Overview

Programmability Overview

The Cisco NX-OS software running on the Cisco Nexus 3000 Series switches is as follows:

- **Resilient**

  Provides critical business-class availability.

- **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 Cisco NX-OS operating system, 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.

For more information on Open NX-OS, see https://developer.cisco.com/site/nx-os/.

Standard Network Manageability Features

- SNMP (V1, V2, V3)

- Syslog

- RMON

- NETCONF

- CLI and CLI scripting

Advanced Automation Feature

The enhanced Cisco NX-OS on the device supports automation. The platform includes support for PowerOn Auto Provisioning (POAP).

PowerOn Auto Provisioning Support

PowerOn Auto Provisioning (POAP) automates the process of installing/upgrading software images and installing configuration files on Cisco Nexus switches that are being deployed in the network for the first time. It reduces the manual tasks required to scale the network capacity.

When a Cisco Nexus switch with the POAP feature boots and does not find the startup configuration, the switch enters POAP mode. It locates a DHCP server and bootstraps itself with its interface IP address, gateway, and DNS server IP addresses. The switch obtains the IP address of a TFTP server or the URL of an HTTP server and downloads a configuration script that enables the switch to download and install the appropriate software image and configuration file.

OpenStack Integration

The Cisco Nexus 3000 Series switches support the Cisco Nexus plugin for OpenStack Networking, also known as Neutron (http://www.cisco.com/web/solutions/openstack/index.html). The plugin allows you to build an infrastructure as a service (IaaS) network and to deploy a cloud network. With OpenStack, you can build an on-demand, self-service, multitenant computing infrastructure. However, implementing OpenStack's VLAN networking model across virtual and physical infrastructures can be difficult.

The OpenStack Networking extensible architecture supports plugins to configure networks directly. However, when you choose a network plugin, only that plugin's target technology is configured. When you are running OpenStack clusters across multiple hosts with VLANs, a typical plugin configures either the virtual network infrastructure or the physical network, but not both.

The Cisco Nexus plugin solves this difficult problem by including support for configuring both the physical and virtual networking infrastructure.

The Cisco Nexus plugin accepts OpenStack Networking API calls and uses the Network Configuration Protocol (NETCONF) by default or the REST API to configure Cisco Nexus switches as well as Open vSwitch (OVS) that runs on the hypervisor. The Cisco Nexus plugin configures VLANs on both the physical and virtual

network. It also allocates scarce VLAN IDs by deprovisioning them when they are no longer needed and reassigning them to new tenants whenever possible. VLANs are configured so that virtual machines that run on different virtualization (compute) hosts that belong to the same tenant network transparently communicate through the physical network. In addition, connectivity from the compute hosts to the physical network is trunked to allow traffic only from the VLANs that are configured on the host by the virtual switch.

Note We recommend configuring the REST API driver. For more configuration details, go to:

http://docwiki.cisco.com/wiki/Neutron_ML2_Driver_For_Cisco_Nexus_Devices_Ocata_Release

The following table lists the features of the Cisco Nexus plugin for OpenStack Networking:

Table 1: Summary of Cisco Nexus Plugin features for OpenStack Networking (Neutron)

Considerations	Description	Cisco Nexus Plugin
Extension of tenant VLANs across virtualization hosts	VLANs must be configured on both physical and virtual networks. OpenStack Networking supports only a single plugin at a time. You must choose which parts of the networks to manually configure.	Accepts networking API calls and configures both physical and virtual switches.
Efficient use of scarce VLAN IDs	Static provisioning of VLAN IDs on every switch rapidly consumes all available VLAN IDs, which limits scalability and makes the network vulnerable to broadcast storms.	Efficiently uses limited VLAN IDs by provisioning and deprovisioning VLANs across switches as tenant networks are created and destroyed.
Easy configuration of tenant VLANs in a top-of-rack (ToR) switch	You must statically provision all available VLANs on all physical switches. This process is manual and error prone.	Dynamically provisions tenant-network-specific VLANs on switch ports connected to virtualization hosts through the Nexus plugin driver.
Intelligent assignment of VLAN IDs	Switch ports connected to virtualization hosts are configured to handle all VLANs. Hardware limits are reached quickly.	Configures switch ports connected to virtualization hosts only for the VLANs that correspond to the networks configured on the host. This feature enables accurate port and VLAN associations.
Aggregation switch VLAN configuration for large multirack deployments.	When compute hosts run in several racks, you must fully mesh top-of-rack switches or manually trunk aggregation switches.	Supports Cisco Nexus 2000 Series Fabric Extenders to enable large, multirack deployments and eliminates the need for an aggregation switch VLAN configuration.

Programmability Support

Cisco NX-OS on Cisco Nexus 3000 Series switches support the following capabilities to aid programmability:

NX-API Support

Cisco NX-API allows for HTTP-based programmatic access to the Cisco Nexus 3000 Series switches. This support is delivered by NX-API, an open source webserver. NX-API provides the configuration and management capabilities of the Cisco NX-OS CLI with web-based APIs. The device can be set to publish the output of the API calls in XML or JSON format. This API enables rapid development on the Cisco Nexus 3000 Series switches.

Python Scripting

Cisco Nexus 3000 Series switches support Python v2.7.5 in both interactive and non-interactive (script) modes.

The Python scripting capability on the devices provide programmatic access to the switch CLI to perform various tasks, and to Power-On Auto Provisioning (POAP) and Embedded Event Manager (EEM) actions. Responses to Python calls that invoke the Cisco NX-OS CLI return text or JSON output.

The Python interpreter is included in the Cisco NX-OS software.

Tcl Scripting

Cisco Nexus 3000 Series switches support tcl (Tool Command Language). Tcl is a scripting language that enables greater flexibility with CLI commands on the switch. You can use tcl to extract certain values in the output of a **show** command, perform switch configurations, run Cisco NX-OS commands in a loop, or define EEM policies in a script.

Broadcom Shell

The Cisco Nexus 3000 Series switches front panel and fabric module line cards contain Broadcom Network Forwarding Engine (NFE). You can access the Broadcom command line shell (bcm-shell) from these NFEs.

Bash

Cisco Nexus 3000 Series switches support direct Bourne-Again SHell (Bash) access. With Bash, you can access the underlying Linux system on the device and manage the system.

Guest Shell

The Cisco Nexus 3000 Series switches support a guest shell that provides Bash access into a Linux execution space on the host system that is decoupled from the host Cisco NX-OS software. With the guest shell you can add software packages and update libraries as needed without impacting the host system software.

Container Tracker Support

Cisco NX-OS is configured to communicate with the Kubernetes API Server to understand the capabilities of the containers behind a given switch port.

The following commands communicate with the Kubernetes API Server:

- The **show containers kubernetes** command obtains data from *kube-apiserver* using API calls over HTTP.

- The **kubernetes watch** *resource* command uses a daemon to subscribe to requested resources and process streaming data from *kube-apiserver*.

- The **action** assigned in the **watch** command is performed on pre-defined triggers. (For example, Add/Delete of a Pod.)