

ANSIBLE

CONFIGURATION AS CODE

METANOOK 2018

Tobias Mende | [@tobias_mende](#) | [tobias-men.de](#)

DAS ERWARTET EUCH

- Configuration as Code
- Über Ansible
- Ansible Grundlagen
- Live Demo
- Weitere Informationen

CONFIGURATION AS CODE

**WORKED FINE IN
DEV**

OPS PROBLEM NOW

„Infrastructure as code is the approach to defining computing and network infrastructure through source code that can then be treated just like any software system.

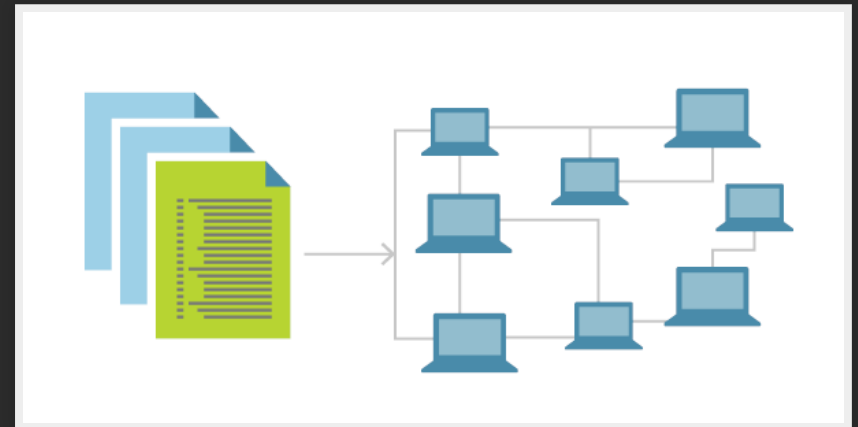
Martin Fowler

PRINZIPIEN

- Konfiguration der Umgebung muss automatisiert sein.
- **Alles** ist unter Versionskontrolle
- So einfach wie möglich halten!
- Konfiguration wird getestet

VIELE VORTEILE!

- Kein implizites Wissen
- Versionskontrolle
- Reviews
- Nachvollziehbarkeit
- Wiederholbarkeit
- Rollbacks
- CI/CD
- ...



ÜBER ANSIBLE

WAS IST ANSIBLE?

- Menschenlesbare Automation für
 - App Deployments
 - Konfigurationsmanagement
 - Orchestration von Abläufen
- „Agentenlos“ mit Standard-Mitteln (SSH, Python)
- Community-Powered und Open Source
- *by Red Hat*

WAS KANN ANSIBLE?

- Systeme in einen gewünschten Zustand bringen
- Befehle auf mehreren Knoten ausführen
- Automatisierung ohne Code
- Deklarative Syntax (*Was, nicht wie*)
- Module für verschiedene Anwendungsgebiete
 - Datenbanken, Netzwerk, Cloud, Dateisysteme, Unix, Windows, ...

MEINE ANSIBLE PROJEKTE

- Server-Konfiguration
 - Webhosting, Mailserver, ...
- RaspberryPi-Konfiguration
 - OpenHab, Homegear, ...
- „Thing“-Konfiguration im IoT
 - Azure Edge, Docker, ...

ANSIBLE GRUNDLAGEN

PLAYBOOKS

- enthalten „Tasks“, „Roles“, „Handlers“, ...
- beschreiben den gewünschten Zustand eines "Inventars"
- sind yml-Files



TASKS

- werden in der definierten Reihenfolge ausgeführt
- werden auf alle Maschinen angewendet
- führen Module mit spezifischen Argumenten aus

```
- name: Installing Packages
  package:
    name: "{{ item }}"
    state: present
  with_items:
    - ufw
    - fail2ban
```

MODULES

- sollten idempotent sein
- bringe bestimmte Eigenschaften eines Systems in einen gewünschten Zustand
- Ansible stellt über 450 verschiedene Module bereit

```
- name: Installing Packages
  package:
    name: "{{ item }}"
    state: present
  with_items:
    - ufw
    - fail2ban
```

HANDLER

werden am Ende ausgeführt, sofern notwendig.

```
- name: write_some_random_foo configuration
  template: src=templates/foo.j2 dest=/etc/foo.conf
  notify: restart apache
```

```
- name: restart apache
  service: name=httpd state=restarted
```


INVENTORIES

- enthalten Gruppen, Hosts und ggf. Variablen
- beschreiben, auf was ein Playbook angewendet wird
- können statische oder dynamische sein
 - Statisch: .ini oder .yml
 - Dynamisch: z.B. Python-Skripte

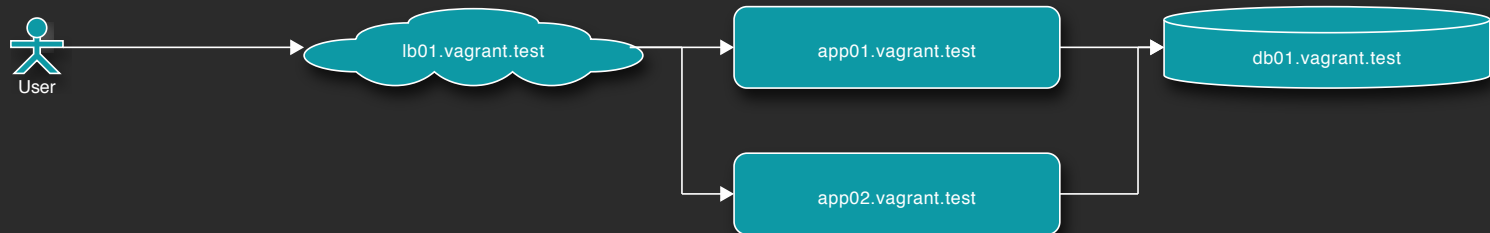
ROLES

- sind Sammlungen von Variablen, Tasks, Dateien, Templates, ...
- bieten Strukturierung von Playbooks
- sind wiederverwendbar und können geteilt werden
- Docker Hub für Ansible Roles:
<https://galaxy.ansible.com>

LIVE DEMO

SZENARIO

Automatischer Aufbau einer Infrastruktur und Deployment für eine Web-Anwendung.



UMSETZUNG

- Vagrant zum Aufbau der Demo-Infrastruktur
- Ansible Playbook
 - Nginx als LB
 - MySQL als Datenbank
 - NodeJS für die Anwendung

[App Source Code – Ansible Playbook](#)

WEITERE INFORMATIONEN

ANFANGEN!

- Python installieren (2 oder 3)
- Ansible installieren (Package Manager oder `pip`)
- ggf. SSH-Key auf Zielrechnern hinterlegen
- Erste Kommandos ausführen mit `ansible`
- Erstes Playbook erstellen (z.B. mit `ansible-generate`)
- Playbook ausführen mit `ansible-playbook`

NÜTZLICHE TOOLS

- VSCode mit Ansible Plugin
- ansible-generator
 - `pip install -U ansible-generator`

BEST PRACTICES

LAUT ANSIBLE DOKU

- Ein Inventory-File pro Umgebung (staging vs. production)
- Top Level Playbooks nach Rollen aufteilen
- Einheitliche Ordnerstruktur

VORTEILE

- schafft Überblick
- ermöglicht feingranulare Ausführbarkeit
- vermeidet falsche Ziele (production statt staging)

ANSIBLE TOWER

- Orchestration and Monitoring
- Credentials
- Permissions
- Visibility/ Compliance

siehe [Ansible Tower Produktseite](#)

REFERENZEN

- [Quickstart Guide](#)
- [Szenario Guides](#)
- [Module Documentation](#)
- [Beispiele auf Github](#)

QUELLECODE & CO.

- [RevealJS Slides](#)
- [Ansible Playbook](#)
- [Node JS Demo App](#)

[Folien online anschauen](#)

VIELEN DANK FÜR DIE AUFMERKSAMKEIT!