



Continuous Integration mit Hudson/Jenkins

Agenda

- 1. Was ist Continuous Integration**
- 2. Hudson vs. Jenkins**
- 3. Einrichten der Umgebung**
- 4. APEX & Continuous Integration**

- 5. Hands On**



Was ist Continuous Integration

Continuous Integration

- Konzepte seit dem Jahre 2000 durch Martin Fowler vorgestellt
- Im eXtreme Programming (XP) ist Continuous Integration ein fester Bestandteil
- Im agilen Umfeld gehört die gehört Continuous Integration zwangsläufig dazu

Continuous Integration

- „Prozess in der Software Entwicklung fortlaufend die Einzelkomponenten zu einer Anwendung zusammenzuführen“
- „Kontinuierliche Steigerung der Softwarequalität durch häufige Integration“
- „Automatisierung wiederholender Arbeitsschritte“
- „Spiegelung und Vervielfältigung der Umgebungen“
- „Einfacher Zugriff“

Was ist Continuous Integration

- Grundlage ist eine gemeinsame Codebasis
- starke Verbreitung innerhalb von Java Projekten oder anderen Projekten wo regelmäßig eine Software „zusammengebaut“ wird
- Wiederholende Tests (auch ohne menschliche Aktion)
- Automatisierte Rückmeldung über Erfolg oder Misserfolg
- Direktes Erzeugen eines Builds nach einchecken möglich

Continuous Integration - Vorteile

- Fehler werden häufiger und schneller entdeckt
- Zeitersparnis durch Automatisierung
- Reduzierung des „Merge“ Aufwands
- Keine Fehler durch händische Arbeitsschritte
- Tester können autark arbeiten (ohne technisches Wissen)
- Arbeitsschritte werden transparenter und durch einzelne Jobs/Prozesse abgebildet

Continuous Integration - Nachteile

- Einarbeitung in die Thematik
- Aufbau einer Umgebung nötig
- Skripte müssen parametrisiert werden
- Versionierung bzw. zentrale Codebasis nötig
- Teils umständliche oder neue Skripte nötig (batch, ...)

Continuous Integration - Software

- Open Source Software

- Jenkins - <https://jenkins-ci.org/>
- Hudson - <http://hudson-ci.org/>
- CruiseControl - <http://cruisecontrol.sourceforge.net/>
- Continuum - <https://continuum.apache.org/>

- Kommerzielle Software

- Atlassian Bamboo - <https://de.atlassian.com/software/bamboo>
- JetBrains Teamcity - <https://www.jetbrains.com/teamcity/>



Hudson vs. Jenkins

Hudson

- Open Source Projekt
- Mischlizenz unter Commons Creative und MIT-Lizenz
- Support für
 - Code-Repositories / Versionierungssysteme (CVS, SVN, Perforce)
 - Build Tools (ANT, Maven/Maven2, CommandLine)
 - unterschiedliche Java JDK's ab Version 5
 - Benachrichtungen über E-Mail, RSS, Google Calender, IRC, ...
 - Reports über Plugins
- Konfiguration über Weboberfläche
- Live-Build-Überwachung

Hudson vs. Jenkins

- Jenkins als auch Hudson stehen für den Ausdruck „Buttler“
- Mit der Übernahme von Sun durch Oracle kam es zu Streitigkeiten mit dem bisherigen Entwicklerteam
<http://kohsuke.org/2011/01/11/bye-bye-hudson-hello-jenkins/>
 - damit wurde Jenkins ins Leben gerufen

Jenkins

- Auch unter Creative Commons Lizenz
- Wird über Betriebssystem spezifische Installation installiert
 - Windows, Ubuntu, Debian, Red Hat, Fedora, CentOS, Mac OS X, openSUSE, FreeBSD, OpenBSD, Gentoo
 - Docker Image (docker pull jenkinsci/jenkins)
 - Lässt sich auch als war-File starten bzw. deployen
- Basiert im Kern noch auf Hudson
- Mehr Plugins (1079) als Hudson
- Läuft initial per Default auf Port 8080 (jenkins.xml anpassbar)
- erstellt Windows Dienste automatisch



Hudson einrichten

Hudson - Installation

- Letzte Version von Hudson unter <http://hudson-ci.org/> herunterladen (aktuell 3.3.0)
- Danach
 - Als Standalone mittels `java -jar hudson.war` starten
- Oder
 - Auf Apache Tomcat (oder anderem Application Server) deployen
- Hudson ist standardmäßig auf <http://localhost:8080> erreichbar.
 - Achtung bei Konflikt mit ORDS oder anderen Systemen/bzw. Ports

Hudson - Initialsetup

- Initialsetup
 - Auswahl Hudson Subversion Plug-in
 - Batch Task Plugin
 - Hudson Pipeline Plugin
 - Hudson build timeout plugin
 - Compact Columns
 - Dashboard View

Hudson - Initialsetup

- Initialsetup
 - Hudson disk-usage plugin
 - Hudson Email Extension Plugin
 - FTP publisher plugin
 - Hudson Global Build Stats Plugin
 - Log Parser Plugin

Hudson - Konfiguration

- Der Wizard stellt 2 Projekte zur Auswahl
 - „Free Style“
 - Ist für eigens definierte Projekte
 - Maven 2/3 Projekt bauen
 - POM.xml enthalten, ist für Maven Projekte geeignet
 - Multikonfigurationsprojekt bauen
 - Externen Job überwachen

Hudson - Parameter

























- Umgebungsparameter
 - Allgemeine Parameter
 - Tag D:\svn\fpp\tags
 - Trunk D:\svn\fpp\trunk
 - JDK
 - JAVA_HOME D:\Program Files\Java\jdk1.7.0_51
 - Global SVN Publisher Settings
 - User & Passwort mit technischem User setzen
 - E-Mail Benachrichtigung
 - SMTP-Server, Standardendung, Email-Adressen und Hudson URL füllen

Hudson - Job

- Ein Prozess wird in Hudson als Job angelegt
- Mehrere Jobs können zu einer Job-Chain zusammengefasst werden
- Ein Job ...
 - kann die Ausführung einer Batchfile sein, einer Operation wie svn update oder eines beliebige kleinstmöglichen Schrittes
 - kann Parameter besitzen
 - kann Benachrichtigungen versenden
 - kann fehlschlagen oder erfolgreich sein
 - hat eine Konsolenausgabe, die im Browser dargestellt wird
 - Besitzt Informationen zu den letzten Ausführungen

Hudson - View

- Ein View fasst logisch mehrere Jobs zusammen

S	W	Job ↓	Letzter Erfolg	Letzter Fehlschlag	Letzte Dauer	Console
		UIT_010_update_tag	11 Stunden (#2043)	Unbekannt	0,59 Sekunden	 
		UIT_020_start_batch_patch_phase_1	11 Stunden (#1501)	6 Tage 4 Stunden (#1495)	11 Sekunden	 
		UIT_023_start_batch_patch_phase_2	10 Stunden (#1453)	6 Tage 4 Stunden (#1446)	2 Minuten 53 Sekunden	 
		UIT_030_export_apex_application	10 Stunden (#500)	Unbekannt	4,8 Sekunden	 
		UIT_030_import_apex_application	10 Stunden (#545)	Unbekannt	53 Sekunden	 
		UIT_040_zip_File_fpp	1 Monat 23 Tage (#142)	Unbekannt	12 Sekunden	 



APEX & Continuous Integration

Continuous Integration Prozesse für APEX

- User/Schema/Tablespace
 - Löschen
 - Erstellen
- Anwendung
 - Export
 - Aufsplitten
 - Versionieren
 - Import
- Statische Dateien
 - Export
 - Versionieren
 - Import
- Installation
 - Update Tag
 - Erstellung eines Installationsscripts
 - Durchführen einer Installation
 - Grants & Synonyme
- Auslieferung & neue Version
 - Neue Version erstellen
 - ZIP-Datei erstellen
- Dump
 - Import
 - Export

Continuous Integration - Parameter

Allgemeine Parameter

- Umgebung
 - Entwicklung, Test, Hotfix-Entwicklung, Hotfix-Test, Prod
- Version
 - Enthält alle bisher entwickelten Versionen
 - Versionen liegen als /tags/-Ordner vor

Continuous Integration - Parameter

Account-Data

- Parameter pro Umgebung werden benötigt
 - Benutzer, Passwörter (ggf. von mehreren Schemata)

 - Tablespaceverzeichnis
 - D:\oracle\oradata\orcl

 - Tablespacennamen
 - user_data, user_idx, user_files

 - TNSNames

Continuous Integration - Job

User/Schema & Tablespace

- Parameter

- Umgebung

- Inhalt

```
REM Anlegen der User und Tablespaces in der vorgebenen Umgebung  
REM set account = "Account_"%Umgebung%.txt  
cd %svn_trunk%  
batch\create_users_tablespaces batch\account_dba_%Umgebung%.txt  
:ende  
REM ENDE
```

- Regular Express: ORA-

- Ggf. Sicherheitsabfrage bei kritischen Builds wie Drop User & Tablespaces nutzen

- Emailbenachrichtigung falls Build fehlschlägt füllen

Continuous Integration - Anwendung

Anwendung exportieren und aufsplitten

- Parameter
 - Umgebung (host:port:sid) – DBUser – DBPassword - Application ID
- Inhalt - Export
 - Exportaufruf von Folien „Anwendungsexport“ aus Session Installationsprozess für APEX Anwendungen
- Inhalt – Aufsplitten
 - Aufruf von Folien „Anwendungsexport“

Continuous Integration - Anwendung

Anwendung versionieren und importieren

- Inhalt Import
 - Folien „Anwendungsimport“
- Inhalt versionieren

```
svn add %svn_tag%\%version%\Anwendung\F%APP_ID%.sql -force  
svn commit -m „Kommentar beim Commit“ %svn_tag%\%version%\Anwendung
```

- Kommentar beim Commit zum Beispiel:
Automatischer Commit der Version %version% der Anwendung über Hudson



Hands On

Hands On – Tomcat installieren

Tomcat

- <https://tomcat.apache.org/download-80.cgi>
 - [32-bit/64-bit Windows Service Installer](#) (herunterladen)
- Port 8081 verwenden
- Dienst starten & stoppen

Hands On – Hudson deployen

Hudson

- <http://hudson-ci.org/>
 - Latest production Release WAR file: 3.3.0 (herunterladen)

- Unter <http://localhost:8081> (Tomcat)
 - bei „Manager App“
 - WAR File to deploy „deployen“

Hands On – Hudson konfigurieren

Hudson

- Dann unter <http://localhost:8081/hudson-3.3.0/> konfigurieren
- Plugins
 - Hudson Email Extension Plugin
 - Text-finder Plugin
- Hudson verwalten
 - System konfigurieren
 - Globale Eigenschaften
 - Name script_path
 - Wert c:\mein\verzeichnis

Hands On – Hudson konfigurieren

Hudson

- Dann unter <http://localhost:8081/hudson-3.3.0/> konfigurieren
- Hudson verwalten
 - System konfigurieren
 - E-Mail Benachrichtigung (optional)
 - Smtп-Server mein.smtp.com
 - Standardendung für E-Mail-Adressen @meine.domain
 - E-Mail-Adresse des Systemadmin meine.adresse@meine.domain
 - Hudson URL http://localhost:8081/hudson%2D3.3.0/
 - Rechts oben „AUTO-AKTUALISIERUNG EINSCHALTEN“ drücken

Hands On – Export APEX Application

Jobs einrichten

- Neuen „Free Style“ Job „01 Export APEX Application“ einrichten
 - Parameter
 - db default localhost:1521:orcl
 - db_user default expertenseminar_2015
 - db_password default expertenseminar_20151
 - apex_app_id default 100
 - Build Schritt hinzufügen
 - Windows Batch Datei ausführen – Kommando:

```
d:  
cd %script_path%  
call export_apex_app.cmd %db% %db_user% %db_password% %apex_app_id%
```

Hands On – Export APEX Application

Jobs einrichten

- Job „01 Export APEX Application“ erweitern
 - Post-Build-Aktionen
 - Hudson Text Finder
 - Regular expression ORA-
 - E-Mail-Benachrichtigung
 - Empfänger: Meine.Email@domain.com

Hands On – Export APEX Application

Batchfile einrichten – export_apex_app.cmd

```
@echo off
Rem Pruefen ob alle Parameter gefuellt sind
if %1!==! goto usage
if %2!==! goto usage
if %3!==! goto usage
if %4!==! goto usage

call export_apex_app.bat -db %1 -user %2 -password %3 -applicationid %4

goto ende
:usage
echo "fehlerhafte Parameter! Korrekte Parameter sind: <SourceTNS> <SourceUser> <SourcePw>
<APP_ID>"
echo "Beispiel: export.cmd host:port:sid dbuser dbpw 100"
:ende
```

Hands On – Export APEX Application

Batchfile einrichten – export_apex_app.bat

```
@echo off
set ORACLE_HOME=C:\oracle\product\11.2.0\dbhome_1
REM APEX HOME auf das Verzeichnis setzen, in das APEX ausgepackt wurde.
set APEX_HOME=D:\Eigene Dateien\APEX\install\apex_4.2.6\apex

set CLASSPATH=%ORACLE_HOME%\oui\jlib\classes12.jar
set CLASSPATH=%CLASSPATH%;%APEX_HOME%\utilities;%ORACLE_HOME%\jdbc\lib\ojdbc6.jar
set CLASSPATH=%CLASSPATH%;

java.exe oracle.apex.APEXExport %*
```

Hands On – Split APEX Application

Jobs einrichten

- Neuen „Free Style“ Job „02 Split APEX Application“ einrichten

- Parameter

- apex_app_id default 100

- Build-Schritt hinzufügen

```
d:  
cd %script_path%  
splitter.bat f%apex_app_id%.sql
```

- Post-Build-Aktionen

- Hudson Text Finder

- Regular expression Could not process

- E-Mail-Benachrichtigung

- Empfänger: Meine.Email@domain.com

Hands On – Split APEX Application

Batchfile einrichten – splitter.bat

```
@echo off
set ORACLE_HOME=C:\oracle\product\11.2.0\dbhome_1
REM APEX HOME auf das Verzeichnis setzen, in das APEX ausgepackt wurde.
set APEX_HOME=D:\Eigene Dateien\APEX\install\apex_5.0.1\apex

set CLASSPATH=%ORACLE_HOME%\oui\jlib\classes12.jar
set CLASSPATH=%CLASSPATH%;%APEX_HOME%\utilities;%ORACLE_HOME%\jdbc\lib\ojdbc6.jar
set CLASSPATH=%CLASSPATH%;

java.exe oracle.apex.APEXExportSplitter %*
```

Hands On – Import APEX Application

Jobs einrichten

- Neuen „Free Style“ Job „02 Export APEX Application“ einrichten

- Parameter

- TNSNames default ORCL
- db_user default expertenseminar_2015
- db_password default expertenseminar_20151
- apex_app_id default 100

- Build Schritt hinzufügen

- Windows Batch Datei ausführen – Kommando:

```
d:
```

```
cd %script_path%
```

```
call import_apex_app.cmd %TNSNames% %db_user% %db_password% %apex_a
```


Hands On – Export APEX Application

Jobs einrichten

- Job „03 Import APEX Application“ erweitern
 - Post-Build-Aktionen
 - Hudson Text Finder
 - Regular expression SP2-0310
 - E-Mail-Benachrichtigung
 - Empfänger: Meine.Email@domain.com

Hands On – Import APEX Application

Batchfile einrichten – import_apex_app.cmd

```
@echo off
Rem Pruefen ob alle Parameter gefuellt sind
if %1!==! goto usage
if %2!==! goto usage
if %3!==! goto usage
if %4!==! goto usage

echo sqlplus -L %2/%3@%1 @import_apex_app.sql f%4 %2
sqlplus -L %2/%3@%1 @import_apex_app.sql f%4 %2

goto ende
:usage
echo "fehlerhafte Parameter! Korrekte Parameter sind: <SourceTNS> <SourceUser> <SourcePw>
<APP_ID>"
echo "Beispiel: export.cmd host:port:sid dbuser dbpw 100 d:\verzeichnis"
:ende
```

Hands On – Import APEX Application

SQL Script einrichten – import_apex_app.sql

```
declare
  l_workspace_id pls_integer;
  l_schema_and_workspace varchar2(30) := upper('&2');
begin
  -- Ermitteln des zugehoerigen Workspaces
  select workspace_id into l_workspace_id
  from apex_workspaces
  where workspace = l_schema_and_workspace;

  -- Anwendungsparameter für neue zu importierende Anwendung setzen
  apex_application_install.set_workspace_id(p_workspace_id => l_workspace_id);
  apex_application_install.generate_application_id;
  apex_application_install.generate_offset;
  apex_application_install.set_schema(p_schema => l_schema_and_workspace);
  apex_application_install.set_application_alias(p_application_alias => 'F' || apex_application_install.get_application_id);
end;
/
@&1
```



Vielen Dank.

MT AG

Balcke-Dürr-Allee 9
40882 Ratingen

Telefon: +49 (0) 21 02 309 61-0
Telefax: +49 (0) 21 02 309 61-101

E-Mail: info@mt-ag.com
www.mt-ag.com