**Travel Memory Application Deployment**

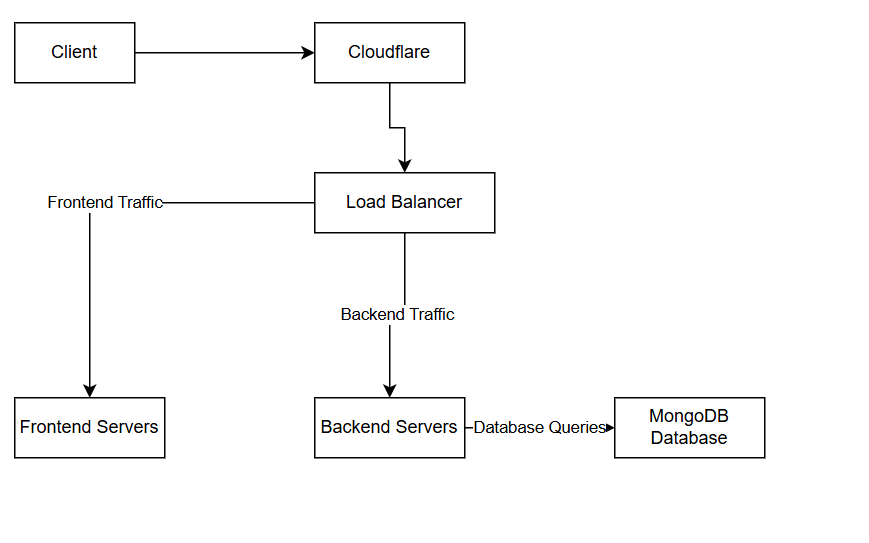
**Project Repository:**

Access the complete codebase of the Travel Memory application from the provided GitHub link: https://github.com/UnpredictablePrashant/TravelMemory

**Objective:**

* Set up the backend running on Node.js.
* Configure the front end designed with React.
* Ensure efficient communication between the front end and back end.
* Deploy the full application on an EC2 instance.
* Facilitate load balancing by creating multiple instances of the application.
* Connect a custom domain through Cloudflare.

**Data Flow Diagram from Draw.io –**



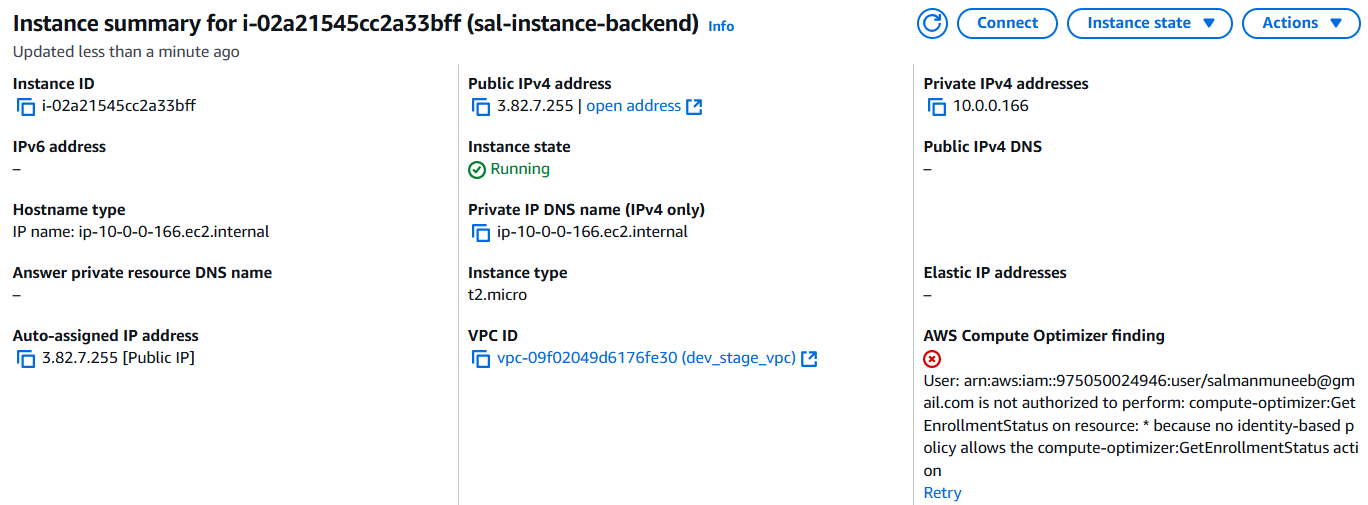
**Tasks:**

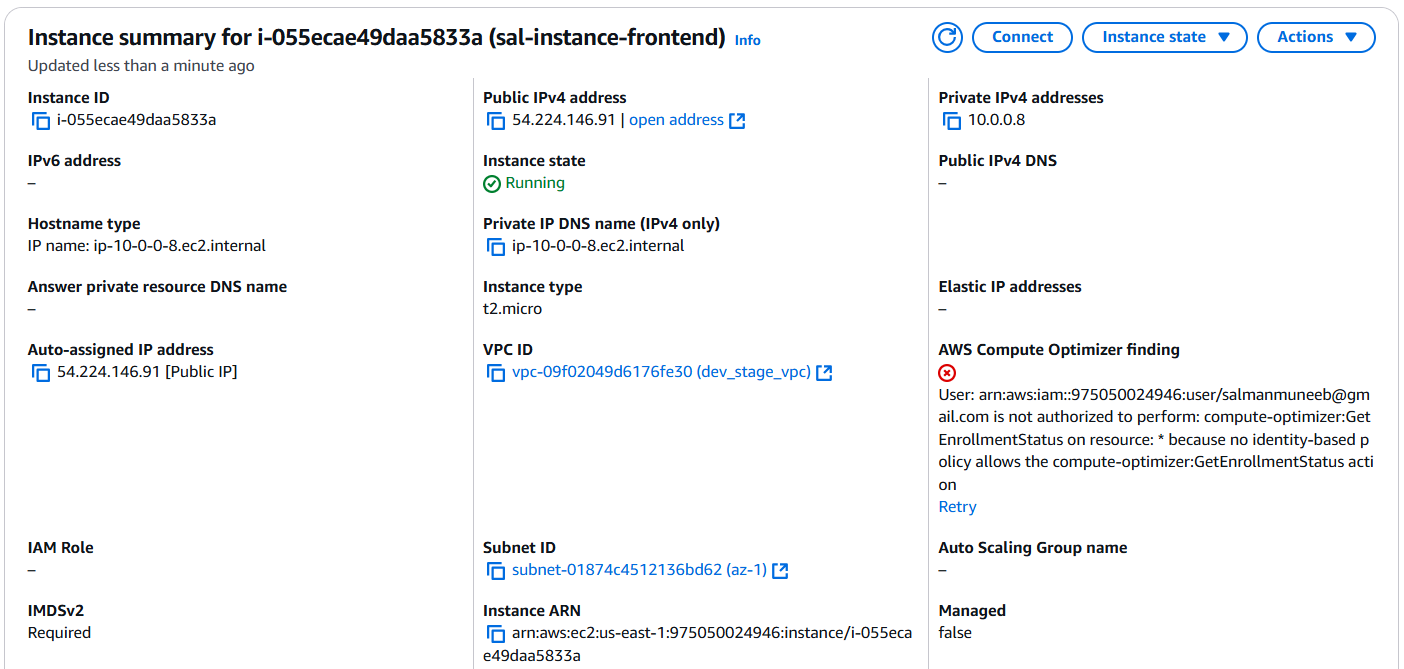
1. **Backend & Frontend Configuration:**

* Clone the repository and navigate to the backend directory.
* The backend runs on port 3000. Set up a reverse proxy using nginx to ensure smooth deployment on EC2.
* Update the .env file to incorporate database connection details and port information.
* Launching the backend & frontend named “**sal-instance-backend**” & “**sal-instance-frontend**” having **Amazon Linux Machine** as a base image, **t2.micro** as Instance type.



* Installing the Nodejs 18 and cloned the git repository of TravelMemory application.
* sudo apt install nodejs -y
* sudo npm install
* sudo git clone <https://github.com/UnpredictablePrashant/TravelMemory>
* The Backend & Frontend instances are up and running.



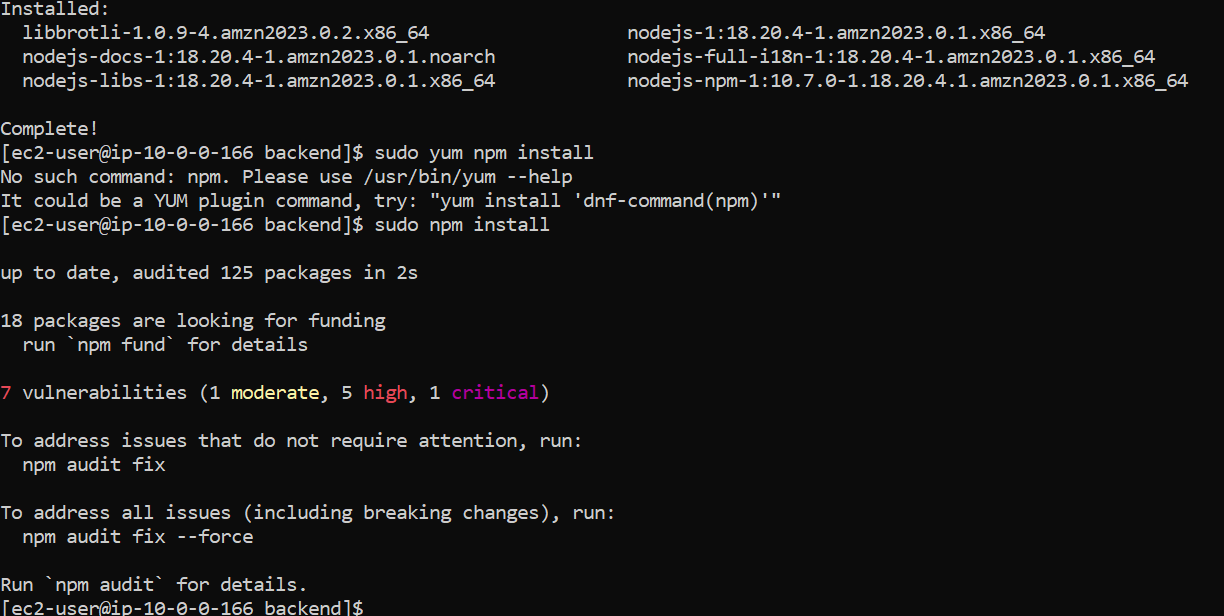


* Create a **.env file** in backend and provide the Mongo DB creds –

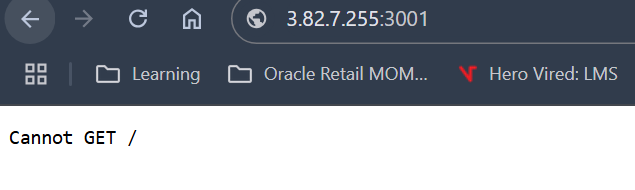
PORT = 3001

MONGO\_URI = mongodb+srv://salmanmuneeb:zGP58fVIXLQBaLXC@cluster14.sjvuc.mongodb.net/

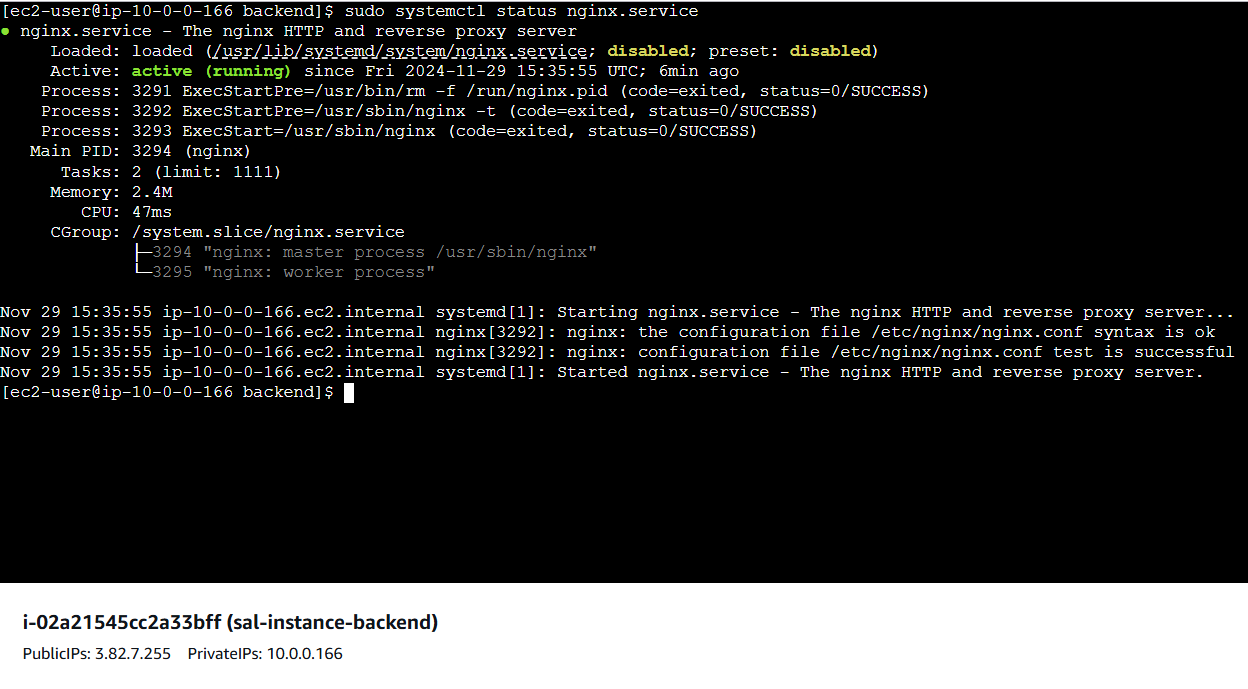
* **sudo npm install** -> to install the dependencies of the node application



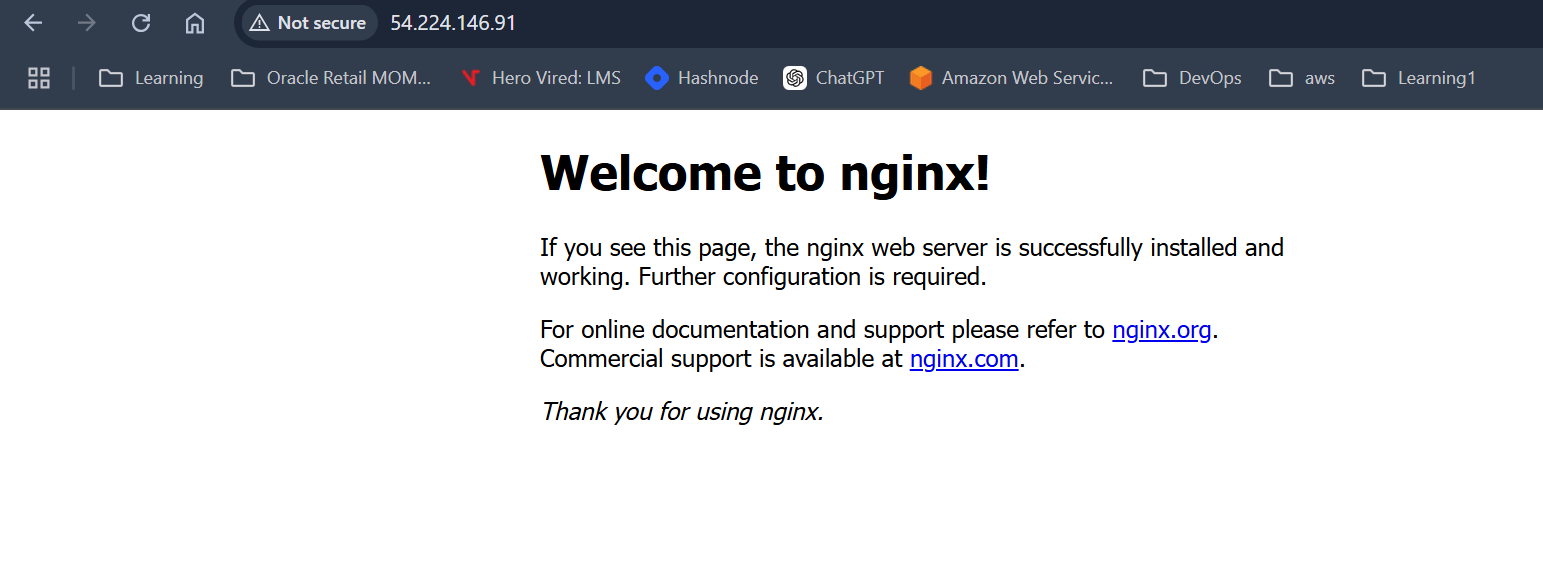
* **sudo node index.js** -> will start the backend at port 3001
* We can access it browser by using the Public IP and port



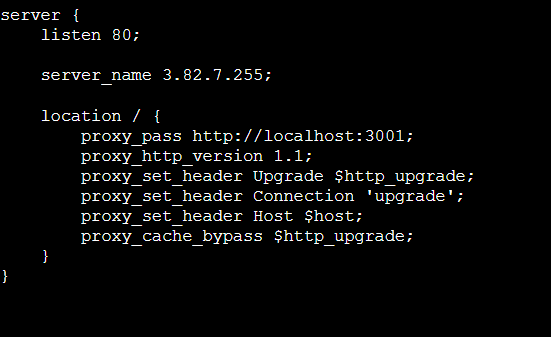
* Now install the ngnix and check the status
* sudo apt install nginx
* sudo systemctl status nginx
* sudo systemctl statrt nginx
* sudo nginx -t



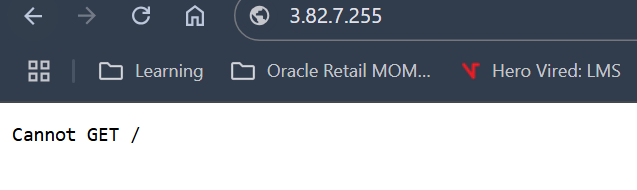
* We can check the nginx is running at port 80



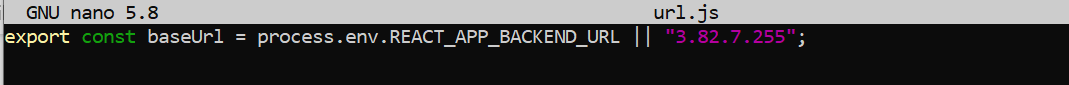
* Now for the Reverse Proxy – Configured Server tag and provided Backend Instance Public ip.



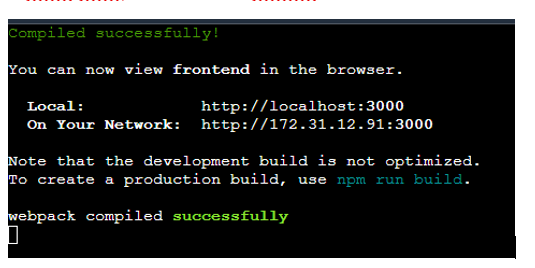
* After restarting nginx, we can access the backend at port 80 because of reverse proxy



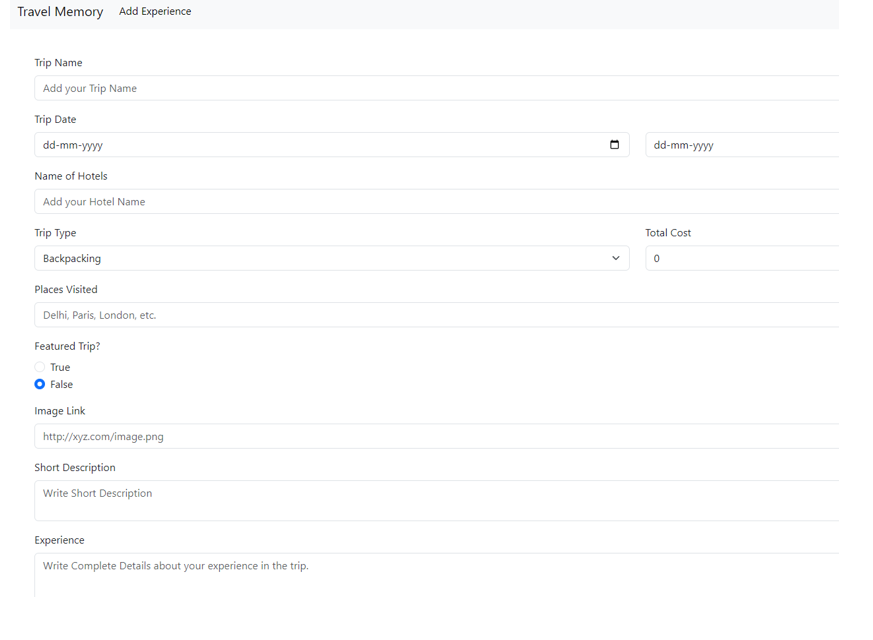
* Navigate to the `urls.js`, path = (/home/ec2-user/travelmemory/TravelMemory/frontend/src/) in the frontend directory.
* Update the file to ensure the frontend communicates effectively with the backend.

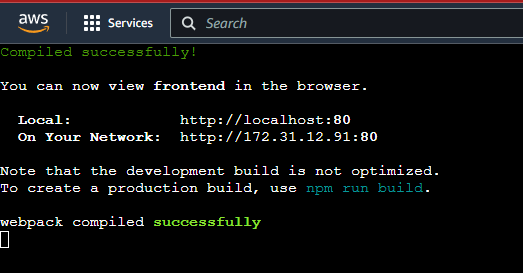


* sudo npm start -> the nodejs frontend application is running at port 3000



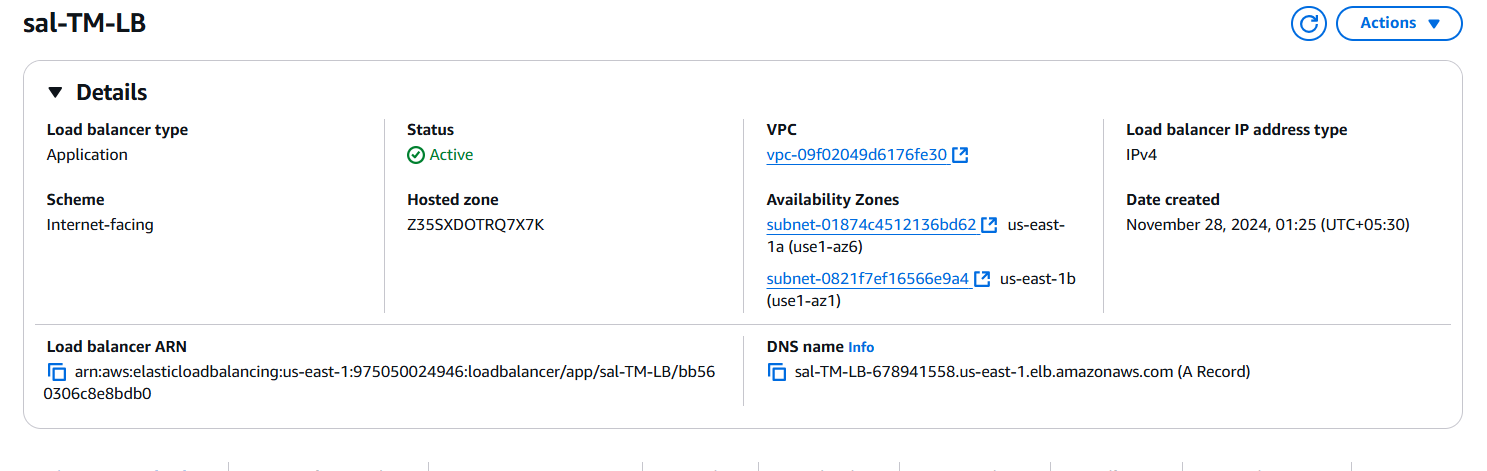
* After doing the reverse proxy frontend will be running at port 80

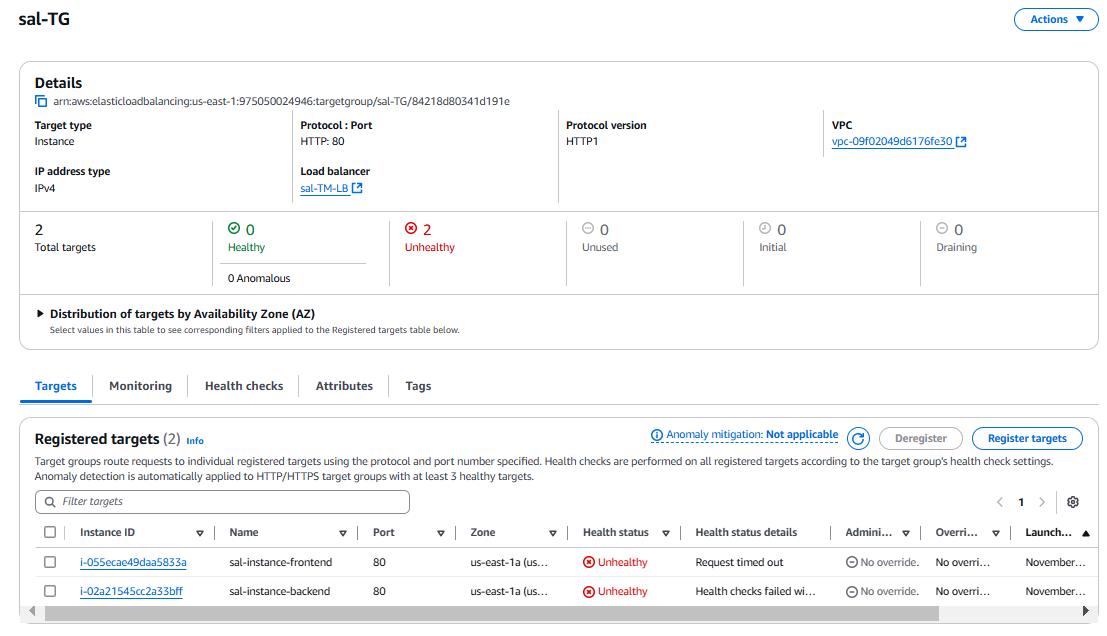




1. Load Balancer Configuration:

* Created an Application Load balancer with 2 Availability zones
* Added a Target Group to add the 2 Instances for frontend and backend





1. **Domain Setup with Cloudflare:**

* Connect your custom domain to the application using Cloudflare.
* Create a CNAME record pointing to the load balancer endpoint.
* Set up an A record with the IP address of the EC2 instance hosting the frontend.

Used the domain [adarshkumars.co.in] to point towards the Frontend Load Balancer by adding a CNAME entry in Cloudflare and used the subdomain.

