Jenkins Training – Continuous Integration with Maven, Jenkins and Nexus

This document provides the curriculum outline of the Knowledge, Skills, and abilities that a Jenkins administrator can be expected to demonstrate. It provide learners with an understanding of the Apache Maven build process, the principles of continuous integration, and the knowledge of how to implement continuous integration with automated test execution using Jenkins, Maven, and the Sonatype Nexus OSS repository manager.

Duration: 3 Days

Hands-On Format: This hands-on class is approximately 80/20 lab to lecture ratio, combining engaging lecture, demos, group activities and discussions with comprehensive machine-based practical programming labs and project work.

Lab: Koenig DC

Module 1 – Introduction to Continuous Integration, Continuous Delivery and Jenkins-CI

- Foundation of Agile AppDev
- XP Flow
- Extreme Programming
- Agile Development
- What is Continuous Integration
- Typical Setup for Continuous Integration
- Setup Notes for Continuous Integration
- CI with Artifact Management
- What is Continuous Delivery?
- Why Continuous Delivery?
- DevOps and Continuous Delivery
- Continuous Delivery Challenges
- Continuous Delivery vs Continuous
- Deployment
- Jenkins Continuous Integration
- Jenkins Features
- Running Jenkins

Module 2 – Introduction to Apache Maven

- Build Tools for Java
- Build Tools for Java (cont'd)
- History of Build Tools
- Traditional Scripting
- 'make'
- Problems with Make
- Manual Build with JavaC
- ANT
- Pros and Cons of Ant
- Apache Maven
- Goals of Maven
- What is Apache Maven?
- What is Apache Maven (cont'd)
- Why Use Apache Maven?
- The Maven EcoSystem
- Consistent Easy-to-Understand Project
- Layout
- Convention Over Configuration
- Maven is Different
- Maven Projects have a Standardized Build
- Effect of Convention Over Configuration

- Importance of Plugins
- A Key Point on Maven!

Module 3 – Installing and Running Apache Maven

- Downloading Maven
- Installing Maven
- Run From Command Line
- Running Inside an IDE
- Settings.xml
- Local Repository

Module 4 – Installing and Running Jenkins

- Downloading and Installing Jenkins
- Running Jenkins as a Stand-Alone Application
- Running Jenkins on an Application Server
- The Jenkins Home Folder
- Installing Jenkins as a Windows Service
- Initial Configuration
- Configuration Wizard
- Configuration Wizard (cont'd)
- Configuring Tools
- Configuring Tools - Best Practices
- Logging in Jenkins
- Custom Log Recorders

Module 5 – Job Types in Jenkins

- Introduction
- Different types of Jenkins Items
- Configuring Source Code Management(SCM)
- Working with Subversion
- Working with Git
- Storing Credentials
- Service Accounts
- Build Triggers
- Schedule Build Jobs
- Polling the SCM
- Polling vs Triggers
- Maven Build Steps

Module 6 – Getting Started With Maven

- Terminology and Basic Concepts
- Artifacts
- Lifecycle
- Default Lifecycle
- Plugins
- Running Maven - the Story So Far
- Running Maven from an IDE
- Common Goals
- pom.xml
- Example
- Example (cont'd)
- Artifact Coordinates
- Standard Layout for Sources

Module 7 – A Web Application in Maven

- A More Complex Project
- Putting it Together With Maven
- Packaging the Target Artifact
- The Source Tree
- Dependencies
- Transitive Dependencies
- Dependency Scope
- Working With Servers
- Declaring and Configuring Plugins
- Running the Plugin
- Binding a Plugin Goal to the Lifecycle
- Archetypes

Module 8 – Commonly Used Plugins

- Maven Plugins
- Declaring and Configuring Plugins
- Running the Plugin
- Binding a Plugin Goal to the Lifecycle
- Maven Surefire Test Plugin
- Failsafe Plugin
- Site Plugin
- JavaDoc Plugin
- PMD Plugin
- Code Coverage – Cobertura

Module 9 – Multi-Module Builds

- Introduction
- The Reactor
- Reactor Sorting
- Multi-Module Build by Example

Module 10 – POM Projects

- Project Object Model (POM)
- The overall POM structure
- Storing POM

Module 11 – Writing Plugins (Maven)

- What is Maven Plugin
- Example of Using a Plugin
- Create a Custom Plugin
- Create a Custom Plugin (cont.)
- Plugin Management

Module 12 – Creating Archetypes

- Introduction to Maven Archetypes
- Using Interactive Mode to generate Goal
- Common Maven Archetypes

Module 13 – Repository Management

- Maven's Approach to Artifacts
- Publishing Artifacts
- Summary of Maven's Artifact Handling
- Repository

- Repository Manager
- Proxy Remote Repositories
- Types of Artifacts
- Release Artifacts
- Snapshot Artifacts
- Reasons to Use a Repository Manager
- Repository Coordinates
- Addressing Resources in a Repository

Module 14 – Release Management

• What is release Management?
• Release Management with Nexus
• Release Management with Maven

Module 15 – Jenkins Plugins

- Introduction
- Jenkins Plugins - SCM
- Jenkins Plugins – Build and Test
- Jenkins Plugins – Analyzers
- Jenkins for Teams
- Installing Jenkins Plugins

Module 16 – Jenkins Security

- Jenkins Security
- Authentication
- Authorization
- Confidentiality
- Activating Security
- Configure Authentication
- Using Jenkins's Internal User Database
- Creating Users
- Authorization
- Matrix-Based Security
- Note – Create the Administrative User
- Project-based Matrix Authorization
- Project-Based Authentication
- Role Based Access Control

Module 17 – Distributed Builds with Jenkins

- Distributed Builds - Overview
- Distributed Builds – How?
- Agent Machines
- Configure Jenkins Master
- Configure Projects

Module 18 – Continuous Delivery and the Jenkins pipeline

- Continuous Delivery
- Continuous Delivery (cont'd)
- DevOps and Continuous Delivery
- Continuous Delivery Challenges
- Continuous Delivery with Jenkins
- The Pipeline Plugin
- The Pipeline Plugin (cont'd)
- Defining a Pipeline
- A Pipeline Example

- Pipeline Example (cont'd)
- Parallel Execution
- Creating a Pipeline
- Invoking the Pipeline
- Interacting with the Pipeline
- Pipeline vs Traditional Jobs

Module 19 – Best Practices for Jenkins

- Best Practices - Secure Jenkins
- Best Practices - Users
- Best Practices - Backups
- Best Practices - Reproducible Builds
- Best Practices - Testing and Reports
- Best Practices - Large Systems
- Best Practices - Distributed Jenkins