

# TP 1 : Installation de Docker et démarrage d'un premier conteneur

## TP 1 : Installation de Docker et démarrage d'un premier conteneur

### 1. Installer Docker

Pour Windows

Pour MacOS

Pour Linux

    Debian / Ubuntu

    Fedora

    RHEL / Rocky

### 2. Lancement d'un conteneur

    2.1. Récupération d'une image depuis le Docker hub

    2.2. Lancement d'un conteneur

    2.3. Lancement d'un conteneur en mode détaché

    2.4. Lancement d'un conteneur shell

    2.5. Redirections de port

## 1. Installer Docker

### Pour Windows

Premièrement il vaut mieux vous assurer que votre système est compatible avec WSL2 et que ce dernier est bien installé. Ensuite, téléchargez Docker Desktop et installez-le, en suivant les instructions. Si on vous le demande, faites bien attention à utiliser WSL au lieu d'Hyper-V.

Redémarrez votre ordinateur. Vous devriez ensuite pouvoir lancer cette commande sans erreur :

```
1 > docker ps
2 CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    PORTS
```

Félicitations, Docker est maintenant installé et fonctionnel !

### Pour MacOS

Assurez-vous que Homebrew est installé sur votre système au préalable.

Ensuite, installez colima, comme ceci :

```
1 brew install colima
```

Ainsi que l'interface en ligne de commande docker :

```
1 brew install docker
```

Pour lancer le runtime Docker, utilisez la commande :

```
1 colima start
```

Vous devriez ensuite pouvoir lancer cette commande sans erreur :

```
1 ~ $ docker ps
2 CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS
```

Félicitations, Docker est maintenant installé et fonctionnel !

## Pour Linux

### Debian / Ubuntu

Installez les dépendances :

```
1 sudo apt update
2 sudo apt install ca-certificates curl gnupg lsb-release
```

Ajoutez la clé GPG Docker :

```
1 sudo mkdir -p /etc/apt/keyrings
2 curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
  /etc/apt/keyrings/docker.gpg
```

Ajoutez le répo APT :

```
1 echo \
2   "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian \
3   $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >
  /dev/null
```

Installez le moteur Docker :

```
1 sudo apt update
2 sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Enfin, ajoutez votre utilisateur au groupe `docker` :

```
1 sudo useradd $(whoami) docker
```

Fermez et rouvrez votre session, et vous devriez pouvoir exécuter la commande suivante :

```
1 ~ $ docker ps
2 CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS
```

Félicitations, Docker est maintenant installé et fonctionnel !

---

## Fedora

Ajoutez le repo Docker :

```
1 sudo dnf -y install dnf-plugins-core  
2 sudo dnf config-manager --add-repo https://download.docker.com/linux/fedora/docker-ce.repo
```

Installez le moteur Docker :

```
1 sudo dnf install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Enfin, ajoutez votre utilisateur au groupe docker :

```
1 sudo usermod -aG docker $(whoami)
```

Fermez et rouvrez votre session, et vous devriez pouvoir exécuter la commande suivante :

```
1 ~ $ docker ps  
2 CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
```

Félicitations, Docker est maintenant installé et fonctionnel !

---

## RHEL / Rocky

Ajoutez le repo Docker :

```
1 sudo yum install -y yum-utils  
2 sudo yum-config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
```

Installez le moteur Docker :

```
1 sudo yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Enfin, ajoutez votre utilisateur au groupe docker :

```
1 sudo usermod -aG docker $(whoami)
```

Fermez et rouvrez votre session, et vous devriez pouvoir exécuter la commande suivante :

```
1 ~ $ docker ps  
2 CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
```

Félicitations, Docker est maintenant installé et fonctionnel !

## 2. Lancement d'un conteneur

### 2.1. Récupération d'une image depuis le Docker hub

Lancez la commande :

```
1 docker pull nginx
```

Que remarquez-vous ?

### 2.2. Lancement d'un conteneur

Exécutez la commande :

```
1 docker run nginx
```

Que se passe-t-il au niveau de votre shell ?

Vous pouvez ouvrir une autre fenêtre de terminal pour vérifier que le conteneur est bien visible via la commande `docker container ls` ou `docker ps`.

### 2.3. Lancement d'un conteneur en mode détaché

Pour lancer le conteneur en mode détaché, exécutez la commande :

```
1 docker run -d nginx
```

Quelle différence avec avant ?

Contrôlez l'état du conteneur avec `docker ps`, puis arrêtez-le avec `docker container stop <container id>` et supprimez-le avec `docker container rm <container id>`.

### 2.4. Lancement d'un conteneur shell

Nous introduirons ici deux nouveaux arguments :

- `-t` : Permet d'allouer un TTY virtuel à un conteneur
- `-i` : Permet de garder l'entrée standard ouverte

Exécutez la commande :

```
1 docker run -d bash
```

Que se passe-t-il ? Contrôlez son état à l'aide de la commande `docker ps --all`.

Réessayez maintenant l'opération en mode attaché :

```
1 docker run -it bash
```

Concluez sur l'utilité du mode attaché.

