## **Travel Memory Deployment Guide**

### Prerequisites:

* AWS Account
* Domain Name
* Cloudflare Account

**Step 1: Launch an EC2 Instance**

1. Log in to the AWS Management Console.
2. Navigate to EC2 and click on "Launch Instance."
3. Choose an Amazon Machine Image (AMI) (I chose Ubuntu).
4. Select an instance type (e.g., t2. micro for free tier).
5. Configure instance details (default settings are fine).
6. Add storage (default settings are fine).
7. Add tags (optional).
8. Configure security group:
   * Add a rule for HTTP (port 80).
   * Add a rule for SSH (port 22) to your IP address.
9. Review and launch the instance
10. Download the key pair and keep it safe.

**Step 2: SSH into Your EC2 Instance**

1. Open your terminal.
2. Change permissions of the key pair file:

chmod 400 traval-memory.pem

1. Connect to your instance:

ssh -i "traval-memory.pem" ec2-user@public-ip

**Step 3: Set Up the Backend**

Install Node.js and Git:

sudo apt-get update -y

curl -sL https://rpm.nodesource.com/setup\_14.x

| sudo bash -

sudo apt install -y nodejs git

Clone the repository and navigate to the backend directory:

Git clone https://github.com/manishsalona/TravelMemory.git

cd TravelMemory/backend

Install backend dependencies:

npm install

Create and update the .env file with the necessary environment variables:

vim .env

PORT=3000 MONGO\_URL=”mongodb+srv://admin:admin@cluster0.d3ea5nw.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0/travalmemory”

Install and configure Nginx:

sudo apt-get install nginx

sudo systemctl start nginx

sudo systemctl enable nginx

Configure Nginx as a reverse proxy:

sudo vim /etc/nginx/nginx.conf

Add the following within the http block:

server {

listen 80;

server\_name \_;

location / {

proxy\_pass http://public\_ip:3000;

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;

}

}

Restart Nginx:

sudo systemctl restart nginx

Start the backend server:

npm start

**Step 4: Set Up the Frontend:**

Navigate to the frontend directory:

cd ../frontend

Update urls.js to point to the backend:

Install frontend dependencies and build the frontend:

npm install

npm run build

Serve the frontend using Nginx:

sudo mv build /usr/share/nginx/html

**Step 5: Scaling the Application**

**Create an AMI of your configured instance:**

Go to the EC2 dashboard.

Select your instance, click on "Actions," then "Create Image."

Launch multiple instances using the AMI:

Launch new instances as described in Step 1.

Configure a Load Balancer:

Navigate to EC2 > Load Balancers > Create Load Balancer.

Choose an Application Load Balancer.

Configure the load balancer and add instances.