AWS Project

Project Title: AWS-Powered Spare Parts Catalog Application

Project Description:

This cloud-based project is a Spare Parts Catalog Management System developed and deployed using Amazon Web Services (AWS). The application allows users to enter and manage spare part details such as part name, part number, price, and quantity through a web-based interface. The frontend is hosted on an EC2 instance and served using Nginx, while the backend is built with Node.js and Express and exposed through an Application Load Balancer (ALB) to handle incoming requests. Data is securely stored in an Amazon RDS MySQL database deployed in a private subnet, ensuring isolation and security. All AWS resources, including VPC, subnets, security groups, EC2, ALB, and RDS, were created and configured manually using the AWS Management Console, providing hands-on experience with real-world cloud infrastructure, networking, and security practices.

Aim of the Project:

The aim of this project is to design and deploy a cloud-based Spare Parts Catalog Management System using AWS that enables users to store and manage spare parts details efficiently. The project focuses on implementing a scalable, secure, and highly available architecture by leveraging AWS services such as EC2, Application Load Balancer (ALB), and Amazon RDS, while gaining hands-on experience in cloud networking, security configuration, and real-world application hosting.

Architecture Overview:

The project follows a **three-tier architecture**:

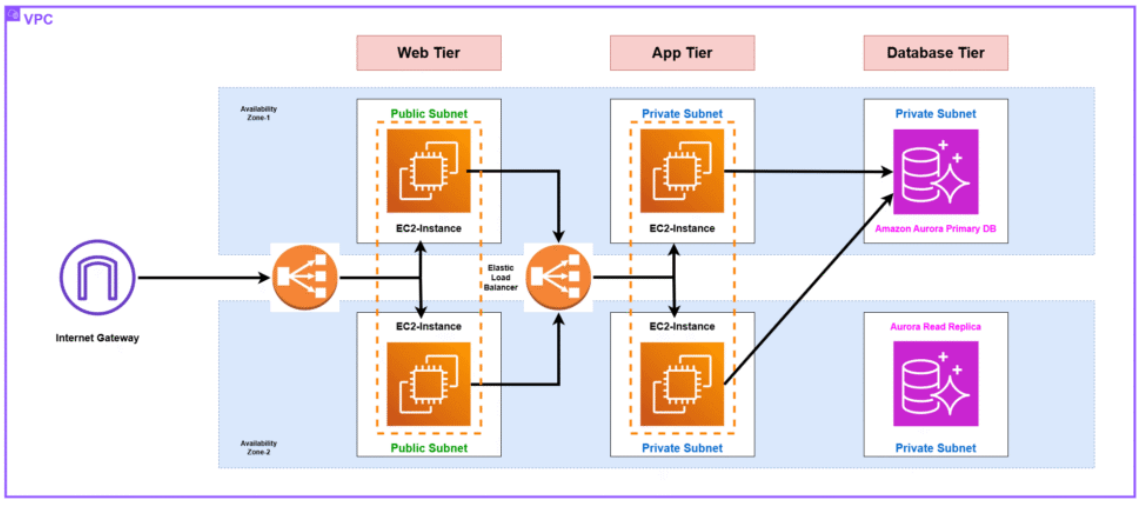
1. **Frontend Layer**
   - Hosted on an **EC2 instance**
   - Served using **Nginx**
   - Accessible via **EC2 public IP**
2. **Backend Layer**
   - Built using **Node.js and Express**
   - Runs on the same EC2 instance on port **3000**
   - Exposed through an **Application Load Balancer (ALB)**

3. **Database Layer**
    - Uses **Amazon RDS (MySQL)**
    - Hosted in a **private subnet**
    - Accessible only from the EC2 instance for security

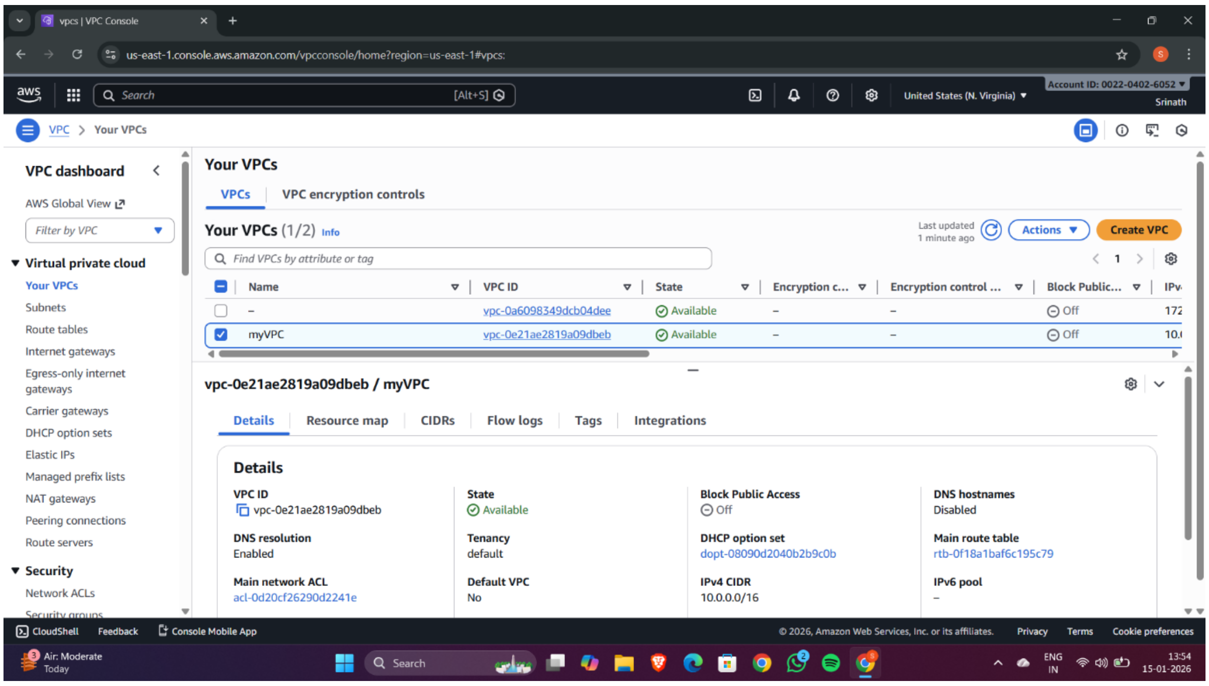
AWS Services Used

- **Amazon EC2** – Hosts frontend and backend application
- **Application Load Balancer (ALB)** – Routes client requests to backend
- **Amazon RDS (MySQL)** – Stores spare parts data
- **Amazon VPC** – Custom networking
- **Public & Private Subnets** – Network isolation
- **Security Groups** – Controlled inbound and outbound access
- **Internet Gateway** – Internet access for EC2

Flow of the Architecture.



Step 1: Creation of VPC



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#vpcs:

Your VPCs (1/2)

Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv
–	vpc-0a6098349dcb04dee	Available	–	–	Off	172
myVPC	vpc-0e21ae2819a09dbeb	Available	–	–	Off	10.(

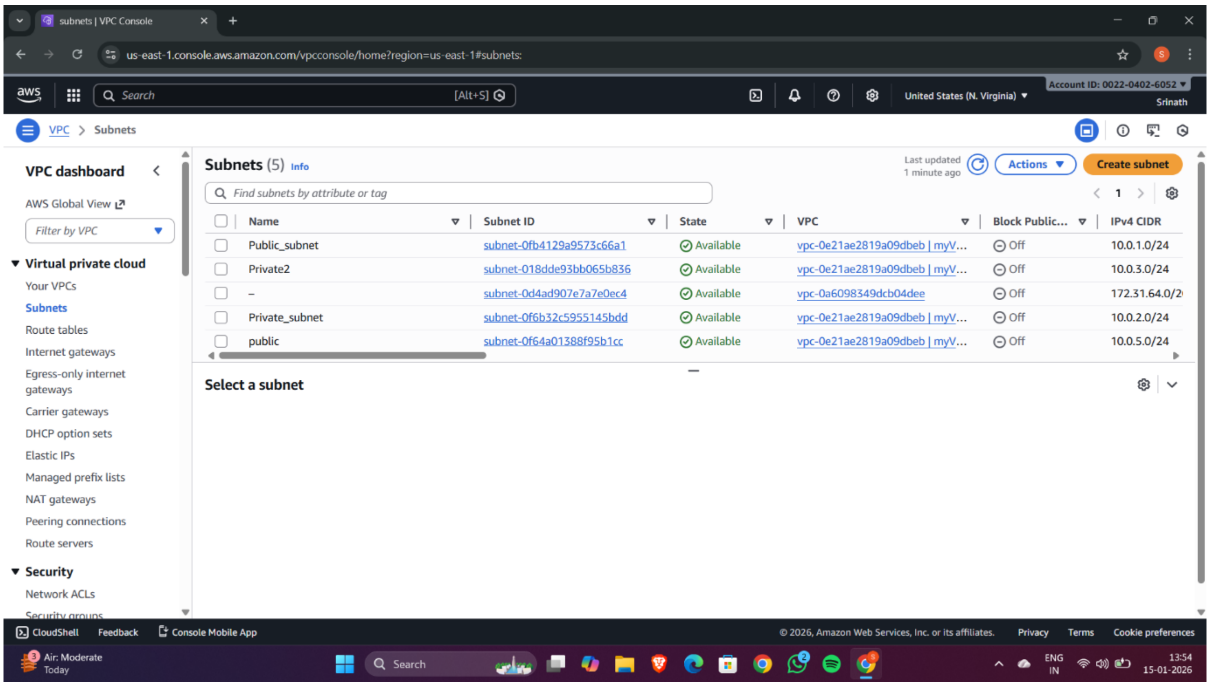
vpc-0e21ae2819a09dbeb / myVPC

Details

VPC ID vpc-0e21ae2819a09dbeb	**State** Available	**Block Public Access** Off	**DNS hostnames** Disabled
DNS resolution Enabled	**Tenancy** default	**DHCP option set** dopt-08090d2040b2b9c0b	**Main route table** rtb-0f18a1baf6c195c79
Main network ACL acl-0d20cf26290d2241e	**Default VPC** No	**IPv4 CIDR** 10.0.0.0/16	**IPv6 pool** –

- **CIDR:** 10.0.0.0/16

Step 2: Creation of Subnet



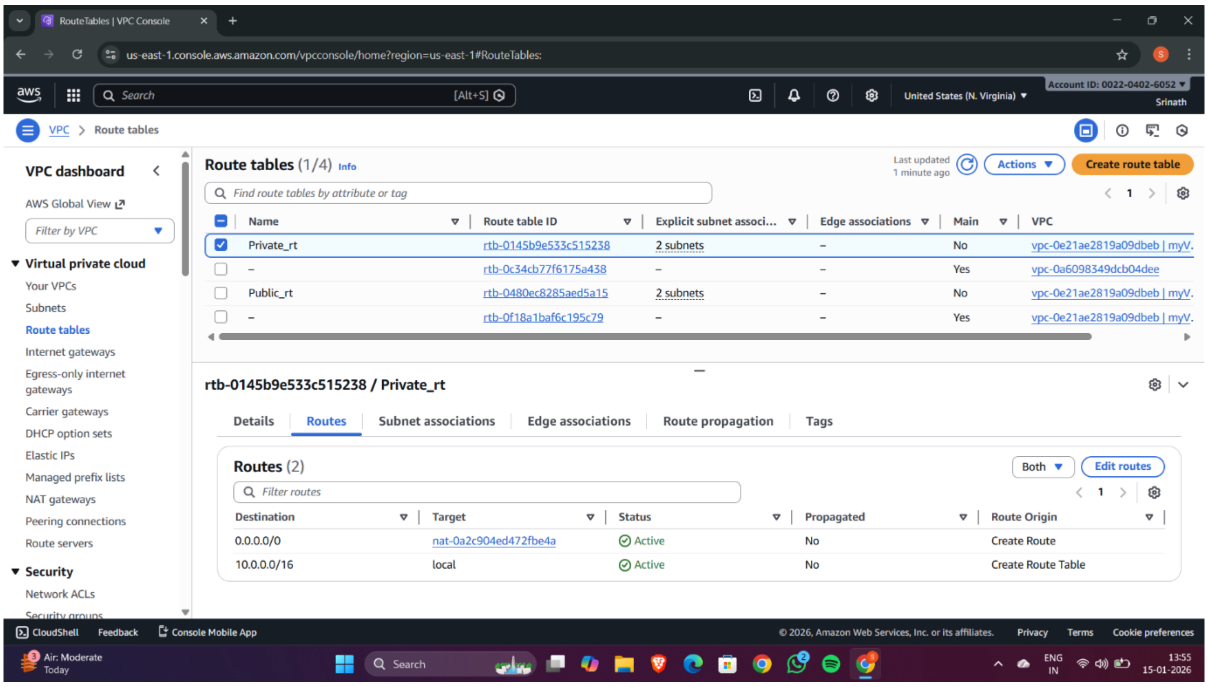
Subnets (5)

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
Public_subnet	subnet-0fb4129a9573c66a1	Available	vpc-0e21ae2819a09dbeb | myV...	Off	10.0.1.0/24
Private2	subnet-018dde93bb065b836	Available	vpc-0e21ae2819a09dbeb | myV...	Off	10.0.3.0/24
–	subnet-0d4ad907e7a7e0ec4	Available	vpc-0a6098349dcb04dee	Off	172.31.64.0/2
Private_subnet	subnet-0f6b32c5955145bdd	Available	vpc-0e21ae2819a09dbeb | myV...	Off	10.0.2.0/24
public	subnet-0f64a01388f95b1cc	Available	vpc-0e21ae2819a09dbeb | myV...	Off	10.0.5.0/24

- 2 Public Subnet and 2 Private Subnet

Step 3: Creation of Route tables

3.1: Create Private Route and Subnet associations with Private subnet.



Route tables (1/4)

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
Private_rt	rtb-0145b9e533c515238	2 subnets	–	No	vpc-0e21ae2819a09dbeb | myV.
–	rtb-0c34cb77f6175a438	–	–	Yes	vpc-0a6098349dcb04dee
Public_rt	rtb-0480ec8285aed5a15	2 subnets	–	No	vpc-0e21ae2819a09dbeb | myV.
–	rtb-0f18a1baf6c195c79	–	–	Yes	vpc-0e21ae2819a09dbeb | myV.

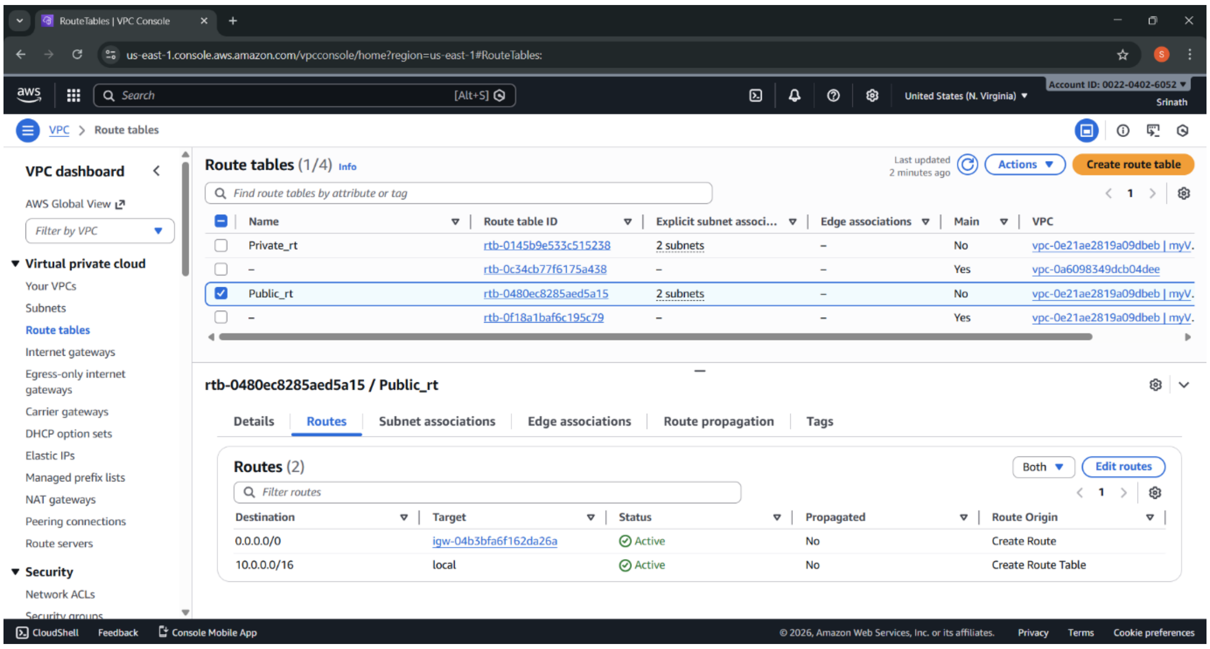
rtb-0145b9e533c515238 / Private_rt

Routes (2)

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	nat-0a2c904ed472fbe4a	Active	No	Create Route
10.0.0.0/16	local	Active	No	Create Route Table

- Attach with Nat gateway

3.2: Create Public Route and Subnet associations with Public subnet.



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTables:

Route tables (1/4)

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
Private_rt	rtb-0145b9e533c515238	2 subnets	–	No	vpc-0e21ae2819a09dbeb | myV.
–	rtb-0c34cb77f6175a438	–	–	Yes	vpc-0a6098349dcb04dee
Public_rt	rtb-0480ec8285aed5a15	2 subnets	–	No	vpc-0e21ae2819a09dbeb | myV.
–	rtb-0f18a1baf6c195c79	–	–	Yes	vpc-0e21ae2819a09dbeb | myV.

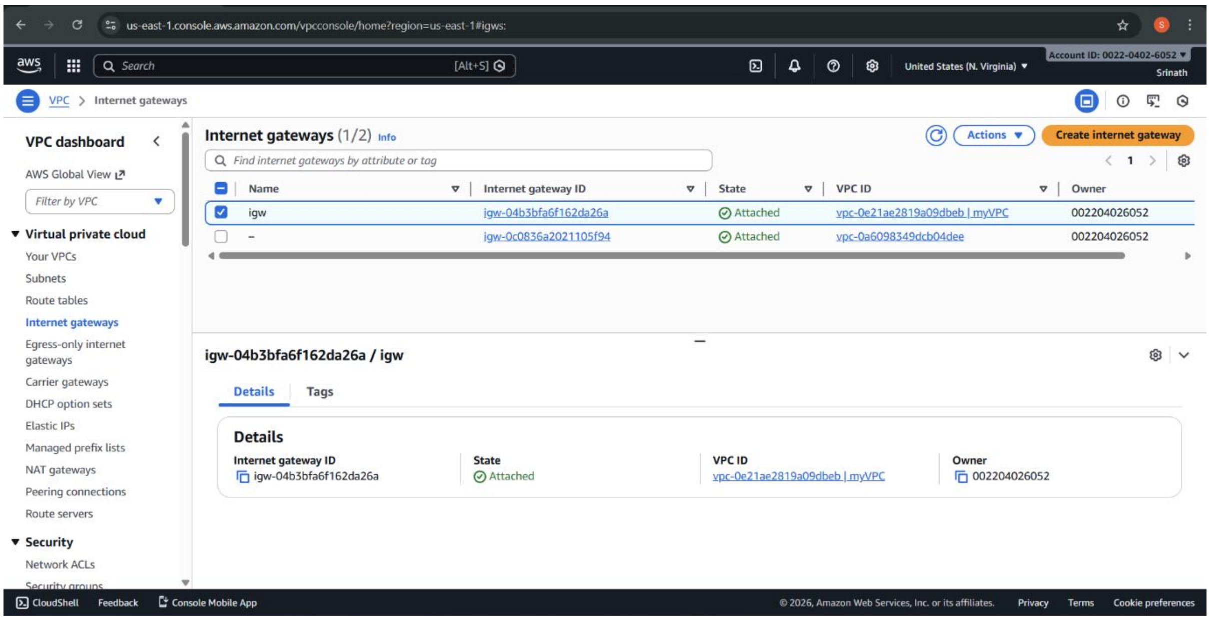
rtb-0480ec8285aed5a15 / Public_rt

Routes (2)

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-04b3bfa6f162da26a	Active	No	Create Route
10.0.0.0/16	local	Active	No	Create Route Table

- Attach with Internet gatway.

Step 4: Creation of Internet gateway and attach to VPC.



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#igws:

Internet gateways (1/2)

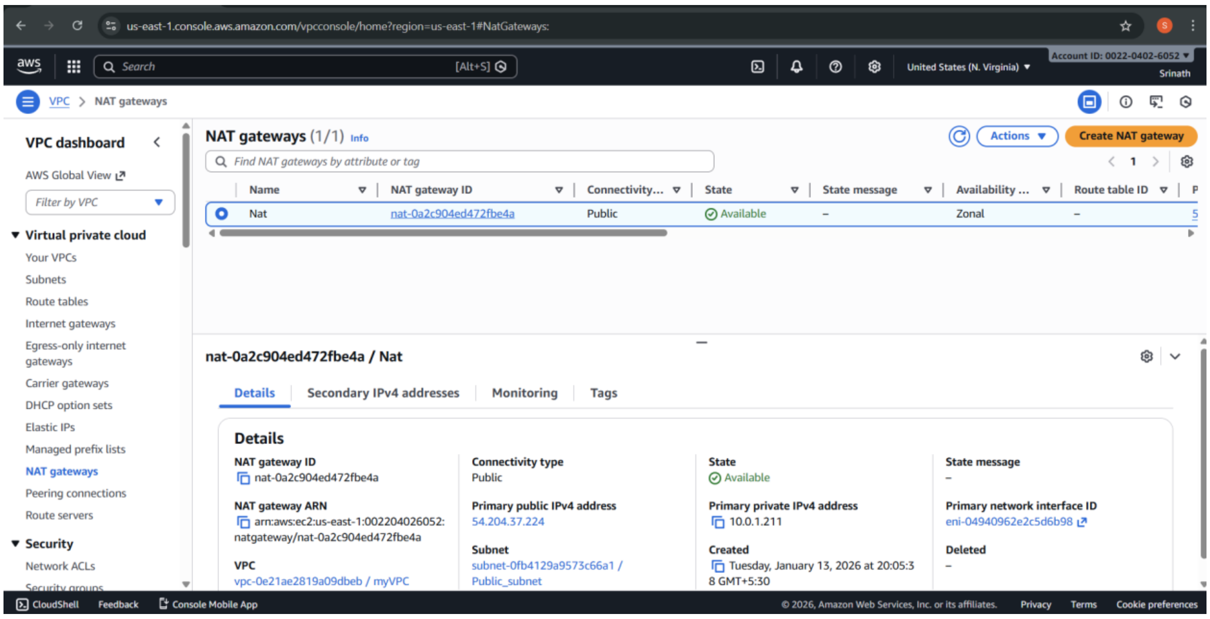
Name	Internet gateway ID	State	VPC ID	Owner
igw	igw-04b3bfa6f162da26a	Attached	vpc-0e21ae2819a09dbeb | myVPC	002204026052
–	igw-0c0836a2021105f94	Attached	vpc-0a6098349dcb04dee	002204026052

igw-04b3bfa6f162da26a / igw

Details

Internet gateway ID	State	VPC ID	Owner
igw-04b3bfa6f162da26a	Attached	vpc-0e21ae2819a09dbeb | myVPC	002204026052

Step 5: Creation of Nat gateway.



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#NatGateways:

VPC > NAT gateways

NAT gateways (1/1)

Name	NAT gateway ID	Connectivity...	State	State message	Availability ...	Route table ID
Nat	nat-0a2c904ed472fbe4a	Public	Available	–	Zonal	–

nat-0a2c904ed472fbe4a / Nat

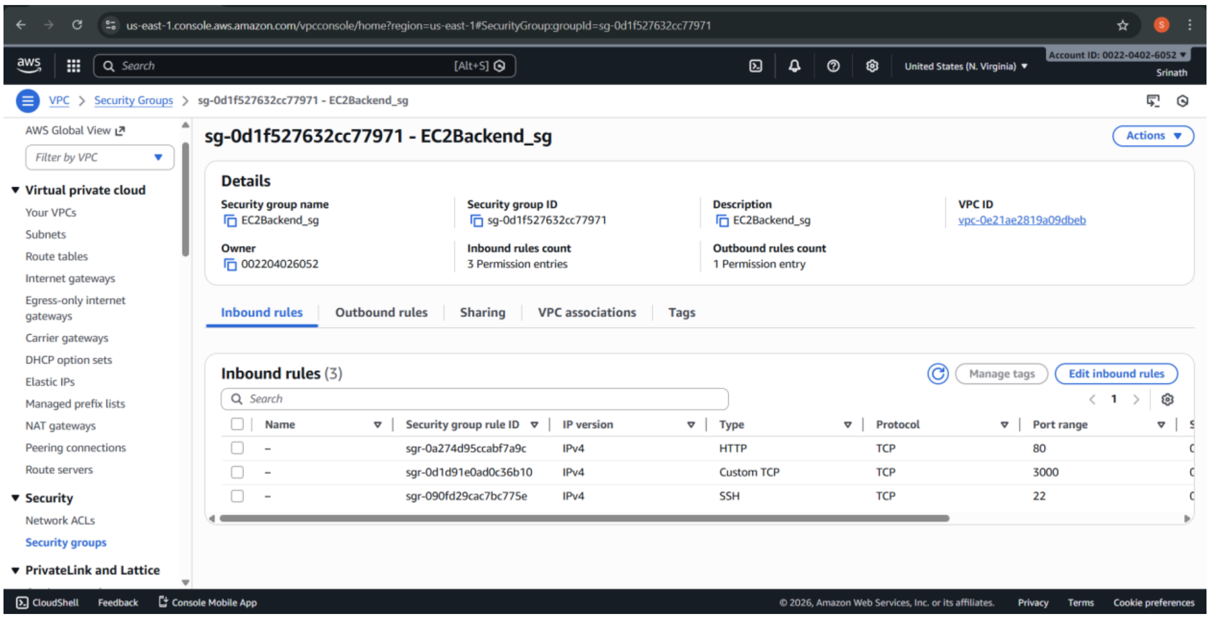
Details | Secondary IPv4 addresses | Monitoring | Tags

Details

NAT gateway ID: nat-0a2c904ed472fbe4a	Connectivity type: Public	State: Available	State message: –
NAT gateway ARN: arn:aws:ec2:us-east-1:002204026052:natgateway/nat-0a2c904ed472fbe4a	Primary public IPv4 address: 54.204.37.224	Primary private IPv4 address: 10.0.1.211	Primary network interface ID: eni-04940962e2c5d6b98
VPC: vpc-0e21ae2819a09dbeb / myVPC	Subnet: subnet-0fb4129a9573c66a1 / Public_subnet	Created: Tuesday, January 13, 2026 at 20:05:38 GMT+5:30	Deleted: –

Step 6: Creation of Security Groups.

6.1: EC2 Instance Security Groups.



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#SecurityGroup:groupId=sg-0d1f527632cc77971

VPC > Security Groups > sg-0d1f527632cc77971 - EC2Backend_sg

sg-0d1f527632cc77971 - EC2Backend_sg

Details

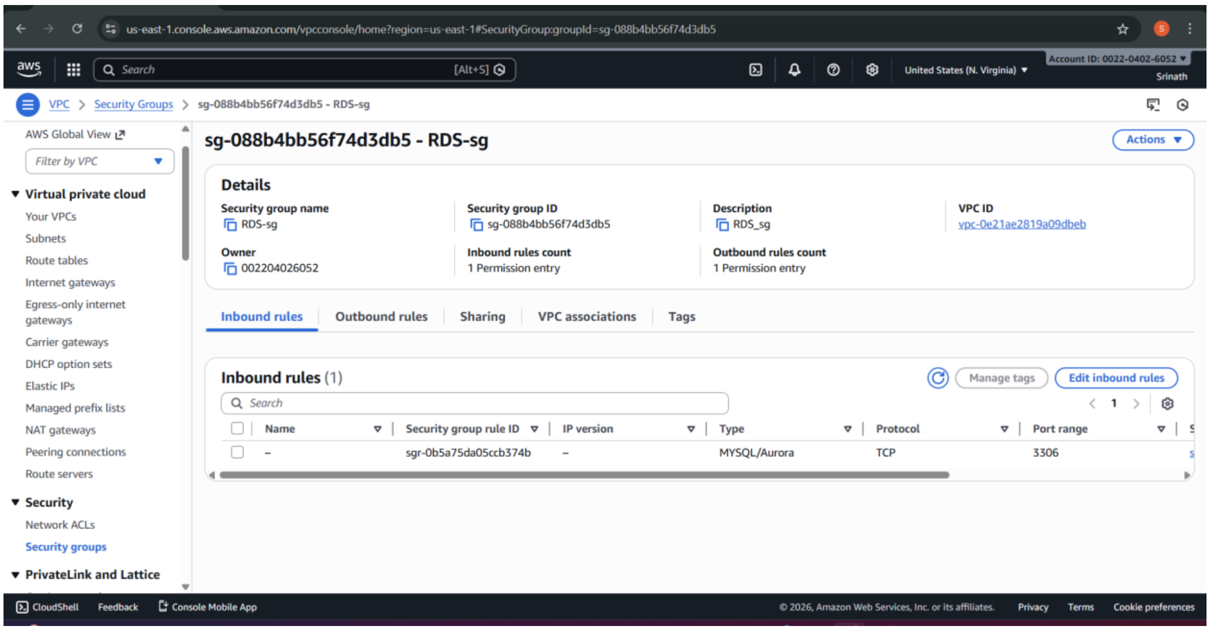
Security group name: EC2Backend_sg	Security group ID: sg-0d1f527632cc77971	Description: EC2Backend_sg	VPC ID: vpc-0e21ae2819a09dbeb
Owner: 002204026052	Inbound rules count: 3 Permission entries	Outbound rules count: 1 Permission entry	

Inbound rules | Outbound rules | Sharing | VPC associations | Tags

Inbound rules (3)

Name	Security group rule ID	IP version	Type	Protocol	Port range
–	sgr-0a274d95ccabf7a9c	IPv4	HTTP	TCP	80
–	sgr-0d1d91e0ad0c36b10	IPv4	Custom TCP	TCP	3000
–	sgr-090fd29cac7bc775e	IPv4	SSH	TCP	22

6.2: RDS Security Groups.


us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#SecurityGroup:groupId=sg-088b4bb56f74d3db5

VPC > Security Groups > sg-088b4bb56f74d3db5 - RDS-sg

sg-088b4bb56f74d3db5 - RDS-sg

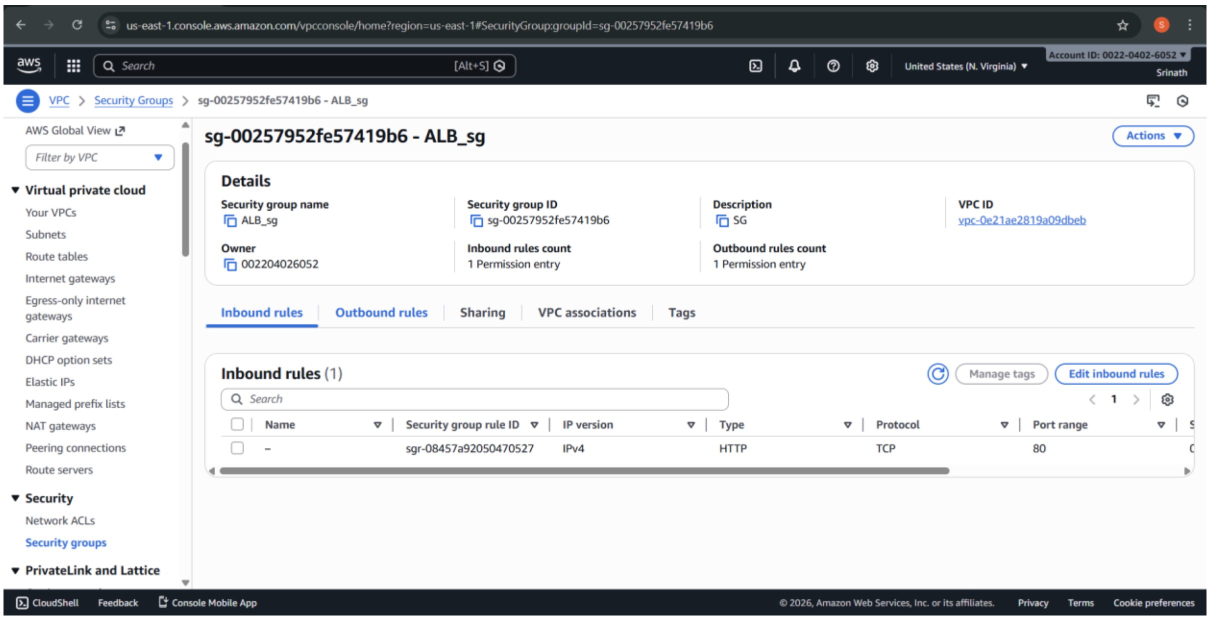
Details

Security group name	Security group ID	Description	VPC ID
RDS-sg	sg-088b4bb56f74d3db5	RDS_sg	vpc-0e21ae2819a09dbeb
Owner	**Inbound rules count**	**Outbound rules count**	
002204026052	1 Permission entry	1 Permission entry	

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range
–	sgr-0b5a75da05ccb374b	–	MYSQL/Aurora	TCP	3306

6.3: ALB Security Groups


us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#SecurityGroup:groupId=sg-00257952fe57419b6

VPC > Security Groups > sg-00257952fe57419b6 - ALB_sg

sg-00257952fe57419b6 - ALB_sg

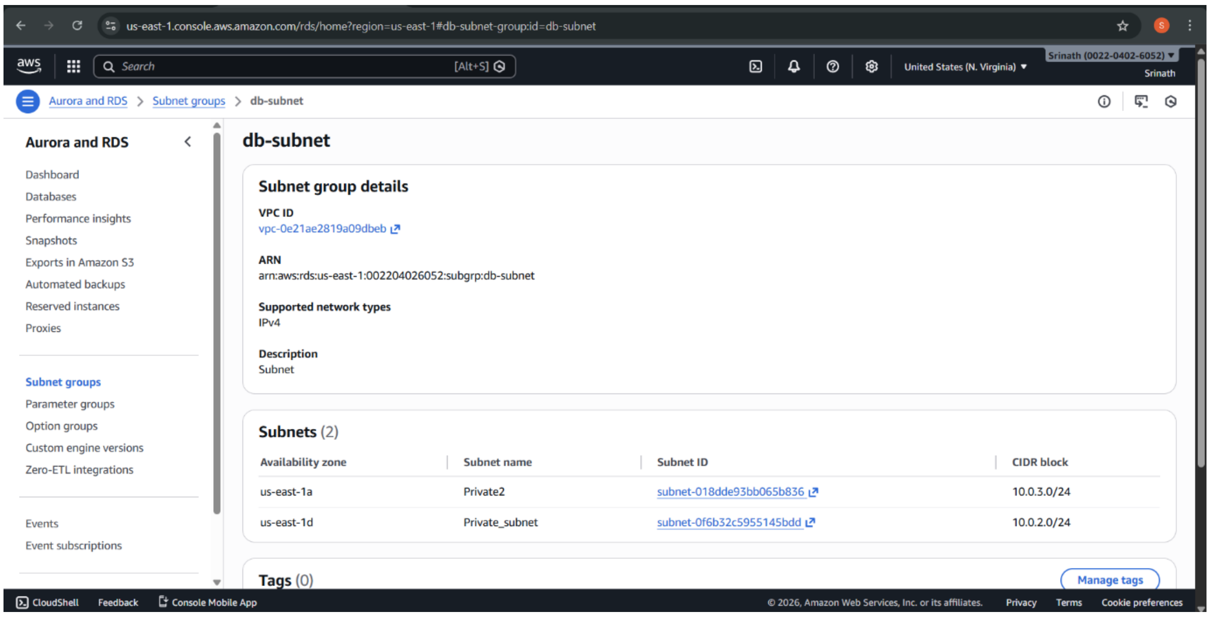
Details

Security group name	Security group ID	Description	VPC ID
ALB_sg	sg-00257952fe57419b6	SG	vpc-0e21ae2819a09dbeb
Owner	**Inbound rules count**	**Outbound rules count**	
002204026052	1 Permission entry	1 Permission entry	

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range
–	sgr-08457a92050470527	IPv4	HTTP	TCP	80

Step 6: Creation of RDS Subnet Groups.


us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#db-subnet-group:id=db-subnet

Aurora and RDS > Subnet groups > db-subnet

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

db-subnet

Subnet group details

VPC ID
vpc-0e21ae2819a09dbeb

ARN
arn:aws:rds:us-east-1:002204026052:subgrp:db-subnet

Supported network types
IPv4

Description
Subnet

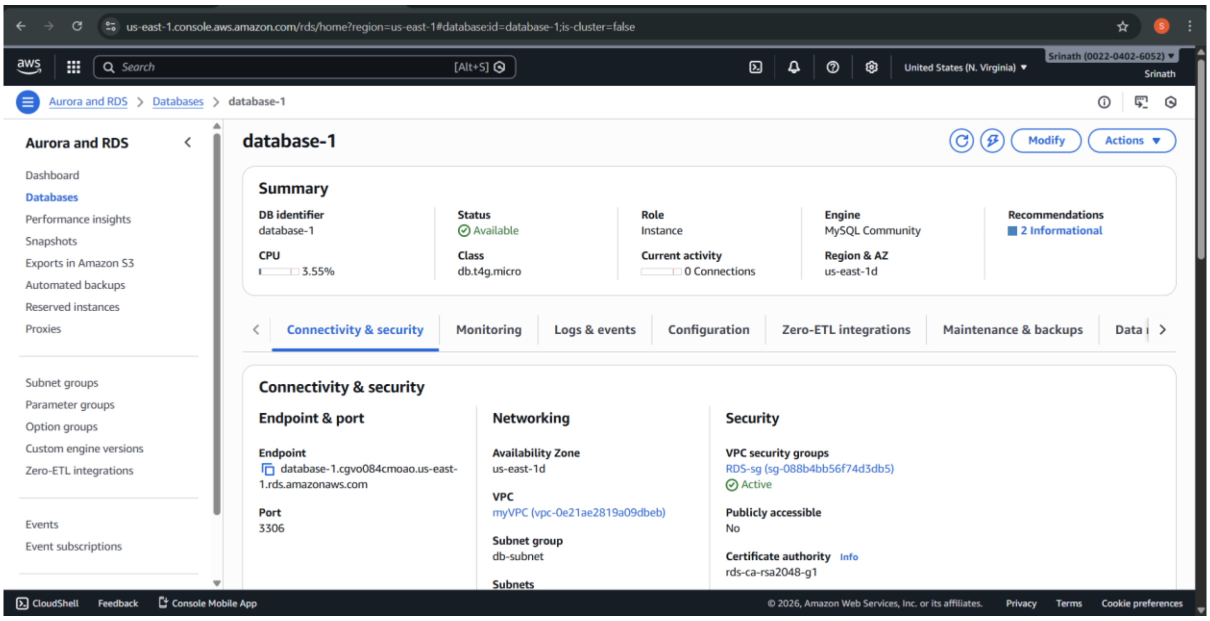
Subnets (2)

Availability zone	Subnet name	Subnet ID	CIDR block
us-east-1a	Private2	subnet-018dde93bb065b836	10.0.3.0/24
us-east-1d	Private_subnet	subnet-0f6b32c5955145bdd	10.0.2.0/24

Tags (0)

- Grouping the 2 Private Subnet.

Step 7: Creation of RDS (MySQL).


us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#database:id=database-1;is-cluster=false

Aurora and RDS > Databases > database-1

database-1

Summary

DB identifier	Status	Role	Engine	Recommendations
database-1	Available	Instance	MySQL Community	2 Informational

CPU	Class	Current activity	Region & AZ
3.55%	db.t4g.micro	0 Connections	us-east-1d

Connectivity & security | Monitoring | Logs & events | Configuration | Zero-ETL integrations | Maintenance & backups | Data

Connectivity & security

Endpoint & port

Endpoint
database-1.cgvo084cmoao.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1d

VPC
myVPC (vpc-0e21ae2819a09dbeb)

Subnet group
db-subnet

Subnets

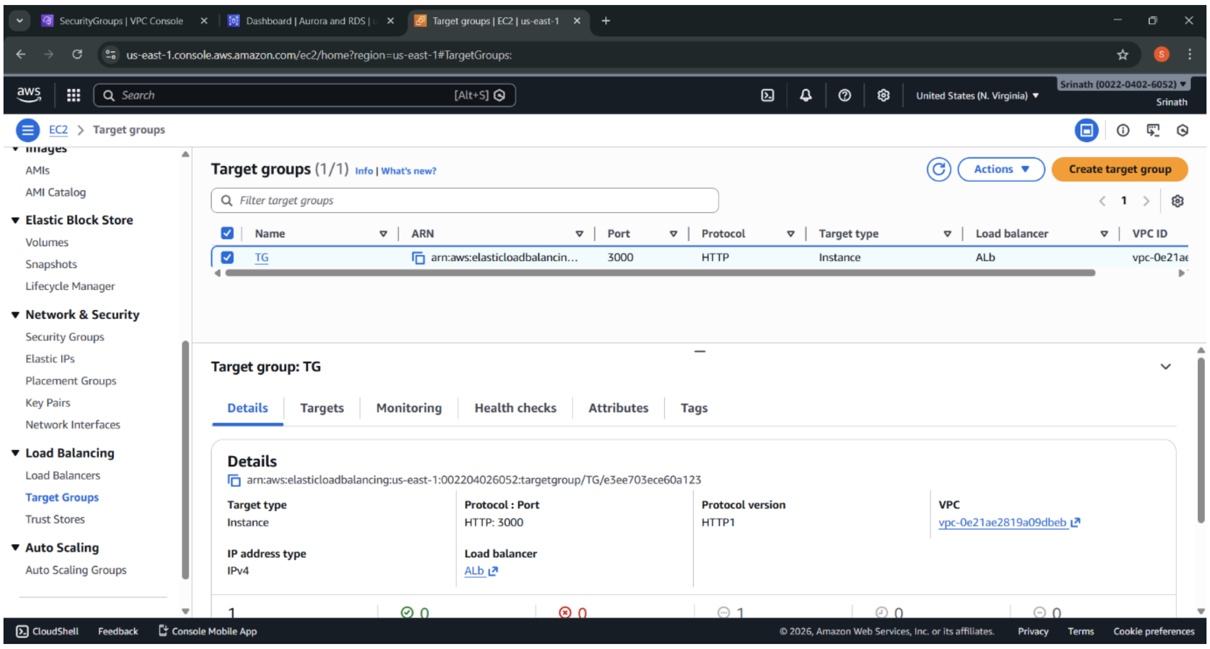
Security

VPC security groups
RDS-sg (sg-088b4bb56f74d3db5)
Active

Publicly accessible
No

Certificate authority Info
rds-ca-rsa2048-g1

Step 8: Creation of Target Groups.


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#TargetGroups:

EC2 > Target groups

Target groups (1/1)

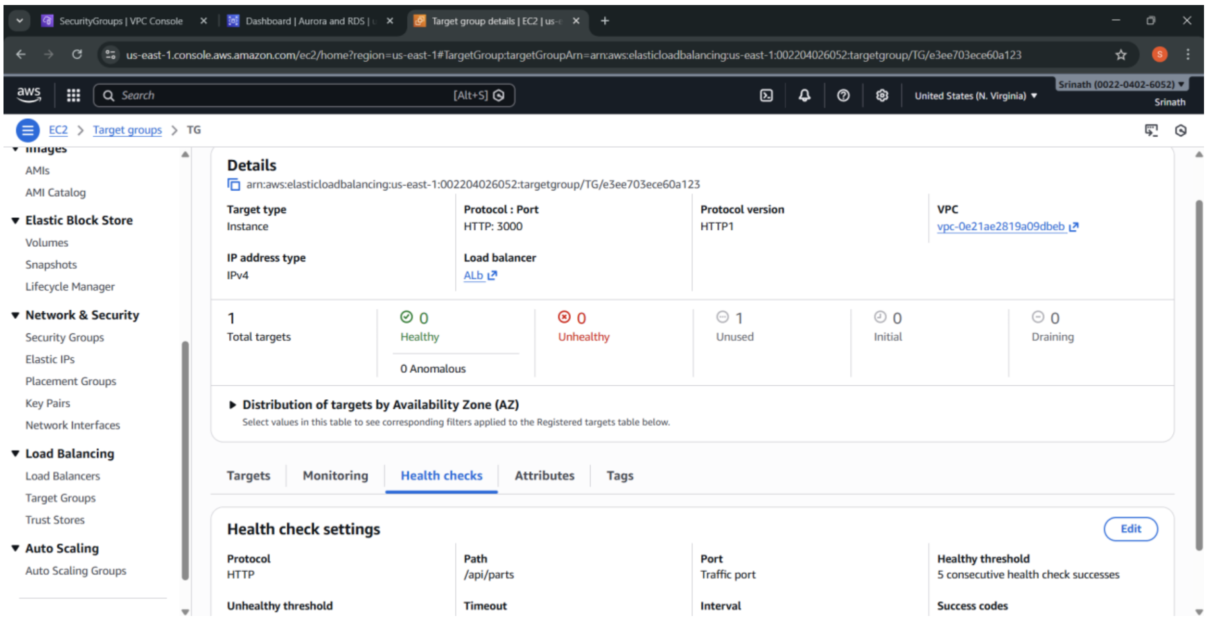
Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
TG	arn:aws:elasticloadbalancin…	3000	HTTP	Instance	ALb	vpc-0e21ae

Target group: TG

Details

arn:aws:elasticloadbalancing:us-east-1:002204026052:targetgroup/TG/e3ee703ece60a123

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 3000	HTTP1	vpc-0e21ae2819a09dbeb
IP address type	**Load balancer**		
IPv4	ALb		


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#TargetGroup:targetGroupArn=arn:aws:elasticloadbalancing:us-east-1:002204026052:targetgroup/TG/e3ee703ece60a123

EC2 > Target groups > TG

Details

arn:aws:elasticloadbalancing:us-east-1:002204026052:targetgroup/TG/e3ee703ece60a123

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 3000	HTTP1	vpc-0e21ae2819a09dbeb
IP address type	**Load balancer**		
IPv4	ALb		

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
1	0	0	1	0	0
	0 Anomalous				

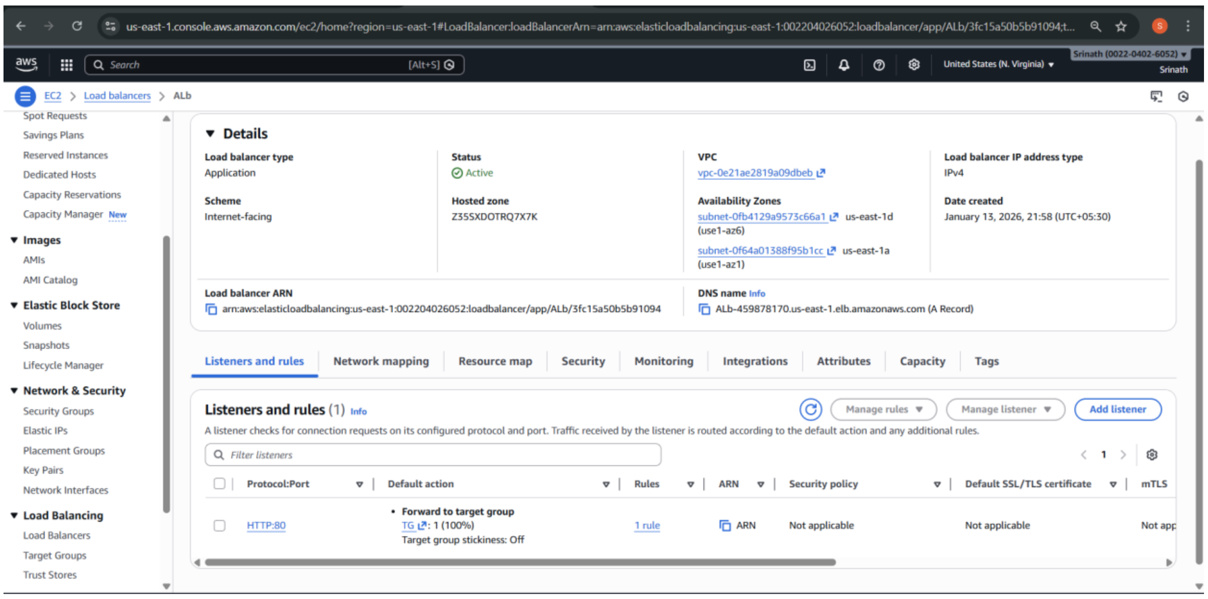
Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Health check settings

Protocol	Path	Port	Healthy threshold
HTTP	/api/parts	Traffic port	5 consecutive health check successes
Unhealthy threshold	**Timeout**	**Interval**	**Success codes**

- Port Number 3000
- Health checks Path /api/parts

Step 9: Creation of Application Load balancer.



Details

Load balancer type: Application
Status: Active
VPC: vpc-0e21ae2819a09dbeb
Load balancer IP address type: IPv4
Scheme: Internet-facing
Hosted zone: Z35SXDOTRQ7X7K
Availability Zones: subnet-0fb4129a9573c66a1 us-east-1d (use1-az6); subnet-0f64a01388f95b1cc us-east-1a (use1-az1)
Date created: January 13, 2026, 21:58 (UTC+05:30)
Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:002204026052:loadbalancer/app/ALb/3fc15a50b5b91094
DNS name: ALb-459878170.us-east-1.elb.amazonaws.com (A Record)

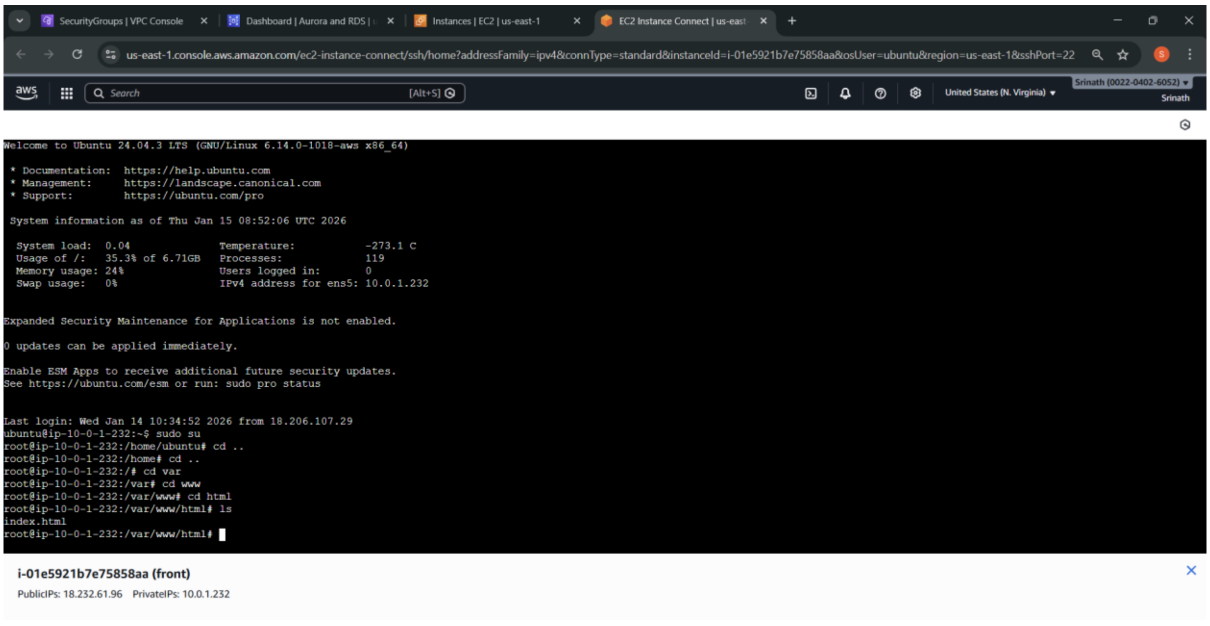
Listeners and rules (1)

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	mTLS
HTTP:80	• Forward to target group TG: 1 (100%) Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not app

Step 10: Creation of EC2.

10.1: Public Instance.


SecurityGroups | VPC Console
Dashboard | Aurora and RDS
Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-01e5921b7e75858aa&osUser=ubuntu&region=us-east-1&sshPort=22
Srinath (0022-0402-6052)
United States (N. Virginia)

```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Thu Jan 15 08:52:06 UTC 2026

  System load:  0.04              Temperature:           -273.1 C
  Usage of /:   35.3% of 6.71GB   Processes:             119
  Memory usage: 24%               Users logged in:       0
  Swap usage:   0%                IPv4 address for ens5: 10.0.1.232


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Wed Jan 14 10:34:52 2026 from 18.206.107.29
ubuntu@ip-10-0-1-232:~$ sudo su
root@ip-10-0-1-232:/home/ubuntu# cd ..
root@ip-10-0-1-232:/home# cd ..
root@ip-10-0-1-232:/# cd var
root@ip-10-0-1-232:/var# cd www
root@ip-10-0-1-232:/var/www# cd html
root@ip-10-0-1-232:/var/www/html# ls
index.html
root@ip-10-0-1-232:/var/www/html# 
```

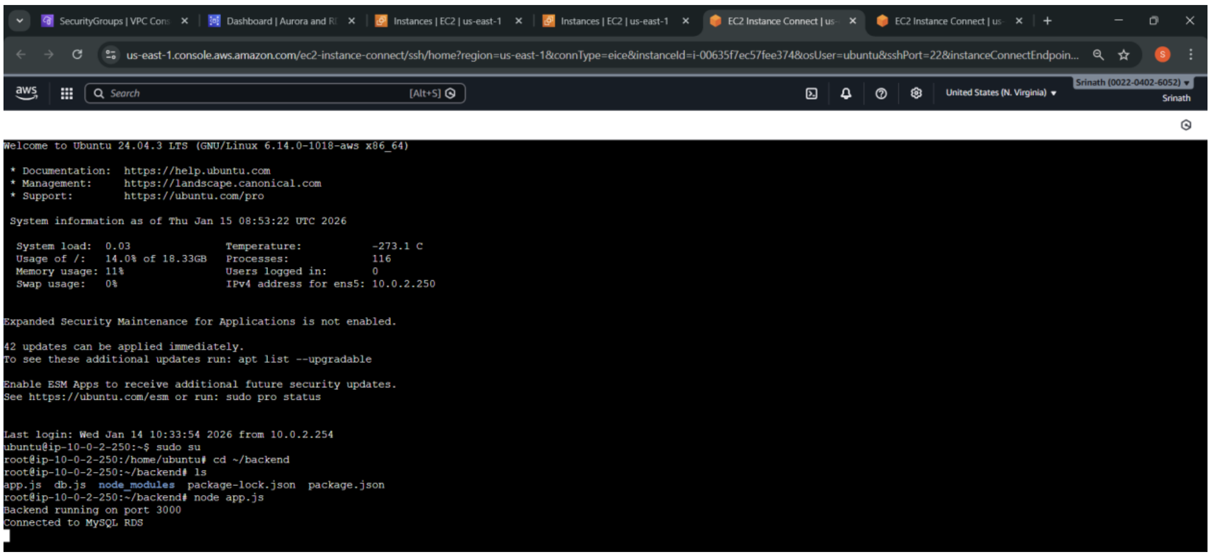
i-01e5921b7e75858aa (front)
PublicIPs: 18.232.61.96   PrivateIPs: 10.0.1.232

Commands:

- **Apt install nginx**
- **Cd var/www/html/ (Create index.html file)**

- **Systemctl start nginx**

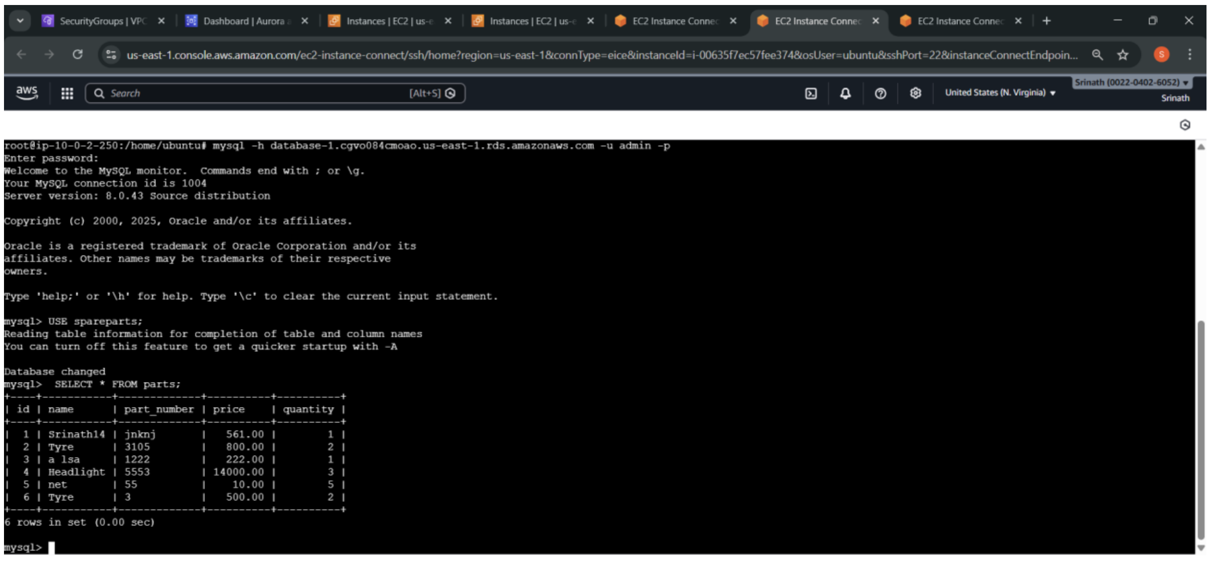
10.2: Private Instance.


SecurityGroups | VPC Cons
Dashboard | Aurora and R
Instances | EC2 | us-east-1
Instances | EC2 | us-east-1
EC2 Instance Connect | us-
EC2 Instance Connect | us-
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=eice&instanceId=i-00635f7ec57fee374&osUser=ubuntu&sshPort=22&instanceConnectEndpoin...
Search [Alt+S]
United States (N. Virginia)
Srinath (0022-0402-6052)
Srinath

```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Thu Jan 15 08:53:22 UTC 2026

  System load:  0.03               Temperature:           -273.1 C
  Usage of /:   14.0% of 18.33GB   Processes:             116
  Memory usage: 11%                Users logged in:       0
  Swap usage:   0%                 IPv4 address for ens5: 10.0.2.250


Expanded Security Maintenance for Applications is not enabled.

42 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Wed Jan 14 10:33:54 2026 from 10.0.2.254
ubuntu@ip-10-0-2-250:~$ sudo su
root@ip-10-0-2-250:/home/ubuntu# cd ~/backend
root@ip-10-0-2-250:~/backend# ls
app.js  db.js  node_modules  package-lock.json  package.json
root@ip-10-0-2-250:~/backend# node app.js
Backend running on port 3000
Connected to MySQL RDS
```

Commands:

- **Mkdir backend (folder name)**
- **Cd backend**
- **apt install nodejs npm -y**
- **npm install express mysql2 cors**
- **vi app.js (node.js code)**
- **vi db.js (database code)**
- **node app.js**


SecurityGroups | VPC
Dashboard | Aurora
Instances | EC2 | us-e
Instances | EC2 | us-e
EC2 Instance Connec
EC2 Instance Connec
EC2 Instance Connec
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=eice&instanceId=i-00635f7ec57fee374&osUser=ubuntu&sshPort=22&instanceConnectEndpoin...
Search
[Alt+S]
United States (N. Virginia)
Srinath (0022-0402-6052)
Srinath

```
root@ip-10-0-2-250:/home/ubuntu# mysql -h database-1.cgvo084cmoao.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1004
Server version: 8.0.43 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE spareparts;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>  SELECT * FROM parts;
+----+-----------+-------------+----------+----------+
| id | name      | part_number | price    | quantity |
+----+-----------+-------------+----------+----------+
|  1 | Srinath14 | jnknj       |   561.00 |        1 |
|  2 | Tyre      | 3105        |   800.00 |        2 |
|  3 | a lsa     | 1222        |   222.00 |        1 |
|  4 | Headlight | 5553        | 14000.00 |        3 |
|  5 | net       | 55          |    10.00 |        5 |
|  6 | Tyre      | 3           |   500.00 |        2 |
+----+-----------+-------------+----------+----------+
6 rows in set (0.00 sec)

mysql> 
```

Commands:

- mysql -h database-1.cpmm06ss6dr0.eu-north-1.rds.amazonaws.com -u admin -p (Connect the mysql database with private instance).

Step 11: Code

11.1: index.html.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Spare Parts Catalog</title>

  <style>
    * {
      box-sizing: border-box;
      font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
    }

    body {
      margin: 0;
      height: 100vh;
      background: linear-gradient(135deg, #667eea, #764ba2);
      display: flex;
      align-items: center;
```

```
 justify-content: center;
}

.card {
 background: #ffffff;
 width: 420px;
 padding: 30px;
 border-radius: 12px;
 box-shadow: 0 15px 35px rgba(0,0,0,0.2);
 animation: fadeIn 0.8s ease;
}

h2 {
 text-align: center;
 margin-bottom: 20px;
 color: #333;
}

label {
 font-size: 14px;
 font-weight: 600;
 color: #555;
}

input {
 width: 100%;
 padding: 10px;
 margin: 6px 0 15px;
 border-radius: 6px;
 border: 1px solid #ccc;
 font-size: 14px;
 transition: 0.3s;
}

input:focus {
 outline: none;
 border-color: #667eea;
 box-shadow: 0 0 5px rgba(102,126,234,0.5);
}
```

```
    button {
      width: 100%;
      padding: 12px;
      background: linear-gradient(135deg, #667eea, #764ba2);
      border: none;
      border-radius: 6px;
      color: white;
      font-size: 16px;
      font-weight: bold;
      cursor: pointer;
      transition: 0.3s;
    }

    button:hover {
      transform: translateY(-2px);
      box-shadow: 0 10px 20px rgba(0,0,0,0.2);
    }

    .footer {
      text-align: center;
      font-size: 12px;
      margin-top: 15px;
      color: #777;
    }

    @keyframes fadeIn {
      from { opacity: 0; transform: translateY(20px); }
      to { opacity: 1; transform: translateY(0); }
    }
  </style>
</head>

<body>

  <div class="card">
    <h2>Spare Parts Entry</h2>

    <label>Part Name</label>
    <input id="name" placeholder="Enter part name">
  <label>Part Number</label>
  <input id="number" placeholder="Enter part number">

  <label>Price</label>
  <input id="price" type="number" placeholder="Enter price">

  <label>Quantity</label>
  <input id="qty" type="number" placeholder="Enter quantity">

  <button onclick="savePart()">Save Part</button>

  <div class="footer">
    Spare Parts Management System
  </div>
</div>

<script>
  function savePart() {
    fetch("http://alb-459878170.us-east-1.elb.amazonaws.com/api/parts", {
      method: "POST",
      headers: {
        "Content-Type": "application/json"
      },
      body: JSON.stringify({
        name: document.getElementById("name").value,
        number: document.getElementById("number").value,
        price: document.getElementById("price").value,
        quantity: document.getElementById("qty").value
      })
    })
    .then(res => res.json())
    .then(data => {
      alert(data.message);
      document.getElementById("name").value = "";
      document.getElementById("number").value = "";
      document.getElementById("price").value = "";
      document.getElementById("qty").value = "";
    })
    .catch(err => alert("Error saving part"));
  }
  </script>

</body>
</html>
```

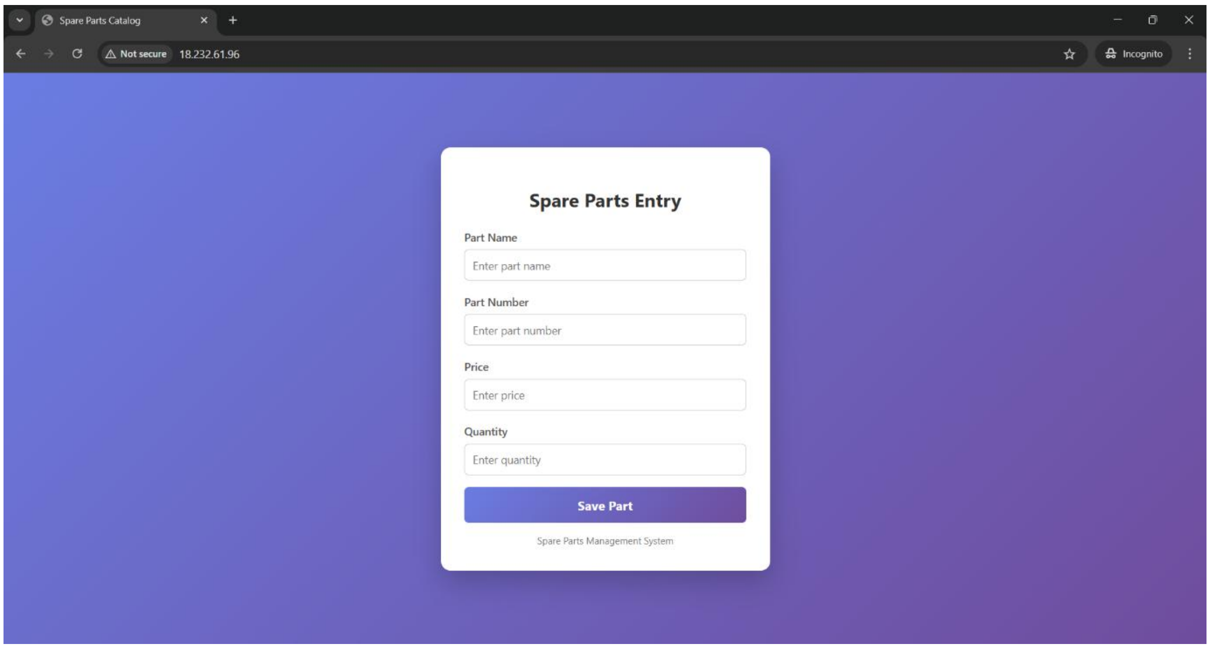
11.2: app.js.

```
const express = require("express");
const cors = require("cors");
const mysql = require("mysql2");

const app = express();

// Middlewares
app.use(cors());
app.use(express.json());

// MySQL RDS connection
const db = mysql.createConnection({
  host: "database-1.cgvo084cmoao.us-east-1.rds.amazonaws.com",
  user: "admin",
  password: "password123",
  database: "spareparts"
});

// Connect to DB
db.connect(err => {
  if (err) {
    console.error("DB connection failed:", err);
  } else {
    console.log("Connected to MySQL RDS");
  }
});
app.post("/api/parts", (req, res) => {
  const { name, number, price, quantity } = req.body;

  if (!name || !number || !price || !quantity) {
    return res.status(400).json({ message: "All fields are required" });
  }
  const sql =
    "INSERT INTO parts (name, part_number, price, quantity) VALUES (?, ?, ?,
?)";

  db.query(sql, [name, number, price, quantity], (err, result) => {
    if (err) {
      console.error("Insert error:", err);
      return res.status(500).json({ message: "DB error" });
    }
    res.json({ message: "Saved" });
  });
});
app.get("/api/parts", (req, res) => {
  db.query("SELECT * FROM parts", (err, rows) => {
    if (err) {
      console.error("Fetch error:", err);
      return res.status(500).json({ message: "DB error" });
    }
    res.json(rows);
  });
});

// Start server
app.listen(3000, () => {
  console.log("Backend running on port 3000");
});
```

11.3: db.js.

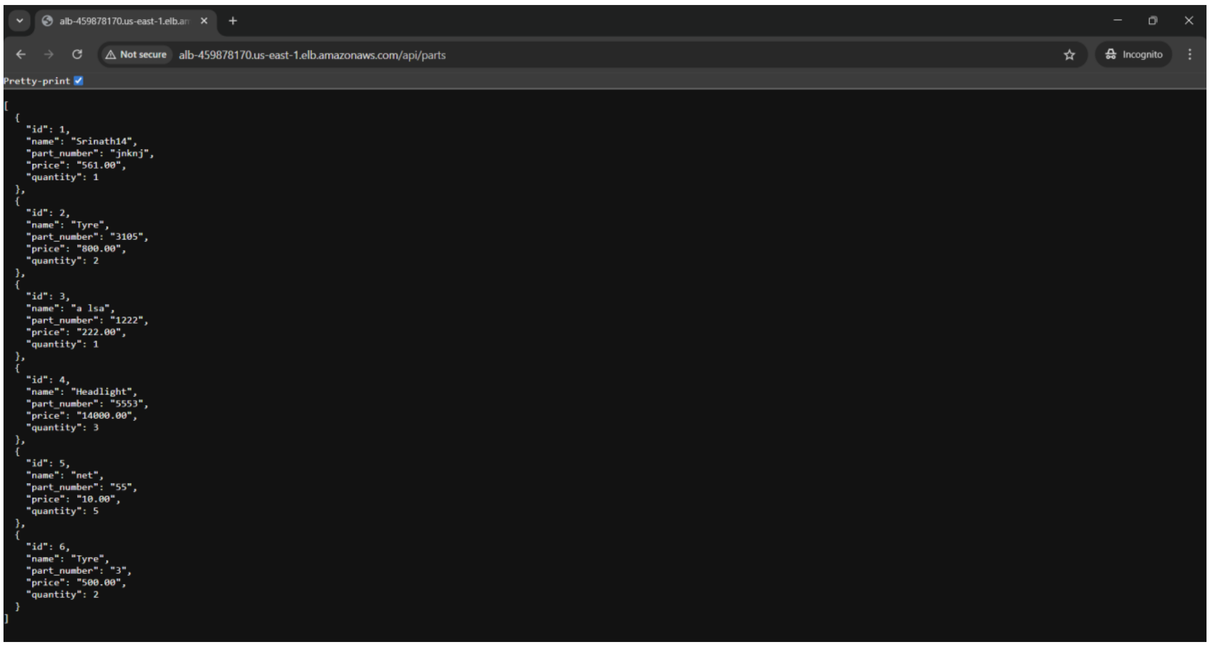
```
const mysql = require("mysql2");

const db = mysql.createConnection({
  host: "database-1.cgvo084cmoao.us-east-1.rds.amazonaws.com",
  user: "admin",
  password: "password123",
  database: "spareparts"
});

db.connect(err => {
  if (err) {
    console.error("DB Connection Failed:", err);
  } else {
    console.log("Connected to MySQL RDS");
  }
});
```

Frontend Output.



- By using frontend instance (Public Instance) IP address.

Data Base Output:


alb-459878170.us-east-1.elb.ar
Not secure
alb-459878170.us-east-1.elb.amazonaws.com/api/parts
Incognito
Pretty-print

```
[
  {
    "id": 1,
    "name": "Srinath14",
    "part_number": "jnknj",
    "price": "561.00",
    "quantity": 1
  },
  {
    "id": 2,
    "name": "Tyre",
    "part_number": "3105",
    "price": "800.00",
    "quantity": 2
  },
  {
    "id": 3,
    "name": "a lsa",
    "part_number": "1222",
    "price": "222.00",
    "quantity": 1
  },
  {
    "id": 4,
    "name": "Headlight",
    "part_number": "5553",
    "price": "14000.00",
    "quantity": 3
  },
  {
    "id": 5,
    "name": "net",
    "part_number": "55",
    "price": "10.00",
    "quantity": 5
  },
  {
    "id": 6,
    "name": "Tyre",
    "part_number": "3",
    "price": "500.00",
    "quantity": 2
  }
]
```

- By using load balancer DNS address.