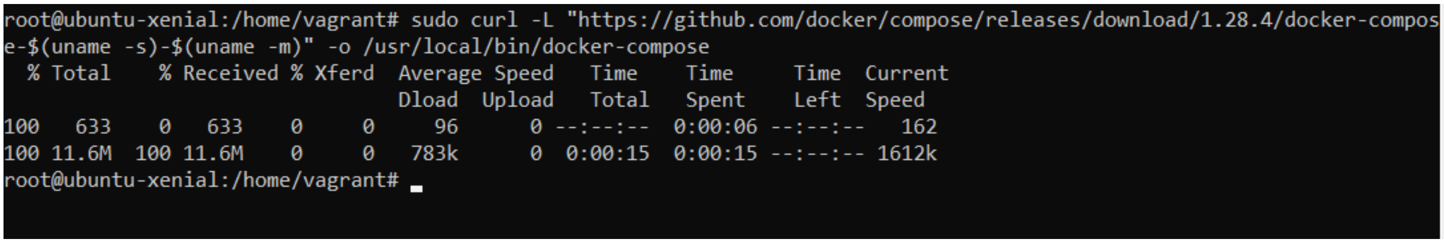
Experiment 5

AIM: docker compose

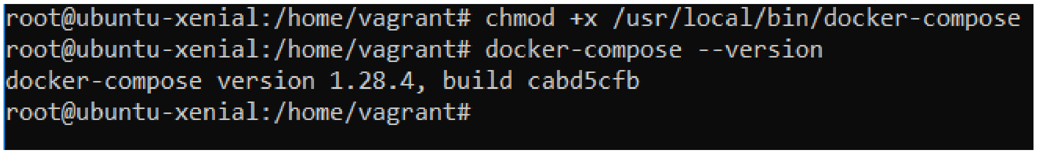
Download docker-compose

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.28.4/docker-compose-

$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```



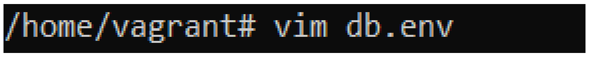
```
root@ubuntu-xenial:/home/vagrant# sudo curl -L "https://github.com/docker/compose/releases/download/1.28.4/docker-compos
e-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   633    0   633    0     0     96      0 --:--:--  0:00:06 --:--:--   162
100 11.6M  100 11.6M    0     0   783k      0  0:00:15  0:00:15 --:--:-- 1612k
root@ubuntu-xenial:/home/vagrant# _
```

Give executable permission to docker-compose file downloaded

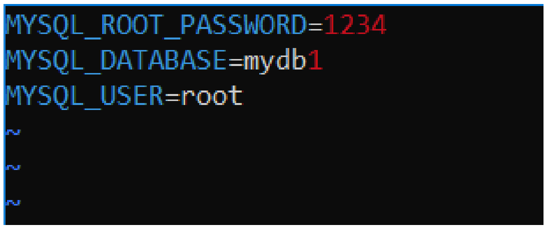


```
root@ubuntu-xenial:/home/vagrant# chmod +x /usr/local/bin/docker-compose
root@ubuntu-xenial:/home/vagrant# docker-compose --version
docker-compose version 1.28.4, build cabd5cfb
root@ubuntu-xenial:/home/vagrant#
```

Create env file to store env variables to be used in docker-compose.yml



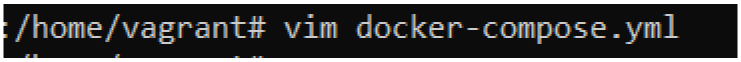
```
/home/vagrant# vim db.env
```



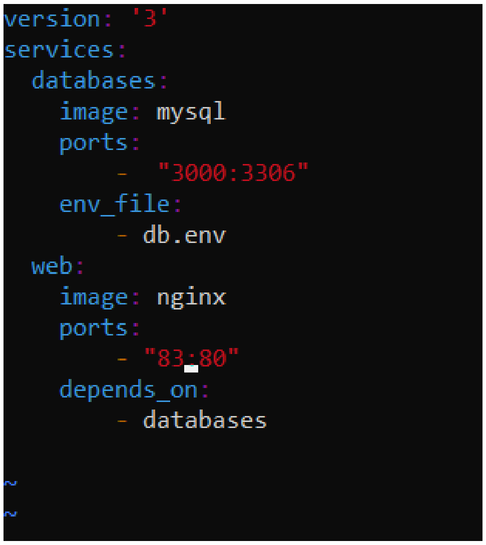
```
MYSQL_ROOT_PASSWORD=1234
MYSQL_DATABASE=mydb1
MYSQL_USER=root
~
~
~
```

Docker-compose file:

```
version: '3'
services:
 databases:
  image: mysql
  ports:
      -  "3000:3306"
  env_file:
      - db.env
 web:
  image: nginx
  ports:
      - "83:80"
  depends_on:
      - databases
```

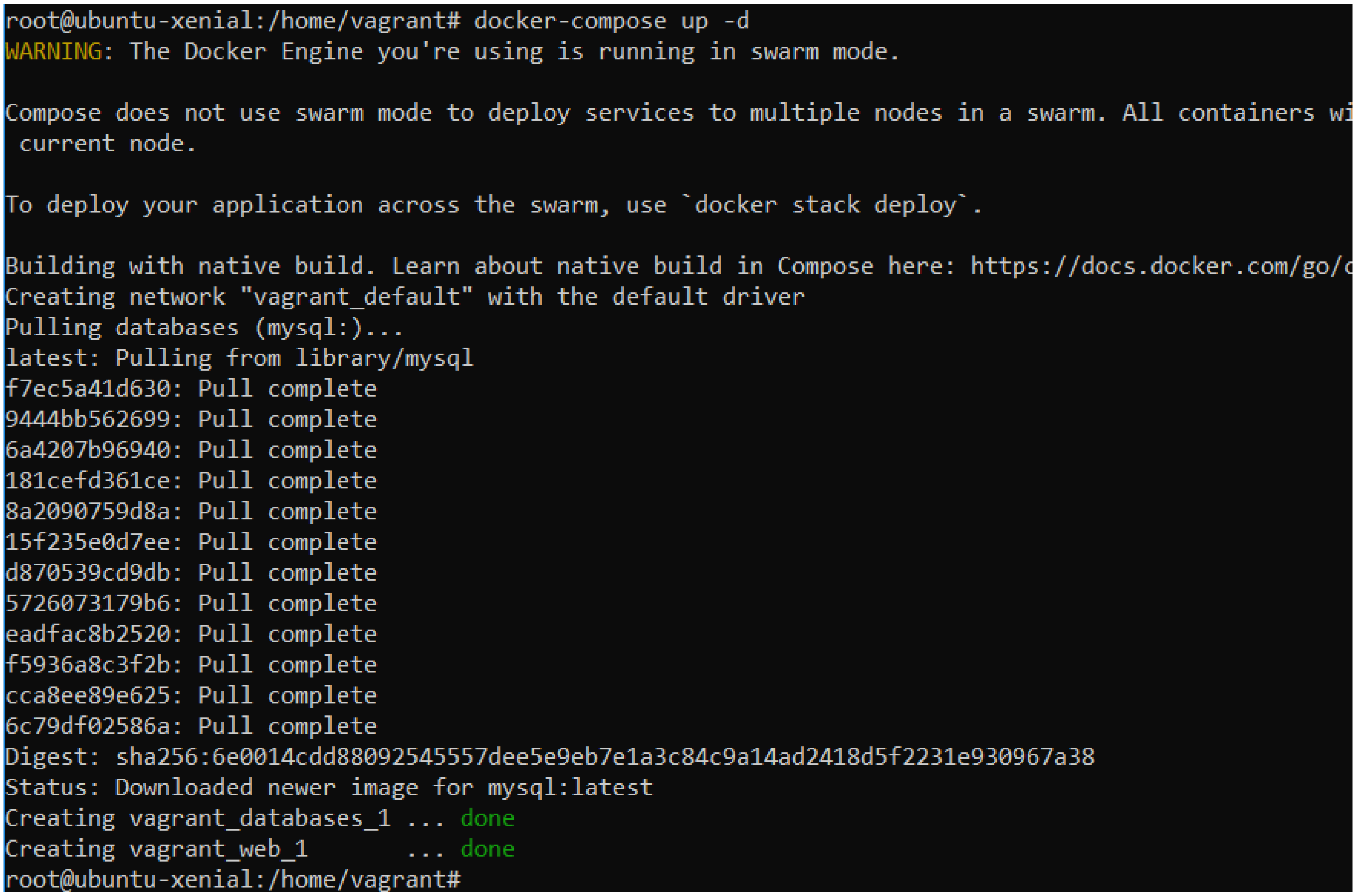


```
:/home/vagrant# vim docker-compose.yml
```



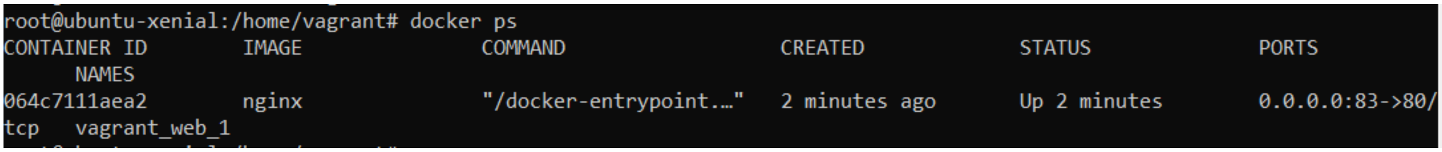
```
version: '3'
services:
  databases:
    image: mysql
    ports:
        -  "3000:3306"
    env_file:
        - db.env
  web:
    image: nginx
    ports:
        - "83:80"
    depends_on:
        - databases

~
~
```

Run “docker-compose up -d” command



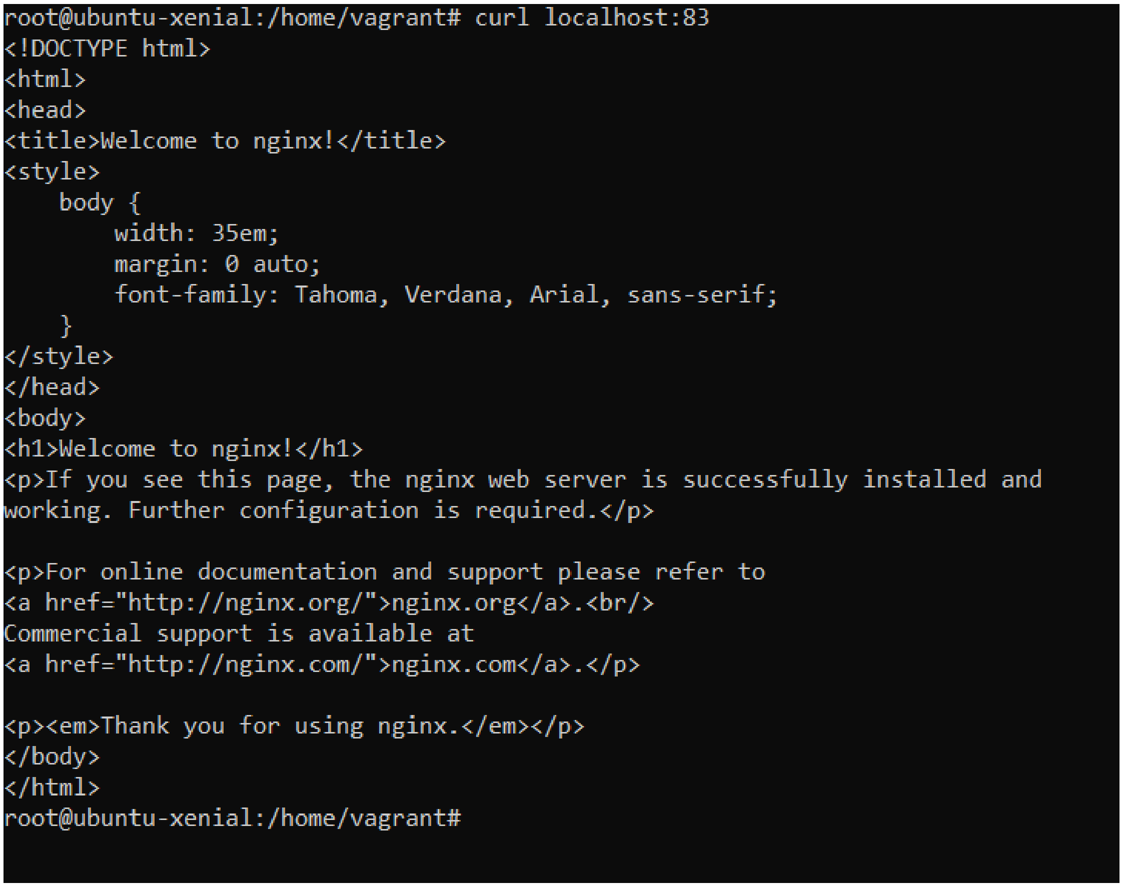
```
root@ubuntu-xenial:/home/vagrant# docker-compose up -d
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm. All containers wi
 current node.

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

Building with native build. Learn about native build in Compose here: https://docs.docker.com/go/c
Creating network "vagrant_default" with the default driver
Pulling databases (mysql:)...
latest: Pulling from library/mysql
f7ec5a41d630: Pull complete
9444bb562699: Pull complete
6a4207b96940: Pull complete
181cefd361ce: Pull complete
8a2090759d8a: Pull complete
15f235e0d7ee: Pull complete
d870539cd9db: Pull complete
5726073179b6: Pull complete
eadfac8b2520: Pull complete
f5936a8c3f2b: Pull complete
cca8ee89e625: Pull complete
6c79df02586a: Pull complete
Digest: sha256:6e0014cdd88092545557dee5e9eb7e1a3c84c9a14ad2418d5f2231e930967a38
Status: Downloaded newer image for mysql:latest
Creating vagrant_databases_1 ... done
Creating vagrant_web_1       ... done
root@ubuntu-xenial:/home/vagrant#
```

List running containers



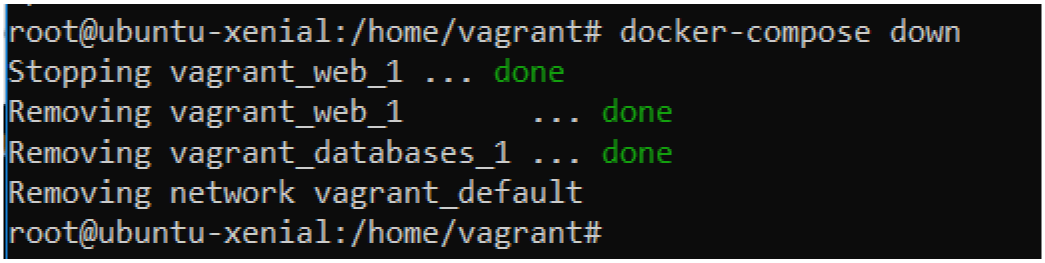
```
root@ubuntu-xenial:/home/vagrant# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
     NAMES
064c7111aea2        nginx               "/docker-entrypoint.…"   2 minutes ago       Up 2 minutes        0.0.0.0:83->80/
tcp   vagrant_web_1
```

Curl / In browser open ip:83 (port set in docker-compose.yml file)



```
root@ubuntu-xenial:/home/vagrant# curl localhost:83
<!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@ubuntu-xenial:/home/vagrant#
```

Stop and remove:



```
root@ubuntu-xenial:/home/vagrant# docker-compose down
Stopping vagrant_web_1 ... done
Removing vagrant_web_1       ... done
Removing vagrant_databases_1 ... done
Removing network vagrant_default
root@ubuntu-xenial:/home/vagrant#
```