PROJECT 1 — 1. ECS with ALB, RDS, Route53, Acm, Iam and SecretsManager

a. Infrastructure Requirements:
i. ECS
1. Create an ECS Cluster with the service running in Private Subnets.
2. Services
a. WordPress Docker Image
b. Custom microservice (a lightweight Node.js application responding with "Hello from Microservice"). Share the Node.js code and DockerFile too.

3. Setup auto scaling based on CPU and Memory.
ii. RDS
1. Choose the appropriate instance type to be used by WordPress as a database.
2. Create custom user and password in RDS to be used with Wordpress that do not auto rotate.
3. Configure automated backups.
4. RDS instance should be deployed in the Private Subnets.
iii. SecretsManager

1. Store the RDS database credentials as secrets.
iv. IAM and Security
1. Configure the ECS task definition to use the secrets stored in AWS SecretsManager, ensuring the WordPress securely connects to the RDS instance.
2. Use IAM roles to grant the ECS service the necessary permissions to access secrets from AWS SecretsManager.
3. Security Groups should be the least privilege.
v. Application Load Balancer (ALB) & Domain Mapping
1. Set up an Application Load Balancer (ALB) in Public Subnets to handle incoming HTTP/HTTPS traffic.

2. Set up SSL certificate. The website should not open HTTP.
3. HTTP traffic should be redirected to HTTPS.
4. Configure the ALB to associate with a domain name.
"wordpress.<domain-name>"
"microservice.<domain-name>"

AWS Services Used

Networking & DNS

1. **Amazon VPC – Custom VPC with public and private subnets**
2. **Amazon Subnets – Public (ALB) and Private (ECS, RDS) subnets**
3. **Internet Gateway (IGW) – Internet access for public subnets**
4. **NAT Gateway – Outbound internet access for ECS tasks in private subnets**
5. **Route Tables – Public, private, and database routing**
6. **Amazon Route 53 – DNS management and domain mapping**

Load Balancing & Security

7. **Application Load Balancer (ALB) – Traffic routing and host-based routing**
8. **AWS Certificate Manager (ACM) – SSL/TLS certificates**
9. **Security Groups – Least-privilege network access control**

Containers & Orchestration

10. **Amazon ECS (Fargate) – Container orchestration without EC2 management**

11. **Amazon ECR – Docker image repository**

Database & Storage

12. **Amazon RDS (MySQL) – Managed relational database for WordPress**

13. **RDS Subnet Group – Database isolation in private subnets**

Security & Identity

14. **AWS IAM – Roles and policies for ECS task execution**

15. **AWS Secrets Manager – Secure storage of database credentials**

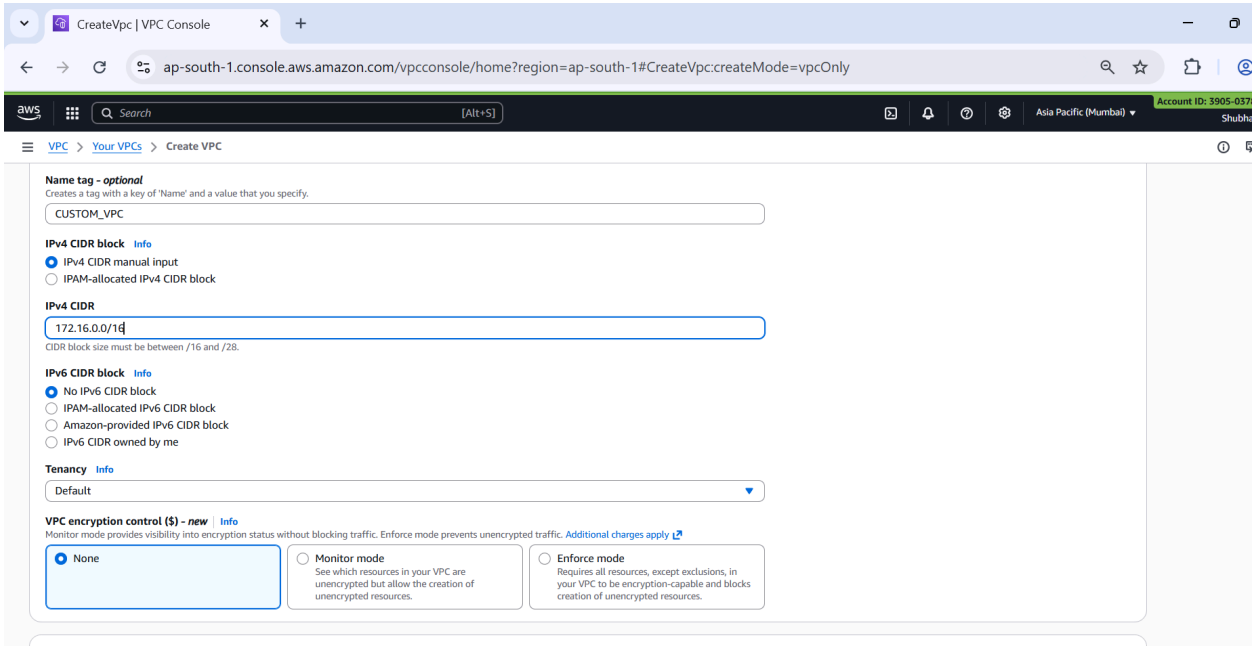
Monitoring & Logging

16. **Amazon CloudWatch – Logs and metrics for ECS tasks and ALB**

Automation & Scaling

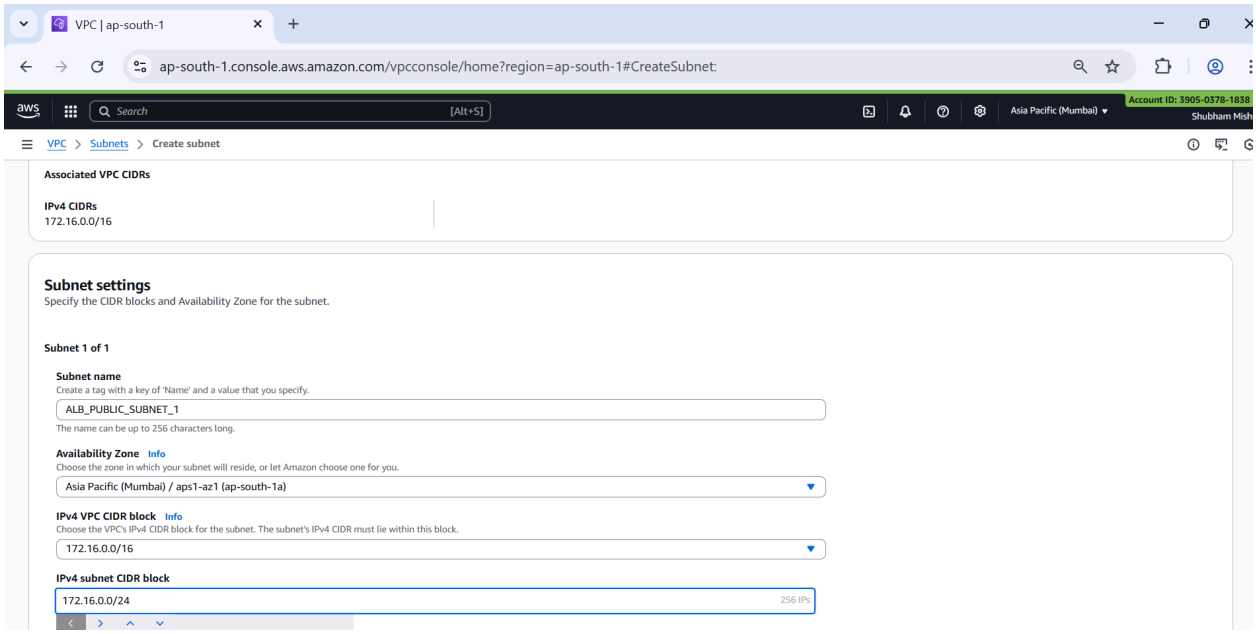
17. ECS Service Auto Scaling – CPU and memory-based auto scaling

Step 1) We will create Vpc with cdir block 172.16.0.0/16

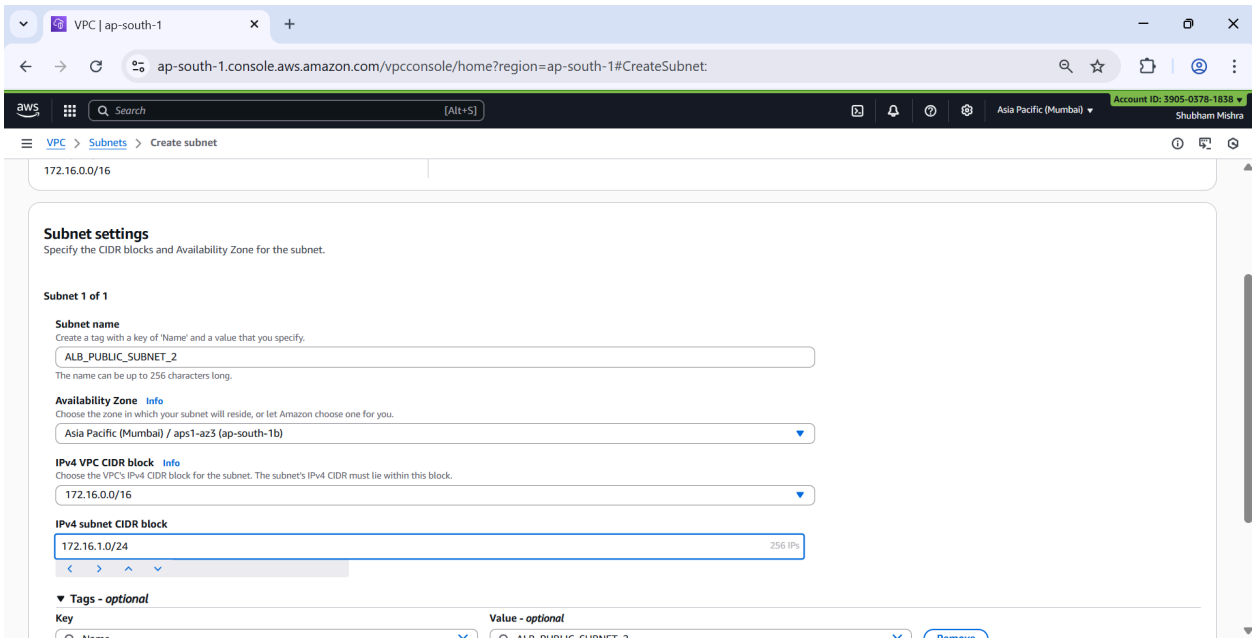


Step 2) We will create subnets for Load balancer, App(ECS clusters), DB subnets.

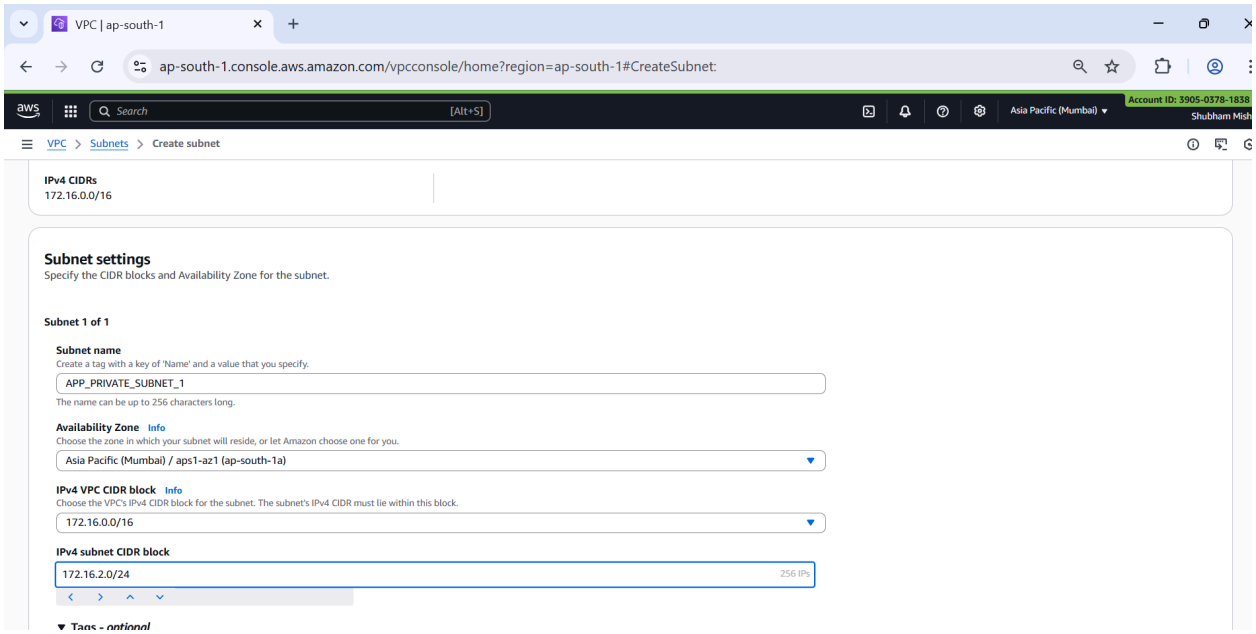
a) Will create Alb private subnet 1 with cidr block 172.16.0.0/24



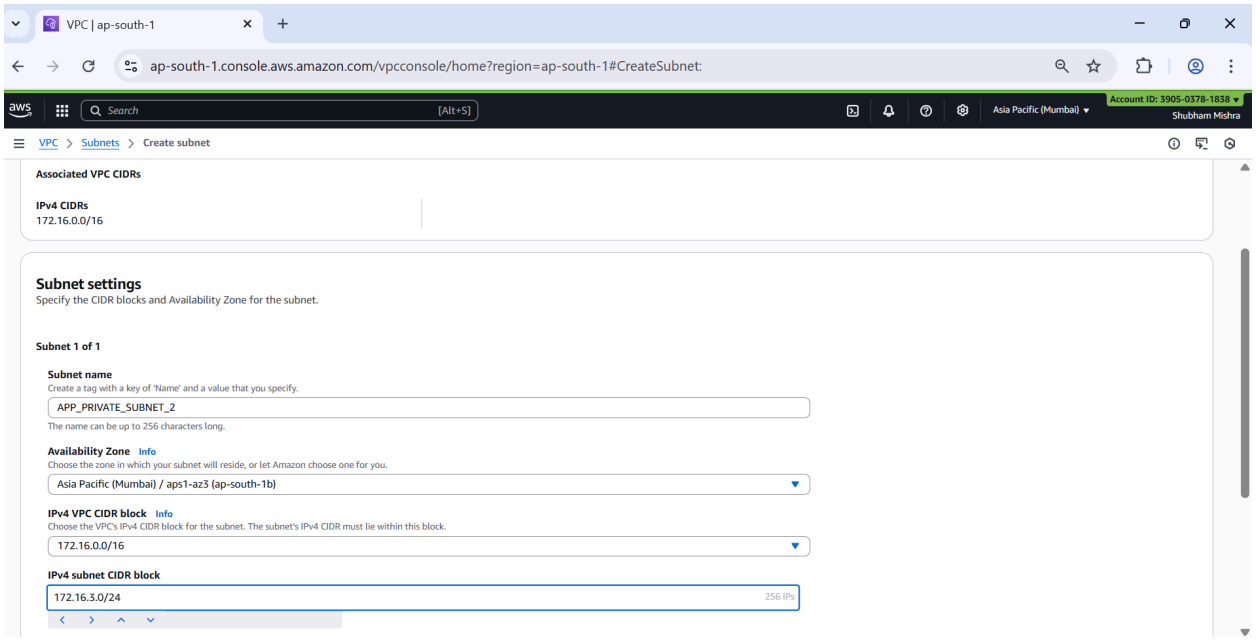
b) We will create alb public subnet 2 with cidr block 172.16.1.0/24



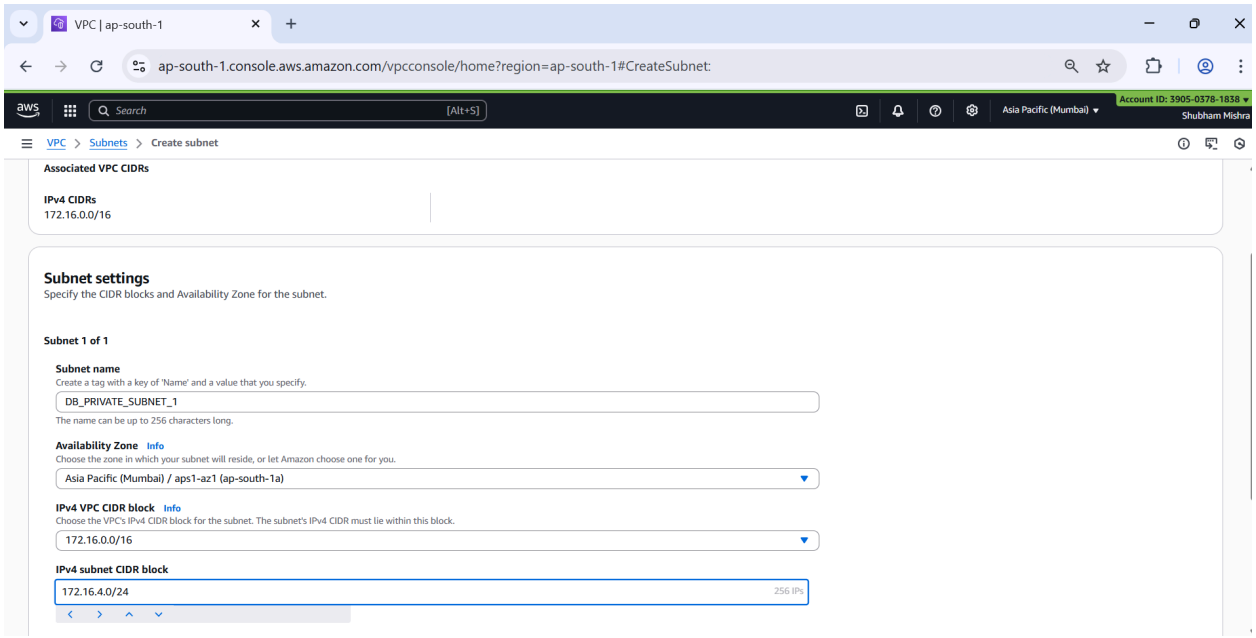
c) We will create App private subnet 1 with cidr block 172.16.2.0/24



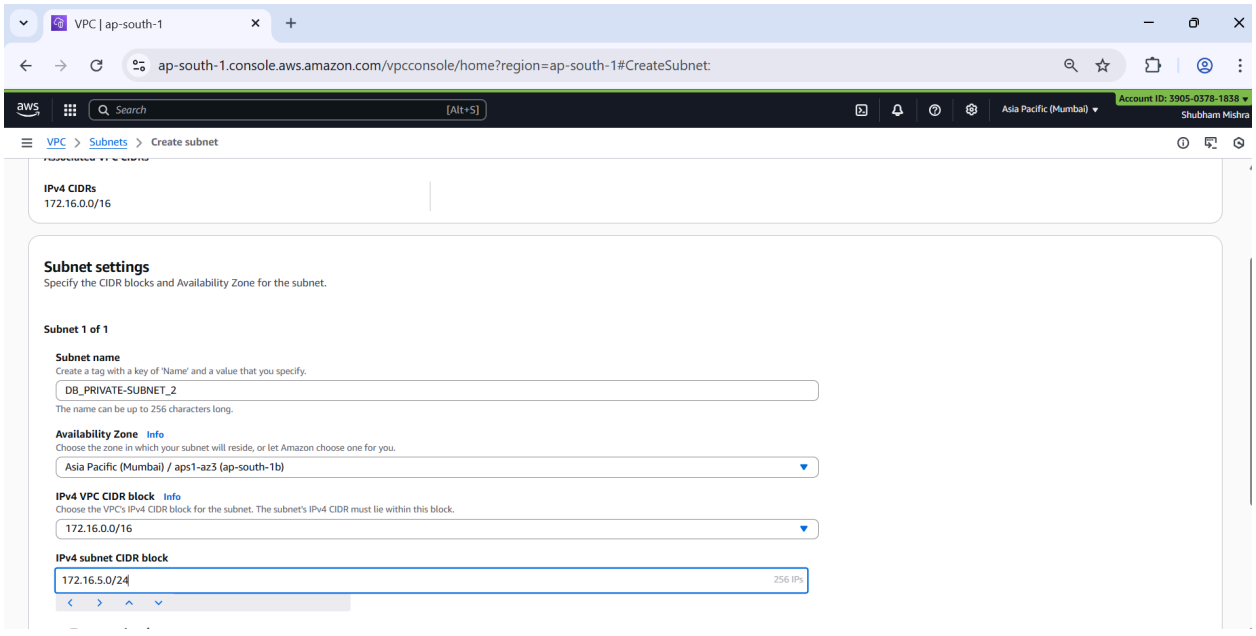
d) We will create app private subnet 2 with cidr block 172.16.3.0/24



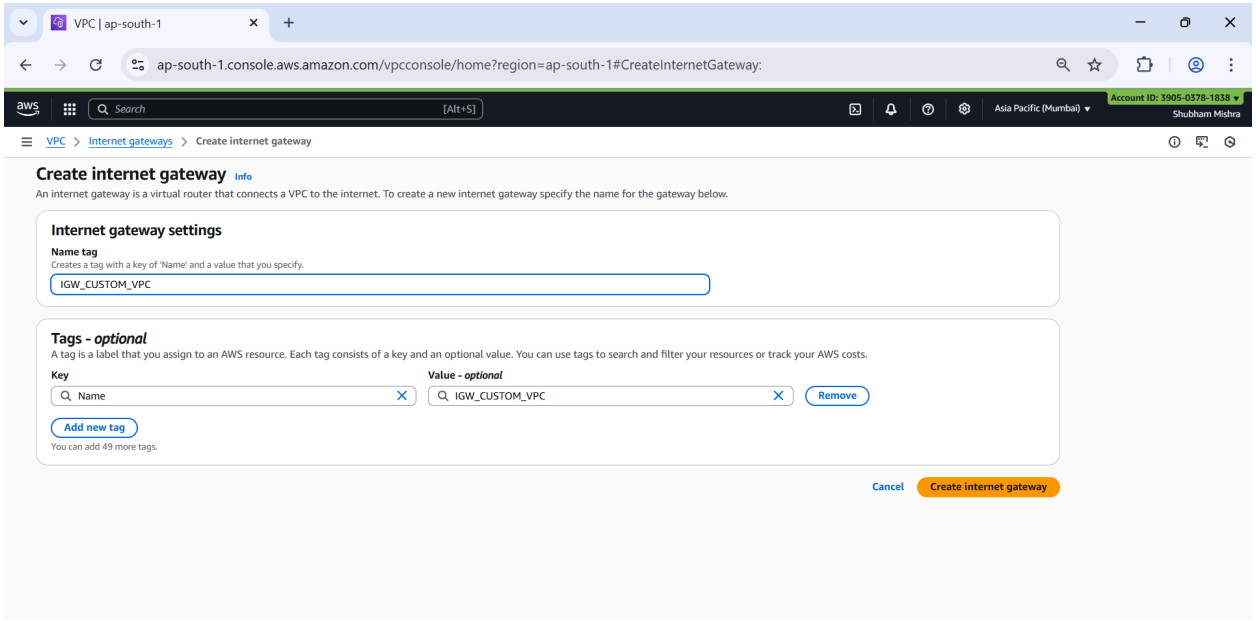
e) We will create now DB subnet 1 fro database with cidr block 172.16.4.0/24



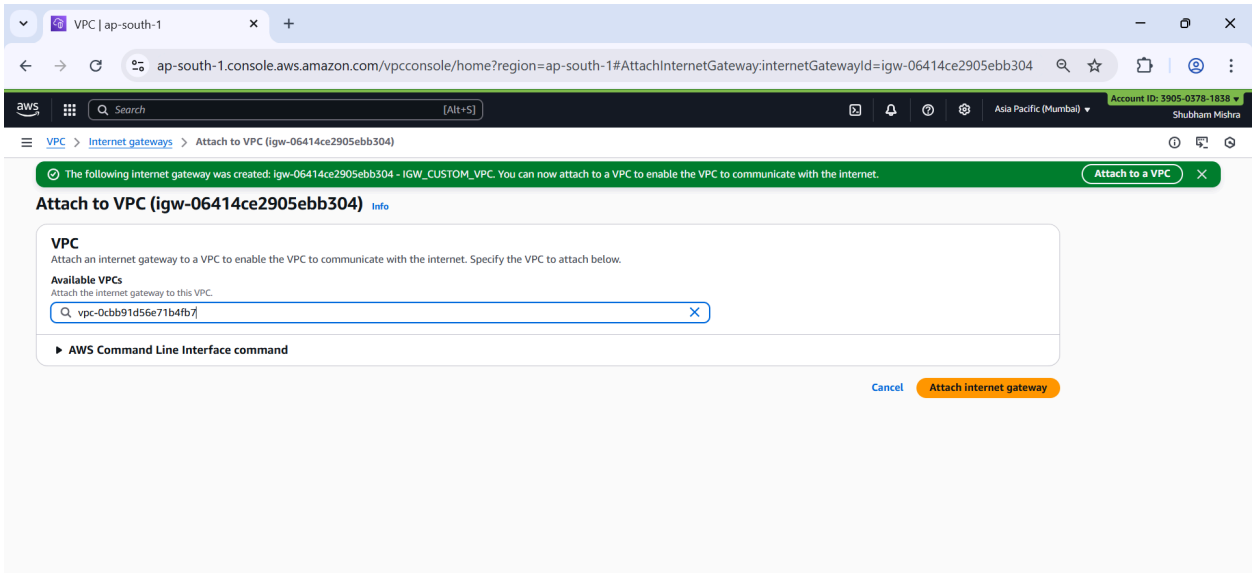
f) We will create DB subnet 2 with cidr block 172.16.5.0/24



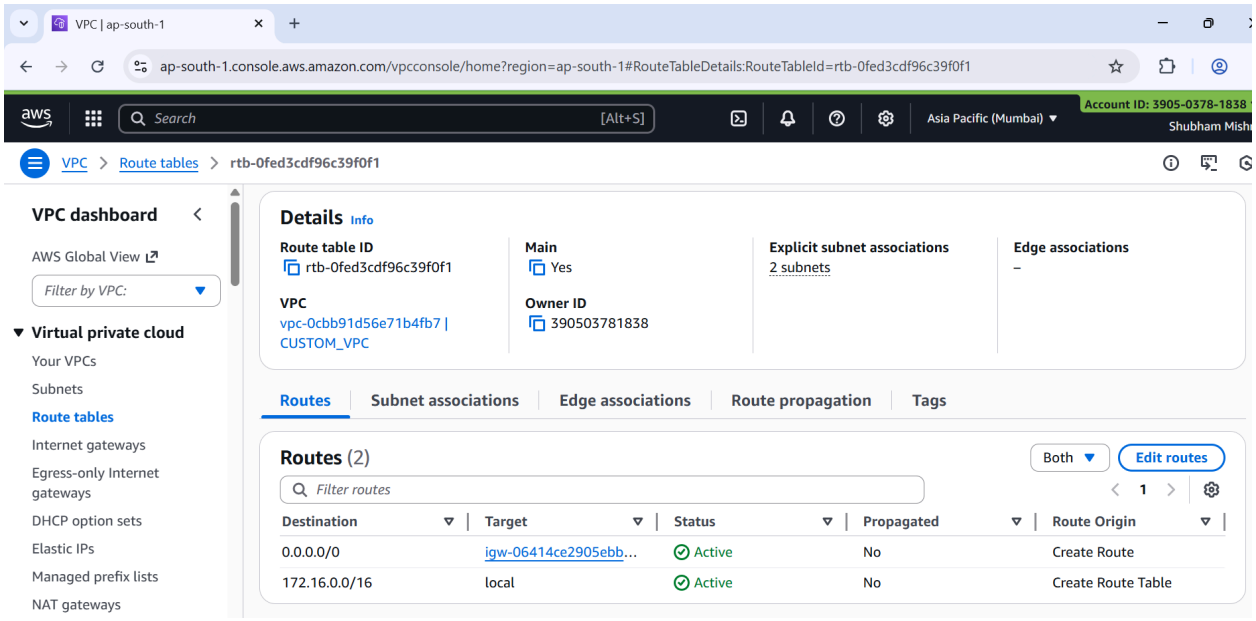
Step 3) We will create internet gateway and attach it to Vpc to provide internet connectivity.



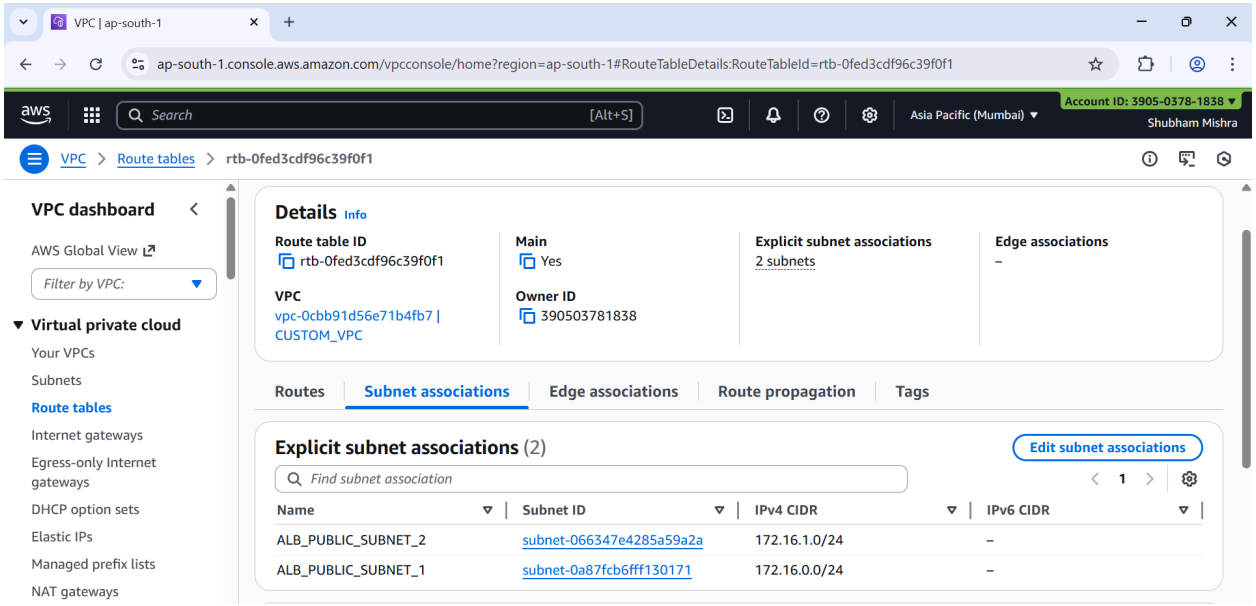
Step 4) We will now attach the Internet gateway to the vpc



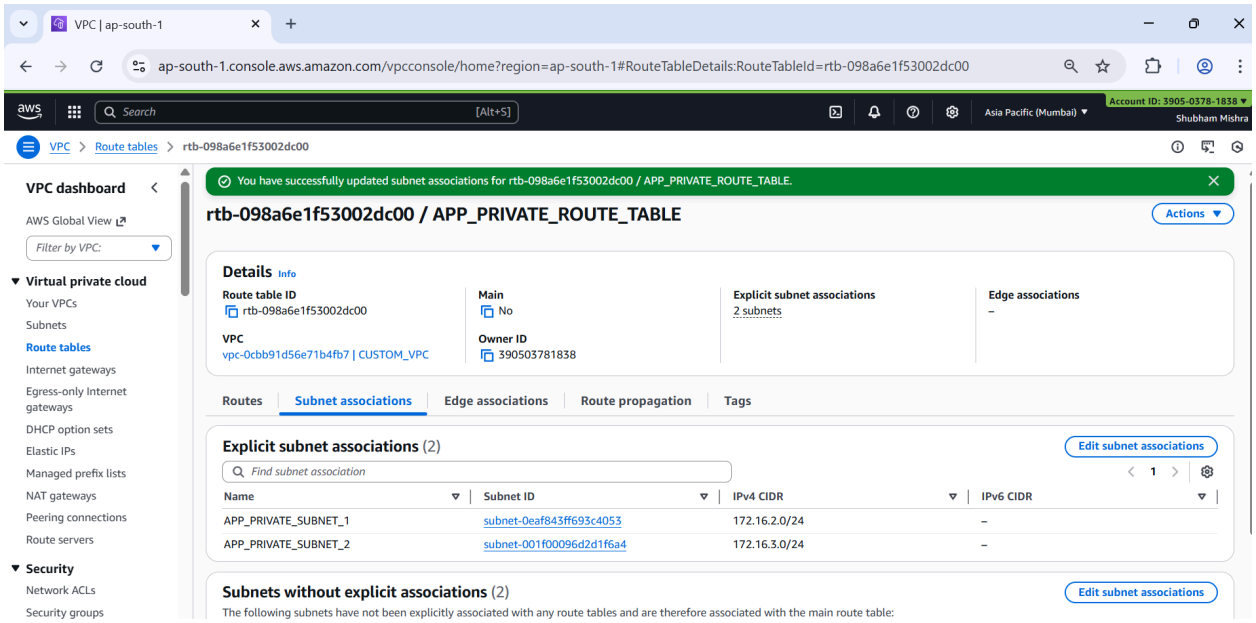
Step 5) Now weill add route to internet in alb route table to make alb subnets as public subnets.



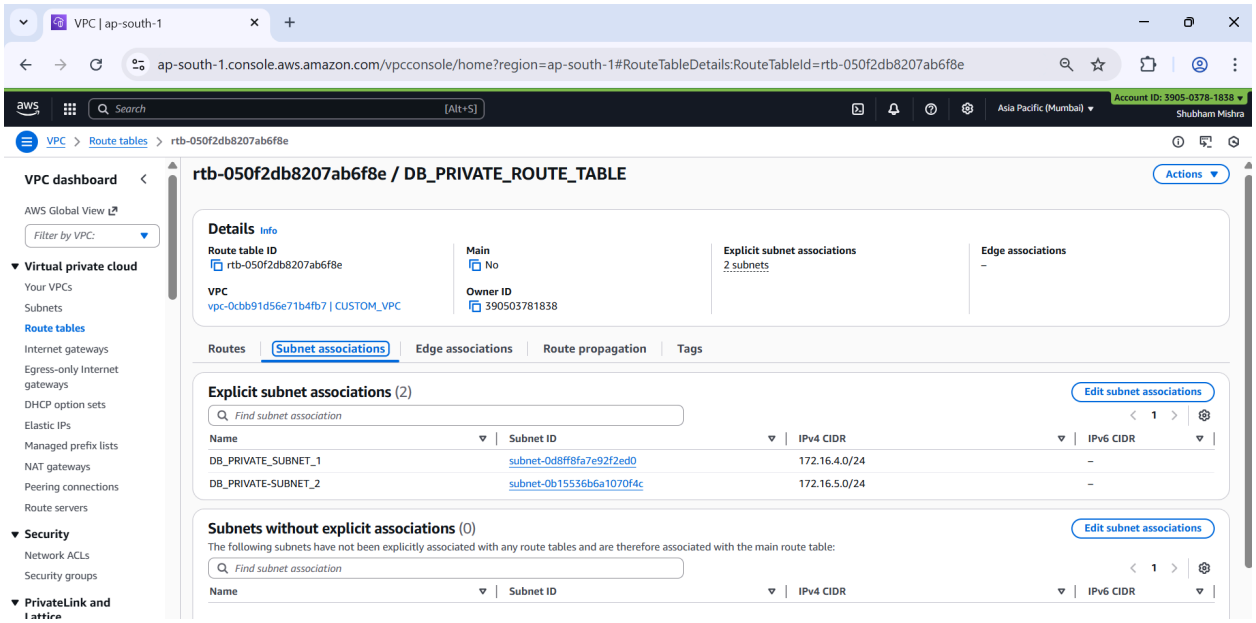
Step 6) We now attach the alb subnets to this route table created above.



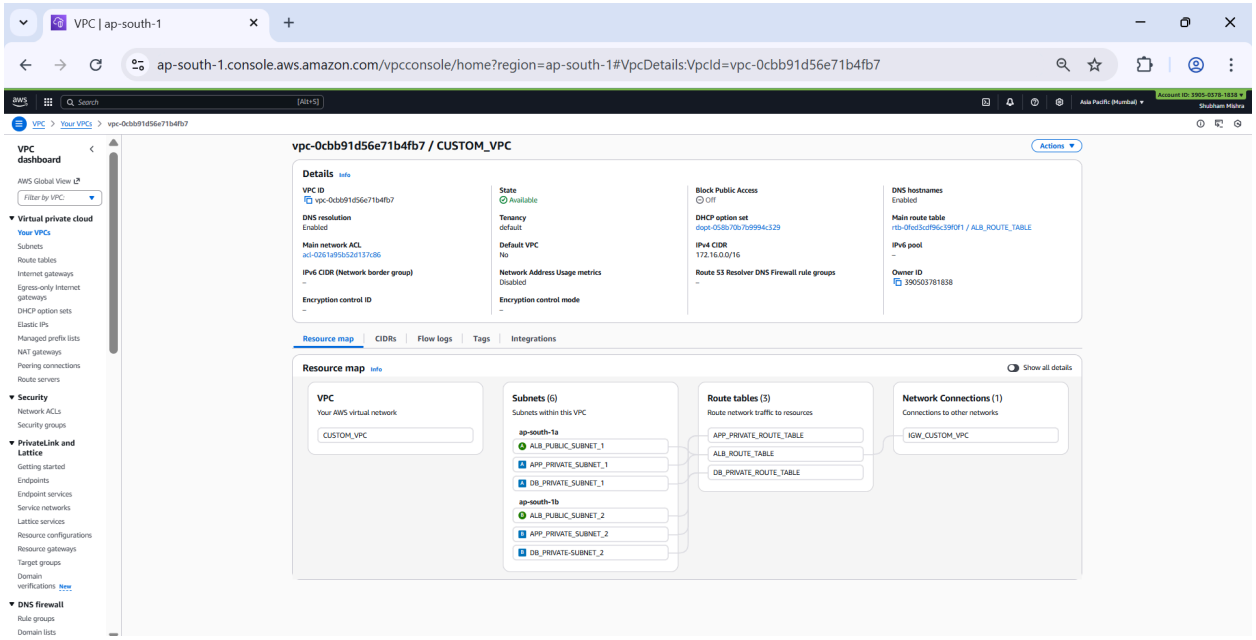
Step 7) We will now create private route table for app subnets and associate app/ecs subnets to this route table.



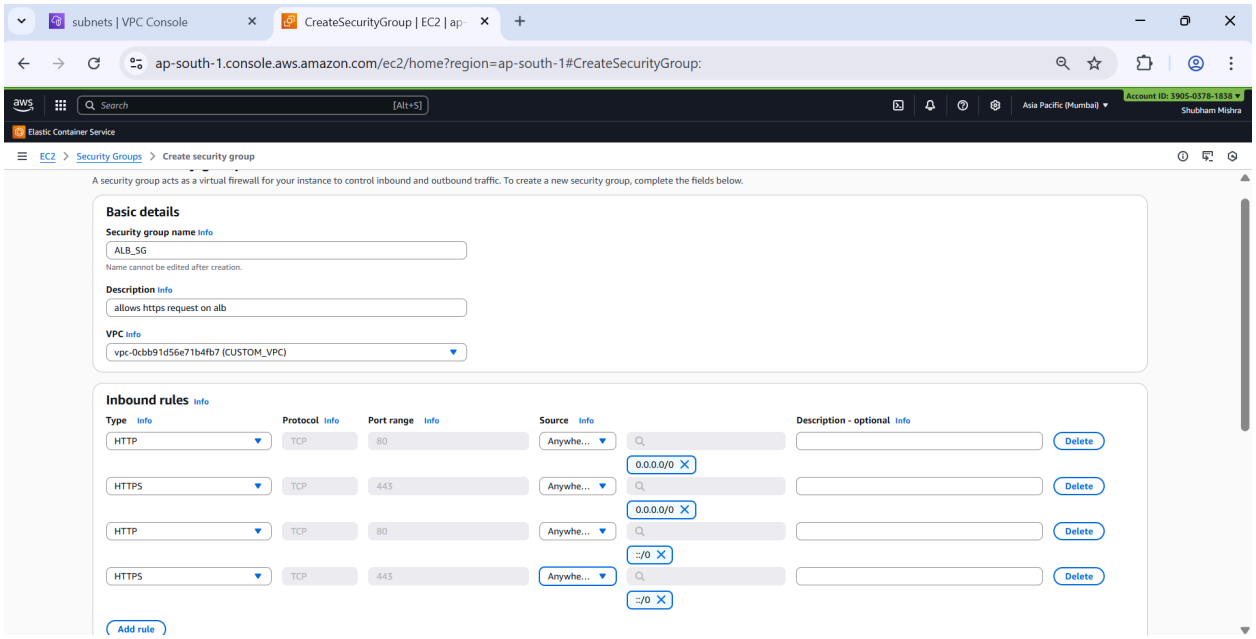
Step 8) We will now create database route table and associate it with db subnets.



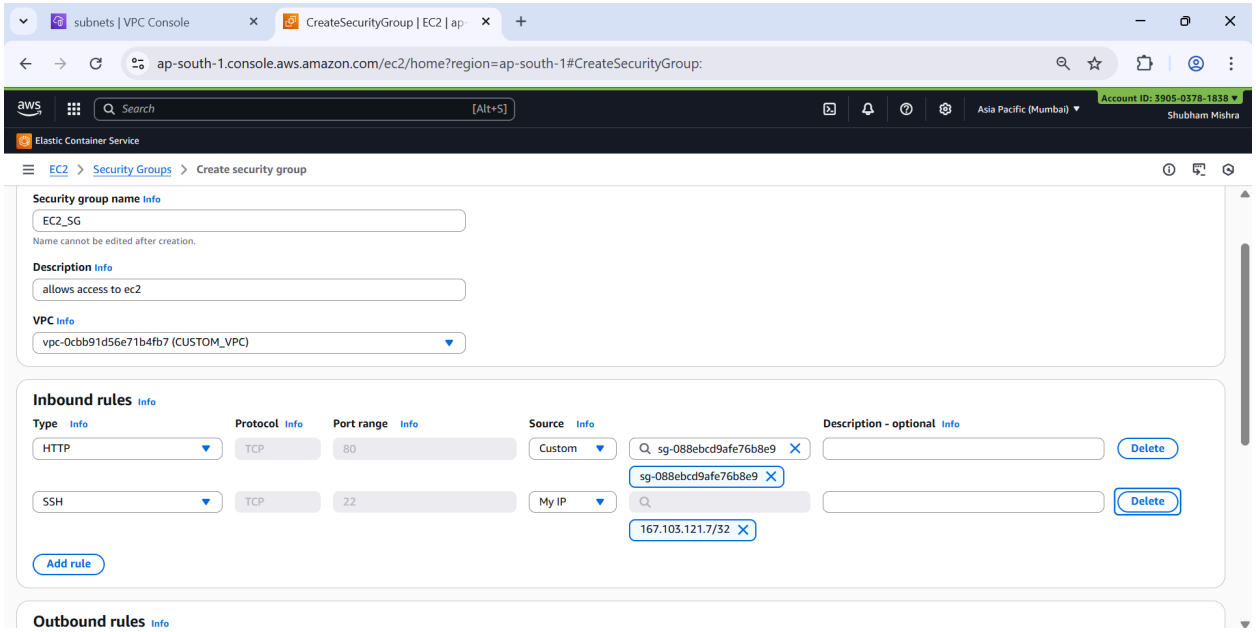
Step 9) We can now verify all the resource created in vpc resource map .



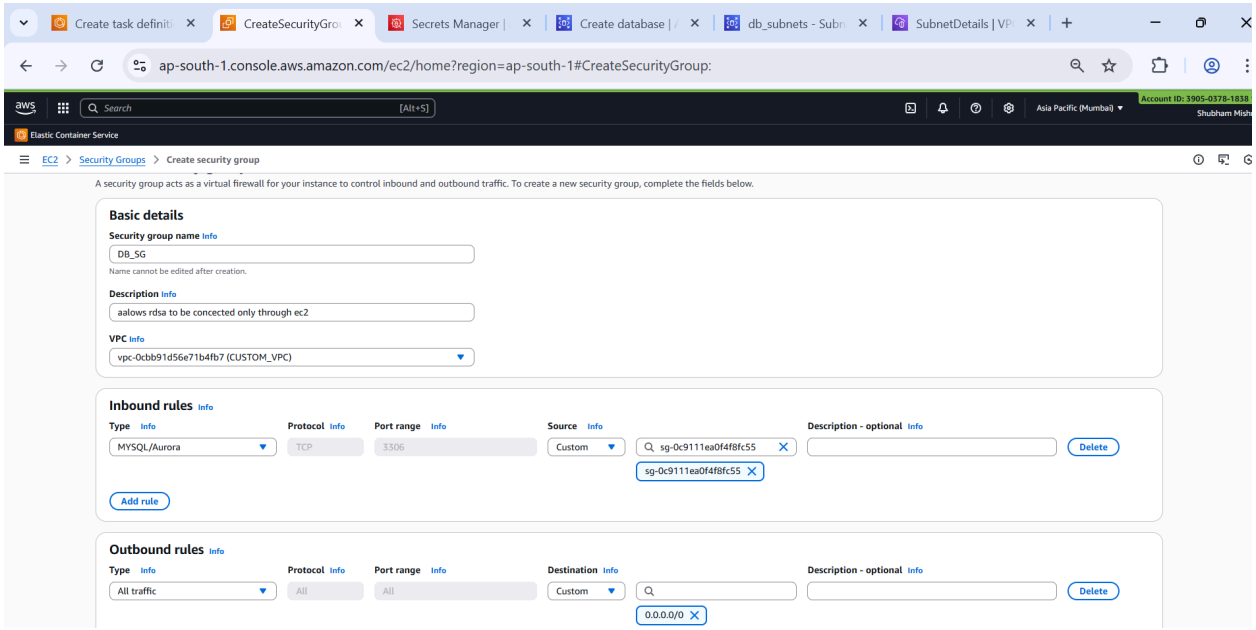
Step 10) We will create now security group for load balancer.



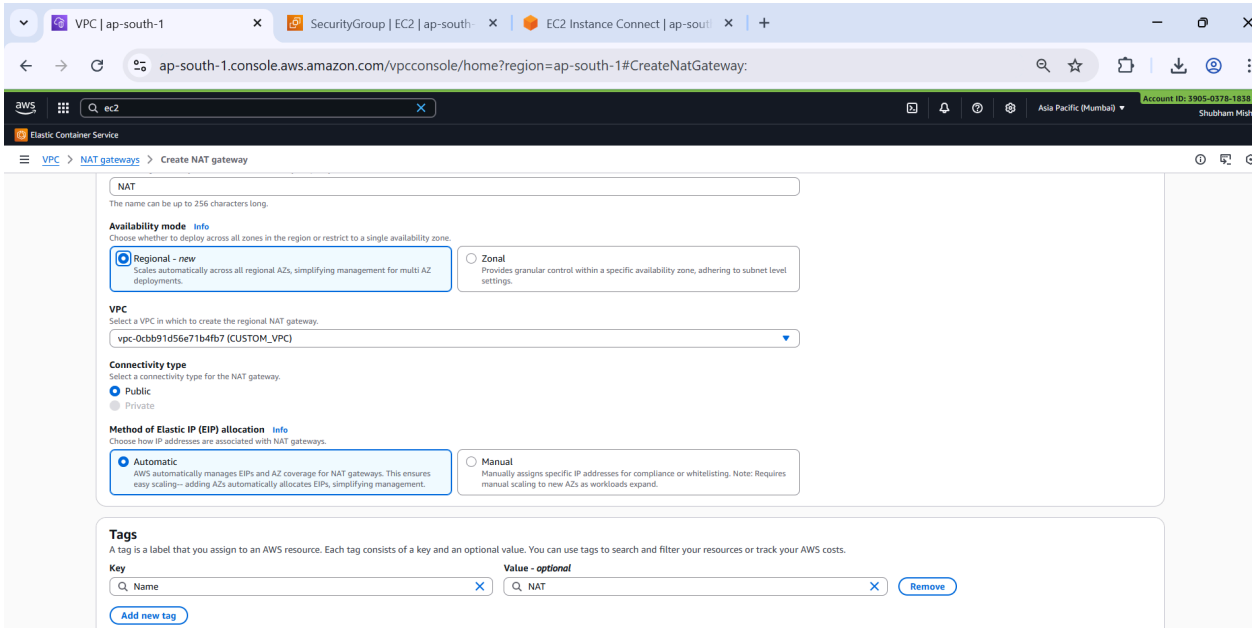
Step 11) We will create security group for ec2/ecs. Here we will allow http traffic from only loadbalancer sg and ssh only from my ip.



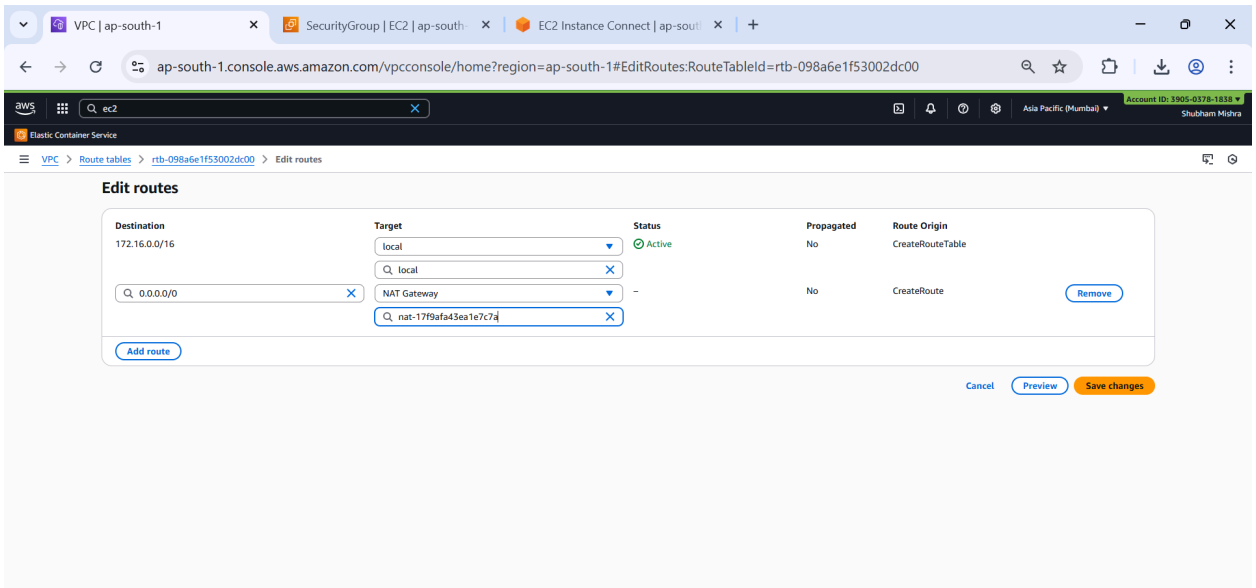
Step 12) We will create sg from database and allow port 3306 only from ec2/ecs security group.



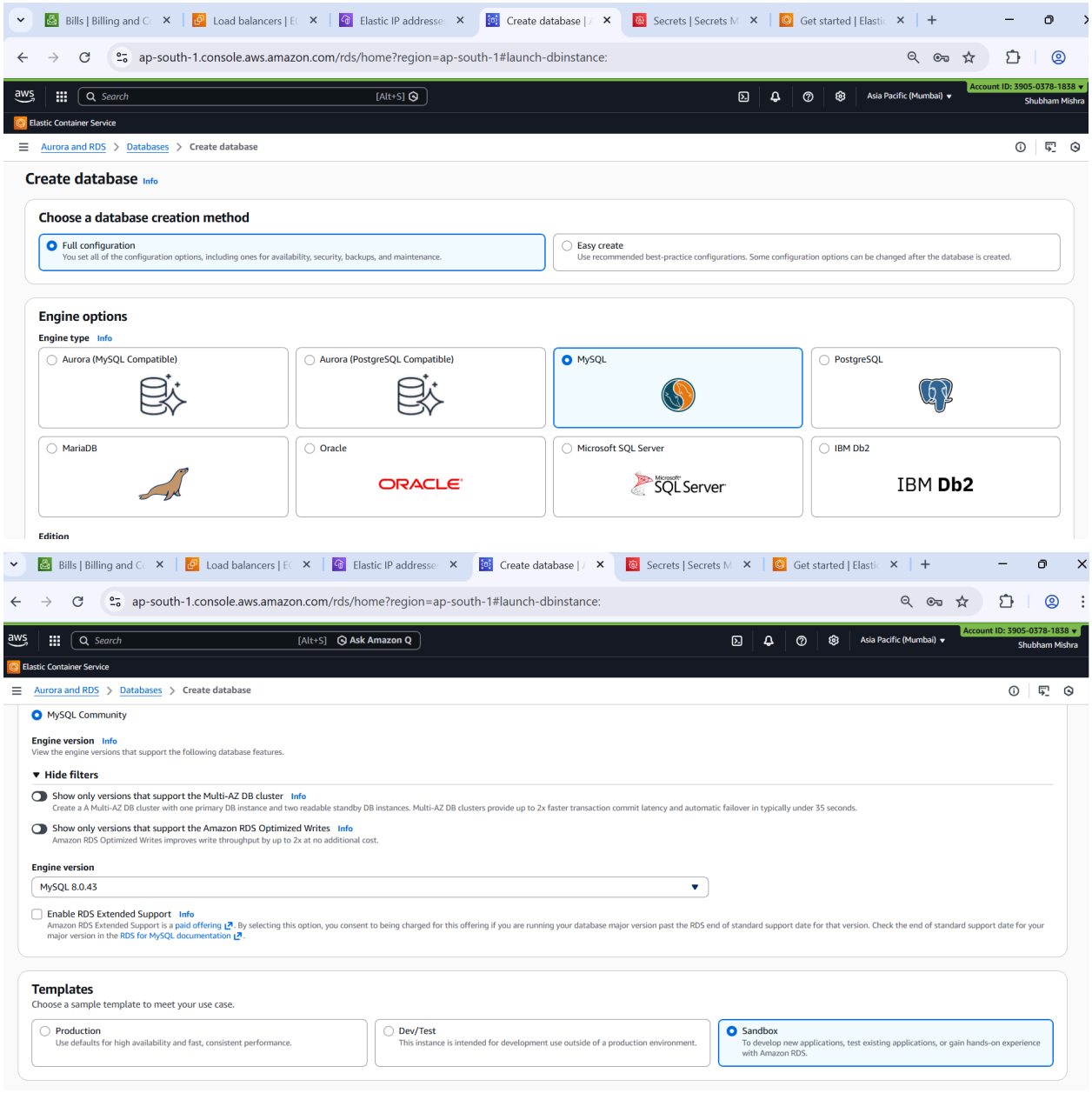
Step 13) Also we will create NAT gateway to provide outbound internet access for resources in private subnet for ec2/ecs



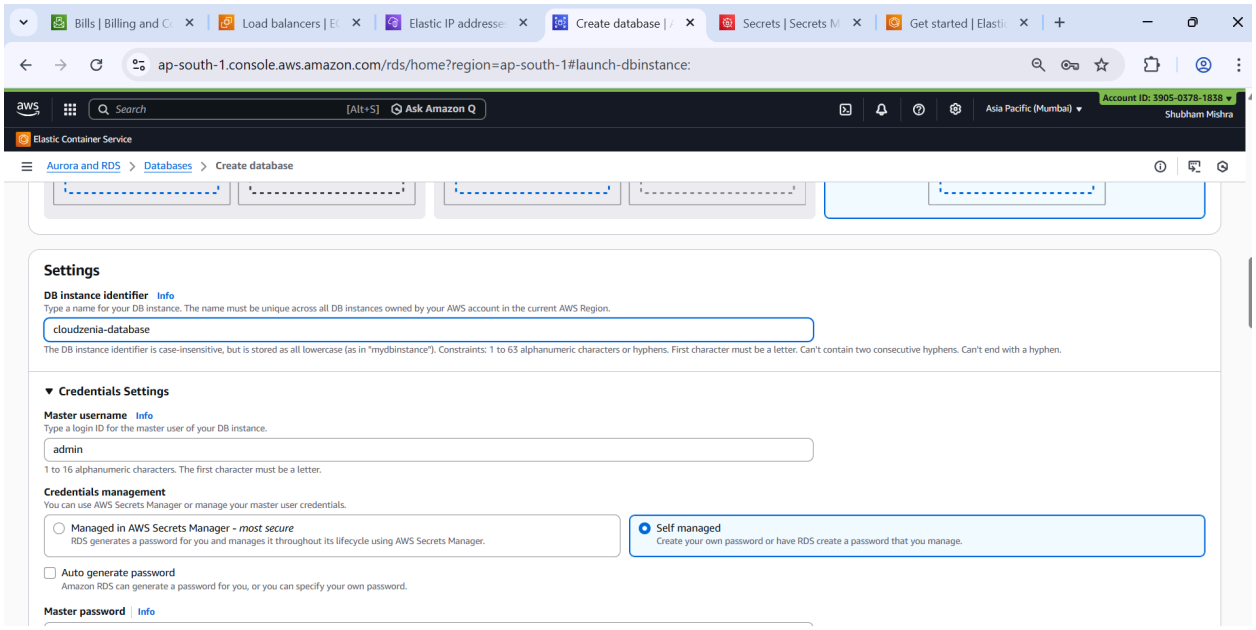
Step 14) We will add route to internet using nat gateway in private route table of app/ecs subnets.

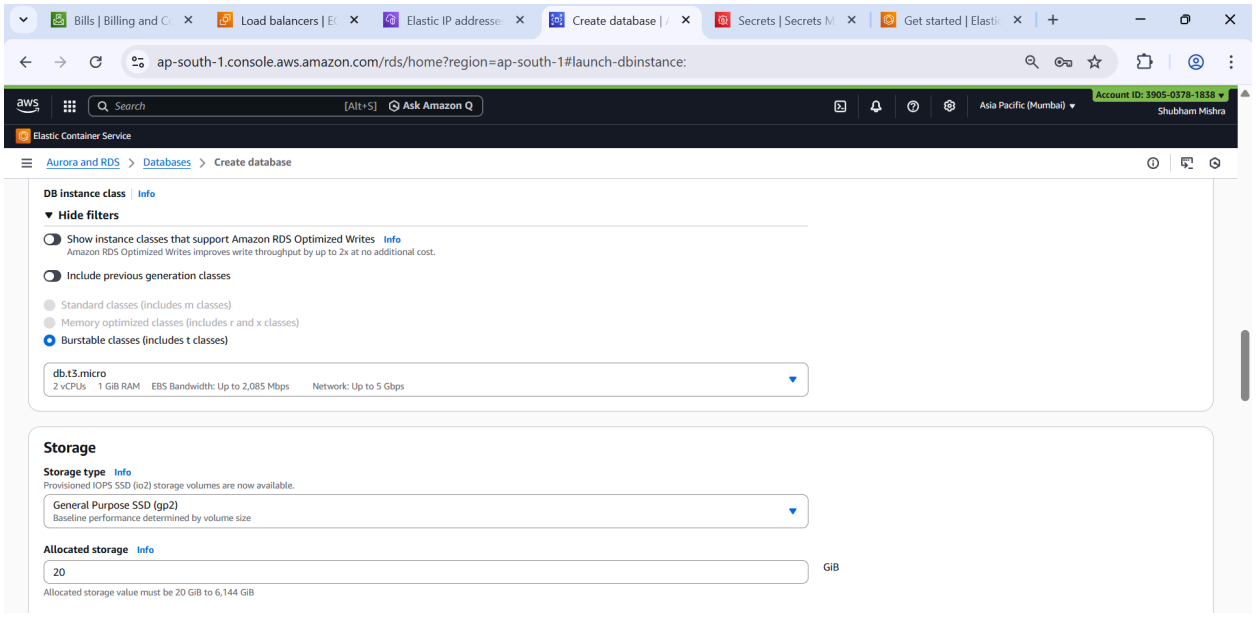


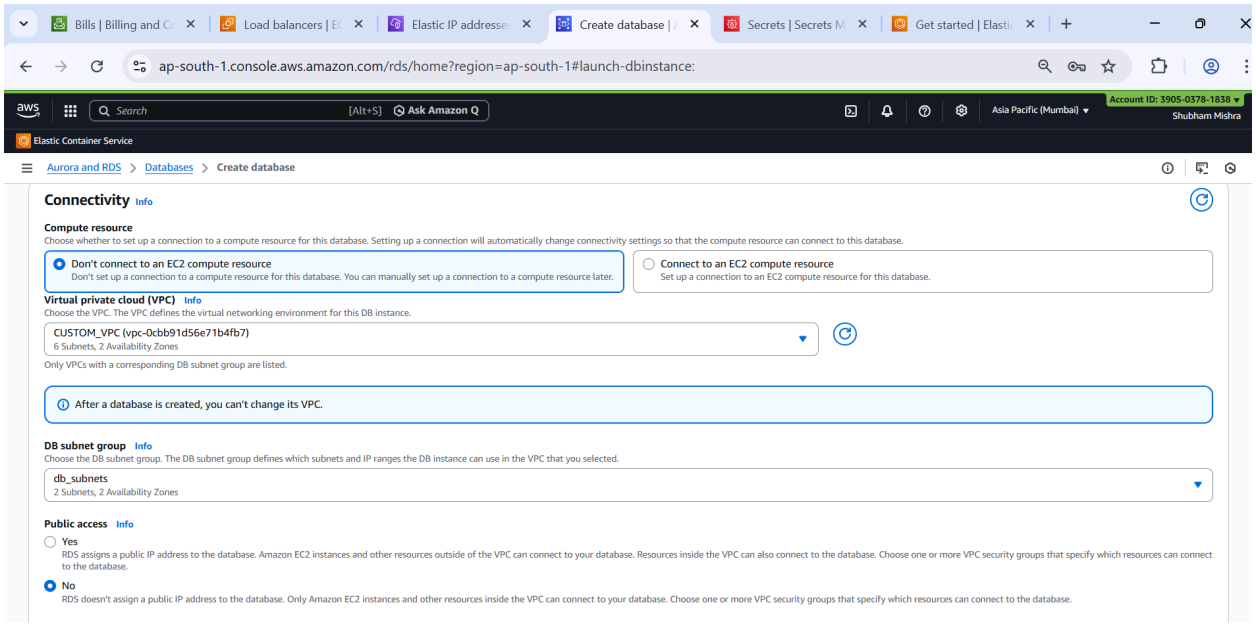
Step 15) Now we will create RDS database

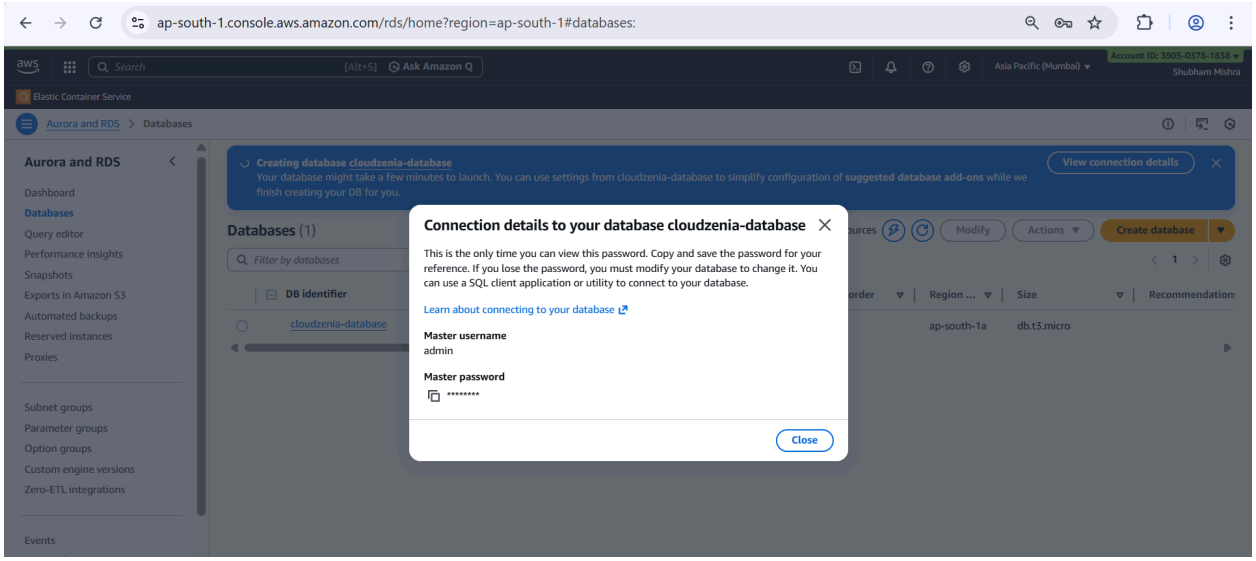

Bills | Billing and C
Load balancers | E
Elastic IP addresse
Create database |
Secrets | Secrets M
Get started | Elasti
ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#launch-dbinstance:
aws
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
Aurora and RDS
Databases
Create database
Create database Info
Choose a database creation method
Full configuration
You set all of the configuration options, including ones for availability, security, backups, and maintenance.
Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.
Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
PostgreSQL
MariaDB
Oracle
ORACLE
Microsoft SQL Server
SQL Server
IBM Db2
IBM Db2
Edition

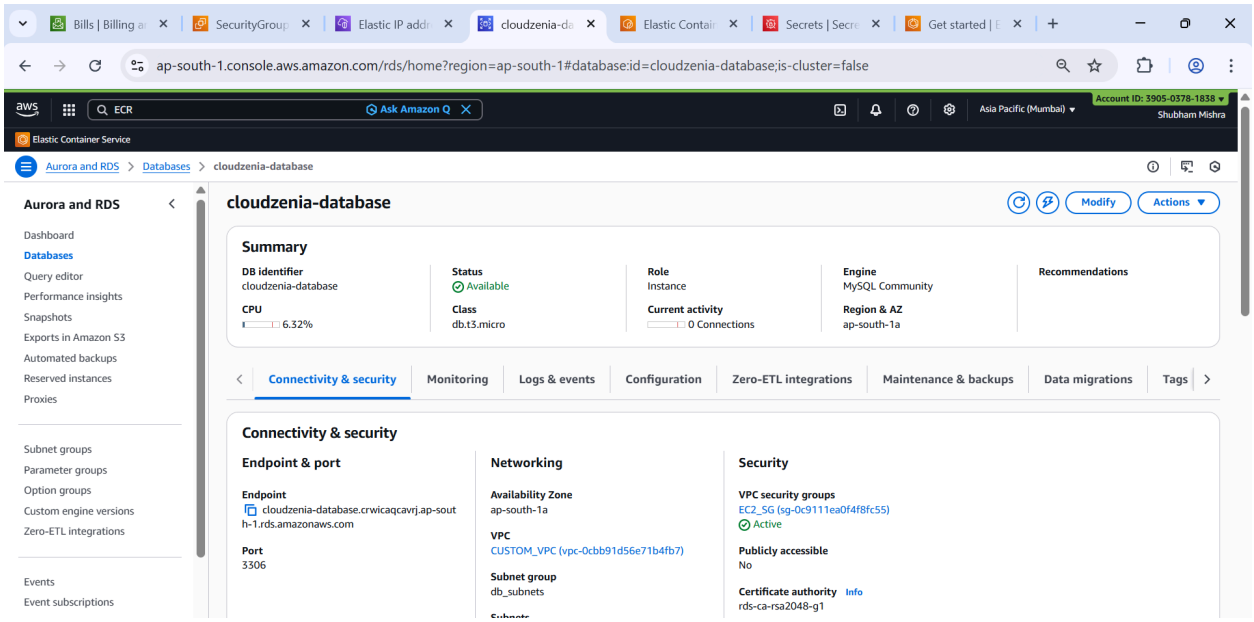
Bills | Billing and C
Load balancers | E
Elastic IP addresse
Create database |
Secrets | Secrets M
Get started | Elasti
ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#launch-dbinstance:
aws
Search
[Alt+S]
Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
Aurora and RDS
Databases
Create database
MySQL Community
Engine version Info
View the engine versions that support the following database features.
Hide filters
Show only versions that support the Multi-AZ DB cluster Info
Create a A Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.
Show only versions that support the Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.
Engine version
MySQL 8.0.43
Enable RDS Extended Support Info
Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the RDS for MySQL documentation.
Templates
Choose a sample template to meet your use case.
Production
Use defaults for high availability and fast, consistent performance.
Dev/Test
This instance is intended for development use outside of a production environment.
Sandbox
To develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.


Bills | Billing and C
Load balancers | E
Elastic IP addresse
Create database |
Secrets | Secrets M
Get started | Elasti
ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#launch-dbinstance:
aws
Search
[Alt+S]
Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
Aurora and RDS
Databases
Create database
Settings
DB instance identifier Info
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
cloudzenia-database
The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.
Credentials Settings
Master username Info
Type a login ID for the master user of your DB instance.
admin
1 to 16 alphanumeric characters. The first character must be a letter.
Credentials management
You can use AWS Secrets Manager or manage your master user credentials.
Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.
Self managed
Create your own password or have RDS create a password that you manage.
Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.
Master password Info

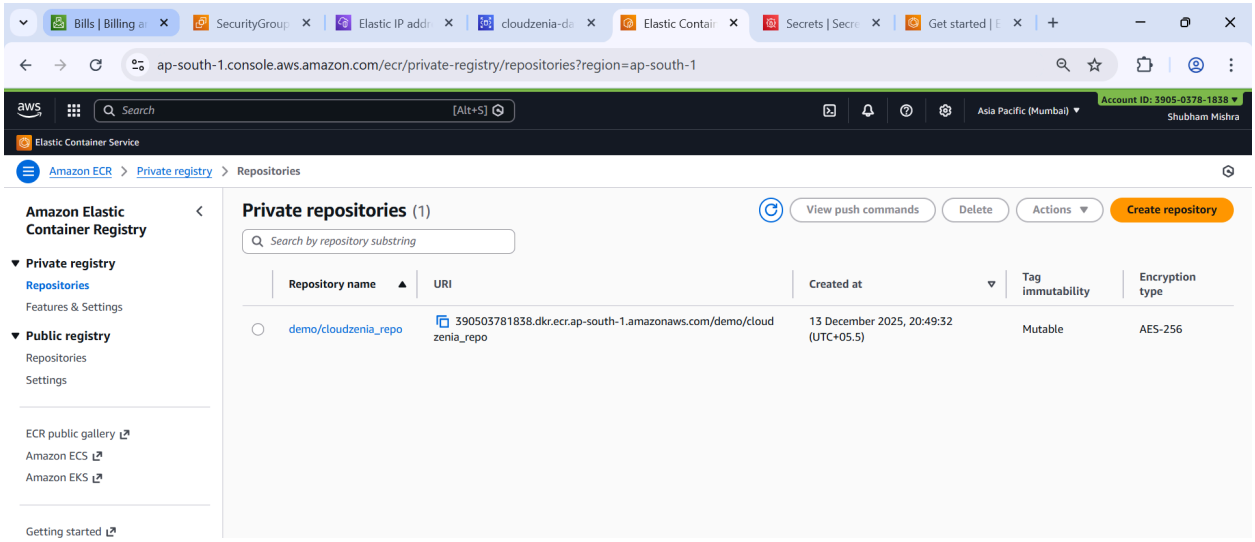

Bills | Billing and C
Load balancers | E
Elastic IP addresse
Create database |
Secrets | Secrets M
Get started | Elasti
ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#launch-dbinstance:
aws
Search
[Alt+S]
Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
Aurora and RDS
Databases
Create database
DB instance class Info
Hide filters
Show instance classes that support Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.
Include previous generation classes
Standard classes (includes m classes)
Memory optimized classes (includes r and x classes)
Burstable classes (includes t classes)
db.t3.micro
2 vCPUs 1 GiB RAM EBS Bandwidth: Up to 2,085 Mbps Network: Up to 5 Gbps
Storage
Storage type Info
Provisioned IOPS SSD (io2) storage volumes are now available.
General Purpose SSD (gp2)
Baseline performance determined by volume size
Allocated storage Info
20
GiB
Allocated storage value must be 20 GiB to 6,144 GiB


Bills | Billing and C
Load balancers | E
Elastic IP addresse
Create database |
Secrets | Secrets M
Get started | Elasti
ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#launch-dbinstance:
Search
[Alt+S]
Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
Aurora and RDS
Databases
Create database
Connectivity Info
Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.
Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.
Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.
Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.
CUSTOM_VPC (vpc-0cbb91d56e71b4fb7)
6 Subnets, 2 Availability Zones
Only VPCs with a corresponding DB subnet group are listed.
After a database is created, you can't change its VPC.
DB subnet group Info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.
db_subnets
2 Subnets, 2 Availability Zones
Public access Info
Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.
No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.


ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#databases:
Aurora and RDS
Dashboard
Databases
Query editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations
Events
Creating database cloudzenia-database
Your database might take a few minutes to launch. You can use settings from cloudzenia-database to simplify configuration of suggested database add-ons while we finish creating your DB for you.
View connection details
Databases (1)
Filter by databases
DB identifier
cloudzenia-database
Modify
Actions
Create database
Region ...
Size
Recommendation
ap-south-1a
db.t3.micro
Connection details to your database cloudzenia-database
This is the only time you can view this password. Copy and save the password for your reference. If you lose the password, you must modify your database to change it. You can use a SQL client application or utility to connect to your database.
Learn about connecting to your database
Master username
admin
Master password
****
Close

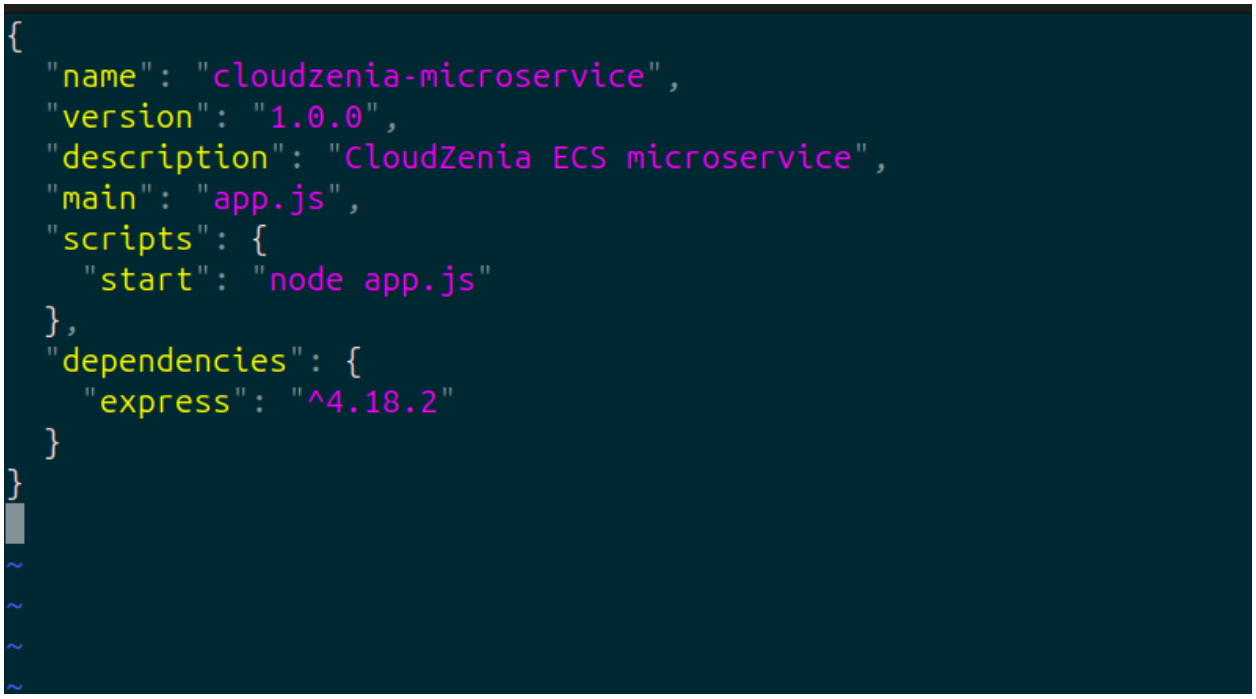


Step 16) Now we will create ECR repository to store images and pull images in ECS



Step 17) Now we will create files for docker image to store in ecr

a) package.json


{
"name": "cloudzenia-microservice",
"version": "1.0.0",
"description": "CloudZenia ECS microservice",
"main": "app.js",
"scripts": {
"start": "node app.js"
},
"dependencies": {
"express": "^4.18.2"
}
}

b) app.js

```
const express = require("express");
const app = express();

app.get("/", (req, res) => {
  res.send("Hello from Microservice");
});

app.get("/health", (req, res) => {
  res.status(200).send("OK");
});

const port = process.env.PORT || 3000;
app.listen(port, () => {
  console.log(`Microservice running on port ${port}`);
});
~
```

c) Dockerfile

```
# Use official lightweight Node image
FROM node:18-alpine

# Create app directory
WORKDIR /app

# Copy dependency files first (best practice)
COPY package.json ./

# Install dependencies
RUN npm install --only=production

# Copy application code
COPY app.js ./

# Expose application port
EXPOSE 3000

# Start the application
CMD ["npm", "start"]
```

```
root@shubham-Inspiron-14-3467:/home/shubham/microservice# ls
app.js  Dockerfile  package.json
root@shubham-Inspiron-14-3467:/home/shubham/microservice#
```

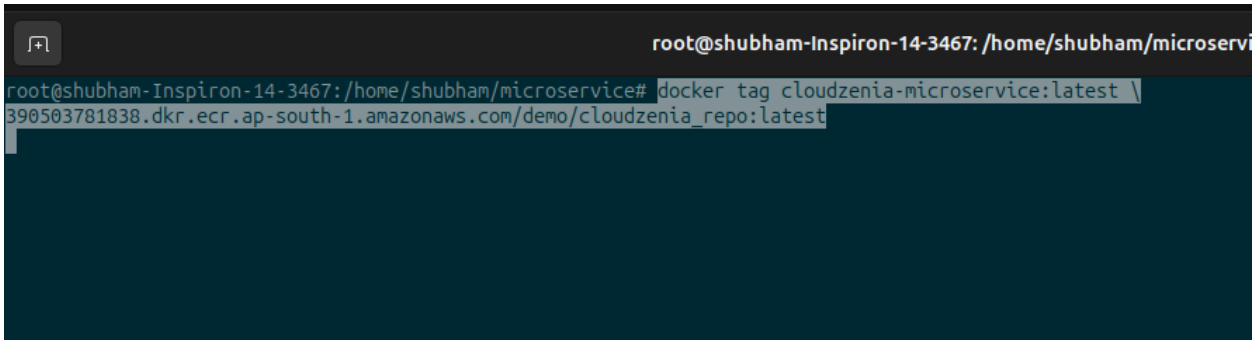
```
root@shubham-Inspiron-14-3467:/home/shubham/microservice# ls
app.js  Dockerfile  package.json
root@shubham-Inspiron-14-3467:/home/shubham/microservice# docker build -t cloudzenia-microservice .
```

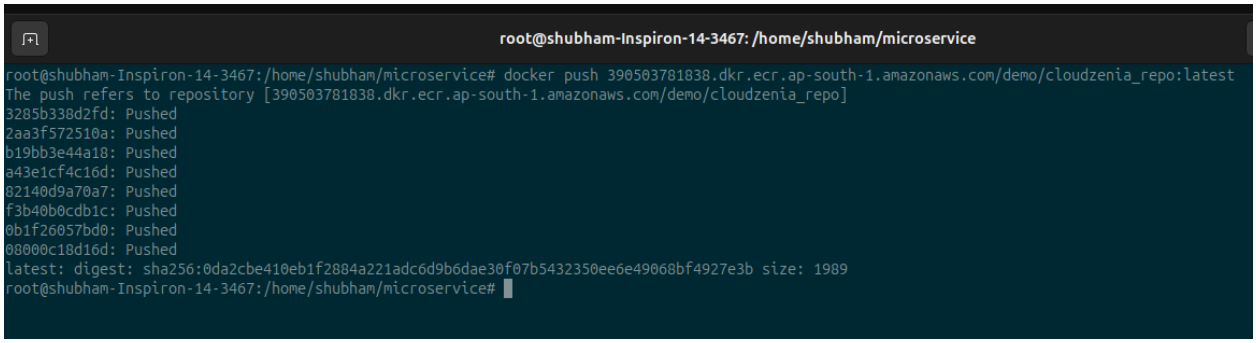
```
root@shubham-Inspiron-14-3467:/home/shubham/microservice# docker image ls
REPOSITORY               TAG         IMAGE ID       CREATED              SIZE
cloudzenia-microservice  latest      48b1d34b3eb9   About a minute ago   135MB
nginx                    latest      60adc2e137e7   3 weeks ago          152MB
node                     18-alpine   ee77c6cd7c18   8 months ago         127MB
root@shubham-Inspiron-14-3467:/home/shubham/microservice#
```

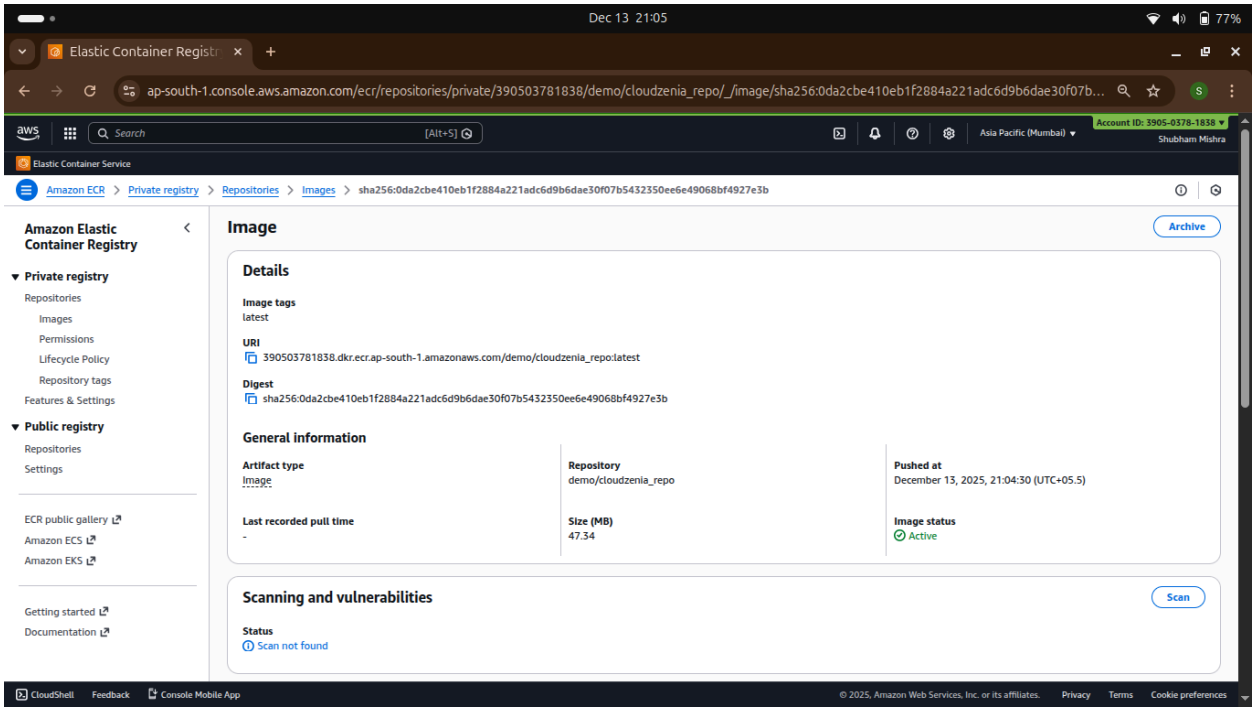
```
root@shubham-Inspiron-14-3467: /home/shubham/microservi
root@shubham-Inspiron-14-3467:/home/shubham/microservice# aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password-stdin \
390503781838.dkr.ecr.ap-south-1.amazonaws.com

WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
root@shubham-Inspiron-14-3467:/home/shubham/microservice#
```

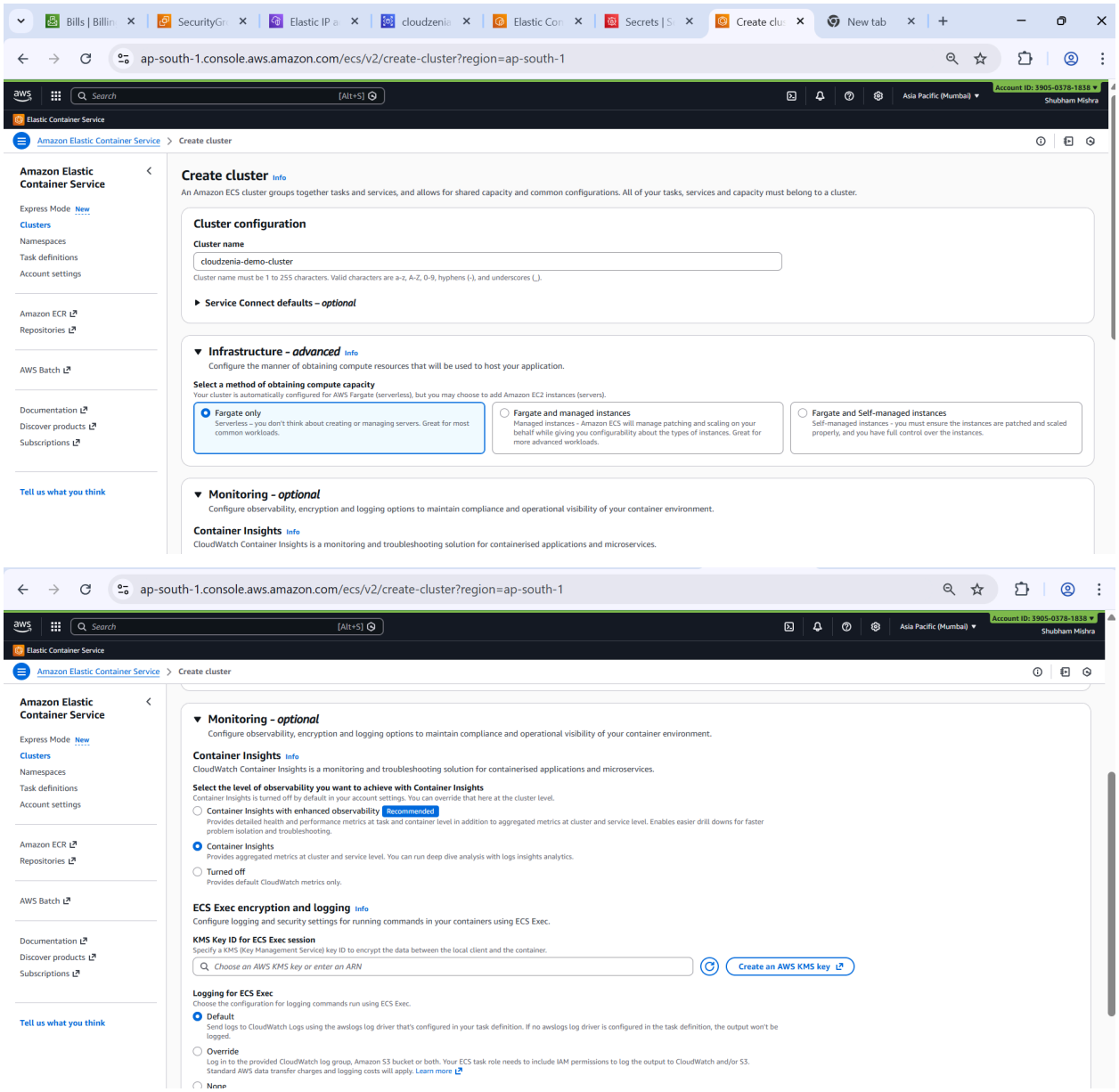




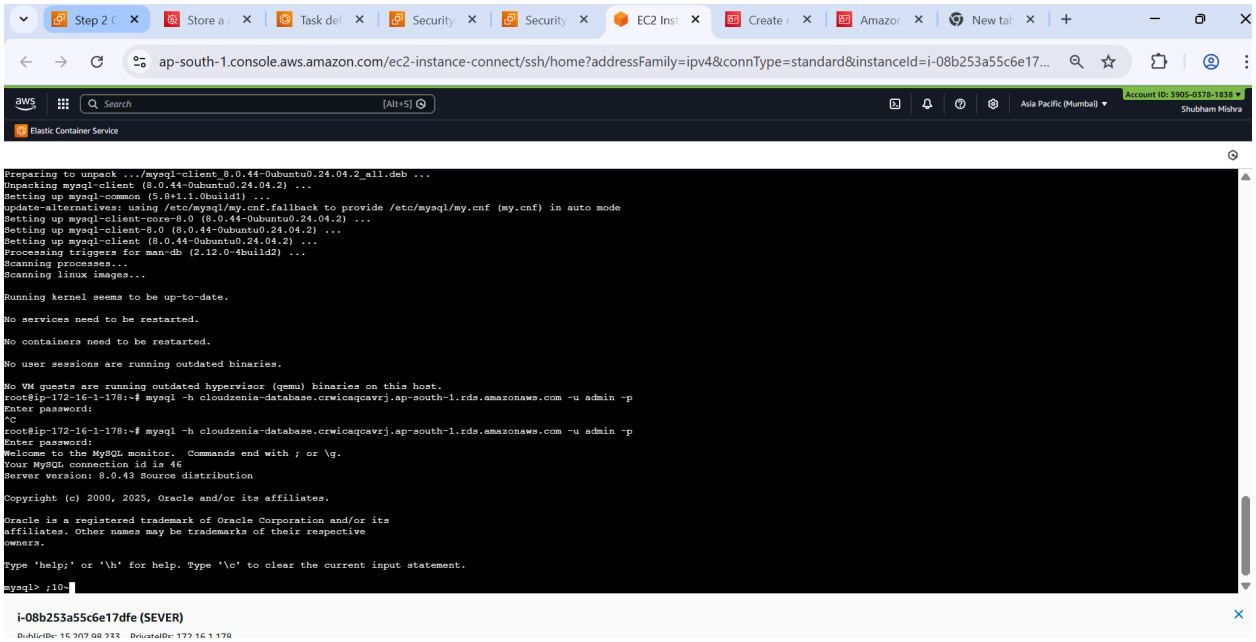


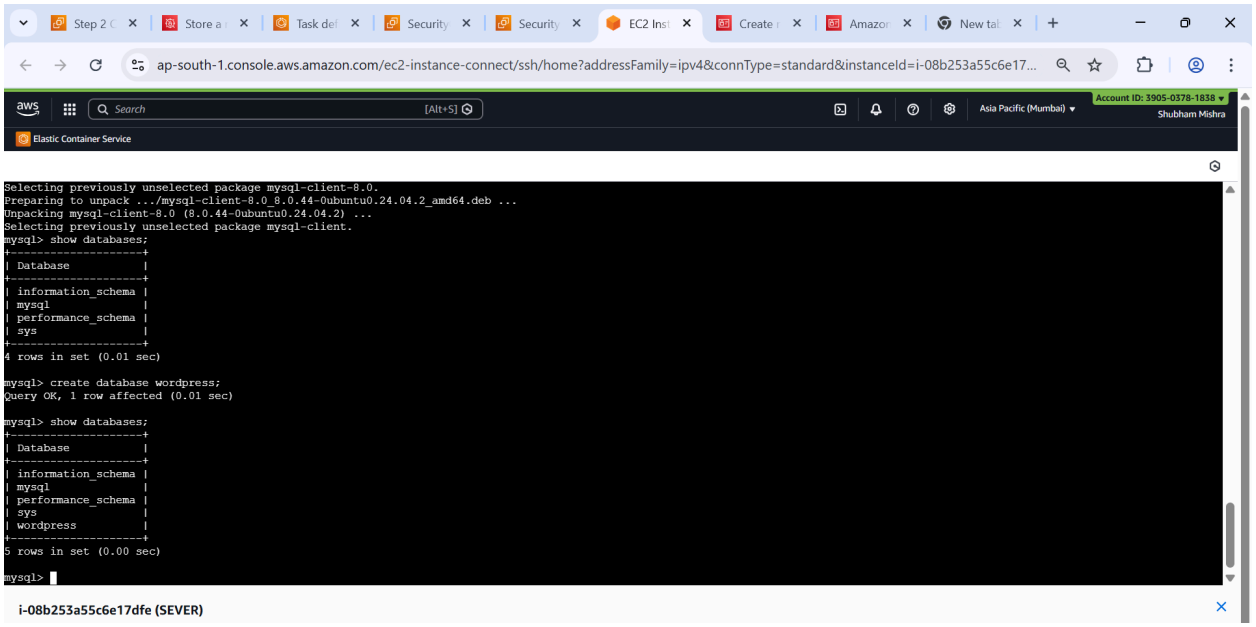
Microservice image in ECR

Step 18) Now we will create ECS cluster

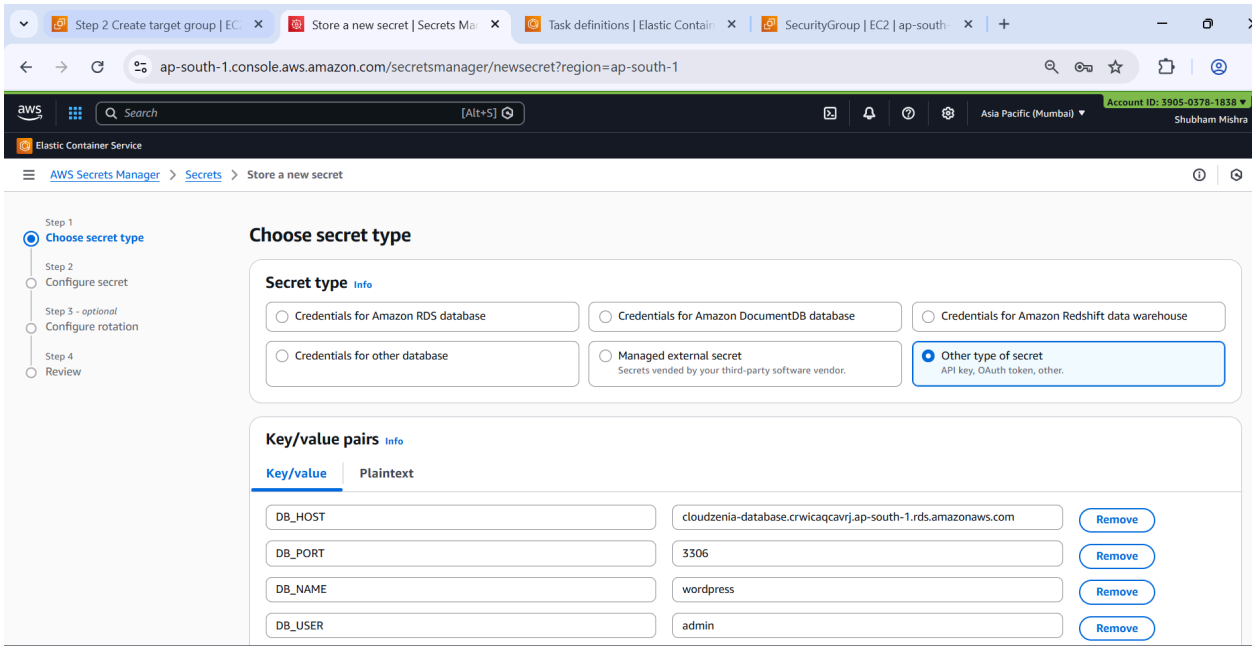


Step 19) Now we will launch any ec2 and install mysql so that we can create database for wordpress



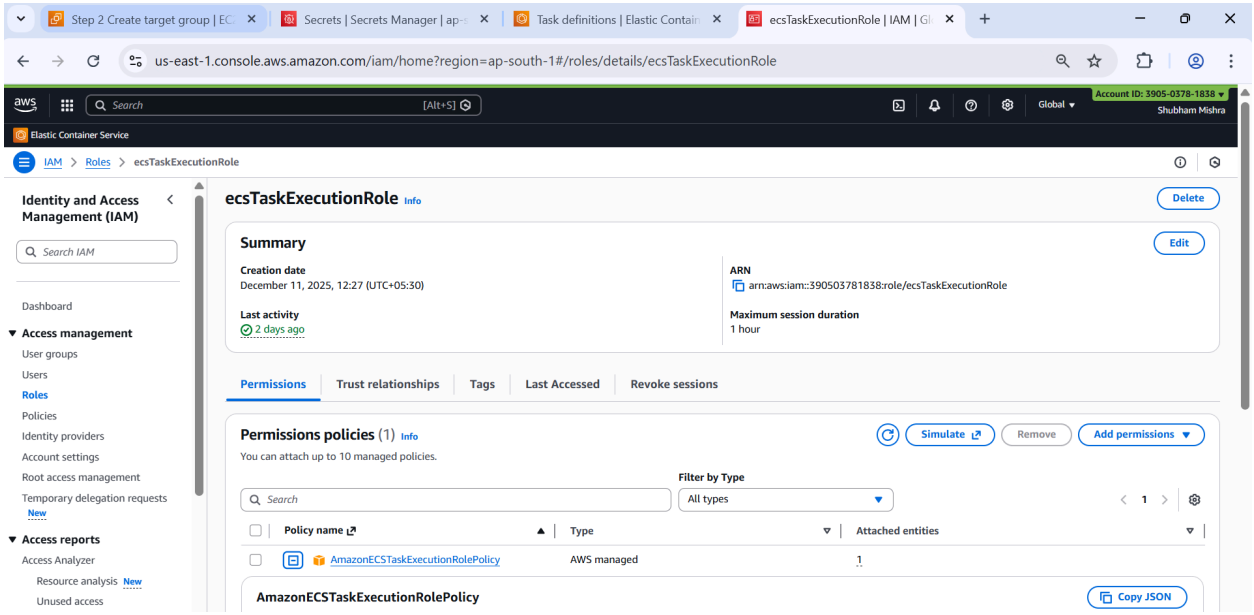


Step 20) Now we will store db secrets in secret manager

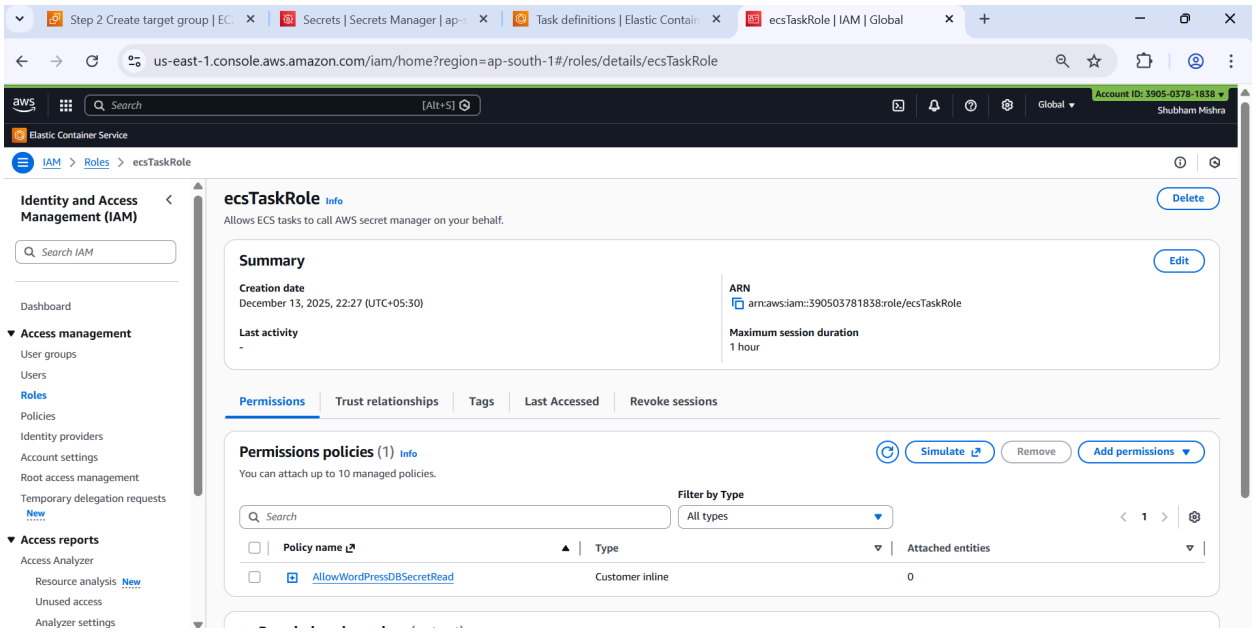


Step 21) Now we will create Iam role for ecs task definition

a) ECS task execution role(used by ECS control plane)

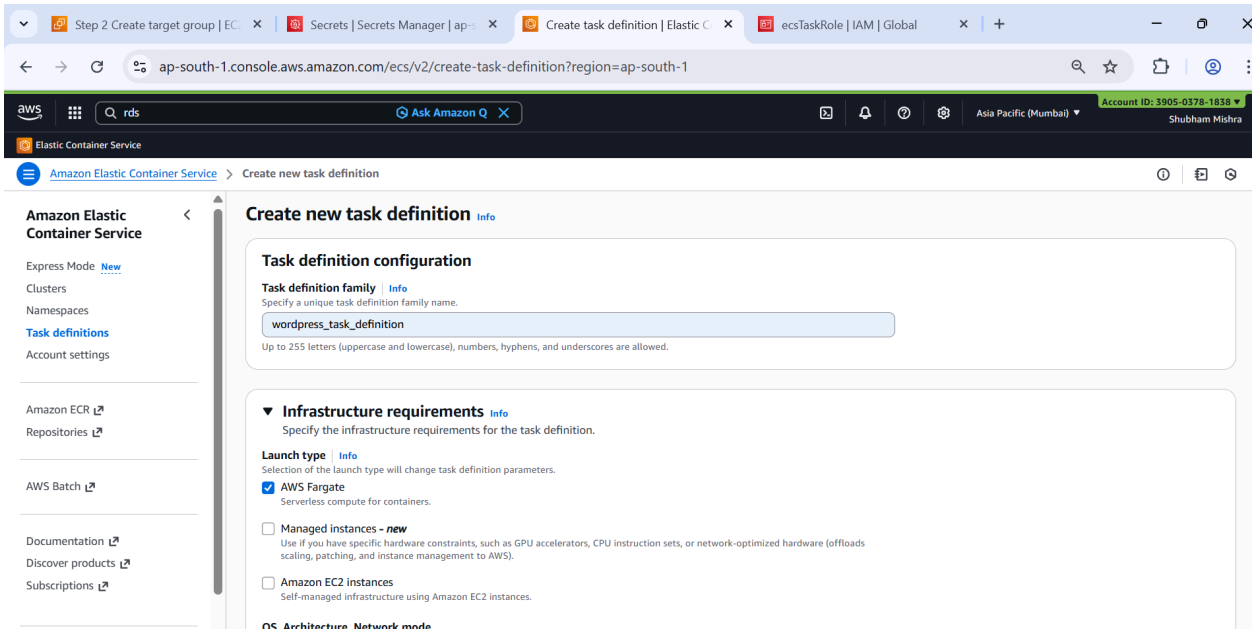


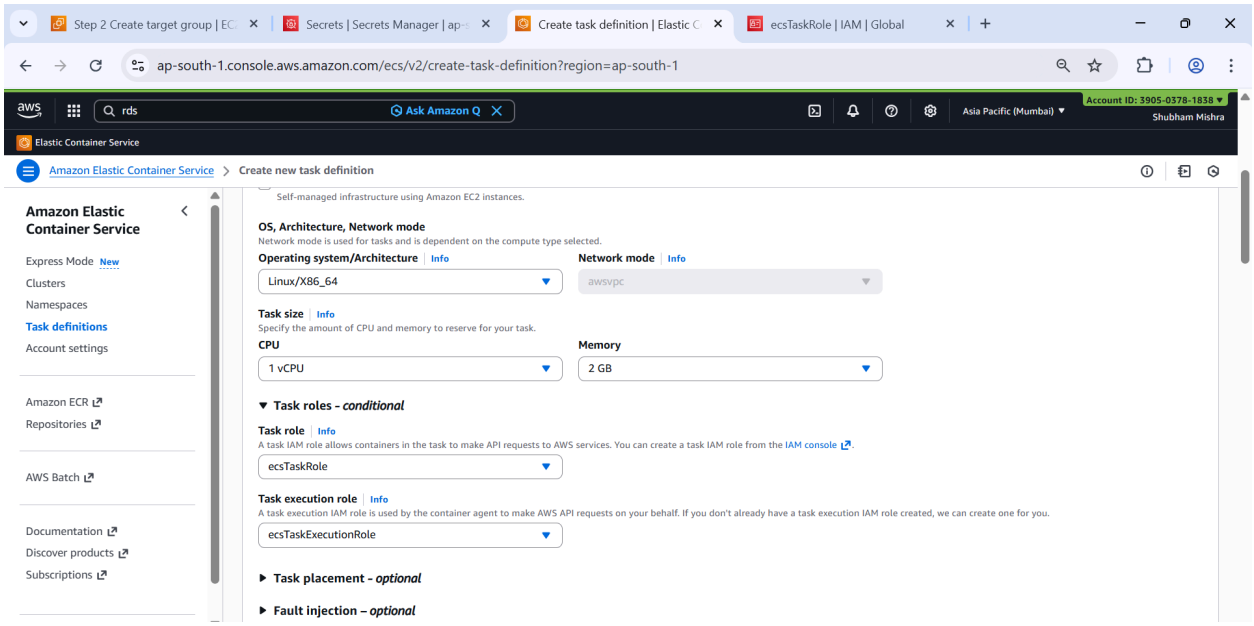
b) ECS task role (used by application inside containers)

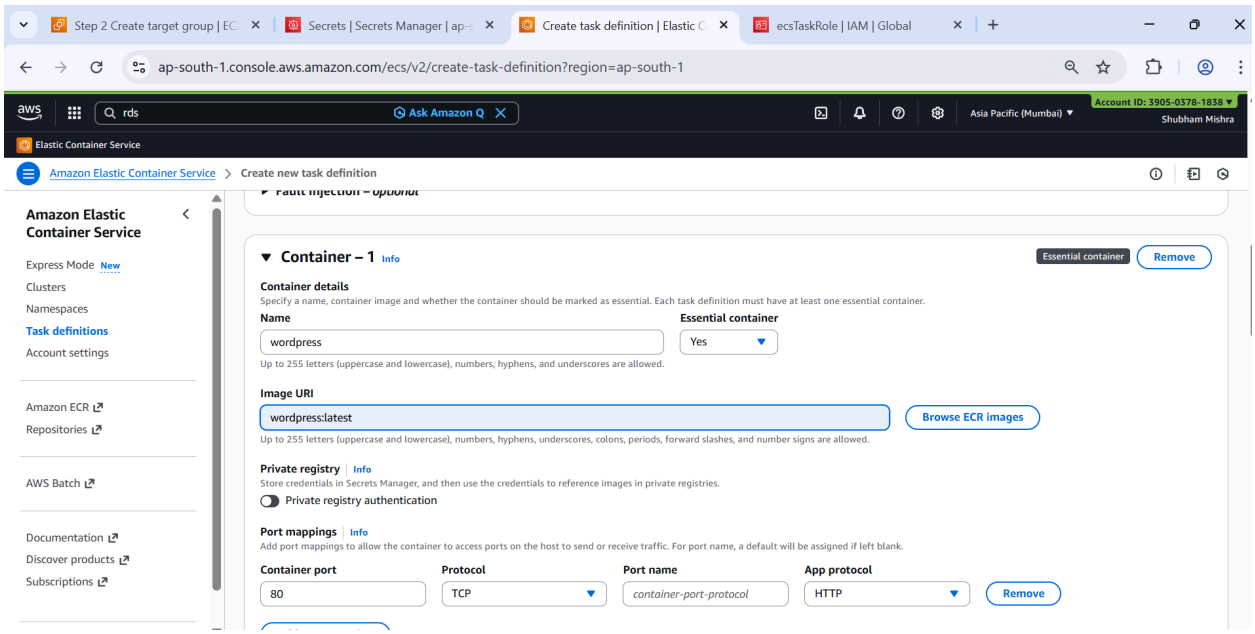


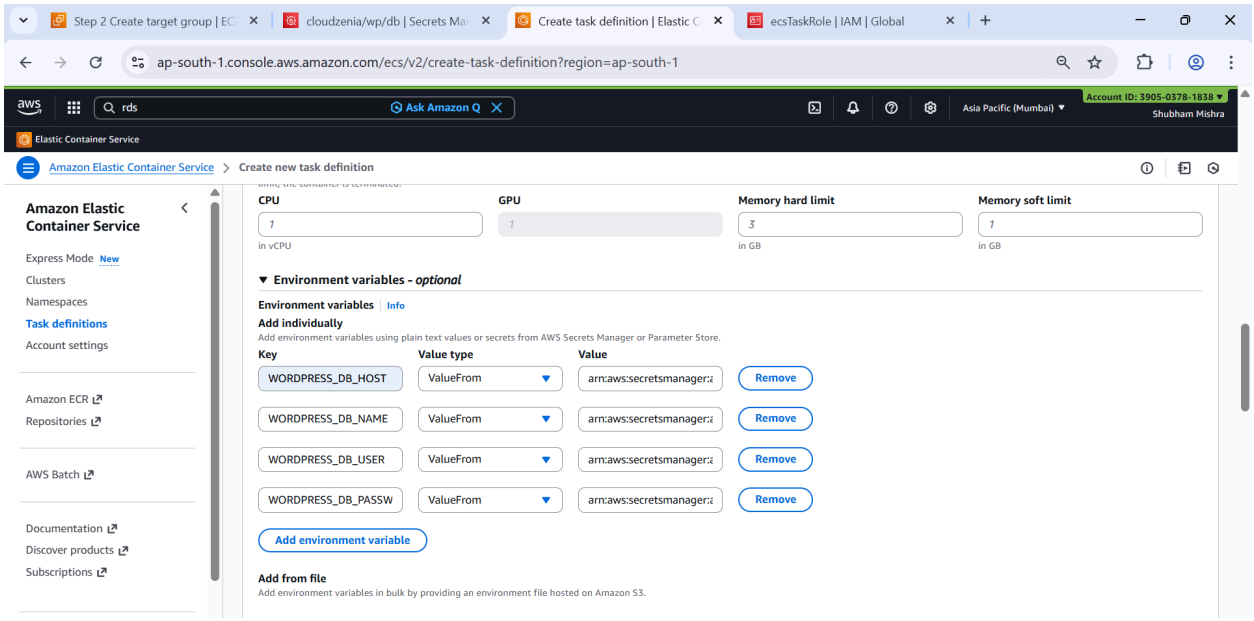
Step 22) Now we will create task definition for ecs cluster

a) Task definition for wordpress

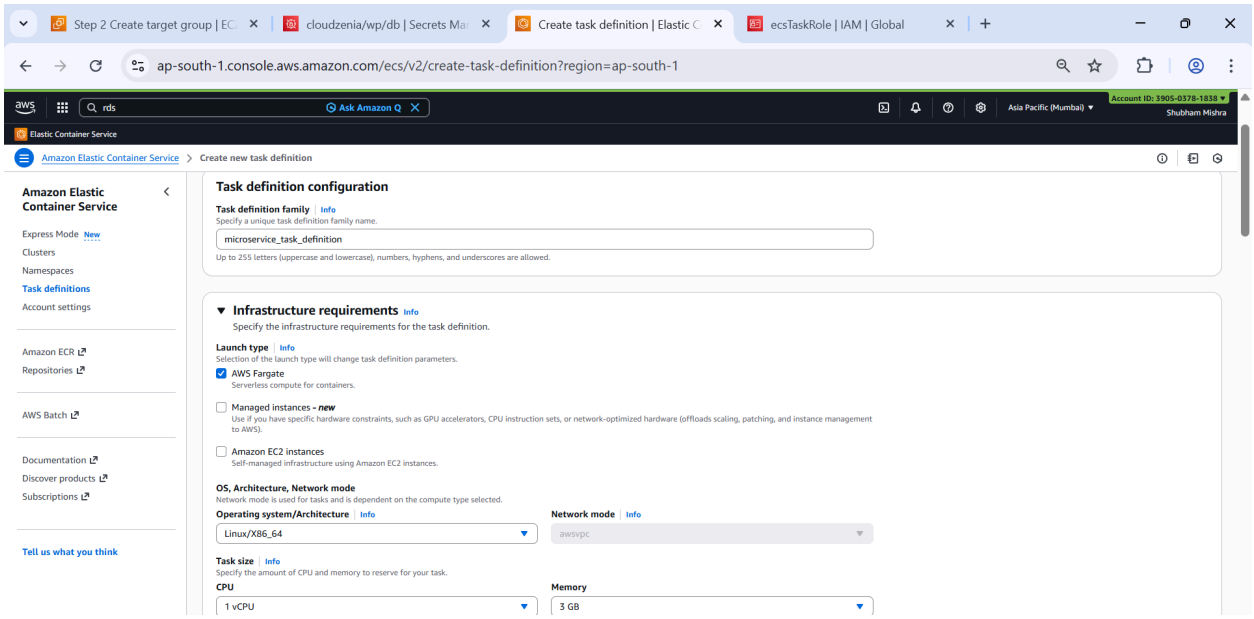


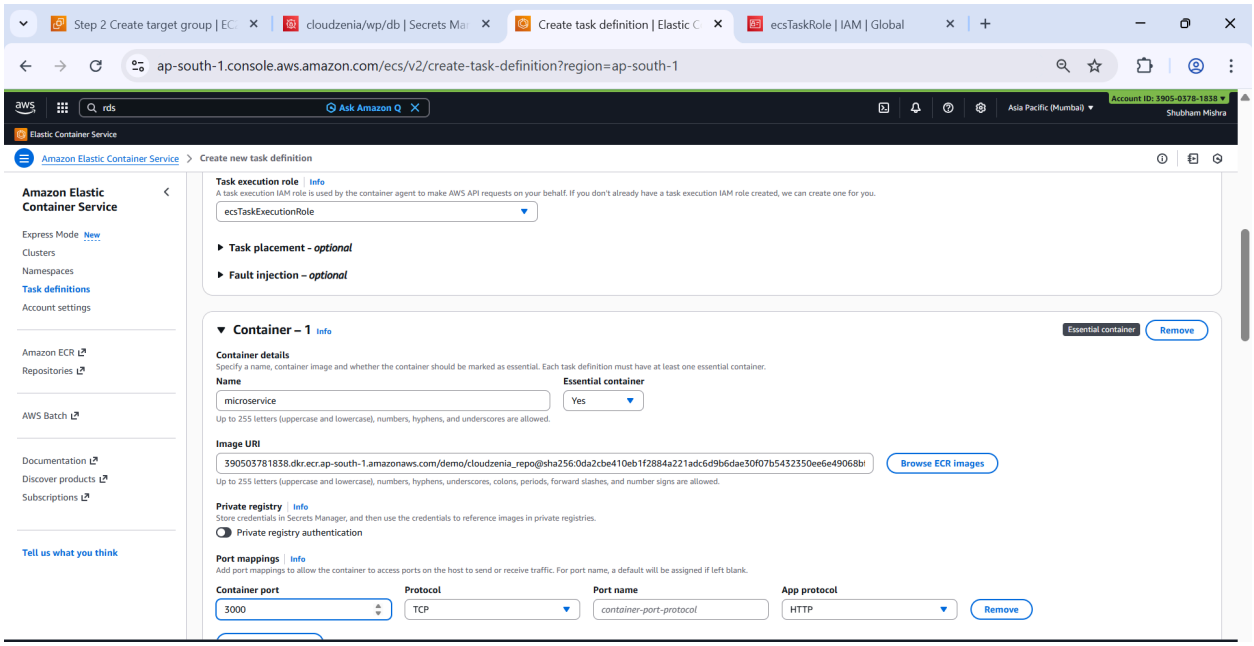

Step 2 Create target group | EC
Secrets | Secrets Manager | ap
Create task definition | Elastic C
ecsTaskRole | IAM | Global
ap-south-1.console.aws.amazon.com/ecs/v2/create-task-definition?region=ap-south-1
aws
rds
Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
Amazon Elastic Container Service
Create new task definition
Amazon Elastic Container Service
Express Mode New
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Self-managed infrastructure using Amazon EC2 instances.
OS, Architecture, Network mode
Network mode is used for tasks and is dependent on the compute type selected.
Operating system/Architecture | Info
Linux/X86_64
Network mode | Info
awsvpc
Task size | Info
Specify the amount of CPU and memory to reserve for your task.
CPU
1 vCPU
Memory
2 GB
Task roles - conditional
Task role | Info
A task IAM role allows containers in the task to make API requests to AWS services. You can create a task IAM role from the IAM console.
ecsTaskRole
Task execution role | Info
A task execution IAM role is used by the container agent to make AWS API requests on your behalf. If you don't already have a task execution IAM role created, we can create one for you.
ecsTaskExecutionRole
Task placement - optional
Fault injection – optional

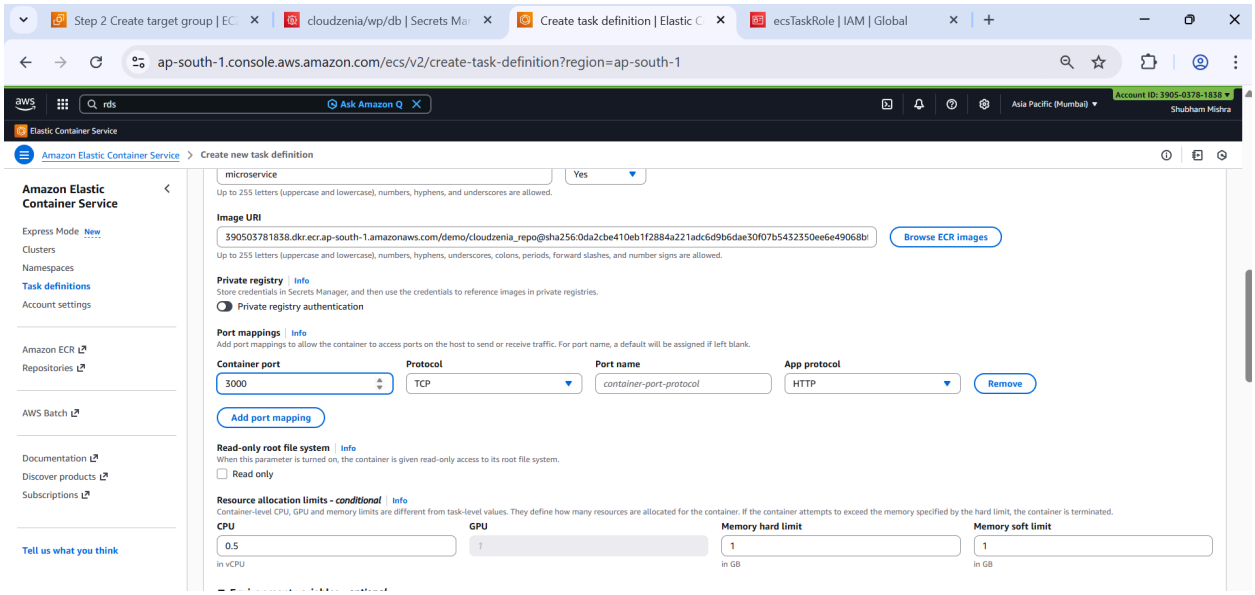

Step 2 Create target group | EC
Secrets | Secrets Manager | ap
Create task definition | Elastic C
ecsTaskRole | IAM | Global
ap-south-1.console.aws.amazon.com/ecs/v2/create-task-definition?region=ap-south-1
aws
rds
Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
Amazon Elastic Container Service
Create new task definition
Amazon Elastic Container Service
Express Mode New
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Container – 1 Info
Essential container
Remove
Container details
Specify a name, container image and whether the container should be marked as essential. Each task definition must have at least one essential container.
Name
wordpress
Essential container
Yes
Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.
Image URI
wordpress:latest
Browse ECR images
Up to 255 letters (uppercase and lowercase), numbers, hyphens, underscores, colons, periods, forward slashes, and number signs are allowed.
Private registry | Info
Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.
Private registry authentication
Port mappings | Info
Add port mappings to allow the container to access ports on the host to send or receive traffic. For port name, a default will be assigned if left blank.
Container port
80
Protocol
TCP
Port name
container-port-protocol
App protocol
HTTP
Remove



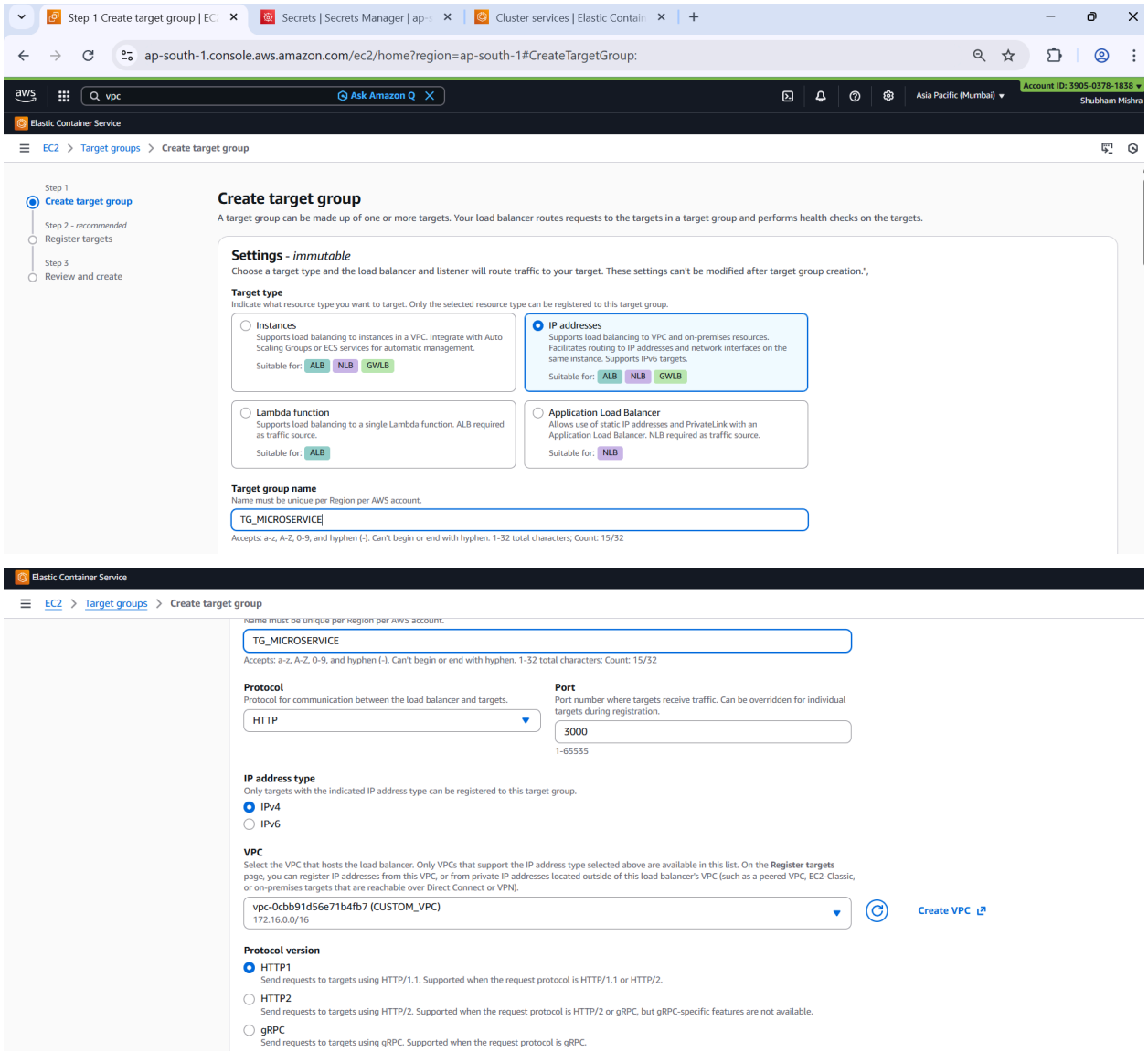
b) Task definition for nodejs microservice

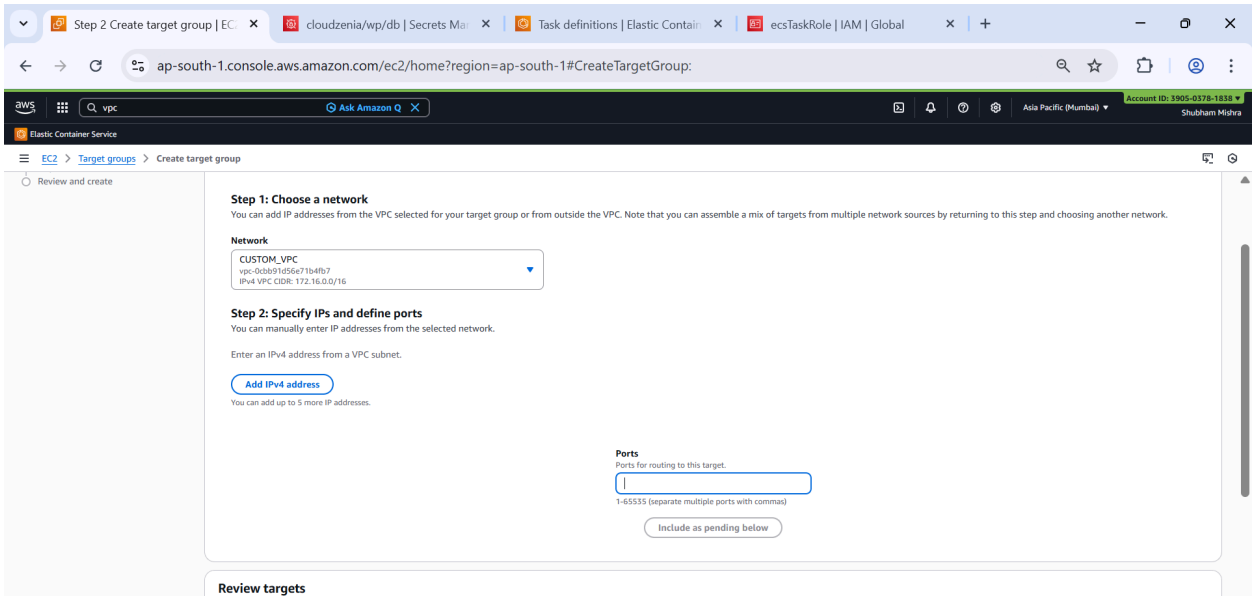


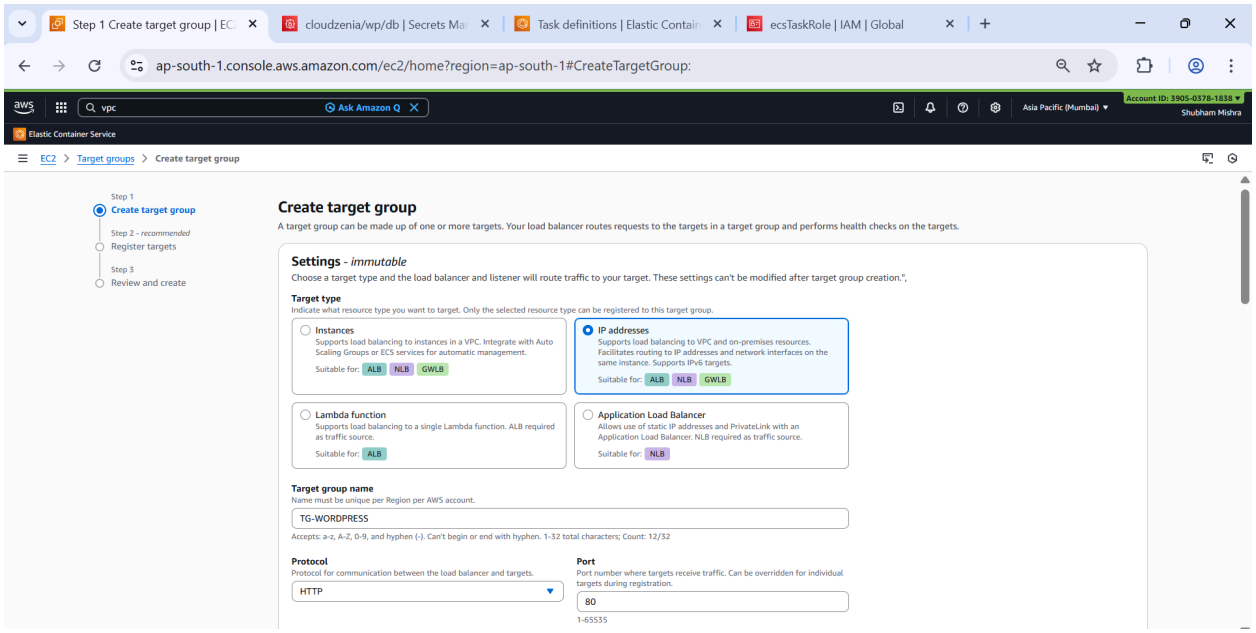


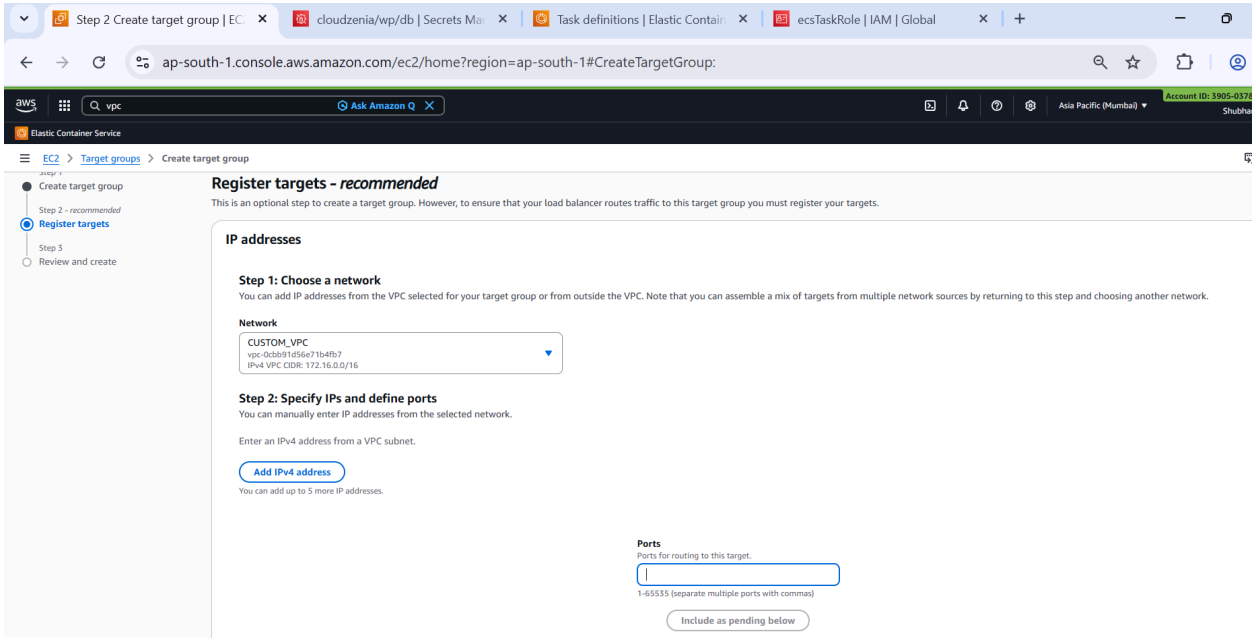


Step 23) Now we will create two Target groups for our services like TG_MICROSERVICE and TG_WORDPRRESS


Step 1 Create target group | EC
Secrets | Secrets Manager | ap-
Cluster services | Elastic Contain
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTargetGroup:
vpc
Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
EC2 > Target groups > Create target group
Step 1
Create target group
Step 2 - recommended
Register targets
Step 3
Review and create
Create target group
A target group can be made up of one or more targets. Your load balancer routes requests to the targets in a target group and performs health checks on the targets.
Settings - immutable
Choose a target type and the load balancer and listener will route traffic to your target. These settings can't be modified after target group creation.",
Target type
Indicate what resource type you want to target. Only the selected resource type can be registered to this target group.
Instances
Supports load balancing to instances in a VPC. Integrate with Auto Scaling Groups or ECS services for automatic management.
Suitable for: ALB NLB GWLB
IP addresses
Supports load balancing to VPC and on-premises resources. Facilitates routing to IP addresses and network interfaces on the same instance. Supports IPv6 targets.
Suitable for: ALB NLB GWLB
Lambda function
Supports load balancing to a single Lambda function. ALB required as traffic source.
Suitable for: ALB
Application Load Balancer
Allows use of static IP addresses and PrivateLink with an Application Load Balancer. NLB required as traffic source.
Suitable for: NLB
Target group name
Name must be unique per Region per AWS account.
TG_MICROSERVICE
Accepts: a-z, A-Z, 0-9, and hyphen (-). Can't begin or end with hyphen. 1-32 total characters; Count: 15/32
Elastic Container Service
EC2 > Target groups > Create target group
Name must be unique per Region per AWS account.
TG_MICROSERVICE
Accepts: a-z, A-Z, 0-9, and hyphen (-). Can't begin or end with hyphen. 1-32 total characters; Count: 15/32
Protocol
Protocol for communication between the load balancer and targets.
HTTP
Port
Port number where targets receive traffic. Can be overridden for individual targets during registration.
3000
1-65535
IP address type
Only targets with the indicated IP address type can be registered to this target group.
IPv4
IPv6
VPC
Select the VPC that hosts the load balancer. Only VPCs that support the IP address type selected above are available in this list. On the Register targets page, you can register IP addresses from this VPC, or from private IP addresses located outside of this load balancer's VPC (such as a peered VPC, EC2-Classic, or on-premises targets that are reachable over Direct Connect or VPN).
vpc-0cbb91d56e71b4fb7 (CUSTOM_VPC)
172.16.0.0/16
Create VPC
Protocol version
HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

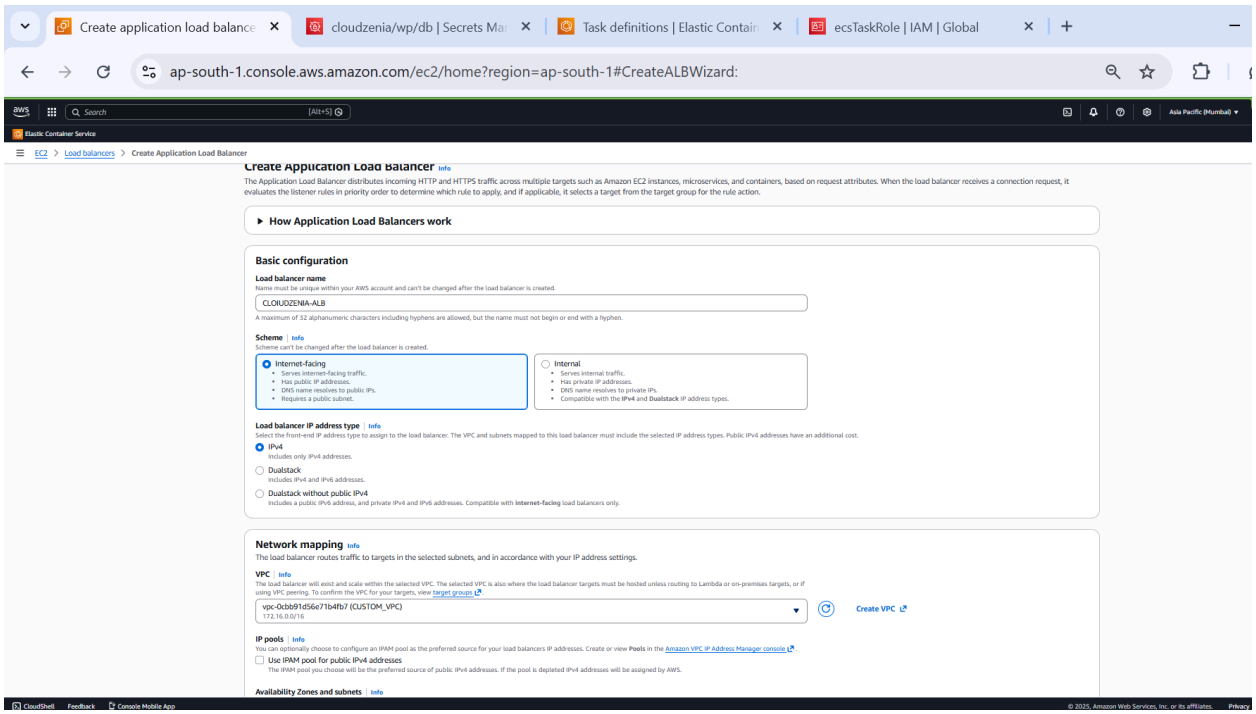

Step 2 Create target group | EC
cloudzenia/wp/db | Secrets Ma
Task definitions | Elastic Contain
ecsTaskRole | IAM | Global
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTargetGroup:
vpc
Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
EC2 > Target groups > Create target group
Review and create
Step 1: Choose a network
You can add IP addresses from the VPC selected for your target group or from outside the VPC. Note that you can assemble a mix of targets from multiple network sources by returning to this step and choosing another network.
Network
CUSTOM_VPC
vpc-0cbb91d56e71b4fb7
IPv4 VPC CIDR: 172.16.0.0/16
Step 2: Specify IPs and define ports
You can manually enter IP addresses from the selected network.
Enter an IPv4 address from a VPC subnet.
Add IPv4 address
You can add up to 5 more IP addresses.
Ports
Ports for routing to this target.
1-65535 (separate multiple ports with commas)
Include as pending below
Review targets

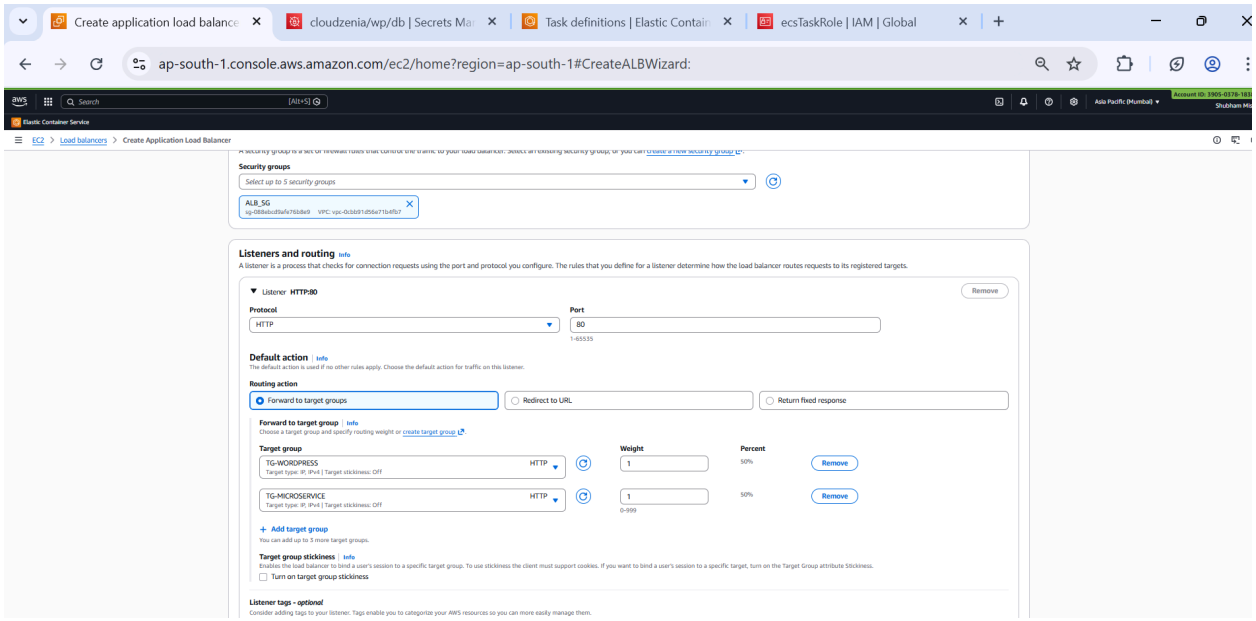

Step 1 Create target group | EC
cloudzenia/wp/db | Secrets Ma
Task definitions | Elastic Contain
ecsTaskRole | IAM | Global
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTargetGroup:
vpc
Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
EC2 > Target groups > Create target group
Step 1
Create target group
Step 2 - recommended
Register targets
Step 3
Review and create
Create target group
A target group can be made up of one or more targets. Your load balancer routes requests to the targets in a target group and performs health checks on the targets.
Settings - immutable
Choose a target type and the load balancer and listener will route traffic to your target. These settings can't be modified after target group creation.",
Target type
Indicate what resource type you want to target. Only the selected resource type can be registered to this target group.
Instances
Supports load balancing to instances in a VPC. Integrate with Auto Scaling Groups or ECS services for automatic management.
Suitable for: ALB NLB GWLB
IP addresses
Supports load balancing to VPC and on-premises resources. Facilitates routing to IP addresses and network interfaces on the same instance. Supports IPv6 targets.
Suitable for: ALB NLB GWLB
Lambda function
Supports load balancing to a single Lambda function. ALB required as traffic source.
Suitable for: ALB
Application Load Balancer
Allows use of static IP addresses and PrivateLink with an Application Load Balancer. NLB required as traffic source.
Suitable for: NLB
Target group name
Name must be unique per Region per AWS account.
TG-WORDPRESS
Accepts: a-z, A-Z, 0-9, and hyphen (-). Can't begin or end with hyphen. 1-32 total characters; Count: 12/32
Protocol
Protocol for communication between the load balancer and targets.
HTTP
Port
Port number where targets receive traffic. Can be overridden for individual targets during registration.
80
1-65535

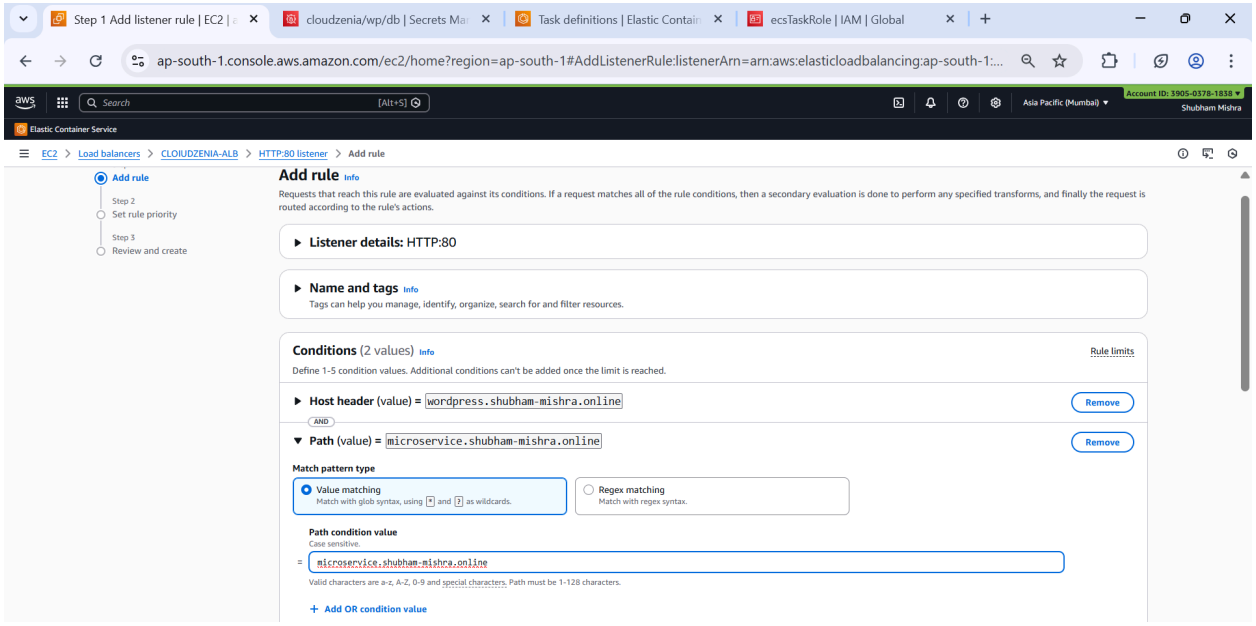


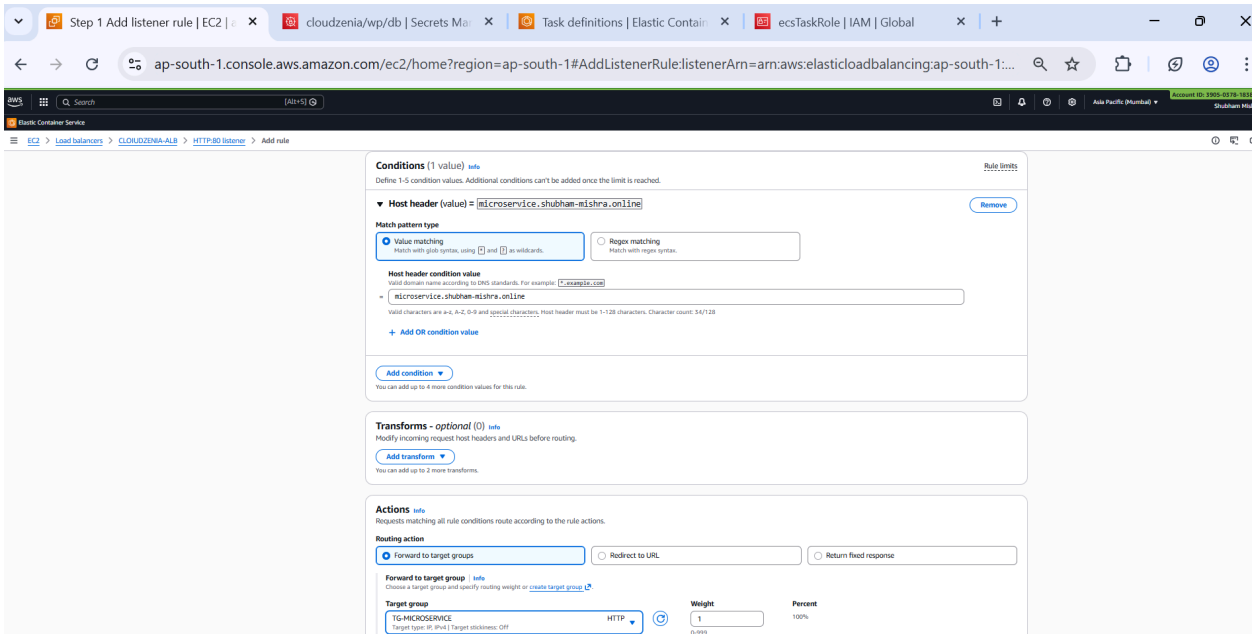
For ECS Fargate, targets are registered automatically by the ECS service, so the target group is initially empty.

STEP 24) Now we will create application loadbalancer

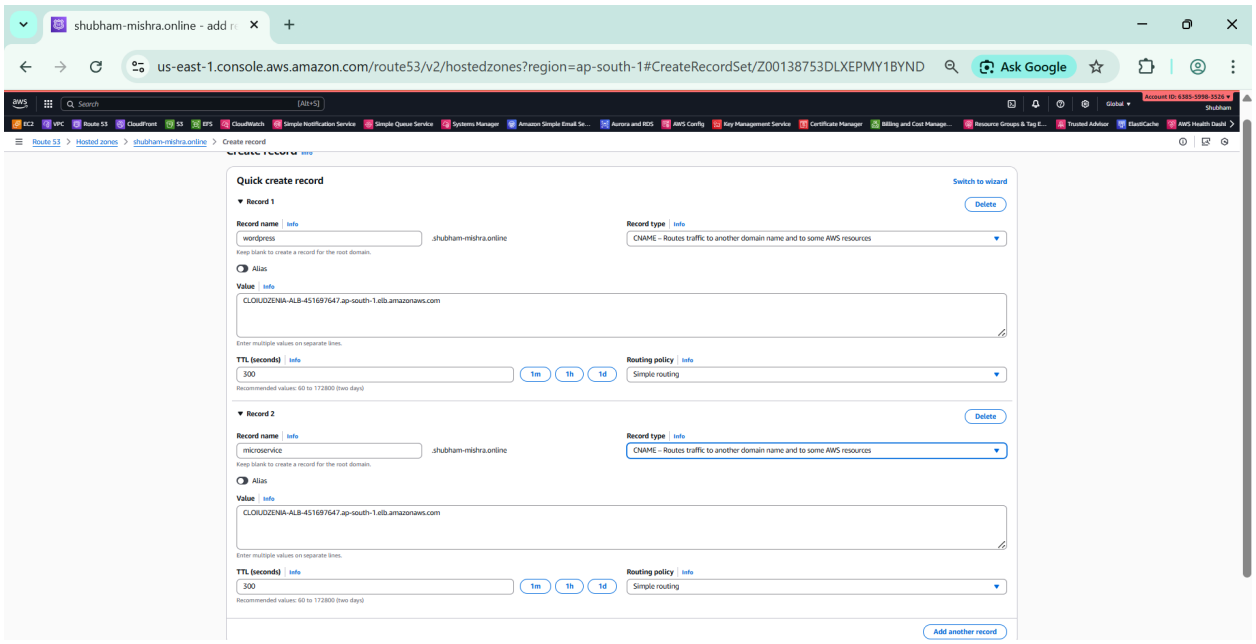



Create application load balance
cloudzenia/wp/db | Secrets Ma
Task definitions | Elastic Contain
ecsTaskRole | IAM | Global
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateALBWizard:
Elastic Container Service
EC2 > Load balancers > Create Application Load Balancer
Security groups
Select up to 5 security groups
ALB_SG
Listeners and routing Info
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.
Listener HTTP:80
Remove
Protocol
HTTP
Port
80
1-65535
Default action Info
The default action is used if no other rules apply. Choose the default action for traffic on this listener.
Routing action
Forward to target groups
Redirect to URL
Return fixed response
Forward to target group Info
Choose a target group and specify routing weight or create target group.
Target group
Weight
Percent
TG-WORDPRESS
HTTP
Target type: IP, IPv4 | Target stickiness: Off
1
50%
Remove
TG-MICROSERVICE
HTTP
Target type: IP, IPv4 | Target stickiness: Off
1
0-999
50%
Remove
Add target group
You can add up to 3 more target groups.
Target group stickiness Info
Enables the load balancer to bind a user's session to a specific target group. To use stickiness the client must support cookies. If you want to bind a user's session to a specific target, turn on the Target Group attribute Stickiness.
Turn on target group stickiness
Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

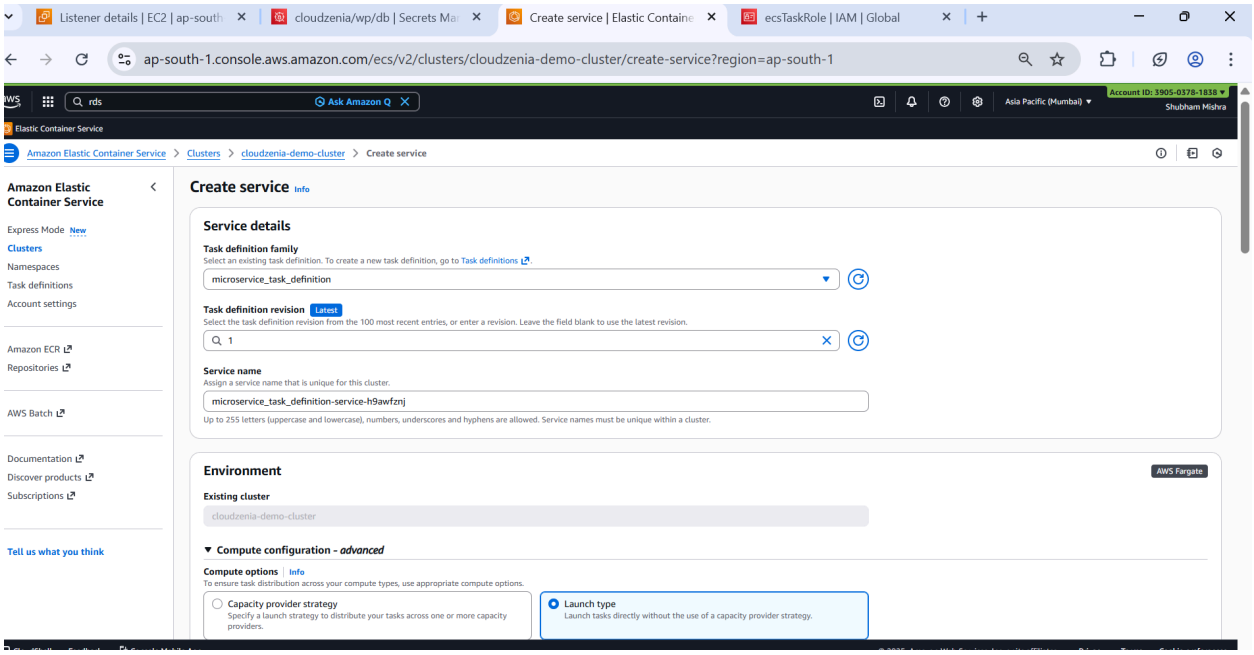

Step 1 Add listener rule | EC2 |
cloudzenia/wp/db | Secrets Ma
Task definitions | Elastic Contain
ecsTaskRole | IAM | Global
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#AddListenerRule:listenerArn=arn:aws:elasticloadbalancing:ap-south-1:...
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
EC2 > Load balancers > CLOIUDZENIA-ALB > HTTP:80 listener > Add rule
Add rule
Step 2
Set rule priority
Step 3
Review and create
Add rule Info
Requests that reach this rule are evaluated against its conditions. If a request matches all of the rule conditions, then a secondary evaluation is done to perform any specified transforms, and finally the request is routed according to the rule's actions.
Listener details: HTTP:80
Name and tags Info
Tags can help you manage, identify, organize, search for and filter resources.
Conditions (2 values) Info
Rule limits
Define 1-5 condition values. Additional conditions can't be added once the limit is reached.
Host header (value) = wordpress.shubham-mishra.online
Remove
AND
Path (value) = microservice.shubham-mishra.online
Remove
Match pattern type
Value matching
Match with glob syntax, using * and ? as wildcards.
Regex matching
Match with regex syntax.
Path condition value
Case sensitive.
microservice.shubham-mishra.online
Valid characters are a-z, A-Z, 0-9 and special characters. Path must be 1-128 characters.
Add OR condition value

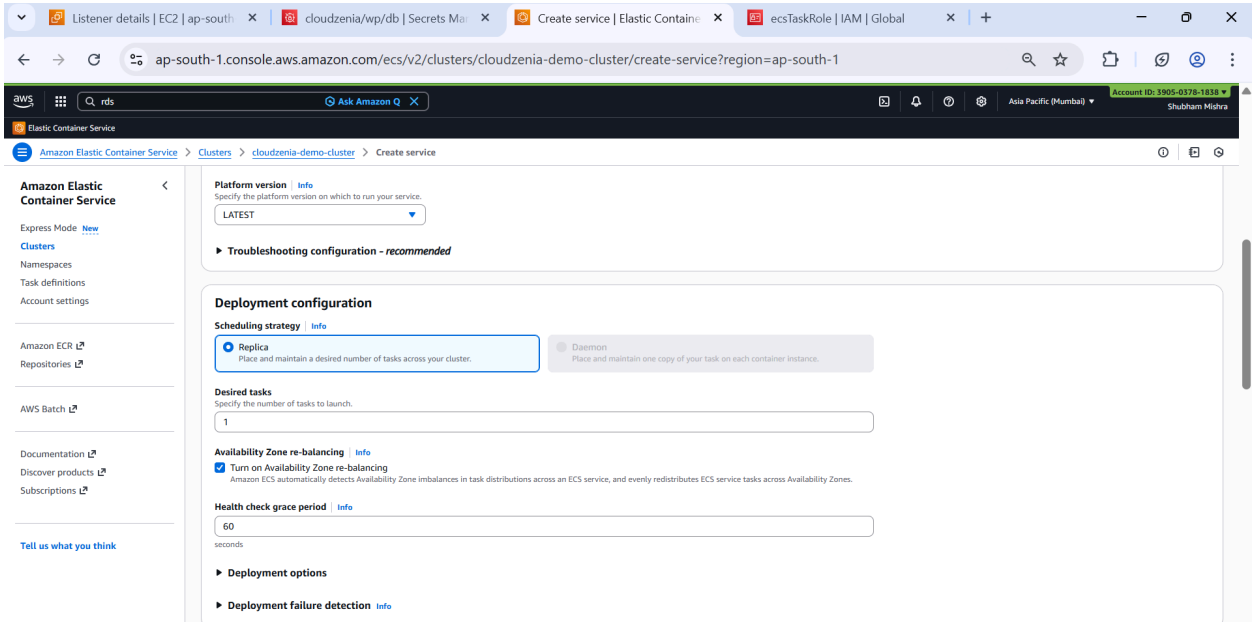


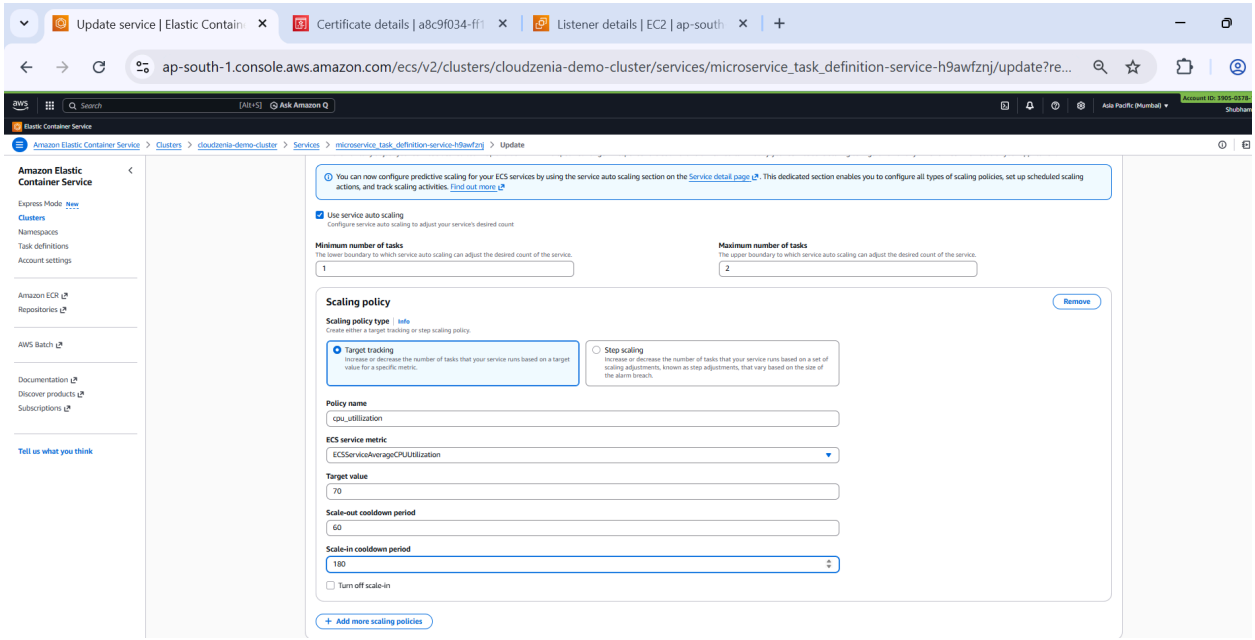
Step 25) lets create subdomain for our microservice and wordpress ecs tasks in different account as i already have domain over there

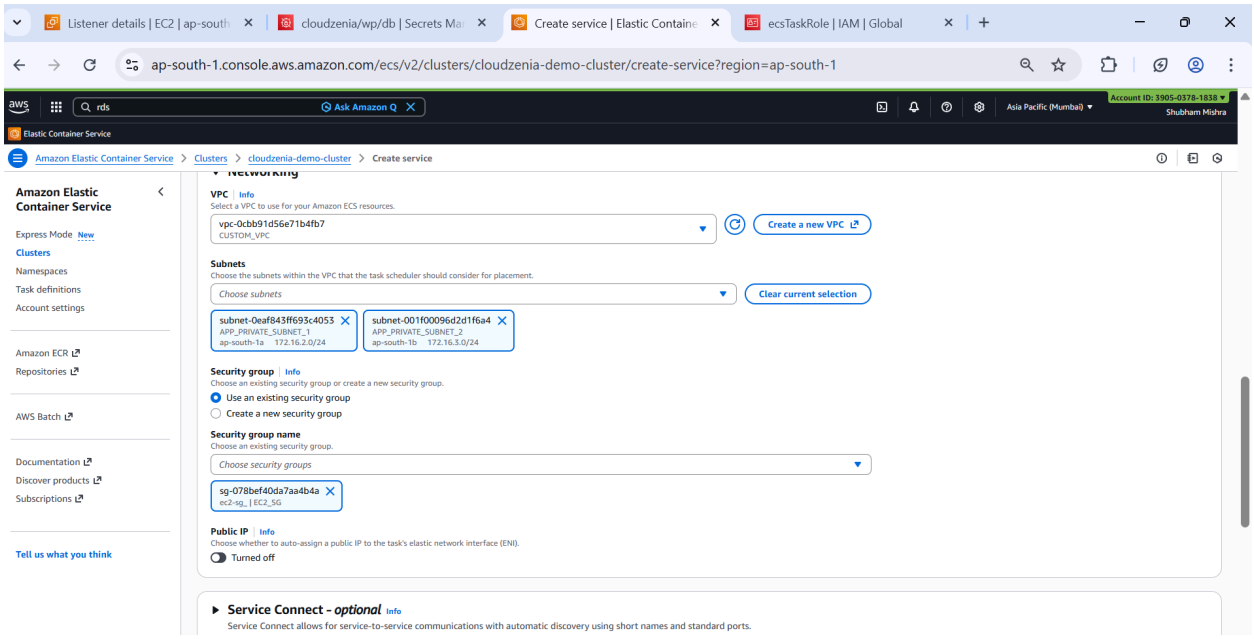


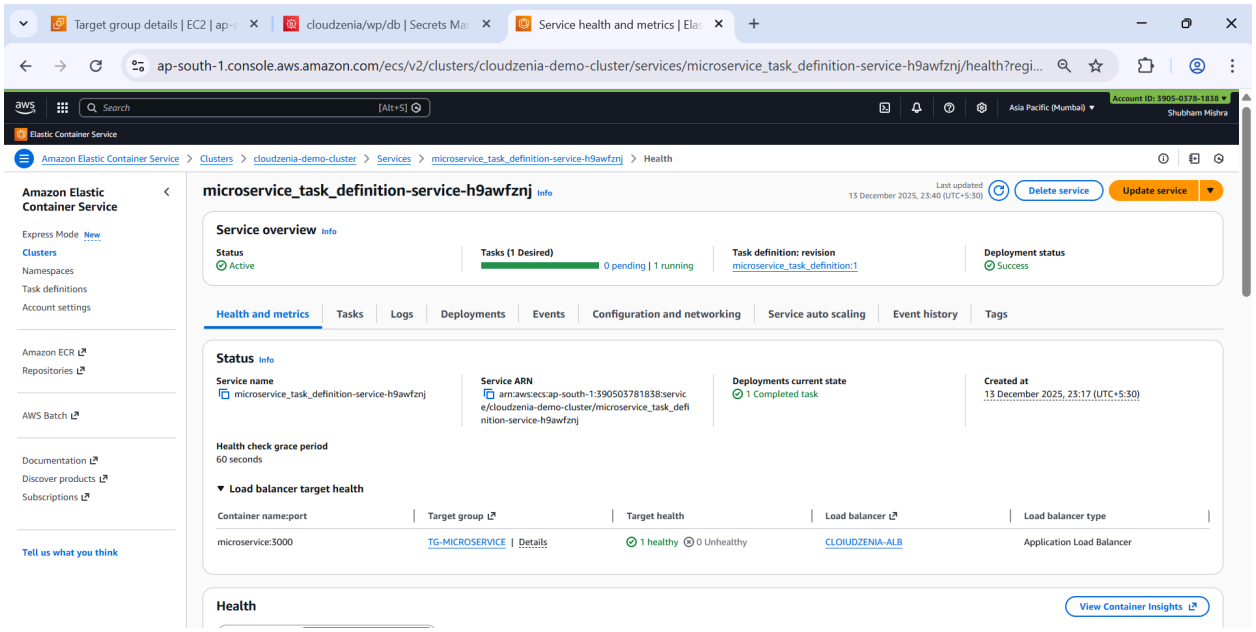
Step 26) Now we will create service for our ECS cluster


Listener details | EC2 | ap-south
cloudzenia/wp/db | Secrets Ma
Create service | Elastic Containe
ecsTaskRole | IAM | Global
ap-south-1.console.aws.amazon.com/ecs/v2/clusters/cloudzenia-demo-cluster/create-service?region=ap-south-1
rds
Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
Amazon Elastic Container Service > Clusters > cloudzenia-demo-cluster > Create service
Amazon Elastic Container Service
Express Mode New
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Tell us what you think
Create service Info
Service details
Task definition family
Select an existing task definition. To create a new task definition, go to Task definitions.
microservice_task_definition
Task definition revision Latest
Select the task definition revision from the 100 most recent entries, or enter a revision. Leave the field blank to use the latest revision.
1
Service name
Assign a service name that is unique for this cluster.
microservice_task_definition-service-h9awfznj
Up to 255 letters (uppercase and lowercase), numbers, underscores and hyphens are allowed. Service names must be unique within a cluster.
Environment
AWS Fargate
Existing cluster
cloudzenia-demo-cluster
Compute configuration - advanced
Compute options Info
To ensure task distribution across your compute types, use appropriate compute options.
Capacity provider strategy
Specify a launch strategy to distribute your tasks across one or more capacity providers.
Launch type
Launch tasks directly without the use of a capacity provider strategy.

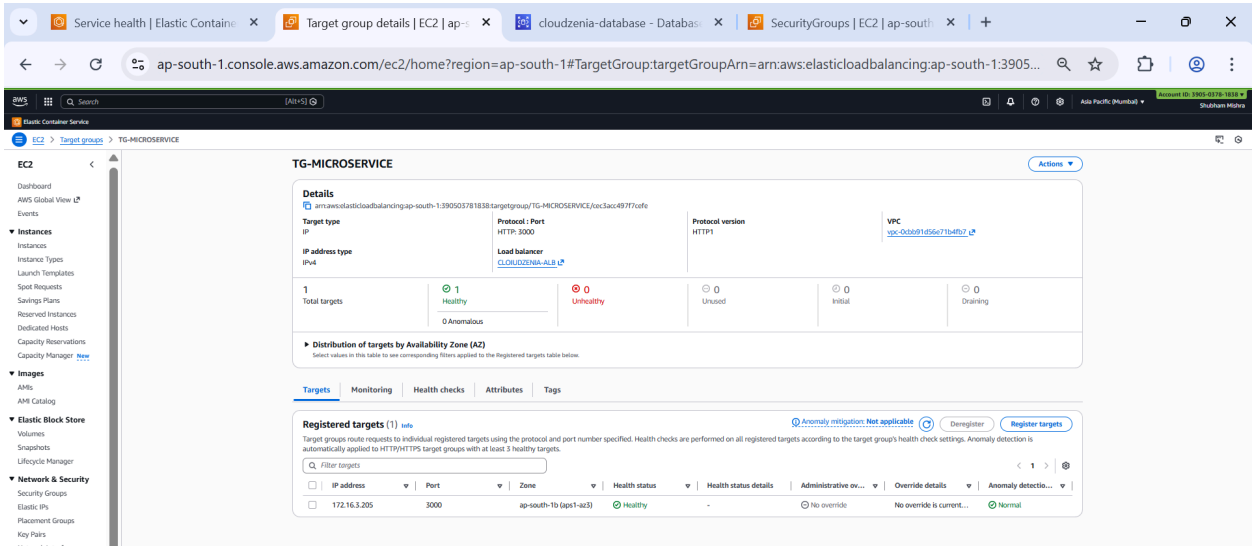

Listener details | EC2 | ap-south
cloudzenia/wp/db | Secrets Ma
Create service | Elastic Containe
ecsTaskRole | IAM | Global
ap-south-1.console.aws.amazon.com/ecs/v2/clusters/cloudzenia-demo-cluster/create-service?region=ap-south-1
rds
Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
Amazon Elastic Container Service > Clusters > cloudzenia-demo-cluster > Create service
Amazon Elastic Container Service
Express Mode New
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Tell us what you think
Platform version Info
Specify the platform version on which to run your service.
LATEST
Troubleshooting configuration - recommended
Deployment configuration
Scheduling strategy Info
Replica
Place and maintain a desired number of tasks across your cluster.
Daemon
Place and maintain one copy of your task on each container instance.
Desired tasks
Specify the number of tasks to launch.
1
Availability Zone re-balancing Info
Turn on Availability Zone re-balancing
Amazon ECS automatically detects Availability Zone imbalances in task distributions across an ECS service, and evenly redistributes ECS service tasks across Availability Zones.
Health check grace period Info
60
seconds
Deployment options
Deployment failure detection Info


Update service | Elastic Contain
Certificate details | a8c9f034-ff1
Listener details | EC2 | ap-south
ap-south-1.console.aws.amazon.com/ecs/v2/clusters/cloudzenia-demo-cluster/services/microservice_task_definition-service-h9awfznj/update?re...
Amazon Elastic Container Service
Express Mode New
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Tell us what you think
You can now configure predictive scaling for your ECS services by using the service auto scaling section on the Service detail page. This dedicated section enables you to configure all types of scaling policies, set up scheduled scaling actions, and track scaling activities. Find out more
Use service auto scaling
Configure service auto scaling to adjust your service's desired count
Minimum number of tasks
The lower boundary to which service auto scaling can adjust the desired count of the service.
1
Maximum number of tasks
The upper boundary to which service auto scaling can adjust the desired count of the service.
2
Scaling policy
Remove
Scaling policy type | Info
Create either a target tracking or step scaling policy.
Target tracking
Increase or decrease the number of tasks that your service runs based on a target value for a specific metric.
Step scaling
Increase or decrease the number of tasks that your service runs based on a set of scaling adjustments, known as step adjustments, that vary based on the size of the alarm breach.
Policy name
cpu_utillization
ECS service metric
ECSServiceAverageCPUUtilization
Target value
70
Scale-out cooldown period
60
Scale-in cooldown period
180
Turn off scale-in
Add more scaling policies


Listener details | EC2 | ap-south
cloudzenia/wp/db | Secrets Ma
Create service | Elastic Containe
ecsTaskRole | IAM | Global
ap-south-1.console.aws.amazon.com/ecs/v2/clusters/cloudzenia-demo-cluster/create-service?region=ap-south-1
Account ID: 3905-0378-1838
Shubham Mishra
Asia Pacific (Mumbai)
Amazon Elastic Container Service > Clusters > cloudzenia-demo-cluster > Create service
Networking
VPC | Info
Select a VPC to use for your Amazon ECS resources.
vpc-0cbb91d56e71b4fb7
CUSTOM_VPC
Create a new VPC
Subnets
Choose the subnets within the VPC that the task scheduler should consider for placement.
Choose subnets
Clear current selection
subnet-0eaf843ff693c4053
APP_PRIVATE_SUBNET_1
ap-south-1a 172.16.2.0/24
subnet-001f00096d2d1f6a4
APP_PRIVATE_SUBNET_2
ap-south-1b 172.16.3.0/24
Security group | Info
Choose an existing security group or create a new security group.
Use an existing security group
Create a new security group
Security group name
Choose an existing security group.
Choose security groups
sg-078bef40da7aa4b4a
ec2-sg_ | EC2_SG
Public IP | Info
Choose whether to auto-assign a public IP to the task's elastic network interface (ENI).
Turned off
Service Connect - optional Info
Service Connect allows for service-to-service communications with automatic discovery using short names and standard ports.

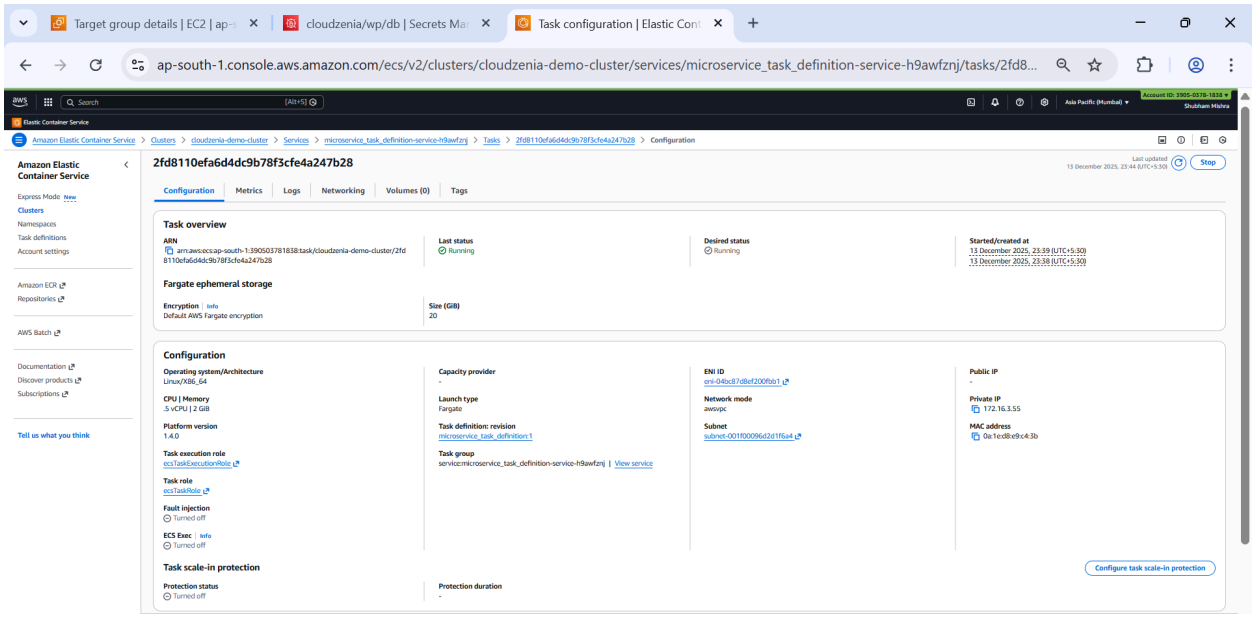


a) Service for microservice task created successfully.



TG_MICROSERVICE registered target

STEP 27) Once service created we can see task(container is running) created.

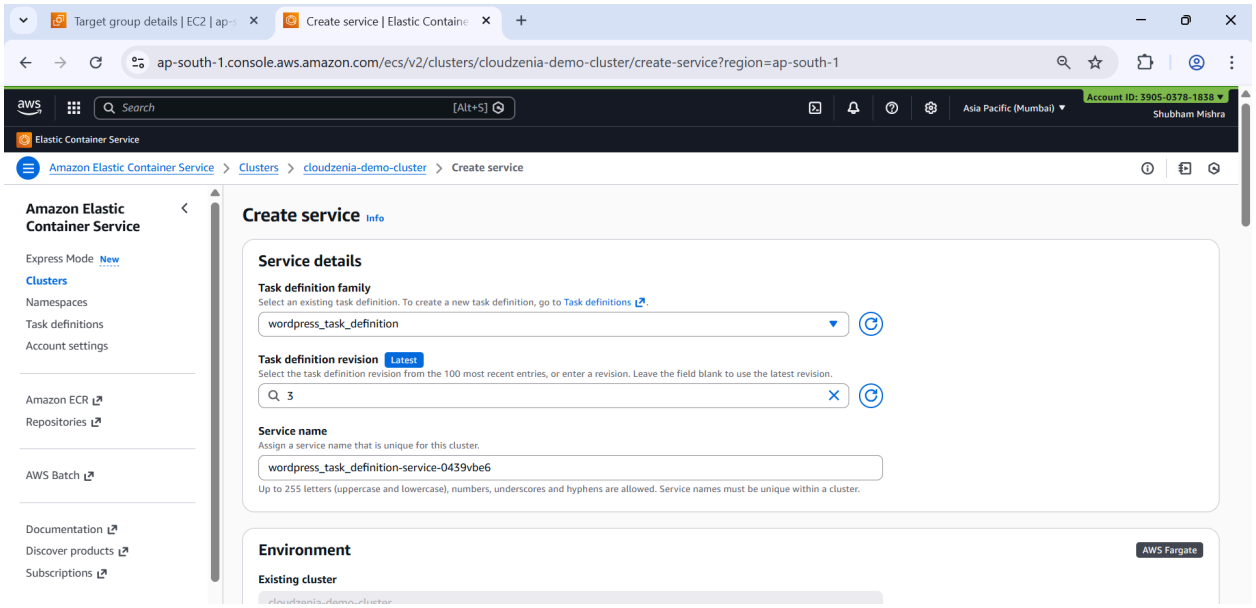


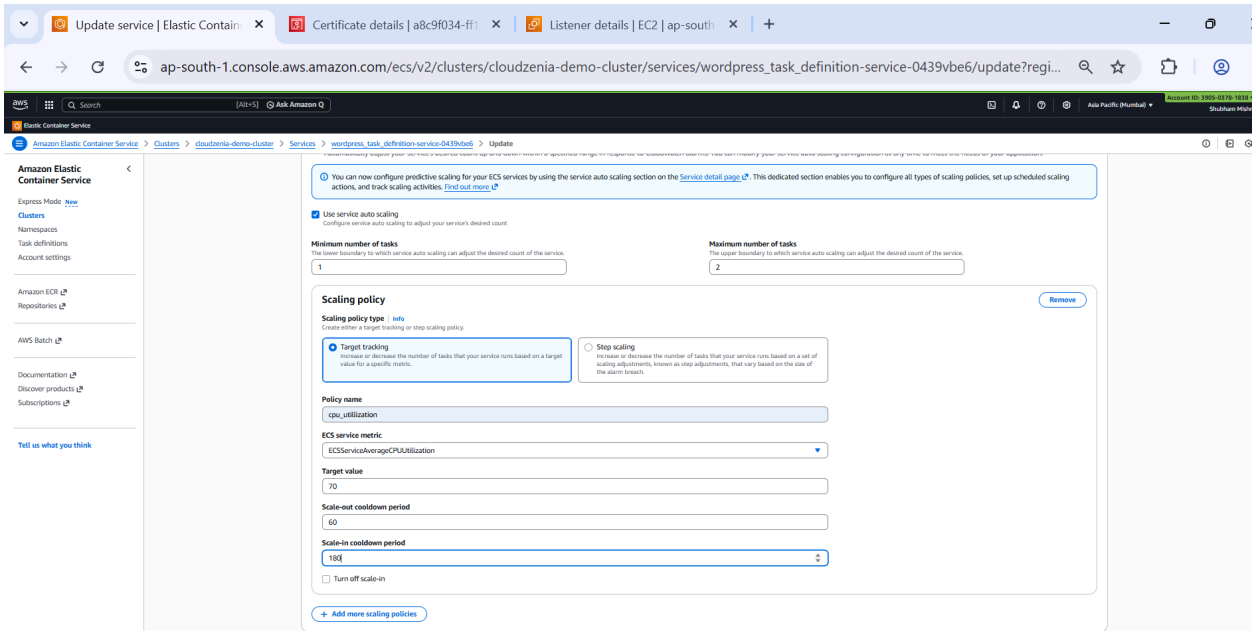
Step 28) Now lets verify accessibility from subdomain

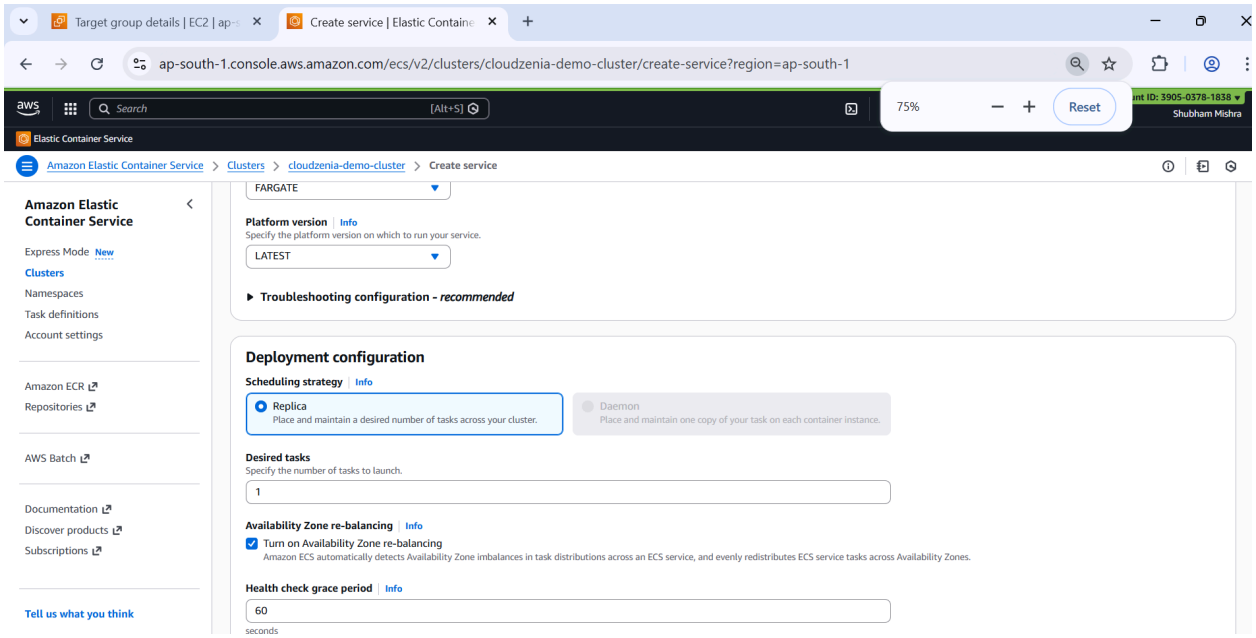
Hello from Microservice

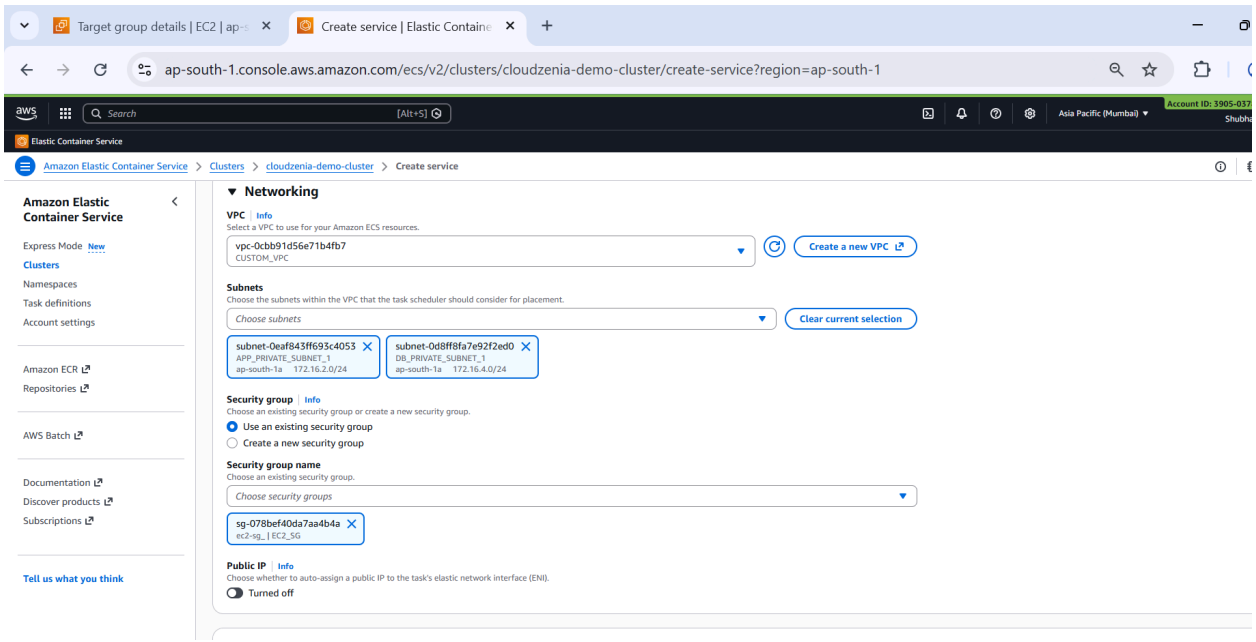
This confirms our micrservice ecs task is successfully created and it is responding over alb tg and route53 .

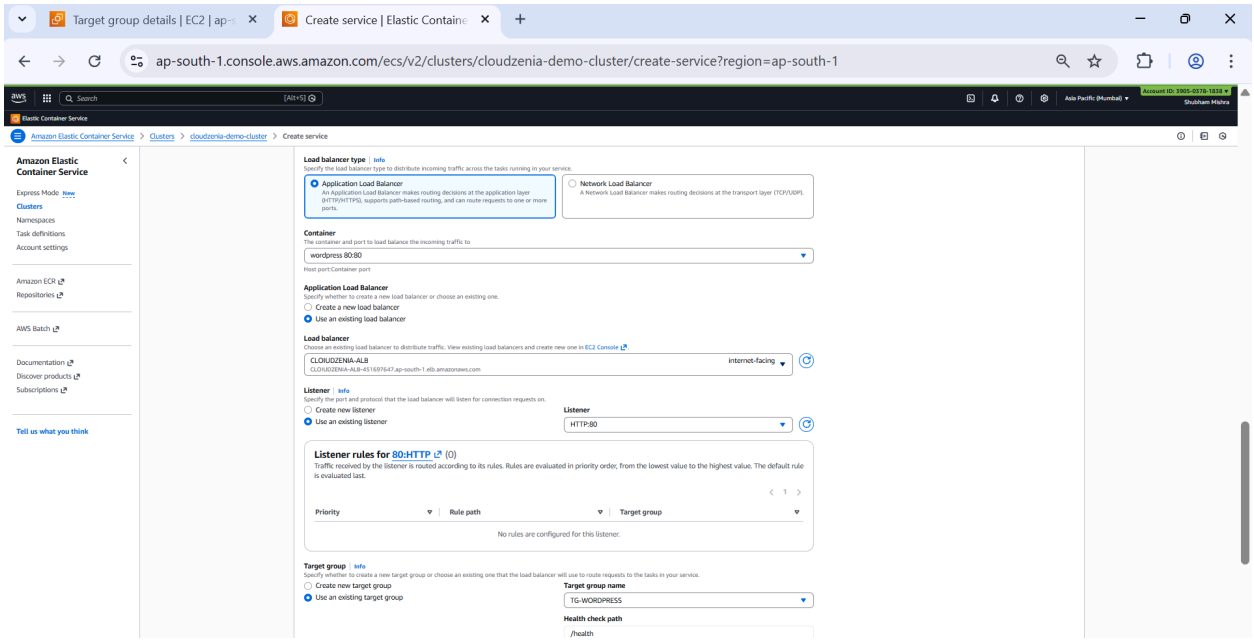
Step 29) Now lets create another service in another task in ecs cluster for wordpress task definition


Target group details | EC2 | ap-s
Create service | Elastic Containe
ap-south-1.console.aws.amazon.com/ecs/v2/clusters/cloudzenia-demo-cluster/create-service?region=ap-south-1
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 3905-0378-1838
Shubham Mishra
Elastic Container Service
Amazon Elastic Container Service > Clusters > cloudzenia-demo-cluster > Create service
Amazon Elastic Container Service
Express Mode New
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Create service Info
Service details
Task definition family
Select an existing task definition. To create a new task definition, go to Task definitions.
wordpress_task_definition
Task definition revision Latest
Select the task definition revision from the 100 most recent entries, or enter a revision. Leave the field blank to use the latest revision.
3
Service name
Assign a service name that is unique for this cluster.
wordpress_task_definition-service-0439vbe6
Up to 255 letters (uppercase and lowercase), numbers, underscores and hyphens are allowed. Service names must be unique within a cluster.
Environment
AWS Fargate
Existing cluster


Update service | Elastic Contain
Certificate details | a8c9f034-ff1
Listener details | EC2 | ap-south
ap-south-1.console.aws.amazon.com/ecs/v2/clusters/cloudzenia-demo-cluster/services/wordpress_task_definition-service-0439vbe6/update?regi...
Amazon Elastic Container Service > Clusters > cloudzenia-demo-cluster > Services > wordpress_task_definition-service-0439vbe6 > Update
You can now configure predictive scaling for your ECS services by using the service auto scaling section on the Service detail page. This dedicated section enables you to configure all types of scaling policies, set up scheduled scaling actions, and track scaling activities. Find out more
Use service auto scaling
Configure service auto scaling to adjust your service's desired count
Minimum number of tasks
The lower boundary to which service auto scaling can adjust the desired count of the service.
1
Maximum number of tasks
The upper boundary to which service auto scaling can adjust the desired count of the service.
2
Scaling policy
Remove
Scaling policy type Info
Create either a target tracking or step scaling policy.
Target tracking
Increase or decrease the number of tasks that your service runs based on a target value for a specific metric.
Step scaling
Increase or decrease the number of tasks that your service runs based on a set of scaling adjustments, known as step adjustments, that vary based on the size of the alarm breach.
Policy name
cpu_utilization
ECS service metric
ECSServiceAverageCPUUtilization
Target value
70
Scale-out cooldown period
60
Scale-in cooldown period
180
Turn off scale-in
Add more scaling policies

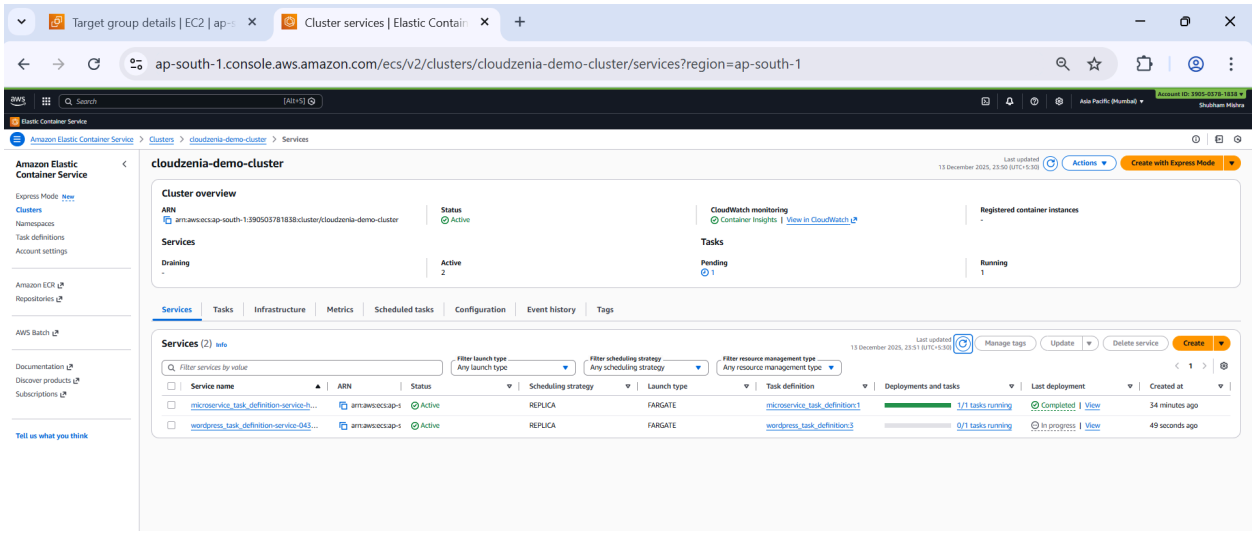

Target group details | EC2 | ap-s
Create service | Elastic Containe
ap-south-1.console.aws.amazon.com/ecs/v2/clusters/cloudzenia-demo-cluster/create-service?region=ap-south-1
75%
Reset
Search
[Alt+S]
Shubham Mishra
Elastic Container Service
Amazon Elastic Container Service > Clusters > cloudzenia-demo-cluster > Create service
Amazon Elastic Container Service
Express Mode New
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Tell us what you think
FARGATE
Platform version Info
Specify the platform version on which to run your service.
LATEST
Troubleshooting configuration - recommended
Deployment configuration
Scheduling strategy Info
Replica
Place and maintain a desired number of tasks across your cluster.
Daemon
Place and maintain one copy of your task on each container instance.
Desired tasks
Specify the number of tasks to launch.
1
Availability Zone re-balancing Info
Turn on Availability Zone re-balancing
Amazon ECS automatically detects Availability Zone imbalances in task distributions across an ECS service, and evenly redistributes ECS service tasks across Availability Zones.
Health check grace period Info
60
seconds


Target group details | EC2 | ap-s
Create service | Elastic Containe
ap-south-1.console.aws.amazon.com/ecs/v2/clusters/cloudzenia-demo-cluster/create-service?region=ap-south-1
Search
[Alt+S]
Asia Pacific (Mumbai)
Elastic Container Service
Amazon Elastic Container Service > Clusters > cloudzenia-demo-cluster > Create service
Amazon Elastic Container Service
Express Mode New
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Tell us what you think
Networking
VPC Info
Select a VPC to use for your Amazon ECS resources.
vpc-0cbb91d56e71b4fb7
CUSTOM_VPC
Create a new VPC
Subnets
Choose the subnets within the VPC that the task scheduler should consider for placement.
Choose subnets
Clear current selection
subnet-0eaf843ff693c4053
APP_PRIVATE_SUBNET_1
ap-south-1a 172.16.2.0/24
subnet-0d8ff8fa7e92f2ed0
DB_PRIVATE_SUBNET_1
ap-south-1a 172.16.4.0/24
Security group Info
Choose an existing security group or create a new security group.
Use an existing security group
Create a new security group
Security group name
Choose an existing security group.
Choose security groups
sg-078bef40da7aa4b4a
ec2-sg_ | EC2_SG
Public IP Info
Choose whether to auto-assign a public IP to the task's elastic network interface (ENI).
Turned off

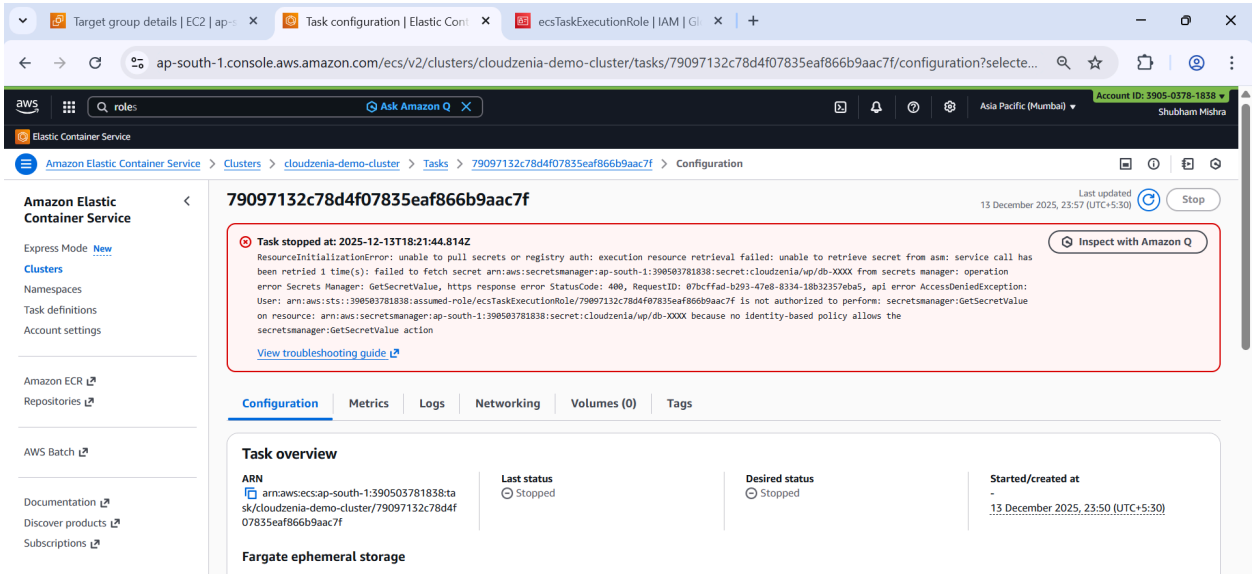


Service created for wordpress task.

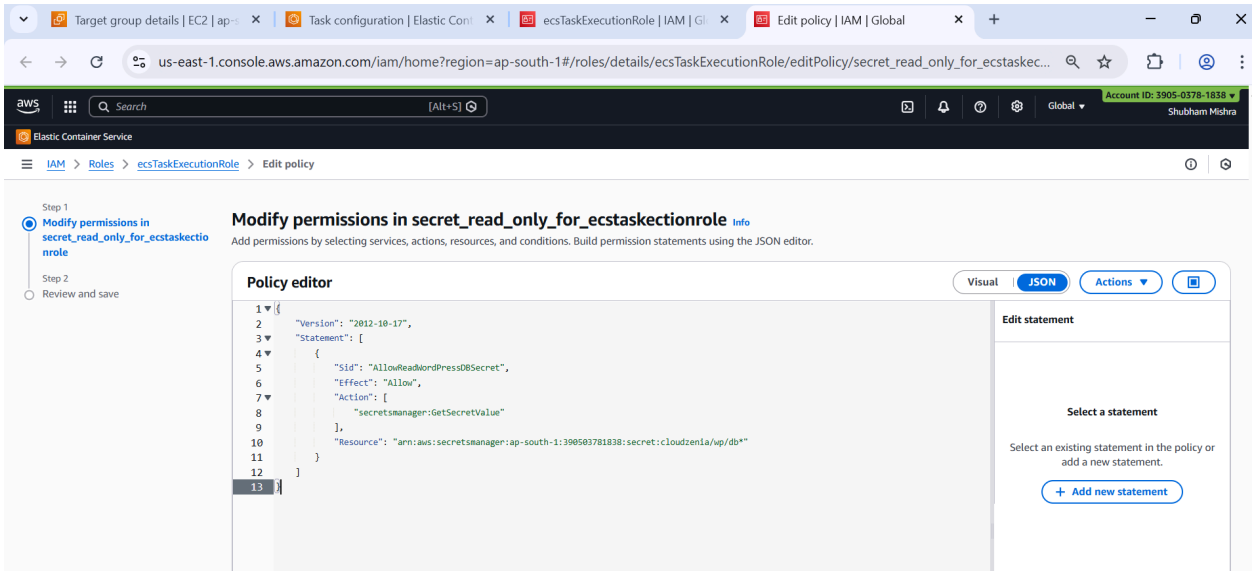
Step 30) Now we can see two task for task definition ie Microservicre and wordpress created successfully.



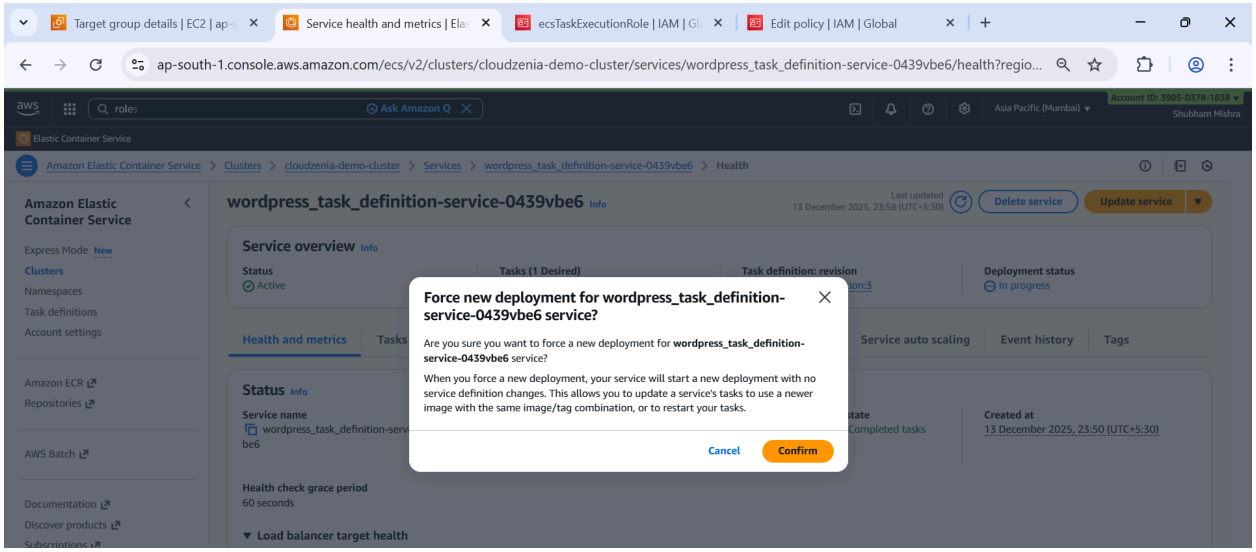
Step 31) Now our task for wordpress is failing due to missing permission ofr ecstaskexecutionrole.



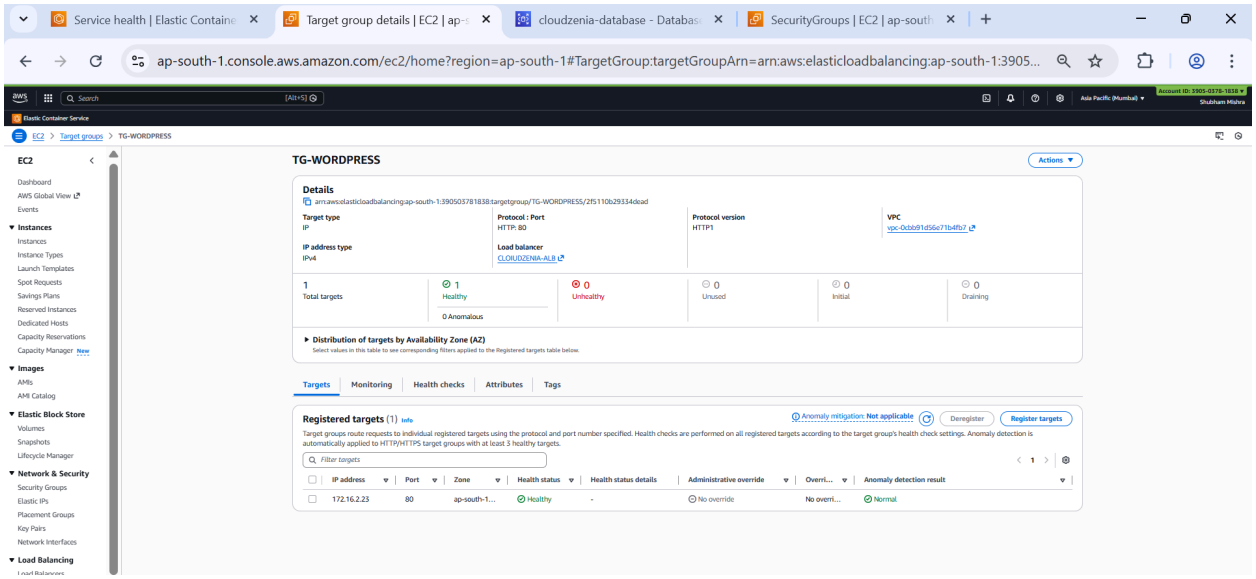
Step 31) Now we will create inpolicy to allow read secrets in secrets manager and attach it to ecstaskexecutionrole.



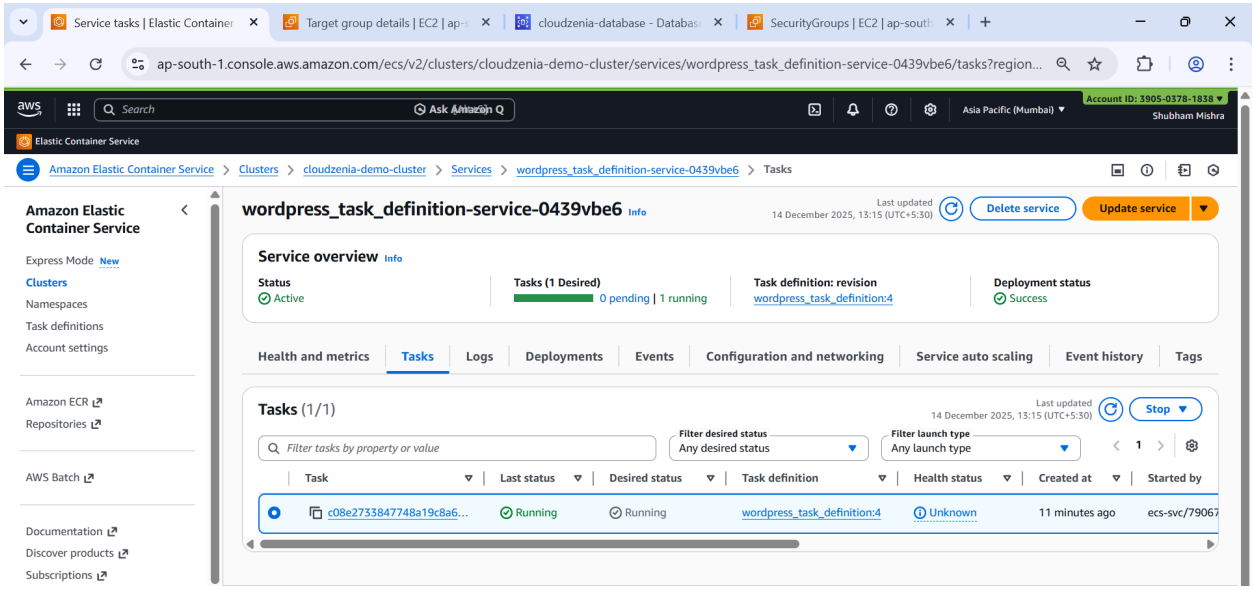
Step 32) Now we will force new deployment after updating role for wordpress task definition.



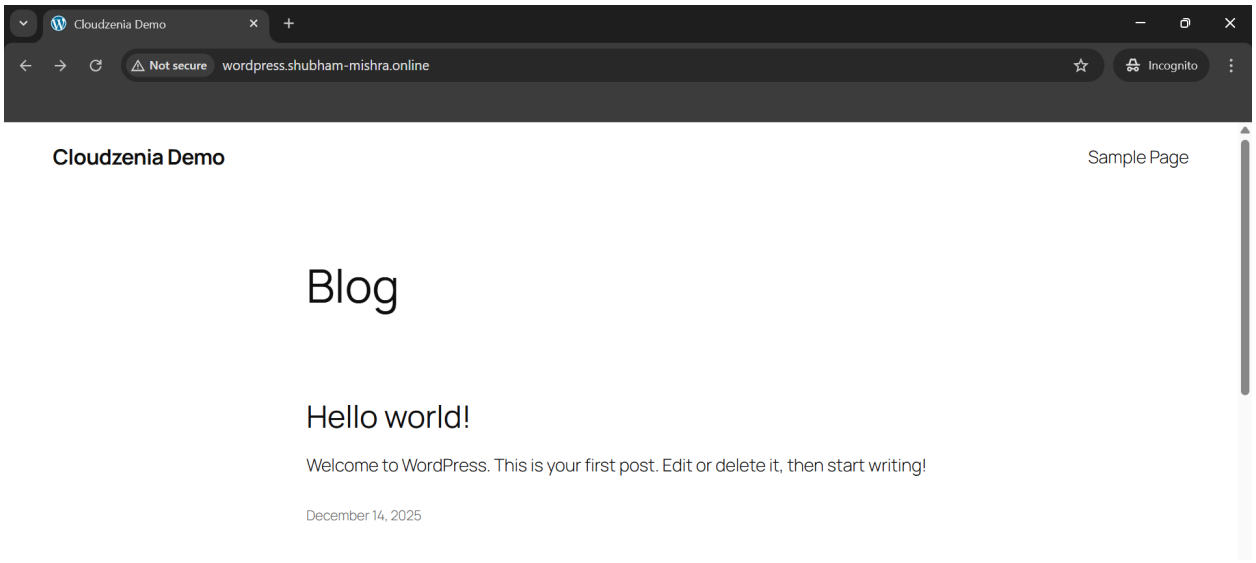
Step 33) Now we can see the target group for worpdress is in healthy state



Step 34) Now we can see healthy container task for wordpress

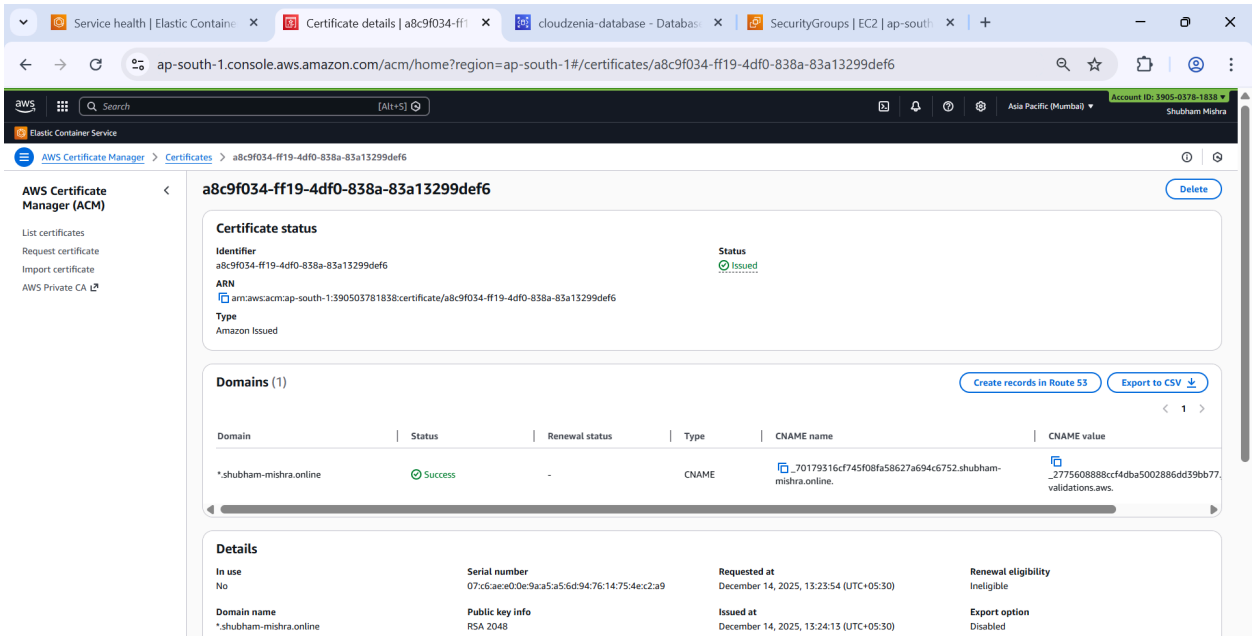


Step 35) Now lets verify worpress.shubham-mishra.online for another task definition



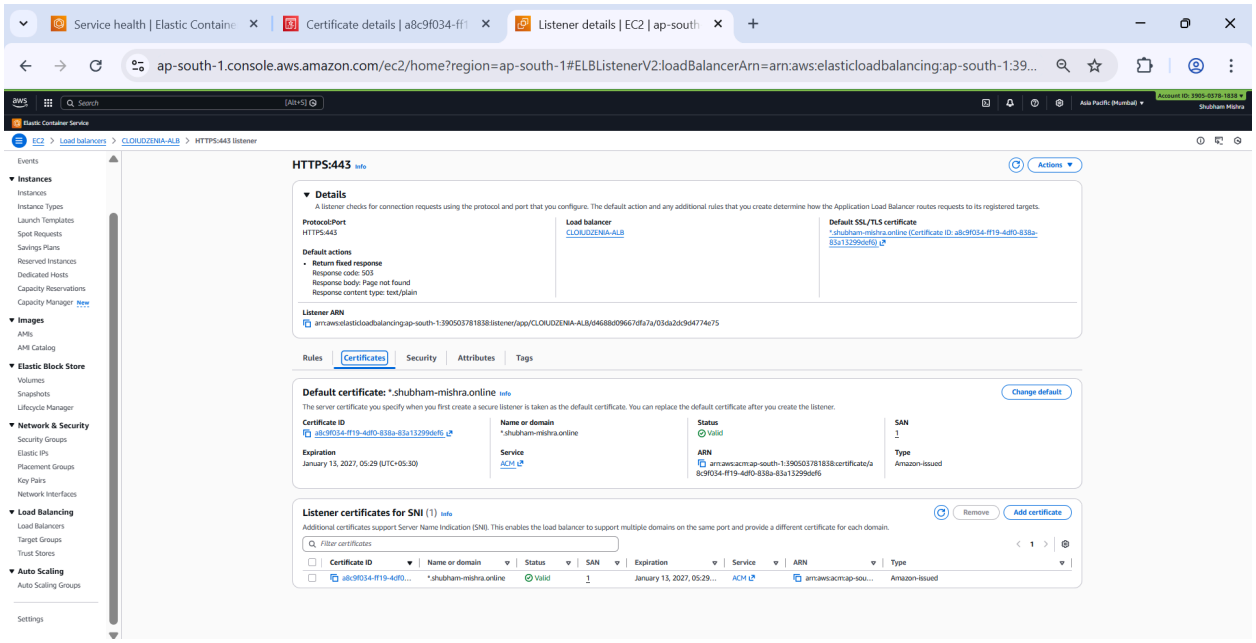
This confirms that wordpress container is running and is accessible by alb through route53 domain.

Step 31) Now Lets add SSL certificate to our domains and make sure only it resolves over https.

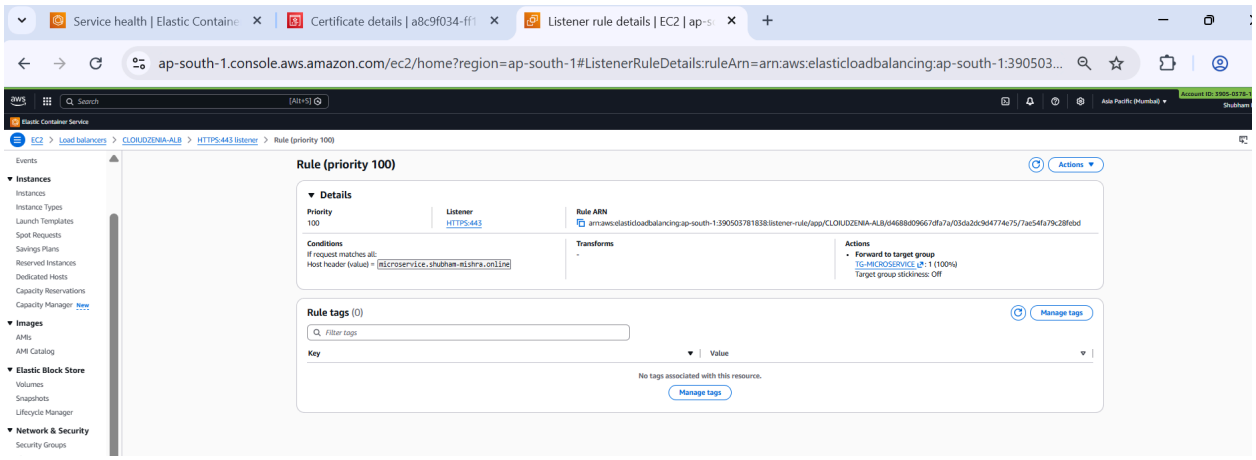


We have successfully added ssl certificate in ACM and configured into Route53 of another account.

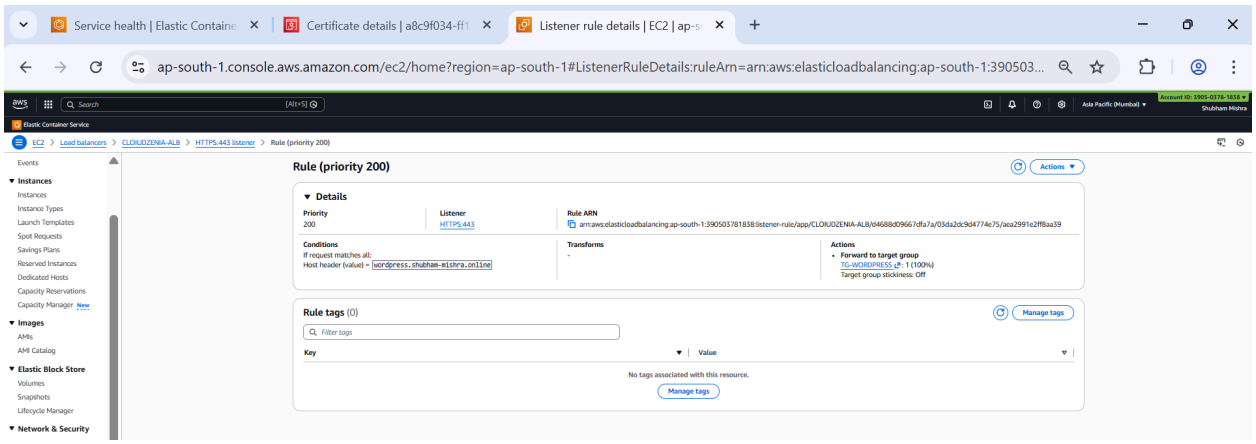
Step 32) Now we will add 443 listener in alb to use SSL certificate.



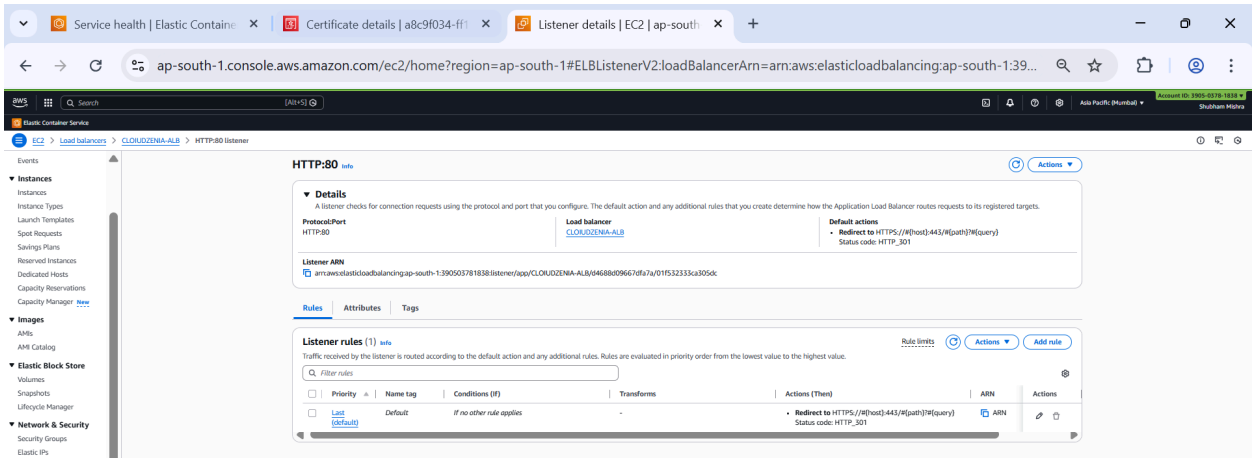
Step 33) Now we will add rule for TG-MICROSERVICE target group if host headed is microservice.shubham-mishra.online then forward to TG-MICROSERVICE.



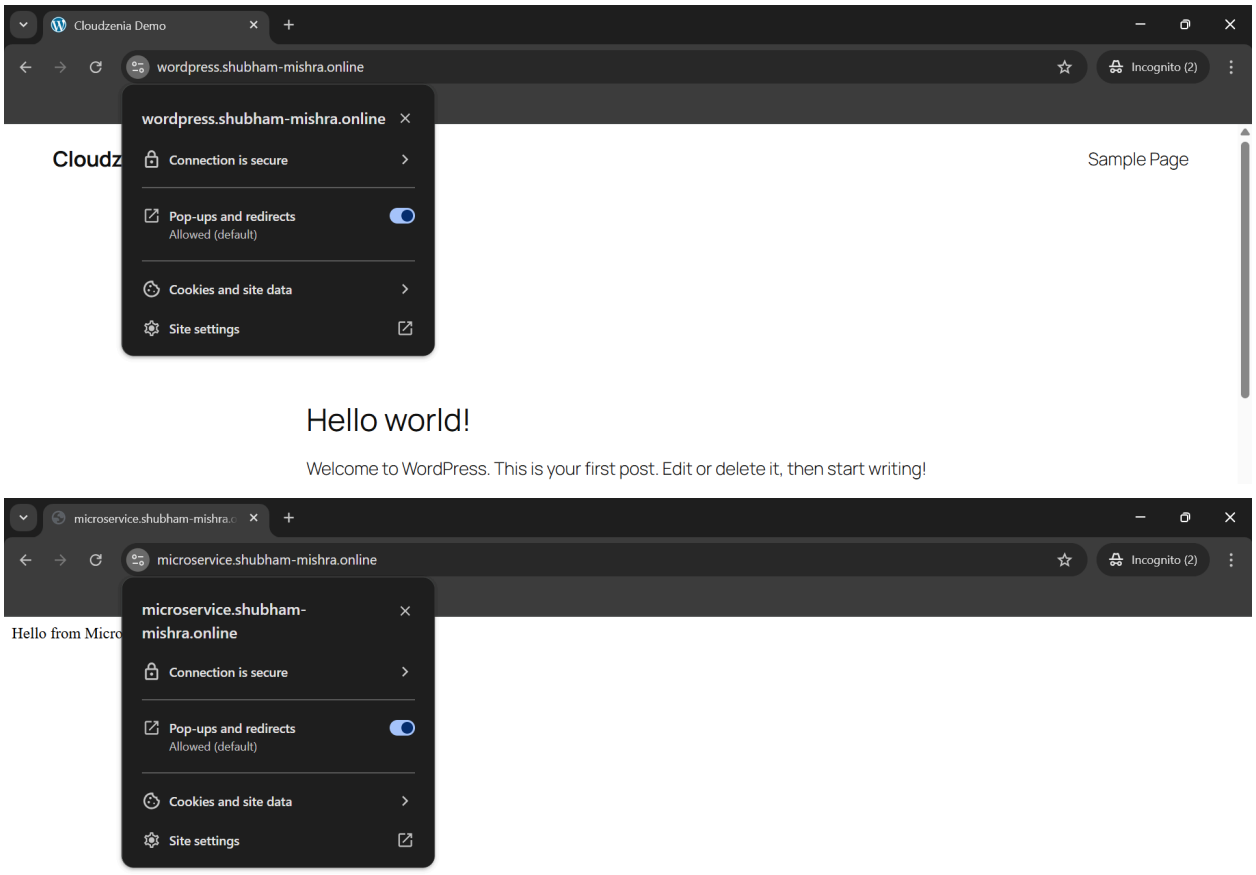
Step 34) Now we will add rule for TG-WORDPRESS target group if host headed is wordpress.shubham-mishra.online then forward to TG-WORDPRESS.



Step 35) Now we will add redirect rule in http if any request on http then redirect to https 443



Step 36) Now we can see ssl certificate is working on our domain



Step 37) Now we can see that even http request is being redirected to https.

```
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>curl http://microservice.shubham-mishra.online
<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
</body>
</html>

C:\Users\UIPL-Admin>curl http://wordpress.shubham-mishra.online
<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
</body>
</html>

C:\Users\UIPL-Admin>
```

```
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>
C:\Users\UIPL-Admin>curl -L https://microservice.shubham-mishra.online
Hello from Microservice
C:\Users\UIPL-Admin>
```

```
C:\Users\UIPL-Admin>curl https://wordpress.shubham-mishra.online
<!DOCTYPE html>
<html lang="en-US">
<head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
<meta name='robots' content='noindex, nofollow' />
<title>Cloudzenia Demo</title>
<link rel="alternate" type="application/rss+xml" title="Cloudzenia Demo &raquo; Feed" href="https://wordpre
eed/" />
<link rel="alternate" type="application/rss+xml" title="Cloudzenia Demo &raquo; Comments Feed" href="https:
```

Project Outcome

- **Successfully designed and implemented a production-ready AWS architecture using ECS Fargate, Application Load Balancer,**

RDS, Secrets Manager, IAM, ACM, and Route 53.

- **Deployed two containerized services:**
    - **WordPress application backed by Amazon RDS (MySQL) running in private subnets.**
    - **Custom Node.js microservice responding with *"Hello from Microservice"*, fully containerized using Docker.**
- **Ensured secure architecture by:**
    - **Running ECS tasks and RDS instances in private subnets.**
    - **Restricting database access using least-privilege security groups.**
    - **Storing and consuming database credentials securely via AWS Secrets Manager.**
    - **Granting ECS access through IAM task roles without hard-coding secrets.**
- **Implemented high availability and scalability:**
    - **Application Load Balancer deployed in public subnets.**
    - **Traffic routed using host-based routing to separate target groups.**
    - **Enabled ECS Service Auto Scaling based on CPU and memory utilization.**
- **Configured secure domain access:**

    - **Integrated custom subdomains:**
        - `wordpress.<domain>`
        - `microservice.<domain>`
    - **Implemented HTTPS using AWS Certificate Manager (ACM).**
    - **Enforced HTTP → HTTPS redirection at the ALB level.**
- **Achieved cost-optimized design:**
    - **ECS services can scale down to zero tasks when not in use.**
    - **NAT Gateway and ECS services can be safely stopped and re-enabled without configuration changes.**
    - **RDS endpoint remains unchanged during stop/start operations.**
- **Validated successful deployment by:**
    - **Accessing both applications through custom HTTPS domains.**
    - **Verifying healthy target group status and successful ALB routing.**
    - **Confirming secure database connectivity from WordPress via Secrets Manager.**