

Опис завдання:

Розгорнути власну версію простого вебсервера у своєму локальному середовищі таким чином, щоб за запитом до IP-адреси учбової віртуальної машини на порт 8080 ми бачили текст "Hello from Student Name!"

Результат завдання:

Завантажити на сервер в окрему теку зі своїм ім'ям Dockerfile, index.html, а також текстовий файл з послідовністю кроків зроблених для виконання завдання (включаючи запуснені команди і їх короткий опис). Наприкінці додайте результат запуску команди.

```
curl http://$VIRTUAL_MACHINE_IP:8080
```

План

1. Створюємо власний простий файл index.html
2. Створюємо Dockerfile для розгортання NGINX
3. Будуємо образ NGINX зі створеного нами Dockerfile
4. Запускаємо контейнер з NGINX
5. Перевіряємо за допомогою curl результат та фіксуємо його
6. Завантажуємо результат (Dockerfile, index.html, + текстовий файл з послідовністю кроків зроблених для виконання завдання) на сервер у окрему теку зі своїм ім'ям

Попередньо маємо налаштовану VM з встановленим Docker, та прописані у файлі hosts ip адреси нашої VM. Також для цього завдання створюємо окрему теку lesson3

1. Створюємо власний простий файл index.html

Запускаємо VM, логінімося, запускаємо термінал, створюємо теку lesson3 та заходимо в неї

Команда:

```
mkdir lesson3
```

Відгук:

...

Створюємо файл index.html

Команда:

```
nano index.html
```

Відгук:

...

Наповнюємо файл вмістом згідно завдання:

```
<html>
  <head>
    <title> lesson 3 </title>
  </head>
  <body>
    <h1> Hello from Oleksandr!</h1>
  </body>
</html>
```

Зберігаємо наш файл зі змінами (Cntr+0, Cntr+X)

2. Створюємо Dockerfile для розгортання NGINX

Створюємо файл Dockerfile

Команда:

```
nano Dockerfile
```

Відгук:

...

Наповнюємо файл вмістом згідно завдання:

```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/index.html
```

Зберігаємо наш файл зі змінами (Cntr+0, Cntr+X)

3. Будуємо образ NGINX зі створеного нами Dockerfile

Команда:

```
sudo docker build -t my-nginx .
```

Відгук:

```
[+] Building 0.1s (7/7) FINISHED
docker:default
=> [internal] load build definition from Dockerfile
0.0s
=> => transferring dockerfile: 104B
0.0s
=> [internal] load metadata for docker.io/library/nginx:latest
0.0s
=> [internal] load .dockerignore
0.0s
=> => transferring context: 2B
0.0s
=> [internal] load build context
0.0s
```

```
=> => transferring context: 31B
0.0s
=> [1/2] FROM docker.io/library/nginx:latest
0.0s
=> CACHED [2/2] COPY index.html /usr/share/nginx/html/index.html
0.0s
=> exporting to image
0.0s
=> => exporting layers
0.0s
=> => writing image
sha256:e28ed6de6845971b60b30592de6ccb5f41e61aca99fe9aa7bdecabc9a940
2ad6e
0.0s
=> => naming to docker.io/library/my-nginx
```

4. Запускаємо контейнер з NGINX

Команда:

```
sudo docker run -d --name less3-nginx -p 8080:80 my-nginx
```

Відгук:

```
0d6cef0fa23df5435633a2eba9f5e1d2ac3c2bebc9e149099fae3a6430c8f278
```

Перевіряємо чи запустився наш контейнер

Команда:

```
sudo docker ps
```

Відгук:

CONTAINER ID	IMAGE	COMMAND	CREATED	NAMES
STATUS	PORTS			
0d6cef0fa23d	my-nginx	"/docker-entrypoint..."	2 minutes ago	
Up 2 minutes	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp			less3-nginx

Бачимо що контейнер запустився і nginx працює на порту 8080

5. Перевіряємо за допомогою curl результат та фіксуємо його

Команда:

```
curl http://localhost:8080
```

Відгук:

```
<html>
<head>
  <title> lesson 3 </title>
</head>
<body>
  <h1> Hello from Oleksandr!</h1>
</body>
</html>
```

Команда:

curl <http://10.0.2.15:8080>

Відгук:

```
<html>
  <head>
    <title> lesson 3 </title>
  </head>
  <body>
    <h1> Hello from Oleksandr!</h1>
  </body>
</html>
```

Домашнє завдання до уроку 3: Розгорнути власну версію простого вебсервера у своєму локальному середовищі таким чином, щоб за запитом до IP-адреси учбової віртуальної машини на порт 8080 ми бачили текст "Hello from Student Name!"

Студент: Олександр Болотов

Дата виконання завдання: 11.02.2025