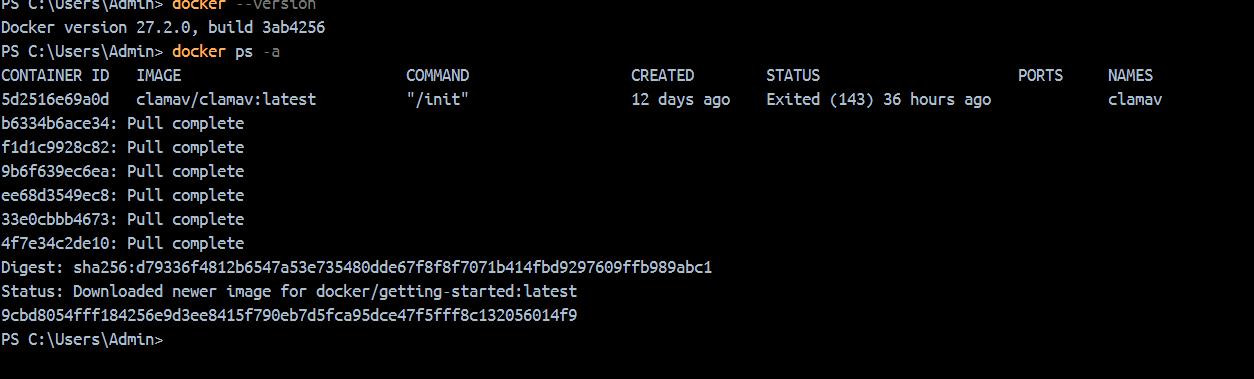
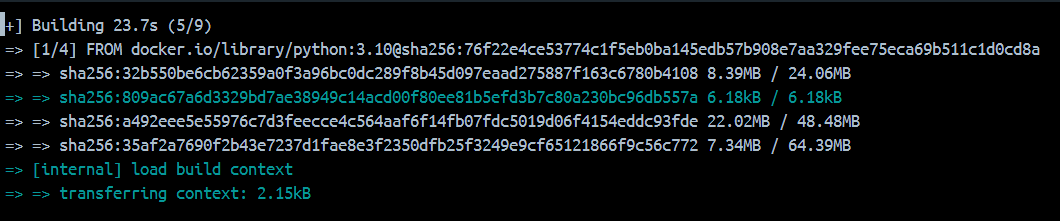
Васьковский Захар 353504 – Doker

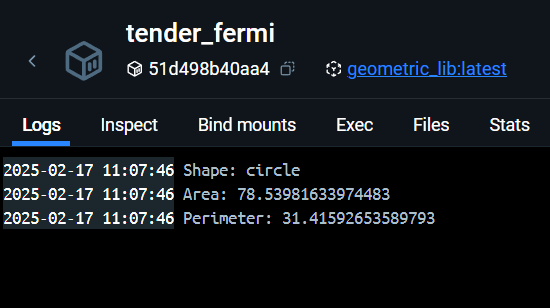
2. Заберем контейнер docker getting started и запустим н внутреннем 80 порту в фоновом режиме



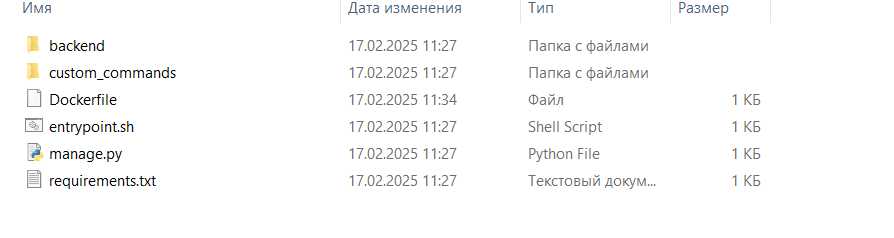
3.1 Далее создадим образ



3.2 Запустим контейнер, взяв необходимые данные из config.json, в котором будут храниться значения переменных, docker run --rm geometric\_lib

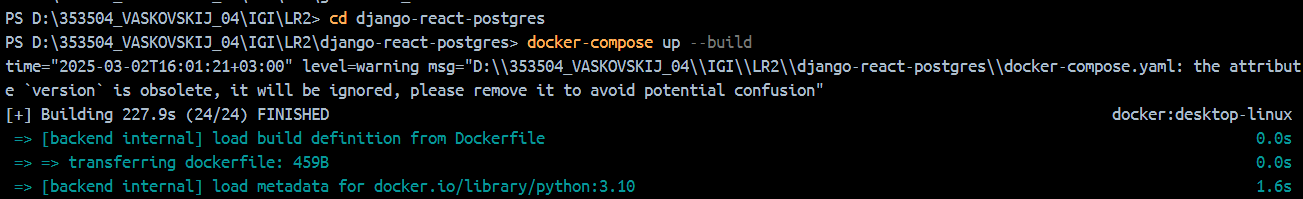


4.1 Затем найдем на github базовый проект, со стеком технологий, соответвующий индивидуальному заданию и клонируем в директорию с лабораторной работой, далее добавляем dockerfiles для frontend и backend – проектов

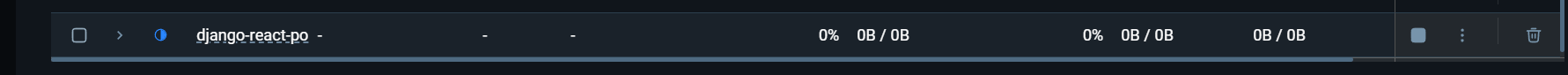




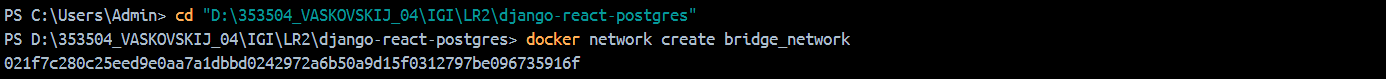
4.1 Затем создадим docker-compose файл и перейдя в директорию с ним, соберем образ:



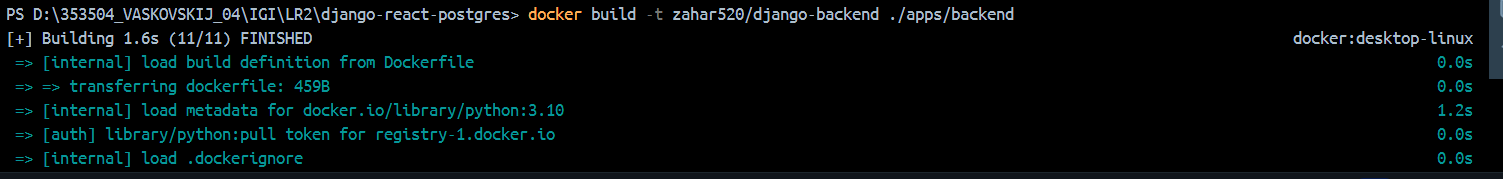
Теперь можно найти во вкладке containers наш compose-stack



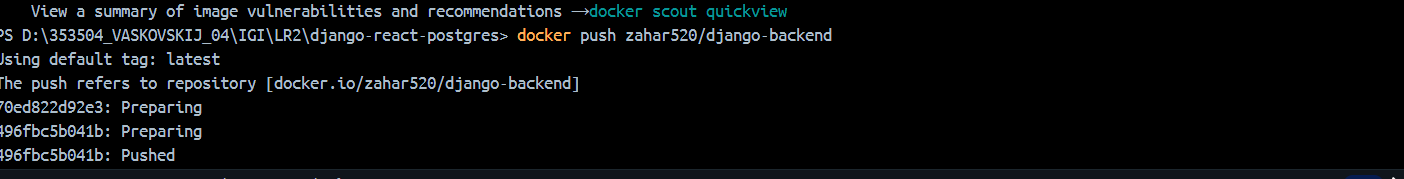
5 Затем создадим сеть



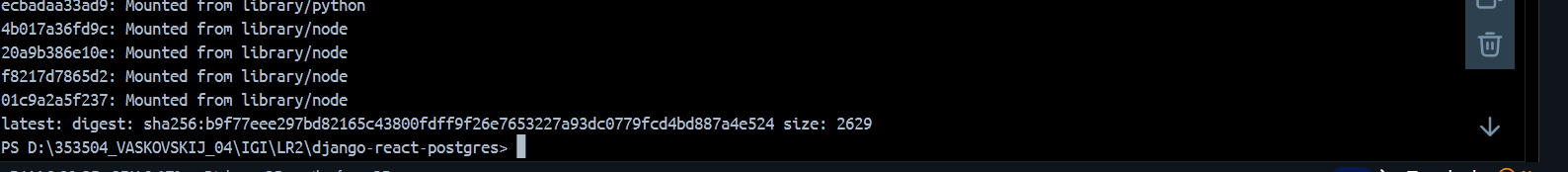
6. Затем соберем образ для каждого из наших контейнеров



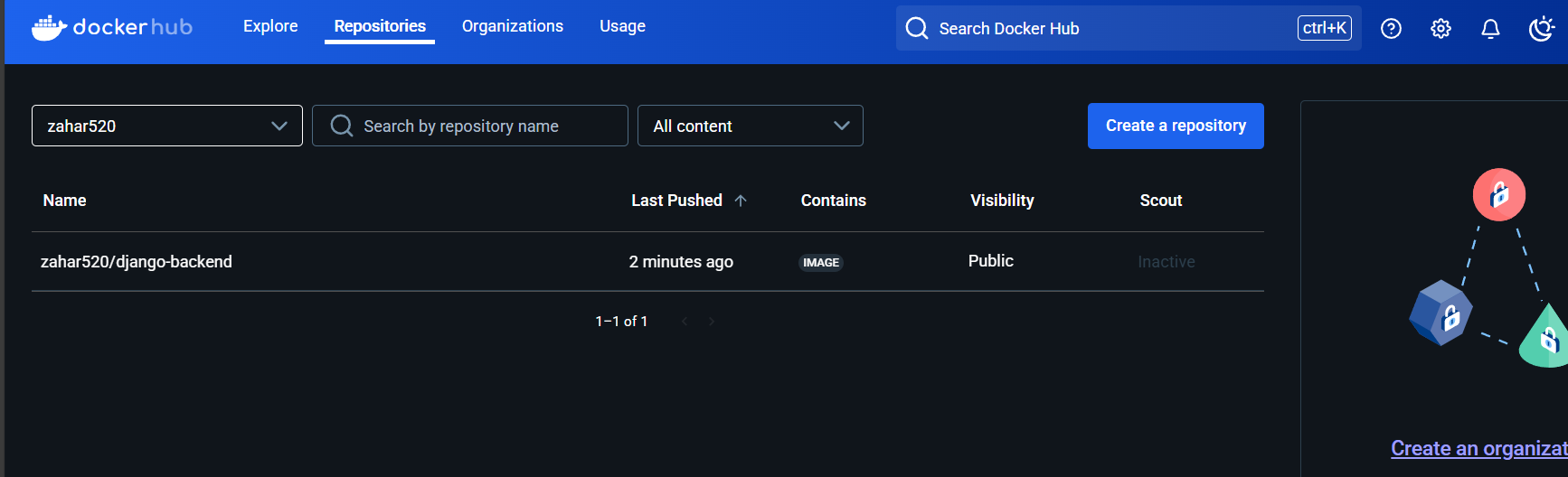
И отправим их в dockerHub



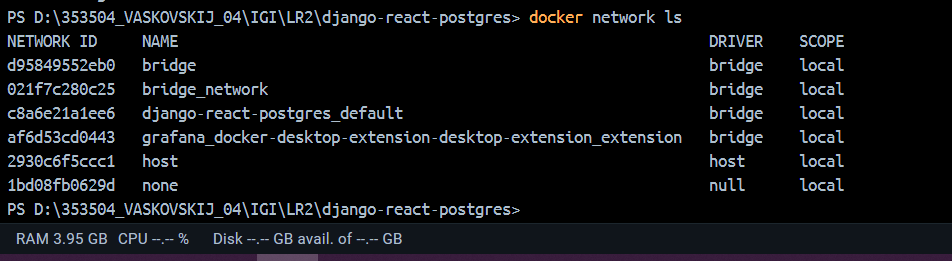
Статус pushed, говорит, что контейнеры запушили успешно



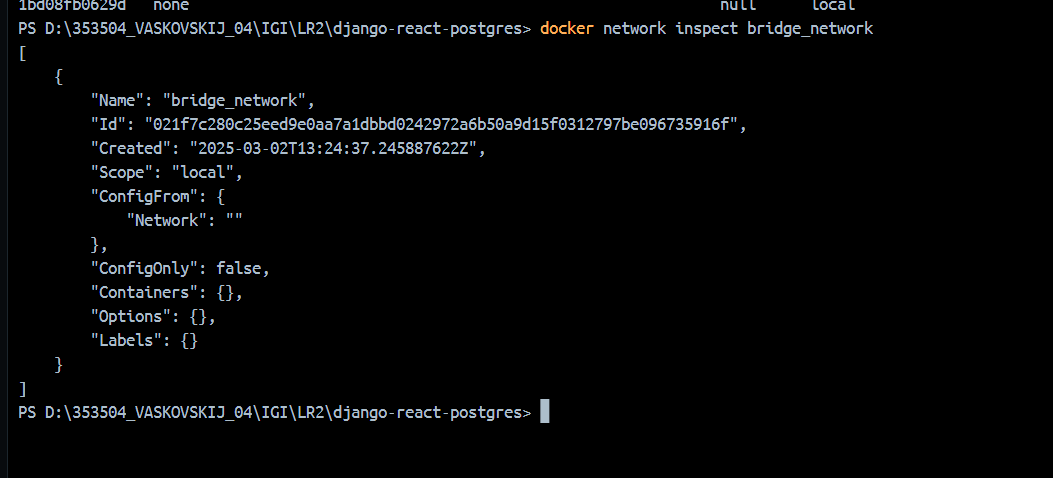
Теперь контейнеры можно найти на dockerhub



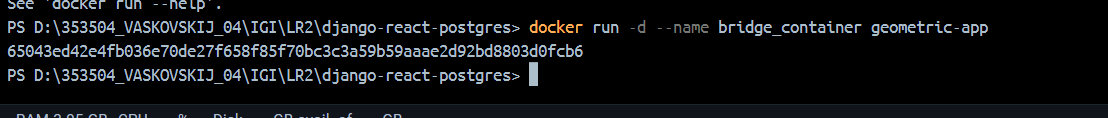
7.1 Получим информацию о всех сетях, работающих на текущем хосте:



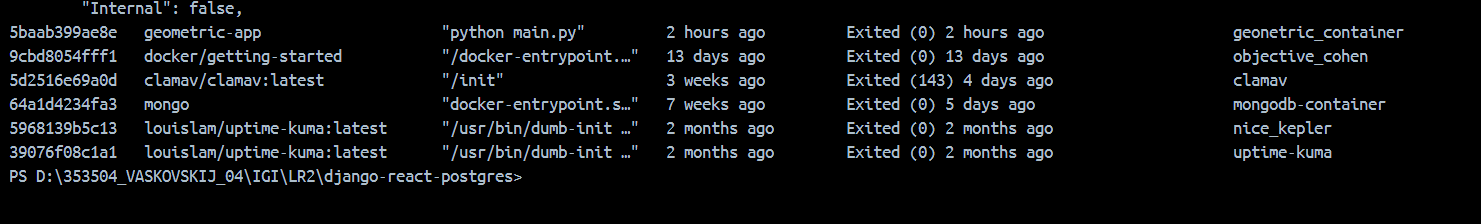
И можно получить развернутую информацию о конкретной сети (например bridge\_network)



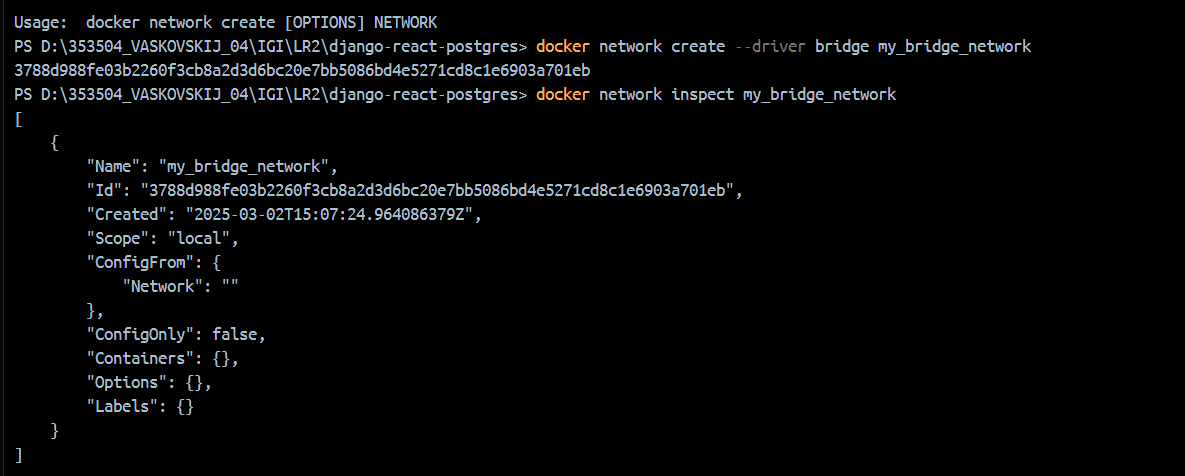
7.2 Создадим сеть bridge и запустим в ней контейнер geometric\_lib



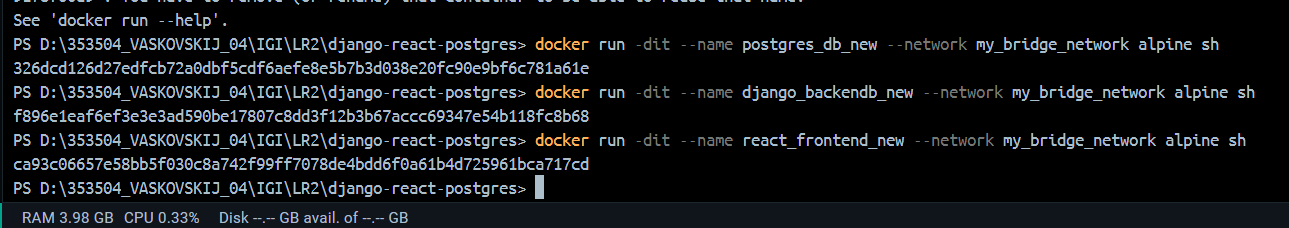
Как видно, контейнер запустился в фоновом режиме и его можно найти через docker ps –a



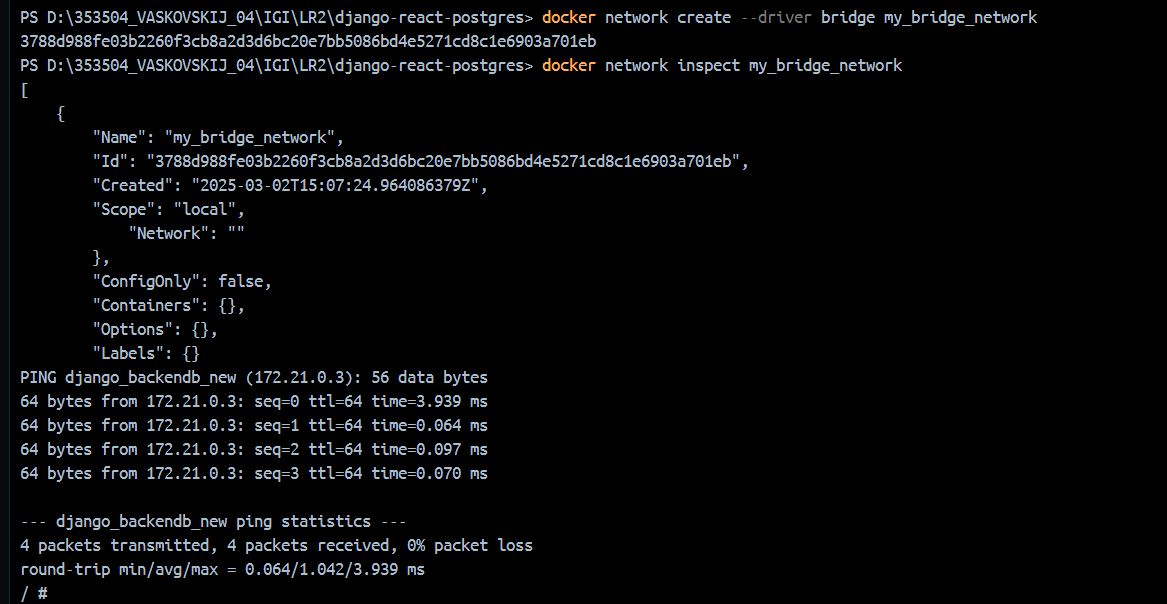
7.3 Создадим еще одну сеть и выведем информацию о ней



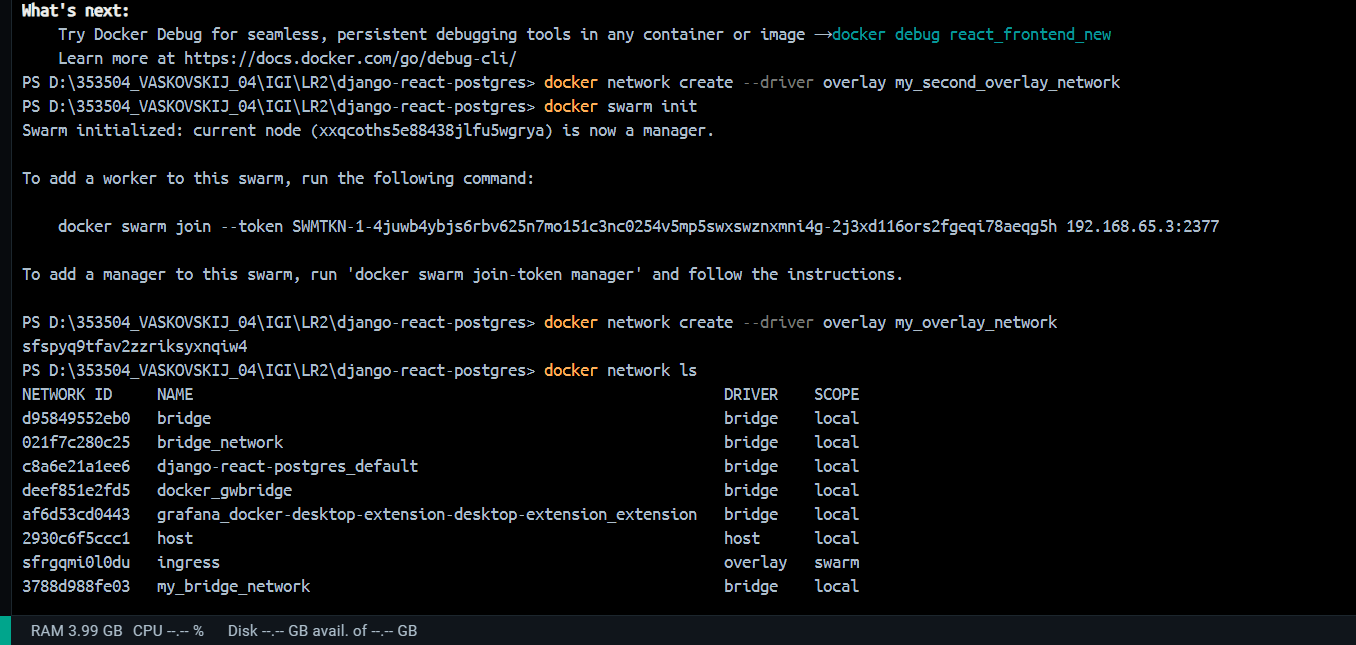
Теперь запустим три контейнера в этой сети



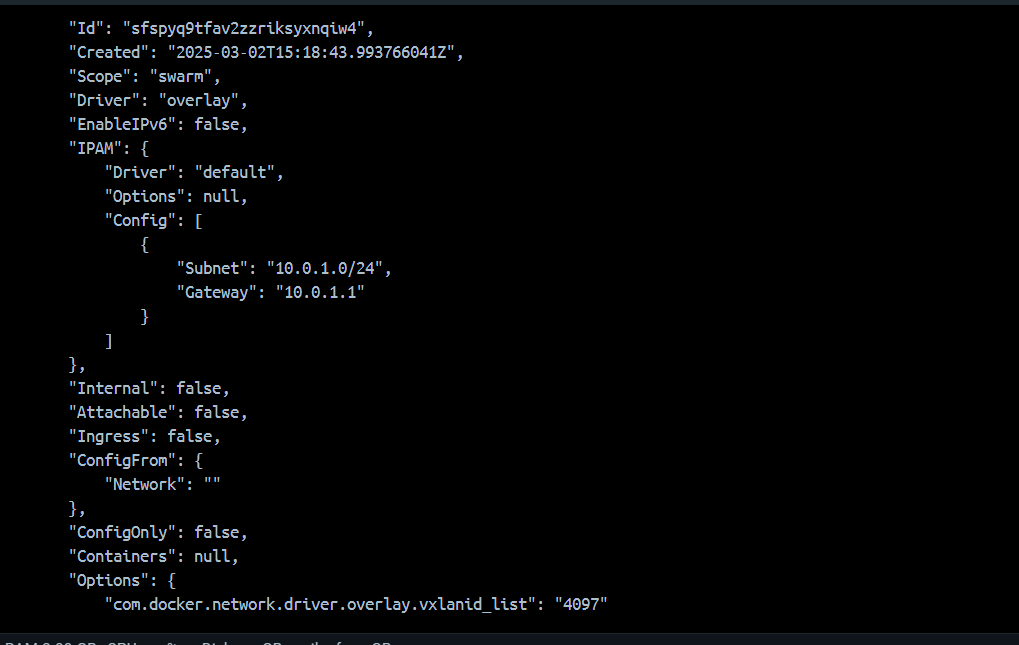
Затем подключимся к одному из них и пропингуем другой



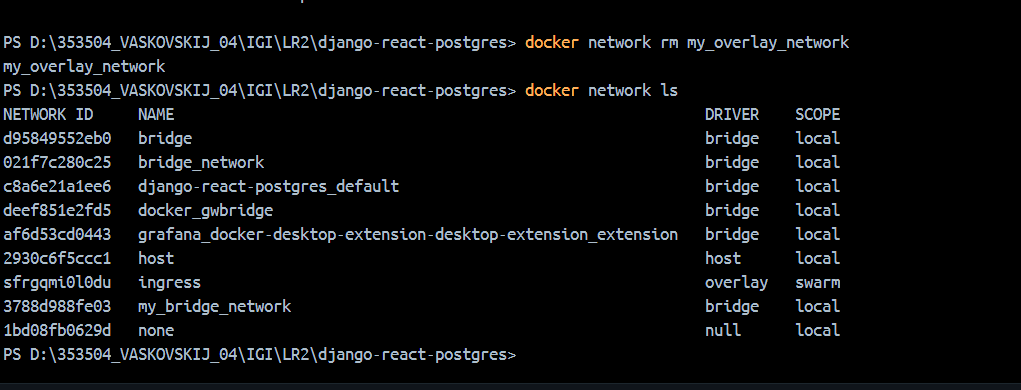
7.4 Далее создадим новую сеть overlay и выведем о ней информацию



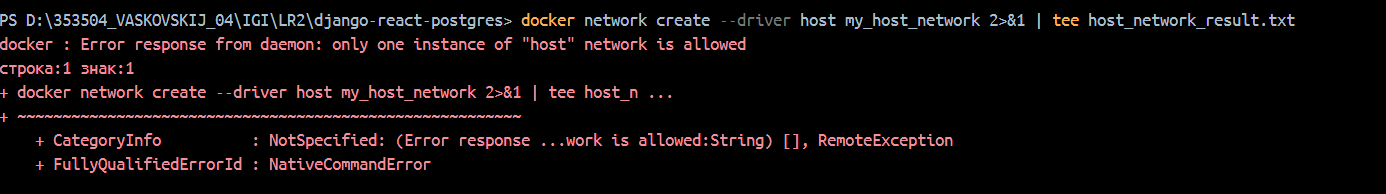
Вывод подробной информации



7.5 Затем удалим эту сеть:



7.6 Пробуем создать сеть host



Как видно из логов, попытка не удалась, т к сеть host уже существует на локальной машине