Preeti.3tier-architecture-AWS

Pre-requisites

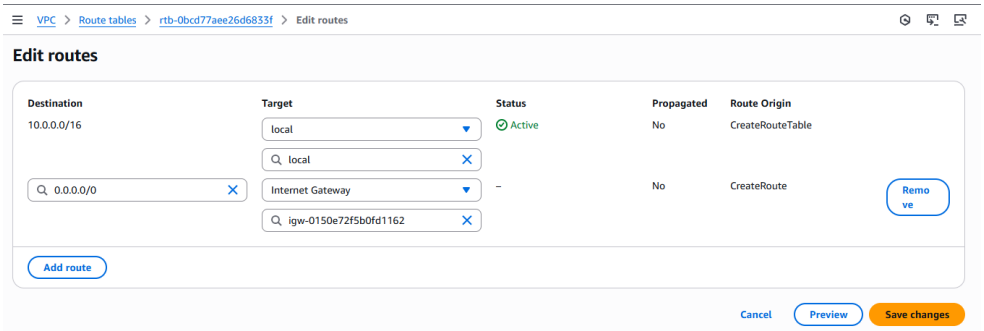
1. Download Code from Github to local

git clone https://github.com/aws-samples/aws-three-tier-web-architecture-workshop.git

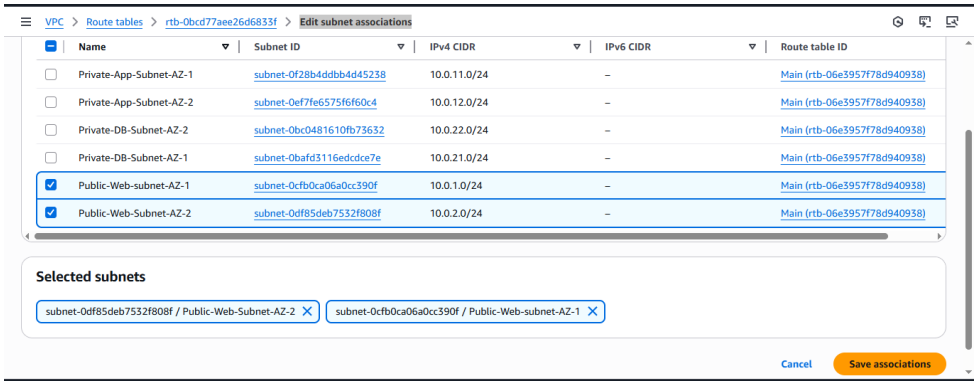
2. S3 Bucket Creation → **3tier-webapp-preeti**
3. IAM EC2 Instance Role Creation → EC2-SSM-S3-3tier
    a. AmazonSSMManagedInstanceCore
    b. AmazonS3ReadOnlyAccess

Network and security

1. Foundation network (VPC)
    c. **VPC**: 10.0.0.0/16 →3tier-vpc
    d. Private-App-Subnet-AZ-1 →10.0.11.0/24
    e. Private-App-Subnet-AZ-2 → 10.0.12.0/24
    f. Private-DB-Subnet-AZ-1 →10.0.21.0/24
    g. Private-DB-Subnet-AZ-2 →10.0.22.0/24
    h. Public-Web-Subnet-AZ-1 →10.0.1.0/24
    i. Public-Web-Subnet-AZ-2 →10.0.2.0/24
2. **IGW** attached to VPC. → 3-tier-igw attach to 3tier-vpc
3. **1 NAT Gateway** in **one** public subnet.
    a. 3tier-NAT-AZ1 →Public-Web-subnet-AZ-1→connectivity-type:Public→allocate-elastic-ip
    b. 3tier-NAT-AZ2 →Public-Web-subnet-AZ-2→connectivity-type:Public→allocate-elastic-ip
4. **Route tables:**
    a. PublicRouteTable →
        i. Edit routes→add route

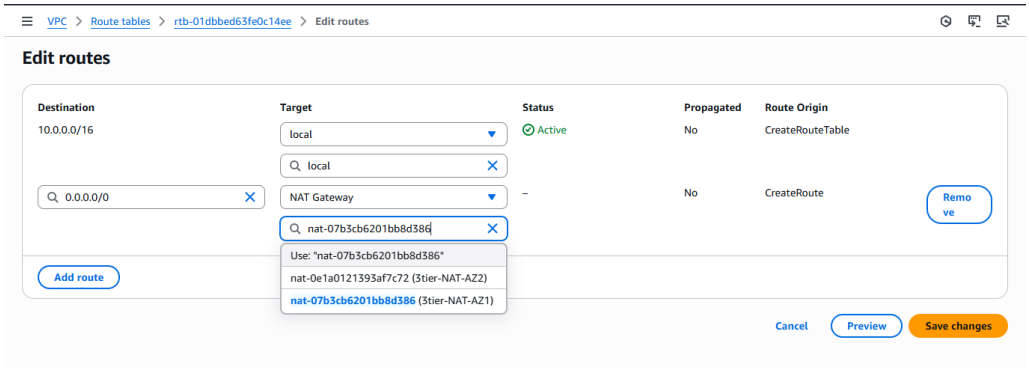


        ii. Edit subnet associations

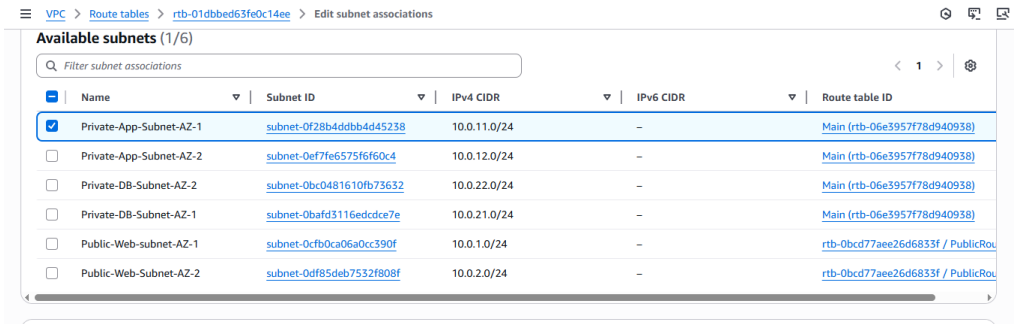


b. **PrivateRouteTable-AZ1→**

i. **Edit routes→add route(**use target natgateway pf AZ1**)**



ii. **Edit subnet associations**



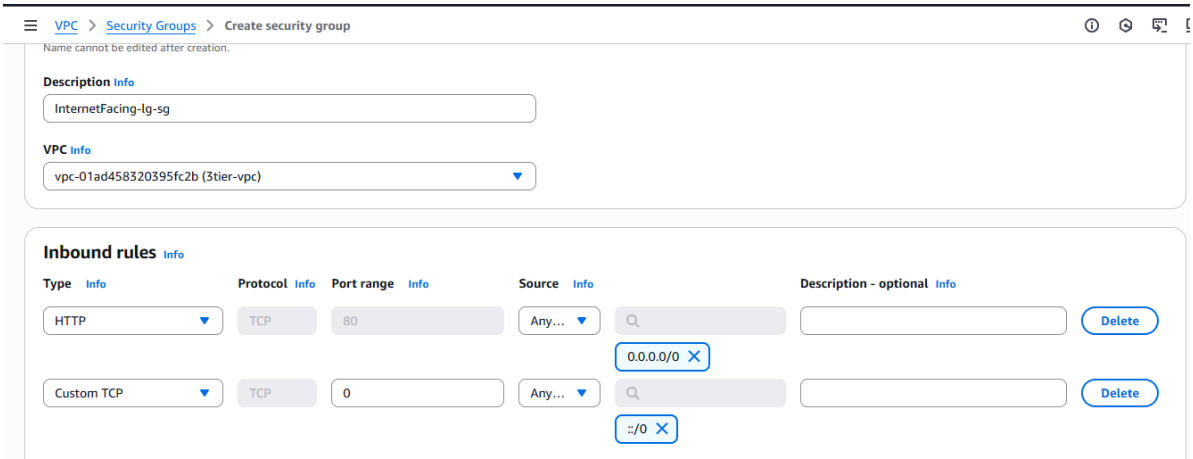
c. **PrivateRouteTable-AZ2→**

i. **Edit routes→add route**

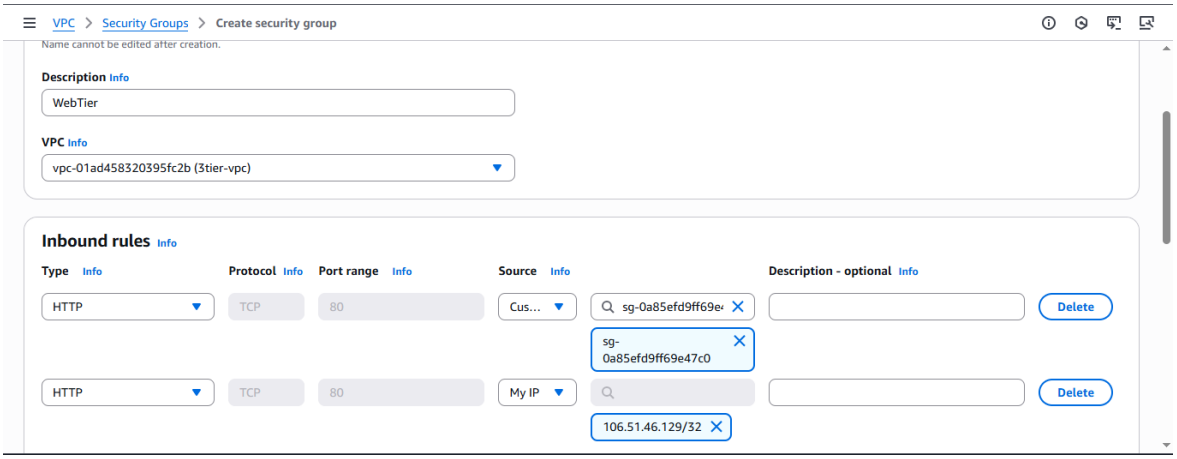
ii. **Edit subnet associations**

5. **Security Groups**

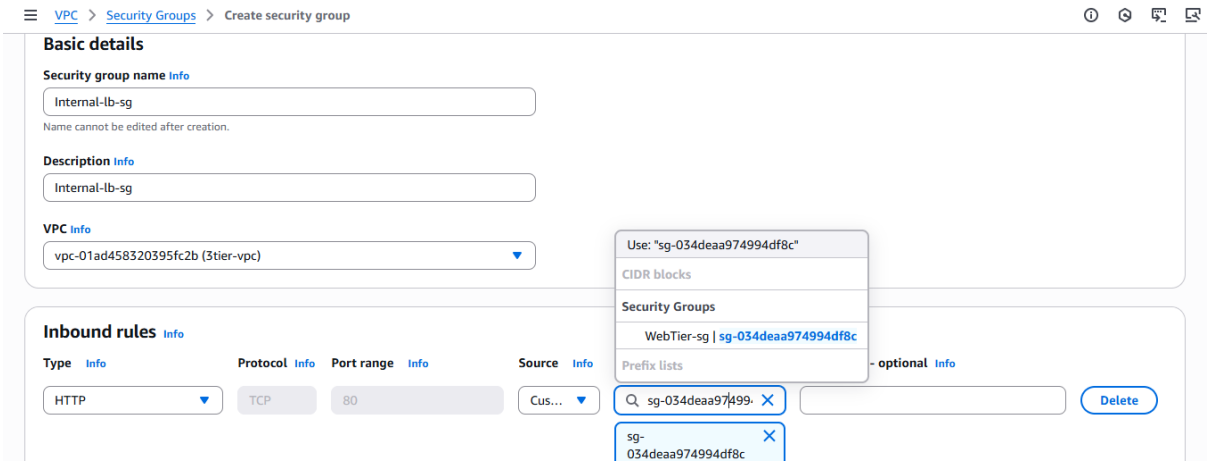
a. **InternetFacing-lg-sg**



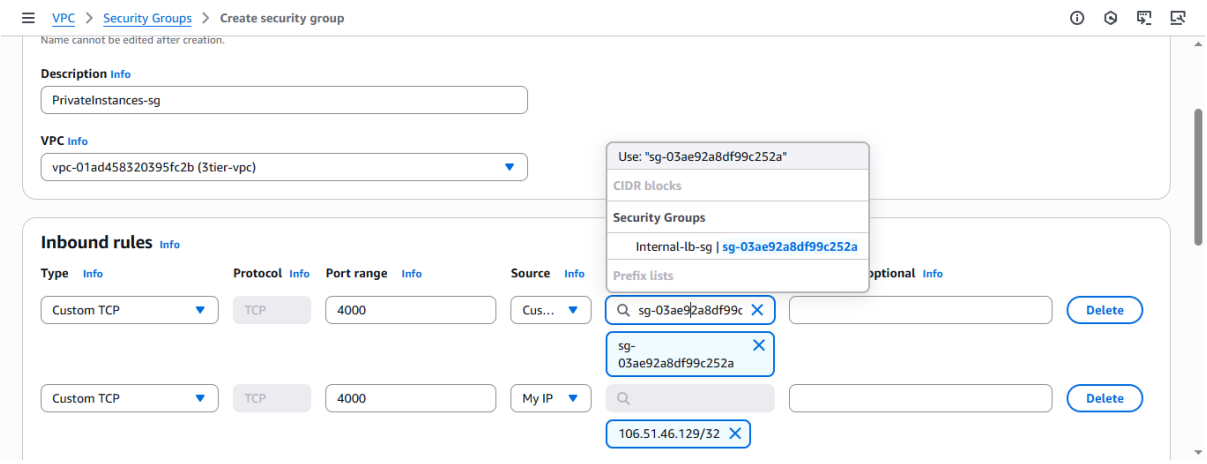
b. WebTier-sg



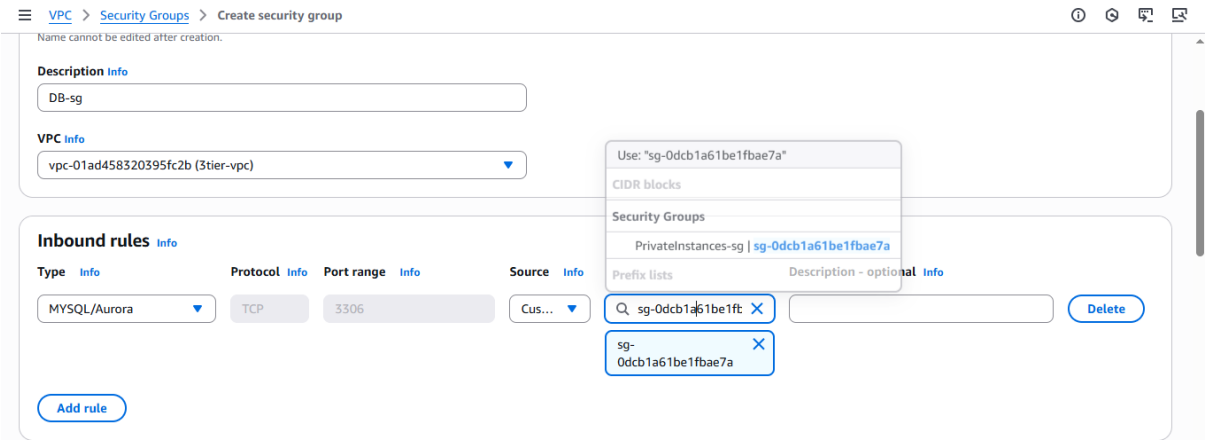
c. Internal-lg-sg



d. PrivateInstances-sg

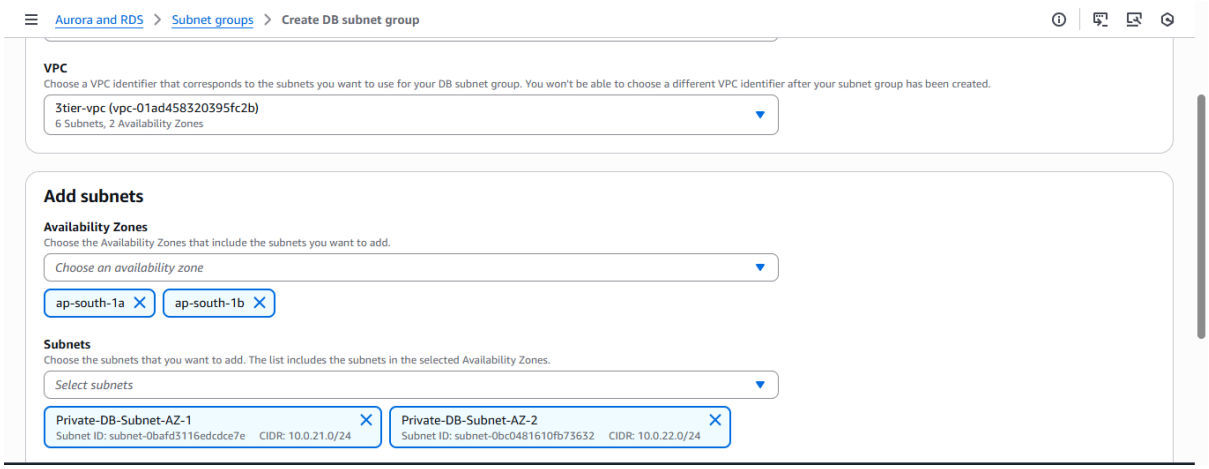


e. DB-SG



DB-deployment(Aurora MySQL (DB tier))

1. Go to RDS → subnet groups→ create DB subnet group



2. Create DB

Aurora and RDS > 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

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.

aws | Search [Alt+S] | Asia Pacific (Mumbai) | Account ID: 1952-7567-5477 | Preeti B

Aurora and RDS > Create database

Master username Info

Type a login ID for the master user of your DB instance.

admin

1 to 32 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

••••••••

Password strength Weak

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

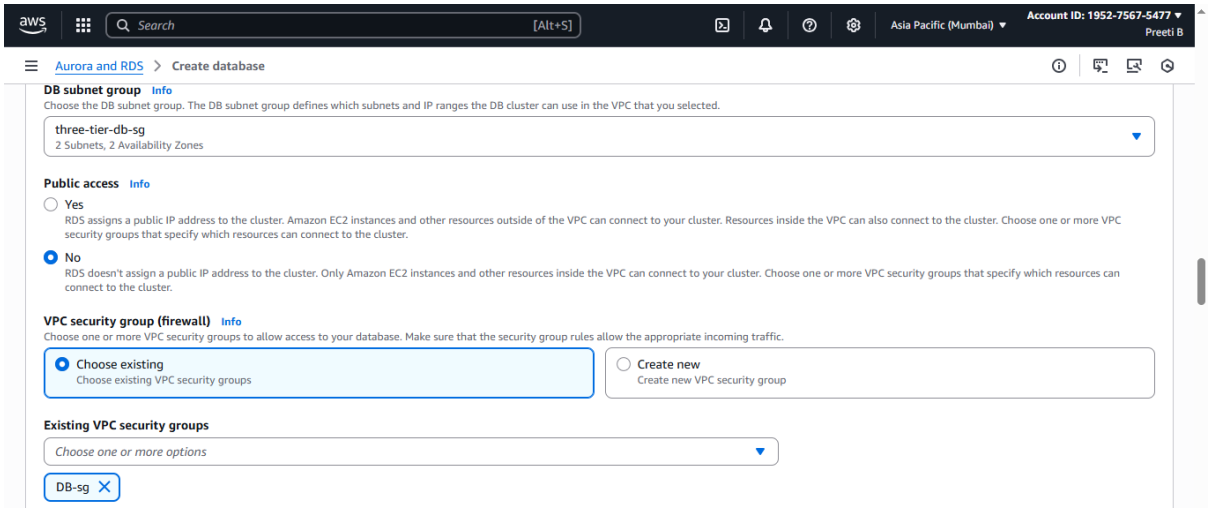
Confirm master password | Info

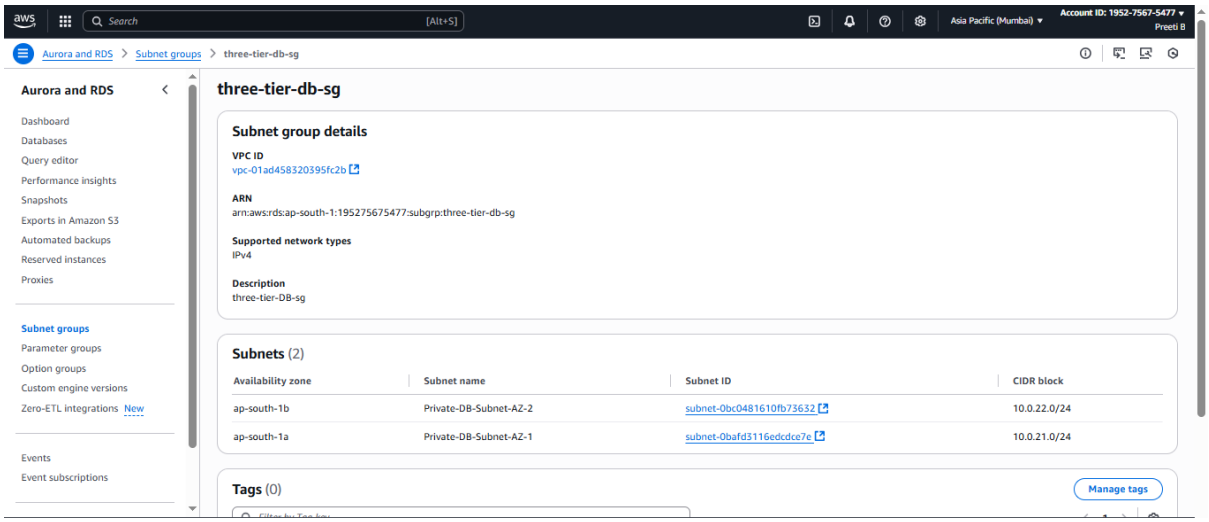
••••••••

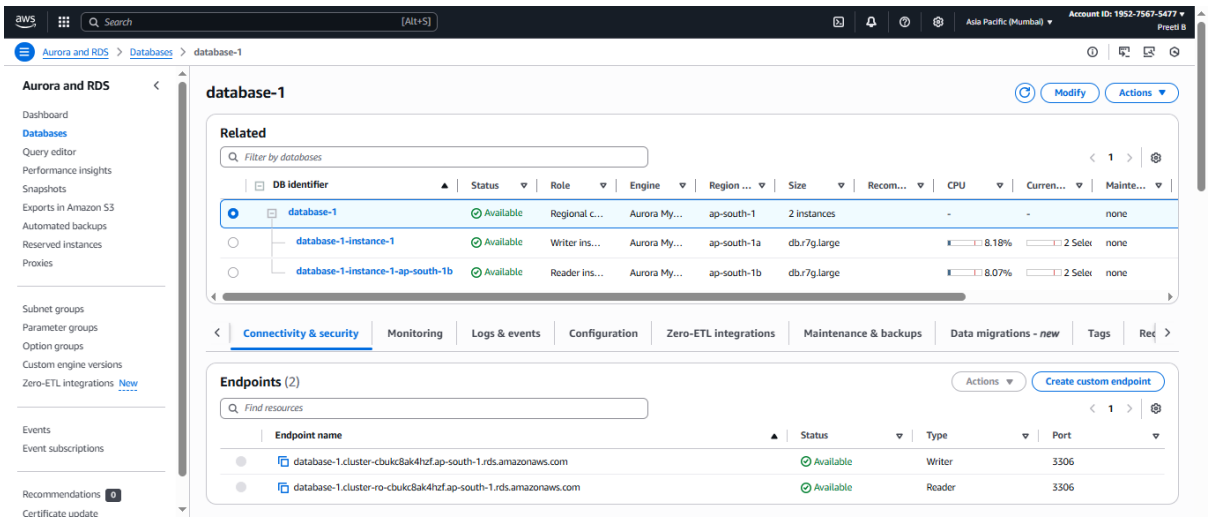
Availability & durability

Multi-AZ deployment Info

- (●) Create an Aurora Replica or Reader node in a different AZ (recommended for scaled availability)
  Creates an Aurora Replica for fast failover and high availability.
- ( ) Don't create an Aurora Replica

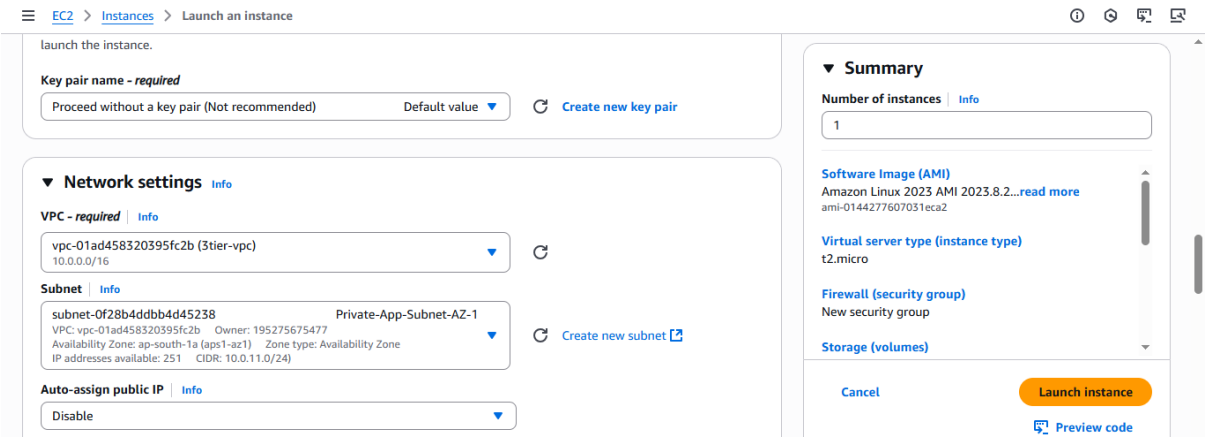


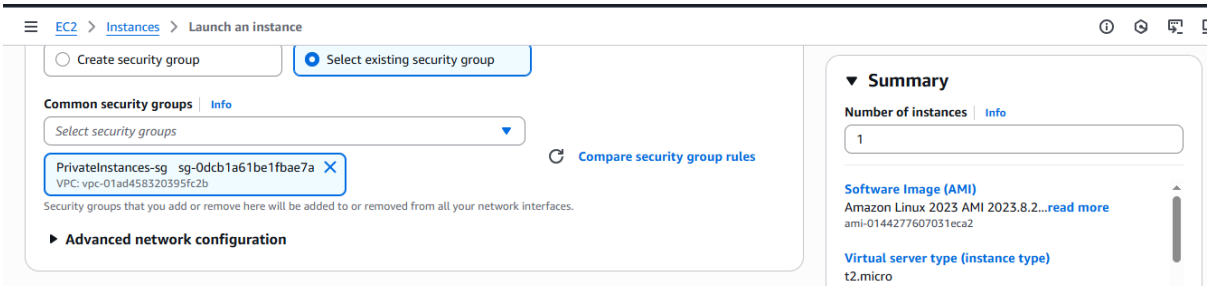


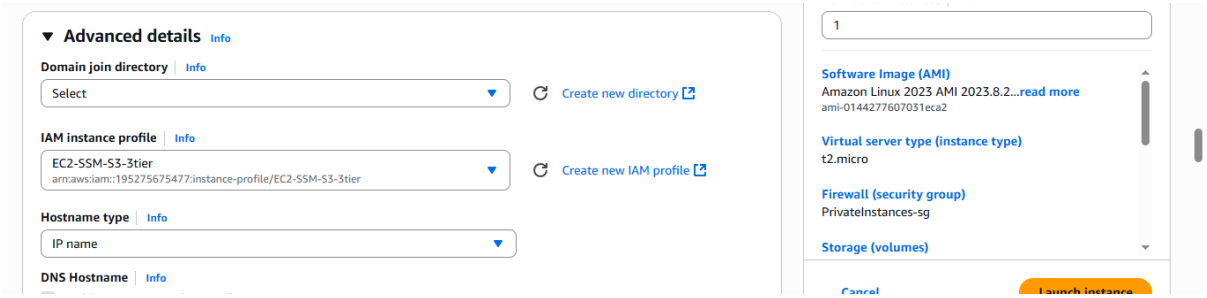


App Tier Instance Deployment:

1. **Create EC2-instance →** AppServer







2. Connect to SSM

a. sudo -su ec2-user
b. ping 8.8.8.8
c. sudo dnf install -y https://dev.mysql.com/get/mysql84-community-release-el9-1.noarch.rpm
d. sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql
e. sudo dnf install -y mysql
f. connect to DB→copy RDS writer endpoint
    a. mysql -h database-1-instance-1.cbukc8ak4hzf.ap-south-1.rds.amazonaws.com -u admin -p
    b. CREATE DATABASE webappdb;
    c. SHOW DATABASES;
    d. USE webappdb;
    e. CREATE TABLE IF NOT EXISTS transactions(id INT NOT NULL AUTO_INCREMENT, amount DECIMAL(10,2), description VARCHAR(100), PRIMARY KEY(id));
    f. SHOW TABLES;

g. INSERT INTO transactions (amount,description) VALUES ('400','groceries');
h. SELECT * FROM transactions;
i. Exit

g. Go to s3 and upload files of app-tier by updating DbConfig.js with your values

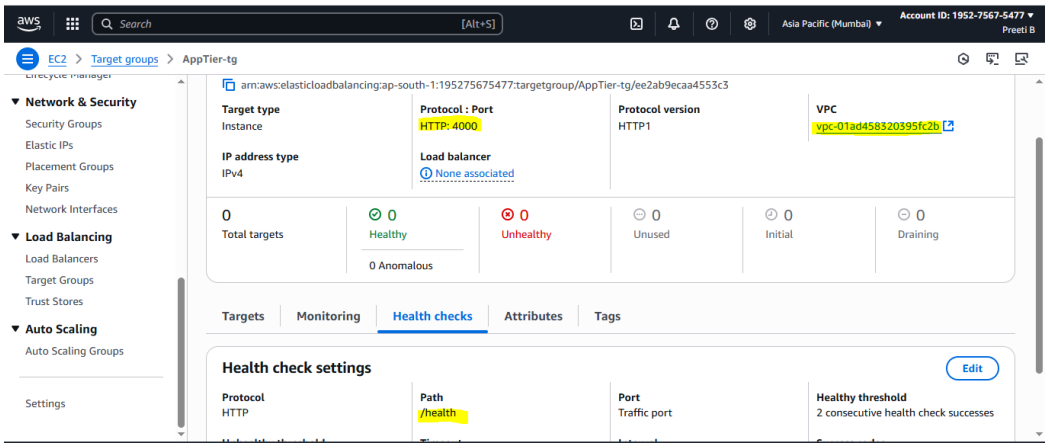
```
module.exports = Object.freeze({
    DB_HOST : 'database-1-instance-1.cbukc8ak4hzf.ap-south-1.rds.amazonaws.com',
    DB_USER : 'admin',
    DB_PWD : '[redacted]',
    DB_DATABASE : 'webappdb'
});
```

h. Go to session manager
   a. curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
   b. source ~/.bashrc
   c. nvm install 16
   d. nvm use 16
   e. npm install -g pm2
   f. cd ~/
   g. aws s3 cp s3://3tier-webapp-preeti/app-tier/ app-tier –recursive
   h. cd ~/app-tier
   i. npm install
   j. pm2 start index.js
   k. pm2 list
   l. pm2 logs
   m. pm2 startup
   n. sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user --hp /home/ec2-user
   o. pm2 save
   p. curl http://localhost:4000/health
   q. curl http://localhost:4000/transaction

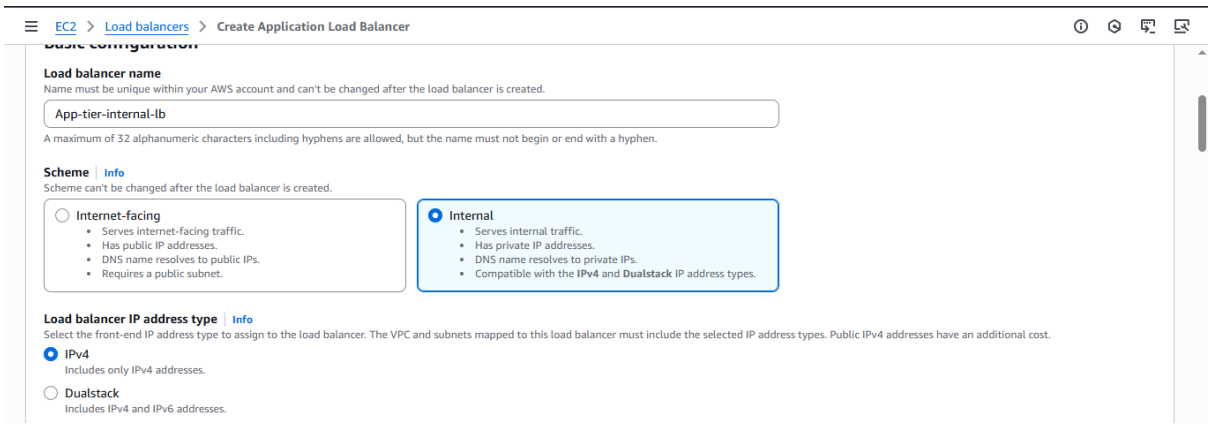
Internal Load Balancing and Auto Scaling

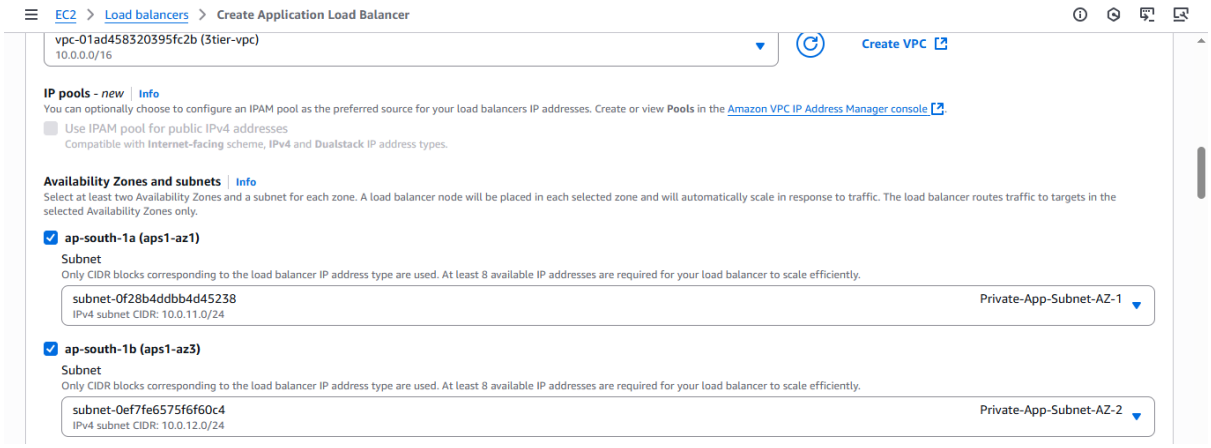
1. Navigate to EC2 dashboard. Select the app tier instance we created and under Actions select Image and templates. Create AMI of it.

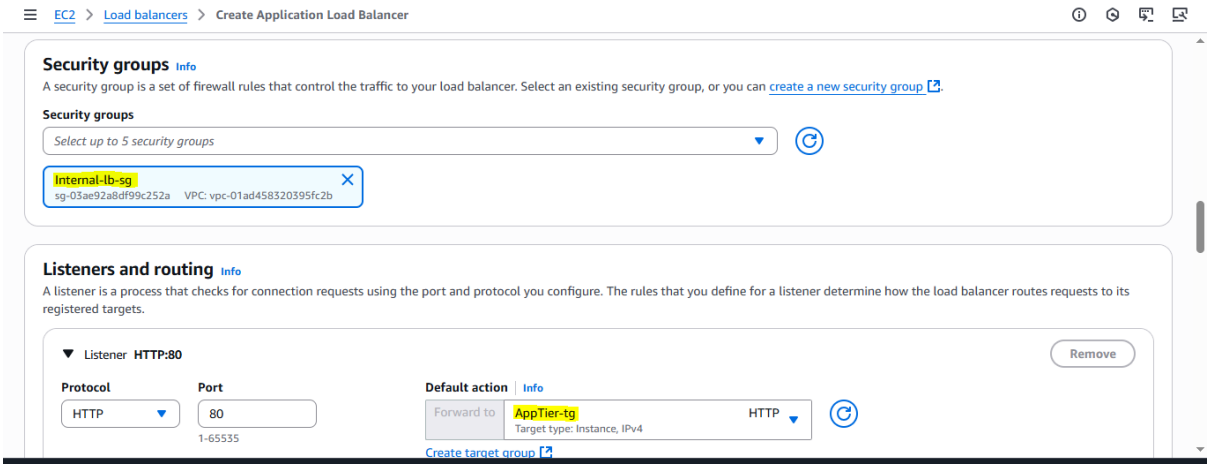
2. Goto target group and create one



3. Internal Load Balancer

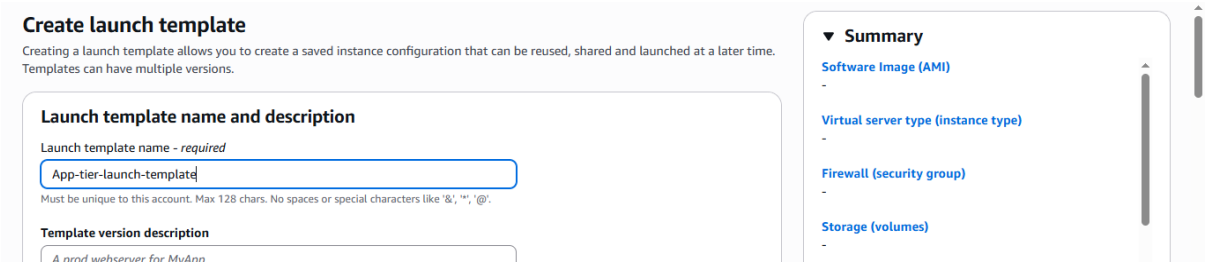


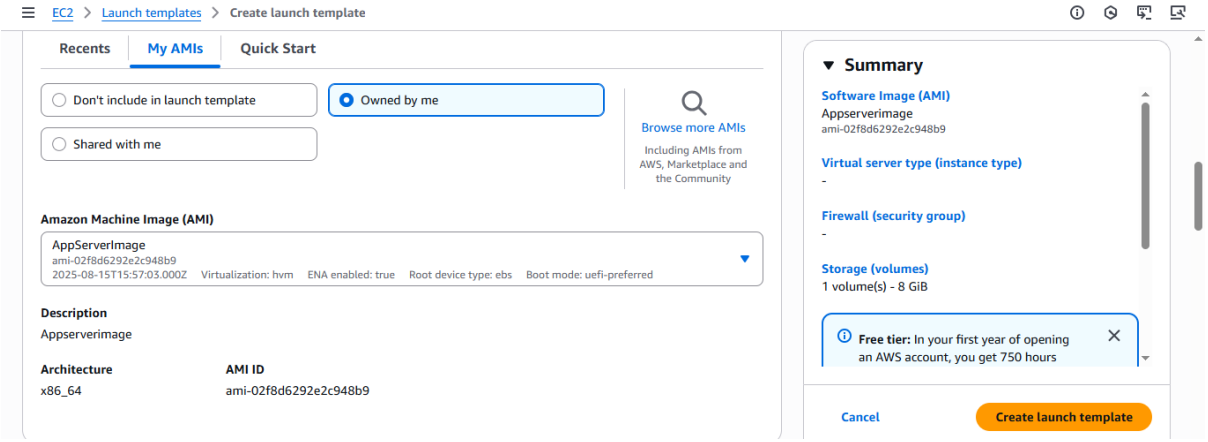


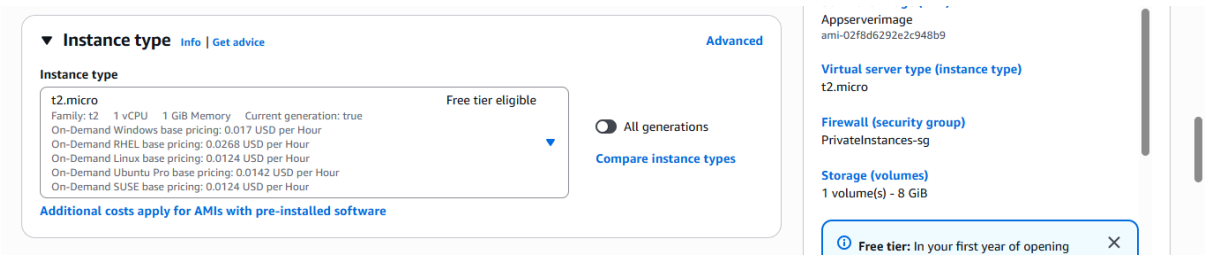


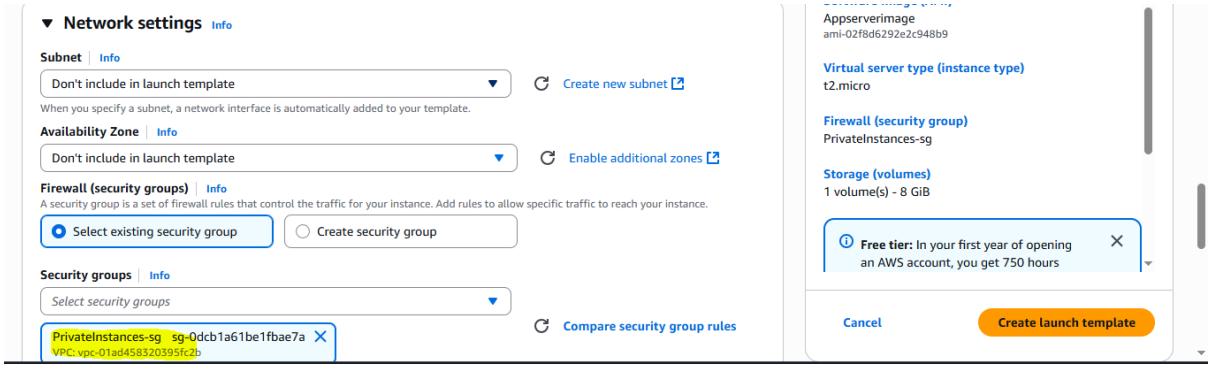
Create load balancer

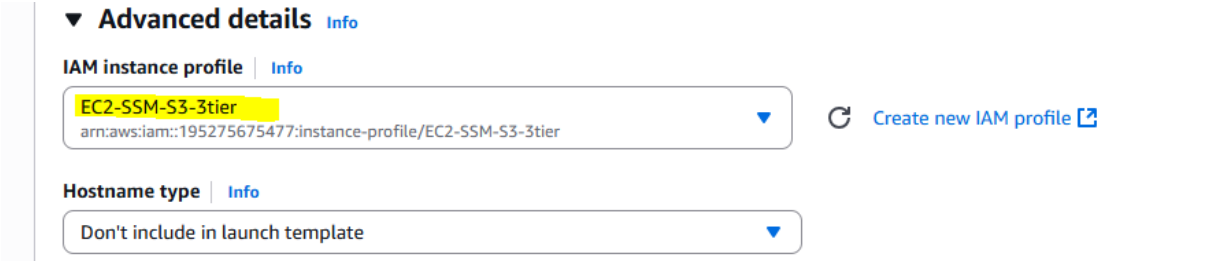
4. Create launch template





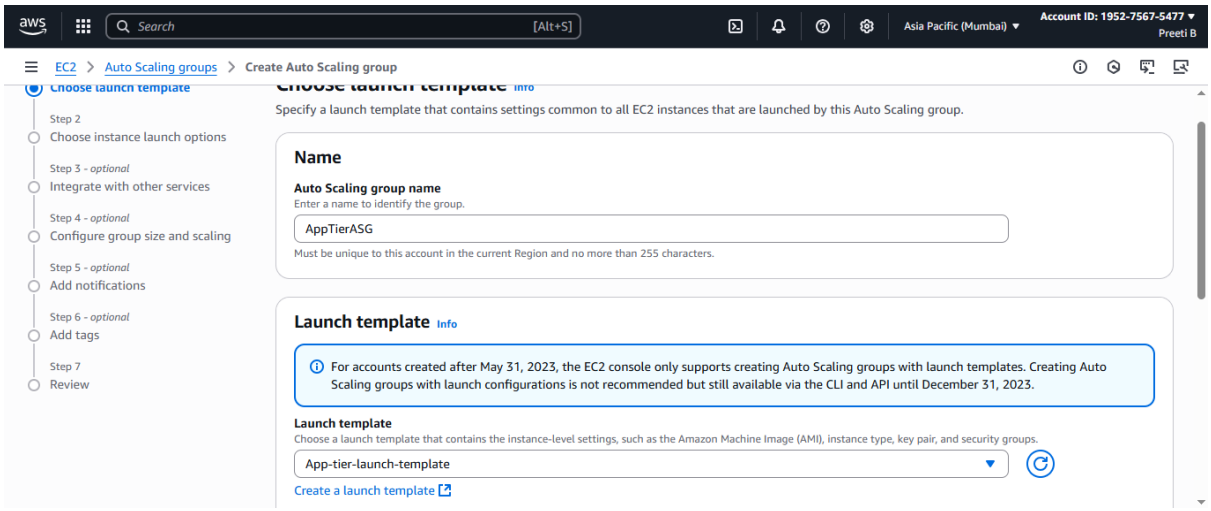


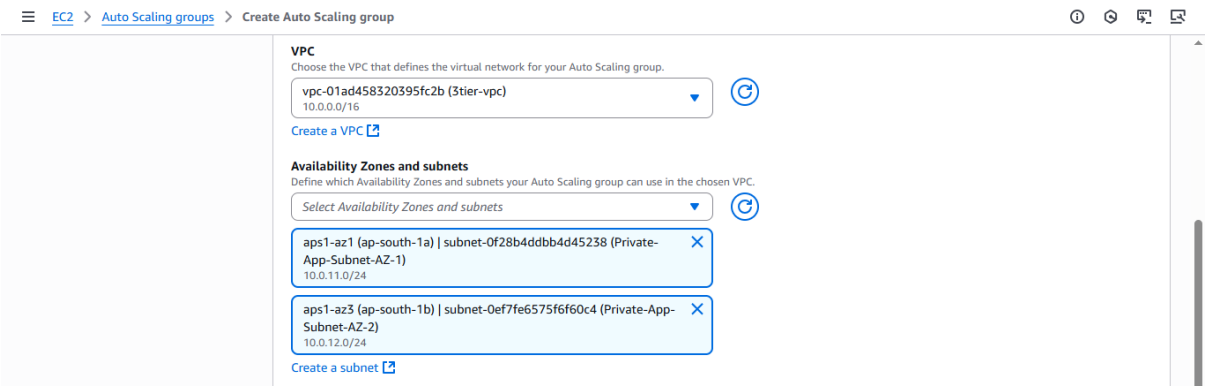


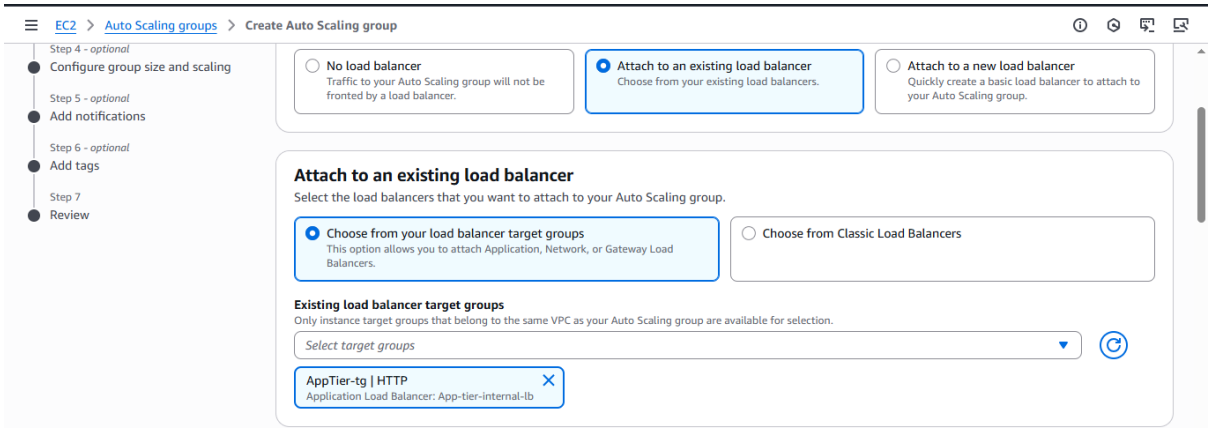


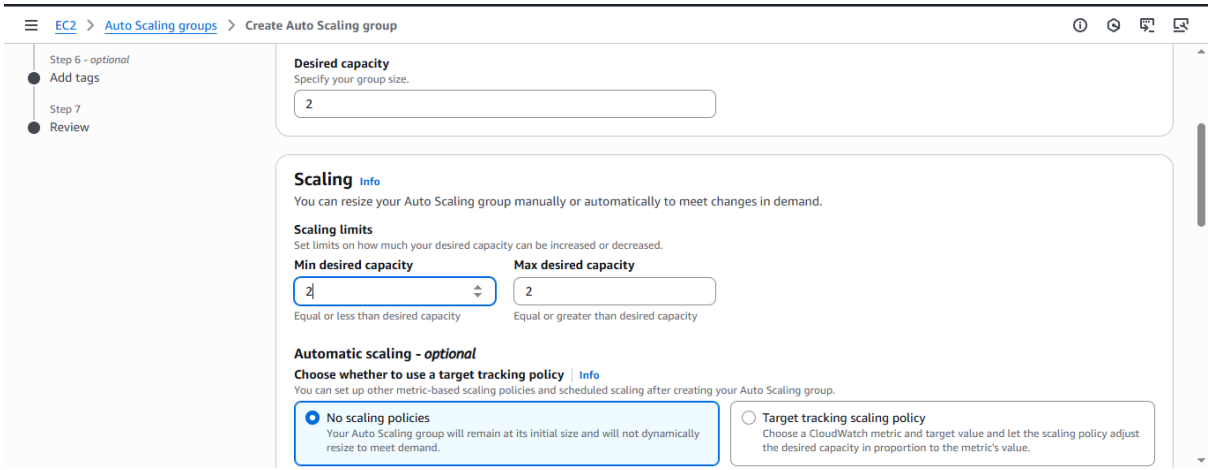
Create auto-scaling-groups

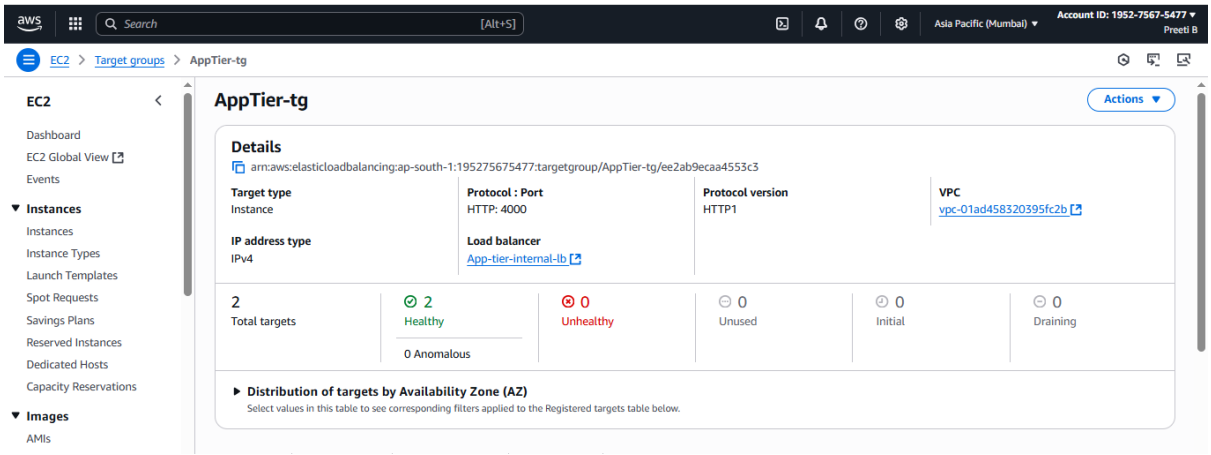
5. Create ASG





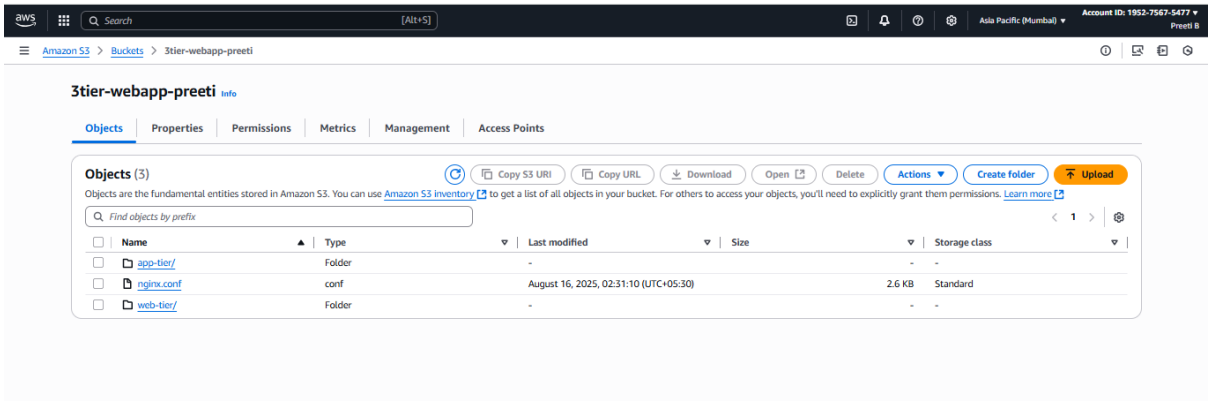


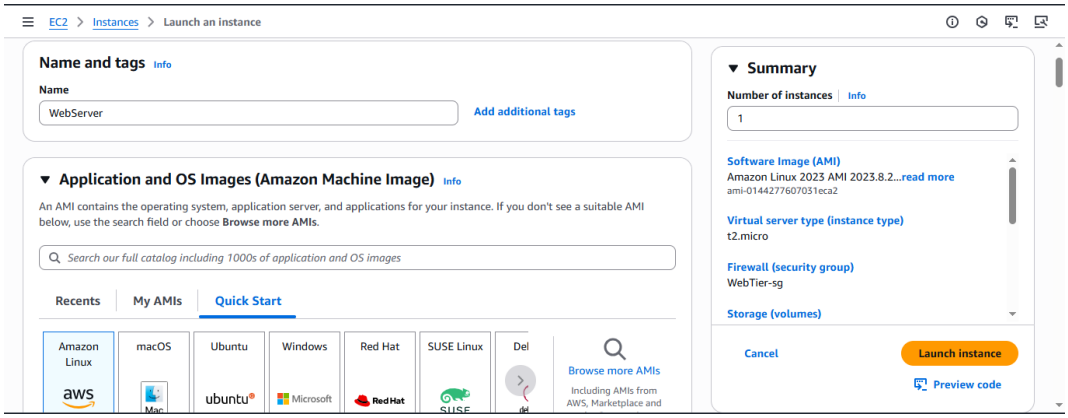


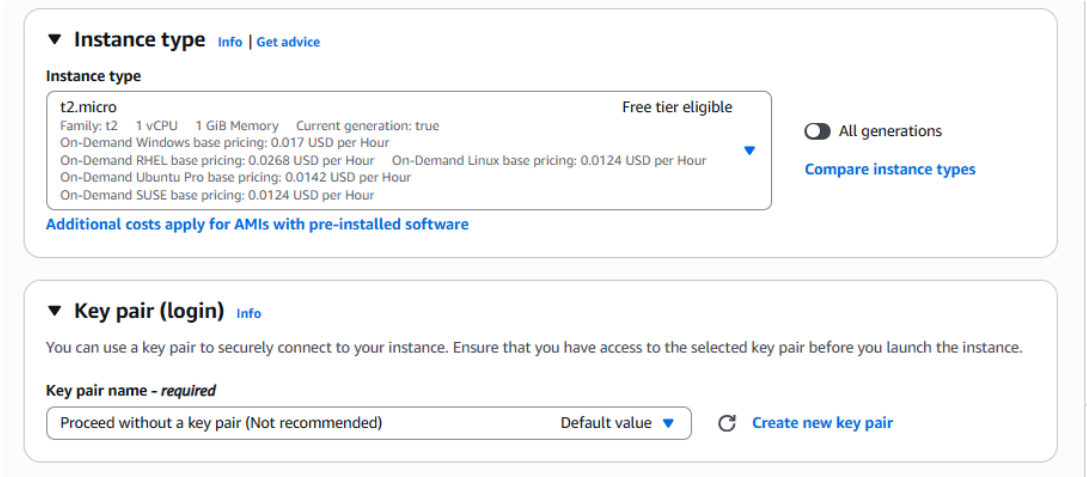


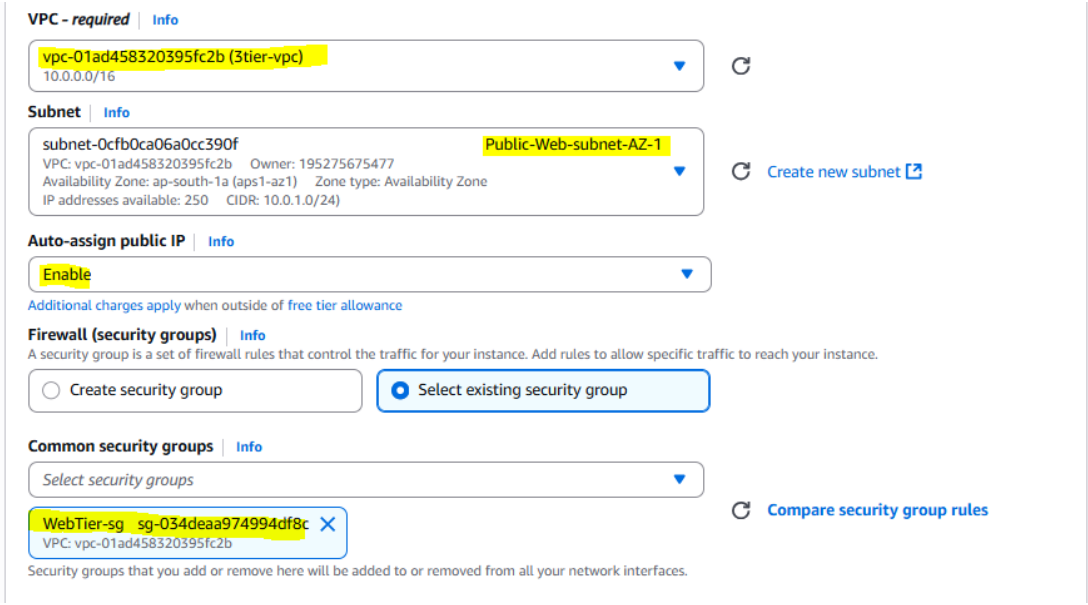
Web Tier Instance Deployment

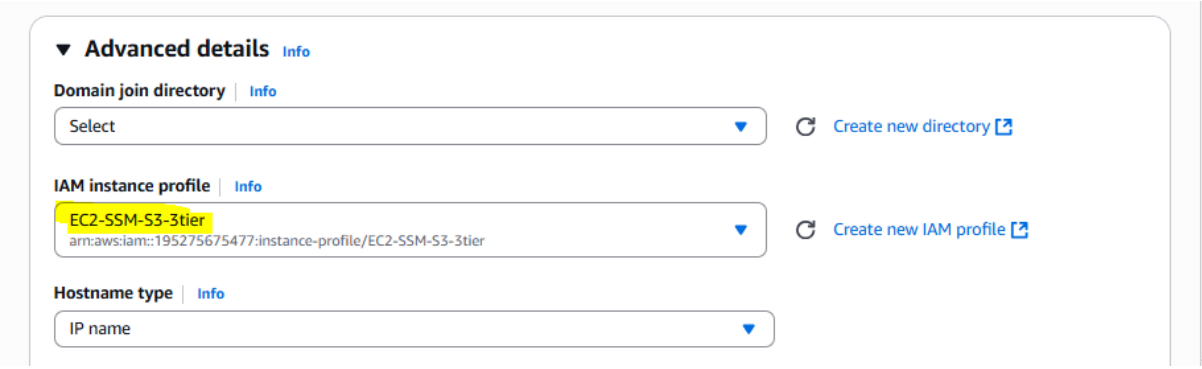
1. Update Config File→ **application-code/nginx.conf**
2. Upload **webtier** and **nginx.conf** to s3 bucket.
3. Create ec2-instance→WebServer


Amazon S3 > Buckets > 3tier-webapp-preeti
3tier-webapp-preeti Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (3)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
app-tier/
Folder
nginx.conf
conf
August 16, 2025, 02:31:10 (UTC+05:30)
2.6 KB
Standard
web-tier/
Folder


EC2 > Instances > Launch an instance
Name and tags Info
Name
WebServer
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose Browse more AMIs.
Search our full catalog including 1000s of application and OS images
Recents
My AMIs
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Browse more AMIs
Including AMIs from AWS, Marketplace and
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...read more
ami-0144277607031eca2
Virtual server type (instance type)
t2.micro
Firewall (security group)
WebTier-sg
Storage (volumes)
Cancel
Launch instance
Preview code

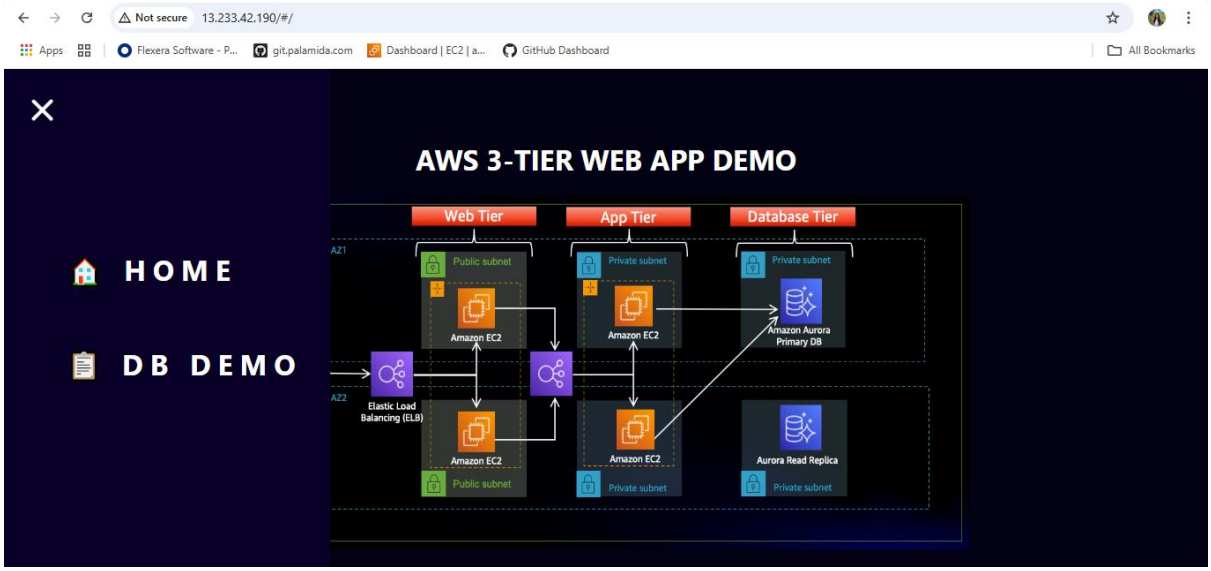

Instance type Info | Get advice
Instance type
t2.micro
Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0268 USD per Hour On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
Proceed without a key pair (Not recommended)
Default value
Create new key pair

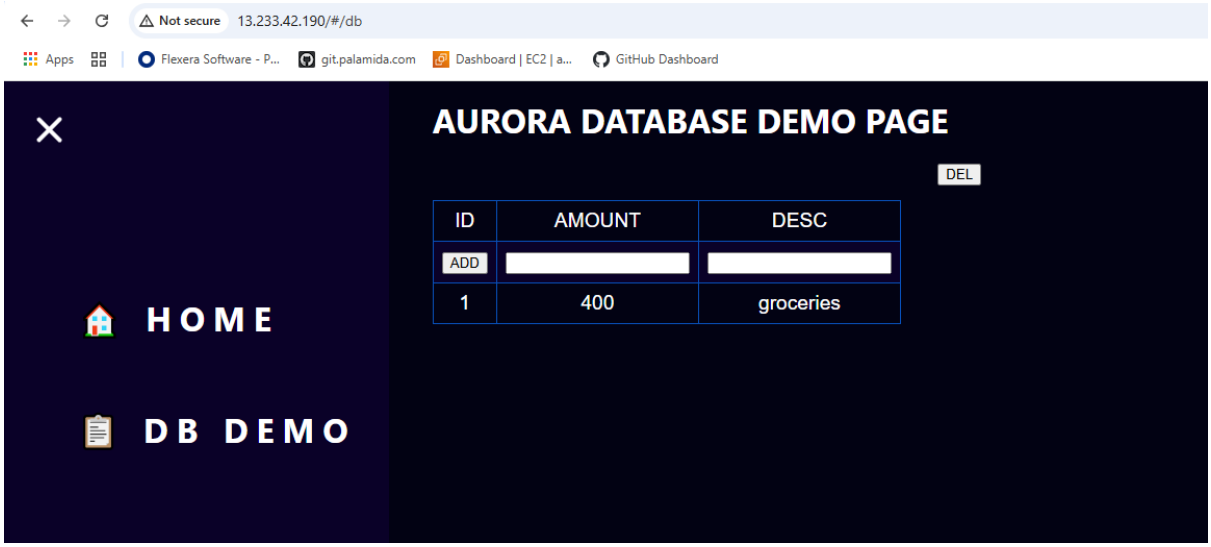




4. Connect to session manager
    a. sudo -su ec2-user
    b. ping 8.8.8.8
    c. curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
    d. source ~/.bashrc
    e. nvm install 16
    f. nvm use 16
    g. cd ~/
    h. aws s3 cp s3://3tier-webapp-preeti/web-tier/ web-tier –recursive
    i. cd ~/web-tier
    j. npm install
    k. npm run build
    l. sudo dnf install -y nginx
    m. cd /etc/nginx
    n. ls
    o. sudo cp nginx.conf nginx.conf_backup

p. sudo rm nginx.conf
q. sudo aws s3 cp s3://3tier-webapp-preeti/nginx.conf .
r. sudo service nginx restart
s. chmod -R 755 /home/ec2-user
t. sudo chkconfig nginx on





5. create image of webserver→WebServerImage
6. create target group → WebTier-tg

EC2 > Target groups > Create target group

Target group name

WebTier-tg

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol
Protocol for load balancer-to-target communication. Can't be modified after creation.

HTTP

Port
Port number where targets receive traffic. Can be overridden for individual targets during registration.

80

1-65535

IP address type
Only targets with the indicated IP address type can be registered to this target group.

- IPv4
  Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.
- IPv6
  Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). Learn more

VPC
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

vpc-01ad458320395fc2b (3tier-vpc)
10.0.0.0/16

Create VPC

EC2 > Target groups > Create target group

- gRPC
  Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path
Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/health

Up to 1024 characters allowed.

▸ **Advanced health check settings**

7. create external-load-balencer→ Web-tier-external-lb

EC2 > Load balancers > Create Application Load Balancer

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

Web-tier-external-lb

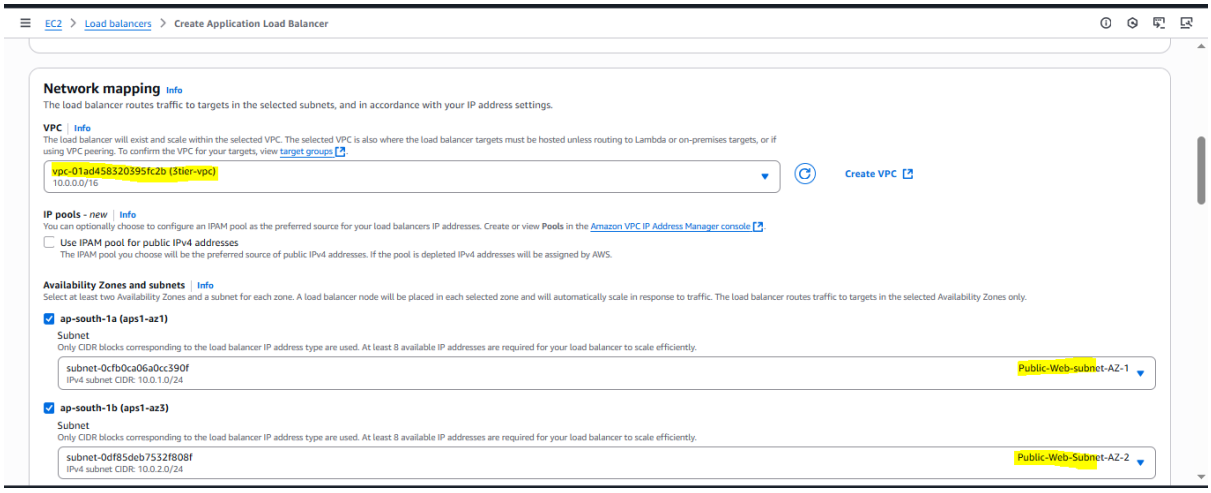
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

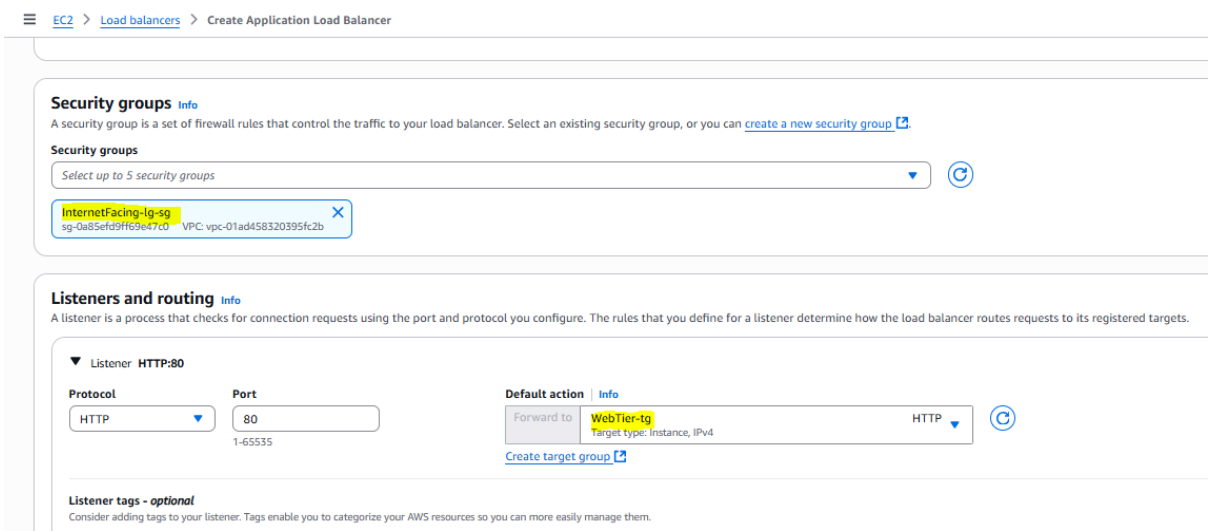
Scheme | Info
Scheme can't be changed after the load balancer is created.

- Internet-facing
  - Serves internet-facing traffic.
  - Has public IP addresses.
  - DNS name resolves to public IPs.
  - Requires a public subnet.
- Internal
  - Serves internal traffic.
  - Has private IP addresses.
  - DNS name resolves to private IPs.
  - Compatible with the **IPv4** and **Dualstack** IP address types.

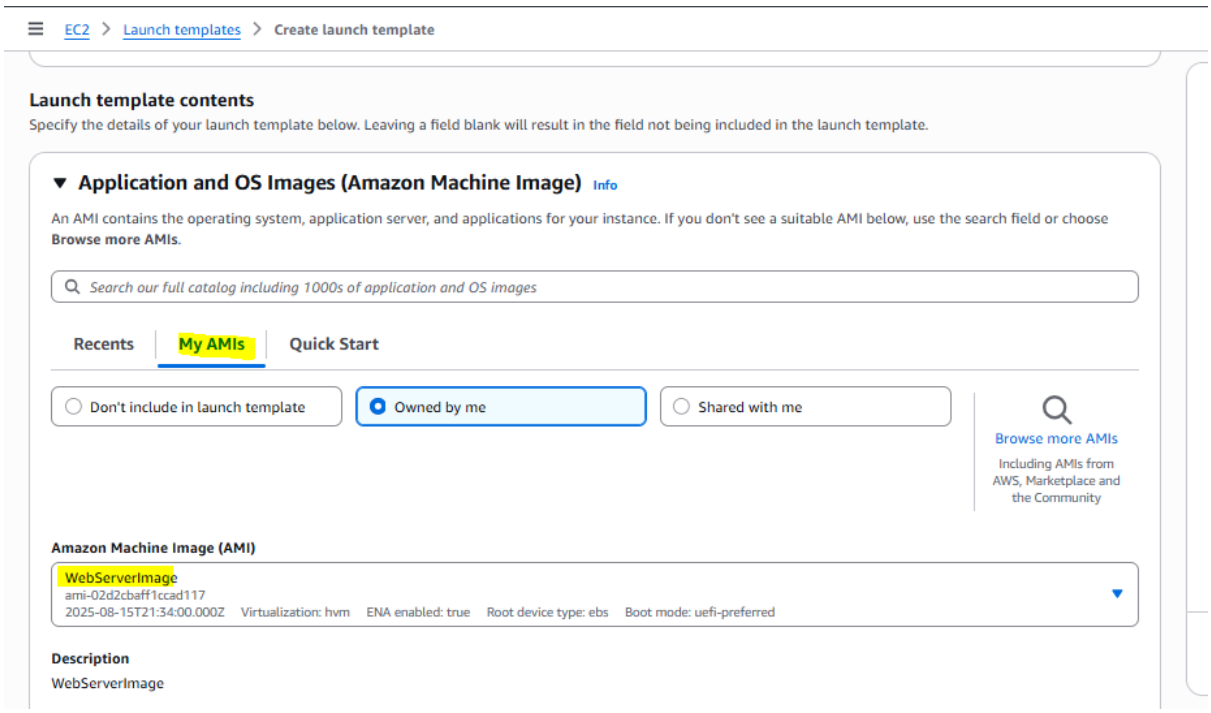
Load balancer IP address type | Info
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

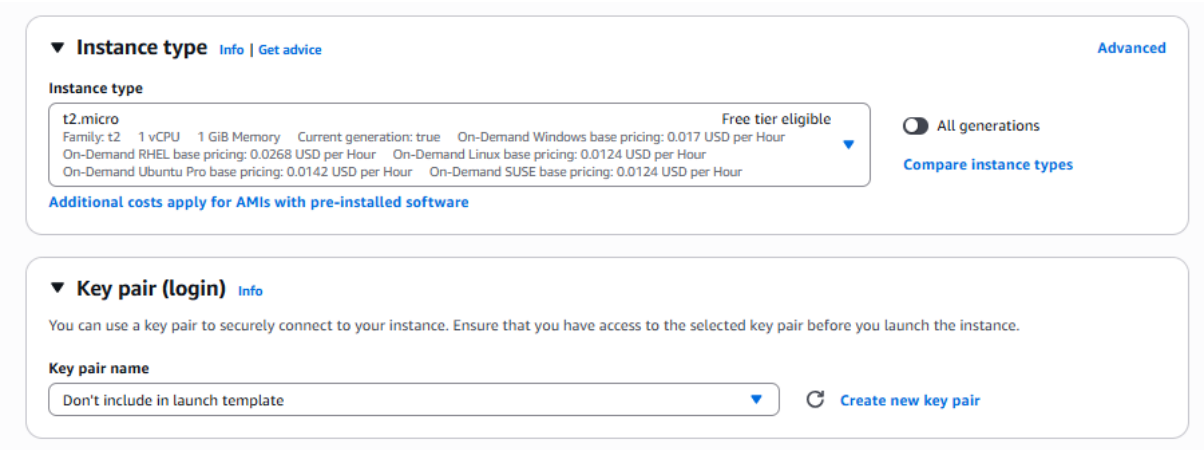
- IPv4
  Includes only IPv4 addresses.
- Dualstack
  Includes IPv4 and IPv6 addresses.
- Dualstack without public IPv4
  Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers only.

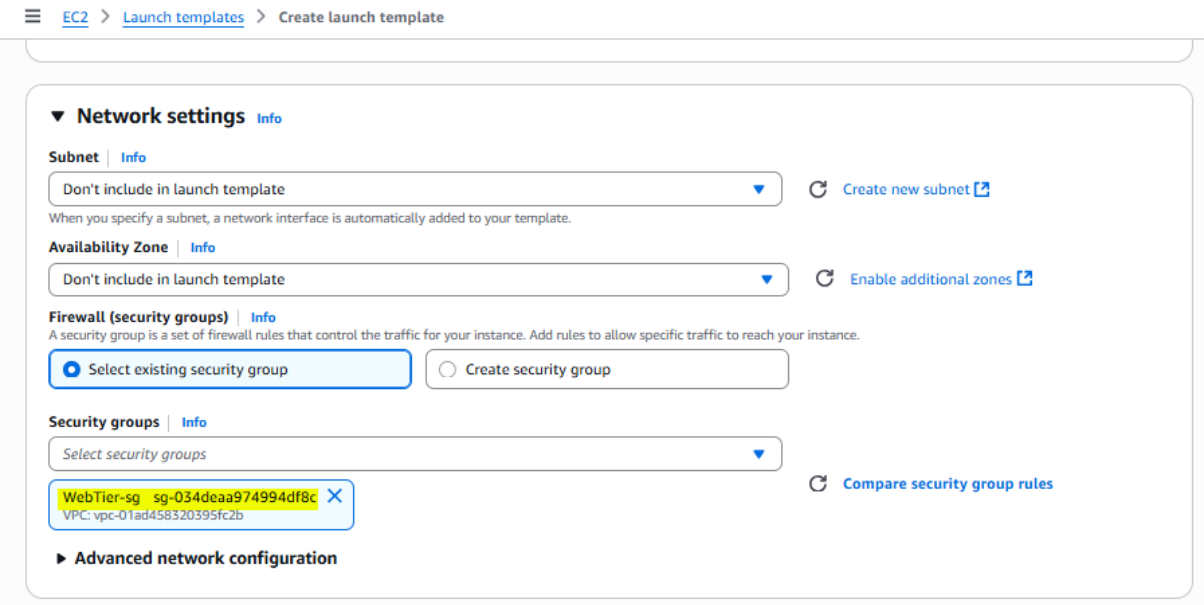


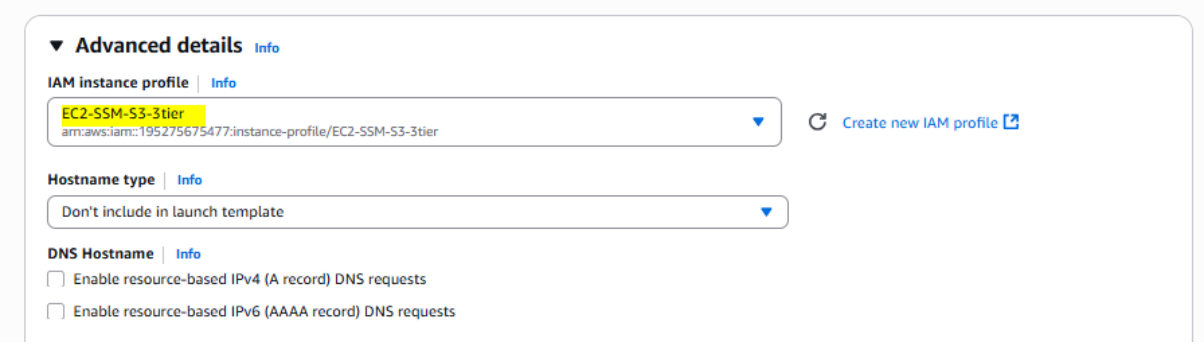


8. Create launch template→ Web-tier-launch-template









9. Create autoscalinggroup→WebTierASG

aws | Search [Alt+S] | Asia Pacific (Mumbai) | Account ID: 1952-7567-5477 Preeti B

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

Choose launch template Info

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.

WebTierASG

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.

Web-tier-launch-template

Create a launch template

Version

Default (1)

Create a launch template version

aws | Search [Alt+S] | Asia Pacific (Mumbai) | Account ID: 1952-7567-5477 Preeti B

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 6 - *optional* Add tags
Step 7 Review

Instance type
t2.micro

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-01ad458320395fc2b (3tier-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-az1 (ap-south-1a) | subnet-0cfb0ca06a0cc390f (Public-Web-subnet-AZ-1)
10.0.1.0/24

aps1-az3 (ap-south-1b) | subnet-0df85deb7532f808f (Public-Web-Subnet-AZ-2)
10.0.2.0/24

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.

aws | Search [Alt+S] | Asia Pacific (Mumbai) | Account ID: 1952-7567-5477 Preeti B

EC2 > Auto Scaling groups > Create Auto Scaling group

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

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.

- 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 that you want to attach to your Auto Scaling group.

- 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.

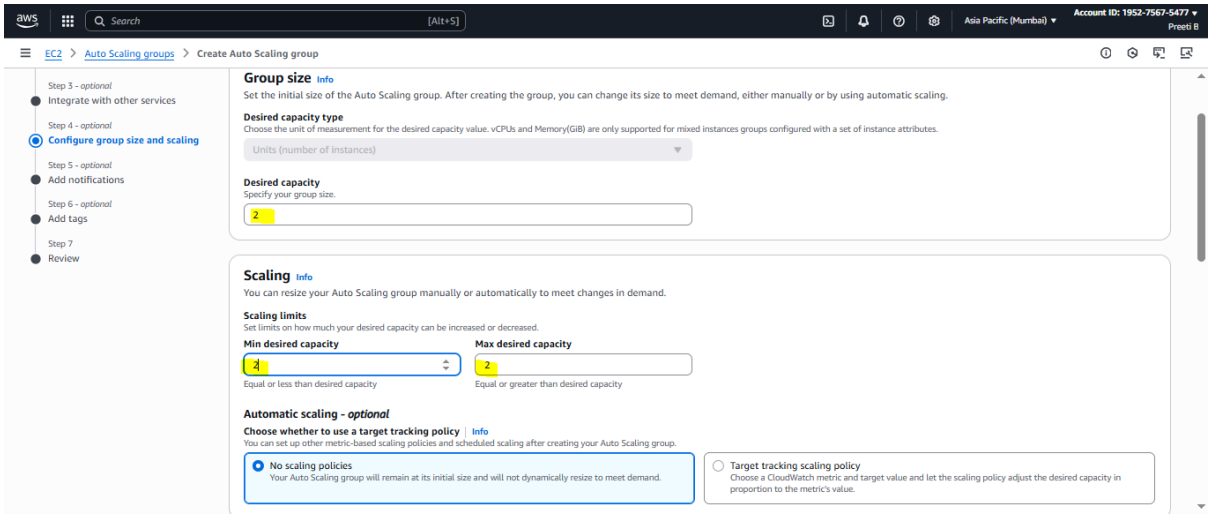
Select target groups

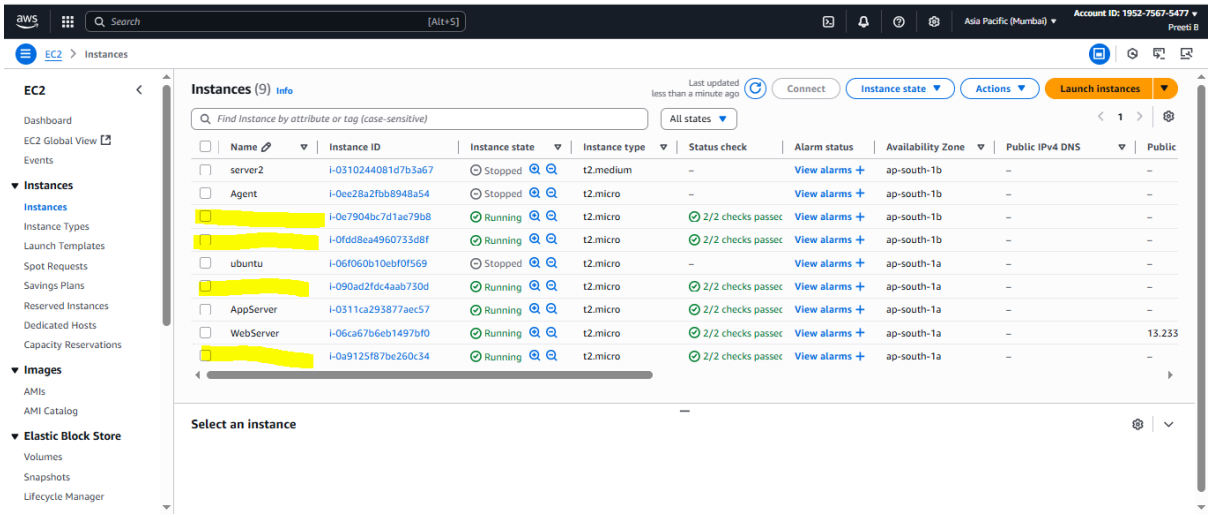
WebTier-tg | HTTP
Application Load Balancer: Web-tier-external-lb

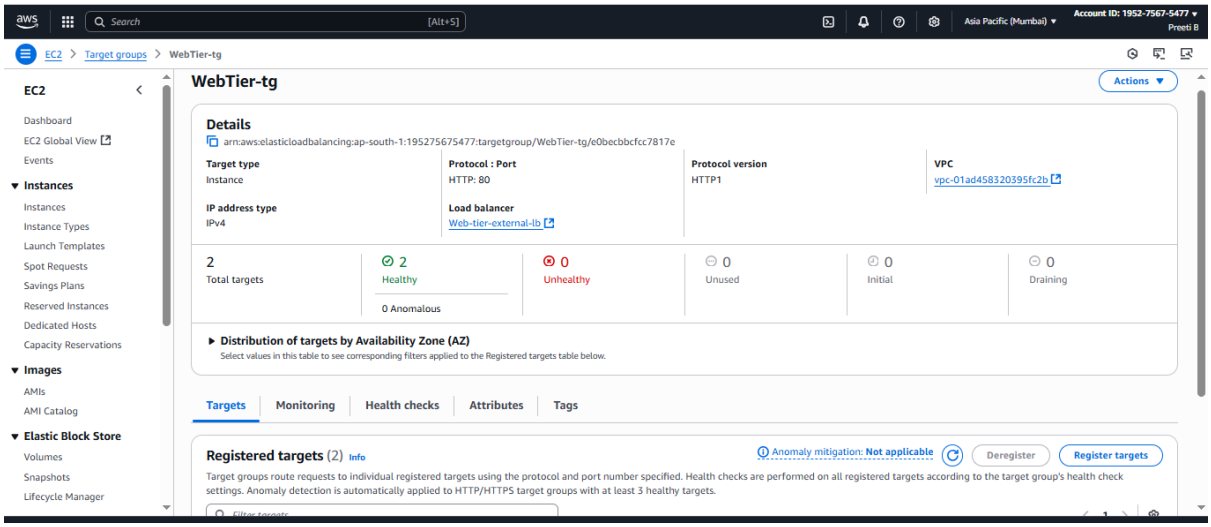
VPC Lattice integration options Info

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach

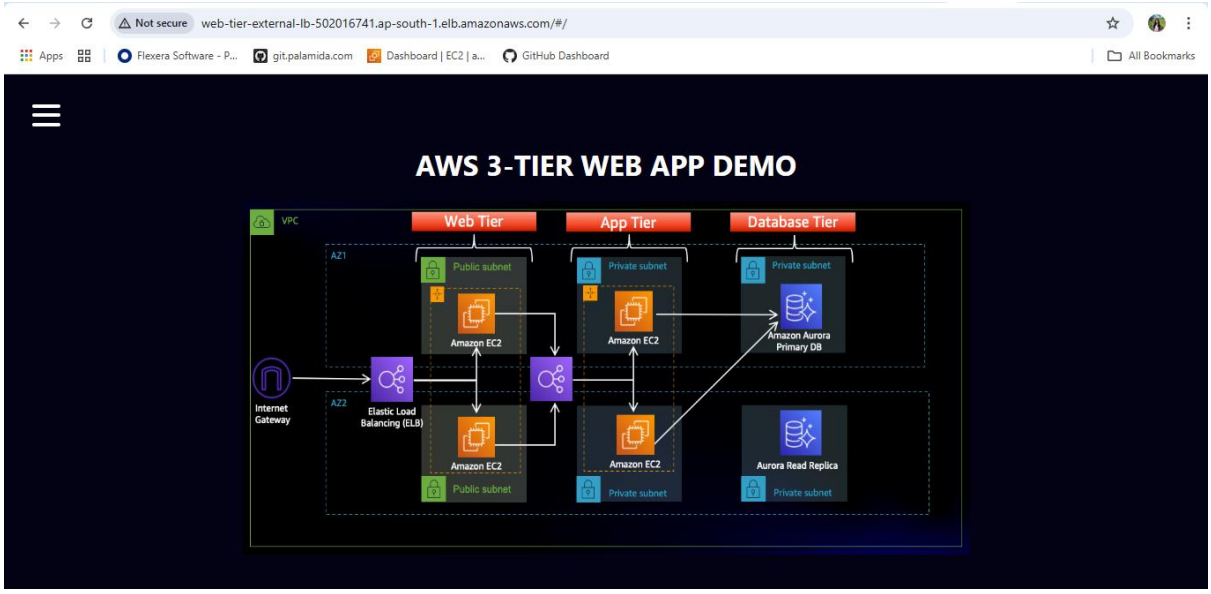


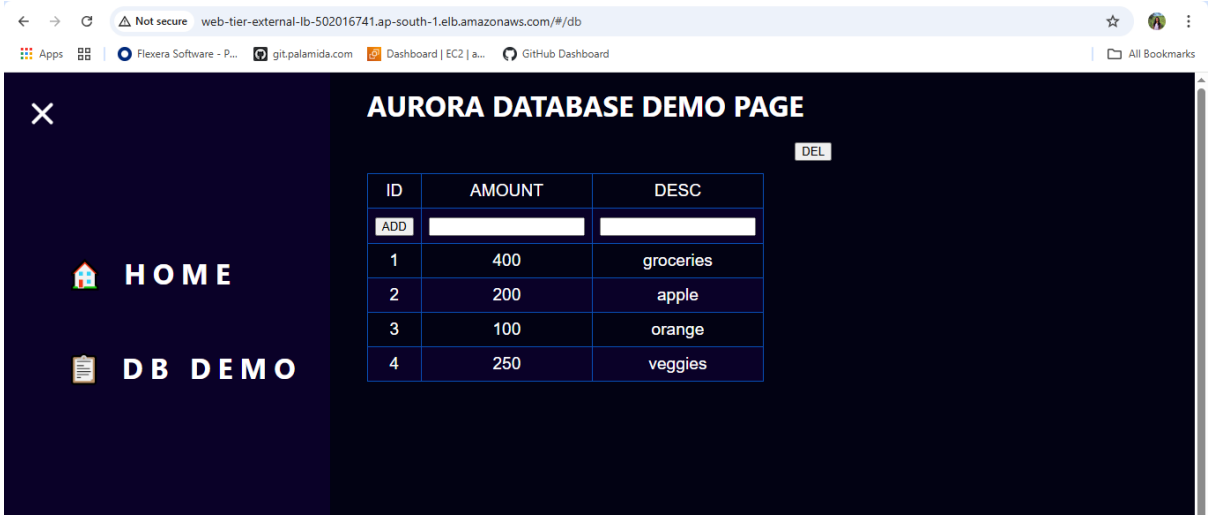




10. Hit the DNS of external loadbalencer→ **Web-tier-external-lb**

Web-tier-external-lb-502016741.ap-south-1.elb.amazonaws.com


Not secure
web-tier-external-lb-502016741.ap-south-1.elb.amazonaws.com/#/
Apps
Flexera Software - P...
git.palamida.com
Dashboard | EC2 | a...
GitHub Dashboard
All Bookmarks
AWS 3-TIER WEB APP DEMO
VPC
Web Tier
App Tier
Database Tier
AZ1
AZ2
Public subnet
Private subnet
Private subnet
Amazon EC2
Amazon EC2
Amazon Aurora
Primary DB
Internet
Gateway
Elastic Load
Balancing (ELB)
Amazon EC2
Amazon EC2
Aurora Read Replica
Public subnet
Private subnet
Private subnet


Not secure
web-tier-external-lb-502016741.ap-south-1.elb.amazonaws.com/#/db
Apps
Flexera Software - P...
git.palamida.com
Dashboard | EC2 | a...
GitHub Dashboard
All Bookmarks
AURORA DATABASE DEMO PAGE
DEL
ID
AMOUNT
DESC
ADD
1
400
groceries
2
200
apple
3
100
orange
4
250
veggies
HOME
DB DEMO