https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

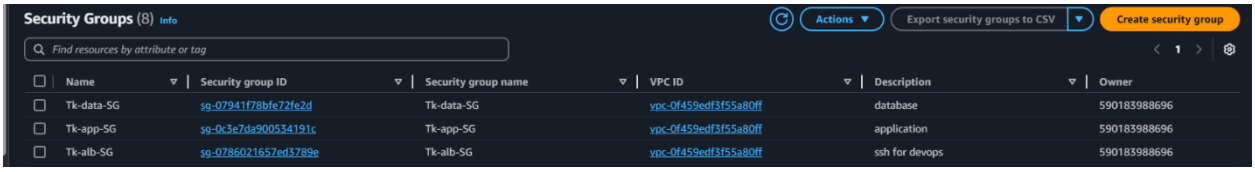
Step-by-Step Guide to Setting Up a Fully Serverless 3-Tier Architecture for Hosting a Dynamic Docker Web Application for a Business

Key:

- ECS: Fargate
- Lambda
- RDS
- IAM
- Secret manager
- CloudFormation
- NAT, VPC, Elastic IP, ALB
- Docker image

Step 1:

- Reproduce the infrastructure below from VPC to Security group



You will need three SG and NACLs.

- **Alb SG public**: Http and Https from anywhere
- **App-SG**: http and Https from “alb-SG”

Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

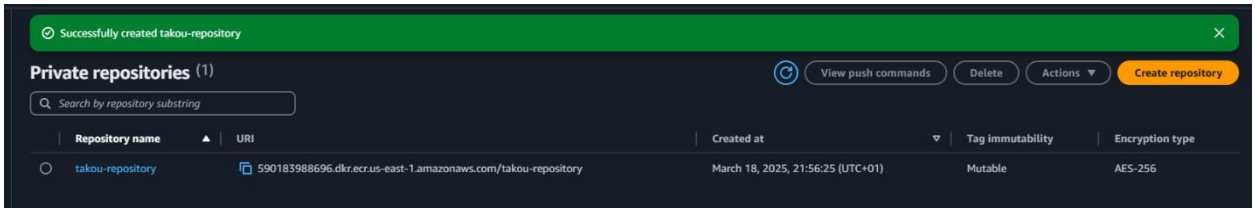
- **Data-SG**: MySQL and MySQL from app-SG & from data-SG itself (to be used by a lambda function in the database subnet.)

You will also need a NAT as seen above

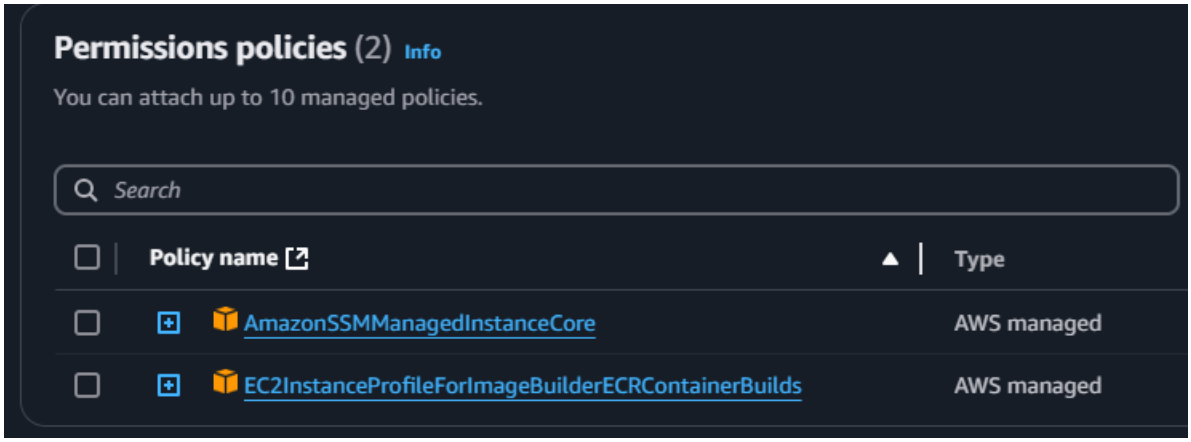
- NB: An Elastic IP will be allocated for the NAT above. Remember to release the Elastic IP at the end of this project.

Step 2:

- **Set up ECS repository:** Access ECR on your management console and create a private repository with AES-256 encryption type.



- **Set up docker image**: Create an IAM role allowing the following permission. (A docker image can be gotten from your developers or anywhere. A docker is a container holding something (An application). For this case our docker will be a dynamic website image.)



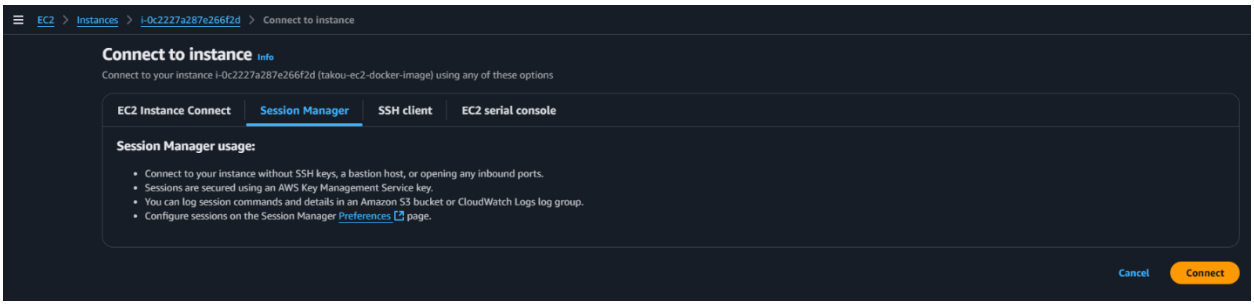
- Launch a Linux EC2 to setup our docker image: Attached the created role above
  - **Use Linux 2023 or Linux 2**
  - **No keypair**
  - **Put it in the "appSubnet"**

Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

- **No public IP**
- **Assign to it the default SG for your VPC**
- **NB: Make sure to attached the IAM role**

- Connecting to our EC2 we will use "session manager" so SSH is not allowed.

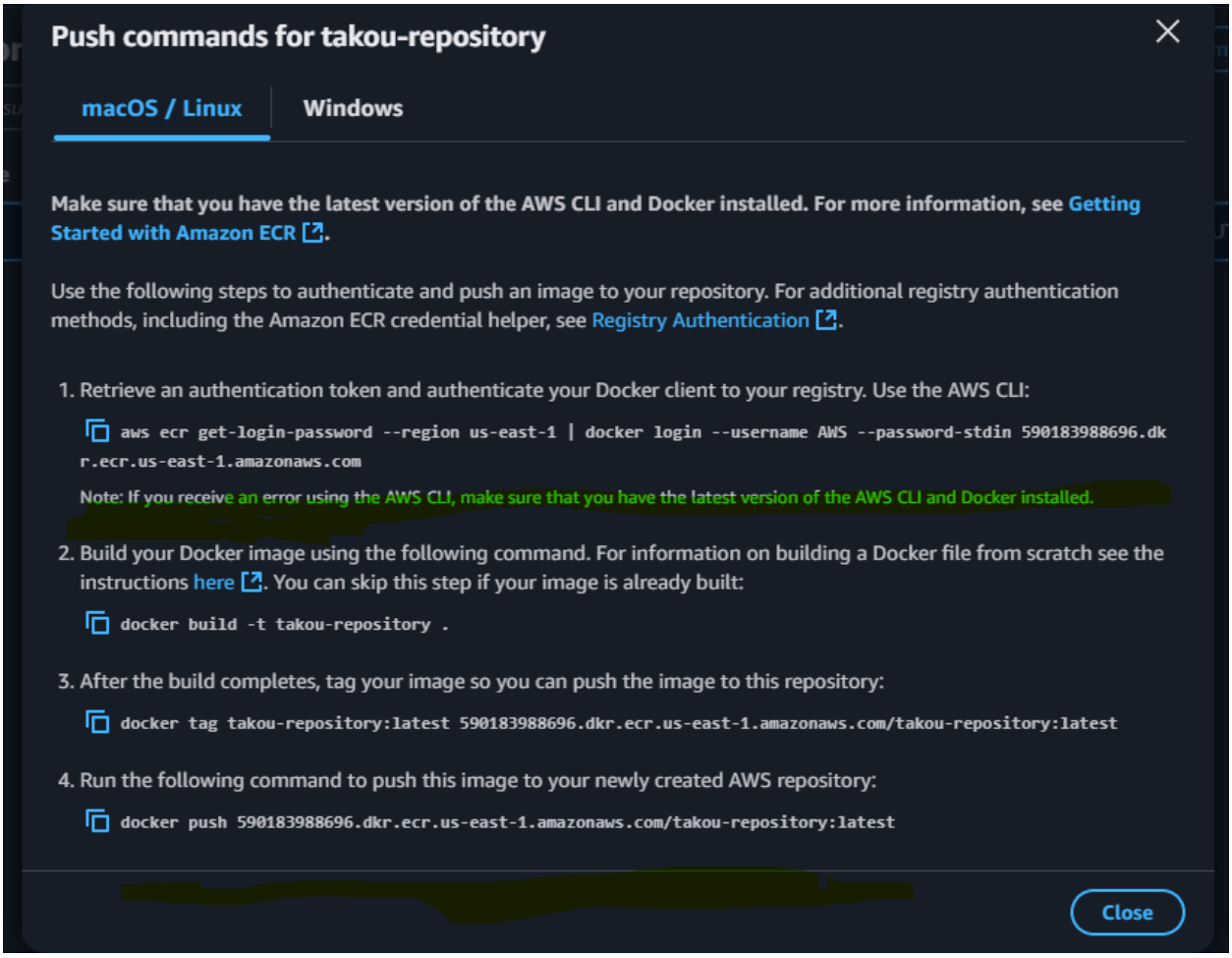


- issue the following command to install docker
  - sudo su – ec2-user
  - Make sure you are working in the: /home/ec2-user by using the "pwd" command.
  - Update instance: "sudo yum update"
  - Install docker: "sudo yum install docker -y"
  - Start docker: "sudo service docker start"
  - Add current user to docker group: "sudo usermod -a -G docker ec2-user"
  - Exit and connect back to EC2
  - Change user to: "sudo su – ec2-user" the default linux user is "ec2-user". You can create a new one and use.
  - Verify your directory: (/home/ec2-user) "pwd"
  - Download docker image from my repository: "wget https://github.com/synthetico/miniprojects/raw/refs/heads/main/amazon-ecs-mini-project/ritual-roast-code.zip "
  - The link above is my repository holding the docker image (dynamic website). You can get any dynamic website and use.
  - Verify it downloaded using: ls
  - Unzip file: unzip filename
  - cd into the unzipped directory
  - ls to view files
  - Build docker image to your ECR: "docker build -t ecr-url ."
  - ***NB: the is a dot at the end of the command above***
  - It will also install MySQL, copy source code t html, export image and layers.
  - When it's done very image: "docker images"

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

- Push docker into ECR
  - You can get these commands on your ECR dashboard

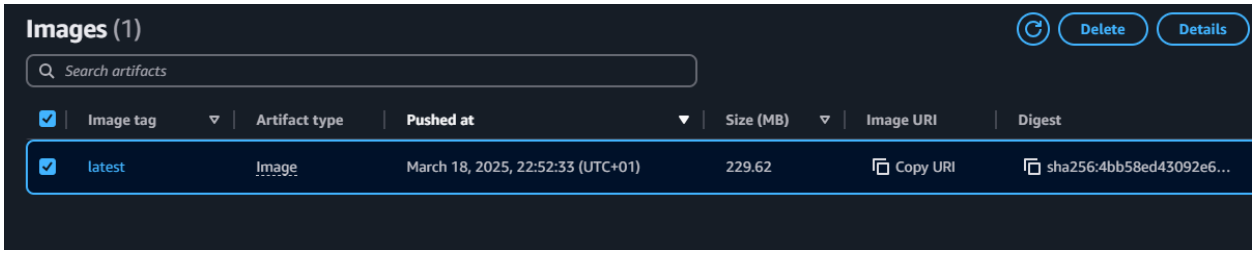
  - From the image below, run the commands in green (**1 and 4)**



Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

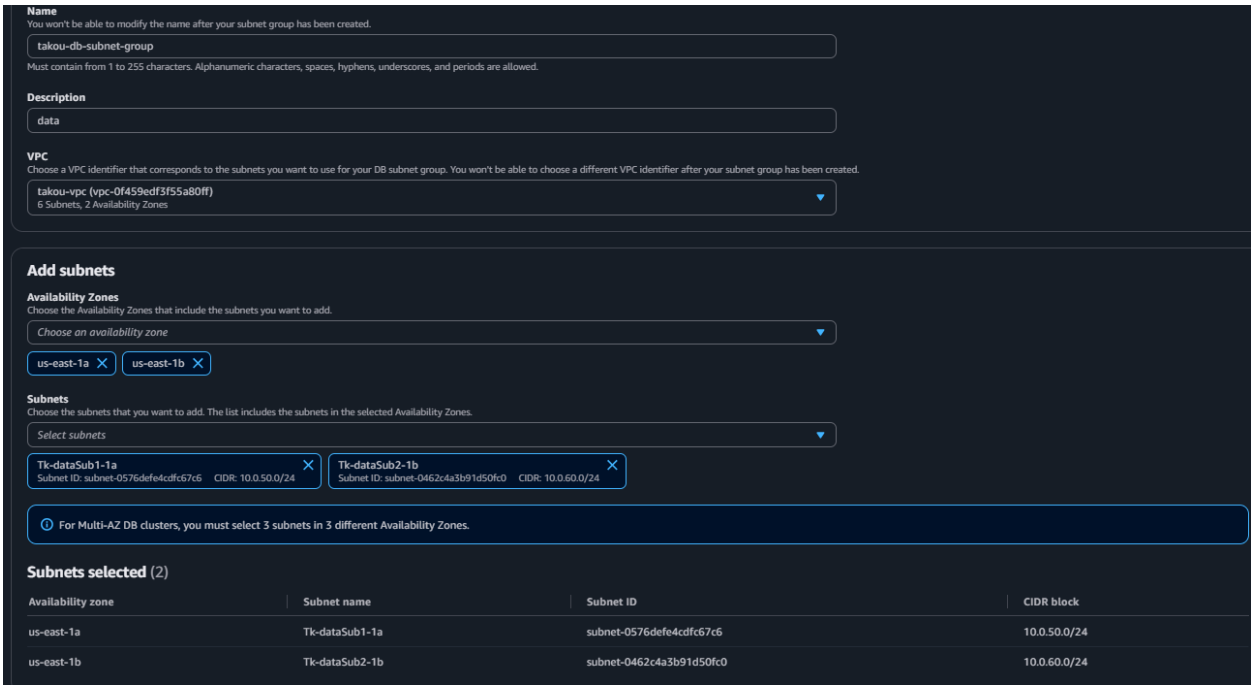
- The second command above was done already and the third is just tagging.
- If no IAM role was attached, the two commands above won't work
- Access your ECR and verify docker image in it.



NB: the EC2 can be deleted as it's no more needed.

Step 3:

- Create a MySQL database: In production use **multi-AZ** but for this lab, we will use free tier
  - Create db subnet group

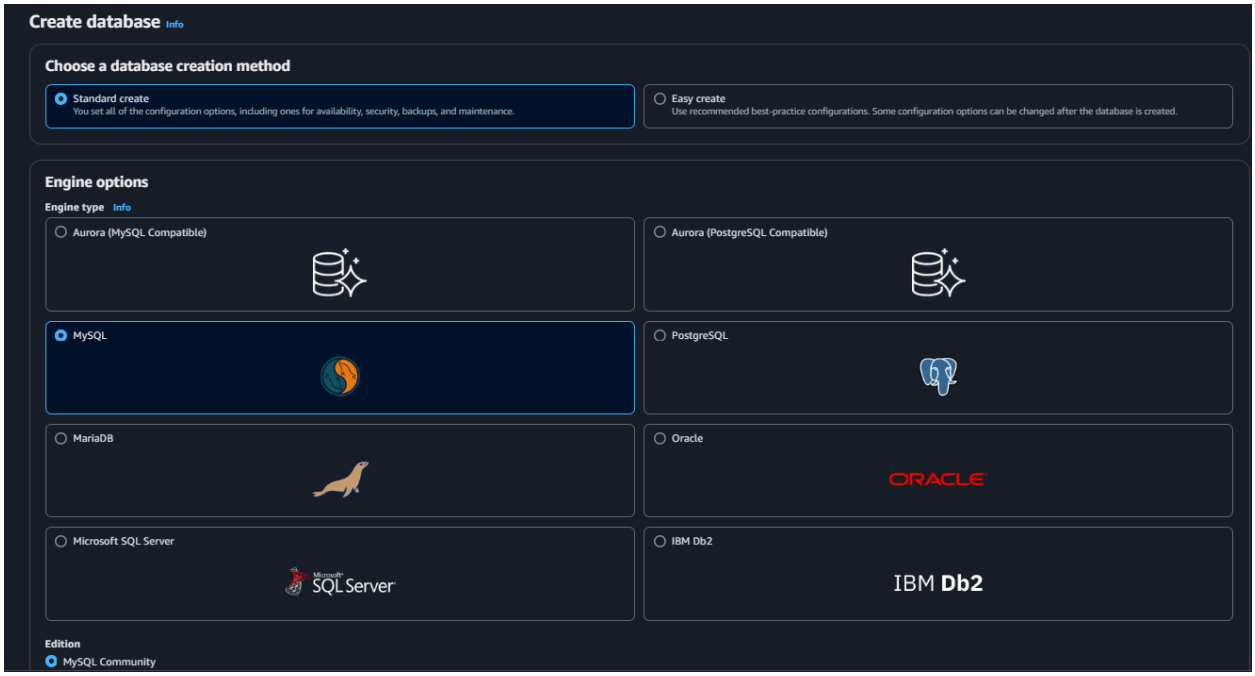


  - Create database using MySQL engine-free tier-data-SG.

Documented by **N.T Samuel**

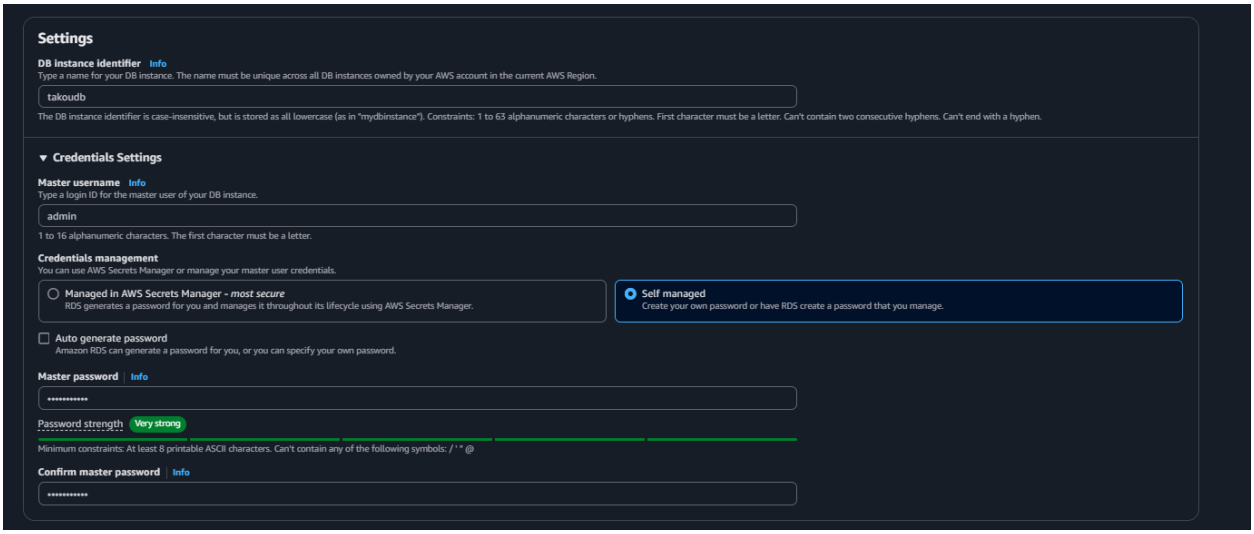
https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

- Name your database and give it a password

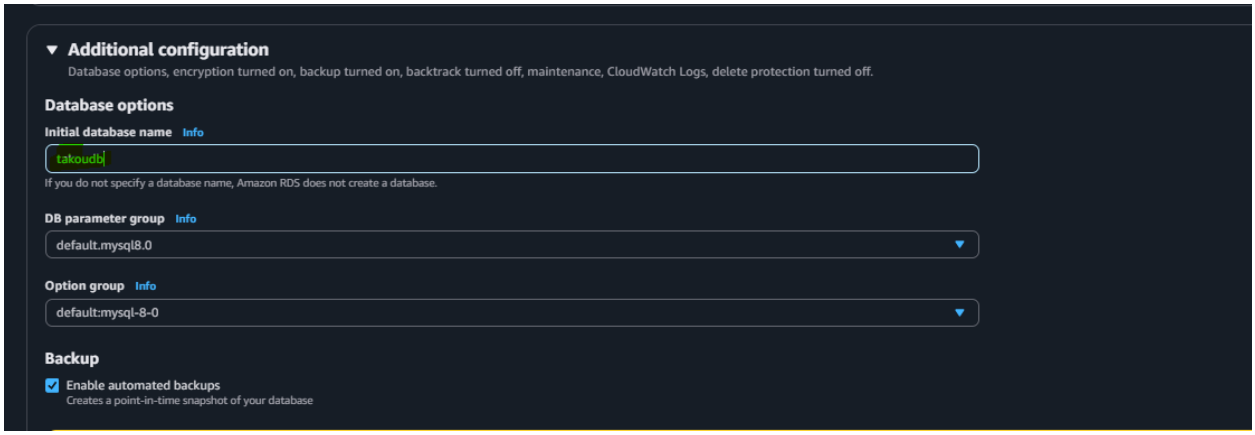


Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV



- make sure to select your VPC, subnet group, database SG and set database name under "**Additional Configuration**" and create



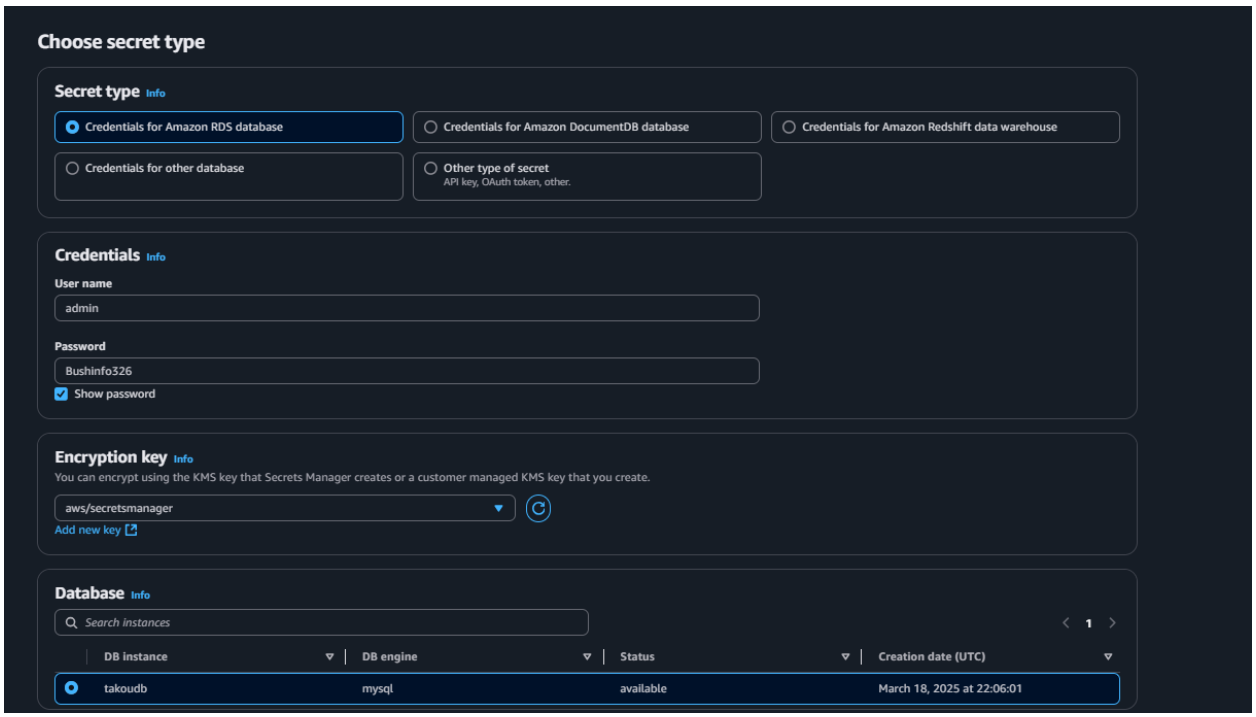
Step 4:

- We will set up secret manager to dynamically retrieve database secret information. This will help in secret authentication.

- Configure AWS secret manager
  - Access **secret manager** and store a new secret

Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

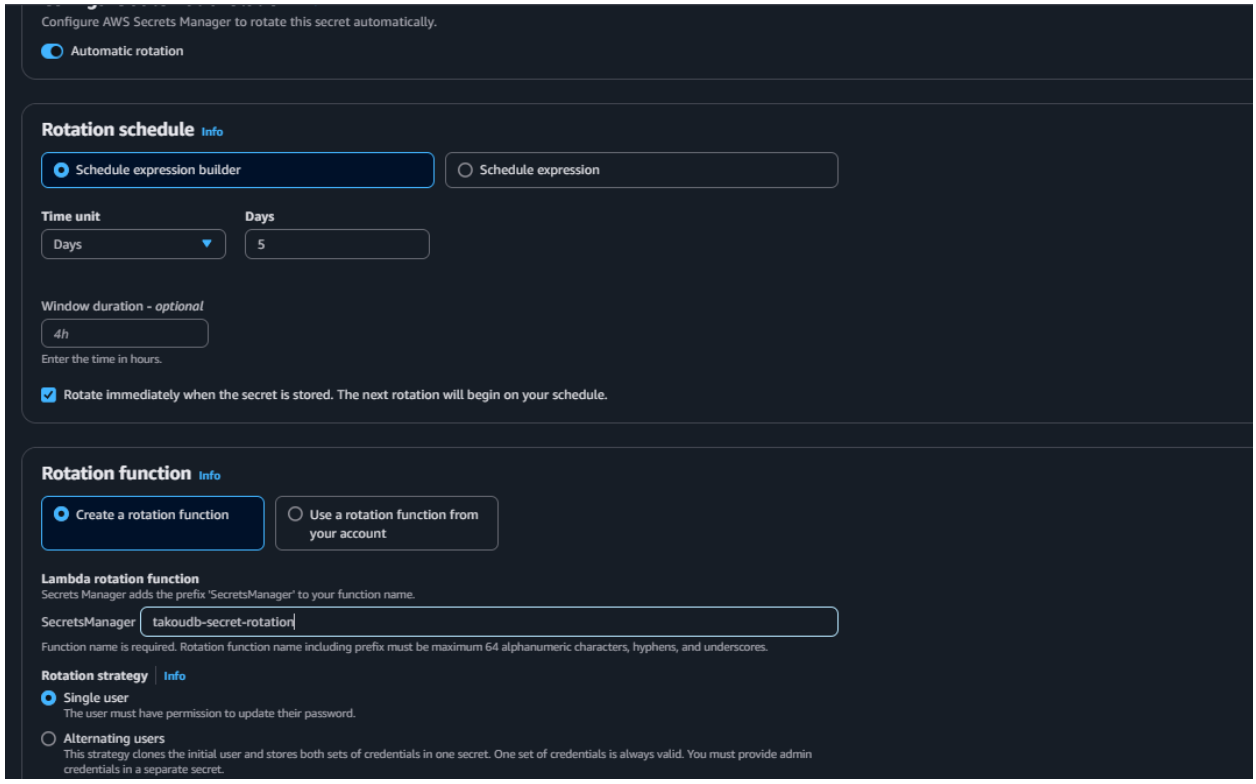


- name secret
- Set up rotation period

Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

- Since the rotation will be managed by a lambda function, create a rotation function as seen below and store secret

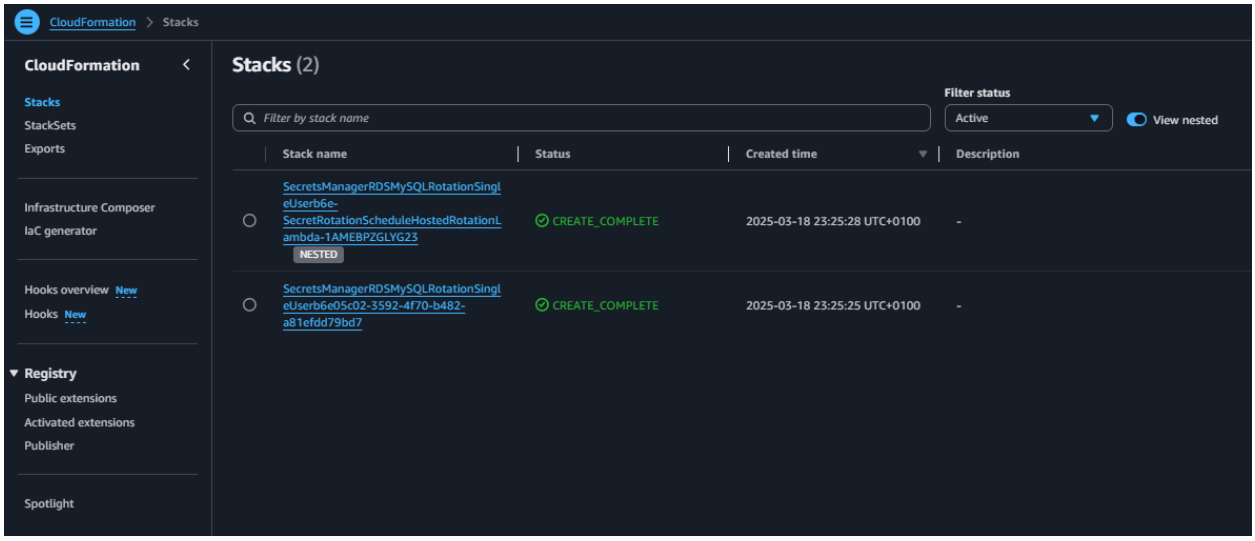


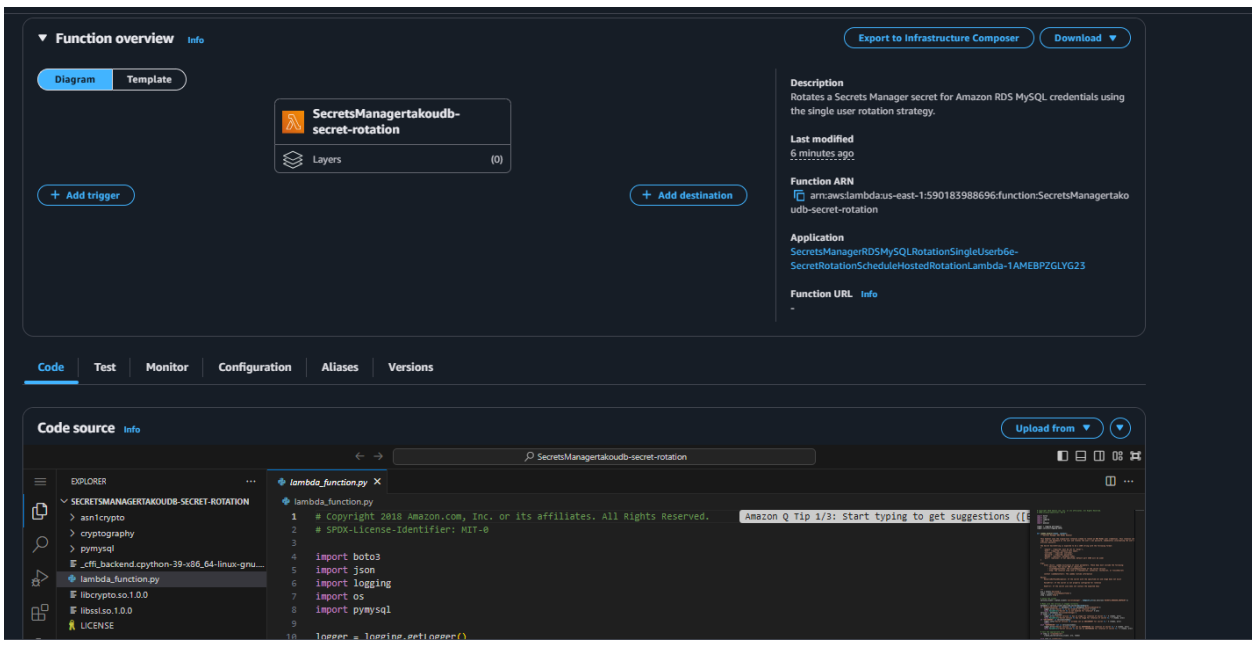
- ***NB: Based on the rotation period configured, lambda will rotate your secrets respecting that period for you and your team. So a lambda function will be created.***

Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

- At this stage **CloudFormation** will come up and set up the lambda, setup the secret manager and establish communication via the data SG between lambda and database.





Step 5:

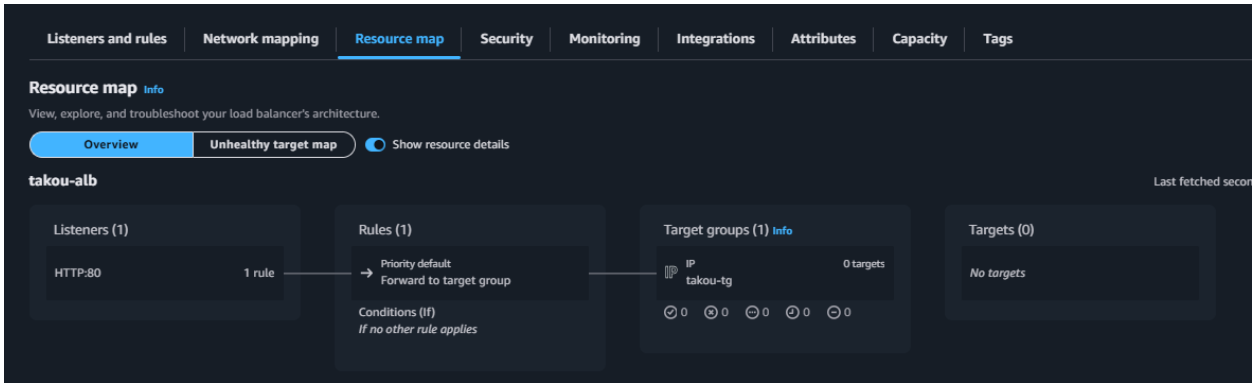
- Create an ALB
  - Create a target group
  - **IP add** as target because we are using EC2 Farget
  - Name target group and select Http as protocol
  - Select VPC

Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

- Assign health check file: “health.html”
- Click next
- Make sure your VPC is selected, click remove on the subnet Ip address below
- Review and create

- **Create ALB**
- Name it
- Select “**internet facing**”
- Select VPC and public subnets
- Select **alb-SG**
- Set listener to **https**
- Select your SSl certificate
- Add a second listener with **http** as protocol **redirecting** it to https port 443.
- Create alb



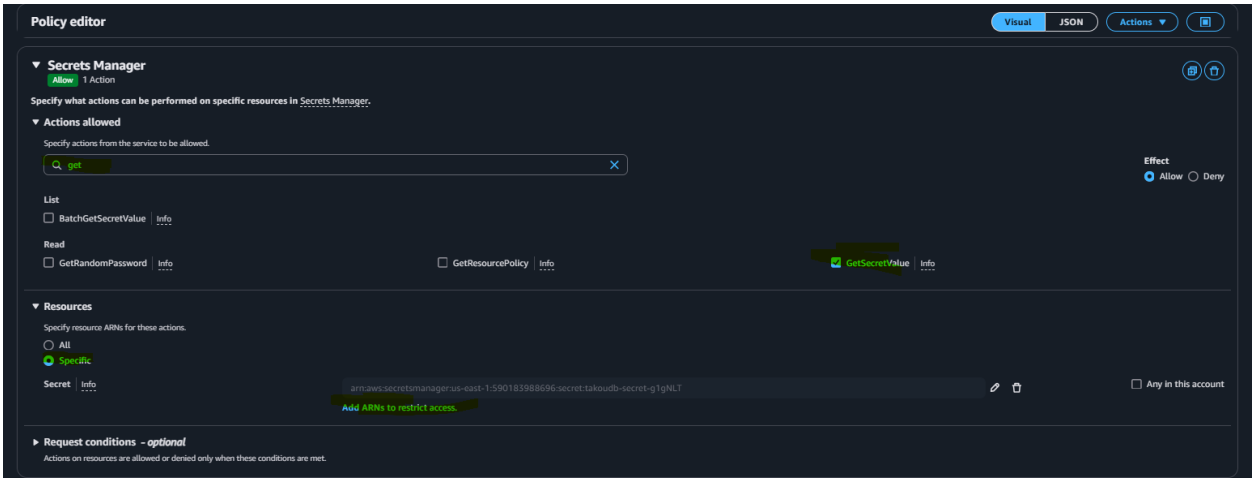
- Above is the ALB movement.

- Create an IAM role that has multiple policies to allow **secret manager** reference, **CloudWatch logging** and **docker image copying**.
  - Access **IAM dashboard**
  - Create a policy for secret manager
  - Click on policies
  - For Action, search for get and select “**Getsecretvalue**”
  - Under resources be specific and attach the **Secret Manager’s ARN** you created on secret manager
  - Access your secret manager and copy the ARN

Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

- Click on “Add ARN” and paste the copied ARN
- Create.




```
{
        "Version": "2012-10-17",
        "Statement": [
                {
                        "Sid": "VisualEditor0",
                        "Effect": "Allow",
                        "Action": "secretsmanager:GetSecretValue",
                        "Resource": "arn:aws:secretsmanager:us-east-
1:590183988696:secret:takoudb-secret-g1gNLT"
                }
        ]
}
```

NB: The Json is for the policy created above.

- Create a role now
- Role will be used by “elastic container service” ECS
- Still on IAM dashboard, click on **create role.**
- Trusted entity is “AWS service”
- Use case “Elastic container service” and choose “Elastic container service task”

Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

- Search the following permissions: ***AmazonECSTaskExecutionRole*** and the **custom policy created above**

Step 6:

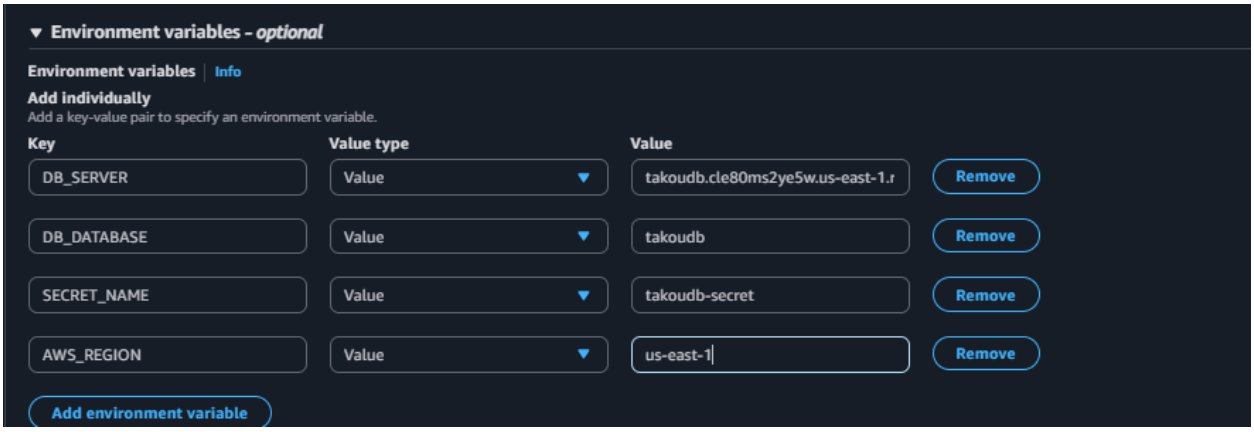
- Create ECS cluster and task definition, then services
- Access your ECS dashboard.
  - **Create a cluster**
  - Name it
  - Choose Fargate
  - Create

  - **Now create a task definition**
  - Name it
  - Choose Fargate as launch type
  - Select created ECS role for both **task role** and **execution role**
  - Name container and input ECR URL as image URL

Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

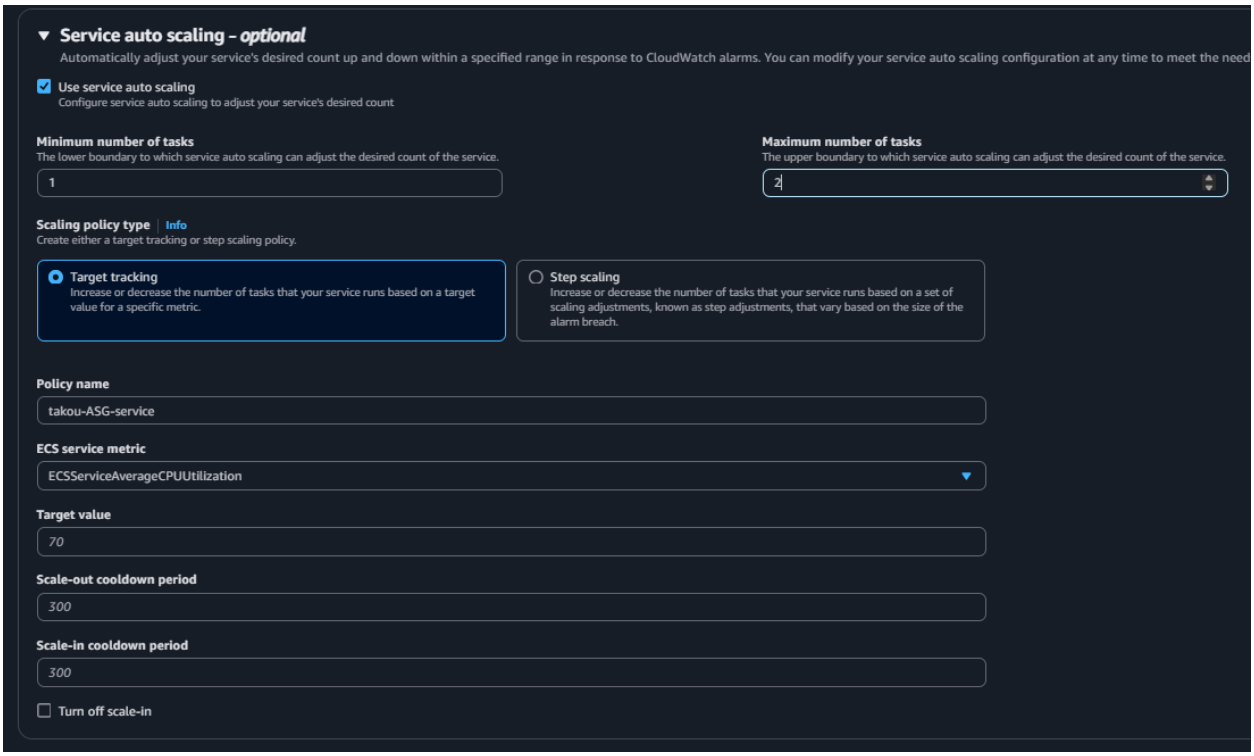
- Set environment variables as seen below customizing the values



- **Create a service**
- ***NB: From cluster creation, CloudFormation start working again at the background. Any failure, verify CloudFormation***
  - Under your cluster
  - Create a service
  - Launch type: Fargate
  - Application type: Service (Combination of various tasks)
  - Select task definition
  - Name your service
  - Screw down to "**networking**"
  - Select your **VPC** and **app subnets**
  - Choose **app subnet SG**
  - Access "**Load balancing**"
  - Select the load balancer and its component
  - ***NB: In the case were your ALB had SSL certificate, select the HTTPS listener.***

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV

- Enable auto scaling as seen below

- 

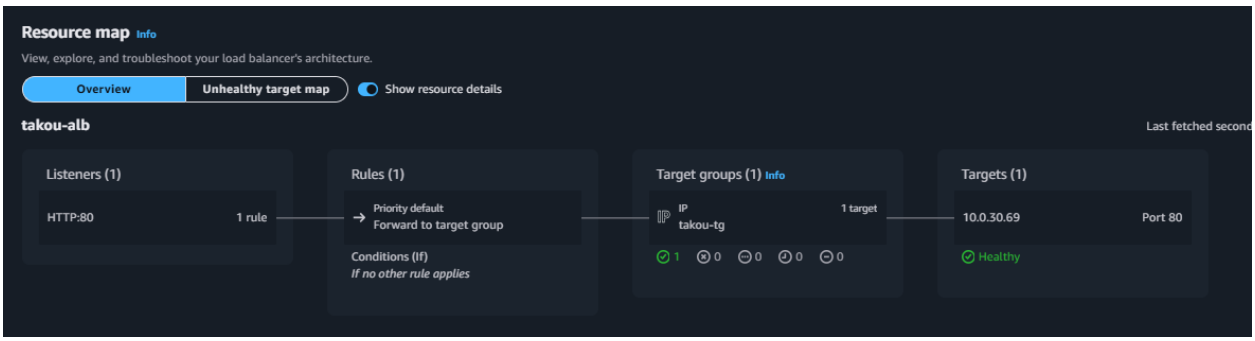
Create **and wait for CloudFormation to deploy your service.**
• **Access Route 53** and create an alias record pointing to your ALB (Application Load Balancer).
• **View the result** in a browser using your custom DNS.
• **Test your RDS** by submitting data through your dynamic website.
• **Troubleshooting:**

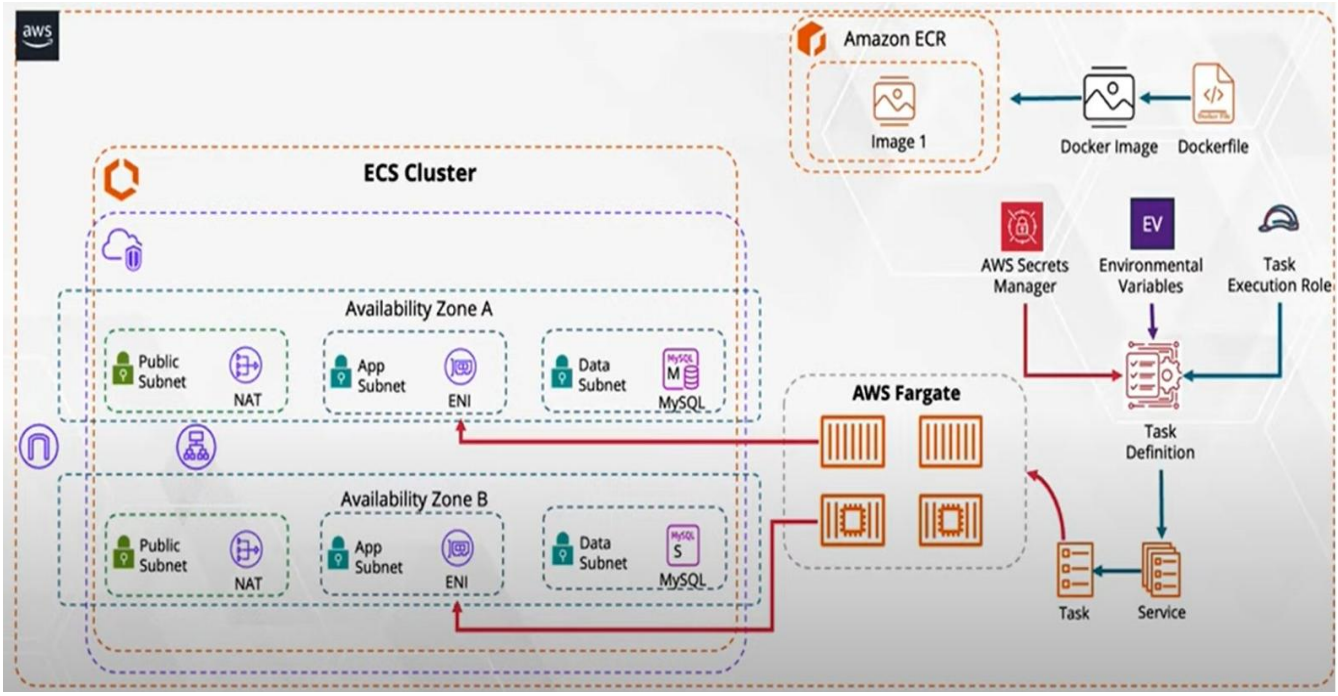
- If you encounter errors, test your site using the ALB DNS.
- Verify ALB listeners and their attachments.
- Ensure the correct environment database variables are used.
- Check CloudFormation for any errors.

• **Note:** If this setup is not for production, delete CloudFormation and all related resources to avoid unnecessary costs.

Documented by **N.T Samuel**

https://youtu.be/C6v1GVHfOow?si=Kt_YvuKsCRvpkFmV





Documented by **N.T Samuel**