

DevOps Foundation for Java Applications (5 Days)

Course Description

This course will utilize a combination of instructor-led discussions, demonstrations and hands-on exercises to introduce DevOps concepts, use of Git, build automation thru Jenkins environment and the concept of Platform as a Service (PaaS) to build, run, manage web applications, and support the entire application lifecycle right from development to deployment. This seminar will focus on; basic concepts of CI and CD, role of Maven, source code via Git, Jenkins architecture, using Master/Slave build machines, integration of Maven projects, plugin environment, code quality metrics and analysis, complex pipeline builds of web- and mobile-based Java applications, usage of automated testing during builds, using parameterized builds, usage of validated merges with Git, security scanning features and build notifications.

Course Objectives

At the completion of this course, attendees will have acquired these skills:

- Understand the role of PaaS, use of Maven and Git and use of Jenkins platform
- Illustrate Jenkins as a tool for automation and review a production deployment of Jenkins environment
- Depict basic server configuration attributes, properties and version control
- Demonstrate how to create, manage and execute projects, and how to monitor their state
- Demonstrate different build job configurations; compile/test, bundling for release, smoke testing and production deployment
- Illustrate Jenkins plugin for Maven and Gradle and common usages
- Depict the importance of integrated tests and metric analysis of build to determine viability
- Understand parameterized build feature and its usage as key building block in creating more complex workflows.
- Illustrate integration of automated testing and reporting abilities
- Depict Jenkins continuous quality metrics monitoring via both static and dynamic code analysis and demonstrate use of common tools
- Utilize Validated Merge feature in Jenkins Enterprise to allow developers to run tests entirely on the server
- Demonstrate Pipeline-based jobs and how to use them to achieve durability, versatility and extensibility.
- Define Jenkins security infrastructure for authentication, authorization and LDAP integration
- Demonstrate notification processes, supported triggers, define build claimants and use of Build Radiators

Audience

Course is designed for Java developers, software architects and technical project leaders that will be using the Jenkins DevOps platform for Continuous Integration and Deployment.

Prerequisites

Solid background in application development, Java is preferable but not required.

Class Format

Lecture (35%), and Lab (65%)

Detailed Course Topics

DevOps Overview

- Defining DevOps concepts
- Core values
 - Culture
 - Automation
 - Measurement
 - Sharing
- Cross-functional Teams
- Value Stream mapping
- Deployment Pipelines
 - Objectives and Goals
 - CD Pipelines
 - CI Pipelines

II. Introduction to Git

- Defining Git
- Version Control
- Local-based Repository
- Git Basics
 - Workflows
 - Push
 - Pull
- Git Event Flow
 - Working Copy
 - Snapshots
 - Staging Area
 - Commit
- Security controls
- Branching and Merging
- Fast Forward Merge
- Using Tags

III. GitFlow

- Concept and Role
- Use of Branches
 - Master
 - Develop
 - Feature
 - Release
- Use of Hotfix

IV. Continuous Code Quality

- Overview
- Benefits and Advantages
- Test types
 - Unit
 - Integration
 - Functional
 - Smoke
- Code Quality Factors
 - Potential Bugs
 - Coding Rules
 - Tests
 - Duplication
 - Comments
 - Architecture & Design
 - Complexity

V. Introduction to CI and CD

- AppDev Foundation
 - Agile
 - Extreme Programming
 - XP Flow
- Continuous Integration
 - Purposes
 - Test-Drive Development
 - QA Automation
 - Automated Build
- Metric Analysis
 - Code Coverage
 - Compliance
 - Static Analysis
- Artifact Management
- Continuous Development
 - Horizontal Scaling
 - Virtualization
 - Quality
 - Traceability
- CD Challenges
- Role of Jenkins

VI. Using Maven

- Java Build Tools
- Goals of Maven
- Use of Repository
 - Role in Build
 - Use of Jar files
 - Plugins
 - Dependencies
- POM File
- Standardized Lifecycle
- Use of Archetypes
 - Templates
 - Common Versions
 - Commands

Jenkins Overview

- Architecture review
- Build Servers
- Slave Servers
- Update Center Administration
- Metrics
- Testing Integration
- Environment Preparation
 - Java
 - Git
 - Configure SSH
 - Repository
- Configurations
 - JDK
 - Maven
 - Git
 - Notifications

VIII. Jenkins Server Configuration

- Config Dashboard
- Global properties
- Build tools
 - Maven
 - Shell Scripting
 - Version Control

IX. Build Jobs

- Jenkins Build Jobs
- Build Triggers
 - Scheduling
 - SCM Polling

- Remote triggering
- Build Steps for Maven
- Post-Build Actions
 - Reporting
 - Archiving
 - Notifications
- Build Execution
- Integration with JaCoCo via plugin

X. Automated Testing

- Overview
- Tools review
- Jenkins configuration
- Using surefire Maven plugin
- Automated Acceptance Tests
- Integrated Performance Tests (JMeter)

XI. Securing Jenkins

- Architecture overview
- Security Realms
 - LDAP
 - Active Directory
 - Container delegation
- Authorizations
 - Matrix-based
 - Project-based
 - Role-based
- Auditing

XII. Notifications

- Notification overview
- Email configuration
- Claiming builds
- Build Radiators
- Instant Messaging and Push notifications
 - Jabber plugin
 - Internet Relay Chat (IRC) plugin
 - Server configuration
- Desktop notification via Eclipse
- Mobile notification (iPhone or Android)
- SMS gateway notification

XIII. Code Quality

- Code Quality Analysis
 - FindBugs
 - CheckStyle
 - Other tools
- Code Quality reporting
- Code complexity reporting

XIV. Advanced Builds

- Parameterized builds
 - Creation
 - Parameterized build scripts
 - Using Git tag
 - Remote execution
 - Job history
- Triggers vs Polling
- Automated Maven jobs
 - Configuration
 - Inheritance
 - Plugin support
- Parallel build coordination
- Using Pipelines
 - Copying artifacts
 - Build promotions
 - Aggregated results

XV. Distributed Builds

- Distributed build architecture
- Master/Slave configuration
 - Master server
 - Slave Agent
 - Remote Service starts
 - Start via SSH
 - Start via Java Web Start
- Build to Slave group association

- Monitor Nodes
- Cloud Computing option

XVI. Automated Deployment and Delivery

- Continuous Deployment
 - Scripts
 - Database updates
 - Smoke testing
 - Rollbacks
- Application Server
 - Java app deployment
 - Script-based deployment
 - .Net apps

XVII. Repository Management

- Use of Artifacts
- Maven Artifact Processing
- Artifact Types
 - Release
 - Snapshot
- Role of Repository Manager
- Release Management
 - Update version
 - Trigger Build
 - Commit/Push (Subversion)
 - Commit to Head

XVIII. Maintenance

- Server load monitoring
- Configuration backup
 - Fundamentals
 - Backup plugin
 - Lightweight automation backups
- Archiving Builds
- Automate Maven Integration testing