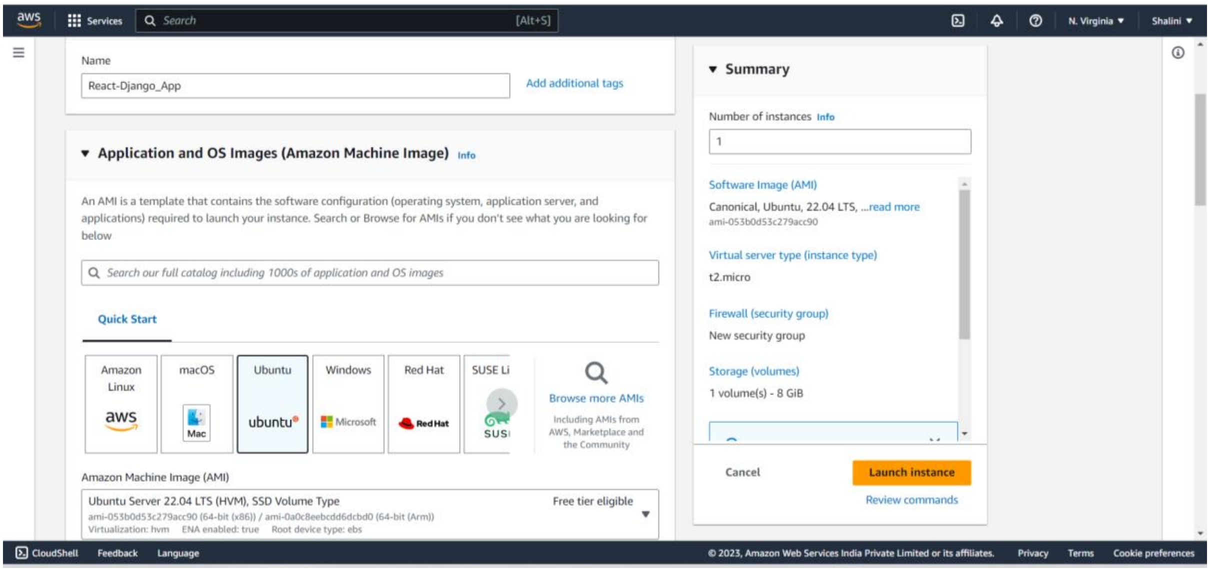
React_Django_App

Create a sample project help of a docker file

Lab – 1: Creating an EC2 Instance

1. To create your EC2 instance, go to Amazon EC2 in the AWS console.
2. Click the button that says Launch instance to open the instance creation.



Name: React-Django_App

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)
Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-053b0d53c279acc90 (64-bit (x86)) / ami-0a0c8eebcdd6dcbd0 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances Info: 1

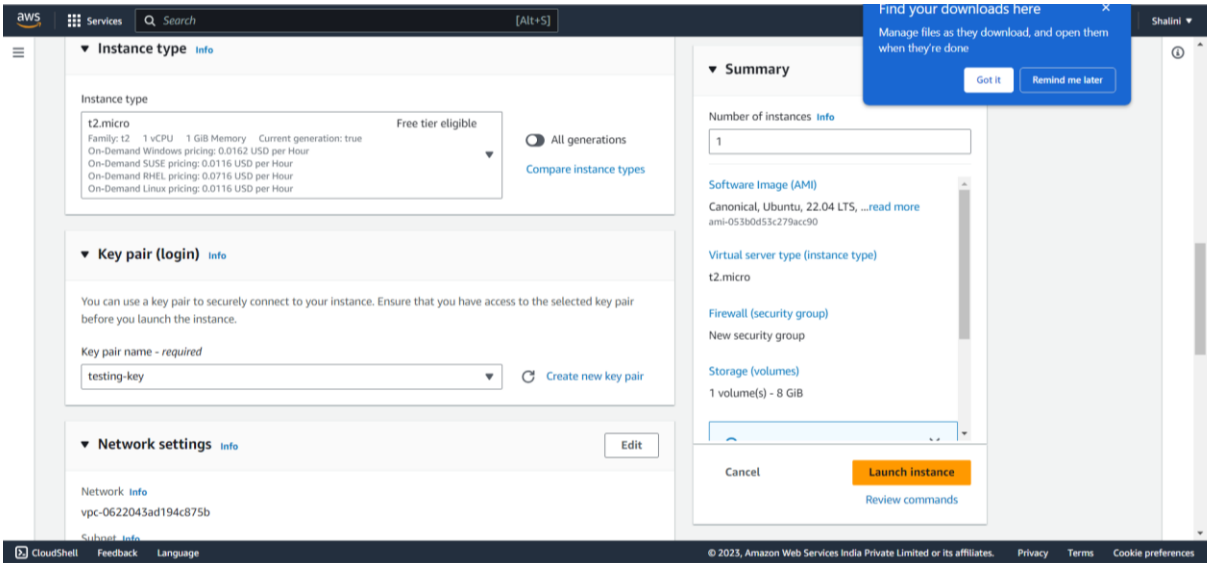
Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

3. I have given the instance name React Django-app
4. For this I choose the amazon linux operating system



Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
On-Demand RHEL pricing: 0.0716 USD per Hour
On-Demand Linux pricing: 0.0116 USD per Hour

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*: testing-key

Network settings

Network: vpc-0622043ad194c875b

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

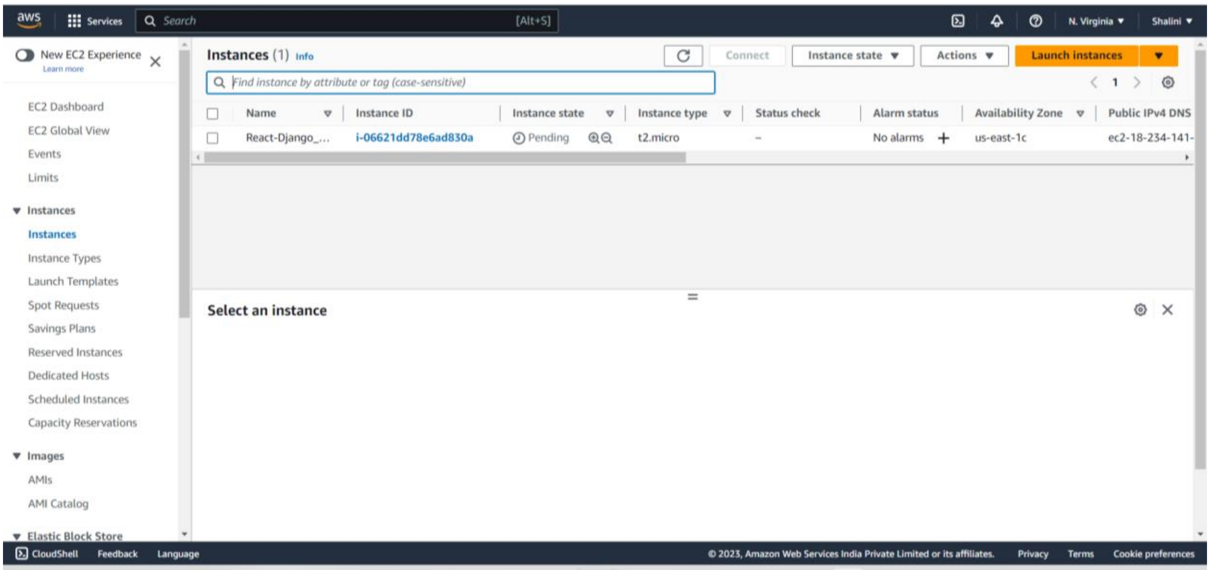
Storage (volumes): 1 volume(s) - 8 GiB

Find your downloads here
Manage files as they download, and open them when they're done

© 2023, Amazon Web Services India Private Limited or its affiliates.

4. I have taken my existing key pair that is testing-key

5. Selected the existing security group with allowing port 8001



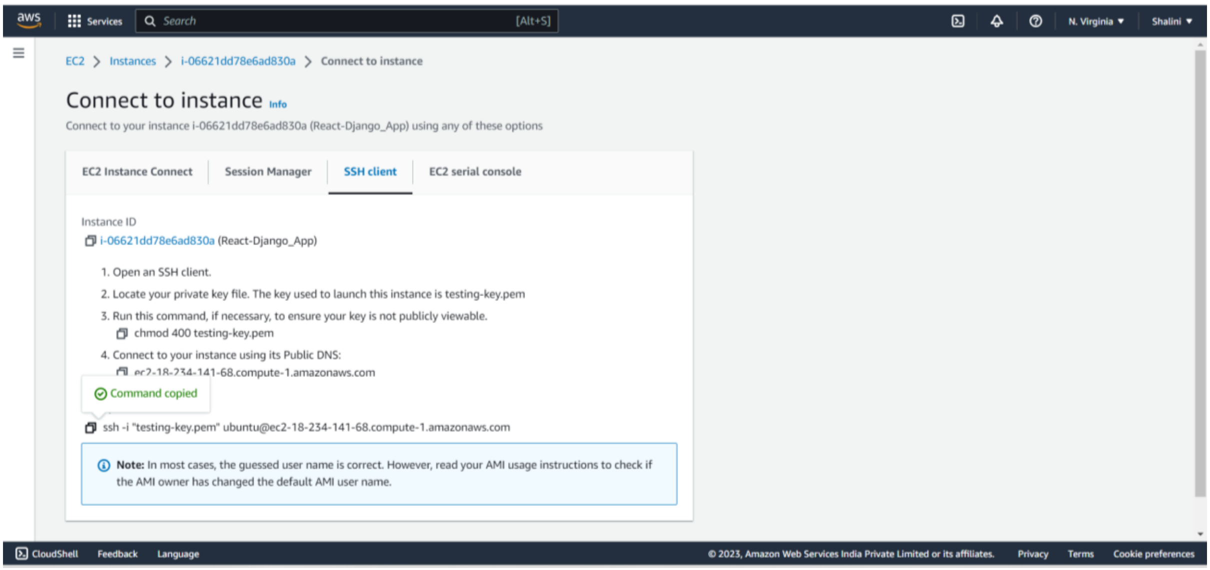
Instances (1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
React-Django_...	i-06621dd78e6ad830a	Pending	t2.micro	–	No alarms	us-east-1c	ec2-18-234-141-

Select an instance

© 2023, Amazon Web Services India Private Limited or its affiliates.

Successfully launched the ec2 instance.



Connect to instance Info

Connect to your instance i-06621dd78e6ad830a (React-Django_App) using any of these options

Instance ID

i-06621dd78e6ad830a (React-Django_App)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is testing-key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 testing-key.pem
4. Connect to your instance using its Public DNS:
   ec2-18-234-141-68.compute-1.amazonaws.com

Command copied

ssh -i "testing-key.pem" ubuntu@ec2-18-234-141-68.compute-1.amazonaws.com

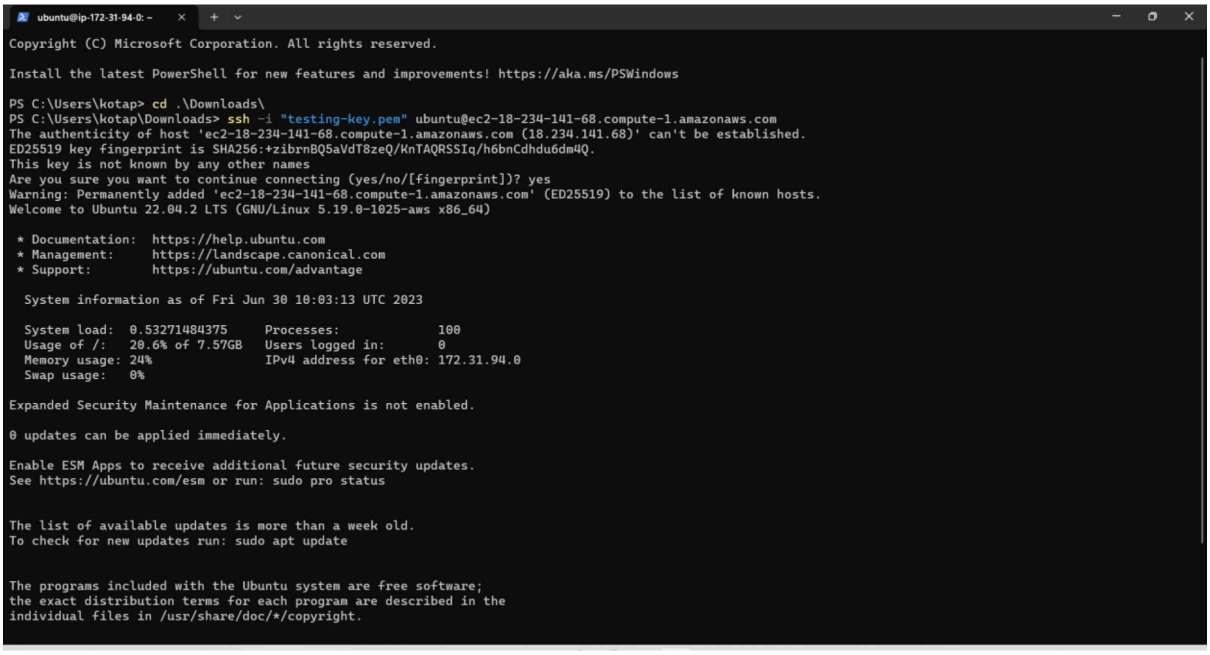
Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

6. Copied the ssh address to connect to the terminal

SSH into your EC2 Instance

1. We are accessing the command line interface with ssh command



```
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\kotap> cd .\Downloads\
PS C:\Users\kotap\Downloads> ssh -i "testing-key.pem" ubuntu@ec2-18-234-141-68.compute-1.amazonaws.com
The authenticity of host 'ec2-18-234-141-68.compute-1.amazonaws.com (18.234.141.68)' can't be established.
ED25519 key fingerprint is SHA256:+zibrnBQ5aVdT8zeQ/KnTAQRSSIq/h6bnCdhdu6dm4Q.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-234-141-68.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Jun 30 10:03:13 UTC 2023

  System load:  0.53271484375     Processes:             100
  Usage of /:   20.6% of 7.57GB   Users logged in:       0
  Memory usage: 24%               IPv4 address for eth0: 172.31.94.0
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

Module – 2: Installing Docker

1. we have to install the docker in our instance by using below command.

- ➢ **sudo apt install docker.io**

start the docker service:

2. After installing docker , start the docker service by running the following command.

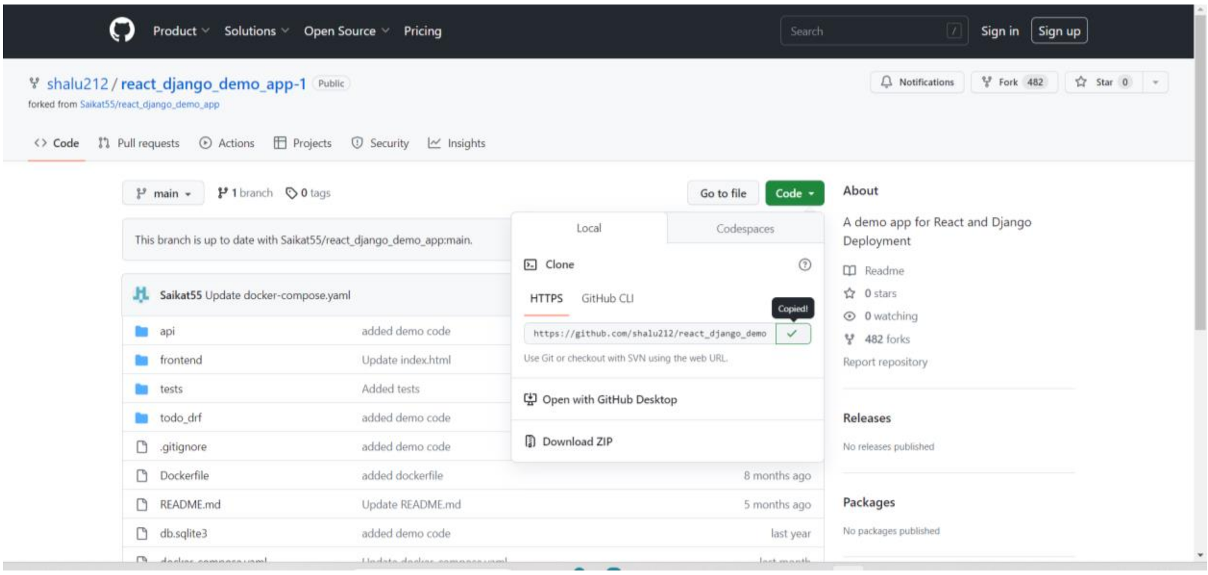
- ➢ **sudo systemctl start docker**

3. I haven given the permission to the docker demon in oder to work with docker by running the below command.

- ➢ **sudo chmod 666 /var/run/docker.sock**

Cloning project source code from github

1. I copied the code from from github.



shalu212 / react_django_demo_app-1 Public

forked from Saikat55/react_django_demo_app

This branch is up to date with Saikat55/react_django_demo_app:main.

Saikat55 Update docker-compose.yaml

api	added demo code	
frontend	Update index.html	
tests	Added tests	
todo_drf	added demo code	
.gitignore	added demo code	
Dockerfile	added dockerfile	8 months ago
README.md	Update README.md	5 months ago
db.sqlite3	added demo code	last year

https://github.com/shalu212/react_django_demo

Use Git or checkout with SVN using the web URL.

About

A demo app for React and Django Deployment

0 stars

0 watching

482 forks

Releases

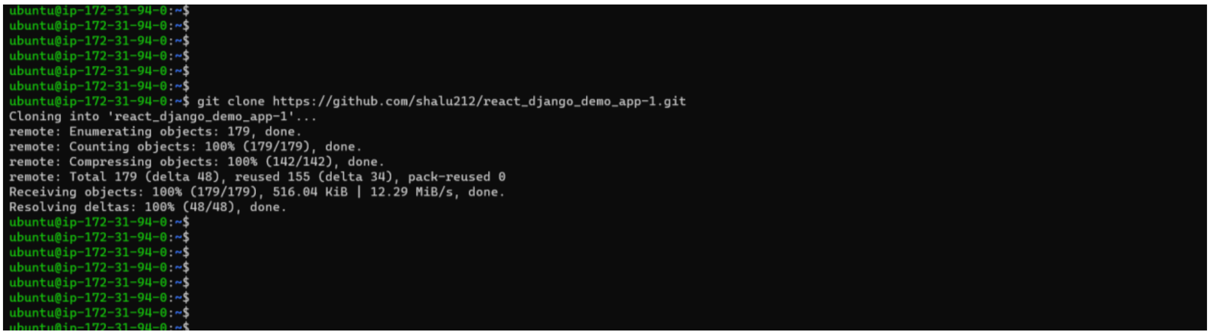
No releases published

Packages

No packages published

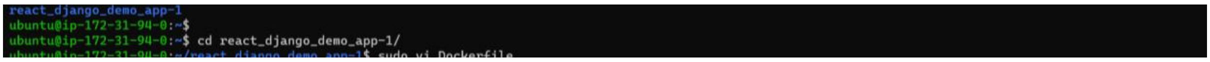
2. Cloned the project source code using the command.

- ➢ **git clone https://github.com/shalu212/react_django_demo_app-1.git**



```
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$ git clone https://github.com/shalu212/react_django_demo_app-1.git
Cloning into 'react_django_demo_app-1'...
remote: Enumerating objects: 179, done.
remote: Counting objects: 100% (179/179), done.
remote: Compressing objects: 100% (142/142), done.
remote: Total 179 (delta 48), reused 155 (delta 34), pack-reused 0
Receiving objects: 100% (179/179), 516.04 KiB | 12.29 MiB/s, done.
Resolving deltas: 100% (48/48), done.
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
```

3. change the directory to the project directory where your application code is located.



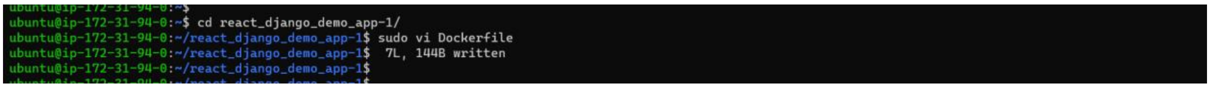
```
react_django_demo_app-1
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$ cd react_django_demo_app-1/
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$ sudo vi Dockerfile
```

cd react_django_demo_app-1

4. In project directory create a new file named "Dockerfile" by running the following command .

   touch Dockerfile

5. By using **sudo vi Dockerfile** command opened the vim editor.



```
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$ cd react_django_demo_app-1/
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$ sudo vi Dockerfile
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$  7L, 144B written
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$
```

6. Edit the Dockerfile and spcify the instructions for building the docker image.

You can see my dockerfile below.

FROM: To pull the base image. FROM is the mandatory keyword specifies the base image.

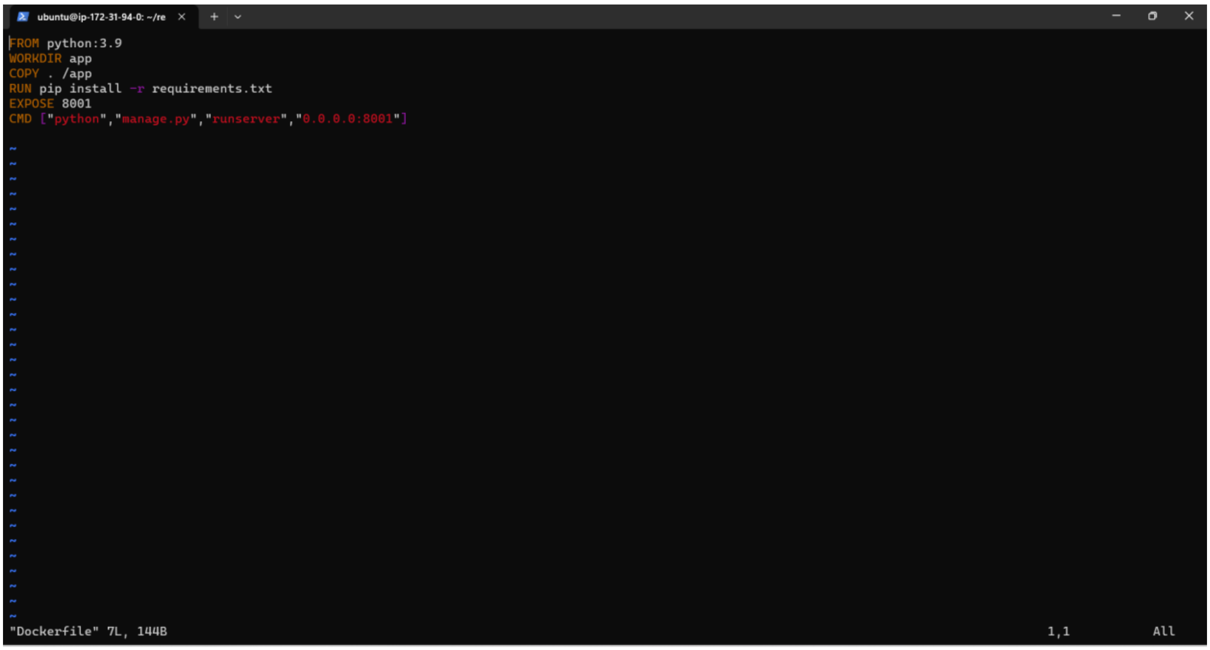
WORKDIR: workdir is used to specify the default directory where the command will be excuted . setting the working directory inside container.

COPY: COPY is used to copy files from host mechine to the container directory .

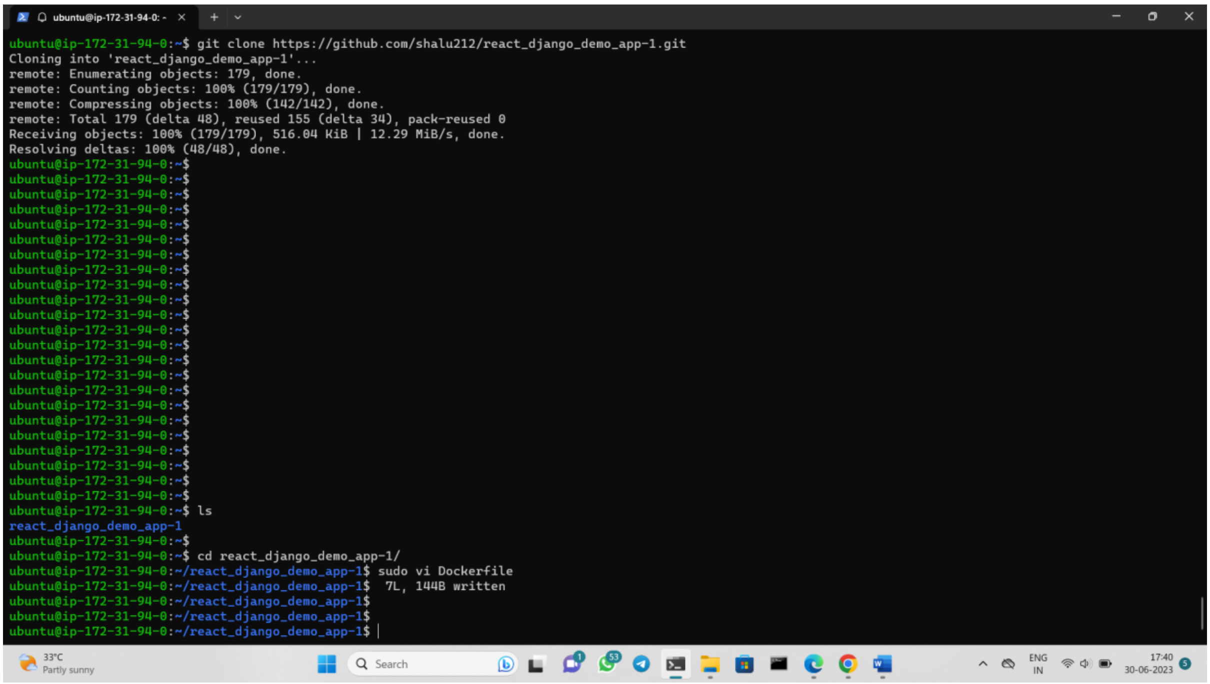
RUN: it excutes command. That can be repeated multiple times. It is used to provide the linux commands like installing packages , uninstalling, upgrading and creating directories.

EXPOSE: The keyword EXPOSE specifies the port number that’s need to be exposed on the container.

CMD: it provides defaults for an excuting container . if you want run commands during execution you can use CMD . whatever we mention it is replaceable.



```
FROM python:3.9
WORKDIR app
COPY . /app
RUN pip install -r requirements.txt
EXPOSE 8001
CMD ["python","manage.py","runserver","0.0.0.0:8001"]

"Dockerfile" 7L, 144B                                              1,1           All
```



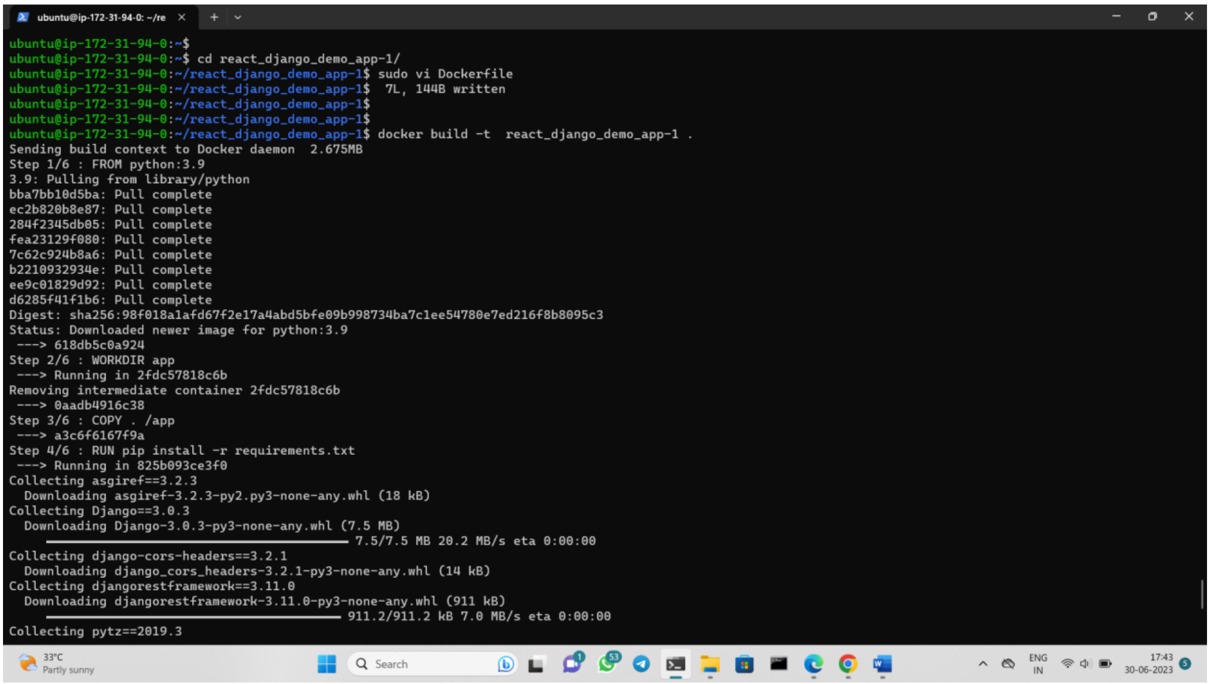
```
ubuntu@ip-172-31-94-0:~$ git clone https://github.com/shalu212/react_django_demo_app-1.git
Cloning into 'react_django_demo_app-1'...
remote: Enumerating objects: 179, done.
remote: Counting objects: 100% (179/179), done.
remote: Compressing objects: 100% (142/142), done.
remote: Total 179 (delta 48), reused 155 (delta 34), pack-reused 0
Receiving objects: 100% (179/179), 516.04 KiB | 12.29 MiB/s, done.
Resolving deltas: 100% (48/48), done.
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$ ls
react_django_demo_app-1
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$ cd react_django_demo_app-1/
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$ sudo vi Dockerfile
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$  7L, 144B written
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$ |
```

7. Save the docker file and exit the text editor by using :wq!.

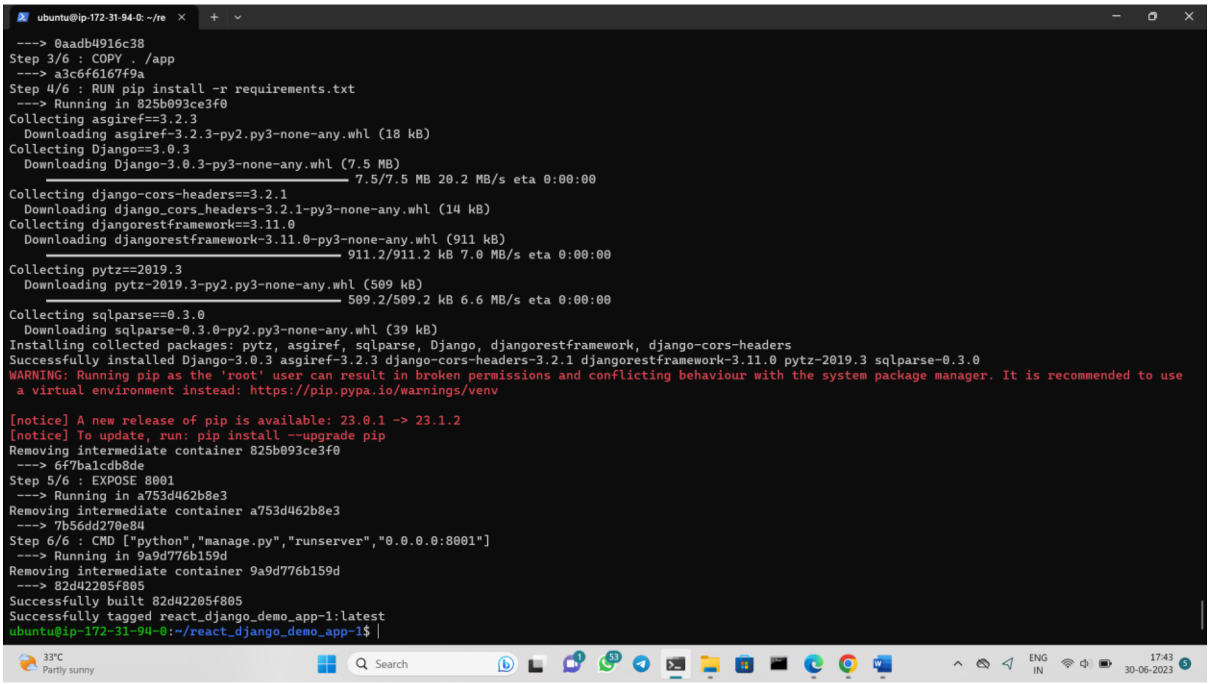
8. After creating the dockerfile, run the following command to build the docker image.

9. The command to create docker image

- **docker build -t react-Django-app-1 .**



```
ubuntu@ip-172-31-94-0:~$
ubuntu@ip-172-31-94-0:~$ cd react_django_demo_app-1/
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$ sudo vi Dockerfile
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$  7L, 144B written
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$ docker build -t  react_django_demo_app-1 .
Sending build context to Docker daemon  2.675MB
Step 1/6 : FROM python:3.9
3.9: Pulling from library/python
bba7bb10d5ba: Pull complete
ec2b820b8e87: Pull complete
284f2345db05: Pull complete
fea23129f080: Pull complete
7c62c924b8a6: Pull complete
b2210932934e: Pull complete
ee9c01829d92: Pull complete
d6285f41f1b6: Pull complete
Digest: sha256:98f018a1afd67f2e17a4abd5bfe09b998734ba7c1ee54780e7ed216f8b8095c3
Status: Downloaded newer image for python:3.9
 ---> 618db5c0a924
Step 2/6 : WORKDIR app
 ---> Running in 2fdc57818c6b
Removing intermediate container 2fdc57818c6b
 ---> 0aadb4916c38
Step 3/6 : COPY . /app
 ---> a3c6f6167f9a
Step 4/6 : RUN pip install -r requirements.txt
 ---> Running in 825b093ce3f0
Collecting asgiref==3.2.3
  Downloading asgiref-3.2.3-py2.py3-none-any.whl (18 kB)
Collecting Django==3.0.3
  Downloading Django-3.0.3-py3-none-any.whl (7.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.5/7.5 MB 20.2 MB/s eta 0:00:00
Collecting django-cors-headers==3.2.1
  Downloading django_cors_headers-3.2.1-py3-none-any.whl (14 kB)
Collecting djangorestframework==3.11.0
  Downloading djangorestframework-3.11.0-py3-none-any.whl (911 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 911.2/911.2 kB 7.0 MB/s eta 0:00:00
Collecting pytz==2019.3
```



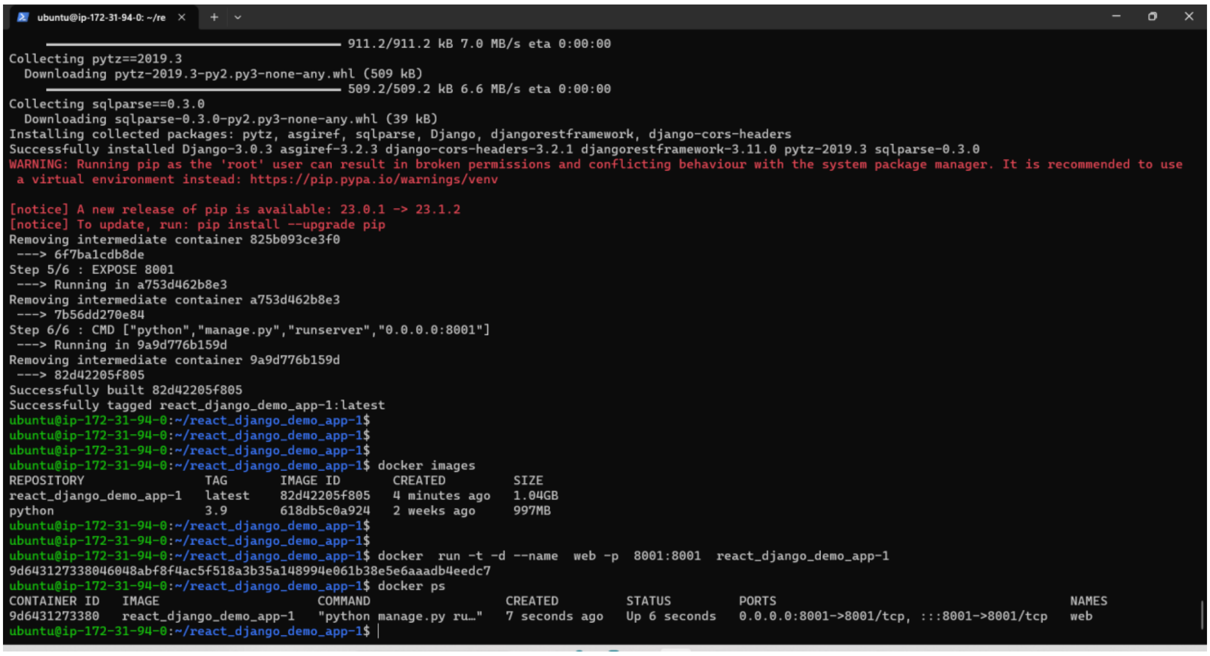
```
 ---> 0aadb4916c38
Step 3/6 : COPY . /app
 ---> a3c6f6167f9a
Step 4/6 : RUN pip install -r requirements.txt
 ---> Running in 825b093ce3f0
Collecting asgiref==3.2.3
  Downloading asgiref-3.2.3-py2.py3-none-any.whl (18 kB)
Collecting Django==3.0.3
  Downloading Django-3.0.3-py3-none-any.whl (7.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.5/7.5 MB 20.2 MB/s eta 0:00:00
Collecting django-cors-headers==3.2.1
  Downloading django_cors_headers-3.2.1-py3-none-any.whl (14 kB)
Collecting djangorestframework==3.11.0
  Downloading djangorestframework-3.11.0-py3-none-any.whl (911 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 911.2/911.2 kB 7.0 MB/s eta 0:00:00
Collecting pytz==2019.3
  Downloading pytz-2019.3-py2.py3-none-any.whl (509 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 509.2/509.2 kB 6.6 MB/s eta 0:00:00
Collecting sqlparse==0.3.0
  Downloading sqlparse-0.3.0-py2.py3-none-any.whl (39 kB)
Installing collected packages: pytz, asgiref, sqlparse, Django, djangorestframework, django-cors-headers
Successfully installed Django-3.0.3 asgiref-3.2.3 django-cors-headers-3.2.1 djangorestframework-3.11.0 pytz-2019.3 sqlparse-0.3.0
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: 23.0.1 -> 23.1.2
[notice] To update, run: pip install --upgrade pip
Removing intermediate container 825b093ce3f0
 ---> 6f7ba1cdb8de
Step 5/6 : EXPOSE 8001
 ---> Running in a753d462b8e3
Removing intermediate container a753d462b8e3
 ---> 7b56dd270e84
Step 6/6 : CMD ["python","manage.py","runserver","0.0.0.0:8001"]
 ---> Running in 9a9d776b159d
Removing intermediate container 9a9d776b159d
 ---> 82d42205f805
Successfully built 82d42205f805
Successfully tagged react_django_demo_app-1:latest
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$ 
```

Successfully created the docker image

10. After building the docker image run the following command the to start the docker container.

11. The command to create container from docker image is.

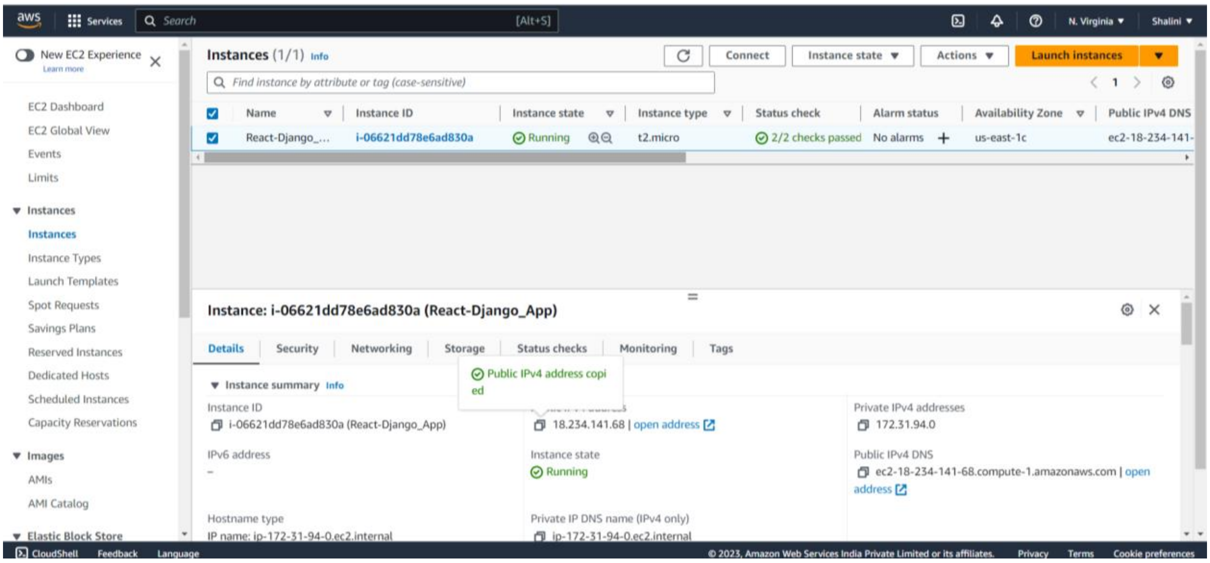
- **docker run -t -d --name web -p 8001:8001 react_django_demo_app-1**



```
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 911.2/911.2 kB 7.0 MB/s eta 0:00:00
Collecting pytz==2019.3
  Downloading pytz-2019.3-py2.py3-none-any.whl (509 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 509.2/509.2 kB 6.6 MB/s eta 0:00:00
Collecting sqlparse==0.3.0
  Downloading sqlparse-0.3.0-py2.py3-none-any.whl (39 kB)
Installing collected packages: pytz, asgiref, sqlparse, Django, djangorestframework, django-cors-headers
Successfully installed Django-3.0.3 asgiref-3.2.3 django-cors-headers-3.2.1 djangorestframework-3.11.0 pytz-2019.3 sqlparse-0.3.0
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: 23.0.1 -> 23.1.2
[notice] To update, run: pip install --upgrade pip
Removing intermediate container 825b093ce3f0
 ---> 6f7ba1cdb8de
Step 5/6 : EXPOSE 8001
 ---> Running in a753d462b8e3
Removing intermediate container a753d462b8e3
 ---> 7b56dd270e84
Step 6/6 : CMD ["python","manage.py","runserver","0.0.0.0:8001"]
 ---> Running in 9a9d776b159d
Removing intermediate container 9a9d776b159d
 ---> 82d42205f805
Successfully built 82d42205f805
Successfully tagged react_django_demo_app-1:latest
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$ docker images
REPOSITORY                TAG       IMAGE ID       CREATED         SIZE
react_django_demo_app-1   latest    82d42205f805   4 minutes ago   1.04GB
python                    3.9       618db5c0a924   2 weeks ago     997MB
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$ docker  run -t -d --name  web -p  8001:8001  react_django_demo_app-1
9d6431273380460 48abf8f4ac5f518a3b35a148994e061b38e5e6aaadb4eedc7
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$ docker ps
CONTAINER ID   IMAGE                     COMMAND                  CREATED         STATUS         PORTS                                       NAMES
9d6431273380   react_django_demo_app-1   "python manage.py ru…"   7 seconds ago   Up 6 seconds   0.0.0.0:8001->8001/tcp, :::8001->8001/tcp   web
ubuntu@ip-172-31-94-0:~/react_django_demo_app-1$ |
```

12. To see if container is successfully created or not run the **docker ps** command it will shows the running containers.

13. My docker container is successfully created you can see in the above image and Browse the app. 19. Once the docker container is running . you can browse the app by opening web browser and navigating to the public ip of your instance ,followed by the **port 8001.**



Instances (1/1) Info

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☑	React-Django_...	i-06621dd78e6ad830a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-18-234-141-

Instance: i-06621dd78e6ad830a (React-Django_App)

Instance summary Info

Public IPv4 address copied

Instance ID
i-06621dd78e6ad830a (React-Django_App)

Public IPv4 address
18.234.141.68 | open address

Private IPv4 addresses
172.31.94.0

IPv6 address
–

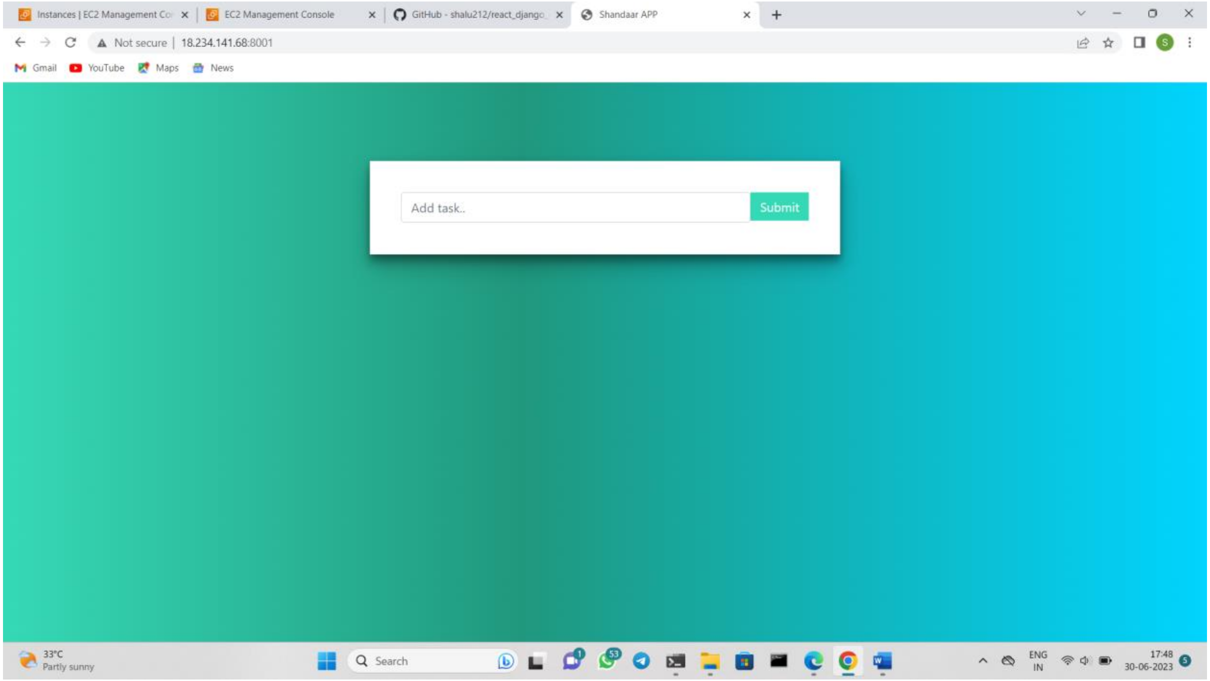
Instance state
Running

Public IPv4 DNS
ec2-18-234-141-68.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-94-0.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-94-0.ec2.internal

http://18.234.141.68:8001/



And the lastly , I had look to see that this was running correctly

Shalini.K