

Bachelorvortrag

Automatisierte Konfiguration des Build-Servers im Saros-Projekt mit Salt und Git

AG Software Engineering
Freie Universität Berlin

Patrick Hobusch, 20.08.2015

Freie Universität



Berlin

Saros-Projekt und Build-Server

- Software für verteilte Paar-Programmierung (IDE-Plugins)
- Open-Source Projekt der AG Software Engineering
- Entwickelt vor allem von Studenten des Instituts für Informatik der Freien Universität Berlin

- Build-Server wichtige Grundlage des Entwicklungsprozesses
 - Versionsverwaltung (Git)
 - Code-Reviews (Gerrit)
 - Continuous Integration (Jenkins)
 - Statische Code-Analyse (SonarQube)
 - Nutzer-Umfragen (phpESP)
 - und weitere Dienste

- Infrastruktur über Jahre organisch gewachsen
 - Ubuntu 10.04.4 LTS
 - teilweise veraltete Dienste
 - schlechte Antwortzeiten
 - fehlende Sicherungen (Angst vor Änderungen)
 - ➔ hoher Administrationsaufwand für Saros-Projekt

- Server erfüllt nicht Anforderungen des IT-Dienstes
 - Firewall
 - Aktualisierungen
 - Integration in Infrastruktur
 - ➔ Administration durch IT-Dienst

- Überhaupt erstmal anfangen, Server neu aufzusetzen
 - Hauptaufgabe dieser Arbeit
- Zustand deklarativ beschreiben
 - Dokumentation
 - Versionsverwaltung
 - Anlehnung an Entwicklungsprozess
- System auf Knopfdruck aufsetzen / korrigieren können
- Einsatz eines Systemkonfigurationswerkzeugs
 - zunächst Puppet

Idee – Beispiel Jenkins

```

node 'hostname.example.com' {
  include jenkins
}

class { 'jenkins':
  executors => 0,
}

jenkins::job { 'test-build-job':
  config => template("${templates}/test-build-job.xml.erb"),
  ...
}

jenkins::user { 'johndoe':
  email      => 'jdoe@example.com',
  password  => 'secret',
}

```

<https://github.com/jenkinsci/puppet-jenkins/blob/master/README.md>

Umsetzung

- Anforderungen finden
 - oft erst späte Erwähnung
 - Grenze zwischen Dienstkonfiguration („Nutzdaten“) und Systemkonfiguration
 - beidseitiger Lernprozess
- Salt statt Puppet
- Einsatz eines robusteren Betriebssystems (Debian)
- Dienstbereitstellung stets über Debian-Pakete
- Verwendung von ZEDAT-Accounts zur Authentifizierung (LDAP)
 - Saros-Projekt kann Rechte-Verwaltung vornehmen

- Debian-Paket für Gerrit
 - enthält war-Datei, Init-Skript, installiert Anwendung, enthält Version, Copyright usw.
 - Fork von <https://github.com/dnaeon/gerrit-debian/>

- Salt-State für Gerrit:

```
gerrit:
  pkgrepo.managed:
    - name:      deb http://simpel.imp.fu-berlin.de/debian ...
    - file:      /etc/apt/sources.list.d/debian-imp.list
    - key_url:   https://simpel.imp.fu-berlin.de/debian/...
    - require_in:
      - pkg: gerrit
  pkg.latest:
    - refresh: True
    - watch_in:
      - service: gerrit
  service.running:
    - enable: True
```

Erledigte Arbeiten

- Exploration von Puppet und Vagrant
 - Was ist technisch überhaupt möglich?
- Bestandsaufnahme des Servers
 - Welche Bestandteile sind relevant und wie arbeiten sie?
- Anforderungsanalyse
 - Stetiger Prozess bis heute, von Saros-Projekt und IT-Dienst
- Werkzeugevaluation
 - Puppet, Salt, Docker
- Umsetzung mit Salt
 - Installationsautomatisierung für Jenkins und Gerrit
 - Skripte, States, Templates, Einsatz von Buildtools
 - Systemkonfiguration mit Paketen
- Migration

Neuer Stand der Maschine(n)

- Maschinen gehen nach und nach vollständig in Verantwortung des IT-Dienstes über
 - Sicherheit
 - Erweiterungen, Anpassungen
 - Backup
 - allg. Systemkonfiguration
- Dienstkonfiguration („Nutzzdaten“) weiter im Saros-Projekt
- In manchen Anwendungsfällen kann gemeinsame Arbeit notwendig sein
- Beispiel Jenkins
 - Neue Jenkins-Version → IT-Dienst
 - Neue Jenkins-Plugin-Version → Saros-Projekt

- fehlendes Verständnis für Anforderungen zwischen beiden Parteien (Devs und Ops – DevOps-Problematik)
 - z.B. Automatisierungsgrad in professioneller IT-Infrastruktur
 - z.B. Verständnis für Open-Source-Projekt am Fachbereich
- Erst Anforderungen, dann Werkzeug(einsatz)
 - Verstehen, dass Anforderungen nicht verstanden
- IT-Infrastruktur des Fachbereichs zur studentischen Mitarbeit vermutlich ungeeignet
 - Kein bzw. nur sehr eingeschränkter Zugang
- Nutzung von Salt entgegen anfänglicher Vorstellung sehr gering
 - größtenteils nicht sinnvoll oder unerwünscht

- Ursprüngliches Thema hochspannend
 - hohe Relevanz für kleine bis mittelständische Firmen
 - potentieller Einsatz auf meinem eigenen Server
 - viel Entwicklungsarbeit in Open-Source-Umfeld
- Abschließende Umsetzung wenig spannend
 - Großteils einfache Systemkonfiguration
 - kaum selbstständige Arbeit möglich
 - praktisch keine Entwicklungsarbeit mehr

Vielen Dank

Freie Universität



Berlin