

С каждой из лабораторных работ, представленных ниже, начните работу аналогично тому как это было сделано на практике 1:

1. Авторизоваться в Docker Hub, введя логин и пароль, созданный на практическом занятии 1, потом авторизоваться тут: <https://labs.play-with-docker.com/>
2. Создаем новый инстанс (new instance).

## Лабораторная работа 1

### Задание 1

Проверяем, что docker работает, выводя версию на экран.

Укажите команду и приведите скриншот результата действия этой команды

Ваш ответ: `docker --version`

```
$ #####  
#                                     WARNING!!!!                                #  
# This is a sandbox environment. Using personal credentials                    #  
# is HIGHLY! discouraged. Any consequences of doing so are                    #  
# completely the user's responsibilities.                                       #  
#                                                                              #  
# The PWD team.                                                                #  
#####  
[node1] (local) root@192.168.0.18 ~  
$ docker --version  
Docker version 27.3.1, build ce12230  
[node1] (local) root@192.168.0.18 ~  
$ █
```

### Задание 2

Каким образом выводится справка по всем командам docker?

Ваш ответ: `docker --help`

```
[node1] (local) root@192.168.0.18 ~  
$ docker --help  
  
Usage:  docker [OPTIONS] COMMAND  
  
A self-sufficient runtime for containers  
  
Common Commands:  
  run       Create and run a new container from an image  
  exec      Execute a command in a running container  
  ps        List containers  
  build     Build an image from a Dockerfile  
  pull      Download an image from a registry  
  push      Upload an image to a registry  
  images    List images  
  login     Authenticate to a registry  
  logout    Log out from a registry  
  search    Search Docker Hub for images  
  version   Show the Docker version information  
  info      Display system-wide information
```

### Задание 3

3.1. Выведите справку по команде docker start, какие параметры должны быть указаны обязательно? Что означают параметры i, a?

Ваш ответ: `docker start --help` перечислите параметры: i, a

```
$ docker start --help

Usage:  docker start [OPTIONS] CONTAINER [CONTAINER...]

Start one or more stopped containers

Aliases:
  docker container start, docker start

Options:
  -a, --attach                Attach STDOUT/STDERR and forward signals
  --checkpoint string         Restore from this checkpoint
  --checkpoint-dir string     Use a custom checkpoint storage directory
  --detach-keys string        Override the key sequence for detaching a container
  -i, --interactive           Attach container's STDIN
```

- -i: позволяет взаимодействовать с контейнером через стандартный ввод.
- -a: подключает контейнер к стандартному выводу.

3.2. Выведите справку по команде docker stop, какие параметры должны быть указаны обязательно?

Ваш ответ: `docker stop --help`

```
$ docker stop --help

Usage:  docker stop [OPTIONS] CONTAINER [CONTAINER...]

Stop one or more running containers

Aliases:
  docker container stop, docker stop

Options:
  -s, --signal string  Signal to send to the container
  -t, --time int       Seconds to wait before killing the container
```

- -s, --сигнал для отправки в контейнер сильного сигнала
- -t, --время ожидания в секундах перед отключением контейнера

3.3. Выведите справку по команде, позволяющей удалить контейнер, какой обязательный параметр для этой команды должен быть указан?

Ваш ответ: `docker rm --help`, параметры: -f (--force), -l (--link), -v (--volumes)

```
$ docker rm --help

Usage:  docker rm [OPTIONS] CONTAINER [CONTAINER...]

Remove one or more containers

Aliases:
  docker container rm, docker container remove, docker rm

Options:
  -f, --force      Force the removal of a running container (uses SIGKILL)
  -l, --link        Remove the specified link
  -v, --volumes     Remove anonymous volumes associated with the container
```

3.4. Как выглядит команда, позволяющей вывести на экран только работающие в данный момент контейнеры

Ваш ответ: `docker ps`

#### Задание 4

Указывая какие ключи в команде по удалению контейнера, можно удалить работающий контейнер?

Ваш ответ: `-f`; `docker rm -f <container_id>`

#### Задание 5

Запустите контейнер, выполнив команду `docker run hello-world`, а затем удалите его.

Ваш ответ: впишите в поле ответа идентификатор контейнера, а затем команду по удалению контейнера с указанием его идентификатора.

```
[node1] (local) root@192.168.0.18 ~
$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
18cecc84c1e7   hello-world  "/hello"   3 seconds ago   Exited (0) 2 seconds ago           hopeful_albattani
[node1] (local) root@192.168.0.18 ~
$ docker rm 18cecc84c1e7
18cecc84c1e7
[node1] (local) root@192.168.0.18 ~
$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
[node1] (local) root@192.168.0.18 ~
$
```

#### Задание 6

Выведите на экран список образов, загруженных из Docker Hub.

Ваш ответ: `docker images`

```
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY      TAG         IMAGE ID      CREATED        SIZE
hello-world     latest     d2c94e258dcb  17 months ago  13.3kB
```

#### Задание 7

Выполним команду `docker run -it ubuntu bash`

Эта команда запускает образ `ubuntu`, а в нем программу `bash`. При этом контейнер запускается в интерактивном режиме (это происходит с помощью ключей `-it`). Если образа до этого в системе не было, он будет скачан из Docker Hub. После ввода команды и нажатия Enter, мы будем находиться внутри контейнера `ubuntu`, изменится строка ввода.

Пример

Было:

northbug@Nikolays-Mac ~ %

Стало:

root@5932fb4e4361:/#

Чтобы выйти из контейнера `ubuntu`, надо набрать `exit` и нажать Enter. Таким образом у нас будет загружен образ `ubuntu` и создан контейнер с этой ОС.

```
$ docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
d1fbec07a2e5: Pull complete
Digest: sha256:ab64a8382e935382638764d8719362bb50ee418d944c1f3d26e0c99fae49a345
Status: Downloaded newer image for ubuntu:latest
root@a5780a9dc383:/# exit
exit
[node1] (local) root@192.168.0.18 ~
$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
a5780a9dc383   ubuntu   "bash"    35 seconds ago   Exited (0) 6 seconds ago           hopeful_cerf
```

**7.1.** Удалите контейнер и образ с ubuntu из списка загруженных образов и созданных контейнеров. Какие команды потребуются для этого сделать?

Ваш ответ: `docker rm <container_id>; docker rmi ubuntu`

```
$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
a5780a9dc383   ubuntu   "bash"    35 seconds ago   Exited (0) 6 seconds ago   hopeful_cerf
[node1] (local) root@192.168.0.18 ~
$ docker rm a5780a9dc383
a5780a9dc383
[node1] (local) root@192.168.0.18 ~
$ docker rmi ubuntu
Untagged: ubuntu:latest
Deleted: sha256:dc4c1391d3701ce1105f6384632f71fd08abf4862c16f6612d74f262adc8665a
Deleted: sha256:fa0f10cc481ea09f029a04c379029a205cd975840c811f16286b28ae103c66ea
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    d2c94e258dcb   17 months ago   13.3kB
[node1] (local) root@192.168.0.18 ~
```

**7.2.** Исследуйте справку команды `docker run` научитесь задавать имя для запускаемого контейнера (если имя не задавать, то оно будет сгенерировано автоматически — см. Предыдущие запуски). Запустите контейнер с ОС debian с помощью команды: `docker run -i -t debian bash` и задайте ему имя 'mydebian'. Установите внутри контейнера программу nano с помощью выполнения команд:

```
apt update
apt install -y nano.
```

Выйдите из контейнера

Ваш ответ: `docker run -it --name mydebian debian bash`  
`apt install -y nano.`  
`exit`

```
$ docker run -it --name mydebian debian bash
Unable to find image 'debian:latest' locally
latest: Pulling from library/debian
cdd62bf39133: Pull complete
Digest: sha256:27586f4609433f2f49a9157405b473c62c3cb28a581c4133939751
Status: Downloaded newer image for debian:latest
root@f7d8482f7b51:/# apt update
0% [Connecting to deb.debian.org]
0% [Connecting to deb.debian.org]
0% [Connecting to deb.debian.org]
0% [Connecting to deb.debian.org]
0% [Connecting to deb.debian.org]^C
root@f7d8482f7b51:/# apt install -y nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package nano is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'nano' has no installation candidate
root@f7d8482f7b51:/# exit
exit
[node1] (local) root@192.168.0.18 ~
```

### Задание 8

Изучите справку команды для вывода логов (отчетов о работе) внутри контейнера, созданного в процессе работы над заданием 7: `docker logs`.

Выведите историю логов работы контейнера, с которым вы работали в предыдущем задании 7.2.

Ваш ответ: `docker logs mydebian`

```
$ docker logs mydebian
root@f7d8482f7b51:/# apt update
0% [Connecting to deb.debian.org]
0% [Connecting to deb.debian.org]

0% [Connecting to deb.debian.org]
0% [Connecting to deb.debian.org]
0% [Connecting to deb.debian.org]^C
root@f7d8482f7b51:/# apt install -y nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package nano is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'nano' has no installation candidate
root@f7d8482f7b51:/# exit
exit
[node1] (local) root@192.168.0.18 ~
$
```

### Задание 9

Запустите команду `docker run ubuntu bash` и опишите результат выполнения (в чем отличие от запуска этого образа и команды с ключами `i`, `t`).

Ваш ответ: с ключами `-it` мы получаем интерактивный терминал внутри контейнера, где мы можем выполнять команды вручную, а контейнер не завершается сразу после запуска `bash`.

### Задание 10

Используя справку `docker` и информацию в интернете ответьте на вопрос, в чем **разница** между командами:

`docker run ...`

`docker start ...`

Ваш ответ:

- `docker run`: комбинирует создание и запуск контейнера.
- `docker start`: запускает уже существующий (ранее созданный) контейнер.