

Ansible

Eine Einführung in das Konfigurationsmanagement

Sebastian Bator

github.com/sbator/ansible_ubucon16

20. November 2016



Gliederung

- 1 Ansible Grundlagen
- 2 Persönliche Konfiguration verteilen
- 3 Einfache Webserver Farm



Ansible

- Beschreibt einen Systemzustand als Quellcode
- Konfiguriert den Host entsprechend

```
- name: Deploy .bash_aliases
  copy:
    src: .bash_aliases
    dest: ~/.bash_aliases
```



Warum Konfigurationsmanagement

- Einmal schreiben, automatisiert anwenden
- Definierter Zustand auf allen Servern
- Reproduzierbar
- Einfacher Rollback
- Wissen wird in Code gegossen



Die Konkurrenz

- Puppet
- Chef
- Salt

Ansible:

- Kein Agent auf dem Host, nur ssh und python
- Niedrige Einstiegsschwelle
- Stark in Entwicklung
- Viele Module für Cloud und Virtualisation
- yaml Code ist einfach und gut lesbar



Begriffe

- host** System, das konfiguriert wird
- contoller** Computer der Ansible ausführt
- inventory** Liste der Hosts
 - play** Liste von tasks für eine Gruppe von Hosts
- playbook** Sammlung von plays
 - task** Konfigurationsanweisung
 - role** Ausgelagerte Sammlung von task, wiederverwendbar
- modul** Idempotente Konfigurationsmethode



Idempotenz

Die Anweisung kann beliebig oft hintereinander ausgeführt werden, sie ergibt immer das gleiche Ergebnis

```
cat 'enabled = 1' >> /etc/example.conf
```

```
- name: Enable example  
  lineinfile:  
    dest: /etc/example.conf  
    line: 'enabled = 1'  
    regexp: '^enabled ='
```



Ich hab da mal was vorbereitet



Webserver aufsetzen



Die Server sind an der Lastgrenze!

Wir müssen skalieren



Weiterführende Literatur

- docs.ansible.com/ansible/intro_getting_started.html
- docs.ansible.com/ansible/modules_by_category.html
- galaxy.ansible.com (Marktplatz für Rollen)
- debops.org (Sammlung ausgeklügelter Rollen, gut für „best practice“)
- github.com/sbator/ansible_ubucon16



IPSec Beispiel

```
inventory
```

```
  [ipsec]
```

```
  ubu[1:4]
```

```
group_vars/ipsec.yml
```

```
  conn_type: 'transport'
```

```
host_vars/ubu1.yml
```

```
  ipsec_ip: '10.10.10.1'
```

```
host_vars/ubu2.yml
```

```
  ipsec_ip: '10.10.10.2'
```



IPSec Konfiguration verteilen

```
{% for host in groups['ipsec'] %}  
{% if not ansible_hostname ==          \  
    hostvars[host].ansible_hostname %}  
conn {{ hostvars[host].ansible_hostname }}  
    type= {{ conn_type }}  
    left= {{ ansible_eth0.ipv4.address }}  
    leftsubnet = {{ ipsec_ip }}/32  
    right= {{ hostvars[host].ansible_eth0.ipv4.address }}  
    rightsubnet = {{ hostvars[host].ipsec_ip }}/32  
  
{% endif %}  
{% endfor %}
```



Das Ergebnis

```
conn ubu2
```

```
type= transport
left= 10.52.108.198
leftsubnet = 10.10.10.1/32
right= 10.52.108.229
rightsubnet = 10.10.10.2/32
```

```
conn ubu3
```

```
type= transport
left= 10.52.108.198
leftsubnet = 10.10.10.1/32
right= 10.52.108.13
rightsubnet = 10.10.10.3/32
```

