

Jurnal Faza 3 / Partea 1

Cerințele rezolvate

Pentru a treia parte a proiectului am facut deploy la aplicatia dezvoltata in Django la faza 2. Pentru a face deploy s-a folosit Docker.

Modul de rezolvare

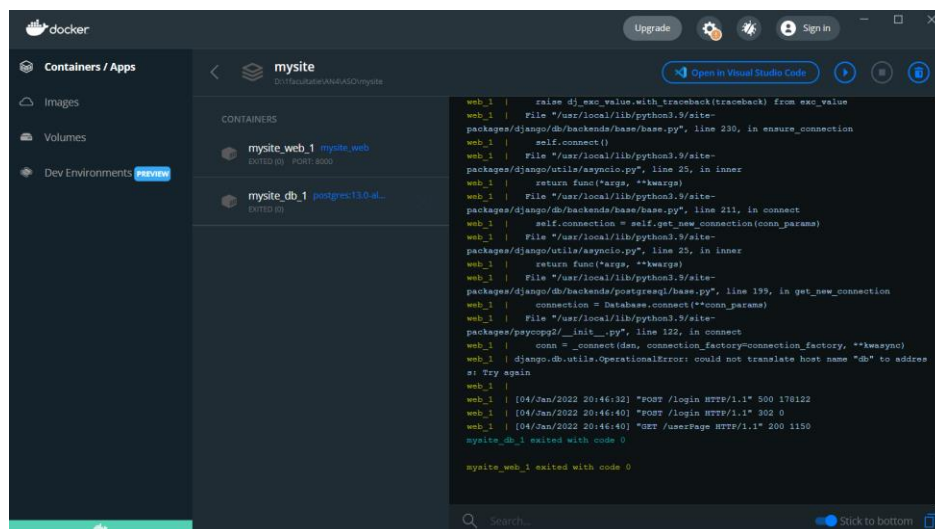
Pentru a realiza partea de deploy, am creat fisierele Dockerfile si docker-compose care face build si creaza imaginile in Docker. Am creat imaginea pentru setarile facute pana acum si am dat run container-ului. Pe laga deploy-ul aplicatiei Django, am realizat si deploy-ul bazei de date care s-a realizat in postgres.

Urmatorul pas este sa stergem baza de date existenta(SQLite) si sa adaugam o baza de date reala.

Pasii urmati pentru a face acest lucru:

- am adaugat un nou service pentru baza de date in fisierul docker-compose.yml
- am adaugat variabile pentru baza de date in fisierul .env.dev
- am actualizat valorile pentru baza de date in fisierul settings.py
- am adaugat comenzi necesare pentru instalarea pachetului psycopg2 in Dockerfile.
- am adaugat psycopg2 in fisierul de requirements In final am recreat imaginea cu modificarile facute si am verificat functionalitatea site-ul cu aceste setari.

Intr-un final imaginile in docker sunt reprezentate in figura de mai jos:



Probleme întâlnite și modul de rezolvare

Singura problema întâlnită a fost eroarea: "Cannot use ImageField because Pillow is not installed". Am rezolvat această problemă adăugând următoarele linii de cod în Dockerfile:

```
# install pillow dependencies
```

```
RUN apk add --no-cache jpeg-dev zlib-dev
```

```
RUN apk add --no-cache --virtual .build-deps build-base linux-headers \ && pip install Pillow
```

Concluzii

În concluzie, dezvoltarea aplicației a dus la aprofundarea termenului de deploy și folosirea și înțelegerea termenilor din Docker. Pe lângă acest lucru am învățat diferite comenzi din Dockerfile și din docker-compose și să înțeleg scopul acestora.