ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT
TECHNIQUE DE MOHAMMEDIA
UNIVERSITÉ HASSAN II DE CASABLANCA

École Normale Supérieure de
l'Enseignement Technique - Mohammedia

Department of Information and Mathematics

IT Engineering, Cybersecurity and Digital Trust engineering

Module : Virtualisation et Cloud computing

TP4 - Mise en place de Docker et Cockpit sur une instance Ubuntu EC2

Élèves :
Yasser Namez

Enseignant :
Pr. Azeddine KHIAT

2 décembre 2025

Table des matières

1 Introduction

Ce rapport présente la réalisation du TP4 ayant pour objectif la mise en place d'un environnement de conteneurisation Docker et d'une interface d'administration Cockpit sur une instance Ubuntu hébergée sur AWS EC2.

1.1 Objectifs du TP

- Créer et configurer une machine virtuelle Ubuntu sur AWS EC2
- Installer et utiliser Docker pour la conteneurisation d'applications
- Déployer Cockpit, une interface web d'administration Linux
- Superviser et gérer les conteneurs et services via une interface graphique

1.2 Technologies utilisées

- **AWS EC2** : Service de machines virtuelles dans le cloud
- **Ubuntu 22.04 LTS** : Système d'exploitation Linux
- **Docker** : Plateforme de conteneurisation
- **Cockpit** : Interface web d'administration système

2 Réalisation pratique

2.1 Étape 1 : Création de l'instance EC2

2.1.1 Configuration de l'instance

J'ai créé une instance EC2 avec les paramètres suivants :

- **AMI :** Ubuntu Server 22.04 LTS
- **Type d'instance :** t2.micro
- **Clé SSH :** tp4-docker-key.pem
- **Groupe de sécurité :** Ports 22 (SSH) et 9090 (Cockpit) ouverts

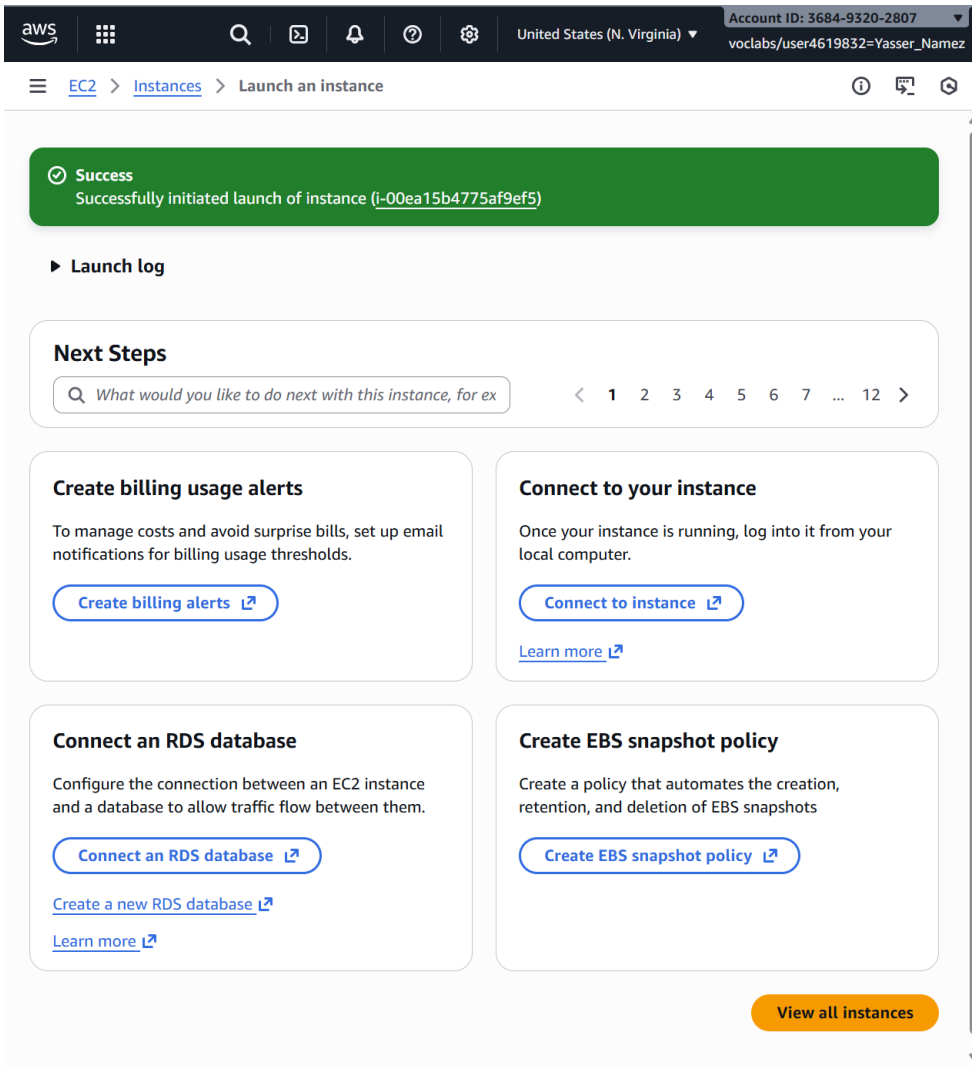

FIGURE 1 – Création de l'instance EC2 sur AWS

2.1.2 Configuration du groupe de sécurité

Les règles de sécurité configurées permettent l'accès SSH et Cockpit :

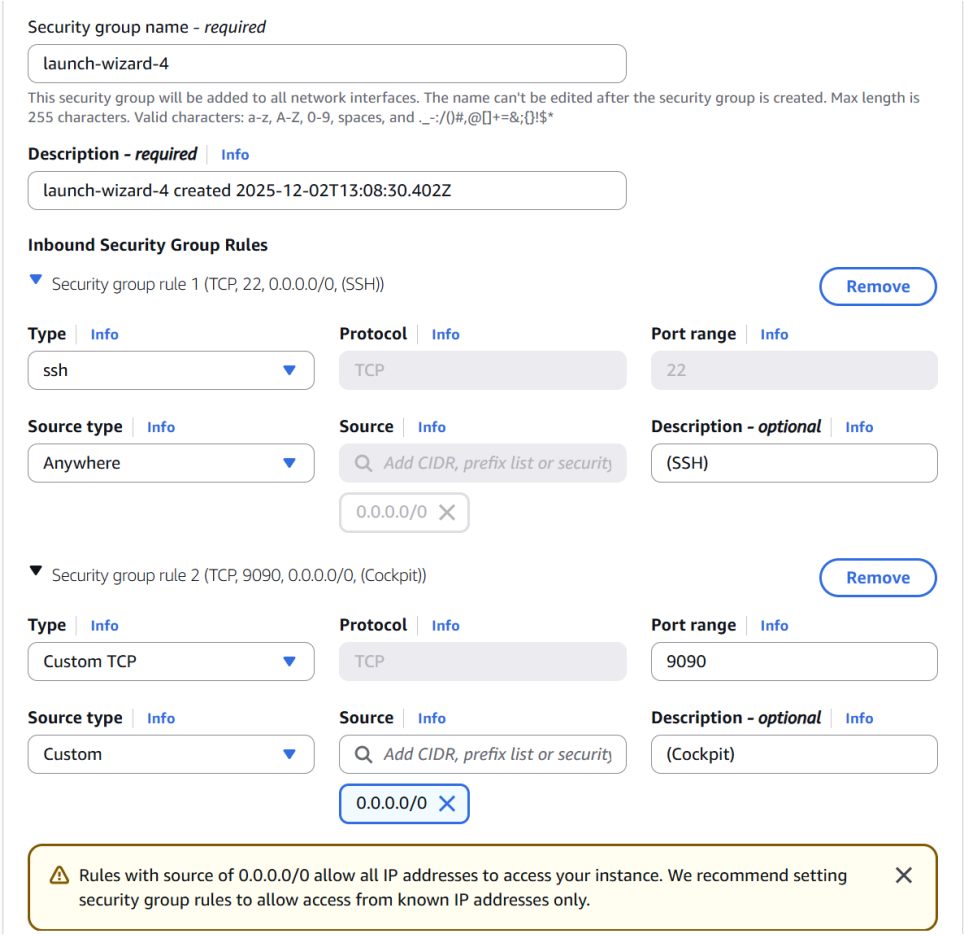

FIGURE 2 – Configuration du groupe de sécurité

2.2 Étape 2 : Connexion à l'instance

Commande utilisée pour se connecter à l'instance :

```
ssh -i "KEY_TP4_VM_LINUX.pem" ubuntu@ec2-3-84-104-201.compute-1.amazonaws.com
```

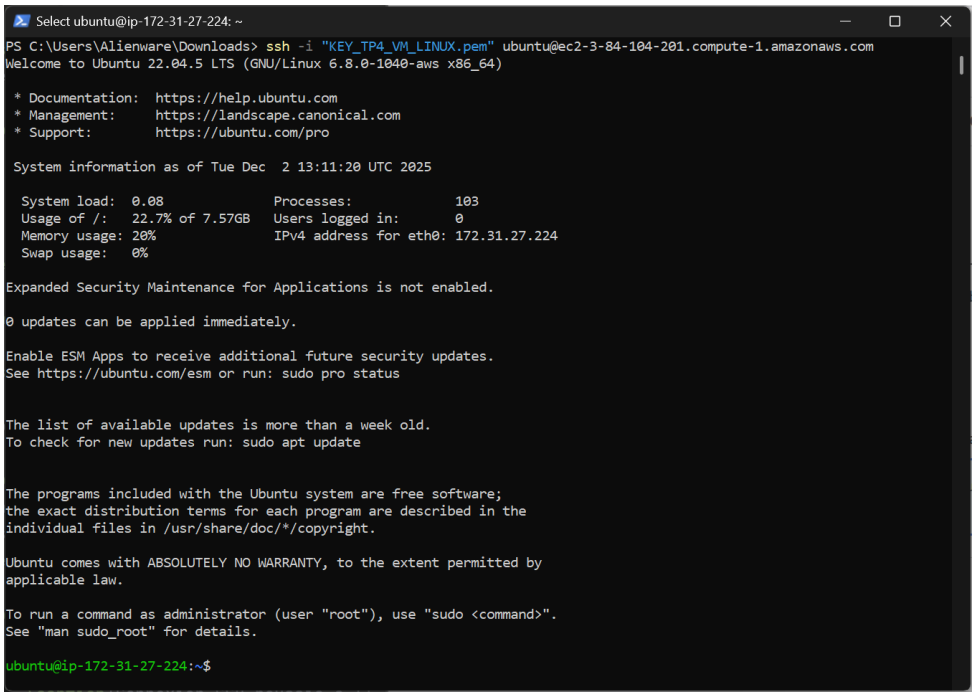

FIGURE 3 – Connexion SSH réussie à l'instance Ubuntu

2.3 Étape 3 : Mise à jour du système

Commandes exécutées :

```
sudo apt update && sudo apt upgrade -y
```

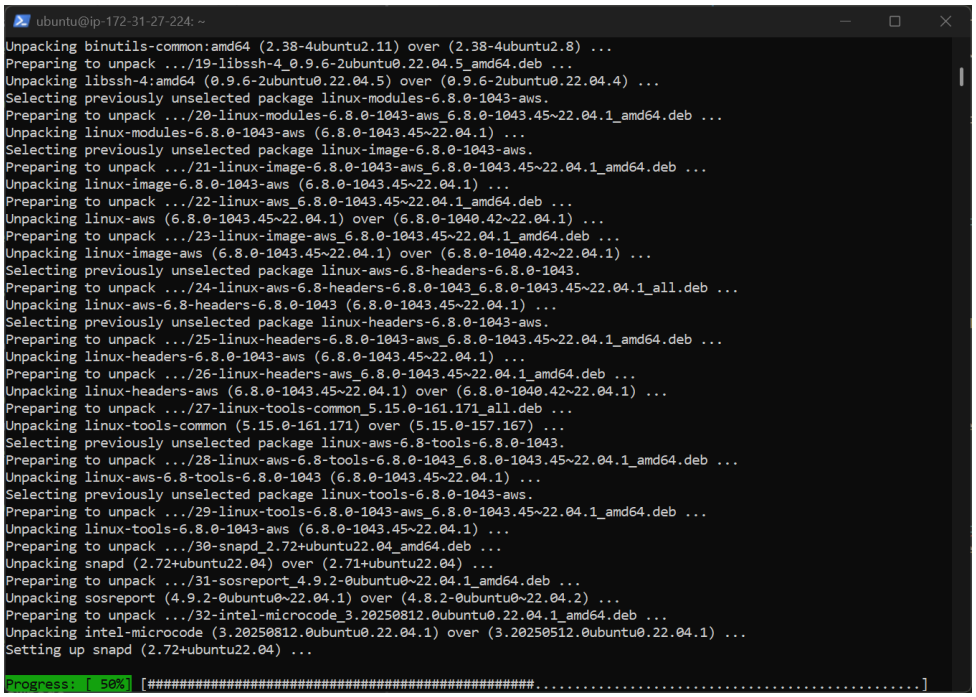

FIGURE 4 – Mise à jour du système Ubuntu

2.4 Étape 4 : Installation de Docker

2.4.1 Installation

Commandes exécutées :

```
sudo apt install -y docker.io
sudo systemctl enable docker
sudo systemctl start docker
sudo usermod -aG docker ubuntu
```

2.4.2 Test de Docker

Après reconnexion, test avec hello-world :

```
docker run hello-world
```

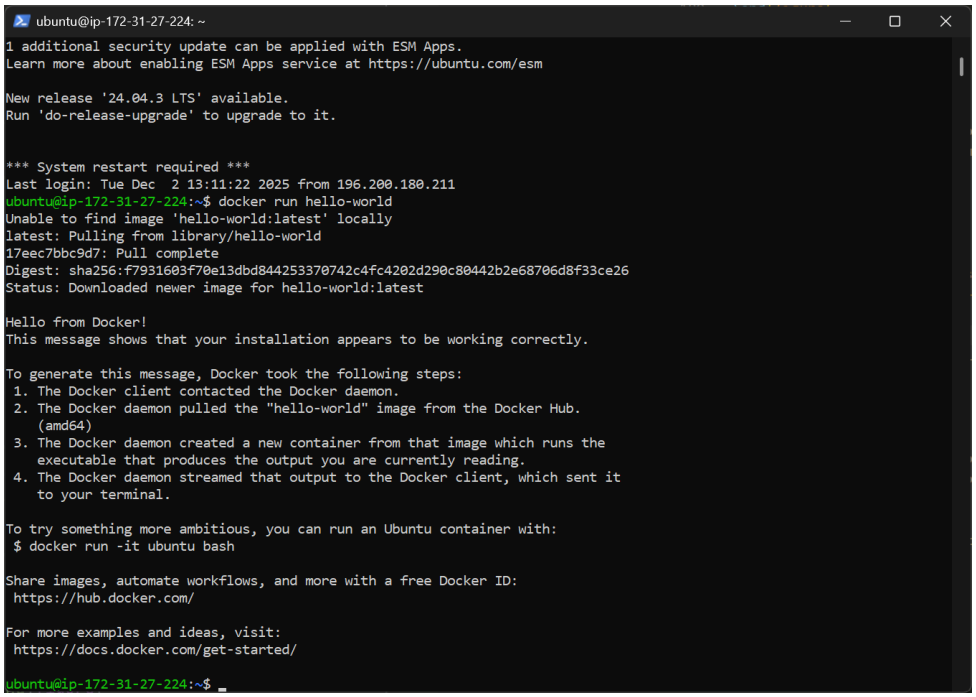

FIGURE 5 – Test de Docker avec hello-world

2.4.3 Déploiement d'un conteneur Nginx

Pour démonstration, j'ai déployé un conteneur Nginx :

```
docker run -d --name test-nginx -p 8080:80 nginx
docker ps
```

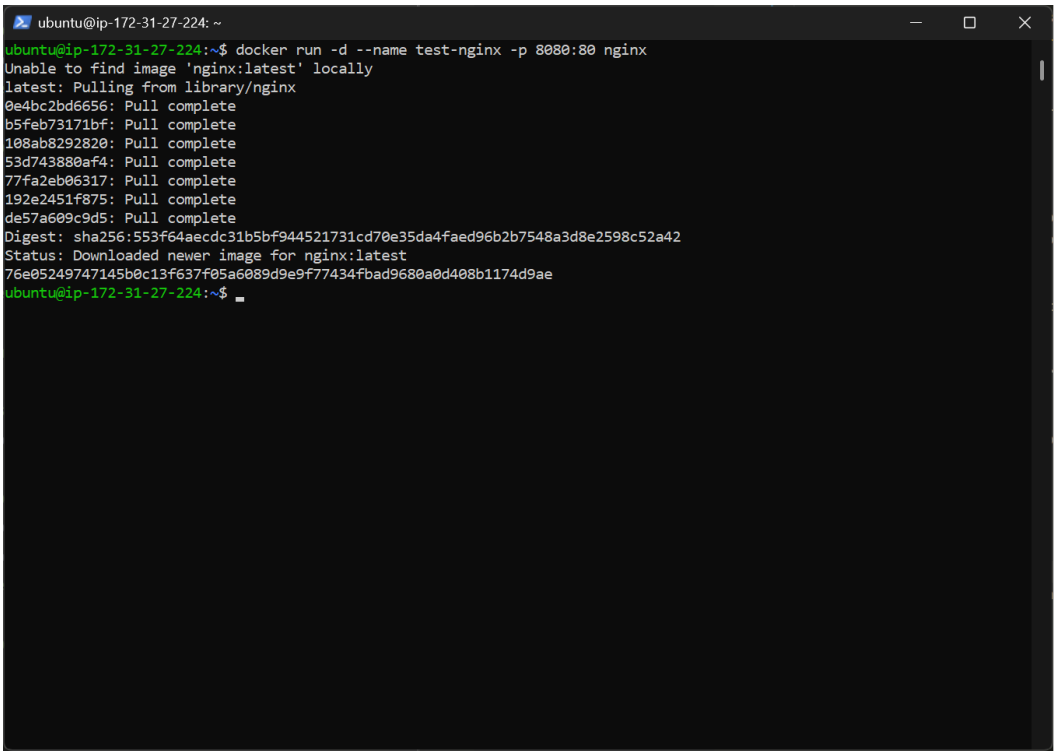

FIGURE 6 – Conteneur Nginx en cours d'exécution

2.5 Étape 5 : Installation de Cockpit

Commandes exécutées :

```
sudo apt install -y cockpit
sudo apt install -y cockpit-docker
sudo systemctl enable --now cockpit.socket
```

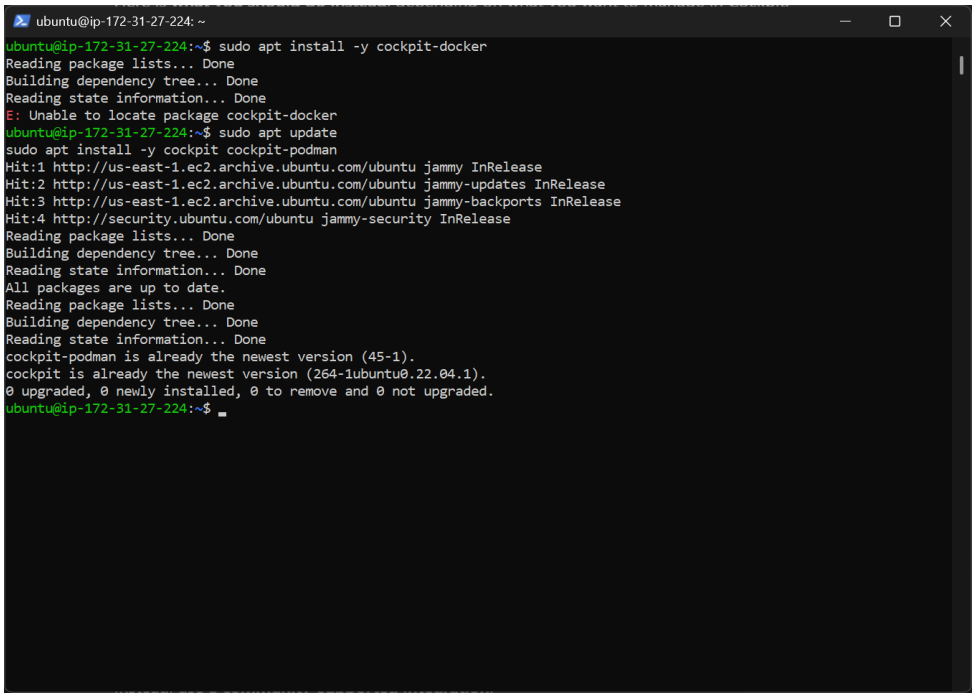

FIGURE 7 – Installation de Cockpit et du plugin Docker

2.6 Étape 6 : Configuration de l'accès Cockpit

2.6.1 Définition du mot de passe

```
sudo passwd ubuntu
```

2.6.2 Vérification du service Cockpit

```
sudo systemctl status cockpit.socket
```

2.7 Étape 7 : Accès à l'interface Cockpit

2.7.1 Page de connexion

Accès via : `https://3.84.104.201:9090`

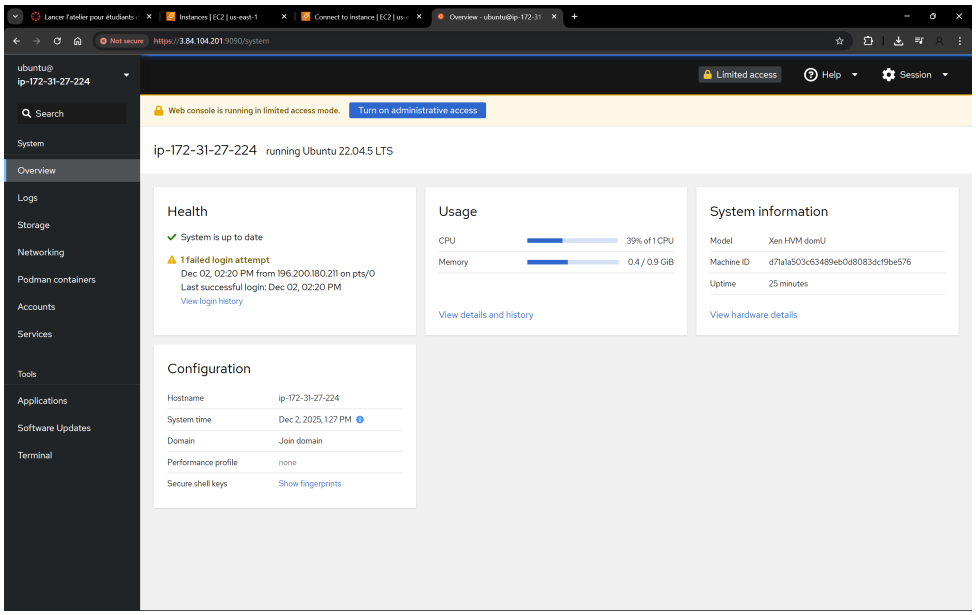

FIGURE 8 – Page de connexion Cockpit