

## Procédure Playbook Ansible (TP) : Installation de Docker via Ansible

Pour ce tutoriel, je pars du principe qu'Ansible est déjà installé sur votre machine « Serveur »

Pour ce tuto, je préconise l'utilisation de deux machines virtuelles. Ici, nous allons les nommer « Ansible (Serveur) » et « Ansible (client) », nous utiliseront la distribution « Ubuntu 22.04 LTS »

**Ansible (serveur) possède l'IP 192.168.1.21**

**Le user est ubuntu02**

**Le mot de passe de la vm est test**

**Ansible (client) possède l'IP 192.168.1.19**

**Le user est ansible**

**Le mot de passe de la vm est test**

### 1) Création d'un répertoire de travail qui contiendra vos fichier Ansible (→ Ansible Serveur )

Pour ce tuto, il s'appellera « AnsibleLearning ». Dans votre répertoire « ~ » , utilisez la commande « mkdir » pour créer votre dossier → mkdir AnsibleLearning

Si vous exécutez la commande « ls » vous devriez voir votre dossier :

```
ubuntu02@ubuntu02-Virtual-Machine:~$ ls
AnsibleLearning  dockercompose  DockerPrepa
```

Rendez vous dans votre dossier avec la commande « cd » (Change Directory) → cd AnsibleLearning (Astuce : vous pouvez utiliser « tab » pour l'autocomplétions de vos commandes)

### 2) Création d'une clé SSH qui servira pour nos commandes Ansible

Notre répertoire étant créé, il nous manque un prérequis avant de passer à la création des fichiers : une clé SSH.

Pour générer une clé ssh, utilisez la commande « ssh-keygen » , vous pouvez préciser le type de clé souhaité via « -t » ce qui donne ici : « ssh-keygen -t rsa » puis « entrer ».

```
ubuntu02@ubuntu02-Virtual-Machine:~/AnsibleLearning$ ssh-keygen -t rsa
```

A ce moment, on vous demande de choisir un nom pour votre clé. Pour ce tuto, je l'appellerai « test\_rsa ». (Pensez bien à indiquer le chemin /home/votre\_user/.ssh/nom\_de\_votre\_clé\_rsa)

```
Enter file in which to save the key (/home/ubuntu02/.ssh/id_rsa): /home/ubuntu02/.ssh/test_rsa
```

Puis une paraphrase est demandée, il s'agit d'un mot de passe qui servira à accepter les échanges entre votre clé privé et votre clé publique sur vos serveurs.

Ici, je vais simplement mettre « test » pour simplifier. Ensuite, ma clé se génère :

```
Your identification has been saved in /home/ubuntu02/.ssh/test_rsa
Your public key has been saved in /home/ubuntu02/.ssh/test_rsa.pub
The key fingerprint is:
SHA256:oC0yHF9qmFQa6Y/U14UZejNJl0vqPmepx3Pixu8VC8k ubuntu02@ubuntu02-Virtual-Machine
The key's randomart image is:
+---[RSA 3072]-----+
|      .+.      |
|      o.=      |
|    o . ..B..  |
|  . =  .oo= .  |
| *. ..oo..SE . |
|+oB.+ . . . o |
| . =  + . . o  |
|  . o @ . .    |
|    o+O.*o     |
+---[SHA256]-----+
ubuntu02@ubuntu02-Virtual-Machine:~/AnsibleLearning$
```

Ceci est une clé de test, mais il n'est évidemment pas conseillé de faire des captures d'écran de vos clés !

Par défaut, vous trouverez vos clés dans /home/votre\_user/.ssh :

```
ubuntu02@ubuntu02-Virtual-Machine:~/.ssh$ ls
id_rsa  id_rsa.pub  known_hosts  known_hosts.old  test_rsa  test_rsa.pub
```

Maintenant que votre clé ssh est créée, il faut la transférer sur notre client « Ansible (client) ». Pour cela, effectuez la commande « ssh-copy-id » comme suit :

**sudo ssh-copy-id -i ~/.ssh/nom\_de\_votre\_cle\_rsa user\_ssh\_distant@ip\_ssh\_distant**

```
ubuntu02@ubuntu02-Virtual-Machine:~/.ssh$ sudo ssh-copy-id -i ~/.ssh/test_rsa ansible@192.168.1.19
```

Une fois cette commande lancée, il faut entrer le mot de passe sudo pour Ansible (serveur) : test , puis si tout se passe bien, la copie s'effectue.

```
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ubuntu02/.ssh/test_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
```

Et on doit entrer le mot de passe du user de l'hôte distant (ansible) pour installer les clés. Ici, le mot de passe est : test. Si tout va bien, vous devriez avoir ce message :

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

Now try logging into the machine, with:  "ssh 'ansible@192.168.1.19'"
and check to make sure that only the key(s) you wanted were added.
```

Il indique que votre clé a bien été installée sur le serveur distant.

### 3) Création des deux fichiers nécessaires pour utiliser Ansible

Ansible a besoin de deux fichiers pour fonctionner :

un fichier `Inventory.ini` qui répertorie nos Hosts et leurs « coordonnées » pour les joindre en SSH  
un fichier en `.yml` qui sera le fichier de votre playbook, il contiendra les instructions pour effectuer notre installation de « Docker ».

Avant toute chose, retournons dans notre répertoire via la commande « `cd AnsibleLearning` ».

Puis commençons par créer notre fichier « `Inventory` » via la commande « `Nano inventory.ini` » et remplissez le comme suit :

```
[clients]
ansible_client ansible_ssh_host=192.168.1.19 ansible_ssh_user=ansible ansible_ssh_private_key_file=~/.ssh/id_rsa
```

Détailons un peu les lignes :

ansible\_client : fait office de « Nom dns » pour notre serveur

Ansible\_ssh\_host : Il s'agit de l'IP de l'hôte distant (ici Ansible client)

Ansible\_ssh\_username : Il s'agit du user utilisé sur l'hôte distant (ici ansible)

Ansible\_ssh\_private\_key\_file : Il s'agit du chemin vers lequel la clé privée SSH de notre serveur (ici Ansible (serveur) ) a été copiée sur le serveur distant (Ansible (client) ).

Sauvegardez puis créez le fichier de notre playbook ici « `test.yml` » via la commande « `nano test.yml` » et remplissez le comme suit :

```
---
- name: Installation de Docker CE sur le client
  hosts: clients #Correspond au groupe de serveur présent dans notre "inventory.ini"
  become: yes #accorde les privilèges sudo

  tasks:
    - name: Mise à jour du cache des paquets
      apt:
        update_cache: yes
        become: yes #accorde les privilèges sudo
      # La tâche "apt" avec l'option "update_cache" met à jour le cache des paquets sur le client.

    - name: Installation des dépendances pour Docker
      apt:
        name:
          - apt-transport-https
          - ca-certificates
          - curl
          - gnupg-agent
          - software-properties-common
        become: yes #accorde les privilèges sudo
      # La tâche "apt" installe les dépendances nécessaires pour Docker, y compris les outils HTTPS, Curl, etc.

    - name: Ajout de la clé GPG du référentiel Docker
      apt_key:
        url: https://download.docker.com/linux/ubuntu/gpg
        state: present #l'état "Présent" signifie que si l'élément n'est pas créé, il le sera, s'il est présent, il sera mis à jour
        become: yes #accorde les privilèges sudo
      # La tâche "apt_key" ajoute la clé GPG du référentiel Docker pour la vérification de l'authenticité des paquets.

    - name: Ajout du référentiel Docker aux sources APT
      apt_repository:
        repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu {{ ansible_distribution_release }} stable
        state: present #l'état "présent" signifie que si l'élément n'est pas créé, il le sera, s'il est présent, il sera mis à jour
        become: yes #accorde les privilèges sudo
      # La tâche "apt_repository" ajoute le référentiel Docker aux sources APT sur le client.

    - name: Installation de Docker CE
      apt:
        name: docker-ce
        state: present
        become: yes #accorde les privilèges sudo
      # La tâche "apt" installe Docker CE sur le client.
```

(Le fichier est dispo dans le répertoire « Fichiers » de teams dans la section « Général »). Je l'ai commenté pour plus de facilité de compréhension.

#### 4) Exécution de notre installation Ansible

Avant toute chose, assurez-vous d'être dans votre répertoire « AnsibleLearning »

Ca y est, nous disposons enfin de tout les éléments nécessaires pour notre installation de Docker via Ansible ! Lancez votre installation via la commande suivante :

```
ubuntu02@ubuntu02-Virtual-Machine:~/AnsibleLearning$ ansible-playbook -i inventory.ini -K install_docker.yml
```

La structure de la commande fonctionne comme suit :

ansible-playbook -i votre\_fichier.ini -K votre\_fichier.yml

Détaillons la commande :

-i : permet de spécifier un fichier inventory

-K : permet de demander le mot de passe Sudo **de l'hôte distant** car certaines opérations le nécessitent

Dans notre cas cela donne : ansible-playbook -i inventory.ini -K test.yml

```
ubuntu02@ubuntu02-Virtual-Machine:~/AnsibleLearning$ ansible-playbook -i inventory.ini -K test.yml
BECOME password: 
```

Ici, le mot de passe a rentrer est celui du user de notre hôte distant, à savoir « ansible ». Dans notre tutoriel le mot de passe est « test ». Si tout se passe bien, l'installation se lance :

```
PLAY [Installation de Docker CE sur le client] *****
TASK [Gathering Facts] *****
Enter passphrase for key '/home/ubuntu02/.ssh/id_rsa': 
```

On vous demande la paraphrase de votre clé privé « test\_rsa », dans notre tuto il s'agit de « test ». Si tout va bien, vous aurez ce message :

```
ok: [ansible_client]
```

Puis ansible passe à la suite de l'installation, voici le résumé complet :

```
PLAY [Installation de Docker CE sur le client] *****
TASK [Gathering Facts] *****
Enter passphrase for key '/home/ubuntu02/.ssh/id_rsa':
ok: [ansible_client]
TASK [Mise à jour du cache des paquets] *****
changed: [ansible_client]
TASK [Installation des dépendances pour Docker] *****
ok: [ansible_client]
TASK [Ajout de la clé GPG du référentiel Docker] *****
ok: [ansible_client]
TASK [Ajout du référentiel Docker aux sources APT] *****
ok: [ansible_client]
TASK [Installation de Docker CE] *****
ok: [ansible_client]
PLAY RECAP *****
ansible_client : ok=6  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
ubuntu02@ubuntu02-Virtual-Machine:~/AnsibleLearning$
```

**ATTENTION** : Ne soyez pas surpris si vous avez plus / moins d'éléments en vert ou en jaune que moi, j'avais déjà effectué l'installation avant de réaliser cette documentation.

### Test « Post-install » de Docker sur « Ansible (client) » :

Pour tester que notre installation s'est bien passée, vous pouvez tenter de lancer une commande « `sudo docker pull ubuntu:latest` » sur « Ansible (client) » :

```
ansible@ansible-Virtual-Machine:~$ sudo docker pull ubuntu:latest
[sudo] Mot de passe de ansible :
latest: Pulling from library/ubuntu
37aaf24cf781: Pull complete
Digest: sha256:9b8dec3bf938bc80fbc758d856e96fdfab5f56c39d44b0cff351e847bb1b01ea
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
ansible@ansible-Virtual-Machine:~$
```

Si j'effectue la commande « `Sudo docker image ls` » je devrais voir Ubuntu :

```
ansible@ansible-Virtual-Machine:~$ sudo docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    3565a89d9e81   2 weeks ago   77.8MB
ansible@ansible-Virtual-Machine:~$
```

Si c'est le cas, c'est que notre installation a bien été effective sur « Ansible (client) ».