Course duration

- 3 days

Course Benefits

- Download and install Maven
- Build a project
- Work with Maven's directory structure, plugins, repositories, and more
- Understand the Project Object Model (POM)
- Build a complete web application using Maven
- Build and activate profiles
- Use Maven from Eclipse via the m2eclipse plugin
- Install and configure Jenkins in a servlet container
- Create Jenkins builds
- Configure and use Apache Ant and Apache Maven with Jenkins
- Use Jenkins to generate Java coding standards reports, code coverage reports, and change notices
- Use Jenkins to automatically deploy software into a testing environment.
- Configure Maven and Jenkins to deploy the generated artifacts to Nexus

Course Outline

1. Introduction to Continuous Integration, Continuous Delivery and Jenkins-CI
    1. Foundation of Agile AppDev
    2. XP Flow
    3. Extreme Programming
    4. Agile Development
    5. What is Continuous Integration
    6. What is Continuous Integration (cont'd)
    7. Typical Setup for Continuous Integration
    8. Setup Notes for Continuous Integration
    9. CI with Artifact Management
    10. What is Continuous Delivery?
    11. Why Continuous Delivery?
    12. DevOps and Continuous Delivery
    13. Continuous Delivery Challenges
    14. Continuous Delivery vs Continuous Deployment
    15. Jenkins Continuous Integration
    16. Jenkins Features
    17. Running Jenkins
    18. Summary

5. Job Types in Jenkins
    1. Introduction
    2. Different types of Jenkins Items
    3. Different types of Jenkins Items (cont'd)
    4. Configuring Source Code Management(SCM)
    5. Working with Subversion
    6. Working with Subversion (cont'd)
    7. Working with Git
    8. Storing Credentials
    9. Service Accounts
    10. Storing Credentials (cont'd)
    11. Build Triggers
    12. Schedule Build Jobs
    13. Polling the SCM
    14. Polling vs Triggers
    15. Maven Build Steps
    16. Summary
6. Getting Started With Maven
    1. Terminology and Basic Concepts
    2. Artifacts
    3. Lifecycle
    4. Default Lifecycle
    5. Plugins
    6. Running Maven - the Story So Far
    7. Running Maven from an IDE
    8. Common Goals
    9. pom.xml
    10. Example
    11. Example (cont'd)
    12. Artifact Coordinates
    13. Standard Layout for Sources
    14. Summary
7. A Web Application in Maven
    1. A More Complex Project
    2. Putting it Together With Maven
    3. Packaging the Target Artifact
    4. The Source Tree
    5. Dependencies
    6. Transitive Dependencies
    7. Dependency Scope
    8. Working With Servers
    9. Declaring and Configuring Plugins
    10. Running the Plugin
    11. Binding a Plugin Goal to the Lifecycle
    12. Archetypes
    13. Summary
8. Commonly Used Plugins

8. Defining a Pipeline
9. A Pipeline Example
10. Pipeline Example (cont'd)
11. Parallel Execution
12. Creating a Pipeline
13. Invoking the Pipeline
14. Interacting with the Pipeline
15. Pipeline vs Traditional Jobs
16. Conclusion

19. Best Practices for Jenkins
    1. Best Practices - Secure Jenkins
    2. Best Practices - Users
    3. Best Practices - Backups
    4. Best Practices - Reproducible Builds
    5. Best Practices - Testing and Reports
    6. Best Practices - Large Systems
    7. Best Practices - Distributed Jenkins
    8. Best Practices - Summary

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Jenkins class:

- Familiarity with Java development practices.