Course duration

- 3 days

Course Benefits

- Learn what DevOps is.
- Learn version control with Git.
- Build Java projects with Maven.
- Learn continuous integration with Jenkins.
- Integrate Git with Jenkins.
- Implement Continuous Code Quality with SonarQube.

Course Outline

1. DevOps Fundamentals
    1. Why DevOps
    2. What is DevOps?
    3. Collaborative, Matrixed and Cross-Functional Teams
    4. Key Components of Successful DevOps Teams
    5. DevOps-ification
    6. DevOps Vocabulary
    7. DevOps Goals
    8. Not DevOps - Crush Buzzwords
    9. Driving Business Outcomes with DevOps
    10. Technology-Enabled Business
    11. DevOps Key Enabler for Digital Transformation
    12. Core Values and Mission
    13. Core Values - Culture
    14. Core Values - Automation
    15. Core Values - Measurement
    16. Core Values - Sharing
    17. Communication
    18. Collaboration
    19. Value Stream Mapping
    20. Behavioral Patterns for Success
    21. DevOps Org Structures
    22. DevOps Team - Separate
    23. DevOps Merged Organization
    24. DevOps Overlapped Organization
    25. Organizational Structure Leadership
    26. What Does Continuous Delivery Mean?

27. Deployment Pipelines
28. Your Organization is Doing CD if Pipelining for CD
29. Continuous Integration
30. CI Pipeline
31. CD & CI Methodologies
32. Key Tool Categories for CI/CD
33. Summary

2. Introduction to Git
    1. What is Git
    2. Git's Design Goals
    3. Git's Design Goals (cont'd)
    4. Branching and Merging
    5. Branching and Merging (cont'd)
    6. Centralized Version Control
    7. Distributed Version Control
    8. Git Basics
    9. Git Basics (Cont'd)
    10. Git Basics (cont'd)
    11. Getting Git
    12. Git on the Server
    13. Git Repository Managers
    14. Git on Somebody Else's Server
    15. Summary
3. Basic Git Operations
    1. Using Git
    2. Definitions
    3. Commit
    4. Commit (continued)
    5. How to Think About Commits
    6. Viewing History
    7. Configuring Git
    8. Configuration Scope
    9. User Identification
    10. GPG Signing
    11. Gnu Privacy Guard
    12. GPG Basics
    13. GPG and Git
    14. .gitignore
    15. Other Useful Configurations
    16. Summary
4. Branching, Merging and Remotes
    1. Branching
    2. Branches in Git
    3. Merge
    4. Fast Forward Merge
    5. -no-ff
    6. More Than One Repository

    7. Working with Remotes
    8. Fetch and Pull
    9. Push
    10. Pull Requests
    11. Tagging a Commit
    12. Lightweight Tags
    13. Annotated Tags
    14. Sharing Tags
    15. Checking Out a Tag
    16. Summary
5. Introduction to GitFlow
    1. What is GitFlow
    2. Benefits
    3. How GitFlow works?
    4. How GitFlow works? (Contd.)
    5. What is GitFlow? (Contd.)
    6. How GitFlow works? (Contd.)
    7. GitFlow Extension
    8. Initializing GitFlow
    9. Features
    10. Release
    11. Hotfixes
    12. Summary
6. Continuous Code Quality
    1. Continuous Code Quality
    2. What is SonarQube
    3. SonarQube - Benefits
    4. SonarQube (Multilingual)
    5. Seven Axes of Quality
    6. Potential Bugs
    7. Tests
    8. Comments and Duplication
    9. Architecture and Design
    10. Complexity
    11. SonarQube Installation
    12. SonarQube Components
    13. Code Quality (LOC, Code Smells)
    14. Code Quality (Project Files)
    15. Code Quality (Code)
    16. Summary
7. 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
8. 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
9. 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
10. 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. The Jenkins Home Folder
    5. Installing Jenkins as a Windows Service
    6. Initial Configuration
    7. Configuration Wizard
    8. Configuring Tools
    9. Configuring Tools - Best Practices
    10. Logging in Jenkins
    11. Custom Log Recorders
    12. Summary
11. Job Types in Jenkins
    1. Introduction
    2. Different types of Jenkins Items
    3. Configuring Source Code Management(SCM)
    4. Working with Subversion
    5. Working with Git
    6. Storing Credentials
    7. Service Accounts
    8. Build Triggers
    9. Schedule Build Jobs
    10. Polling the SCM
    11. Polling vs Triggers
    12. Maven Build Steps
    13. Summary
12. 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
13. 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
14. 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
15. 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
16. 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
17. Release Management
    1. What is release Management?
    2. Release Management with Nexus
    3. Release Management with Maven
    4. Summary
18. Jenkins Plugins
    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
19. 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. Role Based Access Control
    16. Conclusion
20. Distributed Builds with Jenkins
    1. Distributed Builds - Overview
    2. Distributed Builds – How?
    3. Agent Machines
    4. Configure Jenkins Master
    5. Configure Projects
    6. Conclusion
21. Continuous Delivery and the Jenkins Pipeline
    1. Continuous Delivery
    2. DevOps and Continuous Delivery
    3. Continuous Delivery Challenges
    4. Continuous Delivery with Jenkins
    5. The Pipeline Plugin
    6. Defining a Pipeline
    7. A Pipeline Example
    8. Parallel Execution
    9. Creating a Pipeline
    10. Invoking the Pipeline
    11. Interacting with the Pipeline
    12. Pipeline vs Traditional Jobs
    13. Conclusion
22. 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 *would be useful* for this DevOps class:

- Some knowledge of Java is beneficial.