

Large-Scale Development (LSD)

Assignment 5: Der Build-Server *Jenkins*

Autoren: Felix Hefner, Max Jando, Severin Kohler

Stand: 12. November 2017

1 Einleitung

Im Rahmen der Vorlesung LSD (Large-Scale-Development) sollten die Autoren dieses Dokuments den Opensource-Build-Server *Jenkins* aufsetzen und eine Build-Pipeline anlegen, welche den in vorherigen Assignments bereits behandelten Tomcat 6.0.5 baut. Dieses Dokument ist in folgende Teile untergliedert:

1. Installation von Jenkins auf einem Ubuntu-Server
2. Konfiguration des Build-Jobs in Jenkins
 1. Erstellung eines Jenkinsfile
 2. Weitere Konfiguration auf der Weboberfläche von Jenkins

2 Installation von Jenkins auf einem Ubuntu-Server

Da das Ubuntu Server-Betriebssystem mit der Paketverwaltung **apt-get** darherkommt, konnte diese auf relativ simple Weise dazu benutzt werden, das Jenkins-Paket zu installieren. Jedoch war dieses nicht in den Standard-Paketquellen zu finden, sodass eine spezielle von den Entwicklern der Software hinzugefügt werden musste. Genauer wurde wie folgt vorgegangen:

1. Manuelles Hinzufügen des Publickeys der Jenkins.io - Server, damit diesen vertraut wird
2. Hinzufügen der Jenkins-Paketquelle durch

```
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable  
binary/ > /etc/apt/sources.list.d/jenkins.list '
```

3. Aktualisieren der Paketquellen sowie Installation des Pakets durch

```
sudo apt-get update ; sudo apt-get install jenkins
```

4. Da der Paketmanager nach der Installation von selbst `service start jenkins` aufruft, läuft Jenkins ab sofort unter `http://<IP-des-Servers>:8080`.

3 Konfiguration des Build-Jobs in Jenkins

Die Build-Jobs, die in Jenkins zum Kompilieren des Java-Codes, zum Ausführen der Tests sowie zum Deployen des fertigen `.jar`-Archivs wurden primär über sogenannte *Pipelines* anhand von *Jenkinsfiles* erstellt. Hierbei handelt es sich um Scripte, welche in der Sprache Groovy geschrieben werden und sich in Stages unterteilen. Dies sind Abschnitte, welche später auch in der Weboberfläche von Jenkins sichtbar werden. Der Inhalt des **Jenkinsfile** wurde zunächst direkt in der Weboberfläche eingetragen, nach dem ersten Build-Durchlauf konnte jedoch diese Datei aus dem Github-Repository heruntergeladen werden und anschließend von dort verwendet werden.

3.1 Erstellung eines Jenkinsfile

Das folgende Listing zeigt das vollständige Jenkinsfile, welches für dieses Projekt benutzt wurde. Im Anschluss werden die einzelnen Schritte erklärt.

Listing 1: Jenkinsfile zum Bauen, Testen und Verbreiten von Tomcat mit Maven

```
node {  
    //-----  
    stage name: 'clean '  
    //-----  
5 checkout scm  
    env.JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"  
    env.PATH="${env.JAVA_HOME}/bin:${env.PATH}"  
    sh "cd 'tomcat' ; mvn 'clean'"  
    //-----  
10 stage name: 'compile '  
    //-----  
    sh "pwd"  
    sh "cd 'tomcat' ; mvn 'compile'"  
    //-----  
15 stage name: 'test '  
    //-----  
    sh "cd 'tomcat' ; mvn test"  
    //-----  
    stage name: 'assembly '  
20 //-----  
    sh "cd 'tomcat' ; mvn assembly:single"  
}
```