

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 Artifactory

Course Outline

1. Introduction to Continuous Integration, Continuous Deployment and Jenkins-CI
 1. Agile Development
 2. Agile Development (cont'd)
 3. What is Continuous Integration
 4. What is Continuous Integration (cont'd)
 5. What is Continuous Integration (cont'd)
 6. Typical Setup for Continuous Integration
 7. Continuous Deployment
 8. Continuous Deployment (cont'd)
 9. DevOps and Continuous Deployment
 10. Continuous Deployment Challenges
 11. Jenkins Continuous Integration
 12. Jenkins Features
 13. Running Jenkins
 14. Summary
2. Introduction to Apache Maven
 1. Build Tools for Java
 2. Build Tools for Java (cont'd)
 3. History of Build Tools

4. Traditional Scripting
5. 'make'
6. Problems with Make
7. Manual Build with JavaC
8. ANT
9. Pros and Cons of Ant
10. Apache Maven
11. Goals of Maven
12. What is Apache Maven?
13. What is Apache Maven (cont'd)
14. Why Use Apache Maven?
15. The Maven EcoSystem
16. Consistent Easy-to-Understand Project Layout
17. Convention Over Configuration
18. Maven is Different
19. Maven Projects have a Standardized Build
20. Effect of Convention Over Configuration
21. Importance of Plugins
22. A Key Point on Maven!
23. Summary – Key Features of Maven
3. Installing and Running Apache Maven
 1. Downloading Maven
 2. Installing Maven
 3. Run From Command Line
 4. Running Inside an IDE
 5. Settings.xml
 6. Local Repository
 7. Summary
4. Installing and Running Jenkins
 1. Downloading and Installing Jenkins
 2. Running Jenkins as a Stand-Alone Application
 3. Running Jenkins as a Stand-Alone Application (cont'd)
 4. Running Jenkins on an Application Server
 5. The Jenkins Home Folder
 6. Installing Jenkins as a Windows Service
 7. Initial Configuration
 8. Configuration Wizard
 9. Configuration Wizard (cont'd)
 10. Configuring Tools
 11. Configuring Tools - Best Practices
 12. 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. Storing Credentials (cont'd)
10. Build Triggers
11. Schedule Build Jobs
12. Polling the SCM
13. Maven Build Steps
14. 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
 1. Maven Plugins
 2. Declaring and Configuring Plugins
 3. Running the Plugin
 4. Binding a Plugin Goal to the Lifecycle
 5. Maven Surefire Test Plugin
 6. Failsafe Plugin
 7. Site Plugin
 8. JavaDoc Plugin

- 9. PMD Plugin
- 10. Code Coverage – Cobertura
- 11. Summary
- 9. Multi-Module Builds
 - 1. Introduction
 - 2. The Reactor
 - 3. Reactor Sorting
 - 4. Multi-Module Build by Example
 - 5. Summary
- 10. POM Projects
 - 1. Project Object Model (POM)
 - 2. The overall POM structure
 - 3. Storing POM
 - 4. Summary
- 11. Writing Plugins (Maven)
 - 1. What is Maven Plugin
 - 2. Example of Using a Plugin
 - 3. Create a Custom Plugin
 - 4. Create a Custom Plugin (cont.)
 - 5. Plugin Management
 - 6. Summary
- 12. Creating Archetypes
 - 1. Introduction to Maven Archetypes
 - 2. Introduction to Maven Archetypes (cont.)
 - 3. Using Interactive Mode to generate Goal
 - 4. Common Maven Archetypes
 - 5. Summary
- 13. Repository Management
 - 1. Maven's Approach to Artifacts
 - 2. Publishing Artifacts
 - 3. Summary of Maven's Artifact Handling
 - 4. Repository
 - 5. Repository Manager
 - 6. Proxy Remote Repositories
 - 7. Types of Artifacts
 - 8. Release Artifacts
 - 9. Snapshot Artifacts
 - 10. Reasons to Use a Repository Manager
 - 11. Repository Coordinates
 - 12. Addressing Resources in a Repository
 - 13. Summary
- 14. Release Management
 - 1. What is release Management?
 - 2. Release Management with Nexus
 - 3. Release Management with Maven
 - 4. Summary
- 15. Jenkins Plugin

1. Introduction
2. Jenkins Plugins - SCM
3. Jenkins Plugins – Build and Test
4. Jenkins Plugins – Analyzers
5. Jenkins for Teams
6. Installing Jenkins Plugins
7. Summary
16. Securing Jenkins
 1. Jenkins Security - Overview
 2. Jenkins Security
 3. Authentication
 4. Authorization
 5. Confidentiality
 6. Activating Security
 7. Configure Authentication
 8. Using Jenkins's Internal User Database
 9. Creating Users
 10. Authorization
 11. Matrix-Based Security
 12. Note – Create the Administrative User
 13. Project-based Matrix Authorization
 14. Project-Based Authentication
 15. Conclusion
17. Distributed Builds with Jenkins
 1. Distributed Builds - Overview
 2. Distributed Builds – How?
 3. Slave Machines
 4. Configure Jenkins Master
 5. Configure Projects
 6. Conclusion
18. Continuous Deployment and the Jenkins Pipeline
 1. Continuous Deployment
 2. Continuous Deployment (cont'd)
 3. DevOps and Continuous Deployment
 4. Continuous Deployment Challenges
 5. Continuous Deployment with Jenkins
 6. The Pipeline Plugin
 7. The Pipeline Plugin (cont'd)
 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. Conclusion
19. Best Practices for Jenkins

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