



Play with Ansible - Provisioning von
Weblogic mit Ansible

Bio



- Thorsten Wussow
- 20 Jahre IT davon 16 Jahre mit Oracle
- thorsten@slix.de
- Twitter: @thwussi



Agenda

- Einleitung
- Ansible Allgemein
- Ansible für Weblogic
- Demo
- Q&A

- Tools zum einfachen Konfigurationsmanagement immer wichtiger, vor allem im DevOps Umfeld
- Ansible ist Konfigurationsmanagement und Provisioning Tool, ähnlich Puppet, Salt usw.
- Aber einfacher, da nur per SSH und keine Agenteninstallation
- inzwischen von RedHat übernommen



Agenda

- Einleitung
- Ansible Allgemein
- Ansible für Weblogic
- Demo
- Q&A

Ansible Installation



- Ansible wird von einem ManagementServer ausgeführt
- Packages für viele Linuxdistributionen
- Hier Installation für OEL 7.2:

Installation Repository

```
rpm -iUvh http://dl.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-8.noarch.rpm
```

Installation Package

```
yum install ansible
```

Prüfen

```
ansible --version
```



Ansible Funktion



- Ansible kann den gleichen Befehl gleichzeitig an mehrere Server übertragen und ausführen
- dazu `/etc/ansible/hosts` editieren und Liste erstellen

```
[doag_servers]
```

```
192.168.176.129
```

```
192.168.176.130
```

- Server idealerweise versorgt mit public/private key
- aber auch Anwendung mit username/passwort möglich



Ansible erste Kommandos



- einzelne Ansible Befehle werden tasks genannt
- Beispiel: ping aller Server

```
ansible all -m ping
```

- ping ist ein Modul, damit können verschiedene Funktionen wie ping, copy, move, yum, usw. ausgeführt werden. z.B. nginx auf den Servern installieren

```
ansible -m yum -a 'name=nginx state=present'  
doag_servers
```

oder

```
ansible all -a "/bin/echo hello"
```



- Handler wie tasks aber werden in Abhängigkeit eines Tasks ausgeführt

```
- name: Install Nginx
  apt: pkg=nginx state=installed update_cache=true
  register: nginxinstalled
  notify:
    - Start Nginx
```

.....

```
handlers:
  - name: Start Nginx
    service: name=nginx state=started
```



Ansible Playbook



- einzelne Tasks werden in Playbooks zusammengefasst
z.B. nginx.yml

```
- hosts: doag_servers
  tasks:
    - name: Install Nginx
      yum: name=nginx state=installed update_cache=true
```

- **Ausführung Playbook**

```
ansible-playbook nginx.yml
```



Ansible Roles



- Mit Roles können Tasks mit den benötigten Daten und Variablen konfiguriert werden
- Das ganze spiegelt sich im Filesystem in einer Directory-Struktur, es müssen nur die Verzeichnisse da sein die gebraucht werden

```
role-name
  files
  handlers
  meta
  templates
  tasks
  var
```



- Unterordner sind festgelegt
- in manchen Ordnern wird ein main.yml erwartet, welches direkt ausgeführt wird wenn vorhanden
- in anderen z.B. files liegen nur Objekte die dann mit einem entsprechenden Modul verarbeitet werden (bei files z.B. copy)
- In templates-Ordner können templates von Konfigurationsdateien abgelegt werden. Das Format der Variablen entspricht jinja2 und die Endungen sollten .j2 sein



- Variablen werden in main.yml unter vars definiert

Format: variable: wert

- im main.yml unter tasks werden alle Tasks für die Role definiert, wird z.B. copy verwendet kommen die Dateien aus files
- werden templates kopiert, werden die Platzhalter automatisch mit den definierten Variablen ersetzt



Ansible Roles ausführen



- Anlegen eines Repositories z.B. unter `/opt/ansible/roles`
- in Datei `/etc/ansible/ansible.cfg` den Pfad zu dem Repository angeben

```
roles_path=/opt/ansible/roles
```

- Erstellen eines `server.yml` Playbook zum ausführen

```
---  
- hosts: all  
  roles:  
    - rolename
```

- Ausführung mit

```
ansible-playbook server.yml
```



Benutzerwechsel und Owner



- Das Playbook wird als Privilegierter Benutzer ausgeführt. In Demo root

- In den tasks wird ein Benutzerwechsel durchgeführt mit become

```
become: True
```

```
become_user: {{ oracle_user }}
```

```
become_method: su
```

- Bei Dateioperationen kann immer der neue Owner, die neue Gruppe und die Berechtigungen der Ziel-Datei mitgegeben werden

```
owner={{ oracle_user }} group={{ oracle_group }}
```

```
mode=0744
```



Agenda

- Einleitung
- Ansible Allgemein
- Ansible für Weblogic
- Demo
- Q&A

Ansible und Weblogic



- Mit roles gut in Installationsschritte zu unterteilen
- z.B. Konfiguration Betriebssystem, Installation Java, Installation Binaries, Konfiguration Domain, Start, usw...
- Möglichkeit, das ganze durch variable für verschiedene Stages bereitzustellen
- Kann auch nur gewisse Teile bereitstellen, z.B. nur Domains erstellen, oder nur die Software bereitstellen



Agenda

- Einleitung
- Ansible Allgemein
- Ansible für Weblogic
- Demo
- Q&A

Aufbau der Demo



- 2 virtuelle Maschinen OEL 7.2
- Ein Server ist der ManagerServer von dem aus Ansible ausgeführt wird
- Auf dem anderen wird ein Weblogic-Cluster bereitgestellt



Erstellte Roles



- linux-jdk
 - Erstellung der notwendigen Verzeichnisse
 - Installation JDK
- fmw-software
 - Verteilung der Software auf allen Servern
- fmw-domain-generic
 - Erstellung der Weblogic-Domain
- nodemanager
 - Erstellt einen NodeManager



Agenda

- Einleitung
- Ansible Allgemein
- Ansible für Weblogic
- Demo
- Q&A

Fragen und Antworten



Tutorial zum Einstieg

- <https://serversforhackers.com/an-ansible-tutorial>

Weblogicscripting für Ansible

- <http://middlewaresnippets.blogspot.de/2015/03/fun-with-ansible.html?view=mosaic>
- <https://github.com/cvezalis/ansible-weblogic-fmw-infra-12c-R2>

Doku und Referens von Ansible

- <http://docs.ansible.com/ansible>





Thorsten Wussow

SLIX Gesellschaft für Computersysteme mbH
Nandlstädter Weg 6
84072 Au i. D. Hallertau

phone +49 1733208013
mail thorsten.wussow@slix.de
web www.slix.de