



ÉCOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT  
TECHNIQUE DE MOHAMMED VI  
UNIVERSITÉ HASSAN II DE CASABLANCA

# ÉCOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT TECHNIQUE - MOHAMMED VI

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

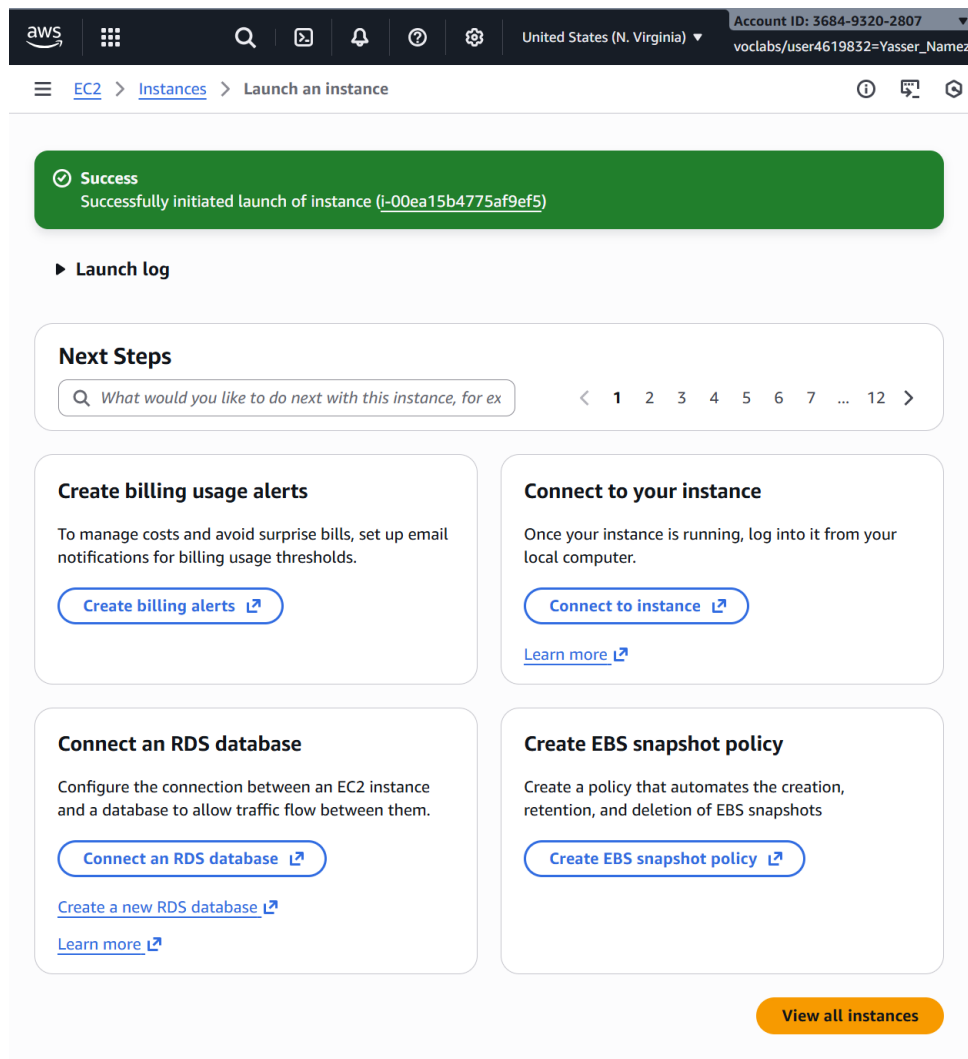


FIGURE 1 – Création de l'instance EC2 sur AWS

### 2.1.2 Configuration du groupe de sécurité

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

Security group name - *required*  
  
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and .-:/()#,@[]+=&{}!\$\*

Description - *required* [Info](#)

**Inbound Security Group Rules**

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0, (SSH)) [Remove](#)

Type [Info](#)

Protocol [Info](#)

Port range [Info](#)

Source type [Info](#)

Source [Info](#)

Description - *optional* [Info](#)

▼ Security group rule 2 (TCP, 9090, 0.0.0.0/0, (Cockpit)) [Remove](#)

Type [Info](#)

Protocol [Info](#)

Port range [Info](#)

Source type [Info](#)

Source [Info](#)

Description - *optional* [Info](#)

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

FIGURE 2 – Configuration du groupe de sécurité

## 2.2 Étape 2 : Connexion à l'instance

Commande utilisée pour se connecter à l'instance :

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

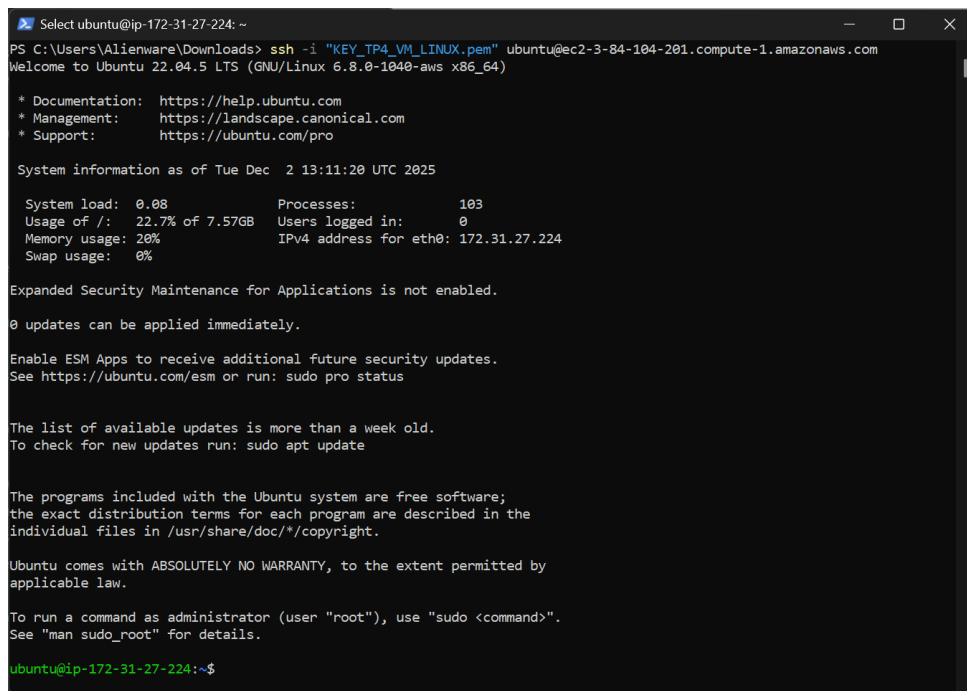


FIGURE 3 – Connexion SSH réussie à l'instance Ubuntu

## 2.3 Étape 3 : Mise à jour du système

Commandes exécutées :

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

```

ubuntu@ip-172-31-27-224: ~
Unpacking binutils-common:amd64 (2.38-4ubuntu2.11) over (2.38-4ubuntu2.8) ...
Preparing to unpack .../19-libssh-4_0.9.6-2ubuntu0.22.04.5_amd64.deb ...
Unpacking libssh-4:amd64 (0.9.6-2ubuntu0.22.04.5) over (0.9.6-2ubuntu0.22.04.4) ...
Selecting previously unselected package linux-modules-6.8.0-1043-aws.
Preparing to unpack .../20-linux-modules-6.8.0-1043-aws_6.8.0-1043.45~22.04.1_amd64.deb ...
Unpacking linux-modules-6.8.0-1043-aws (6.8.0-1043.45~22.04.1) ...
Selecting previously unselected package linux-image-6.8.0-1043-aws.
Preparing to unpack .../21-linux-image-6.8.0-1043-aws_6.8.0-1043.45~22.04.1_amd64.deb ...
Unpacking linux-image-6.8.0-1043-aws (6.8.0-1043.45~22.04.1) ...
Preparing to unpack .../22-linux-aws_6.8.0-1043.45~22.04.1_amd64.deb ...
Unpacking linux-aws (6.8.0-1043.45~22.04.1) over (6.8.0-1040.42~22.04.1) ...
Preparing to unpack .../23-linux-image-aws_6.8.0-1043.45~22.04.1_amd64.deb ...
Unpacking linux-image-aws (6.8.0-1043.45~22.04.1) over (6.8.0-1040.42~22.04.1) ...
Selecting previously unselected package linux-aws-6.8-headers-6.8.0-1043.
Preparing to unpack .../24-linux-aws-6.8-headers-6.8.0-1043_6.8.0-1043.45~22.04.1_all.deb ...
Unpacking linux-aws-6.8-headers-6.8.0-1043 (6.8.0-1043.45~22.04.1) ...
Selecting previously unselected package linux-headers-6.8.0-1043-aws.
Preparing to unpack .../25-linux-headers-6.8.0-1043-aws_6.8.0-1043.45~22.04.1_amd64.deb ...
Unpacking linux-headers-6.8.0-1043-aws (6.8.0-1043.45~22.04.1) ...
Preparing to unpack .../26-linux-headers-aws_6.8.0-1043.45~22.04.1_amd64.deb ...
Unpacking linux-headers-aws (6.8.0-1043.45~22.04.1) over (6.8.0-1040.42~22.04.1) ...
Preparing to unpack .../27-linux-tools-common_5.15.0-161.171_all.deb ...
Unpacking linux-tools-common (5.15.0-161.171) over (5.15.0-157.167) ...
Selecting previously unselected package linux-aws-6.8-tools-6.8.0-1043.
Preparing to unpack .../28-linux-aws-6.8-tools-6.8.0-1043_6.8.0-1043.45~22.04.1_amd64.deb ...
Unpacking linux-aws-6.8-tools-6.8.0-1043 (6.8.0-1043.45~22.04.1) ...
Selecting previously unselected package linux-tools-6.8.0-1043-aws.
Preparing to unpack .../29-linux-tools-6.8.0-1043-aws_6.8.0-1043.45~22.04.1_amd64.deb ...
Unpacking linux-tools-6.8.0-1043-aws (6.8.0-1043.45~22.04.1) ...
Preparing to unpack .../30-snapd_2.72+ubuntu22.04_amd64.deb ...
Unpacking snapd (2.72+ubuntu22.04) over (2.71+ubuntu22.04) ...
Preparing to unpack .../31-sosreport_4.9.2-0ubuntu0~22.04.1_amd64.deb ...
Unpacking sosreport (4.9.2-0ubuntu0~22.04.1) over (4.8.2-0ubuntu0~22.04.2) ...
Preparing to unpack .../32-intel-microcode_3.20250812.0ubuntu0.22.04.1_amd64.deb ...
Unpacking intel-microcode (3.20250812.0ubuntu0.22.04.1) over (3.20250512.0ubuntu0.22.04.1) ...
Setting up snapd (2.72+ubuntu22.04) ...
Progress: [ 50%] [#####]
    
```

FIGURE 4 – Mise à jour du système Ubuntu

## 2.4 Étape 4 : Installation de Docker

### 2.4.1 Installation

Commandes exécutées :

```

1 sudo apt install -y docker.io
2 sudo systemctl enable docker
3 sudo systemctl start docker
4 sudo usermod -aG docker ubuntu
    
```

### 2.4.2 Test de Docker

Après reconnexion, test avec hello-world :

```

1 docker run hello-world
    
```

```
ubuntu@ip-172-31-27-224: ~  
1 additional security update can be applied with ESM Apps.  
Learn more about enabling ESM Apps service at https://ubuntu.com/esm  
  
New release '24.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** System restart required ***  
Last login: Tue Dec 2 13:11:22 2025 from 196.200.180.211  
ubuntu@ip-172-31-27-224:~$ docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
17eec7bbc9d7: Pull complete  
Digest: sha256:f7931603f70e13dbd844253370742c4fc4202d290c80442b2e68706d8f33ce26  
Status: Downloaded newer image for hello-world:latest  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/  
ubuntu@ip-172-31-27-224:~$
```

FIGURE 5 – Test de Docker avec hello-world

### 2.4.3 Déploiement d'un conteneur Nginx

Pour démonstration, j'ai déployé un conteneur Nginx :

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



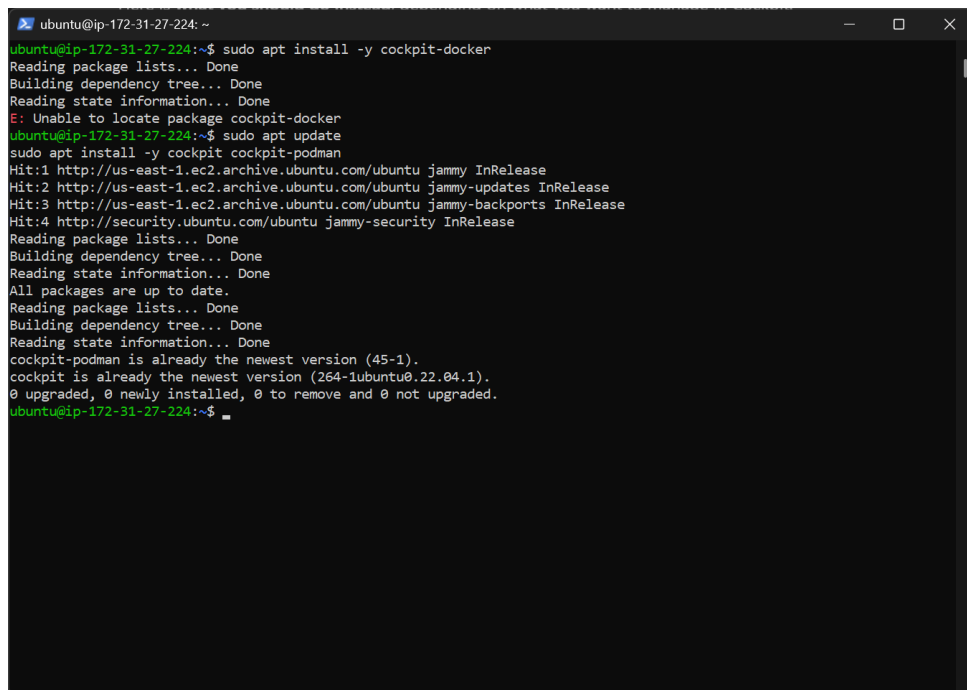
```
ubuntu@ip-172-31-27-224: ~  
ubuntu@ip-172-31-27-224:~$ docker run -d --name test-nginx -p 8080:80 nginx  
Unable to find image 'nginx:latest' locally  
latest: Pulling from library/nginx  
0e4bc2bd6656: Pull complete  
b5feb73171bf: Pull complete  
108ab8292820: Pull complete  
53d743880af4: Pull complete  
77fa2eb06317: Pull complete  
192e2451f875: Pull complete  
de57a609c9d5: Pull complete  
Digest: sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c52a42  
Status: Downloaded newer image for nginx:latest  
76e05249747145b0c13f637f05a6089d9e9f77434fbad9680a0d408b1174d9ae  
ubuntu@ip-172-31-27-224:~$
```

FIGURE 6 – Conteneur Nginx en cours d'exécution

## 2.5 Étape 5 : Installation de Cockpit

Commandes exécutées :

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



```
ubuntu@ip-172-31-27-224: ~  
ubuntu@ip-172-31-27-224:~$ sudo apt install -y cockpit-docker  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
E: Unable to locate package cockpit-docker  
ubuntu@ip-172-31-27-224:~$ sudo apt update  
sudo apt install -y cockpit cockpit-podman  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
All packages are up to date.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
cockpit-podman is already the newest version (45-1).  
cockpit is already the newest version (264-1ubuntu0.22.04.1).  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
ubuntu@ip-172-31-27-224:~$
```

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

```
1 sudo passwd ubuntu
```

### 2.6.2 Vérification du service Cockpit

```
1 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>

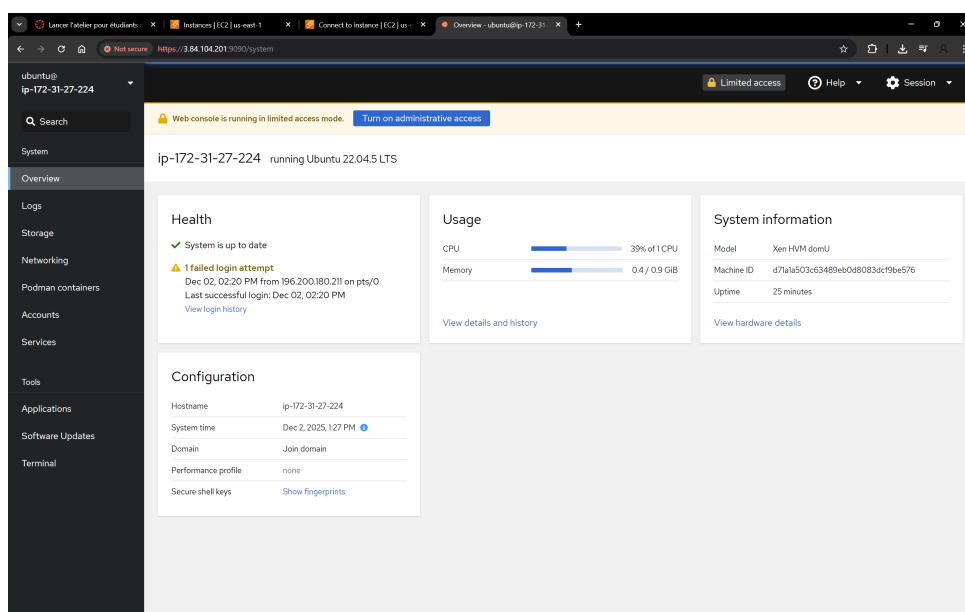


FIGURE 8 – Page de connexion Cockpit