



Continuous Deployment mit Rex und Jenkins

Miriam Bergmann
IT Engineering & Operations
Berlin, 23.05.2013

Agenda

... was erwartet euch?



1. About Me
2. Continuous Deployment
3. Rex
4. Jenkins
5. Kleine Demo
6. Fragen

About Me

...wer steht da denn?



- ▶ Miriam Bergmann
- ▶ Angestellt bei der inovex GmbH als Systems Engineer
- ▶ Aufbau und Betrieb komplexer Systemlandschaften und Applikationen
- ▶ Automatisierung von Betriebsabläufen wie Softwaredeployments

- ▶ Anforderungen des Kunden / Projektes
 - ▶ Agile Releasezyklen erfordern viele kleine atomare Deployments
- ▶ Realität sieht häufig anders aus:
 - ▶ Probleme beim Deployment
 - ▶ Deployment auf Produktion funktioniert nicht
 - ▶ Nicht reproduzierbare Fehler
 - ▶ „works on my box“
 - ▶ komplexe Abhängigkeiten zwischen den Komponenten
 - ▶ Verschiedene Applikationen, verschiedene Deploymentmechanismen – Deployment als Wissenschaft für sich
 - ▶ Komplexität führt zu seltenen zeitintensiven Deployments

Continuous Deployment

... was kann ich mir darunter vorstellen?



- ▶ Einsatz vom einheitlichen Tooling zum Deployment
 - ▶ Für alle Applikationen
 - ▶ Auf allen Umgebungen
- ▶ Deployment als Teil des Softwarereleaseprozesses
 - ▶ Tests des Deploymentprozesses

Continuous Deployment

... wie stelle ich das am Besten an?



- ▶ Idee ist nicht neu
- ▶ Es gibt nicht nur eine Umsetzungsmöglichkeit
- ▶ Wahl der Tools und der Umsetzung abhängig von verschiedenen Faktoren
 - ▶ Skills und Vorlieben des betreuenden Teams
 - ▶ Art der Applikation
 - ▶ Grad der Komplexität
- ▶ Prozess forcieren und aktiv leben

Continuous Deployment

... was haben wir daraus gemacht?



- ▶ Rex als Deployment Tool
- ▶ Jenkins zum Abbilden des Prozesses (Berechtigungen, Freigaben, Durchlauf der Softwarepakete durch die verschiedenen Umgebungen)

Rex

... wer oder was ist das?

- ▶ Steht für Remote Execution
- ▶ <http://rexify.org/>
- ▶ Implementiert in Perl
- ▶ over ssh
- ▶ Configmanagement und Deployments

- ▶ Perl ist unter Admins recht verbreitet
- ▶ Rex erfordert nicht viele Perlkenntnisse
- ▶ ein File
- ▶ zentral
- ▶ sequentielle Abarbeitung
- ▶ Aufteilung in sogenannte Tasks für die Übersichtlichkeit

- ▶ Aufbau eines Repos auf dem zentralen Deploymenthost:

```
root@vm01:/opt$ tree
├── deployment
│   ├── config
│   │   ├── live
│   │   │   ├── testapp
│   │   ├── stage
│   │   │   ├── testapp
│   │   ├── test
│   │   │   ├── testapp
│   ├── keystore
│   │   ├── live
│   │   │   ├── deploy
│   │   │   │   ├── deploy
│   │   │   │   └── deploy.pub
│   │   │   └── rex_conf
│   │   ├── stage
│   │   │   ├── deploy
│   │   │   │   ├── deploy
│   │   │   │   └── deploy.pub
│   │   │   └── rex_conf
│   │   └── test
│   │       ├── deploy
│   │       │   ├── deploy
│   │       │   └── deploy.pub
│   │       └── rex_conf
│   ├── packages
│   │   ├── live
│   │   │   ├── testapp
│   │   ├── stage
│   │   │   ├── testapp
│   │   ├── test
│   │   │   ├── testapp
│   └── rex
│       ├── testapp
│       └── Rexfile
```

- ▶ Nur ssh Zugriff auf den zu deployenden Hosts nötig
- ▶ Definition von Hostgruppen
- ▶ Basisconfig:

```
###  
# Configfile für Rex - Testumgebung  
##  
  
## Der Deploymentuser  
user "deploy";  
password "testtest";  
sudo_password "";  
key_auth;  
private_key "/opt/deployment/keystore/" . environment() . "/deploy/deploy";  
public_key "/opt/deployment/keystore/" . environment() . "/deploy/deploy.pub";  
  
## Hostgruppendefinition  
group fe => "fe01";  
group db => "db01";  
~
```

- ▶ Umgebungsspezifische Pfade für Configs und Packages

```
use strict;
use warnings;
use Rex::Commands::Run;
use Rex::Commands::Rsync;
use Rex::Transaction;
use Sys::Syslog qw(:macros);
use Rex::Helper::SSH2;

# =====
# Konfiguration
# =====

# Environment Configuration
my $env = environment();
my %opts = Rex::Args->getopts;

if (defined $env && length $env > 0) {
    # Repository Configuration
    my $package_path = "/opt/deployment/packages/" . environment() . "/testapp";
    my $config_path = "/opt/deployment/configs/" . environment() . "/testapp";

    # Load Hostgroup Configuration and deployment User
    require "/opt/deployment/keystore/" . environment() . "/rex_conf";
} elsif (!exists $opts{T}) {
    print "-E nicht gesetzt\n";
    die;
}
```

▶ Ein Beispieltask

```
1  
desc "connection test";  
task "testtask", sub {  
  say run "uptime";  
};  
~  
~  
~
```

- ▶ Aufruf ganz simpel:

```
rextest@vm01:/opt/deployment/rex/testapp$ rex -E stage -T
[2013-05-23 04:48:07] INFO - eval your Rexfile.
Tasks
  testtask                connection test

Server Groups
  db
  fe
rextest@vm01:/opt/deployment/rex/testapp$ |
```

```
rextest@vm01:/opt/deployment/rex/testapp$ rex -E test testtask
[2013-05-23 10:43:16] INFO - eval your Rexfile.
[2013-05-23 10:43:16] INFO - Running task testtask on fe01
[2013-05-23 10:43:16] INFO - Connecting to fe01:22 (deploy)
[2013-05-23 10:43:16] INFO - Connected to fe01, trying to authenticate.
[2013-05-23 10:43:16] INFO - Successfully authenticated on fe01.
10:43:16 up 10:01, 3 users, load average: 0.00, 0.01, 0.05
[2013-05-23 10:43:16] INFO - Exiting Rex...
[2013-05-23 10:43:16] INFO - Cleaning up...
rextest@vm01:/opt/deployment/rex/testapp$ |
```

- ▶ CI Server
- ▶ implementiert in Java
- ▶ Tool zur Jobausführung und Jobüberwachung
- ▶ Jenkinsintegration von Rex

- ▶ Konfiguration
 - ▶ Plugins
 - ▶ Jenkins Dynamic Parameter Plugin
 - ▶ Promoted Builds Plugin
 - ▶ Abbilden des Deploymentprozesses
 - ▶ Ansichten = Umgebungen
 - ▶ Jobs = Applikationen
 - ▶ Freigabeprozess
 - ▶ Rechtemanagement
 - ▶ Promote Actions
- ▶ Einbindung von Rex (Aufrufe, Konfiguration der Tasks)

Vielen Dank für die Aufmerksamkeit !

Fragen?

Vielen Dank für Ihre Aufmerksamkeit



Kontakt

Miriam Bergmann
Systems Engineer

inovex GmbH
Office Pforzheim
Karlsruher Str. 72
75179 Pforzheim

0173 3181055
miriam.bergmann@inovex.de

