

Урок 4

Богданов А.Л.

Программирование | 9 | 3901 | 10.12.2022

Задание: 1) необходимо создать Dockerfile, основанный на любом образе (вы в праве выбрать самостоятельно).

В него необходимо поместить приложение, написанное на любом известном вам языке программирования (Python, Java, C, C#, C++).

При запуске контейнера должно запускаться самостоятельно написанное приложение.

или 2) Дома необходимо собрать образ и запустить из него контейнер:

Основой образа должна быть alpine.

Установить необходимо mariaDB.

Также не забудьте об уменьшении размера образа. Способ обсуждался на лекции.

Необходимо открыть порт для коммуникации с другими сущностями.

Для проверки решения необходимо подключить к такому контейнеру phpmyadmin.

Необходимо, чтобы в нем вы увидели данные из вашей БД.

Также при запуске необходимо смонтировать внешнюю папку для хранения данных БД вне контейнера.

Задание: 1) Необходимо создать Dockerfile, основанный на любом образе (вы в праве выбрать самостоятельно).

В него необходимо поместить приложение, написанное на любом известном вам языке программирования (Python, Java, C, C#, C++).

При запуске контейнера должно запускаться самостоятельно написанное приложение.

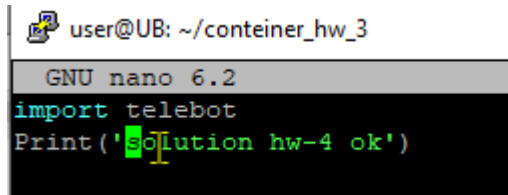
```
mkdir container_hw_3
cd container_hw_3
nano Dockerfile
FROM python:3.12.0b4-bookworm
RUN pip install --upgrade pip && \
    pip install pyTelegramBotAPI==4.12.0
WORKDIR /src
COPY test.py .
CMD python test.py
```

```
user@UB:~/container_hw_3$ docker build -t python_test .
[+] Building 26.4s (9/9) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 193B
=> [internal] load metadata for docker.io/library/python:3.12.0b4-bookworm
=> CACHED [1/4] FROM docker.io/library/python:3.12.0b4-bookworm@sha256:9f990877c5caaa3de50e9272f2
=> [internal] load build context
=> => transferring context: 28B
=> [2/4] RUN pip install --upgrade pip &&      pip install pyTelegramBotAPI==4.12.0
=> [3/4] WORKDIR /src
=> [4/4] COPY test.py .
=> exporting to image
=> => exporting layers
=> => writing image sha256:c05935596f295ecff06c6193c0f1f5d91da487d157b52d0daa62b955f570141d
=> => naming to docker.io/library/python_test
user@UB:~/container_hw_3$
```

```
nano test.py
```

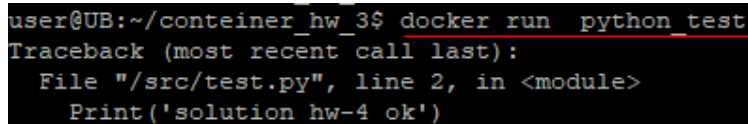
Вставляем код

```
import telebot  
Print('solution hw-4 ok')
```



Запускаем контейнер

```
docker run python_test
```



2) Дома необходимо собрать образ и запустить из него контейнер:

Основой образа должна быть alpine.

Установить необходимо mariaDB.

Также не забудьте об уменьшении размера образа. Способ обсуждался на лекции.

Необходимо открыть порт для коммуникации с другими сущностями.

Для проверки решения необходимо подключить к такому контейнеру phpmyadmin.

Необходимо, чтобы в нем вы увидели данные из вашей БД.

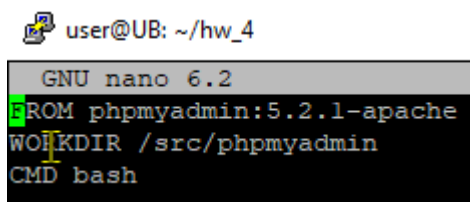
Также при запуске необходимо смонтировать внешнюю папку для хранения данных БД вне контейнера.

```
mkdir hw_4
```

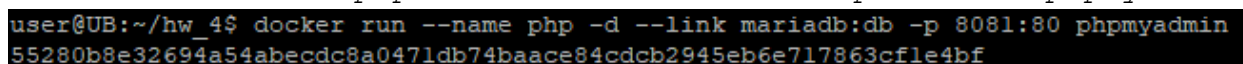
```
nano Dockerfile
```

Вставляем код

```
FROM phpmyadmin:5.2.1-apache  
WORKDIR /src/phpmyadmin  
CMD bash
```



```
docker run --name php -d --link mariadb:db -p 8081:80 phpmyadmin
```



Создаем директорию, где будет храниться база данных

```
mkdir /hw_4_db
```

```
nano Dockerfile
```

Вставляем код

```
FROM phpMyAdmin: 5.2.1-apache  
WORKDIR /src/phpmyadmin  
CMD bash
```

Создаем контейнер базы данных с монтированием директории к хосту

```
docker run --name mariadb -e MARIADB_ROOT_PASSWORD=pass -v /hw_4_db:/var/lib/mysql -d mariadb:10.10.2
```

```
user@UB:~/hw_4$ ^C
user@UB:~/hw_4$ docker run --name mariadb -e MARIADB_ROOT_PASSWORD=pass -v /hw_4_db:/var/lib/mysql -d mariadb:10.10.2
Unable to find image 'mariadb:10.10.2' locally
10.10.2: Pulling from library/mariadb
10ac4908093d: Pull complete
44779101e749: Pull complete
a721db3e3f3d: Pull complete
1850a929b84a: Pull complete
397a918c7da3: Pull complete
806be17e856d: Pull complete
634de6c90876: Pull complete
cd00854cfb1a: Pull complete
Digest: sha256:bfc25a68e113de43d0d112f5a7126df8e278579c3224e3923359e1c1d8d5ce6e
Status: Downloaded newer image for mariadb:10.10.2
deb5edfacdf72ca018d21bf7ddbef10a279b58ba8378fb309c8d04f0af52212
user@UB:~/hw_4$ docker ps
Команда «docker» не найдена. Возможно, вы имели в виду:
  command 'docker' from snap docker (20.10.24)
  command 'docker' from deb docker.io (20.10.21-0ubuntu1~22.04.3)
  command 'docker' from deb podman-docker (3.4.4+ds1-lubuntu1.22.04.1)
See 'snap info <snapname>' for additional versions.
user@UB:~/hw_4$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                NAMES
deb5edfacdf7   mariadb:10.10.2 "docker-entrypoint.s..." 23 seconds ago Up 20 seconds 3306/tcp             mariadb
```

```
docker ps -a
```

```
user@UB:~/hw_4$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                NAMES
55280b8e3269   phpmyadmin     "/docker-entrypoint..." 11 seconds ago Up 10 seconds 0.0.0.0:8081->80/tcp, :::8081->80/tcp   php
deb5edfacdf7   mariadb:10.10.2 "docker-entrypoint.s..." 10 minutes ago Up 10 minutes 3306/tcp             mariadb
user@UB:~/hw_4$
```

Смотрим ip

ip a

Заходим в браузер

http://192.168.1.120:8081/

