



Network Automation Training Course: Python, Ansible, NETCONF and NetDevOps



Network Automation Training Course: Python, Ansible, NETCONF and NetDevOps

Introduction:

Network automation training covering Python, Ansible, NETCONF and NetDevOps is a 5-day course for network engineering, network operations and infrastructure teams, ending with an Automated Configuration Rollout Package for a multi-device network. Organisations still push router, switch and firewall changes by hand, one session at a time, so configuration drift grows, change windows overrun and typing errors cause outages that are hard to reverse. Nominees already configure network devices from the command line, and the course is taught as a lab in which each participant writes, tests and runs automation against virtual devices. CoreConcept Training Center delivers this network automation course.

Course Objectives:

- Assess the current manual change process and select network tasks suited to automation using change logs and outage records
- Write Python scripts that read YAML and JSON device data and collect and parse device output with Netmiko, TextFSM and NAPALM
- Generate standard device configurations from Jinja2 templates fed by structured source-of-truth variables
- Build Ansible inventories and playbooks that apply and verify configuration across multi-vendor routers, switches and firewalls
- Retrieve and change device configuration through NETCONF and RESTCONF using YANG data models
- Run network changes through Git review, CI pipelines and pre and post change validation with a tested rollback path

Target Audience:

- Network engineering staff responsible for designing and changing routed and switched networks
 - Network operations staff responsible for repetitive configuration, audit and fault-finding tasks
 - Security infrastructure staff responsible for firewall rule and access list changes
 - Data centre and campus infrastructure staff responsible for device provisioning and lifecycle work
 - Network tooling staff responsible for scripts, change automation and links to IT service tools
-

Course Outline:

Day 1: Network Automation Drivers, Python Foundations and Current-State Baseline

- Manual Change Risk Review Using Outage and Ticket Records
- Automation Maturity Baseline From Script Inventory and Change Logs
- Python Virtual Environments, pip Packages and Script Structure
- Python Data Types, Loops and Functions Applied to Device Lists
- YAML, JSON and XML Parsing With Python Standard Libraries

Day 2: Device Access Libraries, Configuration Templates and YANG Data Models

- Netmiko SSH Sessions With `send_command` and `send_config_set`
- TextFSM Templates Turning Show Command Output Into Structured Data
- NAPALM Getters for Multi-Vendor Interface, Neighbour and Facts Data
- Jinja2 Templates With Variables, Loops, Filters and Macros
- YANG Data Model Structure Under RFC 7950 With Containers and Lists

Day 3: Ansible Playbooks and Model-Driven Programmability Lab

- Ansible Inventory Groups, Host Variables and Group Variables
- Ansible `network_cli`, `netconf` and `httpapi` Connection Types Compared
- Ansible Playbook Build With Resource Modules, Check Mode and Diff
- NETCONF Operations Under RFC 6241 With `get-config` and `edit-config`
- RESTCONF Requests Under RFC 8040 With JSON Payloads From Python

Day 4: NetDevOps Pipelines, Change Validation and Controller-Based Networks

- Git Branching, Pull Requests and Peer Review for Configuration Changes
- CI Pipeline Stages Running Linting, Template Rendering and Unit Tests
- Batfish Pre-Change Analysis of ACL and Routing Configuration Changes
- NAPALM `compare_config`, `Commit` and `Rollback` for Safe Configuration Replacement
- Controller-Based Networking and SD-WAN Overlay Policy Through Northbound APIs

Day 5: Lab Building an Automated Multi-Device Rollout With Pre and Post Checks

- Rollout Design Brief Covering Device Inventory, Change Window and Rollback
 - Pre-Change State Capture With NAPALM Getters and Parsed Show Output
 - Jinja2 and Ansible Rollout Execution Across a Multi-Vendor Lab Topology
 - Post-Change Validation Comparing Routing, Interface and Neighbour State Snapshots
 - Automated Configuration Rollout Package Completion and Peer Code Review
-

Skills You Will Gain:

- Python Network Scripting
- Structured Data Handling
- Configuration Templating
- Ansible Playbook Development
- Model-Driven Programmability
- Network Source Control
- Pre-Change Validation
- Automated Rollback Planning

Why Attend This Course:

- Hand an Automated Configuration Rollout Package, with playbooks, templates and pre and post check scripts, to the network engineering lead who approves changes
- Decide which network tasks to automate first, and when NETCONF or RESTCONF suits a change better than screen-scraped command-line sessions
- Avoid outages, overrunning change windows and configuration drift caused by manual typing, by testing every change before it reaches production
- Pass on reusable templates, inventory structures and a Git review workflow to colleagues who make network changes

Conclusion:

Back at work, the participant gives the network engineering lead and the change approval board an Automated Configuration Rollout Package containing an inventory, Jinja2 templates, Ansible playbooks, pre and post check scripts and a rollback plan. The unit uses it to decide which recurring changes move from manual sessions to reviewed, tested automation, and how approval works for changes merged through Git. After the first automated change window, the unit should review failed checks, rollback events, execution time and any drift found, then refine templates and tests before adding more device types.

Frequently Asked Questions FAQ:

What should participants know before the network automation training course?

Participants should already configure routers, switches or firewalls from the command line and understand IP routing and VLANs. Prior programming is helpful but not required, because Python is introduced through network tasks from the first day.

How does network automation training differ from a networking fundamentals or network operations centre course?

It assumes participants can already build and run networks and teaches them to change devices through code, templates, APIs and pipelines. Fundamentals courses teach addressing and routing, and operations centre courses cover monitoring, alarms and shifts rather than writing automation.

Is Ansible or Python the better choice for network automation?

Neither replaces the other. Ansible suits repeatable, declarative changes across many devices with little code, while Python gives full control for custom parsing, data handling and integration. Most teams combine them, calling Python libraries and Jinja2 templates from Ansible playbooks.

What does a participant take back from the network automation training course?

Each participant takes back an Automated Configuration Rollout Package for a multi-device lab network: inventory files, Jinja2 templates, Ansible playbooks, pre and post check scripts and a rollback plan, ready to adapt to the changes their own unit makes.