Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

Лабораторна робота №3

з дисципліни «Проектування розподілених систем»

за темою «Розгортання веб-порталу як Docker Container»

**Виконав:**

Студент групи ІA-11мн

Новиков Данило Михайлович

**Перевірив:**

доц. Волокина Артем Миколайович

Київ 2022

**Тема:** Розгортання веб-порталу як Docker Container. Розгортання застосунок за допомогою docker-compose на віртуальному сервері (новому або з Лаб. Работи 1)

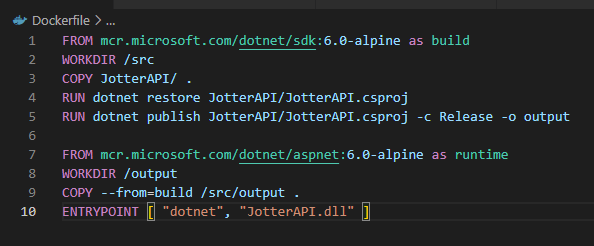
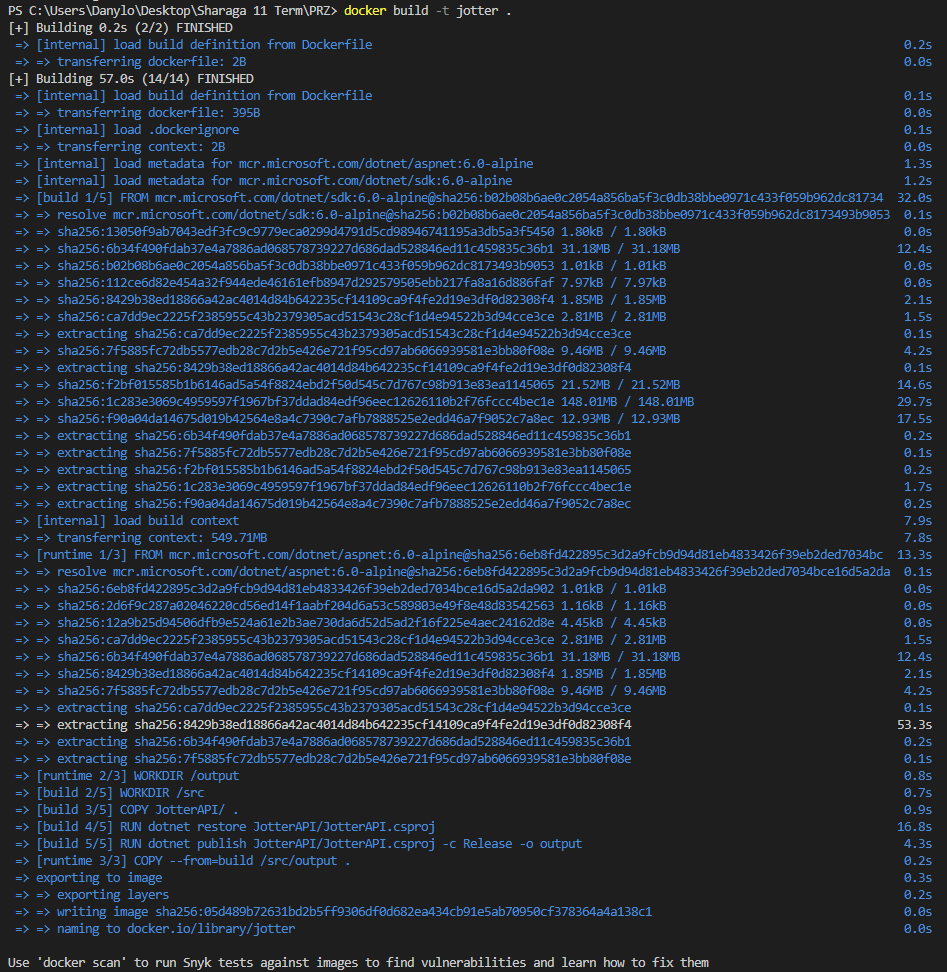
**Завдання:**

* Застосунок має бути зібраний в docker image і розгорнутий локально;
* Опціонально: Розгорнути застосунок локально за допомогою docker-compose;
* Опублікувати docker image з застосунком в публічному Container Registry (Azure Container Registry, Digital Ocean Container Registry або GitLab Container Registry);
* Розгорнути віртуальний сервер і встановити docker engine на ньому;
* Розгорнути застосунок і всі допоміжні сервіси через docker-compose використовуючи публічний Container Registry.

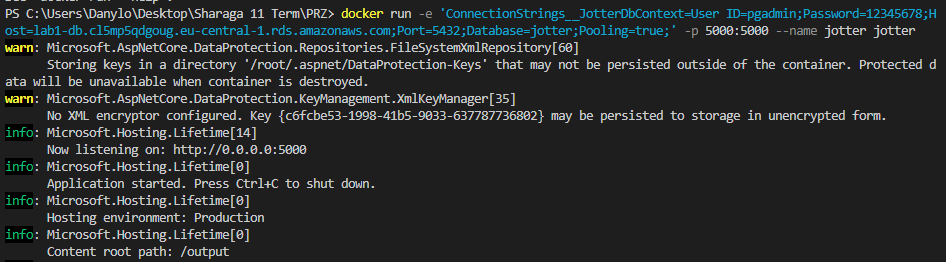
**Хід роботи**

**Репозиторій:** <https://github.com/JokerFunny/PRZ>**.**

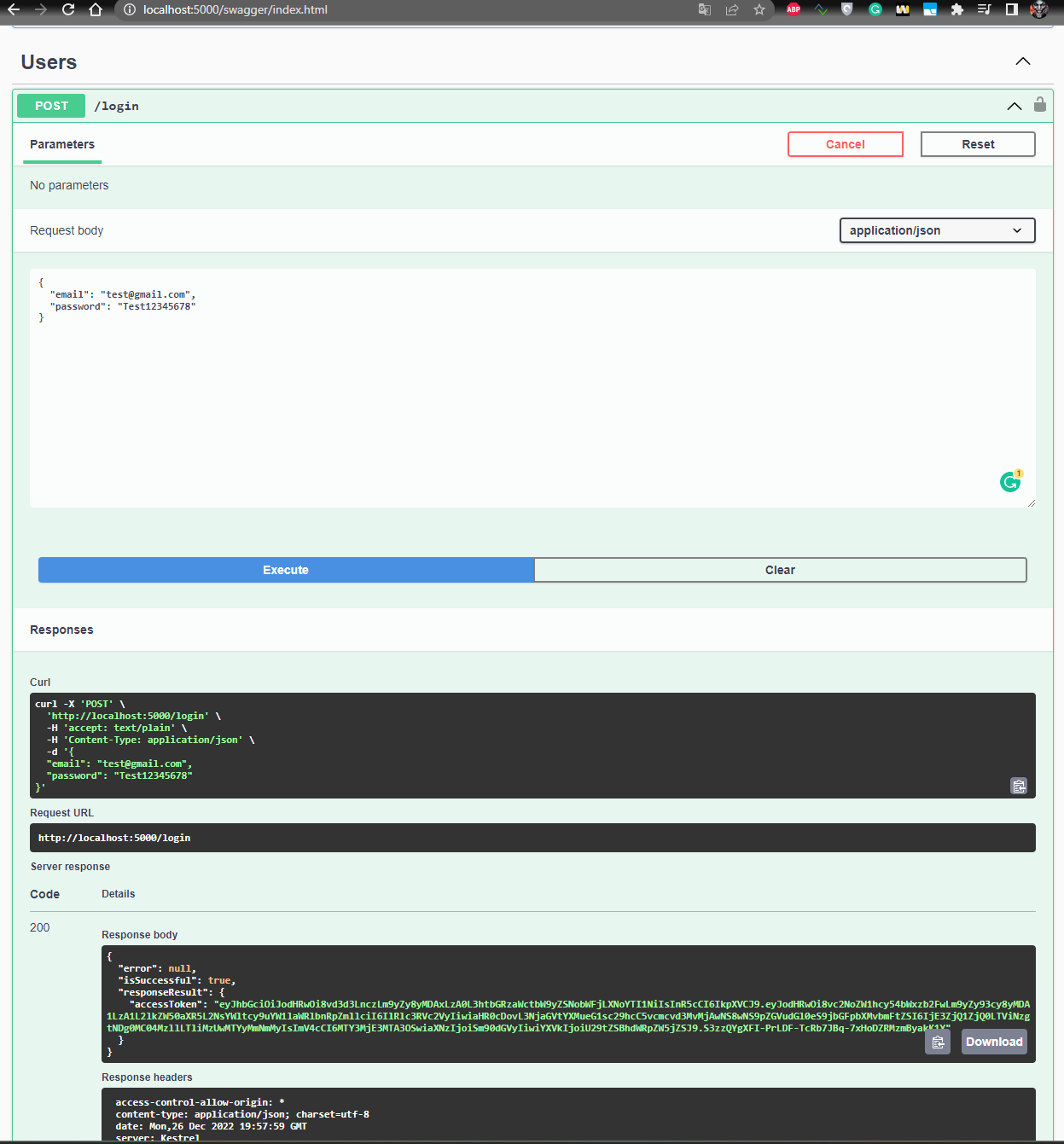
1. Створимо Dockerfile для запуску нашого додатку:

 ****

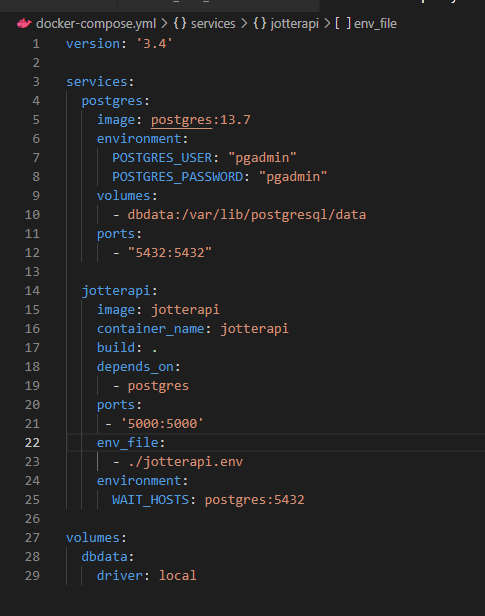
1. **docker build -t jotter .**

****

1. Перевіримо працездатність:

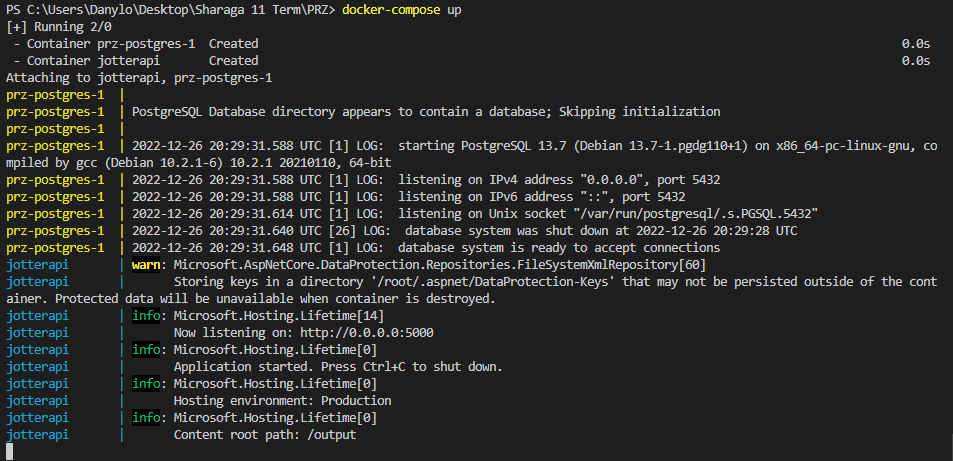
**docker run -e 'ConnectionStrings\_\_JotterDbContext=User ID=pgadmin;Password=12345678;Host=lab1-db.cl5mp5qdgoug.eu-central-1.rds.amazonaws.com;Port=5432;Database=jotter;Pooling=true;' -p 5000:5000 --name jotter jotter  
**

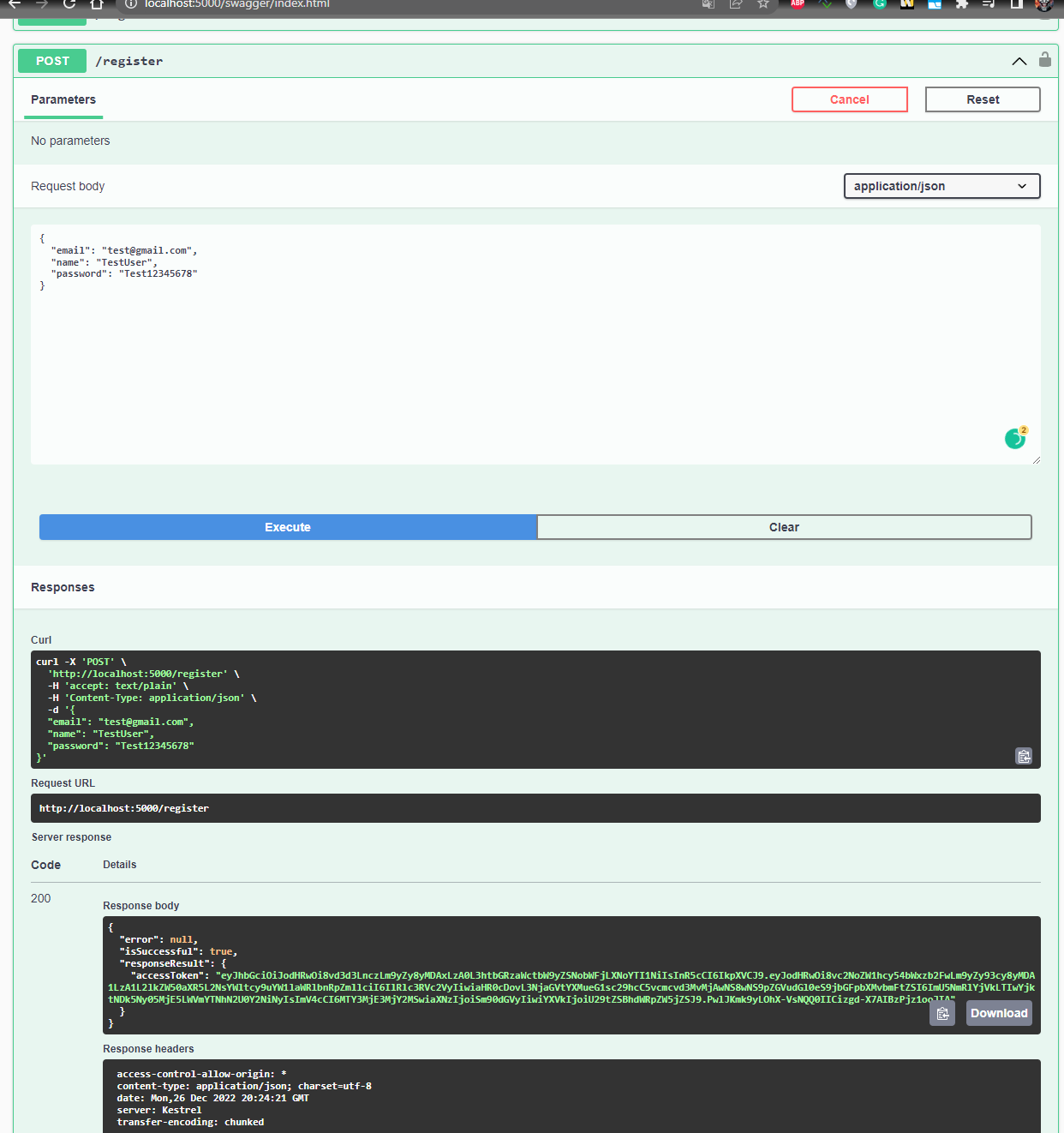
1. Створимо dockecr-compose

****

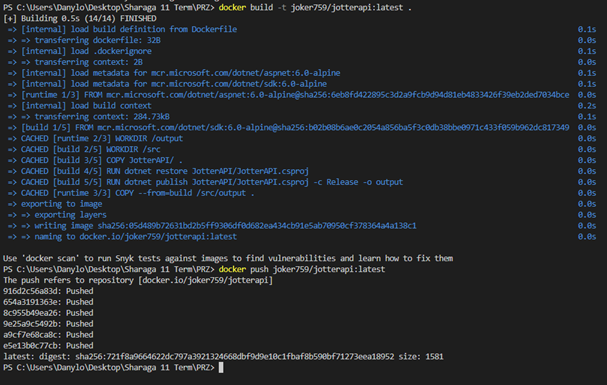
1. Перевіримо працездатність додатку:

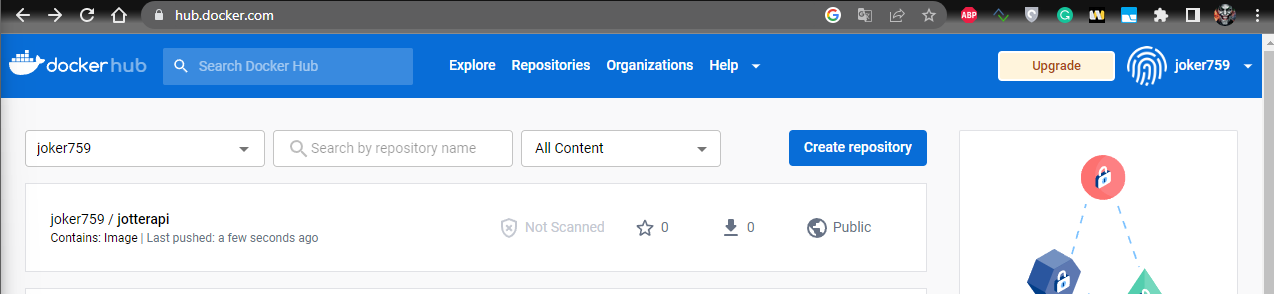
**docker-compose up**

****

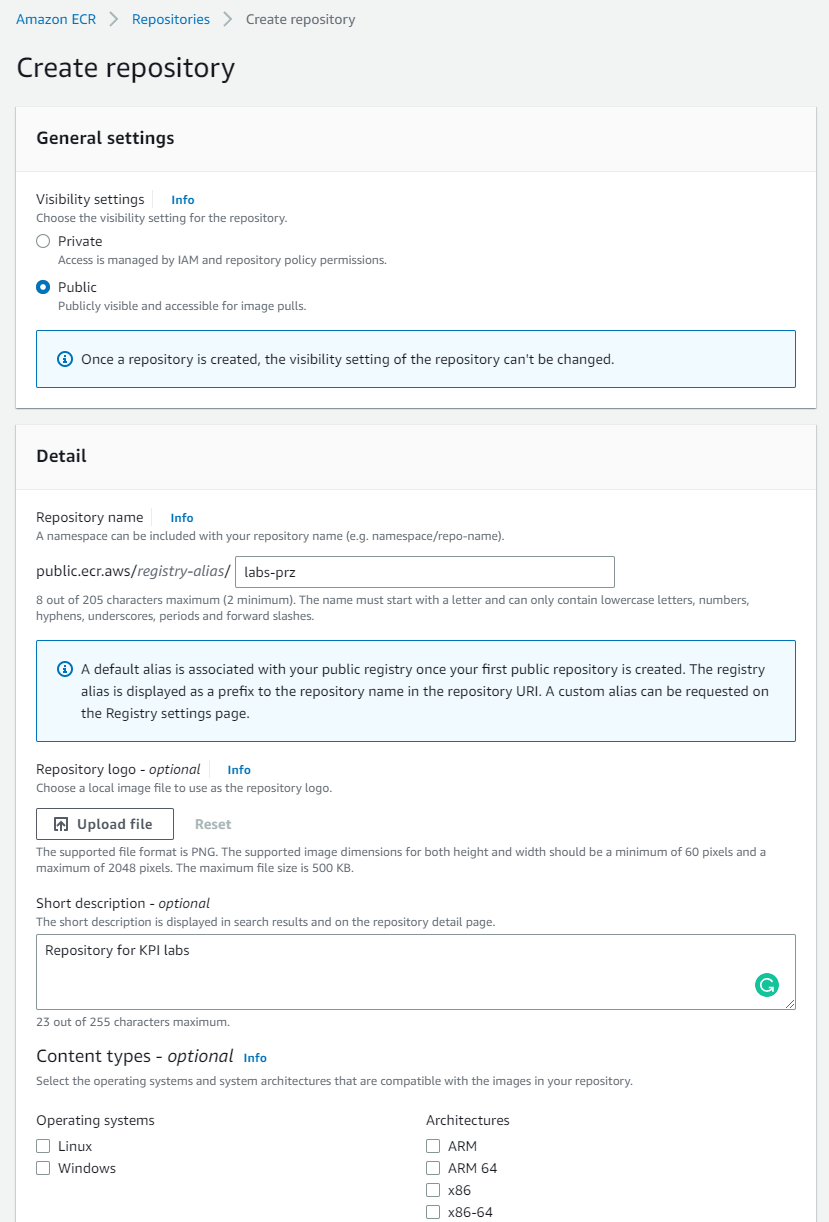
****

1. Опублікуємо docker image на «DockerHub»:

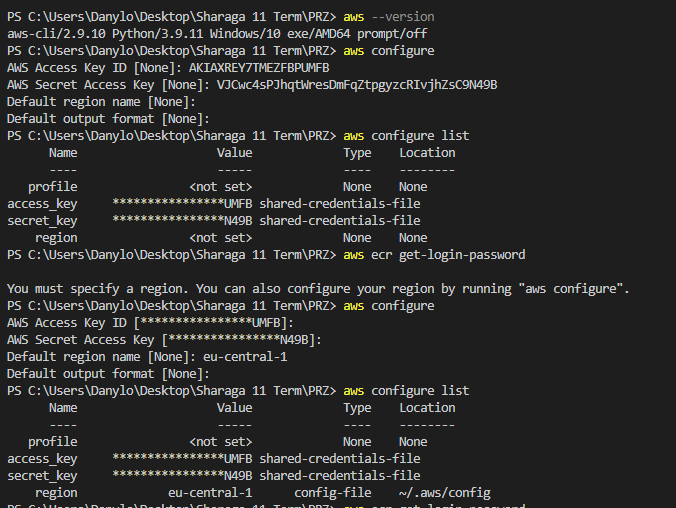


****Але, так як цього варіанту немає, розглянемо через AWS.

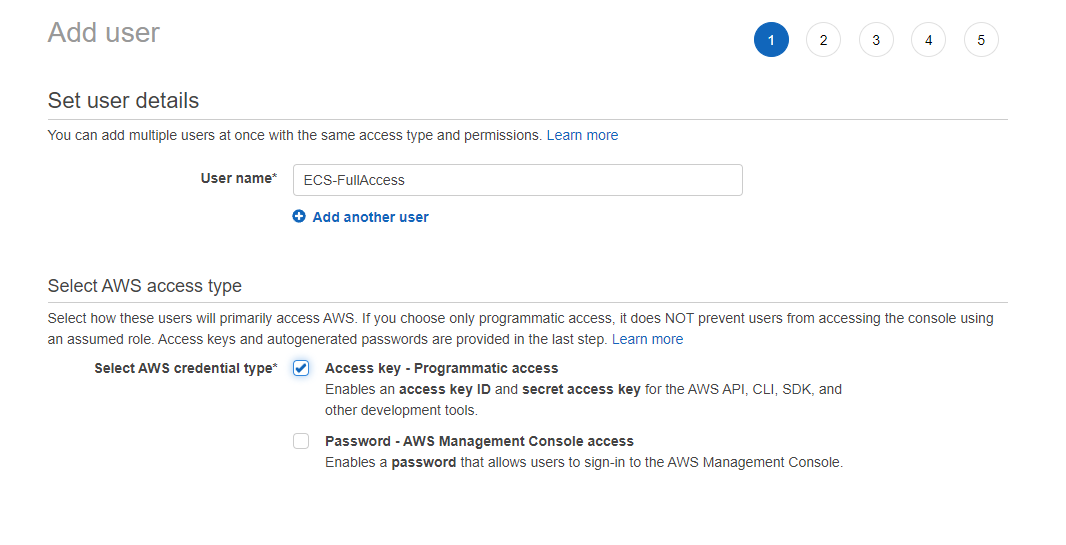
1. Створимо публічний репозиторій використовуючи «Amazon Elastic Container Registry»:

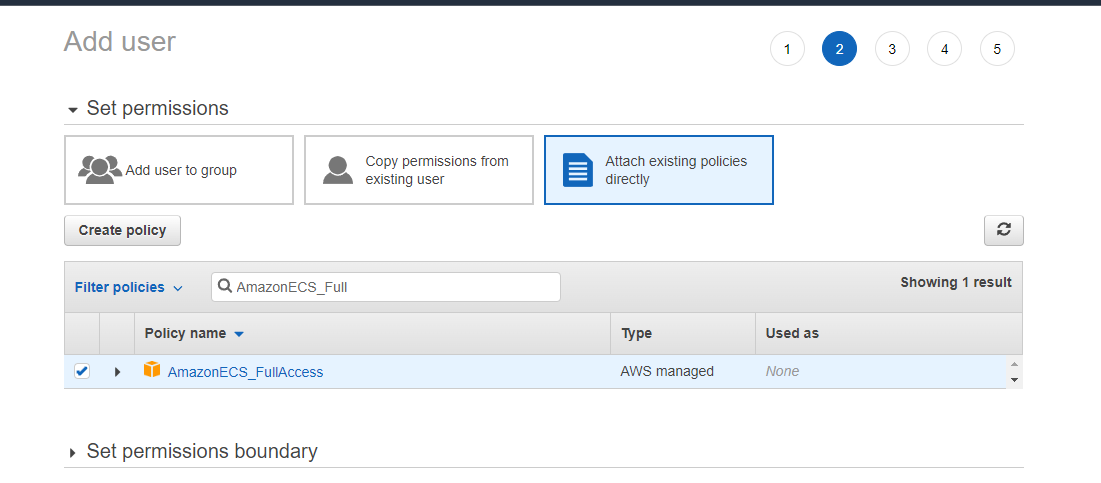
****

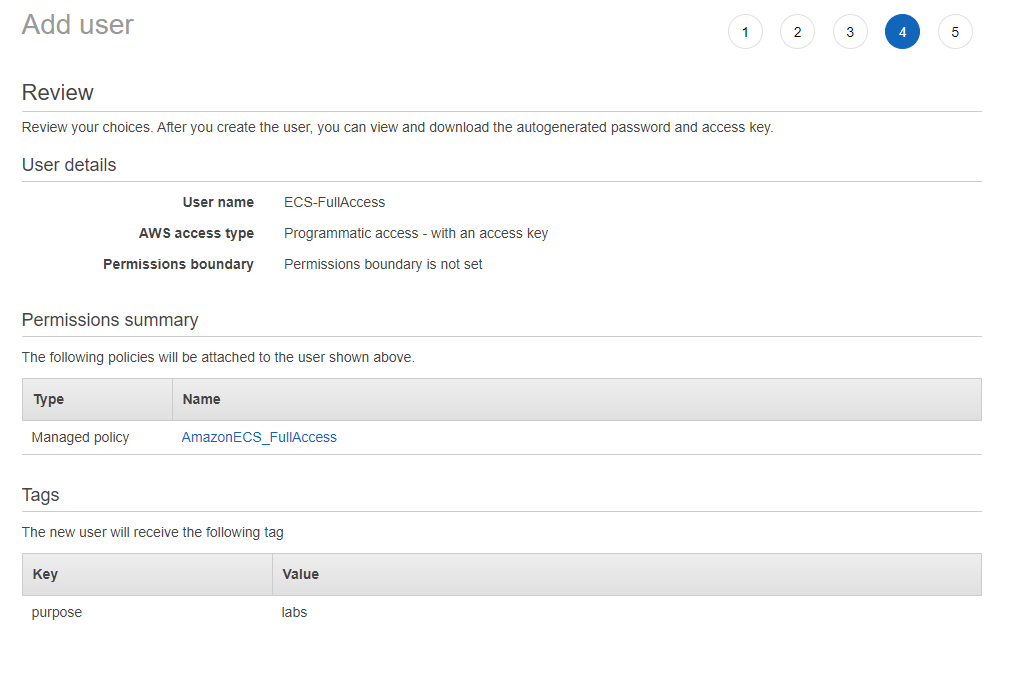
1. Встановимо і налаштуємо **AWSCLI** для роботи з AWS:

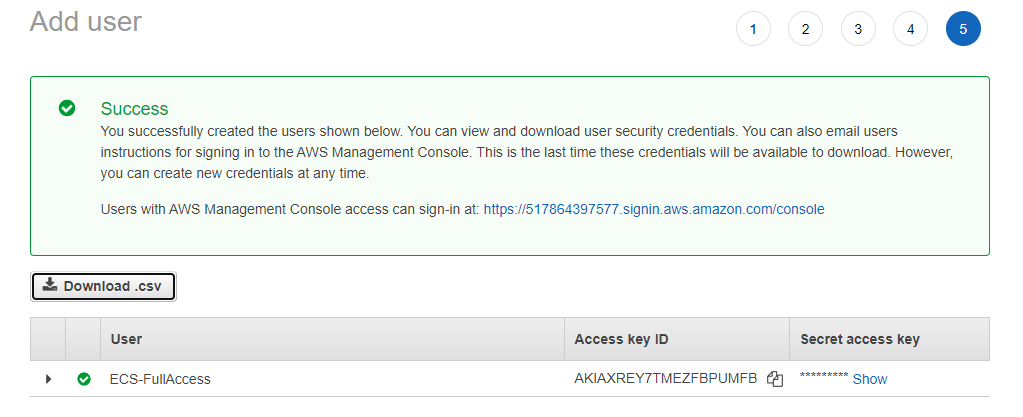
****

1. Створимо юзера для роботи з «Amazon Elastic Container Registry»:

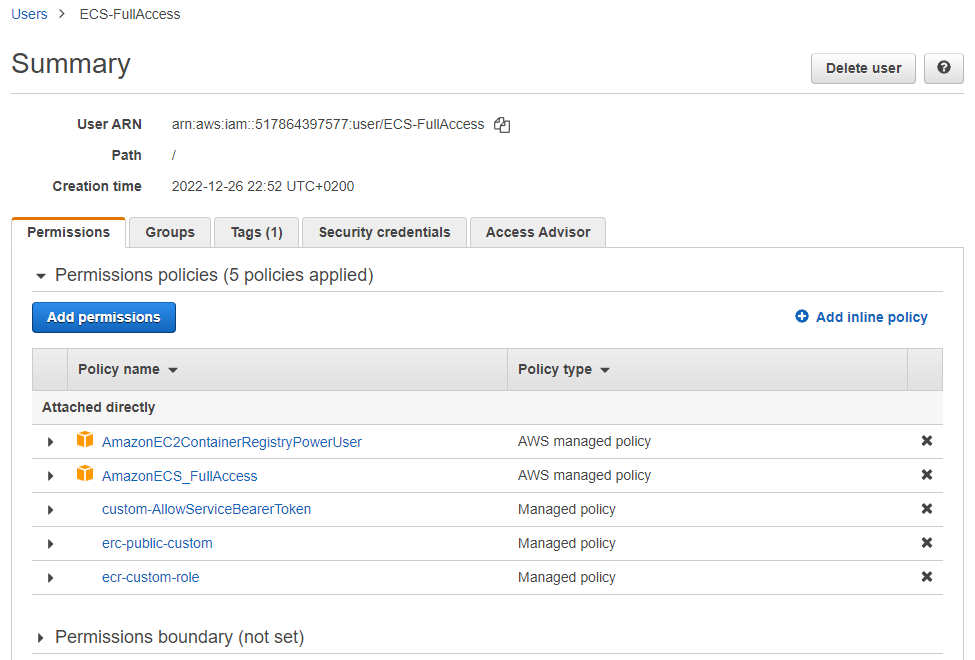
****

****

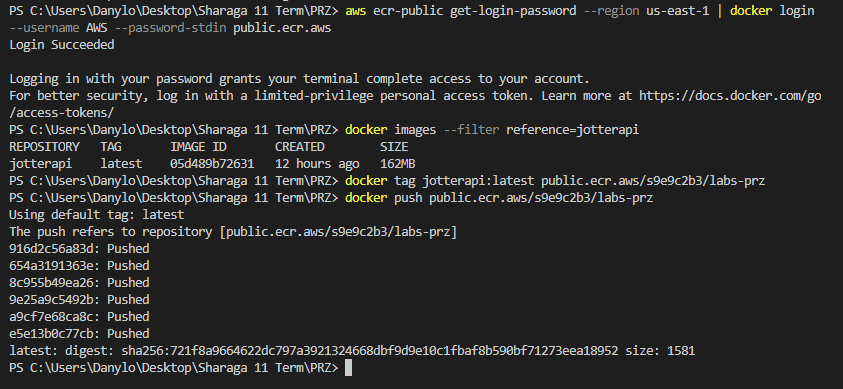
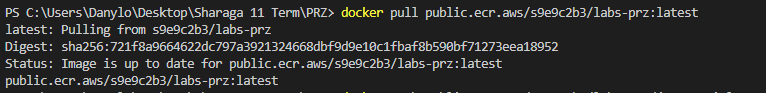
****

****

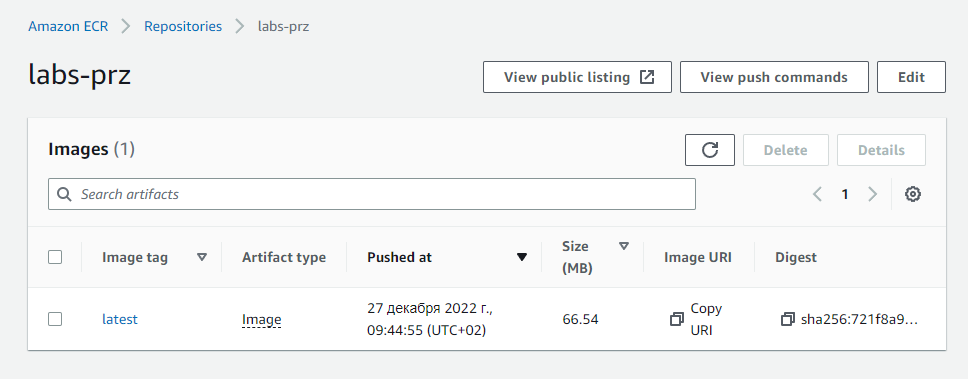
1. Модифікуємо створеного юзера, додаймо додаткові політики безпеки:

****

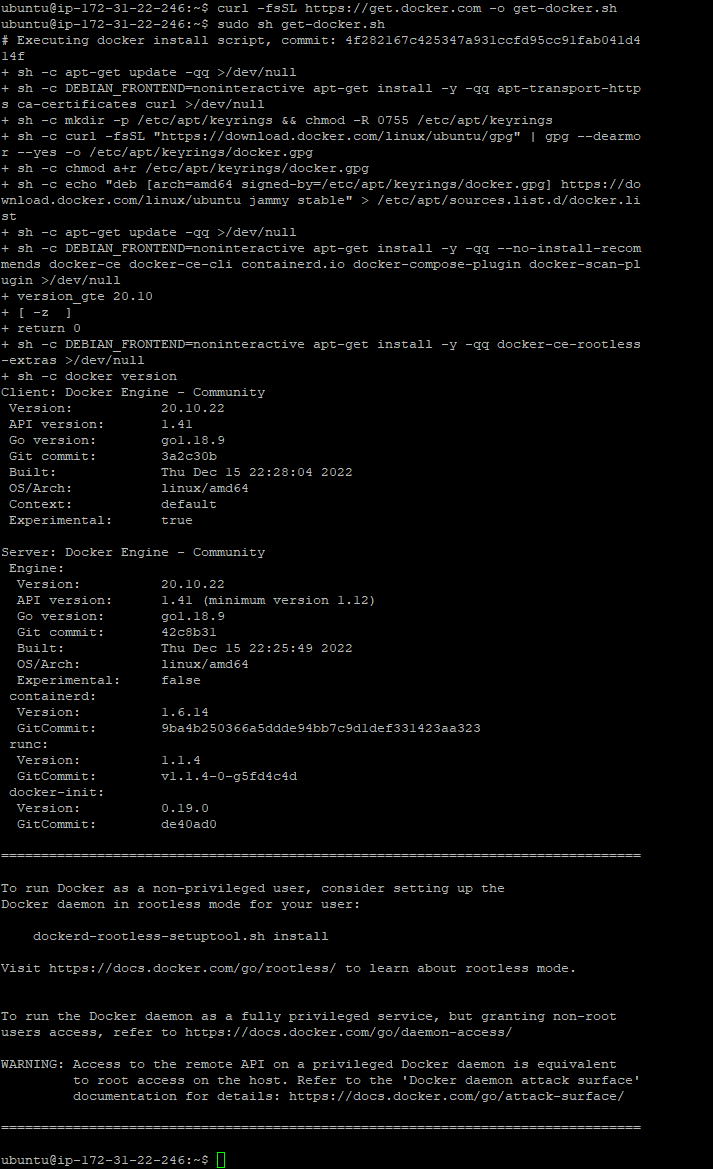
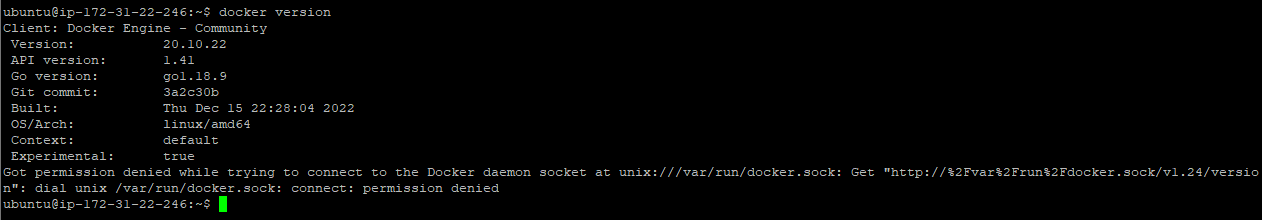
1. Логінемося в AWS ECR використовуючи AWSCLI, пушимо докер образ:

**  
**

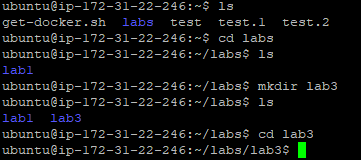
1. Перевіряємо, що образ запушено:

****

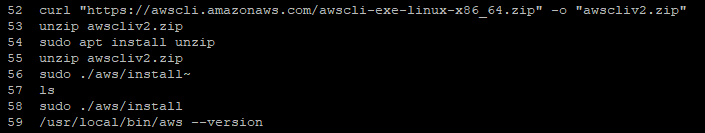
1. Перейдемо на віртуальний сервер, створений в ЛР1. Нам треба встановити та налаштувати docker:

**  
**

1. Створимо папку для роботи з ЛР3:

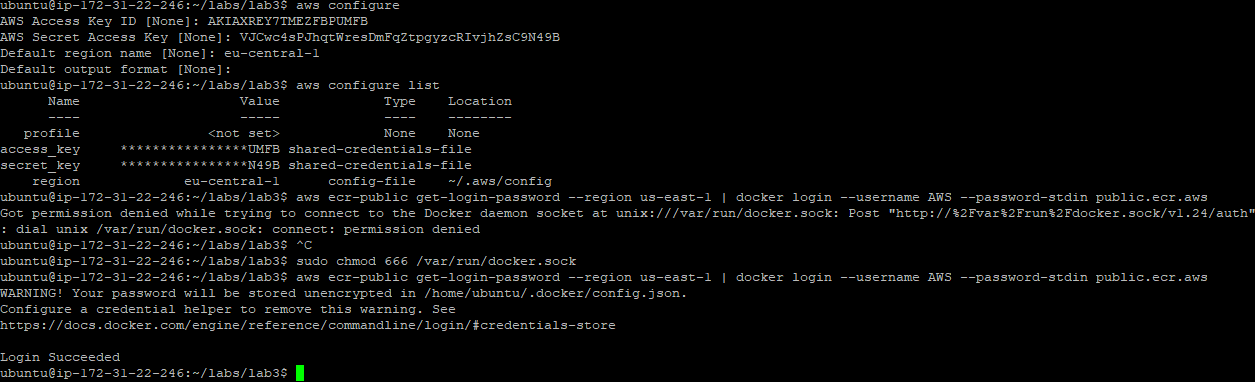
****

1. Встановимо AWSCLI на віртуальну машину:

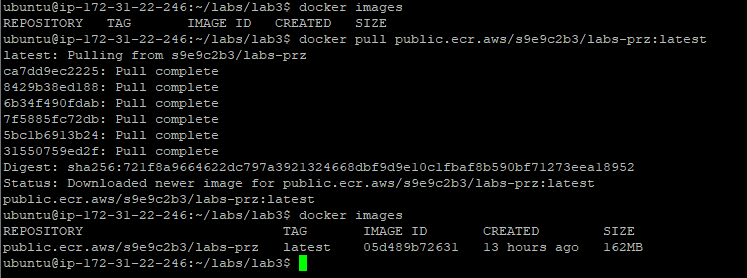
****

****

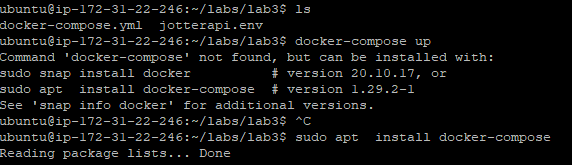
1. Налаштовуємо локального користувача, логінемося:

****

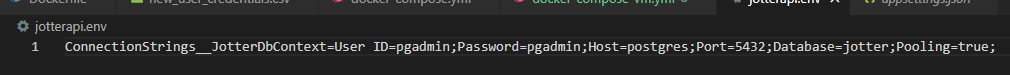
1. Завантажемо наш docker image з AWS ECR на віртуальну машину:

****

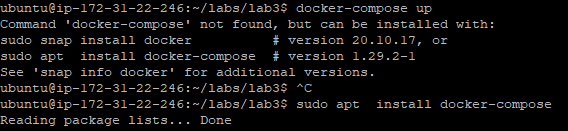
1. Скопіюємо docker-compose, зробимо відповідні зміни:

****

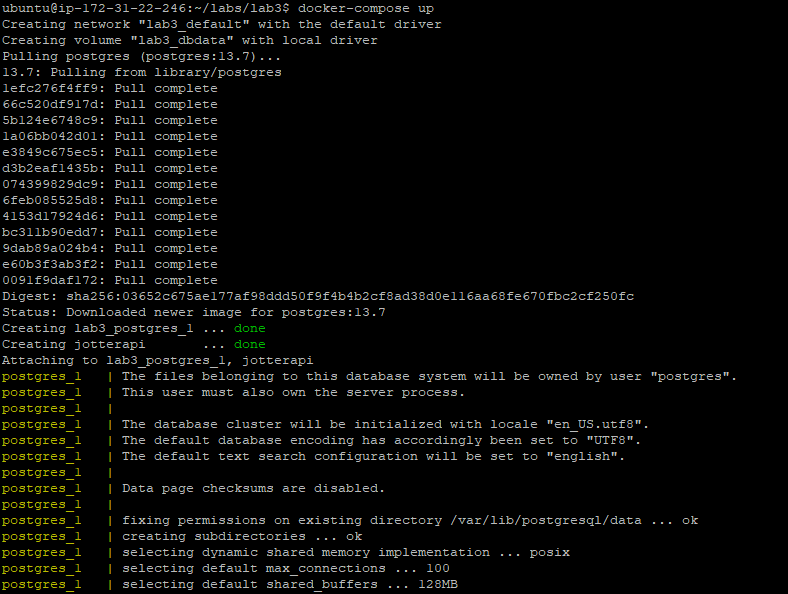
****

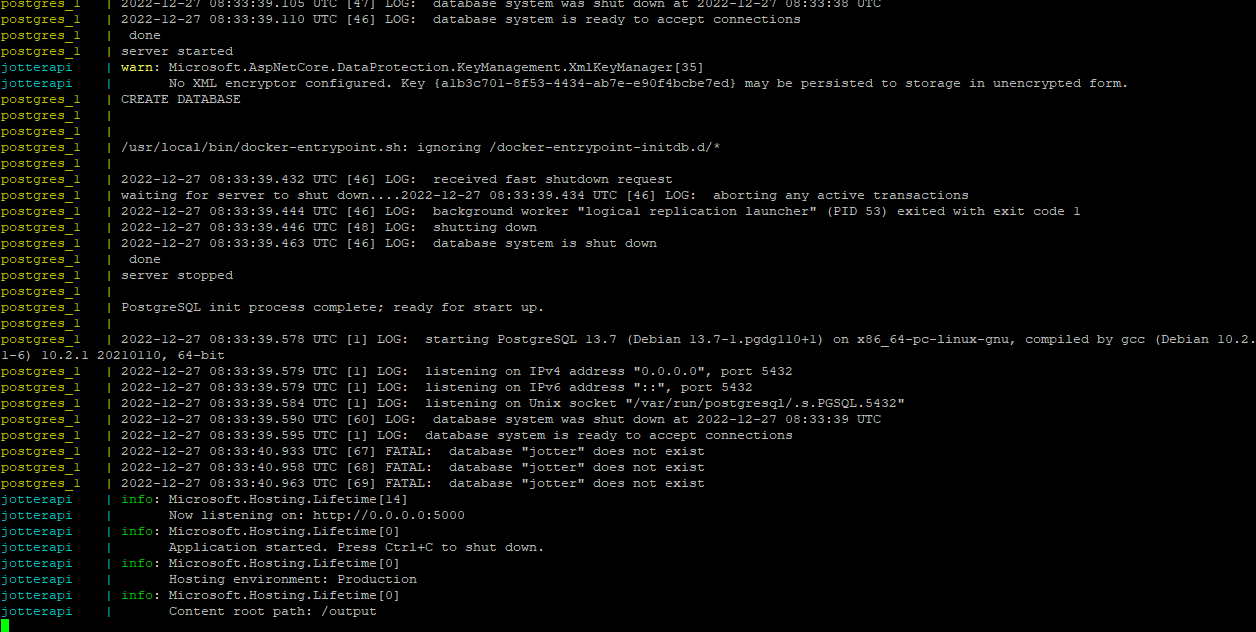
****

1. Встановимо docker-compose:

****

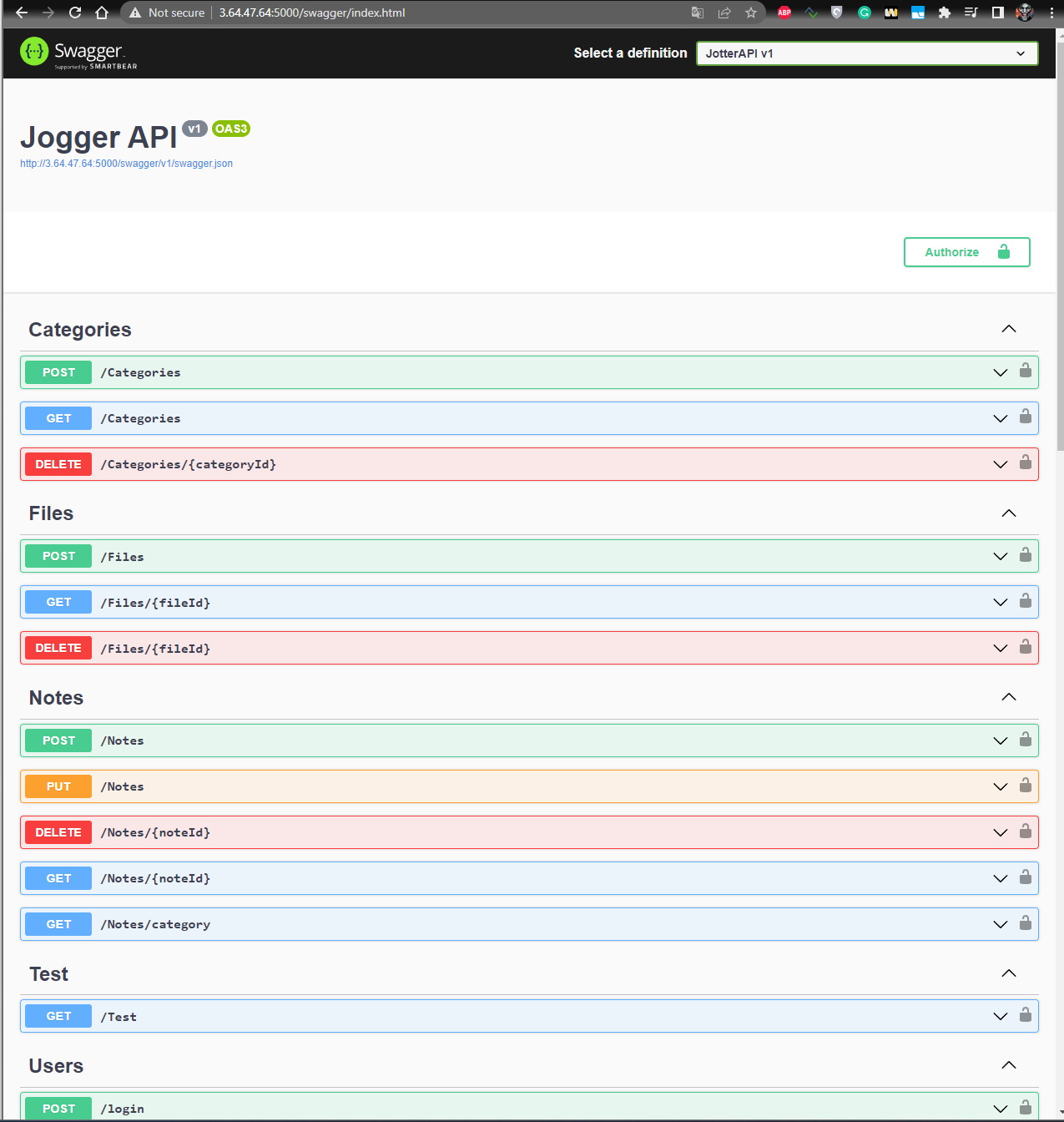
1. Запустимо додаток:

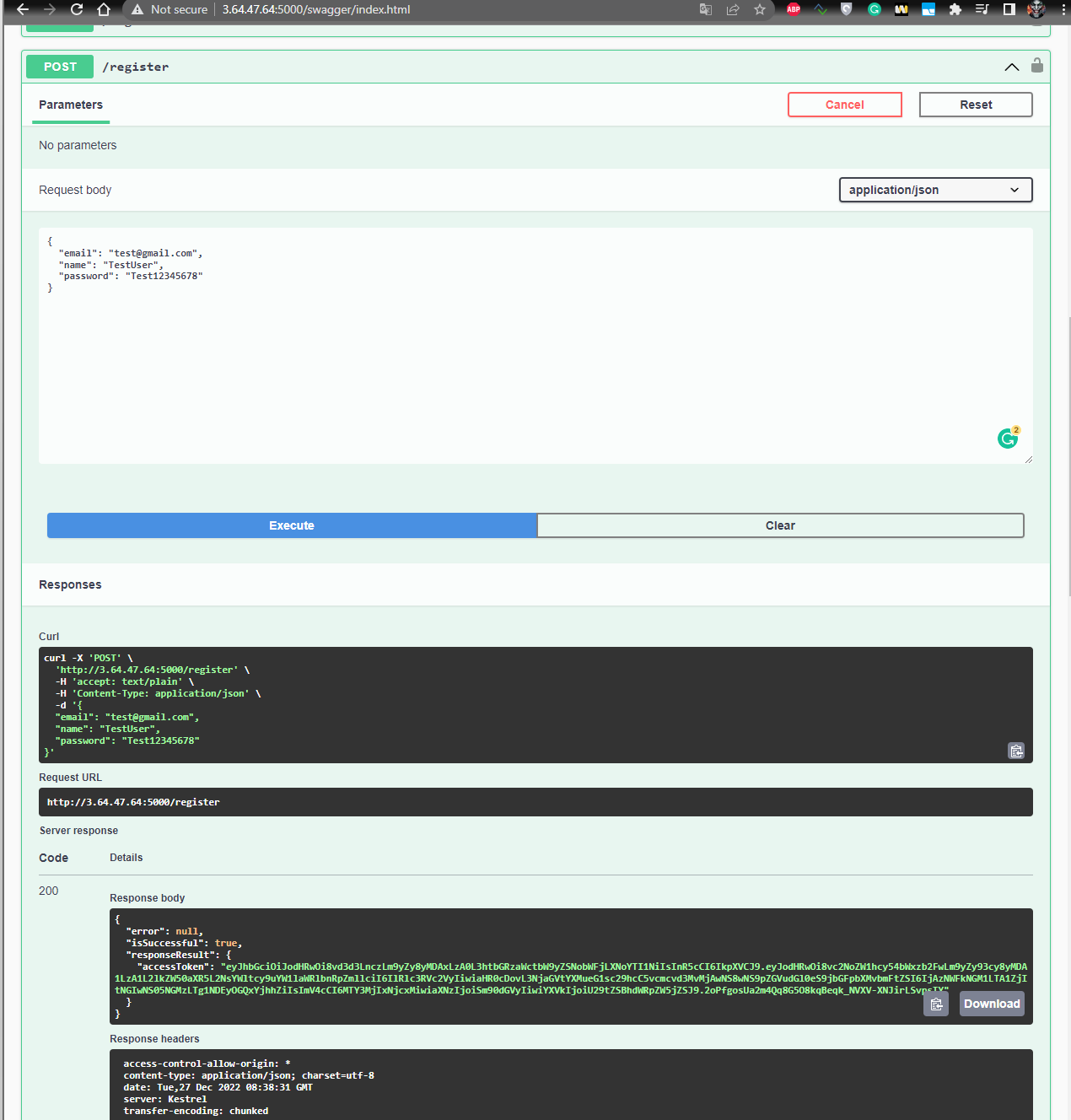
****

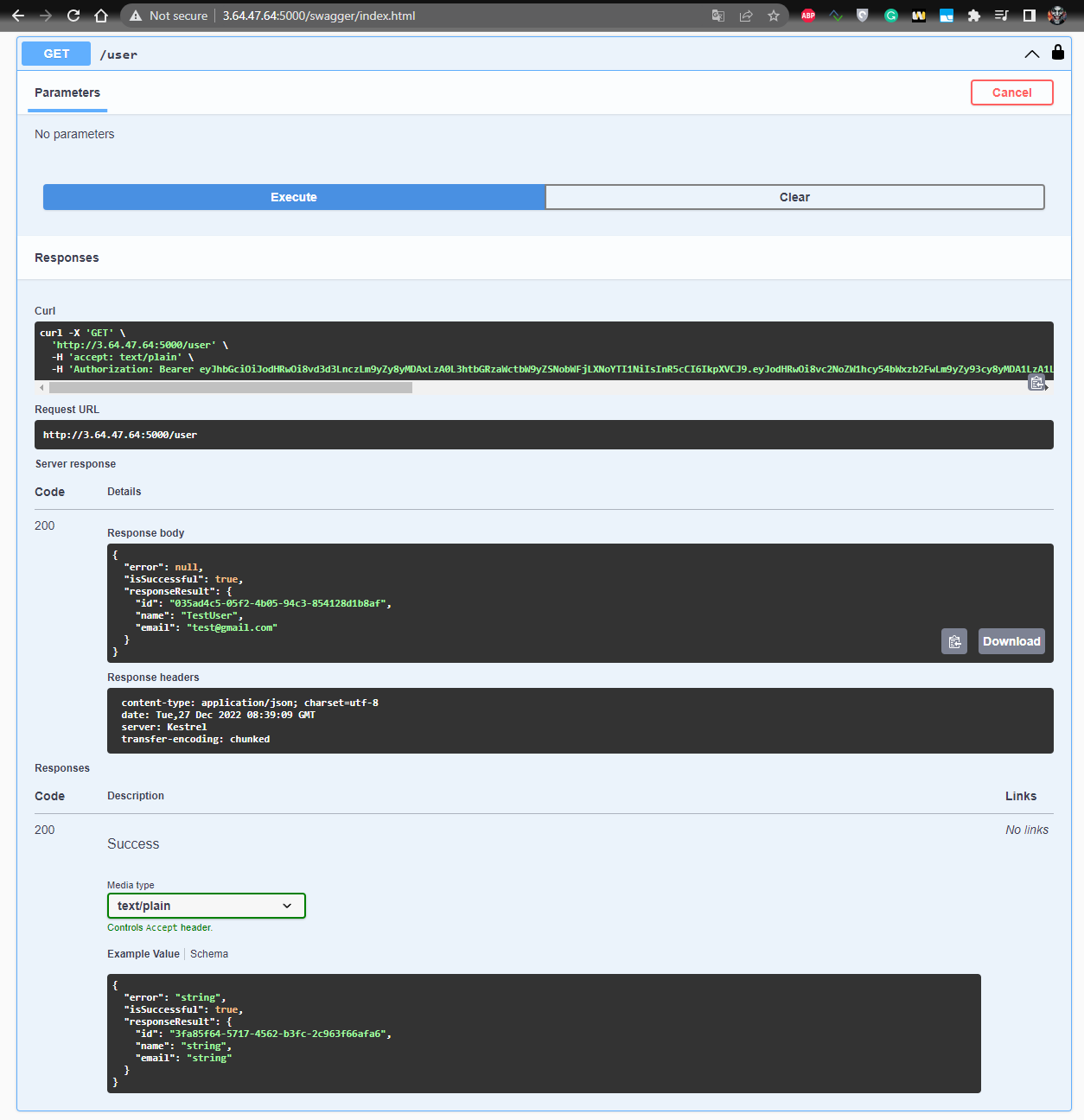
****

1. Перевіримо його на працездатність:

****

****

****

****

**Висновок:**

В результаті виконання лабораторної роботи було розгорнуто .NET застосунок за допомогою Docker (як локально, так і на сервері). docker image було опубліковано в публічному Container registry – Amazon Elastic Container Registry.