

## Commande de base de Docker

```
$ docker ps # Visualiser les conteneurs actifs
$ docker ps -a # Visualiser tous les conteneurs
$ docker rm [container] # Supprimer un conteneur inactif
$ docker rm -f [container] # Forcer la suppression d'un conteneur actif
$ docker images # Lister les images existantes
$ docker rmi [image] # Supprimer une image docker
$ docker exec -t -i [container] /bin/bash # Exécuter des commandes dans un conteneur actif
$ docker inspect [container] # Inspecter la configuration d'un conteneur
$ docker build -f [DockerFile] -t [nom_image] . # Construire une image à partir d'un Dockerfile
$ docker history [image] # Visualiser l'ensemble des couches d'une image
$ docker logs --tail 5 [container] # Visualiser les logs d'un conteneur (les 5 dernières lignes)

-----Interactions avec le registry-----
$ docker login # Se connecter au registry
$ docker search [name] # Rechercher une image
$ docker pull [image] # Récupérer une image
$ docker push [image] # Pousser une image du cache local au registry
$ docker tag [UUID] [image]:[tag] # Tagger une image (pour la push vers un registry)
```

`sudo docker build -f dokcerfile -t nom_image .`

# Job 1 et 2 :

Dans un premier temps, on install Debian 12 en mode CLI.

une fois que nous avons installer notre debian, nous allons créer un script :

```
```bash
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

Une fois que nous avons créer notre script, nous pouvons l'exécuter.

Une fois que notre script c'est bien exécuter, nous pouvons encore exécuter une commande pour installer les dernier packet nécessaire au bon fonctionnement de docker :

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Maintenant pour vérifier que Docker est bien installer, on peut lancer cette commande :

```
sudo docker run hello-world
```

Si notre installation à bien fonctionner on devrait voir ce message apparaitre :

```
raph@DebianXDocker:~/install$ sudo docker run hello-world

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/
```