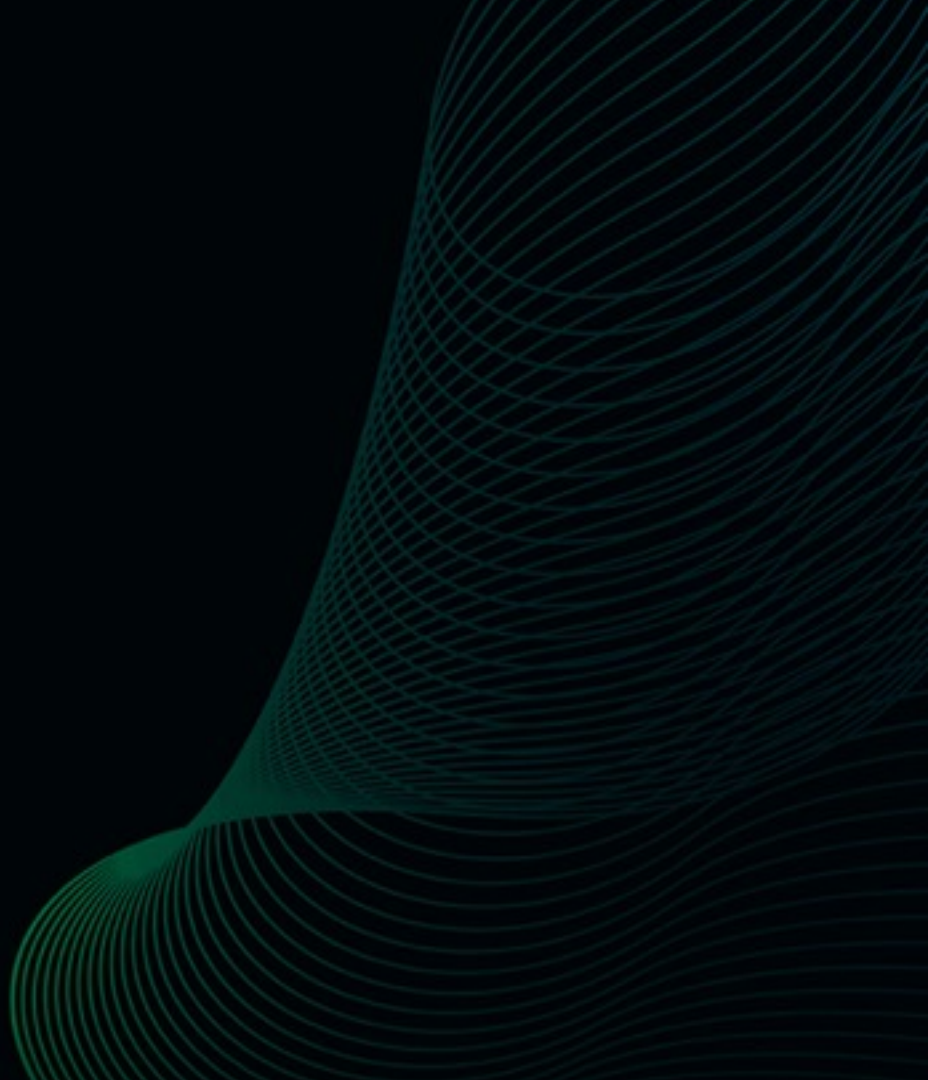
codecentric

Ansible_

Playbooks



Recap

Konfigurationsmanagement

Konfiguration von Systemen - Infrastructure as Code

Verteilung und Orchestrierung von Software

Orchestrierung von Systemen

Zero-Downtime Updates

Ad-hoc Kommandos

Operation am offenen Herzen

Warum Playbooks?

Playbooks

Wiederverwendung

Bündelung

Aufteilung

How To?

Playbooks

Deklarativ vs. Imperativ

WAS?

Erleichterte Konfiguration

Puppet, Chef

WIE?

Konfiguration und Implementierung

Ansible

Playbooks

Basis für Konfigurationsmanagement und mehrschichtigem Deployment

- Konfiguration festlegen
- Orchestrierung von jeglichen manuell durchgeführten Schritten

Einchecken in Source Control System

- Versionsverwaltung

YAML-Syntax

- Leicht lesbar
- Leicht erlernbar

YAML

List

```
---
# A list of tasty fruits
fruits:
    - Apple
    - Orange
    - Strawberry
    - Mango
...
```

Dictionary

```
---
# An employee record
martin:
    name: Martin D'vloper
    job: Developer
    skill: Elite
...
```

Complex

```
---
# An employee record
- martin:
    name: Martin D'vloper
    job: Developer
    skill: Elite
- ella:
    name: Ella Admin
    job:
      - Admin
      - Boss
...
```

http://yaml.org/refcard.html

YAML - Gotchas

```
foo: somebody said I should put a colon here: so I did
foo: “somebody said I should put a colon here: so I did”

foo: {{ variable }}
foo: "{{ variable }}"

foo: "{{ variable }}”/additional/string/literal
foo: "{{ variable }}/additional/string/literal”

version: 1.0
version: “1.0"
```

Playbook Example

```
---
- hosts: webservers
  remote_user: root
  tasks:
  - name: Ist httpd Version aktuell?
    yum: name=httpd state=latest
  - name: Schreibe httpd config
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf
    notify:
    - restart apache
  - name: Läuft httpd?
    service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

Task

Play

Playbook Example

```
- name: Schreibe httpd config
  template: src=/srv/httpd.j2 dest=/etc/httpd.conf
  notify:
  - restart apache
```

```
- name: Schreibe httpd config
  template:
    src: /srv/httpd.j2
    dest: /etc/httpd.conf
  notify:
  - restart apache
```

Playbook Example

```
---
- hosts: webservers
  remote_user: root

  tasks:
  - name: ensure apache is at the latest version
    yum: name=httpd state=latest
  - name: write the apache config file
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf

- hosts: databases
  remote_user: root

  tasks:
  - name: ensure postgresql is at the latest version
    yum: name=postgresql state=latest
  - name: ensure that postgresql is started
    service: name=postgresql state=started
```

S

Erweiterte Themen

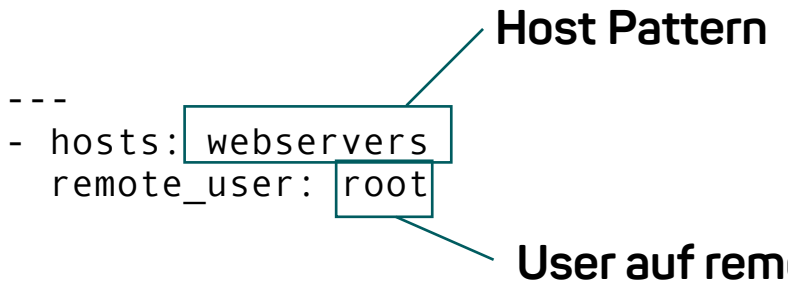
Playbook Handler

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

Tasks nur 1x ausführen

Per Name refererenziert

Playbook Remote User




```
---
- hosts: webservers
  remote_user: root
  tasks:
    - name: test connection
      ping:
      remote_user: yourname
```

Remote User per Task

Playbook Remote User

```
- name: Ensure the httpd service is running
  service:
    name: httpd
    state: started
  become: true
```

Privilige Escalation

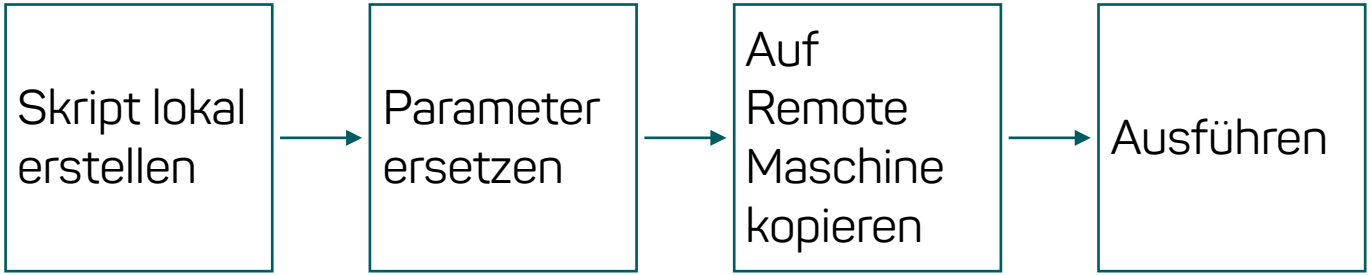
```
- name: Run a command as nobody
  command: somecommand
  become: true
  become_method: su
  become_user: nobody
```

Alle Parameter auch global

```
--ask-become-pass
--become-method=BECOME_METHOD
```

Playbook Remote User - Risiko

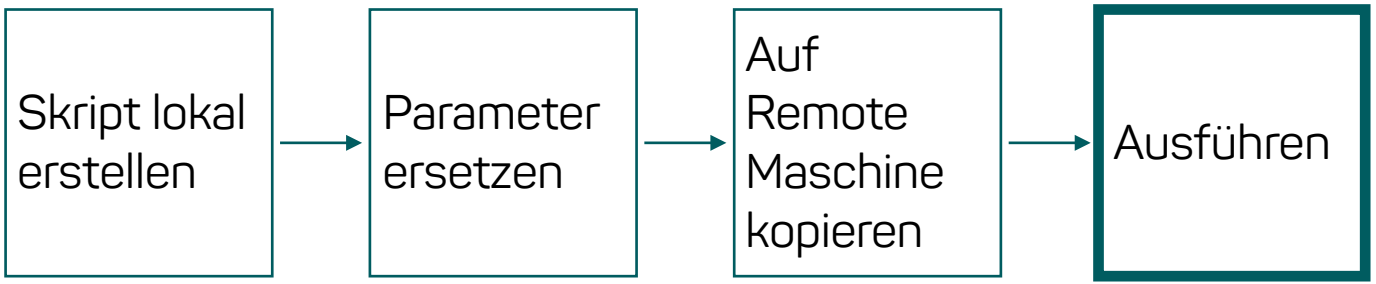
Was passiert bei der Ausführung eines Moduls?



Möglicherweise sensitive Informationen

Playbook Remote User - Risiko

Was geschieht bei der Ausführung eines Moduls?



Welche Rechte werden benötigt?

Kein `become`

`become_user=root` **oder Connection User ist root**

`become_user` **und connection User ist unpriviligiert** | World readable

Playbook Remote User - Risiko

Wie kann man dieses Problem lösen?

Pipelining - Piping zum remote Python Interpreter (nur Python Module)

POSIX acl auf dem Managed Host verwenden

Kein `become_user` mit unpriviliged User

Hands-On: Bootstrapping

Ziel: Bootstrapping einer neuen Instanz

Notwendige Schritte

- Gruppe Ansible erzeugen
- User Ansible erzeugen
- SSH key hinzufügen
- Kein SSH Zugriff für User root
- Kein SSH Zugriff per Password
- Restart