

Cisco Nexus 9000 Series Switches: Integrate Programmability into Your Data Center

What You Will Learn

In this document, you will learn how the Cisco Nexus® 9000 Series Switches align with business and IT needs to integrate programmability and automation into the entire network infrastructure. Automation has always been an important part of administration, but it is even more important now as networks become more robust. To support this trend, Cisco has added more features to the Cisco Nexus 9000 Series Switches while at the same time reducing complexity, cost, and training overhead.

For some organizations, Cisco® Application Centric Infrastructure (ACI) will be the solution of choice, but other organizations will want to take advantage of the programmability offered by Cisco Nexus 9000 Series Switches in Cisco NX-OS standalone mode, which is the focus of this document.

Introduction

Many organizations are currently trying to figure out how to make their IT infrastructure more cost effective and are moving toward an on-demand model by taking advantage of automation and virtualization. They are also striving for better analytics to manage their business more efficiently. The programmability of the Cisco Nexus 9000 Series Switches allows IT departments to quickly meet high-priority business needs at a cost-effective price. This capability is especially important for organizations with large Linux or Unix environments. Although next-generation data centers are enabling business opportunities, many are doing so under decreasing budgets. As the Cisco Nexus 9000 Series gains more open source capabilities and becomes more Linux like, IT departments that already have Linux or Unix skill sets in house will be able to take advantage of this programmability with little training.

The Cisco Nexus 9000 Series also offers the fastest 40- and 100-Gbps platform in its class, along with 15 percent more efficient power and cooling.

By offering an open operating system, the Cisco Nexus 9000 Series can meet your IT and business needs while still helping you reduce capital expenditures (CapEx) and operating expenses (OpEx).

Use Cases for Programmability

Customers continue to want programmability capabilities in the data center for several reasons. One reason is to automate provisioning and configuration, both to save time when provisioning new network devices and to prevent the human errors that often are a byproduct of manual configuration.

Integration with orchestration tools is also another important reason to embrace programmability. The network is not the only part of the infrastructure that administrators need to manage. If they can use a single orchestration tool to manage computing, storage, and networking resources, they will need to spend less time learning new tools.

In addition, the collection of information for analytics becomes more important as tasks are automated.

Organizations can use analytics and health monitoring tools to optimize their environments and to troubleshoot and provide alerts about any problems within the network.

Boot and Provisioning	POAP , PXE *
Package and Application Management	RPM
Open Interfaces	Bash , Broadcom Shell , Python , Guest Shell
Extensibility	Containers
Orchestration Tools	Puppet , Chef , CFEngine , Ansible *

* Some features may not be supported currently. Please check with your Cisco representative for exact feature support.

Booting and Provisioning

You can automate the bootup and provisioning of Cisco Nexus 9000 Series Switches in several ways.

Power-On Auto Provisioning

The Cisco Nexus 9000 Series offers Power-On Auto Provisioning (POAP). POAP enables a switch to upgrade its image on startup without the need for the administrator to touch it, eliminating any errors that might result from manual configuration. POAP works when no startup-config file is found on the switch. At that point, the system will search for a Domain Host Configuration Protocol (DHCP) server, which will assign it an IP address, a default gateway, and an IP address for the Domain Name System (DNS) server. It will also get an IP address for a script server and download the correct script, which will run locally on the switch. The script will usually instruct the system to download and install the correct image and running-config file.

You can also deploy Puppet agent POAP. During the time the router downloads the images and running-config file, it will also run the Puppet agent executable file. When the switch comes up, a Puppet agent will be running in a Linux container. The Puppet agent will then establish a connection with the Puppet Master server.

For configuration details please check the configuration guide:

http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/sw/7-x/fundamentals/configuration/guide/b_Cisco_Nexus_9000_Series_NX-OS_Fundamentals_Configuration_Guide_7x/b_Cisco_Nexus_9000_Series_NX-OS_Fundamentals_Configuration_Guide_7x_chapter_0100.html.

There is also a script available on GitHub:

http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/sw/7-x/fundamentals/configuration/guide/b_Cisco_Nexus_9000_Series_NX-OS_Fundamentals_Configuration_Guide_7x/b_Cisco_Nexus_9000_Series_NX-OS_Fundamentals_Configuration_Guide_7x_chapter_0100.html.

Preboot Execution Environment

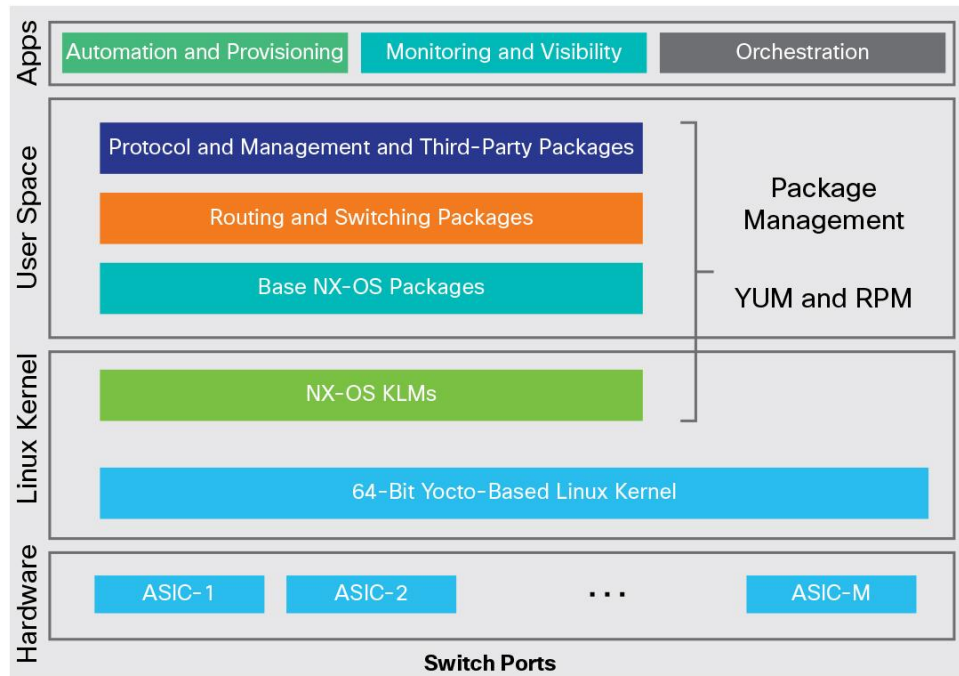
Another advanced boot loader available on the Cisco Nexus 9000 Series is the Preboot Execution Environment (PXE) boot tool. PXE is a common tool with a well-understood framework and should be familiar to the server team. It also fetches an image from the network and allows automated installation and bootup. PXE takes advantage of current skill sets and consolidates solutions because you can use it with both servers and switches.

Package and Application Management

Linux Kernel with Yocto

The switches are based on a 64-bit Yocto (Version 1.2) based Wind River (Version 5) Linux kernel, as shown in Figure 1.

Figure 1. Provisioning, Monitoring, and Orchestration All Based on the Yocto Linux Environment



From a management point of view, this architecture reduces operating costs because the network and server teams already have Linux skills. They'll be able to manage the switches as they would their Linux servers. By providing a more open Cisco NX-OS network, this design also serves businesses and enterprises as well as cloud providers working toward designing and managing massively scalable data centers (MSDCs). Customers also can use the Yocto tool-chain environment to build kernel load modules, and users can take advantage of RPMs and kernel load modules to put their own packages in the kernel or user space of the switch.

RPM Package Manager Using YUM

Many of Cisco's patches and other software packages will now be offered as Red-Hat Package Managers (RPMs). Much as you would update or load software on a Linux server, you can now do the same on the Cisco Nexus 9000 Series Switches. You can also install third-party RPMs and daemons such as `tcpdump`, `tcollector`, and `iperf` in a Linux container or in the Cisco NX-OS kernel itself. These daemons will be managed by the popular Yellowdog Updater, Modified (YUM) package updater, so you will be able to upgrade through a **yum update** command. For example, you could update Border Gateway Protocol (BGP) by using the command **yum update BGP**.

Open Interfaces

Bash Access

Access to the Bash Shell gives customers the capability to run Bash scripts and opens the switch to third-party Layer 1, 2, and 3 tools. The user can issue a command-line interface (CLI) command to easily gain access to the Bash Shell. Within Bash, the user can then use common commands such as **ps** and **grep** for further monitoring and scripting. The Bash Shell also has nonroot privileges to protect against operator errors and offers role-based access control (RBAC). The Bash access exposes the network interfaces as Linux devices (such as eth1/1) so that you can use commands like **ifconfig** and **tcpdump** to manage the switch interfaces in the same way that you would manage network ports on a Linux server. You can also take advantage of the Linux networking stack by using third-party orchestration applications to complete tasks such as injecting routes and even installing third-party routing protocols on switches. Also, with Bash exposed, you can manage Virtual Routing and Forwarding (VRF) instances as Linux namespaces. Forwarding tables for VRF will essentially match Linux namespaces.

Broadcom Shell Access

Cisco also allows access to the Broadcom (BCM) shell for users who want to perform diagnostics on Broadcom chips or change forwarding behavior by using a Python wrapper around the BCM shell to get output. This access also gives users read and write access to hardware tables and access to underlying registers.

Python Scripting

Python is an easy-to-learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an excellent language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standards library are freely available in source or binary form for all major platforms from the Python website: <http://www.python.org/>.

The same site also contains distributions of and pointers to many free third-party Python modules, programs, and tools and additional documentation.

The Cisco Nexus 9000 Series supports Python Release 2.7.5 in both interactive and noninteractive (script) modes.

The Python scripting capability on the Cisco Nexus 9000 Series Switches gives programmatic access to the switch CLI to perform various tasks and POAP and Cisco Embedded Event Manager (EEM) actions. Responses to Python calls that invoke a Cisco NX-OS CLI return JavaScript Object Notation (JSON) output instead of just text output: a powerful feature that makes Python scripting easy and helps ensure that the scripts are forward compatible.

The Python interpreter is available by default in NX-OS.

Various repetitive, manual workflows that are error prone can be automated either on-device or off-device using the Python scripting capabilities on the Cisco Nexus 9000 Series.

Secure Guest Shell

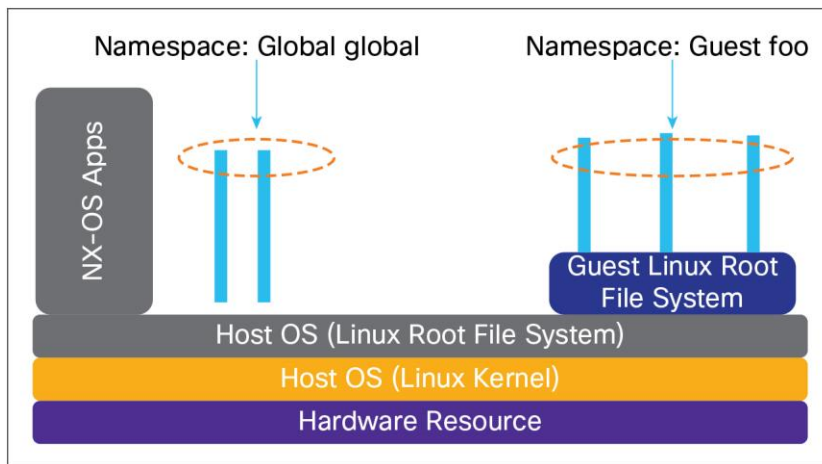
The Cisco Nexus 9000 Series Switches also allow Secure Guest Shell access. Essentially, this is a combination of Bash and a secure Linux environment with a customizable root system. This access allows administrators to run Linux commands and load RPMs natively as they commonly would on a Linux server, but in a controlled open environment, preventing host corruption. Libraries can be updated in the Secure Guest Shell independently of the host as well, making the system more modular. This is contained within a Linux Container.

Extensibility

Linux Containers

Customers can install their own applications in a Linux container (Figure 2), which offers a more secure option than adding daemons directly to the Bash Shell. A Linux container (LXC) is an operating system virtualization technology that shares the host kernel with the guest but provides isolation through namespace extensions to the Linux kernel (see <http://linuxcontainers.org> for more information).

Figure 2. Linux Containers



You can run either Cisco or third-party applications such as Puppet or Chef in the LXC. Because there isn't full access to the shell, capabilities are limited, but the segregation from the kernel is a safer option. Customers can use most standard Linux distributions in the LXC. Users can also use secure Linux technology such as [SMACK](#) to restrict third-party applications from accessing root privileges.

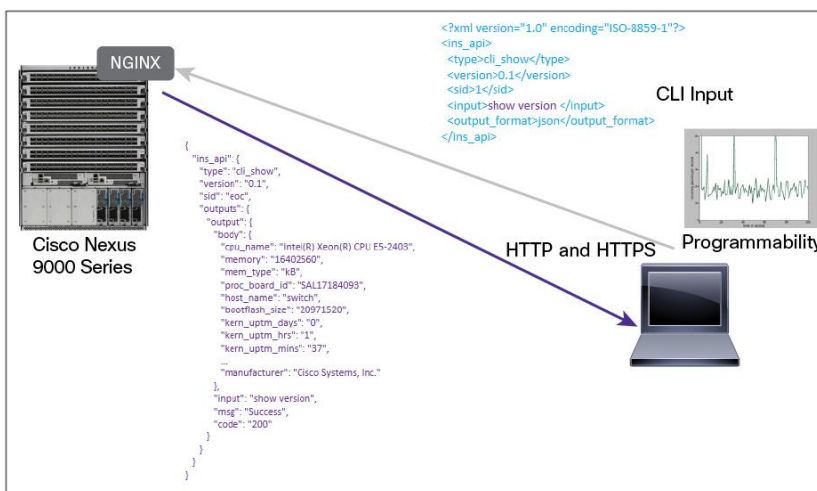
Orchestration Tools

[Puppet](#) and [Chef](#) are third-party orchestration tools that can be used with the Cisco Nexus 9000 Series. Puppet has been available but only within a container, which may limit capabilities. Newer versions of the Nexus 9000 Series switches allow Puppet, Chef, and CFEngine to have more capabilities because they will be available natively, to run as RPMs, on the kernel. These orchestration tools are all agent based, meaning that there is a master Puppet, Chef, and CFEngine server, and the agent is installed on the Cisco Nexus 9000 Series Switches. These languages are mostly declarative in nature, making compliance and configuration policies very effective.

Cisco NX-OS APIs

The Cisco NX-OS API (NX-API) on the Cisco Nexus 9000 Series Switches allows web-based programmatic access to the Cisco Nexus 9000 Series. This support is delivered through an open source web server, NGINX. NX-API exposes the complete configuration and management capabilities of the CLI through web-based APIs. The Cisco Nexus 9000 Series Switches can be instructed to publish the output of the API calls in either XML or JSON format. This comprehensive, easy-to-use API enables rapid deployment on the Cisco Nexus 9000 Series Switches (Figure 3). You can access several scripts on the GitHub site that have already been configured for you. Go to <http://github.com/datacenter/nexus9000> to find these scripts, and more are always being added.

Figure 3. Programmatic Access to Cisco Nexus 9000 Series through Cisco NX-API



Third-party orchestration tools such as [Ansible](#) are also very effective in using NX-API. Ansible is an orchestration tool, like Chef and Puppet, but is agentless. It uses a push model and is Python based, making it a nice fit for use with the Cisco Nexus 9000 Series.

Conclusion

Cisco is a leader in automation and orchestration with the Cisco Nexus 9000 Series Switches. By embracing the open culture of development and operations (DevOps) and creating a more Linux-like environment in the Cisco Nexus 9000 Series, Cisco enables IT departments with strong Linux skill sets to meet business needs efficiently. Customers can automatically provision and configure Cisco Nexus 9000 Series Switches by using POAP and PXE. With the support for Bash Shell, Secure Guest Shell, and LXC access, administrators are free to load both Cisco and third-party RPMs and daemons as they would on a Linux server. Finally, by using orchestration tools like Puppet, Chef, CFEngine, and Ansible within the shell or LXC or through the Cisco NX-API, customers can use the same orchestration tools that are used on servers to configure switches in a touchless manner, thereby reducing human error and speeding up processes.

For More Information

For more information about Cisco Nexus 9000 Series Switches, please go to <http://cisco.com/go/nexus9000>.




Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

 Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)