

TravelMemory Deployment Guide

Steps to Deploy TravelMemory on EC2

1. Launch EC2 Instance

- Log in to AWS Console and go to EC2 Dashboard.
- Click Launch Instance:
 - * AMI: Ubuntu 22.04.
 - * Instance Type: t2.micro.
 - * Security Group: Allow ports 3000, 3001, and 22.
- Launch the instance and note the public IP.
- SSH into your instance:

```
ssh -i <your-key-file.pem> ubuntu@<PUBLIC_IP>
```

2. Install Docker and Git

- Update and install Docker and Git:

```
sudo apt update  
sudo apt install docker.io git -y  
sudo systemctl start docker  
sudo systemctl enable docker
```

3. Clone the Repository

- Clone your repository and navigate to the project folder:

```
git clone https://github.com/ankitalodha05/TravelMemory.git  
cd TravelMemory
```

TravelMemory Deployment Guide

- Verify configurations:

- * Backend .env file -> Port should be 3001.
- * Backend index.js -> Replace 'localhost' with '<PUBLIC_IP>:3001'.
- * Frontend -> src -> url.js -> Replace with '<PUBLIC_IP>:3001'.

4. Create Docker Network

- Create a Docker network for backend and frontend communication:

```
docker network create travelmemory-network
```

5. Build and Run Backend

- Navigate to the backend directory:

```
cd backend
```

- Build the backend image:

```
docker build -t travelmemory-backend .
```

- Run the backend container on the network:

```
docker run -d --name travelmemory-backend \  
  --network travelmemory-network \  
  -p 3001:3001 travelmemory-backend
```

- Verify the backend:

```
http://<PUBLIC_IP>:3001/hello
```

TravelMemory Deployment Guide

6. Build and Run Frontend

- Navigate to the frontend directory:

```
cd ../frontend
```

- Create or update url.js:

```
echo "const BACKEND_URL = 'http://<PUBLIC_IP>:3001'; export default BACKEND_URL;" >
src/url.js
```

- Build the frontend image:

```
docker build -t travelmemory-frontend .
```

- Run the frontend container on the network:

```
docker run -it -d --name travelmemory-frontend \
  --network travelmemory-network \
  -p 3000:3000 \
  -e REACT_APP_BACKEND_URL=http://<PUBLIC_IP>:3001 \
  travelmemory-frontend
```

7. Test Application

- Backend: Test with curl or browser:

```
curl http://<PUBLIC_IP>:3001/hello
```

- Frontend: Open in browser:

TravelMemory Deployment Guide

`http://<PUBLIC_IP>:3000`

8. Optional: Stop and Cleanup

- Stop and remove containers:

```
docker stop <container_id>
```

```
docker rm <container_id>
```

```
docker network rm travelmemory-network
```

Final URLs:

- Frontend: `http://<PUBLIC_IP>:3000`
- Backend: `http://<PUBLIC_IP>:3001`

TravelMemory Deployment Guide

Backend Dockerfile

```
# Use the official Node.js base image
FROM node:16

# Set the working directory inside the container
WORKDIR /app

# Copy package.json and package-lock.json first for better caching
COPY package.json package-lock.json ./

# Install backend dependencies
RUN npm install

# Copy the rest of the application files
COPY . .

# Expose the backend port
EXPOSE 3001

# Start the application
CMD ["node", "index.js"]
```

Frontend Dockerfile

```
# Use the official Node.js base image
FROM node:16

# Set the working directory inside the container
WORKDIR /app

# Copy package.json and package-lock.json first for better caching
```

TravelMemory Deployment Guide

```
COPY package.json package-lock.json ./
```

```
# Install frontend dependencies
```

```
RUN npm install
```

```
# Copy the rest of the application files
```

```
COPY . .
```

```
# Build the frontend for production
```

```
RUN npm run build
```

```
# Install 'serve' to serve the built files
```

```
RUN npm install -g serve
```

```
# Expose the frontend port
```

```
EXPOSE 3000
```

```
# Serve the built frontend
```

```
CMD ["serve", "-s", "build", "-l", "3000"]
```

TravelMemory Deployment Guide

Steps to Deploy TravelMemory on EC2

1. Launch EC2 Instance

- Log in to AWS Console and go to EC2 Dashboard.
- Click Launch Instance:
 - * AMI: Ubuntu 22.04.
 - * Instance Type: t2.micro.
 - * Security Group: Allow ports 3000, 3001, and 22.
- Launch the instance and note the public IP.
- SSH into your instance:

```
ssh -i <your-key-file.pem> ubuntu@<PUBLIC_IP>
```

2. Install Docker and Git

- Update and install Docker and Git:

```
sudo apt update
```



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



```
sudo systemctl start docker
```



```
sudo systemctl enable docker
```

3. Clone the Repository

- Clone your repository and navigate to the project folder:

```
git clone https://github.com/ankitalodha05/TravelMemory.git
```

```
cd TravelMemory
```

- Verify configurations:

- * Backend .env file -> Port should be 3001.

- * Backend index.js -> Replace 'localhost' with '<PUBLIC_IP>:3001'.

- * Frontend -> src -> url.js -> Replace with '<PUBLIC_IP>:3001'.

4. Create Docker Network

- Create a Docker network for backend and frontend communication:

```
docker network create my_network
```

5. Build and Run Backend

- Navigate to the backend directory:

```
cd backend
```

- Build the backend image:

```
sudo docker build . -t backend:latest
```

- Run the backend container on the network:

```
sudo docker run -it -d --name backend_container --network my_network -p 3001:3001 backend:latest
```

- Verify the backend:

`http://<PUBLIC_IP>:3001/hello`

6. Build and Run Frontend

- Navigate to the frontend directory:

```
cd ../frontend
```

- Create or update url.js:

```
echo "const BACKEND_URL = 'http://<PUBLIC_IP>:3001'; export default BACKEND_URL;" > src/url.js
```

- Build the frontend image:

```
sudo docker build . -t frontend:latest
```

- Run the frontend container on the network:

```
sudo docker run -it -d --name frontend_container --network my_network -p 3000:3000 frontend:latest
```

7. Test Application

- Backend: Test with curl or browser:

```
curl http://<PUBLIC_IP>:3001/hello
```

- Frontend: Open in browser:

```
http://<PUBLIC_IP>:3000
```

8. Optional: Stop and Cleanup

- Stop and remove containers:

```
docker stop <container_id>
```

```
docker rm <container_id>
```

```
docker network rm my_network
```

Final URLs:

- Frontend: http://<PUBLIC_IP>:3000

- Backend: http://<PUBLIC_IP>:3001

Updated Backend Dockerfile

```
FROM ubuntu:latest
```

```
WORKDIR /app
```

```
COPY . /app
```

```
RUN apt-get update && apt-get install -y nodejs npm git && npm install
```

```
EXPOSE 3001
```

```
CMD ["node", "index.js"]
```

Updated Frontend Dockerfile

```
FROM ubuntu:latest
```

```
WORKDIR /app
```

```
COPY . /app
```

```
RUN apt-get update && apt-get install -y npm && npm install
```

```
EXPOSE 3000
```

```
CMD ["npm", "start"]
```