

Практична робота 4

Додавання додаткового шару для запису в Docker-контейнері

Мета:

Розширимо нашу попередню роботу зі створенням власного базового образу та використаємо додавання додаткового шару для запису в контейнері. Створення нового Docker образу із цього запущеного контейнера, що дозволить дослідити процеси комітінгу змін у Docker образах та розуміння імутабельності образів.

Передумови:

Встановлений Docker Desktop для Windows.

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

Кроки:

Крок 1. Підготовка проекту:

Використовуйте ту саму папку проекту, що й для попередньої роботи.

Впевніться, що у вас є Dockerfile із вашим власним образом та файл index.html із веб-сторінкою.

Крок 2. Запуск контейнера з можливістю запису:

Запустіть контейнер із вашого власного образу, як було зроблено в попередній роботі, але цього разу забезпечте можливість запису у файлову систему контейнера:

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

Крок 3. Модифікація контенту в контейнері:

Використовуйте команду docker exec для входу в запущений контейнер:

```
docker exec -it my-web-container /bin/bash
```

У контейнері, створіть або модифікуйте файли, наприклад, додайте новий файл або змініть існуючий index.html. Це створить новий шар за запис в контейнері.

Крок 4. Створення нового образу із контейнера:

Вийдіть із контейнера, якщо це необхідно, використовуючи exit.

Створіть новий образ із поточного стану контейнера, використовуючи команду docker commit:

```
docker commit my-web-container my-custom-web:v2
```

Тут my-custom-web:v2 – назва нового образу з тегом v2, що вказує на його версію або стан.

Крок 5. Перевірка нового образу:

Запустіть новий контейнер із створеного образу, щоб перевірити зміни:

```
docker run -d -p 8081:80 --name my-new-web-container my-custom-web:v2
```

Відкрийте веб-браузер і перейдіть за адресою <http://localhost:8081> для перегляду веб-сторінки з нового контейнера.

Підсумок

Ви навчилися створювати власні базові образи Docker, зокрема в частині комітінгу змін у контейнерах та створенні нових образів із цих змін. Робота зі шарами запису в контейнері та розуміння імутабельності образів є важливими навичками для розробки ефективних Docker-застосунків та управління версіями.