

# TP Docker

1. Créer une application à partir d'un dossier vide contenant un fichier python main.py et requirements.txt
2. Créer un environnement virtuel et l'activer.
3. Remplir le fichier main.py avec un backend flask permettant d'afficher à chaque fois un nouveau numéro aléatoire. Voici un exemple d'un backend avec flask :

```
from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/')
def home():
    return 'Hello world'

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

4. Tester le backend en local.
5. Créer un docker-file permettant de lancer votre application en se basant sur une image python 3.10-alpine.
6. Faire un build pour créer l'image en l'appelant test-serveur-flask:v0
7. Faire un run de l'application en arrière plan (flag -d) et ouvrir le port 5000 du conteneur sur le port 8080 de votre machine.
8. Tester si l'application fonctionne (sur le port 8080).
9. Changer la version de python dans le Dockerfile vers la version 3.12-alpine et taguer cette nouvelle version de l'image avec v1
10. Faire un run de l'application en arrière plan (flag -d) et ouvrir le port 5000 du conteneur sur le port 80 de votre machine.
11. Créer un docker-compose et monter un volume sur le fichier main.py
12. Lancer votre application en utilisant docker-compose
13. Changer le contenu du rendu dans le fichier main.py et voir si l'application se met à jour.