AWS Cloud Modernization Capstone project

1. About the Project

My project involves creating a web-based application with both a front-end and a backend. My main focus is on developing and hosting this application entirely within the AWS Cloud environment..

2. Why AWS Cloud?

I'm using AWS to develop and modernize this application. My approach involves building the project step-by-step, first getting a simple version running, and then improving its security by adding layers of protection, and boosting its performance by using various AWS services.

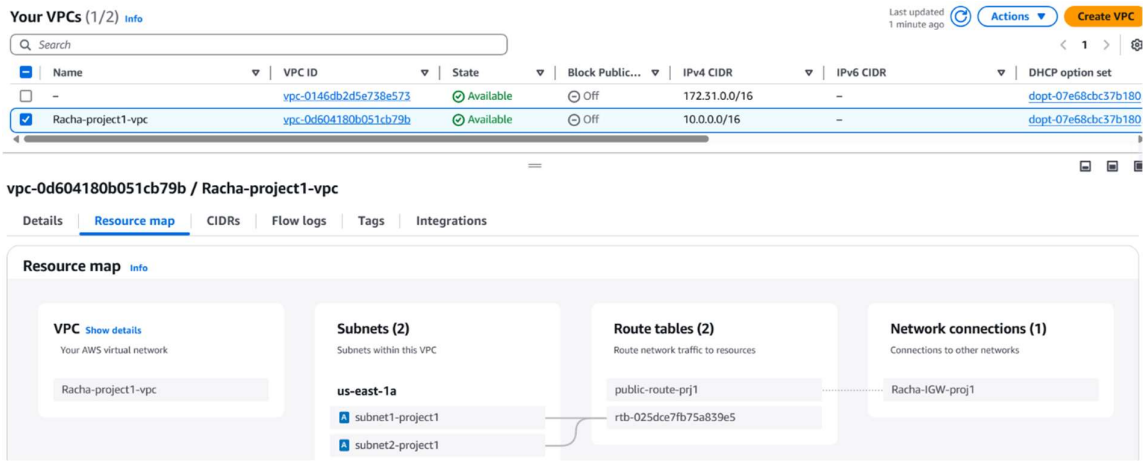
3. Technologies I'm Using

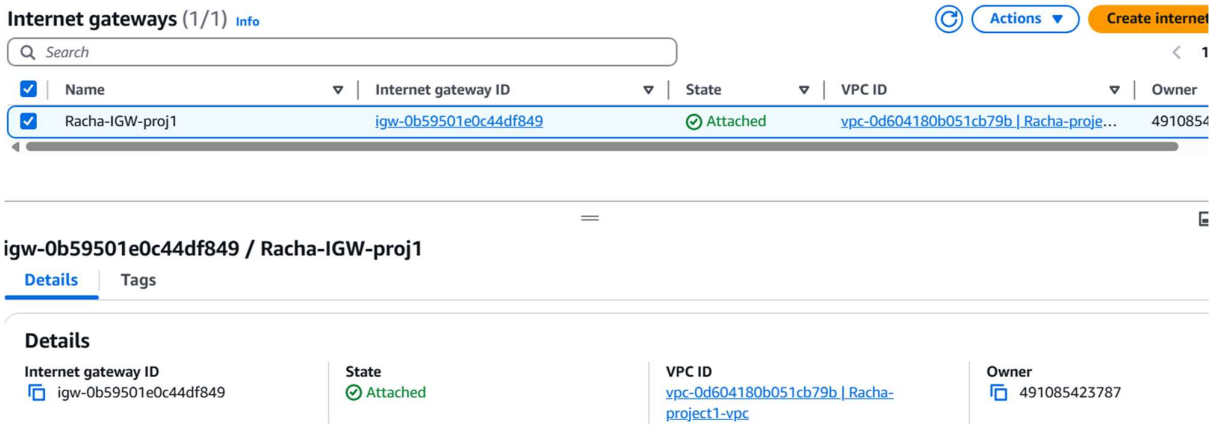
For this project, I'm working with a simple web application (like a "Hello World" app). Here are the key AWS services I have used to integrate:

- **Application**: A basic web app
- **Hosting:** AWS EC2 instance with Nginx or Apache web server
- **Security Features:** IAM, KMS, CloudFront
- **High Availability & Performance Features:** Read Replicas, Auto scaling, CloudWatch

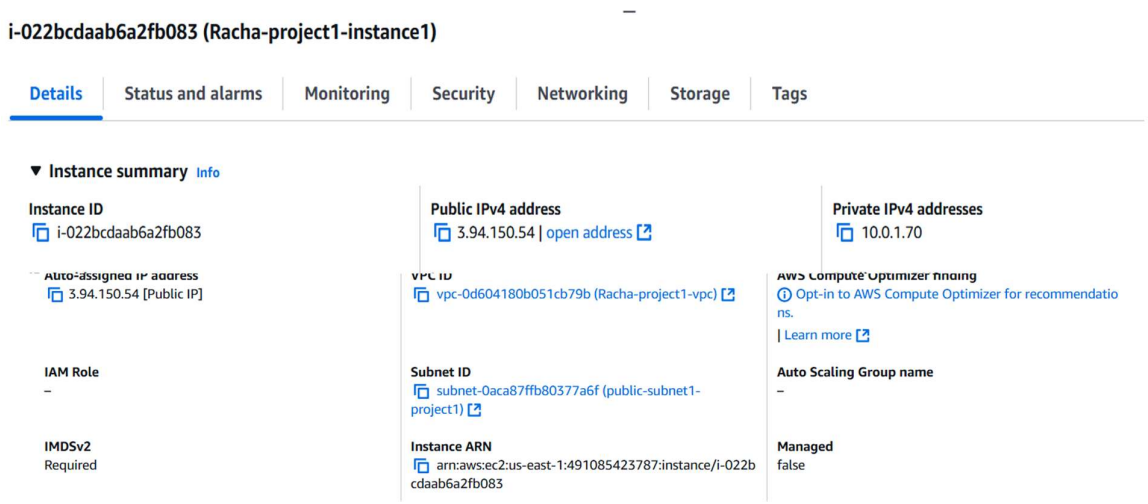
Task1:

VPC:

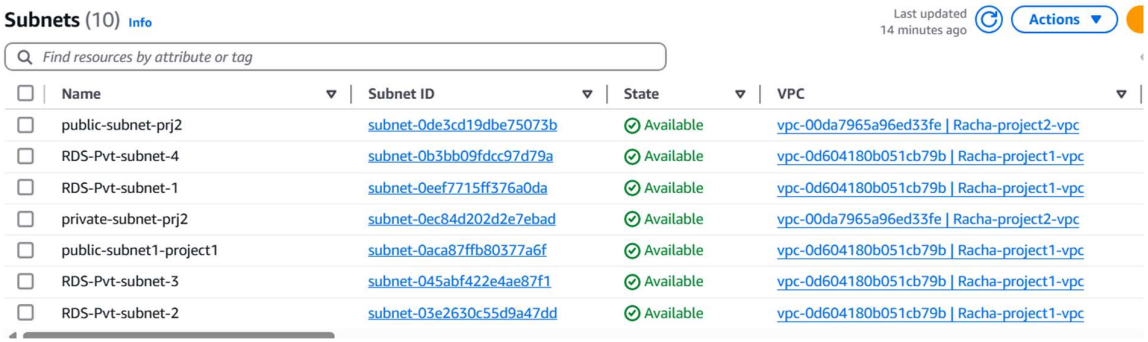


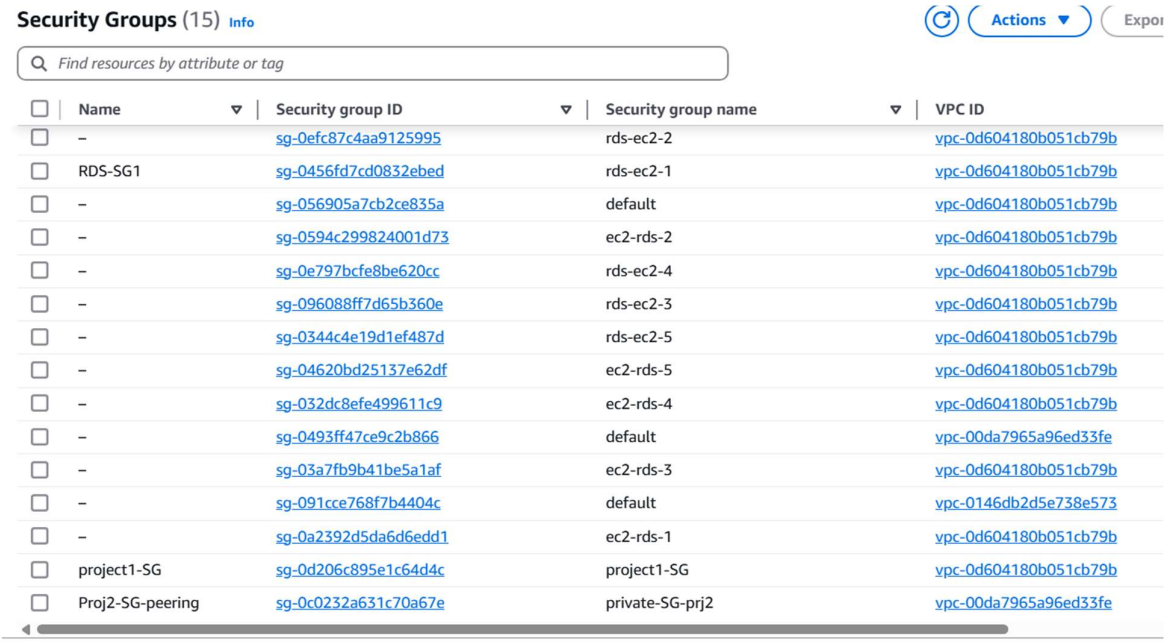


EC2 Instance details:

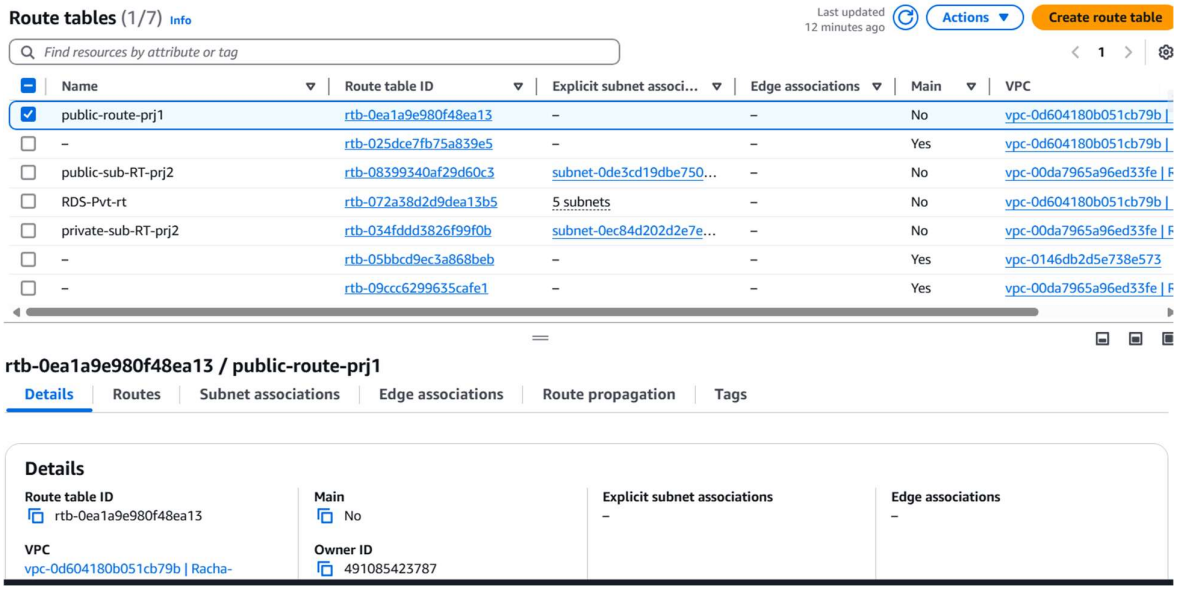


Subnets:

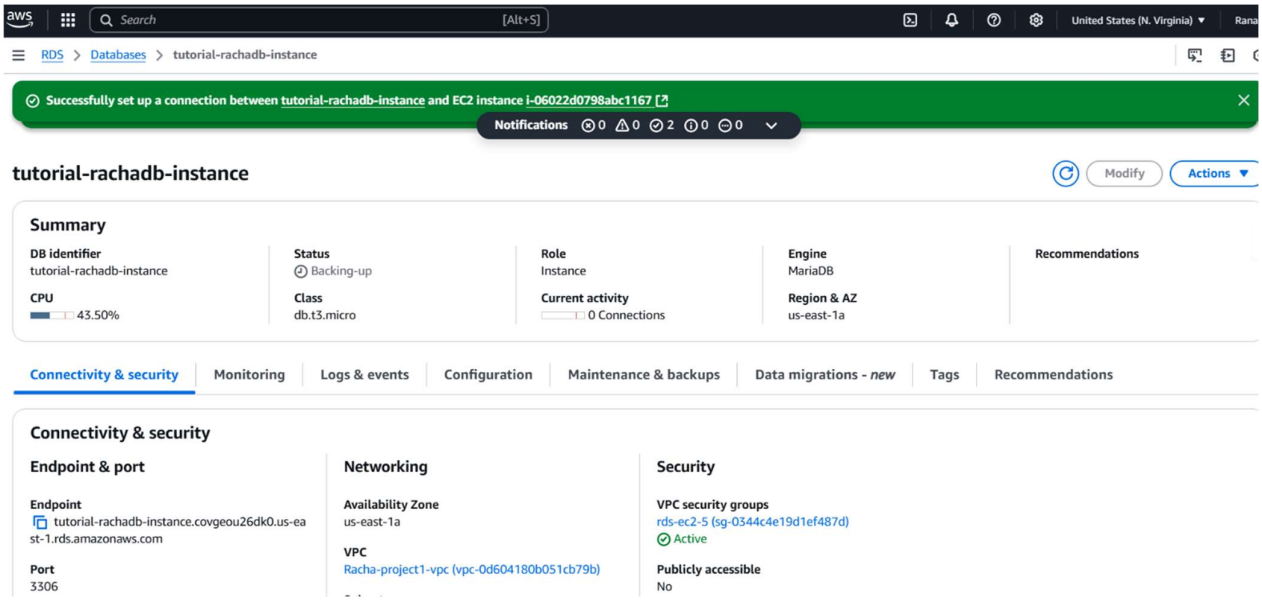




Route tables:



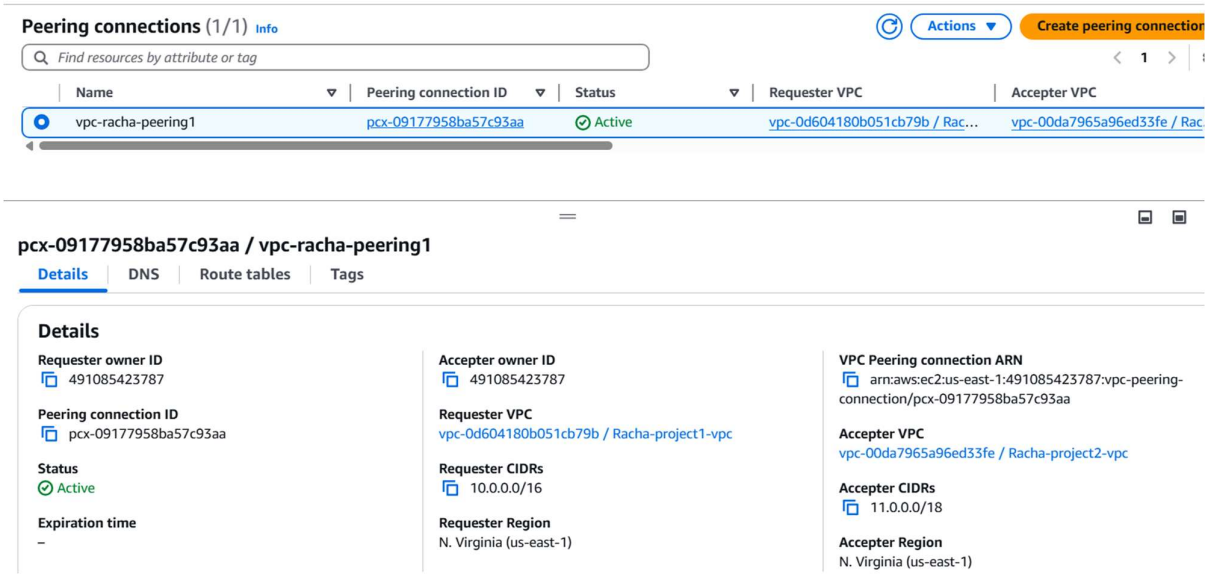
RDS Database:

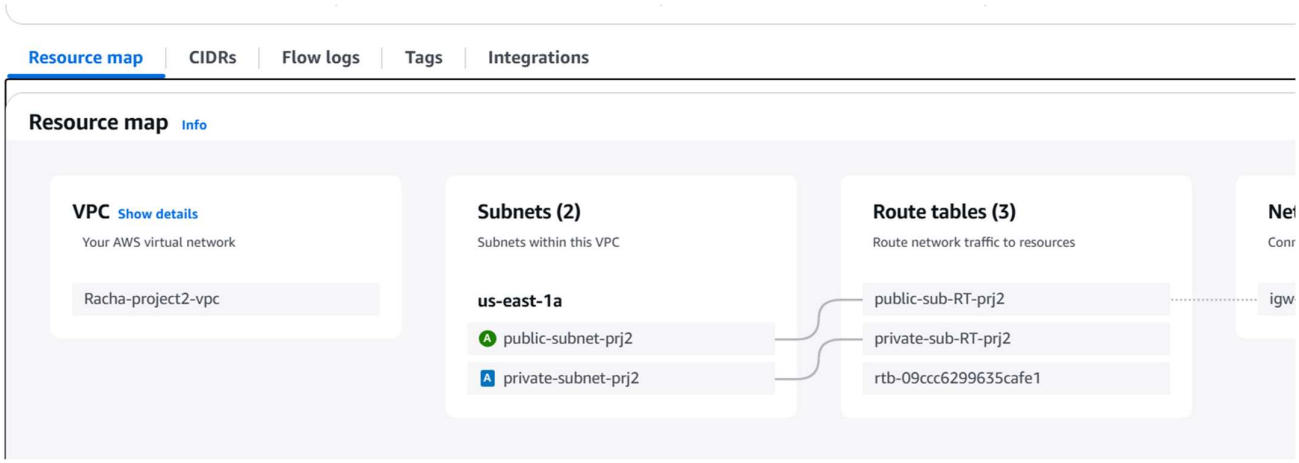


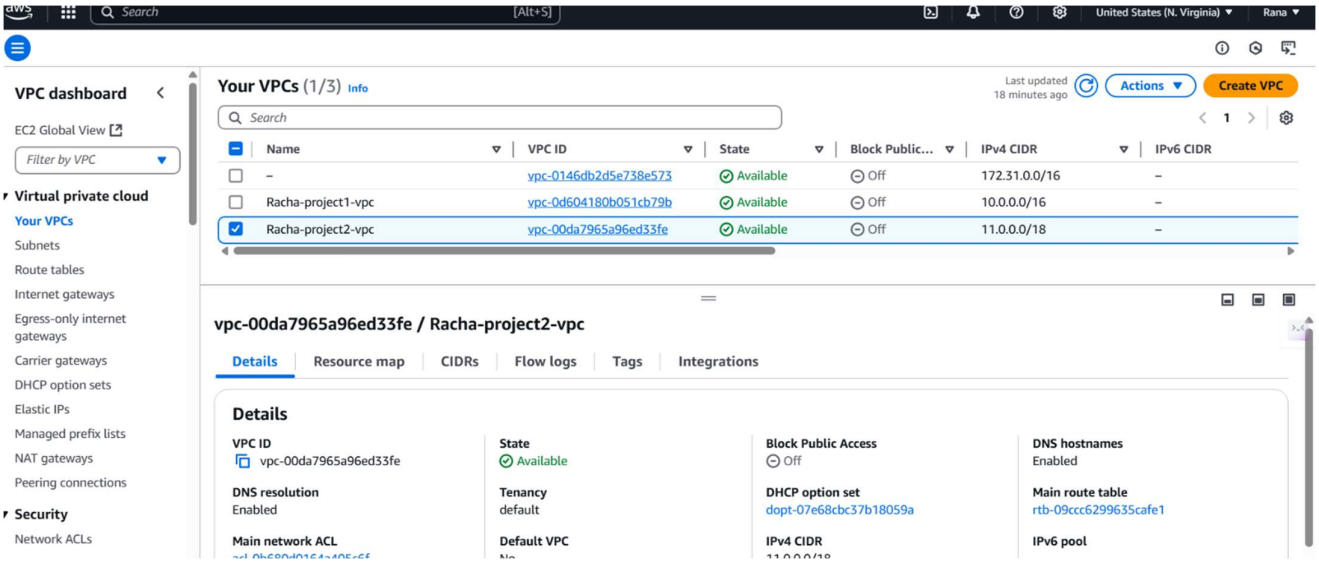
Webserver Output :

Task3:

Peering connections b/w 2 VPC's:



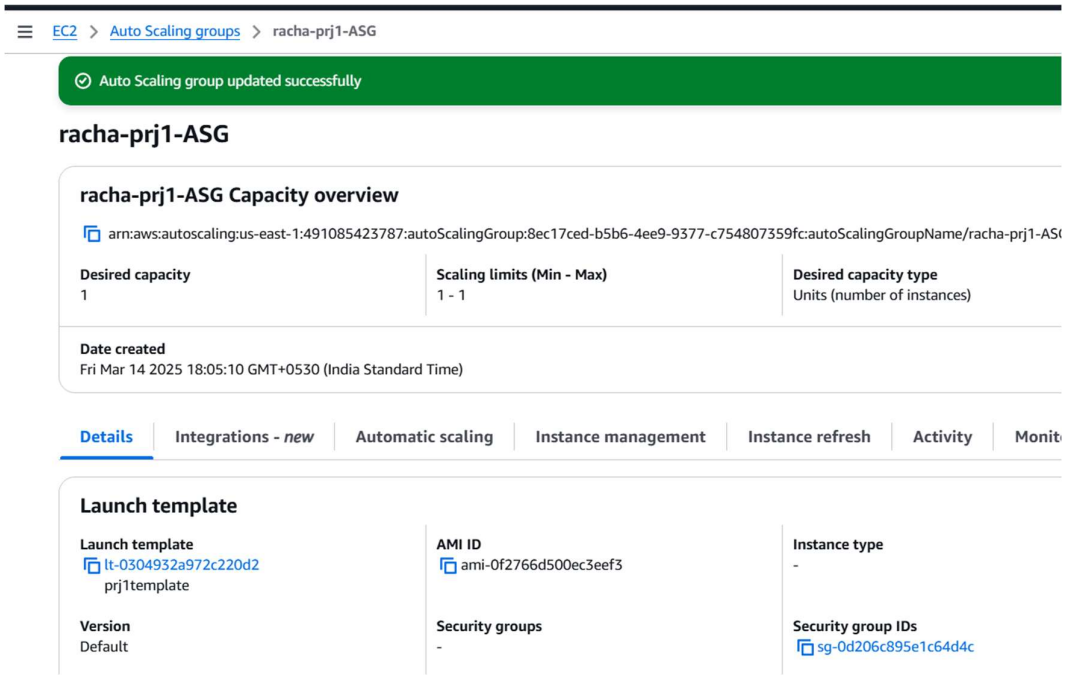


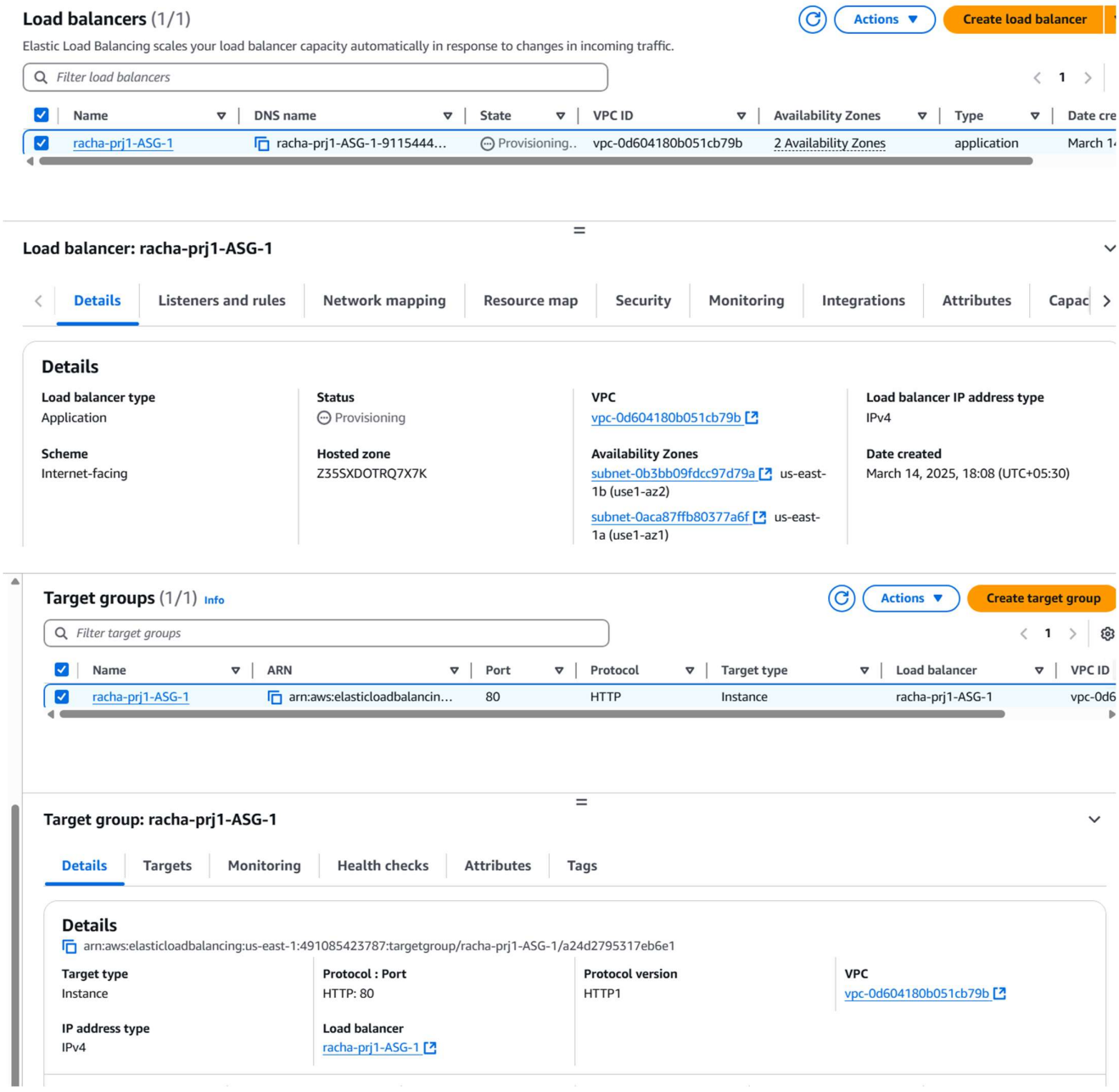


```
[ec2-user@ip-10-0-1-30 ~]$
[ec2-user@ip-10-0-1-30 ~]$ ping 11.0.2.81
PING 11.0.2.81 (11.0.2.81) 56(84) bytes of data.
64 bytes from 11.0.2.81: icmp_seq=1 ttl=127 time=0.709 ms
64 bytes from 11.0.2.81: icmp_seq=2 ttl=127 time=0.721 ms
64 bytes from 11.0.2.81: icmp_seq=3 ttl=127 time=0.774 ms
64 bytes from 11.0.2.81: icmp_seq=4 ttl=127 time=1.52 ms
64 bytes from 11.0.2.81: icmp_seq=5 ttl=127 time=0.858 ms
^C
--- 11.0.2.81 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4130ms
rtt min/avg/max/mdev = 0.709/0.915/1.515/0.304 ms
```

Task4:

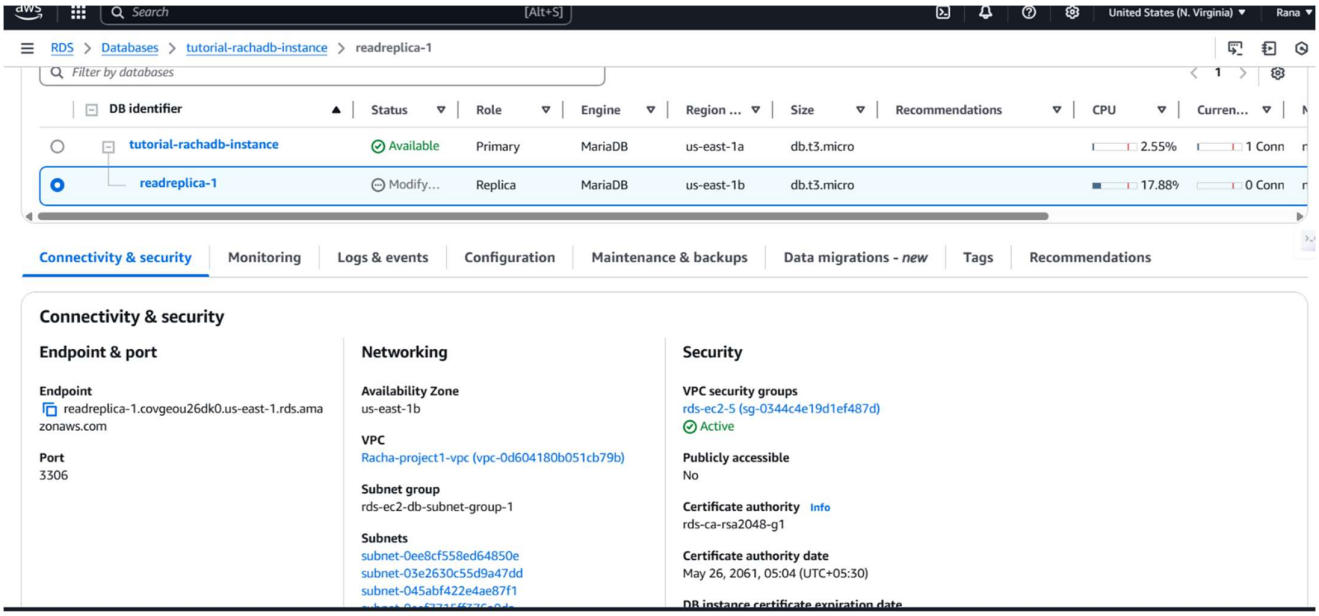
Auto scaling details:



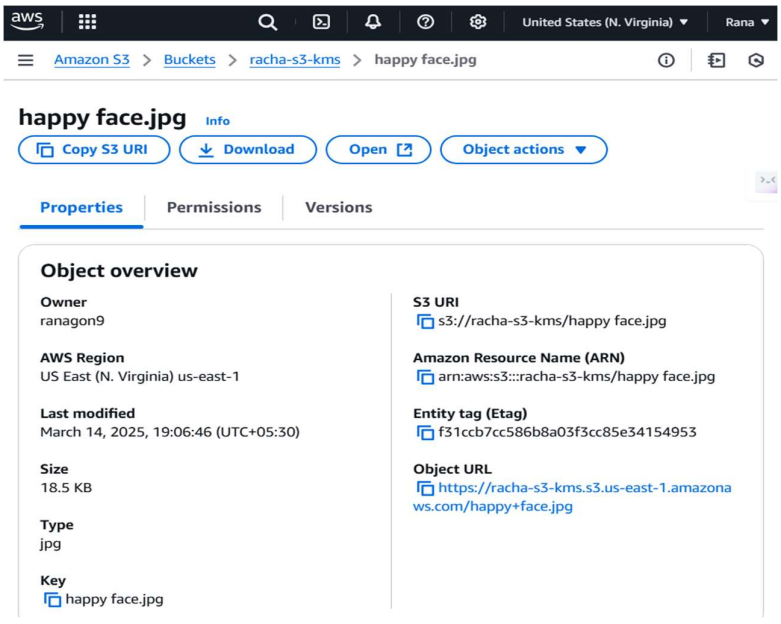


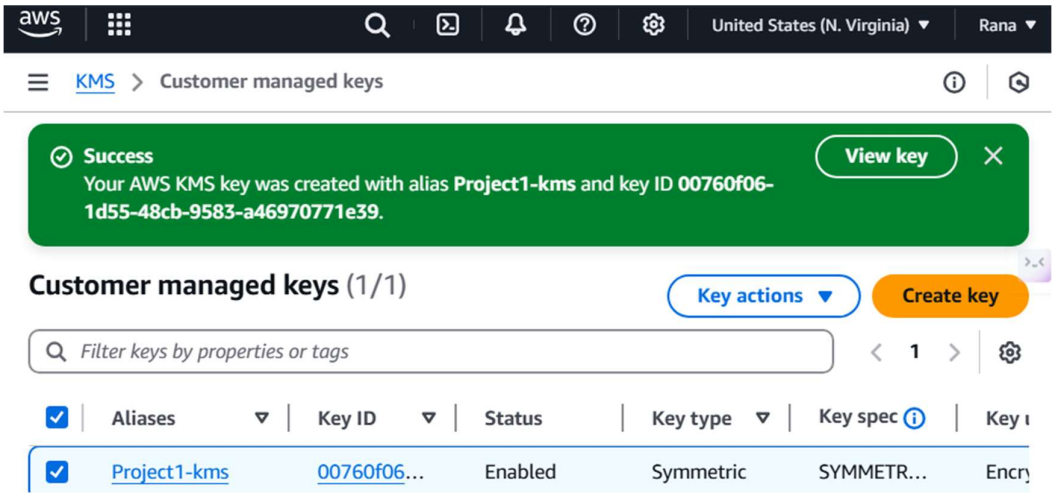
Task5:

Read Replica of RDS Database:

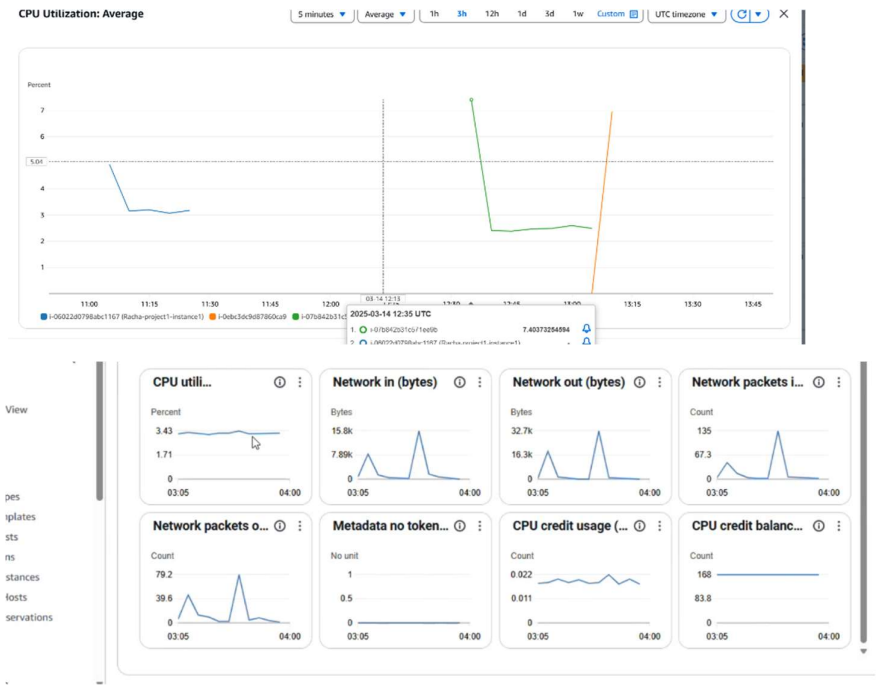


Task6: Encrypt the data on DB layer and S3 using KMS(Key Management Service)





Task7: Monitor the application with CloudWatch



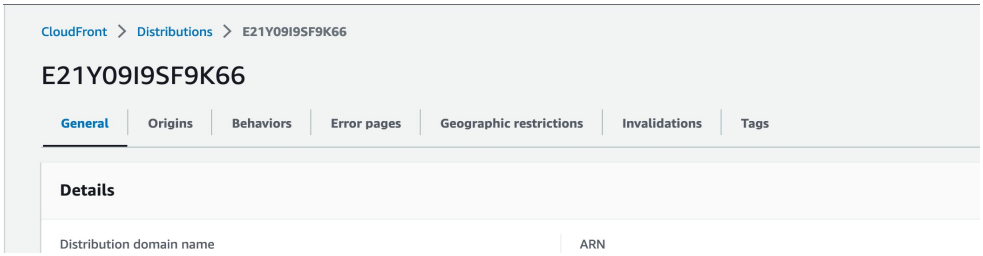
Task8: IAM Policies for Defining User Permissions in AWS

User1 has no access

User2 has Admin access

Task9: Access application using Cloudfront DNS

Cloudfront Distributions:



Webpage output: