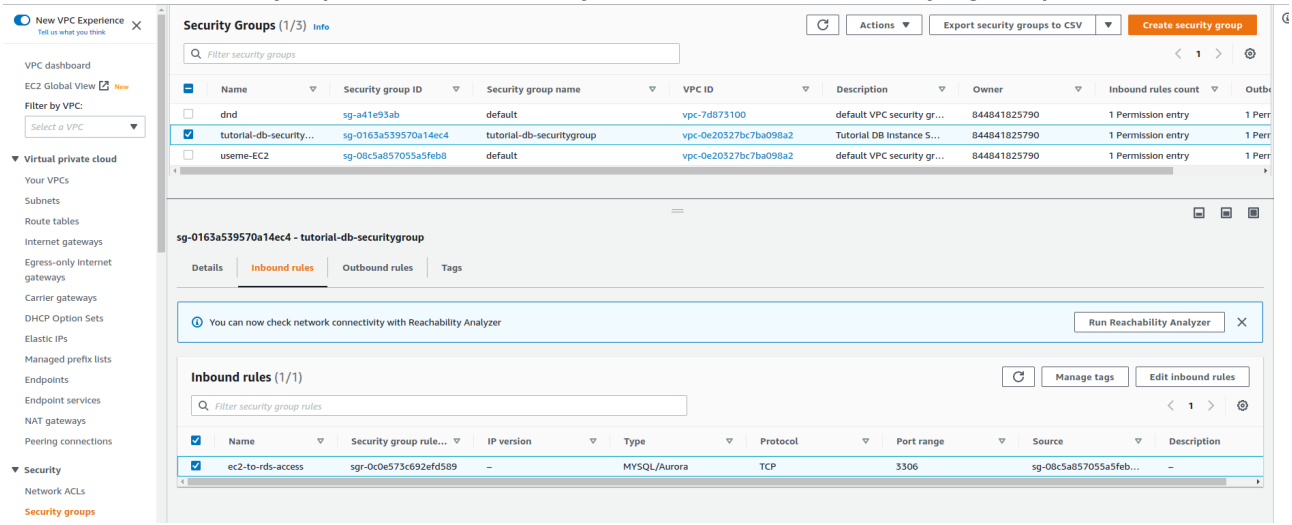
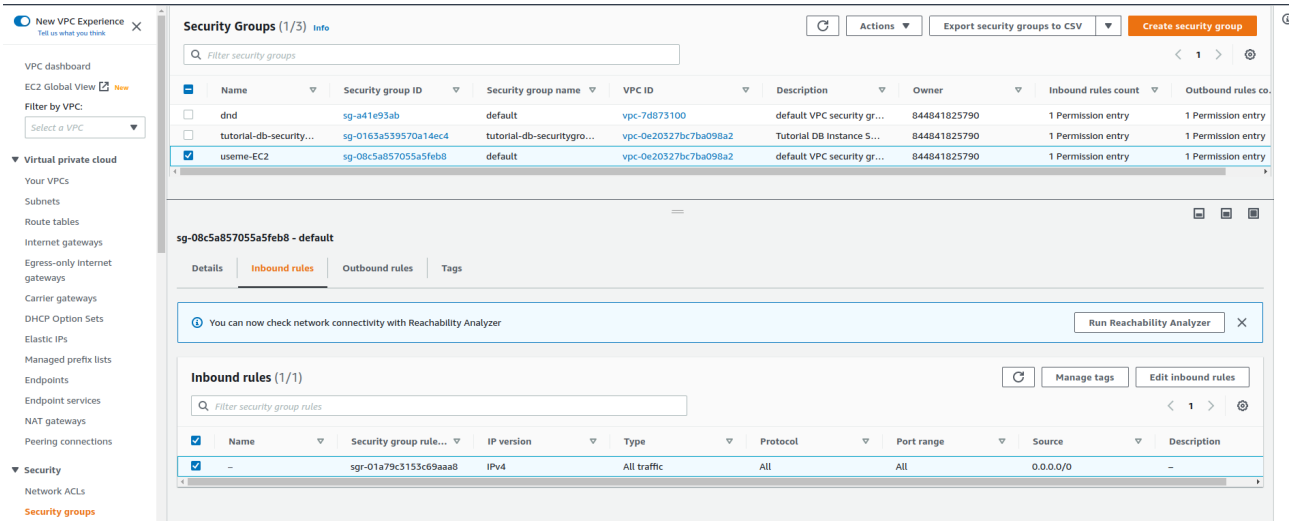
RDS RESUME/CV project

Step 1:
create vpc and all its components: IGW,RT,subnets-pub-1a,pub-1b
vpc: 10.0.0.0/16
pub-1a: 10.0.1.0/24
pub-1b: 10.0.2.0/24

create a security group for you RDS instance, and in Inbound rules: allow mysql traffic only from ec2-Security-group.



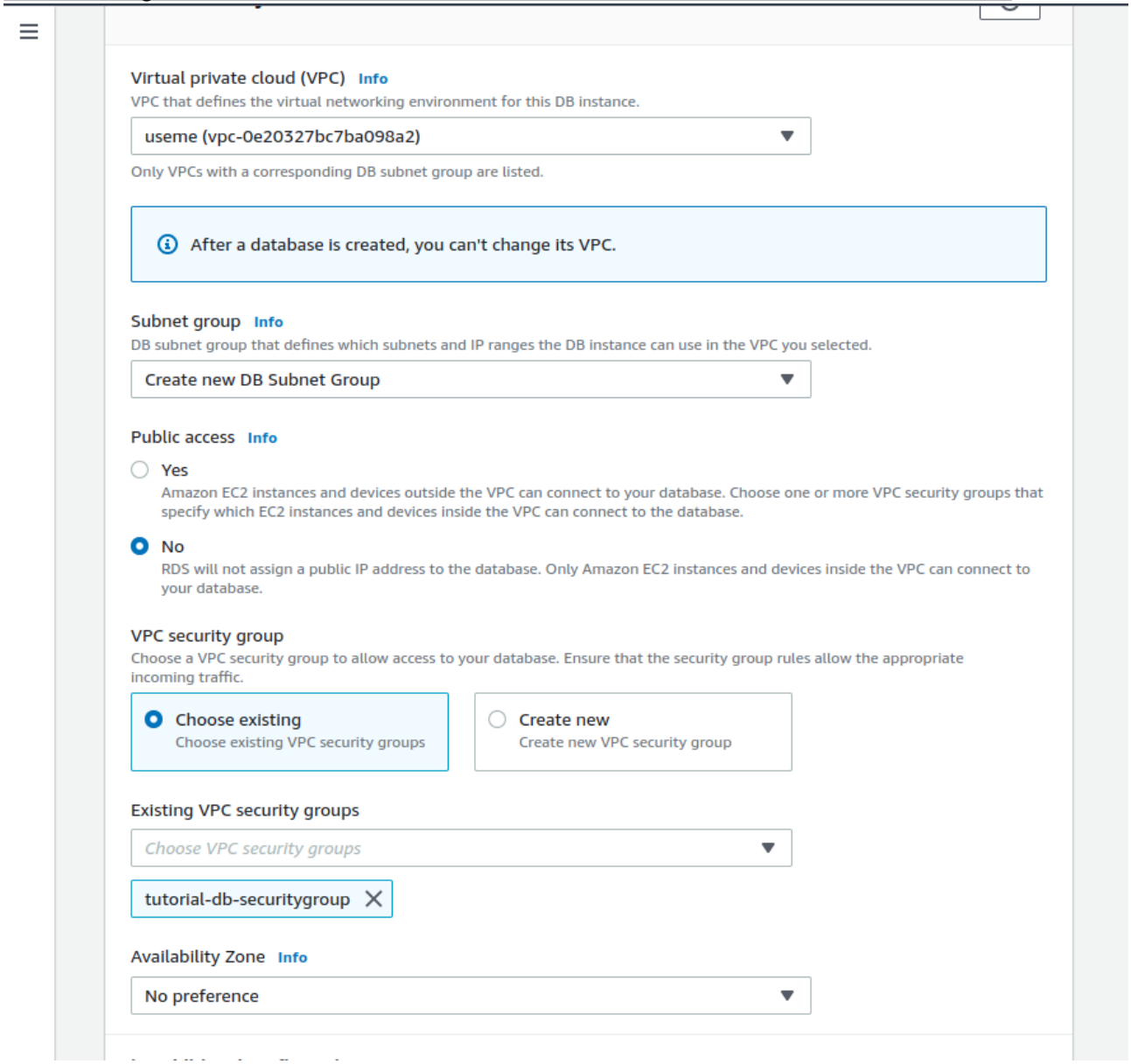
In Inbound Rules: Allow all traffic from anywhere to for use in EC2.



Step 2:
Create a public and private bucket in S3. And upload one images to each. Pub image: your photo.

Step 3:
configure RDS
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateDBInstance.html
follow through: 1. Create a DB Instance , 2. Create a web server

Following is the VPC conf. At RDS Database creation console:



Step 4:

Now Launch an EC2 by hand/manually. And set it up to host SamplePage.php. We will test it and once everything is up and running, we will take an AMI snapshot out of that EC2 Instance state.

Use the bootstrap scipt to while launching EC2:

```
#!/bin/bash
sudo yum -y update
sleep 5
sudo yum -y install httpd
sleep 10
sudo amazon-linux-extras install php8.0 mariadb10.5 -y
sleep 30
sudo systemctl restart httpd
sleep 4
sudo systemctl enable httpd
```

```
sleep 4
sudo mkdir /var/www/inc
sleep 2
```

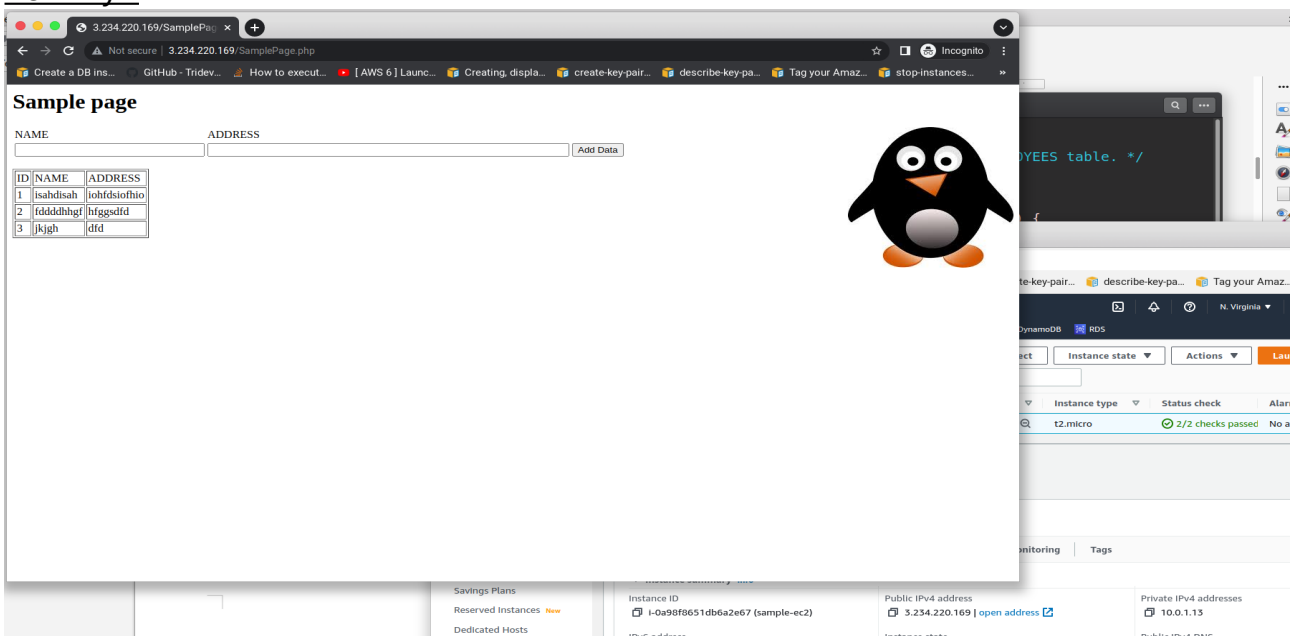
Login to instance and make the neccessary changes:

```
# cat > /var/www/inc/dbinfo.inc
<?php

 define('DB_SERVER', 'tutorial-db-instance.cnekvosk6jnc.us-east-
1.rds.amazonaws.com');
 define('DB_USERNAME', 'tutorial_user');
 define('DB_PASSWORD', 'admin123');
 define('DB_DATABASE', 'sample');

 ?>"
# cat > /var/www/html/SamplePage.php  <-- “Enter the contents from AWS docs” and add img src taag for your S3 public-object url.
```

Verify:



Step 4:

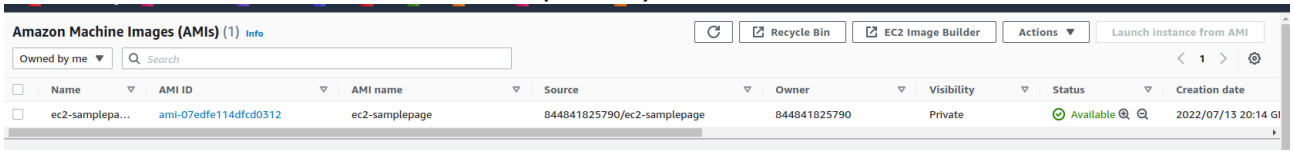
Taking Backups for automated/further use:

Create AMI from instance:

Now create an AMI snapshot of the EC2: instance -> actions -> image and templates -> Create Image

create Launch template from instance:

Now create an AMI snapshot of the EC2: instance -> actions -> image and templates -> create template from instance (conditions: 1. Do not keep AZ, 2. remove the bootstrap script as it is already taken care of in the AMI snapshot)

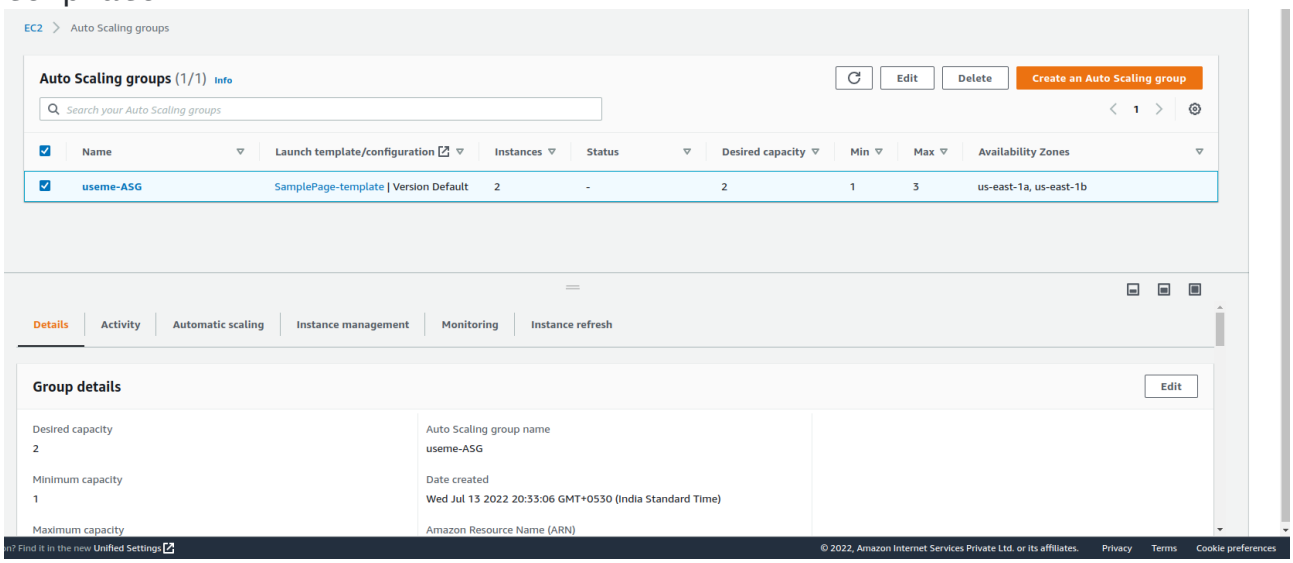


Now,stop/terminate the unneccessary EC2 instance.

Step 5:

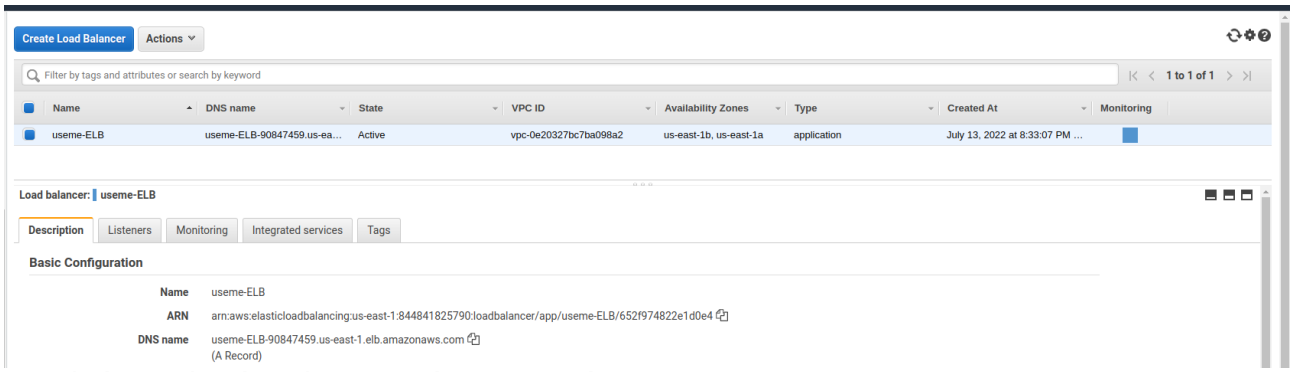
Create AutoScaling Group:

only public subnet - create Auto scaling with ELB, and target group, scale in protection for autoscaling -> off, using launch template

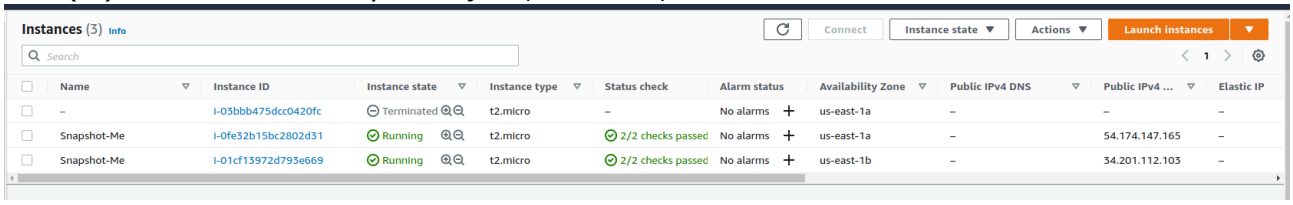


Creating autoscaling group also craete the target group and ELB for me, and launched the instances.

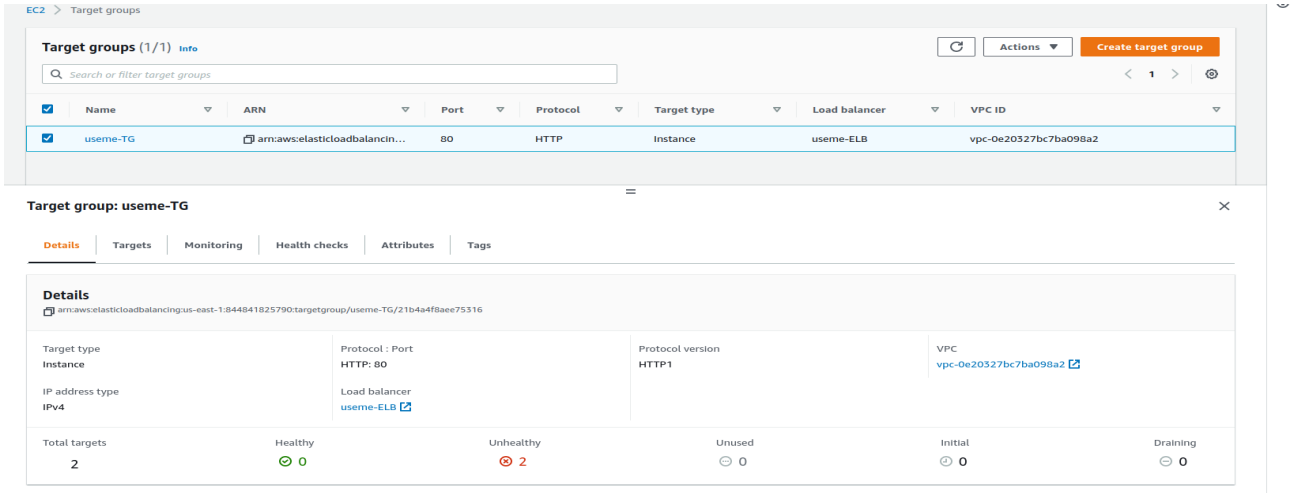
ELB:



EC2(s) – desired capacity=2, min=1, max=3:

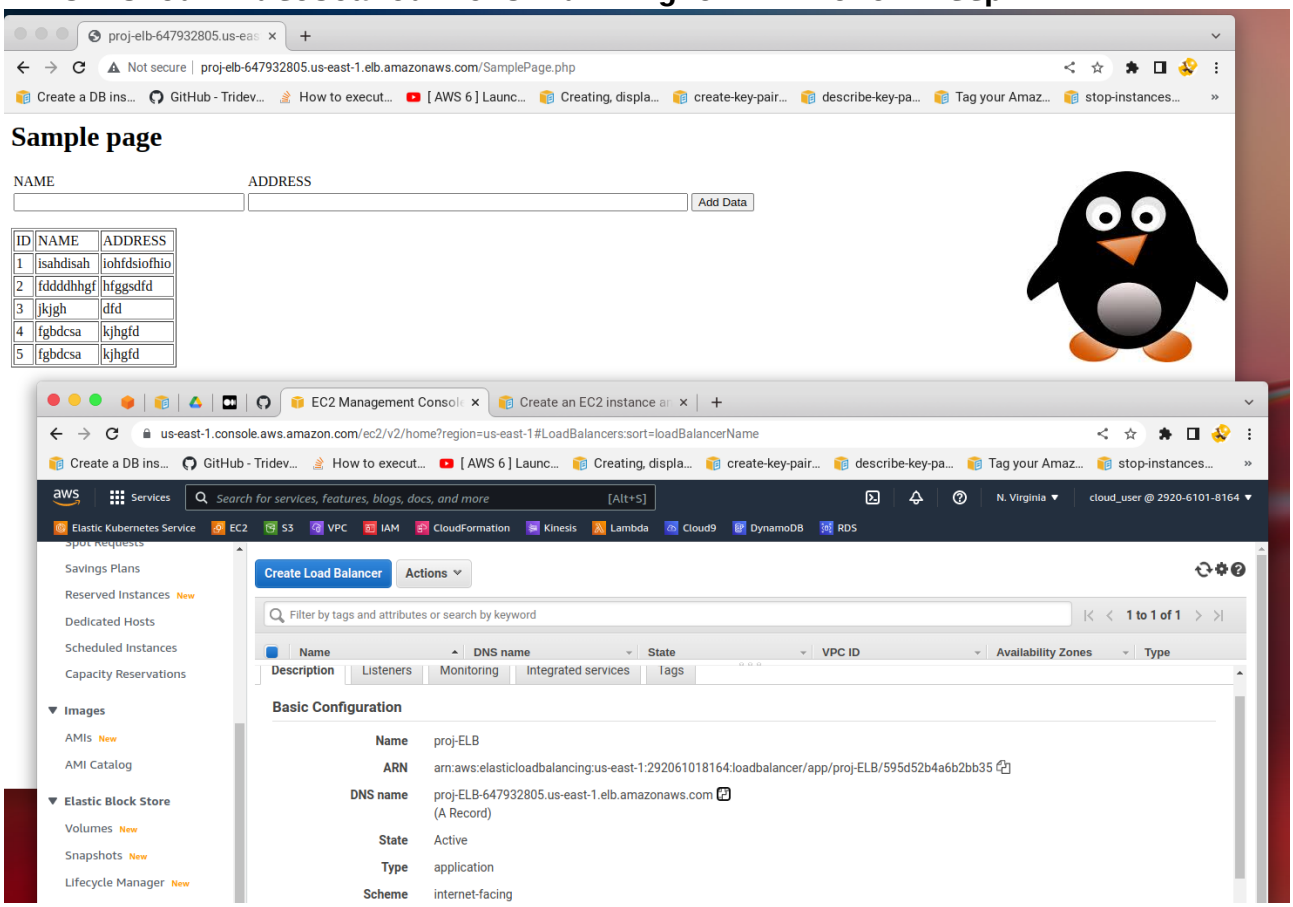


Target Group:



Once the desired number: 2 of ec2-instances are running with all checks passed, and ELB provisioned state: active. Browse, ELB DNS NAME url: http://sample-elb-794607581.us-east-1.elb.amazonaws.com/

This is our Autoscaled-EC2s running on ELB over http.



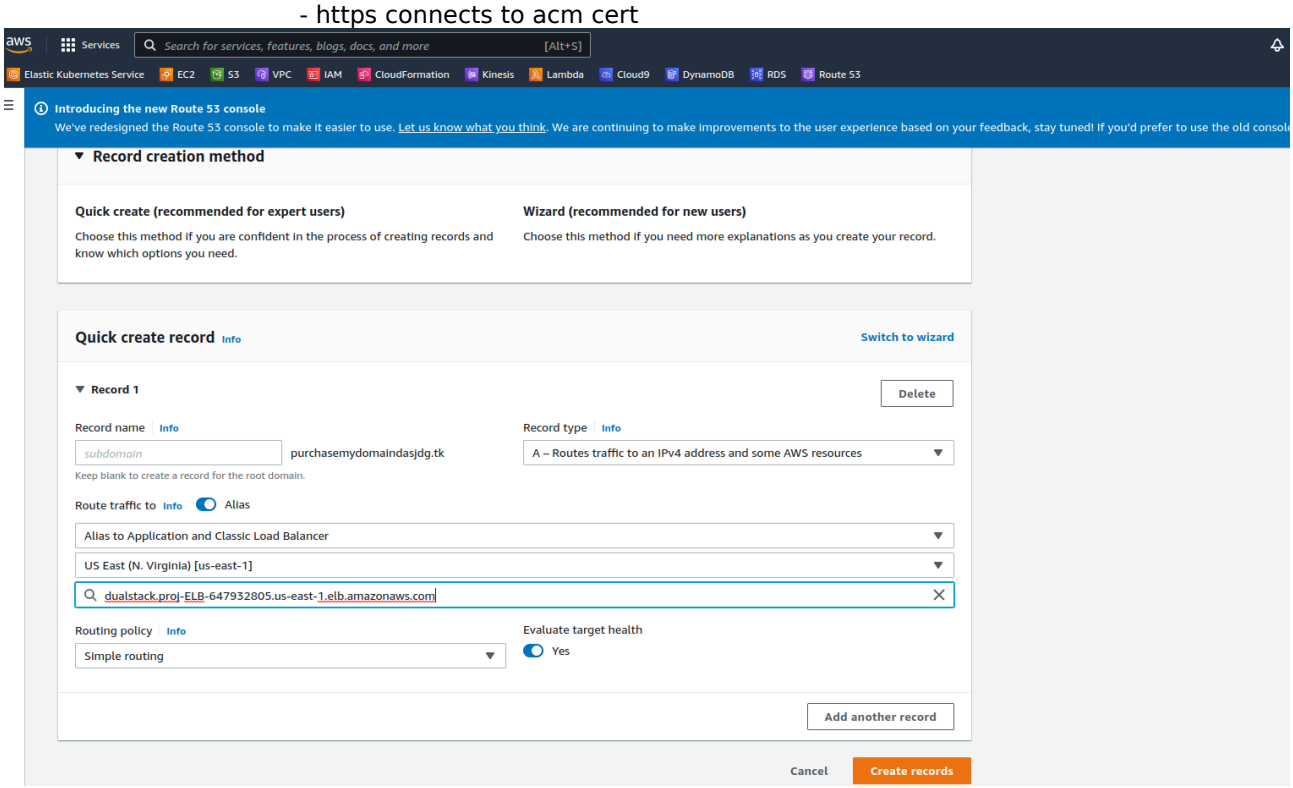
Now, Time to integrate HTTPS and map our freenom domain:

6. For https integration,

in Route53:

- create hosted zone with your full qualified DNS name
- put nameserver details to freenom
- connect record - for elb to hosted zone

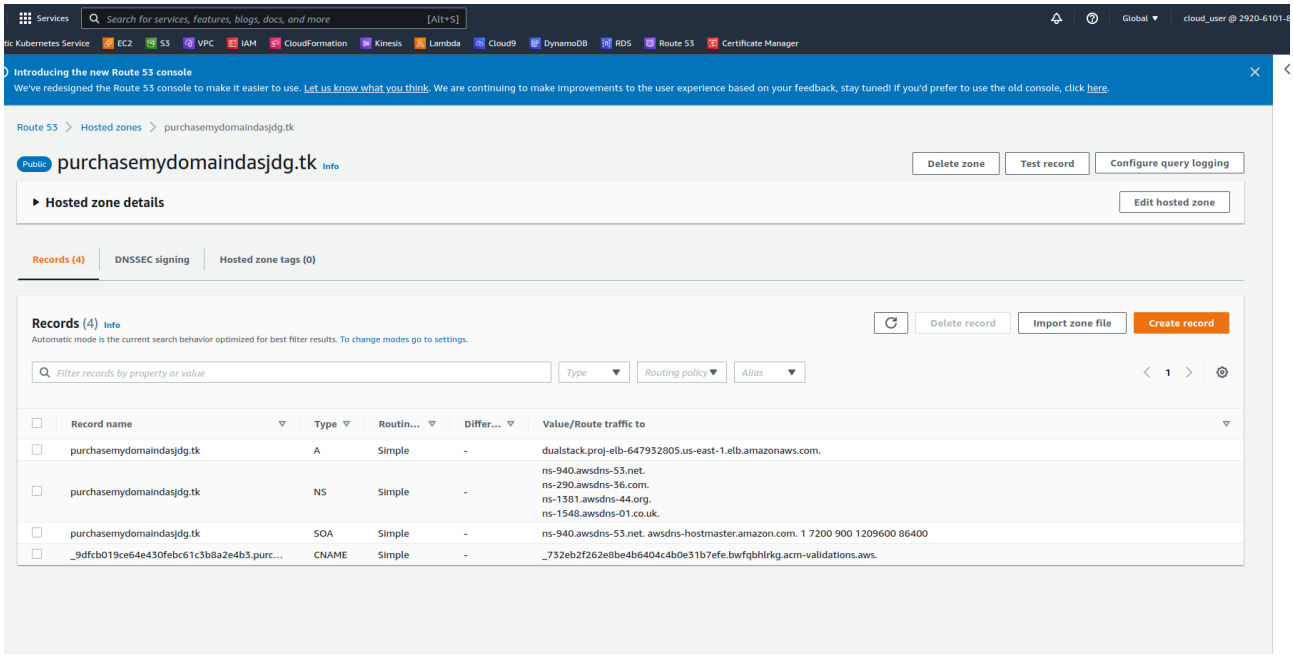
- create ACM cert and and create its cert record to r53
- ELB (listener rule):
- http -> https
- https connects to acm cert



Above: connecting ELB to ROUTE53.

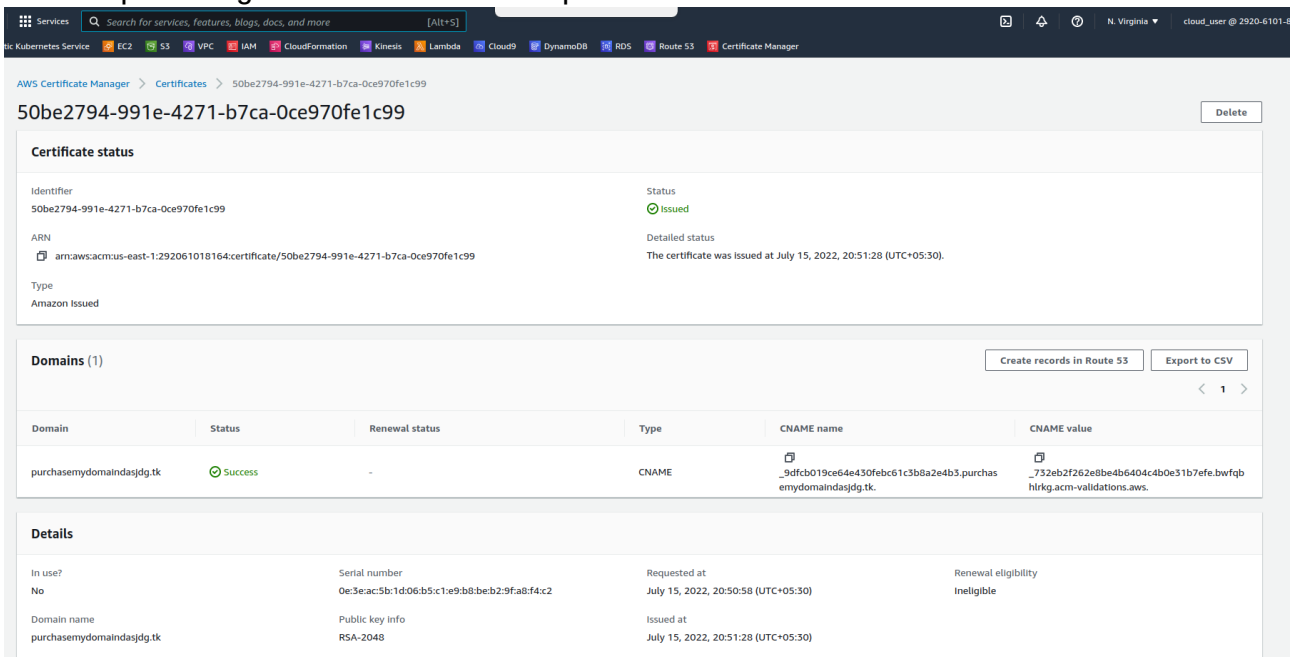
Validate ACM with ROUTE53:
Following: https://docs.aws.amazon.com/acm/latest/userguide/dns-validation.html

ACM -> click on your CERT ID -> create record in Route53 -> clear filter and submit.



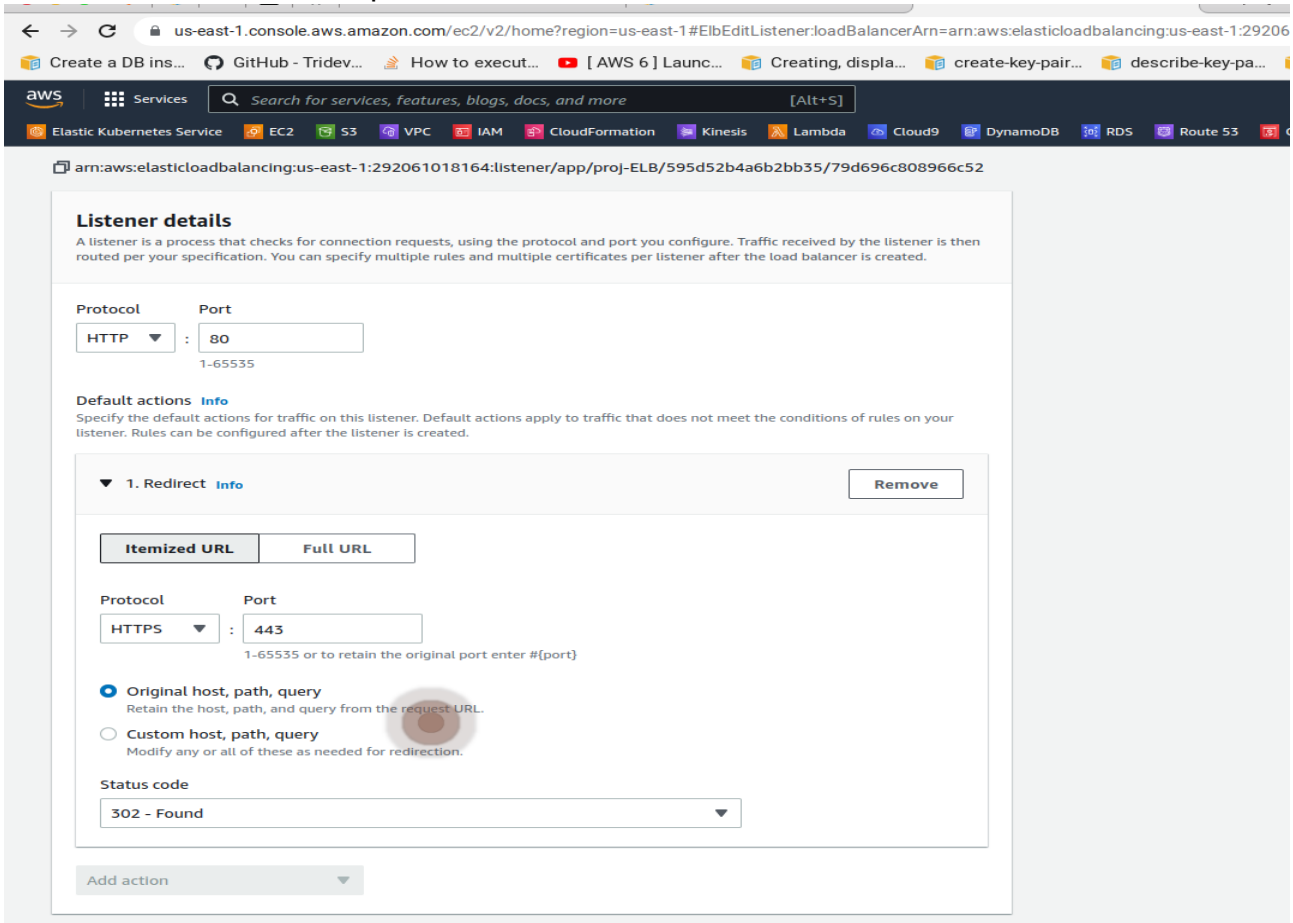
ACM status: Issued.

Note to svae troubles: Do not wait too long to add acm record to avoid pending ACM validation problems.

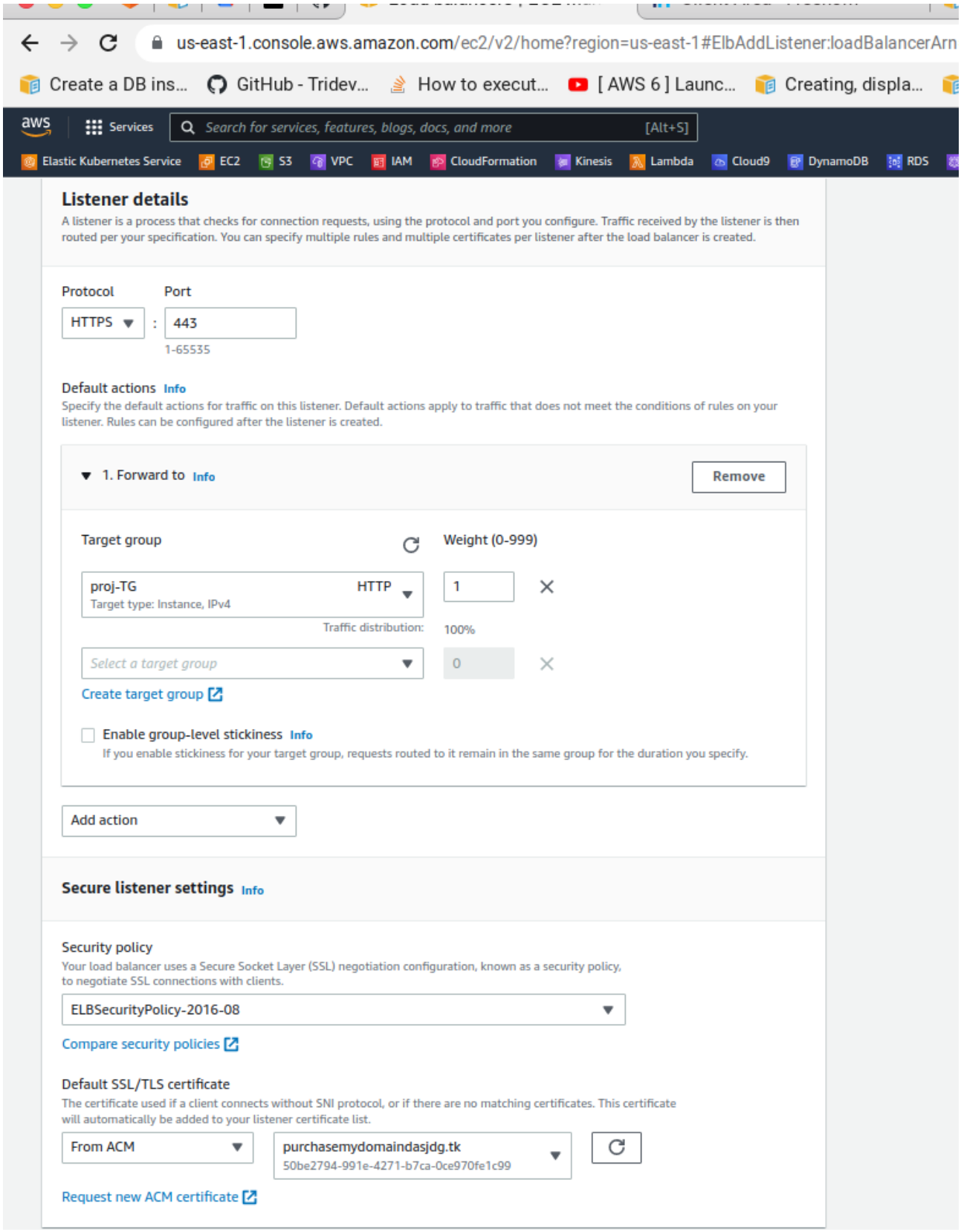


Add/Update ELB Listener rules to make ELB listen to https/route53:

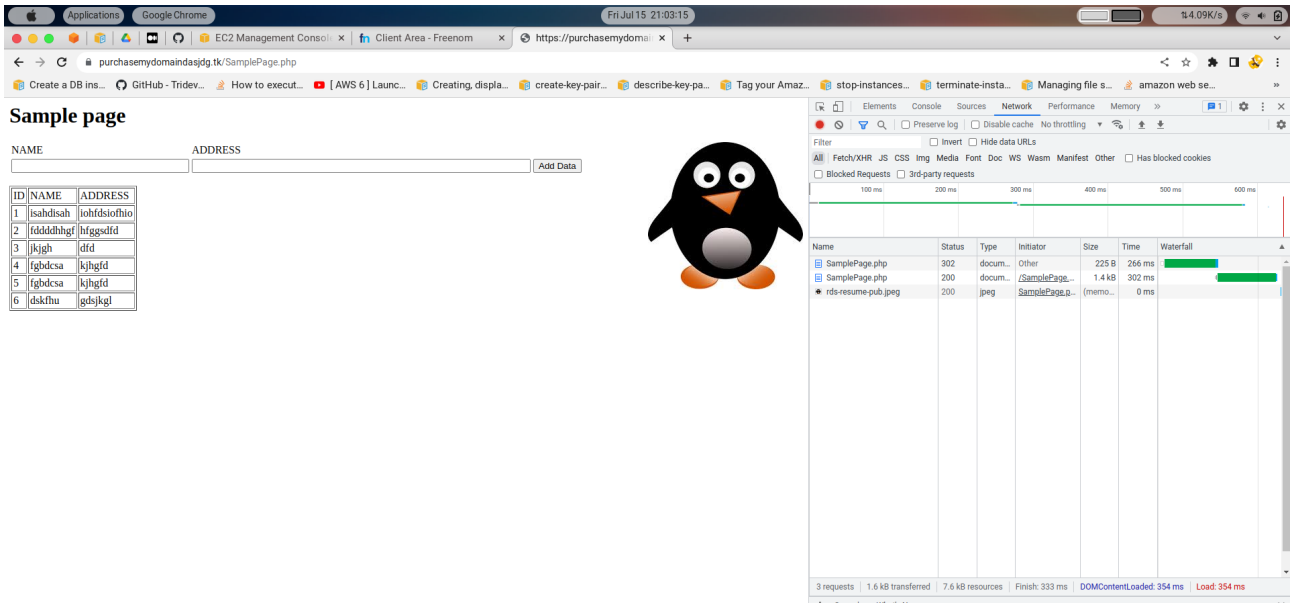
1. Edit rule for http, remove ‘forward to’ and replace with HTTP redirect to 443/https.



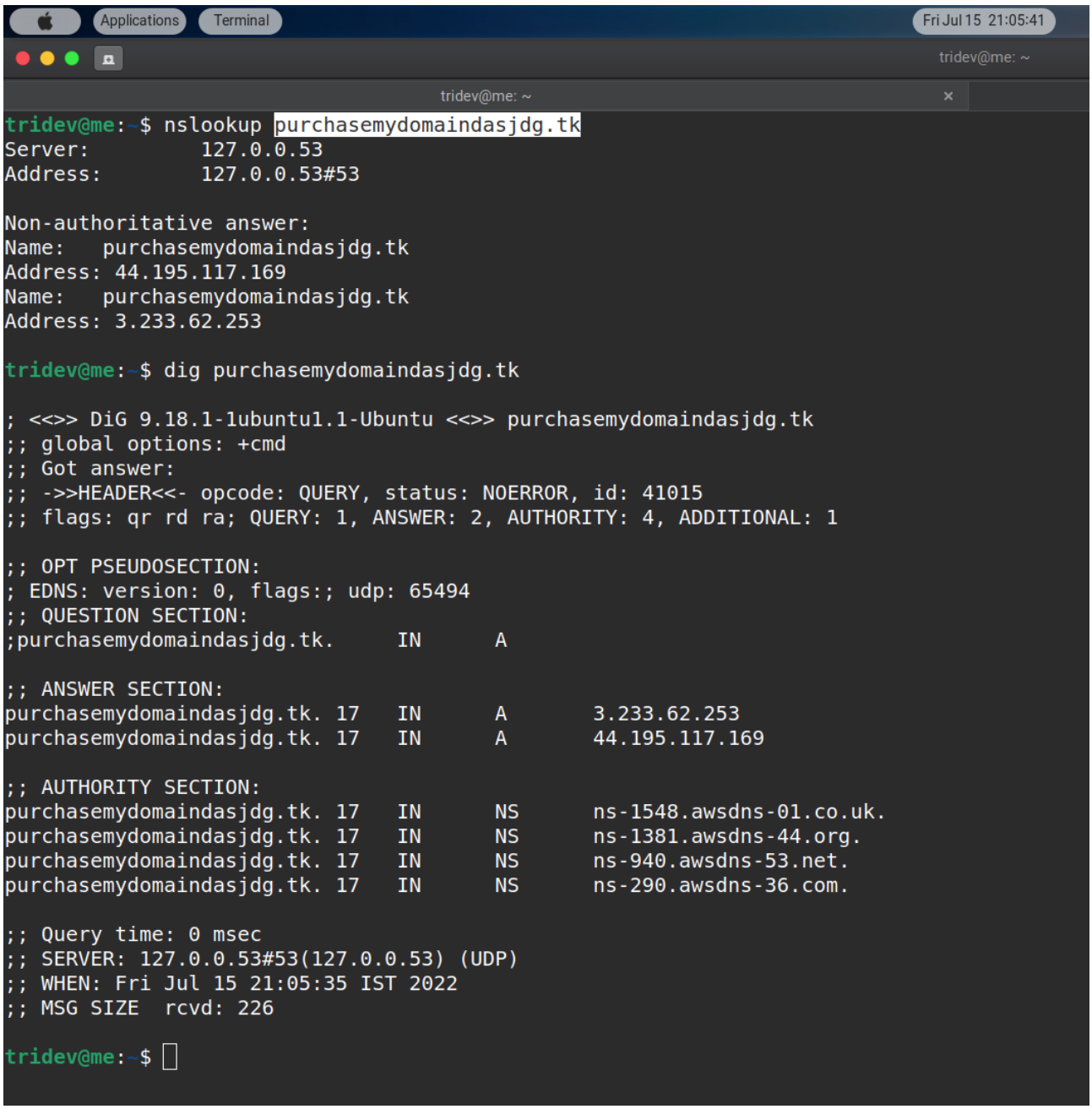
2. Add new listener rule - HTTPS connects with ACM cert:
protocol: https , forward to: target group, ssl/tls cert: our acm cert.



Verify:
Both http to https forceful redirection(security) and https://purchasemydomaindasjdg.tk/SamplePage.php is verified.



(Optional) troubleshootings steps for Nameserver/R53 entry check, incase of issues visting your domain.


Applications
Terminal
Fri Jul 15 21:05:41
tridev@me: ~
tridev@me: ~
tridev@me:~$ nslookup purchasemydomaindasjdg.tk
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
Name: purchasemydomaindasjdg.tk
Address: 44.195.117.169
Name: purchasemydomaindasjdg.tk
Address: 3.233.62.253
tridev@me:~$ dig purchasemydomaindasjdg.tk
; <<>> DiG 9.18.1-1ubuntu1.1-Ubuntu <<>> purchasemydomaindasjdg.tk
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41015
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;purchasemydomaindasjdg.tk.   IN   A
;; ANSWER SECTION:
purchasemydomaindasjdg.tk. 17   IN   A   3.233.62.253
purchasemydomaindasjdg.tk. 17   IN   A   44.195.117.169
;; AUTHORITY SECTION:
purchasemydomaindasjdg.tk. 17   IN   NS   ns-1548.awsdns-01.co.uk.
purchasemydomaindasjdg.tk. 17   IN   NS   ns-1381.awsdns-44.org.
purchasemydomaindasjdg.tk. 17   IN   NS   ns-940.awsdns-53.net.
purchasemydomaindasjdg.tk. 17   IN   NS   ns-290.awsdns-36.com.
;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Fri Jul 15 21:05:35 IST 2022
;; MSG SIZE rcvd: 226
tridev@me:~$