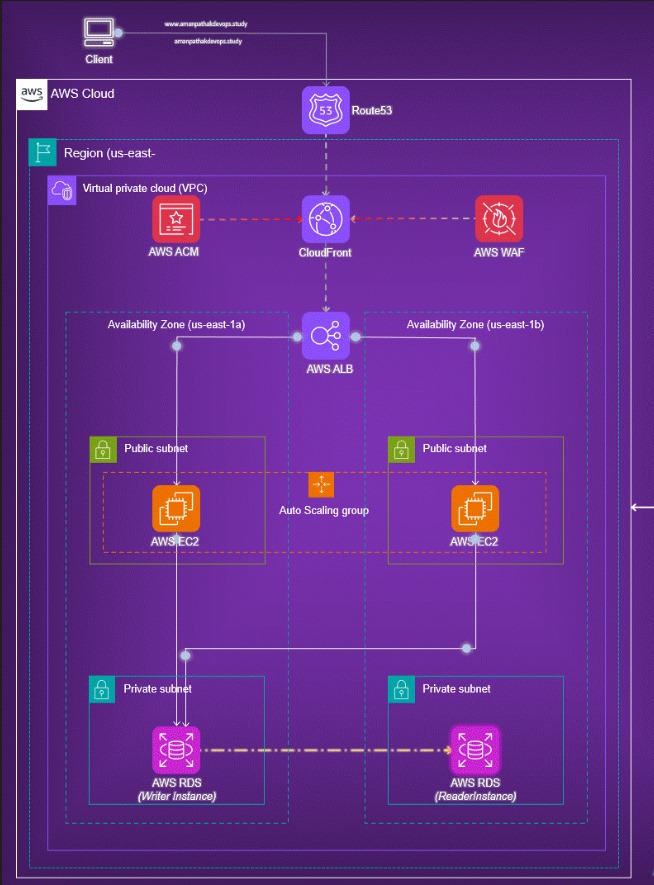
# Prathi Guna Ram Charan

# 125-BATCH

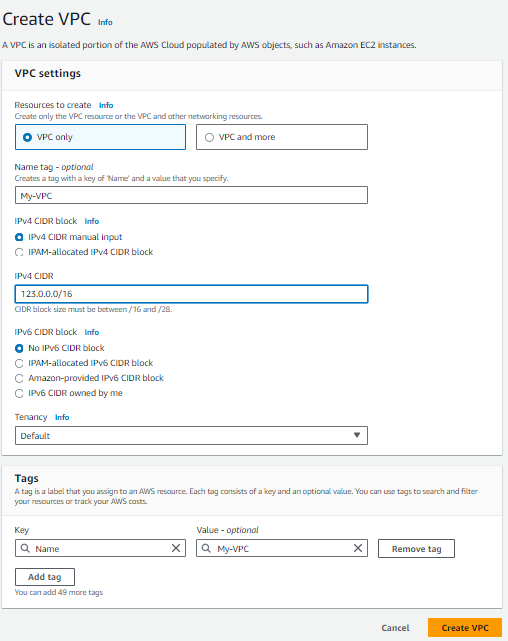
# AWS WED APPLICATION ARCHITECTURE



* The diagram illustrates an AWS wed application architecture. Here’s a detailed explanation of its components and flow:
* Client: Represents the end user accessing the application.
* Route53: AWS's DNS service that routes traffic to the correct resources
* CloudFront: AWS's Content Delivery Network (CDN) to deliver content with low latency.
* AWS WAF (Web Application Firewall): Protects the application from common web exploits.
* AWS ACM (Certificate Manager): Manages SSL/TLS certificates for the secure transfer of data.
* VPC (Virtual Private Cloud): Isolates the network within the AWS cloud.
* Availability Zones: Two availability zones (us-east-1a and us-east-1b) for high availability and fault tolerance.
* In each availability zone:
* Public subnet:
* EC2 instance: Host the application or services.
* Auto Scaling Group: Ensures the right number of EC2 instances to handle the load
* Private subnet:
* AWS RDS (Relational Database Service): Manages the database. There's a writer instance in one subnet and a reader instance in another for load balancing and high availability.
* (We should create another subnet in RDS because Multi A-Z cluster need 3 subnets in three availability zone)
* Network flow:
* Route53 routes user requests to CloudFront.
* CloudFront interacts with AWS WAF for security checks.
* Requests pass through the Application Load Balancer (ALB), which distributes them to the EC2 instances in the public subnets.
* The EC2 instances connect to the AWS RDS instances in private.
* subnets for database operations.

# create a private VCP

* My-VPC – 123.0.0.0/16 in Virginia region.



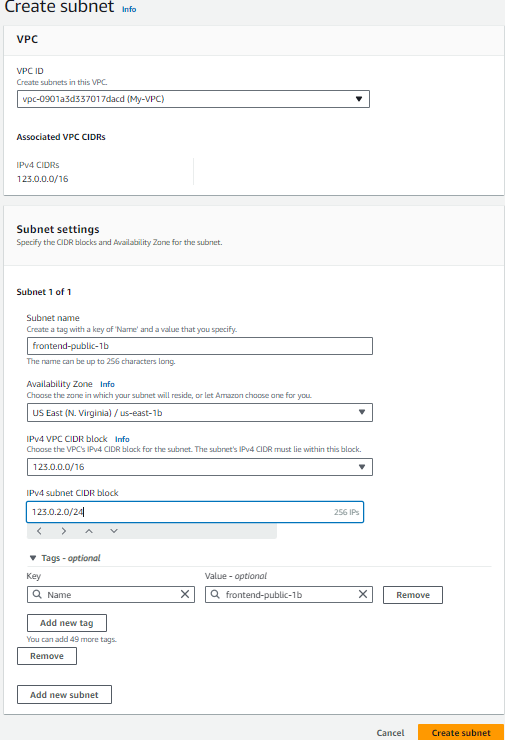
* Click on create VPC

# Create 4subnets-(2 public,2 private)

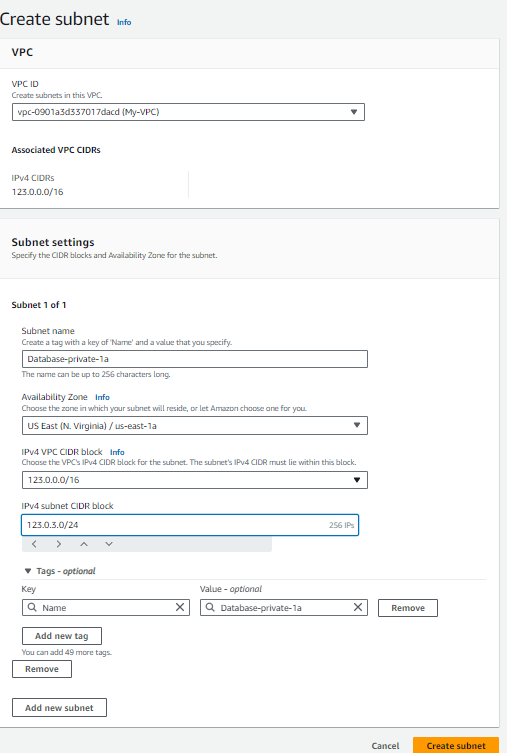
* Select the VPC for all the subnets (My-VPC).
* Create a first public subnet in 1a-availability zone (frontend-public-1a)- 123.0.1.0/24.
* Create a second public subnet in 1b-availability zone (frontend-public-1b)- 123.0.2.0/24.
* Create a first private subnet in 1a-availability zone (Database-private-1a)- 123.0.3.0/24.
* Create a second private subnet in 1b-availability zone (Database-private-1b)- 123.0.4.0/24.
* First public-1a-(123.0.1.0/24)-frontend-public-1a.



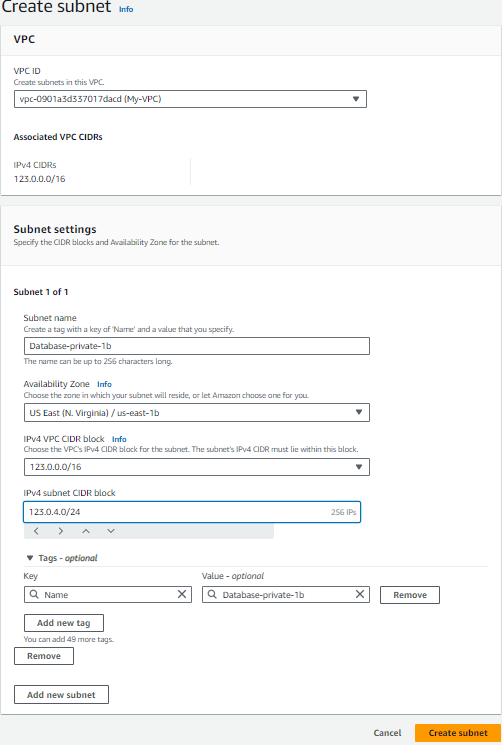
* Second public-1b-(123.0.2.0/24)-frontend-public-1b.



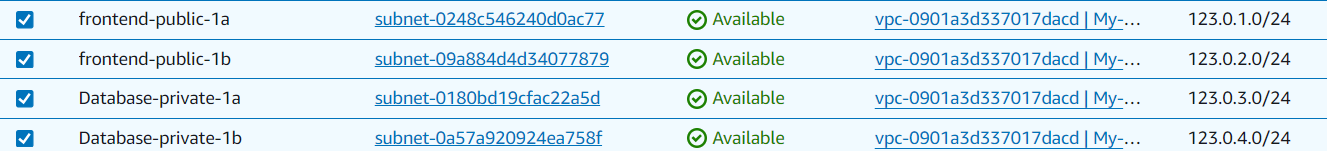
* First private-1b-(123.0.3.0/24)-Database-private-1a.



* Second private-1b-(123.0.4.0/24)-Database-private-1b.

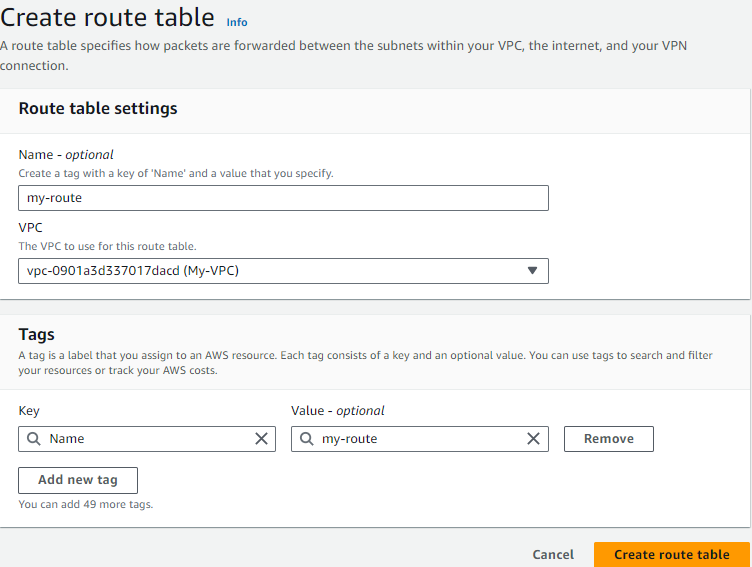


* Created 4 subnets.

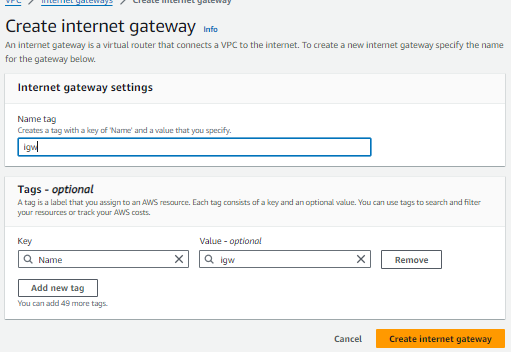


## Create a route-table with Internet- gateway & subnet association

* Create a route table-(my-route) & with My-VPC.



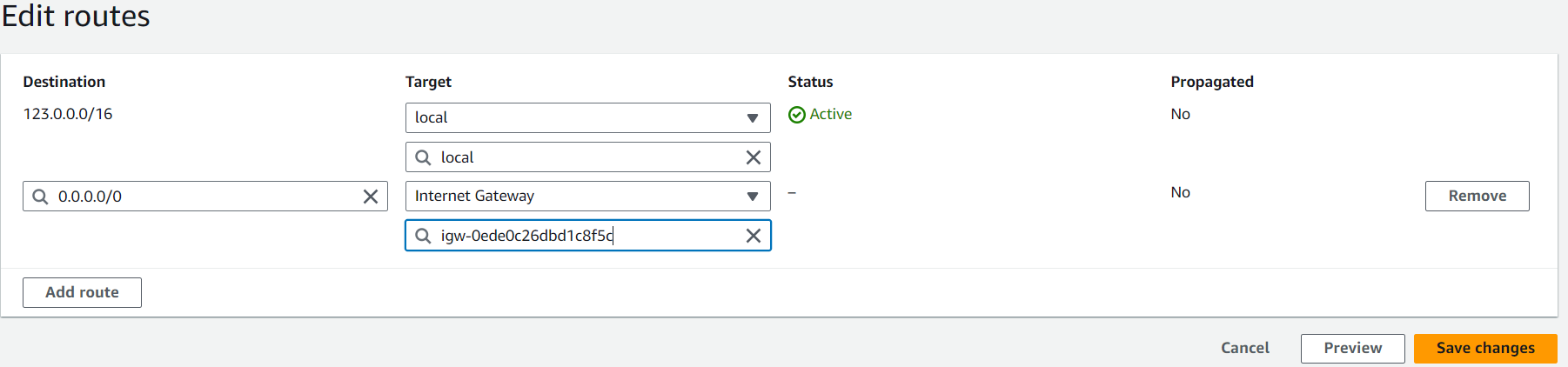
* Next create Internet gateway-(igw).



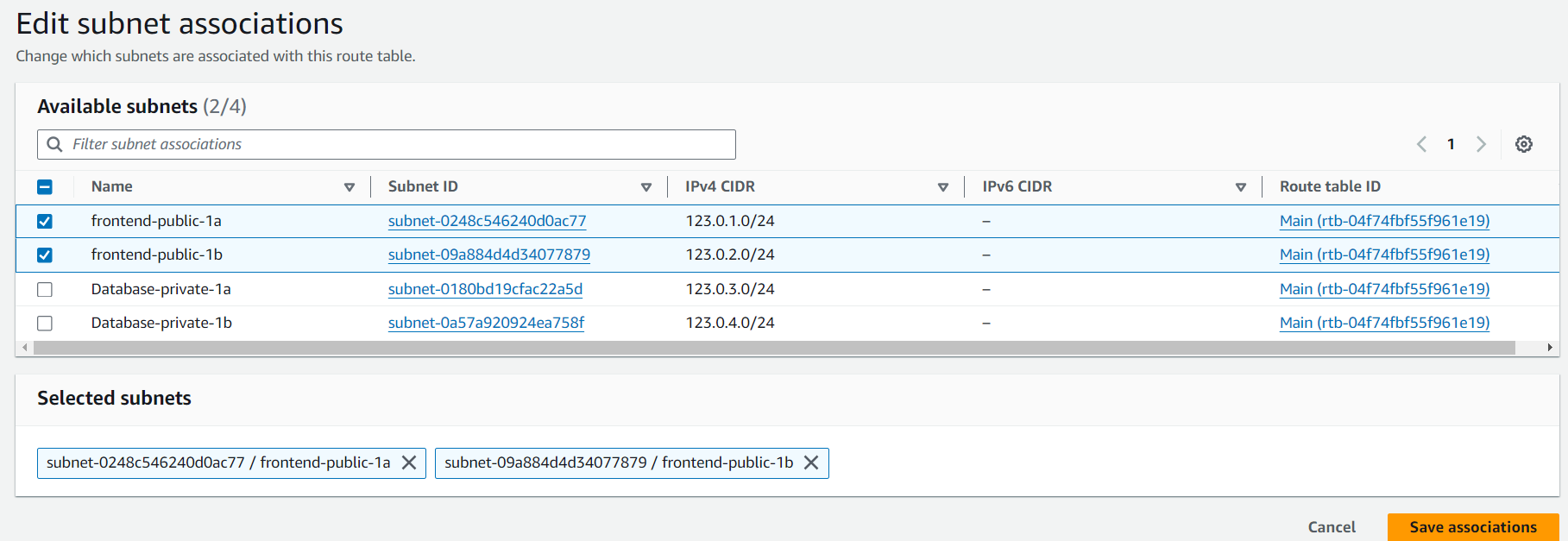
* Attach the VPC



* Now edit the route-table.
* Click on actions edit routes and add the internet-gateway.



* Next edit the subnet associations & and the 2 public subnets.

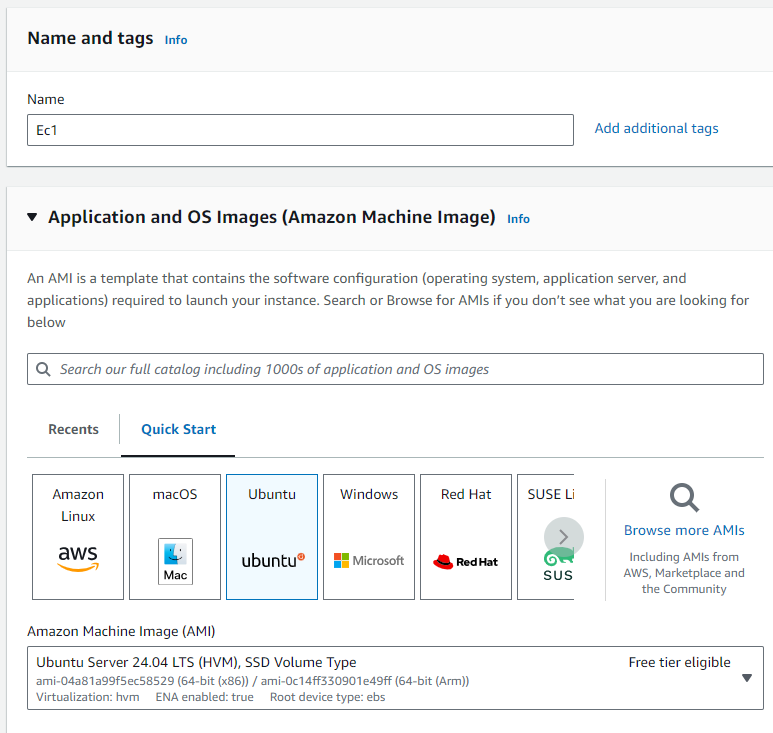


* After add the frontend-public-1a & frontend-public-1b go and save the changes.
* Next create 2 EC2 instance in 2 different availability zones-(1a&1b).

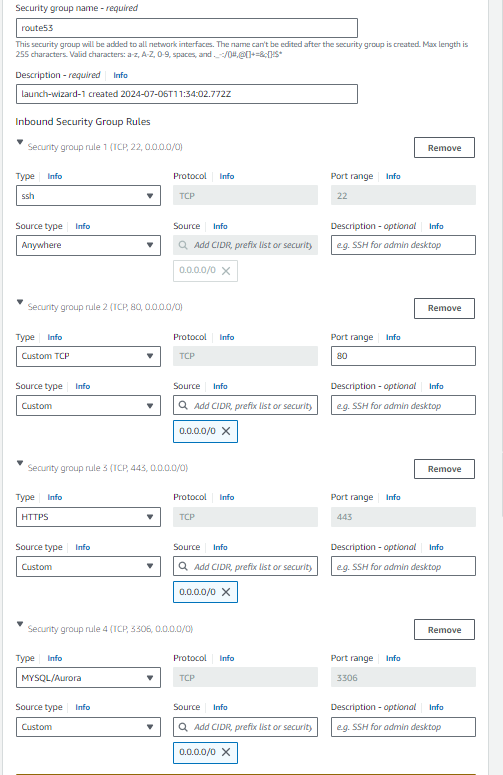
# Create two EC2 instance

* Ec1-public =>create a key pair, select the VPC, 1a-availability zone-(frontend-public-1a), edit security add custom-TCP-80(for nginx), HTTPS-80-ANYWHERE (0.0.0.0/0), MYSQL/AURO-3306-ANYWHERE (0.0.0.0/0).
* Ec2-public => same key pair, select the VPC, 1b-availability zone-(frontend-public-1b), add the same security group.
* Take the AMI Image as (UBUNTU)

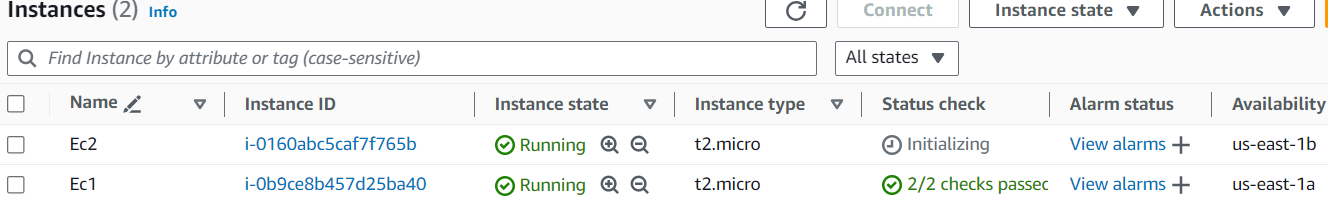
# Ec1







* Click on launch instance
* After Ec1 is created next create the Ec2 with same AMI image, key pair, frontend-public-1b(subnet), same security group (route53) & click on launch instance



* Next connect to Ec1 in the get bash by copying the SSH client and install nginx in the root user
* To install nginx – (apt update -y)

(apt install nginx -y)

(cd /var/www/html)-this is for going into html path to change the index.html file by removing index.html - (rm index.html) & next create the index.html

and past the html code from the browser by ( vi index.html) – insert the code and press- (esc : wq) to exit the vi file next restart the nginx => (systemctl restart nginx)

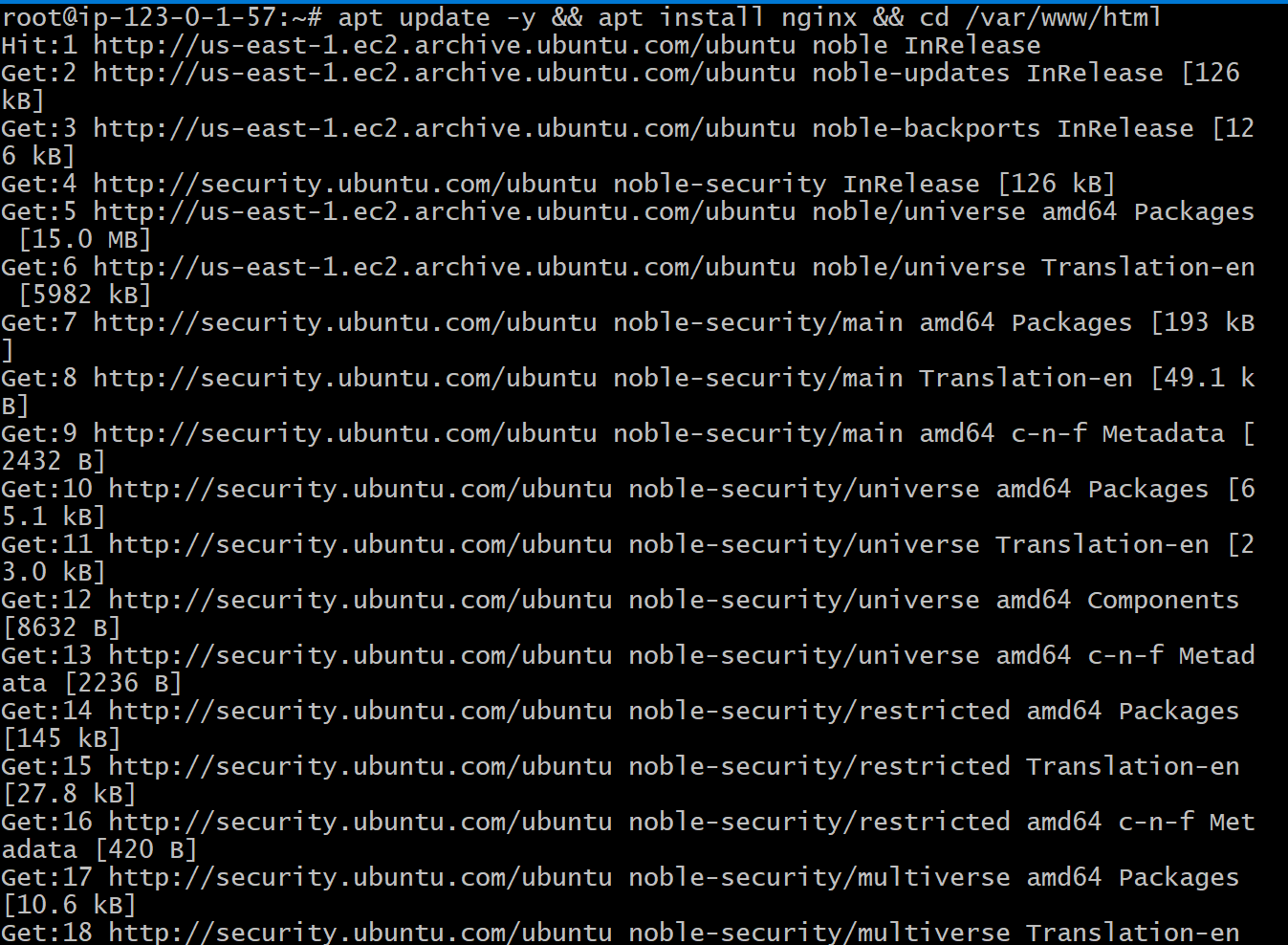
* To install MYSQL – (apt update -y)

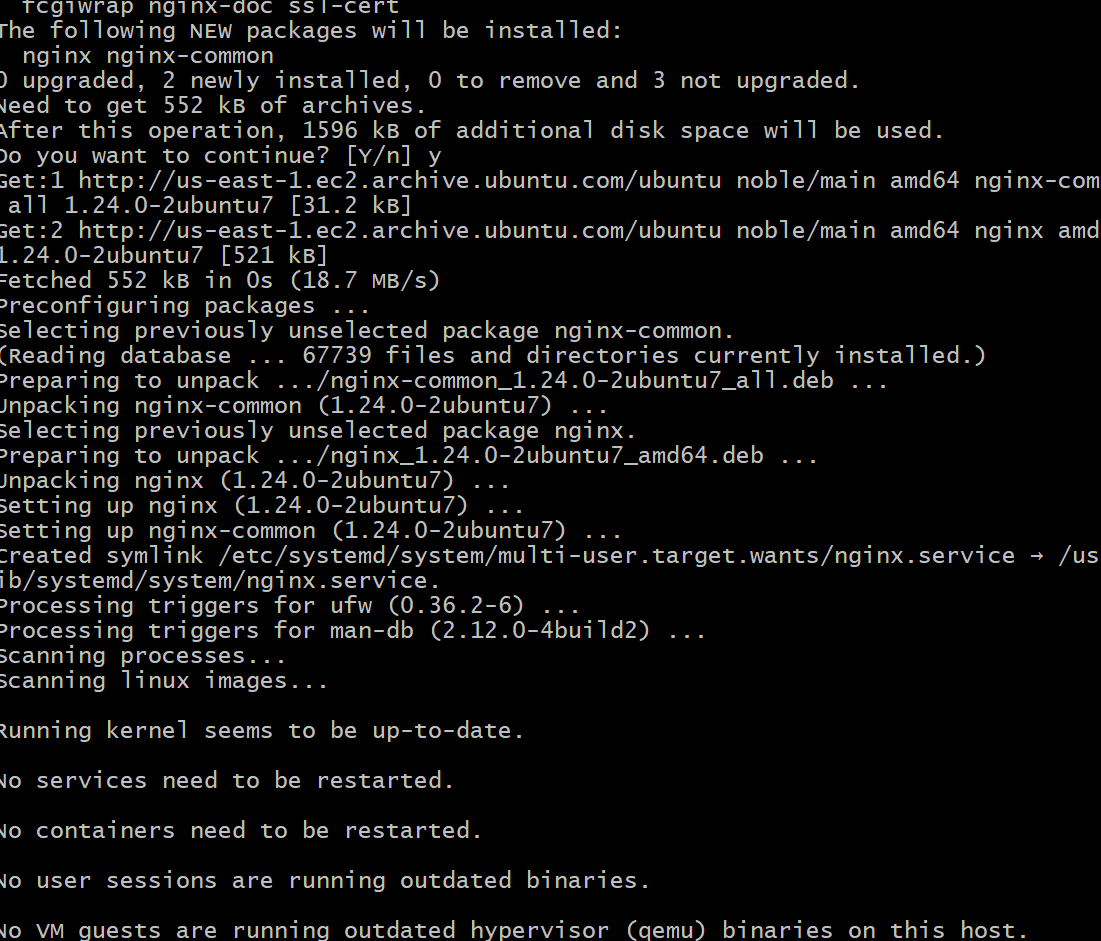
(apt install mysql-server)

(systemctl start mysql.service) – To start the MYSQL server

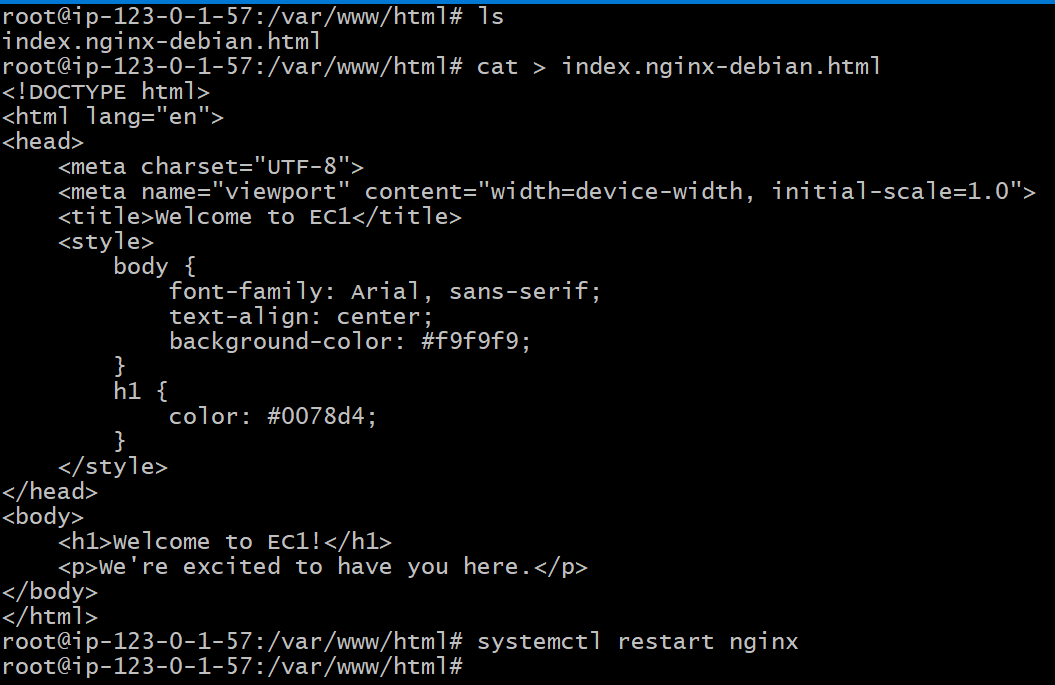
* Do the same in the Ec2 instance (install nginx, MYSQL)

* Installing Nginx, updating , the path to nginx html file

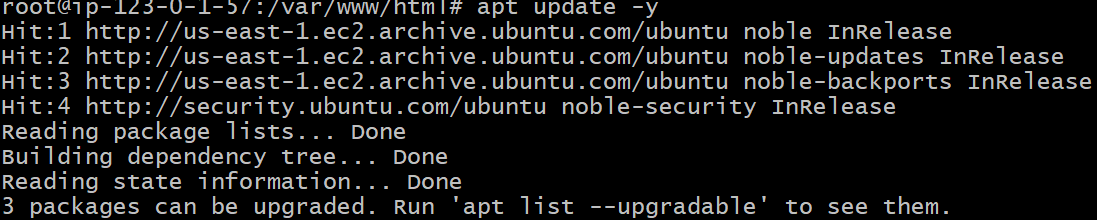


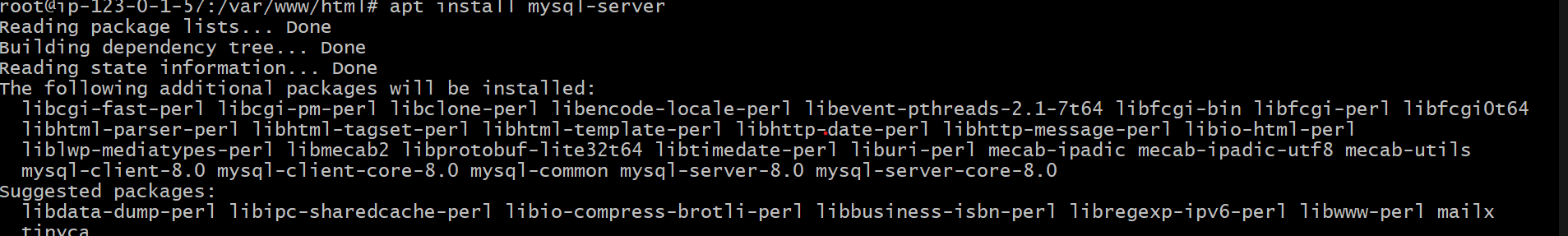


* Editing the index-debian.html file by adding html code into it.



* Next check the html code is working .
* Next install MYSQL in the same server.

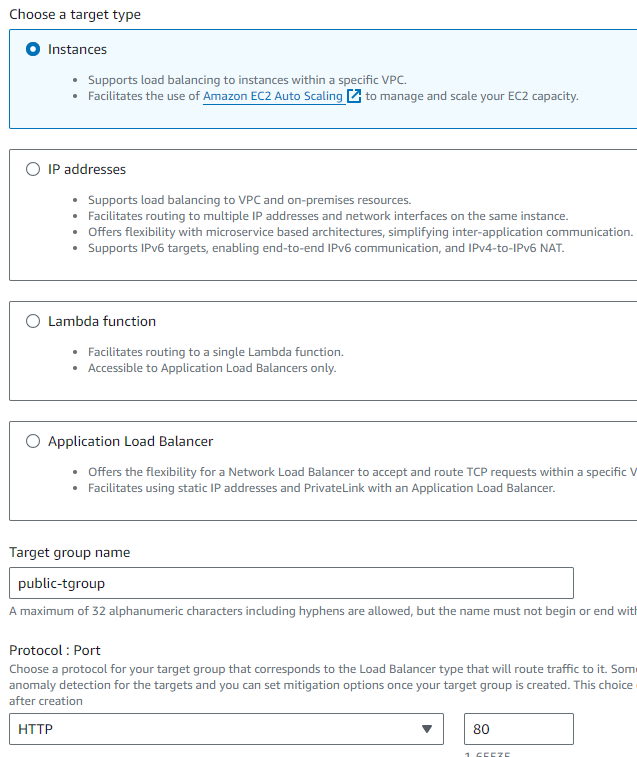




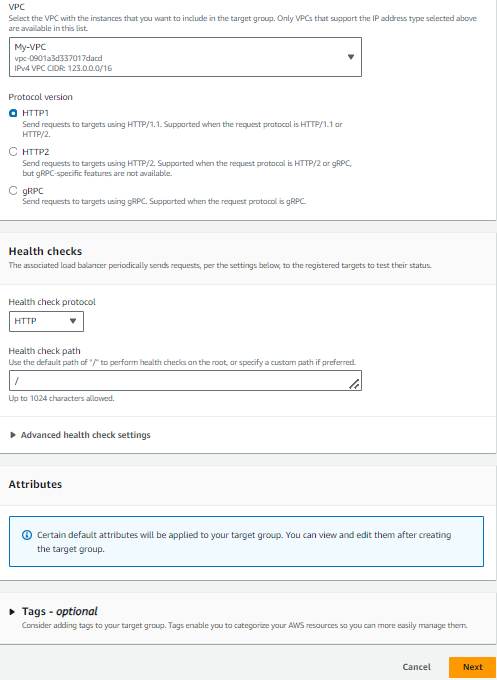
* Next start MYSQL server-(systemctl start mysql.service)
* And do the same process in the Ec2 instance .

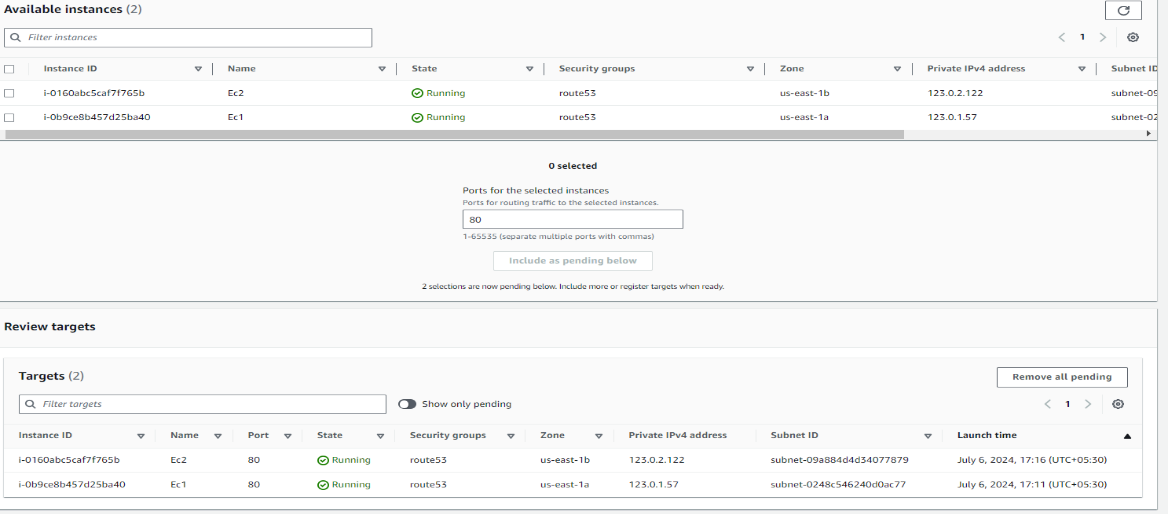
# Create Target-group, Load balancer, Auto scaling

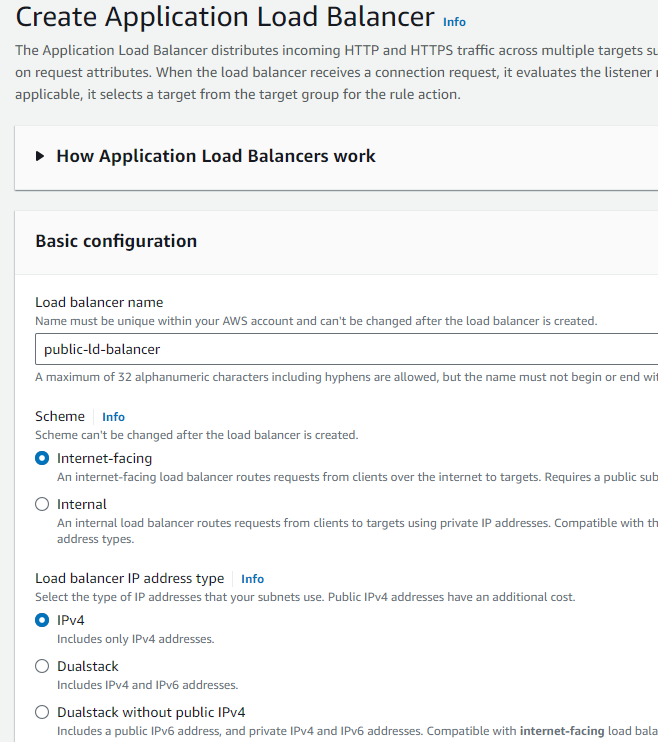
* Target group-type(instance)-name-(public-tgroup)-protocol-HTTP-80-(port number).



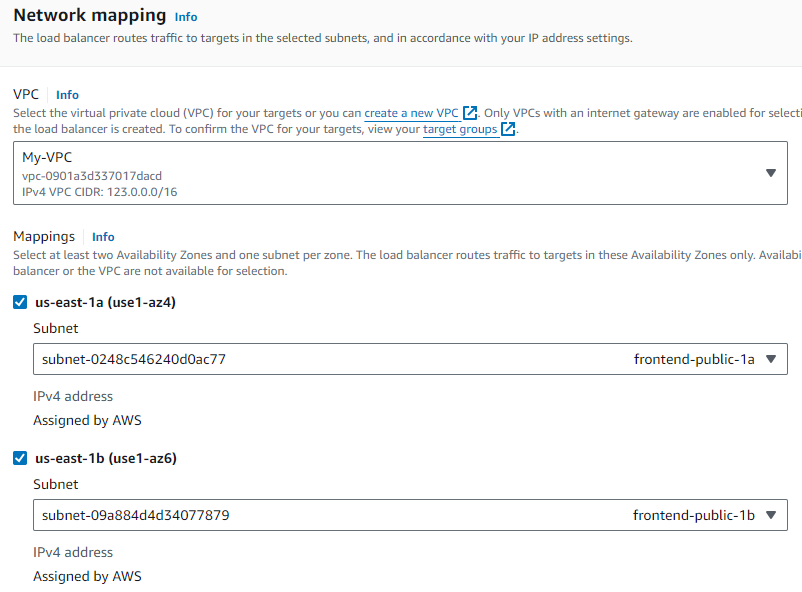
* Next select the VPC & click on next.



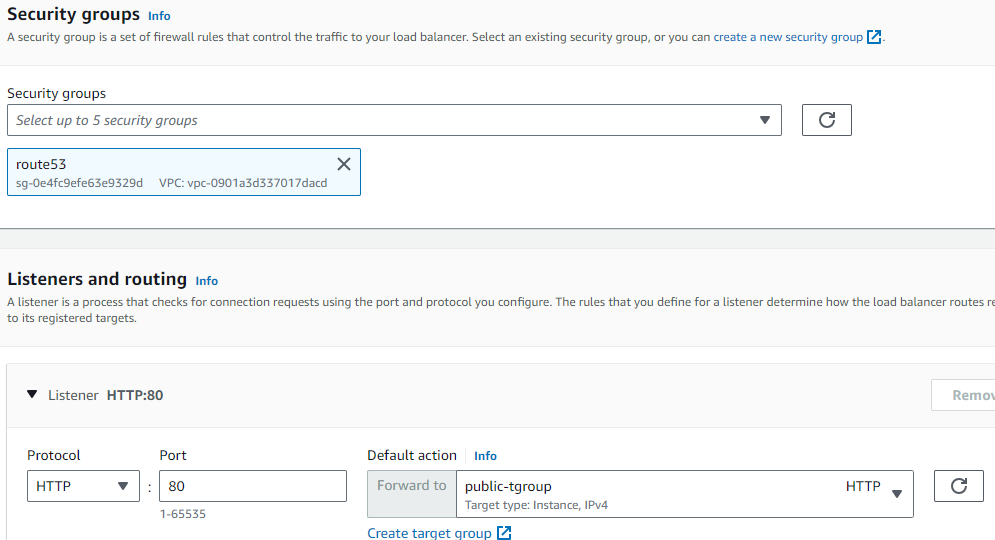
* 
* Next click on create target group.
* Next create load balancer-(application load balancer).



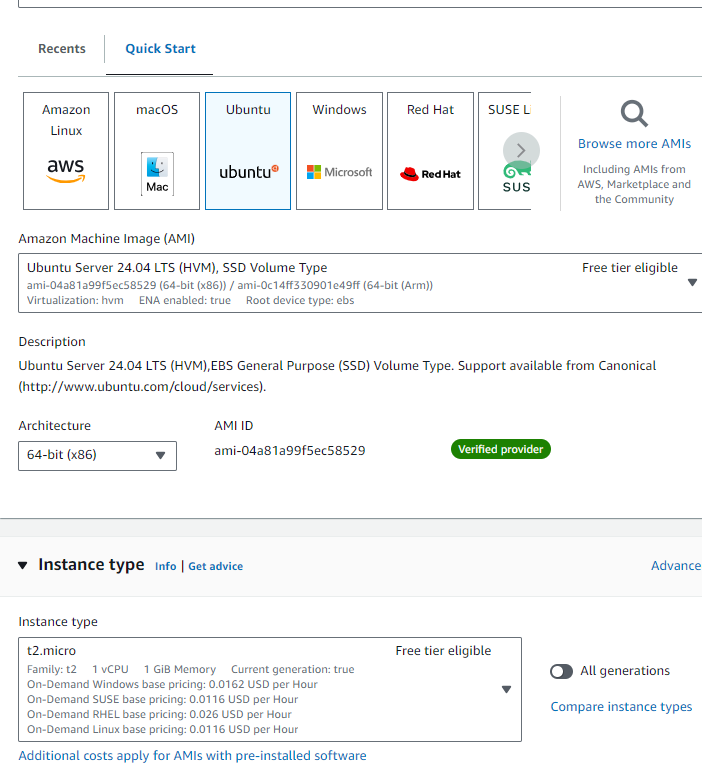
* select the VPC and the subnets in the mapping (frontend-public-1a, frontend-public-1b).

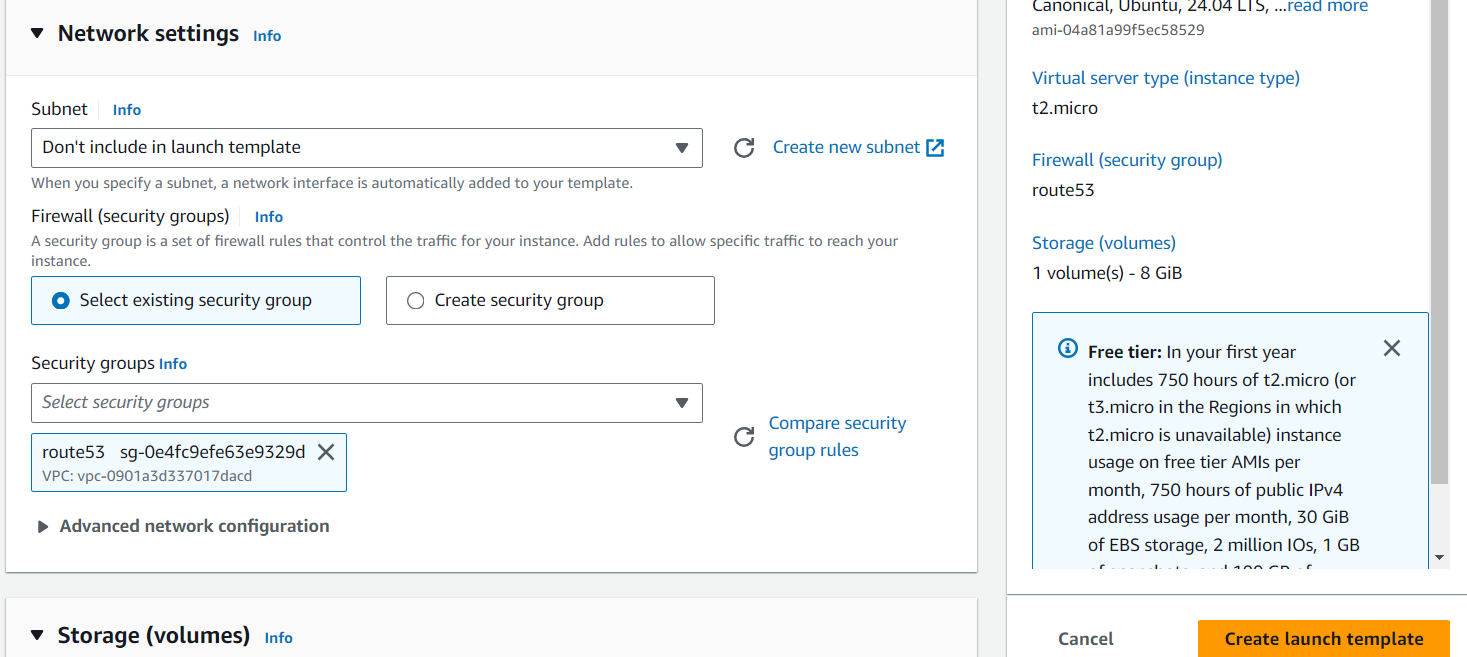


* next add the same security group & target group.

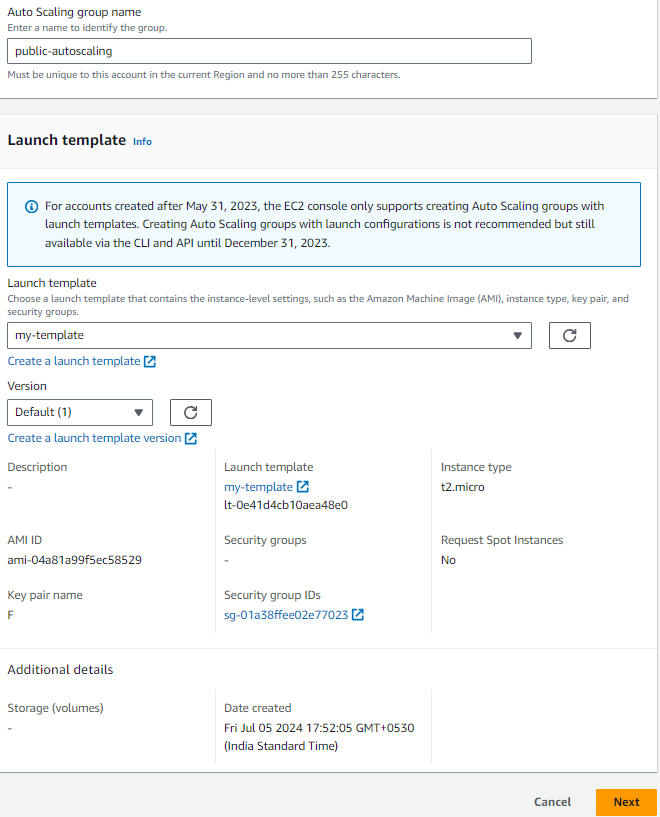


* Next click on create load balancer.
* Now create the Auto-scaling group.
* First create the lunch template- (AMI, key pair, instance type, security group).

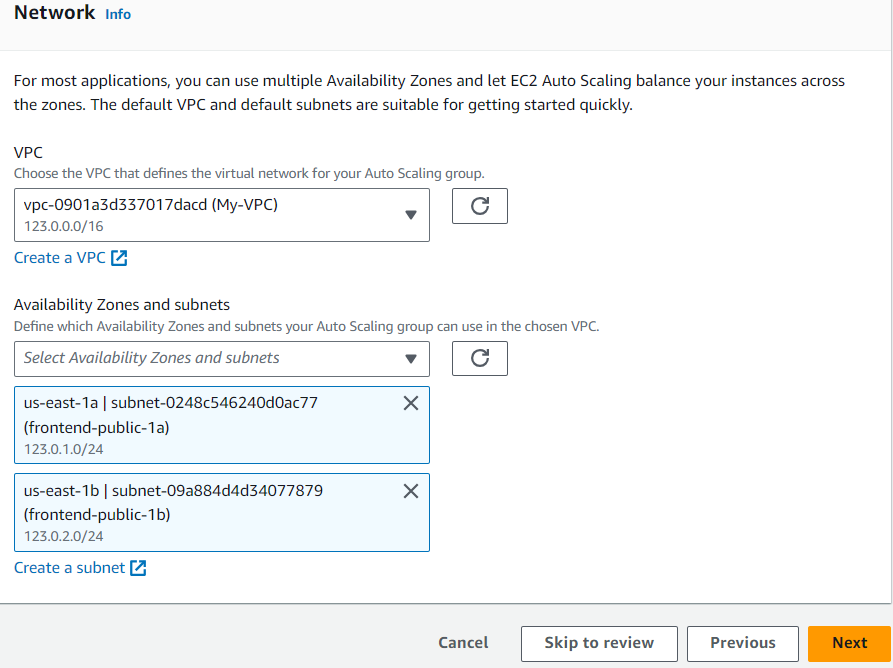




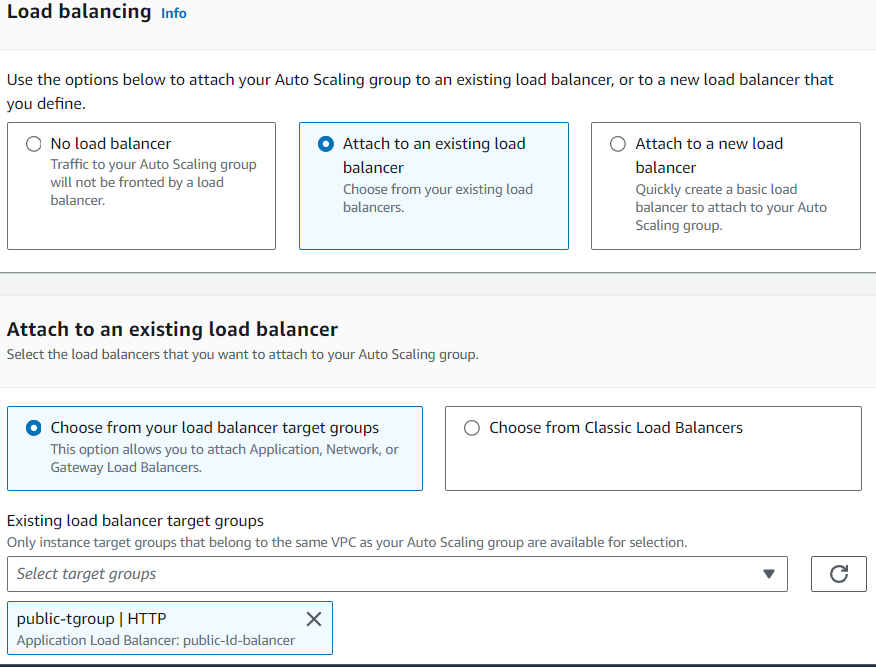
* Click on create lunch template.
* Now go to create the auto-scaling group.



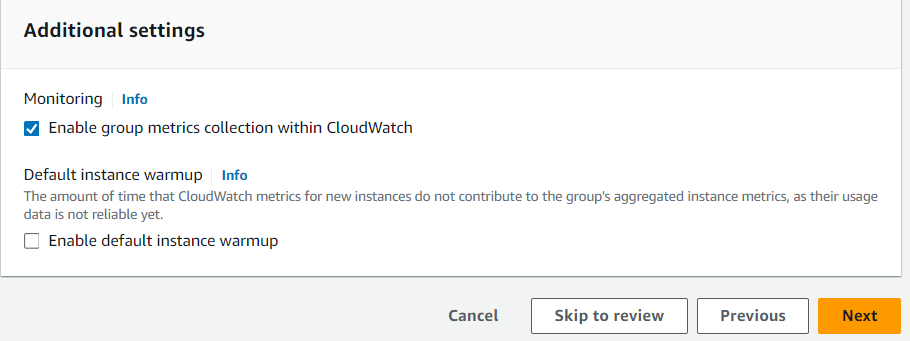
* Click on next & add the VPC and the subnets in the frontend.



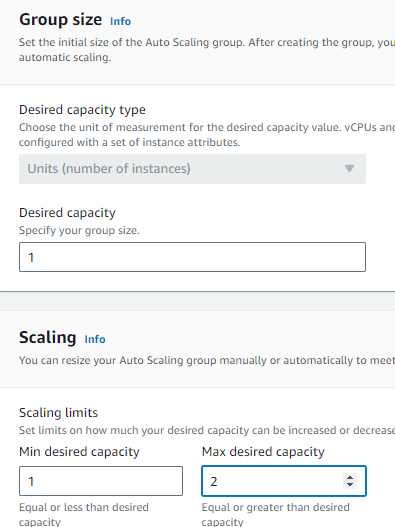
* Attach the load balancer & target group.



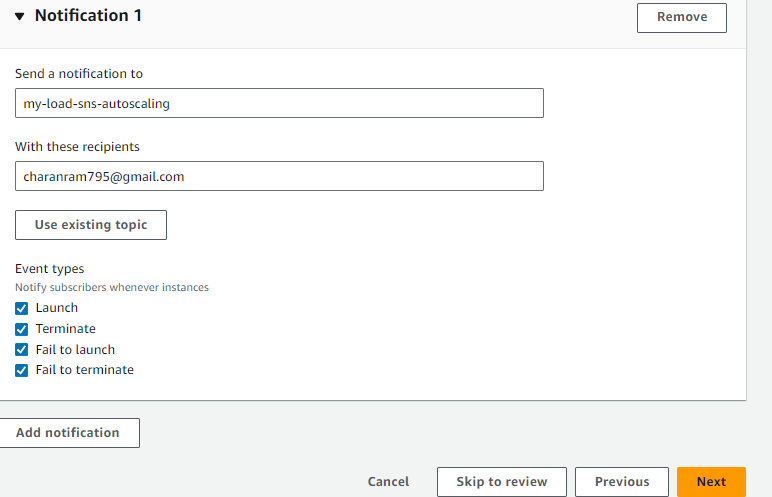
* Enable the cloud watch & click on next.



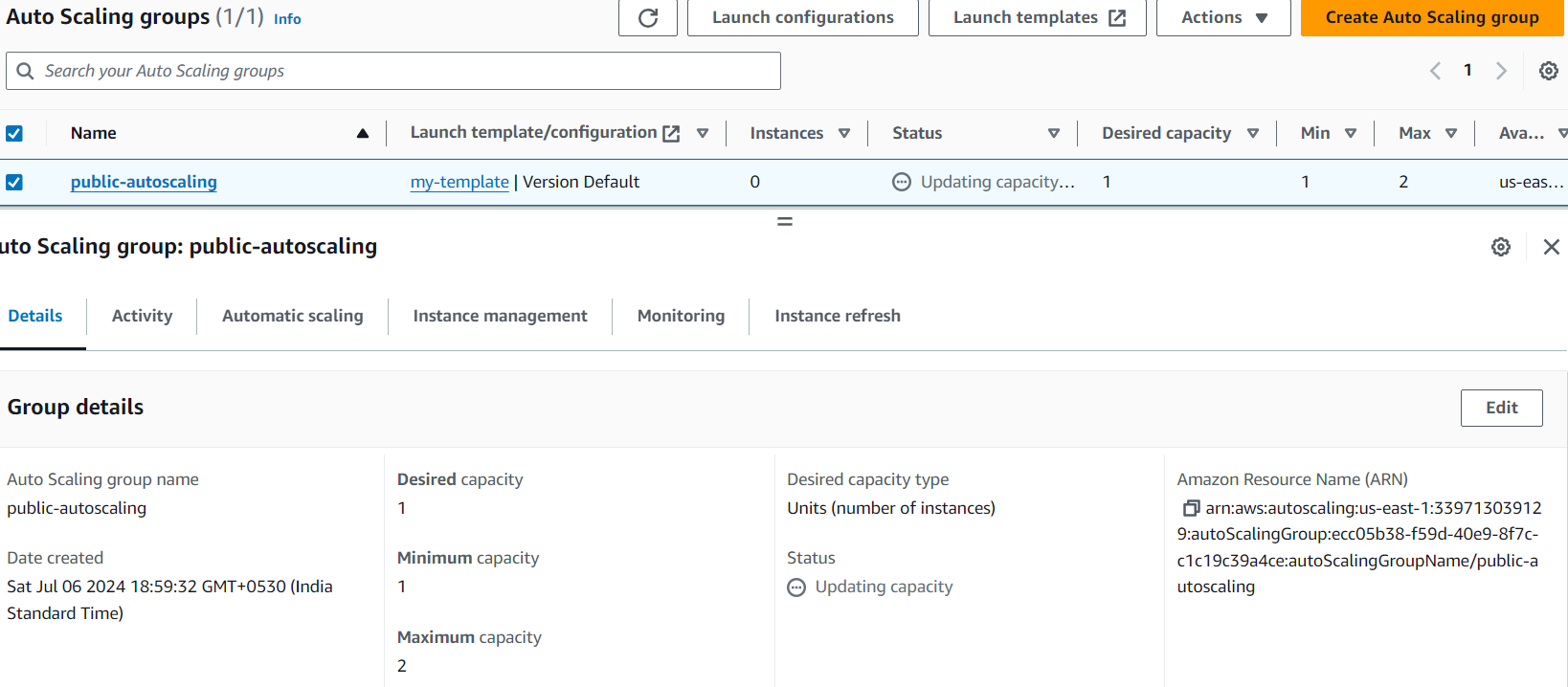
* Give the capacity off the instance to be created.



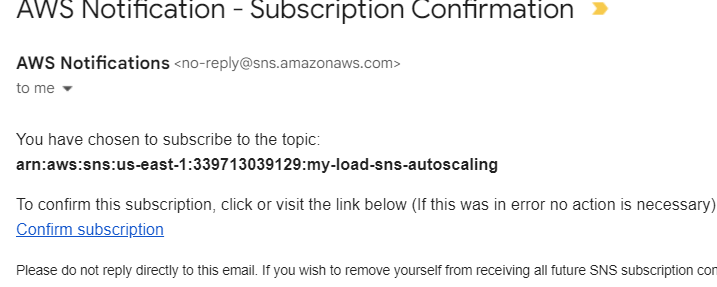
* Scroll down and click on next.
* Add the notifications to the email & type the SNS topic name.



* Click on next & skip the tags and click on next.
* Next click and scroll down and click on create auto-scaling group.

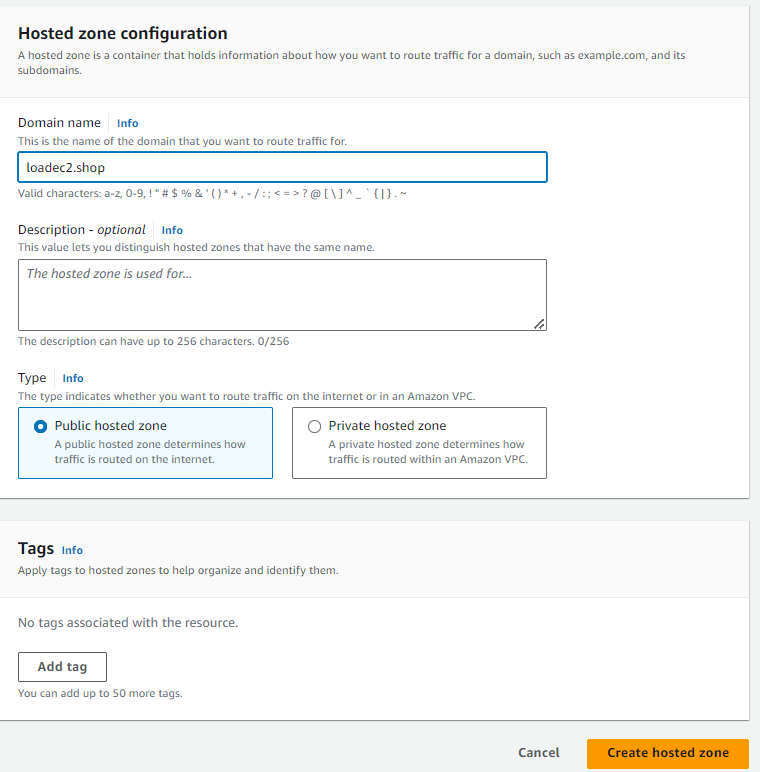


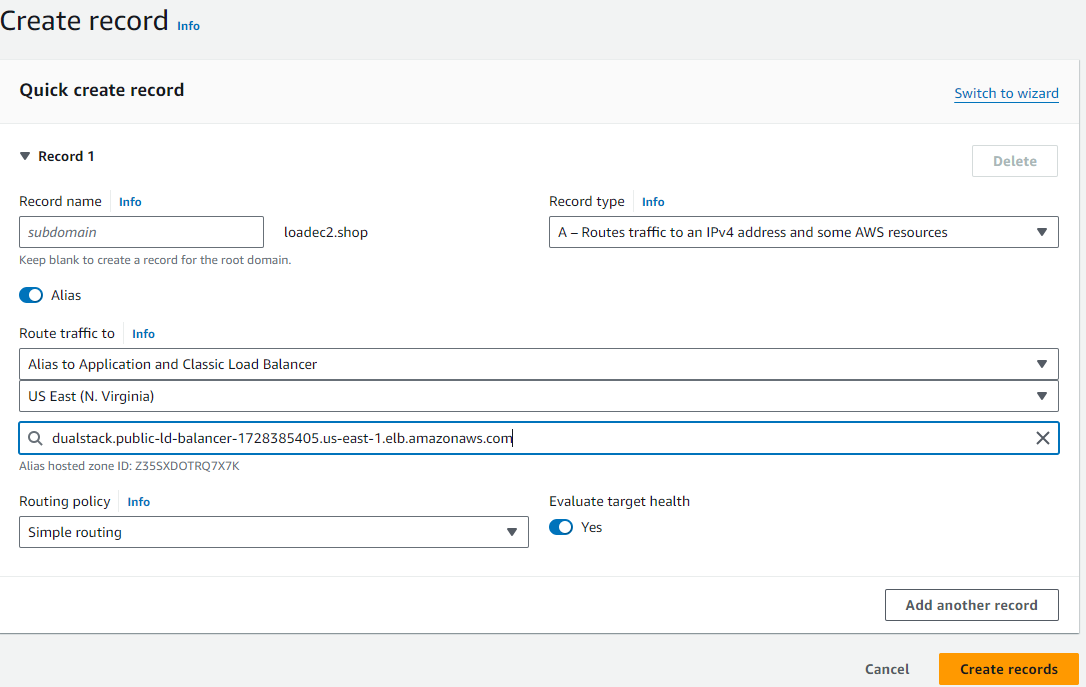
* Go check your email either have you received an SNS notification mail.

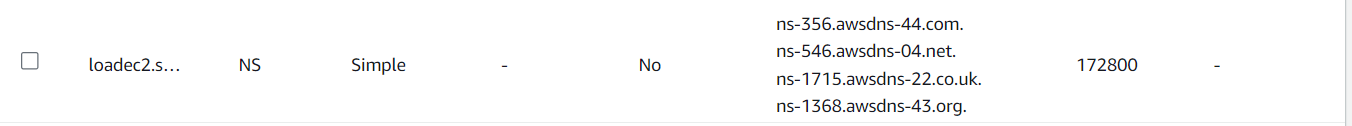


# Create Route53, ACM, CloudFront

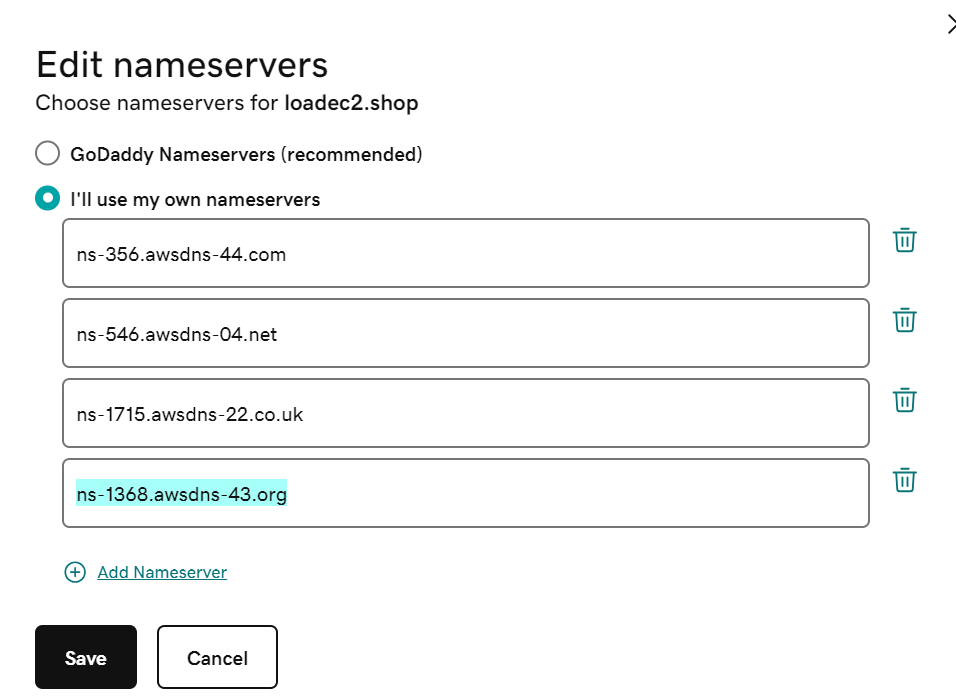
* First go to create route 53.
* Click on create Hosted-zone.
* First buy a domain name in the (godaddy.com) for low price- my domain name (loadec2.shop).
* Copy the domain name to the Hosted zone & create an hosted-zone.



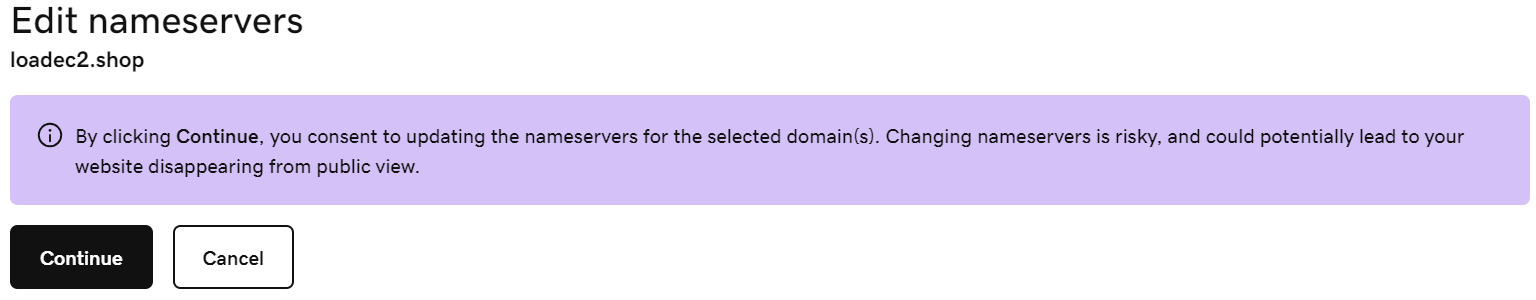
* Click on create a Hosted-zone
* Next click on create a record in the hosted-zone & enable Ailas & add the application load balancer & region & load balancer & click on create records
* Next edit the nameservers in the Domain – (click on manage the DNS in the (godaddy.com)-click on change nameservers on your Domain).
* Copy from the Route-53.



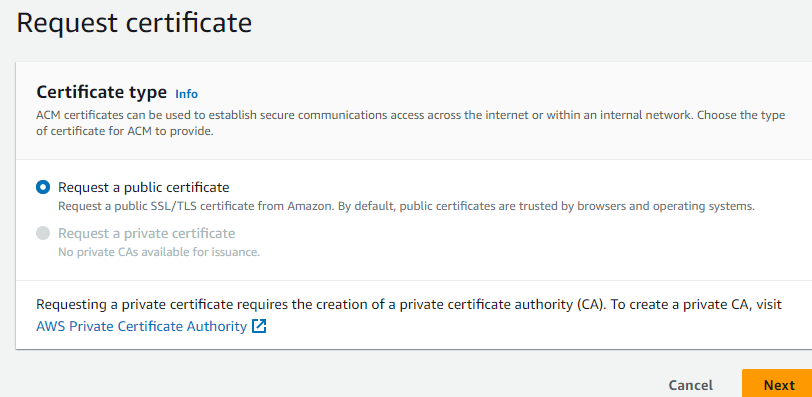
* Past the servers in the Domain nameservers.



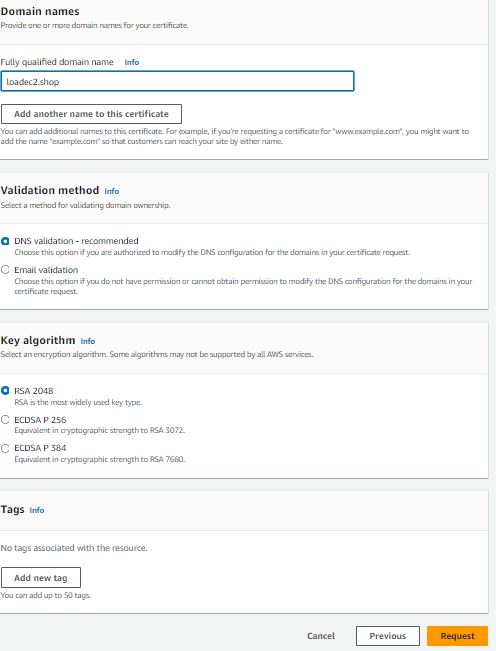
* Click on save & press continue



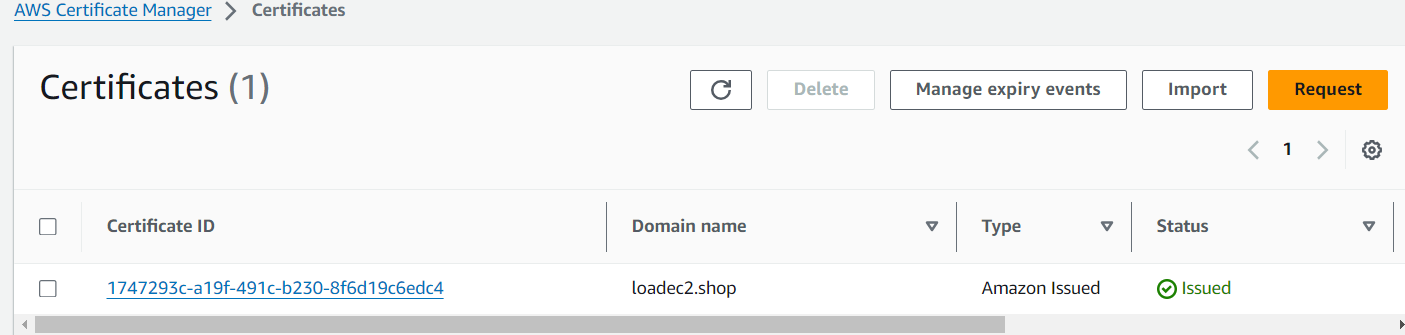
* After the nameservers has been changed in Domain DNS, create ACM
* Next go to ACM certificate manager & click on create certificate.



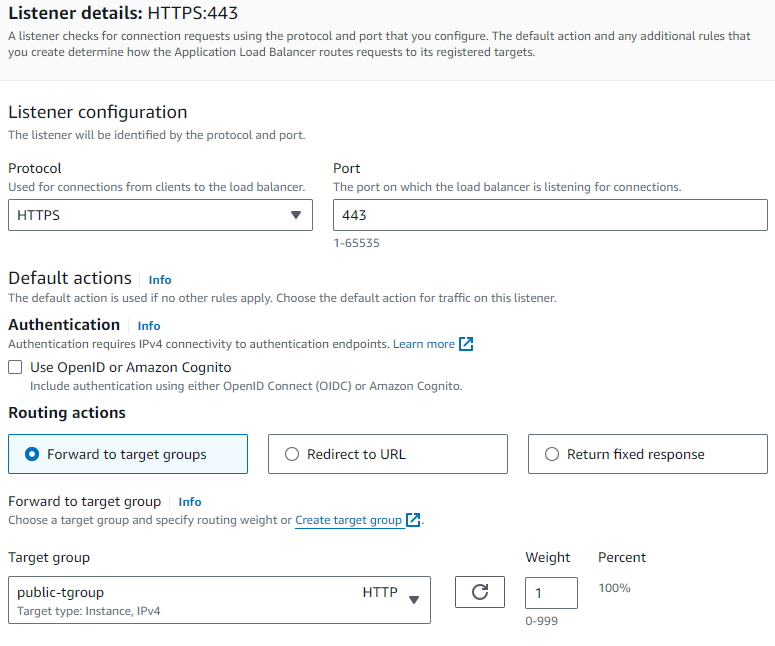
* Next on next and copy the Domain name & scroll down click on request a certificate.



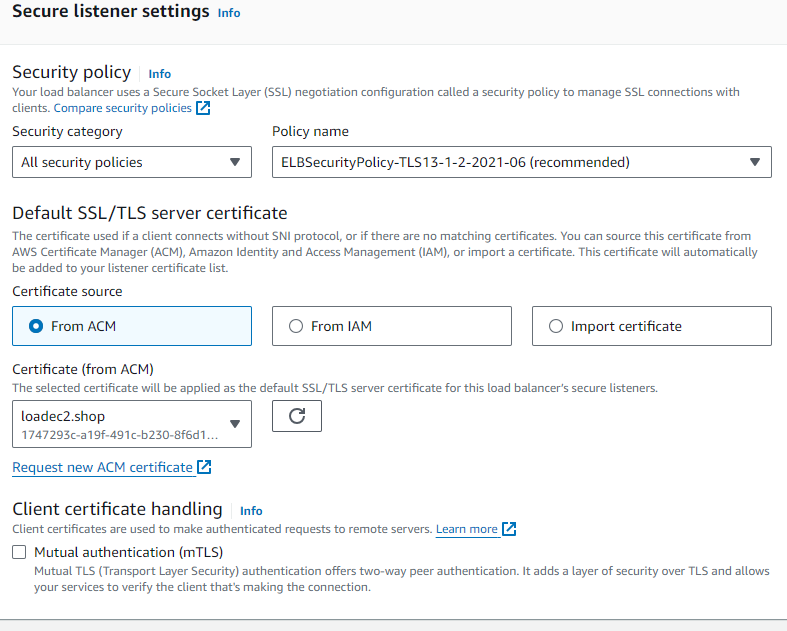
* Wait until the certificate is issued



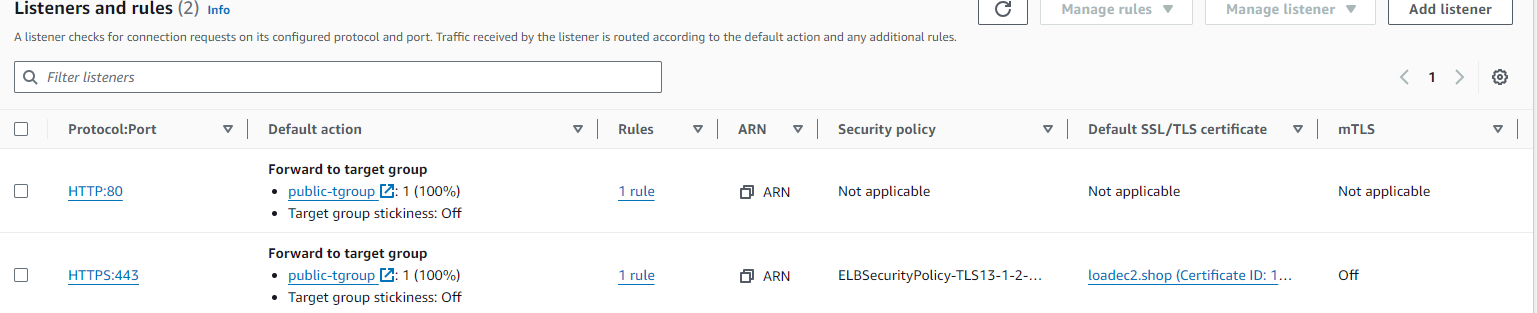
* Check the CNAME is created in the Route53 Is created
* Next go to the load balancer & click on Add listeners the https & add target group



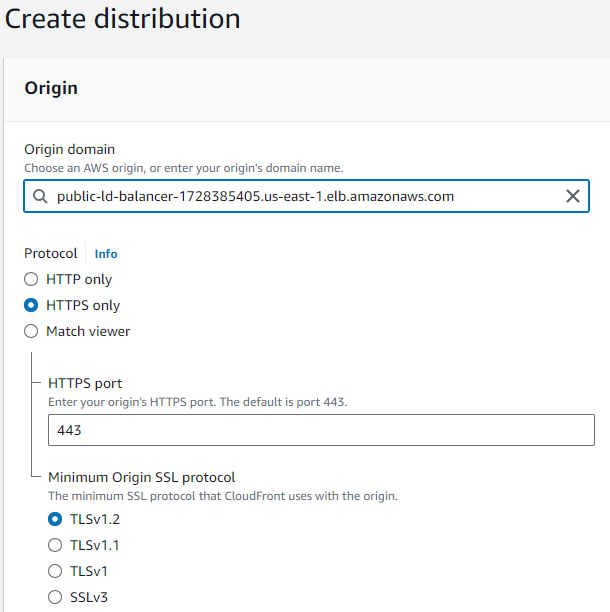
* Now add the certificate in the secure listener settings



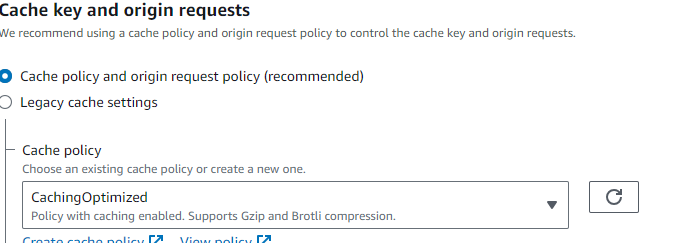
* Click on save changes



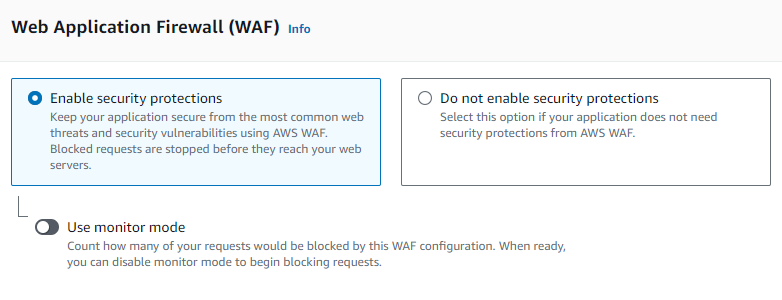
* Next create CloudFront Distribution
* And add load balancer

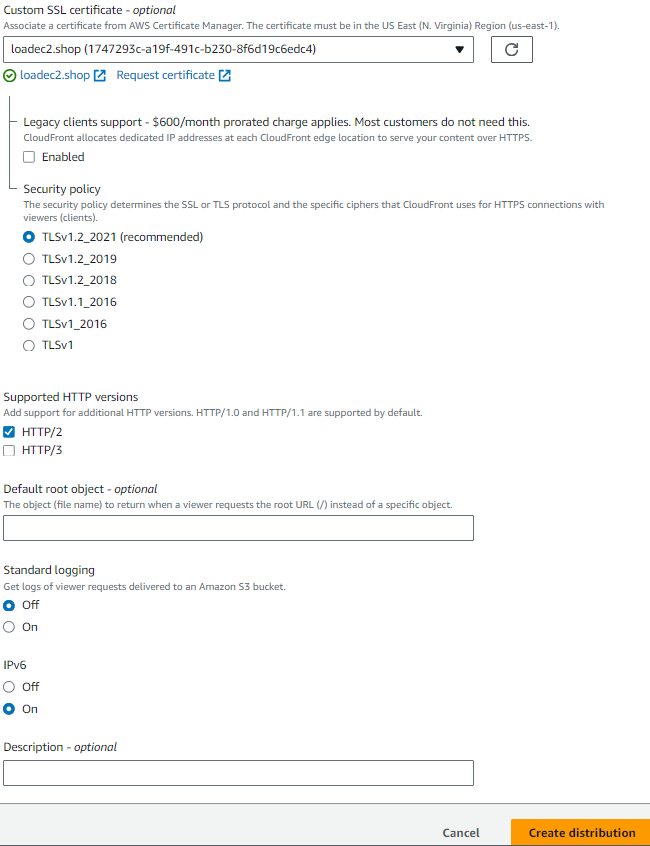


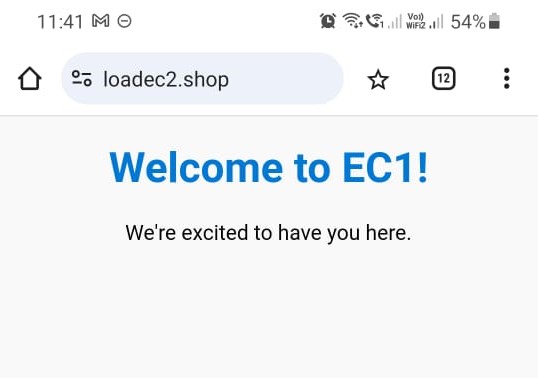
* Add the cache policy-(caching optimized)

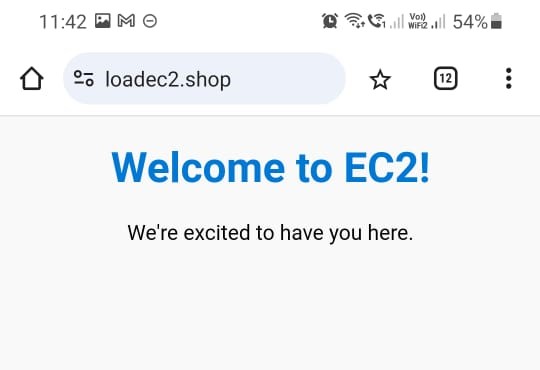


* Enable the WAF protection in the CloudFront.



* Now add the certificate in the setting section-(loade2.shop)
* Next click on create distribution.
* Now check the Domain in the browser – Domain name-(<https://loadec2.shop>)
* That it shows Welcome to Ec1 & Welcome to Ec2 by refreshing

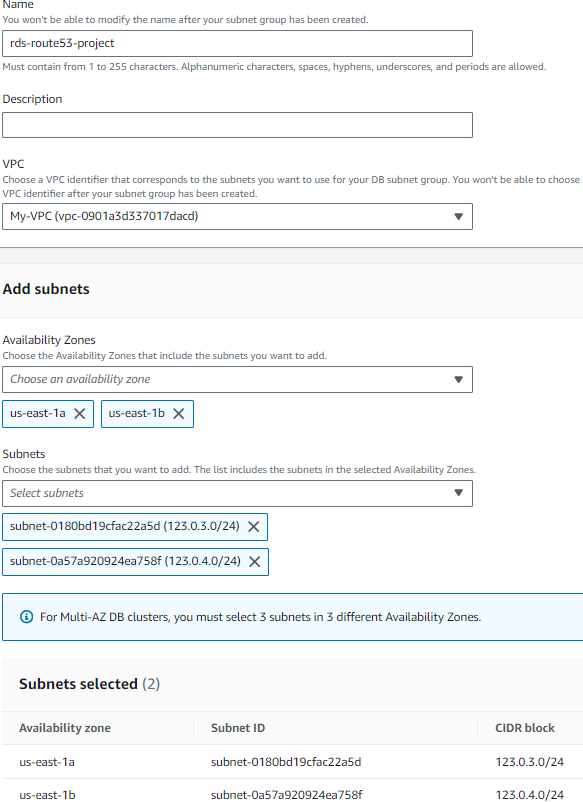




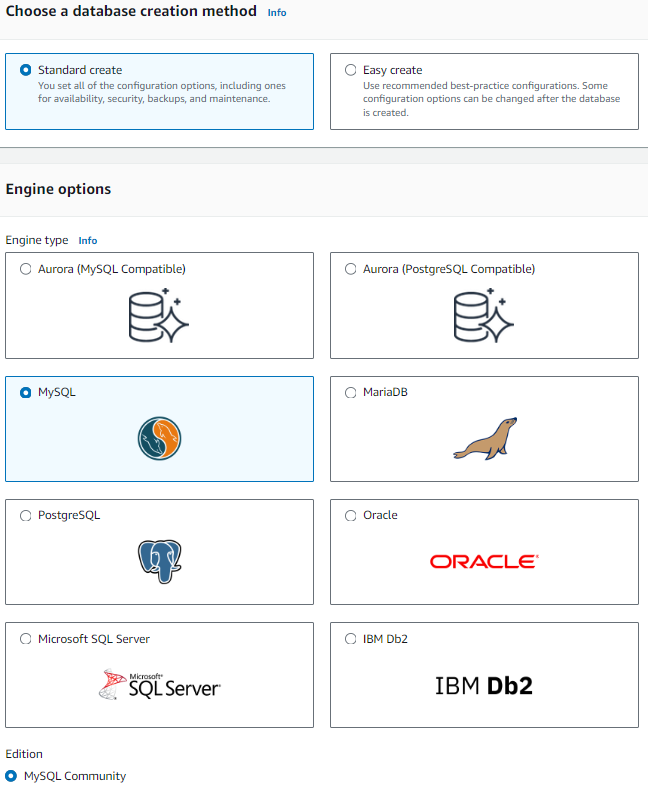
* Next create the RDS in Multi A-Z zone cluster

# RDS

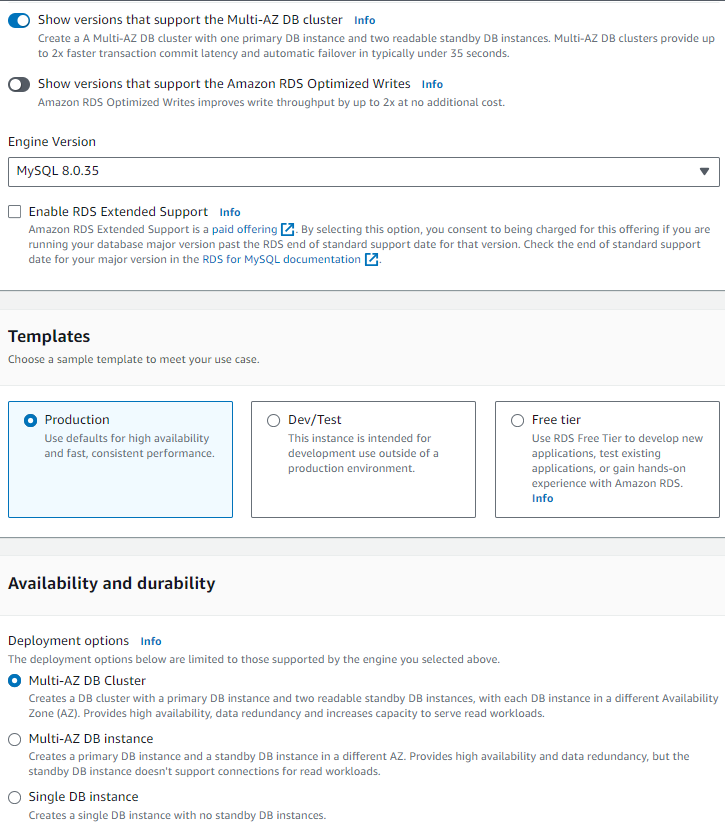
* Next go to RDS and create multi-A-Z zone cluster by adding another subnet separately because we can’t create a multi-A-Z cluster in two subnets.
* First create a DB-subnet group(rds-route53-project), add the Database subnets-(Database-private-1a,Database-private-1b),add the vpc



* Click on create DB-subnet group
* Next create an RDS Data-base-(Private-RDS), with the standard create, MYSQL engine.



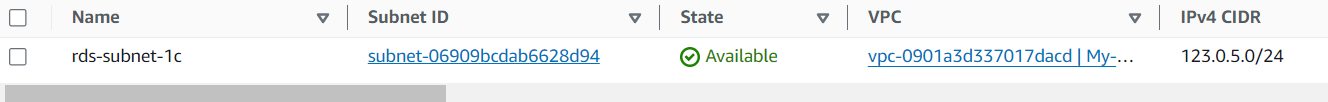
* Enable Multi A-Z zone cluster versioning.
* Template – production tier (for faster creation in Multi A-Z zone)
* In the availability & durability enable – Multi A-Z cluster.



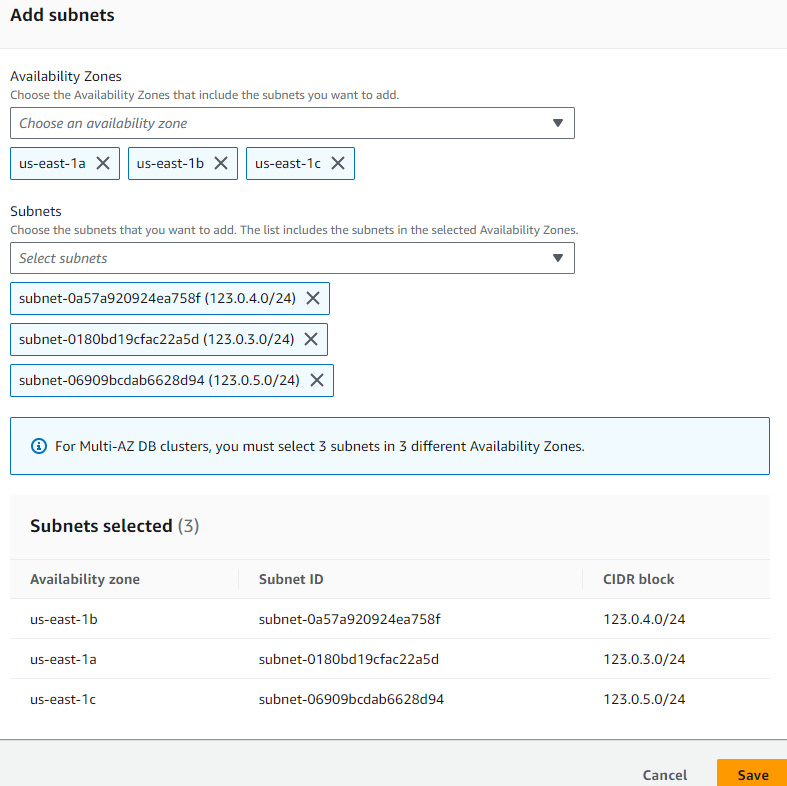
* In settings add the DB-identifier name-(private-RDS), Master username-(admin) by default, enter the master password & confirm it.



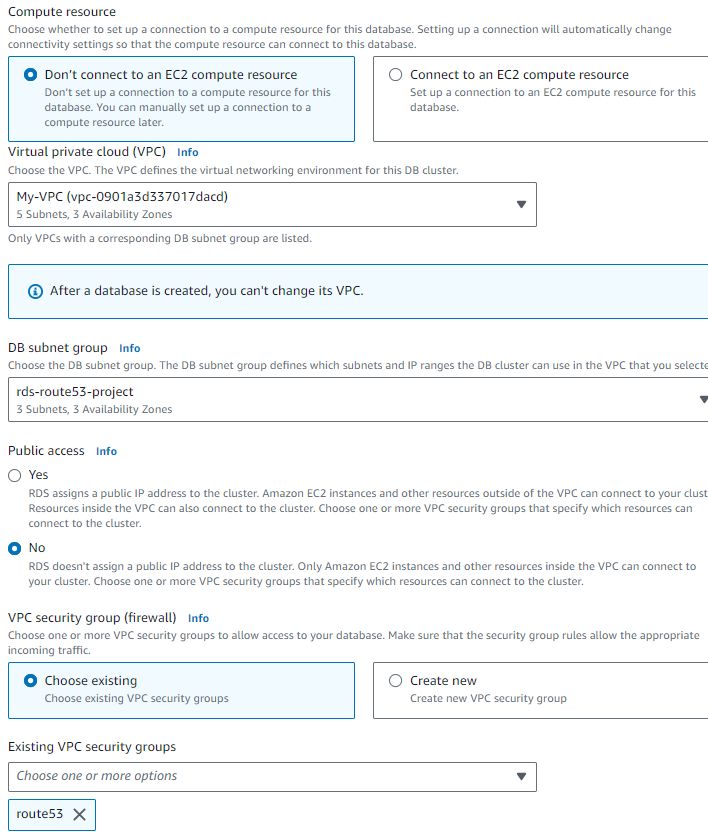
* Now we will get a problem we should go an add another subnet with the same VPC-(rds-subnet-1c)-123.0.5.0/24



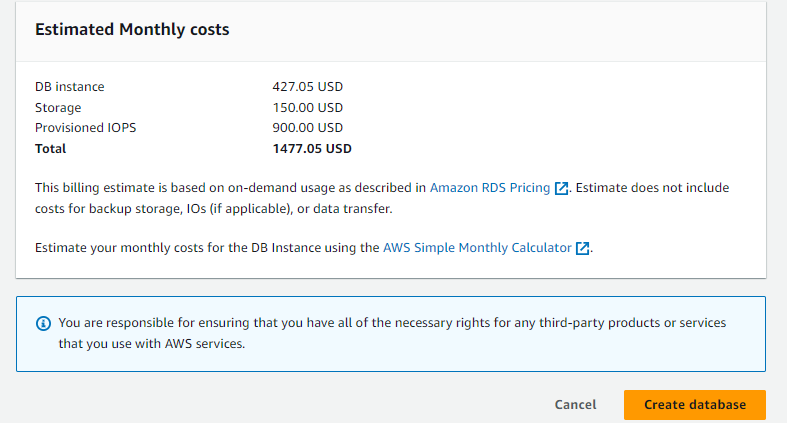
* Next refresh the connectivity Selection in the RDS the VPC with 5 subnets in 3-availability-zones will be reflected
* Next edit the DB-subnet group with the 3 availability zones.



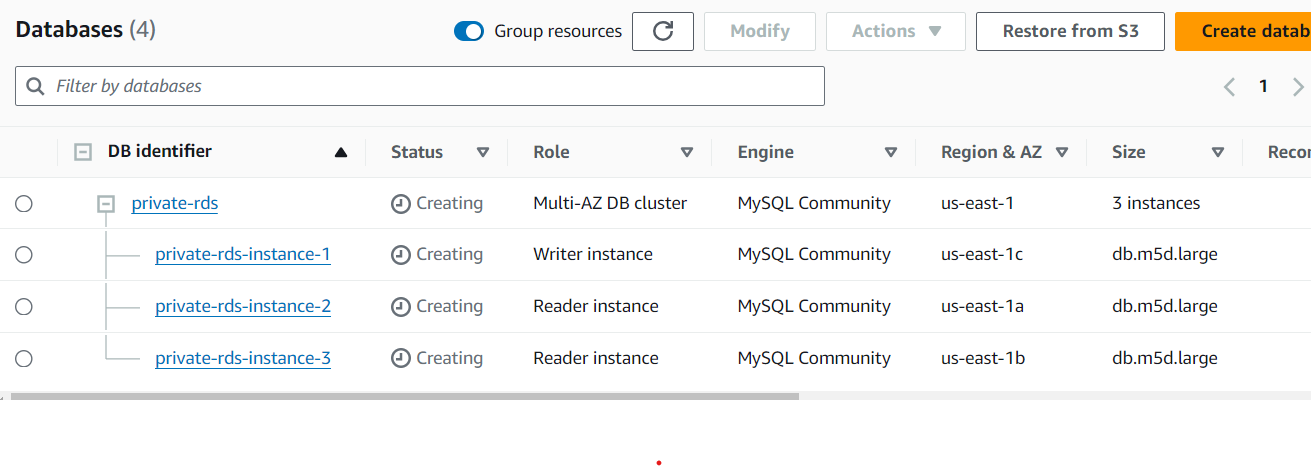
* Next click on save next refresh the connectivity setting in RDS
* Click on don’t connect to Ec2 instance & select the VPC security group



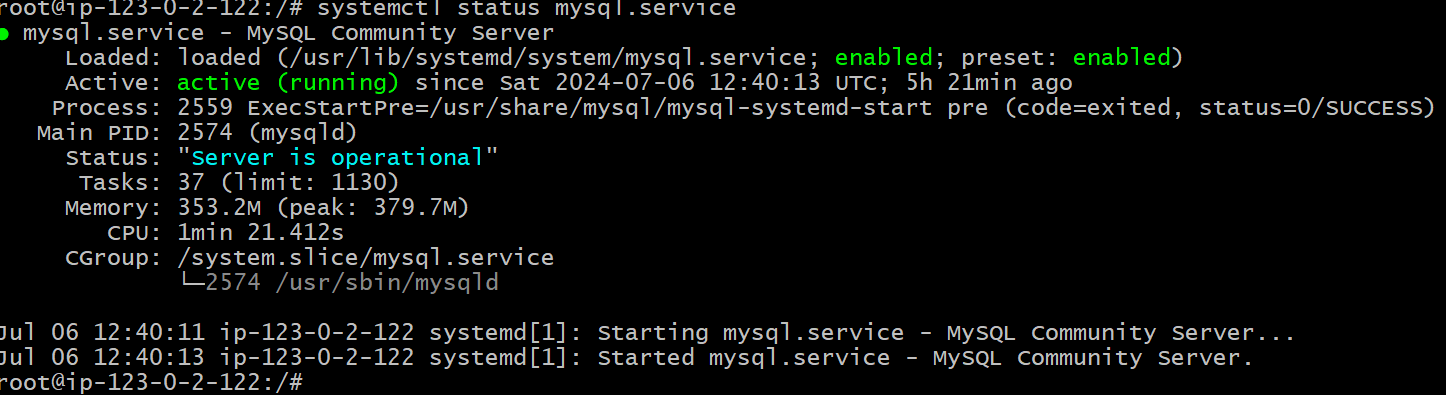
* Next click on click on create database



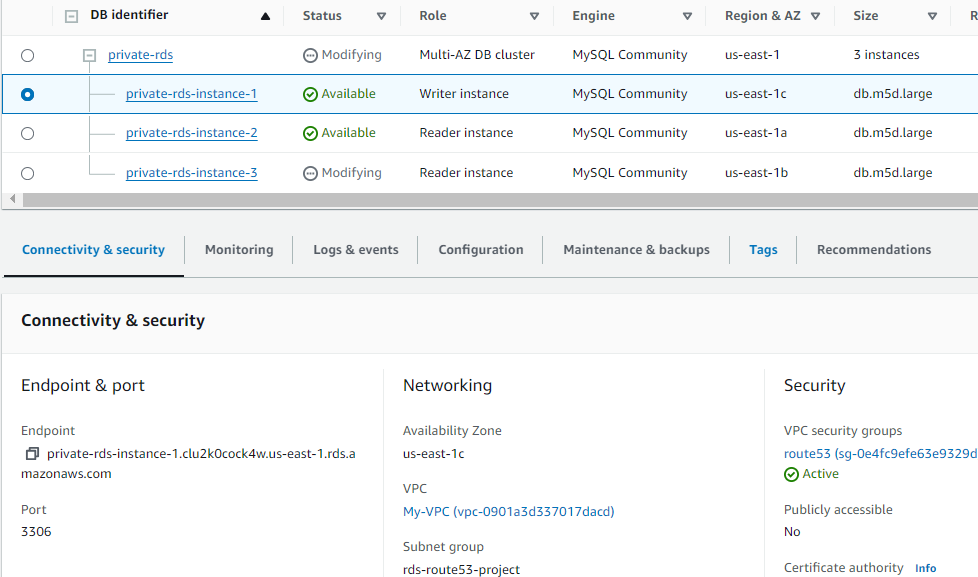
* Wait until the Multi A-Z cluster is in available state.



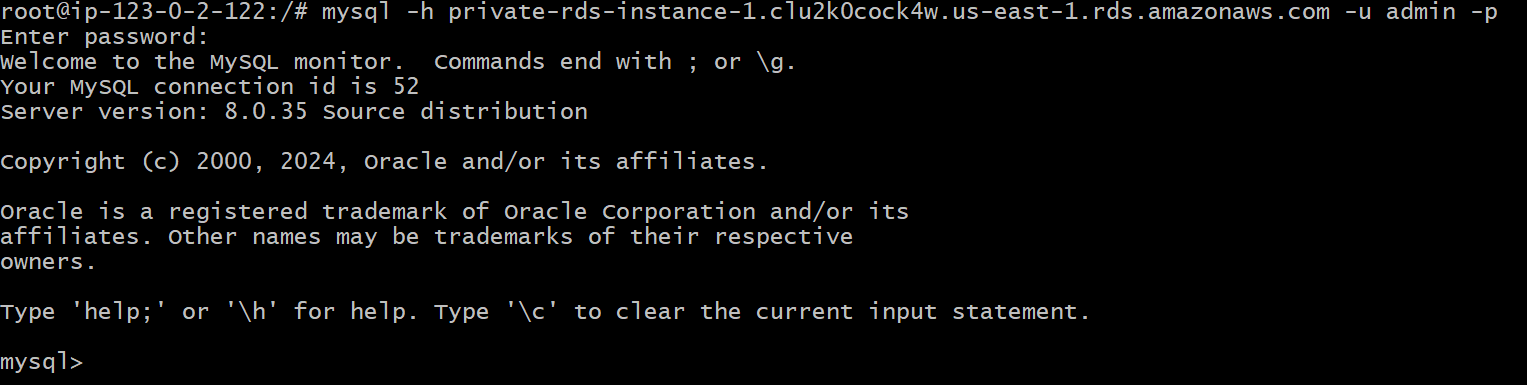
* After the Data base is in available state connect to the writer end point and insert a table
* Next check the reader instance is working
* Check the mysql status in the ec1 instance-(systemctl status mysql.serivce)



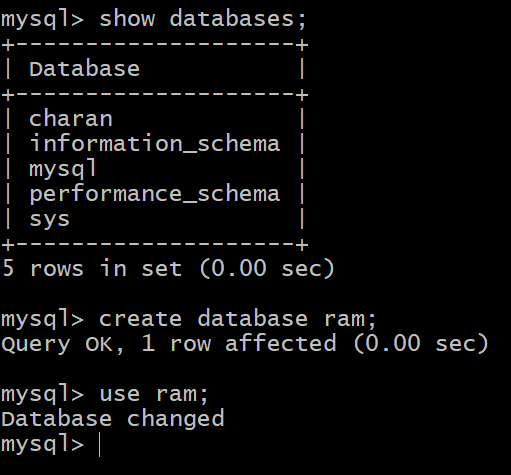
* Now copy the end point of the writer instance from the database & past in the ec1 instance



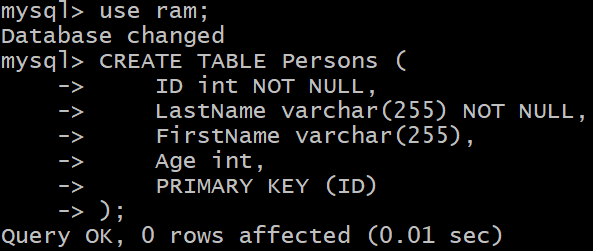
* Path to connect to mysql –(mysql -h private-rds-instance-1.clu2k0cock4w.us-east-1.rds.amazonaws.com -u admin -p) and press enter next type the password it connect to mysql sever.



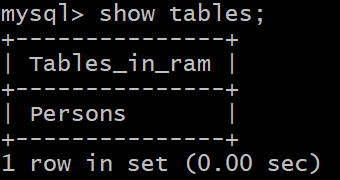
* Click on show database, create a database, use the data base, now create a table in your database (ram).



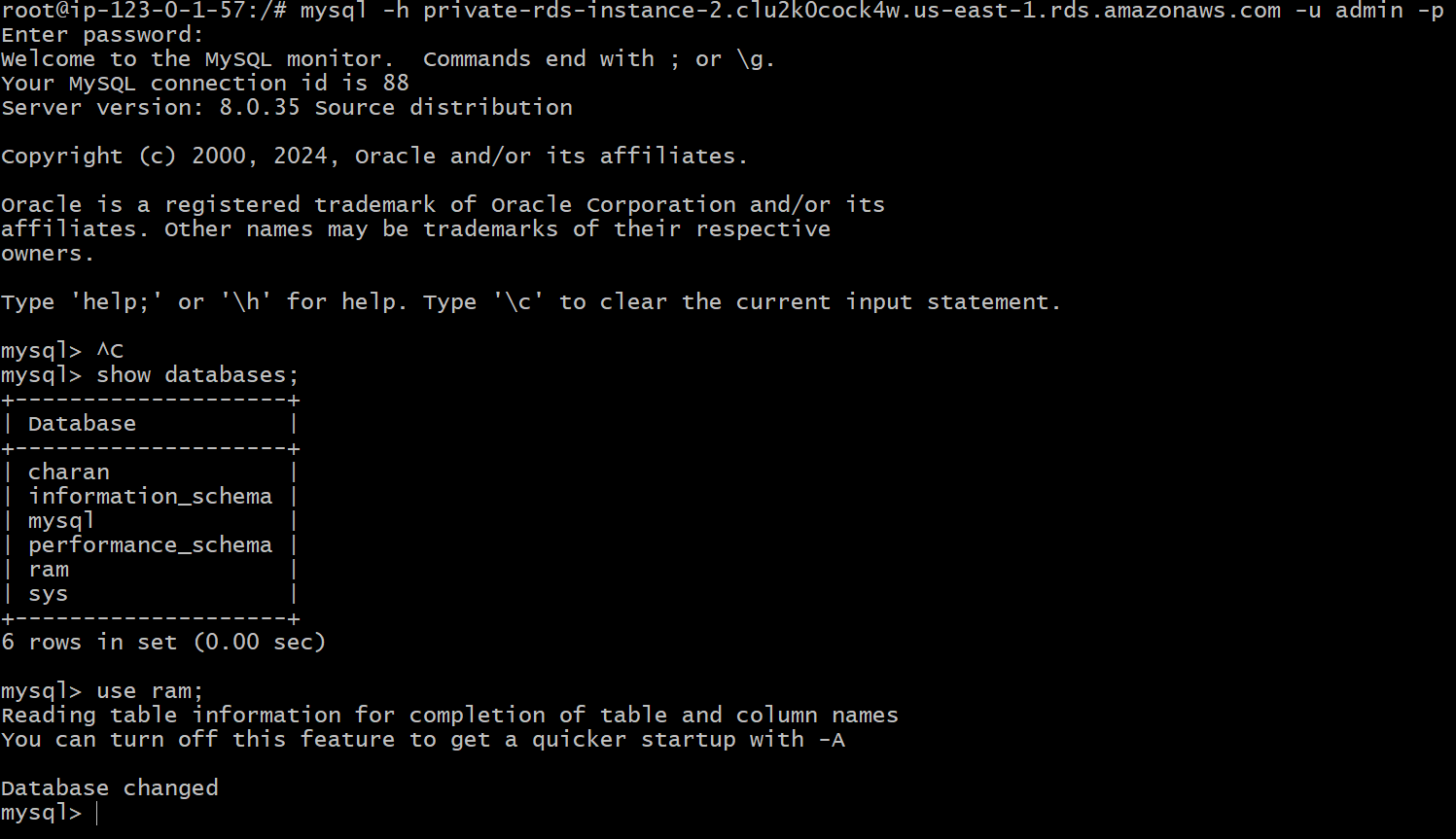
* Now create a table in the data base with primary key, copy from the browser.

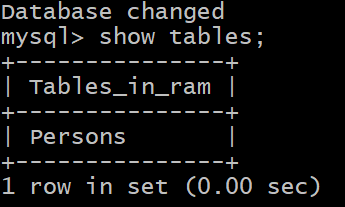


* Now table is created using this command.

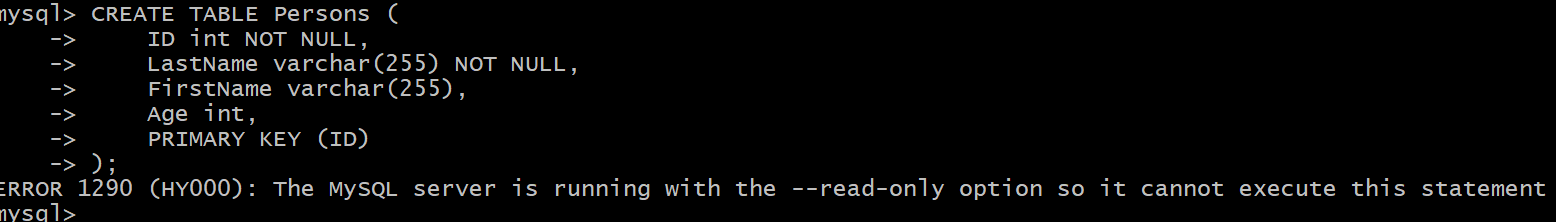


* Now connect to reader instance through the end point & and check the table-( mysql -h private-rds-instance-2.clu2k0cock4w.us-east-1.rds.amazonaws.com -u admin -p).
* Now click on show databases & check the table try to write then an error comes because it’s the reader instance.





* If u try to create a table error come because it’s a reader instance let’s see.



* Finally, it comes with a (error reader) only ---read only access.