Lesson 07 Lesson-End Project

Containerizing Legacy Docker Application

Project Agenda: To containerize the legacy application system using Docker for improved portability, efficient resource utilization, and simplified deployment processes

Description: Your team has been asked to sync the Django application and the database so that the information can be stored and accessed again on demand. This is a legacy application with two components: a Django application and a Postgres database. Now, your project manager asks you to containerize the legacy system using Docker. Therefore, you are required to create Docker images for these components using the Dockerfile and connect them using the docker-compose file.

Tools required: Docker

Prerequisites: You must have installed all the required tools and have a working Docker setup to proceed. In case you do not have them, please refer to Lesson 07 Demo 1 to install and set up Docker.

Expected Deliverables: A dockerized application

Steps to be followed:

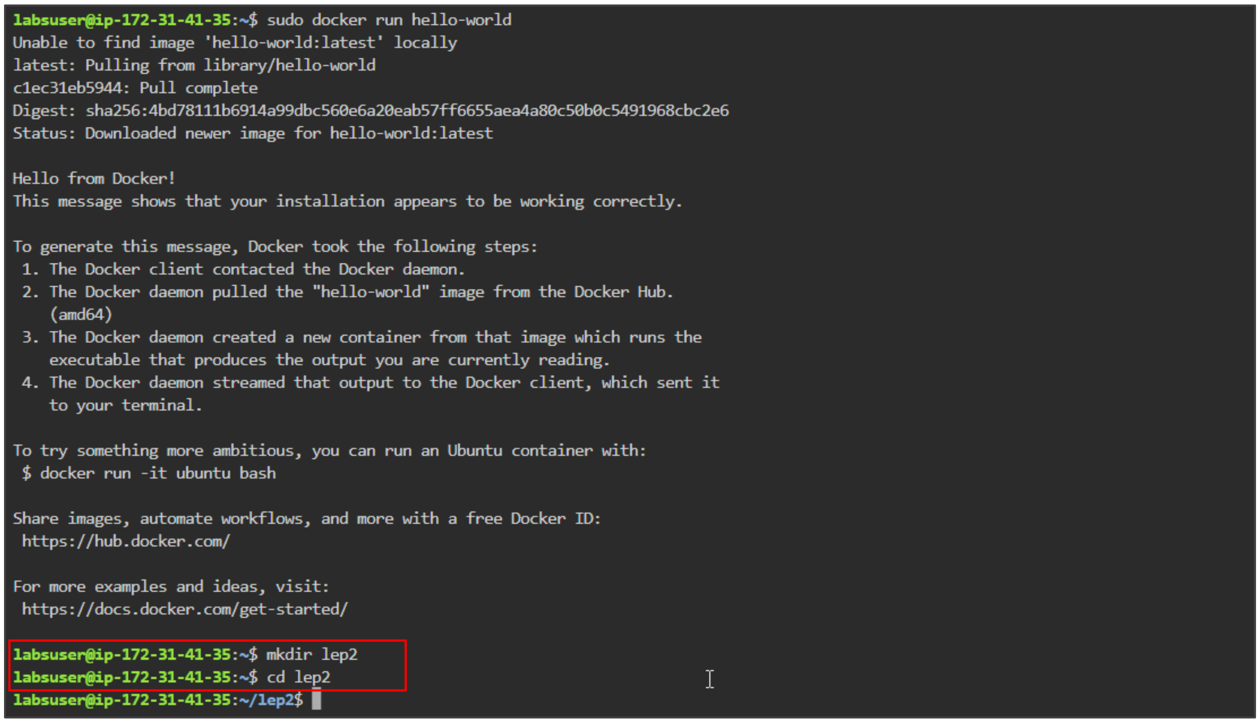
1. Create a new directory
2. Create a new Docker file
3. Install Python requirements defined in the **req.txt**
4. Create a **YAML** file in the project directory
5. Create the Django project
6. List all the contents of the Django project
7. Change the ownership of new files
8. Set up a database connection

Step 1: Create a new directory

1.1 Create a new directory and navigate to it using the following commands:

mkdir lep2

cd lep2

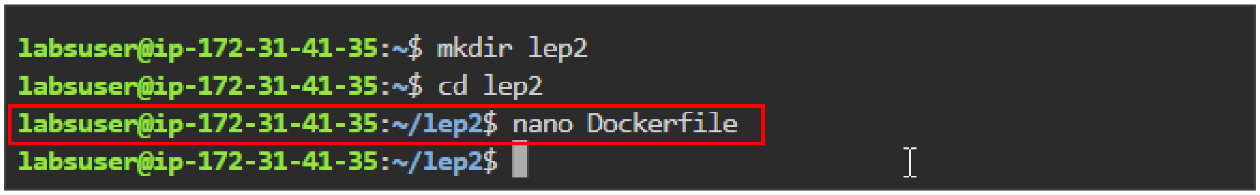


```
labsuser@ip-172-31-41-35:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:4bd78111b6914a99dbc560e6a20eab57ff6655aea4a80c50b0c5491968cbc2e6
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

labsuser@ip-172-31-41-35:~$ mkdir lep2
labsuser@ip-172-31-41-35:~$ cd lep2
labsuser@ip-172-31-41-35:~/lep2$
```

Step 2: Create a new Docker file

2.1 Create a new Docker file using the following command:

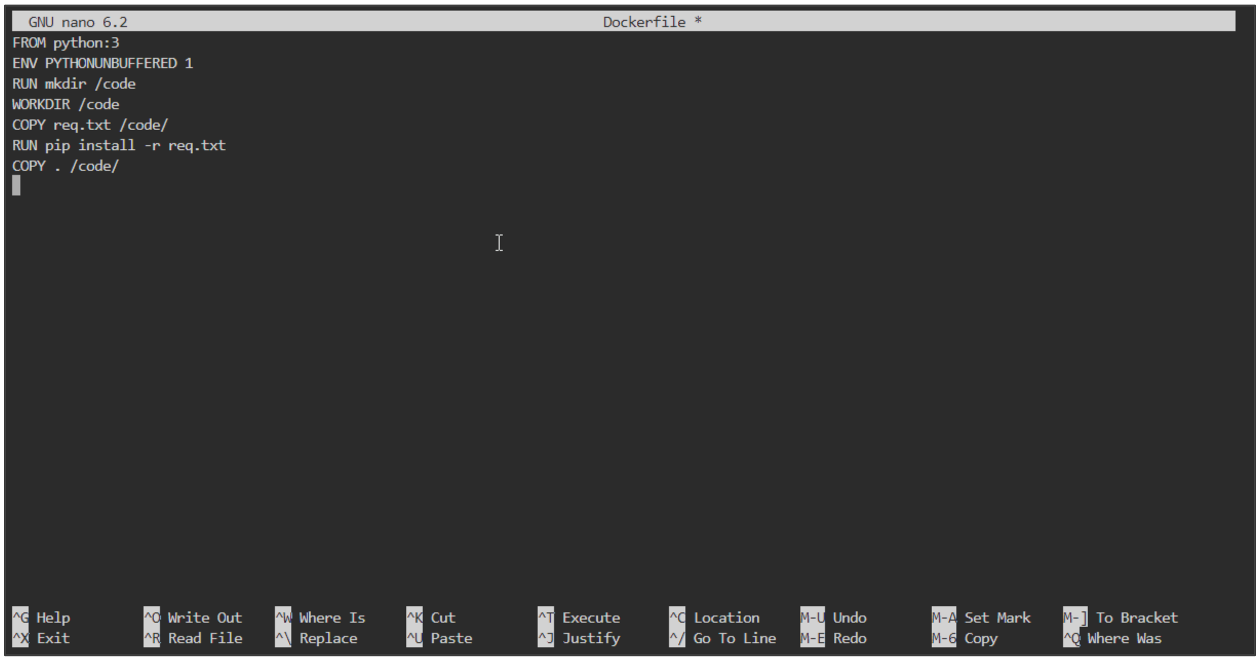
nano Dockerfile



```
labsuser@ip-172-31-41-35:~$ mkdir lep2
labsuser@ip-172-31-41-35:~$ cd lep2
labsuser@ip-172-31-41-35:~/lep2$ nano Dockerfile
labsuser@ip-172-31-41-35:~/lep2$
```

2.2 Add the following code to the Docker file:

FROM python:3
ENV PYTHONUNBUFFERED 1
RUN mkdir /code
WORKDIR /code
COPY req.txt /code/
RUN pip install -r req.txt
COPY . /code/

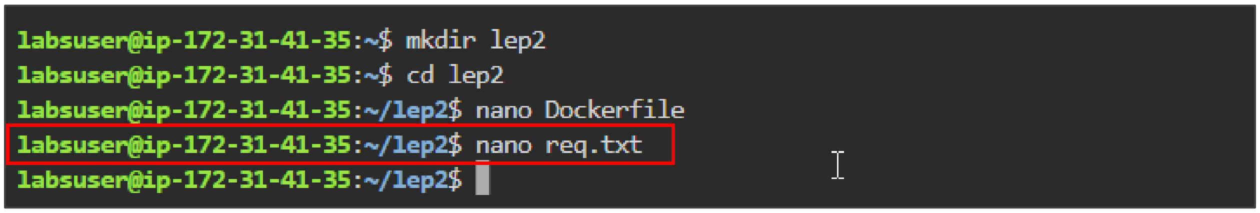


```
  GNU nano 6.2                          Dockerfile *
FROM python:3
ENV PYTHONUNBUFFERED 1
RUN mkdir /code
WORKDIR /code
COPY req.txt /code/
RUN pip install -r req.txt
COPY . /code/

^G Help      ^O Write Out  ^W Where Is  ^K Cut    ^T Execute  ^C Location   M-U Undo  M-A Set Mark  M-] To Bracket
^X Exit      ^R Read File  ^\ Replace   ^U Paste  ^J Justify  ^/ Go To Line M-E Redo  M-6 Copy      ^Q Where Was
```

Note: Press the **ctrl + o** keys to write, and then press the enter key; press the **ctrl + x** keys to exit the editor

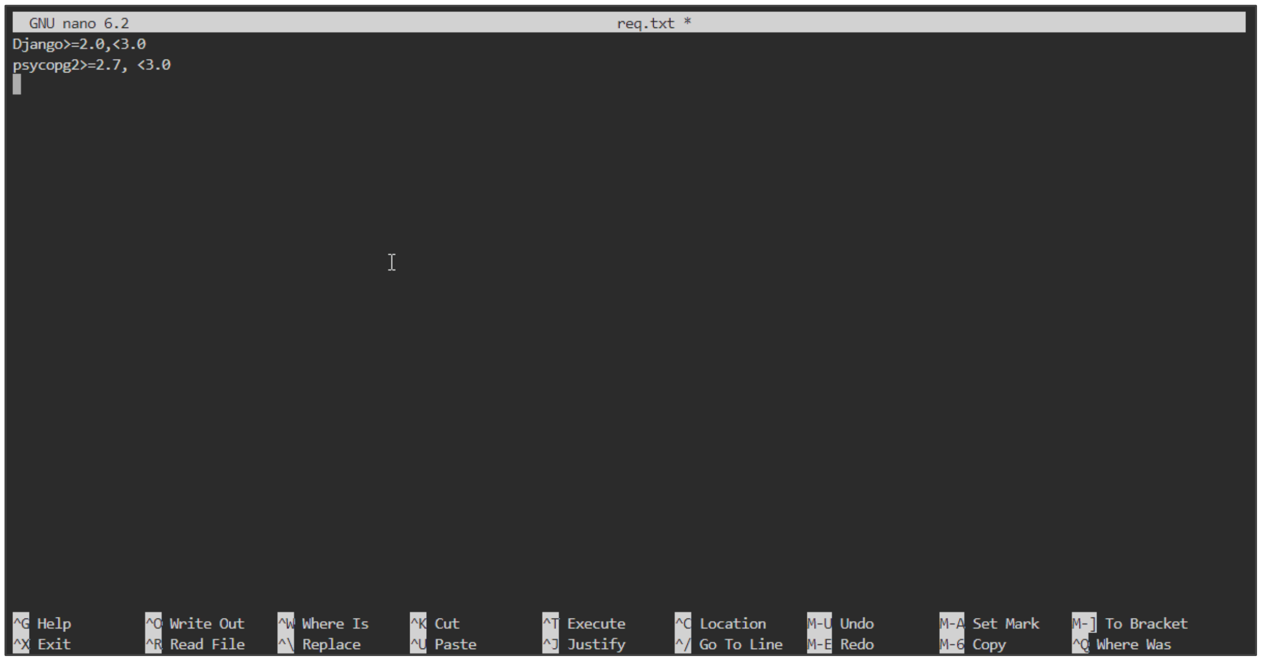
Step 3: Install Python requirements defined in the req.txt

3.1 Create the **req.txt** file project directory named **lep2** using the following command:
nano req.txt



```
labsuser@ip-172-31-41-35:~$ mkdir lep2
labsuser@ip-172-31-41-35:~$ cd lep2
labsuser@ip-172-31-41-35:~/lep2$ nano Dockerfile
labsuser@ip-172-31-41-35:~/lep2$ nano req.txt
labsuser@ip-172-31-41-35:~/lep2$
```

3.2 Add the following specification in the **req.txt** file:
Django>=2.0,<3.0
psycopg2>=2.7, <3.0

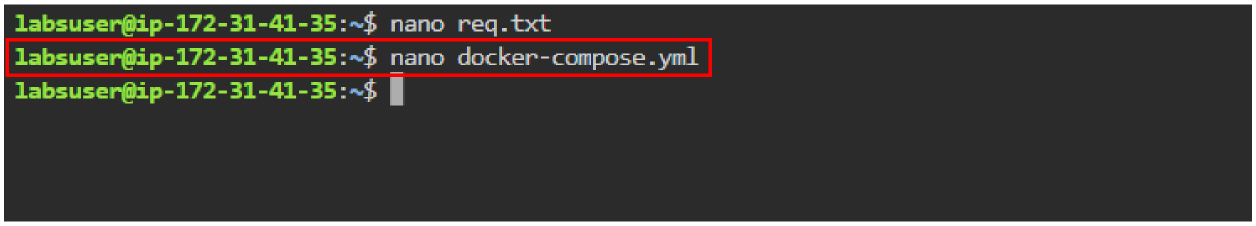


```
  GNU nano 6.2                                  req.txt *
Django>=2.0,<3.0
psycopg2>=2.7, <3.0

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was
```

Note: Press the **ctrl + o** keys to write, and then press the enter key; press the **ctrl + x** keys to exit the editor

Step 4: Create a YAML file in the project directory

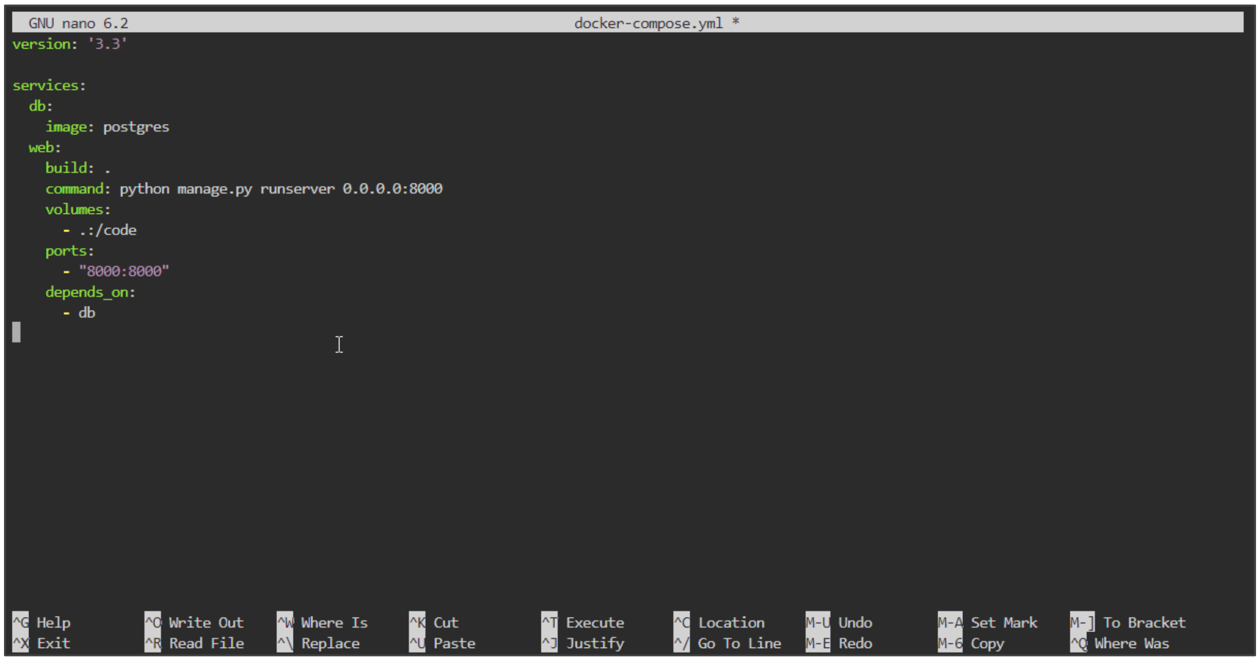
4.1 Create a **docker-compose.yml** file in the project directory named lep2 using the following command:
nano docker-compose.yml



```
labsuser@ip-172-31-41-35:~$ nano req.txt
labsuser@ip-172-31-41-35:~$ nano docker-compose.yml
labsuser@ip-172-31-41-35:~$
```

4.2 Add the following code to define the services:

```
version: '3.3'

services:
  db:
    image: postgres
  web:
    build: .
    command: python manage.py runserver 0.0.0.0:8000
    volumes:
      - .:/code
    ports:
      - "8000:8000"
    depends_on:
      - db
```



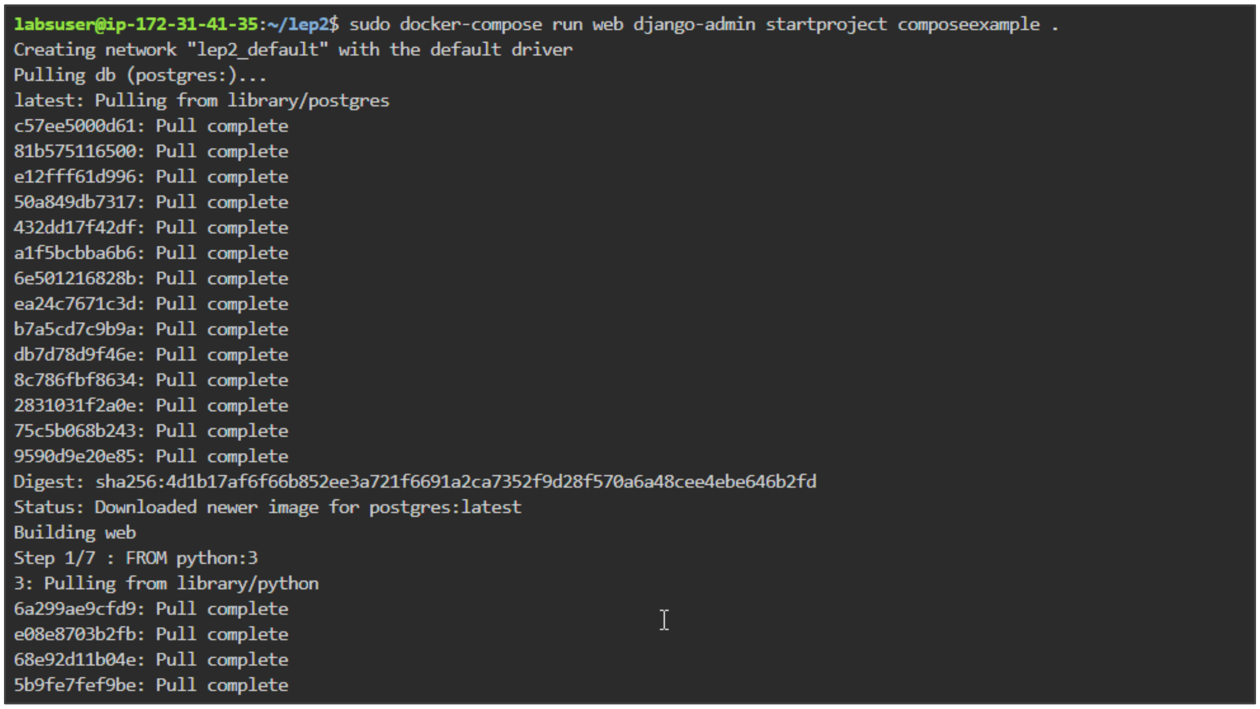
```
  GNU nano 6.2                          docker-compose.yml *
version: '3.3'

services:
  db:
    image: postgres
  web:
    build: .
    command: python manage.py runserver 0.0.0.0:8000
    volumes:
      - .:/code
    ports:
      - "8000:8000"
    depends_on:
      - db

^G Help      ^O Write Out  ^W Where Is  ^K Cut     ^T Execute  ^C Location   M-U Undo  M-A Set Mark  M-] To Bracket
^X Exit      ^R Read File  ^\ Replace   ^U Paste   ^J Justify  ^/ Go To Line M-E Redo  M-6 Copy      ^Q Where Was
```

Note: Press the **ctrl + o** keys to write, and then press the enter key; press the **ctrl + x** keys to exit the editor

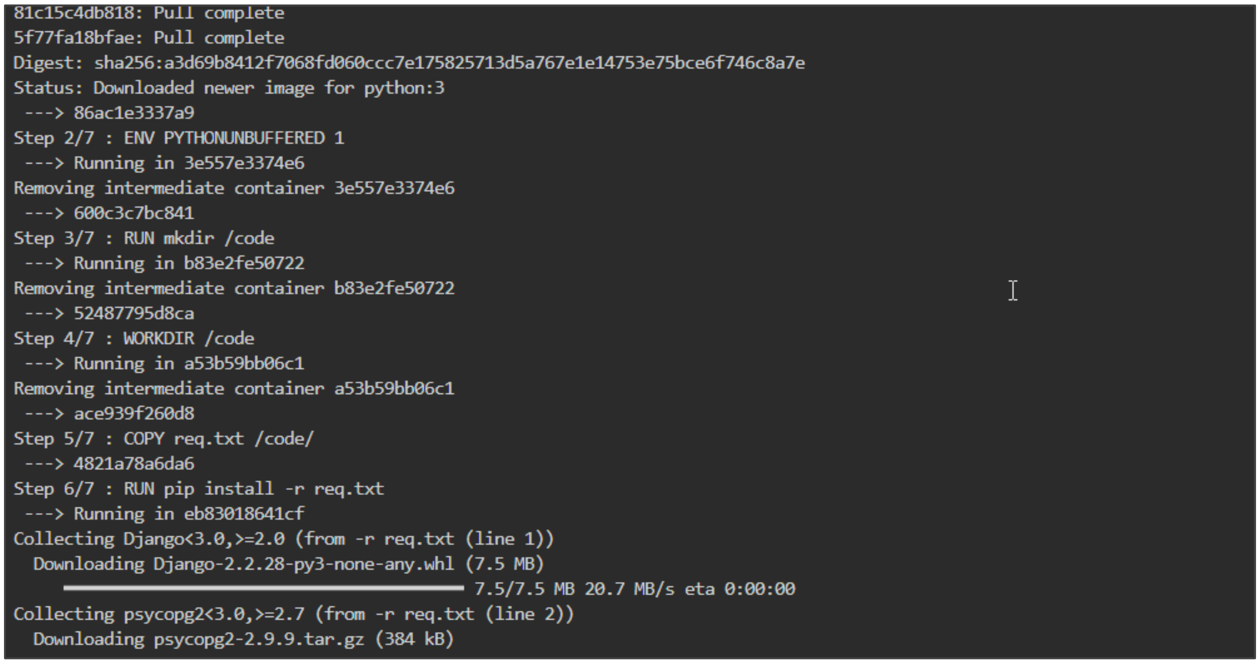
Step 5: Create a Django project

5.1 Use the following command to create a Django project:

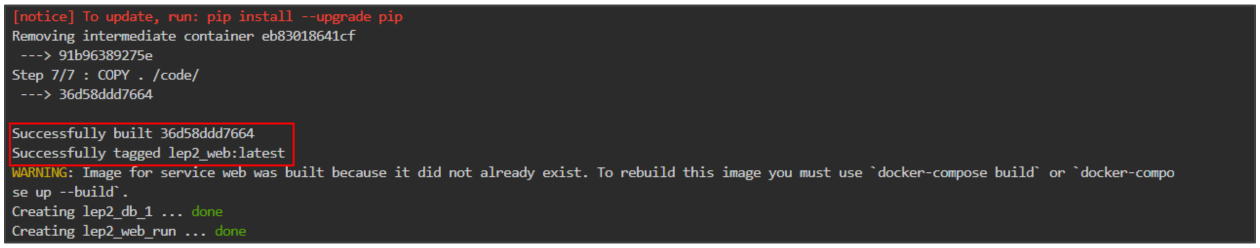
sudo docker-compose run web django-admin startproject composeexample



```
labsuser@ip-172-31-41-35:~/lep2$ sudo docker-compose run web django-admin startproject composeexample .
Creating network "lep2_default" with the default driver
Pulling db (postgres:)...
latest: Pulling from library/postgres
c57ee5000d61: Pull complete
81b575116500: Pull complete
e12fff61d996: Pull complete
50a849db7317: Pull complete
432dd17f42df: Pull complete
a1f5bcbba6b6: Pull complete
6e501216828b: Pull complete
ea24c7671c3d: Pull complete
b7a5cd7c9b9a: Pull complete
db7d78d9f46e: Pull complete
8c786fbf8634: Pull complete
2831031f2a0e: Pull complete
75c5b068b243: Pull complete
9590d9e20e85: Pull complete
Digest: sha256:4d1b17af6f66b852ee3a721f6691a2ca7352f9d28f570a6a48cee4ebe646b2fd
Status: Downloaded newer image for postgres:latest
Building web
Step 1/7 : FROM python:3
3: Pulling from library/python
6a299ae9cfd9: Pull complete
e08e8703b2fb: Pull complete
68e92d11b04e: Pull complete
5b9fe7fef9be: Pull complete
```



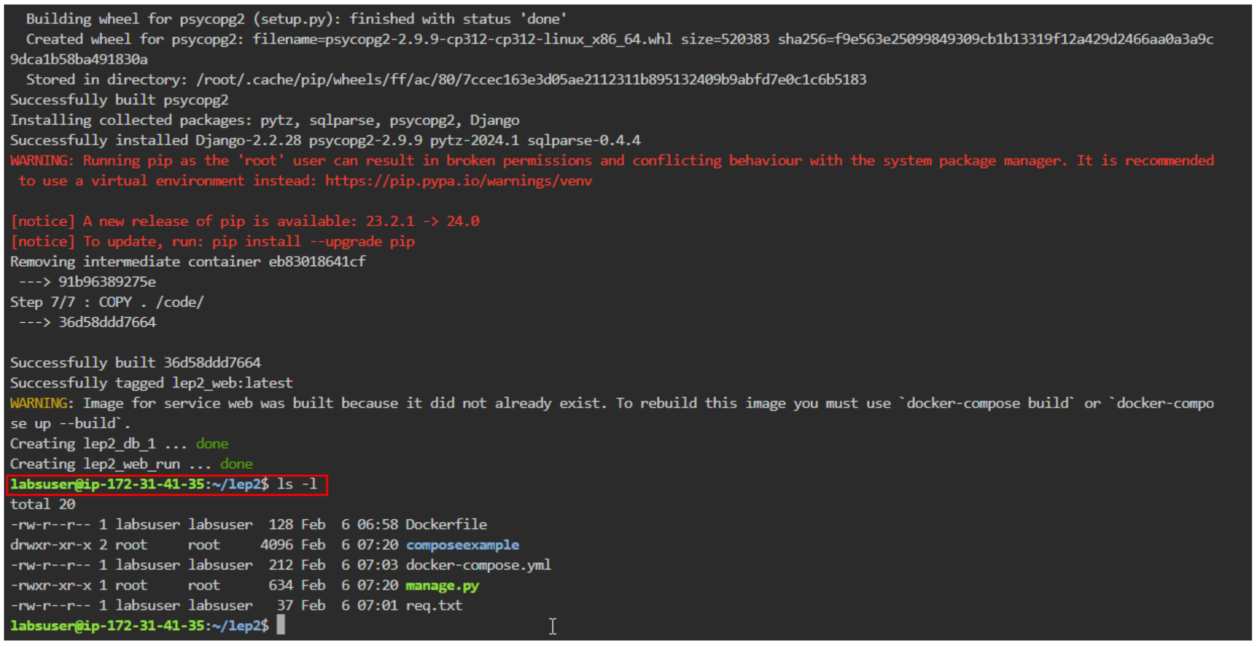
```
81c15c4db818: Pull complete
5f77fa18bfae: Pull complete
Digest: sha256:a3d69b8412f7068fd060ccc7e175825713d5a767e1e14753e75bce6f746c8a7e
Status: Downloaded newer image for python:3
 ---> 86ac1e3337a9
Step 2/7 : ENV PYTHONUNBUFFERED 1
 ---> Running in 3e557e3374e6
Removing intermediate container 3e557e3374e6
 ---> 600c3c7bc841
Step 3/7 : RUN mkdir /code
 ---> Running in b83e2fe50722
Removing intermediate container b83e2fe50722
 ---> 52487795d8ca
Step 4/7 : WORKDIR /code
 ---> Running in a53b59bb06c1
Removing intermediate container a53b59bb06c1
 ---> ace939f260d8
Step 5/7 : COPY req.txt /code/
 ---> 4821a78a6da6
Step 6/7 : RUN pip install -r req.txt
 ---> Running in eb83018641cf
Collecting Django<3.0,>=2.0 (from -r req.txt (line 1))
  Downloading Django-2.2.28-py3-none-any.whl (7.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.5/7.5 MB 20.7 MB/s eta 0:00:00
Collecting psycopg2<3.0,>=2.7 (from -r req.txt (line 2))
  Downloading psycopg2-2.9.9.tar.gz (384 kB)
```



```
[notice] To update, run: pip install --upgrade pip
Removing intermediate container eb83018641cf
 ---> 91b96389275e
Step 7/7 : COPY . /code/
 ---> 36d58ddd7664

Successfully built 36d58ddd7664
Successfully tagged lep2_web:latest
WARNING: Image for service web was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compo
se up --build`.
Creating lep2_db_1 ... done
Creating lep2_web_run ... done
```

Step 6: List all the contents of the Django project

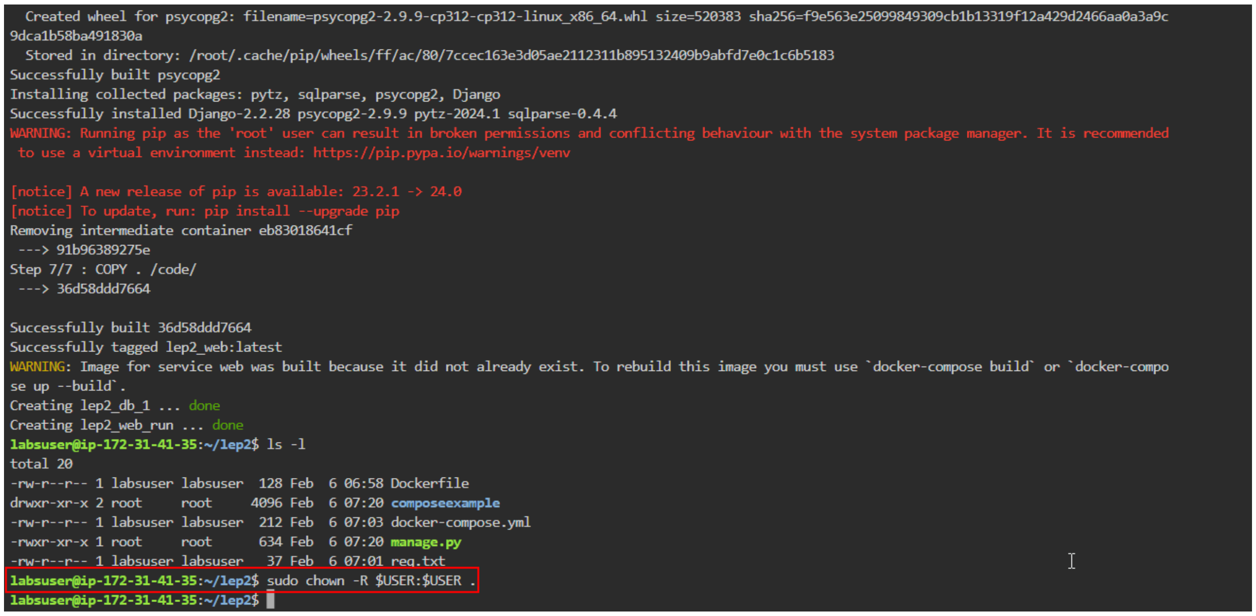
6.1 Use the following command to list all the contents of the Django project:
ls -l



```
  Building wheel for psycopg2 (setup.py): finished with status 'done'
  Created wheel for psycopg2: filename=psycopg2-2.9.9-cp312-cp312-linux_x86_64.whl size=520383 sha256=f9e563e25099849309cb1b13319f12a429d2466aa0a3a9c
9dca1b58ba491830a
  Stored in directory: /root/.cache/pip/wheels/ff/ac/80/7ccec163e3d05ae2112311b895132409b9abfd7e0c1c6b5183
Successfully built psycopg2
Installing collected packages: pytz, sqlparse, psycopg2, Django
Successfully installed Django-2.2.28 psycopg2-2.9.9 pytz-2024.1 sqlparse-0.4.4
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended
 to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.2.1 -> 24.0
[notice] To update, run: pip install --upgrade pip
Removing intermediate container eb83018641cf
 ---> 91b96389275e
Step 7/7 : COPY . /code/
 ---> 36d58ddd7664

Successfully built 36d58ddd7664
Successfully tagged lep2_web:latest
WARNING: Image for service web was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compo
se up --build`.
Creating lep2_db_1 ... done
Creating lep2_web_run ... done
labsuser@ip-172-31-41-35:~/lep2$ ls -l
total 20
-rw-r--r-- 1 labsuser labsuser  128 Feb  6 06:58 Dockerfile
drwxr-xr-x 2 root     root     4096 Feb  6 07:20 composeexample
-rw-r--r-- 1 labsuser labsuser  212 Feb  6 07:03 docker-compose.yml
-rwxr-xr-x 1 root     root      634 Feb  6 07:20 manage.py
-rw-r--r-- 1 labsuser labsuser   37 Feb  6 07:01 req.txt
labsuser@ip-172-31-41-35:~/lep2$
```

Step 7: Change the ownership of new files

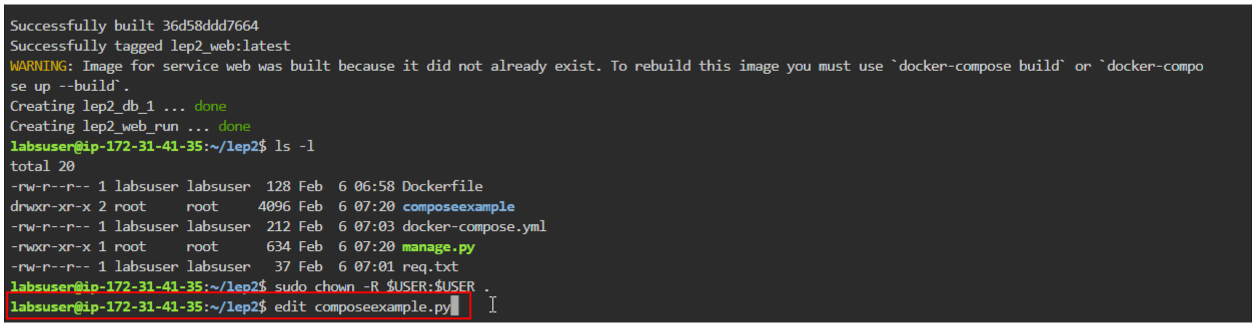
7.1 Use the following command to change the ownership of new files to root:
sudo chown -R $USER:$USER



```
  Created wheel for psycopg2: filename=psycopg2-2.9.9-cp312-cp312-linux_x86_64.whl size=520383 sha256=f9e563e25099849309cb1b13319f12a429d2466aa0a3a9c
9dca1b58ba491830a
  Stored in directory: /root/.cache/pip/wheels/ff/ac/80/7ccec163e3d05ae2112311b895132409b9abfd7e0c1c6b5183
Successfully built psycopg2
Installing collected packages: pytz, sqlparse, psycopg2, Django
Successfully installed Django-2.2.28 psycopg2-2.9.9 pytz-2024.1 sqlparse-0.4.4
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended
 to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.2.1 -> 24.0
[notice] To update, run: pip install --upgrade pip
Removing intermediate container eb83018641cf
 ---> 91b96389275e
Step 7/7 : COPY . /code/
 ---> 36d58ddd7664

Successfully built 36d58ddd7664
Successfully tagged lep2_web:latest
WARNING: Image for service web was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compo
se up --build`.
Creating lep2_db_1 ... done
Creating lep2_web_run ... done
labsuser@ip-172-31-41-35:~/lep2$ ls -l
total 20
-rw-r--r-- 1 labsuser labsuser  128 Feb  6 06:58 Dockerfile
drwxr-xr-x 2 root     root     4096 Feb  6 07:20 composeexample
-rw-r--r-- 1 labsuser labsuser  212 Feb  6 07:03 docker-compose.yml
-rwxr-xr-x 1 root     root      634 Feb  6 07:20 manage.py
-rw-r--r-- 1 labsuser labsuser   37 Feb  6 07:01 req.txt
labsuser@ip-172-31-41-35:~/lep2$ sudo chown -R $USER:$USER .
labsuser@ip-172-31-41-35:~/lep2$
```

Step 8: Set up a database connection

8.1 Edit the **cmps_eg/settings.py** file in the project directory using the following command:
edit composeexample.py

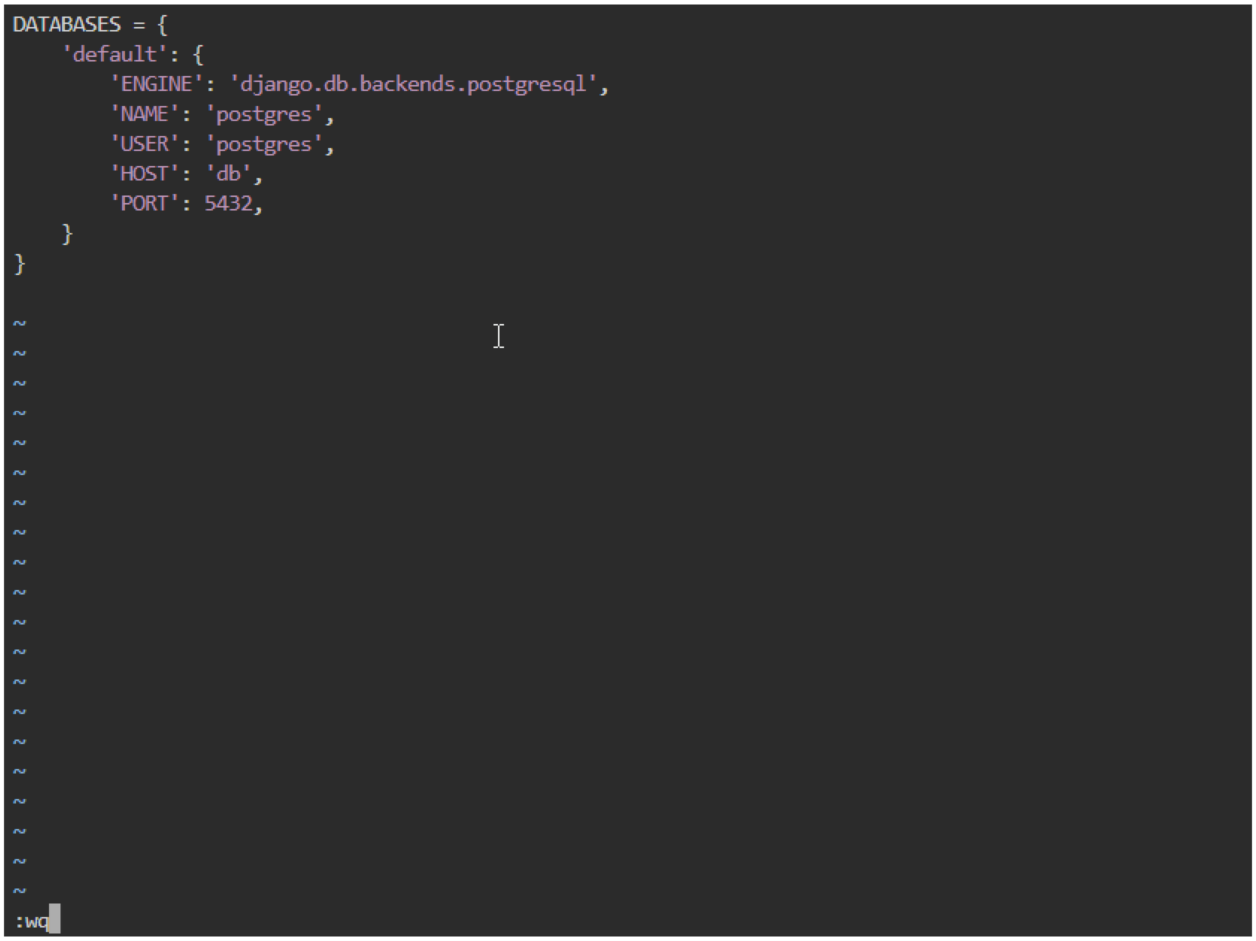


```
Successfully built 36d58ddd7664
Successfully tagged lep2_web:latest
WARNING: Image for service web was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compo
se up --build`.
Creating lep2_db_1 ... done
Creating lep2_web_run ... done
labsuser@ip-172-31-41-35:~/lep2$ ls -l
total 20
-rw-r--r-- 1 labsuser labsuser  128 Feb  6 06:58 Dockerfile
drwxr-xr-x 2 root     root     4096 Feb  6 07:20 composeexample
-rw-r--r-- 1 labsuser labsuser  212 Feb  6 07:03 docker-compose.yml
-rwxr-xr-x 1 root     root      634 Feb  6 07:20 manage.py
-rw-r--r-- 1 labsuser labsuser   37 Feb  6 07:01 req.txt
labsuser@ip-172-31-41-35:~/lep2$ sudo chown -R $USER:$USER .
labsuser@ip-172-31-41-35:~/lep2$ edit composeexample.py
```

Note: Press the **I** key to edit the **composeexample.py** file

8.2 Add the following code **composeexample.py** file:

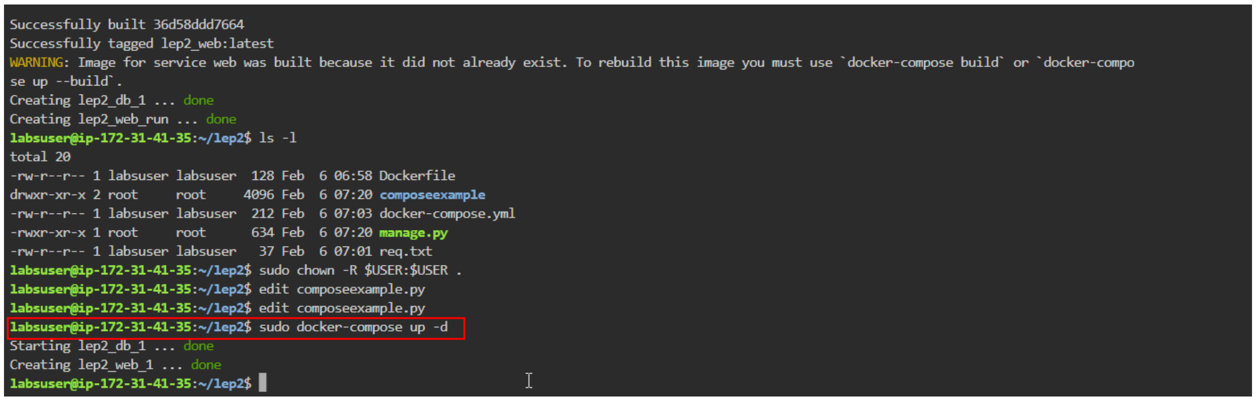
```
    DATABASES = {
  'default': {
          'ENGINE': 'django.db.backends.postgresql',
          'NAME': 'postgres',
          'USER': 'postgres',
          'HOST': 'db',
          'PORT': 5432,
      }
    }
```



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'postgres',
        'USER': 'postgres',
        'HOST': 'db',
        'PORT': 5432,
    }
}

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

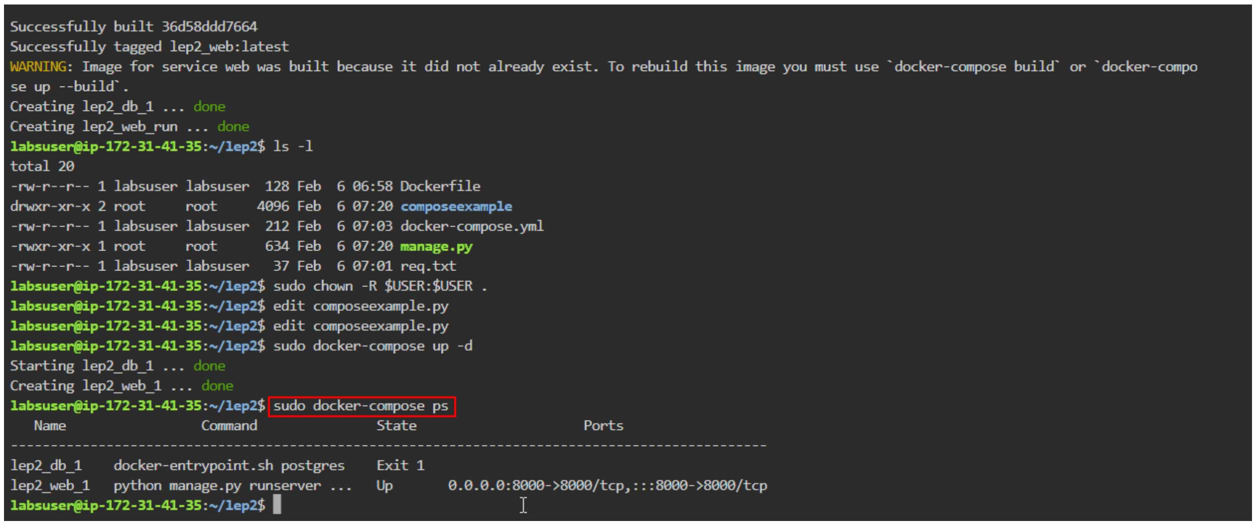
Note: Press the **esc** button, type **wq,** and press **enter** key

8.3 Run the following command to start the application:
sudo docker-compose up -d



```
Successfully built 36d58ddd7664
Successfully tagged lep2_web:latest
WARNING: Image for service web was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compo
se up --build`.
Creating lep2_db_1 ... done
Creating lep2_web_run ... done
labsuser@ip-172-31-41-35:~/lep2$ ls -l
total 20
-rw-r--r-- 1 labsuser labsuser  128 Feb  6 06:58 Dockerfile
drwxr-xr-x 2 root     root     4096 Feb  6 07:20 composeexample
-rw-r--r-- 1 labsuser labsuser  212 Feb  6 07:03 docker-compose.yml
-rwxr-xr-x 1 root     root      634 Feb  6 07:20 manage.py
-rw-r--r-- 1 labsuser labsuser   37 Feb  6 07:01 req.txt
labsuser@ip-172-31-41-35:~/lep2$ sudo chown -R $USER:$USER .
labsuser@ip-172-31-41-35:~/lep2$ edit composeexample.py
labsuser@ip-172-31-41-35:~/lep2$ edit composeexample.py
labsuser@ip-172-31-41-35:~/lep2$ sudo docker-compose up -d
Starting lep2_db_1 ... done
Creating lep2_web_1 ... done
labsuser@ip-172-31-41-35:~/lep2$
```

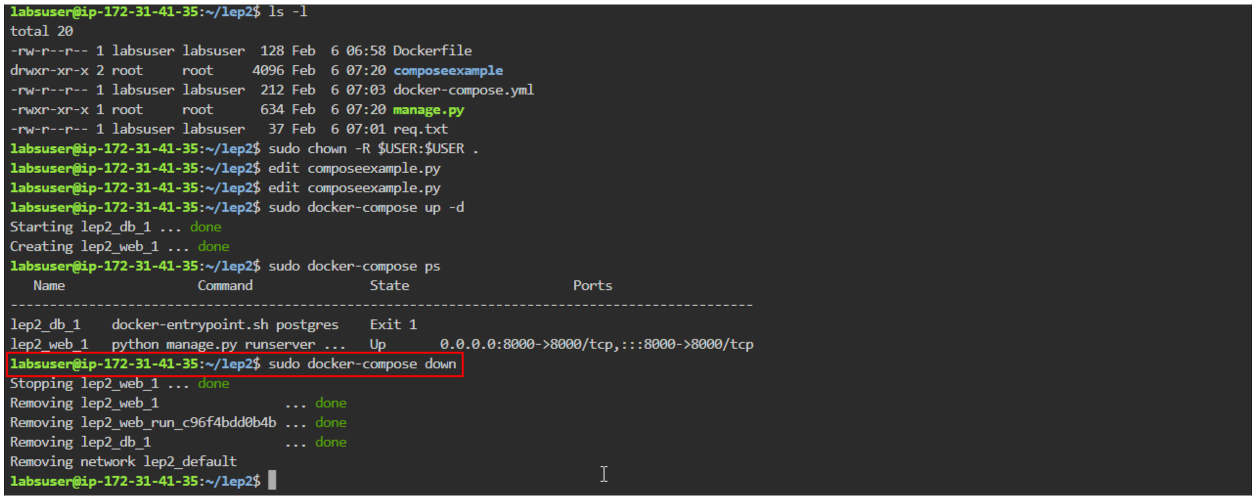
8.4 List the containers using this command:
sudo docker-compose ps



```
Successfully built 36d58ddd7664
Successfully tagged lep2_web:latest
WARNING: Image for service web was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compo
se up --build`.
Creating lep2_db_1 ... done
Creating lep2_web_run ... done
labsuser@ip-172-31-41-35:~/lep2$ ls -l
total 20
-rw-r--r-- 1 labsuser labsuser  128 Feb  6 06:58 Dockerfile
drwxr-xr-x 2 root     root     4096 Feb  6 07:20 composeexample
-rw-r--r-- 1 labsuser labsuser  212 Feb  6 07:03 docker-compose.yml
-rwxr-xr-x 1 root     root      634 Feb  6 07:20 manage.py
-rw-r--r-- 1 labsuser labsuser   37 Feb  6 07:01 req.txt
labsuser@ip-172-31-41-35:~/lep2$ sudo chown -R $USER:$USER .
labsuser@ip-172-31-41-35:~/lep2$ edit composeexample.py
labsuser@ip-172-31-41-35:~/lep2$ edit composeexample.py
labsuser@ip-172-31-41-35:~/lep2$ sudo docker-compose up -d
Starting lep2_db_1 ... done
Creating lep2_web_1 ... done
labsuser@ip-172-31-41-35:~/lep2$ sudo docker-compose ps
   Name               Command               State                     Ports
-------------------------------------------------------------------------------------------------
lep2_db_1    docker-entrypoint.sh postgres    Exit 1
lep2_web_1   python manage.py runserver ...   Up       0.0.0.0:8000->8000/tcp,:::8000->8000/tcp
labsuser@ip-172-31-41-35:~/lep2$
```

8.5 Use the following command to bring the application down:

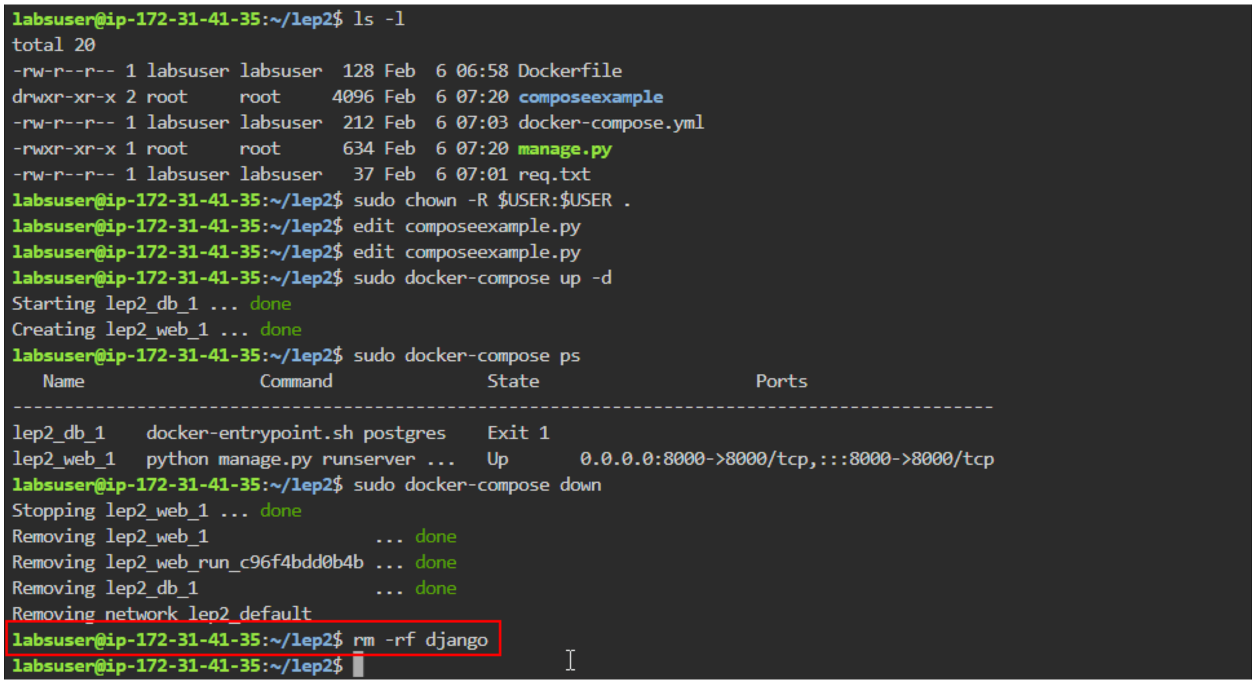
sudo docker-compose down



```
labsuser@ip-172-31-41-35:~/lep2$ ls -l
total 20
-rw-r--r-- 1 labsuser labsuser  128 Feb  6 06:58 Dockerfile
drwxr-xr-x 2 root     root     4096 Feb  6 07:20 composeexample
-rw-r--r-- 1 labsuser labsuser  212 Feb  6 07:03 docker-compose.yml
-rwxr-xr-x 1 root     root      634 Feb  6 07:20 manage.py
-rw-r--r-- 1 labsuser labsuser   37 Feb  6 07:01 req.txt
labsuser@ip-172-31-41-35:~/lep2$ sudo chown -R $USER:$USER .
labsuser@ip-172-31-41-35:~/lep2$ edit composeexample.py
labsuser@ip-172-31-41-35:~/lep2$ edit composeexample.py
labsuser@ip-172-31-41-35:~/lep2$ sudo docker-compose up -d
Starting lep2_db_1 ... done
Creating lep2_web_1 ... done
labsuser@ip-172-31-41-35:~/lep2$ sudo docker-compose ps
   Name                 Command               State                      Ports
-------------------------------------------------------------------------------------------------
lep2_db_1    docker-entrypoint.sh postgres    Exit 1
lep2_web_1   python manage.py runserver ...   Up       0.0.0.0:8000->8000/tcp,:::8000->8000/tcp
labsuser@ip-172-31-41-35:~/lep2$ sudo docker-compose down
Stopping lep2_web_1 ... done
Removing lep2_web_1                  ... done
Removing lep2_web_run_c96f4bdd0b4b ... done
Removing lep2_db_1                   ... done
Removing network lep2_default
labsuser@ip-172-31-41-35:~/lep2$
```

8.6 Remove the Django app by using the following command:

rm -rf django



```
labsuser@ip-172-31-41-35:~/lep2$ ls -l
total 20
-rw-r--r-- 1 labsuser labsuser  128 Feb  6 06:58 Dockerfile
drwxr-xr-x 2 root     root     4096 Feb  6 07:20 composeexample
-rw-r--r-- 1 labsuser labsuser  212 Feb  6 07:03 docker-compose.yml
-rwxr-xr-x 1 root     root      634 Feb  6 07:20 manage.py
-rw-r--r-- 1 labsuser labsuser   37 Feb  6 07:01 req.txt
labsuser@ip-172-31-41-35:~/lep2$ sudo chown -R $USER:$USER .
labsuser@ip-172-31-41-35:~/lep2$ edit composeexample.py
labsuser@ip-172-31-41-35:~/lep2$ edit composeexample.py
labsuser@ip-172-31-41-35:~/lep2$ sudo docker-compose up -d
Starting lep2_db_1 ... done
Creating lep2_web_1 ... done
labsuser@ip-172-31-41-35:~/lep2$ sudo docker-compose ps
   Name                 Command               State                      Ports
-------------------------------------------------------------------------------------------------
lep2_db_1    docker-entrypoint.sh postgres    Exit 1
lep2_web_1   python manage.py runserver ...   Up       0.0.0.0:8000->8000/tcp,:::8000->8000/tcp
labsuser@ip-172-31-41-35:~/lep2$ sudo docker-compose down
Stopping lep2_web_1 ... done
Removing lep2_web_1                  ... done
Removing lep2_web_run_c96f4bdd0b4b ... done
Removing lep2_db_1                   ... done
Removing network lep2_default
labsuser@ip-172-31-41-35:~/lep2$ rm -rf django
labsuser@ip-172-31-41-35:~/lep2$
```

By following these steps, you have successfully containerized the legacy Django application and Postgres database using Docker to ensure improved portability, resource utilization, and simplified deployment processes.