**Азбукин Даниил БСБО-10-21**

**Практическая №4**

App.py:

from flask import Flask  
  
app = Flask(\_\_name\_\_)  
  
  
@app.route('/')  
def hello\_world():  
 return 'Hello, Docker!'  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(host='0.0.0.0', port=8081)

Dockerfile:

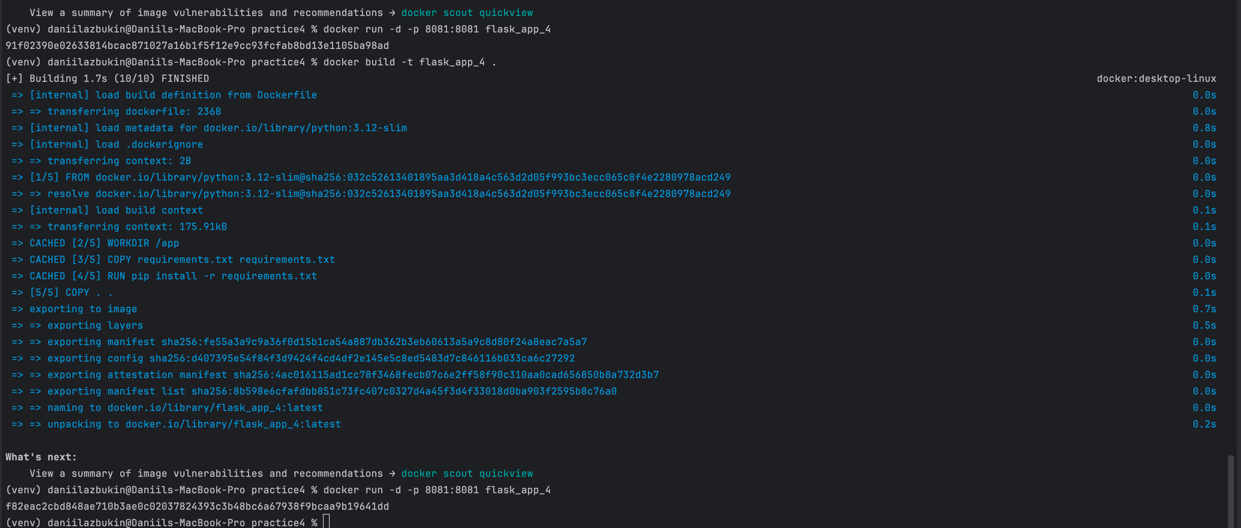
FROM python:3.12-slim  
  
WORKDIR /app  
  
COPY requirements.txt requirements.txt  
RUN pip install -r requirements.txt  
  
COPY . .  
  
EXPOSE 8081  
  
CMD ["python", "app.py"]

Requirements.txt

Flask~=3.0.3

docker build -t flask\_app\_4 . – Создаем docker образ нашего приложения

docker run -p 8081:8081-d flask\_app\_4 – Запускаем контейнер с приложением



Чтобы уменьшить размер образа и контейнера, используем более легкую версию python:

Dockerfile:

FROM python:3.12-alpine  
  
WORKDIR /app  
  
COPY requirements.txt requirements.txt  
RUN pip install -r requirements.txt  
  
COPY . .  
  
EXPOSE 8081  
  
CMD ["python", "app.py"]

Создадим тесты и добавим их запуск в Dockerfile:

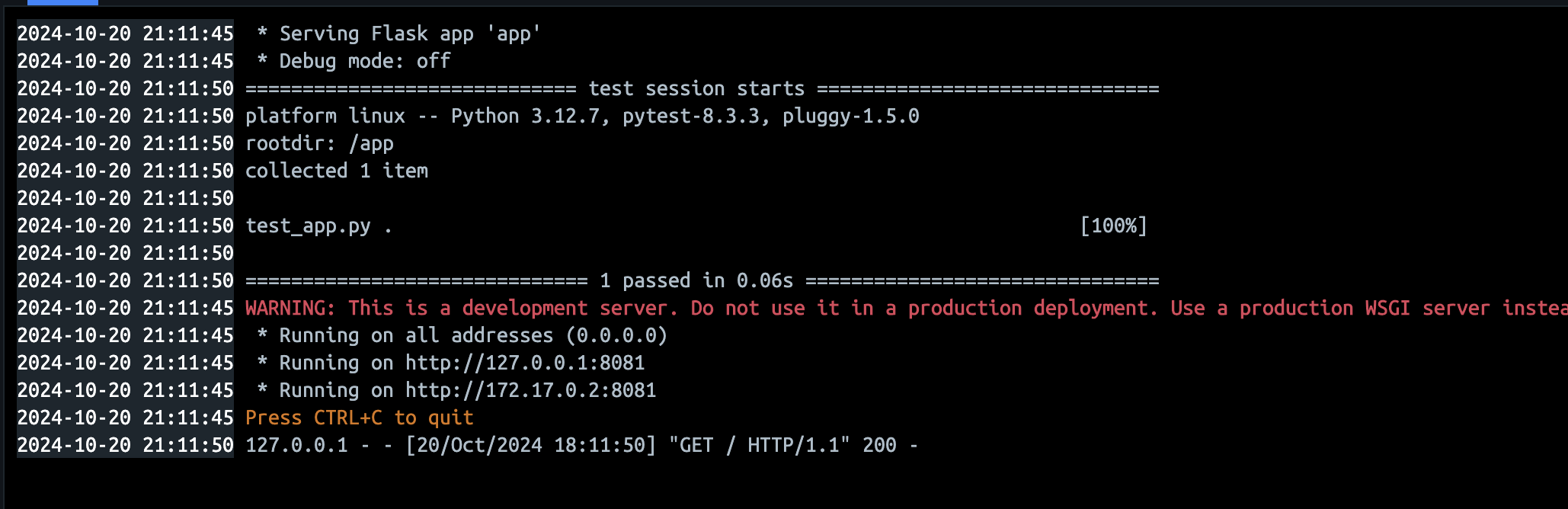
test\_app.py:

from pytest import mark, fixture  
from requests import Session, Response  
  
  
class BaseSession(Session):  
 def \_\_init\_\_(self, \*\*kwargs):  
 self.base\_url = kwargs.pop('base\_url')  
 super().\_\_init\_\_(\*\*kwargs)  
  
 def request(self, method, url, \*\*kwargs) -> Response:  
 response = super().request(method, url=f'{self.base\_url}{url}', \*\*kwargs)  
 return response  
  
  
@fixture  
def session():  
 return BaseSession(base\_url='http://localhost:8081')  
  
  
def test\_check\_status\_code(session):  
 response = session.get('/')  
 assert response.status\_code == 200

Dockerfile:

FROM python:3.12-alpine  
  
WORKDIR /app  
COPY requirements.txt requirements.txt  
RUN pip install -r requirements.txt  
  
COPY . .  
  
EXPOSE 8081  
  
CMD ["sh", "-c", "python app.py & sleep 5 && pytest"]

Перебилдим и запустим проект и посмотрим логи контейнера:



Создадим docker-compose.yaml с сервисами: наше приложение и postgres:

docker-compose.yaml:

version: '3'  
services:  
 web:  
 build: .  
 ports:  
 - "8081:8081"  
 depends\_on:  
 - db  
 db:  
 image: postgres:13  
 environment:  
 POSTGRES\_USER: user  
 POSTGRES\_PASSWORD: password  
 POSTGRES\_DB: flask\_db  
 ports:  
 - "5432:5432"

docker-compose build – Билдим проект

docker-compose up – Поднимаем приложение с базой данных

