Lab 1 – Installation

Introduction

Configuration requise

Machine de contrôle

Vous pouvez exécuter **Ansible** sur n'importe quelle machine sur laquelle Python 2.6 ou 2.7 est installé (Windows n'est pas pris en charge pour la machine de contrôle).

Ansible prend en charge RedHat, Debian, CentOS, OS X, tous les BSD.

Nœuds clients

Les machines clientes doivent au moins avoir Python 2 (version 2.6 ou ultérieure) ou Python 3 (version 3.5 ou ultérieure)

Si SELinux est activé sur les nœuds distants, vous devrez installer le package `libselinux-python` sur les nœuds avant d'utiliser toute fonction liée à `copy` / `file` / `template` dans Ansible

Environnement

```
| Nom d'hôte   | Adresse IP       | OS                       | Objectif            |
|--------------|------------------|--------------------------|---------------------|
| server.local | 192.168.208.163  | CentOS 7 / Ubuntu 18.04 | Machine de contrôle |
| node1.local  | 192.168.208.164  | CentOS 7                 | Nœud géré 1         |
| node2.local  | 192.168.208.165  | Ubuntu 18.04             | Nœud géré 2         |
```

Avant de commencer l'installation de Ansible sur l'environnement de travail, il faut identifier la lister les IPs et leurs noms dans le fichier **`/etc/hosts`** de toutes les machines.

```
sudo vi /etc/hosts
```

```
192.168.208.163 server
192.168.208.164 node1
192.168.208.165 node2
```

Installer Ansible sur CentOS 7 / RHEL 7 / Ubuntu 18.04 / 16.04 et Debian 9

Configuration de la machine de contrôle

1. Pour installer Ansible, nous devrons ***activer le référentiel EPEL sur CentOS 7 / RHEL7*** .

```
### CentOS 7 ###
sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

```
### RHEL 7 ###
subscription-manager repos --enable rhel-7-server-ansible-2.6-rpms
```

```
### Ubuntu 18.04 / Ubuntu 16.04 ###
sudo apt-get update
sudo apt-get install software-properties-common
sudo apt-add-repository ppa:ansible/ansible
sudo apt-get update
```

```
### Debian 9 ###
sudo apt-get install dirmngr
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
93C4A3FD7BB9C367
echo "deb http://ppa.launchpad.net/ansible/ansible/ubuntu trusty main" | sudo tee -
a /etc/apt/sources.list.d/ansible.list
sudo apt-get update
```

2. Installez Ansible.

```
### CentOS 7 / RHEL 7 & Fedora 28 ###
sudo yum install -y ansible
```

```
### Ubuntu 18.04 / 16.04 & Debian 9 ###
sudo apt-get install -y ansible
```

Une fois Ansible installé, vérifiez la version d'Ansible en exécutant la commande ci-dessous.

```
ansible --version
```

Output:

```
ansible 2.9.18
config file = /etc/ansible/ansible.cfg
configured module search path = [u'/home/formation/.ansible/plugins/modules',
u'/usr/share/ansible/plugins/modules']
ansible python module location = /usr/lib/python2.7/site-packages/ansible
executable location = /usr/bin/ansible
python version = 2.7.5 (default, Apr  2 2020, 13:16:51) [GCC 4.8.5 20150623 (Red
Hat 4.8.5-39)]
```

Configurer les nœuds gérés

Les machines clientes doivent au moins avoir Python 2 (version 2.6 ou ultérieure) ou Python 3 (version 3.5 ou ultérieure).

```
### CentOS 7 / RHEL 7 & Fedora ###
sudo yum install -y python
```

```
### Ubuntu 18.04 / 16.04 & Debian 9 ###
sudo apt-get install -y python
```

SELinux (CentOS / RHEL / Fedora)

Si SELinux est activé sur les nœuds gérés, vous devrez installer le package ci-dessous sur les nœuds avant d'utiliser les fonctionnalités `copy` / `file` / `template` dans Ansible.

```
yum install -y libselinux-python
```

Authentification SSH

Comme indiqué précédemment, Ansible utilise *OpenSSH* pour la communication à distance. Ansible prend en charge l'authentification sans **mot de passe** et par **mot de passe** pour exécuter des commandes sur les nœuds gérés.

Authentification par clé SSH (*authentification sans mot de passe*)

Lorsqu'il s'agit d'authentification **ssh**, par défaut, il utilise des clés ssh (authentification sans mot de passe) pour s'authentifier auprès de la machine distante.

Pour activer le tunnel SSH entre la machine serveur et les nœuds à controler, il faut créer une paire de clé (Publique/Privée) sur le serveur, puis copier la clé publique sur chaque nœud à contrôler.

Génération des clés

```
[ansible@server ~]$ ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansible/.ssh/id_rsa):
Created directory '/home/ansible/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ansible/.ssh/id_rsa.
Your public key has been saved in /home/ansible/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:5I/N5tdmpTlOUW4lFGYj/Upu4QbbvgLH+lKmZYiuJis ansible@server
The key's randomart image is:
+---[RSA 2048]----+
|             ..*. |
|              =.. |
|         .     ..o|
|       o    . o+o|
|        S. o B.oo|
|        .=o O Bo.|
|       .. =X =.+ |
|  E . . .o+ o.O  |
|   ..+..  .+.=oo |
+----[SHA256]-----+
```

Copiez la clé publique dans la machine node1

```
[ansible@server ~]$ ssh-copy-id root@node1
/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/ansible/.ssh/id_rsa.pub"
The authenticity of host 'node1 (192.168.208.164)' can't be established.
ECDSA key fingerprint is SHA256:8HB6ZSbBDRixAUN1hcDN75VPQizlGZTyY/xSaWVwQYo.
ECDSA key fingerprint is MD5:e8:59:48:5e:ca:0b:6e:f5:31:a8:7a:08:7e:68:d8:99.
Are you sure you want to continue connecting (yes/no)? yes
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
ansible@node1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ansible@node1'"
and check to make sure that only the key(s) you wanted were added.
```

Copiez la clé publique dans la machine node2

```
[ansible@server ~]$ ssh-copy-id root@node2
/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/ansible/.ssh/id_rsa.pub"
The authenticity of host 'node2 (192.168.208.165)' can't be established.
ECDSA key fingerprint is SHA256:Vrplx5WQR83acMSYOk75pTVX/n+ODkhvDhfCBZNlNnY.
ECDSA key fingerprint is MD5:79:4b:62:c6:09:17:6b:16:76:3e:fc:fc:3c:86:cb:5d.
Are you sure you want to continue connecting (yes/no)? yes
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
ansible@node2's password:
```

```
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ansible@node2'"
and check to make sure that only the key(s) you wanted were added.
```

Une fois que vous avez configuré la communication sans mot de passe, vérifiez-la.

```
$ ssh ansible@192.168.208.164
$ ssh ansible@192.168.208.165
```

Vous devriez maintenant pouvoir vous connecter à la machine distante sans mot de passe.

Authentification par mot de passe

L'authentification par mot de passe peut également être utilisée si nécessaire en fournissant l'option **--ask-pass**. Cette option nécessite **sshpass** sur la machine de contrôle.

```
### CentOS 7 / RHEL 7 et Fedora ###
yum install -y sshpass
```

```
### Ubuntu 18.04 / 16.04 & Debian 9 ###
sudo apt-get update
sudo apt-get install -y sshpass
```

Créer un inventaire Ansible

Modifiez (ou créez) le fichier **/etc/ansible/hosts**. Ce fichier contient l'inventaire des hôtes distantes auxquelles Ansible se connectera via SSH pour les gérer.

```
### CentOS 7 / RHEL 7 & Fedora ###
sudo vi /etc/ansible/hosts
```

```
### Ubuntu 18.04 / 16.04 & Debian 9 ###
sudo nano /etc/ansible/hosts
```

Mettez un ou plusieurs systèmes distants et groupez-les. Ici, on ajoute les deux machines au groupe **demo_servers.**

Les groupes sont utilisés pour classer les systèmes pour un usage particulier. Si vous ne spécifiez aucun groupe, ils agiront comme des hôtes non groupés.

```
[demo_servers]

192.168.208.164
192.168.208.165
```

Premières commandes

Il est maintenant temps de vérifier tous nos nœuds en faisant simplement un ping depuis la machine de contrôle, pour ce faire, nous utiliserons la commande **ansible** avec les options **-m** (chargement de module) et **all** (tous les serveurs).

 # all servers - Fonctionne lorsque le nom d'utilisateur du serveur et du client est le même (sans mot de passe)

```
ansible all -m ping
```

 # all servers - "vagrant" est l'utilisateur du nœud géré (sans mot de passe)

```
ansible all -u ansible -m ping
```

OR

 # Only demo_servers group - "vagrant" est l'utilisateur du nœud géré (sans mot de passe)

```
ansible demo_servers -u ansible -m ping
```

OR

 # Si vous utilisez l'authentification par mot de passe

```
ansible -m ping all -u ansible --ask-pass
```

Output:

```
192.168.208.164 | SUCCESS => {
"changed": faux,
"ping": "pong"
}

192.168.208.165 | SUCCESS => {
"changed": false,
```

```
"ping": "pong"
}
```

Dans l'exemple ci-dessus, nous avons utilisé le module ping avec la commande **ansible** pour envoyer un **ping** à tout ou au groupe d'hôtes distantes.

De la même manière, nous pouvons utiliser différents modules avec la commande **ansible** , vous pouvez trouver les modules disponibles ici https://docs.ansible.com/ansible/latest/user_guide/modules_intro.html