

Qui suis je ?

- + - Consultant en cloud computing chez redhat
- + - Amoureux de l'opensource
- + - et de Montreal

Introduction

Un lookup permet de récupérer des données à partir du poste où est exécuté le playbook Ansible.

Ces données peuvent être en local, ou être récupéré ailleurs ☹.

Les lookups présentés dans la documentation permettent de:

- + - Générer un mot de passe
- + - Récupérer des données d'un fichier csv, ini
- + - Accéder à des mots de passe AWS stockés grâce à Credstash
- + - Interroger un DNS
- + - etc

Fonctionnalité avancée ?

Note :

This is considered an advanced feature, and many users will probably not rely on these features.

FAUX

Leur utilisation pour la plupart est la suivante:

```
{{ lookup('env', 'HOME') }}
```

Ainsi si on veut récupérer le champs SPF d'un nom de domaine

```
- debug: msg="Le champs SPF de example.com est  
{{ lookup('dig', 'example.com.', 'qtype=SPF') }}"
```

Où les utiliser

Dans une tâche:

```
- user:
  name: mocte
  uid: "{{ lookup('csvfile',
                  'zuma file=passwd delimiter=: col=3') }}"
  group: "{{ lookup('csvfile',
                    'zuma file=passwd delimiter=: col=4') }}"
```

Assigné à une variable

```
motd_value: "{{ lookup('file', '/etc/motd') }}"
```

Ou encore dans une template jinja

Question

Exemple qui utilise le lookup `with_items`:

```
- debug: msg="J'aime les {{ item }}"  
  with_items:  
    - "fraises"  
    - "noix"  
    - "graines"
```

Resultat:

```
TASK [debug] *****  
ok: [localhost] => (item=fraises) => {  
    "item": "fraises",  
    "msg": "J'aime les fraises "  
}  
ok: [localhost] => (item=noix) => {  
    "item": "noix",  
    "msg": "J'aime les noix "  
}  
ok: [localhost] => (item=graines) => {  
    "item": "graines",  
    "msg": "J'aime les graines "  
}
```


Aussi je pourrai aussi utiliser ce même lookup comme suit:

```
- debug: msg="{{ lookup('items',  
                        ['fraises', 'noix', 'graines']) }}"
```

Resultat:

```
TASK [debug] *****
```

```
ok: [localhost] => {  
    "msg": "J'aime les fraises,noix,graines "  
}
```

Autre exemple

```
- debug: msg="Item - {{ item }}"
  with_csvfile:
  - "root file=passwd delimiter=: col=6"
  - "puppet file=passwd delimiter=: col=6"
  - "halt file=passwd delimiter=: col=6"
```

Résultat:

```
TASK [debug] *****
ok: [localhost] => (item=/bin/bash) => {
    "item": "/bin/bash",
    "msg": "Item - /bin/bash "
}
ok: [localhost] => (item=/sbin/nologin) => {
    "item": "/sbin/nologin",
    "msg": "Item - /sbin/nologin "
}
ok: [localhost] => (item=/sbin/halt) => {
    "item": "/sbin/halt",
    "msg": "Item - /sbin/halt "
}
```

Mise en pratique

Nous allons modifier dynamiquement le mot de passe d'un utilisateur mysql suivant le contenu du fichier .my.cnf

Exemple d'un .my.cnf:

```
[mysql]
username=root
password=mypassword
```

Donc l'appel à notre lookup ressemblera à ça:

```
{{lookup ('ini', 'password section=client file=~/.my.cnf') }}
```

DEMO...

```
- hosts: server
vars:
  password: "{{ lookup('ini', 'password section=client file=~/.my.cnf') }}"
tasks:
  - name: Mysql user creation
    mysql_user:
      name: bob
      password: "{{ password }}"
      priv: "*.*:ALL"
      host: "%"
      state: present
```

Questions

mail: pblanc@redhat.com
irc: pblanc (oftc, freenode)