



CISCO Nexus 9000 Series NX-OS Fundamentals Configuration Guide Release 10.4 User Guide

[Home](#) » [Cisco](#) » CISCO Nexus 9000 Series NX-OS Fundamentals Configuration Guide Release 10.4 User Guide



Contents

- [1 Nexus 9000 Series NX-OS Fundamentals Configuration Guide Release 10.4](#)
- [2 Preface](#)
- [3 New and Changed Information](#)
- [4 Overview](#)
- [5 Using the Cisco NX-OS Setup Utility](#)
- [6 Using PowerOn Auto Provisioning](#)
- [7 Using Network Plug and Play](#)
- [8 Understanding the Command-Line Interface](#)
- [9 Documents / Resources](#)
 - [9.1 References](#)



Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide, Release 10.4(x)
First Published: 2023-08-18

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706

USA

<http://www.cisco.com>

Tel: 408 526-4000

800 553-NETS (6387)

Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE.

EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at:

<http://www.cisco.com/go/softwareterms>. Cisco product warranty information is available at <http://www.cisco.com/go/warranty>. US Federal Communications Commission Notices are found here <http://www.cisco.com/c/en/us/products/us-fcc-notice.html>.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and-if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

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. (1721R)

© 2023 Cisco Systems, Inc. All rights reserved.

Preface

This preface includes the following sections:

- Audience, on page xi
- Document Conventions, on page xi
- Related Documentation for Cisco Nexus 9000 Series Switches, on page xii
- Documentation Feedback, on page xii
- Communications, Services, and Additional Information, on page xii

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
<i>Italic</i>	Italic text indicates arguments for which you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x y}	Braces enclosing keywords or arguments that are separated by a vertical bar indicate a required choice.
[x {y z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
variable	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.

Examples use the following conventions:

Convention	Description
screen font	Terminal sessions and information the switch displays are in screen font.
boldface screen font	Information that you must enter is in boldface screen font.
<i>italic screen font</i>	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

New and Changed Information

- New and Changed Information, on page 1

New and Changed Information

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
Secure POAP – CA group bundle	Added support for POAP on Cisco Nexus 9804 platform switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.	10.4(1)F	Secure Download of POAP Script, on page 24 Guidelines and Limitations for POAP, on page 42

Overview

This chapter contains the following sections:

- Licensing Requirements, on page 3
- Supported Platforms, on page 3
- Software Image, on page 3
- Software Compatibility, on page 4
- Serviceability, on page 4
- Manageability, on page 5
- Programmability, on page 6
- Traffic Routing, Forwarding, and Management, on page 7
- Quality of Service, on page 9
- Network Security Features, on page 9
- Supported Standards, on page 10

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the Cisco NX-OS Licensing Guide and the Cisco NX-OS Licensing Options Guide.

Supported Platforms

Starting with Cisco NX-OS release 7.0(3)I7(1), use the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.

Software Image

The Cisco NX-OS software consists of one NXOS software image. This image runs on all Cisco Nexus 3400 Series switches.

Software Compatibility

The Cisco NX-OS software interoperates with Cisco products that run any variant of the Cisco IOS software.

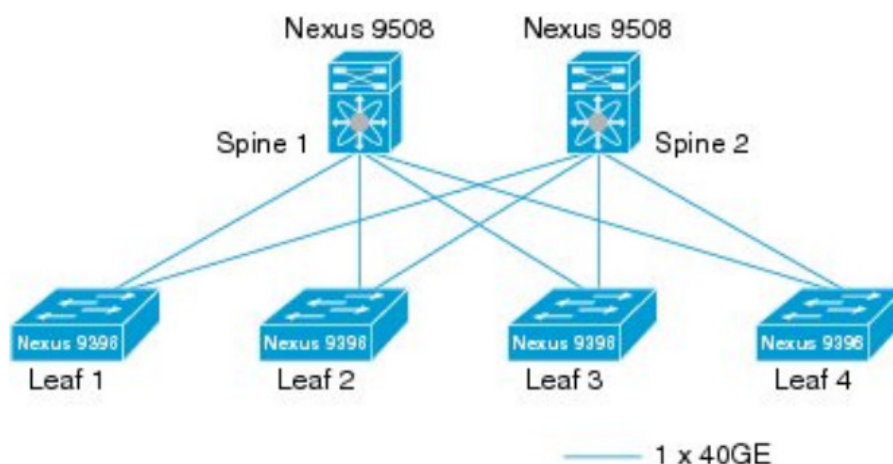
The Cisco NX-OS software also interoperates with any networking operating system that conforms to the IEEE and RFC compliance standards.

Spine/Leaf Topology

The Cisco Nexus 9000 Series switches support a two-tier spine/leaf topology.

Figure 1: Spine/Leaf Topology

This figure shows an example of a spine/leaf topology with four leaf switches (Cisco Nexus 9396 or 93128) connecting into two spine switches (Cisco Nexus 9508) and two 40G Ethernet uplinks from each leaf to each



Modular Software Design

The Cisco NX-OS software supports distributed multithreaded processing on symmetric multiprocessors (SMPs), multi-core CPUs, and distributed data module processors. The Cisco NX-OS software offloads computationally intensive tasks, such as hardware table programming, to dedicated processors distributed across the data modules. The modular processes are created on demand, each in a separate protected memory space. Processes are started and system resources are allocated only when you enable a feature. A real-time

preemptive scheduler helps to ensure the timely processing of critical functions.

Serviceability

The Cisco NX-OS software has serviceability functions that allow the device to respond to network trends and events. These features help you with network planning and improving response times.

Switched Port Analyzer

The Switched Port Analyzer (SPAN) feature allows you to analyze all traffic between ports (called the SPAN source ports) by nonintrusively directing the SPAN session traffic to a SPAN destination port that has an external analyzer attached to it. For more information about SPAN, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

Ethalyzer

Ethalyzer is a Cisco NX-OS protocol analyzer tool based on the Wireshark (formerly Ethereal) open source code. Ethalyzer is a command-line version of Wireshark for capturing and decoding packets. You can use Ethalyzer to troubleshoot your network and analyze the control-plane traffic. For more information about Ethalyzer, see the Cisco Nexus 9000 Series NX-OS Troubleshooting Guide.

Smart Call Home

The Call Home feature continuously monitors hardware and software components to provide e-mail-based notification of critical system events. A versatile range of message formats is available for optimal compatibility with pager services, standard e-mail, and XML-based automated parsing applications. It offers alert grouping capabilities and customizable destination profiles. You can use this feature, for example, to directly page a network support engineer, send an e-mail message to a network operations center (NOC), and employ Cisco Auto Notify services to directly generate a case with the Cisco Technical Assistance Center (TAC). For more information about Smart Call Home, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

Online Diagnostics

Cisco generic online diagnostics (GOLD) verify that hardware and internal data paths are operating as designed. Boot-time diagnostics, continuous monitoring, and on-demand and scheduled tests are part of the Cisco GOLD feature set. GOLD allows rapid fault isolation and continuous system monitoring. For information about configuring GOLD, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

Embedded Event Manager

Cisco Embedded Event Manager (EEM) is a device and system management feature that helps you to customize behavior based on network events as they happen. For information about configuring EEM, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

Manageability

This section describes the manageability features for the Cisco Nexus 9000 Series switches.

Simple Network Management Protocol

The Cisco NX-OS software is compliant with Simple Network Management Protocol (SNMP) version 1, version 2, and version 3. A large number of MIBs is supported. For more information about SNMP, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

Configuration Verification and Rollback

The Cisco NX-OS software allows you to verify the consistency of a configuration and the availability of necessary hardware resources prior to committing the configuration. You can preconfigure a device and apply the verified configuration at a later time. Configurations also include checkpoints that allow you to roll back to a known good configuration as needed. For more information about rollbacks, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

Role-Based Access Control

With role-based access control (RBAC), you can limit access to device operations by assigning roles to users. You can customize access and restrict it to the users who require it. For more information about RBAC, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Cisco NX-OS Device Configuration Methods

You can use these methods to configure Cisco NX-OS devices:

- The CLI from a Secure Shell (SSH) session, a Telnet session, or the console port. SSH provides a secure connection to the device. The CLI configuration guides are organized by feature. For more information, see the Cisco NX-OS configuration guides. For more information about SSH and Telnet, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.
- The XML management interface, which is a programmatic method based on the NETCONF protocol that

complements the CLI. For more information, see the Cisco NX-OS XML Interface User Guide.

- The Cisco Data Center Network Management (DCNM) client, which runs on your local PC and uses web services on the Cisco DCNM server. The Cisco DCNM server configures the device over the XML management interface. For more information about the Cisco DCNM client, see the Cisco DCNM Fundamentals Guide.

Programmability

This section describes the programmability features for the Cisco Nexus 9000 Series switches.

Python API

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 ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python website: <http://www.python.org/>. The Python scripting capability gives programmatic access to the CLI to perform various tasks and Power-On Auto Provisioning (POAP) or Embedded Event Manager (EEM) actions. For more information about the Python API and Python scripting, see the Cisco Nexus 9000 Series NX-OS Programmability Guide.

Tcl

Tool Command Language (Tcl) is a scripting language. With Tcl, you gain more flexibility in your use of the CLI commands on the device. 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.

Cisco NX-API

The Cisco NX-API provides web-based programmatic access to the Cisco Nexus 9000 Series switches. This support is delivered through the NX-API open-source web server. The Cisco NX-API exposes the complete configuration and management capabilities of the command-line interface (CLI) through web-based APIs.

You can configure the switch to publish the output of the API calls in either XML or JSON format. For more information about the Cisco NX-API, see the Cisco Nexus 9000 Series NX-OS Programmability Guide.



Note

NX-API performs authentication through a programmable authentication module (PAM) on the switch. Use cookies to reduce the number of PAM authentications and thus reduce the load on PAM.

Bash Shell

The Cisco Nexus 9000 Series switches support direct Linux shell access. With Linux shell support, you can access the Linux system on the switch in order to use Linux commands and manage the underlying system. For more information about Bash shell support, see the Cisco Nexus 9000 Series NX-OS Programmability Guide.

Broadcom Shell

The Cisco Nexus 9000 Series switch front-panel and fabric module line cards contain several Broadcom ASICs. You can use the CLI to access the command-line shell (bcm shell) for these ASICs. The benefit of using this method to access the bcm shell is that you can use Cisco NX-OS command extensions such as pipe include and redirect output to file to manage the output. In addition, the activity is recorded in the system accounting log for audit purposes, unlike commands entered directly from the bcm shell, which are not recorded in the accounting log. For more information about Broadcom shell support, see the Cisco Nexus 9000 Series NX-OS Programmability Guide.



Caution

Use Broadcom shell commands with caution and only under the direct supervision or request of Cisco Support personnel.

Traffic Routing, Forwarding, and Management

This section describes the traffic routing, forwarding, and management features supported by the Cisco NX-OS software.

Ethernet Switching

The Cisco NX-OS software supports high-density, high-performance Ethernet systems and provides the following Ethernet switching features:

- IEEE 802.1D-2004 Rapid and Multiple Spanning Tree Protocols (802.1w and 802.1s)
- IEEE 802.1Q VLANs and trunks
- IEEE 802.3ad link aggregation

- Unidirectional Link Detection (UDLD) in aggressive and standard modes

For more information, see the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide and the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide.

IP Routing

The Cisco NX-OS software supports IP version 4 (IPv4) and IP version 6 (IPv6) and the following routing protocols:

- Open Shortest Path First (OSPF) Protocol Versions 2 (IPv4) and 3 (IPv6)
- Intermediate System-to-Intermediate System (IS-IS) Protocol (IPv4 and IPv6)
- Border Gateway Protocol (BGP) (IPv4 and IPv6)
- Enhanced Interior Gateway Routing Protocol (EIGRP) (IPv4 only)
- Routing Information Protocol Version 2 (RIPv2) (IPv4 only)

The Cisco NX-OS software implementations of these protocols are fully compliant with the latest standards and include 4-byte autonomous system numbers (ASNs) and incremental shortest path first (SPF). All unicast protocols support Non-Stop Forwarding Graceful Restart (NSF-GR). All protocols support all interface types, including Ethernet interfaces, VLAN interfaces, sub interfaces, port channels, and loopback interfaces. For more information, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

IP Services

The following IP services are available in the Cisco NX-OS software:

- Virtual routing and forwarding (VRF)
- Dynamic Host Configuration Protocol (DHCP) helper
- Hot Standby Router Protocol (HSRP)
- Enhanced object tracking
- Policy-based routing (PBR)
- Unicast graceful restart for all protocols in IPv4 unicast graceful restart for OPSFv3 in IPv6

For more information, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

IP Multicast

The Cisco NX-OS software includes the following multicast protocols and functions:

- Protocol Independent Multicast (PIM) Version 2 (PIMv2)
- PIM sparse mode (Any-Source Multicast [ASM] for IPv4)
- Anycast rendezvous point (Anycast-RP)
- Multicast NSF for IPv4
- RP-Discovery using bootstrap router (BSR) (Auto-RP and static)
- Internet Group Management Protocol (IGMP) Versions 1, 2, and 3 router role
- IGMPv2 host mode
- IGMP snooping
- Multicast Source Discovery Protocol (MSDP) (for IPv4)



Note

The Cisco NX-OS software does not support PIM dense mode.

For more information, see the Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide.

Quality of Service

The Cisco NX-OS software supports quality of service (QoS) functions for classification, marking, queuing, policing, and scheduling. Modular QoS CLI (MQC) supports all QoS features. You can use MQC to provide uniform configurations across various Cisco platforms. For more information, see the Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide.

Network Security Features

The Cisco NX-OS software includes the following security features:

- Control Plane Policing (CoPP)
- Message-digest algorithm 5 (MD5) routing protocol authentication
- Authentication, authorization, and accounting (AAA)
- RADIUS and TACACS+
- SSH Protocol Version 2
- SNMPv3
- Policies based on MAC and IPv4 addresses supported by named ACLs (port-based ACLs [PACLs], VLAN-based ACLs [VACLs], and router-based ACLs [RACLs])
- Traffic storm control (unicast, multicast, and broadcast)

For more information, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Supported Standards

This table lists the IEEE compliance standards.

Table 2: IEEE Compliance Standards

Standard	Description
802.1D	MAC Bridges
802.1p	Class of Service Tagging for Ethernet frames
802.1Q	VLAN Tagging
802.1s	Multiple Spanning Tree Protocol
802.1w	Rapid Spanning Tree Protocol
802.3ab	1000Base-T (10/100/1000 Ethernet over copper)
802.3ad	Link aggregation with LACP
802.3ae	10-Gigabit Ethernet

This table lists the RFC compliance standards. For information on each RFC, see www.ietf.org.

Table 3: RFC Compliance Standards

Standard	Description
BGP	
RFC 1997	<i>BGP Communities Attribute</i>
RFC 2385	<i>Protection of BGP Sessions via the TCP MD5 Signature Option</i>
RFC 2439	<i>BGP Route flap damping</i>
RFC 2519	<i>A Framework for Inter-Domain Route Aggregation</i>
RFC 2545	<i>Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 2918	<i>Route Refresh Capability for BGP-4</i>
RFC 3065	<i>Autonomous System Confederations for BGP</i>

Standard	Description
RFC 3392	<i>Capabilities Advertisement with BGP-4</i>
RFC 4271	<i>BGP version 4</i>
RFC 4273	<i>BGP4 MIB – Definitions of Managed Objects for BGP-4</i>
RFC 4456	<i>BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)</i>
RFC 4486	<i>Subcodes for BGP cease notification message</i>
RFC 4724	<i>Graceful Restart Mechanism for BGP</i>
RFC 4893	<i>BGP Support for Four-octet AS Number Space</i>
RFC 5004	<i>Avoid BGP Best Path Transitions from One External to Another</i>
RFC 5396	<i>Textual Representation of Autonomous System (AS) Numbers</i> Note RFC 5396 is partially supported. The asplain and asdot notations are supported, but the asdot+ notation is not.
RFC 5549	<i>Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop</i>
RFC 5668	<i>4-Octet AS Specific BGP Extended Community</i>
ietf-draft	Bestpath transition avoidance (draft-ietf-idr-avoid-transition-05.txt)
ietf-draft	Peer table objects (draft-ietf-idr-bgp4-mib-15.txt)
ietf-draft	Dynamic Capability (draft-ietf-idr-dynamic-cap-03.txt)
IP Multicast	

Standard	Description
RFC 2236	<i>Internet Group Management Protocol, Version 2</i>
RFC 3376	<i>Internet Group Management Protocol, Version 3</i>
RFC 3446	<i>Anycast Rendezvous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)</i>
RFC 3569	<i>An Overview of Source-Specific Multicast (SSM)</i>
RFC 3618	<i>Multicast Source Discovery Protocol (MSDP)</i>
RFC 4601	<i>Protocol Independent Multicast – Sparse Mode (PIM-SM): Protocol Specification (Revised)</i>
RFC 4607	<i>Source-Specific Multicast for IP</i>
RFC 4610	<i>Anycast-RP Using Protocol Independent Multicast (PIM)</i>
RFC 6187	<i>X.509v3 Certificates for Secure Shell Authentication</i>
ietf-draft	Mtrace server functionality, to process mtrace-requests, draft-ietf-idmr-traceroute-ipm-07.txt
IP Services	
RFC 768	<i>UDP</i>
RFC 783	<i>TFTP</i>
RFC 791	<i>IP</i>
RFC 792	<i>ICMP</i>
RFC 793	<i>TCP</i>
RFC 826	<i>ARP</i>
RFC 854	<i>Telnet</i>
RFC 959	<i>FTP</i>
RFC 1027	<i>Proxy ARP</i>

Standard	Description
RFC 1305	<i>NTP v3</i>
RFC 1519	<i>CIDR</i>
RFC 1542	<i>BootP relay</i>
RFC 1591	<i>DNS client</i>
RFC 1812	<i>IPv4 routers</i>
RFC 2131	<i>DHCP Helper</i>
RFC 2338	<i>VRRP</i>
IS-IS	
RFC 1142 (OSI 10589)	<i>OSI 10589 Intermediate system to intermediate system intra-domain routing exchange protocol</i>
RFC 1195	<i>Use of OSI IS-IS for routing in TCP/IP and dual environment</i>
RFC 2763	<i>Dynamic Hostname Exchange Mechanism for IS-IS</i>
RFC 2966	<i>Domain-wide Prefix Distribution with Two-Level IS-IS</i>
RFC 2973	<i>IS-IS Mesh Groups</i>
RFC 3277	<i>IS-IS Transient Blackhole Avoidance</i>
RFC 3373	<i>Three-Way Handshake for IS-IS Point-to-Point Adjacencies</i>
RFC 3567	<i>IS-IS Cryptographic Authentication</i>
RFC 3847	<i>Restart Signaling for IS-IS</i>
ietf-draft	Internet Draft Point-to-point operation over LAN in link-state routing protocols (draft-ietf-isis-igp-p2p-over-lan-06.txt)
OSPF	
RFC 2328	<i>OSPF Version 2</i>
RFC 2370	<i>OSPF Opaque LSA Option</i>
RFC 2740	<i>OSPF for IPv6 (OSPF version 3)</i>

Standard	Description
RFC 3101	<i>OSPF Not-So-Stubby-Area (NSSA) Option</i>
RFC 3137	<i>OSPF Stub Router Advertisement</i>
RFC 3509	<i>Alternative Implementations of OSPF Area Border Routers</i>
RFC 3623	<i>Graceful OSPF Restart</i>
RFC 4750	<i>OSPF Version 2 MIB</i>
Per-Hop Behavior (PHB)	
RFC 2597	<i>Assured Forwarding PHB Group</i>
RFC 3246	<i>An Expedited Forwarding PHB</i>
RIP	
RFC 1724	<i>RIPv2 MIB extension</i>
RFC 2082	<i>RIPv2 MD5 Authentication</i>
RFC 2453	<i>RIP Version 2</i>
SNMP	
RFC 2579	<i>Textual Conventions for SMIv2</i>
RFC 2819	<i>Remote Network Monitoring Management Information Base</i>
RFC 2863	<i>The Interfaces Group MIB</i>
RFC 3164	<i>The BSD syslog Protocol</i>
RFC 3176	<i>InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks</i>
RFC 3411 and RFC 3418	<i>An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks</i>
RFC 3413	<i>Simple Network Management Protocol (SNMP) Applications</i>
RFC 3417	<i>Transport Mappings for the Simple Network Management Protocol (SNMP)</i>

Using the Cisco NX-OS Setup Utility

This chapter contains the following sections:

- About the Cisco NX-OS Setup Utility, on page 15
- Prerequisites for the Setup Utility, on page 16
- Setting Up Your Cisco NX-OS Device, on page 17
- Additional References for the Setup Utility, on page 22

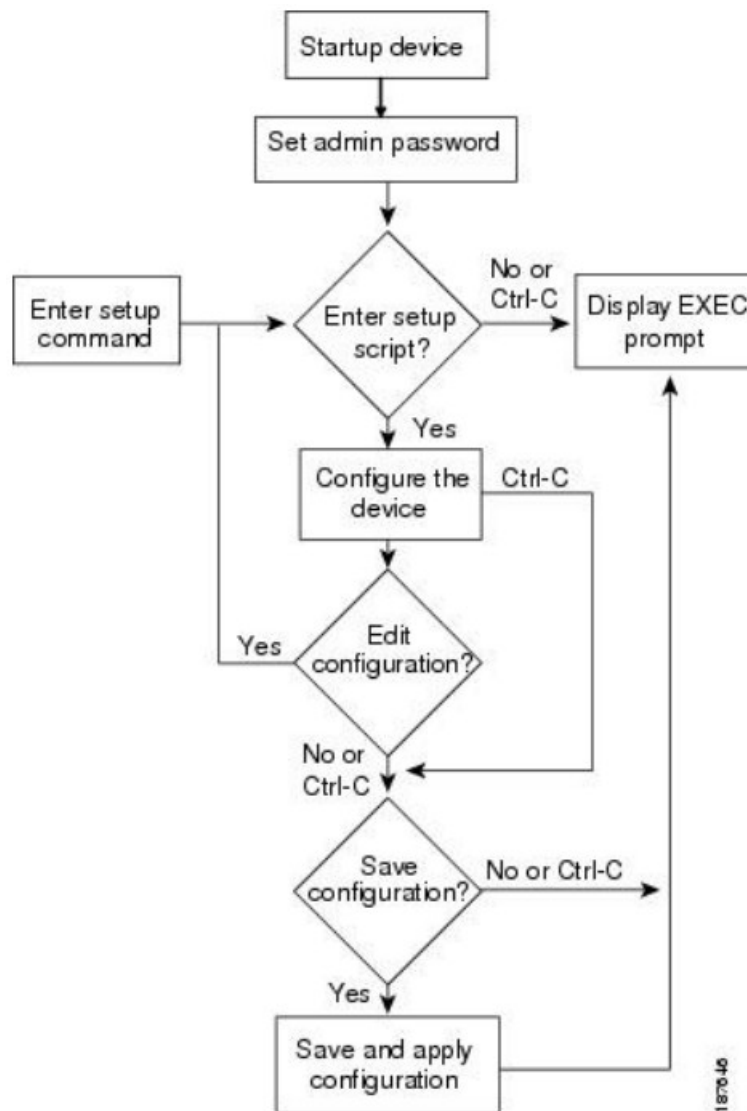
About the Cisco NX-OS Setup Utility

The Cisco NX-OS setup utility is an interactive command-line interface (CLI) mode that guides you through a basic (also called a startup) configuration of the system. The setup utility allows you to configure only enough connectivity for system management.

The setup utility allows you to build an initial configuration file using the System Configuration Dialog. The setup

starts automatically when a device has no configuration file in NVRAM. The dialog guides you through initial configuration. After the file is created, you can use the CLI to perform additional configuration. You can press Ctrl-C at any prompt to skip the remaining configuration options and proceed with what you have configured up to that point, except for the administrator password. If you want to skip answers to any questions, press Enter. If a default answer is not available (for example, the device hostname), the device uses what was previously configured and skips to the next question.

Figure 2: Setup Script Flow



This figure shows how to enter and exit the setup script.

You use the setup utility mainly for configuring the system initially, when no configuration is present. However, you can use the setup utility at any time for basic device configuration. The setup utility keeps the configured values when you skip steps in the script. For example, if you have already configured the mgmt0 interface, the setup utility does not change that configuration if you skip that step. However, if there is a default value for the step, the setup utility changes to the configuration using that default, not the configured value. Be sure to carefully check the configuration changes before you save the configuration.

Note

Be sure to configure the IPv4 route, the default network IPv4 address, and the default gateway IPv4 address to enable SNMP access. If you enable IPv4 routing, the device uses the IPv4 route and the default network IPv4 address. If IPv4 routing is disabled, the device uses the default gateway IPv4 address.

Note

The setup script only supports IPv4.

Prerequisites for the Setup Utility

The setup utility has the following prerequisites:

- Have a password strategy for your network environment.
- Connect the console port on the supervisor module to the network. If you have dual supervisor modules, connect the console ports on both supervisor modules to the network.
- Connect the Ethernet management port on the supervisor module to the network. If you have dual supervisor modules, connect the Ethernet management ports on both supervisor modules to the network.

Setting Up Your Cisco NX-OS Device

To configure basic management of the Cisco NX-OS device using the setup utility, follow these steps:

Step 1 Power on the device.

Step 2 Enable or disable password-strength checking.

A strong password has the following characteristics:

- At least eight characters long
- Does not contain many consecutive characters (such as “abcd”)
- Does not contain many repeating characters (such as “aaabbb”)
- Does not contain dictionary words
- Does not contain proper names
- Contains both uppercase and lowercase characters
- Contains numbers

Example:

— System Admin Account Setup —

Do you want to enforce secure password standard (yes/no) [y]: **y**

Step 3 Enter the new password for the administrator.

Note

If a password is trivial (such as a short, easy-to-decipher password), your password configuration is rejected. Passwords are case sensitive. Be sure to configure a strong password that has at least eight characters, both uppercase and lowercase letters, and numbers.

Example:

Enter the password for “admin”: <password>

Confirm the password for “admin”: <password>

— Basic System Configuration Dialog —

This setup utility will guide you through the basic configuration of the system. Setup configures only enough connectivity for management of the system.

Please register Cisco Nexus 9000 Family devices promptly with your supplier. Failure to register may affect response times for initial service calls. Nexus devices must be registered to receive entitled support services.

Press Enter at anytime to skip a dialog. Use ctrl-c at anytime to skip the remaining dialogs.

Step 4 Enter the setup mode by entering yes.

Example:

Would you like to enter the basic configuration dialog (yes/no): **yes**

Step 5

Create additional accounts by entering yes (no is the default).

Example:

Create another login account (yes/no) [n]:yes

a) Enter the user login ID.

Example:

Enter the User login Id : user_login

Caution

Username must begin with an alphanumeric character and can contain only these special characters:

(+ = . _ \ -). The # and ! symbols are not supported. If the username contains characters that are not allowed, the specified user is unable to log in.

b) Enter the user password.

Example:

Enter the password for "user1": user_password
Confirm the password for "user1": user_password
c) Enter the default user role.

Example:

Enter the user role (network-operator|network-admin) [network-operator]: default_user_role
For information on the default user roles, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Step 6 Configure an SNMP community string by entering yes.

Example:

Configure read-only SNMP community string (yes/no) [n]: **yes**
SNMP community string : snmp_community_string
For information on SNMP, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

Step 7

Enter a name for the device (the default name is switch).

Example:

Enter the switch name: switch_name

Step 8

Configure out-of-band management by entering yes. You can then enter the mgmt0 IPv4 address and subnet mask.

Note You can only configure IPv4 address in the setup utility. For information on configuring IPv6, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

Example:

Continue with Out-of-band (mgmt0) management configuration? [yes/no]: **yes**
Mgmt0 IPv4 address: mgmt0_ip_address
Mgmt0 IPv4 netmask: mgmt0_subnet_mask

Step 9

Configure the IPv4 default gateway (recommended) by entering yes. You can then enter its IP address.

Example:

Configure the default-gateway: (yes/no) [y]: **yes**
IPv4 address of the default-gateway: default_gateway

Step 10

Configure advanced IP options such as the static routes, default network, DNS, and domain name by entering yes.

Example:

Configure Advanced IP options (yes/no)? [n]: yes

Step 11

Configure a static route (recommended) by entering yes. You can then enter its destination prefix, destination prefix mask, and next hop IP address.

Example:

Configure static route: (yes/no) [y]: yes
Destination prefix: dest_prefix
Destination prefix mask: dest_mask
Next hop ip address: next_hop_address

Step 12

Configure the default network (recommended) by entering yes. You can then enter its IPv4 address.

Note The default network IPv4 address is the same as the destination prefix in the static route configuration.

Example:

Configure the default network: (yes/no) [y]: **yes**
Default network IP address [dest_prefix]: dest_prefix

Step 13

Configure the DNS IPv4 address by entering yes. You can then enter the address.

Example:

Configure the DNS IP address? (yes/no) [y]: **yes**
DNS IP address: ipv4_address

Step 14

Configure the default domain name by entering yes. You can then enter the name.

Example:

Configure the DNS IP address? (yes/no) [y]: **yes**
DNS IP address: ipv4_address

Step 15

Enable the Telnet service by entering **yes**.

Example:

Enable the telnet service? (yes/no) [y]: **yes**

Step 16

Enable the SSH service by entering **yes**. You can then enter the key type and number of key bits. For more information, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Example:

Enable the ssh service? (yes/no) [y]: **yes**

Type of ssh key you would like to generate (dsa/rsa) : **key_type**

Number of key bits <768-2048> : **number_of_bits**

Step 17

Configure the NTP server by entering **yes**. You can then enter its IP address. For more information, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

Example:

Configure NTP server? (yes/no) [n]: **yes**

NTP server IP address: **ntp_server_IP_address**

Step 18

Specify a default interface layer (L2 or L3).

Example:

Configure default interface layer (L3/L2) [L3]: **interface_layer**

Step 19

Enter the default switchport interface state (shutdown or no shutdown). A shutdown interface is in an administratively

down state. For more information, see the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.

Example:

Configure default switchport interface state (shut/noshut) [shut]: **default_state**

Step 20

Enter **yes** (no is the default) to configure basic Fibre Channel configurations.

Example:

Enter basic FC configurations (yes/no) [n]: **yes**

Step 21

Enter **shut** (noshut is the default) to configure the default Fibre Channel switch port interface to the shut (disabled) state.

Example:

Configure default physical FC switchport interface state (shut/noshut) [noshut]: **shut**

Step 22

Enter **on** (on is the default) to configure the switch port trunk mode

Example:

Configure default physical FC switchport trunk mode (on/off/auto) [on]: **on**

Step 23

Enter **permit** (deny is the default) to permit a default zone policy configuration.

Example:

Configure default zone policy (permit/deny) [deny]: **permit** Permits traffic flow to all members of the default zone.

Example:

Note

If you are executing the setup script after entering a write erase command, you explicitly must change the default zone policy to permit for VSAN 1 after finishing the script using the following command:

```
switch(config)# zone default-zone permit vsan 1
```

Step 24

Enter **yes** (no is the default) to enable a full zone set distribution.

Example:

Enable full zoneset distribution (yes/no) [n]: **yes**

Step 25

Enter the best practices profile for control plane policing (CoPP). For more information, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Example:

Configure best practices CoPP profile (strict/moderate/lenient/none) [strict]: **moderate** The system now summarizes the complete configuration and asks if you want to edit it.

Step 26

Continue to the next step by entering no. If you enter yes, the setup utility returns to the beginning of the setup and repeats each step.

Example:

Would you like to edit the configuration? (yes/no) [y]: yes

Step 27

Use and save this configuration by entering yes. If you do not save the configuration at this point, none of your changes are part of the configuration the next time the device reboots. Enter yes to save the new configuration. This step ensures that the boot variables for the nx-os image are also automatically configured.

Example:

Use this configuration and save it? (yes/no) [y]: yes

Caution

If you do not save the configuration at this point, none of your changes are part of the configuration the next time that the device reboots. Enter yes to save the new configuration to ensure that the boot variables for the nx-os image are also automatically configured.

Additional References for the Setup Utility

This section includes additional information related to using the setup utility.

Related Documents for the Setup Utility

Related Topic	Document Title
Licensing	<i>Cisco NX-OS Licensing Guide</i>
SSH and Telnet	<i>Cisco Nexus 9000 Series NX-OS Security Configuration Guide</i>
User roles	<i>Cisco Nexus 9000 Series NX-OS Security Configuration Guide</i>
IPv4 and IPv6	<i>Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide</i>
SNMP and NTP	<i>Cisco Nexus 9000 Series NX-OS System Management Configuration Guide</i>

Using PowerOn Auto Provisioning

This chapter contains the following sections:

- About PowerOn Auto Provisioning, on page 23
- POAPv3, on page 40
- Guidelines and Limitations for POAP, on page 42
- Setting Up the Network Environment to Use POAP, on page 44
- Configuring a Switch Using POAP, on page 44
- Creating md5 Files, on page 45
- Verifying the Device Configuration, on page 46
- Troubleshooting for POAP, on page 47
- Managing the POAP Personality, on page 48

About PowerOn Auto Provisioning

PowerOn Auto Provisioning (POAP) automates the process of upgrading software images and installing configuration files on devices that are being deployed in the network for the first time.

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



Note

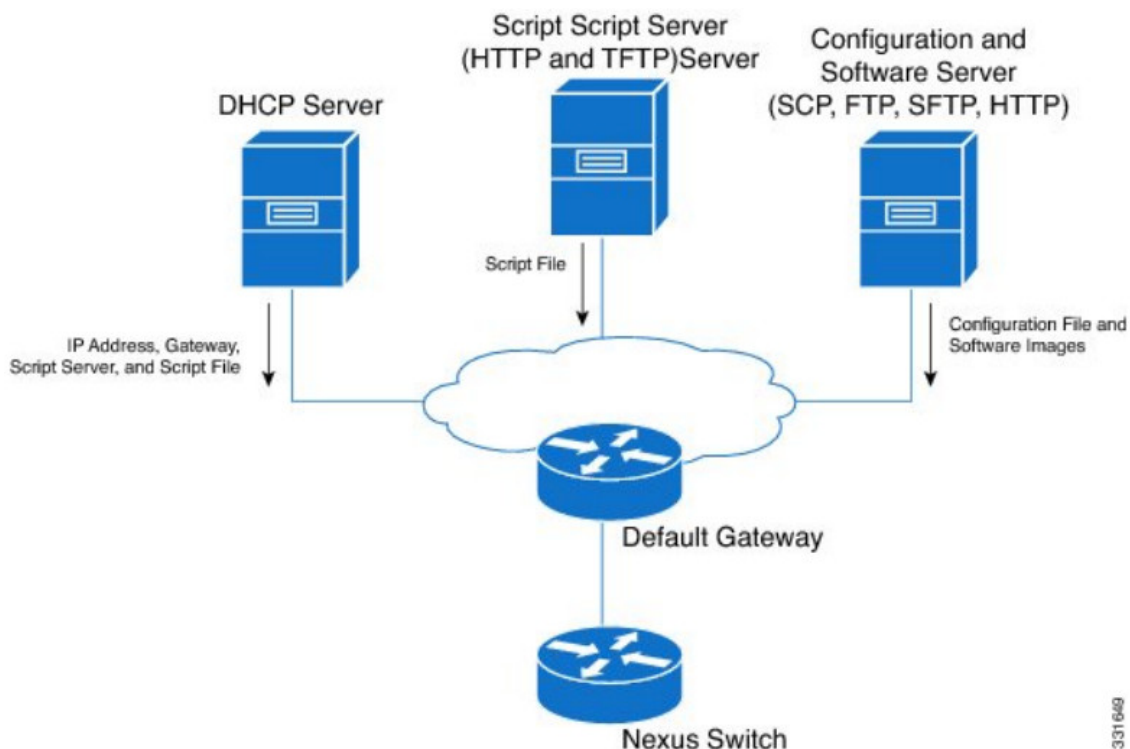
The DHCP information is used only during the POAP process.

Network Requirements for POAP

POAP requires the following network infrastructure:

- A DHCP server to bootstrap the interface IP address, gateway address, and Domain Name System (DNS) server.
- A TFTP server that contains the configuration script used to automate the software image installation and configuration process.
- One or more servers that contains the desired software images and configuration files.
- If you use USB, then no DHCP server or TFTP server are required for POAP.

Figure 3: POAP Network Infrastructure



Secure Download of POAP Script

Beginning with Cisco NX-OS Release 10.2(3)F, you have the option of securely downloading the POAP script. When a device with the POAP feature boots and does not find the startup configuration, the device enters POAP mode, locates a DHCP server, and bootstraps itself with its interface IP address, gateway, and DNS server IP addresses. The device also obtains the IP address of an HTTPS server and downloads POAP script securely. The script enables the switch to download and install the appropriate software image and configuration file.

To download the POAP script securely, you need to select specific POAP options. Until Cisco NX-OS Release 10.2(3)F, POAP used options 66 and 67 for IPv4, and options 77 and 15 for IPv6 to extract the booting script information. However, the transfer of the script uses http, and is not very secure. Beginning with Cisco NX-OS Release 10.2(3)F, option 43 specifies the secure POAP related provisioning script information for IPv4 and option 17 specifies the same for IPv6. Additionally, these options allow the POAP to reach the file server in a secure manner. The POAP options 66, 67, 77, and 15 continue to be supported in Cisco NX-OS Release 10.2(3)F. Furthermore, if you are using option 43 or 17, you can use the earlier options as fallback options, if required. From Cisco NX-OS Release 10.4(1)F, you can use Root-CA bundles instead of single .pem certificate for Secure POAP.



Note

The maximum character length is 512 bytes for both option 43 and option 17.

The sub-options available for option 43 and option 17 are discussed in the following sections:

- Option 43 – IPv4

- Option 17 – IPv6

IPv4

Option 43 has the following sub-options for IPv4:

- option space poap length width 2;
- option poap.version code 1 = unsigned integer 8;



This sub-option is mandatory.

- option poap.ca_list code 50 = text;
- option poap.url code 2 = text;



This sub-option is mandatory.

- option poap.debug code 51 = unsigned integer 8;
- option poap.ntp code 3 = ip-address;



This sub-option is only supported for IPv4 (Option 43).

- option poap.flag code 52 = unsigned integer 8;



Flag is used to skip server certificate validation in the client.

Sample configuration for IPv4 is as follows:

```
host dhclient-n9kv {
hardware ethernet 00:50:56:85:c5:30;
fixed-address 3.3.3.1;
default-lease-time 3600;
option broadcast-address 192.168.1.255;
#option log-servers 1.1.1.1;
max-lease-time 3600;
option subnet-mask 255.255.255.0;
option routers 10.77.143.1;
#option domain-name-servers 1.1.1.1;
    vendor-option-space poap;
option poap.version 1;
option poap.ca_list "https://<ip>/poap/ca_file1.pem, https://<ip>/poap/ca_file2.pem";
option poap.url "https://<url>/poap.py";
option poap.debug 1;
option poap.ntp 10.1.1.39;
option poap.flag 0;
}
```

IPv6

Option 17 has the following sub-options for IPv6:

- option space poap_v6 length width 2;
- option poap_v6.version code 1 = unsigned integer 8;



Note

This sub-option is mandatory.

- option poap_v6.ca_list code 50 = text;
- option poap_v6.url code 3 = text;



Note

This sub-option is mandatory.

- option poap_v6.debug code 51 = unsigned integer 8;
- option vsio.poap_v6 code 9 = encapsulate poap_v6;

Sample configuration for IPv6 is as follows:

```
option dhcp6.next-hop-rt-prefix code 242 = { ip6-address, unsigned integer 16,
unsigned integer 16, unsigned integer 32, unsigned integer 8, unsigned integer 8, ip6-address
};
option dhcp6.bootfile-url code 59 = string;

default-lease-time 3600;
max-lease-time 3600;
log-facility local7;
subnet6 2003::/64 {

    # This statement configures actual values to be sent
    # RTPREFIX option code = 243, RTPREFIX length = 22
    # Ignore value 22. It is something related to option-size RT_PREFIX option length.
    # lifetime = 9000 seconds
    # route ETH1_IPV6_GW/64
    # metric 1
    option dhcp6.next-hop-rt-prefix 2003::2222 243 22 9000 0 1 ::;
    #ipv6 ::/0 2003::2222
    #Another example - support not there in NXOS - CSCvs05271:
    #option dhcp6.next-hop-rt-prefix 2003::2222 243 22 9000 112 1 2003::1:2:3:4:5:0;
    #ipv6 2003::1:2:3:4:5:0/112 2003::2222

    # Additional options
    #option dhcp6.name-servers fec0:0:0:1::1;
    #option dhcp6.domain-search "domain.example";

    range6 2003::b:1111 2003::b:9999;
    option dhcp6.bootfile-url "tftp://2003::1111/poap_github_v6.py";
    vendor-option-space poap_v6;
    option poap_v6.version 1;
    option poap_v6.ca_list "https://<ip>/new_ca.pem,https://<ip>/another_ca.pem";
    option poap_v6.url "https://<ip>/poap_github_v4.py";

    option poap_v6.debug 1;
}
```

Network Requirements for Secure POAP

Secure POAP requires the following network infrastructure:

- A DHCP server to bootstrap the interface IP address, gateway address, and Domain Name System (DNS) server.
- An HTTPS server that contains the POAP script used for software image installation and configuration process.



- If the HTTPS server runs on a non-SUDI device, a physical USB drive with the CA certificates of the file-server is required.
- In case of secure download of POAP script, the TFTP server is replaced with the HTTPS server. Hence, when you read the content related to the TFTP server in this chapter, remember to read the TFTP server as the HTTPS server.
- One or more servers that contain the desired software images and configuration files.

Deployment Scenarios

Cisco devices have a unique identifier known as the Secure Unique Device Identifier (SUDI). The hardware SUDI can be used for authentication, as it can be used for asymmetric key operations such as encryption, decryption, signing, and verifying that allow passage of the data to be operated upon. All non-Cisco devices are classified as non-SUDI devices. For a non-SUDI device, the root-CA bundle is required to authenticate the file server. However, the file server can be hosted on either a SUDI or a non-SUDI device.

Based on all these capabilities, you can use one of the following deployment scenarios to download the POAP script in a secure way:

- SUDI Supported Device as File Server
- Non-SUDI Supported Device as a File Server

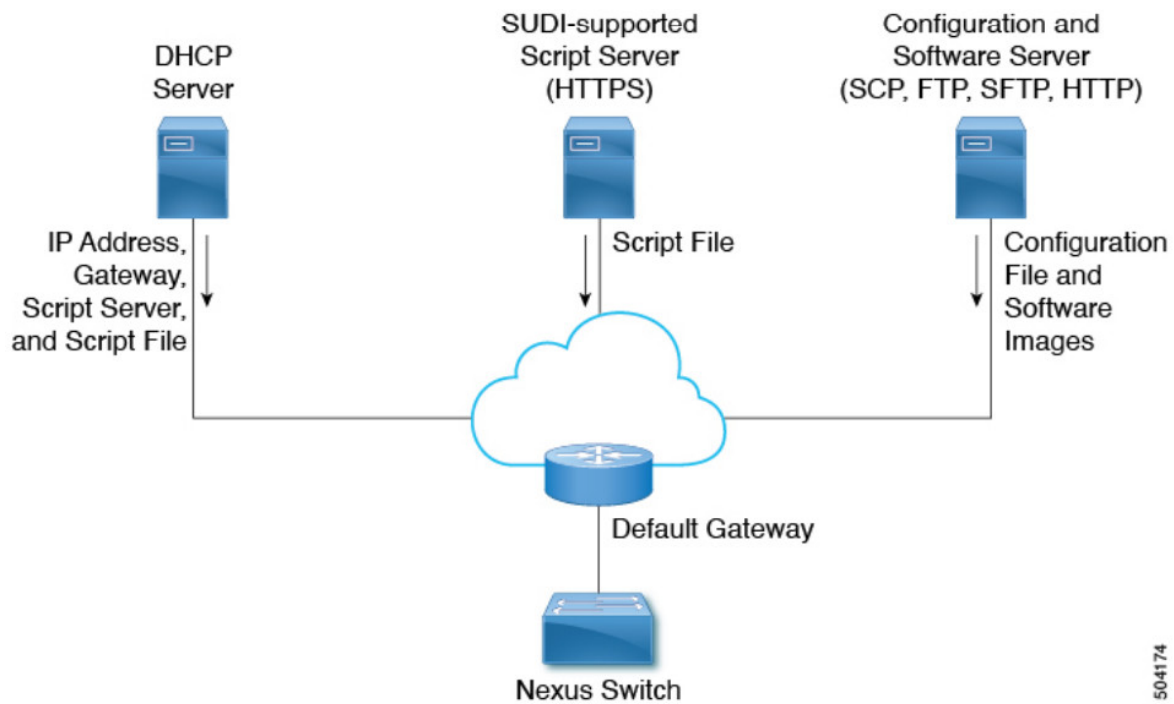
SUDI Supported Device as File Server

The SUDI supported devices are Cisco devices. Unlike the earlier implementation, the DHCP server now provides a https location rather than http/tftp. In this scenario, only the DHCP server and the SUDI supported script server (HTTPS server) are required, other than one or more servers that contain the required software images and configuration files.



SUDI only supports TLSv1.2 or below. Also, the SUDI solution only considers secure download using https, but not sftp.

Figure 4: SUDI Supported Device as File Server Infrastructure



The workflow for SUDI supported devices is as follows:

- Booting device is SUDI capable and has the needed trust store to verify a SUDI certificate
- Booting device sends out DHCP discover
- DHCP server responds to booting device with https server details
- Device establishes the secure channel using standard SSL APIs
- Authentication is done by verifying SUDI on both sides
- Downloads poap.py

Non-SUDI Supported Device as a File Server

In this scenario, the Root-CA bundle must be installed in the booting device. The Root-CA bundle is required for authentication. Here, the DHCP server, intermediate device, and non-SUDI supported script server (HTTPS server) are required, other than one or more servers that contain the required software images and configuration files.

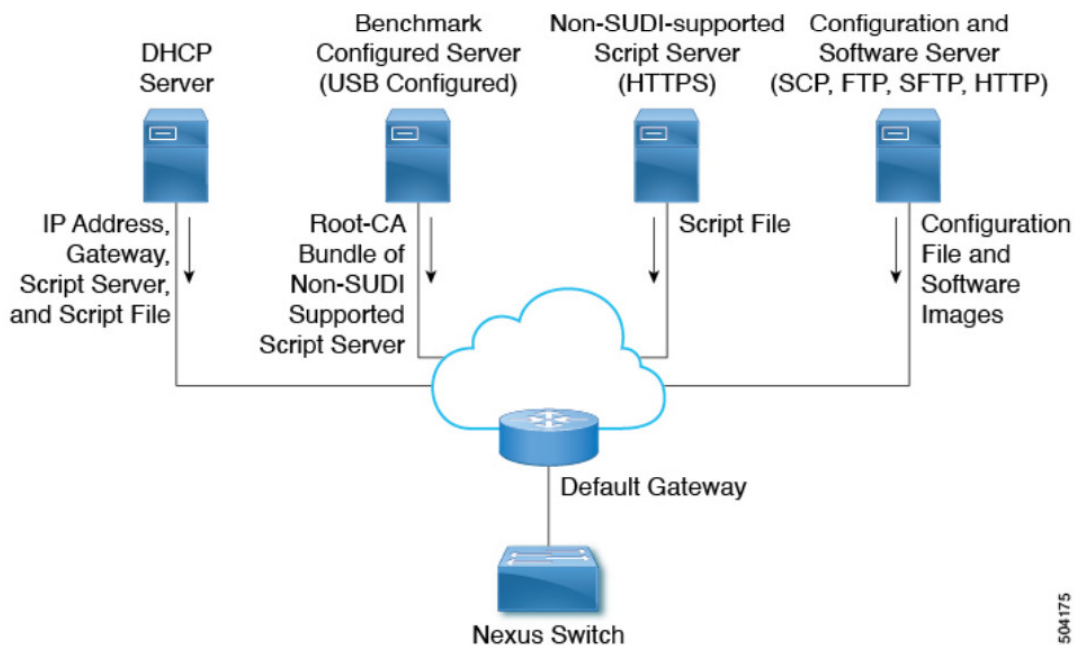
The DHCP offer has the details of intermediate server that has the Root-CA bundle available. The intermediate device should support SUDI. The booting device uses the intermediate device to download the Root-CA bundle, install it, and then communicate with the file server. The intermediate devices should be provisioned first.



Note

The intermediate device requires that you provide the Root-CA bundle manually. For more information, see Bench Configured Device Hosting Root-CA Bundle.

Figure 5: Non-SUDI Supported Device as File Server Infrastructure



504175

The workflow for non-SUDI supported devices is as follows:

- Booting device is SUDI capable and has the needed trust store to verify a SUDI certificate
- Intermediate device that hosts a server with Root-CA bundle is also SUDI capable
- Booting device sends out DHCP discover
- DHCP server responds to booting device with https server details and Root-CA server details
- Booting device reaches to intermediate device, gets the CA bundle, adds it to the trust store
- Booting device reaches the file server to download poap.py

Bench Configured Device Hosting Root-CA Bundle

A bench configured device requires manual intervention during bootup to install the Root-CA bundle by inserting a USB drive.

The workflow is as follows:

- Devices acting as intermediate devices, should be supplied with a USB drive during bootup.
- This USB drive will have poap_usb.py and Root-CA bundle.
- The poap_usb.py file in the USB copies Root-CA to the device, adds Root-CA to trust store, and returns a failure to POAP to trigger DHCP discover.



Note

- The required script is available as a template in GitHub.
- To change port on the Bench Configured Device, use the file-server <port-number> command. Avoid using standard ports such as port 80 (http) and port 443 (https).
- The DHCP discover phase helps in provisioning the device.
- When the device boots up after provisioning, it has an additional server that hosts the Root-CA bundle.

Secure POAP on a Device Shipped with Old Image

Support for secure POAP will be available only for devices that are shipped with image that has secure POAP feature.

If the device does not have the secure POAP feature, then use the legacy DHCP options to move the device to a later version of the image that supports secure POAP. Then these devices can be reloaded and use the secure POAP feature.

Troubleshooting Secure POAP

Perform the following steps to collect debugging information regarding secure POAP:

1. Set the debug option for IPv4 in option 43 to 1 and for IPv6 in option 17. The debug option enables additional logs.
2. Allow the switch to run one cycle of POAP.
3. Abort POAP.
4. When the system boots up, run the show tech-support poap command.

This command displays POAP status and configuration.

Disabling POAP

POAP is enabled when there is no configuration in the system. It runs as a part of bootup. However, you can bypass POAP enablement during initial setup. If you want to disable POAP permanently (even when there is no configuration in the system), you can use the 'system no poap' command. This command ensures that POAP is not started during the next boot (even if there is no configuration). To enable POAP, use the 'system poap' command or the 'write erase poap' command. The 'write erase poap' command erases the POAP flag and enables POAP.

• Example: Disabling POAP

```
switch# system no poap
switch# sh boot
Current Boot Variables:
  sup-1
NXOS variable = bootflash:/nxos.9.2.1.125.bin
Boot POAP Disabled
```

```
POAP permanently disabled using 'system no poap'
```

```
Boot Variables on next reload:
```

```
sup-1  
NXOS variable = bootflash:/nxos.9.2.1.125.bin  
Boot POAP Disabled
```

```
POAP permanently disabled using 'system no poap'
```

```
switch# sh system poap  
System-wide POAP is disabled using exec command 'system no poap'  
POAP will be bypassed on write-erase reload.  
(Perpetual POAP cannot be enabled when system-wide POAP is disabled)
```

· Example: Enabling POAP

```
switch# system poap  
  
switch# sh system poap  
  
System-wide POAP is enabled
```

· Example: Erase POAP

```
switch# write erase poap  
This command will erase the system wide POAP disable flag only if it is set.  
Do you wish to proceed anyway? (y/n) [n] y  
System wide POAP disable flag erased.  
  
switch# sh system poap  
System-wide POAP is enabled
```

POAP Configuration Script

The reference script supplied by Cisco supports the following functionality:

- Retrieves the switch-specific identifier, for example, the serial number.
- Downloads the nx-os software image if the files do not already exist on the switch. The nx-os image is installed on the switch and is used at the next reboot.
- Schedules the downloaded configuration to be applied at the next switch reboot.
- Stores the configuration as the startup configuration.

Cisco has sample configuration scripts that were developed using the Python programming language and Tool Command Language (Tcl). You can customize one of these scripts to meet the requirements of your network environment. You can access the Python script to perform POAP on the Cisco Nexus 9000 Series switch at this link: <https://github.com/datacenter/nexus9000/tree/master/nx-os/poap>.

The Python programming language uses two APIs that can execute CLI commands. These APIs are described in the following table. The arguments for these APIs are strings of the CLI commands.

API	Description
cli()	Returns the raw output of CLI commands, including the control/special characters.
clid()	For CLI commands that support XML, this API puts the command output in a Python dictionary. This API can be useful to help search the output of show commands.

POAP Configuration Script

We provide a sample configuration script that is developed using the Python programming language. We recommend using the provided script and modifying it to meet the requirements of your network environment. The POAP script can be found at <https://github.com/datacenter/nexus9000/blob/master/nx-os/poap/poap.py>. To modify the script using Python, see the Cisco NX-OS Python API Reference Guide for your platform.

Using the POAP Script and POAP Script Options

Before using the POAP script, perform the following actions:

1. Edit the options dictionary at the top of the script to ensure that all relevant options for your setup are included in the script. Do not change the defaults (in the default options function) directly.
2. Update the MD5 checksum of the POAP script as shown using shell commands.

```
f=poap_nexus_script.py ; cat $f | sed '/^#md5sum/d' > $f.md5 ; sed -i  
"s/^#md5sum=.*/#md5sum=\"$f(md5sum $f.md5 | sed 's/.*//')\"/" $f
```
3. If the device has a startup configuration, perform a write erase and reload the device.

The following POAP script options can be specified to alter the POAP script behavior. When you download files from a server, the hostname, username, and password options are required. For every mode except personality, the target_system_image is also required. Required parameters are enforced by the script, and the script aborts if the required parameters are not present. Every option except hostname, username, and password has a default option. If you do not specify the option in the options dictionary, the default is used.

- **username**

The username to use when downloading files from the server.

- **password**

The password to use when downloading files from the server.

- **hostname**

The name or address of the server from which to download files.

- **mode**

The default is serial_number.

Use one of the following options:

- **personality**

A method to restore the switch from a tarball.

- **serial_number**

The serial number of the switch to determine the configuration filename. The format for the serial number in the configuration file is conf.serialnumber. Example: conf.FOC123456

- **hostname**

The hostname as received in the DHCP options to determine the configuration filename. The format for the hostname in the configuration file is conf_hostname.cfg. Example: conf_3164-RS.cfg

- **mac**

The interface MAC address to determine the configuration filename. The format for the hostname in the configuration file is conf_macaddress.cfg. Example: conf_7426CC5C9180.cfg

- **raw**

The configuration filename is used exactly as provided in the options. The filename is not altered in any way.

- **location**

The CDP neighbors are used to determine the configuration filename. The format for the location in the configuration file is conf_host_intf.cfg, where host is the host connected to the device over the POAP interface, and intf is the remote interface to which the POAP interface is connected.

Example: conf_remote-switch_Eth1_8.cfg

- **required_space**

The required space in KB for that particular iteration of POAP. The default is 100,000. For multi-step upgrades, specify the size of the last image in the upgrade path of the target image.

- **transfer_protocol**

Any transfer protocol such as http, https, ftp, scp, sftp, or tftp that is supported by VSH. The default is scp.

- **config_path**

The path to the configuration file on the server. Example: /tftpboot. The default is /var/lib/tftpboot.

- **target_system_image**

The name of the image to download from the remote server. This is the image you get after POAP completes. This option is a required parameter for every mode except personality. The default is "".

- **target_image_path**

The path to the image on the server. Example: /tftpboot. The default is /var/lib/tftpboot.

- **destination_path**

The path to which to download images and MD5 sums. The default is /bootflash.

- **destination_system_image**

The name for the destination image filename. If not specified, the default will be the target_system_image name.

- **user_app_path**

The path on the server where the user scripts, agents, and user data are located. The default is /var/lib/tftpboot.

- **disable_md5**

This is True if MD5 checking should be disabled. The default is False.

- **midway_system_image**

The name of the image to use for the midway system upgrade. By default, the POAP script finds the name of any required midway images in the upgrade path and uses them. Set this option if you prefer to pick a different midway image for a two-step upgrade. The default is "".

- **source_config_file**

The name of the configuration file when raw mode is used. The default is poap.cfg.

- **vrf**

The VRF to use for downloads and so on. The VRF is automatically set by the POAP process. The default is the POAP_VRF environment variable.

- **destination_config**

The name to use for the downloaded configuration. The default is poap_replay.cfg.

- **split_config_first**

The name to use for the first configuration portion if the configuration needs to be split. It is applicable only when the configuration requires a reload to take effect. The default is poap_1.cfg.

- **split_config_second**

The name to use for the second configuration portion if the configuration is split. The default is poap_2.cfg.

- **timeout_config**

The timeout in seconds for copying the configuration file. The default is 120. For non-legacy images, this option is not used, and the POAP process times out. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **timeout_copy_system**

The timeout in seconds for copying the system image. The default is 2100. For non-legacy images, this option is not used, and the POAP process times out. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **timeout_copy_personality**

The timeout in seconds for copying the personality tarball. The default is 900. For non-legacy images, this option is not used, and the POAP process times out. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **timeout_copy_user**

The timeout in seconds for copying any user scripts and agents. The default is 900. For non-legacy images, this option is not used, and the POAP process times out. For legacy images, FTP uses this timeout for the login process and not for the copy process, while scp and other protocols use this timeout for the copy process.

- **personality_path**

The remote path from which to download the personality tarball. Once the tarball is downloaded and the personality process is started, the personality will download all files in the future from locations specified inside the tarball configuration. The default is /var/lib/tftpboot.

- **source_tarball**

The name of the personality tarball to download. The default is personality.tar.

- **destination_tarball**

The name for the downloaded personality tarball after it is downloaded. The default is personality.tar.

Setting up the DHCP Server without DNS for POAP

Beginning with Cisco NX-OS Release 7.0(3)I6(1), the tftp-server-name can be used without the DNS option.

To enable POAP functionality without DNS on earlier releases, a custom option of 150 must be used to specify the tftp-server-address.

To use the tftp-server-address option, specify the following at the start of your dhcpd.conf file.

option tftp-server-address code 150 = ip-address;

For example:

```
host MyDevice {
    option dhcp-client-identifier "\000SAL12345678";
    fixed-address 2.1.1.10;
    option routers 2.1.1.1;
    option host-name "MyDevice";
    option bootfile-name "poap_nexus_script.py";
    option tftp-server-address 2.1.1.1;
}
```

Downloading and Using User Data, Agents, and Scripts as part of POAP

Under the options dictionary, you can find the download_scripts_and_agents function. If you choose to download user scripts and data, uncomment the first poap_log line and then use a series of download_user_app function calls to download each application. Since older Cisco NX-OS versions do not support recursive copy of

directories, such directories must be put into a tarball (TAR archive) and then unpacked once on the switch. The parameters for the `download_scripts_and_agents` function are as follows:

- `source_path` – The path to where the file or tarball is located. This is a required parameter. Example: `/var/lib/tftpboot`.
- `source_file` – The name of the file to download. This is a required parameter. Example: `agents.tar`, `script.py`, and so on.
- `dest_path` – The location to download the file on the switch. Any directories that do not exist earlier will be created. This is an optional parameter. The default is `/bootflash`.
- `dest_file` – The name to give the downloaded file. This is an optional parameter. The default is unchanged `source_file`.
- `unpack` – Indicates whether a tarball exists for unpacking. Unpacking is done with `tar -xf tarfile -C /bootflash`. This is an optional parameter. The default is `False`.
- `delete_after_unpack` – Indicates whether to delete the downloaded tarball after unpack is successful. There is no effect if `unpack` is `False`. The default is `False`.

Using the download functionality, you can download all the agents and files needed to run POAP. To start the agents, you should have the configuration present in the running configuration downloaded by POAP. Then the agents, scheduler, and cron entry, along with EEM, can be used.

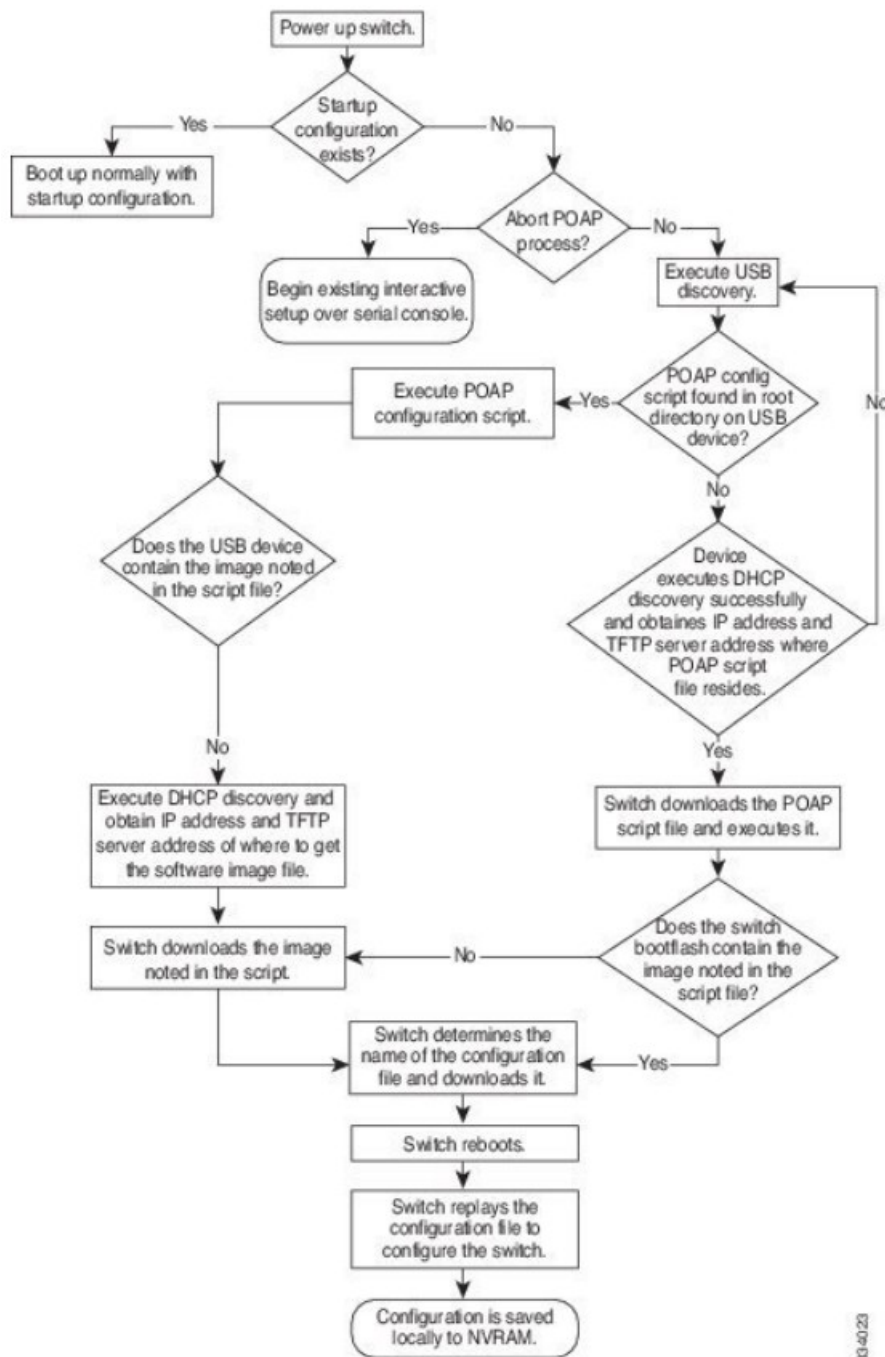
POAP Process

The POAP process has the following phases:

1. Power up
2. USB discovery
3. DHCP discovery
4. Script execution
5. Post-installation reload

Within these phases, other process and decision points occur. The following illustration shows a flow diagram of the POAP process.

Figure 6: POAP Process



33-4023

Power-Up Phase

When you powerup the device for the first time, it loads the software image that is installed at manufacturing and tries to find a configuration file from which to boot. When a configuration file is not found, POAP mode starts. During startup, a prompt appears asking if you want to abort POAP and continue with a normal setup. You can choose to exit or continue with POAP.



Note

No user intervention is required for POAP to continue. The prompt that asks if you want to abort POAP remains available until the POAP process is complete.

If you exit POAP mode, you enter the normal interactive setup script. If you continue in POAP mode, all the front-panel interfaces are set up in the default configuration.

USB Discovery Phase

When POAP starts, the process searches the root directory of all accessible USB devices for the POAP script file (the Python script file, `poap_script.py`), configuration files, and system and kickstart images.

If the script file is found on a USB device, POAP begins running the script. If the script file is not found on the USB device, POAP executes DHCP discovery. (When failures occur, the POAP process alternates between USB discovery and DHCP discovery, until POAP succeeds or you manually abort the POAP process.)

If the software image and switch configuration files specified in the configuration script are present, POAP uses those files to install the software and configure the switch. If the software image and switch configuration files are

not on the USB device, POAP does some cleanup and starts DHCP phase from the beginning.

DHCP Discovery Phase

The switch sends out DHCP discover messages on the front-panel interfaces or the MGMT interface that solicit DHCP offers from the DHCP server or servers. (See the following figure.) The DHCP client on the Cisco Nexus switch uses the switch serial number in the client-identifier option to identify itself to the DHCP server. The DHCP server can use this identifier to send information, such as the IP address and script filename, back to the DHCP client.

POAP requires a minimum DHCP lease period of 3600 seconds (1 hour). POAP checks the DHCP lease period. If the DHCP lease period is set to less than 3600 seconds (1 hour), POAP does not complete the DHCP negotiation.

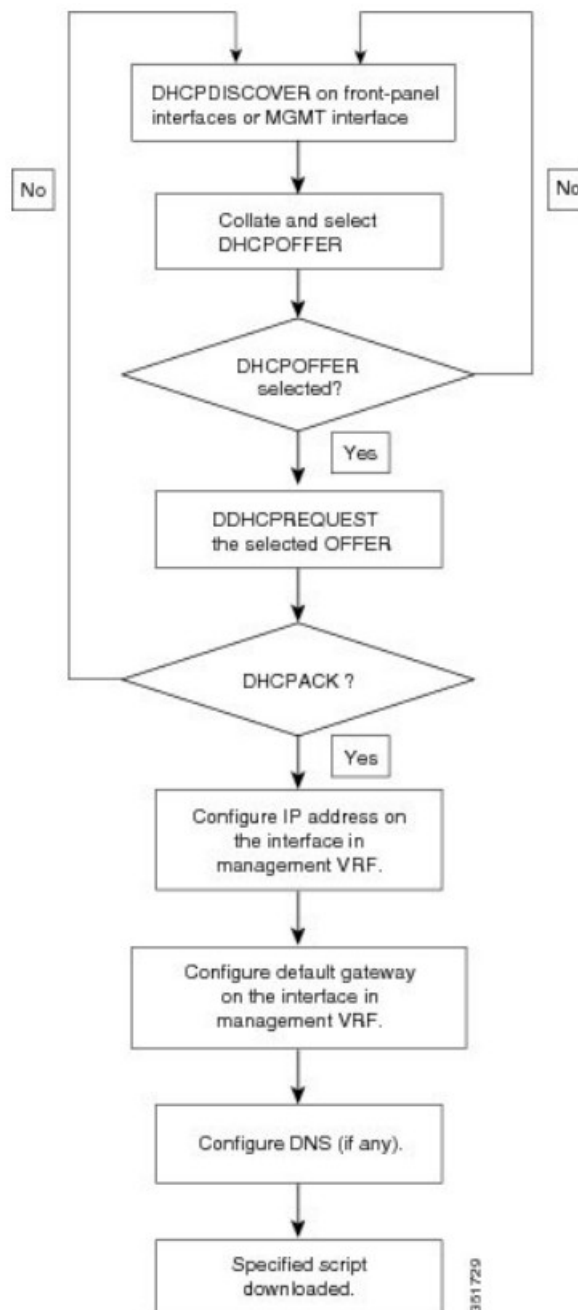
The DHCP discover message also solicits the following options from the DHCP server:

- TFTP server name or TFTP server address—The DHCP server relays the TFTP server name or TFTP server address to the DHCP client. The DHCP client uses this information to contact the TFTP server to obtain the script file.
- Bootfile name—The DHCP server relays the bootfile name to the DHCP client. The bootfile name includes the complete path to the bootfile on the TFTP server. The DHCP client uses this information to download the script file.

When multiple DHCP offers that meet the requirement are received, the one arriving first is honored and the POAP process moves to next stage. The device completes the DHCP negotiation (request and acknowledgment) with the selected DHCP server, and the DHCP server assigns an IP address to the switch. If a failure occurs in any of the subsequent steps in the POAP process, the IP address is released back to the DHCP server.

If no DHCP offers meet the requirements, the switch does not complete the DHCP negotiation (request and acknowledgment) and an IP address is not assigned.

Figure 7: DHCP Discovery Process



POAP Dynamic Breakout

Beginning with Cisco NX-OS Release 7.0(3)I4(1), POAP dynamically breaks out ports in an effort to detect a DHCP server behind one of the broken-out ports. Previously, the DHCP server used for POAP had to be directly connected to a normal cable because breakout cables were not supported.

POAP determines which breakout map (for example, 10gx4, 50gx2, 25gx4, or 10gx2) will bring up the link connected to the DHCP server. If breakout is not supported on any of the ports, POAP skips the dynamic breakout process. After the breakout loop completes, POAP proceeds with the DHCP discovery phase as normal.



Note

For more information on dynamic breakout, see the interfaces configuration guide for your device.

Script Execution Phase

After the device bootstraps itself using the information in the DHCP acknowledgement, the script file is downloaded from the TFTP server.

The switch runs the configuration script, which downloads and installs the software image and downloads a switch-specific configuration file.

However, the configuration file is not applied to the switch at this point, because the software image that currently runs on the switch might not support all of the commands in the configuration file. After the switch reboots, it begins running the new software image, if an image was installed. At that point, the configuration is applied to the switch.



Note

If the switch loses connectivity, the script stops, and the switch reloads its original software images and bootup variables.

Post-Installation Reload Phase

The switch restarts and applies (replays) the configuration on the upgraded software image. Afterward, the switch copies the running configuration to the startup configuration.

POAPv3

PowerOn Auto Provisioning version 3 (POAPv3) is introduced in Cisco NX-OS Release 9.3(5). With this feature you can install license, RPM, and certificate through POAP.

Perform the following steps to install license or RPM or certificate through POAP.

1. Create a folder on the POAP server with serial number of the box as the name.
2. Create .yaml or .yml file with files to be installed. Make sure the file name is in <serial-number>.yaml or <serial-number>.yml format.
3. Create MD5 checksum for the .yaml or .yml file.
4. Make sure the format of the .yaml file should be similar to the below format:

```
Version : 1

Target-image : nxos.9.3.4.bin

Description : Yaml for box XYZ12345 poap provisioning. N9k Leaf mode box

License : [license1.lic, XYZ12345/license2.lic, folder1/license3.lic]

RPM :

- rpml.rpm

- patches/reload/rpm2-reload.rpm

- rpm3.rpm

Certificate : [ssh1.pub, XYZ12345/ssh2key.pub]

Trustpoint :

  CA1 :

    cert_1.p12 : password1 (priv_key_passphrase)

    XYZ12345/CA1/cert_2.pfx : password2

  CA2 :

    CA2/XYZ12345/cert_3.p12 : password3
```

5. Note that the yaml keywords must match the format shown in above example.
6. Place all files in appropriate path.
7. Update the POAP script with install_path variable as the path where folder with the serial number as name is placed.

The following list provides the guidelines and limitations related to POAPv3:

- YAML is a human friendly data serialization standard for all programming languages. YAML stands for YAML Ain't Markup Language, and this file format technology is used in documents. These documents are saved in plain text format and are appended with the . yml extension. YAML is the file format and .yml is the file extension.
- YAML is a superset of JSON and the YAML parser understands JSON. YAML file formats are used for

configuration management because it is easy to read and comments are useful.

- The Target_image mentioned in yaml should be kept only in the target_system_image path mentioned within POAP script. Relative path is not supported for the Target_image in yaml file.
- Both .yaml and .yml extensions are supported. You have an option to choose to use any of these extensions. If you don't choose any option, the <serial>.yaml extension will be tried first and if it fails the <serial>.yml is considered.
- The MD5 files of yaml/yml is required similar to the configuration file. But if the disable_md5 is 'True' then the MD5 files of yaml/yml are not required.
- Although 'install_path' is set in the POAP script file if no yaml file for device is found, then POAP workflow will proceed with the legacy path, i.e., without any installation of RPMs, licenses and certificates.
- Install reset is highly preferred over write erase if PoAP with RPM installation is done in scenarios apart from Day-0.
- ISSU is the new default for moving to new image via PoAP. Note that you need to use "use_nxos_boot": True, if legacy boot nxos <> is required.
- The Filetype checks for .pfx,.p12 in trustpoints; .lic in license; and .rpm in rpms and aborts the current POAP if the checks/fileformats are not honoured.
- In case of .rpm, you need to provide the original file name in the yaml file.
For example: if you renamed customCliGoApp-1.0-1.7.5.x86_64.rpm to custom.rpm then PoAP will bail out indicating the name mismatch.

To get the original name of rpm:

```
bash-4.3$ rpm -qp -qf '%{NAME}-%{VERSION}-%{RELEASE}-%{ARCH}.rpm' custom.rpm
customCliGoApp-1.0-1.7.5.x86_64.rpm
bash-4.3$
```

- Once ISSU via POAP begins, abort of PoAP will be blocked. If ISSU fails for some reason, then abort capability will be re-enabled.

Guidelines and Limitations for POAP

POAP configuration guidelines and limitations are as follows:

- The bootflash:poap_retry_debugs.log is a file populated by POAP-PNP for internal purposes only. This file has no relevance in case of any POAP failures.
- Due to limitations in Syslog, securePOAP pem file name characters length is limited to 230 characters, though secure POAP supports 256 characters length for a pem file name.
- The switch software image must support POAP for this feature to function.
- POAP does not support provisioning of the switch after it has been configured and is operational. Only auto-provisioning of a switch with no startup configuration is supported.
- The https_ignore_certificate option should be turned on to use the ignore-certificate keyword with https protocol in POAP. This would enable you to successfully perform HTTPS transfer in the POAP script and without this option https as protocol cannot work with POAP.
- For those who uses HTTP/HTTPS servers for Day 0 provisioning, provisioning instructions will be given based on the MAC information and other related details in the HTTP header. POAP uses these details from HTTP GET headers so that the correct provisioning script is identified and used. This was available for other vendors (and other Cisco OSs). These additional information will be available in HTTP get headers from Cisco NX-OS Release 10.2(1) for Cisco Nexus 9000. This feature will be available by default for POAP and non-POAP HTTP

get operations.

- When you use copy http/https GET commands, the following fields are shared as part of the HTTP header:

Host: IP address

User-Agent: cisco-nxos

X-Vendor-SystemMAC: System MAC

X-Vendor-ModelName: Switch-Model

X-Vendor-Serial: Serial_Num

X-Vendor-HardwareVersion: Hardwareversion

X-Vendor-SoftwareVersion: sw_version

X-Vendor-Architecture: Architecture

- If you use POAP to bootstrap a Cisco Nexus device that is a part of a virtual port channel (vPC) pair using static port channels on the vPC links, the Cisco Nexus device activates all of its links when POAP starts up. The dually connected device at the end of the vPC links might start sending some or all of its traffic to the port-channel member links that are connected to the Cisco Nexus device, which causes traffic to get lost. To work around this issue, you can configure Link Aggregation Control Protocol (LACP) on the vPC links so that the links do not incorrectly start forwarding traffic to the Cisco Nexus device that is being bootstrapped using POAP.
- If you use POAP to bootstrap a Cisco Nexus device that is connected downstream to a Cisco Nexus 9000 Series switch through a LACP port channel, the Cisco Nexus 9000 Series switch defaults to suspend its member port if it cannot bundle it as a part of a port channel. To work around this issue, configure the Cisco Nexus 9000 Series switch to not suspend its member ports by using the no lacp suspend-individual command from interface configuration mode.
- Important POAP updates are logged in the syslog and are available from the serial console.
- Critical POAP errors are logged to the bootflash. The filename format is date-time_poap_PID_[init,1,2].log, where date-time is in the YYYYMMDD_hhmmss format and PID is the process ID.
- You can bypass the password and the basic POAP configuration by using the skip option at the POAP prompt. When you use the skip option, no password is configured for the admin user. The copy running-config startup-config command is blocked until a valid password is set for the admin user.
- If the boot poap enable command (perpetual POAP) is enabled on the switch, on a reload, a POAP boot is triggered even if there is a startup configuration present. If you do not want to use POAP in this scenario, remove the boot poap enable configuration by using the no boot poap enable command.
- Script logs are saved in the bootflash directory. The filename format is date-time_poap_PID_script.log, where date-time is in the YYYYMMDD_hhmmss format and PID is the process ID.
You can configure the format of the script log file. Script file log formats are specified in the script. The template of the script log file has a default format; however, you can choose a different format for the script execution log file.
- The POAP feature does not require a license and is enabled by default. However for the POAP feature to function, appropriate licenses must be installed on the devices in the network before the deployment of the network.
- USB support for POAP enables checking a USB device containing the configuration script file in POAP mode. This feature is supported on the Nexus 9300-EX, -FX, -FX2, -FX3, and Nexus 9200-X, -FX2 switches.
- POAP DHCP transaction may fail if the device receives high traffic rate. This issue happens when POAP uses a front panel. To avoid this issue, make sure POAP uses a management port.
- Beginning with NX-OS 7.0(3)I7(4), RFC 3004 (User Class Option for DHCP) is supported. This enables POAP

to support user-class option 77 for DHCPv4 and user-class option 15 for DHCPv6. The text displayed for the user class option for both DHCPv4 and DHCPv6 is “Cisco-POAP”.

- With RFC 3004 (User Class Option for DHCP) support, POAP over IPv6 is supported on Nexus 9000 switches.
- Beginning with NX-OS 9.2(2), POAP over IPv6 is supported on Nexus 9504 and Nexus 9508 switches with –R line cards.

The POAP over IPv6 feature enables the POAP process to use IPv6 when IPv4 fails. The feature is designed to cycle between IPv4 and IPv6 protocols when a connection failure occurs.

- For secure POAP, ensure that DHCP snooping is enabled.
- To support POAP, set firewall rules to block unintended or malicious DHCP servers.
- To maintain system security and make POAP more secure, configure the following:
 - Enable DHCP snooping.
 - Set firewall rules to block unintended or malicious DHCP servers.
- POAP is supported on both MGMT ports and in-band ports.
- Beginning with Cisco NX-OS Release 10.2(3)F, the Hardware SUDI for POAP/HTTPS feature provides an option to securely download the POAP script.
- To collect the debugging information on POAP, use the show tech-support poap command, post abort of POAP.
- Beginning with Cisco NX-OS Release 10.3(1)F, POAP is supported on Cisco Nexus X9836DM-A line card of the Cisco Nexus 9808 platform switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, POAP is supported on Cisco Nexus X98900CD-A line card of Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, POAP is supported on the Cisco Nexus 9804 platform switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.

Setting Up the Network Environment to Use POAP

Step 1

Modify the basic configuration script provided by Cisco or create your own script. For information, see the Python Scripting and API Configuration Guide.

Step 2

Every time you make a change to the configuration script, ensure that you recalculate the MD5 checksum by running `# f=poap_nexus_script.py ; cat $f | sed '/^#md5sum/d' > $f.md5 ; sed -i "s/^#md5sum=.*/#md5sum=\"$$(md5sum $f.md5 | sed 's/ .*//')\"/" $f` using a bash shell. For more information, see the Python API Reference Guide.

Step 3

(Optional) Put the POAP script and any other desired software image and switch configuration files on a USB device accessible to the switch.

Step 4

Deploy a DHCP server and configure it with the interface, gateway, and TFTP server IP addresses and a bootfile with the path and name of the configuration script file. (This information is provided to the switch when it first boots.) You do not need to deploy a DHCP server if all software image and switch configuration files are on the USB device.

Step 5

Deploy a TFTP or HTTP server to host the configuration script. In order to trigger the HTTP request to the server, prefix HTTP:// to the TFTP server name. HTTPS is not supported.

Step 6

Add the URL portion into the TFTP script name to show correct path to the file name.

Step 7

Deploy one or more servers to host the software images and configuration files.

Configuring a Switch Using POAP

Before you begin

Make sure that the network environment is set up to use POAP.

Step 1

Install the switch in the network.

Step 2

Power on the switch.

If no configuration file is found, the switch boots in POAP mode and displays a prompt that asks if you want to abort POAP and continue with a normal setup.

No entry is required to continue to boot in POAP mode.

Step 3

(Optional) If you want to exit POAP mode and enter the normal interactive setup script, enter y (yes).

The switch boots, and the POAP process begins.

What to do next

Verify the configuration.

Creating md5 Files

Every time you make a change to the configuration script, ensure that you recalculate the MD5 checksum by running `# f=poap_fabric.py ; cat $f | sed '/^#md5sum/d' > $f.md5 ; sed -i "s/^#md5sum=.*/#md5sum=\"$(md5sum $f.md5 | sed 's/ ./ /g')\"/" $f` using a bash shell.

This procedure replaces md5sum in poap_fabric.py with a new value if there was any change in that file.



Note

Steps 1-4 and 7-8 are needed only if you are using the BASH shell. If you have access to any other Linux server, these steps are not required.

Before you begin

Access to the BASH shell.

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature bash-shell Example: switch(config)# feature bash-shell	Enable BASH shell feature.
Step 3	exit Example: switch(config)# exit	Exit configuration mode.
Step 4	run bash Example: switch# run bash	Open Linux BASH.
Step 5	md5sum /bootflash/nxos.release_number.bin > /bootflash/nxos.release_number.bin.md5 Example: bash-4.2\$ md5sum /bootflash/nxos.7.0.3.l6.1.bin > /bootflash/nxos.7.0.3.l6.1.bin.md5	Creates md5sum for the .bin file.
Step 6	md5sum /bootflash/poap.cfg > /bootflash/poap.cfg.md5 Example: bash-4.2\$ md5sum /bootflash/poap.cfg > /bootflash/poap.cfg.md5	Creates md5sum for the .cfg file.
Step 7	exit Example: switch(config)# exit	Exit the BASH shell.
Step 8	dir i .md5 Example: switch# dir i .md5 65 Jun 09 12:38:48 2017 nxos.7.0.3.l6.1.bin.md5 54 Jun 09 12:39:36 2017 poap.cfg.md5 67299 Jun 09 12:48:58 2017 poap.py.md5	Display the .md5 files.
Step 9	copy bootflash:poap.cfg.md5 scp://ip_addresses/ Example: copy bootflash:poap.cfg.md5 scp://10.1.100.3/ Enter vrf (If no input, current vrf 'default' is considered): management Enter username: root root@10.1.100.3's password: poap.cfg.md5 100% 54 0.1KB/s 00:00 Copy complete.	Uploads the files to the Configuration and Software Server.

Verifying the Device Configuration

To verify the configuration, use one of the following commands:

Command	Purpose
show running-config [[exclude] command] [sanitized]	<p>Displays the contents of the currently running configuration or a subset of that configuration, use the show running-config command in the appropriate mode. :</p> <ul style="list-style-type: none"> • exclude: (Optional) Excludes a specific configuration from the display. Use the exclude keyword followed by a <i>command</i> argument to exclude a specific configuration from the display. • <i>command</i>: (Optional) Displays only a single command or a subset of commands available under a specified command mode. • sanitized: (Optional) Displays a sanitized configuration for safe distribution and analysis. <p>Beginning with Cisco NX-OS Release 10.3(2)F, sanitized keyword is supported on Cisco Nexus 9000 series switches.</p>
show startup-config	Displays the startup configuration.
show time-stamp running-config last-changed	Displays the timestamp when the running configuration was last changed.

The following example shows sample output of show running-config command with the sanitized keyword. The sanitized configuration is used to share a configuration without exposing some configuration details. This option masks the sensitive words in running configuration output with <removed> keyword.

```
switch# show running-config sanitized

!Command: show running-config sanitized
!Running configuration last done at: Wed Oct 12 09:14:54 2022
!Time: Wed Oct 12 13:52:55 2022

version 10.3(2) Bios:version 07.69

username admin password 5 <removed> role network-admin

copp profile strict
snmp-server user admin network-admin auth md5 <removed> priv aes-128 <removed> localizedV2key
rmon event 1 log trap <removed> description FATAL(1) owner PMON@FATAL
rmon event 2 log trap <removed> description CRITICAL(2) owner PMON@CRITICAL
rmon event 3 log trap <removed> description ERROR(3) owner PMON@ERROR
rmon event 4 log trap <removed> description WARNING(4) owner PMON@WARNING
rmon event 5 log trap <removed> description INFORMATION(5) owner PMON@INFO
--More--
```

Troubleshooting for POAP

The following is a list of known issues and suggestions while using POAP:

- Issue: POAP script execution fails immediately with no syslogs or output except for a “Script execution failed” statement.
Suggestion: Use the python script-name command on the server and make sure there are no syntax errors. The options dictionary is a Python dictionary so each entry must be comma separated and have the key or option and the value separated by a colon.
- Issue: A TypeError exception occurs at various places depending on the incorrectly used option.
Suggestion: Some options use integers (for example, timeouts and other numeric values). Check the options dictionary for numeric values that are enclosed in quotes. Refer to the options list for the correct usage.
- Issue: POAP over USB is not finding the files that are present.
Suggestion: Some devices have two USB slots. If you are using USB slot 2, you need to specify that as an option.

- Issue: Any issue with POAP.

Suggestion: Abort POAP, and when the system boots up, run the `show tech-support poap` command, which displays POAP status and configuration.

Managing the POAP Personality

POAP Personality

The POAP personality feature, which is introduced in Cisco NX-OS Release 7.0(3)I4(1), enables user data, Cisco NX-OS and third-party patches, and configuration files to be backed up and restored. In previous releases, POAP can restore only the configuration.

The POAP personality is defined by tracked files on the switch. The configuration and package list in the personality file are ASCII files.

Binary versions are recorded in the personality file, but the actual binary files are not included. Because binary files are typically large, they are accessed from a specified repository.

The personality file is a .tar file, which would typically be extracted into a temporary folder. Here is an example:

```
switch# dir bootflash: 042516182843personality # timestamp name
46985 Dec 06 23:12:56 2015 running-config Same as "show running-configuration" command.
20512 Dec 06 23:12:56 2015 host-package-list Package/Patches list
58056 Dec 06 23:12:56 2015 data.tar User Data
25 Dec 06 23:12:56 2015 IMAGEFILE Tracked image metadata
```

Backing Up the POAP Personality

You can create a backup of the POAP personality either locally on the switch or remotely on the server. The personality backup taken from the switch should be restored only on a switch of the same model.



Note

If you are using the Cisco scheduler feature for backups, you can configure it to also back up the POAP personality, as shown in the following example. For more information on the scheduler, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

```
switch(config)# scheduler schedule name weeklybkup
switch(config-schedule)# time weekly mon:07:00
switch(config-schedule)# job name personalitybkup
switch(config-schedule)# exit
switch(config)# scheduler job name personalitybkup
switch(config-job)# personality backup bootflash:/personality-file ; copy
bootflash:/personality-file tftp://10.1.1.1/ vrf management
```

SUMMARY STEPS

1. personality backup [bootflash:uri | scp:uri]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Required: personality backup [bootflash:uri scp:uri] Example: switch# personality backup bootflash:personality1.tar Example: switch# personality backup scp://root@2.1.1.1/var/lib/tftpboot/backup.tar	Creates a backup of the POAP personality.

Configuring the POAP Personality

You can specify whether the POAP personality should be derived from the running state of the system or the committed (startup) state.

SUMMARY STEPS

1. configure terminal
2. personality
3. track [running-state | startup-state | data local-directories-or-files]
4. binary-location source-uri-folder

DETAILED STEPS

	Command or Action	Purpose
Step 1	Required: configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	Required: personality Example: switch# personality switch(configers onality)#	Enters personality configuration mode.

Step 3	<p>Required: track [running-state start-up-state data local-directories-or-files] Example:</p> <pre>switch(config-personality)# track data bootflash:myfile1</pre> <p>Example:</p> <pre>switch(config-personality)# track data bootflash:user_scripts/*.py</pre> <p>Example:</p> <pre>switch(config-personality)# track data bootflash:basedir/*/backup_data</pre>	<p>Specifies how the POAP personality is derived. The following options are available:</p> <ul style="list-style-type: none"> • running-state—Captures the following information: the running configuration (as shown in the show running-config command), active Cisco NX-OS patches and third-party packages in the host system, and the image name (as shown in the show version command). This is the default option. • startup-state—Captures the following information: the startup configuration (as shown in the show startup-config command), committed Cisco NX-OS patches and third-party packages in the host system, and the image name (as shown in the show version command). • data local-directories-or-files—Specifies a directory or file to be backed up. You can enter this command multiple times to back up multiple directories and files. UNIX-style wildcard characters are supported. In the example, one folder and two directories are specified. <p>Note Do not use this command to backup binary files in the bootflash and do not point to the entire bootflash.</p> <p>Note Guest Shell packages are not tracked.</p> <p>Note Signed RPMs (which require a key) are not supported. The POAP personality feature does not work with signed RPMs.</p>
Step 4	<p>Required: binary-location source-uri -folder</p> <p>Example:</p> <pre>switch(config-personality)# binary-location scp://remote-dir1/nxos_patches/</pre>	<p>Specifies the local or remote directory from which to pick up binary files when the POAP personality is restored. You can enter this command multiple times (in order of priority) to specify multiple locations.</p>

Restoring the POAP Personality

During the POAP script execution phase, the personality module in the script restores the POAP personality, provided that the currently booted switch image is Cisco NX-OS Release 7.0(3)I4(1) or later. If necessary, upgrade the switch to the correct software image.



Note

A personality restore is done with the same software image used for the personality backup. Upgrading to a newer image is not supported through the POAP personality feature. To upgrade to a newer image, use the regular POAP script.



Note

If the personality script fails to execute for any reason (such as not enough space in the bootflash or a script execution failure), the POAP process returns to the DHCP discovery phase.

The restore process performs the following actions:

1. Untars and unzips the personality file in the bootflash.
2. Validates the personality file.
3. Reads the configuration and package list files from the personality file to make a list of the binaries to be downloaded.
4. If the current image or patches are not the same as specified in the personality file, downloads the binaries to the bootflash (if not present) and reboots with the correct image and then applies the packages or patches.
5. Unzips or untars the user data files relative to "/".

6. Copies the configuration file in the POAP personality to the startup configuration.
7. Reboots the switch.

POAP Personality Sample Script

The following sample POAP script (poap.py) includes the personality feature:

```
#md5sum="b00a7fffb305d13a1e02cd0d342afca3"
# The above is the (embedded) md5sum of this file taken without this line, # can be # created
  this way:
# f=poap.py ; cat $f | sed '/^#md5sum/d' > $f.md5 ; sed -i "s/^#md5sum=.*/#md5sum=$(md5sum
  $f.md5 | sed 's/ .*//')/" $f # This way this script's integrity can be checked in case you
  do not trust # tftp's ip checksum. This integrity check is done by /isan/bin/poap.bin).
# The integrity of the files downloaded later (images, config) is checked # by downloading
  the corresponding file with the .md5 extension and is # done by this script itself.

from poap.personality import POAPPersonality import os

# Location to download system image files, checksums, etc.
download_path = "/var/lib/tftpboot"
# The path to the personality tarball used for restoration personality_tarball =
"/var/lib/tftpboot/foo.tar"
# The protocol to use to download images/config protocol = "scp"
# The username to download images, the personality tarball, and the # patches and RPMs
during restoration username = "root"
# The password for the above username
password = "passwd754"
# The hostname or IP address of the file server server = "2.1.1.1"

# The VRF to use for downloading and restoration vrf = "default"
if os.environ.has_key('POAP_VRF'):
    vrf = os.environ['POAP_VRF']

# Initialize housekeeping stuff (logs, temp dirs, etc.) p = POAPPersonality(download_path,
  personality_tarball, protocol, username, password, server, vrf)

p.get_personality()
p.apply_personality()

sys.exit(0)
```

Using Network Plug and Play

This chapter contains the following sections:

- About Network Plug and Play, on page 53
- Troubleshooting Examples for Network Plug and Play, on page 61

About Network Plug and Play

Network Plug and Play (PnP) is a software application that runs on a Cisco Nexus 9500 Series Switch (specifically, N9K-C9504, N9K-C9508, and N9K-C9516). The PnP feature provides a simple, secure, unified, and integrated offering to make a new branch or campus rollouts much easier, or for provisioning updates to an existing or a new network. This feature provides a unified approach to provision networks comprising multiple devices with a near-zero-touch deployment experience.

Simplified deployment reduces the cost and complexity and increases the speed and security of the deployments. The PnP feature helps simplify the deployment of any Cisco device by automating the following deployment-related operational tasks:

- Establishing initial network connectivity for a device.
- Delivering device configuration to the controller.

- Delivering software and firmware images to the controller.
- Provisioning local credentials of a switch.
- Notifying other management systems about deployment-related events.

The PnP is a client-server based model. The client (agent) runs on a Cisco Nexus 9500 Series Switch and the server (controller) runs on the Cisco DNA Controller.

PnP uses a secure connection to communicate between the agent and the controller. This communication is encrypted.

For information on configuring and managing the needed security certificate(s) for PnP functionality, see the Cisco Digital Network Architecture Center Security Best Practices Guide.

The PnP agent converge solutions that exist in a network into a unified agent and adds additional functionality to enhance the current solutions. The main objectives of the PnP agent is to provide consistent Day 0 deployment solution for all the deployment scenarios.

Features Provided by the Network Plug and Play (PnP) Agent

Day 0 Provisioning

Day 0 bootstrapping includes the configuration, image, and other files. When a device is powered on for the first time, the PnP discovery process, which is embedded in the device, gets enabled in the absence of a startup configuration file and attempts to discover the address of the PnP controller or server. The PnP agent uses methods such as DHCP, Domain Name System (DNS), and others to acquire the desired IP address of the PnP server.

When the PnP agent successfully acquires the IP address, it initiates a long-term, bidirectional Layer 3 connection with the server and waits for a message from the server. The PnP server application sends messages to the corresponding agent, requesting for information about the devices and the services to be performed on the device.

The agent running on the Cisco Nexus 9500 Series switch then configures the IP address on receiving the DHCP acknowledgment and establishes a secure channel with the controller to provision the configurations.

The switch then upgrades the image and applies the configurations.

Discovery Methods

A PnP agent discovers the PnP controller or server using one of the following methods:

- DHCP-based discovery
- DNS-based discovery
- PnP connect

After the discovery, the PnP agent writes the discovered information into a file, which is then used to handshake with the PnP server (DNA controller/DNA-C).

The following tasks are carried out by the agent in the PnP discovery phase:

- Brings up all the interfaces.
- Sends a DHCP request in parallel for all the interfaces.
- On receiving a DHCP reply, configures the IP address and mask, default route, DNS server, domain name, and writes the PnP server IP in a lease-parsing file. Note that there is no DHCP client in Cisco Nexus Switches and static configuration is required.
- Brings down all the interfaces.



Note

POAP is the first order of choice for Day 0 provisioning. Only when there is no valid POAP offer, PnP discovery is attempted. Also, PnP is supported only on Cisco Nexus 9000 EoR models N9K-C9504, N9K-C9508, and N9K-

C9516. PnP is not supported on Cisco Nexus 9000 ToRs.

DHCP-Based Discovery

When the switch is powered on and if there is no startup configuration, the PnP starts with DHCP discovery. DHCP discovery obtains the PnP server connectivity details.

The PnP agent configures the following:

- IP address
- Netmask
- Default gateway
- DNS server
- Domain name

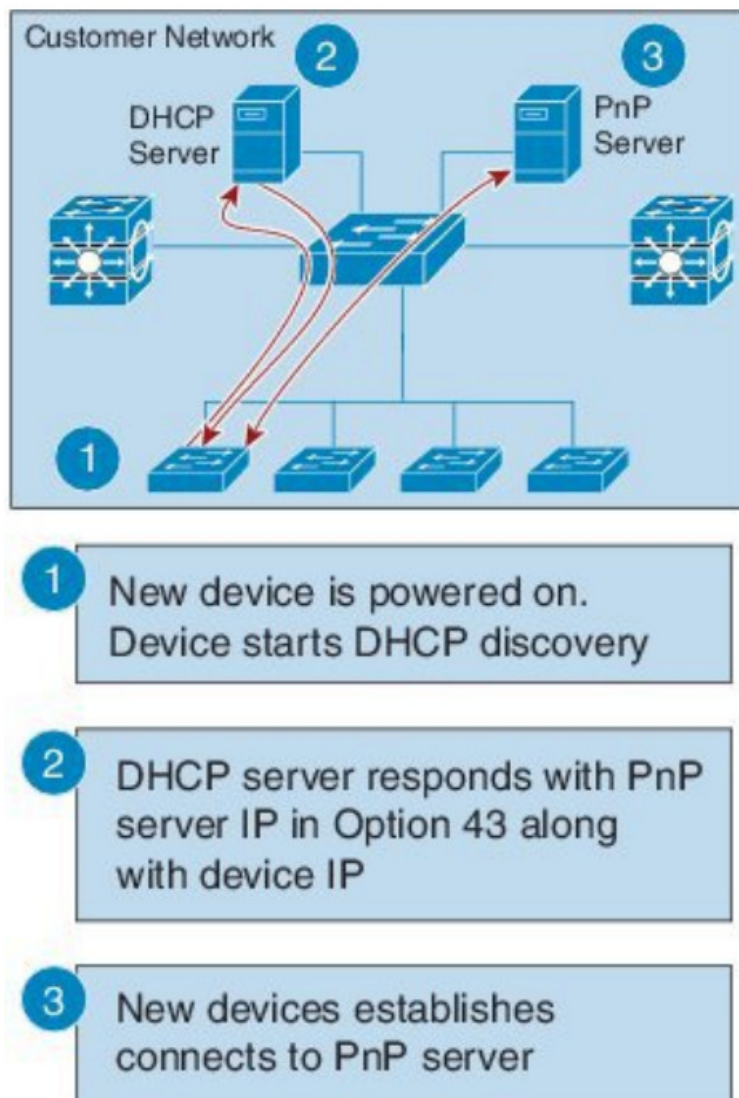
If the agent configuration fails, you should manually intervene and configure the switch.

DHCP discovery has the following flow:

- Power on the switch.
- Switch will boot up, the PnP process will be started, as there is no configuration present.
- Start DHCP discovery.
- DHCP Server replies with the PnP server configuration.
- PnP agent handshakes with the PnP server.
- Download the image, install, and reload.
- Download and apply the configuration from the controller.

A device with no startup configuration in the NV-RAM triggers the day 0 provisioning and goes through the POAP process (as detailed in [m_using_poweron_auto_provisioning_92x.ditamap#id_70221](#)). When there is no valid POAP offer, the PnP agent is initiated. The DHCP server can be configured to insert additional information using vendor-specific Option 43. Upon receiving Option 60 from the device with the string (cisco pnp), to pass on the IP address or hostname of the PnP server to the requesting device. When the DHCP response is received by the device, the PnP agent extracts the Option 43 from the response to get the IP address or the hostname of the PnP server. The PnP agent then uses this IP address or hostname to communicate with the PnP server.

Figure 8: DHCP Discovery Process for PnP Server



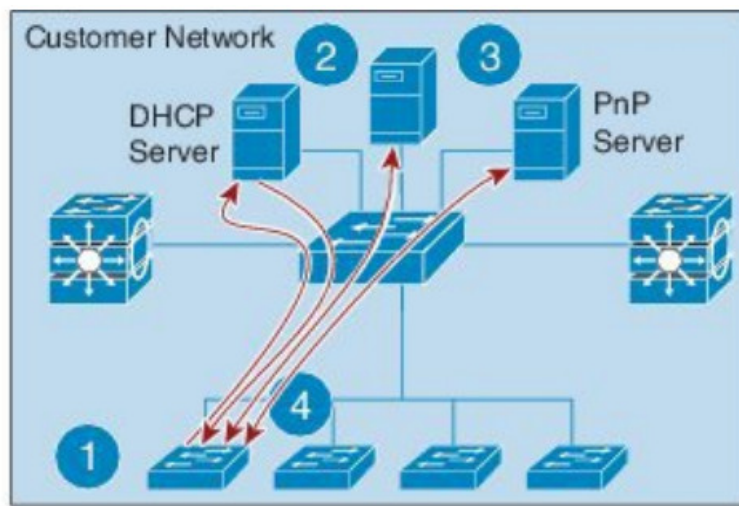
DNS-Based Discovery

When the DHCP discovery fails to get the PnP server, the agent falls back to DNS-based discovery. To start the DNS-based discovery, the following information is required from DHCP:

- IP address and netmask
- Default gateway
- DNS server IP
- Domain name

The agent obtains the domain name of the customer network from the DHCP response and constructs the fully qualified domain name (FQDN). The following FQDN is constructed by the PnP agent using a preset deployment server name and the domain name information for the DHCP response. The agent then looks up the local name server and tries to resolve the IP address for the above FQDN.

Figure 9: DNS Lookup for pnpserver.[domainname].com



- 1 New device is powered on. Device starts DHCP discovery
- 2 DHCP server responds with device IP, domain name and DNS server
- 3 Device reads domain name and creates predefined PnP server name (pnpserver.cisco.com) and resolves for IP address
- 4 New devices establishes connects to PnP server

391501

Note

The device reads domain name and creates predefined PnP server name as pnpserver.[domain name].com, for example; pnpserver.cisco.com.

Plug and Play Connect

When the DHCP and the DNS discovery fail, the PnP agent discovers and communicates with Cisco Cloud-based deployment service for initial deployment. The PnP agent directly opens an HTTPS channel using the Python library, which internally invokes OpenSSL to talk with cloud for configuration.

Cisco Power On Auto Provisioning

Cisco Power On Auto Provisioning (PoAP) communicates with the DHCP and TFTP servers to download the image and configurations. With the introduction of the PnP feature, PnP and PoAP coexist together in a Cisco Nexus 9500 Series Switch. PoAP and PnP interworking has the following processes:

- PoAP starts first when no start-up configuration is present in the system.
- PnP starts later if PoAP does not get provisioned.
- PoAP and PnP discover the controller alternatively.
- The controller discovery process continues until a controller or until the admin aborts auto provision.
- The process (POAP or PnP) that finds the controller continues provisioning and the other process that does not find the controller is notified and eventually terminated.

Services and Capabilities of the Network Plug and Play Agent

The PnP agent performs the following tasks:

- Backoff
- Capability
- CLI execution
- Configuration upgrade
- Device information
- Certificate install
- Image install
- Redirection



Note

The PnP controller or server provides an optional checksum tag to be used in the image installation and configuration upgrade service requests by the PnP agent. When the checksum is provided in a request, the image install process compares the checksum against the current running image checksum.

If the checksums are same, the image being installed or upgraded is the same as the current image running on the device. The image install process will not perform any other operation in this scenario.

If the checksums are not the same, the new image will be copied to the local file system, and the checksum will be calculated again and compared with the checksum provided in the request. If they are the same, the image install process continues to install the new image or upgrade the device to the new image. If the checksums are not the same, the process exits with an error.

Backoff

A Cisco NX-OS device that supports PnP protocol, which uses HTTP transport, requires the PnP agent to send the work request to the PnP server continuously. If the PnP server does not have any scheduled or outstanding PnP service for the PnP agent to execute, the continuous no-operation work requests exhaust both the network bandwidth and the device resources. This PnP backoff service allows the PnP server to inform the PnP agent to rest for the specified time and call back later.

Capability

Capability service request is sent by the PnP server to the PnP agent on a device to query the supported services by the agent. The server then sends an inventory service request to query the device's inventory information; and then sends an image installation request to download an image and install it. After getting the response from the agent, the list of supported PnP services and features are enlisted and returned back to the Server.

CLI Execution

Cisco NX-OS supports two modes of command execution, privileged EXEC mode and global configuration mode. Most of the EXEC commands are one-time commands, such as show commands, which show the current configuration status, and clear commands, which clear counters or interfaces. The EXEC commands are not saved when a device reboots. Configuration mode commands allow user to make changes to the running configuration. If you save the configuration, these commands are saved when a device reboots.

Configuration Upgrade

Two types of configuration upgrades takes place in a Cisco device—copying new configuration files to the startup configuration and copying new configuration files to the running configuration.

Copying new configuration files to the startup configuration—A new configuration file is copied from the file server to the device using the copy command, and the file check task is performed to check the validity of the file. If the file is valid, the file is copied to the startup configuration. The previous configuration file is backed up if enough disk space is available. The new configuration comes into effect when the device reloads again.

Copying new configuration files to the running configuration—A new configuration file is copied from the file server to the device using the copy command or configure replace command. Replace and rollback of configuration files may leave the system in an unstable state if rollback is performed inefficiently. Therefore, configuration upgrade by copying the files is preferred.

Device Information

The PnP agent provides the capability to extract device inventory and other important information to the PnP server on request. The following device-profile request types are supported:

- all—Returns complete inventory information, which includes unique device identifier (UDI), image, hardware, and file system inventory data.
- filesystem—Returns file system inventory information, which includes file system name and type, local size (in bytes), free size (in bytes), read flag, and write flag.
- hardware—Returns hardware inventory information, which includes hostname, vendor string, platform name, processor type, hardware revision, main memory size, I/O memory size, board ID, board rework ID, processor revision, mid-plane revision, and location.
- image—Returns image inventory information, which includes version string, image name, boot variable, return to ROMMON reason, bootloader variable, configuration register, configuration register on next boot, and configuration variables.
- UDI—Returns the device UDI.

Certificate Install

Certificate install is a security service through which a PnP server requests the PnP agent on a device for trust pool or trust point certificate installation or uninstallation. This service also specifies the agent about the primary and backup servers for reconnection. The following prerequisites are required for a successful certificate installation:

- The server from which the certificate or trust pool bundle needs to be downloaded should be reachable.
- There should not be any permission issues to download the certificate or the bundle.
- The PKI API should be available and accessible for the PnP agent so that the agent could call to download and install the certificate or the bundle.
- There is enough memory on the device to save the downloaded certificate or bundle.

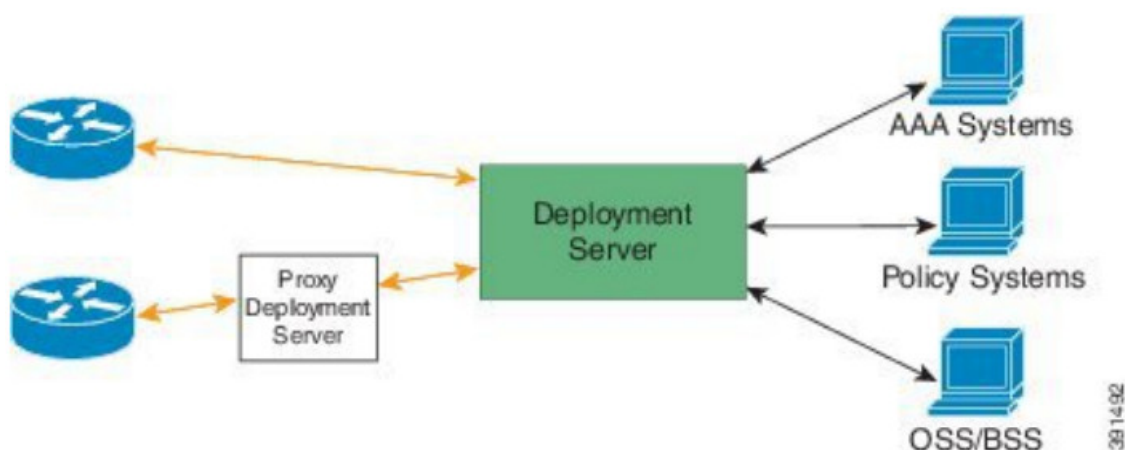
PnP Agent

The PnP agent is an embedded software component that is present in all Cisco network devices that support simplified deployment architecture. The PnP agent understands and interacts only with a PnP server. The PnP agent first tries to discover a PnP server, with which it can communicate. After a server is found and connection established, the agent performs deployment-related activities such as configuration, image and file updates by communicating with the server. It also notifies the server of all interesting deployment-related events such as out-of-band configuration changes and new device connections on an interface.

PnP Server

The PnP server is a central server that encodes the logic of managing and distributing deployment information (images, configurations, files, and licenses) for the devices being deployed. The server communicates with the agent on the device that supports the simplified deployment process using a specific deployment protocol.

Figure 10: Simplified Deployment Server



The PnP server also communicates with proxy servers such as deployment applications on smart phones and PCs, or other PnP agents acting as Neighbor Assisted Provisioning Protocol (NAPP) servers, and other types of

The PnP server can redirect the PnP agent to another deployment server. A common example of redirection is a PnP server redirecting a device to communicate with it directly after sending the bootstrap configuration through a NAPP server. A PnP server can be hosted by an enterprise. This solution allows for a cloud-based deployment service provided by Cisco. In this case, a device discovers and communicates with Cisco cloud-based deployment service for initial deployment. After that, it can be redirected to the customer's deployment server. In addition to communicating with the devices, the server interfaces with a variety of external systems such as authentication, authorizing, and accounting (AAA) systems, provisioning systems, and other management applications.

The following steps indicate the PnP agent deployment procedure on Cisco devices:

- ## PnP Agent Network Topology

The diagram illustrates a network architecture for Plug-and-Play (PnP) and Cisco Discovery Protocol (CDP). A central cloud connects to a DHCP Server, File Server, and PnP Server. A networkAdmin box (containing a laptop and phone) is connected to a PnP Agent router. The router is connected to three switches, each with multiple PnP Agents. The PnP Server communicates with the PnP Agent router via XMPP. A dashed box indicates the PnP server sends a request for neighbor information. Another dashed box indicates a device invokes CDP and responds with neighbor information/error status.

The PnP agent is enabled by default, but can be initiated on a device when the startup configuration is not available.

New Cisco devices are shipped to customers with no startup configuration file in the NVRAM of the devices. When a new device is connected to a network and powered on, the absence of a startup configuration file on the device automatically triggers the PnP agent to discover the PnP server IP address.

CLI Configuration for the PnP Agent

PnP supports devices that are using VLAN 1 by default.

Guidelines and Limitations for Network Plug and Play

Network Plug and Play (PnP) guidelines and limitations are as follows:

- Beginning with NX-OS 9.2(3), PnP is supported on the management port of Cisco Nexus 9500 platform switches.
- PnP runs on both the in-band and the management interfaces. In-band is supported only on FX-series line cards (specifically N9K-X9736C-FX for PnP).
- The PnP deployment method depends on the discovery process that is required for finding the PnP controller or server.
- The discovery mechanism must be deployed, either as a DHCP server discovery process or a Domain Name Server (DNS) discovery process, before launching PnP.
- Configure the DHCP server or the DNS server before deploying PnP.
- The PnP server must communicate with the PnP agent.
- PnP connect does not require a DHCP or DNS configuration.
- IPv6 support for PnP is not available for Cisco Nexus 9500 Series devices.

Cisco DNA Center Support

The following guidelines and limitations are specific for PnP connectivity to the Cisco DNA Center:

- Cisco DNA Center supports the following functionality on the Cisco Nexus 9504, Cisco Nexus 9508, and Cisco Nexus 9516 switches:
 - Discovery
 - Inventory
 - Topology
 - Template Programmer
 - Software Image Management
 - Basic Monitoring
- The following PnP guidelines and limitations are only for the Cisco DNA Center version 1.2.6 and earlier:
- The startup configuration that is provided during plug and play must ensure that the connectivity for the interface that is connected to the Cisco DNA Center remains intact.
- The system image .bin and startup configuration must be uploaded to the Cisco DNA Center.
- The bootflash must have enough space to download the image and configurations from the Cisco DNA Center.
[Click here for the user documentation for the Cisco DNA Center.](#)

Troubleshooting Examples for Network Plug and Play

Example: Troubleshooting PnP

The following examples shows the PnP troubleshooting command outputs:


```

Switch# show pnp status
PnP Agent is running
server-connection
    status: Success
    time: 08:41:26 Jan 11
interface-info
    status: Success
    time: 08:34:00 Jan 11
device-info
    status: Success
    time: 08:33:46 Jan 11
config-upgrade
    status: Success
    time: 08:31:36 Jan 11
capability
    status: Success
    time: 08:33:50 Jan 11
backoff
    status: Success
    time: 08:41:26 Jan 11
topology
    status: Success
    time: 08:33:54 Jan 11

```

```

Switch# show pnp version
PnP Agent Version Summary

```

```

PnP Agent: 1.6.0
Platform Name: nxos
PnP Platform: 1.5.0.rc2

```

```

Switch# show pnp profiles

```

```

Created by UDI
DHCP Discovery PID:N9K-C9504,VID:V01,SN:FOX1813GCZ8

```

```

    Primary transport: https
    Address: 10.105.194.248
    Port: 443
    CA file: /etc/pnp/certs/trustpoint/pnplabel

```

```

    Work-Request Tracking:
        Pending-WR: Correlator=
Cisco-PnP-POSIX-nxos-1.6.0-21-589a466a-0d88-427b-a17e-69afb7d0a226-1
        Last-WR: Correlator=
Cisco-PnP-POSIX-nxos-1.6.0-20-ab225de4-b0ef-46c5-9c4f-e3bd9f7c6b87-1
    PnP Response Tracking:
        Last-PR: Correlator=
Cisco-PnP-POSIX-nxos-1.6.0-20-ab225de4-b0ef-46c5-9c4f-e3bd9f7c6b87-1

```

```

Switch# show pnp lease

```

```

{
    "lease": {
        "uptime": "Fri Jan 11 05:32:17 2019",
        "intf": "Vlan1",
        "ip_addr": "10.77.143.239",
        "mask": "255.255.255.0",
        "gw": "10.77.143.1",
        "domain": "",
        "port 43": "5A1D:B2:K4:T10.105.194.248:T80"
    }
}

```

```
    "opu_id": "0112/22/11/10.100.10.10/000",  
    "lease": "3600",  
    "server": "10.77.143.231",  
    "vrf": "1"  
  }  
}
```

Switch# **show pnp internal trace**

- 1) Event:E_DEBUG, length:49, at 907122 usecs after Fri Jan 11 08:30:44 2019
[104] pnp_ascii_gen: ascii gen completed rcode[0]
- 2) Event:E_DEBUG, length:16, at 907094 usecs after Fri Jan 11 08:30:44 2019
[104] pss type: 5
- 3) Event:E_DEBUG, length:31, at 907069 usecs after Fri Jan 11 08:30:44 2019
[104] Entering pnp_ascii_gen_cfg

4) Event:E_DEBUG, length:22, at 907061 usecs after Fri Jan 11 08:30:44 2019
[104] Calling Ascii gen

5) Event:E_DEBUG, length:16, at 907051 usecs after Fri Jan 11 08:30:44 2019
[104] pss type: 2

6) Event:E_DEBUG, length:49, at 907018 usecs after Fri Jan 11 08:30:44 2019
[104] pnp_ascii_gen: fu_num_acfg_pss_entries[0x2]

7) Event:E_DEBUG, length:49, at 973813 usecs after Fri Jan 11 08:29:51 2019
[104] pnp_ascii_gen: ascii gen completed rcode[0]

8) Event:E_DEBUG, length:16, at 973787 usecs after Fri Jan 11 08:29:51 2019
[104] pss type: 5

9) Event:E_DEBUG, length:31, at 973760 usecs after Fri Jan 11 08:29:51 2019
[104] Entering pnp_ascii_gen_cfg

10) Event:E_DEBUG, length:22, at 973751 usecs after Fri Jan 11 08:29:51 2019
[104] Calling Ascii gen

11) Event:E_DEBUG, length:16, at 973742 usecs after Fri Jan 11 08:29:51 2019
[104] pss type: 2

12) Event:E_DEBUG, length:49, at 973707 usecs after Fri Jan 11 08:29:51 2019
[104] pnp_ascii_gen: fu_num_acfg_pss_entries[0x2]

13) Event:E_DEBUG, length:35, at 535794 usecs after Fri Jan 11 08:04:15 2019
[104] pnp_pi_spawn_finalize pid 690

14) Event:E_DEBUG, length:41, at 228291 usecs after Fri Jan 11 08:04:13 2019
[104] + pnp_pi_spawn child_pid: 0xdd526da0

15) Event:E_DEBUG, length:76, at 132853 usecs after Fri Jan 11 08:03:26 2019
[104] Rx: Direction: PnP PI -> PnP PD, Type: Device Provisioned, Cfg: Present

16) Event:E_DEBUG, length:35, at 440380 usecs after Fri Jan 11 08:03:18 2019
[104] !!! ACKED Unconfigure Ret:1!!!

17) Event:E_DEBUG, length:61, at 440347 usecs after Fri Jan 11 08:03:18 2019
[104] Tx: Direction: Max, Type: DHCP Unconfigure Done, Len: 16

18) Event:E_DEBUG, length:35, at 440331 usecs after Fri Jan 11 08:03:18 2019
[102] Unknown timer cancel requested

19) Event:E_DEBUG, length:35, at 440311 usecs after Fri Jan 11 08:03:18 2019
[104] pnp_pss_runtime_commit success

20) Event:E_DEBUG, length:57, at 440103 usecs after Fri Jan 11 08:03:18 2019
[104] pnp_pss_runtime_commit: Stored values in runtime PSS

21) Event:E_DEBUG, length:23, at 440051 usecs after Fri Jan 11 08:03:18 2019
[104] - pnp_vsh_halt:206

22) Event:E_DEBUG, length:17, at 950291 usecs after Fri Jan 11 08:03:15 2019
[104] Adding "end"

23) Event:E_DEBUG, length:58, at 950269 usecs after Fri Jan 11 08:03:15 2019
[104] Adding "configure terminal ; no clock protocol none "

24) Event:E_DEBUG, length:33, at 945994 usecs after Fri Jan 11 08:03:15 2019
[104] - pnp_vsh_config_l3_intf:788

25) Event:E_DEBUG, length:29, at 945979 usecs after Fri Jan 11 08:03:15 2019
[104] + pnp_vsh_config_l3_intf

26) Event:E_DEBUG, length:39, at 945963 usecs after Fri Jan 11 08:03:15 2019
[104] Adding "no feature interface-vlan"

27) Event:E_DEBUG, length:32, at 945932 usecs after Fri Jan 11 08:03:15 2019
[104] Adding "configure terminal"

28) Event:E_DEBUG, length:40, at 945886 usecs after Fri Jan 11 08:03:15 2019
[104] Got Semaphore, vsh halt continue...

29) Event:E_DEBUG, length:46, at 945870 usecs after Fri Jan 11 08:03:15 2019
[104] sem_timedwait Success, Start VSH clean up

30) Event:E_DEBUG, length:19, at 945843 usecs after Fri Jan 11 08:03:15 2019
[104] + pnp_vsh_halt

31) Event:E_DEBUG, length:35, at 945831 usecs after Fri Jan 11 08:03:15 2019
[104] pnp_pss_runtime_commit success

32) Event:E_DEBUG, length:57, at 945643 usecs after Fri Jan 11 08:03:15 2019
[104] pnp_pss_runtime_commit: Stored values in runtime PSS

33) Event:E_DEBUG, length:33, at 945607 usecs after Fri Jan 11 08:03:15 2019
[104] !!! Received Unconfigure !!!

34) Event:E_DEBUG, length:74, at 945578 usecs after Fri Jan 11 08:03:15 2019
[104] Rx: Direction: PnP PI -> PnP PD, Type: DHCP Unconfigure, Cfg: Present

35) Event:E_DEBUG, length:49, at 789616 usecs after Fri Jan 11 08:01:52 2019
[104] pnp_ascii_gen: ascii gen completed rcode[0]

36) Event:E_DEBUG, length:16, at 789579 usecs after Fri Jan 11 08:01:52 2019
[104] pss type: 5

37) Event:E_DEBUG, length:31, at 789522 usecs after Fri Jan 11 08:01:52 2019
[104] Entering pnp_ascii_gen_cfg

38) Event:E_DEBUG, length:22, at 789514 usecs after Fri Jan 11 08:01:52 2019
[104] Calling Ascii gen

39) Event:E_DEBUG, length:16, at 789506 usecs after Fri Jan 11 08:01:52 2019
[104] pss type: 2

40) Event:E_DEBUG, length:49, at 789489 usecs after Fri Jan 11 08:01:52 2019
[104] pnp_ascii_gen: fu_num_acfg_pss_entries[0x2]

41) Event:E_DEBUG, length:35, at 789365 usecs after Fri Jan 11 08:01:52 2019
[104] pnp_pss_runtime_commit success

42) Event:E_DEBUG, length:57, at 789135 usecs after Fri Jan 11 08:01:52 2019
[104] pnp_pss_runtime_commit: Stored values in runtime PSS

43) Event:E_DEBUG, length:26, at 789096 usecs after Fri Jan 11 08:01:52 2019
[104] Phase Init -> Monitor

44) Event:E_DEBUG, length:35, at 788967 usecs after Fri Jan 11 08:01:52 2019
[104] pnp_pi_spawn_finalize pid 1c9

45) Event:E_DEBUG, length:41, at 831561 usecs after Fri Jan 11 08:01:49 2019
[104] + pnp_pi_spawn child_pid: 0xffff7e28

46) Event:E_DEBUG, length:45, at 831550 usecs after Fri Jan 11 08:01:49 2019

```
[104] Have startup config, Starting PnP PI....

47) Event:E_DEBUG, length:40, at 831538 usecs after Fri Jan 11 08:01:49 2019
[104] Posix log directory creation failed

48) Event:E_DEBUG, length:50, at 831479 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_fire_event: PNP_EVENT_HAVE_STARTUP_CONFIG

49) Event:E_DEBUG, length:35, at 831465 usecs after Fri Jan 11 08:01:49 2019
[104] Inside : pnp_other_msg_handler

50) Event:E_DEBUG, length:80, at 831437 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_get_data_from_queue: dequeued event 0x1102e0cc 25/cat 11 from pending Q

51) Event:E_DEBUG, length:50, at 831368 usecs after Fri Jan 11 08:01:49 2019
[104] Injecting Event PNP_EVENT_HAVE_STARTUP_CONFIG

52) Event:E_DEBUG, length:59, at 831303 usecs after Fri Jan 11 08:01:49 2019
[104] Have Startup Config, move the process state to monitor

53) Event:E_DEBUG, length:57, at 799379 usecs after Fri Jan 11 08:01:49 2019
[104] Accelerating PnP, Break Point: Break Point PoAP Init

54) Event:E_DEBUG, length:35, at 799334 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit success

55) Event:E_DEBUG, length:57, at 799239 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit: Stored values in runtime PSS

56) Event:E_DEBUG, length:23, at 799226 usecs after Fri Jan 11 08:01:49 2019
[104] Phase None -> Init

57) Event:E_DEBUG, length:53, at 799200 usecs after Fri Jan 11 08:01:49 2019
[104] Initilizing PnP-agent State machine curr_state 3

58) Event:E_DEBUG, length:35, at 799188 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit success

59) Event:E_DEBUG, length:57, at 799070 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit: Stored values in runtime PSS

60) Event:E_DEBUG, length:26, at 798965 usecs after Fri Jan 11 08:01:49 2019
[104] !!! Box is Online !!!

61) Event:E_DEBUG, length:35, at 798954 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit success

62) Event:E_DEBUG, length:57, at 798770 usecs after Fri Jan 11 08:01:49 2019
[104] pnp_pss_runtime_commit: Stored values in runtime PSS

63) Event:E_DEBUG, length:70, at 370297 usecs after Fri Jan 11 07:55:41 2019
[102] pnp_demux_mts(463): (Warning) unexpected mts msg (opcode - 7655)

64) Event:E_DEBUG, length:41, at 092701 usecs after Fri Jan 11 07:55:33 2019
[104] PnP Init Internal subsystem, Done!!!

65) Event:E_DEBUG, length:32, at 089920 usecs after Fri Jan 11 07:55:33 2019
[104] PnP Init Internal subsystem
```

```
Switch# show pnp posix_pi configs
```

```
/isan/etc/pnp/platform_config.cfg:
```

```
/isan/etc/pnp/file_paths.cfg:
```



```
/isan/etc/pnp/pnp_config.cfg:  
  
/isan/etc/pnp/policy_discovery.conf:  
  
/isan/etc/pnp/certs/platform.json:  
  
/isan/etc/pnp/certs/pnp_status.json:  
  
/isan/etc/pnp/certs/job_status.json:
```

Understanding the Command-Line Interface

This chapter contains the following sections:

- About the CLI Prompt, on page 67
- Command Modes, on page 68
- Special Characters, on page 72
- Keystroke Shortcuts, on page 72
- Abbreviating Commands, on page 75
- Completing a Partial Command Name, on page 75
- Identifying Your Location in the Command Hierarchy, on page 76
- Using the no Form of a Command, on page 76
- Configuring CLI Variables, on page 77
- Command Aliases, on page 79
- Command Scripts, on page 81
- Context-Sensitive Help, on page 83
- Understanding Regular Expressions, on page 84
- Searching and Filtering show Command Output, on page 86
- Searching and Filtering from the –More– Prompt, on page 90
- Using the Command History, on page 91
- Enabling or Disabling the CLI Confirmation Prompts, on page 93
- Setting CLI Display Colors, on page 94
- Sending Commands to Modules, on page 94
- Sending Command Output in Email, on page 95
- BIOS Loader Prompt, on page 97
- Examples Using the CLI, on page 97

About the CLI Prompt

Once you have successfully accessed the device, the CLI prompt displays in the terminal window of your console port or remote workstation as shown in the following example:

```
User Access Verification  
login: admin  
Password:<password>  
Cisco Nexus Operating System (NX-OS) Software  
TAC support: http://www.cisco.com/tac
```

Copyright (c) 2002-2013, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
<http://www.opensource.org/licenses/gpl-2.0.php> and
<http://www.opensource.org/licenses/lgpl-2.1.php>
switch#

You can change the default device hostname.
From the CLI prompt, you can do the following:

- Use CLI commands for configuring features
- Access the command history
- Use command parsing functions



Note

In normal operation, usernames are case sensitive. However, when you are connected to the device through its console port, you can enter a login username in all uppercase letters regardless of how the username was defined. As long as you provide the correct password, the device logs you in.

Command Modes

This section describes command modes in the Cisco NX-OS CLI.

EXEC Command Mode

When you first log in, the Cisco NX-OS software places you in EXEC mode. The commands available in EXEC mode include the show commands that display the device status and configuration information, the clear commands, and other commands that perform actions that you do not save in the device configuration.

Global Configuration Command Mode

Global configuration mode provides access to the broadest range of commands. The term indicates characteristics or features that affect the device as a whole. You can enter commands in global configuration mode to configure your device globally or to enter more specific configuration modes to configure specific elements such as interfaces or protocols.

SUMMARY STEPS

1. configure terminal

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode. Note The CLI prompt changes to indicate that you are in global configuration mode.

Interface Configuration Command Mode

One example of a specific configuration mode that you enter from global configuration mode is interface configuration mode. To configure interfaces on your device, you must specify the interface and enter interface configuration mode.

You must enable many features on a per-interface basis. Interface configuration commands modify the operation of the interfaces on the device, such as Ethernet interfaces or management interfaces (mgmt 0).

For more information about configuring interfaces, see the Cisco Nexus 9000 Series NX-OS Interfaces

SUMMARY STEPS

1. configure terminal
2. interface type number

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>type number</i> Example: switch(config)# interface ethernet 2/2 switch(config-if)#	Specifies the interface that you want to configure. The CLI places you into interface configuration mode for the specified interface. Note The CLI prompt changes to indicate that you are in interface configuration mode.

Subinterface Configuration Command Mode

From global configuration mode, you can access a configuration submode for configuring VLAN interfaces called subinterfaces. In subinterface configuration mode, you can configure multiple virtual interfaces on a single physical interface. Subinterfaces appear to a protocol as distinct physical interfaces.

Subinterfaces also allow multiple encapsulations for a protocol on a single interface. For example, you can configure IEEE 802.1Q encapsulation to associate a subinterface with a VLAN.

For more information about configuring subinterfaces, see the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.

SUMMARY STEPS

1. configure terminal
2. interface type number.subint

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>type number.subint</i> Example: switch(config)# interface ethernet 2/2.1 switch(config-subif)#	Specifies the VLAN interface to be configured. The CLI places you into a subinterface configuration mode for the specified VLAN interface. Note The CLI prompt changes to indicate that you are in subinterface configuration mode.

Saving and Restoring a Command Mode

The Cisco NX-OS software allows you to save the current command mode, configure a feature, and then restore

the previous command mode. The push command saves the command mode, and the pop command restores the command mode.

The following example shows how to save and restore a command mode:

```
switch# configure terminal
switch(config)# event manager applet test
switch(config-applet)# push
switch(config-applet)# configure terminal
switch(config)# username testuser password newtest
switch(config)# pop
switch(config-applet)#
```

Exiting a Configuration Command Mode

SUMMARY STEPS

1. exit
2. end
3. (Optional) Ctrl-Z

DETAILED STEPS

	Command or Action	Purpose
Step 1	exit Example: switch(config-if)# exit switch(config)#	Exits from the current configuration command mode and returns to the previous configuration command mode.
Step 2	end Example: switch(config-if)# end switch#	Exits from the current configuration command mode and returns to EXEC mode.
Step 3	(Optional) Ctrl-Z Example: switch(config-if)# ^Z switch#	Exits the current configuration command mode and returns to EXEC mode. Caution If you press Ctrl-Z at the end of a command line in which a valid command has been typed, the CLI adds the command to the running configuration file. In most cases, you should exit a configuration mode using the exit or end command.

Command Mode Summary

This table summarizes information about the main command modes.

Table 4: Command Mode Summary

Mode	Access Method	Prompt	Exit Method
EXEC	From the login prompt, enter your username and password.	switch#	To exit to the login prompt, use the exit command.
Global configuration	From EXEC mode, use the configure terminal command.	switch(config)#	To exit to EXEC mode, use the end or exit command or press Ctrl-Z .
Interface configuration	From global configuration mode, specify an interface with an interface command.	switch(config-if)#	To exit to global configuration mode, use the exit command. To exit to EXEC mode, use the exit command or press Ctrl-Z .
Subinterface configuration	From global configuration mode, specify a subinterface with an interface command.	switch(config-subif)#	To exit to global configuration mode, use the exit command. To exit to EXEC mode, use the end command or press Ctrl-Z .
VRF configuration	From global configuration mode, use the vrf command and specify a routing protocol.	switch(config-vrf)#	To exit to global configuration mode, use the exit command. To exit to EXEC mode, use the end command or press Ctrl-Z .
EXEC for a non default VRF	From EXEC mode, use the routing-context vrf command and specify a VRF.	switch-red#	To exit to the default VRF, use the routing-context vrf default command.

Special Characters

This table lists the characters that have special meaning in Cisco NX-OS text strings and should be used only in regular expressions or other special contexts.

Table 5: Special Characters

Character	Description
%	Percent
#	Pound, hash, or number
...	Ellipsis
	Vertical bar
< >	Less than or greater than
[]	Brackets
{ }	Braces

Keystroke Shortcuts

This table lists command key combinations that can be used in both EXEC and configuration modes.

Table 6: Keystroke Shortcuts

Keystrokes	Description
Ctrl-A	Moves the cursor to the beginning of the line.

Ctrl-B	Moves the cursor one character to the left. When you enter a command that extends beyond a single line, you can press the Left Arrow or Ctrl-B keys repeatedly to scroll back toward the system prompt and verify the beginning of the command entry, or you can press the Ctrl-A key combination.
Ctrl-C	Cancels the command and returns to the command prompt.
Ctrl-D	Deletes the character at the cursor.
Ctrl-E	Moves the cursor to the end of the line.
Ctrl-F	Moves the cursor one character to the right.
Ctrl-G	Exits to the previous command mode without removing the command string.
Ctrl-K	Deletes all characters from the cursor to the end of the command line.
Ctrl-L	Redisplays the current command line.
Ctrl-N	Displays the next command in the command history.
Ctrl-O	Clears the terminal screen.
Ctrl-P	Displays the previous command in the command history.
Ctrl-R	Redisplays the current command line.
Ctrl-T	Transposes the character under the cursor with the character located to the right of the cursor. The cursor is then moved to the right one character.
Ctrl-U	Deletes all characters from the cursor to the beginning of the command line.
Ctrl-V	Removes any special meaning for the following keystroke. For example, press Ctrl-V before entering a question mark (?) in a regular expression.
Ctrl-W	Deletes the word to the left of the cursor.
Ctrl-X, H	Lists the history of commands you have entered. When using this key combination, press and release the Ctrl and X keys together before pressing H.
Ctrl-Y	Recalls the most recent entry in the buffer (press keys simultaneously).
Ctrl-Z	Ends a configuration session, and returns you to EXEC mode. When used at the end of a command line in which a valid command has been typed, the resulting configuration is first added to the running configuration file.
Up arrow key	Displays the previous command in the command history.
Down arrow key	Displays the next command in the command history.
Right arrow key Left arrow key	Moves your cursor through the command string, either forward or backward, allowing you to edit the current command.
?	Displays a list of available commands.

Tab	<p>Completes the word for you after you enter the first characters of the word and then press the Tab key. All options that match are presented.</p> <p>Use tabs to complete the following items:</p> <ul style="list-style-type: none"> • Command names • Scheme names in the file system • Server names in the file system • Filenames in the file system <p>Example:</p> <pre>switch(config)# xm<Tab> switch(config)# xml<Tab> switch(config)# xml server</pre>
	<p>Example:</p> <pre>switch(config)# c<Tab> callhome class-map clock cdp cli control-plane switch(config)# cl<Tab> class-map cli clock switch(config)# cla<Tab> switch(config)# class-map</pre>
	<p>Example:</p> <pre>switch# cd bootflash:<Tab> bootflash:/// bootflash://sup-1/ bootflash://sup-active/ bootflash://s up-local/ bootflash://module-27/ bootflash://module-28/</pre>
	<p>Example:</p> <pre>switch# cd bootflash://mo<Tab> bootflash://module-27/ bootflash://module-28/ switch# cd boo tflash://module-2</pre> <p>Note You cannot access remote machines using the cd command. If you are on slot 27 and enter the cd bootflash://module-28 command, the following message appears: "Changing directory to a non-local server is not allowed."</p>

Abbreviating Commands

You can abbreviate commands and keywords by entering the first few characters of a command. The abbreviation must include sufficient characters to make it unique from other commands or keywords. If you are having trouble entering a command, check the system prompt and enter the question mark (?) for a list of available commands. You might be in the wrong command mode or using incorrect syntax.

This table lists examples of command abbreviations.

Table 7: Examples of Command Abbreviations

Command	Abbreviation
configure terminal	conf t
copy running-config startup-config	copy run start
interface ethernet 1/2	int e 1/2
show running-config	sh run

Completing a Partial Command Name

If you cannot remember a complete command name or if you want to reduce the amount of typing you have to perform, enter the first few letters of the command, and then press the Tab key. The command line parser will complete the command if the string entered is unique to the command mode. If your keyboard does not have a Tab key, press Ctrl-I instead.

The CLI recognizes a command once you have entered enough characters to make the command unique. For example, if you enter conf in EXEC mode, the CLI will be able to associate your entry with the configure command, because only the configure command begins with conf.

In the following example, the CLI recognizes the unique string for conf in EXEC mode when you press the Tab key:

```
switch# conf<Tab>
switch# configure
```

When you use the command completion feature, the CLI displays the full command name. The CLI does not

execute the command until you press the Return or Enter key. This feature allows you to modify the command if the full command was not what you intended by the abbreviation. If you enter a set of characters that could indicate more than one command, a list of matching commands displays.

For example, entering co<Tab> lists all commands available in EXEC mode beginning with co:

```
switch# co<Tab>
configure copy
switch# co
```

Note that the characters you entered appear at the prompt again to allow you to complete the command entry.

Identifying Your Location in the Command Hierarchy

Some features have a configuration submode hierarchy nested more than one level. In these cases, you can display information about your present working context (PWC).

SUMMARY STEPS

- 1. where detail

DETAILED STEPS

	Command or Action	Purpose
Step 1	where detail Example: switch# configure terminal switch(config)# interface mgmt0 switch(config-if)# where detail mode: conf interface mgmt0 username: admin routing-context vrf: default	Displays the PWC.

Using the no Form of a Command

Almost every configuration command has a no form that can be used to disable a feature, revert to a default value, or remove a configuration.

This example shows how to disable a feature:

```
switch# configure terminal
switch(config)# feature tacacs+
switch(config)# no feature tacacs+
```

This example shows how to revert to the default value for a feature:

```
switch# configure terminal
switch(config)# banner motd #Welcome to the switch#
switch(config)# show banner motd
Welcome to the switch
switch(config)# no banner motd
switch(config)# show banner motd
User Access Verification
```

This example shows how to remove the configuration for a feature:

```
switch# configure terminal
switch(config)# radius-server host 10.10.2.2
switch(config)# show radius-server
retransmission count:0
timeout value:1
```

```

deadtime value:1
total number of servers:1

following RADIUS servers are configured:
  10.10.1.1:
    available for authentication on port:1812
    available for accounting on port:1813
  10.10.2.2:
    available for authentication on port:1812
    available for accounting on port:1813

switch(config)# no radius-server host 10.10.2.2
switch(config)# show radius-server
retransmission count:0
timeout value:1
deadtime value:1
total number of servers:1

following RADIUS servers are configured:
  10.10.1.1:
    available for authentication on port:1812
    available for accounting on port:1813

```

This example shows how to use the **no** form of a command in EXEC mode:

```

switch# cli var name testinterface ethernet1/2
switch# show cli variables
SWITCHNAME="switch"
TIMESTAMP="2013-05-12-13.43.13"
testinterface="ethernet1/2"

switch# cli no var name testinterface
switch# show cli variables
SWITCHNAME="switch"
TIMESTAMP="2013-05-12-13.43.13"

```

Configuring CLI Variables

This section describes CLI variables in the Cisco NX-OS CLI.

About CLI Variables

The Cisco NX-OS software supports the definition and use of variables in CLI commands.

You can refer to CLI variables in the following ways:

- Entered directly on the command line.
- Passed to a script initiated using the run-script command. The variables defined in the parent shell are available for use in the child run-script command process.

CLI variables have the following characteristics:

- Cannot have nested references through another variable
- Can persist across switch reloads or exist only for the current session

Cisco NX-OS supports one predefined variable: **TIMESTAMP**. This variable refers to the current time when the command executes in the format YYYY-MM-DD-HH.MM.SS.



Note

The **TIMESTAMP** variable name is case sensitive. All letters must be uppercase.

Configuring CLI Session-Only Variables

You can define CLI session variables to persist only for the duration of your CLI session. These variables are

useful for scripts that you execute periodically. You can reference the variable by enclosing the name in parentheses and preceding it with a dollar sign (\$), for example \$(variable-name).

SUMMARY STEPS

1. cli var name variable-name variable-text
2. (Optional) show cli variables

DETAILED STEPS

	Command or Action	Purpose
Step 1	cli var name <i>variable-name</i> <i>variable-text</i> Example: switch# cli var name testinterface ethernet 2/1	Configures the CLI session variable. The <i>variable-name</i> argument is alphanumeric, case sensitive, and has a maximum length of 31 characters. The <i>variable-text</i> argument is alphanumeric, case sensitive, can contain spaces, and has a maximum length of 200 characters. Note Beginning with Cisco NX-OS Release 7.0(3)I4(1), variables can include hyphens (-) and underscores (_).
Step 2	(Optional) show cli variables Example: switch# show cli variables	Displays the CLI variable configuration.

Configuring Persistent CLI Variables

You can configure CLI variables that persist across CLI sessions and device reloads.

SUMMARY STEPS

1. configure terminal
2. cli var name variable-name variable-text
3. exit
4. (Optional) show cli variables
5. (Optional) copy running-config startup-config

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	cli var name <i>variable-name variable-text</i> Example: switch(config)# cli var name testinterface ethernet 2/1	Configures the CLI persistent variable. The variable name is a case-sensitive, alphanumeric string and must begin with an alphabetic character. The maximum length is 31 characters. Note Beginning with Cisco NX-OS Release 7.0(3)I4(1), variables can include hyphens (-) and underscores (_).
Step 3	exit Example: switch(config)# exit switch#	Exits global configuration mode.
Step 4	(Optional) show cli variables Example: switch# show cli variables	Displays the CLI variable configuration.
Step 5	(Optional) copy running-config startup-config Example: switch(config)# copy running-config startup-config	Copies the running configuration to the startup configuration.

Command Aliases

This section provides information about command aliases.

About Command Aliases

You can define command aliases to replace frequently used commands. The command aliases can represent all or part of the command syntax.

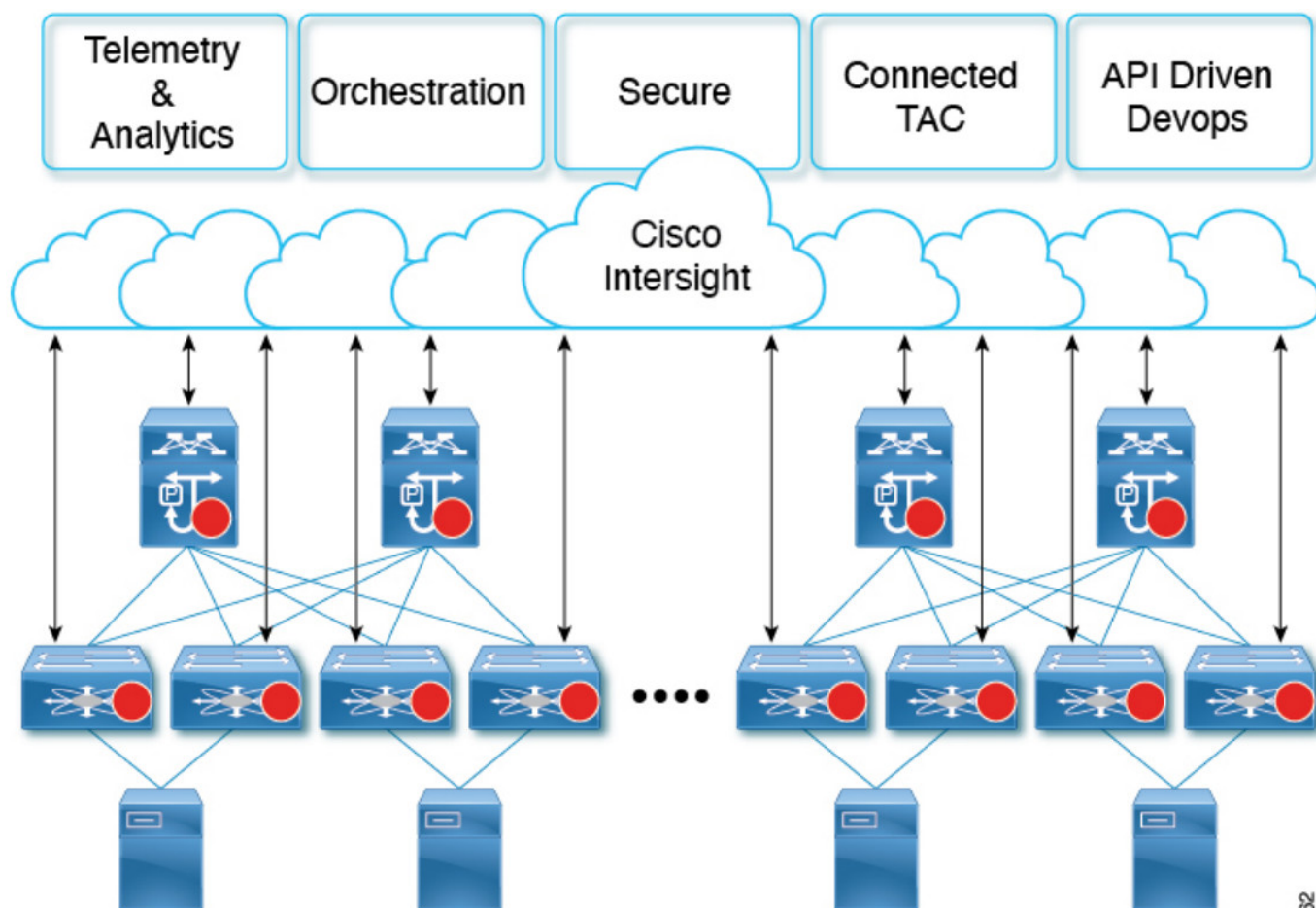
Command alias support has the following characteristics:

- Command aliases are global for all user sessions.
- Command aliases persist across reboots if you save them to the startup configuration.
- Command alias translation always takes precedence over any keyword in any configuration mode or submode.
- Command alias configuration takes effect for other user sessions immediately.



=





● Nexus Device Intersight connector


Example:

```
payload={
  "ins_api": {
    "version": "1.0",
    "type": "bash",
    "chunk": "0",
    "sid": "sid",
    "input": "ip netns exec management curl http://localhost:8889/HttpProxies",
    "output_format": "json"
  }
}
```

Result:

```
{
  "ins_api": {
    "version": "1.0",
    "sid": "eoc",
    "type": "bash",
    "outputs": {
      "output": {
        "body": "[\n {\n   \"ProxyHost\": \"\", \n   \"ProxyPort\": 0, \n   \"Preference\": 0, \n   \"ProxyType\": \"Disabled\" \n } \n]",
        "code": "200",
        "msg": "Success"
      }
    }
  }
}
```

504162

 <p>Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide, Release 10.4</p>	<p>CISCO Nexus 9000 Series NX-OS Fundamentals Configuration Guide Release 10.4 [pdf] User Guide</p> <p>Nexus 9000 Series NX-OS Fundamentals Configuration Guide Release 10.4, Nexus 9000 Series, NX-OS Fundamentals Configuration Guide Release 10.4, Configuration Guide Release 10.4, Guide Release 10.4</p>
---	--

References

- [User Manual](#)