

Voici un résumé des étapes réalisées suite à la consigne pour la prise en main de Docker :

**Installation de Docker :** J'ai commencé par installer Docker sur une base Debian configurée en accès réseau par pont. Pour cela, j'ai suivi les instructions de la documentation officielle de Docker pour Debian, en prenant soin de lire attentivement toutes les informations fournies.

**Vérification de l'installation :** Après l'installation, j'ai exécuté la commande `docker run hello-world`. Cette commande a permis de vérifier que Docker était correctement installé. Un message de confirmation "Hello from Docker!" s'est affiché, indiquant que l'installation était réussie et que Docker fonctionnait correctement.

**Quand conteneur arrêté : Exited (0) 2 minutes ago**

**Quand conteneur relancé : Up 5 seconds**

Premier conteneur : my-nginx-container

```
root@6c261a6ba325:/# uname -a
Linux 6c261a6ba325 5.10.0-25-amd64 #1 SMP Debian 5.10.191-1 (2023-08-16) x86_64 GNU/Linux
root@6c261a6ba325:/#
```

2eme conteneur : conteneurCyber

mdp : nelson

Récapitulatif sur les conteneurs, les étapes importantes et fiche de procédure :

Création et gestion des images :

- **Installation de docker :**

`docker --version`

- **Créer un dockerfile, par exemple en utilisant nginx :**

```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/index.html
```

- **Construire une image Docker à partir d'un dockerfile :**

`docker build -t my-nginx .`

- **Commande pour lister les images présentes :**

`docker images`

### Exécution et gestion des conteneurs :

- **Exécuter un conteneur :**

`docker run --name my-nginx-container -d -p 8080:80 my-nginx`

*--name my-nginx-container* : Donne un nom au conteneur (ici, my-nginx-container).

*-d* : Exécute le conteneur en arrière-plan (mode détaché).

*-p 8080:80* : Mappe le port 8080 de l'hôte au port 80 du conteneur (NGINX écoute par défaut sur le port 80).

*my-nginx* : Indique l'image à utiliser pour démarrer le conteneur.

- **Lister les conteneurs en cours d'exécution (Commandes importantes) :**

`docker ps`

lister tous les conteneurs même arrêtés : `docker ps -a`

consulter les logs d'un conteneur : `docker logs my-nginx-container`

- **Accéder à un conteneur en cours d'exécution :**

`docker exec -it my-nginx-container /bin/bash`

### Contrôle des conteneurs :

- **Pour démarrer un conteneur :**

`docker start my-nginx-container`

- **Pour arrêter un conteneur en cours d'exécution :**

`docker stop my-nginx-container`

- **Pour supprimer un conteneur (arrêté) :**

`docker rm my-nginx-container`

- **Supprimer une image Docker :**

`docker rmi my-nginx`

**version: '3.8'**

**services:**

**db:**

**image: postgres:13**

**environment:**

**POSTGRES\_DB: wordpress**

**POSTGRES\_USER: wordpress**

**POSTGRES\_PASSWORD: wordpress**

**volumes:**

**- db\_data:/var/lib/postgresql/data**

**networks:**

**- wpnet**

**wordpress:**

**image: wordpress:latest**

**ports:**

**- "8080:80"**

**environment:**

**WORDPRESS\_DB\_HOST: db:5432**

**WORDPRESS\_DB\_USER: wordpress**

**WORDPRESS\_DB\_PASSWORD: wordpress**

**WORDPRESS\_DB\_NAME: wordpress**

**volumes:**

**- wp\_data:/var/www/html**

**networks:**

**- wpnet**

**volumes:**

**db\_data:**

**wp\_data:**

**networks:**

**wpnet:**

**services:**

**image: nginx:latest**

**ports:**

**- "8080:80"**