



Docker

Основные понятия

Docker — это платформа для разработки, доставки и запуска контейнерных приложений. Docker позволяет создавать контейнеры, автоматизировать их запуск и развертывание, управляет жизненным циклом. Он позволяет запускать множество контейнеров на одной хост-машине.

Основные части Docker'a:

1. Container

Контейнеры — это способ упаковать приложение и все его зависимости в единый образ. Этот образ запускается в изолированной среде, не влияющей на основную операционную систему. Контейнеры позволяют отделить приложение от инфраструктуры: разработчикам не нужно задумываться, в каком окружении будет работать их приложение, будут ли там нужные настройки и зависимости. Они просто создают приложение, упаковывают все зависимости и настройки в единый образ. Затем этот образ можно запускать на других системах, не беспокоясь, что приложение не запустится.

2. Image

Image - это неизменяемый файл (образ), из которого разворачиваются контейнеры. Приложения упаковываются именно в образы, из которых потом уже создаются контейнеры.

(своими словами про image) Это что-то вроде всех зависимостей, то есть для того, чтобы запустить приложение на питоне, то сначала надо скачать image питона, причем у каждого image есть теги, которые по факту являются версией этой самой зависимости.

3. Docker Hub (<https://hub.docker.com/>)

Склад всех image'й (например, ссылка на image питона
- https://hub.docker.com/_/python)

Список полезных команд

1. Команда, которая проверит, что docker есть на компьютере

```
u18589902@CAB-WSM-0011308 ~ % docker
```

2. Версия docker'a

```
u18589902@CAB-WSM-0011308 ~ % docker version
Client:
 Cloud integration: 1.0.17
 Version:          20.10.7 # версия самого docker'a
 API version:      1.41
 Go version:       go1.16.4 # (docker написан на Go)
 Git commit:       f0df350
 Built:            Wed Jun  2 11:56:22 2021
 OS/Arch:          darwin/amd64 # ОС самого докера
 Context:          default
 Experimental:     true
```

3. Вывод всех image'й

```
u18589902@CAB-WSM-0011308 ~ % docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
python        latest    6beb0d435def   5 days ago     911MB
u18589902@CAB-WSM-0011308 TestDocker % docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
<none>        <none>    69beca55c430   44 minutes ago 911MB
```

4. Вывод всех процессов

```
u18589902@CAB-WSM-0011308 ~ % docker ps -a # вообще все контейнеры (которые и завершились)
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS              PORTS
NAMES
e52a518325f1   python    "python3"               20 seconds ago   Exited (0) 8 seconds ago
stoic_bartik
```

```
u18589902@CAB-WSM-0011308 ~ % docker ps # только активные контейнеры, которые прямо сейчас запущены
```

```
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS          NAMES
```

5. Запуск контейнера с интерактивным взаимодействием с питоном

```
u18589902@CAB-WSM-0011308 ~ % docker run -it python # -it = interactive
```

6. Удаление контейнера

```
u18589902@CAB-WSM-0011308 ~ % docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS              PORTS
NAMES
**e52a518325f1**   python    "python3"               33 minutes ago   Exited (0) 33 minutes ago
stoic_bartik
u18589902@CAB-WSM-0011308 ~ % docker rm **e52a518325f1**
e52a518325f1
u18589902@CAB-WSM-0011308 ~ % docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS          NAMES
```

В команду `docker rm` можно передавать сразу несколько `id`-контейнеров

7. Удаление **всех** контейнеров, которые остановлены и не используются

```
u18589902@CAB-WSM-0011308 ~ % docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
```

8. Запуск не в интерактивном режиме с именем контейнера + порты

```
docker run -d -p 3000:3000 --name first_container --rm <id_image>
```