

## Отчет Лабораторной работы №2

### 1) Цель работы

Изучить принципы работы контейнерной технологии Docker, установить Docker Desktop, создать и запустить собственный контейнер, а также научиться работать с Dockerfile и системой контроля версий GitHub.

### 2) Ход работы:

#### a) Установка docker desktop

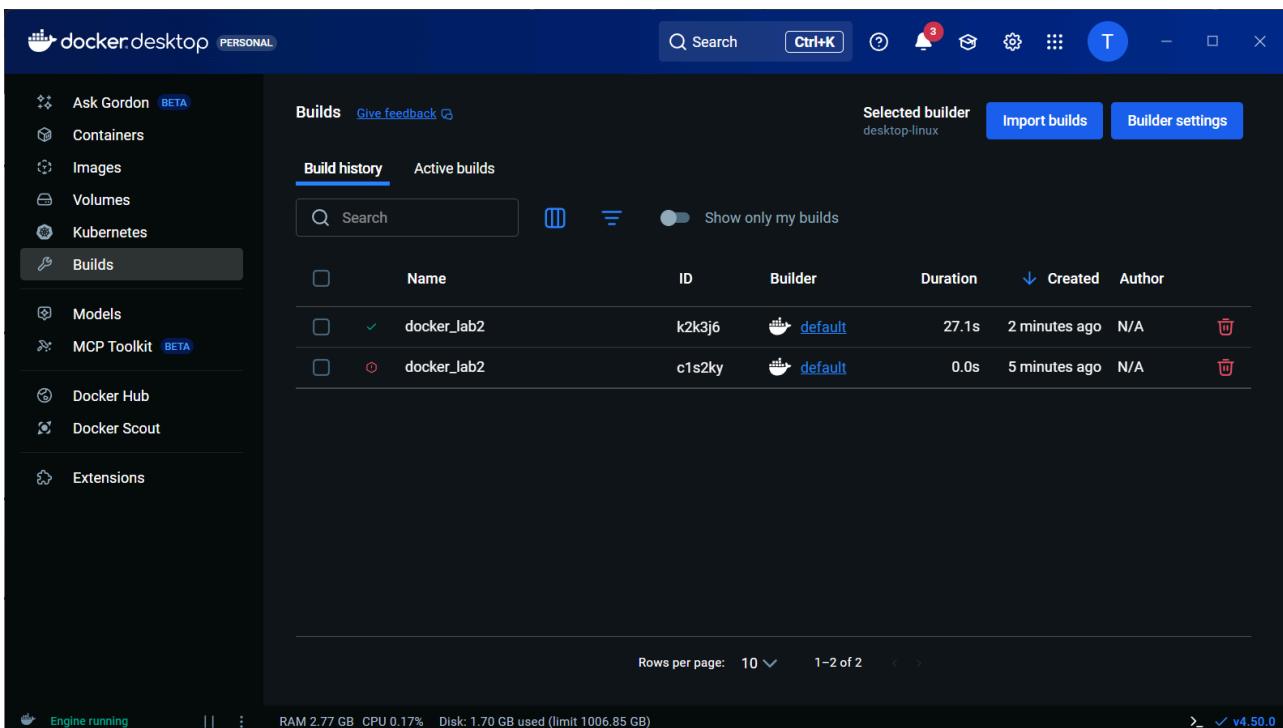
```
C:\Users\tpwrz>docker --version
Docker version 28.5.1, build e180ab8
```

#### b) Создание репозитория GitHub

#### c) Создание Dockerfile

#### d) Сборка образа

```
Q:\uni\docker\docker_lab2>docker build -t lab2 .
[+] Building 27.2s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 474B
=> [internal] load metadata for docker.io/library/ubuntu:latest
=> [auth] library/ubuntu:pull token for registry-1.docker.io
=> [internal] load .dockerrcignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/ubuntu:latest@sha256:66460d557b25769b102175144d538d88219c077c678a49af4afca6fbfc1
=> => resolve docker.io/library/ubuntu:latest@sha256:66460d557b25769b102175144d538d88219c077c678a49af4afca6fbfc1
=> => sha256:4b3ffd8ccb5201a0fc03585952effb4ed2d1ea5e704d2e7330212fb8b16c86a3 29.72MB / 29.72MB
=> => extracting sha256:4b3ffd8ccb5201a0fc03585952effb4ed2d1ea5e704d2e7330212fb8b16c86a3
=> => transferring context: 89B
=> [2/4] RUN apt update && apt install -y python3
=> [3/4] WORKDIR /app
=> [4/4] COPY main.py .
=> exporting to image
=> exporting layers
=> exporting manifest sha256:04e21c02a145b829141ee244545995c53ff079c37c79f5ad87a12fd6965e567
=> exporting config sha256:008bff7d208a1a2999505dbbf77e52071f087c096c1e2c58c3c48f895a7adce2
=> => exporting attestation manifest sha256:44e45e0d8e910b33b962dfcc9e51585f64fd2b01dec119aeaf5ac5b1acb93e
=> => exporting manifest list sha256:2df69c92b69c049b5e789307fb603e5882187814085ae2792451c397e1cbd6ce
=> => naming to docker.io/library/lab2:latest
=> => unpacking to docker.io/library/lab2:latest
WARNING: current commit information was not captured by the build: git was not found in the system: exec: "git.exe": executable file not found in %PATH%
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/k2k3j64oazrbh1bh9blpnu3s2
```



```
Q:\uni\docker\docker_lab2>docker images
REPOSITORY      TAG          IMAGE ID      CREATED             SIZE
lab2            latest        2df69c92b69c   About a minute ago   273MB
```

е) Запуск контейнера

```
Q:\uni\docker\docker_lab2>docker run --name mylab2 lab2
Привет из контейнера Docker!
```

ф) Проверка запущенных контейнеров

```
Q:\uni\docker\docker_lab2>docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
821e67b22063 lab2 "python3 main.py" 21 seconds ago Exited (0) 20 seconds ago
mylab2
```

г) Удаление и повторный запуск контейнера

```
Q:\uni\docker\docker_lab2>docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
821e67b22063 lab2 "python3 main.py" 21 seconds ago Exited (0) 20 seconds ago
mylab2

Q:\uni\docker\docker_lab2>docker stop mylab2
mylab2

Q:\uni\docker\docker_lab2>docker rm mylab2
mylab2

Q:\uni\docker\docker_lab2>docker run --name mylab2 lab2
Привет из контейнера Docker!
```

х) Добавление файлов в GitHub

```
tpwrz@tania MINGW64 /q/uni/docker/docker_lab2 (main)
$ git add .

tpwrz@tania MINGW64 /q/uni/docker/docker_lab2 (main)
$ git commit -m "Added dockerfile and python program"
error: pathspec '-' did not match any file(s) known to git
error: pathspec 'm' did not match any file(s) known to git
error: pathspec 'Added dockerfile and python program' did not match any file(s)
known to git

tpwrz@tania MINGW64 /q/uni/docker/docker_lab2 (main)
$ git commit -m "Added dockerfile and python program"
[main (root-commit) 6c212d3] Added dockerfile and python program
 2 files changed, 10 insertions(+)
  create mode 100644 Dockerfile
  create mode 100644 main.py

tpwrz@tania MINGW64 /q/uni/docker/docker_lab2 (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 612 bytes | 612.00 KiB/s, done.
```

3) Контрольные вопросы

а) Что такое контейнер и чем он отличается от виртуальной машины?

Контейнер - это изолированная среда, где запускается приложение со всеми зависимостями. Он отличается от виртуальной машины тем, что использует ядро хост-системы, а не отдельную ОС, поэтому работает быстрее и занимает меньше памяти.

б) Для чего используется Dockerfile?

Dockerfile - это файл с пошаговыми инструкциями для сборки образа: какие пакеты установить, какие файлы скопировать и какую команду выполнить при запуске.

с) Что делает команда docker build?

Команда читает Dockerfile и создаёт из него образ, который потом можно использовать для запуска контейнера.

- d) В чем разница между образом (image) и контейнером?  
Образ - это шаблон с настройками и файлами, а контейнер - это запущенный экземпляр этого образа.
- e) Как удалить все неиспользуемые контейнеры и образы в Docker?  
`docker system prune -a`
- f) Что произойдет, если изменить содержимое `main.py` и заново выполнить `docker build`?  
Если изменить `main.py` и снова выполнить `docker build`, Docker пересоберёт образ - пересоздаст слой, где лежит изменённый файл, и обновит результат.

4) Ссылка на репозиторий: <https://github.com/tpwrzz/docker-lab2>