После установки приложения Docker Compose и изучения базовых команд, был создан docker контейнер docker/getting-started (рис.1).

**Рисунок 1**

Для запуска проекта geometric\_lib создан файл main.py с использованием функций из репозитория, затем Dockerfile, где прописаны инструкции к запуску контейнера. Сборка контейнера и его запуск показан на рисунке 3.



Рисунок 3 – сборка и запуск контейнера для geometric\_lib

Проверить, запущены ли проекты, можно через Docker Desktop (рис. 6)

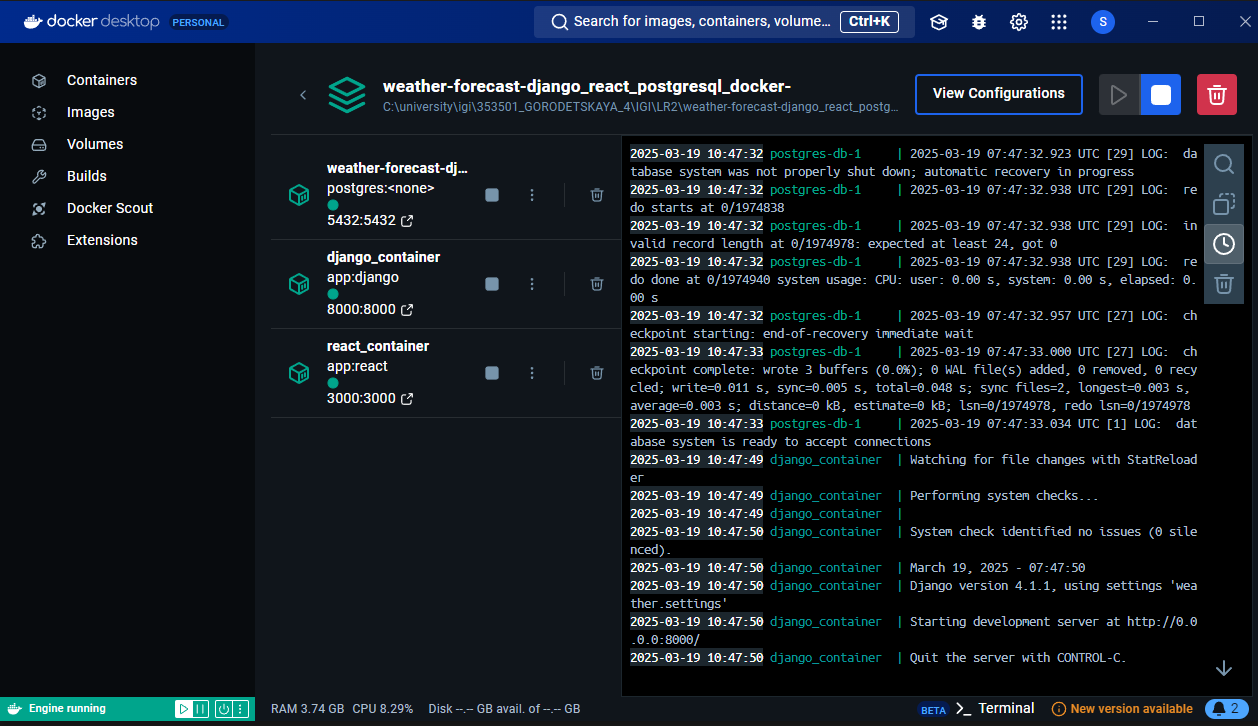
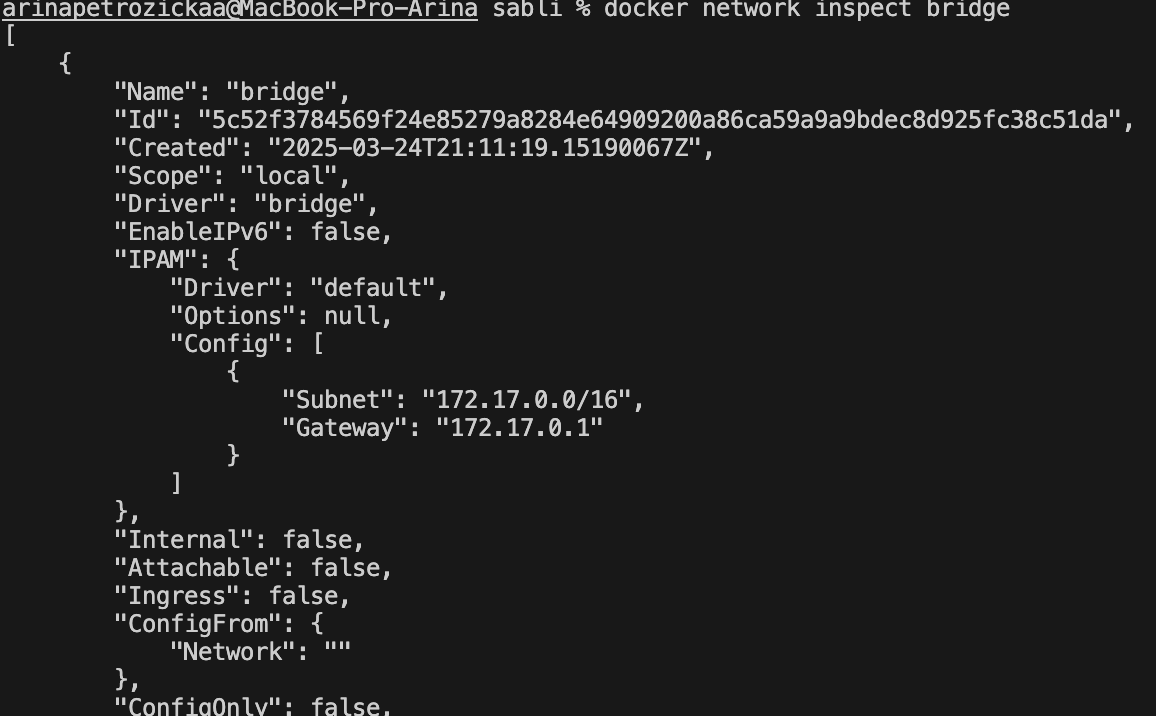
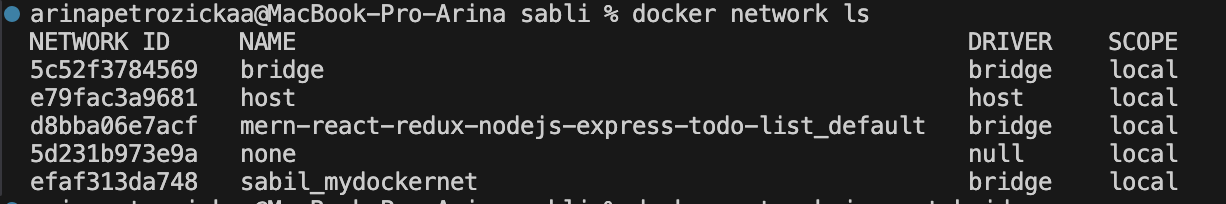
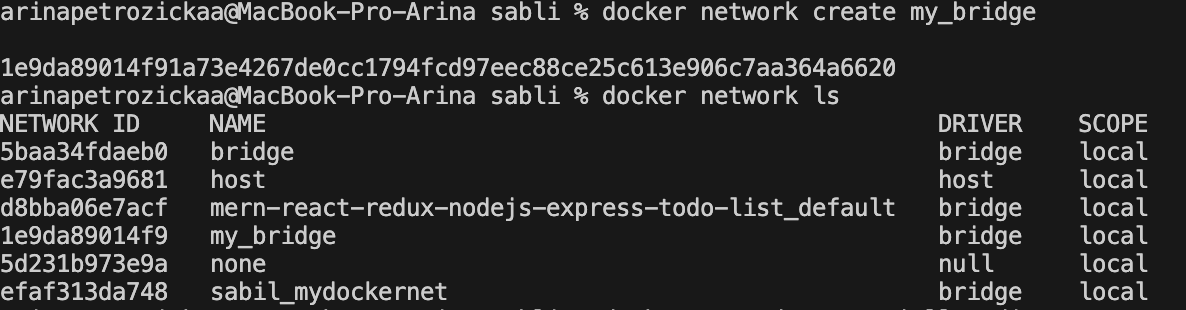


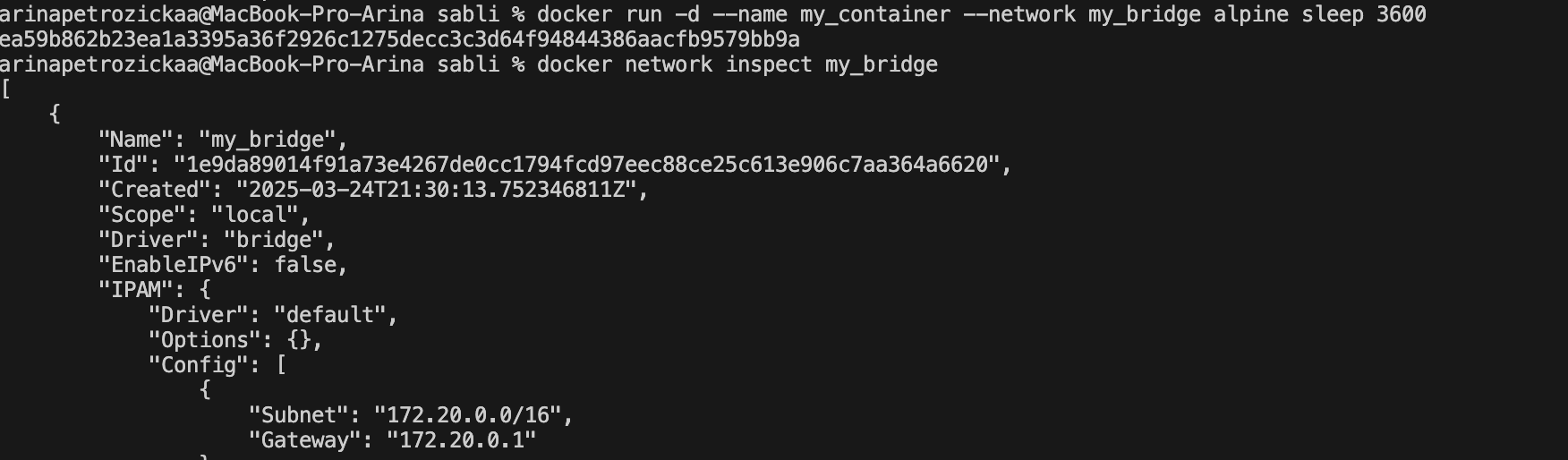
Рисунок 6 – запущенные контейнеры в Docker Desktop

7.1. docker network ls команда выведет список всех сетей Docker, работающих на текущем хосте. Для получения подробной информации о конкретной сети (например, bridge): docker network inspect bridge



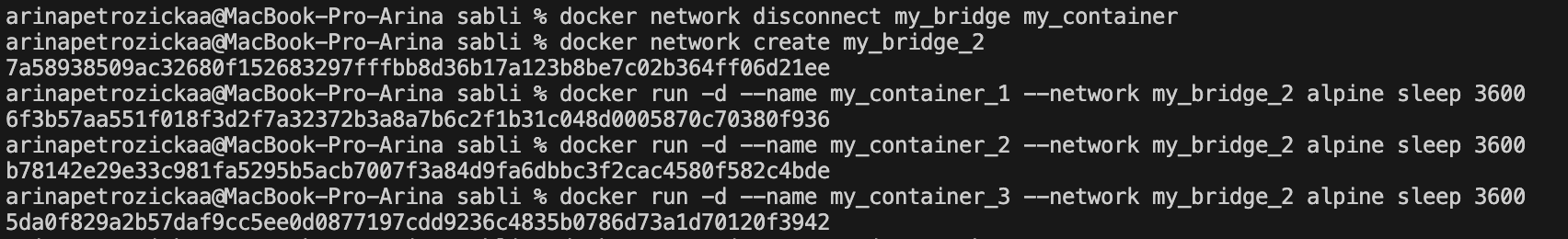
7.2. Создать свою сеть bridge, запустить контейнер в ней и проверить

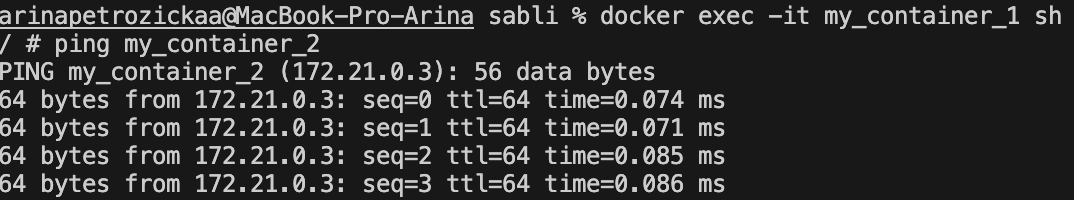
Создание bridge

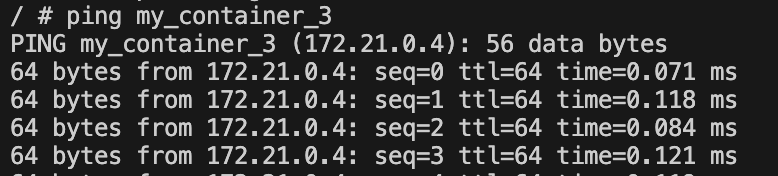
Запуск контейнера в созданной сети и получение информации о нём

Команда docker network disconnect my\_bridge\_network my\_container отключает контейнер от сети.

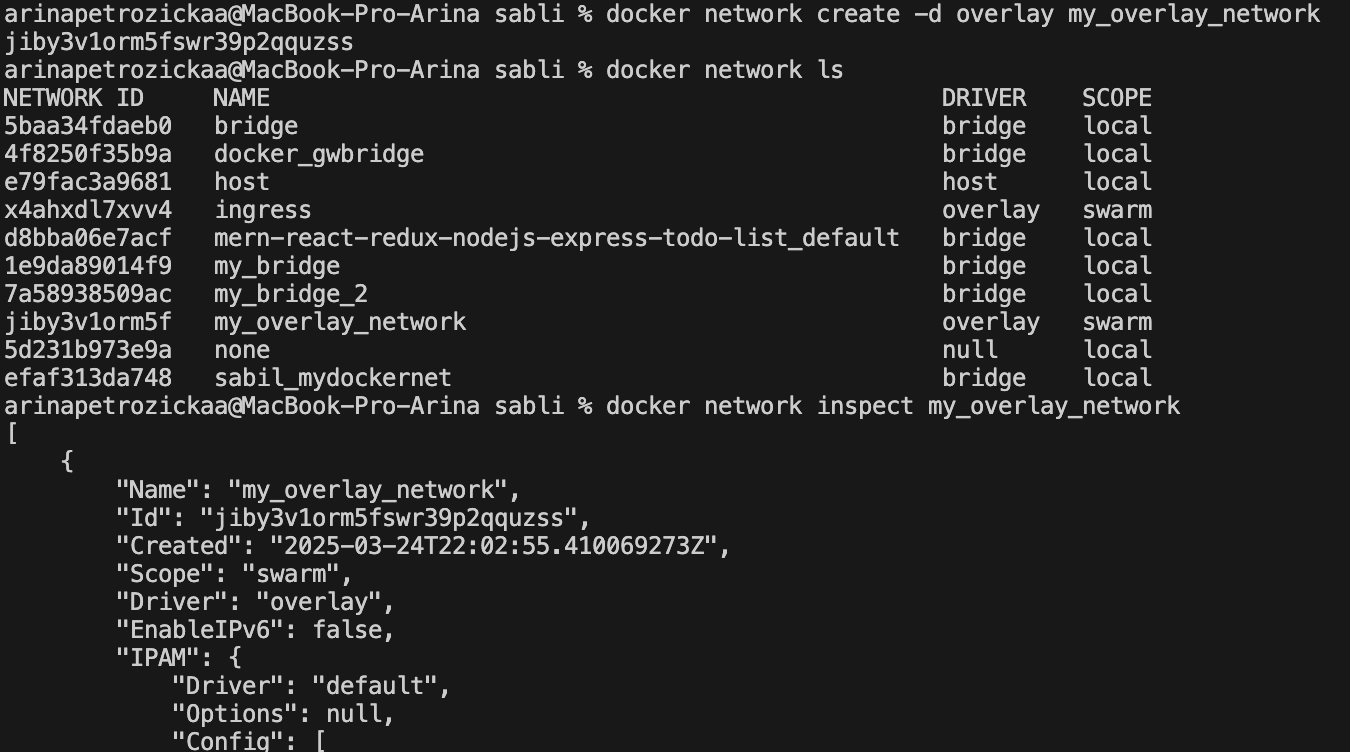
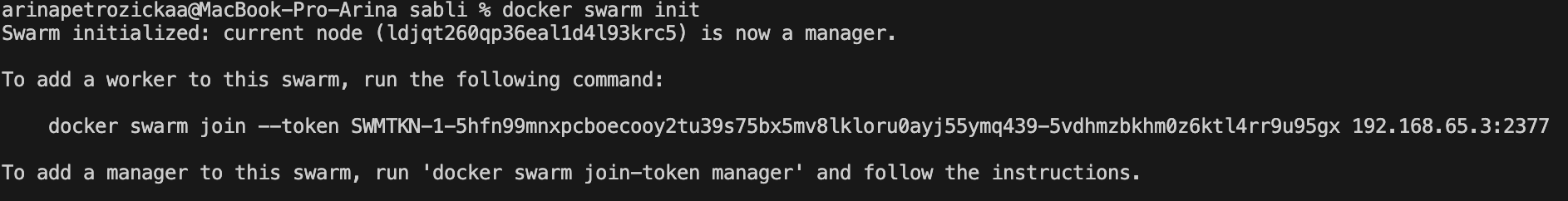
7.3. Создать еще одну сеть bridge, запустить три контейнера и проверить связь

Создание второй сети bridge и запуск 3 контейнеров внутри неё

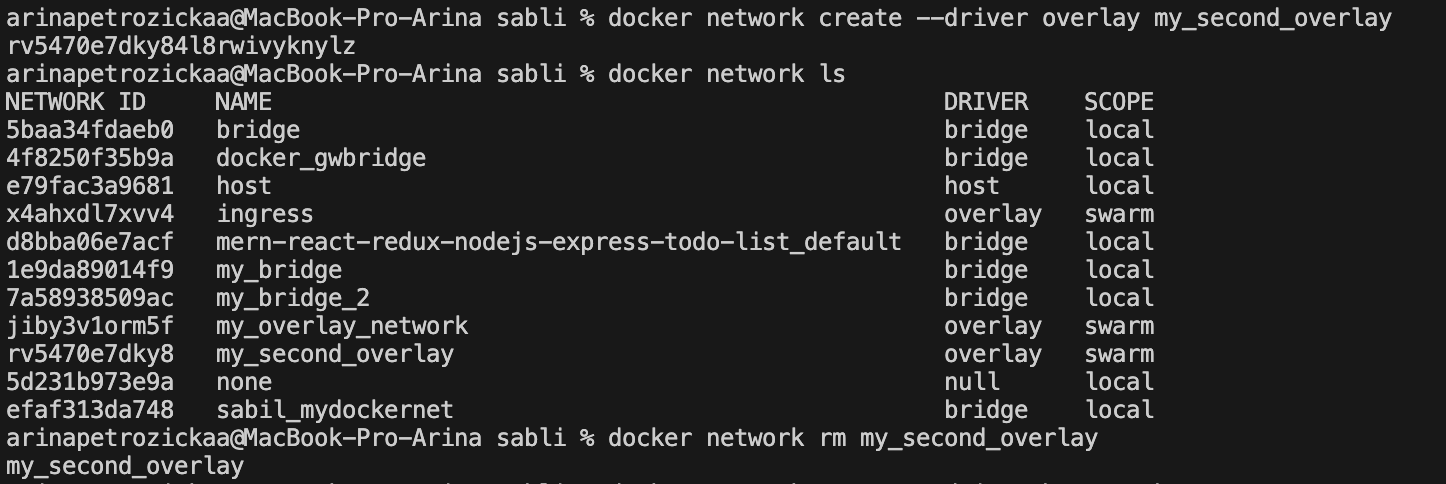
Подключение к первому контейнеру и пингование других из оболочки первого

****

7.4. Создание сети overlay и проверка



7.5 Создать еще одну сеть overlay, проверить и удалить



7.6. Создание сети host. На Windows эта команда не сработает, так как сеть host поддерживается только на Linux.