

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**МИРЭА** – **Российский технологический университет**»

**РТУ МИРЭА**

**Институт комплексной безопасности и цифровых технологий (ИКБ) Кафедра КБ-14 «Цифровые технологии обработки данных»**

**Администрирование баз данных**

**Практическая работа №2**

Выполнили студенты 3 курса

Зарин Н.Н.

Луговой И.И.

 БСБО-05-20

Москва 2023

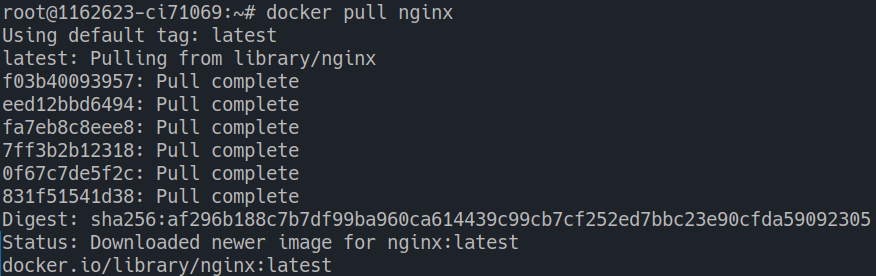
Для выполнения задания был использован удаленный арендованный сервер.

**Задание 1.**

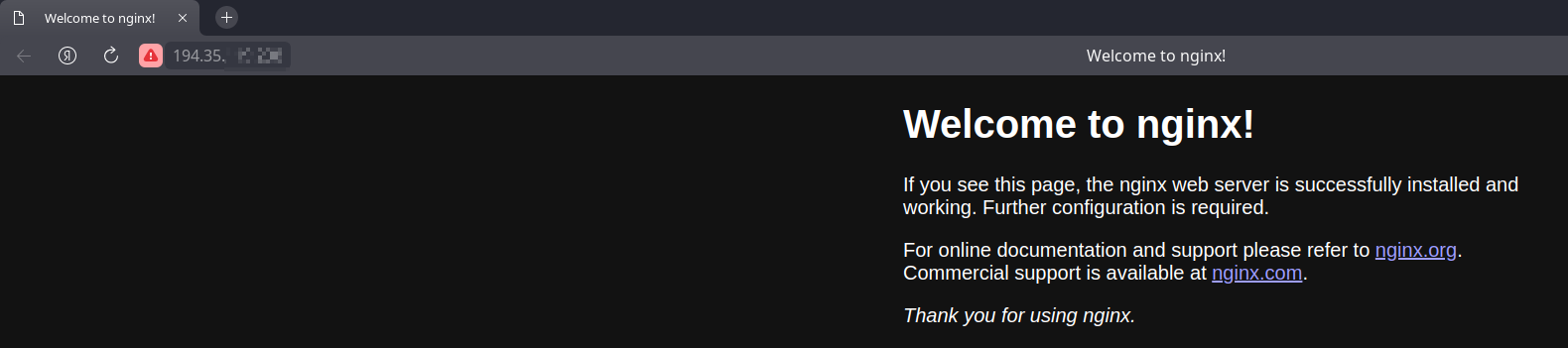
1. Установите Docker на локальной машине.



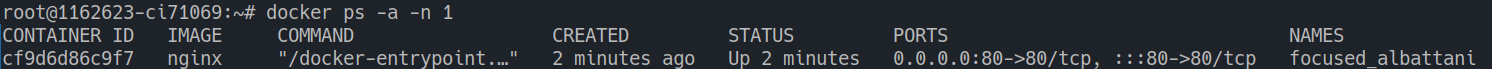
1. Используя Docker Hub, загрузите образ Nginx



1. Запустите контейнер Nginx в фоновом режиме



1. Вывести список работающих контейнеров



1. Остановите действующий контейнер



1. Переименуйте остановленный контейнер на имя webhost



1. Заново запустите контейнер с именем webhost

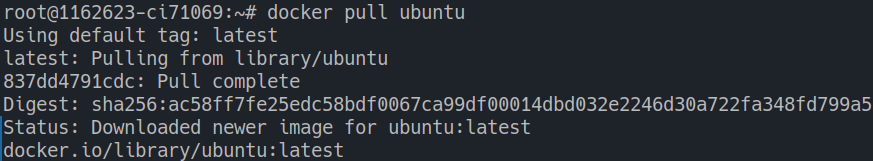


1. Выведите логи контейнера webhost



**Задание 2.**

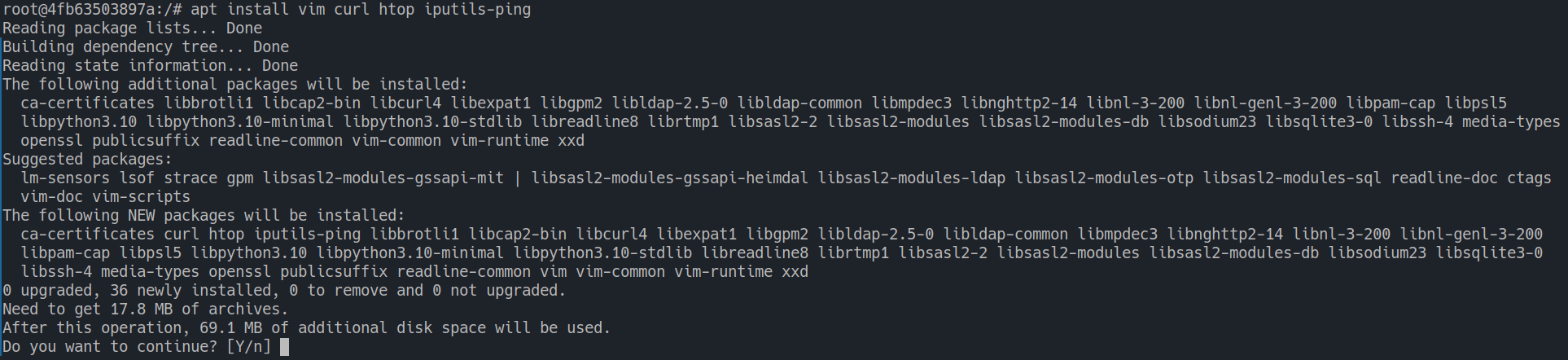
1. Загрузите образ Ubuntu из Docker Hub.



1. Запустите контейнер с Ubuntu из загруженного образа и подключитесь к нему с помощью команды docker exec -it.



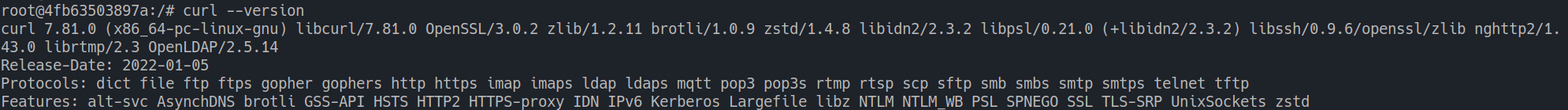
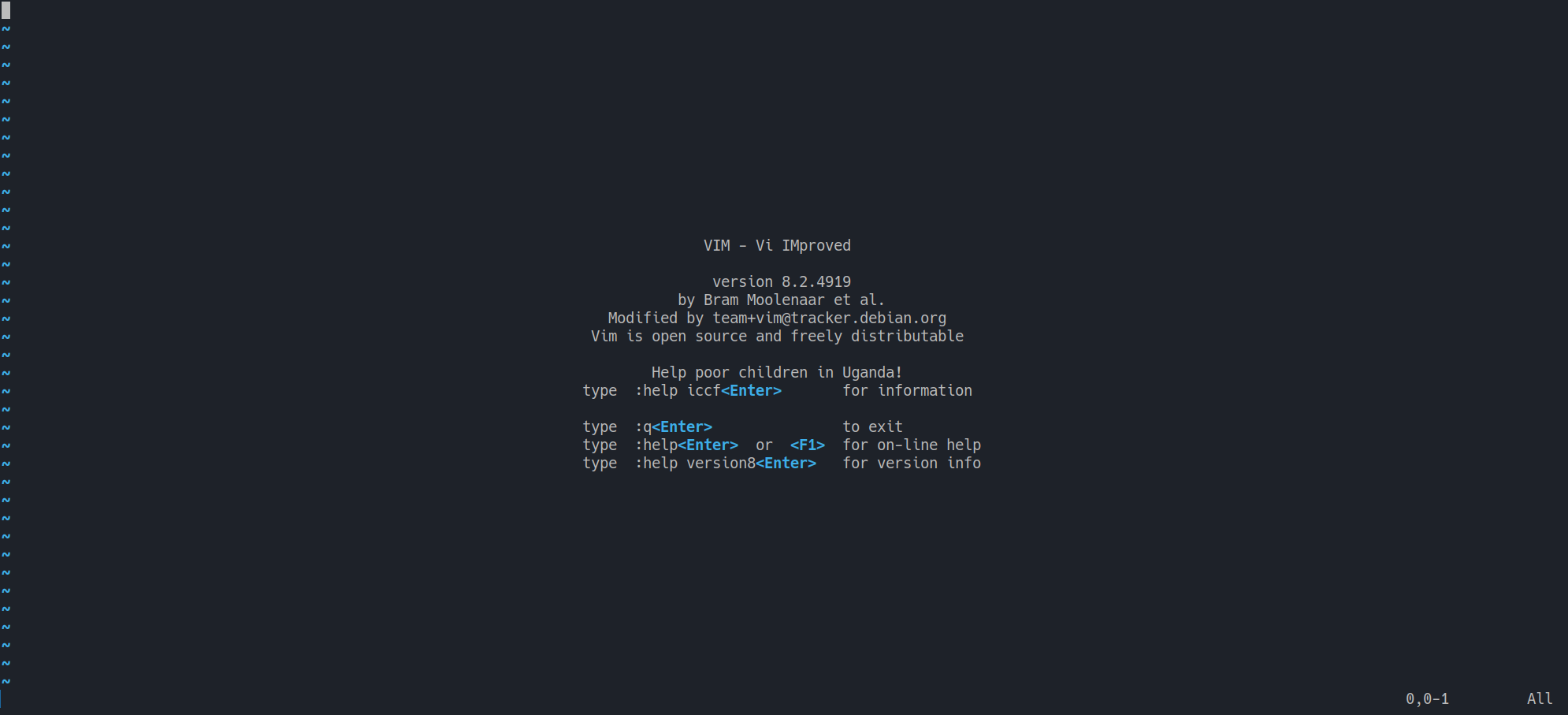
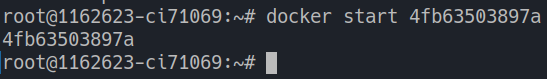
1. Используя команды Ubuntu, установите в контейнере несколько пакетов:, vim, curl, htop, ping.



1. Остановите контейнер



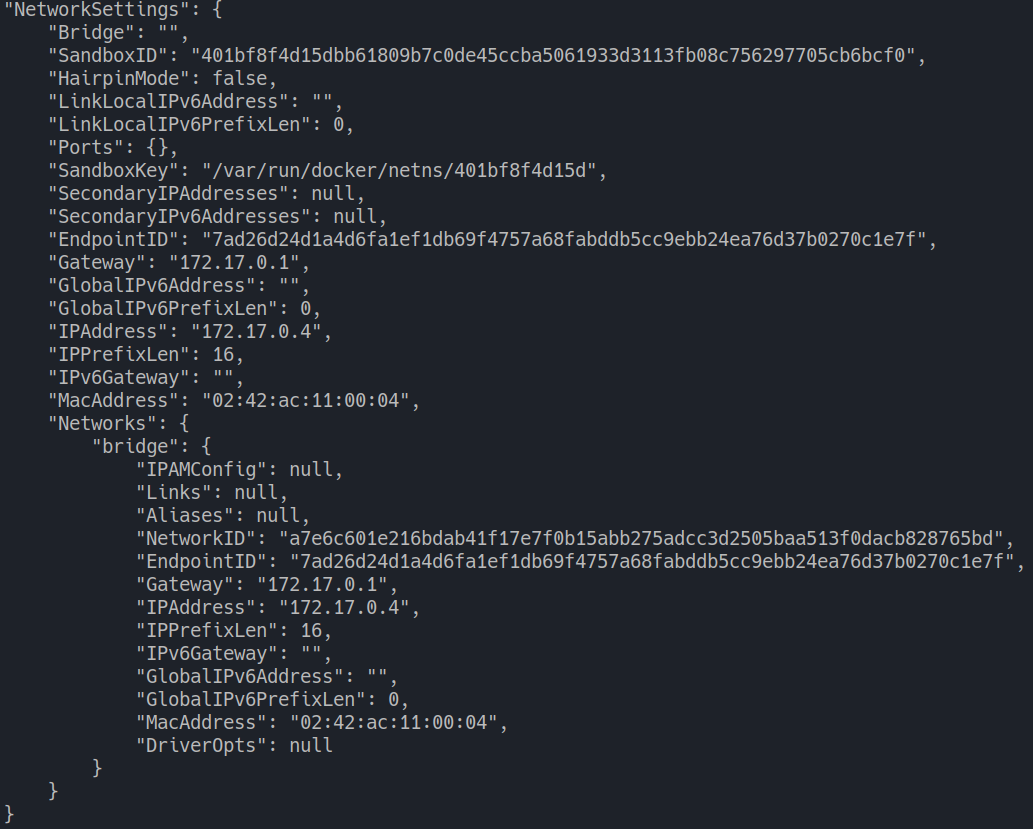
1. Запустите его заново в интерактивном режиме и убедитесь что все пакеты установлены



**Задание 3.**

1. Выведите информацию о сети каждого существующего контейнера

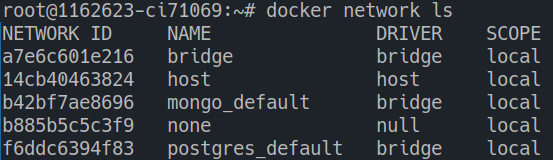
Ubuntu:



Nginx:



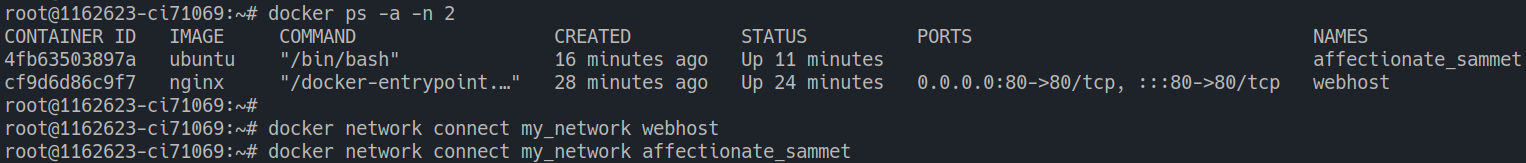
1. Выведите все существующие сети в docker



1. Создайте свою собственную сеть типа Bridge

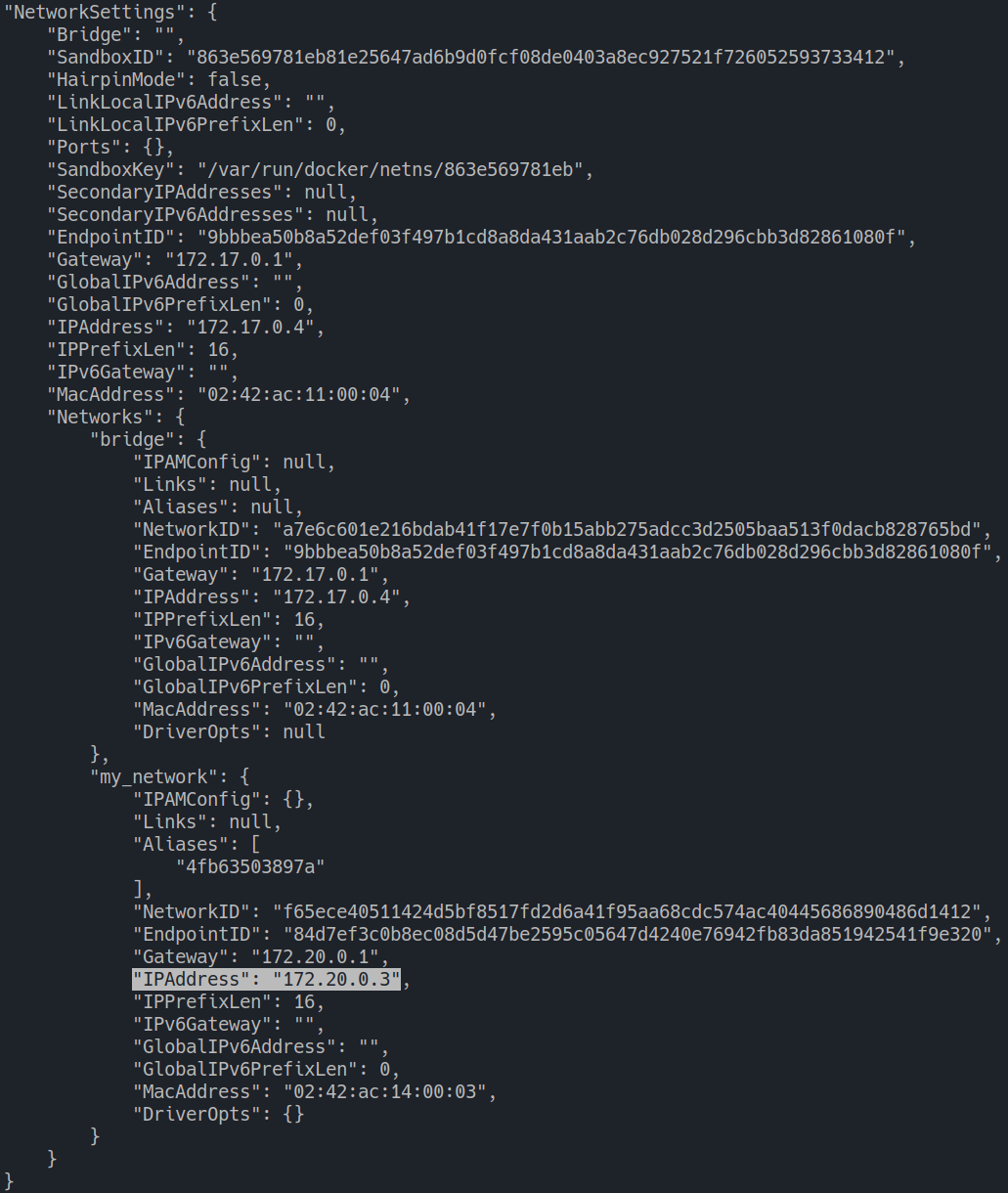


1. Подключите туда два созданных контейнера

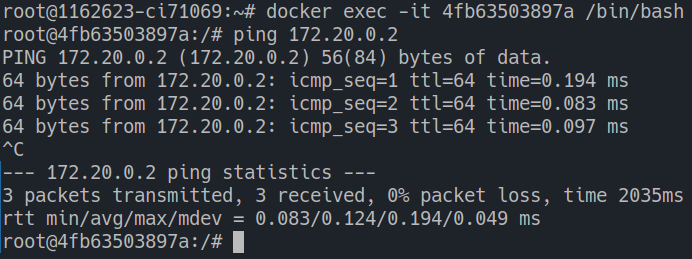


1. При помощи утилиты ping проверьте что контейнеры видят друг друга

Ubuntu:

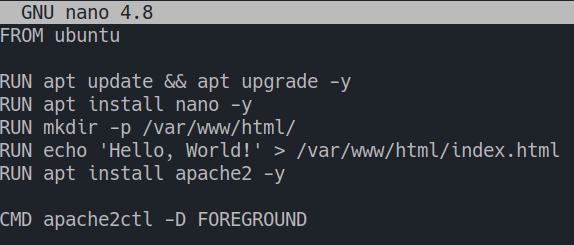


Nginx:

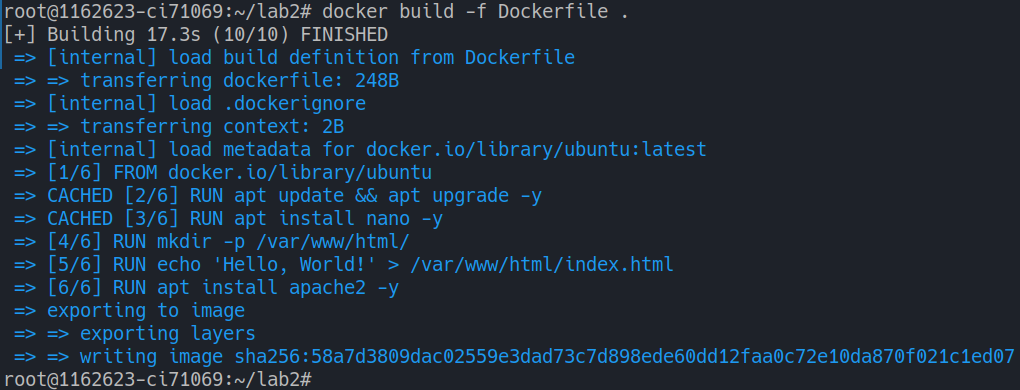


**Задание 4.**

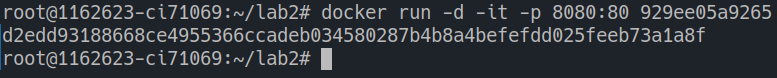
1. Создайте Dockerfile:
   1. Задайте свой образ на основе исходного образа Ubuntu
   2. Установите пакет nano
   3. Запустите команду "echo 'Hello, World!' > index.html" внутри контейнера
   4. Задайте рабочую директорию /usr/local/apache2/htdocs/
   5. Копируйте файл index.html в указанную директорию
   6. Установите пакет apache2
   7. Запустите Apache в foreground mode (чтобы контейнер не завершался сразу после запуска)



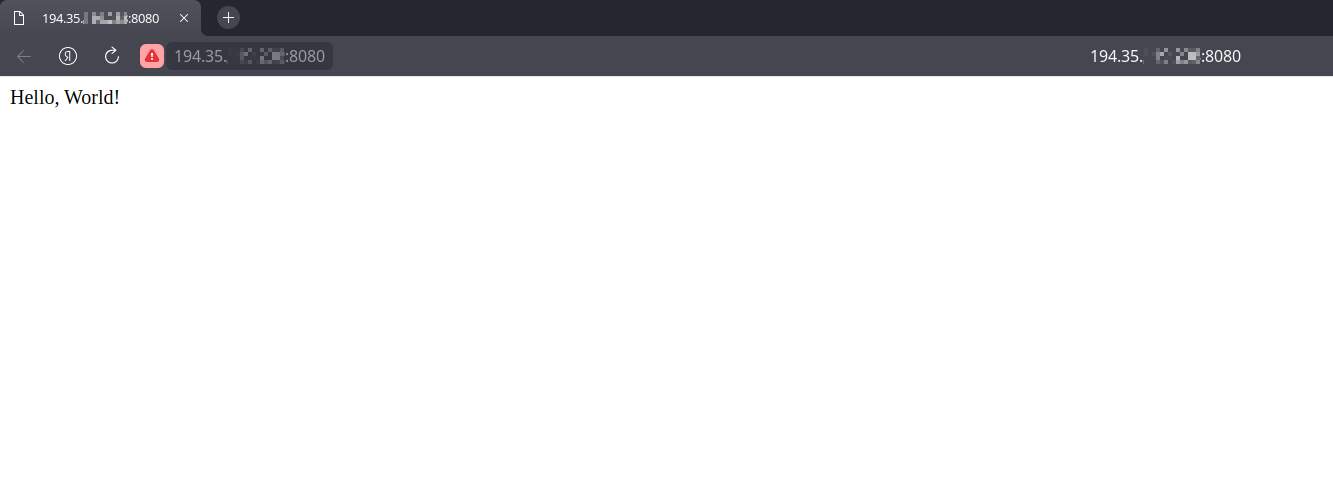
1. Соберите образ используя Dockerfile



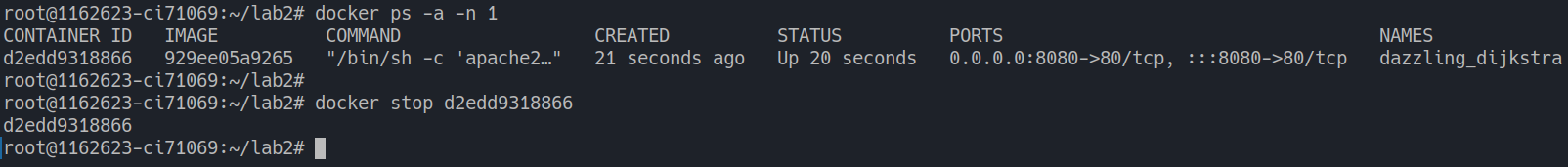
1. Запустите контейнер используя ваш образ



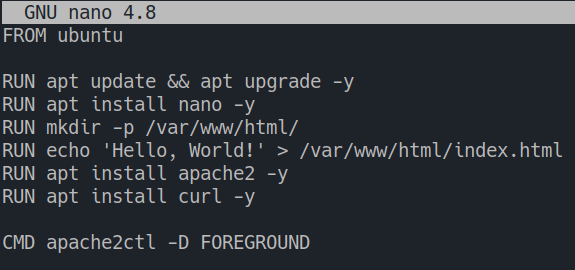
1. Проверьте, что ваш веб-сервер работает корректно, открыв http://localhost в вашем браузере



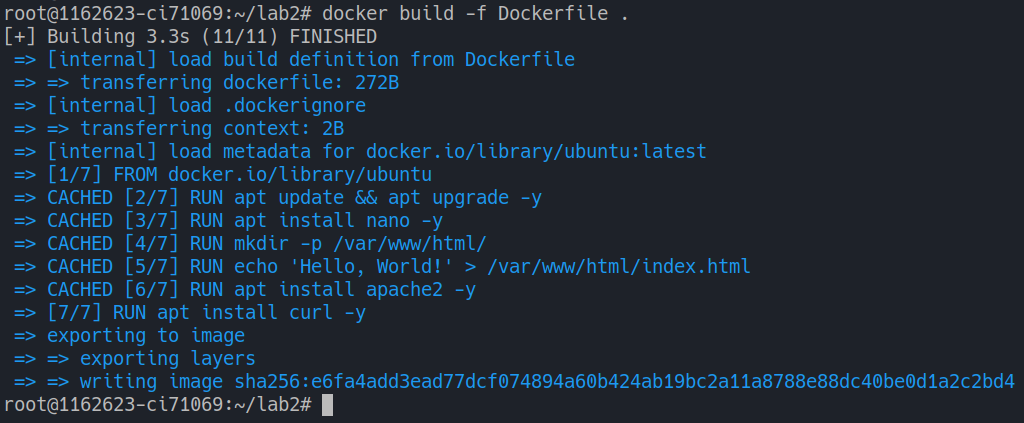
1. Остановите и удалите контейнер.



1. Добавьте в Dockerfile команду, которая устанавливает пакет curl



1. Пересоберите образ и запустите контейнер заново



1. Проверьте, что пакет curl установлен внутри контейнера

