**TravelMemory Application Deployment Using MERN Stack**

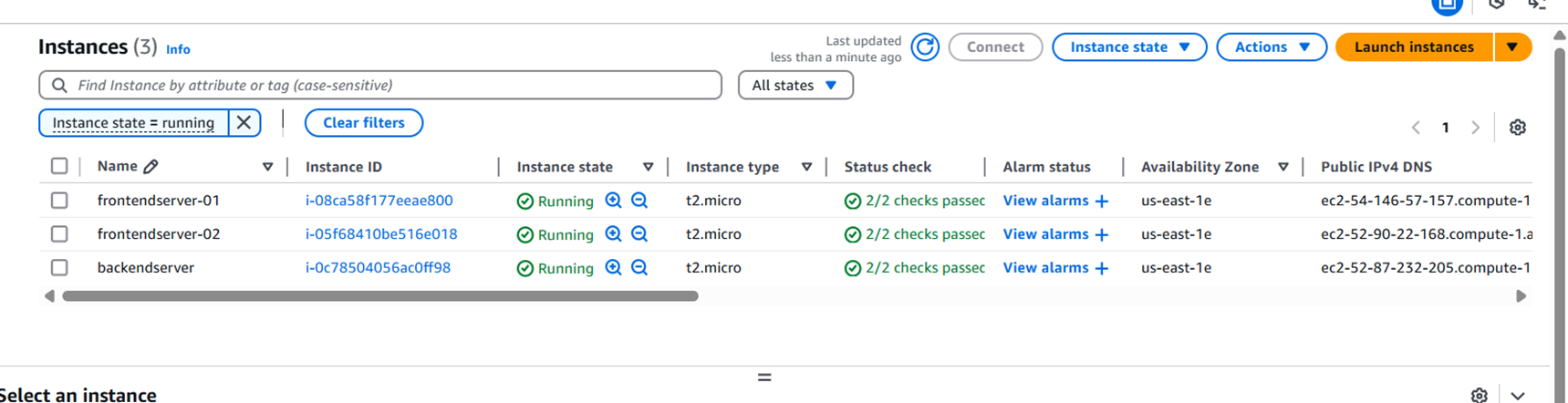
**Objective;-**

* Deploy the TravelMemory application using the MERN stack on AWS.
* Configure backend and frontend EC2 instances.
* Set up a secure, scalable, and high-performance architecture with a load balancer and Cloudflare integration.

**Step 1: Create EC2 Instances**

Create three EC2 instances on AWS with Ubuntu as the operating system:

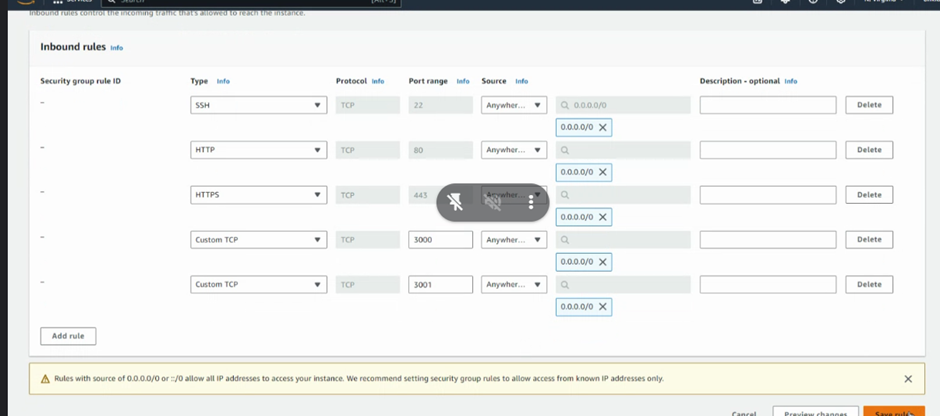
1. **Backend Server**: backendserver
2. **Frontend Servers**: frontendserver-01, frontendserver-02



**Security Group Configuration:-**

Configure the security group to allow the following ports:

* **SSH (Port 22):** For remote server management.
* **HTTP (Port 80):** For web traffic.
* **HTTPS (Port 443):** For secure web traffic.
* **Port 3000:** For the backend server.
* **Port 3001:** For frontend servers.



**Step 2: Backend Server Setup**

***Install Necessary Tools***

*Run the following commands to install required software:*

* sudo apt update -y
* sudo apt i install nginx
* sudo apt install git
* sudo apt install nodejs
* sudo apt install npm

Here are the commands to install and configure Nginx on your Ubuntu EC2 instance:

***Step 1: Install Nginx***

Run the following commands on the backend server to install and configure Nginx:

1. Update Package List: **sudo apt update**
2. Install Nginx: **sudo apt install -y nginx**
3. Start and Enable Nginx Service: **sudo systemctl start nginx**

**sudo systemctl enable nginx**

1. Check Nginx Status: **sudo systemctl status nginx**

Ensure it shows active (running).

***Step 2: Configure Nginx***

1. Edit the Default Nginx Configuration File:

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

2 Replace the File Content with the Following Configuration:

server {

listen 80;

server\_name <your-ec2-public-ip>;(35.168.1.127)

location / {

proxy\_pass http://127.0.0.1:3000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

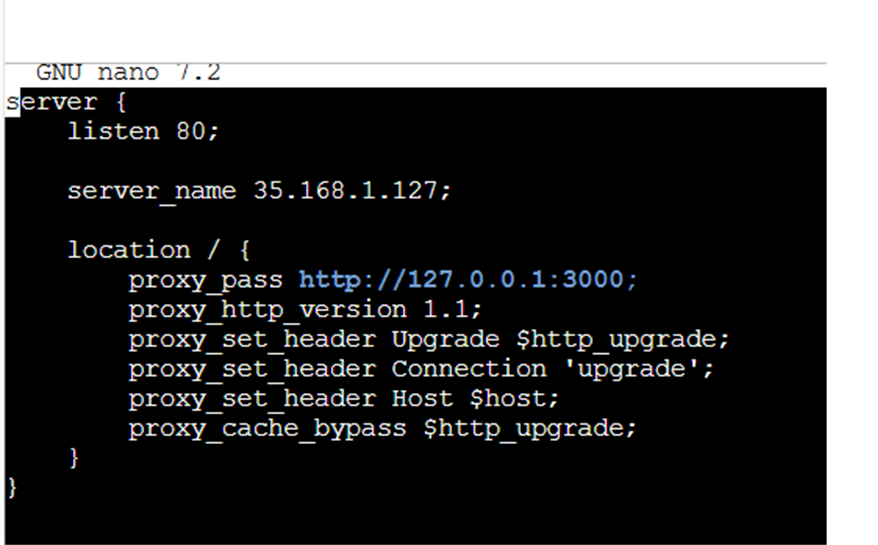
proxy\_cache\_bypass $http\_upgrade;

}

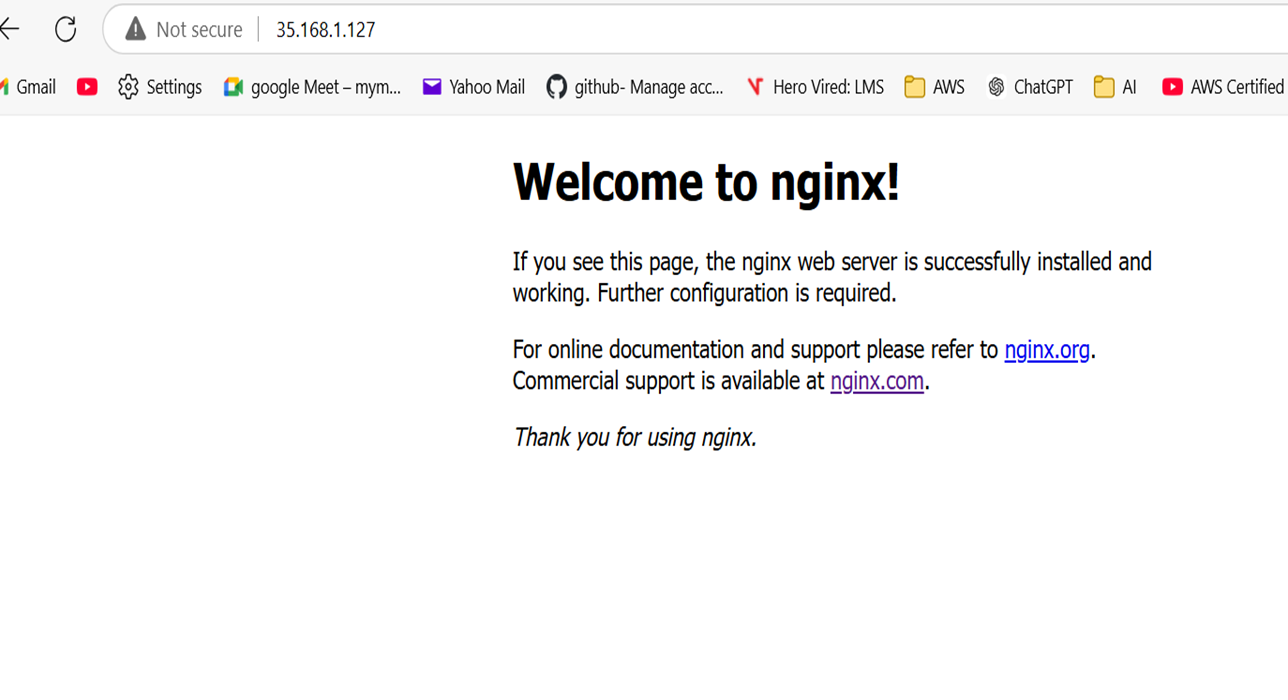
}

Replace <your-ec2-public-ip> with the backend server's public IP address.

(35.168.1.127)



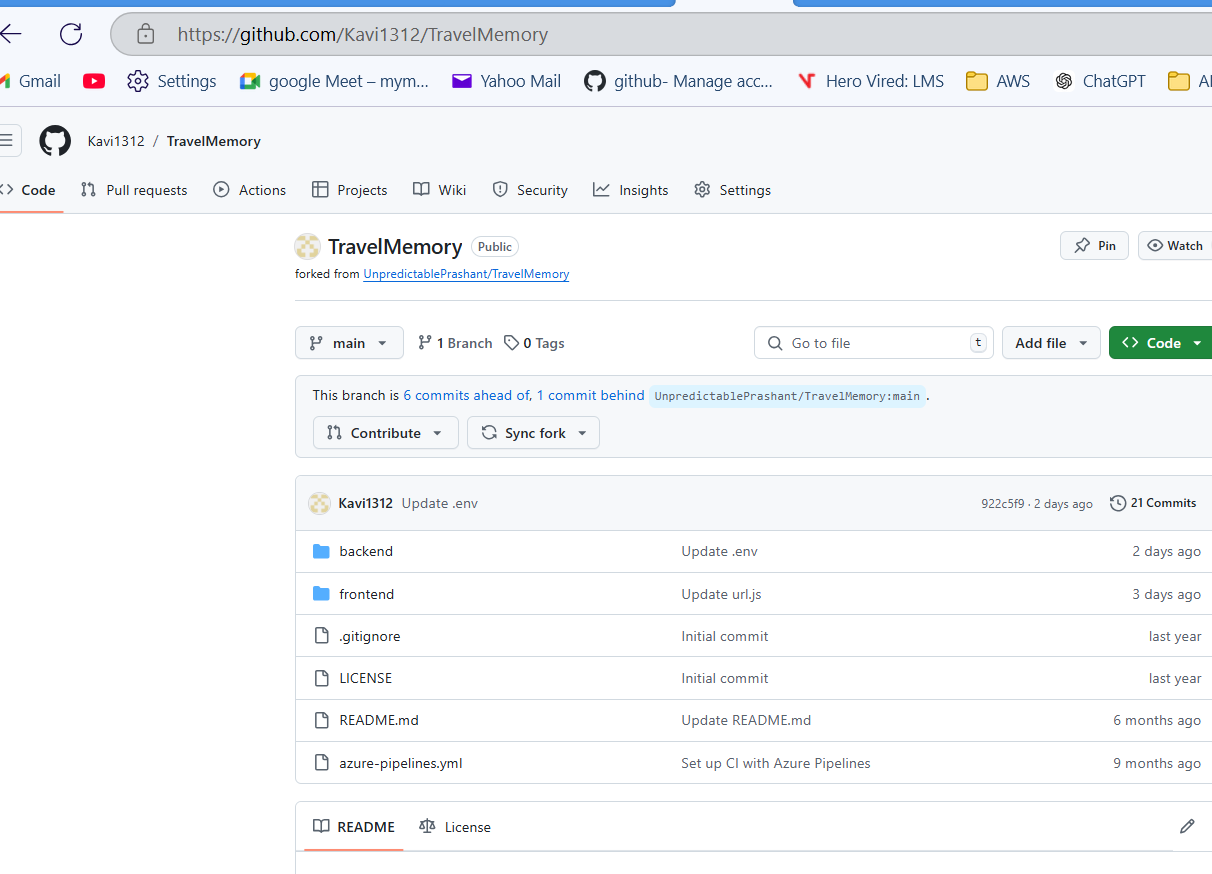
1. **Test the Nginx Configuration: sudo nginx -t**
2. **Restart Nginx to Apply Changes : sudo systemctl restart nginx**
3. **Verify Nginx:** Open your backend EC2 public IP in a browser. You should see the Nginx default page.



***Step 3: Proceed to Clone the Repository***

After installing and configuring Nginx, you can now proceed with the next steps to clone the repository and set up the backend application.

1. **Clone the Backend Repository: sudo git clone**<https://github.com/Kavi1312/TravelMemory.git>



1. **Navigate to folder :**  **cd TravelMemory/backend**
2. **Install Required Packages for the Backend: sudo apt install -y nodejs npm**

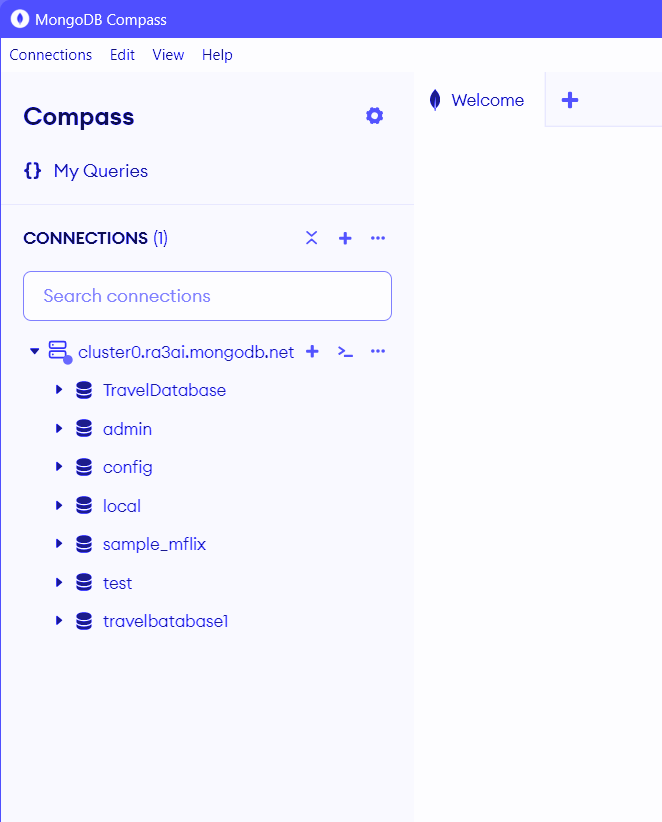
**sudo npm install**

1. **Configure the Backend Environment Variables: sudo nano .env**

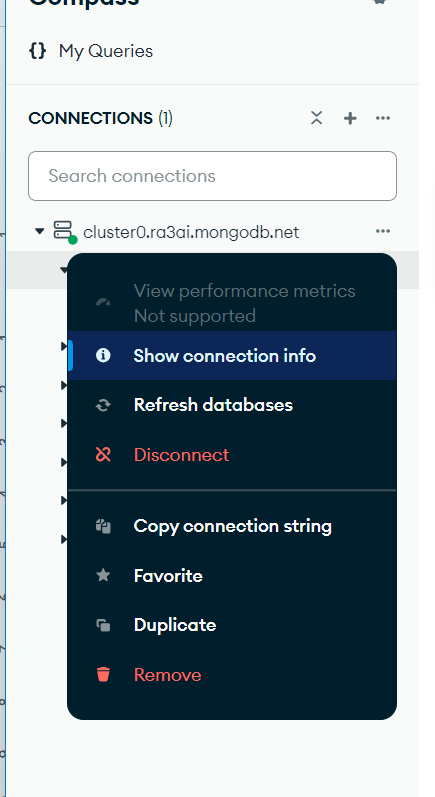
Add the following content: makefile

| PORT=3000  MONGO\_URI="mongodb+srv://vika2k:5j4tyZQSnucMiK1h@cluster0.b7mwy.mongodb.net/TravelDatabase" |
| --- |

(your MonogoDb username password and connecting string details and your database name)

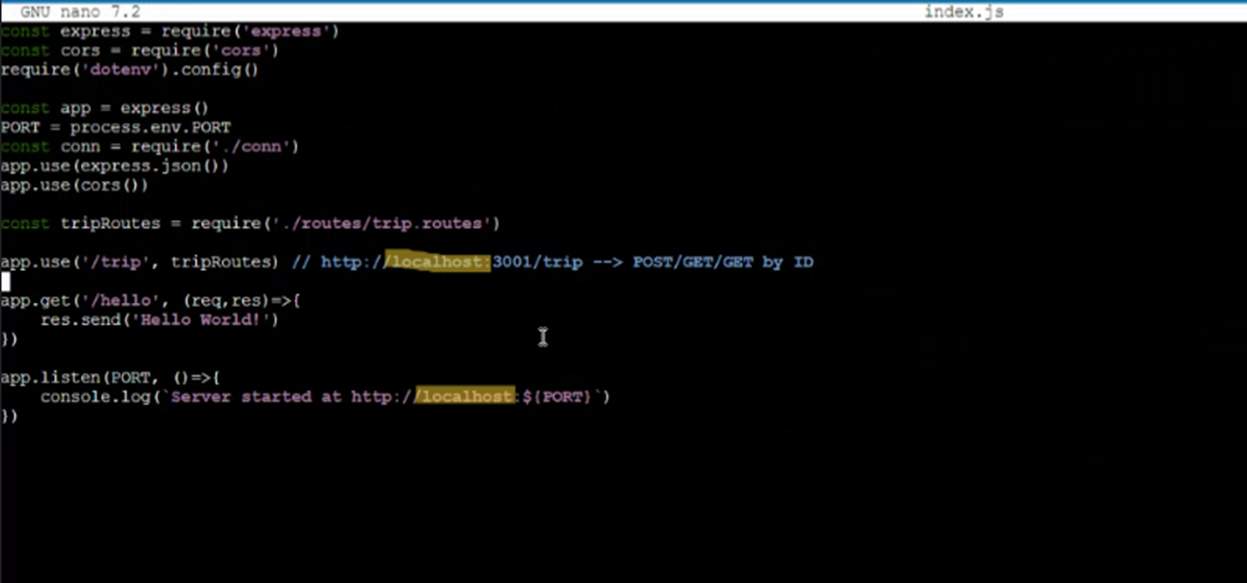


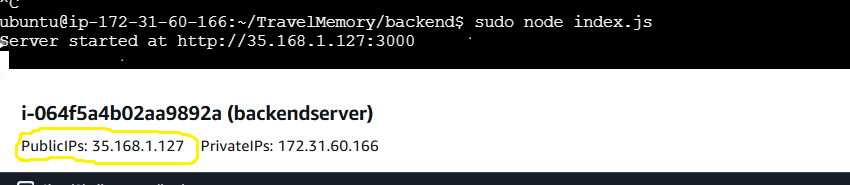
* *Copy connection string details from MongoDb Compass ;*
* *Click cluster name 3 dots and select connection details.*
* *Use this Link to create cluster by login :-* [*Pro*ject Overview | Cloud: MongoDB Cloud](https://cloud.mongodb.com/v2/671e687afea2cc610791efc5#/overview)

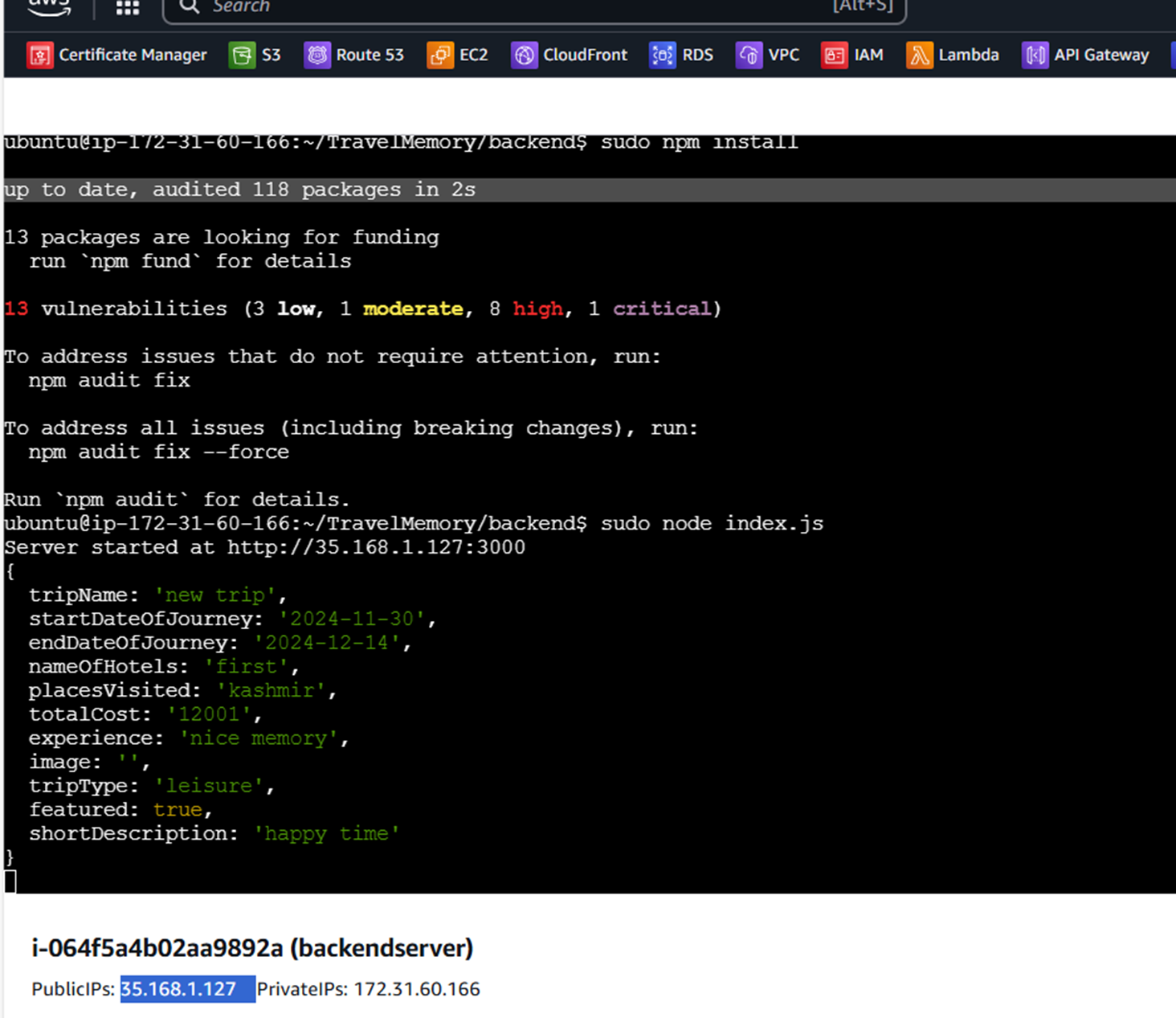


**Start the Backend Application (Optional for Testing): sudo node index.js**

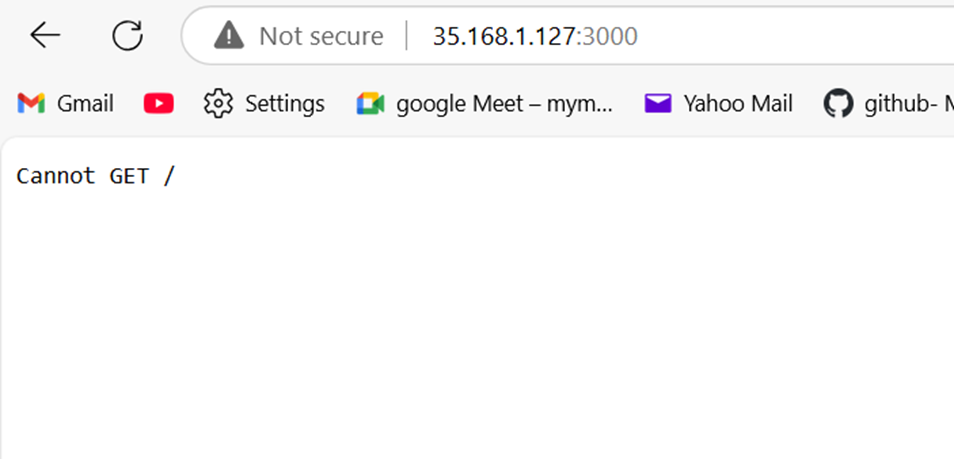
**Modify index.js to Replace localhost with Backend IP Address: sudo nano index.js**

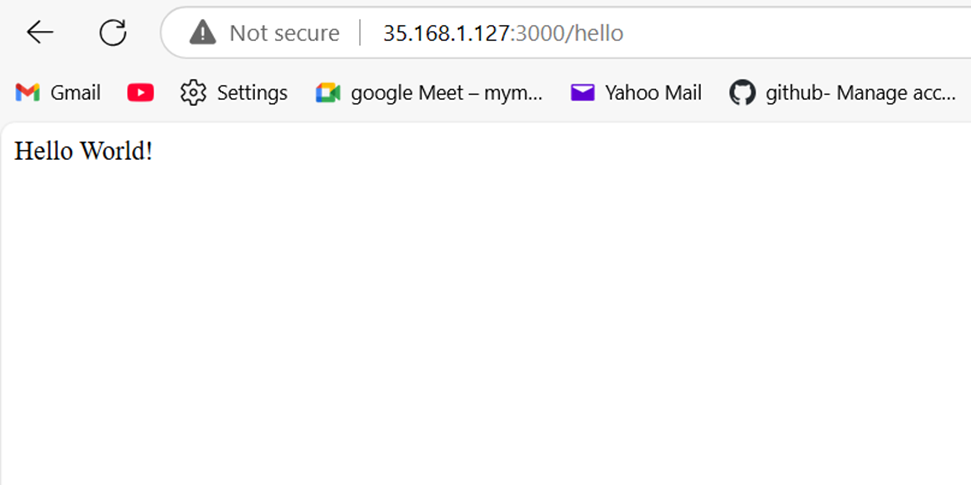






Update the host with the backend EC2 public IP. ( check with or without port and / hello)





***Ensured Backend is working fine***