Name: Shweta Singh
Sap Id: 500098159
Course & Batch: Btech CSE(DevOps)-B4

Submitted to: Dr Hitesh Kumar Sharma

EXPERIMENT 6

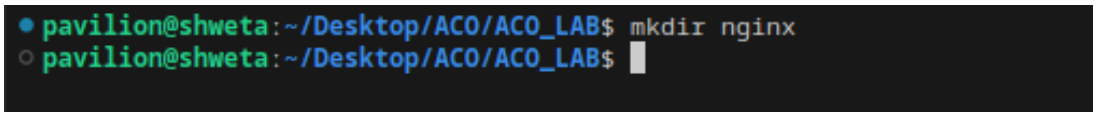
AIM: Working with Docker Compose File to Control Multiple Containers

Steps to Complete:

Creating compose files

- Create a directory named nginx in your root.

```
mkdir nginx
```



- Switch to that directory and create a file named docker-compose.yaml

```
cd nginx
```

```
vi docker-compose.yml
```

- Use docker-compose version 2 to create docker-compose.yaml file.
  Create a service named "databases". Use image named "mysql"

  Map container 3306 port to host machine 3306 port.

  Add environment variables named "MYSQL_ROOT_PASSWORD", "MYSQL_DATABASE", "MYSQL_USER" and "MYSQL_PASSWORD" along with corresponding values for all.

nginx > docker-compose.yml

```yaml
version: '2'
services:
  databases:
    image: mysql
    ports:
      - "3307:3306"
    env_file:
      - evs.env
  web:
    image: nginx
    ports:
      - "80:80"
    depends_on:
      - databases
```

```
cat evs.env
```

```
MYSQL_ROOT_PASSWORD=redhat08
MYSQL_DATABASE=nginxdb
MYSQL_USER=root
```

nginx > evs.env

```
MYSQL_ROOT_PASSWORD=redhat08
MYSQL_DATABASE=nginxdb
MYSQL_USER=root
MYSQL_PASSWORD=yourpassword
```

Add another service named "web"
Use image "nginx"

```
cat docker-compose.yml
```

```
pavilion@shweta:~/Desktop/ACO/ACO_LAB$ cd nginx/
pavilion@shweta:~/Desktop/ACO/ACO_LAB/nginx$ cat docker-compose.yml
version: '2'
services:
  databases:
    image: mysql
    ports:
      - "3307:3306"
    env_file:
      - evs.env
  web:
    image: nginx
    ports:
      - "80:80"
    depends_on:
      - databases
```

Running images using docker-compose

- Save docker-compose.yaml file and do docker-compose up.

```
docker-compose up -d
```

```
pavilion@shweta:~/Desktop/ACO/ACO_LAB/nginx$ docker-compose up -d
Creating network "nginx_default" with the default driver
Pulling databases (mysql:)...
latest: Pulling from library/mysql
8e0176adc18c: Pull complete
14e977b0f4b2: Pull complete
a7b58dd6f78b: Pull complete
fba70cc872a5: Pull complete
5db2cc6eab8f: Pull complete
081f41f573ba: Pull complete
86bf2dc4ded9: Pull complete
47f08b0e916e: Pull complete
850e29ae8eeb: Pull complete
13517fe0d921: Pull complete
Digest: sha256:f61944ff3f2961363a4d22913b2ac581523273679d7e14dd26e8db8c9f571a7e
Status: Downloaded newer image for mysql:latest
Creating nginx_databases_1 ... done
Creating nginx_web_1       ... done
pavilion@shweta:~/Desktop/ACO/ACO_LAB/nginx$
```

- Verify nginx service is up and is accessible on machine.

```
curl localhost:80
```

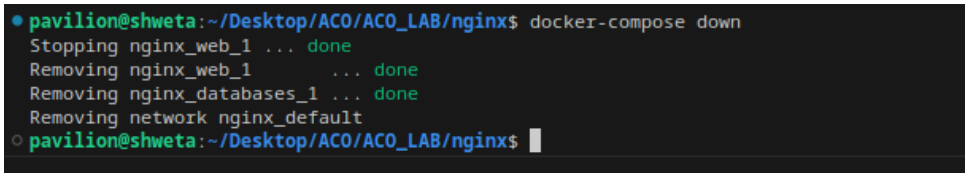
```
pavilion@shweta:~/Desktop/ACO/ACO_LAB/nginx$ curl localhost:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
pavilion@shweta:~/Desktop/ACO/ACO_LAB/nginx$
```

Stop and remove your docker container using docker-compose.

```
docker-compose down
```


pavilion@shweta:~/Desktop/ACO/ACO_LAB/nginx$ docker-compose down
Stopping nginx_web_1 ... done
Removing nginx_web_1 ... done
Removing nginx_databases_1 ... done
Removing network nginx_default
pavilion@shweta:~/Desktop/ACO/ACO_LAB/nginx$