

## Docker (Cisco DevNet Associate)

Homework Assignments

## Прокидання портів у Docker

### Завдання

Проект, над яким зараз працює ваша компанія, вимагає наявності в ньому контейнерів з apache, nginx і mysql. Перед вашою командою стойте завдання розгорнути контейнери і забезпечити доступ до них мережею.

#### Попередні вимоги

- Встановлена і базово сконфігурата ОС Ubuntu 18.04 LTS (Ubuntu 20.04 LTS);
- Встановлений пакет Docker на ОС Ubuntu 18.04 LTS (Ubuntu 20.04 LTS).

### 1. Підготовка контейнерів

- Завантажимо образи необхідних контейнерів:
- Apache — останній офіційний образ.

\$ docker pull httpd

- Nginx — останній офіційний образ.

\$ docker pull nginx

- MySQL — останній офіційний образ.

\$ docker pull mysql

- Перевіримо завантажені образи.

```
$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
httpd          latest    1132a4fc88fa  13 hours ago  143MB
mysql          latest    ecac195d15af  3 days ago   516MB
nginx          latest    87a94228f133  10 days ago  133MB
```

Рисунок 1

### 2. Запуск контейнерів із прокиданням портів

- Створимо контейнер з образу Apache і запустимо його. За умовою нам необхідно, щоб apache був доступний по порту 8081:

```
$ docker run -d --name apache --publish 8081:80 httpd
a504b035bb8b7e44e14c3d446ea19a980dd1cf03c4d54ccb858132882f9e522a
```

Рисунок 2

Опція --publish 8081:80 (-p 8081:80) — прокидання портів. Спочатку вказується порт (можна разом з іп) основної системи, потім — порт у контейнері.

- Запустимо nginx, він має бути доступний по 80 порту:

```
$ docker run -d --name nginx -p 80:80 nginx
cd8ea2a34495057d7c385bca274cd092f08b37b367d51c92031629245fbf4853
```

Рисунок 3

- Запустимо mysql, залишимо для нього стандартний порт:

```
$ docker run -dit --name MySQL -e MYSQL_ROOT_PASSWORD=secretpass -p 3306:3306 mysql
51f9981053bf1710a97784a56d7e0586dfb41c61259b83381d12b089ace52499
```

Рисунок 4

Для запуску цього контейнера необхідно задати пароль root. За допомогою опції -e.

- Переконаємося, що всі контейнери запущені і працюють:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NA
MES						
51f9981053bf	mysql	"docker-entrypoint.s..."	16 minutes ago	Up 16 minutes	0.0.0.0:3306->3306/tcp, 33060/tcp	My
SQL						
cd8ea2a34495	nginx	"/docker-entrypoint..."	35 minutes ago	Up 35 minutes	0.0.0.0:80->80/tcp	ng
inx						
95a4dd7e763e	httpd	"httpd-foreground"	35 minutes ago	Up 35 minutes	0.0.0.0:8080->80/tcp	ap
ache						

Рисунок 5

### 3. Перевірка доступності прокинутих портів

- Перевіримо доступність apache на потрібному порту SERVER\_IP:8080:

# It works!

Рисунок 6

- Перевіримо доступність nginx на потрібному порту SERVER\_IP:80:

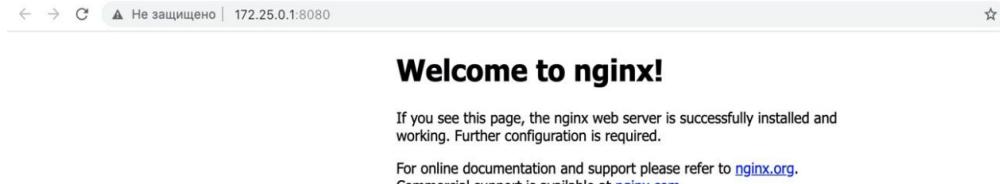


Рисунок 7

- Доступність MySQL перевіримо за допомогою CLI Linux. Як пароль користувача root використовуємо пароль, заданий під час запуску контейнера.

```
mysql@ubuntu-new-srv:~$ mysql -h 172.25.0.1 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.27 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

Рисунок 8

## Ваше завдання

- Для подальшого налаштування балансування на nginx, вам потрібно розгорнути ще два контейнери з apache. Прокинути зовнішні порти 8081 і 8082 всередину двох нових контейнерів.
- Перевірити доступ до двох створених контейнерів.
- Налаштювати балансування на nginx, у якому nginx виступає в ролі фронту, а три apache — в ролі бека.
- Знайти, на якому рисунку в цьому ДЗ є помилка).