

Red Hat DevOps Pipelines and Processes: CI/CD with Jenkins, Git, and Test Driven Development

Course Duration: 32 Hours

Course code: DO400

1. Course Overview

This four-day course focuses on implementing DevOps pipelines and processes using industry-standard tools such as Jenkins and Git, along with Test Driven Development (TDD) practices. You will learn how to automate software delivery, integrate continuous integration and continuous deployment (CI/CD) pipelines, and improve code quality through testing and collaboration.

2. What you'll learn?

By the end of the course, you will be able to:

- Understand DevOps principles, culture, and lifecycle
- Implement version control using Git
- Build and manage CI/CD pipelines using Jenkins
- Apply Test Driven Development (TDD) practices
- Automate build, test, and deployment workflows
- Integrate tools for continuous monitoring and feedback
- Manage code repositories and branching strategies
- Improve collaboration between development and operations teams

3. Target Audience

This course is ideal for:

- DevOps Engineers
- Software Developers
- System Administrators
- QA Engineers and Testers

- IT Professionals involved in application delivery

4. Pre-Requisites

Before taking this course, you should have:

- Basic knowledge of Linux/Red Hat environment
- Familiarity with software development concepts
- Basic understanding of scripting (Bash, Python, or similar)
- Awareness of software testing fundamentals (recommended)

5. Course content

Module 1: Course Introduction

- Course objectives and structure
- Overview of DevOps tools and lab environment

Module 2: Introduction to DevOps and CI/CD

- DevOps culture and principles
- Continuous Integration, Continuous Delivery, Continuous Deployment
- Benefits of CI/CD pipelines
- DevOps lifecycle and workflows

Module 3: Version Control with Git

- Introduction to version control systems
- Installing and configuring Git
- Working with repositories (local and remote)
- Branching and merging strategies
- Resolving merge conflicts
- Git workflows (Git Flow, feature branching)

Module 4: Managing Source Code Repositories

- Working with GitHub/GitLab
- Repository management and access control
- Pull requests and code reviews
- Webhooks and integrations

Module 5: Introduction to Jenkins

- Jenkins architecture and components
- Installing and configuring Jenkins
- Managing plugins
- Jenkins dashboard and user interface

Module 6: Building CI Pipelines with Jenkins

- Creating Freestyle and Pipeline jobs
- Jenkinsfile and pipeline as code
- Automating builds and tests
- Scheduling and triggering builds

Module 7: Integrating Git with Jenkins

- Connecting Git repositories to Jenkins
- Automating builds on code commits
- Webhooks and polling mechanisms
- Managing credentials securely

Module 8: Test Driven Development (TDD)

- Introduction to TDD principles
- Writing unit tests before code
- TDD lifecycle (Red-Green-Refactor)
- Benefits of automated testing

Module 9: Automated Testing in CI/CD Pipelines

- Integrating unit tests into pipelines
- Test frameworks (JUnit, PyTest, etc.)
- Test reporting and analysis
- Code quality tools integration

Module 10: Continuous Delivery and Deployment

- Packaging and artifact management
- Automating deployments
- Deployment strategies (Blue-Green, Canary)
- Environment management

Module 11: Pipeline Automation and Orchestration

- Multi-stage pipelines
- Parallel execution
- Pipeline optimization techniques
- Handling failures and retries

Module 12: Containerization and Integration (Optional DevOps Extension)

- Introduction to Docker
- Building and managing containers
- Integrating Docker with Jenkins pipelines
- Container-based deployments

Module 13: Monitoring and Feedback Loops

- Continuous monitoring concepts
- Integrating monitoring tools
- Logging and alerting
- Feedback loops in DevOps

Module 14: Security in DevOps (DevSecOps)

- Secure coding practices
- Managing secrets and credentials
- Integrating security scans into pipelines
- Compliance and governance

Module 15: Real-World Use Cases and Hands-on Labs

- End-to-end CI/CD pipeline implementation
- Git + Jenkins + TDD integration project
- Troubleshooting pipeline issues
- Best practices for enterprise DevOps