



Mastering Continuous Integration with Jenkins

1. Course Objectives

Students will walk away with a solid understanding of how to implement a Continuous Integration (CI) environment with Jenkins, including:

- Setting up a production-grade instance of a Jenkins server, complete with automated builds, tests, code quality audits and reports
- Learning administration of Jenkins instances, such as scripted maintenance, security and slave administration
- Integrating Jenkins with other tools in the development environment, such as issue tracking systems and source code browsers
- Learning the additional features available in CloudBees Jenkins Enterprise

2. Audience

- Beginner and intermediate Java developers
- Software architects
- Technical project managers
- Build managers
- Development and QA engineers



Mastering Continuous Integration with Jenkins

3. Approach

The course is modular, flexible and highly interactive, with extensive hands-on examples. The course roughly breaks into 40% theory and 60% practical content.

4. Course Modules

What is CI?

We'll recap the basic benefits of continuous integration and why we utilize it.

Introducing Jenkins CI (Lab: Installing the Lab Environment)

We'll introduce Jenkins as a tool for automation, then look in more detail how to prepare and set up a production deployment of Jenkins, as well as what you need to consider beforehand.

Jenkins Build Jobs (Labs: Creating a Freestyle Job / Creating a Maven Build Job)

We'll introduce the freestyle job type and the Maven job type in Jenkins. The key parts of the UI and the configuration screen are explained and then additional features in the Maven job type are discussed.

Monitoring Build Jobs (Lab: Displaying Test Results)

More parts of the Jenkins UI are discussed, then we'll look at the stereotypical continuous integration workflow and see how test reports are processed by Jenkins.



Mastering Continuous Integration with Jenkins

Organizing Build Jobs (Lab: Creating an Integration Tests Build)

As a Jenkins instance grows in size, one of the challenges is to keep the UI usable by organizing large numbers of jobs. We'll look at several plugins and techniques to deal with this.

Jenkins Plugins

Jenkins derives a good part of its power from a vibrant plugin ecosystem. In this module, we'll look at how to tap into that, where to find plugins and so on.

Working with Version Control (Lab: Integrating with an SCM Browser)

We'll look into the value of integrating a repository browser with Jenkins and use Subversion as an example to try this out in the lab environment.

Code Quality Metrics (Labs: Code Quality Metrics / Code Coverage Metrics)

One of the typical things companies deploy on Jenkins is continuous quality metrics monitoring. We'll look into both static code analysis and dynamic code analysis and spend some time implementing them in the lab, as well as discuss how to practically deploy continuous quality metrics monitoring in the real world.

Command-Line Interface (Lab: Command-Line Interface)

Jenkins administrators need to automate various maintenance tasks on Jenkins. We'll look into those, with the main focus on command-line interface. The instructor will also touch on other mechanisms, such as the REST API and the Groovy console.

Access Control (Lab: Access Control)

Production Jenkins installations almost always have to be secured with an integration to the corporate backend identity service. We'll look at this part of Jenkins and learn how to implement access control in Jenkins.



Mastering Continuous Integration with Jenkins

Build Promotion (Lab: Build Promotion)

As automation spreads across teams, you'll want to automate workflows that span multiple teams. We'll look into how to create such a workflow. The instructor will also discuss several other emerging plugins in this space.

Parameterized Builds (Lab: Parameterized Builds)

The parameterized build feature in Jenkins is a key building block in creating more complex workflows in Jenkins. We'll look at how this feature works and experiment with it in the lab.

Automated Deployment (Lab: Automated Deployment)

With our eyes on continuous deployment, we'll look at how to make Jenkins deploy to Java application servers, using Tomcat as an example.

Distributed Builds (Lab: Distributed Builds)

A production Jenkins instance almost always needs multiple computers to spread the workload. We'll discuss the basics of distributed builds, mechanisms to utilize slaves effectively and various features Jenkins provides to generally simplify management.

CloudBees Jenkins Enterprise – Folders and Folders Plus Plugins (Lab: Job Organization Using Folders)

We'll revisit how to organize a large number of Jenkins jobs with the Folder mechanism in CloudBees Jenkins Enterprise and how this maps naturally to typical organizational needs.

CloudBees Jenkins Enterprise – Role-based Access Control (RBAC) Plugin (Lab: Secure Jenkins)

The RBAC and Folders plugins work well together to enable sophisticated yet simple access control mechanisms for large organizations. We'll look at the basics of RBAC and how it works.



Mastering Continuous Integration with Jenkins

CloudBees Jenkins Enterprise – Validated Merge Plugin (Lab: Validated Merge with Git)

The Validated Merge feature in CloudBees Jenkins Enterprise allows developers to run tests entirely on the server, while ensuring that no bad commits land on the team repository. This module shows how to implement Validated Merge.

CloudBees Jenkins Enterprise – Templates Plugin (Labs: Builder Template / Job Template)

We'll discuss how the Templates feature in CloudBees Jenkins Enterprise allows administrators to manage a large number of similar jobs and builds effectively, as well as hides the complexity of the common build script.

CloudBees Jenkins Enterprise High Availability (HA)

We'll see how the active/standby HA of CloudBees Jenkins Enterprise works and how to administer this feature.

CloudBees Jenkins Enterprise – Custom Update Center Plugin

In larger organizations, often multiple Jenkins masters are deployed. We'll see how the Custom Update Center plugin enables delivery of in-house plugins, as well as ensures uniformity in those installations.

CloudBees Jenkins Enterprise and CloudBees Jenkins Operations Center are built on top of open source Jenkins, an independent community project. Read more at: www.cloudbees.com/jenkins/about

© 2015 CloudBees, Inc. CloudBees is a registered trademark and CloudBees Jenkins Enterprise, CloudBees Jenkins Operations Center and DEV@cloud are trademarks of CloudBees, Inc. Other product or brand names may be trademarks or registered trademarks of their respective holders.

0315v00