Steps to Deploy FastAPI App on EC2

1 Create an AWS Account

- Go to the AWS website and sign up for an account.
- Enter your contact information, billing details, and payment method. AWS offers a free tier that allows limited usage of many of its services at no cost.
- Verify your identity with a phone number.

2 Launch an EC2 Instance

- **Log in to AWS Management Console:** Go to the AWS Management Console and choose **EC2** from the Services menu.
- **Launch Instance:** In the EC2 Dashboard, click **Launch Instance** to create a new EC2 instance.
- Choose an Amazon Machine Image (AMI), such as **Ubuntu Server 20.04 LTS** or another suitable Linux distribution.
- Select an instance type, for example, `t2.micro` (which is eligible for the AWS Free Tier).
- **Configure Instance:** Set up the instance details (default options are generally fine). Add storage if needed (the default is usually fine).
- **Configure Security Group:** Set up a security group to allow traffic on SSH (port 22) and HTTP (port 8080). You can add more ports later as needed.
- **Review and Launch:** Review your instance configuration and click **Launch**.
- Create or select an existing key pair for SSH access. If you create a new one, download the `.pem` file, as you'll need it to access the EC2 instance later.

3 SSH into the EC2 Instance

- **Access EC2 Instance:** Use SSH to connect to your EC2 instance:

```
chmod 400 path_to_your_key.pem
ssh -i path_to_your_key.pem ubuntu@<your-ec2-public-ip>
(for windows, we copy ssh to home directory using mv /mnt/c/Users/USER/Documents/10
chmod 400 ~/fastapi-key-pair.pem
and ssh -i ~/fastapi-key-pair.pem ubuntu@3.143.204.49
```

4 Set Up the EC2 Instance

- **Update the Instance:** Run the following commands to update your EC2 instance:

```
sudo apt update && sudo apt upgrade -y
```

- **Install Necessary Dependencies:** Install Docker, Docker Compose, and Python packages:

```
sudo apt install python3 python3-pip python3-venv docker.io docker-compose -y
```

5 Transfer Your Application Files to EC2

- clone your complete project from github and cd into the project OR
- **Create a Directory for Your App:** On the EC2 instance, create a directory for your FastAPI project:

```
mkdir ~/mi_fatality_prediction
cd ~/mi_fatality_prediction
```

- **Transfer Files:** Use `scp` (secure copy) to transfer your application files (such as `main.py`, `Dockerfile`, `docker-compose.yml`, and model files) from your local machine to the EC2 instance:

```
scp -i path_to_your_key.pem main.py Dockerfile docker-compose.yml <your-ec2-public-
```

6 Create Docker Environment

- **Dockerfile:** Create a `Dockerfile` in your project directory to containerize your FastAPI application. Example `Dockerfile`:

```
FROM python:3.9-slim

WORKDIR /app

COPY . /app

RUN pip install --no-cache-dir -r requirements.txt

CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8080"]
```

- **docker-compose.yml:** Create a `docker-compose.yml` file to manage the app container. Example `docker-compose.yml`:

```
version: '3.8'
services:
  fastapi_app:
    build: .
    ports:
      - "8080:8080"
```

7 Prepare Python Requirements

- **Create a `requirements.txt`:** Inside the project directory, create a `requirements.txt` file that includes all necessary Python packages:

```
fastapi
uvicorn
joblib
```

8 Build and Run the Docker Container

- **Build the Docker Container:** On the EC2 instance, build the Docker image and run the app using Docker Compose:

```
sudo docker-compose up --build -d
```

9 Check Application Logs

- **Check Logs:** To verify that everything is running correctly, check the logs of the FastAPI container:

```
sudo docker-compose logs -f
```

10 Open Ports on EC2 Security Group

- **Configure Security Group:** Go to the **EC2 Dashboard** in the AWS console and navigate to **Security Groups**.
- Select the security group attached to your EC2 instance.
- Add an inbound rule for **HTTP** on port `8080` to allow external access to the application.

11 Test the FastAPI Application

- **Test with curl:** On the EC2 instance or from your local machine, test the FastAPI endpoint using `curl`:

```
curl -X POST http://<your-ec2-public-ip>:8080/predict \
  -H "Content-Type: application/json" \
  -d '[{ "AGE": 0, "SEX": 0, "INF_ANAM": 0, ... }]'
```

- **Verify the Output:** Ensure that the application returns a prediction or response as expected.

12 Access Application from Browser

- **Access the Application:** You should now be able to access the application by going to `http://<your-ec2-public-ip>:8080/predict` from any web browser or API testing tool like **Postman**.

13 Finalize and Document

- **Document the Deployment:** Document the steps you've followed to deploy your FastAPI application, including:
  - How to configure EC2 and set up security groups.
  - How to build and deploy your Dockerized FastAPI application.
  - How to test the application using `curl` or **Postman**.