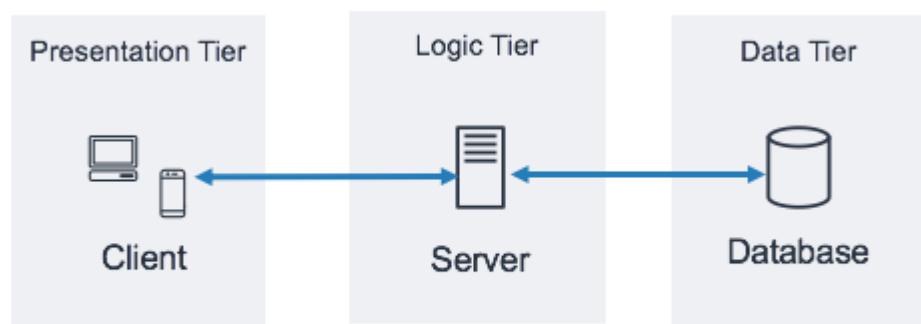


# 3- Tier Architecture

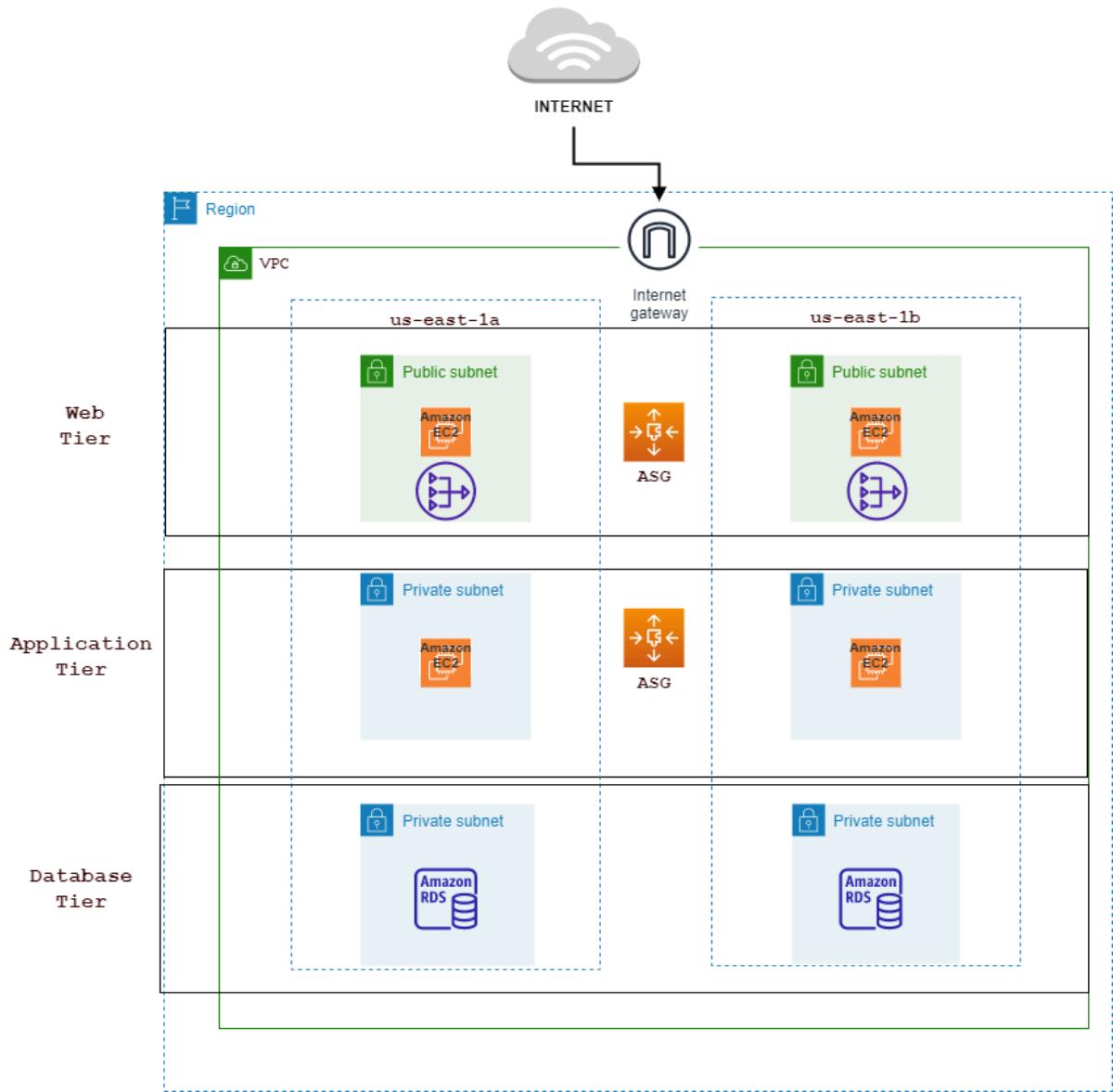
## 3- Tier Architecture:-

- The three-tier architecture is the most popular implementation of a multi-tier architecture and consists of a single presentation tier, logic tier, and data tier. The following illustration shows an example of a simple, generic three-tier application.



- **Why 3-tier?** This form of architecture addresses all the issues stated above. It provides increased scalability, availability, and security by spreading the application into multiple Availability Zones and separating it into three layers that serve different functions, independent of each other. If an AZ does down for some reason, the application has the ability to automatically scale resources to another AZ, without affecting the rest of the application tiers. Each tier has its own security group that only allows the inbound/outbound traffic needed to perform specific tasks.
1. **Web Tier:** Houses the user-facing elements of the application, such as web servers and the interface/frontend.
  2. **Application Tier:** Houses the backend and application source code needed to process data and run functions.
  3. **Data Tier:** Houses and manages the application data, often where the databases are stored.

## Project:-



## VPC :

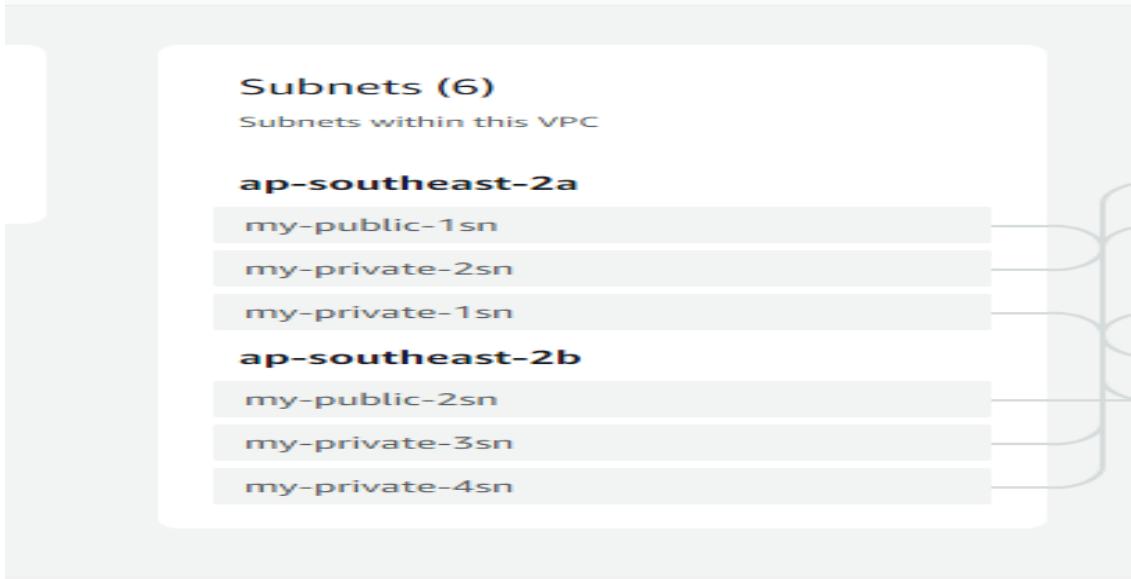
- Create a VPC with specific name (my-vpc-1) in a region sydney

The screenshot shows the AWS VPC dashboard with the following details for the VPC named "my-vpc-1":

Details	Info		
VPC ID vpc-05f74256c8628d507	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-09cc54e480a98c6e	Main route table rtb-01d3e08bdadb8abe0	Main network ACL acl-01fba91ee2f04e9f5
Default VPC No	IPv4 CIDR 192.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 992382705985	

Below the table, there are tabs for "Resource map" and "Info".

- Now, create 6 subnets (1 public, 2 private subnets in one availability zone and same for the other 3 subnets but in different availability zone) with in the same VPC



- Create a Internet Gateway and click on actions and attach it to the VPC

VPC > Internet gateways > igw-0527104185c5daabb

igw-0527104185c5daabb / my-igw1

Details		Info
Internet gateway ID <a href="#">igw-0527104185c5daabb</a>	State <a href="#">Attached</a>	VPC ID <a href="#">vpc-017be77419d60c65b   my-vpc-1</a>
Owner <a href="#">992382705985</a>		

**Tags**

Search tags	
Key	Value
Name	my-igw1

- And create a route tables and associate them with the subnets along with the internet connection attached to the (1a & 2a) public subnets

VPC > Route tables > [rtb-09f29c677fd17d077](#) > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

**Available subnets (2/6)**

Available subnets (2/6)					
<a href="#">Filter subnet associations</a>					
<input checked="" type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	my-public-1sn	<a href="#">subnet-0b99ffa9fd90213a</a>	192.0.0.0/24	-	<a href="#">rtb-09f29c677fd17d077 / my-rt-1</a>
<input type="checkbox"/>	my-private-1sn	<a href="#">subnet-03e6faf68e68e05ba</a>	192.0.1.0/24	-	<a href="#">rtb-0797365d55bcb2d77 / my-rt-2</a>
<input type="checkbox"/>	my-private-2sn	<a href="#">subnet-0cc0fb52977212258</a>	192.0.4.0/24	-	<a href="#">rtb-0e1faa923e4e7c2ea / my-rt-3</a>
<input checked="" type="checkbox"/>	my-public-2sn	<a href="#">subnet-0ae392f23c40346be</a>	192.0.5.0/24	-	<a href="#">rtb-09f29c677fd17d077 / my-rt-4</a>
<input type="checkbox"/>	my-private-3sn	<a href="#">subnet-060254ce91332ec81</a>	192.0.6.0/24	-	<a href="#">rtb-06abd75cfffbfb0c02 / my-rt-5</a>
<input type="checkbox"/>	my-private-4sn	<a href="#">subnet-0873fef48cda59f8a</a>	192.0.7.0/24	-	<a href="#">rtb-06a4d1f86f4e311cf / my-rt-6</a>

- And also Set the public route tables as a main route tables

Route tables (1/7) [Info](#)

Find resources by attribute or tag

Name	Route table ID	Explicit subnet assoc...
-	<a href="#">rtb-0761c185b1aea872</a>	-
-	<a href="#">rtb-0b121f0cef5caa20b</a>	-
<input checked="" type="checkbox"/> my-rt-public	<a href="#">rtb-09f29c677fd17d077</a>	2 subnets
<input type="checkbox"/> my-rt-private-1	<a href="#">rtb-0797365d55bcb2d77</a>	<a href="#">subnet-03e6faf68e68e05...</a>
<input type="checkbox"/> mv-rt-nrvate-2	<a href="#">rtb-0e1faa92c34e7c2ea</a>	<a href="#">subnet-0rr0fh52977212...</a>

[Actions ▲](#) [Create route table](#)

View details [Main](#)

Set main route table [Yes](#)

Edit subnet associations [Yes](#)

Edit edge associations [Yes](#)

**Edit route propagation** [Yes](#)

Edit routes [No](#)

Manage tags [None](#)

Delete route table [Delete](#)

- Goto the subnet and click on edit subnets to auto assign the IPv4

VPC > Subnets > [subnet-0b99ffa9fda90213a](#) > Edit subnet settings

### Edit subnet settings [Info](#)

**Subnet**

Subnet ID	Name
<a href="#">subnet-0b99ffa9fda90213a</a>	<a href="#">my-public-1sn</a>

**Auto-assign IP settings** [Info](#)  
Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

Enable auto-assign public IPv4 address [Info](#)

Enable auto-assign customer-owned IPv4 address [Info](#)  
Option disabled because no customer owned pools found.

**Resource-based name (RBN) settings** [Info](#)  
Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

- Create a nat gateway for one way traffic attach to the all private subnets

VPC > NAT gateways > [nat-0e420c3b1725e22bd](#)

### nat-0e420c3b1725e22bd / my-nat-1

[Actions ▾](#)

**Details**

NAT gateway ID <a href="#">nat-0e420c3b1725e22bd</a>	Connectivity type Public	State <a href="#">Pending</a>	State message <a href="#">Info</a> -
NAT gateway ARN <a href="#">arn:aws:ec2:ap-southeast-2:992382705985:natgateway/nat-0e420c3b1725e22bd</a>	Primary public IPv4 address -	Primary private IPv4 address <a href="#">172.31.32.82</a>	Primary network interface ID <a href="#">eni-0f0af33c2e69b64af</a>
VPC <a href="#">vpc-0359ed21b991d4948</a>	Subnet <a href="#">subnet-013f4413a7d9f0251</a>	Created <a href="#">Monday, March 11, 2024 at 09:42:42 GMT+5:30</a>	Deleted -

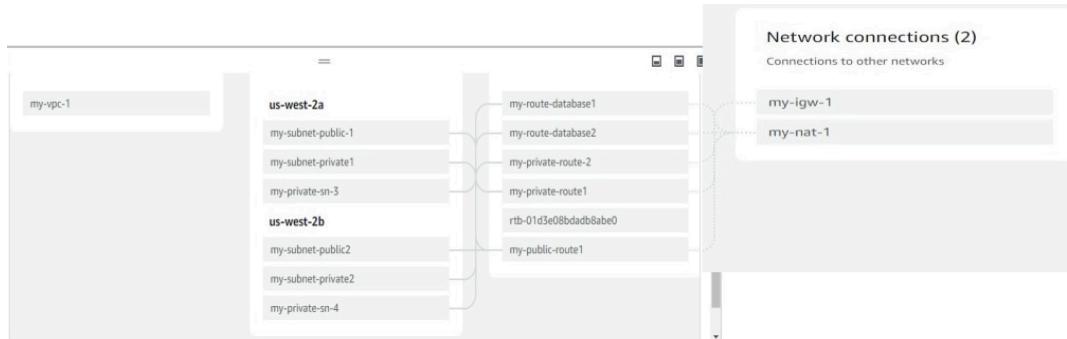
VPC > Route tables > [rtb-0e1faa92c34e7c2ea](#) > Edit routes

### Edit routes

Destination	Target	Status	Propagated
192.0.0.0/16	local <input type="button" value="X"/>	<input checked="" type="radio"/> Active	No
<input type="text" value="0.0.0.0"/> X	NAT Gateway <input type="button" value="X"/>	<input checked="" type="radio"/> Active	No
<a href="#">Add route</a>			

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

- After all connections established in a VPC



## Web Tier:

- Create a web tier launch template
  - Give a Specific Name my-template1
  - Click on AMI: Amazon 2 Linux
  - Instance type: t2.micro (1GB – Free Tier)
  - A new or existing key pair
  - Create a new security group and add inbound rules for SSH & HTTP

The screenshot shows the AWS EC2 Security Groups details page for 'sg-088b58fcf644c6c94 - my-sg-1'. It displays the following information:

- Details:**
  - Security group name: my-sg-1
  - Security group ID: sg-088b58fcf644c6c94
  - Description: allow ssh
  - VPC ID: vpc-05f74256c8628d507
  - Owner: 992382705985
  - Inbound rules count: 3 Permission entries
  - Outbound rules count: 2 Permission entries
- Inbound rules:** Outbound rules (2)
- Outbound rules (2):** Manage tags, Edit outbound rules

The screenshot shows the 'Edit inbound rules' page for the same security group. It lists two rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-01529332653c61903	SSH	TCP	22	Cus... 0.0.0.0/0	
sgr-0a2d7feb23dc64fd9	HTTP	TCP	80	Cus... 0.0.0.0/0	

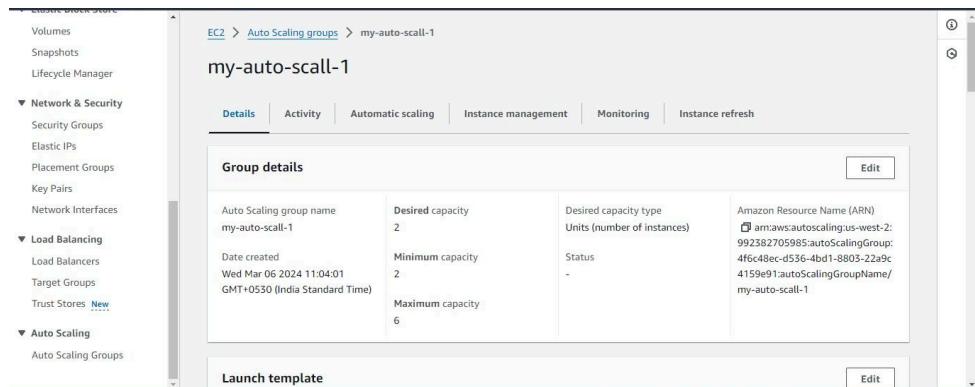
Buttons include 'Add rule' and 'Delete' for each row.

- Now let's Launch Template

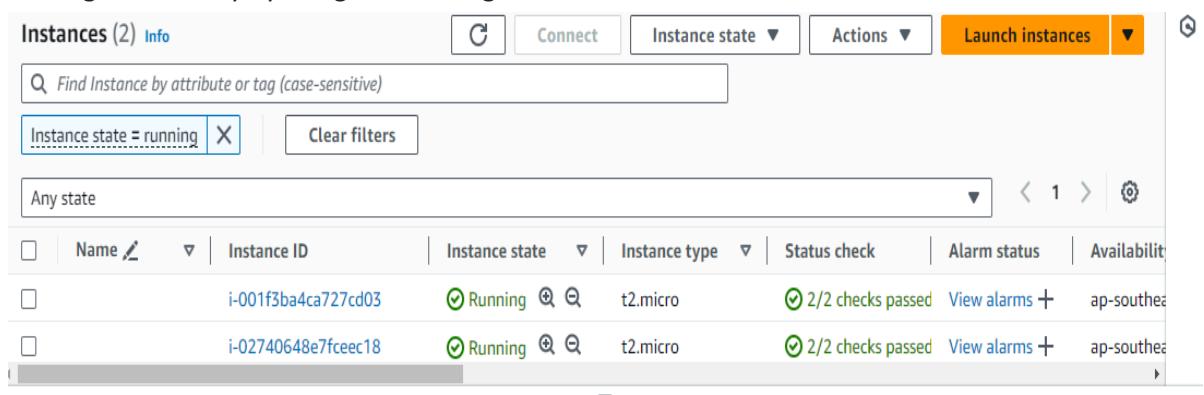
The screenshot shows the AWS EC2 Launch Templates details page for 'my-template1 (lt-087157bc8a38aec82)'. It displays the following information:

- Launch template details:**
  - Launch template ID: lt-087157bc8a38aec82
  - Launch template name: my-template1
  - Default version: 1
  - Owner: arn:aws:iam::992382705985:root
- Launch template version details:**
  - Version: 1 (Default)
  - Description: -
  - Date created: 2024-03-07T04:52:59.000Z
  - Created by: arn:aws:iam::992382705985:root

- Create auto -scaling with following settings
  - 1) Click on the auto-scaling group and give a name.
  - 2) Attach the created template.
  - 3) Select the my-vpc-1 along with two public subnets.
  - 4) Create a new application load balancer with target groups.
  - 5) Configure the group size and scaling for desired capacity servers needed



- Now we can goto the EC2(Elastic Compute Cloud) to check weather the 2 instances are running successfully by using auto scaling



- Let's try to connect to one of the instances in web



## Application Tier:-

- Create a web tier launch template
  - 1) Give a Specific Name my-template2
  - 2) AMI: Amazon 2 Linux
  - 3) Instance type: t2.micro (1GB – Free Tier)
  - 4) A new or existing key pair
  - 5) Create a new security group and add inbound rules for SSH & HTTP

## 6) Now let's Launch Template

- Create auto -scaling with following settings
  - 1) Click on the auto-scaling group and give a name.
  - 2) Attach the created template.
  - 3) Select the my-vpc-1 along with two private subnets.
  - 4) Create a new application load balancer with target groups.
  - 5) Configure the group size and scaling for desired capacity servers needed

- Now , we can see that two more instances using auto scaling are created within two private subnets that I have assigned a specific name to identify the private subnets as pri 1 & pri 2.

- Lets connect to the private Ec2 instance from the public by using the key.pem

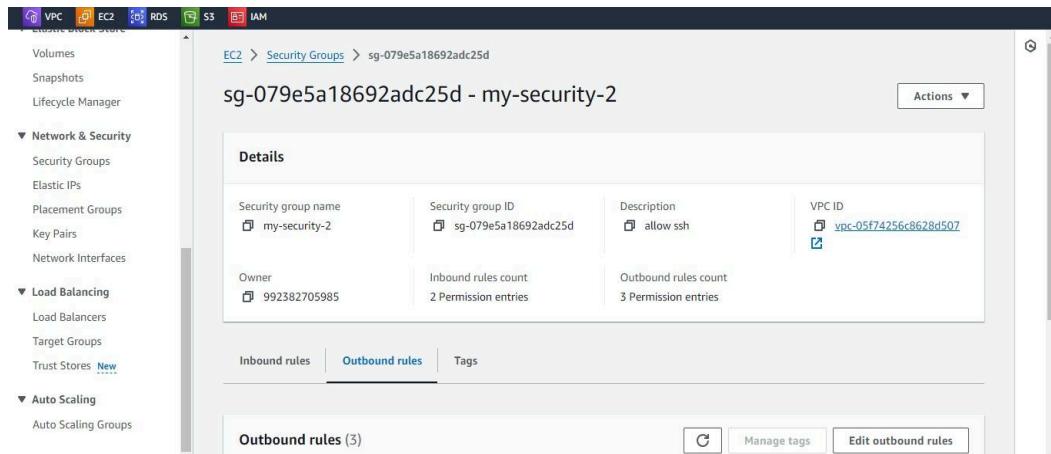
```

[ec2-user@ip-192-0-1-213 ~]$ sudo -i
[root@ip-192-0-1-213 ~]# vi key.pem
[root@ip-192-0-1-213 ~]# ls
key.pem
[root@ip-192-0-1-213 ~]# chmod -400 "key.pem"
[root@ip-192-0-1-213 ~]# ssh -i "key.pem" ec2-user@192.0.2.63
The authenticity of host '192.0.2.63' (192.0.2.63) can't be established.
ED25519 key fingerprint is SHA256:F95mx9ijDVv7IodBHauZfailTixltagMupXdoQu2ozOU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.0.2.63' (ED25519) to the list of known hosts.
$ WARNING: UNPROTECTED PRIVATE KEY FILE!
$ Permissions 0244 for `key.pem` are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "key.pem": bad permissions
ec2-user@192.0.2.63: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[root@ip-192-0-1-213 ~]# cd

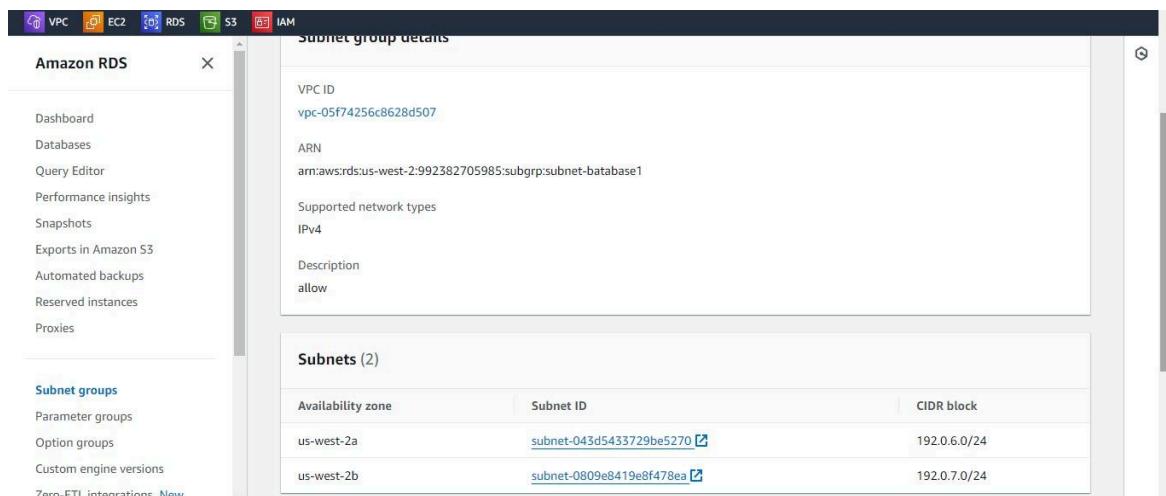
```

## Data Tier:-

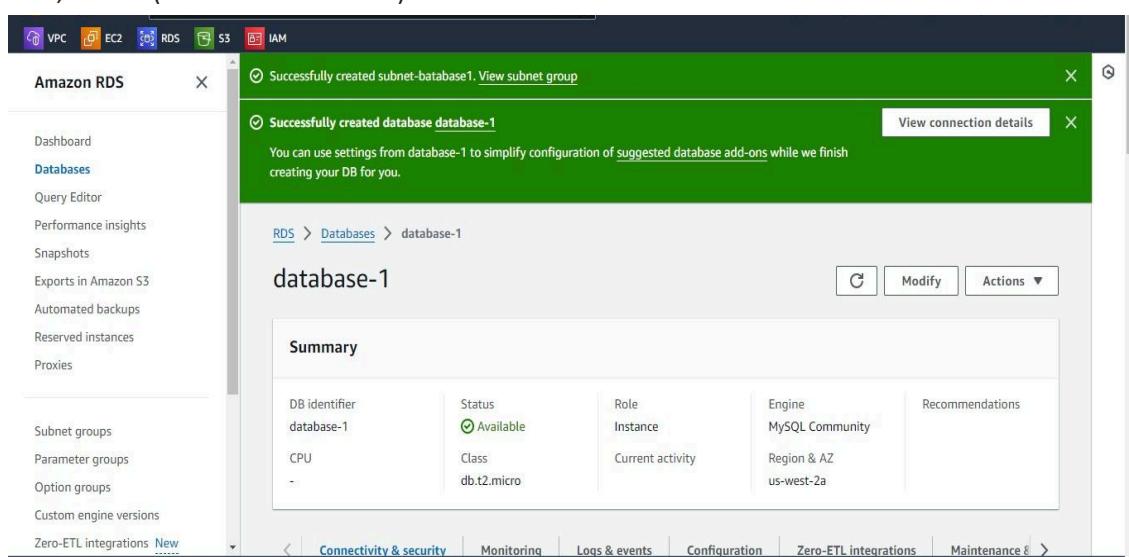
- Create a security group add inbound and outbound rules for mysql/Arora



- Create database subnet groups by attaching the 2 private subnets which are in different availability zones in RDS and attach the my-vpc-1.

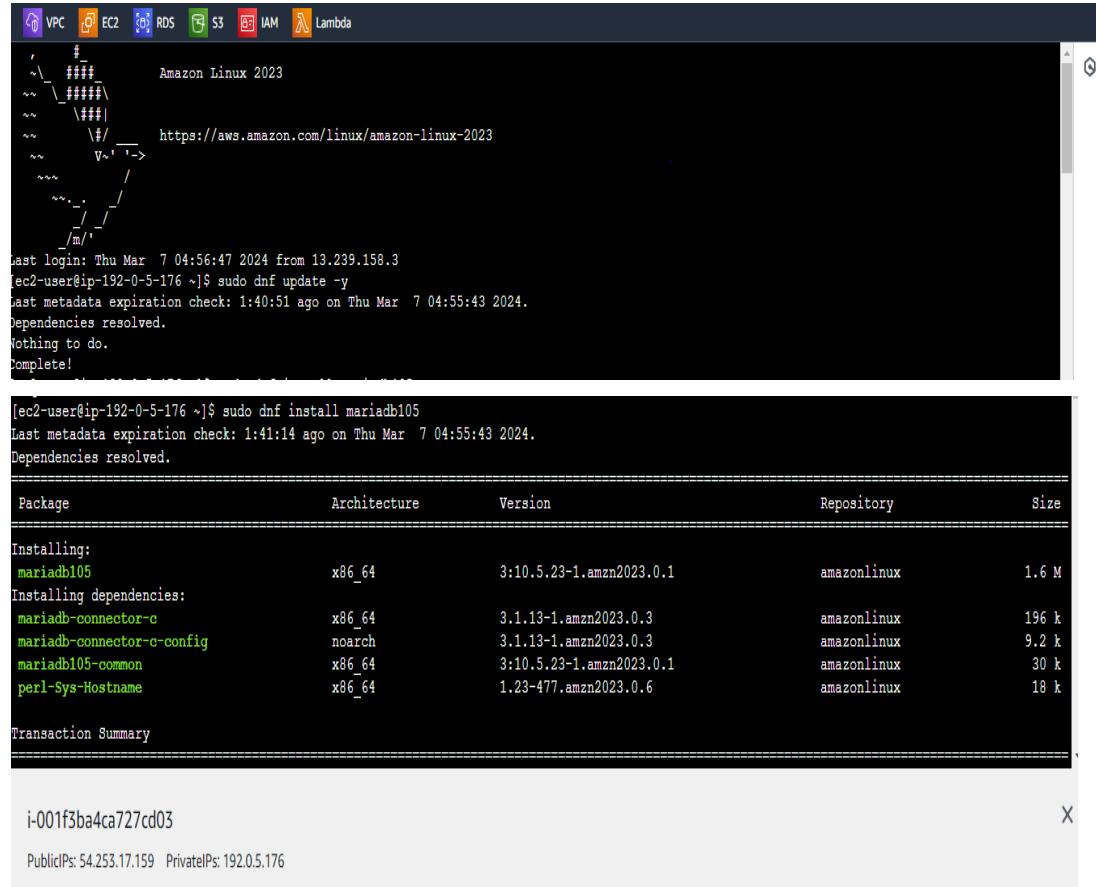


- Now, create (Relational Database) DataBase



- Connect to the database

1. Connect to the server
2. Install mysql



```

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Thu Mar  7 04:56:47 2024 from 13.239.158.3
[ec2-user@ip-192-0-5-176 ~]$ sudo dnf update -y
Last metadata expiration check: 1:40:51 ago on Thu Mar  7 04:55:43 2024.
Dependencies resolved.
Nothing to do.
Complete!

[ec2-user@ip-192-0-5-176 ~]$ sudo dnf install mariadb105
Last metadata expiration check: 1:41:14 ago on Thu Mar  7 04:55:43 2024.
Dependencies resolved.

=====
          Package           Architecture   Version      Repository  Size
=====
Installing:
mariadb105                x86_64        3:10.5.23-1.amzn2023.0.1    amazonlinux  1.6 M
Installing dependencies:
mariadb-connector-c          x86_64        3.1.13-1.amzn2023.0.3    amazonlinux  196 k
mariadb-connector-c-config    noarch       3.1.13-1.amzn2023.0.3    amazonlinux  9.2 k
mariadb105-common            x86_64        3:10.5.23-1.amzn2023.0.1    amazonlinux  30 k
perl-Sys-Hostname             x86_64        1.23-477.amzn2023.0.6    amazonlinux  18 k

Transaction Summary
=====

i-001f3ba4ca727cd03
X
PublicIPs: 54.253.17.159 PrivateIPs: 192.0.5.176

```

3. And enter command `mysql -h YOUR_DB_ENDPOINT -P 3306 -u YOUR_DB_USERNAME -p` in these command instead of `your_db_endpoint` we should enter our database endpoint and also edit username as well after that enter password that you have given while creating a relational database.

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MySQL connection id is 115

Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement

MySQL [(none)]> |