# **Лабораторная работа №8**

# **DOCKER: ОСНОВЫ РАБОТЫ С КОНТЕЙНЕРАМИ**

**Цель работы:** освоить основные принципы работы с Docker, научиться создавать, запускать и управлять контейнерами, а также работать с Docker образами.

**Задание:**

* ***Установка Docker***

- Установите Docker на вашу систему.

- Проверьте корректность установки командой docker --version.

* ***Работа с образами***

- Загрузите образ nginx из Docker Hub.

- Запустите контейнер на основе этого образа и откройте веб-страницу Nginx в браузере.

* ***Создание пользовательского образа***

- Напишите Dockerfile для создания образа с простым веб-приложением на Python (например, Flask).

- Соберите образ и запустите контейнер.

* ***Работа с томами (Volumes)***

- Создайте том и подключите его к контейнеру для сохранения данных.

- Убедитесь, что данные сохраняются после перезапуска контейнера.

* ***Сеть в Docker***

- Создайте два контейнера (например, веб-сервер и базу данных).

- Настройте сеть для взаимодействия между ними.

* ***Docker Compose***

- Напишите файл docker-compose.yml для развёртывания многоконтейнерного приложения (например, веб-сервер + БД).

- Запустите приложение с помощью Docker Compose.

* ***Очистка ресурсов***

- Удалите все созданные контейнеры, образы и тома.

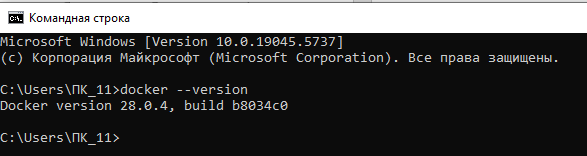


Рисунок 8.1 – **Результат выполнения задания 1**

1. Команды для загрузки и запуска ngnix:

- docker pull nginx

- docker run --name mynginx -d nginx -p 8080:80

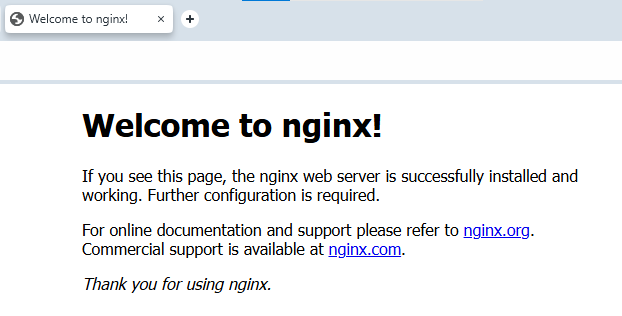


Рисунок 8.2 – **Результат выполнения задания 2**

1. Создание образа:

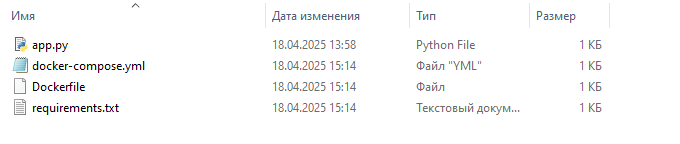


Рисунок 8.3 – **Структура образа**

Команда для запуска образа:

- docker build -t lab8 .

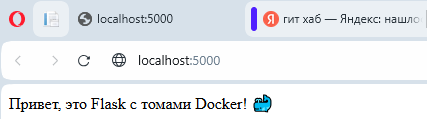


Рисунок 8.4 – **Результат выполнения**

1. Сначала необходимо изменить структуру приложения:

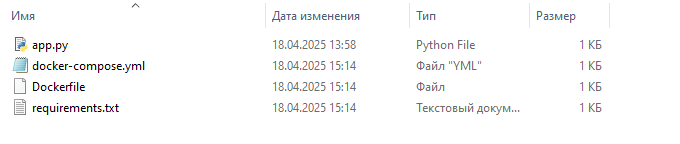


Рисунок 8.5 – **Структура приложения**

Команда для создания volume lab8data:

- docker volume create lab8data

Команда для запуска контейнера:

- docker run -d -p 8080:8080 --name lab8-containe -v lab8data:/app/data lab8

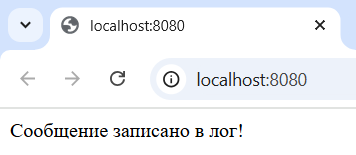


Рисунок 8.6 – **Результат выполнения задания 4**

Команда для проверки наличия записей в логе:

- docker run --rm -v lab8data:/data alpine cat /data/log.txt

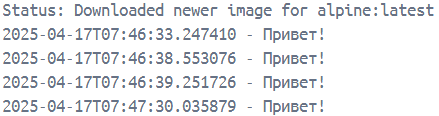


Рисунок 8.7 – **Результат выполнения команды**

1. Необходимо снова изменить структуру приложения:

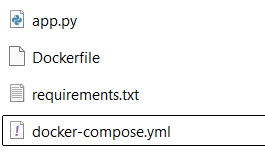


Рисунок 8.8 – **Структура приложения для задания 5**

1. Листинг docker-compose.yml:
2. version: '3.8'
3. services:
4. web:
5. build: .
6. ports:
7. - "8080:8080"
8. depends\_on:
9. - db
10. networks:
11. - mynet
12. db:
13. image: postgres:15
14. environment:
15. POSTGRES\_DB: testdb
16. POSTGRES\_USER: testuser
17. POSTGRES\_PASSWORD: testpass
18. volumes:
19. - pgdata:/var/lib/postgresql/data
20. networks:
21. - mynet
22. networks:
23. mynet:
24. volumes:
25. pgdata:

Запуск приложение с помощью docker-compose:

* docker-compose up –build

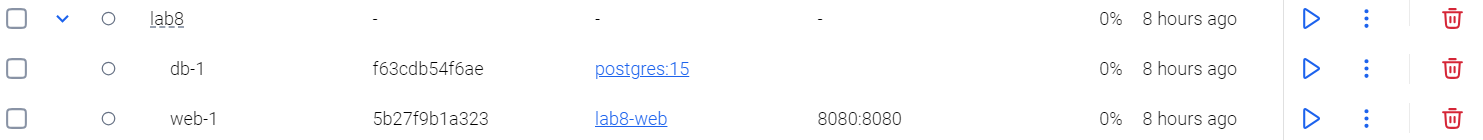


Рисунок 8.9 – **Появившиеся контейнеры**

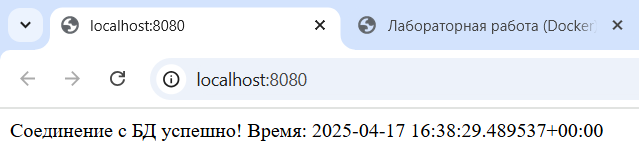
****

Рисунок 8.10 – **Результат выполнения задания 6**

**7)** Команда для удаления всех контейнеров:

- docker rm $(docker ps -aq)

Команда для удаления всех образов:

- docker rmi -f $(docker images -q)

Команда для удаления всех томов:

- docker volume rm $(docker volume ls -q)