Домашнее задание 19.06.2025

### Задание 1. Запуск контейнеров и базовые команды

1. Запусти контейнер с именем mynginx на основе образа nginx в фоне, пробросив порт 8080 на 80.

2. Посмотри список работающих контейнеров. Останови контейнер.

3. Удали контейнер.

4. Запусти контейнер с образом alpine в интерактивном режиме с командой sh.

Выполнение:

Установка doskerdesktop, регистрация в <https://hub.docker.com/>

В doskerdesktop: Resources -> Resources WSL integration -> Enable integration with my default WSL distro и Ubuntu

1. В LWS:

DOCKER\_LOG\_LEVEL=debug docker run -it --name mynginx -p 8080:80 nginx

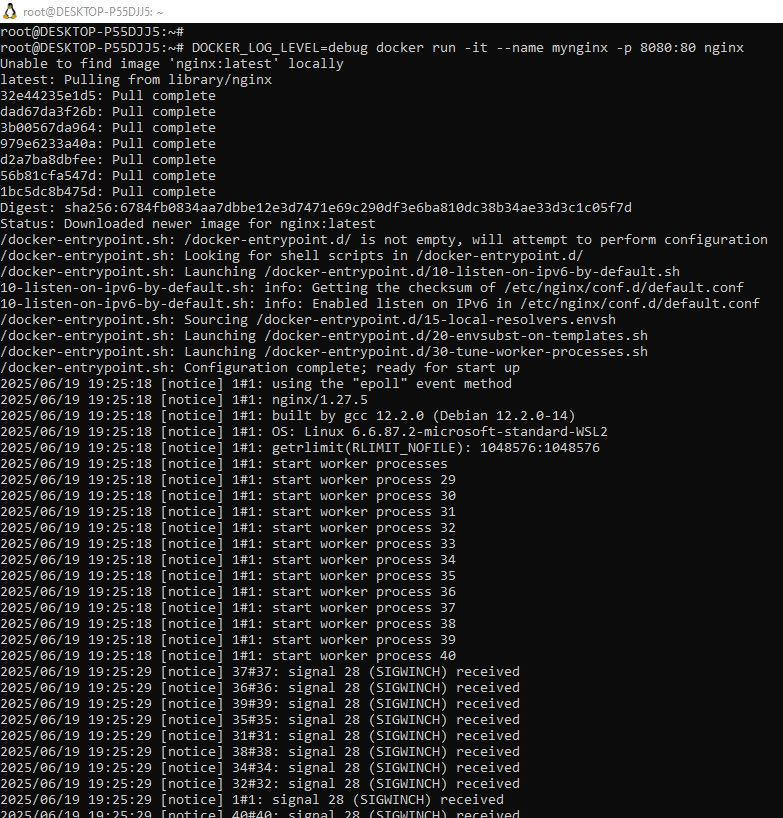
Параметры команды:

-it - запуск контейнера в интерактивном режиме.

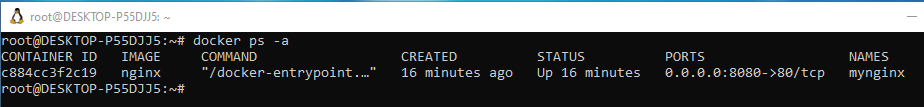
--name mynginx — задание имени контейнера.

-p 8080:80 - проброс порта: порт 8080 машины привязывается к порту 80 контейнера.

nginx - имя образа Docker.



Проверка статуса контейнера: docker ps -a

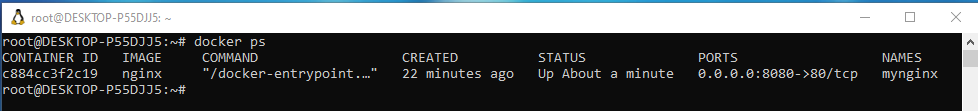


Просмотр логов контейнера: docker logs mynginx

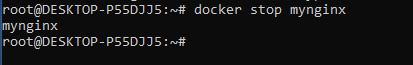
Перезапуск контейнера: docker restart mynginx



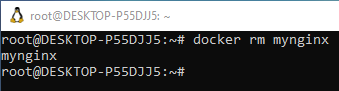
2. Список работающих контейнеров: **docker ps**



Остановка контейнера: **docker stop mynginx**



3. Удаление контейнера: **docker rm mynginx**



4. Запуск контейнера с образом alpine в интерактивном режиме с командой sh:

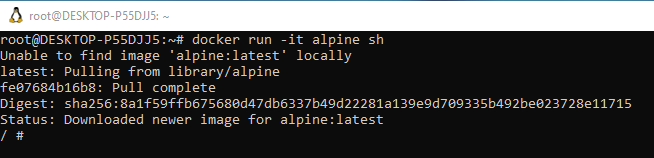
**docker run -it alpine sh**

Параметры команды:

-it - интерактивный режим (interactive terminal).

alpine - имя образа Docker.

sh - запуск командной оболочки Alpine Linux.



### Задание 2. Работа с образами

1. Посмотри список всех образов на системе.

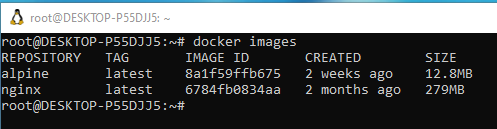
2.Удали образ nginx.

3. Скачай образ ubuntu:22.04 без запуска.

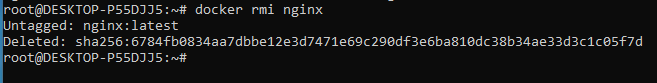
4. Проверь, скачался ли образ.

Выполнение:

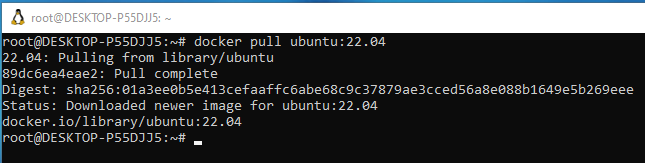
1. Список всех образов на системе: **docker images**



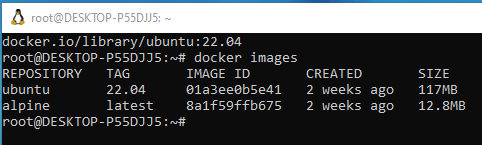
2. Удаление образа nginx : **docker rmi nginx**



3. Скачивание образа ubuntu:22.04 без запуска: **docker pull ubuntu:22.04**



4. Проверка, скачался ли образ: **docker images**



### Задание 3. Использование ARG и ENV

1. Добавь в Dockerfile переменную сборки ARG VERSION.

2. Установи ENV переменную APP\_VERSION равной значению ARG.

3. Добавь RUN, который выводит значение APP\_VERSION.

4. При сборке передай VERSION=2.5.1 через –build-arg.

5. Запусти контейнер

Выполнение:

Создание Dockerfile:

1. Добавление в Dockerfile переменной сборки ARG VERSION:

2. Установка в Dockerfile ENV переменной APP\_VERSION равной значению ARG:

ENV APP\_VERSION="${VERSION}"

3. Добавление в Dockerfile RUN, который выводит значение APP\_VERSION:

RUN echo "App version: $APP\_VERSION"

FROM alpine:latest

# Добавляем переменную сборки ARG VERSION

ARG VERSION

# Устанавливаем переменную APP\_VERSION равную значению VERSION

ENV APP\_VERSION="${VERSION}"

# Выводим значение переменной APP\_VERSION

RUN echo "App version: $APP\_VERSION"

# Копируем исходники приложения (если есть)

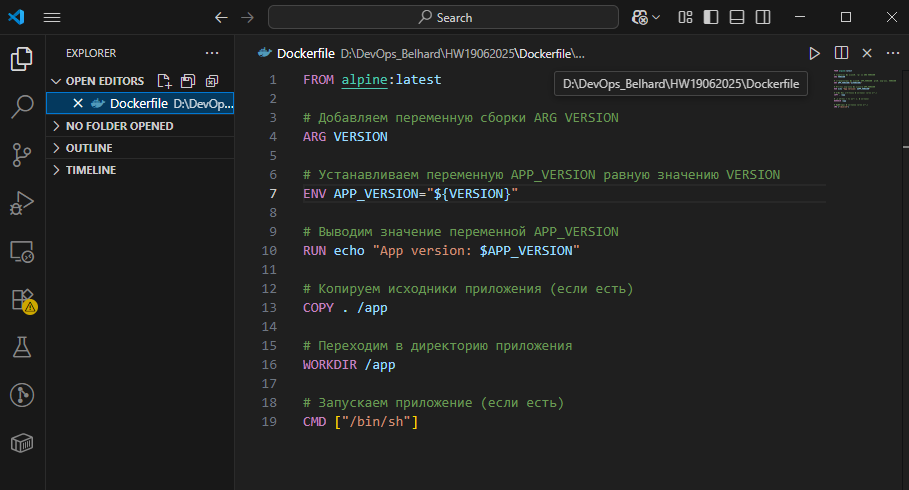
COPY . /app

# Переходим в директорию приложения

WORKDIR /app

# Запускаем приложение (если есть)

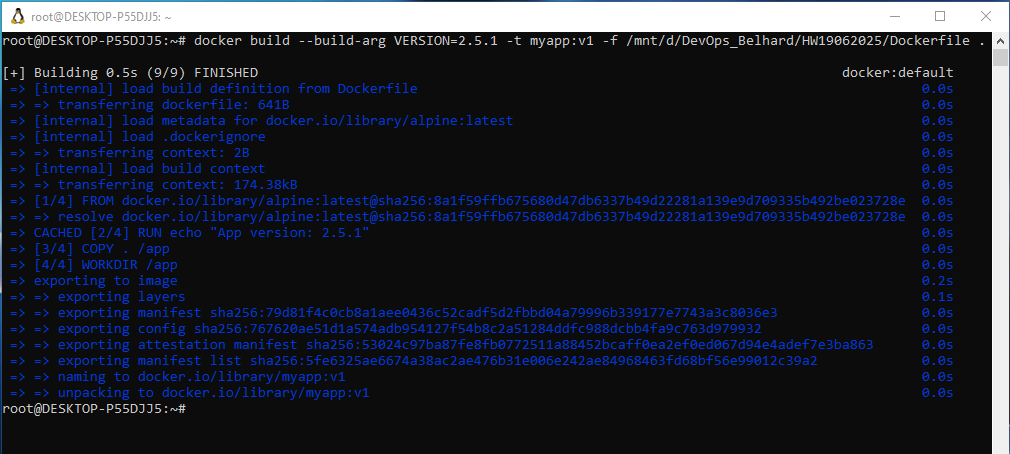
CMD ["/bin/sh"]

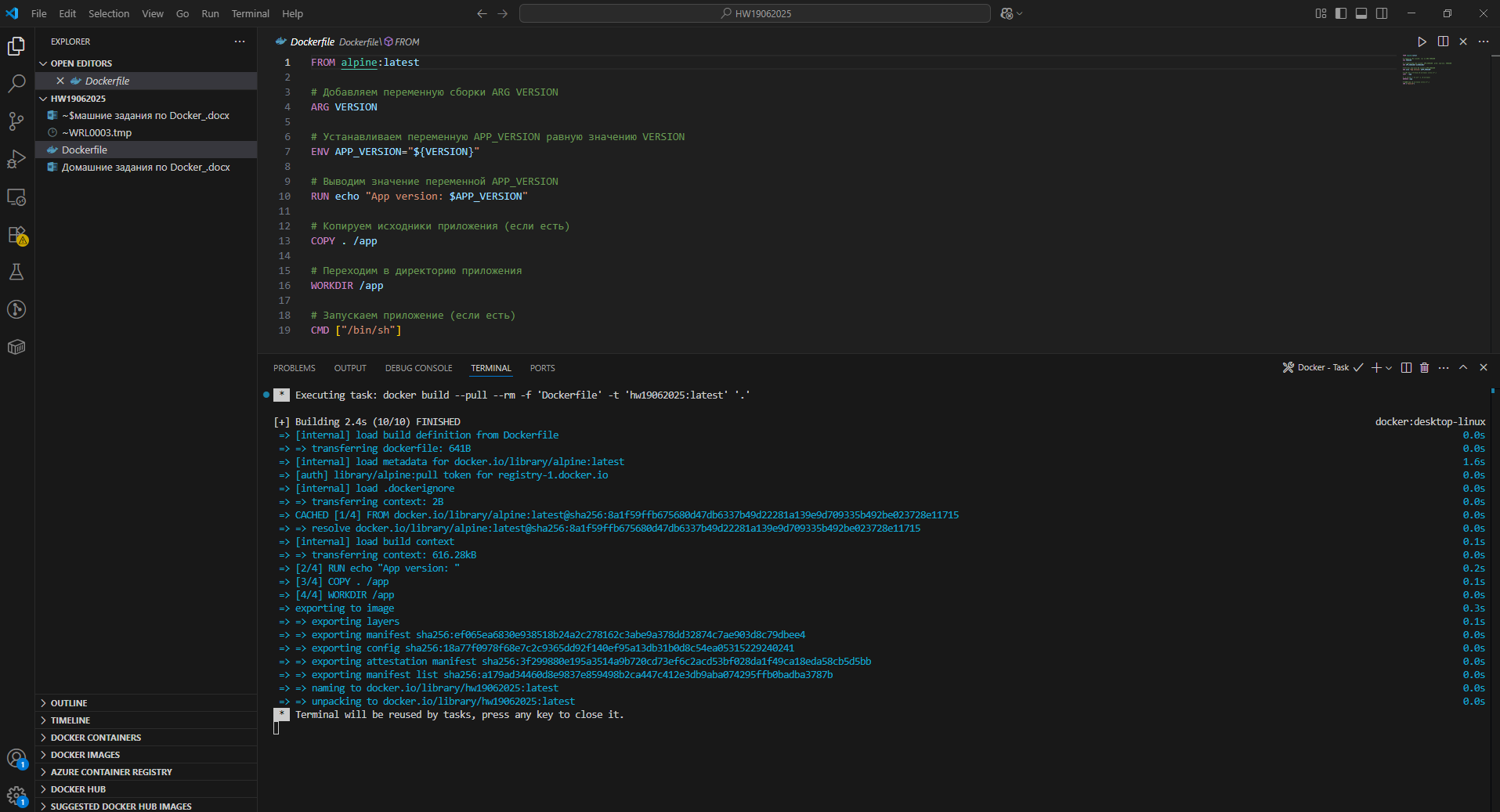


4. При сборке передача VERSION=2.5.1 через –build-arg:

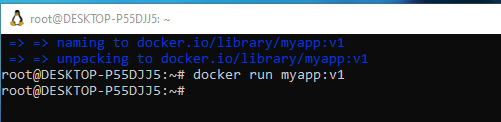
В LWS:

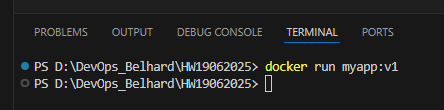
**docker build --build-arg VERSION=2.5.1 -t myapp:v1 -f /mnt/d/DevOps\_Belhard/HW19062025/Dockerfile .**





5. Запуск контейнера: docker run myapp:v1





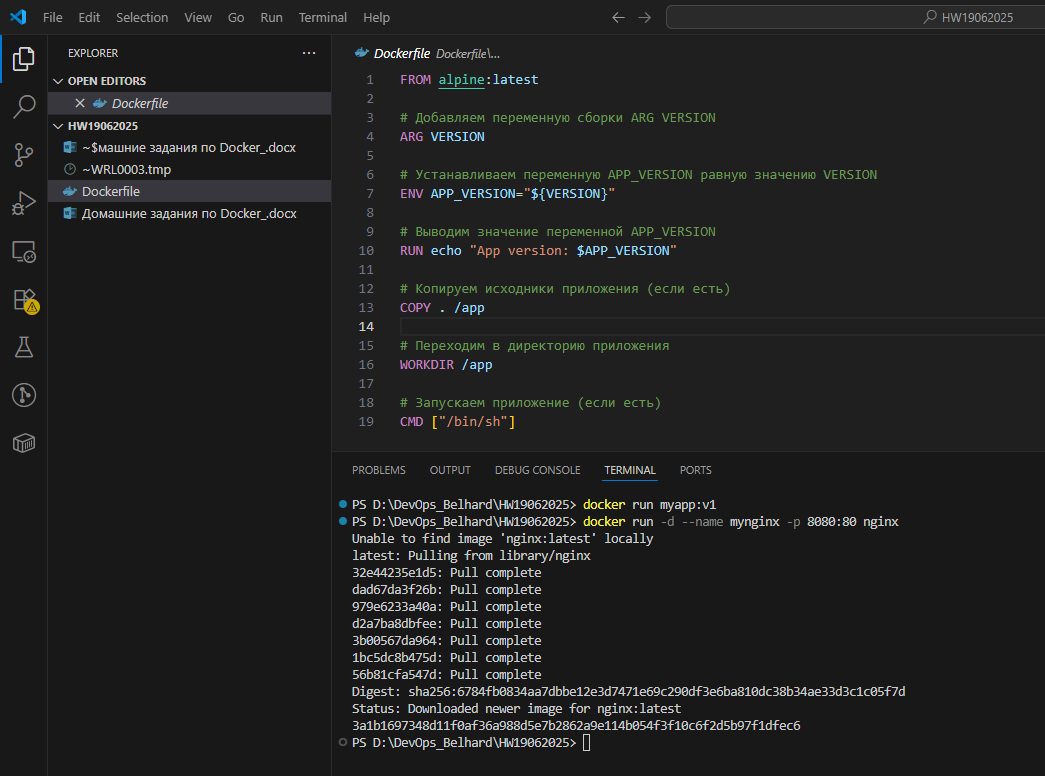
### Задание 4. Проброс портов

1. Запусти контейнер с nginx, пробросив порт 8080.

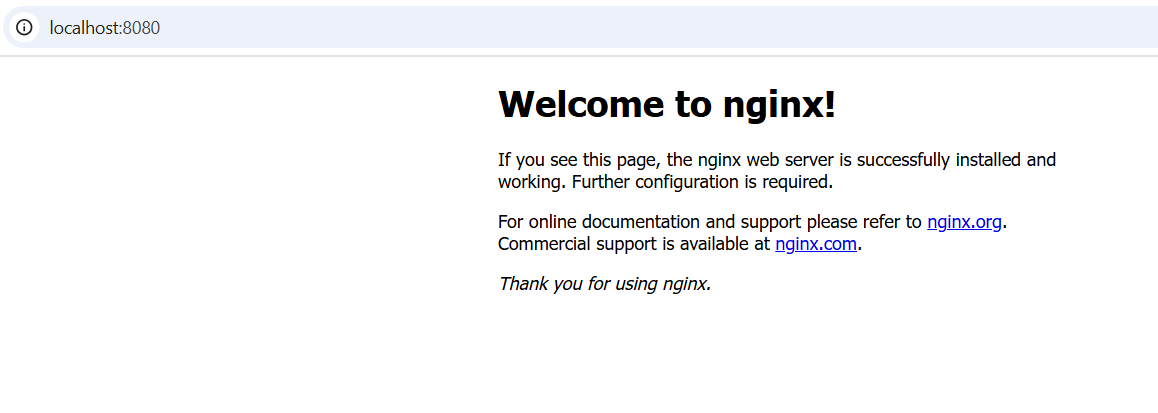
2. Перейди в браузере на http://localhost:8080 и убедись, что сервер отвечает.

Выполнение:

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



2. <http://localhost:8080>



### Задание 5. Подключение к контейнеру и выполнение команд

1. Запусти контейнер на базе ubuntu

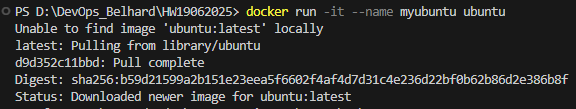
2. Подключись внутрь через docker exec -it

3. Создай файл /tmp/hello.txt внутри контейнера

4. Выйди, скопируй файл из контейнера на хост

Выполнение:

1. Запуск контейнера на базе ubuntu: docker run -it --name myubuntu ubuntu



2. Подключение внутрь через docker exec -it : docker exec -it myubuntu bash

3. Создание файла /tmp/hello.txt внутри контейнера : echo "Hello, World!" > /tmp/hello.txt

4. Выход, копирование файла из контейнера на хост: exit

docker cp myubuntu:/tmp/hello.txt /tmp

Параметры команды:

myubuntu:/tmp/hello.txt — путь к файлу внутри контейнера.

/tmp — путь на хост-компьютере, куда копируется файл.

