

1. Микросервисная архитектура и контейнеризация

1.1. Урок 2. Docker

Домашнее задание к уроку 2 - Docker

Напишите Dockerfile к любому приложению из директорий golang или python на ваш выбор (можно к обоим).

Образ должен собираться из официального базового образа для выбранного языка. На этапе сборки должны устанавливаться все необходимые зависимости, а так же присутствовать команда для запуска приложения.

Старайтесь следовать рекомендациям (Best Practices) из лекции при написании Dockerfile.

При запуске контейнера из образа с указанием проксирования порта (флаг -p или -P если указан EXPOSE) при обращении на localhost:port должно быть доступно приложение в контейнере (оно отвечает Hello, World!).

Сохраните получившийся Dockerfile в любом публичном Git репозитории, например GitHub, и пришлите ссылку на репозиторий.

Содержимое Dockerfile:

```
# syntax=docker/dockerfile:1

FROM python:3.11
LABEL maintainer="pin@to25.ru"

WORKDIR /usr/src/app

ADD . .

RUN pip install --no-cache-dir -r ./requirements.txt

EXPOSE 8080
ENTRYPOINT python ./app.py
```

Команда сборки:

```
it@gb-devops:/repos/GB-DevOps-Containerization/HW02$ docker build -t gb_hw:test .
[+] Building 47.7s (11/11) FINISHED
docker:rootless
.....
7.0s
=> => exporting layers
6.7s
=> => writing image
sha256:93778a9b9f12cbcb0cdb5acb8fa410248e6bf8d4bb87379dedc20e5feb7ddefd
0.1s
=> => naming to docker.io/library/gb_hw:test
0.2s
```

Запуск контейнера:

```
docker run -p 8080:8080 gb_hw:test
```

Проверка:

```
it@gb-devops:/repos/GB-DevOps-Containerization/HW02$ docker run -p 8080:8080 gb_hw:test
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 274-615-244
172.17.0.1 - - [25/Sep/2023 12:25:14] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [25/Sep/2023 12:25:14] "GET /favicon.ico HTTP/1.1" 404 -
172.17.0.1 - - [25/Sep/2023 12:25:16] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [25/Sep/2023 12:25:16] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [25/Sep/2023 12:25:16] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [25/Sep/2023 12:25:16] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [25/Sep/2023 12:25:17] "GET / HTTP/1.1" 200 -
```

```
it@gb-devops:/repos/GB-DevOps-Containerization$ curl localhost:8080 -v
* Trying 127.0.0.1:8080...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET / HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Content-Type: text/html; charset=utf-8
< Content-Length: 13
< Server: Werkzeug/1.0.1 Python/3.11.5
< Date: Mon, 25 Sep 2023 12:31:18 GMT
<
* Closing connection 0
Hello, World!it@gb-devops:/repos/GB-DevOps-Containerization$ |
```

gb-devops.local:8080

Hello, World!

Задание выполнено.