

Assignment SDE

Use source code from 'aesthisia-demo/' directory & use the src to build docker image for the same. To run the app, use command: 'npm start' Run the docker image on port 3000 & check the output on '<http://localhost:3000>'

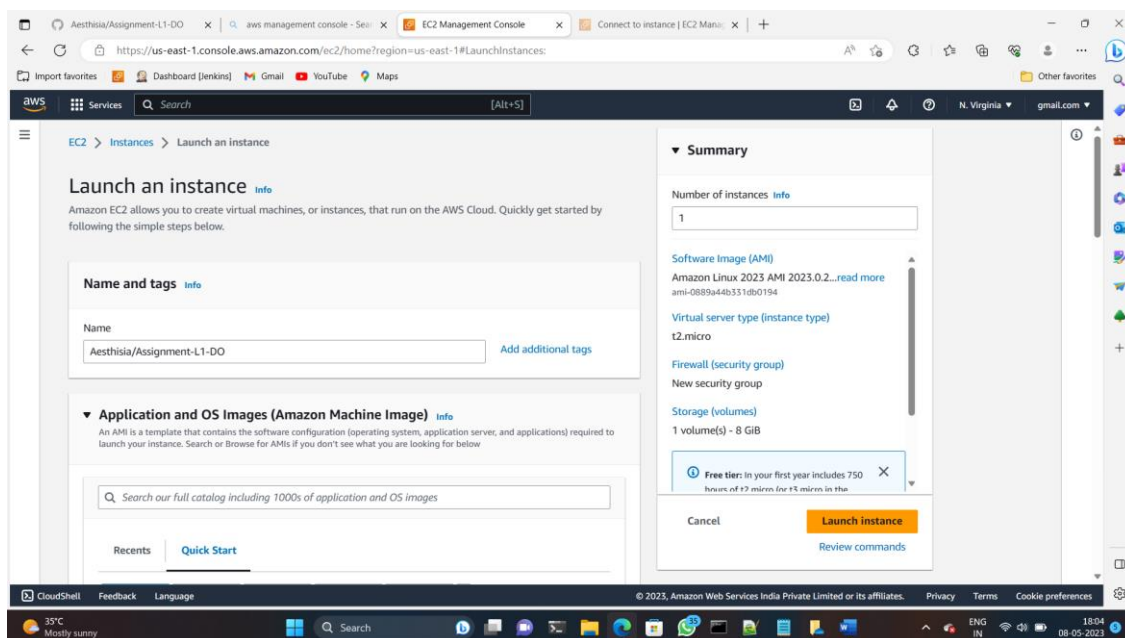
Tools and version required

- Docker
- Git
- Port: 3000

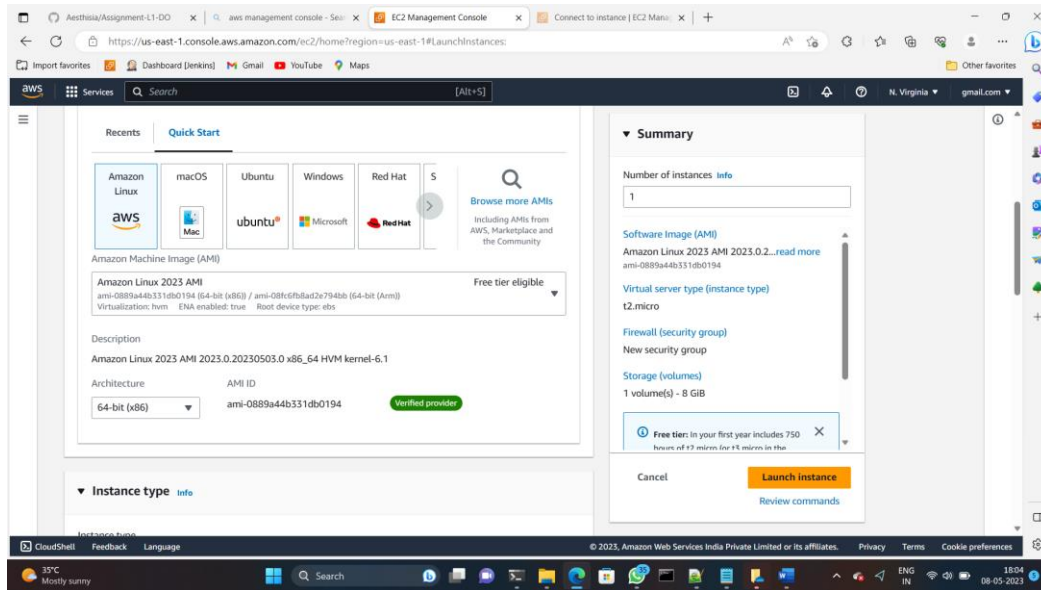
Lab – 1: Creating an EC2 Instance

1. To create your EC2 instance, go to Amazon EC2 in the AWS console.

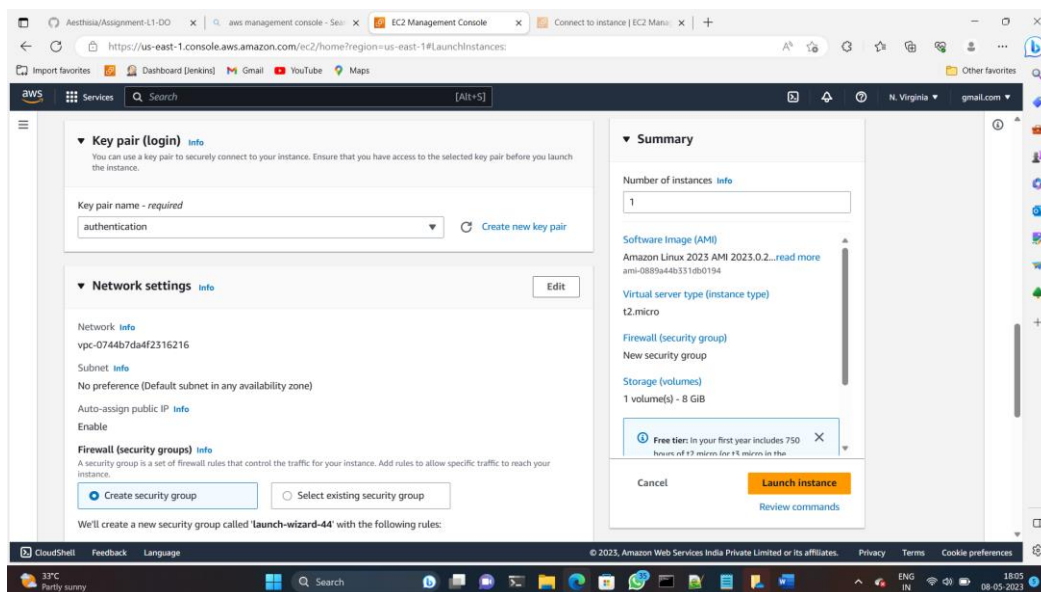
Click the button that says Launch instance to open the instance creation.



I have given the instance name Aesthisia/Assignment-L1-DO

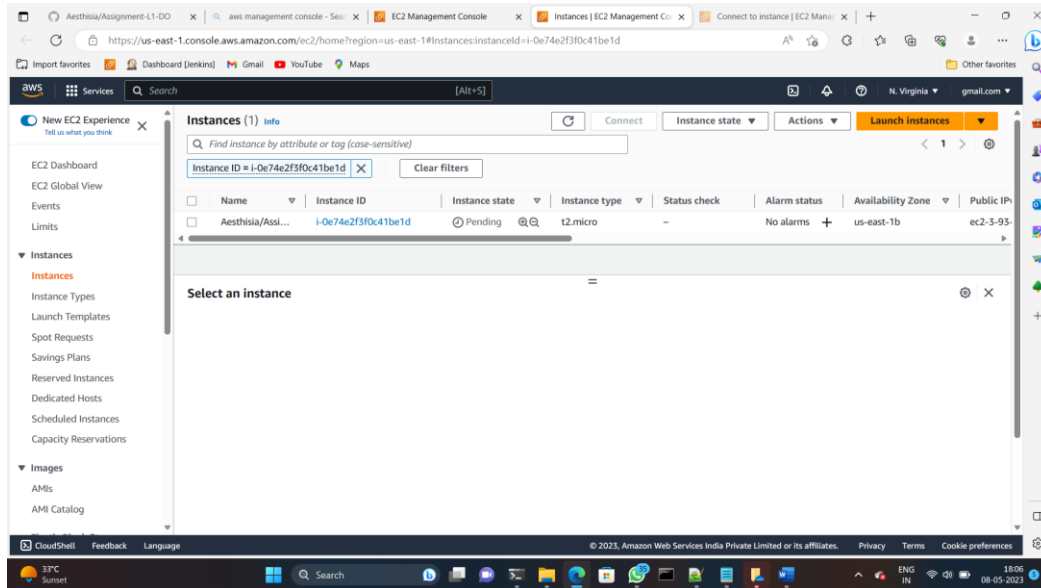
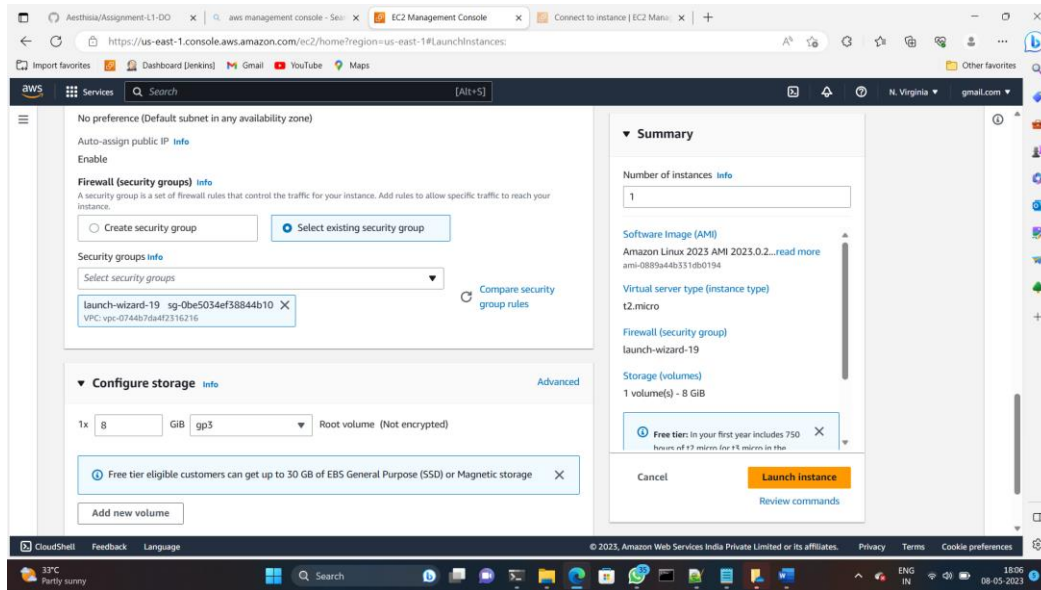


3. For this I choose the amazon linux operating system

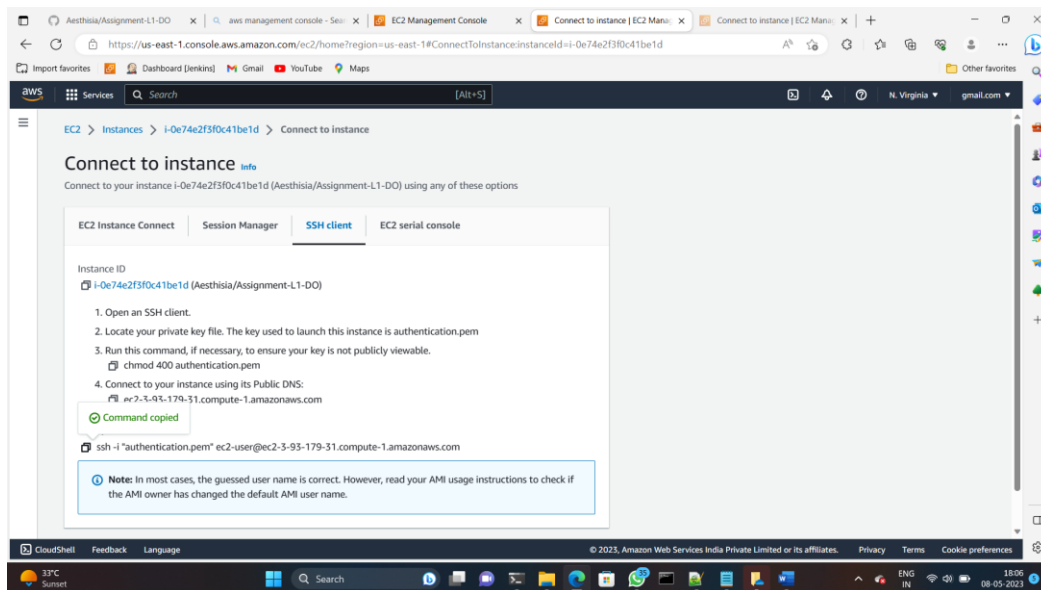


4. I have taken my existing key pair that is **authentication.pem**

5. Selected the existing security group with allowing port **3000**.



Successfully launched the ec2 instance.



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

```
ec2-user@ip-172-31-88-7:~$ ssh -i "authentication.pem" ec2-user@ec2-3-93-179-31.compute-1.amazonaws.com
Windows PowerShell
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 "authentication.pem" ec2-user@ec2-3-93-179-31.compute-1.amazonaws.com
The authenticity of host 'ec2-3-93-179-31.compute-1.amazonaws.com (3.93.179.31)' can't be established.
ED25519 key fingerprint is SHA256:3doIjeSpCs2YXrva/YR+0+90573tQYdkLDuU64JpfJI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-93-179-31.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#
# Amazon Linux 2023
#
# https://aws.amazon.com/linux/amazon-linux-2023
#

[ec2-user@ip-172-31-88-7 ~]$
```

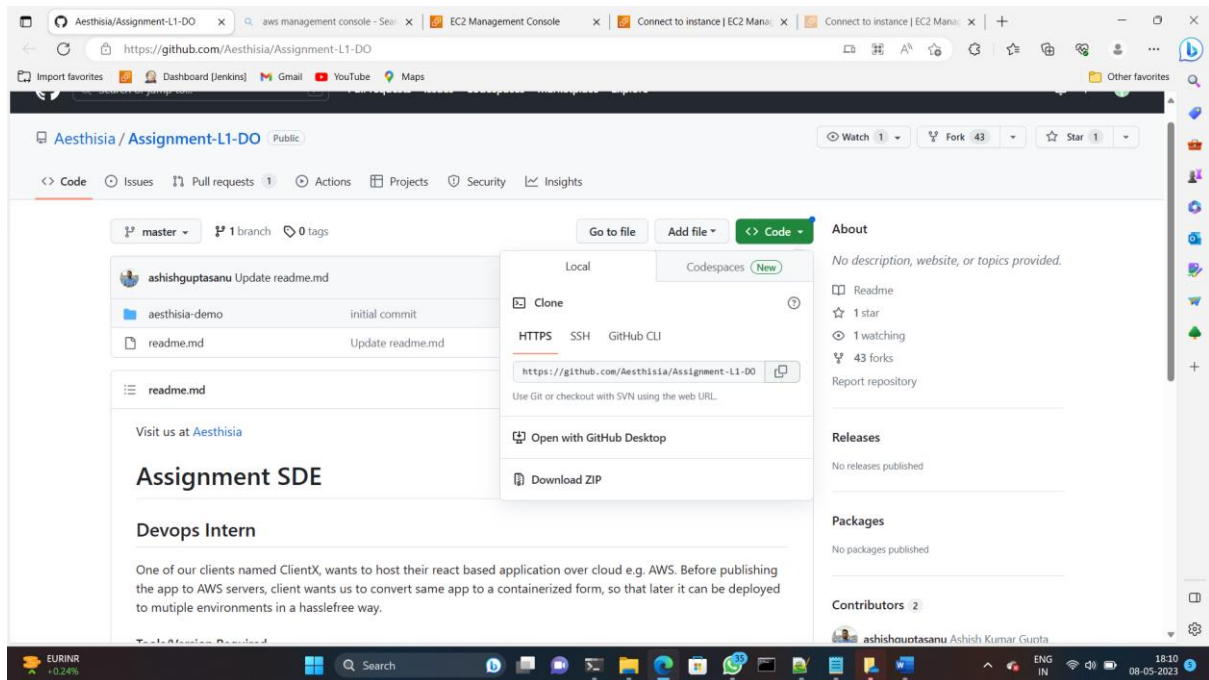
Module – 2: Installing GIT, Docker, and related repos

1. we have to install the git and docker in our instance by using below commands.

➤ **sudo yum -y install git**

Cloning project source code from github:

- I copied the code from from github



1. Cloned the project source code using the command.

- **git clone** <https://github.com/Aesthisia/Assignment-L1-DO.git>

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

3. **cd Aesthisia/**

4. **cd aesthisia-demo/**

start the docker service:

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

- **sudo systemctl start docker**

6. Enable the docker service by running the following command .

➤ **sudo systemctl enable docker**

7. 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**

```
ec2-user@ip-172-31-88-7:~$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$ sudo systemctl start docker  
[ec2-user@ip-172-31-88-7 ~]$ sudo systemctl enable docker  
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$ git clone https://github.com/Aesthisia/Assignment-L1-D0.git  
Cloning into 'Assignment-L1-D0'...  
remote: Enumerating objects: 30, done.  
remote: Counting objects: 100% (1/1), done.  
remote: Total 30 (delta 0), reused 0 (delta 0), pack-reused 29  
Receiving objects: 100% (30/30), 302.23 KiB | 14.39 MiB/s, done.  
Resolving deltas: 100% (2/2), done.  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$  
[ec2-user@ip-172-31-88-7 ~]$ ls  
Assignment-L1-D0  
[ec2-user@ip-172-31-88-7 ~]$ cd Assignment-L1-D0/  
[ec2-user@ip-172-31-88-7 Assignment-L1-D0]$ cd aesthisia-demo/  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$ touch Dockerfile  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$ ls
```

8. In project directory create a new file named “**Dockerfile**” by running the following command .

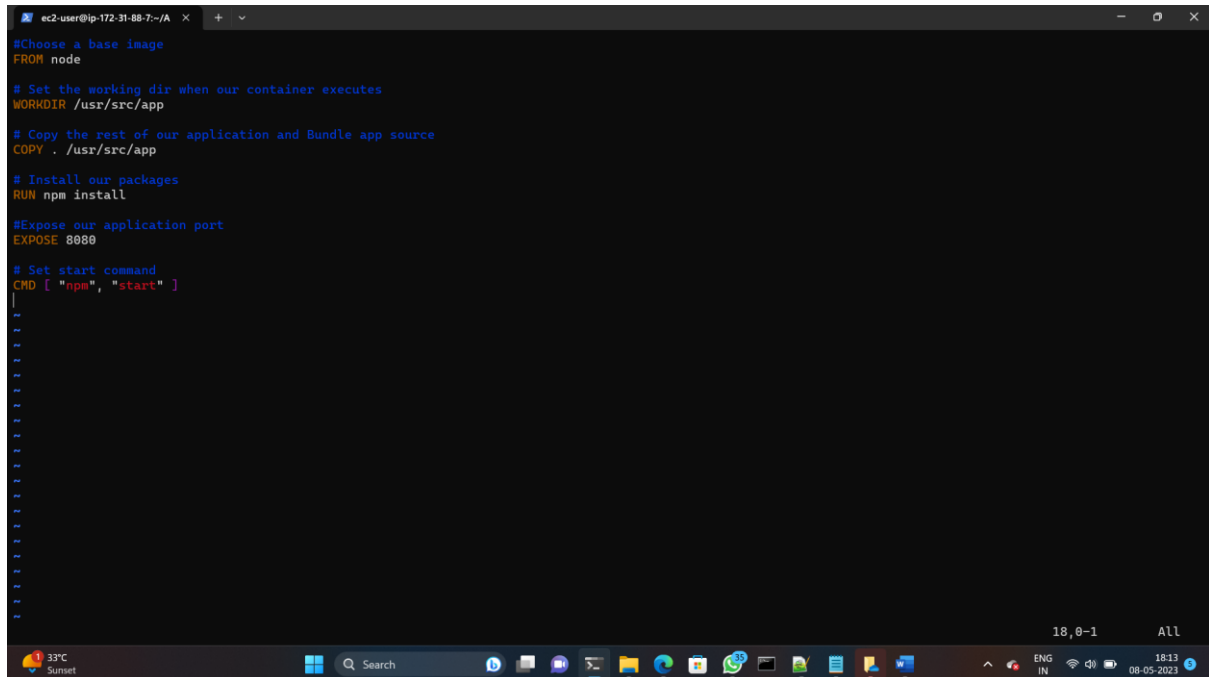
➤ **touch Dockerfile**

```
[ec2-user@ip-172-31-88-7 aesthisia-demo]$  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$ touch Dockerfile  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$ ls  
Dockerfile README.md package-lock.json package.json public src  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$ sudo vi Dockerfile  
18L, 308B written  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$  
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
```

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

10. Edit the Dockerfile and specify the instructions for building the docker image.

11. You can see my dockerfile below.



```
ec2-user@ip-172-31-88-7:~/A
#Choose a base image
FROM node

# Set the working dir when our container executes
WORKDIR /usr/src/app

# Copy the rest of our application and Bundle app source
COPY . /usr/src/app

# Install our packages
RUN npm install

#Expose our application port
EXPOSE 8080

# Set start command
CMD [ "npm", "start" ]
```

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

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

14. The command to create docker image

➤ **docker build -t react .**

react is my image name


```
ec2-user@ip-172-31-88-7:~$ docker build -t react .
Sending build context to Docker daemon 1.226MB
Step 1/6 : FROM node
latest: Pulling from library/node
918547b94326: Pull complete
5d79863a01c5: Pull complete
4eedd9c5abf7: Pull complete
9cdadd40855f: Pull complete
b58a51cba8a5: Pull complete
b95104135278: Pull complete
155685169ccf: Pull complete
fe4676ed347: Pull complete
Digest: sha256:0efc3ef3fea2822c9d16da084c40181ed7f74b6f45141100580f9887ccc8e9a1
Status: Downloaded newer image for node:latest
----> acd15857ce39
Step 2/6 : WORKDIR /usr/src/app
----> Running in 893273d68ce7
Removing intermediate container 893273d68ce7
----> c8605a7efee8
Step 3/6 : COPY . /usr/src/app
----> 01102b48251f
Step 4/6 : RUN npm install
----> Running in 156527f1533f
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.
npm WARN deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm WARN deprecated source-map-resolve@0.6.0: See https://github.com/lydell/source-map-resolve#deprecated

added 1444 packages, and audited 1445 packages in 49s

194 packages are looking for funding
  run `npm fund` for details

14 vulnerabilities (13 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix
```

```
----> 01102b48251f
Step 4/6 : RUN npm install
----> Running in 156527f1533f
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.
npm WARN deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm WARN deprecated source-map-resolve@0.6.0: See https://github.com/lydell/source-map-resolve#deprecated

added 1444 packages, and audited 1445 packages in 49s

194 packages are looking for funding
  run `npm fund` for details

14 vulnerabilities (13 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
npm notice
npm notice New patch version of npm available! 9.6.4 -> 9.6.6
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.6>
npm notice Run `npm install -g npm@9.6.6` to update!
npm notice
Removing intermediate container 156527f1533f
----> 23f034ca8b05
Step 5/6 : EXPOSE 8080
----> Running in 4824ef805e35
Removing intermediate container 4824ef805e35
----> 4781f2e87040
Step 6/6 : CMD [ "npm", "start" ]
----> Running in 33c5027cc1c4
Removing intermediate container 33c5027cc1c4
----> 272aeb121890
Successfully built 272aeb121890
Successfully tagged react:latest
ec2-user@ip-172-31-88-7:~$
```

```
ec2-user@ip-172-31-88-7:~$ npm audit
added 1444 packages, and audited 1445 packages in 49s

194 packages are looking for funding
  run `npm fund` for details

14 vulnerabilities (13 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
npm notice
npm notice New patch version of npm available! 9.6.4 -> 9.6.6
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.6>
npm notice Run `npm install -g npm@9.6.6` to update!
npm notice
Removing intermediate container 156527f1533f
----> 23f034ca8b05
Step 5/6 : EXPOSE 8080
----> Running in 4824ef805e35
Removing intermediate container 4824ef805e35
----> 4781f2e87040
Step 6/6 : CMD [ "npm", "start" ]
----> Running in 33c5027ccl4
Removing intermediate container 33c5027ccl4
----> 272aeb121890
Successfully built 272aeb121890
Successfully tagged react:latest
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$ docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
react         latest   272aeb121890  About a minute ago  1.27GB
node          latest   acd15857ce39  3 days ago     1GB
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
```

Successfully created the docker image .

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

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

➤ **docker build -t -d --name npm -p 3000:3000 react**

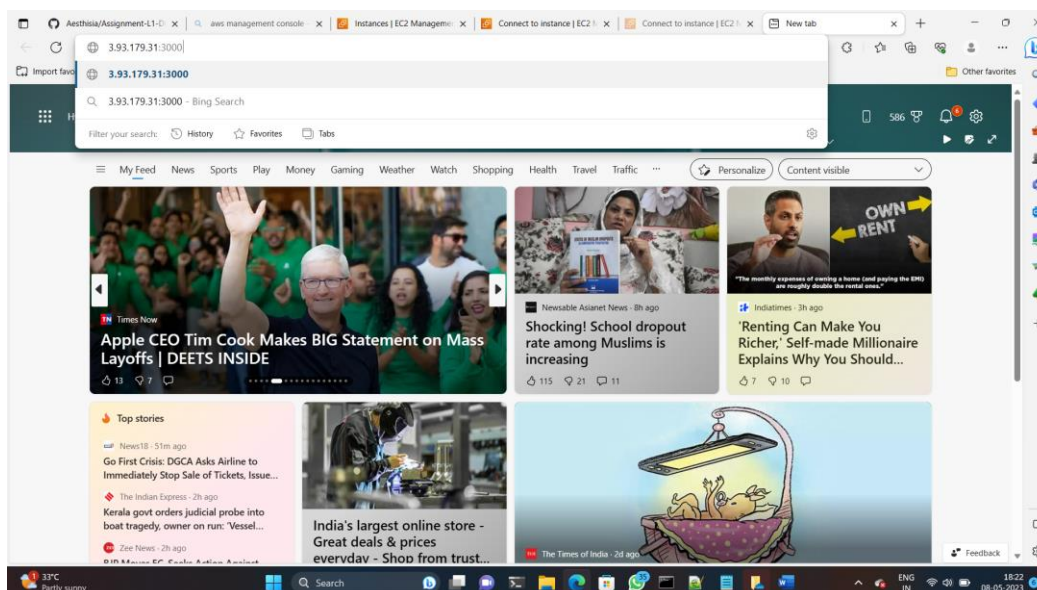
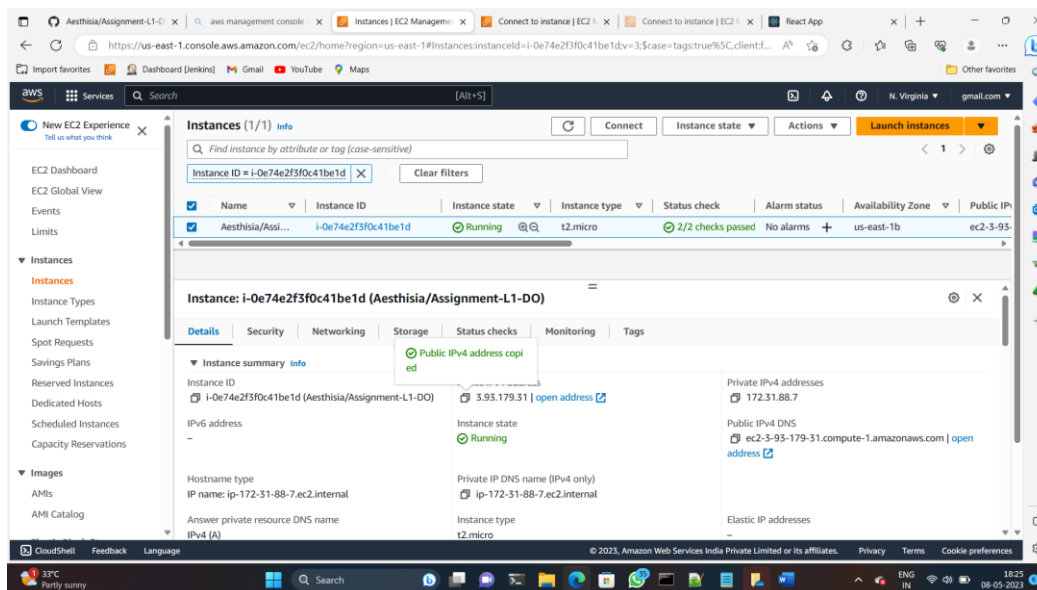
```
ec2-user@ip-172-31-88-7:~$ npm audit fix --force

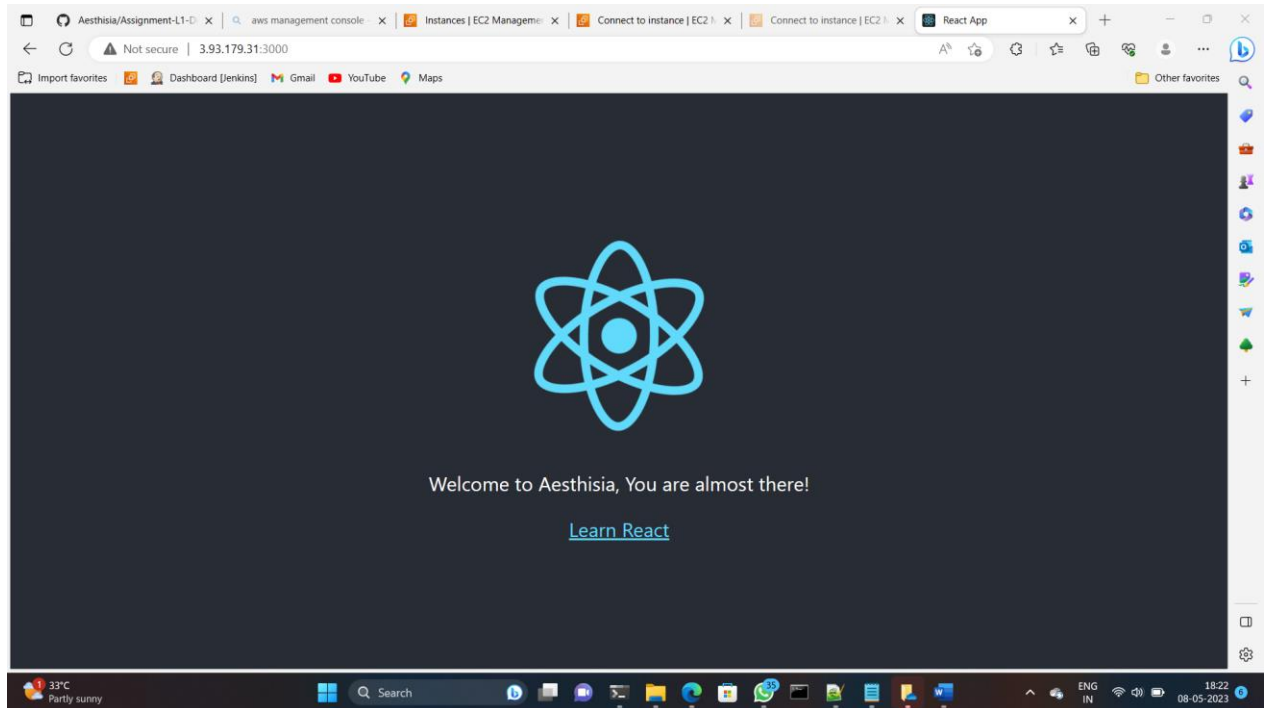
Run `npm audit` for details.
npm notice
npm notice New patch version of npm available! 9.6.4 -> 9.6.6
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.6>
npm notice Run `npm install -g npm@9.6.6` to update!
npm notice
Removing intermediate container 156527f1533f
----> 23f034ca8b05
Step 5/6 : EXPOSE 8080
----> Running in 4824ef805e35
Removing intermediate container 4824ef805e35
----> 4781f2e87040
Step 6/6 : CMD [ "npm", "start" ]
----> Running in 33c5027ccl4
Removing intermediate container 33c5027ccl4
----> 272aeb121890
Successfully built 272aeb121890
Successfully tagged react:latest
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$ docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
react         latest   272aeb121890  About a minute ago  1.27GB
node          latest   acd15857ce39  3 days ago     1GB
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$ docker run -t -d --name npm -p 3000:3000 react
70a206199e9a5242655083a9fc9361c10537c3206e8d1aa64f0be57d27be1832
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
[ec2-user@ip-172-31-88-7 aesthisia-demo]$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
70a206199e9a   react     "docker-entrypoint.s..." 23 seconds ago Up 21 seconds 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp, 8080/tcp  npm
[ec2-user@ip-172-31-88-7 aesthisia-demo]$
```

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

18. 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 3000.





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

Thank you ,

Shalini.K