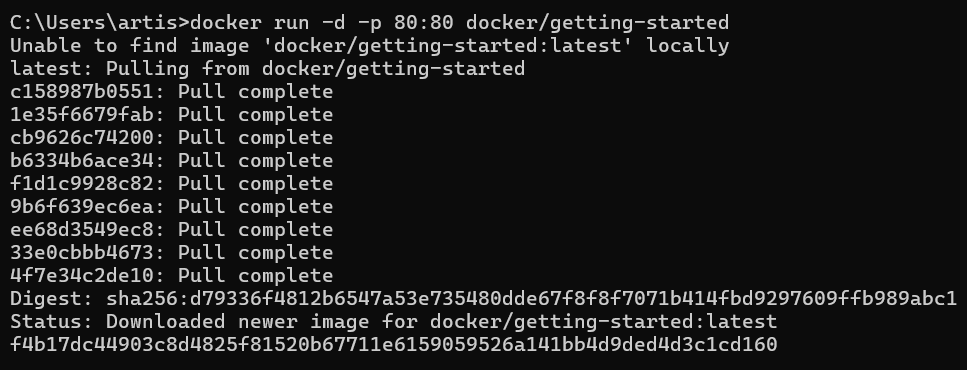
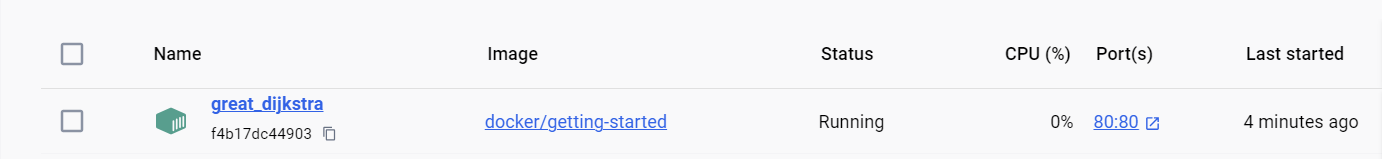
Лабораторная работа №2

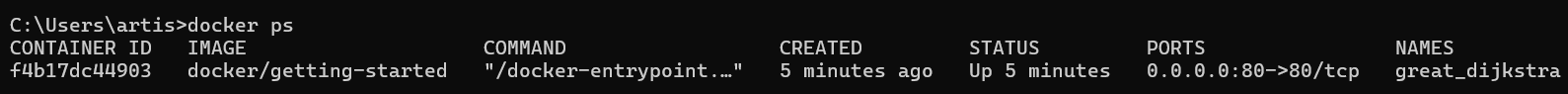
**Тема**: Работа с Docker.

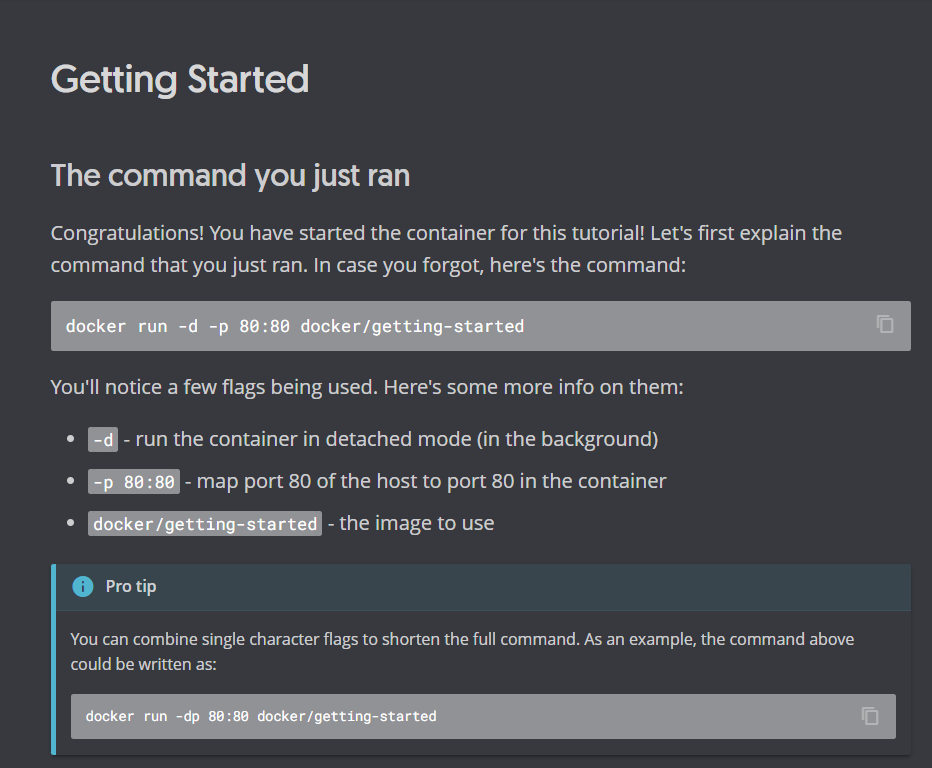
**Цель**: Познакомиться с возможностями и получить практические навыки работы с Docker.

1. Изучить простейшие консольные команды и возможности Docker Desktop, создать собственный контейнер docker/getting-started, открыть в браузере и изучить tutorial

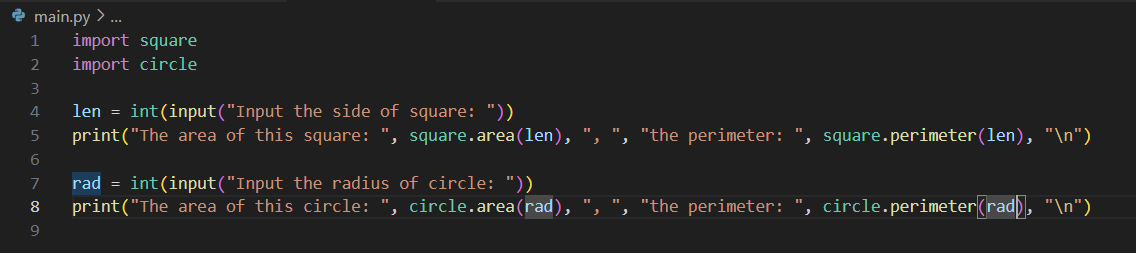
**

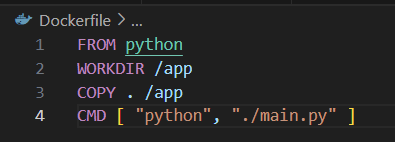
**

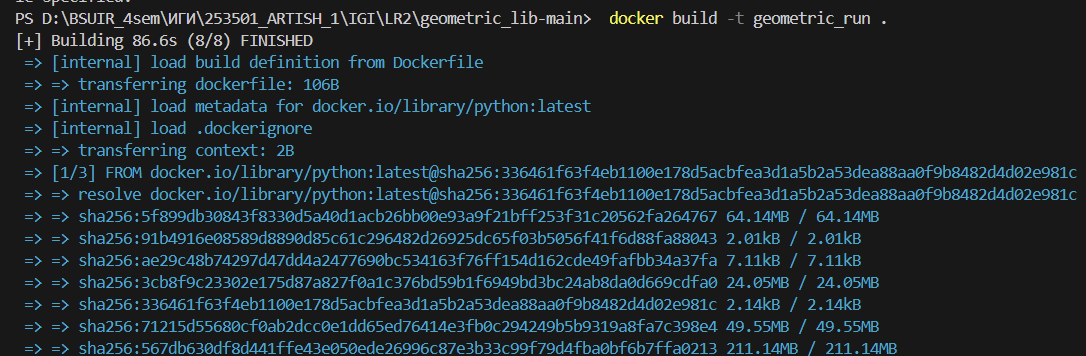
**

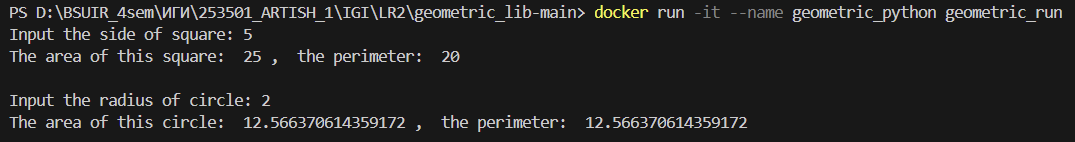
**

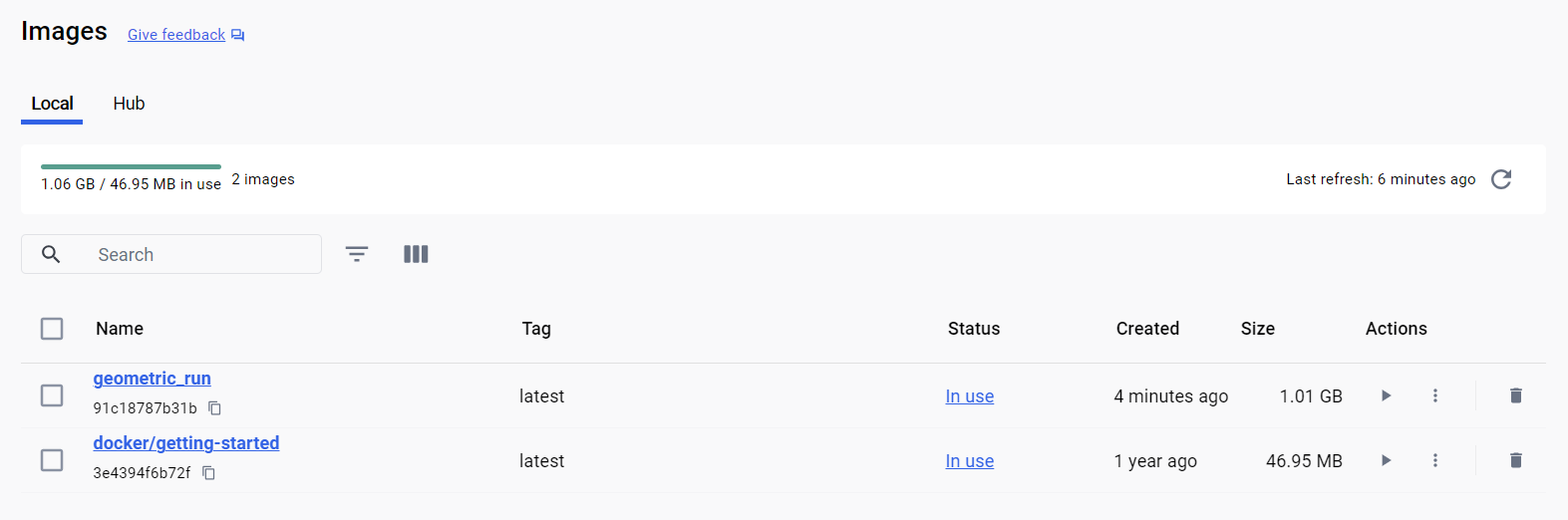
1. Создайте docker image, который запускает скрипт с использованием функций из <https://github.com/smartiqaorg/geometric_lib>.

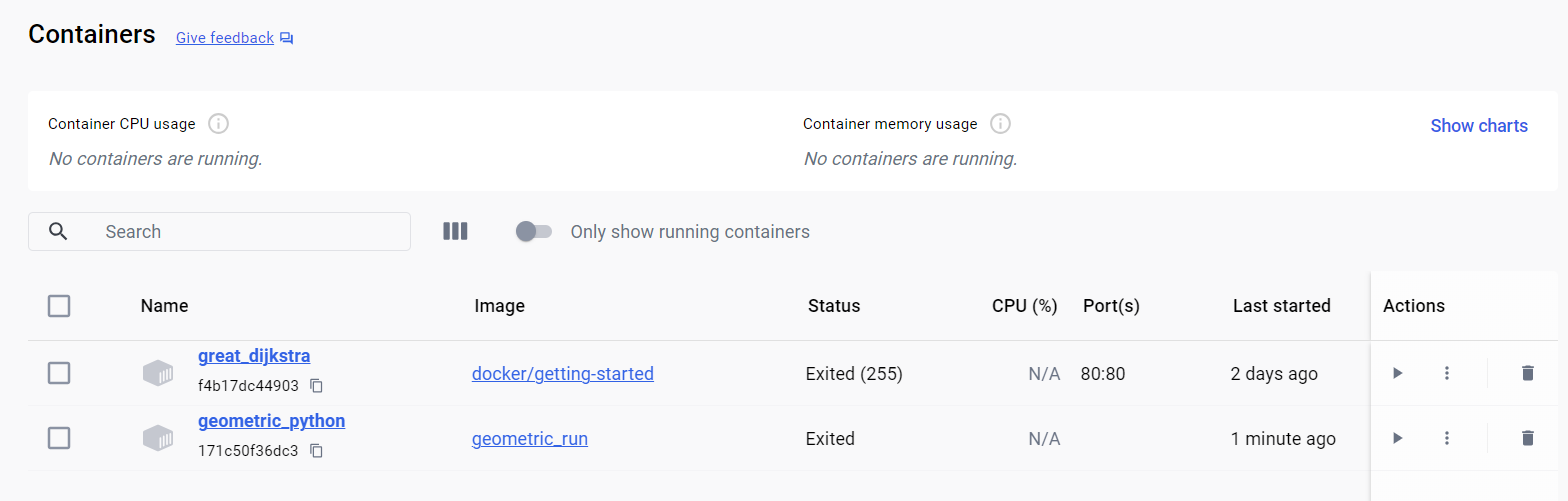
**

**

**

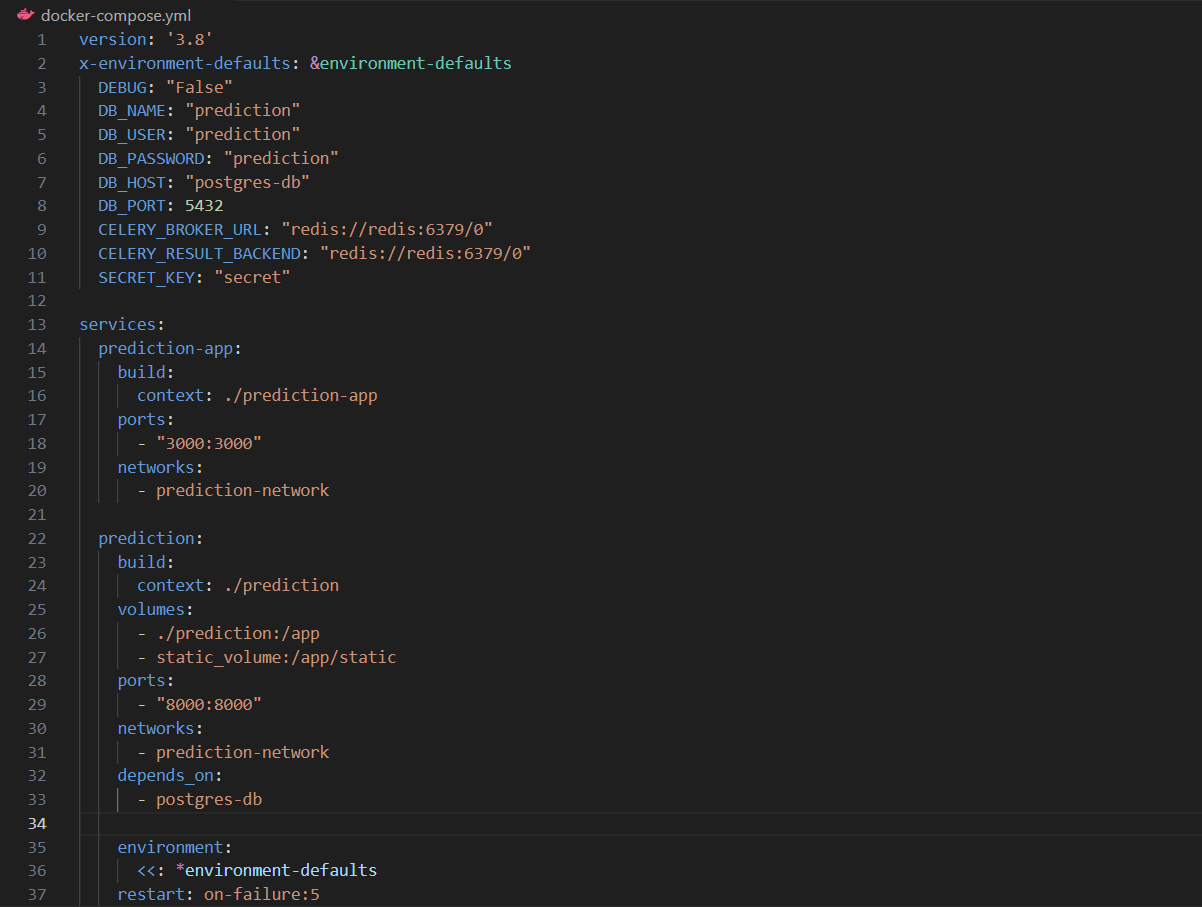
**

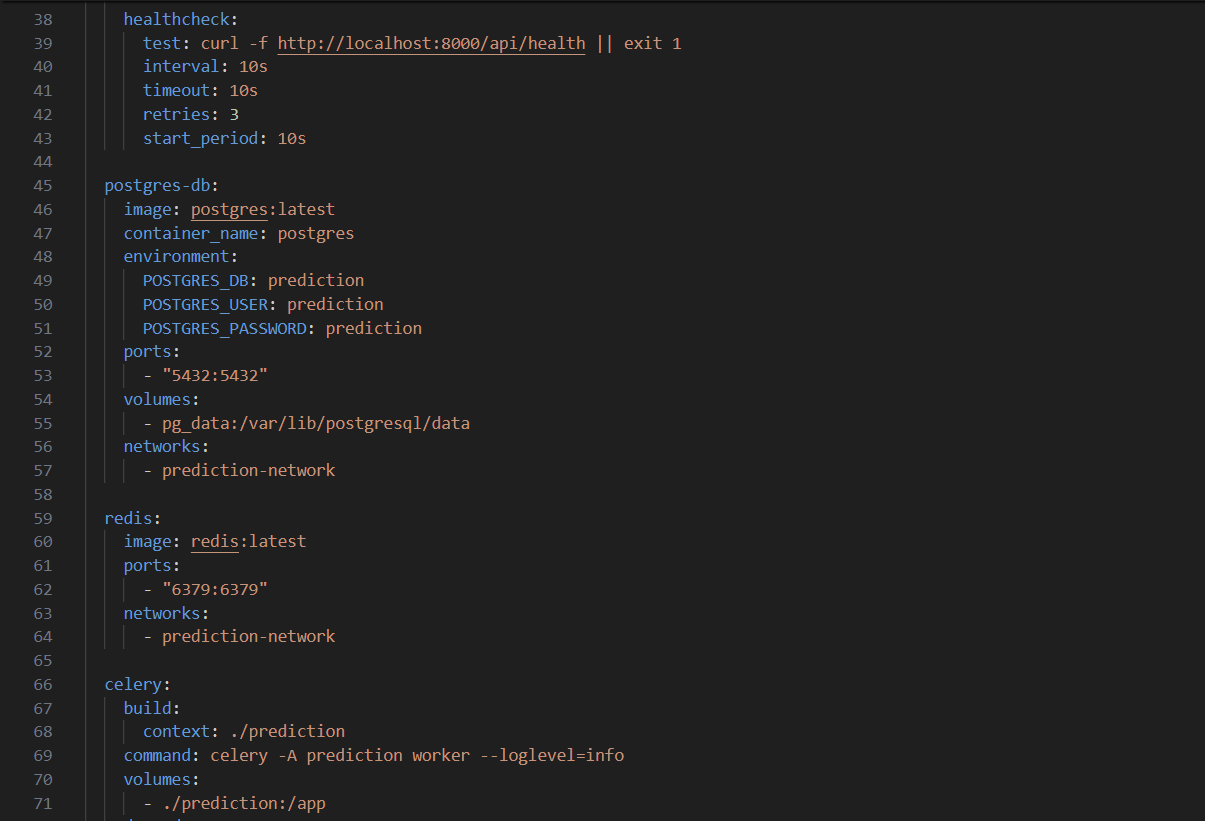
**

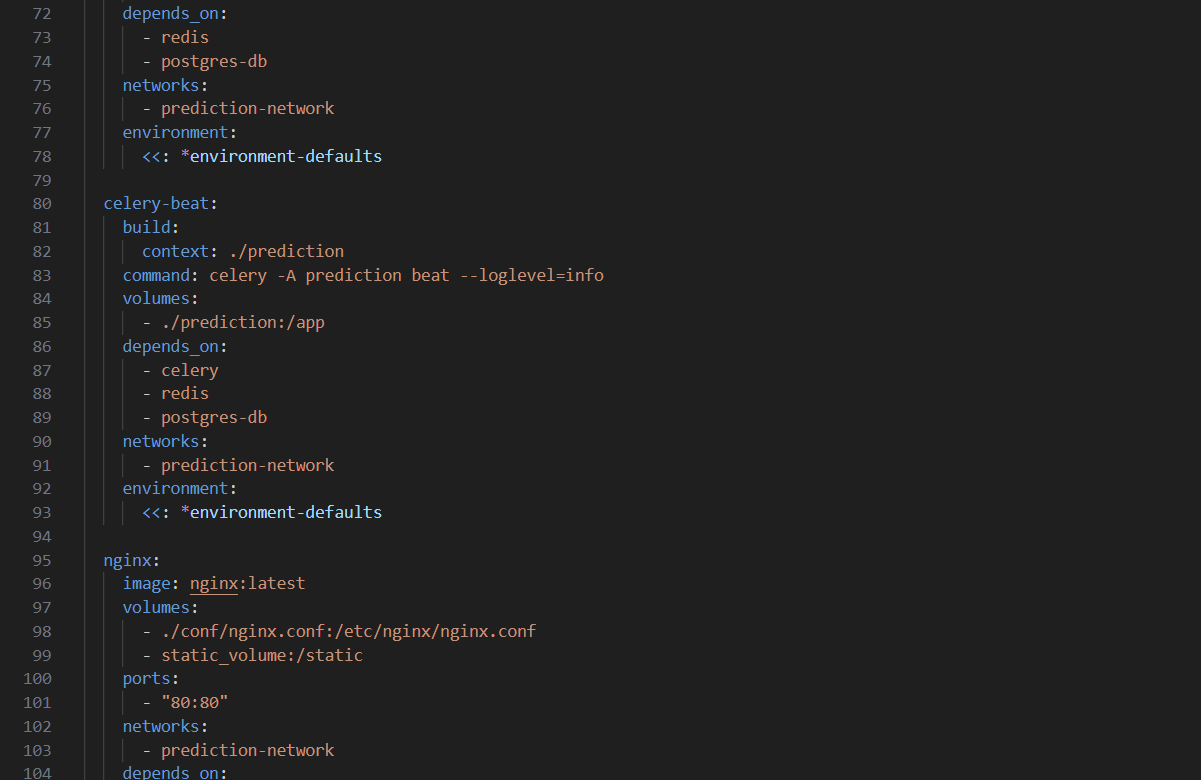
**

1. Скачать любой доступный проект с GitHub с произвольным стеком технологий или использовать свой, ранее разработанный. Создать для него необходимый контейнер, используя Docker Compose для управления многоконтейнерными приложениями. Запустить проект в контейнере.( Примеры Images: <https://hub.docker.com/_/phpmyadmin>, <https://hub.docker.com/_/mysql>, https://hub.docker.com/\_/postgres)
2. Настроить сети и тома для обеспечения связи между контейнерами и сохранения данных (исходные данные, логин, пароль и т.д.)

Файл compose.yaml

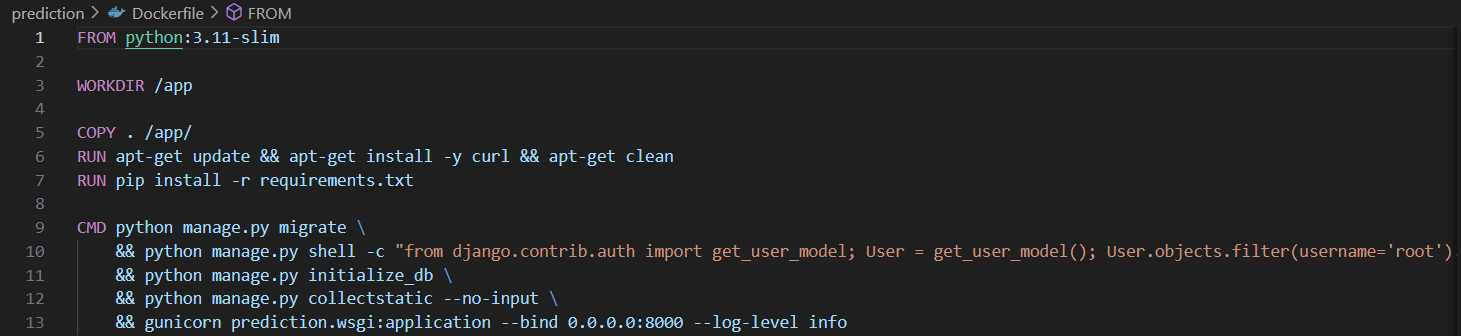




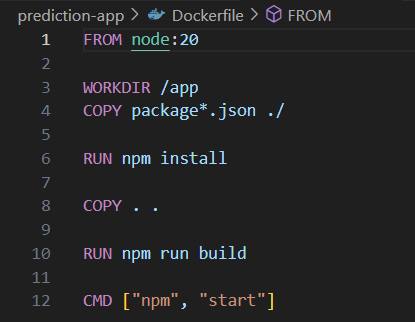




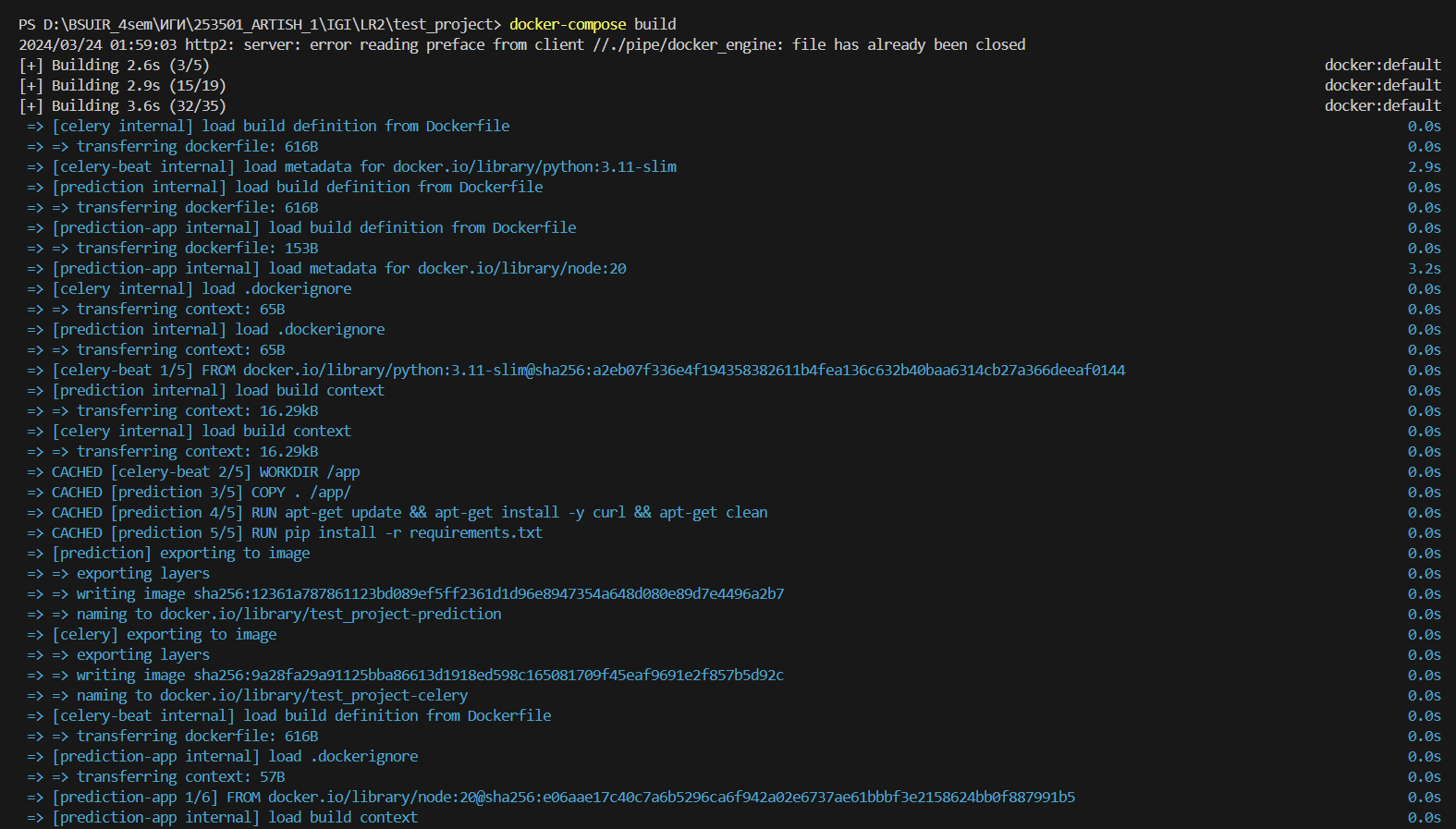
Backend Dockerfile

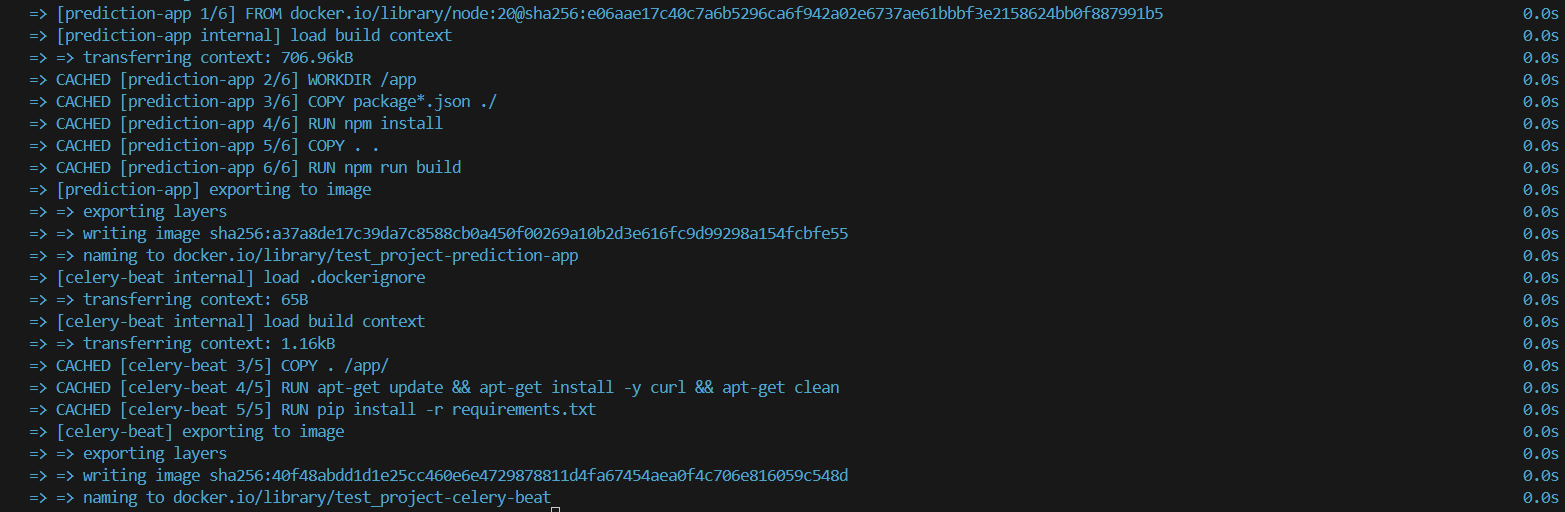


Frontend Dockerfile

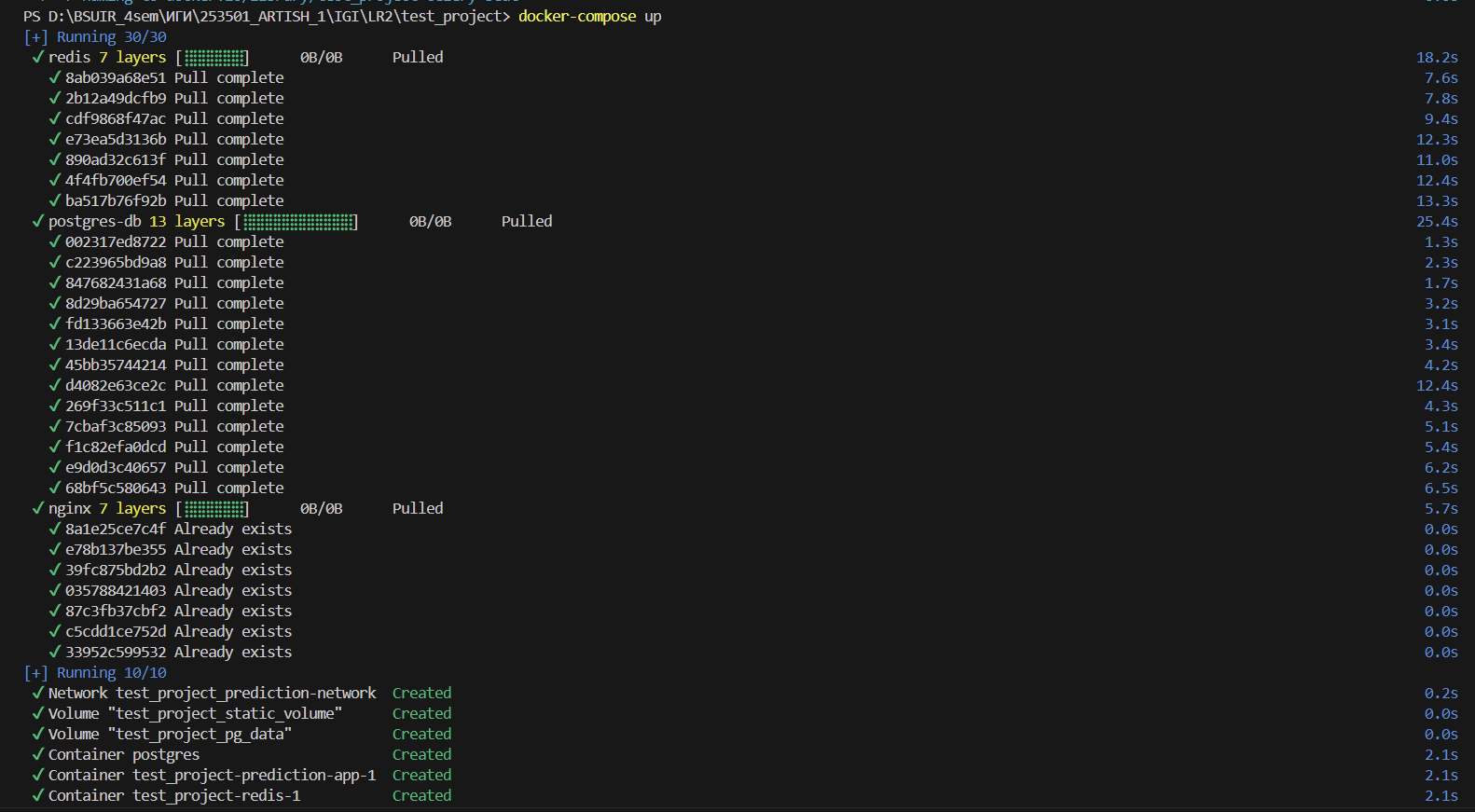


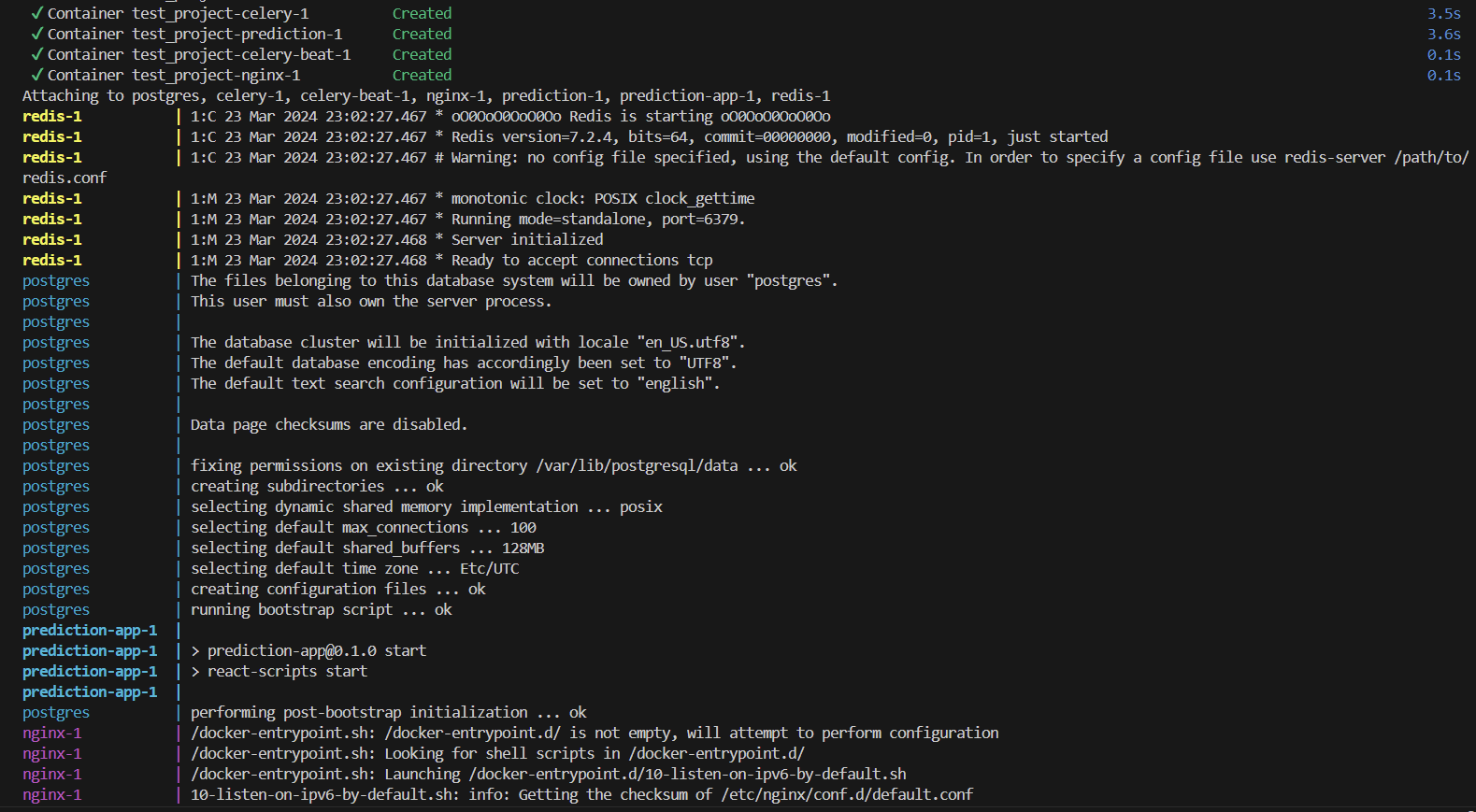
Docker-compose build (сборка проекта)



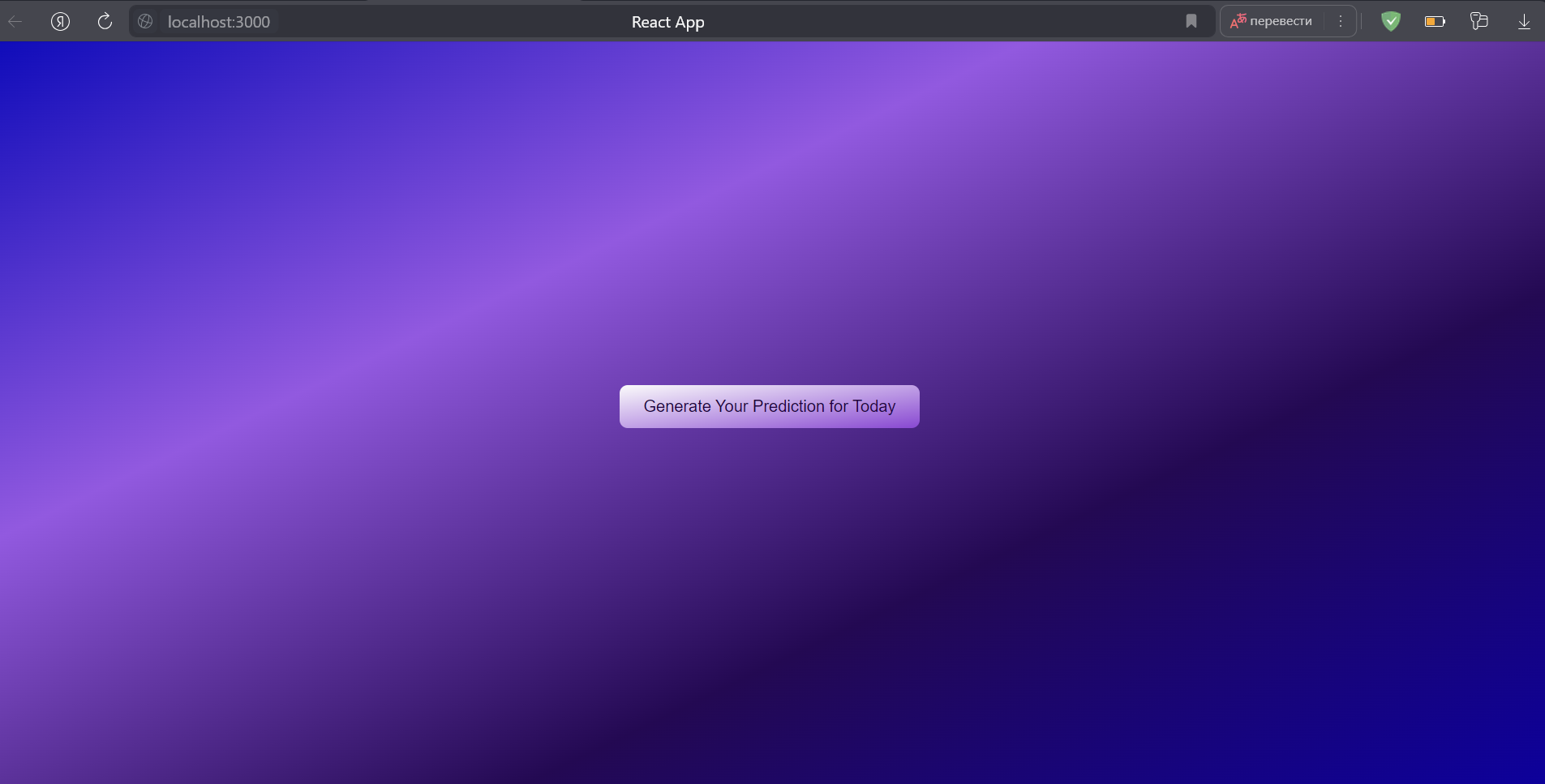


Docker-compose up

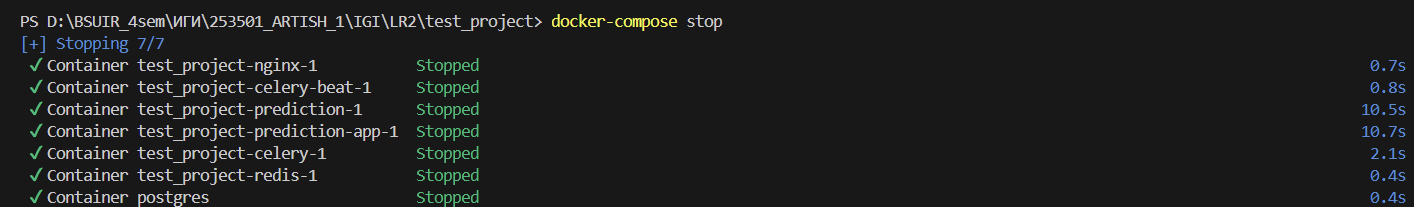




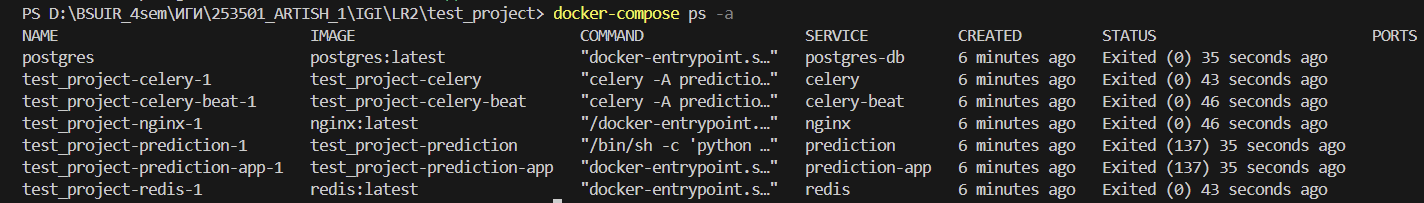
Переходим на http://localhost:3000/



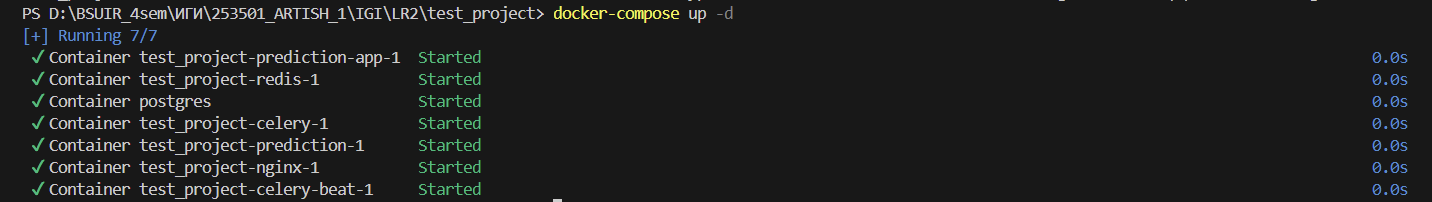
Docker-compose stop

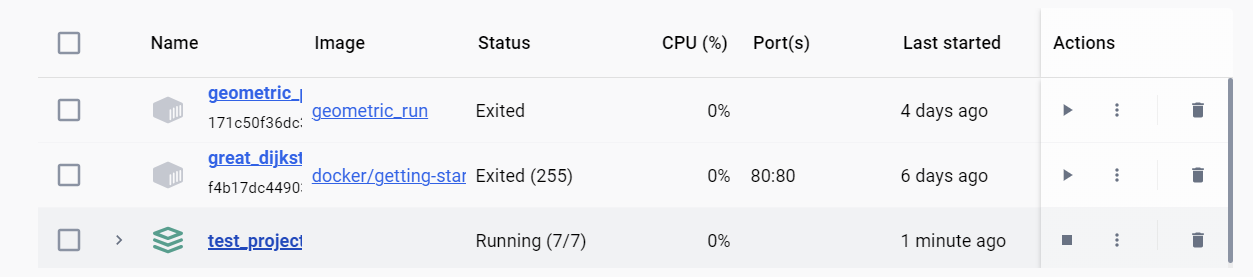


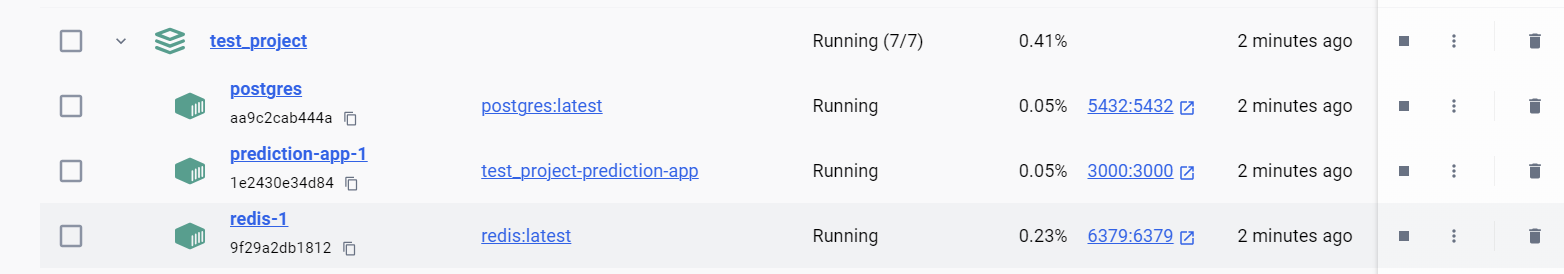
Docker-compose ps -a



Docker-compose up -d (запуск контейнеров фоновом режиме)

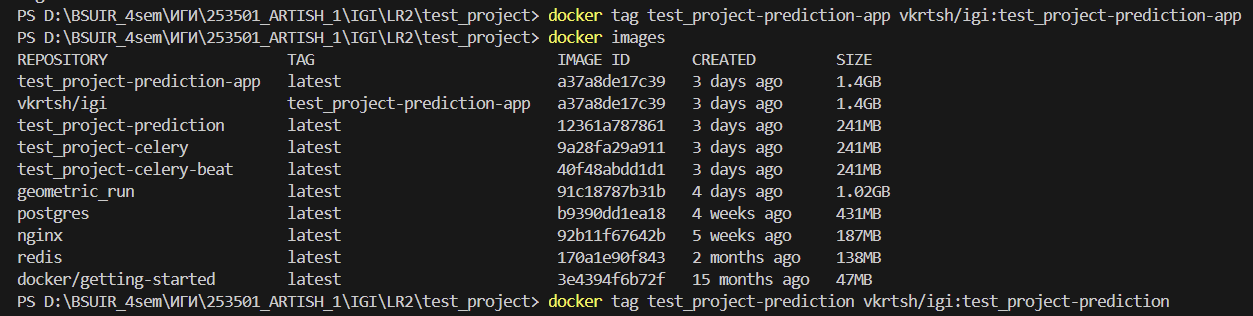


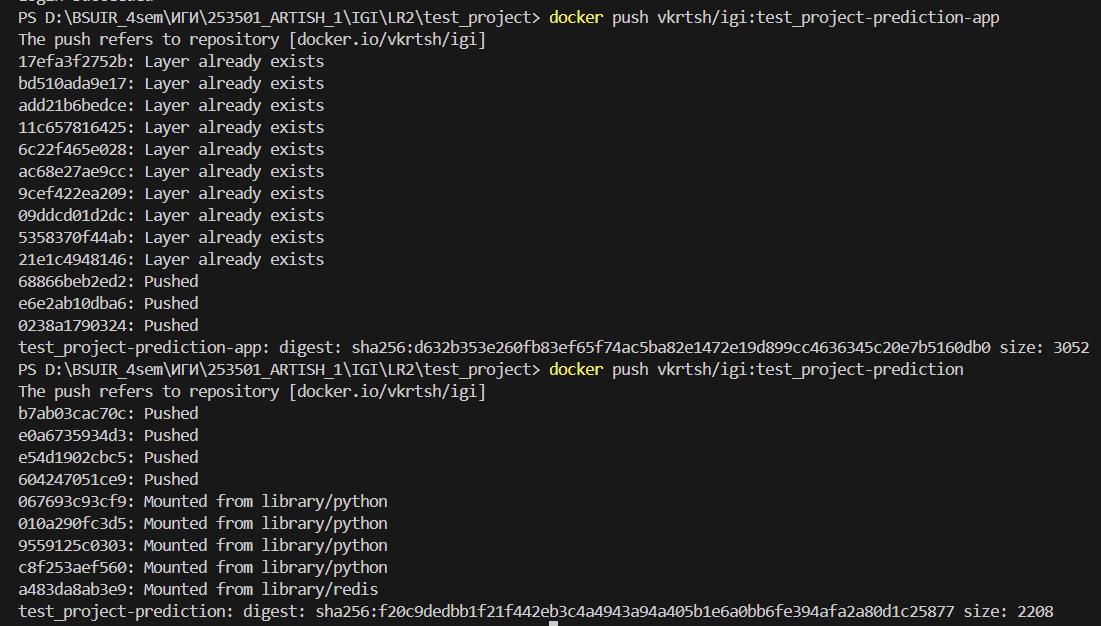


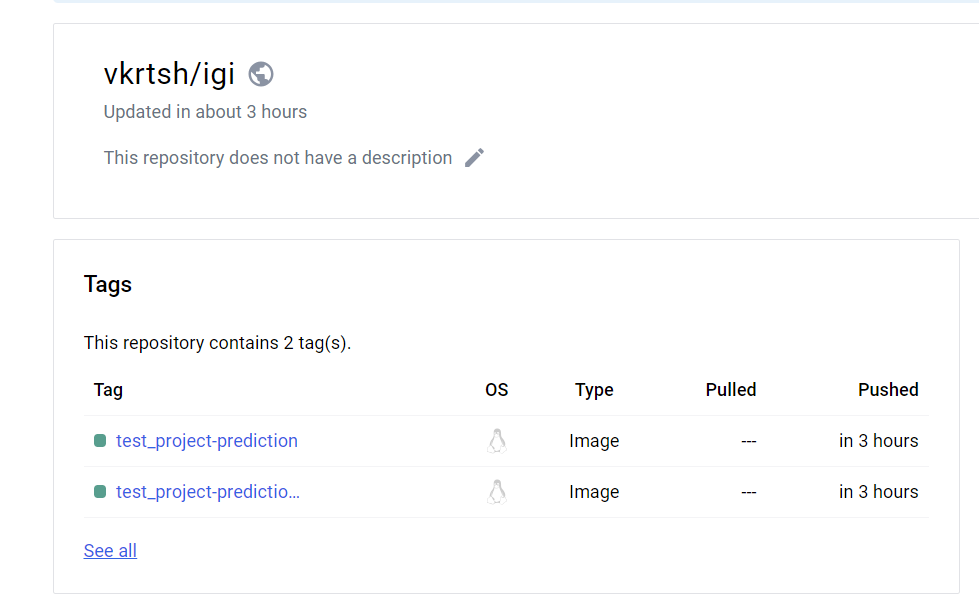




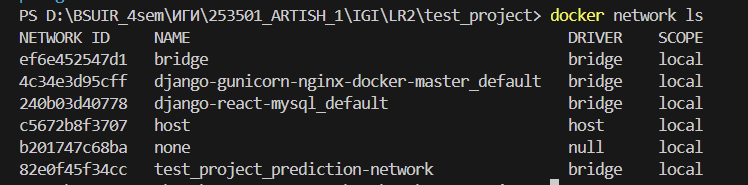
1. Разместим результат в созданный репозиторий в DockerHub



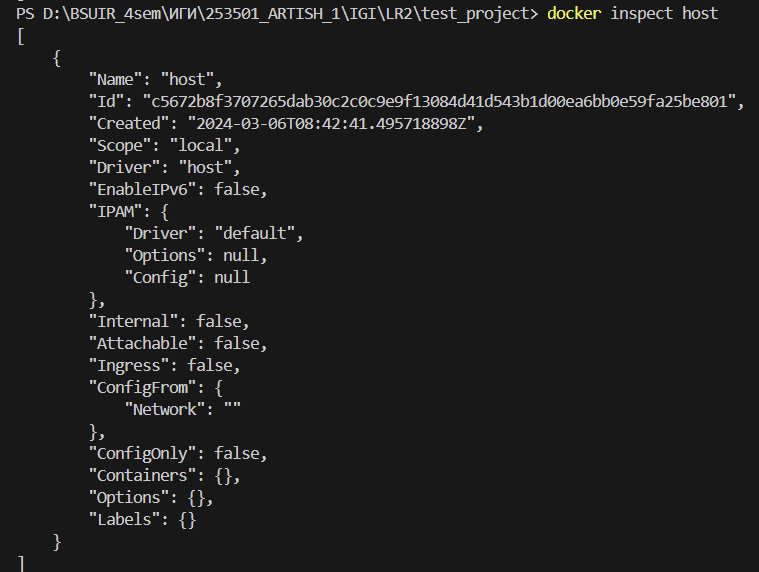
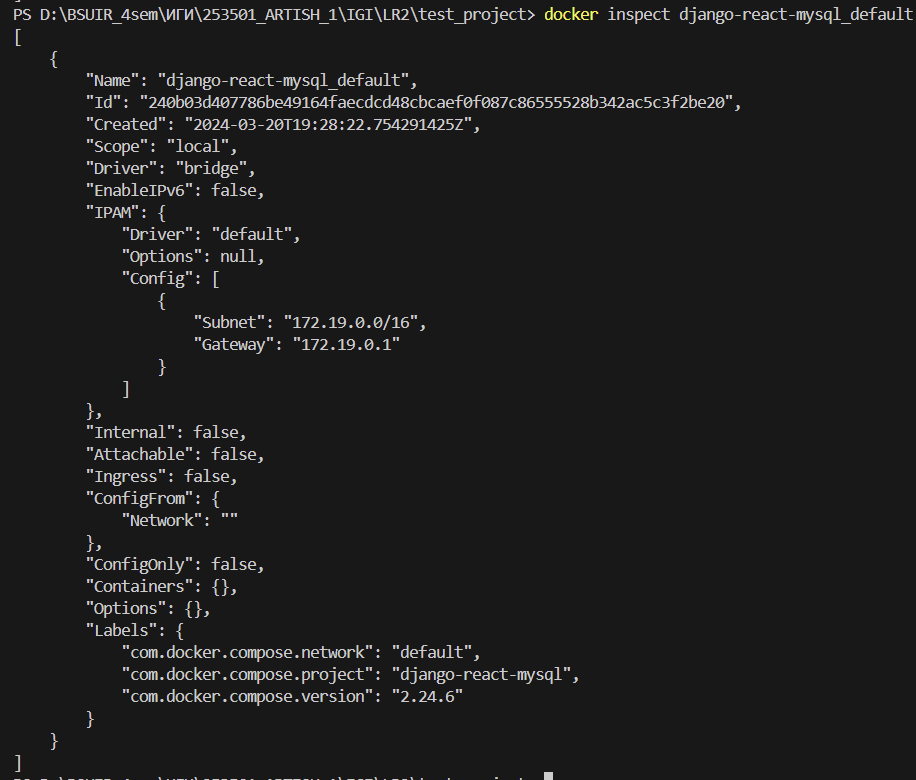
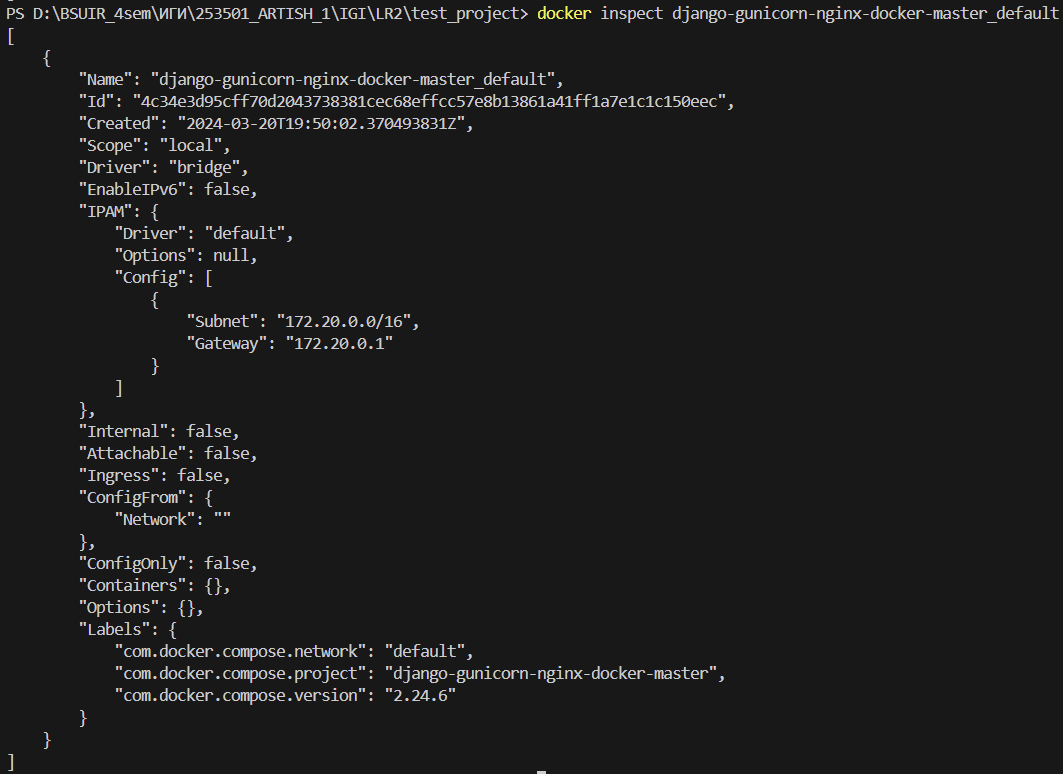


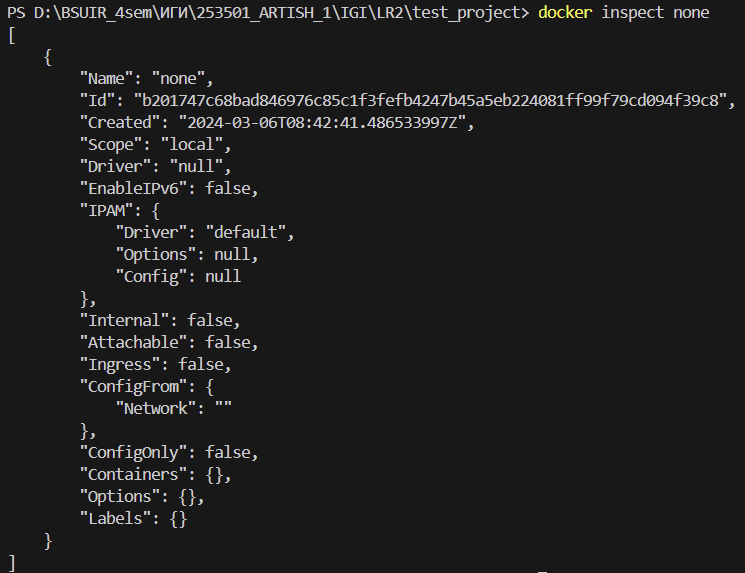


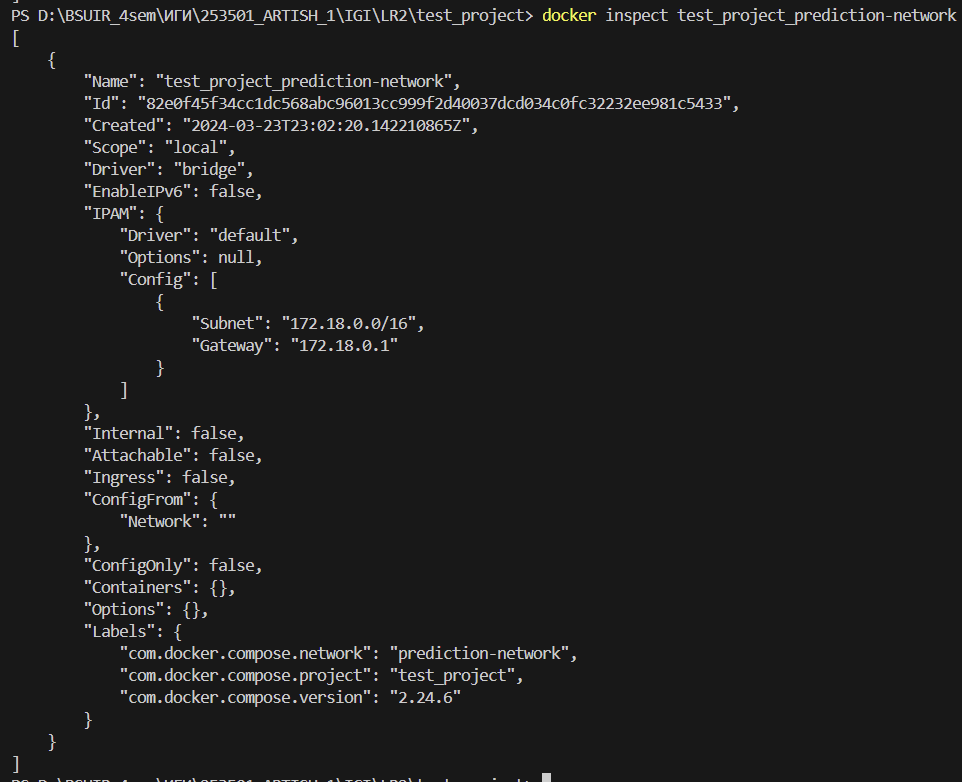
7.1. Получить информацию о всех сетях, работающих на текущем хосте и подробности о каждом типе сети











7.2. Создать свою собственную сеть bridge, проверить, создана ли она,

запустить Docker-контейнер в созданной сети, вывести о ней всю

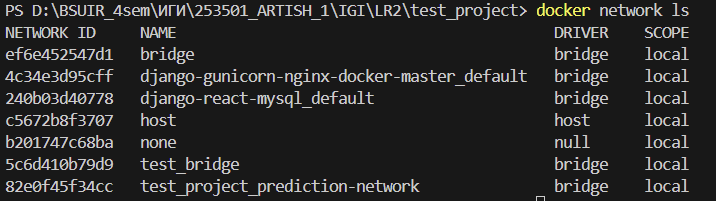
информацию(включая IP-адрес контейнера), отключить сеть от

контейнера

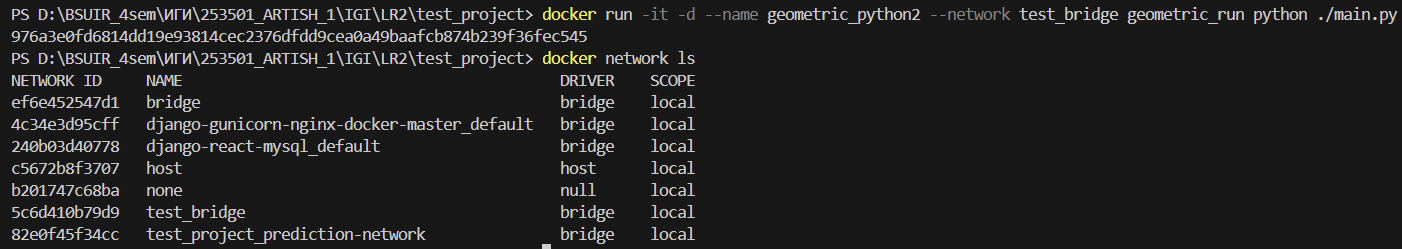
Создание сети bridge



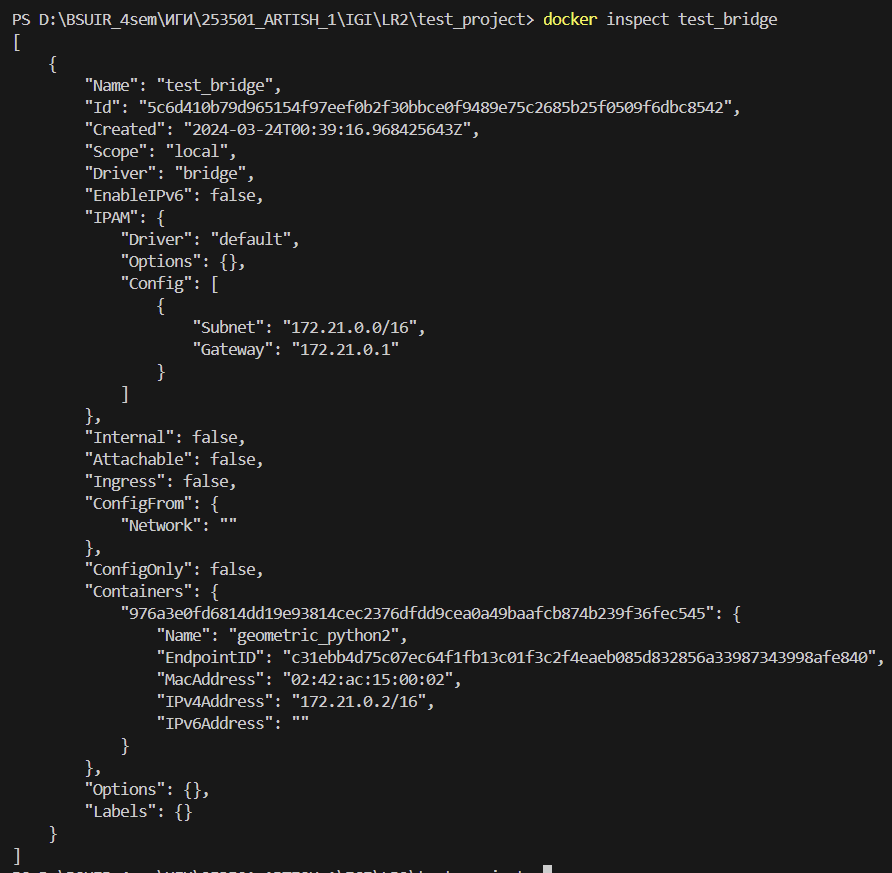
Проверка создания



Запуск контейнера в созданной сети



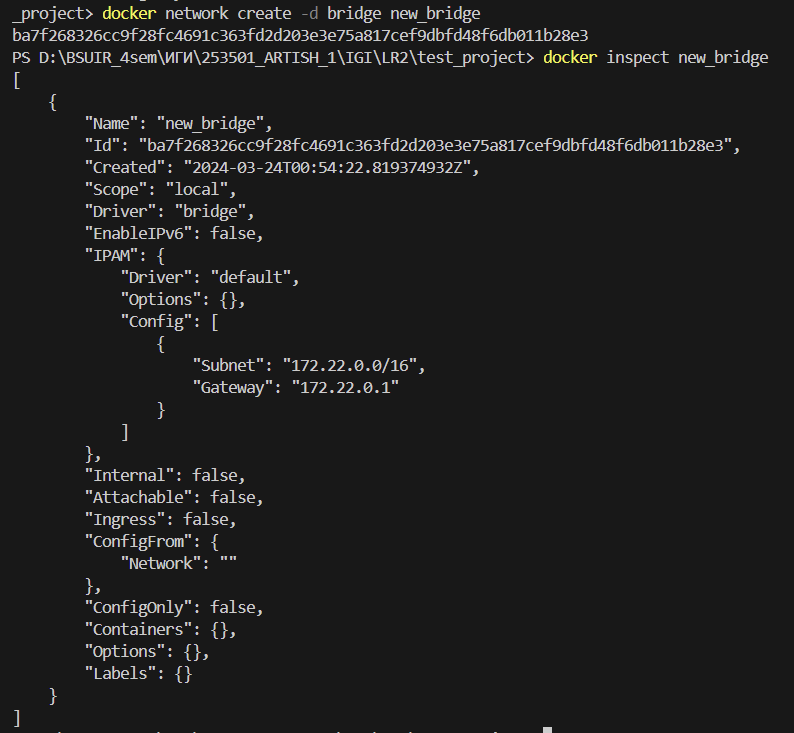
Свойства сети



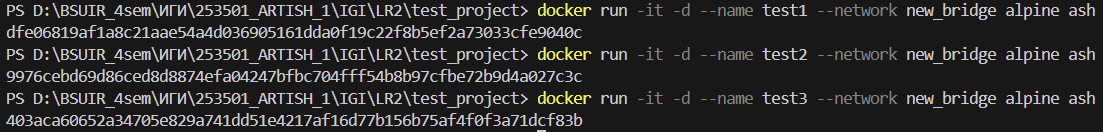
Отключение сети от контейнера



7.3. Создание еще одной сети bridge, вывод информации о ней

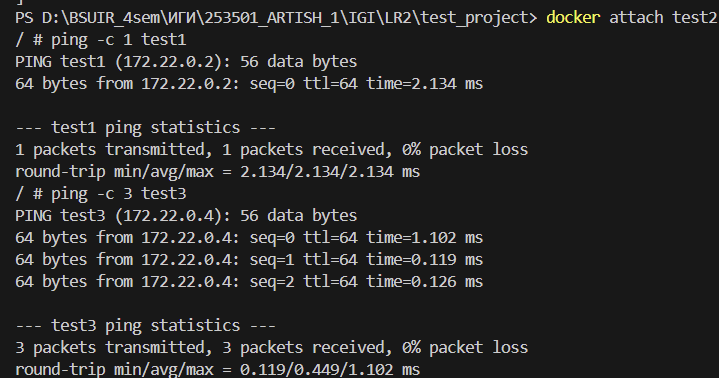


Запуск трех контейнеров





Подключение к любому из контейнеров и пинг два других из оболочки контейнера (по имени контейнера), общение по IP-адресу

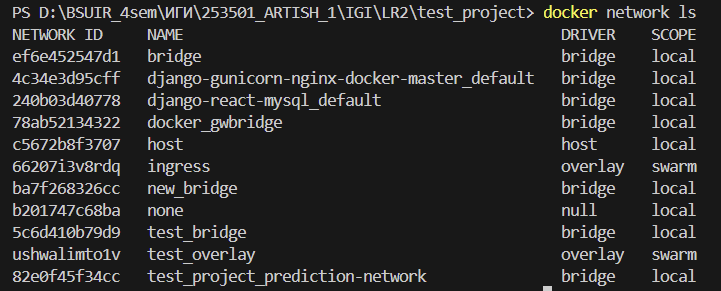
****

7.4. Создать свою собственную сеть overlay, проверить, создана ли она, вывести о ней всю информацию

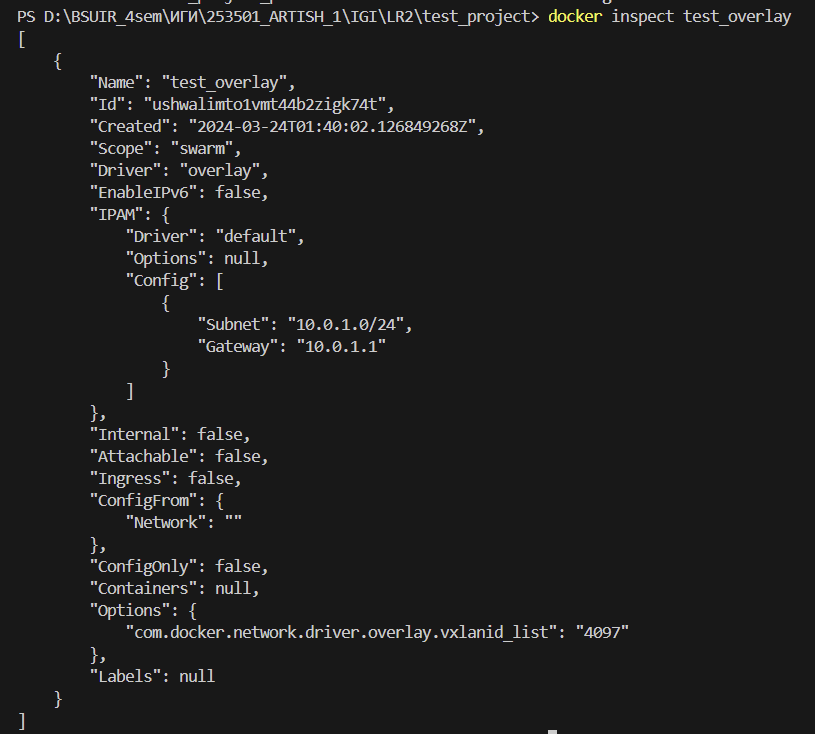
Создание сети

****

Проверка, создана ли сеть

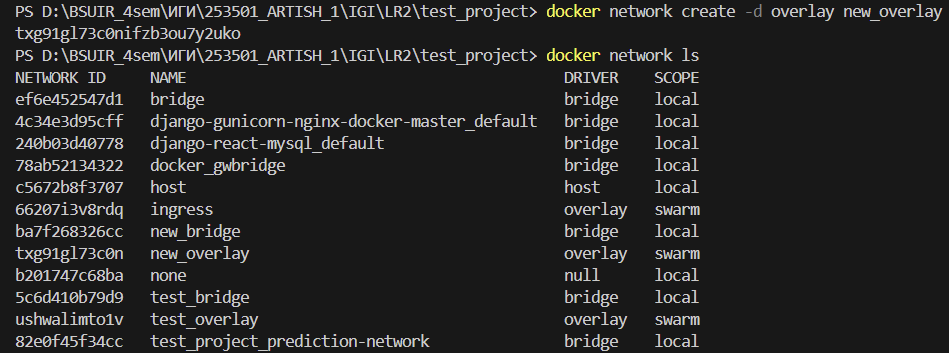
****

Вывод информации о сети

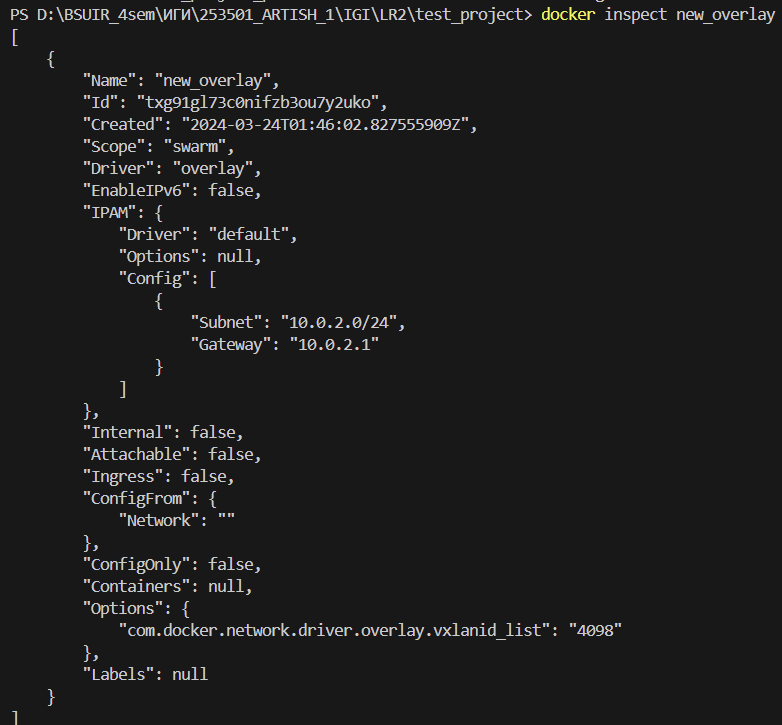
****

7.5. Создать еще одну сеть overlay, проверить, создана ли она, вывести о ней всю информацию, удалить сеть

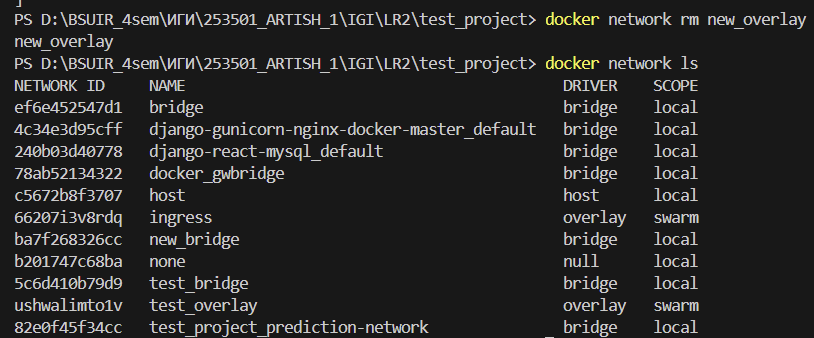
Создание сети и проверка

****

Вывод информации о сети

****

Удаление сети

****

7.6. Попробовать создать сеть host, сохранить результат в отчет



Разрешен только один экземпляр хостовой сети.