



## Installation

### Étape 1: mettre à jour la base de données locale

Avant l'installation, commencez par mettre à jour la liste existante des packages

```
sudo apt update
```

### Étape 2: Installez les dépendances prenant en charge https

Installez des packages qui permettront à apt de transférer des fichiers via https

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

C'est ce que fait chaque commande;

- `apt-transport-https`: Autoriser apt à transférer des fichiers et des données via https
- `ca-certificates`: faites en sorte que l'ordinateur ou le navigateur vérifie les certificats de sécurité
- `curl`: Un outil de transfert de fichiers
- `software-properties-common`: ajoute des scripts de gestion de logiciels

### Étape 3: Ajouter la clé GPG de Docker

Le prochain serait d'ajouter la clé GPG - une fonction de sécurité qui garantit l'authenticité des fichiers d'installation.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

### Étape 4: Installez le référentiel Docker

Dans cette étape, utilisez la commande ci-dessous pour ajouter le référentiel Docker au apt sources.

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
```

### Étape 5: Mettre à jour les référentiels

Mettez à jour la base de données avec les packages Docker du dépôt que vous venez d'ajouter

```
sudo apt update
```

Pour vous assurer que vous exécutez l'installation à partir du référentiel Docker officiel, exécutez la commande

```
apt-cache policy docker-ce
```

### Étape 6: Installez la dernière version de Docker

Une fois confirmé, utilisez la commande suivante pour installer Docker.



# Docker

```
sudo apt install docker-ce
```

Cela installera Docker, démarrera le démon et lui permettra de démarrer automatiquement au démarrage. Pour confirmer que Docker est actif et fonctionne, exécutez :

```
sudo systemctl status docker
```

Vérifier la version du docker :

```
docker --version
```

```
sudo docker run -d -p 80:80 docker/getting-started
```

Pour afficher tous les conteneurs — actifs et inactifs, exécutez `docker ps` avec le commutateur `-a`:

```
sudo docker ps -a
```

```
http://localhost:80 to verify
```



## Tp

### Vérifier l'installation de Docker :

Pour vérifier si vous pouvez accéder aux images et les télécharger à partir de Docker Hub:

```
docker run hello-world
```

### Définir un conteneur avec Dockerfile :

utilise l'image officielle du framework node.js comme image parente pour cette application :

```
FROM node:current-slim
```

cela change directement à ce qu'il suit, nous entrons donc ici dans le répertoire :

```
WORKDIR /usr/src/app
```

nous copions le package de ce fichier JSON dans un point, ce qui signifie le répertoire actuel du conteneur :

```
COPY package.json .
```

L'instruction run exécutera toutes les commandes à l'intérieur de l'image, nous exécutons donc ici npm install pour installer notre projet npm :

```
RUN npm install
```

cela signifie que lorsque nous exécuterons notre application, elle sera accessible dans le port 8080 :

```
EXPOSE 8080
```

CMD est une commande pour exécuter le conteneur :

```
CMD [ "npm", "start" ]
```



pour copier tous les fichiers du répertoire courant de votre machine dans le répertoire courant de l'image plus sombre :

```
COPY . .
```

## Construire l'image :

Build an image from a Dockerfile :

```
docker build --tag first-app .
```

## Exécuter l'image en tant que conteneur :

nous pouvons créer un premier conteneur par :

```
docker run --name first-app --publish 8080:8080 first-app
```

===== push:=====

Comment configurer un compte docker hub et le connecter au bureau docker ?

## Créer docker hub repository:

Lister les images :

```
docker image ls
```

tag notre image et push cette image vers docker hub repository :

```
docker tag first-app (your docker id)/first-app:1.0
```

```
docker push (your docker id)/first-app:1.0
```

pour importer des images à partir de docker hub :

```
docker pull name : tag
```

par ex : docker pull ak2lamhour/first-app:1.0