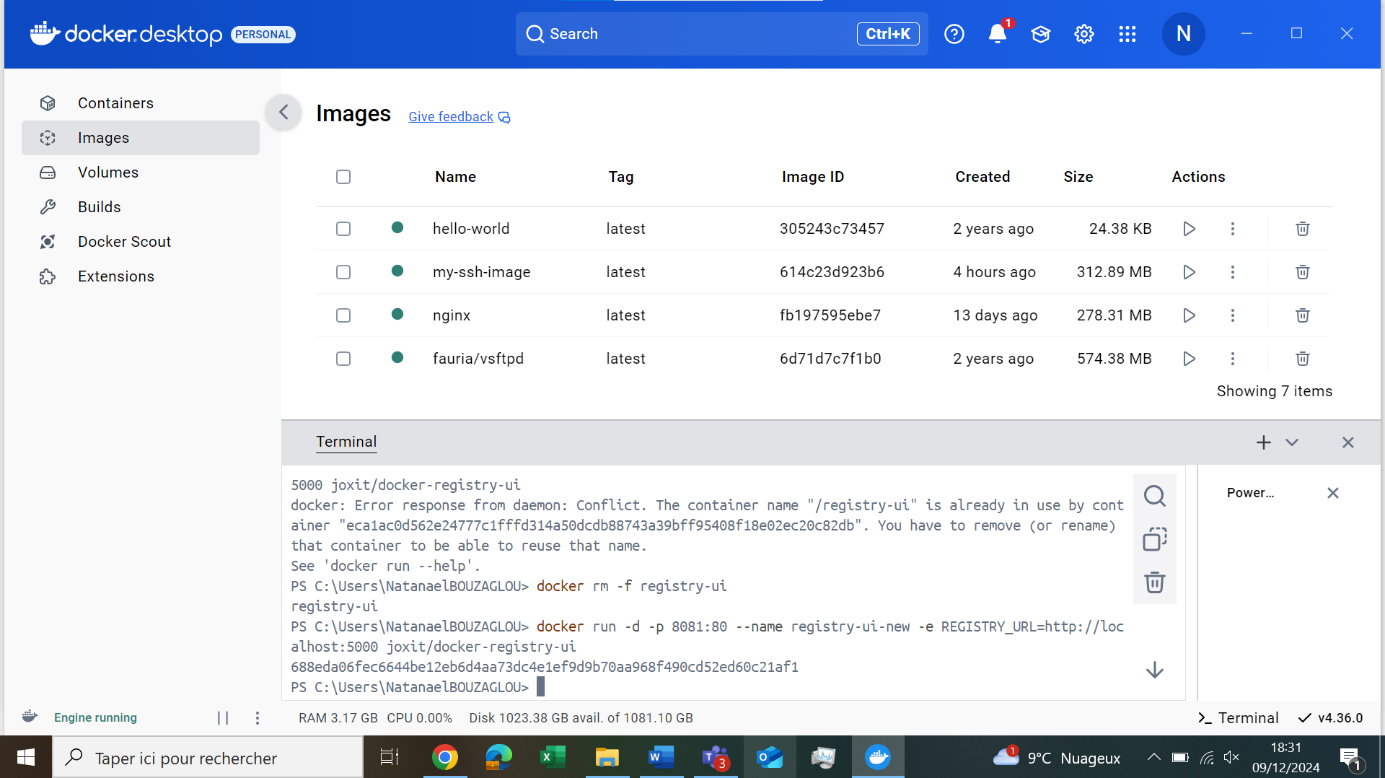
**TP Docker Natanael Bouzaglou**

**09/12/2024**

**Job01 : Installation de Docker directement sur mon poste**

**Capture d'écran :**

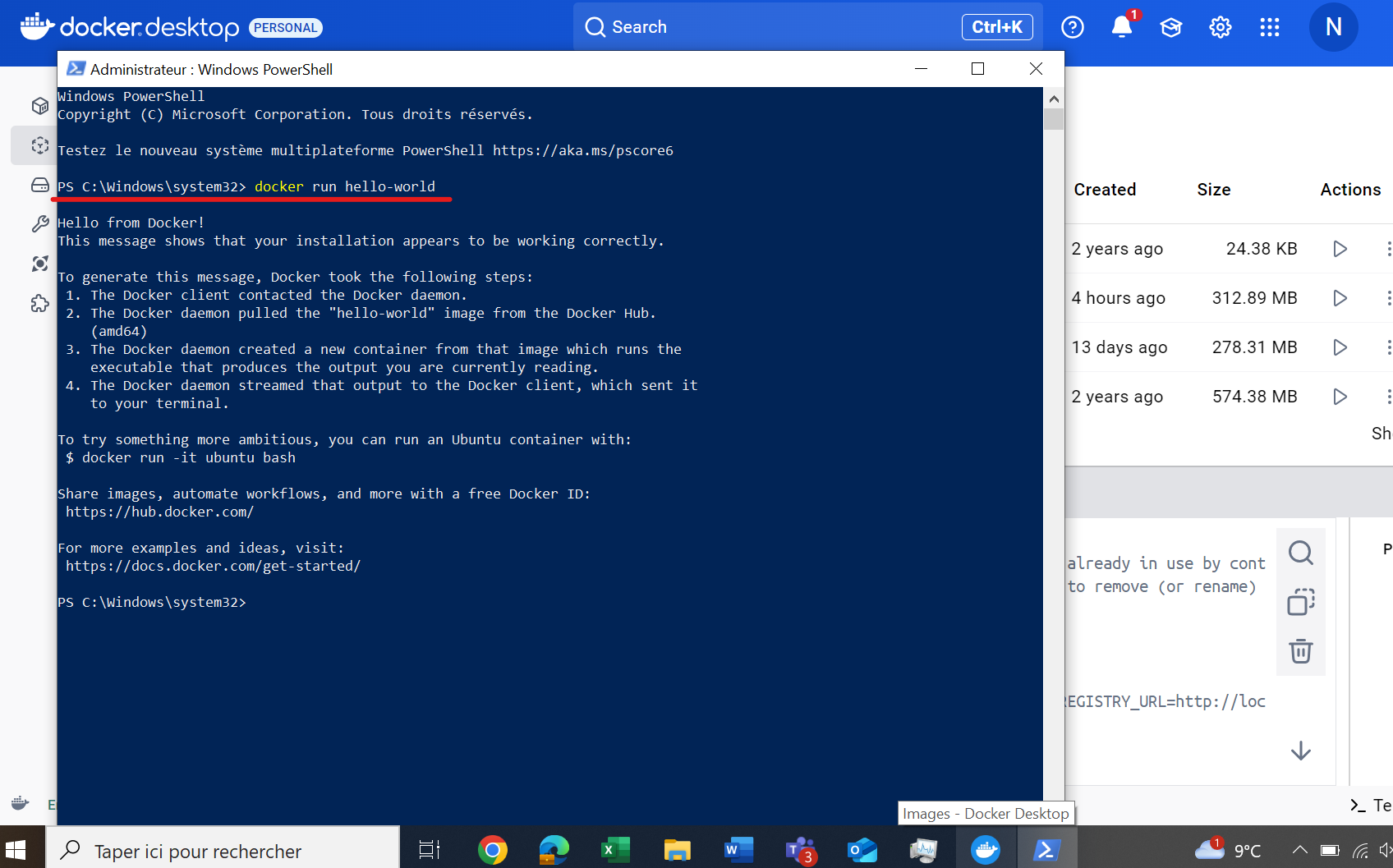
****

**Job02 : Tester l’installation de Docker avec le conteneur "helloworld"**

* Tester le bon fonctionnement de Docker avec le conteneur helloworld :

docker run hello-world

**Capture d'écran :**



**Job03 : Utilisation de Dockerfile pour recréer le conteneur "helloworld" depuis une image Debian minimum**

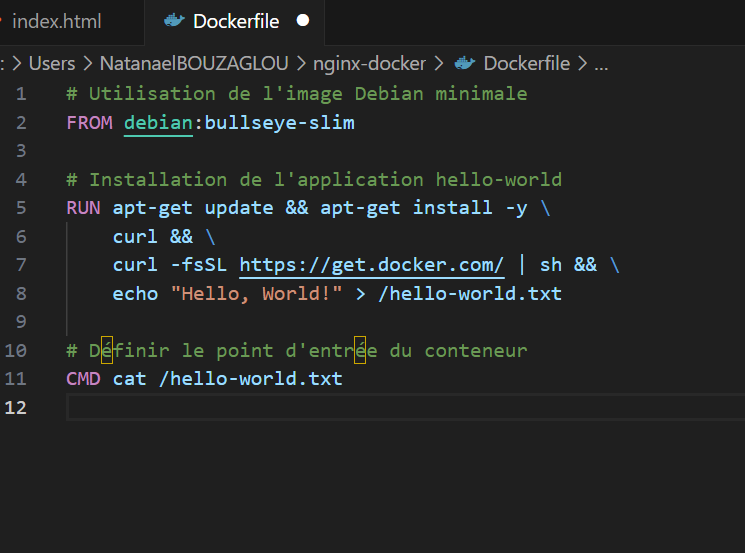
* Créer un fichier Dockerfile pour reproduire l'image helloworld :

Dockerfile

FROM debian:latest

CMD echo "Hello, World!"

**Capture d'écran:**

* 

**Job04 : Création d'une image SSH sans image SSH existante**

* Créer une image Docker pour un serveur SSH avec un compte root (root / root123) et une redirection de port autre que le port 22.
* Exemple de Dockerfile :

Dockerfile

FROM debian:latest

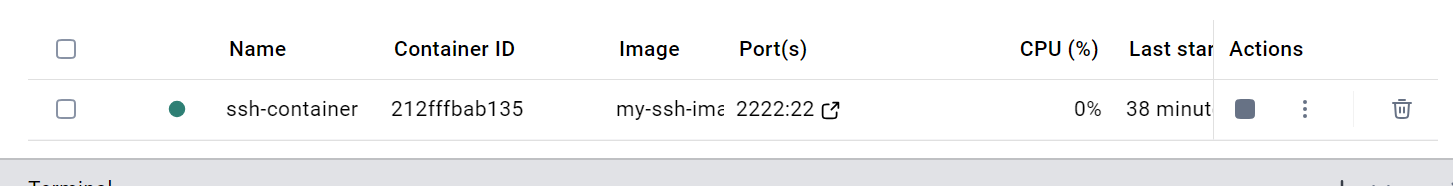
RUN apt-get update && apt-get install -y openssh-server

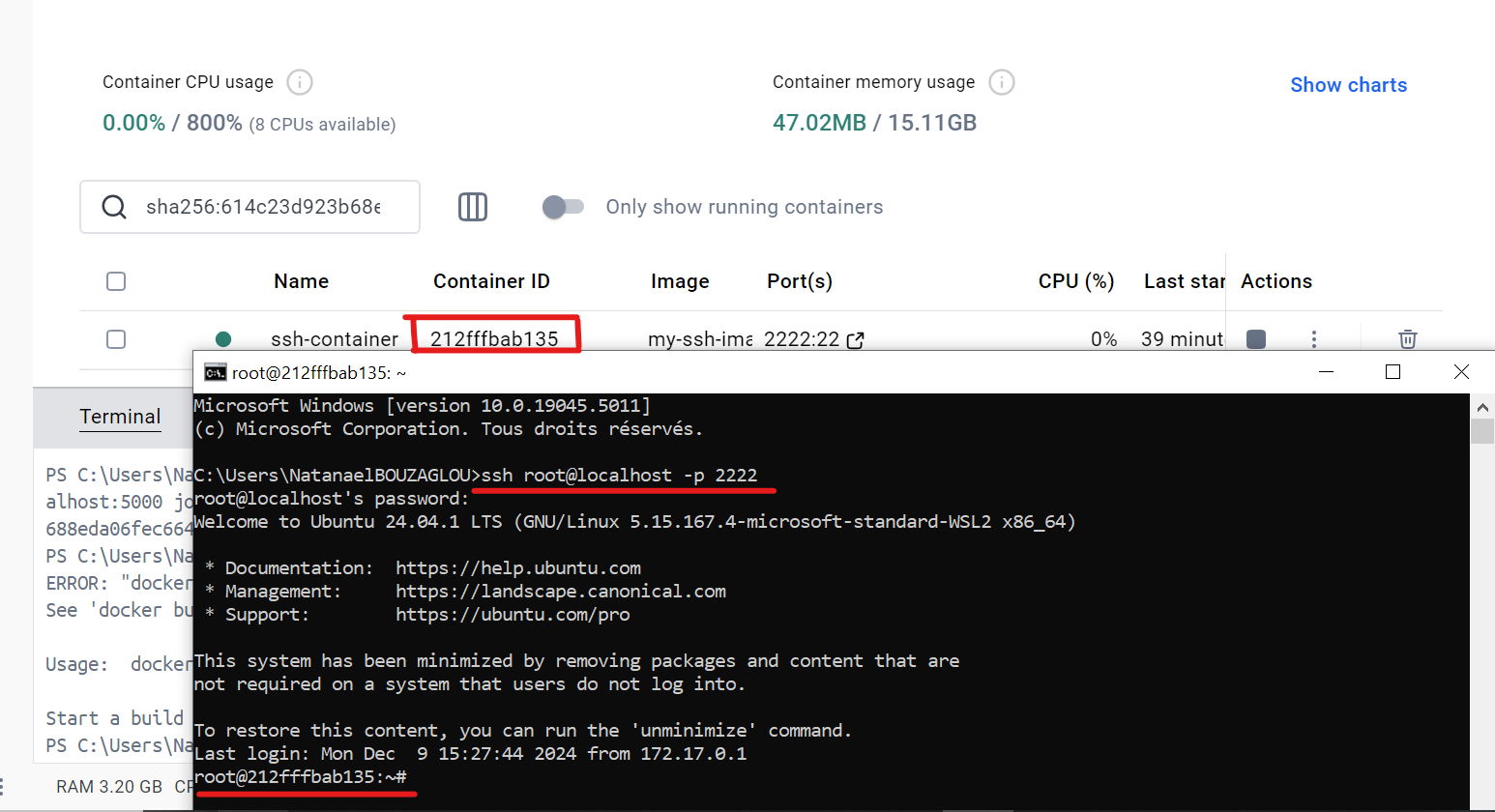
RUN echo 'root:root123' | chpasswd

CMD /usr/sbin/sshd -D

**Capture d'écran :**

* Capture montrant la création du conteneur SSH et la connexion via SSH.





**Job05 : Création d’alias pour les commandes Docker**

* Ajouter des alias dans le fichier ~/.bashrc pour simplifier les commandes Docker :

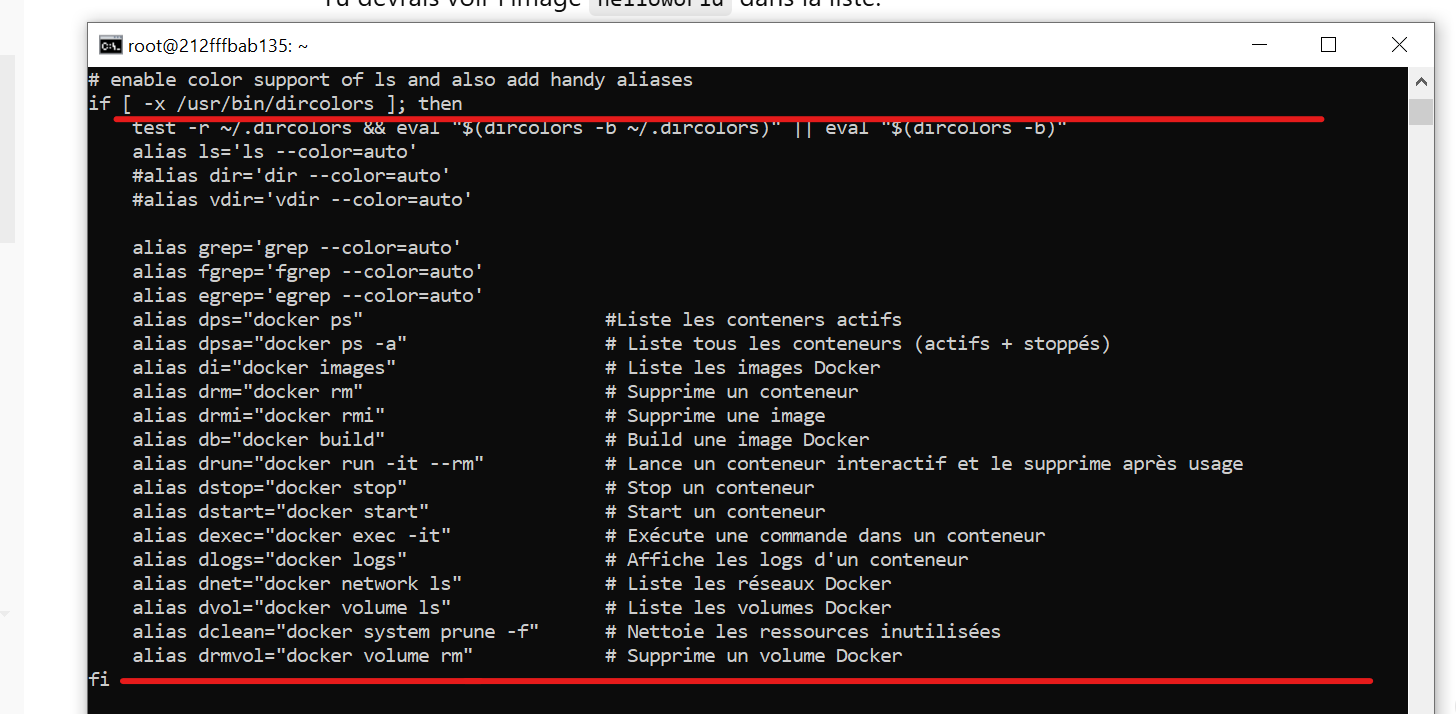
alias dps='docker ps'

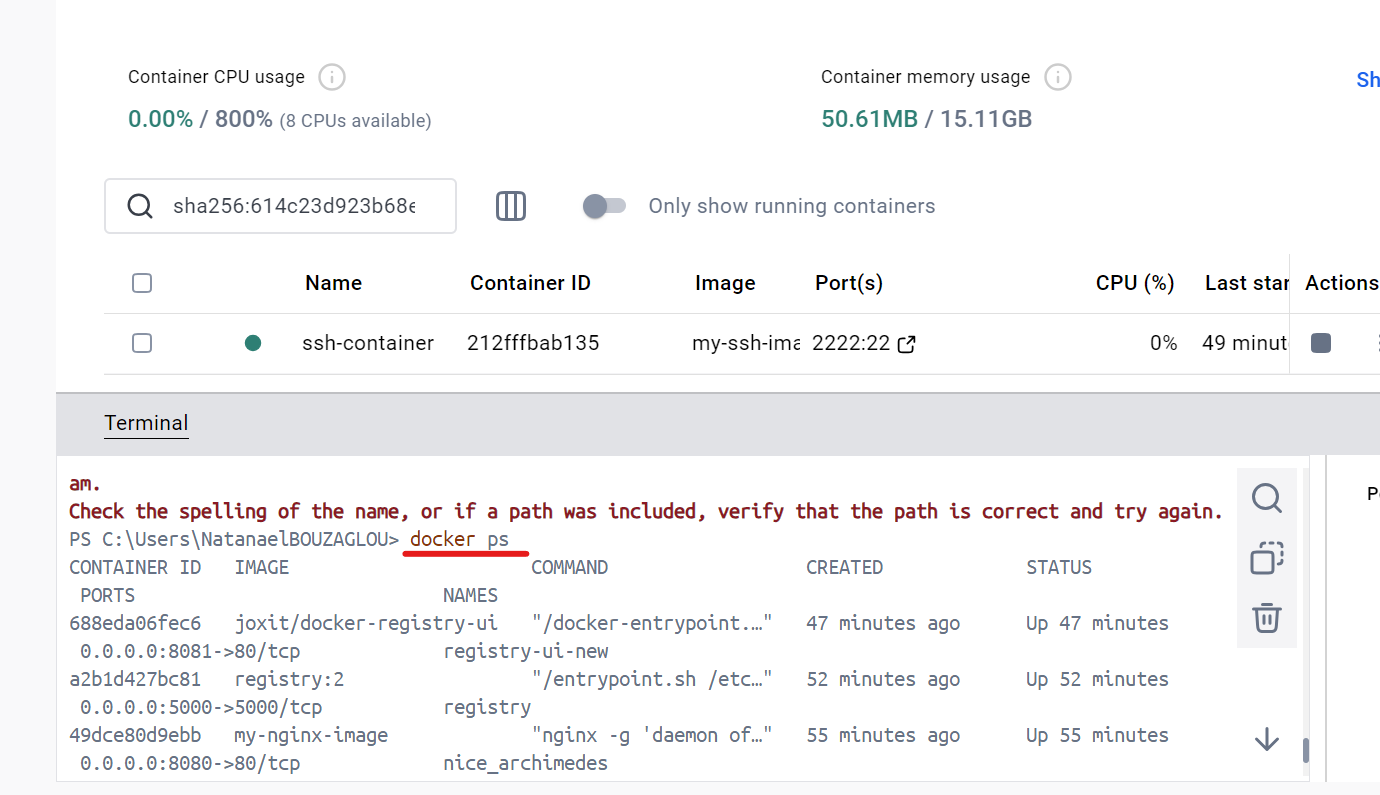
alias dstop='docker stop'

alias dstart='docker start'

**Capture d'écran à insérer :**

* Capture du terminal montrant la modification du fichier ~/.bashrc et l'utilisation des alias.





**Job06 : Utilisation de Volumes entre 2 conteneurs**

* Créer un volume Docker partagé entre deux conteneurs pour partager des fichiers :

bash

docker volume create shared-volume

**Capture d'écran à insérer :**

* Capture des commandes pour créer et inspecter les volumes, et l’accès au volume entre les conteneurs.

**Job07 : Utilisation de Docker Compose avec Nginx et FTP**

* Créer un fichier docker-compose.yml pour faire communiquer un conteneur Nginx avec un conteneur FTP :

yaml

Copier le code

version: '3'

services:

nginx:

image: nginx

ports:

- "80:80"

ftp:

image: fauria/vsftpd

environment:

- FTP\_USER=user

- FTP\_PASS=pass

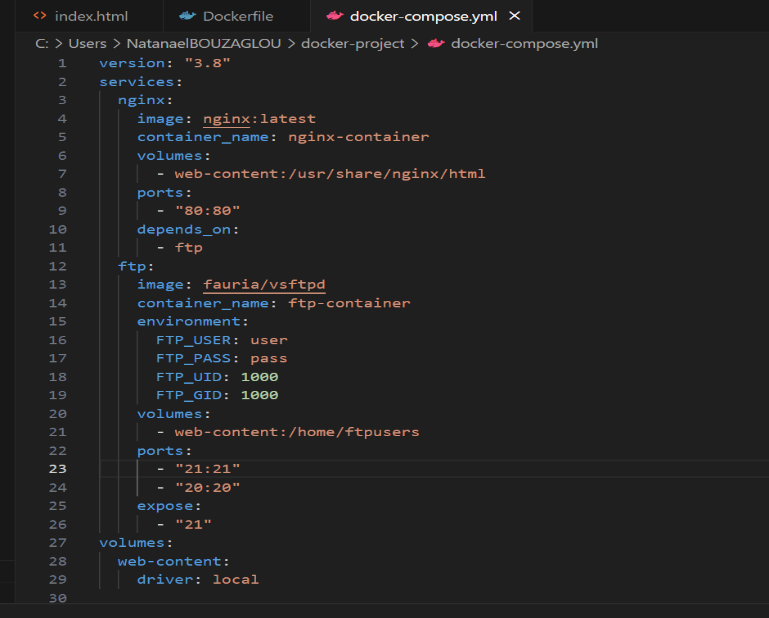
volumes:

- ./data:/data

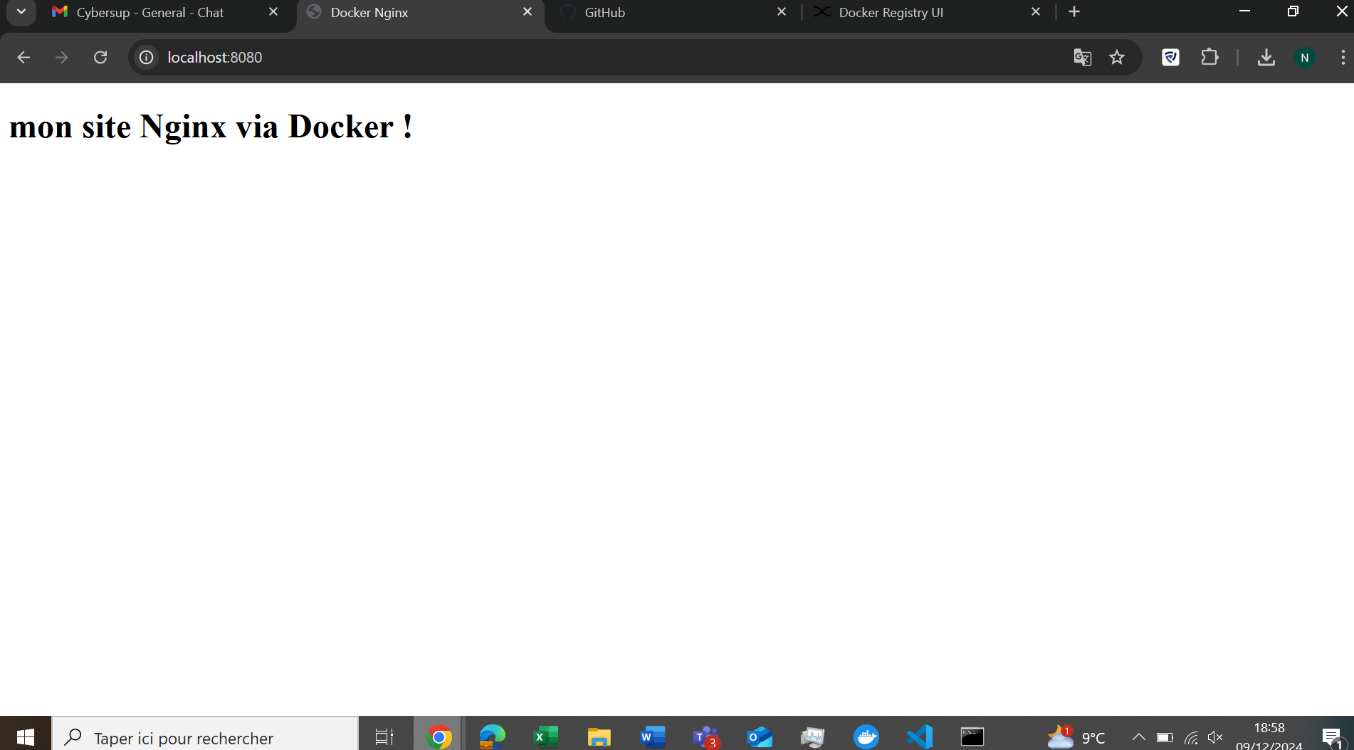
* Créer un fichier index.html avec votre nom/prénom et l'envoyer via FTP.

**Capture d'écran à insérer :**

* Capture du fichier docker-compose.yml.



Capture du résultat affiché sur le navigateur via Nginx.



**Job08 : Création d’un conteneur Nginx personnalisé**

* Créer un Dockerfile pour créer un conteneur Nginx sans utiliser une image existante :

Dockerfile

FROM debian

RUN apt-get update && apt-get install -y nginx

COPY ./index.html /usr/share/nginx/html

CMD ["nginx", "-g", "daemon off;"]

**Job09 : Création et utilisation d’un "Registry" local avec UI**

* Créer un registry local Docker et ajouter une interface graphique pour le gérer.

bash

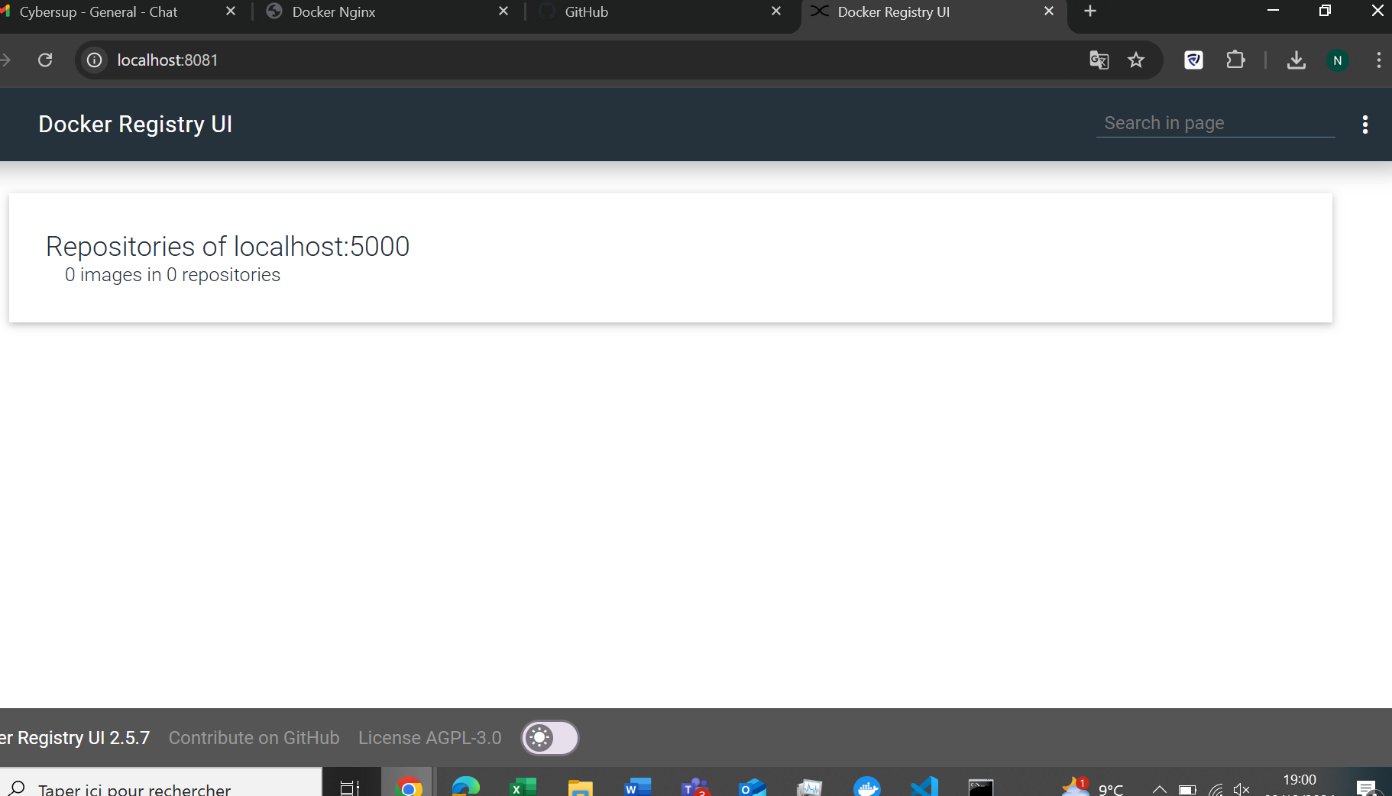
Copier le code

docker run -d -p 5000:5000 --name registry registry:2

docker run -d -p 8081:80 --name registry-ui -e REGISTRY\_URL=http://localhost:5000 joxit/docker-registry-ui

**Capture d'écran à insérer :**

* Capture de l'interface graphique du Docker Registry (via navigateur).



**Conclusion :**

Au cours de ce projet, j'ai réussi à :

* Déployé des conteneurs simples comme "hello-world" pour valider l'installation et m'habituer aux commandes Docker.
* Créé des Dockerfiles pour construire mes propres images, comme celle du conteneur SSH et du serveur Nginx.
* Manipulé des volumes pour comprendre leur utilisation dans le partage de données entre conteneurs.
* Utilisé Docker Compose pour configurer et orchestrer plusieurs conteneurs, comme un serveur FTP et Nginx interconnectés.
* Déployé un registre local avec une interface graphique (Docker Registry UI) pour gérer mes images en local.