Using the NX-OS Style CLI

This chapter contains the following sections:

Accessing the NX-OS Style CLI

Note From Cisco APIC Release 1.0 until Release 1.2, the default CLI was a Bash shell with commands to directly operate on managed objects (MOs) and properties of the Management Information Model. Beginning with Cisco APIC Release 1.2, the default CLI is a NX-OS style CLI. The object model CLI is available by typing the **bash** command at the initial CLI prompt.

Procedure

Step 1 From a secure shell (SSH) client, open an SSH connection to APIC at *username@ip-address*.

Use the administrator login name and the out-of-band management IP address that you configured during the initial setup. For example, admin@192.168.10.1.

Step 2 When prompted, enter the administrator password.

What to Do Next

When you enter the NX-OS style CLI, the initial command level is the EXEC level. From this level, you can reach these configuration modes:

- To continue in the NX-OS style CLI, you can stay in EXEC mode or you can type **configure** to enter global configuration mode.

  For information about NX-OS style CLI commands, see the *Cisco APIC NX-OS Style CLI Command Reference*.

- To reach the object model CLI, type **bash**.

  For information about object mode CLI commands, see the *Cisco APIC Command-Line Interface User Guide, APIC Releases 1.0 and 1.1*.

Using the NX-OS Style CLI for APIC

Using CLI Command Modes

The NX-OS style CLI is organized in a hierarchy of command modes with EXEC mode as the root, containing a tree of configuration submodes beginning with global configuration mode. The commands available to you depend on the mode you are in. To obtain a list of available commands in any mode, type a question mark (?) at the system prompt.

This table lists and describes the two most commonly used modes (EXEC and global configuration) along with an example submode (DNS). The table shows how to enter and exit the modes, and the resulting system prompts. The system prompt helps to identify which mode you are in and the commands that are available to you in that mode.

Mode	Access Method	Prompt	Exit Method
EXEC	From the APIC prompt, enter execsh.	`apic#`	To exit to the login prompt, use the **exit** command.
Global configuration	From EXEC mode, enter the configure command.	`apic(config)#`	To exit from a configuration submode to its parent mode, use the **exit** command. To exit from any configuration mode or submode to EXEC mode, use the **end** command.
DNS configuration	From global configuration mode, enter the dns command.	`apic(config-dns)#`	

CLI Command Hierarchy

Configuration mode has several submodes, with commands that perform similar functions grouped under the same level. For example, all commands that display information about the system, configuration, or hardware are grouped under the **show** command, and all commands that allow you to configure the switch are grouped under the **configure** command.

To execute a command that is not available in EXEC mode, you navigate to its submode starting at the top level of the hierarchy. For example, to configure DNS settings, use the **configure** command to enter the global configuration mode, then enter the **dns** command. When you are in the DNS configuration submode, you can query the available commands. as in this example:

```
apic1# configure
apic1(config)# dns
apic1(config-dns)# ?
 address   Configure the ip address for dns servers
 domain    Configure the domains for dns servers
 exit      Exit from current mode
 fabric    Show fabric related information
 no        Negate a command or set its defaults
 show      Show running system information
 use-vrf   Configure the management vrf for dns servers
 where     Show the current mode

apic1(config-dns)# end
apic1#
```

Each submode places you further down in the prompt hierarchy. To view the hierarchy for the current mode, use the **configure** command, as shown in this example:

```
apic1# where
exec
apic1# configure
apic1(config)# pod 1
apic1(config-pod)# ntp
apic1(config-ntp)# where
configure; pod 1; ntp
```

To leave the current level and return to the previous level, type **exit**. To return directly to the EXEC level, type **end**.

EXEC Mode Commands

When you start a CLI session, you begin in EXEC mode. From EXEC mode, you can enter configuration mode. Most EXEC commands are one-time commands, such as show commands, which display the current configuration status.

Configuration Mode Commands

Configuration mode allows you to make changes to the existing configuration. When you save the configuration, these commands are saved across switch reboots. Once you are in configuration mode, you can enter a variety of protocol-specific modes. Configuration mode is the starting point for all configuration commands.

Listing Commands and Syntax

In any command mode, you can obtain a list of available commands by entering a question mark (?).

```
apic1(config-dns)# ?
 address   Configure the ip address for dns servers
 domain    Configure the domains for dns servers
 exit      Exit from current mode
 fabric    Show fabric related information
 no        Negate a command or set its defaults
 show      Show running system information
 use-vrf   Configure the management vrf for dns servers
 where     Show the current mode

apic1(config-dns)# end
apic1#
```

To see a list of commands that begin with a particular character sequence, type those characters followed by a question mark (?). Do not include a space before the question mark.

```
apic1# sh?
 show       show running system information
 shutdown   shutdown controller

apic1#
```

To complete a command after you begin typing, type a tab.

```
apic1# shu<TAB>
apic1# shutdown
```

To list keywords or arguments, enter a question mark in place of a keyword or argument. Include a space before the question mark. This form of help is called command syntax help because it reminds you which keywords or arguments are applicable based on the commands, keywords, and arguments you have already entered.

```
apic1(config-dns)# use-vrf ?
 inband-mgmt  Configure dns on inband
 oob-mgmt     Configure dns on out-of-band

apic1(config-dns)#
```

You can also abbreviate a command if the abbreviation is unambiguous. In this example, the **configure** command is abbreviated.

```
apic1# conf
apic1(config)#
```

Undoing or Reverting to Default Values or Conditions Using the 'no' Prefix

For many configuration commands, you can precede the command with the **no** keyword to remove a setting or to restore a setting to the default value. This example shows how to remove a previously-configured DNS address from the configuration.

```
apic1(config-dns)# address 192.0.20.123 preferred
apic1(config-dns)# show dns-address
 Address             Preferred
 ------------------- ---------
 192.0.20.123        yes

apic1(config-dns)# no address 192.0.20.123
apic1(config-dns)# show dns-address
 Address             Preferred
 ------------------- ---------
```

Executing BASH Commands From the NX-OS Style CLI

To execute a single command in the bash shell, type **bash -c '***path/command***'** as shown in this example.

```
apic1# bash -c '/controller/sbin/acidiag avread'
```

You can execute a bash command from any mode or submode in the NX-OS style CLI.

Entering Configuration Text with Spaces or Special Characters

When a configuration field consists of user-defined text, special characters such as '$' should be escaped ('\$') or the entire word or string should be wrapped in single quotes to avoid misinterpretation by Bash.

About Getting Started with APIC Examples

The steps in several examples in this guide include a parameter name. These parameter names are provided as examples for convenience and ease of your understanding, and it is not required for you to use them.

Switch Discovery with the APIC

The APIC is a central point of automated provisioning and management for all the switches that are part of the ACI fabric. A single data center might include multiple ACI fabrics; each data center might have its own APIC cluster and Cisco Nexus 9000 Series switches that are part of the fabric. To ensure that a switch is managed only by a single APIC cluster, each switch must be registered with that specific APIC cluster that manages the fabric.

The APIC discovers new switches that are directly connected to any switch it currently manages. Each APIC instance in the cluster first discovers only the leaf switch to which it is directly connected. After the leaf switch is registered with the APIC, the APIC discovers all spine switches that are directly connected to the leaf switch. As each spine switch is registered, that APIC discovers all the leaf switches that are connected to that spine switch. This cascaded discovery allows the APIC to discover the entire fabric topology in a few simple steps.

Switch Registration with the APIC Cluster

Note Before you begin registering a switch, make sure that all switches in the fabric are physically connected and booted in the desired configuration. For information about the installation of the chassis, see http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/products-installation-guides-list.html.

After a switch is registered with the APIC, the switch is part of the APIC-managed fabric inventory. With the Application Centric Infrastructure fabric (ACI fabric), the APIC is the single point of provisioning, management, and monitoring for switches in the infrastructure.

Note The infrastructure IP address range must not overlap with other IP addresses used in the ACI fabric for in-band and out-of-band networks.

Registering Unregistered Switches Using the NX-OS Style CLI

Note The infrastructure IP address range must not overlap with other IP addresses used in the ACI fabric for in-band and out-of-band networks.

Procedure

Step 1 Start in Configuration mode, shown as follows:

Example:

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

Step 2 Register the switches, as shown in the following example:

Note To obtain the serial number, find the serial number that is physically printed on the node itself, or use the command **acidiag fnvread** for a list of discovered node serial numbers.

Example:

```
apic1(config)# system switch-id FGE173900ZD 101 leaf1
```

Step 3 Repeat the previous step for the remaining switches.

Switch Discovery Validation and Switch Management from the APIC

After the switches are registered with the APIC, the APIC performs fabric topology discovery automatically to gain a view of the entire network and to manage all the switches in the fabric topology.

Each switch can be configured, monitored, and upgraded from the APIC without having to access the individual switches.

Configuring Network Time Protocol

Time Synchronization and NTP

Within the Cisco Application Centric Infrastructure (ACI) fabric, time synchronization is a crucial capability upon which many of the monitoring, operational, and troubleshooting tasks depend. Clock synchronization is important for proper analysis of traffic flows as well as for correlating debug and fault time stamps across multiple fabric nodes.

An offset present on one or more devices can hamper the ability to properly diagnose and resolve many common operational issues. In addition, clock synchronization allows for the full utilization of the atomic counter capability that is built into the ACI upon which the application health scores depend. Nonexistent or improper configuration of time synchronization does not necessarily trigger a fault or a low health score. You should configure time synchronization before deploying a full fabric or applications so as to enable proper usage of these features. The most widely adapted method for synchronizing a device clock is to use Network Time Protocol (NTP).

Prior to configuring NTP, consider what management IP address scheme is in place within the ACI fabric. There are two options for configuring management of all ACI nodes and Application Policy Infrastructure Controllers (APICs), in-band management and/or out-of-band management. Depending upon which management option is chosen for the fabric, configuration of NTP will vary. Another consideration in deploying time synchronization is where the time source is located. The reliability of the source must be carefully considered when determining if you will use a private internal clock or an external public clock.

In-Band Management NTP

Note

- Make sure the Management EPG is configured for the NTP servers, otherwise the servers will not get configured on the switches.
- See the Adding Management Access section in this guide for information about in-band management access.

- In-Band Management NTP—When an ACI fabric is deployed with in-band management, consider the reachability of the NTP server from within the ACI in-band management network. In-band IP addressing used within the ACI fabric is not reachable from anywhere outside the fabric. To leverage an NTP server external to the fabric with in-band management, construct a policy to enable this communication..

NTP over IPv6

NTP over IPv6 addresses is supported in hostnames and peer addresses. The gai.conf can also be set up to prefer the IPv6 address of a provider or a peer over an IPv4 address. The user can provide a hostname that can be resolved by providing an IP address (both IPv4 or IPv6, depending on the installation or preference).

Configuring NTP Using the NX-OS Style CLI

When an ACI fabric is deployed with out-of-band management, each node of the fabric is managed from outside the ACI fabric. You can configure an out-of-band management NTP server so that each node can individually query the same NTP server as a consistent clock source.

Procedure

Step 1 **configure**

Enters configuration mode.

Example:

```
apic1# configure
```

Step 2 **pod** *pod-number*

Specifies the pod to be configured.

Example:

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

Step 3 **ntp**

Enters NTP configuration mode.

Example:

```
apic1(config-pod)# ntp
```

Step 4 **[no] server** *name-or-ipaddress* **[prefer]** [**use-vrf** {**inband-mgmt** | **oob-mgmt**}] [**key** *key-value*]

Configures an NTP server for the default NTP policy. To make this server the preferred default NTP server, include the **prefer** keyword. If NTP authentication is enabled, specify a reference key ID.

Example:

```
apic1(config-ntp)# server 192.0.20.123 prefer use-vrf oob-mgmt
```

Step 5 **[no] authenticate**

Enables (or disables) NTP authentication.

Example:

```
apic1(config-ntp)# no authenticate
```

Step 6 **[no] authentication-key** *key-value*

Configures an authentication NTP authentication. The range is 1 to 65535.

Example:

```
apic1(config-ntp)# authentication-key 12345
```

Step 7 **[no] trusted-key** *key-value*

Configures a trusted NTP authentication. The range is 1 to 65535.

Example:

```
apic1(config-ntp)# trusted-key 54321
```

Step 8 **end**

Returns to EXEC mode.

Example:

```
apic1(config-ntp)# end
```

Examples

This example shows how to configure a preferred out-of-band NTP server and how to verify the configuration and deployment.

```
apic1# configure
apic1(config)# pod 1
apic1(config-pod)# ntp
apic1(config-ntp)# server 192.0.20.123 prefer use-vrf oob-mgmt
apic1(config-ntp)# end
apic1# show run pod 1 ntp
# Command: show running-config pod 1 ntp
# Time: Mon Nov 9 17:17:18 2015
  pod 1
    ntp
      server 192.0.20.123 prefer use-vrf oob-mgmt
    exit
  exit

apic1# show ntpq
 nodeid     remote        refid   st    t   when   poll   reach  delay    offset   jitter

 ------  -  ------------  ------  ----  --  -----  -----  -----  ------   ------   ------
 1       *  192.0.20.123  .GPS.         u   27     64     377    76.427   0.087    0.067

 2       *  192.0.20.123  .GPS.         u   3      64     377    75.932   0.001    0.021
 3       *  192.0.20.123  .GPS.         u   3      64     377    75.932   0.001    0.021
```

Verifying NTP Operation Using the NX-OS Style CLI

Procedure

Verify that the NTP policy is deployed to APIC using the NX-OS CLI using **show ntp**:

Example:

```
apic1# show ntp
```

Verifying NTP Policy Deployed to Each Node Using the NX-OS Style CLI

Procedure

Step 1 Log onto an APIC controller in the fabric using the SSH protocol.

Step 2 Attach to a node and check the NTP peer status, shown as follows:

```
apic1# fabric  node_name show ntp peer-status
```

Step 3 Repeat step 2 for different nodes in the fabric.

Creating User Accounts

Configuring a Local User Using the NX-OS Style CLI

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

AV Pair on the External Authentication Server

You can add a Cisco attribute/value (AV) pair to the existing user record to propagate the user privileges to the APIC controller. The Cisco AV pair is a single string that you use to specify the Role-Based Access Control (RBAC) roles and privileges for an APIC user. An example configuration for an open RADIUS server (/etc/raddb/users) is as follows:

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

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

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

Procedure

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

Example:

```
apic1#
apic1# configure
```

Step 2 Configure the aaa user default role.

Example:

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

Step 3 Configure the aaa authentication login methods.

Example:

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

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

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

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

Best Practice for Assigning AV Pairs

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

Configuring an AV Pair on the External Authentication Server

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

Procedure

Configure an AV pair on the external authentication server.
The Cisco AV pair definition is as follows (Cisco supports AV pairs with and without UNIX user IDs specified):

Example:

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

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

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

The following is an example:

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

Configuring a Remote User Using the NX-OS Style CLI

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

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

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

Configuring a Remote User With the NX-OS Style CLI

Procedure

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

Example:

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

Step 2 Create a RADIUS provider, shown in the following example:

Example:

```
apic1(config)# radius-server
 host     RADIUS server's DNS name or its IP address
 retries  Global RADIUS server retransmit count
 timeout  Global RADIUS server timeout period in seconds

apic1(config)# radius-server host 1.1.1.1
apic1(config-host)#
 descr     RADIUS server descr for authentication
 exit      Exit from current mode
 fabric    show fabric related information
 key       RADIUS server key for authentication
 no        Negate a command or set its defaults
 port      RADIUS server port for authentication
 protocol  RADIUS server protocol for authentication
 retries   RADIUS server retries for authentication
 show      Show running system information
 timeout   RADIUS server timeout for authentication
 where     show the current mode

apic1(config-host)# exit
```

Step 3 Create a TACACS+ provider, shown in the following example:

Example:

```
 apic1(config)# tacacs-server
 host     TACACS+ server's DNS name or its IP address
 retries  Global TACACS+ server retries period in seconds
 timeout  Global TACACS+ server timeout period in seconds

apic1(config)# tacacs-server host 1.1.1.1
apic1(config-host)# exit
```

Step 4 Create an LDAP provider, shown in the following example:

Example:

```
apic1(config)# ldap-server
 attribute  An LDAP endpoint attribute to be used as the CiscoAVPair
 basedn     The LDAP base DN for user lookup in the LDAP directory tree
 filter     LDAP search filter for the LDAP endpoint
 host       LDAP server DNS name or IP address
 retries    Global LDAP server retransmit count
 timeout    Global LDAP server timeout period in seconds

apic1(config)# ldap-server host 1.1.1.1
apic1(config-host)#
```

```
 enable-ssl             enabling an SSL connection with the LDAP provider
 exit                   Exit from current mode
 fabric                 show fabric related information
 filter                 Set the LDAP filter to be used in a user search
 key                    LDAP server key for authentication
 no                     Negate a command or set its defaults
 port                   LDAP server port for authentication
 retries                LDAP server retries for authentication
 show                   Show running system information
 ssl-validation-level  Set the LDAP Server SSL Certificate validation level
 timeout                LDAP server timeout for authentication
 where                  show the current mode

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

Adding Management Access

IPv4/IPv6 Addresses and In-Band Policies

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

Adding Management Access Using the NX-OS Style CLI

In-band management access—You can configure in-band management connectivity to the APIC and the ACI fabric. You first configure the VLANs that will be used by APIC when the APIC is communicating with the leaf switches, and then you configure the VLANs that the VMM servers will use to communicate with the leaf switches.

Note Do not configure the APIC selector (the set of leaf ports to which the APIC is connected) when configuring the simulator with in-band management access.

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

Configuring In-Band Management Access for APIC Controller, Spine, Leaf Switches Using the NX-OS CLI

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

Procedure

Step 1 Start in the configuration mode to modify one or more of the IP addresses of the in-band management interface of the APIC controllers, as shown in the following example:

Example:

```
apic1# configure
apic1(config)# controller 1
apic1(config-controller)# interface inband-mgmt0
apic1(config-controller-if)# ip address 10.13.1.1/24 gateway 10.13.1.254
apic1(config-controller-if)# exit
```

Note You wouldn't have this for the inband management interface for the switches.

Step 2 You can configure the in-band management interface of spine and leaf switches by entering into the switch configuration mode. Enter the switch followed by the ID of the switch, shown as follows:

Example:

```
apic1(config)# switch 101
apic1(config-switch)# interface inband-mgmt0
apic1(config-switch-if)# ip address 10.13.1.101/24 gateway 10.13.1.254
```

Note Switch 101 in the above example can be a leaf or spine switch. There are two types of switches (spines and leafs), but for the purposes of management configuration, it does not matter if the switches are spines or leafs, you can use the same configuration for both.

Example:

To configure a range of switches using successive addresses from a IP address pool, the **ip address-range** command can be used. The following example shows how you can configure IP addresses of the inband management ports for a range of switches at the same time.

```
apic1(config)# switch 101-104
apic1(config-switch)# interface mgmt0
apic1(config-switch-if)# ip address-range 172.23.48.21/21 gateway 172.23.48.1
apic1(config-switch-if)# exit
apic1(config-switch)# exit
```

Step 3 To establish connectivity to in-band management ports from the outside network, perform the following configuration steps:

a) Create a VLAN domain for the VLAN used for external in-band connectivity.

Example:

Note In the following example, the Management station that is used to connect to the in-band network is connected to Leaf 102, port 2, on VLAN 11, and is in the subnet 179.10.1.0/24.

```
apic1(config)# vlan-domain external-inband
apic1(config-vlan)# vlan 11
apic1(config-vlan)# exit
```

b) Add the port connected to the external Management station to the VLAN domain and open up the VLAN on the port for in-band connectivity, as shown in the following example:

Example:

Note The address 179.10.1.254/24 is the gateway address used by the external management station and the gateway functionality is provided by the ACI fabric.

```
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 1/2
apic1(config-leaf-if)# switchport trunk allowed vlan 11 inband-mgmt 179.10.1.254/24
```

With the previous configuration, the external management station can connect to the in-band port on the spine and leaf switches.

For connectivity to the APIC controller inband port, additional configuration is required in order to open up the VLAN on the port connected to the controller, as explained in the following example. Controller 1 is connected to port on Ethernet 1/1 on leaf 110 and VLAN 10 is used for the APIC controller in-band connectivity.

To configure the in-band VLAN on the controller:

```
apic1(config)# controller 1
apic1(config-controller)# interface inband-mgmt0
apic1(config-controller-if)# ip address x.x.x.x gateway x.x.x.y
apic1(config-controller-if)# vlan 10
apic1(config-controller-if)# inband-mgmt epg inb-default
```

To create a VLAN domain for the APIC in-band VLAN:

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

To allow the VLAN on the port connected to the controller:

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/1
apic1(config-leaf-if)# vlan-domain member apic-inband
```

Note With the previous configuration, the external management station can connect to the in-band port on the controller. Note that the inband VLAN (VLAN 10 in this example) must be the same on all the controllers.

Step 4 To provide access control for specific protocols on APIC in-band ports from the external network:

Example:

```
apic1(config)# tenant mgmt
apic1(config-tenant)# access-list inband-default
apic1(config-tenant-acl)# no match raw inband-default
apic1(config-tenant-acl)# match tcp dest 443
apic1(config-tenant-acl)# match tcp dest 22
```

In the previous example, "no match raw inband-default" deletes the allow all entry in the default access-list filter. The following match tcp dest 443 and 22 allow access to only these tcp ports on the in-band ports.

What to Do Next

- You must use the new IP address to reconnect to the APIC controller.
- You must delete the old IP address of the controller once a new IP address is assigned to it.

IPv6 Table Modifications to Mirror the Existing IP Tables Functionality

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

Existing IP Tables

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

Modifications to IP Tables

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

   ```
   -A INPUT -s <OOB Address Ipv4/Ipv6> -j apic-default
   ```
5. When a new rule is added, it presents in the IP tables-new file and not in the IP tables file, and it signifies that there is some error in the IP tables-new file. Only if the restoration is successful, the file is saved and new rules are seen in the IP tables file.

Note

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

Configuring a VLAN Domain

Configuring a VLAN Domain Using the NX-OS Style CLI

The ACI fabric can be partitioned into groups of 4K VLANs to allow a large number of Layer 2 (L2) domains across the fabric, which can be used by multiple tenants.

A VLAN domain represents a set of VLANs that can be configured on group of nodes and ports. VLAN domains allow multiple tenants to share the common fabric resources, such as nodes, ports, and VLANs, without conflicting with each other and independently managing them. A tenant can be provided access to one or more VLAN domains.

VLAN domains can be static or dynamic. Static VLAN domains support static VLAN pools, while dynamic VLAN domains can support both static and dynamic VLAN pools. VLANs in the static VLAN pools are managed by the user and are used for applications such as connectivity to bare metal hosts. VLANs in the dynamic VLAN pool are allocated and managed by the APIC without user intervention and are used for applications such as VMM. The default type for VLAN domains and VLAN pools within the domain is static.

You must perform this procedure before tenants can start using the fabric resources for their L2/L3 configurations. For detailed steps with examples on how to use the NX-OS CLI for this procedure, see Creating Tenants, VRFs, and Bridge Domains Using the NX-OS Style CLI, on page 22.

Procedure

Step 1 Create a VLAN domain and assign VLANs in each VLAN domain.

Example:

```
apic1# configure
apic1(config)# vlan-domain dom1
apic1(config-vlan)# vlan 5-100
apic1(config-vlan)# exit
apic1(config)# vlan-domain dom2 dynamic
apic1(config-vlan)# vlan 101-200
apic1(config-vlan)# vlan 301-400 dynamic
apic1(config-vlan)# exit
apic1(config)# vlan-domain dom3
apic1(config-vlan)# vlan 401-500
```

Step 2 Configure the VLAN domain membership on the ports on the leaf switches.

Example:

```
apic1(config)# leaf 101,102
apic1(config-leaf)# interface ethernet 1/10-20
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# vlan-domain member dom2
apic1(config-leaf-if)#exit
apic1(config-leaf)# interface ethernet 1/21
apic1(config-leaf-if)# vlan-domain member dom3
apic1(config-leaf-if)#exit
```

Step 3 Convert some ports to be used as L3 Port for External-L3 connectivity through L3Port or through its sub-interfaces.

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/21
```

```
apic1(config-leaf-if)# no switchport
In this example, sub-interface encapsulations on ethernet1/21 come from vlans allowed in
dom3.
```

Step 4 Verify the configuration.

Related Topics

Creating Tenants, VRFs, and Bridge Domains Using the NX-OS Style CLI, on page 22

Configuring a VMM Domain

Configuring a VMM Domain Using the NX-OS Style CLI

Configuring Virtual Machine Networking Policies

The APIC integrates with third-party VM manager (VMM) (for example, VMware vCenter and SCVMM) to extend the benefits of ACI to the virtualized infrastructure. The APIC enables the ACI policies inside the VMM system to be used by its administrator.

This section provides examples of VMM integration using VMware vCenter and vShield. For details about the different modes of Cisco ACI and VMM integration, see the ACI Virtualization Guide.

About the VM Manager

Note Information about the necessary configuration of the APIC for integration with the vCenter is described here. For instructions about configuring the VMware components, see the VMware documentation.

The following are details of some VM manager terms:

- A VM controller is an external virtual machine management entity such as VMware vCenter, and the VMware vShield. The APIC communicates with the controller to publish network policies that are applied to virtual workloads. A VM controller administrator provides an APIC administrator with a VM controller authentication credential; multiple controllers of the same type can use the same credential.
- A virtual machine mobility domain (vCenter mobility domain) is a grouping of VM controllers with similar networking policy requirements. This mandatory container holds one or more VM controllers with policies such as for a VLAN pool, server to network MTU policy, or server to network access LACP policy. When an endpoint group gets associated with a vCenter domain, network policies get pushed to all the VM controllers in the vCenter domain.
- See Configuring a VLAN Domain Using the NX-OS Style CLI, on page 17 for information about VLAN domains.
- For a VMware VCenter to be deployed, it must operate in VLAN mode or VXLAN mode. A VMM domain must be associated with a VLAN pool and a vShield must be associated with the vCenter.

Prerequisites for Creating a VMM Domain Profile

To configure a VMM domain profile, you must meet the following prerequisites:

- All fabric nodes are discovered and configured.
- Inband (inb) or out-of-band (oob) management has been configured on the APIC.
- A Virtual Machine Manager (VMM) is installed, configured, and reachable through the inb/oob
- You have the administrator/root credential to the VMM (for example vCenter).

Note If you prefer not to use the vCenter admin/root credentials, you can create a custom user account with minimum required permissions. See Custom User Account with Minimum VMware vCenter Privileges , on page 19 for a list of the required user privileges.

- A DNS policy for the APIC must be configured if you plan to reference the VMM by hostname rather than an IP address.
- A DHCP server and relay policy must be configured if you are creating a domain profile for VMware vShield.

Custom User Account with Minimum VMware vCenter Privileges

To configure the vCenter from Cisco APIC, your credentials must allow the following minimum set of privileges within the vCenter:

- Alarms
- Datacenter
- Folder
- Distributed Switch
- dvPortgroup
- Network
- VM
- Host

Creating a VMM Domain Profile

This section describes how to create a VMM domain profile using the NX-OS CLI and provides examples for a vCenter domain or vCenter and vShield domains.

Creating a vCenter Domain Profile Using the NX-OS Style CLI

Before You Begin

This section describes how to create a vCenter domain profile using the NX-OS style CLI:

Procedure

Step 1 In the CLI, enter configuration mode:

Example:

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

Step 2 Configure a VLAN domain:

Example:

```
apic1(config)# vlan-domain dom1 dynamic
apic1(config-vlan)# vlan 150-200 dynamic
apic1(config-vlan)# exit
apic1(config)#
```

Step 3 Add interfaces to this VLAN domain. These are the interfaces to be connected to VMware hypervisor uplink ports:

Example:

```
apic1(config)# leaf 101-102
apic1(config-leaf)# interface ethernet 1/2-3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

Step 4 Create a VMware domain and add VLAN domain membership:

Example:

```
apic1(config)# vmware-domain vmmdom1
apic1(config-vmware)# vlan-domain member dom1
apic1(config-vmware)#
```

Step 5 Configure the domain type to DVS:

Example:

```
apic1(config-vmware)# configure-dvs
apic1(config-vmware-dvs)# exit
apic1(config-vmware)#
```

Step 6 Configure a controller in the domain:

Example:

```
apic1(config-vmware)# vcenter 192.168.66.2 datacenter prodDC
apic1(config-vmware-vc)# username administrator
Password:
Retype password:
apic1(config-vmware-vc)# exit
apic1(config-vmware)# exit
apic1(config)# exit
```

Note When configuring the password, you must precede special characters such as '$' or '!' with a backslash ('\$') to avoid misinterpretation by the Bash shell. The escape backslash is necessary only when configuring the password; the backslash does not appear in the actual password.

Step 7 Verify configuration:

Example:
```
apic1# show running-config vmware-domain vmmdom1
# Command: show running-config vmware-domain vmmdom1
# Time: Wed Sep  2 22:14:33 2015
  vmware-domain vmmdom1
    vlan-domain member dom1
    vcenter 192.168.66.2 datacenter prodDC
      username administrator password *****
    configure-dvs
      exit
    exit
```

Creating a vCenter and a vShield Domain Profile Using the NX-OS Style CLI

Before You Begin

This section describes how to create a vCenter and vShield domain profile using the NX-OS CLI:

Procedure

Step 1 In the NX-OS CLI, enter configuration mode as follows:

Example:
```
apic1# configure
apic1(config)# exit
```

Step 2 Configure a VLAN domain as follows:

Example:
```
apic1(config)# vlan-domain dom1 dynamic
apic1(config-vlan)# vlan 150-200 dynamic
apic1(config-vlan)# exit
apic1(config)#
```

Step 3 Add interfaces to this VLAN domain. These are the interfaces to be connected to VMware hypervisor uplink ports as follows:

Example:
```
apic1(config)# leaf 101-102
apic1(config-leaf)# interface ethernet 1/2-3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)#
```

Step 4 Create a VMware domain and add VLAN domain membership as follows:

Example:

```
apic1(config)# vmware-domain vmmdom1
apic1(config-vmware)# vlan-domain member dom1
apic1(config-vmware)#
```

Step 5 Configure the domain type to DVS as follows:

Example:

```
apic1(config-vmware)# configure-dvs
apic1(config-vmware-dvs)# exit
apic1(config-vmware)#
```

Step 6 Configure a vCenter controller in the domain as follows:

Example:

```
apic1(config-vmware)# vcenter 192.168.66.2 datacenter prodDC
apic1(config-vmware-vc)# username administrator password “password”
apic1(config-vmware-vc)#
```

Step 7 Configure a VShield controller attached to this VCenter, and configure vxlan and multicast address pools for this VShield as follows:

Example:

```
apic1(config-vmware-vc)# vshield 123.4.5.6
apic1(config-vmware-vc-vs)# username administrator password "password"
apic1(config-vmware-vc-vs)# vxlan pool 10000-12000
apic1(config-vmware-vc-vs)# vxlan multicast-pool 224.3.4.5-224.5.6.7
apic1(config-vmware-vc-vs)# exit
apic1(config-vmware-vc)#
```

Step 8 Verify the configuration as follows:

Example:

```
apic1# show running-config vmware-domain vmmdom1
# Command: show running-config vmware-domain vmmdom1
# Time: Wed Sep  2 22:14:33 2015
  vmware-domain vmmdom1
    vlan-domain member dom1
    vcenter 192.168.66.2 datacenter prodDC
      username administrator password *****
      vshield 123.4.5.6
        username administrator password *****
        vxlan pool 10000-12000
        vxlan multicast-pool 224.3.4.5-224.5.6.7
        exit
      exit
    configure-dvs
      exit
    exit
```

Creating Tenants, VRFs, and Bridge Domains

Creating Tenants, VRFs, and Bridge Domains Using the NX-OS Style CLI

This section provides information on how to create tenants, VRFs, and bridge domains.

Note Before creating the tenant configuration, you must create a VLAN domain using the **vlan-domain** command and assign the ports to it.

Procedure

Step 1 Create a VLAN domain (which contains a set of VLANs that are allowable in a set of ports) and allocate VLAN inputs, as follows:

Example:
In the following example ("exampleCorp"), note that VLANs 50 - 500 are allocated.

```
apic1# configure
apic1(config)# vlan-domain dom_exampleCorp
apic1(config-vlan)# vlan 50-500
apic1(config-vlan)# exit
```

Step 2 Once the VLANs have been allocated, specify the leaf (switch) and interface for which these VLANs can be used. Then, enter "vlan-domain member" and then the name of the domain you just created.

Example:
In the following example, these VLANs (50 - 500) have been enabled on leaf 101 on interface ethernet 1/2-4 (three ports including 1/2, 1/3, and 1/4). This means that if you are using this interface, you can use VLANS 50-500 on this port for any application that the VLAN can be used for.

```
apic1(config-vlan)# leaf 101
apic1(config-vlan)# interface ethernet 1/2-4
apic1(config-leaf-if)# vlan-domain member dom_exampleCorp
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

Step 3 Create a tenant in global configuration mode, as shown in the following example:

Example:

```
apic1(config)# tenant exampleCorp
```

Step 4 Create a private network (also called VRF) in tenant configuration mode as shown in the following example:

Example:

```
apic1(config)# tenant exampleCorp
apic1(config-tenant)# vrf context exampleCorp_v1
apic1(config-tenant-vrf)# exit
```

Step 5 Create a bridge domain (BD) under the tenant, as shown in the following example:

Example:

```
apic1(config-tenant)# bridge-domain exampleCorp_b1
apic1(config-tenant-bd)# vrf member exampleCorp_v1
apic1(config-tenant-bd)# exit
```

Note In this case, the VRF is "exampleCorp_v1".

Step 6 Allocate IP addresses for the BD (ip and ipv6), as shown in the following example.

Example:

```
apic1(config-tenant)# interface bridge-domain exampleCorp_b1
apic1(config-tenant-interface)# ip address 172.1.1.1/24
apic1(config-tenant-interface)# ipv6 address 2001:1:1::1/64
apic1(config-tenant-interface)# exit
```

What to Do Next

The next section describes how to add an application profile, create an application endpoint group (EPG), and associate the EPG to the bridge domain.

Related Topics

Configuring a VLAN Domain Using the NX-OS Style CLI, on page 17

Creating an Application Profile and EPG Using the NX-OS Style CLI

Before You Begin

Before you can create an application profile and an application endpoint group (EPG), you must create a VLAN domain, tenant, VRF, and BD (as described in the previous section).

Procedure

Step 1 Create an application profile, as shown in the following example ("exampleCorp_web1"):

Example:

```
apic1(config)# tenant exampleCorp
apic1(config-tenant)# application exampleCorp_web1
```

Step 2 Create an EPG under the application, as shown in the following example ("exampleCorp_webepg1"):

Example:

```
apic1(config-tenant-app)# epg exampleCorp_webepg1
```

Step 3 Associate the EPG to the bridge domain, shown as follows:

Example:

```
apic1(config-tenant-app-epg)# bridge-domain member exampleCorp_b1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
```

Note Every EPG belongs to a BD. An EPG can belong to a BD from the same tenant (or) from tenant Common. If you look at the chain, the lowest end is the EPG, and above that is the BD. The BD belongs to a VRF, and the VRF belongs to the tenant.

What to Do Next

These examples have shown how to configure an application EPG on a tenant. The next section discusses how to map a VLAN on a port to the EPG.

Mapping a VLAN on a Port to the EPG Using the NX-OS Style CLI

This step discusses how to open or enable a VLAN on a port on the leaf switch and associate it with an application EPG. The pre-requisite for this step is that the interface should be a member of a VLAN domain (vlan-domain), which contains this VLAN. Creation of VLAN domain is discussed in Configuring a VLAN Domain Using the NX-OS Style CLI, on page 17.

Procedure

Step 1 Enter into leaf configuration mode by providing the ID of the leaf switch.

Example:

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

Note To apply the same configuration on multiple leaf switches, "-" or "," separated IDs can be used (such as leaf 101-103).

Step 2 Enter the mode shown as follows using the previous example of "interface ethernet 1/2".

Example:

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

Step 3 Enter the command "switchport trunk allowed vlan" followed by the VLAN, then the tenant, application, and the EPG (shown as follows using the previous examples for each):

Example:

```
apic1(config-leaf-if)#switchport trunk allowed vlan 50 tenant exampleCorp application
exampleCorp_web1 epg exampleCorp_webepg1
```

Deploying an Application Policy

Three-Tier Application Deployment

Application profiles enable you to model application requirements that the APIC then automatically renders in the network and data center infrastructure. The application profiles enable administrators to approach the resource pool in terms of applications rather than infrastructure building blocks. The application profile is a container that holds EPGs that are logically related to one another. EPGs can communicate with the other EPGs in the same application profile and with EPGs in other application profiles.

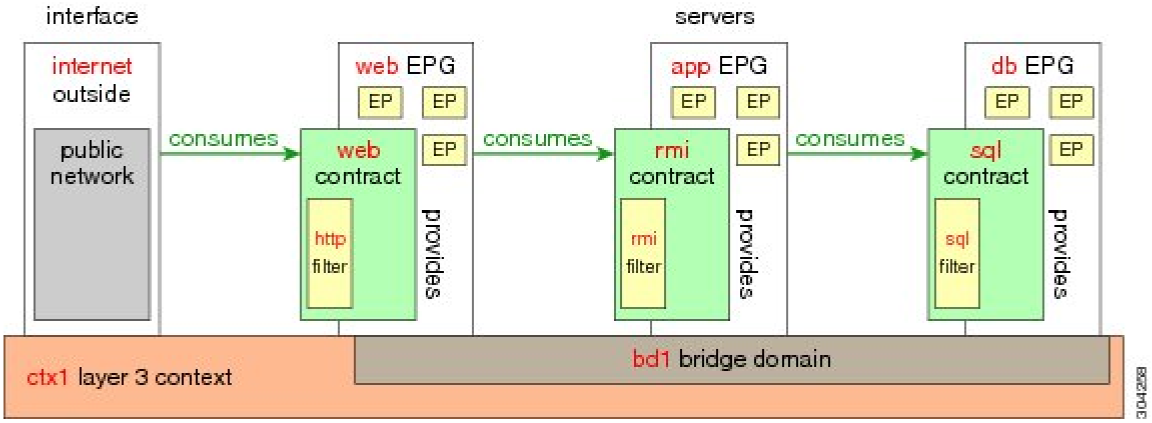
Contracts are policies that enable inter-End Point Group (inter-EPG) communication. These policies are the rules that specify communication between application tiers. If no contract is attached to the EPG, inter-EPG communication is disabled by default. No contract is required for intra-EPG communication because intra-EPG communication is always allowed.

An access list (also referred to as a "filter") specifies the data protocols to be allowed or denied by a contract that contains the access list. A contract can contain multiple subjects. A subject can be used to realize uni- or bidirectional access lists. A unidirectional access list that is used in one direction, either from consumer-to-provider (IN) or from provider-to-consumer (OUT) access list. A bi-directional access list is the same access list that is used in both directions. It is not reflexive.

To deploy an application policy, you must create the required application profiles, access lists (filters), and contracts. Typically, the APIC fabric hosts a three-tier application within a tenant network. In this example, the application is implemented by using three servers (a web server, an application server, and a database server). See the following figure for an example of a three-tier application.

The web server has the HTTP access list, the application server has the Remote Method Invocation (RMI) access list, and the database server has the Structured Query Language (SQL) access list. The application server consumes the SQL contract to communicate with the database server. The web server consumes the RMI contract to communicate with the application server. The traffic enters from the web server and communicates with the application server. The application server then communicates with the database server, and the traffic can also communicate externally.

Figure 1: Three-Tier Application Diagram



Parameters to Create an Access List for HTTP

The parameters to create an access list (filter) for http in this example is as follows:

Parameter Name	Access List (Filter) for HTTP
Name	http
Number of Entries	2
Entry Name	Dport-80 Dport-443

Parameter Name	Access List (Filter) for HTTP
Ethertype	IP
Protocol	tcp tcp
Destination Port	http https

Parameters to Create an Access List for RMI and SQL

The parameters to create filters for RMI and SQL in this example are as follows:

Parameter Name	Filter for RMI	Filter for SQL
Name	rmi	sql
Number of Entries	1	1
Entry Name	Dport-1099	Dport-1521
Ethertype	IP	IP
Protocol	tcp	tcp
Destination Port	1099	1521

Example Application Profile Database

The application profile database in this example is as follows:

EPG	Provided Contracts	Consumed Contracts
web	web	rmi
app	rmi	sql
db	sql	--

Deploying an Application Policy Using the NX-OS Style CLI

The port the EPG uses must belong to one of the VM Managers (VMM) or physical domains associated with the EPG.

Procedure

Step 1 To get into the configuration mode using the NX-OS CLI, enter the following:

Example:

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

Step 2 Create an application network profile for the tenant.
The application network profile in this example is OnlineStore.

Example:

```
apic1(config)# tenant exampleCorp
apic1(config-tenant)# application OnlineStore
apic1(config-tenant-app)#
```

Step 3 Create application web, db, and app EPGs for this application network profile of the tenant.

Example:

```
apic1(config-tenant-app)# epg web
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# epg db
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# epg app
apic1(config-tenant-app-epg)# exit
```

Step 4 Get back into the tenant mode to create an access list (filter) for different traffic types between these EPGs.

Example:

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

Step 5 Create an access list (filter) for the http and https traffic.

Example:

```
apic1(config-tenant)# access-list http
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# match tcp dest 443
apic1(config-tenant-acl)# exit
```

Step 6 Create an access list (filter) for Remote Method Invocation (RMI) traffic.

Example:

```
apic1(config-tenant)# access-list rmi
apic1(config-tenant-acl)# match tcp dest 1099
apic1(config-tenant-acl)# exit
```

Step 7 Create an access list (filter) for the SQL/database traffic.

Example:

```
apic1(config-tenant)# access-list sql
apic1(config-tenant-acl)# match tcp dest 1521
apic1(config-tenant)# exit
```

Step 8 Create the contracts and assign an access group (filters) for RMI traffic between EPGs.

Example:

```
apic1(config)# tenant exampleCorp
apic1(config-tenant)# contract rmi
apic1(config-tenant-contract)# subject rmi
apic1(config-tenant-contract-subj)# access-group rmi both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
```

Step 9 Create the contracts and assign an access group (filters) for web traffic between EPGs.

Example:

```
apic1(config-tenant)# contract web
apic1(config-tenant-contract)# subject web
apic1(config-tenant-contract-subj)# access-group http both
apic1(config-tenant-contract-subj)# exit
```

Step 10 Create the contracts and assign an access group (filters) for SQL traffic between EPGs.

Example:

```
apic1(config-tenant)# contract sql
apic1(config-tenant-contract)# subject sql
apic1(config-tenant-contract-subj)# access-group sql both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
```

Step 11 Attach the bridge domain and contracts to the web EPG.

Example:

```
apic1(config-tenant)# application OnlineStore
apic1(config-tenant-app)# epg web
apic1(config-tenant-app-epg)# bridge-domain member exampleCorp_b1
apic1(config-tenant-app-epg)# contract consumer rmi
apic1(config-tenant-app-epg)# contract provider web
apic1(config-tenant-app-epg)# exit
```

Step 12 Attach the bridge domain and contracts to the db EPG.

Example:

```
apic1(config-tenant-app)# epg db
apic1(config-tenant-app-epg)# bridge-domain member exampleCorp_b1
apic1(config-tenant-app-epg)# contract provider sql
apic1(config-tenant-app-epg)# exit
```

Step 13 Attach the bridge domain and contracts to the application EPG.

Example:

```
apic1(config-tenant-app)# epg app
```

```
apic1(config-tenant-app-epg)# bridge-domain member exampleCorp_b1
```

Step 14 Associate the provider contracts to the application EPGs.

Example:

```
apic1(config-tenant-app-epg)# contract provider rm1
apic1(config-tenant-app-epg)# contract consumer sql
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
```

Step 15 Associate the ports and VLANs to the EPGs app, db, and web.

Example:

```
apic1(config)# leaf 103
apic1(config-leaf)# interface ethernet 1/2-4
apic1(config-leaf-if)# vlan-domain member exampleCorp
apic1(config-leaf)# exit
apic1(config)# leaf 103
apic1(config-leaf)# interface ethernet 1/2
apic1(config-leaf-if)# switchport
access  trunk  vlan
apic1(config-leaf-if)#  switchport trunk allowed vlan 100 tenant exampleCorp application
OnlineStore epg app
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)#  switchport trunk allowed vlan 101 tenant exampleCorp application
OnlineStore epg db
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/4
apic1(config-leaf-if)#  switchport trunk allowed vlan 102 tenant exampleCorp application
OnlineStore epg web
apic1(config-leaf-if)# exit
```

Configuring External L3 Connectivity for Tenants

Configuring an MP-BGP Route Reflector for the ACI Fabric

To distribute routes within the ACI fabric, an MP-BGP process must first be operating, and the spine switches must be configured as BGP route reflectors.

The following is an example of an MP-BGP route reflector configuration:

Note In this example, the BGP fabric ASN is 100. Spine switches 104 and 105 are chosen as MP-BGP route-reflectors.

```
apic1(config)# pod 1
apic1(config-pod)# bgp fabric
apic1(config-pod-bgp)# asn 100
apic1(config-pod-bgp)# route-reflector spine 104,105
```

Creating an OSPF External Routed Network for a Tenant Using the NX-OS CLI

Configuring external routed network connectivity involves the following steps:

1. Create a VRF under Tenant.
2. Configure L3 networking configuration for the VRF on the border leaf switches, which are connected to the external routed network. This configuration includes interfaces, routing protocols (BGP, OSPF, EIGRP), protocol parameters, route-maps.
3. Configure policies by creating external-L3 EPGs under tenant and deploy these EPGs on the border leaf switches. External routed subnets on a VRF which share the same policy within the ACI fabric form one "External L3 EPG" or one "prefix EPG".

Configuration is realized in two modes:

- Tenant mode: VRF creation and external-L3 EPG configuration
- Leaf mode: L3 networking configuration and external-L3 EPG deployment

The following steps are for creating an OSPF external routed network for a tenant. To create an OSPF external routed network for a tenant, you must choose a tenant and then create a VRF for the tenant.

Note The examples in this section show how to provide external routed connectivity to the "web" epg in the "OnlineStore" application for tenant "exampleCorp".

Procedure

Step 1 Configure the VLAN domain.

Example:

```
apic1(config)# vlan-domain dom_exampleCorp
apic1(config-vlan)# vlan 5-1000
apic1(config-vlan)# exit
```

Step 2 Configure the tenant VRF and enable policy enforcement on the VRF.

Example:

```
apic1(config)# tenant exampleCorp
apic1(config-tenant)# vrf context
 exampleCorp_v1
apic1(config-tenant-vrf)# contract enforce
apic1(config-tenant-vrf)# exit
```

Step 3 Configure the tenant BD and mark the gateway IP as “public”. The entry "scope public" makes this gateway address available for advertisement through the routing protocol for external-L3 network.

Example:

```
apic1(config-tenant)# bridge-domain exampleCorp_b1
apic1(config-tenant-bd)# vrf member exampleCorp_v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# interface bridge-domain exampleCorp_b1
```

```
apic1(config-tenant-interface)# ip address 172.1.1.1/24 scope public
apic1(config-tenant-interface)# exit
```

Step 4 Configure the VRF on a leaf.

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf exampleCorp_v1
```

Step 5 Configure the OSPF area and add the route map.

Example:

```
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant exampleCorp vrf exampleCorp_v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.1 route-map map100 out
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
```

Step 6 Assign the VRF to the interface (sub-interface in this example) and enable the OSPF area.

Example:

Note For the sub-interface configuration, the main interface (ethernet 1/11 in this example) must be converted to an L3 port through “no switchport” and assigned a vlan-domain (dom_exampleCorp in this example) that contains the encapsulation VLAN used by the sub-interface. In the sub-interface ethernet1/11.500, 500 is the encapsulation VLAN.

```
apic1(config-leaf)# interface ethernet 1/11
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vlan-domain member dom_exampleCorp
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/11.500
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf exampleCorp_v1
apic1(config-leaf-if)# ip address 157.10.1.1/24
apic1(config-leaf-if)# ip router ospf default area 0.0.0.1
```

Step 7 Configure the external-L3 EPG policy. This includes the subnet to match for identifying the external subnet and consuming the contract to connect with the epg "web".

Example:

```
apic1(config)# tenant t100
apic1(config-tenant)# external-l3 epg  l3epg100
apic1(config-tenant-l3ext-epg)# vrf member v100
apic1(config-tenant-l3ext-epg)# match ip 145.10.1.0/24
apic1(config-tenant-l3ext-epg)# contract consumer web
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)#exit
```

Step 8 Deploy the external-L3 EPG on the leaf switch.

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t100 vrf v100
apic1(config-leaf-vrf)# external-l3 epg l3epg100
```