

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 1

## Лабораторна робота № 4. Створення зразка веб-застосунку в Docker-контейнері

Метою даної лабораторної роботи є отримання базових навичок по роботі з Docker.

Завдання на лабораторну роботу

- Запустити DEVASC VM
- Створити простий bash-скрипт
- Створити зразок веб-застосунку
- Налаштувати веб-застосунок для використання файлів веб-сайту
- Створити скрипт bash для збірки та запуску контейнера Docker
- Зібрати, запустити і перевірити контейнер Docker.

### Хід роботи:

#### 1. Запустити DEVASC VM.

#### 2. Створити простий bash-скрипт

Знання Bash має вирішальне значення для роботи з неперервною інтеграцією, неперервним розгортанням, контейнерами та з середовищем розробки. Скрипти Bash допомагають автоматизувати різноманітні завдання в одному файлі скрипта.

##### 2.1 Створіть порожній файл скрипту bash.

Змініть свій робочий каталог на `~/labs/devnet-src/sample-app` і додайте новий файл під назвою **user-input.sh**

```
devasc@labvm:~$ cd labs/devnet-src/sample-app/  
devasc@labvm:~/labs/devnet-src/sample-app$ touch user-input.sh
```

##### 2.2 Відкрийте файл у текстовому редакторі nano.

Використайте команду **nano**, щоб відкрити текстовий редактор nano

```
devasc@labvm:~/labs/devnet-src/sample-app$ nano user-input.sh
```

##### 2.3 Додайте 'she-bang' у верхній частині скрипту.

Звідси ви можете вводити команди для вашого скрипту bash. Використовуйте клавіші зі стрілками для навігації в **nano**. Зверніть увагу на команди,

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 2

розташовані внизу (тут не показано) для керування файлом. Символ каретки (^) означає, що ви використовуєте клавішу CTRL або Command на клавіатурі.

Наприклад, щоб вийти з **nano**, введіть CTRL+X.

Додайте 'she-bang', який повідомляє системі, що цей файл містить команди, які потрібно запускати в оболонці bash.

```
#!/bin/bash
```

#### 2.4 Додайте прості команди bash до скрипту.

Введіть декілька простих команд bash для вашого скрипту. Наступні команди запитають у користувача ім'я, встановлять його до змінної **UserName**, і виведуть рядок тексту з ім'ям користувача.

```
echo -n "Enter Your Name: "  
read userName  
echo "Your name is $userName."
```

#### 2.5 Вийдіть з nano і збережіть ваш скрипт.

Натисніть сполучення клавіш **CTRL+X**, потім **Y**, а потім **ENTER**, щоб вийти з **nano** і зберегти скрипт.

#### 2.6 Запустіть ваш скрипт з командного рядка

Ви можете запустити його безпосередньо з командного рядка, використовуючи таку команду.

```
devasc@labvm:~/labs/devnet-src/sample-app$ bash user-input.sh  
Enter Your Name: Bob  
Your name is Bob.  
devasc@labvm:~/labs/devnet-src/sample-app$
```

#### 2.7 Змініть режим скрипту на виконуваний файл для всіх користувачів.

Змініть режим роботи скрипта на виконуваний за допомогою команди **chmod**.

Встановіть параметри на **a+x**, щоб скрипт виконувався (x) усіма користувачами (a). Після використання **chmod** зверніть увагу, що дозволи були змінені для користувачів, груп та інших осіб, додалось «x» (виконуваний файл).

```
devasc@labvm:~/labs/devnet-src/sample-app$ ls -l user-input.sh  
-rw-rw-r-- 1 devasc devasc 84 Jun  7 16:43 user-input.sh  
  
devasc@labvm:~/labs/devnet-src/sample-app$ chmod a+x user-input.sh  
  
devasc@labvm:~/labs/devnet-src/sample-app$ ls -l user-input.sh  
-rwxrwxr-x 1 devasc devasc 84 Jun  7 16:43 user-input.sh
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б / -2021
	Екземпляр № 1	Арк 11 / 3

## 2.8. Переіменуйте файл, щоб видалити розширення `.sh` .

Ви можете переіменувати файл, щоб видалити розширення, щоб користувачам не потрібно було додавати `.sh` до команди для виконання скрипта.

```
devasc@labvm:~/labs/devnet-src/sample-app$ mv user-input.sh user-input
```

## 2.9. Виконайте скрипт з командного рядка.

Тепер скрипт можна запускати з командного рядка без команди `source` або розширення. Щоб запустити скрипт `bash` без команди `source`, вам необхідно поставити перед скриптом `«./»`.

```
devasc@labvm:~/labs/devnet-src/sample-app$ ./user-input
Enter Your Name: Bob
Your name is Bob.
devasc@labvm:~/labs/devnet-src/sample-app$
```

## 3. Створити зразок веб-застосунку

Перш ніж ми зможемо запустити застосунок в контейнері `Docker`, ми спочатку повинні його мати. В цій частині ви буде створено простий скрипт `Python`, який відобразить `IP`-адресу клієнта, коли клієнт відвідує веб-сторінку.

### 3.1 Встановіть `Flask` і відкрийте порт на брандмауері `VM DEVASC`.

Розробники веб-застосунків, які використовують `Python`, зазвичай використовують фреймворки. `Framework` — це бібліотека коду, яка спрощує розробникам створення надійних, масштабованих і прийнятних для підтримки веб-застосунків. `Flask` — це веб-застосунок, написаний на `Python`. Інші фреймворки це `Tornado` і `Pyramid`.

Ви будете використовувати цей фреймворк для створення зразка веб-застосунку. `Flask` отримує запити, а потім надає відповідь користувачеві у веб-застосунку. Це зручно для динамічних веб-застосунків, оскільки дозволяє взаємодіяти з користувачем та динамічним вмістом. Те, що робить ваш зразок веб-застосунку динамічним, полягає в тому, що він буде відображати `IP`-адресу клієнта.

Відкрийте вікно терміналу та імпортуйте `flask`.

```
devasc@labvm:~/labs/devnet-src/sample-app$ pip3 install flask
```

### 3.2 Відкрийте файл `sample_app.py`

Відкрийте файл `sample_app.py`, розташований в каталозі `/sample-app`. Ви можете зробити це всередині `VS Code` або ви можете використовувати текстовий редактор командного рядка, наприклад `nano` або `vim`.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б / -2021
	Екземпляр № 1	Арк 11 / 4

### 3.3 Додайте команди для імпортування методів з flask.

Додайте наведені нижче команди, щоб імпортувати потрібні методи з бібліотеки flask.

```
from flask import Flask
from flask import request
```

### 3.4 Створіть екземпляр класу Flask.

Створіть екземпляр класу Flask і назвіть його **sample**. Обов'язково використовуйте два символи підкреслення перед і після «name».

```
sample = Flask(__name__)
```

### 3.5 Визначте метод відображення IP-адреси клієнта.

Потім налаштуйте Flask таким чином, щоб коли користувач відвідуватиме сторінку за замовчуванням (кореневий каталог), відображалось повідомлення з IP-адресою клієнта.

```
@sample.route("/")
def main():
    return "You are calling me from " + request.remote_addr + "\n"
```

Зверніть увагу на оператор Flask `@sample.route(«/»)`. Такі фреймворки, як Flask, використовують метод маршрутизації (`.route`) для посилення на URL-адресу застосунку (це не слід плутати з мережною маршрутизацією). Тут `«/»` (кореневий каталог) прив'язаний до функції `main()`. Отже, коли користувач переходить до URL-адреси `http://localhost:8080/` (кореневий каталог), результат оператора `return` буде відображений у браузері.

### 3.6 Налаштуйте застосунок для запуску локально.

Нарешті, налаштуйте Flask для запуску застосунку локально за адресою `http://0.0.0.0:8080`, яка аналогічна `http://localhost:8080`. Обов'язково використовуйте два символи підкреслення до і після «name», а також до і після «main».

```
if __name__ == "__main__":
    sample.run(host="0.0.0.0", port=8080)
```

### 3.7 Збережіть і запустіть зразок веб-застосунку.

Збережіть свій скрипт і запустіть його з командного рядка. Ви маєте побачити наступний результат, який вказує на те, що ваш сервер «sample-app» працює. Якщо ви не бачите наведеного нижче результату або отримуєте повідомлення

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 5

про помилку, уважно перевірте скрипт `sample_app.py`.

```
devasc@labvm:~/labs/devnet-src/sample-app$ python3 sample_app.py
* Serving Flask app "sample-app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

### 3.8 Перевірте, чи працює сервер.

Ви можете перевірити чи сервер працює одним із двох способів.

a. Відкрийте веб-переглядач Chromium і введіть в поле URL `0.0.0.0:8080`. Ви маєте отримати наступний результат:

**You are calling me from 127.0.0.1**

Якщо ви отримали відповідь “HTTP 400 Bad Request”, уважно перевірте скрипт `sample_app.py`.

b. Відкрийте інше вікно терміналу та скористайтесь інструментом URL-адреси командного рядка (cURL), щоб перевірити відповідь сервера.

```
devasc@labvm:~/labs/devnet-src/sample-app$ curl http://0.0.0.0:8080
You are calling me from 127.0.0.1
devasc@labvm:~/labs/devnet-src/sample-app$
```

### 3.9 Зупиніть сервер.

Поверніться до вікна терміналу, де запущено сервер, і натисніть комбінацію клавіш `CTRL+C`, щоб зупинити сервер.

## 4. Налаштування веб-застосунку для використання файлів веб-сайту

В цій частині створіть зразок веб-застосунку, включивши до нього сторінку `index.html` та специфікацію `style.css`. Як правило, `index.html` - це перша сторінка, яка завантажується до веб-переглядача клієнта під час відвідування вашого веб-сайту. `style.css` - це таблиця стилів, яка використовується для налаштування зовнішнього вигляду веб-сторінки.

### 4.1 Перегляньте каталоги, які використовуватимуться веб-застосунком.

Каталоги `templates` та `static` вже знаходяться в каталозі `sample-app`. Відкрийте `index.html` і `style.css`, щоб переглянути їх вміст. Якщо ви вмієте працювати з HTML і CSS, сміливо налаштовуйте ці каталоги і файли так, як вам це зручно.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 6

Проте, обов'язково збережіть вбудований код `{{request.remote_addr}}` Python у файлі **index.html**, оскільки це динамічний елемент зразка веб-застосунку.

```
devasc@labvm:~/labs/devnet-src/sample-app$ cat templates/index.html
<html>
<head>
  <title>Sample app</title>
  <link rel="stylesheet" href="/static/style.css" />
</head>
<body>
  <h1>You are calling me from {{request.remote_addr}}</h1>
</body>
</html>
devasc@labvm:~/labs/devnet-src/sample-app$ cat static/style.css
body {background: lightsteelblue;}
devasc@labvm:~/labs/devnet-src/sample-app$
```

## 4.2 Оновіть код Python для зразка веб-застосунку.

Після того, як ви вивчили основні файли веб-сайтів, вам потрібно оновити файл **sample\_app.py** так, щоб він відображав файл **index.html**, а не просто повертав дані. Створення HTML-вмісту за допомогою коду Python може бути громіздким, особливо при використанні умовних операторів або повторюваних структур. Файл HTML може бути відтворений у Flask автоматично за допомогою функції **render\_template**. Для цього потрібно імпортувати метод **render\_template** з бібліотеки `flask` та відредагувати його у функцію **return**. Внесіть виділені зміни у свій скрипт.

```
from flask import Flask
from flask import request
from flask import render_template

sample = Flask(__name__)

@sample.route("/")
def main():
    return render_template("index.html")

if __name__ == "__main__":
    sample.run(host="0.0.0.0", port=8080)
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б / -2021
	Екземпляр № 1	Арк 11 / 7

### 4.3 Збережіть і запустіть ваш скрипт.

Збережіть і запустіть ваш **sample-app.py** скрипт. Ви повинні отримати результат, подібний до наведеного нижче:

```
devasc@labvm:~/labs/devnet-src/sample-app$ python3 sample_app.py
* Serving Flask app "sample-app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

Примітка: Якщо ви отримали результат Traceback і помилку з повідомленням на кшталт: **OSError: [Errno 98] Address already in use**, то це означає, що ви не завершили роботу свого попереднього сервера. Поверніться до вікна терміналу, де запущений цей сервер, і натисніть CTRL+C, щоб завершити процес сервера. Повторно запустіть ваш скрипт.

### 4.4 Переконайтеся, що ваша програма запущена.

Знову ж таки, ви можете переконаватися, що ваша програма працює одним із двох способів.

- Відкрийте веб-переглядач Chromium і введіть в поле URL 0.0.0.0:8080. Ви маєте отримати той самий результат, що і раніше. Однак ваш фон буде світло-сталевий синій, а текст буде у форматі - H1.

#### You are calling me from 127.0.0.1

- Відкрийте інше вікно терміналу та використайте команду **curl** для перевірки відповіді сервера. Тут ви побачите результат автоматичного відтворення HTML-коду за допомогою функції `render_template`. В цьому випадку ви отримаєте весь HTML-контент. Однак динамічний код Python буде замінений із значенням для `{{request.remote_addr}}`. Крім того, зверніть увагу, що ваше запрошення буде в тому ж рядку, що і останній рядок HTML-виведення. Натисніть клавішу ENTER, щоб перейти на новий рядок.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 8

```
devasc@labvm:~/labs/devnet-src/sample-app$ curl http://0.0.0.0:8080
<html>
<head>
  <title>Sample app</title>
  <link rel="stylesheet" href="/static/style.css" />
</head>
<body>
  <h1>You are calling me from 127.0.0.1</h1>
</body>
</html>devasc@labvm:~/labs/devnet-src/sample-app$
devasc@labvm:~/labs/devnet-src/sample-app$
```

#### 4.5 Зупиніть сервер.

Поверніться до вікна терміналу, де запущений сервер, і натисніть комбінацію клавіш CTRL+C, щоб зупинити сервер.

### 5: Створення скрипту Bash для збірки та запуску контейнера Docker

Застосунок можна розгорнути на сервері з «голового заліза» (фізичний сервер, виділений для одно-клієнтського середовища ) або на віртуальній машині, яку ви щойно зробили у попередній частині. Його також можна розгорнути у контейнерному рішенні, такому як Docker. В цій частині ви створите скрипт bash і додасте до нього команди, які виконують наступні завдання для збірки та запуску контейнера Docker:

- Створення тимчасових каталогів для зберігання файлів веб-сайту.
- Копіювання каталогів веб-сайту та sample\_app.py до тимчасового каталогу.
- Збірка Dockerfile.
- Збірка контейнера Docker.
- Запуск контейнера і перевірка його роботи.

#### 5.1 Створіть тимчасові каталоги для зберігання файлів веб-сайту.

Відкрийте файл скрипту bash sample-app.sh в каталозі ~/labs/devnet-src/sample-app. Додайте 'she-bang' і команди, щоб створити структуру каталогів з tempdir як батьківський каталог.

```
#!/bin/bash

mkdir tempdir
mkdir tempdir/templates
mkdir tempdir/static
```

#### 5.2 Скопіюйте каталоги веб-сайту та sample\_app.py у тимчасовий каталог.

В файл sample-app.sh додайте команди для копіювання каталогу веб-сайту та



Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 9

скрипту в **tempdir**.

```
cp sample_app.py tempdir/.
cp -r templates/* tempdir/templates/.
cp -r static/* tempdir/static/.
```

### 5.3 Виконайте збірку Dockerfile.

На цьому кроці ви введете необхідні команди **bash echo** в файл **sample-app.sh** для створення Dockerfile в **tempdir**. Цей Dockerfile буде використаний для збірки контейнера.

- Вам потрібно запустити Python в контейнері, тому додайте команду Docker **FROM**, щоб встановити Python в контейнер.

```
echo "FROM python" >> tempdir/Dockerfile
```

- Для вашого скрипту **sample\_app.py** необхідний Flask, тому додайте команду Docker **RUN**, щоб встановити Flask у контейнер.

```
echo "RUN pip install flask" >> tempdir/Dockerfile
```

- Для запуску застосунку вашому контейнеру знадобляться папки веб-сайту та скрипт **sample\_app.py**, тому додайте команди Docker **COPY**, щоб додати їх до каталогу в контейнері Docker. В цьому прикладі ви створите **/home/myapp** як батьківський каталог всередині контейнера Docker. Окрім копіювання файлу **sample\_app.py** у Dockerfile, ви також скопіюєте файл **index.html** із каталогу **templates** та файл **style.css** із каталогу **static**

```
echo "COPY ./static /home/myapp/static/" >> tempdir/Dockerfile
echo "COPY ./templates /home/myapp/templates/" >> tempdir/Dockerfile
echo "COPY sample_app.py /home/myapp/" >> tempdir/Dockerfile
```

- За допомогою команди Docker **EXPOSE** відкрийте порт 8080 для використання веб-сервером.

```
echo "EXPOSE 8080" >> tempdir/Dockerfile
```

- І на кінець, додайте команду Docker **CMD** для виконання скрипту Python.

```
echo "CMD python3 /home/myapp/sample_app.py" >> tempdir/Dockerfile
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б / -2021
	Екземпляр № 1	Арк 11 / 10

#### 5.4 Виконайте збірку контейнера Docker.

Додайте команди до файлу **sample-app.sh**, щоб перейти до каталогу **tempdir** і виконати збірку контейнера Docker. Параметр **-t** команди **docker build** дозволяє вказати ім'я контейнера, а точка вкінці (.) вказує, що ви хочете, щоб контейнер був вбудований у поточний каталог.

```
cd tempdir
docker build -t sampleapp .
```

#### 5.5 Запустіть контейнер і перевірте, чи він запущений.

а. Додайте команду **docker run** до файлу **sample-app.sh**, щоб запустити контейнер.

```
docker run -t -d -p 8080:8080 --name samplerunning sampleapp
```

Параметри **docker run** вказують на наступне:

- **-t** вказує на те, що ви хочете створити термінал для контейнера, щоб ви могли отримати до нього доступ у командному рядку.
- **-d** вказує на те, що ви хочете, щоб контейнер працював у фоновому режимі та друкував ідентифікатор контейнера під час виконання команди **docker ps -a**
- **-p** вказує на те, що ви хочете опублікувати внутрішній порт контейнера на вузлі. Перший "8080" посилається на порт для застосунку, що працює в контейнері docker (наш sample-app). Другий "8080" вказує docker використовувати цей порт на вузлі. Ці значення не повинні обов'язково бути однаковими. Наприклад, внутрішній порт 80 на зовнішній 800 (**80:800**).
- **--name** вказує спочатку те, що ви хочете викликати екземпляр контейнера (**samplerunning**), а потім образ контейнера, на якому буде заснований екземпляр (**sampleapp**). Ім'я екземпляра може бути довільним. Натомість ім'я образу має співпадати з іменем контейнера, яке ви вказали в команді **docker build (sampleapp)**.

б. Додайте команду **docker ps -a**, щоб відобразити всі запущені в даний момент контейнери Docker.

Ця команда буде останньою, виконаною скриптом **bash**.

```
docker ps -a
```

#### 5.6 Збережіть ваш **bash** скрипт.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 11

## 6. Збірка, запуск та перевірка контейнера Docker

### 6.1 Виконайте скрипт bash.

Виконайте скрипт `bash` з командного рядка. Ви маєте отримати результат, подібний до наведеного нижче. Після створення каталогів `tempdir`, скрипт виконує команди для збірки контейнера Docker. Зверніть увагу, що на Кроці 7/7 у вихідних даних виконується `sample_app.py`, який створює веб-сервер. Крім того, зверніть увагу на ідентифікатор контейнера. Ви побачите це в командному рядку Docker пізніше в лабораторній роботі.

```
devasc@labvm:~/labs/devnet-src/sample-app$ bash ./sample-app.sh
Sending build context to Docker daemon 6.144kB
Step 1/7 : FROM python
latest: Pulling from library/python
90fe46dd8199: Pulling fs layer
35a4f1977689: Pulling fs layer
bbc37f14aded: Pull complete
74e27dc593d4: Pull complete
4352dcff7819: Pull complete
deb569b08de6: Pull complete
98fd06fa8c53: Pull complete
7b9cc4fdefe6: Pull complete
512732f32795: Pull complete
Digest: sha256:ad7fb5bb4770e08bf10a895ef64a300b288696a1557a6d02c8b6fba98984b86a
Status: Downloaded newer image for python:latest
--> 4f7cd4269fa9
Step 2/7 : RUN pip install flask
--> Running in 32d28026afea
Collecting flask
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
Collecting click>=5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
Collecting Jinja2>=2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting Werkzeug>=0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting MarkupSafe>=0.23
  Downloading MarkupSafe-1.1.1-cp38-cp38-manylinux1_x86_64.whl (32 kB)
Installing collected packages: click, MarkupSafe, Jinja2, Werkzeug, itsdangerous, flask
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 12

```

Successfully installed Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2
flask-1.1.2 itsdangerous-1.1.0
Removing intermediate container 32d28026afea
---> 619aee23fd2a
Step 3/7 : COPY ./static /home/myapp/static/
---> 15facl237eec
Step 4/7 : COPY ./templates /home/myapp/templates/
---> dc807b5cf615
Step 5/7 : COPY sample_app.py /home/myapp/
---> d4035a63ae14
Step 6/7 : EXPOSE 8080
---> Running in 40c2d35aa29a
Removing intermediate container 40c2d35aa29a
---> eb789099a678
Step 7/7 : CMD python3 /home/myapp/sample_app.py
---> Running in 41982e2c6209
Removing intermediate container 41982e2c6209
---> a2588e9b0593
Successfully built a2588e9b0593
Successfully tagged sampleapp:latest
8953a95374ff8ebc203059897774465312acc8f0ed6abd98c4c2b04448a56ba5
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS              NAMES
8953a95374ff      sampleapp          "/bin/sh -c 'python ..." 1 second ago
Up Less than a second 0.0.0.0:8080->8080/tcp  samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$

```

## 6.2 Дослідіть запущений Docker контейнер і веб-застосунок

а. Створення каталогів **tempdir** не відображається у вихідних даних скрипта. Ви можете додати команди **echo** для друку повідомлення, коли вони успішно створені. Ви також можете перевірити їх наявність за допомогою команди **ls**. Пам'ятайте, що цей каталог має файли та папки, які використовуються для збірки контейнера та запуску веб-застосунку. Це не той контейнер, який був побудований.

```

devasc@labvm:~/labs/devnet-src/sample-app$ ls tempdir/
Dockerfile sample_app.py static templates
devasc@labvm:~/labs/devnet-src/sample-app$

```

б. Зверніть увагу на створений вашим скриптом **bash Dockerfile**. Відкрийте цей файл, щоб побачити, як він виглядає в кінцевому варіанті без команд **echo**.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 13

```
devasc@labvm:~/labs/devnet-src/sample-app$ cat tempdir/Dockerfile
FROM python
RUN pip install flask
COPY ./static /home/myapp/static/
COPY ./templates /home/myapp/templates/
COPY sample_app.py /home/myapp/
EXPOSE 8080
CMD python3 /home/myapp/sample_app.py
```

с. Виведення команди `docker ps -a` може бути важко прочитати, залежно від ширини дисплея терміналу. Ви можете переспрямувати його в текстовий файл, де ви зможете краще переглядати його без перенесення слів.

```
devasc@labvm:~/labs/devnet-src/sample-app$ docker ps -a >> running.txt
devasc@labvm:~/labs/devnet-src/sample-app$
```

d. Контейнер Docker створює власну IP-адресу з адресного простору приватної мережі. Переконайтеся, що веб-застосунок запущений та повідомляє IP-адресу. У веб-переглядачі за адресою **http://localhost:8080**, ви маєте побачити повідомлення: **You are calling me from 172.17.0.1** у форматі H1 на світло-сталево-синьому фоні. Ви також можете скористатися командою `curl`, якщо захочете.

```
devasc@labvm:~/labs/devnet-src/sample-app$ curl http://172.17.0.1:8080
<html>
<head>
  <title>Sample app</title>
  <link rel="stylesheet" href="/static/style.css" />
</head>
<body>
  <h1>You are calling me from 172.17.0.1</h1>
</body>
</html>
devasc@labvm:~/labs/devnet-src/sample-app$
devasc@labvm:~/labs/devnet-src/sample-app$
```

e. За замовчуванням Docker для створення мереж контейнерів використовує підмережу IPv4 172.17.0.0/16. (Цю адресу за необхідності можна змінити.) Введіть `ip address` команду, щоб відобразити всі IP-адреси, які використовуються вашим екземпляром віртуальної машини DEVASC. Ви маєте побачити адресу зворотної петлі 127.0.0.1, яку раніше використовував веб-застосунок в лабораторній роботі, та новий інтерфейс Docker з IP-адресою 172.17.0.1.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 14

```

devasc@labvm:~/labs/devnet-src/sample-app$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
<output omitted>
4: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default
    link/ether 02:42:c2:d1:8a:2d brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
    inet6 fe80::42:c2ff:fed1:8a2d/64 scope link
        valid_lft forever preferred_lft forever
<output omitted>

```

### 6.3 Отримайте доступ та дослідіть запущений контейнер.

Пам'ятайте, що контейнер Docker - це спосіб інкапсулювати все, що потрібно для запуску вашого застосунку для того, щоб його можна було легко розгорнути в різних середовищах - не лише у вашій віртуальній машині DEVASC.

а. Щоб отримати доступ до запущеного контейнера, введіть команду **docker exec -it** із зазначенням імені запущеного контейнера (`samplerunning`) та потрібної оболонки `bash (/bin/bash)`. Параметр `-i` вказує на те, що ви хочете, щоб він був інтерактивним, а параметр `-t` визначає те, що ви хочете отримати доступ до терміналу. Запрошення змінюється на **root@containerID**. Ваш ідентифікатор контейнера буде відрізнятися від наведеного нижче. Зверніть увагу на те, що ідентифікатор контейнера співпадає з ідентифікатором, зазначеним у вихідних даних з **docker ps -a**.

```

devasc@labvm:~/labs/devnet-src/sample-app$ docker exec -it samplerunning
/bin/bash
root@8953a95374ff:/#

```

б. Тепер ви маєте кореневий доступ для контейнера Docker **samplerunning**. Звідси ви можете використовувати команди Linux для дослідження контейнера Docker. Введіть **ls**, щоб побачити структуру каталогів на кореновому рівні.

```

root@8953a95374ff:/# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root  sbin  sys  usr
root@8953a95374ff:/#

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 15

с. Нагадаємо, що у своєму скрипті `bash` ви додали команди до `Dockerfile`, які скопіювали каталоги та файли вашого застосунку в каталог `home/myapp`. Введіть ще раз команду `ls` для цієї папки, щоб переглянути ваш скрипт `sample_app.py` та каталоги. Щоб краще зрозуміти, що входить до вашого контейнера `Docker`, ви можете використати команду `ls` для вивчення інших каталогів, таких як `/etc` та `/bin`.

```
root@8953a95374ff:/# ls home/myapp/
sample_app.py static templates
root@8953a95374ff:/#
```

d. Вийдіть із контейнера `Docker`, щоб повернутися до командного рядка віртуальної машини `DEVASC`.

```
root@8953a95374ff:/# exit
exit
devasc@labvm:~/labs/devnet-src/sample-app$
```

## 6.4 Зупиніть та видаліть контейнер `Docker`.

a. Ви можете зупинити контейнер `Docker` командою `docker stop`, вказавши ім'я запущеного контейнера. Очищення та кешування контейнера займе кілька секунд. Ви можете переконаватися в тому, що він все ще існує, ввівши команду `docker ps -a`. Однак, якщо ви оновите веб-сторінку для `http://localhost:8080`, то побачите, що веб-застосунок більше не працює.

```
devasc@labvm:~/labs/devnet-src/sample-app$ docker stop samplerunning
samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS              NAMES
df034cb53e72      sampleapp          "/bin/sh -c 'python ..." 49 minutes ago
Exited (137) 20 seconds ago      samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$
```

b. Ви можете перезапустити зупинений контейнер за допомогою команди `docker start`. Контейнер відразу ж запуститься.

```
devasc@labvm:~/labs/devnet-src/sample-app$ docker start samplerunning
samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 16

с. Щоб назавжди видалити контейнер, спочатку зупиніть його, а потім видаліть командою **docker rm**. Ви завжди зможете відновити його знову, виконавши програму **sample-app**. За допомогою команди **docker ps -a** переконайтеся, що контейнер було видалено.

```

devasc@labvm:~/labs/devnet-src/sample-app$ docker stop samplerunning
samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$ docker rm samplerunning
samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS              NAMES
devasc@labvm:~/labs/devnet-src/sample-app$

```

Запишіть висновки по виконаній роботі.