

Course Outline

Implementing a CI/CD Pipeline Course DEVOPSPIPELINE3: 3 days Instructor Led

About this course

Building, testing, and deploying code can be complex and time-consuming. This three-day, instructor-led, hands-on, technical class provides the solution of how to create an automated CI/CD pipeline that builds, tests, and deploys code every time a developer pushes a code change, ensuring only high-quality and approved code gets deployed.

Developers don't use isolated software tools, so why learn tools in isolation? This course combines the most widely used enterprise technologies into single, integrated, end-to-end pipelines. We'll teach you git from the ground up so you can securely push code to a remote repository such as Github or Bitbucket. Then we'll teach you how to configure servers using Ansible from code stored in git. Next, we'll integrate git and Ansible with Jenkins CI/CD to build, test, and deploy your code into Staging and Production environments, creating an automated end-to-end DevOps pipeline.

If desired, private clients can substitute Bitbucket for Github and/or Chef for Ansible. It is also possible to expand the class duration to include Terraform and/or Docker. Find your solution with our DevOps Course Builder.

If you are using Azure DevOps Pipelines, please review the Implementing Azure DevOps Pipelines course.

Audience profile

Required Prerequisites:

- Understand the purpose of infrastructure servers
- Familiar with scripting of any kind
- Experience working at the command line of Unix, Linux, or PowerShell
- Experience with a text editor of any kind

Professionals who may benefit include:

- DevOps Engineers
- System Operations
- Code Developers
- Software Developers
- Quality Assurance Professionals
- Software Testers
- Product Owners
- Infrastructure Engineers
- Development Team Members
- Operations Team Members
- Project Managers
- Technical Product Managers

Course Outline

At course completion

After completing this course, students will be able to:

- Maintain code using Git and GitHub
- Create, test, and deploy Ansible playbooks through a Jenkins CI/CD pipeline
- Complete integration tests
- Write CI/CD as code using Jenkins
- Compile, test, package, and permanently store Java applications
- Store artifacts

Course Outline

Part 1: Technology Overview

1. Git - Source Control Management
2. Chef - Configuration Management
3. Jenkins - Continuous Integration/Continuous Deployment

Part 2: Git - Source Control Management

1. Purpose overview and use cases
2. Git workflow
3. Configuring git on your local machine
4. Getting help with Git
5. Local vs. Global vs. System configurations
6. Basic Git Commands
7. Creating local git repositories
8. Branching and merging
9. Using remote repositories
10. Pushing code to Github using public and private SSH keys

Part 3: Ansible - Configuration Management

1. Ansible purpose and use cases
2. Architecture and call flow
3. Ansible installation, configuration, and validation
4. Control nodes and managed nodes
5. Ansible managed hosts
6. Host inventory; hosts and groups
7. Repeatable code: Playbooks
8. Introduction to YAML
9. Modularizing code: Roles
10. Ansible variables
11. Dynamic configuration with facts
12. Finding errors: Ansible unit testing
13. Ensuring code quality: Ansible integration testing

Course Outline

Part 4: Jenkins – Continuous Integration / Continuous Deployment

1. CI/CD overview, use cases and history
2. Plugin architecture
3. Initializing a Jenkins server
4. Projects and jobs
5. Freestyle jobs
6. CI/CD as Code: Pipeline projects
7. Declarative vs. scripted pipelines
8. Jenkins Environment variables and parameters
9. Distributed architecture: Master and agent nodes
10. Views and Folders
11. Managing credentials and secrets
12. Integrating with git Source Control Management
13. Triggers: Webhooks and Polling
14. Notifications: Instant messaging and SMTP Email
15. Approval inputs
16. Testing Ansible playbooks in Jenkins
17. Multibranch Pipelines: Reading entire repositories
18. Conditional Logic
19. Deploying Chef cookbooks with Jenkins: An automated end-to-end deployment pipeline