

ElectroShop

1. Project Setup and Containerisation

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\mongodb-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).
PS C:\Windows\system32>
```

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' }
npm WARN EBADENGINE }
```

```
back&#223;end > .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 Node.js Driver 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 since Node.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 stand
  and 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: 'onAfterSetupMiddleware' is depreca
  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: 'onBeforeSetupMiddleware' is depreca
  ated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view electromart-app in the browser.

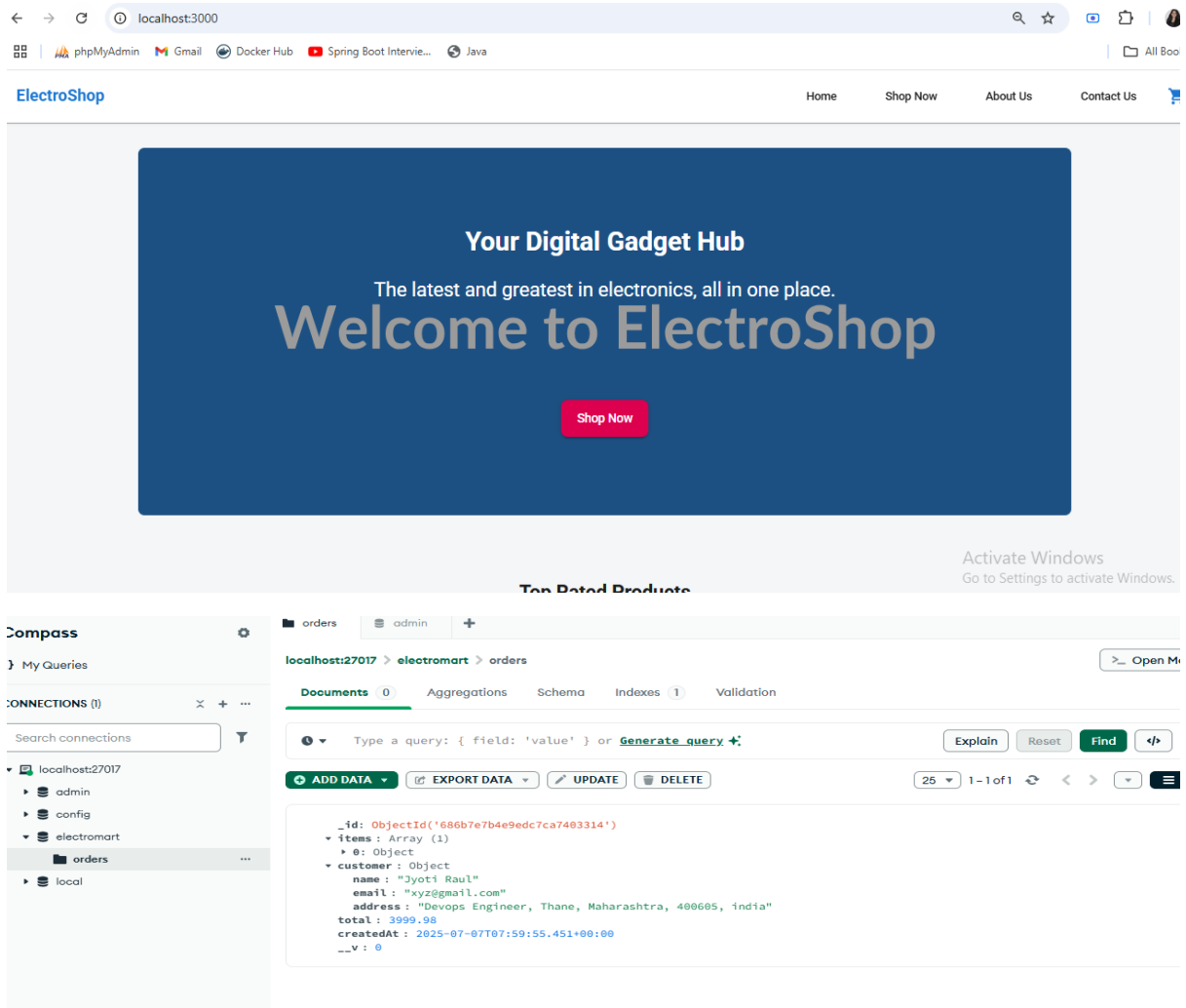
  Local:            http://localhost:3000
  On Your Network:  http://192.168.56.1:3000

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 ./

```

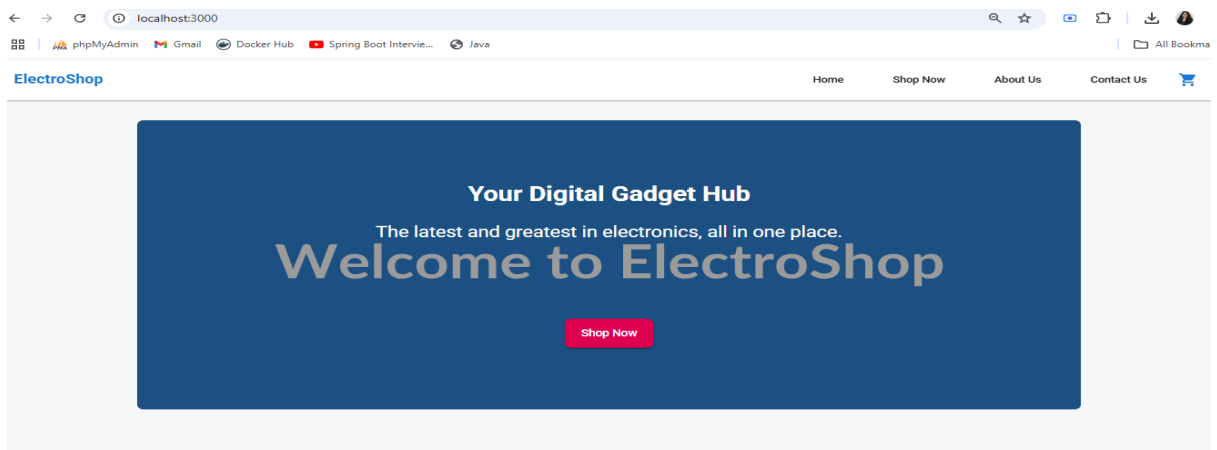
```

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/de
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>



localhost:3000/checkout

ElectroShop

Home Shop Now About Us Contact Us

Shipping Details

Full Name *	Email Address *
ragav juyal	ragav@gmail.com
Address Line 1 *	
dadar	
City *	State/Province *
Mumbai	Maharashtra
Postal Code *	Country *
400610	India

Order Summary

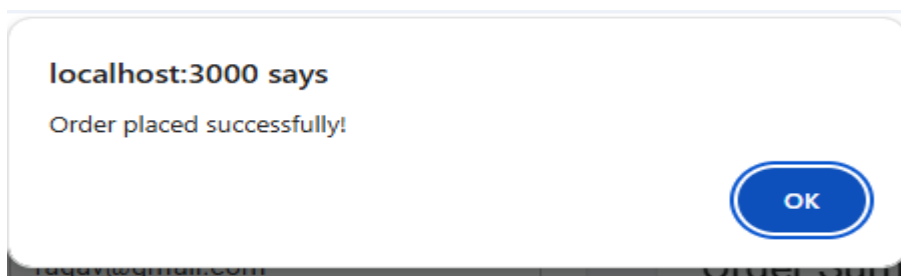
ZenBook Pro Duo	\$3,999.98
Quantity: 2	
<hr/>	
Total	\$3,999.98
<button>Place Order</button>	

Payment Method

☐ Credit Card (Simulated)

☐ PayPal (Simulated)

☒ Cash on Delivery



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/core/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> 

```

Week 2— Aws infrastructure and provisioning

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 > Your VPCs > Create VPC

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

VPC settings

Resources to create [Info](#)

Create only the VPC resource or the VPC and other networking resources.

☐ VPC only ☒ VPC and more

Name tag auto-generation [Info](#)

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☒ Auto-generate

electroshop-vpc

IPv4 CIDR block [Info](#)

Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block

☐ Amazon-provided IPv6 CIDR block

Tenancy [Info](#)

Default

Number of Availability Zones (AZs) [Info](#)

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 2 3

Preview

VPC [Show details](#)

Your AWS virtual network

electroshop-vpc-vpc

Subnets (4)

Subnets within this VPC

ap-south-1a

- electroshop-vpc-subnet-public1-ap-
- electroshop-vpc-subnet-private1-

ap-south-1b

- electroshop-vpc-subnet-public2-ap-
- electroshop-vpc-subnet-private2-

Route tables (3)

Route network traffic to resources

- electroshop-vpc-rtb-public
- electroshop-vpc-rtb-private1-ap-south-
- electroshop-vpc-rtb-private2-ap-south-

Network connections (2)

Connections to other networks

- electroshop-vpc-igw
- electroshop-vpc-vpc-s3

VPC > Your VPCs > Create VPC

Customize AZs

Number of public subnets [Info](#)

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 | 2

Number of private subnets [Info](#)

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 | 2 | 4

Customize subnets CIDR blocks

NAT gateways (\$) [Info](#)

Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway

None | In 1 AZ | 1 per AZ

VPC endpoints [Info](#)

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None | S3 Gateway

DNS options [Info](#)

☒ Enable DNS hostnames

☒ Enable DNS resolution

Additional tags

Add tags to the VPC and all resources within the VPC. Do not set the Name tag

VPC look like→

The screenshot shows the AWS VPC console for a VPC with ID vpc-04459c7e5e7660396. The left sidebar shows the navigation menu with 'Virtual private cloud' and 'Security' sections. The main content area displays the VPC details, including its ID, state (Available), DNS resolution (Enabled), Main network ACL (acl-00b7fe169232a5571), IPv6 CIDR (Network border group), DHCP option set (dopt-0578df08d41d63845), Block Public Access (Off), IPv4 CIDR (10.0.0.0/16), Route 53 Resolver DNS Firewall rule groups, DNS hostnames (Enabled), Main route table (rtb-0cbb9ffdf3f7d678a), and IPv6 pool. Below the details is a 'Resource map' section.

The screenshot shows the AWS VPC console for the 'Subnets' page. The left sidebar shows the navigation menu. The main content area displays a list of subnets for the VPC vpc-04459c7e5e7660396. The table has columns for Name, Subnet ID, State, VPC, Block Public Access, and IPv4 CIDR. All subnets are in the 'Available' state.

Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR
electroshop-vpc-subnet-public2-ap-south-1b	subnet-04c1d3b138e2c778c	Available	vpc-04459c7e5e7660396 elec...	Off	10.0.0.0/16
electroshop-vpc-subnet-private1-ap-south-1a	subnet-03ddd534fd4e8e2d3	Available	vpc-04459c7e5e7660396 elec...	Off	10.0.0.0/16
electroshop-vpc-subnet-private2-ap-south-1b	subnet-081f912594a5ab000	Available	vpc-04459c7e5e7660396 elec...	Off	10.0.0.0/16
electroshop-vpc-subnet-public1-ap-south-1a	subnet-02446efa52681d70a	Available	vpc-04459c7e5e7660396 elec...	Off	10.0.0.0/16

The screenshot shows the AWS VPC console for the 'Route tables' page. The left sidebar shows the navigation menu. The main content area displays a list of route tables for the VPC vpc-04459c7e5e7660396. The table has columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, and VPC. The route tables are associated with the VPC.

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
-	rtb-0cbb9ffdf3f7d678a	-	-	Yes	vpc-04459c7e5e7660396 ele
electroshop-vpc-rtb-private2-ap-south-...	rtb-065eac8b3114f59a0	subnet-081f912594a5ab...	-	No	vpc-04459c7e5e7660396 ele
electroshop-vpc-rtb-private1-ap-south-...	rtb-0f1717698cd819a32	subnet-03ddd534fd4e8e...	-	No	vpc-04459c7e5e7660396 ele
electroshop-vpc-rtb-public	rtb-0c5ded9f9cc2a6bc8	2 subnets	-	No	vpc-04459c7e5e7660396 ele
-	rtb-0d09137b0e24303e8	-	-	Yes	vpc-05207bcd537ef5172

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.

Elastic IP address 13.202.199.27 (eipalloc-0fe80d5eb49cf185b) allocated.

NAT gateway settings

Name - optional

Create a tag with a key of 'Name' and a value that you specify.

my-nat-gateway-01

The name can be up to 256 characters long.

Subnet

Select a subnet in which to create the NAT gateway.

subnet-0607f104ca9dfb4d8 (electroshop-subnet-public1-ap-south-1a)

Connectivity type

Select a connectivity type for the NAT gateway.

- ☒ Public
☐ Private

Elastic IP allocation ID [Info](#)

Assign an Elastic IP address to the NAT gateway.

eipalloc-0fe80d5eb49cf185b

[Allocate Elastic IP](#)

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-

- For destination 0.0.0.0/0, the **NAT Gateway** should now appear in the dropdown
- Select it and **save changes**
- Repeat for the second private route table

Edit routes

Destination	Target	Status	Propagated
pl-78a54011	vpce-07f4af18d9882a005	Active	No
10.0.0.0/16	local	Active	No
Q 0.0.0.0/0	Q local		
	NAT Gateway	-	No
	Q nat-05e4e5b83b24034db		

[Add route](#)

[Cancel](#) [Preview](#) [Save changes](#)

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

Entity and Access management (IAM)

Search IAM

board

ess management

groups

You can attach up to 10 managed policies.

Filter by Type

Policy name	Type	Attached entities
AmazonEC2ContainerServiceRole	AWS managed	2
AmazonECSTaskExecutionRole...	AWS managed	2
CloudWatchLogsFullAccess	AWS managed	2

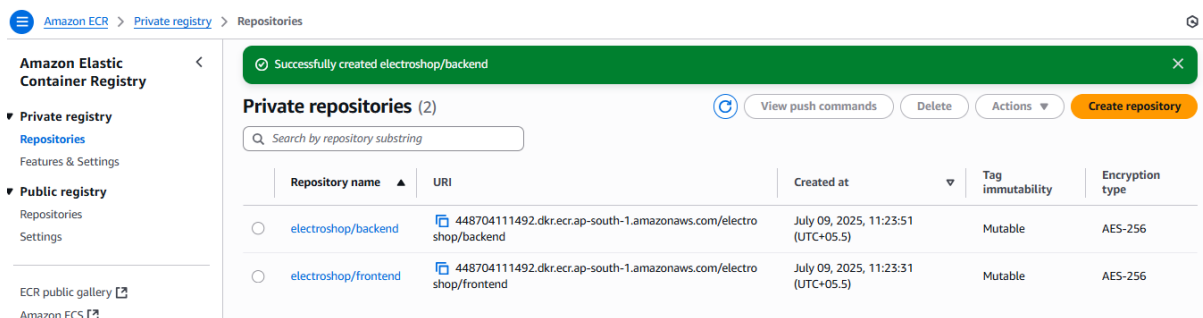
4. Create Elastic Container Registry-

Go to Amazon ECR > Create Repository

Create:

- electroshop/frontend
- electroshop/backend

Set them to Private Repositories

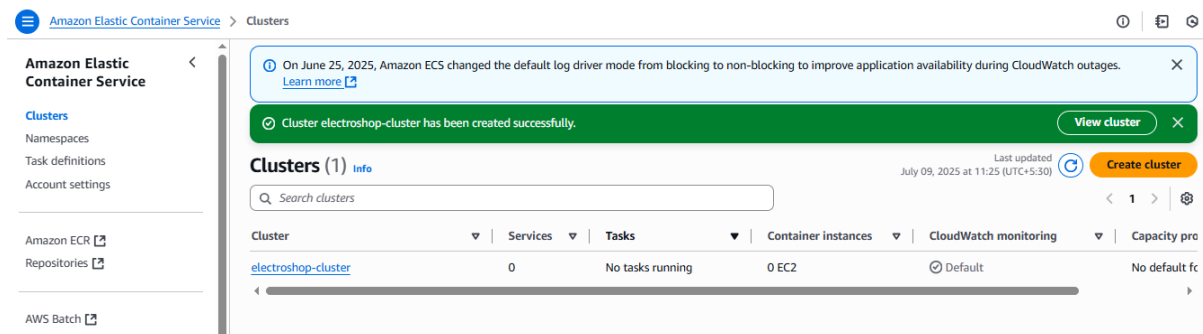


5. Create ECS Cluster

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**

EC2 > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details
Security group name Info
electroshop-alb-sg
Name cannot be edited after creation.
Description Info
Allow inbound HTTP/HTTPS for ALB
VPC Info
vpc-0e1b34f7f4438a3c9 (electroshop-vpc)

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional	
HTTP	TCP	80	Anywhere...		Delete
HTTPS	TCP	443	Anywhere...		Delete

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Outbound rules Info

Create a second one:

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

EC2 > Security Groups > sg-040a86ff4cfe9938 - electroshop-ecs-sg > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sg-0a5e752806d0ea47c	HTTP	TCP	80	Cust...		Delete
sg-0b4f1b15490ae438e	Custom TCP	TCP	5000	Cust...		Delete

Week 3 : Manual Deployment to AWS-

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`

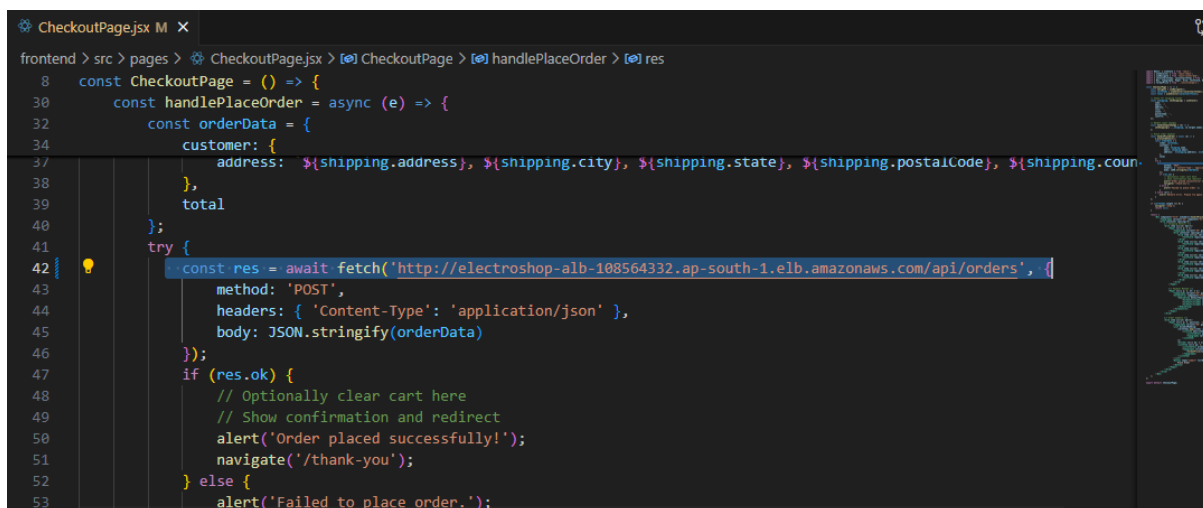
Replace <your-account-id> with your actual AWS account ID.

```
PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop> aws ecr get-login-password --region ap-south-1 | docker 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 find 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 find 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-tokens/
```

Tag and Push Docker Images

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



```
frontend > src > pages > CheckoutPage.jsx > CheckoutPage > handlePlaceOrder > res
8  const CheckoutPage = () => {
30    const handlePlaceOrder = async (e) => {
32      const orderData = {
34        customer: {
37          address: `${shipping.address}, ${shipping.city}, ${shipping.state}, ${shipping.postalCode}, ${shipping.country},
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 :

`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                                0.4s
[+] Building 2.1s (2/4)
=> [internal] load build definition from Dockerfile                                0.4s
[+] Building 2.2s (2/4)
=> [internal] load build definition from Dockerfile                                0.4s
[+] Building 313.1s (16/16) FINISHED
```

```

PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\frontend> docker tag electroshop-frontend:latest 448704111492.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
[2025-07-09T11:41:48.000Z][docker-credential-desktop][W] Windows version might not be up-to-date: The system cannot find 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                                0.2s
[+] Building 8.0s (12/12) FINISHED

```

```

PS C:\assignment\Sparknet-Innovation\Sparknet-ElectroShop\ElectroShop-\backend> docker tag electroshop-backend:latest 448704111492.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.amazonaws.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 find 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:

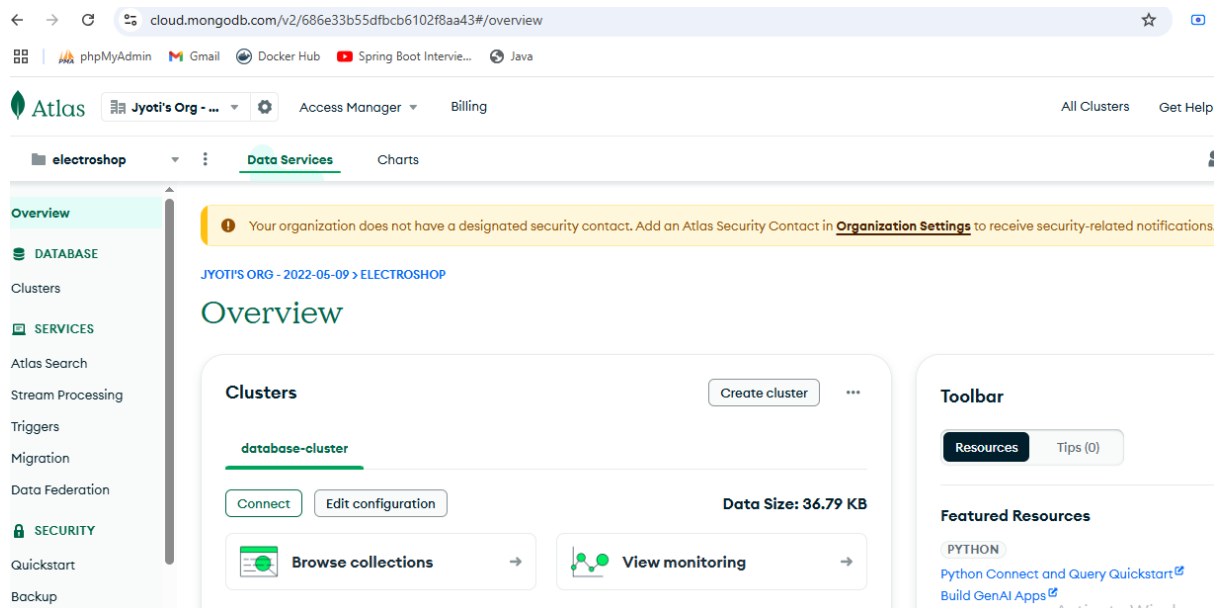
Login to <https://cloud.mongodb.com/>

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

EC2 CloudFormation CloudWatch

Amazon Elastic Container Service > Task definitions > electroshop-backend-task > Revision 1 > Containers

Amazon Elastic Container Service

- Clusters
- Namespaces
- Task definitions**
- Account settings

Amazon ECR Repositories

AWS Batch

Documentation Discover products Subscriptions

On June 25, 2025, Amazon ECS changed the default log driver mode from blocking to non-blocking to improve application availability during CloudWatch outages. [Learn more](#)

Task definition successfully created
electroshop-backend-task:1 has been successfully created. You can use this task definition to deploy a service or run a task. [View task definition](#)

electroshop-backend-task:1 [Deploy](#) [Actions](#) [Create new revision](#)

Overview [Info](#)

ARN arn:aws:ecs:ap-south-1:44870411:1492:task-definition/electroshop-backend-task:1	Status ACTIVE	Time created July 09, 2025 at 12:27 (UTC+5:30)	App environment Fargate
Task role -	Task execution role ecsTaskExecutionRole	Operating system/Architecture Linux/X86_64	Network mode awsvpc
Fault injection <input type="radio"/> Turned off			

EC2 CloudFormation CloudWatch

Amazon Elastic Container Service > Task definitions > electroshop-frontend-task > Revision 1 > Containers

Amazon Elastic Container Service

- Clusters
- Namespaces
- Task definitions**
- Account settings

Amazon ECR Repositories

AWS Batch

Documentation Discover products Subscriptions

On June 25, 2025, Amazon ECS changed the default log driver mode from blocking to non-blocking to improve application availability during CloudWatch outages. [Learn more](#)

Task definition successfully created
electroshop-frontend-task:1 has been successfully created. You can use this task definition to deploy a service or run a task. [View task definition](#)

electroshop-frontend-task:1 [Deploy](#) [Actions](#) [Create new revision](#)

Overview [Info](#)

ARN arn:aws:ecs:ap-south-1:44870411:1492:task-definition/electroshop-frontend-task:1	Status ACTIVE	Time created July 09, 2025 at 12:32 (UTC+5:30)	App environment Fargate
Task role -	Task execution role ecsTaskExecutionRole	Operating system/Architecture Linux/X86_64	Network mode awsvpc
Fault injection <input type="radio"/> Turned off			

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 ,

EC2 > Target groups > frontend-target-group

EC2

Dashboard
EC2 Global View
Events

Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

Images
AMIs
AMI Catalog

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

frontend-target-group

Actions

Details
arn:aws:elasticloadbalancing:ap-south-1:448704111492:targetgroup/frontend-target-group/8c24087b1401475b

Target type IP	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-0e1b34f7f4438a3c9
IP address type IPv4	Load balancer None associated		

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

Targets | Monitoring | Health checks | Attributes | Tags

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

EC2 > Target groups > backend-target-group

EC2

Dashboard
EC2 Global View
Events

Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

Images
AMIs
AMI Catalog

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

backend-target-group

Actions

Details
arn:aws:elasticloadbalancing:ap-south-1:448704111492:targetgroup/backend-target-group/f3a2d2d9d2e8b8cd

Target type IP	Protocol : Port HTTP: 5000	Protocol version HTTP1	VPC vpc-0e1b34f7f4438a3c9
IP address type IPv4	Load balancer None associated		

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

Targets | Monitoring | Health checks | Attributes | Tags

EC2 > Load balancers > electroshop-alb

EC2

Dashboard
EC2 Global View
Events

Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

Images
AMIs
AMI Catalog

Load balancer ARN
arn:aws:elasticloadbalancing:ap-south-1:448704111492:loadbalancer/app/electroshop-alb/c68e0e04ee66a815

DNS name
electroshop-alb-108564332.ap-south-1.elb.amazonaws.com (A Record)

Listeners and rules | Network mapping | Resource map | Security | Monitoring | Integrations | Attributes | Capacity | Tags

Listeners and rules (1) Info

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

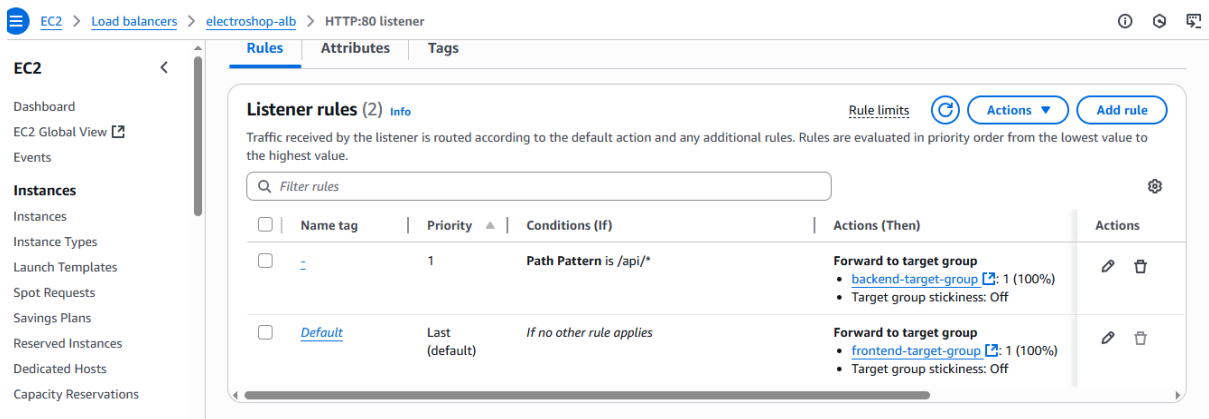
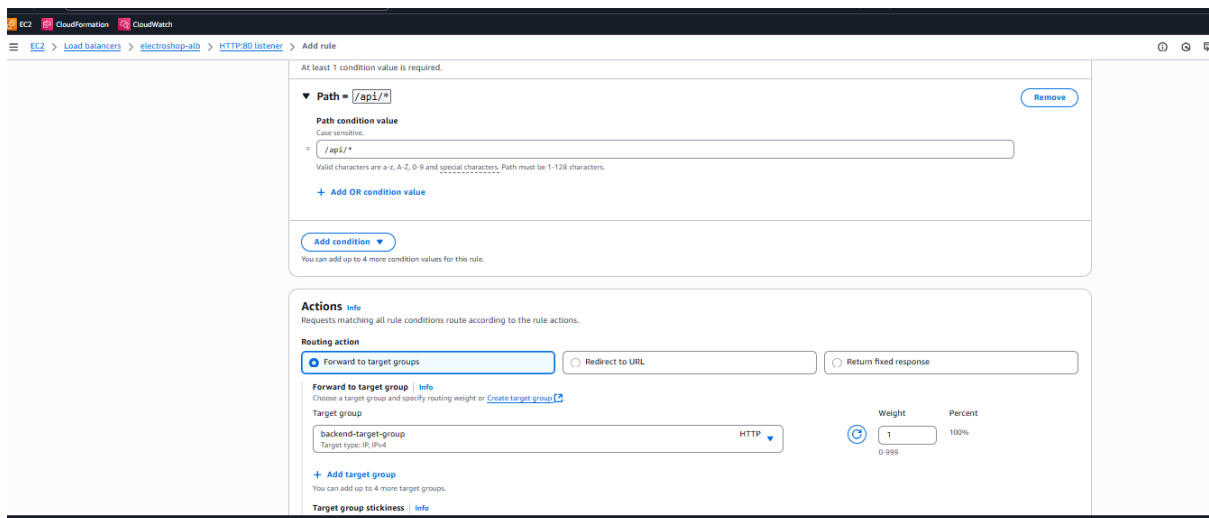
Filter listeners

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS
HTTP:80	Forward to target group <ul style="list-style-type: none"> frontend-target-group: 1 (100%) Target group stickiness: Off 	1 rule	ARN	Not applicable	Not applicable

Add the /api/* Rule for Backend Routing-

2 "View/edit rules" or a section called "Rules"

2 Click on "View/edit rules" for Listener: HTTP:80



6. Deploy ECS Services

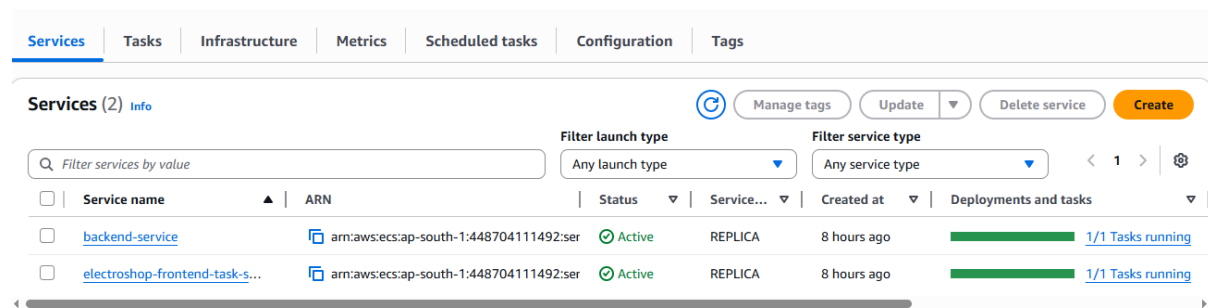
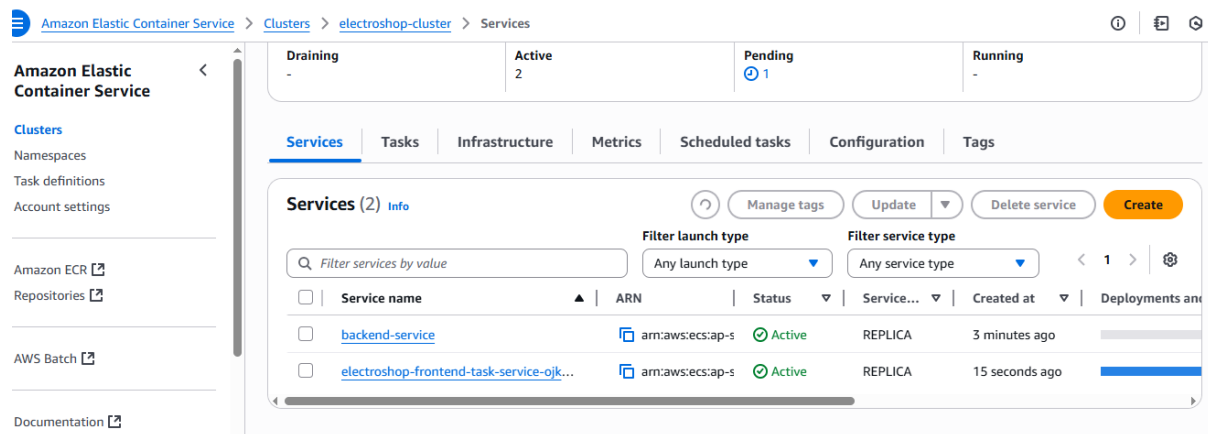
Go to: **ECS > Clusters > Your Cluster > Create Service**

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

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/>

