

Open Source-basierte Softwareentwicklung

Praktikum 9 - Continuous Integration

Szenario

In dieser Woche lernen Sie die Verwendung des CI-Werkzeugs »Jenkins« in Kombination mit Git und Eclipse.

Arbeitsschritte

Schritt 1

Gehen Sie auf die Seite <https://kube.informatik.hs-furtwangen.de/jenkins> und melden Sie sich mit Ihrem HFU-Account an.

Schritt 2

Erstellen Sie ein neues Projekt, welches (aus Gründen der Eindeutigkeit) mit Ihrem Namenskürzel anfängt. Wählen Sie die Option »Maven Projekt bauen« aus.

Schritt 3

In der Formularmaske wählen Sie unter Source-Code-Management »Git« aus und geben als URL die HTTPS-URL Ihres Gitlab-Projekts an. Die URL können Sie unter <https://gitlab.informatik.hs-furtwangen.de> nachschlagen.

Als Build-Auslöser sollte »Build when a change is pushed to GitLab« ausgewählt werden. Die Webhook URL müssen Sie sich kopieren, sie wird im nächsten Schritt benötigt. Unter Build/Goals und Optionen geben Sie »install« ein.

Schritt 4

Fügen Sie die Webhook URL aus Schritt 3 in Gitlab unter Settings/Integrations/URL ein und drücken Sie »Add Webhook«. Zum testen, ob die Integration funktioniert, sollten Sie auf derselben Seite in Gitlab bei Ihrem neu eingefügten Webhook Test/Push Events auswählen.

Das Projekt wird jetzt (erfolgreich) in Jenkins gebaut. Nachdem der Build durchgelaufen ist, schauen Sie sich bitte in Jenkins die Konsolenausgabe und die Testergebnisse an.

Schritt 5

Ändern Sie Ihre Quelldateien in Eclipse an einer beliebigen Stelle, sodass ein Unit-Test fehlt schlägt. Committen und Pushen Sie die Änderung zum Gitlab-Repository. Aufgrund der

Integration von Gitlab und Jenkins wird der Webhook ausgeführt, der Jenkins anweist, alle Projekte des geänderten Git-Repositories neu zu bauen (im Praktikum ist das dann nur Ihr Projekt). Das Bauen schlägt aufgrund des fehlerhaften Unit-Tests fehl.

Schritt 6

Korrigieren Sie Ihren Unit-Test, sodass in Eclipse keine Test-Fehler mehr angezeigt werden. Committed und Pushen Sie die Änderungen, kurze Zeit später zeigt auch Ihr Jenkins-Projekt keine Fehler mehr an.

Schritt 7

Damit Sie nicht bei jedem Commit auf der Webseite nach dem Ergebnis eines Builds suchen müssen, sollten Sie eine E-Mail-Benachrichtigung aktivieren. Gehen Sie in die Konfiguration Ihres Projekts und fügen Sie unter Post-Build-Aktionen eine »Editable Email Notification« hinzu. Damit Sie bei allen Build-Aktionen eine E-Mail erhalten, gehen Sie in die »Advanced Settings...«, löschen Sie den Failure-Trigger und fügen einen neuen Trigger »Always« hinzu. Speichern Sie die ganze Konfiguration, erzeugen und pushen Sie wieder einen Commit. Kurze Zeit später sollten Sie eine E-Mail von Jenkins bekommen.