FULLSTACK APPLICATION DEPLOYMENT ON ECS-FARGATE

THROUGH TERRAFORM

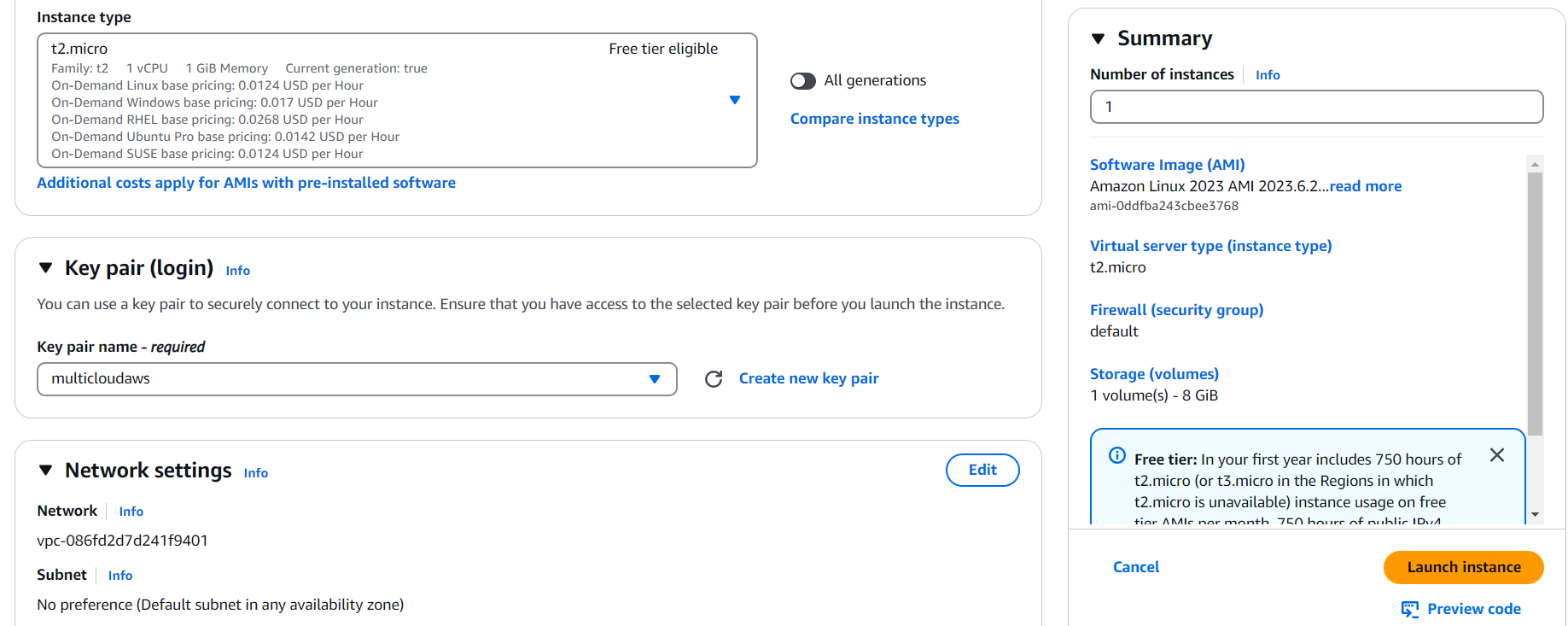
**1. Overview of Three-Tier Architecture** : Deploying an ECS task on private subnets ensures that your containers remain isolated from direct internet access while still being able to communicate internally within your VPC. Here’s a step-by-step guide to deploying an ECS task in private subnets.

A three-tier architecture consists of:

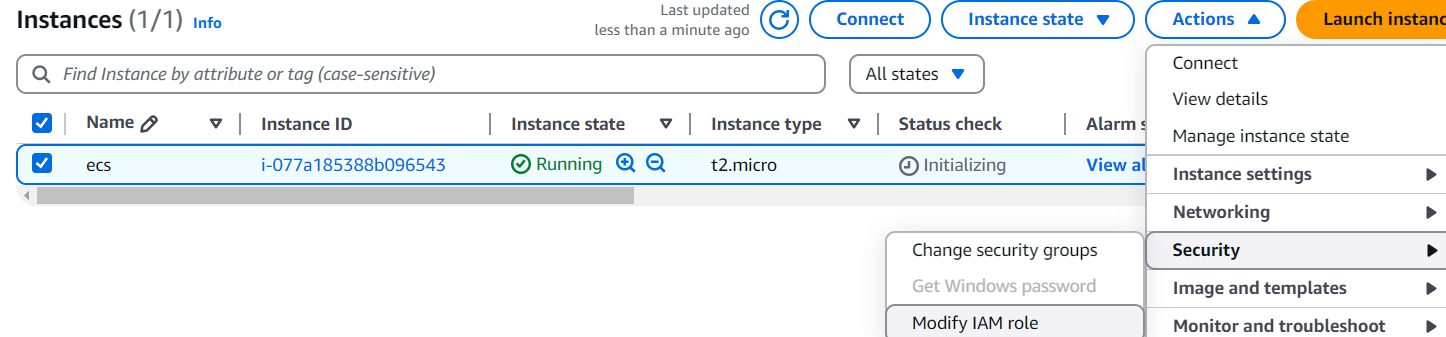
1. **Presentation Layer (Frontend)**
   * This is the user interface (UI) of the application, such as a web or mobile frontend.
   * Deployed on **ECS (Fargate or EC2) running containers** with Nginx, Apache, or a React/Angular application.
2. **Application Layer (Backend)**
   * This layer handles business logic and processing.
   * Deployed on **ECS (Fargate or EC2) running containers** with Node.js, Java, Python, or another backend service.
3. **Data Layer (Database)**
   * A managed relational database, such as **Amazon RDS (MySQL, PostgreSQL, or Aurora).**
   * Stores and manages application data.

**Key AWS Services Used**

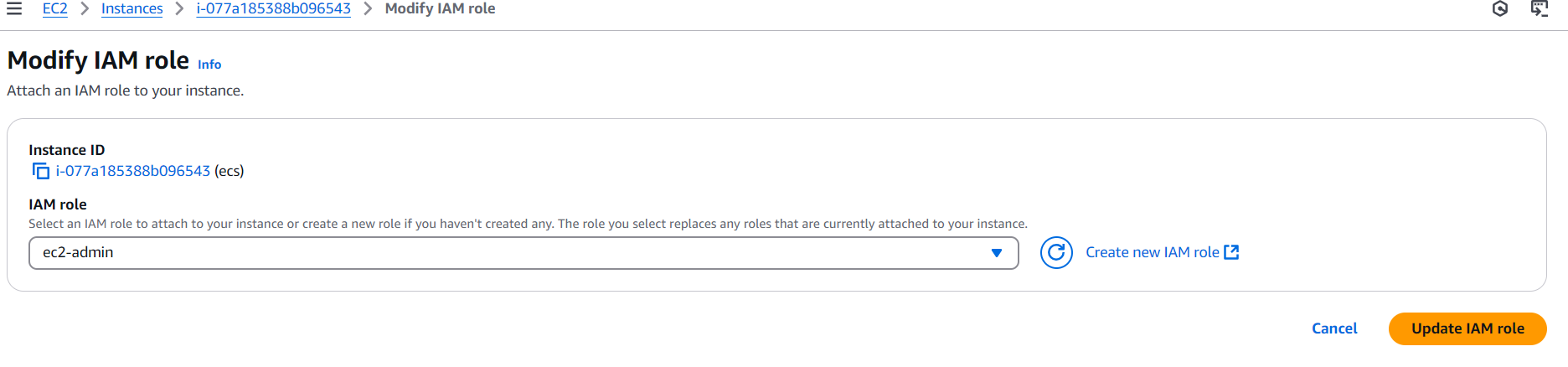
* **Amazon ECS (Fargate or EC2):** Manages containerized workloads.
* **Elastic Load Balancer (ALB):** Routes traffic to frontend/backend services.
* **AWS RDS:** Managed relational database service.
* **Amazon VPC:** Segregates frontend, backend, and database tiers into subnets.
* **IAM & Security Groups:** Controls access between different layers.
* Create a ec2



* Attach Iam role to ec2



* Update the role



* Connect to the ec2



* Install the git and docker by following commands

sudo yum install git -y

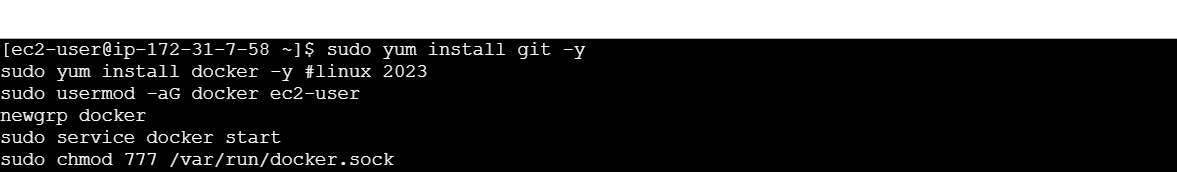
sudo yum install docker -y #linux 2023

sudo usermod -aG docker ec2-user

newgrp docker

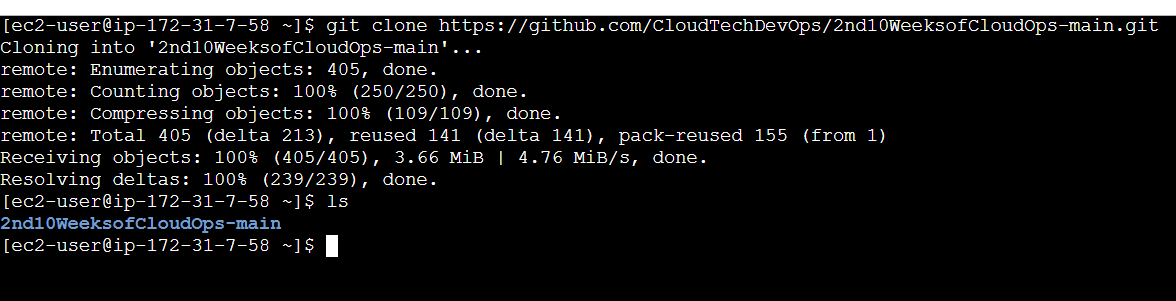
sudo service docker start

sudo chmod 777 /var/run/docker.sock



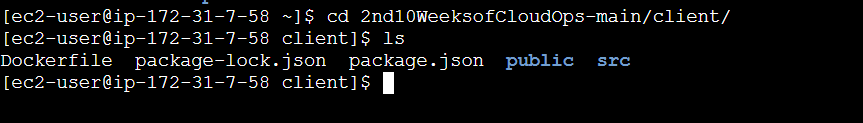
* Clone the git repo for docker image build purpose

git clone https://github.com/CloudTechDevOps/2nd10WeeksofCloudOps-main.git



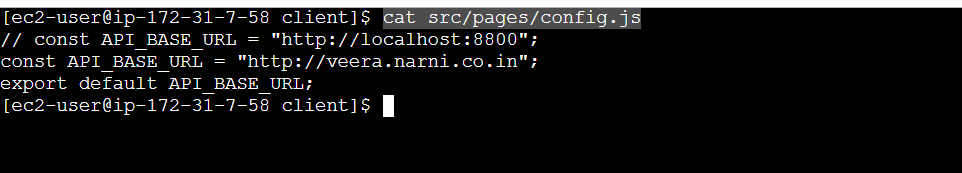
* Switch to client directory

cd 2nd10WeeksofCloudOps-main/client

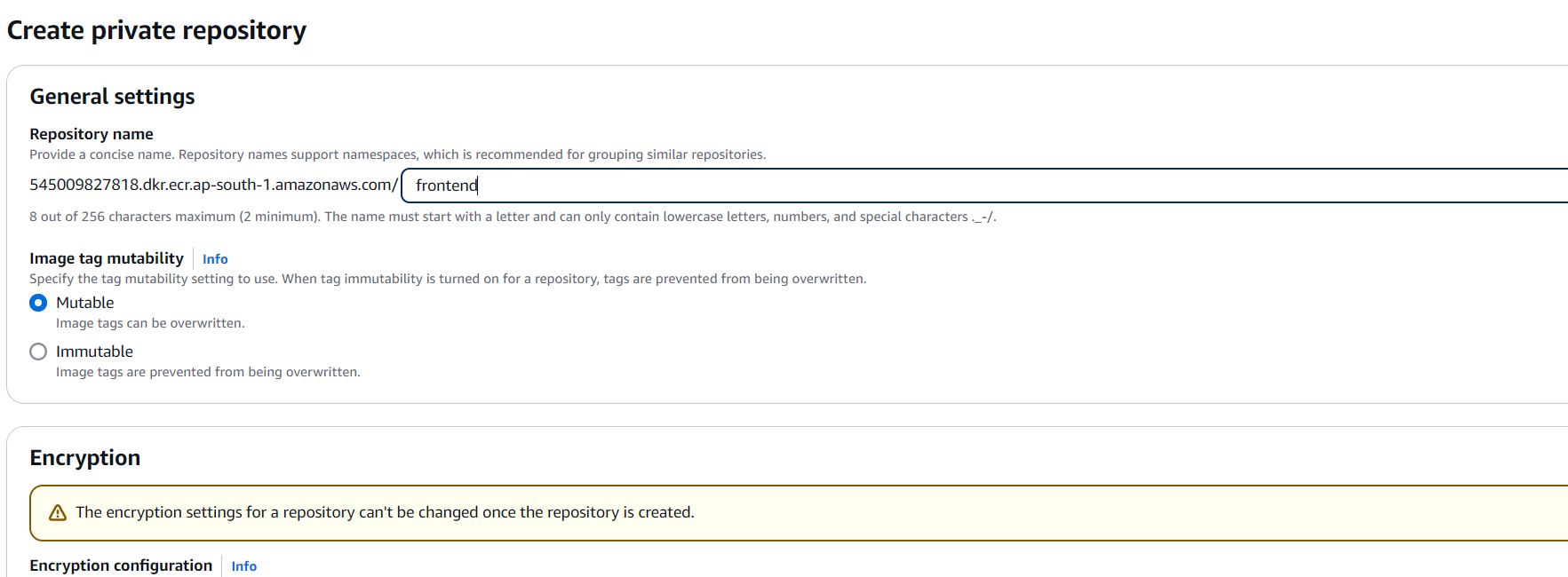


* Change the api base url according to the your requirement

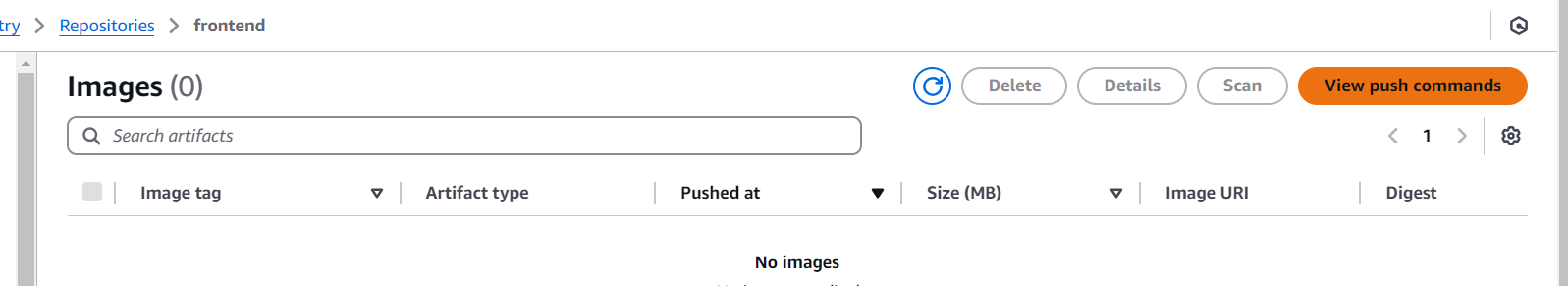
vi src/pages/config.js



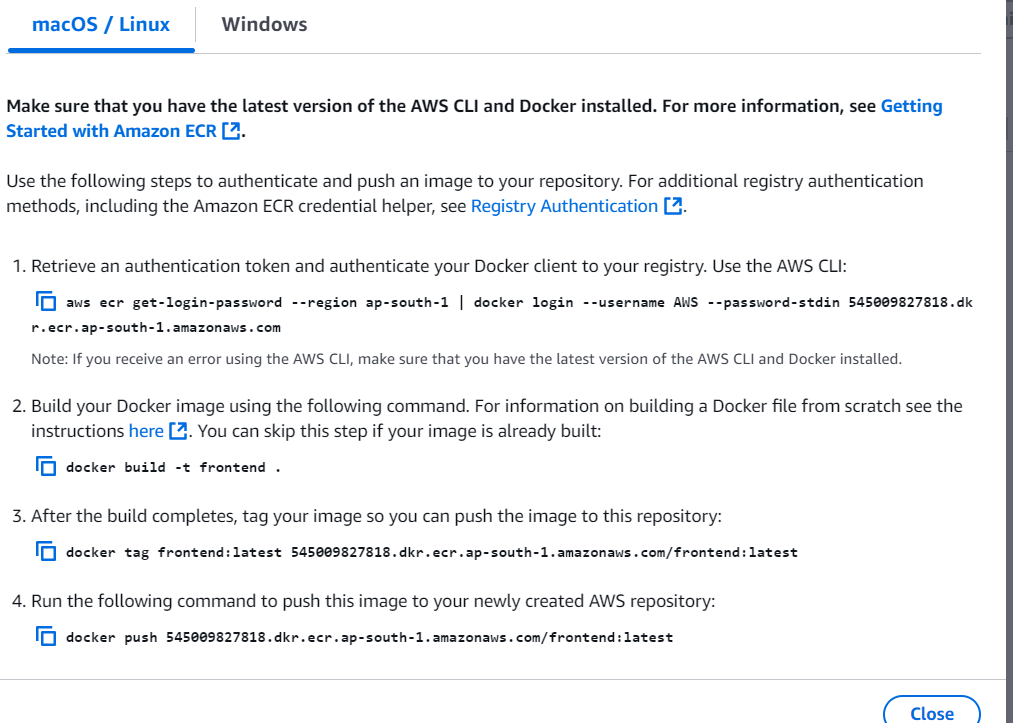
* Create a ecr repository named as frontend



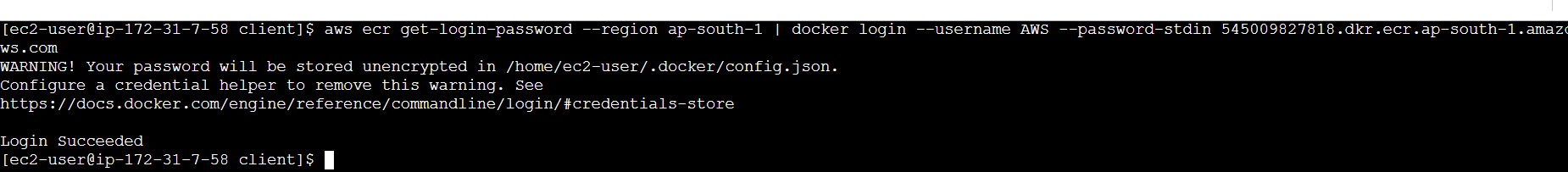
* Open the frontend ecr and click on **view push commands**



* Copy all commands paste it on ec2 one by one in client directory
* Copy the 1st login command and paste it on ec2 client directory



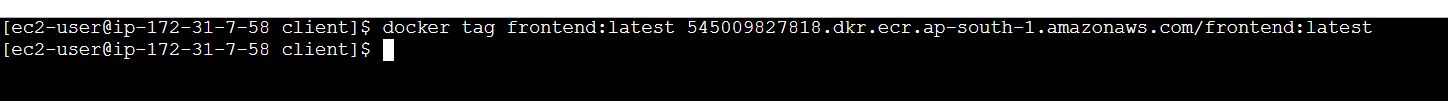
* Copy the 1st login command and paste it on ec2 client dirictory



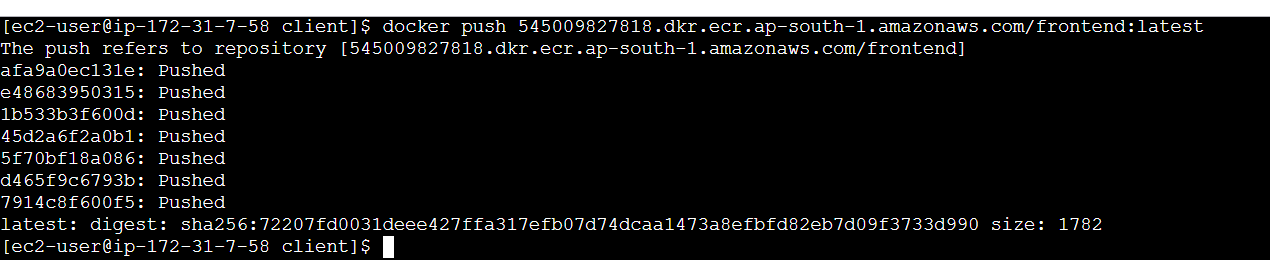
* Copy the 2nd build command and paste it on ec2 client directory



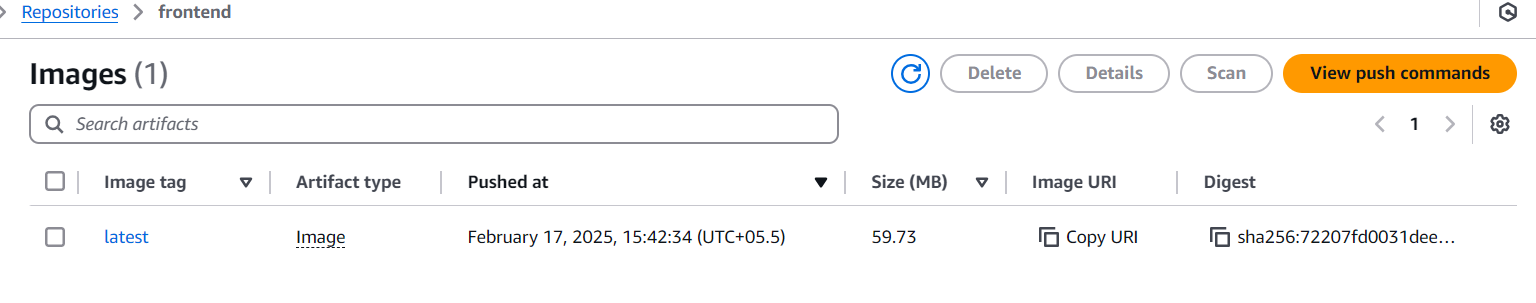
* Copy the 3rd tag command and paste it on ec2 client directory



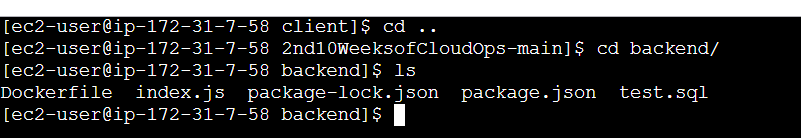
* Copy the 4th push command and paste it on ec2 client directory



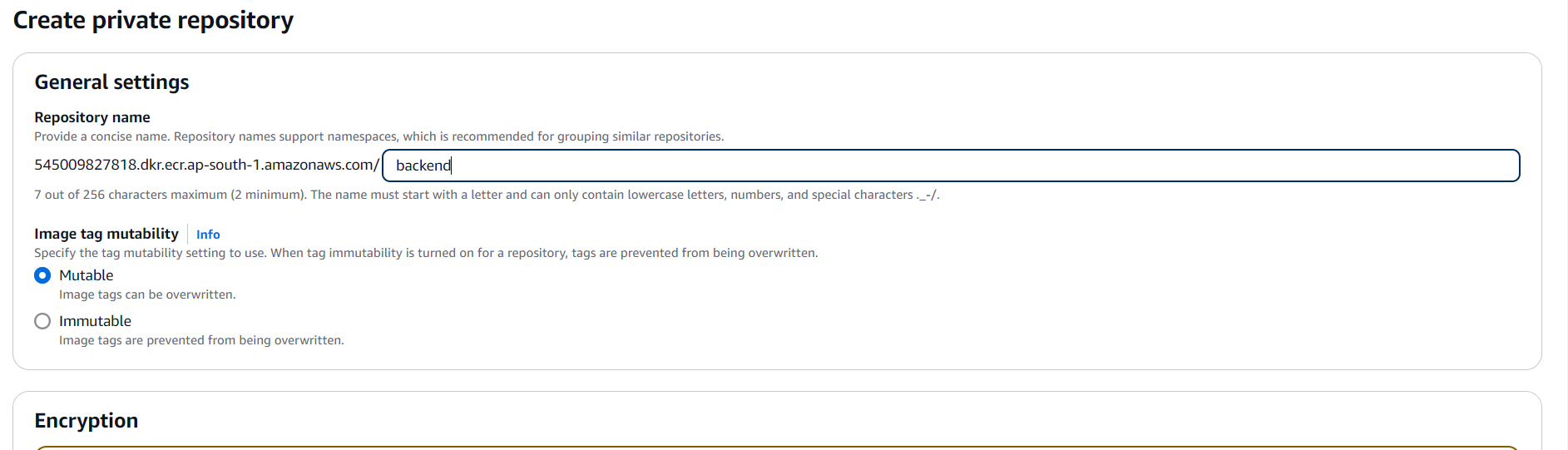
* Frontend image is pushed into ecr successfully



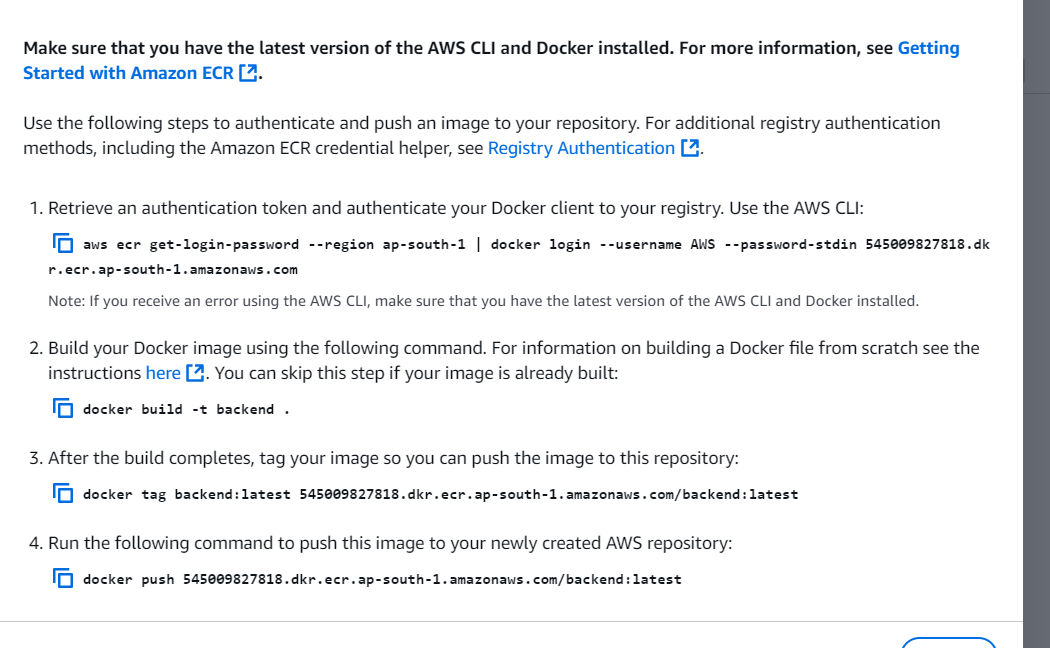
* Switch to backend directory



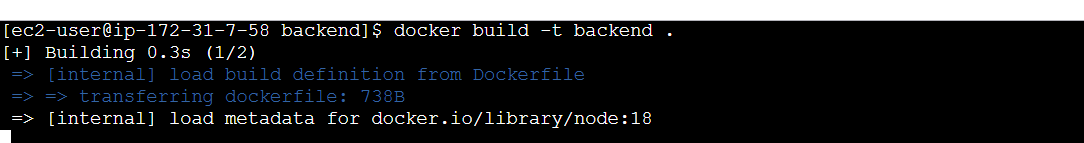
* Create ecr repo for backend named as backend



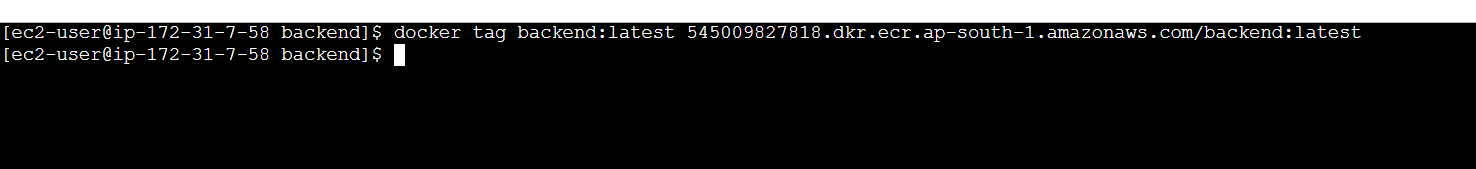
* Copy all commands paste it on ec2 one by one in backend directory
* Copy the 1st login command and paste it on ec2 backend directory



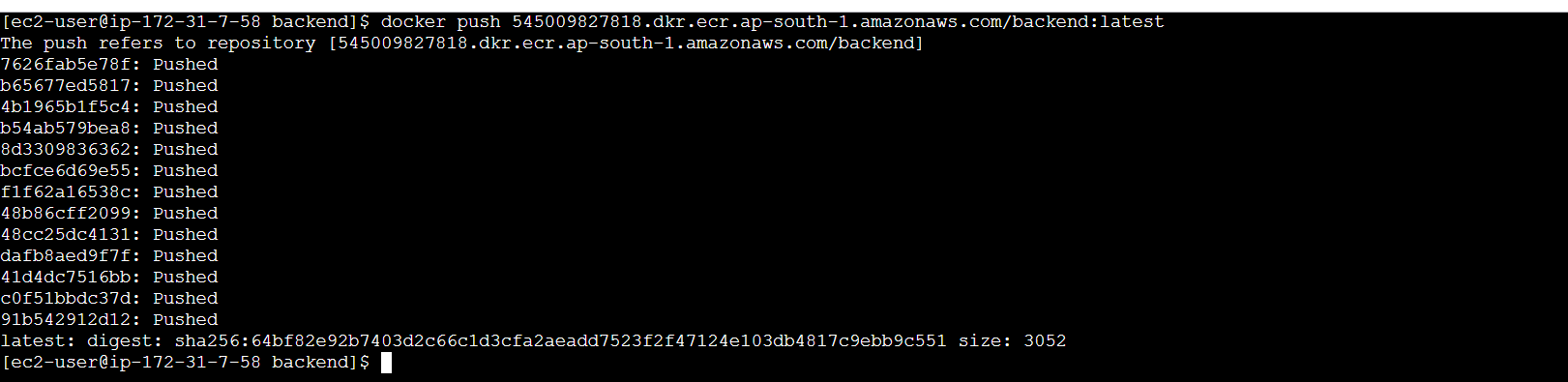
* Copy the 2nd build command and paste it on ec2 backend directory



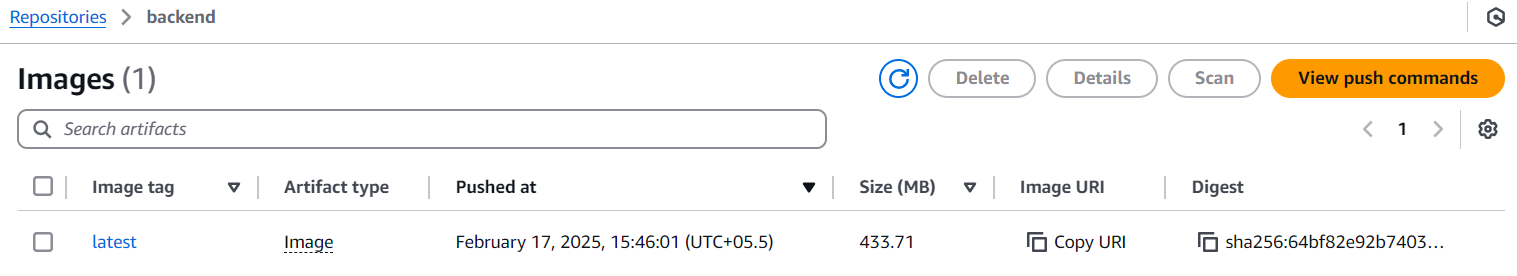
* Copy the 3rd tag command and paste it on ec2 backend directory



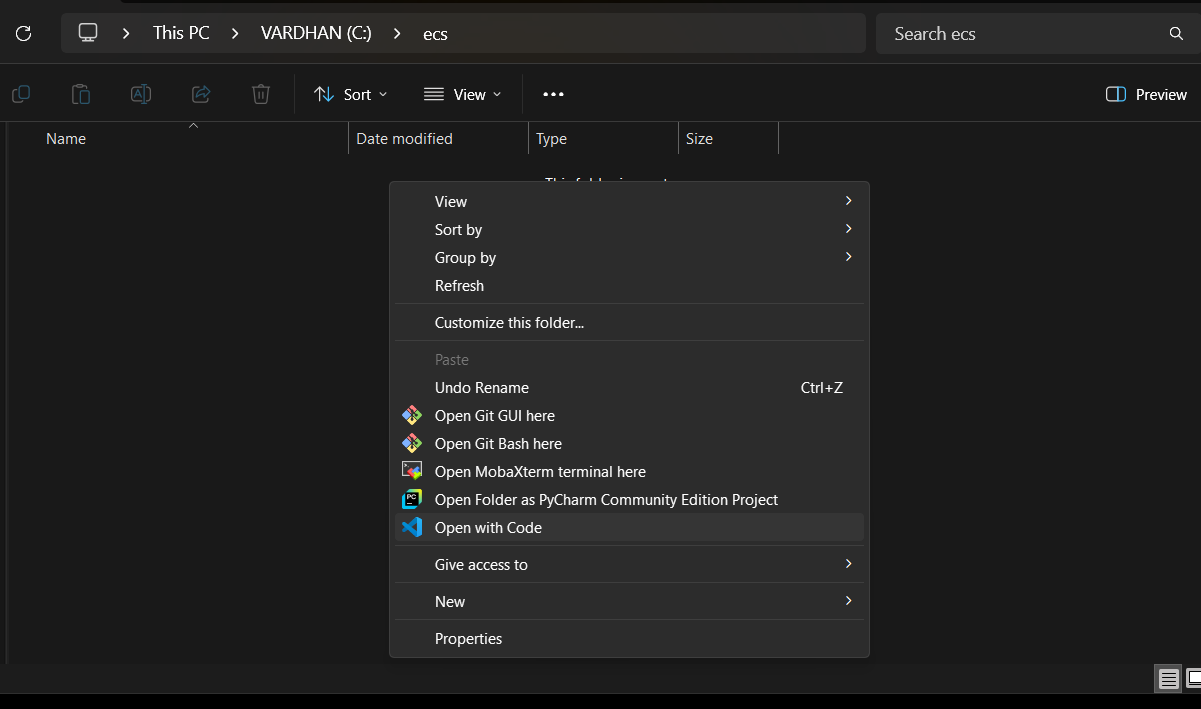
* Copy the 4th push command and paste it on ec2 backend directory



* Pushed the backend image to ecr successfully

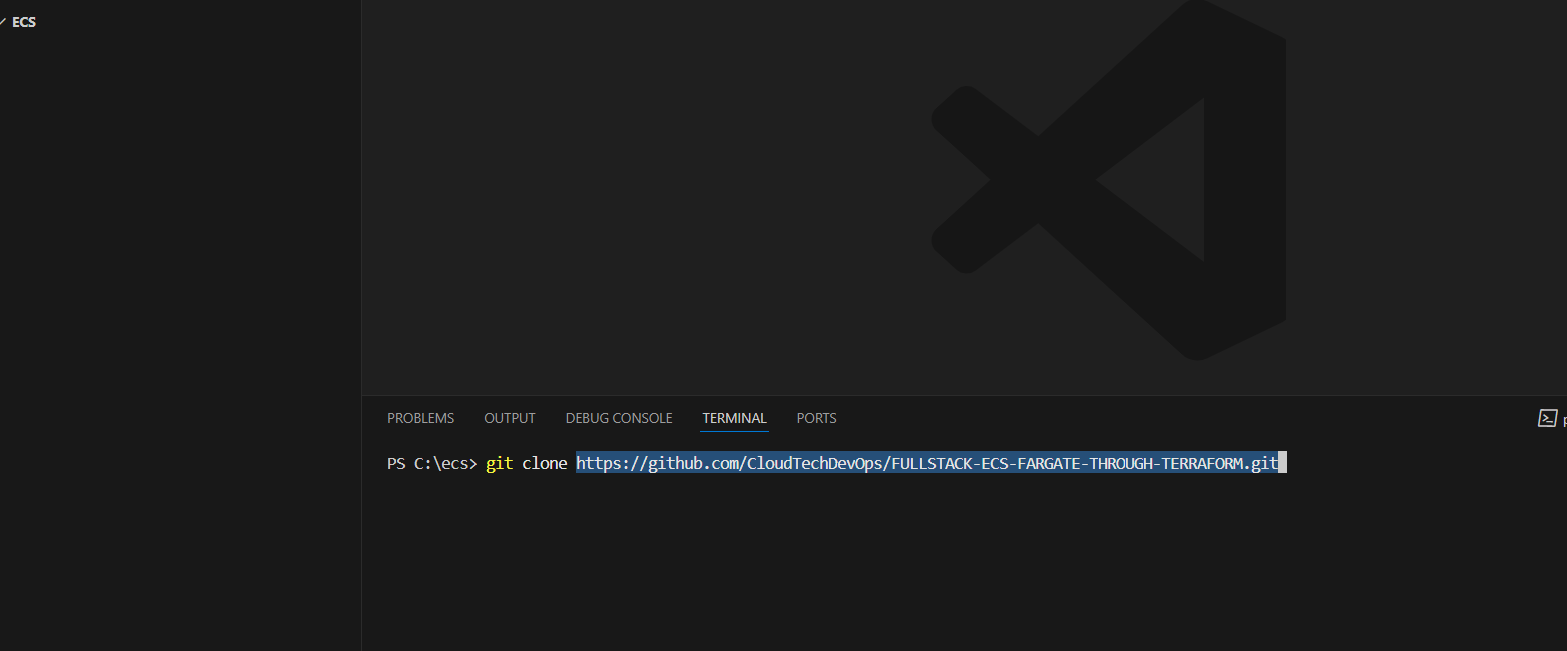


* In your local drive create a folder
* Open that folder In vs code

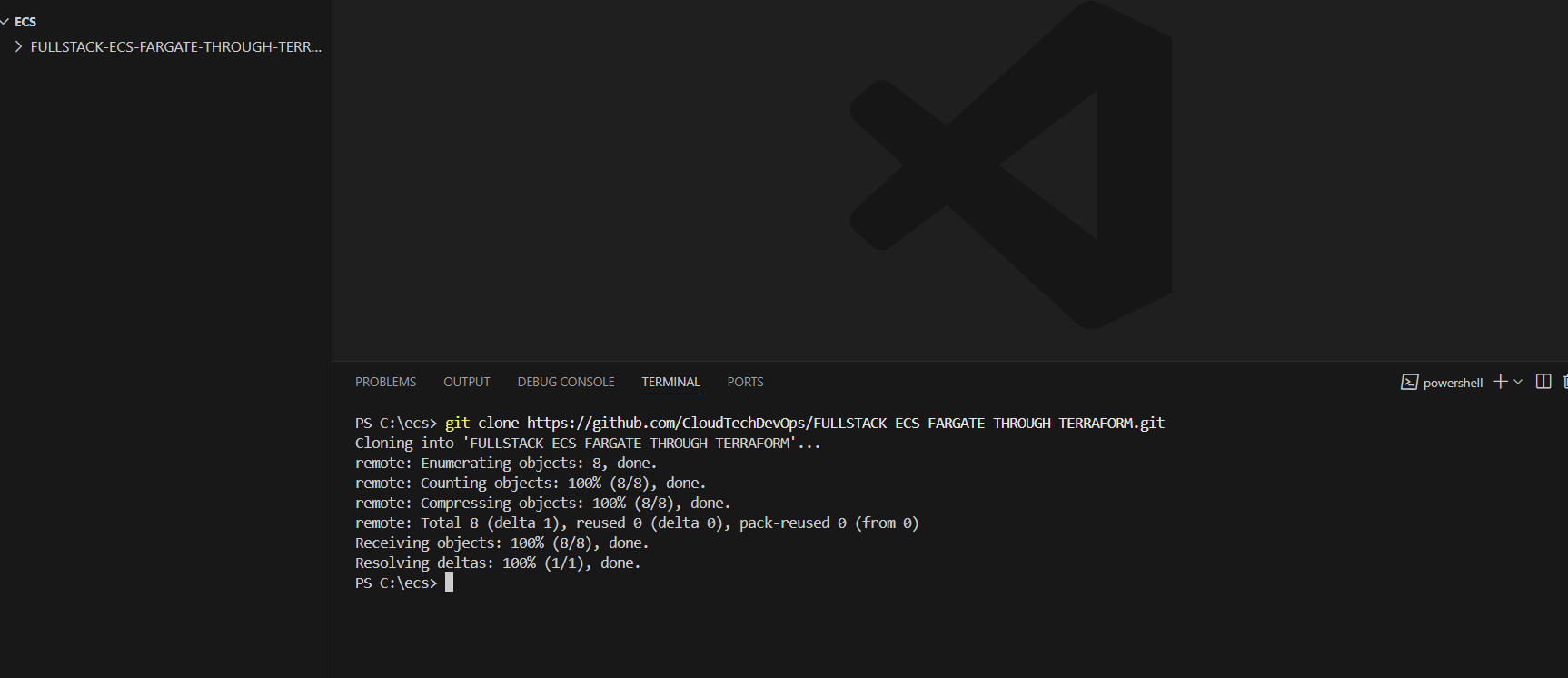


* Configure the aws credentials in local macine by using aws configure
* Prerequisites for this process aws cli,terraform,vs code must be installed on your local machine
* After that Just clone the ecs terraform repo into your local

git clone https://github.com/CloudTechDevOps/FULLSTACK-ECS-FARGATE-THROUGH-TERRAFORMM.git

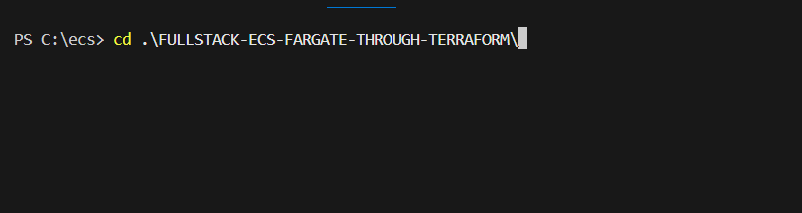


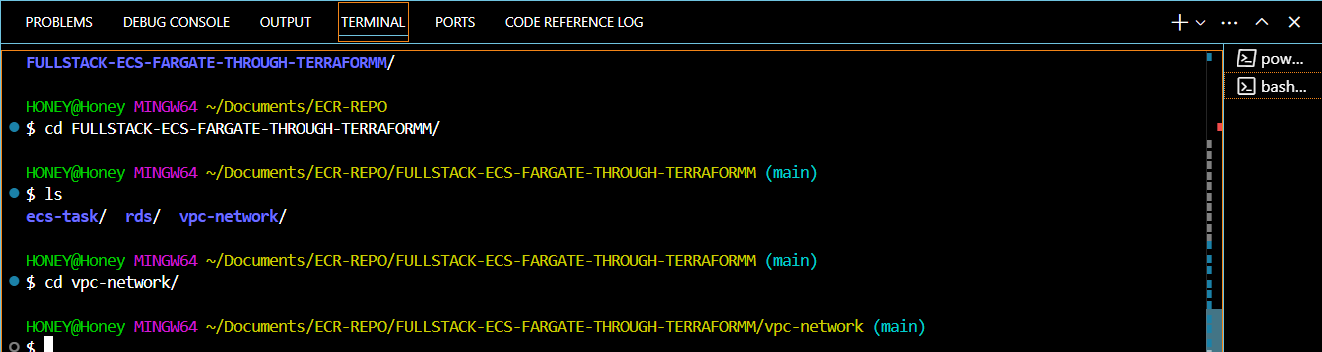
* Cloned the repo successfully



* Switch to repo

cd [FULLSTACK-ECS-FARGATE-THROUGH-TERRAFORM](https://github.com/CloudTechDevOps/FULLSTACK-ECS-FARGATE-THROUGH-TERRAFORM.git)M





* **cd vpc-network**

Run terraform init, terraform plan and terraform apply

Then give cd ..

* **After change the “rds” Directory.**

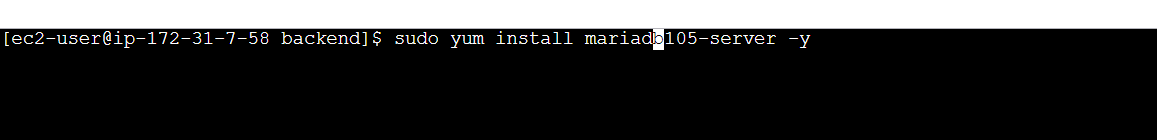
Run terraform init, terraform plan and terraform apply – **after creating a rds you should launch one manual server**

\*install these commands on the server

sudo yum install git -y

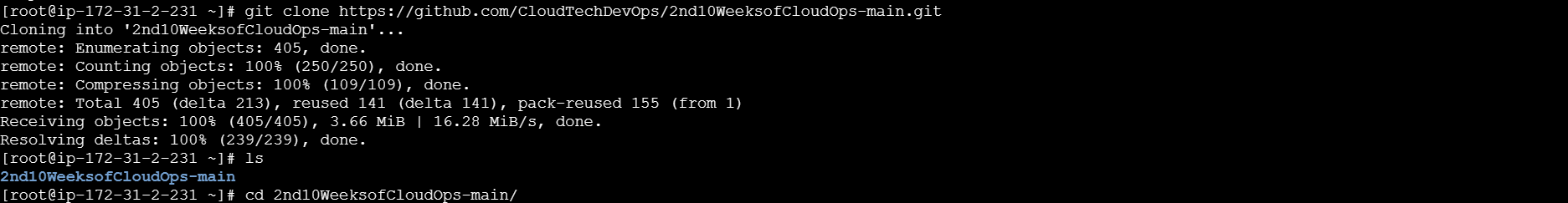
sudo yum install mariadb105-server -y



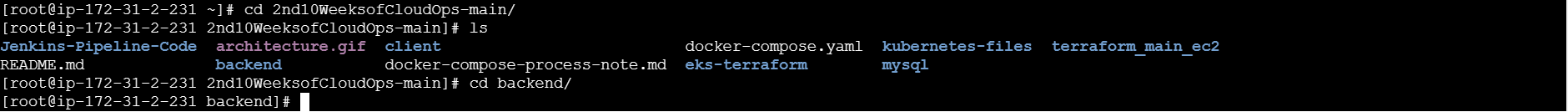


After installilng the commands, you need to **git clone** the repository.

Git clone - https://github.com/CloudTechDevOps/2nd10WeeksofCloudOps-main.git



After that switch to the backend directory using the cd command.



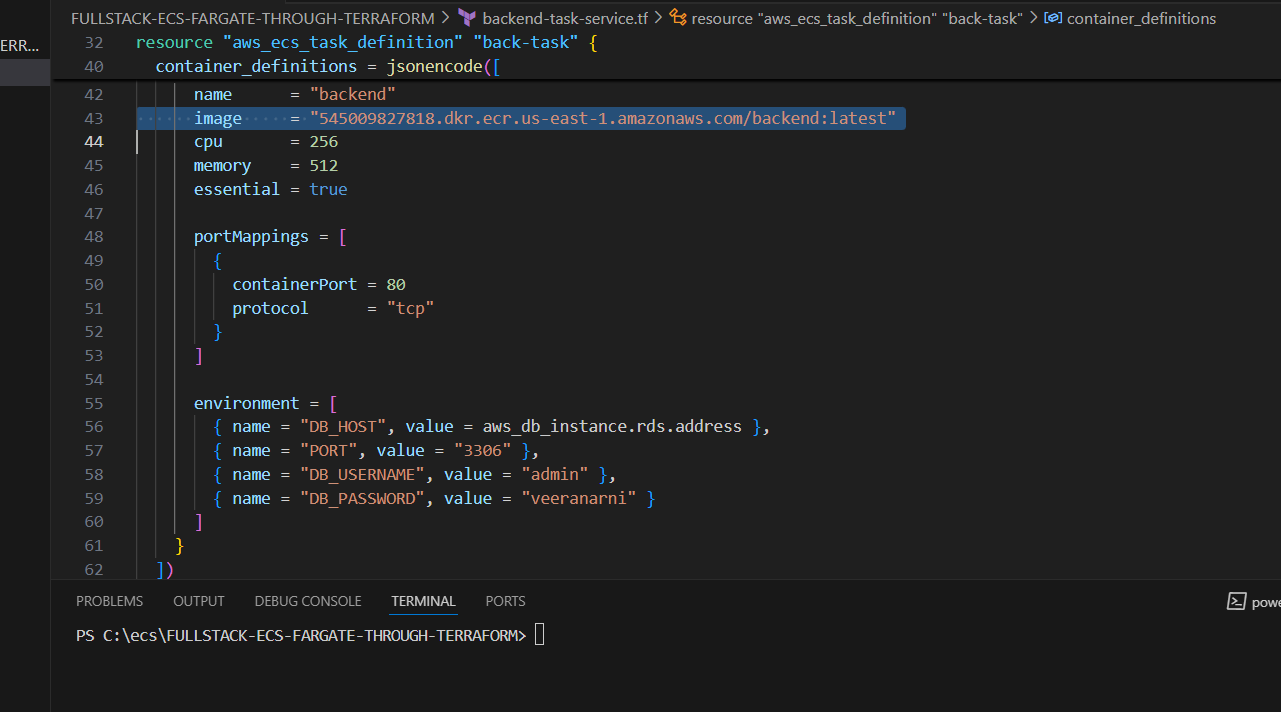
After switching run the command mentioned below

mysql -h rds-endpoint -u admin -pveeranarni < test.sql

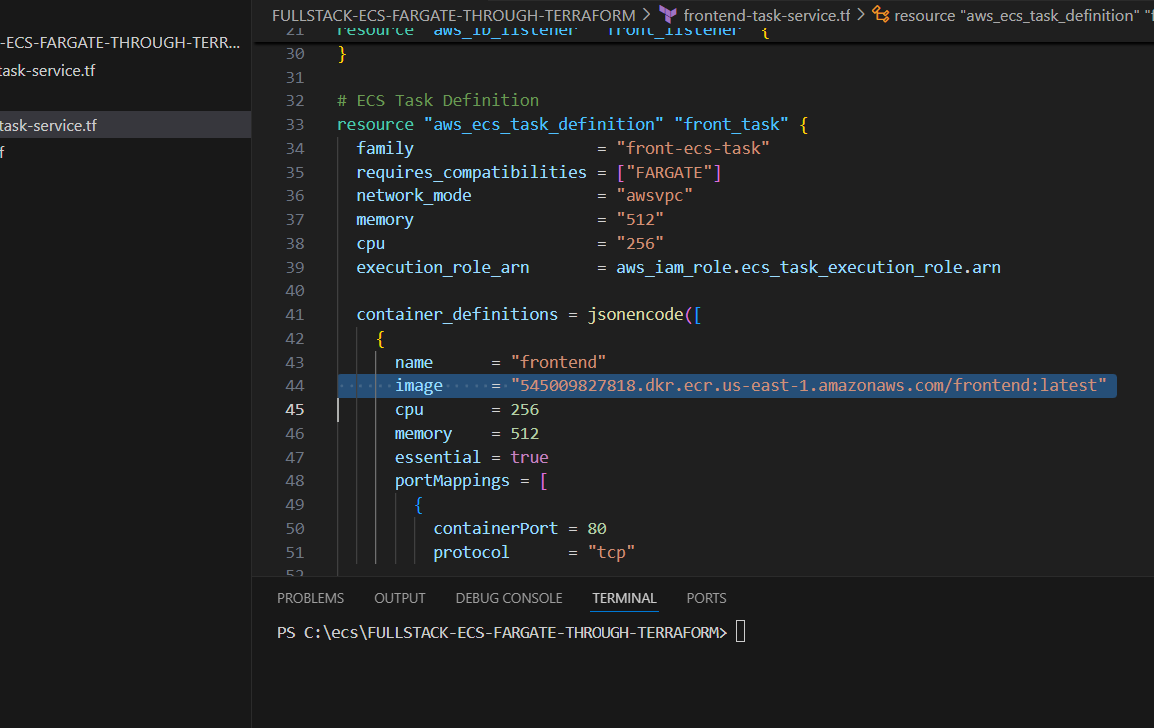
* Intilize database in backend directory. Before running the command check your project directory must be in backend directory

Come back to VS code then change the **ecs-task** directory. And add **rds endpoint** near the **db host**.

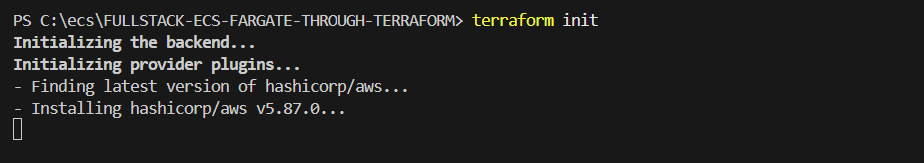
* Change the ecr image urls in the frontend,backend task files



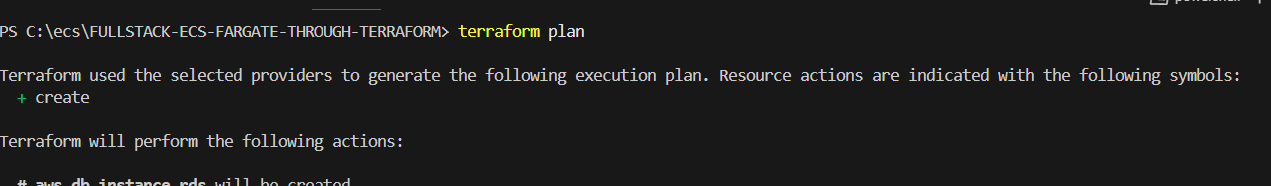
* Change the image urls



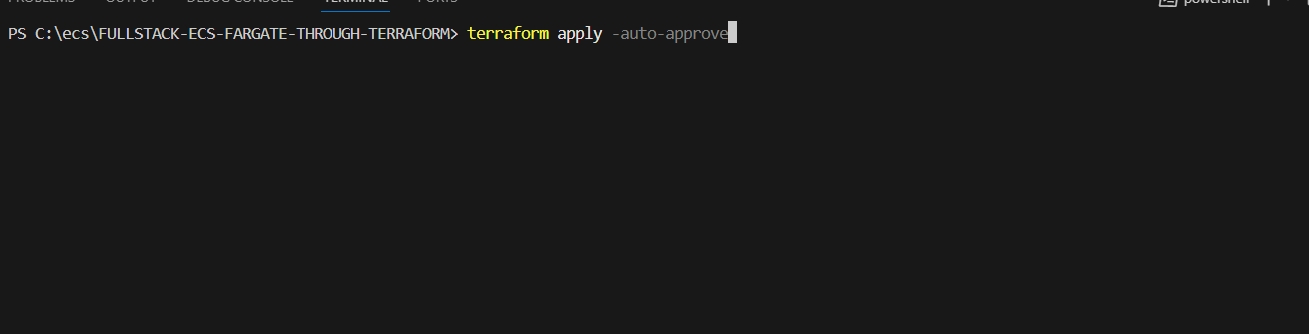
* Give the terraform init



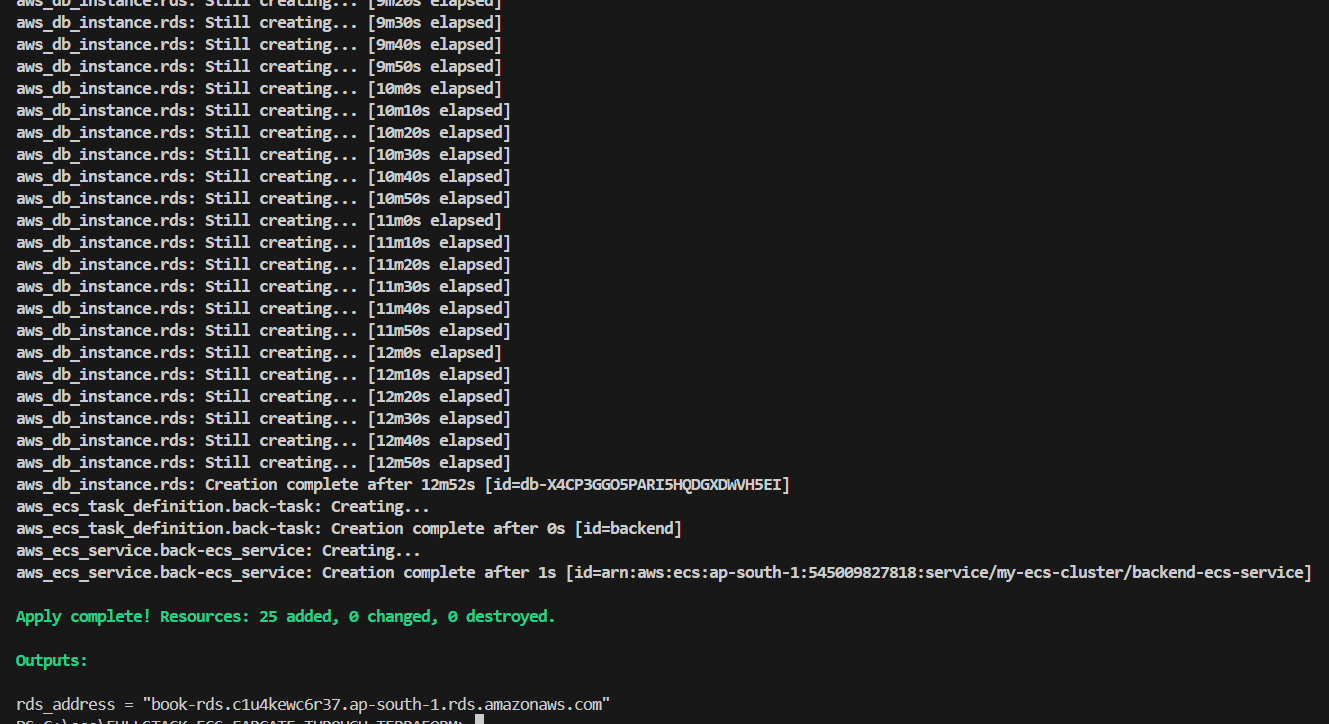
* Give the terraform paln



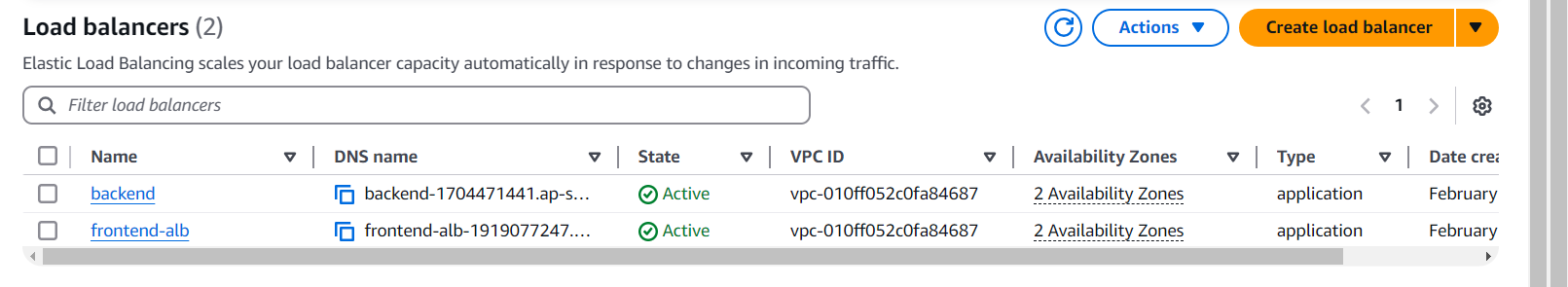
* Give the terraform apply

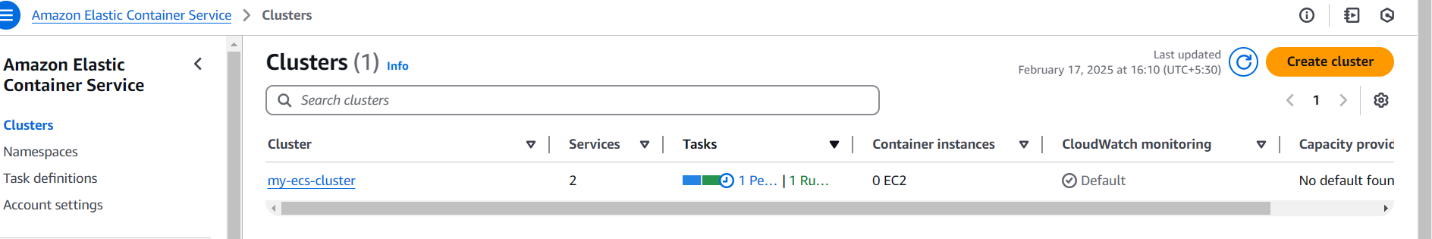


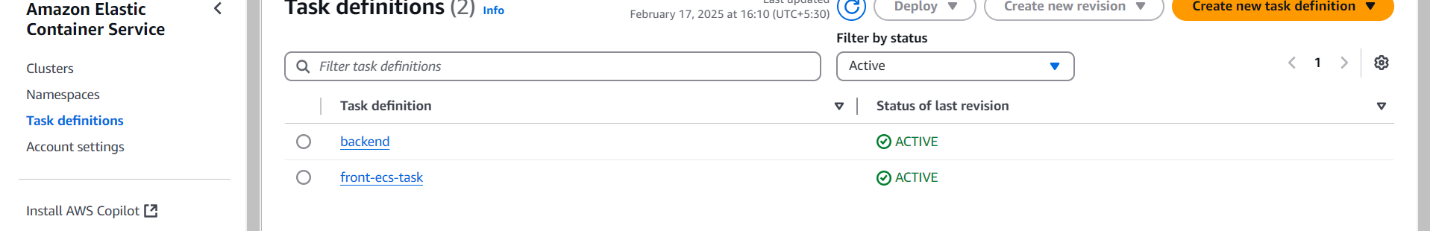
* Resources created



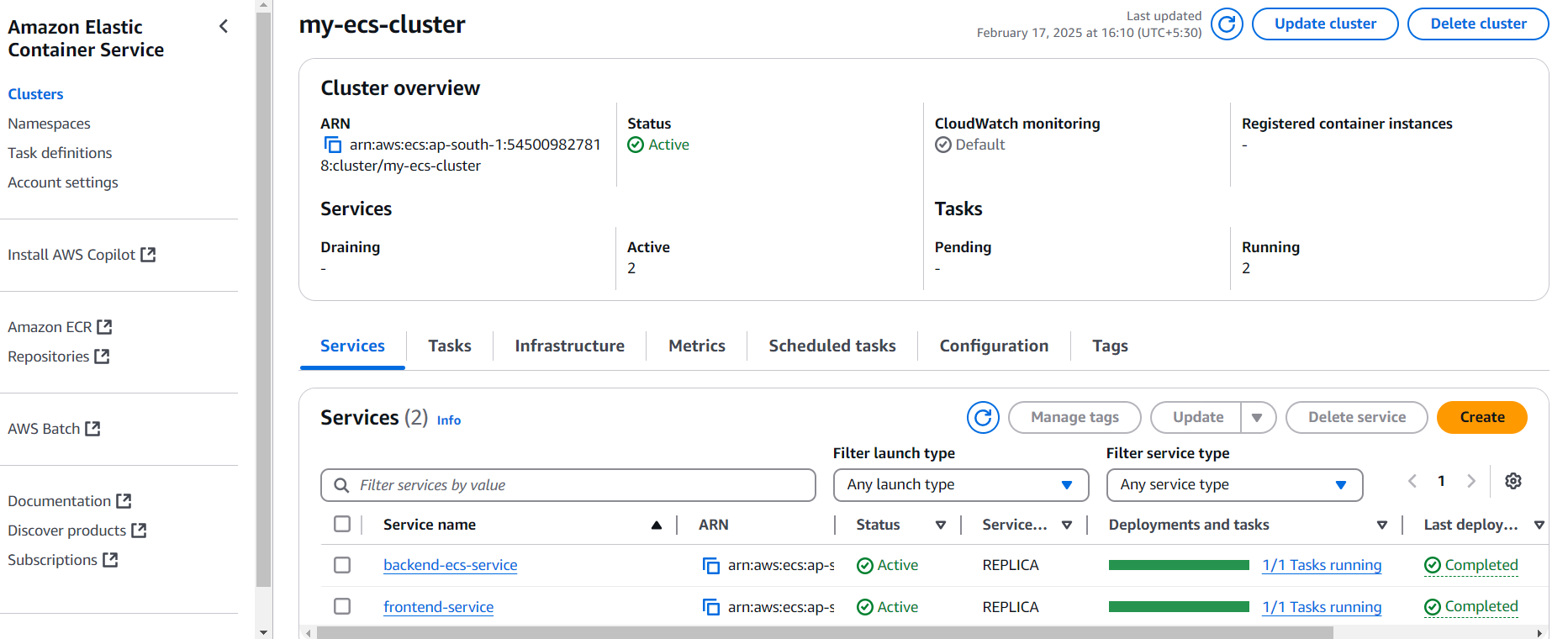
* Open the console and check the load balancers



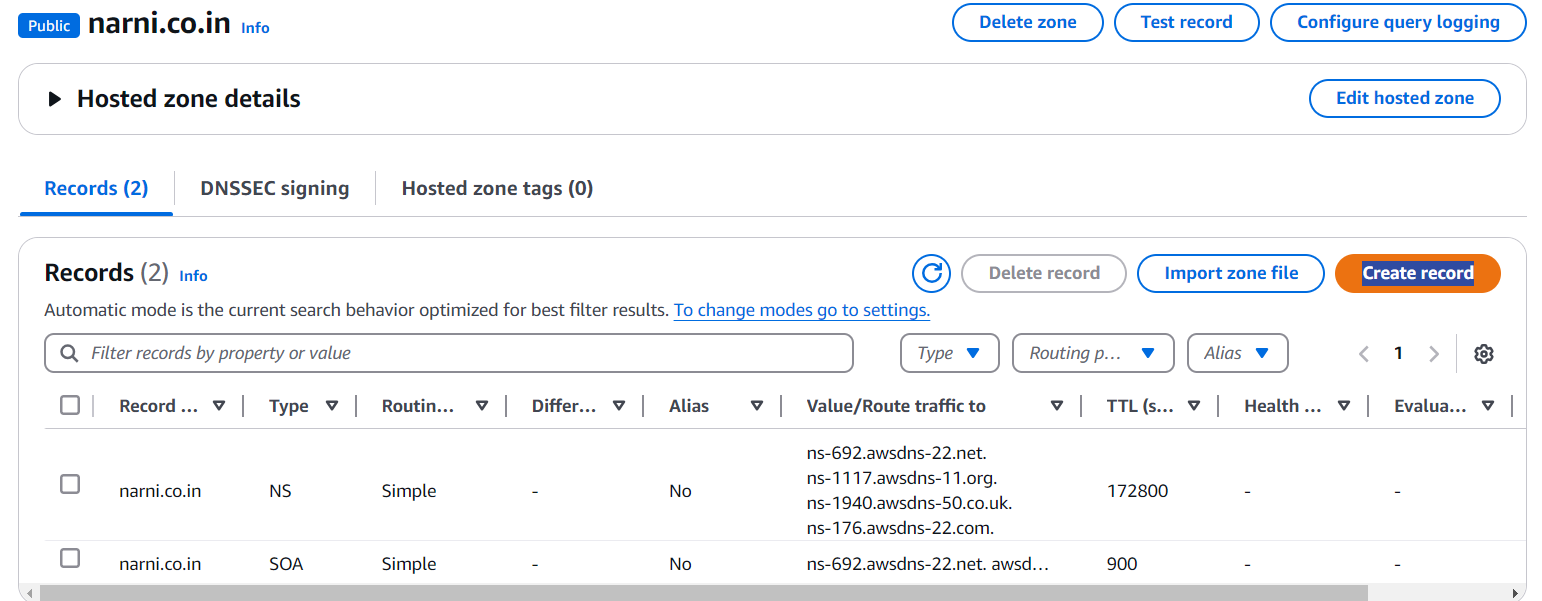
* Open the ecs check the cluster
* check the task definitions



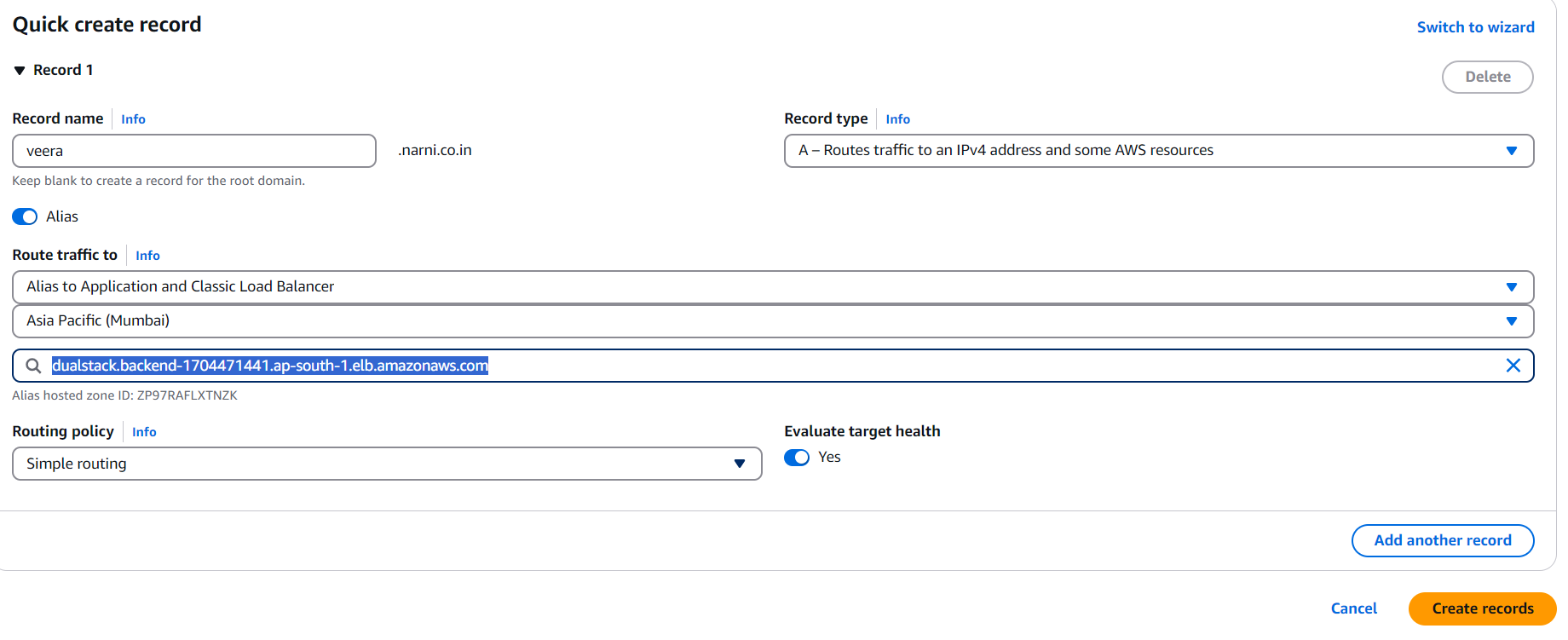
* Open your cluster and check the task status



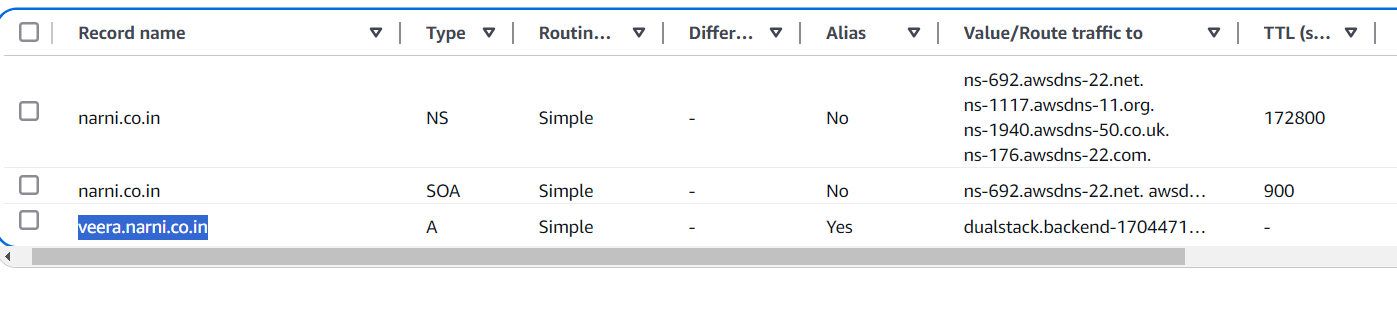
* Open your domain name
* Click on create record



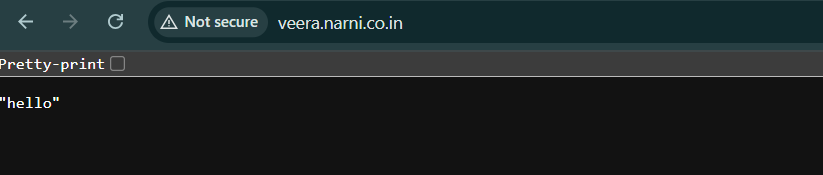
* Give sub domain as per your config.js value
* Select alias
* Select application and classic loadbalancer
* Select load balancer region
* Select your backend load balancer
* Create the record



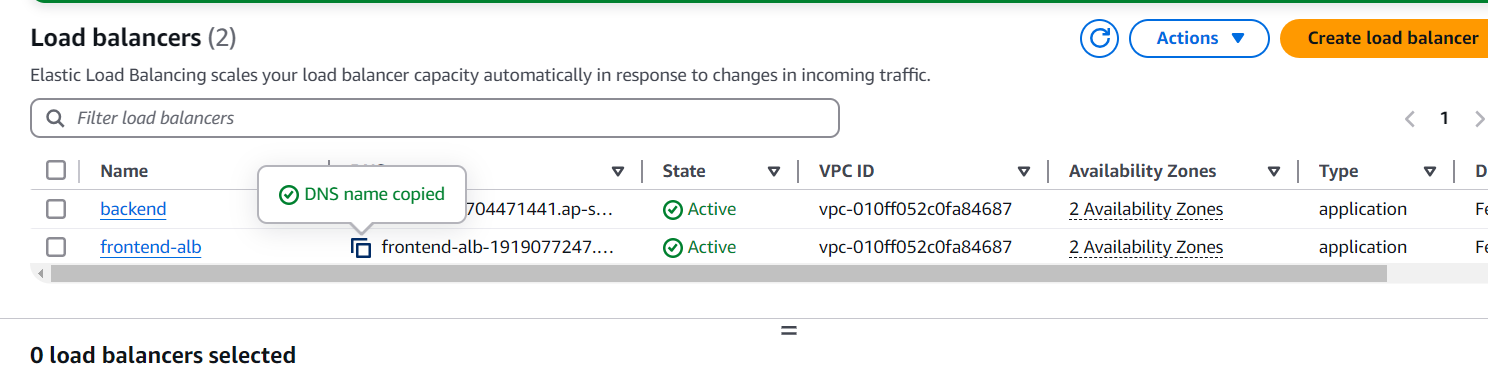
* Copy the record name



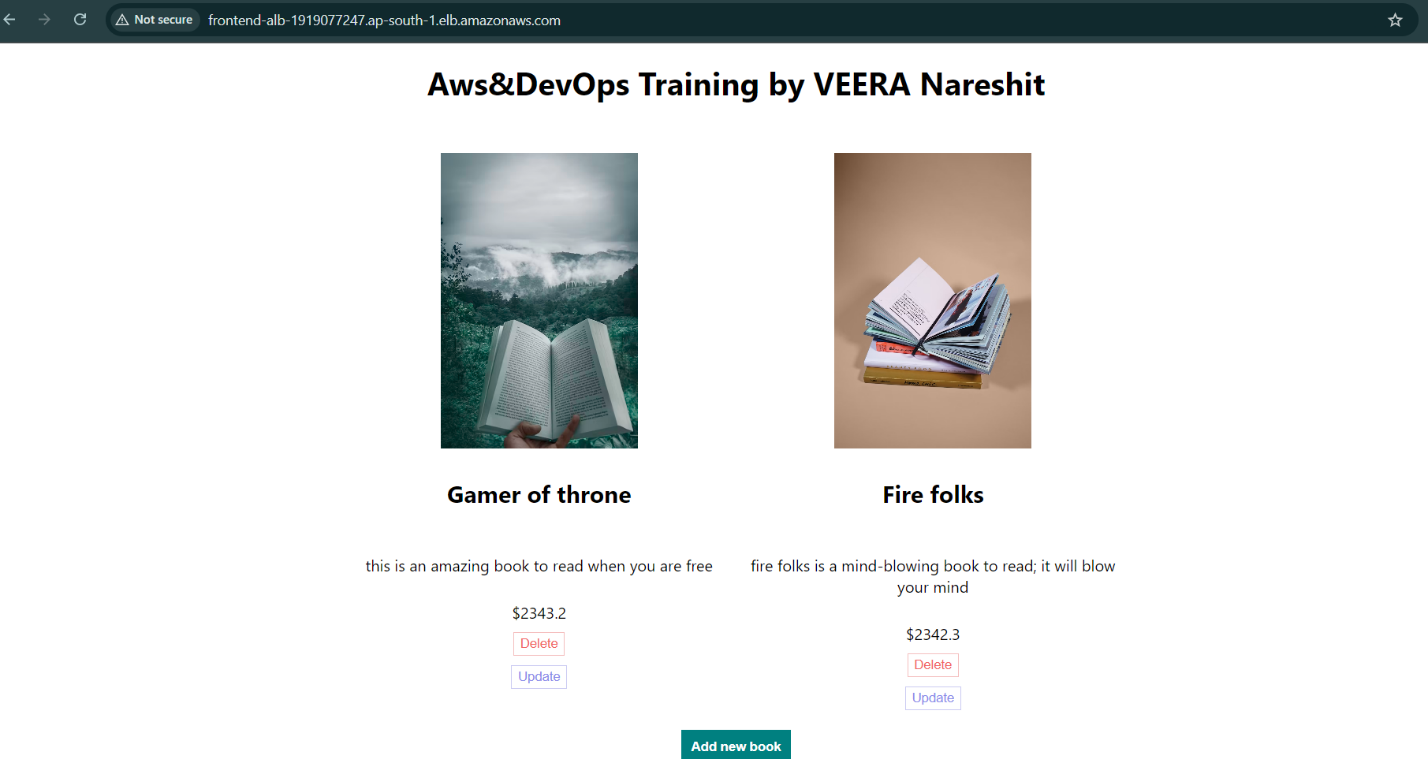
* Search on browser you will get hello response



* Copy the frontend loadblancer url



* Search on web you will get final out put
* Click on add book



-----------------------------------------------------THANK YOU----------------------------------------------------------