

Ansible

Automatisierung leichtgemacht

ANSIBLE | WOFÜR

- Application Deployment
- Configuration Management
- Continuous Delivery (Jenkins etc.)
- Vagrant Provisioner

ANSIBLE | WESHALB

- einfach zu bedienen
- einfach zu konfigurieren (YAML)
- ohne Agents, Daemons, Server, Datenbanken
- sicher durch SSH-Verbindung
- Open-Source, Github
- keine Programmierung

ANSIBLE | INSTALLATION

Erstellen einer Python virtualenv

```
$ pip install ansible
```

System-Paketverwaltung

```
$ apt-get/yum/brew install ansible
```

Source von Github

```
$ git clone git://github.com/ansible/ansible.git
```

```
$ cd ./ansible
```

```
$ sudo make install
```

ANSIBLE | INVENTAR

Hosts und Gruppen:

INI-Datei `/etc/ansible/hosts`

```
mail.example.com
```

```
[webservers]
```

```
foo.example.com
```

```
bar.example.com
```

```
[dbservers]
```

```
one.example.com
```

ANSIBLE | BENUTZEN

Ad-Hoc:

```
$ ansible all -m ping -o
```

```
web1 | success >> {"changed": false, "ping": "pong"}
```

```
web2 | success >> {"changed": false, "ping": "pong"}
```

```
db1 | success >> {"changed": false, "ping": "pong"}
```

Playbooks:

```
$ ansible-playbook playbook.yml --list-hosts.
```

ANSIBLE | PLAYBOOKS

- YAML - Dateien
- Abfragen auf bestimmte Zustände der Systeme
- Verwendung von Variablen, Templates, Loops
- Können Handler enthalten
- Können wiederverwendet werden

ANSIBLE | PLAYBOOKS

- **hosts:** webservers

vars:

http_port: 80

max_clients: 200

remote_user: root

tasks:

- name: ensure apache is at the latest version

yum: pkg=httpd state=latest

- name: write the apache config file

template: src=/srv/httpd.j2 dest=/etc/httpd.conf

notify:

- restart apache

- name: ensure apache is running

service: name=httpd state=started

handlers:

- name: restart apache

service: name=httpd state=restarted

ANSIBLE | MODULE

ca. 180 Module

apt - Debian/Ubuntu Pakete installieren/entfernen

command - Shellbefehl ausführen

copy - Datei auf das Zielsystem kopieren

file - Ordner/Symlinks erstellen, Rechte vergeben

mysql_db - MYSQL Datenbanken hinzufügen/entfernen

service - Services starten/ neustarten / anhalten

ANSIBLE | INCLUDES

- **name:** this is a play at the top level of a file

hosts: all

remote_user: root

tasks:

- **name:** say hi

tags: foo

shell: echo "hi..."

- **include:** load_balancers.yml
- **include:** webservers.yml
- **include:** dbservers.yml

ANSIBLE | ROLLEN

Ordnerstruktur

```
webservers.yml
```

```
roles/
```

```
    common/
```

```
        files/
```

```
        templates/
```

```
        tasks/
```

```
        handlers/
```

```
        vars/
```

```
        meta/
```

ANSIBLE | ROLLEN

Aufruf der Rollen

- **hosts:** webservers

roles:

- common

- webservers

ANSIBLE | ROLLEN - ABHÄNGIGKEITEN

- Rufen automatisch benötigte Rollen auf
- Liegen unter /meta/main.yml

dependencies:

- { role: common, some_parameter: 3 }
- { role: apache, port: 80 }
- { role: postgres, dbname: blarg, other_parameter: 12 }

ANSIBLE | SCHLEIFEN & BEDINGUNGEN

- **name:** add several users

user: name={{ item }} state=present groups=wheel

with_items:

- testuser1
- testuser2

when: ansible_hostname != "db1"

ANSIBLE | Ressourcen

- Webseite <http://ansible.com>
- Dokumentation <http://docs.ansible.com>
- Twitter [`ansible`](#)
- Newsletter <http://devopsu.com>
- Roles Community <http://galaxy.ansible.com>