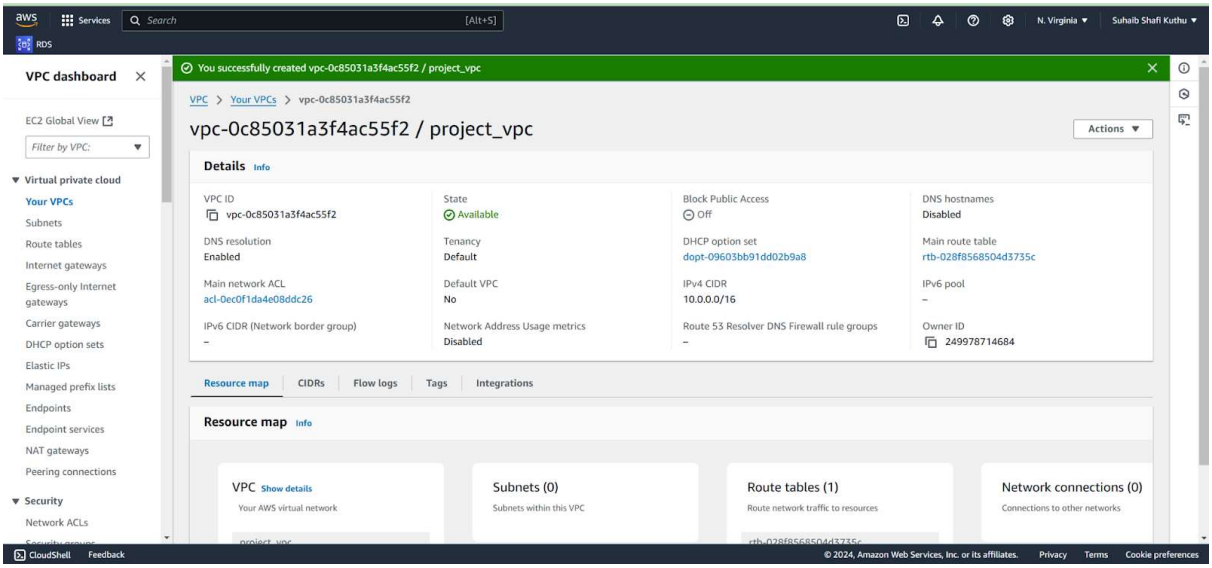
Project2

Below are the steps for the procedure to be followed to achieve the required architecture

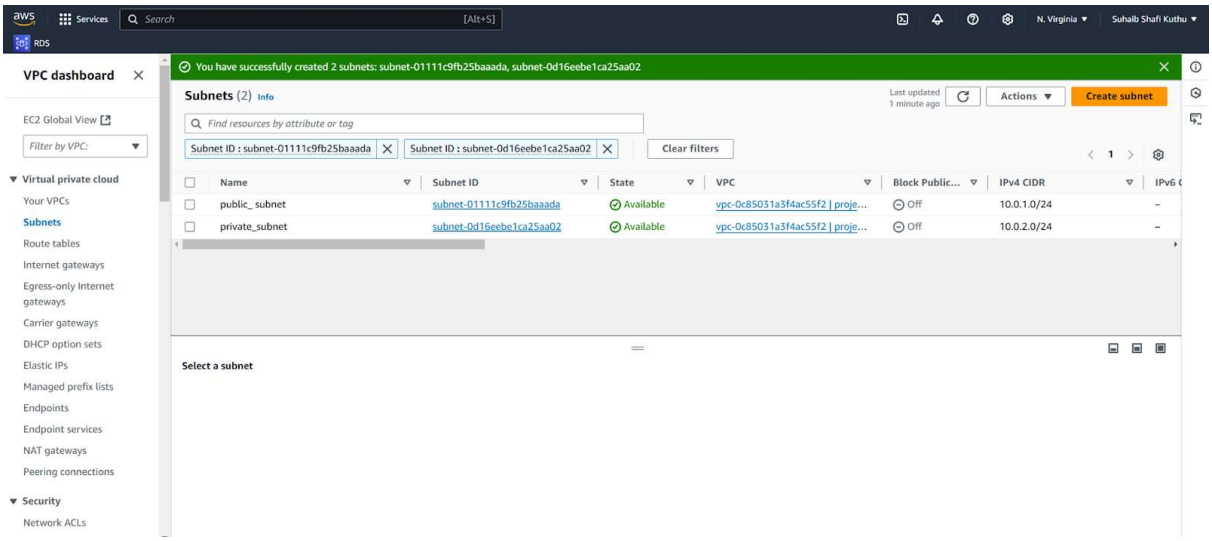
**They are concerned about the s
ecurity of the environment, so they have decided to virtually isolate their network from the rest of the customers and the rest of the
environments in the same AWS Cloud Account**

Step1:-

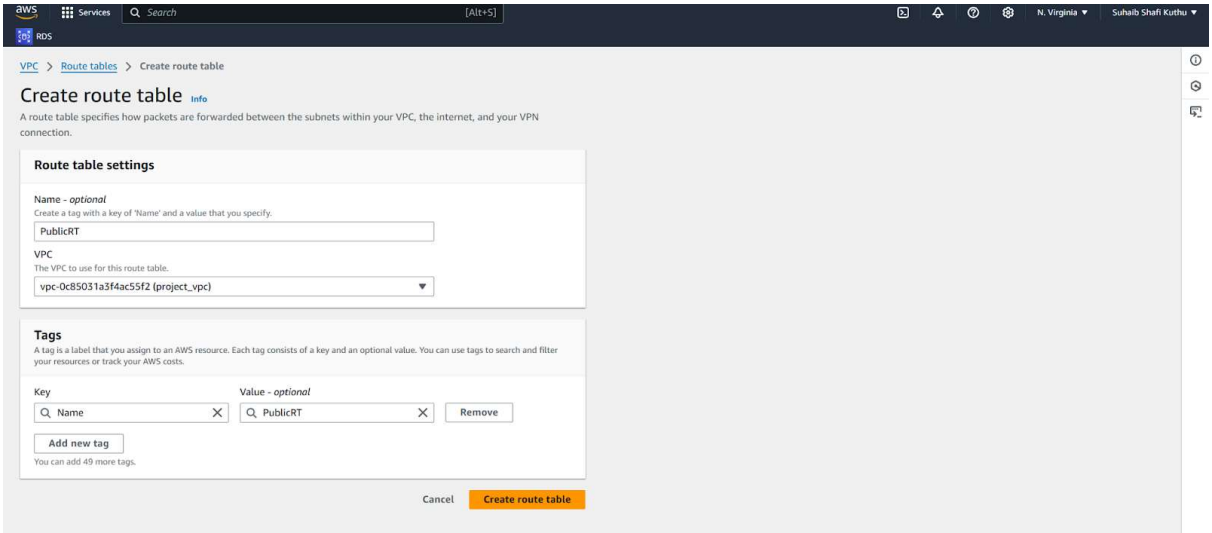
Create a vpc where the front end is kept in the public subnet and backend in private subnet.

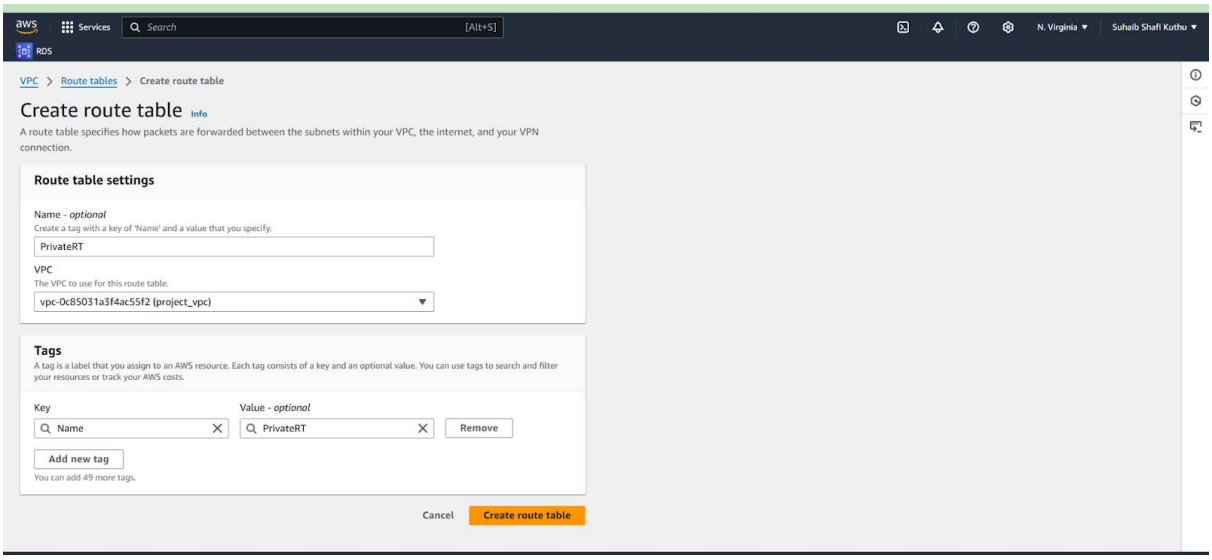


Step 2:- create subnets in the vpc with public access and private access

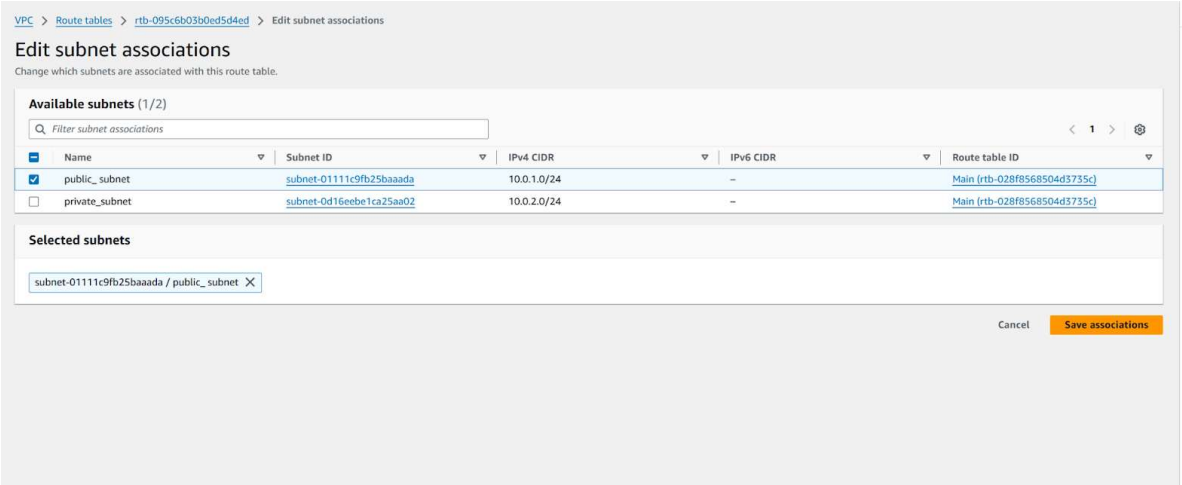


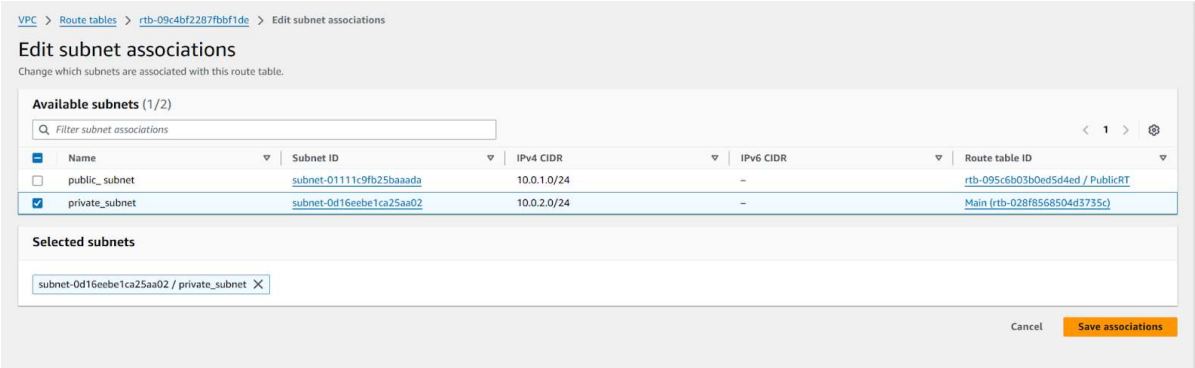
Step 3 :- create route tables for public and private subnets and associate accordingly



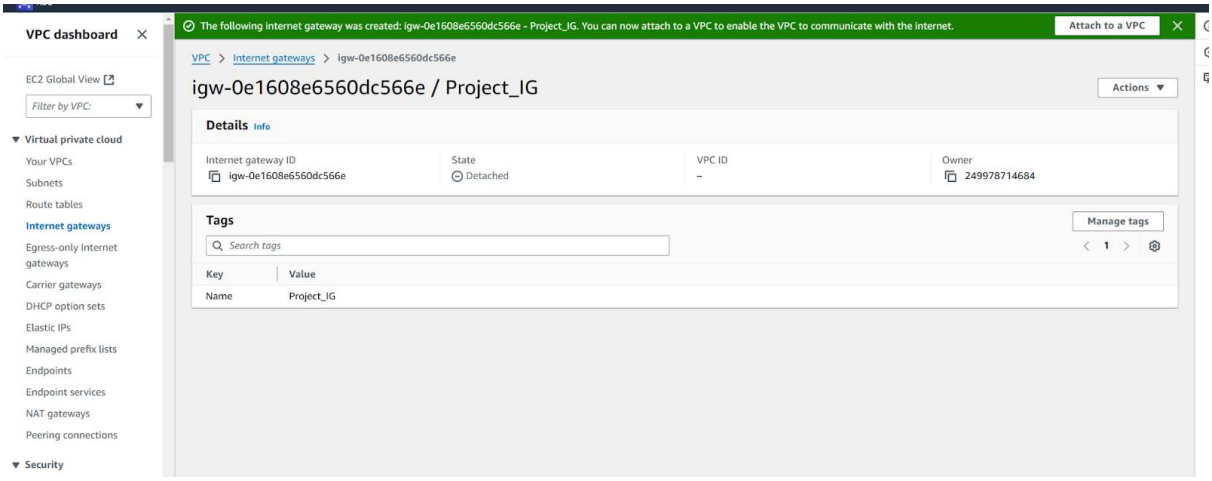


Step 4
Subnet associations ,assoiciate public subnet with publicRt and private subnet for public rt

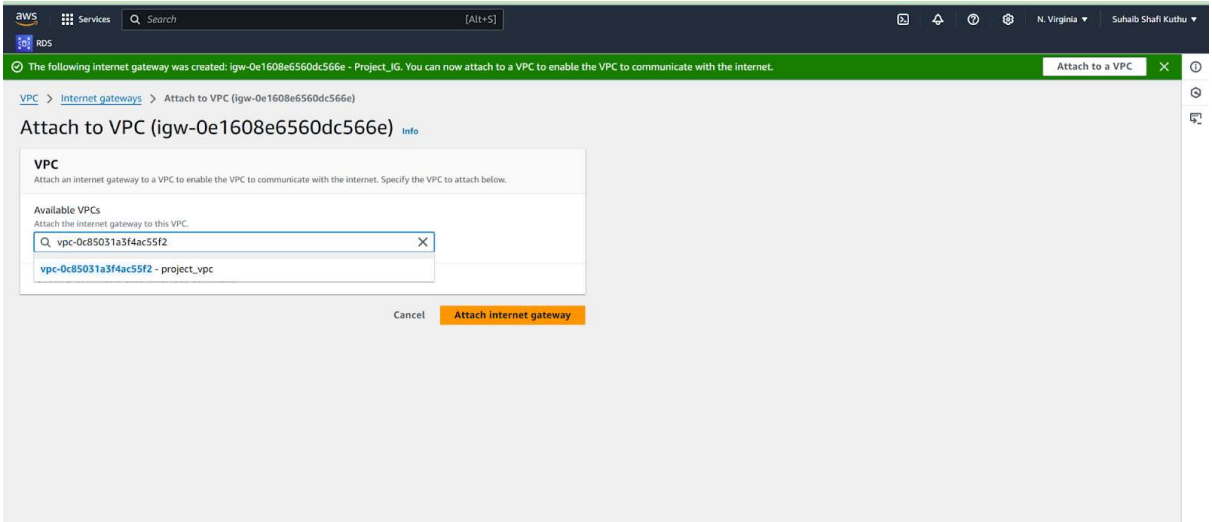




Step 5
Create an internet gateway and attach it to vpc to make it available outside as public url.

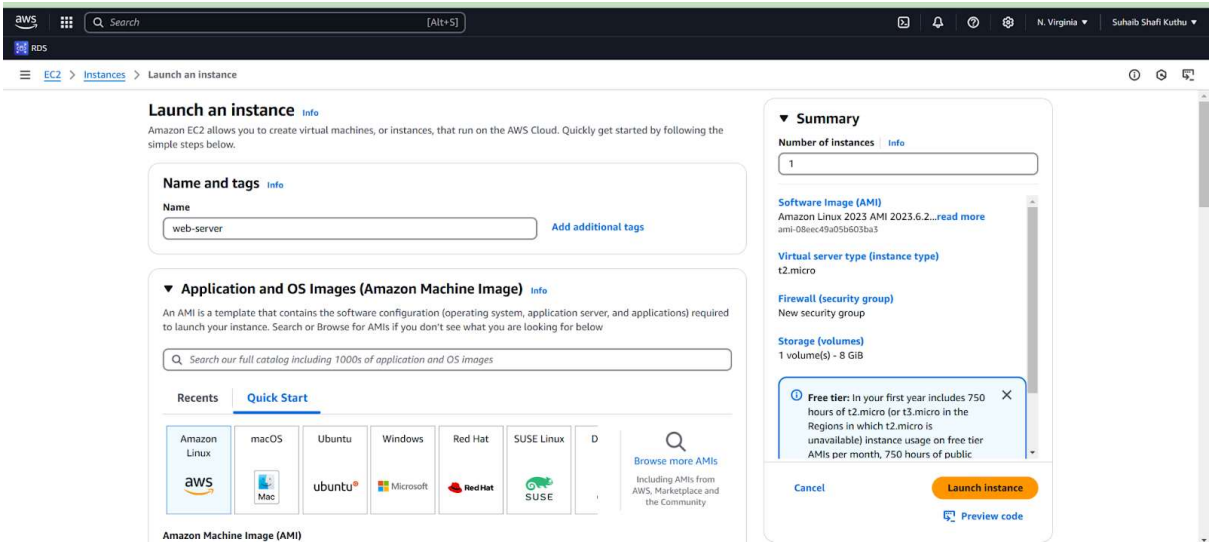


Attach it created vpc



Due to the budget issue, the company cannot afford a dedicated DB engineer, So they are willing to outsource the DB management from a Cloud provider, to store and maintain the customer information received by PHP application. You must pick the right solution from AWS, which should be a Platform as a Service. It should also Provide high availability, patching, and back

Step 6 :- now go to the ec2 console and launch an ins tance to connect to database,the database and the server should be launched in the the same vpc that was created above.



As we need a platform as a service Db due to lack of dedicated db engineer, we can go with amazon RDS that could provide database solutions,as for the procedure follow the following steps.

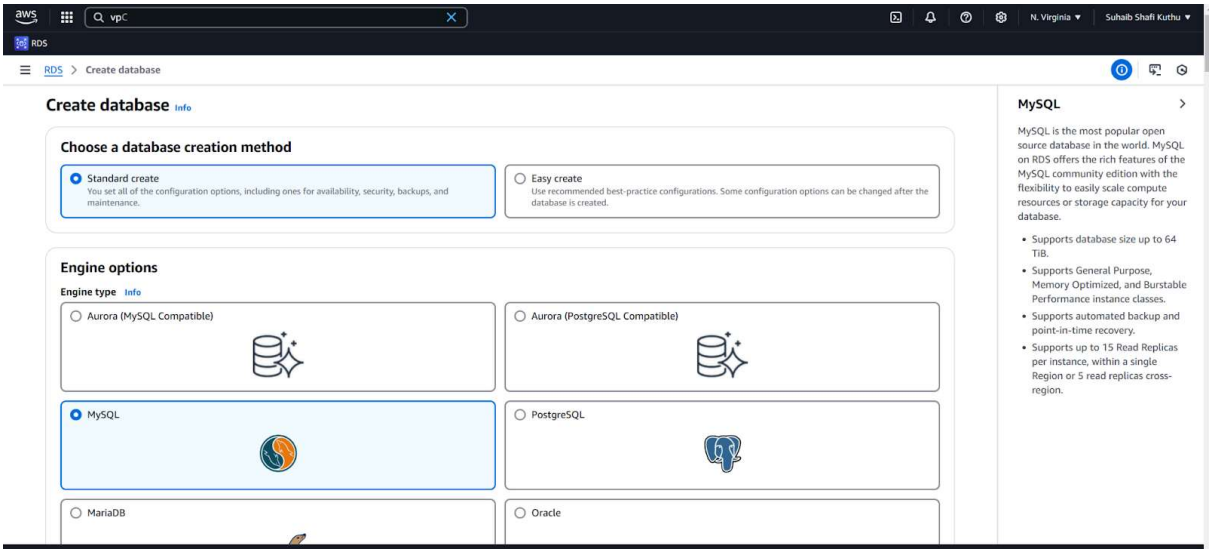
1)Use Amazon RDS (MySQL or PostgreSQL) for a fully managed database solution.

2)Place the RDS instance in private subnets to enhance security.

3)Enable Multi-AZ Deployment for high availability.

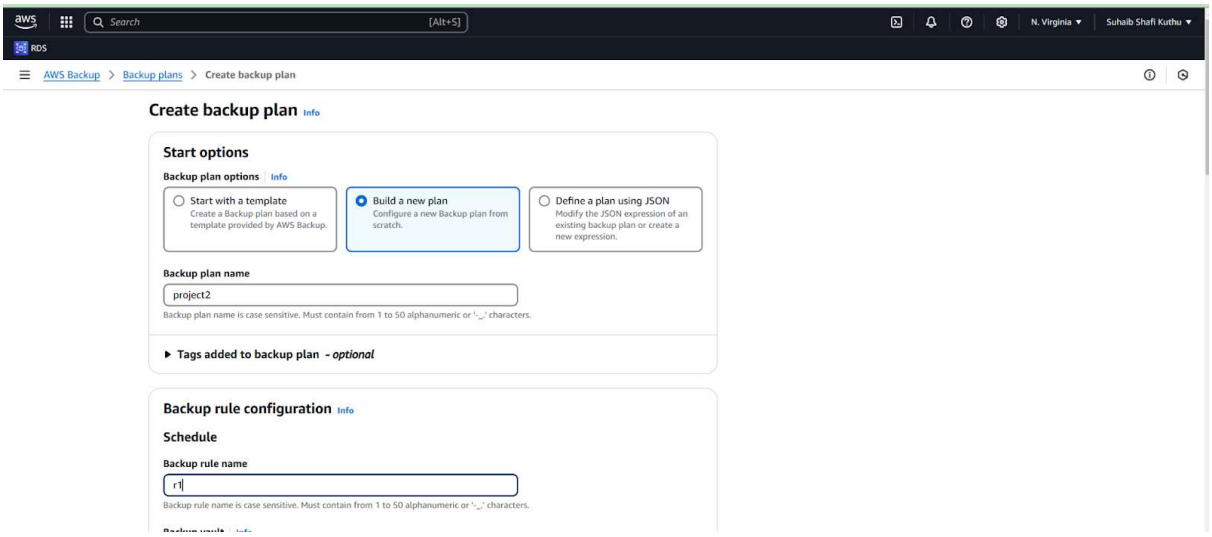
4)Configure automatic backups, snapshots, and patching.

5)Create a DB Subnet Group to define which subnets RDS can use.



To retain the backup enable backups in database settings

Backups for your Web service may include creating snapshots of the instance or creating an Image of the Instance. Or
Enable AWS Backup for both EC2 and RDS instances.
Configure daily automated backups and retain snapshots based on the company's retention policy.
Use Cross-Region Replication to store critical backups in another region for additional resilience.

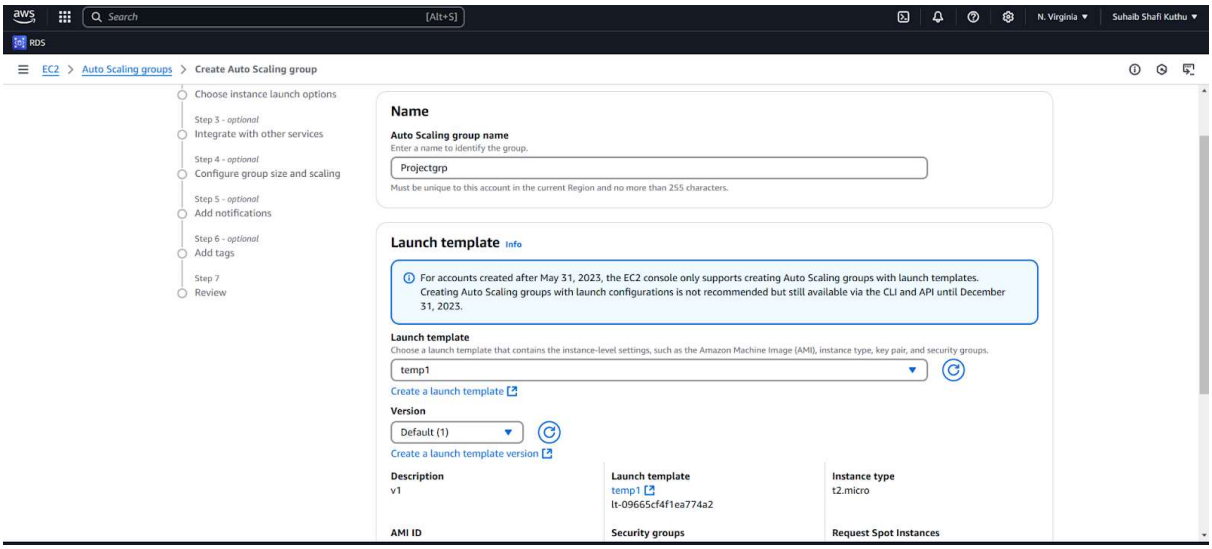


Design a dynamic website where the customers can enter their details, which should be stored in a database,
They are uncertain about the traffic pattern that how low or high it can be, so they have a requirement that the environm ent should be running at least two EC2 servers all time, and when there is a high load, they must burst up to four servers in total

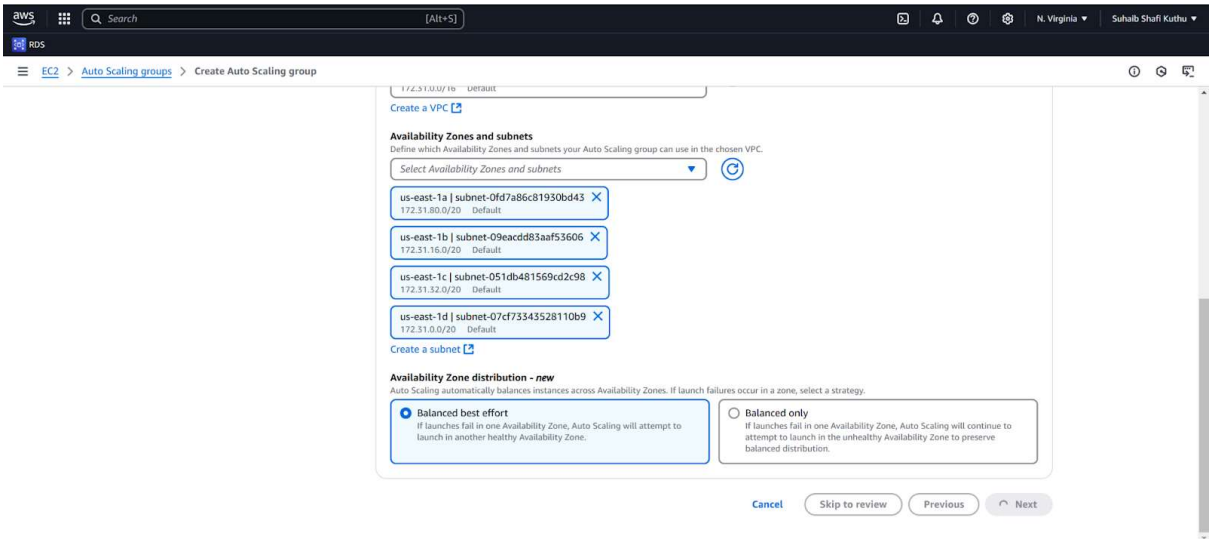
Both of these could be achieved simultaneously

First go to the ec2 console and launch an ec2 instance,,to host the application download a web server in the instance

then create an ami of the instance ,and create a launch template out if that ami nad use it in the creation of auto scaling goup

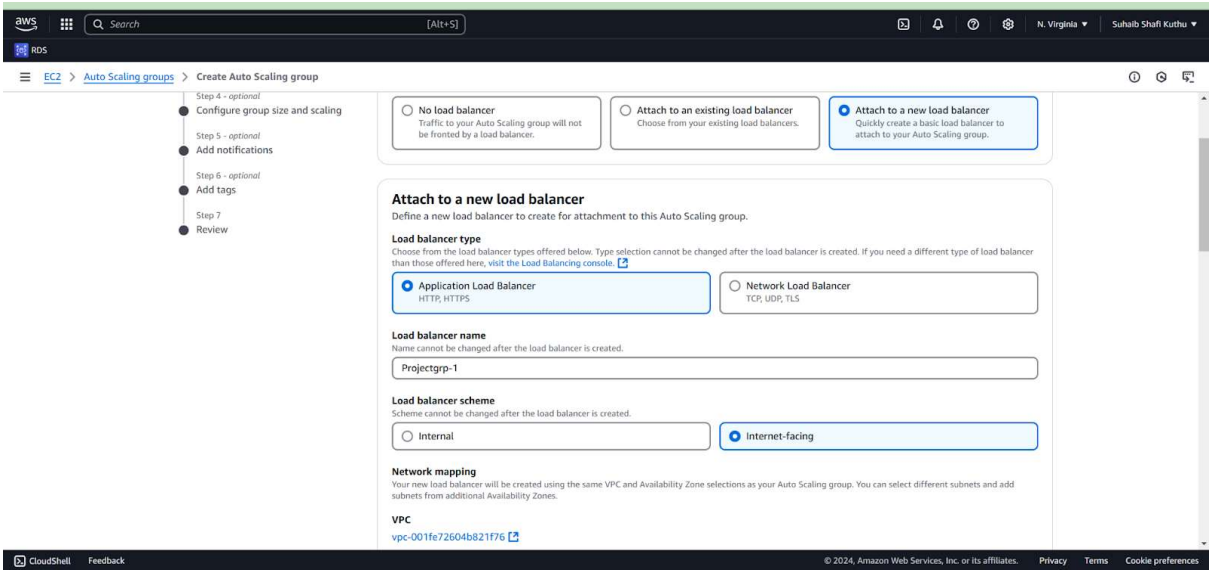


In the further steps choose the availability zones in which the instances should be launched for high availability

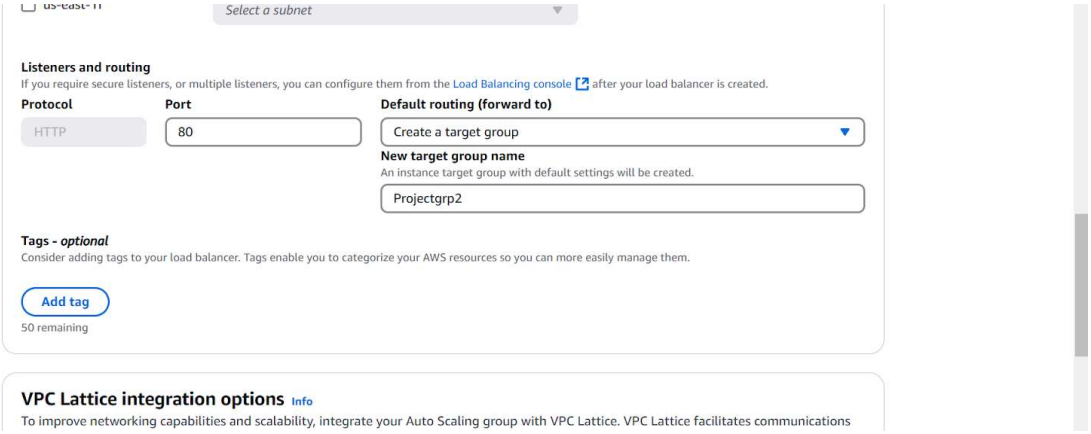


The application should be highly available, even if a VM fails to respond to queries, there should be a mechanism to shift the connection to another healthy VM automatically

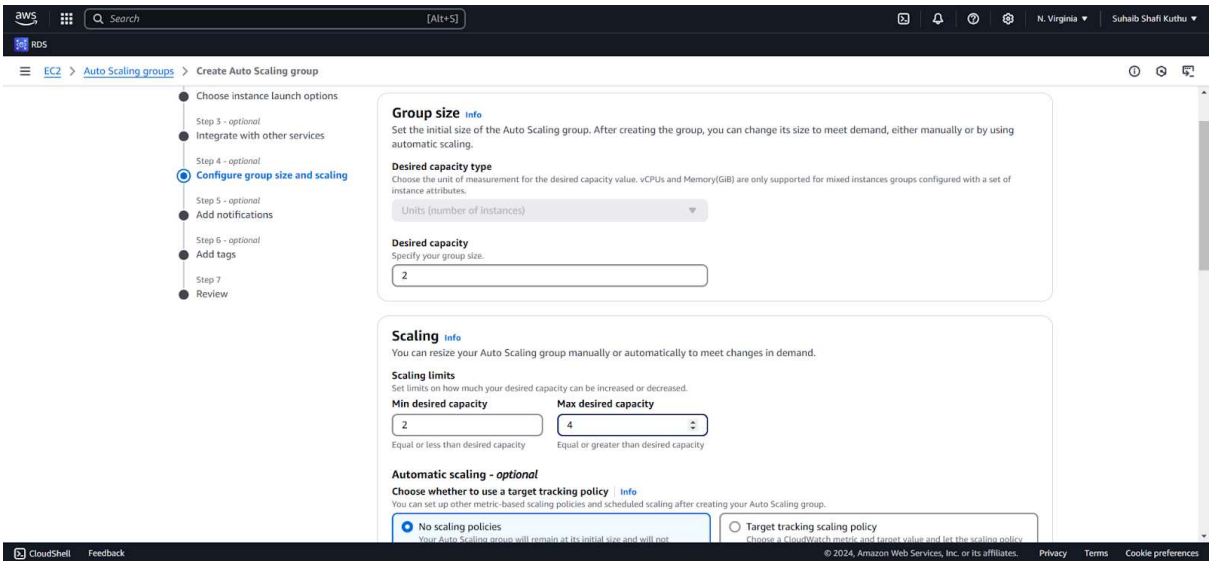
Here we will launch a load balancer within the auto-scaling group for load distribution



Also create a target group for load bancer to distribute the traffic to

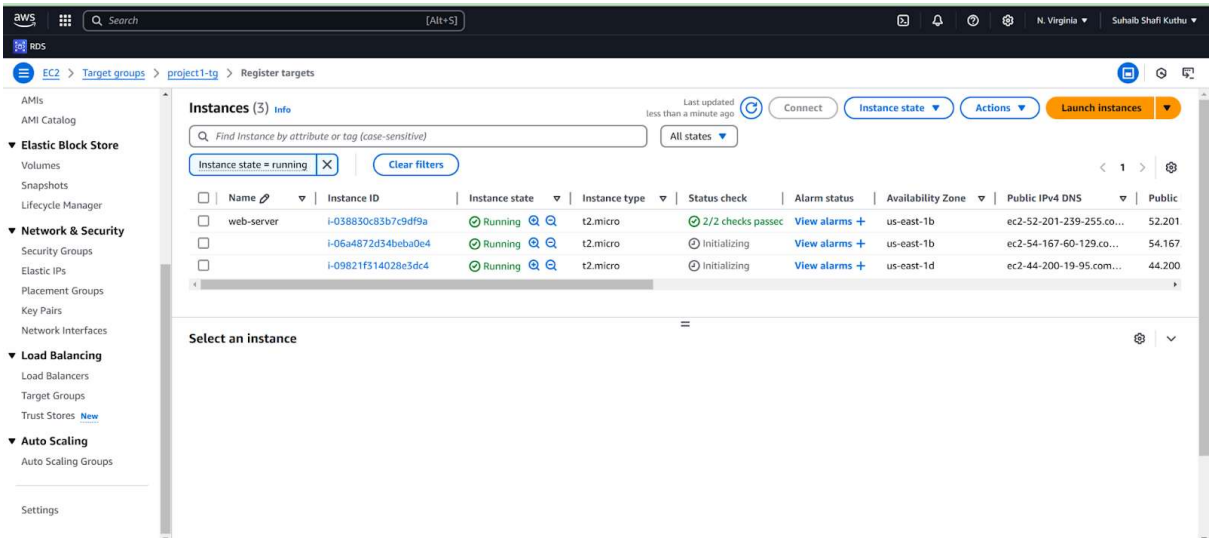


Now in group size and scaling specify the minimum as maximum desired capacity as per the project it is 2 and 4



Go Ahead and launch

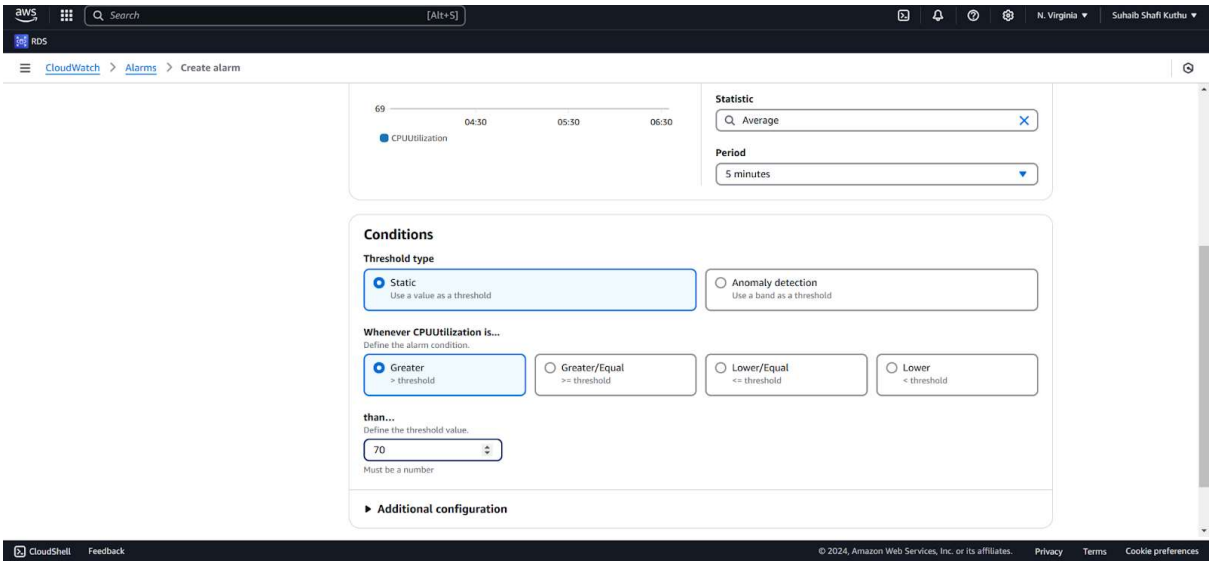
As we can see two more instances have been launched by ASG



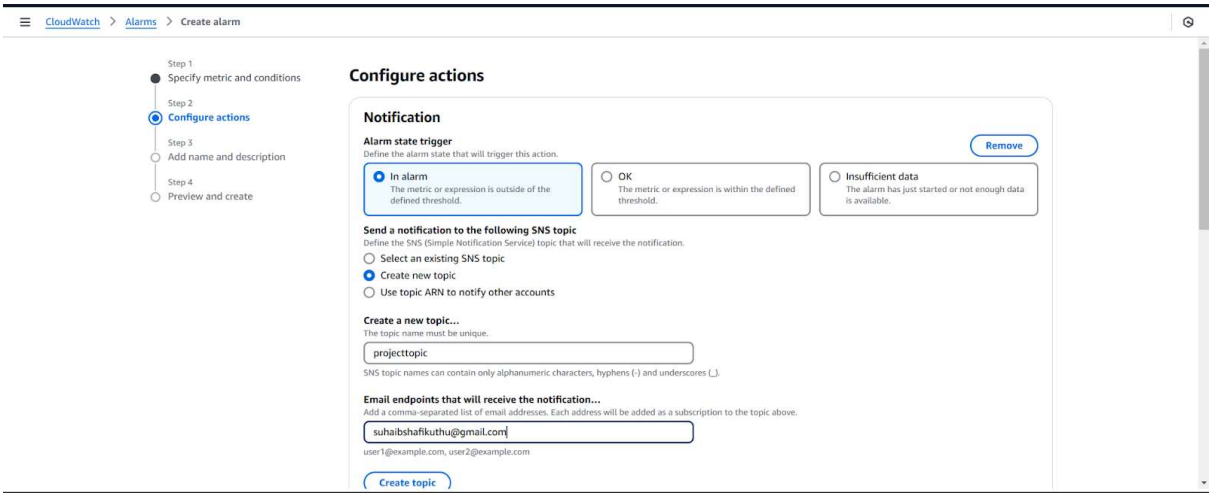
Now the company cannot afford a dedicated engineer for monitoring, so you must automate the incident management through an event notification. Anytime there is an increase and decrease in the VM's due to high or low traffic, you must receive a notification via email.

To achieve this step we can go to billing and alarms and create and alarm and in the select metric secion choose cpu utilization

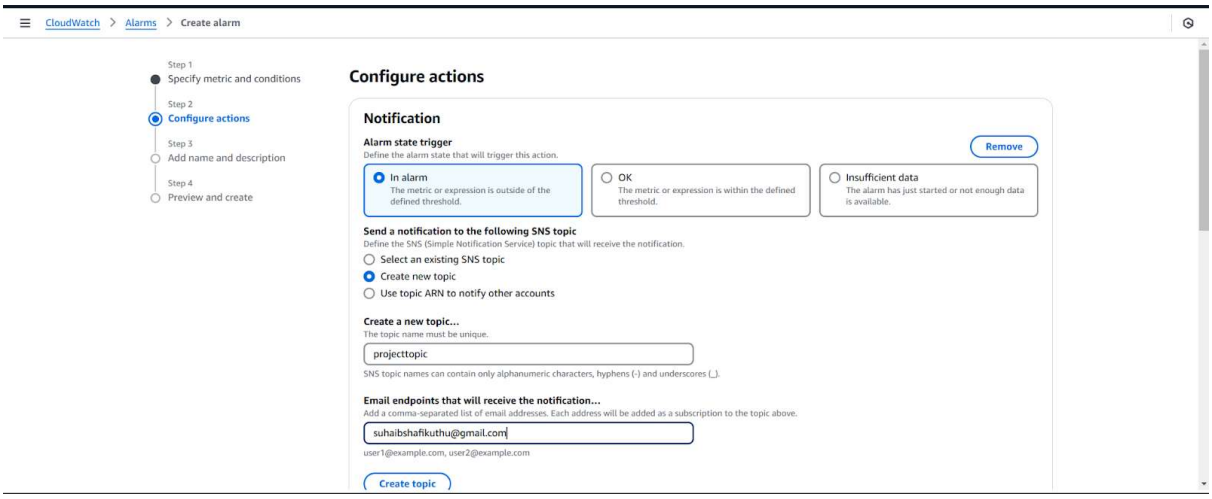
Now in specify metric and condition choose greater than 70%



Now in configure actions configure sns topic to send a notification for the same

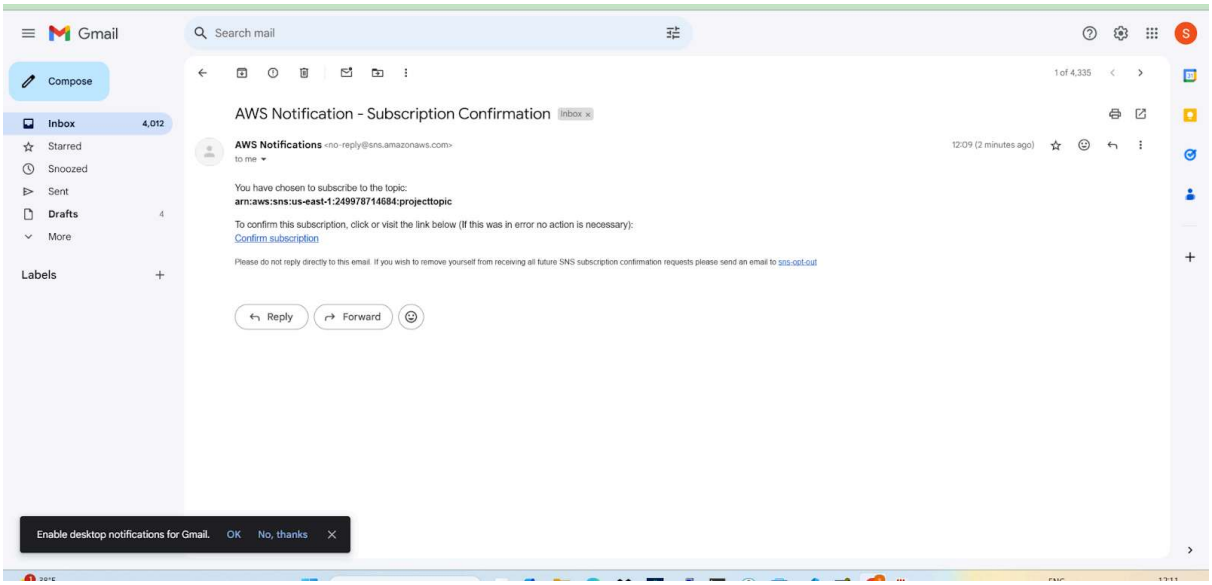


Add name and description



Go on and create the alarm

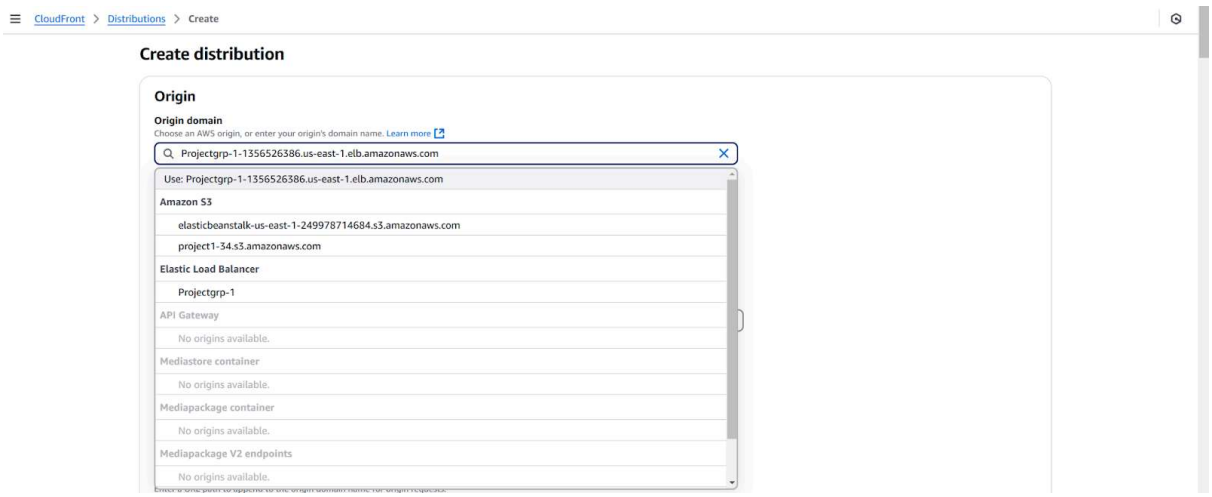
One more important thing dont forget to accept subscription which would be sent to registered e-mail Id



Your Dynamic Website should also be cached globally, so users worldwide can access it with less latency. The customer is okay if we get an unfriendly AWS generated URL for accessing the website

To achieve this follow

Create a CloudFront distribution to cache our application globally. Navigate to your CloudFront and click on create distribution Under web section click on get stared On the Create Distribution page,under Origin Settings, choose the ELB that you created earlier Give the original path as project.php



Choose the origin domain the dns or the load balancer created above amd keep the rest settings as default

Hence following the steps we can design a highly available php application with the need requirements.