

# Rapport de TP : Introduction à Docker avec Application Flask

## Informations générales

**Nom des étudiants :** Bassma Alaoui Sossey, Safae Mamouni Alaoui, Sara Zahidi

**Filière :** Cybersécurité

**professeur :** AMAMOU Ahmed

---

## Objectifs du TP

Nous avons fait ce TP en classe. Ce TP avait pour but de nous familiariser avec Docker, un outil de virtualisation très utilisé dans le déploiement d'applications. L'objectif était de :

- Installer Docker sur une machine Linux
  - Lancer et gérer des conteneurs Docker
  - Créer une image Docker personnalisée
  - Déployer une application Python simple avec Flask
- 

## 1. Installation de Docker

Nous avons installé Docker sur un système Ubuntu via la commande suivante :

```
sudo apt update
sudo apt install docker.io -y
sudo systemctl enable docker
sudo systemctl start docker
```

Nous avons vérifié l'installation avec :

```
docker version
docker info
sudo systemctl status docker
```

Pour exécuter Docker sans sudo, nous avons ajouté l'utilisateur au groupe Docker :

```
sudo usermod -aG docker $USER
```

---

## 2. Téléchargement et gestion d'une image Docker

Nous avons utilisé l'image Nginx comme exemple de base :

```
docker pull nginx
docker run -d -p 8080:80 nginx
docker run -d -p 8081:80 --name mon-nginx nginx
```

Pour gérer les conteneurs :

```
docker ps
docker ps -a
docker stop <container_id>
docker rm <container_id>
docker container prune
```

Pour accéder à un conteneur en cours d'exécution :

```
docker exec -it <container_id> /bin/bash
```

---

## 3. Création d'une image Docker personnalisée

Nous avons créé un fichier Dockerfile contenant :

```
FROM python:3.11
COPY ./app /app
WORKDIR /app
RUN pip install -r requirements.txt
EXPOSE 5000
CMD ["python", "app.py"]
```

Pour construire et vérifier l'image :

```
docker build -t mon-app:1.0 .
docker images
```

---

## 4. Application Python avec Flask

Nous avons développé une application Flask très simple :

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def home():
    return "Hello, Docker!"

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

Fichier requirements.txt :

```
flask
```

Nous avons construit et lancé l'application avec :

```
docker build -t mon-flask-app .
docker run -d -p 5000:5000 mon-flask-app
```

Puis nous avons accédé à l'application via : <http://localhost:5000>

---

## Difficultés rencontrées

- Problème de permission lors de l'exécution de Docker sans sudo (résolu via usermod)
- Erreur dans le chemin du Dockerfile au début
- Compréhension du port mapping entre le conteneur et l'hôte
- Plusieurs ordinateurs en salle ne fonctionnaient pas correctement, ce qui a retardé certaines étapes du TP

---

## Conclusion

Ce TP nous a permis de découvrir concrètement les fonctionnalités de Docker. Nous avons appris à :

- Installer et configurer Docker
- Lancer et gérer des conteneurs
- Créer nos propres images Docker
- Déployer une application web avec Flask dans un environnement isolé

Ces compétences seront très utiles pour la suite, notamment dans les projets DevOps et le déploiement d'applications cloud.