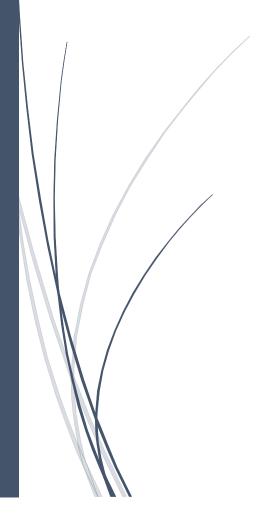
04/06/2022

TP1

DevOps - Docker



ABED Karim EFREI PARIS Voici le wrapper qui permettra de faire la connexion à l'API de OpenWeather, il permet aussi de requêter l'Api pour récupérer les informations météorologiques.

```
🥏 арр.ру
from requests import Request, Session, Response
      import os
      def weather(lat, long, apiKey):
          try:
             url = 'http://api.openweathermap.org/data/2.5/weather'
             params = {
                 'lon': long,
                 'appid': apiKey
             session = Session()
             request = Request('GET', url, params=params)
             prepped = request.prepare()
             response = session.send(prepped)
             return response.json()
          except Exception as e:
             return e
      lat = os.environ['LAT']
      long = os.environ['LONG']
      apiKey = os.environ['API KEY']
      print(weather(lat,long,apiKey))
```

Voici le Dockerfile qui permettra de définir les étapes qui seront exécuter dans un ordre bien précis.

```
Dockerfile X
UNTITLED (WORKSPACE)
                                  ● Projets > DevOps > TP1_docker > → Dockerfile > ...
                                   1 FROM python:3.8
∨ O Projets
 > Advanced Web
                                        RUN apt update
 RUN apt install python3
    🙀 ~$1_devops_ABED_Karim.odt
   🥏 арр.ру
                                        WORKDIR D:/Projets/DevOps/TP1_docker
    🔒 requirements.txt
                                       COPY app.py ./
    TP1_devops_ABED_Karim.odt
                                        # copy the dependencies file to the working directory
                                        COPY requirements.txt .
                                        RUN pip install -r requirements.txt
```

Construisons l'image docker avec cette commande :

docker build -t devopstp1.

```
Administrateur : Windows PowerShell
(base) PS D:\Projets\DevOps\TP1 docker> docker build -t devopstp1 .
[+] Building 64.4s (13/13) FINISHED
```

Maintenant que l'image docker est construite, nous pouvons tester de la faire fonctionner

```
Administrateur: Windows PowerShell

Administrateur: Windows PowerShell

base) PS D: \Projets \DevOps\TP1_docker> docker run --env LAT="5.902785" --env LONG="102.754175" --env API_KEY="240aa659f4db4e154a07d0459c30a347" devopstp1

'coord': ('lon': 102.7542, 'lat': 5.9028), 'weather': {{'id': 804, 'main': 'clouds', 'description': 'overcast clouds', 'icon': '04n'}], 'base': 'stations', 'main': {'temp': 300.76, 'feels_like': 302.86, 'temp_
n': 300.76, 'temp_nax': 300.76, 'pressure': 1008, 'humidity': 68, 'sea_level': 1008, 'grnd_level': 982), 'visibility': 10000, 'wind': {'speed': 2.44, 'deg': 77, 'gust': 2.46), 'clouds': {'all': 100}, 'dt': 165

45437, 'sys': {'country': 'MY', 'sunrise': 1654296841, 'sunset': 1654341646}, 'timezone': 28800, 'id': 1736405, 'name': 'Jertih', 'cod': 200}
```

Et lorsqu'on essaye de la faire marcher, elle nous renvoie bien les infos météorologiques.

Et enfin, nous allons pusher le conteneur sur DockerHub pour qu'il soit accessible et utilisable.

```
| Reposition | No. | Public | December | Reposition | Real | December | Reposition | Reposition
```