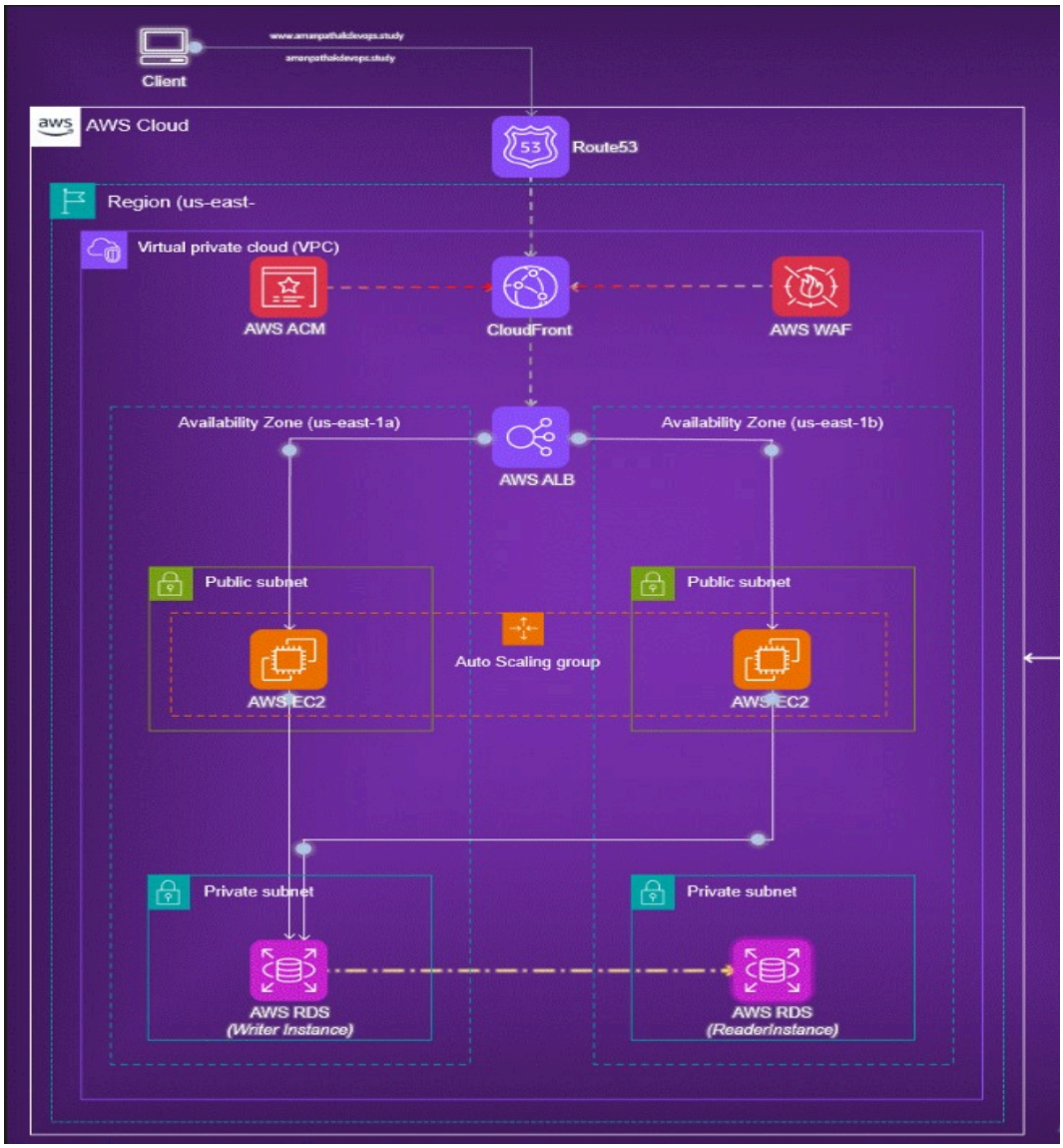
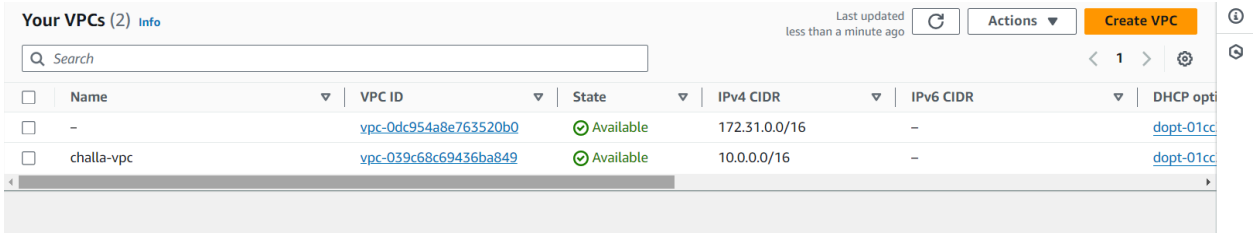
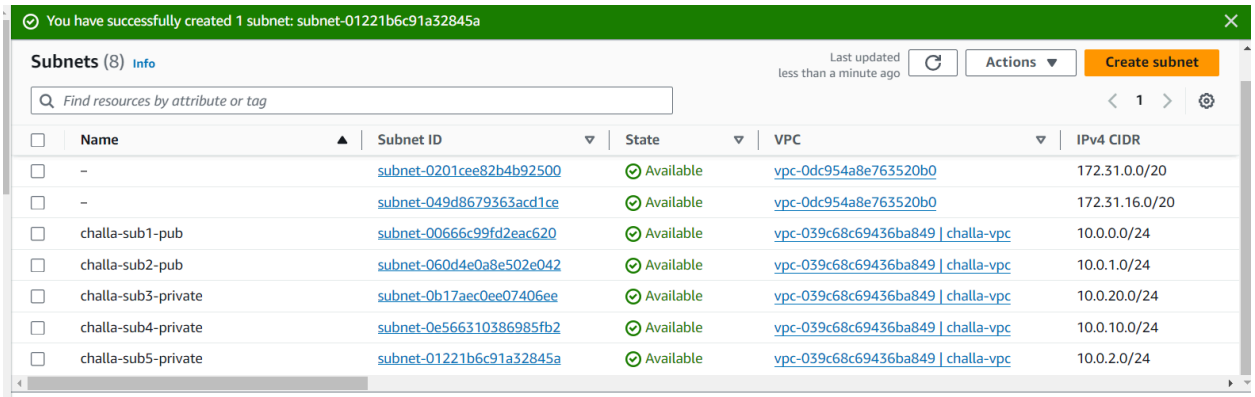
Creating 2 Tiered Architecture in AWS



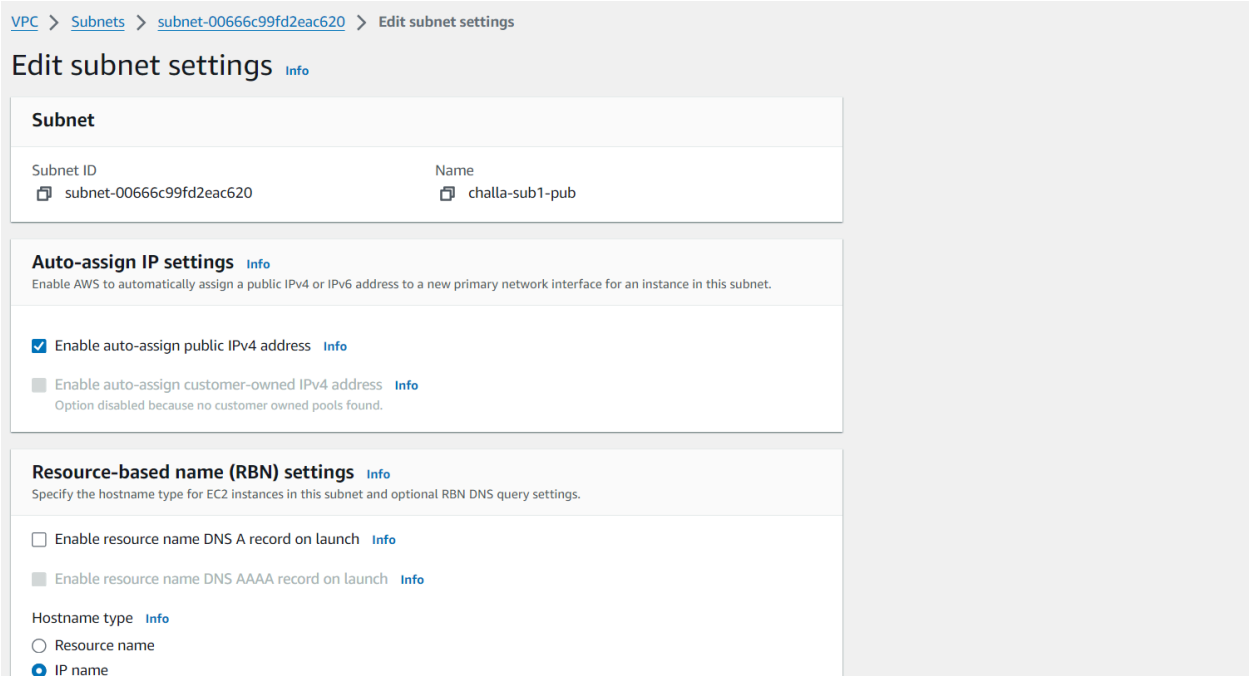
1) Create one vpc name=challa-vpc



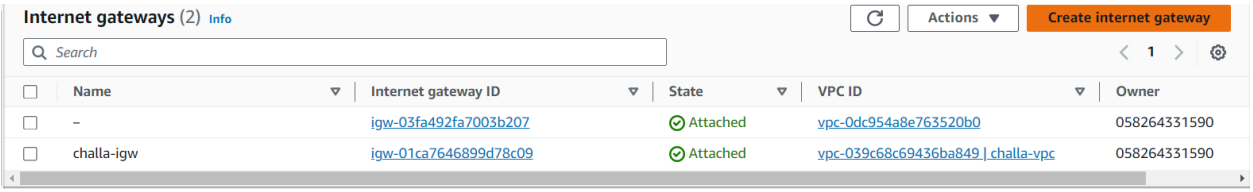
2) Crete 2 public subnets in 1a and 1b availability zone for ec2 server and 3 private subnets for rds in 1a and 1b and 1c availability zones



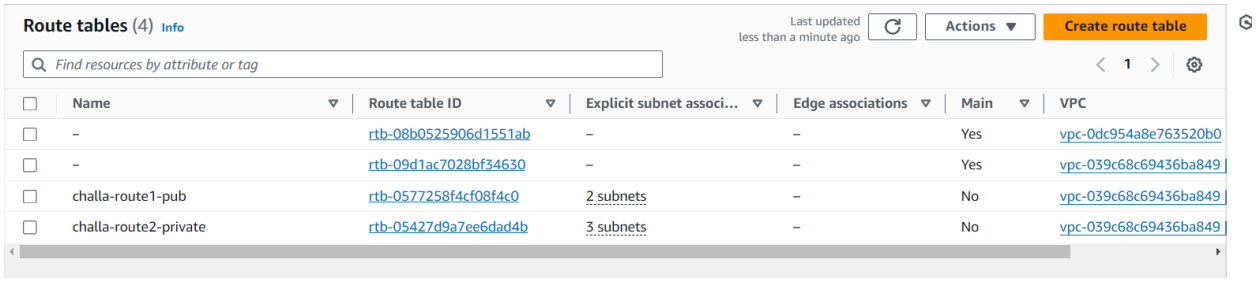
3) Now go to 2 public subnets click action edit subnet setting click on enable Enable auto-assign public IPv4 address



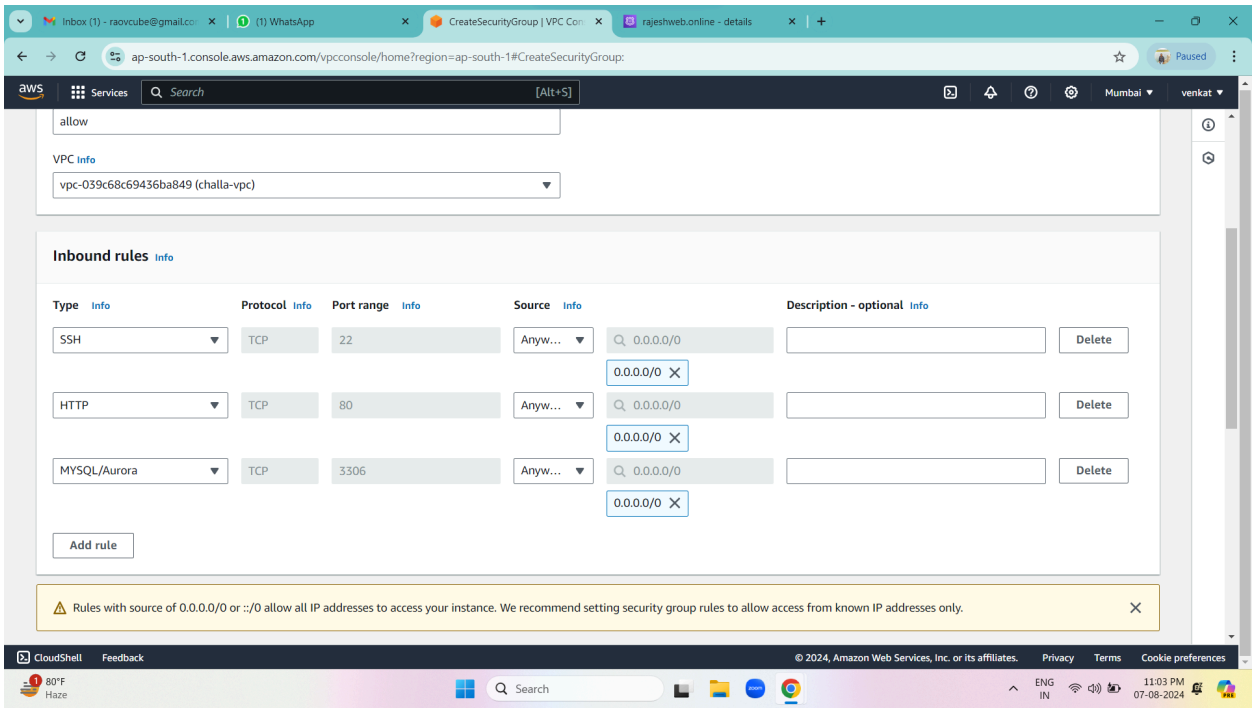
4) Crete challa-igw and attach to challa-vpc

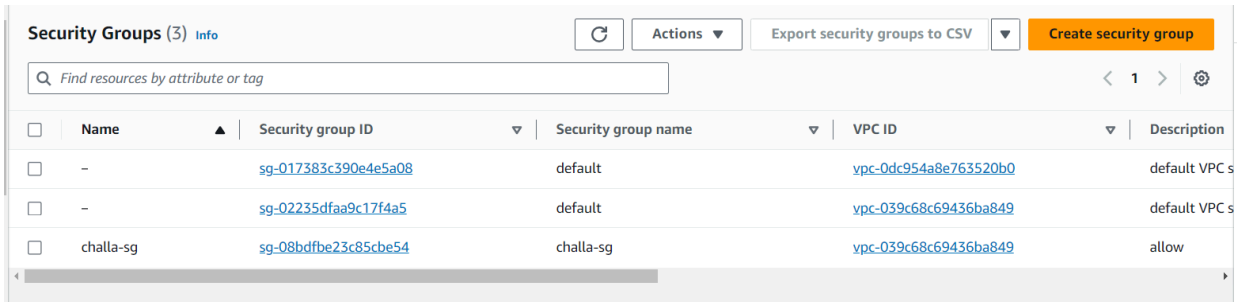


5) Now create 2 route tables one is for public subnets and another one is private subnets
6) And route igw and subnet association

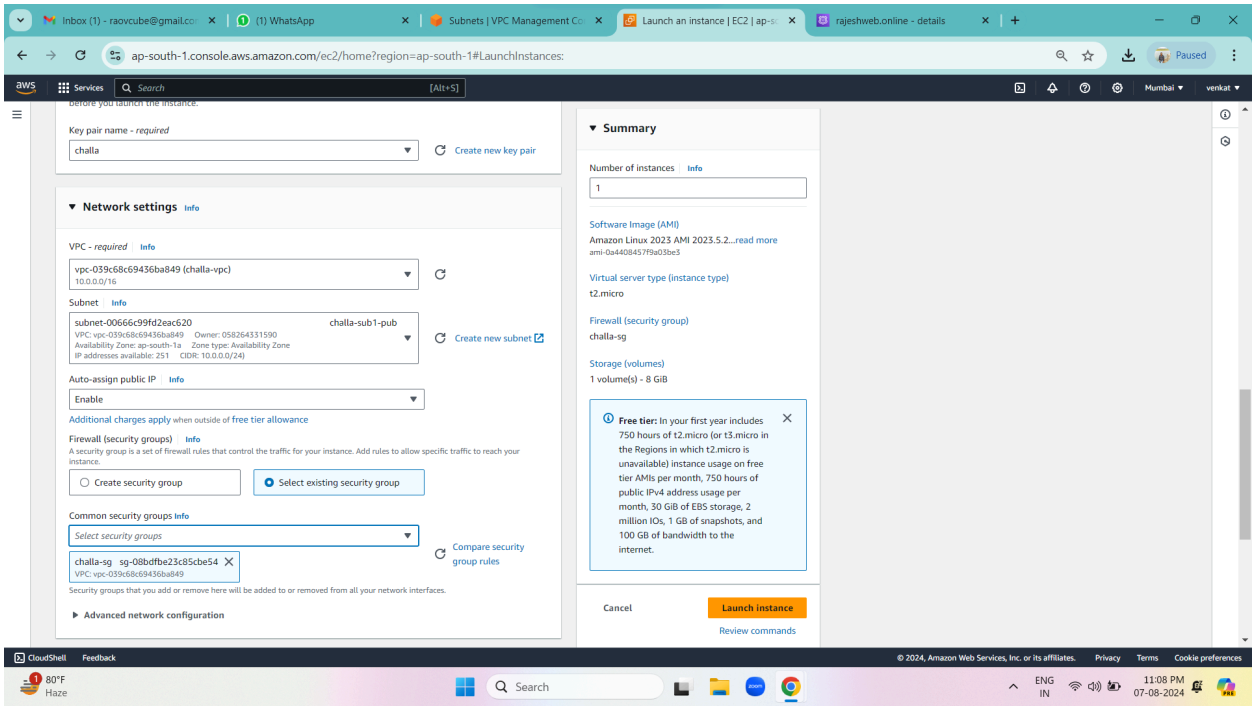


7) Create security group name =challa-sg allow ssh and http and mysql add port numbers 443, 80,3306,22

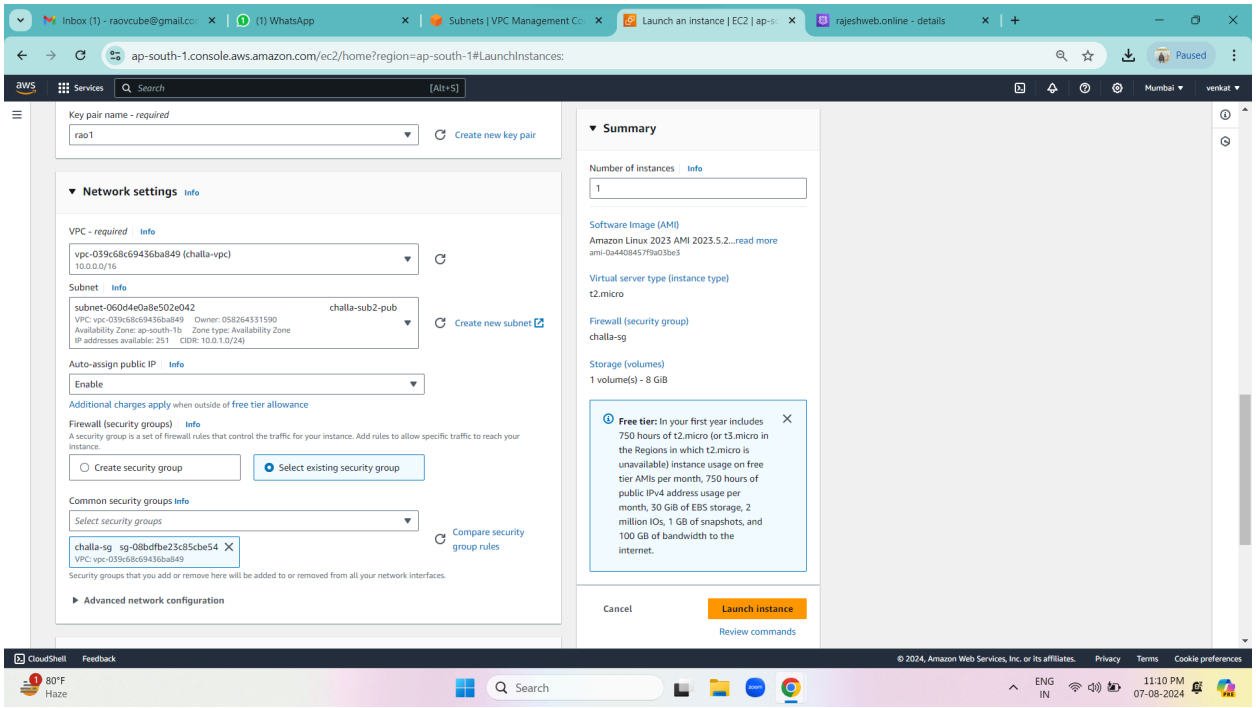


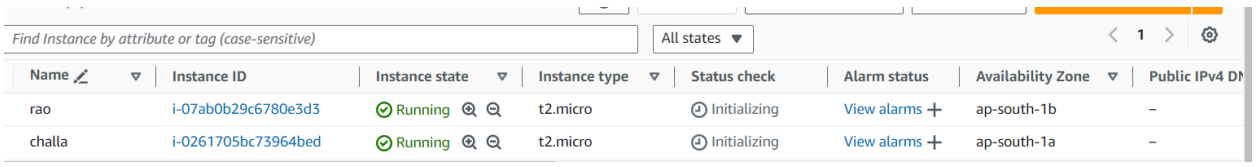


8) Now launch ec2 instance 2 using public subnet



9) Same way launch 2 instancename rao





10) Now connect challa and install nginx

11) Using #sudo -i

12) #yum update
13) #yum install nginx -y
14) #cd /usr/share/nginx/html
15) #rm -f index.html
16) #vi index.html (hi this is challa save and exit)
17) #systemctl restart nginx

```
[root@ip-10-0-0-138 html]# history
    1  yum update
    2  yum install nginx
    3  cd /usr/share/nginx/html/
    4  rm -f index.html
    5  vi index.html
    6  systemctl restart nginx
    7  clear
    8  history
[root@ip-10-0-0-138 html]#
```

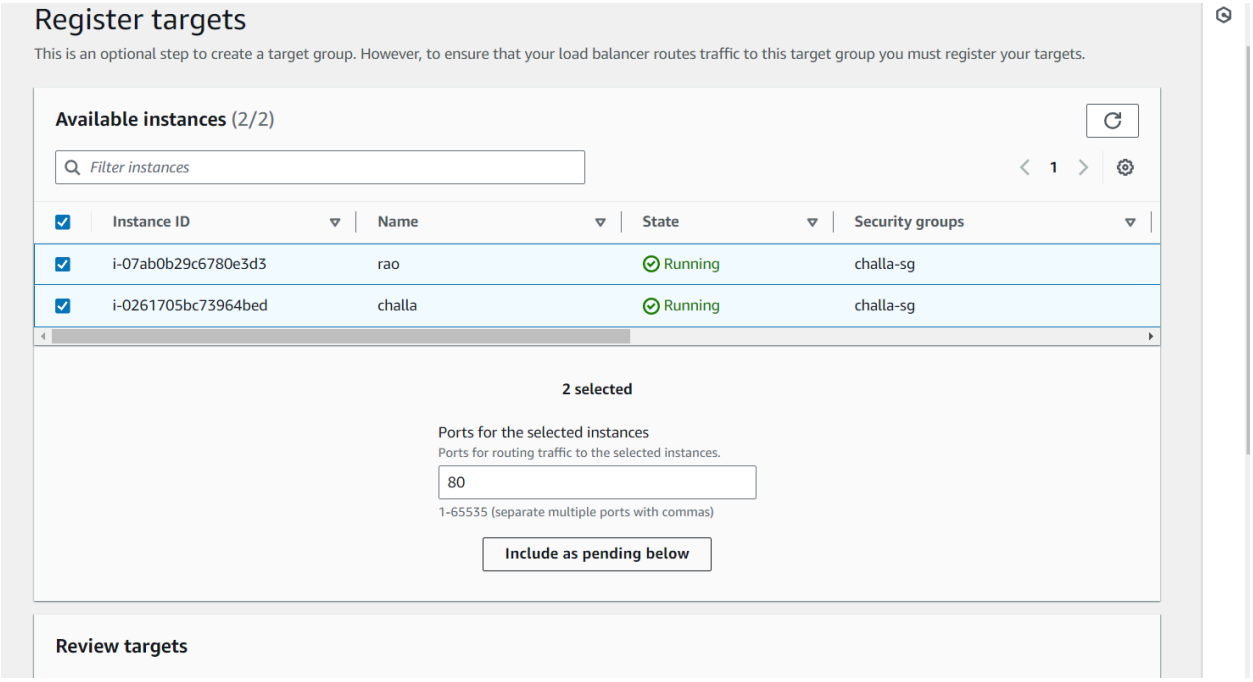
18) Now copy challa server public ip paste in google

hi this is challa

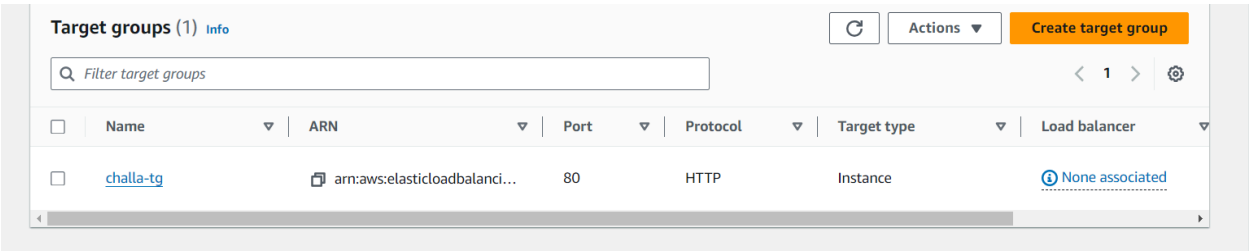
19) Now connect rao server do same above steps
20) And copy public ip paste in google

hi this is rao

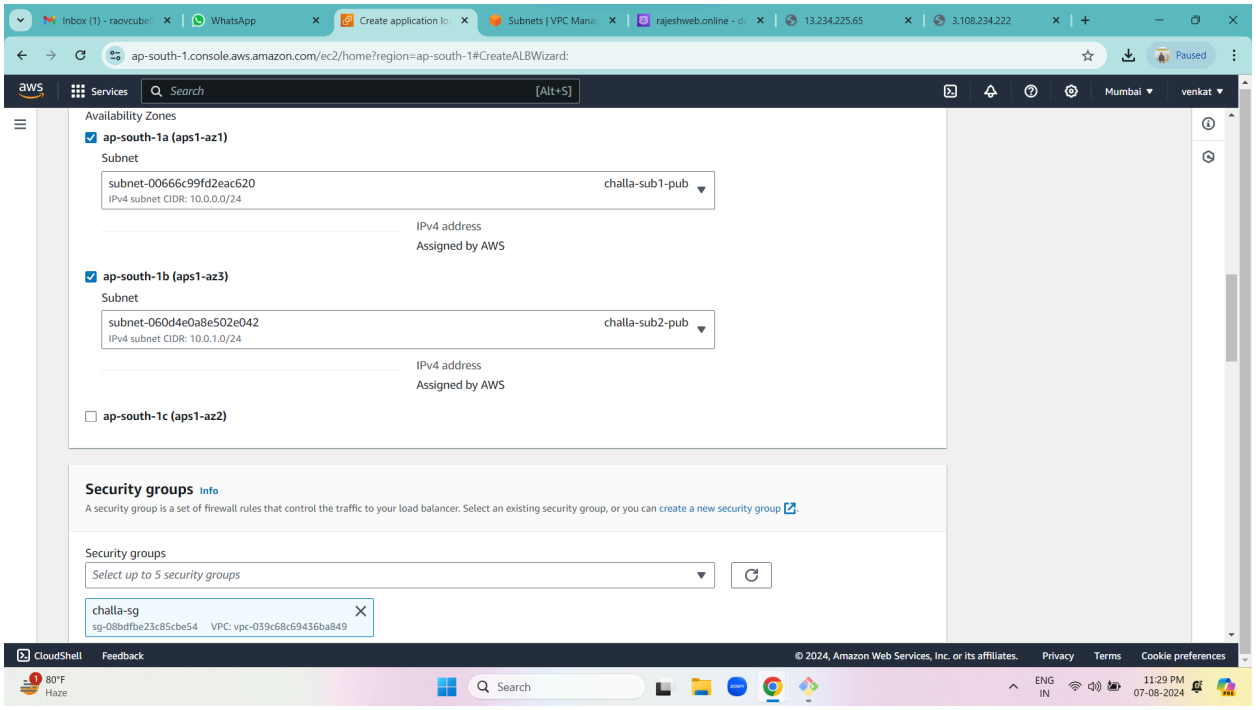
21) Now create target group name challa-tg
22) Chose instance target type= instances
23) Target group name = challa-tg
24) vpc= challa-vpc and click next



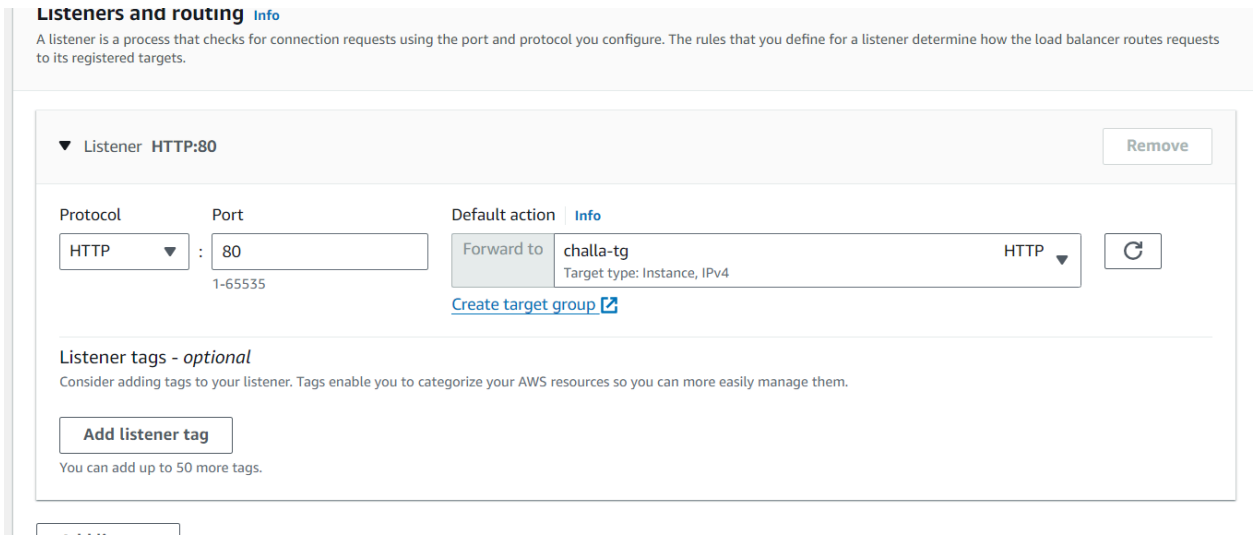
25) Select both rao and challa and click includes as pending below and click create target group



26) Now create load balancer andd Select application load balancer
27) Load balancer name= challa-loadbalancer
28) scheme= internet facing
29) vpc= challa-vpc
30) Select both availability zones public and select challa-sg

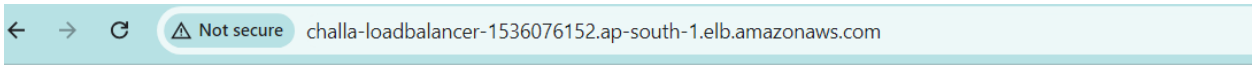


31) Listener = challa-tg



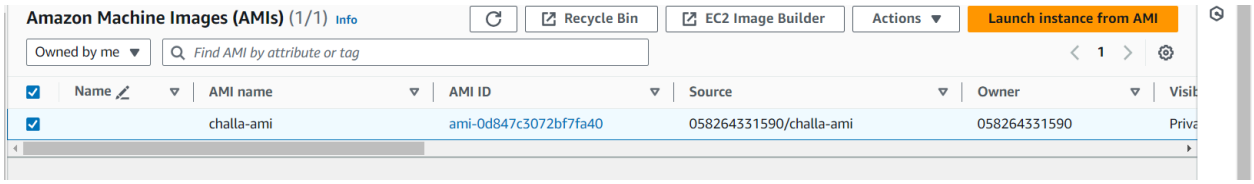
32) Copy challa load balancer dns name paste google do refresh check

hi this is rao

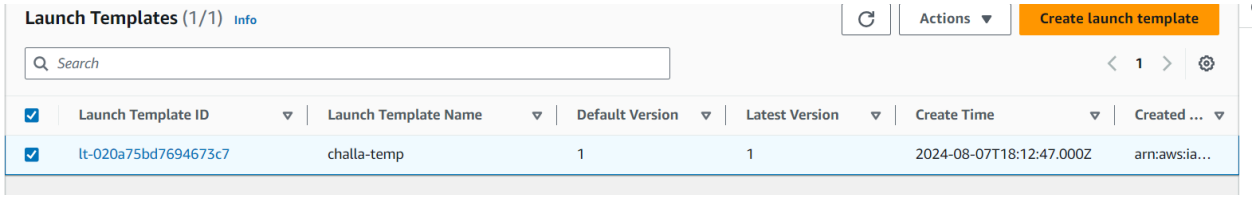


hi this is challa

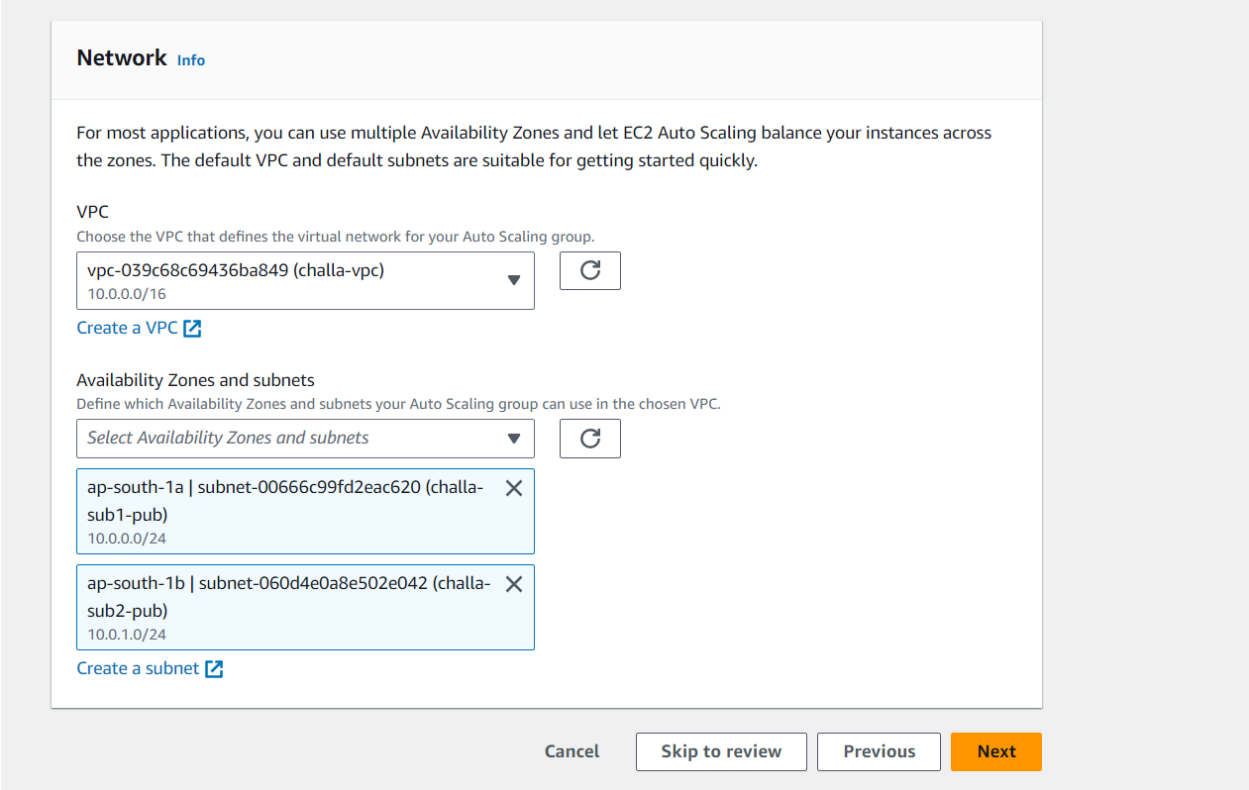
33) Now create ami using challa instance
34) Select challa instance click actions click create image and template and click create image
35) Image name= challa-ami
36) Enable no reboot
37) Create image



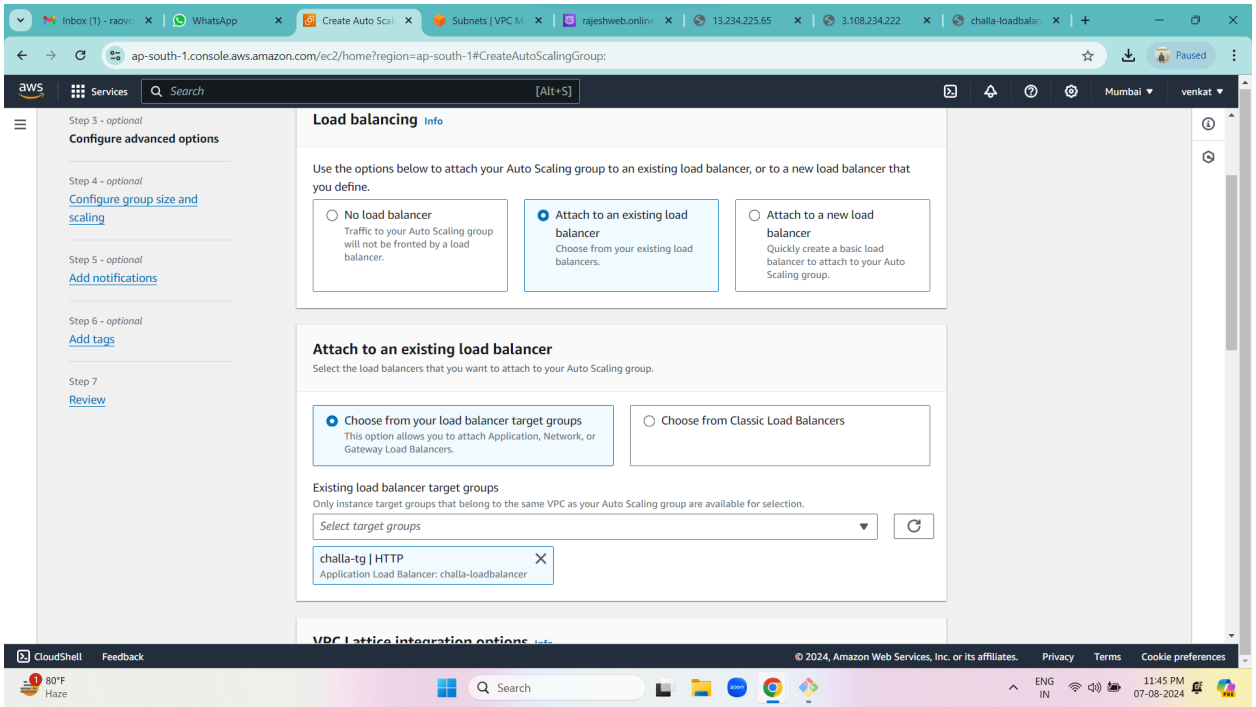
38) Now go to template create templet
39) Templet name= challa-temp
40) ami= challa-ami
41) Instance type =t2.micro
42) keypair= challa
43) Security group =challa-sg
44) Click create launch templet

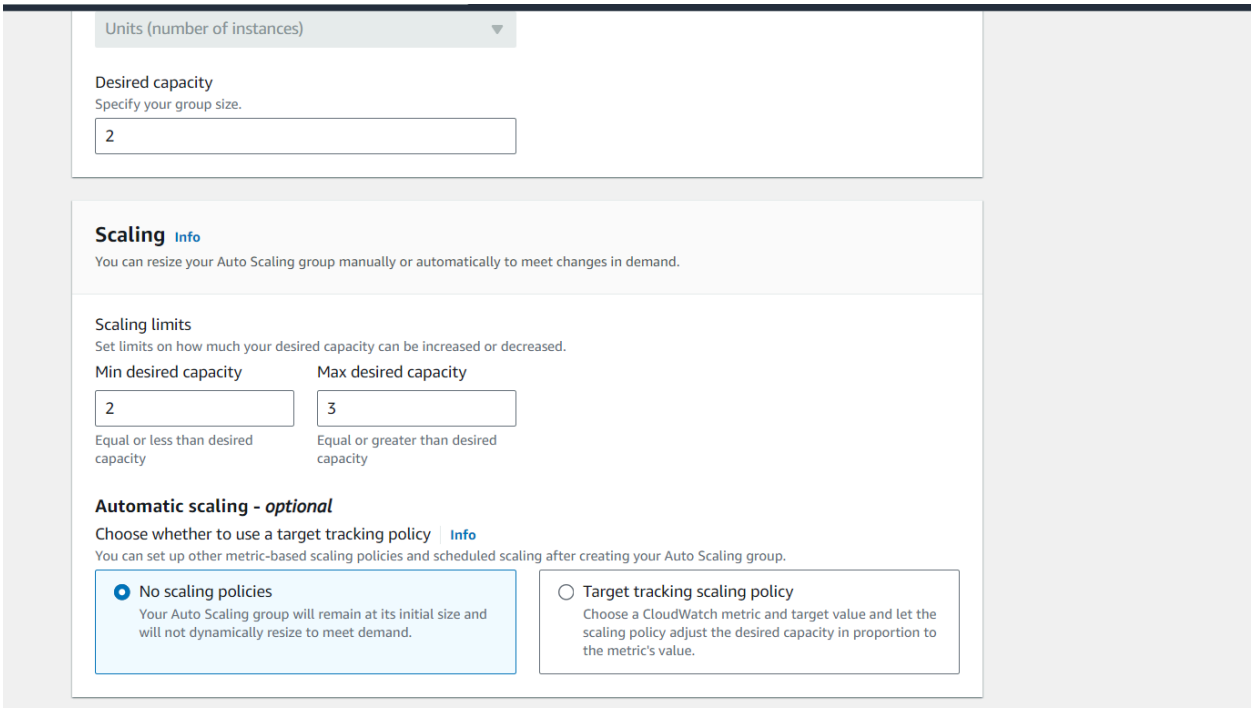


45) Now create auto scaling
46) name= challa-autosg
47) templet= challa-temp
48) Select vpc and availability zones

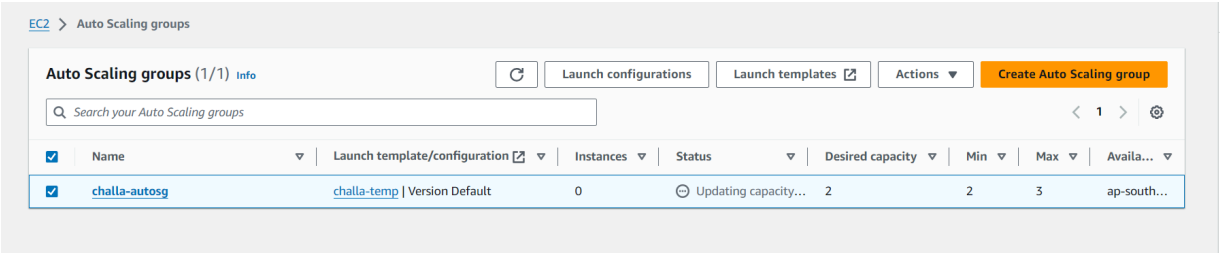


49) Select attach to existing load balancer and select choose from your loadbalancer target group select challa-tg

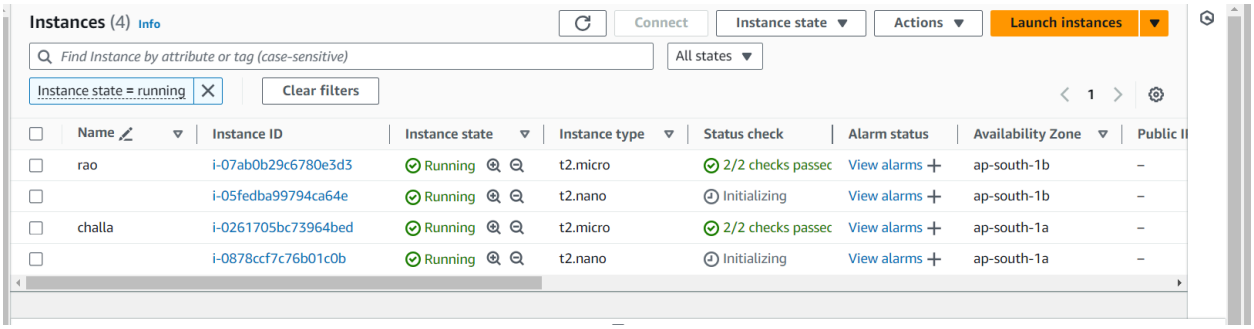




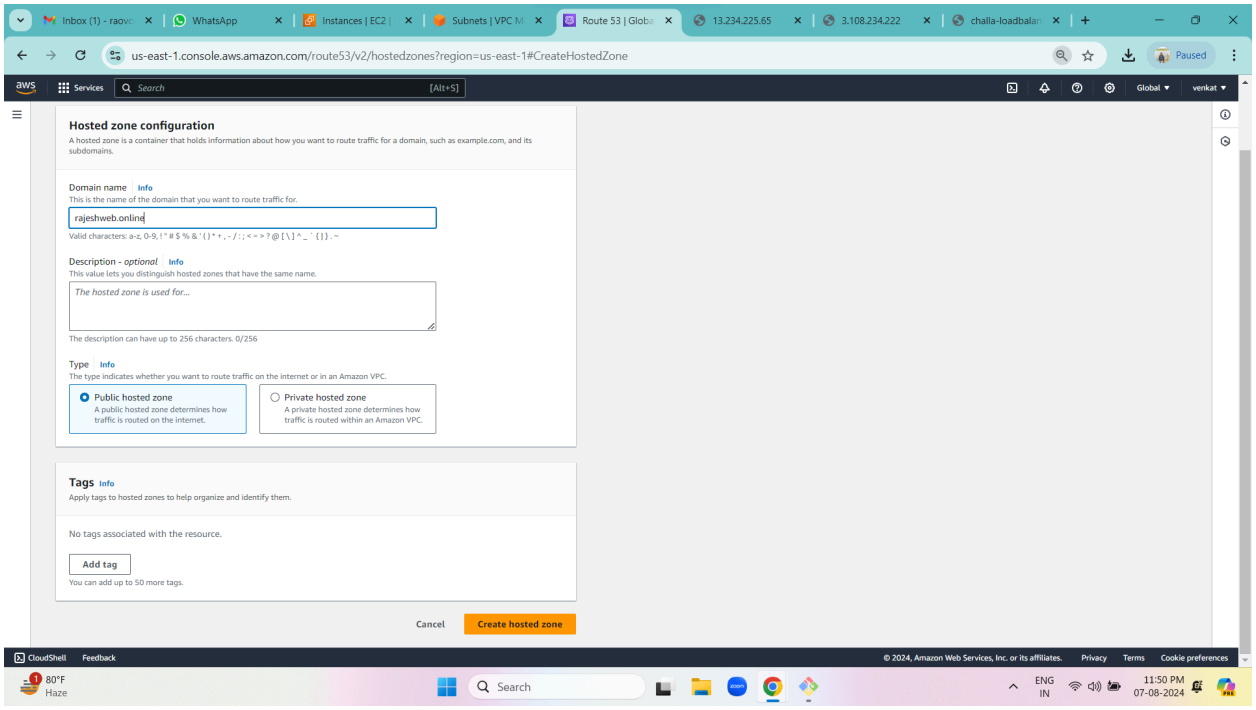
50) Desired capacity =2
51) Min desired capacity =2
52) Max desired capacity =3
53) Select no auto scaling policy click create autoscaling group

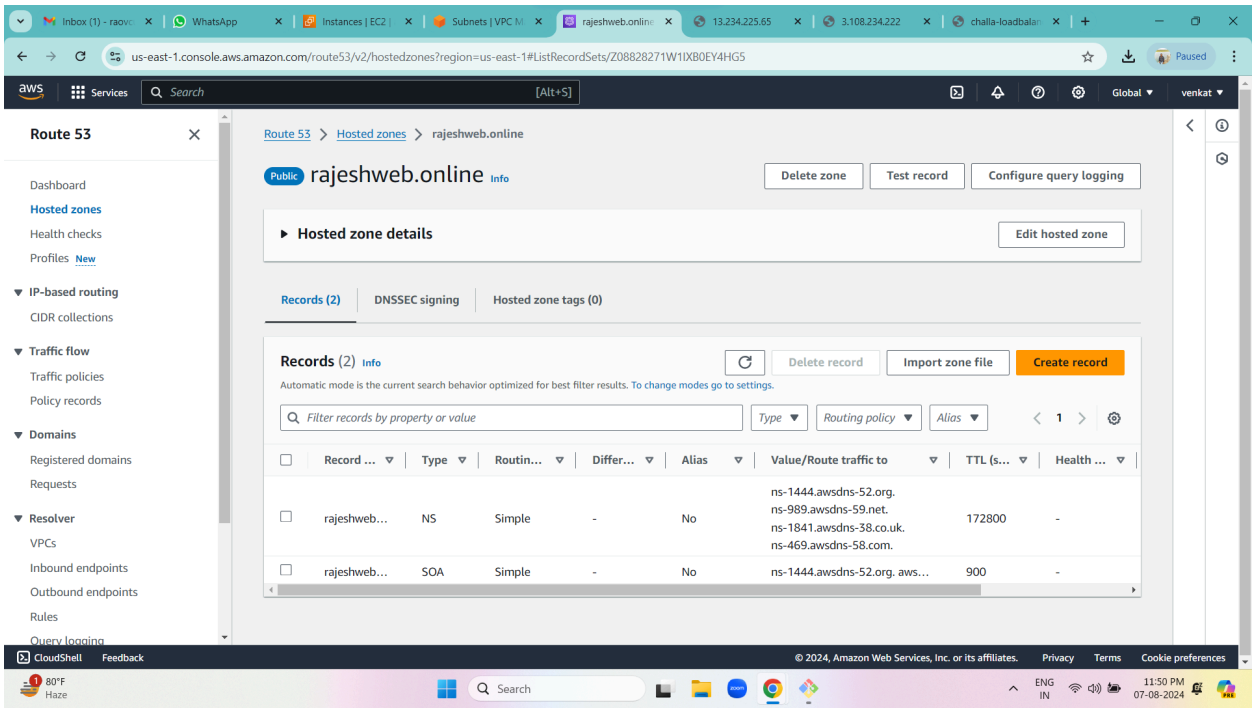


54) Go ec2 instances and check new instances create automatically

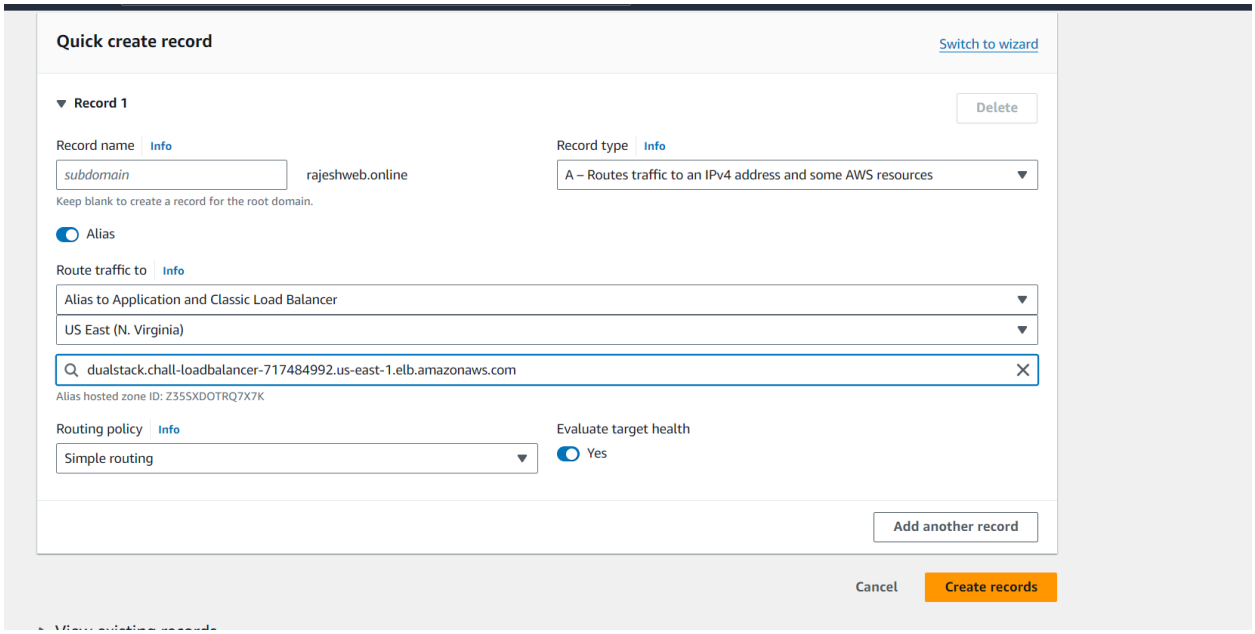


55) Now go to route 53 create hosted zone
56) Domain name = give your domain name ex= rajeshweb.online
57) Select public hosted zones and create hosted zone





58) Go to go-daddy and update name servers there
59) Now go to rajeshweb.online next create record turn on alias
60) Route traffic = alias to application load balancer
61) Region = us-east (N -verginia)
62) Select challa load balancer
63) Click crete record



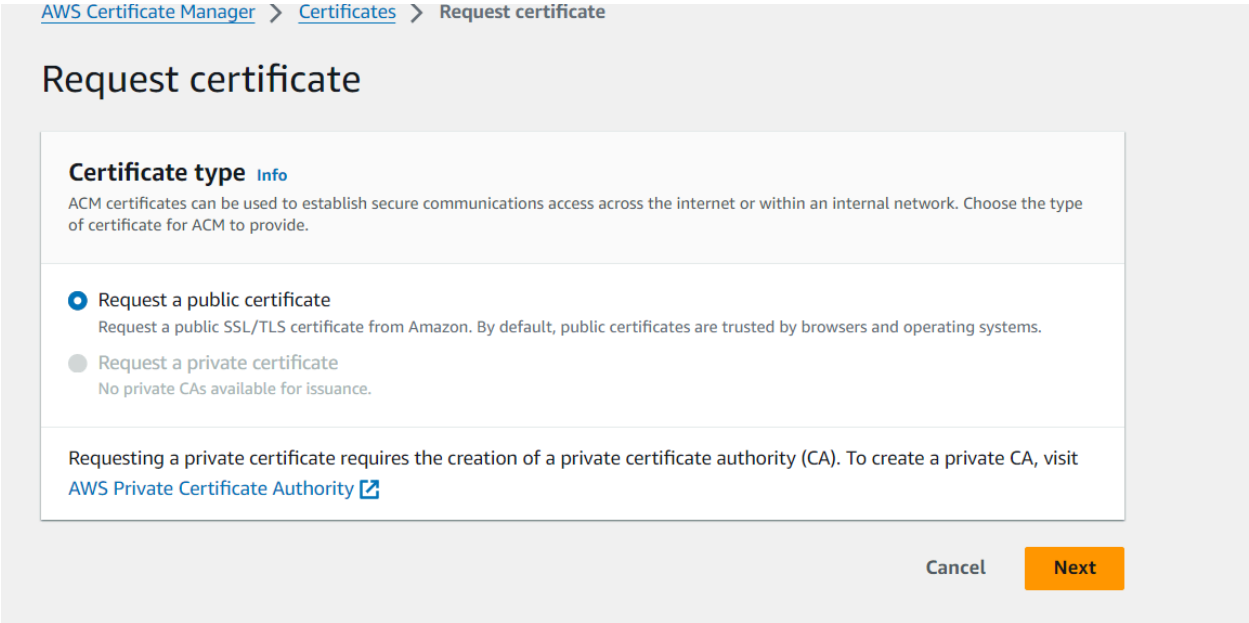
64) Now go to google search rejeshweb.online

hi this is challa

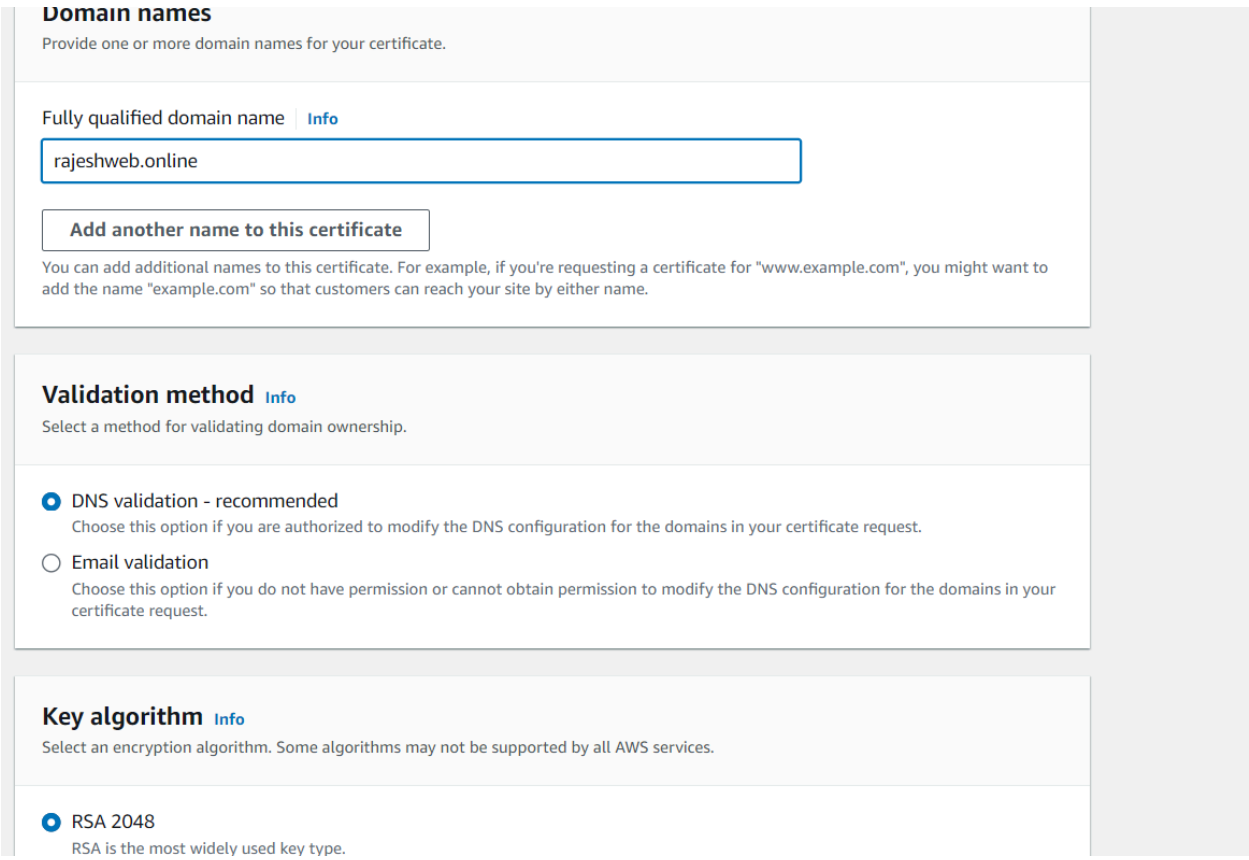
hi this is rao

65) Now go acm

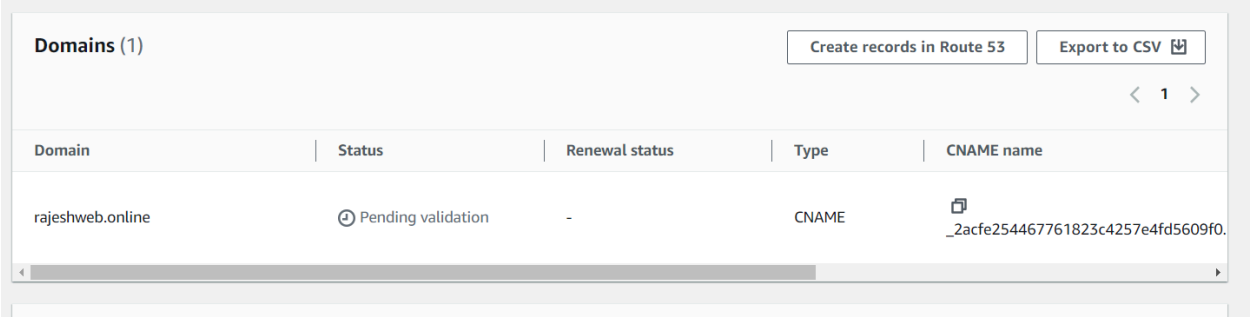
66) Select region and click request certificate

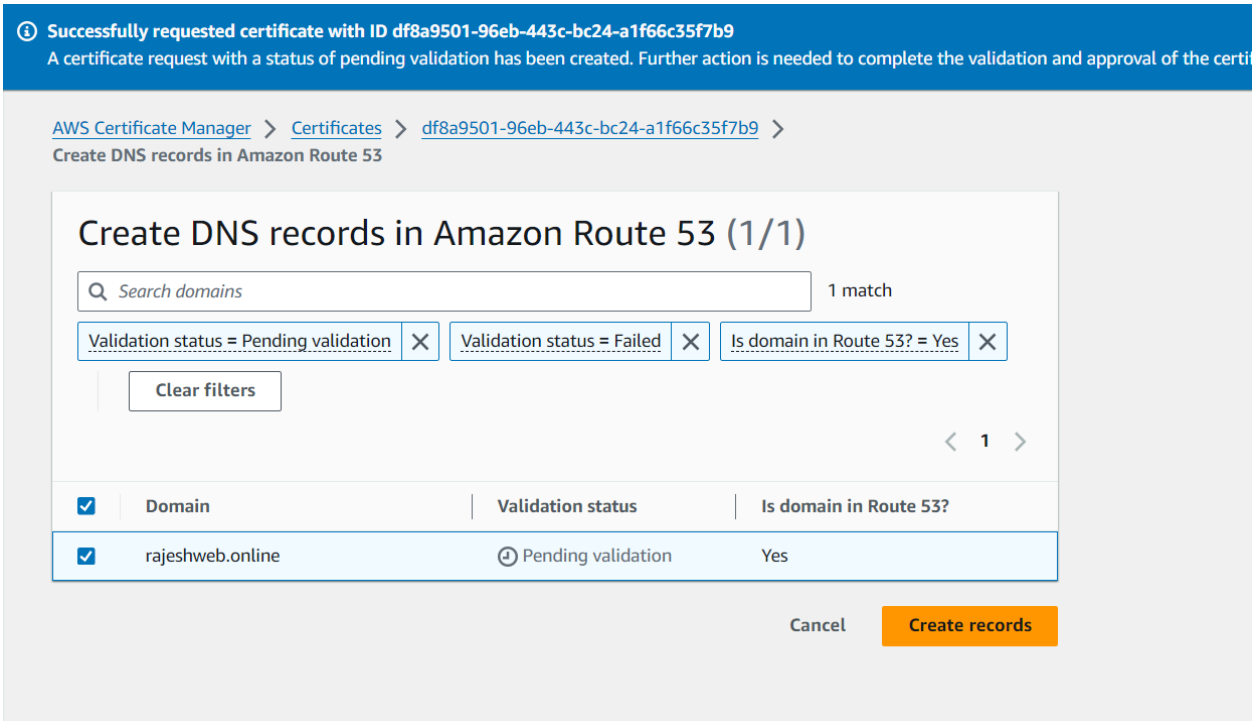


67) Give fully qualified domain name= rajeshweb.online
68) Validation method = DNs validation

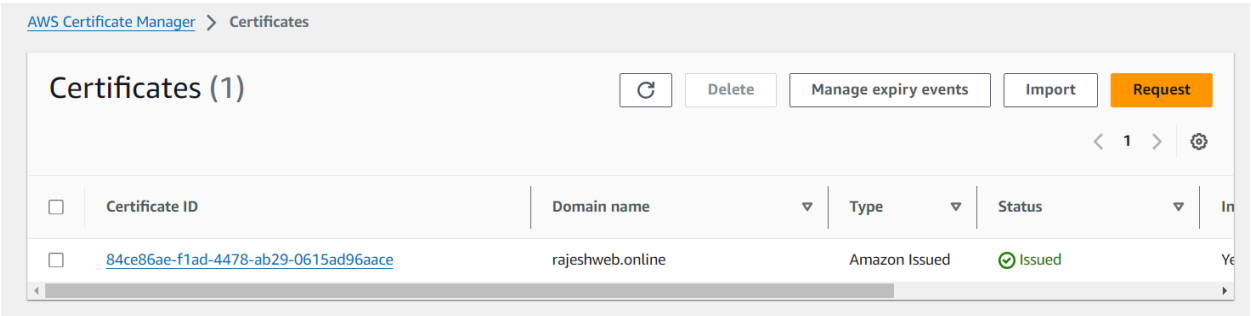


69) Click on request
70) Click create record in route 53

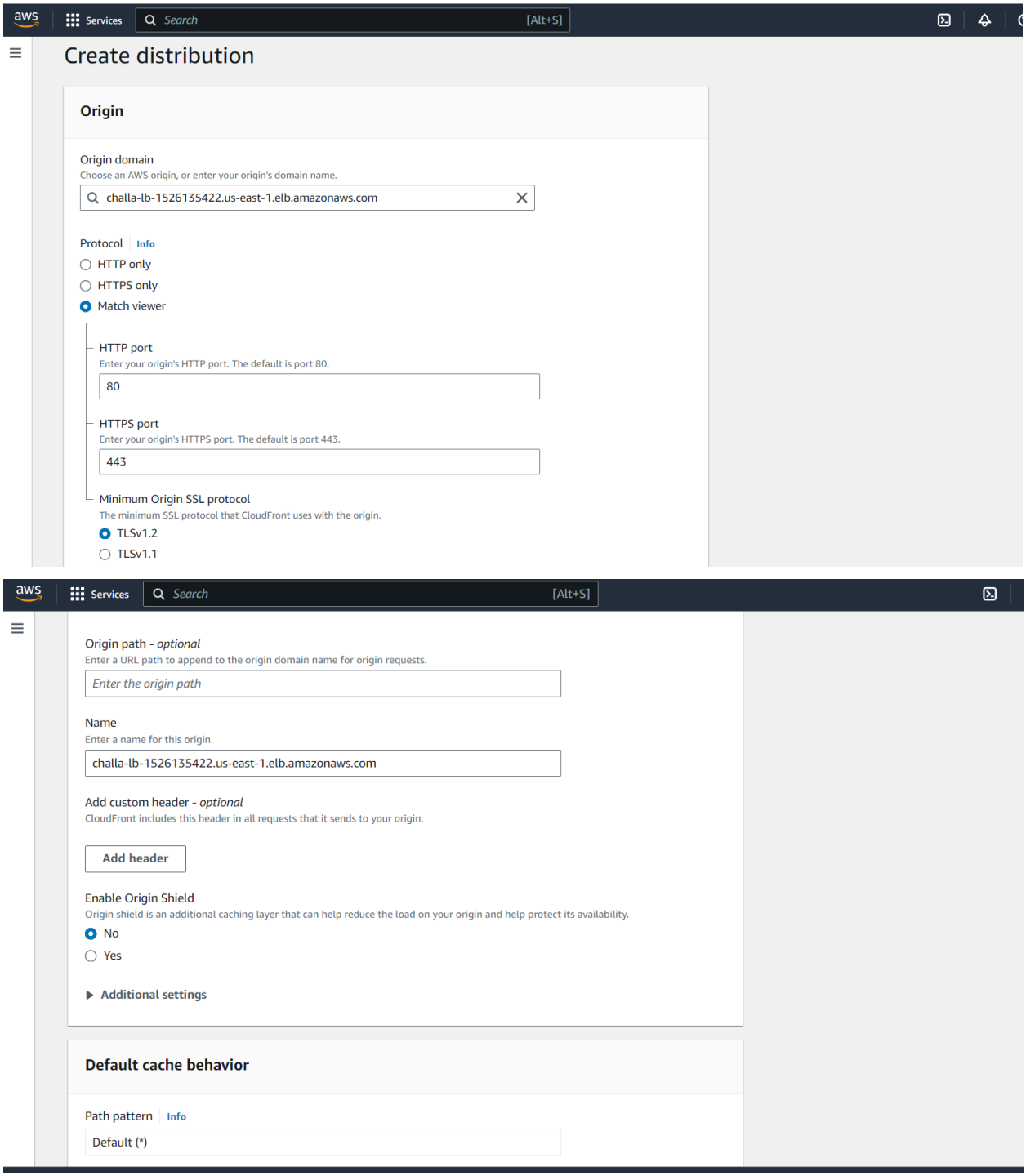




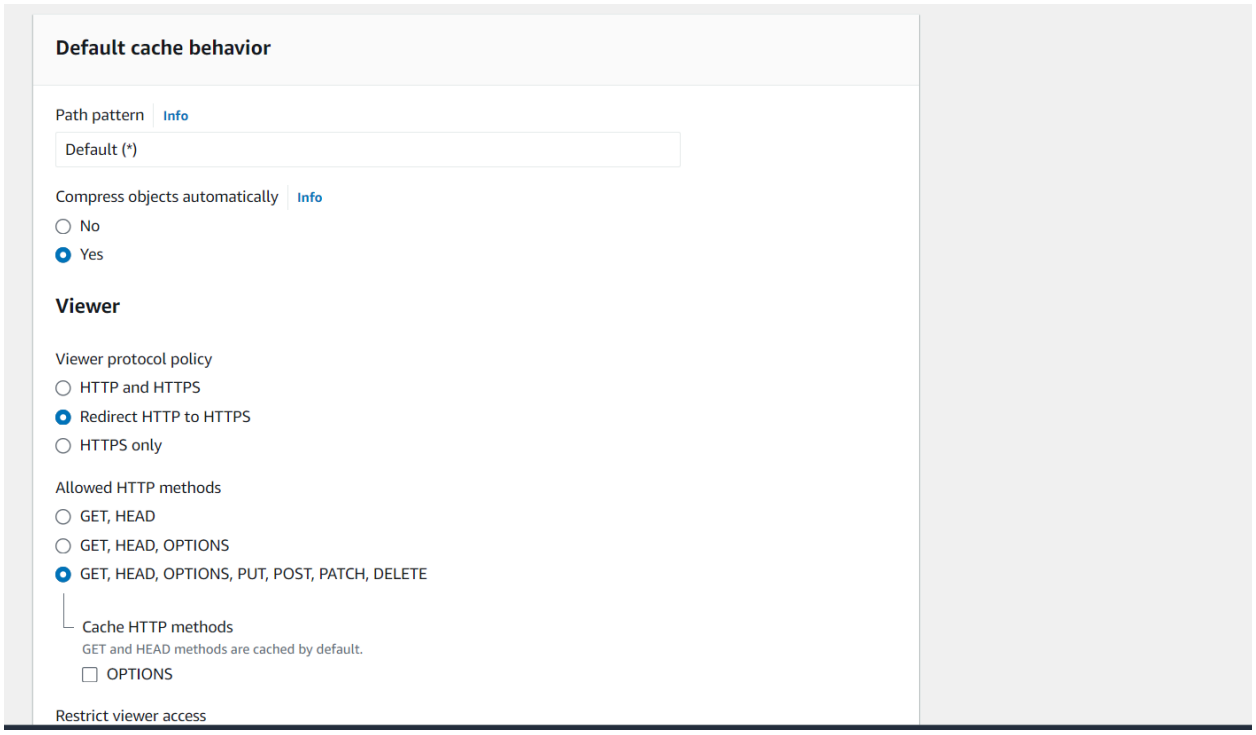
71) Select rajeshweb.online create record it update CNAME in route 53 record check there

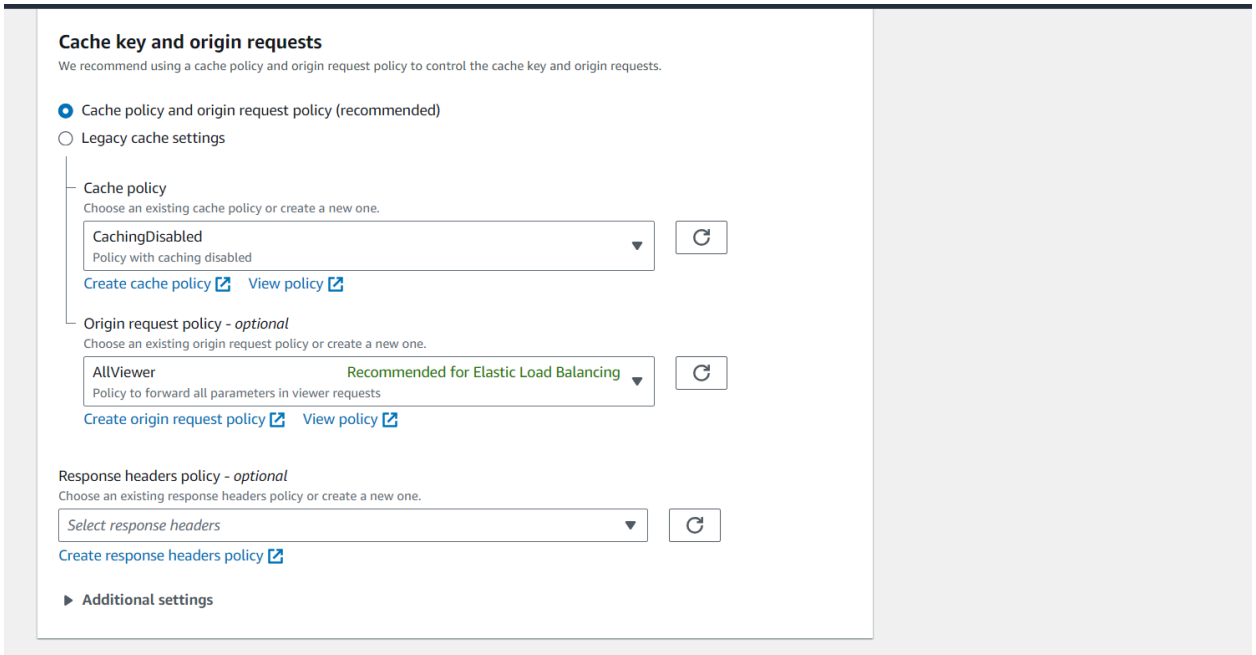


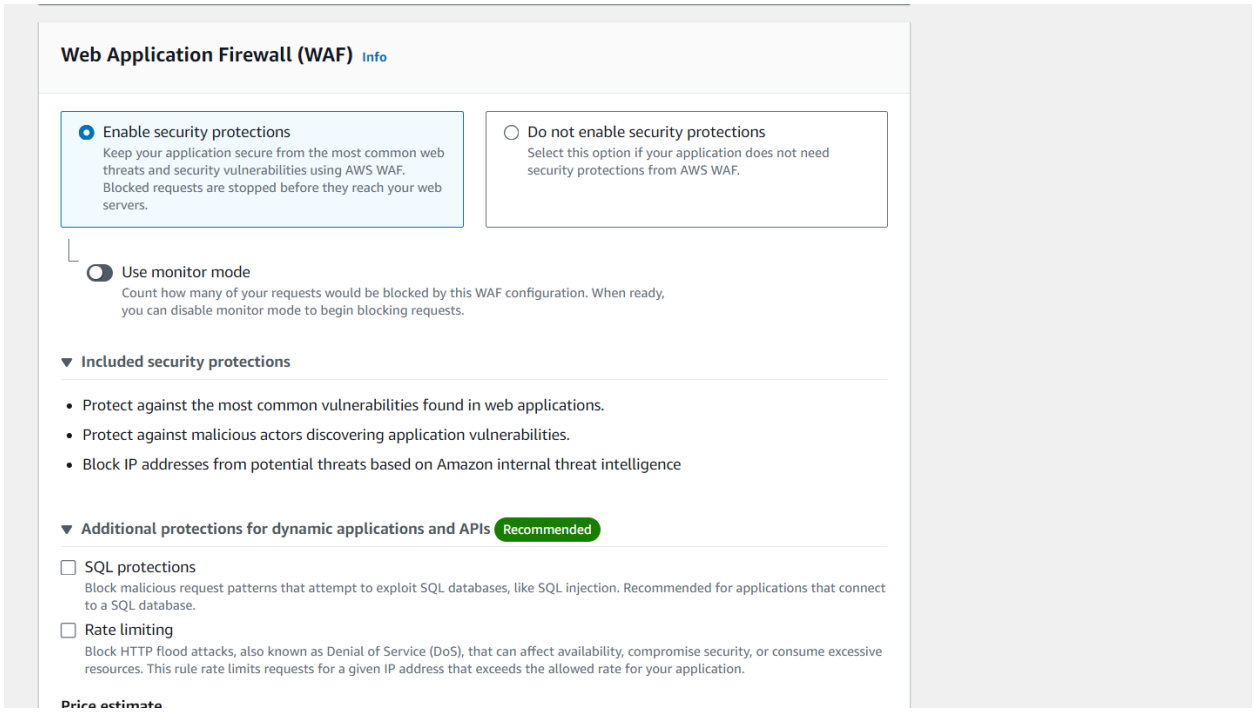
72) Now go to cloud front create a distributions
73) Origin domain = challa-lb (select load balancer)
74) protocol= select http only (mach viewer)



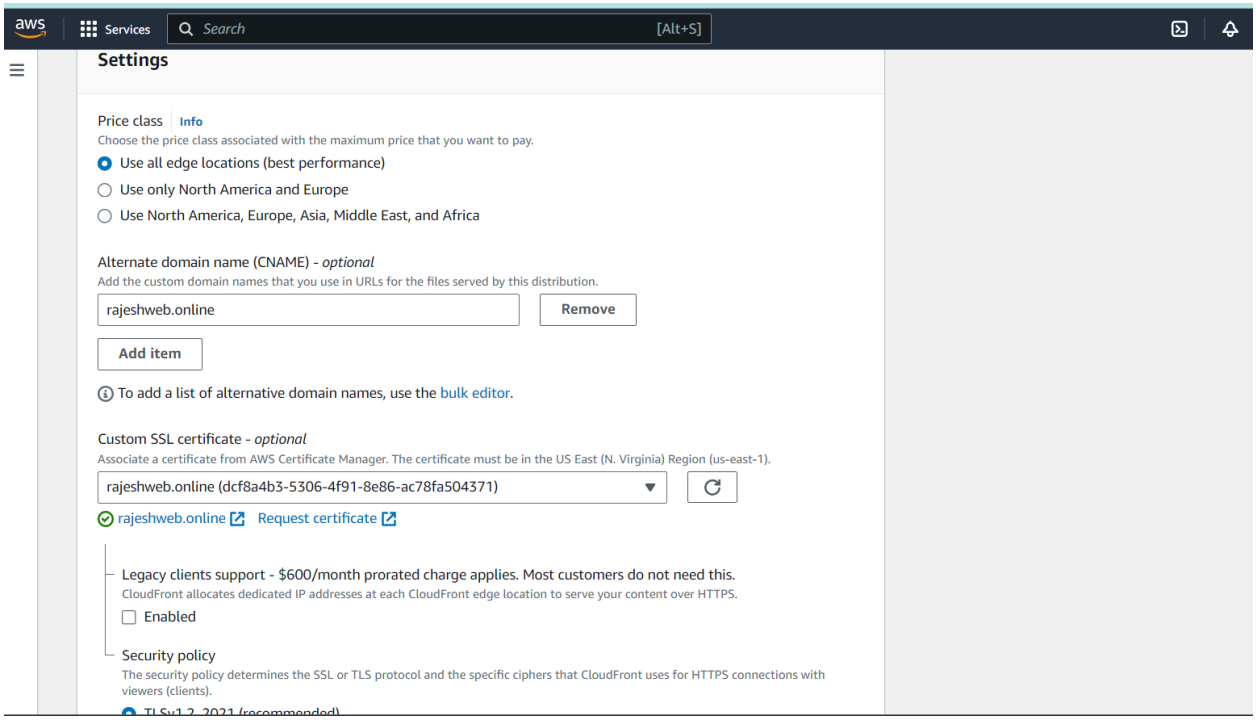
75) Viewer protocol policy = redirect toHTTP to HTTPS

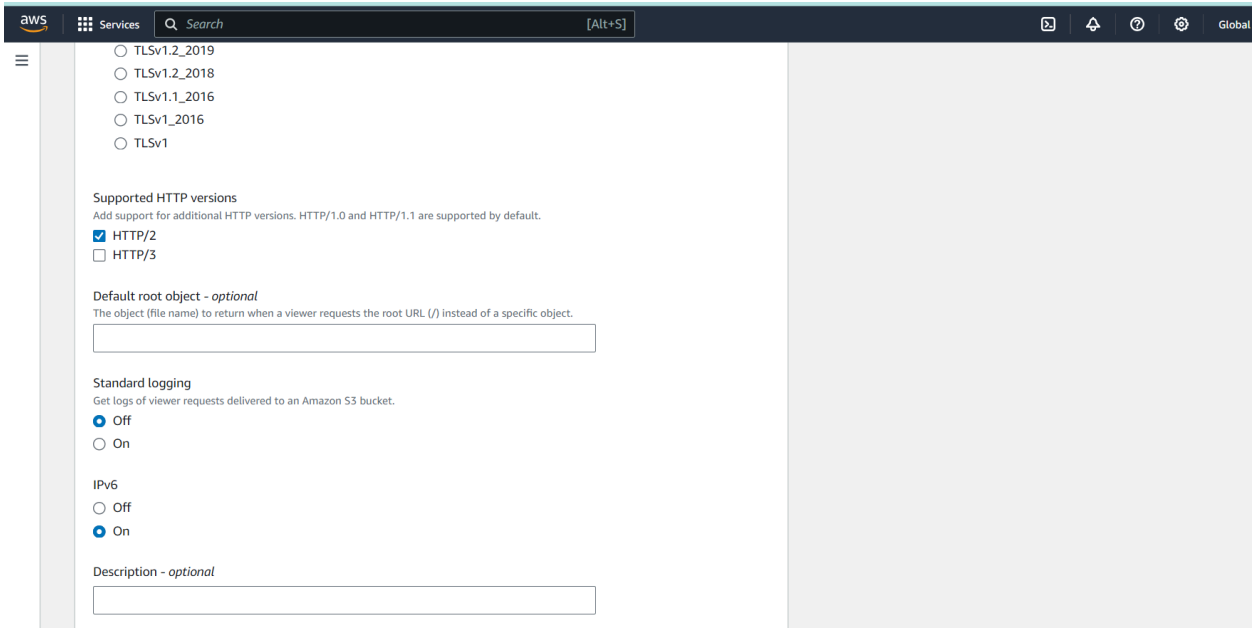

Default cache behavior
Path pattern Info
Default (*)
Compress objects automatically Info
No
Yes
Viewer
Viewer protocol policy
HTTP and HTTPS
Redirect HTTP to HTTPS
HTTPS only
Allowed HTTP methods
GET, HEAD
GET, HEAD, OPTIONS
GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE
Cache HTTP methods
GET and HEAD methods are cached by default.
OPTIONS
Restrict viewer access

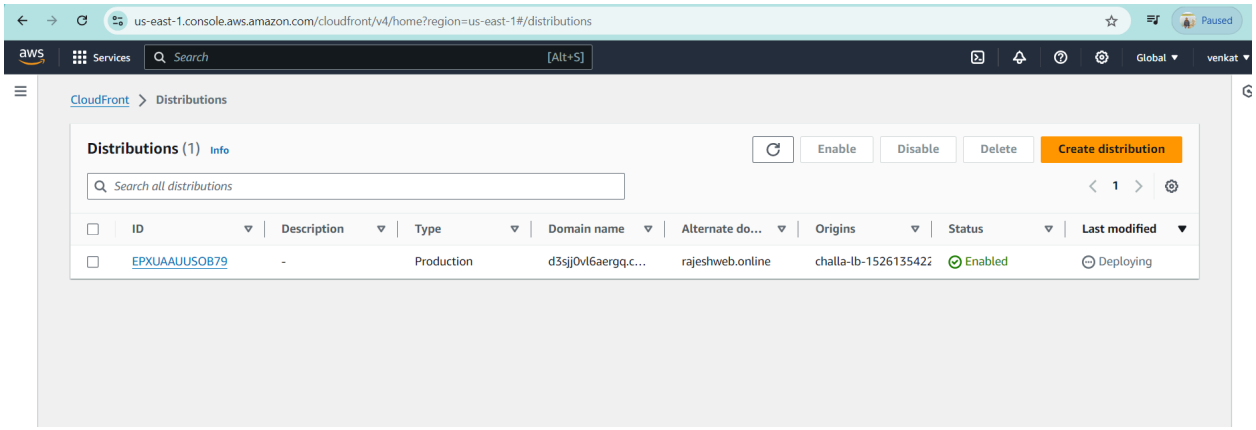

Cache key and origin requests
We recommend using a cache policy and origin request policy to control the cache key and origin requests.
Cache policy and origin request policy (recommended)
Legacy cache settings
Cache policy
Choose an existing cache policy or create a new one.
CachingDisabled
Policy with caching disabled
Create cache policy
View policy
Origin request policy - optional
Choose an existing origin request policy or create a new one.
AllViewer
Recommended for Elastic Load Balancing
Policy to forward all parameters in viewer requests
Create origin request policy
View policy
Response headers policy - optional
Choose an existing response headers policy or create a new one.
Select response headers
Create response headers policy
Additional settings



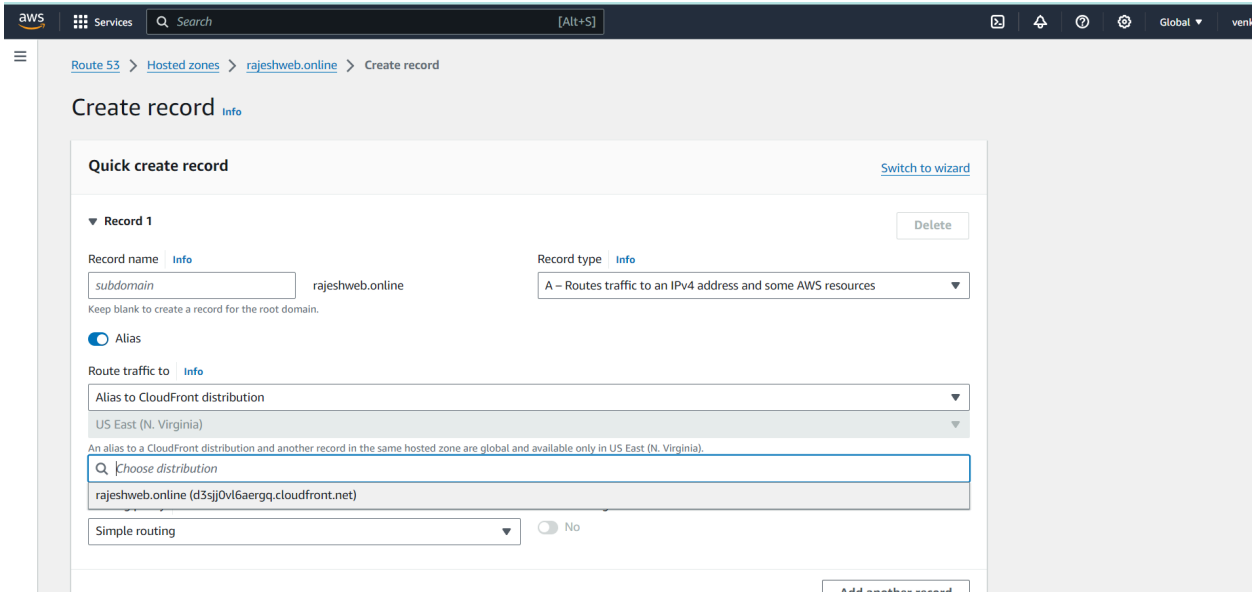
76) Settings use all edge locations
77) Alternate domain name click add item = rajeshweb.online
78) Custom ssl certificate = Select your acm certificate click create deploying







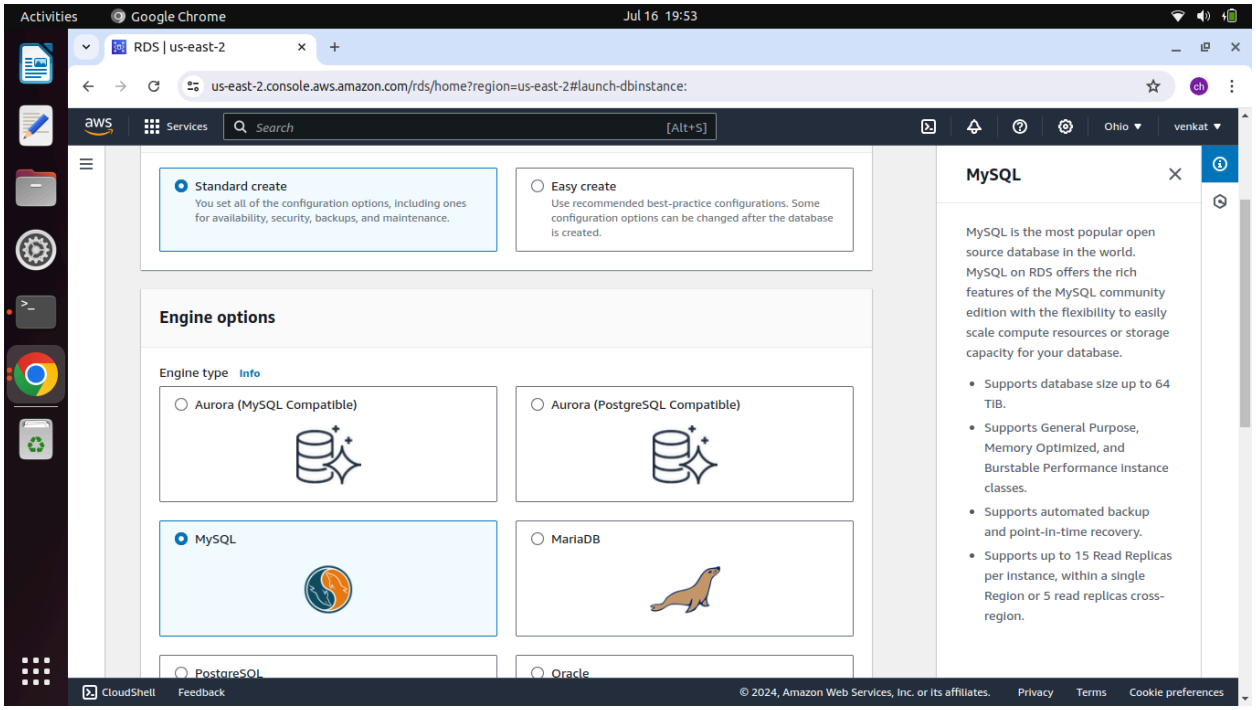
79) Now go to route 53 and delete load balancer record create new record
80) Turnon alias
81) Select cloudfront distribution
82) Select cloudfront name



83) Click create record
84) Now go to search with your domain now it show secure way

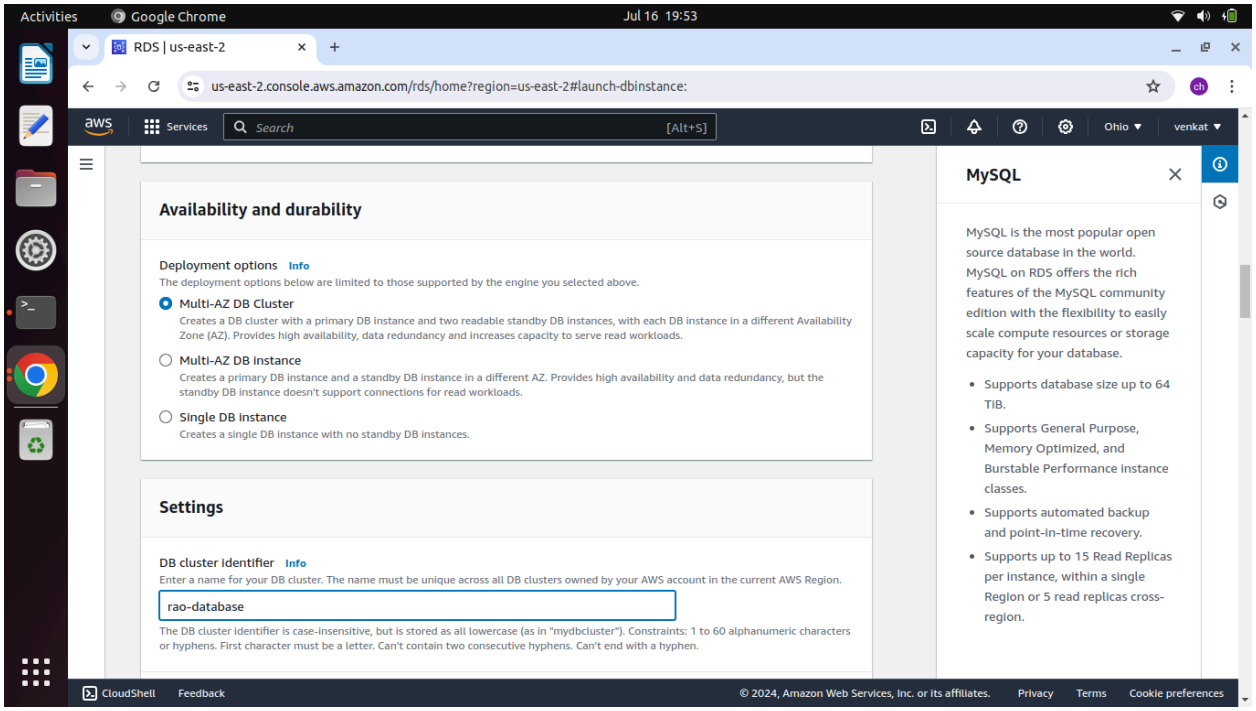
hi this is rao

85) Now go to rds and crete rds
86) Now go to databases
87) Select standard create select MY SQL



88) Select multi-az-db cluster

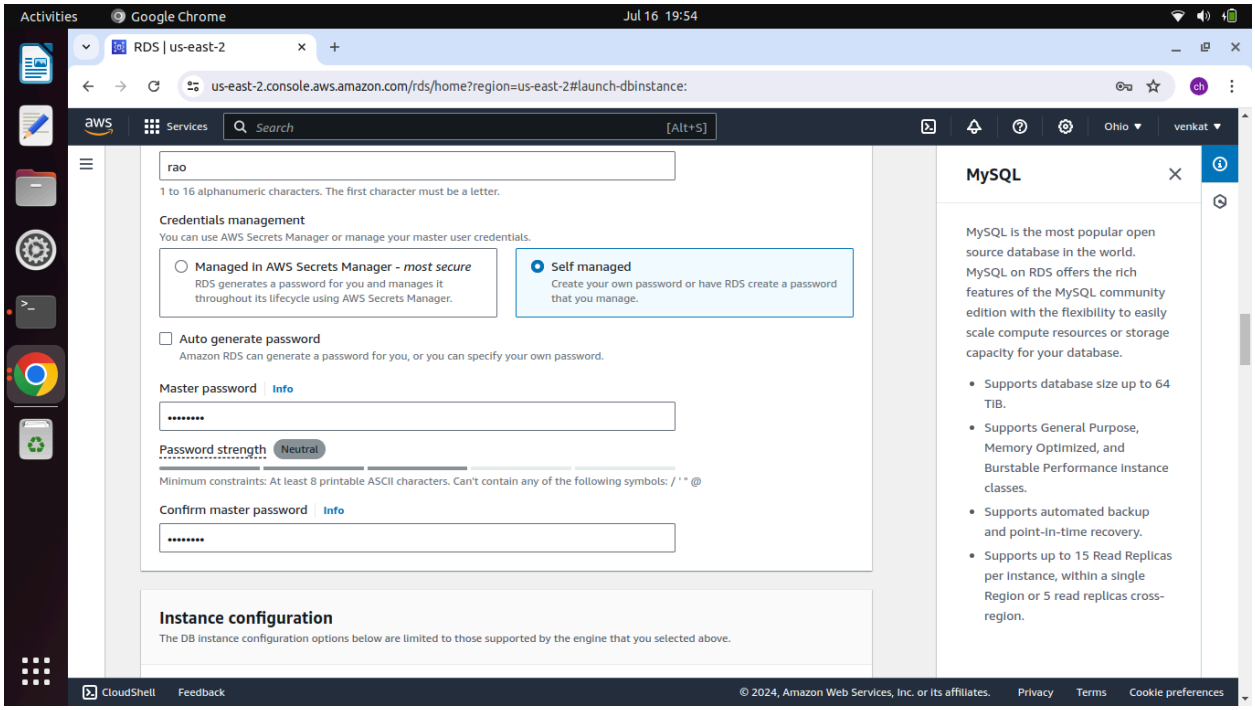
89) Db cluster identifier=rao-database



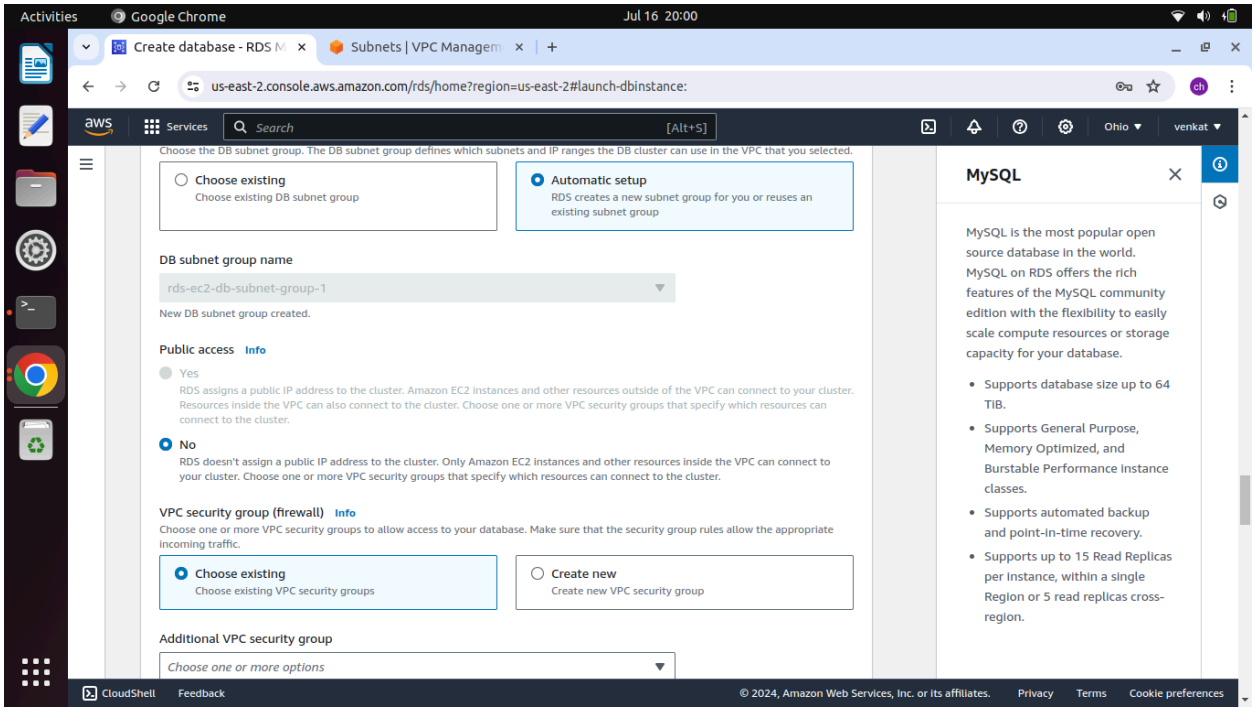
90) username= rao

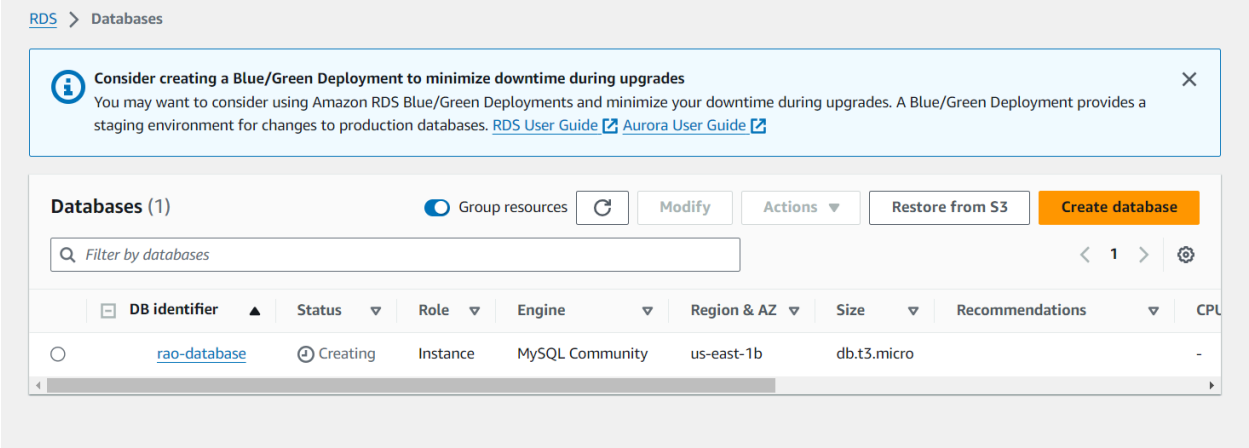
91) Cluster management type= self mange

92) Master password= rao1234



93) connectivity= connect to an ec2 compute resource
94) Click automatic setup
95) Public access= no
96) Vpc sg=choose exciting
97) Select sg click create database





98) Now go to challa instance and install mysql in challa server

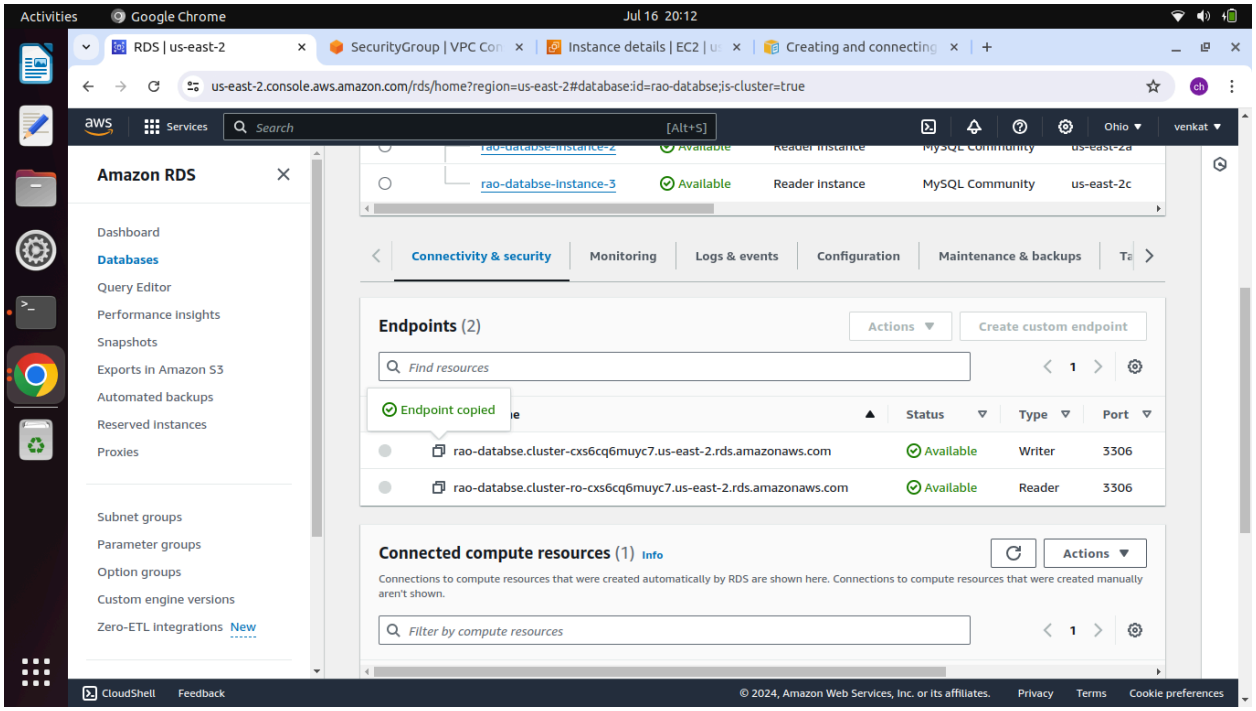
99) Use this link to install my sql in amazon linx https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.CreatingConnecting.MySQL.html

100) #sudo dnf update -y

101) #sudo dnf install mariadb105

102) #mysql - - version

103) Now go to rao-database click rao and copy writer end point



104) Go to terminal

105) #mysql -h paste here endpoint -u username -p click enter and type password now you connect my sql

```
  mariadb105-common-3:10.5.25-1.amzn2023.0.1.x86_64
  perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64

Complete!
[root@ip-10-0-1-58 ~]# mysql --version
mysql  Ver 15.1 Distrib 10.5.25-MariaDB, for Linux (x86_64) using  EditLine wra
pper
[root@ip-10-0-1-58 ~]# mysql -h rao-database.cneeks86817d.us-east-1.rds.amazona
ws.com -u rao -p venkatlovessri
Enter password:
ERROR 1049 (42000): Unknown database 'venkatlovessri'
[root@ip-10-0-1-58 ~]# mysql -h rao-database.cneeks86817d.us-east-1.rds.amazona
ws.com -u rao -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 8.0.35 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)]>
```

106) > show databases;

107) > create database rao;

108) > show databases;

```
root@ip-10-0-1-58:~
MySQL [(none)]> clear
MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.004 sec)

MySQL [(none)]> create database rao;
Query OK, 1 row affected (0.006 sec)

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| rao                |
| sys                |
+--------------------+
5 rows in set (0.001 sec)
```

==========