**Homework\_Lesson\_22**

Цель: получить практический опыт написания Dockerfile, развертывания приложений с использованием Docker-compose

Задание 1: Вам необходимо написать Dockerfile для создания контейнера с приложением веб-сервера на основе образа Ubuntu 20.04. Приложение должно быть запущено на порту 8080 и должно отдавать статические файлы из каталога /app/static.

Шаги, которые необходимо выполнить:

1. Создайте новый файл Dockerfile в пустой директории на вашем локальном компьютере.

2. Напишите инструкцию FROM, которая указывает базовый образ Ubuntu 20.04.

3. Установите необходимые зависимости с помощью инструкции RUN.

Установите пакеты nginx и curl, а также создайте каталог /app/static.

4. Скопируйте файл конфигурации nginx из вашего локального каталога внутрь контейнера с помощью инструкции COPY.

5. Скопируйте статические файлы из каталога /app/static на вашем локальном компьютере внутрь контейнера с помощью инструкции COPY.

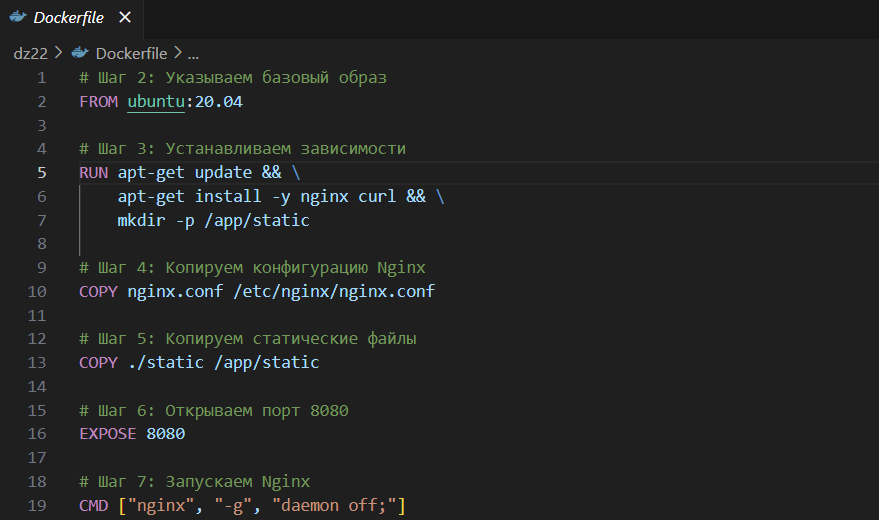
6. Используйте инструкцию EXPOSE для открытия порта 8080.

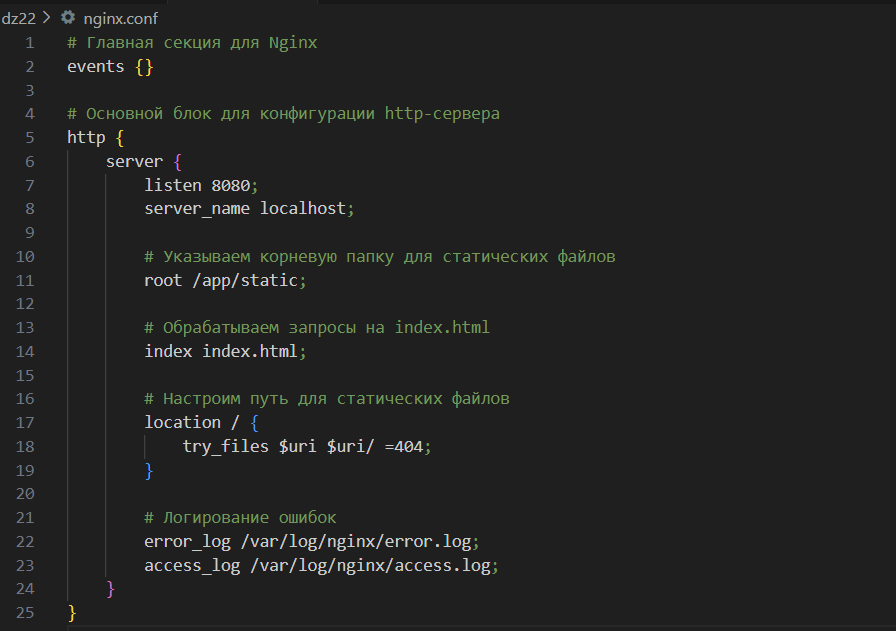
7. Используйте инструкцию CMD для запуска команды nginx с указанием пути к файлу конфигурации, который вы скопировали на шаге 4.

8. Сохраните файл Dockerfile и соберите образ с помощью команды docker build.

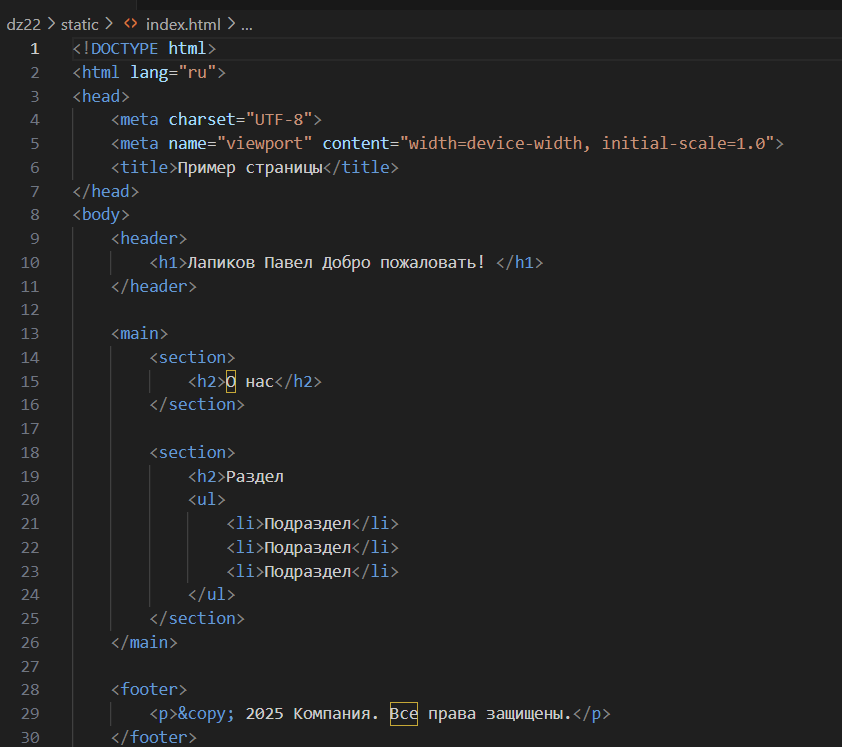
9. Запустите контейнер из образа с помощью команды docker run и проверьте, что веб-сервер отдает статические файлы из каталога /app/static на порту 8080.

**Подготовим Dockerfile и конфиг nginx**

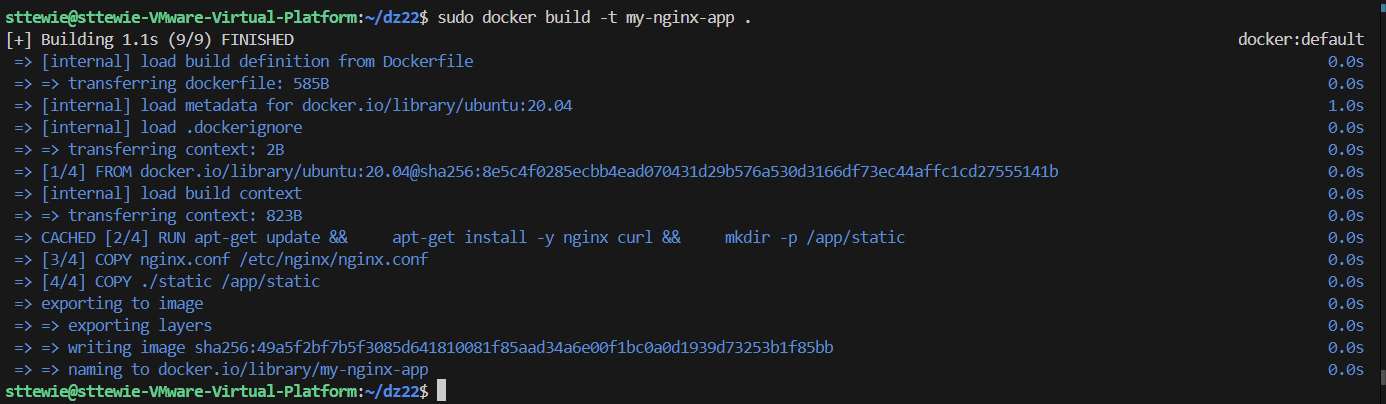




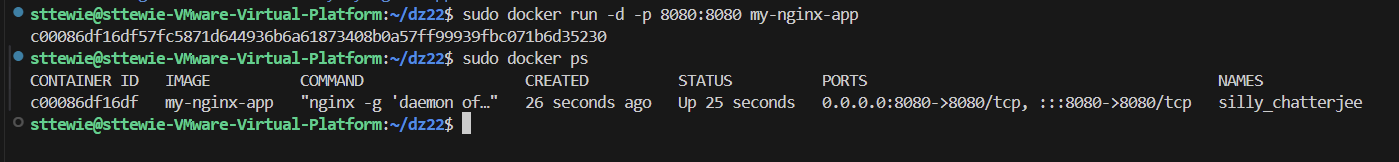
**Подготовим простой html файл**



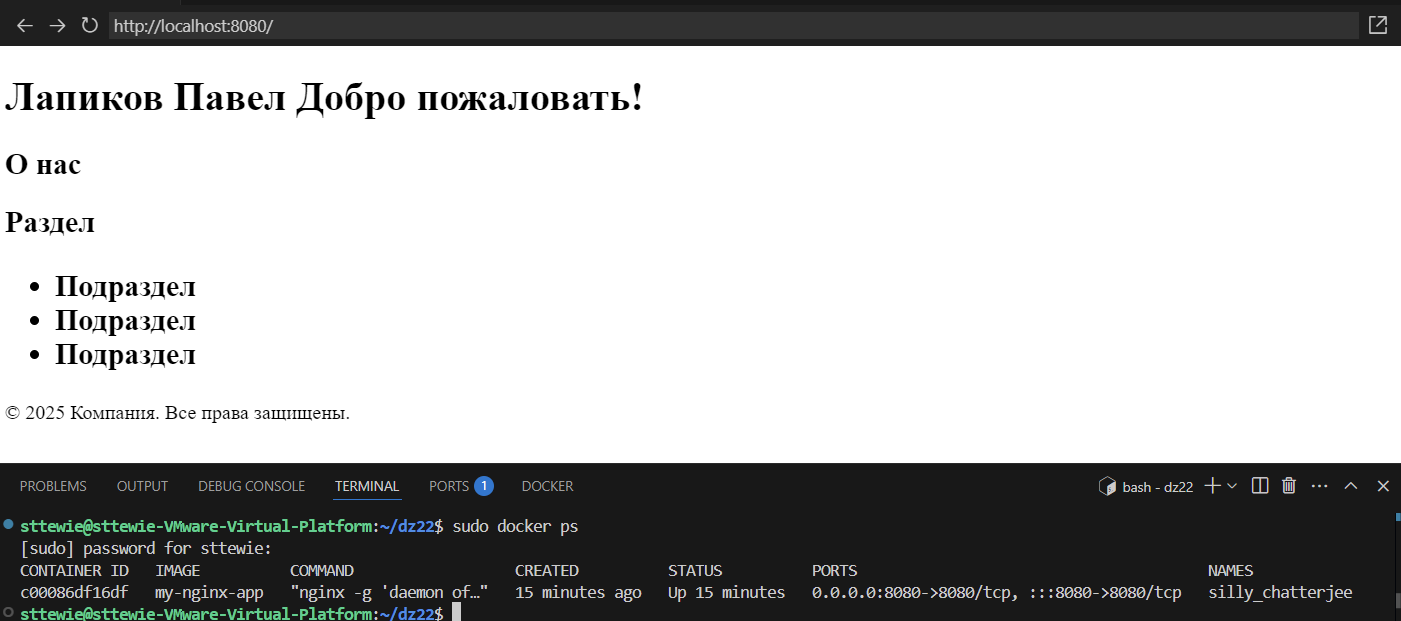
**Соберем Docker образ с нашим веб-сервером:**

****

**Запускаем контейнер:**

****

**И проверяем его работу**

****

Задание 2 – развертывание приложения с помощью Docker-compose

1. Создайте новый файл docker-compose.yml в пустой директории на вашем локальном компьютере.

2. Напишите инструкцию version в версии 3.

3. Определите сервис для базы данных PostgreSQL. Назовите его "db".

Используйте образ postgres:latest, задайте переменные окружения

POSTGRES\_USER, POSTGRES\_PASSWORD и POSTGRES\_DB для установки пользовательского имени, пароля и имени базы данных соответственно.

4. Определите сервис для веб-сервера на основе образа NGINX. Назовите его "web". Используйте образ nginx:latest. Определите порт, на котором должен работать сервер, с помощью инструкции ports. Задайте путь к файлам конфигурации NGINX внутри контейнера, используя инструкцию volumes.

5.\* Определите ссылку на сервис базы данных в сервисе веб-сервера.

Используйте инструкцию links.

6. Сохраните файл docker-compose.yml и запустите приложение с

помощью команды docker-compose up.

7. Проверьте, что приложение работает, перейдя в браузере на

localhost:80.