

Jurnal Faza 3

1. Cerințele rezolvate:

Obiectivul acestei etape este de crea un deploy cat mai portabil pentru aplicatie prin utilizarea containerelor Docker. Baza de date este inlocuita cu o baza de date reala, nu mai este doar un fisier si va rula intr-un contaier separat.

Modul de rezolvare:

Pentru început, am conectat aplicatia la o baza de date reala si anume PostgreSQL.

```
DATABASES = {
    "default": {
        "ENGINE": os.environ.get("SQL_ENGINE", "django.db.backends.sqlite3"),
        "NAME": os.environ.get("SQL_DATABASE", BASE_DIR / "db.sqlite3"),
        "USER": os.environ.get("SQL_USER", "user"),
        "PASSWORD": os.environ.get("SQL_PASSWORD", "password"),
        "HOST": os.environ.get("SQL_HOST", "localhost"),
        "PORT": os.environ.get("SQL_PORT", "5432"),
    }
}
```

Pentru deploy-ul aplicatiei in mod debug am creat un Dockerfile care va crea imaginea pentru python, directorul de lucru, va crea legatura pentru baza de date prin entrypoint, si se va asigura ca cerintele din fisierul requirements sunt indeplinite. Intre baza de date si site-ul web se face legatura prin fisierul docker-compose.yml.

Pentru deploy am creat un fisier Dockerfile si unul docker-compose.yml pentru a crea o imagine de Docker, care va urma sa fie uploadata pe Docker. Docker-compose realizeaza si legatura cu baza de date.

```

# pull official base image
FROM python:3.9.6-alpine

# set work directory
WORKDIR /usr/src/app

# set environment variables
ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1

# install psycopg2 dependencies
RUN apk update \
    && apk add postgresql-dev gcc python3-dev musl-dev

# install dependencies
RUN pip install --upgrade pip
COPY ./requirements.txt .
RUN pip install -r requirements.txt

# copy project
COPY . .

```

```

version: "3.8"

services:
  web:
    build: .
    command: python manage.py runserver 0.0.0.0:8000
    ports:
      - 8000:8000
    env_file:
      - ../.env.dev
  db:
    image: postgres:13.0-alpine
    volumes:
      - postgres_data:/var/lib/postgresql/data/
    environment:
      - POSTGRES_USER=hello_django
      - POSTGRES_PASSWORD=hello_django
      - POSTGRES_DB=hello_django_dev

volumes:
  postgres_data:

```

3. Probleme înatâlnite și modul de rezolvare :

Deoarece am implementat doar primi 2 pasi ai tutorialului, nu am intampinat nici o problema/dificultate.

4. Concluzii:

Am invatat sa creez un Dockerfile si sa ii dau deploy proiectului cu baza de date pe Docker.

Clotan Andrei
Gr 30462