

Міністерство освіти і науки України Національний технічний університет України „КПІ

імені Ігоря Сікорського ”

Факультет інформатики та обчислювальної техніки Кафедра інформатики і програмної інженерії

# ЗВІТ

лабораторної роботи № 2

з курсу «Інфраструктура програмного забезпечення WEB- застосувань»

Тема: «Дослідження спільних ресурсів хостової та гостьової систем в Docker»

|  |  |
| --- | --- |
| Перевірив:  Викл. Орленко Сергій Петрович | Виконали:  Борисик Владислав  Мєшков Андрій  Мочалов Дмитро |

Київ 2024

# 1. Завдання

**Мета роботи:** полягає у дослідженні специфіки запуску Docker контейнерів, ознайомленні з репозиторієм Docker Hub та, за потреби, Docker Desktop. Навчитися прокидати порти з гостьової на хостову машини, що дасть змогу працювати з власним веб-сервером nginx.

# Вхідні дані ЛР2

У якості вхідних даних для ЛР2 є:

– виконана ЛР1 та її Docker-образи.

Додатково слід пам’ятати що в цій лабораторній роботі:

\*– без символів “<>”;

\*\* – прізвища ААА=1+1+1=3=8003, а ZYXZ=26+25+24+26=101=8101.

# Вихідні дані ЛР2

У якості вихідних даних для ЛР2 є: робочий контейнер з веб-сервером nginx, звіт.

**Завдання**

1. Навчитися використовувати спільні ресурси хостової та гостьової систем на прикладі Docker nginx.

2. Переглянути контет сторінок Docker nginx з хостової машини.

# Хід роботи

1. Для того, щоб на сайт Nginx, який працює всередині контейнера, можна було зайти через певний порт на хостовій машині, потрібно прокинути порт.

B + M + M = 2 + 13 + 13 = 28 –> 8028

docker run -p 8028:80 lab01\_1brbmm – запускає Docker-контейнер з прокиданням порту.

* -p 8028:80 — прокидання порту 8028 на хостовій машині до порту 80 всередині контейнера.

Перевіремо сторінку за посиланням http://127.0.0.1:8028/.

На рисунках 1.1-1.3 показані ці команди та результати виконання

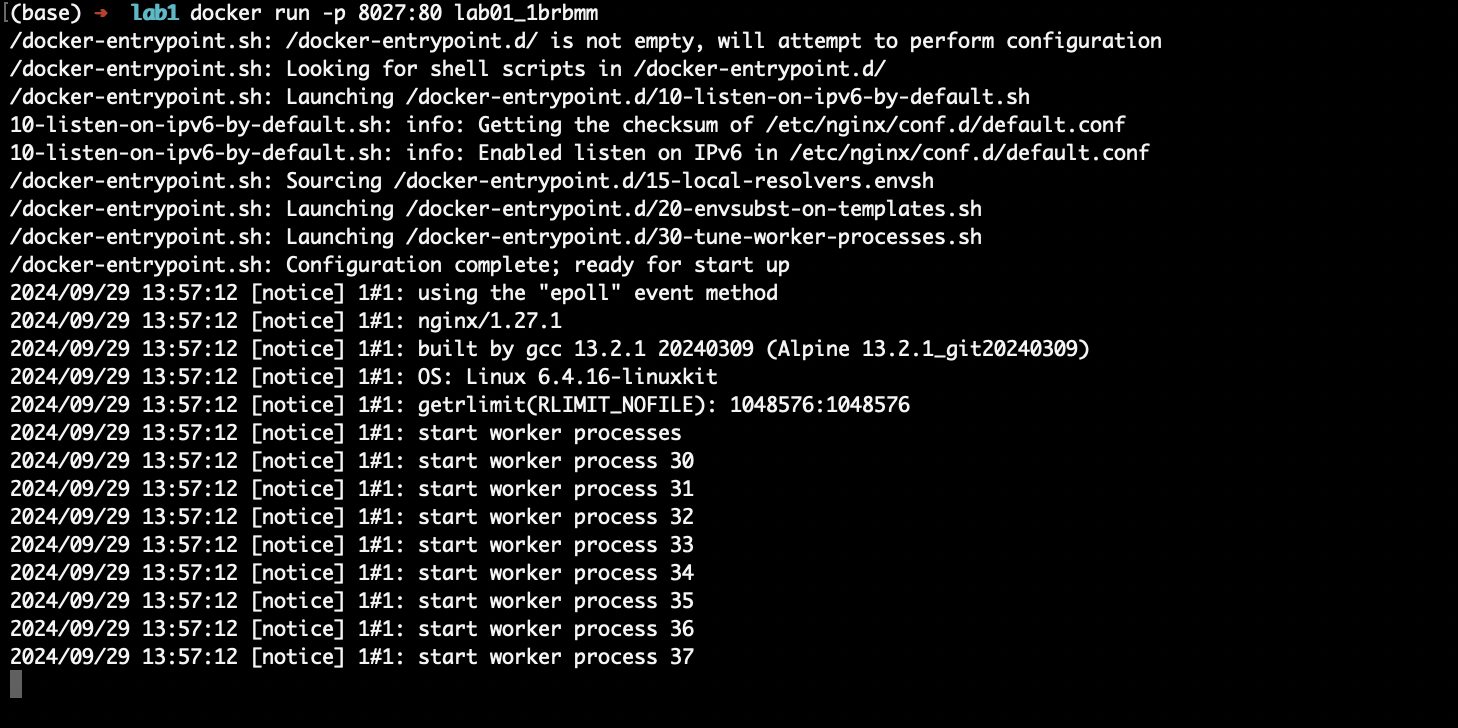


Рисунок 1.1 – команди та результати виконання

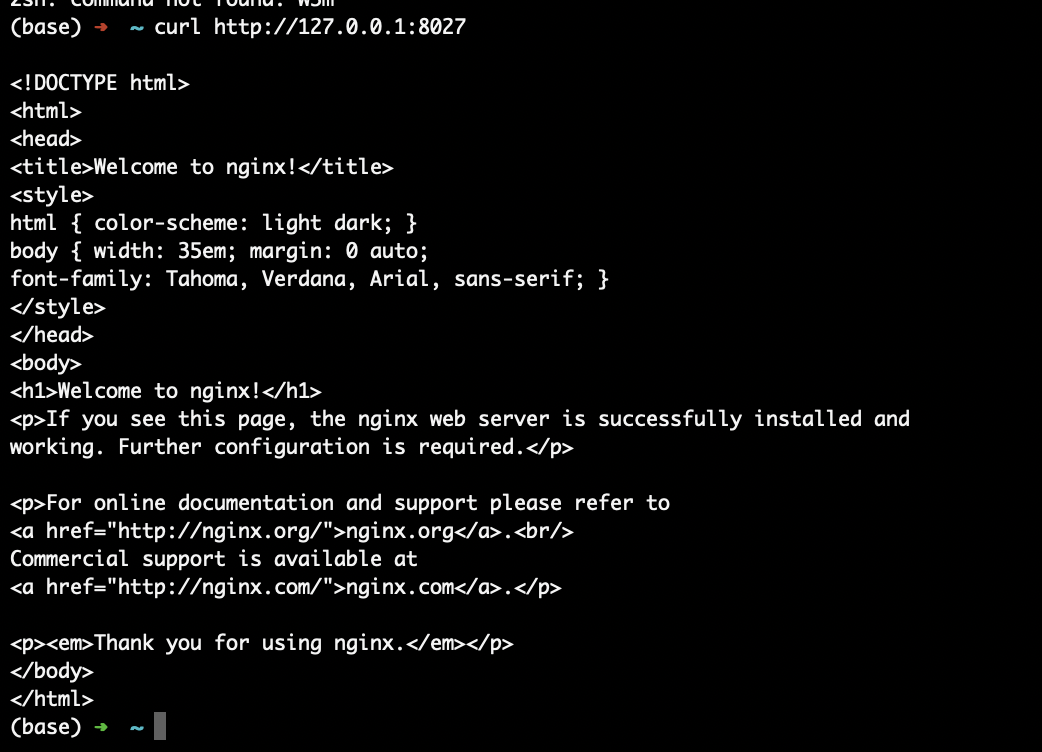


Рисунок 1.2 – команди та результати виконання

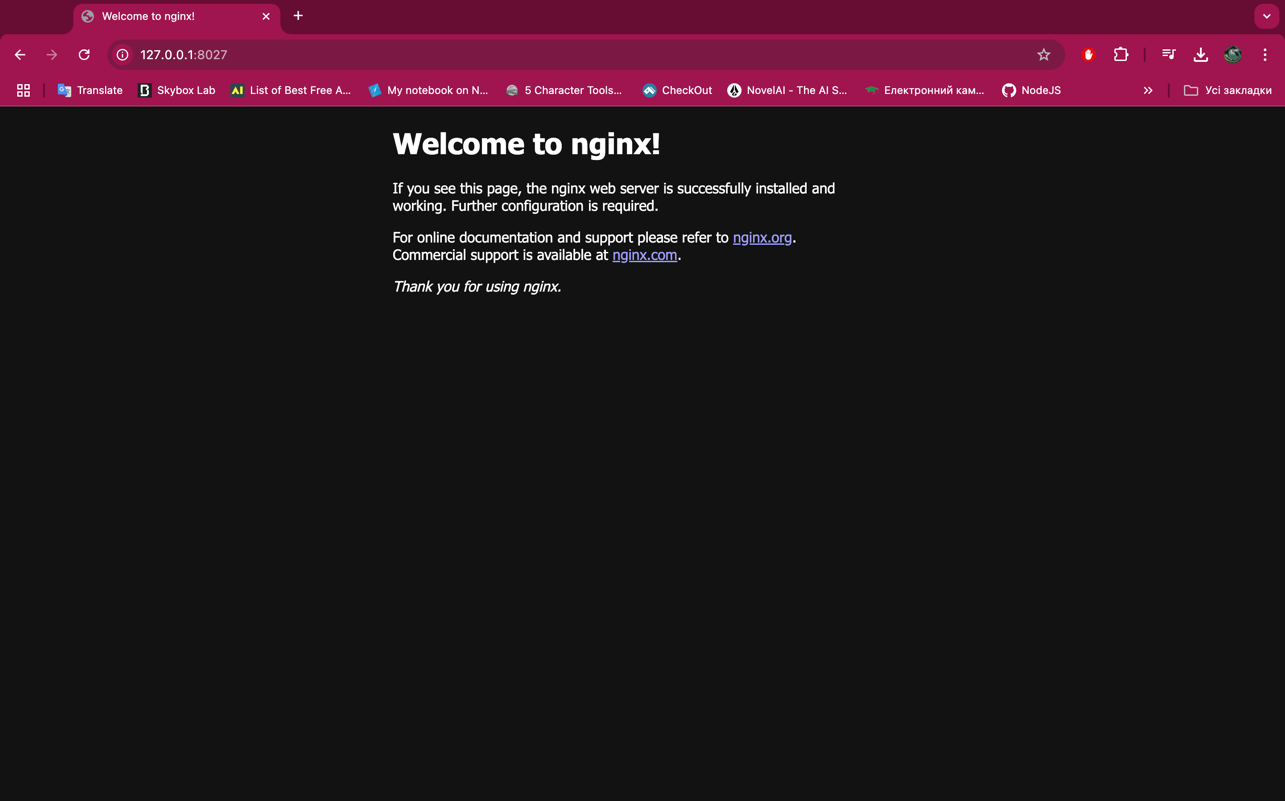




Рисунок 1.3 – веб-сторінка

2. Щоб замінити контент дефолтної сторінки Nginx на власний, включаючи перелік ПІБ членів бригади і поточну дату створення образу, потрібно внести зміни в Dockerfile і створити новий образ.

mkdir html – створимо папку для html файлу.

nano html/index.html – створимо html файл.

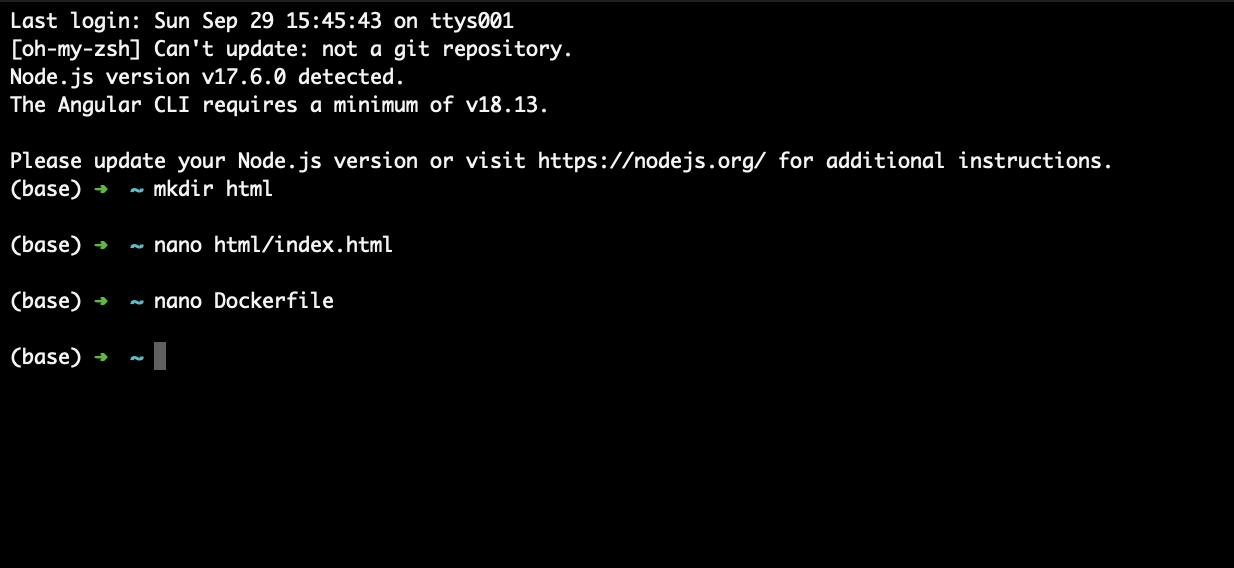


Рисунок 1.4 – команди та результати виконання

Наповнимо сторінку кодом.



Рисунок 1.5 – код нової веб-сторінки

Тепер необхідно оновити Dockerfile, щоб замінити дефолтну сторінку Nginx на вашу HTML сторінку.

nano Dockerfile - відкрийте Dockerfile для редагування. Внесемо новий зміст.

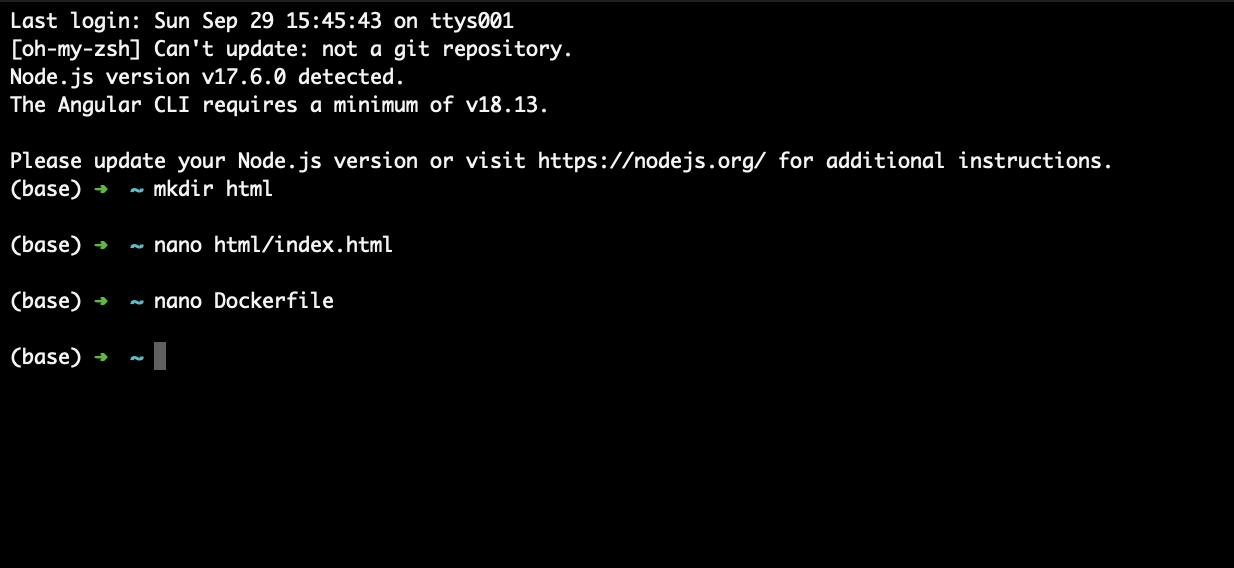


Рисунок 1.6 – команди та результати виконання

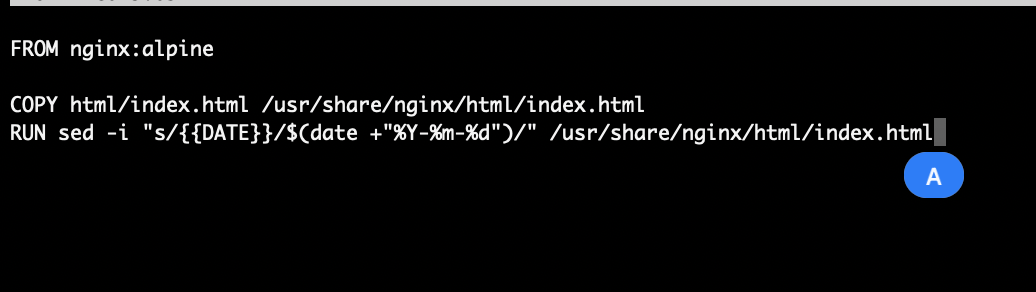


Рисунок 1.7 – новий зміст Docker файлу

Створимо новий Docker-образ з ім'ям lab01\_2bbmm (де bmm — це перші літери прізвищ членів бригади).

docker build -t lab01\_2bbmm .

docker run -p 8027:80 lab01\_2bbmm – запустимо контейнер.

Перейдемо у браузер і відкриємо сторінку http://127.0.0.1:8028. Ми повинні побачити сторінку з переліком членів бригади та датою створення.

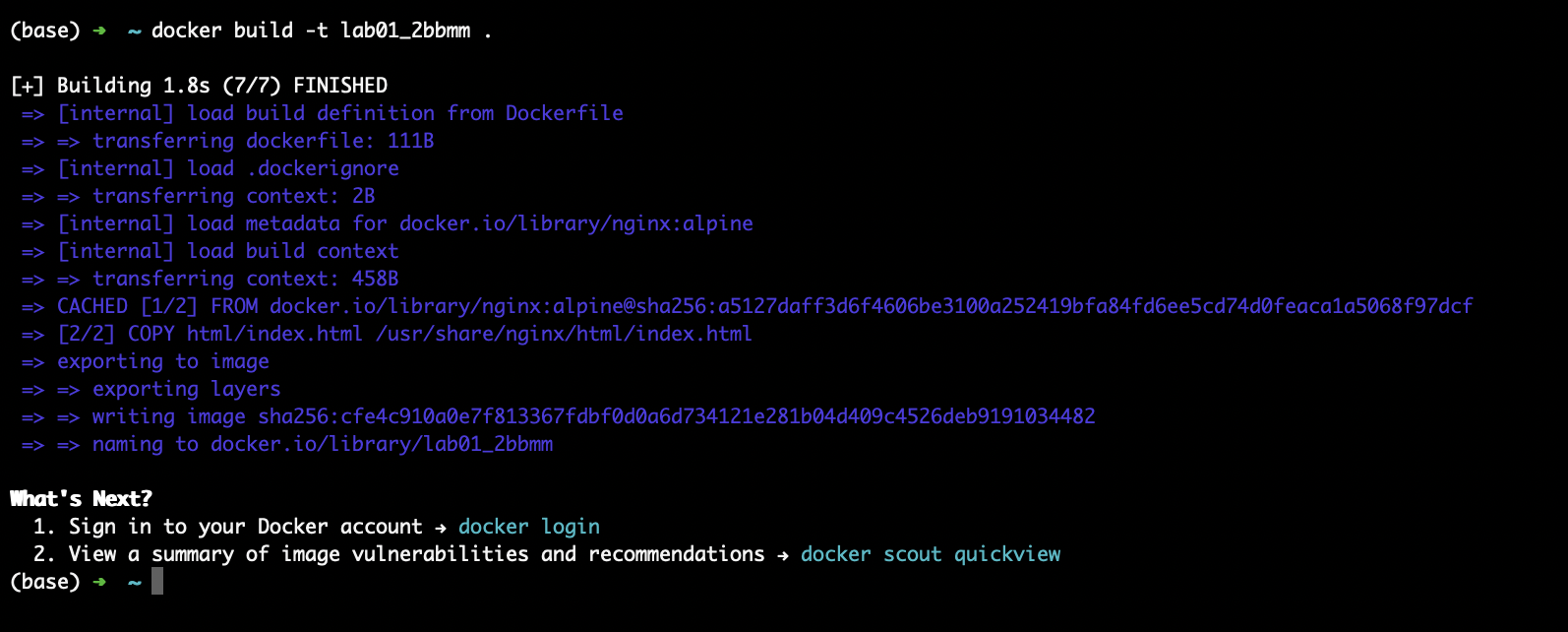
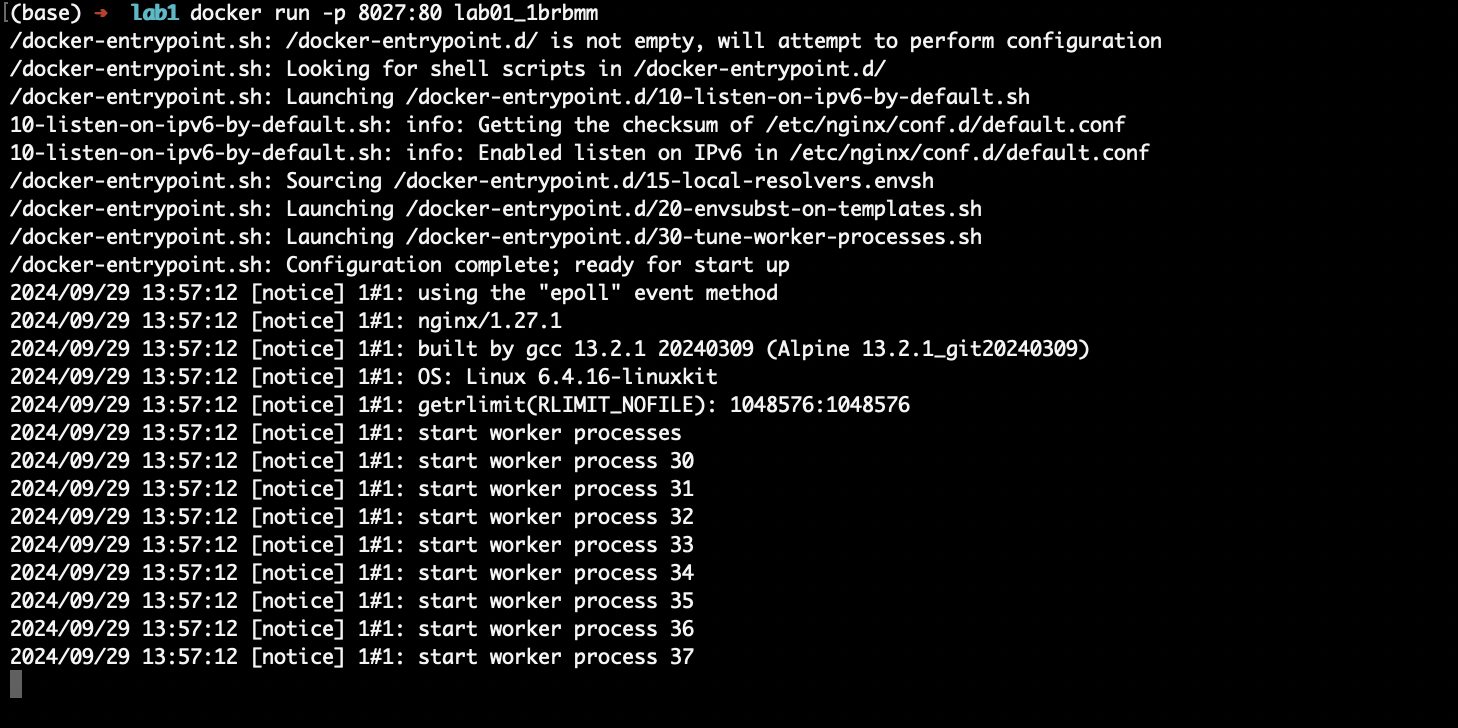


Рисунок 1.8 – команди та результати виконання



2bbmm



Рисунок 1.9 – команди та результати виконання

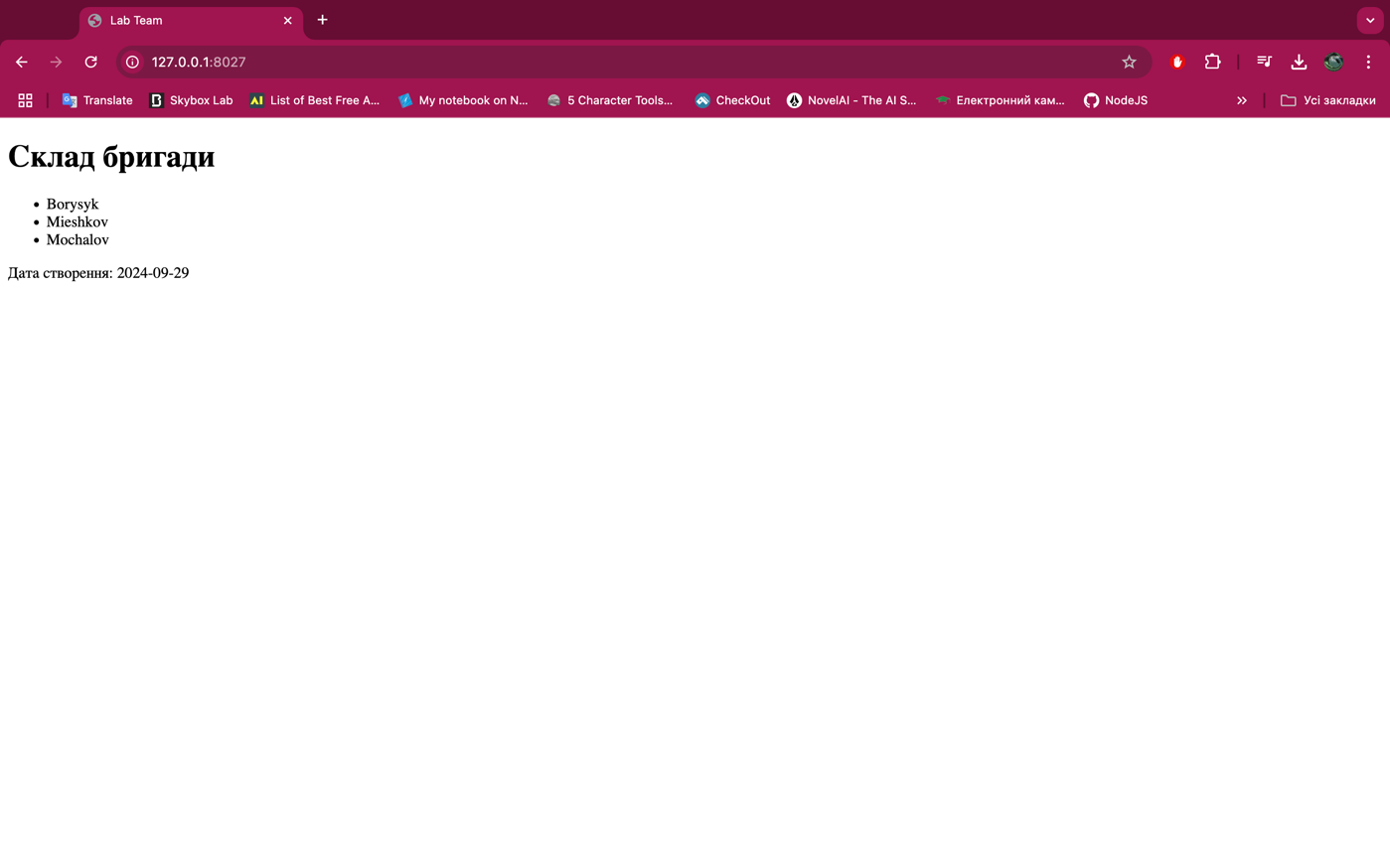




Рисунок 1.10 – новий вміст сторінки

3.У каталозі ./lab1 на локальній машині створемо HTML-файл з потрібною інформацією.

nano index.html – створемо файл.

Занесемо у файл код сторінки. Збережемо зміни у файлі.

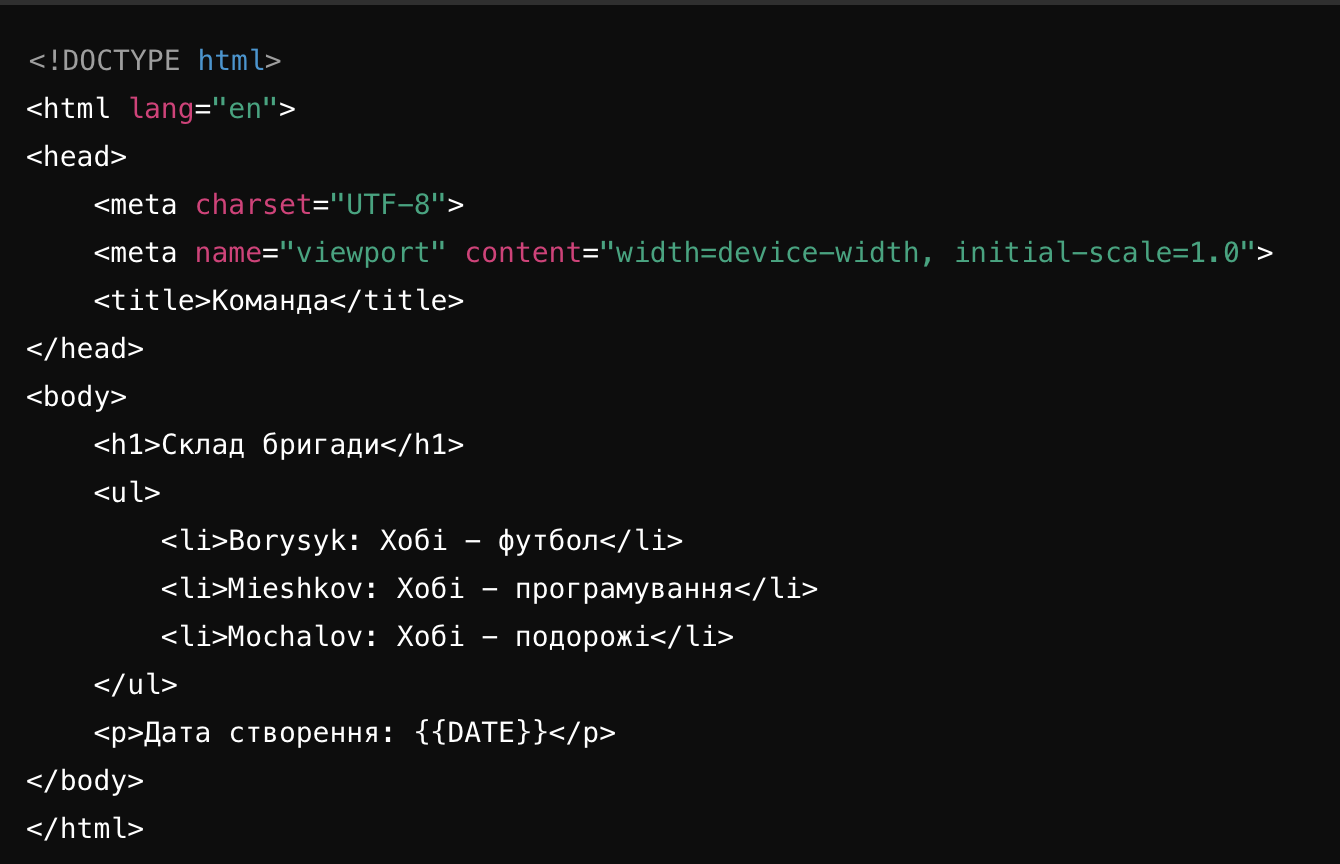




Рисунок 1.11 – код сторінки

Відкриймо Dockerfile для редагування. Оновимо Dockerfile так, щоб він використовував VOLUME для розшарювання веб-застосунку з папки ./lab1 в контейнер.

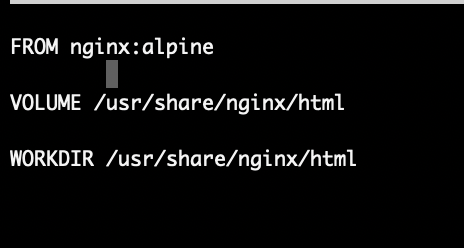


Рисунок 1.12 – зміст Dockerfile

docker build -t lab01\_3brbmm . – Побудуймо новий образ з оновленим Dockerfile.

docker run -p 8027:80 -v $(pwd)/lab01:/usr/share/nginx/html lab01\_3brbmm– запустимо контейнер.

* -v $(pwd)/lab1:/usr/share/nginx/html — розшарює папку ./lab1 з нашим HTML-контентом на хостовій машині в папку Nginx всередині контейнера.



Рисунок 1.13 – команди та результати виконання

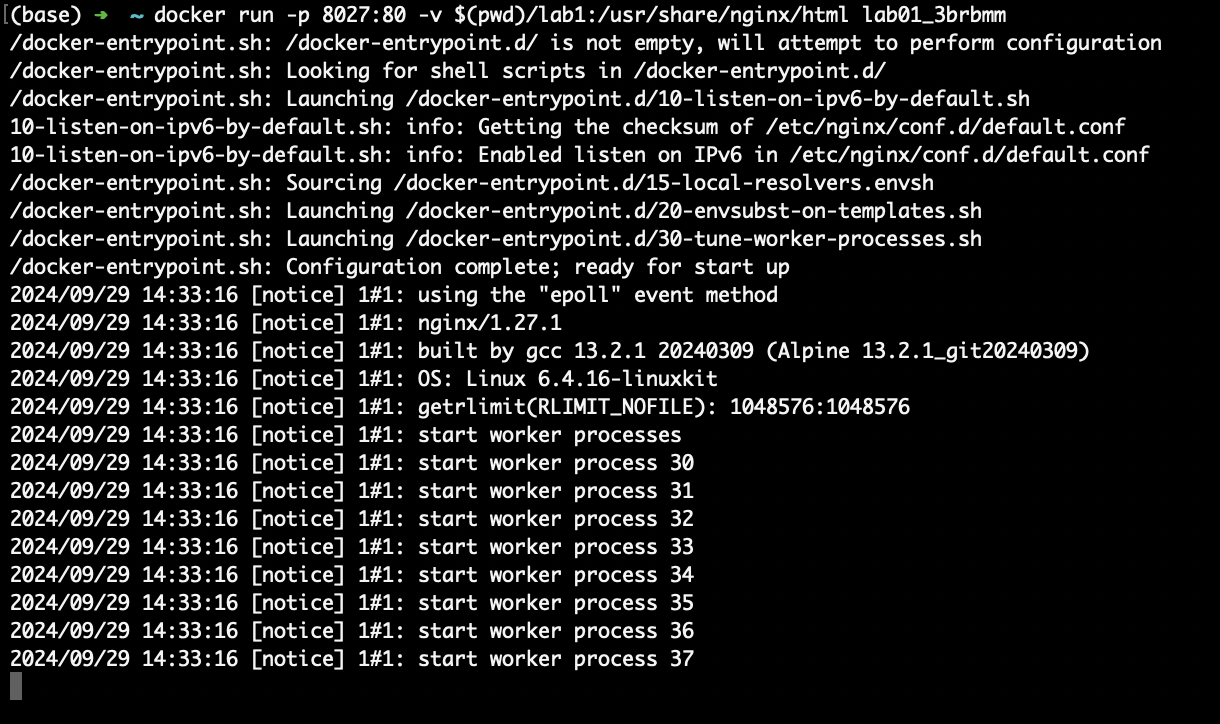


Рисунок 1.14 – команди та результати виконання

Перевіремо нову веб-сторінку.

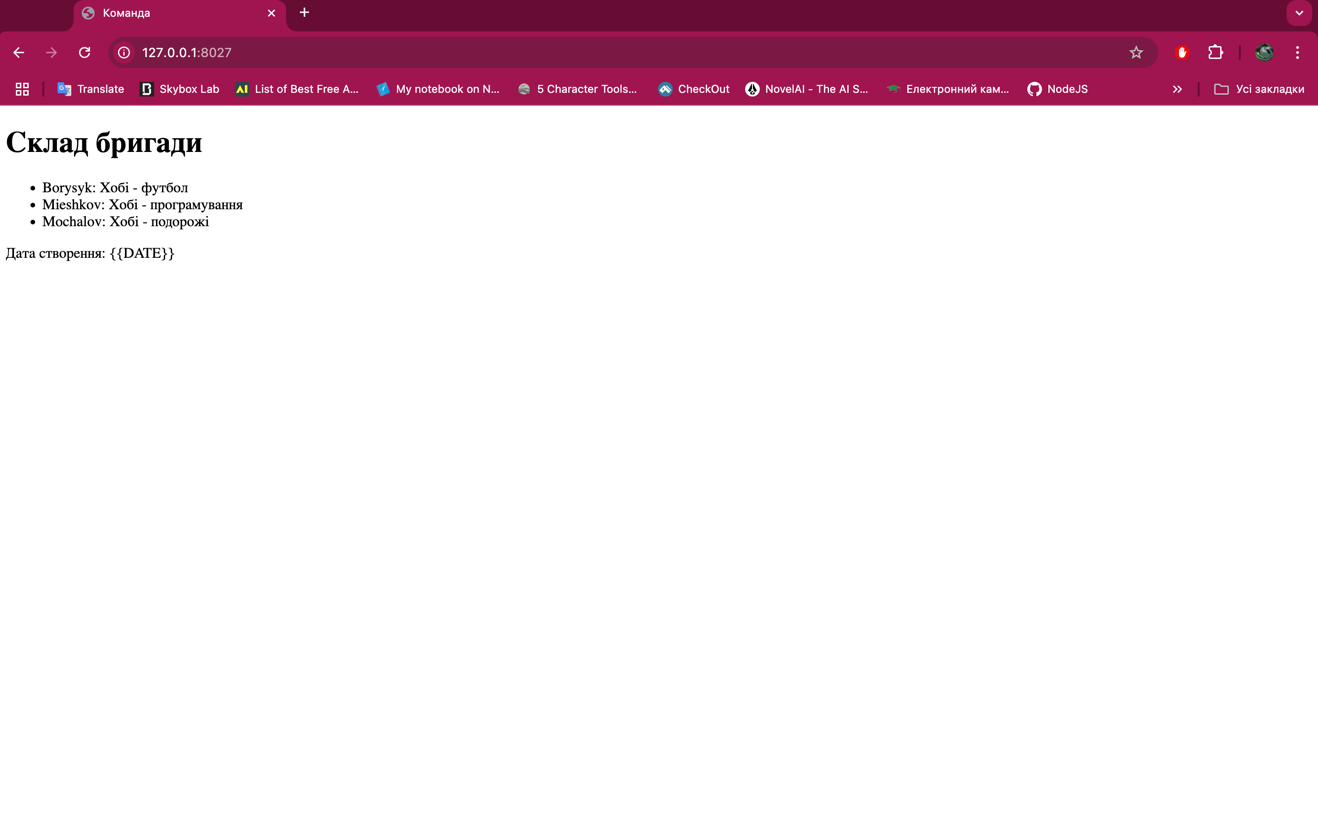




Рисунок 1.15 – новий вміст сторінки

ВИСНОВКИ

У процесі виконання лабораторної роботи були отримані практичні навички з використання Docker для створення, запуску та управління контейнерами. Зокрема, було продемонстровано, як створити Docker-образ на основі офіційного образу Nginx, а також як використовувати команду `docker run` для запуску контейнера з перенаправленням портів.

В процесі лабораторної роботи було досліджено, як Nginx працює як веб-сервер. Було проведено тестування веб-сайту, розгорнутого в контейнері, з можливістю доступу через браузер. Це допомогло краще зрозуміти принципи роботи веб-серверів та особливості налаштування.

Лабораторна робота включала створення спільного ресурсу між хостовою і гостьовою системами через Docker Volume. Це дозволило зберегти контент веб-сайту на локальному комп'ютері і з легкістю оновлювати його без необхідності перебудови Docker-образу.

Було продемонстровано, як за допомогою команди `RUN` у Dockerfile можна динамічно оновлювати контент веб-сторінки, вставляючи поточну дату. Це підтверджує гнучкість Docker у створенні і налаштуванні середовища.

Лабораторна робота надала можливість не лише теоретично ознайомитися з Docker та Nginx, але й на практиці впровадити отримані знання. Дослідження спільних ресурсів між хостовою та гостьовою системами продемонструвало потужність та гнучкість контейнеризації у розробці та розгортанні веб-додатків.