École Mohammadia d'Ingénieurs

Atelier DevOps / CI-CD
Automatisation de déploiement : Ansible & Jenkins

Rapport d'Atelier — Déploiement WAR avec Ansible et exécution via Jenkins

---

Élèves :
Sami FAOUZI

Encadré par :
Asmaa RETBI

2ème année Génie Informatique
23 décembre 2025

Table des matières

1 Tentative initiale sous Windows + VirtualBox (non finalisée)

1.1 Installation et validation d'Ansible sur la VM Ubuntu (VirtualBox)

Après installation des paquets nécessaires, Ansible a été vérifié via la commande `ansible -version`.

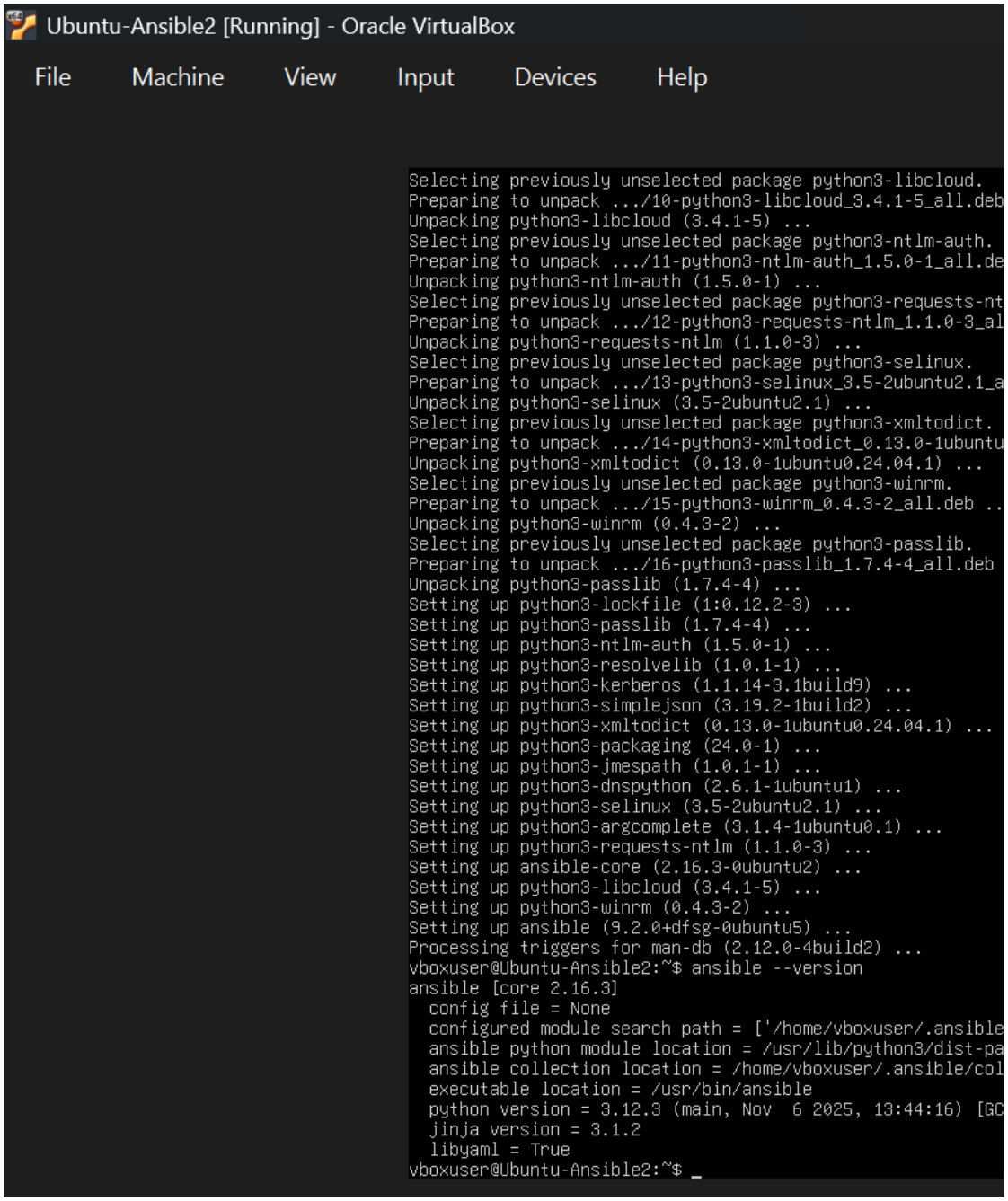

FIGURE 1 – Installation d'Ansible et exécution de `ansible -version` (VM VirtualBox)

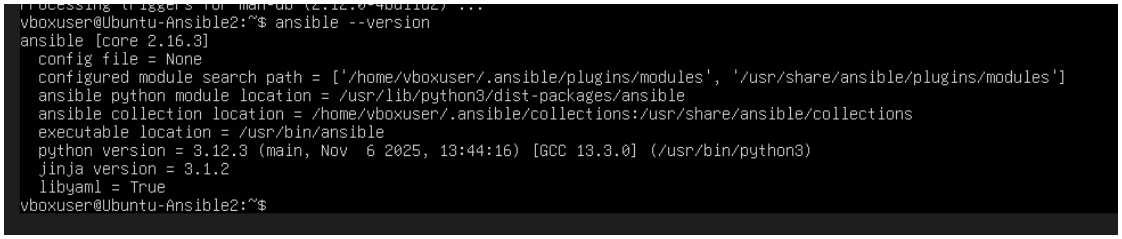

FIGURE 2 – Détail : sortie de `ansible -version` confirmant l'installation

1.2 Création de l'utilisateur `user_sup` côté VM Ubuntu

Un utilisateur `user_sup` a été créé sur la VM, puis ajouté au groupe `sudo` et validé via les commandes de contrôle.

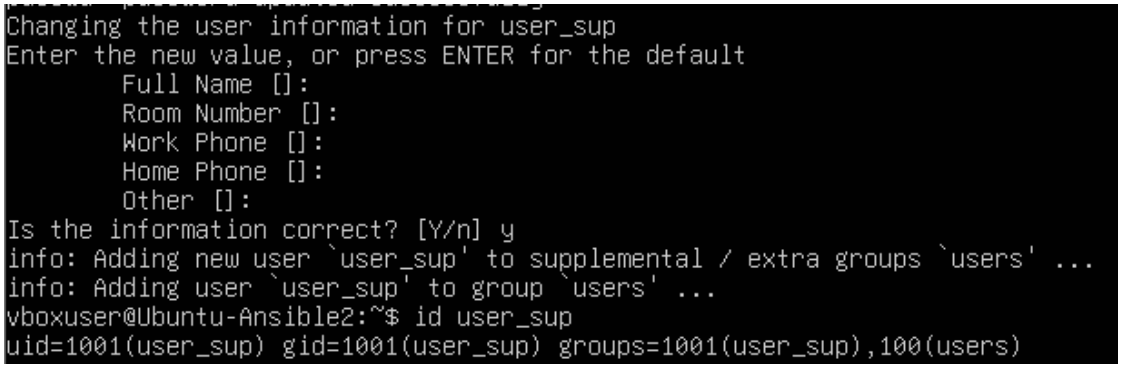

FIGURE 3 – Création de `user_sup` sur la VM et vérification via `id user_sup`

FIGURE 4 – Ajout au groupe `sudo` et validation des droits (commande `sudo -l -U`)

1.3 Préparation du côté Windows (compte + droits + SSH Server)

Pour permettre la connexion distante depuis la VM Ubuntu, un compte `user_sup` a été préparé sur Windows, puis ajouté au groupe **Administrateurs**. Ensuite, le service **OpenSSH Server** a été activé et la règle pare-feu vérifiée. Enfin, le port 22 a été contrôlé en écoute.

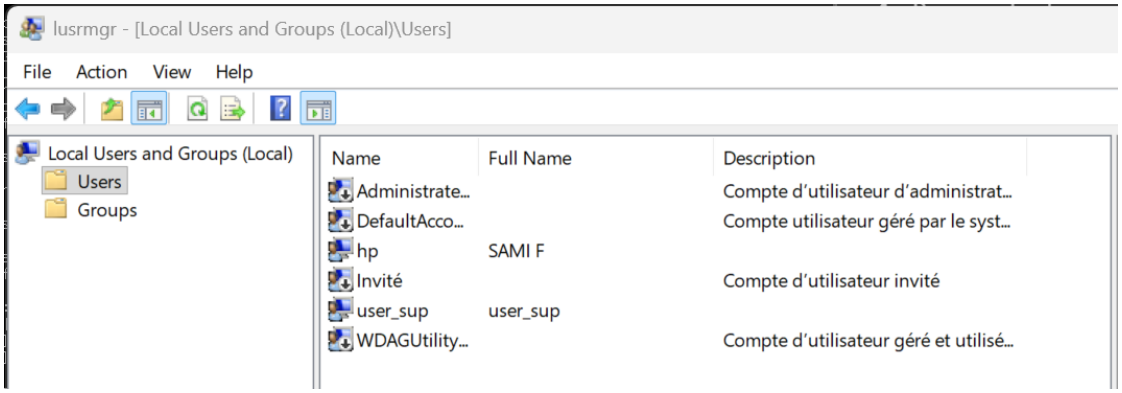

FIGURE 5 – Création du compte `user_sup` sur Windows (Local Users and Groups)

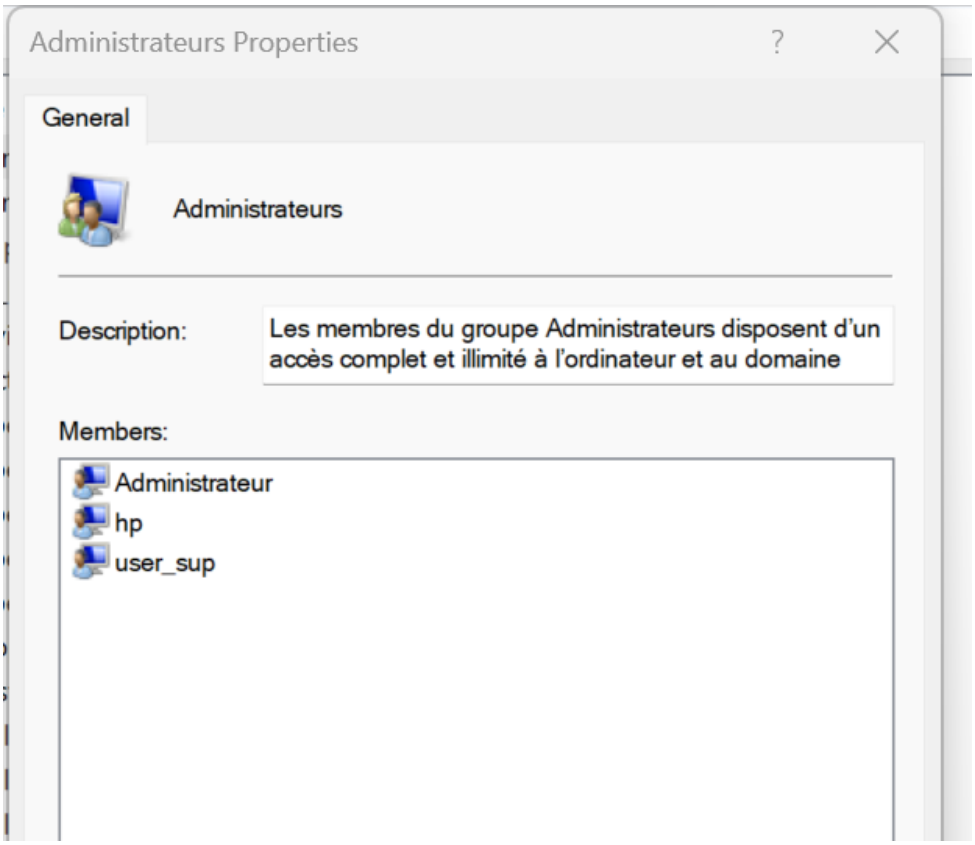

FIGURE 6 – Ajout de `user_sup` au groupe Administrateurs

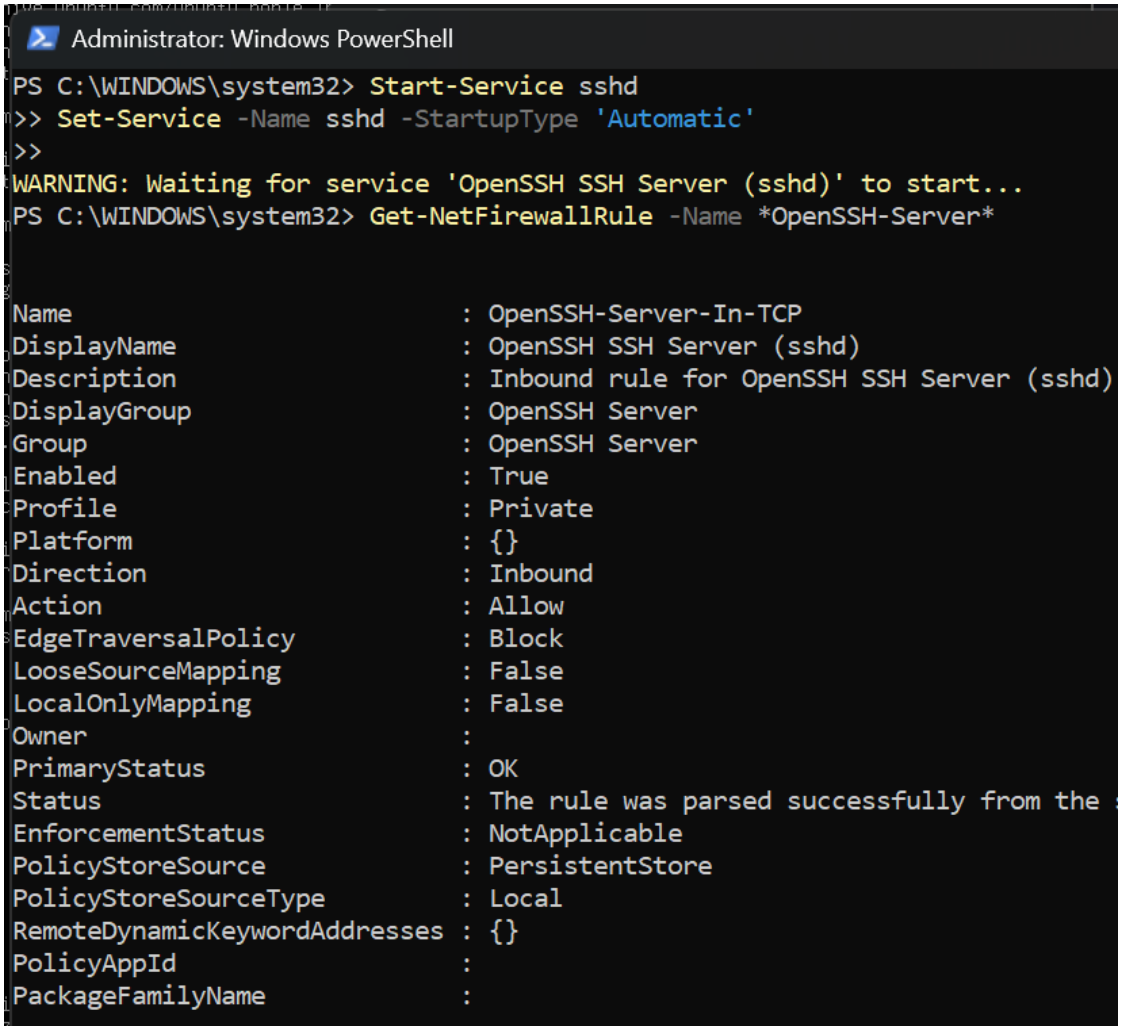

FIGURE 7 – Activation du service SSH (`sshd`) et vérification des règles pare-feu

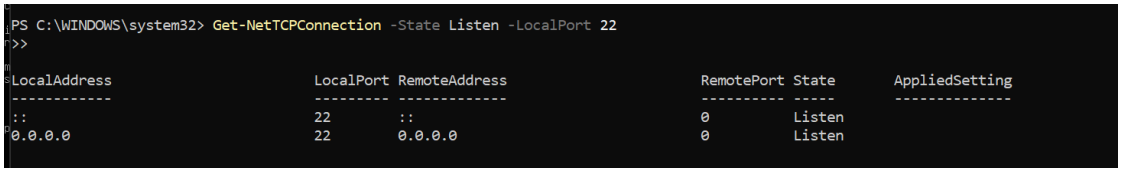

FIGURE 8 – Preuve : le port 22 est en état `Listen` côté Windows

1.4 Mise en place d'authentification par clé SSH (VM → Windows)

Une clé SSH a été générée sur la VM Ubuntu, puis la clé publique a été ajoutée au fichier `authorized_keys` dans le profil `C:\Users\user_sup\.ssh`.

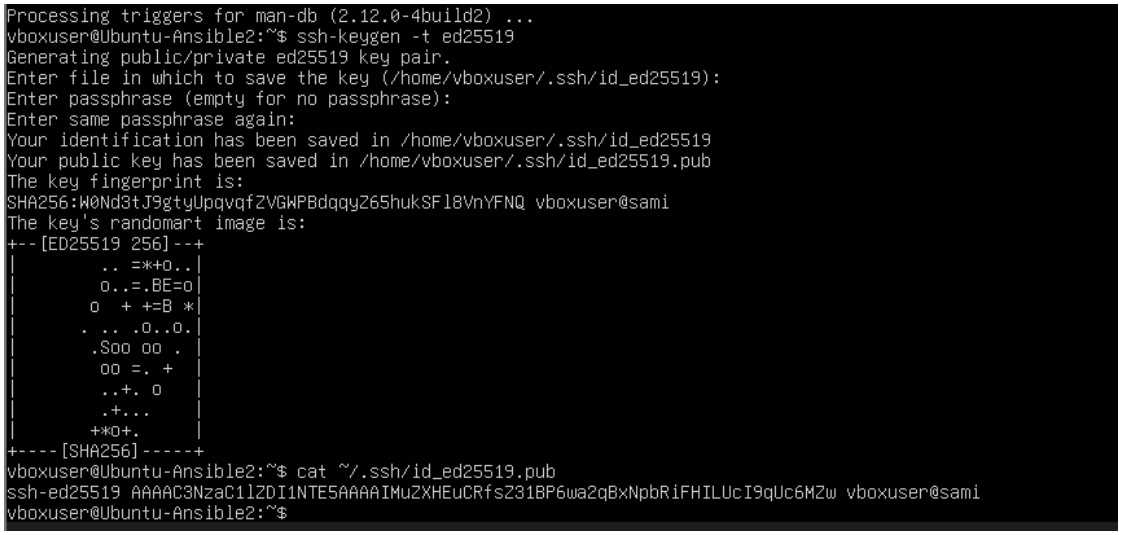

FIGURE 9 – Génération de clé `ed25519` et affichage de la clé publique

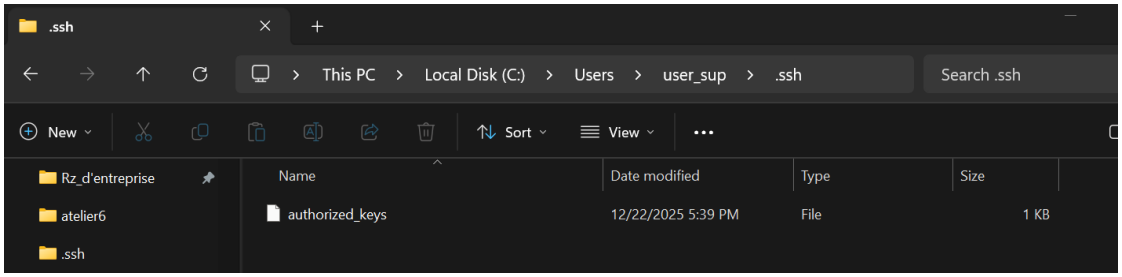

FIGURE 10 – Fichier `authorized_keys` créé côté Windows pour `user_sup`

1.5 Limite rencontrée et décision de bascule

Malgré ces étapes de préparation (*Ansible installé*, *compte créé*, *SSH activé*, *port 22 en écoute*, *clé configurée*), la suite du travail n'a pas pu être finalisée sous Windows/VirtualBox à cause de **problèmes récurrents de la VM** (instabilités et blocages répétés).

Ubuntu (solution stable), comme présenté dans la partie suivante du rapport.

2 Préparation de l'Atelier

2.1 Objectif

L'objectif de cet atelier est de mettre en place une infrastructure d'automatisation de déploiement utilisant Ansible comme outil de configuration et Jenkins pour l'orchestration via pipeline CI/CD.

2.2 Architecture Utilisée

— **Machine Contrôleur (VM Ansible) :** Ubuntu Server 24.04 LTS sur KVM/-QEMU
— **Machine Cible (Host Ubuntu) :** Ubuntu Desktop (machine physique)

— **IP VM :** 192.168.122.61
— **IP Host :** 192.168.122.1 (virbr0)
— **Utilisateur SSH :** user_sup

3 Étape 0 : Création de la VM et Installation d'Ansible

3.1 Création de la Machine Virtuelle (KVM / virt-manager)

```
sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-
    utils virt-manager -y
sudo systemctl enable --now libvirtd
sudo usermod -aG libvirt $USER
sudo usermod -aG kvm $USER
```

Listing 1 – Installation de KVM

3.2 Installation d'Ubuntu Server

— ISO utilisé : Ubuntu Server 24.04.3 LTS
— RAM allouée : 2 GB
— Disque : 25 GB
— Réseau : NAT via virbr0 (192.168.122.0/24)

3.3 Installation d'Ansible sur la VM

```
sudo apt update
sudo apt install ansible -y
ansible --version
```

Listing 2 – Installation Ansible sur la VM

FIGURE 11 – Ansible installé et fonctionnel sur la VM

4 Étape 1 : Création du Super Utilisateur (user_sup)

4.1 Sur la VM (Ansible)

L'utilisateur `user_sup` a été créé lors de l'installation avec les droits sudo.

FIGURE 12 – Création de user_sup avec droits sudo sur la VM

4.2 Sur le Host Ubuntu

```
sudo adduser user_sup
sudo usermod -aG sudo user_sup
```

Listing 3 – Création user_sup sur le Host

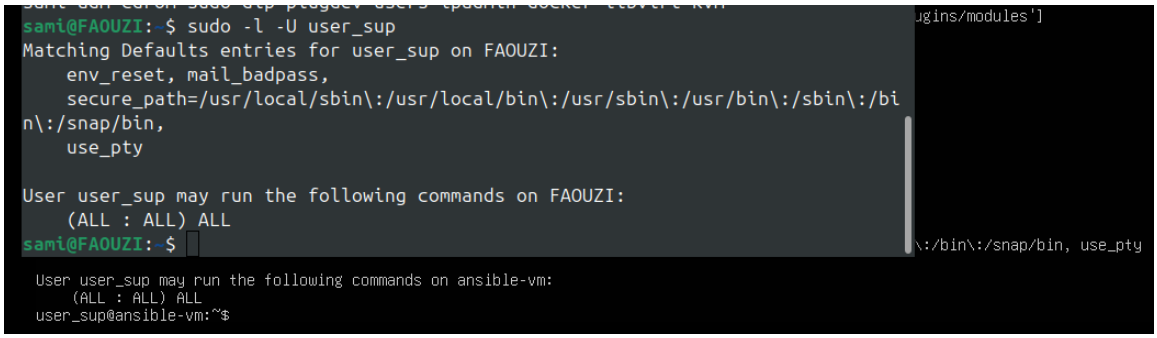

FIGURE 13 – Création de user_sup avec droits sudo sur la machine Host

5 Étapes 2–3 : Configuration SSH (VM → Host)

5.1 Activation du serveur SSH sur le Host

```
sudo apt install openssh-server -y
sudo systemctl enable --now ssh
sudo ss -tlnp | grep :22
```

Listing 4 – Installation et activation SSH

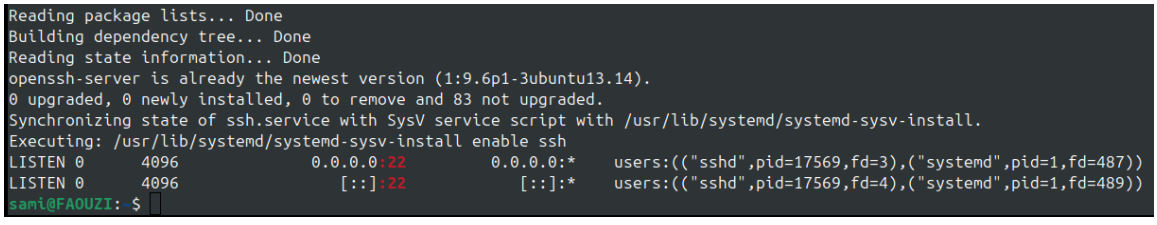

FIGURE 14 – Activation du serveur SSH sur le Host (port 22)

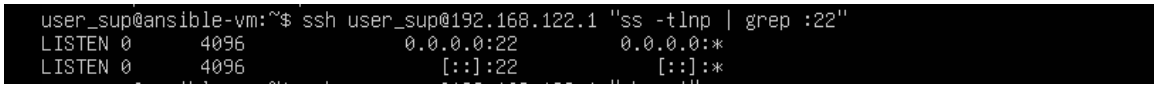

FIGURE 15 – Preuve que SSH est actif sur le Host (écoute sur le port 22)

5.2 Génération et copie de clé SSH

```
ssh-keygen -t rsa -b 4096
ssh-copy-id user_sup@192.168.122.1
```

Listing 5 – Génération + copie de clé SSH

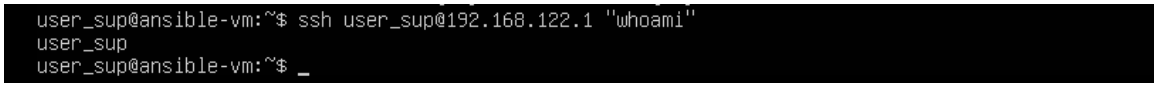

FIGURE 16 – Preuve de connexion SSH de la VM vers le Host

6 Étape 4 : Inventory Ansible + Tests

6.1 Création du fichier inventory

```
sudo mkdir -p /etc/ansible
echo "[ubuntu_hosts]" > ~/inventory
echo "192.168.122.1" >> ~/inventory
```

Listing 6 – Création de l'inventory

6.2 Test de connectivité (ping Ansible)

```
ansible all -i ~/inventory -m ping -u user_sup
```

Listing 7 – Ping Ansible

FIGURE 17 – Résultat du ping Ansible (SUCCESS)

6.3 Test de validation (shell)

```
ansible all -i ~/inventory -m shell -a "whoami && hostname" -u user_sup
```

Listing 8 – Commande shell via Ansible

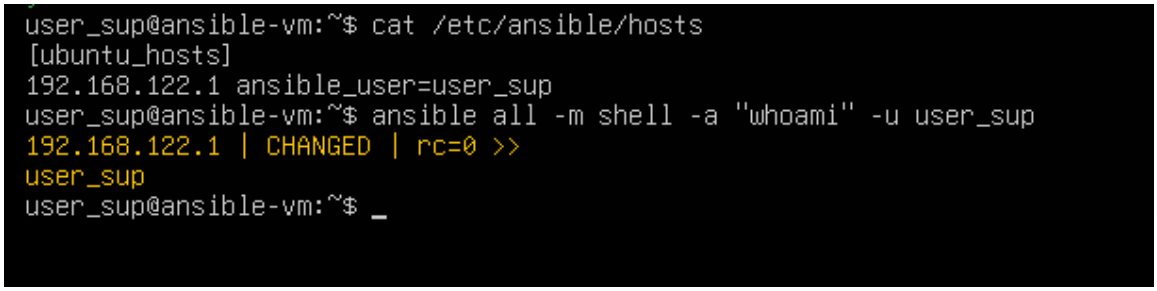

FIGURE 18 – Inventory + test de validation (whoami/hostname)

7 Étapes 5–6 : Playbook de Déploiement WAR

7.1 Préparation du dossier webapps (Host)

```
sudo mkdir -p /opt/tomcat/webapps
sudo chown user_sup:user_sup /opt/tomcat/webapps
```

Listing 9 – Création du dossier webapps

7.2 WAR de test

```
echo "Sample WAR content" > ~/app.war
```

Listing 10 – Création WAR de test

7.3 Playbook deploy-war.yml

```
---
- name: Deploiement WAR vers le serveur
  hosts: all
  vars:
    war_file: /home/user_sup/app.war
    webapps_dest: /opt/tomcat/webapps/

  tasks:
    - name: Verifier que le WAR existe sur Ubuntu
      local_action:
        module: stat
        path: "{{ war_file }}"
      register: war_check

    - name: Copier le WAR vers webapps via SSH
      copy:
        src: "{{ war_file }}"
        dest: "{{ webapps_dest }}"

    - name: Confirmer le deploiement
      debug:
        msg: "OK: WAR deploye avec succes!"
```

Listing 11 – Playbook deploy-war.yml

```
user_sup@ansible-vm:~$ cat ~/deploy-war.yml
---
- name: Deploiement WAR vers le serveur
  hosts: ubuntu_hosts
  vars:
    war_file: /home/user_sup/sample-app.war
    webapps_dest: /opt/tomcat/webapps/

  tasks:
    - name: Verifier que le WAR existe sur Ubuntu
      local_action:
        module: stat
        path: "{{ war_file }}"
      register: war_check

    - name: Copier le WAR vers webapps via SSH
      copy:
        src: "{{ war_file }}"
        dest: "{{ webapps_dest }}"

    - name: Confirmer le deploiement
      debug:
        msg: "OK: WAR deploye avec succes!"
```

FIGURE 19 – Fichier Playbook (deploy-war.yml)


```
user_sup@ansible-vm:~$ cat ~/deploy-war.yml
---
- name: Deploiement WAR vers le serveur
  hosts: ubuntu_hosts
  vars:
    war_file: /home/user_sup/sample-app.war
    webapps_dest: /opt/tomcat/webapps/

  tasks:
    - name: Verifier que le WAR existe sur Ubuntu
      local_action:
        module: stat
        path: "{{ war_file }}"
      register: war_check

    - name: Copier le WAR vers webapps via SSH
      copy:
        src: "{{ war_file }}"
        dest: "{{ webapps_dest }}"

    - name: Confirmer le deploiement
      debug:
        msg: "OK: WAR deploye avec succes!"
```


FIGURE 20 – Capture du fichier Playbook

7.4 Exécution du playbook

```
ansible-playbook -i ~/inventory ~/deploy-war.yml
```

Listing 12 – Exécution du playbook

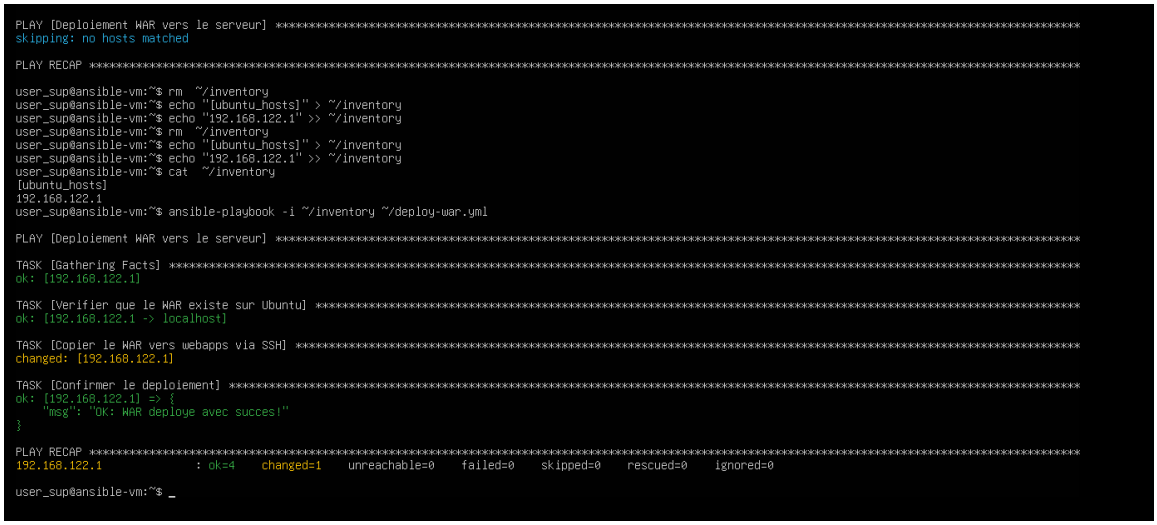

FIGURE 21 – Exécution du playbook (PLAY RECAP)

7.5 Vérification du déploiement

```
ls -la /opt/tomcat/webapps/
```

Listing 13 – Vérification sur le Host

FIGURE 22 – Preuve : WAR présent dans `/opt/tomcat/webapps`

8 Étape 7 : Jenkins et Pipeline Ansible

8.1 Installation du plugin Ansible

1. Accéder à Jenkins : `http://localhost:8080`
2. `Manage Jenkins` → `Plugins` → `Available plugins`
3. Rechercher `Ansible` et l'installer
4. Redémarrer Jenkins si nécessaire

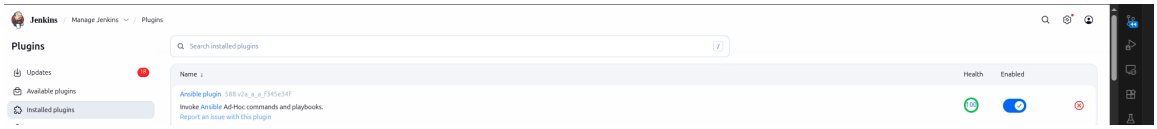

FIGURE 23 – Plugin Ansible installé sur Jenkins

8.2 Jenkinsfile (pipeline)

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                sh 'echo "WAR content" > app.war'
            }
        }
        stage('Transfer') {
            steps {
                sh 'scp -o StrictHostKeyChecking=no app.war user_sup@192.168.122.61:~/app.war'
            }
        }
        stage('Deploy') {
            steps {
                sh 'ssh -o StrictHostKeyChecking=no user_sup@192.168.122.61 "ansible-playbook -i ~/inventory ~/deploy-war.yml"'
            }
        }
    }
}
```

Listing 14 – Jenkinsfile (Pipeline)

FIGURE 24 – Jenkinsfile : configuration du pipeline

8.3 Résultat d'exécution (console)

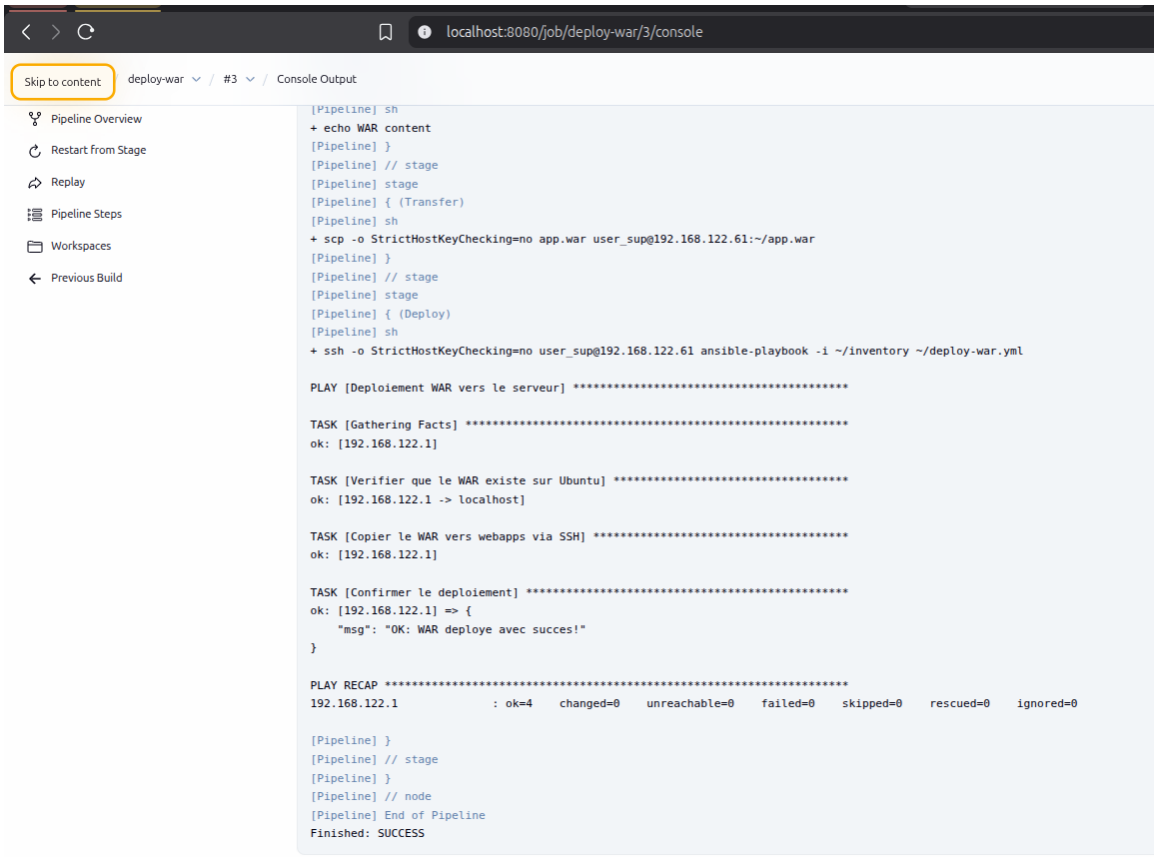

FIGURE 25 – Console Output Jenkins : SUCCESS

8.4 Vérification finale du WAR (via pipeline Jenkins)

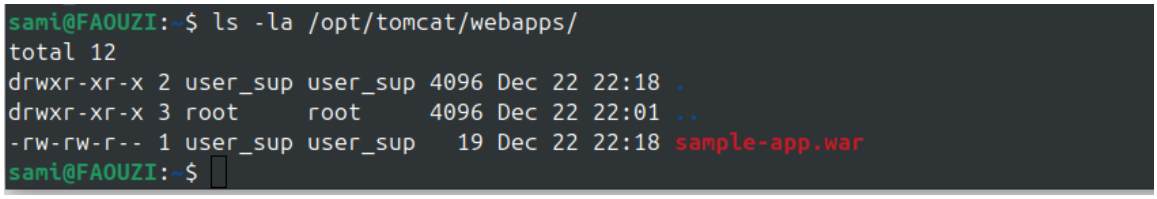

FIGURE 26 – WAR présent dans webapps après exécution Jenkins

9 Observations et Difficultés Rencontrées

1. **VirtualBox vs KVM :** VirtualBox posait problème (conflit/contraintes). Solution : utilisation de KVM/virt-manager.
2. **Libvirt daemon :** service `libvirtd` non démarré initialement ⇒ activation manuelle.
3. **/etc/ansible inexistant :** création manuelle, puis utilisation d'un inventory local `~/inventory`.

4. **Format inventory :** erreurs de parsing corrigées en écrivant le fichier ligne par ligne.
5. **SSH Jenkins :** erreurs “Host key verification failed” et “Permission denied” ⇒ réglées via clés SSH + options SSH.

— Fin du Rapport —