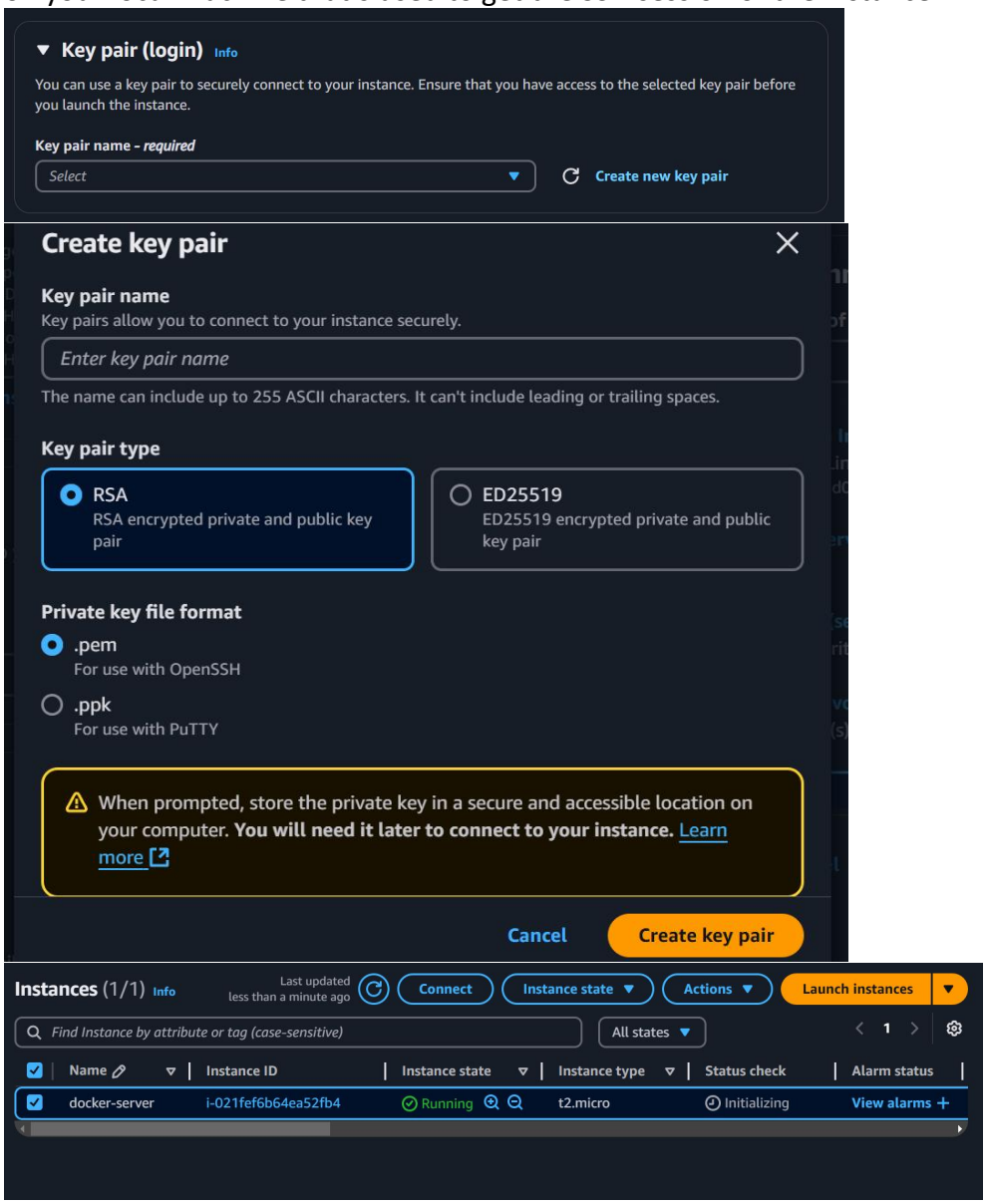
RUNNING DOCKER ON AWS EC2 INSTANCE AND DEPLOY A STATIC HTML PAGE THROUGH ECS

Launch an EC2 instance through AWS console with desired OS (preferably Ubuntu)
Create a new key pair on the launch window. A file with key pair will be downloaded on your local machine that's used to get the SSH session of the instance.



Once the instance is launched, connect to the EC2 instance by selecting the server and clicking on connect button. Then go to the SSH client tab and copy the example command.

Now open terminal and change directory to where the key pair file is located.
Run the command so you can enter the SSH session of the EC2 instance.

In the SSH session of EC2 instance, download Docker engine for Ubuntu. Instructions to install Docker engine can be found on official Docker page.

https://docs.docker.com/engine/install/ubuntu/

Now that your EC2 instance has Docker installed, we are ready to dockerize the application. Change directory to where your source code is.
I have a very basic HTML page

```
<!DOCTYPE html>
<html>
<head>
<title>My HTML Page</title>
</head>
<body>

<h1>Welcome to My Website</h1>
<p>This is a test HTML page which I will be using in my projects.</p>

</body>
</html>
```

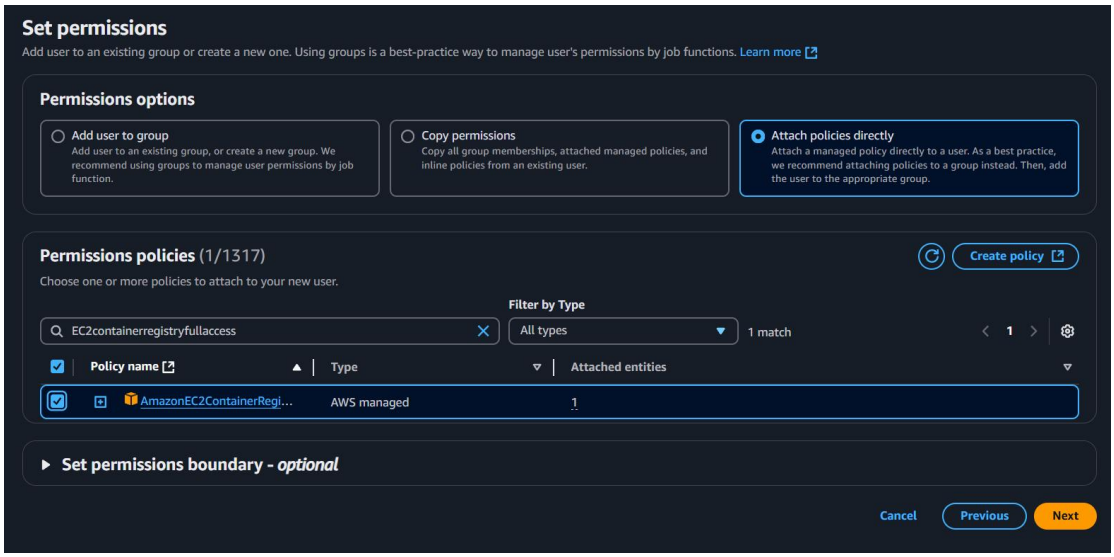
Now create a Dockerfile to build image with your source code.

```
FROM nginx:latest

COPY . /usr/share/nginx/html

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```

Now in order to push the image to AWS Elastic Container Registry, first install AWS CLI in your EC2 instance

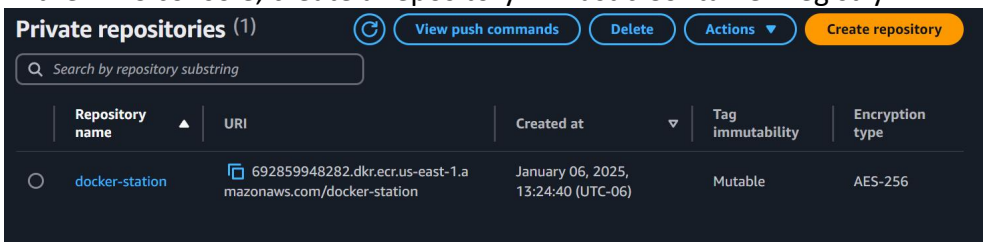
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install

In AWS Console, create an IAM user with policy "AmazonEC2ContainerRegistryFullAccess" assigned to it. This policy will give full access to the ECR to user.



Once the user is created, generate access key for the user and save the access key and secret access key.

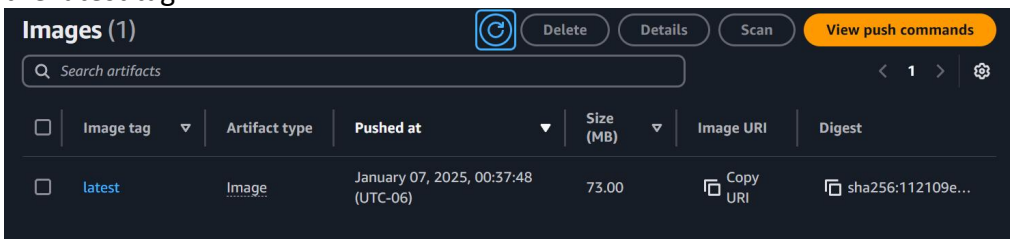
In the terminal, create the credentials by running ‘aws configure’ command and use the access key and secret access key that we generated in the preceding step.
In the AWS console, create a repository in Elastic Container Registry



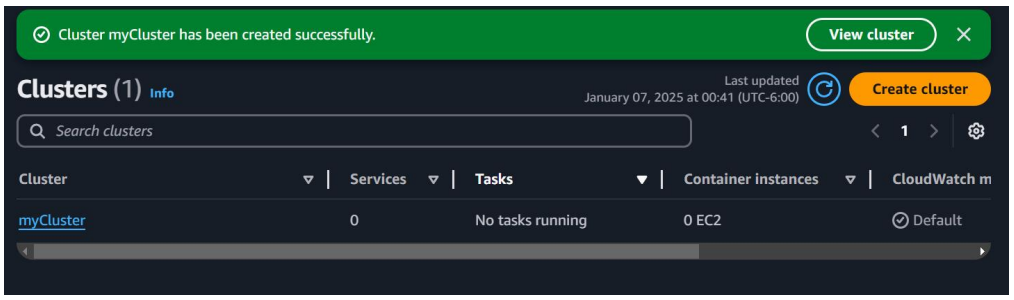
Select the registry and click on “View Push Commands”

And then run each command in the EC2 instance terminal.
Running these commands will retrieve an authentication token and authenticate your Docker client to your registry, then build your docker image, tag your docker image so it is ready to push to the ECR repository, and finally push the docker image to ECR repository.

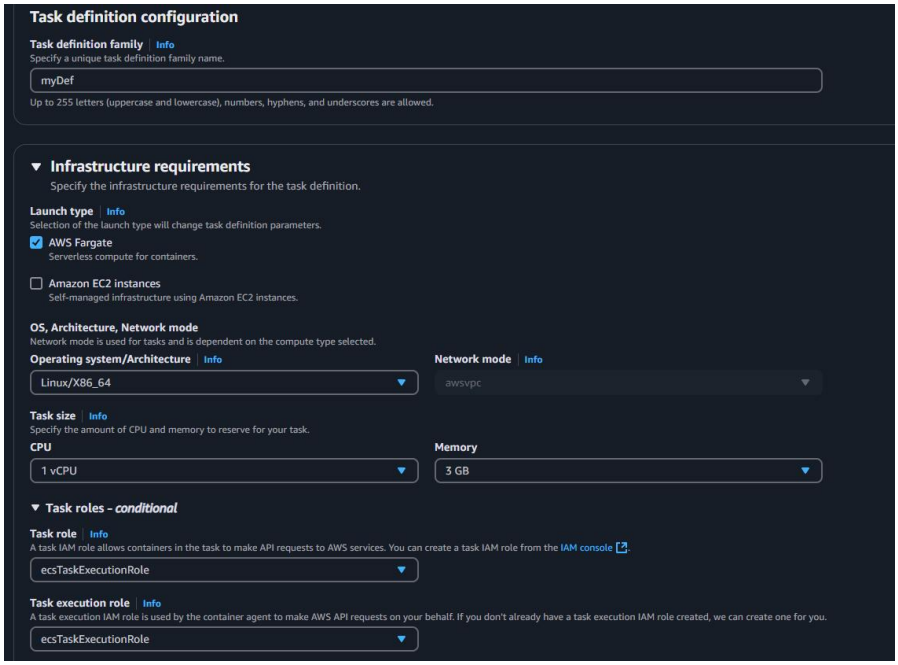
In AWS Console, click on the repository and you will see the image got pushed with the latest tag



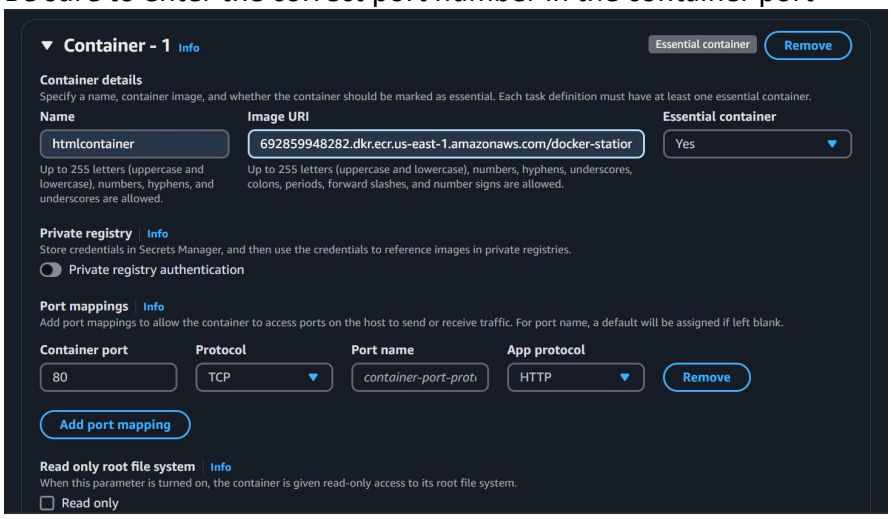
Now copy image URI, and open the Elastic Container Service (ECS)
Create a cluster



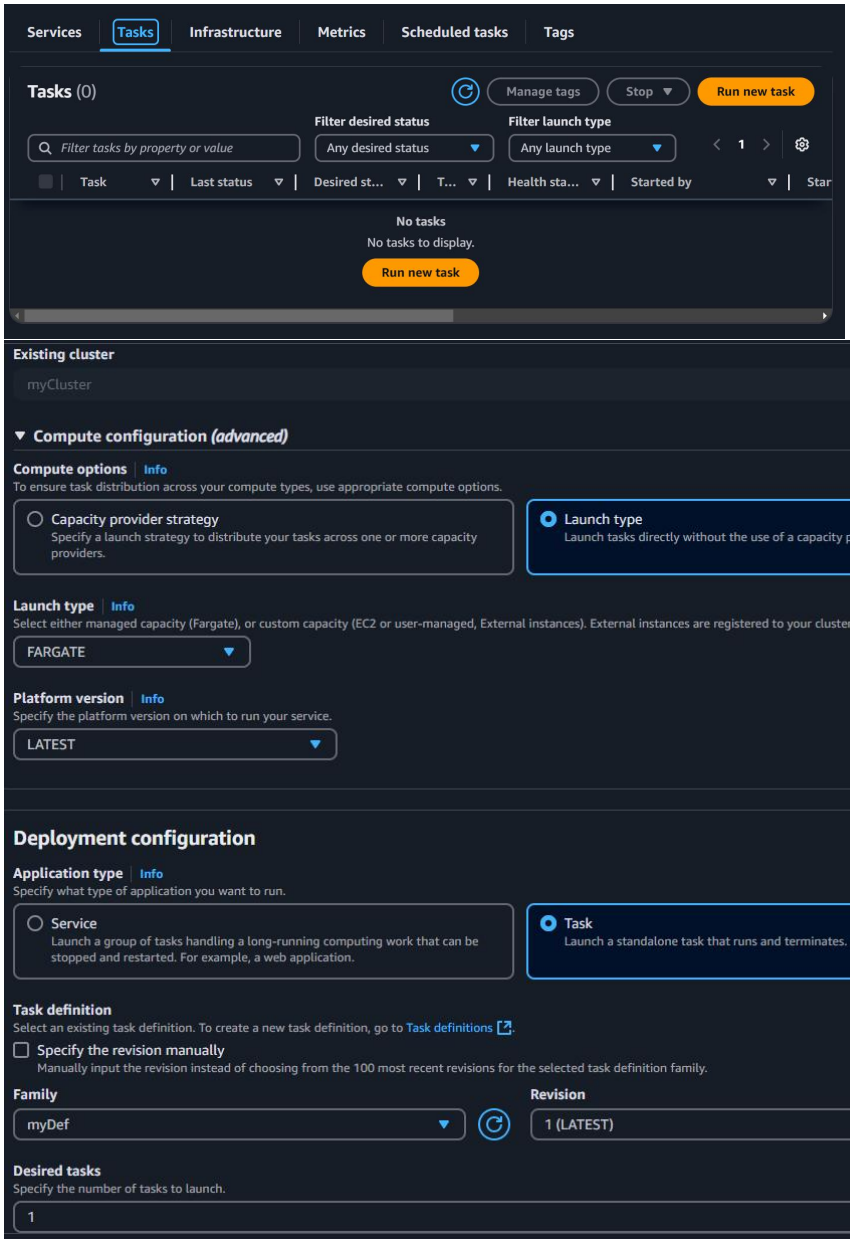
Then create a task definition and select “ecsTaskExecutionRole” in Task Role



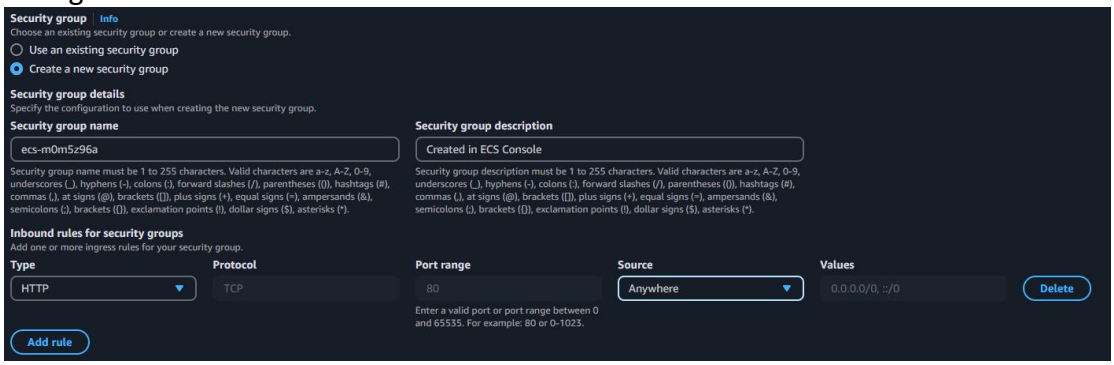
In the essential container, paste the image URI that we copied in the preceding step
Be sure to enter the correct port number in the container port



Once the task is created, go back to clusters, select the cluster, go to tasks tab, and select “Run new task”



In the Networking section, create a custom security group with following configuration



Once the task is running, click on the running task and locate the Public IP Address in the configuration. Open address, and you should be able to see your application there

Welcome to My Website

This is a test HTML page which I will be using in my projects.