NAME: KUSHAGRA BANSAL

ROLL_NO: 116

SAP_ID: 500067414

Experiment -5

(Docker-Compose)

Aim: Multi-container setup using Docker Compose.

Step: 1 Installing Docker-Compose

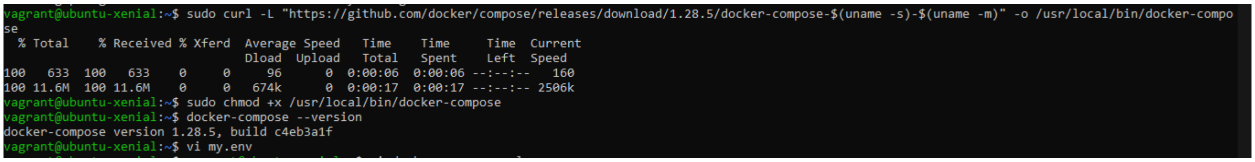
Command:

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.28.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

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

Command:

A. sudo chmod +x /usr/local/bin/docker-compose
B. docker-compose --version

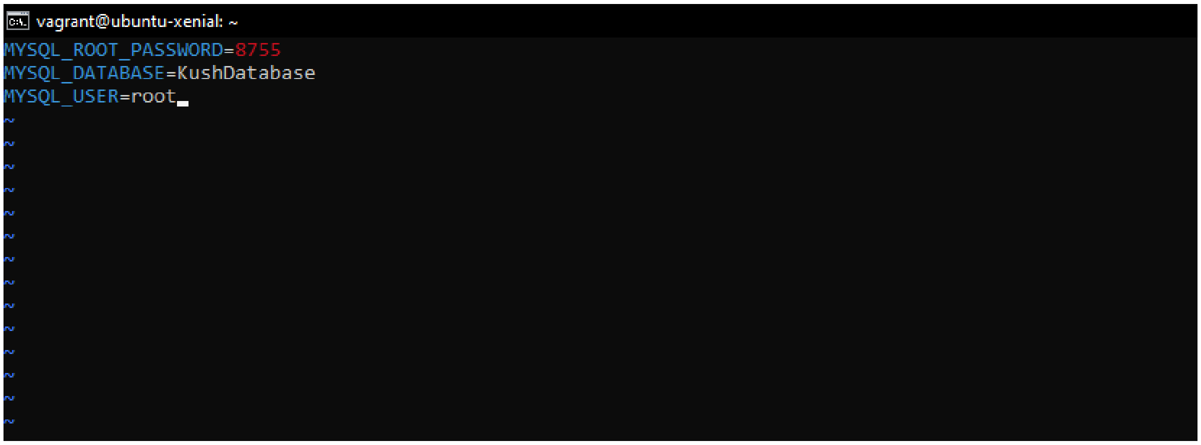


```
vagrant@ubuntu-xenial:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.28.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compo
se
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   633  100   633    0     0     96      0  0:00:06  0:00:06 --:--:--   160
100 11.6M  100 11.6M    0     0   674k      0  0:00:17  0:00:17 --:--:-- 2506k
vagrant@ubuntu-xenial:~$ sudo chmod +x /usr/local/bin/docker-compose
vagrant@ubuntu-xenial:~$ docker-compose --version
docker-compose version 1.28.5, build c4eb3a1f
vagrant@ubuntu-xenial:~$ vi my.env
```

Step: 2 Create my.env file with below code

Command:

- ⇨ **vi my.env**
  the editor file will open press 'i' to activate insert mode



```
MYSQL_ROOT_PASSWORD=8755
MYSQL_DATABASE=KushDatabase
MYSQL_USER=root
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

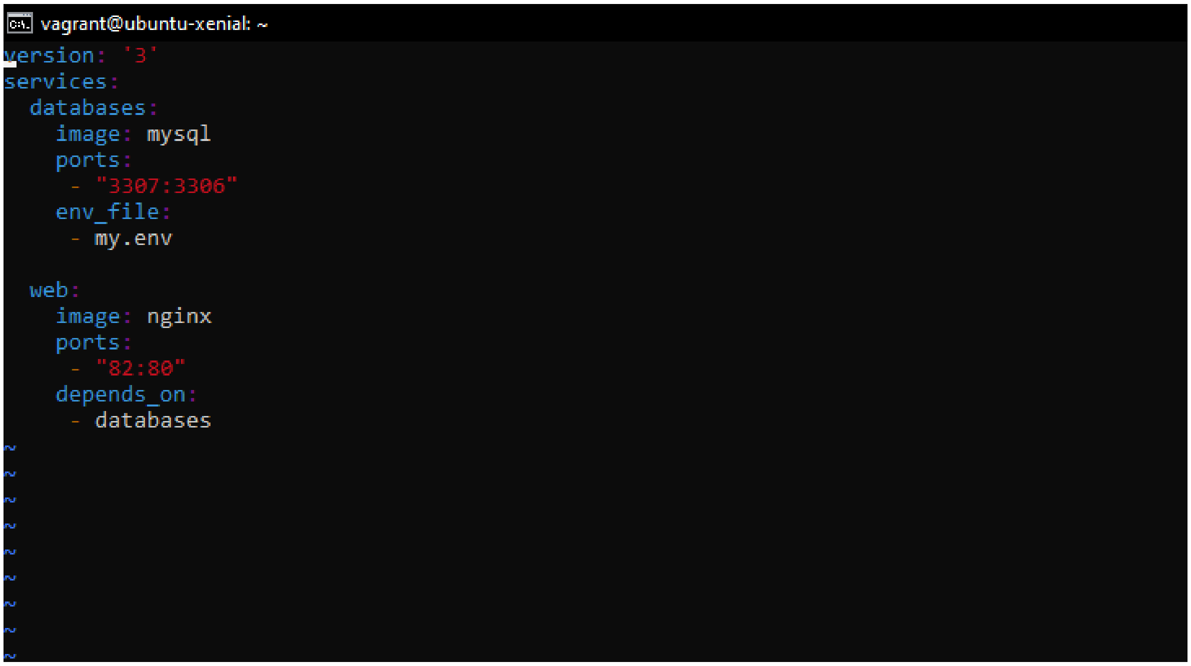
Step3: Create docker-compose.yml file with below code

Command:

Vi docker-compose.yml

Remember about the segmentation otherwise it will show an error

The editor file will open press 'i' to activate insert mode



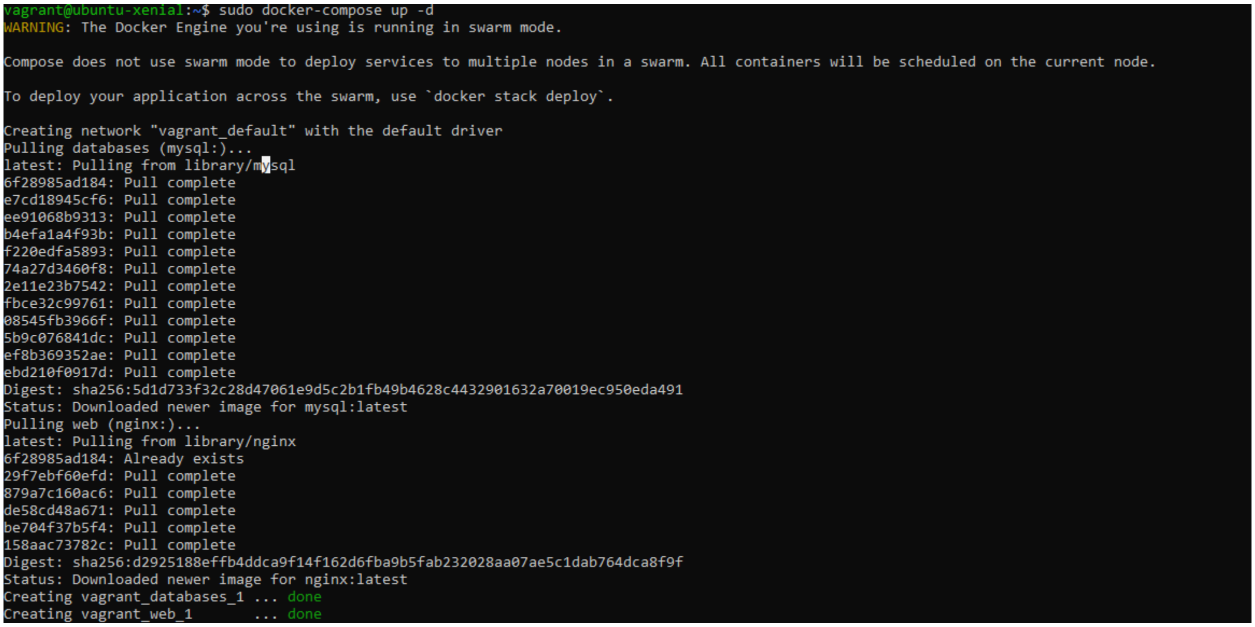
vagrant@ubuntu-xenial: ~

```
version: '3'
services:
  databases:
    image: mysql
    ports:
     - "3307:3306"
    env_file:
     - my.env

  web:
    image: nginx
    ports:
     - "82:80"
    depends_on:
     - databases
~
~
~
~
~
~
~
~
~
```

Step 4: Run docker-compose File

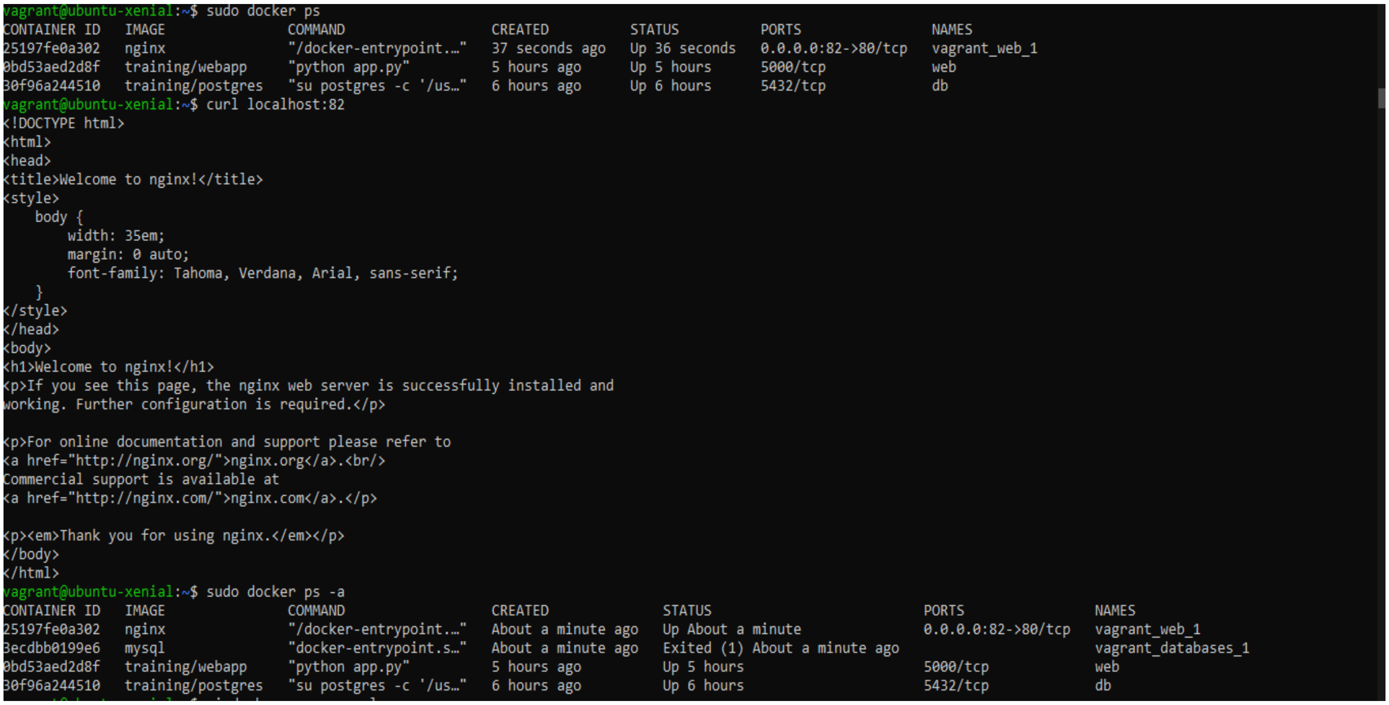
Command:

sudo docker-compose up –d



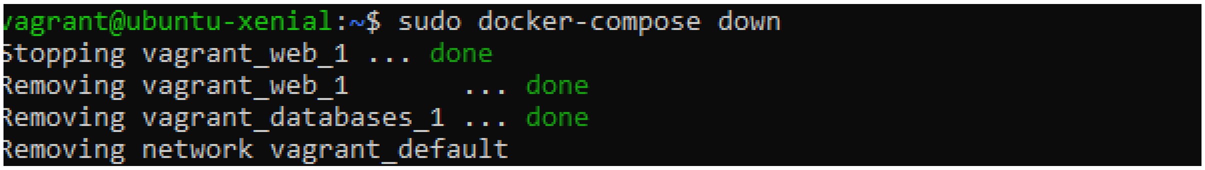
```
vagrant@ubuntu-xenial:~$ sudo docker-compose up -d
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm. All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

Creating network "vagrant_default" with the default driver
Pulling databases (mysql:)...
latest: Pulling from library/mysql
6f28985ad184: Pull complete
e7cd18945cf6: Pull complete
ee91068b9313: Pull complete
b4efa1a4f93b: Pull complete
f220edfa5893: Pull complete
74a27d3460f8: Pull complete
2e11e23b7542: Pull complete
fbce32c99761: Pull complete
08545fb3966f: Pull complete
5b9c076841dc: Pull complete
ef8b369352ae: Pull complete
ebd210f0917d: Pull complete
Digest: sha256:5d1d733f32c28d47061e9d5c2b1fb49b4628c4432901632a70019ec950eda491
Status: Downloaded newer image for mysql:latest
Pulling web (nginx:)...
latest: Pulling from library/nginx
6f28985ad184: Already exists
29f7ebf60efd: Pull complete
879a7c160ac6: Pull complete
de58cd48a671: Pull complete
be704f37b5f4: Pull complete
158aac73782c: Pull complete
Digest: sha256:d2925188effb4ddca9f14f162d6fba9b5fab232028aa07ae5c1dab764dca8f9f
Status: Downloaded newer image for nginx:latest
Creating vagrant_databases_1 ... done
Creating vagrant_web_1       ... done
```

Step 5: We can see two containers have been launched and we can run nginx server at port 82

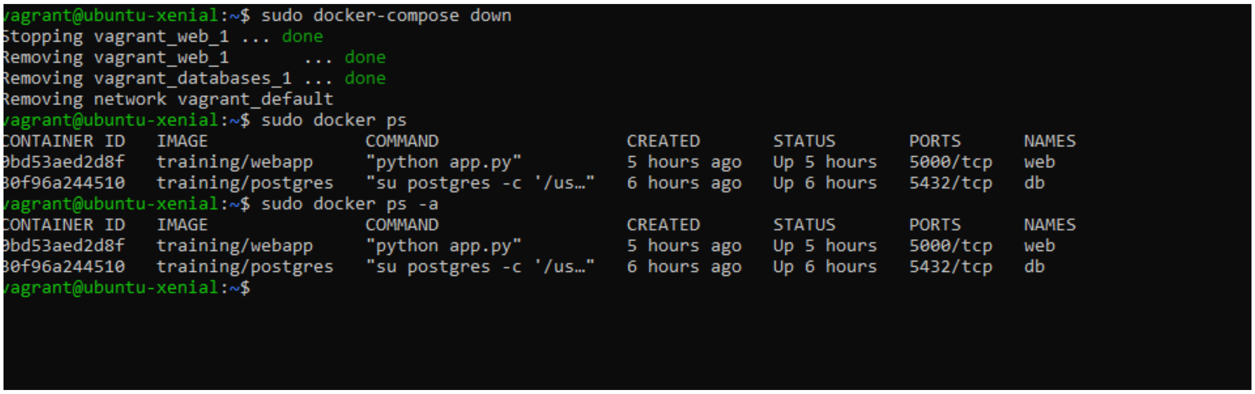


```
vagrant@ubuntu-xenial:~$ sudo docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED          STATUS          PORTS                NAMES
25197fe0a302   nginx               "/docker-entrypoint.…"   37 seconds ago   Up 36 seconds   0.0.0.0:82->80/tcp   vagrant_web_1
0bd53aed2d8f   training/webapp     "python app.py"          5 hours ago      Up 5 hours      5000/tcp             web
30f96a244510   training/postgres   "su postgres -c '/us…"   6 hours ago      Up 6 hours      5432/tcp             db
vagrant@ubuntu-xenial:~$ 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>
vagrant@ubuntu-xenial:~$ sudo docker ps -a
CONTAINER ID   IMAGE               COMMAND                  CREATED              STATUS                          PORTS                NAMES
25197fe0a302   nginx               "/docker-entrypoint.…"   About a minute ago   Up About a minute               0.0.0.0:82->80/tcp   vagrant_web_1
3ecdbb0199e6   mysql               "docker-entrypoint.s…"   About a minute ago   Exited (1) About a minute ago                        vagrant_databases_1
0bd53aed2d8f   training/webapp     "python app.py"          5 hours ago          Up 5 hours                      5000/tcp             web
30f96a244510   training/postgres   "su postgres -c '/us…"   6 hours ago          Up 6 hours                      5432/tcp             db
```

Step 6: Run docker-compose down command to delete running containers through docker-compose.yml file.



```
vagrant@ubuntu-xenial:~$ sudo docker-compose down
Stopping vagrant_web_1 ... done
Removing vagrant_web_1       ... done
Removing vagrant_databases_1 ... done
Removing network vagrant_default
```



```
vagrant@ubuntu-xenial:~$ sudo docker-compose down
Stopping vagrant_web_1 ... done
Removing vagrant_web_1       ... done
Removing vagrant_databases_1 ... done
Removing network vagrant_default
vagrant@ubuntu-xenial:~$ sudo docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED       STATUS       PORTS      NAMES
0bd53aed2d8f   training/webapp     "python app.py"          5 hours ago   Up 5 hours   5000/tcp   web
30f96a244510   training/postgres   "su postgres -c '/us…"   6 hours ago   Up 6 hours   5432/tcp   db
vagrant@ubuntu-xenial:~$ sudo docker ps -a
CONTAINER ID   IMAGE               COMMAND                  CREATED       STATUS       PORTS      NAMES
0bd53aed2d8f   training/webapp     "python app.py"          5 hours ago   Up 5 hours   5000/tcp   web
30f96a244510   training/postgres   "su postgres -c '/us…"   6 hours ago   Up 6 hours   5432/tcp   db
vagrant@ubuntu-xenial:~$
```