ProTech protechtraining.com *"Charting the Course ...*

... to Your Success!"

Continuous Integration with Jenkins

Course Summary

Description

This course will provide participants with an understanding of the principles of continuous integration (CI), and the knowledge of how to implement continuous integration with automated test execution using Jenkins. Participants will use a local copy of Jenkins to create and run jobs, link to a version control system, run automated testing and generate development reports.

Objectives

By the end of this course, students will be able to:

- Install and configure Jenkins in a servlet container
- Create Jenkins builds
- Configure and use Apache Ant and Apache Maven with Jenkins
- Configure and use Node.js with Jenkins
- Use Jenkins to generate Java coding standards reports, code coverage reports, and change notices
- Understand how to use the Jenkins Remote API
- Use Jenkins to automatically deploy software into a testing environment
- Use Jenkins to implement complicated software delivery workflows, called Pipelines

Topics

- Introducing Continuous Integration and Jenkins
- Installing and Running Jenkins
- Simple Jenkins Job
- Securing Jenkins
- Advanced Jenkins
- Jenkins Plugins
- Distributed Jenkins
- Jenkins with Node.js
- Jenkins Remote API
- Extending Jenkins
- Jenkins Pipeline
- Best Practices for Jenkins

Audience

This course is designed for developers.

Prerequisites

Participants should have familiarity with Java development practices.

Duration

Two days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Continuous Integration with Jenkins

Course Outline

I. **Introducing Continuous Integration and Jenkins**
   A. Agile Manifesto
   B. Agile Principles
   C. Agile Methodologies
   D. CI Needs
   E. Challenges
   F. Software Deployment Environments
   G. DevOps
   H. DevOps Needs
   I. Where Does DevOps Fit?
   J. History of Jenkins
   K. State of Jenkins Community
   L. Jenkins Plugin Management
   M. Jenkins Architecture
   N. Summary

II. **Installing and Running Jenkins**
   A. Running Jenkins from a WAR
   B. Installing in a Servlet Container
   C. Setup Security
   D. Email and Version Control
   E. Master/Slave Configuration
   F. Exercise: Install Jenkins
   G. Summary

III. **Simple Jenkins Job**
   A. Create a Job
   B. Configuring a Job
   C. Run a Job Manually
   D. Exercise: Create a Job
   E. Running Jobs Automatically
   F. Run a Job on a Schedule
   G. Summary

IV. **Securing Jenkins**
   A. Authentication and Authorization
   B. Setting up Jenkins Security
   C. LDAP
   D. Authorization
   E. Master/Slave Security
   F. Exercise: Securing Jenkins
   G. Summary

V. **Advanced Jenkins**
   A. Monitoring External Jobs
   B. Distributed Builds
   C. File Fingerprint Tracking
   D. Using Jenkins for Non-Java Projects
   E. Matrix Projects
   F. Splitting Jobs into Smaller Jobs
   G. Exercise: Complex Projects
   H. Summary

VI. **Jenkins Plugins**
   A. Jenkins Plugin
   B. Change Reporting
   C. Core Plugins
   D. Common Plugins
   E. Manage Plugins - Available
   F. Updating Plugins
   G. Advanced Tag
   H. Explore Plugins
   I. Code Coverage
   J. Static Analysis
   K. Performance Reporting
   L. Style Checking
   M. Exercise: Plugins
   N. Summary

VII. **Distributed Jenkins**
   A. Why Distribute Workload?
   B. Setting up a Distributed System
   C. Node Configuration
   D. Running Jobs on Nodes
   E. Exercise: Distributed Build
   F. Summary

VIII. **Jenkins with Node.js**
   A. Node.js or NodeJS?
   B. Node.js Support in Jenkins
   C. Structure of JavaScript Projects
   D. package.json
   E. Integration with TestRunner
   F. Style Checking JavaScript Projects
   G. Starting Integration Servers
   H. Example Node.js Web App
   I. Exercise: NodeJS Plugin
   J. Summary

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Continuous Integration with Jenkins

Course Outline (cont'd)

IX. Jenkins Remote API

- A. Service Calls
- B. Remote API Security
- C. Getting the API Token
- D. Triggering a Build
- E. Retrieving Information
- F. API Libraries - Python
- G. API Libraries - Ruby
- H. Jenkins Discovery
- I. Summary

X. Extending Jenkins

- A. Jenkins Architecture
- B. Persistence
- C. Example Job
- D. Writing Command Line Utilities
- E. Writing Plugins
- F. Plugins Interaction
- G. Plugin Architecture
- H. Extending Existing Plugins
- I. Exercise: Extending Jenkins
- J. Summary

XI. Jenkins Pipeline

- A. Why Pipeline?
- B. What is a Pipeline?
- C. Building a CD Pipeline
- D. Introduction to Groovy
- E. Syntax Differences from java
- F. Key Concepts for Pipeline
- G. Build Steps
- H. Build Steps Diagram
- I. Example Build Script
- J. Gathering Human Input
- K. Allocation Projects and Workspaces
- L. Using Distributed Workers
- M. Workspaces and Workers Diagram
- N. Remote API for Pipeline
- O. Interacting with Pipeline API
- P. Sample Job DSL Script
- Q. Build Pipeline Plugin
- R. Exercise: Pipeline
- S. Summary

XII. Best Practices for Jenkins

- A. General Best Practices
- B. Projects and Dependencies
- C. SDLC Integration
- D. Traceability
- E. Notification
- F. Performance
- G. Summary

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically