ElectroShop – End-to-End Cloud-Native E-Commerce Platform

ElectroShop is a modern e-commerce application with full CI/CD automation, secure deployment on AWS, containerized microservices architecture, centralized logging, monitoring, and robust cloud infrastructure.

Technologies and Tools Used

Category	Tools / Technologies
Frontend	React.js, HTML/CSS/JS
Backend	Node.js
Database	MongoDB Atlas, MongoDB Compass
Containerization	Docker, Docker Compose
Cloud Infrastructure	AWS VPC, Subnets, ECS Fargate, ECR, ALB
CI/CD	GitHub Actions, Trivy
Monitoring & Logging	AWS CloudWatch, AWS SNS
Security	AWS Secrets Manager, AWS WAF & Shield
Development Tools	VS Code, Git & GitHub

System Architecture Overview

The architecture follows a microservices deployment on AWS ECS Fargate with the following components:
- Frontend (React) and Backend (Node.js/Express) running as separate containers.
- MongoDB Atlas as the managed database.
- Load Balancing via ALB with routing rules for frontend and backend.
- CI/CD via GitHub Actions with vulnerability scanning.
- Monitoring and alerts using CloudWatch and SNS.

Phase 1. Containerisation and AWS Deployment.

1: Project Setup and Containerisation

In this phase, the core application (frontend and backend) was developed and containerized for local and eventual cloud deployment.
Key tasks included:
- Installing MongoDB and MongoDB Compass for local database testing.
- Running backend and frontend applications on localhost (port 3000).
- Creating Dockerfiles for both frontend and backend applications.
- Configuring docker-compose for local orchestration.

Backend-

Install mongo db & mongodb compass for the database.

```
PS C:\Windows\system32> mongod --version
db version v8.1.2
Build Info: {
    "version": "8.1.2",
    "gitVersion": "bcba0709b2665cca6b1b44a1803a6f8249e6ee39",
    "modules": [],
    "allocator": "tcmalloc-gperf",
    "environment": {
        "distmod": "windows"
    }
}
```

```
PS C:\Windows\system32> choco install mongodb-compass -y
Chocolatey v2.3.0
Installing the following packages:
mongodb-compass
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading mongodb-compass 1.46.5... 100%

mongodb-compass v1.46.5 [Approved]
mongodb-compass package files install completed. Performing other installation steps.
Downloading mongodb-compass 64 bit
  from 'https://github.com/mongodb-js/compass/releases/download/v1.46.5/mongodb-compass-1.46.5-win32-x64.msi'
Progress: 100% - Completed download of C:\Users\THE SHIKSHAK\AppData\Local\Temp\chocolatey\mongodb-compass\1.46.5\mongod
b-compass-1.46.5-win32-x64.msi (150.14 MB).
Download of mongodb-compass-1.46.5-win32-x64.msi (150.14 MB) completed.
Hashes match.
Installing mongodb-compass...
mongodb-compass has been installed.
  mongodb-compass may be able to be automatically uninstalled.
 The install of mongodb-compass was successful.
  Software installed as 'MSI', install location is likely default.

Chocolatey installed 1/1 packages.
 See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
```

Running backend application on local machine

```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\backend> npm install
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'bson@6.10.4',
npm WARN EBADENGINE   required: { node: '>=16.20.1' },
npm WARN EBADENGINE   current: { node: 'v16.17.0', npm: '8.19.2' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'mongodb@6.17.0',
npm WARN EBADENGINE   required: { node: '>=16.20.1' },
npm WARN EBADENGINE   current: { node: 'v16.17.0', npm: '8.19.2' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'mongoose@8.16.1',
npm WARN EBADENGINE   required: { node: '>=16.20.1' },
npm WARN EBADENGINE   current: { node: 'v16.17.0', npm: '8.19.2' }
```

```
README.md    .env
backend > .env
1   MONGO_URI=mongodb://localhost:27017/electromart
2   PORT=5000
```

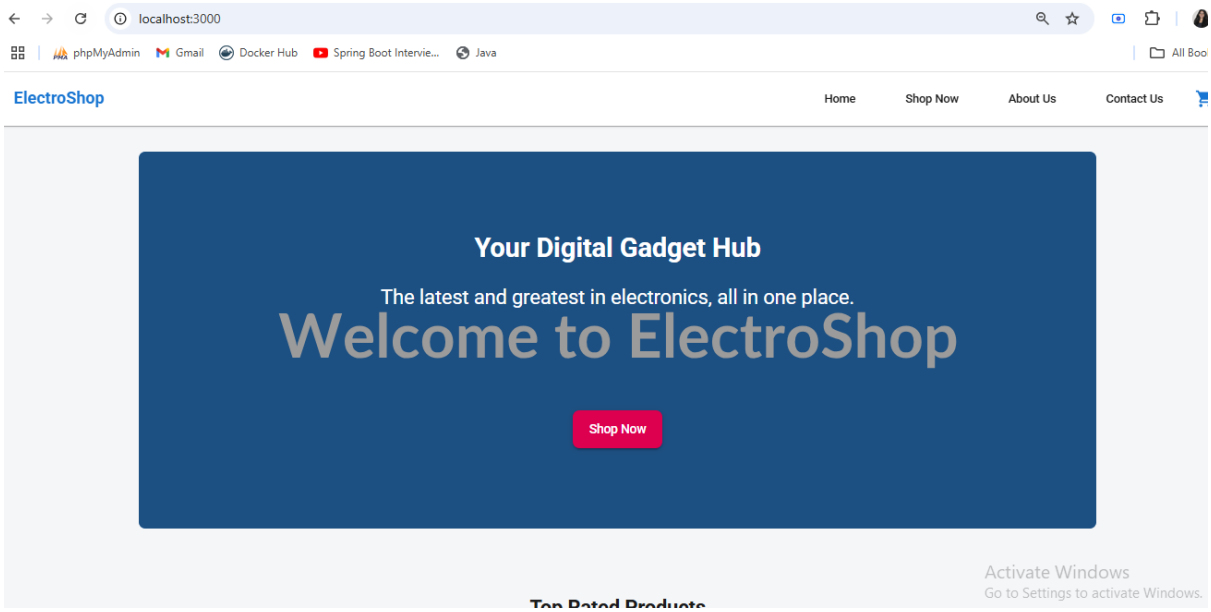
```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\backend> npm run dev

> electromart-backend@1.0.0 dev
> nodemon server.js

[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
(node:11872) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since No
iver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:11872) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect si
.js Driver version 4.0.0 and will be removed in the next major version
Server running on port 5000
```

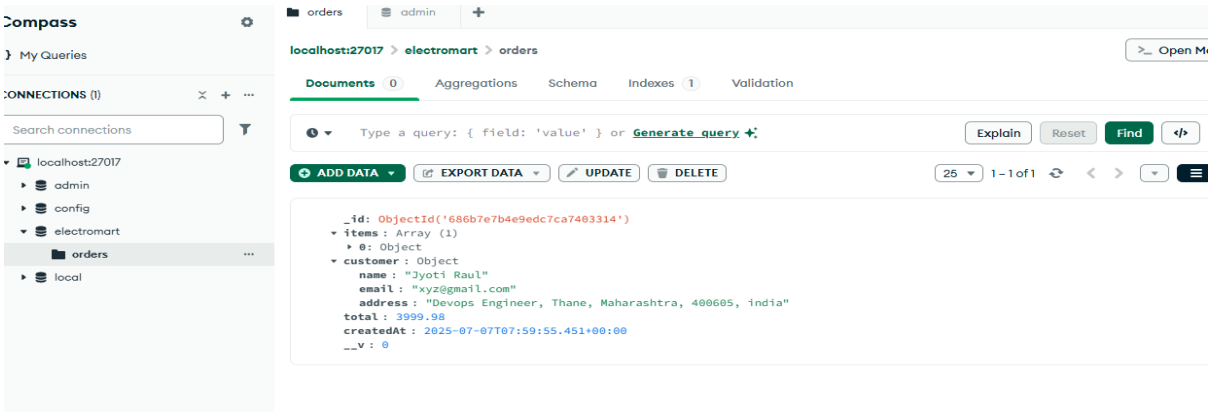
Running frontend application on Local machine

```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-> cd .\frontend\
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\frontend> npm install
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if
 want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm WARN deprecated @babel/plugin-proposal-numeric-separator@7.18.6: This proposal has been merged to the ECMAScript stand
 and thus this plugin is no longer maintained. Please use @babel/plugin-transform-numeric-separator instead.
npm WARN deprecated @babel/plugin-proposal-private-methods@7.18.6: This proposal has been merged to the ECMAScript standar
nd thus this plugin is no longer maintained. Please use @babel/plugin-transform-private-methods instead.
npm WARN deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMASc
t standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator in
ad.
npm WARN deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standa
and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm WARN deprecated @humanwhocodes/config-array@0.13.0: Use @eslint/config-array instead
npm WARN deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecate
See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array
rt#browser_compatibility
npm WARN deprecated @babel/plugin-proposal-private-property-in-object@7.21.11: This proposal has been merged to the ECMASc
```

```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\frontend> npm start

> electromart-app@0.1.0 start
> react-scripts start

(node:18688) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddlewa
ated. Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:18688) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddle
ecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view electromart-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.56.1:3000

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

webpack compiled successfully
Compiling...
Compiled successfully!
```

We can see application on port localhost:3000





Docker-

Created Dockerfile for frontend and Backend. Also created docker-compose file for local orchestration.

Dockerfile for frontend is located in frontend/Dockerfile

Dockerfile for backend is located in backend/Dockerfile

Docker-compose file is located in root path

Command- docker-compose up –build

Docker compose up or docker composer up -d

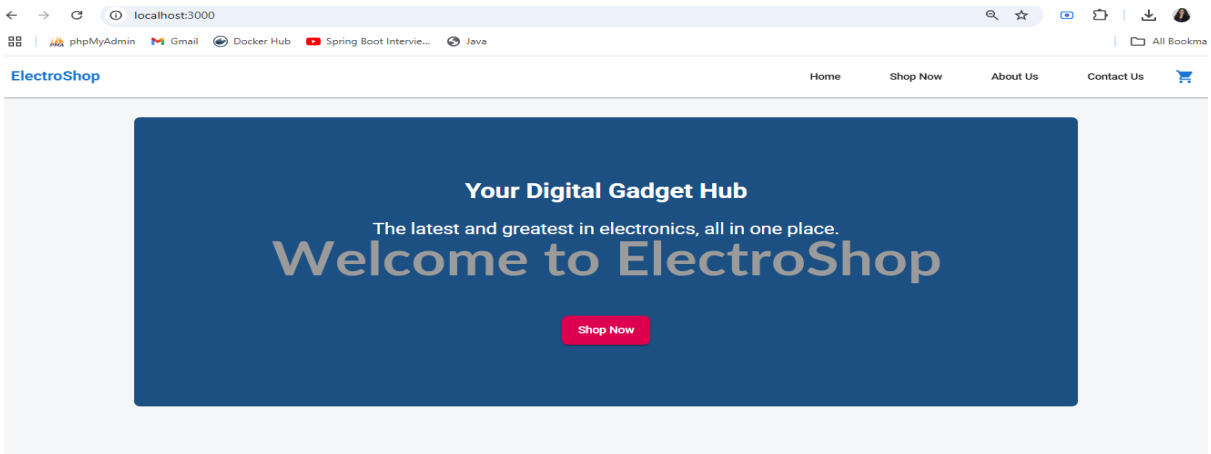
Open http://localhost:3000

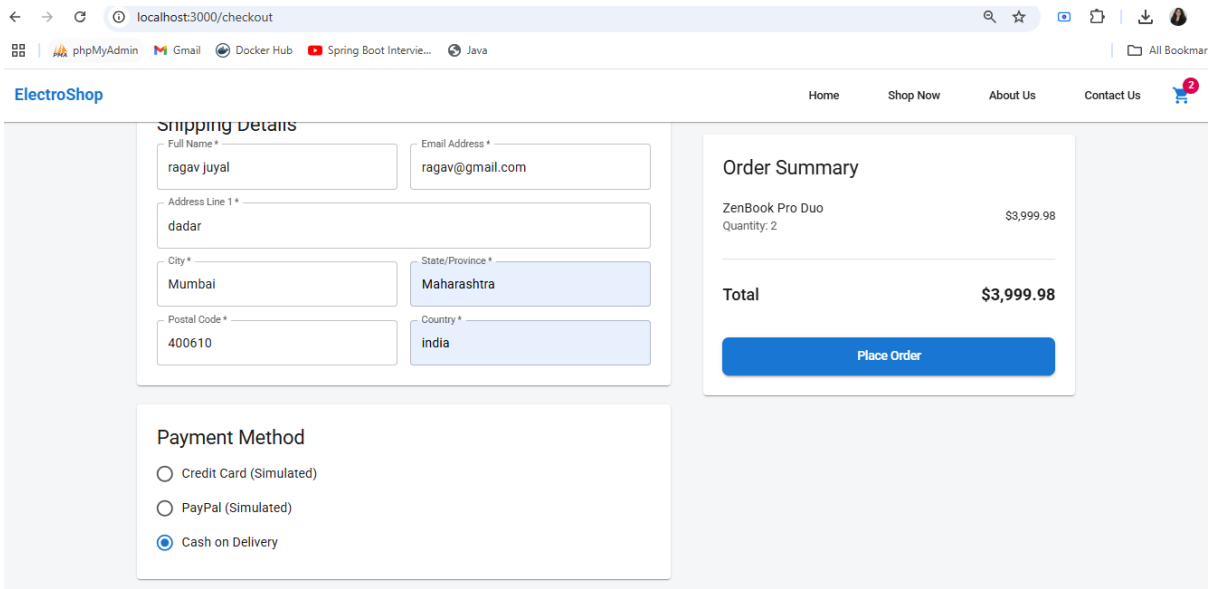
```
PS C:\assignment\Sparknet-Innovation\ElectroShop\ElectroShop-> docker-compose up --build
[+] Running 0/1
[+] Running 9/9g
 ✓ mongo 8 layers [⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled
   ✓ e735f3a6b701 Pull complete
   ✓ a764c7a5cbf4 Pull complete
   ✓ 9aa3a52c924b Pull complete
   ✓ 65e7dee31a65 Pull complete
   ✓ d94512df8710 Pull complete
   ✓ 8974b590250f Pull complete
   ✓ 0272ef2abd88 Pull complete
   ✓ 6440c965ca82 Pull complete
[+] Building 1.9s (2/3)
[+] Building 2.2s (2/3)
[+] Building 2.3s (2/3)
[+] Building 9.6s (13/15)
[+] Building 10.5s (14/15)
[+] Building 10.8s (14/15)
[+] Building 11.2s (14/15)
[+] Building 75.3s (24/24) FINISHED
 => [backend internal] load .dockerignore
 => => transferring context: 2B
 => [backend internal] load build definition from Dockerfile
 => => transferring dockerfile: 334B
 => [frontend internal] load metadata for docker.io/library/node:18-alpine
 => [backend auth] library/node:pull token for registry-1.docker.io
 => [frontend builder 1/6] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a
 => [backend internal] load build context
 => => transferring context: 133.25kB
 => CACHED [frontend builder 2/6] WORKDIR /app
 => CACHED [backend builder 3/5] COPY package*.json ./
```

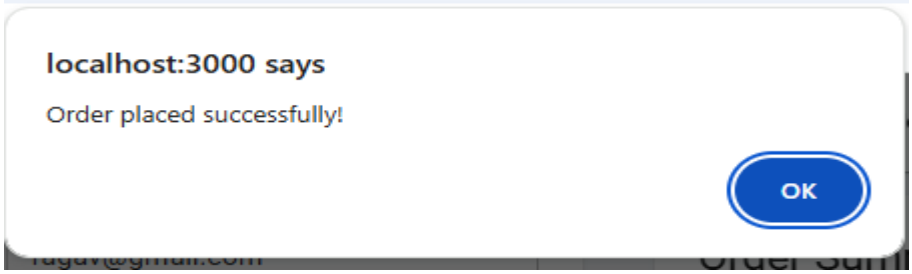
```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-> docker-compose  up
[+] Building 0.0s (0/0)
[+] Running 4/4
 ✓ Network electroshop-_electroshop-net  Created
 ✓ Container electroshop-mongo           Created
 ✓ Container electroshop--backend-1      Created
 ✓ Container electroshop--frontend-1     Created
Attaching to electroshop--backend-1, electroshop--frontend-1, electroshop-mongo
electroshop--frontend-1  | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform con
electroshop--frontend-1  | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
electroshop--frontend-1  | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
electroshop--frontend-1  | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/defau
electroshop--frontend-1  | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/def
electroshop--frontend-1  | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
electroshop--frontend-1  | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
electroshop--frontend-1  | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
```

Validation-

Application successfully running on http://localhost:3000







Check database—

```
PS C:\assignment\Sparknet-Innovation\ElectroShop\ElectroShop-> docker exec -it electroshop-mongo mongosh
Current Mongosh Log ID: 686ba094c343eb3912baa8b8
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.3
Using MongoDB:          6.0.24
Using Mongosh:          2.5.3

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/


To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting
   2025-07-07T10:22:45.279+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/co
notes-filesystem
   2025-07-07T10:22:49.880+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2025-07-07T10:22:49.880+00:00: vm.max_map_count is too low
```

```
test> show dbs
admin        40.00 KiB
config       60.00 KiB
electroshop  72.00 KiB
local        72.00 KiB
test> use electroshop
switched to db electroshop
electroshop> show collections
orders
electroshop> db.orders.find().pretty()
[
  {
    _id: ObjectId('686b9ef1a0151819bc3e8d8b'),
    items: [
      {
        name: 'ZenBook Pro Duo',
        price: 1999.99,
        quantity: 1,
        _id: ObjectId('686b9ef1a0151819bc3e8d8c')
      }
    ],
    customer: {
      name: 'abc',
      email: 'xyz@gmail.com',
      address: 'thane, Thane, Maharashtra, 400605, india'
    },
    total: 1999.99,
```

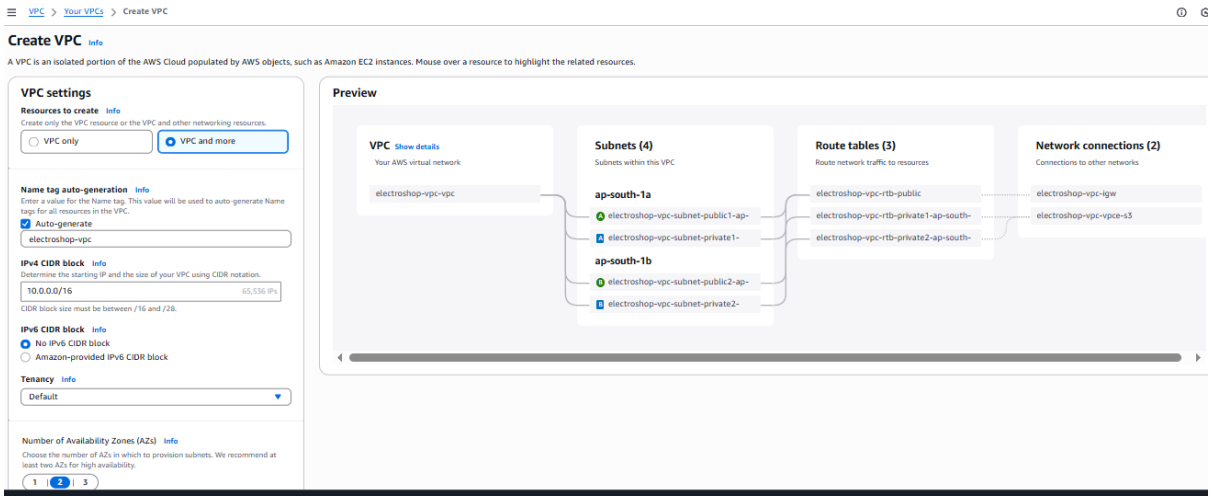
```
    ],
    customer: {
      name: 'abc',
      email: 'xyz@gmail.com',
      address: 'thane, Thane, Maharashtra, 400605, india'
    },
    total: 1999.99,
    createdAt: ISODate('2025-07-07T10:18:25.461Z'),
    __v: 0
  },
  {
    _id: ObjectId('686ba06b8344c8b4156da470'),
    items: [
      {
        name: 'ZenBook Pro Duo',
        price: 1999.99,
        quantity: 2,
        _id: ObjectId('686ba06b8344c8b4156da471')
      }
    ],
    customer: {
      name: 'ragav juyal',
      email: 'ragav@gmail.com',
      address: 'dadar, Mumbai, Maharashtra, 400610, india'
    },
    total: 3999.98,
    createdAt: ISODate('2025-07-07T10:24:43.222Z'),
    __v: 0
  }
]
electroshop>
```

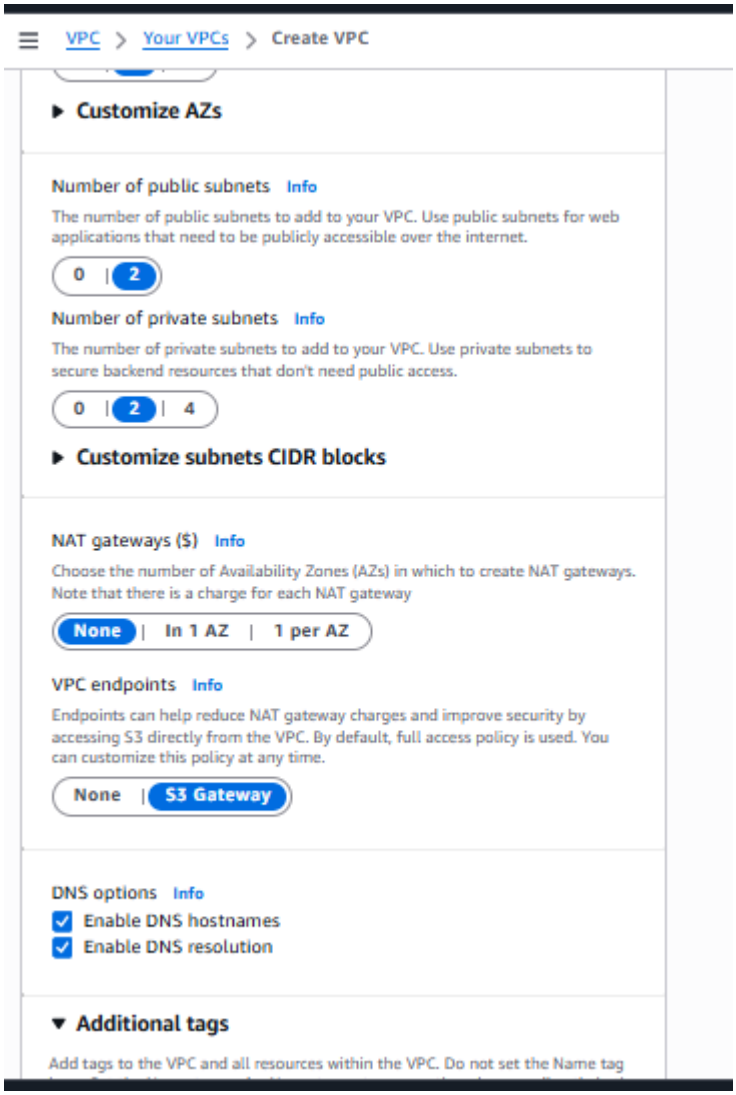
2. Aws infrastructure and provisioning

This phase involved the design and provisioning of the cloud infrastructure using AWS:
- Virtual Private Cloud (VPC) creation with multiple subnets (public/private).
- Setup of NAT Gateway and Elastic IP for private subnet access.
- Configuration of IAM Roles and Policies for ECS access.
- Creating Elastic Container Registry (ECR) repositories for storing Docker images.
- Security groups for ALB and ECS tasks to ensure appropriate access rules.

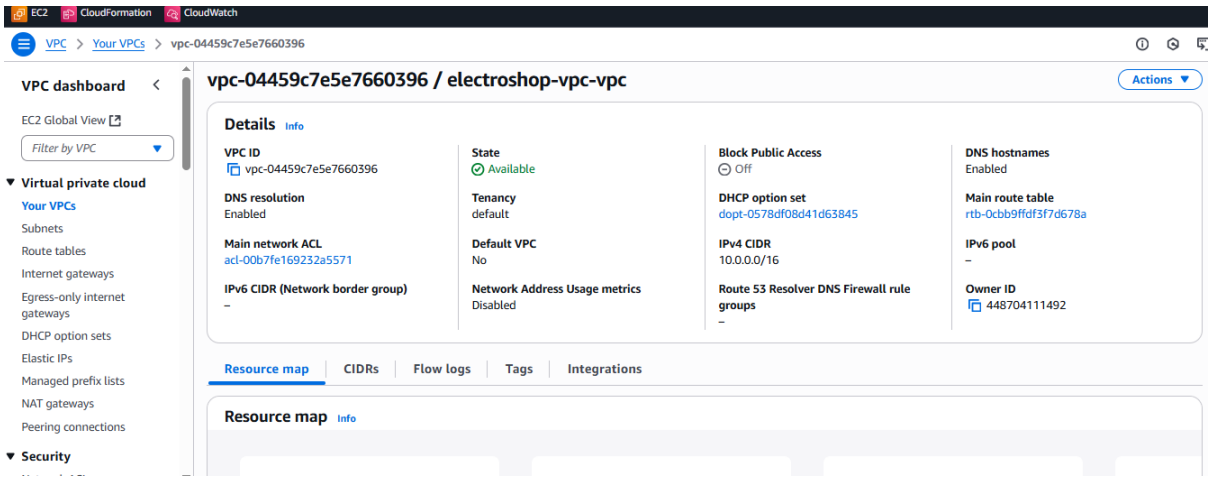
1. Virtual private network setup -

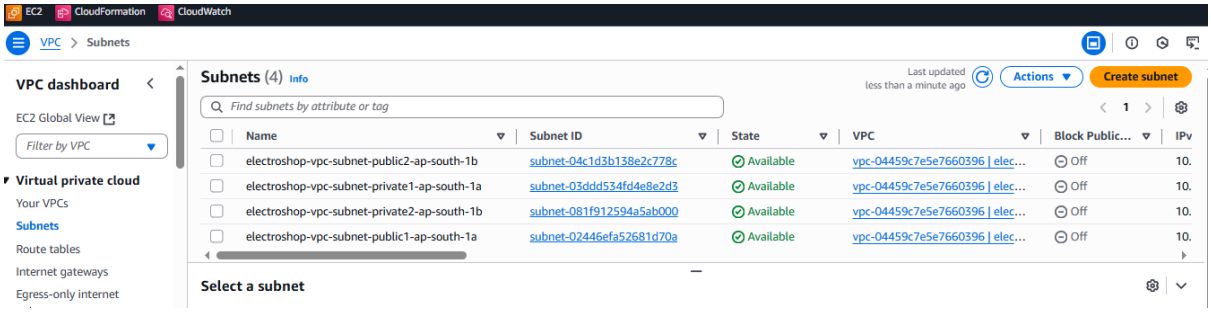
Go to vpc-> create vpc-> click on VPC and More -> tag - electroshop-vpc -> availability zone 2-> public subnet 2-> private subnet 2 (default configuration)

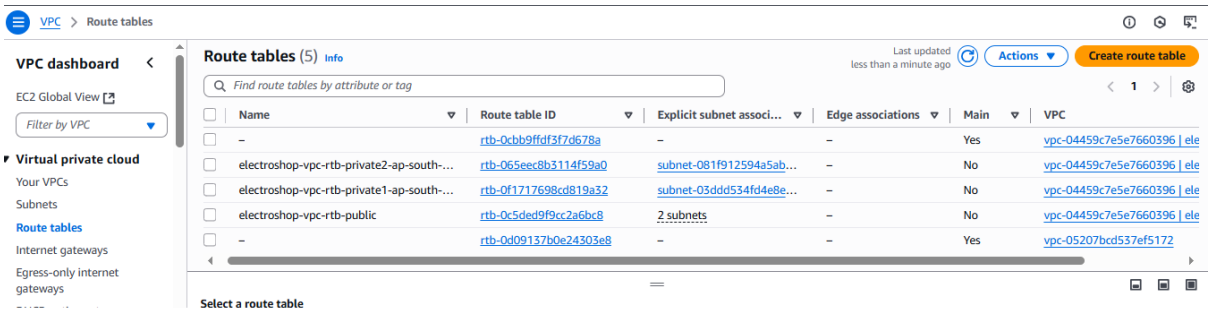




VPC look like→





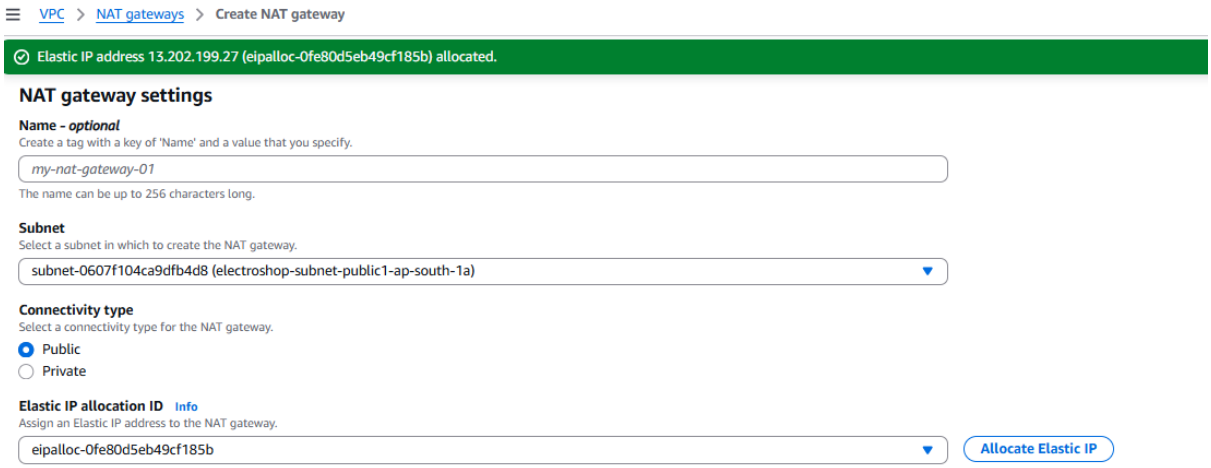


2. Creating nat gateway—

Name tag: electroshop-nat-gw

Subnet: Select **one of your public subnets.**

Elastic IP: Click **"Allocate Elastic IP"**, then **"Create a new one"** and select it.



Go Back to Route Table and Add NAT Route

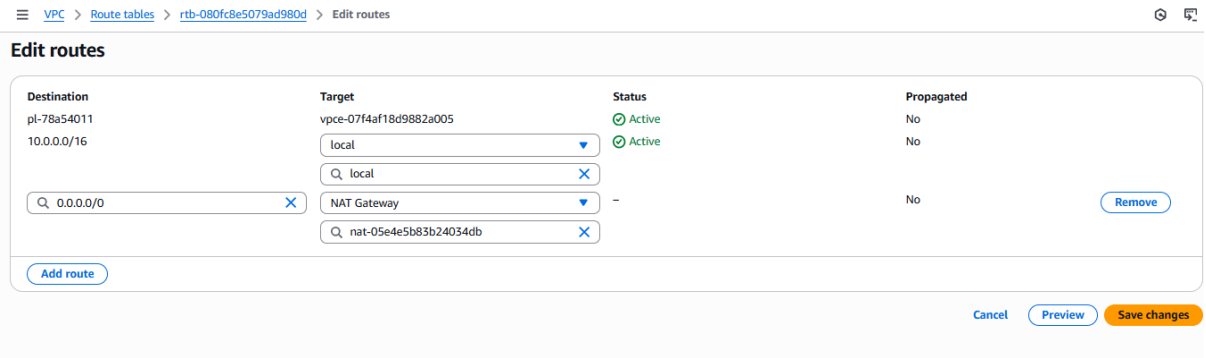
Go back to your private route table (e.g., electroshop-rtb-private1-ap-south-1a)

Edit Routes-

1. For destination 0.0.0.0/0, the **NAT Gateway should now appear in the dropdown**

2. Select it and **save changes**

3. Repeat for the second private route table

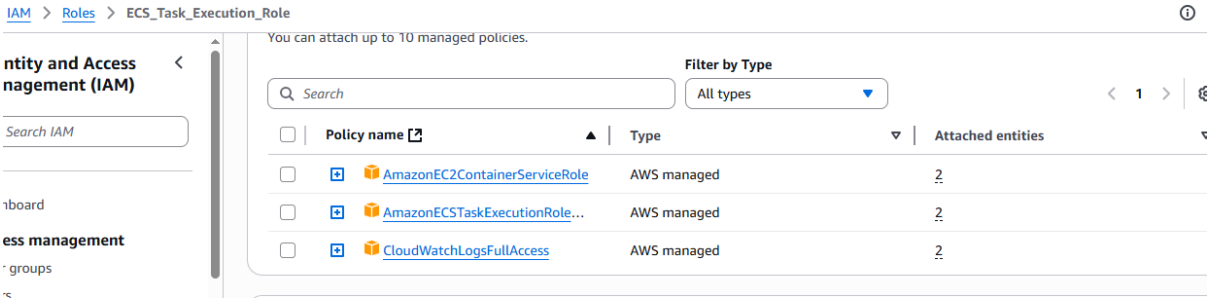


3. Create IAM Roles and attached policy:

IAM > Roles > Create Role -> AWS Service → Use Case: Elastic Container Service ->

Next-> Role name: ECS_Task_Execution_Role -> create role

Add permission/ policy -> AmazonECSTaskExecutionRolePolicy , CloudWatchLogsFullAccess



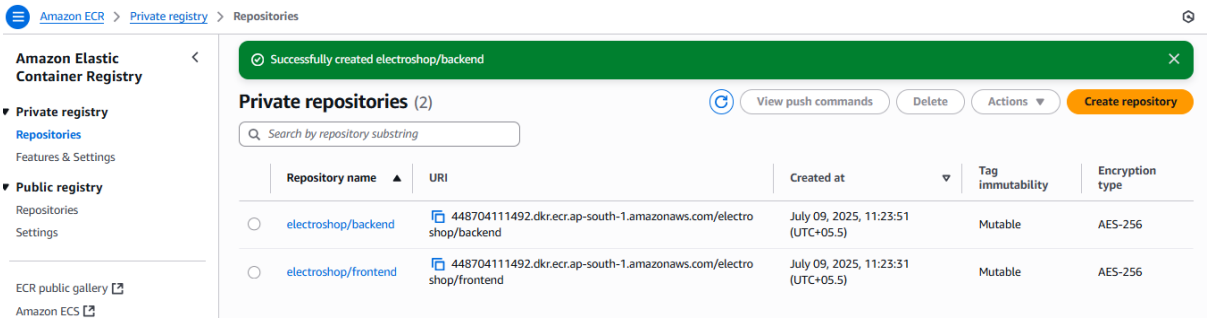
4. Create Elastic Container Registry-

Go to Amazon ECR > Create Repository

Create:

- electroshop/frontend
- electroshop/backend

Set them to Private Repositories

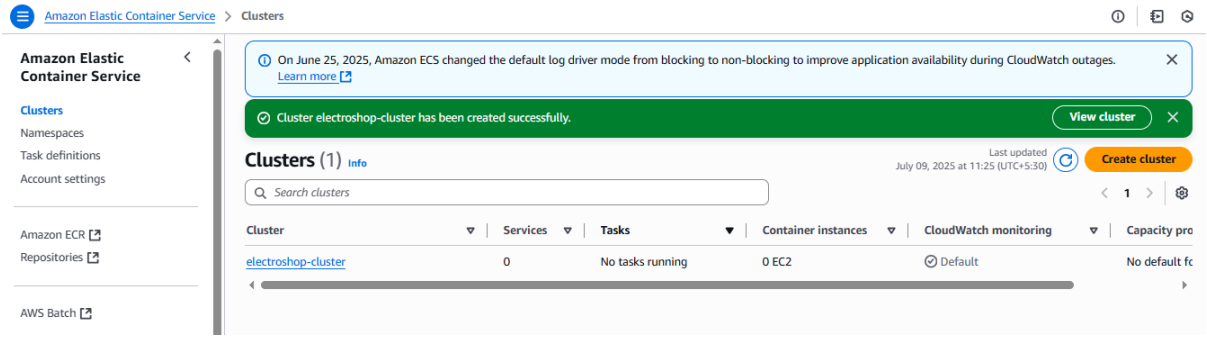


5. Create ECS Cluster

Go to **ECS > Clusters > Create Cluster**

Launch type: **Fargate**

Cluster name: electroshop-cluster



6. create security group

Go to AWS Console → Search for "EC2" → Open EC2 Dashboard

In the sidebar, go to "Security Groups" under the "Network & Security" section

Click "Create security group"

Security Group 1: electroshop-alb-sg (For Load Balancer)

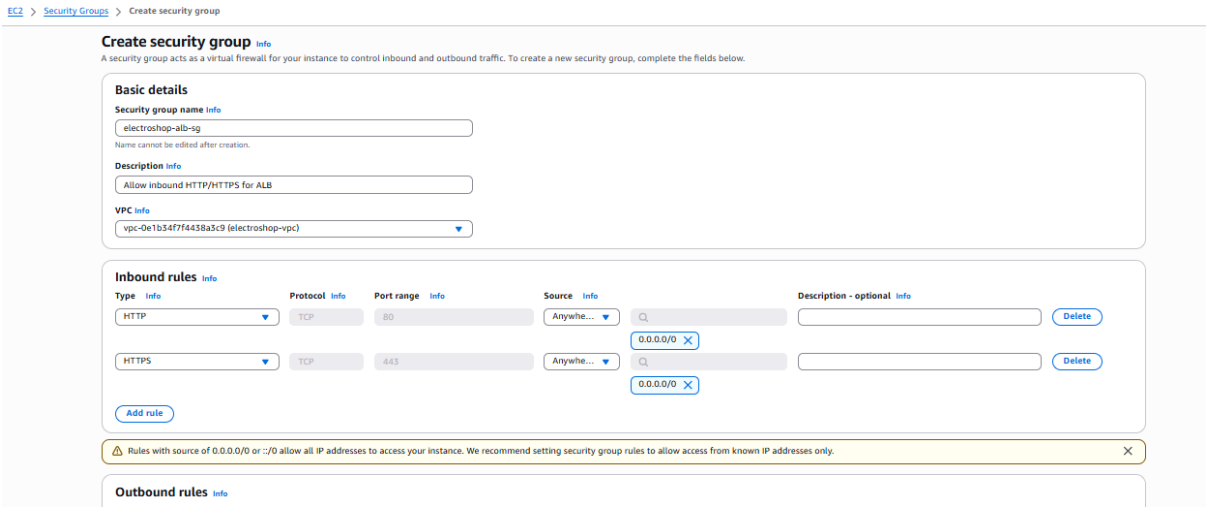
Security group name: electroshop-alb-sg

Description: Allow inbound HTTP/HTTPS for ALB

VPC: Select your electroshop-vpc

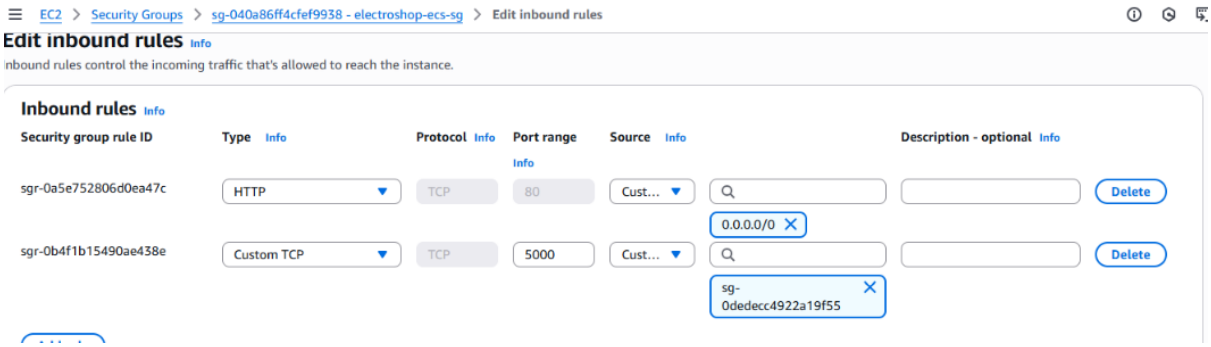
Type	Protocol	Port Range	Source
HTTP	TCP	80	Anywhere (0.0.0.0/0)
HTTPS	TCP	443	Anywhere (0.0.0.0/0)

Outbound Rules: Default is **All traffic (Anywhere)** — keep as is. Click **Create security group**



Create a second one:

- **Security group name**: electroshop-ecs-sg
- **Description**: Allow traffic from ALB to ECS tasks
- **VPC**: Same as above (electroshop-vpc)



3- Manual Deployment to AWS-

Manual deployment included building Docker images, pushing them to ECR, and configuring ECS Fargate services:
- Backend and frontend images were built, tagged, and pushed to ECR.
- MongoDB cluster was set up using MongoDB Atlas with a secure connection string.
- ECS task definitions and services were created with ALB for routing.

1. aws configure

```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-> aws configure
AWS Access Key ID [****************R25G]:
AWS Secret Access Key [****************WCE6]:
Default region name [ap-south-1]:
Default output format [None]:
```

2. Push container image

aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin <your-account-id>.dkr.ecr.ap-south-1.amazonaws.com

Replace <your-account-id> with your actual AWS account ID.

```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-> aws ecr get-login-password --region ap-south-1 | dock
er login --username AWS --password-stdin 448704111492.dkr.ecr.ap-south-1.amazonaws.com
[2025-07-09T06:15:47.726495300Z][docker-credential-desktop][W] Windows version might not be up-to-date: The system cannot fin
d the file specified.
Login Succeeded
[2025-07-09T06:15:50.648471900Z][docker-credential-desktop][W] Windows version might not be up-to-date: The system cannot fin
d the file specified.

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-t
okens/
```

Tag and Push Docker Images

Before build and push image to ecr just change below line-

```
CheckoutPage.jsx M ×
frontend > src > pages > CheckoutPage.jsx > CheckoutPage > handlePlaceOrder > res
8   const CheckoutPage = () => {
30      const handlePlaceOrder = async (e) => {
32          const orderData = {
34              customer: {
                    address: ${shipping.address}, ${shipping.city}, ${shipping.state}, ${shipping.postalCode}, ${shipping.coun
38              },
39              total
40          };
41          try {
42              const res = await fetch('http://electroshop-alb-108564332.ap-south-1.elb.amazonaws.com/api/orders', {
43                  method: 'POST',
44                  headers: { 'Content-Type': 'application/json' },
45                  body: JSON.stringify(orderData)
46              });
47              if (res.ok) {
48                  // Optionally clear cart here
49                  // Show confirmation and redirect
50                  alert('Order placed successfully!');
51                  navigate('/thank-you');
52              } else {
53                  alert('Failed to place order.');
```

Frontend :

npm run build

ls build/index.html

docker build -t electroshop-frontend .

docker tag electroshop-frontend:latest 448704111492.dkr.ecr.ap-south-1.amazonaws.com/electroshop/frontend

docker push 448704111492.dkr.ecr.ap-south-1.amazonaws.com/electroshop/frontend

```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\frontend> docker build -t electroshop-frontend .
[+] Building 1.2s (2/4)
 => [internal] load build definition from Dockerfile                                                    0.4s
[+] Building 1.5s (2/4)
 => [internal] load build definition from Dockerfile                                                    0.4s
[+] Building 1.6s (2/4)
 => [internal] load build definition from Dockerfile                                                    0.4s
[+] Building 1.9s (2/4)
 => [internal] load build definition from Dockerfile                                                    0.4sn
[+] Building 2.1s (2/4)
 => [internal] load build definition from Dockerfile                                                    0.4sn
[+] Building 2.2s (2/4)
 => [internal] load build definition from Dockerfile                                                    0.4sn
[+] Building 313.1s (16/16) FINISHED
```

```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\frontend> docker tag electroshop-frontend:latest 44870
4111492.dkr.ecr.ap-south-1.amazonaws.com/electroshop/frontend
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\frontend> docker push 448704111492.dkr.ecr.ap-south-1.
amazonaws.com/electroshop/frontend
Using default tag: latest
[2025-07-09T11:41:48.000749100Z][docker-credential-desktop][W] Windows version might not be up-to-date: The system cannot fin
d the file specified.
The push refers to repository [448704111492.dkr.ecr.ap-south-1.amazonaws.com/electroshop/frontend]
fae46558d396: Layer already exists
7e881413b635: Layer already exists
3db4ef8dec24: Layer already exists
2bfd71b953a3: Layer already exists
b0debeaa68c5: Layer already exists
5f5a3d906b11: Layer already exists
9c2b6e6f2e2e: Layer already exists
4babb02c7c40: Layer already exists
08000c18d16d: Layer already exists
latest: digest: sha256:cefaed41e950f1d91fef1759bf93a65e40670700cb859fb000e65469364f1624 size: 2199
```

Backend :

docker build -t electroshop-backend .

docker tag electroshop-backend:latest 448704111492.dkr.ecr.ap-south-1.amazonaws.com/electroshop/backend

docker push 448704111492.dkr.ecr.ap-south-1.amazonaws.com/electroshop/backend

```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\backend> docker build -t electroshop-backend .
[+] Building 0.4s (2/2)
 => [internal] load build definition from Dockerfile                                                                  0.2s
[+] Building 0.5s (2/3)
 => [internal] load build definition from Dockerfile                                                                  0.2s
[+] Building 0.8s (2/3)
 => [internal] load build definition from Dockerfile                                                                  0.2sn
[+] Building 8.0s (12/12) FINISHED
```

```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\backend> docker tag electroshop-backend:latest 4487041
11492.dkr.ecr.ap-south-1.amazonaws.com/electroshop/backend
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\backend> docker push 448704111492.dkr.ecr.ap-south-1.a
mazonaws.com/electroshop/backend
Using default tag: latest
[2025-07-09T11:43:11.526096400Z][docker-credential-desktop][W] Windows version might not be up-to-date: The system cannot fin
d the file specified.
The push refers to repository [448704111492.dkr.ecr.ap-south-1.amazonaws.com/electroshop/backend]
8d4abdb1eeae: Layer already exists
9cefd24d97b6: Layer already exists
82140d9a70a7: Layer already exists
f3b40b0cdb1c: Layer already exists
0b1f26057bd0: Layer already exists
08000c18d16d: Layer already exists
latest: digest: sha256:db4af408b9806dcefa9d9d81bbd5f93d376746bb50789d2106f7c1656537d6d2 size: 1575
```

3. Connection for database:

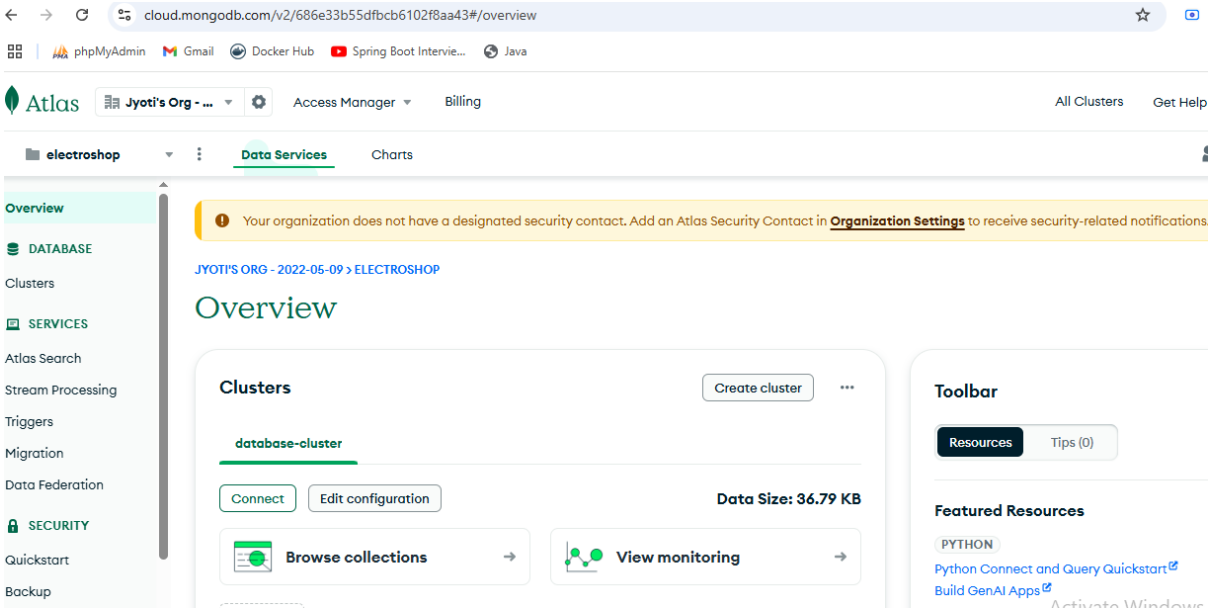
Login to https://cloud.mongodb.com/

Create new project-> your project name -> next -> create project

Create cluster -> select free tier -> clustername-> provider -aws -> create deployment.

Create user -> done ->

Click on cluster -> select details which you have need -> copy url



4. Create ECS Task Definitions

ECS > Task Definitions > Create new > Fargate

Create new task definition
a) electroshop-backend-task :

AWS Fargate

Container – backend

Image uri-- 448704111492.dkr.ecr.ap-south-1.amazonaws.com/electroshop/backend

Port 5000 HTTP

Environment variable –

MONGO_URI

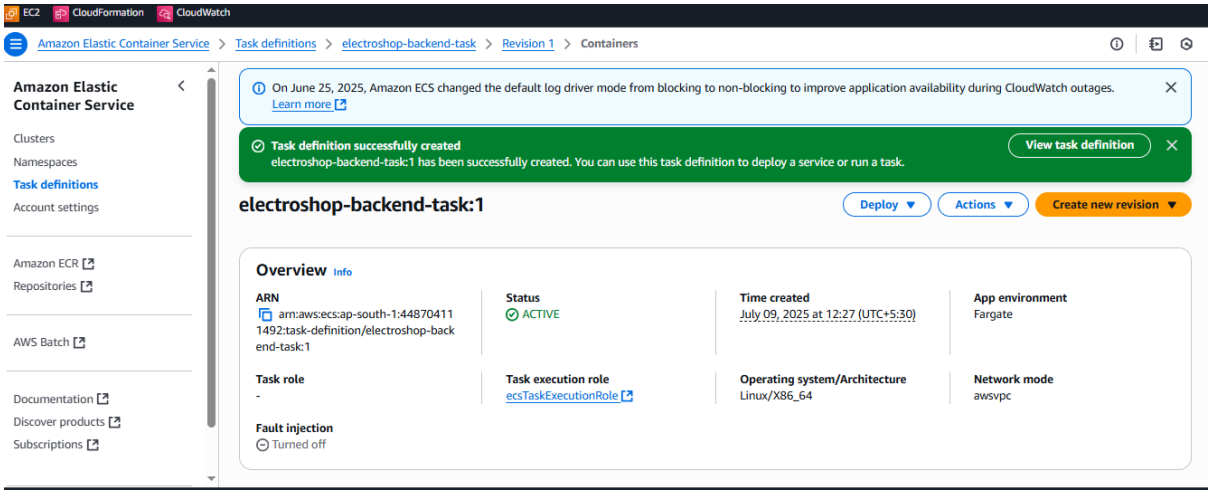
Special characters (like @) must be URL-encoded:

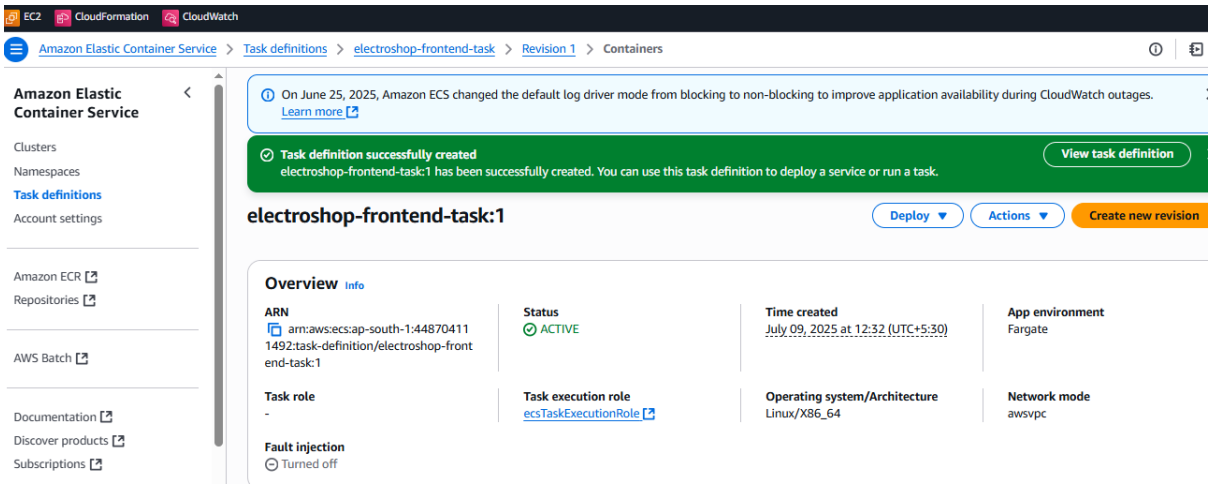
- @ → %40

Name	Value
MONGO_URI	mongodb+srv://<your-user>:<your-pass>@cluster0.mongodb.net/electroshop?retryWrites=true&w=majority

Use log collection –

mongodb+srv://Jyotiraul74:Jyoti%401994@database-cluster.ixtjrot.mongodb.net/electroshop?ssl=true&authSource=admin&retryWrites=true&w=majority



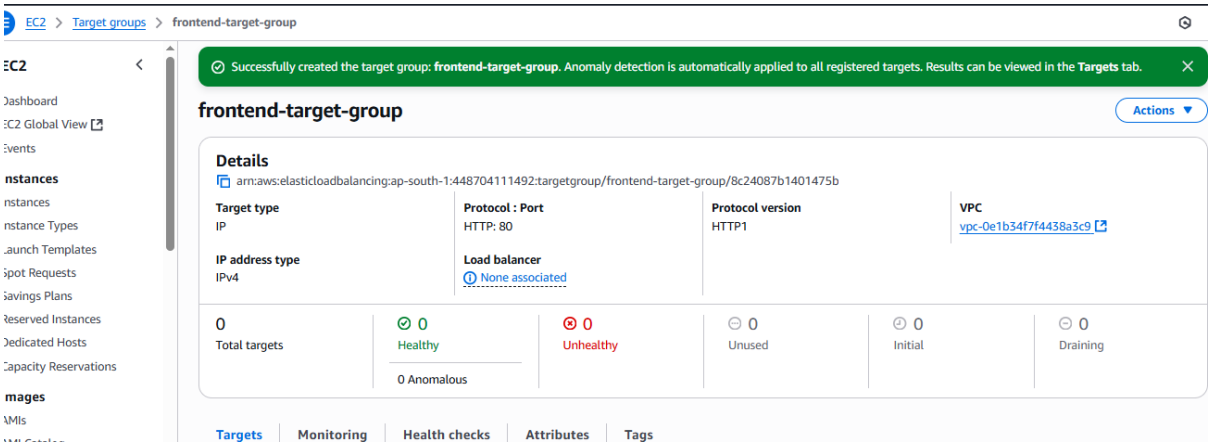


5. Create Application Load Balancer (ALB)

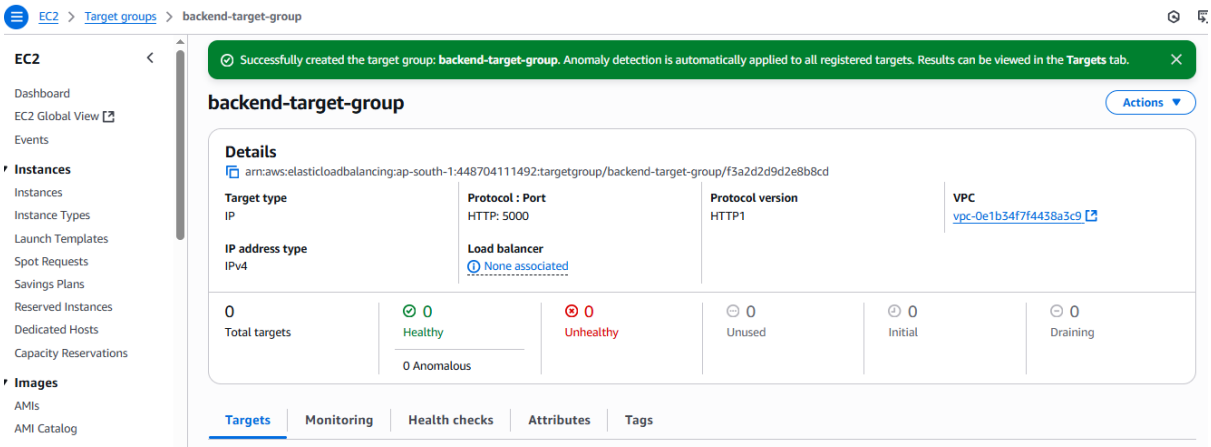
EC2 > Load Balancers > Create- Application Load Balancer-

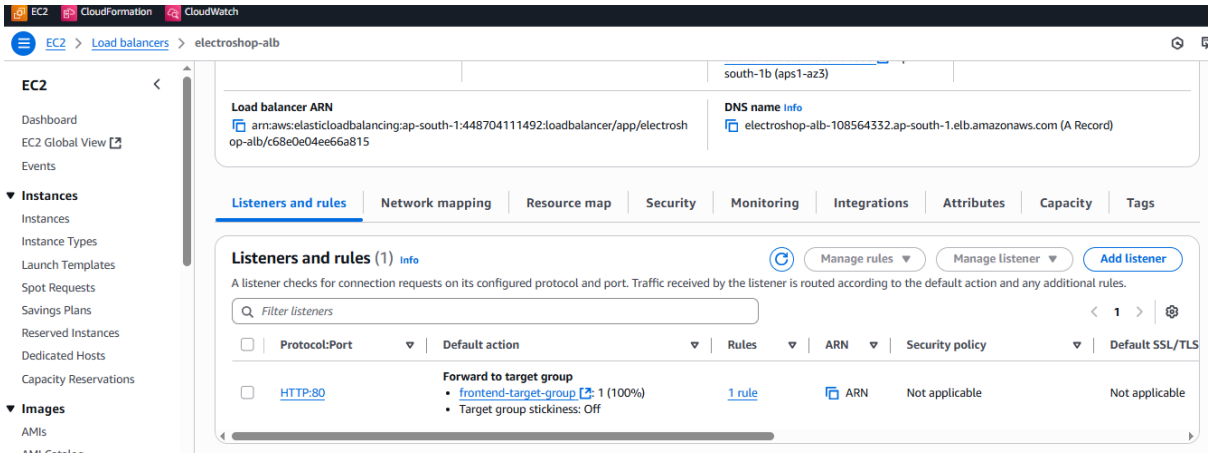
Loadbalancer name : electroshop-alb, select internet facing, ipv4, electroshop vpc, **Availability Zones and subnets (choose 2) , security group- electroshop-alb-sg, port 80, Default Rule → forward to frontend-target-group**

+create target group 1 → ip address, frontend-target-group, http 80, electroshop vpc ,



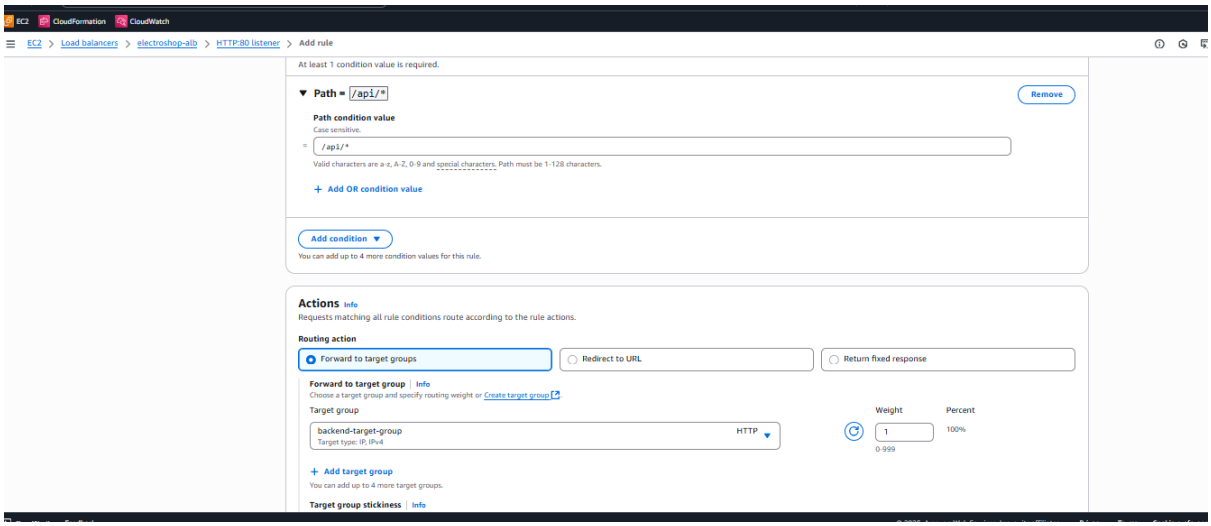
+create target group 2→ ip address, backend-target-group, http 5000, electroshop vpc ,

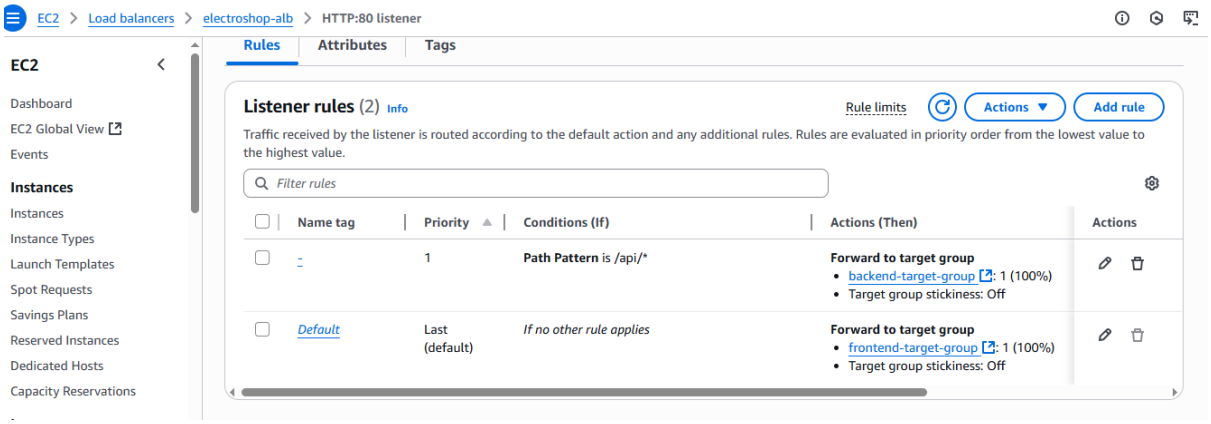




Add the /api/* Rule for Backend Routing-

- **View/edit rules"** or a section called **"Rules"**
- Click on **"View/edit rules"** for **Listener: HTTP:80**





6. Deploy ECS Services

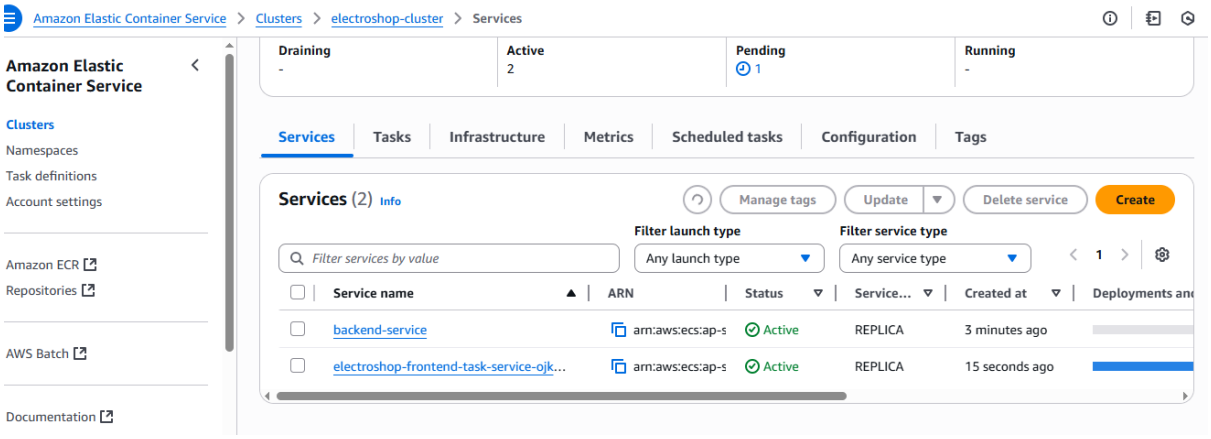
Go to: **ECS > Clusters > Your Cluster > Create Service**

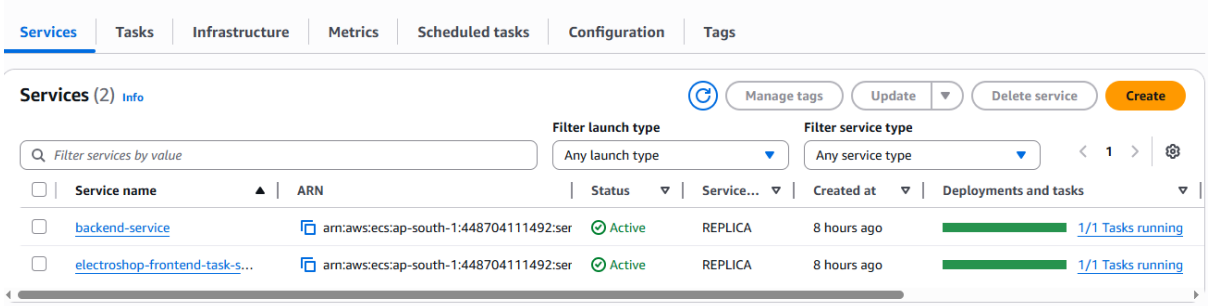
Create **two services**: one for frontend, one for backend

Service Setup:

- Launch Type: Fargate
- Task Definition: select respective one
- Cluster: your ECS cluster
- Service name: e.g., frontend-service
- Number of tasks: 1 or 2
- Load Balancer: Yes
- ALB: select your ALB
- Listener: HTTP 80
- Target Group: corresponding one (frontend or backend)

Also choose the correct subnets (private) and attach the electroshop-ecs-sg





Note: for debugging and if application failed, follow below step for deployment.

Go to ECS > Task Definitions (Revision)

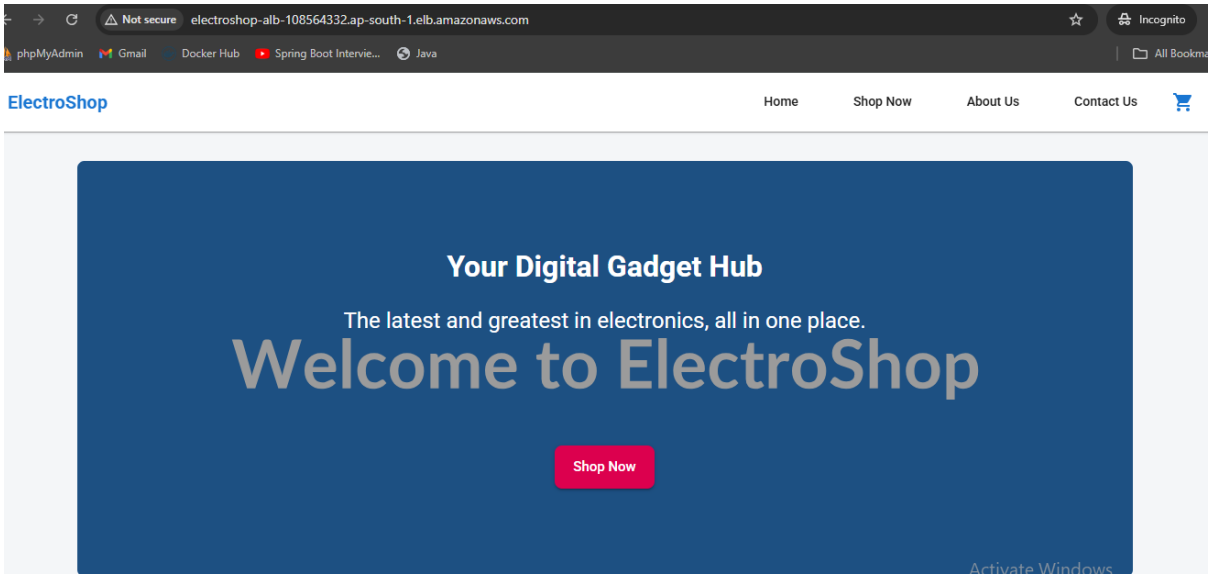
1. Click electroshop-backend-task / electroshop-backend-task
2. Click Create new revision
3. Do changes if needed
4. Save the revision.

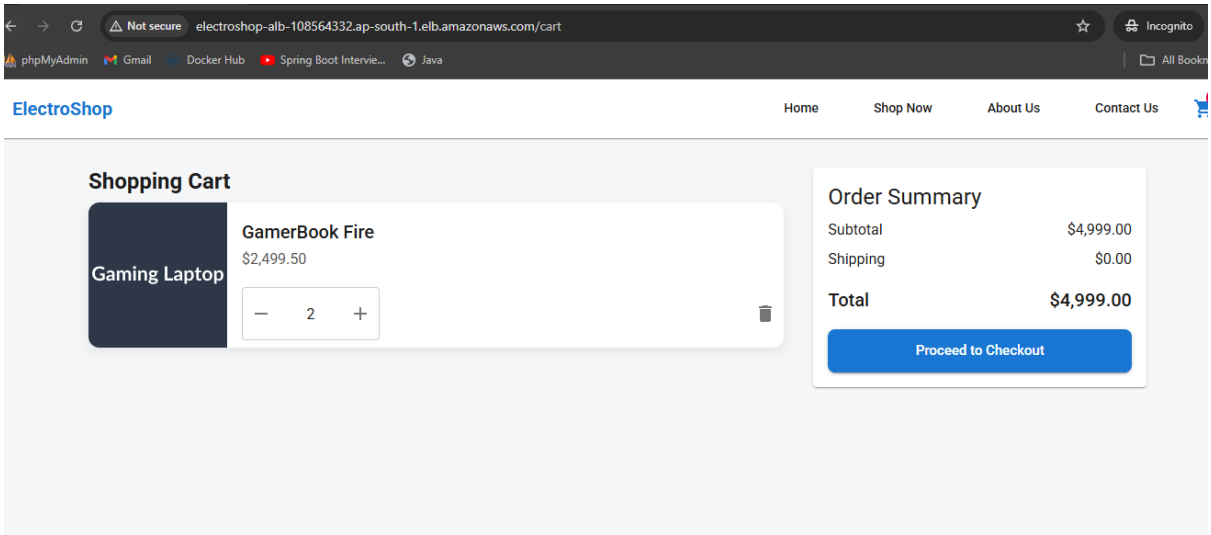
Redeploy Backend Service if need-

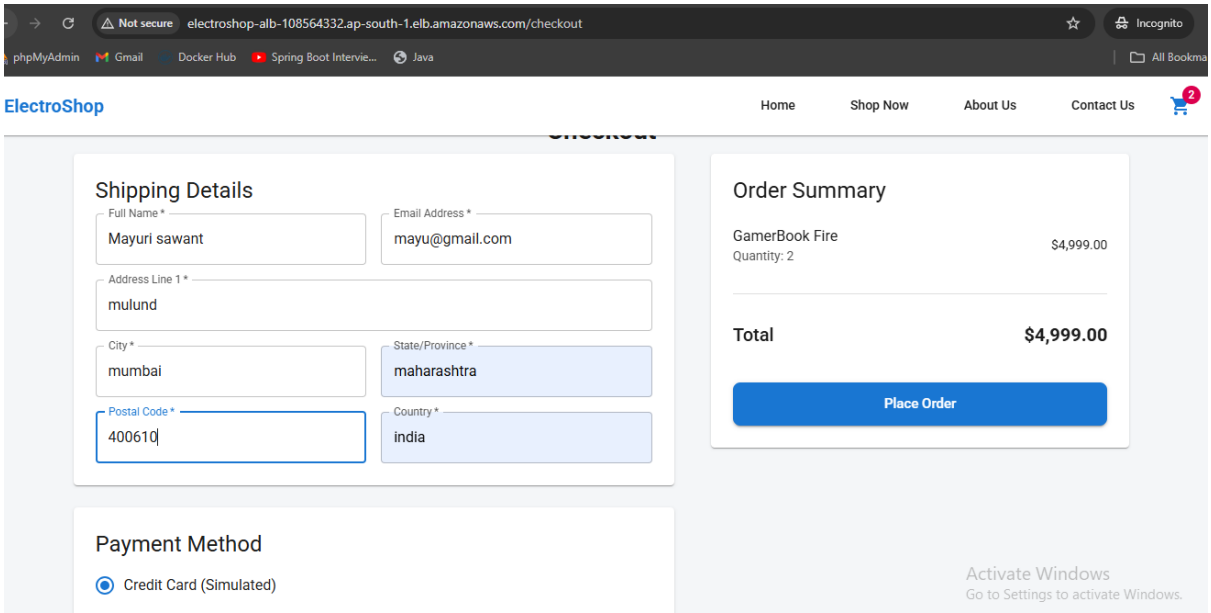
1. Go to ECS > Clusters > electroshop-cluster > Services
2. Click backend-service/ frontend service
3. Click Update
4. Choose the new task definition revision
5. Leave all other settings as-is → Update service

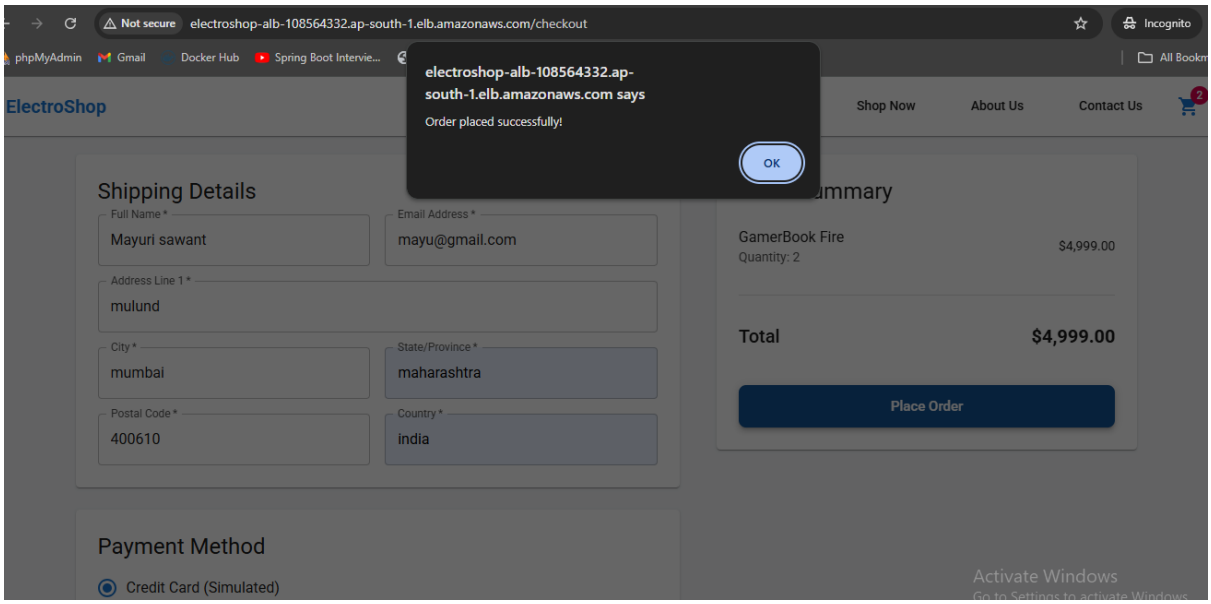
6. **Validate**

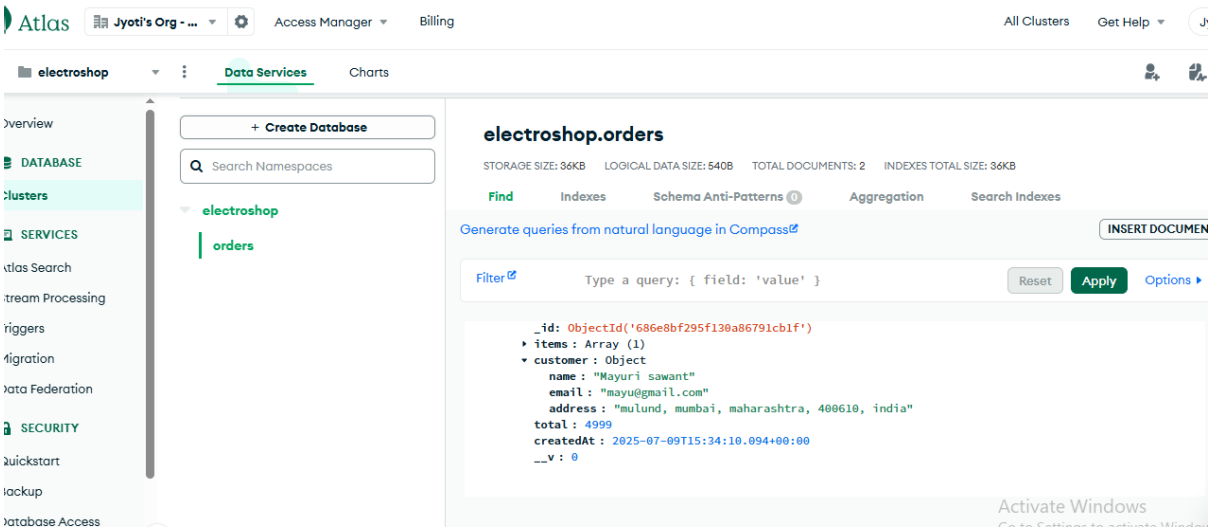
http://electroshop-alb-108564332.ap-south-1.elb.amazonaws.com/


Not secure
electroshop-alb-108564332.ap-south-1.elb.amazonaws.com
Incognito
phpMyAdmin
Gmail
Docker Hub
Spring Boot Intervie...
Java
ElectroShop
Home
Shop Now
About Us
Contact Us
Your Digital Gadget Hub
The latest and greatest in electronics, all in one place.
Welcome to ElectroShop
Shop Now


Not secure
electroshop-alb-108564332.ap-south-1.elb.amazonaws.com/cart
Incognito
phpMyAdmin
Gmail
Docker Hub
Spring Boot Intervie...
Java
ElectroShop
Home
Shop Now
About Us
Contact Us
Shopping Cart
Gaming Laptop
GamerBook Fire
$2,499.50
2
Order Summary
Subtotal
$4,999.00
Shipping
$0.00
Total
$4,999.00
Proceed to Checkout


Not secure
electroshop-alb-108564332.ap-south-1.elb.amazonaws.com/checkout
Incognito
phpMyAdmin
Gmail
Docker Hub
Spring Boot Intervie...
Java
ElectroShop
Home
Shop Now
About Us
Contact Us
2
Shipping Details
Full Name *
Mayuri sawant
Email Address *
mayu@gmail.com
Address Line 1 *
mulund
City *
mumbai
State/Province *
maharashtra
Postal Code *
400610
Country *
india
Order Summary
GamerBook Fire
Quantity: 2
$4,999.00
Total
$4,999.00
Place Order
Payment Method
Credit Card (Simulated)
Activate Windows
Go to Settings to activate Windows.

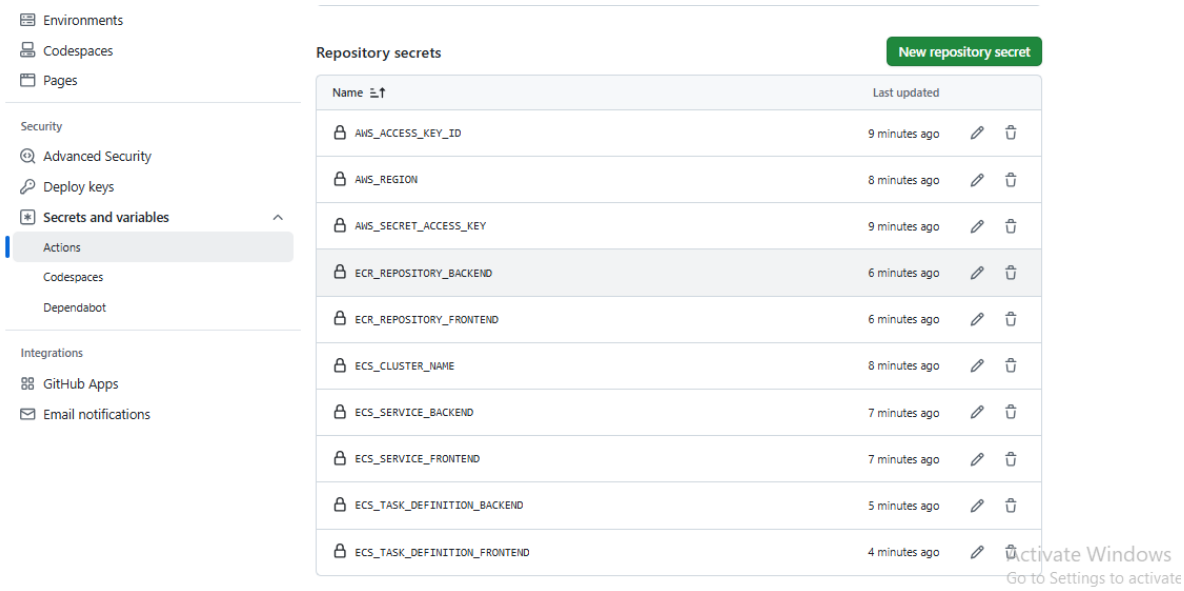

Not secure
electroshop-alb-108564332.ap-south-1.elb.amazonaws.com/checkout
Incognito
phpMyAdmin
Gmail
Docker Hub
Spring Boot Intervie...
All Bookm
electroshop-alb-108564332.ap-south-1.elb.amazonaws.com says
Order placed successfully!
OK
ElectroShop
Shop Now
About Us
Contact Us
Shipping Details
Full Name *
Mayuri sawant
Email Address *
mayu@gmail.com
Address Line 1 *
mulund
City *
mumbai
State/Province *
maharashtra
Postal Code *
400610
Country *
india
ummary
GamerBook Fire
Quantity: 2
$4,999.00
Total
$4,999.00
Place Order
Payment Method
Credit Card (Simulated)
Activate Windows


Atlas
Jyoti's Org - ...
Access Manager
Billing
All Clusters
Get Help
electroshop
Data Services
Charts
Overview
DATABASE
Clusters
SERVICES
Atlas Search
Stream Processing
Triggers
Migration
Data Federation
SECURITY
Quickstart
Backup
Database Access
+ Create Database
Search Namespaces
electroshop
orders
electroshop.orders
STORAGE SIZE: 36KB
LOGICAL DATA SIZE: 540B
TOTAL DOCUMENTS: 2
INDEXES TOTAL SIZE: 36KB
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
Generate queries from natural language in Compass
INSERT DOCUMEN
Filter
Type a query: { field: 'value' }
Reset
Apply
Options
_id: ObjectId('686e8bf295f130a86791cb1f')
items : Array (1)
customer : Object
name : "Mayuri sawant"
email : "mayu@gmail.com"
address : "mulund, mumbai, maharashtra, 400610, india"
total : 4999
createdAt : 2025-07-09T15:34:10.094+00:00
__v : 0
Activate Windows

Phase 2- CI/CD, monitoring & security

4. CI/CD Pipeline automation

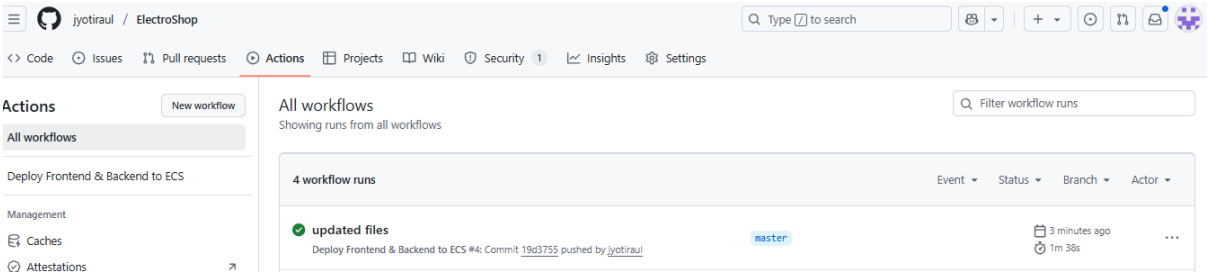
CI/CD pipeline was implemented using GitHub Actions. Any code change (e.g., a new message on the about us page) would automatically reflect on deployment.



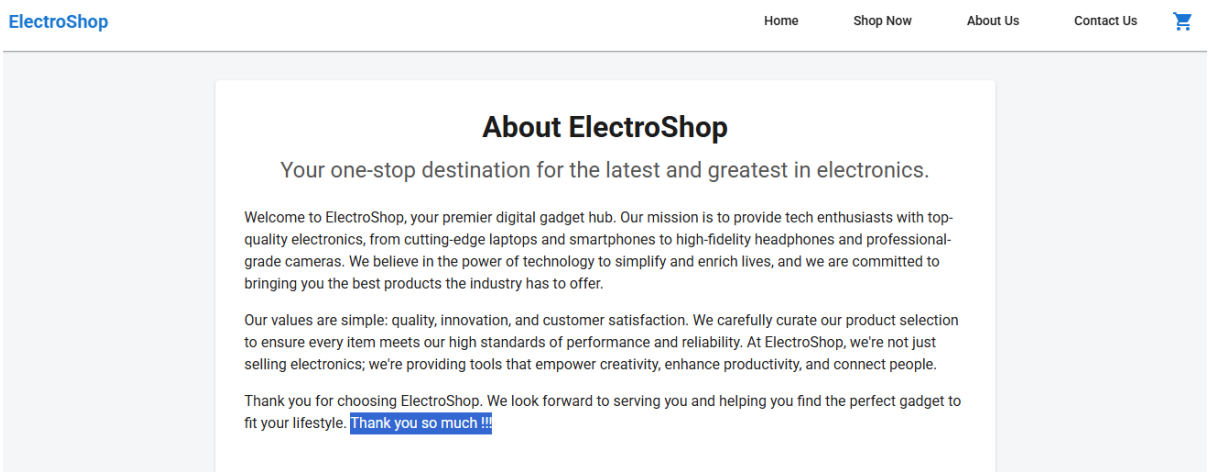
Added Thank you so much !!! in aboutUsPage.jsx

```
<Typography variant="body1" paragraph>
  Thank you for choosing ElectroShop. We l
  Thank you so much !!!
</Typography>
```

```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-> git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   frontend/src/pages/AboutUsPage.jsx

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-> git add frontend/src/pages/AboutUsPage.jsx
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-> git commit -m "pushing changes for deploymnet testing purpose"

[master f2734ba] pushing changes for deploymnet testing purpose
 1 file changed, 1 insertion(+)
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-> git push origin master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 521 bytes | 130.00 KiB/s, done.
Total 6 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/jyotiraul/ElectroShop.git
   75a1fc4..f2734ba  master -> master
```



Changes reflect automatically.



5: Monitoring and Logging

Monitoring and performance analysis were enabled using AWS CloudWatch:
- Centralized logging from ECS containers.
- Performance dashboards created for CPU and memory metrics.
- Alerts configured using SNS for threshold violations.

Centeralized logging –

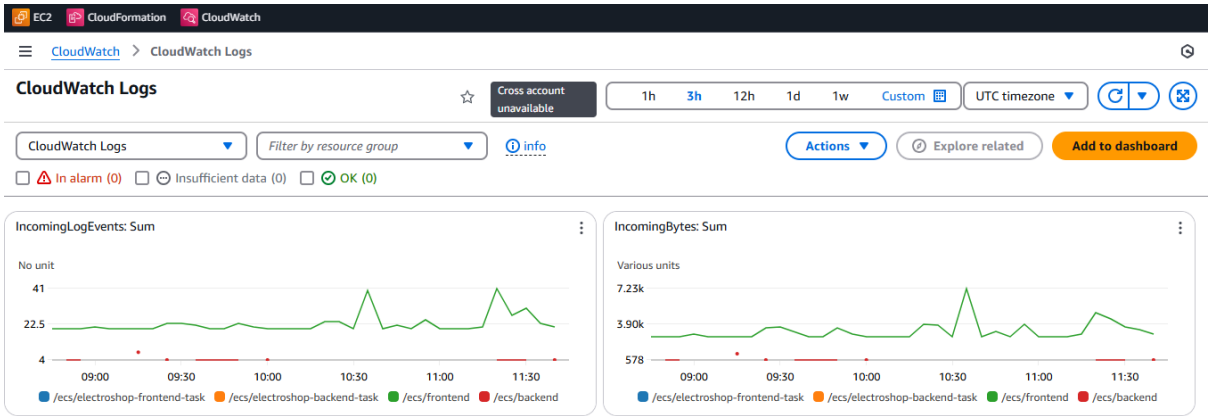
Task definitions > backend > Revision 4 > JSON

```
19          {
20            "name": "MONGO_URI",
21            "value": "mongodb+srv://Jyotiraul74:Jyoti%40199
ssl=true&authSource=admin&retryWrites=true&w=majority"
22          }
23        ],
24        "mountPoints": [],
25        "volumesFrom": [],
26        "logConfiguration": {
27          "logDriver": "awslogs",
28          "options": {
29            "awslogs-group": "/ecs/backend",
30            "awslogs-create-group": "true",
31            "awslogs-region": "ap-south-1",
32            "awslogs-stream-prefix": "ecs"
33          }
```

Task definitions > frontend > Revision 4 > JSON

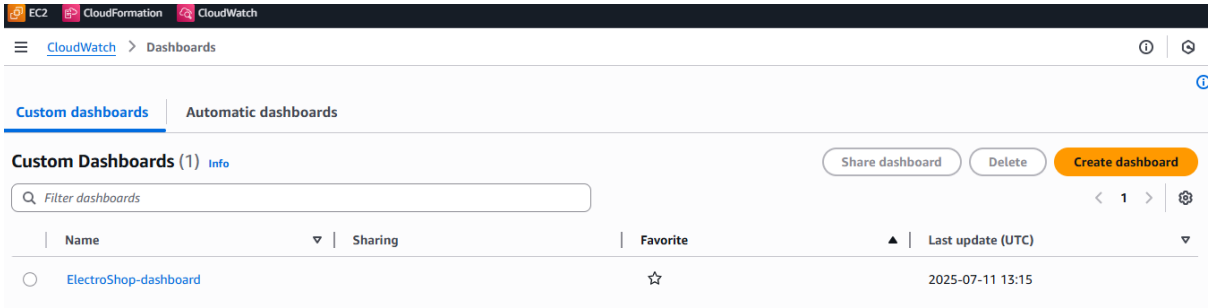
```
17          "essential": true,
18          "environment": [],
19          "mountPoints": [],
20          "volumesFrom": [],
21          "logConfiguration": {
22            "logDriver": "awslogs",
23            "options": {
24              "awslogs-group": "/ecs/frontend",
25              "awslogs-create-group": "true",
26              "awslogs-region": "ap-south-1",
27              "awslogs-stream-prefix": "ecs"
28            }
29          },
30          "systemControls": []
```

Logs appears in CloudWatch under ecs/



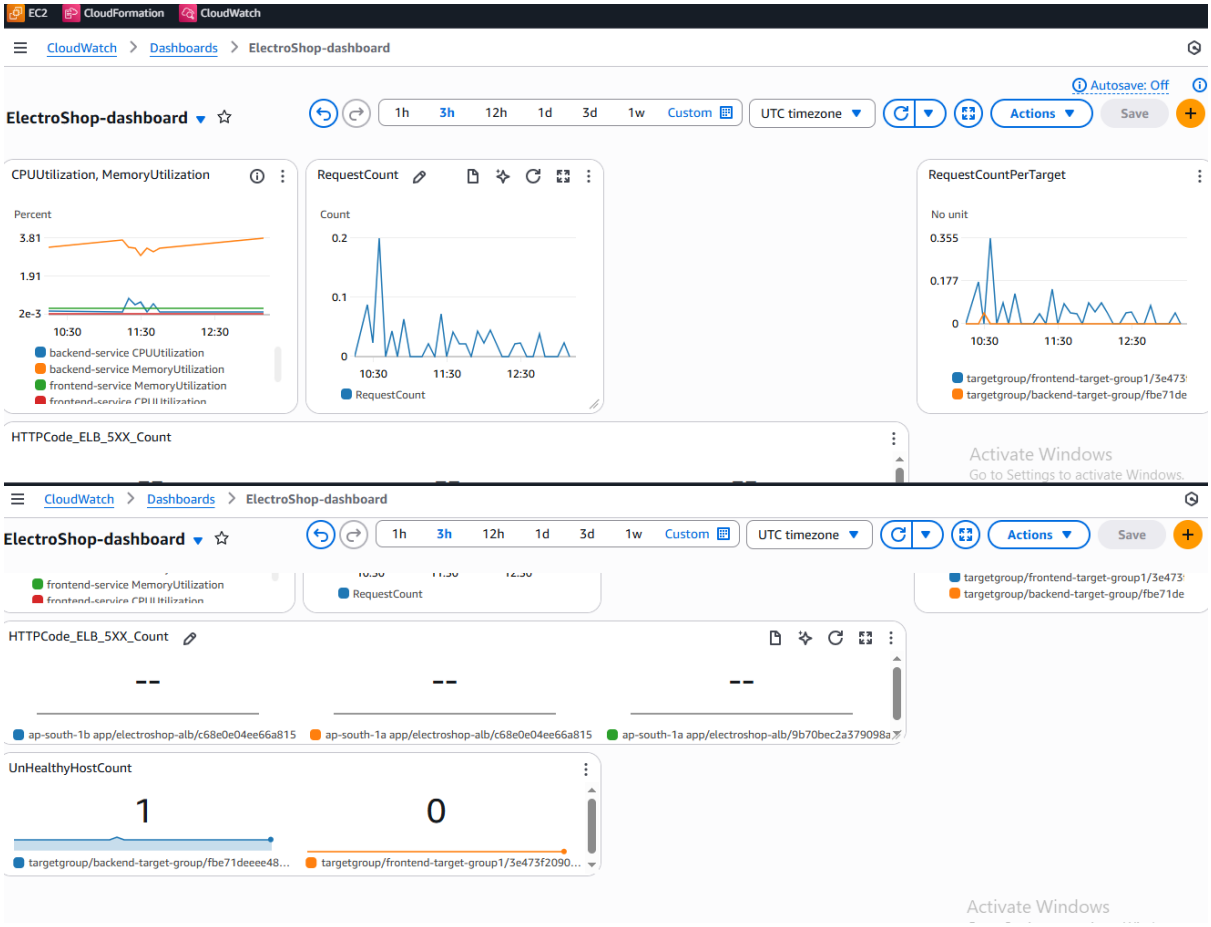
1. Performace Dashboards

Go to Cloudwatch -> Dashboard -> create Dashboard



Add Ecs metrics –

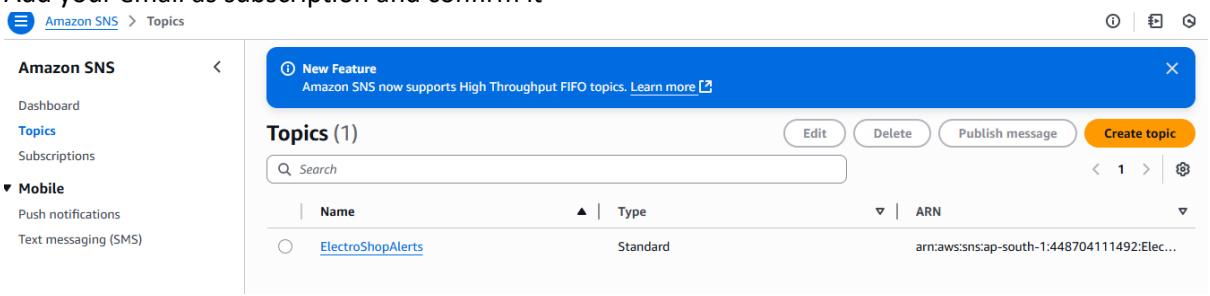
Click on + -> Widget Configuration -> metrics -> lines-> … > create widget.

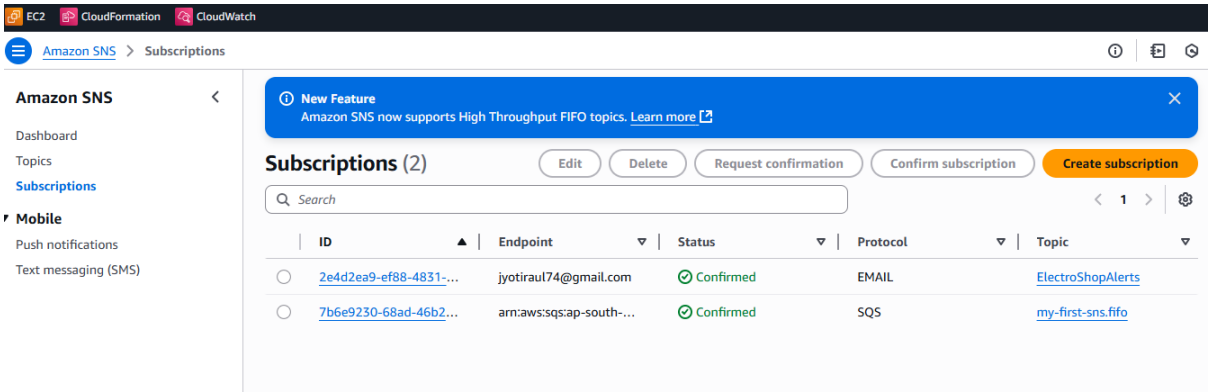


2. Automated alerting –

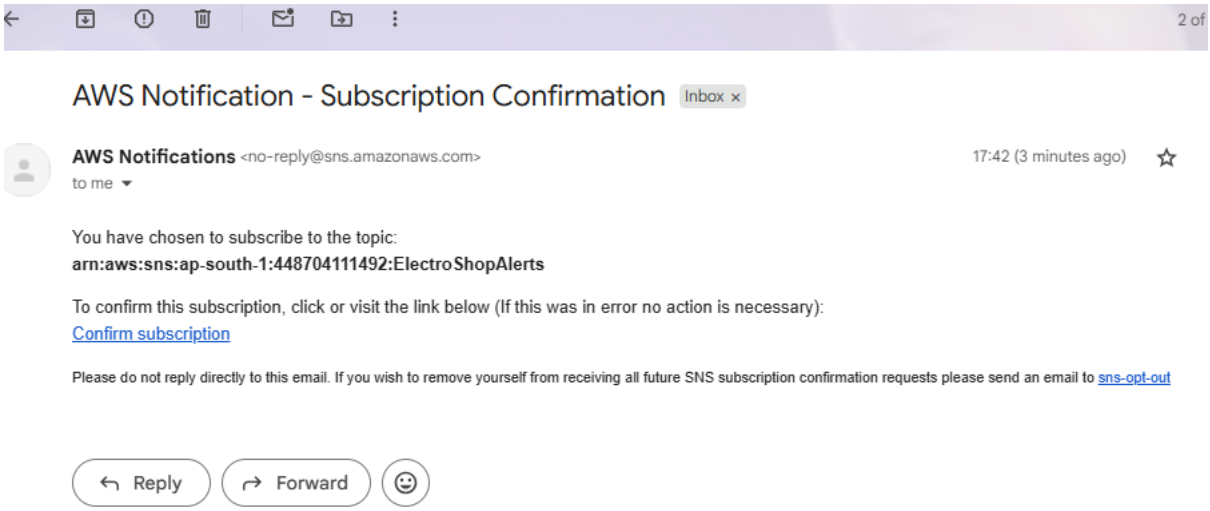
Go to sns -> create topic (ElectroShopAlerts)

Add your email as subscription and confirm it

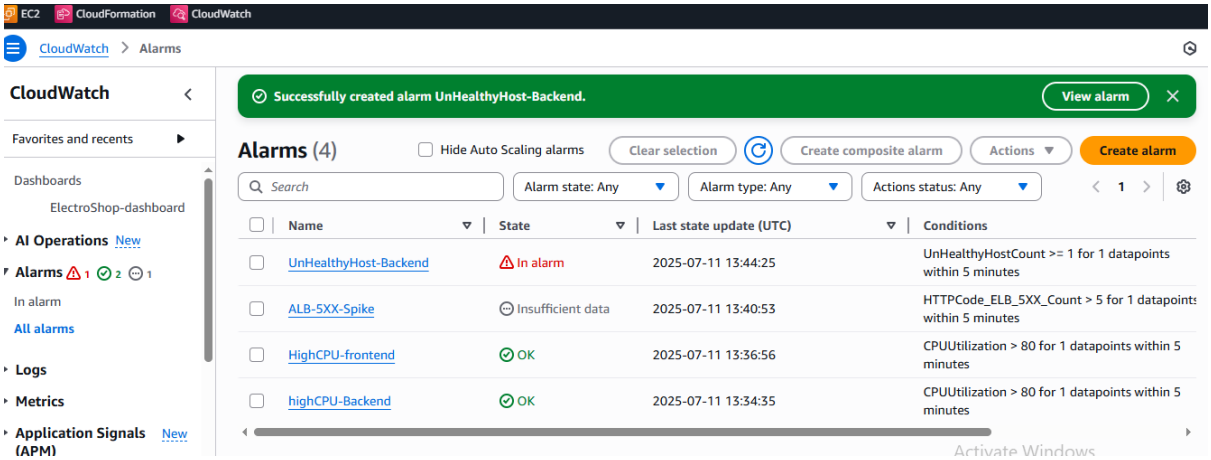




Add your email as subscription and confirm it.



Cloudwatch -> alarm -> create alaram ->
For cpu utilization—backend frontend



6: Security Hardening

Security was strengthened across the stack:
- Secrets (Mongo_URI, JWT secret) stored securely using AWS Secrets Manager.
- Automated vulnerability scanning added using Trivy and GitHub Actions workflow.
- Network protection tested using AWS WAF and Shield.

Secure secret handling-

Go to aws console -> secrets manager

Click -store a new secret

Choose – other type of secret

Add key value pair –
Key Mongo_URI and value
Key JWT_secret and value
Name it and store.

1. Automated Vulnerability scanning-

   I have added code for trivy in .github/workflow/deploy.yml file.
   Note: you can check report in pipeline also.

Build and Deploy Both Services
succeeded 2 minutes ago in 1m 56s

Search logs

Trivy Scan - Backend Image

```
309 | app/node_modules/@types/webidl-conversions/package.json | node-pkg |  0  |  -  |
310 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
311 | app/node_modules/@types/whatwg-url/package.json          | node-pkg |  0  |  -  |
312 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
313 | app/node_modules/accepts/package.json                    | node-pkg |  0  |  -  |
314 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
315 | app/node_modules/array-flatten/package.json              | node-pkg |  0  |  -  |
316 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
317 | app/node_modules/body-parser/package.json                | node-pkg |  0  |  -  |
318 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
319 | app/node_modules/bson/package.json                       | node-pkg |  0  |  -  |
320 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
321 | app/node_modules/bson/vendor/base64/package.json         | node-pkg |  0  |  -  |
322 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
323 | app/node_modules/bson/vendor/text-encoding/package.json  | node-pkg |  0  |  -  |
324 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
325 | app/node_modules/bytes/package.json                      | node-pkg |  0  |  -  |
326 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
327 | app/node_modules/call-bind-apply-helpers/package.json    | node-pkg |  0  |  -  |
328 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
329 | app/node_modules/call-bound/package.json                 | node-pkg |  0  |  -  |
330 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
331 | app/node_modules/content-disposition/package.json        | node-pkg |  0  |  -  |
332 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
333 | app/node_modules/content-type/package.json               | node-pkg |  0  |  -  |
334 ├──────────────────────────────────────────────────────────┼──────────┼─────────┼─────┤
335 | app/node_modules/cookie-signature/package.json           | node-pkg |  0  |  -  |
```

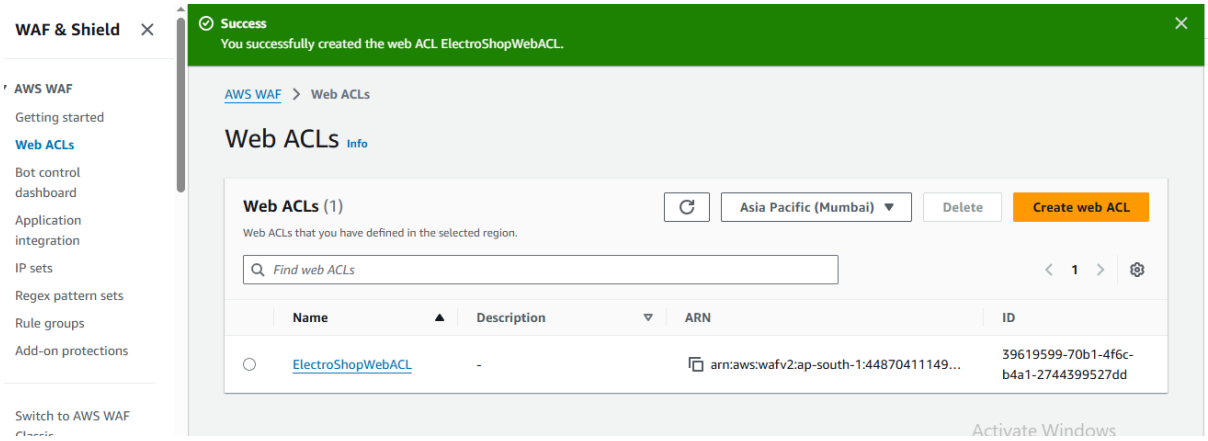
```
383 | app/node_modules/hasown/package.json                                  | node-pkg |        0        |    -    |
384 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
385 | app/node_modules/http-errors/package.json                             | node-pkg |        0        |    -    |
386 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
387 | app/node_modules/iconv-lite/package.json                              | node-pkg |        0        |    -    |
388 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
389 | app/node_modules/inherits/package.json                                | node-pkg |        0        |    -    |
390 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
391 | app/node_modules/ipaddr.js/package.json                               | node-pkg |        0        |    -    |
392 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
393 | app/node_modules/kareem/package.json                                  | node-pkg |        0        |    -    |
394 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
395 | app/node_modules/math-intrinsics/package.json                         | node-pkg |        0        |    -    |
396 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
397 | app/node_modules/media-typer/package.json                             | node-pkg |        0        |    -    |
398 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
399 | app/node_modules/memory-pager/package.json                            | node-pkg |        0        |    -    |
400 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
401 | app/node_modules/merge-descriptors/package.json                       | node-pkg |        0        |    -    |
402 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
403 | app/node_modules/methods/package.json                                 | node-pkg |        0        |    -    |
404 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
405 | app/node_modules/mime-db/package.json                                 | node-pkg |        0        |    -    |
406 ├──────────────────────────────────────────────────────────────────────┼──────────┼─────────────────┼─────────┤
407 | app/node_modules/mime-types/package.json                              | node-pkg |        0        |    -    |
```

Trivy Scan - Frontend Image

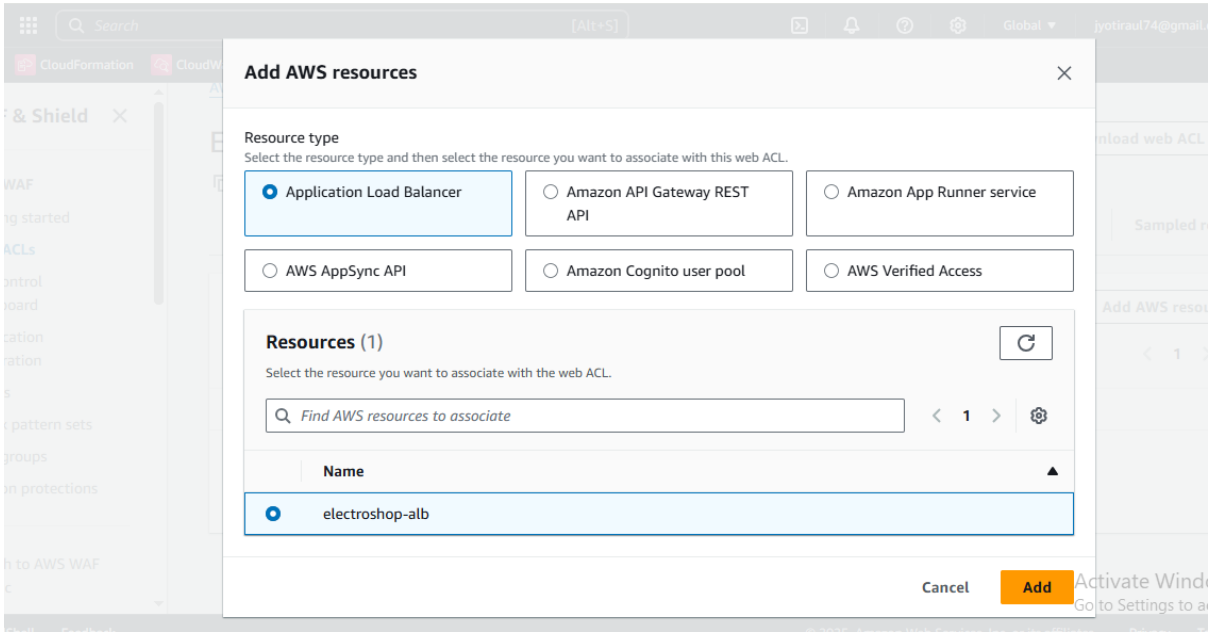
```
293 Report Summary
294
295 ┌──────────────────────────────────────────────────────────────────────┬──────┬─────────────────┬─────────┐
296 |                                Target                                | Type | Vulnerabilities | Secrets |
297 ├──────────────────────────────────────────────────────────────────────┼──────┼─────────────────┼─────────┤
298 | *** (alpine | alpine |        2        |    -    |
299 | 3.21.3)                                                              |      |                 |         |
300 └──────────────────────────────────────────────────────────────────────┴──────┴─────────────────┴─────────┘
301 Legend:
302 - '-': Not scanned
303 - '0': Clean (no security findings detected)
304
305
306 For OSS Maintainers: VEX Notice
307 --------------------------------
308 If you're an OSS maintainer and Trivy has detected vulnerabilities in your project that you believe are not actually exploitable, consider issuing a VEX
    (Vulnerability Exploitability eXchange) statement.
309 VEX allows you to communicate the actual status of vulnerabilities in your project, improving security transparency and reducing false positives for you
    users.
310 Learn more and start using VEX: https://trivy.dev/v0.64/docs/supply-chain/vex/repo#publishing-vex-documents
311
312 To disable this notice, set the TRIVY_DISABLE_VEX_NOTICE environment variable.
313
314
```

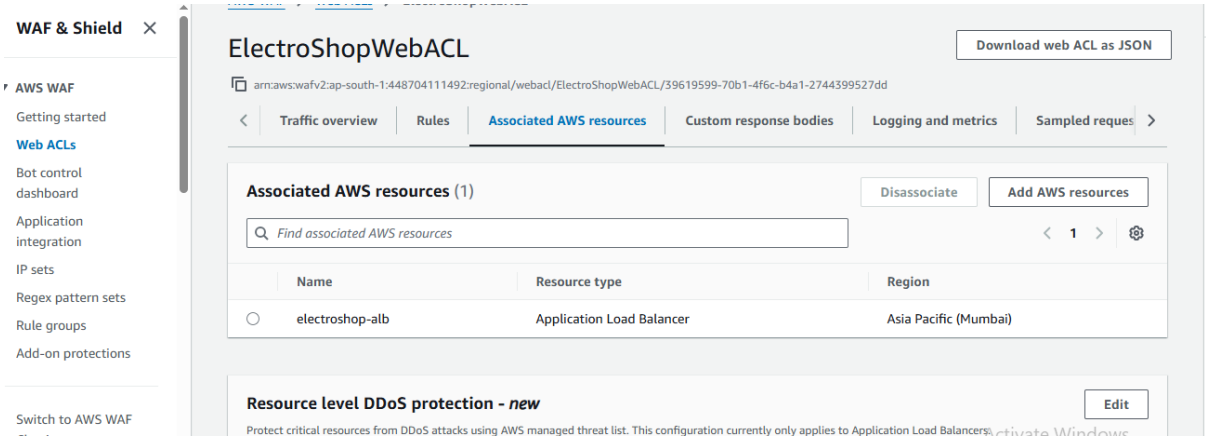
3. Network Protection

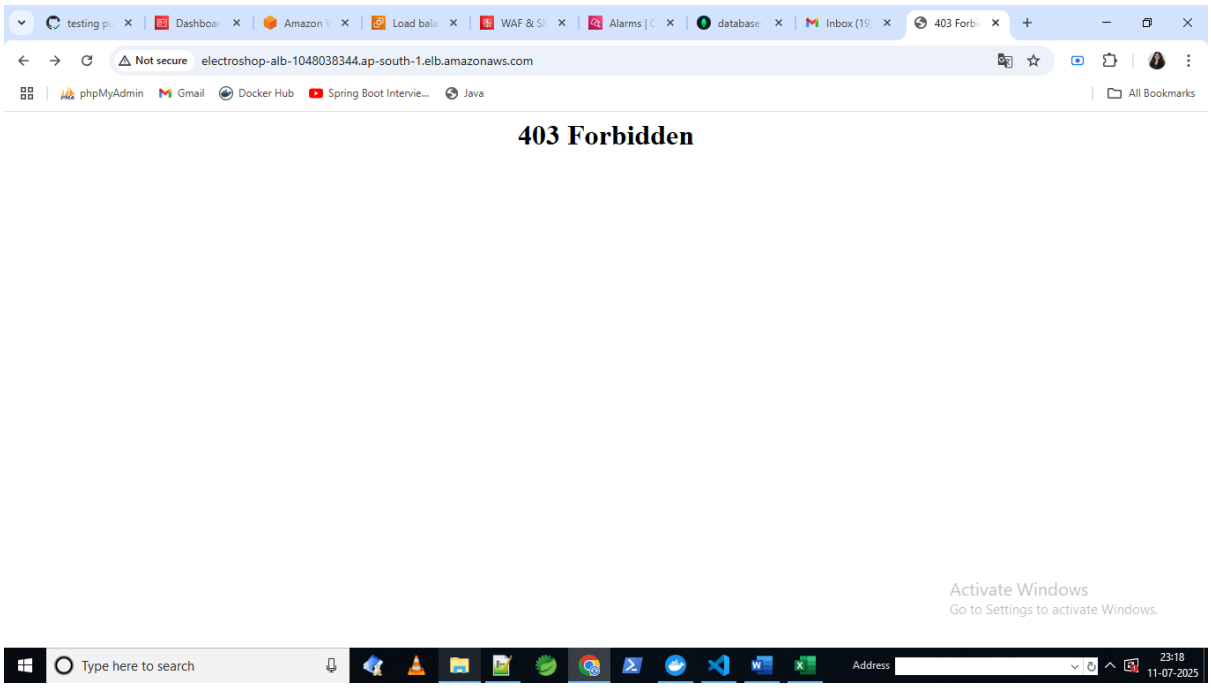
Aws console -> WAF & shield


WAF & Shield
Success
You successfully created the web ACL ElectroShopWebACL.
AWS WAF
Getting started
Web ACLs
Bot control dashboard
Application integration
IP sets
Regex pattern sets
Rule groups
Add-on protections
Switch to AWS WAF
AWS WAF > Web ACLs
Web ACLs Info
Web ACLs (1)
Web ACLs that you have defined in the selected region.
Asia Pacific (Mumbai)
Delete
Create web ACL
Find web ACLs
Name
Description
ARN
ID
ElectroShopWebACL
-
arn:aws:wafv2:ap-south-1:44870411149...
39619599-70b1-4f6c-b4a1-2744399527dd

AWS WAF > Web ACLs > ElectroShopWebACL
ElectroShopWebACL
Download web ACL as JSON
arn:aws:wafv2:ap-south-1:448704111492:regional/webacl/ElectroShopWebACL/39619599-70b1-4f6c-b4a1-2744399527dd
Traffic overview
Rules
Associated AWS resources
Custom response bodies
Logging and metrics
Rules (3)
Edit
Delete
Add rules
Find rules
Name
Action
Priority
Custom response
AWS-AWSManagedRulesSQLiRuleSet
Use rule actions
0
AWS-AWSManagedRulesKnownBadInputsRuleSet
Use rule actions
1
AWS-AWSManagedRulesCommonRuleSet
Use rule actions
2


Add AWS resources
Resource type
Select the resource type and then select the resource you want to associate with this web ACL.
Application Load Balancer
Amazon API Gateway REST API
Amazon App Runner service
AWS AppSync API
Amazon Cognito user pool
AWS Verified Access
Resources (1)
Select the resource you want to associate with the web ACL.
Find AWS resources to associate
Name
electroshop-alb
Cancel
Add





Security is applied. It will gives error. Site is not secured.

So I disassociate aws resource to continue website.

phpMyAdmin Gmail Docker Hub Spring Boot Intervie... Java All Bookn

ElectroShop

Home Shop Now About Us Contact Us

Your Digital Gadget Hub

The latest and greatest in electronics, all in one place.

Welcome to ElectroShop

Shop Now

Activate Windows

ElectroShop

Home Shop Now About Us Contact Us

Search Products

Sort By
Default

Categories

- All Products
- Laptops
- Smartphones
- Headphones
- Cameras
- Smartwatches
- Drones

Laptop

ZenBook Pro Duo
Laptops
★★★★★ (4.8)
$1,999.99
View Details

Laptop

Spectre x360
Laptops
★★★★½ (4.7)
$1,549.00
View Details

Gaming Laptop

GamerBook Fire
Laptops
★★★★★ (4.9)
$2,499.50
View Details

Activate Windows
Go to Settings to activate Windows.

phpMyAdmin Gmail Docker Hub Spring Boot Intervie... Java All Book

ElectroShop

Home Shop Now About Us Contact Us

About ElectroShop

Your one-stop destination for the latest and greatest in electronics.

Welcome to ElectroShop, your premier digital gadget hub. Our mission is to provide tech enthusiasts with top-quality electronics, from cutting-edge laptops and smartphones to high-fidelity headphones and professional-grade cameras. We believe in the power of technology to simplify and enrich lives, and we are committed to bringing you the best products the industry has to offer.

Our values are simple: quality, innovation, and customer satisfaction. We carefully curate our product selection to ensure every item meets our high standards of performance and reliability. At ElectroShop, we're not just selling electronics; we're providing tools that empower creativity, enhance productivity, and connect people.

Thank you for choosing ElectroShop. We look forward to serving you and helping you find the perfect gadget to fit your lifestyle.

Thank you for choosing ElectroShop.

Activate Windows

phpMyAdmin Gmail Docker Hub Spring Boot Intervie... Java All Bookm

ElectroShop

Home Shop Now About Us Contact Us

Contact Us

Have a question or feedback? Drop us a line!

Get in Touch

Address: 123 Tech Avenue, Silicon Valley, CA 94000
Phone: (123) 456-7890
Email: support@electroshop.com

Name *

Email *

Subject *

Message *

Activate Windows
Go to Settings to activate Windows.

ElectroShop

Home Shop Now About Us Contact Us 1

Shopping Cart

Laptop

ZenBook Pro Duo
$1,999.99

− 1 +

Order Summary

Subtotal	$1,999.99
Shipping	$0.00
Total	**$1,999.99**

Proceed to Checkout

phpMyAdmin Gmail Docker Hub Spring Boot Intervie... Java All Bookm

ElectroShop

Home Shop Now About Us Contact Us 1

Checkout

Shipping Details

Full Name *: Devansh Dalvi

Email Address *: dev@gmail.com

Address Line 1 *: Dadar

City *: mumbai

State/Province *: maharashtra

Postal Code *: 400612

Country *: india

Order Summary

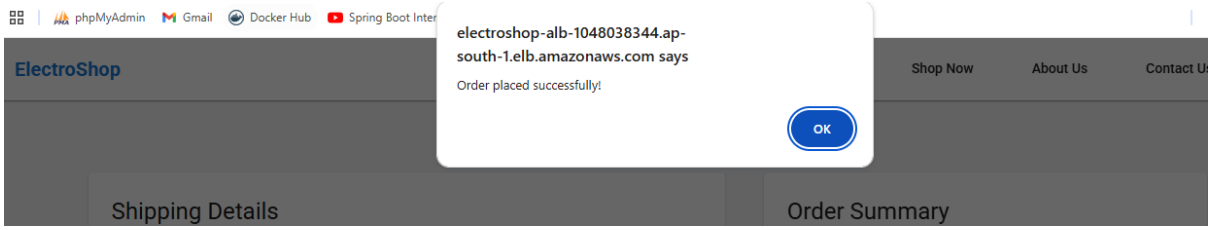
ZenBook Pro Duo
Quantity: 1 $1,999.99

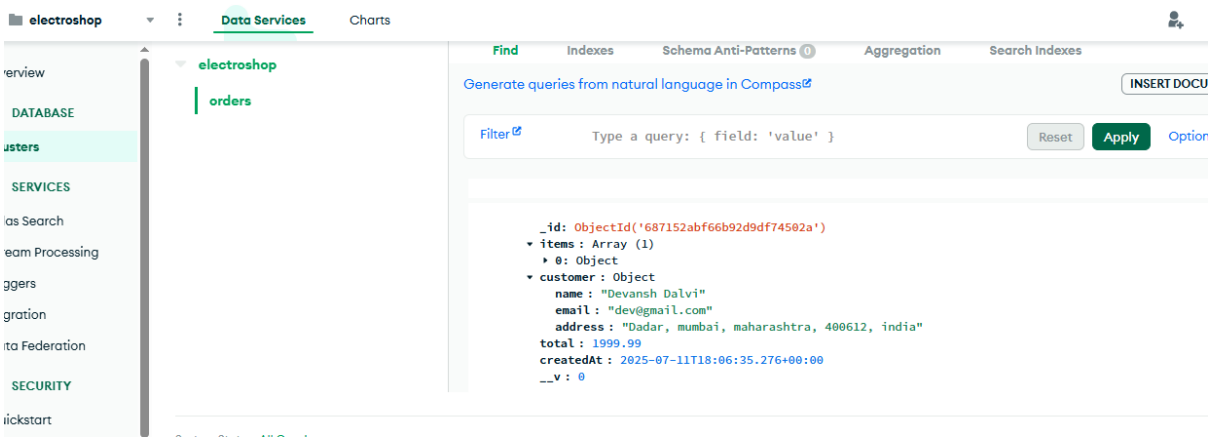
Total **$1,999.99**

Place Order

Payment Method

Activate Windows
Go to Settings to activate Windows.





Challenges Faced

- Difficulty in VPC and NAT Gateway configuration for internet access in private subnets.
- Handling secret values securely during ECS task execution.
- Troubleshooting deployment failures in ECS due to misconfigured task definitions.
- Initial setup of CloudWatch metrics and SNS for alerting.

Key Achievements

- Fully automated deployment pipeline integrated with GitHub Actions.
- Secure and scalable AWS infrastructure provisioning using best practices.
- Real-time performance monitoring and alerts for ECS-based workloads.
- Hardened application security with secrets management and vulnerability scanning.
- Successful deployment and public access of application via Load Balancer.