TP2 DEVOPS :

Objectifs :

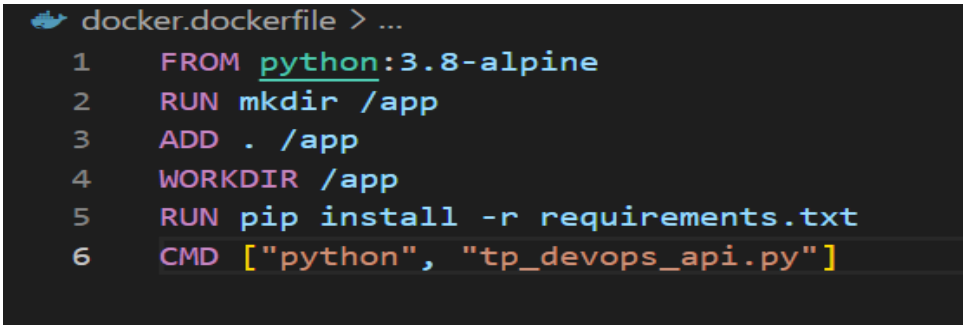
l'Objectif du TP 2 devops est de configurer un workflow Github Action, de le transférer en API à l'aide des fichiers du premier TP. Ensuite, automatiser la publication à chaque push sur Docker Hub.

Contenu du code :

Fichiers :

- Dockerfile : Les Dockerfiles sont des fichiers qui permettent de construire une image Docker adaptée à nos besoins, c'est pour cela qu'on l'utilise pour créer notre environnement.

  Code :



- tp_devops_api : Le wrapper est nécessaire pour faire l'appel à notre API, pour cela on utilise le os.environ pour récupérer les variables depuis la commande docker -env. A l'aide du request on peut récupérer les données des villes qu'on veut et les afficher au format souhaité (JSON). L''api est basé sur le wrapper, on récupère la latitude et longitude avec des request flask pour permettre de les ajouter dans l'url. On lance l'api sur le localhost port:8081 comme demandé sur le TP.

  Code :

```python
import requests
import json
from flask import Flask,  request, jsonify
import os
app = Flask(__name__)
api_key = os.environ['API_KEY']


with app.app_context():
    @app.route("/")
    def meteo():
        lat = request.args.get('lat')
        lon = request.args.get('lon')
        url = "https://api.openweathermap.org/data/2.5/weather?lat=%s&lon=%s&appid=%s&units=metric" % (
            lat, lon, api_key)
        response = requests.get(url)
        data = json.loads(response.text)
        if response.status_code != 200:
            return jsonify({
                'status': 'error',
                'message': 'La requête à l\'API météo n\'a pas fonctionné. Voici le message renvoyé par l\'API : {}'.format(data)
            }), 500

        return jsonify({
            'status': 'ok',
            'data': data
        })


if __name__ == "__main__":
    app.run(port=8081, debug=True)
```

- requirements.txt : Il s'agit de la liste des paquets Python dont l'installation est requise dans un environnement virtuel pour que l'application s'exécute correctement.

Automatisation du push :

- le work FLOW : à l'aide de GitHub actions on peut faire la configuration qui nous permet par la suite d'automatiser le push de l'image sur Docker Hub.

```yaml
name: Docker Image CI


on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]
```

```yaml
jobs:

  build:

    runs-on: ubuntu-latest

    steps:
      -
        name: Set up QEMU
        uses: docker/setup-qemu-action@v2
      -
        name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v2
      -
        name: Login to DockerHub
        uses: docker/login-action@v2
        with:
          username: ${{ secrets.DOCKERHUB_USERNAME }}
          password: ${{ secrets.DOCKERHUB_TOKEN }}
      -
        name: hadolint
        uses: reviewdog/action-hadolint@v1
        with:
          reporter: github-pr-review
      -
        name: Build and push
        uses: docker/build-push-action@v3
        with:
          push: true
          tags: anaszaghloul/tp2
```

- Secrets : on peut cacher nos identifiant de connexion Docker Hub dans la partie secrets de GITHUB.

Bonus :

- Données sensibles :

  Aucune données sensibles est stockées dans l'image ou le code source.

- handolint :

```yaml
name: Docker Image CI


on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]


  jobs:

    build:

      runs-on: ubuntu-latest

      steps:
        -
          name: Set up QEMU
          uses: docker/setup-qemu-action@v2
        -
          name: Set up Docker Buildx
          uses: docker/setup-buildx-action@v2
        -
          name: Login to DockerHub
          uses: docker/login-action@v2
          with:
            username: ${{ secrets.DOCKERHUB_USERNAME }}
            password: ${{ secrets.DOCKERHUB_TOKEN }}
        -
          name: hadolint
          uses: reviewdog/action-hadolint@v1
          with:
            reporter: github-pr-review
        -
          name: Build and push
          uses: docker/build-push-action@v3
          with:
            push: true
            tags: anaszaghloul/tp2
```