

COURSE OUTLINE

Cisco NSO Administration and DevOps (NSO303) v4.0

Duration: 4 Days

Prerequisites:

Before you take this training, we recommend that you have the knowledge and skills obtainable by attending the Cisco Network Services Orchestrator Foundation (NSO201) class, including:

- Basic knowledge of the Cisco Command-Line Interface (CLI)
- Basic knowledge of the CLI of UNIX-like operating systems
- Basic knowledge of Yet Another Next Generation (YANG) data modelling
- Basic knowledge of Python software development

Course Description:

The Cisco Network Services Orchestrator (NSO) Administration and DevOps (NSO303) v4.0 training continues the learning journey of the NSO Essentials for Programmers and Network Architects (NSO201) v4.0 and NSO Advanced for Python Programmers (NSO300) v4.0 trainings by introducing you to the system administration and DevOps focusing on NSO; the robust bridge linking network automation and orchestration tools, examining the development, operation, and administration task functions. You will learn how to set up, configure, deploy, and maintain a Cisco Network Services Orchestrator solution, and learn best practices for using DevOps. The examples shown in this training demonstrate real-world scenarios to prepare you for deployment and management of new or existing NSO instances. The training guides you through the setup of a production ready NSO instances using system installation with access control settings, the deployment of NSO in Docker containers, and introduces modern DevOps concepts and tools such as Git and Continuous Delivery/Continuous Deployment (CI/CD). You will learn how to migrate CDM devices, how to build NETCONF NEDs from the NSO CLI, how to handle NSO Alarms, and many more features that benefit you in your journey with Cisco NSO.

Course Objectives:

- Describe network and IT convergence
- Describe Cisco NSO architecture
- Describe Linux
- Configure Cisco NSO
- Set up access control to Cisco NSO system

- Describe Cisco NSO Integration Options
- Explain version control systems and basic git concepts
- Describe the purpose of continuous integration and continuous delivery
- Implement Cisco NSO high availability
- Describe scalable system management
- Describe software development methodologies
- Describe service maintenance
- Perform NED upgrades
- Use Cisco NSO for managing services and their associated device configurations
- Describe Cisco NSO change management
- Explain service problem management
- Use Cisco NSO for service monitoring and compliance reporting
- Describe Cisco NSO inventory management
- Describe Cisco NSO use cases

Intended Audience:

- DevOps engineers
- Integration engineers
- Network and software architects
- Network engineers
- Software engineers
- System administrators

Course Outlines:

- Introducing Network and IT Convergence
- Introducing Cisco NSO Architecture
- Introducing Linux
- Explaining Cisco NSO Setup
- Exploring Access Control
- Describing Integration Options
- Explaining Version Control System
- Describing Continuous Integration and Continuous Delivery
- Introducing Scalability and High Availability
- Describing Scalable System Management
- Describing Software Development Methodologies

- Introducing Service Maintenance
- Performing Network Element Driver (NED) Upgrades
- Introducing Configuration Management
- Describing Change Management
- Explaining Service Problem Management
- Explaining Service Monitoring and Compliance Reporting
- Introducing Inventory Management
- Describing Cisco NSO Use Cases

Lab outline

- Perform NSO System Install
- Implement Role-Based Access and PAM
- Using Cisco NSO APIs
- Learn to work with Git
- Use NSO in Docker
- Configure High Availability
- Migrating a Monolithic Service to LSA
- Deploying the LSA Services
- Use the Network Connectivity Tool (NCT)
- Perform Service Backup and Restore
- Migrate a CDM Device
- Build a NETCONF NED
- Replacing a Device
- Troubleshoot NSO Alarms and Services
- Creating a Compliance Report

REGISTER NOW!

training@trends.com.ph
 (+632) 8863-2123
 www.trendssacademy.com.ph