

Практична робота 2

Робота із шарами та створення власного базового образу

Мета:

Створення власного базового Docker образу для веб-сторінки, використовуючи декілька додаткових інструкцій у Dockerfile, які дозволять краще контролювати процес збірки образу та його конфігурацію.

Передумови:

Встановлений Docker Desktop для Windows.

Базове розуміння роботи з Docker, включаючи збірку образів і запуск контейнерів.

Кроки:

Крок 1. Запуск Docker Desktop:

Після встановлення запустіть Docker Desktop. Переконайтеся, що програма запущена та працює без помилок.

Крок 2. Створення Dockerfile

Створіть нову папку для проекту на вашому комп'ютері.

Всередині цієї папки створіть файл без розширення з іменем Dockerfile.

Відкрийте Dockerfile у текстовому редакторі і додайте наступний вміст:

```
FROM ubuntu:latest
LABEL maintainer="yourname@example.com"
RUN apt-get update && apt-get install -y apache2
ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
COPY ./index.html /var/www/html/
EXPOSE 80
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

Тут ми створюємо образ, базуючись на останній версії Ubuntu, встановлюємо Apache веб-сервер, встановлюємо необхідні змінні оточення, копіюємо нашу веб-сторінку у відповідну директорію і налаштовуємо образ на використання порту 80.

Роз'яснення використаних інструкцій:

FROM визначає базовий образ. У цьому випадку, ubuntu:latest.

LABEL додає метадані до образу, наприклад, електронну пошту розробника.

RUN виконує команди в новому шарі над поточним образом і зберігає результат як новий шар.

ENV встановлює змінні оточення, які можуть бути використані в інших інструкціях Dockerfile або в контейнері під час його виконання.

COPY копіює нові файли або директорії з вихідного розташування і додає їх до файлової системи контейнера.

EXPOSE інформує Docker про те, що контейнер слухає на певних портах в мережі.

CMD вказує команду та її параметри, які будуть виконані при запуску контейнера. В даному випадку, запускаємо Apache у передньому плані.

Крок 3. Створення простої веб-сторінки

У тій же папці, де знаходиться ваш Dockerfile, створіть файл index.html.

Відкрийте index.html у текстовому редакторі і додайте простий HTML-код.

Наприклад:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Друга лабораторна робота</title>
  </head>
  <body>
    <h1>Вітаю в Docker!</h1>
    <p>Це моя друга лабораторна робота з хмарних технологій, яка демонструє створення власного базового образу.</p>
  </body>
</html>
```

Крок 4. Збірка Docker образу

Відкрийте командний рядок або PowerShell у папці проекту.

Виконайте команду для збірки Docker образу з вашим веб-сервером:

```
docker build -t my-custom-web .
```

Тут my-custom-web – це назва вашого Docker образу.

Крок 5. Запуск Docker контейнера

Після успішної збірки образу запусить контейнер за допомогою наступної команди:

```
docker run -d -p 8080:80 my-custom-web
```

Ця команда запускає контейнер в фоновому режимі (-d), прив'язуючи порт 8080 на вашій машині до порту 80 у контейнері.

Крок 6. Перегляд веб-сторінки

Відкрийте веб-браузер і перейдіть за адресою <http://localhost:8080>.

Ви повинні побачити вашу просту веб-сторінку, розміщену в Docker контейнері.

Підсумок

Ви навчилися створювати власні базові образи Docker, використовуючи різноманітні інструкції у Dockerfile. Ви також навчились розрізняти різні шари образу та їх призначення, а також використовувати ці знання для ефективно збірки та управління Docker образами. Кожна інструкція, яку ви додали, відіграє ключову роль у створенні та налаштуванні образу, що робить цей процес гнучким та масштабованим.