Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет  
 информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Избранные главы информатики

ОТЧЕТ

к лабораторной работе

на тему

Работа с Docker

БГУИР КП 1-40 04 01

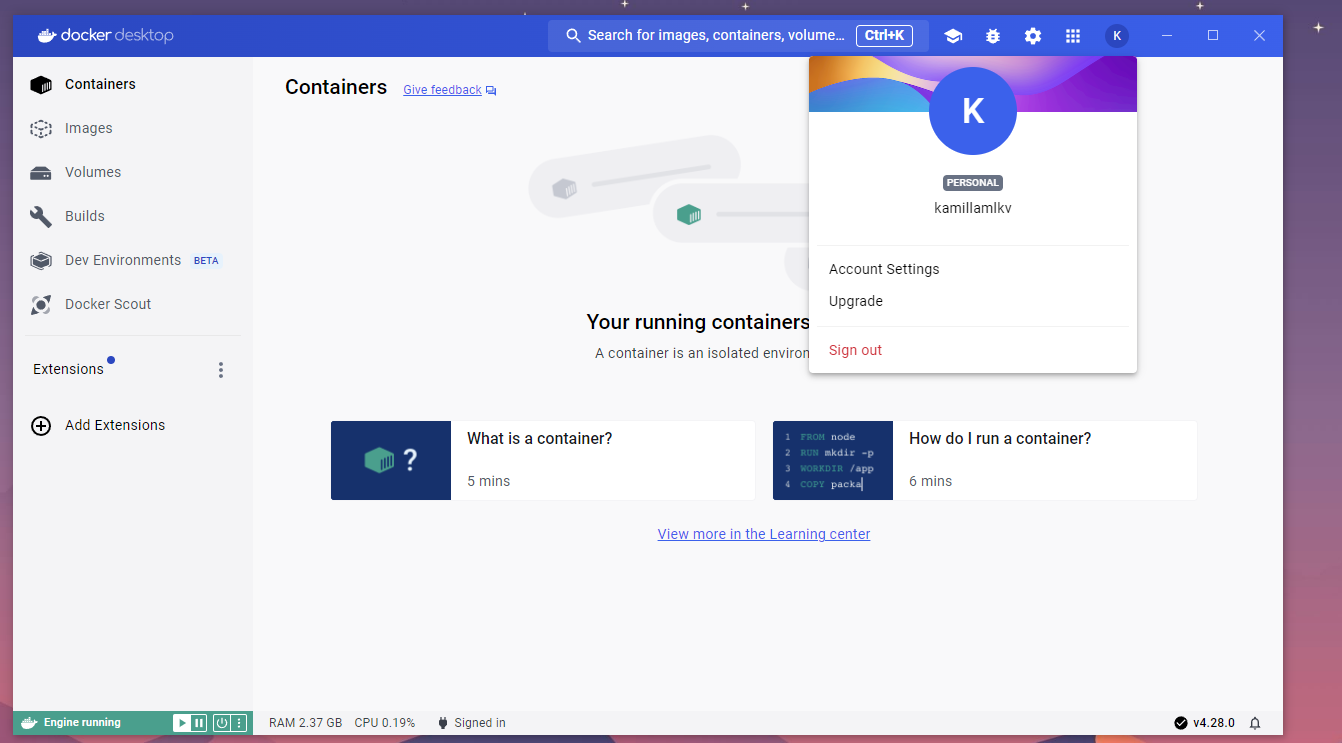
Выполнила: студентка группы 253502,

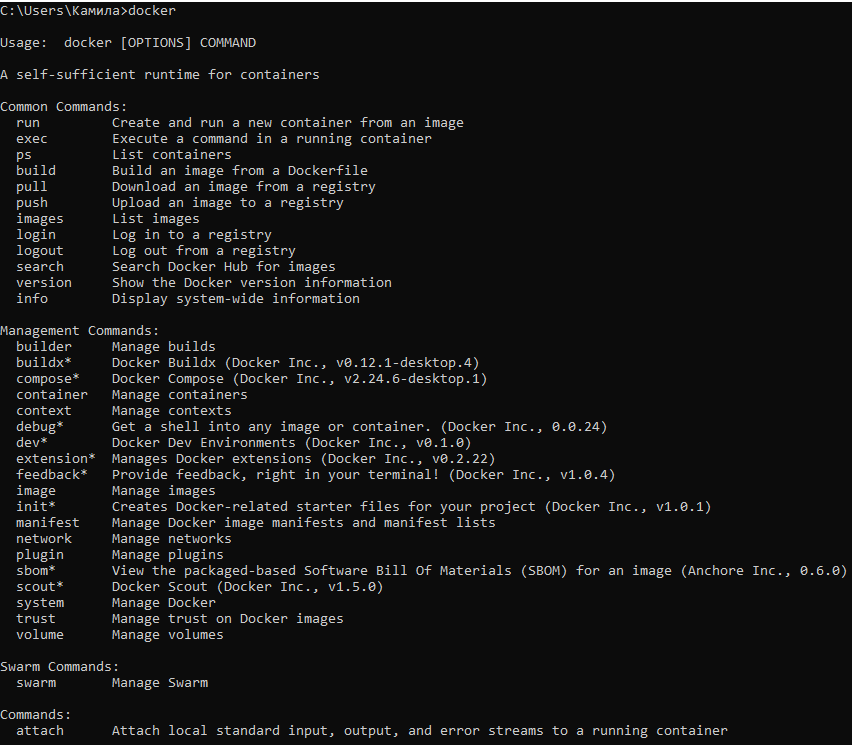
Меликова Камилла Раидовна

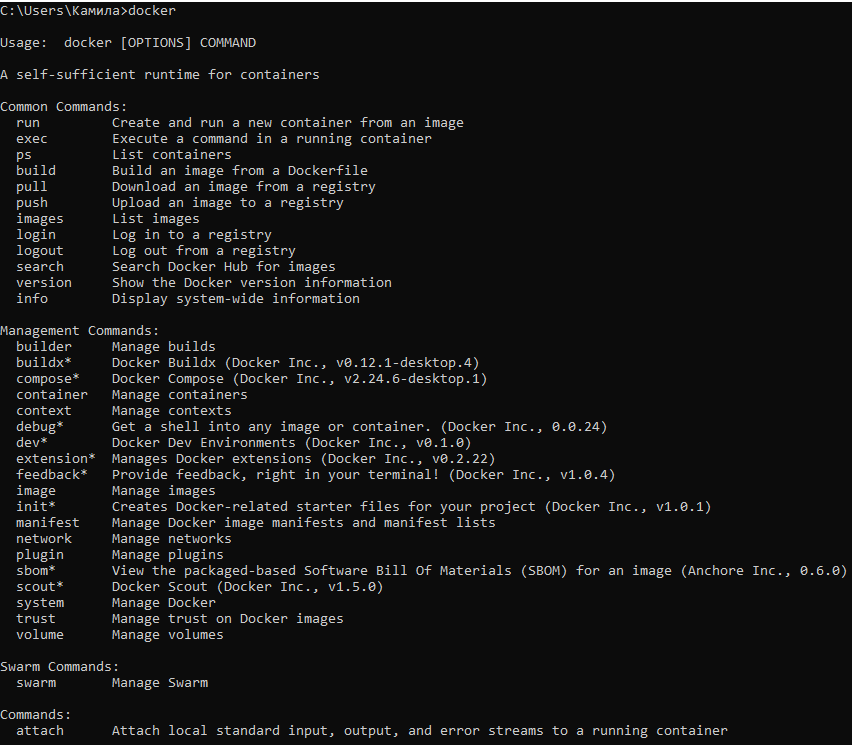
Проверила: Жвакина Анна Васильевна

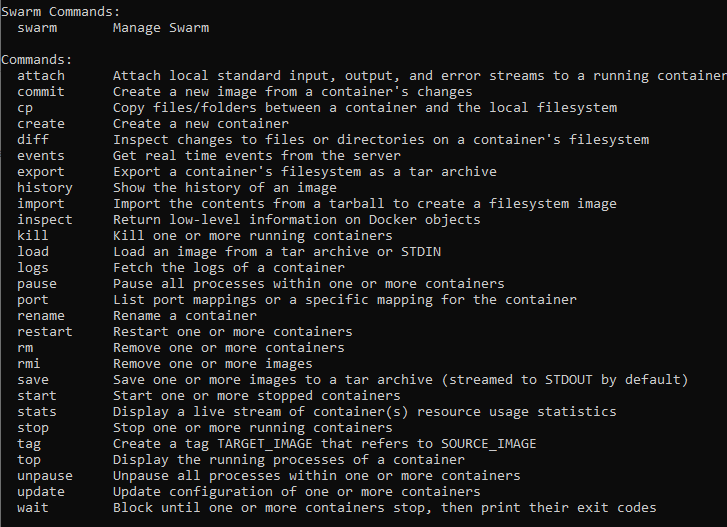
**Цель:** Познакомиться с возможностями и получить практические навыки работы с Docker.

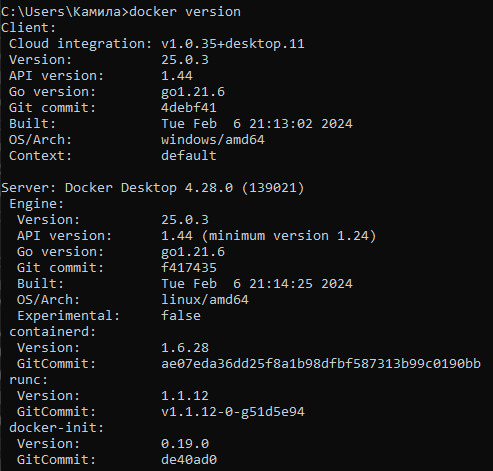
**Задание 1.** Подготовьте рабочее окружение в соответствии с типом вашей операционной системы (установите Docker, выполните базовую настройку).







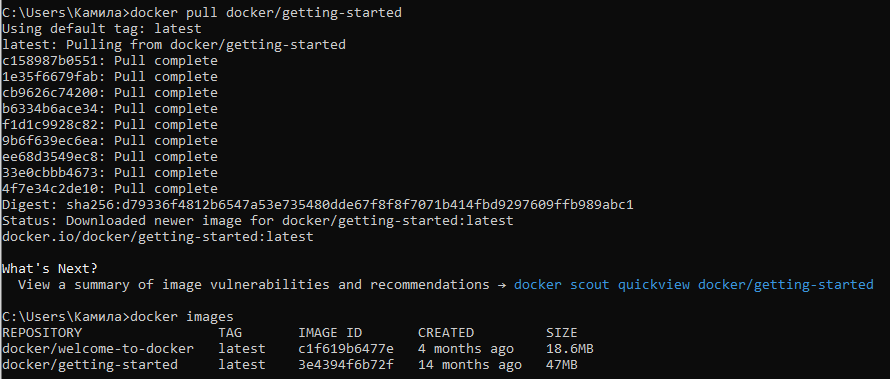


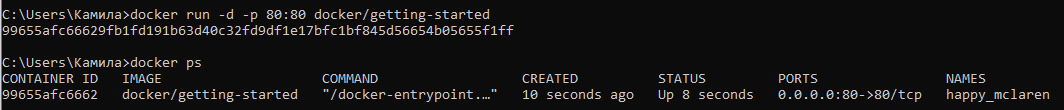


**Задание 2.** Изучите простейшие консольные команды и возможности Docker Desktop (см. лекцию), создать собственный контейнер docker/getting-started, открыть в браузере и изучить tutorial.

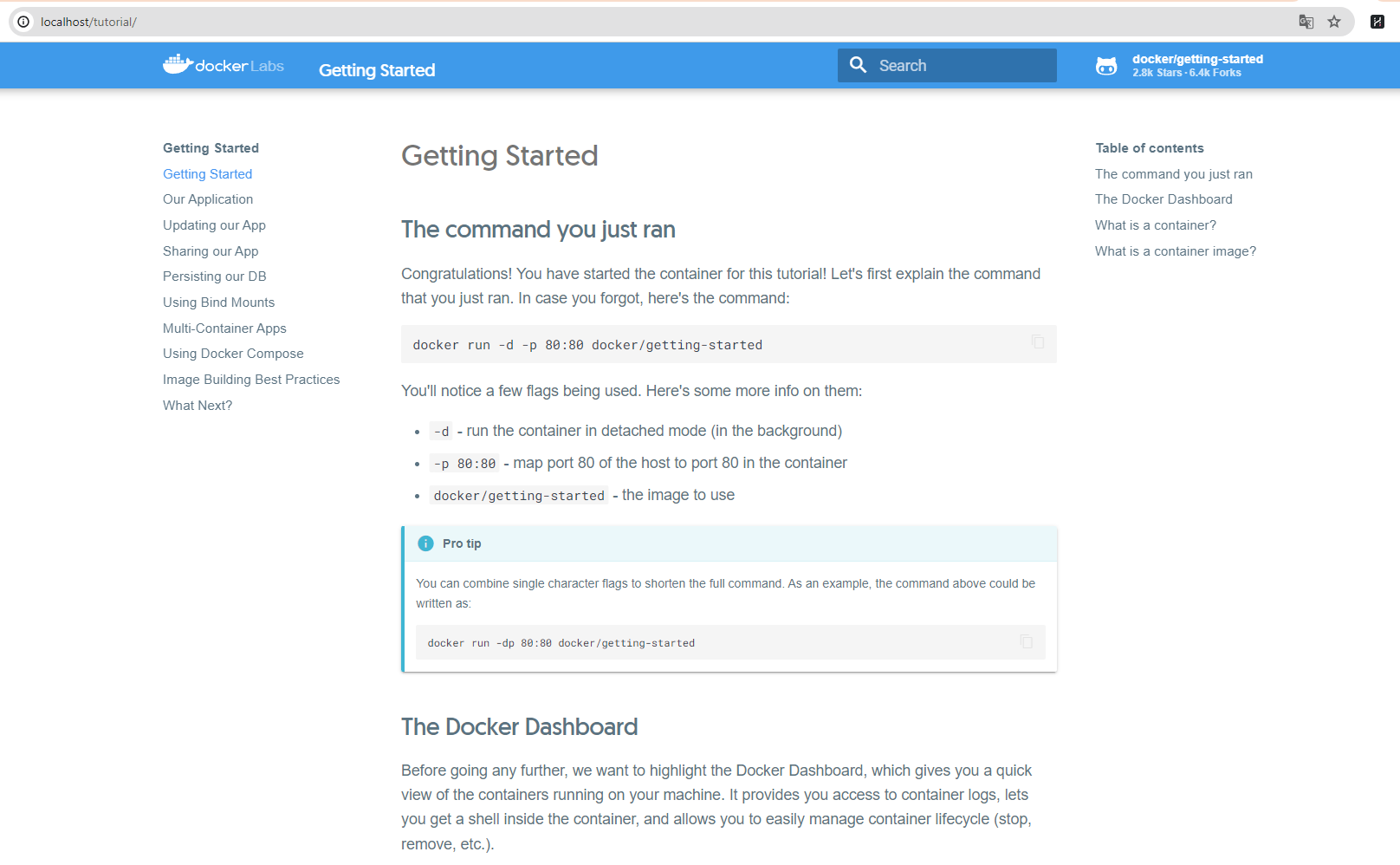


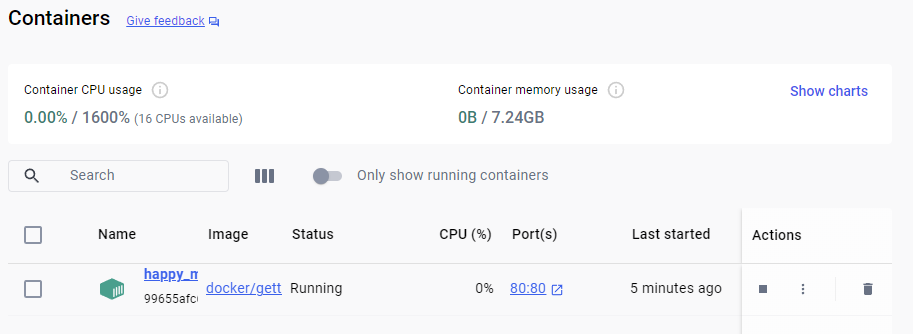




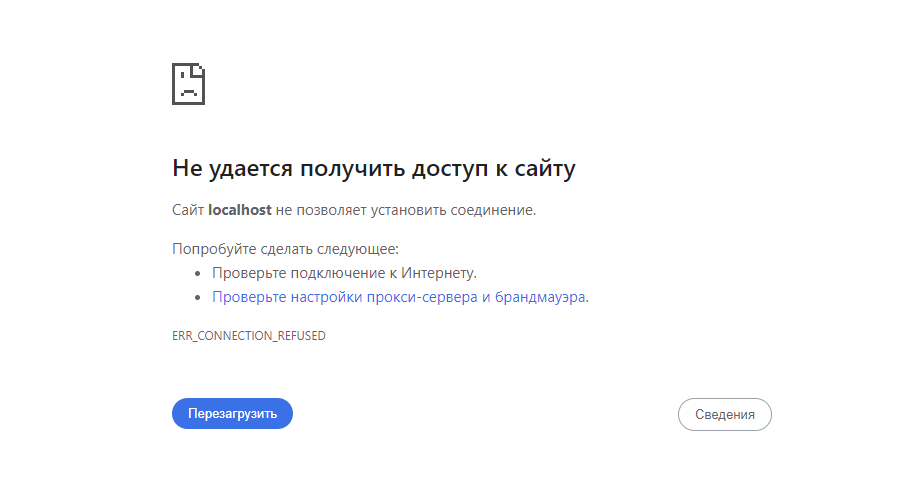


http://localhost

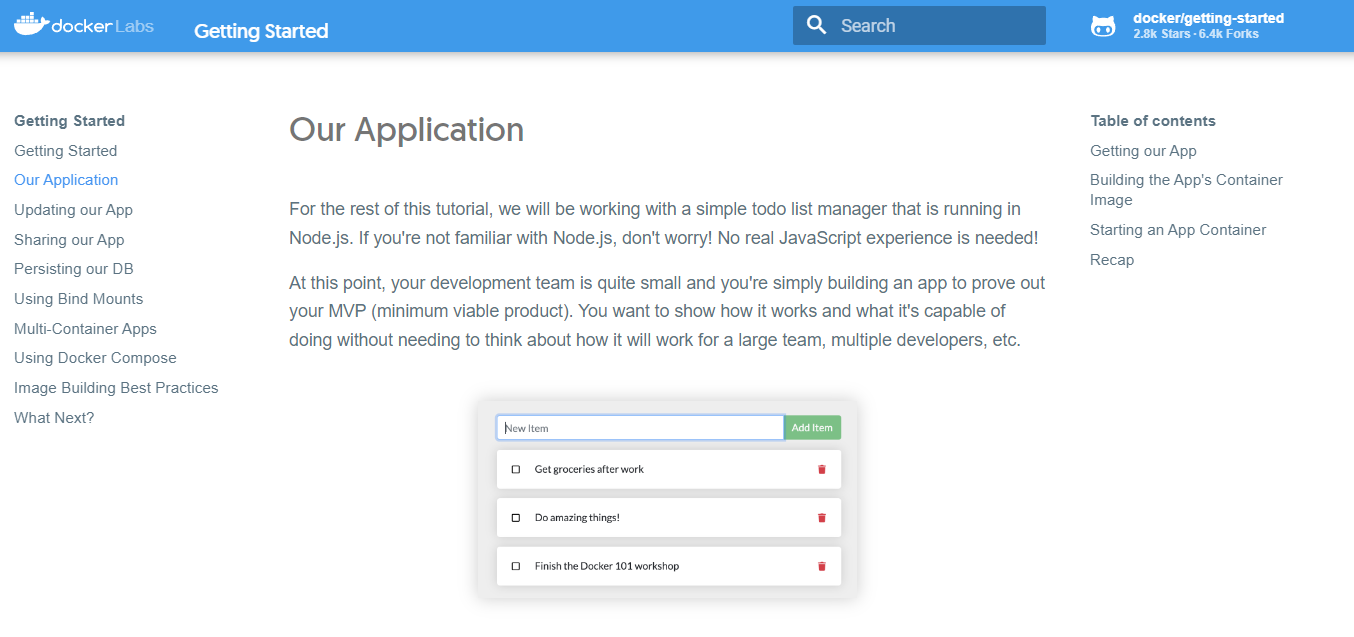








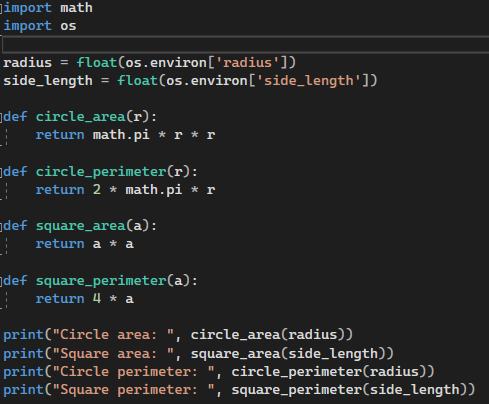




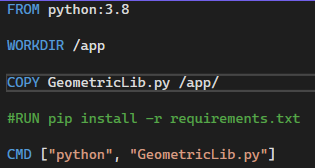
**Задание 3.** Создайте docker image, который запускает скрипт с использованием функций из <https://github.com/smartiqaorg/geometric_lib>.

1. Способ передачи данных – переменные окружения.
2. Создать Dockerfile для реализации сборки собственных Docker образов
3. Использовать его для создания контейнера. Протестировать использование контейнера

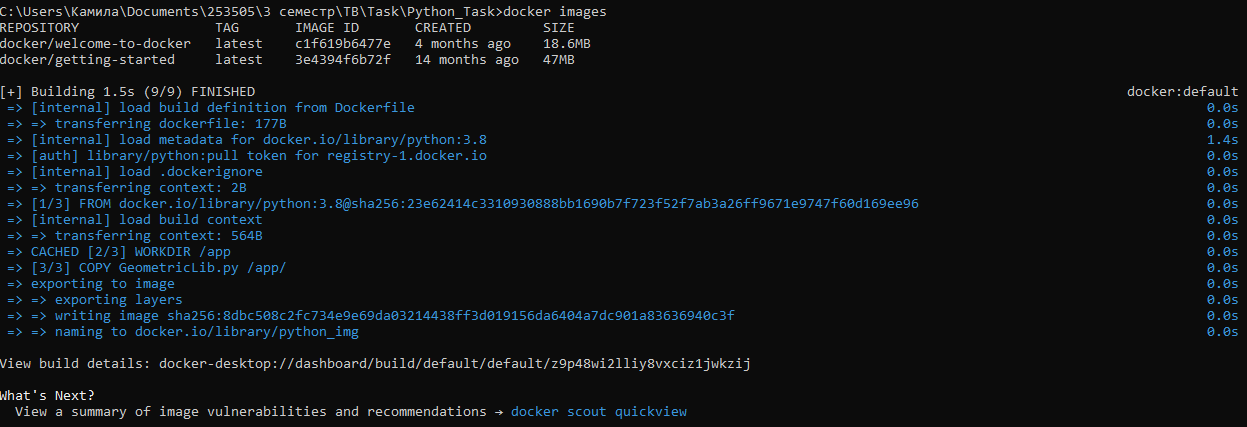
Скрипт:

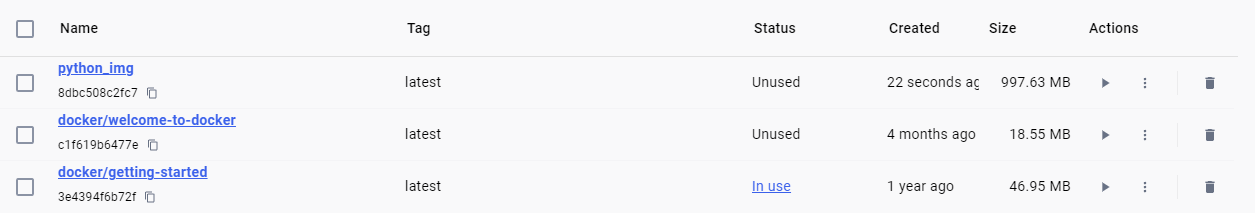


Dockerfile:

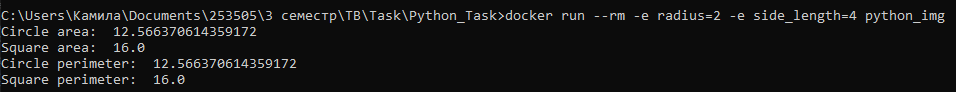


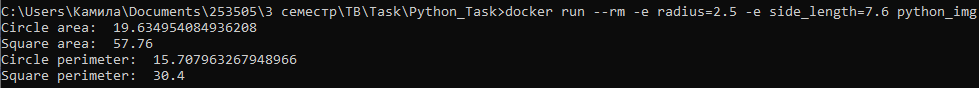
Создание образа с помощью команды docker build -t python\_img .





Запуск контейнера на основе созданного образа с передачей данных:





Здесь используется флаг rm для автоматического удаления контейнера после его завершения, а также флаг e для передачи значений переменных окружения.

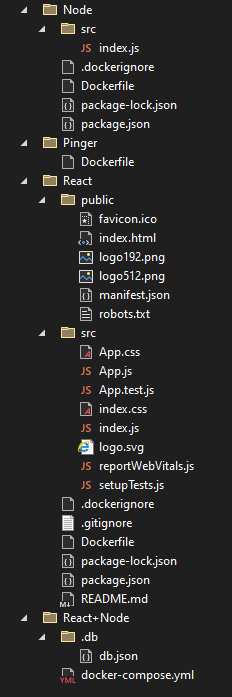
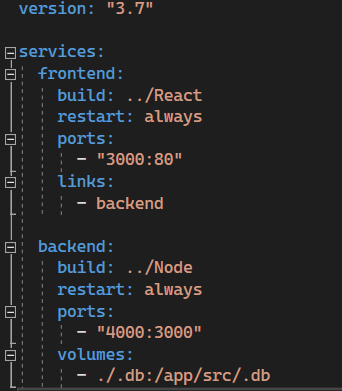
**Задание 4-5.** Скачать любой доступный проект с GitHub с произвольным стеком технологий или использовать свой, ранее разработанный. Создать для него необходимый контейнер, используя Docker Compose для управления многоконтейнерными приложениями. Запустить проект в контейнере. Настроить сети и тома для обеспечения связи между контейнерами и сохранения данных.

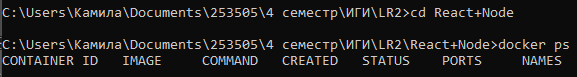
Ссылка на проект GitHub:

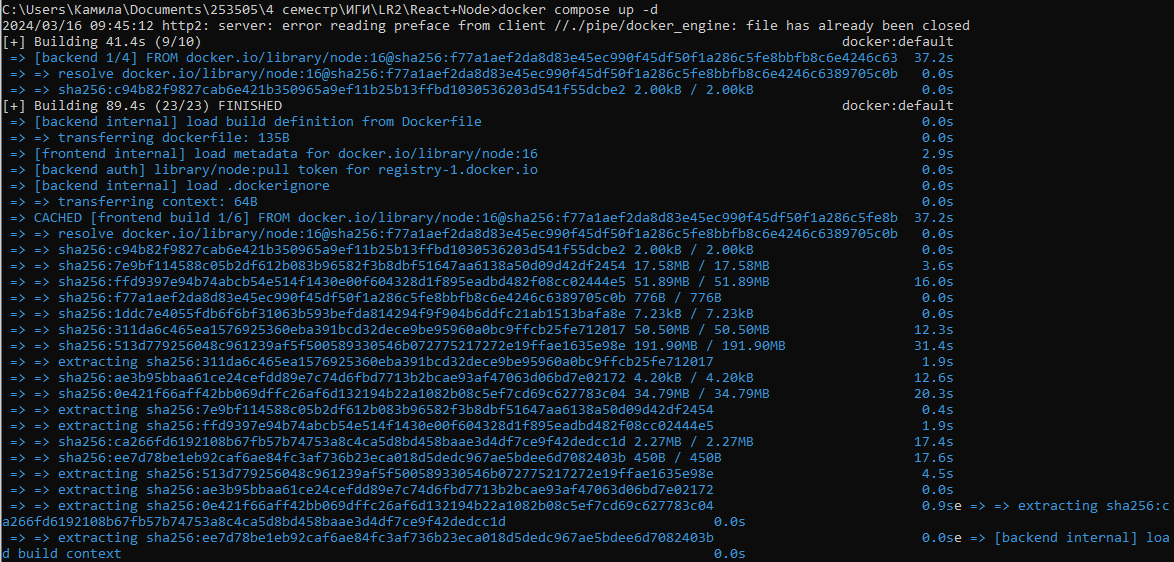
<https://github.com/abduvik/just-enough-series/tree/master>

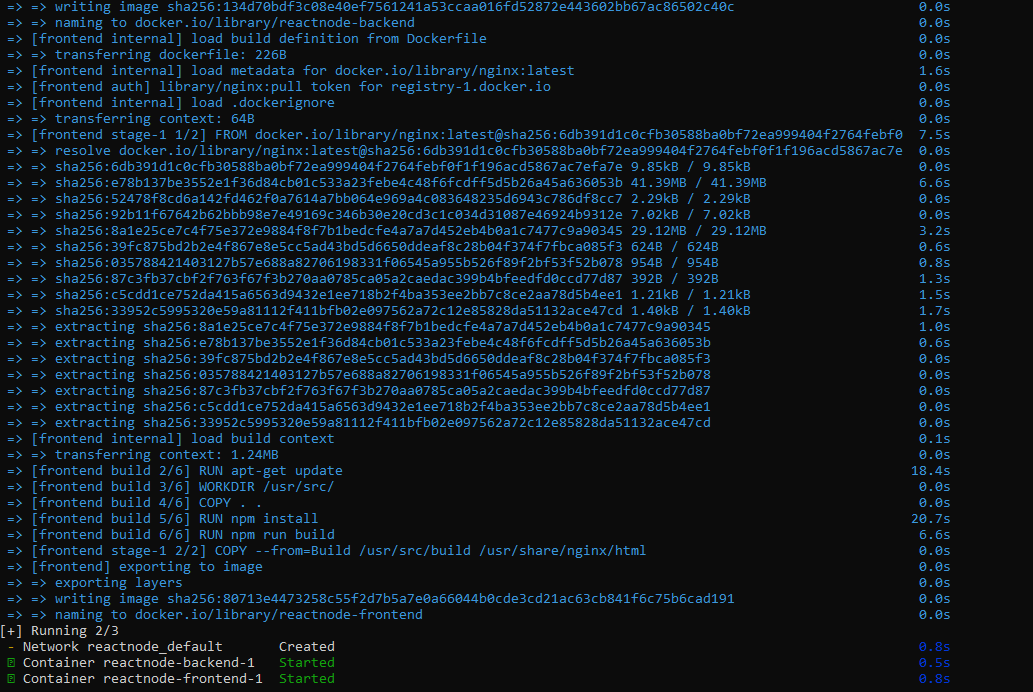
Стек технологий: Node+React

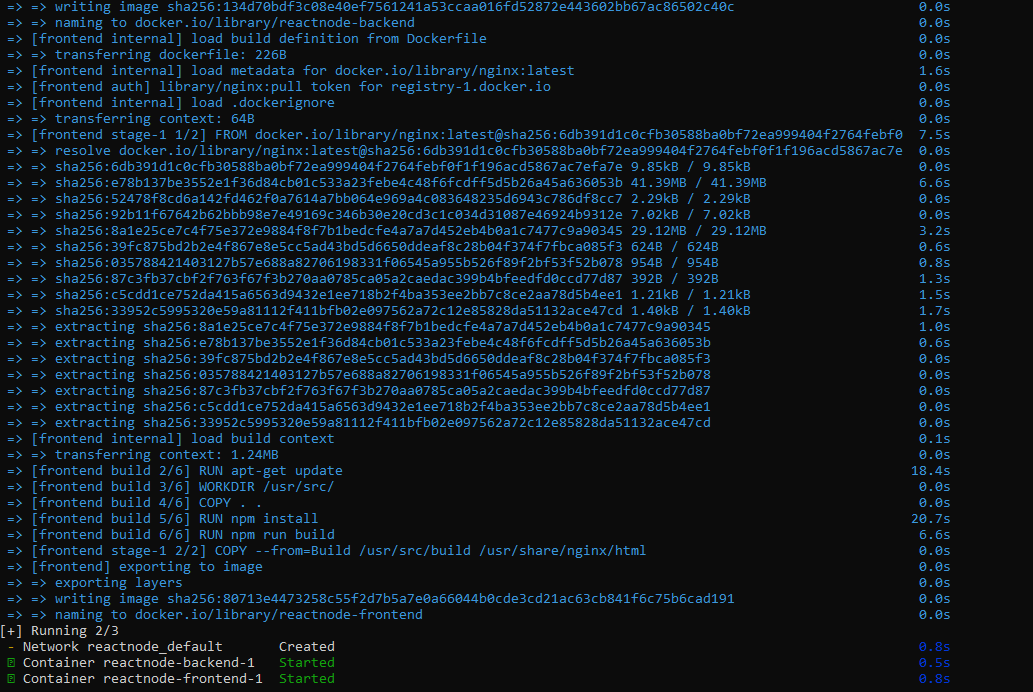
Структура проекта и файл docker-compose:

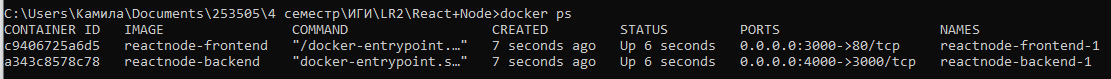
 

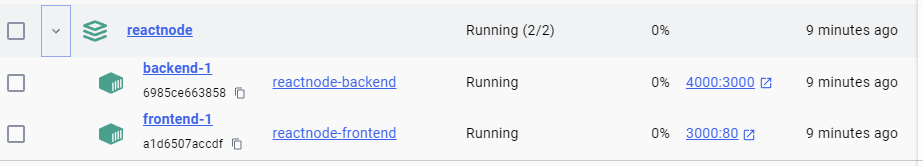


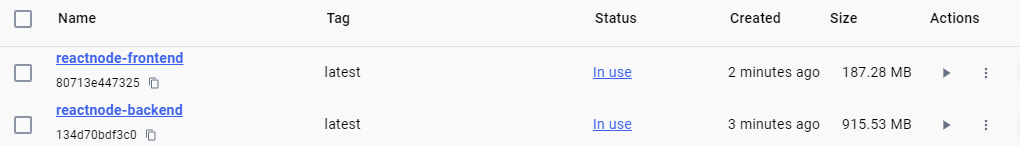




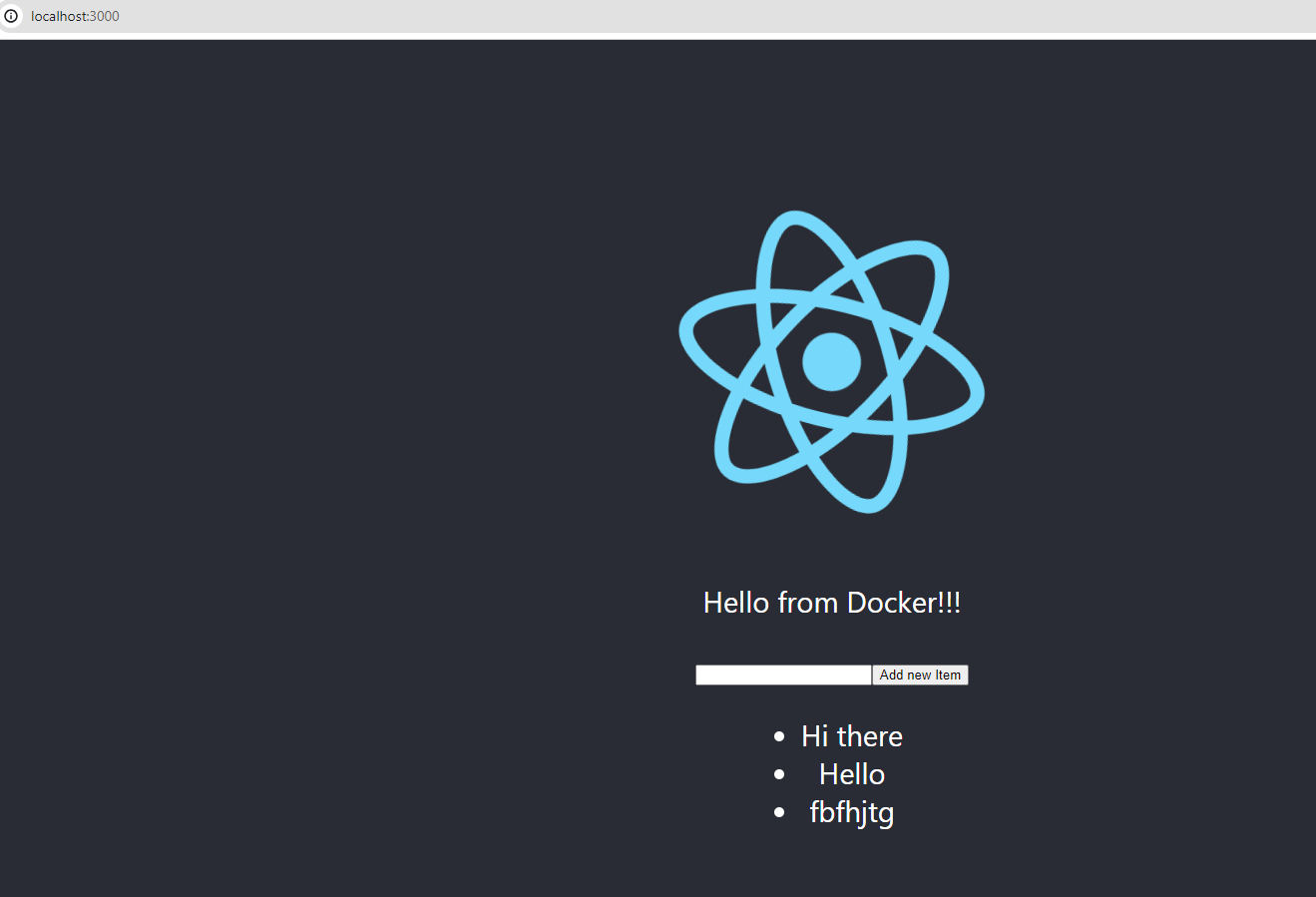




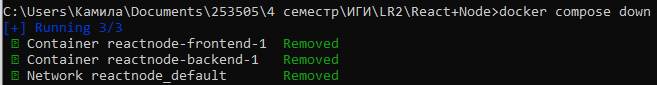


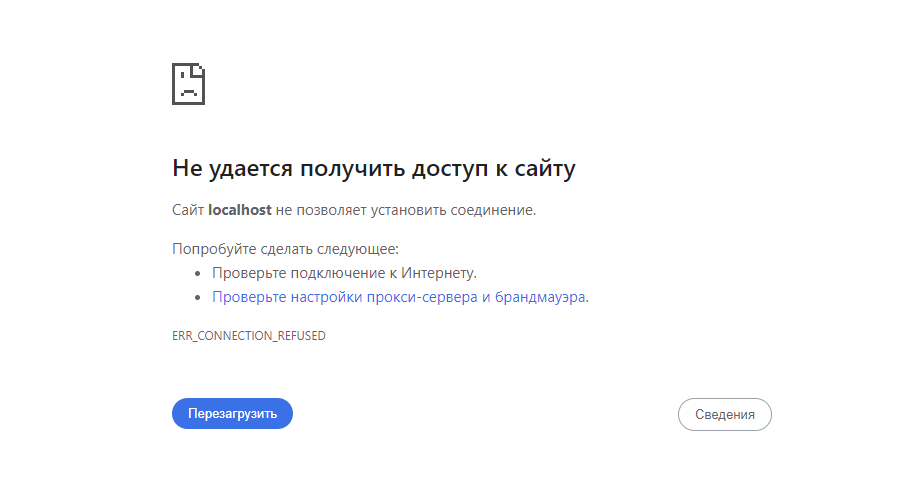


Работа программы:

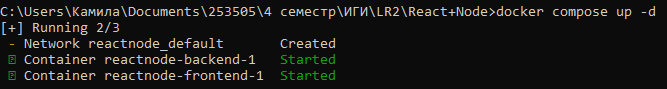


Остановим контейнер:

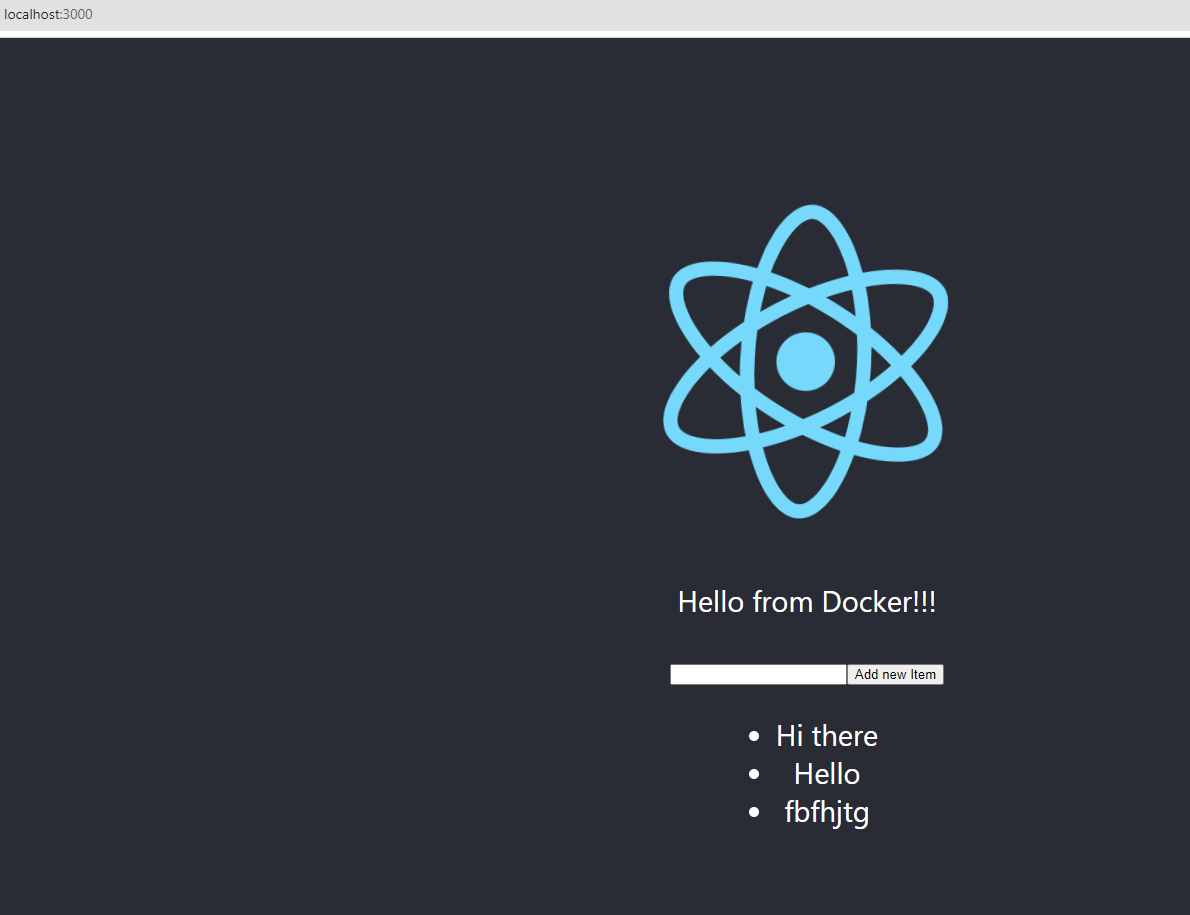




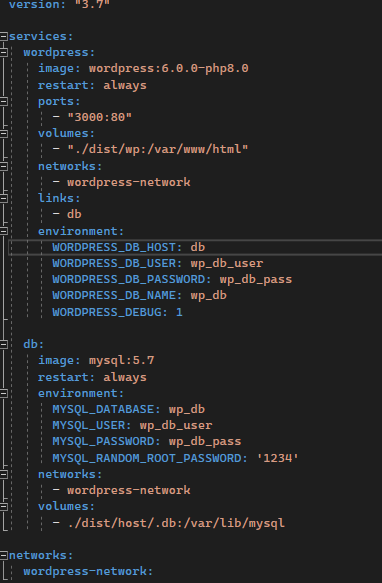
Запустим снова:

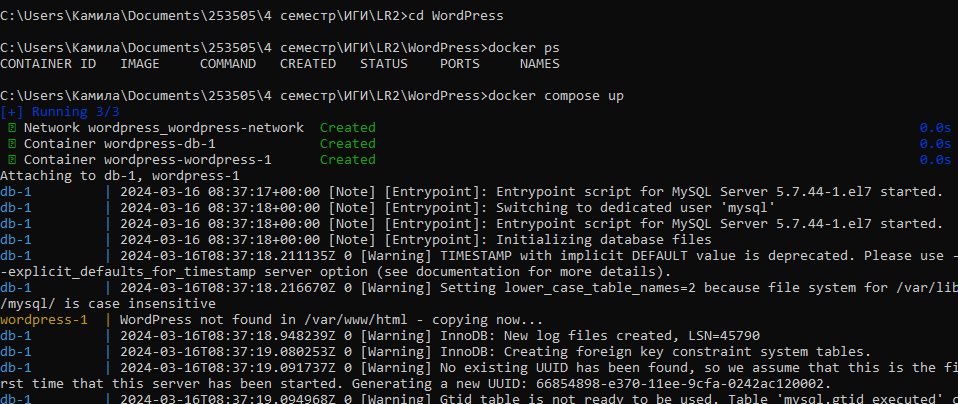


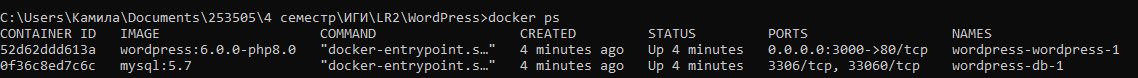
Как видим, данные были сохранены благодаря использованию docker volumes:



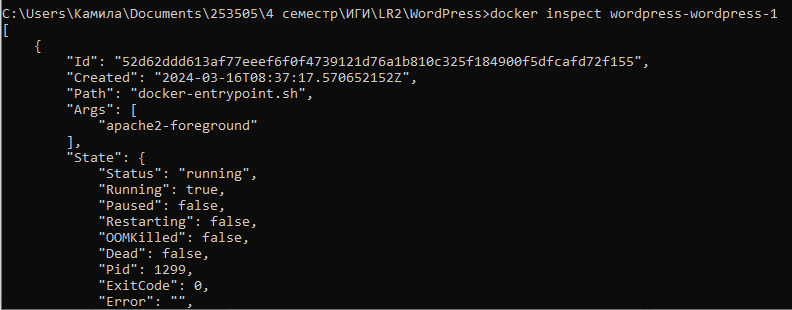
Стек технологий: WordPress+MySQL





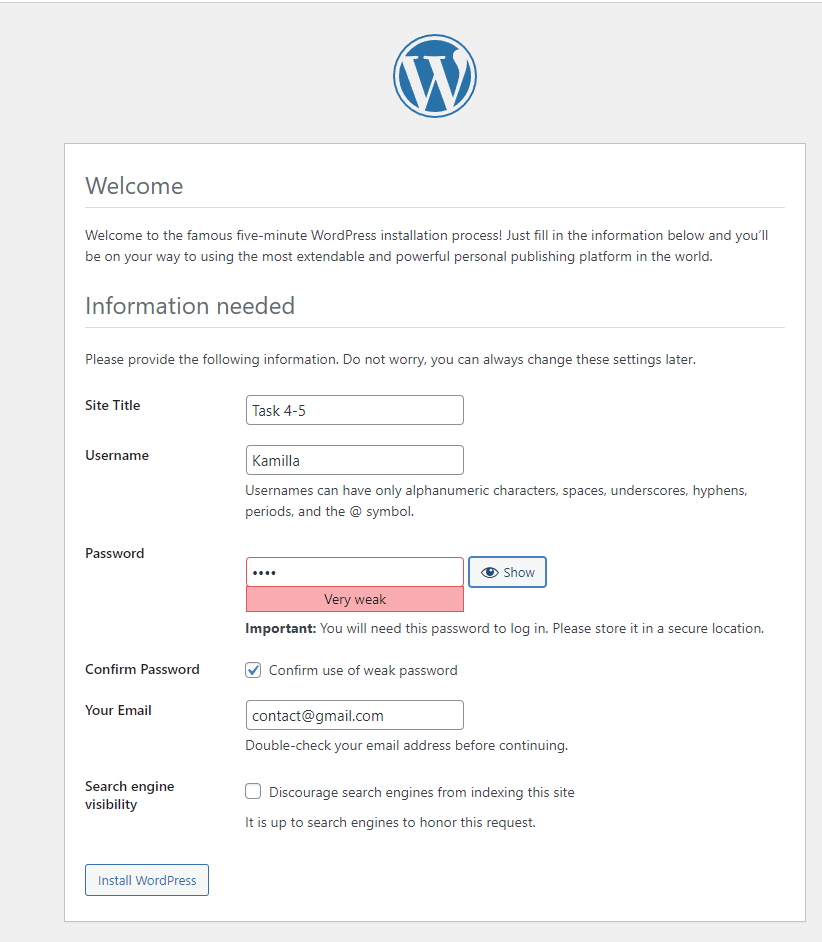


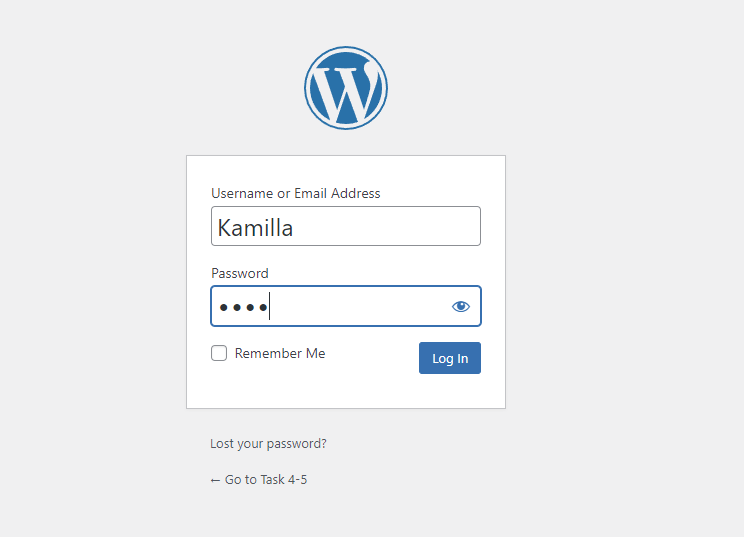
Подробная информация о контейнере:

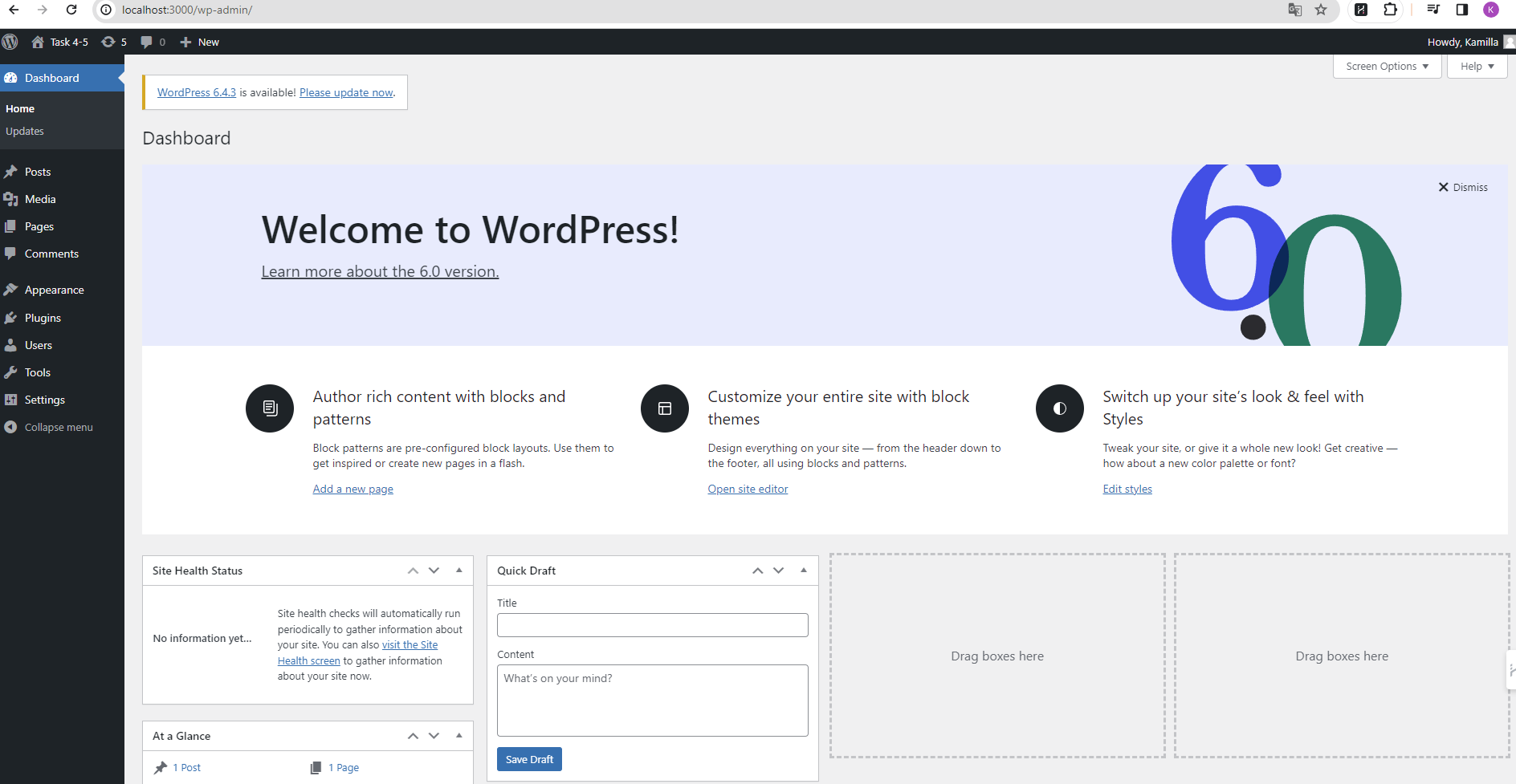




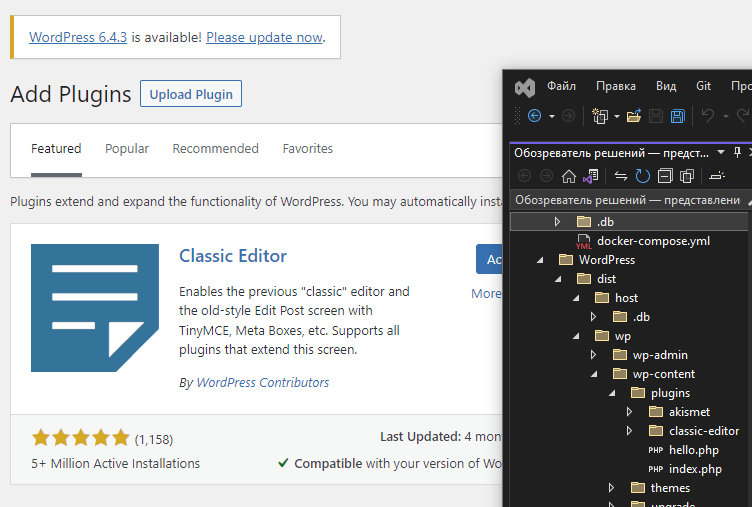
localhost:3000



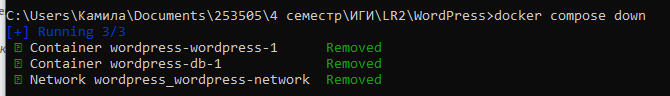


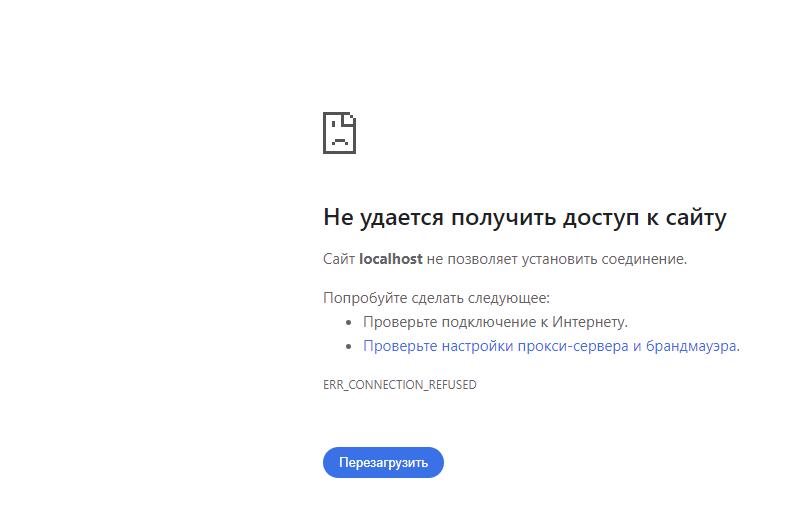


Добавим плагин Classic Editor:

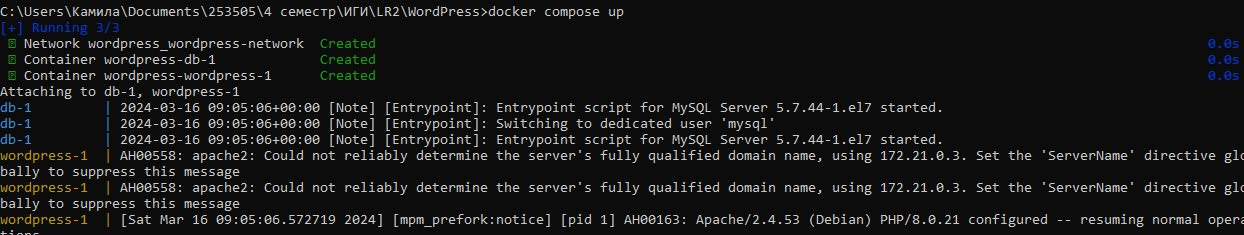


Остановим контейнер:

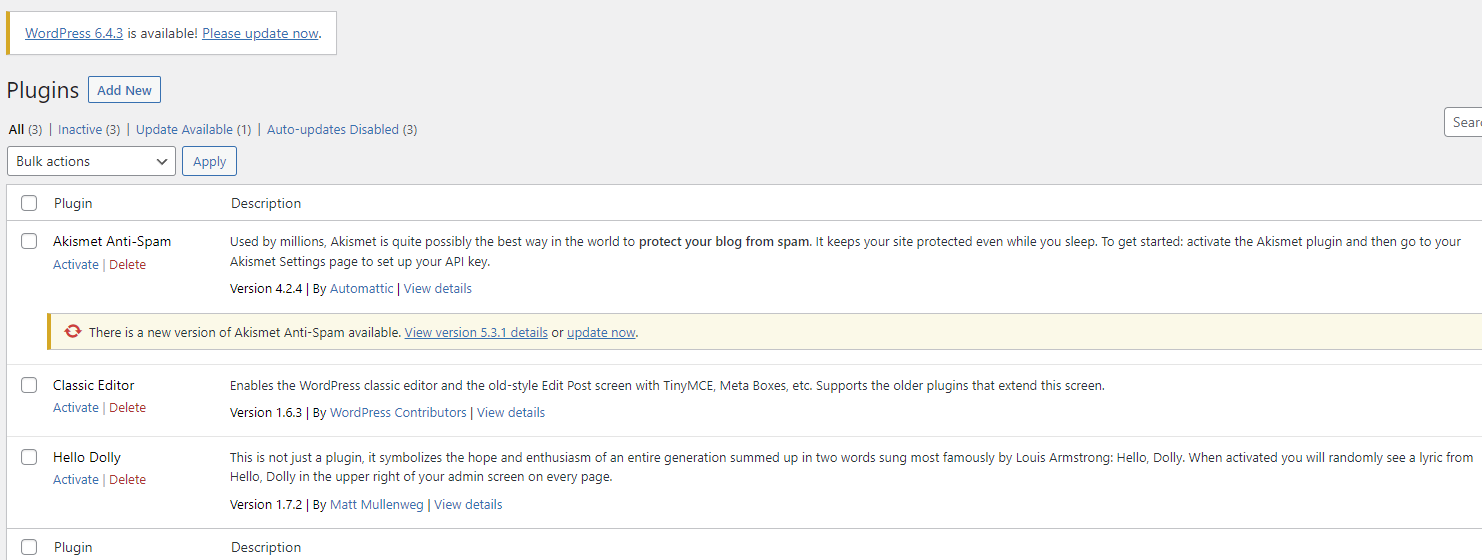




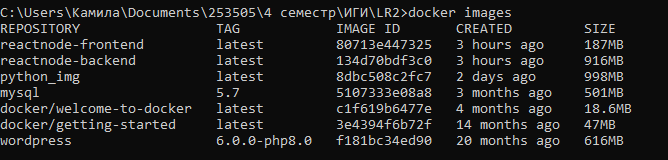
Снова запустим:

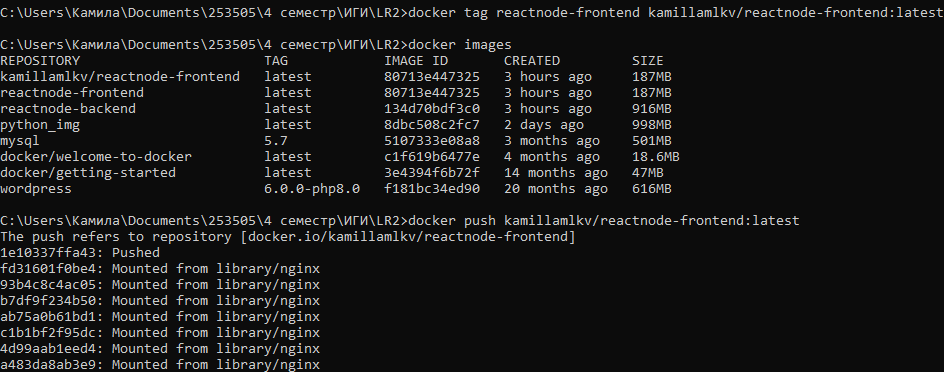


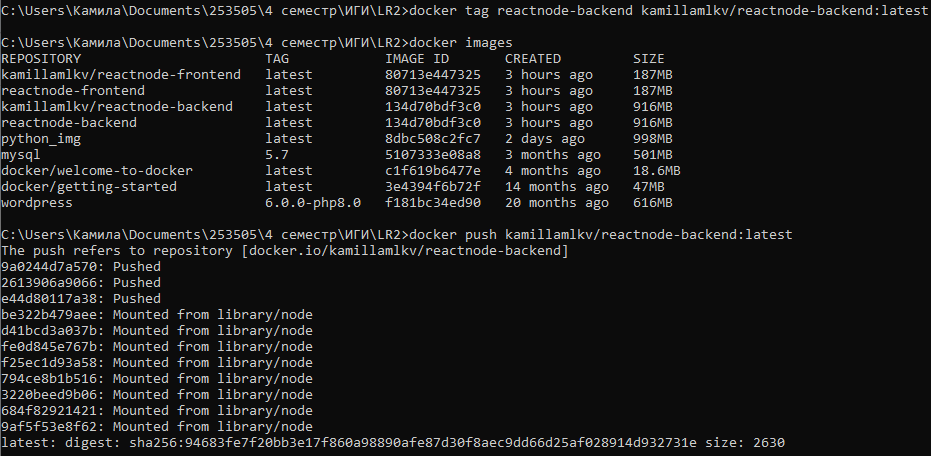
Видим, что внесенные изменения были сохранены:

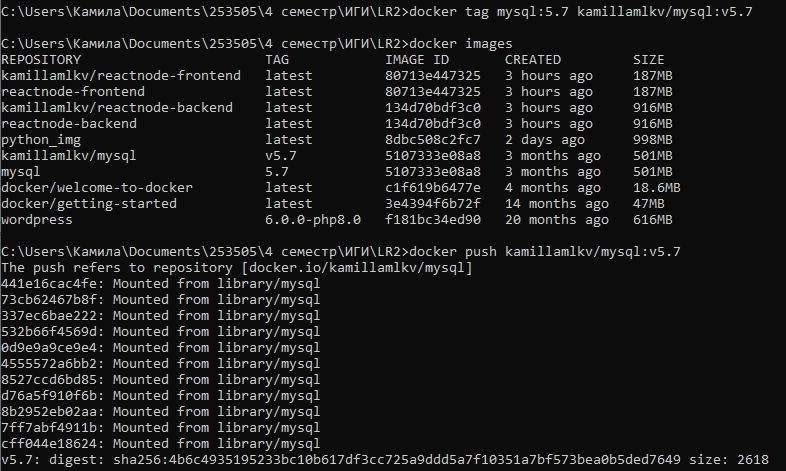


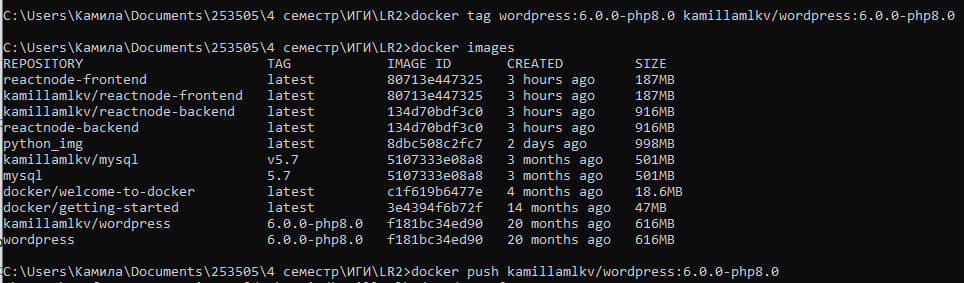
**Задание 6.** Разместите результат в созданный репозиторий в DockerHub.

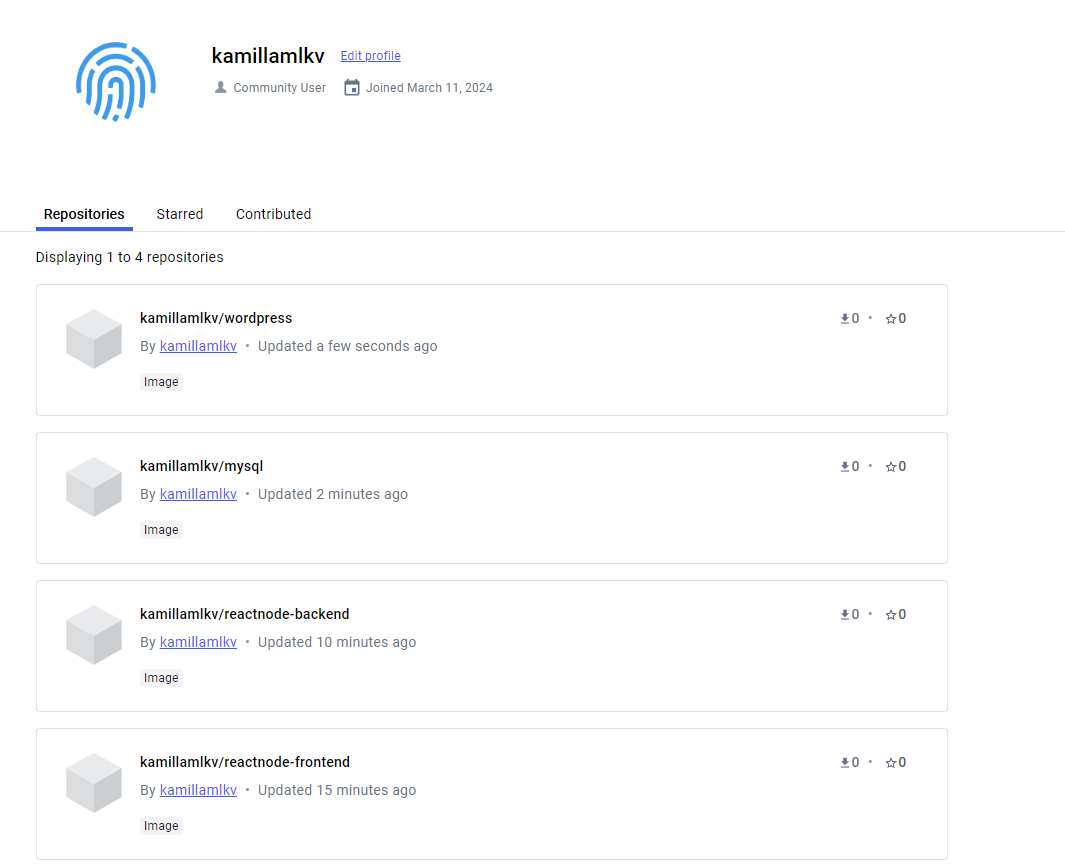






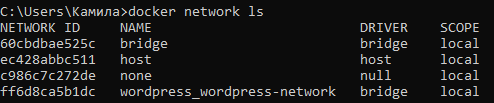


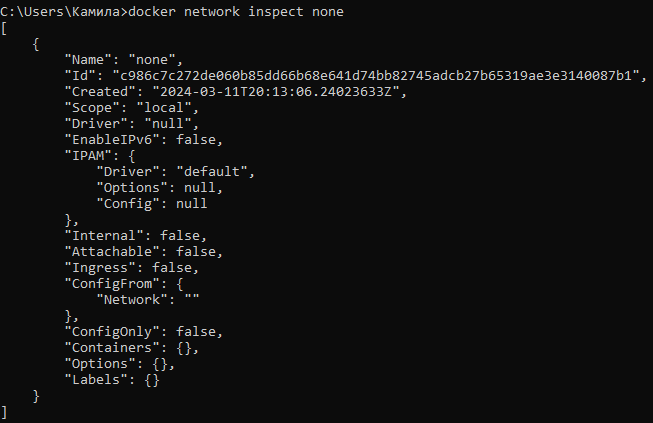


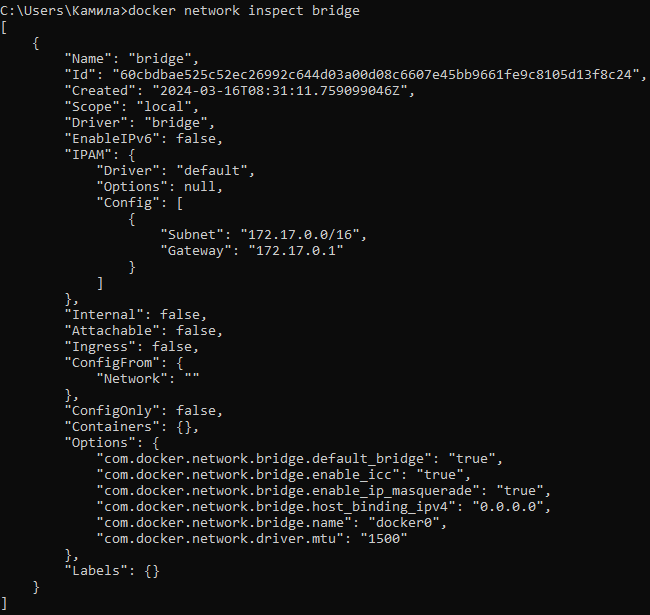


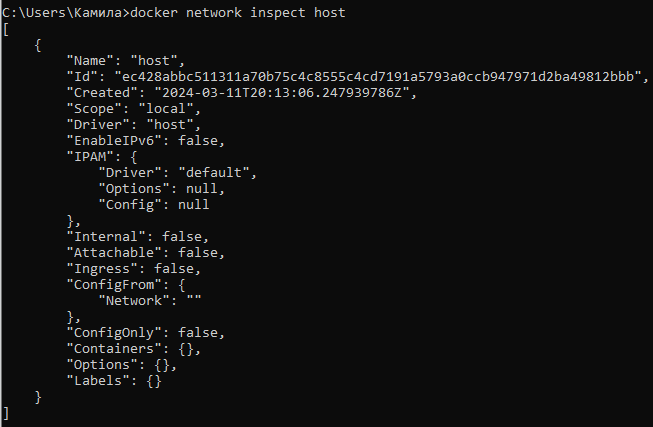
**Задание 7.** Выполните следующие действия с целью изучить особенности сетевого взаимодействия:

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

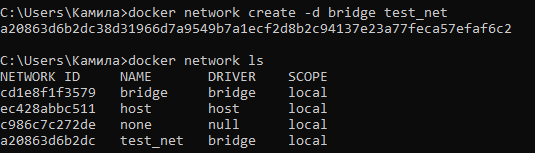


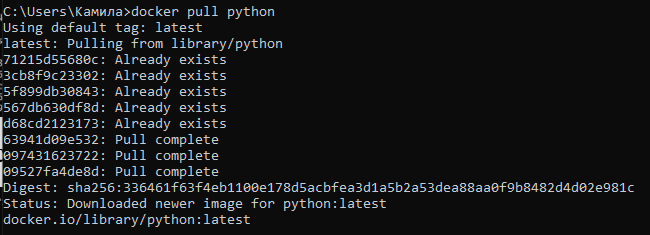


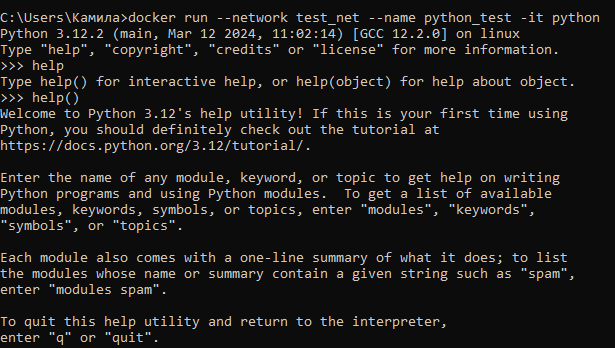


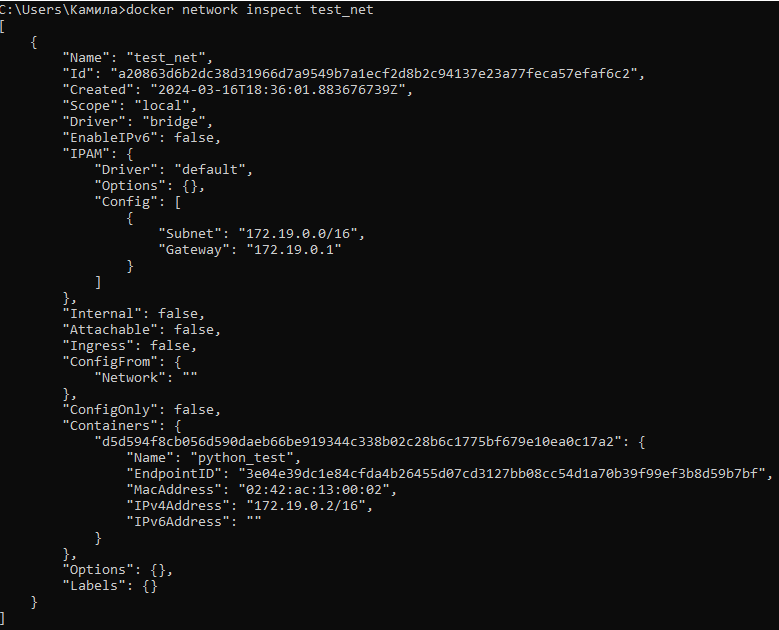


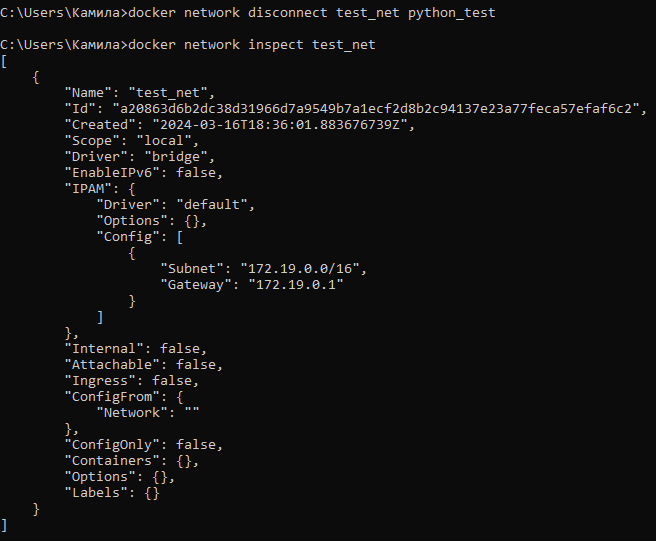
1. Создать свою собственную сеть bridge, проверить, создана ли она, запустить Docker-контейнер в созданной сети, вывести о ней всю информацию(включая IP-адрес контейнера), отключить сеть от контейнера.



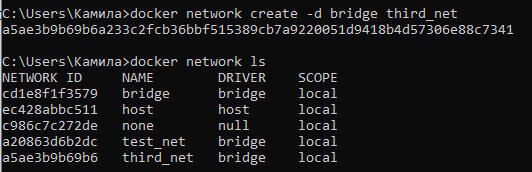


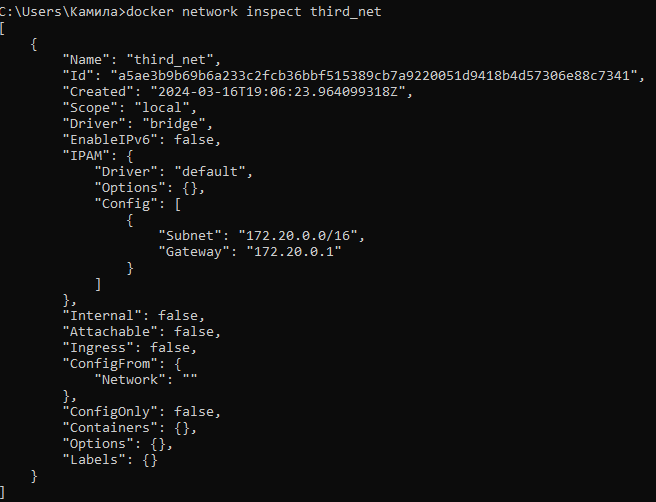




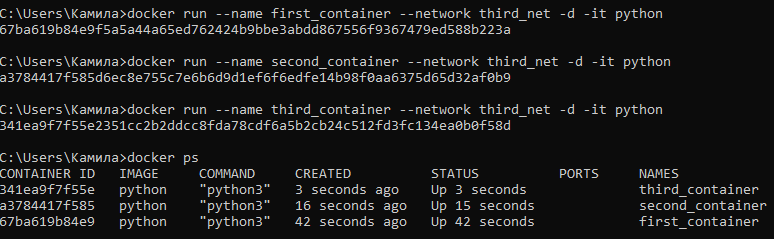


1. Создать еще одну сеть bridge, вывести о ней всю информацию, запустить в ней три контейнера, подключиться к любому из контейнеров и пропинговать два других из оболочки контейнера, убедиться, что между контейнерами происходит общение по IP-адресу.



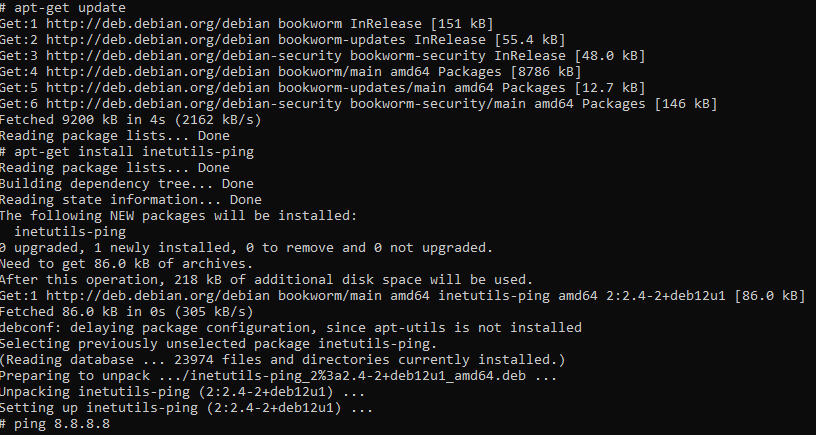


Запуск трех контейнеров:

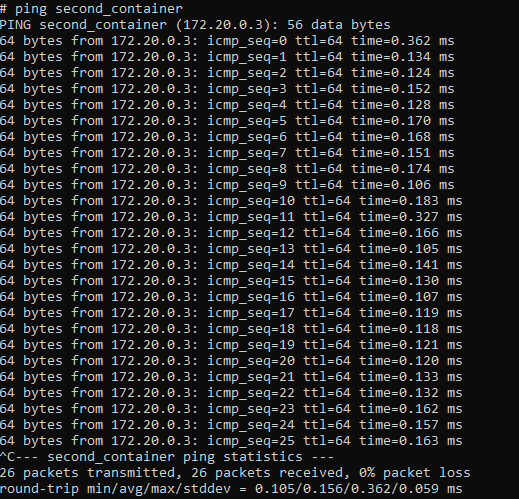


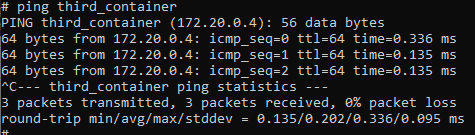
Подключение к первому контейнеру:



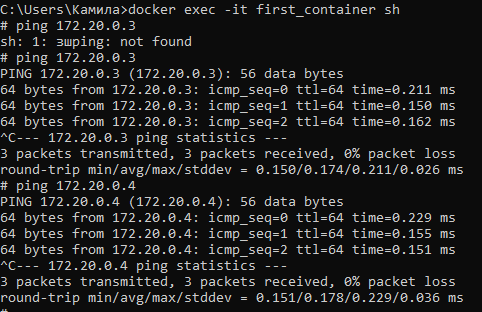


Пинг по имени:

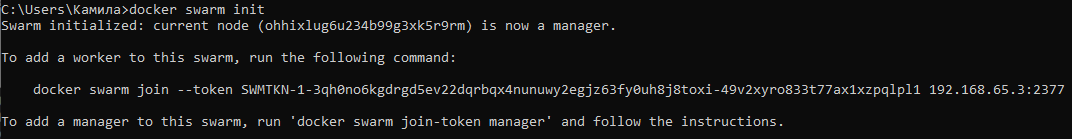


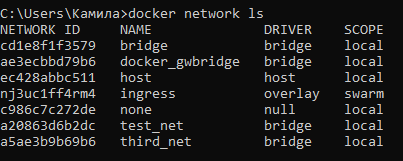


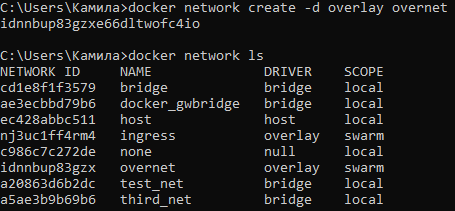
Пинг по IP-адресу:

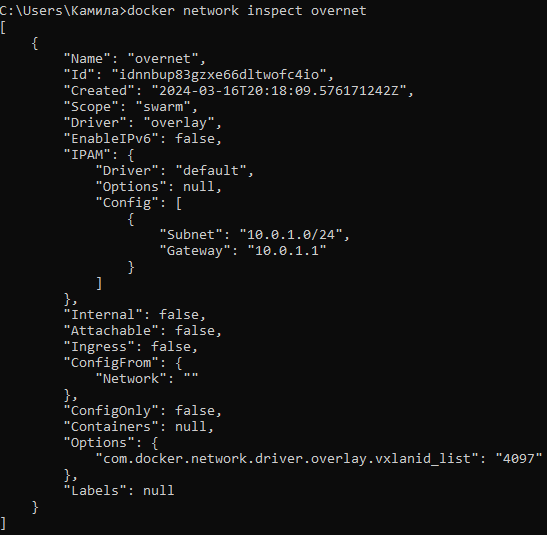


1. Создать свою собственную сеть overlay, проверить, создана ли она, вывести о ней всю информацию.

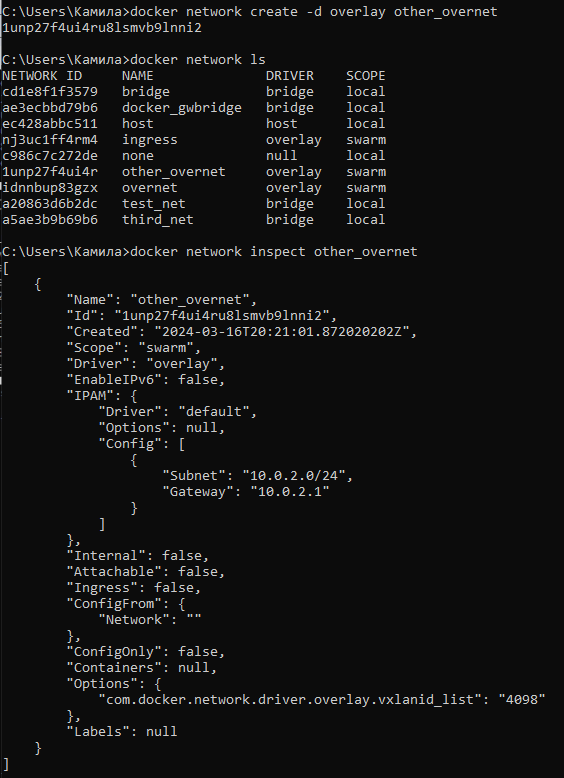


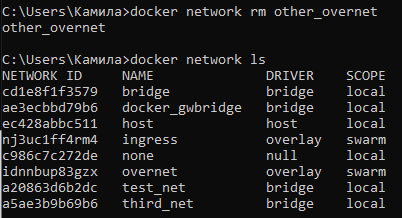






1. Создать еще одну сеть overlay, проверить, создана ли она, вывести о ней всю информацию, удалить сеть.



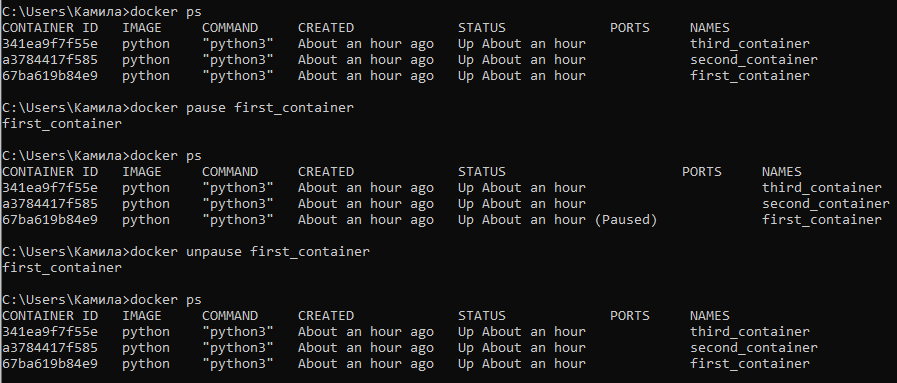


1. Попробовать создать сеть host.

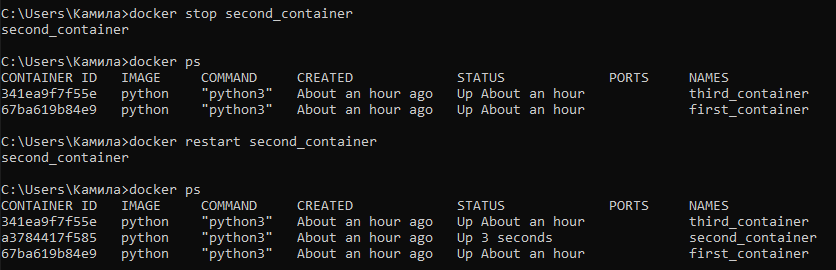


Демонстрация некоторых команд:

1. pause, unpause



1. restart



1. logs

