

WORKSHOP

Jenkins Training Advanced

Praxisorientiertes Training für professionelle
Anwender



Überblick

Jenkins ist die Nummer 1 unter den Continuous Integration Servern. Wenn auch mit einer geringen Lernkurve schon eine gut funktionierende CI-Umgebung aufzubauen ist, so vermittelt dieser Workshop eine Menge Know-how für den professionellen Einsatz und Betrieb von Jenkins. Das Werkzeug lässt sich mit dem im Training gewonnenen Wissen hinsichtlich der Automatisierungsmöglichkeiten im Projekt optimal ausschöpfen.



Inhalte

Refresh CI Basics

//

Wir rekapitulieren die grundlegenden Vorteile von Continuous Integration und warum wir es wie im Projekt einsetzen.

Einführung Jenkins CI (Lab: Installieren der Lab-Umgebung)

//

Wir führen Jenkins als Werkzeug für die Automatisierung ein, schauen uns im Detail an, welche Voraussetzungen für eine Produktivumgebung zu beachten sind und setzen diese auf.

Jenkins Build Jobs (Lab: Erstellen eines Freestyle / Maven Build Job)

//

Wir lernen den Freestyle und den Maven Job Type kennen. Die wichtigsten Teile des UI und der Job-Konfiguration werden erklärt, sowie die zusätzlichen Features des Maven Job Type diskutiert.

Monitoring von Build Jobs (Lab: Anzeigen von Test Results)

//

Weitere Teile des Jenkins UI werden detailliert diskutiert, wir werfen einen Blick auf den stereotypen Continuous Integration Workflow und sehen wie Test Reports von Jenkins ausgeführt werden.

Format

Modularer Workshop

Sehr flexibel und interaktiv

Viele praktische Übungen

40 % Theorie, 60 % Praxis

Dauer: 2 Tage

Für Java Entwickler,
Software Architekten,
Technische Projektmanager,
Build Manager, QA
Manager

Organisieren von Build Jobs (Lab: Einen Integration Test Build aufsetzen)

//

Mit dem Wachsen einer Jenkins-Instanz wächst auch die Herausforderung, die steigende Zahl von Build Jobs gut zu organisieren. Uns interessieren weitere Teile des Jenkins UI, wir schauen auf den stereotypen Continuous Integration Workflow und sehen wie Test Reports von Jenkins ausgeführt werden.

Jenkins Plugins

//

Jenkins erfährt einen Großteil seiner Power durch das große und ständig wachsende Angebot an Plugins, inzwischen weit über 1.000. Wie lässt sich dieses Plugin-Ökosystem effizient nutzen? Welche Plugins sind absolut sinnvoll? etc.

Zusammenspiel mit Version Control (Lab: Integration mit einem VCS

Browser)

//

Wir schauen uns den Wert der Integration von Jenkins mit einem Repository Browser an und testen die Integration mit Subversion oder Git exemplarisch in der LAB-Umgebung aus.

Code Quality Metriken (Lab: Code Quality Metrics / Code Coverage Metrics)

//

Unternehmen nutzen Jenkins typischerweise auch dazu, die kontinuierliche Source Code Qualität zu messen. Wir beschäftigen uns sowohl mit der statischen als auch der dynamischen Source Code Analyse, implementieren sie als Lab-Aufgabe und diskutieren die Anwendung in der Praxis.

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

//

Jenkins Administratoren müssen eine Reihe von Wartungsaufgaben automatisieren. Im Fokus unserer Betrachtung liegt das Command-Line Interface. Zudem streifen wir andere Interfaces wie die REST API und die Groovy Console.

Access Control (Lab: Access Control)

//

Produktive Jenkins-Installationen müssen meist durch die Einbindung in einen zentralen Authentifizierungs-Service abgesichert werden. Wir lernen Möglichkeiten kennen, die Jenkins hier bietet und nehmen im LAB beispielhaft die Implementierung vor.

Build Promotion (Lab: Build Promotion)

//

Automatisierte Arbeitsabläufe sind oftmals über mehrere Teams verteilt. Wir sehen uns die Automatisierung teamübergreifender Workflows und entsprechende Plugins für diese Aufgaben genauer an.

Parametrisierte Builds (Lab: Parametrisierte Builds)

//

Das „Parametrisierte Builds“-Feature ist ein wichtiger Baustein, um komplexere Arbeitsabläufe in Jenkins zu erstellen. Wir sehen, wie dieses Feature funktioniert und experimentieren damit im Lab.

Automatisiertes Deployment (Lab: Automatisiertes Deployment)

//

Mit Blick auf Continuous Deployment interessiert uns, wie mit Jenkins das automatische Deployment auf Java Application Server funktioniert und nutzen hier Tomcat als Beispiel.

Distributed Builds (Lab: Distributed Builds)

//

Eine produktive Jenkins-Instanz braucht oftmals mehrere Server zur Verteilung der Last. Wir besprechen die Grundlagen verteilter Builds, wie man Build Slaves verwendet und das Management eines verteilten Setups möglichst einfach gestaltet.

Pipeline Plugin (vormals Workflow Plugin)

//

Mit dem Pipeline Plugin lassen sich Continuous Delivery Pipelines aufbauen und komplexere Build-/Test- und Delivery-Workflows über die verschiedenen Stages gut steuern. Wir schauen uns an, wie man mit der betreffenden Workflow DSL und Standardbausteinen zum Ziel kommt.

Multibranch Workflows

//

Dynamisch und automatisiert neue Jobs für neue Entwicklungslinien aufsetzen, das geht mit Workflow Multibranch. Wir schauen uns praktisch das Setup mit dem Workflow-Description-File „Jenkinsfile“ an.

Inhalt CloudBees Jenkins Enterprise

Folders und Folders Plus Plugin (Lab: Job Management mit Foldern)

//

Wir lernen wie sich eine große Anzahl von Jenkins Jobs mittels Foldern gut organisieren lässt, um damit die typischen Unternehmensanforderungen abzudecken.

Role Based Access Control (RBAC) Plugin (Lab: Jenkins absichern)

//

Das RBAC und das Folder Plugin arbeiten gut zusammen, um eine ausgefeilte Zugriffskontrolle, wie für größere Organisationen erforderlich, zu ermöglichen. Wir sehen uns die Grundlagen des RBAC im Detail an und vollziehen die Funktionsweise im Lab nach.

Validated Merge Plugin (Lab: Validated Merge mit Git)

//

Das Validated Merge-Feature ermöglicht es Entwicklern, Tests vollständig auf dem Server auszuführen, um sicher zu stellen, dass keine Bad Commits im Team Repository landen. In diesem Trainingsmodul widmen wir uns der Implementierung von Validated Merge.

Templates Plugin (Lab: Builder Template/ Job Template)

//

Wir diskutieren, wie das Template-Feature durch Jenkins-Admins genutzt werden kann, um eine große Anzahl ähnlicher Jobs und Builds effektiv managen und die Komplexität umfangreicher Build Scripts steuern zu können.

Jenkins High Availability (HA)

//

In diesem Modul blicken wir auf das Active/Standby HA Feature von CloudBees und seine Administration.

Custom Update Center Plugin

//

In größeren Organisationen sind oft mehrere Jenkins Master im Einsatz. Wir erfahren, wie Custom Update Center die Lieferung von in-house-Plugins ermöglichen und so die Einheitlichkeit und Durchgängigkeit der Installationen im Unternehmen gewährleisten können.

Stage View Plugin für Pipelines

//

Mit der Visualisierung von Continuous Delivery Pipelines über Stage View lassen sich Fehler wesentlich leichter lokalisieren, manuelle Freigaben und Checkpoints für das Wiederaufsetzen im Prozess an geeigneter Stelle realisieren. Dieses Modul führt detailliert in Stage View mit praktischer Anwendung ein.

