

# DevOps

## Automatisierung

Prof. Dr.-Ing. Andreas Heil

 Licensed under a Creative Commons Attribution 4.0 International license. Icons by The Noun Project.

v1.1.0

# Brainstorming

- Was machen Sie eigentlich alles am Rechner von Hand...?

# Problem manuell installierter Server

- Schwer wartbar
- Kosten- und zeitintensiv
- Setup des Servers ist nicht einfach (wenn überhaupt) reproduzierbar
- Oft voller Bugs und Fehler

Alternative?

- Skripte

# Skripte

- Insbesondere Shell-Skripte (Unix)
- Aber auch Batch, PowerShell unter Windows
- Eigentlich schon immer da...

# Vorteile von Skripten

- Einfach zu verstehen
- Straight-forward
- Wird interpretiert

# Nachteile von Skripten

- Wird interpretiert
- Kompatibilität (z.B. bei unterschiedlichen Versionen und Shell-Varianten wie *bash* und *sh*)
- Nur lokale Ausführung
- Keine Parallelität
- Langsam (pro Kommando ein Prozess vgl. Vorlesung Betriebssysteme)

# Ziele

- Was wollen wir speziell im Betriebsumfeld automatisiert skripten?
  - Installationen
  - Updates
  - Konfigurationen

# Betriebsumfeld

Ist allen klar was "Betriebsumfeld" bedeutet?

- Betrieb
  - Zur Erinnerung: Will keine Änderungen, Stabilität
- Entwicklung
  - Will permanent Änderungen



# Configuration Management

- Was wollen wir speziell als Entwickler?
  - Versionskontrolle
- Für den Betrieb auch keine schlechte Idee

# Logische Konsequenz

- Wir schreiben wieder und wieder die gleichen Shell- und Batch-Skripte
- Wir fassen die Skripte, ergänzen diese mit div. Programmen und bauen ein Framework
- Wir generalisieren das Framework so weit, dass es andere verwenden können...

# Frameworks

- Salt / SaltStack
- Puppet
- Chef
- Ansible

# Was ist Ansible

- Framework zur Automatisierung administrativer Tätigkeiten
- Leicht verständlich
- Deklarativ
- Idempotent
- Modular

# Eigenschaften von Ansible

- Kein Master
- Keine Agenten
- Konfiguration in YAML
- Einfach zu lernen (Hausaufgabe!)

# Was wird benötigt

- Ansible verwendet keinen zentralen Server
- Alle Aufgaben werden in *Playbooks* gespeichert
- Control Node
  - Python (2.7/3.x)
  - Ansible
- Nodes
  - ssh-server
  - Python (2.7/3.x)

# Inventory

- Liste von Servern auf die ein Playbook angewendet werden kann

```
[db_master]  
193.161.1.14
```

```
[db_replica]  
163.161.1.21  
163.161.1.22  
163.161.1.23
```

```
[web_01]  
163.161.1.2
```

# Inventory (Forts.)

- Einfache Textdateien
- Beschreiben die Server
- IP-Adressen *oder* DNS-Namen
- Werden mittels Namen Gruppiert



# Inventory (Forts.)

- Variablen und Enumerationen
- Bei *vielen* Systemen sinnvoll

```
[db_master]  
db-[a:c]-srv
```

```
[db_replica]  
db[1:3]-srv
```

```
[web_01]  
{{web_srv}}
```

# Playbook

- Beschreibt den Zielzustand des Nodes
- Deklarativ
- Nutzt YAML

# Playbook (Forts.)

```
-hosts: dev_server    # Gruppenname aus Host-Datei
  user: ubuntu        # Authentifizierung
  sudo: true
  roles
    - java            # Rollen, die installiert werden sollen
    - springboot
```

# Variablen

- Ermöglichen die Wiederverwendung von Playbooks
- Verschlüsselung (z.B. für Passwörter)
- Können für Tasks, Kommandozeile, Dateien etc. verwendet werden

# Variablen (Forts.)

provisioning/vars/main.yml :

```
web_srv
```

provisioning/hosts :

```
[web_01]  
{{web_srv}}
```

# Rollen-Verzeichnisse

```
└─ provisioning/
   ├── .gitignore
   ├── dev-playbook.yml
   ├── prod-playbook.yml
   ├── group_vars/
   │   └─ all.yml
   └─ roles/
       ├── java/
       │   ├── defaults/ # Variablen
       │   ├── files/
       │   ├── handlers/
       │   └─ tasks/
       ├── ...
       └─ springboot/
           ├── ...
           └─ ...
```

# Module

- Quasi für "Alles" Module verfügbar

```
- name: Copy the required files
  copy:
    src: files/
    dest: "{{config_dir}}"
    force: yes
  tags:
    - config
```

# Shell Module

Für alles, für das es keine Module gibt ;-)

```
- name: Magic happens hier  
  shell: /opt/hardening.sh  
  tags:  
    - shell
```

Hinweis: Datei vorher auf Server kopieren, Executable Flag setzen etc.



# Handlers

provisioning/roles/db/handlers/main.yml :

```
---
```

- name: restart db  
service: name=db state=refresh
- name: Copy DB config  
copy: src=db.conf dest=/etc/db/db.conf  
notify: restart db

# Spezifische Module ausführen

```
sudo ansible-playbook -i hosts playbook.yml --tags hardening
```

Hinweis:

- Es werden nur Rollen in Betracht gezogen, die im Playbook aufgeführt sind
- Mehrere Tags pro Eintrag möglich

# Referenzen