Integrated Python for Network Engineers — Revised Curriculum (Client‑Aligned)

Scope updated to include client‑requested topics: NetOps use cases, RESTful APIs, SNMP/syslog parsing, firewall connectivity, multi‑vendor configuration at scale, Ansible introduction, and explicit inventory builder lab.

Version: v2 — 2025-09-04

# Course Overview

A 5‑day intensive program that integrates Python fundamentals with hands‑on network automation across Cisco and Juniper platforms. The sequence builds from core Python skills to APIs and idempotent multi‑vendor configuration, ending with a capstone automation demo.

# Duration

5 consecutive days × 8 hours (40 hours total).

# Audience & Prerequisites

* Network Engineers and Administrators moving into automation.
* Systems Engineers and SREs supporting NetOps.
* Managers seeking a working understanding of network automation workflows.
* Basic CLI familiarity; Python experience helpful but not required.

# Program Objectives (updated)

* Master Python essentials for NetOps (syntax, types, control flow, functions, modules, classes, exceptions, logging).
* Explain modern network‑management use cases (configuration management, inventory, compliance/diffs, change windows, backups, telemetry/event triage, and zero‑touch provisioning) and decide when to use Python vs. Ansible.
* Work with RESTful APIs end‑to‑end (authentication, JSON, status codes, GET/POST/PUT/PATCH/DELETE, pagination, retries) using Cisco RESTCONF and Junos REST (with offline fixtures).
* Safely connect to routers, switches, and firewalls with Netmiko/NAPALM/HTTPX (timeouts, retries, prompts/privilege modes).
* Parse and normalize device data (text/regex, CSV, JSON, XML) and SNMP/syslog into SQLite/JSON for reporting.
* Automate multi‑vendor configuration at scale (Cisco + Juniper) with idempotence: dry‑run, diffs, validation, and rollback plans.
* Use Ansible (inventories, variables, network modules, check mode/diffs) and understand Python↔Ansible hand‑offs.
* Deliver a capstone: inventory → API/SNMP collection → validated multi‑vendor change with evidence artifacts.

# Software & Tools (updated)

* Python 3.11+ with venv/pip; Black formatter; VS Code.
* Libraries: Netmiko, NAPALM, Nornir, HTTPX/Requests, TextFSM, Jinja2, sqlite3, pysnmp, logging.
* Ansible (collections: community.network, cisco.ios, junipernetworks.junos).
* Device sandboxes: Cisco DevNet & Juniper vLabs (offline fixtures included).
* Git for version control.

# 5‑Day Schedule (revised with client topics)

## Day 1 — Foundations, Device Access & Use Cases

* Orientation and modern NetOps use cases: where Python and Ansible fit in the toolchain.
* Python essentials: variables, types, loops, conditionals, functions, exceptions, logging.
* Device connectivity (routers/switches/firewalls): Netmiko session management, prompts, privilege modes; safe execution patterns.
* Lab 1 — Inventory bootstrap: build an inventory from CSV/YAML; connect to sample routers/switches/firewalls; fetch facts and store in SQLite/JSON.

## Day 2 — Data Structures, Persistence, SNMP & Syslog

* Python data structures; parsing text/regex, CSV, JSON; normalization and persistence to SQLite.
* SNMP module: concepts, pysnmp walks (e.g., ifHCInOctets, ifOperStatus).
* Syslog parsing: consume logs for events (link flaps, config changes) and summarize.
* Lab 2 — Observability mini‑pipeline: collect SNMP counters into SQLite and parse syslog files into a health report.

## Day 3 — RESTful APIs in Network Automation

* REST fundamentals: resources, representations, GET/POST/PUT/PATCH/DELETE, status codes, retries/backoff, authentication.
* Cisco RESTCONF & Junos REST (with mocks/offline fixtures); pagination and filtering.
* Concurrency (ThreadPoolExecutor) for parallel API calls; idempotent read‑modify‑write patterns.
* Lab 3 — API workflows: query device state (GET), stage a change (PUT/PATCH), validate (GET/compare), and roll back (DELETE or compensating change).

## Day 4 — Multi‑Vendor Configuration & Ansible

* Idempotent configuration at scale: templates (Jinja2), diffs/dry‑run/validation, and error handling.
* Lab 4A — Multi‑vendor config: push VLAN creation + interface descriptions to Cisco & Juniper; validate with NAPALM getters and saved diffs.
* Lab 4B — Routing protocol option: automate OSPF adjacency parameters across platforms with validation (optional stretch).
* Ansible introduction (90 minutes): inventories, variables, connection plugins, network modules, check mode, diffs; Python→Ansible hand‑off for the same VLAN/OSPF tasks.

## Day 5 — Capstone Integration & Demo

* Design review: define change window, guardrails, failure/rollback plan.
* Capstone — End‑to‑end automation: use the inventory builder to target devices; gather SNMP/API state; run policy checks; execute a multi‑vendor change (VLAN/interface or OSPF) via Python or Ansible; capture dry‑run, diffs, and validation artifacts.
* Presentations, post‑assessment, and next steps.

# Hands‑On Lab Catalog (explicit list)

* Inventory builder: enumerate devices; collect facts; write to SQLite/JSON.
* SNMP/syslog parser: ingest counters/events and produce a health report.
* Device connectivity: scripts that connect to routers, switches, and firewalls; execute show commands safely.
* REST automation: end‑to‑end workflow using GET/POST/PUT/DELETE against RESTCONF/Junos REST.
* Configuration automation: VLAN creation, interface descriptions, and optional routing protocol (OSPF) across Cisco & Juniper with idempotent validation.
* Ansible intro lab: reproduce the config automation task using playbooks in check mode with diffs.

# Mapping to Client’s Highlighted Requirements

* Use cases in modern network management → Day 1 (orientation block).
* RESTful APIs & their role → Day 3 (lecture + Lab 3).
* Automating configuration across multiple vendors (Cisco/Juniper) → Day 4 (Lab 4A/4B) and Day 5 capstone.
* Introduction to Ansible → Day 4 (90‑minute module + lab).
* Using Python to parse SNMP data or syslog messages → Day 2 (module + Lab 2).
* Writing Python scripts to connect to routers, switches, and firewalls → Day 1 (device connectivity + Lab 1).
* Configuration automation (VLAN creation, interface configs, routing protocols) → Day 4 (Lab 4A/4B).
* Network inventory script → Day 1 (Lab 1) + Day 5 capstone.