ProTech Professional Technical Services, Inc.

protechtraining.com

Course Outline

Practical Network Programming in Python (PNPP)

Course Summary

Description

This 5-day course balances lecture and lab exercises to ensure students can apply key Python programming knowledge and skills required to address network automation challenges in a multi-vendor infrastructure.

Objectives

At the end of this course, students will be able to:

- Explain what Python is and why it is the language of choice for network automation
- Describe how a Python script executes
- Explain data types, application flow control, and libraries
- Use an Integrated Development Environment (IDE) to develop, debug and run scripts
- Use different techniques to capture and use network device output from a variety of equipment
- Apply configuration via multiple libraries
- Validate that executed scripts achieve the desired network functionality
- Demonstrate a programmatic way of planning and implementing a new network service
- Build custom libraries for frequently used functions
- Apply learned skills to data center services including routing, switching, security and more
- Implement techniques for managing code and applying version control
- Apply Continuous Improvement Continuous Development (CICD) for network engineers
- Prepares engineers for hybrid cloud deployments

Topics

- Introduction to Python and Programming Languages
- IDE – Intro to PyCharm Community Edition
- Data-types in Python
- Flow control
- Network Programming Library Overview
- Modeling Network Configuration CLI in python
- Configuration Management
- Auditing and State Validation
- ˜Show command’ parsing
- Device Capabilities
- Device Real-time Interaction and debugging
- Intro to device orchestration
- Environment and command line variables and Python
- Working with a Datacenter Design
- Multi Device and Dependency Management – Sequencing
- Full configuration management
- Gitlab – CICD Introduction
- RobotFramework

Audience

This course is designed for network architects, network engineers and application developers that have experience designing, building and maintaining enterprise and/or service provider networks or building network automation applications.

Prerequisites

Network engineers participating in this class should have a basic understanding of Python or other scripting/programming languages and working experience with enterprise network infrastructure equipment. Application developers participating in this class should have a basic understanding of networking equipment and their functions on the network. Recommended pre-course reading or training for network engineers include: Automate the Boring Stuff with Python and Introduction to Python Programming courses.
Pre-course work for Application Developers include ENCOR level knowledge or equivalent: ENCOR

Duration

Five days

Course Outline

Practical Network Programming in Python (PNPP)

Course Outline

I. MODULE 1 – INTRO TO PYTHON

A. Lesson 1: Introduction to Python

In this lesson students will review programming basics, learn the fundamentals of Python and examine its code structure and program flows. They will also learn how libraries are used to communicate with networking equipment.

After completing this lesson, the student will be able to:

- Demonstrate basic programming knowledge
- Use the PyCharm IDE to create, edit, execute and debug and manage Python applications
- Describe Python code flow and execution
- List various network programming libraries and describe how to use them to communicate with various network devices through a Python application
- Examine network device configuration from a Python perspective

1. Introduction to Python and Programming Languages
2. Interpreted vs Compiled Languages
3. How a script executes
4. IDE – Intro to PyCharm Community Edition
5. Creating a project
6. Editing code, code completion
7. Running a program
8. breakpoints and debugging
9. version control – Gitlab
10. Data-types in Python
11. Numerics, Strings, Lists, Tuples, Dictionaries
12. Flow control
13. for loops
14. if/then/elif
15. while loops
16. Network Programming Library Overview
17. sshpass, pexpect and calling command line tools from python
18. netmmiko
19. Yang Explained
20. Netconf, ncclient
21. Restconf

B. Lesson 2: Exploring Network Device Data Formats

This lesson focuses on communicating with network devices and the various formats of data produced through interactions with those pieces of networking equipment. Through this exploration students will become comfortable with working with network programming libraries.

After completing this lesson, the student will be able to:

- Issue commands and Inspect/manipulate varied network device output using Python libraries
- Describe the type of output provided by different network programming libraries
- Use the IDE to examine capabilities and limitations of various network programming libraries
- Use the IDE debugger to step through Python code to discover how to communicate with a device programmatically

1. Modeling Network Configuration CLI in python
2. Templates in Jinja2/Yaml
3. Configuration Management
4. Auditing and State Validation
5. JSON vs XML vs raw CLI
6. â€˜show command’ parsing
7. python string operations
8. TextFSM
9. Genie parser
10. json-pretty filter
11. Device Capabilities:
12. Cisco: IOS, XE, XR, NXOS (all different approaches)
13. YDK, Yang Explorer
14. pyATS
15. Juniper JUNOS:
16. unified library: pyez
17. Arista EOS:
18. unified library: pyeapi
19. Device Real-time Interaction and debugging
20. netmiko programming
21. Nexus NX-API
22. IOS-XE Rest API

II. MODULE 2 – NETWORK OPERATIONS AUTOMATION

A. Lesson 1: Network Device Orchestration

In this lesson students will leverage their strong understanding of the network from an architectural view and how to use a services-model approach to plan, automate and validate network functionality. Students will learn how to think like a programmer when addressing network configuration challenges; this insight will also help improve communications in environments where their developers and network engineers exist in separate teams.

Course Outline (cont'd)

After completing this lesson, the student will be able to:

- Approach network configuration from a programmer's perspective
- Use a service-chaining approach to building applications
- Validate and audit operational state and configurations ensuring performance as expected

1. Intro to device orchestration
2. Environment and command line variables and Python
3. Code re-use/abstraction
4. Working with a Datacenter Design
5. Service chain definition – enabling a new application launch in the datacenter
6. vlan management automation
7. Layer 2 validation
8. interface management automation
9. Layer 3 validation
10. BGP/OSPF/Routing configuration automation
11. Routing and Protocol validation
12. Security Automation
13. ACL Validation
14. CDP/LLDP Validation

B. Lesson 2: Configuration, Validation and Auditing

In this lesson students will create/modify live device configurations. Topics covered include non-destructive configuration modifications, configuration management, sequential vs. parallel execution and other best practices. Students will apply what they learned in the previous model to validate and audit their automated network configuration changes.

After completing this lesson, the student will be able to:

- Build and modify device configurations
- Validate and audit applied configurations and changes
- Plan for practical deployments and assess dependencies and scaling requirements
- Turn large configuration files into a database structure for querying

1. Multi Device and Dependency Management – Sequencing
2. Full configuration management
3. Jinja2 / YAML config creation
4. artifact storage
5. configlets (Jinja)
6. devce state templates (Genie, TextFSM)
7. CMDB – Device Inventory
8. Data based python scripting
9. SQLite

C. Lesson 3: Network Automation Ecosystems

In this lesson students will explore the principles of Continuous Integration/Continuous Development (CICD) for the purposes of version control and pipeline execution. Additionally, students will learn to use the RobotFramework to drive their custom automation solutions and to call 3rd party Python libraries, import their own custom libraries, and leverage its powerful validation, auditing and reporting functionality.

After completing this lesson, the student will be able to:

- Customize their automation toolset to address their specific device mix and functional needs
- Use RobotFramework to call Python libraries and audit network functionality
- Discuss how they can apply the solutions developed in this course to automation frameworks beyond RobotFramework such as Ansible

1. Gitlab – CICD Introduction
2. Version Control
3. Pipeline execution
4. RobotFramework
5. Why Robot – a look at the automation ecosystems available
6. Ansible, Puppet, Chef, Salt
7. Robot scripts and keywords
8. Calling 3rd party and custom python libraries from Robot scripts
9. Combining:
10. Configuration Management
11. Health Checks
12. Auditing and Reporting