

# TP 1 – Préparation de l’environnement

---

## Objectifs

---

- Comprendre la topologie des ateliers et comment accéder à l’environnement.
- Savoir configurer l’accès ssh de la machine de management ansible vers les nodes cibles.

## Guide

---

### Environnement de travail

---

Dans les labs de cette formation vous aurez accès aux hôtes suivants:

Nom d’hôte	Rôle	OS
ansible	Hôte de contrôle Ansible	Alma Linux 9
node1	Hôte cible	Ubuntu 22.04
node2	Hôte cible	Ubuntu 22.04

## Créer l’environnement

---

- VirtualBox n'autorise que les adresses IP de la plage 192.168.56.0/21 à être attribuées aux VMs. Si d'autres plages sont souhaitées, elles peuvent être activées en créant `/etc/vbox/networks.conf` et en y spécifiant les plages autorisées.

Créez `/etc/vbox/networks.conf` et ajoutez le ligne suivante pour autoriser l’attribution de toute IP.

*\* 0.0.0.0/0 ::/0*

```
brahim@Training:~$ sudo mkdir /etc/vbox
brahim@Training:~$ sudo vim /etc/vbox/networks.conf
brahim@Training:~$
brahim@Training:~$ cat /etc/vbox/networks.conf
* 0.0.0.0/0 ::/0
brahim@Training:~$
brahim@Training:~$
```

- Git clonez l'environnement des labs depuis le dépôt Git ci-dessous :

```
git clone https://github.com/brahimhamdi/ansible-lab
```

- Sous le répertoire ansible-lab, démarrez le déploiement des 4 VMs :

```
cd ansible-lab  
vagrant up
```

```
brahim@Training:~$ git clone https://github.com/brahimhamdi/ansible-lab
Clonage dans 'ansible-lab'...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (71/71), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 71 (delta 18), reused 71 (delta 18), pack-reused 0
Réception d'objets: 100% (71/71), 8.55 Kio | 8.55 Mio/s, fait.
Résolution des deltas: 100% (18/18), fait.
brahim@Training:~$ cd ansible-lab/
brahim@Training:~/ansible-lab$
brahim@Training:~/ansible-lab$ vagrant up
Bringing machine 'node1' up with 'virtualbox' provider...
Bringing machine 'node2' up with 'virtualbox' provider...
==> node1: Importing base box 'generic/ubuntu2204'...
==> node1: Matching MAC address for NAT networking...
==> node1: Checking if box 'generic/ubuntu2204' version '4.2.16' is up to date...
==> node1: A newer version of the box 'generic/ubuntu2204' for provider 'virtualbox' is
==> node1: available! You currently have version '4.2.16'. The latest is version
==> node1: '4.3.4'. Run `vagrant box update` to update.
==> node1: Setting the name of the VM: ansible-lab_node1_1698908564406_72309
==> node1: Clearing any previously set network interfaces...
==> node1: Preparing network interfaces based on configuration...
    node1: Adapter 1: nat
    node1: Adapter 2: hostonly
==> node1: Forwarding ports...
    node1: 22 (guest) => 2222 (host) (adapter 1)
==> node1: Running 'pre-boot' VM customizations...
==> node1: Booting VM...
==> node1: Waiting for machine to boot. This may take a few minutes...
    node1: SSH address: 127.0.0.1:2222
    node1: SSH username: vagrant
    node1: SSH auth method: private key
    node1: Warning: Connection reset. Retrying...
    node1: Warning: Remote connection disconnect. Retrying...
```

- Testez la connectivité avec node1 et node2.

```
ping 192.168.201.11  
ping ...  
...
```

- Connectez-vous à la VM d'ansible (Alma Linux 9) et installez la dernière version d'ansible.

```
sudo dnf install -y epel-release  
sudo dnf install ansible -y
```

Il est possible d'installer la dernière version depuis le dépôt github

([https://docs.ansible.com/ansible/latest/installation\\_guide/intro\\_installation.html](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html))

- Le fichier de configuration **par défaut** de ansible est /etc/ansible/ansible.cfg.  
Donnez le nom et le chemin par défaut du fichier d'inventaire ansible.

## Configuration de l'accès ssh

---

Dans cette partie, vous allez configurer l'accès ssh d'ansible management vers les 3 nodes cibles. Il y a deux méthodes d'authentification ssh : par mot de passe (par défaut) et par clés.

Dans un premier temps, vous allez tester l'authentification par mot de passe, mais à la fin vous allez configurer et tester l'authentification par clés, c'est la méthode la plus sûr.

- Toujours sous ansible management, testez l'accès ssh vers les 2 nodes :

```
ssh vagrant@192.168.201.11  
ssh vagrant@192.168.201.12
```

```
brahim@Training:~/ansible-lab$ ssh vagrant@192.168.201.11  
The authenticity of host '192.168.201.11 (192.168.201.11)' can't be established.  
ED25519 key fingerprint is SHA256:6Gtq2Wvb6BI7Fo9HdBxU210cPvBBBxX5kC2SxsoWryM.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.201.11' (ED25519) to the list of known hosts.  
vagrant@192.168.201.11's password:  
vagrant@node1:~$  
vagrant@node1:~$ exit  
logout  
Connection to 192.168.201.11 closed.  
brahim@Training:~/ansible-lab$ ssh vagrant@192.168.201.12  
The authenticity of host '192.168.201.12 (192.168.201.12)' can't be established.  
ED25519 key fingerprint is SHA256:qVL9of1Dpc86D5uyhofs445i4G5GMiX8UN/5hcAqRQk.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.201.12' (ED25519) to the list of known hosts.  
vagrant@192.168.201.12's password:  
vagrant@node2:~$  
vagrant@node2:~$ exit  
logout  
Connection to 192.168.201.12 closed.  
brahim@Training:~/ansible-lab$
```

**Si** l'une des commandes précédentes affiche une erreur sur les permissions, c'est parce que l'authentification par mot de passe sur ces VMs est désactivée.

Pour résoudre ce problème, il faut se connecter à ces machines et activer l'authentification par mot de passe :

```
vagrant ssh node1
sudo vi /etc/ssh/sshd_config
PasswordAuthentication yes
sudo systemctl reload sshd
```

- Si tout va bien, générez une paire de clés RSA avec :
  - passphrase vide
  - gardez les noms et chemins par défaut des clés RSA

```
ssh-keygen -t rsa
```

```
brahim@Training:~/ansible-lab$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/brahim/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/brahim/.ssh/id_rsa
Your public key has been saved in /home/brahim/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:nQrDZuIsUff9B9qnCqdiM+9YGYs3SKgRtWKIQSufC+c brahim@Training
The key's randomart image is:
+---[RSA 3072]-----+
|+=0+.      .      |
|oo*oo+ . o .      |
|=.oE+ + o o o      |
|. + o = . o =      |
|. . o X S +        |
|  + + 0 o          |
| . B + o           |
| o B               |
| ..o               |
+----[SHA256]-----+
brahim@Training:~/ansible-lab$
brahim@Training:~/ansible-lab$ ls /home/brahim/.ssh/id_rsa*
/home/brahim/.ssh/id_rsa  /home/brahim/.ssh/id_rsa.pub
brahim@Training:~/ansible-lab$
brahim@Training:~/ansible-lab$ □
```

Par défaut, ssh-keygen créera une paire de clés RSA (une clé publique et une clé privée) de 2048 bits, ce qui est suffisamment sûr pour la plupart des cas d'utilisation (vous pouvez éventuellement passer l'indicateur -b 4096 pour créer une clé plus grande de 4096 bits).

Quel est le nom et le chemin de chaque clé ?

- L'étape suivante consiste à placer la clé publique sur votre serveur afin que vous puissiez utiliser l'authentification basée sur la clé SSH pour vous connecter.

La façon la plus rapide de copier votre clé publique sur l'hôte cible est d'utiliser un utilitaire appelé `ssh-copy-id`. Cette méthode est fortement recommandée si elle est disponible. Si vous ne disposez pas de `ssh-copy-id` sur votre machine cliente, vous pouvez utiliser l'une des deux méthodes alternatives (copie via SSH basé sur un mot de passe, ou copie manuelle de la clé).

Pour que l'utilitaire `ssh-copy-id` fonctionne, vous devez déjà disposer d'un accès SSH à votre serveur, basé sur un mot de passe.

Copiez la clé publique sur les 3 nodes cibles :

```
ssh-copy-id -i /$HOME/.ssh/id_rsa.pub vagrant@192.168.201.11  
...
```

```
brahim@Training:~/ansible-lab$ ssh-copy-id -i /$HOME/.ssh/id_rsa.pub vagrant@192.168.201.11
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/brahim/.ssh/id_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
vagrant@192.168.201.11's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'vagrant@192.168.201.11'"
and check to make sure that only the key(s) you wanted were added.

brahim@Training:~/ansible-lab$ ssh-copy-id -i /$HOME/.ssh/id_rsa.pub vagrant@192.168.201.12
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/brahim/.ssh/id_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
vagrant@192.168.201.12's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'vagrant@192.168.201.12'"
and check to make sure that only the key(s) you wanted were added.

brahim@Training:~/ansible-lab$ 
brahim@Training:~/ansible-lab$ □
```

Ensuite, l'utilitaire recherchera sur votre compte local la clé `id_rsa.pub` que vous avez créée précédemment. Lorsqu'il trouvera la clé, il vous demandera le mot de passe du compte de l'utilisateur distant.

L'utilitaire se connectera au compte sur l'hôte distant en utilisant le mot de passe que vous avez fourni. Il copiera ensuite le contenu de votre clé `~/ .ssh/id_rsa.pub` dans le fichier `~/ .ssh/authorized_keys` du compte distant.

- Pour tester, vous devriez pouvoir vous connecter à l'hôte distant *sans* le mot de passe du compte distant.

Le processus initial est le même que pour l'authentification par mot de passe :

```
brahim@Training:~/ansible-lab$ ssh vagrant@192.168.201.11
Last login: Thu Nov  2 07:10:30 2023 from 192.168.201.1
vagrant@node1:~$
vagrant@node1:~$
logout
Connection to 192.168.201.11 closed.
brahim@Training:~/ansible-lab$ ssh vagrant@192.168.201.12
Last login: Thu Nov  2 07:10:45 2023 from 192.168.201.1
vagrant@node2:~$
vagrant@node2:~$
logout
Connection to 192.168.201.12 closed.
brahim@Training:~/ansible-lab$ 
```