Experiment: 5 (Docker-Compose)

Aim: Multi-container setup using Docker Compose.

Step: 1 Installing Docker-Compose

```
[root@devashish ~]# sudo curl -L "https://github.com/docker/compose/releases/download/1.28.4/dock
er-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   633  100   633    0     0    613      0  0:00:01  0:00:01 --:--:--   613
100 11.6M  100 11.6M    0     0   816k      0  0:00:14  0:00:14 --:--:-- 1202k
[root@devashish ~]#
```

Step: 2 Apply executable permissions to the binary and checking whether docker-compose has been installed or not.

```
[root@devashish ~]# sudo chmod +x /usr/local/bin/docker-compose
[root@devashish ~]# docker-compose --version
docker-compose version 1.28.4, build cabd5cfb
```

Step: 3 Create my.env file with below code

```
[root@devashish ~]# vi my.env
[root@devashish ~]#
```

root@devashish:~

```
MYSQL_ROOT_PASSWORD=1234
MYSQL_DATABASE=mysqldb
MYSQL_USER=root
~
~
~
~
~
~
~
```

Step: 4 Create docker-compose.yml file with below code

```
[root@devashish ~]# vi docker-compose.yml
[root@devashish ~]#
```

root@devashish:~

```
version: '3'
services:
  databases:
      image: mysql
      ports:
        - "3010:3306"
      env_file:
        - my.env
  web:
      image: nginx
      ports:
        - "82:80"
      depends_on:
        - databases
~
~
~
~
```

Step: 5 Run docker-compose up –d command

```
[root@devashish ~]# docker-compose up -d
Building with native build. Learn about native build in Compose here: https://docs.docker.com/go/compose-native-build/
Creating network "root_default" with the default driver
Pulling databases (mysql:)...
latest: Pulling from library/mysql
45b42c59be33: Pull complete
b4f790bd91da: Pull complete
325ae51788e9: Pull complete
adcb9439d751: Pull complete
174c7fe16c78: Pull complete
698058ef136c: Pull complete
4690143a669e: Pull complete
f7599a246fd6: Pull complete
35a55bf0c196: Pull complete
790ac54f4c47: Pull complete
b0ddd5d1b543: Pull complete
1aefd67cb33d: Pull complete
Digest: sha256:7706e4c382be813b58ef514f2bdac747cd463a6866c6c81165d42a1d0e4fe947
Status: Downloaded newer image for mysql:latest
Pulling web (nginx:)...
latest: Pulling from library/nginx
45b42c59be33: Already exists
8acc495f1d91: Pull complete
ec3bd7de90d7: Pull complete
19e2441aeeab: Pull complete
f5a38c5f8d4e: Pull complete
83500d851118: Pull complete
Digest: sha256:f3693fe50d5b1df1ecd315d54813a77afd56b0245a404055a946574deb6b34fc
Status: Downloaded newer image for nginx:latest
Creating root_databases_1 ... done
Creating root_web_1       ... done
[root@devashish ~]#
```

Step :6 We can see two containers have been launched.

Step: 7 Now, we are all set to use nginx server at 81 port of our machine.

```
[root@devashish ~]# curl localhost:82
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    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>
[root@devashish ~]#
```

Step: 8 Run docker-compose down command to delete running containers through docker-compose.yml file.

```
[root@devashish ~]# docker-compose down
Stopping root_web_1       ... done
Stopping root_databases_1 ... done
Removing root_web_1       ... done
Removing root_databases_1 ... done
Removing network root_default
[root@devashish ~]#
```