

Installer le nécessaire pour faire fonctionner ansible sur vos 3 VM

Dans cette partie vous allez installer le serveur Ansible.

Installation ansible sur le serveur

1) Connectez-vous au serveur Ansible en SSH (vagrant ssh)

2) Installez le serveur Ansible via les commandes suivantes :

```
sudo apt update
sudo apt install -yqq git software-properties-common tree
sudo apt-add-repository --yes --update ppa:ansible/ansible
sudo apt install -yqq ansible
```

Attention si vous n'accédez pas à internet depuis un vm :
conf des vm (ansible, client1,client2) pour accéder à internet :
editer /etc/resolv.conf
mettre nameserver 8.8.8.8 à la place de 127.0...

en ligne de commande :

```
sudo sed -i -e "s/127.0.0.53/8.8.8.8/g" /etc/resolv.conf
```

Le problème peut ré-apparaître ensuite

3) Vérifier que le serveur ansible est bien installé en affichant sa version. **Quelle est sa version ?**

L'installation crée l'arborescence suivante (placez vous dans le dossier /etc/ansible et faite tree -R) :

Le fichier « ansible.cfg » contient la configuration par défaut d'Ansible.

Le fichier « hosts » est le fichier d'inventaire.

Le dossier « roles » contiendra tous les rôles ansible.

Configurations ansible et les deux clients

Configuration du serveur ansible et des clients

Génération paire de clés SSH

Le serveur Ansible utilise SSH comme protocole de communication avec les clients
Ansible sous Linux, et WinRM pour les clients Windows.

Dans cette partie nous allons nous focaliser sur la configuration de SSH pour les communications avec les clients Linux.

La configuration de SSH passe par les étapes suivantes :

1) Créez l'utilisateur « ansible » via la commande suivante sur les 3 machines de votre laboratoire :

```
sudo useradd -m -s /bin/bash -p $(openssl passwd -crypt azerty) ansible
sudo usermod -aG sudo ansible
```

2) Faites en sorte que l'utilisateur ansible puisse faire le l'élévation de privilège « sudo » sans renseigner son mot de passe en modifiant sa configuration de sudoer comme suit :

Ouvrez le fichier « /etc/sudoers » puis modifiez la définition du groupe de sudoers comme suit (toujours sur les 3 machines) :

```
sudo nano /etc/sudoers
...
%sudo ALL=(ALL:ALL) NOPASSWD: ALL
```

3) Connectez-vous sur la machine « ansible »

4) Générez la paire de clés nécessaire à Ansible pour se connecter aux machines clientes avec les commandes suivantes (que sur la machine ansible) :

```
sudo mkdir -p /etc/keys-ansible/
sudo ssh-keygen -f /etc/keys-ansible/ansible_rsa
sudo chown ansible /etc/keys-ansible/*
```

Laissez-les tous les paramètres vides.

Configuration "ansible.cfg"

Le fichier « ansible.cfg » contient les configurations utilisées par Ansible.

Le fichier « ansible.cfg » peut se retrouver aux emplacements suivants (Classés par ordre de priorité) :

L'emplacement « par défaut » est donc « /etc/ansible/ansible.cfg ».

1) Générez un template de ansible.cfg via la commande :

```
sudo ansible-config init --disabled > /etc/ansible/ansible.cfg

passer en root ( sudo su -) si la commande est refusée puis quitter root (exit)
```

2) Ouvrez le fichier « /etc/ansible/ansible.cfg » décommentez (enlever ; au début) puis modifiez les valeurs suivantes :

```
private_key_file = /etc/keys-ansible/ansible_rsa
host_key_checking = false
```

remote_user = ansible
deprecation_warnings=False

N'oubliez pas de décommenter les lignes (enlever ; au début)

3) À quoi servent les 4 variables remplacées précédemment ?

Note : Prenez le temps de parcourir les différentes options de personnalisation disponibles dans le fichier « ansible.cfg ».

Configuration SSH

Nous utiliserons comme compte de service ansible le compte utilisateur « ansible ».

Le compte utilisé par ansible doit disposer des droits sudoers sur les clients.

1) Copiez le contenu du fichier de la clé publique (à afficher via la commande « cat ») « /etc/keys-ansible/ansible_rsa.pub » (située sur la machine ansible) dans le fichier « /home/ansible/.ssh/authorized_keys » (à ajouter à la suite des lignes déjà présentes, ne surtout pas remplacer le fichier ! Ou créer le fichier s'il n'existe pas) sur toutes machines qui seront gérées par Ansible (y compris la machine ansible elle-même !)

Exemple pour la machine « ansible » :

```
sudo su -  
mkdir -p /home/ansible/.ssh/  
cat /etc/keys-ansible/ansible_rsa.pub > /home/ansible/.ssh/authorized_keys  
chown -R ansible /home/ansible/  
#revenez sur votre utilisateur courant  
exit
```

Pour les clients, afficher avec la commande « cat » le contenu de la clé, copier le contenu, créer le fichier /home/ansible/.ssh/authorized_keys sur les clients et copier dedans le contenu.

```
#Sur la machine ansible, copier tout le contenu du fichier ansible_rsa.pub  
cat /etc/keys-ansible/ansible_rsa.pub  
#Sur les deux clients (client1 et client2)  
sudo su -  
mkdir -p /home/ansible/.ssh/  
nano /home/ansible/.ssh/authorized_keys  
#Copier le contenu du fichier dans authorized_keys, enregistrer et quitter  
chown -R ansible /home/ansible/  
#revenez sur votre utilisateur courant  
exit
```

Installation python

1) Installez le paquet « python » sur les clients via la commande suivante :

```
sudo apt update
```

```
sudo add-apt-repository ppa:deadsnakes/ppa
```

```
sudo apt update
```

```
sudo apt install python3.7
```

À ce stade, le serveur Ansible et les clients linux sont bien configurés.