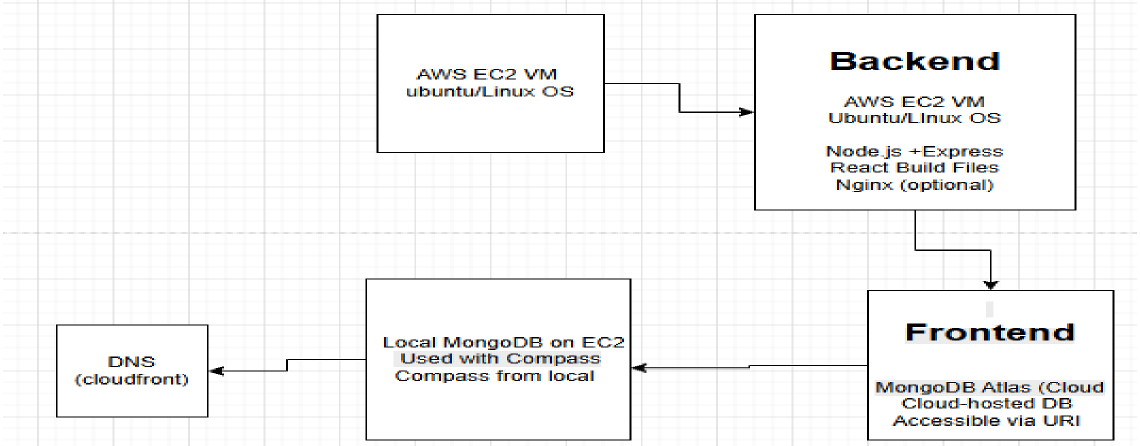
Project about deployment of MERN stack application on EC2

In this project, we will learn out EC2 instances, security groups,Load balancer , target groups React, node.js

Tools Covered:

- Linux
- Git and GitHub
- Mongodb Atlas
- MongoDB compass

project architecture diagram and explanation for deploying a MERN stack application



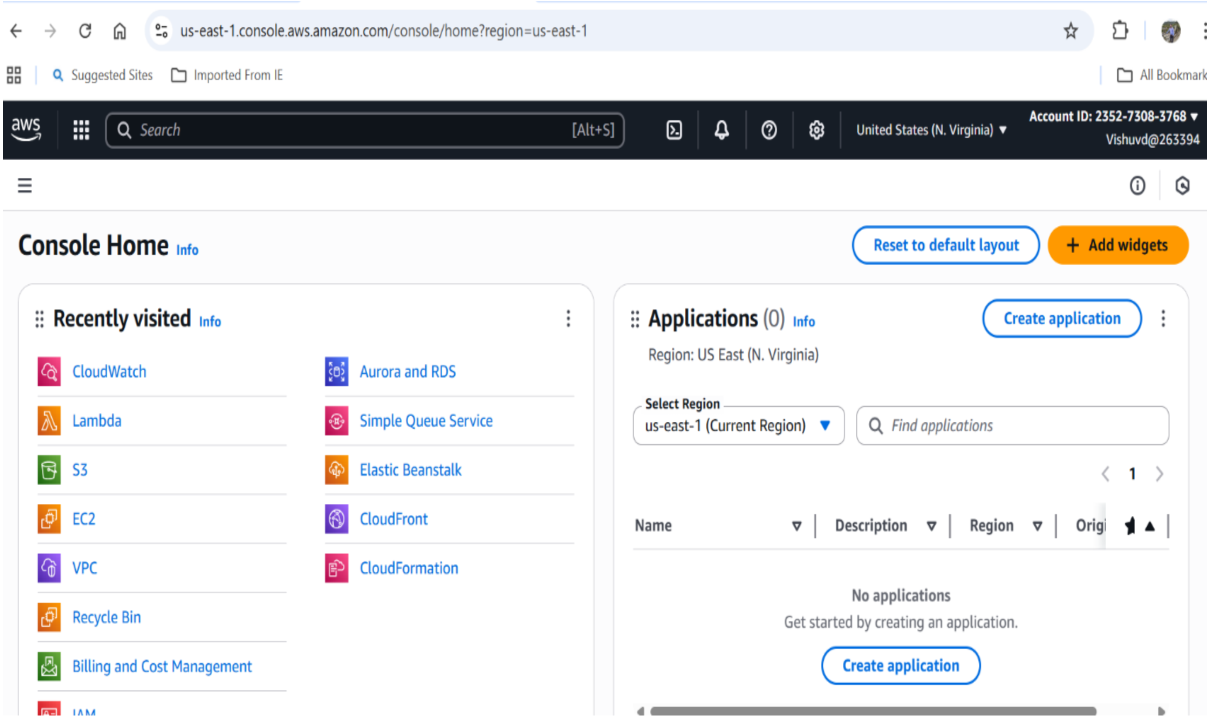
Pre-requisites to implement this project:

- An AWS account
- Basic understanding of cloud computing
- Ec2 instances
- Security groups
- S3 Bucket
- Load balancer
- cloudfront

- Familiarity with the MERN stack
- Access to AWS Management Console

Step by step launching AWS EC2 instances

1. Open AWS console

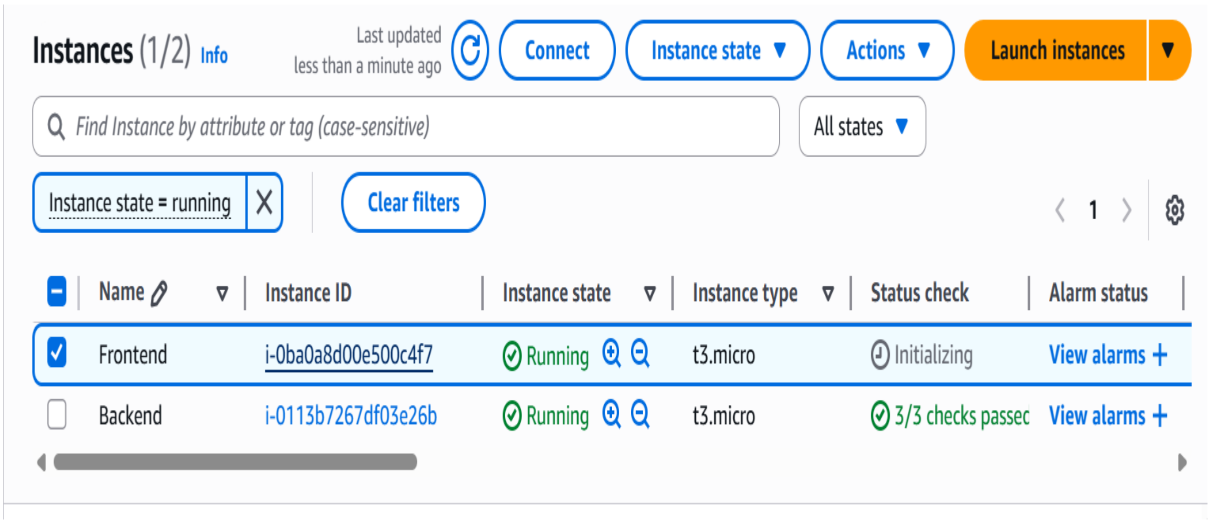


us-east-1.console.aws.amazon.com/console/home?region=us-east-1

Account ID: 2352-7308-3768

Vishuvd@263394

2. Launch EC2 instance for frontend and Backend using an Ubuntu image in the default VPC (Virtual Private Cloud)



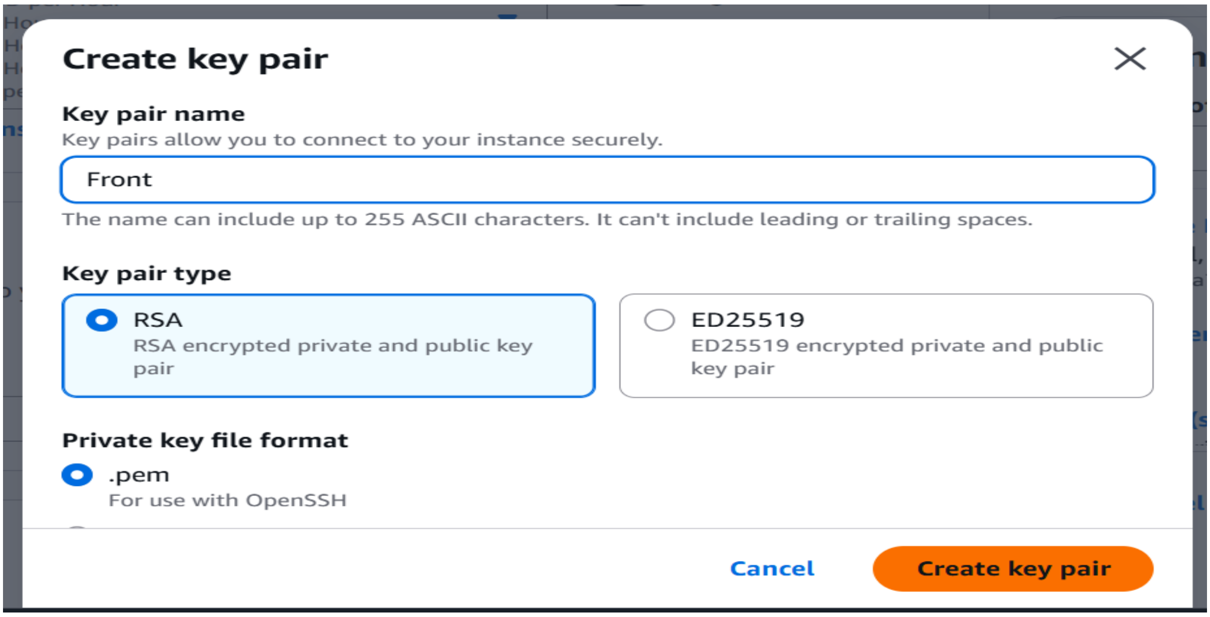
Instances (1/2) Info

Last updated less than a minute ago

Instance state = running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☑	Frontend	i-0ba0a8d00e500c4f7	Running	t3.micro	Initializing	View alarms +
□	Backend	i-0113b7267df03e26b	Running	t3.micro	3/3 checks passec	View alarms +

3. Create a key pair for both instances



Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

Front

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

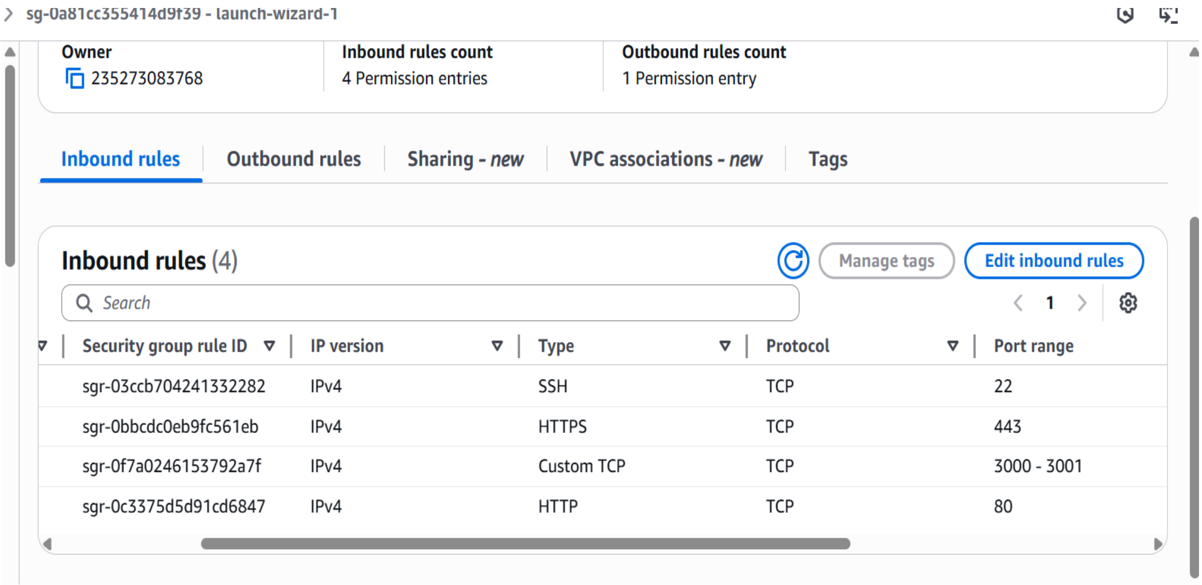
● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair

Private key file format

● .pem
For use with OpenSSH

4. Create a security group and configure port for both instances



sg-0a81cc355414d9f39 - launch-wizard-1

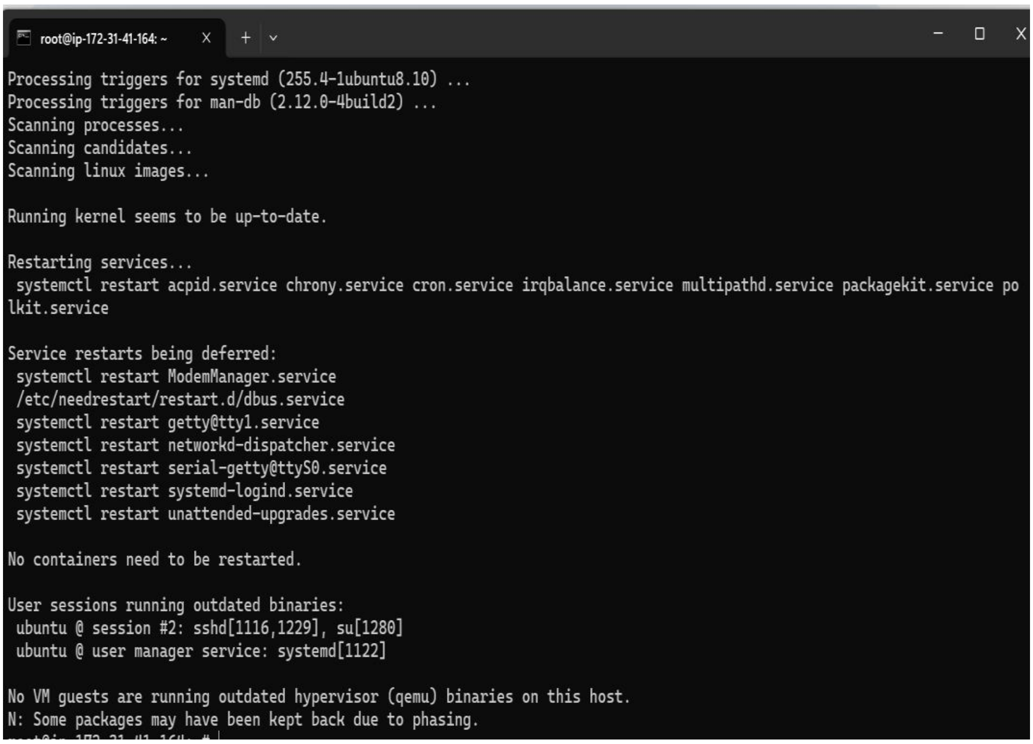
Owner	Inbound rules count	Outbound rules count
235273083768	4 Permission entries	1 Permission entry

Inbound rules (4)

Security group rule ID	IP version	Type	Protocol	Port range
sgr-03ccb704241332282	IPv4	SSH	TCP	22
sgr-0bbcdc0eb9fc561eb	IPv4	HTTPS	TCP	443
sgr-0f7a0246153792a7f	IPv4	Custom TCP	TCP	3000 - 3001
sgr-0c3375d5d91cd6847	IPv4	HTTP	TCP	80

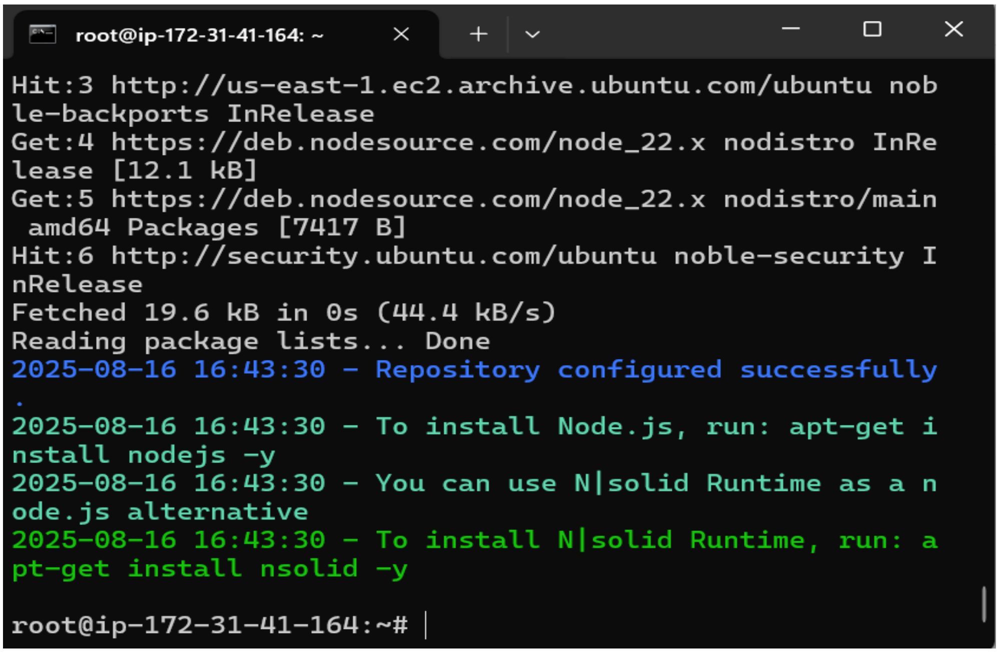
5. Connect Backend Ec2 instance Via SSH

- Update EC2 instance



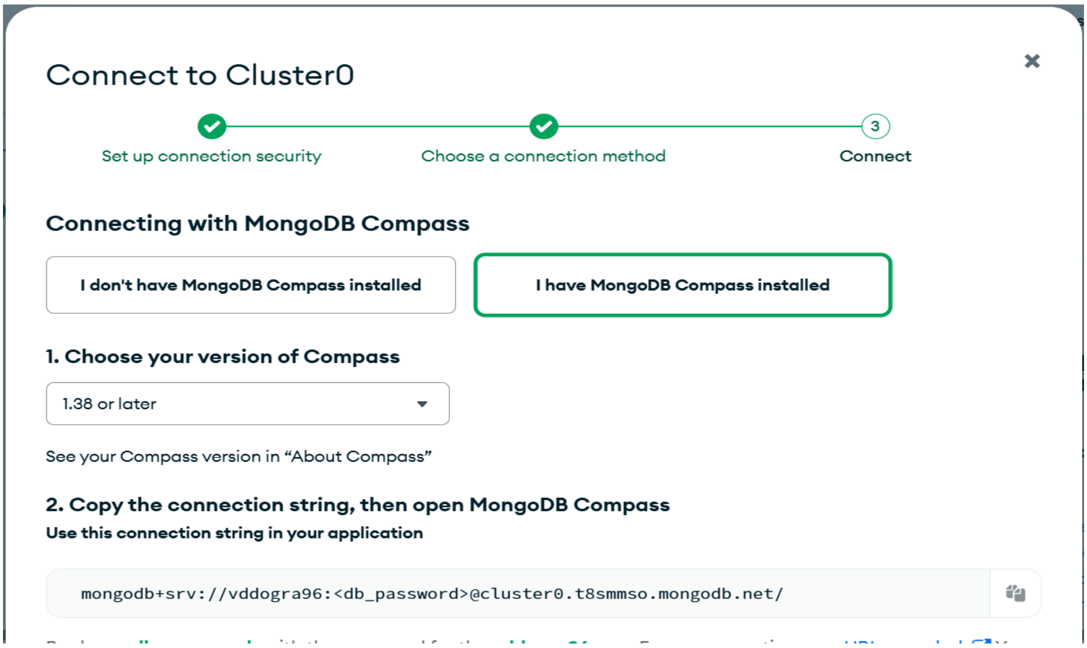
```
Processing triggers for systemd (255.4-1ubuntu8.10) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...
 systemctl restart acpid.service chrony.service cron.service irqbalance.service multipathd.service packagekit.service po
lkit.service

Service restarts being deferred:
 systemctl restart ModemManager.service
 /etc/needrestart/restart.d/dbus.service
 systemctl restart getty@tty1.service
 systemctl restart networkd-dispatcher.service
 systemctl restart serial-getty@ttyS0.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #2: sshd[1116,1229], su[1280]
 ubuntu @ user manager service: systemd[1122]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
N: Some packages may have been kept back due to phasing.
```

- Install **Installing Node.js**



```
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu nob
le-backports InRelease
Get:4 https://deb.nodesource.com/node_22.x nodistro InRe
lease [12.1 kB]
Get:5 https://deb.nodesource.com/node_22.x nodistro/main
 amd64 Packages [7417 B]
Hit:6 http://security.ubuntu.com/ubuntu noble-security I
nRelease
Fetched 19.6 kB in 0s (44.4 kB/s)
Reading package lists... Done
2025-08-16 16:43:30 - Repository configured successfully
.
2025-08-16 16:43:30 - To install Node.js, run: apt-get i
nstall nodejs -y
2025-08-16 16:43:30 - You can use N|solid Runtime as a n
ode.js alternative
2025-08-16 16:43:30 - To install N|solid Runtime, run: a
pt-get install nsolid -y

root@ip-172-31-41-164:~#
```

6. **Create a Cluster on MongoDB Atlas and Save Your Cluster Connection URI with password**
    - **Allow access access from every where**



Connect to Cluster0

✕
✓ Set up connection security
✓ Choose a connection method
3 Connect

Connecting with MongoDB Compass

I don't have MongoDB Compass installed
I have MongoDB Compass installed

1. Choose your version of Compass

1.38 or later ▾

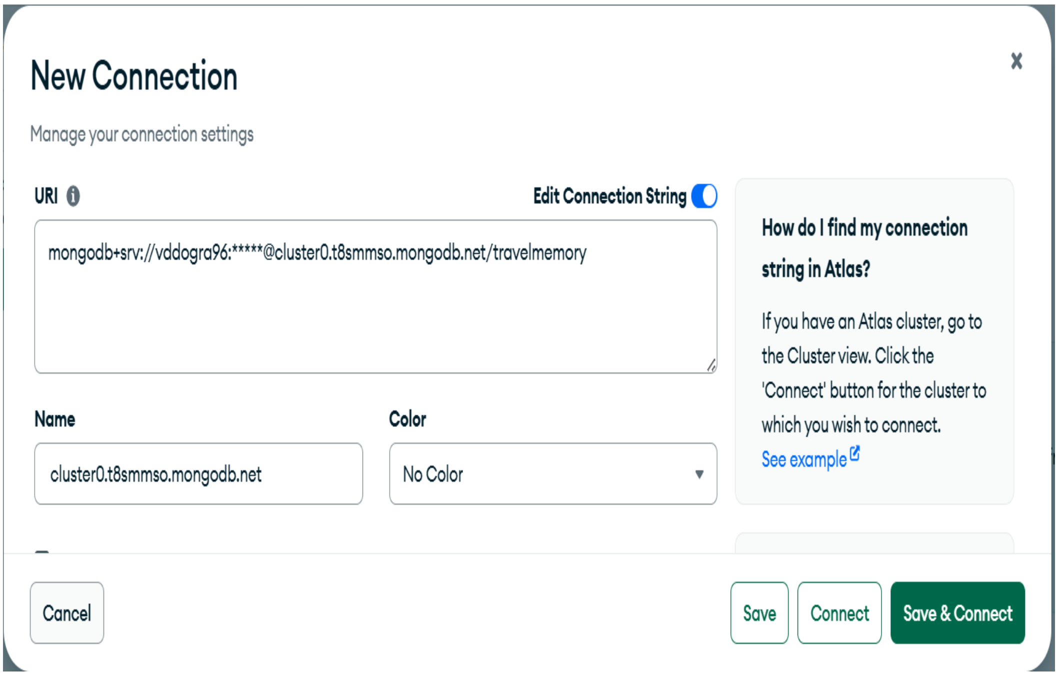
See your Compass version in “About Compass”

2. Copy the connection string, then open MongoDB Compass

Use this connection string in your application

```
mongodb+srv://vddogra96:<db_password>@cluster0.t8smmso.mongodb.net/
```

7. **Download and install MongoDB Compass**
    - **Connect to cluster using URI**
    - **Create a database**
    - **Create 'travelmemory' database and collections**



New Connection

Manage your connection settings

URI ⓘ

Edit Connection String ●

```
mongodb+srv://vddogra96:*****@cluster0.t8smmso.mongodb.net/travelmemory
```

How do I find my connection string in Atlas?

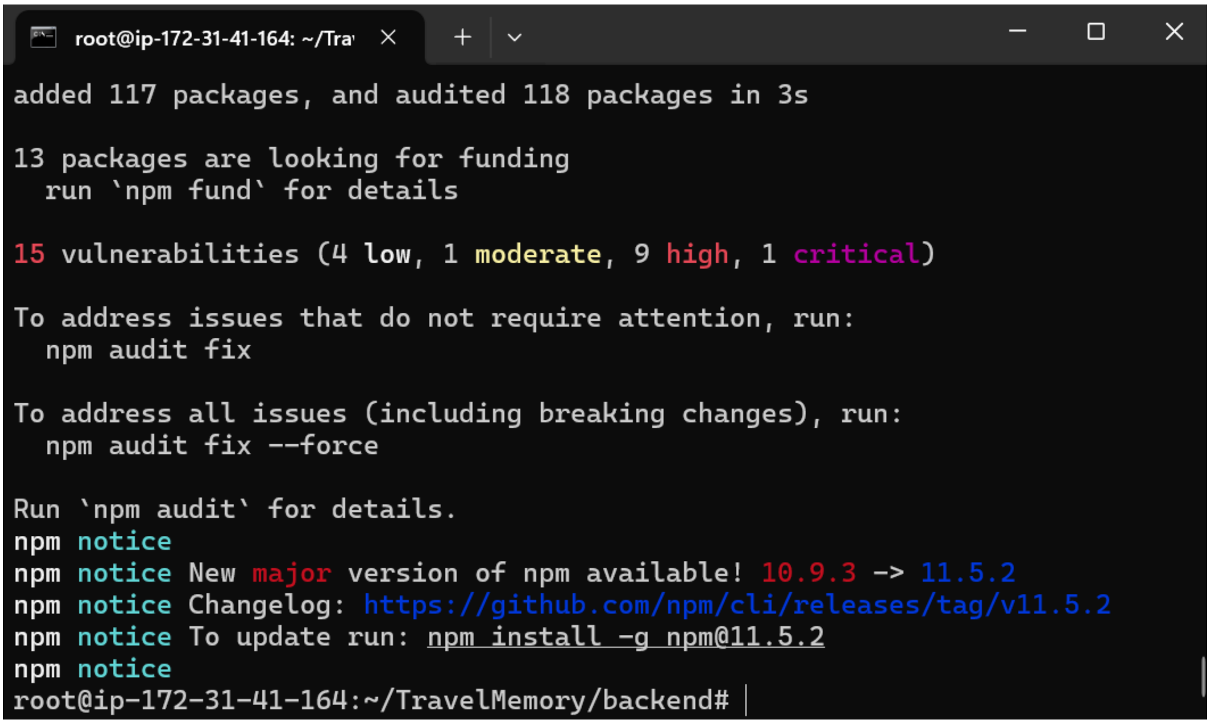
If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.

See example

Name: cluster0.t8smmso.mongodb.net

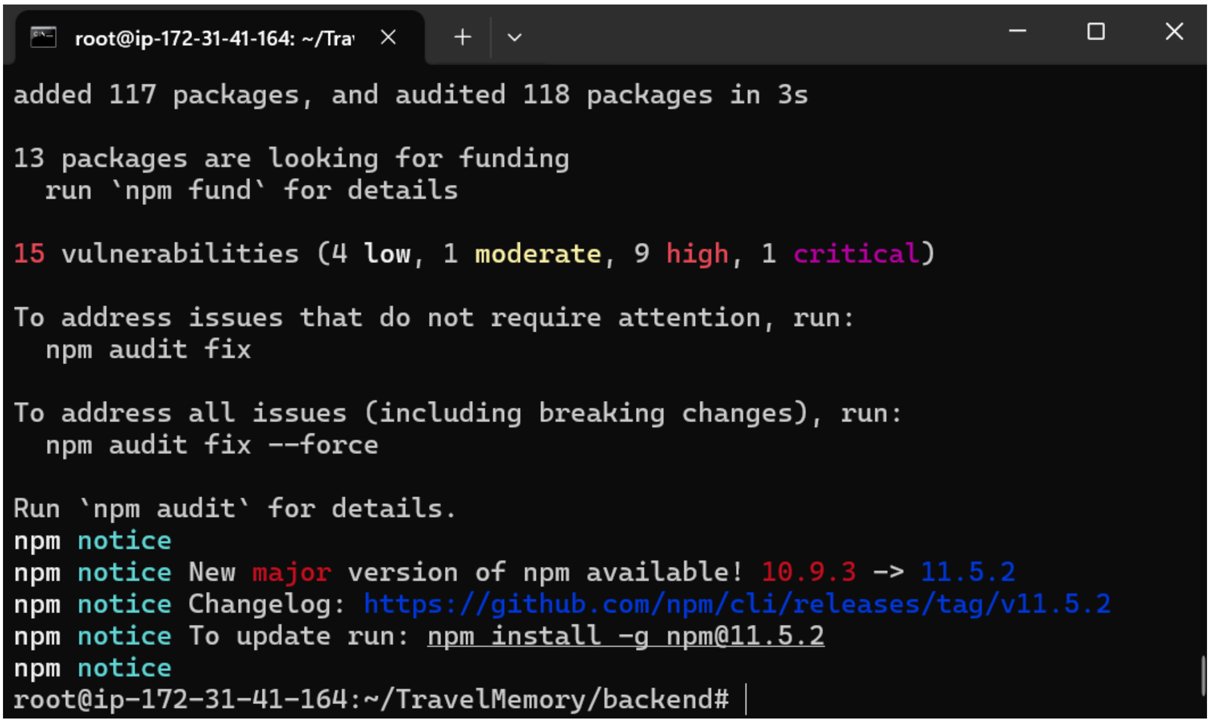
Color: No Color ▾

8 . Deploying a Node.js Application on Backend EC2 and Cloning the Repository of Travel memory



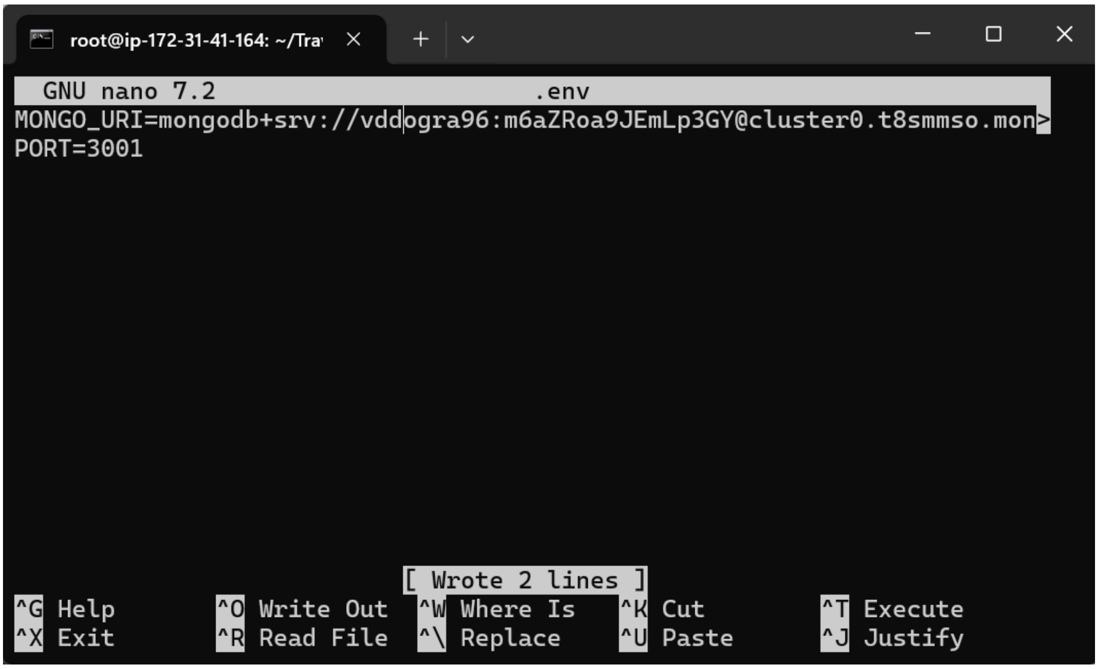
```
added 117 packages, and audited 118 packages in 3s

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

15 vulnerabilities (4 low, 1 moderate, 9 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.
npm notice
npm notice New major version of npm available! 10.9.3 -> 11.5.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.5.2
npm notice To update run: npm install -g npm@11.5.2
npm notice
root@ip-172-31-41-164:~/TravelMemory/backend#
```

8. clone the GitHub repository to your backend



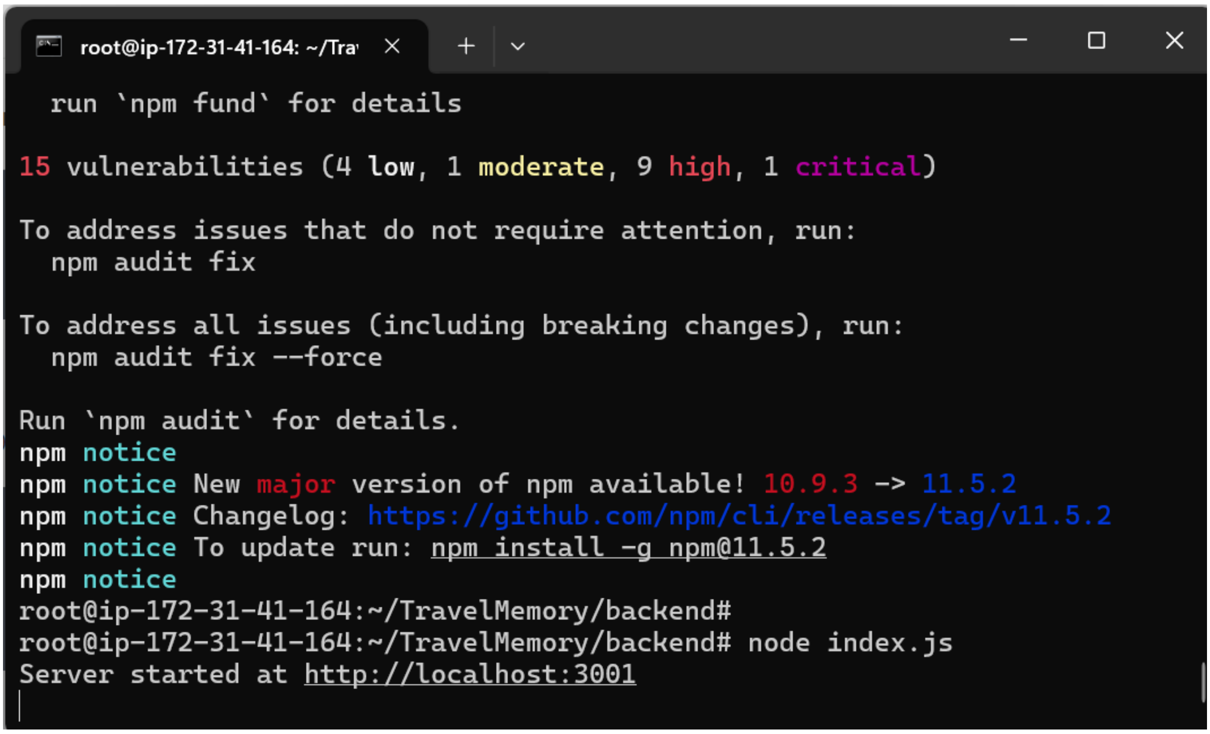
```
added 117 packages, and audited 118 packages in 3s

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

15 vulnerabilities (4 low, 1 moderate, 9 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.
npm notice
npm notice New major version of npm available! 10.9.3 -> 11.5.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.5.2
npm notice To update run: npm install -g npm@11.5.2
npm notice
root@ip-172-31-41-164:~/TravelMemory/backend#
```

9. Create a .env file in the backend directory to store your environment variables. Add the following content to your .env file, replacing 'ENTER_YOUR_URL' with your MongoDB URI from Atlas:
   - Also define the port no. 3001



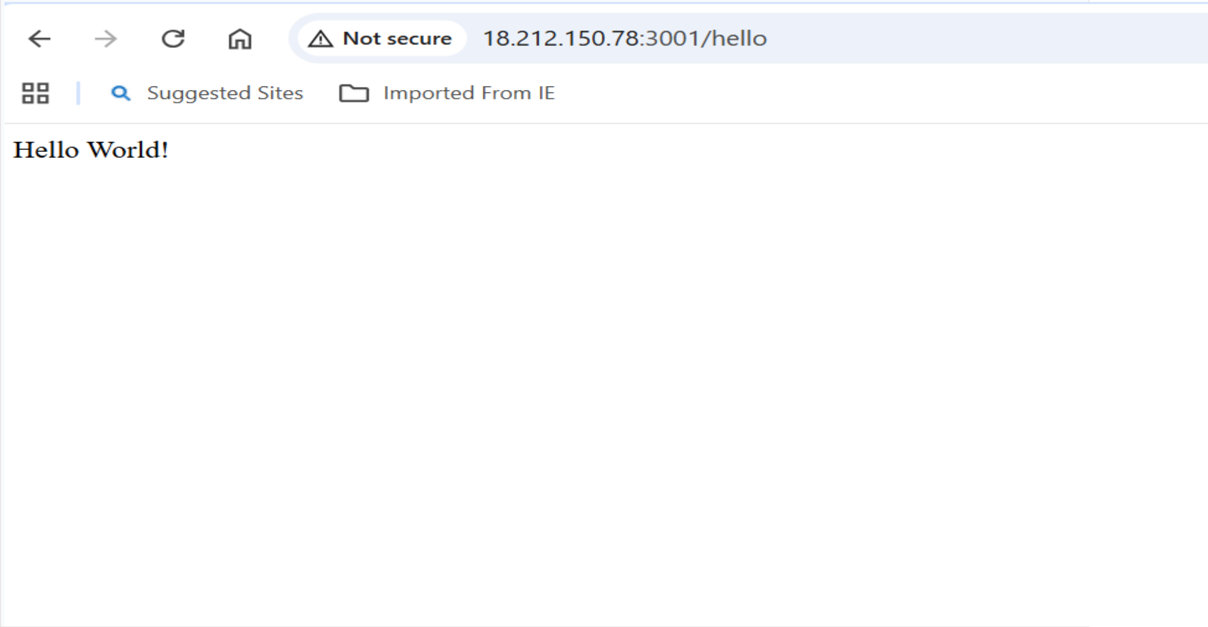
```
  GNU nano 7.2                              .env
MONGO_URI=mongodb+srv://vddogra96:m6aZRoa9JEmLp3GY@cluster0.t8smmso.mon>
PORT=3001





                                                  [ Wrote 2 lines ]
^G Help          ^O Write Out    ^W Where Is     ^K Cut          ^T Execute
^X Exit          ^R Read File    ^\ Replace      ^U Paste        ^J Justify
```

10. Install packages npm and run the backend application through node index.js command



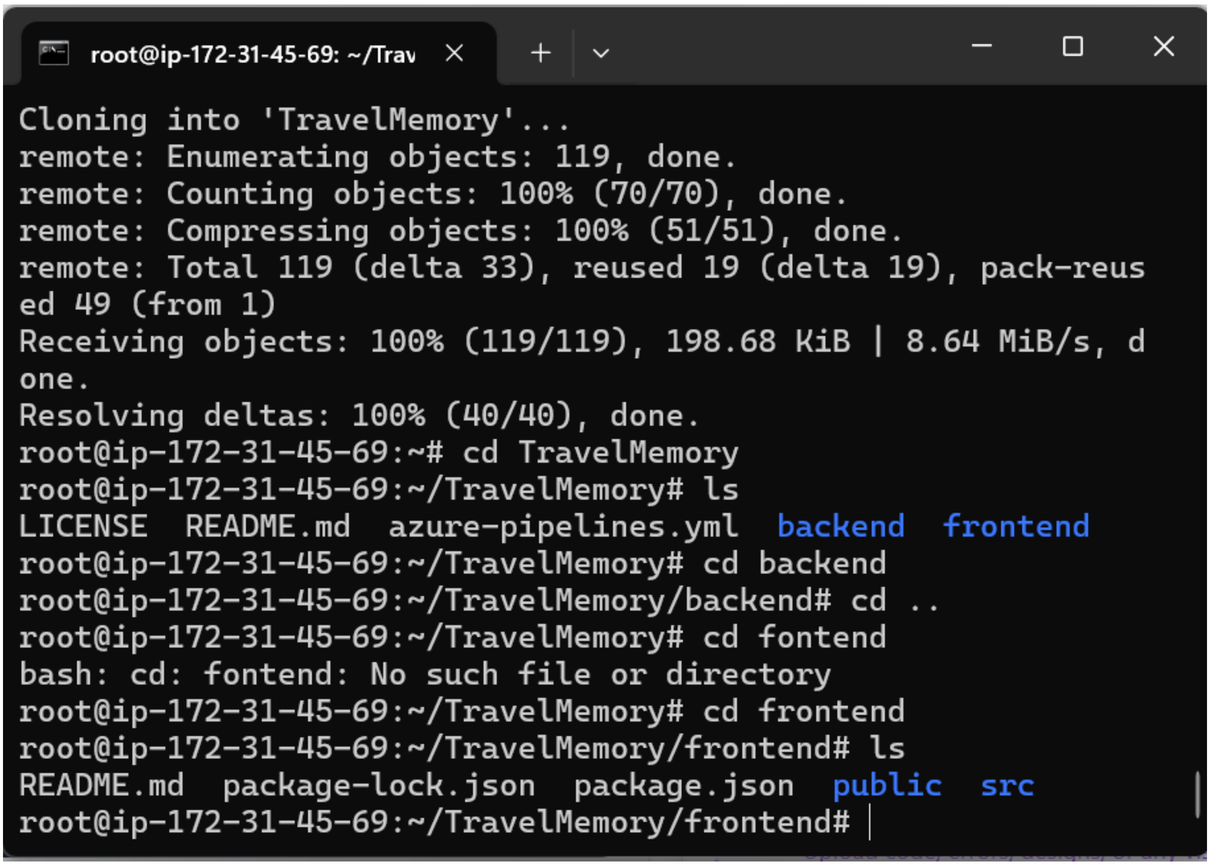
```
  run `npm fund` for details

15 vulnerabilities (4 low, 1 moderate, 9 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.
npm notice
npm notice New major version of npm available! 10.9.3 -> 11.5.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.5.2
npm notice To update run: npm install -g npm@11.5.2
npm notice
root@ip-172-31-41-164:~/TravelMemory/backend#
root@ip-172-31-41-164:~/TravelMemory/backend# node index.js
Server started at http://localhost:3001
```

11. **Backend is running successfully, we can navigate though our Backend EC2 IP address followed by :3001/hello**



Hello World!

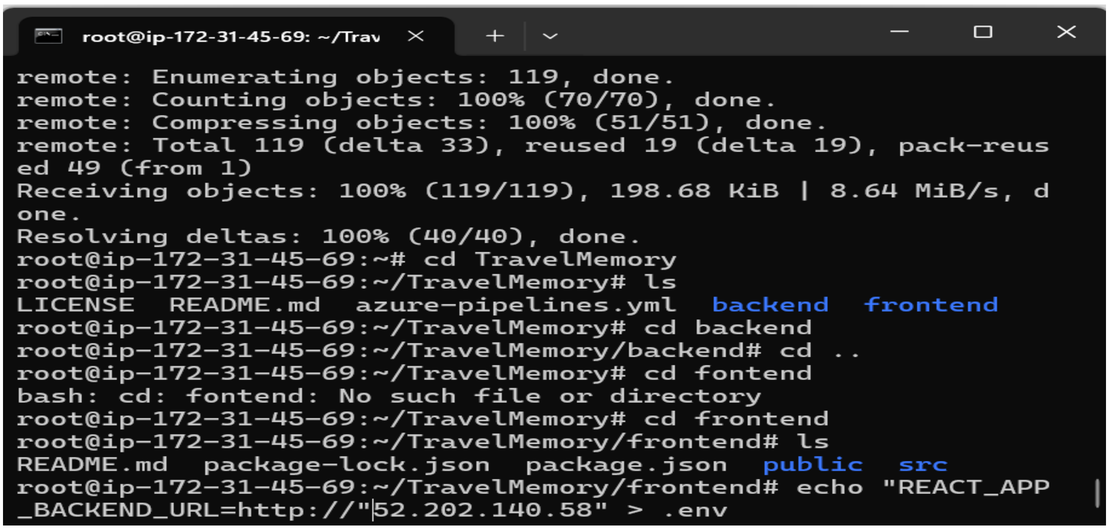
12. **Connect SSH to Frontend Your EC2 Instance ,update EC2 instance and clone the Github Repository**



```
Cloning into 'TravelMemory'...
remote: Enumerating objects: 119, done.
remote: Counting objects: 100% (70/70), done.
remote: Compressing objects: 100% (51/51), done.
remote: Total 119 (delta 33), reused 19 (delta 19), pack-reus
ed 49 (from 1)
Receiving objects: 100% (119/119), 198.68 KiB | 8.64 MiB/s, d
one.
Resolving deltas: 100% (40/40), done.
root@ip-172-31-45-69:~# cd TravelMemory
root@ip-172-31-45-69:~/TravelMemory# ls
LICENSE  README.md  azure-pipelines.yml  backend  frontend
root@ip-172-31-45-69:~/TravelMemory# cd backend
root@ip-172-31-45-69:~/TravelMemory/backend# cd ..
root@ip-172-31-45-69:~/TravelMemory# cd fontend
bash: cd: fontend: No such file or directory
root@ip-172-31-45-69:~/TravelMemory# cd frontend
root@ip-172-31-45-69:~/TravelMemory/frontend# ls
README.md  package-lock.json  package.json  public  src
root@ip-172-31-45-69:~/TravelMemory/frontend#
```

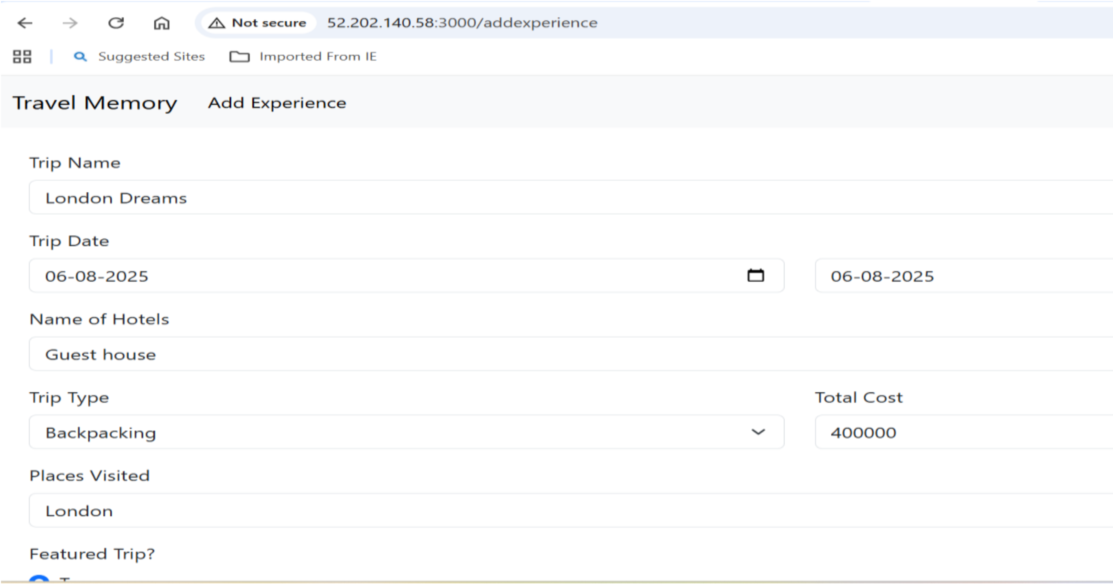
13. **Create the .env File and this file will contain the backend URL that your React app will use to connect to the server.**

```
echo "REACT_APP_BACKEND_URL=http://EC2_PUBLIC_IP:3001" > .env
```



```
remote: Enumerating objects: 119, done.
remote: Counting objects: 100% (70/70), done.
remote: Compressing objects: 100% (51/51), done.
remote: Total 119 (delta 33), reused 19 (delta 19), pack-reus
ed 49 (from 1)
Receiving objects: 100% (119/119), 198.68 KiB | 8.64 MiB/s, d
one.
Resolving deltas: 100% (40/40), done.
root@ip-172-31-45-69:~# cd TravelMemory
root@ip-172-31-45-69:~/TravelMemory# ls
LICENSE  README.md  azure-pipelines.yml  backend  frontend
root@ip-172-31-45-69:~/TravelMemory# cd backend
root@ip-172-31-45-69:~/TravelMemory/backend# cd ..
root@ip-172-31-45-69:~/TravelMemory# cd fontend
bash: cd: fontend: No such file or directory
root@ip-172-31-45-69:~/TravelMemory# cd frontend
root@ip-172-31-45-69:~/TravelMemory/frontend# ls
README.md  package-lock.json  package.json  public  src
root@ip-172-31-45-69:~/TravelMemory/frontend# echo "REACT_APP
_BACKEND_URL=http://"52.202.140.58" > .env
```

14. **Install the packages of npm through CMD npm install and then npm start and after start server . paste Public IP address of Frontend EC2 on port no. 3000 and access your deployed React application `http://EC2_PUBLIC_IP:3000`**



Trip Name

London Dreams

Trip Date

06-08-2025 | 06-08-2025

Name of Hotels

Guest house

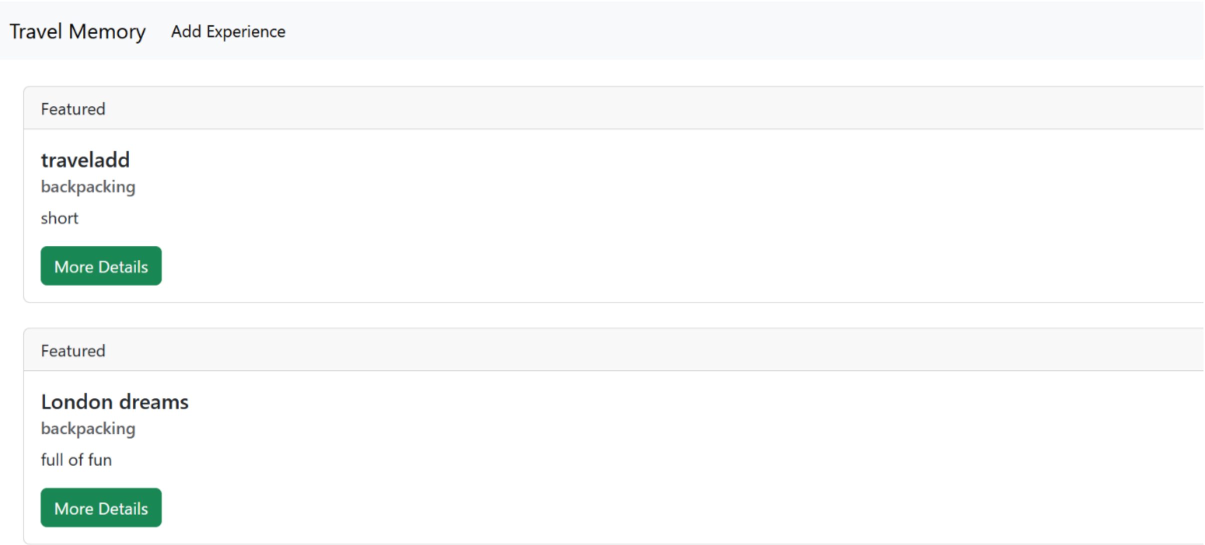
Trip Type: Backpacking

Total Cost: 400000

Places Visited

London

Featured Trip?



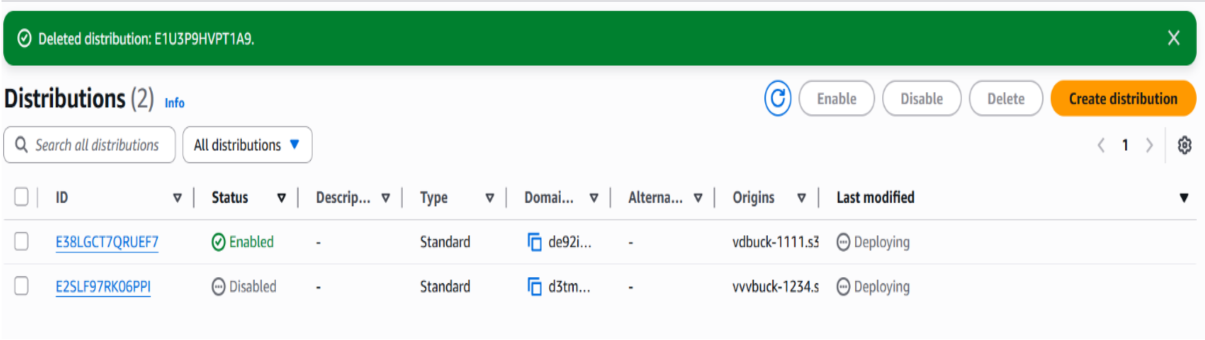
Featured

traveladd
backpacking
short

Featured

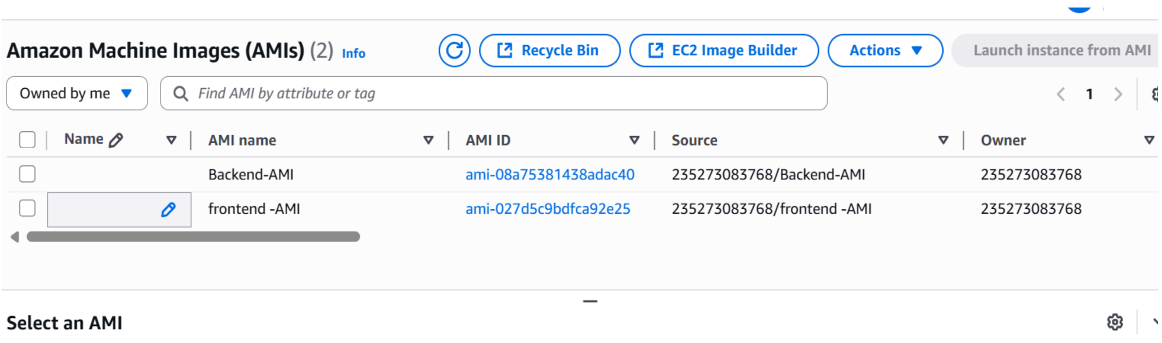
London dreams
backpacking
full of fun

15. Create a Cloud front distribution and add s3 bucket


Deleted distribution: E1U3P9HVPT1A9.

☐	ID	Status	Descrip...	Type	Domai...	Alterna...	Origins	Last modified
☐	E38LGCT7QRUEF7	✓ Enabled	-	Standard	de92i...	-	vdbuck-1111.s3	Deploying
☐	E2SLF97RK06PPI	Disabled	-	Standard	d3tm...	-	vvvbuck-1234.s	Deploying

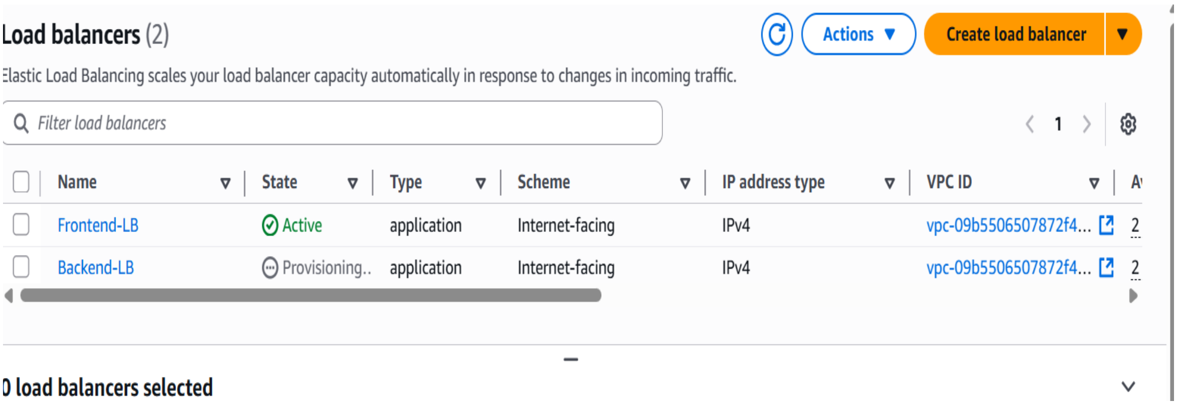
16. Create AMI's of both Backend and Frontend instance



☐	Name	AMI name	AMI ID	Source	Owner
☐		Backend-AMI	ami-08a75381438adac40	235273083768/Backend-AMI	235273083768
☐		frontend -AMI	ami-027d5c9bdfca92e25	235273083768/frontend -AMI	235273083768

Select an AMI

17. Create backend and frontend Load balancer

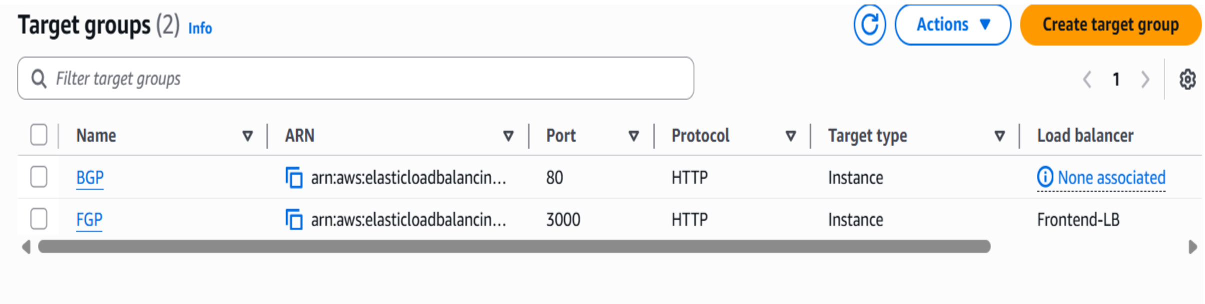


Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

☐	Name	State	Type	Scheme	IP address type	VPC ID	A
☐	Frontend-LB	✓ Active	application	Internet-facing	IPv4	vpc-09b5506507872f4...	2
☐	Backend-LB	Provisioning..	application	Internet-facing	IPv4	vpc-09b5506507872f4...	2

0 load balancers selected

18 reate a target groups for Frontend load balancer and backend load balancer



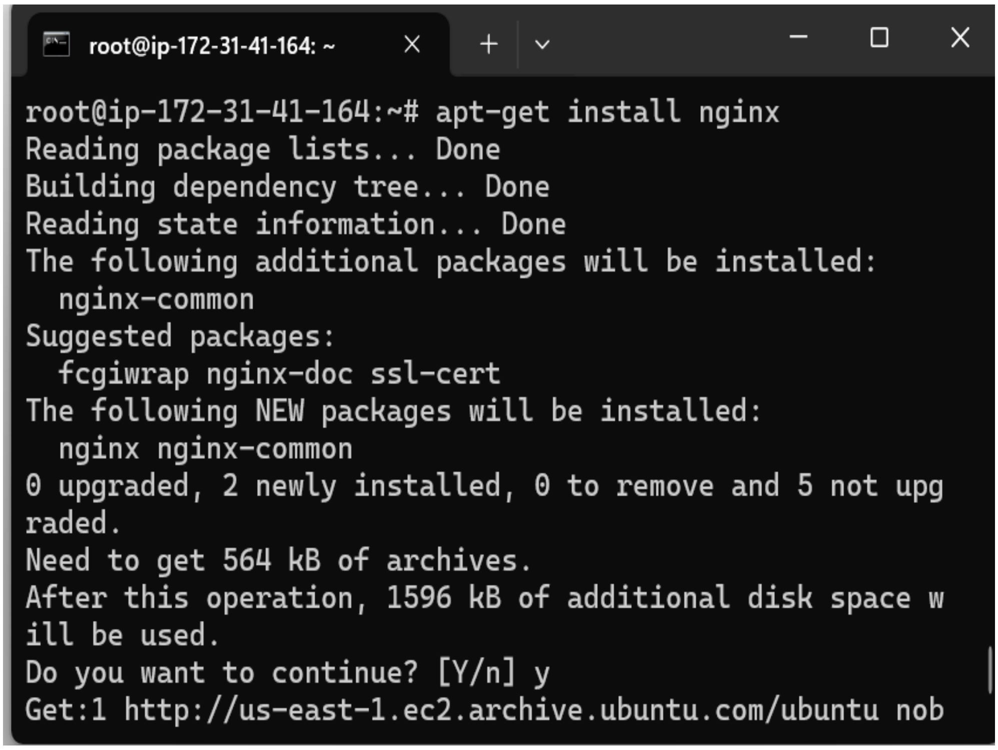
Target groups (2) Info

Actions ▼ | Create target group

Filter target groups

	Name	ARN	Port	Protocol	Target type	Load balancer
□	BGP	arn:aws:elasticloadbalancin...	80	HTTP	Instance	None associated
□	FGP	arn:aws:elasticloadbalancin...	3000	HTTP	Instance	Frontend-LB

19.Install the ningx og Ec2 instances chances the file in site-availables



root@ip-172-31-41-164: ~

```
root@ip-172-31-41-164:~# apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 5 not upg
raded.
Need to get 564 kB of archives.
After this operation, 1596 kB of additional disk space w
ill be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu nob
```

18. Purchase a domain from Cloudfare and add the in C Name group and add the DNS and then deploy on browser with DNS