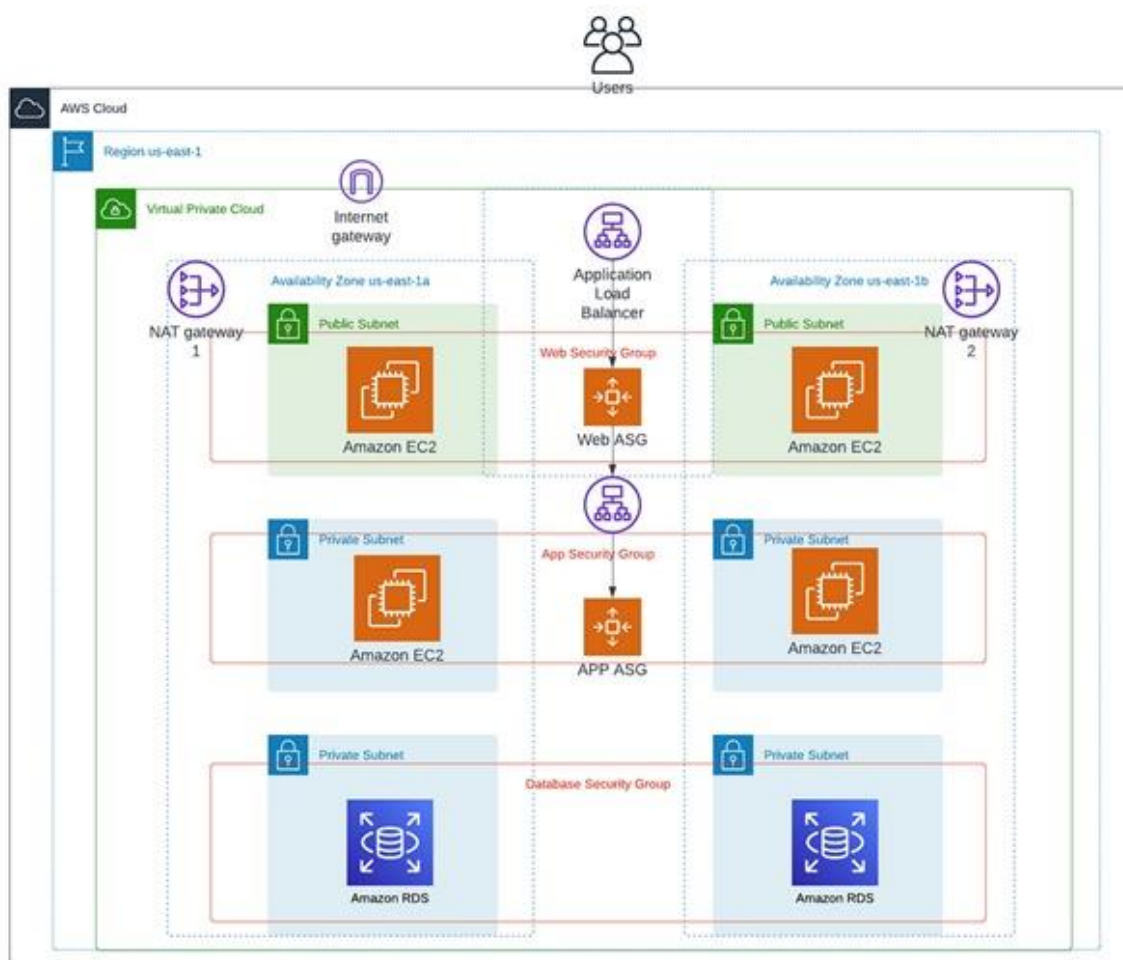


Project-1

NAME : Ch. Hari Siva Krishna

Given AWS Architecture :



The above architecture is the “3 Tier AWS Architecture”.

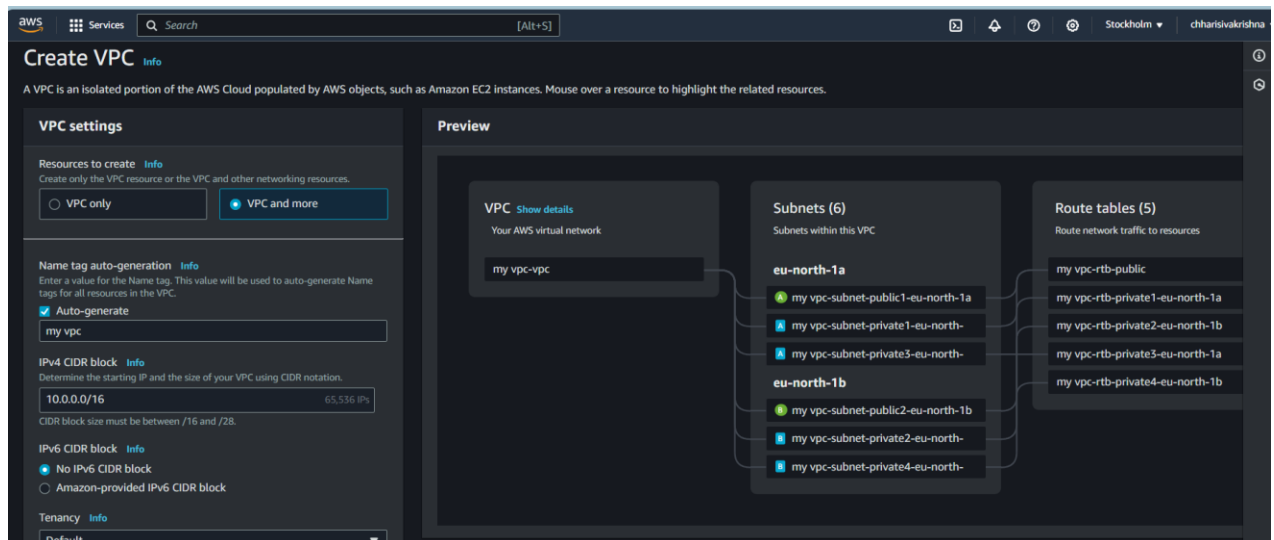
- ✚ EC2 instances are WEB Tier.
- ✚ Application Load Balancers and Auto Scaling Groups are APP Tier.
- ✚ RDS is Database Tier

The following are the steps to build the AWS Architecture :

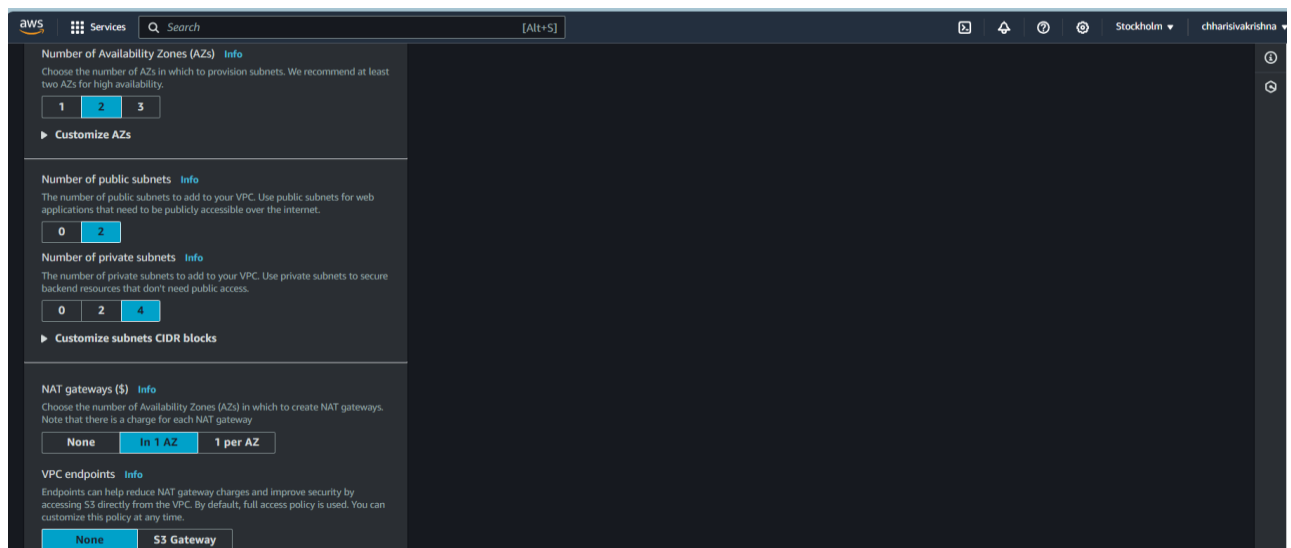
1. Create VPC, Subnets – 6, Internet gate way – 1, Route tables – 2, Nat gate way – 1.
2. Launch an EC2 instances.
3. Create Application load balancers and Autoscaling group.
- 4 Create Database (RDS).
5. Establish connection with RDS.

Step 1: Creating the vpc with required subnets , route tables ,internet gateways, nat gateways.

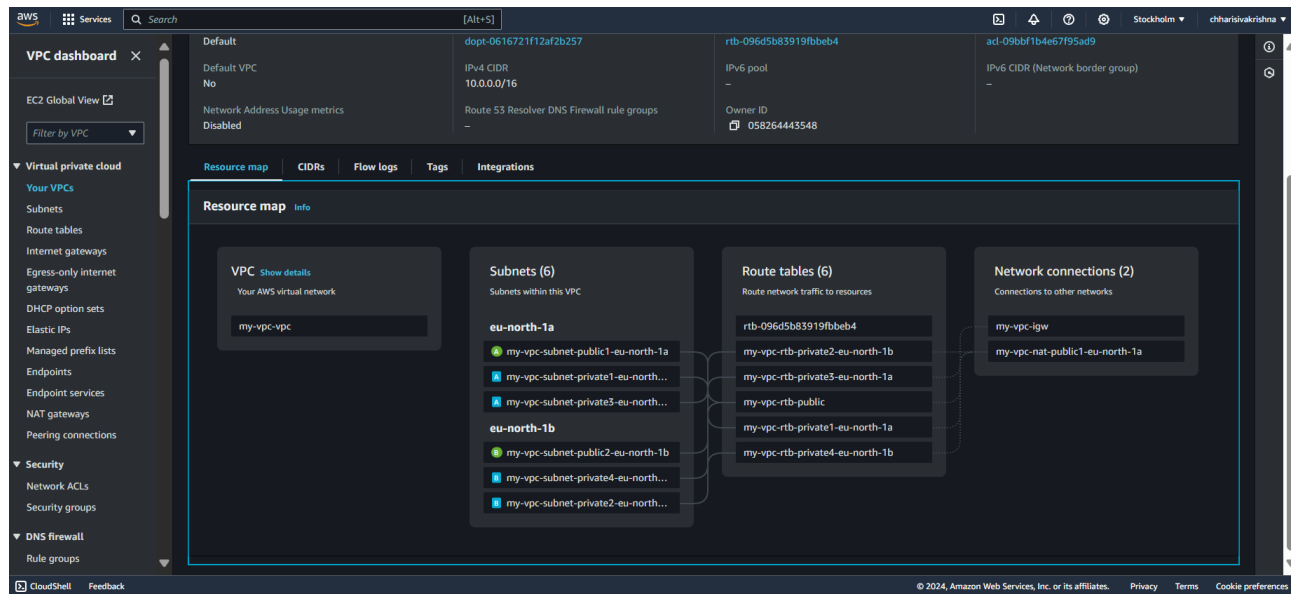
- For easy way and time saving process ,choose “vpc and more” to create vpc .
- We can create all the required 2 public and 4 private subnets , route tables ,internet gateways, nat gateways in 2 availability zones as shown in the architecture ,except it creates 4 route tables for 4 private subnets instead of 1 route table.



- Select “vpc and more” and “my vpc” as name .
- Give IPv4 CIDR block as 10.0.0.0/16 .



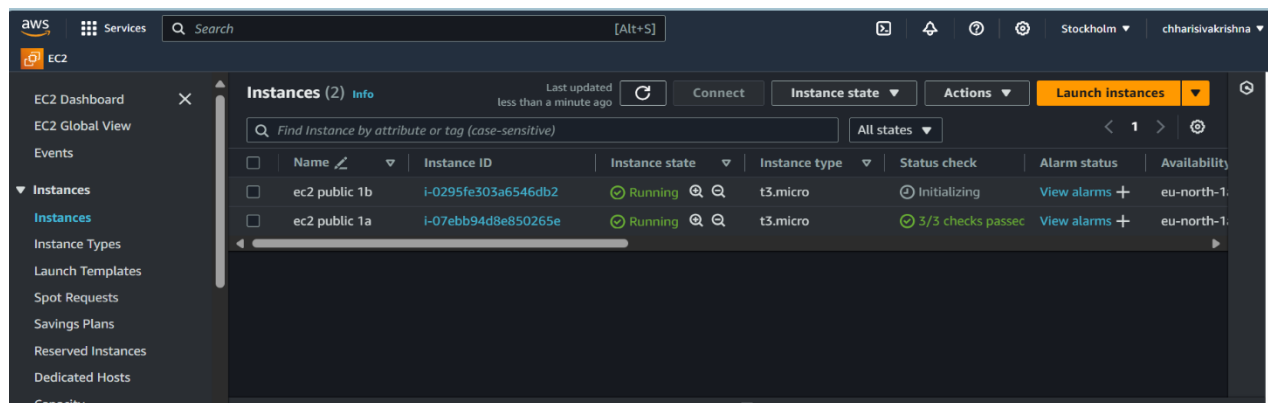
- Select 2 no. of Availability zones.
- Select 2 Public subnets and 4 Private subnates .
- Select 1 Nat gateway in one region ,which will be assigned to all private subnets.
- None of the Vpc endpoints are needed.
- All the required Internet gateways and Route tables for all the subnets are created which is the advantage of the “VPC and MORE”.



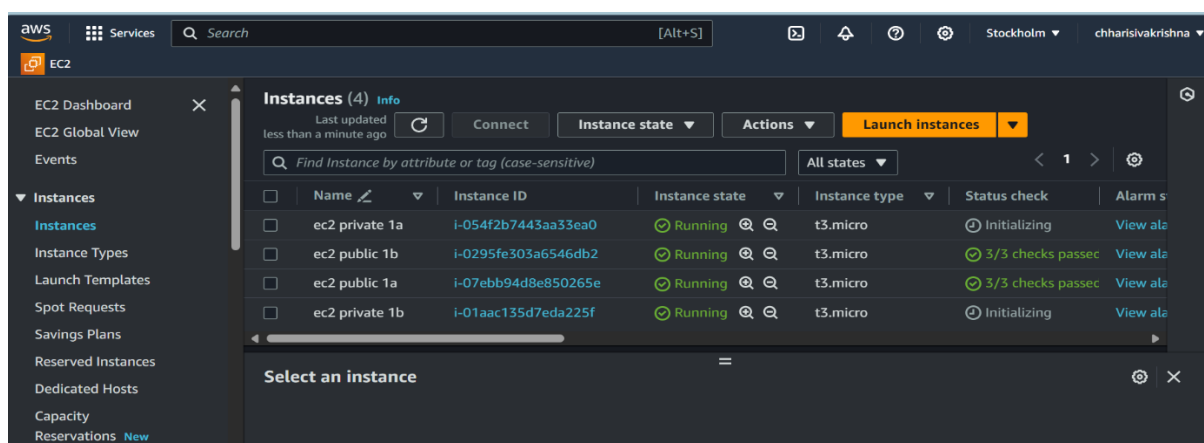
- This is the our VPC architecture

Step 2 : Launch EC2 instances – WEB Tier

- Launch two instances using two public subnets, each in two availability zones “1a and 1b”.
- Assign names for the two instances and select the Ubuntu server.
- Select the instance type as t3.micro and create new key pair.
- In the network settings ,select the vpc and subnet that you will use for instances
- And also in network settings ,create a security group with inbound rules of port 22 for SSH, port 80 for HTML, port 3306 for RDS which will come in upcoming tasks ,so that we can use the same security group for the instances .



- In the same way launch 2 instances using 2 private subnets, each in 2 availability zones “1a and 1b”.



Step 3: Create Application load balancers and Auto Scaling group – APP Tier

- For creating App LB and ASG ,the Target groups and Launch Templates are necessary.
- First create App LB and ASG for Public instances .

➤ So, first create Target groups.

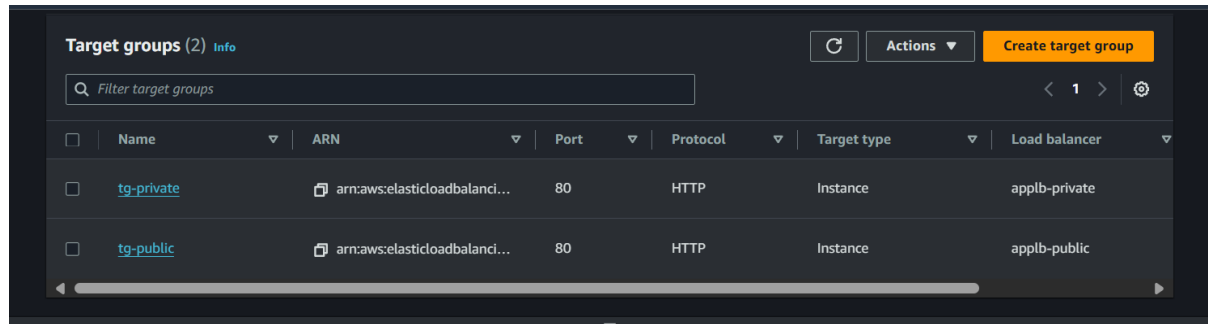
The screenshot shows the 'Specify group details' page in the AWS Management Console. The breadcrumb trail is 'EC2 > Target groups > Create target group'. The left sidebar shows 'Step 1 Specify group details' and 'Step 2 Register targets'. The main content area is titled 'Specify group details' with a subtitle 'Your load balancer routes requests to the targets in a target group and performs health checks on the targets.' Below this is the 'Basic configuration' section, which states 'Settings in this section can't be changed after the target group is created.' Under 'Choose a target type', there are four radio button options: 'Instances' (selected), 'IP addresses', 'Lambda function', and 'Application Load Balancer'. Each option has a list of bullet points describing its capabilities. At the bottom, there is a 'Target group name' field with the text 'tg public' entered. A note below the field states: 'A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.'

➤ Select target type as insatnce and give tg name as “tg-public”

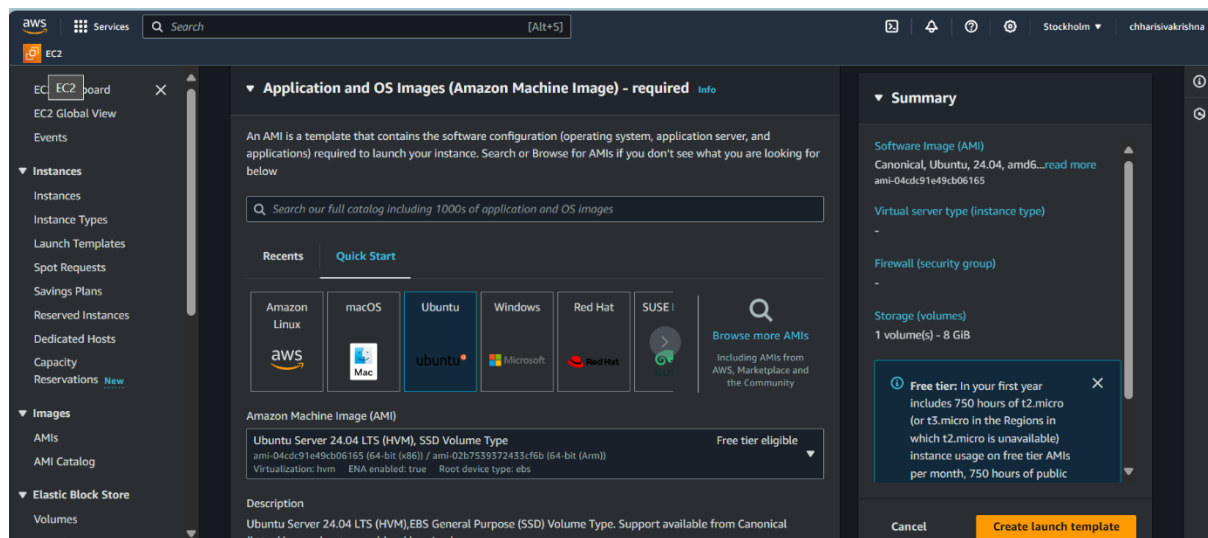
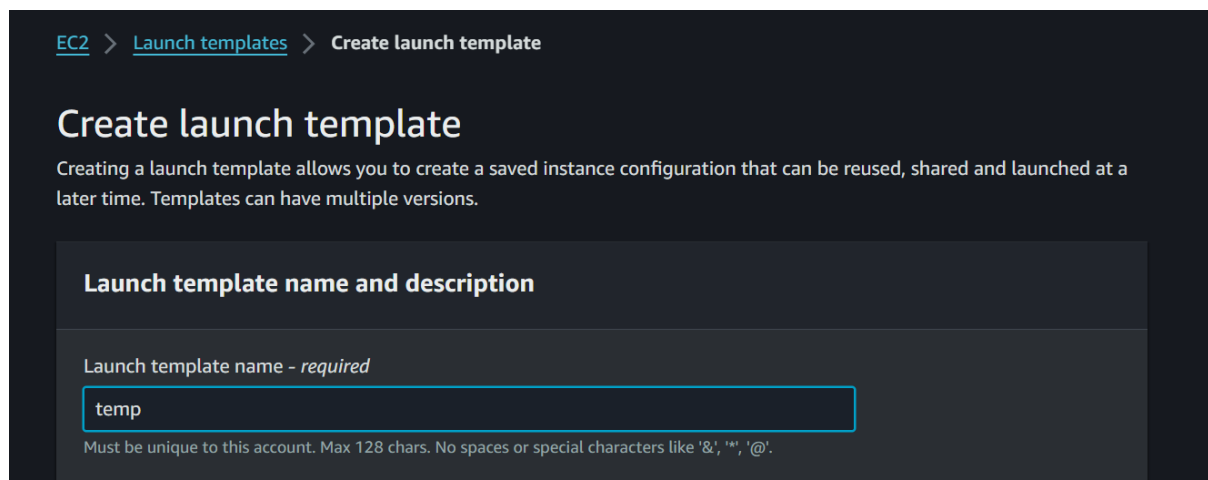
The screenshot shows the 'Register targets' page in the AWS Management Console. The breadcrumb trail is 'EC2 > Target groups > Create target group'. The left sidebar shows 'Step 1 Specify group details' and 'Step 2 Register targets'. The main content area is titled 'Register targets' with a subtitle 'This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.' Below this is the 'Available instances (2/2)' section. It includes a search bar 'Filter instances' and a table with two columns: 'Instance ID' and 'Name'. The table shows two instances: 'i-0295fe303a6546db2' (ec2 public 1b) and 'i-07ebb94d8e850265e' (ec2 public 1a), both with a 'Running' state. Below the table, it says '2 selected'. There is a section for 'Ports for the selected instances' with a text input field containing '80' and a note '1-65535 (separate multiple ports with commas)'. At the bottom right, there is a button labeled 'Include as pending below'.

➤ Select both the public instances and include as pending below.

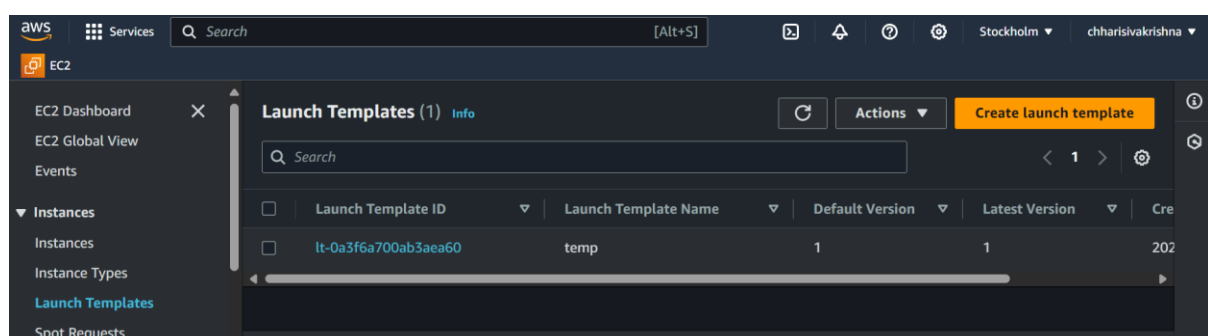
- In the same way, create a target group for private instances as “tg-private”.



- Now Launch template



- Choose the quick start and select the ubuntu server.
- Choose one of the AMI of your two instances.

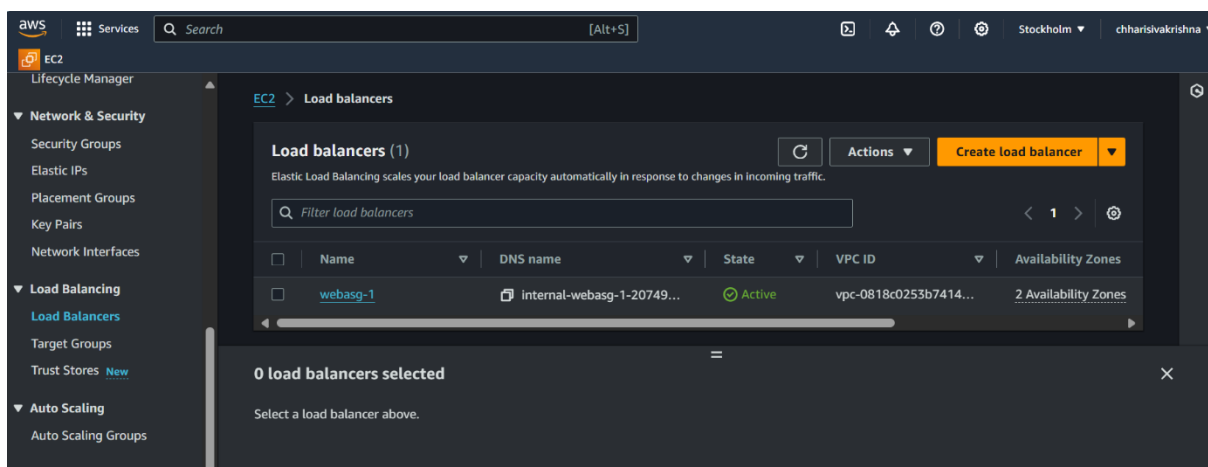
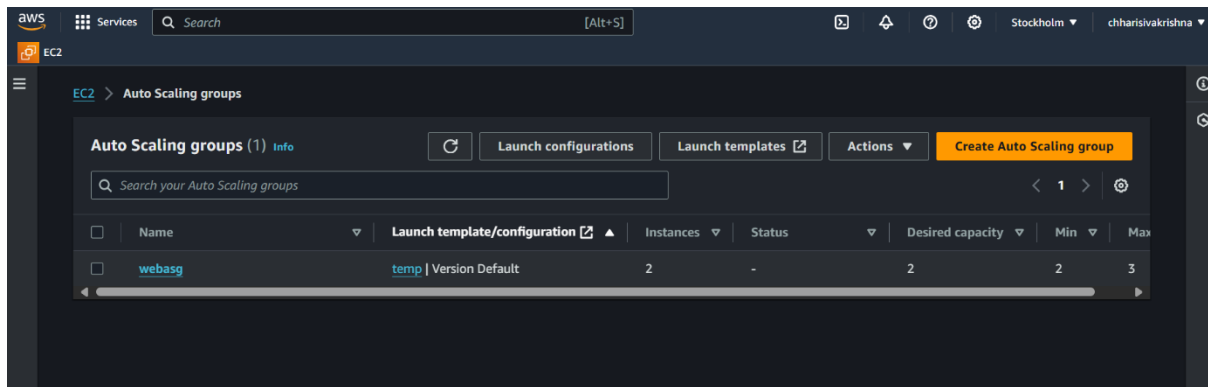
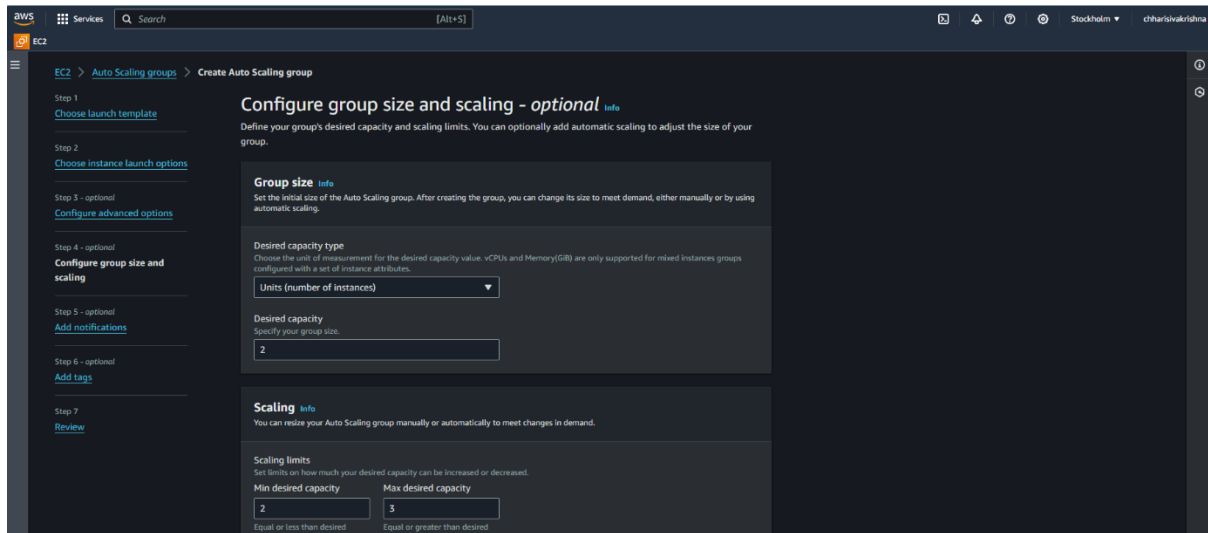


- Create Auto Scaling Group

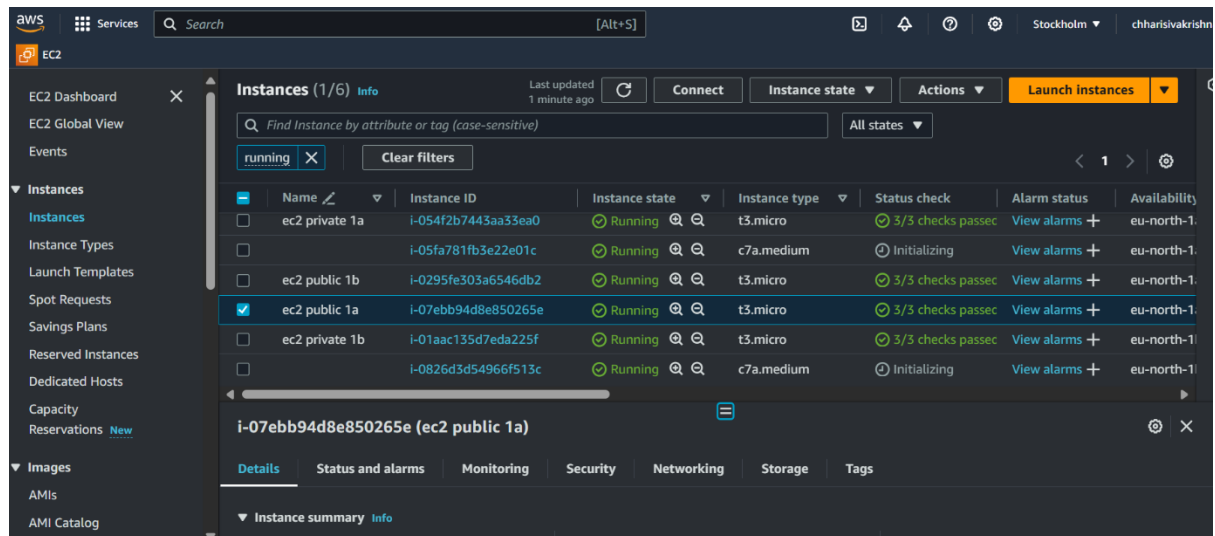
- Auto Scaling groups > Click Create Auto Scaling group button. Give it a name and select the launch template and click Next button.
- Select VPC we created and select subnets created for app tier. Click Next button. Choose Attach to a new load balancer.
- Then, select Application Load Balancer. For app tier, we need to select Internal Load balancer scheme.
- Select the target group. Keep the default settings for Health checks and click Next button. Review and then click Create Auto Scaling group button.

The screenshot shows the 'Choose launch template' step in the AWS Management Console. The left sidebar lists steps from 1 to 7, with Step 1 'Choose launch template' selected. The main content area has a title 'Choose launch template' and a subtitle 'Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.' There are two main sections: 'Name' and 'Launch template'. The 'Name' section has a text input for 'Auto Scaling group name' with the value 'webasg' entered. The 'Launch template' section has a dropdown menu with 'temp' selected. A blue information box states: '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.'

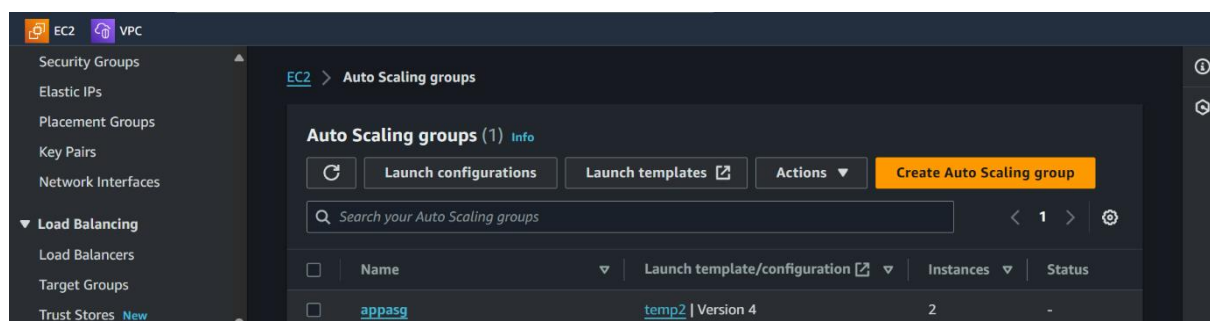
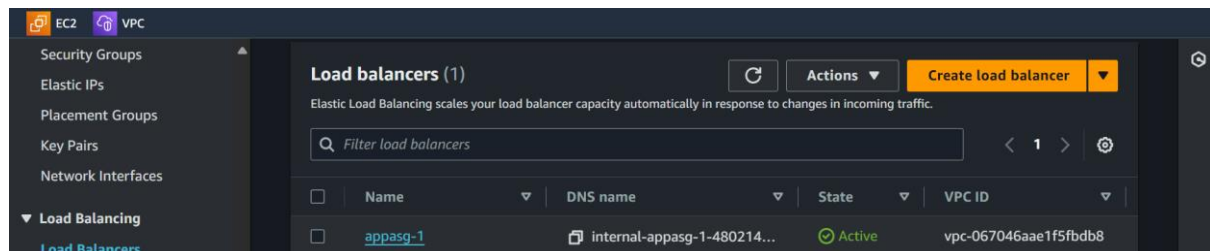
The screenshot shows the 'Configure advanced options - optional' step in the AWS Management Console. The left sidebar lists steps from 1 to 7, with Step 3 'Configure advanced options' selected. The main content area has a title 'Configure advanced options - optional' and a subtitle 'Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over health check replacements and monitoring.' There are two main sections: 'Load balancing' and 'Attach to a new load balancer'. The 'Load balancing' section has three radio button options: 'No load balancer', 'Attach to an existing load balancer', and 'Attach to a new load balancer'. The 'Attach to a new load balancer' option is selected. The 'Attach to a new load balancer' section has a subtitle 'Define a new load balancer to create for attachment to this Auto Scaling group.' and a 'Load balancer type' section with two radio button options: 'Application Load Balancer' and 'Network Load Balancer'. The 'Application Load Balancer' option is selected.



- The Auto Scaling Group and Application Load Balancers are created.



- The Auto Scaling Group instances are created.
- In the same way, create App LB and ASG for private ec2 instances as “appasg-1” and “appasg”.



- The Application load balancers and Auto Scaling group for private subnets are created.

Step 4: Create Database (RDS) – Database Tier

- Create database

The screenshot shows the 'Create database' page in the Amazon RDS console. At the top, there's a breadcrumb 'RDS > Create database'. The main heading is 'Create database'. Below it, a section titled 'Choose a database creation method' has two options: 'Standard create' (selected with a blue radio button) and 'Easy create'. The 'Standard create' option has a description: 'You set all of the configuration options, including ones for availability, security, backups, and maintenance.' The 'Easy create' option has a description: 'Use recommended best-practice configurations. Some configuration options can be changed after the database is created.' Below this is the 'Engine options' section. It has a sub-section 'Engine type' with four options: 'Aurora (MySQL Compatible)', 'Aurora (PostgreSQL Compatible)', 'MySQL' (selected with a blue radio button), and 'MariaDB'. Each option has a corresponding icon: a cylinder with a star for Aurora, a cylinder with a star for PostgreSQL, a yin-yang for MySQL, and a seal for MariaDB.

- Choose standard create and select mysql.

The screenshot shows the 'Engine Version' and 'Templates' sections of the Amazon RDS console. The 'Engine Version' section has a dropdown menu set to 'MySQL 8.0.35'. Below it, there's a checkbox for 'Enable RDS Extended Support' which is unchecked. The 'Templates' section has a heading 'Templates' and a sub-heading 'Choose a sample template to meet your use case.' Below this, there are three template options: 'Production' (unchecked), 'Dev/Test' (unchecked), and 'Free tier' (selected with a blue radio button). The 'Free tier' option has a description: 'Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.' and an 'Info' link.

- Select Free tier template.

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.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 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.

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](#)

Password strength Neutral

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

Confirm master password [Info](#)



➤ Give database name, Master username and password in self managed management.

Connectivity
Info

☐
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.

EC2 instance
Info

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-077ee371dd5bd163d

ec2 pri 1c

Some VPC settings can't be changed when a compute resource is added

Adding an EC2 compute resource automatically selects the VPC, DB subnet group, and public access settings for this database. To allow the EC2 instance to access the database, a VPC security group rds-ec2-X is added to the database and another called ec2-rds-X to the EC2 instance. You can remove the new security group for the database only by removing the compute resource.

Virtual private cloud (VPC)
Info

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

my-vpc-vpc (vpc-04e80902e8a066ccb)

6 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

- Connect to the private instance that is created for RDS 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

Additional VPC security group

Choose one or more options

sg_project

- Choose the security group.

rds-database Refresh Modify Actions

Summary

DB identifier rds-database	Status Available	Role Instance	Engine MySQL Community	Recommendations
CPU 3.52%	Class db.t3.micro	Current activity 0 Connections	Region & AZ eu-north-1a	

Connectivity & security
Monitoring
Logs & events
Configuration
Maintenance & backups
Tags
Recommendations

Connectivity & security

Endpoint & port Endpoint rds-database.cf44860sub2w.eu-north-1.rds.amazonaws.com Port 3306	Networking Availability Zone eu-north-1a VPC my-vpc-vpc (vpc-04e80902e8a066ccb)	Security VPC security groups sg_project (sg-02cd2b798b7b7929f) Active rds-ec2-3 (sg-0f1e2f9ef9dd0415c) Active Publicly accessible
--	--	--

- The RDS database is created.

Step 5: Establish connection with RDS.

- Now connect to the private server and install Mysql.
 - sudo apt update
 - sudo apt install mysql-server
 - sudo systemctl start mysql.service
- Connect to the RDS database by end point address

```

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> show databases,
-> \c
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> create database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to
on for the right syntax to use near '' at line 1
mysql> create database hsk;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| hsk |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

```

- Check the databases that are present and create your database.
- Create Table and Insert the data into the Table.

```

mysql> CREATE TABLE Persons (
-> ID int NOT NULL,
-> LastName varchar(255) NOT NULL,
-> FirstName varchar(255),
-> Age int,
-> PRIMARY KEY (ID)
-> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE frnd(
-> num int,
-> schl_frnd varchar(90),
-> region varchar(90)
-> );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO frnd (num,schl_frnd,region)
-> VALUES ('1','hari siva krishna','Tanuku');
Query OK, 1 row affected (0.01 sec)

mysql> select *from frnd;
+-----+-----+-----+
| num | schl_frnd | region |
+-----+-----+-----+
| 1 | hari siva krishna | Tanuku |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> |

```

- Repeat the process , to build rds for other private subnet in another availability zone as shown in the AWS Architecture.

Thus, the 3-TierAWS Architecture is successfully built.