



Как собрать и запустить свой образ

Пошаговая инструкция простого приложения

1. Собираем образ

```
u18589902@CAB-WSM-0011308 TestDocker % docker build .
```

Точка в конце - это путь до Dockerfile'a. В примере я запускаю команду из места, где он и лежит

(опционально) Если вдруг надо как-то назвать image, то можно ввести команду `docker build -t name_image .`

Лог билда:

```
[+] Building 0.2s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
0.0s
=> => transferring dockerfile: 107B
0.0s
=> [internal] load .dockerignore
0.0s
=> => transferring context: 2B
0.0s
=> [internal] load metadata for docker.io/library/python:latest
0.0s
=> [1/3] FROM docker.io/library/python
0.1s
=> [internal] load build context
0.0s
=> => transferring context: 161B
0.0s
```

```
=> [2/3] WORKDIR /test
0.0s
=> [3/3] COPY . .
0.0s
=> exporting to image
0.0s
=> => exporting layers
0.0s
=> => writing image sha256:69beca55c4302fd3c819b465be5663425623ed73918a9c21f490e43a323fad
13                                0.0s
```

В конце нам выдали image, теперь надо его посмотреть

2. Смотрим image и находим id нашего image, который мы только что запускали

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<none>	<none>	69beca55c430	2 minutes ago	911MB
python	latest	6beb0d435def	5 days ago	911MB

3. Запустим наш контейнер

```
u18589902@CAB-WSM-0011308 TestDocker % docker run 69beca55c430
Hello, world!
```

Команда `docker run` каждый раз создает новый контейнер, с новым id.

4. Как убить контейнер, если это сервер

```
-> docker ps # чтобы проверить, какой контейнер запущен
-> docker stop <id_контейнера>
```

5. После того как мы убили наш контейнер, мы можем найти его id'шник по команде `docker ps -a` и запустить его снова

```
docker start <id_контейнера>
```

Здесь очень важно, что это не будет интерактивный режим и контейнер запускается в фоновом режиме и найти его можно будет по команде `docker ps`

6. Переименовать image

```
docker tag <old_name>:tag <new_name>:tag
```

Можно особо не париться и просто на месте тегов везде писать `latest`

7. Логин в докер-хаб

```
docker login
```

8. Пуш в докер-хаб

```
docker push <имя репозитория>
```

9. Запуск приложения через удаленный docker hub

```
docker run <имя репозитория>:tag
```

Запуск сервера через Dockerfile

В примере, где мы запускаем контейнер через `docker run` не запустилось бы приложение, которое как-то обращается к локальному или докеровскому порту, поэтому эту команду надо написать немного по-другому

```
docker run -p 3000:3000 <id_image>
```

Флажок `-p` как раз отвечает за порт, то есть так докер понимает, что дальше будет передавать порт.

`3000:3000` - это штука, которая маппит порты локально и внутри докера, то есть если локально у тебя проект собирается на порте `8080`, то правильный синтаксис команды был бы вот такой - `docker run -p 8080:3000 <id_image>`, то есть <порт локально>:<порт внутри докера>

Запуск сервера в фоновом режиме

Во всех предыдущих примерах после запуска команды `docker run` мы проваливались в консоль этого контейнера, но это не очень удобно. Чтобы этого избежать, можно написать вот такую команду

```
docker run -d -p 3000:3000 <id_image>
```

Флажок `-d` означает detached и не перекинет нас в консоль этого контейнера.