

Міністерство освіти і науки України

Національний технічний університет України „КПІ

імені Ігоря Сікорського ”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

**ЗВІТ**

лабораторної работи №2

курсу «Інфраструктура Програмного забезпечення WEB - застосувань »

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

Перевірив: Виконали студенти:

Орленко С. П. Гр. ІП-01

Смислов Даніл

Хернуф Валід

Пасальський Олександр

Київ 2023

**ЛАБОРАТОРНА РОБОТА № 2**

**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.

SPA = 19 + 8 + 16 = 43 = 8043

**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.

**Програма проведення експерименту ЛР 2**

1. Прокинемо порти. В команду docker run додати такі ключі, щоб на сайт контейнера можна було зайти через 127.0.0.1:\*\*. Вивести інформацію по активним контейнерам. Після отримання результату, потрібно зафіксувати його у вигляді скриншоту. Результат додати до звіту.

Для перевірки результату через браузер гостьової машини потрібно звернутися до відповідної адреси і порту, а саме перейти на 127.0.0.1:; у браузері. Після отримання результату, потрібно зафіксувати його у вигляді скриншоту. Результат додати до звіту.

2. Замінити контент дефолтної сторінки nginx на власний – перелік ПІБ всіх членів бригади + поточна дата створення image. Замінити через ADD/COPY та, за потреби RUN, у Dockerfile. Створити НОВИЙ image NAME=lab01\_2b на базі імейджу lab01\_1brxxx. Зайти на сторінку через браузер. Приклад показаний.

3. Замінити контент дефолтної сторінки nginx на власний, але розшарений з каталогу ./lab01 – односторінковий WEB-застосунок, який презентуватиме склад бригади (з вказанням хобі, тощо), який але створіть файл у папці ./lab01 і зробити так, щоб через VOLUME цей файл ставав сторінкою. Створити НОВИЙ image NAME=lab01\_3brxxx на базі імейджу lab01\_1brxxx. Зайти на сторінку через браузер. Після отримання результату, потрібно зафіксувати його у вигляді скриншоту. Результат додати до звіту.

4. Сформувати звіт.

**Хід роботи**

1. Прокинемо порти. В команду docker run додамо такі ключі, щоб на сайт контейнера можна було зайти через 127.0.0.1:8043. Виведемо інформацію по активним контейнерам. Приклад виконання показаний на рис.1.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 1 – Запуск контейнера з прокиданням портів.

Для перевірки результату, через браузер гостьової машини звернемося до потрібної адреси і порту 127.0.0.1:8043 (рис.2.). В браузері маємо сторінку, показано на рис. 3.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2 – Приклад запиту.

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 3 – Вікно браузера.

2. Замінимо контент дефолтної сторінки nginx на власний – перелік ПІБ всіх членів бригади + поточна дата створення image. Використаємо команду RUN у Dockerfile для встановлення нового вмісту index.html (рис. 5.). Створимо новий image NAME=lab01\_2bshp на базі імейджу lab01\_1brshp (рис. 6). Запустимо його (Рис. 7.) Зайдемо на сторінку через браузер (Рис. 8.).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4 – Перегляд вмісту файлу index.html в контейнері.

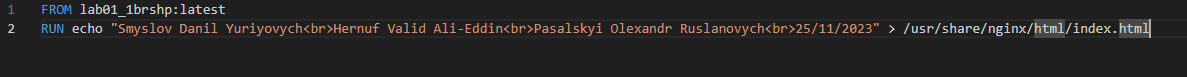


Рисунок 5 – Команда в DockerFile для зміни вмісту index.html file під час створення контейнеру.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6 – Створення нового Image.

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

Рисунок 7 – Запуск нового Image.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 8 – Сторінка з новим вмістом.

3. Замінимо контент дефолтної сторінки nginx на власний, який презентуватиме склад бригади (з вказанням хобі, тощо), але створемо файл у папці ./lab01 і зробимо так, щоб через VOLUME цей файл ставав сторінкою (Рис. 9.). Створимо новий image NAME=lab01\_3brshp на базі імейджу lab01\_1brshp (Рис. 10.). Зайдемо на сторінку через браузер (Рис. 11.).

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 9 – Команда в DockerFile для зміни вмісту index.html file під час створення контейнеру на файл що розташований в папці ./lab01.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 10 – Створимо і запустимо новий контейнер з Image lab01\_3bshp.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 11 – Сторінка в браузері .

**Висновок**

Під час виконання лабораторної роботи з дослідження спільних ресурсів хостової та гостьових систем в Docker, ми успішно вивчили та застосували різні можливості для встановлення з'єднання між контейнерами та хостовою машиною.

У першому етапі експерименту було використано ключі команди docker run для прокидання портів, дозволяючи доступ до веб-сервера nginx у контейнері через адресу та порт з хостової машини. У другому та третьому завдання були створені нові Docker образи з власним вмістом, сторінки nginx, використано механізм VOLUME та COPY для розшарювання каталогу хостової машини для контейнеру Docker, та відображення WEB-застосунку. Результати були наведені у вигляді скріншотів.

В результаті лабораторної роботи ми отримали практичний досвід використання Docker для роботи з спільними ресурсами, а також вміння налаштовувати та оптимізувати вміст контейнерів відповідно до наших потреб.