
Junos Automation

Duration: 4 Days Course Code: JAUT Version: 17.a

Overview:

This five-day course provides students with knowledge of how to automate Junos using DevOps automation tools, protocols, and technologies. Students receive hands-on development experience with tools and languages relevant to automating the Junos OS platform in a DevOps environment. The course includes an introduction to the Junos XML API, and NETCONF but focuses on using Python, PyEZ, and Ansible to automate Junos. The course introduces students to Junos commit, operation (op), event, and SNMP scripts. JSON, YAML, and Jinja2 are introduced as these languages facilitate Junos automation. The course also introduces the Junos Extension Toolkit and related APIs. Finally, the course discusses the use of JSNAPy and Junos ZTP autoinstallation tools. Through demonstrations and hands-on labs, students will gain experience in automating the Junos operating system and device operations. This course uses Junos OS Release 17.1R1, PyEZ 2.0, Python 2.7, and Ansible 2.3

Target Audience:

This course benefits individuals responsible for configuring and monitoring devices running the Junos OS

Objectives:

- Describe the NETCONF protocol.
 - Explain the capabilities of the Junos OS XML API.
 - Describe the use of XSLT, SLAX, and XPath in the XML API.
 - Describe the Junos Automation UI and explain the role of gRPC, NETCONF, and REST in Junos Automation.
 - Identify the languages, frameworks, management suites, and tools used in automating Junos.
 - Describe the YANG Protocol and explain the capabilities of YANG.
 - Use the YANG model to issue Junos commands and to configure Junos.
 - Explain the benefits of using JSON and YAML.
 - List where JSON and YAML are used in Junos Automation.
 - Convert between JSON, YAML, and XML.
 - Describe the features and benefits of using Python in Junos automation.
 - Configure Junos devices to use Python and create simple Python scripts.
 - Describe the function of Junos operation, commit, event, and SNMP scripts.
 - Implement Junos operation, commit, event, and SNMP scripts using Python.
 - Identify how Junos automation uses Jinja2 and create Python scripts that use Jinja2.
 - Explain how PyEZ makes Junos automation easier.
 - Use PyEZ to gather facts from Junos, perform configuration tasks, and use PyEZ to manipulate the file system and perform system upgrades to Junos.
 - Implement OpenConfig in the Junos OS.
 - Describe the process of implementing custom YANG modules.
 - Implement a translation script for a custom YANG module.
 - Explain the use of the Junos REST API in automation.
 - Use the Junos REST API to get information from Junos.
 - Describe what JET is and what it includes.
 - Create a project in the JET IDE.
 - Execute scripts using on-box and off-box automation.
 - Describe how Ansible is used in Junos automation and install Ansible.
 - Create Ansible playbooks to automate Junos.
 - Describe how JSNAPy can help automate Junos devices.
 - Implement JSNAPy into a Junos environment.
 - Describe how ZTP works.
 - Configure in-band ZTP and out-of-band ZTP
-

Prerequisites:

Students should have intermediate-level networking knowledge and an understanding of the Open Systems Interconnection (OSI) model and the TCP/IP protocol suite. Students should also have familiarity with a programming language such as C, C++, Perl, Python, Ruby, or Java. Students should also attend the Introduction to the Junos Operating System (IJOS) course prior to attending this class. Lastly, a high level understanding of object-oriented programming is a plus, but not a requirement.

Content:

COURSE INTRODUCTION

- Why Automate
- Junos MGD Based Automation
- Junos JSD Based Automation
- Automation Languages, Libraries, and Frameworks
- Automation Management Systems

- NETCONF and the XML API
- NETCONF
- XML API
- XML API Programming Languages
- XML API Tools

LAB 1: Exploring the XML API

DAY 2:

- JSON and YAML
- Origins of JSON and YAML
- Features and Benefits
- JSON and YAML uses in Junos Automation
- Creating Well-Formed JSON and YAML Documents
- Conversion To and From XML LAB 2: Using JSON and YAML

LAB 2: Using JSON and YAML

- Introduction to Python and PyEZ
- Python Development Environment
- Working with RPCs using Junos PyEZ
- Working with an Unstructured Junos Configuration
- Working with Junos PyEZ Tables and Views
- Junos PyEZ Exception Handling
- Introduction to Python and PyEZ
- Python Development Environment
- Working with RPCs using Junos PyEZ
- Working with an Unstructured Junos Configuration
- Working with Junos PyEZ Tables and Views
- Junos PyEZ Exception Handling

LAB 3: Implementing Python and Junos PyEZ in Junos

- Jinja2 Overview
- Jinja2 Syntax
- Creating a Junos PyEZ, YAML, and Jinja2 Solution
- Jinja2 Overview
- Jinja2 Syntax
- Creating a Junos PyEZ, YAML, and Jinja2 Solution

LAB 2: Using JSON and YAML

- Introduction to Python and PyEZ
- Python Development Environment
- Working with RPCs using Junos PyEZ
- Working with an Unstructured Junos Configuration
- Working with Junos PyEZ Tables and Views
- Junos PyEZ Exception Handling
- Introduction to Python and PyEZ
- Python Development Environment
- Working with RPCs using Junos PyEZ
- Working with an Unstructured Junos Configuration
- Working with Junos PyEZ Tables and Views
- Junos PyEZ Exception Handling

LAB 3: Implementing Python and Junos PyEZ in Junos

- Jinja2 Overview
- Jinja2 Syntax
- Creating a Junos PyEZ, YAML, and Jinja2 Solution
- Jinja2 Overview
- Jinja2 Syntax
- Creating a Junos PyEZ, YAML, and Jinja2 Solution

LAB 4: Using Jinja2 Templates with PyEZ

DAY3:

- Ansible Overview
- Installing Ansible
- Creating and executing Ansible playbooks to manage devices running Junos OS
- Use Case - Using Ansible to configure devices running Junos OS
- Use Case - Using Ansible to install software on devices running Junos OS
- Use Case - Using Ansible to reboot or shut down devices running Junos OS
- Use Case - Using Ansible to revert a device running Junos OS to a factory Default Configuration

Lab 5: Automating Junos with Ansible

Lab 7: Junos Commit and Op Scripts

- Junos Automation Scripting Overview
- Creating Junos OS Commit Scripts
- Creating Junos OS Op Scripts

Lab 8: Junos event policies and scripts

- YANG Overview
- YANG Modules
- YANG Syntax
- Junos YANG Case Studies
- Creating and Using Non-Native YANG Modules

- Describe the Advantages of OpenConfig
- Modify the Junos OS Configuration using OpenConfig
- Describe using OpenConfig with the Junos Telemetry Interface (JTI)

Lab 9: Implementing OpenConfig

DAY 5 :

- Junos Extension Toolkit (JET)
- JET Overview and Components
- Setup a JET VM
- Create JET Packages
- Use the JET API

Appendix A:

- Zero Touch Provisioning
- ZTP Overview
- The purpose and value of ZTP
- The components and operations of ZTP
- How to deploy a QFX5100 Series switch using ZTP

- Describe the Purpose of the Junos OS REST API
- Create REST API RPC Queries
- Use the REST API Explorer
- Use Case - Using the Junos REST API

Lab 10: Implementing the Junos REST API

LAB 4: Using Jinja2 Templates with PyEZ

DAY 2:

- JSON and YAML
- Origins of JSON and YAML
- Features and Benefits
- JSON and YAML uses in Junos Automation
- Creating Well-Formed JSON and YAML Documents
- Conversion To and From XML LAB 2: Using JSON and YAML

- Origins of JSON and YAML
- Features and Benefits
- JSON and YAML uses in Junos Automation
- Creating Well-Formed JSON and YAML Documents
- Conversion To and From XML

- JSNAPy Overview
- Installation
- Integration into Ansible
- JSNAPy d.Snapcheck, Snapshot, and Diff Functions

Lab 6: Configuring JSNAPy

- Junos Automation Scripting Overview
- Creating Junos OS Commit Scripts
- Creating Junos OS Op Scripts

Further Information:

For More information, or to book your course, please call us on Head Office Tel.: +974 40316639

training@globalknowledge.qa

www.globalknowledge.com/en-qa/

Global Knowledge, Qatar Financial Center, Burj Doha, Level 21, P.O.Box 27110, West Bay, Doha, Qatar