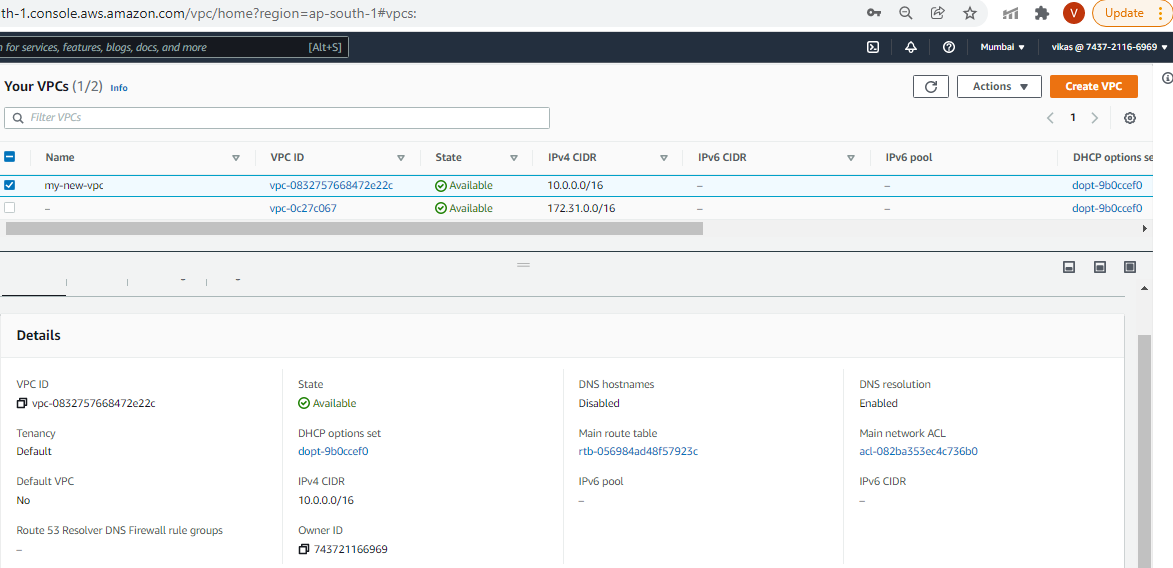
**3-Tier Architecture in AWS**

A three-tier architecture has the frontend, the backend and the database.

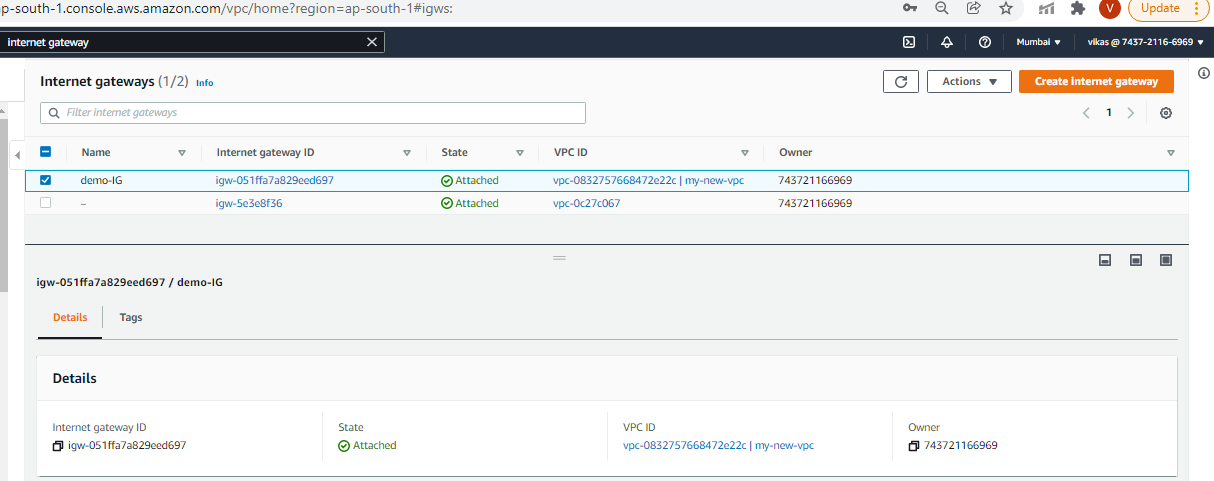
In order to design a 3-tier architecture in AWS, I am making use of the following AWS services:

* [Virtual Private Cloud](https://aws.amazon.com/vpc/?hp=tile&so-exp=below)(VPC)
* [Internet Gateway](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html)
* [Elastic Load Balancer (ELB)](https://aws.amazon.com/elasticloadbalancing/?hp=tile&so-exp=below),
* [Elastic Compute Cloud (EC2)](https://aws.amazon.com/ec2/?hp=tile&so-exp=below)
* [Auto Scaling Group](https://aws.amazon.com/ec2/autoscaling/?hp=tile&so-exp=below)
* [Security Groups](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html)

1. [Virtual Private Cloud](https://aws.amazon.com/vpc/?hp=tile&so-exp=below)(VPC) : Created a VPC to manage AWS resources in a secured manner.

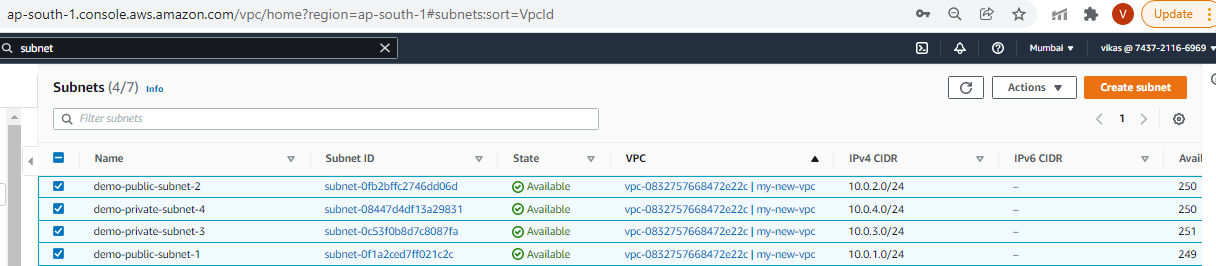


1. Internet Gateway: Created Internet Gateway to allow communication between the EC2 instances in the VPC and the internet.

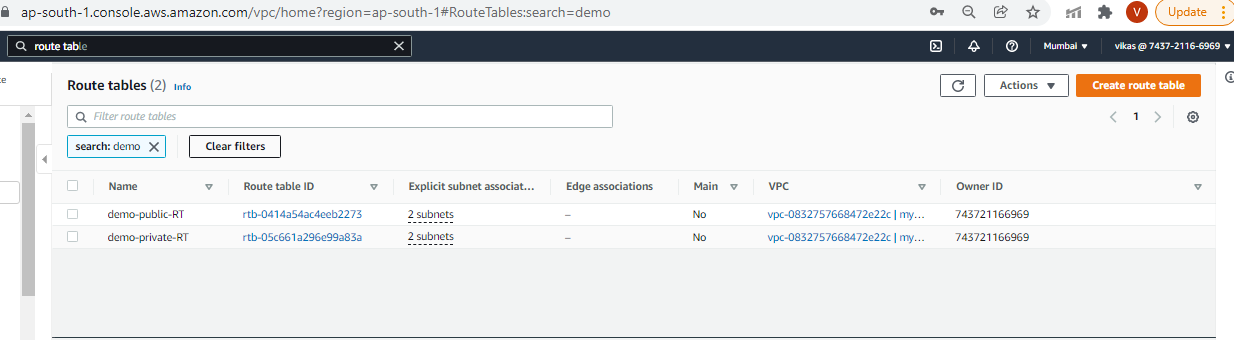


1. Subnets: Created 4 subnets (2 public and 2 private subnets)

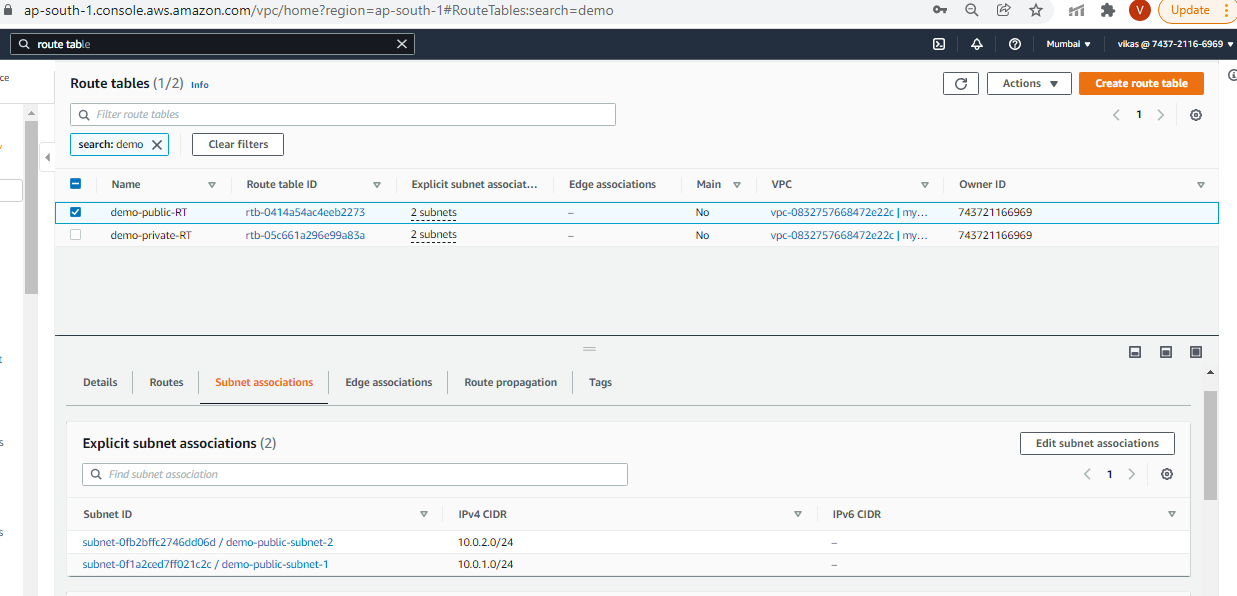
EC2 instances within a public subnet have public IPs and can directly access the internet while those in the private subnet does not have public IPs and can only access the internet through a [NAT](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html) gateway.

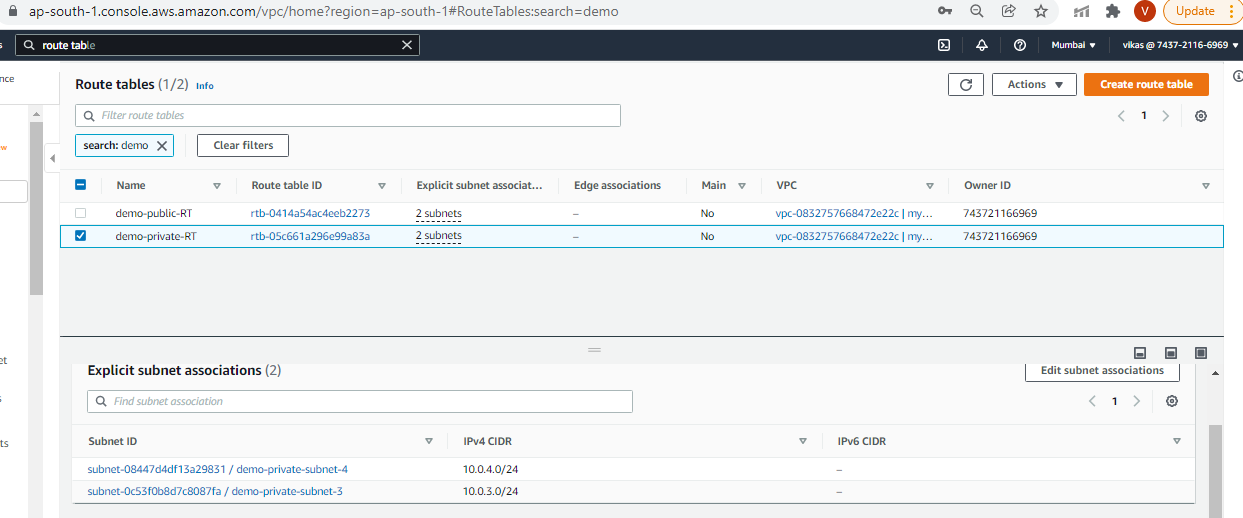


1. Route tables: Created 2 route tables, public route table to define which subnets will have direct access to the internet (public subnets) and private route table to define which subnet goes through the NAT gateway (private subnets)

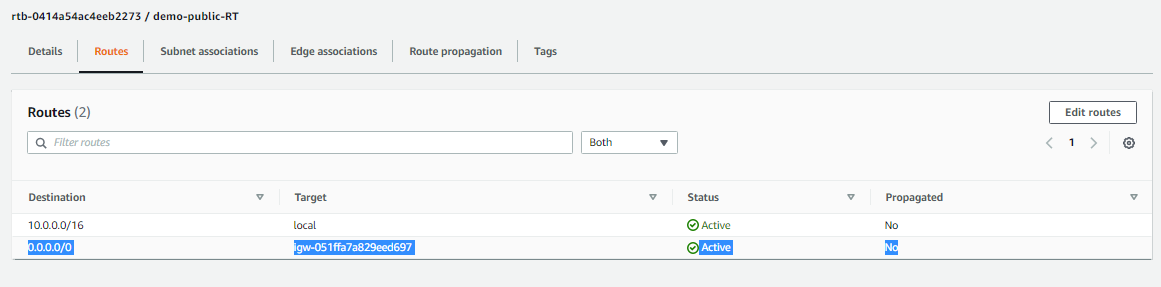


* 1. Associating public subnets with public route table and private subnets with private route table

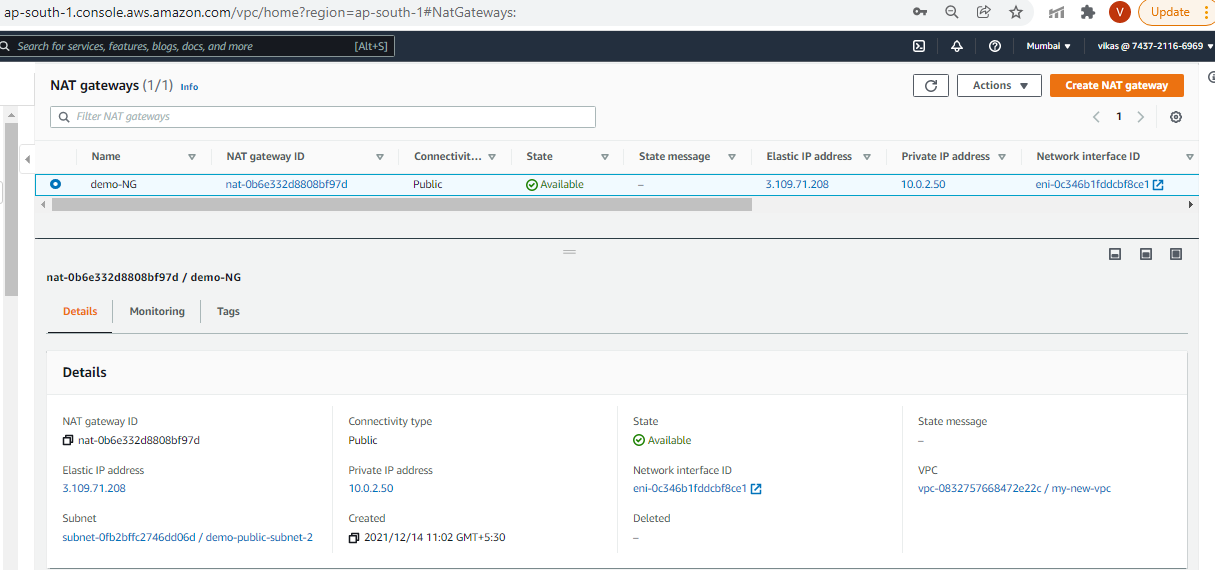




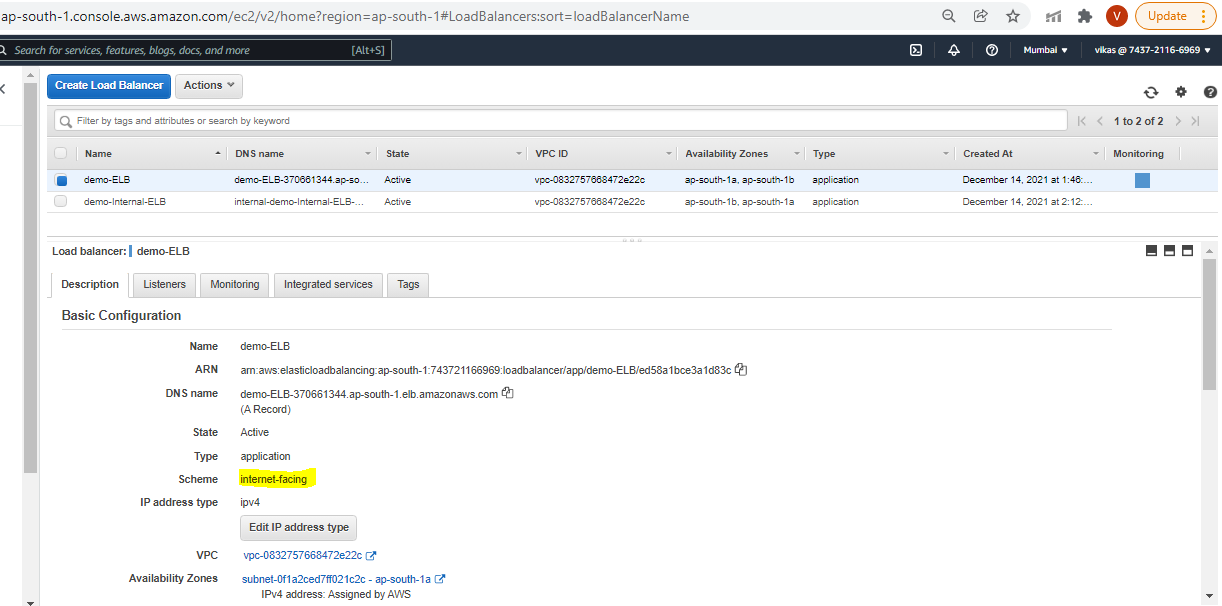
* 1. Internet gateway for public route table

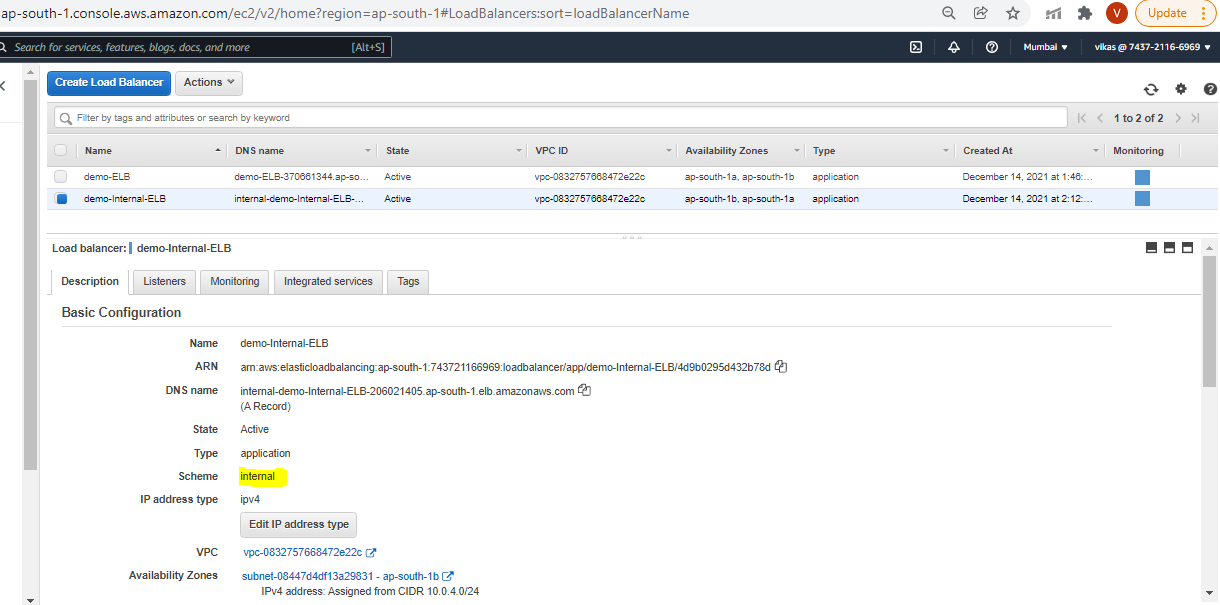


1. NAT Gateway: Created NAT Gateway to enable the EC2 instances in the private subnet to access the internet

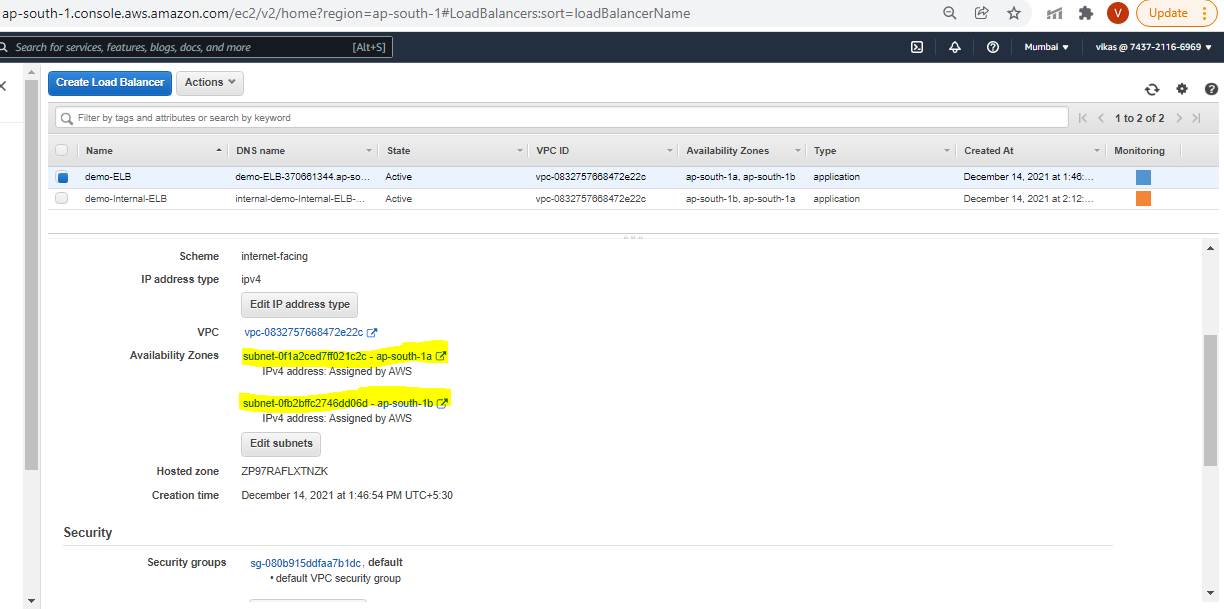


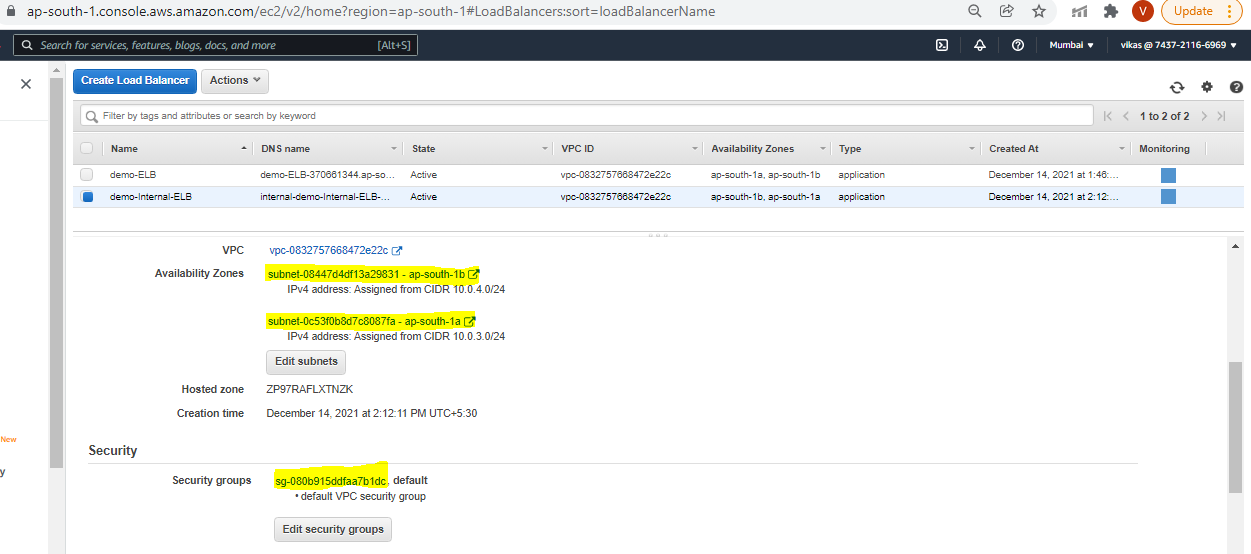
1. Elastic Load Balancer: Created two 2 ELBs, one for frontend tier to only accept traffic from the elastic load balancer which connects directly with the internet gateway and other for backend tier to receive traffic through the internal load balancer.



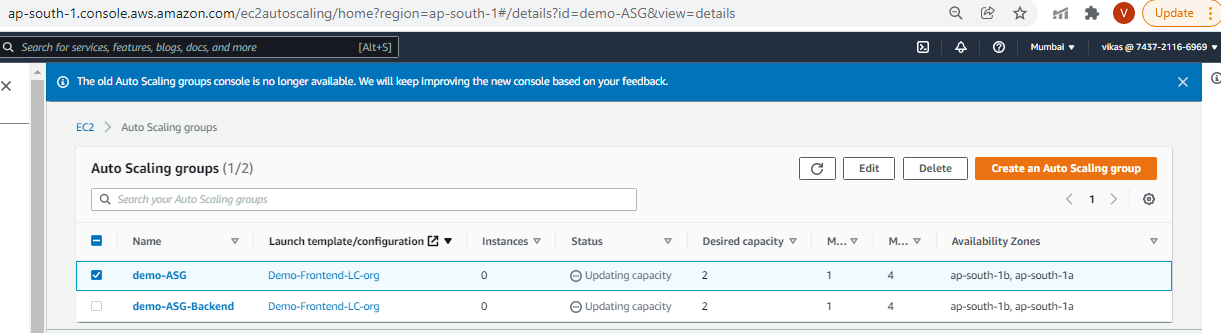


* 1. Selected public subnets for Frontend ELB and private subnets for Backend ELB and attached Security group for them

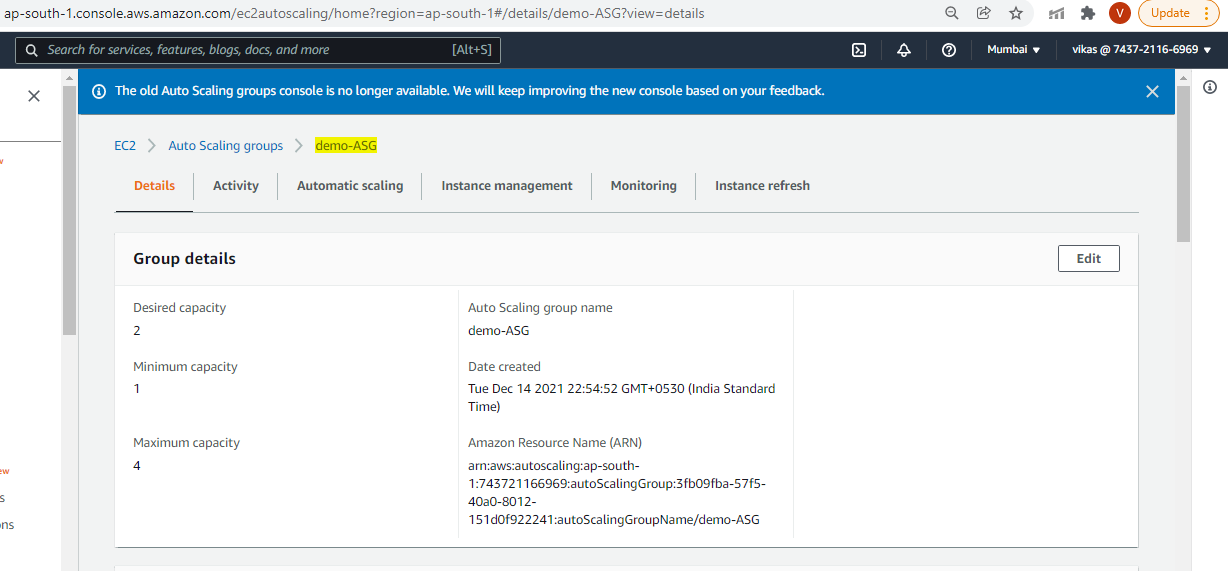


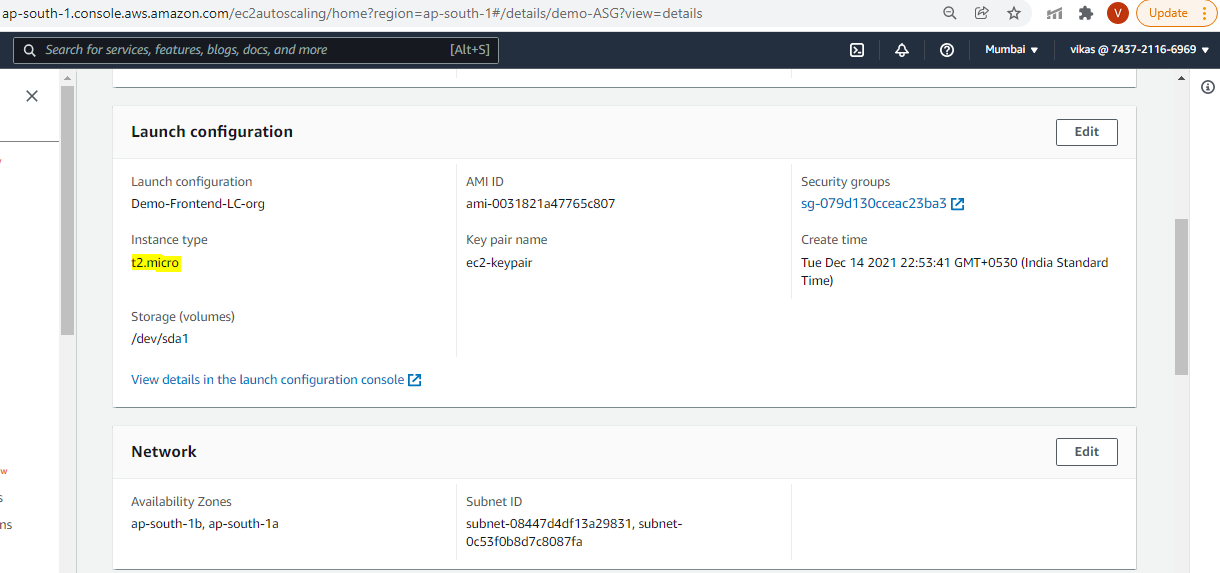


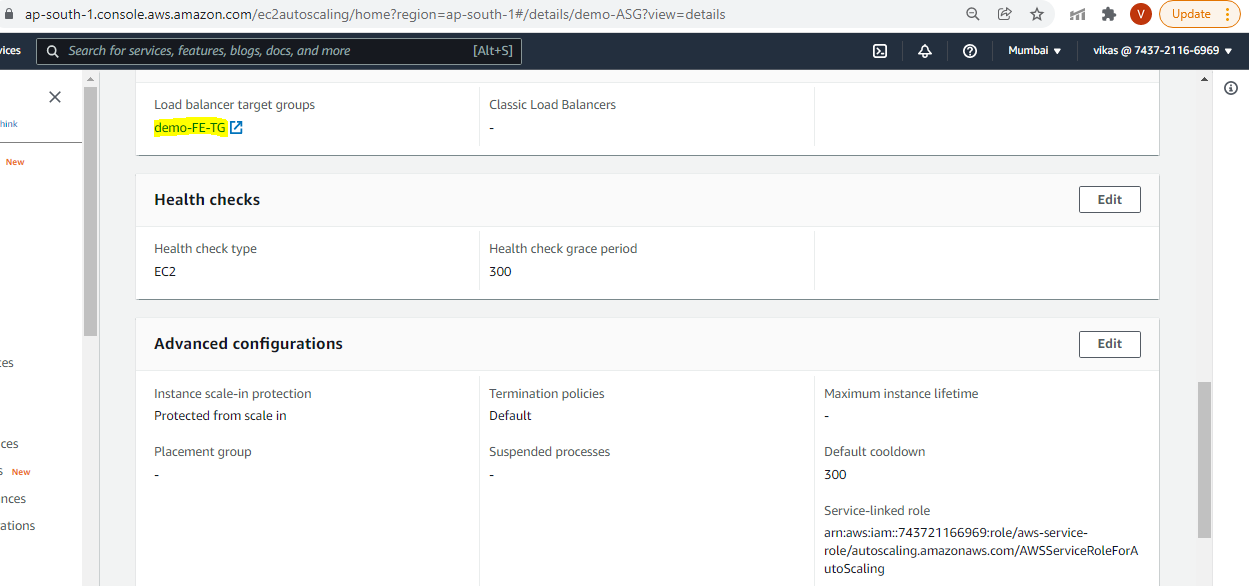
1. Auto Scaling Group: Created 2 Auto Scaling Groups(for frontend and backend) to automatically adjust the size of the EC2 instances serving the application based on need



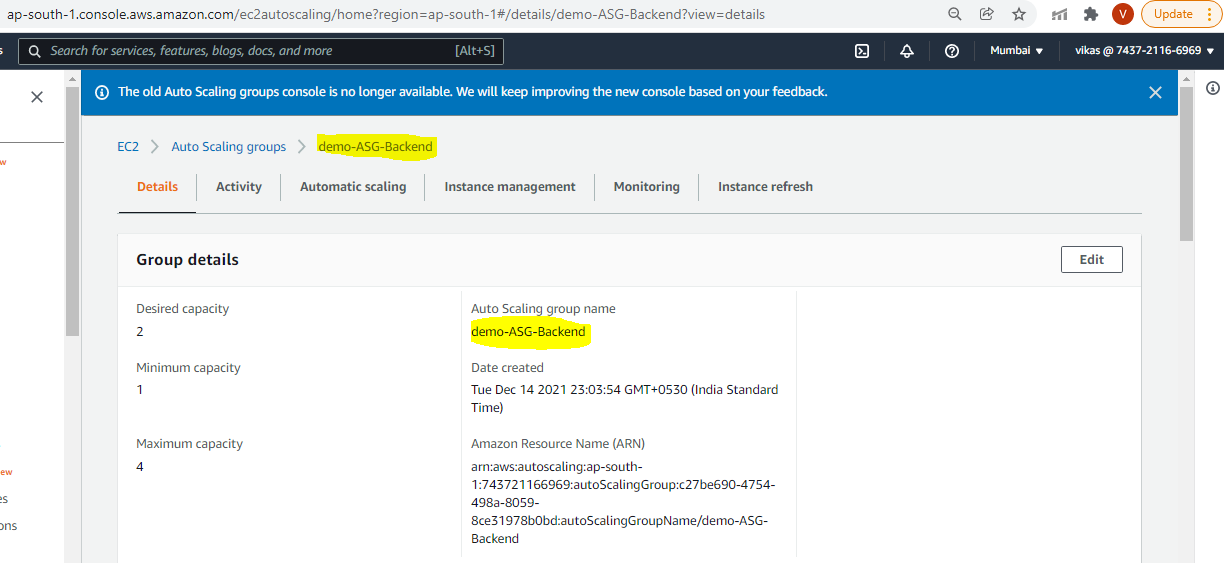
1. ASG for frontend:

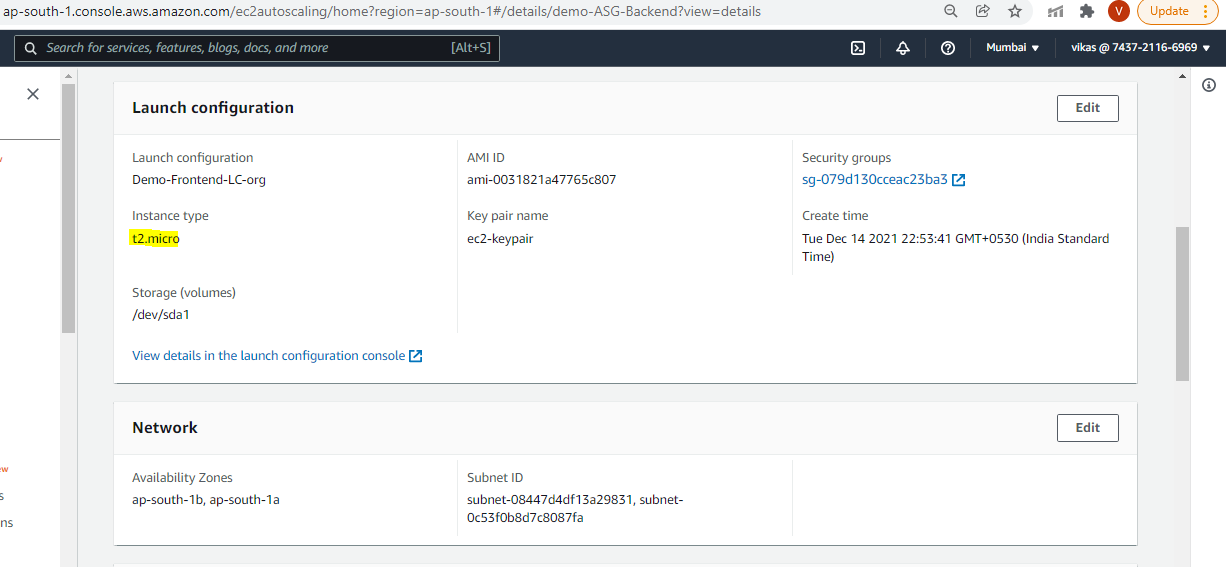


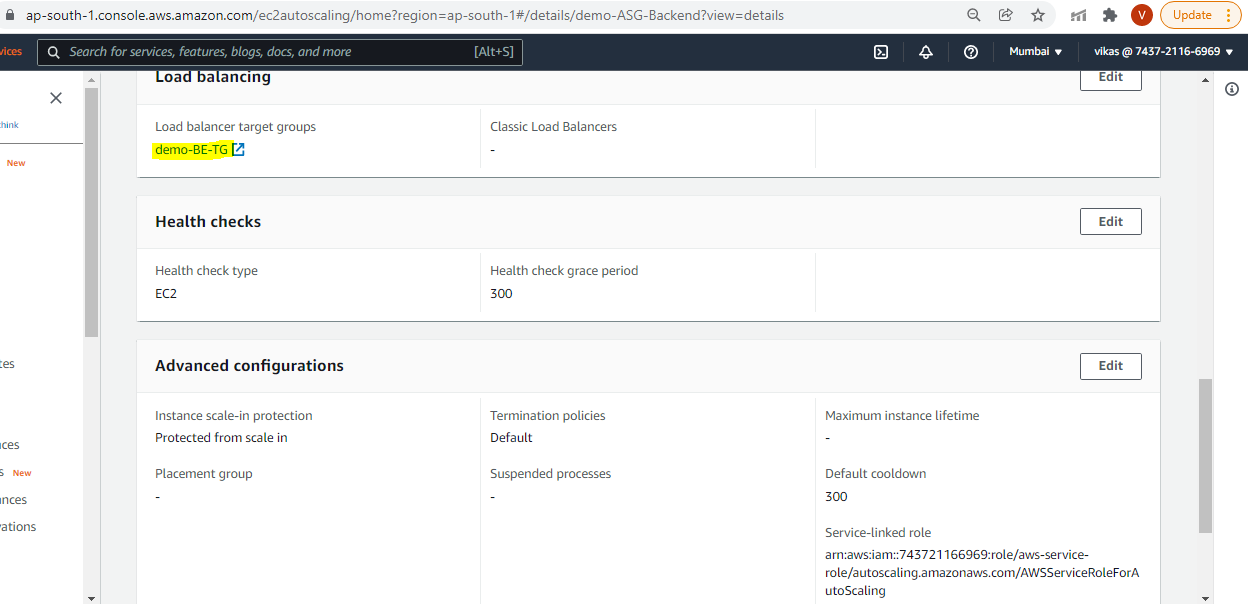




1. ASG for backend







1. Bastion Host: Created Bastion Host to SSH into the EC2 instances in the private subnet

