AWS Cloud Infrastructure Deployment Project

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

Agri

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

▶ **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

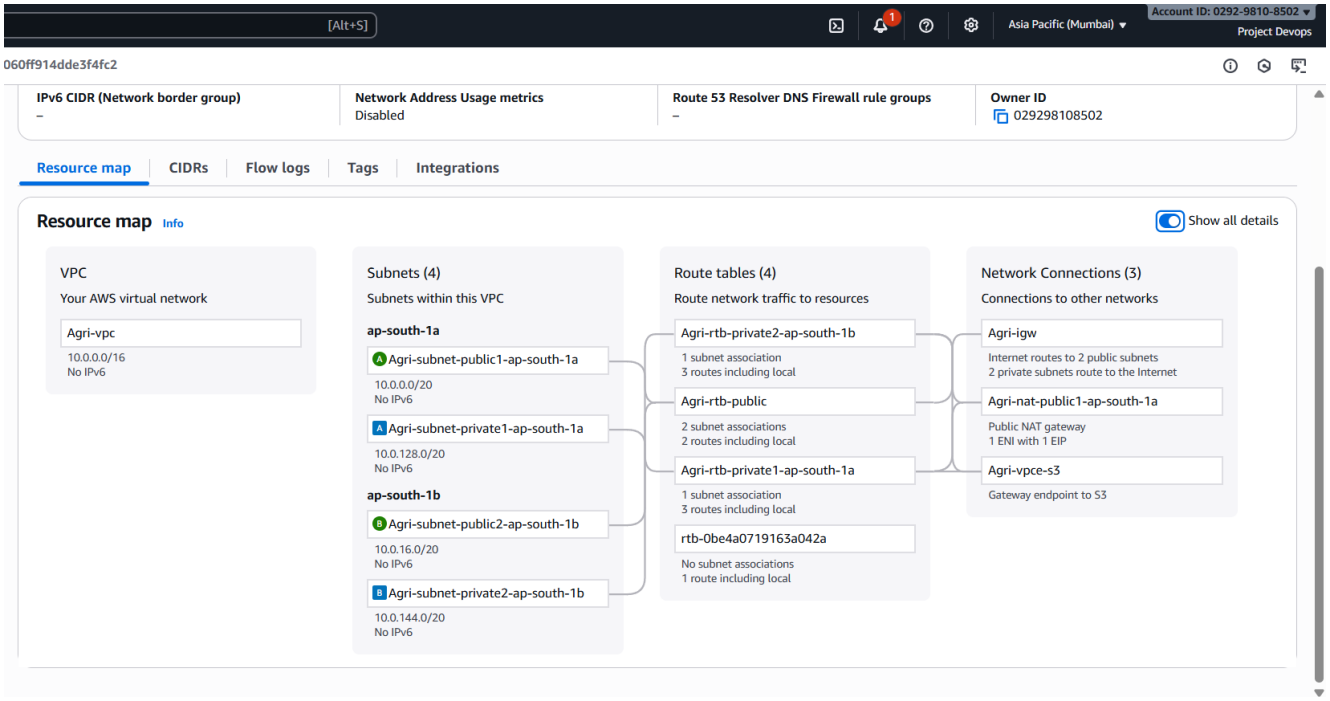
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None | S3 Gateway

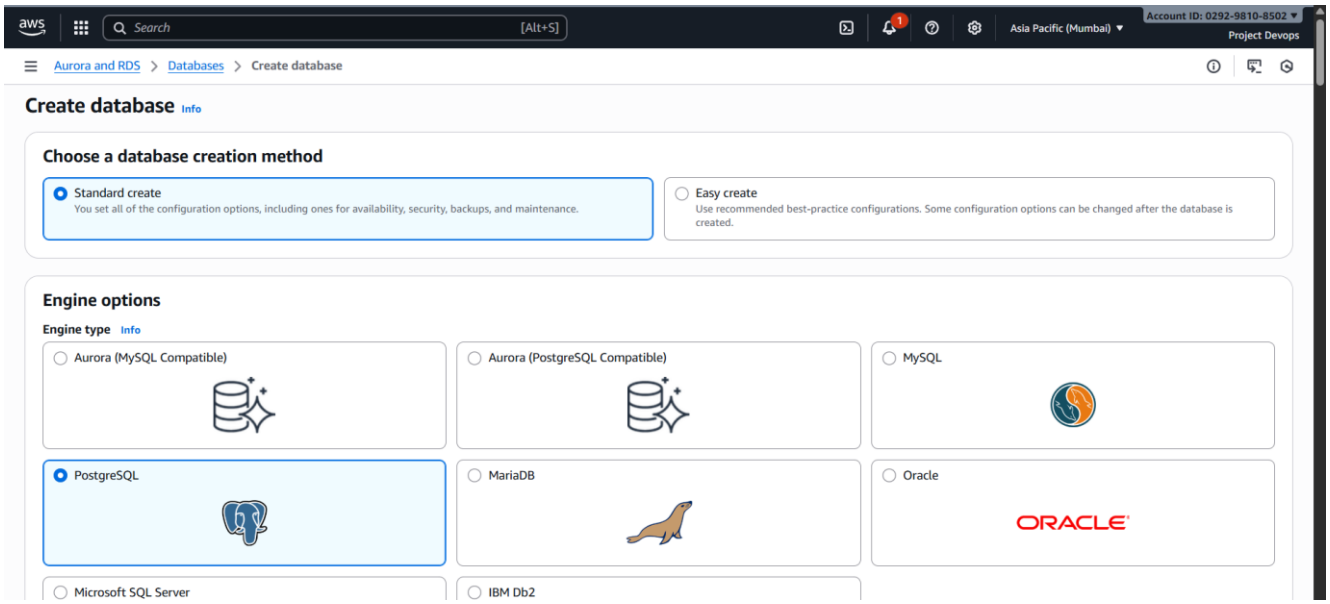
DNS options Info

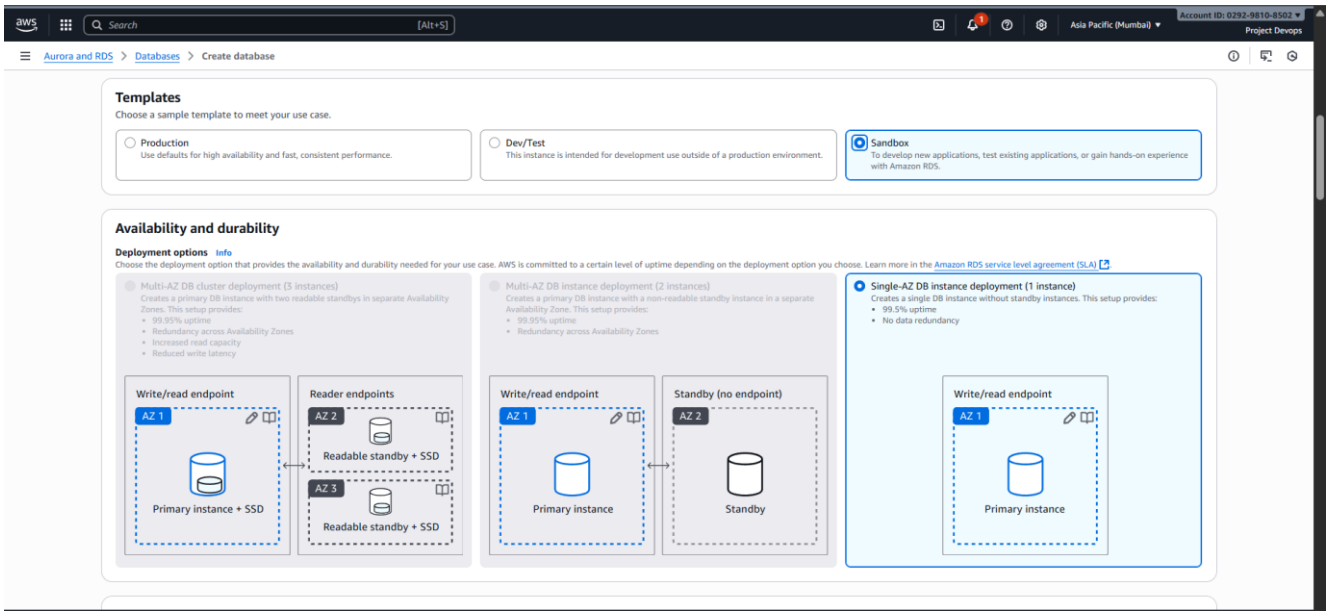
- ☑ Enable DNS hostnames
- ☑ Enable DNS resolution

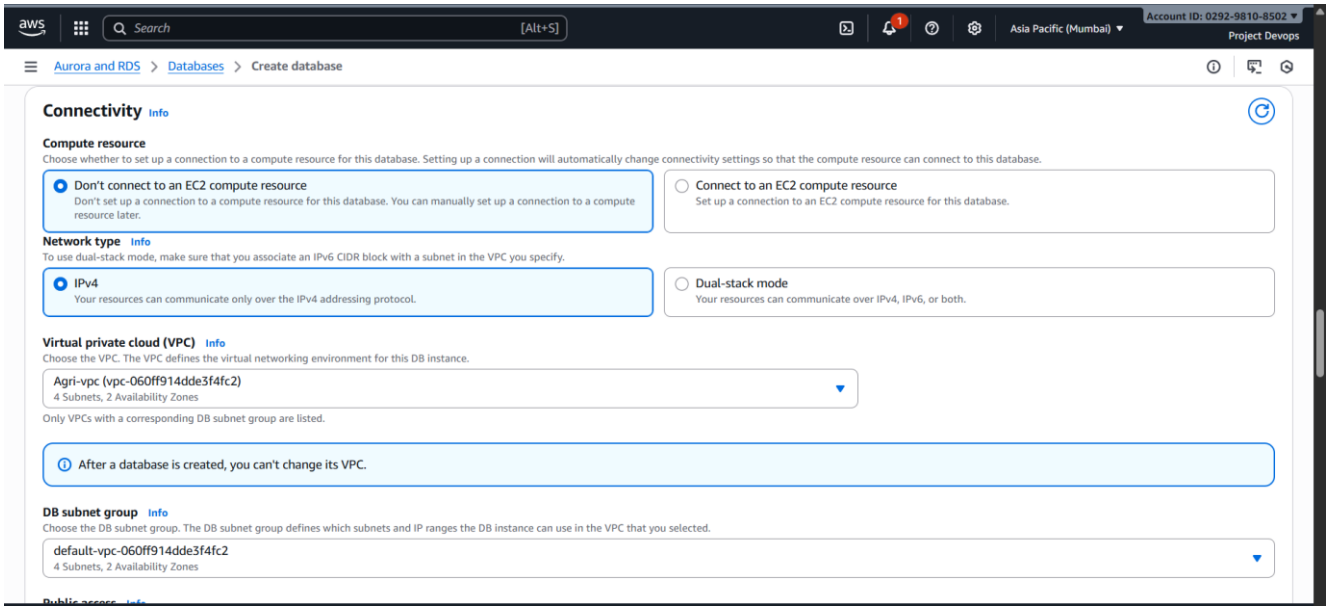
▶ **Additional tags**

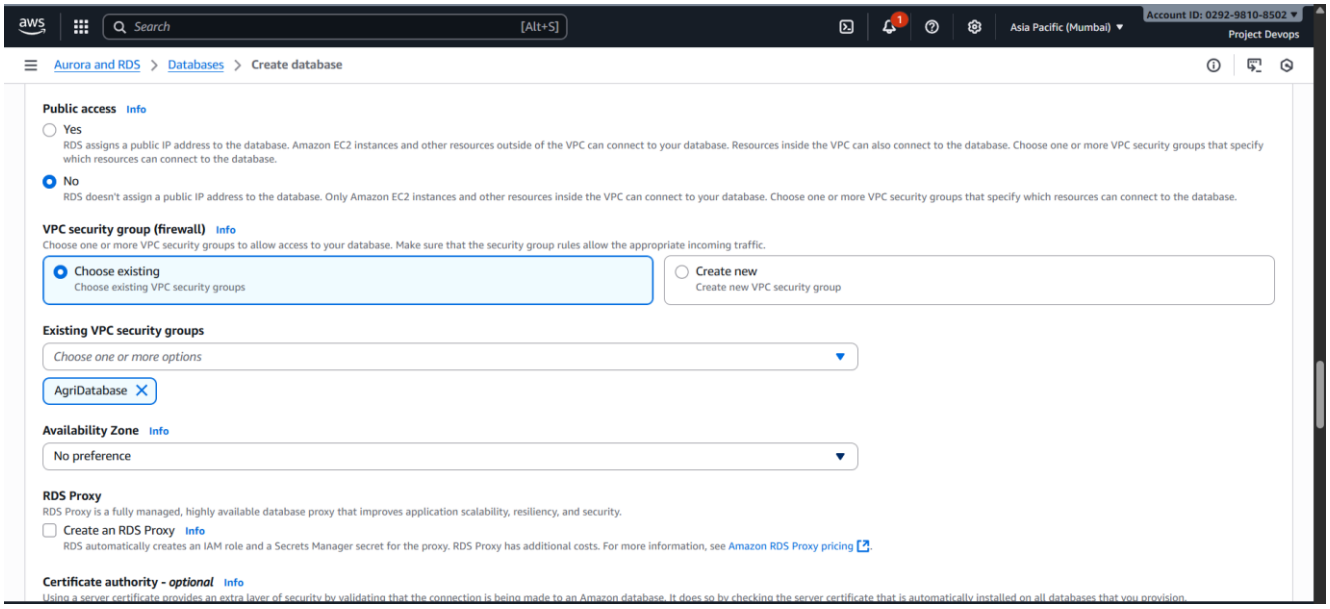

[Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
060ff914dde3f4fc2
IPv6 CIDR (Network border group)
–
Network Address Usage metrics
Disabled
Route 53 Resolver DNS Firewall rule groups
–
Owner ID
029298108502
Resource map
CIDRs
Flow logs
Tags
Integrations
Resource map Info
Show all details
VPC
Your AWS virtual network
Agri-vpc
10.0.0.0/16
No IPv6
Subnets (4)
Subnets within this VPC
ap-south-1a
Agri-subnet-public1-ap-south-1a
10.0.0.0/20
No IPv6
Agri-subnet-private1-ap-south-1a
10.0.128.0/20
No IPv6
ap-south-1b
Agri-subnet-public2-ap-south-1b
10.0.16.0/20
No IPv6
Agri-subnet-private2-ap-south-1b
10.0.144.0/20
No IPv6
Route tables (4)
Route network traffic to resources
Agri-rtb-private2-ap-south-1b
1 subnet association
3 routes including local
Agri-rtb-public
2 subnet associations
2 routes including local
Agri-rtb-private1-ap-south-1a
1 subnet association
3 routes including local
rtb-0be4a0719163a042a
No subnet associations
1 route including local
Network Connections (3)
Connections to other networks
Agri-igw
Internet routes to 2 public subnets
2 private subnets route to the Internet
Agri-nat-public1-ap-south-1a
Public NAT gateway
1 ENI with 1 EIP
Agri-vpce-s3
Gateway endpoint to S3

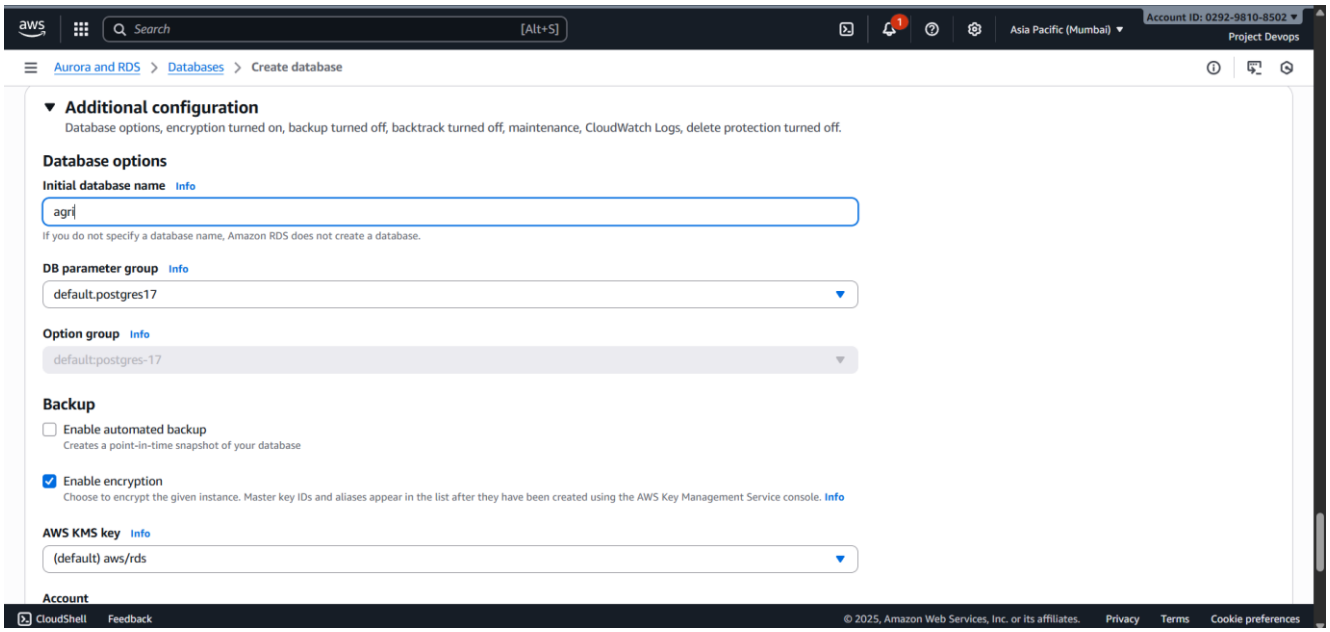
Database:


Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
Aurora and RDS > Databases > Create database
Create database Info
Choose a database creation method
Standard create
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
Microsoft SQL Server
IBM Db2

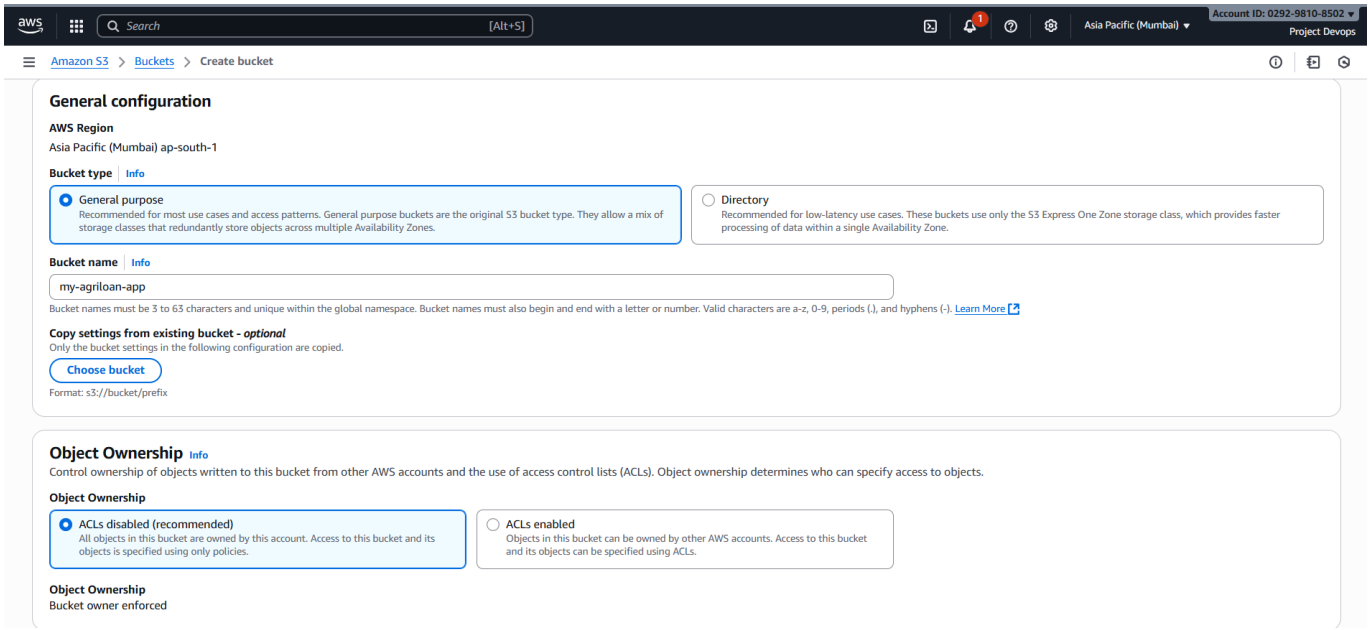

Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
Aurora and RDS > Databases > Create database
Templates
Choose a sample template to meet your use case.
Production
Use defaults for high availability and fast, consistent performance.
Dev/Test
This instance is intended for development use outside of a production environment.
Sandbox
To develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Availability and durability
Deployment options Info
Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the Amazon RDS service level agreement (SLA)
Multi-AZ DB cluster deployment (3 instances)
Multi-AZ DB instance deployment (2 instances)
Single-AZ DB instance deployment (1 instance)
Creates a single DB instance without standby instances. This setup provides:
99.95% uptime
No data redundancy
Write/read endpoint
Reader endpoints
Standby (no endpoint)
AZ 1
AZ 2
AZ 3
Primary instance + SSD
Readable standby + SSD
Primary instance
Standby

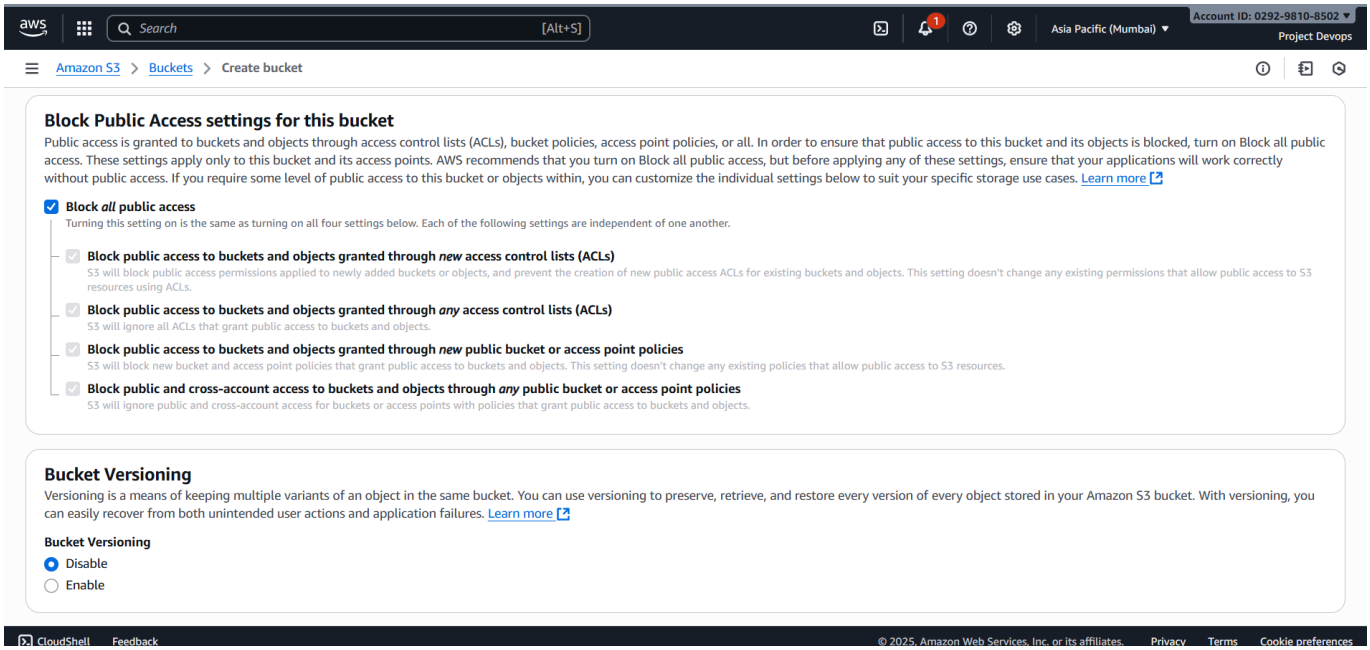

Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
Aurora and RDS > Databases > Create database
Connectivity Info
Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity 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.
Network type Info
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.
IPv4
Your resources can communicate only over the IPv4 addressing protocol.
Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.
Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.
Agri-vpc (vpc-060ff914dde3f4fc2)
4 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.
default-vpc-060ff914dde3f4fc2
4 Subnets, 2 Availability Zones

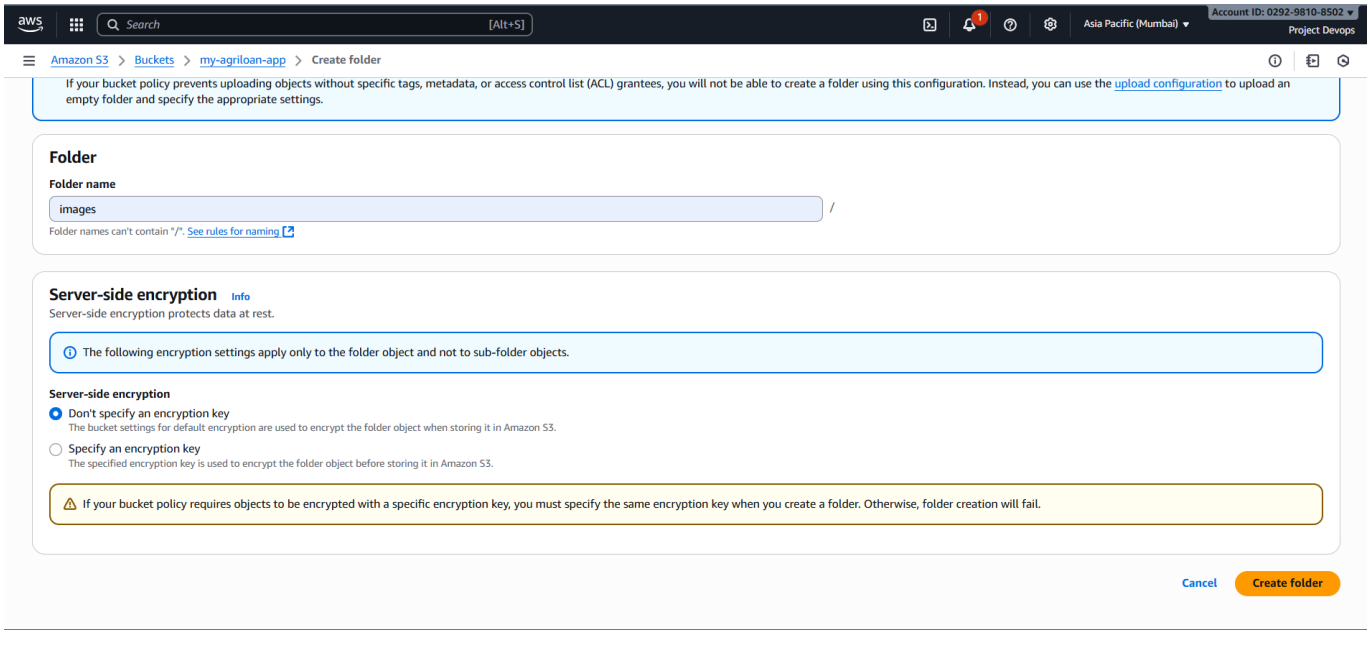

Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
Aurora and RDS > Databases > Create database
Public access Info
Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside 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.
VPC security group (firewall) Info
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.
Choose existing
Choose existing VPC security groups
Create new
Create new VPC security group
Existing VPC security groups
Choose one or more options
AgriDatabase
Availability Zone Info
No preference
RDS Proxy
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.
Create an RDS Proxy Info
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see Amazon RDS Proxy pricing
Certificate authority - optional Info


Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
Aurora and RDS > Databases > Create database
Additional configuration
Database options, encryption turned on, backup turned off, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.
Database options
Initial database name Info
agri
If you do not specify a database name, Amazon RDS does not create a database.
DB parameter group Info
default.postgres17
Option group Info
default:postgres-17
Backup
Enable automated backup
Creates a point-in-time snapshot of your database
Enable encryption
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. Info
AWS KMS key Info
(default) aws/rds
Account
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

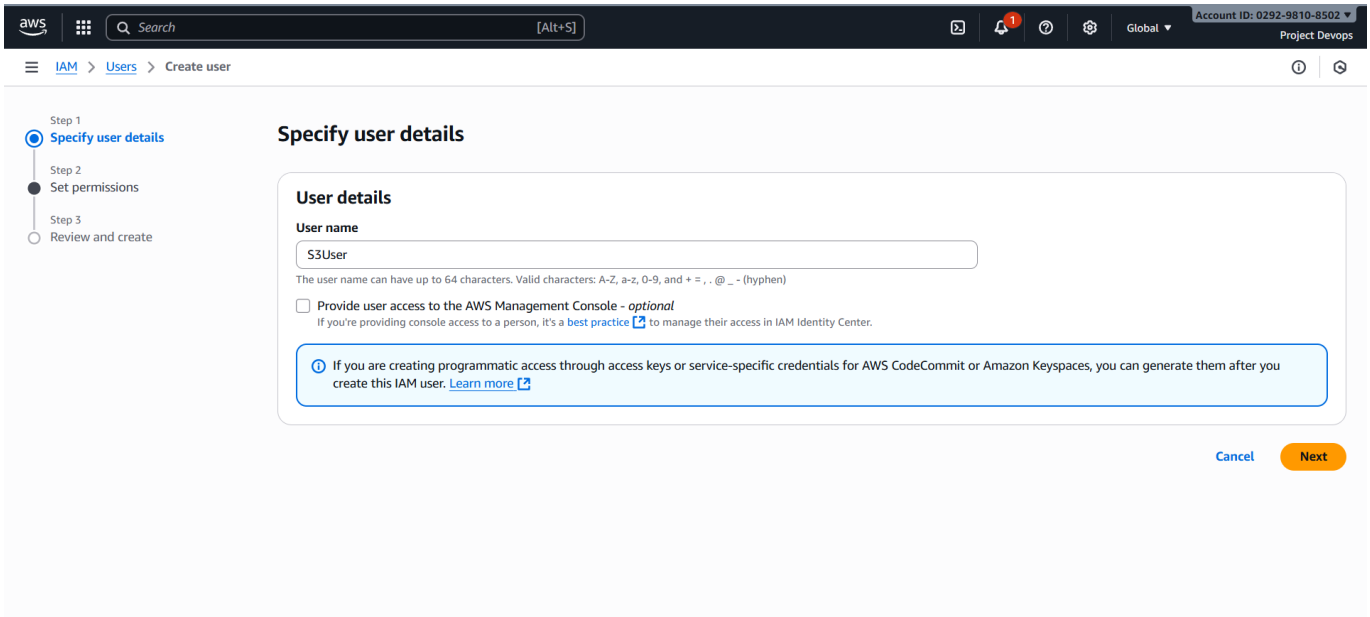
S3

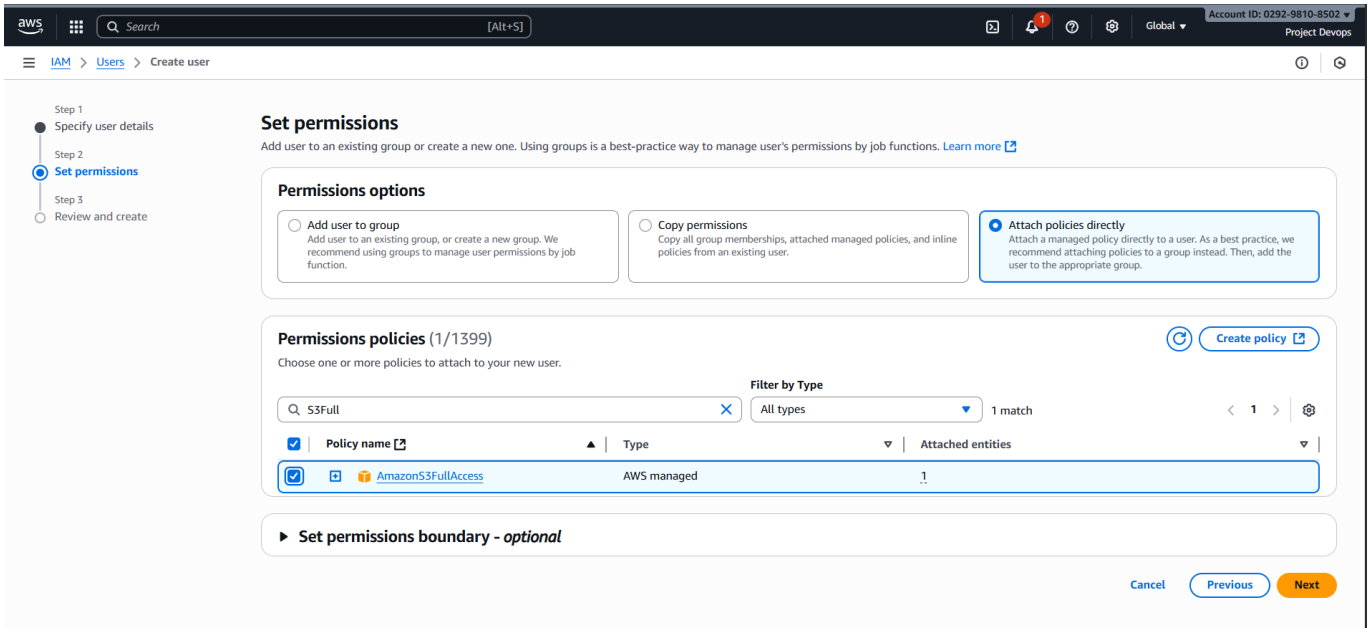

Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
Amazon S3 > Buckets > Create bucket
General configuration
AWS Region
Asia Pacific (Mumbai) ap-south-1
Bucket type Info
General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.
Bucket name Info
my-agriloan-app
Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). Learn More
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix
Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
Object Ownership
ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.
Object Ownership
Bucket owner enforced

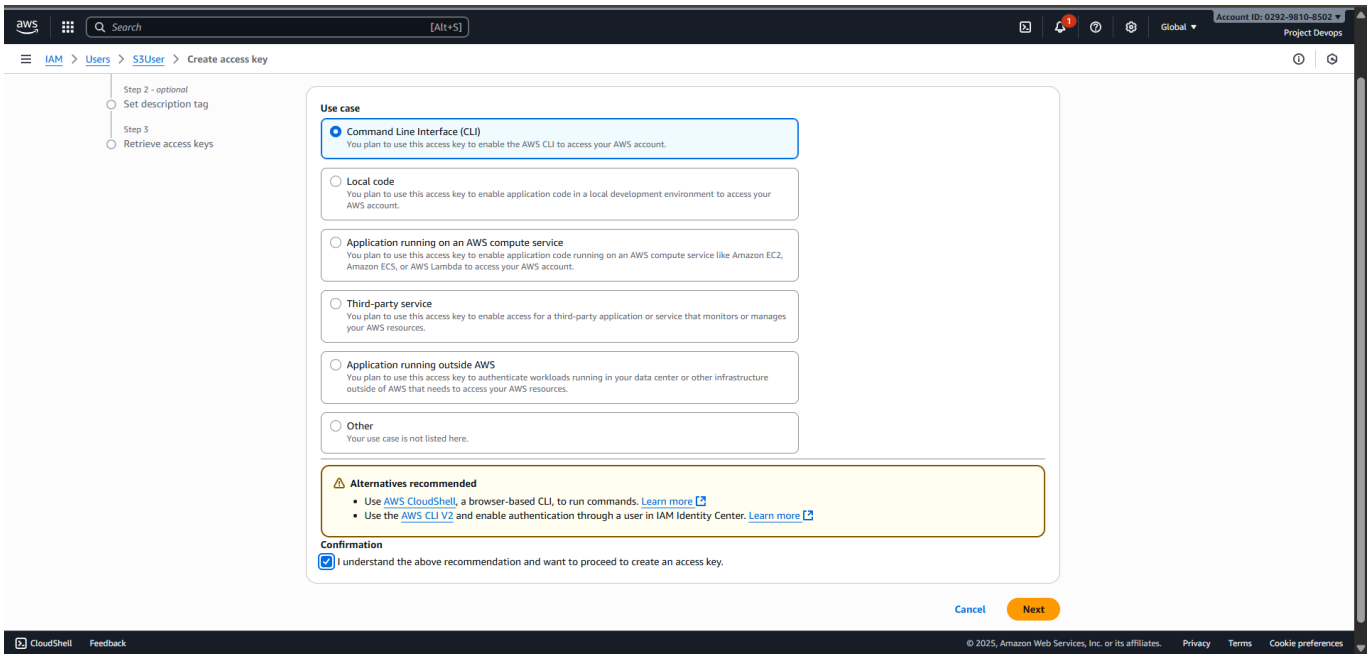

Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
Amazon S3 > Buckets > Create bucket
Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.
Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more
Bucket Versioning
Disable
Enable
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
Amazon S3 > Buckets > my-agriloan-app > Create folder
If your bucket policy prevents uploading objects without specific tags, metadata, or access control list (ACL) grantees, you will not be able to create a folder using this configuration. Instead, you can use the upload configuration to upload an empty folder and specify the appropriate settings.
Folder
Folder name
images
/
Folder names can't contain "/". See rules for naming
Server-side encryption Info
Server-side encryption protects data at rest.
The following encryption settings apply only to the folder object and not to sub-folder objects.
Server-side encryption
Don't specify an encryption key
The bucket settings for default encryption are used to encrypt the folder object when storing it in Amazon S3.
Specify an encryption key
The specified encryption key is used to encrypt the folder object before storing it in Amazon S3.
If your bucket policy requires objects to be encrypted with a specific encryption key, you must specify the same encryption key when you create a folder. Otherwise, folder creation will fail.
Cancel
Create folder

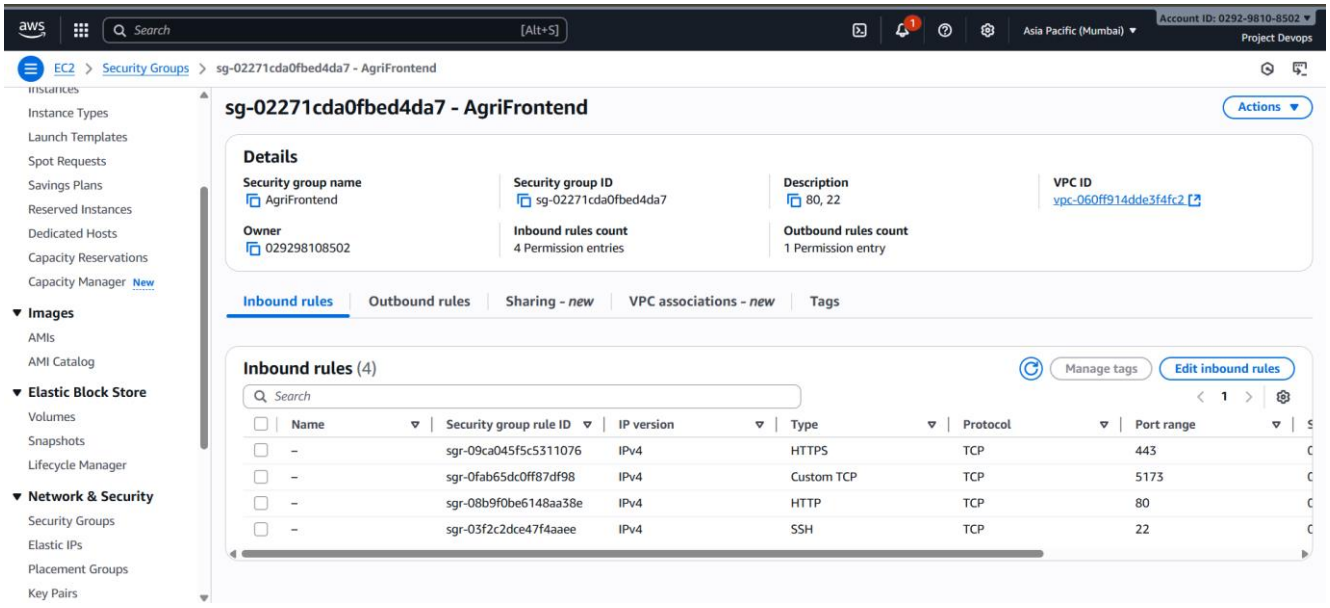
IAM

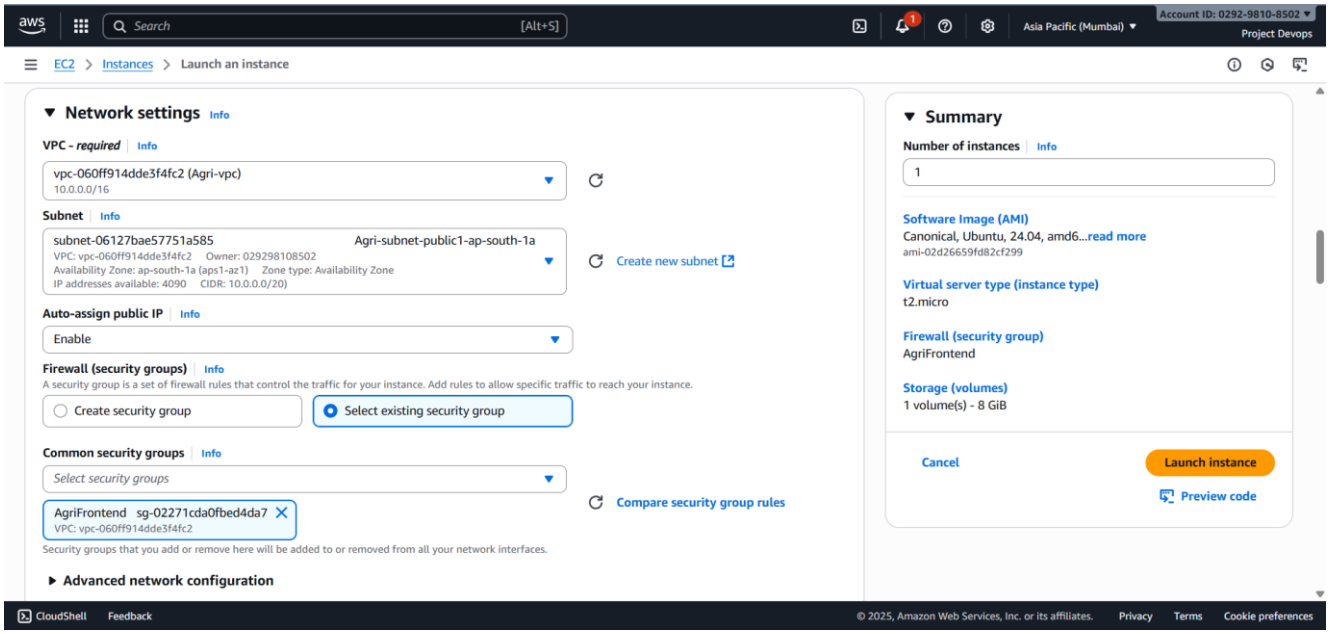

Search [Alt+S]
Global
Account ID: 0292-9810-8502
Project Devops
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Specify user details
User details
User name
S3User
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)
Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more
Cancel
Next


Search [Alt+S]
Global
Account ID: 0292-9810-8502
Project Devops
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Set permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
Permissions policies (1/1399)
Choose one or more policies to attach to your new user.
Create policy
Filter by Type
S3Full
All types
1 match
Policy name
Type
Attached entities
AmazonS3FullAccess
AWS managed
1
Set permissions boundary - optional
Cancel
Previous
Next


Search [Alt+S]
Global
Account ID: 0292-9810-8502
Project Devops
IAM > Users > S3User > Create access key
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Use case
Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.
Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.
Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
Other
Your use case is not listed here.
Alternatives recommended
Use AWS CloudShell, a browser-based CLI, to run commands. Learn more
Use the AWS CLI V2 and enable authentication through a user in IAM Identity Center. Learn more
Confirmation
I understand the above recommendation and want to proceed to create an access key.
Cancel
Next
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

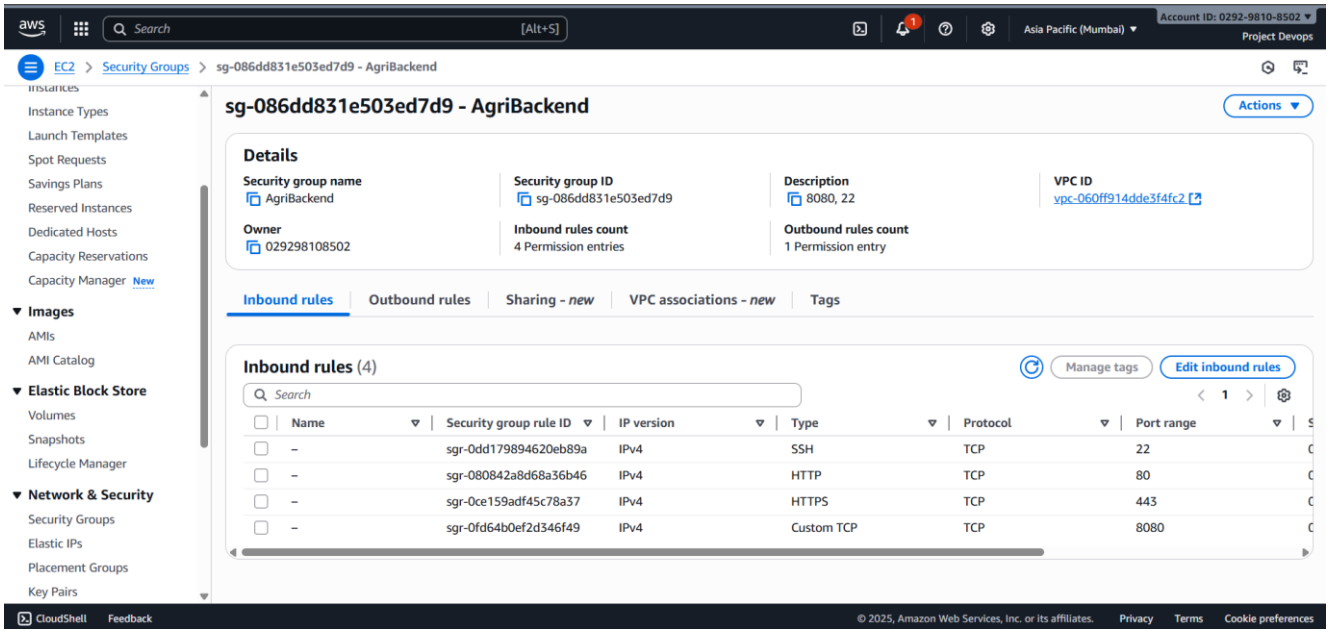
Frontend Instance

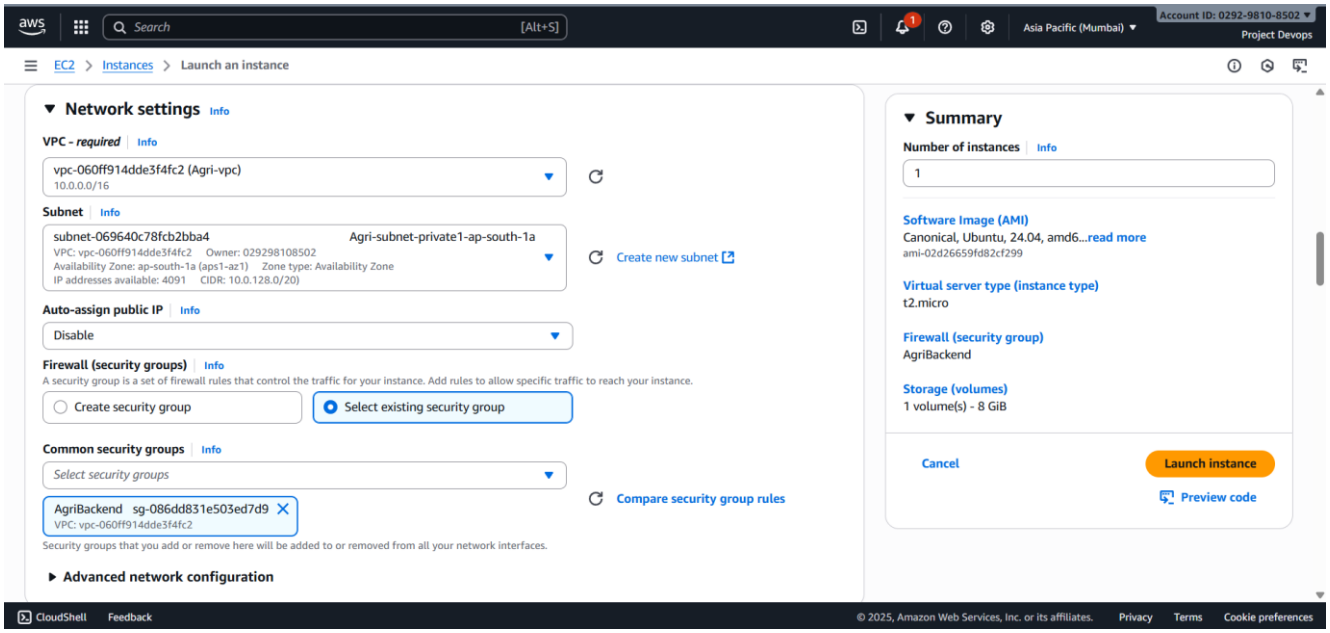





```
#!/bin/bash

set -e

sudo apt update -y

sudo apt upgrade -y

sudo apt install -y curl

curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -

sudo apt install -y nodejs

sudo apt install npm -y
```

Backend Instance






```
#!/bin/bash

set -e

sudo apt update -y

sudo apt upgrade -y

sudo apt install openjdk-17-jdk -y

sudo apt install maven -y
```

Send the pem key to public instance(Frontend) for access the private instance(Backend)

C:\Users\Anbarasu\Downloads>**scp -i agri.pem agri.pem ubuntu@13.235.254.132:/home/ubuntu/**

```
C:\Users\Anbarasu\Downloads>scp -i agri.pem agri.pem ubuntu@13.235.254.132:/home/ubuntu/
The authenticity of host '13.235.254.132 (13.235.254.132)' can't be established.
ED25519 key fingerprint is SHA256:sS2lLUYAhvuIpnqsq/bQR+afGBgIkNE4ifYptBAmBbo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '13.235.254.132' (ED25519) to the list of known hosts.
agri.pem                                      100% 1674     3.6KB/s   00:00
```

C:\Users\Anbarasu\Downloads>**ssh -i agri.pem ubuntu@13.235.254.132**

```
ubuntu@ip-10-0-5-128:~$ node -v
v18.19.1
ubuntu@ip-10-0-5-128:~$ npm -v
9.2.0
ubuntu@ip-10-0-5-128:~$ ls
agri.pem
ubuntu@ip-10-0-5-128:~$
```

ubuntu@ip-10-0-4-190:~$ **sudo apt install -y curl nginx**

ubuntu@ip-10-0-4-190:/$ **sudo mkdir -p /var/www/frontend**

ubuntu@ip-10-0-4-190:/$ **sudo chown ubuntu:ubuntu /var/www/frontend**

ubuntu@ip-10-0-4-190:/$ **cd /var/www**

ubuntu@ip-10-0-4-190:/var/www$ **git clone https://github.com/Anbarasu-A-N/Agriculture_Frontend.git frontend**

ubuntu@ip-10-0-4-190:/var/www$ **cd frontend**

ubuntu@ip-10-0-4-190:/var/www/frontend$ **sudo npm ci --force**

ubuntu@ip-10-0-5-128:/var/www/frontend$ **cd src**

ubuntu@ip-10-0-5-128:/var/www/frontend/src$ **sudo nano Config.jsx**

// ✅ Paste your frontend public IP or domain

export const BASE_URL = 'http://13.232.140.236/api';

ubuntu@ip-10-0-4-190:/var/www/frontend$ **cd .. && npm run build**

ubuntu@ip-10-0-5-128:/var/www/frontend$ **ls /var/www/html/**

index.nginx-debian.html

ubuntu@ip-10-0-5-128:/var/www/frontend$ **sudo rm -rf /var/www/html/***

ubuntu@ip-10-0-5-128:/var/www/frontend$ **ls /var/www/html/**

ubuntu@ip-10-0-4-190:/var/www/frontend$ **sudo cp -r dist/* /var/www/html/**

ubuntu@ip-10-0-5-128:/var/www/frontend$ **ls /var/www/html/**

Allsmart.png assets index.html

ubuntu@ip-10-0-4-190:/var/www/frontend$ **sudo ls /etc/nginx/sites-enabled/**

default

ubuntu@ip-10-0-4-190:/var/www/frontend$ **sudo nano /etc/nginx/sites-available/default**

Paste this below para

```
server {
    listen 80;
    server_name _;

    root /var/www/html;
    index index.html;

    # serve static files
    location / {
        try_files $uri /index.html;
    }

    # proxy API requests to backend private IP
    location /api/ {
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_pass http://10.0.131.108:8080/;
        proxy_read_timeout 90;
    }

    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-Content-Type-Options "nosniff";
}
```

ubuntu@ip-10-0-5-128:/var/www/frontend$ **sudo nginx -t**

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok

nginx: configuration file /etc/nginx/nginx.conf test is successful

ubuntu@ip-10-0-5-128:/var/www/frontend$ **sudo systemctl reload nginx**

ubuntu@ip-10-0-5-128:/var/www/frontend$ **sudo systemctl enable nginx**

Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.

Executing: /usr/lib/systemd/systemd-sysv-install enable nginx

ubuntu@ip-10-0-5-128:/var/www/frontend$ **sudo systemctl restart nginx**

ubuntu@ip-10-0-5-128:/var/www/frontend$ **cd ~**

ubuntu@ip-10-0-5-128:~$ **ls**

agri.pem

ubuntu@ip-10-0-5-128:~$ **chmod 700 agri.pem**

Login into Backend using private IP

ubuntu@ip-10-0-5-128:~$ **ssh -i agri.pem ubuntu@10.0.131.108**

ubuntu@ip-10-0-131-108:~$ **java --version**

ubuntu@ip-10-0-131-108:~$ **mvn -v**

ubuntu@ip-10-0-131-108:~$ **cd /opt**

ubuntu@ip-10-0-131-108:/opt$ **sudo mkdir -p agriculture-backend**

ubuntu@ip-10-0-131-108:/opt$ **sudo chown ubuntu:ubuntu /opt/agriculture-backend**

ubuntu@ip-10-0-131-108:/opt$ **cd /opt/agriculture-backend**

ubuntu@ip-10-0-131-108:/opt/agriculture-backend$ **git clone https://github.com/Anbarasu-A-N/Agriculture_Backend.git .**

ubuntu@ip-10-0-131-108:/opt/agriculture-backend$ **sudo nano src/main/resources/application.properties**

Change this Configuration

RDS Endpoint, Database name, username, password

spring.datasource.url=jdbc:postgresql://project.cnk8gomow5fx.ap-south-1.rds.amazonaws.com/agri

spring.datasource.username=$ MENTION_YOUR_USERNAME_HERE

spring.datasource.password=MENTION_YOUR_PASSWORD_HERE

spring.datasource.driver-class-name=org.postgresql.Driver

SMTP Server Configuration

SMTP username and password

spring.mail.host=smtp.gmail.com

spring.mail.port=587

spring.mail.properties.mail.smtp.auth=true

spring.mail.properties.mail.smtp.starttls.enable=true

spring.mail.username= MENTION_YOUR_USERNAME_HERE

spring.mail.password=MENTION_YOUR_PASSWORD_HERE

Frontend Public IP, Backend Private IP

allowed.origins=http://13.235.254.132:80,http://13.235.254.132:80/**

jwt.server.url=http://10.0.131.108:8080

S3 Bucket, Folder name, Access key, Secret Key

AWS S3 Configuration

aws.s3.bucket=MENTION_YOUR_S3_NAME _HERE

aws.s3.folder=MENTION_YOUR_S3_FOLDER_NAME_HERE

aws.access.key.id=MENTION_YOUR_SECRET_ACCESS_KEY_HERE

aws.secret.access.key=MENTION_YOUR_SECRET_ACCESS_KEY_HERE

aws.region=us-east-1

Now Test

ubuntu@ip-10-0-131-108:/opt/agriculture-backend$ **mvn clean install package**

ubuntu@ip-10-0-131-108:/opt/agriculture-backend$ **java -jar target/app.jar**

 # Find the port and kill

sudo ss -tulnp | grep 8080

kill -9 <pid>

Simple Background Run (temporary)

If you just want it to keep running after logout (not on reboot):

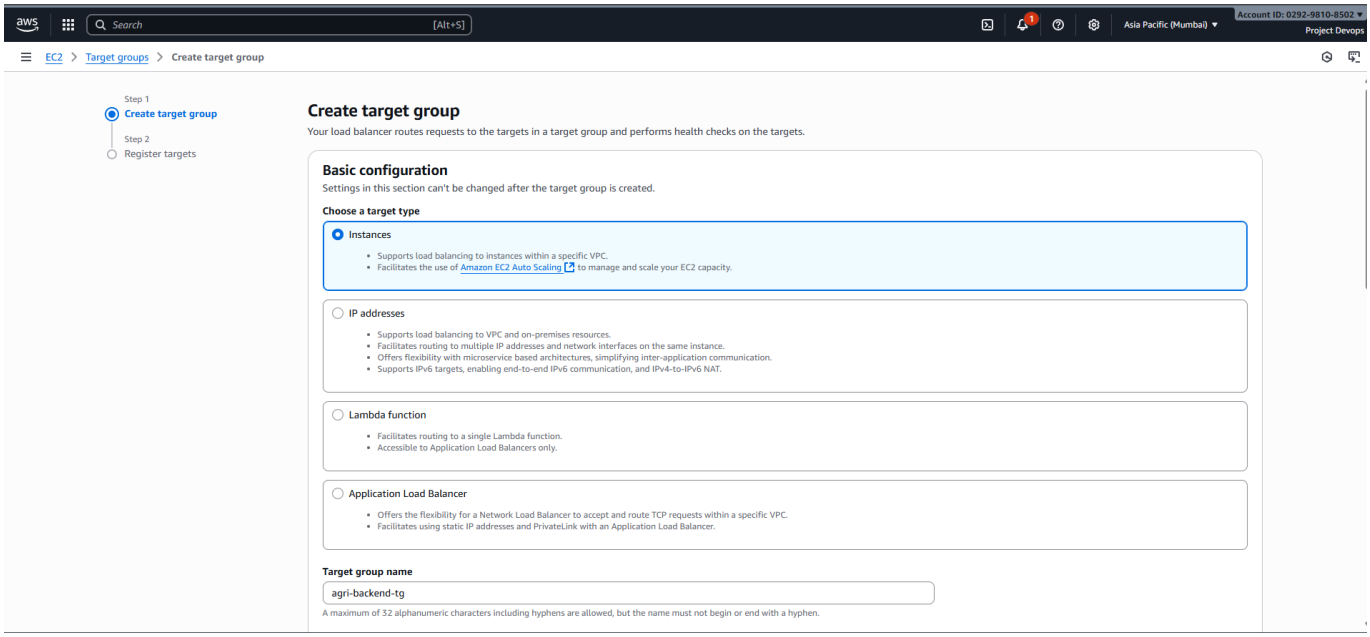
ubuntu@ip-10-0-131-108:/opt/agriculture-backend$ **nohup java -jar target/app.jar > app.log 2>&1 &**

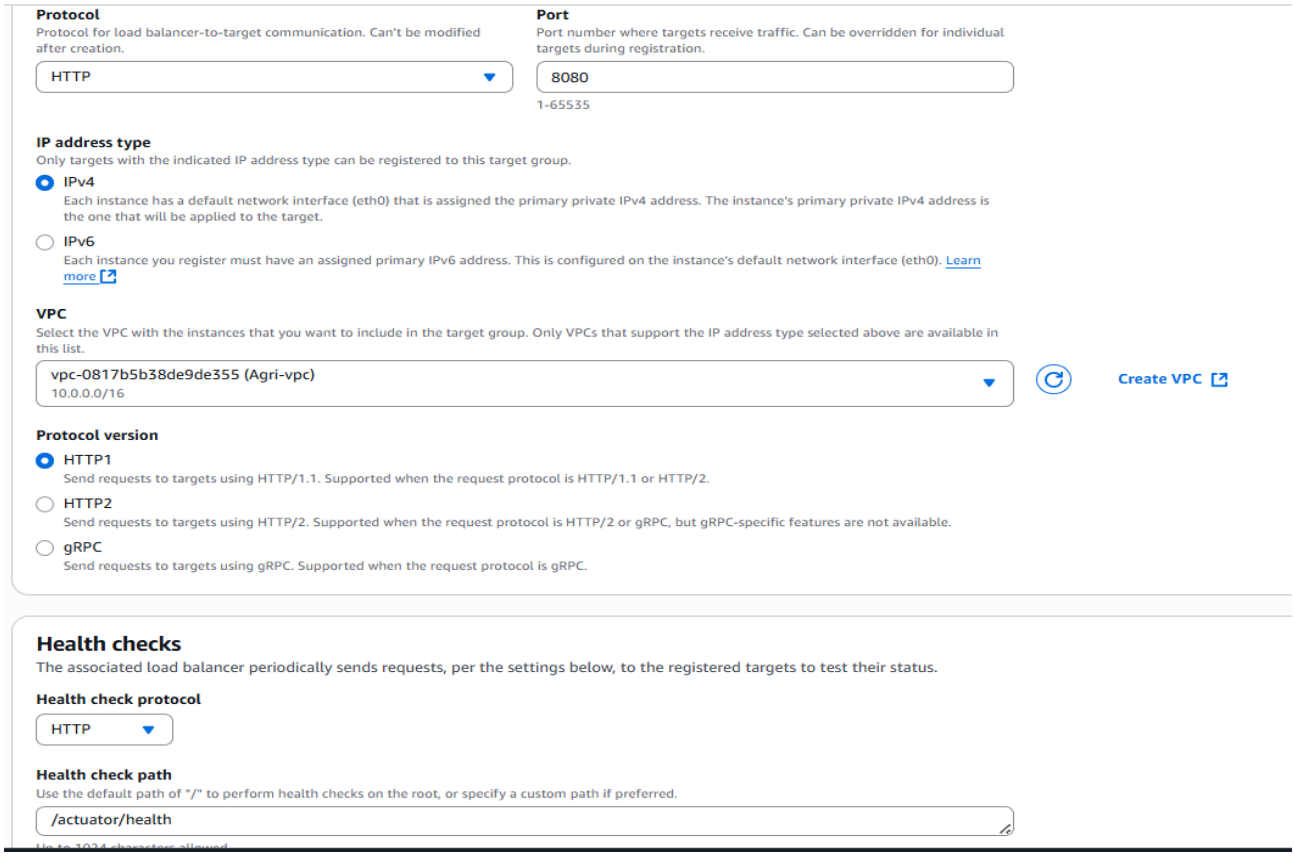
Above steps for communicate the frontend and backend without Load Balancer and ASG

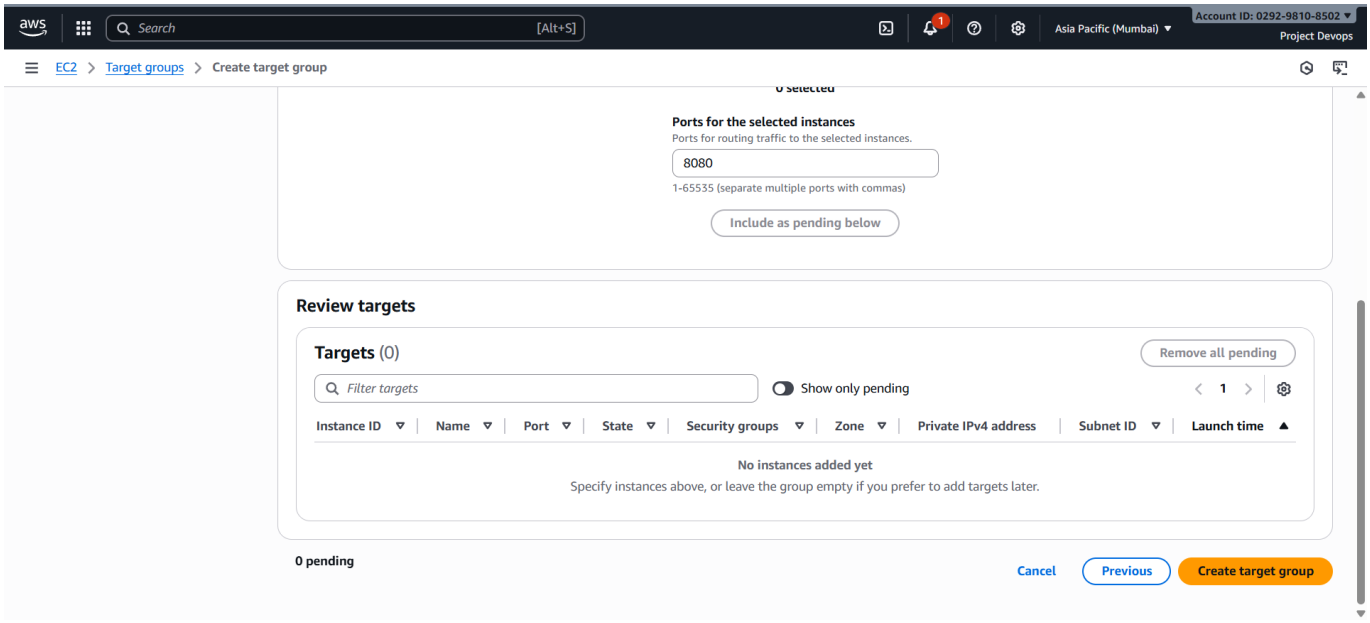
Create Target Groups

Backend Target Group

1. **Go to EC2 → Target Groups → Create target group**
2. **Target type → Instances**
3. **Protocol → HTTP, Port → 8080**
4. **Name → agri-backend-tg**
5. **VPC → same as your backend instance**
6. **Health check path → /actuator/health** # Default lightweight health endpoint in Spring Boot

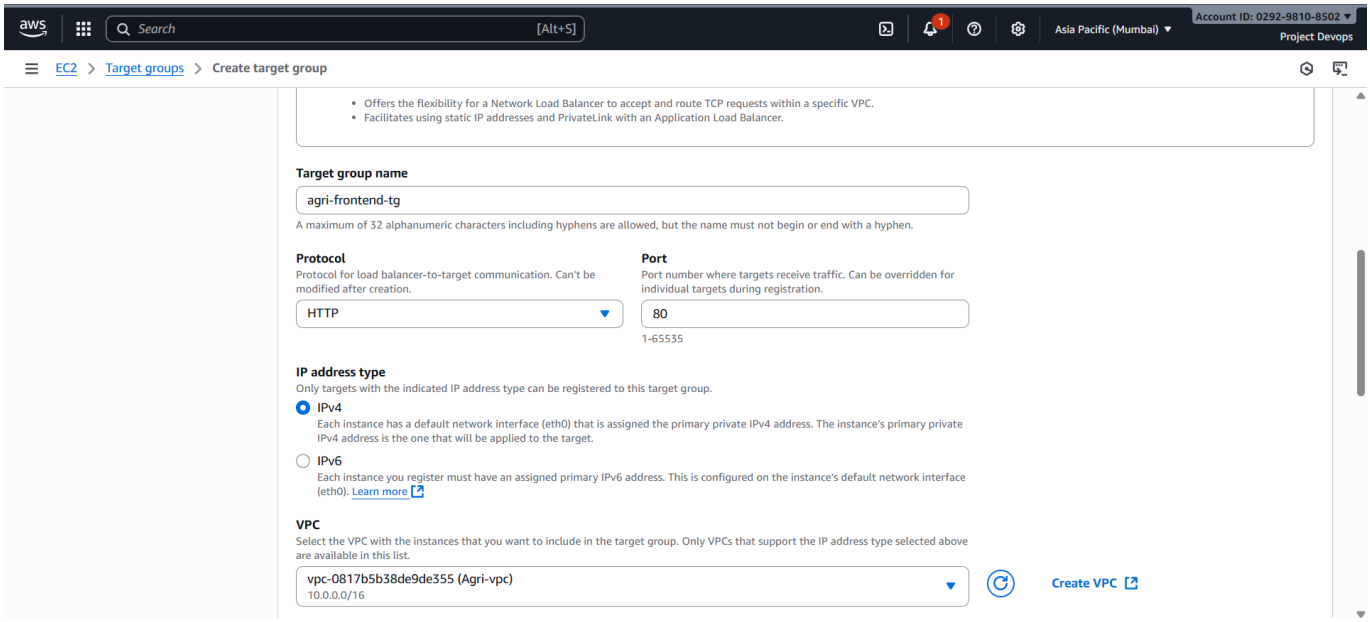






Frontend Target Group

1. Target type → Instances
2. Protocol → HTTP, Port → 80
3. Name → agri-frontend-tg
4. Health check path → /

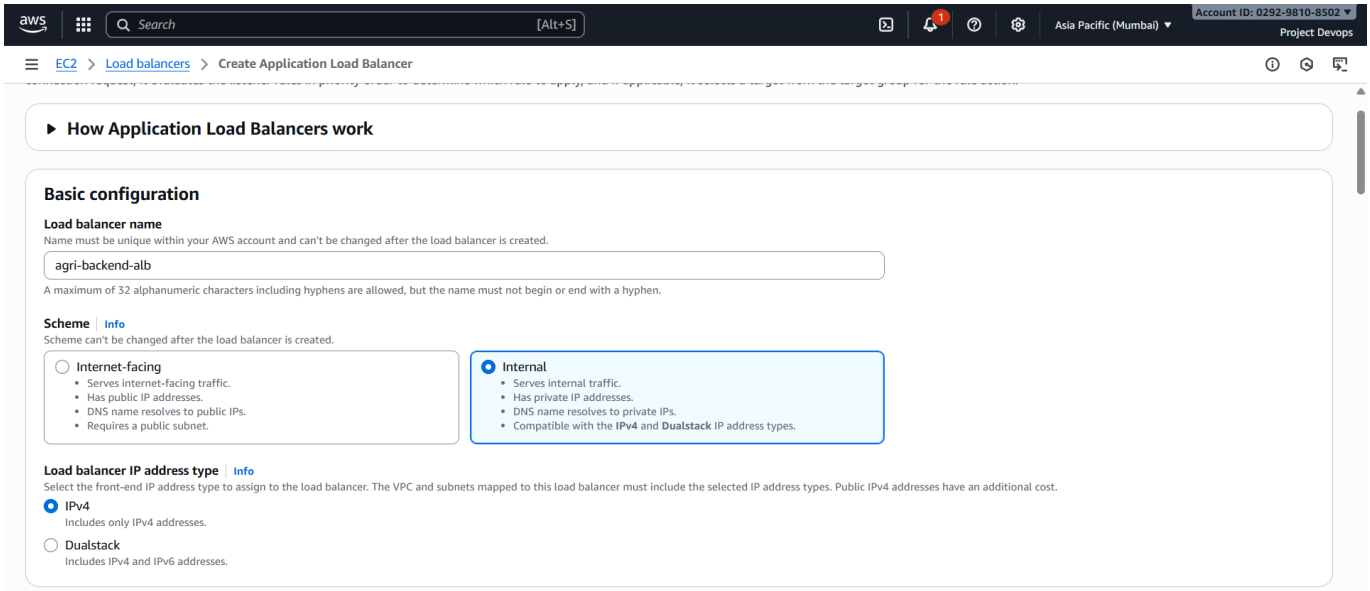


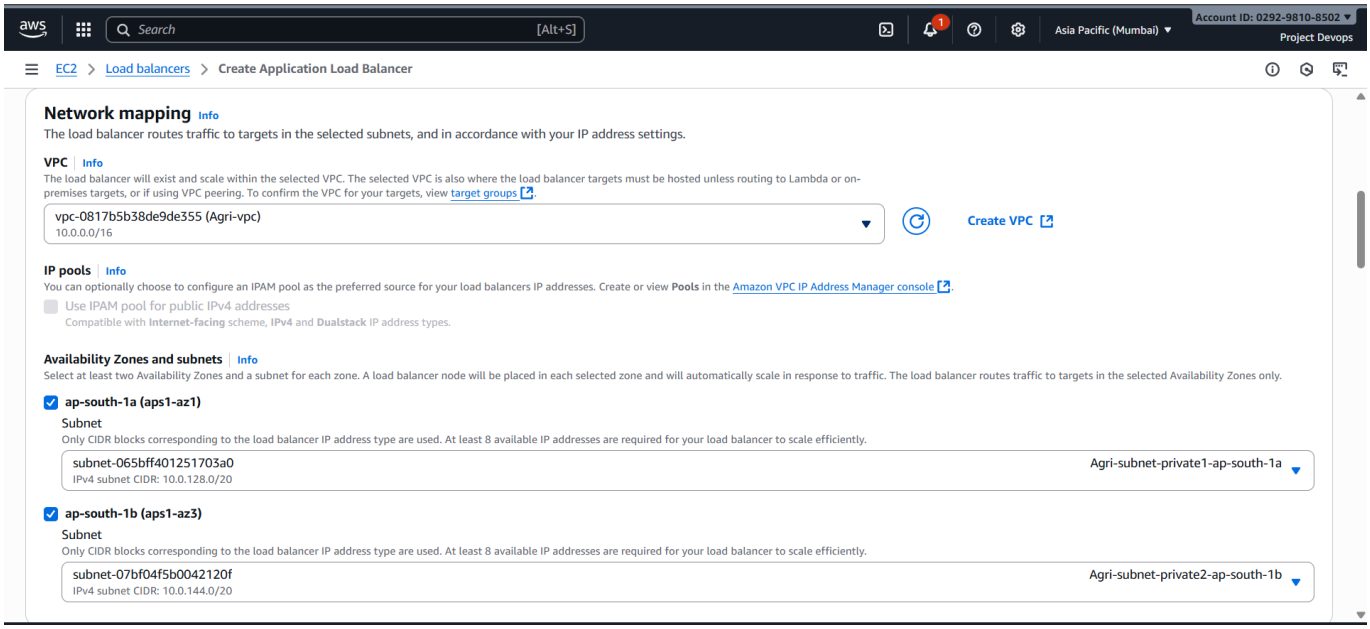
Create Application Load Balancers

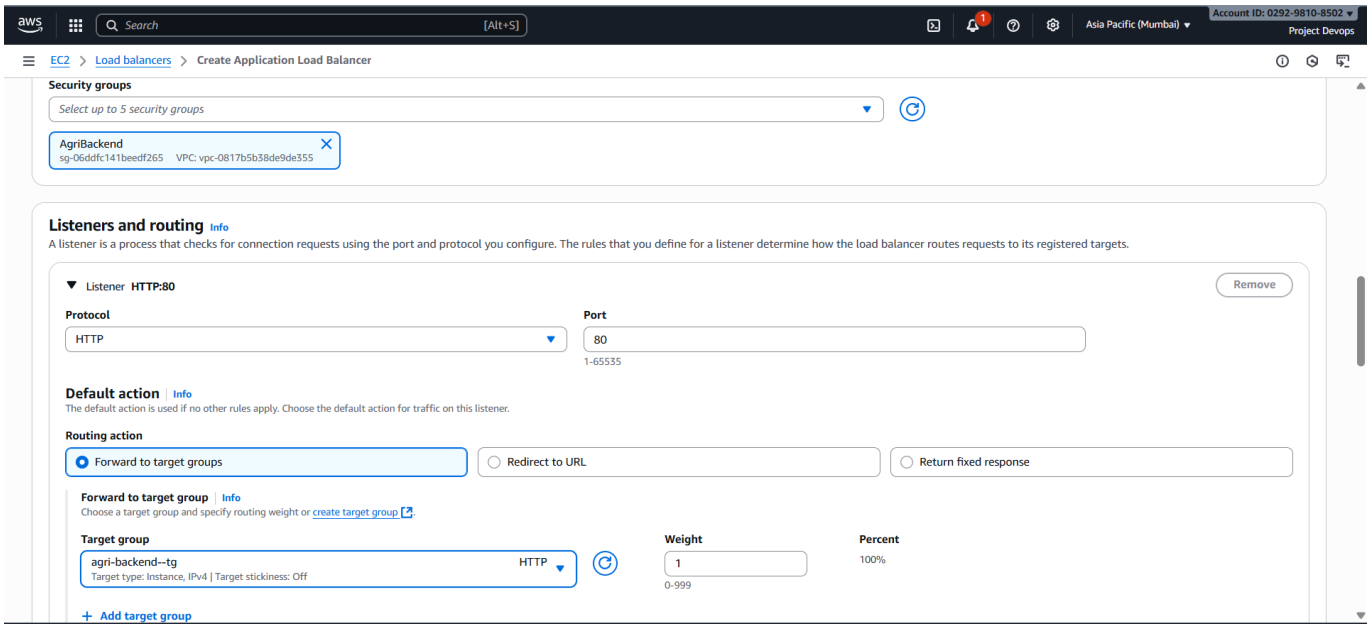
Backend ALB (Internal)

1. **EC2 → Load Balancers → Create Load Balancer**
2. Type → **Application Load Balancer**
3. Name → agri-backend-alb
4. Scheme → **Internal**
5. Network → same VPC

6. Select **two private subnets**
7. Security Group → allow **HTTP (8080)** only from frontend subnet or frontend ALB SG
8. Listener → HTTP:80 → forward to agri-backend-tg

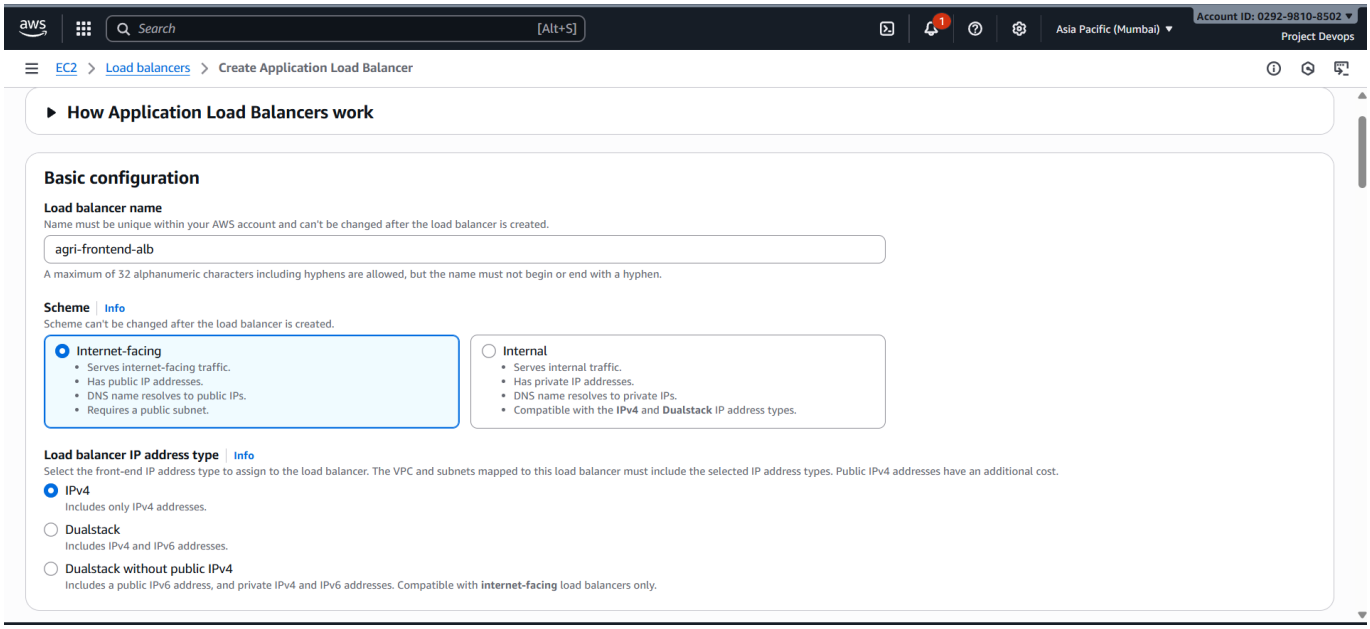


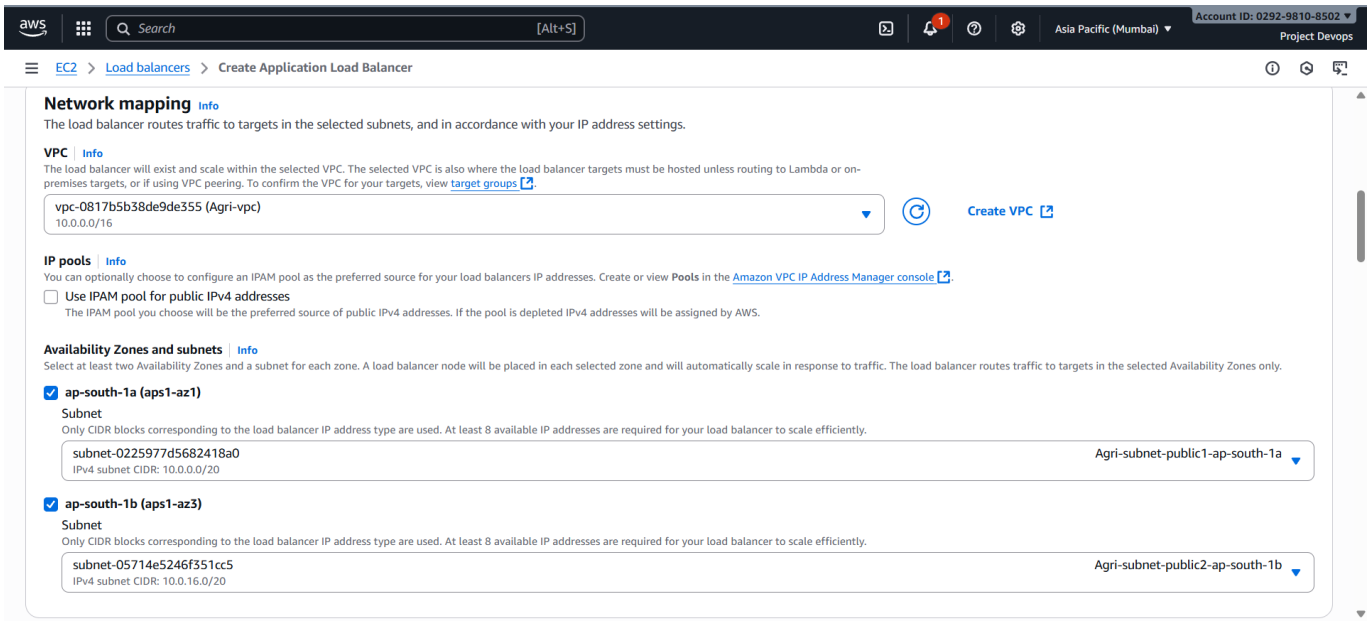


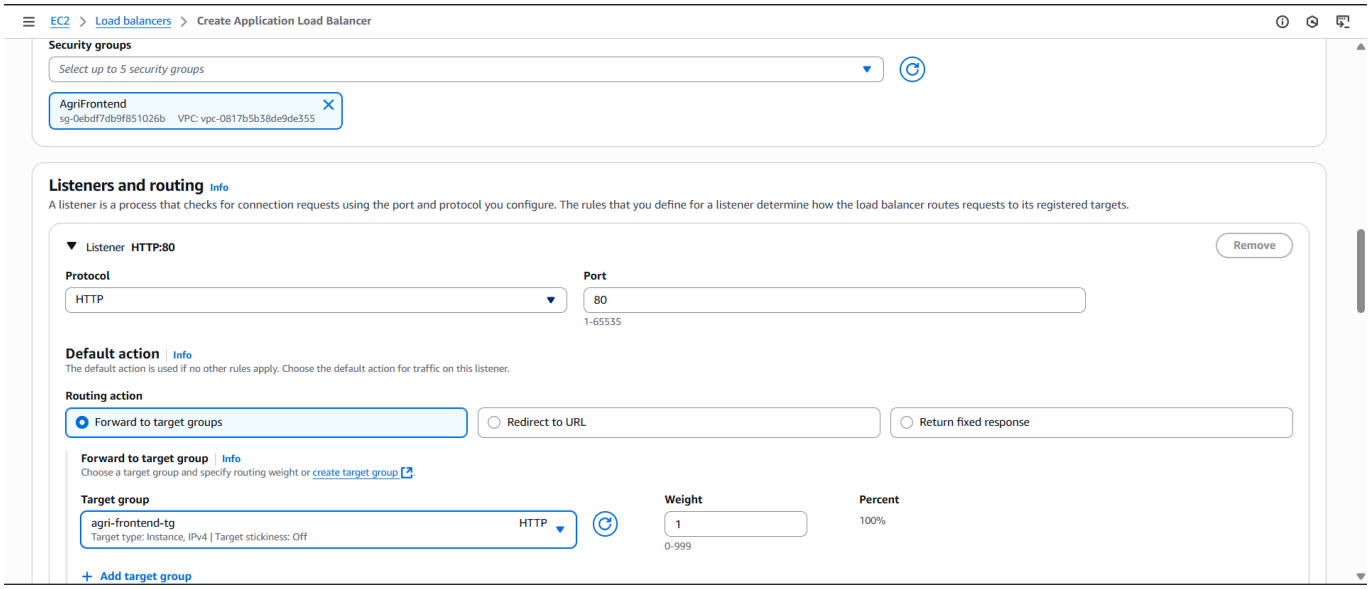


Frontend ALB (Public)

1. Name → agri-frontend-alb
2. Scheme → **Internet-facing**
3. Network → same VPC
4. Select **two public subnets**
5. Security Group → allow **HTTP (80)** from anywhere (0.0.0.0/0)
6. Listener → HTTP:80 → forward to agri-frontend-tg







Backend Setting!!!

Update the application.properties file

sudo nano /opt/agriculture-backend/src/main/resources/application.properties

Frontend Public IP, Backend Private IP

allowed.origins=http://agri-frontend-alb-1705890550.ap-south-1.elb.amazonaws.com

jwt.server.url=http://internal-agri-backend-alb-886281513.ap-south-1.elb.amazonaws.com

cd /opt/agriculture-backend

ubuntu@ip-10-0-139-42:/opt/agriculture-backend$ **mvn clean package -DskipTests**

ubuntu@ip-10-0-139-42:/opt/agriculture-backend$ **sudo chmod 777 run.sh**

ubuntu@ip-10-0-139-42:/opt/agriculture-backend$ **./run.sh**

🔧 Setting up agriculture-backend service...

Created symlink /etc/systemd/system/multi-user.target.wants/agriculture-backend.service → /etc/systemd/system/agriculture-backend.service.

✅ agriculture-backend service has been started successfully!

 Status command: sudo systemctl status agriculture-backend

📜 Logs command: sudo journalctl -u agriculture-backend -f

Frontend Setting!!!

Update Backend IP to Backend Endpoint /etc/nginx/sites-available/default

sudo nano /etc/nginx/sites-available/default

```
server {
    listen 80;
    server_name _;

    root /var/www/html;
    index index.html;

    # Serve React app
    location / {
        try_files $uri /index.html;
    }

    # Proxy API to Backend ALB
    location /api/ {
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $http_x_forwarded_proto;

        # PASTE YOUR BACKEND ALB DNS HERE
        proxy_pass http://internal-agri-backend-alb-886281513.ap-south-1.elb.amazonaws.com/;
        proxy_read_timeout 90;
    }

    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-Content-Type-Options "nosniff";
}
```

ubuntu@ip-10-0-12-31:~$ **cd /var/www/frontend/src/**

ubuntu@ip-10-0-12-31:/var/www/frontend/src$ **sudo nano Config.jsx**

// config.jsx

```
export const BASE_URL = 'http://agri-frontend-alb-1705890550.ap-south-1.elb.amazonaws.com/api';
```

ubuntu@ip-10-0-12-31:/var/www/frontend/src$ **cd ..**

ubuntu@ip-10-0-12-31:/var/www/frontend$ **npm run build**

ubuntu@ip-10-0-12-31:/var/www/frontend$ **sudo rm -rf /var/www/html/***

ubuntu@ip-10-0-12-31:/var/www/frontend$ **sudo cp -r dist/* /var/www/html/**

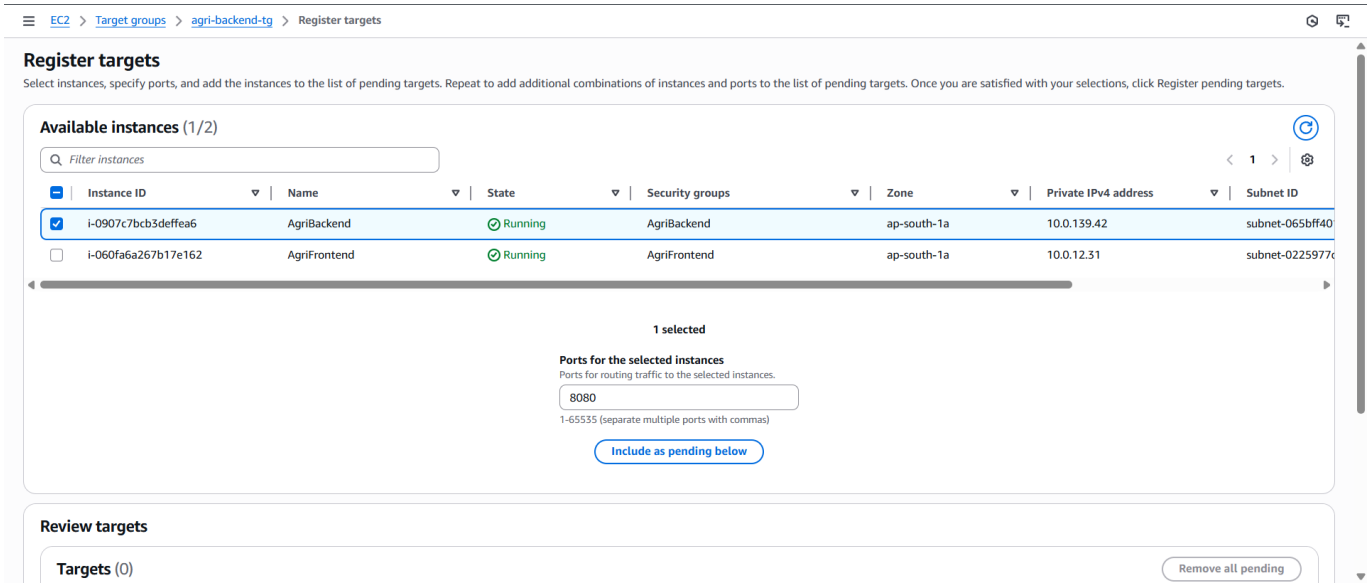
ubuntu@ip-10-0-12-31:/var/www/frontend$ **sudo systemctl reload nginx**

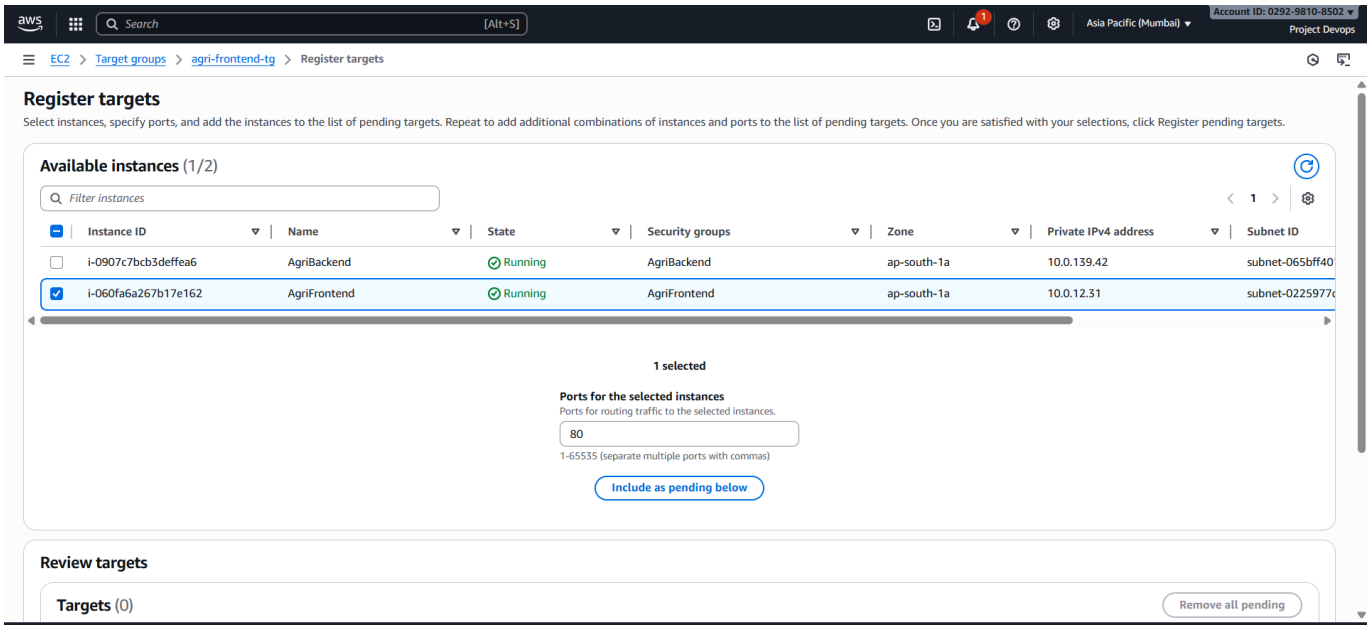
ubuntu@ip-10-0-12-31:/var/www/frontend$ **sudo nginx -t**

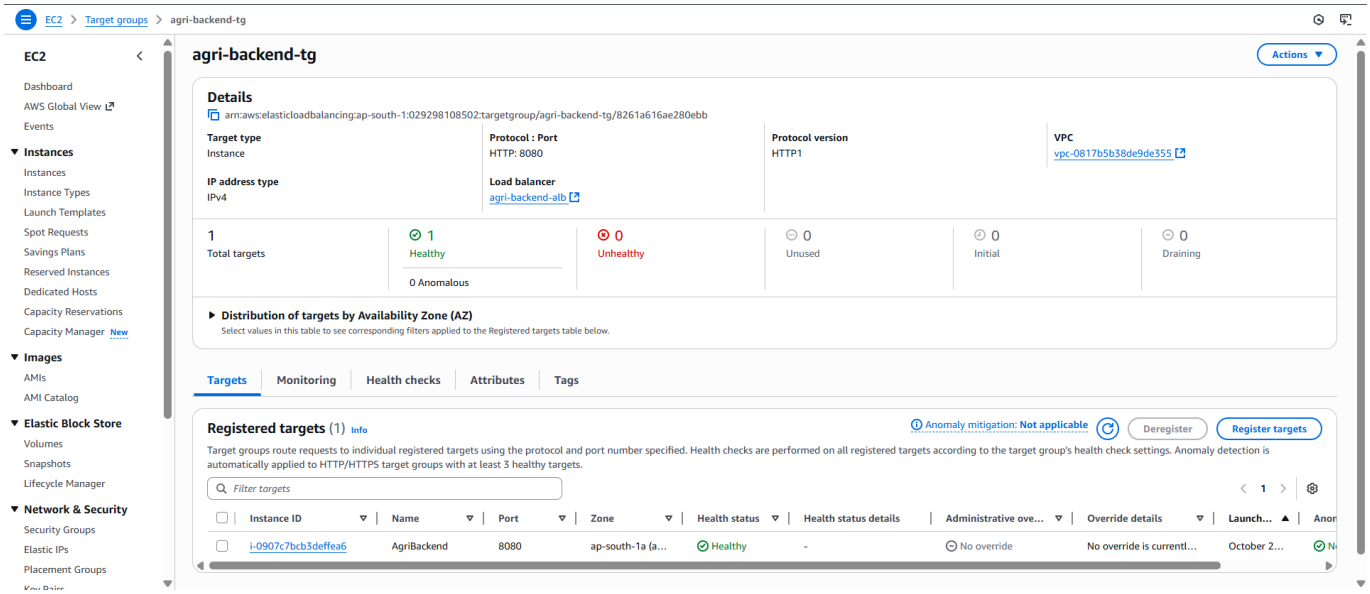
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok

nginx: configuration file /etc/nginx/nginx.conf test is successful

ubuntu@ip-10-0-12-31:/var/www/frontend$ **sudo systemctl restart nginx**

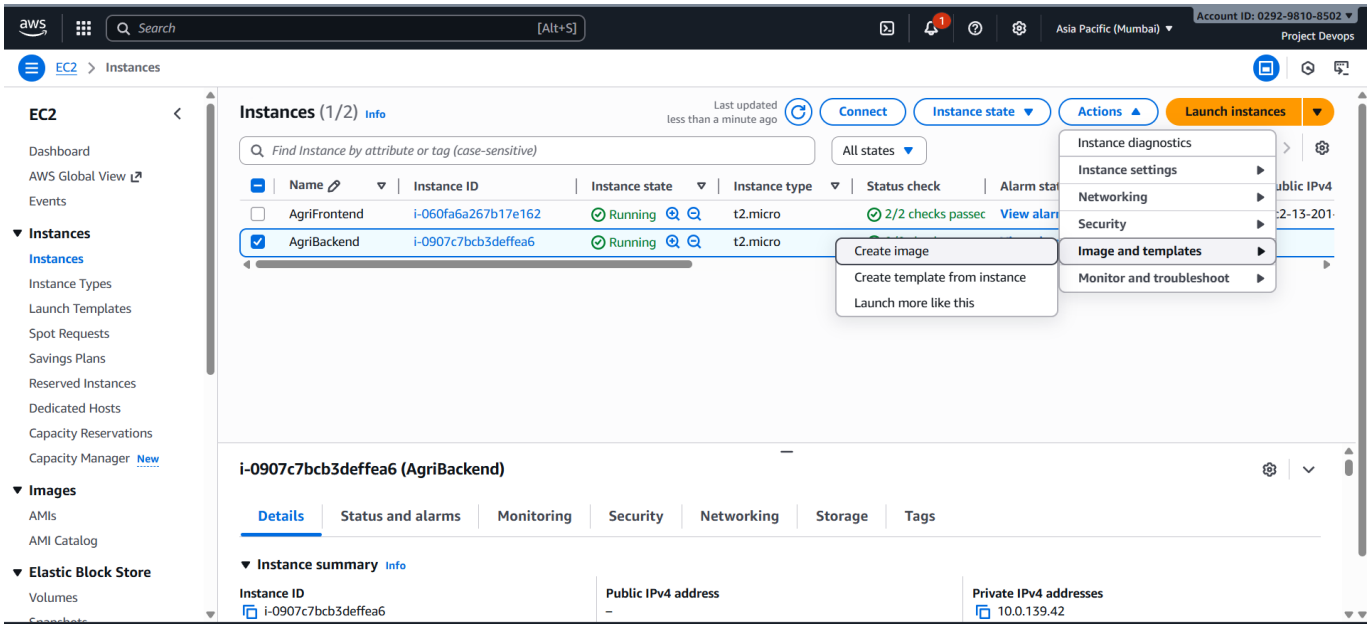


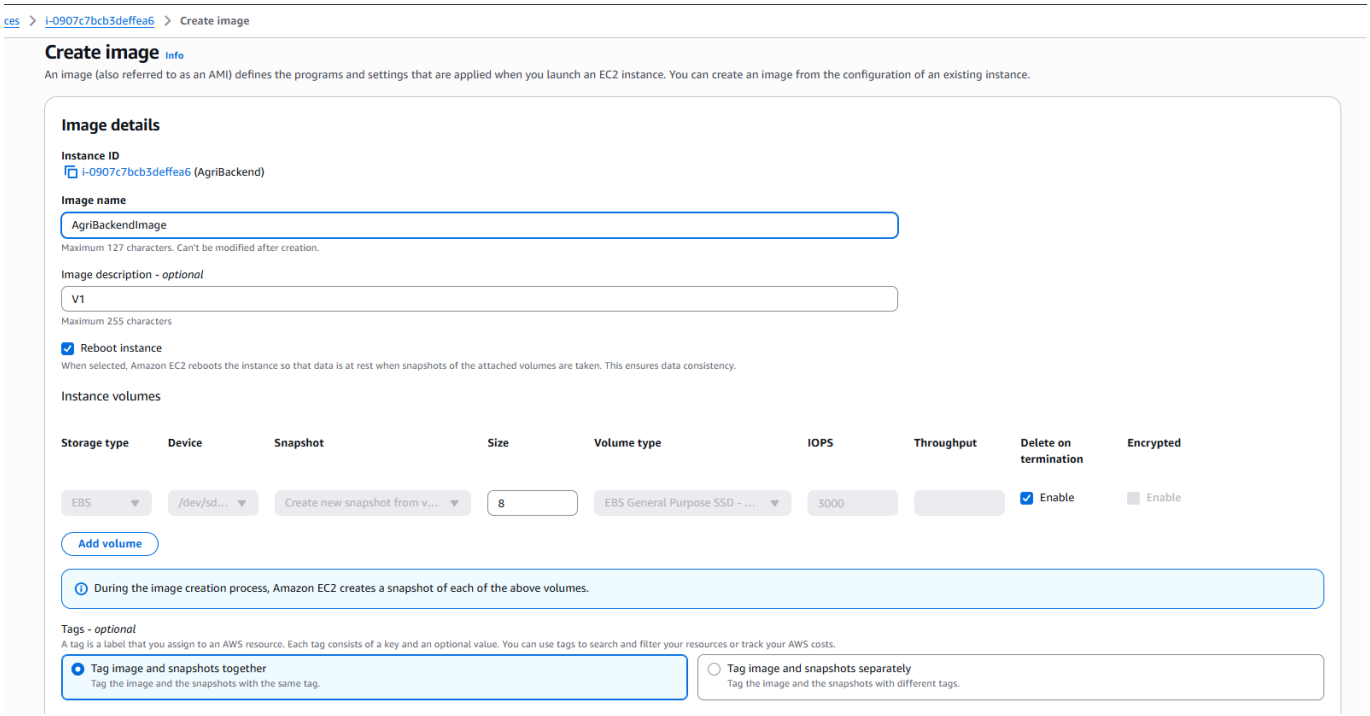




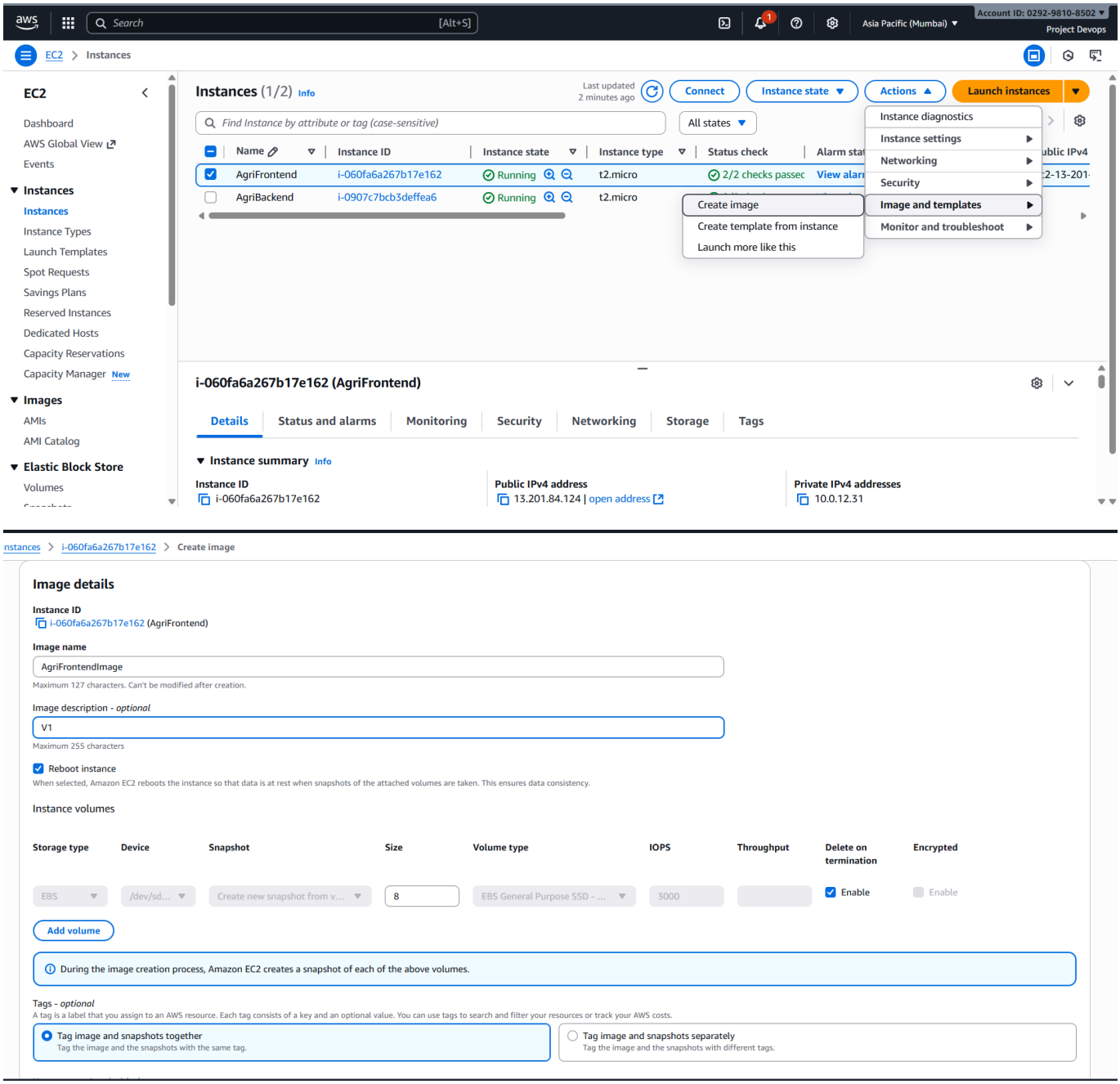
Auto Scaling Group

AMI for Frontend

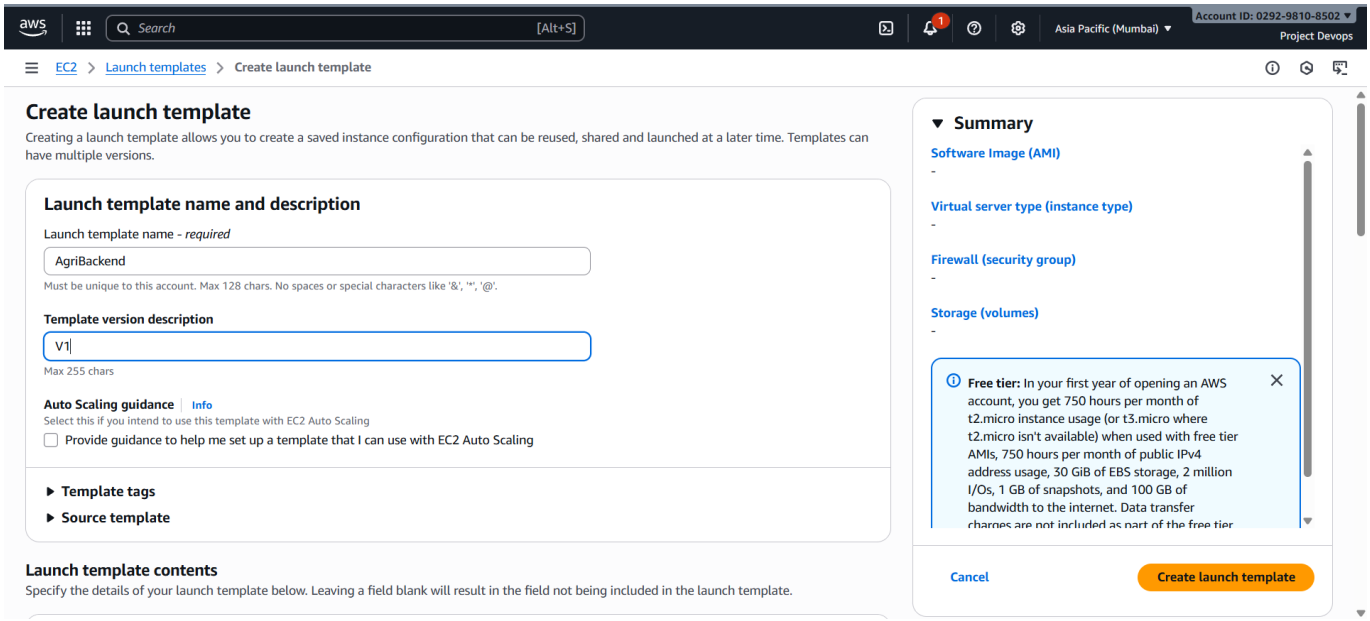


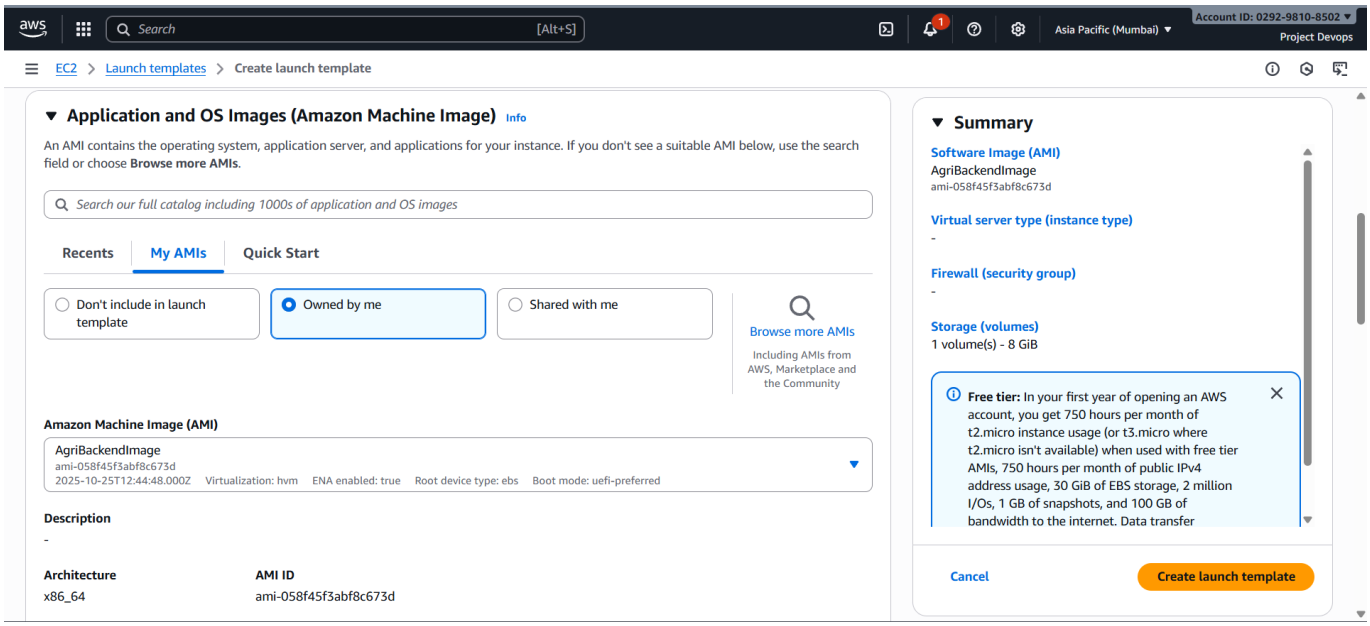


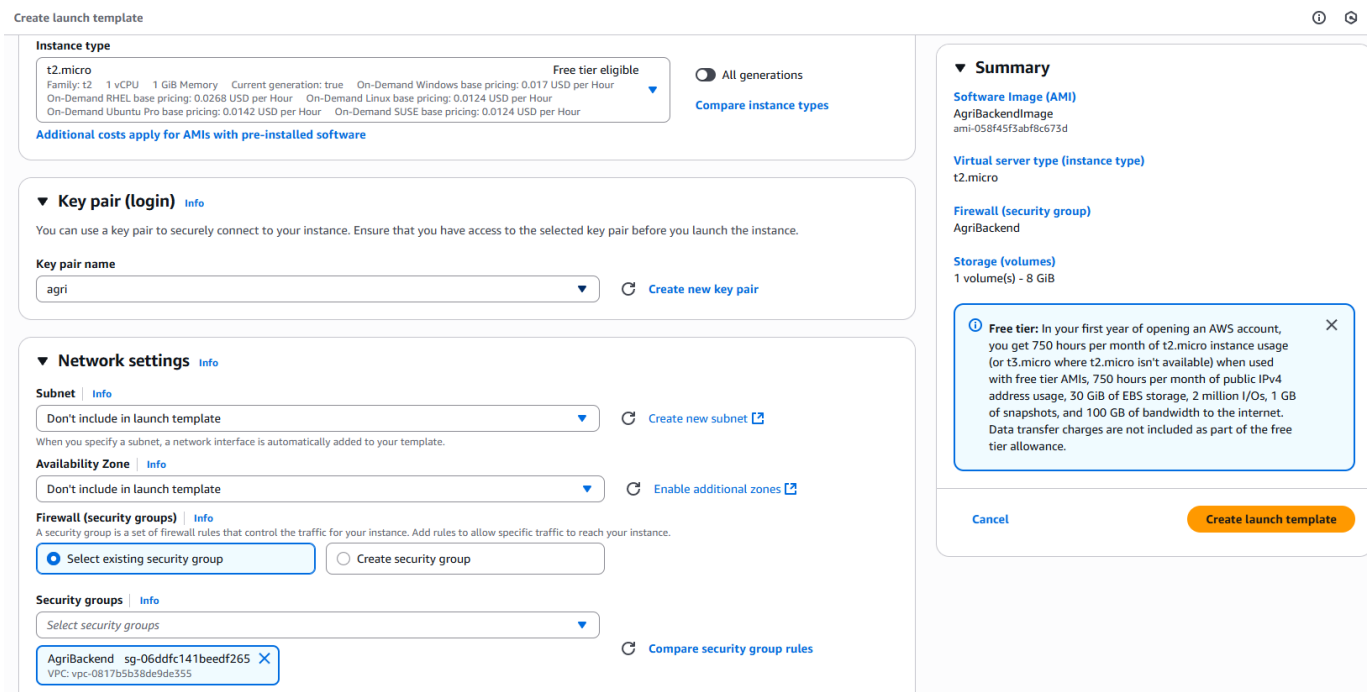
AMI for Backend



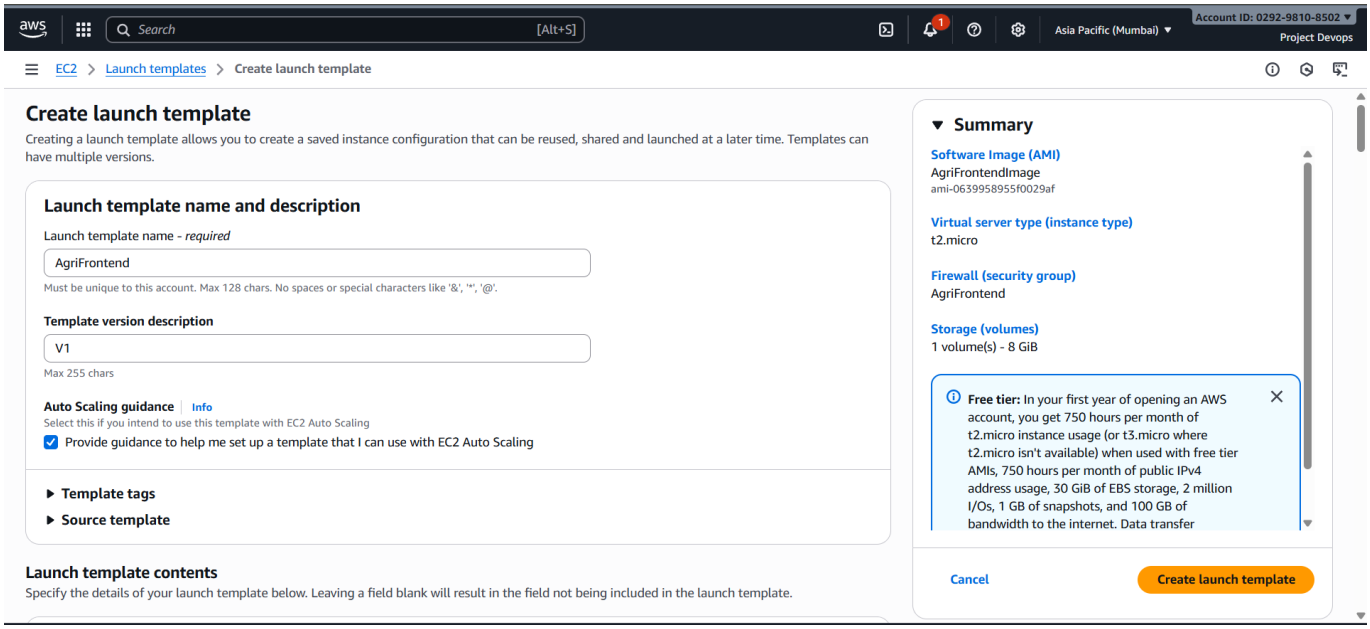
Backend Launch Template

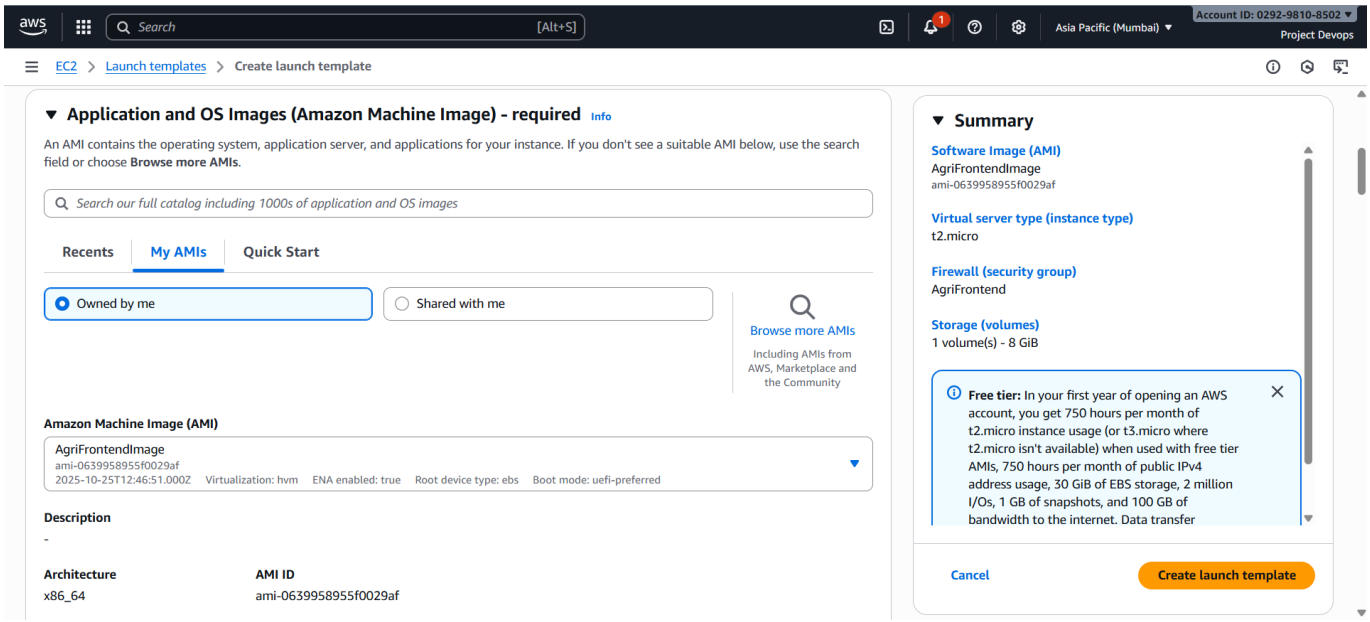


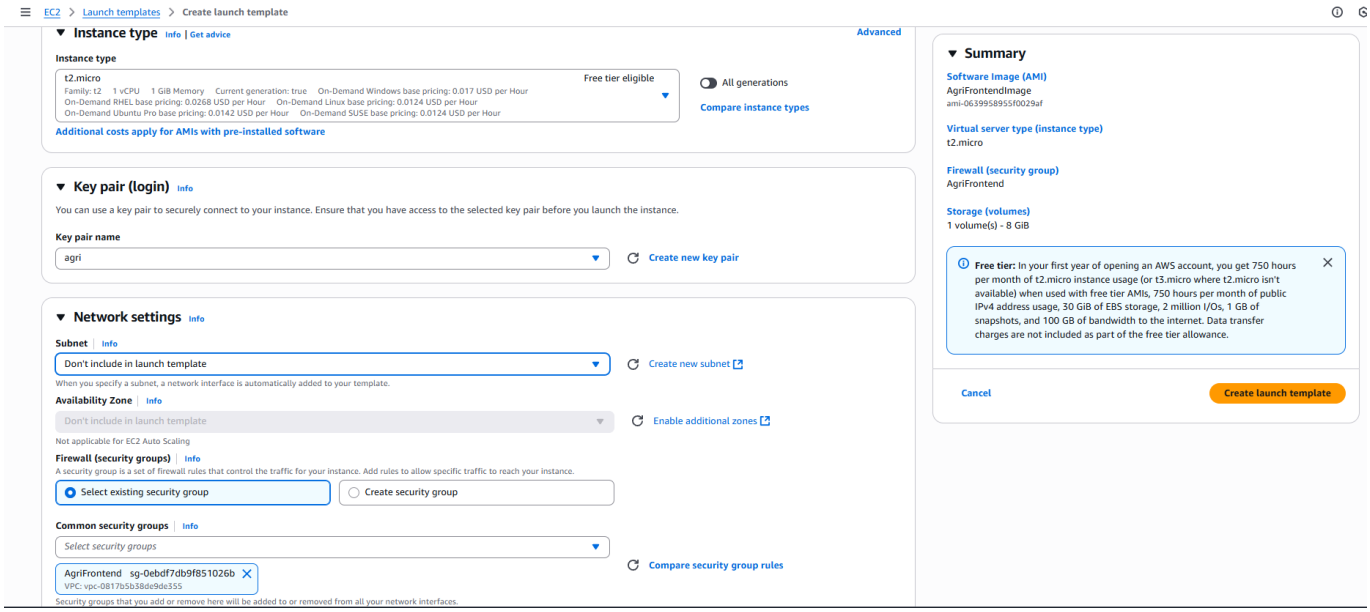




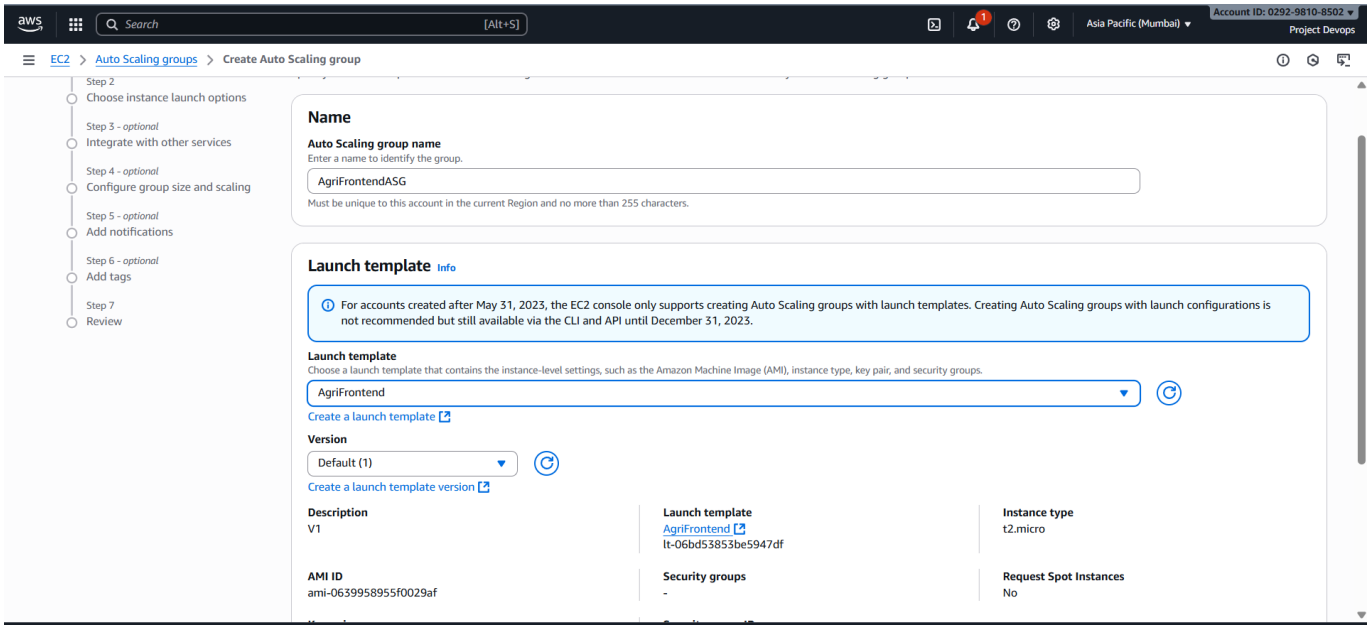
Frontend Launch Template

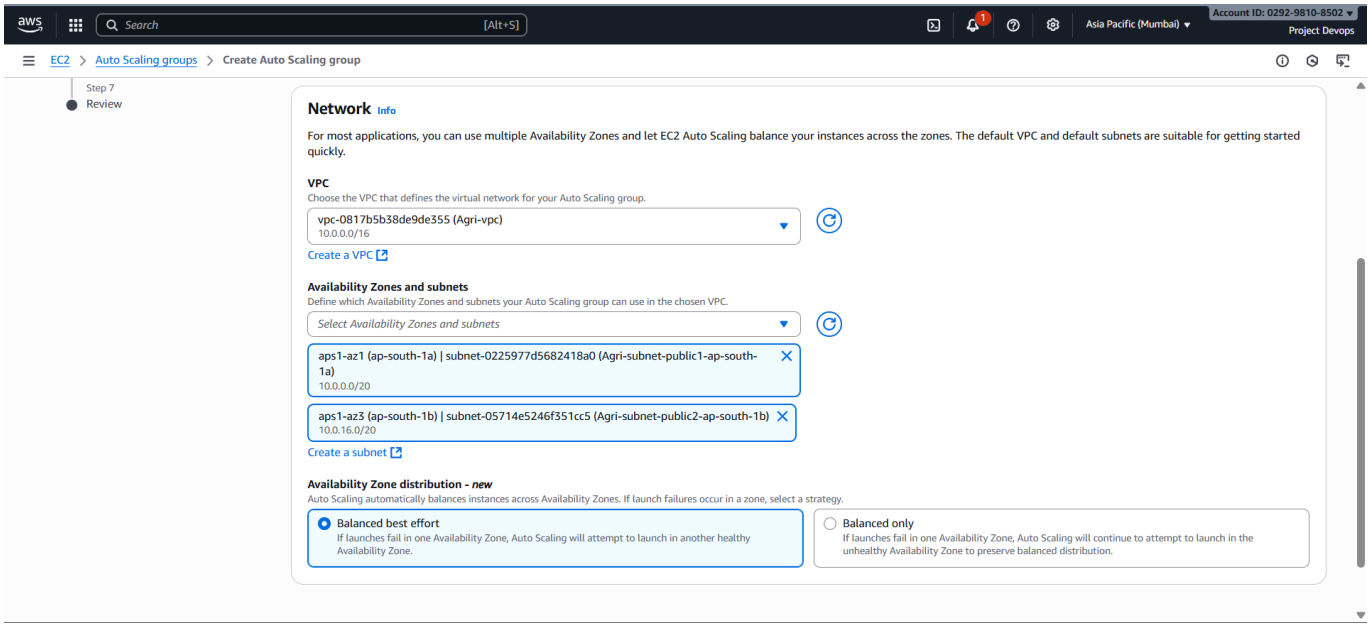


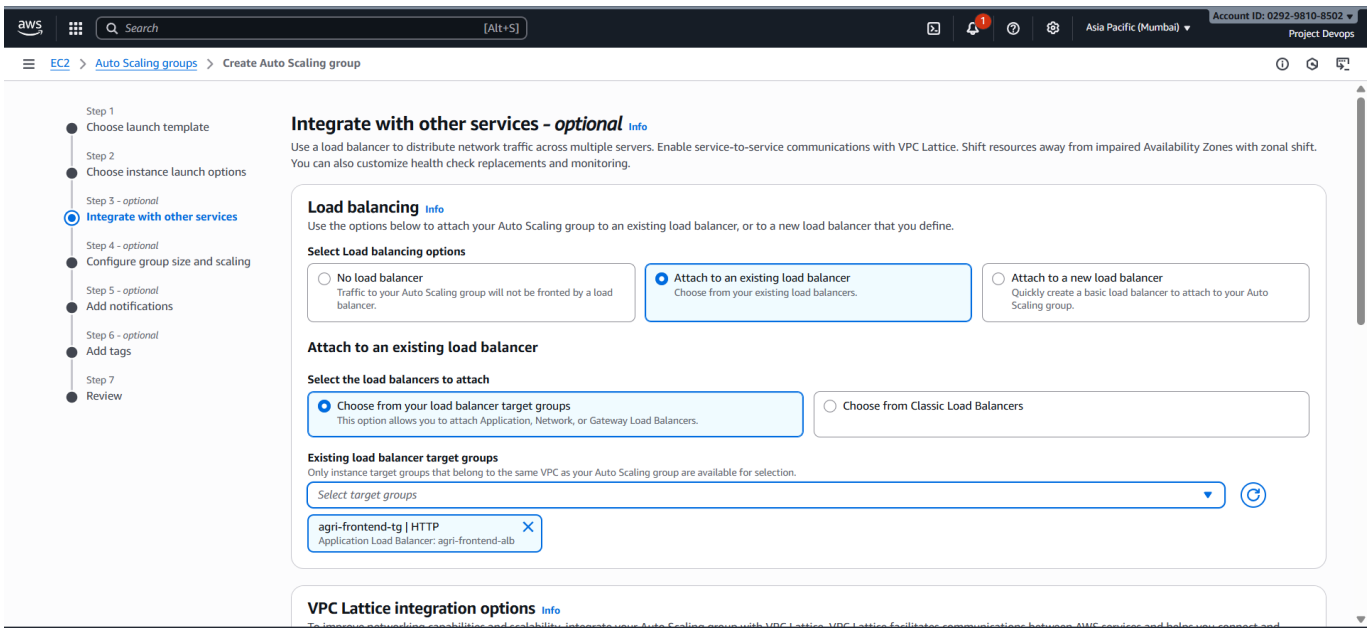


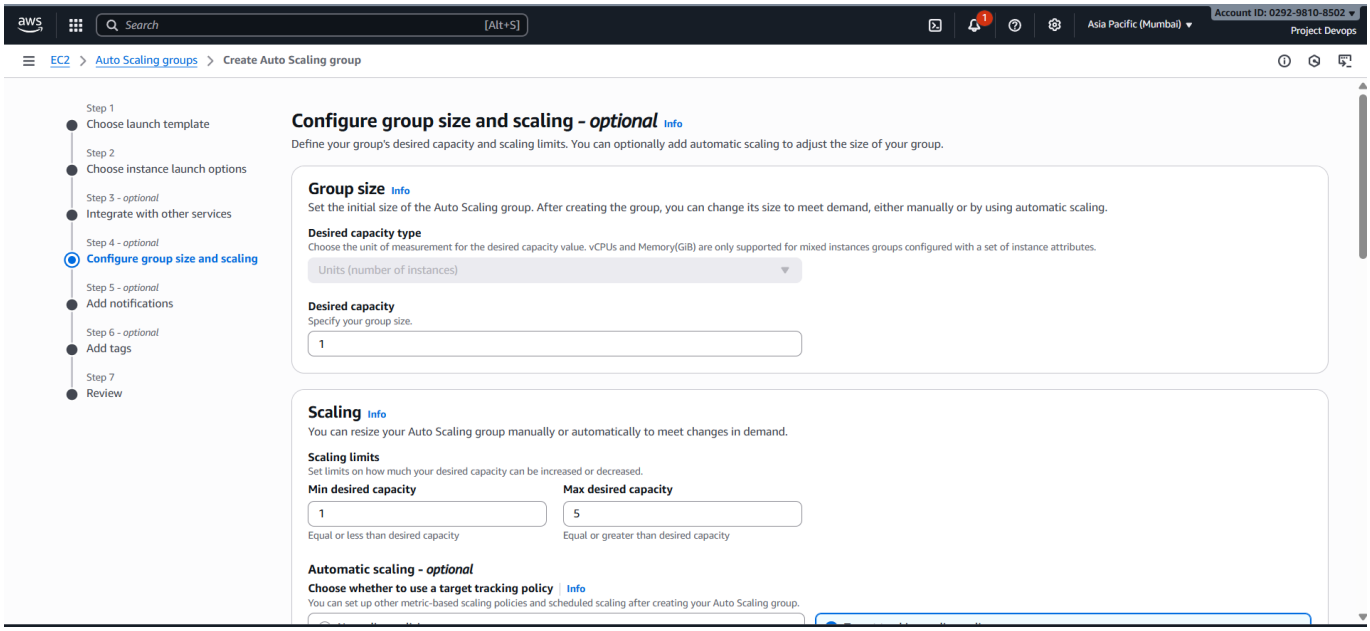


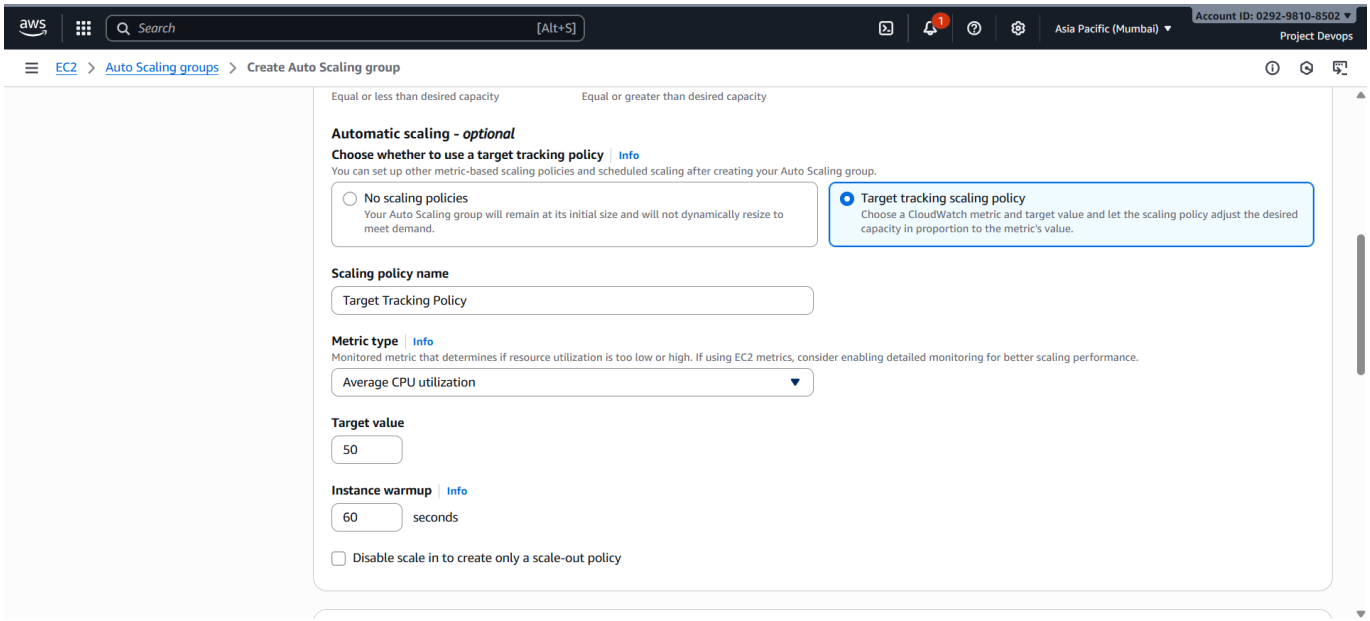
Frontend ASG

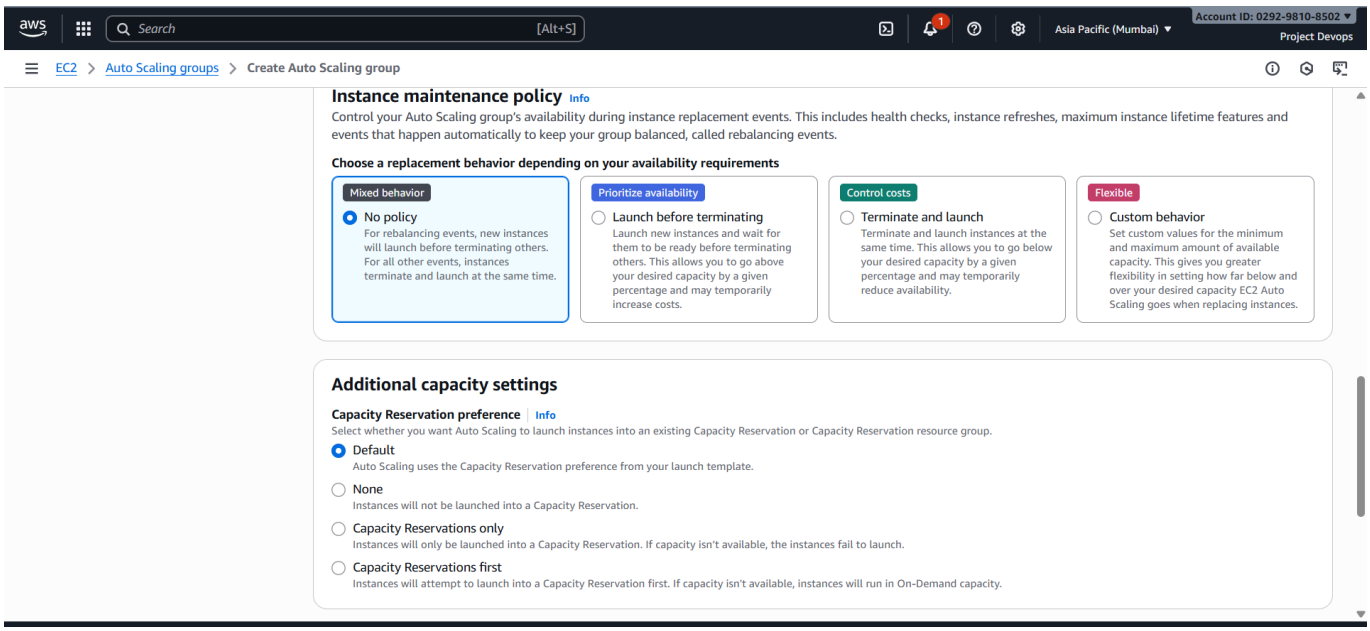




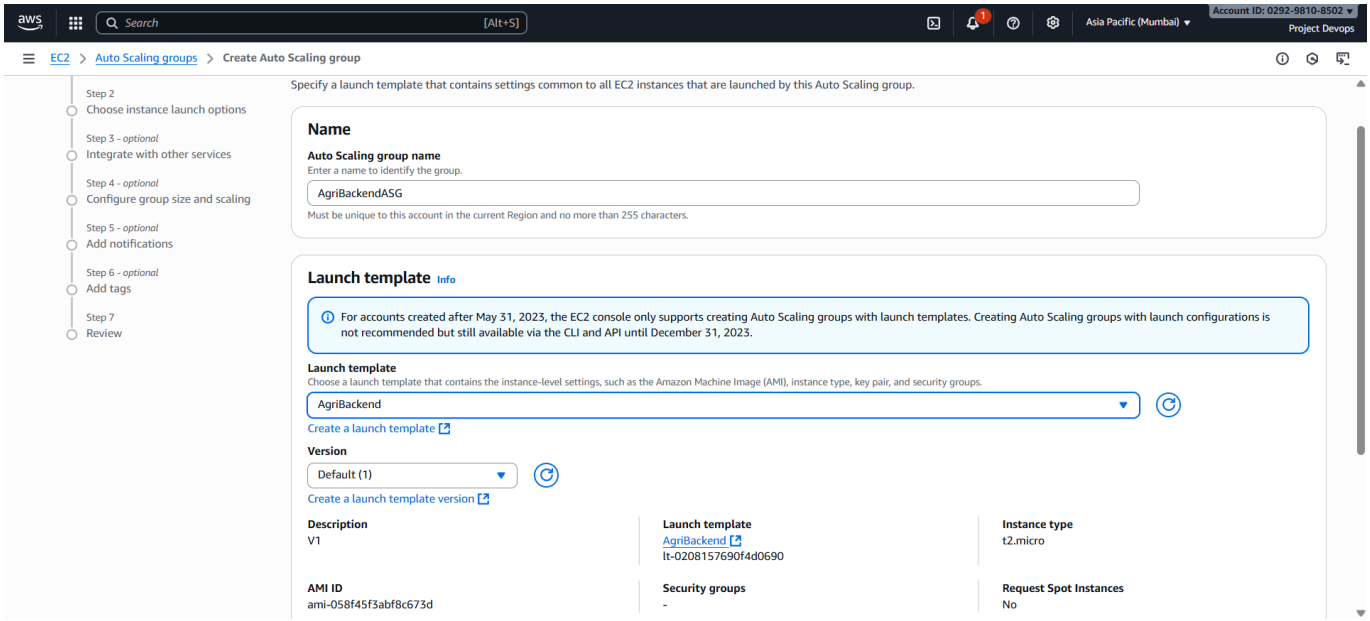


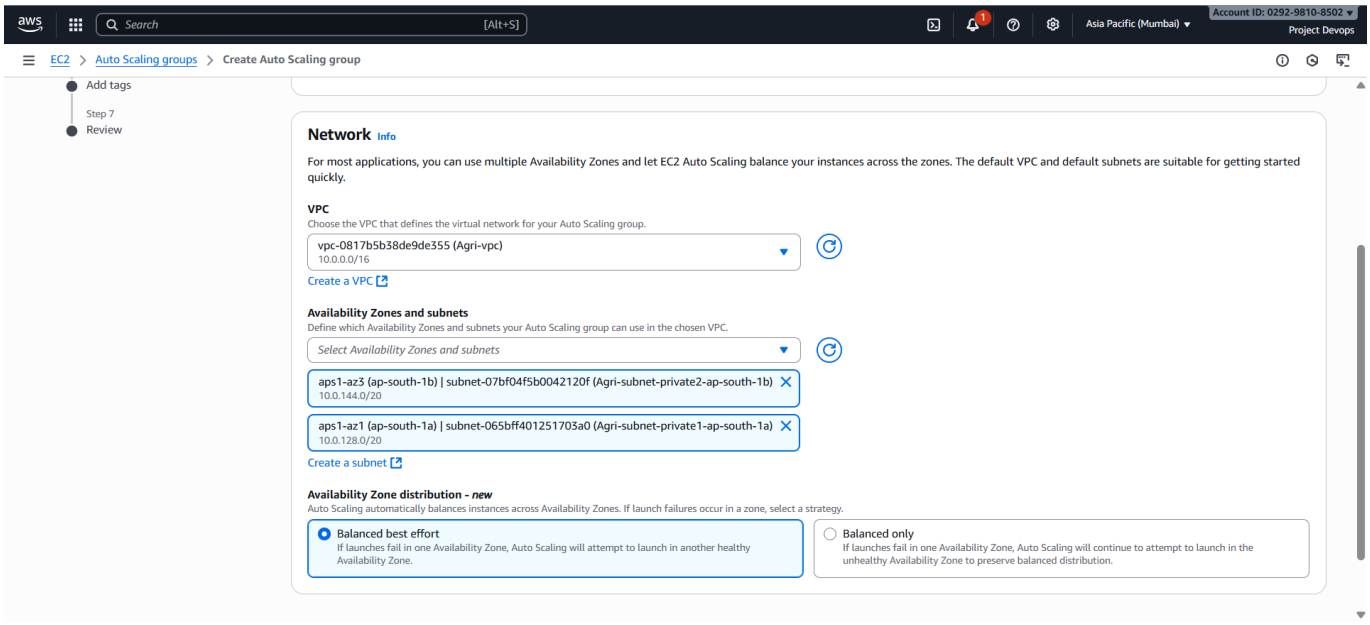

Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - optional
Integrate with other services
Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Review
Configure group size and scaling - optional Info
Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.
Group size Info
Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.
Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.
Units (number of instances)
Desired capacity
Specify your group size.
1
Scaling Info
You can resize your Auto Scaling group manually or automatically to meet changes in demand.
Scaling limits
Set limits on how much your desired capacity can be increased or decreased.
Min desired capacity
1
Equal or less than desired capacity
Max desired capacity
5
Equal or greater than desired capacity
Automatic scaling - optional
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

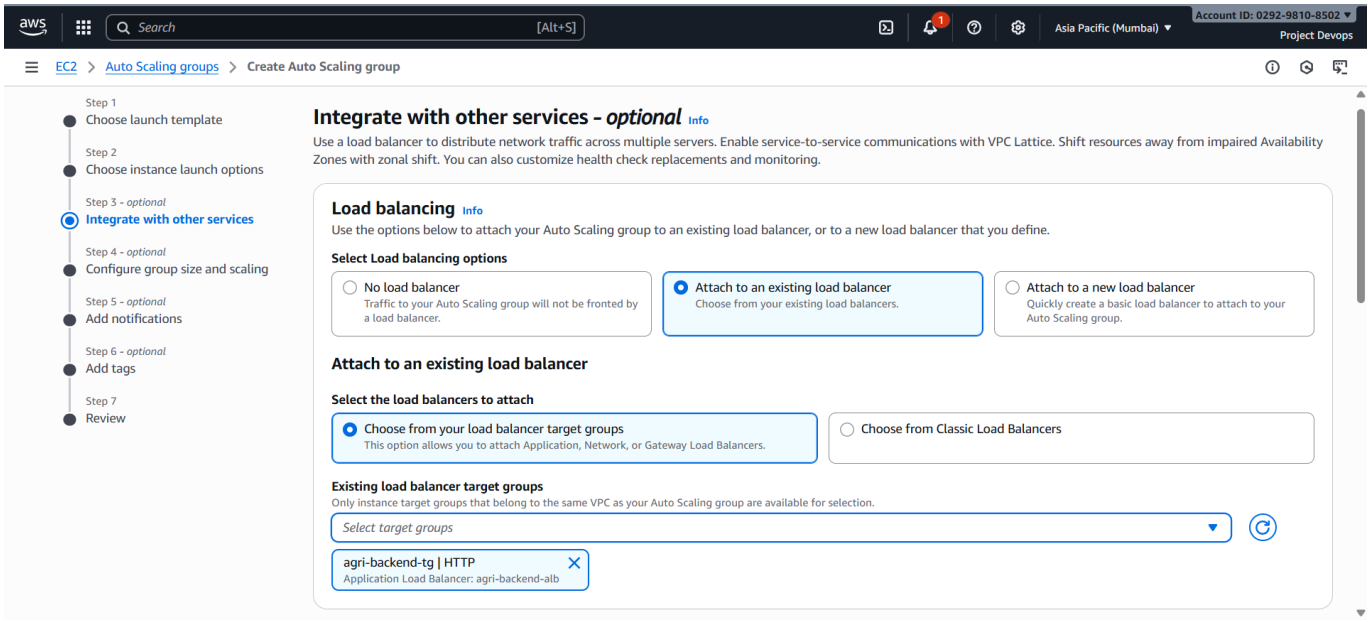

Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
EC2 > Auto Scaling groups > Create Auto Scaling group
Equal or less than desired capacity
Equal or greater than desired capacity
Automatic scaling - optional
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.
No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.
Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.
Scaling policy name
Target Tracking Policy
Metric type Info
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.
Average CPU utilization
Target value
50
Instance warmup Info
60
seconds
Disable scale in to create only a scale-out policy

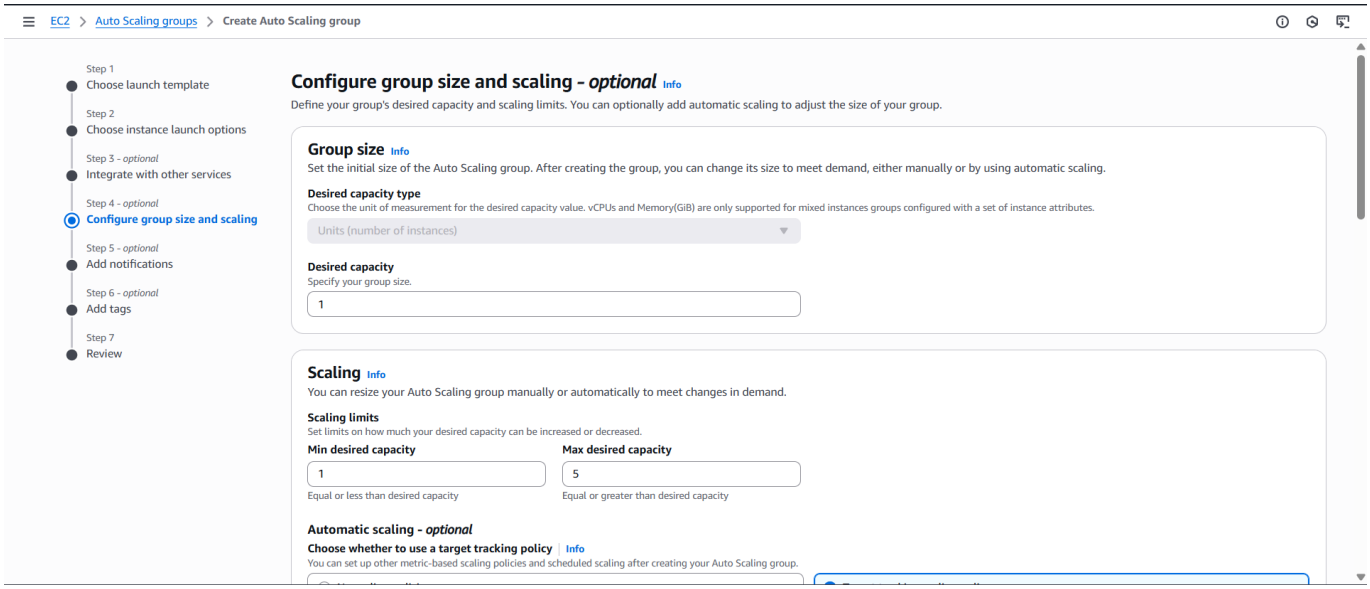

Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
EC2 > Auto Scaling groups > Create Auto Scaling group
Instance maintenance policy Info
Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.
Choose a replacement behavior depending on your availability requirements
Mixed behavior
No policy
For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.
Prioritize availability
Launch before terminating
Launch new instances and wait for them to be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.
Control costs
Terminate and launch
Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.
Flexible
Custom behavior
Set custom values for the minimum and maximum amount of available capacity. This gives you greater flexibility in setting how far below and over your desired capacity EC2 Auto Scaling goes when replacing instances.
Additional capacity settings
Capacity Reservation preference Info
Select whether you want Auto Scaling to launch instances into an existing Capacity Reservation or Capacity Reservation resource group.
Default
Auto Scaling uses the Capacity Reservation preference from your launch template.
None
Instances will not be launched into a Capacity Reservation.
Capacity Reservations only
Instances will only be launched into a Capacity Reservation. If capacity isn't available, the instances fail to launch.
Capacity Reservations first
Instances will attempt to launch into a Capacity Reservation first. If capacity isn't available, instances will run in On-Demand capacity.

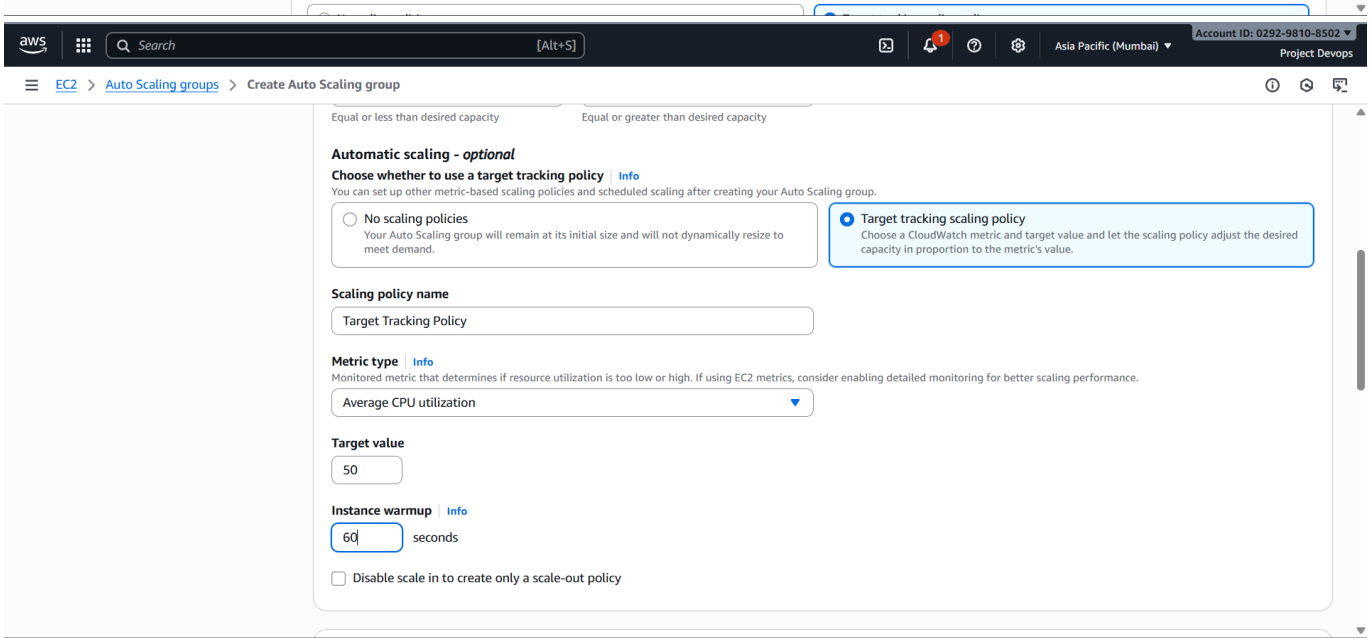
Backend ASG

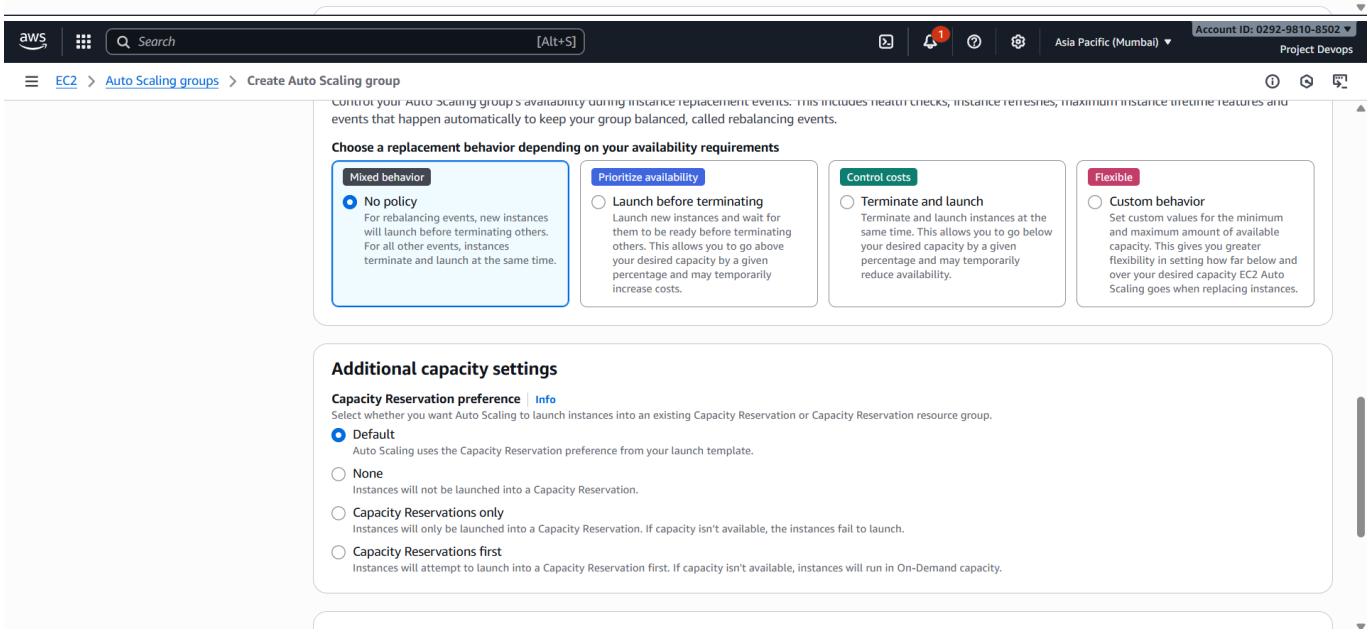

Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 2 Choose instance launch options
Step 3 - optional Integrate with other services
Step 4 - optional Configure group size and scaling
Step 5 - optional Add notifications
Step 6 - optional Add tags
Step 7 Review
Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.
Name
Auto Scaling group name
Enter a name to identify the group.
AgriBackendASG
Must be unique to this account in the current Region and no more than 255 characters.
Launch template Info
For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.
Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
AgriBackend
Create a launch template
Version
Default (1)
Create a launch template version
Description
V1
Launch template
AgriBackend
lt-0208157690f4d0690
Instance type
t2.micro
AMI ID
ami-058f45f3abf8c673d
Security groups
-
Request Spot Instances
No

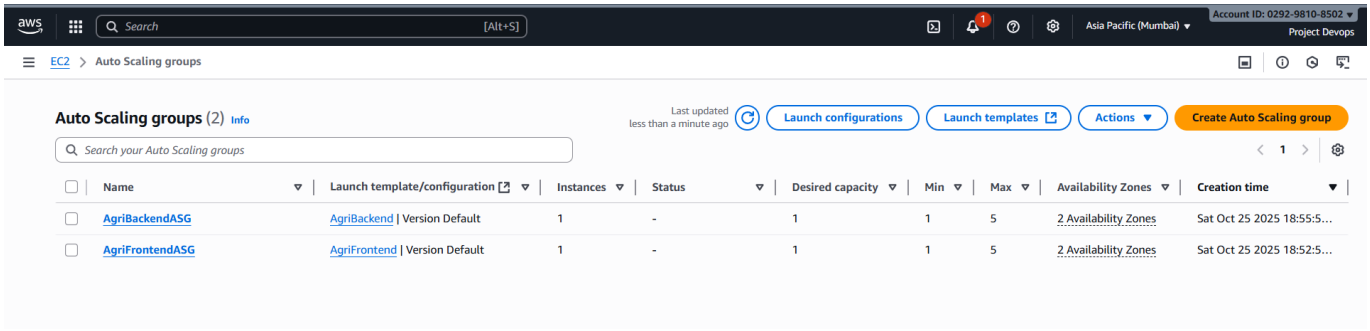

Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
EC2 > Auto Scaling groups > Create Auto Scaling group
Add tags
Step 7 Review
Network Info
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.
VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-0817b5b38de9de355 (Agri-vpc)
10.0.0.0/16
Create a VPC
Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
Select Availability Zones and subnets
aps1-az3 (ap-south-1b) | subnet-07bf04f5b0042120f (Agri-subnet-private2-ap-south-1b)
10.0.144.0/20
aps1-az1 (ap-south-1a) | subnet-065bff401251703a0 (Agri-subnet-private1-ap-south-1a)
10.0.128.0/20
Create a subnet
Availability Zone distribution - new
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.
Balanced best effort
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.
Balanced only
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

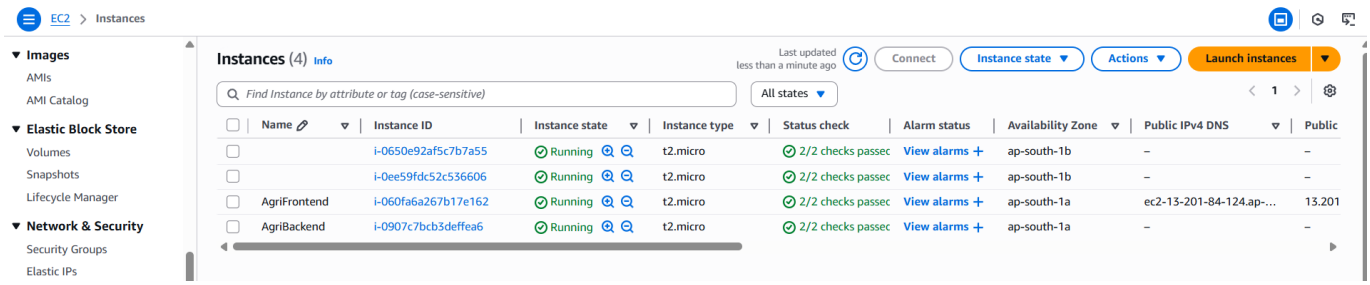

Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1 Choose launch template
Step 2 Choose instance launch options
Step 3 - optional Integrate with other services
Step 4 - optional Configure group size and scaling
Step 5 - optional Add notifications
Step 6 - optional Add tags
Step 7 Review
Integrate with other services - optional Info
Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.
Load balancing Info
Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.
Select Load balancing options
No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.
Attach to an existing load balancer
Choose from your existing load balancers.
Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.
Attach to an existing load balancer
Select the load balancers to attach
Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.
Choose from Classic Load Balancers
Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.
Select target groups
agri-backend-tg | HTTP
Application Load Balancer: agri-backend-alb

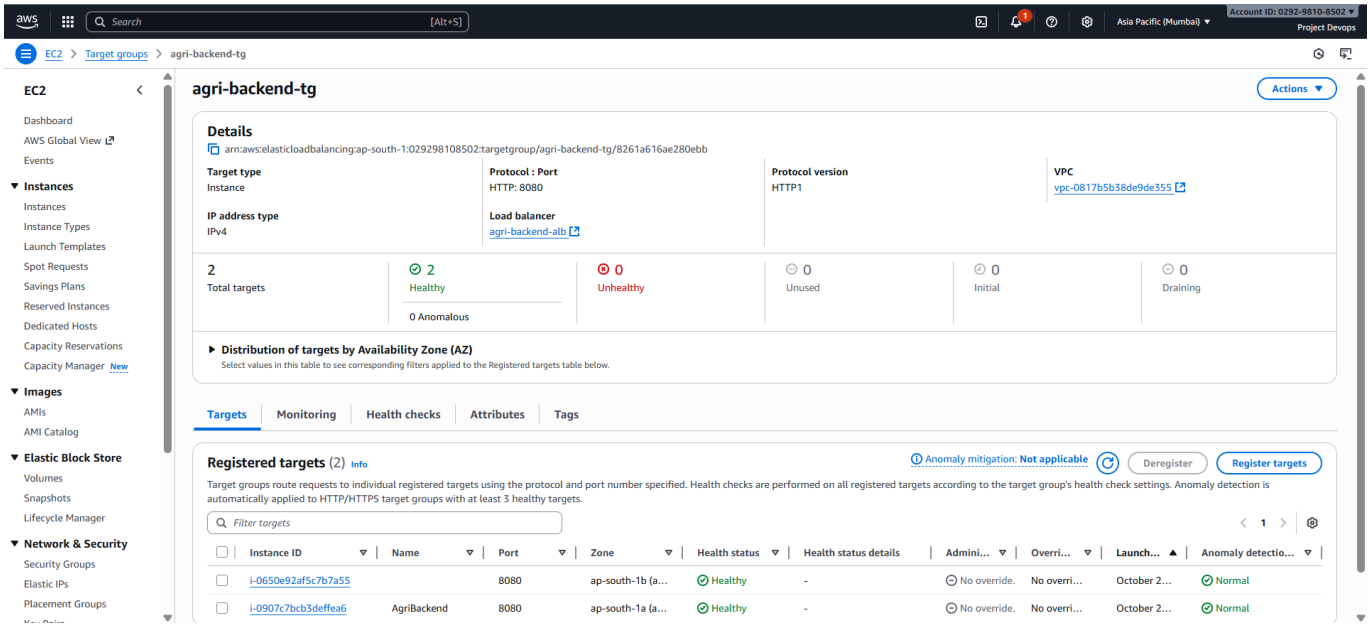

EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - optional
Integrate with other services
Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Review
Configure group size and scaling - optional Info
Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.
Group size Info
Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.
Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.
Units (number of instances)
Desired capacity
Specify your group size.
1
Scaling Info
You can resize your Auto Scaling group manually or automatically to meet changes in demand.
Scaling limits
Set limits on how much your desired capacity can be increased or decreased.
Min desired capacity
1
Equal or less than desired capacity
Max desired capacity
5
Equal or greater than desired capacity
Automatic scaling - optional
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

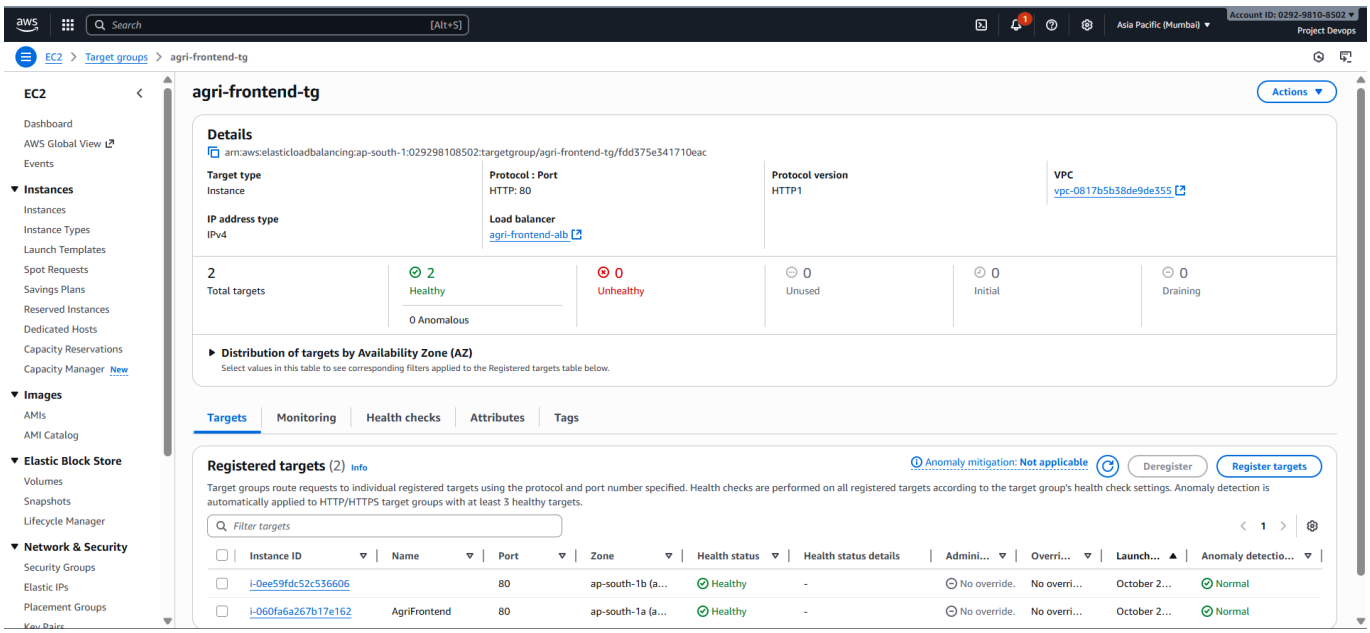

aws
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
EC2 > Auto Scaling groups > Create Auto Scaling group
Equal or less than desired capacity
Equal or greater than desired capacity
Automatic scaling - optional
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.
No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.
Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.
Scaling policy name
Target Tracking Policy
Metric type Info
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.
Average CPU utilization
Target value
50
Instance warmup Info
60
seconds
Disable scale in to create only a scale-out policy


aws
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
EC2 > Auto Scaling groups > Create Auto Scaling group
events that happen automatically to keep your group balanced, called rebalancing events.
Choose a replacement behavior depending on your availability requirements
Mixed behavior
No policy
For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.
Prioritize availability
Launch before terminating
Launch new instances and wait for them to be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.
Control costs
Terminate and launch
Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.
Flexible
Custom behavior
Set custom values for the minimum and maximum amount of available capacity. This gives you greater flexibility in setting how far below and over your desired capacity EC2 Auto Scaling goes when replacing instances.
Additional capacity settings
Capacity Reservation preference Info
Select whether you want Auto Scaling to launch instances into an existing Capacity Reservation or Capacity Reservation resource group.
Default
Auto Scaling uses the Capacity Reservation preference from your launch template.
None
Instances will not be launched into a Capacity Reservation.
Capacity Reservations only
Instances will only be launched into a Capacity Reservation. If capacity isn't available, the instances fail to launch.
Capacity Reservations first
Instances will attempt to launch into a Capacity Reservation first. If capacity isn't available, instances will run in On-Demand capacity.


Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
EC2 > Auto Scaling groups
Auto Scaling groups (2) Info
Last updated less than a minute ago
Launch configurations
Launch templates
Actions
Create Auto Scaling group
Search your Auto Scaling groups
Name | Launch template/configuration | Instances | Status | Desired capacity | Min | Max | Availability Zones | Creation time
AgriBackendASG | AgriBackend | Version Default | 1 | - | 1 | 1 | 5 | 2 Availability Zones | Sat Oct 25 2025 18:55:5...
AgriFrontendASG | AgriFrontend | Version Default | 1 | - | 1 | 1 | 5 | 2 Availability Zones | Sat Oct 25 2025 18:52:5...


EC2 > Instances
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Instances (4) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public
i-0650e92af5c7b7a55 Running t2.micro 2/2 checks passec View alarms ap-south-1b
i-0ee59fdc52c536606 Running t2.micro 2/2 checks passec View alarms ap-south-1b
AgriFrontend i-060fa6a267b17e162 Running t2.micro 2/2 checks passec View alarms ap-south-1a ec2-13-201-84-124.ap-... 13.201
AgriBackend i-0907c7bcb3deffea6 Running t2.micro 2/2 checks passec View alarms ap-south-1a


Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
EC2 > Target groups > agri-backend-tg
EC2
Dashboard
AWS Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
agri-backend-tg
Actions
Details
arn:aws:elasticloadbalancing:ap-south-1:029298108502:targetgroup/agri-backend-tg/8261a616ae280ebb
Target type
Instance
Protocol : Port
HTTP: 8080
Protocol version
HTTP1
VPC
vpc-0817b5b38de9de355
IP address type
IPv4
Load balancer
agri-backend-alb
2 Total targets
2 Healthy
0 Anomalous
0 Unhealthy
0 Unused
0 Initial
0 Draining
Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.
Targets
Monitoring
Health checks
Attributes
Tags
Registered targets (2) Info
Anomaly mitigation: Not applicable
Deregister
Register targets
Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.
Filter targets
Instance ID | Name | Port | Zone | Health status | Health status details | Admini... | Overri... | Launch... | Anomaly detectio...
i-0650e92af5c7b7a55 8080 ap-south-1b (a... Healthy - No override. No overri... October 2... Normal
i-0907c7bcb3deffea6 AgriBackend 8080 ap-south-1a (a... Healthy - No override. No overri... October 2... Normal


Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0292-9810-8502
Project Devops
EC2 > Target groups > agri-frontend-tg
EC2
Dashboard
AWS Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
agri-frontend-tg
Actions
Details
arn:aws:elasticloadbalancing:ap-south-1:029298108502:targetgroup/agri-frontend-tg/fdd375e341710eac
Target type
Instance
Protocol : Port
HTTP: 80
Protocol version
HTTP1
VPC
vpc-0817b5b38de9de355
IP address type
IPv4
Load balancer
agri-frontend-alb
2 Total targets
2 Healthy
0 Anomalous
0 Unhealthy
0 Unused
0 Initial
0 Draining
Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.
Targets
Monitoring
Health checks
Attributes
Tags
Registered targets (2) Info
Anomaly mitigation: Not applicable
Deregister
Register targets
Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.
Filter targets
Instance ID | Name | Port | Zone | Health status | Health status details | Admini... | Overri... | Launch... | Anomaly detectio...
i-0ee59fdc52c536606 80 ap-south-1b (a... Healthy - No override. No overri... October 2... Normal
i-060fa6a267b17e162 AgriFrontend 80 ap-south-1a (a... Healthy - No override. No overri... October 2... Normal