Ansible - Installation

Difficulté : Confirmé

Notions : automation, ansible, CI/CD

I. Introduction

Ansible est un Outil de Gestion de Configuration. Il permettra d'automatiser l'ensemble des tâches de déploiement, installation, configuration ainsi que l'ensemble des tâches courantes, qu'elles soient récurrente ou non.

II. Installation ansible

2.1 Installer les prérequis

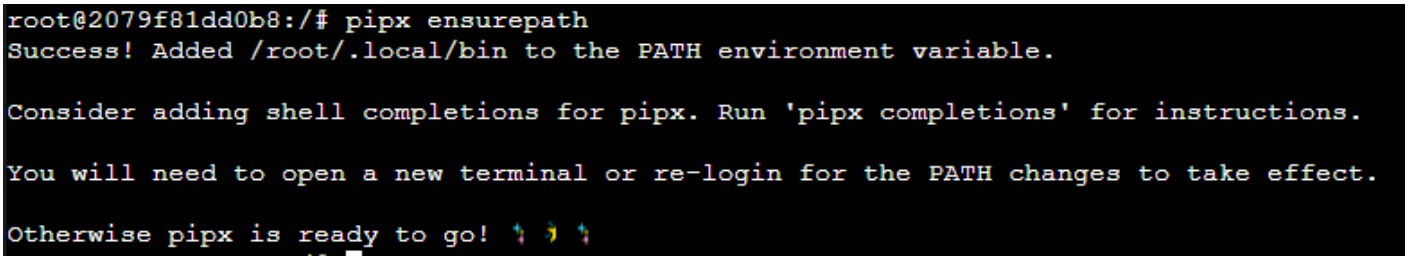
Ansible se base essentiellement sur du python.

Installer '***pipx***' qui servira à installer ansible et les différents modules.

```
apt install pipx
```

intégrer ensuite les commandes pipx au commandes shell :

```
pipx ensurepath
```

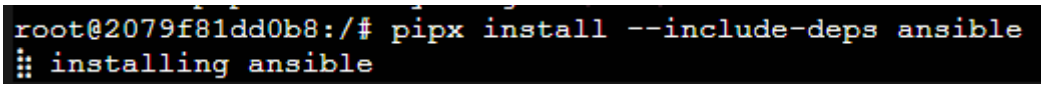


2.2 Installer Ansible

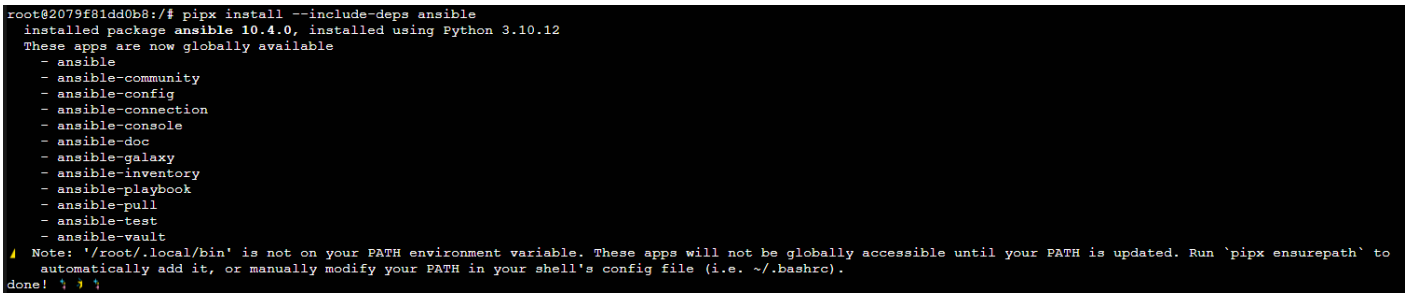
Pour installer Ansible, cela se fait avec pip :

```
pipx install --include-deps ansible
```

L'installation se lance.



A la fin de l'installation :



III. Préparer la connexion aux clients

3.1 Créer le compte ansible

Ansible, par défaut, se connectera en SSH aux clients.

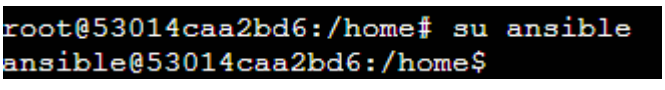
Il faut donc créer un compte ansible pour la connexion SSH ainsi que ses clé. Puis il faudra ensuite pousser les clés sur l'ensemble des clients (à minima linux).

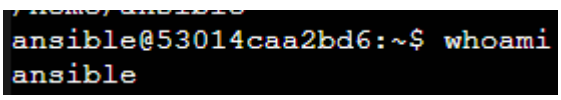
Créer l'utilisateur :

```
useradd -b /bin/bash -d /home/ansible -m ansible
```

Puis se connecter en tant que l'utilisateur ansible :

```
su ansible
```





aller dans le répertoire home avec '***cd***'.

```
ansible@53014caa2bd6:~$ cd
ansible@53014caa2bd6:~$ pwd
/home/ansible
```

3.2 Créer les clés SSH

Pour se connecter aux machine distantes, il faudra créer les clé SSH pour le compte ansible.

en tant que l'utilisateur ansible, lancer la commande suivante :

```
ssh-keygen -t rsa -b 4096 -C "SSH key For ansible automation account"
```

```
ansible@53014caa2bd6:~$ ssh-keygen -t rsa -b 4096 -C "SSH key For ansible automation account"
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:
[redacted] SSH key For ansible automation account
The key's randomart image is:
+---[RSA 4096]----+
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
+----[SHA256]-----+
ansible@53014caa2bd6:~$
```

Il faudra ensuite afficher la clé publique pour la partager aux autres machines.

```
cat .ssh/id_rsa.pub
```

Et copier/stocker le résultat de la commande dans un vault.

Installation de la clé publique sur les clients

Sur les clients, il faudra tout d'abord créer le compte ansible et l'ajouter aux sudoers.

```
useradd -b /bin/bash -d /home/ansible -m -G sudo ansible
```

> **Attention** : Si l'instance SSH a été configurée sur les clients pour autoriser la connexion à un groupe particulier, il faut également ajouter le user ansible au groupe SSH.

```
useradd -b /bin/bash -d /home/ansible -m -G sudo,<groupessh> ansible
```

Pour installer la clé sur les clients, utiliser la commande :

```
su <utilisateur ansible>
mkdir ~/.ssh
chmod 700 ~/.ssh
touch ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
echo '<clé publique précédement copiée>' >> ~/.ssh/authorized_keys
```

Pour vérifier la présence de la clé :

```
su <utilisateur ansible>
cat ~/.ssh/authorized_keys
```

Il faudra également lui définir un mot de passe pour qu'il puisse utiliser le sudo.

```
passwd ansible
```

Ensuite, depuis la machine de contrôle ansible, connecter en ssh :

```
su ansible
ssh ansible@<FQDN>
```

3.3 Lancer l'agent SSH

Afin de ne pas avoir à retaper la clé ssh à chaque fois, lancer le ssh agent qui se chargera de placer les informations en cache et valider les connexions. Pour lancer l'agent :

```
eval $(ssh-agent -s)
```

```
ansible@53014caa2bd6:~$ eval 'ssh-agent'
SSH_AUTH_SOCK=/tmp/ssh-XXXXXXyzEkQy/agent.6304; export SSH_AUTH_SOCK;
SSH_AGENT_PID=6305; export SSH_AGENT_PID;
echo Agent pid 6305;
```

> **ASTUCE** : Pour que l'agent se lance au démarrage système, ajouter cette commande dans le fichier '***~/.bashrc***' de l'utilisateur.

Puis ajouter l'identité SSH à l'agent avec :

```
ssh-add ~/.ssh/id_rsa
```

et retaper le mot de passe de la clé.

```
ansible@53014caa2bd6:~$ ssh-add .ssh/id_rsa
Enter passphrase for .ssh/id_rsa:
Identity added: .ssh/id_rsa (SSH key For ansible automation account)
```

Ansible est maintenant lancé, et prêt à se connecter aux clients.

Pour tester la connexion aux machines, on peut créer un premier inventaire et utiliser la commande :

```
ansible all --inventory <chemin/vers/inventaire> -m ping
```

IV. Configurer ansible

4.1 Modifier la configuration

Le fichier de configuration par défaut d'Ansible est situé dans '***/etc/ansible/ansible.cfg***'

Il permet d'attribuer un comportement par défaut pour ansible.

Il pourra être ensuite overrided par un fichier '***ansible.cfg***' présent dans le playbook.

> **Informations** : voir ***ici*** pour les paramètres possibles dans le fichier de configuration.

4.2 Lier au gitlab

> **Prérequis :** il faut que le compte git qui sera utilisé existe et qu'un token ai été généré et que le paquet '***git***' soit installé sur le serveur.

Il peut être utile de se connecter à un repo git pour récupérer les playbooks dévellopés et les mettre à jour.

Pour cela, enregistrer les creds avec la commande :

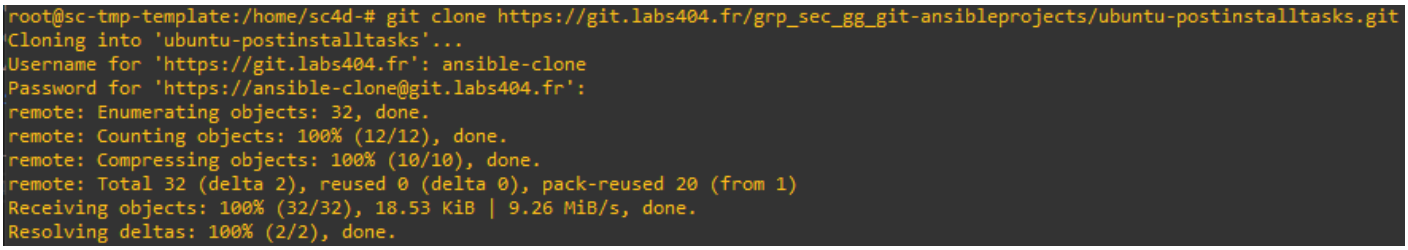
```
git config --global credential.helper store
```

Cette commande permettra d'enregistrer les prochains identifiants utilisés dans le vault.

Puis faire :

```
git clone <repo url>
```

Les credentials sont alors demandés et seront enregistrés pour un usage futur.



Pour mettre à jour les playbooks à l'avenir, il faudra se rendre dans le dossier du playbook puis faire :

```
git update
```

Revision #2
Created 2026-07-30 07:30:51 UTC by Olivier
Updated 2026-07-30 07:33:19 UTC by Olivier