CloudFront & Route 53-Task

1. Configure VPC peering in cross regions.

- VPC Peering allows **two VPCs to privately communicate** using AWS backbone network(not public internet).It works like a **private tunnel between two VPCs**.
- **Example: - VPC-A** → 10.0.0.0/16**, VPC-B** → 192.168.0.0/16
- We want EC2 in VPC-A to talk to EC2 in VPC-B.

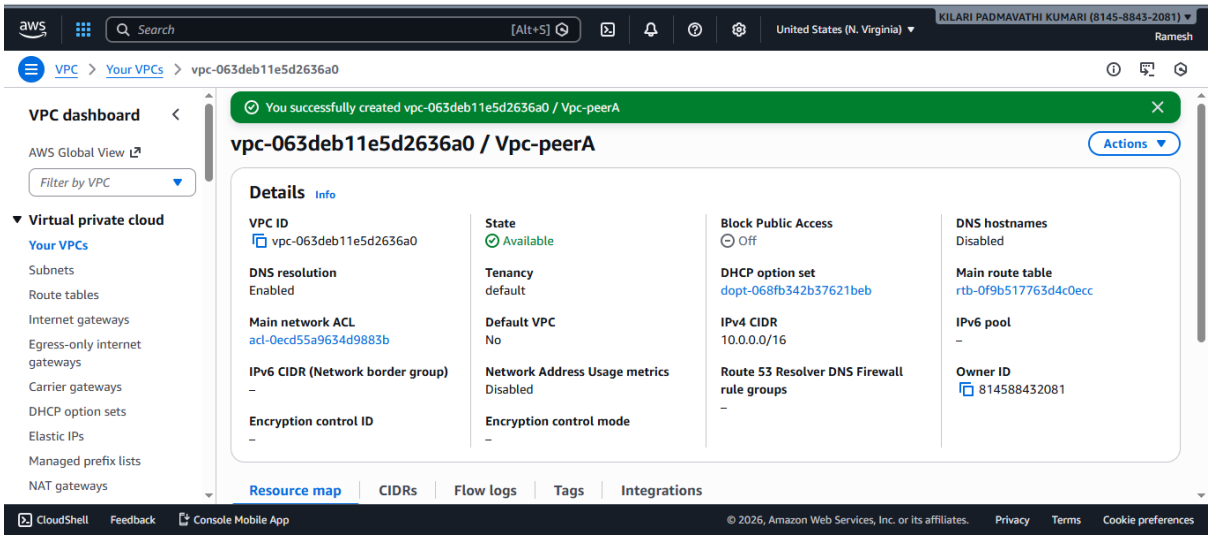
Pre-Requisites:

- ➢ CIDR blocks must **NOT overlap**
- ➢ VPCs can be same or different AWS accounts
- ➢ Same region or different regions allowed

Step 1:

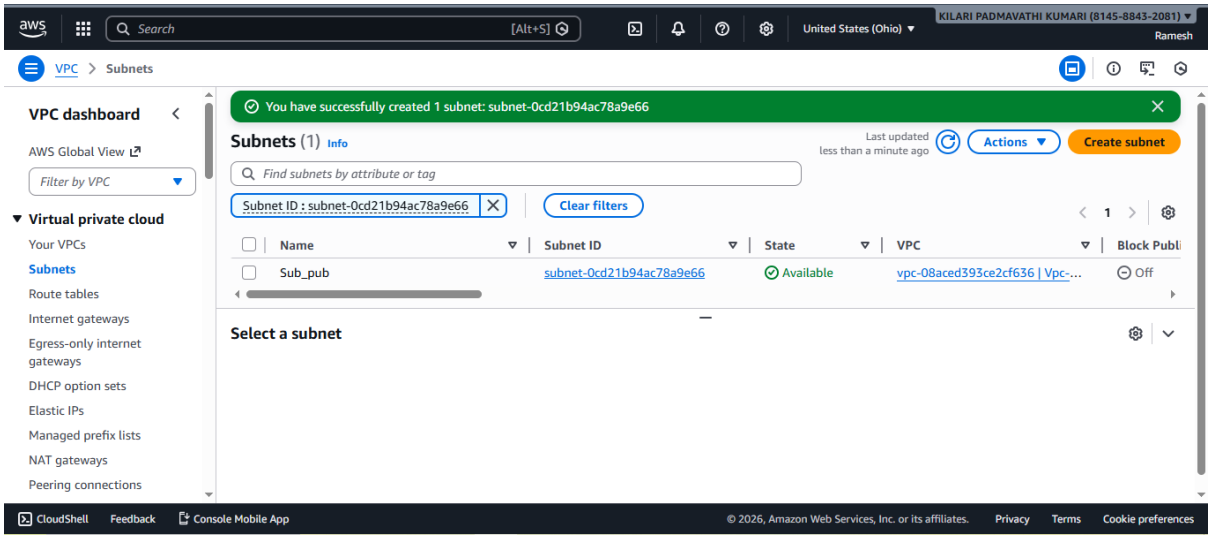
Go to AWS console and search for VPC

Click on create VPC

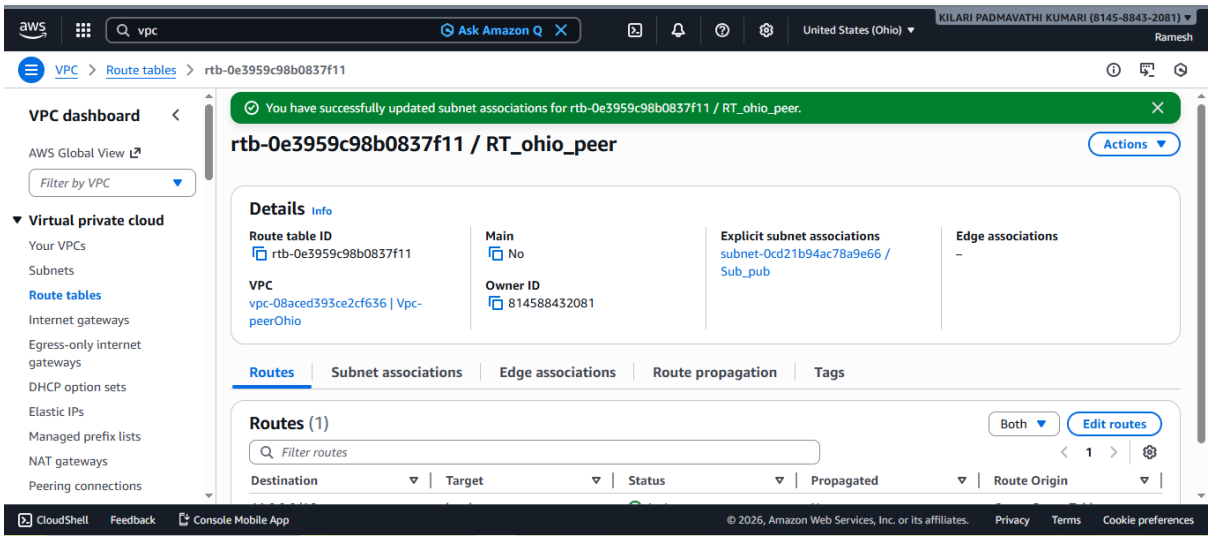


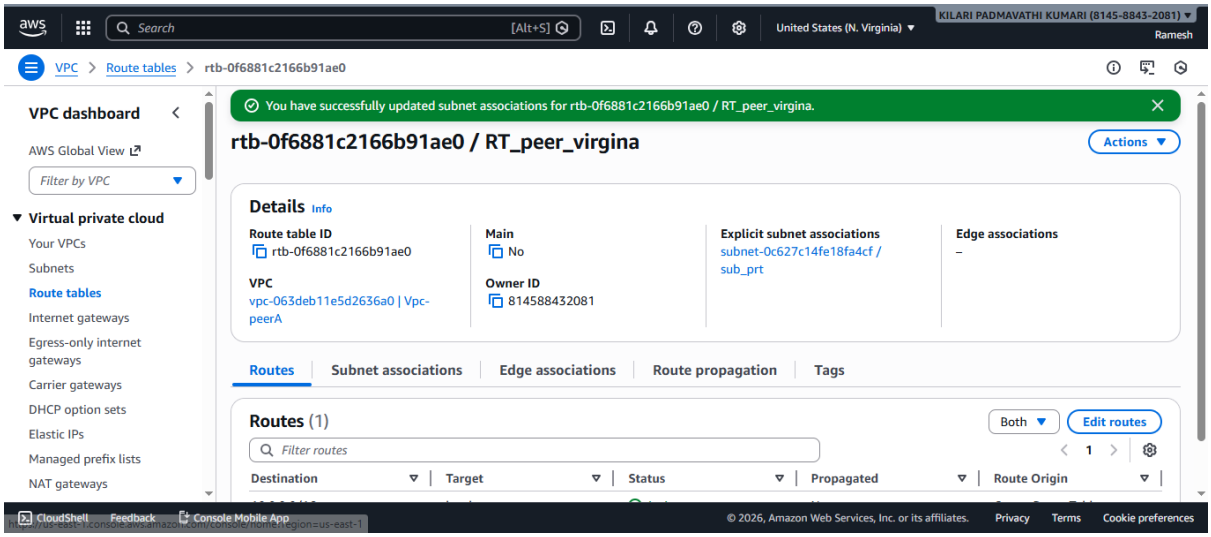
- Click on subnets and create a subnet with CIDR range.

CloudFront & Route 53-Task



- Click route tables
- Edit routes
- Add subnet associations to route table



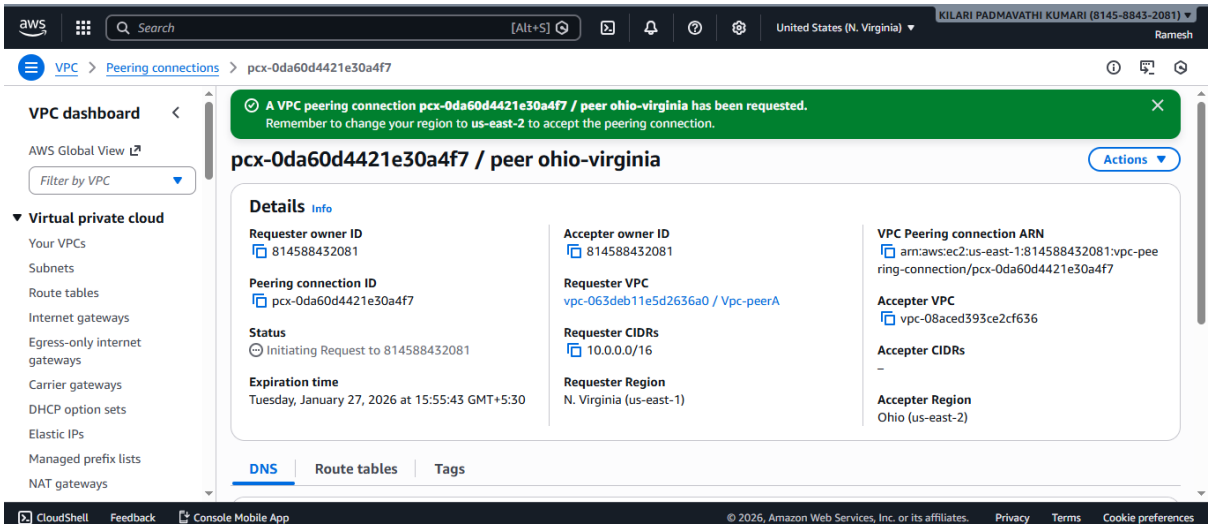


CloudFront & Route 53-Task

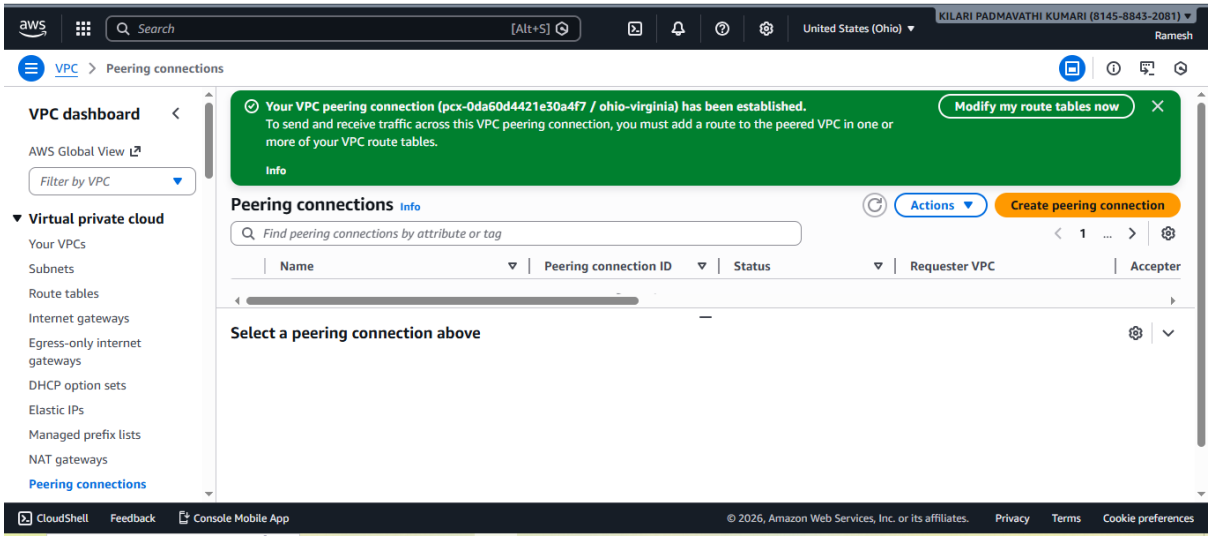
Step 2: Create VPC peering connection

Go to:

VPC Console → Peering Connections → Create Peering Connection→ Action send request to other region(ohio).

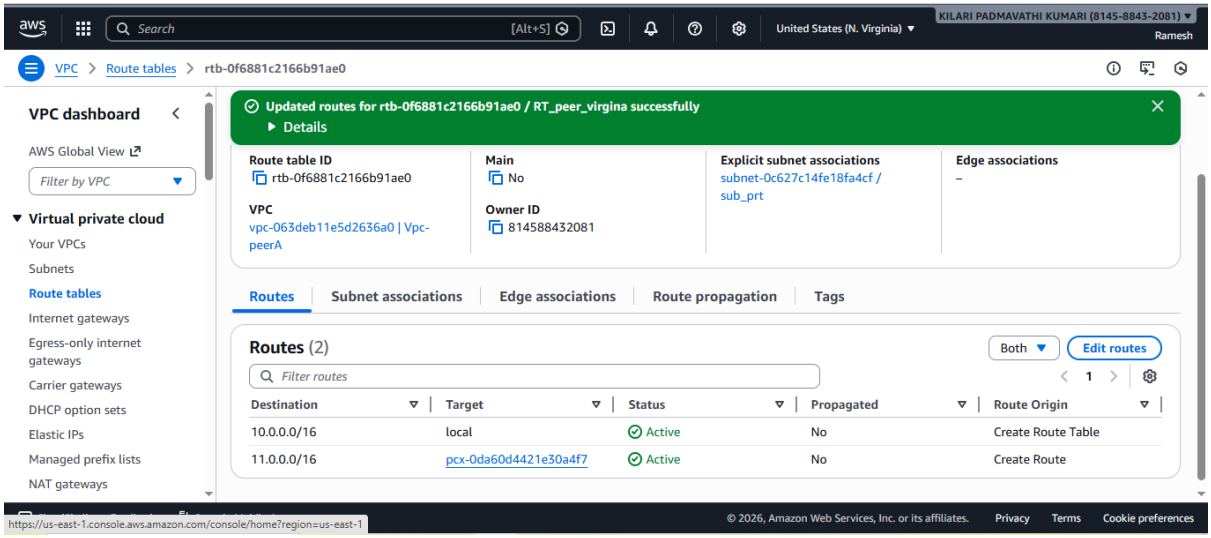


- Go to ohio region and click on peering connection and in action accept the request .

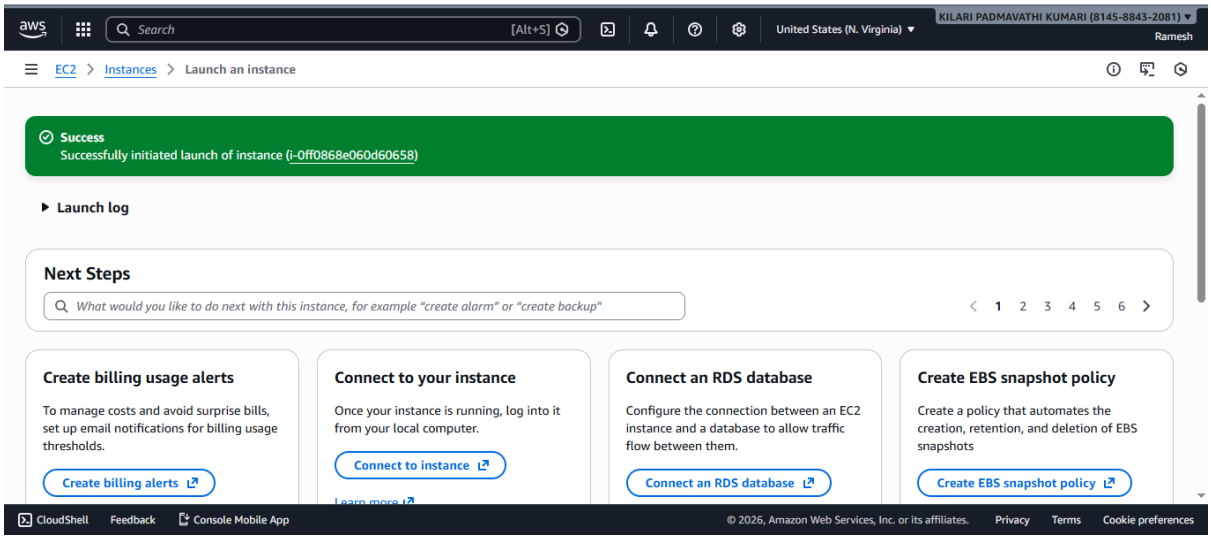


CloudFront & Route 53-Task

- In route tables edit routes and add peering connection ID of opposite region in both regions route tables should update .

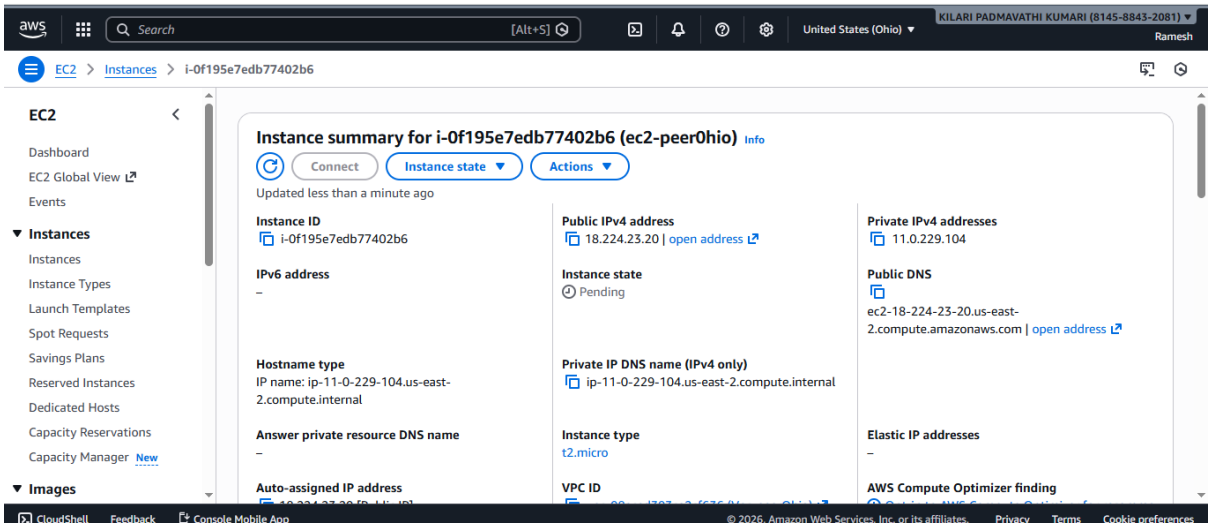


- Go to EC2 click on instance
- And launch one instance as shown below

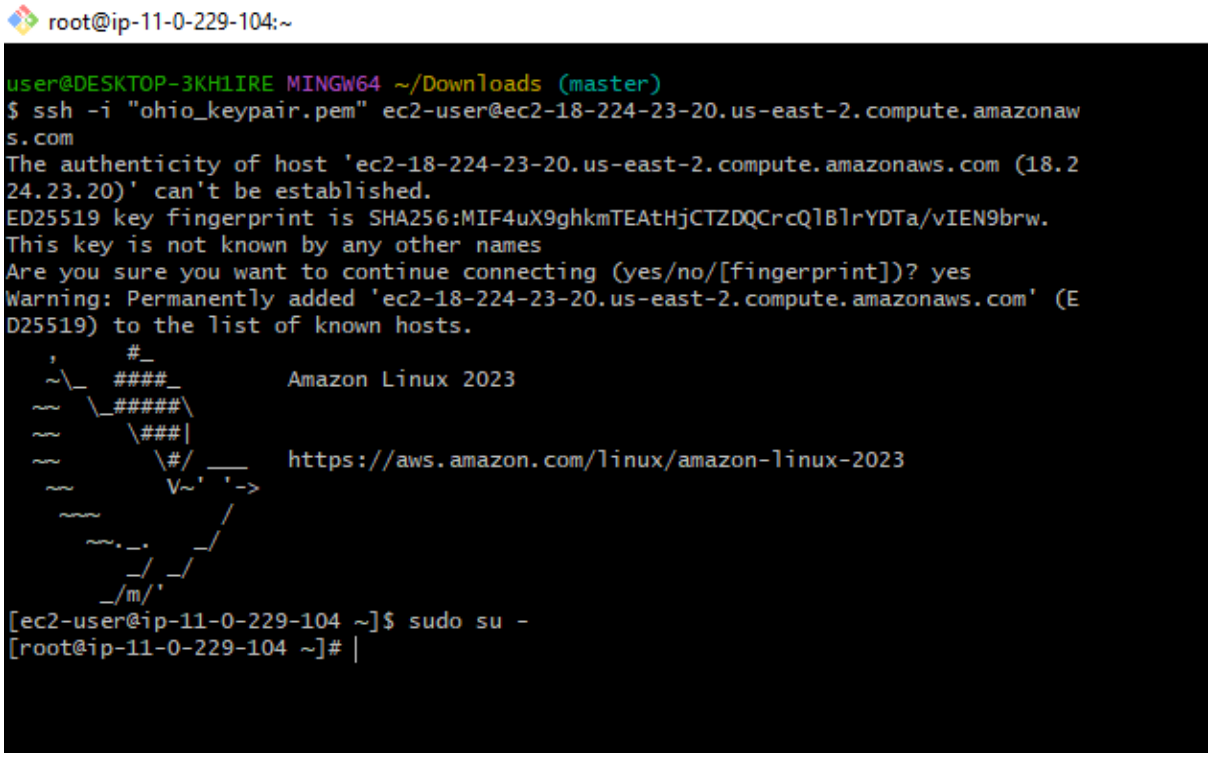


CloudFront & Route 53-Task

- Go to ohio region and launch instance with public ip



- Check connectivity using ssh -i ec2-user@ip



CloudFront & Route 53-Task

Error: After connecting with public we have to connect with private ip here ping is not coming to EC2 private ip .

- So that first we have to switch user to root for avoid error (extra privileges)
- And in SG we should add ICMP protocol.

root@ip-11-0-229-104:~

```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ssh -i "ohio_keypair.pem" ec2-user@ec2-18-224-23-20.us-east-2.compute.amazonaw
s.com
The authenticity of host 'ec2-18-224-23-20.us-east-2.compute.amazonaws.com (18.2
24.23.20)' can't be established.
ED25519 key fingerprint is SHA256:MIF4uX9ghkmTEAtHjCTZDQCrcQlBlrYDTa/vIEN9brw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-224-23-20.us-east-2.compute.amazonaws.com' (E
D25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-11-0-229-104 ~]$ sudo su -
[root@ip-11-0-229-104 ~]# ping 10.0.219.45
PING 10.0.219.45 (10.0.219.45) 56(84) bytes of data.
```

```
[root@ip-11-0-229-104 ~]# ping 10.0.219.45
PING 10.0.219.45 (10.0.219.45) 56(84) bytes of data.
64 bytes from 10.0.219.45: icmp_seq=160 ttl=127 time=12.3 ms
64 bytes from 10.0.219.45: icmp_seq=161 ttl=127 time=12.1 ms
64 bytes from 10.0.219.45: icmp_seq=162 ttl=127 time=12.3 ms
64 bytes from 10.0.219.45: icmp_seq=163 ttl=127 time=12.2 ms
64 bytes from 10.0.219.45: icmp_seq=164 ttl=127 time=12.1 ms
64 bytes from 10.0.219.45: icmp_seq=165 ttl=127 time=12.7 ms
64 bytes from 10.0.219.45: icmp_seq=166 ttl=127 time=12.2 ms
64 bytes from 10.0.219.45: icmp_seq=167 ttl=127 time=12.1 ms
64 bytes from 10.0.219.45: icmp_seq=168 ttl=127 time=12.2 ms
64 bytes from 10.0.219.45: icmp_seq=169 ttl=127 time=12.5 ms
64 bytes from 10.0.219.45: icmp_seq=170 ttl=127 time=12.8 ms
64 bytes from 10.0.219.45: icmp_seq=171 ttl=127 time=12.1 ms
64 bytes from 10.0.219.45: icmp_seq=172 ttl=127 time=12.2 ms
64 bytes from 10.0.219.45: icmp_seq=173 ttl=127 time=12.8 ms
64 bytes from 10.0.219.45: icmp_seq=174 ttl=127 time=12.5 ms
64 bytes from 10.0.219.45: icmp_seq=175 ttl=127 time=12.5 ms
64 bytes from 10.0.219.45: icmp_seq=176 ttl=127 time=12.3 ms
^C
--- 10.0.219.45 ping statistics ---
176 packets transmitted, 17 received, 90.3409% packet loss, time 181385ms
rtt min/avg/max/mdev = 12.093/12.355/12.828/0.234 ms
[root@ip-11-0-229-104 ~]#
```

- Conclusion: The above image shows ec2 in both regions are connected with private ip’s

CloudFront & Route 53-Task

2. Purchase one domain from GoDaddy.

➢ **Step-1:** Open GoDaddy

Go to 👉 https://www.godaddy.com
Click **Sign In** → Login or Create account

➢ **Step-2: Search for Domain**

- In search box type your domain name
  Example:
- neelimaranidevops.online
- Click **Search**

It will show:

- ✅ Available
- ❌ Taken
- 🔄 Suggested alternatives

➢ **Step-3: Add to Cart**

Click **Add to Cart**
Click **Continue to Cart**
⚠ Skip all extras:

- Domain protection
- Website builder
- Email

(you don't need them for AWS)
Click **Continue to Checkout**

➢ **Step-4: Payment**

Choose:

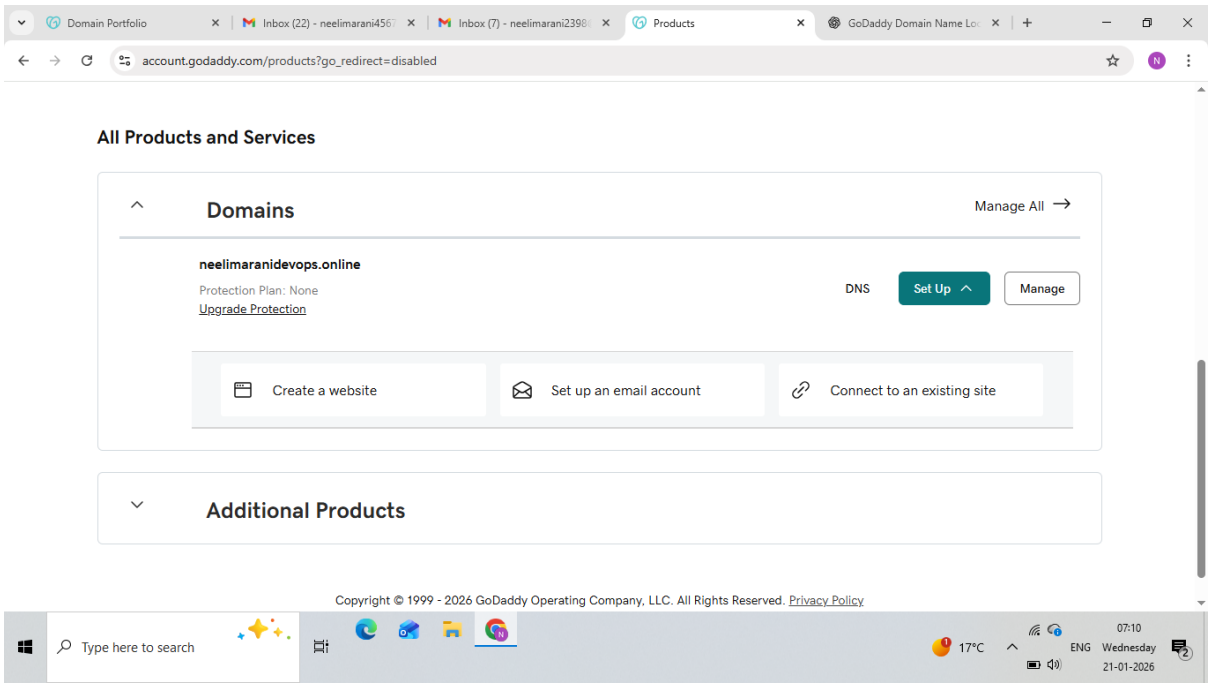
- 1 year (enough)
- Payment method (UPI / Card)

Click **Complete Purchase**

➢ **Step-5: Verify Domain**

- GoDaddy will send verification email
  Open mail → Click **Verify Domain**
- Without this, DNS will not work ❌

CloudFront & Route 53-Task

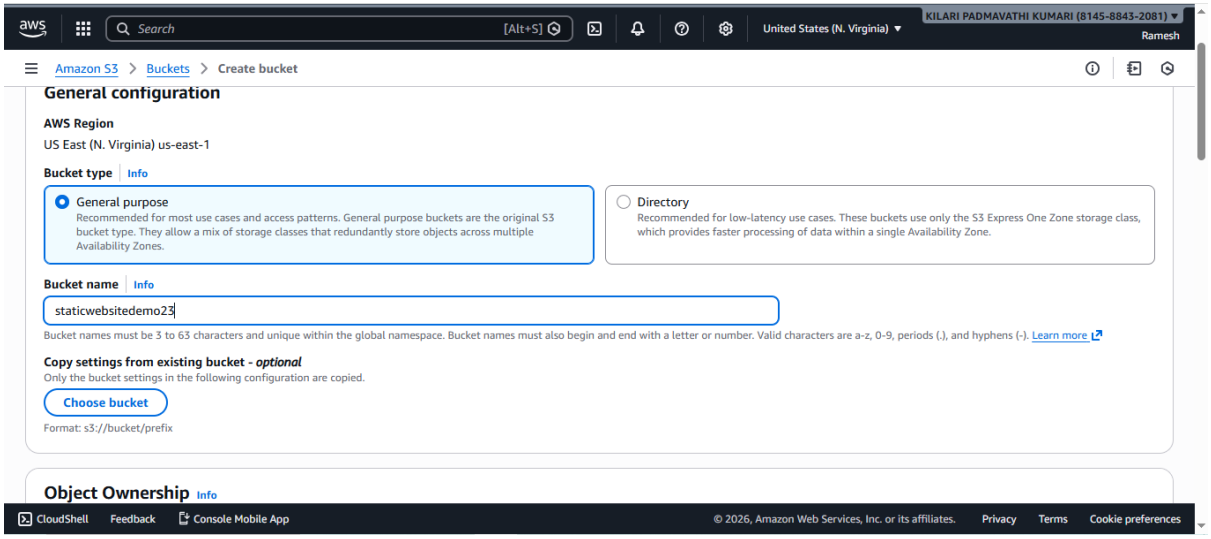


The above image shows Domain in Godaddy website in my products.

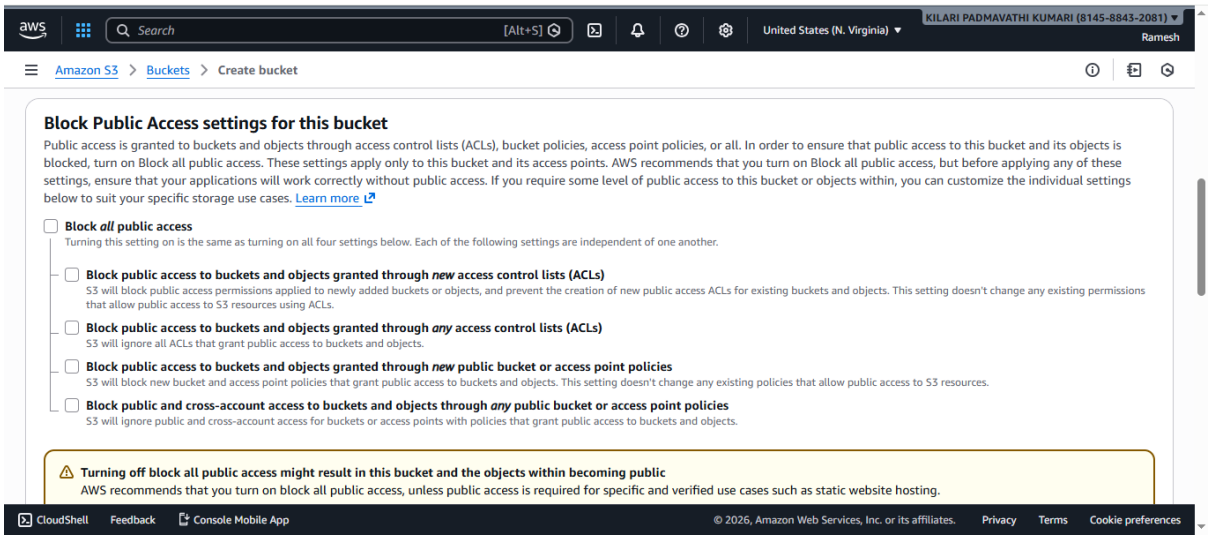
3. Deploy static website in S3.

Step-1: Create S3 Bucket

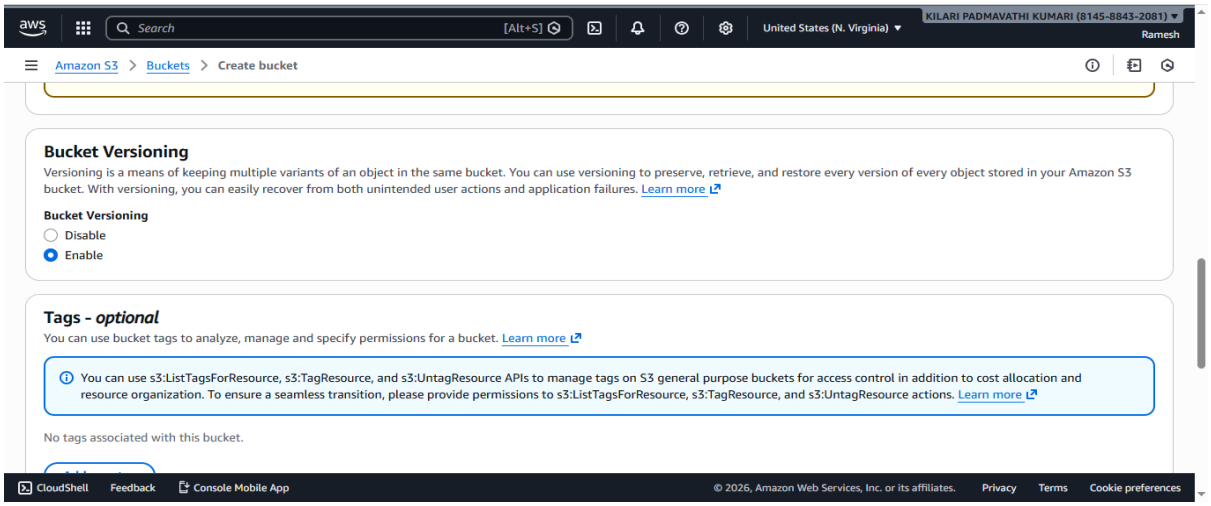
- Go to:
  AWS → S3 → Create bucket
- Bucket name: staticwebsitedemo23(bucket name should be unique)



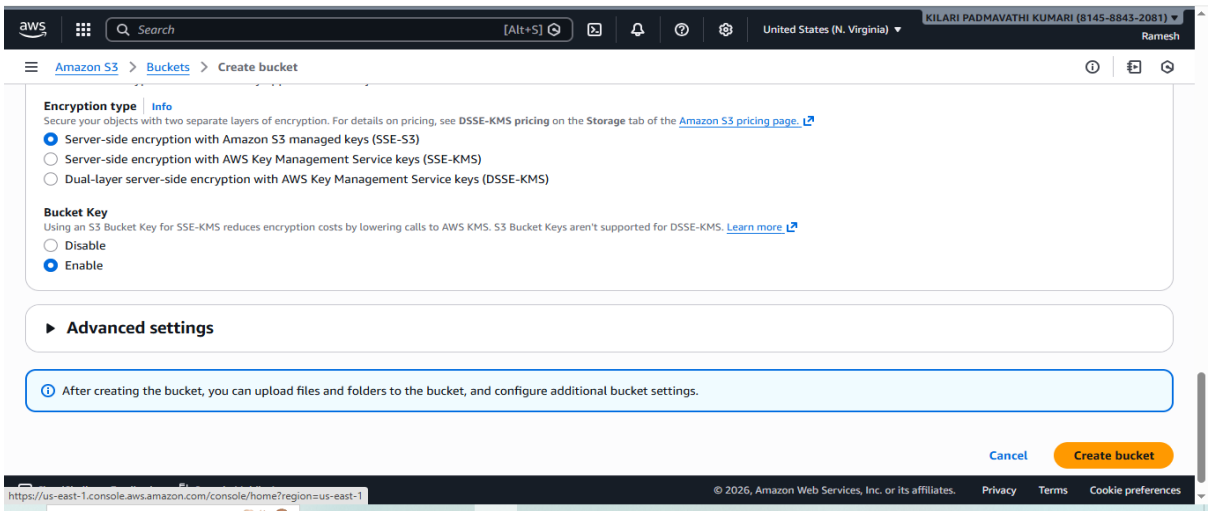
CloudFront & Route 53-Task



- Block Public Access settings for this bucket uncheck it.

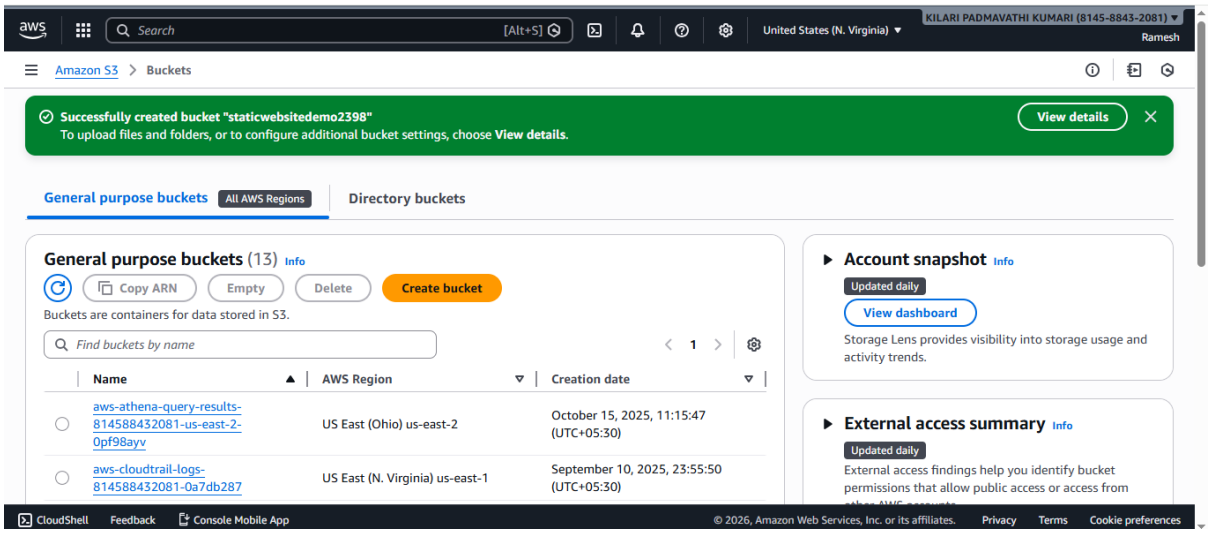


- Bucket versioning must be enable

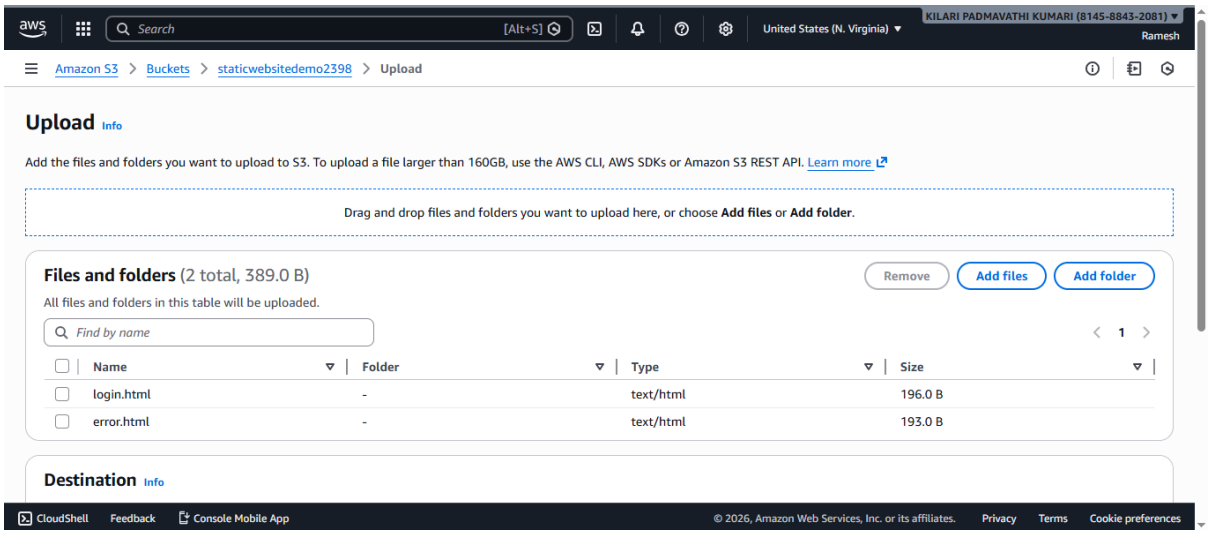


- Click on create bucket.

CloudFront & Route 53-Task

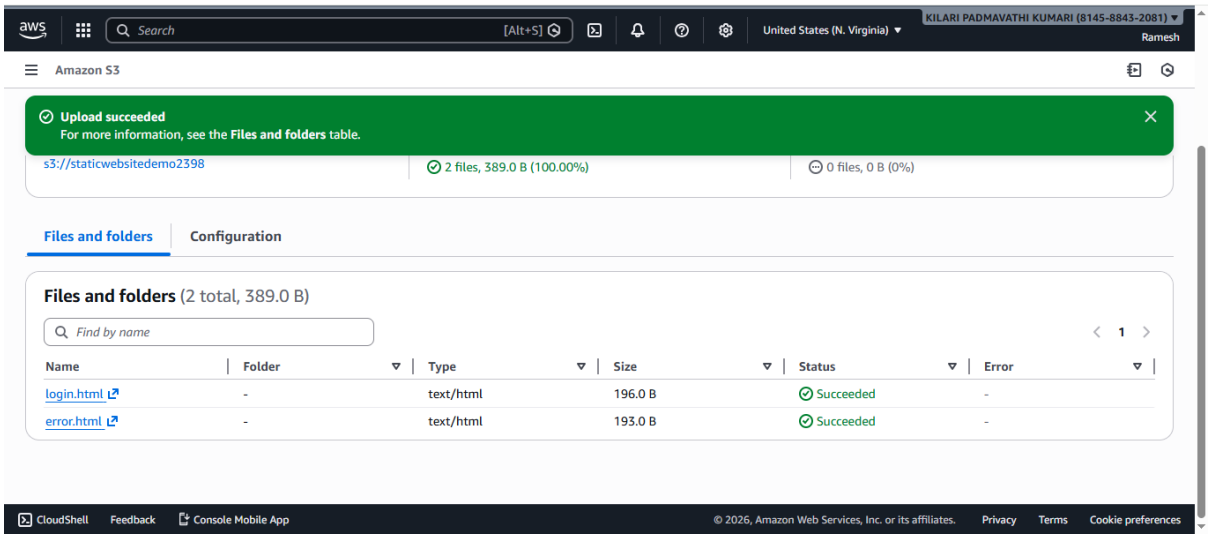


- The above image shows bucket created successfully.



- Search for bucket we created and click on it
- Click on upload and add files to bucket.

CloudFront & Route 53-Task



- Click on upload, we can see files uploaded into bucket.

- When we are trying to access bucket by object url it showing error so to rectify it we show follow some steps:-

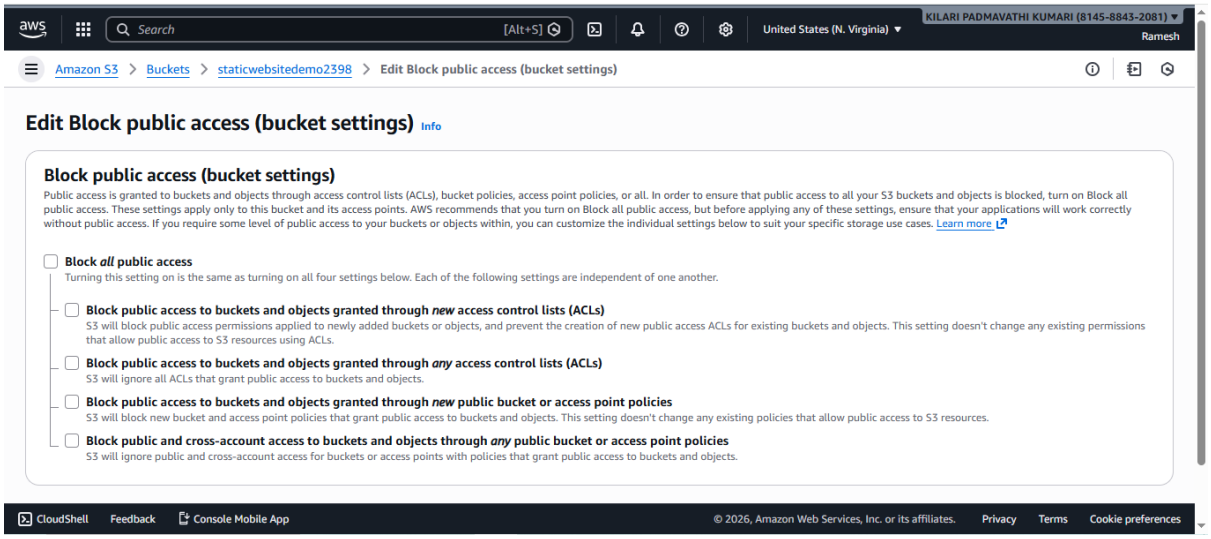
Step-1: Turn OFF Block Public Access

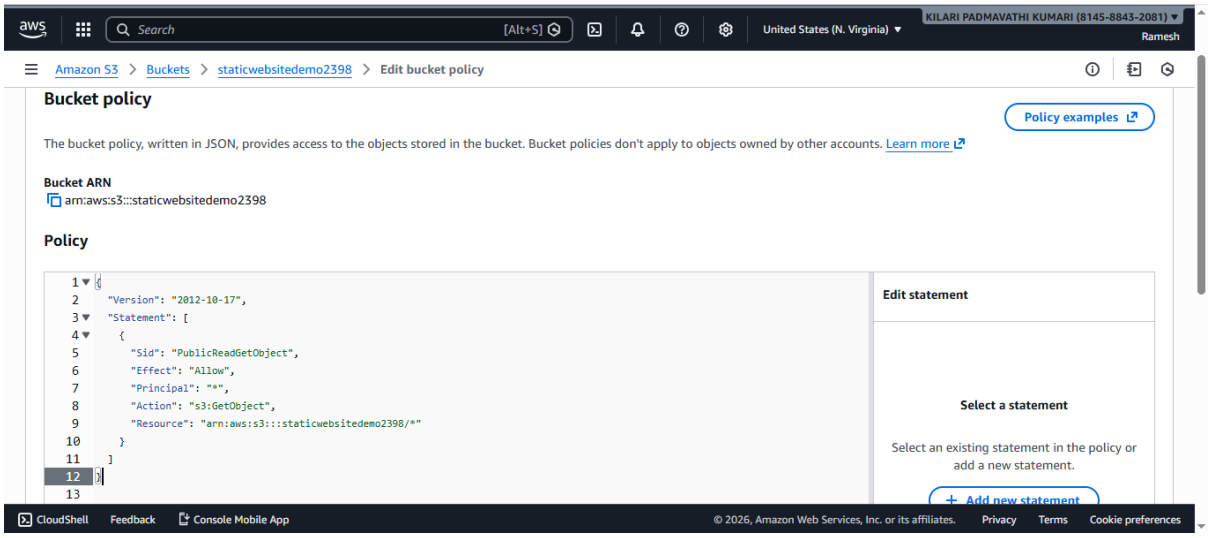
Go to

S3 → Your bucket → Permissions

- Click **Block public access (Edit)**
- Uncheck **ALL** boxes
  Check **"I acknowledge…"**
- Click **Save**

CloudFront & Route 53-Task



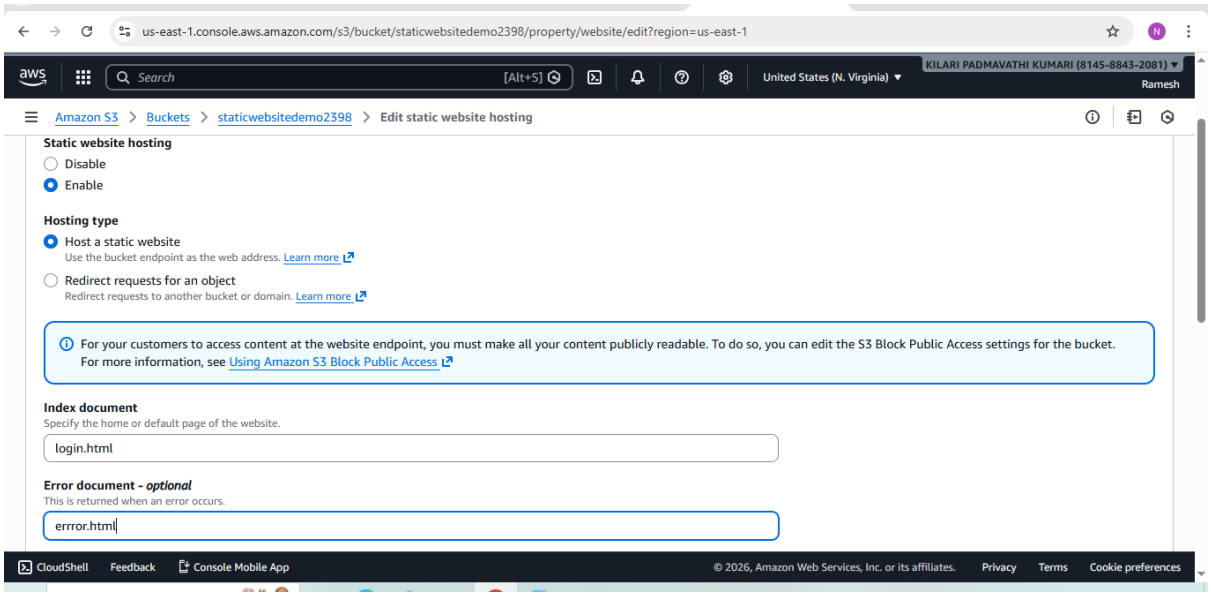


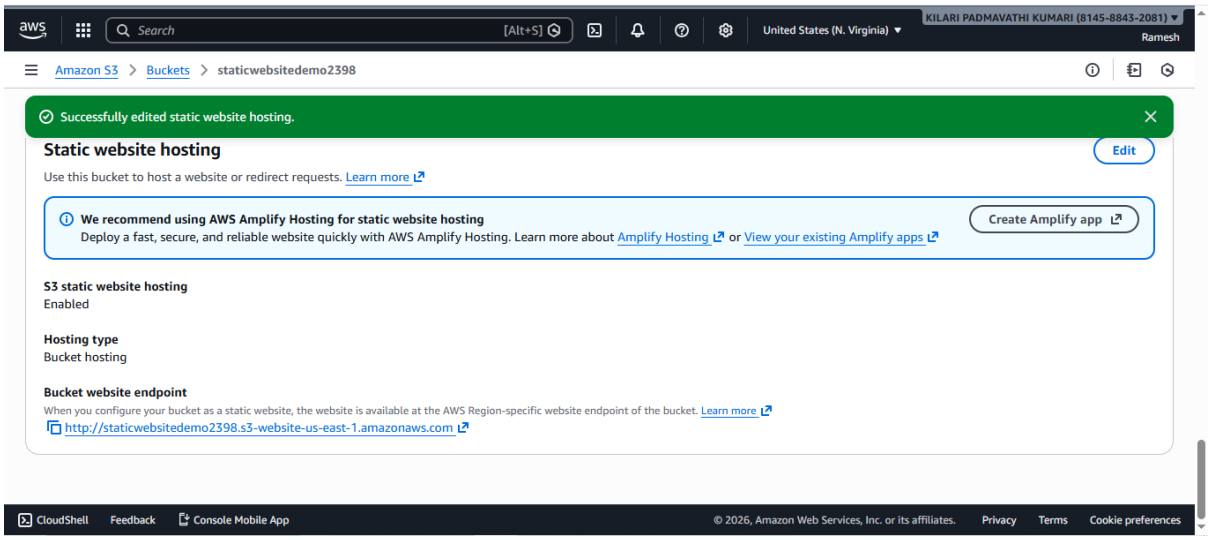
- Edit bucket policy and add policy to it.

Step-3: Confirm Static Website Hosting

- Go to **Properties → Static website hosting**
- Specify the default page of the website

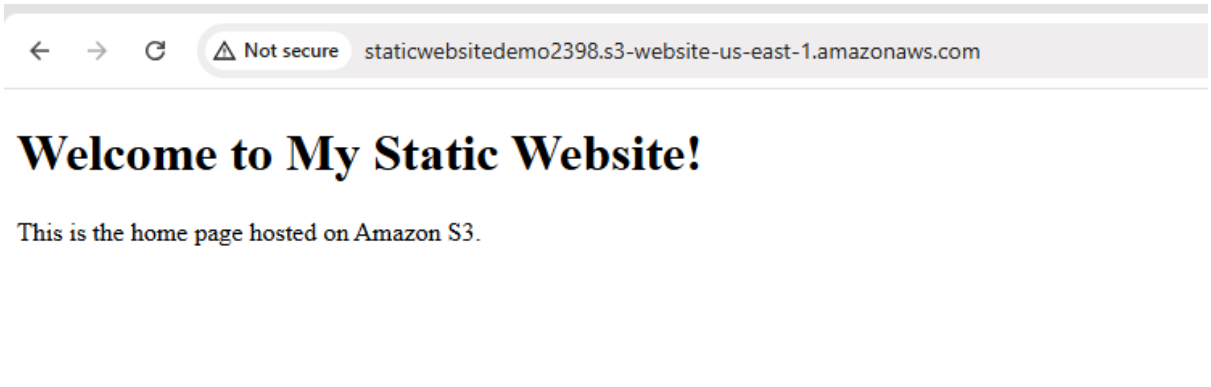
CloudFront & Route 53-Task





- It shows successfully edited static website hosting
- It provided bucket website endpoint
- By clicking on URL we access our static website.

CloudFront & Route 53-Task



Conclusion: The above image shows static website page which is present in s3 bucket.

CloudFront & Route 53-Task

4. Create a CDN and attach one SSL certificate.

- CDN and Attach one SSL certificate,for making your S3 website secure and global

  Requriements:
- Domain Purchased(neelimaranidevops.online)
- Subdomain Planned(www.neelimaranidevops.online)
- S3 Static Website Ready
- Bucket Name Must Match Domain

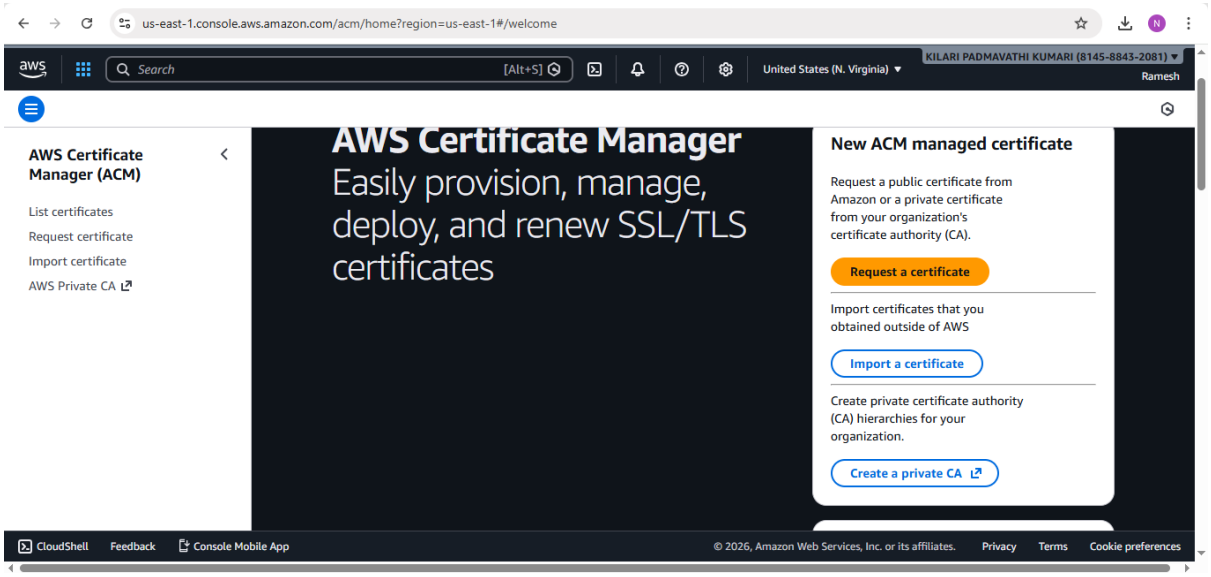
STEP 1— Request SSL Certificate (ACM)

CloudFront **needs SSL in us-east-1**

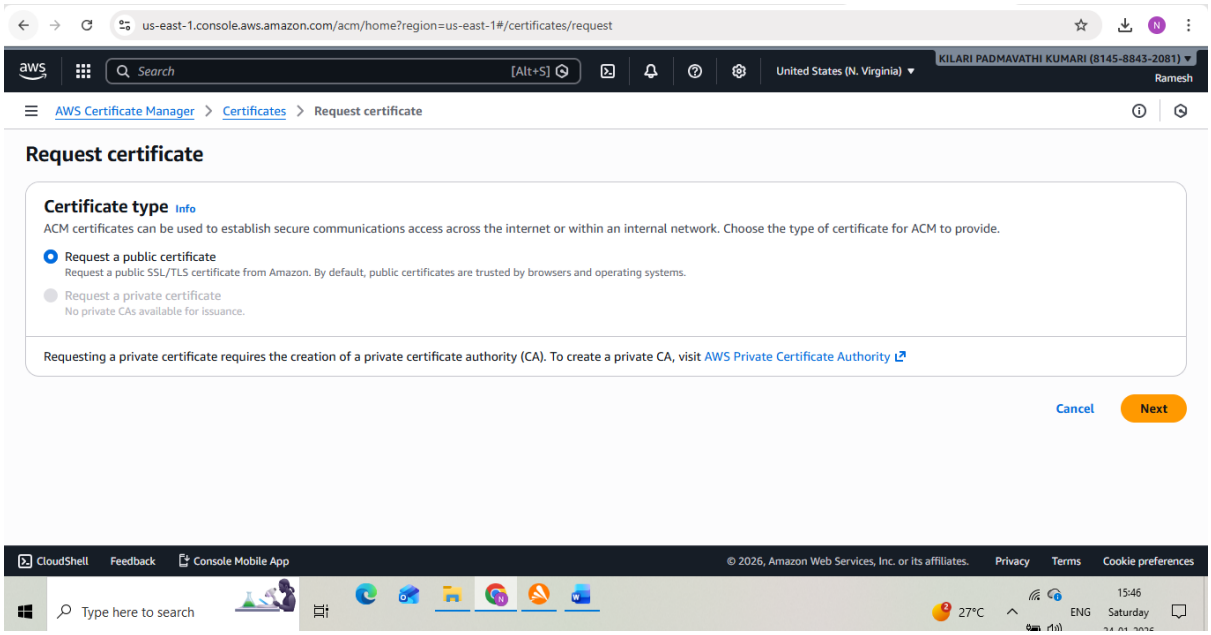
1. AWS → **ACM**
2. Change region → **N. Virginia (us-east-1)**
3. Click **Request certificate**
4. Enter:

neelimaranidevops.online

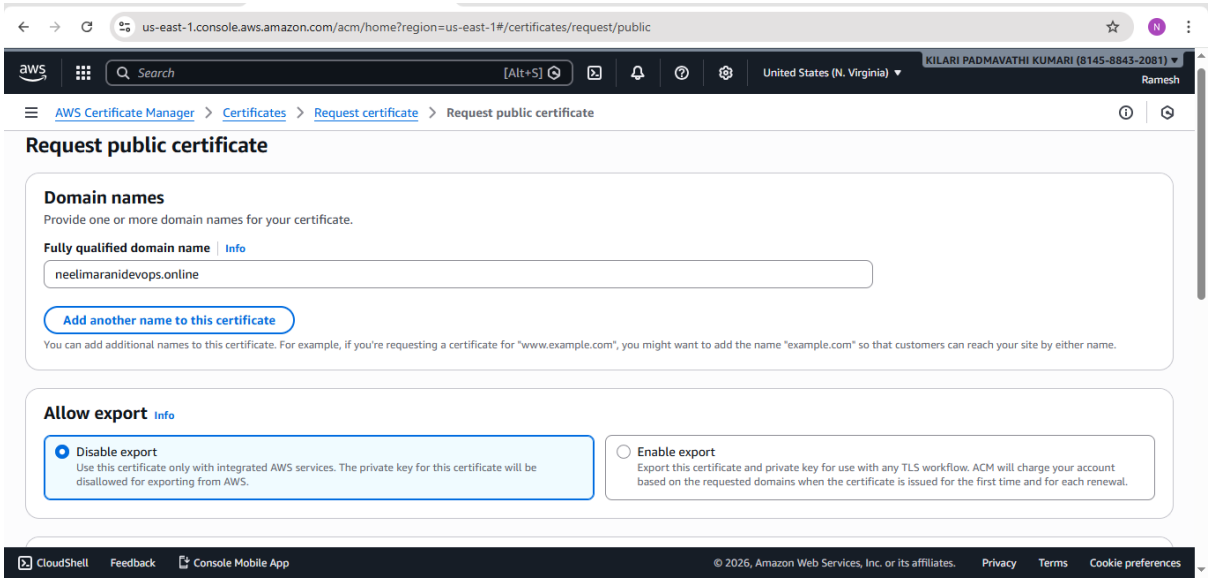
www.neelimaranidevops. online



CloudFront & Route 53-Task

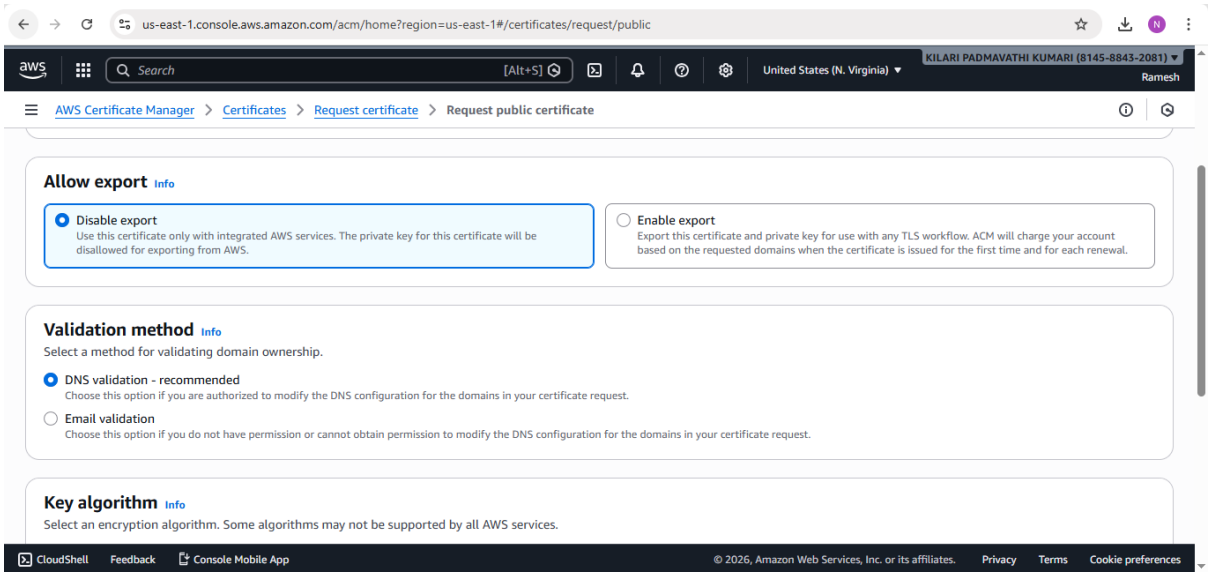


- Click on Request a public certificate.

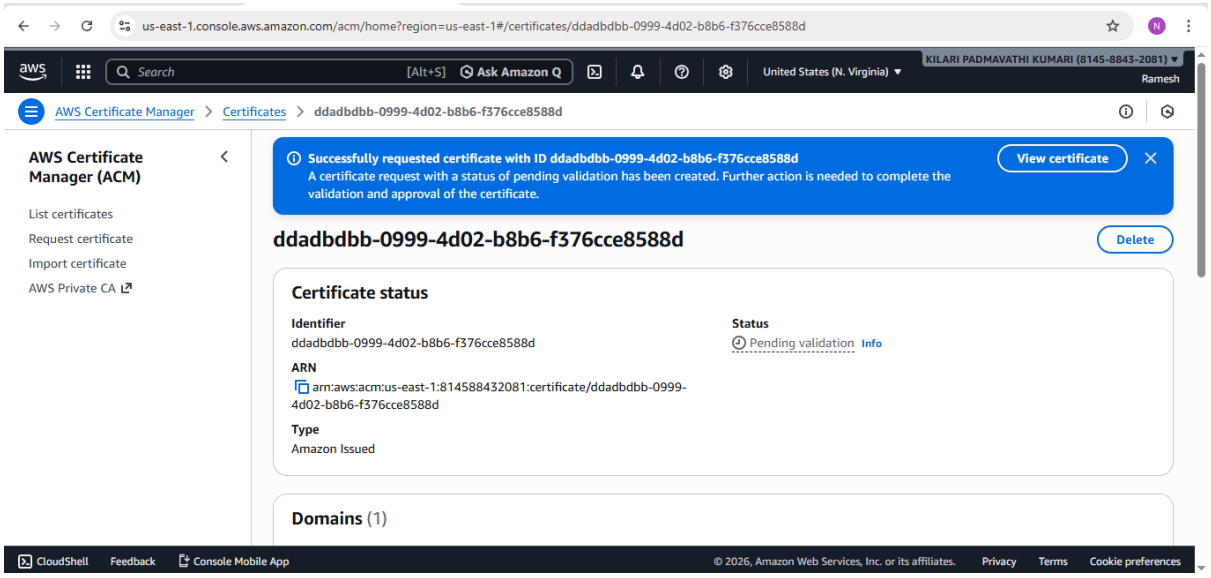


- Add domain name :neelimaranidevops.online.

CloudFront & Route 53-Task

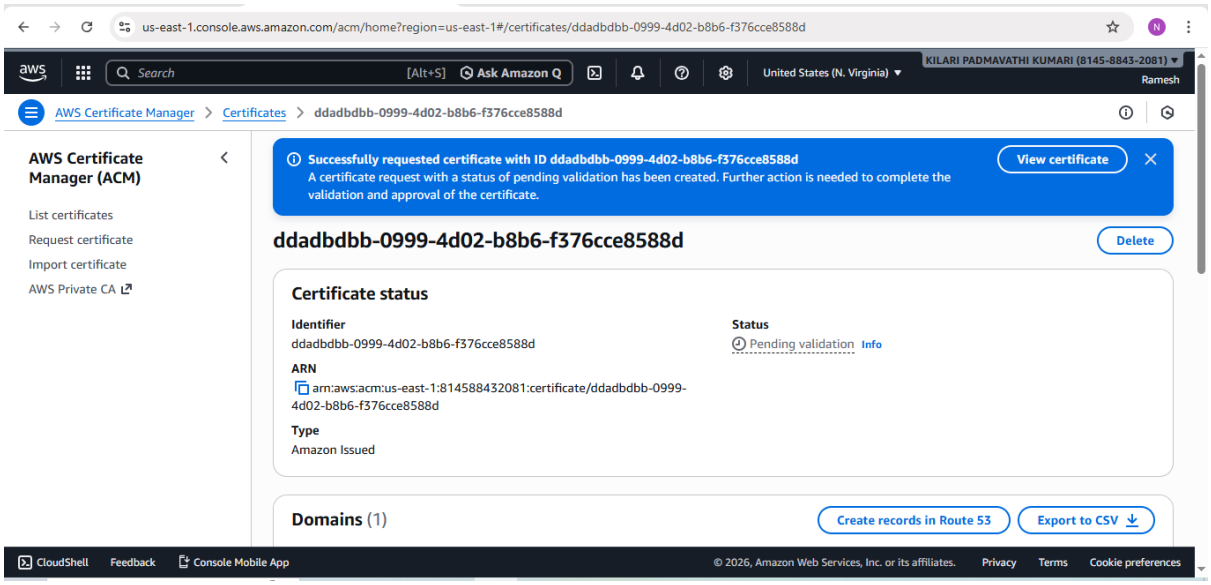


- Validation method: click on DNS validations
- Click on create certificate.



- The above image shows successfully request certificate with ID XXXx.

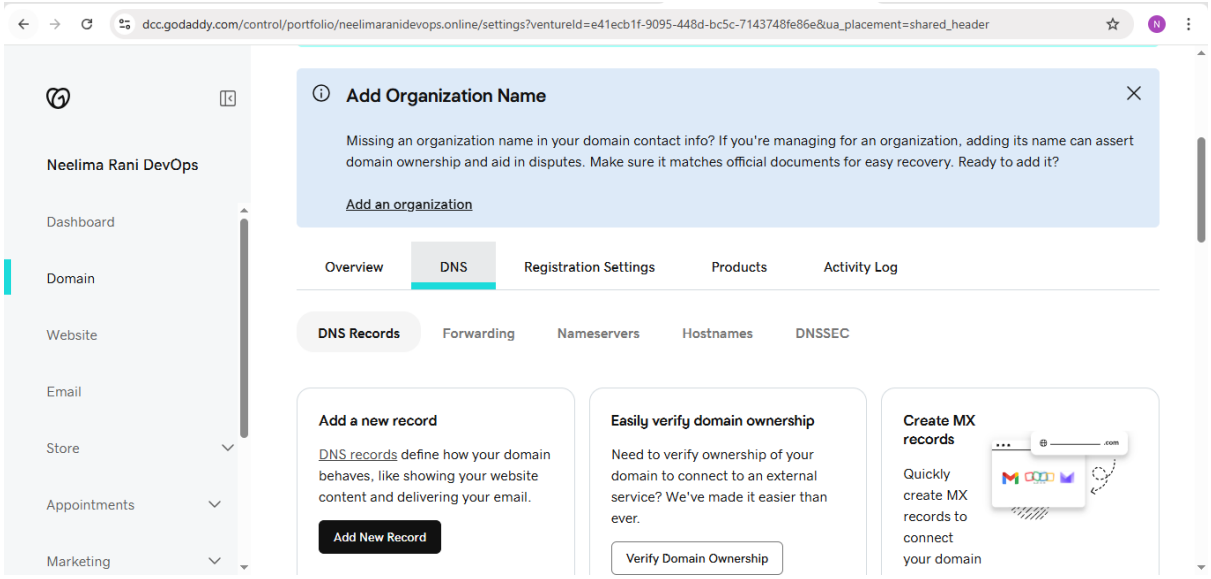
CloudFront & Route 53-Task



- Scroll to **Domains → neelimaranidevops.online**
  You will see **CNAME Name** and **CNAME Value** (DNS validation records).

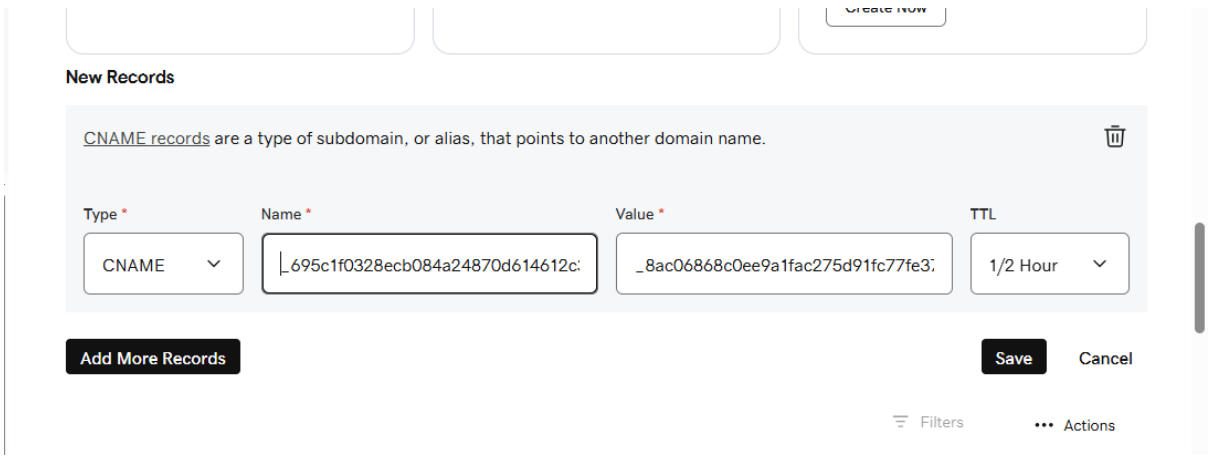
It will look like:

Name	Value
_a1b2c3.neelimaranidevops.online	_x9y8z.acm-validations.aws

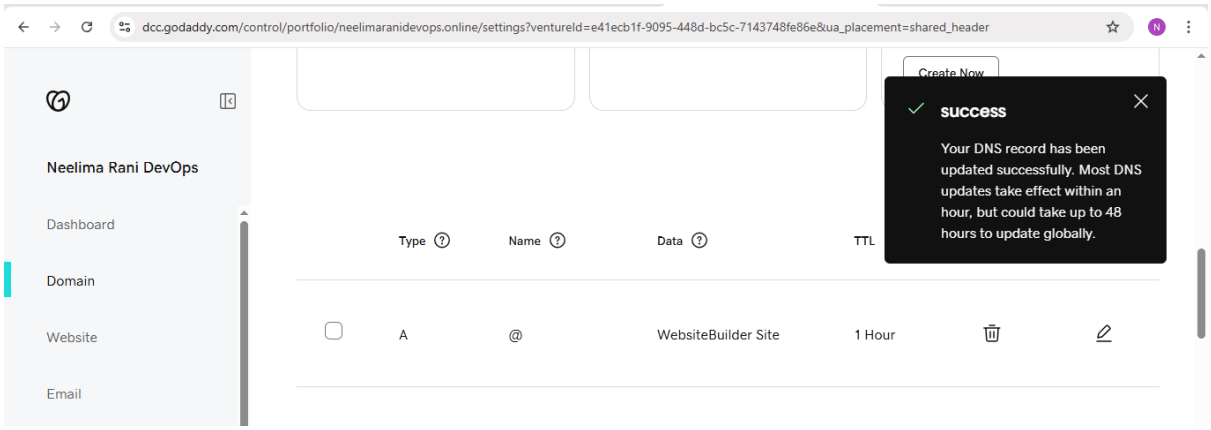


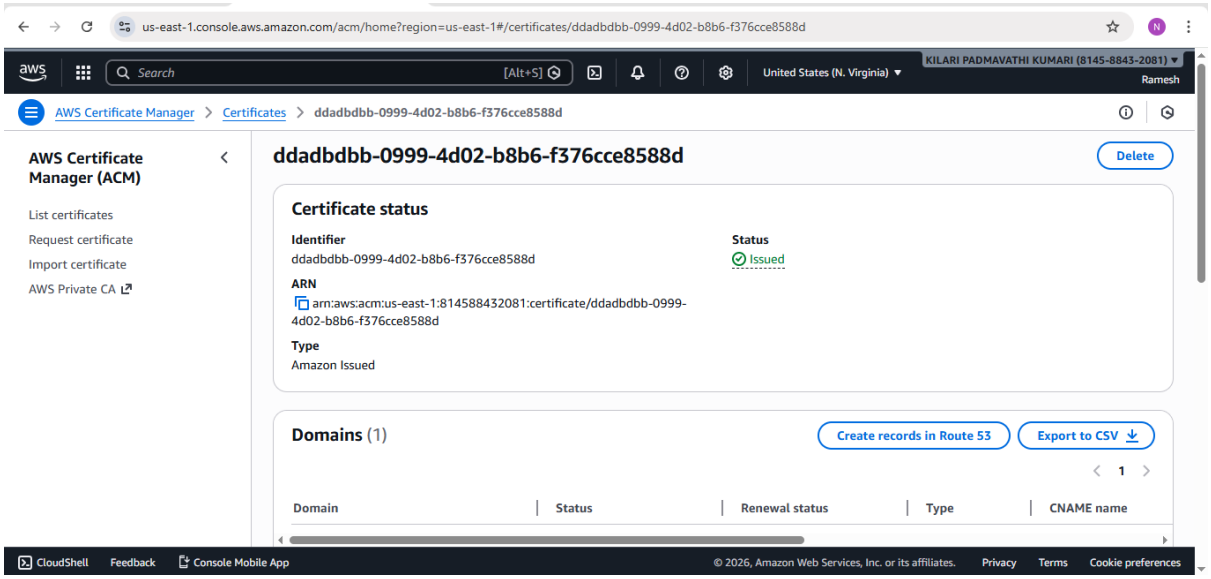
- GO to Godaddy website and click on Domain →DNS
- Add a new record in it.

CloudFront & Route 53-Task



Add TYPE: CNAME ,NAME,VALUE,TTL and click on save.





- And certificate status will be shown as issued.

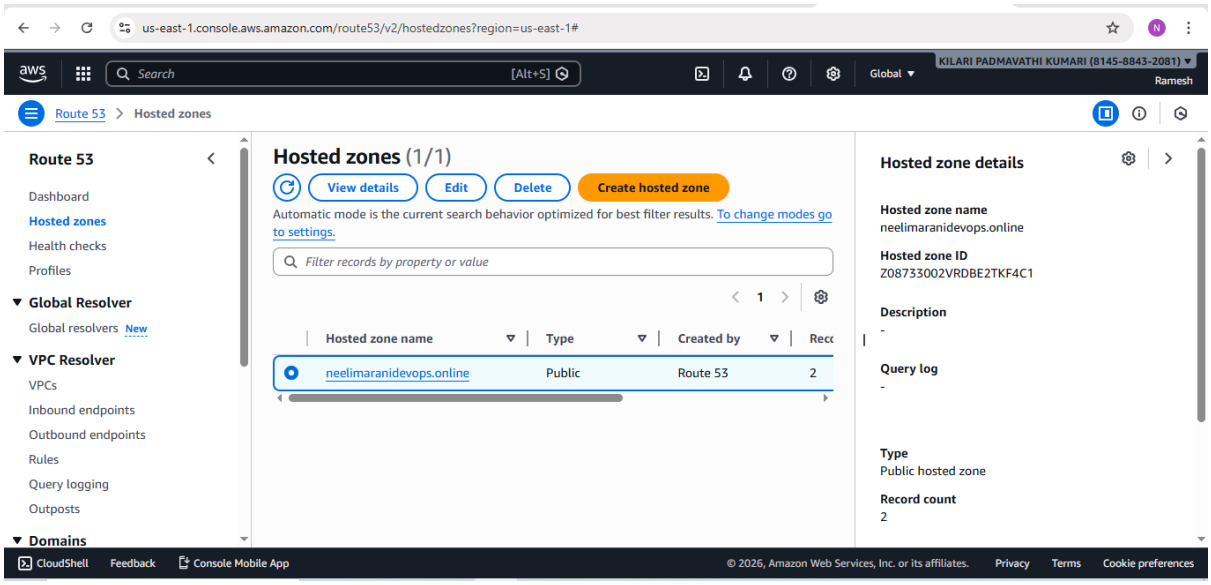
STEP 4 — Add CloudFront Records in Route 53

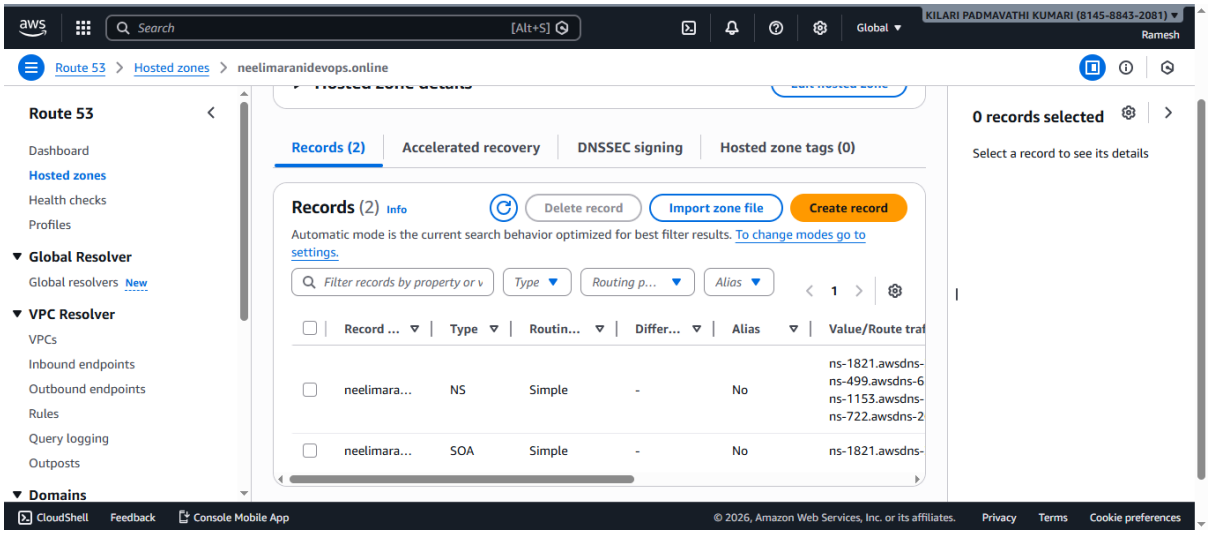
Go back to:

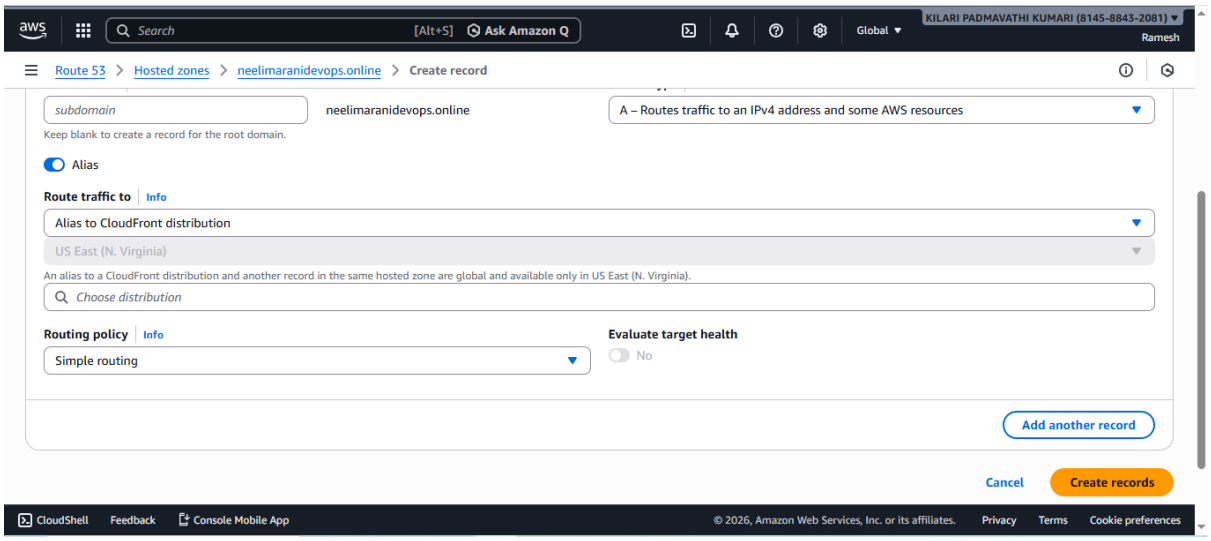
Route 53 → Hosted Zone → neelimaranidevops.online → Create record

CloudFront & Route 53-Task

Create:

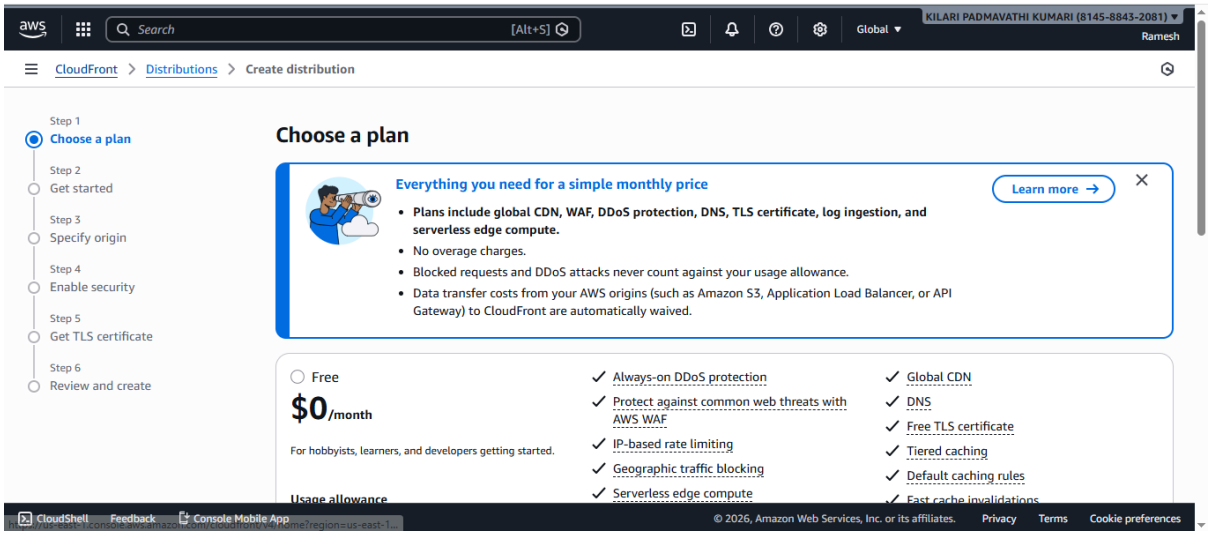


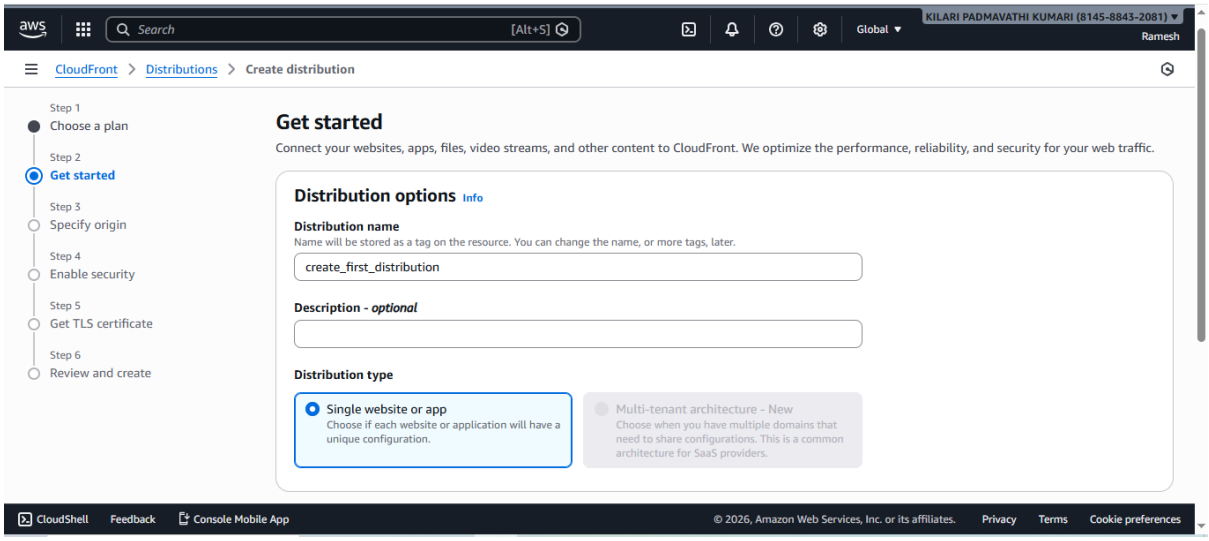




CloudFront & Route 53-Task

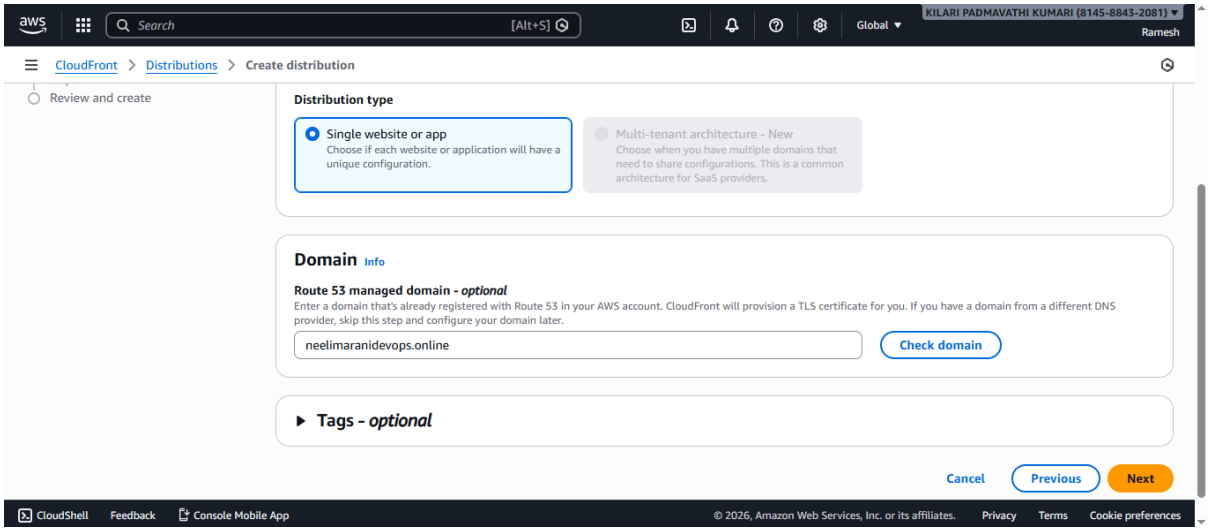
- We required cloud front distrubtion.



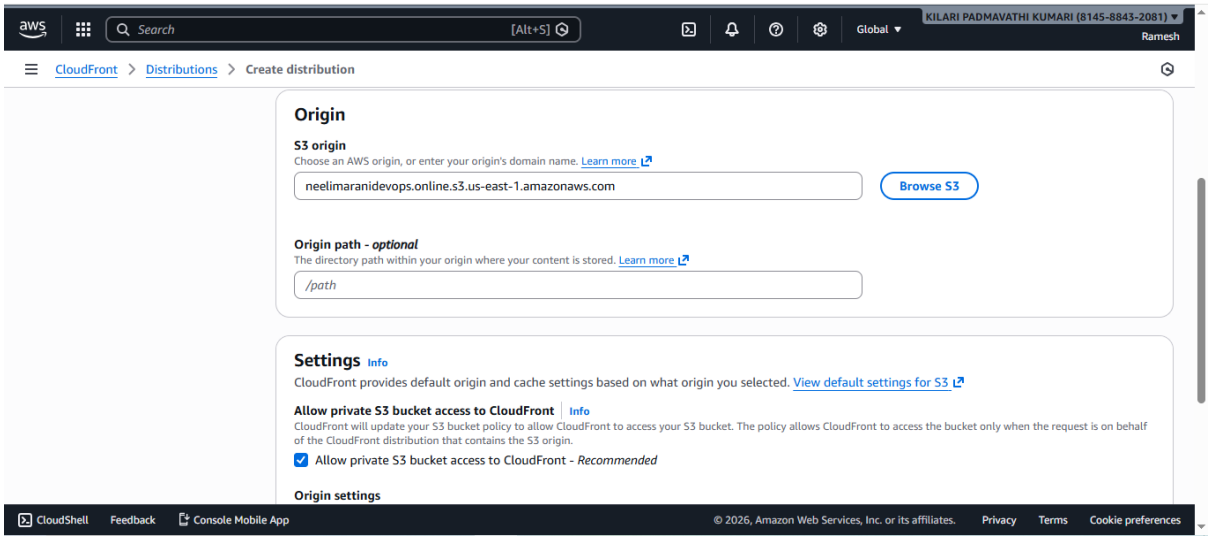


- Go to cloud front → distribution→ create distribution.

CloudFront & Route 53-Task

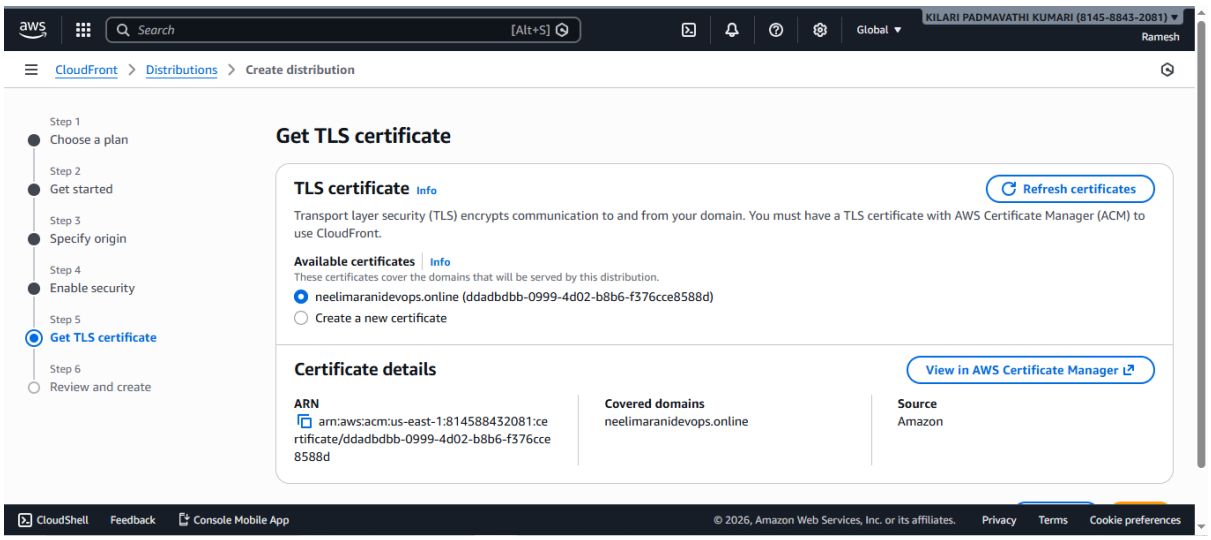


- Check domain and click next

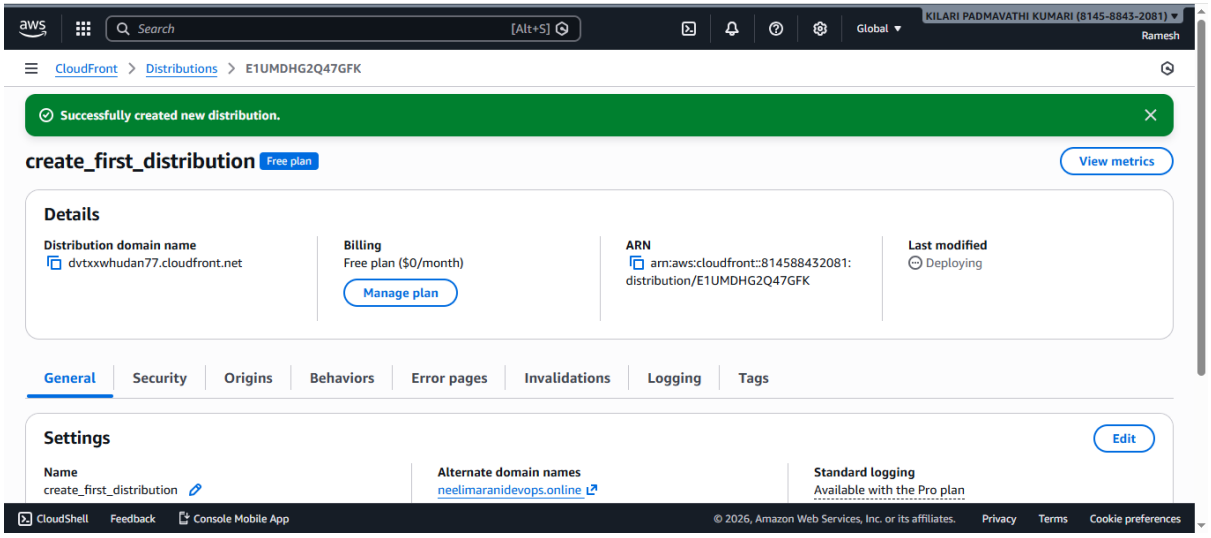


- Specify origin →s3 origin browse s3
- Settings→Allow private s3 bucket access to CloudFront

CloudFront & Route 53-Task

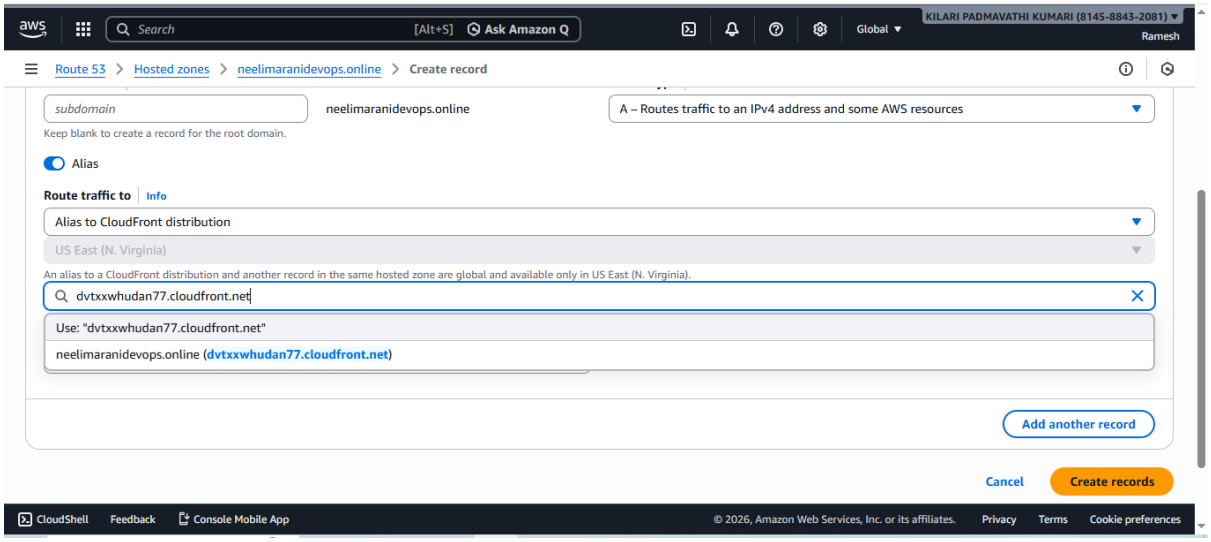


- Get TLS certificate
- Select Available certificate
- Click on create distribution.

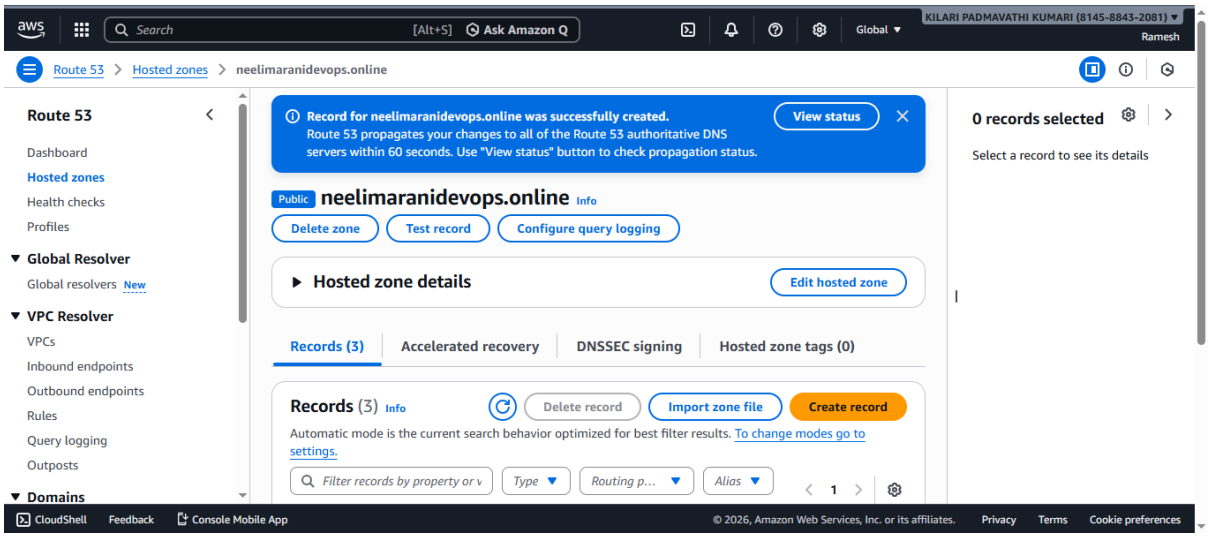


- The above image shows distribution created successfully.

CloudFront & Route 53-Task



- Go to Route 53 →Hosted zones →neelimaranidevops.online
- Route traffic to Alias to cloudfront distribution.
- Add cloudfront distribution ID
- Click on create records



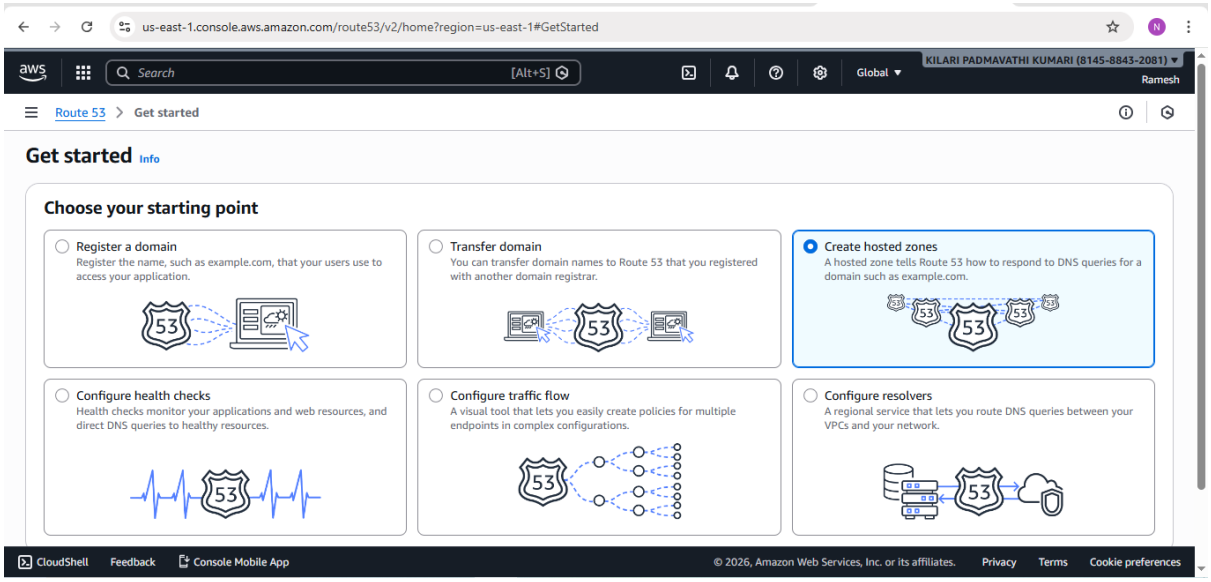
- The above image shows Records for domain was successfully created.

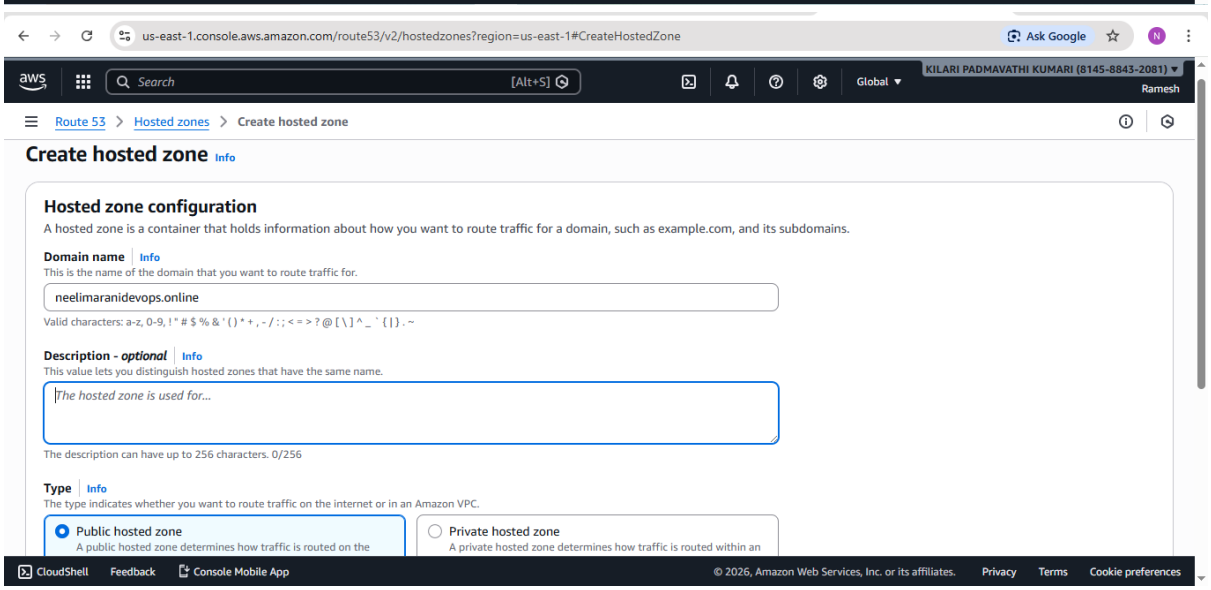
CloudFront & Route 53-Task

5. Create a Route 53 hosted zone and map the domain with the CDN.

- Go to:

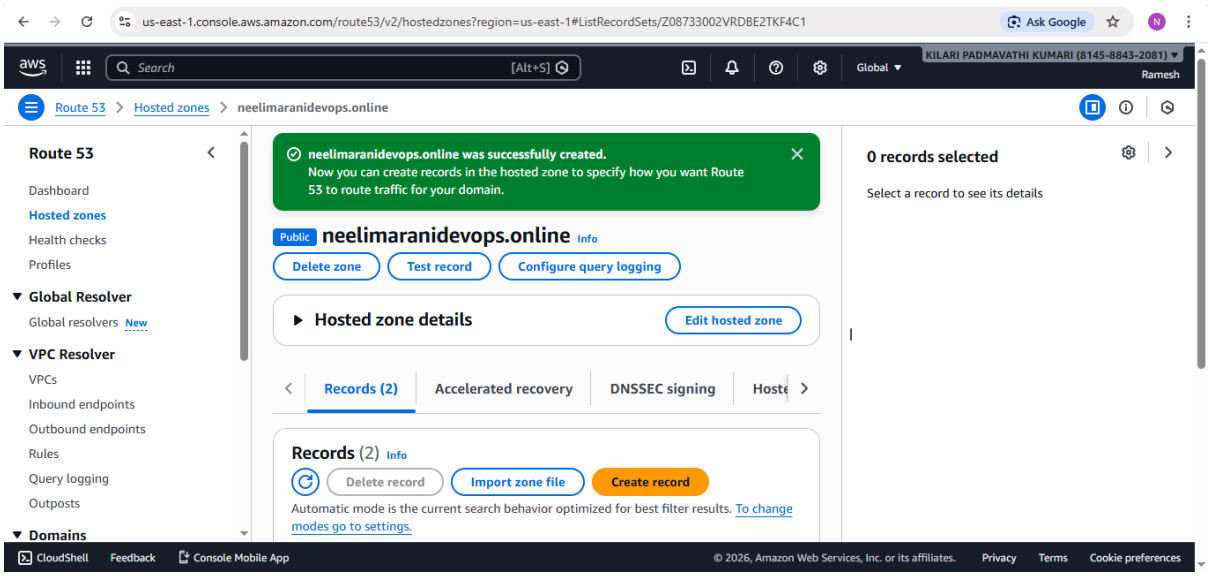
  AWS → Route 53 → Hosted Zones → Create hosted zone



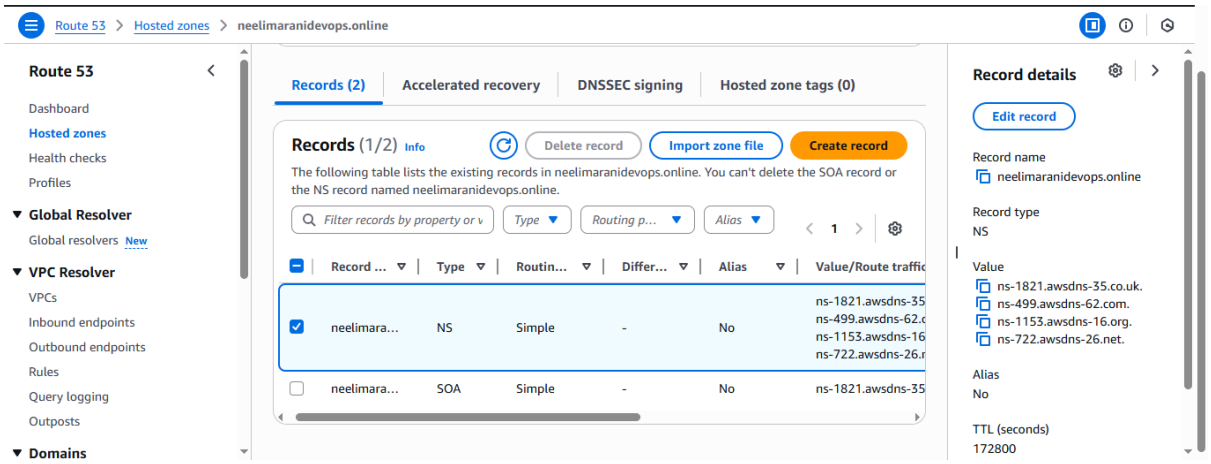


- Add domain name as neelimaranidevops.online
- Type: public hosted zone

CloudFront & Route 53-Task

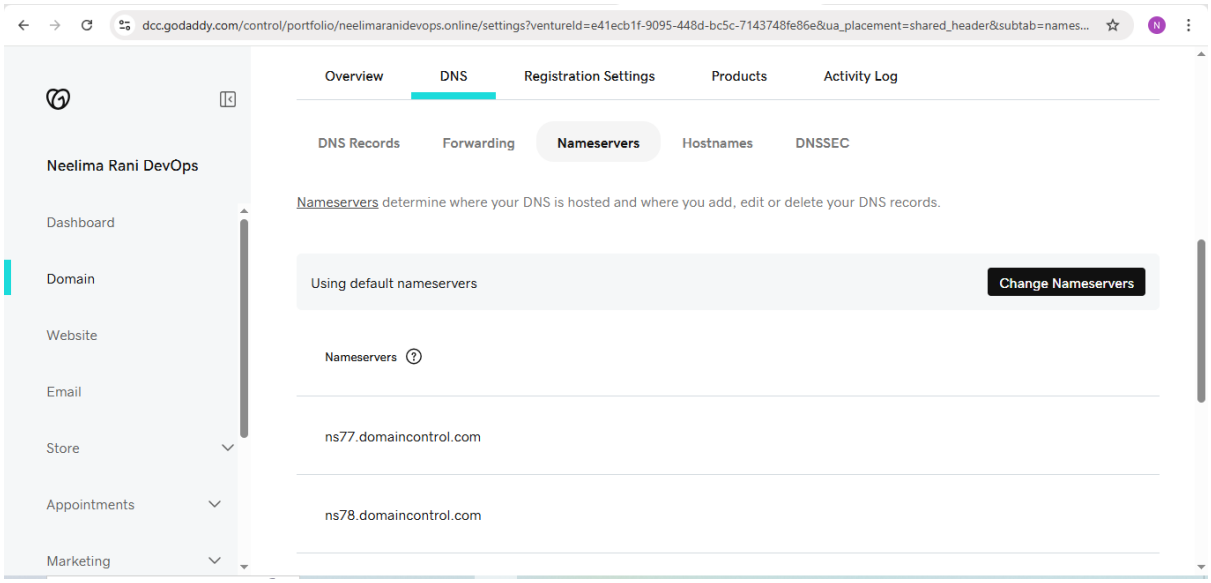


- Click on create zone.
- The above image shows create records in the hosted zone to specify how you want to route 53 to route traffic for your domain.

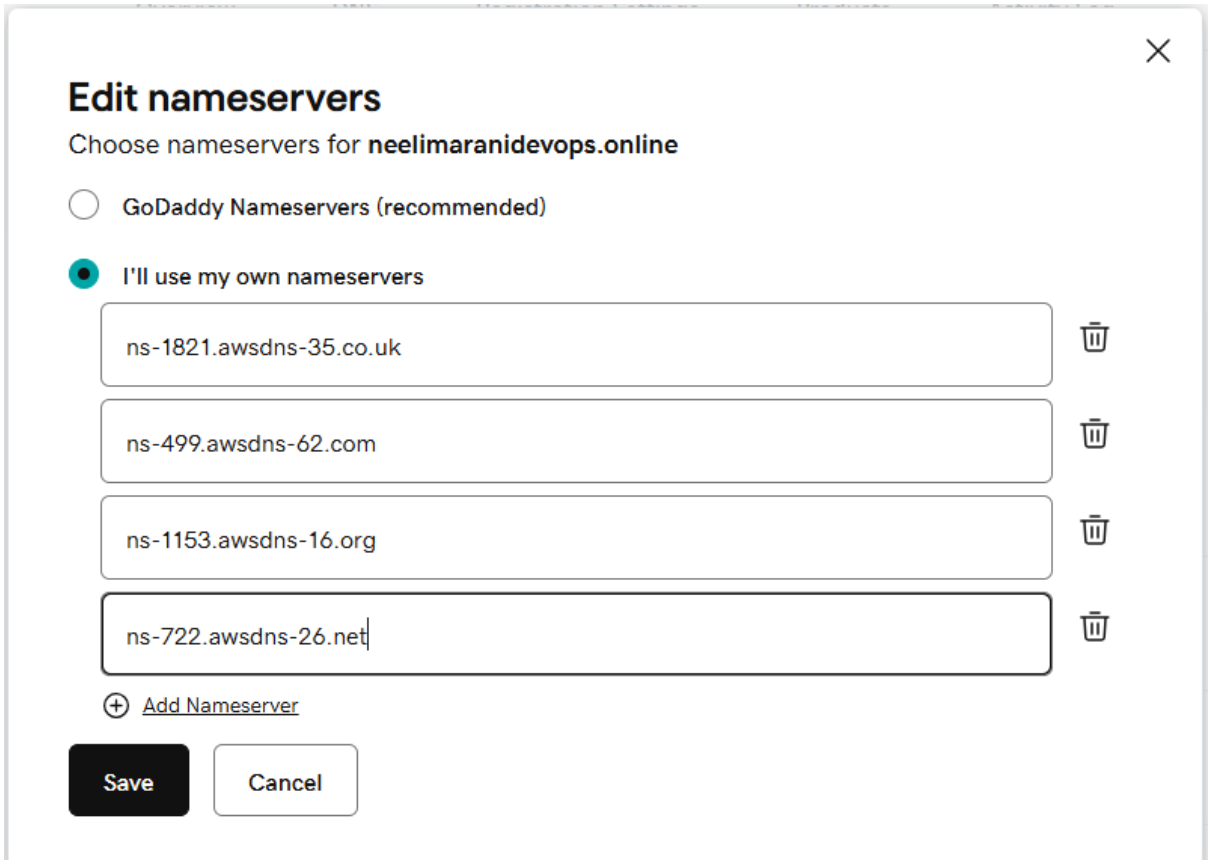


- Go to route→hosted Zones→ neelimaranidevops.online
- Click on record details in that we can see namesevers.

CloudFront & Route 53-Task



- Now go to go daddy website in that domain →DNS→namesevers
- Click on change namesevers.



- Add namesevers which are present in route53.

CloudFront & Route 53-Task

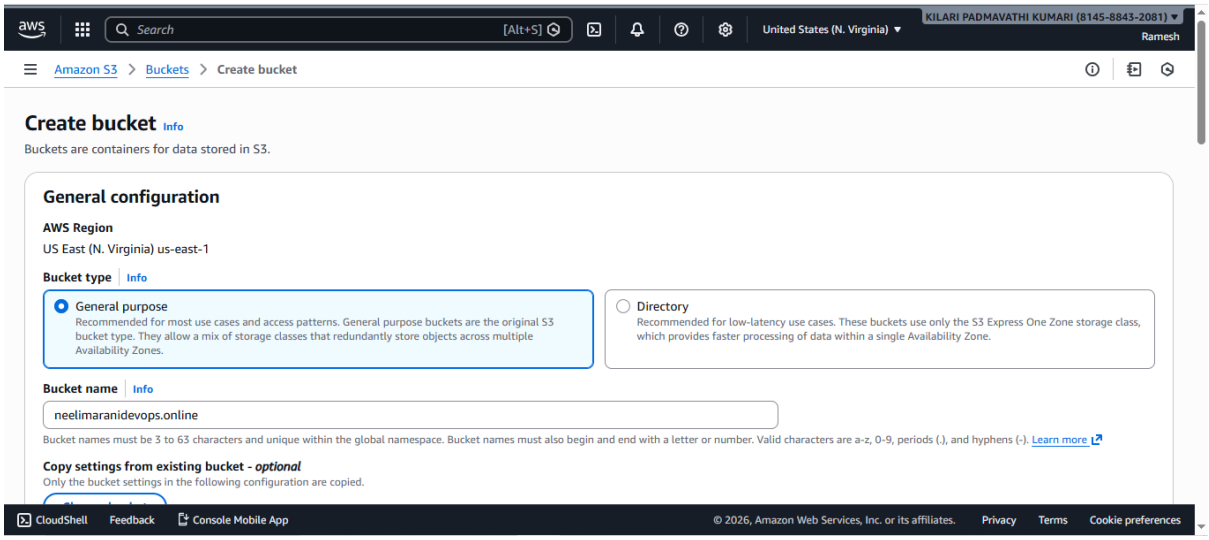
```
user@DESKTOP-3KH1IRE MINGW64 ~ (master)
$ nslookup -type=NS neelimaranidevops.online
Non-authoritative answer:
Server:  gpon.net
Address:  fe80::1

neelimaranidevops.online        nameserver = ns-1153.awsdns-16.org
neelimaranidevops.online        nameserver = ns-499.awsdns-62.com
neelimaranidevops.online        nameserver = ns-722.awsdns-26.net
neelimaranidevops.online        nameserver = ns-1821.awsdns-35.co.uk
```

Conclusion :To verify where they are added or not

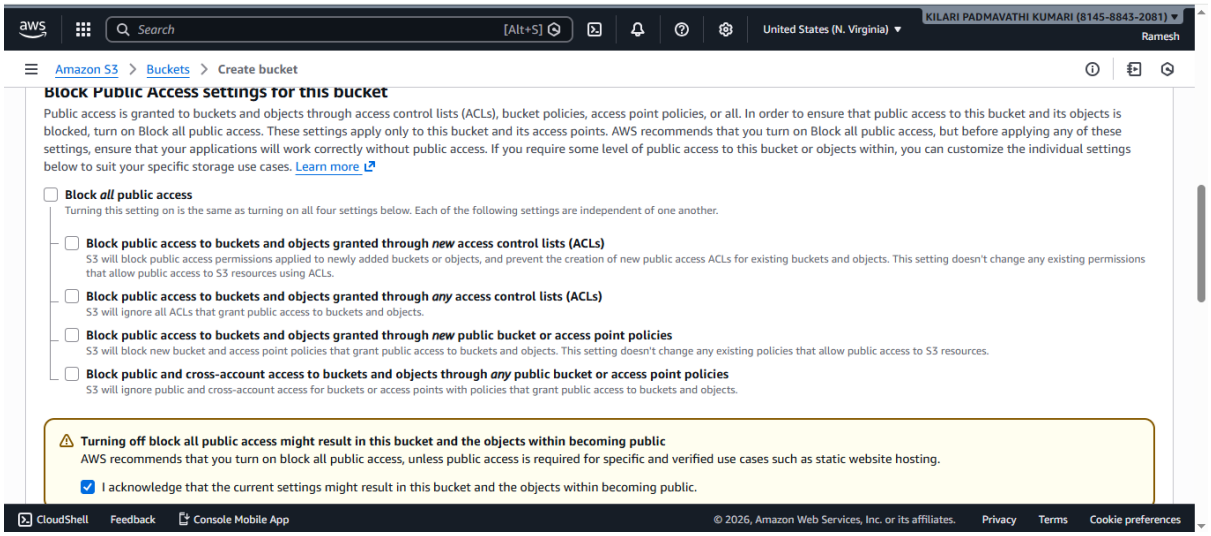
Command:- Nslookup -type=NS neelimaranidevops.online

6. Update the index.html in the S3 bucket and ensure the updated file is accessible using the domain name.

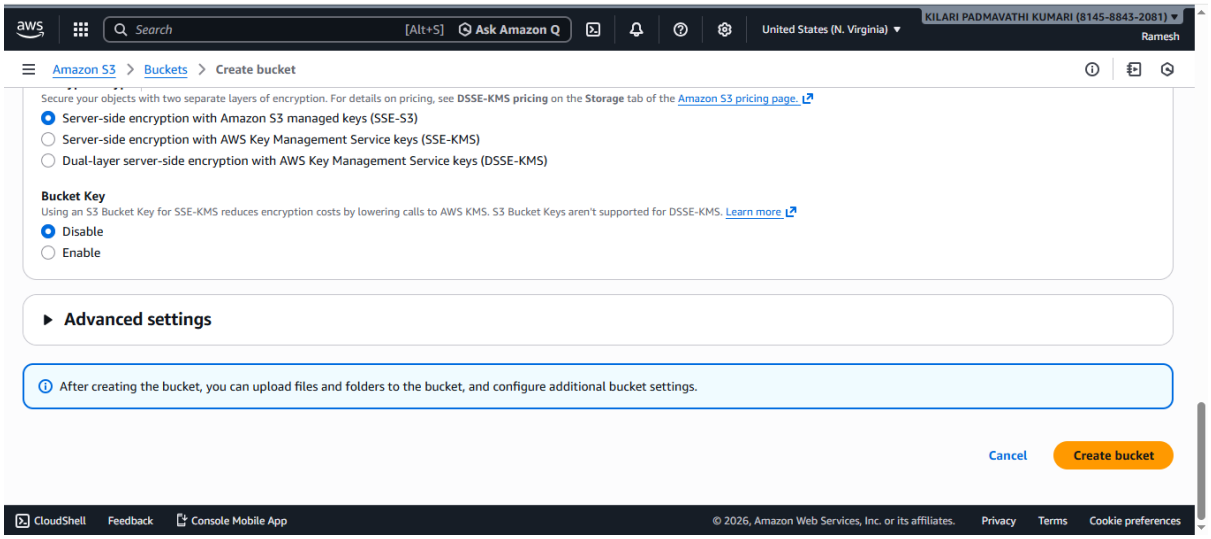
- Go to Aws console →S3→create bucket
- Bucket name should be same as domain name.



CloudFront & Route 53-Task

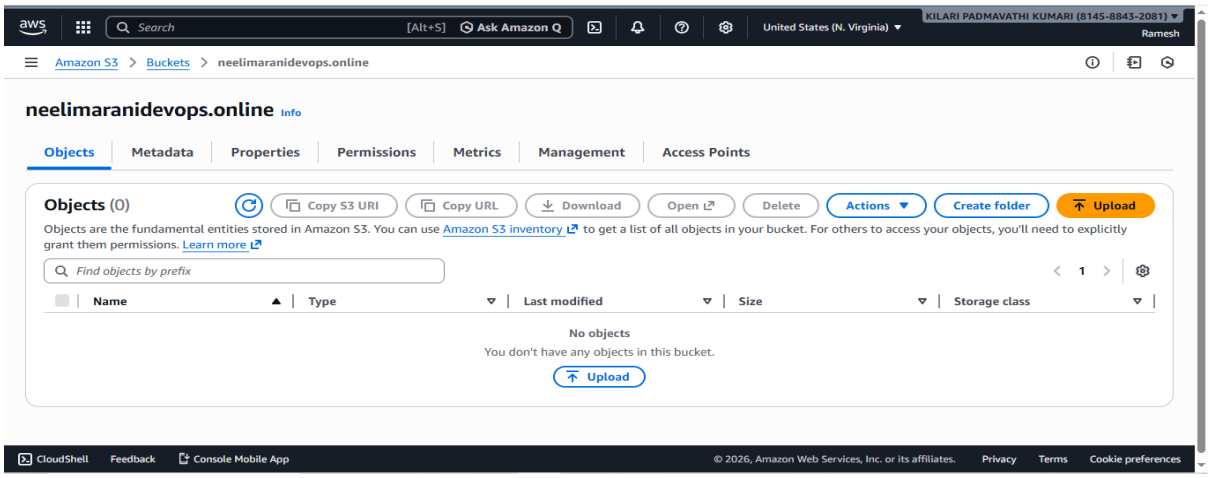


- Block public Access settings for this bucket
- Click on confirmation.

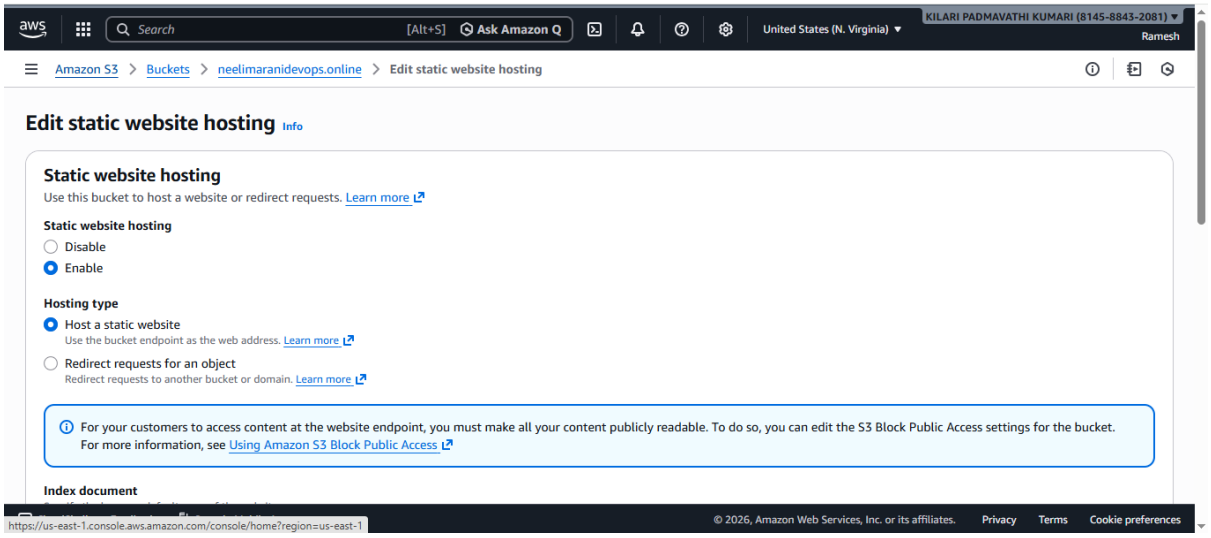


- Click on create bucket

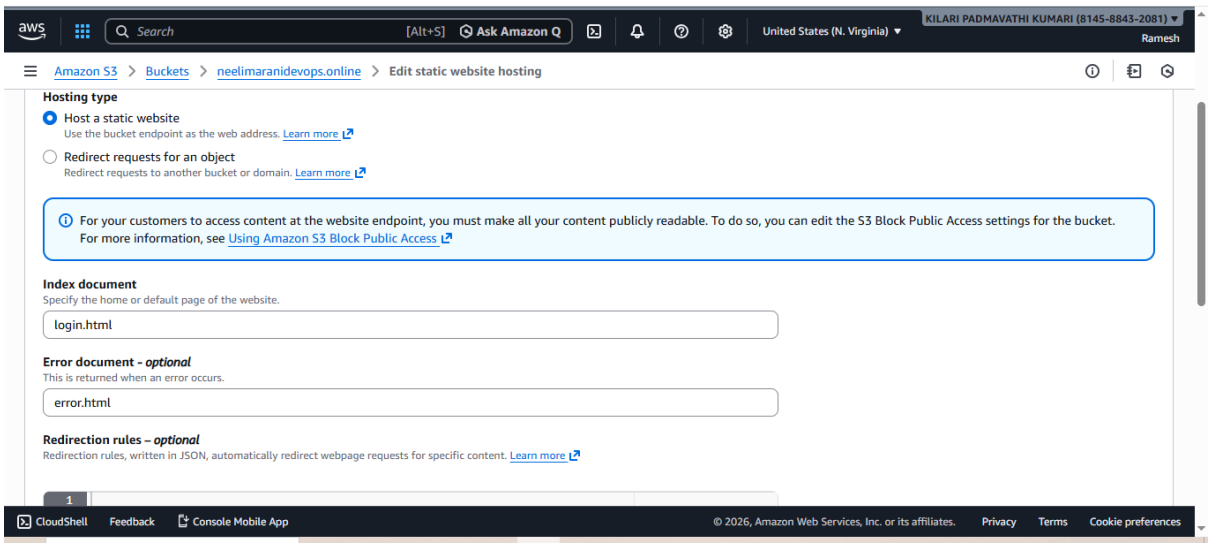
CloudFront & Route 53-Task



- Go to Bucket and click on objects
- Upload files into bucket.

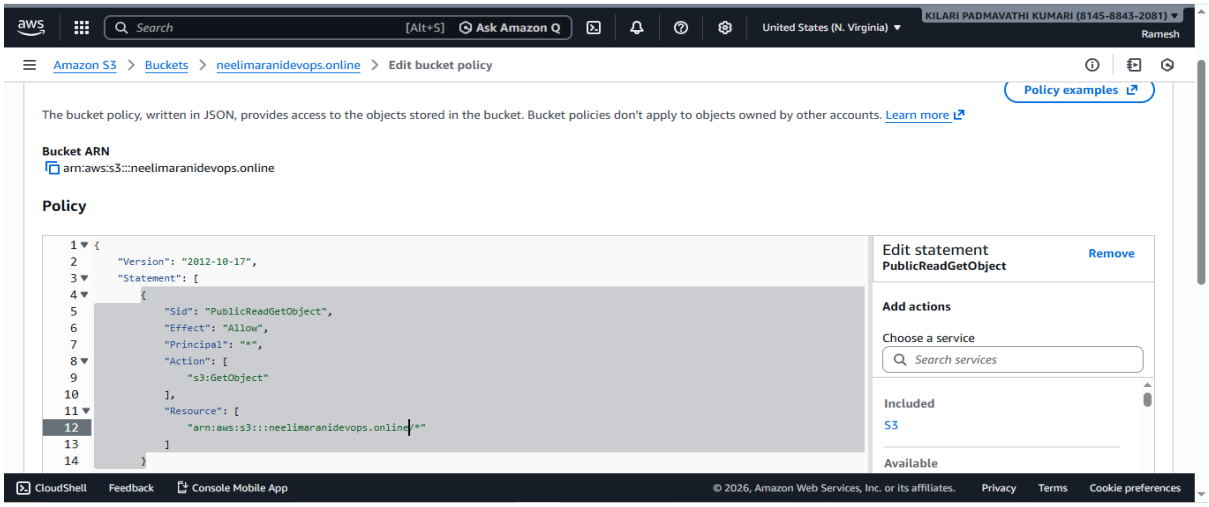


- Static website hosting should be enable



CloudFront & Route 53-Task

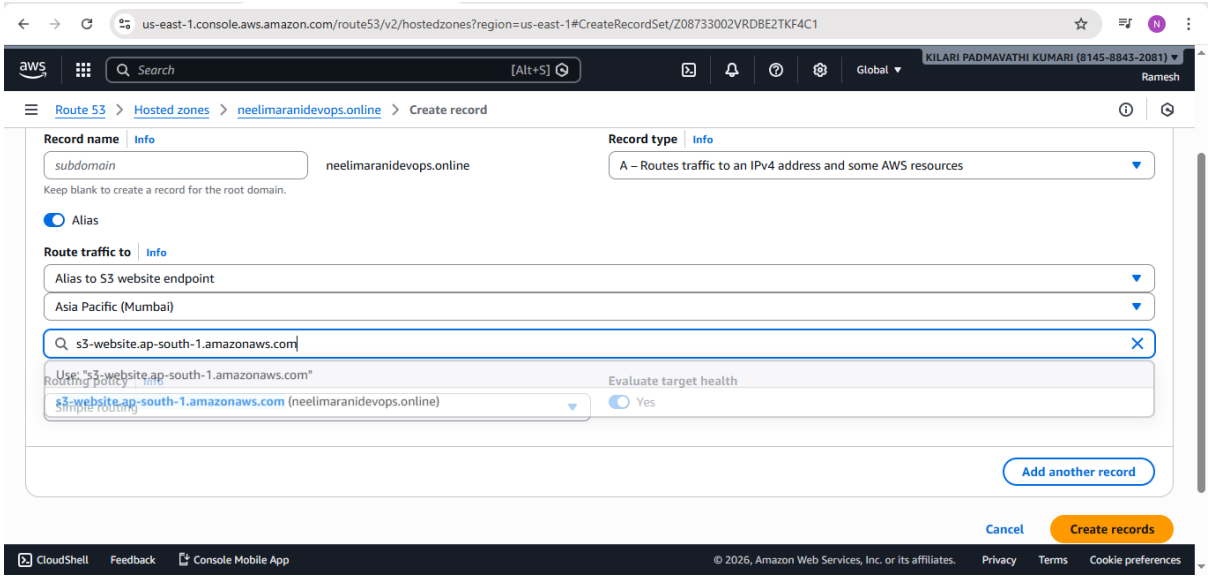
- Hosting type host a static website.
- Add Index document.

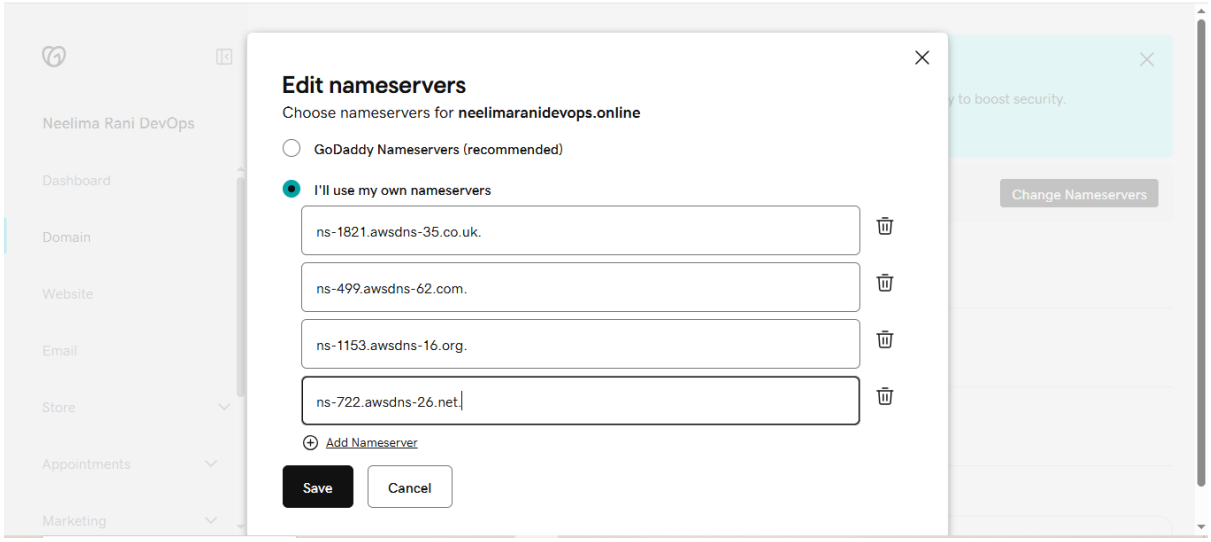


- Edit bucket policy
- Add policy to bucket
- By using bucket URL we can access our static website as shown below.

CloudFront & Route 53-Task

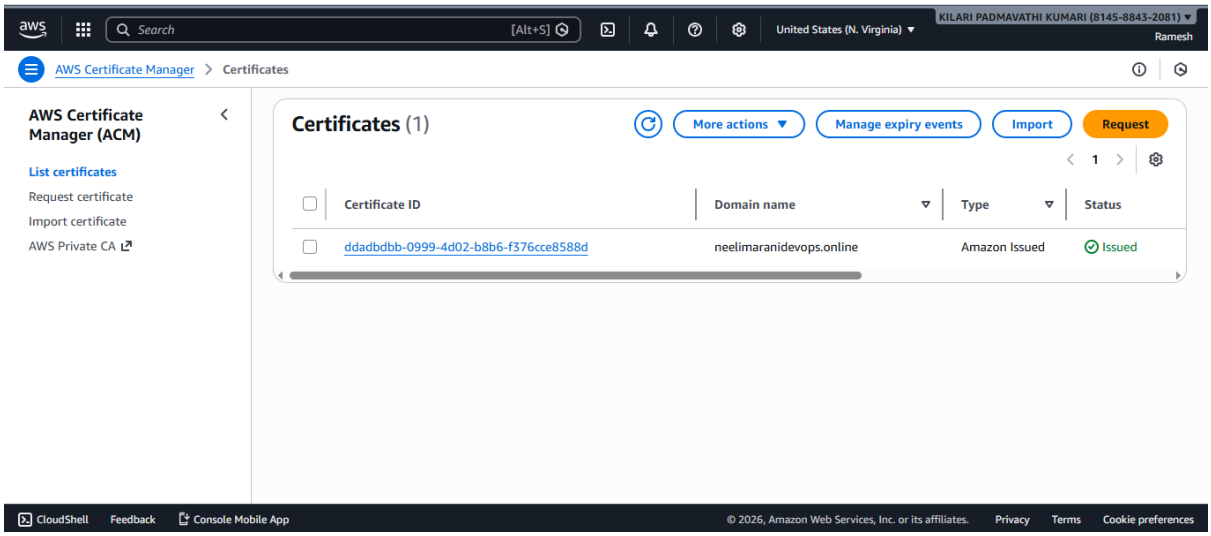
- Now Go to Route53 →hosted Zones→neelimaranidevops.online
- Create Record.
- Click on Alias Route traffic to Alias to s3 website endpoint.
- Click on create Records.



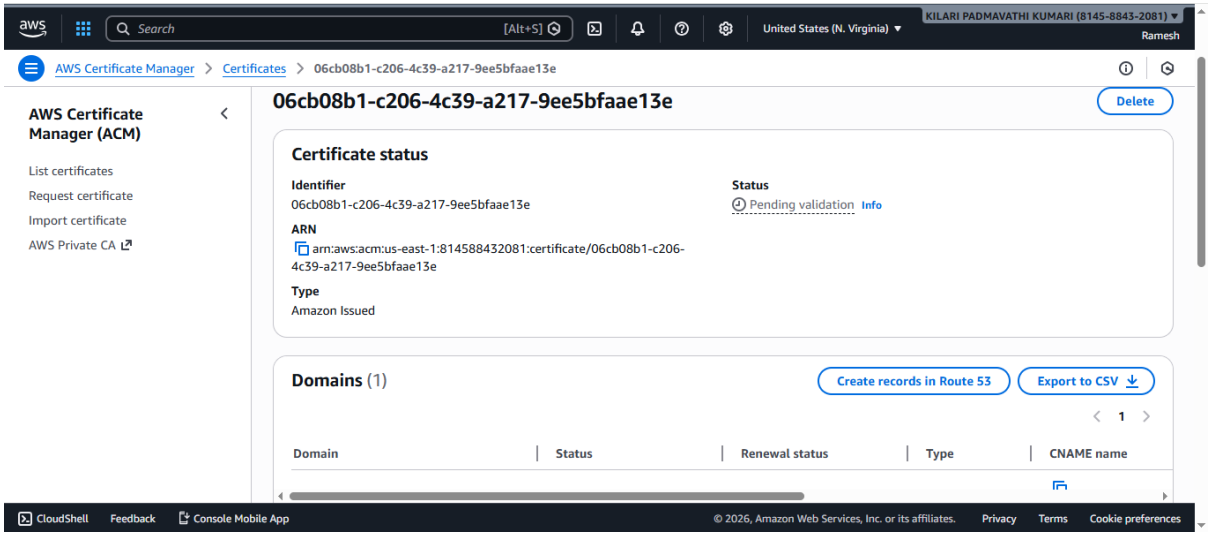


- Go to godaddy website and domain→DNS→ add namesevers to it

CloudFront & Route 53-Task

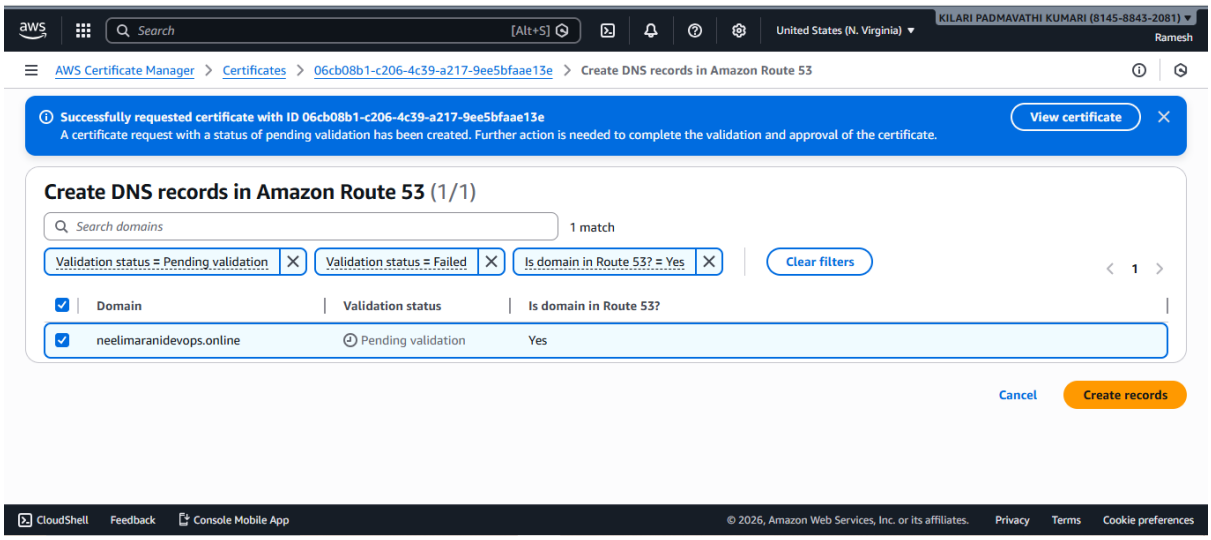


- Go to Aws certificate Manager→certificates→request

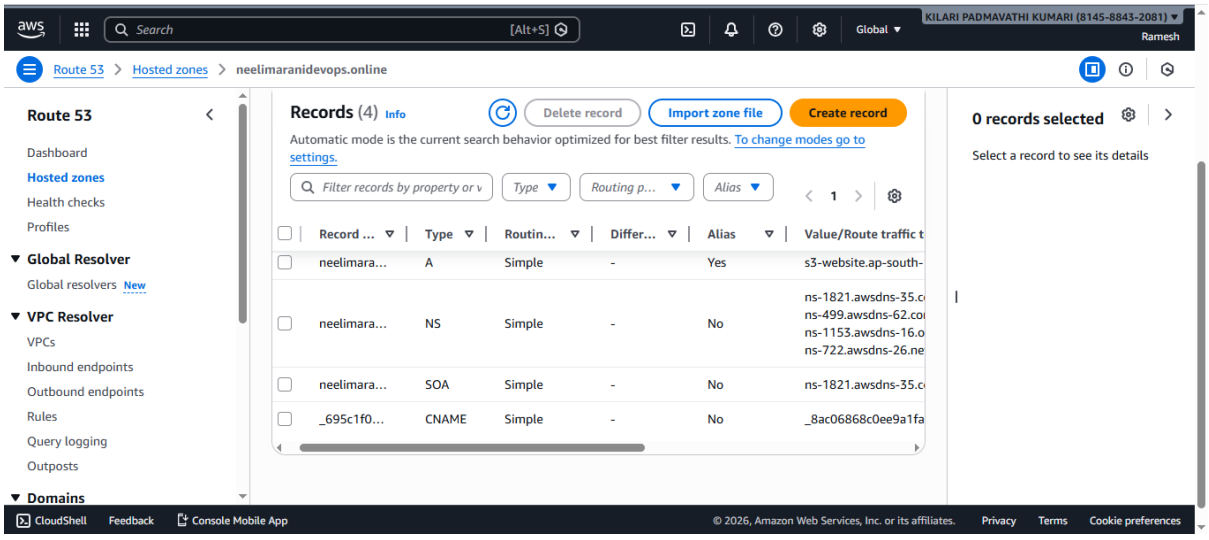


- Request a certificate to our domain.
- Click on Record in route53

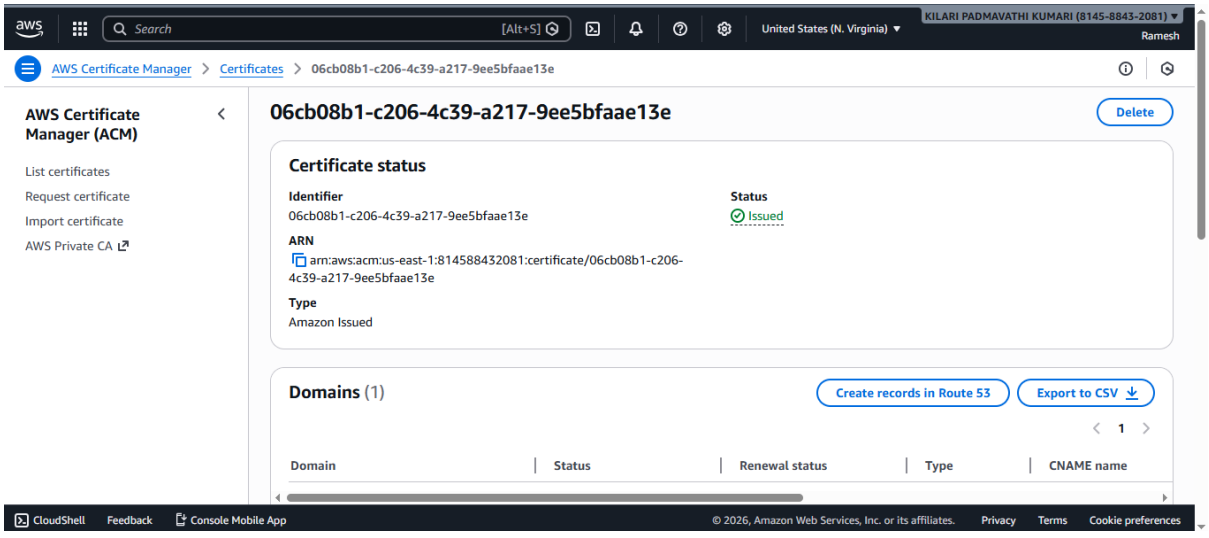
CloudFront & Route 53-Task



- After clicking on create record

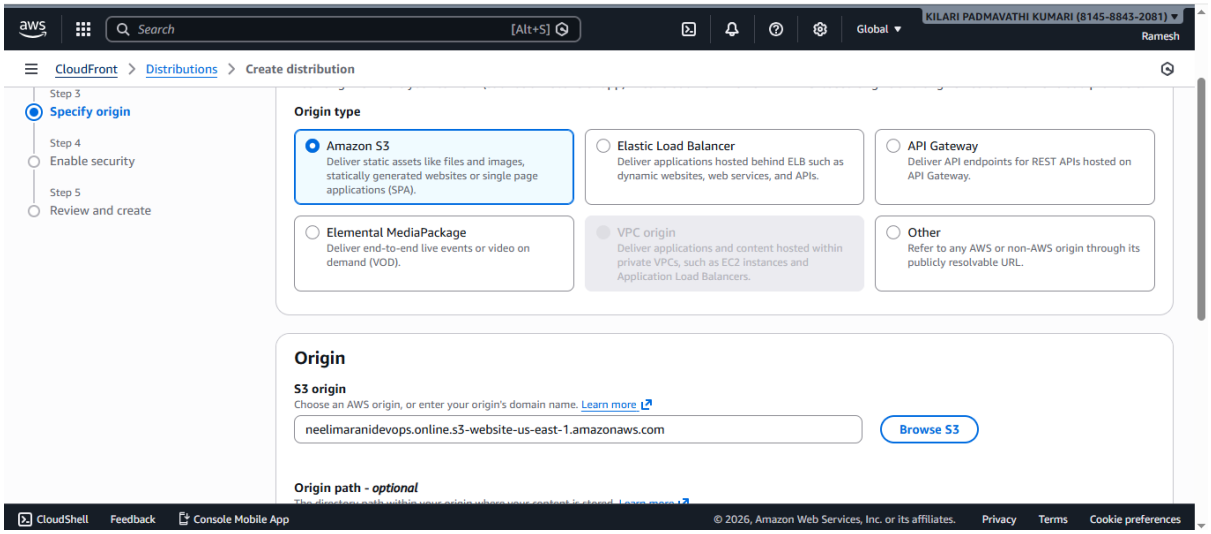


Check in Route53 →hosted zones→neelimaranidevops.online→A record is added.

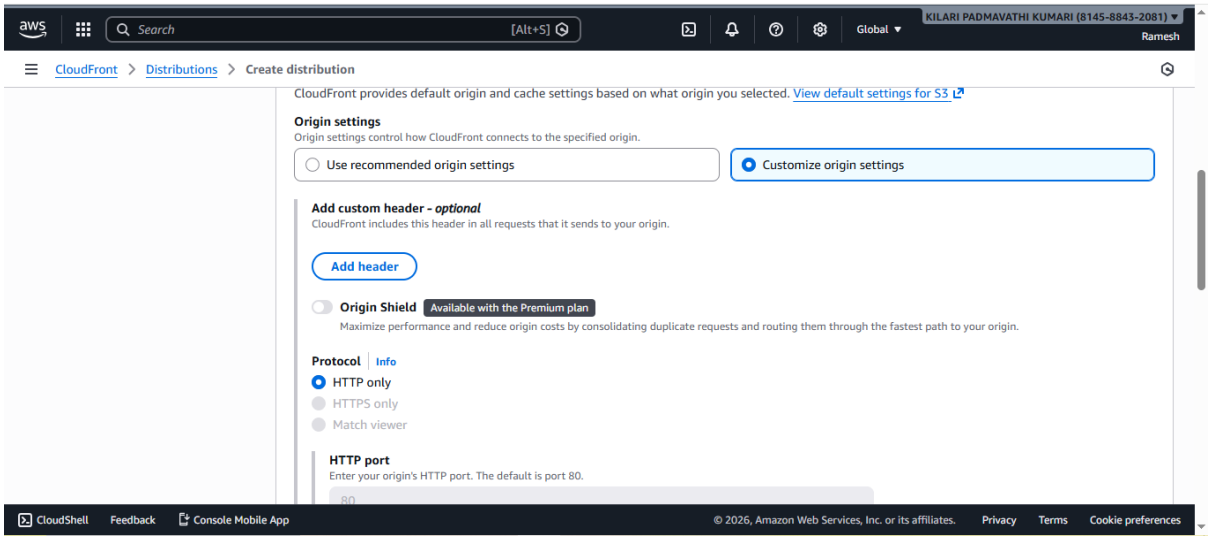


- Then in AWS certificate status shown as **issued**

CloudFront & Route 53-Task

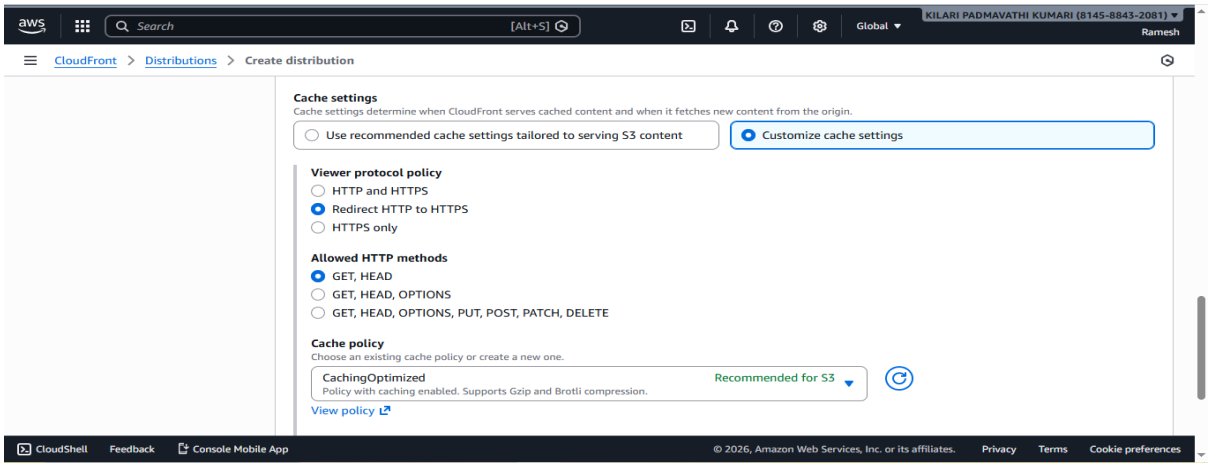


- Go to cloudfront →distributions→create distributions.
- Origin type as Amazon s3.

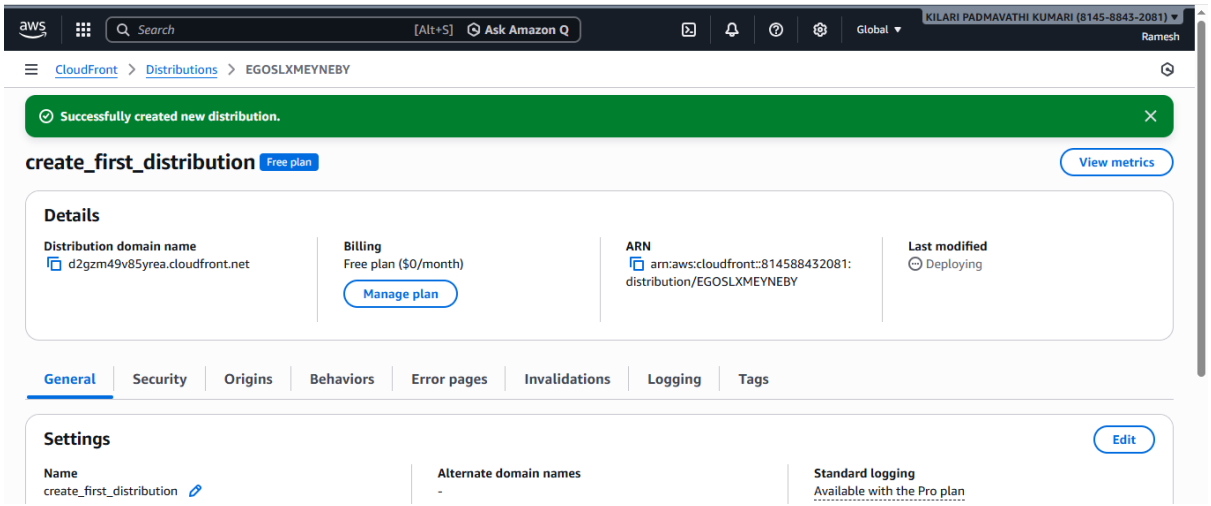


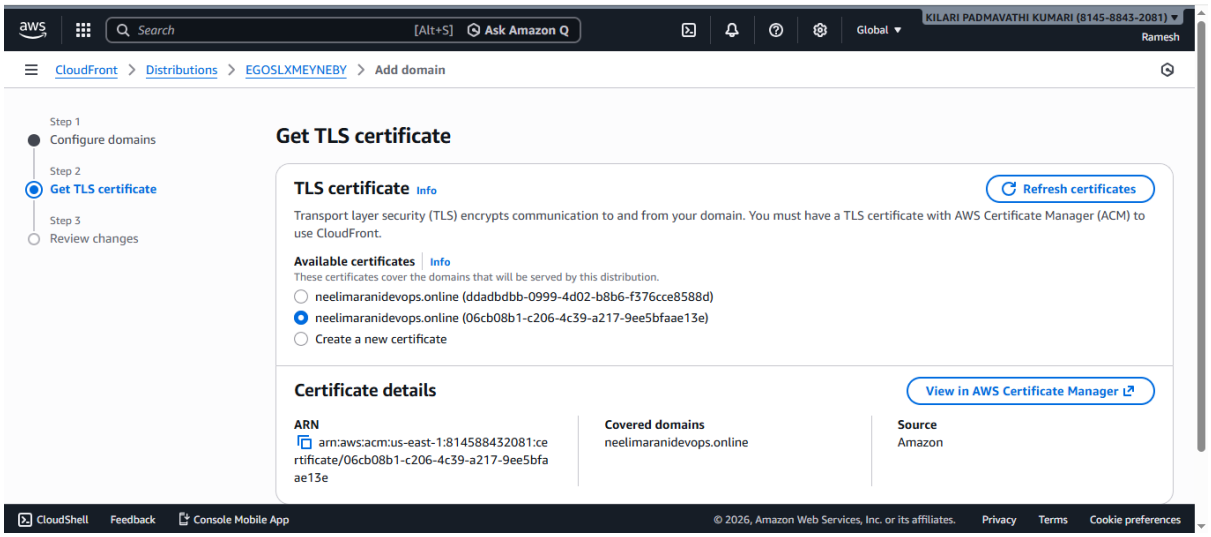
- Origin settings customize origin settings
- Protocol as HTTP only.

CloudFront & Route 53-Task



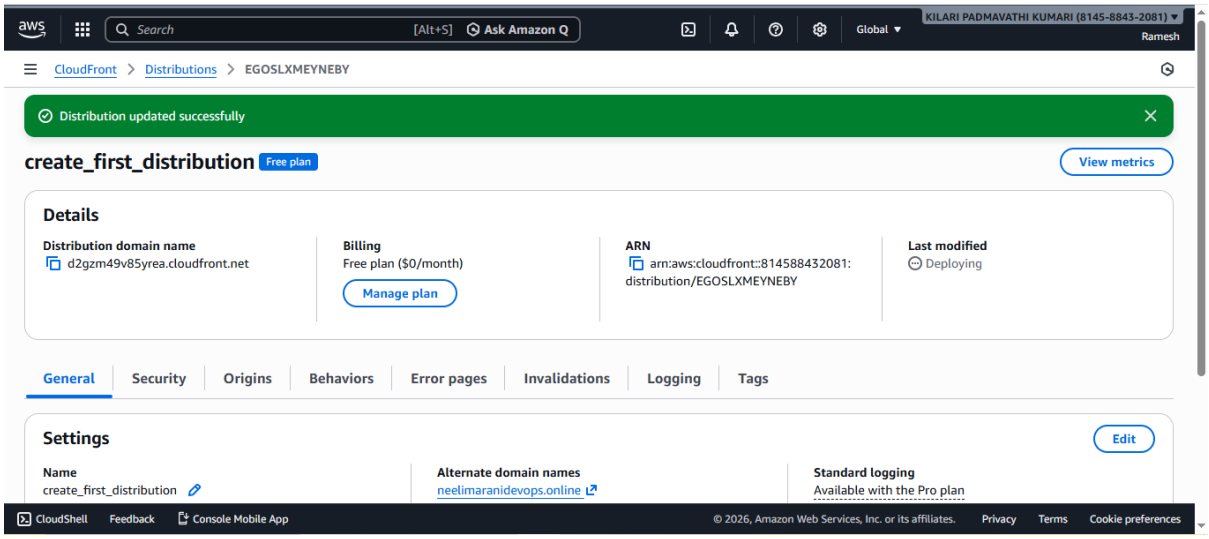
- Cache setting →customize cache settings
- Viewer protocol policy redirect HTTP to HTTPS,
- Allowed HTTP methods GET, HEAD
- Click on Create distribution



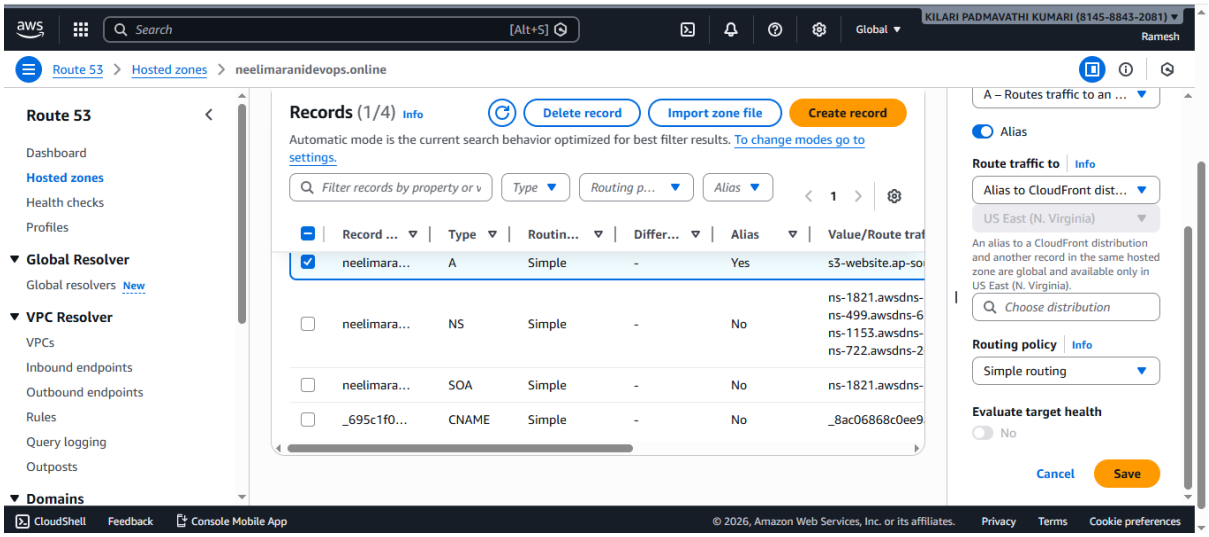


CloudFront & Route 53-Task

- Click on Distribution Id → Add domain
- Get TLS certificate
- Choose certificate issued for our domain.

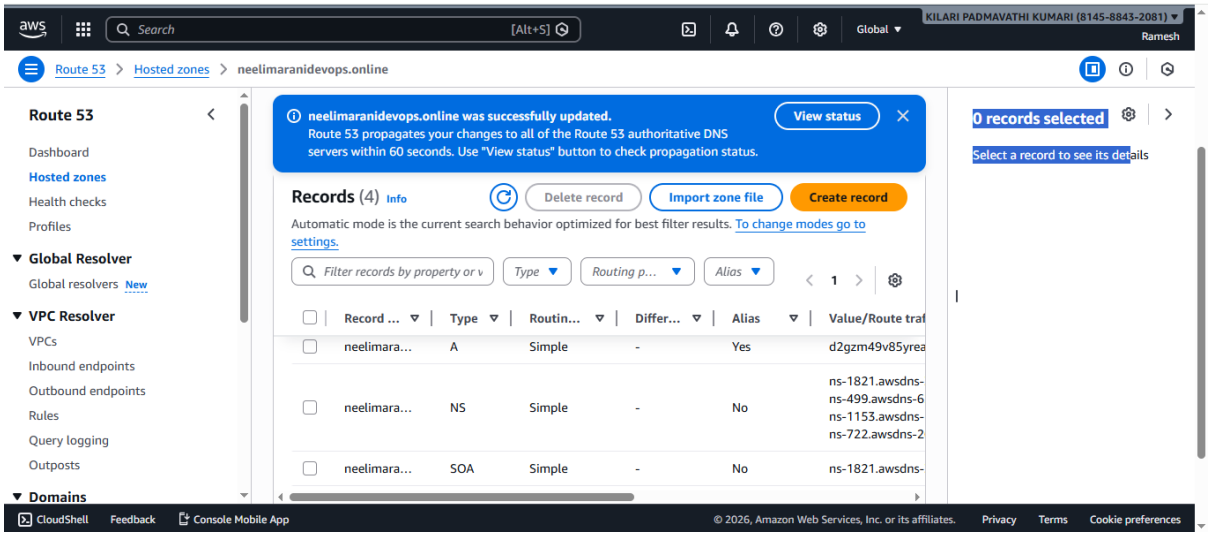


- Add Alternate Domain names→neelimaranidevops.online



- Go to route 53
- Click on A record edit record route traffic to Alias to cloud front distribution instead of s3 bucket endpoint.
- Click on save.

CloudFront & Route 53-Task



- Here we can see update record successfully.

- Now we can access our website with our domain name.

7. Share the domain name in Slack to test the connectivity.

Domain name: neeelimaranidevops.online