**Cisco Partner**

**Nexus 9000**

**Demo Guide**

Version 1.5

**ATTENTION**

The information contained in this guide is for demonstration purposes only.This guide contains information and activities that, while beneficial for purposes of demonstration in close, non-productive environment, can result in downtime or other severe consequences and, therefore, are not intended as a reference guide.

No part of this document may be reproduced in any form or by any means (graphic, electronic, mechanical, including photocopying, recording or taping, or storage in an electronic retrieval system).

Contents

[Intro 6](#_Toc427756186)

[Getting Started 7](#_Toc427756187)

[Prerequisites 7](#_Toc427756188)

[Demo Topology 7](#_Toc427756189)

[VXLAN Setup 8](#_Toc427756190)

[Estimated Time 8](#_Toc427756191)

[Requirements 8](#_Toc427756192)

[Steps 8](#_Toc427756193)

[Downloading VXLAN Configurations 8](#_Toc427756194)

[Verification Commands 8](#_Toc427756195)

[Additional Notes 8](#_Toc427756196)

[Getting Python Scripts onto Nexus 9000 Switches 10](#_Toc427756197)

[Estimated Time 10](#_Toc427756198)

[Steps 10](#_Toc427756199)

[Programmability Demo – Finding Unused IP Addresses 11](#_Toc427756200)

[Estimated Time 11](#_Toc427756201)

[Preparation 11](#_Toc427756202)

[Steps 11](#_Toc427756203)

[Programmability Demo – Super Commands Demo 12](#_Toc427756204)

[Estimated Time 12](#_Toc427756205)

[Steps 12](#_Toc427756206)

[Additional Note 12](#_Toc427756207)

[Programmability Demo – CDP-to-Interface Description Demo 13](#_Toc427756208)

[Estimated Time 13](#_Toc427756209)

[Requirement 13](#_Toc427756210)

[Steps 13](#_Toc427756211)

[Programmability Demo – LLDP-to-Interface Description Demo 14](#_Toc427756212)

[Estimated Time 14](#_Toc427756213)

[Requirement 14](#_Toc427756214)

[Steps 14](#_Toc427756215)

[Programmability Demo – OSPF Demo 15](#_Toc427756216)

[Estimated Time 15](#_Toc427756217)

[Requirements 15](#_Toc427756218)

[Steps 15](#_Toc427756219)

[Programmability Demo – PIM Demo 16](#_Toc427756220)

[Estimated Time 16](#_Toc427756221)

[Requirements 16](#_Toc427756222)

[Steps 16](#_Toc427756223)

[Automation Demo – Generating CLI VXLAN Configurations for 2-Spine, 2-Leaf Topology 17](#_Toc427756224)

[Estimated Time 17](#_Toc427756225)

[Steps 17](#_Toc427756226)

[Automation Demo – NX-API Sandbox Demo 18](#_Toc427756227)

[Estimated Time 18](#_Toc427756228)

[Requirements 18](#_Toc427756229)

[Steps 18](#_Toc427756230)

[Automation Demo – NXOS Version Compliance Demo 19](#_Toc427756231)

[Estimated Time 19](#_Toc427756232)

[Requirement 19](#_Toc427756233)

[Steps 19](#_Toc427756234)

[Additional Notes 19](#_Toc427756235)

[Troubleshooting Demo – CPU Monitoring Demo 20](#_Toc427756236)

[Estimated Time 20](#_Toc427756237)

[Steps 20](#_Toc427756238)

[Additional Note 20](#_Toc427756239)

[Troubleshooting Demo – Memory Monitoring Demo 21](#_Toc427756240)

[Estimated Time 21](#_Toc427756241)

[Steps 21](#_Toc427756242)

[Additional Note 21](#_Toc427756243)

[Troubleshooting Demo – Troubleshooting Traffic Loss Demo 22](#_Toc427756244)

[Estimated Time 22](#_Toc427756245)

[Steps 22](#_Toc427756246)

[Additional Note 22](#_Toc427756247)

# Getting Started

This guide is for Cisco Partners who are looking for use cases that demonstrate the rich capabilities of the Nexus 9000 running NX-OS Mode. The intent is for Partners to take some or all of the demos described in this document and showcase to their customers. While the initial focus is on programmability and automation, new capabilities and content will be added based on Partner/customer feedback and needs.

Note that the number of features shown and the order in which they are given is arbitrary. Partners may pick and choose as appropriate for their specific needs. We encourage Partners to build on this material or contribute custom content and share with the broader Partner community.

## Prerequisites

3 Nexus 9000 Switches

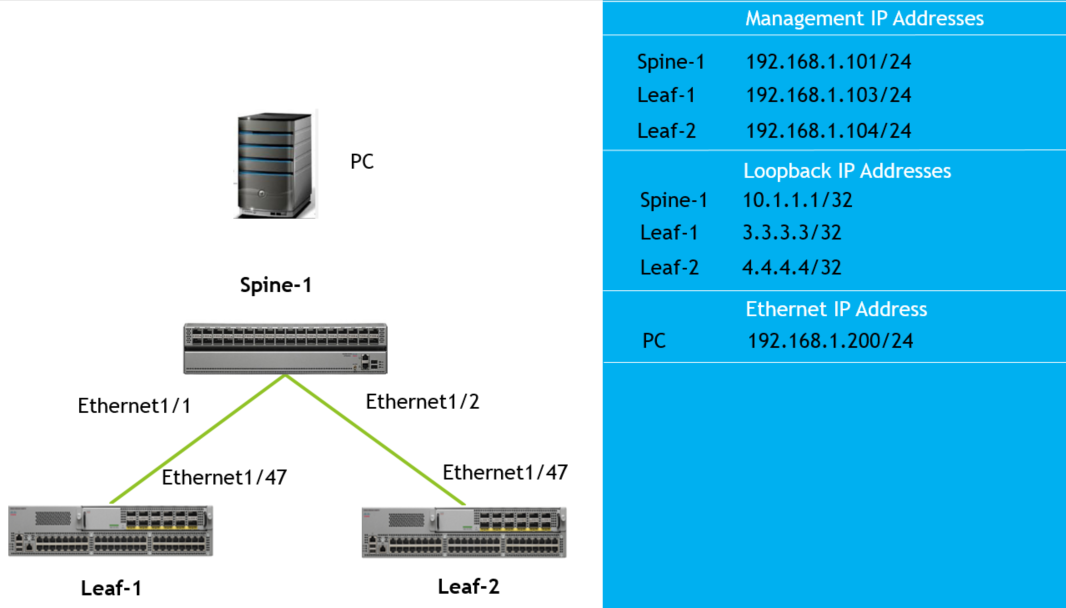
7.0(3)I1(2) or above

L2 Hub/Switch (needed for accessing Nexus 9000 management ports)

PC with Python 2.7.x (<http://python.org>) and Git (<http://git-scm.com>) installed

Management interfaces must be manually configured per the Pod information sheet

## Demo Topology



# VXLAN Setup

## Estimated Time

5 minutes

## Requirements

Switches must be cabled per the topology described in the Getting Started section.

Switches and PC must be connected to a dumb hub or L2 switch in the same bridge domain.

## Steps

Getting EVPN VXLAN with MP-BGP overlay up and running is fast and easy using pre-configured Cisco recommended spine/leaf CLI configurations.

Assuming a 1 spine, 2 leaf topology, simply copy/paste the provided CLI configurations to each Nexus 9000 switch.

## Downloading VXLAN Configurations

VXLAN configurations are available in GitHub.

First, install git (<http://git-scm.com>) if not installed on your PC.

From your PC’s terminal window, issue the following git command:

git clone –b partner\_demo <https://github.com/onecloud/nexus9000.git>

The VXLAN configurations will be created under {your current path}/nexus9000/demo\_vxlan directory.

## Verification Commands

show vxlan

show nve vni

show ip mroute 225.1.1.4

show bgp l2vpn evpn summary

show nve peers

show mac address-table

show bgp l2vpn evpn 1.1.4.20

show nve peers control-plane-vni

show forwarding nve l2 ingress-replication-peers

## Additional Notes

1. Because traffic is required in order to bring up the VXLAN tunnel, you need at least one end point attached to each leaf and have them pass traffic across the fabric (eg: ICMP pings).
2. While copy/pasting, ignore these informational messages if encountered:

Warning: Disabling IGMP snooping for VLAN 3900.

Warning: Deleted all L3 config on interface Vlan3900

Warning: Disabling IGMP snooping for VLAN 30x.

Warning: Deleted all L3 config on interface Vlan30x

Warning: Please configure TCAM region for Ingress ARP-Ether ACL

for ARP supression to work.

Advertise evpn already disabled

# Getting Python Scripts onto Nexus 9000 Switches

## Estimated Time

5 minutes

## Requirements

1. TFTP (or SCP) server
2. Git (<http://git-scm.com>)

## Steps

In order to demo the on-box programmability and troubleshooting Python scripts, you will need to clone the demo scripts GitHub repository then copy the scripts onto one of the switches. It is recommended all scripts get copied on the Spine switch although any Leaf switch should be fine for demonstration purposes. The reason the Spine is recommended is so the CDP/LLDP/OSPF/Multicast demonstrations more clearly show that multiple interfaces can be programmatically configured at the same time. This is not as evident on a Leaf with just one northbound interface up to a single Spine.

First, install git (<http://git-scm.com>) if not installed on your PC.

From your PC’s terminal window, issue the following git command:

git clone –b partner\_demo <https://github.com/onecloud/nexus9000.git>

The VXLAN configurations will be created under {your current path}/nexus9000/demo\_scripts directory.

Note that you may need to move these files to your tftp/scp download directory when copying to the Spine’s bootflash.

The filenames you will need to copy to the Spine are:

find\_freeip.py

sh\_proc\_cpu\_sort.py

sh\_proc\_mem.py

sh\_int\_count.py

sh\_switch\_details.py

cdp.py

lldp.py

ospf.py

pim.py

Example steps of copying Python scripts to Nexus 9000 using SCP:

From switch, typecopy scp: bootflash:

Enter source filename: (provide python filename here)

Enter vrf (If no input, current vrf ‘default’ is considered): management

Enter hostname for the scp server: (provide SCP server’s ip address)

Enter username: (provide SCP username here)

Enter password: (provide SCP password here)

# Programmability Demo – Finding Unused IP Addresses

## Estimated Time

5 minutes

## Preparation

If you don’t have end points directly attached to any of the Leafs, you can instead configure a Loopback interface on a Leaf to simulate end hosts for the purpose of this demonstration. The example Loopback interface below is used to simulate hosts with IP addresses 172.16.11.2 and 172.16.11.4. You will then run the script on the same switch where this dummy Loopback is configured and ping sweep across an IP range that includes these addresses to show they are used.

Example dummy Loopback:

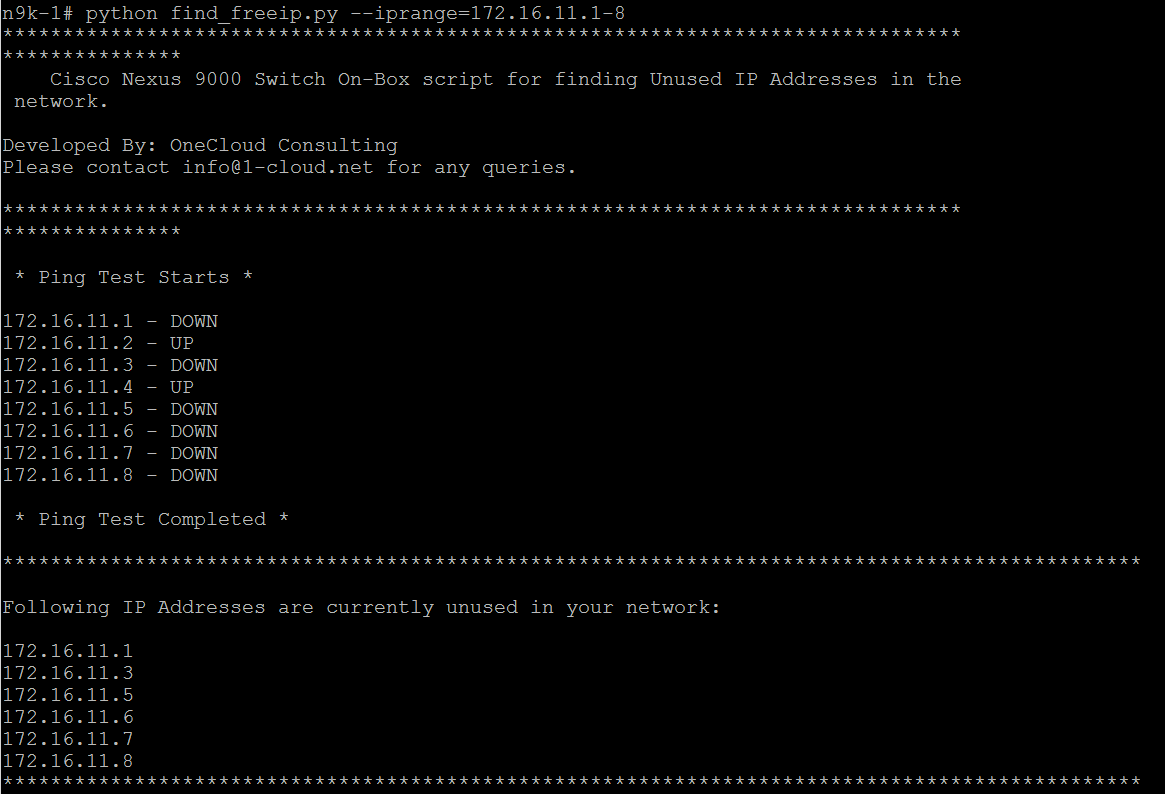
interface loopback100

ip address 172.16.11.2/32

ip address 172.16.11.4/32 secondary

## Steps

1. python find\_freeip.py --iprange=172.16.11.1-8



# Programmability Demo – Super Commands Demo

## Estimated Time

5 minutes

## Steps

1. python sh\_switch\_details.py --ip 3.3.3.3

## Additional Note

3.3.3.3 is a loopback IP address of a Leaf switch learned via OSPF. If you are running the script from the switch configured using this address, simply use a different IP address that is reachable from that switch and ensure it’s reachable.

# Programmability Demo – CDP-to-Interface Description Demo

## Estimated Time

5 minutes

## Requirement

1. CDP must be enabled on switches (‘feature cdp’)

## Steps

1. Issue “show cdp neighbor” to show list of CDP neighbors
2. Issue “show run interface” to show that there are no descriptions configured under any interfaces
3. Run “python cdp.py” on switch
4. Issue “show run interface” again to show that there are now interface descriptions derived from CDP information.



# Programmability Demo – LLDP-to-Interface Description Demo

## Estimated Time

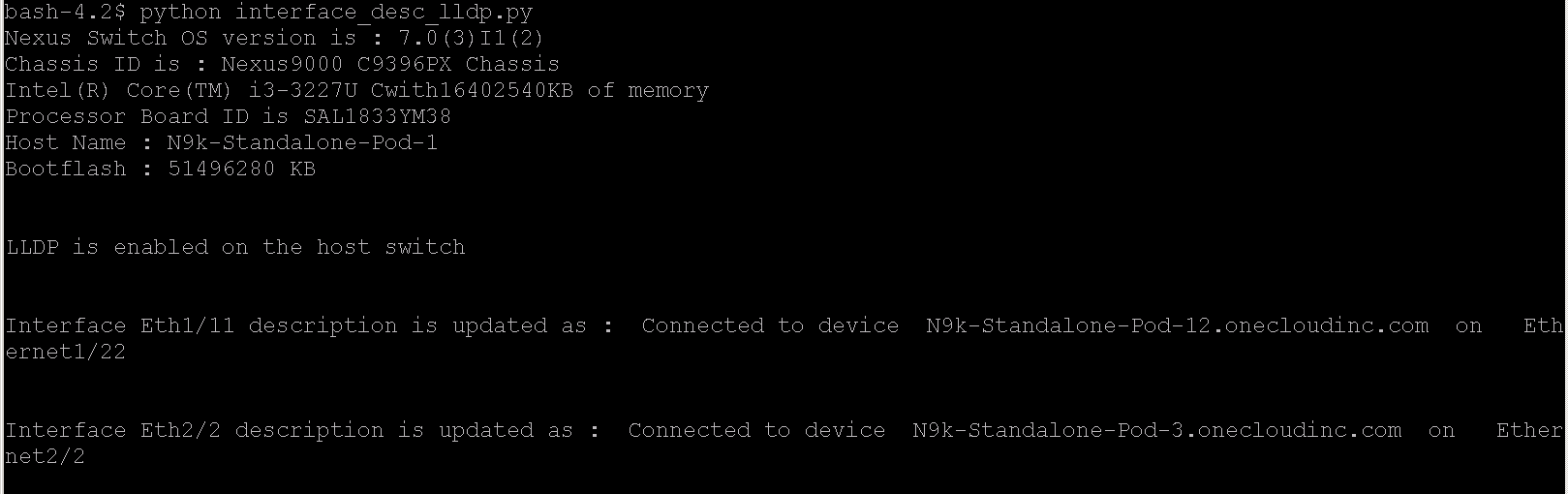
5 minutes

## Requirement

1. LLDP must be enabled on switches (‘feature lldp’)

## Steps

1. Issue “show cdp neighbor” to show list of CDP neighbors
2. Issue “show run interface” to show that there are no descriptions configured under any interfaces
3. Run “python cdp.py” on switch
4. Issue “show run interface” again to show that there are now descriptions derived from LLDP information.



# Programmability Demo – OSPF Demo

## Estimated Time

5 minutes

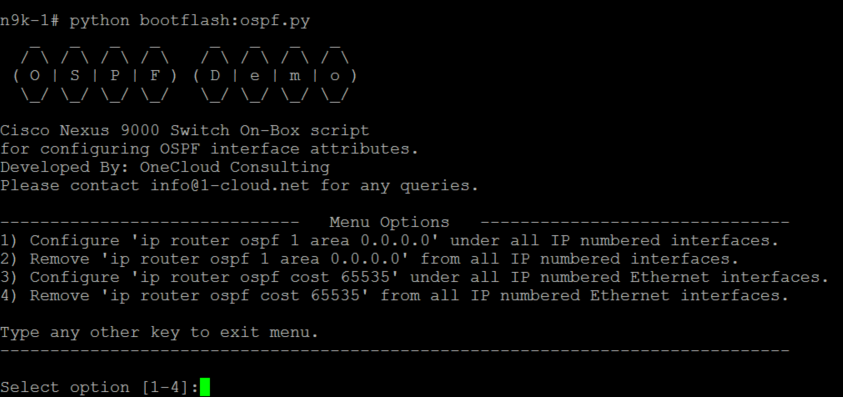
## Requirements

1. OSPF must be enabled on switches:

feature ospf

router ospf 1

## Steps

1. python ospf.py

# Programmability Demo – PIM Demo

## Estimated Time

5 minutes

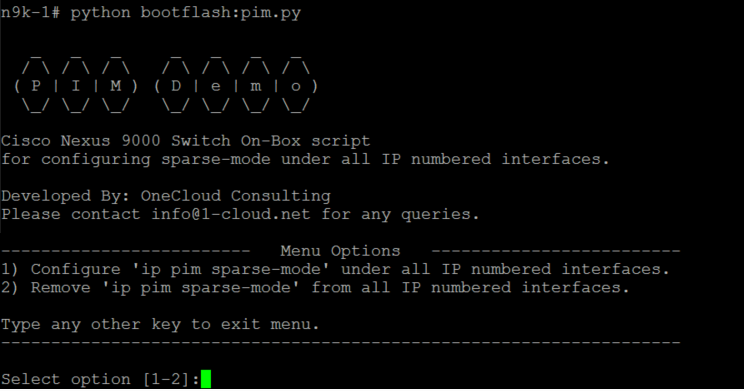
## Requirements

1. PIM must be enabled on switch:

feature pim

## Steps

1. python pim.py



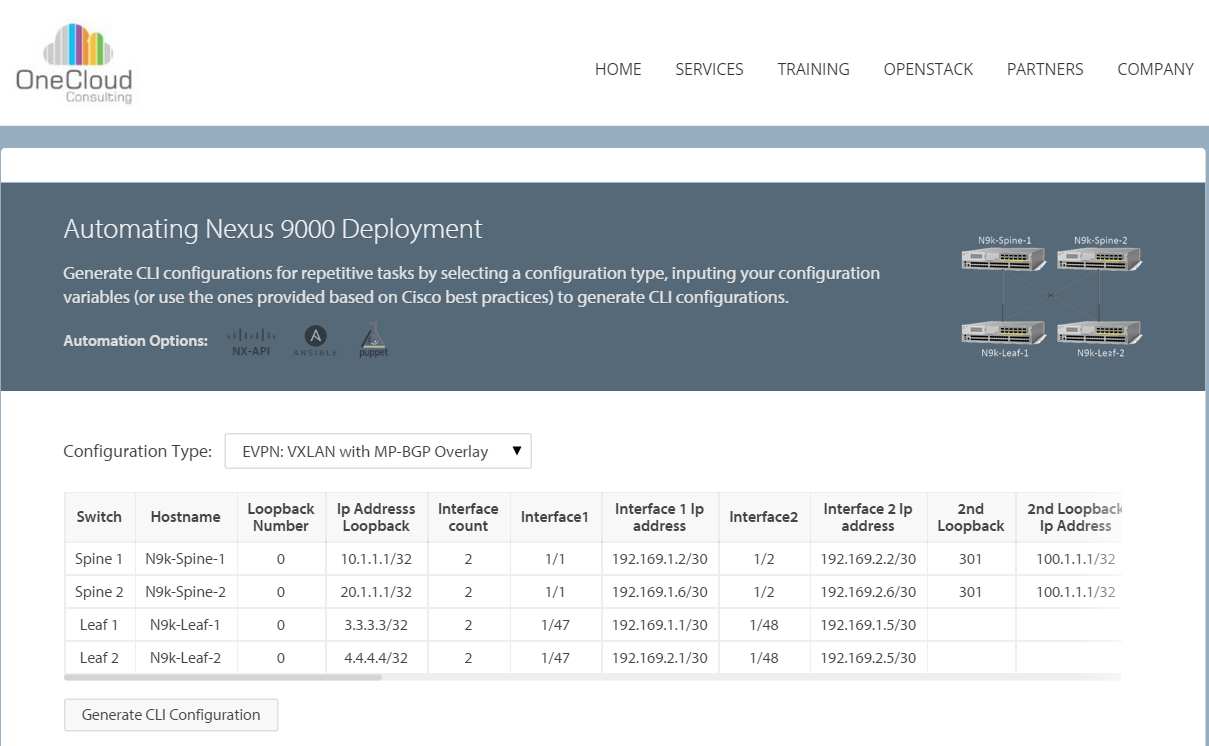
# Automation Demo – Generating CLI VXLAN Configurations for 2-Spine, 2-Leaf Topology

## Estimated Time

5 minutes

## Steps

1. Go to: <http://1-cloud.net/nexus9k-automation/>
2. Click button “Generate CLI Configuration”:



1. Unzip download file and open a sample VXLAN configuration file.
2. Mention table can be modified to reflect customer’s setup to generate customized VXLAN CLI configurations.

# Automation Demo – NX-API Sandbox Demo

## Estimated Time

5 minutes

## Requirements

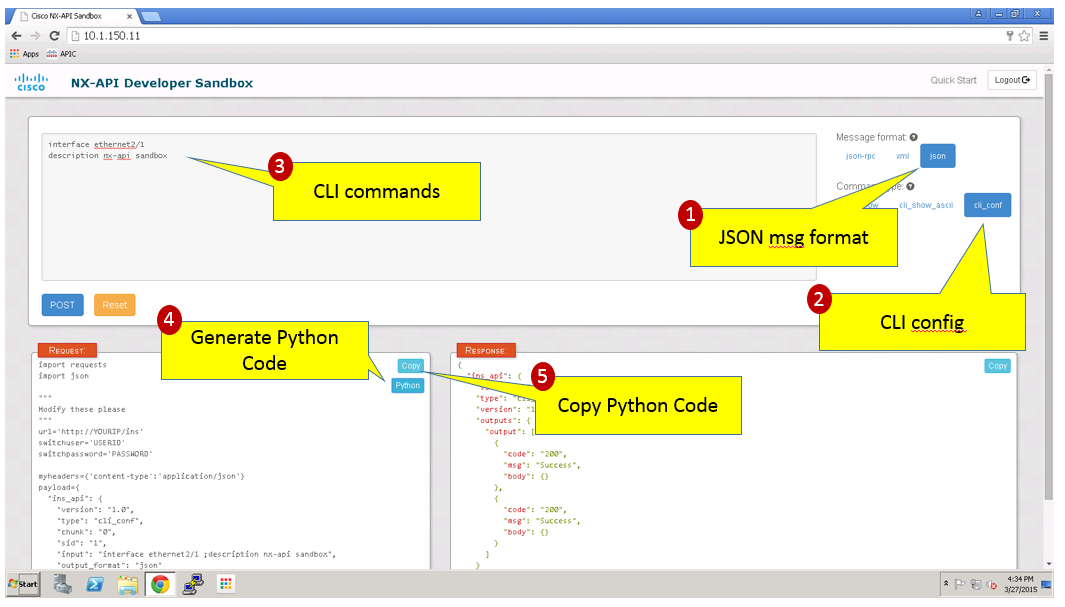
1. NX-API enabled on switches:

feature nxapi

1. Web browser with IP connectivity to switches
2. Python 2.7.x installed on off-box machine where web browser resides

## Steps

1. Type IP address of switch from Chrome browser and follow steps below:



After successfully generating Python code from the NX-API Sandbox, copy/paste Python code into an IDE then run the script. Pycharm has been installed on the Ubuntu virtual machine for you so there is no need to install an IDE.

# Automation Demo – NXOS Version Compliance Demo

## Estimated Time

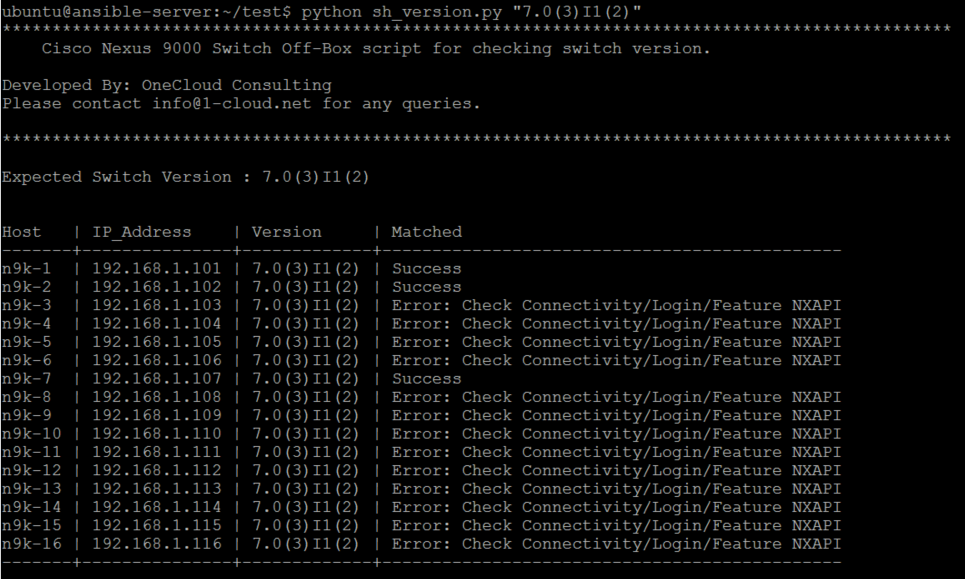
5 minutes

## Requirement

1. IP connectivity between Ubuntu virtual machine and switches.

## Steps

1. python sh\_version.py “*nxos-version*” (where “nxos-version” is the NXOS switch version from “show version”. Example below uses “7.0(3)I1(2)”.



## Additional Notes

Common outputs under the “Matched” column:

Success 🡺 versions matched

Error 🡺 connectivity/login issue or nxapi is not enabled on switch

Version not matched 🡺 version not in compliance (not shown in example output above)

This script uses an accompanying file named *Nexus\_Login\_Info.csv* used to store the list of management IP addresses of the switches in order to query and compare NX-OS versions. This file can be modified to add more spines/leafs in the future.

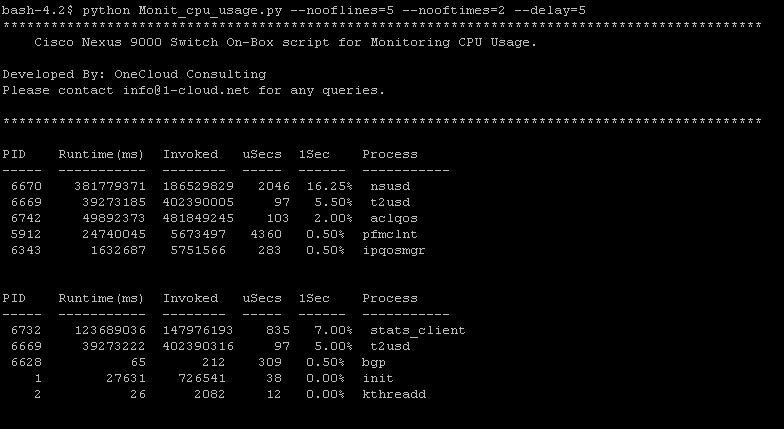
# Troubleshooting Demo – CPU Monitoring Demo

## Estimated Time

5 minutes

## Steps

1. python sh\_proc\_cpu\_sort.py --nooflines=5 --nooftimes=2 --delay=5



## Additional Note

Mandatory Options:

--nooflines == number of lines of ‘show proc cpu sort’ to display

--nooftimes== number of times to execute ‘show proc cpu sort’

--delay == delay in seconds

--log == log output to bootflash (script must be run from Bash for this to work)

Non-mandatory Option:

--log yes == logs output to file on bootflash

This option is useful for capturing output to file and becomes even more powerful when the script is combined with EEM or scheduler when no one is around to login to the switch.

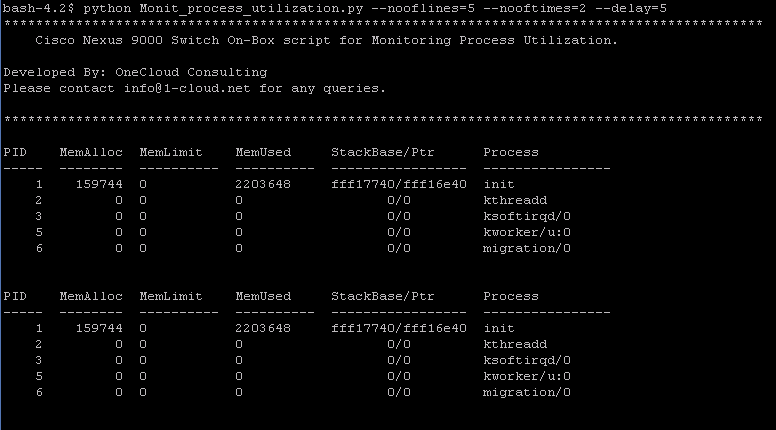
# Troubleshooting Demo – Memory Monitoring Demo

## Estimated Time

5 minutes

## Steps

1. python sh\_proc\_mem.py --nooflines=5 --nooftime=2 --delay=5



## Additional Note

Mandatory Options:

--nooflines == number of lines of ‘show proc memory’ to display

--nooftimes== number of times to execute ‘show proc mem’

--delay == delay in seconds

--log == log output to bootflash (script must be run from Bash for this to work)

Non-mandatory Option:

--log yes == logs output to file on bootflash

This option is useful for capturing output to file and becomes even more powerful when the script is combined with EEM or scheduler when no one is around to login to the switch.

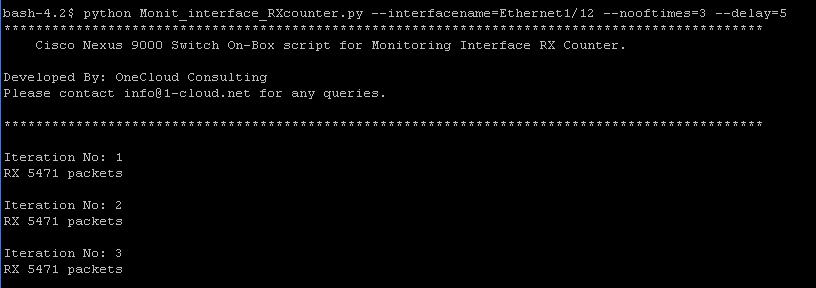
# Troubleshooting Demo – Troubleshooting Traffic Loss Demo

## Estimated Time

5 minutes

## Steps

1. python sh\_int\_count.py --interfacename=Ethernet1/12 --nooftimes=3 --delay=5



## Additional Note

Mandatory Options:

--interfacename == name of switch interface (shorted names can be used)

--nooftimes== number of times to parse interface RX packet counter

--delay == delay in seconds

--log == log output to bootflash (script must be run from Bash for this to work)

Non-mandatory Option:

--log yes == logs output to file on bootflash

This option is useful for capturing output to file and becomes even more powerful when the script is combined with EEM or scheduler when no one is around to login to the switch.

1. You have reached the end of the Cisco Partner Nexus 9000 demo guide.