

# 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.

`cat evs.env`

`MYSQL_ROOT_PASSWORD=redhat08`

`MYSQL_DATABASE=nginxdb`

`MYSQL_USER=root`

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

`cat docker-compose.yml`

```
version: '3'
services:
  databases:
    image: mysql
    ports:
      - "3307:3306"
    env_file:
      - evs.env
  web:
    image: nginx
    ports:
      - "80:80"
    depends_on:
      - databases
```

```

● anuvagarg@Anuvas-MacBook-Air nginx % pwd
/Users/anuvagarg/Docker/nginx
● anuvagarg@Anuvas-MacBook-Air nginx % ls
docker-compose.yml      evs.env
● anuvagarg@Anuvas-MacBook-Air nginx % cat docker-compose.yml
version: '3'
services:
  databases:
    image: mysql
    ports:
      - "3307:3306"
    env_file:
      - evs.env
  web:
    image: nginx
    ports:
      - "80:80"
    depends_on:
      - databases
● anuvagarg@Anuvas-MacBook-Air nginx % cat evs.env
MYSQL_ROOT_PASSWORD=redhat08
MYSQL_DATABASE=nginxdb
MYSQL_USER=root%

```

## Running images using docker-compose

- ❖ Save docker-compose.yaml file and do docker-compose up. `docker-compose up -d`

```

● anuvagarg@Anuvas-MacBook-Air nginx % docker-compose up -d
[+] Running 0/8
  :: databases Pulling
  :: 051f419db9dd Waiting
  :: 7627573fa82a Waiting
  :: a44b358d7796 Waiting
  :: 95753aff4b95 Waiting
  :: a1fa3bee53f4 Waiting
  :: f5227e0d612c Waiting
  :: b4b4368b1983 Waiting
[+] Running 0/192 Waiting
  :: databases Pulling
  :: 051f419db9dd Waiting
  :: 7627573fa82a Waiting
  :: a44b358d7796 Waiting
  :: 95753aff4b95 Waiting
  :: a1fa3bee53f4 Waiting
  :: f5227e0d612c Waiting
  :: b4b4368b1983 Waiting

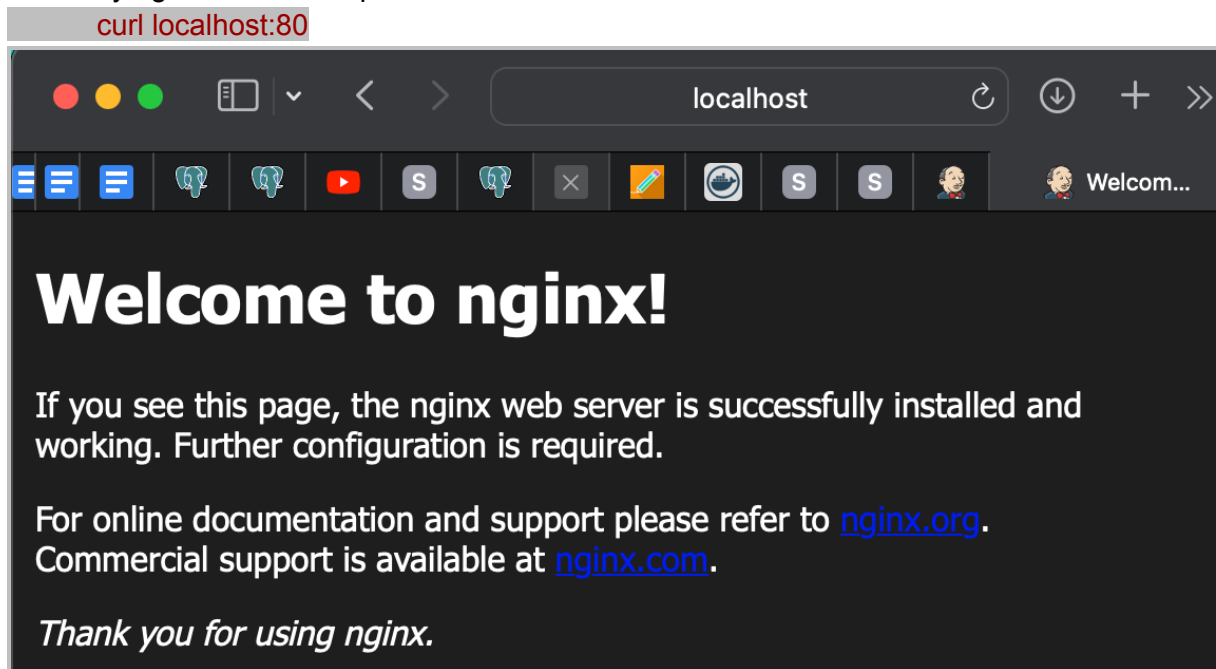
```

```
[+] Running 3/3
  :: Network nginx_default      Created          0.2s
  :: Container nginx-databases-1 Started          1.7s
  :: Container nginx-web-1      Started          2.9s
● anuvagarg@Anuvas-MacBook-Air 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>
```

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



Stop and remove your docker container using docker-compose.

**docker-compose down**

```
● anuvagarg@Anuvas-MacBook-Air nginx % docker-compose down
[+] Running 3/3
  :: Container nginx-web-1      Removed          0.4s
  :: Container nginx-databases-1 Removed          0.2s
  :: Network nginx_default      Removed          0.1s
```