

## Course duration

- 2 days

## Course Benefits

- Install and configure Jenkins and Nexus in a servlet container
- Create Nexus repositories of various kinds
- Configure Maven projects to use Nexus repositories
- Create and configure Apache Ivy/Ant build scripts to access Nexus repositories.
- Configure and use Apache Ant and Apache Maven with Jenkins
- Execute a release process on deployable artifacts

## Course Outline

1. Introduction to Dependency Management
  1. Terminology and Basic Concepts
  2. Artifacts
  3. Build Process
  4. Source Code and Source Code Management
  5. Development Process
  6. Managing Code
  7. Dependencies
  8. Repeatable Build
  9. Dependency Management
  10. Historical Dependency Management Practices
  11. Dependencies in Version Control?
  12. Modern Practices
  13. Conclusion
2. Introduction to Continuous Integration 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. Jenkins Continuous Integration
  8. Jenkins Features
  9. Running Jenkins
  10. Summary
3. 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
4. Installing and Running Jenkins
  1. Downloading and Installing Jenkins
  2. Running Jenkins as a Stand-Alone Application
  3. Running Jenkins on an Application Server
  4. Installing Jenkins as a Windows Service
  5. Summary
5. A Jenkins Job
  1. Different types of Jenkins job
  2. Configuring Source Code Management(SCM)
  3. Working with Subversion
  4. Build Triggers
  5. Schedule Build Jobs
  6. Polling the SCM
  7. Maven Build Steps
  8. Summary
6. Apache Ivy
  1. What is Apache Ivy
  2. Using Ivy
  3. Basic Operation
  4. Typical Dependency Life Cycle
  5. Ivy.xml
  6. Using Maven Dependencies
  7. The 'build.xml' file
  8. When You Run the Build Script...
  9. Reporting Dependencies
  10. Installing Ivy
  11. Ivy Settings File
  12. Conclusion
7. 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
8. 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
9. 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
10. Nexus Repositories
  1. Sonatype Nexus
  2. Nexus Editions
  3. Types of Repositories
  4. Publishing Artifacts – From Maven
  5. Publishing Artifacts – Manually
  6. Lab

## 11. Release Management

1. What is release Management?
2. Release Management with Nexus
3. Release Management with Maven
4. Summary
5. Chapter 12. Introduction to DevOps
6. DevOps
7. Collaboration of People
8. Convergence of Process
9. DevOps Builds on Process Theory
10. DevOps Tools
11. Conclusion

## 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 Maven class:

- Familiarity with Java development practices.