

---

# Ansible

Ansible centralise et automatise les tâches d'administration. Il est sans agent (il ne nécessite pas de déploiements spécifiques sur les clients) et utilise le protocole SSH pour configurer à distance les clients Linux.

L'interface graphique web d'Ansible est payante.



L'ouverture des flux SSH vers l'ensemble des clients depuis le serveur Ansible font de lui un élément critique de l'architecture qu'il faudra attentivement surveiller.

## 1. Installation sur le serveur

Ansible est disponible dans le dépôt EPEL :

- Installation d'EPEL :

```
$ sudo yum install epel-release
```

La configuration du serveur se situe sous `/etc/ansible`.

Deux fichiers de configuration :

- Le fichier de configuration principal *ansible.cfg* : commandes, modules, plugins, configuration ssh ;
- Le fichier de gestion des machines clientes *hosts* : déclaration des clients, des groupes.

### Le fichier `/etc/ansible/hosts`.

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
```

```
# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern you can specify
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com
```

## 2. Gestion des clients

Les serveurs clients doivent être ajoutés dans le fichiers `/etc/ansible/hosts`.

Un groupe "centos6" est créé :

**Le fichier `/etc/ansible/hosts`.**

```
[centos6]
```

```
172.16.1.217
172.16.1.192
```

## 2.1. Tester avec le module ping

Par défaut la connexion par mot de passe n'est pas autorisée par Ansible.

Décommenter la ligne suivante de la section [defaults] dans le fichier de configuration /etc/ansible/ansible.cfg :

```
ask_pass      = True
```

Lancer un ping sur chacun des serveurs du groupe CentOS 6 :

```
# ansible centos6 -m ping
SSH password:
172.16.1.192 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
172.16.1.217 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```



Le mot de passe root des serveurs distants vous est demandé, ce qui pose un problème de sécurité...

## 2.2. Authentification par clef

L'authentification par mot de passe va être remplacée par une authentification par clefs privée/publique beaucoup plus sécurisée.

### Création d'une clef SSH

La bi-clefs va être générée avec la commande **ssh-keygen** :

```
# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

```

Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
d8:7b:45:bb:10:63:d9:fe:ae:71:37:3a:49:d4:fa:7f root@ansible-srv
The key's randomart image is:
+--[ RSA 2048]-----+
|
|       o       |
|      = o .    |
|     o . = o . |
|    . S . = .  |
|   . o =       |
|  . . + =..    |
|   .  * .oE    |
|   .oo +       |
+-----+

```

La clef publique peut être copiée sur les serveurs :

```

# ssh-copy-id root@172.16.1.192
# ssh-copy-id root@172.16.1.217

```

Re-commenter la ligne suivante de la section [defaults] dans le fichier de configuration /etc/ansible/ansible.cfg pour empêcher l'authentification par mot de passe :

```
#ask_pass      = True
```

## Test d'authentification par clef privée

Pour le prochain test, le module shell, permettant l'exécution de commandes à distance, est utilisé :

```

# ansible centos6 -m shell -a "uptime"
172.16.1.192 | SUCCESS | rc=0 >>
 12:36:18 up 57 min,  1 user,  load average: 0.00, 0.00, 0.00

172.16.1.217 | SUCCESS | rc=0 >>
 12:37:07 up 57 min,  1 user,  load average: 0.00, 0.00, 0.00

```

Aucun mot de passe n'est demandé, l'authentification par clef privée/publique fonctionne !

### 3. Utilisation

Ansible peut être utilisé depuis l'interpréteur de commandes ou via des playbooks.

#### 3.1. Les modules

La liste des modules classés par catégories se trouve à l'adresse [http://docs.ansible.com/ansible/modules\\_by\\_category.html](http://docs.ansible.com/ansible/modules_by_category.html). Ansible en propose plus de 750 !

Un module s'invoque avec l'option -m de la commande ansible

#### *Exemples d'installation logiciel*

Le module yum permet d'installer des logiciels sur les clients cibles :

```
# ansible centos6 -m yum -a name="httpd"
172.16.1.192 | SUCCESS => {
    "changed": true,
    "msg": "",
    "rc": 0,
    "results": [
        ...
        \n\nComplete!\n"
    ]
}
172.16.1.217 | SUCCESS => {
    "changed": true,
    "msg": "",
    "rc": 0,
    "results": [
        ...
        \n\nComplete!\n"
    ]
}
```

Le logiciel installé étant un service, il faut maintenant le démarrer avec le module service (centos 6) ou systemd (centos 7) :

```
# ansible centos6 -m service -a "name=httpd state=started"
172.16.1.192 | SUCCESS => {
    "changed": true,
    "name": "httpd",
    "state": "started"
}
172.16.1.217 | SUCCESS => {
    "changed": true,
    "name": "httpd",
    "state": "started"
}
```

### 3.2. Les playbooks

Les playbooks ansible décrivent une politique à appliquer à des systèmes distants, pour forcer leur configuration. Les playbooks sont écrits dans un format texte facilement compréhensible regroupant un ensemble de tâches.

#### *Exemple de playbook apache et mysql*

Le playbook suivant permet d'installer apache et mysql sur nos serveurs cibles :

```
---
- hosts: centos6
  remote_user: root

  tasks:
    - name: ensure apache is at the latest version
      yum: name=httpd,php,php-mysql state=latest
    - name: ensure httpd is started
      service: name=httpd state=started
    - name: ensure mysql is at the latest version
      yum: name=mysql-server state=latest
    - name: ensure mysqld is started
      service: name=mysqld state=started
```

L'exécution du playbook s'effectue avec la commande **ansible-playbook** :

```
$ ansible-playbook test
```

```
PLAY [centos6]
```

```
*****
```

```
TASK [setup]
*****
ok: [172.16.1.192]
ok: [172.16.1.217]

TASK [ensure apache is at the latest version]
*****
ok: [172.16.1.192]
ok: [172.16.1.217]

TASK [ensure httpd is started]
*****
changed: [172.16.1.192]
changed: [172.16.1.217]

TASK [ensure mysql is at the latest version]
*****
changed: [172.16.1.192]
changed: [172.16.1.217]

TASK [ensure mysqld is started]
*****
changed: [172.16.1.192]
changed: [172.16.1.217]

PLAY RECAP
*****
172.16.1.192      : ok=5    changed=3    unreachable=0
                  failed=0
172.16.1.217      : ok=5    changed=3    unreachable=0
                  failed=0
```

