Faysal Aziz

Mastering Jenkins

A Beginner's Guide to Continuous Integration and Automation - 2 days

FA-0308 · 2 days

DevOps, Cloud & Infrastructure

Introduction

In today’s fast-paced software development world, the need for continuous integration and automation has become essential. Jenkins, an open-source automation server, has emerged as a leading tool in this space, allowing developers and DevOps teams to automate parts of the software development process such as building, testing, and deploying code. Whether you are just starting your journey in software development or transitioning to a DevOps role, mastering Jenkins is an invaluable skill.

This two-day course is designed for complete beginners, aiming to demystify Jenkins and empower you with the foundational skills needed to integrate it into your workflows. With real-world examples and hands-on labs, this course will equip you to automate repetitive tasks, improve code quality, and accelerate your development pipeline.

Learning outcomes

By the end of this course, participants will be able to:

- Understand the core concepts and architecture of Jenkins.
- Set up and configure a Jenkins server from scratch.
- Create and manage Jenkins jobs to automate builds and tests.
- Integrate Jenkins with popular version control systems like Git.
- Implement continuous integration workflows using Jenkins.
- Troubleshoot common Jenkins issues and optimize performance.
- Use Jenkins plugins to extend its capabilities.
- Secure Jenkins and manage user roles and permissions.

Prerequisites

- Basic understanding of software development principles.
- Familiarity with version control systems (e.g., Git) is helpful but not mandatory.
- No prior experience with Jenkins is required.

Training outline

1. Introduction to Jenkins

Overview of Continuous Integration (CI) and Continuous Deployment (CD)

- Understanding the DevOps culture and CI/CD pipelines
- Importance of automation in modern software development

Introduction to Jenkins

- What is Jenkins and why is it popular?
- History and evolution of Jenkins
- Jenkins vs. other CI/CD tools

2. Setting Up Jenkins

Installing Jenkins

- System requirements and prerequisites
- Downloading and installing Jenkins on different platforms (Windows, macOS, Linux)
- Initial configuration and accessing the Jenkins dashboard

Configuring Jenkins

- Understanding Jenkins home directory structure
- Configuring system settings (Java Home, Jenkins URL, etc.)
- Setting up Jenkins as a Windows service or Linux daemon
- Introduction to Jenkins' user interface and navigation

3. Jenkins Plugins and Extensibility

Introduction to Jenkins Plugins

- What are plugins and why are they essential?
- Exploring the Jenkins plugin ecosystem
- Installing and managing plugins
- Updating and removing plugins

Essential Jenkins Plugins for Beginners

- Git Plugin: Integrating Jenkins with Git repositories
- Pipeline Plugin: Creating and managing pipelines
- Email Extension Plugin: Setting up email notifications
- Credentials Plugin: Managing secure credentials within Jenkins

4. Creating Your First Jenkins Job

Understanding Jenkins Jobs and Builds

- What is a Jenkins job?
- Different types of Jenkins jobs: Freestyle vs. Pipeline

Creating a Simple Freestyle Job

- Step-by-step walkthrough of creating your first Jenkins job
- Configuring source code management (SCM) with Git
- Adding build steps and post-build actions
- Running a Jenkins job manually
- Analyzing build results and logs

Scheduling Jobs with Build Triggers

- Setting up periodic builds
- Triggering builds based on SCM changes
- Manual vs. automated job triggers

5. Introduction to Jenkins Pipelines

Understanding Jenkins Pipelines

- What is a Jenkins Pipeline?
- Declarative vs. Scripted Pipelines
- Benefits of using pipelines over freestyle jobs

Creating a Simple Pipeline

- Writing a basic Jenkinsfile
- Pipeline syntax and structure
- Building and running a pipeline job

Visualizing Pipelines

- Using the Blue Ocean plugin for intuitive pipeline visualization
- Analyzing pipeline stages and logs

6. Integrating Jenkins with Version Control Systems

Introduction to Version Control Integration

- The role of version control in CI/CD
- Supported version control systems in Jenkins

Integrating Jenkins with Git

- Setting up Jenkins to work with GitHub, GitLab, or Bitbucket
- Configuring Git SCM in Jenkins jobs
- Handling branches and pull requests in Jenkins
- Automating pull request validation with Jenkins

7. Continuous Integration with Jenkins

Automating Builds with Jenkins

- Compiling code and running automated tests
- Managing build artifacts and archiving results
- Publishing build results (e.g., test reports, code coverage)

Implementing Continuous Integration Workflows

- Building a CI pipeline from scratch
- Integrating automated tests into the pipeline
- Best practices for a robust CI setup

Notifications and Alerts

- Setting up email notifications for build status
- Integrating Jenkins with messaging platforms (e.g., Slack)
- Configuring alerts for failed builds

8. Advanced Jenkins Features

Securing Jenkins

- Configuring user authentication and authorization
- Setting up role-based access control (RBAC)
- Managing credentials and secrets securely

Jenkins Performance Optimization

- Monitoring Jenkins performance and health
- Configuring Jenkins for optimal performance
- Scaling Jenkins with master-slave architecture

Troubleshooting Jenkins

- Common issues and how to resolve them
- Reading and interpreting Jenkins logs
- Managing and troubleshooting Jenkins jobs

9. Conclusion and Next Steps

Recap of Key Concepts

- Reviewing what was covered during the course
- Highlighting important takeaways for continuous learning

Resources for Further Learning

- Jenkins documentation and community resources
- Recommended books and online courses
- Getting involved in the Jenkins community

Q&A and Open Discussion

- Addressing any remaining questions
- Sharing experiences and challenges with Jenkins

By the end of this comprehensive two-day course, you will have gained a solid understanding of Jenkins and its role in modern software development. You'll be able to confidently set up Jenkins, create jobs and pipelines, and integrate it into your development workflow to achieve continuous integration and automation.

© 2026 Faysal Aziz · Kuala Lumpur