

## CONTENEURISATION AVEC DOCKER(B2)

### TP A1: Docker ( Dockerfile, Image et Conteneur)

#### Objectifs

1. Comprendre les concepts de base de Docker : Dockerfile, image, conteneur.
2. Créer un Dockerfile pour déployer une application simple.
3. Construire une image Docker.
4. Exécuter un conteneur à partir de cette image.

#### Prérequis

- Avoir Docker installé sur votre machine.
- Connaissance de base en ligne de commande.

#### 1. Création du Dockerfile

Créez un fichier nommé Dockerfile dans un nouveau répertoire de travail, puis ajoutez le contenu suivant :

```
# Utilisation de l'image officielle Python comme base
FROM python:3.9
# Définition du répertoire de travail
WORKDIR /app
# Copie des fichiers de l'application dans l'image
COPY app.py /app/
# Installation des dépendances (si nécessaire)
COPY requirements.txt /app/
RUN pip install -r requirements.txt
# Définition de la commande de lancement
CMD ["python", "app.py"]
```

#### 2. Application simple

Créez un fichier app.py dans le même répertoire avec le contenu suivant :

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def home():
    return "Bonjour à tous , Ceci est une simple application conteneuriser avec Docker par
mon_nom!"
if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000)
```

Et un fichier requirements.txt contenant :

```
flask
```

#### 3. Construction de l'image Docker

- Exécutez la commande construire une image Docker nommée **mon\_nom**-flask-app.

**NB** : Remplacer « mon\_nom » par votre nom

- Pousser cette image sur votre Dockerhub en public

#### 4. Exécution du conteneur

Lancez un conteneur à partir de l'image construite

**Résultat attendu :**

- Le lien du dépôt github contenant :
  - Le dossier de l'application
  - Un fichier contenant l'historique des commandes exécutées
  - Une capture le lien web de l'application
  
- Le lien de l'image sur dockerhub