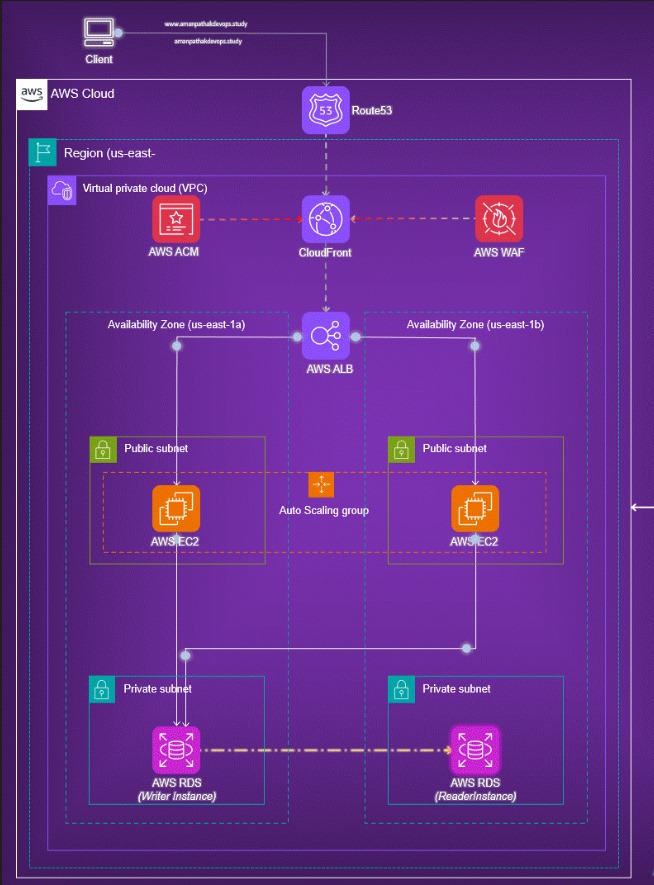
***Name: Chandra sekhar***

***Batch no:125***

***Aws Devops***

**AWS WEB 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.

Got to AWS dashboard search on VPC

1. Click on VPC > Select More VPC, VPC CIDR block (10.0.0.0/16)

2. Select 2-Available Zones ( 1a A”Z “, 1b A”Z” )

**3. Subnet’s** (Public-2, Private-2) **One**-Internet Gateway, **Route table** (1-public, 2-private)

4. Customize subnet CIDR blocks

Public – Subnet -1a **(10.0.1.0/24)**

Private – Subnet -1a **(10.0.2.0/24)**

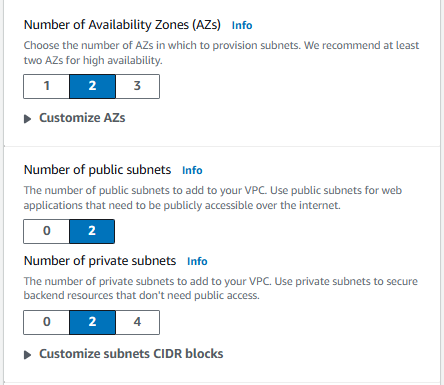
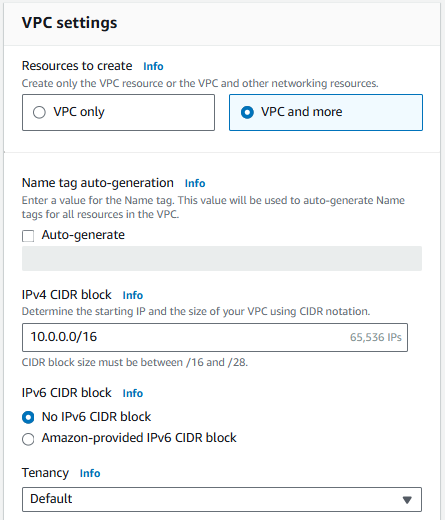
Public -Subnet -1b **(10.0.3.0/24)**

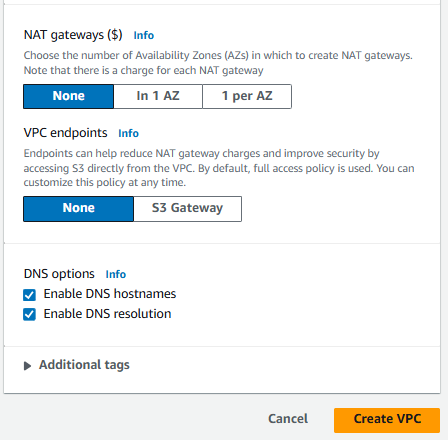
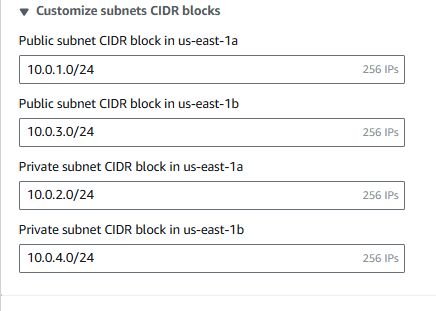
Private -Subnet -1b **(10.0.4.0/24)**

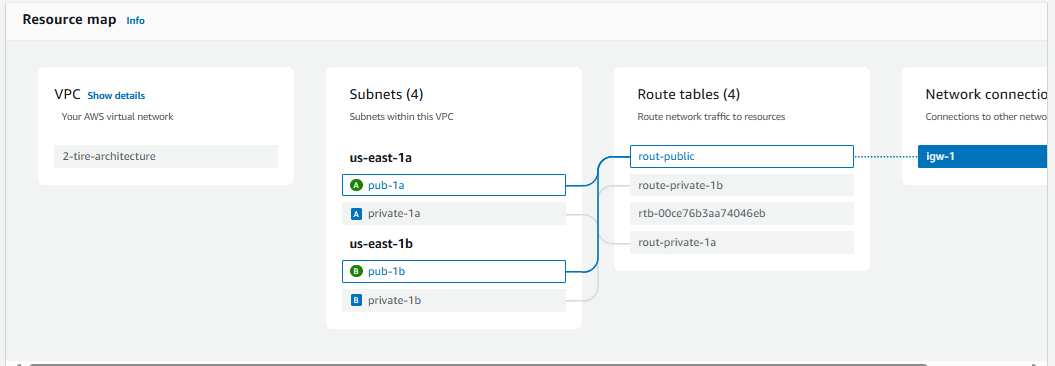
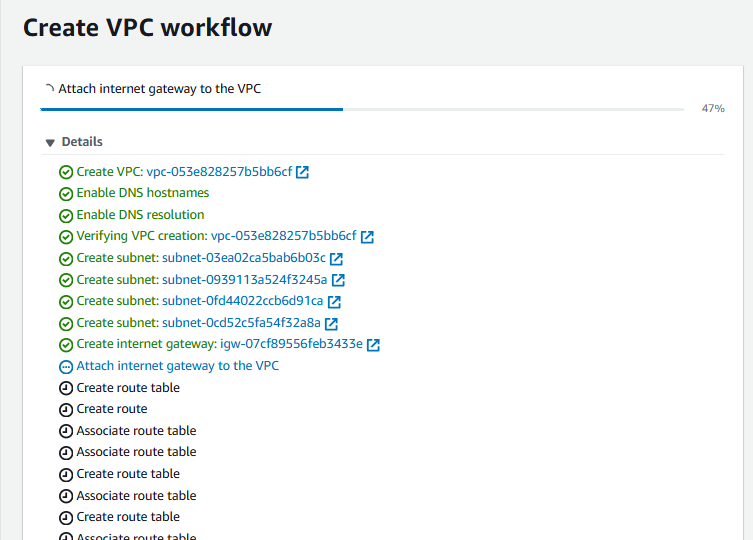
5. Remaining all default Then Click On

6. Create more VPC

(Subnet, Route tables, Internet Gateway, VPC Automatically Assigned As show in figure)







KNOW WE ARE CREATING INSTANCES IN DIFFERENT AVAILABLE ZONES LIKE ( 1a , 1b ) **(1A ) (PRESENTATION / WEB ) TIER**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Got to **AWS dashboard**  2. search  **Ec2** 3. Select: **Ec2**

4. Click on **Launch Instance**,

5. Create Name: **EC-1**

6. Select “App” OS Images: **Ubuntu**, Instance type: **t2.micro**, Keypair **Create-new**

7.Click on **Edit** Networking Settings:

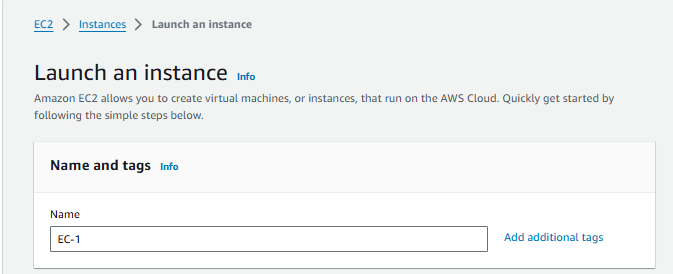
8. Select (VPC) **2-tire-Architecture**, Subnet **Pub-1a**, Auto-assign public IP **Enable**

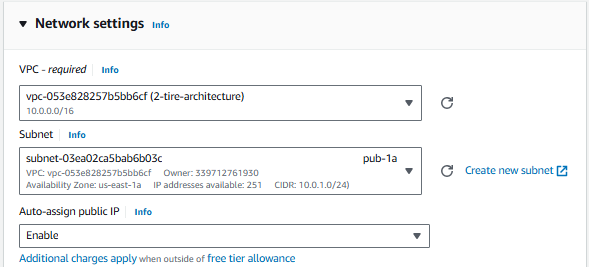
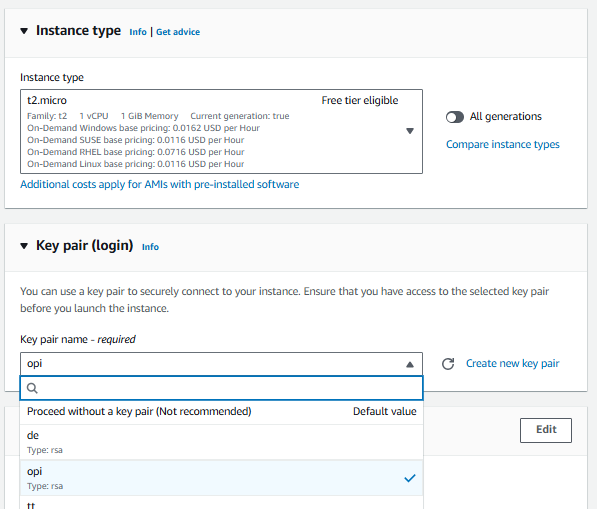
9. Click on create **Security Group** Name : **launch-wizard-125**, Add **INBOUND RULES.**

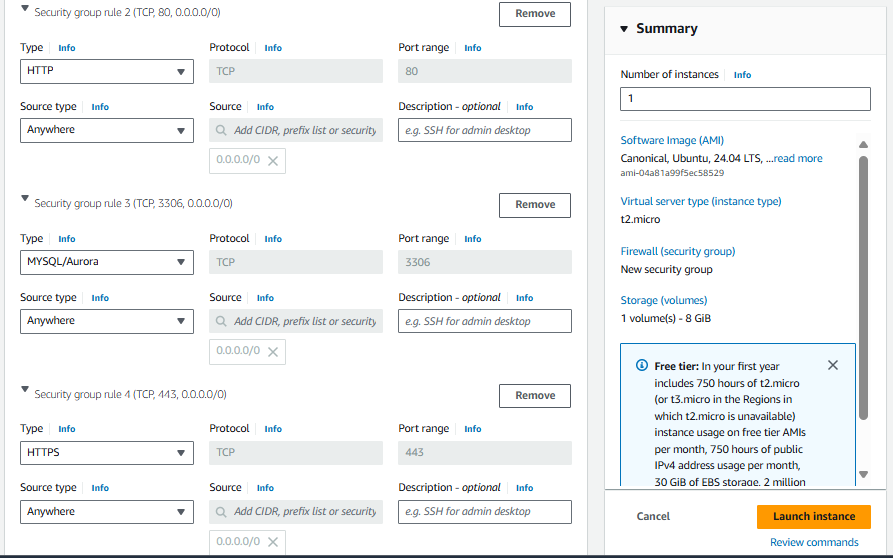
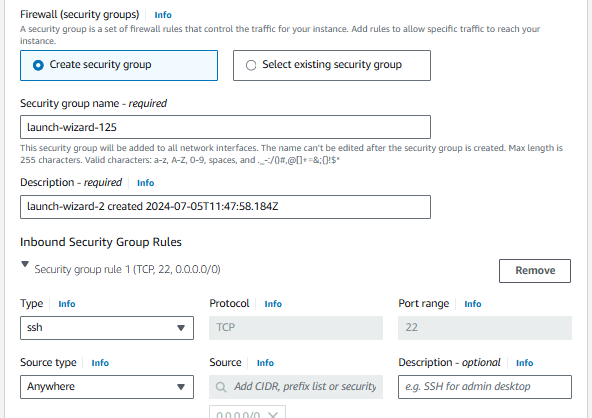
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | Protocol | Port Range | Source Type | Source |
| **ssh** | TCP | **22** | Anywhere | 0.0.0.0/0 |
| **HTTP** | TCP | **80** | Anywhere | 0.0.0.0/0 |
| **HTTPS** | TCP | **443** | Anywhere | 0.0.0.0/0 |
| **MYSQL/Aurora** | TCP | **3306** | Anywhere | 0.0.0.0/0 |

10. CLICK ON **LAUNCH INSTANCE.**

**( As so in the Below Pictures )**

****





\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(1B ) (PRESENTATION / WEB ) TIER**.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Got to **AWS dashboard**  2. search  **Ec2** 3. Select: **Ec2**

4. Click on **Launch Instance**,

5. Create Name: **EC-2**

6. Select “App” OS Images: **Ubuntu**, Instance type: **t2.micro**, Keypair **select ( EC-1)**

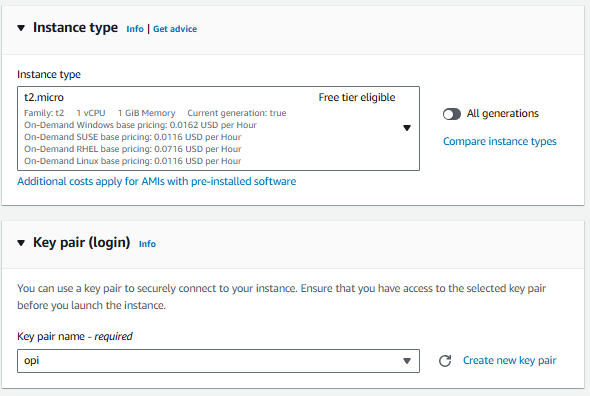
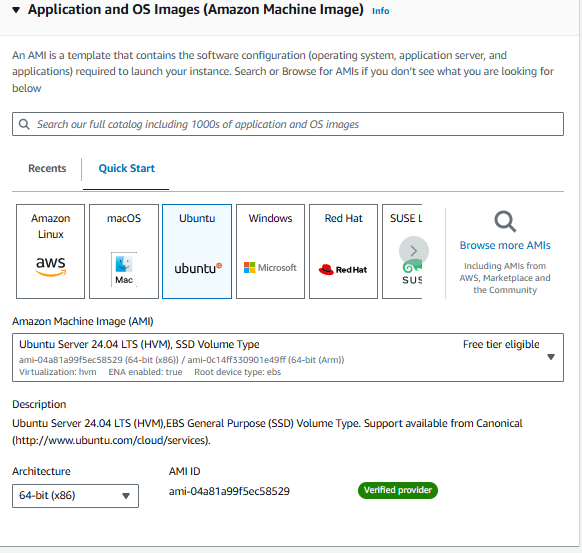
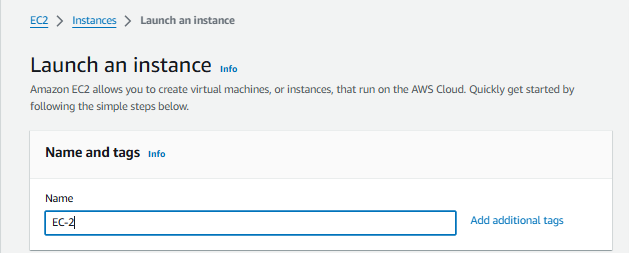
7.Click on **Edit** Networking Settings:

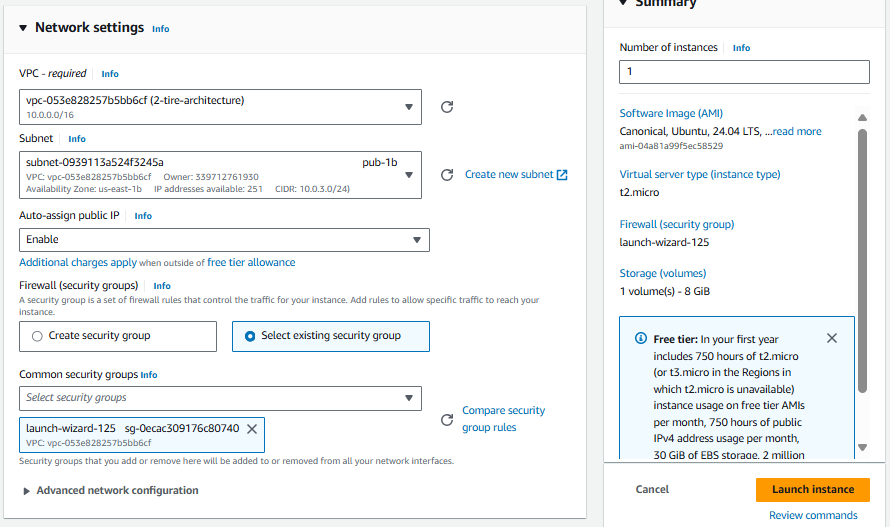
8. Select (VPC) **2-tire-Architecture**, Subnet **Pub-1b**, Auto-assign public IP **Enable**

9. Click on select **Existing Security Group,** Select: **(launch-wizard-125 )**

10. CLICK ON **LAUNCH INSTANCE.**

**( As so in the Below Pictures )**

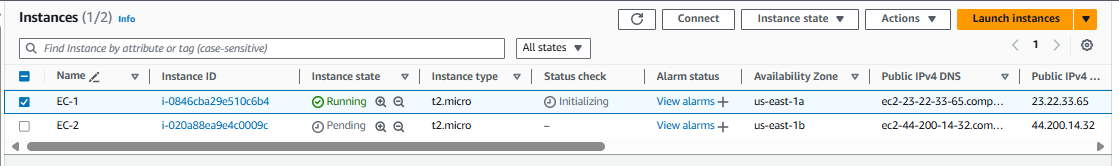




We are Creating Two Instance Like This As show in the Picture.

1a Available Zone instance is **( EC-1 )**

1b Available Zone Instance is **( EC-2 )**



**Installing NGINX Servers & Installing Database MYSQL both Instances**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🡪sudor -i

🡪apt update -y && apt install nginx -y && apt install mysql-server && cd /var/www/html

🡪cat > index.html

( I’m inserting html code in the **index.html** file I want output for **Welcome to EC-1 )**

🡪systemctl restart nginx && systemctl status nginx

Same Step’s Both instance connect

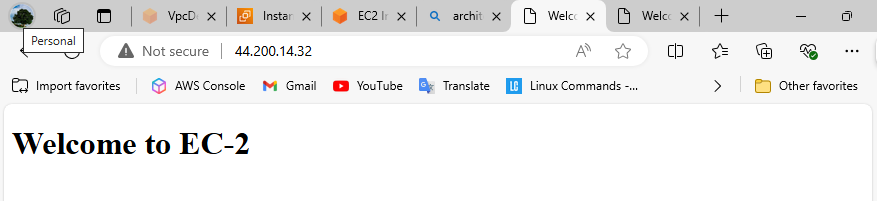
As show in the below Picture is EC-1 Public (1a)



**EC-1 Instance output below Picture in the Place of URL (Public IP address EC-1)**



**EC-2 Instance output below Picture in the Place of URL (Public IP address EC-2)**



**CREATING TARGET GROUP & LOAD BALANCER USING WITH BOTH INSTANCES**

Scroll down in the EC-2 Dashboard

1. Click on **Target group**,

2. Click on create target group,

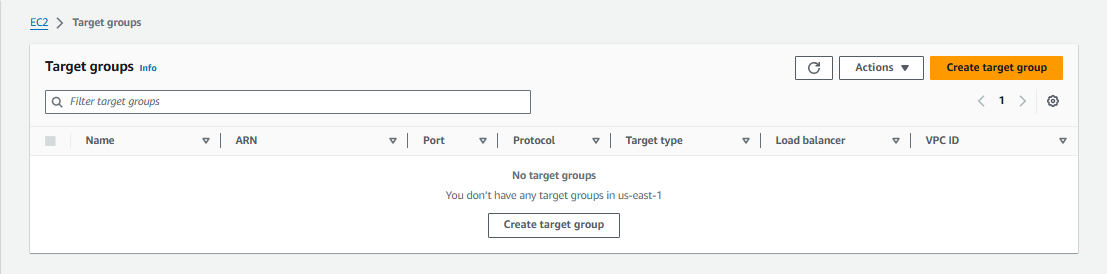
3. Choose a target type **INSTANCES,**

4. Target group name: **Public-target** And Select VPC: **( 2-tire-Architecture ),**

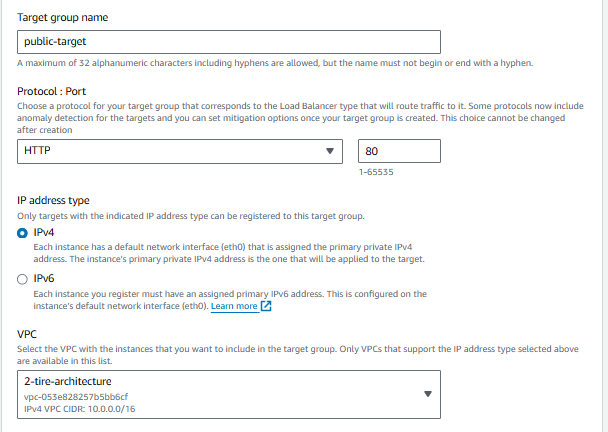
5. Select Available Instance [ **(EC-1)**, **(EC-2)** ],

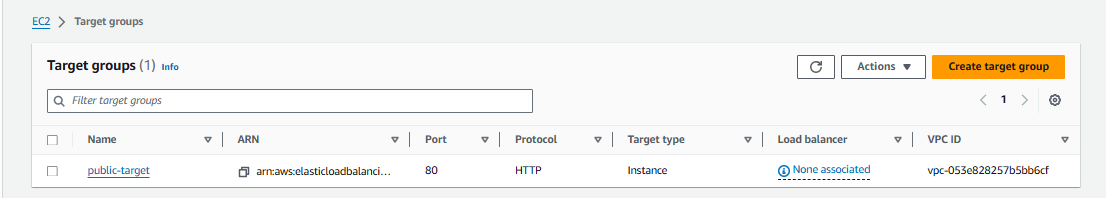
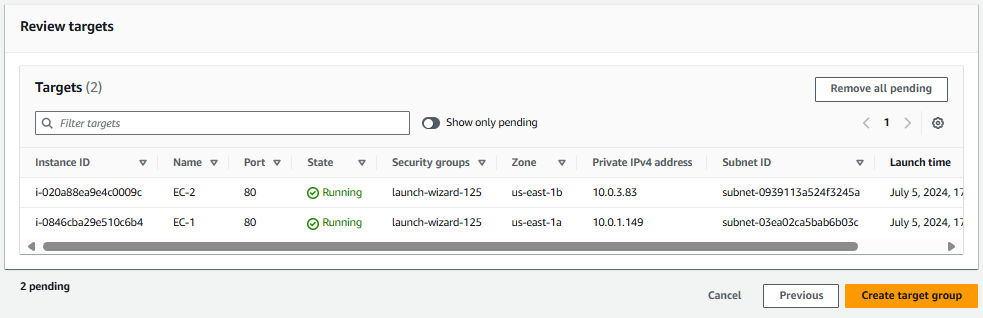
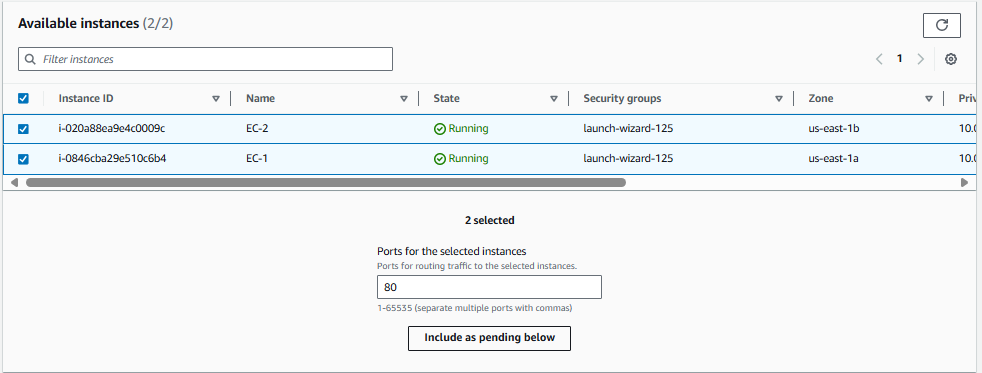
6. Click on **Include as pending below** button,

7. Click on **Create target Group.**









**CREATING LOADBALANCER**

Go to Ec2 Dashboard scroll down Select **load balancer**

1. Click on create **(** **Load balancer )**

2. Select **(** **Application load balancer )**

3. Click on **( Create** )

4. Enter your Load Balancer Name**: (** **myloadbalancer )**

5. Select [ **VPC (2-tire-architecture) ]**

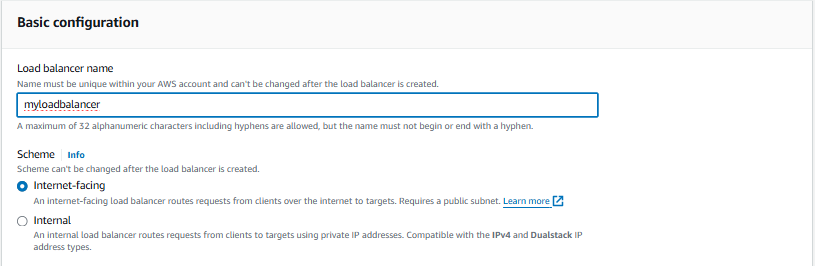
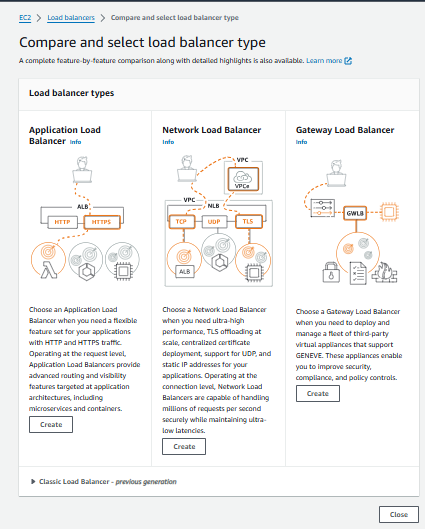
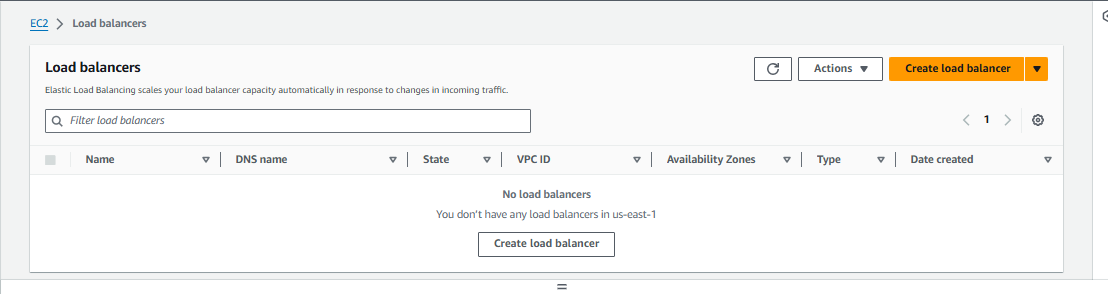
Select subnet **[ us-east-1a (use-az6)** **Pub-1a** ]

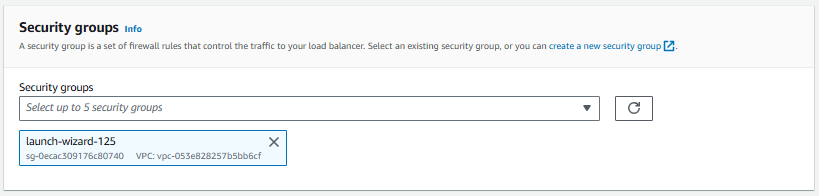
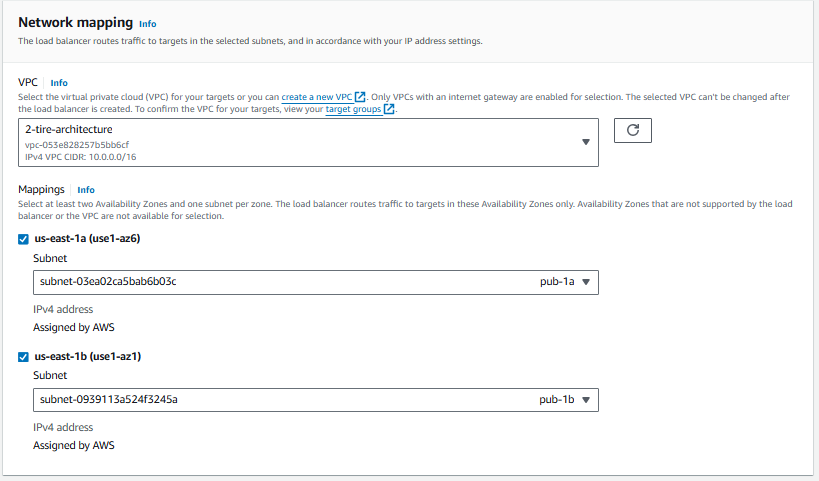
Select subnet **[ us-east-1b (use-az6) Pub-1b ]**

6. Select security Groups **(Launch-wizard-125 )**

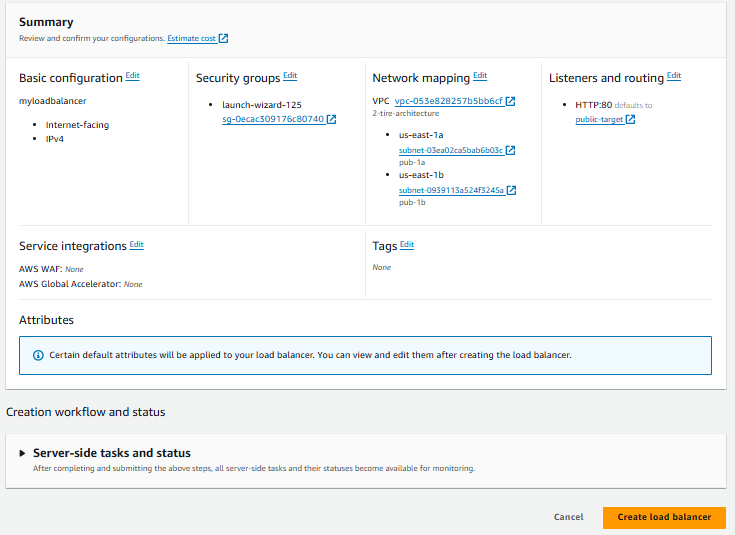
7. Select Target Group **(** **Public-target )**

8. Click on **( create** **load balancer )**









**CREATING A LAUNCH TEMPLATE**

Before creating Auto scaling group, we can create Launch Template

Go to Ec2 Instances dashboard

1. Select launch template

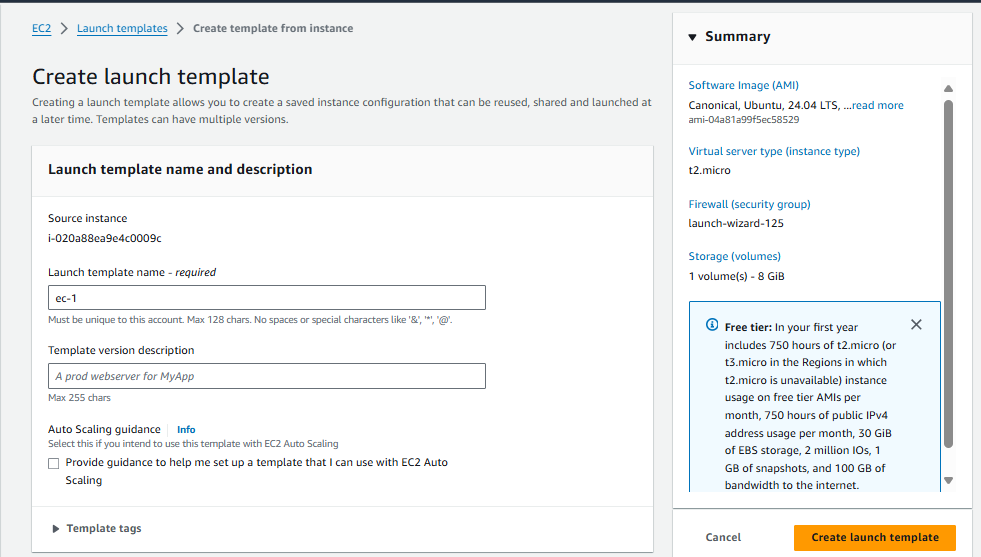
2. Click on create launch template

3. click on name dashboard

4. Type launch Template Name: **Ec-1**

Note:( Reaming all details default )

5. click on Create **Launch Template**



**Know creating Autoscaling Group**

Scroll down in the EC-2 Dashboard

1. Select **Autoscaling Group**

2. Click on **Create Autoscaling Group**

3. Autoscaling Group Name :  **ASG**

4. Choose Launch template **(ec-1)**

5. Click on **NEXT**

6. Select **VPC (2-Tire-Architecture)**

7. Select **Available Zone and Subnets (Pub-1)**

8. Click on **NEXT**

9. Select **Attach to** **Existing load balancer**

10. select **Choose from your balancer target group**

11. Existing load balancer target group **public-target**

12. Group size

**Desired capacity type: (1 )**

**Min desired capacity : (1 )**

**Max desired capacity : ( 2 )**

13. Click on **NEXT**

14. SNS ( optional )

Send a notification to **: ASG-instance**

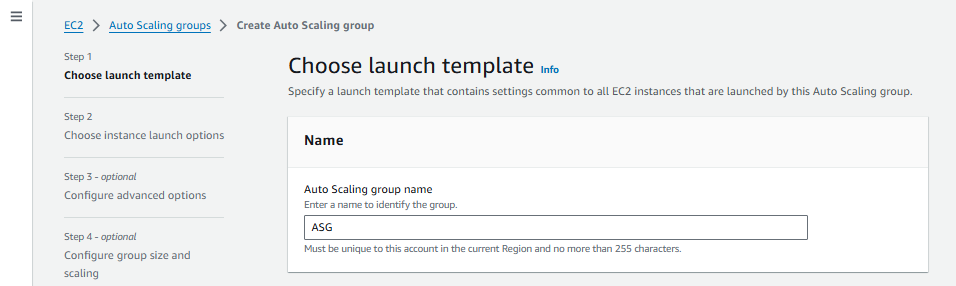
With these recipients : [example@gmail.com](mailto:example@gmail.com) Event type : **Select All**

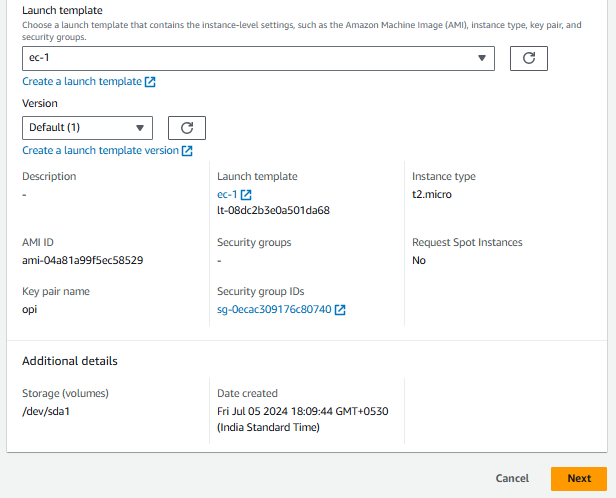
15. Click on **next**

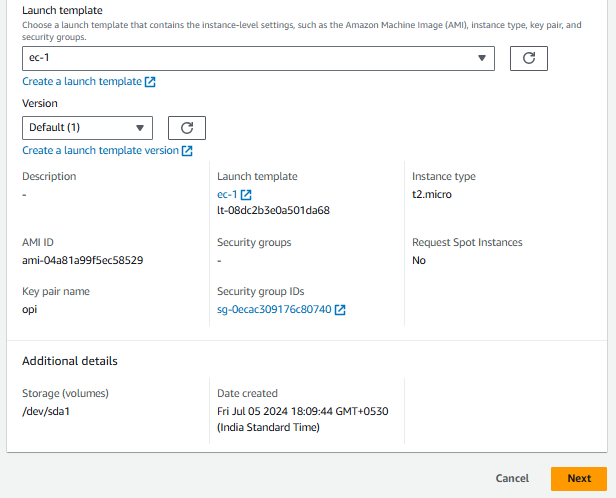
16. Click on **Next** (Tag Optional)

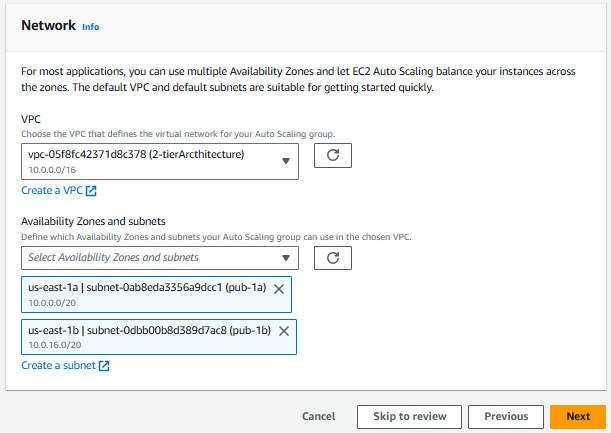
17. Click On Create **Auto scaling group** ( Check Review then )

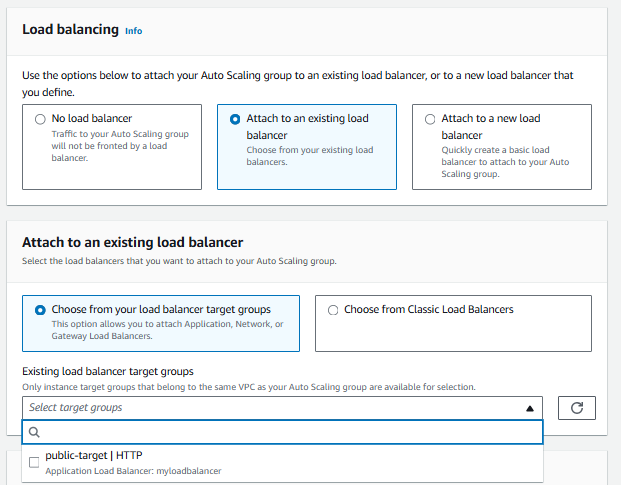
**As show in the Picture Explaining Autoscaling group**

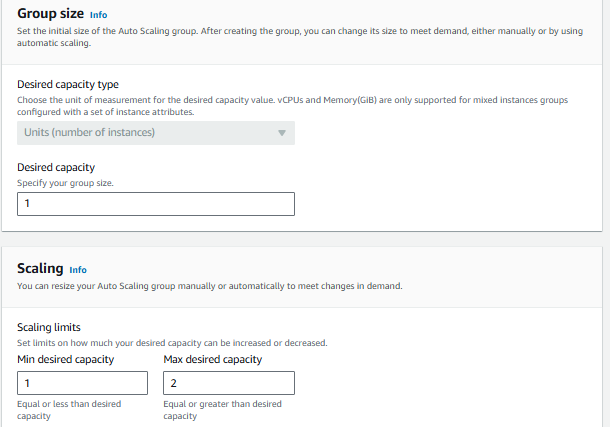


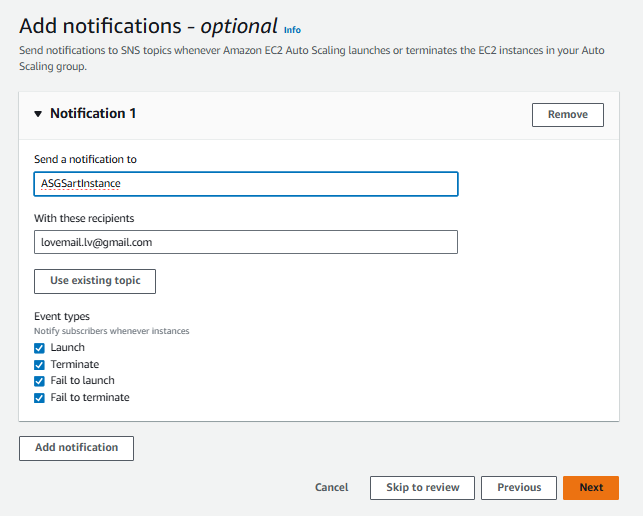


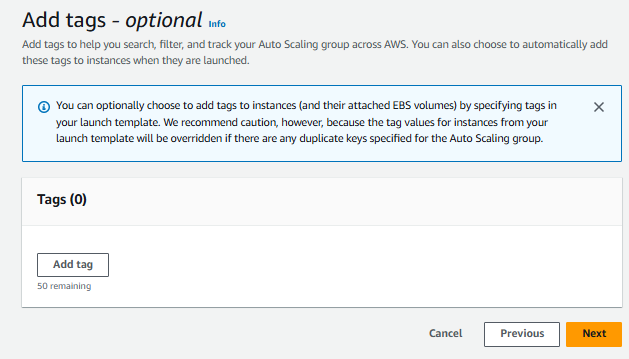




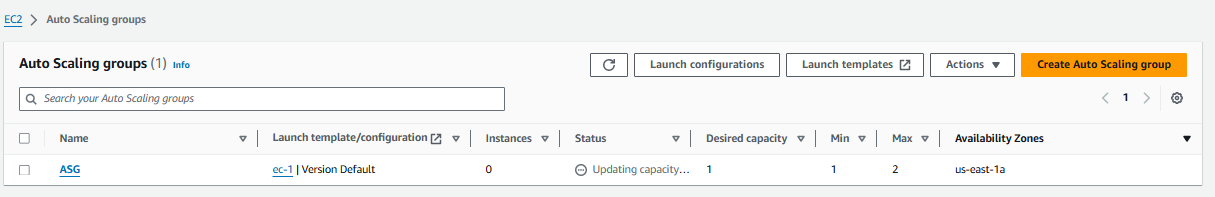


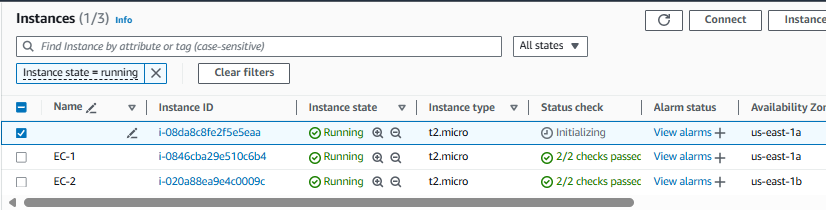
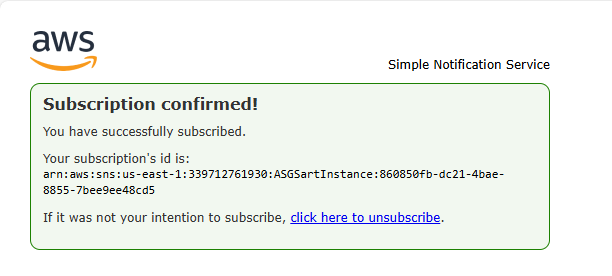
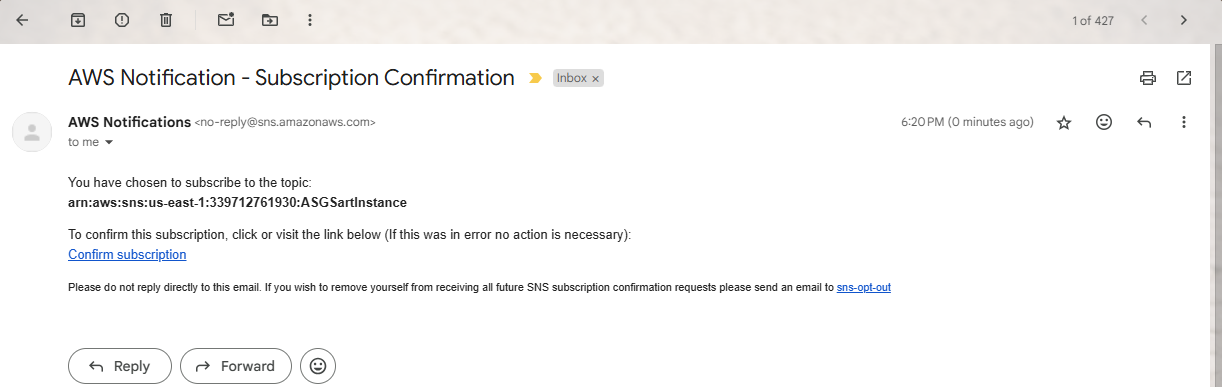












**CREATING ROUTE 53 AND ACM & CLOUDFRONT**

**Creating Route 53:**

Go to search bar type Route 53 then click on it

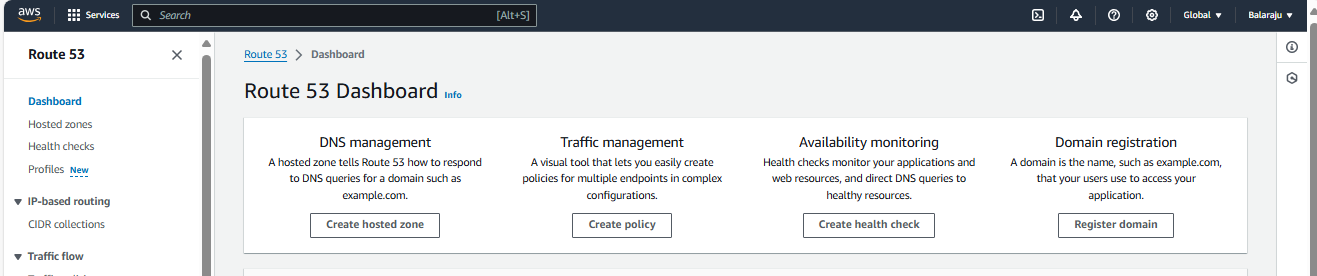
We have an already Domain We directly go to DNS management

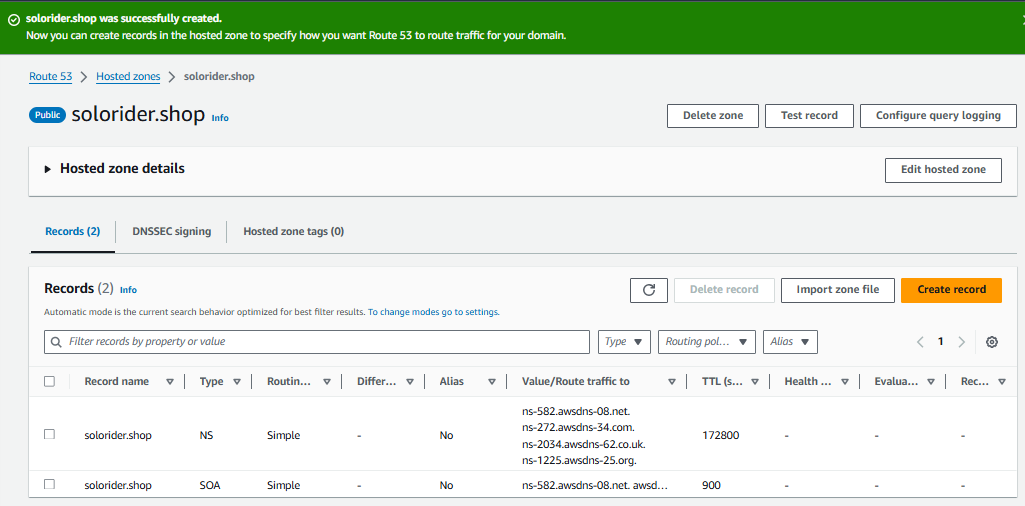
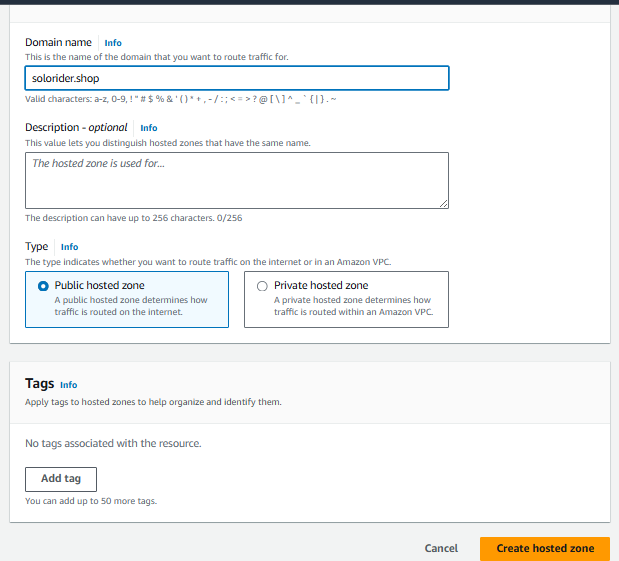
1. Click on **(** **Create hosted zone )**

2. Enter your Domain Name: **( solorider.shop)**

3. Select type**: (Public hosted zone)**

4. Click on **(** **create hosted zone )**

****



**After creating Route 53**

1. Click Create Record

2. Click on Enable

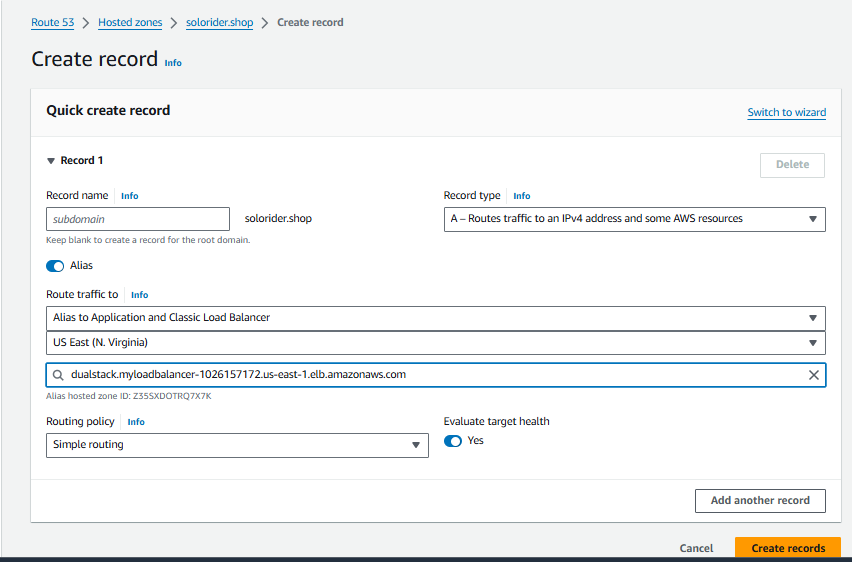
3. Click on Alias Button Enable it

4. Select Route traffic to Hear we are attaching load balancer

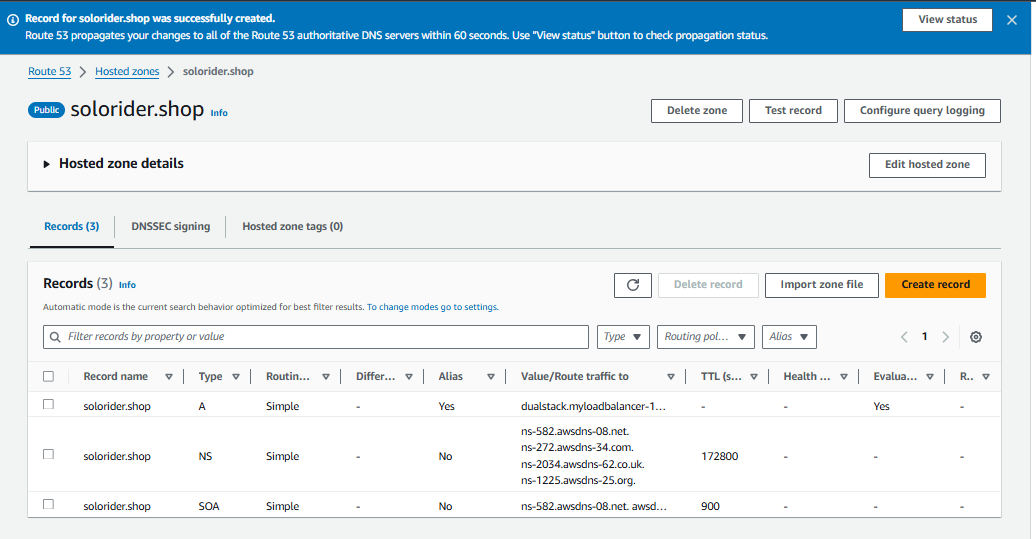
Choose your endpoint : **( Application classic load balancer )**

Choose Your Region : **( US East (N. Virginia )**

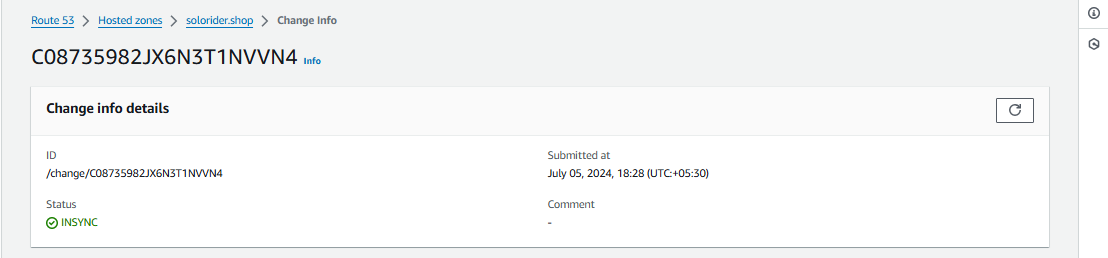
Choose your load balancer : **( myloadbalancer )**

****

One Record created in the records table Record name: **solorider.shop** Type( **A )**



Wait For few seconds then it’s Status **INSYNC**



**Copy the Route 53 NS (Name servers) and Edit and Paste in the**

**Personal Domain Name servers**

**Route 53** created

**[** Type **NS** (Nameservers), **Value/Route traffic to ]**

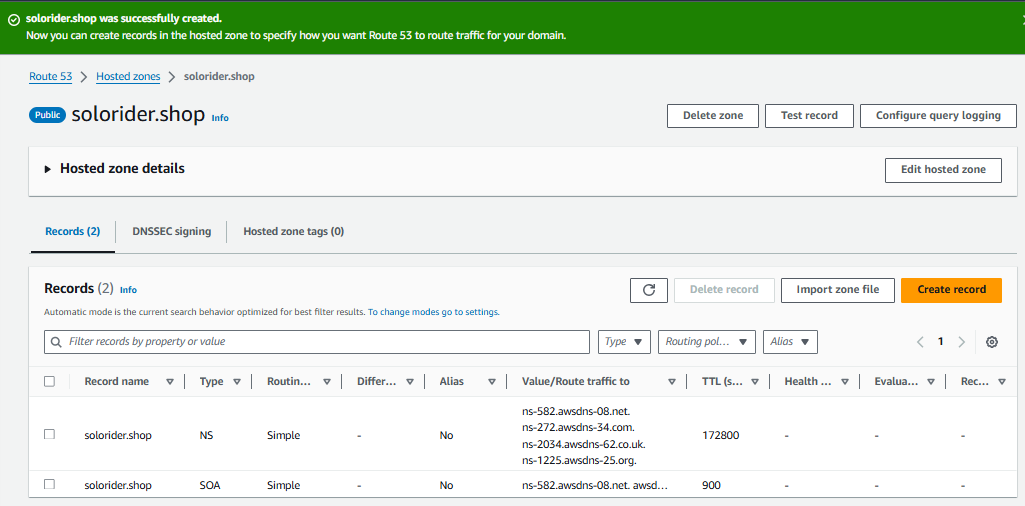
ns-582.awsdns-08.net

ns-272.awsdns-34.com

ns-2034.awsdns-62.co.uk

ns-1225.awsdns-25.org

Copy All the Date without . (dot) paste in Domain NS



I’m buying one Domain Name in **Godaddy website**

1. Login **Godaddy website**

2. Go to **My Products**

3. Click on

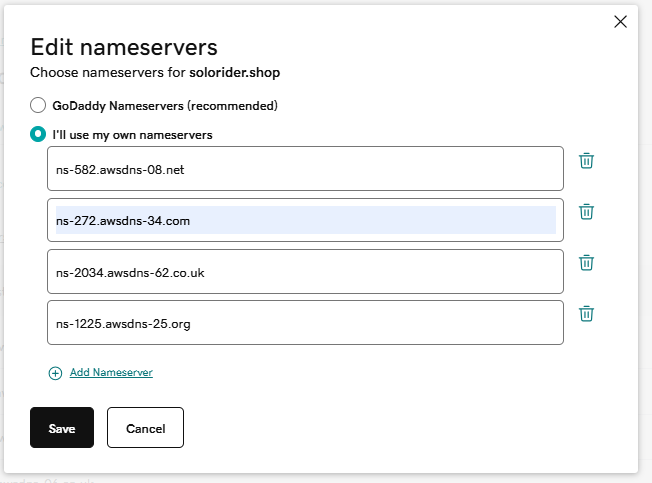
**[** 1. **Domain 🡪** 2. **Domain** **setting** 🡪 3. **DNS**, 🡪 4. **Nameservers 🡪 5. Change**

**Nameservers ]**

4. select **I’ll use my nameservers** Paste in the Domain NS All copy data one by one

5. Click on **Save**

Like as show in the Figure



**CREATING ACM (CERTIFICATE MANAGEGER)**

Go to search bar type ACM then click on it

Click on **Request Certificate**

Select Certificate type: **Request a public certificate**

Click on **Next**

Enter name : **solorider.shop,** Validation method: **DNS validation**

Key algorithm **RSA 2048** (Default)

Click on **Request**

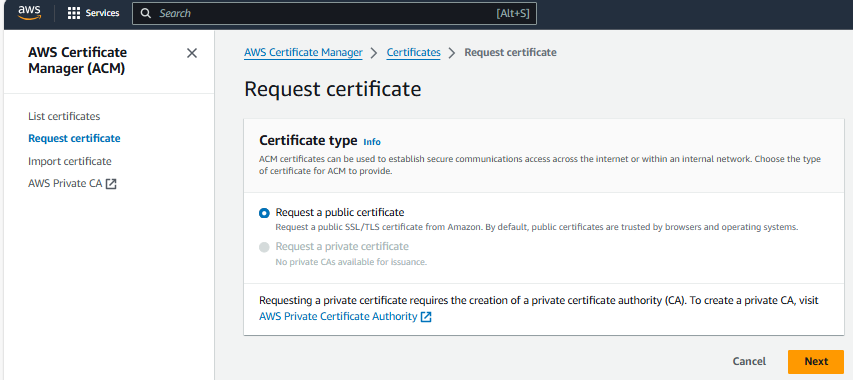
After Completed Request

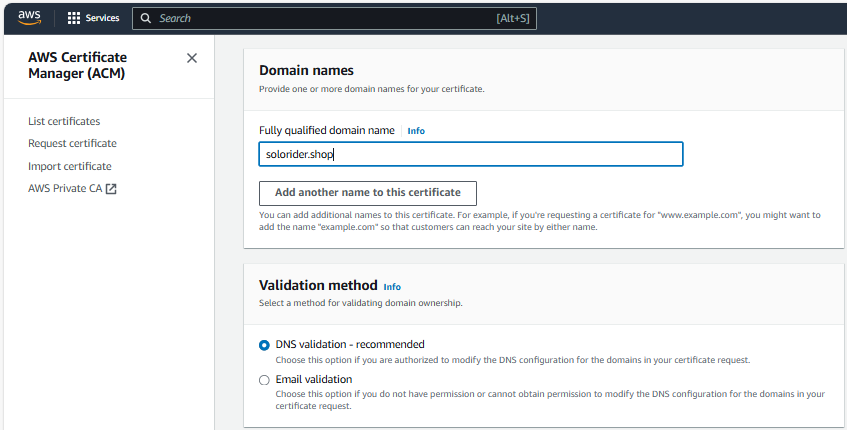
Click on **Create records in Route 53**

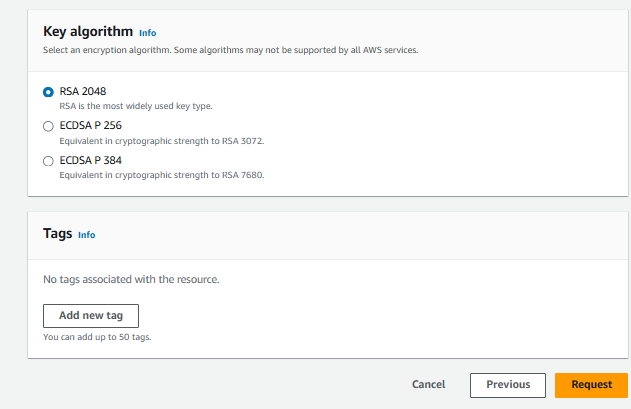
Select **Domain**  (solorider.shop)

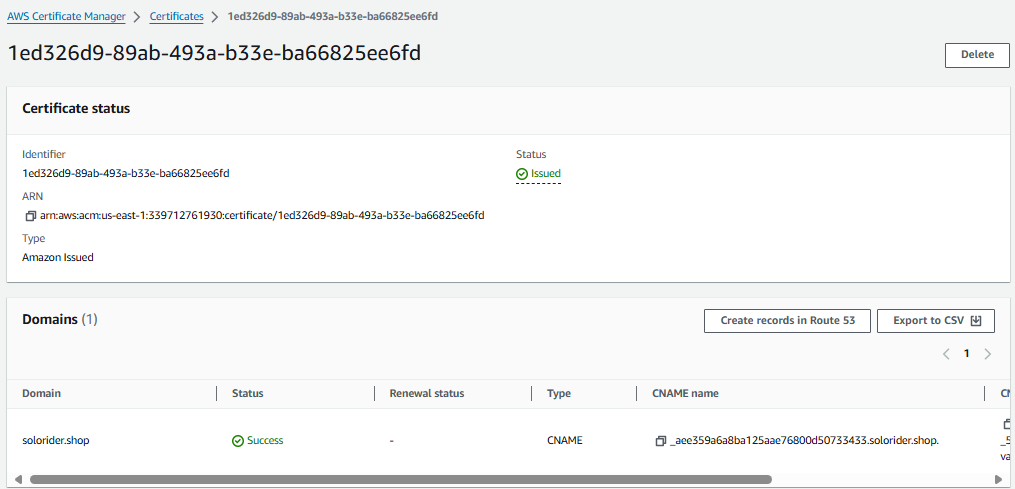
Click on **Create records**

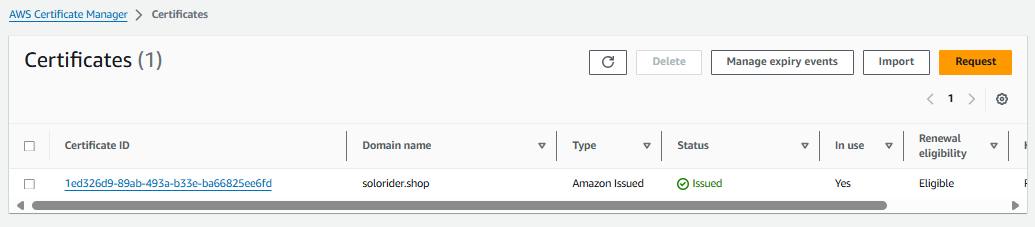
Certificate Creating **Few mints wait for up to status is Issued**

****









**CREATING CLOUDFORNT AND ATTACH LOAD BALAMNCER , CERTIFICATION, WAF (FIREWALL)**

1. Click on C**reate CloudFront distribution.**

2. Select Origin domain: **( myloadbalancer )**

(Note:- don’t change default values after selecting loadbalancer )

3. Go to cache key and origin request Dashboard

Select Cache policy:  **CachingOptimized**

4. Got to Custom SSL certificate optional Dashboard

Choose certificate: **Solorider.shop**

**( Note :-** don’t change any values after upload certificate )

5. Go to Security dashboard

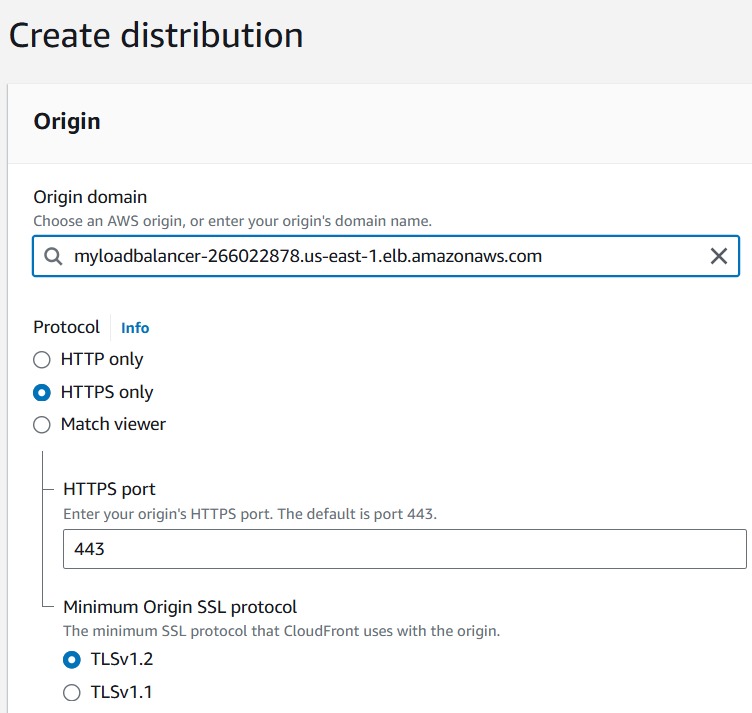
Select **Enable** Web Application Firewall **(WAF): ( Enable security protections )**

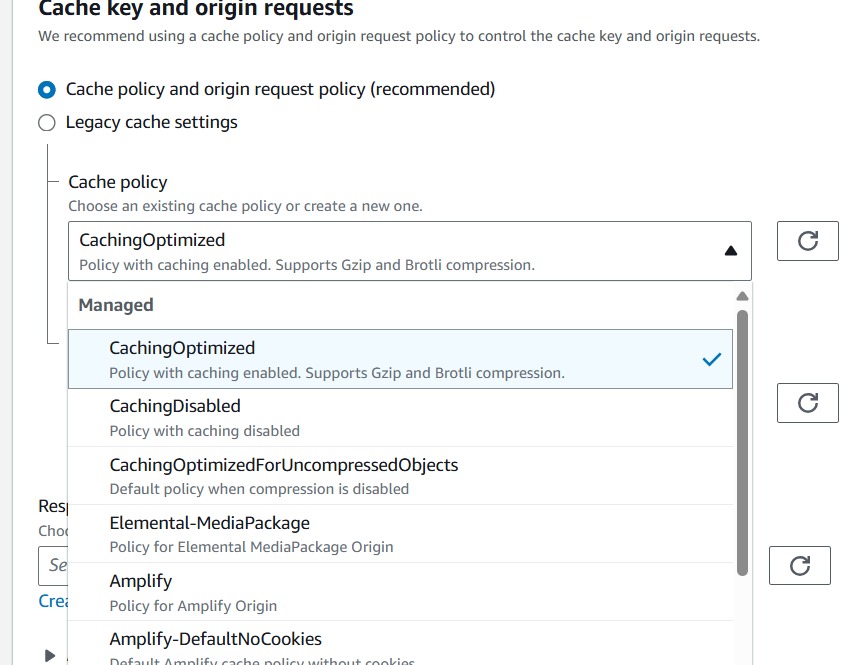
6. Click on **Create Distribution**

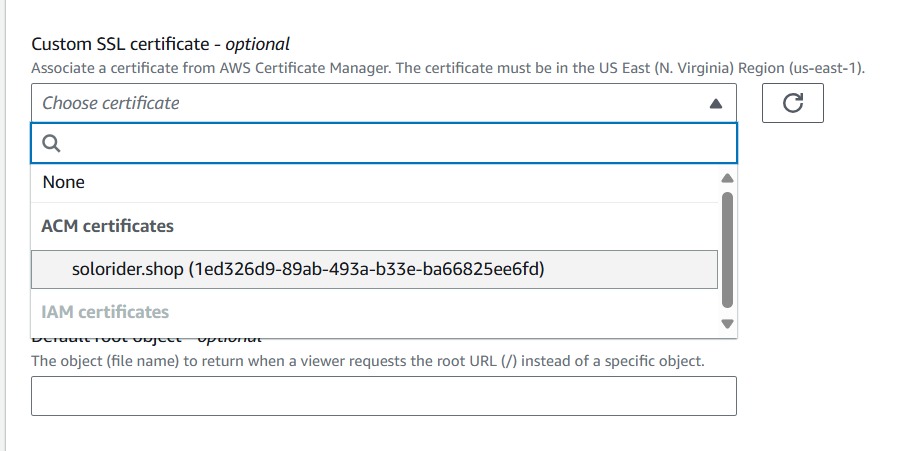
**Note: - (** Wait for few mints **)**

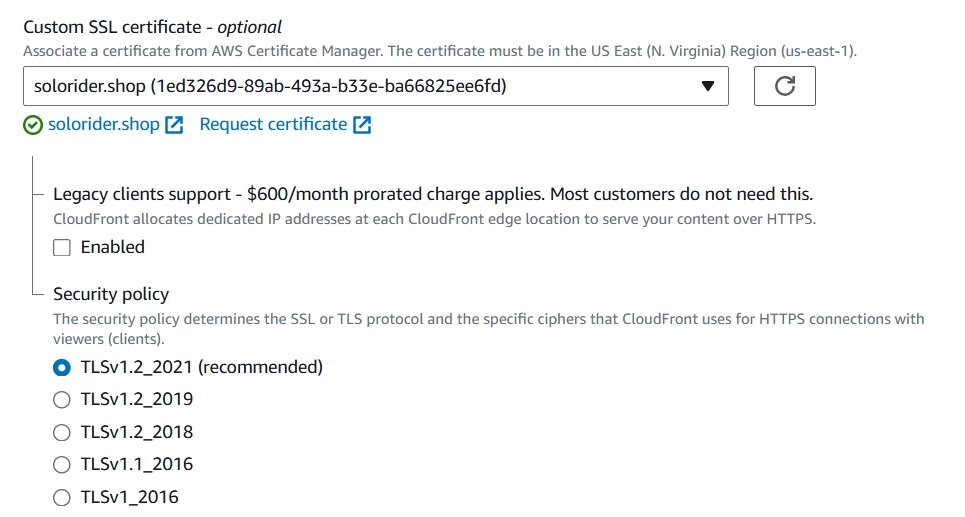
7. Refresh your page after some time its **last modification values change**

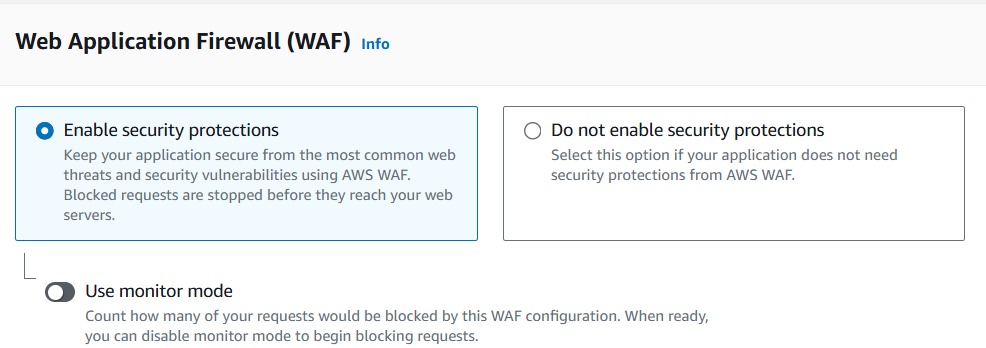
As show in the picture’s

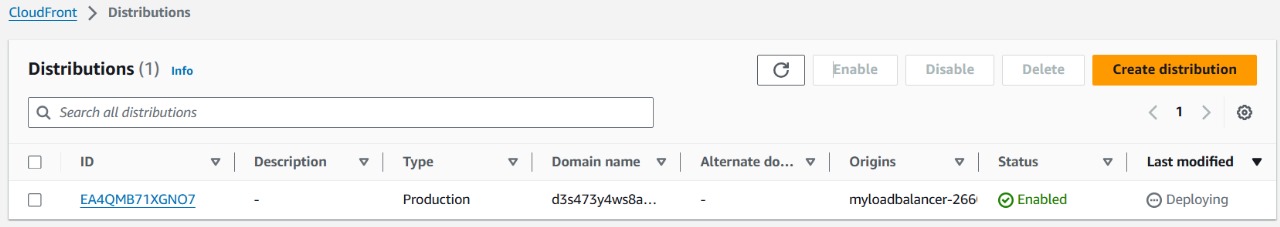


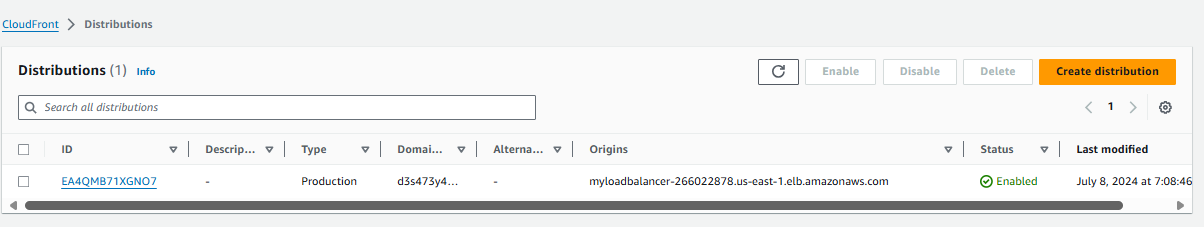












**Add listeners inbound rules**

**https & Target Group & ACM Certificate**

Go to the Load balancer

Go to inside load balancer

Click on listeners

Add listeners

Protocol: **HTTPS** Port: 443

Go to routing actions : Select **Forward to target group**

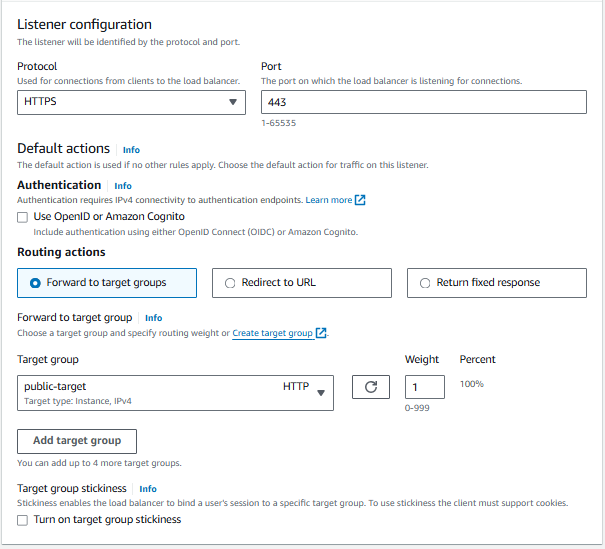
Select target group: **Public-target**

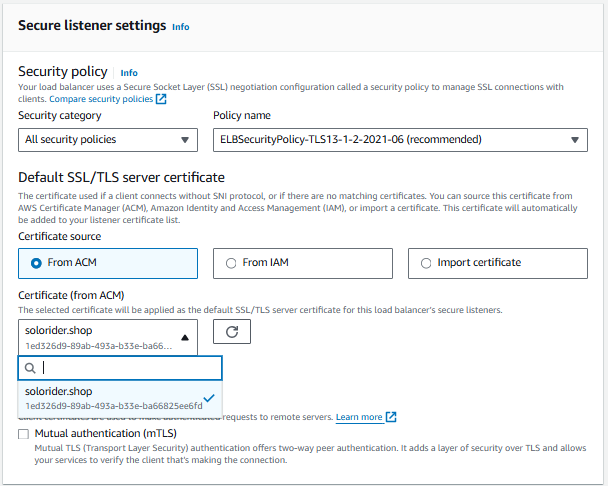
Go to Secure listener Setting

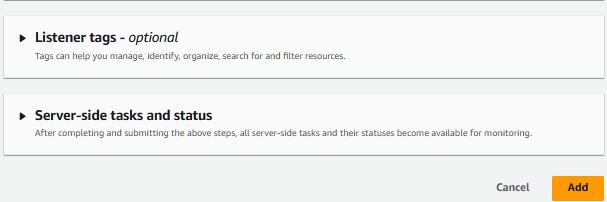
Select: (certification source) **From ACM**

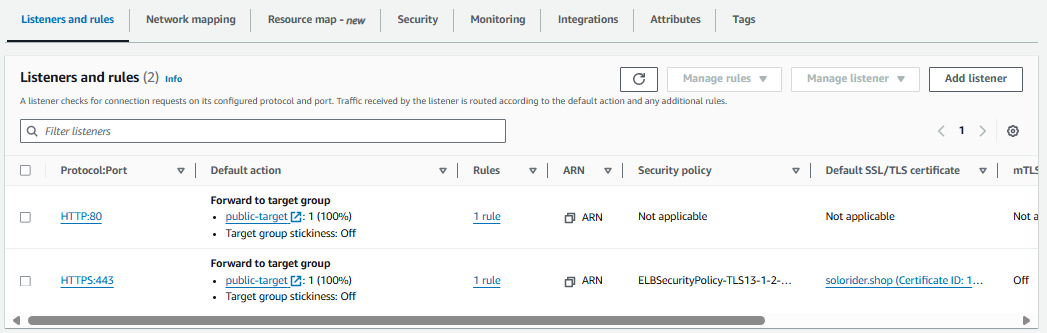
Select certificate (From ACM) : **solorider.shop**

Click on **SAVE CHANGES**



****

****

****

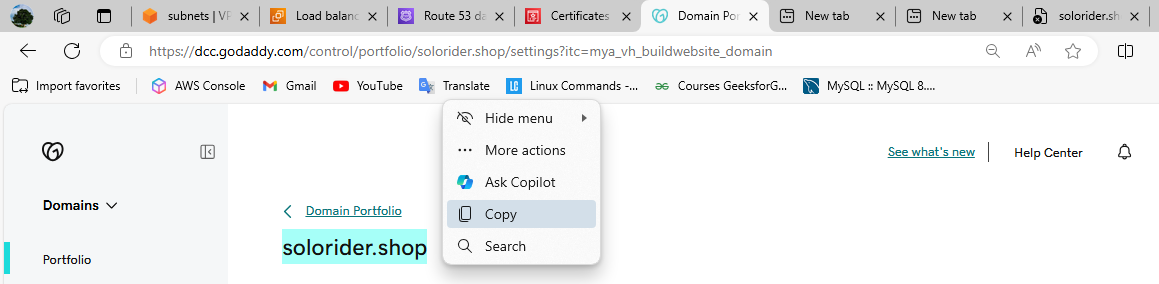
**CHECKING OUR DOMAIN IS SECURE OR NOT**

Go to chrom browser and open godaddy website

Login website

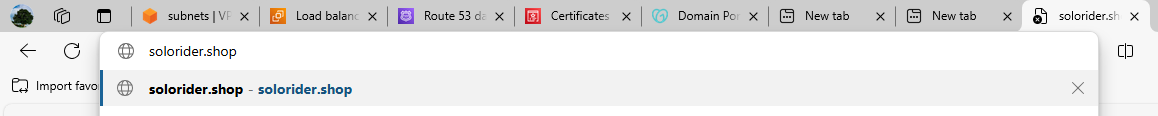
Go to my products

Click on domain then copy the Domain name.

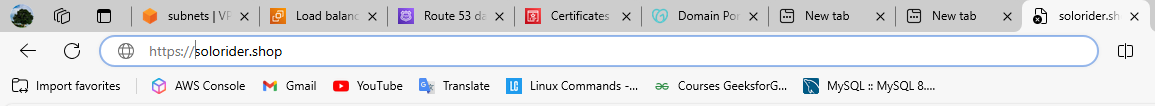


Take a new tab on chrom dashboard

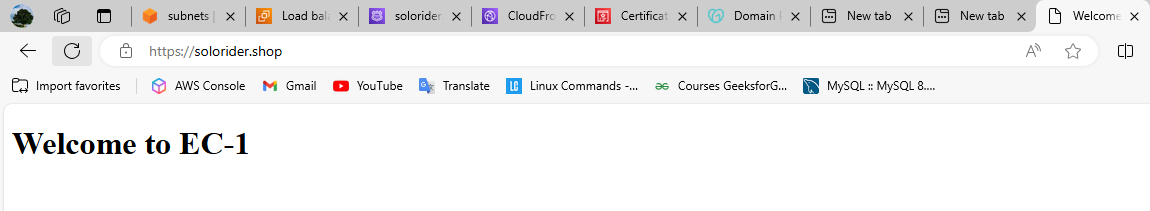
**Paste here copy Domain name**



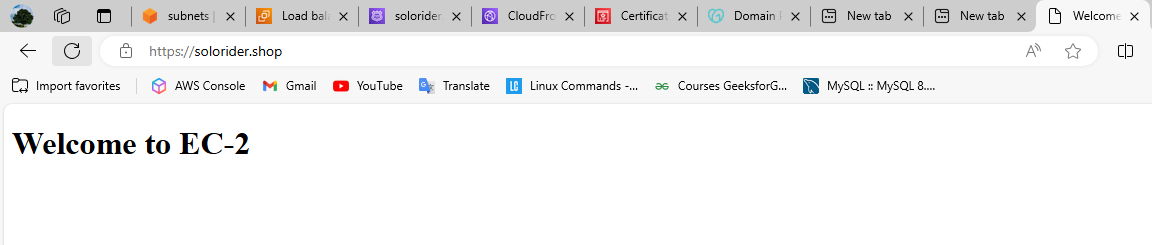
**with adding (https)**



**Then press enter**



**Refresh ones**



KNOW OUR DOMAIN IS SECURE

**CREATE A RDS AND & ATTACHTHE**

Go to Ec2 Dashboard and search RDS

1. Click on create RDS

2. Click on Subnet Groups to create subnets

3. Subnet group Name: **dbsubnetgroup**

4. Description: **db**

5. Select VPC: **2-tair-architecture**

6. Add subnets

Available zone:

**( us-east-1a )**

**( us-east-1b )**

**( us-east-1c )**

**Note: -** Create one subnet and attach to VPC 2-tair-architecture Bcz RDS required 3-subnet & 3-avalable zones

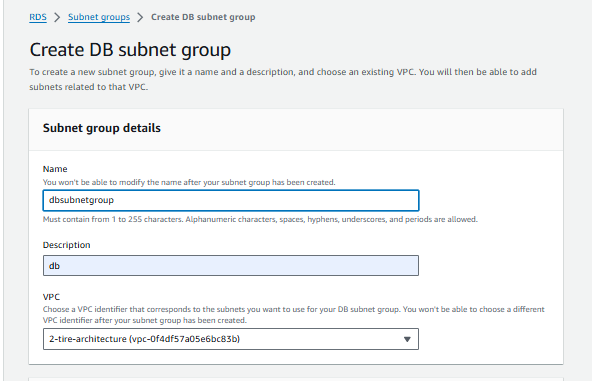
7. Subnets:

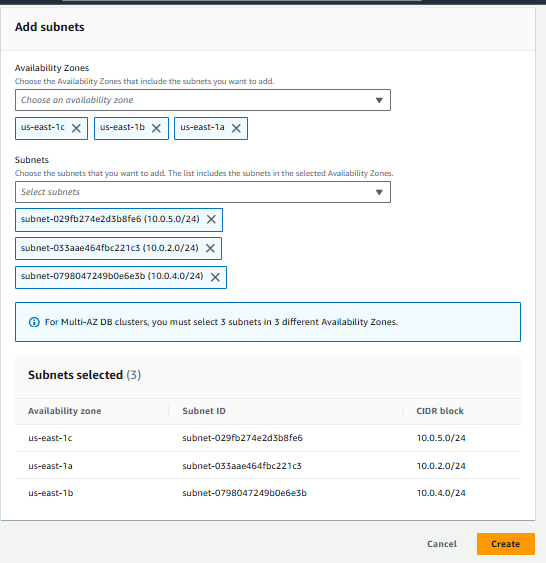
**subnet ( 10.0.2.0/24 )**

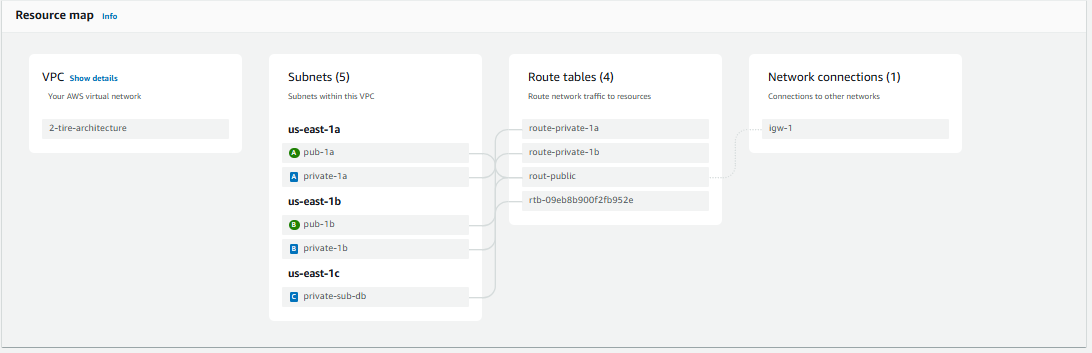
**subnet ( 10.0.4.0/24 )**

**subnet ( 10.0.4.0/24 )**

8. Click on **Create**

****

****

****

**CREATATING RDS**

1. Click on Create Database

2. Choose a Database: **Standard Create**

3. Select Engine: MYSQL

4. Select: **(Enable)**  Show versions that support the Multi-AZ DB cluster

5. Select Templates: **PRODUCTION**

6. Select: **(Enable)**  Mulita-AZ DB Cluster

7. DB cluster identifier: **database-1 (default)**

8. Master username: admin

9. Credential management: Self-managed

Choose Password

Master password: 1234admin

Confirm master Password: 1234admin

10. Instance configuration: Select **Standard classes**

11. Storage:

Storage type: General purpose SSD (gp3)

Allocated storage: 200

12. Connectivity:

Select : (**Don’t connect to an Ec2 Compute resource)**

Select VPC : ( **2-tire-architecture )**

Select db Subnet group:  **(dbsubnetgroup)**

Select : **Public access (** **NO)**

13. VPC Security group ( firewall )

Select : **Choose existing**

Select : Existing VPC security groups: **launch-wizard-125**

14. Database Authentication

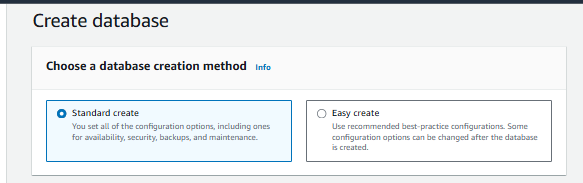
Select : **password authentication**

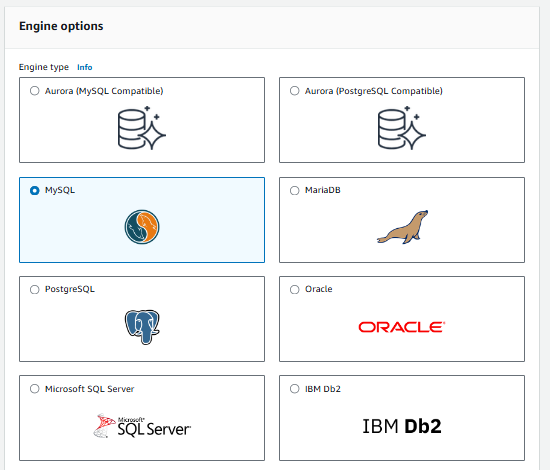
15. Monitoring

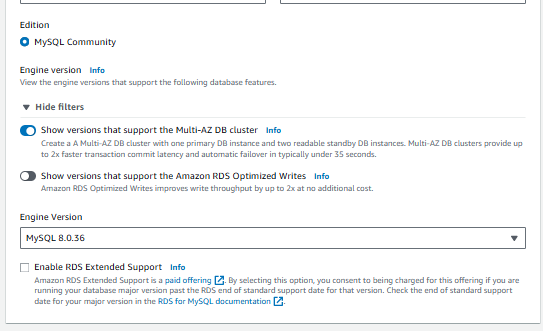
Use default values all

16. Click on Create Database

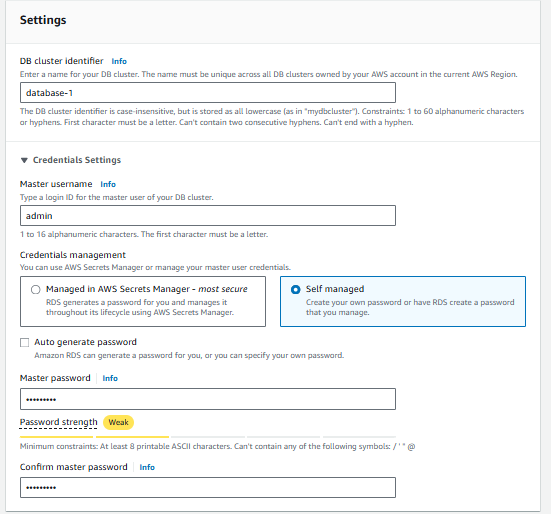
show in the Picture Below all ( GUI version )

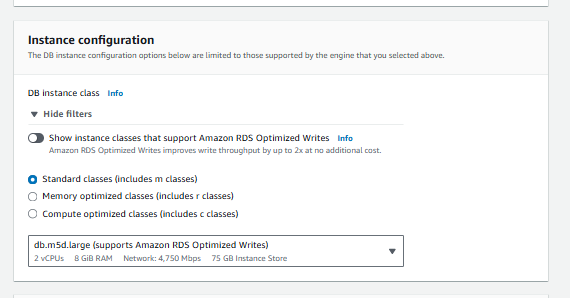


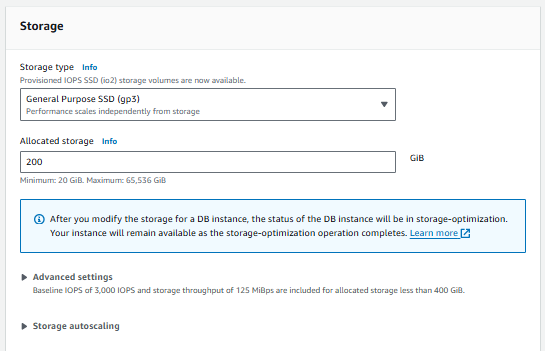


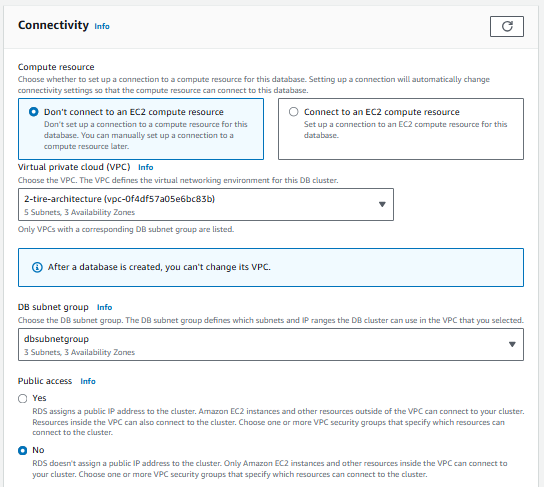


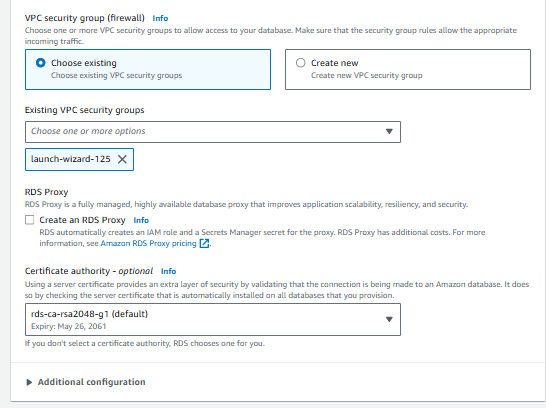


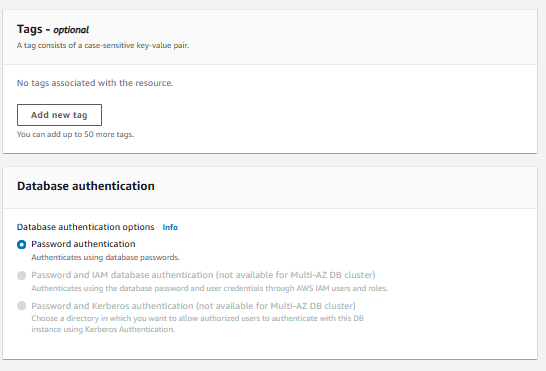


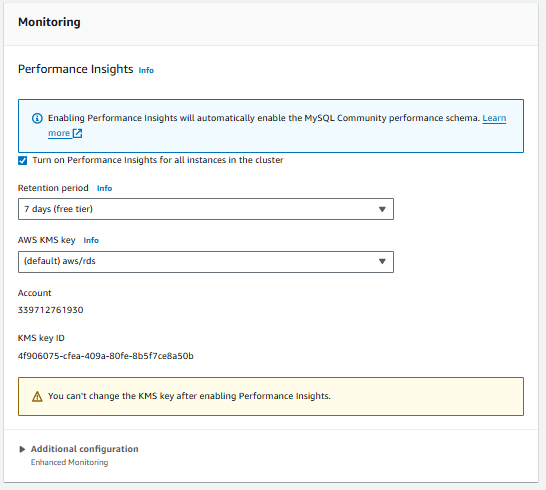


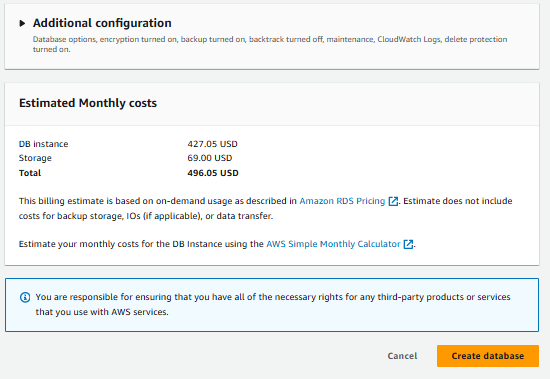




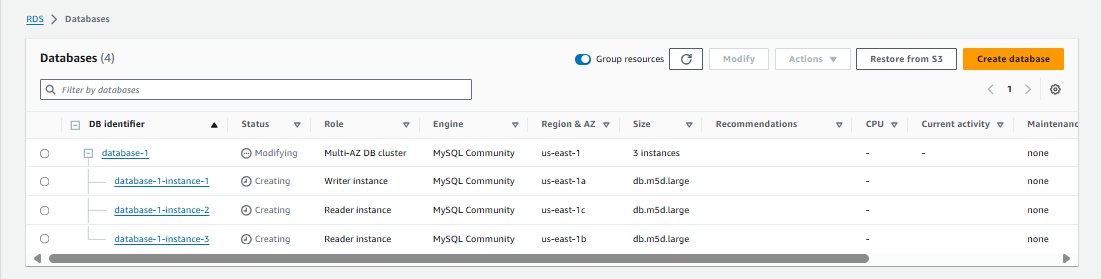








**Wait for some time because it is Taken time to create RDS**



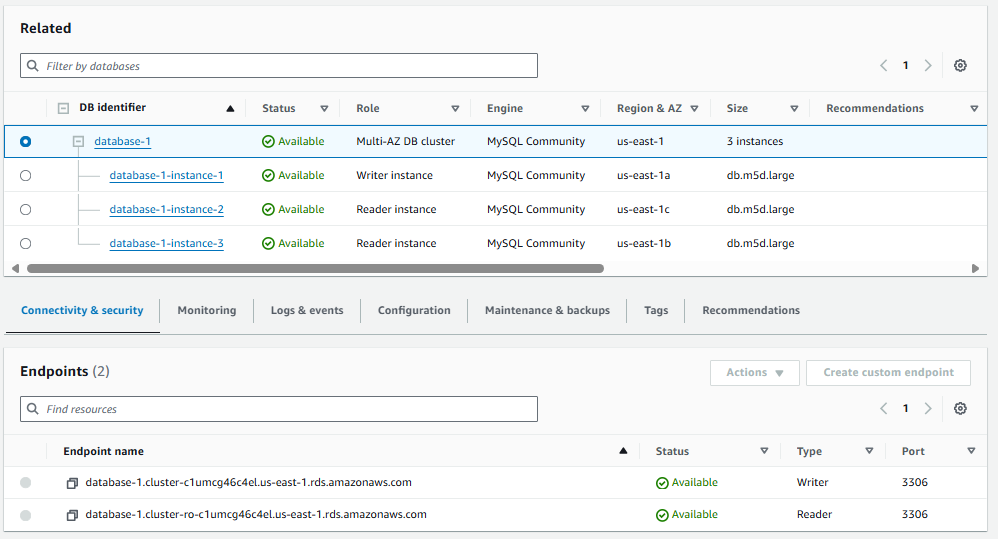
**RDS is completed to creating like below showing**

Database – 1 Multi-AZ DB cluster

Writer – 1 available (1a)

Reader - 1 available (1b)

Reader - 1 available (1c)



**Know I’m connecting to instance Ec-1 And Creating Table**

Select EC-1 Instance And connect root user like show in below

🡪sudor -i

Connect to database use writer Endpoint Like

( database-1.cluster-clumcg46c4el.us-east-1.rds.amazonaws.com )this command

🡪mysql -h database-1.cluster-clumcg46c4el.us-east-1.rds.amazonaws.com -u admin -p

🡪Enter your password: 1234admin

🡪show databases;

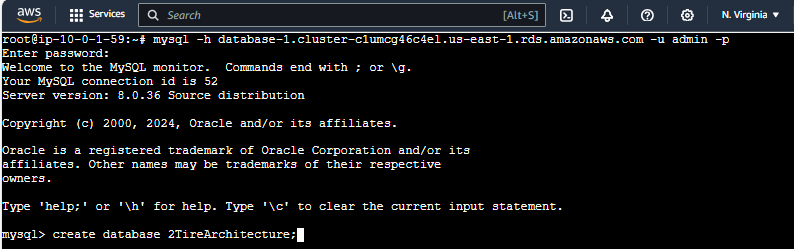
🡪create database Balraj;

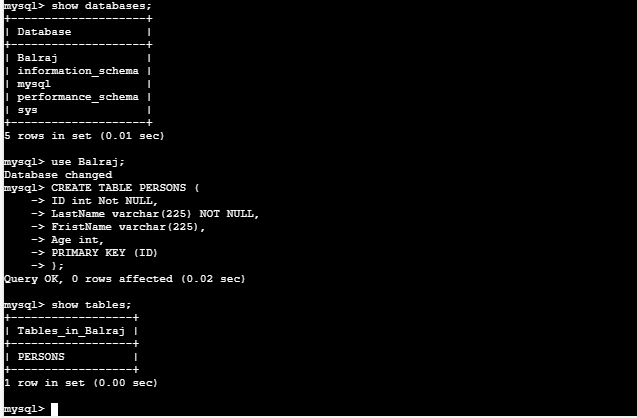
🡪use Balraj;

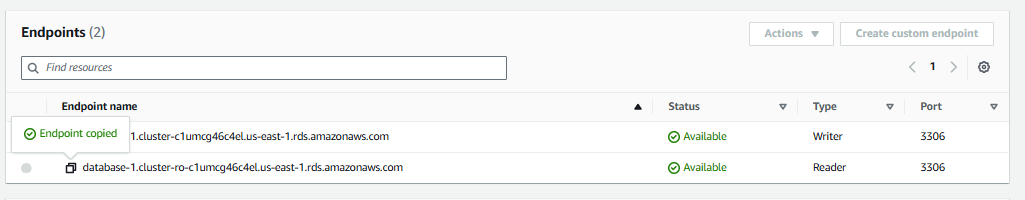
Database changed to store in this space any table format data

🡪CREATE TABLE PERSONS ( ID int NOT NULL, LastName varchar(225) NOT NULL, FirstName varchar(225), Age int, PRIMARY KEY (ID));

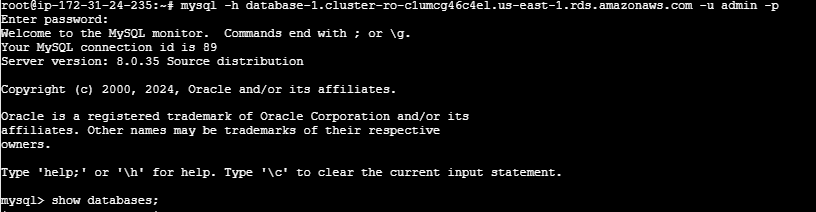
🡪show tables;

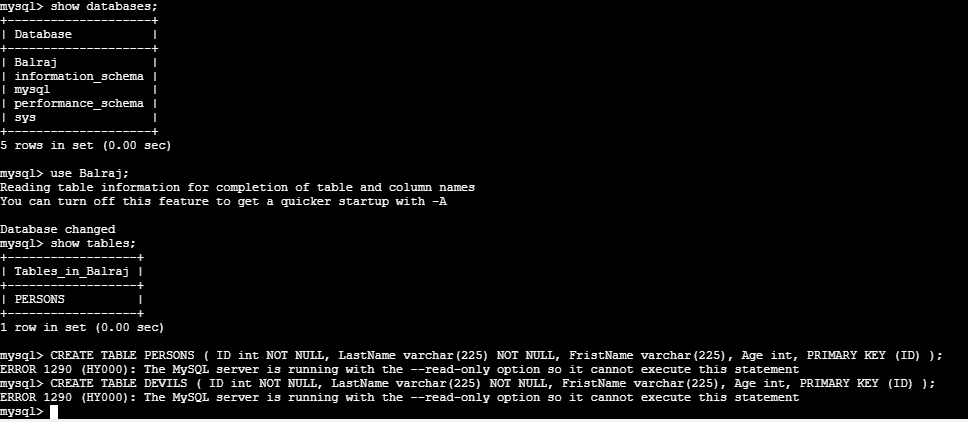






**Connecting to Reader endpoint**





ERROR 1290 (HY000): The MySQL server is running with the --read-only option so it cannot execute this statement

Above error message showing Reader endpoint to inserting ant data.