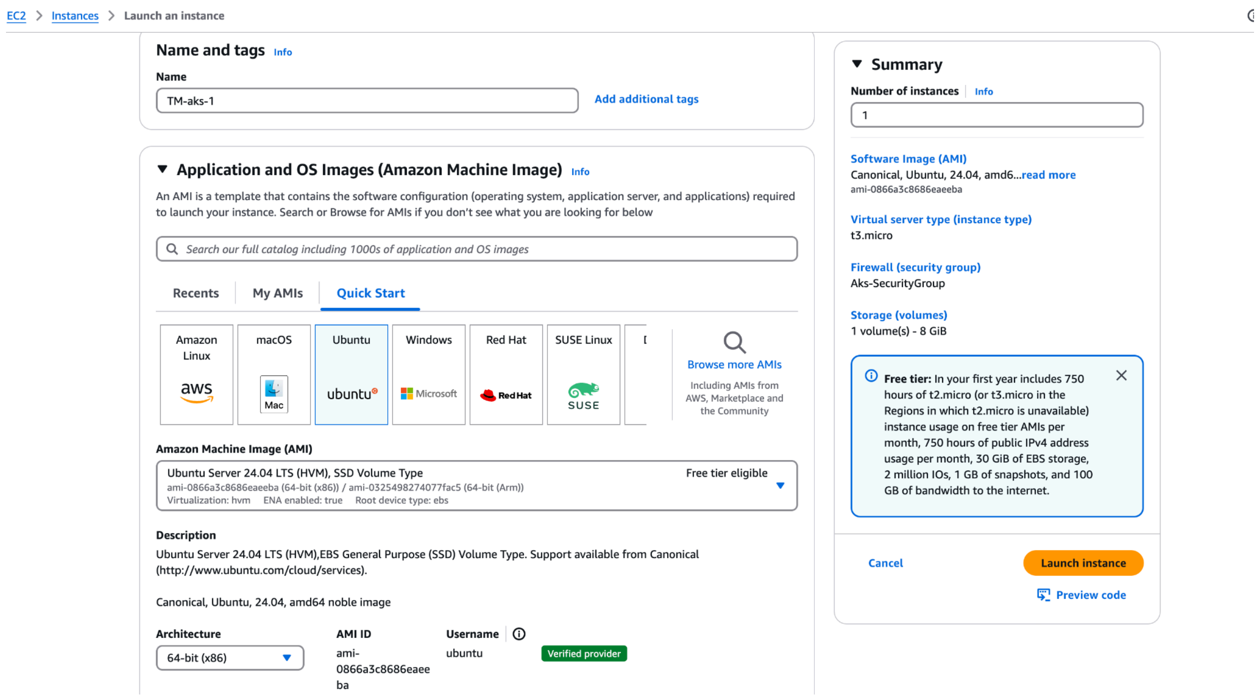
TravelMemory application has been developed using the MERN stack

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:



EC2 > Instances > Launch an instance

Name and tags Info

Name: TM-aks-1

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | My AMIs | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Browse more AMIs — Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0866a3c8686eaeeba (64-bit (x86)) / ami-0325498274077fac5 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Username	
64-bit (x86)	ami-0866a3c8686eaeeba	ubuntu	Verified provider

Summary

Number of instances: 1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0866a3c8686eaeeba

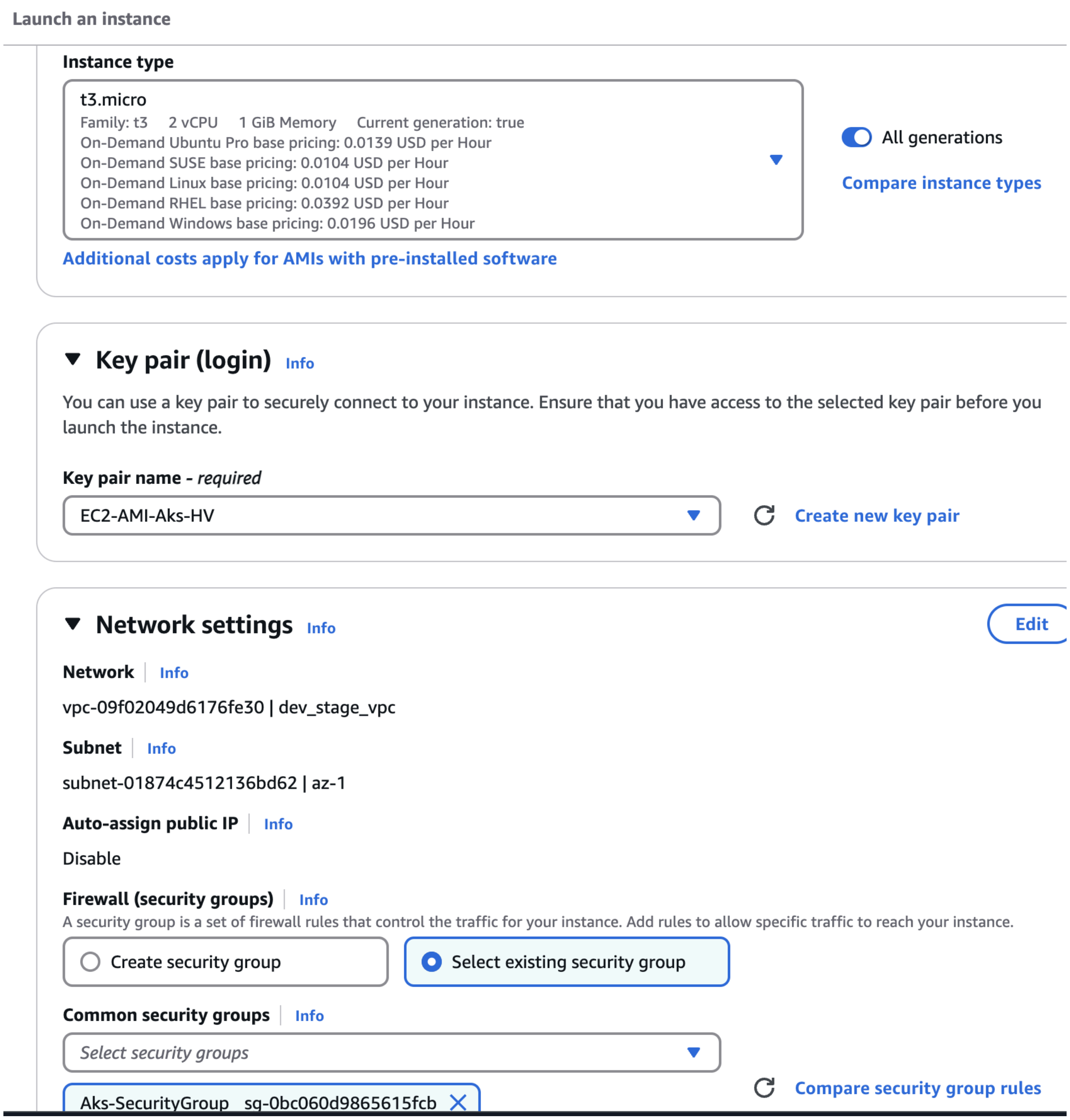
Virtual server type (instance type)
t3.micro

Firewall (security group)
Aks-SecurityGroup

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Launch instance | Preview code


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

Compare security group rules

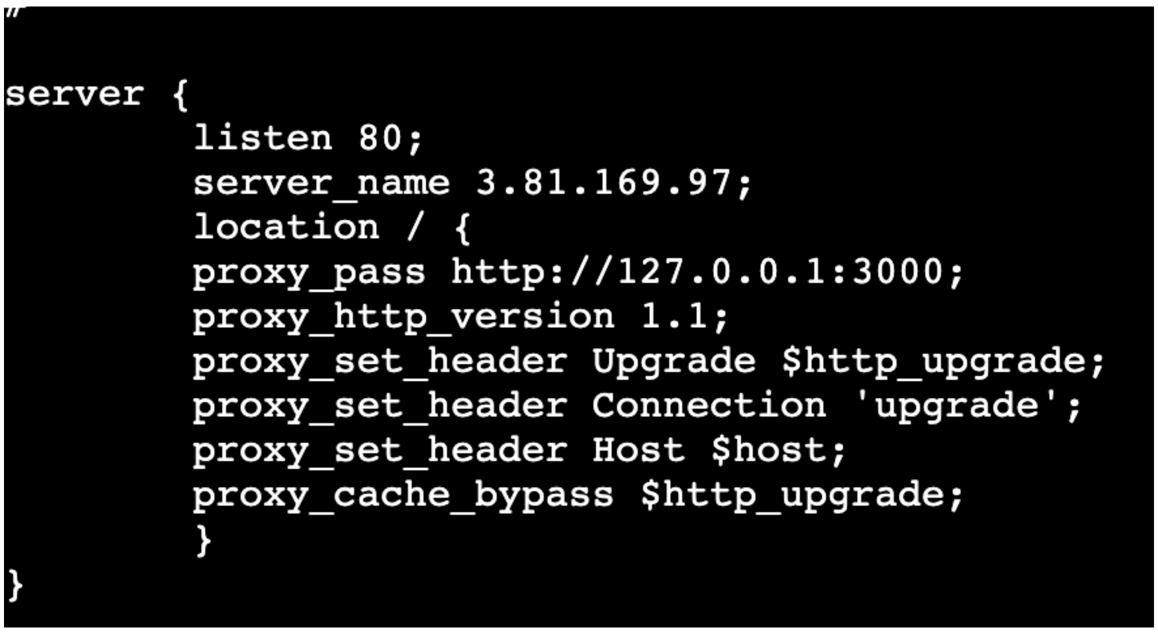
Aks-SecurityGroup sg-0bc060d9865615fcb

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

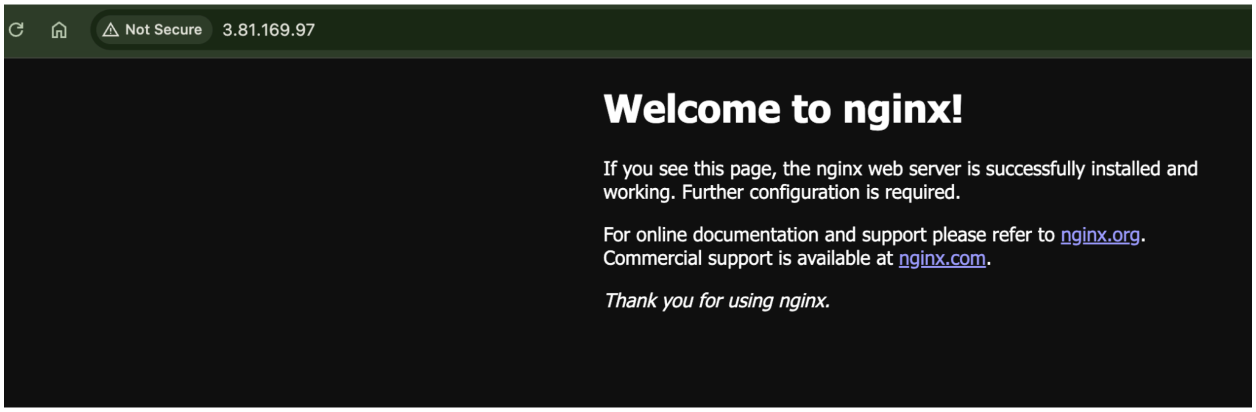
- sudo apt update -y
- sudo apt install git
- git clone https://github.com/UnpredictablePrashant/TravelMemory.git
- sudo apt install -y nginx
- sudo apt install nodejs
- sudo apt install npm

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

Path /etc/nginx/sites-available/default



```
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;
        }
}
```



Welcome to nginx!

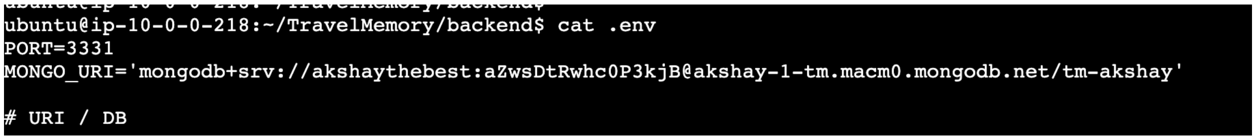
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

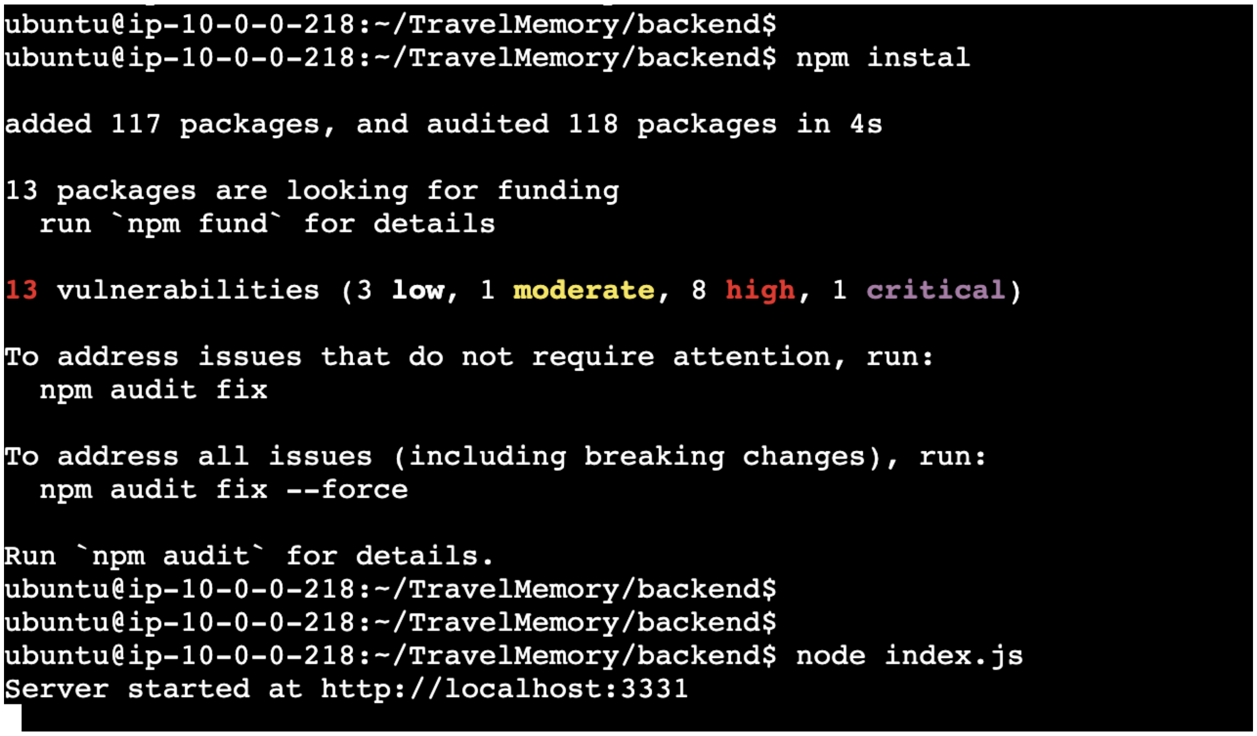
Setup Mongodb for this project

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



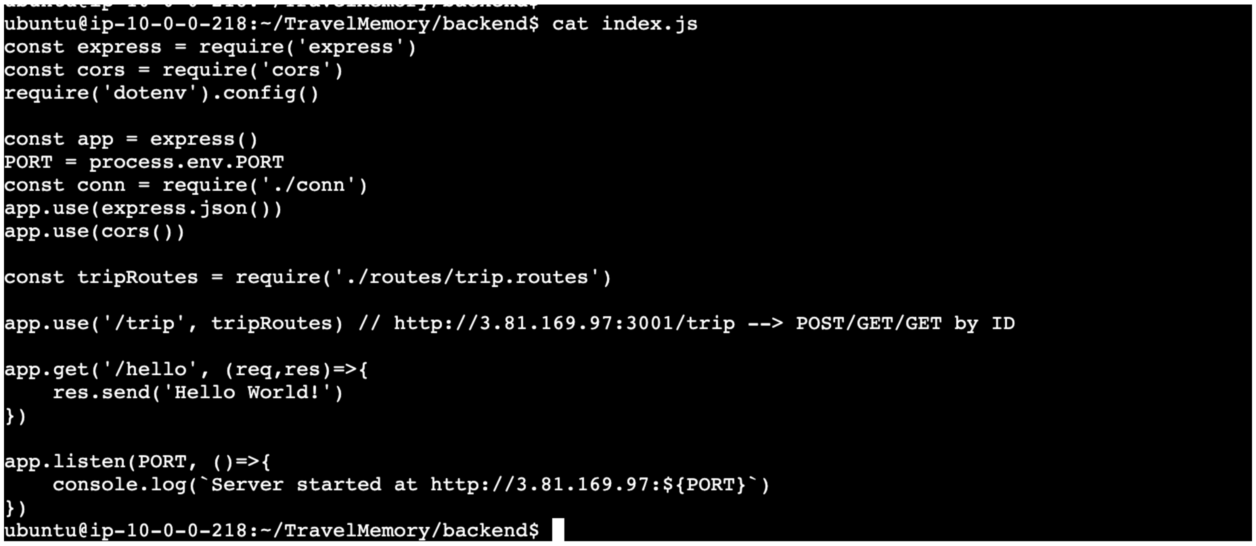
```
ubuntu@ip-10-0-0-218:~/TravelMemory/backend$ cat .env
PORT=3331
MONGO_URI='mongodb+srv://akshaythebest:aZwsDtRwhc0P3kjB@akshay-1-tm.macm0.mongodb.net/tm-akshay'

# URI / DB
```

Install npm and run “node index.js”

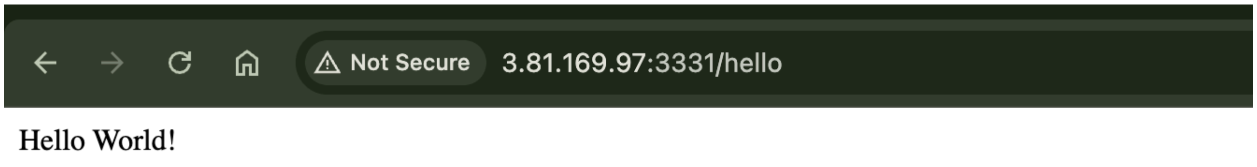


```
ubuntu@ip-10-0-0-218:~/TravelMemory/backend$
ubuntu@ip-10-0-0-218:~/TravelMemory/backend$ npm instal

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!

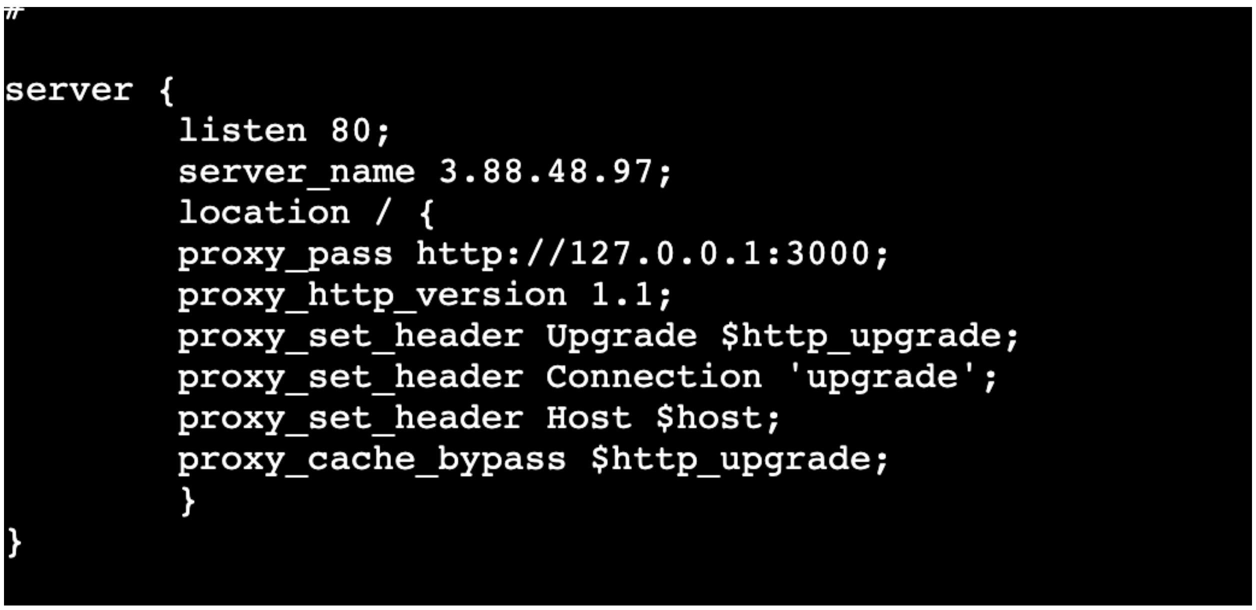
The backend is working now…..

Perform the same and create another backend EC2 instance.

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.

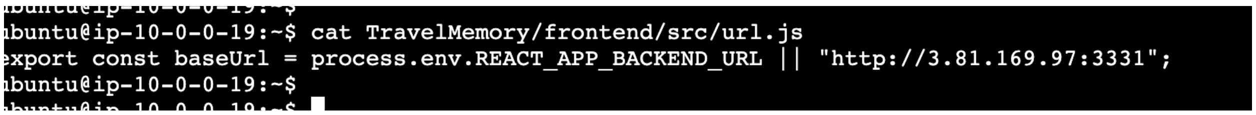
Note— Perform the same installation and git clone as performed on backend server **as page no 3 on this doc**

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



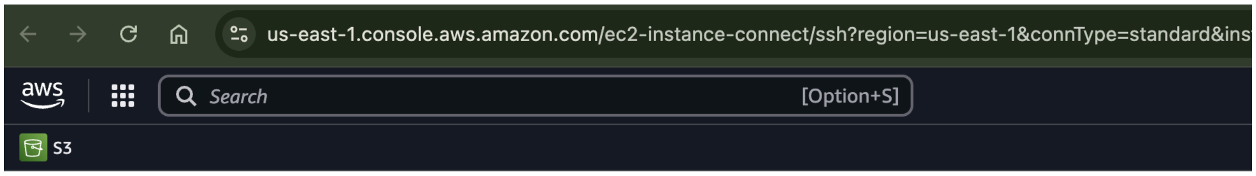
```
#

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;
        }
}
```

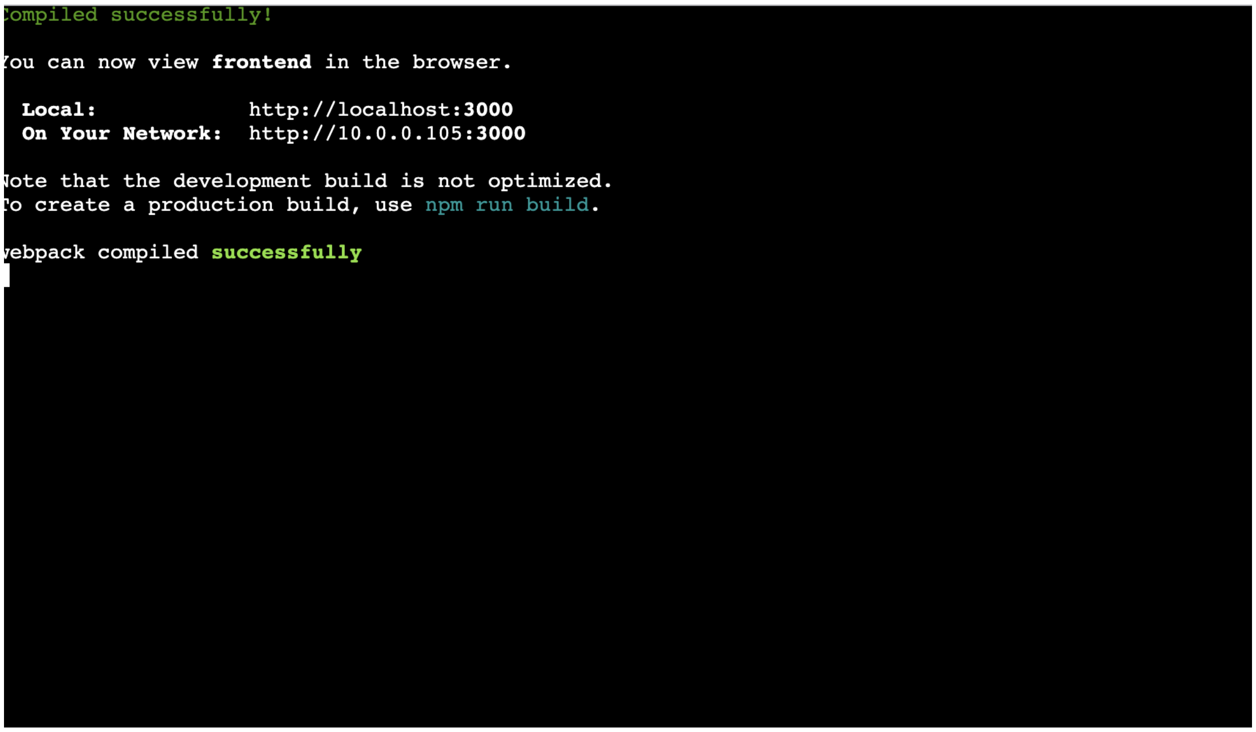
– Modify file ***frontend/src/url.js*** to point to the backend server IP address:



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

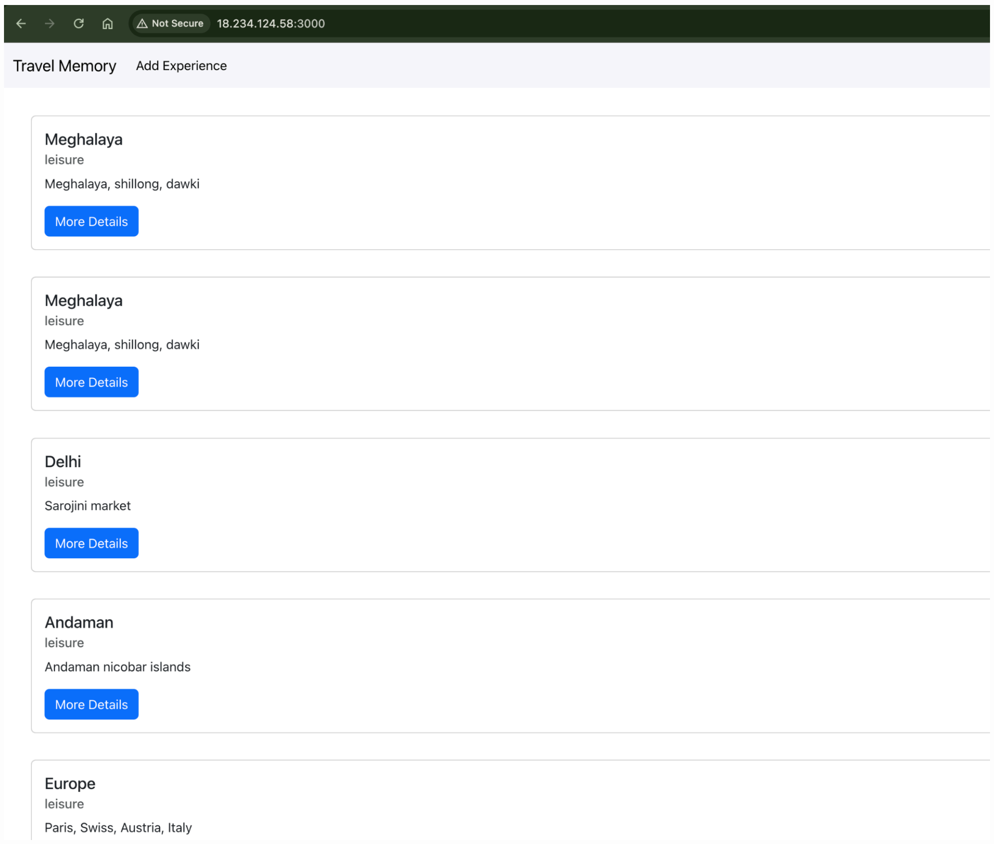
Run below from frontend path:
Npm install
Npm start


us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&ins
aws
Search
[Option+S]
S3



```
ompiled successfully!

ou can now view frontend in the browser.

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

ote that the development build is not optimized.
o create a production build, use npm run build.

ebpack compiled successfully
```

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


Not Secure 18.234.124.58:3000

Travel Memory   Add Experience

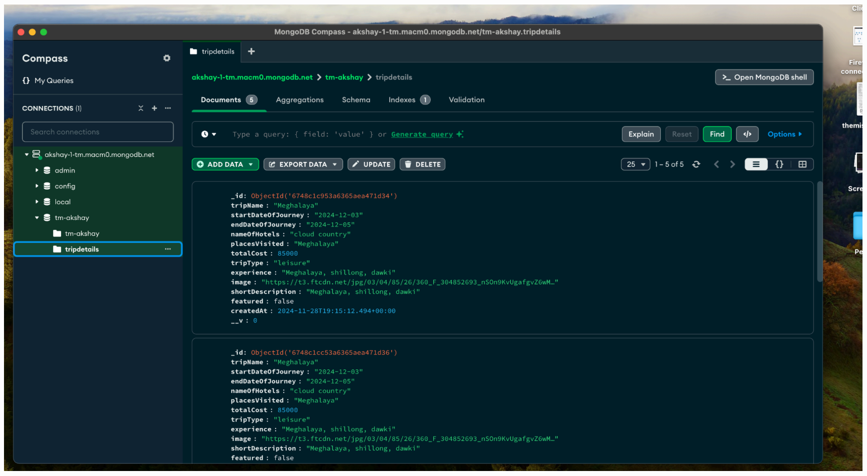
Meghalaya
leisure
Meghalaya, shillong, dawki
More Details

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


MongoDB Compass - akshay-1-tm.macm0.mongodb.net/tm-akshay.tripdetails

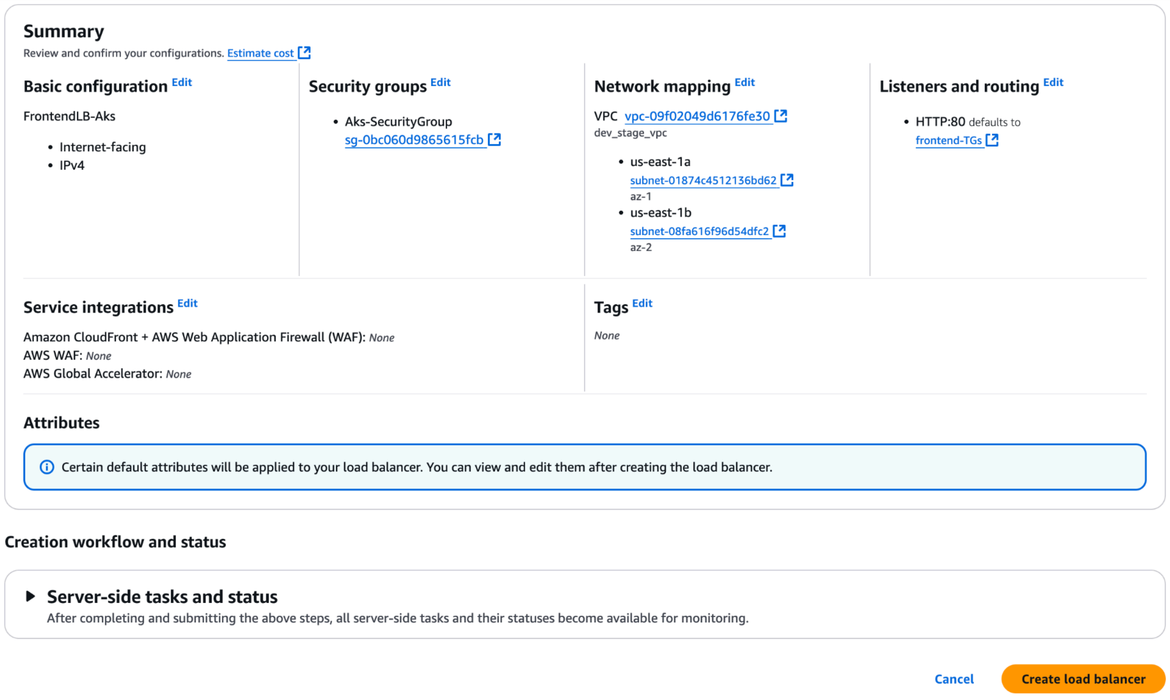
akshay-1-tm.macm0.mongodb.net > tm-akshay > tripdetails

```
_id: ObjectId('6748c1c953a6365aea471d34')
tripName : "Meghalaya"
startDateOfJourney : "2024-12-03"
endDateOfJourney : "2024-12-05"
nameOfHotels : "cloud country"
placesVisited : "Meghalaya"
totalCost : 85000
tripType : "leisure"
experience : "Meghalaya, shillong, dawki"
image : "https://t3.ftcdn.net/jpg/03/04/85/26/360_F_304852693_nSOn9KvUgafgvZ6wM…"
shortDescription : "Meghalaya, shillong, dawki"
featured : false
createdAt : 2024-11-28T19:15:12.494+00:00
__v : 0

_id: ObjectId('6748c1cc53a6365aea471d36')
tripName : "Meghalaya"
startDateOfJourney : "2024-12-03"
endDateOfJourney : "2024-12-05"
nameOfHotels : "cloud country"
placesVisited : "Meghalaya"
totalCost : 85000
tripType : "leisure"
experience : "Meghalaya, shillong, dawki"
image : "https://t3.ftcdn.net/jpg/03/04/85/26/360_F_304852693_nSOn9KvUgafgvZ6wM…"
shortDescription : "Meghalaya, shillong, dawki"
featured : false
```

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.

<u>LoudBalancer creation</u>



Summary
Review and confirm your configurations. Estimate cost

Basic configuration Edit
FrontendLB-Aks
- Internet-facing
- IPv4

Security groups Edit
- Aks-SecurityGroup
  sg-0bc060d9865615fcb

Network mapping Edit
VPC vpc-09f02049d6176fe30
dev_stage_vpc
- us-east-1a
  subnet-01874c4512136bd62
  az-1
- us-east-1b
  subnet-08fa616f96d54dfc2
  az-2

Listeners and routing Edit
- HTTP:80 defaults to
  frontend-TGs

Service integrations Edit
Amazon CloudFront + AWS Web Application Firewall (WAF): *None*
AWS WAF: *None*
AWS Global Accelerator: *None*

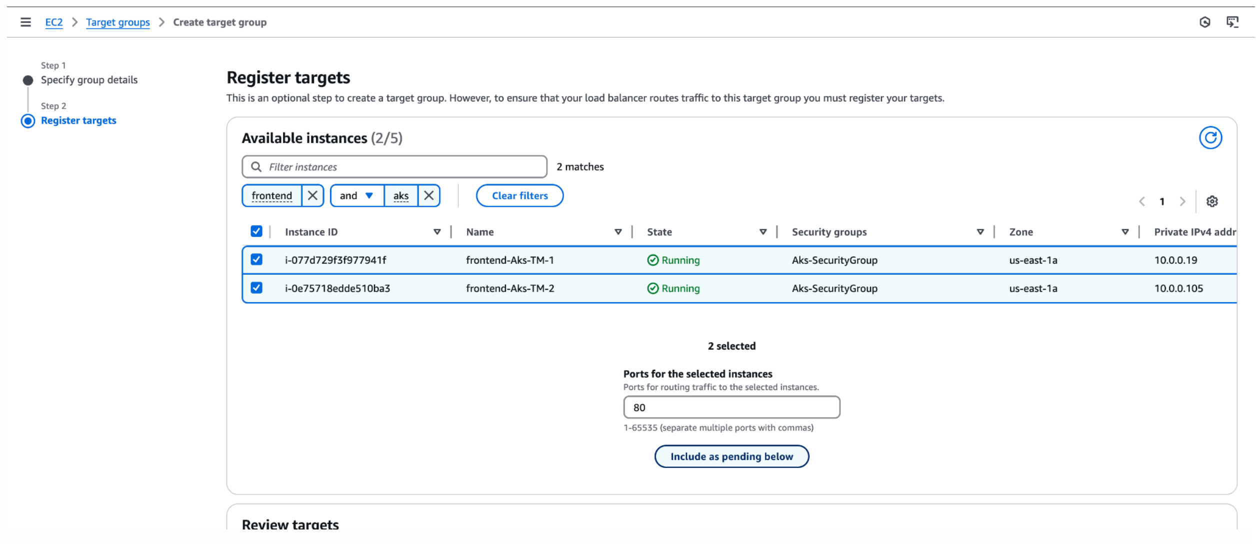
Tags Edit
None

Attributes
Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

Creation workflow and status

Server-side tasks and status
After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

Adding Targets to frontend LB:



Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/5)

Filter instances — 2 matches

frontend and aks — Clear filters

	Instance ID	Name	State	Security groups	Zone	Private IPv4 addr
✓	i-077d729f3f977941f	frontend-Aks-TM-1	Running	Aks-SecurityGroup	us-east-1a	10.0.0.19
✓	i-0e75718edde510ba3	frontend-Aks-TM-2	Running	Aks-SecurityGroup	us-east-1a	10.0.0.105

2 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

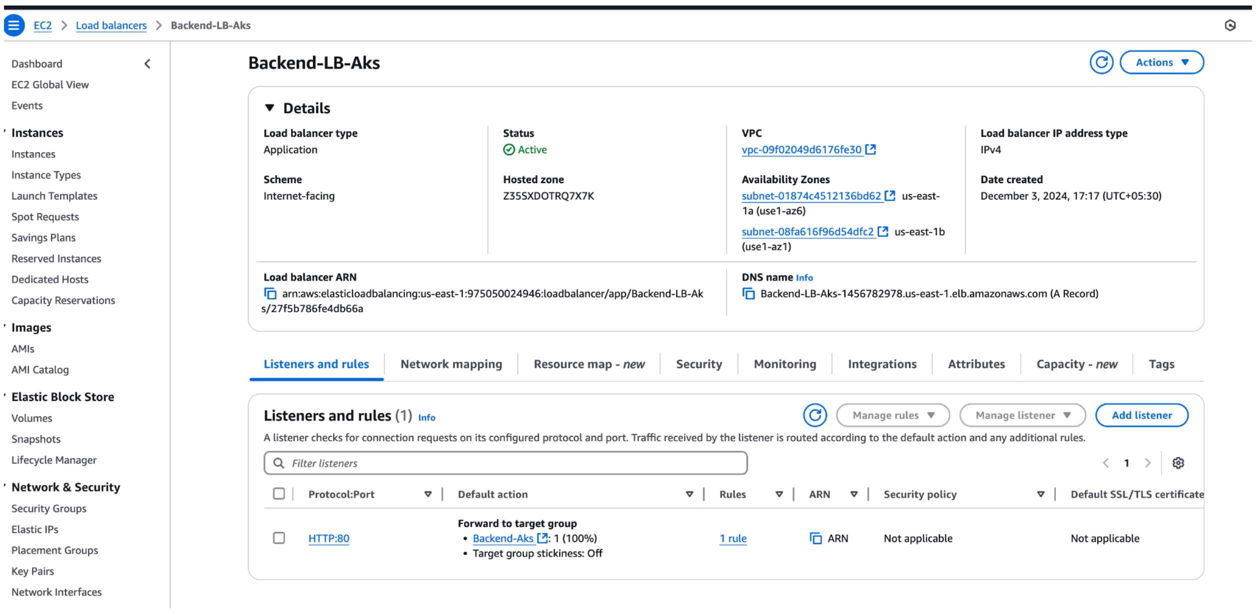
80

1-65535 (separate multiple ports with commas)

Include as pending below

Review targets

Creating backend loadbalancer for backend servers:



Backend-LB-Aks

Details

Load balancer type	Status	VPC	Load balancer IP address type
Application	Active	vpc-09f02049d6176fe30	IPv4
Scheme	**Hosted zone**	**Availability Zones**	**Date created**
Internet-facing	Z35SXDOTRQ7X7K	subnet-01874c4512136bd62 us-east-1a (use1-az6) subnet-08fa616f96d54dfc2 us-east-1b (use1-az1)	December 3, 2024, 17:17 (UTC+05:30)

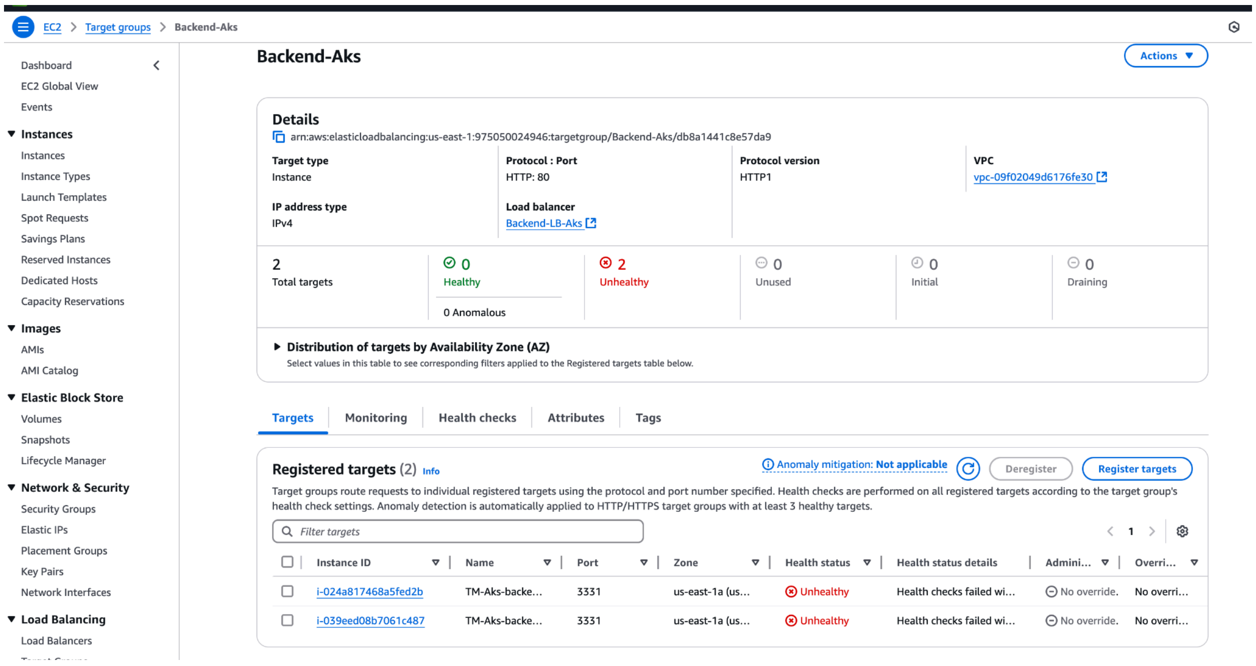
Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:975050024946:loadbalancer/app/Backend-LB-Aks/27f5b786fe4db66a

DNS name Info
Backend-LB-Aks-1456782978.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate
HTTP:80	Forward to target group • Backend-Aks: 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable



Backend-Aks

Details
arn:aws:elasticloadbalancing:us-east-1:975050024946:targetgroup/Backend-Aks/db8a1441c8e57da9

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-09f02049d6176fe30
IP address type	**Load balancer**		
IPv4	Backend-LB-Aks		

2	0	2	0	0	0
Total targets	Healthy 0 Anomalous	Unhealthy	Unused	Initial	Draining

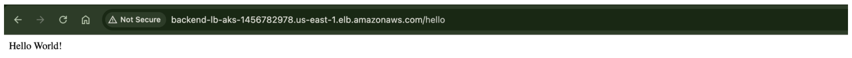
Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (2) Info — Anomaly mitigation: Not applicable

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Instance ID	Name	Port	Zone	Health status	Health status details	Admini...	Overri...
i-024a817468a5fed2b	TM-Aks-backe...	3331	us-east-1a (us...	Unhealthy	Health checks failed wi...	No override.	No overri...
i-039eed08b7061c487	TM-Aks-backe...	3331	us-east-1a (us...	Unhealthy	Health checks failed wi...	No override.	No overri...

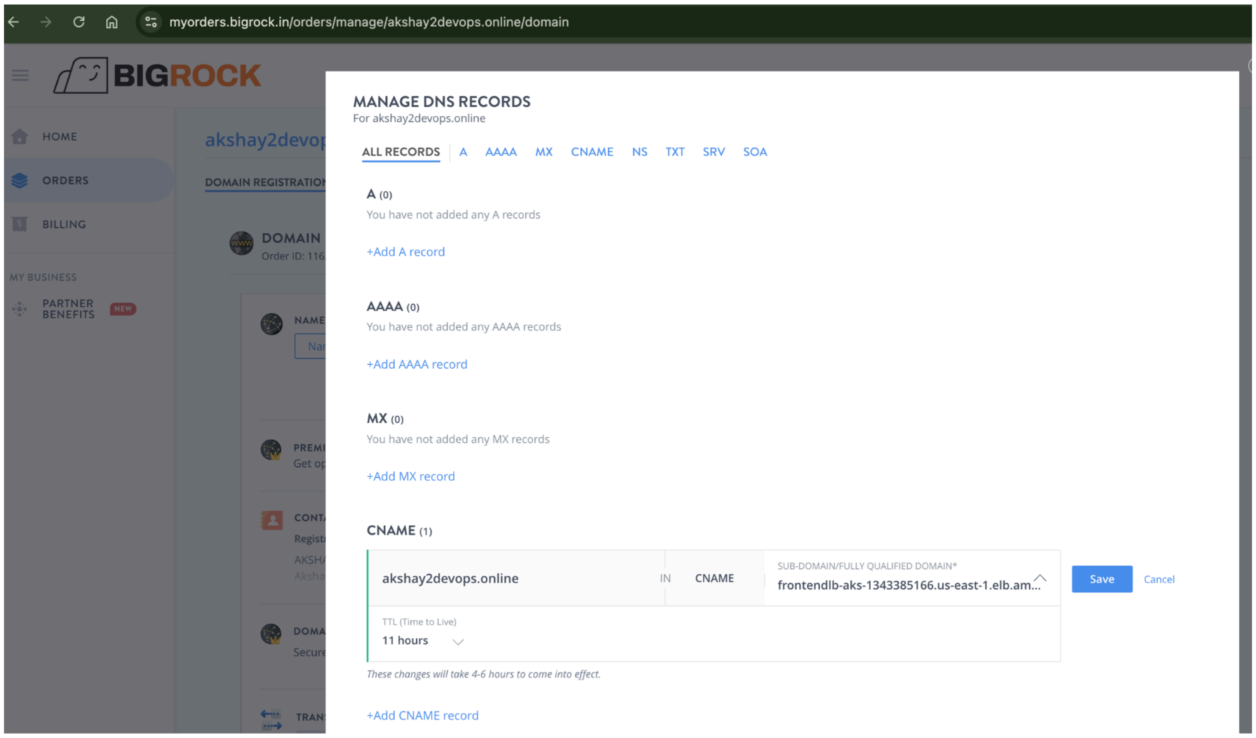
Backend LB DNS can be used to validate if its sending the load to backend servers.



Not Secure backend-lb-aks-1456782978.us-east-1.elb.amazonaws.com/hello

Hello World!

Purchased a domain and created CNAME to point frontend loadbalancer:



myorders.bigrock.in/orders/manage/akshay2devops.online/domain

MANAGE DNS RECORDS

For akshay2devops.online

ALL RECORDS | A | AAAA | MX | CNAME | NS | TXT | SRV | SOA

A (0)

You have not added any A records

+Add A record

AAAA (0)

You have not added any AAAA records

+Add AAAA record

MX (0)

You have not added any MX records

+Add MX record

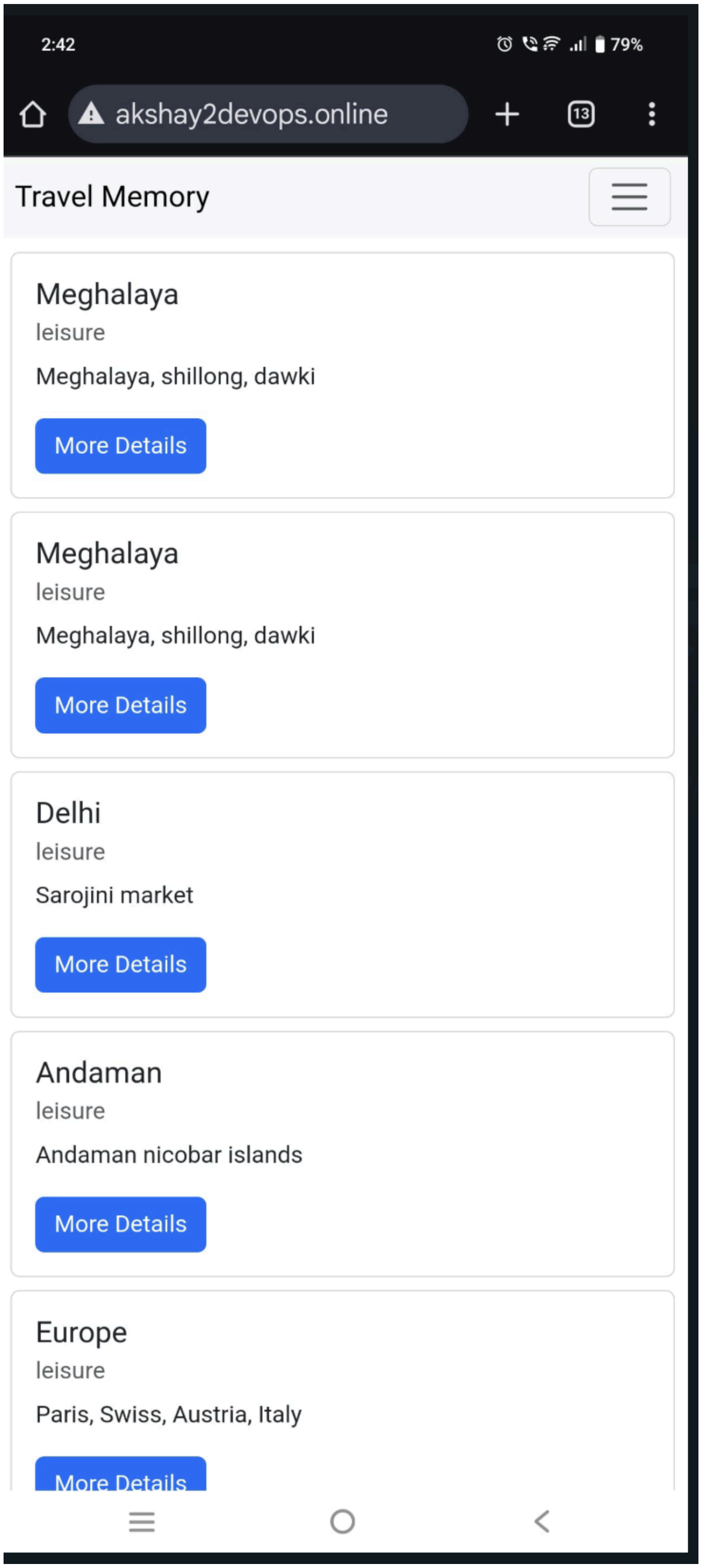
CNAME (1)

akshay2devops.online	IN	CNAME	SUB-DOMAIN/FULLY QUALIFIED DOMAIN* frontendlb-aks-1343385166.us-east-1.elb.am...
TTL (Time to Live) 11 hours			

These changes will take 4-6 hours to come into effect.

+Add CNAME record

Validate the check on purchased domain name:



Travel Memory

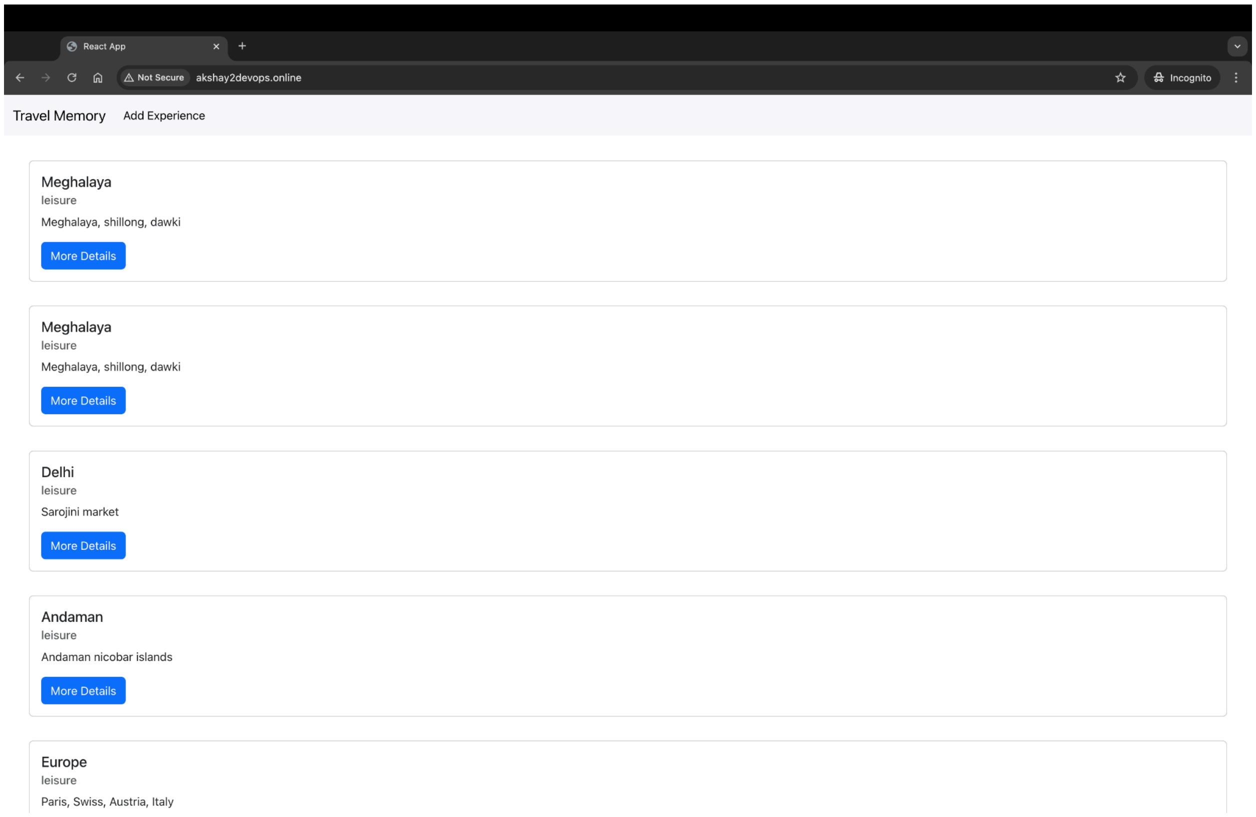
Meghalaya
leisure
Meghalaya, shillong, dawki
More Details

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



Travel Memory Add Experience

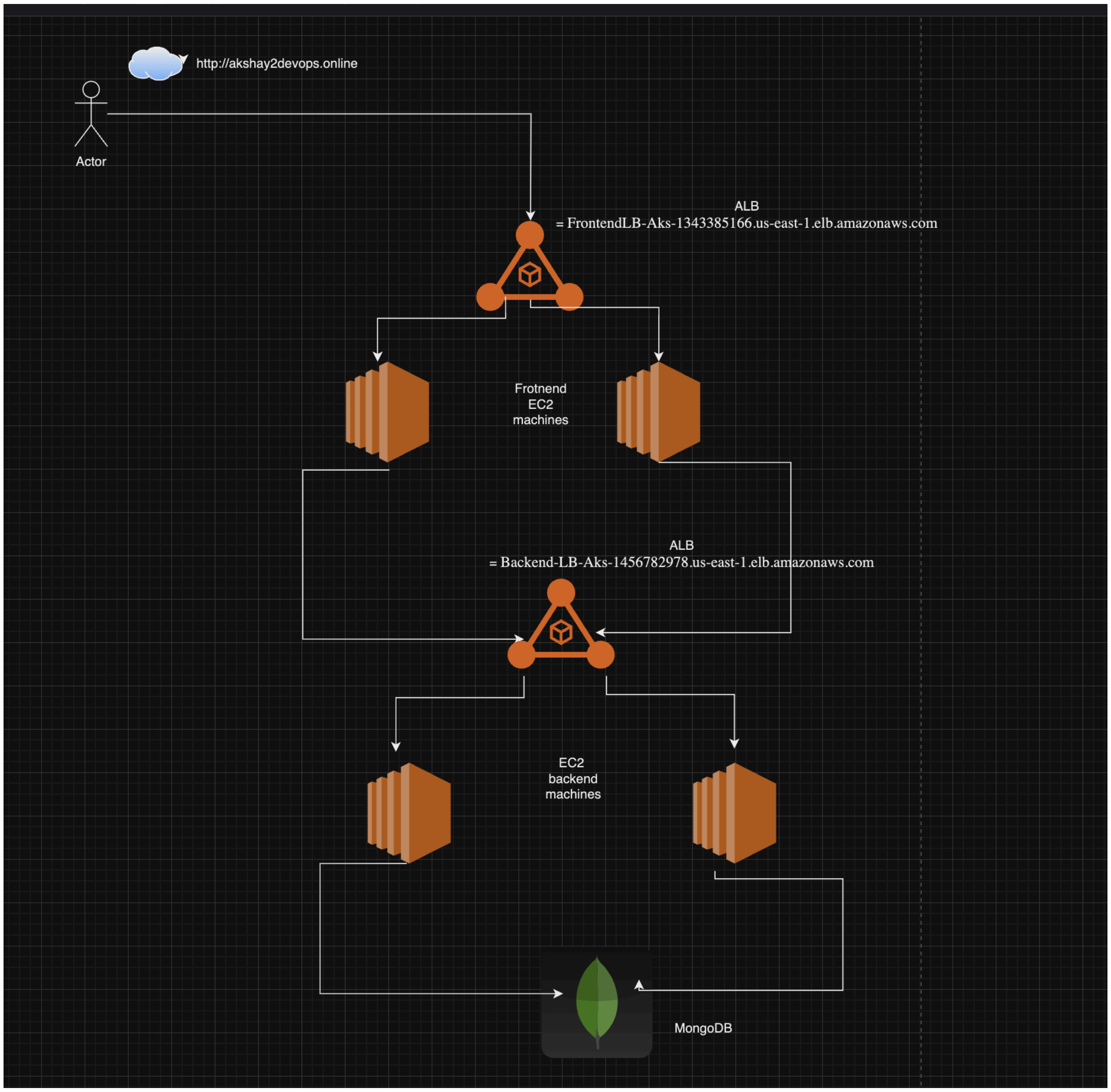
Meghalaya
leisure
Meghalaya, shillong, dawki
More Details

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


http://akshay2devops.online
Actor
ALB
= FrontendLB-Aks-1343385166.us-east-1.elb.amazonaws.com
Frotnend
EC2
machines
ALB
= Backend-LB-Aks-1456782978.us-east-1.elb.amazonaws.com
EC2
backend
machines
MongoDB