

Helpful Cisco Links

<https://forge.puppet.com/puppetlabs/ciscopuppet>

<https://supermarket.chef.io/cookbooks/cisco-cookbook>

<https://github.com/datacenter/nxos/tree/master/ansible>

<https://developer.cisco.com/site/nx-os/>

<https://github.com/>

Cisco ACI Overview

As traditional IT departments are under pressure to provide more agility and better outcomes to the business, a new model for operations has emerged, which is called Fast IT
Cisco ACI enables Fast IT by providing a common policy-based operational model across the entire Cisco ACI-ready system
This model drastically reduces cost and complexity

- **Cisco Application-Centric Infrastructure - ACI**
- **Cisco Application Policy Infrastructure Controller - APIC**

Cisco ACI enables Fast IT by providing a common policy-based operational model across the entire Cisco ACI-ready system. This model drastically reduces cost and complexity.

Cisco Nexus 9000 Series Switches

centralized policy management with Cisco APIC

Integrated physical and virtual infrastructure, with mechanisms to simplify initialization and discovery

Fully automated, secure, and redundant network connectivity across multi cloud and multisite

22% of all network outages are caused by human error

The design of ACI is based on the fabric, instead of treating the fabric switches individually

All physical components form the overall system

The ACI fabric appears as a single switch to the outside world, capable of bridging and routing

- Application-policy based model
- Simplifies network management
- Application optimization made easy
- Accelerated provisioning of services

Application-centric fabric connectivity:

The ACI fabric uses a spine-leaf topology, these are Nexus 9000 series switches

These switches relate to high-bandwidth links – which provide which creates an integrated overlay used by host routing

All host traffic that arrives at the ingress leaf is carried over this integrated overlay

- All access links from endpoints are attached to the leaves,
- The leaf switches provide high port density, while the spine switches (minimum of two spines for redundancy) aggregates the fabric bandwidth
- Cisco ACI fabric uses a spine-leaf topology
- High speed fabric
- Minimum of two spines
- High-bandwidth links between the spine and leaf provide transport to an integrated overlay used by host routing
- All host traffic that arrives at the ingress leaf is carried over an integrated overlay
- The Cisco ACI fabric is composed of the Cisco Application Policy Infrastructure Controller (Cisco APIC) and the Cisco Nexus 9000 Series spine and leaf switches.
- The leaf switches are attached to the spines but are never connected to each other
- The spines are attached only to the leaf switches
- The Cisco APIC, and all other endpoints and devices in the data center, are connected to the leaf switches only
- Cisco Application Policy Infrastructure Controller - APIC
- The APIC is the unifying point of automation and management for the ACI fabric

The APIC provides centralized access to all fabric information, optimizes for scale and performance, allowing us to provision applications across physical and virtual resources

ACI virtual machine networking provides hypervisors from multiple vendors programmable and automated access to high-performance scalable virtualized data center infrastructure.

Programmability and automation are critical features of scalable data center virtualization infrastructure. The ACI open REST API enables virtual machine (VM) integration with and orchestration of the policy-model-based ACI fabric. ACI VM networking enables consistent enforcement of policies across both virtual and physical workloads managed by hypervisors from multiple vendors.

Fabric

The ACI fabric decouples the endpoint address (IP or MAC address) from the location of that endpoint and defines the endpoint by its VXLAN tunnel endpoint VTEP address

Forwarding occurs between VTEPs,

Mapping for host and tenant MAC and IP addresses to the VTEP location is performed using a mapping database that is sent to and used by all switches to forward traffic

Mapping for host MAC and IP addresses to the VTEP location is performed using a mapping database that is sent to, and used by, all fabric switches to forward traffic

The fabric of spine-leaf topology is easier to build, test, and support

The symmetrical topology allows for optimized forwarding behavior.

- Every host-to-host connection will traverse two hops.
- Using this design allows a high-bandwidth, low-latency, low oversubscription, and scalable solution at low cost
- Scale is supported by simply adding more leaf nodes if there are not enough ports for connecting host traffic.
- You can add spine nodes if the fabric is not large enough to carry the load of the host traffic.
- In summary, the advantages of the spine-leaf topology include:
- Simple and consistent topology.
- Scalability for connectivity and bandwidth.
- Least-cost design for high bandwidth.
- Low latency and oversubscription.

The Cisco APIC is a central point of automated provisioning and management for all switches that are part of the ACI fabric. This approach enables a simplified fabric initialization and switch discovery process. Once the APIC is set up, it can discover switches that are directly connected, and other leaf and spine switches in the Cisco ACI.

Nexus 9000 Series

<https://apps.kaonadn.net/4357027/index.html#C276>

<https://www.cisco.com/c/en/us/products/switches/nexus-9000-series-switches/index.html#~stickynav=1>

- The Nexus 9000 Series is the next generation of data center switching infrastructure.
- The Nexus 9000 Series Switches operate in one of two modes: Cisco ACI or Nexus Operating System (NX-OS).
- In ACI mode, the Nexus 9000 provide the spines and leaves that build the fabric
- The series offers a powerful combination of hardware and software that is custom-developed to provide a complete solution.
- Environments with NX-OS mode and Cisco ACI, while the fixed-port Cisco Nexus 9200 switches provide on. We can use modular Cisco Nexus 9500 switches or the Fixed-port Cisco Nexus 9300 switches for both traditional Cisco data center NX-OS functionality.
- The APIC is a central point of automated provisioning and management for all switches that are part of the ACI fabric

Once the APIC is set up, it can discover switches that are directly connected, and other leaf and spine switches in the ACI. The ACI fabric uses LLDP and DHCP-based fabric discovery to automatically discover the fabric switch nodes, assign the infrastructure VTEP addresses, and install the firmware on the switches

Before this automated process, a minimal bootstrap configuration must be performed on the APIC

Use at least 3 APIC for redundancy

Leaves can only be connected to spines. There should be no cabling between the leaves, **even when the leaves are being configured as vPC peer devices**. Spines can only be connected to leaves. Spines cannot be interconnected

A Cisco APIC must be attached to a leaf. Cisco APICs should be dual-homed (connected to two different leaves) for redundancy. All end points, Layer 2, Layer 3, Layer 4-Layer 7 devices must connect to leaves. Nothing should be connected to spines other than leaves

ACI Fabric Discovery

1. **APIC Bootstrap configuration** - APIC is configured with the cluster, fabric name, IP address
2. The leaf switch discovers the attached Cisco APIC through LLDP and requests the TEP address and boot file through DHCP.
3. The spine switch discovers the attached leaf switch through LLDP and requests TEP address and boot file through DHCP.
4. All nodes in the same Cisco APIC cluster should contain the same bootstrap information if the nodes are intended to form a cluster.
5. The fabric can be discovered and initialized from multiple sources concurrently.
6. The fabric will self-assemble starting from multiple Cisco APIC sources.
7. A Cisco APIC cluster will form when members discover each other through the appliance vector

APIC automatically learns about the other APIC controllers in Cisco ACI through switches.

During the cluster discovery, the APIC servers:

- Use internal private IP addresses to communicate with ACI switches and other APIC servers
- Discover the IP addresses of other APIC servers in the cluster through LLDP
- Proceed to discover leaves, through LLDP, which in turn discover spines
- Cisco APIC automatically learns about the other Cisco APIC controllers in Cisco ACI through switches.
- The APIC then discovers the leaves through LLDP and programs them with the new appliance vector
- Switches then start advertising this new AV to their neighbors

ACI Fabric Policies

Fabric access policies enable communication of systems that are attached to the ACI fabric.

In this demonstration , we will implement out-of-band (OOB) connectivity for the fabric switches and configure an access policy for the hypervisor

Fabric access policy with multiple configuration elements:

- **Pool:** Defines a range of identifiers, such as VLANs
- **Physical/external/VMM domain:** References a pool. You can think of it as a resource container
- **Attachable access entity profile (AAEP):** References a domain, and therefore specifies the resource pool that is activated on an interface
- **Interface policy:** Defines a protocol or interface properties that are applied to interfaces
- **Interface policy group:** Gathers multiple interface policies into one set and binds them to an AAEP
- **Interface selector:** Identifies one or more interfaces (interface blocks) and associates them with an interface policy group
- **Interface profile:** Groups one or more interface selectors, effectively specifying the policies consumed by the interface blocks
- **Switch profile:** Chooses one or more leaf switches and associates them with an interface profile, effectively specifying the policies consumed by the interface blocks on a given switch
- **Access policies** configure external-facing interfaces that connect to devices such as virtual machine controllers and hypervisors, hosts, network attached storage, routers, or fabric extender (FEX) interfaces
- Access policies enable the configuration of port channels and virtual port channels, protocols such as LLDP, Cisco Discovery Protocol (formerly known as CDP), or Link Aggregation Control Protocol (LACP), and features such as statistics gathering, monitoring, and diagnostics.
- **Switch profiles:** Specify which switches to configure and the switch configuration policy.
- **Module profiles:** Specify which leaf switch access cards and access modules to configure and the leaf switch configuration policy.
- **Interface profiles:** Specify which access interfaces to configure and the interface configuration policy.
- **Global policies:** Enable the configuration of DHCP, QoS, and Attachable Access Entity Profile (AAEP).
- **Pools:** Specify VLAN, VXLAN, and multicast address pools.
- **Physical and external domains:** Define external bridged domain, external routed domain, and physical domain policies.
- **Monitoring and troubleshooting policies:** Specify what to monitor, the thresholds, how to handle faults and logs, and how to perform diagnostics.

Virtual eXtensible LAN

- VXLAN is a Layer 2 overlay scheme over a Layer 3 network
- A 24-bit VXLAN network identifier, also known as VNID is included in the encapsulation to provide up to 16-MB VXLAN segments for traffic isolation or segmentation
- This ability by far exceeds the limit of 4K segments that are addressable with VLANs
- Each segment represents a unique Layer 2 broadcast domain. The segment can uniquely identify the address space or subnet of a tenant.

Examining Nexus OS Automation and Scripting Tools

Examine the tools that allows us to Automate the day-to-day management, monitoring, and configuration to increase efficiency and help eliminate errors

- NX-OS
- Cisco Embedded Event Manager
- Bash Shell

Tcl– Tool Command Language

This traditional method has its deficiencies. It is error prone, and it is not scalable.

Traditional network management process uses CLI to get and send commands to devices.

- Engineers prepare configuration in text editor, using copy and paste
- Some sort of automation is done via Tool Command Language (Tcl)/Expect scripts
- CLI can return unexpected output
- The process is error prone

The challenges with traditional network management process:

- Managing scalable infrastructure consumes too much time, energy, resources
- The network lags industry automation capabilities.

Therefore, it is very important to include automation in the network management process.

An essential part of an automation is programmability.

Programmability brings these benefits:

- Saves resources
- Enables fast and flexible service delivery
- Minimizes human error
- Allows customization and innovation
- An essential part of automation is programmability.

The Cisco Nexus software running on Cisco Nexus switches is:

- **Modular:** Has extensions that accommodate business needs.
- **Highly programmatic:** Allows for rapid automation and orchestration through Application Programming Interfaces (APIs).
- **Secure:** Protects and preserves data and operations.
- **Flexible:** Integrates and enables new technologies.
- **Scalable:** Accommodates and grows with the business and its requirements.
- **Easy to use:** Reduces the amount of learning required, simplifies deployment, and provides ease of manageability.

With the Nexus operating system (Cisco NX-OS), the device functions in the unified fabric mode to provide network connectivity with programmatic automation functions.

- Cisco NX-OS contains Open Source Software (OSS) and commercial technologies that provide automation, orchestration, programmability, monitoring, and compliance support.
- NX-OS software supports traditional management interfaces like CLI, Simple Network Management Protocol (SNMP), syslog, and others.
- NX-OS also supports Network Configuration Protocol (NETCONF), which is newer network management protocol
- All those protocols can be used to implement some degree of an automation.

The typical network management protocols and technologies are:

CLI

- It is designed as a human-readable interface.
- Returns unstructured data, which needs post-processing.

SNMP

- Widely used for monitoring of network devices.
- Cisco NX-OS supports SNMP v1, v2, v3.
- NX-OS programmable interface agents:

NETCONF

- The Nexus 9000 switch supports a variety of programmable and automation tools and methods
- The Nexus 9000 Series devices support Python v2.7.5 in both interactive and noninteractive (script) modes and are available in the Guest Shell.
- The Python scripting gives programmable access to the device's CLI to perform various tasks
- Python also can be accessed from the Bash shell

Job Scheduler and EEM

There are two main components of a scheduler:

Job: A routine task or tasks defined as a command list

Schedule: The timetable for completing a job

- **Periodic mode:** A recurring interval (daily, weekly, monthly, delta)
- **One-time mode:** A job is completed only once at a specified time

The scheduler has the following configuration guidelines and limitations:

- The scheduler can fail if it encounters one of the following while performing a job:
- The license has expired for a feature at the time the job for that feature is scheduled.
- A feature is disabled at the time when a job for that feature is scheduled.
- You have removed a module from a slot and a job for that slot is scheduled.

Verify that you have configured the time. The scheduler does not apply a default timetable. If you create a schedule and assign jobs and do not configure the time, the job is not started.

This example shows the scheduler job configuration:

You can use CLI variables in the Nexus CLI. To set a variable, use the command `cli var name`, and to reference the variable use the syntax `$(MY_VAR)`. There are some predefined variables, like `TIMESTAMP`.

1. The example first sets the variable `timestamp`
2. Then the running configuration is copied to the bootflash
3. The last step is to copy the configuration file to the TFTP server

```
switch# configure terminal
switch(config)# scheduler job name MY_CFG_BACKUP
switch(config-job)# cli var name timestamp $(TIMESTAMP) ;copy running-config
bootflash:/$(SWITCHNAME)-cfg.$(timestamp) ;copy bootflash:/$(SWITCHNAME)-
cfg.$(timestamp) tftp://10.1.1.1/ vrf management
switch(config-job)# end
!
switch(config)# scheduler schedule name MYDAILY
switch(config-schedule)# job name MY_CFG_BACKUP
switch(config-schedule)# time daily 4:00
switch(config-schedule)# end
!
switch# show scheduler schedule
Schedule Name : MYDAILY
-----
User Name : admin
Schedule Type : Run every day at 4 Hrs 00 Mins
Last Execution Time : Sat Oct 21 4:00:00 2019
Last Completion Time: Sat Oct 21 4:00:01 2019
Execution count : 2

show scheduler config: displays the scheduler configuration
show scheduler job: displays the jobs configured
show scheduler logfile: displays the contents of the scheduler log file
show scheduler schedule: displays the schedules configured

To verify the scheduler, use these commands:
show scheduler config: Displays the scheduler configuration.
show scheduler job: Displays the jobs configured.
show scheduler logfile: Displays the contents of the scheduler log file.
show scheduler schedule: Displays the schedules configured.
```

Bash Shell and Guest Shell for Nexus

Sometimes you would need traditional tools from the server space to manage network devices.

Nexus 9000 devices support direct **Bourne-Again Shell** (Bash) access to the underlying Linux system on the device, from where you can manage the system

In addition, the Nexus 9000 Series devices support a **guest shell** that provides Bash access into a Linux execution space on the host system that is decoupled from the host NX-OS software

Cisco NX-OS supports two Linux environments:

Bash shell

- Allows access to the underlying Linux system
- Disabled by default

Guest shell

- Secure Linux container environment
- Decoupled from the host Nexus 9000 NX-OS software
- Allows you to add software packages and update libraries as needed without impacting the host system software
- Enabled by default

```
switch# configure terminal
switch(config)# feature bash-shell
switch(config)# end
switch# run bash
bash-4.2$ whoami
admin
bash-4.2$ pwd
/bootflash/home/admin
username sdouglas shelltype bash
#!/bin/bash
i=0
while [ $i -lt 120 ]
do
    echo "`date`: `vsh -c \"show ip route\" | grep ubest | wc -l`" >> route_count
    sleep 30
    i=$((i+1))
done
```


Run BASH Script in NX-OS

```
switch# run bash /bin/bash /bootflash/home/admin/script.sh
switch# show file bootflash:home/admin/route_count
Fri Oct 20 10:27:09 UTC 2019: 16
Fri Oct 20 10:27:39 UTC 2019: 16
Fri Oct 20 10:28:10 UTC 2019: 16
bash-4.2$ yum list installed | grep n9000
base-files.n9000                3.0.14-r74.2                installed
bfd.lib32_n9000                 1.0.0-r0                    installed
core.lib32_n9000                1.0.0-r0                    installed
eigrp.lib32_n9000               1.0.0-r0                    installed
eth.lib32_n9000                 1.0.0-r0                    installed
isis.lib32_n9000                1.0.0-r0                    installed
```

Guest Shell

In addition to the NX-OS CLI and Bash access on the underlying Linux environment, the Nexus 9000 Series devices support access to Guest Shell, which is a decoupled execution space running within a Linux Container (LXC).

Guest Shell is accessible to the users with the **network-admin** role

Characteristics of Guest Shell:

- It is automatically enabled in the system
- The Guest Shell is populated with CentOS 7 Linux
- Use the run guestshell or guestshell commands to access the Guest Shell
- Use the run guestshell command command to execute the command in Guest Shell
- Use the dohost command command to run Cisco NX-OS command from Guest Shell
- **Guest Shell has various utilities and capabilities available by default**
- **Guest Shell provides the ability to use yum install software for installing the packages**

Guest Shell is pre-populated with many of the common Linux tools:

- net-tools
- iproute
- tcpdump
- OpenSSH

GuestShell Commands

run guestshell or guestshell commands to access the Guest Shell
run guestshell command command to execute the command in Guest Shell
dohost command command to run NX-OS command from Guest Shell
guestshell enable installs and activates the Guest Shell
guestshell disable shuts down and disables the Guest Shell
guestshell upgrade deactivates and upgrades the Guest Shell
guestshell reboot deactivates the Guest Shell and then reactivates it
guestshell destroy deactivates and uninstalls the Guest Shell
guestshell resize changes the allotted resources available for the Guest Shell
show guestshell detail displays details about the Guest Shell
switch# show guestshell detail
[admin@guestshell ~]\$ ifconfig Eth1-47
[admin@guestshell ~]\$ ls -al /var/run/netns
[admin@guestshell ~]\$ ifconfig | grep Eth1
[admin@guestshell ~]\$ chvrf management
[admin@guestshell ~]\$ ifconfig | grep Eth1
[admin@guestshell ~]\$ ifconfig

Command	Description
guestshell enable	Installs and activates the Guest Shell.
guestshell disable	Shuts down and disables the Guest Shell.
guestshell upgrade	Deactivates and upgrades the Guest Shell.
guestshell reboot	Deactivates the Guest Shell and then reactivates it.
guestshell destroy	Deactivates and uninstalls the Guest Shell.
guestshell resize	Changes the allotted resources available for the Guest Shell.
show guestshell detail	Displays details about the Guest Shell.

Guest Shell is enabled by default. To disable Guest Shell, use the guestshell disable command.

```
switch# guestshell disable
switch# guestshell destroy
switch# guestshell enable
switch# show guestshell detail
```

```
[admin@guestshell ~]$ ifconfig Eth1-47
[admin@guestshell ~]$ ls -al /var/run/netns
[admin@guestshell ~]$ ifconfig | grep Eth1
[admin@guestshell ~]$ chvrf management
[admin@guestshell ~]$ ifconfig | grep Eth1
[admin@guestshell ~]$ ifconfig
```

There are various options with the yum command:

yum list installed: Displays a list of the NX-OS feature RPMs installed on the switch.

yum list available: Displays a list of the available RPMs.

sudo yum -y install rpm: Installs an available Red Hat Package Manager (RPM) package.

sudo yum -y upgrade rpm: Upgrades an installed RPM.

sudo yum -y downgrade rpm: Downgrades the RPM if any of the yum repositories have a lower version of the RPM.

sudo yum -y erase rpm: Erases the RPM.

yum list --patch-only: Displays a list of the patch RPMs present on the switch.

sudo yum install --add URL_of_patch: Adds the patch to the repository.

sudo yum install patch_RPM --nocommit: Activates the patch RPM, where patch_RPM is a patch that is located in the repository.

sudo yum install patch_RPM --commit: Commits the patch RPM. The patch RPM must be committed to keep it active after reloads.

sudo yum erase patch_RPM --nocommit: Deactivates the patch RPM.

sudo yum install --remove patch_RPM: Removes an inactive patch RPM.