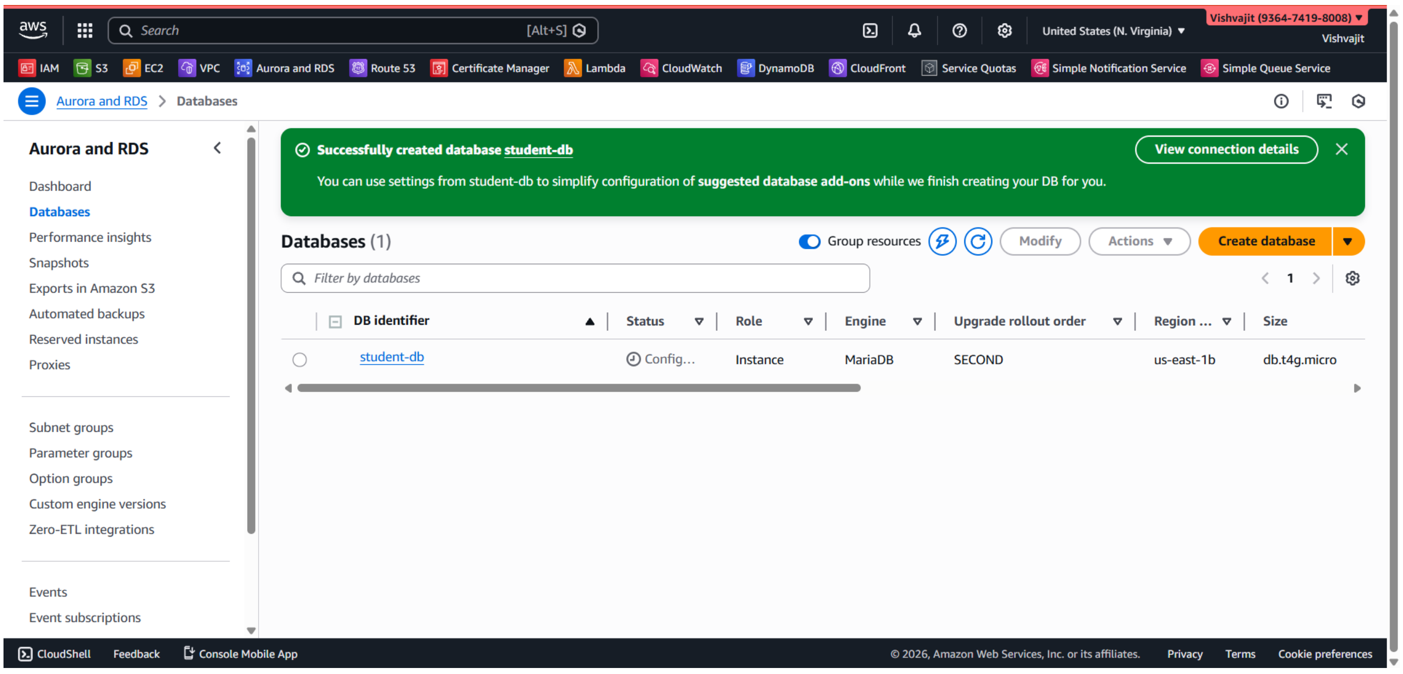
Task

Application deploy using load balancer and auto scaling

First, I need a database that store the user details so I created a student-db using RDS service in that I choose mariadb database engine


aws
Search [Alt+S]
United States (N. Virginia)
Vishvajit (9364-7419-8008)
Vishvajit
IAM S3 EC2 VPC Aurora and RDS Route 53 Certificate Manager Lambda CloudWatch DynamoDB CloudFront Service Quotas Simple Notification Service Simple Queue Service
Aurora and RDS > Databases
Aurora and RDS
Dashboard
Databases
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations
Events
Event subscriptions
View connection details
Group resources
Modify
Actions
Create database
Filter by databases
CloudShell Feedback Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

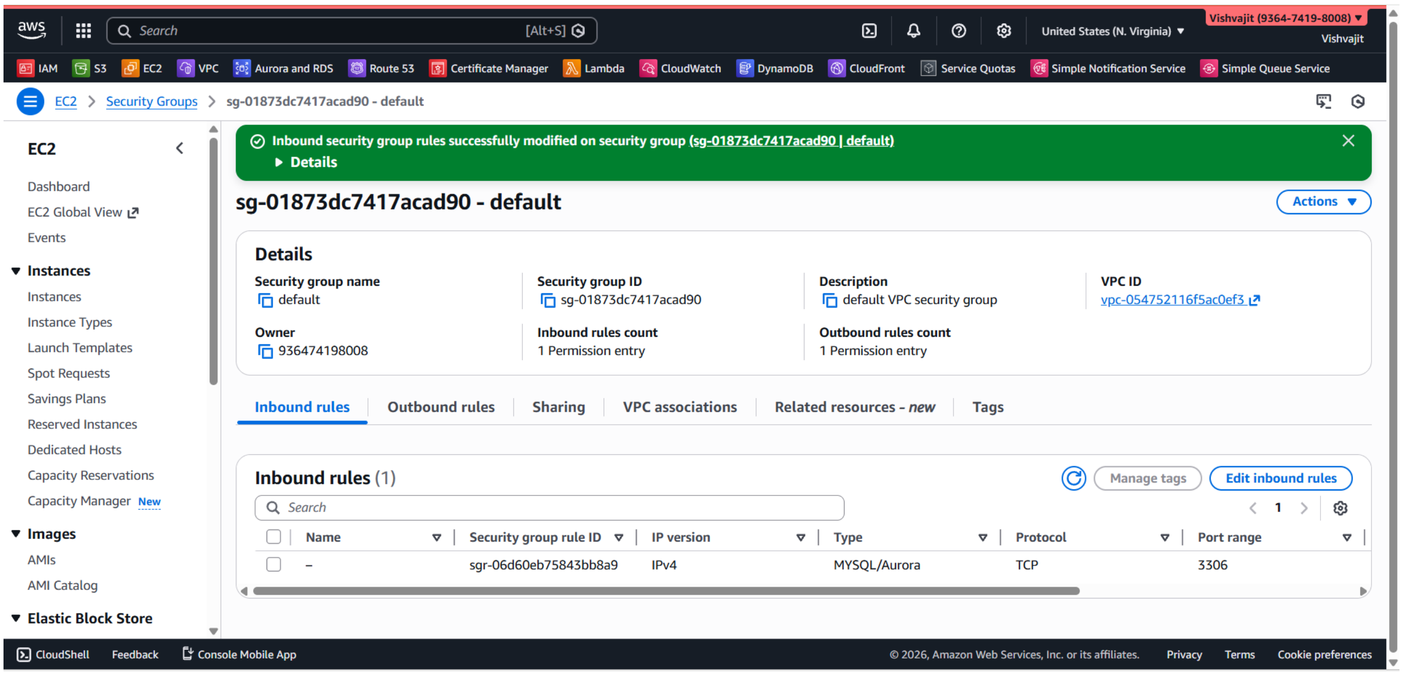
Successfully created database student-db

You can use settings from student-db to simplify configuration of **suggested database add-ons** while we finish creating your DB for you.

Databases (1)

DB identifier	Status	Role	Engine	Upgrade rollout order	Region ...	Size
student-db	Config...	Instance	MariaDB	SECOND	us-east-1b	db.t4g.micro

After creating the database we need to allow 3306 port in the security group


aws
Search [Alt+S]
United States (N. Virginia)
Vishvajit (9364-7419-8008)
Vishvajit
IAM S3 EC2 VPC Aurora and RDS Route 53 Certificate Manager Lambda CloudWatch DynamoDB CloudFront Service Quotas Simple Notification Service Simple Queue Service
EC2 > Security Groups > sg-01873dc7417acad90 - default
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Actions
Inbound rules Outbound rules Sharing VPC associations Related resources - new Tags
Manage tags
Edit inbound rules
Search
CloudShell Feedback Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Inbound security group rules successfully modified on security group (sg-01873dc7417acad90 | default)

Details

sg-01873dc7417acad90 - default

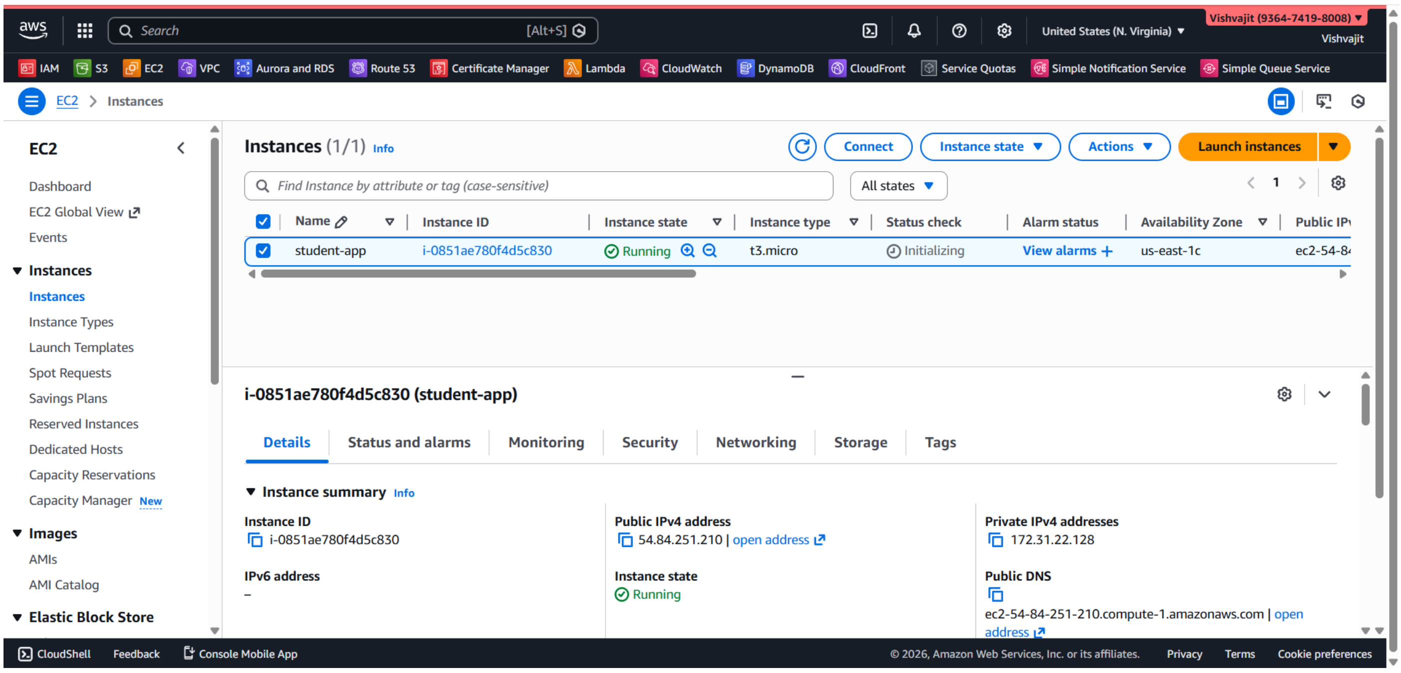
Details

Security group name: default	Security group ID: sg-01873dc7417acad90	Description: default VPC security group	VPC ID: vpc-054752116f5ac0ef3
Owner: 936474198008	Inbound rules count: 1 Permission entry	Outbound rules count: 1 Permission entry	

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range
–	sgr-06d60eb75843bb8a9	IPv4	MYSQL/Aurora	TCP	3306

Then I am going to create instance to host application and install docker



Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
student-app	i-0851ae780f4d5c830	Running	t3.micro	Initializing	View alarms +	us-east-1c	ec2-54-8

i-0851ae780f4d5c830 (student-app)

Instance summary Info

Instance ID: i-0851ae780f4d5c830

Public IPv4 address: 54.84.251.210 | open address

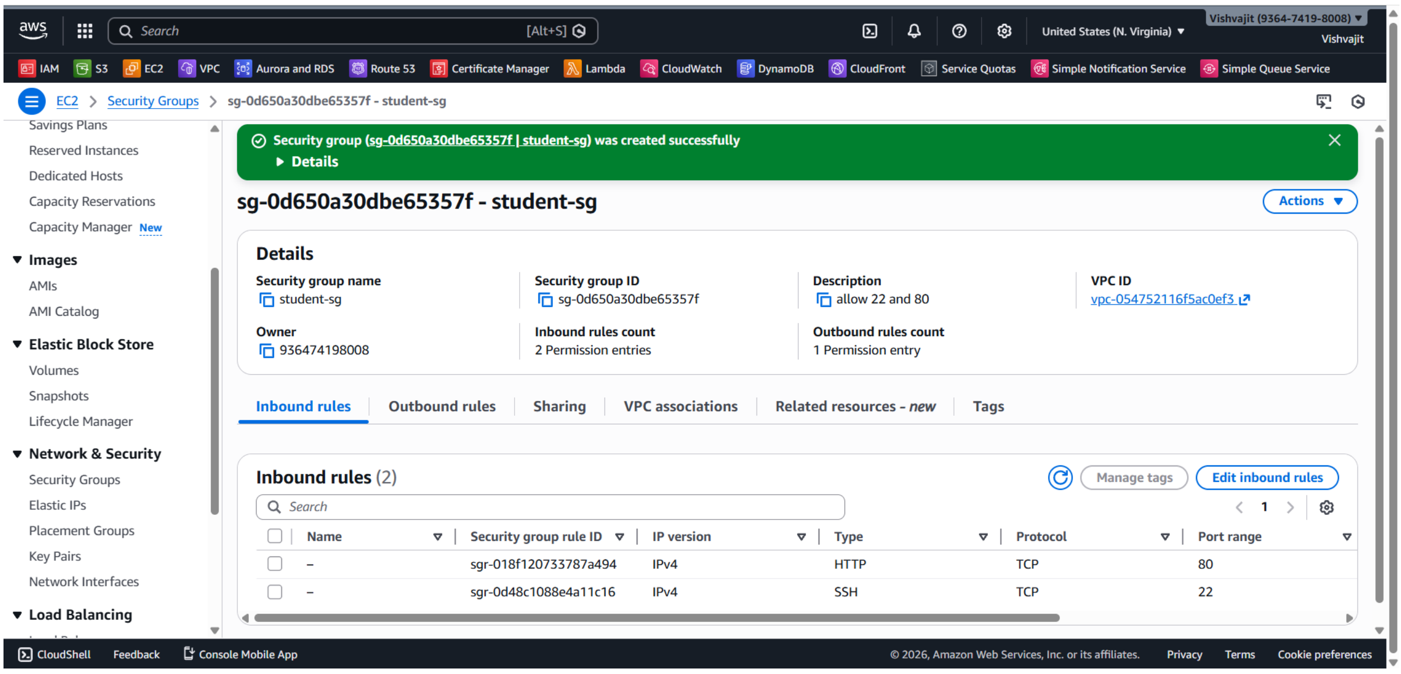
Private IPv4 addresses: 172.31.22.128

IPv6 address: –

Instance state: Running

Public DNS: ec2-54-84-251-210.compute-1.amazonaws.com | open address

After creating the instance i edit the security group and allow port 80 and 22 is allow by default.



Security group (sg-0d650a30dbe65357f | student-sg) was created successfully
▸ Details

sg-0d650a30dbe65357f - student-sg

Details

Security group name: student-sg

Security group ID: sg-0d650a30dbe65357f

Description: allow 22 and 80

VPC ID: vpc-054752116f5ac0ef3

Owner: 936474198008

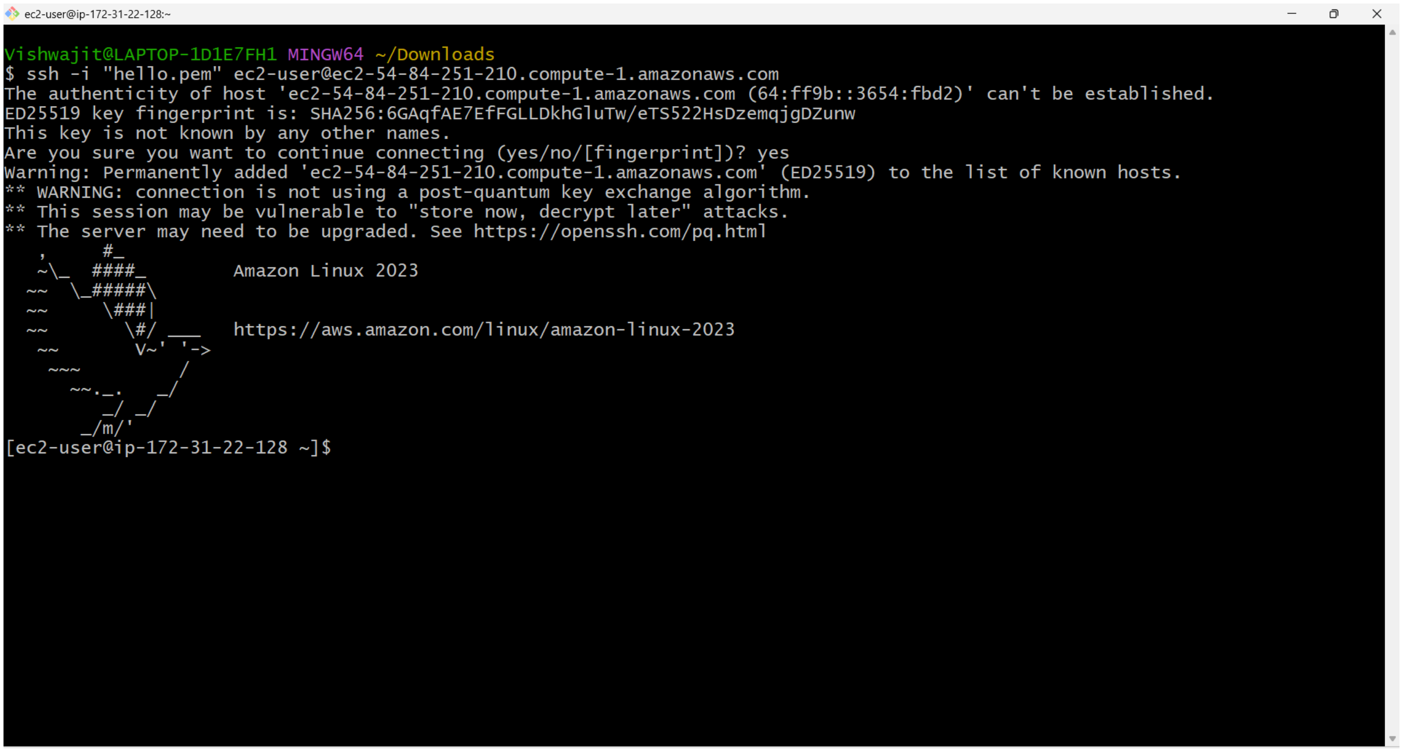
Inbound rules count: 2 Permission entries

Outbound rules count: 1 Permission entry

Inbound rules (2)

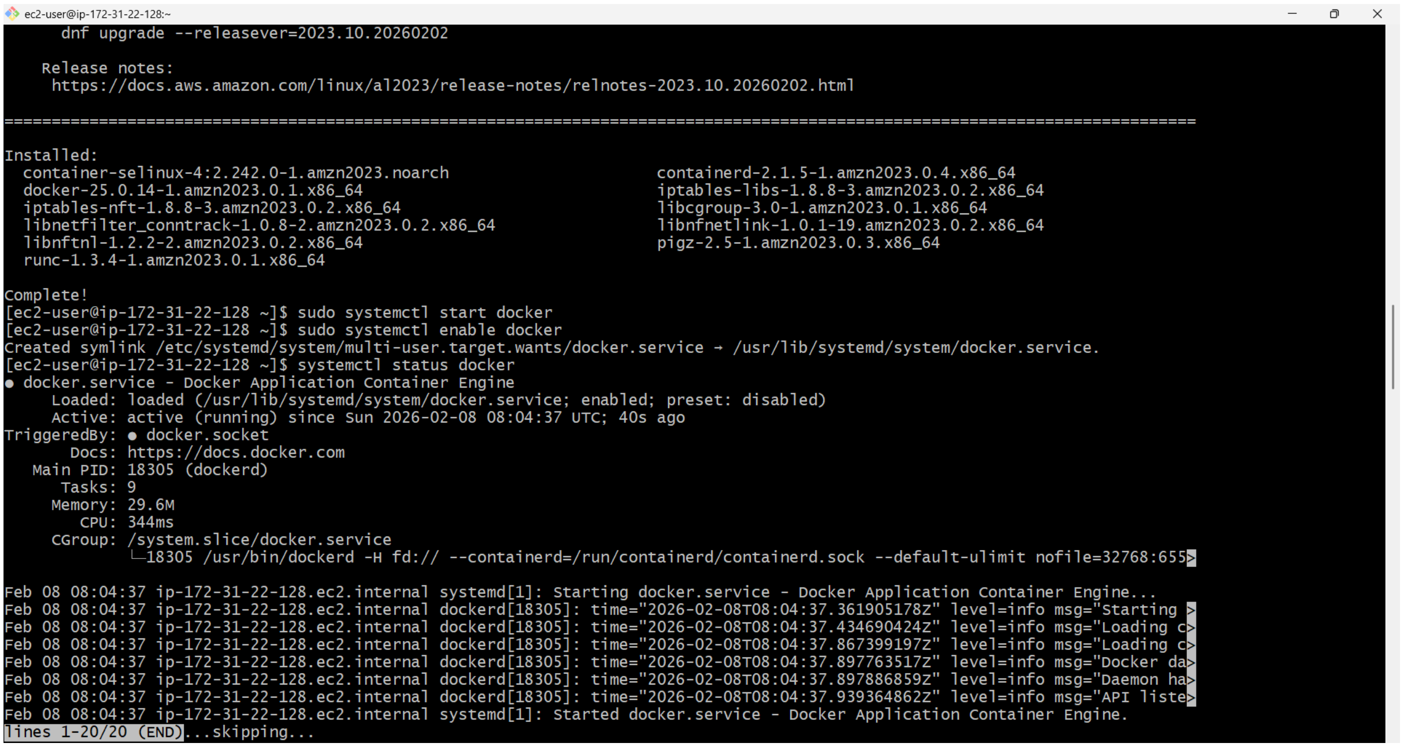
Name	Security group rule ID	IP version	Type	Protocol	Port range
–	sgr-018f120733787a494	IPv4	HTTP	TCP	80
–	sgr-0d48c1088e4a11c16	IPv4	SSH	TCP	22

Then I connect the ec2 instance using git tool.


ec2-user@ip-172-31-22-128:~

```
Vishwajit@LAPTOP-1D1E7FH1 MINGW64 ~/Downloads
$ ssh -i "hello.pem" ec2-user@ec2-54-84-251-210.compute-1.amazonaws.com
The authenticity of host 'ec2-54-84-251-210.compute-1.amazonaws.com (64:ff9b::3654:fbd2)' can't be established.
ED25519 key fingerprint is: SHA256:6GAqfAE7EfFGLLDkhGluTw/eTS522HsDzemqjgDZunw
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-84-251-210.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-22-128 ~]$
```

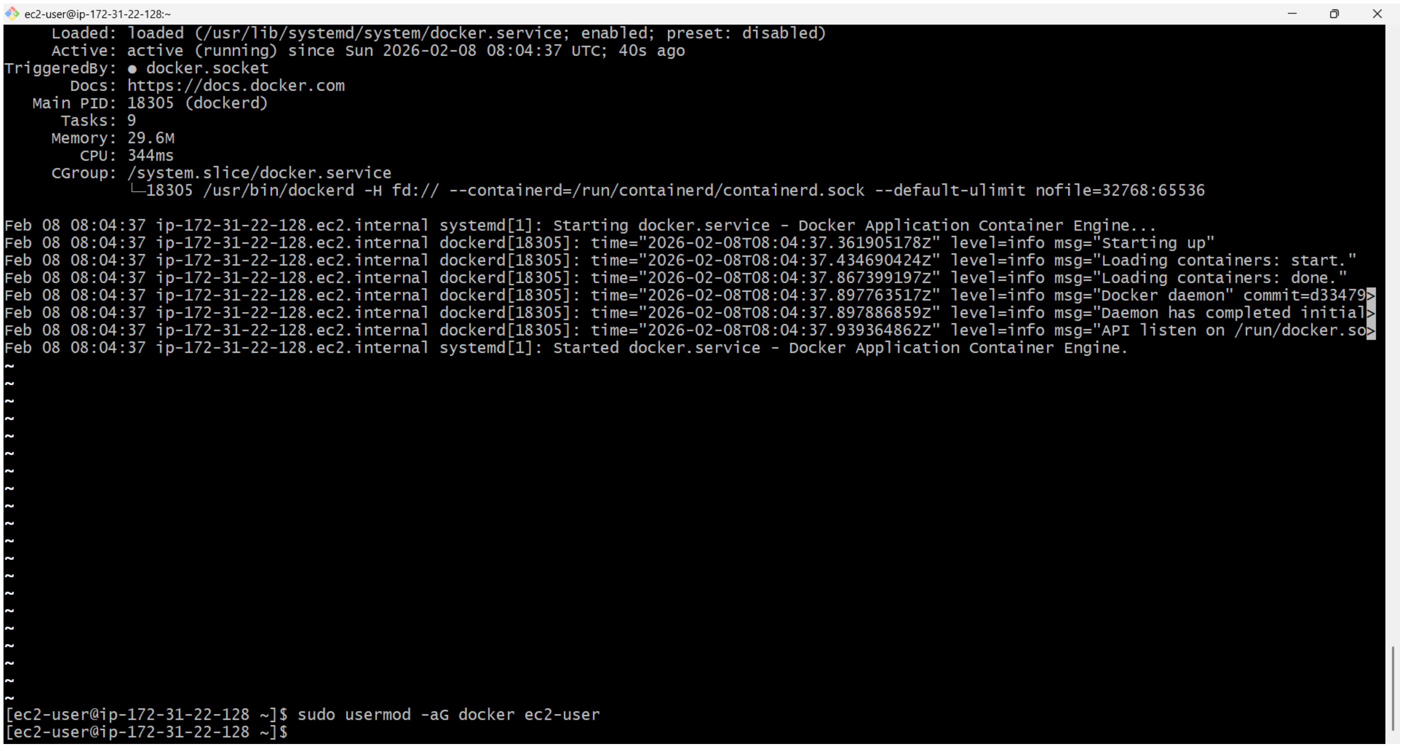
After that I update the system first, then install the docker, start and enable the docker using yum package manager.


ec2-user@ip-172-31-22-128:~

```
     dnf upgrade --releasever=2023.10.20260202

   Release notes:
    https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.10.20260202.html

===================================================================================================================

Installed:
  container-selinux-4:2.242.0-1.amzn2023.noarch                    containerd-2.1.5-1.amzn2023.0.4.x86_64
  docker-25.0.14-1.amzn2023.0.1.x86_64                             iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                         libcgroup-3.0-1.amzn2023.0.1.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64               libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64
  libnftnl-1.2.2-2.amzn2023.0.2.x86_64                             pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.3.4-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-22-128 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-22-128 ~]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-22-128 ~]$ systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)
     Active: active (running) since Sun 2026-02-08 08:04:37 UTC; 40s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 18305 (dockerd)
      Tasks: 9
     Memory: 29.6M
        CPU: 344ms
     CGroup: /system.slice/docker.service
             └─18305 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:655>

Feb 08 08:04:37 ip-172-31-22-128.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.361905178Z" level=info msg="Starting >
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.434690424Z" level=info msg="Loading c>
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.867399197Z" level=info msg="Loading c>
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.897763517Z" level=info msg="Docker da>
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.897886859Z" level=info msg="Daemon ha>
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.939364862Z" level=info msg="API liste>
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-20/20 (END)...skipping...
```

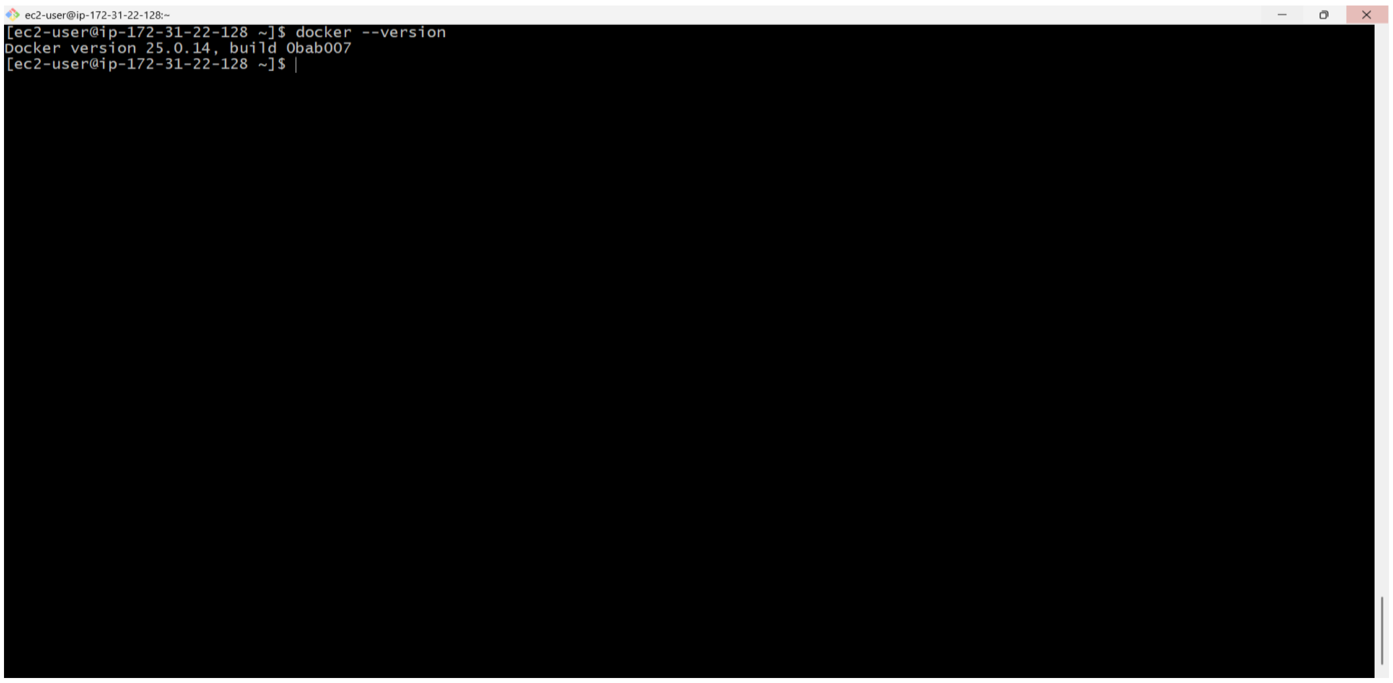
Now the docker is in running state.

Then I just add the ec2-user to docker group that is already created by default because without docker group I need to use sudo to run the docker commands and that will take more time after adding the ec2-user in the docker group I can run docker command without using sudo.



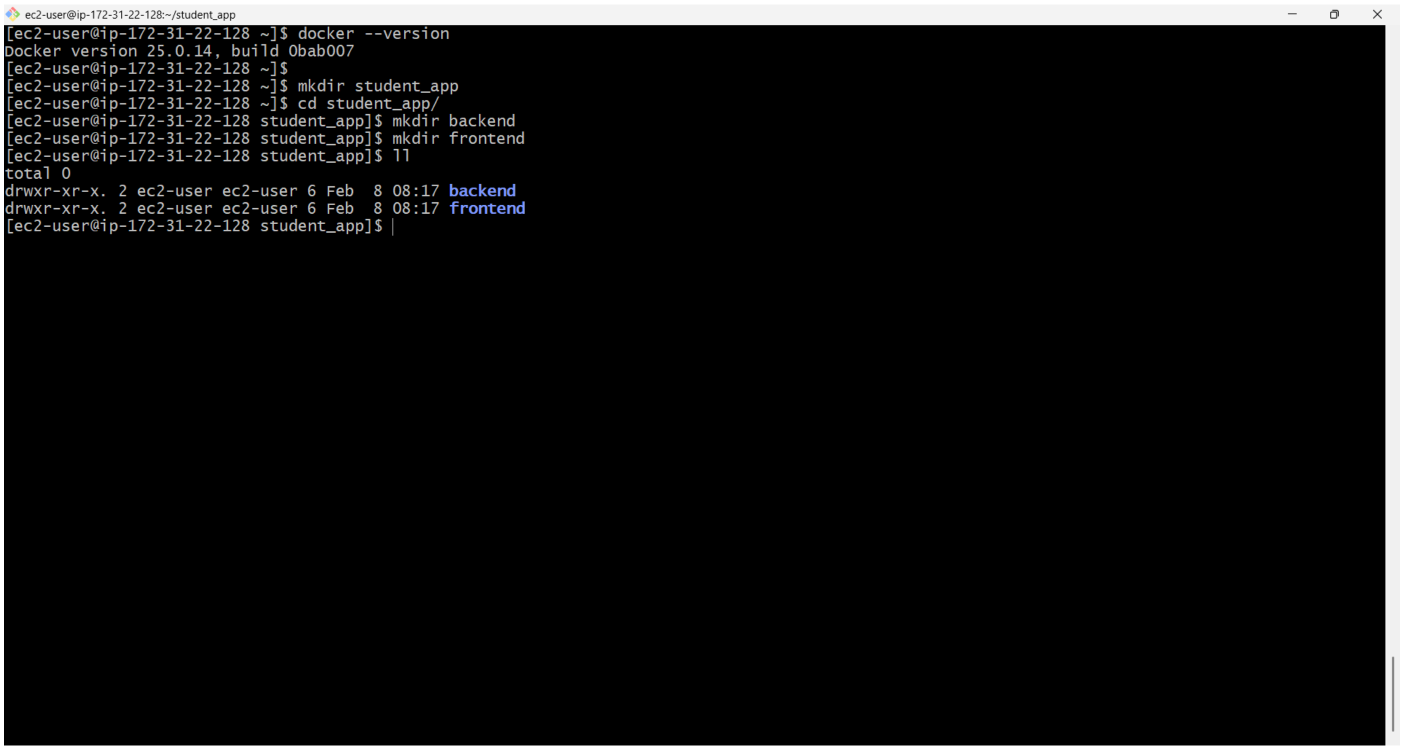
```
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)
     Active: active (running) since Sun 2026-02-08 08:04:37 UTC; 40s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 18305 (dockerd)
      Tasks: 9
     Memory: 29.6M
        CPU: 344ms
     CGroup: /system.slice/docker.service
             └─18305 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Feb 08 08:04:37 ip-172-31-22-128.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.361905178Z" level=info msg="Starting up"
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.434690424Z" level=info msg="Loading containers: start."
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.867399197Z" level=info msg="Loading containers: done."
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.897763517Z" level=info msg="Docker daemon" commit=d33479>
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.897886859Z" level=info msg="Daemon has completed initial>
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal dockerd[18305]: time="2026-02-08T08:04:37.939364862Z" level=info msg="API listen on /run/docker.so>
Feb 08 08:04:37 ip-172-31-22-128.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
[ec2-user@ip-172-31-22-128 ~]$ sudo usermod -aG docker ec2-user
[ec2-user@ip-172-31-22-128 ~]$
```

After that I check the docker is install or not using docker –version command



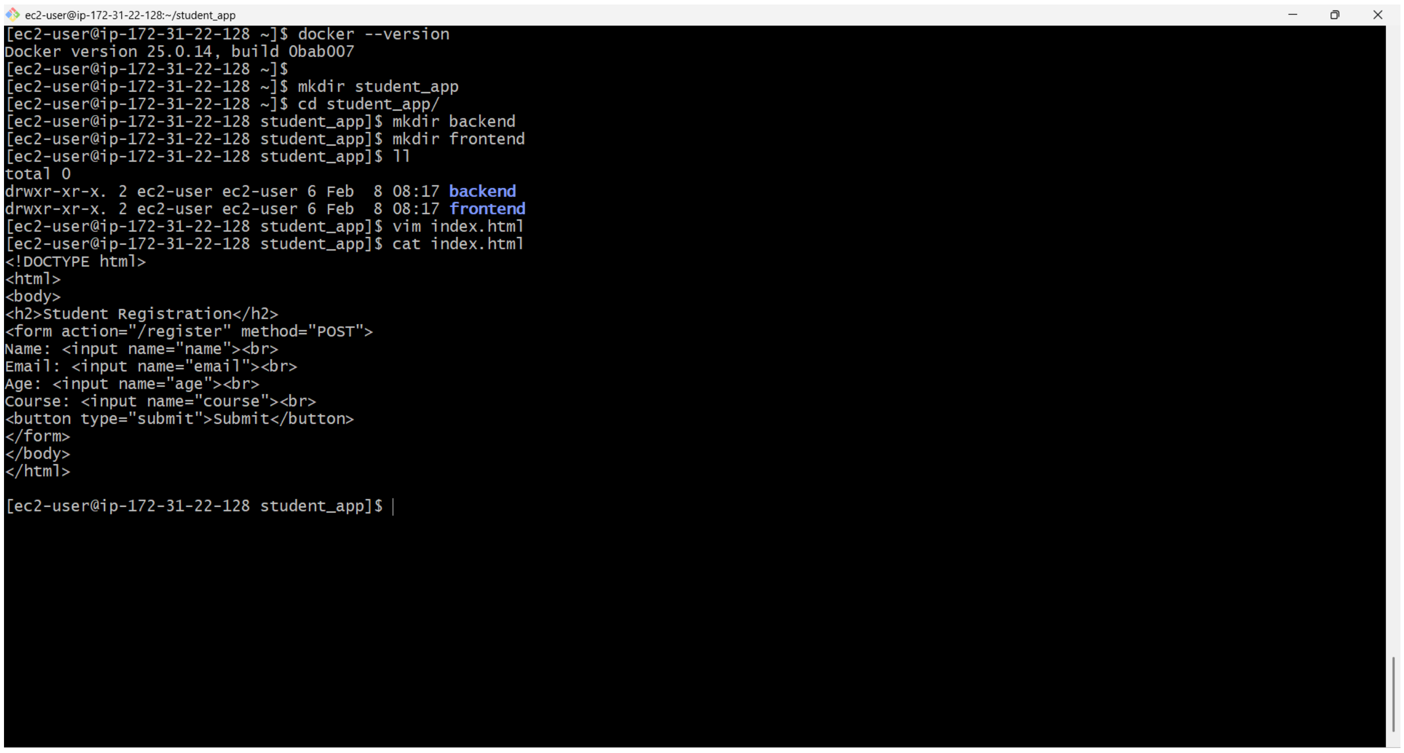
```
[ec2-user@ip-172-31-22-128 ~]$ docker --version
Docker version 25.0.14, build 0bab007
[ec2-user@ip-172-31-22-128 ~]$ |
```

After that I create a working directory that contain my frontend and backend code and dependencies.



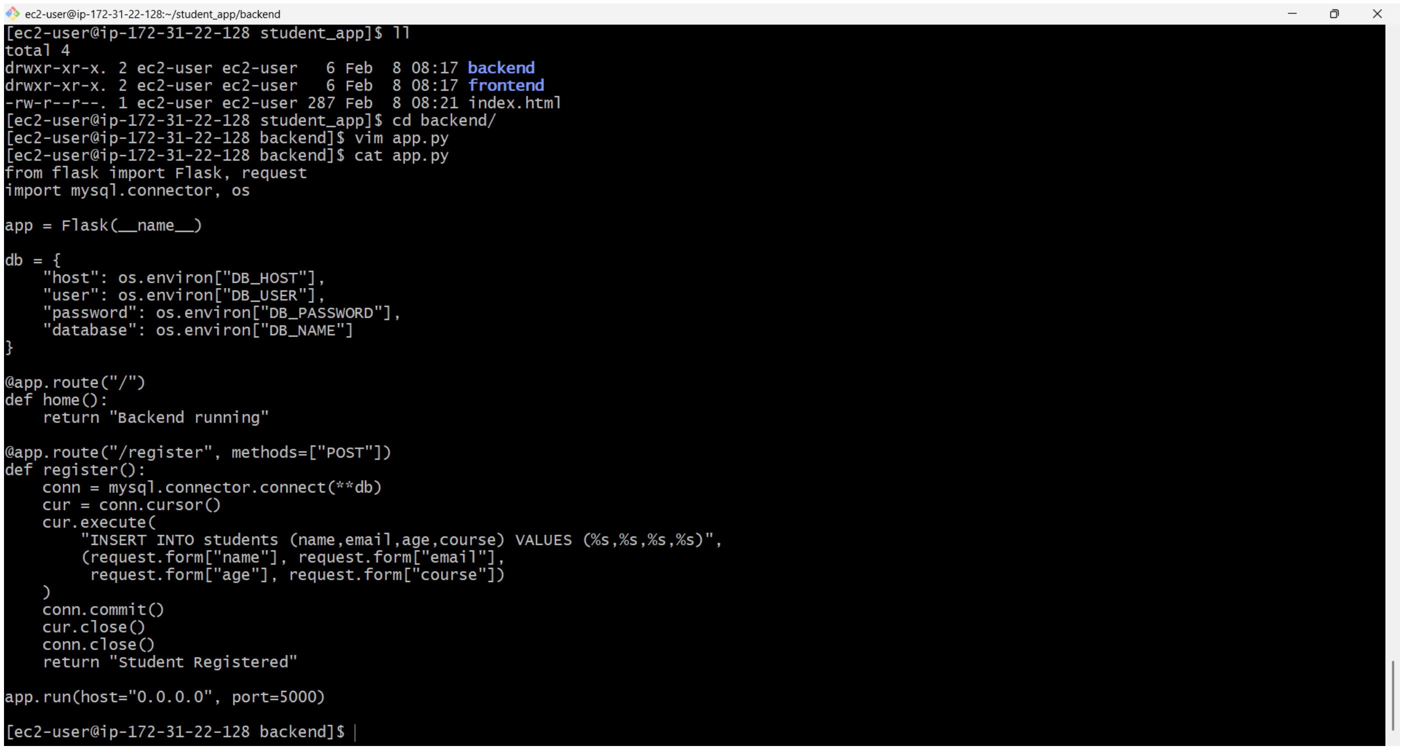
```
[ec2-user@ip-172-31-22-128 ~]$ docker --version
Docker version 25.0.14, build 0bab007
[ec2-user@ip-172-31-22-128 ~]$
[ec2-user@ip-172-31-22-128 ~]$ mkdir student_app
[ec2-user@ip-172-31-22-128 ~]$ cd student_app/
[ec2-user@ip-172-31-22-128 student_app]$ mkdir backend
[ec2-user@ip-172-31-22-128 student_app]$ mkdir frontend
[ec2-user@ip-172-31-22-128 student_app]$ ll
total 0
drwxr-xr-x. 2 ec2-user ec2-user 6 Feb  8 08:17 backend
drwxr-xr-x. 2 ec2-user ec2-user 6 Feb  8 08:17 frontend
[ec2-user@ip-172-31-22-128 student_app]$ |
```

After that I created a index.html file using vim editor that contain the frontend code



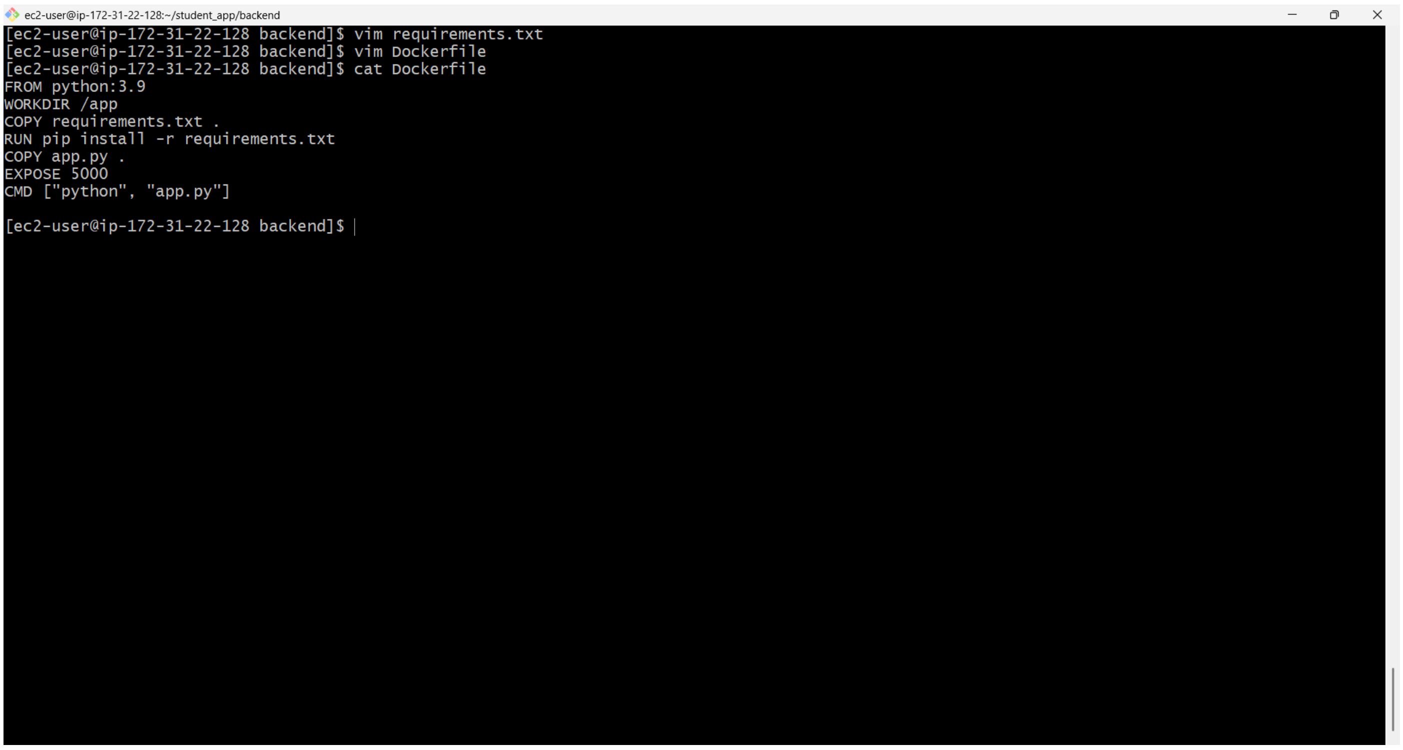
```
[ec2-user@ip-172-31-22-128 ~]$ docker --version
Docker version 25.0.14, build 0bab007
[ec2-user@ip-172-31-22-128 ~]$
[ec2-user@ip-172-31-22-128 ~]$ mkdir student_app
[ec2-user@ip-172-31-22-128 ~]$ cd student_app/
[ec2-user@ip-172-31-22-128 student_app]$ mkdir backend
[ec2-user@ip-172-31-22-128 student_app]$ mkdir frontend
[ec2-user@ip-172-31-22-128 student_app]$ ll
total 0
drwxr-xr-x. 2 ec2-user ec2-user 6 Feb  8 08:17 backend
drwxr-xr-x. 2 ec2-user ec2-user 6 Feb  8 08:17 frontend
[ec2-user@ip-172-31-22-128 student_app]$ vim index.html
[ec2-user@ip-172-31-22-128 student_app]$ cat index.html
<!DOCTYPE html>
<html>
<body>
<h2>Student Registration</h2>
<form action="/register" method="POST">
Name: <input name="name"><br>
Email: <input name="email"><br>
Age: <input name="age"><br>
Course: <input name="course"><br>
<button type="submit">Submit</button>
</form>
</body>
</html>

[ec2-user@ip-172-31-22-128 student_app]$ |
```

After that I create the app.py file inside the backend folder that contain the backend code.



```
[ec2-user@ip-172-31-22-128 student_app]$ ll
total 4
drwxr-xr-x. 2 ec2-user ec2-user   6 Feb  8 08:17 backend
drwxr-xr-x. 2 ec2-user ec2-user   6 Feb  8 08:17 frontend
-rw-r--r--. 1 ec2-user ec2-user 287 Feb  8 08:21 index.html
[ec2-user@ip-172-31-22-128 student_app]$ cd backend/
[ec2-user@ip-172-31-22-128 backend]$ vim app.py
[ec2-user@ip-172-31-22-128 backend]$ cat app.py
from flask import Flask, request
import mysql.connector, os

app = Flask(__name__)

db = {
    "host": os.environ["DB_HOST"],
    "user": os.environ["DB_USER"],
    "password": os.environ["DB_PASSWORD"],
    "database": os.environ["DB_NAME"]
}

@app.route("/")
def home():
    return "Backend running"

@app.route("/register", methods=["POST"])
def register():
    conn = mysql.connector.connect(**db)
    cur = conn.cursor()
    cur.execute(
        "INSERT INTO students (name,email,age,course) VALUES (%s,%s,%s,%s)",
        (request.form["name"], request.form["email"],
         request.form["age"], request.form["course"])
    )
    conn.commit()
    cur.close()
    conn.close()
    return "Student Registered"

app.run(host="0.0.0.0", port=5000)

[ec2-user@ip-172-31-22-128 backend]$ |
```

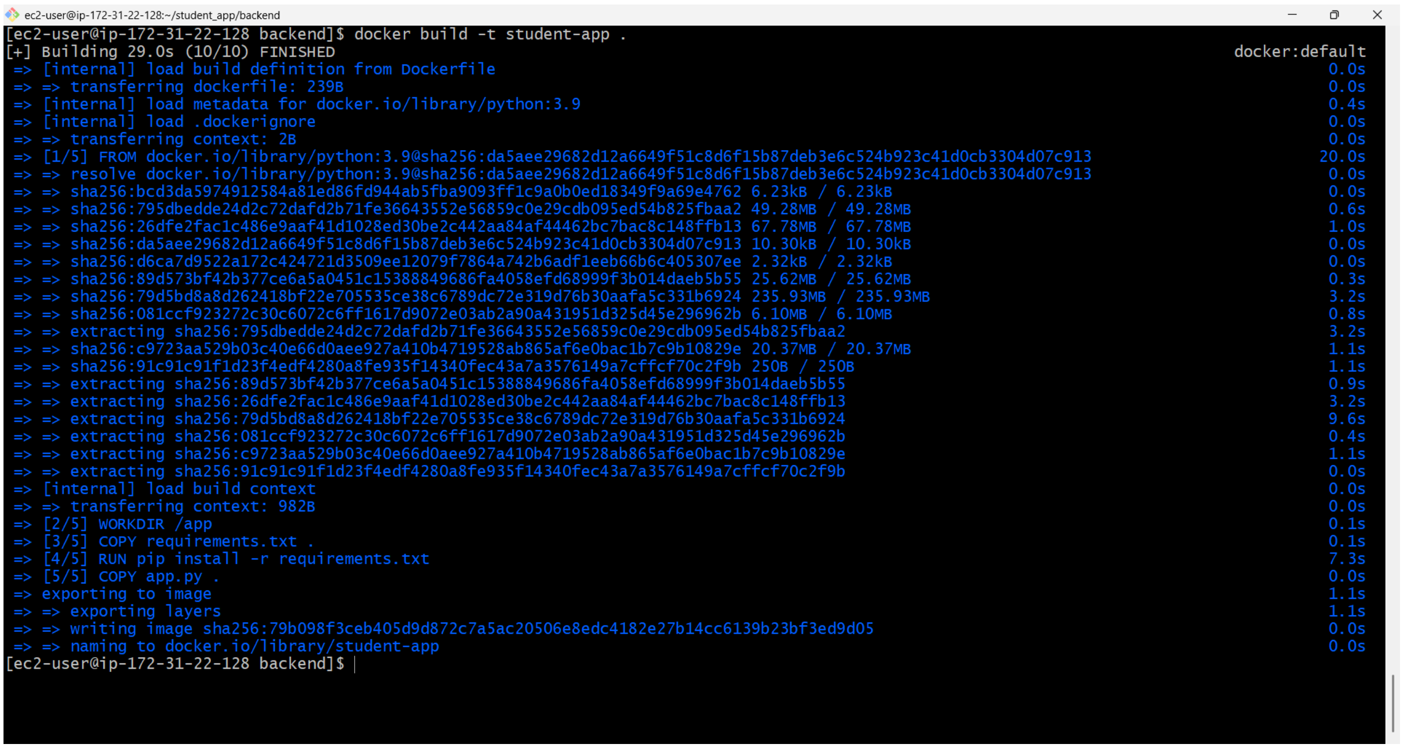
After that I created the a Dockerfile and requirements.txt in the backend folder.



ec2-user@ip-172-31-22-128:~/student_app/backend

```
[ec2-user@ip-172-31-22-128 backend]$ vim requirements.txt
[ec2-user@ip-172-31-22-128 backend]$ vim Dockerfile
[ec2-user@ip-172-31-22-128 backend]$ cat Dockerfile
FROM python:3.9
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY app.py .
EXPOSE 5000
CMD ["python", "app.py"]

[ec2-user@ip-172-31-22-128 backend]$
```

After that I build the image from Dockerfile in the same folder

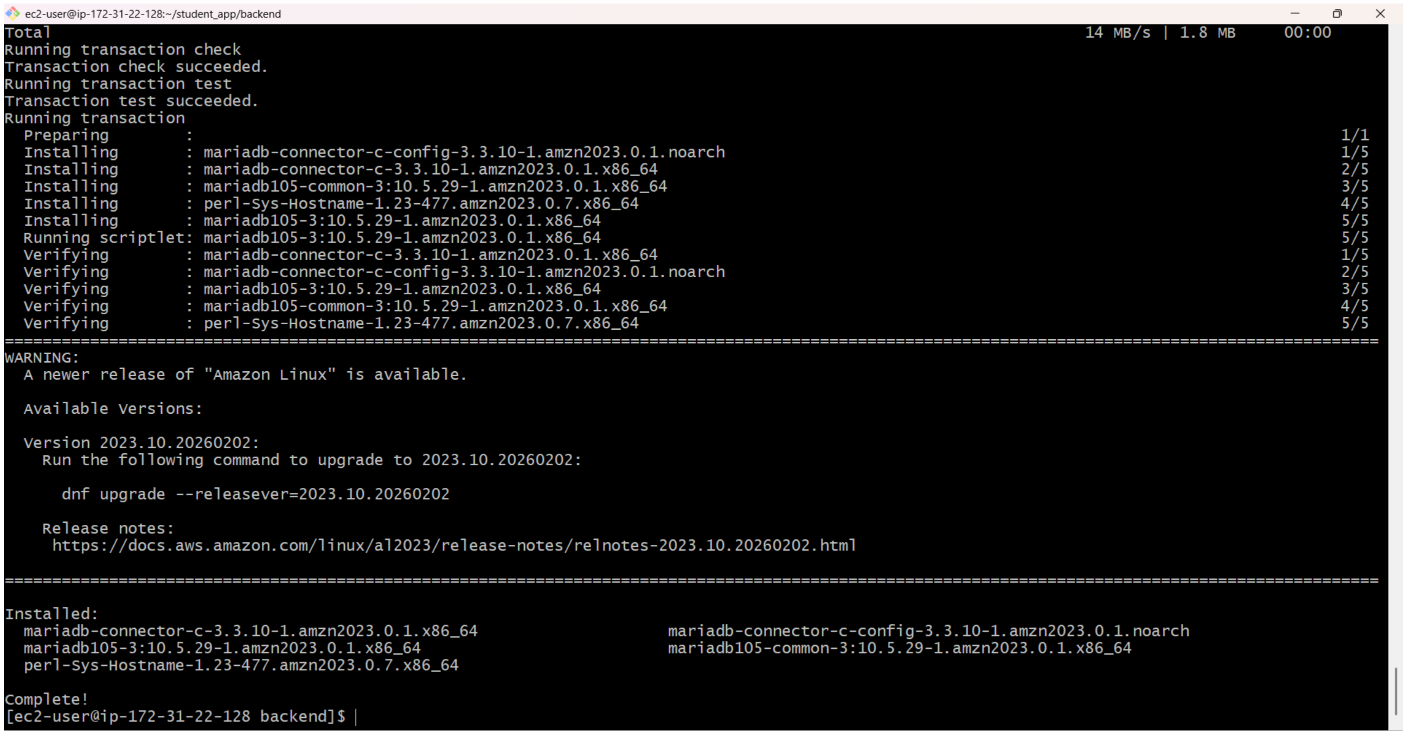


ec2-user@ip-172-31-22-128:~/student_app/backend

```
[ec2-user@ip-172-31-22-128 backend]$ docker build -t student-app .
[+] Building 29.0s (10/10) FINISHED                                                                                           docker:default
 => [internal] load build definition from Dockerfile                                                                                    0.0s
 => => transferring dockerfile: 239B                                                                                                    0.0s
 => [internal] load metadata for docker.io/library/python:3.9                                                                           0.4s
 => [internal] load .dockerignore                                                                                                       0.0s
 => => transferring context: 2B                                                                                                         0.0s
 => [1/5] FROM docker.io/library/python:3.9@sha256:da5aee29682d12a6649f51c8d6f15b87deb3e6c524b923c41d0cb3304d07c913                    20.0s
 => => resolve docker.io/library/python:3.9@sha256:da5aee29682d12a6649f51c8d6f15b87deb3e6c524b923c41d0cb3304d07c913                     0.0s
 => => sha256:bcd3da5974912584a81ed86fd944ab5fba9093ff1c9a0b0ed18349f9a69e4762 6.23kB / 6.23kB                                          0.0s
 => => sha256:795dbedde24d2c72dafd2b71fe36643552e56859c0e29cdb095ed54b825fbaa2 49.28MB / 49.28MB                                        0.6s
 => => sha256:26dfe2fac1c486e9aaf41d1028ed30be2c442aa84af44462bc7bac8c148ffb13 67.78MB / 67.78MB                                        1.0s
 => => sha256:da5aee29682d12a6649f51c8d6f15b87deb3e6c524b923c41d0cb3304d07c913 10.30kB / 10.30kB                                        0.0s
 => => sha256:d6ca7d9522a172c424721d3509ee12079f7864a742b6adf1eeb66b6c405307ee 2.32kB / 2.32kB                                          0.0s
 => => sha256:89d573bf42b377ce6a5a0451c15388849686fa4058efd68999f3b014daeb5b55 25.62MB / 25.62MB                                        0.3s
 => => sha256:79d5bd8a8d262418bf22e705535ce38c6789dc72e319d76b30aafa5c331b6924 235.93MB / 235.93MB                                      3.2s
 => => sha256:081ccf923272c30c6072c6ff1617d9072e03ab2a90a431951d325d45e296962b 6.10MB / 6.10MB                                          0.8s
 => => extracting sha256:795dbedde24d2c72dafd2b71fe36643552e56859c0e29cdb095ed54b825fbaa2                                               3.2s
 => => sha256:c9723aa529b03c40e66d0aee927a410b4719528ab865af6e0bac1b7c9b10829e 20.37MB / 20.37MB                                        1.1s
 => => sha256:91c91c91f1d23f4edf4280a8fe935f14340fec43a7a3576149a7cffcf70c2f9b 250B / 250B                                              1.1s
 => => extracting sha256:89d573bf42b377ce6a5a0451c15388849686fa4058efd68999f3b014daeb5b55                                               0.9s
 => => extracting sha256:26dfe2fac1c486e9aaf41d1028ed30be2c442aa84af44462bc7bac8c148ffb13                                               3.2s
 => => extracting sha256:79d5bd8a8d262418bf22e705535ce38c6789dc72e319d76b30aafa5c331b6924                                               9.6s
 => => extracting sha256:081ccf923272c30c6072c6ff1617d9072e03ab2a90a431951d325d45e296962b                                               0.4s
 => => extracting sha256:c9723aa529b03c40e66d0aee927a410b4719528ab865af6e0bac1b7c9b10829e                                               1.1s
 => => extracting sha256:91c91c91f1d23f4edf4280a8fe935f14340fec43a7a3576149a7cffcf70c2f9b                                               0.0s
 => [internal] load build context                                                                                                       0.0s
 => => transferring context: 982B                                                                                                       0.0s
 => [2/5] WORKDIR /app                                                                                                                  0.1s
 => [3/5] COPY requirements.txt .                                                                                                       0.1s
 => [4/5] RUN pip install -r requirements.txt                                                                                           7.3s
 => [5/5] COPY app.py .                                                                                                                 0.0s
 => exporting to image                                                                                                                  1.1s
 => => exporting layers                                                                                                                 1.1s
 => => writing image sha256:79b098f3ceb405d9d872c7a5ac20506e8edc4182e27b14cc6139b23bf3ed9d05                                            0.0s
 => => naming to docker.io/library/student-app                                                                                          0.0s
[ec2-user@ip-172-31-22-128 backend]$
```

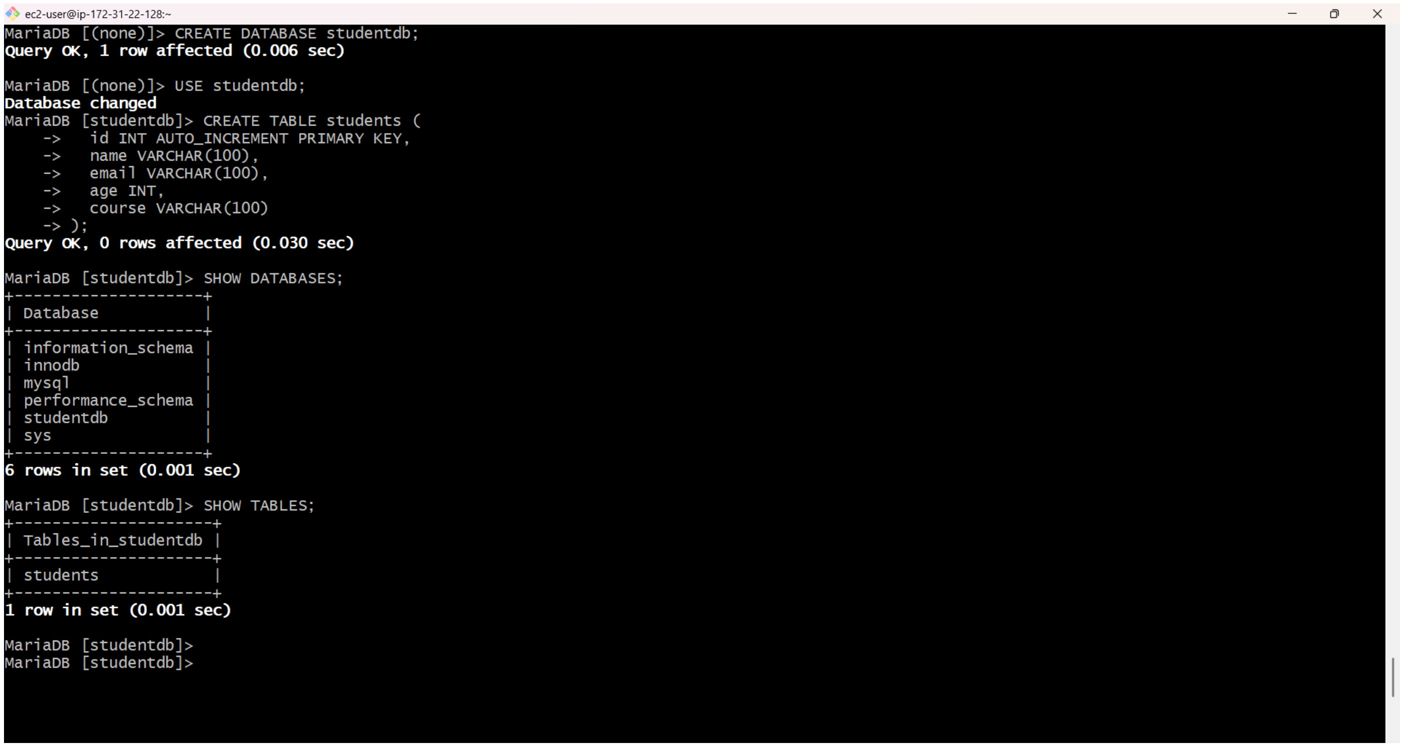
So, the frontend and backend is done.

after that I install the mariadb105 packaged using yum package manager to configure the database using mysql -h command.



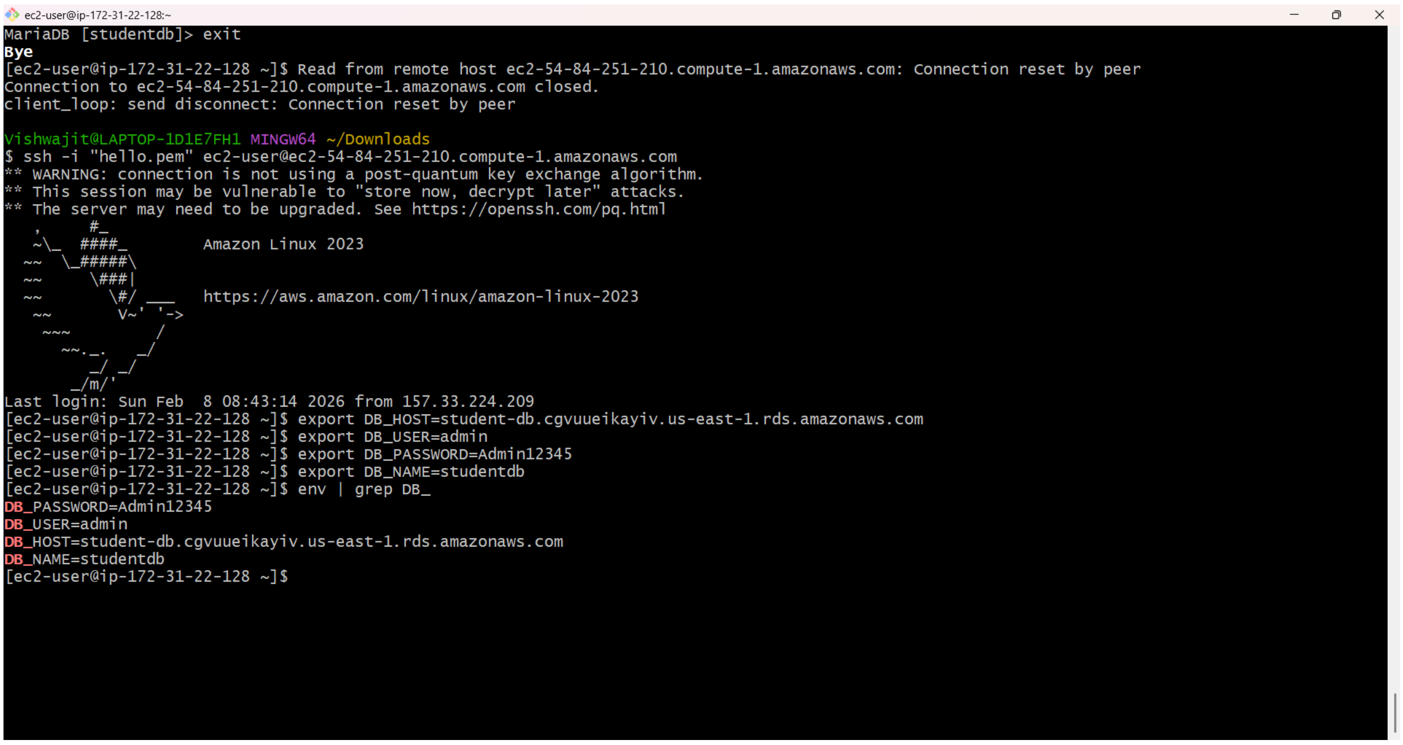
```
Total                                                                                                                   14 MB/s | 1.8 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                                          1/1
  Installing       : mariadb-connector-c-config-3.3.10-1.amzn2023.0.1.noarch                                                                1/5
  Installing       : mariadb-connector-c-3.3.10-1.amzn2023.0.1.x86_64                                                                       2/5
  Installing       : mariadb105-common-3:10.5.29-1.amzn2023.0.1.x86_64                                                                      3/5
  Installing       : perl-Sys-Hostname-1.23-477.amzn2023.0.7.x86_64                                                                         4/5
  Installing       : mariadb105-3:10.5.29-1.amzn2023.0.1.x86_64                                                                             5/5
  Running scriptlet: mariadb105-3:10.5.29-1.amzn2023.0.1.x86_64                                                                             5/5
  Verifying        : mariadb-connector-c-3.3.10-1.amzn2023.0.1.x86_64                                                                       1/5
  Verifying        : mariadb-connector-c-config-3.3.10-1.amzn2023.0.1.noarch                                                                2/5
  Verifying        : mariadb105-3:10.5.29-1.amzn2023.0.1.x86_64                                                                             3/5
  Verifying        : mariadb105-common-3:10.5.29-1.amzn2023.0.1.x86_64                                                                      4/5
  Verifying        : perl-Sys-Hostname-1.23-477.amzn2023.0.7.x86_64                                                                         5/5
==================================================================================================================================================
WARNING:
  A newer release of "Amazon Linux" is available.

  Available Versions:

  Version 2023.10.20260202:
    Run the following command to upgrade to 2023.10.20260202:

      dnf upgrade --releasever=2023.10.20260202

    Release notes:
     https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.10.20260202.html

==================================================================================================================================================

Installed:
  mariadb-connector-c-3.3.10-1.amzn2023.0.1.x86_64                     mariadb-connector-c-config-3.3.10-1.amzn2023.0.1.noarch
  mariadb105-3:10.5.29-1.amzn2023.0.1.x86_64                           mariadb105-common-3:10.5.29-1.amzn2023.0.1.x86_64
  perl-Sys-Hostname-1.23-477.amzn2023.0.7.x86_64

Complete!
[ec2-user@ip-172-31-22-128 backend]$ |
```

After connected to the database I created a studentdb database and then create the table.



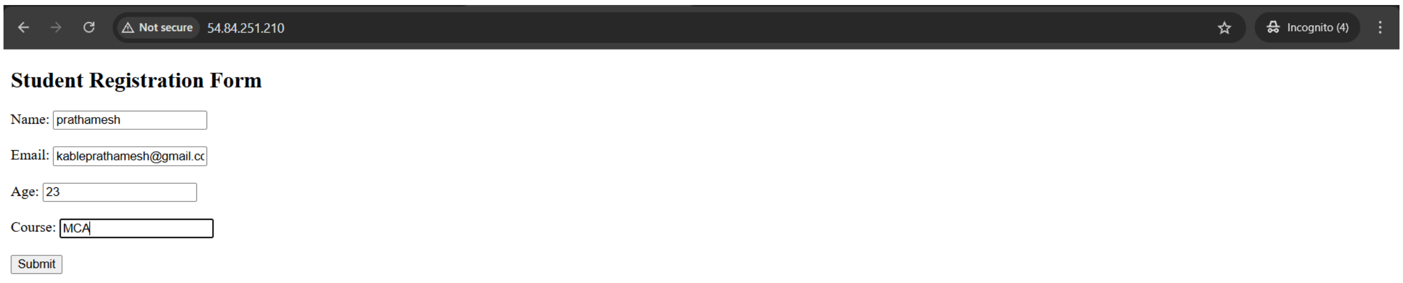
```
MariaDB [(none)]> CREATE DATABASE studentdb;
Query OK, 1 row affected (0.006 sec)

MariaDB [(none)]> USE studentdb;
Database changed
MariaDB [studentdb]> CREATE TABLE students (
    ->   id INT AUTO_INCREMENT PRIMARY KEY,
    ->   name VARCHAR(100),
    ->   email VARCHAR(100),
    ->   age INT,
    ->   course VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.030 sec)

MariaDB [studentdb]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mysql              |
| performance_schema |
| studentdb          |
| sys                |
+--------------------+
6 rows in set (0.001 sec)

MariaDB [studentdb]> SHOW TABLES;
+---------------------+
| Tables_in_studentdb |
+---------------------+
| students            |
+---------------------+
1 row in set (0.001 sec)

MariaDB [studentdb]>
MariaDB [studentdb]>
```

After that I just created the environment variable to help the app to know where the DB is.



```
MariaDB [studentdb]> exit
Bye
[ec2-user@ip-172-31-22-128 ~]$ Read from remote host ec2-54-84-251-210.compute-1.amazonaws.com: Connection reset by peer
Connection to ec2-54-84-251-210.compute-1.amazonaws.com closed.
client_loop: send disconnect: Connection reset by peer

Vishwajit@LAPTOP-1D1E7FH1 MINGW64 ~/Downloads
$ ssh -i "hello.pem" ec2-user@ec2-54-84-251-210.compute-1.amazonaws.com
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
         #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun Feb  8 08:43:14 2026 from 157.33.224.209
[ec2-user@ip-172-31-22-128 ~]$ export DB_HOST=student-db.cgvuueikayiv.us-east-1.rds.amazonaws.com
[ec2-user@ip-172-31-22-128 ~]$ export DB_USER=admin
[ec2-user@ip-172-31-22-128 ~]$ export DB_PASSWORD=Admin12345
[ec2-user@ip-172-31-22-128 ~]$ export DB_NAME=studentdb
[ec2-user@ip-172-31-22-128 ~]$ env | grep DB_
DB_PASSWORD=Admin12345
DB_USER=admin
DB_HOST=student-db.cgvuueikayiv.us-east-1.rds.amazonaws.com
DB_NAME=studentdb
[ec2-user@ip-172-31-22-128 ~]$
```

After I run the docker container. And then check the website is running or not.



Student Registration Form

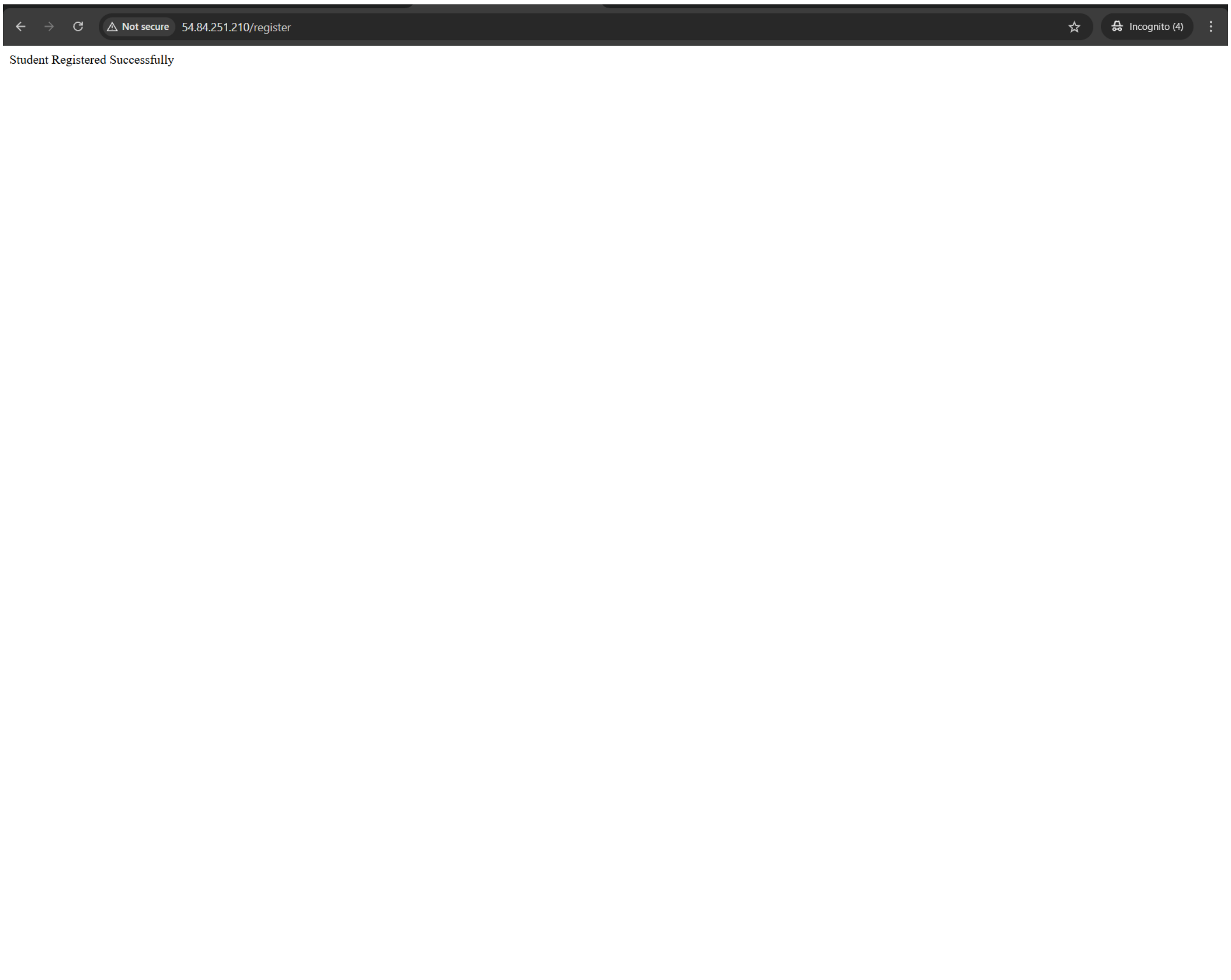
Name: prathamesh

Email: kableprathamesh@gmail.cc

Age: 23

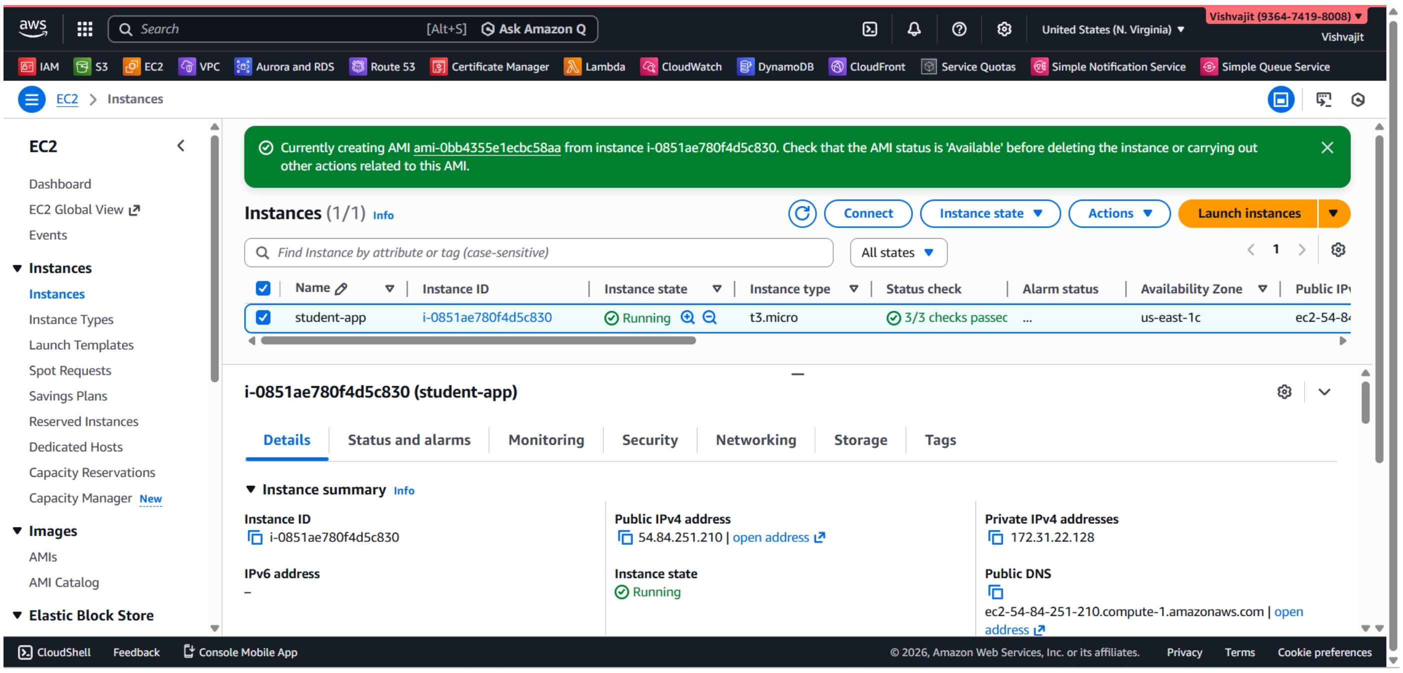
Course: MCA

Submit


Not secure 54.84.251.210/register

Student Registered Successfully

After that I create the AMI for auto scaling.


EC2 > Instances

Currently creating AMI ami-0bb4355e1ecbc58aa from instance i-0851ae780f4d5c830. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI.

Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
student-app	i-0851ae780f4d5c830	Running	t3.micro	3/3 checks passec	...	us-east-1c	ec2-54-8

i-0851ae780f4d5c830 (student-app)

Instance summary

Instance ID: i-0851ae780f4d5c830

Public IPv4 address: 54.84.251.210 | open address

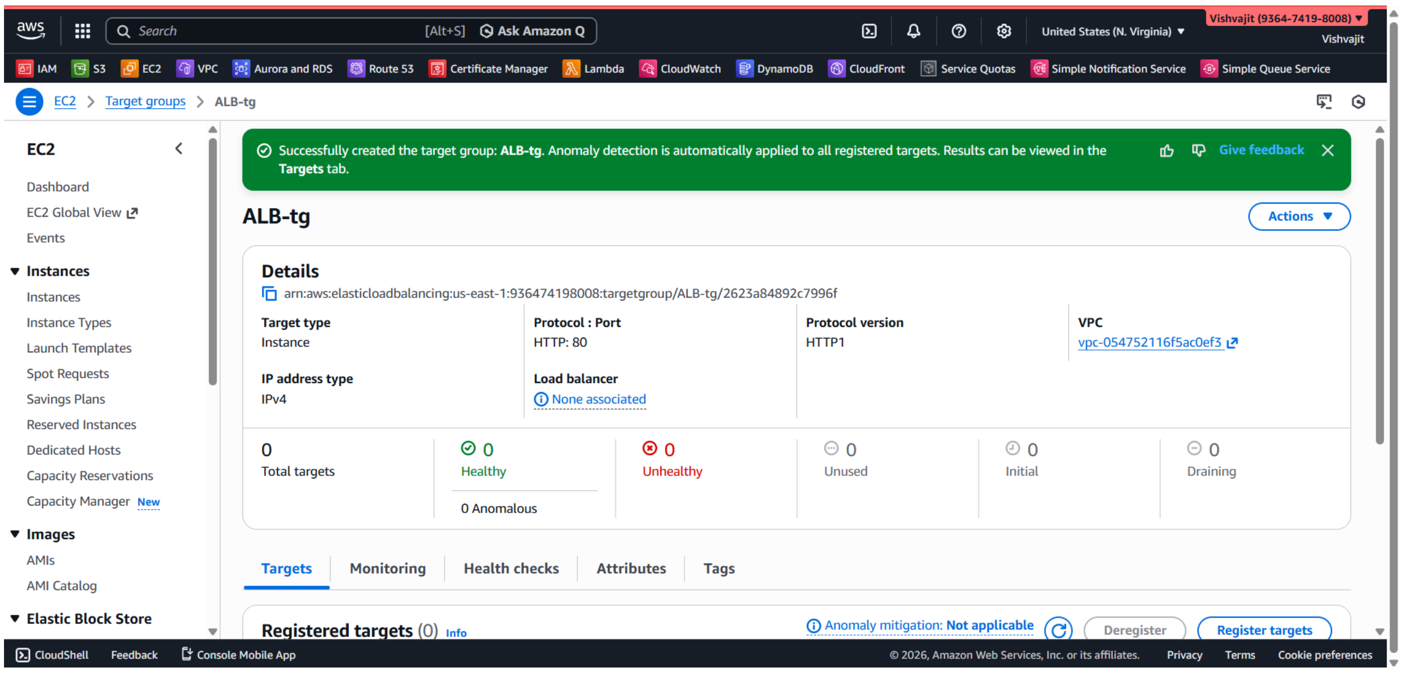
Private IPv4 addresses: 172.31.22.128

IPv6 address: –

Instance state: Running

Public DNS: ec2-54-84-251-210.compute-1.amazonaws.com | open address

After that I create a target group for load balancer.


EC2 > Target groups > ALB-tg

Successfully created the target group: **ALB-tg**. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the **Targets** tab.

ALB-tg

Details

arn:aws:elasticloadbalancing:us-east-1:936474198008:targetgroup/ALB-tg/2623a84892c7996f

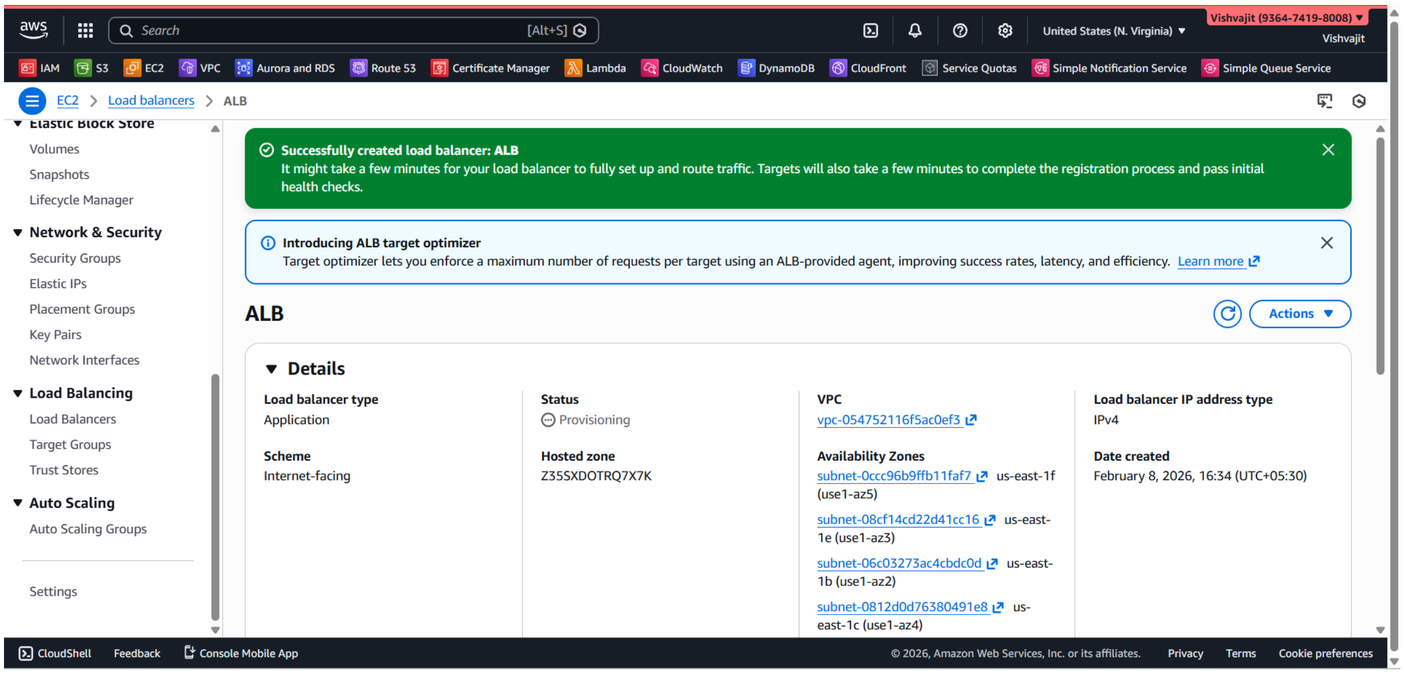
Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-054752116f5ac0ef3
IP address type	**Load balancer**		
IPv4	None associated		

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

Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (0) Info — Anomaly mitigation: Not applicable

Then I created the application load balancer to equally distribute traffic that come in load balancer.


EC2 > Load balancers > ALB

Successfully created load balancer: ALB
It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

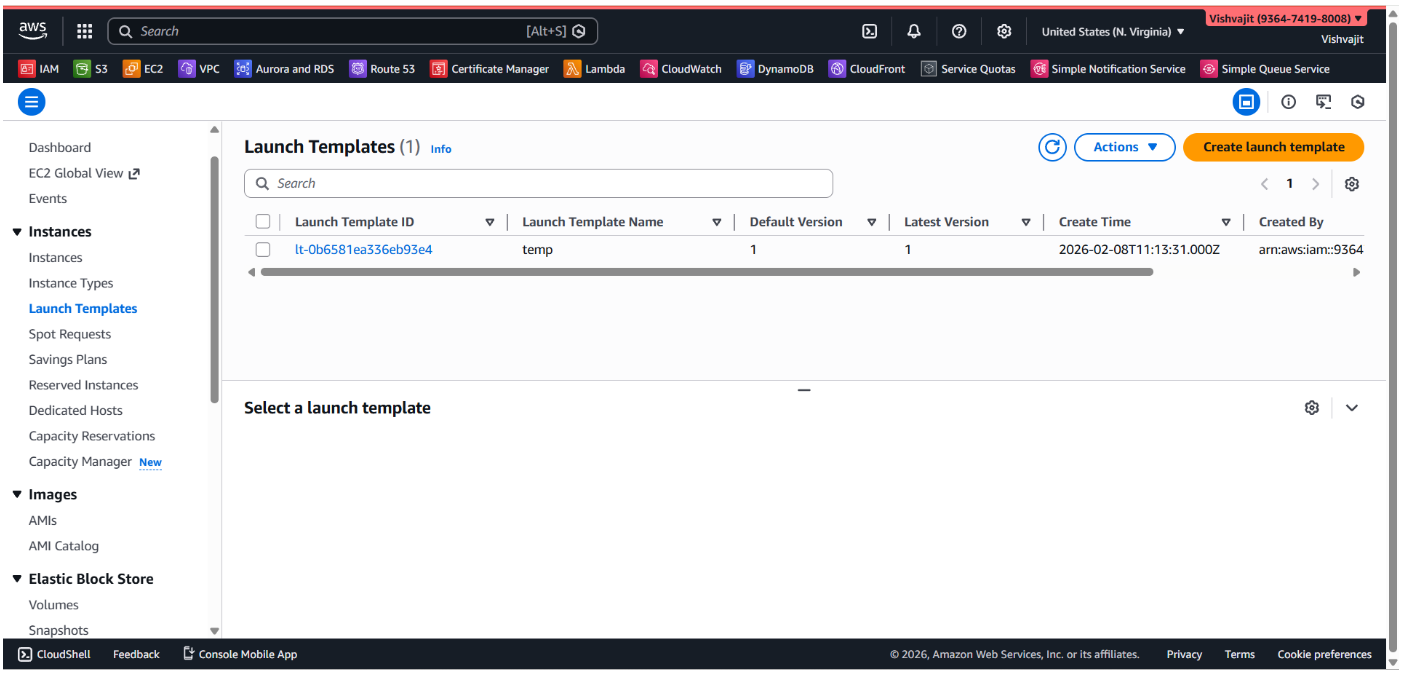
Introducing ALB target optimizer
Target optimizer lets you enforce a maximum number of requests per target using an ALB-provided agent, improving success rates, latency, and efficiency. Learn more

ALB

Details

Load balancer type	Status	VPC	Load balancer IP address type
Application	Provisioning	vpc-054752116f5ac0ef3	IPv4
Scheme	**Hosted zone**	**Availability Zones**	**Date created**
Internet-facing	Z35SXDOTRQ7X7K	subnet-0ccc96b9ffb11faf7 us-east-1f (use1-az5) subnet-08cf14cd22d41cc16 us-east-1e (use1-az3) subnet-06c03273ac4cbdc0d us-east-1b (use1-az2) subnet-0812d0d76380491e8 us-east-1c (use1-az4)	February 8, 2026, 16:34 (UTC+05:30)

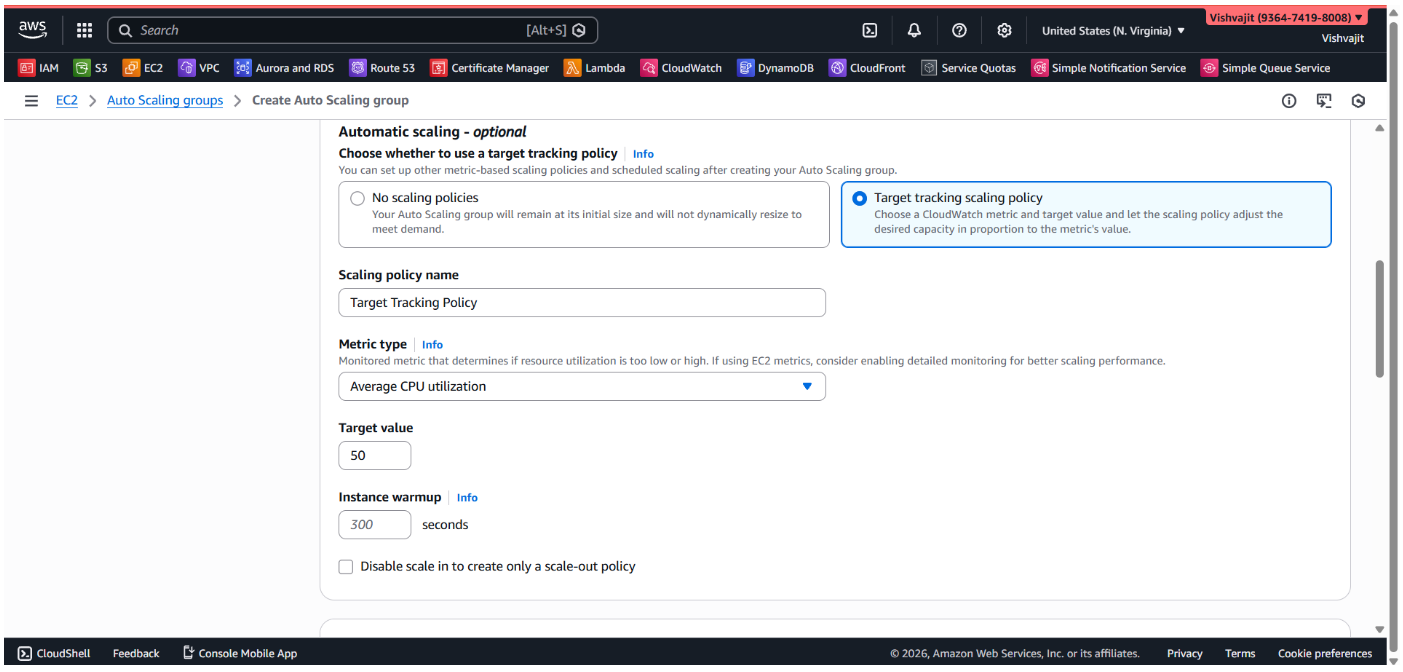
Then I created the template for auto scaling using the AMI.


Launch Templates (1)

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-0b6581ea336eb93e4	temp	1	1	2026-02-08T11:13:31.000Z	arn:aws:iam::9364

Select a launch template

Then I created the auto scaling group using template and set the target tracking policy.


EC2 > Auto Scaling groups > Create Auto Scaling group

Automatic scaling - *optional*

Choose whether to use a target tracking policy

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies [✓ Target tracking scaling policy]

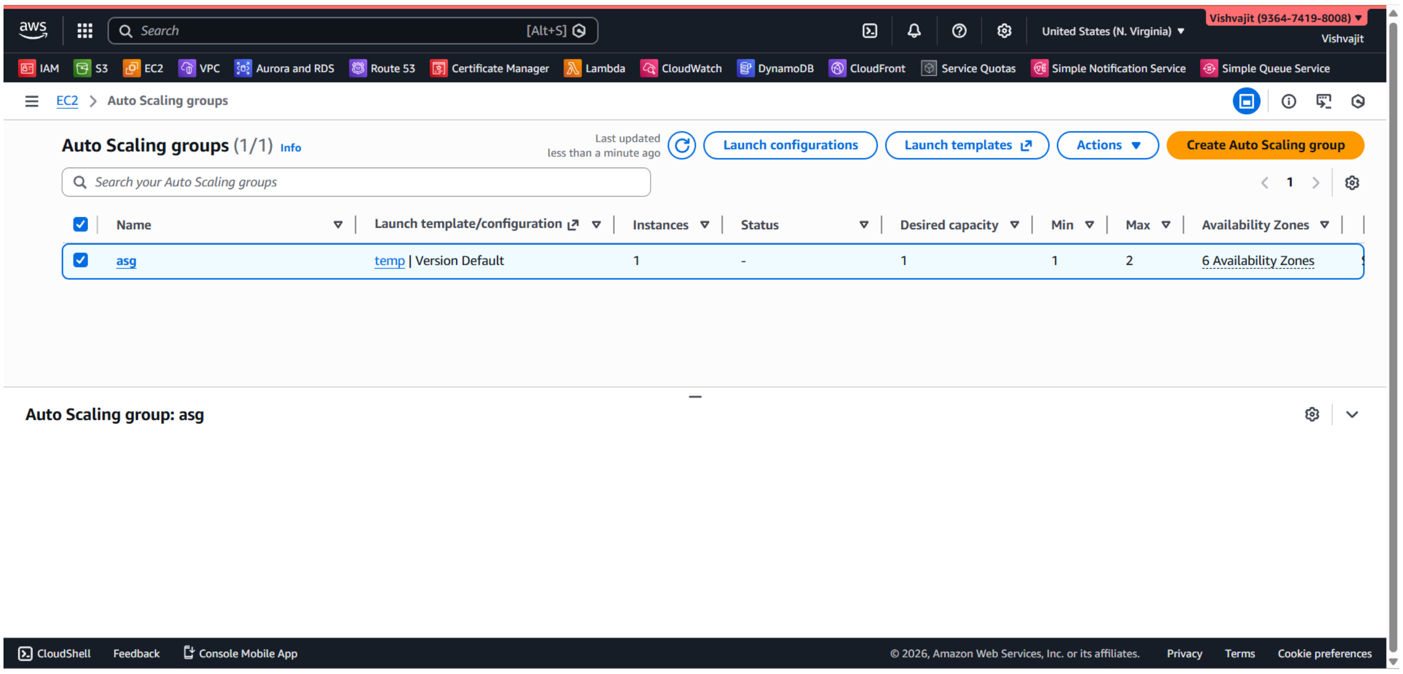
Scaling policy name: Target Tracking Policy

Metric type: Average CPU utilization

Target value: 50

Instance warmup: 300 seconds

☐ Disable scale in to create only a scale-out policy

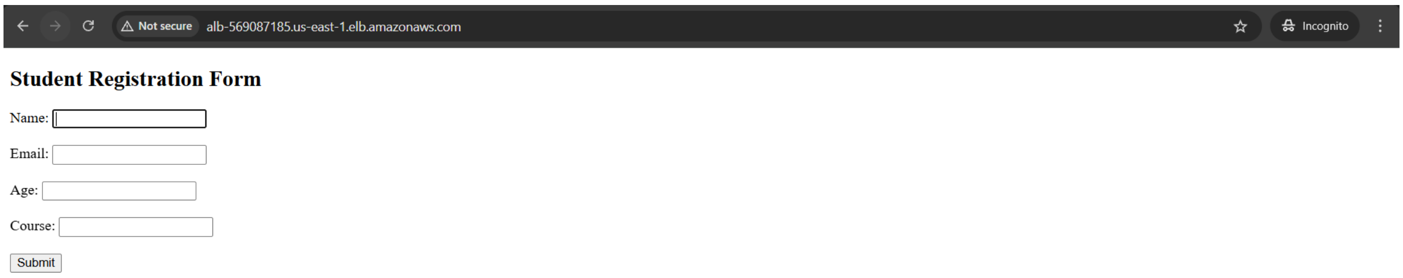

aws
Search [Alt+S]
United States (N. Virginia)
Vishvajit (9364-7419-8008)
Vishvajit
IAM S3 EC2 VPC Aurora and RDS Route 53 Certificate Manager Lambda CloudWatch DynamoDB CloudFront Service Quotas Simple Notification Service Simple Queue Service
EC2 > Auto Scaling groups
Last updated less than a minute ago
Launch configurations
Launch templates
Actions
Create Auto Scaling group
Search your Auto Scaling groups
CloudShell Feedback Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Auto Scaling groups (1/1) Info

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
asg	temp | Version Default	1	-	1	1	2	6 Availability Zones

Auto Scaling group: asg

After that I check my website is running or not using load balancer endpoint.



Student Registration Form

Name:

Email:

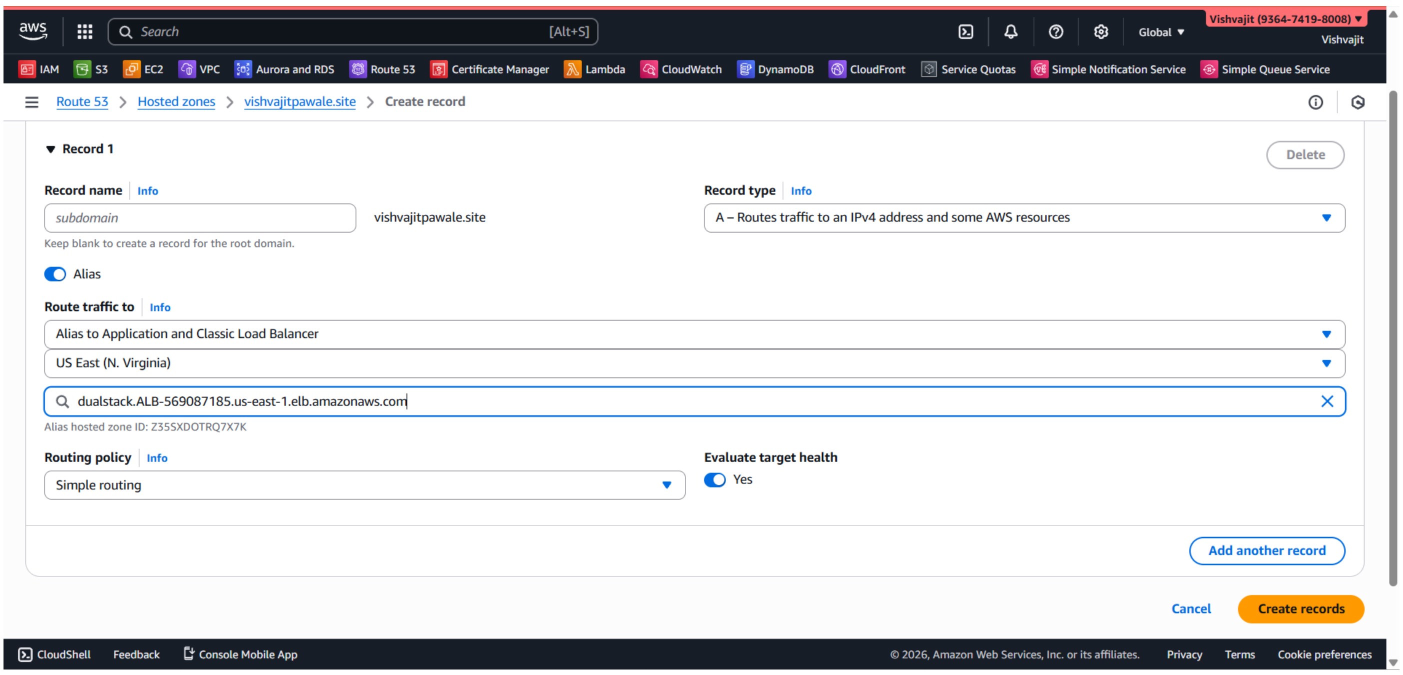
Age:

Course:

Submit

Then I create a hosted zone then create using alias feature I map the endpoint to the domain name.

Domain name is = **vishvajitpawale.site**



Record name: subdomain vishvajitpawale.site

Keep blank to create a record for the root domain.

Record type: A – Routes traffic to an IPv4 address and some AWS resources

Alias: on

Route traffic to: Alias to Application and Classic Load Balancer

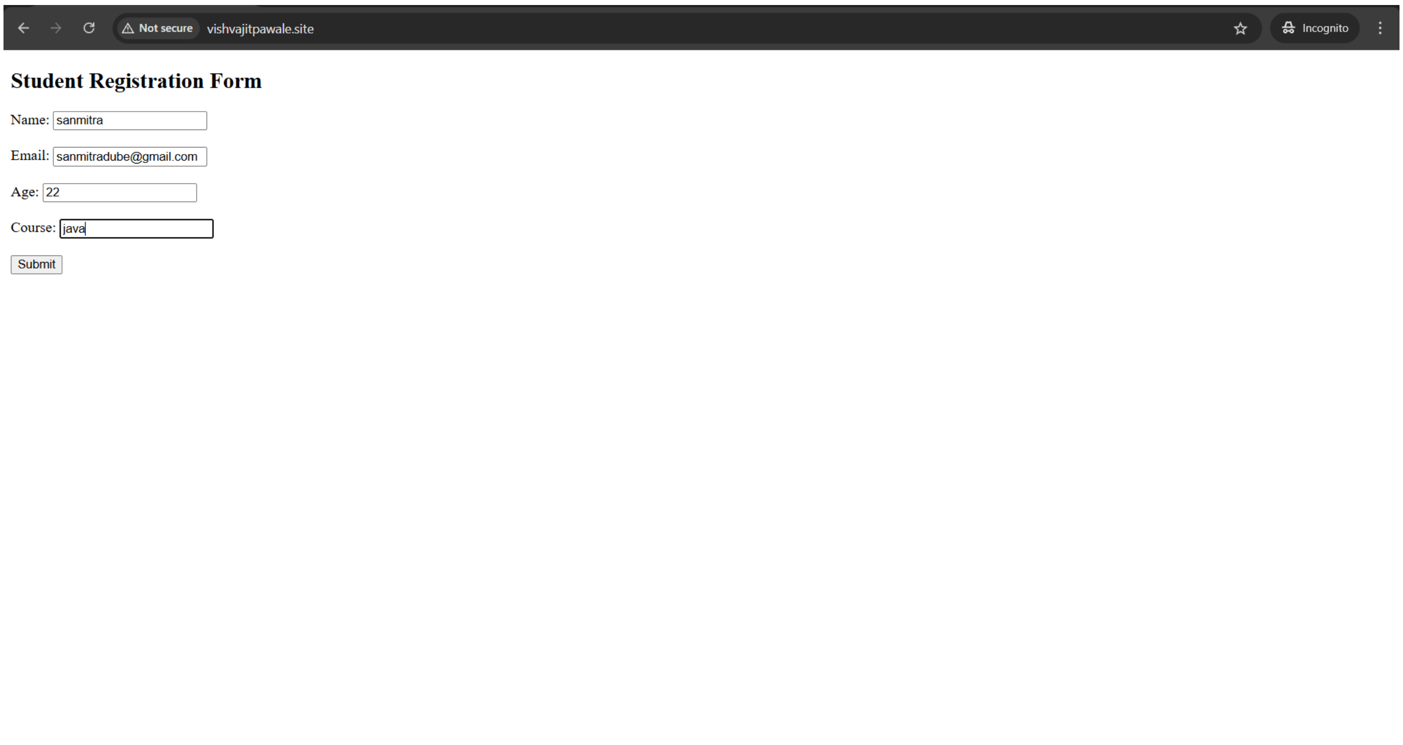
US East (N. Virginia)

dualstack.ALB-569087185.us-east-1.elb.amazonaws.com

Alias hosted zone ID: Z35SXDOTRQ7X7K

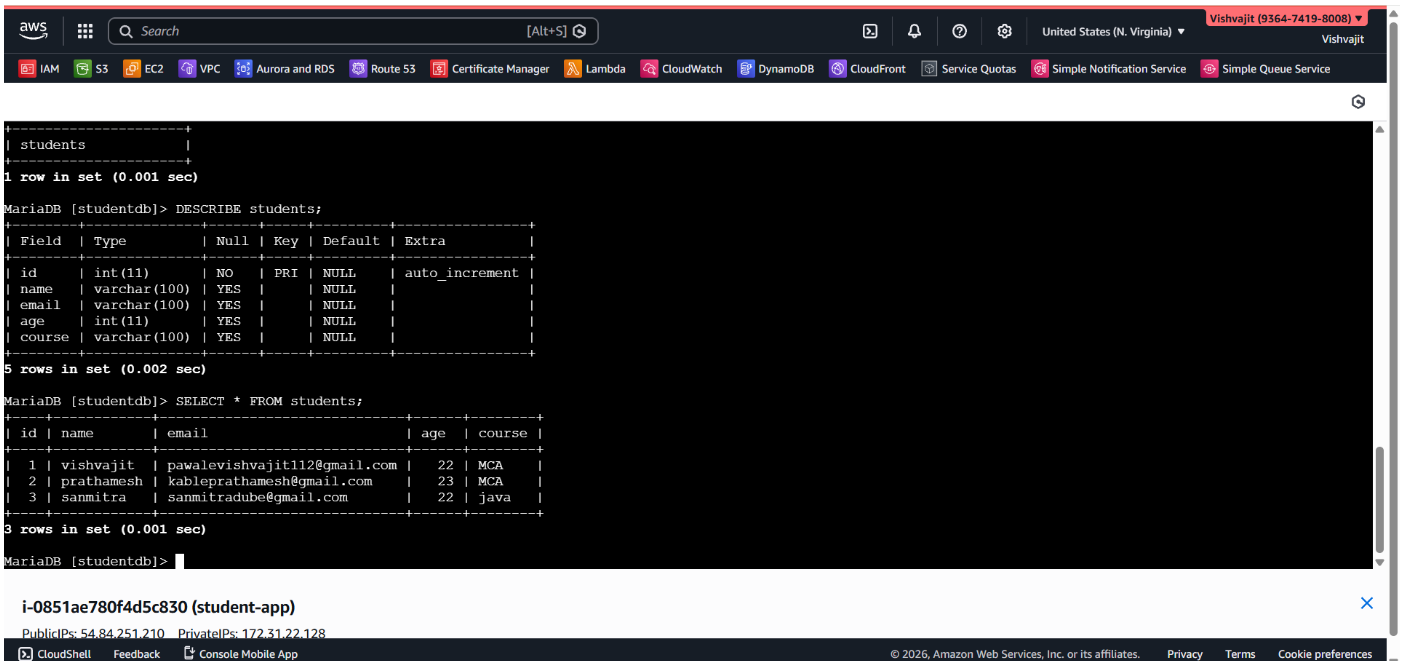
Routing policy: Simple routing

Evaluate target health: Yes



Student Registration Form

Name:	sanmitra
Email:	sanmitradube@gmail.com
Age:	22
Course:	java



```
+----------------------+
| students             |
+----------------------+
1 row in set (0.001 sec)

MariaDB [studentdb]> DESCRIBE students;
+--------+--------------+------+-----+---------+----------------+
| Field  | Type         | Null | Key | Default | Extra          |
+--------+--------------+------+-----+---------+----------------+
| id     | int(11)      | NO   | PRI | NULL    | auto_increment |
| name   | varchar(100) | YES  |     | NULL    |                |
| email  | varchar(100) | YES  |     | NULL    |                |
| age    | int(11)      | YES  |     | NULL    |                |
| course | varchar(100) | YES  |     | NULL    |                |
+--------+--------------+------+-----+---------+----------------+
5 rows in set (0.002 sec)

MariaDB [studentdb]> SELECT * FROM students;
+----+-----------+-----------------------------+------+--------+
| id | name      | email                       | age  | course |
+----+-----------+-----------------------------+------+--------+
|  1 | vishvajit | pawalevishvajit112@gmail.com |   22 | MCA    |
|  2 | prathamesh | kableprathamesh@gmail.com   |   23 | MCA    |
|  3 | sanmitra  | sanmitradube@gmail.com      |   22 | java   |
+----+-----------+-----------------------------+------+--------+
3 rows in set (0.001 sec)

MariaDB [studentdb]>
```

i-0851ae780f4d5c830 (student-app)

PublicIPs: 54.84.251.210 PrivateIPs: 172.31.22.128