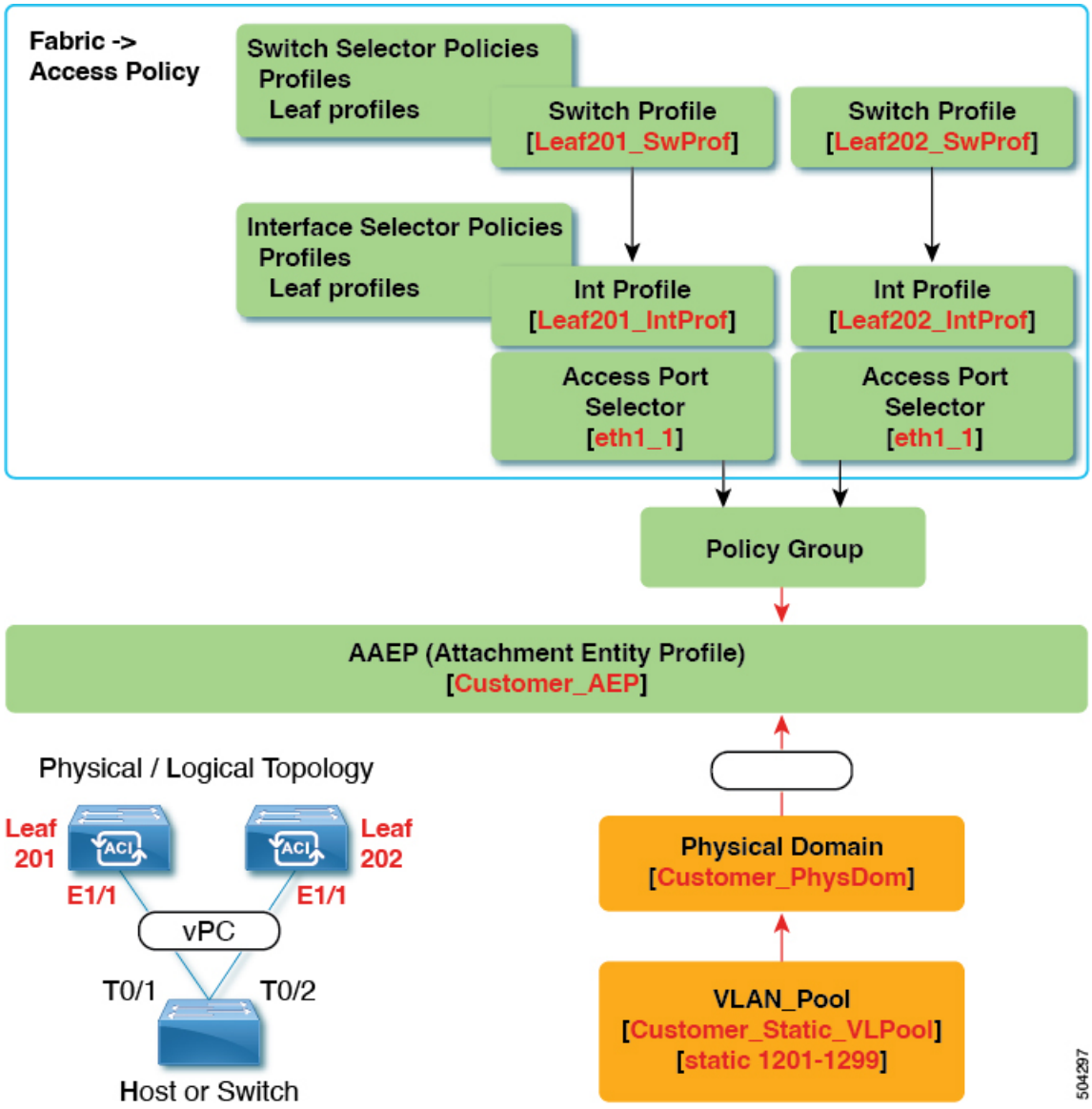
for those switches. You could preconfigure your switch, interface, access port selector, and vPC interface policy groups in such a way that allowed you to plug in 48 chassis-type servers with minimal effort.

vPC With the Same Leaf Switch Interfaces Across Two Leaf Switches with Individual Profiles

For the example of this use case, you define the following things:

- Individual switch profiles called `Leaf201_SwProf` and `Leaf202_SwProf` (node 201 and node 202).
- Individual interface profiles called `Leaf201_IntProf` and `Leaf202_IntProf` (node 201 and node 202)
- Access port selectors called `Eth1_1` (under the `Leaf201` and `Leaf202` interface profiles) is pointing toward the same vPC interface policy group.
- The vPC interface policy group is pointing toward an AAEP called `Customer_AEP`.
- The AEP (`Customer_AEP`) has an association with the `Customer_PhysDom`.
- The `Customer_PhysDom` has an association with a VLAN pool called `Customer_Static_VLPool`.

Figure 10: vPC With the Same Leaf Switch Interfaces Across Two Leaf Switches with Individual Profiles



What This Configuration Does

On switches `Leaf201` and `Leaf202`, configure port Eth1/1 to be a part of a vPC. This vPC interface will have access to VLANs 1201 through 1299. Depending on the interface policy group, you can enable LACP active and other interface specific policy configurations.

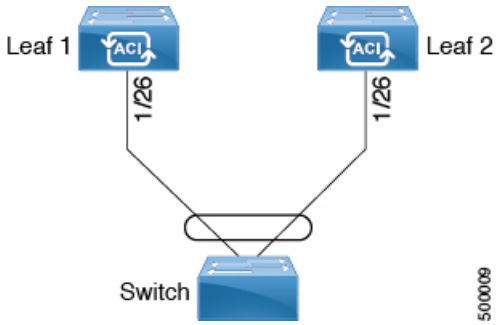
When to Use This Configuration

Use this configuration when you have leaf switches that support mixed workloads, such as compute, services, or Cisco Application Policy Infrastructure Controllers (APICs). In this case, having individual interface profiles allows for the most amount of flexibility, while allowing you to keep your **Fabric** > **Access Policies** configuration as clean and manageable as possible.

Configuring a Cisco ACI Leaf Switch Virtual Port Channel Using the Interface Configuration Model Using the GUI

This procedure uses the "Interface Configuration" methodology to attach a trunked switch to a Cisco Application Centric Infrastructure (ACI) leaf switch virtual port channel. The steps are the same for attaching other kinds of devices to an Cisco ACI leaf switch interface.

Figure 11: Switch Virtual Port Channel Configuration



Before you begin

- The Cisco ACI fabric is installed, the Cisco Application Policy Infrastructure Controllers (APICs) are online, and the Cisco APIC cluster is formed and healthy.
- A Cisco APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.
- The target leaf switches are registered in the Cisco ACI fabric and available.

Note When creating a vPC domain between two leaf switches, ensure that the hardware of the two leaf switches that are going to be part of the same vPC pair is compatible. For more information, see Virtual Port Channels in Cisco ACI, on page 27.

Procedure

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

Step 2 Use one of the following methods to open the **Configure Interfaces** dialog:

Method 1:

a) In the **Navigation** pane, choose **Quick Start**.
b) In the **Work** pane, click **Configure Interfaces**.

Method 2:

a) In the **Navigation** pane, choose **Interface Configuration**.
b) In the **Work** pane, choose **Actions** > **Configure Interfaces**.

Step 3 In the **Configure Interfaces** dialog, perform the following actions:

a) For **Node Type**, click **Leaf**.
b) For **Port Type**, click **Access**.
c) For **Interface Type**, click **Ethernet**.
d) For **Interface Aggregation Type**, choose **vPC**.
e) For **vPC Leaf Switch Pair**, click **Select vPC Leaf Switch Pair**, put a check in the box for the desired switch pair, then click **Select**. You can select multiple switches. Optionally, click **Create vPC Leaf Switch Pair** and fill out the fields as desired, then select the pair and click **Select**.
f) For **Interfaces For All Switches**, enter the range of desired interfaces.
g) For **PC/vPC Interface Policy Group**, click **Select PC/vPC Interface Policy Group**.
h) In the **Select PC/vPC Interface Policy Group** dialog, choose an existing vPC policy group and click **Select**. Optionally, click **Create PC/vPC Interface Policy Group** to create a new vPC policy group, fill out the fields, and click **Save**, then choose that policy group and click **Select**.
i) For **Port Channel Member Policy**, click **Select Port Channel Member Policy**, then choose a policy and click **Select**. Optionally, click **Create Port Channel Member Policy**, fill out the fields as desired, and click **Save**, then choose that policy and click **Select**..
j) Click **Save**.

Verification: Use the CLI **show int** command on the leaf switches where the external switch is attached to verify that the vPC is configured accordingly.

What to do next

This completes the switch virtual port channel configuration steps.

Note While this configuration enables hardware connectivity, no data traffic can flow without a valid application profile, EPG, and contract that is associated with this hardware configuration.

Configuring Virtual Port Channels in Leaf Nodes and FEX Devices Using the NX-OS CLI

A virtual port channel (vPC) is an enhancement to port-channels that allows connection of a host or switch to two upstream leaf nodes to improve bandwidth utilization and availability. In NX-OS, vPC configuration is done in each of the two upstream switches and configuration is synchronized using peer link between the switches.

The Cisco Application Centric Infrastructure (ACI) model does not require a peer link and vPC configuration can be done globally for both the upstream leaf nodes. A global configuration mode called **vpc context** is introduced in Cisco ACI and vPC interfaces are represented using a type **interface vpc** that allows global configuration applicable to both leaf nodes.

Two different topologies are supported for vPC in the Cisco ACI model: vPC using leaf ports and vPC over FEX ports. It is possible to create many vPC interfaces between a pair of leaf nodes and similarly, many vPC interfaces can be created between a pair of FEX modules attached to the leaf node pairs in a straight-through topology.

vPC considerations include:

- The vPC name used is unique between leaf node pairs. For example, only one vPC 'corp' can be created per leaf pair (with or without FEX).
- Leaf ports and FEX ports cannot be part of the same vPC.
- Each FEX module can be part of only one instance of vPC corp.
- vPC context allows configuration
- The vPC context mode allows configuration of all vPCs for a given leaf pair. For vPC over FEX, the *fex-id* pairs must be specified either for the vPC context or along with the vPC interface, as shown in the following two alternative examples.

```
(config)# vpc context leaf 101 102
(config-vpc)# interface vpc Reg fex 101 101
```

or

```
(config)# vpc context leaf 101 102 fex 101 101
(config-vpc)# interface vpc Reg
```

In the Cisco ACI model, vPC configuration is done in the following steps (as shown in the examples below).

Note A VLAN domain is required with a VLAN range. It must be associated with the port-channel template.

1. VLAN domain configuration (global config) with VLAN range
2. vPC domain configuration (global config)
3. Port-channel template configuration (global config)
4. Associate the port channel template with the VLAN domain
5. Port-channel configuration for vPC (global config)
6. Configure ports to vPC in leaf nodes
7. Configure Layer 2, Layer 3 for vPC in the vPC context

Before you begin

Ensure that the hardware of the two leaf switches that are going to be part of the same vPC pair is compatible. For more information, see Virtual Port Channels in Cisco ACI, on page 27.

Procedure

Step 1 configure

Enters global configuration mode.

Example:

```
apic1# configure
```

Step 2 **vlan-domain***name*[**dynamic**] [**type** *domain-type*]

Configures a VLAN domain for the virtual port-channel (here with a port-channel template).

Example:

```
apic1(config)# vlan-domain dom1 dynamic
```

Step 3 **vlan***range*

Configures a VLAN range for the VLAN domain and exits the configuration mode. The range can be a single VLAN or a range of VLANs.

Example:

```
apic1(config-vlan)# vlan 1000-1999
apic1(config-vlan)# exit
```

Step 4 **vpc domain explicit** *domain-id* **leaf** *node-id1 node-id2*

Configures a vPC domain between a pair of leaf nodes. You can specify the vPC domain ID in the explicit mode along with the leaf node pairs.

Alternative commands to configure a vPC domain are as follows:

- **vpc domain** [**consecutive** | **reciprocal**]

  The consecutive and reciprocal options allow auto configuration of a vPC domain across all leaf nodes in the Cisco ACI fabric.

- **vpc domain consecutive** *domain-start* **leaf** *start-node end-node*

  This command configures a vPC domain consecutively for a selected set of leaf node pairs.

Example:

```
apic1(config)# vpc domain explicit 1 leaf 101 102
```

Step 5 **peer-dead-interval** *interval*

Configures the time delay the Leaf switch waits to restore the vPC before receiving a response from the peer. If it does not receive a response from the peer within this time, the Leaf switch considers the peer dead and brings up the vPC with the role as a master. If it does receive a response from the peer it restores the vPC at that point. The range is from 5 seconds to 600 seconds. The default is 200 seconds.

Example:

```
apic1(config-vpc)# peer-dead-interval 10
```

Step 6 **exit**

Returns to global configuration mode.

Example:

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

Step 7 **template port-channel** *channel-name*

Creates a new port-channel or configures an existing port-channel (global configuration).

All vPCs are configured as port-channels in each leaf pair. The same port-channel name must be used in a leaf pair for the same vPC. This port-channel can be used to create a vPC among one or more pairs of leaf nodes. Each leaf node will have only one instance of this vPC.

Example:

```
apic1(config)# template port-channel corp
```

Step 8 **vlan-domain member***vlan-domain-name*

Associates the port channel template with the previously configured VLAN domain.

Example:

```
vlan-domain member dom1
```

Step 9 **switchport access vlan** *vlan-id* **tenant** *tenant-name* **application** *application-name* **epg** *epg-name*

Deploys the EPG with the VLAN on all ports with which the port-channel is associated.

Example:

```
apic1(config-po-ch-if)# switchport access vlan 4 tenant ExampleCorp application Web epg webEpg
```

Step 10 **channel-mode active**

Note
A port-channel must be in active channel-mode for a vPC.

Example:

```
apic1(config-po-ch-if)# channel-mode active
```

Step 11 **exit**

Returns to configure mode.

Example:

```
apic1(config-po-ch-if)# exit
```

Step 12 **leaf** *node-id1 node-id2*

Specifies the pair of leaf switches to be configured.

Example:

```
apic1(config)# leaf 101-102
```

Step 13 **interface** *typeleaf/interface-range*

Specifies the interface or range of interfaces that you are configuring to the port-channel.

Example:

```
apic1(config-leaf)# interface ethernet 1/3-4
```

Step 14 [**no**] **channel-group** *channel-name* **vpc**

Assigns the interface or range of interfaces to the port-channel. Use the keyword **no** to remove the interface from the port-channel. To change the port-channel assignment on an interface, you can enter the **channel-group** command without first removing the interface from the previous port-channel.

Note
The **vpc** keyword in this command makes the port-channel a vPC. If the vPC does not already exist, a vPC ID is automatically generated and is applied to all member leaf nodes.

Example:

```
apic1(config-leaf-if)# channel-group corp vpc
```

Step 15 **exit**

Example:

```
apic1(config-leaf-if)# exit
```

Step 16 **exit**

Example:

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

Step 17 **vpc context leaf** *node-id1 node-id2*

The vPC context mode allows configuration of vPC to be applied to both leaf node pairs.

Example:

```
apic1(config)# vpc context leaf 101 102
```

Step 18 **interface vpc** *channel-name*

Example:

```
apic1(config-vpc)# interface vpc blue fex 102 102
```

Step 19 (Optional) [**no**] **shutdown**

Administrative state configuration in the vPC context allows changing the admin state of a vPC with one command for both leaf nodes.

Example:

```
apic1(config-vpc-if)# no shut
```

Example

This example shows how to configure a basic vPC.

```
apic1# configure
apic1(config)# vlan-domain dom1 dynamic
apic1(config-vlan)# vlan 1000-1999
apic1(config-vlan)# exit
apic1(config)# vpc domain explicit 1 leaf 101 102
apic1(config-vpc)# peer-dead-interval 10
apic1(config-vpc)# exit
apic1(config)# template port-channel corp
apic1(config-po-ch-if)# vlan-domain member dom1
apic1(config-po-ch-if)# channel-mode active
apic1(config-po-ch-if)# exit
apic1(config)# leaf 101-102
apic1(config-leaf)# interface ethernet 1/3-4
apic1(config-leaf-if)# channel-group corp vpc
apic1(config-leaf-if)# exit
apic1(config)# vpc context leaf 101 102
```

This example shows how to configure vPCs with FEX ports.

```
apic1(config-leaf)# interface ethernet 101/1/1-2
apic1(config-leaf-if)# channel-group Reg vpc
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# interface vpc corp
apic1(config-vpc-if)# exit
```

```
apic1(config-vpc)# interface vpc red fex 101 101
apic1(config-vpc-if)# switchport
apic1(config-vpc-if)# exit
apic1(config-vpc)# interface vpc blue fex 102 102
apic1(config-vpc-if)# shut
```

Reflective Relay (802.1Qbg)

Reflective relay is a switching option beginning with Cisco APIC Release 2.3(1). Reflective relay—the tagless approach of IEEE standard 802.1Qbg—forwards all traffic to an external switch, which then applies policy and sends the traffic back to the destination or target VM on the server as needed. There is no local switching. For broadcast or multicast traffic, reflective relay provides packet replication to each VM locally on the server.

One benefit of reflective relay is that it leverages the external switch for switching features and management capabilities, freeing server resources to support the VMs. Reflective relay also allows policies that you configure on the Cisco APIC to apply to traffic between the VMs on the same server.

In the Cisco ACI, you can enable reflective relay, which allows traffic to turn back out of the same port it came in on. You can enable reflective relay on individual ports, port channels, or virtual port channels as a Layer 2 interface policy using the APIC GUI, NX-OS CLI, or REST API. It is disabled by default.

The term *Virtual Ethernet Port Aggregator* (VEPA) is also used to describe 802.1Qbg functionality.

Reflective Relay Support

Reflective relay supports the following:

- IEEE standard 802.1Qbg tagless approach, known as reflective relay.

  Cisco APIC Release 2.3(1) release does not support the IEE standard 802.1Qbg S-tagged approach with multichannel technology.

- Physical domains.

  Virtual domains are not supported.

- Physical ports, port channels (PCs), and virtual port channels (vPCs).

  Cisco Fabric Extender (FEX) and blade servers are not supported. If reflective relay is enabled on an unsupported interface, a fault is raised, and the last valid configuration is retained. Disabling reflective relay on the port clears the fault.

- Cisco Nexus 9000 series switches with *EX* or *FX* at the end of their model name.

Enabling Reflective Relay Using the GUI

Reflective relay is disabled by default; however, you can enable it on a port, port channel, or virtual port channel as a Layer 2 interface policy on the switch. You first configure a policy and then associate the policy with a policy group.

Before you begin

This procedure assumes that you have set up the Cisco Application Centric Infrastructure (ACI) fabric and installed the physical switches.

Procedure

Step 1 Choose **Fabric** > **External Access Policies** > > **Interface Policies** and then open the **Policies** folder.
Step 2 Right-click the **L2 Interface** folder and choose **Create L2 Interface Policy**.
Step 3 In the **Create L2 Interface Policy** dialog box, enter a name in the **Name** field.
Step 4 In the **Reflective Relay (802.1Qbg)** area, click **enabled**.
Step 5 Choose other options in the dialog box as needed.
Step 6 Click **SUBMIT**.
Step 7 In the **Policies** navigation pane, open the **Policy Groups** folder and click the **Leaf Policy Groups** folder.
Step 8 In the **Leaf Policy Groups** central pane, expand the **ACTIONS** drop-down list, and choose **Create Leaf Access Port Policy Group**, **Create PC Interface Policy Group**, **Create vPC Interface Policy Group**, or **Create PC/vPC Override Policy Group**.
Step 9 In the policy group dialog box, enter a name in the **Name field**.
Step 10 From the **L2 Interface Policy** drop-down list, choose the policy that you just created to enable Reflective Relay.
Step 11 Click **Submit**.

Enabling Reflective Relay Using the NX-OS CLI

Reflective relay is disabled by default; however, you can enable it on a port, port channel, or virtual port channel as a Layer 2 interface policy on the switch. In the NX-OS CLI, you can use a template to enable reflective relay on multiple ports or you can enable it on individual ports.

Before you begin

This procedure assumes that you have set up the Cisco Application Centric Infrastructure (ACI) fabric and installed the physical switches.

Procedure

Enable reflective relay on one or multiple ports:

Example:

This example enables reflective relay on a single port:

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/2
apic1(config-leaf-if)# switchport vepa enabled
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

Example:

This example enables reflective relay on multiple ports using a template:

```
apic1(config)# template policy-group grp1
apic1(config-pol-grp-if)# switchport vepa enabled
apic1(config-pol-grp-if)# exit
```

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/2-4
apic1(config-leaf-if)# policy-group grp1
```

Example:

This example enables reflective relay on a port channel:

```
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po2
apic1(config-leaf-if)# switchport vepa enabled
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)#
```

Example:

This example enables reflective relay on multiple port channels:

```
apic1(config)# template port-channel po1
apic1(config-if)# switchport vepa enabled
apic1(config-if)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/3-4
apic1(config-leaf-if)# channel-group po1
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

Example:

This example enables reflective relay on a virtual port channel:

```
apic1(config)# vpc domain explicit 1 leaf 101 102
apic1(config-vpc)# exit
apic1(config)# template port-channel po4
apic1(config-if)# exit
apic1(config)# leaf 101-102
apic1(config-leaf)# interface eth 1/11-12
apic1(config-leaf-if)# channel-group po4 vpc
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# interface vpc po4
apic1(config-vpc-if)# switchport vepa enabled
```

Configuring Port, PC, and vPC Connections to FEX Devices

FEX connections and the profiles used to configure them can be created using the GUI, NX-OS-style CLI, or the REST API. Interface profiles for configuring FEX connections are supported since the Cisco Application Policy Infrastructure Controller (APIC) 3.0(1k) release.

The configuration consists of two key steps:

- Defining the connectivity between the FEX and the Cisco Application Centric Infrastructure (ACI) leaf switch
- Configuring the FEX ports connected to the servers

After you configure the FEX connectivity to the Cisco ACI leaf switch, the configuration of the FEX host-facing ports is no different than the configuration of Cisco ACI leaf switch ports as individual interfaces, port channels, or vPCs.

For information on how to configure ports, PCs, and vPCs using the GUI, the NX-OS-style CLI, or the REST API, see the following sections:

- Physical Ports Configuration, on page 4
- Port Channels, on page 11
- Virtual Port Channels in Cisco ACI, on page 27

ACI FEX Guidelines

Observe the following guidelines when deploying a FEX:

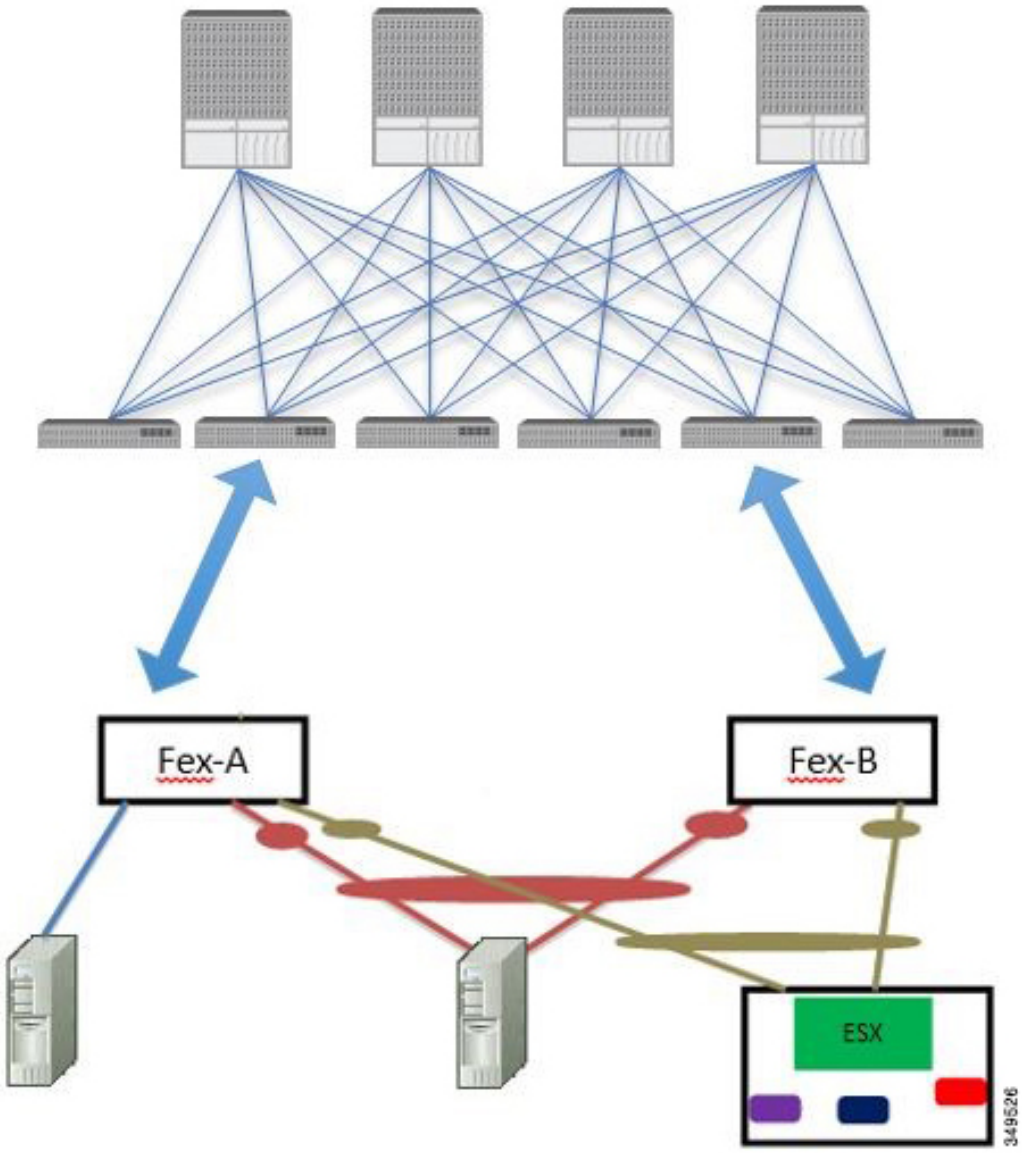
- Assuming that no leaf switch front panel ports are configured to deploy and EPG and VLANs, a maximum of 10,000 port EPGs are supported for being deployed using a FEX.
- For each FEX port or vPC that includes FEX ports as members, a maximum of 20 EPGs per VLAN are supported.
- A vPC with FEX interfaces ignores the minimum and maximum number of links configured in its port-channel policy. The vPC remains up even if the number of links is less than the minimum or greater than the maximum.

FEX Virtual Port Channels

The ACI fabric supports Cisco Fabric Extender (FEX) server-side virtual port channels (vPC), also known as an FEX straight-through vPC.

Note When creating a vPC domain between two leaf switches, ensure that the hardware of the two leaf switches that are going to be part of the same vPC pair is compatible. For more information, see Virtual Port Channels in Cisco ACI, on page 27.

Figure 12: Supported FEX vPC Topologies



Supported FEX vPC port channel topologies include the following:

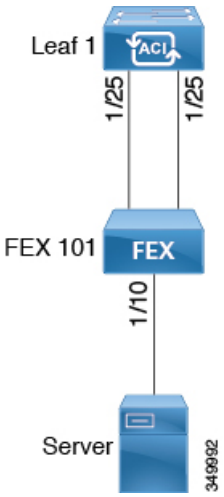
- Both VTEP and non-VTEP hypervisors behind a FEX.
- Virtual switches (such as AVS or VDS) connected to two FEXs that are connected to the ACI fabric (vPCs directly connected on physical FEX ports is not supported - a vPC is supported only on port channels).

Note When using GARP as the protocol for notification of IP-to-MAC binding changes to different interfaces on the same FEX, you must set the bridge domain mode to **ARP Flooding** and enable **EP Move Detection Mode**: **GARP-based Detection**, on the **L3 Configuration** page of the bridge domain wizard. This workaround is only required with Generation 1 switches. With Generation 2 switches or later, this is not an issue.

Configuring a FEX Connection to the ACI Leaf Switch Using the GUI

This procedure provides the steps for attaching a server to the FEX. The steps would be the same for attaching any device to a Cisco Application Centric Infrastructure (ACI)-attached FEX.

Figure 13: Basic FEX Configuration



Note Configuring FEX connections with FEX IDs `165` to `199` is not supported in the APIC GUI. To use one of these FEX IDs, configure the profile using the NX-OS style CLI. For more information, see *Configuring FEX Connections Using Interface Profiles with the NX-OS Style CLI*.

Before you begin

- The Cisco ACI fabric is installed, Cisco Application Policy Infrastructure Controllers (APICs) are online, and the Cisco APIC cluster is formed and healthy.
- An Cisco APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.
- The target leaf switches, interfaces, and protocols are configured and available.
- The FEX is powered on and connected to the target leaf switch interfaces

Note A maximum of eight members are supported in fabric port channels connected to FEXs.

Procedure

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

Step 2 In the **Navigation** pane, choose **Interface Configuration**.

Step 3 In the **Work** pane, choose **Actions** > **Fabric Extender**.

Step 4 In the **Fabric Extender** dialog, perform the following actions:

a) For **Node**, click **Select Node**, put a check in the box for the desired node, then click **OK**. You can select multiple nodes.

b) For **Interfaces For All Switches**, enter the range of desired interfaces.

c) For **Connected FEX ID**, enter the ID of the FEX.

You must configure FEX IDs 165 - 199, using the NX-OS-style CLI. See *Configuring FEX Connections Using Interface Profiles with the NX-OS Style CLI*.

d) Click **Save**.
The Cisco APIC auto-generates the necessary FEX profile (*switch-policy-name*_FexP*FEX-ID*) and selector (*switch-policy-name*_ifselctor).

Verification: Use the CLI **show fex** command on the switch where the FEX is attached to verify that the FEX is online.

Step 5 You can now configure FEX interfaces, such as regular Cisco ACI leaf switch interfaces by using **Fabric** > **Fabric Access** > **Interface Configuration**.

What to do next

Note While this configuration enables hardware connectivity, no data traffic can flow without a valid application profile, EPG, and contract that is associated with this hardware configuration.

Configuring FEX Connectivity to an ACI leaf switch Using Profiles with the NX-OS-Style CLI

Use this procedure to configure FEX connections to leaf nodes using the NX-OS style CLI.

Note Configuring FEX connections with FEX IDs 165 to 199 is not supported in the Cisco Application Policy Infrastructure Controller (APIC) GUI. To use one of these FEX IDs, configure the profile using the following commands.

Procedure

Step 1 **configure**

Enters global configuration mode.

Example:

```
apic1# configure
```

Step 2 **leaf-interface-profile** *name*

Specifies the leaf interface profile to be configured.

Example:

```
apic1(config)# leaf-interface-profile fexIntProf1
```

Step 3 **leaf-interface-group** *name*

Specifies the interface group to be configured.

Example:

```
apic1(config-leaf-if-profile)# leaf-interface-group leafIntGrp1
```

Step 4 **fex associate** *fex-id* [**template** *template-typefex-template-name*]

Attaches a FEX module to a leaf node. Use the optional template keyword to specify a template to be used. If it does not exist, the system creates a template with the name and type you specified.

Example:

```
apic1(config-leaf-if-group)# fex associate 101
```

Example

This merged example configures a leaf interface profile for FEX connections with ID 101.

```
apic1# configure
apic1(config)# leaf-interface-profile fexIntProf1
apic1(config-leaf-if-profile)# leaf-interface-group leafIntGrp1
apic1(config-leaf-if-group)# fex associate 101
```

Configuring Port Profiles

Uplink and downlink conversion is supported on Cisco Nexus 9000 series switches with names that end in EX or FX, and later (for example, N9K-C9348GC-FXP or N9K-C93240YC-FX2). A FEX connected to converted downlinks is also supported.

For information about the supported supported Cisco switches, see Port Profile Configuration Summary, on page 52.

When an uplink port is converted to a downlink port, it acquires the same capabilities as any other downlink port.

Restrictions

- Fast Link Failover policies and port profiles are not supported on the same port. If port profile is enabled, Fast Link Failover cannot be enabled or vice versa.
- The last 2 uplink ports of supported leaf switches cannot be converted to downlink ports (they are reserved for uplink connections).
- Dynamic breakouts (both 100Gb and 40Gb) are supported on profiled QSFP ports on the N9K-C93180YC-FX switch. Breakout and port profile are supported together for conversion of uplink to downlink on ports 49-52. Breakout (both **10g-4x** and **25g-4x** options) is supported on downlink profiled ports.
- The N9K-C9348GC-FXP does not support FEX.
- Breakout is supported only on downlink ports, and not on fabric ports that are connected to other switches.
- A Cisco ACI leaf switch cannot have more than 56 fabric links.
- Reloading a switch after changing a switch's port profile configuration interrupts traffic through the data plane.
- If a port profile is configured on any LEM type and you want to replace that LEM, the replacement LEM type must match the LEM type you removed.

Guidelines

In converting uplinks to downlinks and downlinks to uplinks, consider these guidelines.

Subject	Guideline
Port profile guidelines for N9K-X9400-8D	These guidelines apply for this LEM: • This LEM has 8 ports with a default of 4 downlinks and 4 uplinks. • Port profile conversion is supported on the first 6 ports. • This LEM does not have a port group dependency.
Port profile guidelines for N9K-X9400-16W	These guidelines apply for this LEM: • This LEM has 16 ports, with a default of 12 downlinks and 4 uplinks. • Port profile conversion is supported on the first 6 ports. • This LEM has a port group dependency of 2 ports for port profile conversion. Which means that ports 1-2, 3-4, and 5-6 have a port group dependency. For example, if port 2 is to be converted as an uplink, port 1 should also be converted to an uplink.

Subject	Guideline
Port profile guidelines for N9K-X9400-22L	The 6.1(2) release adds support for this LEM. These guidelines apply: • This LEM has 22 ports with a default of 14 downlinks and 8 uplinks. • Port profile conversion is supported on the first 18 ports. • This LEM has a port group dependency of 4 ports for port profile conversion, except port 9 and 10. This means that ports 1-4, 5-8, 11-14, and 15-18 are part of a port group. For example, if port 2 is to be converted to an uplink, ports 1-4 all need to be converted. • Port 9 and 10 are a port group of 2. If port 10 is to be converted as an uplink, port 9 must become an uplink, also.
LEM Mismatch	On the N9K-C9400-SUP-A, if you have an N9K-X9400-22L LEM with its port profile configured on ports 7-18 and you replace it with another LEM type, like the N9K-X9400-8D or N9K-X9400-16W, the 8D or 16W module will show module status: **LEM type mismatch**. The failure occurs because the port profile conversions do not match. For the 8D and 16W modules, the port profile conversion is ports 1-6 only. To recover the 8D or 16W LEMs from mismatch, remove the port-profile configurations present on all the ports on the previous LEM (i.e., the N9K-X9400-22L LEM). There should not be any port profile configuration. After this process, do a clean reload to recover the LEMs.
Decommissioning nodes with port profiles	If a decommissioned node has the Port Profile feature deployed on it, the port conversions are not removed even after decommissioning the node. It is necessary to manually delete the configurations after decommission, for the ports to return to the default state. To do this, log onto the switch, run the `setup-clean-config.sh` script, and wait for it to run. Then, enter the `reload` command. Optionally, you can specify `-k` with the `setup-clean-config.sh` script to allow the port-profile setting to persist across the reload, making an additional reboot unnecessary. Beginning with 6.0(5), the port-profile setting persists across the reload when running the `setup-clean-config.sh` script with no option, `-k` or `--keep-port-profile`. To manually delete the configuration, run the `setup-clean-config.sh` script with `-d` or `--delete-profiles`.

Subject	Guideline
Maximum uplink port limit	When the maximum uplink port limit is reached and ports 25 and 27 are converted from uplink to downlink and back to uplink on Cisco 93180LC-EX switches: On Cisco N9K-93180LC-EX switches, ports 25 and 27 are the original uplink ports. Using the port profile, if you convert port 25 and 27 to downlink ports, ports 29, 30, 31, and 32 are still available as four original uplink ports. Because of the threshold on the number of ports (which is maximum of 12 ports) that can be converted, you can convert 8 more downlink ports to uplink ports. For example, ports 1, 3, 5, 7, 9, 13, 15, 17 are converted to uplink ports and ports 29, 30, 31 and 32 are the 4 original uplink ports (the maximum uplink port limit on Cisco 93180LC-EX switches). When the switch is in this state and if the port profile configuration is deleted on ports 25 and 27, ports 25 and 27 are converted back to uplink ports, but there are already 12 uplink ports on the switch (as mentioned earlier). To accommodate ports 25 and 27 as uplink ports, 2 random ports from the port range 1, 3, 5, 7, 9, 13, 15, 17 are denied the uplink conversion and this situation cannot be controlled by the user. Therefore, it is mandatory to clear all the faults before reloading the leaf node to avoid any unexpected behavior regarding the port type. It should be noted that if a node is reloaded without clearing the port profile faults, especially when there is a fault related to limit-exceed, the port might not be in an expected operational state.

Breakout Limitations

Switch	Releases	Limitations
N9K-C93180LC-EX	Cisco APIC 3.1(1) and later	• 40Gb and 100Gb dynamic breakouts are supported on ports 1 through 24 on odd numbered ports. • When the top ports (odd ports) are broken out, then the bottom ports (even ports) are error disabled. • Port profiles and breakouts are not supported on the same port. However, you can apply a port profile to convert a fabric port to a downlink, and then apply a breakout configuration.

Switch	Releases	Limitations
N9K-C9336C-FX2-E	Cisco APIC 5.2(4) and later	• 40Gb and 100Gb dynamic breakouts are supported on ports 1 through 34. • A port profile cannot be applied to a port with breakout enabled. However, you can apply a port profile to convert a fabric port to a downlink, and then apply a breakout configuration. • All 34 ports can be configured as breakout ports. • If you want to apply a breakout configuration on 34 ports, you must configure a port profile on the ports to have 34 downlink ports, then you must reboot the leaf switch. • If you apply a breakout configuration to a leaf switch for multiple ports at the same time, it can take up to 10 minutes for the hardware of 34 ports to be programmed. The ports remain down until the programming completes. The delay can occur for a new configuration, after a clean reboot, or during switch discovery.
N9K-C9336C-FX2	Cisco APIC 4.2(4) and later	• 40Gb and 100Gb dynamic breakouts are supported on ports 1 through 34. • A port profile cannot be applied to a port with breakout enabled. However, you can apply a port profile to convert a fabric port to a downlink, and then apply a breakout configuration. • All 34 ports can be configured as breakout ports. • If you want to apply a breakout configuration on 34 ports, you must configure a port profile on the ports to have 34 downlink ports, then you must reboot the leaf switch. • If you apply a breakout configuration to a leaf switch for multiple ports at the same time, it can take up to 10 minutes for the hardware of 34 ports to be programmed. The ports remain down until the programming completes. The delay can occur for a new configuration, after a clean reboot, or during switch discovery.

Switch	Releases	Limitations
N9K-C9336C-FX2	Cisco APIC 3.2(1) up through, but not including, 4.2(4)	• 40Gb and 100Gb dynamic breakouts are supported on ports 1 through 30. • Port profiles and breakouts are not supported on the same port. However, you can apply a port profile to convert a fabric port to a downlink, and then apply a breakout configuration. • A maximum of 20 ports can be configured as breakout ports.

N9K-C93180YC-FX	Cisco APIC 3.2(1) and later	• 40Gb and 100Gb dynamic breakouts are supported on ports 49 though 52, when they are on profiled QSFP ports. To use them for dynamic breakout, perform these steps:
		  • Convert ports 49-52 to front panel ports (downlinks).
		  • Perform a port-profile reload, using one of these methods:
		  • In the Cisco APIC GUI, navigate to **Fabric** > **Inventory** > **Pod** > **Leaf**, right-click **Chassis** and choose **Reload**.
		  • In the iBash CLI, enter the **reload** command.
		  • Apply breakouts on the profiled ports 49-52.
		• Ports 53 and 54 do not support either port profiles or breakouts.

N9K-C93240YC-FX2	Cisco APIC 4.0(1) and later	Breakout is not supported on converted downlinks.

Port Profile Configuration Summary

The following table summarizes supported uplinks and downlinks for the switches that support port profile conversions from uplink to downlink and downlink to uplink.

Switch Model	Default Links	Max Uplinks (Fabric Ports)	Max Downlinks (Server Ports)	Release Supported
N9K-C9348GC-FXP[1] N9K-C9348GC-FX3	48 x 100M/1G BASE-T downlinks 4 x 10/25 Gbps SFP28 downlinks 2 x 40/100 Gbps QSFP28 uplinks	48 x 100M/1G BASE-T downlinks 4 x 10/25 Gbps SFP28 uplinks 2 x 40/100 Gbps QSFP28 uplinks	Same as default port configuration	3.1(1) 6.0(5)
N9K-C93180LC-EX	24 x 40 Gbps QSFP28 downlinks (ports 1-24) 2 x 40/100 Gbps QSFP28 uplinks (ports 25, 27) 4 x 40/100 Gbps QSFP28 uplinks (ports 29-32) Or 12 x 100 Gbps QSFP28 downlinks (odd number ports from 1-24) 2 x 40/100 Gbps QSFP28 uplinks (ports 25, 27) 4 x 40/100 Gbps QSFP28 uplinks (ports 29-32)	18 x 40 Gbps QSFP28 downlinks (from 1-24) 6 x 40 Gbps QSFP28 uplinks(from 1-24) 2 x 40/100 Gbps QSFP28 uplinks(25, 27) 4 x 40/100 Gbps QSFP28 uplinks(29-32) Or 6 x 100 Gbps QSFP28 downlinks(odd number from 1-24) 6 x 100 Gbps QSFP28 uplinks(odd number from 1-24) 2 x 40/100 Gbps QSFP28 uplinks(25, 27) 4 x 40/100 Gbps QSFP28 uplinks(29-32)	24 x 40 Gbps QSFP28 downlinks(1-24) 2 x 40/100 Gbps QSFP28 downlinks(25, 27) 4 x 40/100 Gbps QSFP28 uplinks(29-32) Or 12 x 100 Gbps QSFP28 downlinks(odd number from 1-24) 2 x 40/100 Gbps QSFP28 downlinks (25, 27) 4 x 40/100 Gbps QSFP28 uplinks(29-32)	3.1(1)
N9K-C93180YC-EX N9K-C93180YC-FX N9K-C93180YC-FX3	48 x 10/25 Gbps fiber downlinks 6 x 40/100 Gbps QSFP28 uplinks	Same as default port configuration 48 x 10/25 Gbps fiber uplinks 6 x 40/100 Gbps QSFP28 uplinks	48 x 10/25 Gbps fiber downlinks 4 x 40/100 Gbps QSFP28 downlinks 2 x 40/100 Gbps QSFP28 uplinks	3.1(1) 4.0(1) 5.1(3)

Switch Model	Default Links	Max Uplinks (Fabric Ports)	Max Downlinks (Server Ports)	Release Supported
N9K-C93108TC-EX[2] N9K-C93108TC-FX[2] N9K-C93108TC-FX3	48 x 10GBASE-T downlinks 6 x 40/100 Gbps QSFP28 uplinks	Same as default port configuration	48 x 10/25 Gbps fiber downlinks 4 x 40/100 Gbps QSFP28 downlinks 2 x 40/100 Gbps QSFP28 uplinks	3.1(1)
				4.0(1)
				5.1(3)
N9K-C9336C-FX2	30 x 40/100 Gbps QSFP28 downlinks 6 x 40/100 Gbps QSFP28 uplinks	18 x 40/100 Gbps QSFP28 downlinks 18 x 40/100 Gbps QSFP28 uplinks	Same as default port configuration	3.2(1)
		18 x 40/100 Gbps QSFP28 downlinks 18 x 40/100 Gbps QSFP28 uplinks	34 x 40/100 Gbps QSFP28 downlinks 2 x 40/100 Gbps QSFP28 uplinks	3.2(3)
		36 x 40/100 Gbps QSFP28 uplinks	34 x 40/100 Gbps QSFP28 downlinks 2 x 40/100 Gbps QSFP28 uplinks	4.1(1)
N9K-C9336C-FX2-E	30 x 40/100 Gbps QSFP28 downlinks 6 x 40/100 Gbps QSFP28 uplinks	36 x 40/100 Gbps QSFP28 uplinks	34 x 40/100 Gbps QSFP28 downlinks 2 x 40/100 Gbps QSFP28 uplinks	5.2(4)
N9K-93240YC-FX2	48 x 10/25 Gbps fiber downlinks 12 x 40/100 Gbps QSFP28 uplinks	Same as default port configuration	48 x 10/25 Gbps fiber downlinks 10 x 40/100 Gbps QSFP28 downlinks 2 x 40/100 Gbps QSFP28 uplinks	4.0(1)
		48 x 10/25 Gbps fiber uplinks 12 x 40/100 Gbps QSFP28 uplinks		4.1(1)
N9K-C93216TC-FX2	96 x 10G BASE-T downlinks 12 x 40/100 Gbps QSFP28 uplinks	Same as default port configuration	96 x 10G BASE-T downlinks 10 x 40/100 Gbps QSFP28 downlinks 2 x 40/100 Gbps QSFP28 uplinks	4.1(2)

Switch Model	Default Links	Max Uplinks (Fabric Ports)	Max Downlinks (Server Ports)	Release Supported
N9K-C93360YC-FX2	96 x 10/25 Gbps SFP28 downlinks 12 x 40/100 Gbps QSFP28 uplinks	44 x 10/25Gbps SFP28 downlinks 52 x 10/25Gbps SFP28 uplinks 12 x 40/100Gbps QSFP28 uplinks	96 x 10/25 Gbps SFP28 downlinks 10 x 40/100 Gbps QSFP28 downlinks 2 x 40/100 Gbps QSFP28 uplinks	4.1(2)
N9K-C93600CD-GX	28 x 40/100 Gbps QSFP28 downlinks (ports 1-28) 8 x 40/100/400 Gbps QSFP-DD uplinks (ports 29-36)	28 x 40/100 Gbps QSFP28 uplinks 8 x 40/100/400 Gbps QSFP-DD uplinks	28 x 40/100 Gbps QSFP28 downlinks 6 x 40/100/400 Gbps QSFP-DD downlinks 2 x 40/100/400 Gbps QSFP-DD uplinks	4.2(2)
N9K-C9364C-GX	48 x 40/100 Gbps QSFP28 downlinks (ports 1-48) 16 x 40/100 Gbps QSFP28 uplinks (ports 49-64)	56 x 40/100 Gbps QSFP28 uplinks	62 x 40/100 Gbps QSFP28 downlinks 2 x 40/100 Gbps QSFP28 uplinks	4.2(3)
N9K-C9316D-GX	12 x 40/100/400 Gbps QSFP-DD downlinks (ports 1-12) 4 x 40/100/400 Gbps QSFP-DD uplinks (ports 13-16)	16 x 40/100/400 Gbps QSFP-DD uplinks	14 x 40/100/400 Gbps QSFP-DD downlinks	5.1(4)
N9K-C9332D-GX2B	2 x 1/10 Gbps SFP+ downlinks (ports 33-34) 24 x 40/100/400 Gbps QSFP-DD downlinks (ports 1-24) 8 x 40/100/400 Gbps QSFP-DD uplinks (ports 25-32)	2 x 1/10 Gbps SFP+ downlinks 32 x 40/100/400 Gbps QSFP-DD uplinks	2 x 1/10 Gbps SFP+ downlinks 30 x 40/100/400 Gbps QSFP-DD downlinks 2 x 40/100/400 Gbps QSFP-DD uplinks	5.2(3)

Switch Model	Default Links	Max Uplinks (Fabric Ports)	Max Downlinks (Server Ports)	Release Supported
N9K-C9348D-GX2A	2 x 1/10 Gbps SFP+ downlinks (ports 49-50) 36 x 40/100/400 Gbps QSFP-DD downlinks (ports 1-36) 12 x 40/100/400 Gbps QSFP-DD uplinks (ports 37-48)	2 x 1/10 Gbps SFP+ downlinks 48 x 40/100/400 Gbps QSFP-DD uplinks	2 x 1/10 Gbps SFP+ downlinks 46 x 40/100/400 Gbps QSFP-DD downlinks 2 x 40/100/400 Gbps QSFP-DD uplinks	5.2(5)
N9K-C9364D-GX2A	2 x 1/10 Gbps SFP+ downlinks (ports 65-66) 48 x 40/100/400 Gbps QSFP-DD downlinks (ports 1-48) 16 x 40/100/400 Gbps QSFP-DD uplinks (ports 49-64)	2 x 1/10 Gbps SFP+ downlinks 56 x 40/100/400 Gbps QSFP-DD uplinks	2 x 1/10 Gbps SFP+ downlinks 62 x 40/100/400 Gbps QSFP-DD downlinks 2 x 40/100/400 Gbps QSFP-DD uplinks	5.2(5)
N9K-C9408 with N9K-X9400-8D[3]	6 x 40/100/400 Gbps QSFP-DD downlinks 2 x 40/100/400 Gbps QSFP-DD uplinks	8 x 40/100/400 Gbps QSFP-DD uplinks	Same as default port configuration	6.0(2)
N9K-C9408 with N9K-X9400-16W[3]	12 x 100/200 Gbps QSFP56 downlinks 4 x 100/200 Gbps QSFP56 uplinks	6 x 100/200 Gbps QSFP56 uplinks (ports 1-6) 6 x 100/200 Gbps QSFP56 downlinks (ports 7-12) 4 x 100/200 Gbps QSFP56 uplinks (ports 13-16)	Same as default port configuration	6.0(2)[4]

1 Does not support FEX.

2 Only uplink to downlink conversion is supported.

3 Only ports 1 through 6 support port profile conversion.

4 The 6.0(2) release does not support 200 Gbps.

Changing an Uplink to a Downlink or Downlink to an Uplink Using the GUI

This procedure explains how to configure a port profile, which determines the port type: uplink or downlink. You can configure ports as uplinks or downlinks by using **Fabric** > **Access Policies** > **Interface Configuration** > **Actions** > **Convert Interfaces**. You can also use **Fabric** > **Inventory** > **Topology** > **Convert Interfaces**. The two methods provide the same workflow.

Before you begin

- The Cisco Application Centric Infrastructure (ACI) fabric is installed, Cisco Application Policy Infrastructure Controller (APIC) are online, and the Cisco APIC cluster is formed and healthy.
- An Cisco APIC fabric administrator account is available that will enable creating or modifying the necessary fabric infrastructure configurations.
- The target leaf switches are registered in the Cisco ACI fabric and available.

Procedure

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

Step 2 In the Navigation pane, choose **Interface Configuration**.

Step 3 In the Work pane, choose **Actions** > **Convert Interfaces**.

Step 4 In the **Interface Configuration Support Type** drop-down list, choose **Convert to Uplink** or **Convert to Downlink**.

Step 5 For the **Node** field, click **Select Node** and select the nodes

Step 6 In the **Interfaces for All Switches** field, enter the desired interfaces.

After converting a downlink to uplink or uplink to downlink, you must reload the switch using the GUI or CLI `reload` command. Power cycling the switch will not work.

Changing an Uplink to a Downlink or Downlink to an Uplink Using the NX-OS-Style CLI

To configure a port profile using the NX-OS-style CLI, perform the following steps:

Before you begin

- The Cisco Application Centric Infrastructure (ACI) fabric is installed, Cisco Application Policy Infrastructure Controller (APIC) are online, and the Cisco APIC cluster is formed and healthy.
- An Cisco APIC fabric administrator account is available that will enable creating or modifying the necessary fabric infrastructure configurations.
- The target leaf switches are registered in the Cisco ACI fabric and available.

Procedure

Step 1 **configure**

Enters global configuration mode.

Example:

```
apic1# configure
```

Step 2 **leaf** *node-id*

Specifies the leaf switches to be configured.

Example:

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

Step 3 **interface** *type*

Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet *slot* / *port*.

Example:

```
apic1(config-leaf)# interface ethernet 1/2
```

Step 4 **port-direction** {**uplink** | **downlink**}

Determines the port direction or changes it. This example configures the port to be a downlink.

On the N9K-C9336C-FX switch, changing a port from uplink to downlink is not supported.

Example:

```
apic1(config-leaf-if)# port-direction downlink
```

Step 5 Log on to the leaf switch where the port is located and enter the **reload** command.

Verifying Port Profile Configuration and Conversion Using the NX-OS Style CLI

You can verify the configuration and the conversion of the ports using the **show interface brief** CLI command.

Note Port profile can be deployed only on the top ports of a Cisco N9K-C93180LC-EX switch, for example, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, and 23. When the top port is converted using the port profile, the bottom ports are hardware disabled. For example, if Eth 1/1 is converted using the port profile, Eth 1/2 is hardware disabled.

Procedure

Step 1 This example displays the output for converting an uplink port to downlink port. Before converting an uplink port to downlink port, the output is displayed in the example. The keyword **routed** denotes the port as uplink port.

Example:

```
switch# show interface brief
<snip>
Eth1/49          --      eth  routed  down   sfp-missing             100G(D)   --
Eth1/50          --      eth  routed  down   sfp-missing             100G(D)   --
<snip>
```

Step 2 After configuring the port profile and reloading the switch, the output is displayed in the example. The keyword **trunk** denotes the port as downlink port.

Example:

```
switch# show interface brief
<snip>
Eth1/49          0       eth  trunk   down   sfp-missing             100G(D)   --
Eth1/50          0       eth  trunk   down   sfp-missing             100G(D)   --
<snip>
```

Editing an Interface Configuration

This procedure describes how to edit the configuration of an interface that you previously configured, which enables you to change the port policy group or description of the interface.

Before you begin

You must have at least one interface configured.

Procedure

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

Step 2 In the Navigation pane, choose **Interface Configuration**.

Step 3 In the work pane, click the **...** at the right end of the row for the interface whose configuration you want to edit, then choose **Edit Interface Configuration**.

Step 4 In the **Edit Policy Group for *interface-name*** dialog, change the configuration as required.

Step 5 Click **Save**.

Note
For any existing configurations done using the node or port profile, you can migrate the entire FEX configuration using the APIC REST API Configuration Procedures.