

Advanced Junos Platform Automation and DevOps (AJAUT)

Engineering Simplicity

COURSE LEVEL

Advanced Junos Platform Automation and DevOps (AJAUT) is an advanced-level course.

AUDIENCE

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

PREREQUISITES

Students should have taken the *Junos Platform Automation and DevOps (JAUT)* course or have equivalent knowledge.

COURSE OVERVIEW

The three-day AJAUT course gives students hands-on experience with DevOps and infrastructure as code (IaC) with devices running the Junos OS. Students will learn the tools needed to operate an open-source DevOps environment. Specifically, students will learn to use Docker, GitLab, Ansible, The Robot Framework, and Jenkins. Students will learn and utilize the tools to build a working DevOps project using two Juniper vMX devices.

This course uses Junos OS Release 17.3R1, PyEZ 2.1, Python 2.7, Git 2.17, and Ansible 2.4.

OBJECTIVES

- Understand DevOps and how the DevOps process can improve Junos Automation.
- Create, configure, and manage Docker Containers.
- Use GitLab as a repository for code and configuration data.
- Use Ansible and Jinja2 templates to configure multiple Junos devices.
- Use Ansible to enforce design constraints using templates.
- Use Ansible to build Ansible playbooks that work in multi-vendor environments.
- Install and configure Robot to perform automated tests on Junos devices.
- Use Jenkins to implement continuous code and configuration integration.
- Implement a DevOps automated lab testing solution.

ASSOCIATED CERTIFICATION

JNCIP-DevOps

RELEVANT JUNIPER PRODUCT

Automation

RECOMMENDED NEXT COURSE

N/A

CONTACT INFORMATION

training@juniper.net

COURSE CONTENT

Day 1

1	COURSE INTRODUCTION
2	Introduction to DevOps and Event Driven Infrastructure <ul style="list-style-type: none"> • DevOps • Infrastructure as Code • Event Driven Infrastructure (EDI)
3	Using Docker for DevOps <ul style="list-style-type: none"> • Introduction to Docker Containers • Installing and Configuring Docker • Managing Docker Networking • Managing Applications Running in Docker • Monitoring and Troubleshooting Docker LAB 1: Using Docker Containers

4	Using GitLab as a Configuration and Code Repository <ul style="list-style-type: none"> • Version Control Workflow • Git and GitLab Explained • GitLab Install Overview • Creating GitLab Projects • Creating Git Repositories • Staging and Committing Files • Cloning and Pushing Repository Data • Branching and Merging • Resolving Merge Conflicts LAB 2: Using GitLab
----------	---

Day 2

5	Using Ansible to Manage Networking Devices <ul style="list-style-type: none"> • Review of Ansible Basics • Using Ansible with Jinja2 Templates • Using Ansible to Enforce Network Design Constraints using Templates • Using Ansible for (NOOB) deployments while maintaining Idempotency • Creating Multi-Vendor Playbooks • Using Ansible to Check Code In and Out of GitLab • Using GitLab with Ansible for Automated Version Control • Using Ansible for Auditing • Using Ansible with Vagrant LAB 3: Using Ansible for Network Deployments
----------	--

Day 3

6	Robot Framework <ul style="list-style-type: none"> • Robot Overview • Perform Automated Testing using Robot • The pybot_jrouter Modules • Automated Testing - Use Case LAB 4: Using the Robot Framework for Automated Testing on Junos Devices
7	Jenkins <ul style="list-style-type: none"> • Jenkins Overview • Importing Jobs into Jenkins • Implementing Continuous Integration • Git Module • Robot Module • Ansible Module LAB 5: Using Jenkins to Implement Continuous Integration