White Paper

# Gain Network Programmability and Automation with Open Cisco NX-OS

Organizations today need to deploy applications much faster, making use of agile software, and improve existing processes to support software upgrades. They need to manage and operate networks more productively using common tools that the network, server, and software teams can use in collaboration. In environments in which constant change is the norm, organizations need to deploy high-quality user-focused applications that deliver immediate business value. Accomplishing these goals requires a new approach to the data center. This approach needs to fill the gap between the development and operation teams, enabling developers to push code to the infrastructure quickly without negatively affecting the overall behavior of the network.

To understand the ways in which operations need to change, consider the ways in which networks are set up and provisioned today. Most organizations spend most of their time trying to make sure that the network works by staging and testing it. Furthermore, they use cumbersome, disjointed, and error-prone manual tasks to provision and change the network to accommodate application needs. They also use manual processes to identify the potential sources of network problems and to perform repetitive tasks, consuming large amounts of network engineers' time.

To improve data center operations and better meet business needs, organizations need process automation and holistic architecture provisioning. Achieving these goals requires cultural changes by IT teams. Organizations need better and more open libraries and interfaces, with common DevOps tools to automate scripting and provide higher-level programmatic control. They need agents and distributed processes to collect and process information about the state of the network and its components.

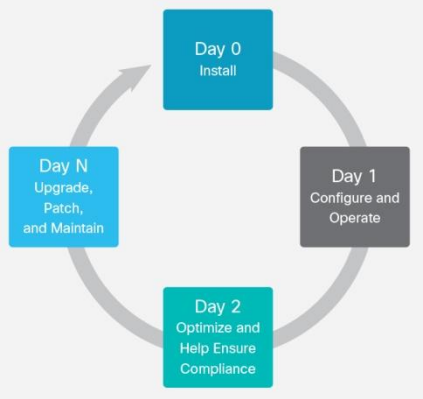
This document explores how a highly programmable network can enable automation of the full network lifecycle, supporting agile development and achieving operational efficiency. It examines several use cases that use the power of the programmable and open Cisco® NX-OS Software operating system:

- Extended interfaces (Cisco NX-API command-line interface (CLI), NX-API representational state transfer [REST], and Broadcom shell access)
- Native Linux-based management
- Open-source tools (Ignite, Preboot Execution Environment [PXE] and Power-on Autoprovisioning (POAP))
- Linux containers (LXCs) in either a guest or native shell and Red-Hat Package Manager (RPM)
- Scripting languages such as Python, Ruby, and data representation (JavaScript Object Notation [JSON] and XML)
- Configuration management and orchestration tools (Puppet, Chef, Ansible, OpenStack plug-in, and Cisco UCS® Director)

## Use Cases

This document explores use cases that reflect day-zero, day-one, and day-two operations (Figure 1) as well as operations that network IT teams perform every day, such as network troubleshooting, configuration backup, scaling, and orchestration.

**Figure 1.** Network Lifecycle Operation




### Day-Zero Operation: Installing New Switches

**Challenge:** The focus of day zero is on bringing up and discovering new devices as quickly as possible with features and functions that don't change much over the lifecycle of the network: the device name, administrator user name and password, management IP address, console access, out-of-band management and interface, etc. In a typical environment, the startup process is manual and may take hours or even days. The challenge: How to reduce this process to minutes?

**Solution:** Using the open-source Ignite tool to facilitate initial network bootstrapping using POAP (upgrading software images and installing configuration files on Cisco Nexus® switches) and PXE, Cisco Nexus switches can be automatically discovered and installed in minutes, eliminating human error. Ignite also gives administrators the capability to define configuration templates, fabric topology, and resource pools.

In addition, POAP and PXE can install agents for configuration management tools such as Puppet and Chef during switch startup.

### Day-One Operation: Configuring and Operating Switches

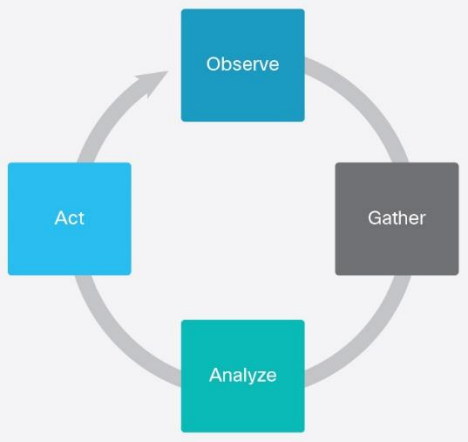
**Challenge:** Organizations must be able to rapidly implement changes in the data center environment, quickly adding and removing virtual machines and creating and setting up VLANs, quality-of-service (QoS) policy, Virtual Routing and Forwarding (VRF) instances, virtual port channels (vPCs), etc. They must also establish connectivity between the virtual machines and devices. The ability to make changes quickly and to move away from manual device-by-device configuration is crucial to help businesses avoid errors, security breaches, and downtime.

Other possible tasks include configuring Border Gateway Protocol (BGP), verifying that a certain version of NX-OS BGP software is running on the switches in a simplified way, and responding quickly by setting up access control lists (ACLs) to contain certain attacks on the network. The challenge is for IT to do this all quickly across the entire network.

**Solution:** Cisco Nexus switches support configuration management tools such as agent-based Puppet and Chef and agentless Ansible. These tools enable organizations to automate the tasks shown in Figure 2 by building consistent and repeatable processes to implement and verify changes, as well as to remediate exceptions and violations that occur in the infrastructure. By creating a centralized repository of configurations that the switches can periodically employ, these tools allow IT to efficiently and consistently manage the infrastructure with little effort. For more information, see GitHub.

IT administrators can also use the NX-API REST capability in NX-OS (object model) to configure and check the status of an object. Objects can be physical ports on the switch or specific features such as BGP, VRF instances, and VLANs. The use of object models enables a hierarchical and standardized representation of configurations that eliminates the need to pass CLI commands and scripts.

**Figure 2.** Configuring and Operating Switches

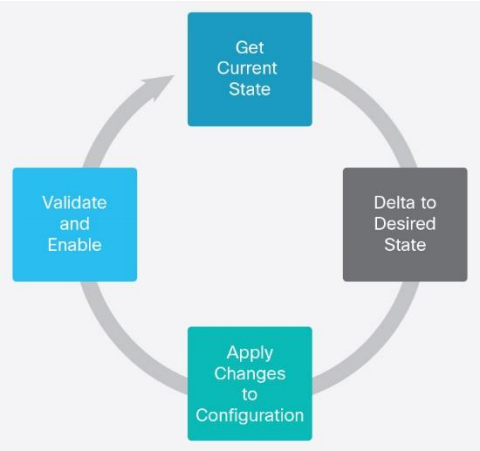



### Day-Two Operation: Optimizing the Network and Helping Ensure Compliance

**Challenge:** Ongoing requests to make changes in the network typically take time to plan and implement. These requests include updating switch images, patching software, and creating security policy to support application requirements and typically are performed manually. Manual processes tend to cause delays in deployment and the potential for errors that can lead to security breaches and an inefficient operation model. Furthermore, optimizing switch behavior to support application needs can be cumbersome, and continuously adding feature upgrades in safely and simply without breaking the network can be complicated.

**Solution:** As Figure 3 shows, network administrators can define in a central repository what needs to be changed and updated, as well as which switches and ports will be affected. Using a configuration management tool such as Puppet enables rapid integration of applications into the customer's operational tool chain and triggers configuration from a central repository. With this process, a large deployment of switches can be updated in minutes. This updating is performed in the same way as for computing nodes.

**Figure 3.** Optimizing the Network and Helping Ensure Compliance



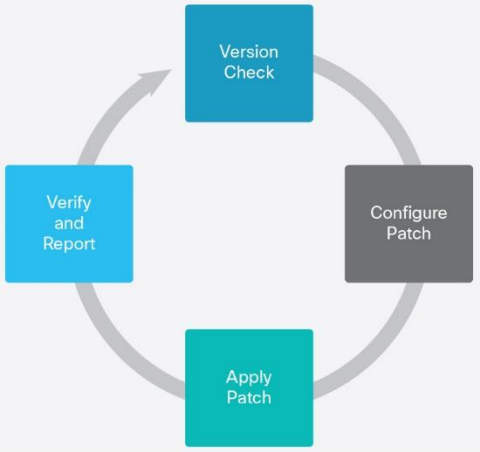

### Day-N Operation: Software Upgrading, Patching, and Maintenance

**Challenge:** Most organizations today can't afford to take devices offline for a period of time to update software. To avoid disruption, they implement updates during nonpeak hours, leading to a less productive environment.

**Solution:** At the core of this use case is the extensibility of open NX-OS, which uses standard package management tools such as RPM and Yum for software management. The same tools can be used for open NX-OS process patching and for installing external or custom-developed programs on a switch. Organizations can install native RPM and third-party applications running processes as they would on a Linux server. RPM-based packages enable organizations to load only the services and packages required, and they can perform patching with RPM instead of implementing a monolithic upgrade (Figure 4).

The capability to load and unload modules in the kernel as needed; isolate faults in features, services, and user application; and perform graceful restart and removal of processes reduces maintenance windows and allows the organization to update specific modules without bringing down the entire switch.

**Figure 4.** Software Upgrading, Patching, and Maintenance

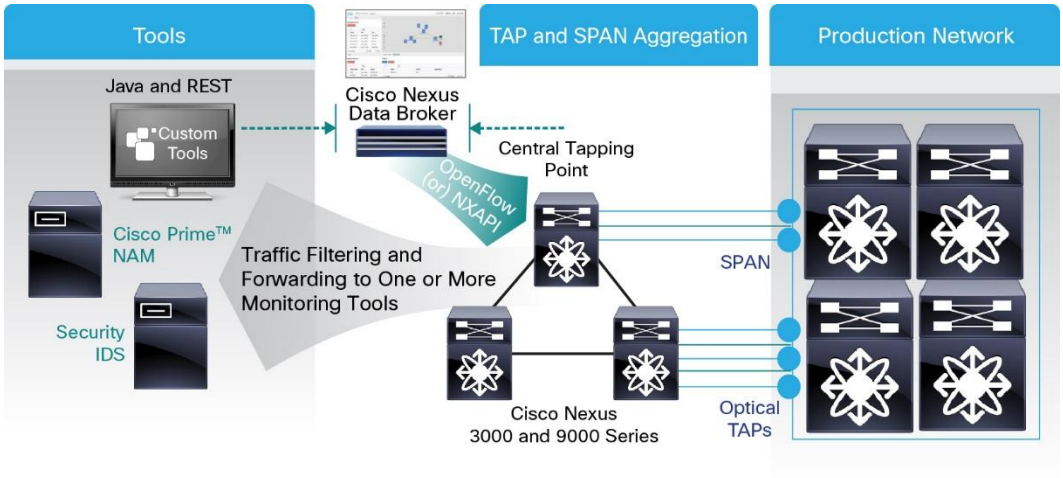


## Visibility, Monitoring, and Troubleshooting

**Challenge:** Troubleshooting and locating the source of a problem can be a difficult and long process for many organizations. Device-by-device monitoring, maintenance, and analysis can be cumbersome, especially in large networks, with organizations typically relying on network test access points (TAPs) to collect data. Production traffic collected through a TAP and through Cisco Switched Port Analyzer (SPAN) and Remote SPAN (RSPAN) is valuable to IT, and access to more data in a manageable way can help determine what is happening inside the switch.

**Solution:** Cisco Nexus Data Broker, supported by Cisco Nexus 3000 and 9000 Series Switches, replaces traditional purpose-built matrix switches to let you build a scalable TAP and SPAN aggregation infrastructure that you can interconnect at 1, 10, 40, and 100 Gbps. You can dedicate ports for TAP and for SPAN and for traditional Ethernet connectivity. IT can access the data broker application through the web-based GUI or REST API. The data broker offers a simple, scalable, and cost-effective solution for enterprise customers who need to monitor high-volume and business-critical traffic (Figure 5).

**Figure 5.** Visibility, Monitoring, and Troubleshooting




In addition, full access to the ioctl and netdevice interface libraries using Linux Bash allows customers to install tools such as tcpdump. These tools provide additional device visibility and performance information by tapping into ports and VLANs and sending the output to a collector port. With this approach, organizations do not need to place physical TAPs on every device.

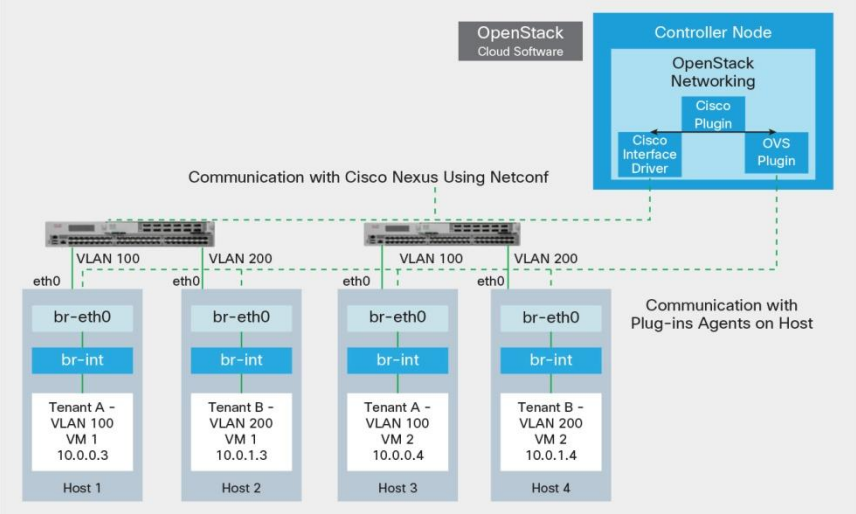
Using RPM capabilities, a third-party monitoring application such as Splunk forwarder, Tcollector, or ganglia can be installed in a secure container and provide further visualization and analysis.

## Scalability, Automation, and Orchestration

**Challenge:** Customers are striving to automate and orchestrate data center resources from open resource tools to optimize operations and lower costs. In today's environment, these capabilities are limited by lack of integration with network devices. In addition, customers are challenged to build an efficient architecture that can help them scale easily and quickly to match automation and orchestration needs.

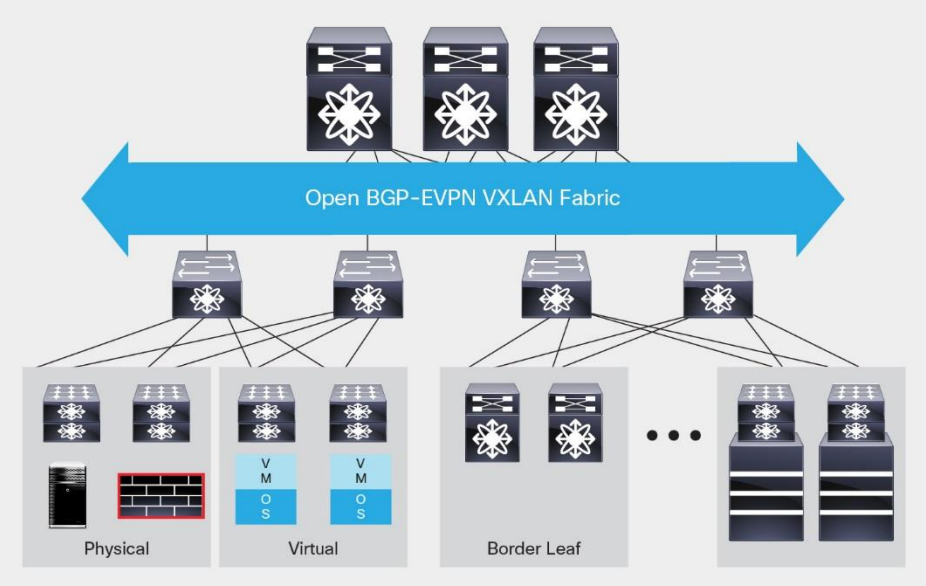
**Solution:** Using the OpenStack ML2 network plug-in for Cisco Nexus rather than the CLI, customers can create and orchestrate network resources from a single location. Customers also have the option to use Cisco UCS Director for orchestration. The capability to use OpenStack or Cisco UCS Director allows customers to orchestrate network, computing, and storage resources from a single place (Figure 6).

**Figure 6.** Scalability, Automation, and Orchestration




Next-generation Virtual Extensible LAN (VXLAN) fabric using standards-based BGP and Ethernet VPN (EVPN) overcomes the scaling and workload mobility limitations of flood-and-learn processes. Customers can build programmable software-defined networking (SDN) overlay networks to deliver multitenancy and transparent host mobility at cloud scale. In addition, with Puppet and Chef agent and Ansible agentless integration into Cisco Nexus switches, you can implement VXLAN provisioning and automation with little effort (Figure 7).

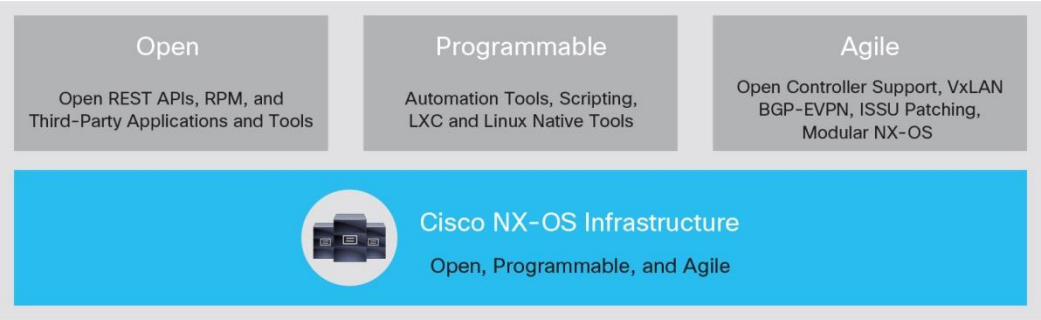
**Figure 7.** Open VXLAN Fabric



## Conclusion

With the open-system approach of Cisco Nexus switches, IT can provision the network more quickly, reduce time to recovery after a failure, and gain flexibility in an environment that is familiar to server administrators (Figure 8).

**Figure 8.** Open Cisco NX-OS Infrastructure




## For More Information

- Cisco DevNet community
- Detailed information about NX-OS and the use cases
- Open-source repository: GitHub

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.



Printed in USA C11-736824-00 03/16