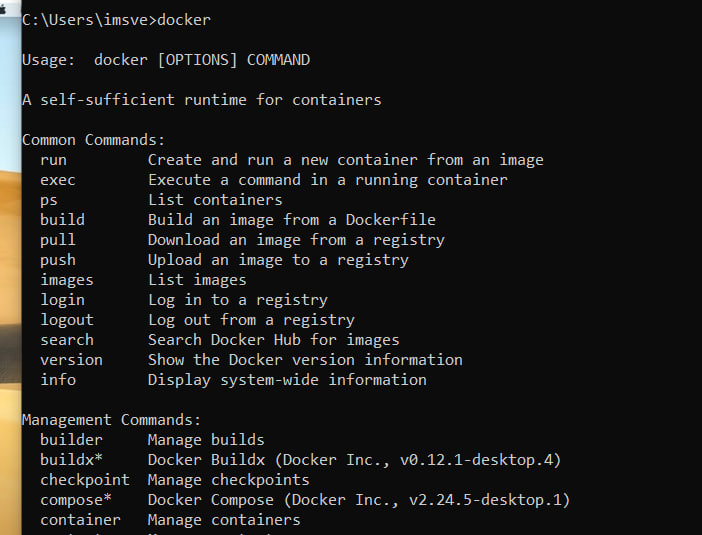
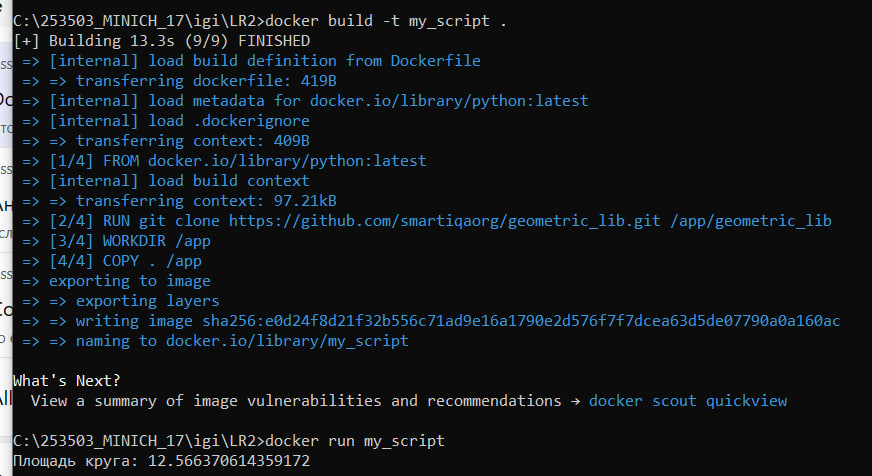
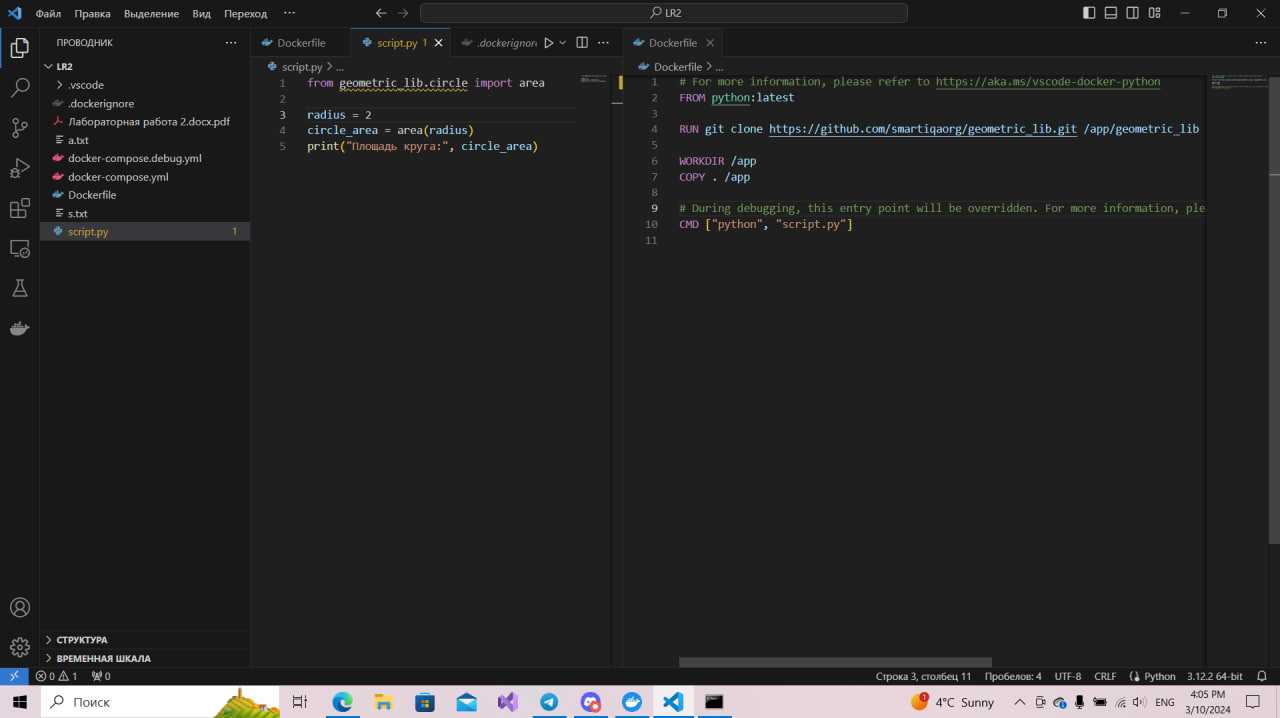
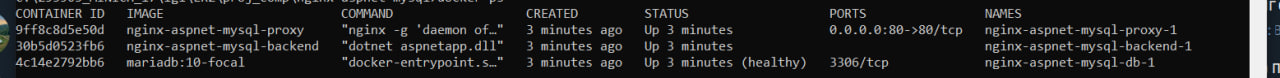
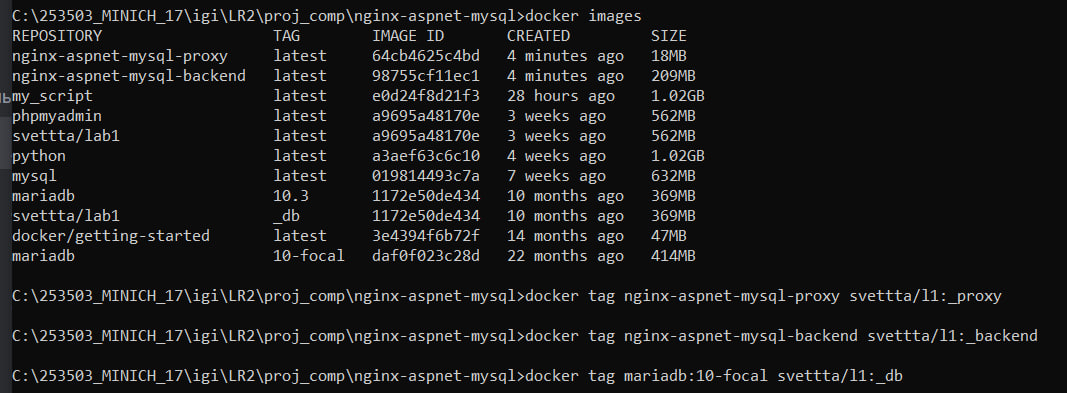
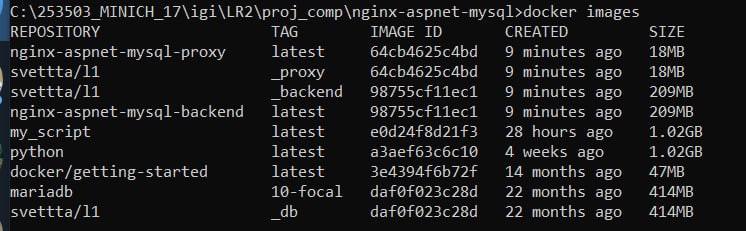
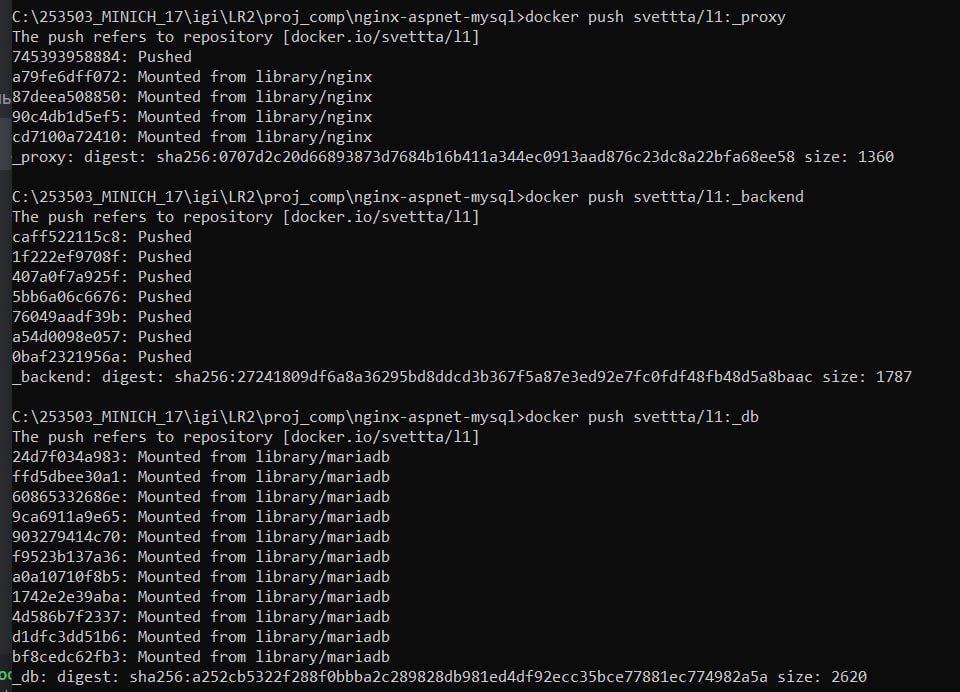
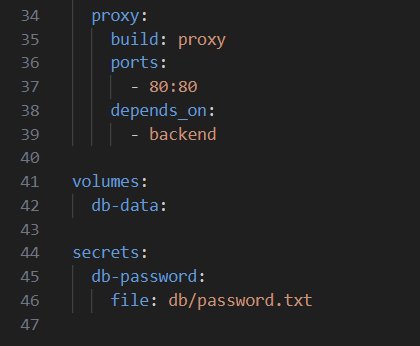
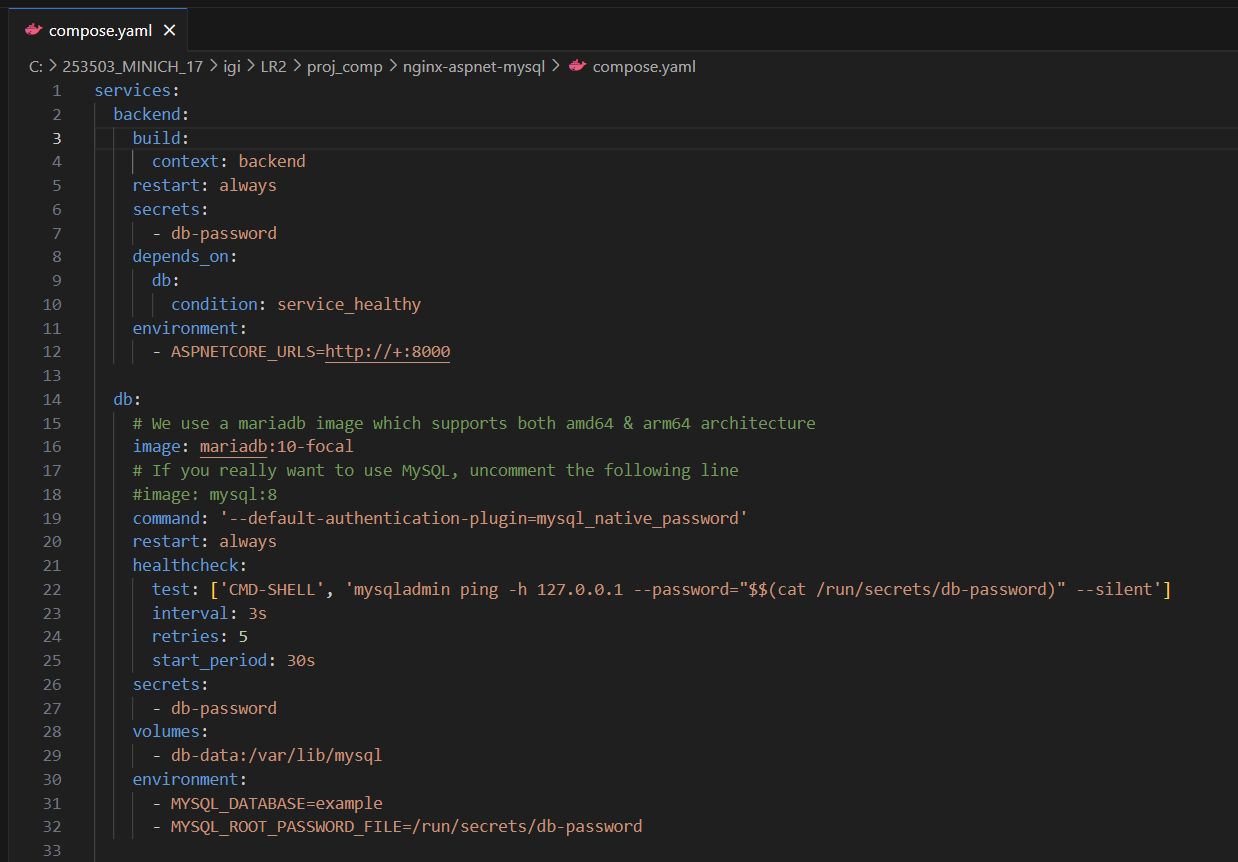
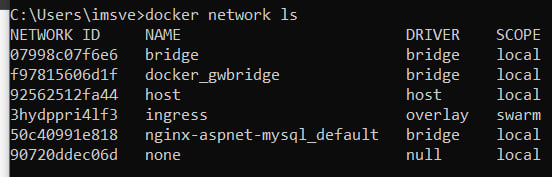
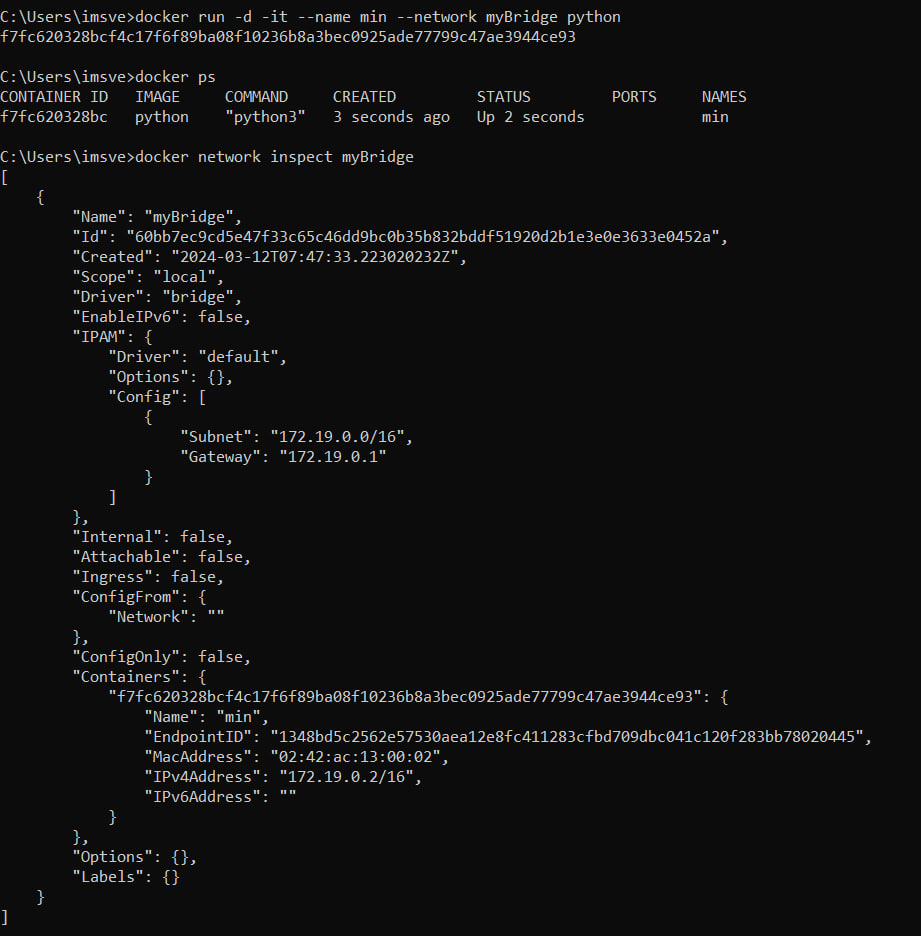
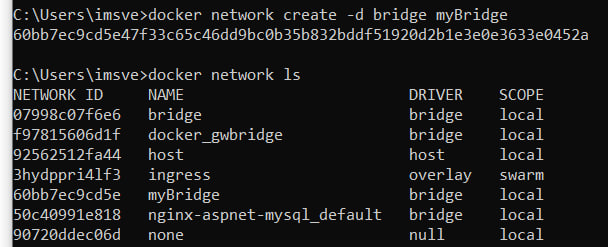
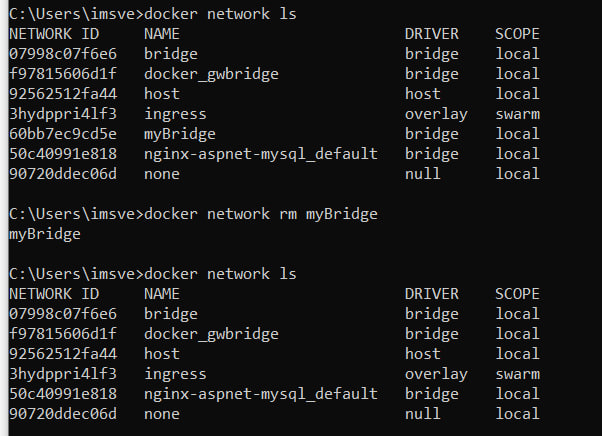
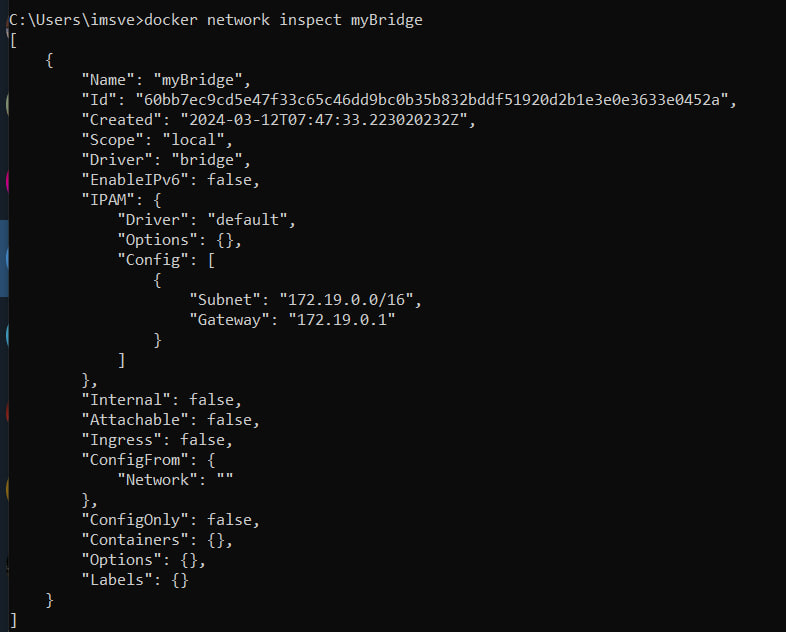
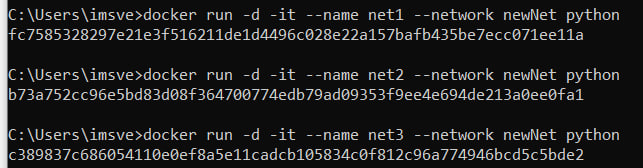
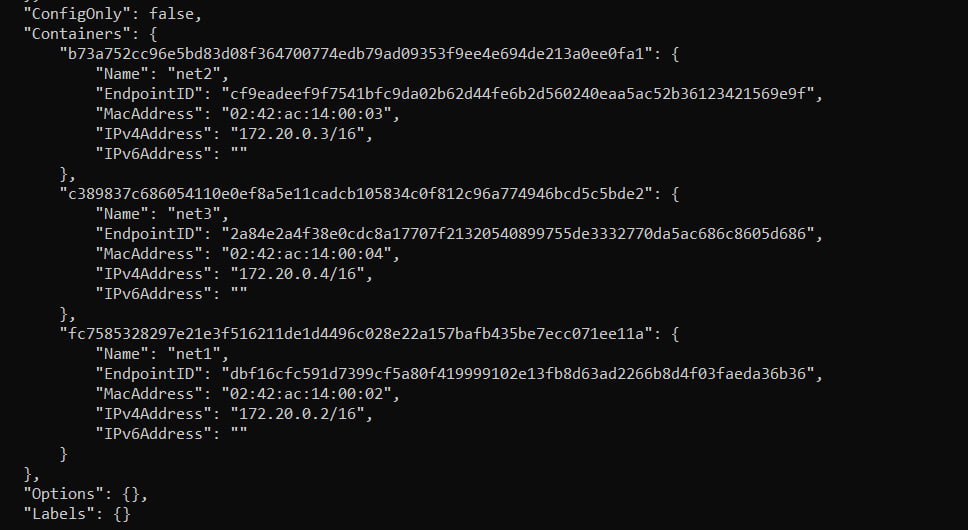
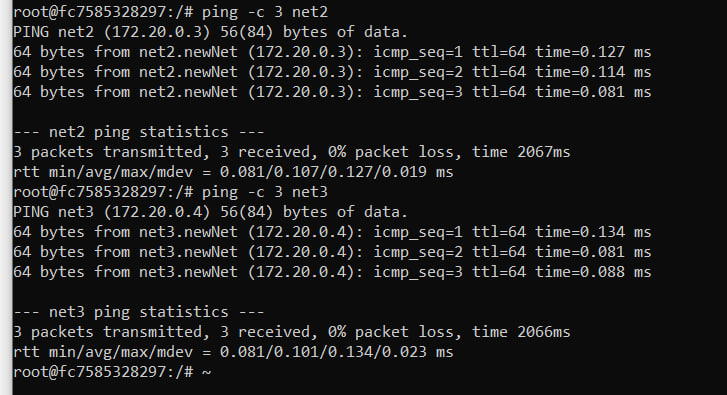
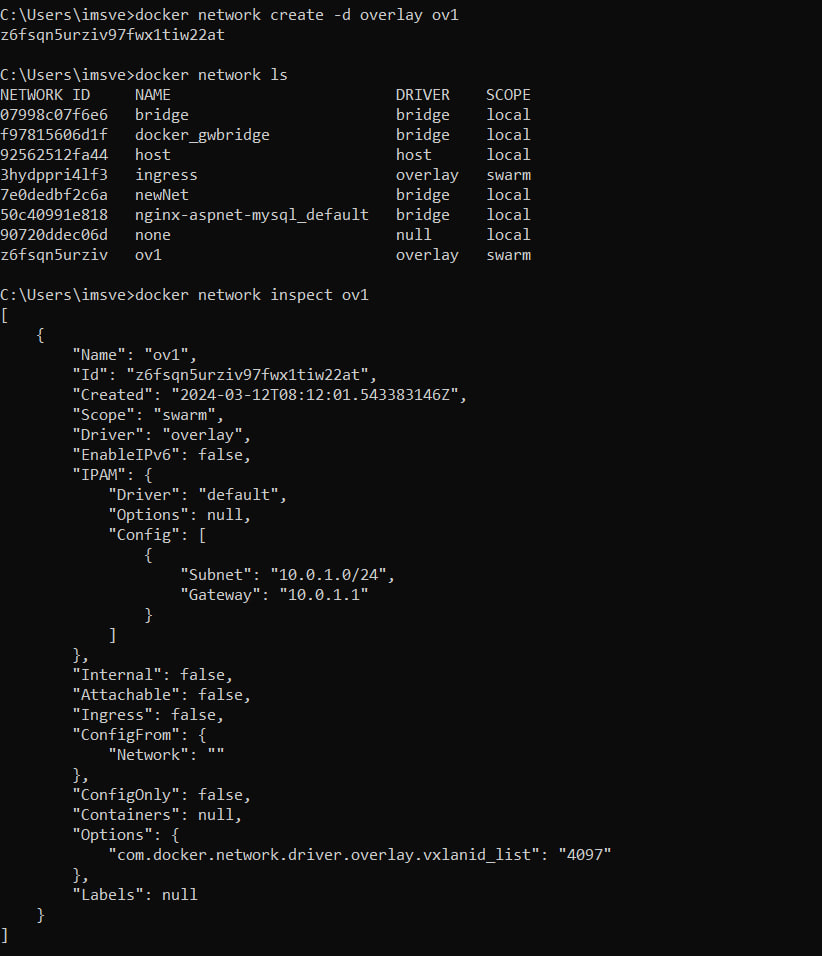
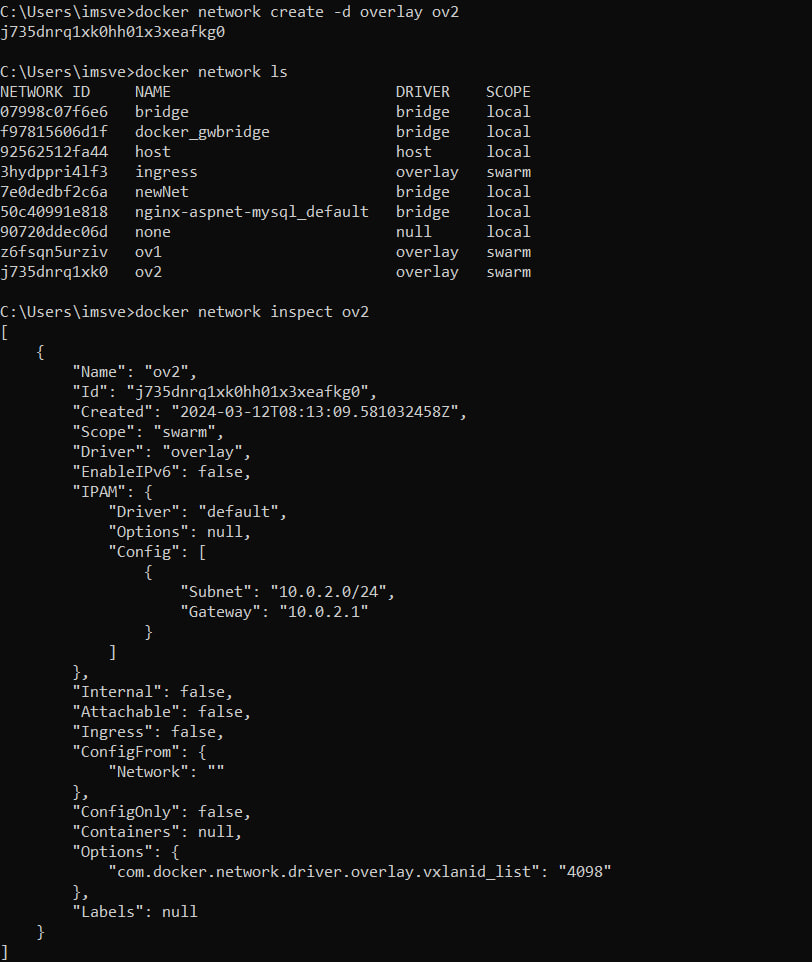
ИГИ - ЛАБОРАТОРНАЯ РАБОТА 2

1. Подготовьте рабочее окружение в соответствии с типом вашей операционной системы:  
   
2. Изучите простейшие консольные команды и возможности Docker Desktop (см. лекцию), создать собственный контейнер docker/getting-started, открыть в браузере и изучить tutorial:  
   
3. Создайте docker image, который запускает скрипт с использованием функций из <https://github.com/smartiqaorg/geometric_lib>. Данные необходимые для работы скрипта передайте любым удобным способом (например: конфиг файл через docker volume, переменные окружения, перенаправление ввода). Изучите простейшие консольные команды для работы с docker(см. лекцию). Зарегистрируйтесь на DockerHub и выберите необходимые для проекта образы. Создать Dockerfile для реализации сборки собственных Docker образов. Использовать его для создания контейнера. Протестировать использование контейнера.  
   
4. 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>):  
     
   Проект с GitHub с произвольным стеком технологий: <https://github.com/docker/awesome-compose>Создаем контейнер через Docker Compose:   
     
   Запущенные контейнеры:  
   Образы:  
   Тэгаем образы одного проекта одним названием с разными тегами (Задание 6):  
   И пушим их в репозиторий DockerHub (Задание 6):  
   
5. Настроить сети и тома для обеспечения связи между контейнерами и сохранения данных (исходные данные, логин, пароль и т.д.):  
   
6. Разместите результат в созданный репозиторий в DockerHub:  
   См. Выше (задание 4).
7. Получить информацию о всех сетях, работающих на текущем хосте и подробности о каждом типе сети:  
   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:  
     
   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:  
     
     
   Создать свою собственную сеть overlay, проверить, создана ли она, вывести о ней всю информацию:  
     
     
   Создать еще одну сеть overlay, проверить, создана ли она, вывести о ней всю информацию, удалить сеть:  
     
     
   Попробовать создать сеть host, сохранить результат в отчет:  
   