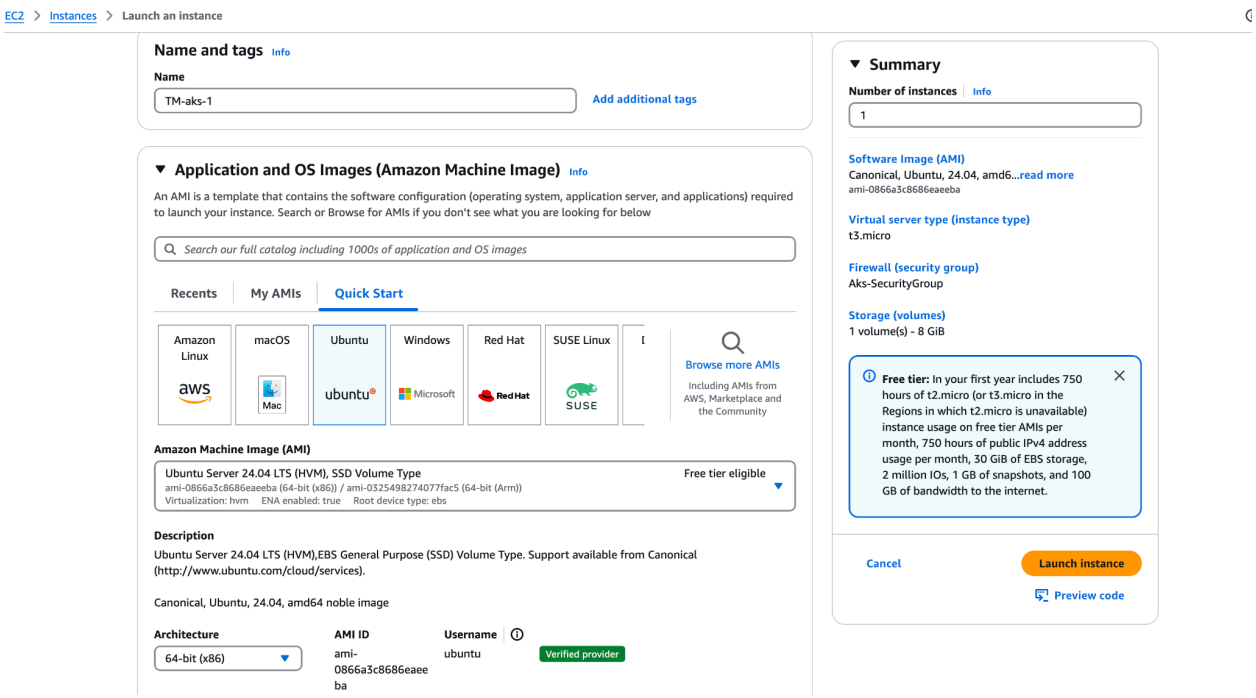
TravelMemory application has been developed using the MERN stack

Tasks:

Backend 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.

EC2 instance creation:



Launch an instance

Instance type

t3.micro
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour
On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour
On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

EC2-AMI-Aks-HV

Create new key pair

Network settings Info

Edit

Network | Info

vpc-09f02049d6176fe30 | dev_stage_vpc

Subnet | Info

subnet-01874c4512136bd62 | az-1

Auto-assign public IP | Info

Disable

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

Common security groups | Info

Select security groups

Aks-SecurityGroup sg-0bc060d9865615fcb

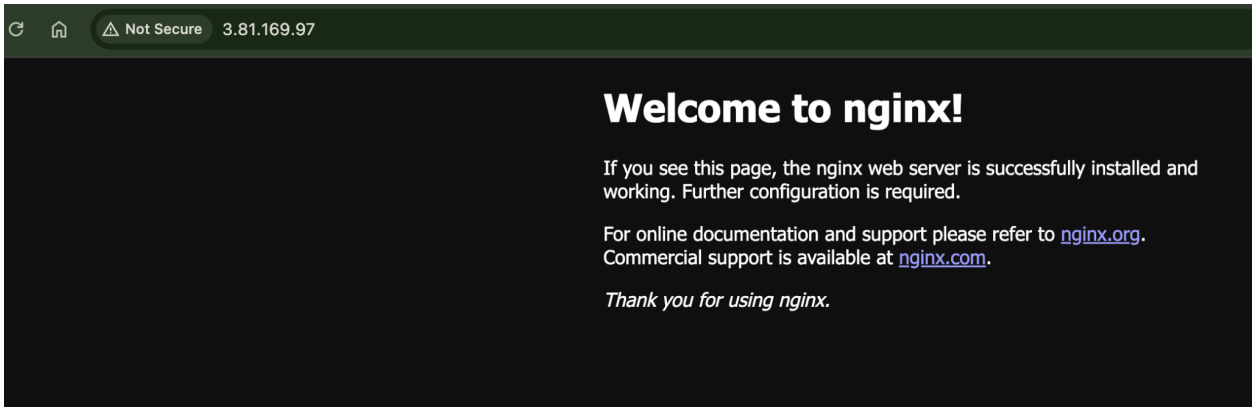
Compare security group rules

Once instances are running and checks are passed. Install services required and clone git repository:

- sudo apt update -y
- sudo apt install git
- git clone https://github.com/UnpredictablePrashant/TravelMemory.git
- sudo apt install -y nginx
- sudo apt install nodejs
- sudo apt install npm

Update nginx config as below and validate via new public IP:

```
"
server {
        listen 80;
        server_name 3.81.169.97;
        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;
        }
}
```



Setup Mongodb for this project

- create new database user
- save credentials
- copy connection string for backend folder, file named **.env**:

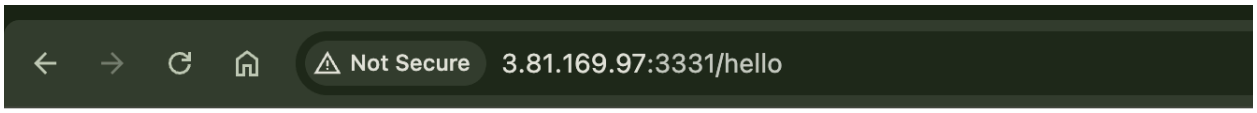
```
ubuntu@ip-10-0-0-218:~/TravelMemory/backend$ cat .env
PORT=3331
MONGO_URI='mongodb+srv://akshaythebest:aZwsDtRwhc0P3kjB@akshay-1-tm.macm0.mongodb.net/tm-akshay'

# URI / DB
```

Install npm and run “node index.js”

```
ubuntu@ip-10-0-0-218:~/TravelMemory/backend$
ubuntu@ip-10-0-0-218:~/TravelMemory/backend$ npm instal

added 117 packages, and audited 118 packages in 4s

13 packages are looking for funding
  run `npm fund` for details

13 vulnerabilities (3 low, 1 moderate, 8 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
ubuntu@ip-10-0-0-218:~/TravelMemory/backend$
ubuntu@ip-10-0-0-218:~/TravelMemory/backend$
ubuntu@ip-10-0-0-218:~/TravelMemory/backend$ node index.js
Server started at http://localhost:3331
```

Verify the status by replacing localhost to EC2 public IP address.

```
ubuntu@ip-10-0-0-218:~/TravelMemory/backend$ cat index.js
const express = require('express')
const cors = require('cors')
require('dotenv').config()

const app = express()
PORT = process.env.PORT
const conn = require('./conn')
app.use(express.json())
app.use(cors())

const tripRoutes = require('./routes/trip.routes')

app.use('/trip', tripRoutes) // http://3.81.169.97:3001/trip --> POST/GET/GET by ID

app.get('/hello', (req,res)=>{
    res.send('Hello World!')
})

app.listen(PORT, ()=>{
    console.log(`Server started at http://3.81.169.97:${PORT}`)
})
ubuntu@ip-10-0-0-218:~/TravelMemory/backend$
```



Hello World!

The backend is working now…..

2. Frontend and Backend Connection:
- Navigate to the `urls.js` in the frontend directory.
- Update the file to ensure the front end communicates effectively with the backend.

Note— Perform the same installation and git clone as performed on backend server **as page no 3 on this doc**

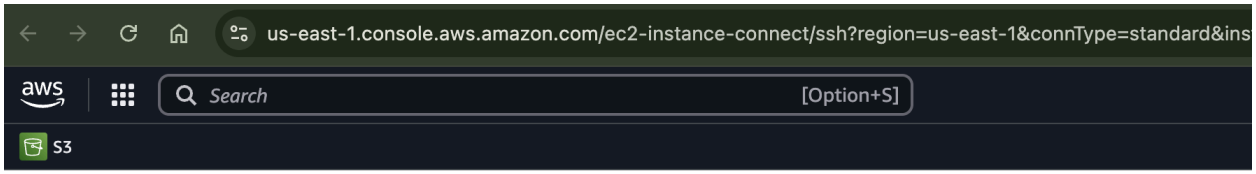
Apply changes to nginx folder **/etc/nginx/sites-available/default**
as did on the backend server.

```
#

server {
        listen 80;
        server_name 3.88.48.97;
        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;
        }
}
```

– Modify file ***frontend/src/url.js*** to point to the backend server IP address:

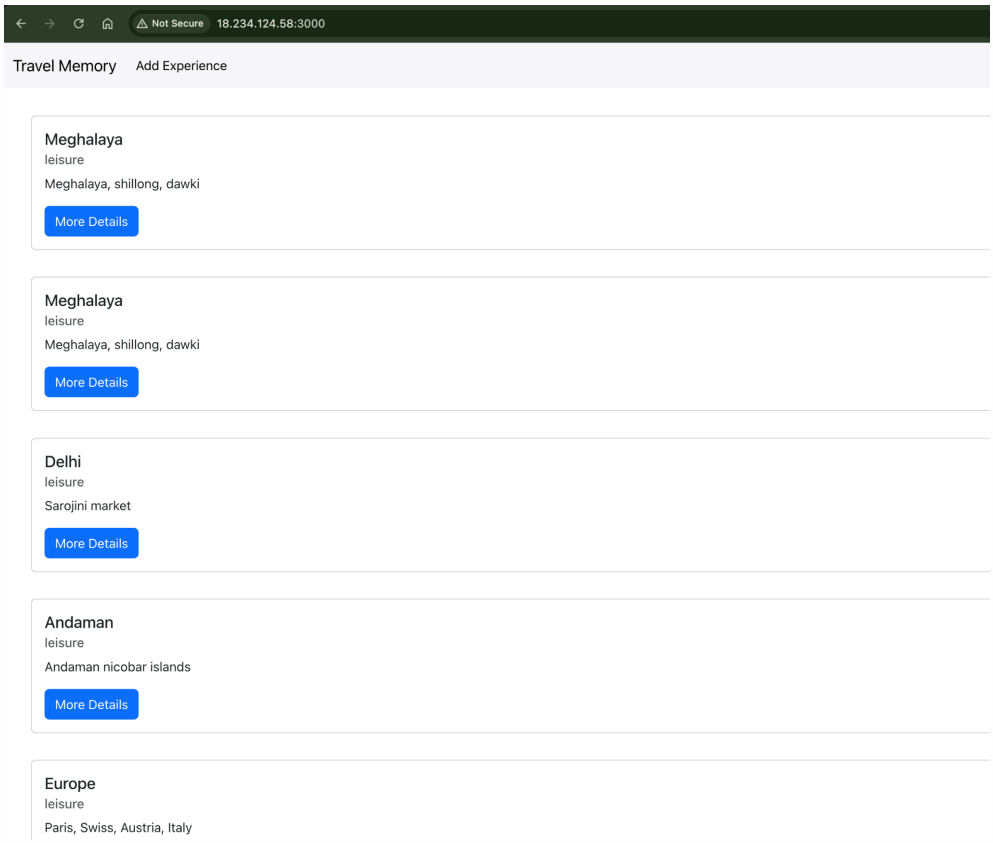
```
ubuntu@ip-10-0-0-19:~$ cat TravelMemory/frontend/src/url.js
export const baseUrl = process.env.REACT_APP_BACKEND_URL || "http://3.81.169.97:3331";
ubuntu@ip-10-0-0-19:~$
```

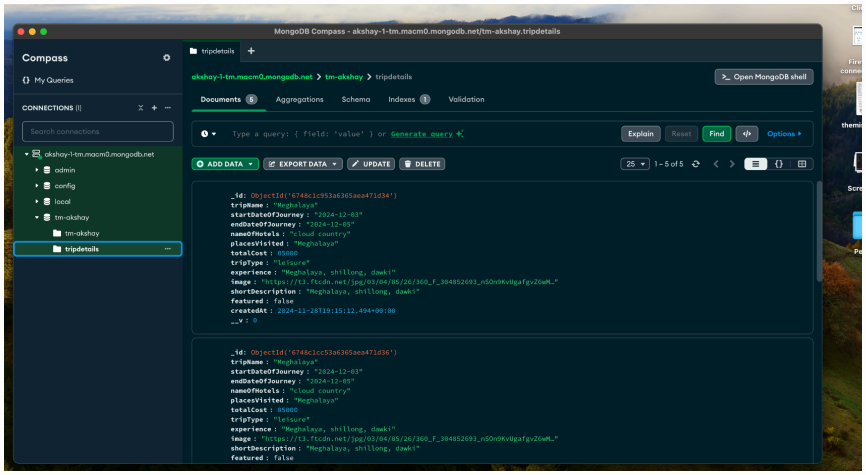
Run below from frontend path:
Npm install
Npm start




```
Compiled successfully!

You can now view frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.0.0.105:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Enter the data in Add experience and save or submit .Create files should appear in mongodb database:

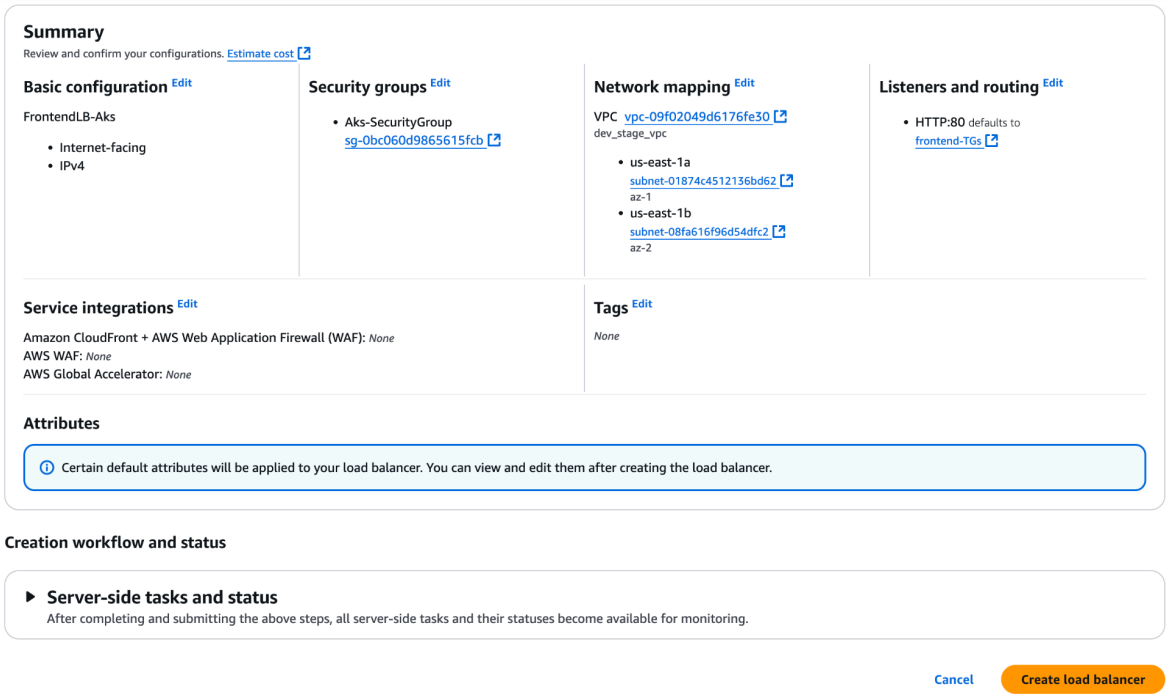




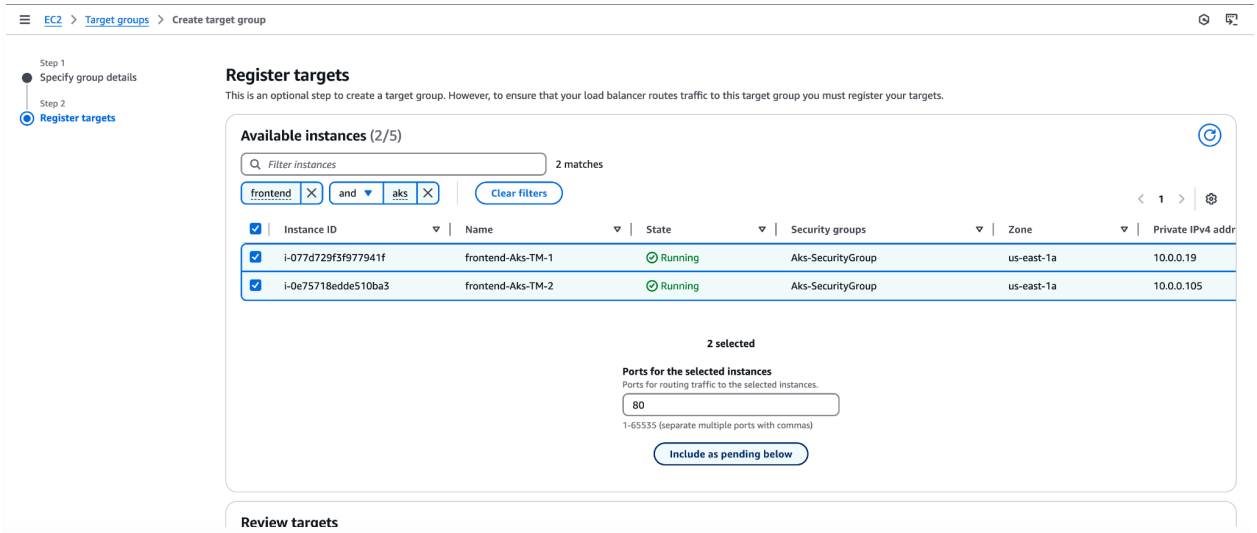
Frontend is connecting to the backend server now.

3. Scaling the Application:
- Create multiple instances of both the frontend and backend servers.
- Add these instances to a load balancer to ensure efficient distribution of incoming traffic.

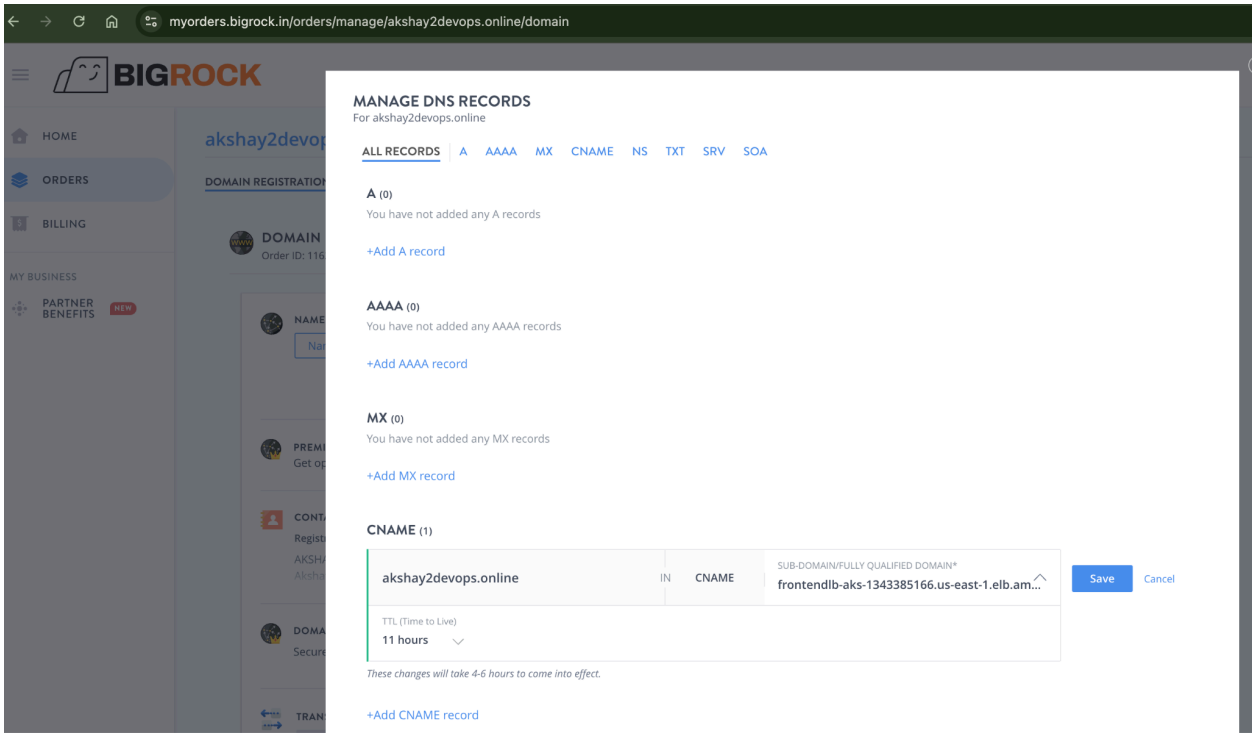
LoudBalancer creation



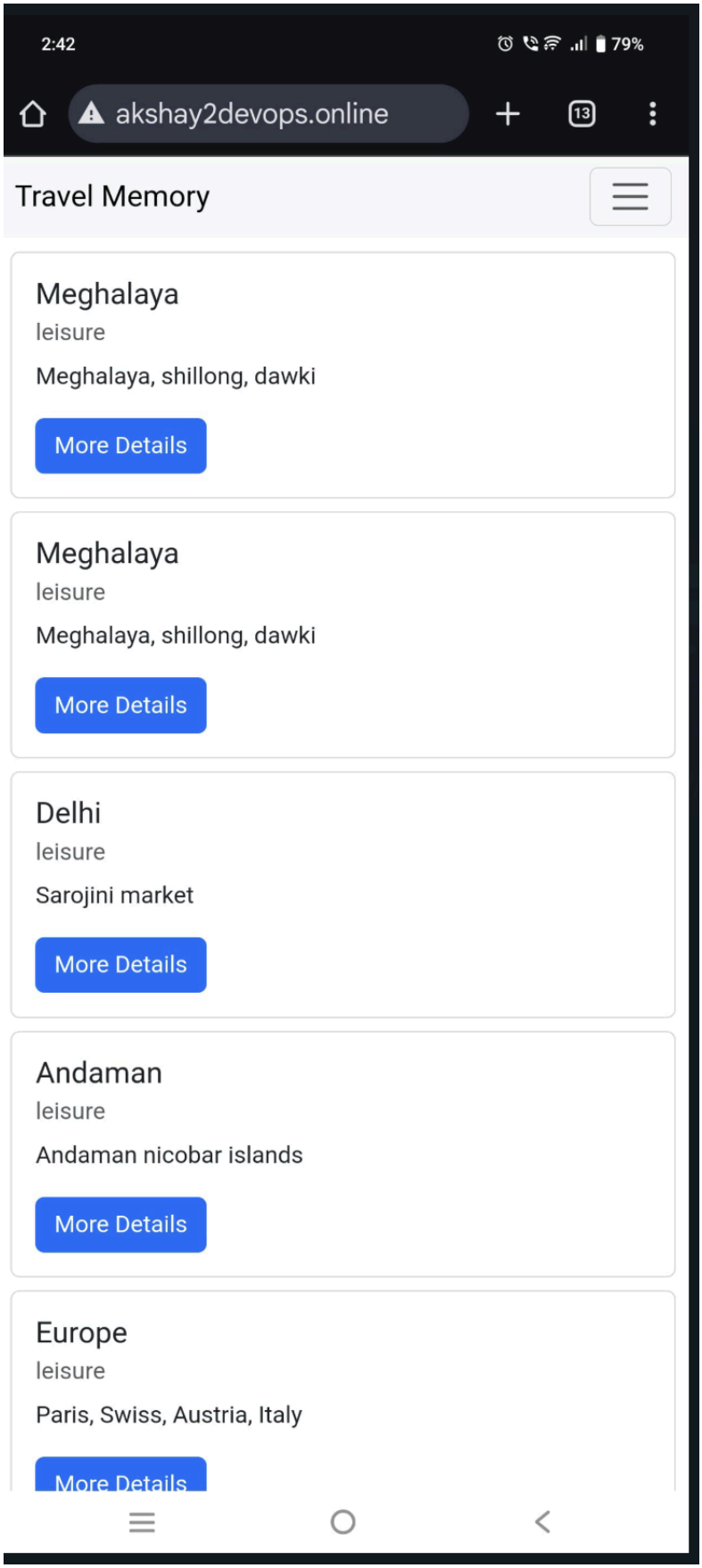
Adding Targets to frontend LB:

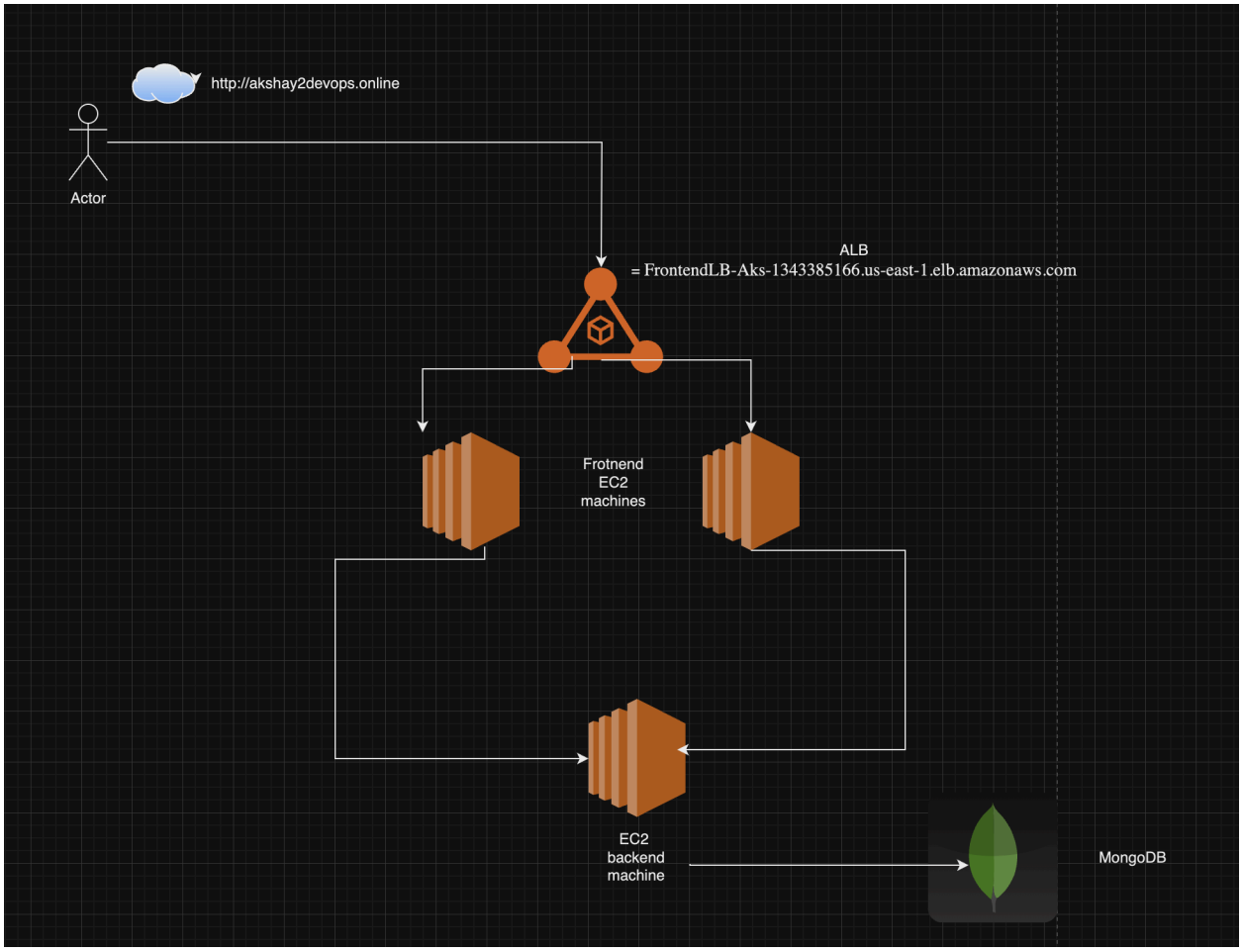


Purchased a domain and created CNAME to point frontend loadbalancer:



Validate the check on purchased domain name:


2:42
79%
akshay2devops.online
13
Travel Memory
Meghalaya
leisure
Meghalaya, shillong, dawki
More Details
Meghalaya
leisure
Meghalaya, shillong, dawki
More Details
Delhi
leisure
Sarojini market
More Details
Andaman
leisure
Andaman nicobar islands
More Details
Europe
leisure
Paris, Swiss, Austria, Italy
More Details


http://akshay2devops.online
Actor
ALB
= FrontendLB-Aks-1343385166.us-east-1.elb.amazonaws.com
Frotnend
EC2
machines
EC2
backend
machine
MongoDB