Deploying the “Ask for a Date” App on Docker Container with Nginx on EC2

Prerequisites

Before you begin, make sure you have the following:

1. ***AWS Account:*** *You should have an AWS account set up.*
2. ***EC2 Instance:*** *A running EC2 instance (preferably with Ubuntu or another Linux-based distribution).*
3. ***SSH Key Pair:*** *SSH key pair is used to access the EC2 instance.*
4. ***Domain Name*** (optional)*: If you're planning to use a custom domain, configure it beforehand.*
5. ***Docker and Docker Compose*** *are installed on your local machine for local testing (optional but recommended).*

Step 1: Launch an EC2 Instance

1. ***Log in to AWS Console:***
   - *Go to the [AWS Management Console](#)*.
   - *In the search bar, type “EC2” and select EC2 to open the EC2 Dashboard.*
2. ***Create a New Instance:***
   - *Click on* ***Launch Instance.***
   - *Select an* ***AMI (Amazon Machine Image).*** *Choose an Ubuntu Server image (e.g., Ubuntu 20.04 LTS).*
   - *Select an instance type based on your requirements (e.g., t2.micro for small projects).*
   - *Configure instance details and select your security group. Ensure that ports* ***22 (SSH), 80 (HTTP)****, and* ***443 (HTTPS)*** *are open in your security group so you can access the server and web traffic.*
   - *Select the key pair you want to use for SSH access.*

3. **Launch the Instance:**
    - Review your instance configuration, and then click **Launch**.
4. **Access the EC2 Instance:**
    - Once the instance runs, click on the instance ID to access its details.
    - Copy the **Public IP** of your EC2 instance.

Use SSH to access the instance:
bash
CopyEdit

```
ssh -i /path/to/your-key.pem ubuntu@your-ec2-public-ip
```

Step 2: Install Docker on EC2 Instance

1. **Update the Package Index:**

Run the following command to update the package index on your EC2 instance:
bash
CopyEdit

```
sudo apt-get update
```

2. **Install Docker:**

Install Docker by running the following commands:
bash
CopyEdit

```
sudo apt-get install -y docker.io
```

3. **Start Docker:**

Start the Docker service:
bash
CopyEdit

```
sudo systemctl start docker
```

4. **Enable Docker to Start on Boot:**

Run the following command to enable Docker to start on boot:
bash
CopyEdit

```
sudo systemctl enable docker
```

5. **Verify Docker Installation:**

Check the Docker version to verify the installation:
bash
CopyEdit

```
docker --version
```

Step 3: Install Docker Compose

Docker Compose allows you to define and run multi-container Docker applications.

1. **Download Docker Compose:**

Download the Docker Compose binary by running:
bash
CopyEdit

```
sudo curl -L
```

```
"https://github.com/docker/compose/releases/download/1.2
9.2/docker-compose-$(uname -s)-$(uname -m)" -o
/usr/local/bin/docker-compose
```

2. ***Apply Executable Permissions:***

Give the necessary permissions to Docker Compose:
bash
CopyEdit

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

3. ***Verify Docker Compose Installation:***

Check the Docker Compose version:
bash
CopyEdit

```
docker-compose --version
```

Step 4: Set Up the Application and Dockerize it

1. ***Clone the "Ask for a Date" App:***

Clone the repository of the app to your EC2 instance (replace with the actual GitHub repository URL):
bash
CopyEdit

```
git clone
https://github.com/your-username/ask-for-a-date.git
cd ask-for-a-date
```

2. ***Dockerfile Configuration:***

Ensure the application’s **Dockerfile** is properly configured to build the app. The Dockerfile should define the app environment and necessary dependencies. Here is a sample `Dockerfile`:

dockerfile

CopyEdit

```
FROM node:14
WORKDIR /app
COPY package.json .
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
```

3. **Docker Compose Configuration:**

Create a **docker-compose.yml** file to define the application services. A basic Docker Compose file for this setup might look like this:

yaml

CopyEdit

```
version: '3'
services:
  web:
    build: .
    ports:
      - "3000:3000"
    networks:
      - app-network
networks:
  app-network:
    driver: bridge
```

Step 5: Build and Run the Docker Container

1. ***Build the Docker Image:***

Run the following command to build the Docker image:
bash
CopyEdit

```
sudo docker-compose build
```

2. ***Start the Docker Container:***

Start the application with the following command:
bash
CopyEdit

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

3. ***Verify the Application:***
   - *Check if the app is running by accessing the EC2 instance's public IP with the appropriate port (e.g.,* `http://your-ec2-public-ip:3000`***).***

Step 6: Install and Configure Nginx as a Reverse Proxy

1. ***Install Nginx:***

Install Nginx on the EC2 instance:
bash

CopyEdit

```
sudo apt-get install -y nginx
```

2. ***Configure Nginx as a Reverse Proxy:***

Edit the Nginx configuration to act as a reverse proxy for the app.
Open the default Nginx configuration file:
bash
CopyEdit

```
sudo nano /etc/nginx/sites-available/default
```

Modify the `server` *block to include:*
nginx
CopyEdit

```
server {
    listen 80;

    server_name your-domain.com;  # Use your domain or
EC2 public IP

    location / {
        proxy_pass http://localhost:3000;  # Forward
traffic to Docker container
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

3. **Restart Nginx:**

Restart Nginx to apply the changes:
bash
CopyEdit

```
sudo systemctl restart nginx
```

Step 7: Test the Application

1. **Access the Application:**
   - Open a browser and go to `http://your-ec2-public-ip` or your domain name (if configured). You should see the "Ask for a Date" application running.
2. **Troubleshooting:**

If there are any issues, check the logs of the Docker containers and Nginx:
bash
CopyEdit

```
sudo docker-compose logs
sudo tail -f /var/log/nginx/error.log
```