**ЛАБОРАТОРНАЯ РАБОТА №7**

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:**

1. **Установка Docker**

* Установите Docker на вашу систему.
* Проверьте корректность установки командой docker --version.

1. **Работа с образами**

* Загрузите образ nginx из Docker Hub.
* Запустите контейнер на основе этого образа и откройте веб-страницу Nginx в браузере.

1. **Создание пользовательского образа**

* Напишите Dockerfile для создания образа с простым веб-приложением на Python (например, Flask).
* Соберите образ и запустите контейнер.

1. **Работа с томами (Volumes)**

* Создайте том и подключите его к контейнеру для сохранения данных.
* Убедитесь, что данные сохраняются после перезапуска контейнера.

1. **Сеть в Docker**

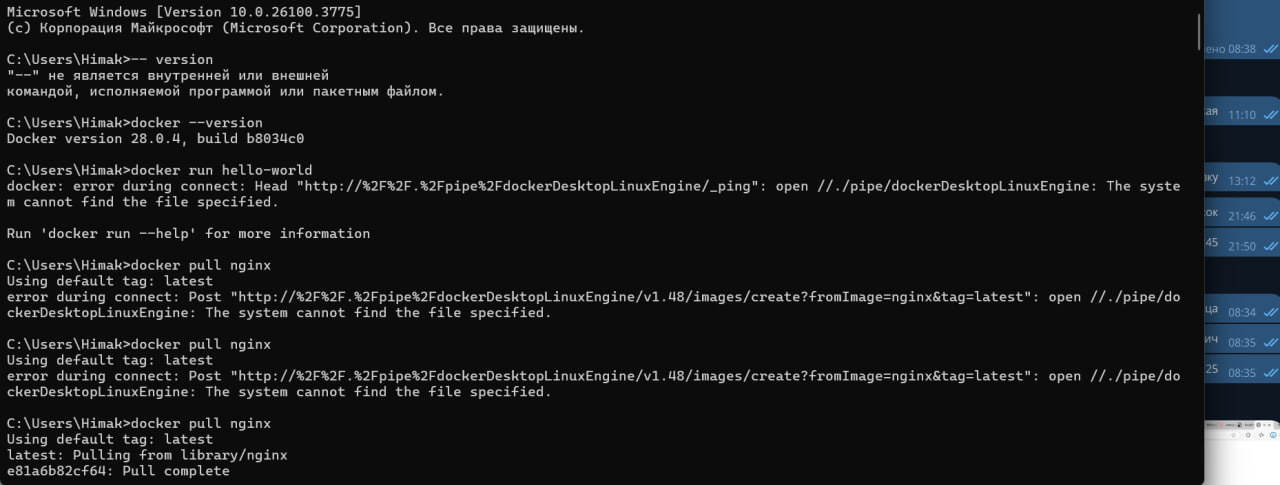
* Создайте два контейнера (например, веб-сервер и базу данных).
* Настройте сеть для взаимодействия между ними.

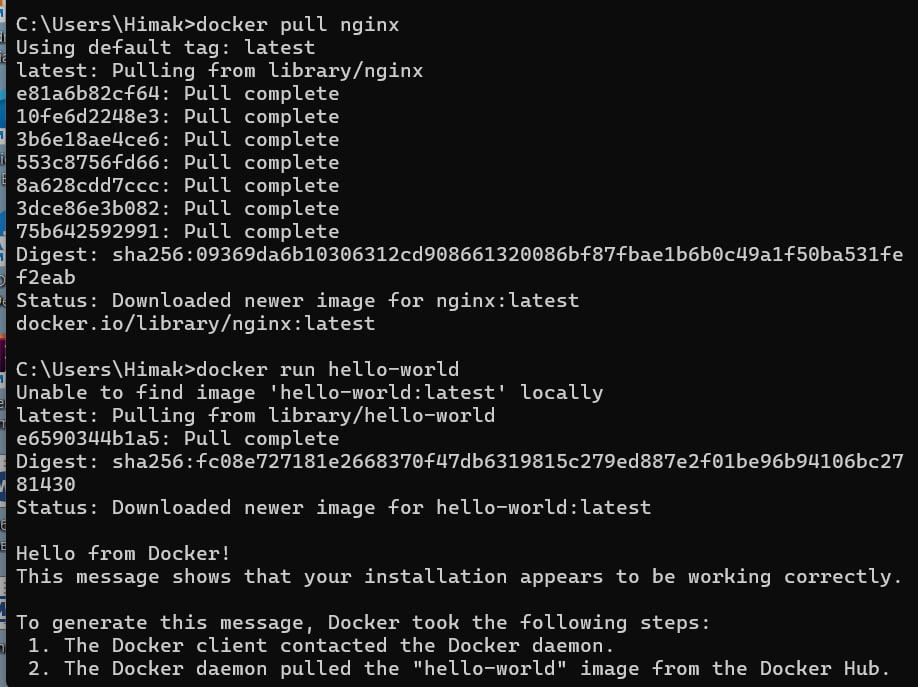
1. **Docker Compose**

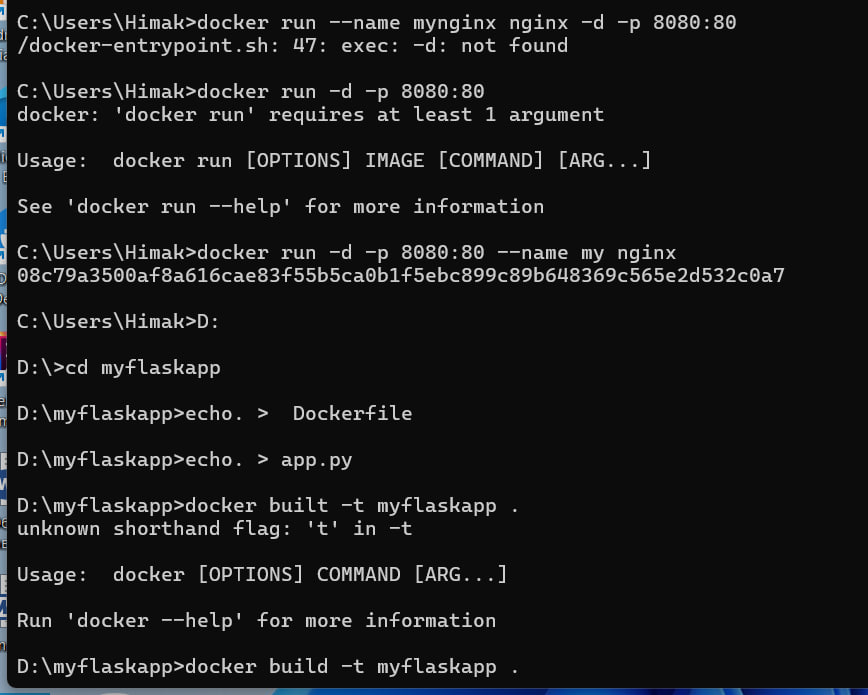
* Напишите файл docker-compose.yml для развёртывания многоконтейнерного приложения (например, веб-сервер + БД).
* Запустите приложение с помощью Docker Compose.

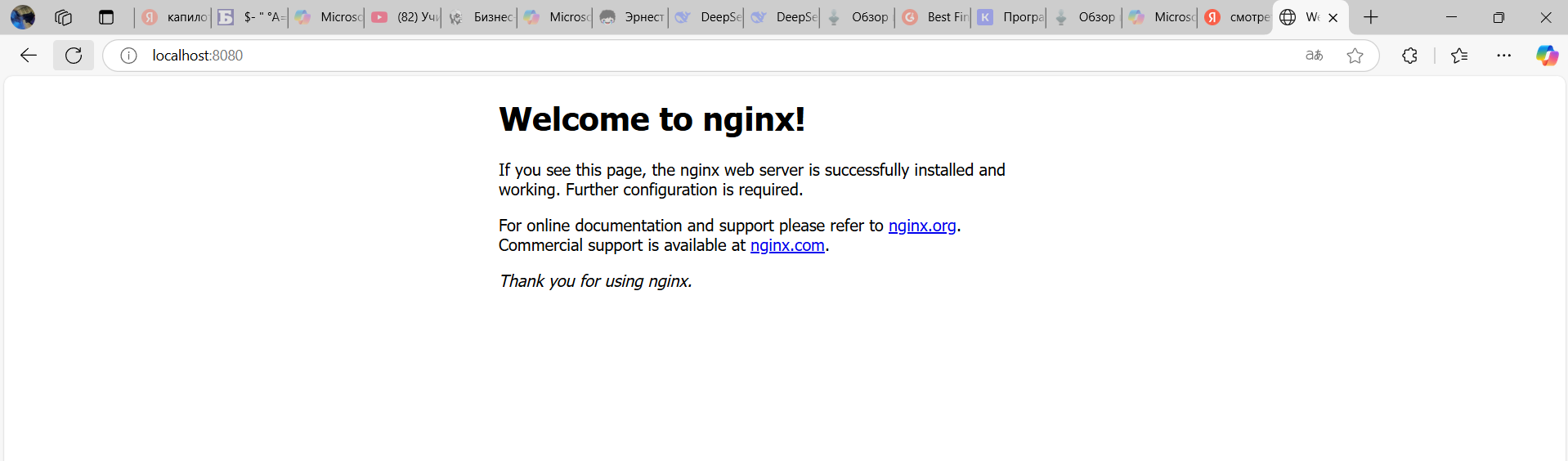
1. **Очистка ресурсов**

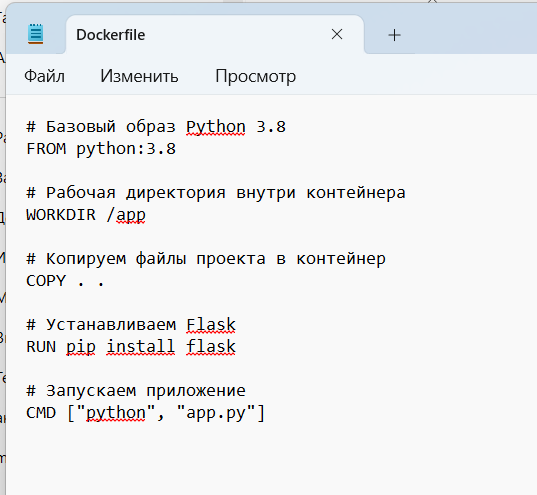
* Удалите все созданные контейнеры, образы и тома.

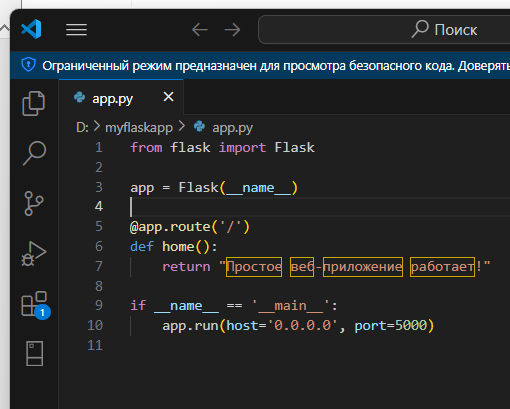


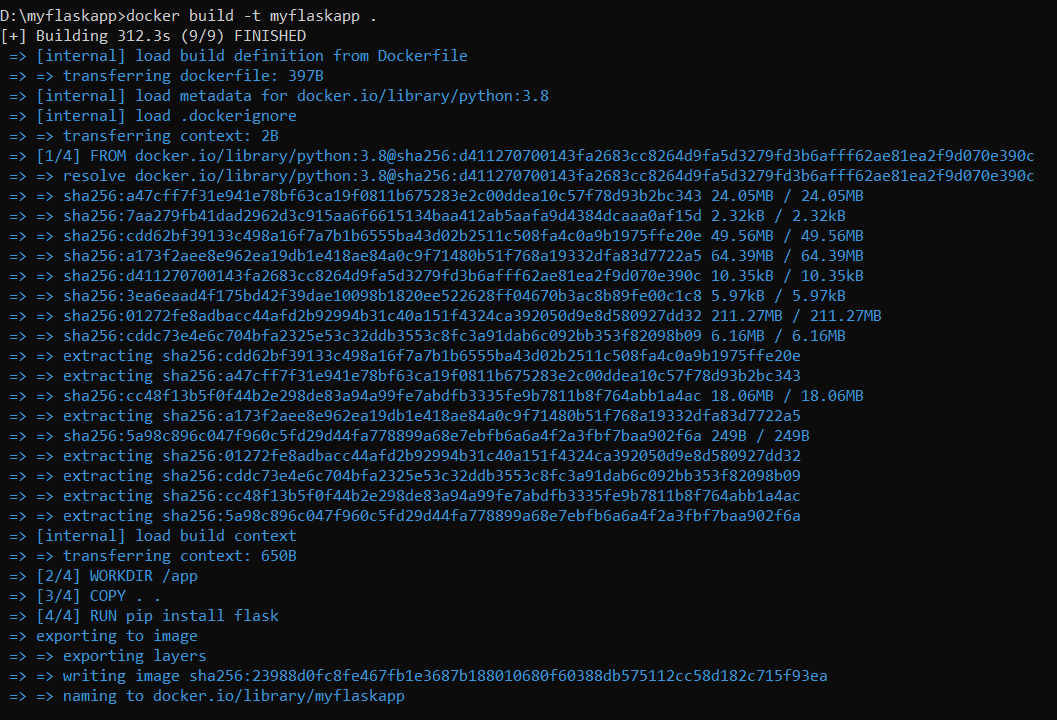


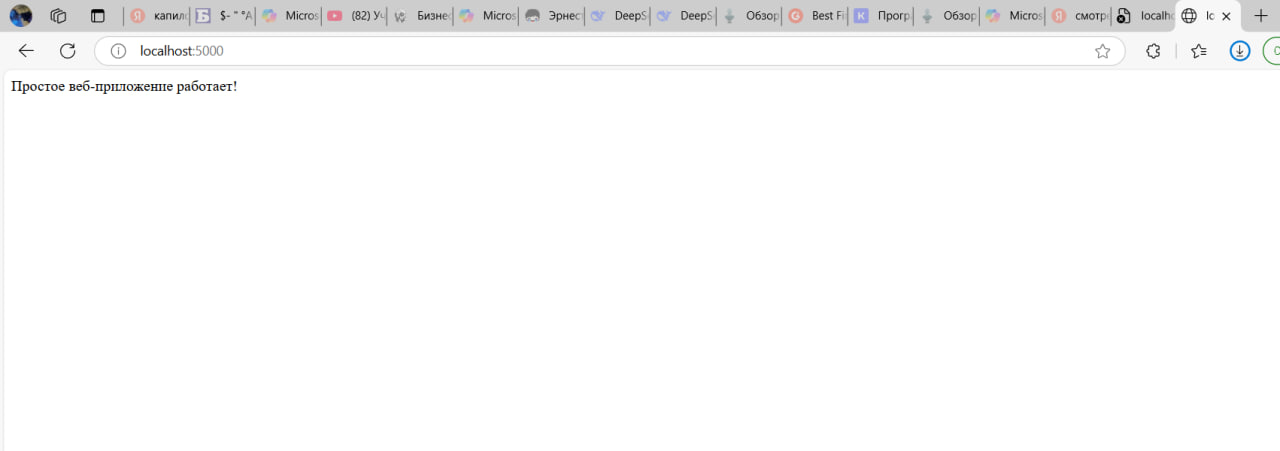












1. После этого была протестирована работа с томами. Создан том mydata (docker volume create mydata), который затем был подключён к контейнеру Nginx при его запуске. Проверка каталога /usr/share/nginx/html внутри контейнера показала наличие стандартных файлов index.html и 50x.html. Далее был создан файл test.html с тестовым содержимым, который оставался доступным после перезапуска контейнера, подтверждая успешность сохранения данных внутри тома.

