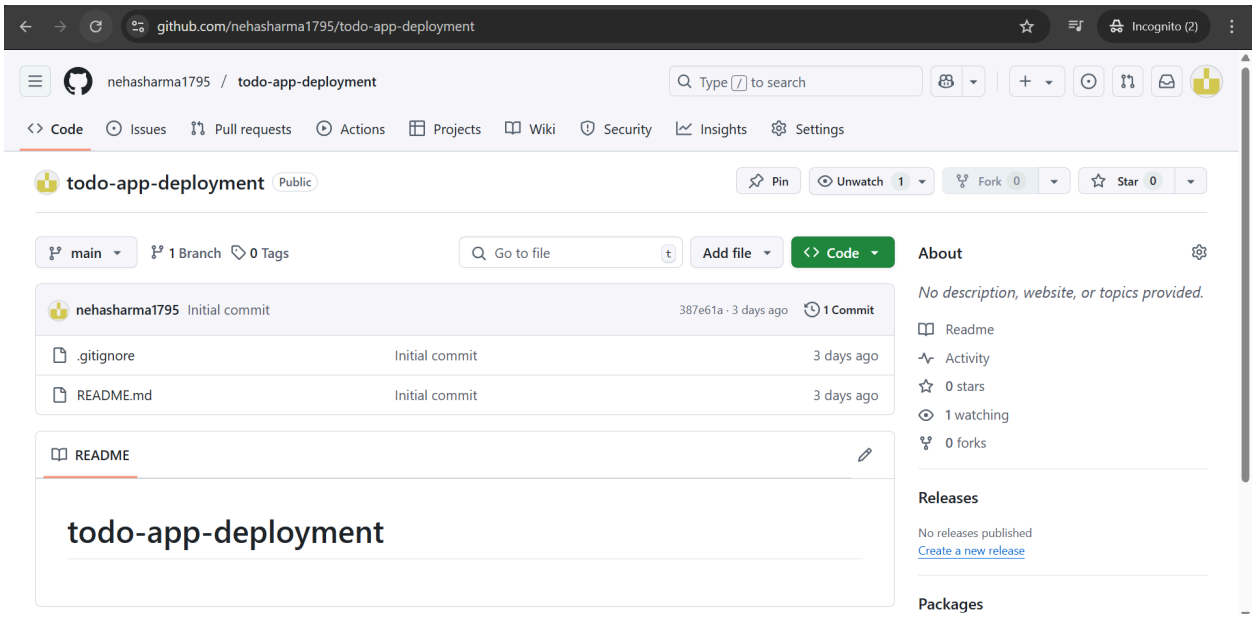
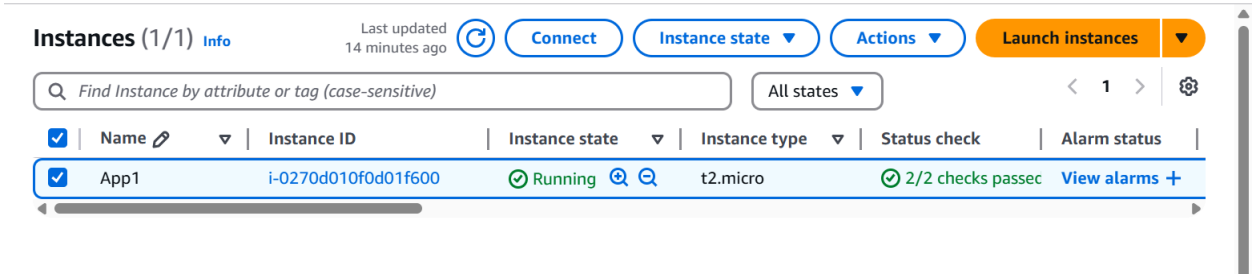
Document created by Neha Sharma

I'll provide a structured approach to completing DevOps assignment,
**********AWS & Git Clone**********

1) Create a github repo:



2) Create a AWS account
3) Create a EC2 instance



4) **Login to the EC2(VM) instance using ssh to the mobaxterm.**
Copy .pem key on a local path then
Open mobaxterm
Go to that path and execute the below command
ssh -i "app1.pem" ec2-user@ec2-65-2-149-133.ap-south-1.compute.amazonaws.com

```
2025-03-24  22:52.33  /home/mobaxterm/Desktop  ssh -i "app1.pem" ec2-user@ec2-65-2-149-133.ap-south-1.compute.amazonaws.com
X11 forwarding request failed on channel 0
Register this system with Red Hat Insights: rhc connect

Example:
# rhc connect --activation-key <key> --organization <org>

The rhc client and Red Hat Insights will enable analytics and additional
management capabilities on your system.
View your connected systems at https://console.redhat.com/insights

You can learn more about how to register your system
using rhc at https://red.ht/registration
[ec2-user@ip-172-31-34-187 ~]$
[ec2-user@ip-172-31-34-187 ~]$
[ec2-user@ip-172-31-34-187 ~]$
[ec2-user@ip-172-31-34-187 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-34-187 ~]$ client_loop: send disconnect: Connection reset by peer
```

5) Install git on EC2 instance

Sudo -i

Yum install git -y

```
  perl-vars-1.05-481.el9.noarch

Complete!
[root@ip-172-31-34-187 ~]# git --version
git version 2.43.5
[root@ip-172-31-34-187 ~]# yum install git -y
```

6) Set Up SSH Keys

If using SSH authentication, generate an SSH key:

Go to Mobaxterm

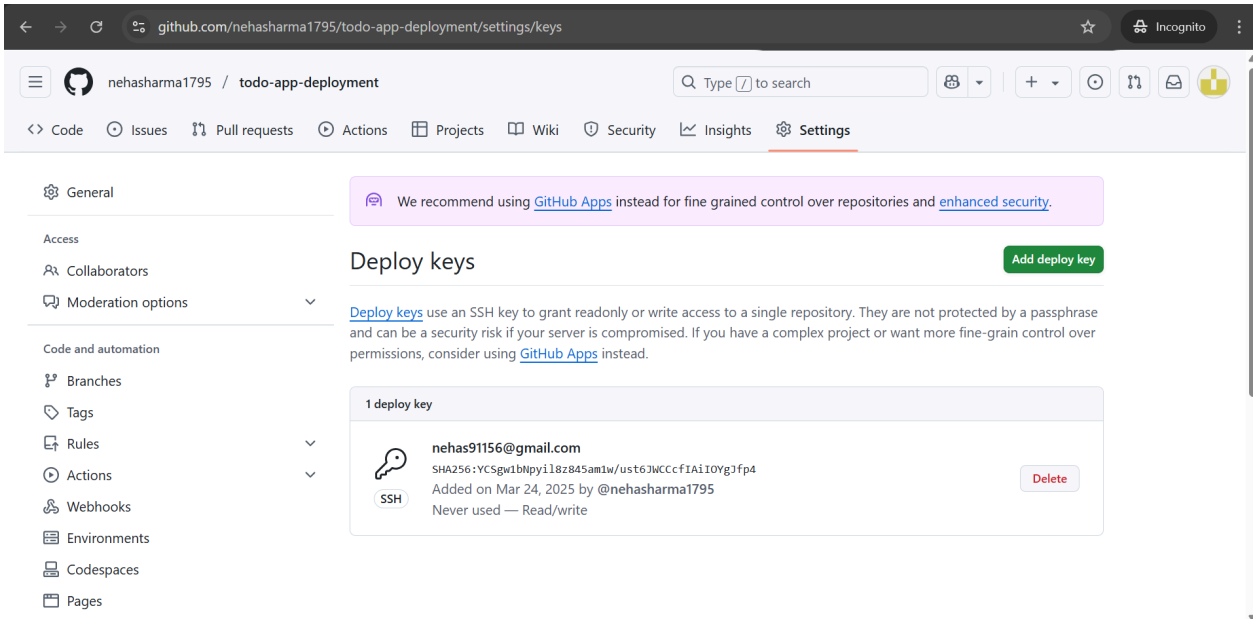
Execute the command : ssh-keygen -t rsa -b 4096 -C "nehas91156@gmail.com"

```
[ec2-user@ip-172-31-34-187 ~]$ ssh-keygen -t rsa -b 4096 -C "nehas91156@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ec2-user/.ssh/id_rsa): yes
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in yes
Your public key has been saved in yes.pub
The key fingerprint is:
SHA256:rA1wzZmzrKoCcItB0v7cwwyFHap97N52bVHnmVNRnH8 nehas91156@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
| .   o..       .+|
|... ..oo o     o.|
|o.  o.. *       o|
|o oo.+ o o    . E|
|.+.+.=+ S    . o=|
|o . oo==    .  +.|
```

cat ~/.ssh/authorized_keys Git.pub

```
You can learn more about how to register your system
using rhc at https://red.ht/registration
Last login: Mon Mar 24 17:42:02 2025 from 151.186.194.3
[ec2-user@ip-172-31-34-187 ~]$ cat ~/.ssh/authorized_keys
.bash_logout   .bash_profile  .bashrc        .cache/        Git            Git.pub        .ssh/
[ec2-user@ip-172-31-34-187 ~]$ cat ~/.ssh/authorized_keys Git.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCiG5kVWkEul51YNNsEmM1W8koi2wIgiWGxiDwZh/teQ3J/XXAPs1st7h6hIDT1tBtDiikny1GN4Ma0BSVVF4+dHP
LHlxpKDMT0f7ZmcsekV1wvl9k4jbEJmRZuLxBwjcDtbGkheqBZOiXtcmqrb/UDZMQbrXvSLVndJ7cky72KZf9YOibyySgarAlhp2lyjMT1hMwcV2aMOIRWhmKlXkwd
KsHu1TZJe7XG3QK1ln3sXKdW0JxigLUjyuUo8rZAw2xacZrfIS5JnqExIPpklq15NGN2QNMqFwCgOO12RZNAth+ED8JR4/27UlygdUDoTNYwZ2n5j0NIrQDPnNavhr
jX app1
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCp9Bi3XWPDc3BDd3GRW6VmGeRGnd4H+qygjadlhqKVZCshpPE7EKxu/qmIeI1NWTmnvimpmJsF5JamZIJDUaRvna
0D9yLVob6YT0ukfjIvsIFh3hT9XaF8Ik2m6NvYo88EXb9neKyk+hoflNLxKecwVK7Y9qTKJs/Q3/+OAqf36yT9AO/JAWBMSenmaNCsMZfhdaqGFzrFY+leiiE15B5v
u4I5P1+NOIxHyKfxbaafWiAuQkp29aPp3k7burMhpHrtYQ9uzTa2x6z6BXigI9XWfZAy/DKlzT3+Y8lSckAt8o8YYgxBbPGjB9ShMQdHJ9ARJT8uK3ojy/yIokdQrQ
CjzouGcn4Pd11w6wCvPRzjBq8p9ckOPaXLaz4W5U6glBiM37HF+rcidmBJo59r0PJQWkM3Ex3TJpqXy7kLIjEtlG+BIFzbJEGETi7V4vU0ej+Go0dbhde5uszwiJq+
LlW7kYsqe/nXrDd4cFFWM9vOitOP8/EmcUt9noQj+sqzM83ZaZ0ZV4Nua3byxEIco62d8IDfxc55B7EekRG/CQJxvbi8k8EVcnaQfyl9NhchlW5+XZSUg6GOycpdWO
VWmOHApgnAKoyJQLB+cXbN3EfxDN8ENty/+2s5cLuRUw9V4ccfOc2cAdFu3++V7tFaFG85eERDZReUeRBMJH0rxzJdew== nehas91156@gmail.com
[ec2-user@ip-172-31-34-187 ~]$
```

Go to **GitHub → Settings →Deploy keys → New SSH Key = "paste the key"**
Paste the key and save.



7) Clone the Repository

Navigate to the directory where you want to store the project:
cd /path/to/your/directory
git clone https://github.com/your-username/repository-name.git

```
[ec2-user@ip-172-31-34-187 opt]$ sudo -i
[root@ip-172-31-34-187 ~]# mkdir project
[root@ip-172-31-34-187 ~]# git clone  https://github.com/nehasharma1795/todo-app-deployment.git
Cloning into 'todo-app-deployment'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
[root@ip-172-31-34-187 ~]# ll
total 0
drwxr-xr-x. 2 root root  6 Mar 24 18:03 project
drwxr-xr-x. 3 root root 53 Mar 24 18:03 todo-app-deployment
[root@ip-172-31-34-187 ~]# cd todo-app-deployment/
[root@ip-172-31-34-187 todo-app-deployment]# ll
total 4
-rw-r--r--. 1 root root 21 Mar 24 18:03 README.md
[root@ip-172-31-34-187 todo-app-deployment]# cd ..
```

8) **Verify the Clone**
Enter the project folder:
cd repository-name
Check the repository status:
git status
To list remote repositories:
git remote -v

**********Docker SetuP**********

1) **Download docker packages from internet using wget command**

wget https://download.docker.com/linux/static/stable/x86_64/docker-25.0.3.tgz

```
[root@ip-172-31-34-187 opt]# wget https://download.docker.com/linux/static/stable/x86_64/docker-25.0.3.tgz
--2025-03-25 13:25:59--  https://download.docker.com/linux/static/stable/x86_64/docker-25.0.3.tgz
Resolving download.docker.com (download.docker.com)... 13.225.5.103, 13.225.5.128, 13.225.5.54, ...
Connecting to download.docker.com (download.docker.com)|13.225.5.103|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 71570637 (68M) [application/x-tar]
Saving to: 'docker-25.0.3.tgz'

docker-25.0.3.tgz              100%[=====================================================>]  68.25M  18.1MB/s    in 4.6s

2025-03-25 13:26:04 (14.7 MB/s) - 'docker-25.0.3.tgz' saved [71570637/71570637]

[root@ip-172-31-34-187 opt]# ll
total 69896
-rw-r--r--. 1 root root 71570637 Feb 27  2024 docker-25.0.3.tgz
[root@ip-172-31-34-187 opt]#
```

Or, if your server is air-gapped, download the packages offline, copy them to your server, and then install them manually using the yum install command.

Fix 1: Check for Missing Dependencies

1) **Check if Docker is correctly moved to /usr/bin/**

ls -l /usr/bin/docker

2) **If it's missing, move it again:**

sudo mv /opt/docker/* /usr/bin/

3) **Make sure the binary files have execution permissions:**

sudo chmod +x /usr/bin/docker*

4) **Since Docker was installed manually, systemctl won't recognize it yet. Try starting it directly:**

sudo dockerd &

5) **Then, test if Docker is working:**

docker --version
docker run hello-world

6) **Create a Systemd Service for Docker**

If you want to use systemctl to manage Docker, create a systemd service file.

```
sudo tee /etc/systemd/system/docker.service > /dev/null <<EOF
[Unit]
```

```
Description=Docker Application Container Engine
After=network-online.target firewalld.service
Requires=network-online.target

[Service]
ExecStart=/usr/bin/dockerd
Restart=always
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Delegate=yes
KillMode=process

[Install]
WantedBy=multi-user.target
EOF
```

7) Reload Systemd and Enable Docker

```
systemctl daemon-reload
sudo systemctl enable docker
sudo systemctl start docker
```

Or try starting Docker manually and checking errors:

```
sudo dockerd
```

8) Docker requires containerd and runc. Ensure they exist:

```
which containerd
which runc
```

```
[root@ip-172-31-34-187 bin]# which containerd
/bin/containerd
[root@ip-172-31-34-187 bin]# which runc
/bin/runc
[root@ip-172-31-34-187 bin]# cd /var/lib/docker/
```

Fix 2: Verify Docker Directory Permissions

Docker needs access to /var/lib/docker and /run/docker.sock. Fix permissions:

```
sudo mkdir -p /var/lib/docker
sudo chmod 755 /var/lib/docker
sudo chown root:root /var/lib/docker

sudo mkdir -p /run/docker
sudo chmod 777 /run/docker
sudo chown root:docker /run/docker
sudo systemctl restart docker
```

```
[root@ip-172-31-34-187 lib]# ll | grep docker/
[root@ip-172-31-34-187 lib]# ll | grep docker
drwx--x---. 13 root   root     189 Mar 25 13:38 docker
[root@ip-172-31-34-187 lib]# chmod 755 /var/lib/docker
[root@ip-172-31-34-187 lib]# cd /run/docker
[root@ip-172-31-34-187 docker]# ll
total 0
drwx------. 3 root root 160 Mar 25 13:39 containerd
drw-------. 2 root root  60 Mar 25 13:38 libnetwork
srwxr-xr-x. 1 root root   0 Mar 25 13:38 metrics.sock
drwxr-xr-x. 2 root root  40 Mar 25 13:40 netns
drwx------. 2 root root  40 Mar 25 13:38 plugins
drwx------. 3 root root  60 Mar 25 13:39 runtime-runc
drwx------. 2 root root  40 Mar 25 13:38 swarm
[root@ip-172-31-34-187 docker]# cd ..
[root@ip-172-31-34-187 run]# ll
total 36
-rw-------.  1 root   root      0 Mar 24 16:23 agetty.reload
```

Fix 3: Check for Conflicting Services (Podman or Old Docker)

```
[root@ip-172-31-34-187 run]# ps aux | grep dockerd
root       60893  0.0 10.5 1893868 82720 pts/3   Sl   13:37   0:00 dockerd
root       61465  0.0  0.2   6404  2176 pts/3    S+   13:45   0:00 grep --color=auto dockerd
[root@ip-172-31-34-187 run]# kill -9 60893
[root@ip-172-31-34-187 run]# systemctl restart docker
[1]+  Killed                  dockerd  (wd: /usr/bin)
(wd now: /run)
```

```
[root@ip-172-31-34-187 run]# systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/etc/systemd/system/docker.service; enabled; preset: disabled)
     Active: active (running) since Tue 2025-03-25 13:46:12 UTC; 4s ago
   Main PID: 61468 (dockerd)
      Tasks: 15 (limit: 4377)
     Memory: 35.9M
        CPU: 332ms
     CGroup: /system.slice/docker.service
             ├─61468 /usr/bin/dockerd
             └─61474 containerd --config /var/run/docker/containerd/containerd.toml

Mar 25 13:46:12 ip-172-31-34-187.ap-south-1.compute.internal dockerd[61474]: time="2025-03-25T13:46:12.998344078Z" level=info
Mar 25 13:46:12 ip-172-31-34-187.ap-south-1.compute.internal dockerd[61474]: time="2025-03-25T13:46:12.998480984Z" level=info
Mar 25 13:46:12 ip-172-31-34-187.ap-south-1.compute.internal dockerd[61474]: time="2025-03-25T13:46:12.998594700Z" level=info
Mar 25 13:46:14 ip-172-31-34-187.ap-south-1.compute.internal dockerd[61468]: time="2025-03-25T13:46:14.020651395Z" level=info
Mar 25 13:46:14 ip-172-31-34-187.ap-south-1.compute.internal dockerd[61468]: time="2025-03-25T13:46:14.022022221Z" level=info
Mar 25 13:46:14 ip-172-31-34-187.ap-south-1.compute.internal dockerd[61468]: time="2025-03-25T13:46:14.188357041Z" level=info
Mar 25 13:46:14 ip-172-31-34-187.ap-south-1.compute.internal dockerd[61468]: time="2025-03-25T13:46:14.244433450Z" level=info
Mar 25 13:46:14 ip-172-31-34-187.ap-south-1.compute.internal dockerd[61468]: time="2025-03-25T13:46:14.258885959Z" level=info
Mar 25 13:46:14 ip-172-31-34-187.ap-south-1.compute.internal dockerd[61468]: time="2025-03-25T13:46:14.259378465Z" level=info
Mar 25 13:46:14 ip-172-31-34-187.ap-south-1.compute.internal dockerd[61468]: time="2025-03-25T13:46:14.294454300Z" level=info
lines 1-21/21 (END)
client_loop: send disconnect: Connection reset by peer
```

If Podman is installed, it may conflict with Docker. Remove it:

sudo yum remove -y podman # Linux

Check for old Docker processes:

ps aux | grep dockerd

If any are running, kill them:

sudo kill -9 <PID>

Then restart Docker:

sudo systemctl start docker

sudo systemctl restart docker

```
sudo systemctl status docker
systemctl daemon-reload
```

****

Run Locally

1. Ensure Docker is installed.
2. Run docker-compose up --build
3. The frontend should be available on http://localhost:3000, backend on http://localhost:5000.

File Format & Naming

- The file **must** be named Dockerfile (without an extension like .txt or .sh).
- It should be saved in the **root directory** of the respective service (backend or frontend).
- Make sure **all commands are written in uppercase** (e.g., FROM, COPY, RUN, etc.).

1) **GitClone**

sudo git clone https://github.com/Buddywise/todo-application.git /opt/todo-application

```
[root@ip-172-31-34-187 opt]# git clone https://github.com/Buddywise/todo-application.git
Cloning into 'todo-application'...
remote: Enumerating objects: 42, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 42 (delta 3), reused 38 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (42/42), 298.14 KiB | 1.53 MiB/s, done.
Resolving deltas: 100% (3/3), done.
[root@ip-172-31-34-187 opt]#
```

2) **Move into the directory where you want to create the Dockerfile**

cd /path/to/backend

touch Dockerfile

Vi Dockerfile

3) Frontend Dockerfile (frontend/Dockerfile)

```
# Use Node.js as the base image
FROM node:18-alpine

# Set the working directory inside the container
WORKDIR /app

# Install extra dependencies needed for npm build (Alpine fix)
RUN apk add --no-cache python3 g++ make

# Copy package.json and package-lock.json first to leverage Docker caching
COPY package.json package-lock.json ./

# Install dependencies
RUN npm install

# Copy the rest of the frontend app
COPY . .

# Build the frontend
RUN npm run build

# Expose the port the frontend runs on
EXPOSE 3000

# Start the frontend application
CMD ["npm", "start"]
```

Check if the file is created:

ls -l Dockerfile

Build the Docker Image

docker build -t my-frontend.

Run a Container from the Image

docker run -p 5000:5000 my-backend

OR

4) **Below is an example for a Node.js backend.**

```
# Use Python as the base image
FROM python:3.10

# Set the working directory inside the container
WORKDIR /app

# Copy requirements.txt first to leverage Docker caching
COPY requirements.txt .

# Install dependencies
RUN pip install --no-cache-dir -r requirements.txt

# Copy the rest of the application
COPY . .

# Expose the port FastAPI runs on
EXPOSE 8000

# Command to start FastAPI using Uvicorn
CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

5) **Create docker-compose.yml outside the backend and frontend dir**

```
version: "3.8"

services:
  backend:
    build: ./backend
    ports:
      - "8000:8000"
    environment:
```

```
    - NODE_ENV=production
  depends_on: []  # Backend does NOT depend on frontend

frontend:
  build: ./frontend
  ports:
    - "3000:3000"
  environment:
    - NODE_ENV=production
  depends_on:
    - backend  # Frontend starts AFTER backend
```

6) **: Install Docker Compose**

Run the following commands to download and install Docker Compos sudo curl -L

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

Step 2: Give Execution Permissions

sudo chmod +x /usr/local/bin/docker-compose

Verify Installation Location

check if docker-compose exists in /usr/local/bin/:

ls -lah /usr/local/bin/docker-compose

If it exists and has execute permissions (-rwxr-xr-x), then proceed.

docker-compose version

Step 3: check if docker-compose exists in /usr/local/bin/:

ls -lah /usr/local/bin/docker-compose

If it exists and has execute permissions (-rwxr-xr-x), then proceed.

Step 4: Add /usr/local/bin to Your Path

export PATH=$PATH:/usr/local/bin

docker-compose version

```
You can learn more about how to register your system
using rhc at https://red.ht/registration
Last login: Tue Mar 25 15:00:05 2025 from 103.252.216.67
[ec2-user@ip-172-31-34-187 ~]$ ls -lah /usr/local/bin/docker-compose
-rwxr-xr-x. 1 root root 72M Mar 25 15:02 /usr/local/bin/docker-compose
[ec2-user@ip-172-31-34-187 ~]$ export PATH=$PATH:/usr/local/bin
[ec2-user@ip-172-31-34-187 ~]$ docker-compose version
Docker Compose version v2.34.0
[ec2-user@ip-172-31-34-187 ~]$
```

7) **Run the Application with Docker Compose**

Stop and Remove Existing Containers

docker-compose down -v

Rebuild and Restart Everything

docker-compose up --build -d

Rebuild and Restart frontend app

docker-compose up --build frontend

If the above steps are correct but it's still not accessible, check Network ACLs:
Go to AWS Console → VPC → Network ACLs
Find the associated ACL of your subnet
Ensure Inbound Rules allow port 8000 & 3000 for 0.0.0.0/0
If changes were made, restart your instance

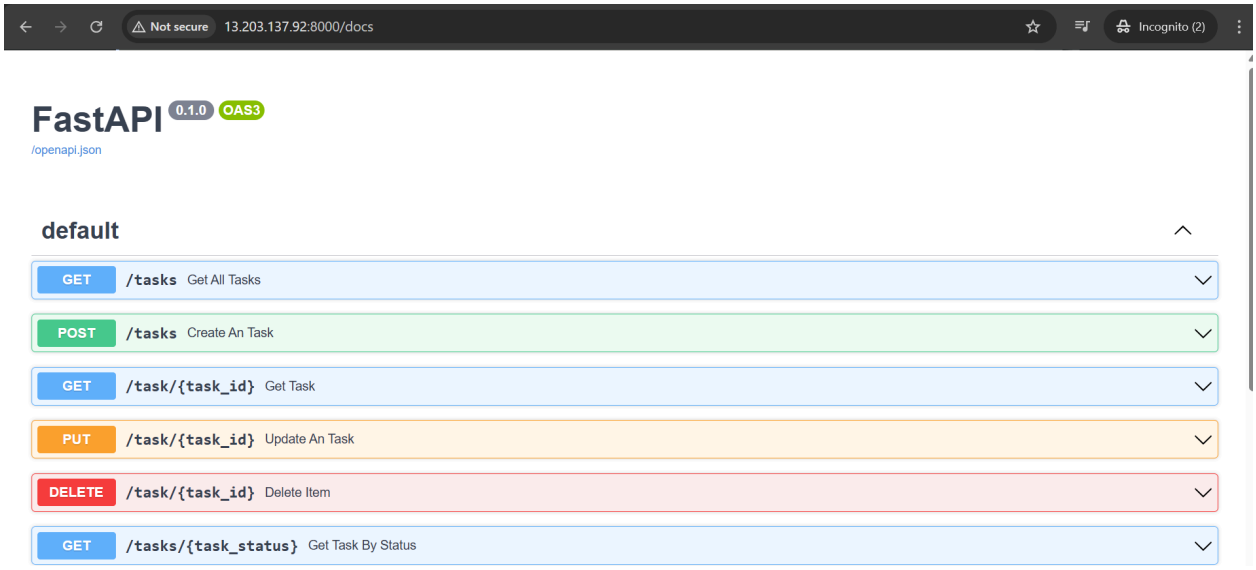
```
[root@ip-172-31-34-187 ~]# cd /local-deployment/todo-application/
[root@ip-172-31-34-187 todo-application]# ll
total 8
drwxr-xr-x. 3 root root 172 Apr 18 17:37 backend
-rw-r--r--. 1 root root 374 Apr 18 17:37 docker-compose.yml
drwxr-xr-x. 3 root root  63 Apr 18 17:22 frontend
-rw-r--r--. 1 root root 183 Apr 18 17:13 README.md
[root@ip-172-31-34-187 todo-application]# docker-compose down -v

WARN[0019] /local-deployment/todo-application/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, ple
ase remove it to avoid potential confusion

[+] Running 3/3
 ✔ Container todo-application-frontend-1  Removed                                                                    10.7s
 ✔ Container todo-application-backend-1   Removed                                                                     0.6s
 ✔ Network todo-application_default       Removed                                                                     0.1s
[root@ip-172-31-34-187 todo-application]#
[root@ip-172-31-34-187 todo-application]#
[root@ip-172-31-34-187 todo-application]# docker-compose up --build -d
```

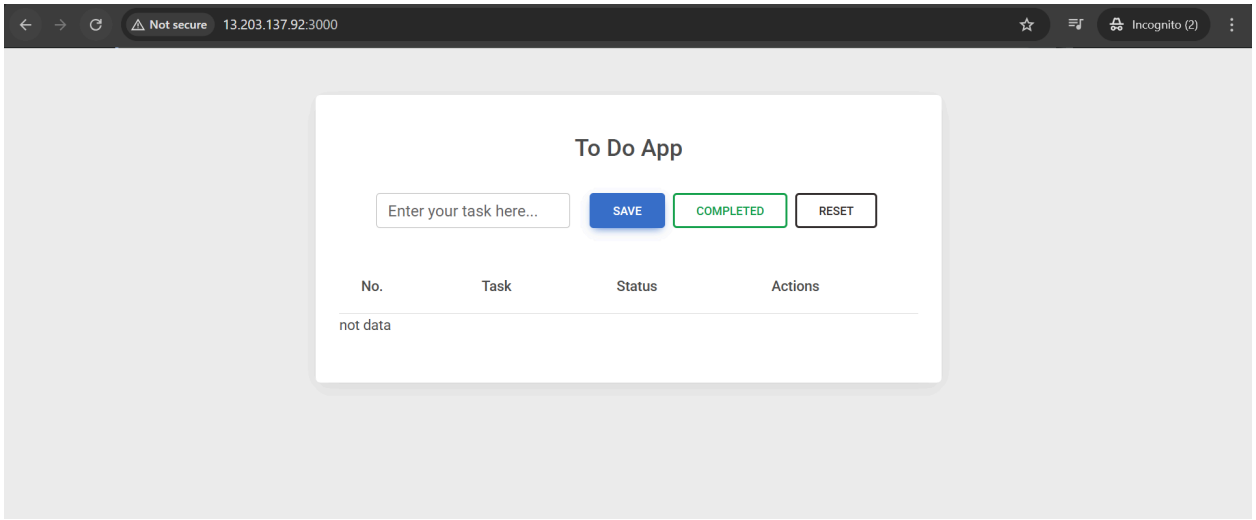
```
 => CACHED [backend 4/5] RUN pip install --no-cache-dir -r requirements.txt                                          0.0s
 => CACHED [backend 5/5] COPY . .                                                                                    0.0s
 => [backend] exporting to image                                                                                     0.0s
 => => exporting layers                                                                                              0.0s
 => => writing image sha256:02c50a02c7baf0ce298f80b342bb512d2a020af53748ff913e3ee705e7b85886                         0.0s
 => => naming to docker.io/library/todo-application-backend                                                          0.0s
 => [backend] resolving provenance for metadata file                                                                 0.0s
 => [frontend internal] load build definition from Dockerfile                                                        0.0s
 => => transferring dockerfile: 583B                                                                                 0.0s
 => [frontend internal] load metadata for docker.io/library/node:18-alpine                                           1.0s
 => [frontend internal] load .dockerignore                                                                           0.0s
 => => transferring context: 2B                                                                                      0.0s
 => [frontend 1/7] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4  0.0s
 => [frontend internal] load build context                                                                           0.0s
 => => transferring context: 1.44kB                                                                                  0.0s
 => CACHED [frontend 2/7] WORKDIR /app                                                                               0.0s
 => CACHED [frontend 3/7] RUN apk add --no-cache python3 g++ make                                                    0.0s
 => CACHED [frontend 4/7] COPY package.json package-lock.json ./                                                     0.0s
 => CACHED [frontend 5/7] RUN npm install                                                                            0.0s
 => CACHED [frontend 6/7] COPY . .                                                                                   0.0s
 => CACHED [frontend 7/7] RUN npm run build                                                                          0.0s
 => [frontend] exporting to image                                                                                    0.0s
 => => exporting layers                                                                                              0.0s
 => => writing image sha256:d02c65b7f40f0e7c2b39dde171d42a97ab490c70cf20d726c6d1ee3177571387                         0.0s
 => => naming to docker.io/library/todo-application-frontend                                                         0.0s
 => [frontend] resolving provenance for metadata file                                                                0.0s
[+] Running 5/5
 ✔ backend                              Built                                                                        0.0s
 ✔ frontend                             Built                                                                        0.0s
 ✔ Network todo-application_default     Created                                                                      0.1s
 ✔ Container todo-application-backend-1   Started                                                                    0.4s
 ✔ Container todo-application-frontend-1  Started                                                                    0.8s
[root@ip-172-31-34-187 todo-application]#
```

Test Access from EC2 Instance Itself

curl http://localhost:8000/docs

curl http://localhost:3000


Not secure
13.203.137.92:8000/docs
Incognito (2)
FastAPI
0.1.0
OAS3
/openapi.json
default
GET /tasks Get All Tasks
POST /tasks Create An Task
GET /task/{task_id} Get Task
PUT /task/{task_id} Update An Task
DELETE /task/{task_id} Delete Item
GET /tasks/{task_status} Get Task By Status


Not secure
13.203.137.92:3000
Incognito (2)
To Do App
Enter your task here...
SAVE
COMPLETED
RESET
No.
Task
Status
Actions
not data