TECOPS-2201

From Zero to Hero:

Cisco Network Services Orchestrator (NSO)

Intermediate

Sofia Athanasiou - Customer Success Specialist

Hector Oses – Customer Delivery Software Architect

Table of Contents

[Overview 3](#_Toc157796381)

[Learning Objectives 3](#_Toc157796382)

[Disclaimer 4](#_Toc157796383)

[NSO Overview 4](#_Toc157796384)

[Scenario 5](#_Toc157796385)

[Network Diagram 5](#_Toc157796386)

[Task 1: Configure build-in High Availability 5](#_Toc157796387)

[Step 1: Configure… [Task Step by Step Instructions] 6](#_Toc157796388)

[Step 2: Configure/Verify… [Task Step by Step Instructions] 6](#_Toc157796389)

[Task 2: Layered Services Architecture 6](#_Toc157796390)

[Task 3: Kickers and Subscribers 6](#_Toc157796391)

[Task 4: Nano-services 7](#_Toc157796392)

[Related Sessions at Ciscolive 7](#_Toc157796393)

[Summary 7](#_Toc157796394)

Overview

Network operators and service providers today are struggling to control the difference between the growth of their operating costs and their revenue. Introduction and deployment of new services is much slower compared to service demand and availability on the market. It is because of inadequate provisioning processes where services are either configured manually or hard coded inside the Operations Support Systems (OSS). Cisco Network Service Orchestrator (NSO) is the answer to the above challenge. NSO architecture decouples network services from specific components, while automatically configuring the network according to the service specifications. NSO enabled by NETCONF and YANG models, enables operators to dynamically adopt the service configuration solution according to changes in the offered service portfolio.

This session is intended to familiarize the novice NSO user with the architecture and capabilities of the platform, touching standards utilized by NSO, such as NETCONF and YANG. The session will further discuss NSO components, service and device abstraction, integration with northbound systems via Application Programming Interfaces (APIs), communication procedure with southbound devices via Network Element Drivers (NEDs), configuration compliance, and configuration data collection.

Learning Objectives

Upon completion of this lab, you will be able to:

* Configure NSO build-in High Availability
* Scale your services by using Layered Services Architecture
* Configure kickers and use them inside your service
* Use nano-services to create l3mplsvpn service

The lab has 4 main excercises:

* **Exercise 1:** Configure build-in High Availability. It will help you get familiar with the concept of High-Availability and its benefits.
* **Exercise 2:** Create a service using Layered Service Architecture.
* **Exercise 3:** Configure a service taking advantage of FASTMAP using kickers.
* **Exercise 4:** L3MPLSVPN service creation usingnano-services.

Disclaimer

This training document is to familiarize with Cisco NSO intermediate topics for Automating your network. Although the lab design and configuration examples could be used as a reference, it’s not a real design, thus not all recommended features are used, or enabled optimally. For the design related questions please contact your representative at Cisco, or a Cisco partner.

# NSO Overview

Cisco® Network Services Orchestrator (NSO) enabled by Tail-f® is an industry-leading orchestration platform for hybrid networks. It provides comprehensive lifecycle service automation to enable you to design and deliver high-quality services faster and more easily.



OSS

Service Order

Minimal Device Reconfigurations

**NSO**

The network is a foundation for revenue generation. Therefore, service providers must implement network orchestration to simplify the entire lifecycle management for services. For today’s virtualized networks, this means transparent orchestration that spans multiple domains in your network and includes network functions virtualization (NFV) and software-defined networking (SDN) as well as your traditional physical network and all its components

NSO is a model driven (YANG) platform for automating your network orchestration. It supports multi-vendor networks through a rich variety of Network Element Drivers (NEDs).

We support the process of validating, implementing and abstracting your network config and network services, providing support for the entire transformation into intent based networking.

Scenario

There are two servers with NSO version 6.1.5 installed that are part of the same subnet. NSO is running 4 simulated devices taking the roles of Provider Edge in the network. Two of them run Cisco IOS, and the other two Cisco IOS XR.

# Network Diagram

A diagram of a cloud computing system

Description automatically generated

**Table:** Lab first aid

|  |  |  |  |
| --- | --- | --- | --- |
| Host name | IP address | Username | Password |
| NSO-01 | 1 | Lorem ipsum | Lorem ipsum |
| NSO-02 | 2 | Lorem ipsum | Lorem ipsum |
| PE\_00 | 3 | Lorem ipsum | Lorem ipsum |
| PE\_01 | 4 | Lorem ipsum | Lorem ipsum |
| PE\_11 |  |  |  |
| PE\_10 |  |  |  |

**Lab environment**

The lab runs inside dCloud in a Windows machine. NSO it’s installed in a Linux host and can be reached through SSH, GUI and some APIs (RESTCONF will be used) from the windows machine. Ways of development possible for this lab.

1. A picture containing company name

   Description automatically generated(preferred) Visual Studio Code. You can find a shortcut in the desktop to this application. When you start it, you will be connected to NSO server and you will be able to view and edit files from your local Windows. A Terminal is available as well. See Appendix A for more information.
2. Logo

   Description automatically generatedEdit files locally using Notepad++ or Visual Studio Code and then upload them to NSO server through SFTP. Filezilla is installed and prepared to connect to NSO, with a shortcut in the Desktop.
3. Logo

   Description automatically generated with low confidenceConnect to NSO host through putty and edit the files directly there by ‘vim’. Desktop shortcut available.

Lab Introduction and Verification

Logo

Description automatically generated with low confidenceThe NSO version 6.1.5 is already installed and the required Network Element Drivers (NEDS) are loaded.

Desktop shortcut ‘NSO Host’ allows you to connect to the Linux host where NSO is running as user ‘cisco’.

1. Let’s verify the 4 netsim devices are loaded into NSO

admin@ncs# **show devices list**

NAME ADDRESS DESCRIPTION NED ID ADMIN STATE

--------------------------------------------------------

PE\_00 127.0.0.1 - cisco-ios unlocked

PE\_01 127.0.0.1 - cisco-ios unlocked

PE\_10 127.0.0.1 - cisco-ios-xr unlocked

PE\_11 127.0.0.1 - cisco-ios-xr unlocked

1. Verify the required packages are loaded

admin@ncs# **show packages package package-version**

PACKAGE

NAME VERSION

------------------------------

cisco-ios-cli-6.80 6.80.1

cisco-iosxr-cli-7.39 7.39.2

l3vpn 1.0

admin@ncs# **show packages package oper-status**

PACKAGE

PROGRAM META FILE

CODE JAVA BAD NCS PACKAGE PACKAGE CIRCULAR DATA LOAD ERROR

NAME UP ERROR UNINITIALIZED VERSION NAME VERSION DEPENDENCY ERROR ERROR INFO

---------------------------------------------------------------------------------------------------------

cisco-ios-cli-6.80 **X** - - - - - - - - -

cisco-iosxr-cli-7.39 **X** - - - - - - - - -

- -

l3vpn X - - - - - - - - -

admin@ncs#

# Task 1: Configure build-in High Availability

Let’s start exploring how to configure build-in high-availability in NSO 6.1.5.

1. Connect to NSO-01 CLI and verify that high-availability is not enable:
2. Verify the same for NSO-01
3. Verify that both NSOs are having same packages:
4. Verify netsim devices are up and running in NSO-01
5. Verify there are no services configured

Step 1: Enable High Availability

1. Connect to NSO Host 1 and navigate to /etc/ncs directory. There you will find the configuration files of NSO.

cd /etc/ncs

ls

1. Use vi to navigate in ncs.conf file and find ha field:

vi ncs.conf

/<ha>

1. Enable High-Availability by changing value in true
2. Replicate the same for NSO-02
3. Restart NSO-01 and NSO-02
4. Configure High-Availability to NSO-01, that NSO will act as Primary
5. Configure High-Availability to NSO-02, that NSO will act as Secondary
6. Enable High-Availability in NSO-01
7. Enable High-Availability in NSO-02
8. Verify High-Availability is up and running and Primary with Secondary has formed a cluster.

Step 2: Verify High-Availability

1. Enter configuration mode in NSO-02

You should not be able to enter since right now NSO-02 is working as Secondary and has read only abilities

1. Verify in NSO-02 that there is no loopback interface configured in PE\_00
2. Go back to NSO-01 and configure device PE\_00 with a loopback interface:
3. Verify that loopback interface is configured for PE\_00 in both NSOs:
4. Now configure service loopback through NSO-01 and verify the same will be visible in NSO-02:
5. It is time to bring down NSO-01 and see NSO-02 taking over, stop NSO-01
6. Verify that NSO-02 has taken over
7. Bring back NSO-01 and revert roles :
8. Verify High-Availability is working between the two nodes.
9. To conclude disable HA again from both by changing ncs.conf file and restart both NSOs:

During High-Availability, a NSO node can be in three different modes:

1. Primary
2. Secondary
3. None

None is supporting so that there are no additional commits permitted while Primary is Down, that way we are not facing any longer problems of Split Brain.

Another important note is that while configuring the high-availability token, in local install the token between Primary and Secondary might be different. In system install they should be the **same**, otherwise Primary and Secondary will not be able to communicate and form a cluster.

# Task 2: Layered Services Architecture

# Task 3: Kickers and Subscribers

# Task 4: Nano-services

**Table:** Lorem ipsum dolor sit amet

|  |  |  |  |
| --- | --- | --- | --- |
| Lorem ipsum | Lorem ipsum | Lorem ipsum | Lorem ipsum |
| Lorem ipsum | 1 | Lorem ipsum | Lorem ipsum |
| Lorem ipsum | 2 | Lorem ipsum | Lorem ipsum |
| Lorem ipsum | 3 | Lorem ipsum | Lorem ipsum |
| Lorem ipsum | 4 | Lorem ipsum | Lorem ipsum |

# Summary

Deliver high-quality services faster and more easily through network automation. Cisco Network Services Orchestrator (NSO) is industry-leading software for automating services across traditional and virtualized networks. Use NSO to add, change, and delete services without disrupting overall service, and help ensure that services are delivered in real time.

NSO is now [free to download](https://developer.cisco.com/site/nso/?utm_campaign=nso19&utm_source=website&utm_medium=nso-try) for non-production use! Download NSO to evaluate and learn how to automate your network and orchestrate your services using NETCONF and YANG today. Also do not miss the opportunity to practice more through : <https://developer.cisco.com/site/nso/> , there you can find lab guides, sandbox labs and learning tracks to support you in your automation journey.

[Graphical user interface, text, application, chat or text message

Description automatically generated](https://www.cisco.com/c/en/us/solutions/service-provider/solutions-cloud-providers/network-services-orchestrator-solutions/business-benefits-of-automation-and-orchestration.html)

Ref.: <https://www.cisco.com/c/en/us/solutions/service-provider/solutions-cloud-providers/network-services-orchestrator-solutions.html>

Related Sessions at Ciscolive

You can search CiscoLive Amsterdam content catalog with specific keyword and recommend sessions that are relevant to your lab.

[Content Catalog Link](https://www.ciscolive.com/emea/learn/sessions/content-catalog/)

XYZ Troubleshooting [BRKABC-2000] Name of the Speaker, Technical Leader, Cisco