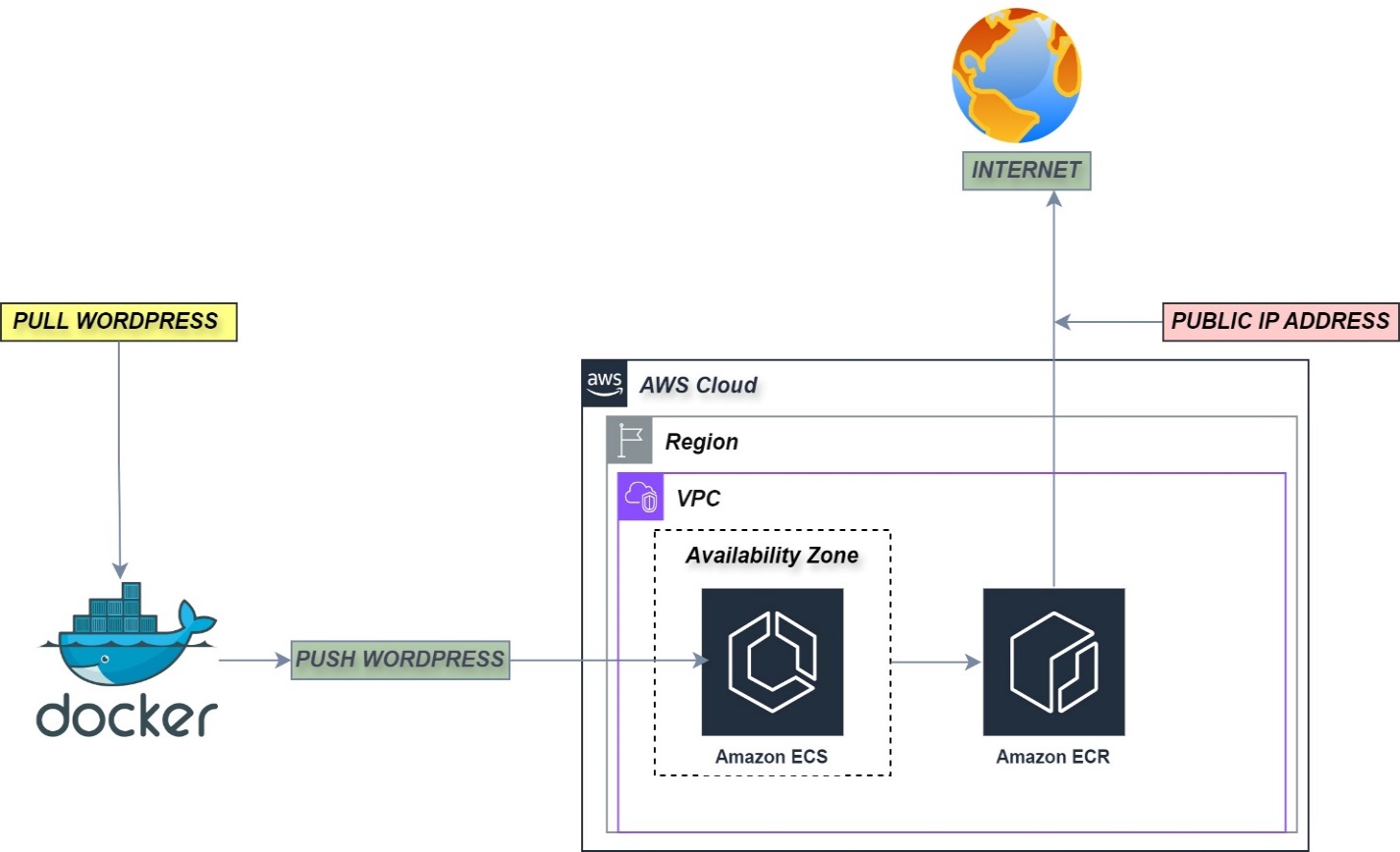
* **WORDPRESS HOSTING WITH ECS/ FARGATE: --**
* ***Project Overview***: -
* ***Objective:*** The project aimed to deploy a WordPress site in a containerized environment using AWS services, specifically ECS (Elastic Container Service) with Fargate. This approach leverages modern container orchestration and serverless computing to achieve scalability, manageability, and high availability for the WordPress application.
* ***Architecture:***

****

* ***COMPONENTS:*** -

1. ***Amazon Elastic Container Service (ECS)****:* A managed container orchestration service that supports Docker containers.
2. ***AWS Fargate****:* A compute engine for ECS that allows you to run containers without managing servers or clusters.
3. ***AWS ECR*:** Managed Docker container registry to store the WordPress Docker image.
4. ***DOCKER:*** Docker is a platform that enables developers to package applications and their dependencies into standardized units called containers. Containers are lightweight, portable, and can run consistently across different environments.
5. ***WORDPRESS***: WordPress is a popular open-source content management system (CMS) used for building and managing websites. It's highly versatile and can be used to create anything from simple blogs to complex e-commerce sites.

* ***Provision EC2 Instance Setup***:
* Launch two EC2 instances in your preferred region.
* Configure security groups to allow necessary traffic (SSH, HTTP, MYSQL)
* ***Launch Virtual Machine using AWS EC2***
* Here is a detailed list of the basic requirements and setup for the EC2 instance i have used for running Docker, including the specifics of the instance type, AMI, and security groups.
* **1. Instance Type**

- Instance Type: `t2.micro`

- vCPUs: 1

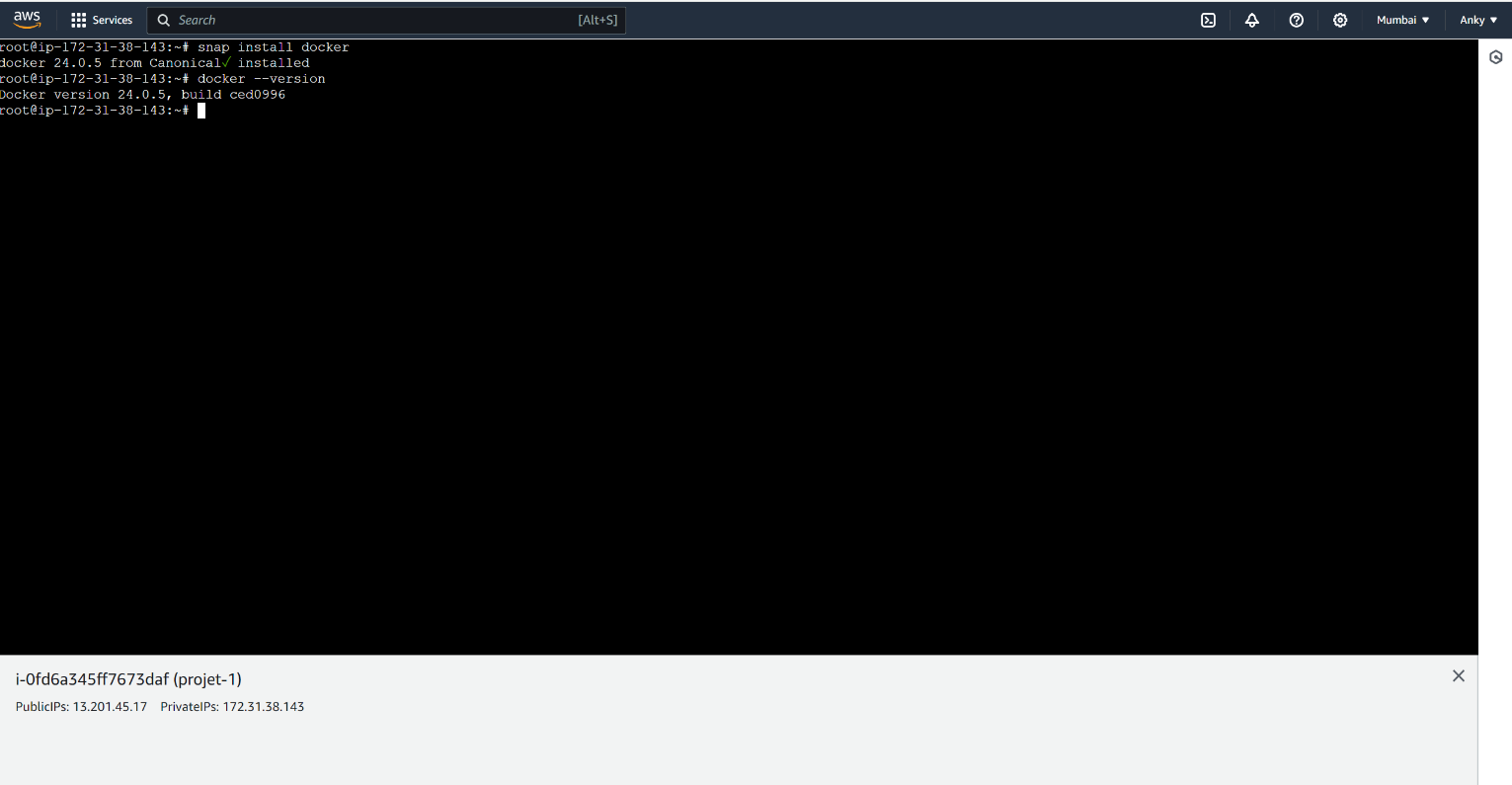
- Memory: 8GB

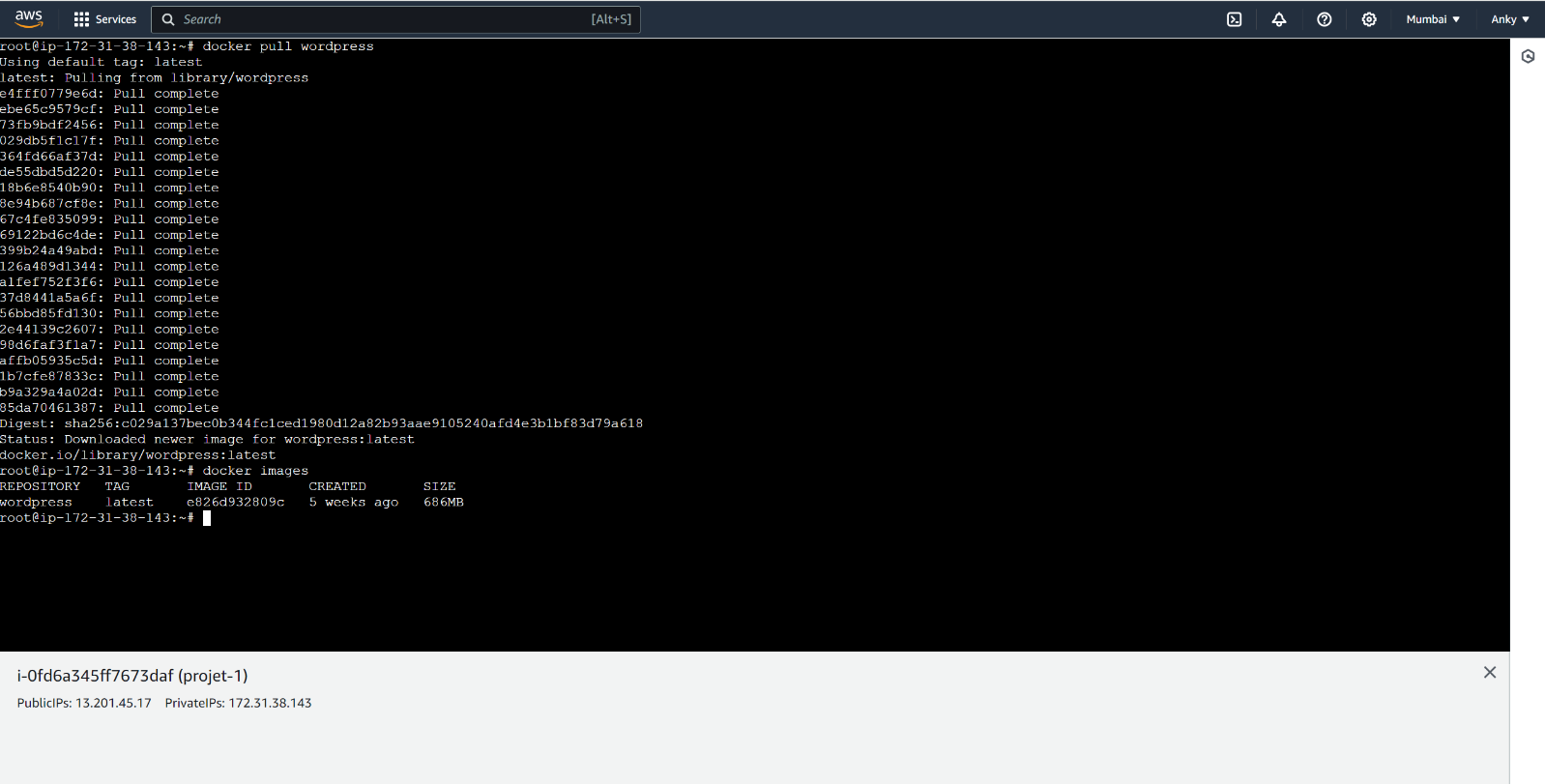
* **2. Amazon Machine Image (AMI)**

- AMI: Ubuntu Server 24.04 LTS (HVM)

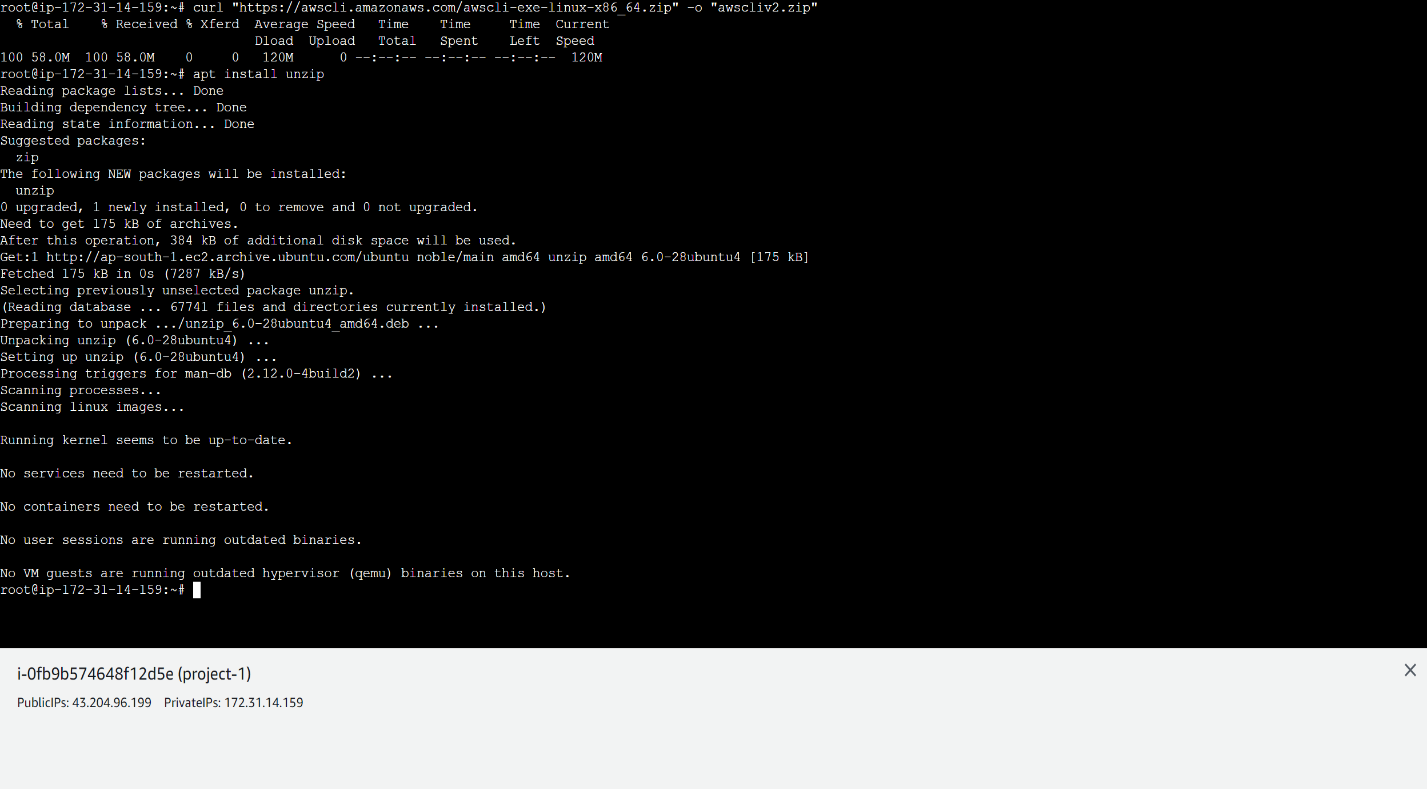
* **3. Security Groups**

- Security groups act as a virtual firewall for your instance to control inbound and outbound traffic

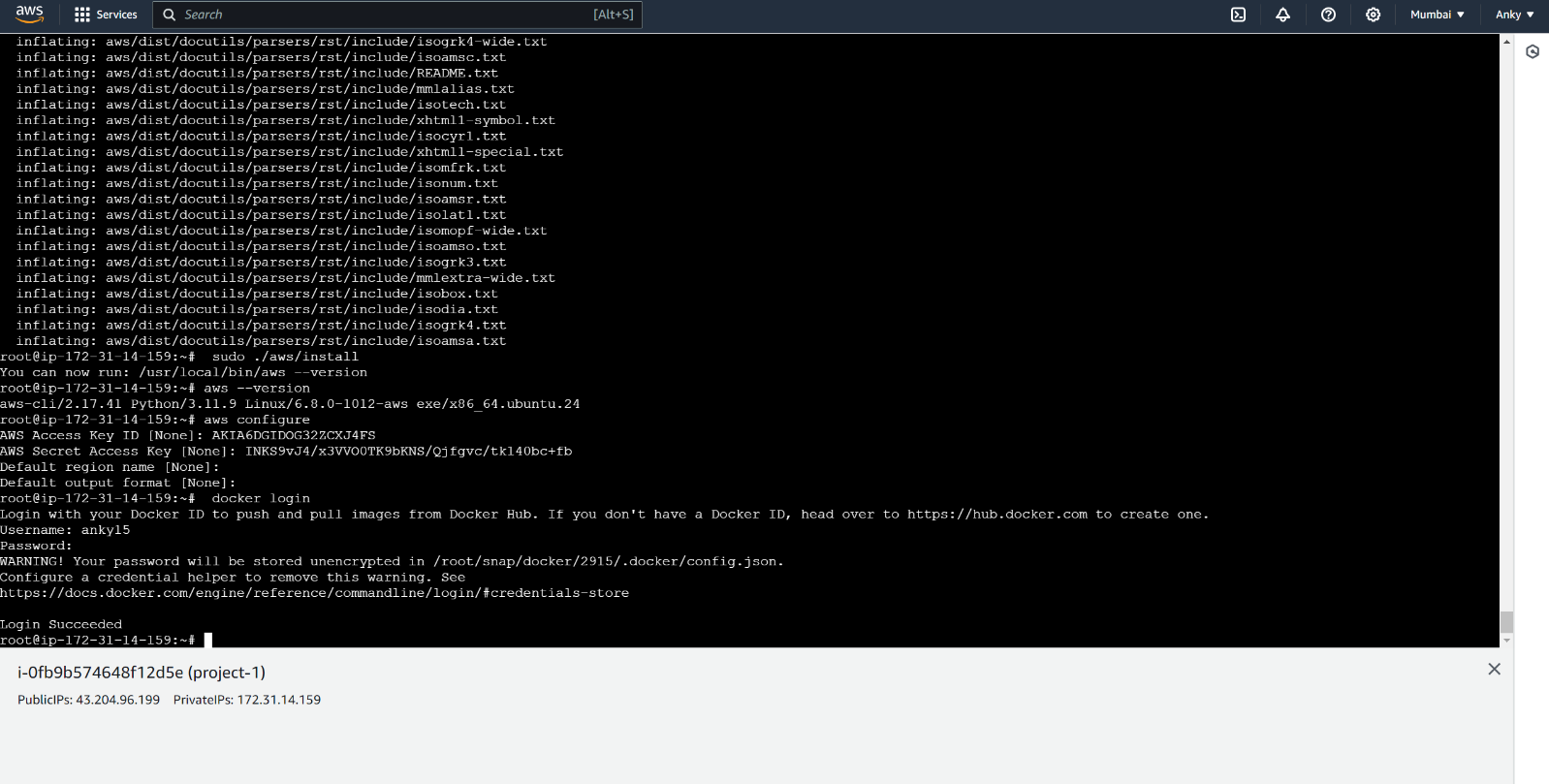
* ***STEP - 1***
* snap install docker
* docker --version
* ***STEP-2***



* docker pull wordpress:latest
* docker images
* ***STEP - 3***

****

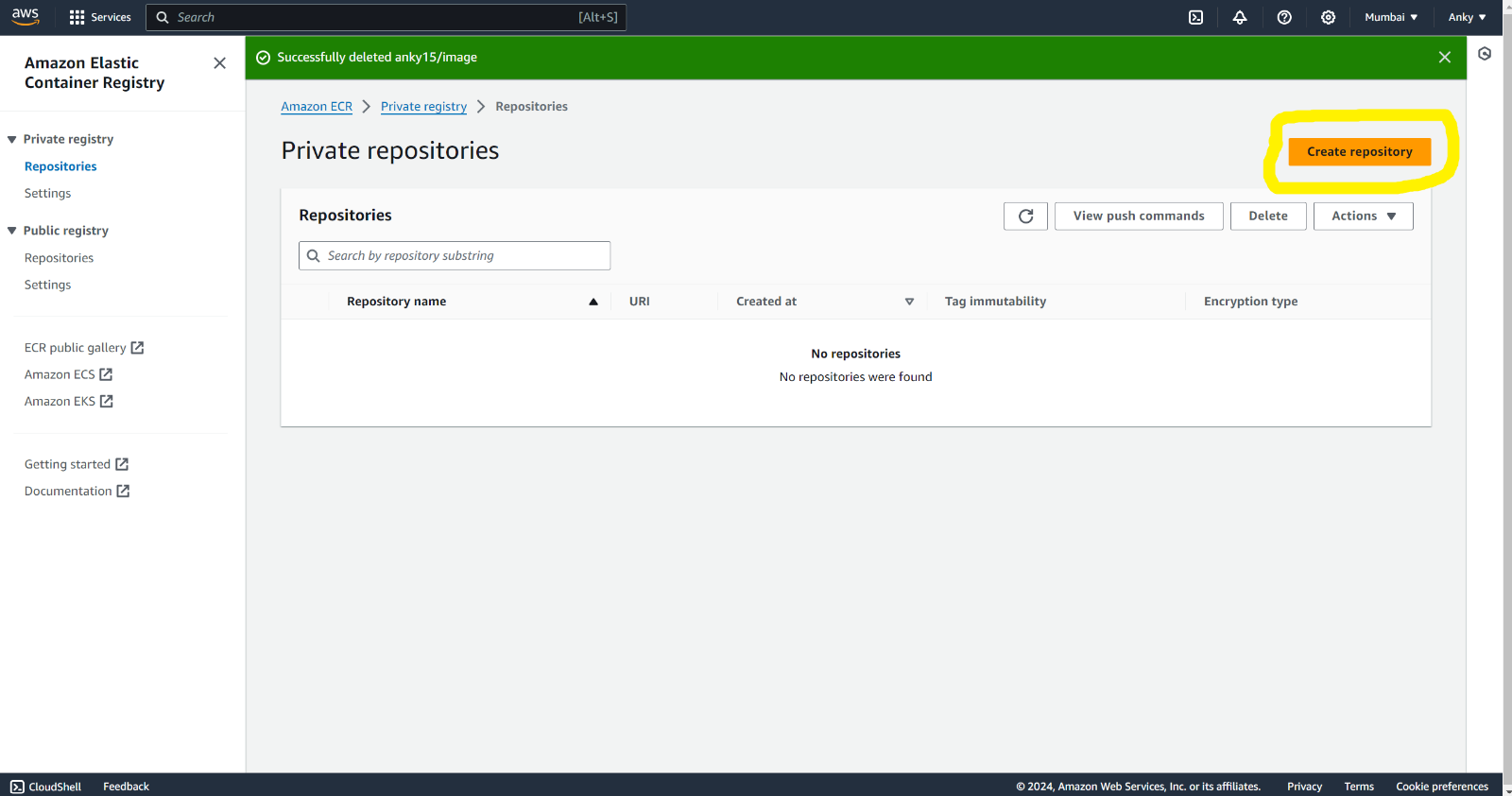
* curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"
* apt install unzip
* ***STEP - 4***



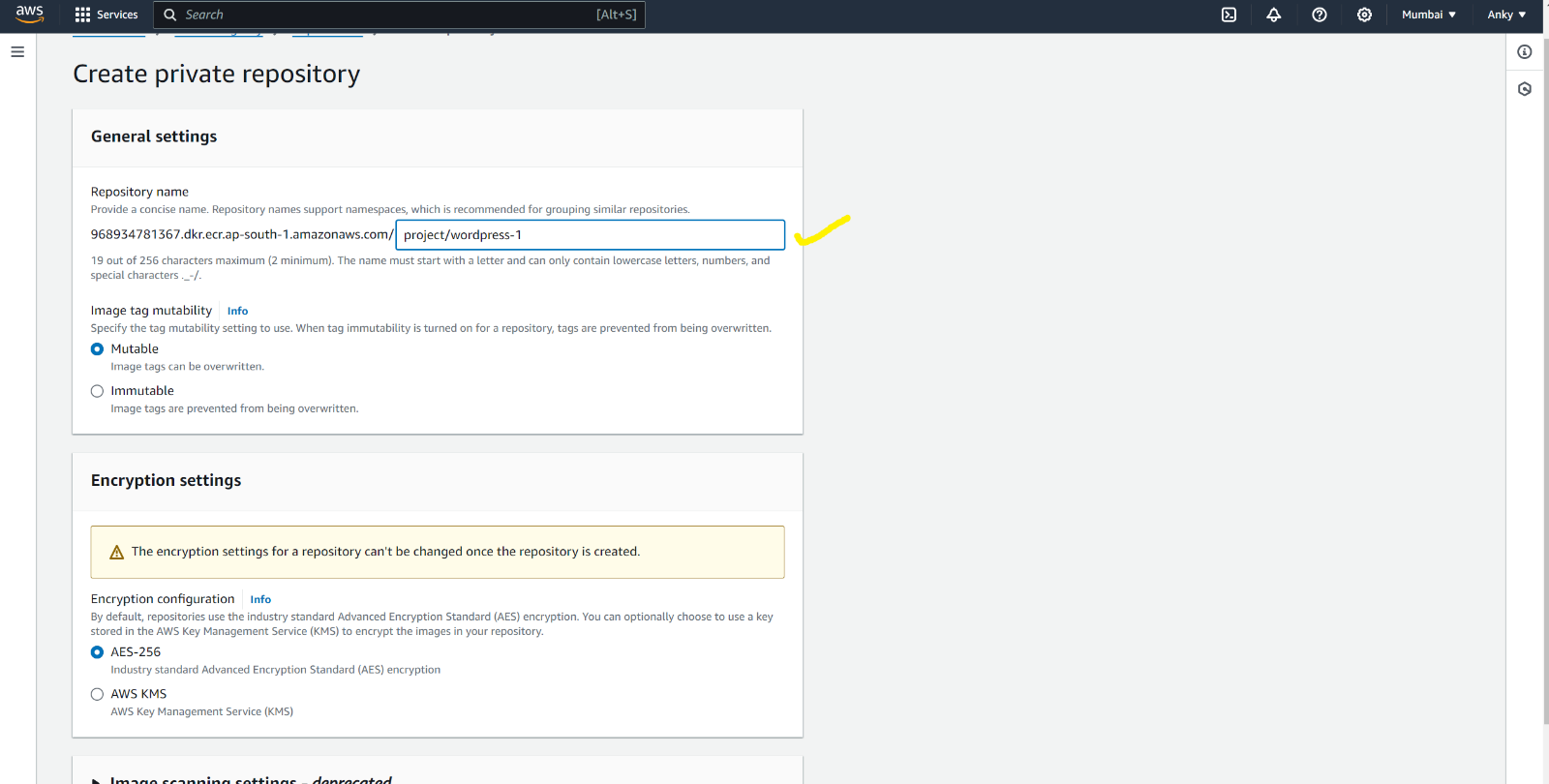
* unzip awscliv2.zip
* sudo ./aws/install
* aws configure – ADD - 1) AWS Access Key ID

2) AWS Secret Access Key

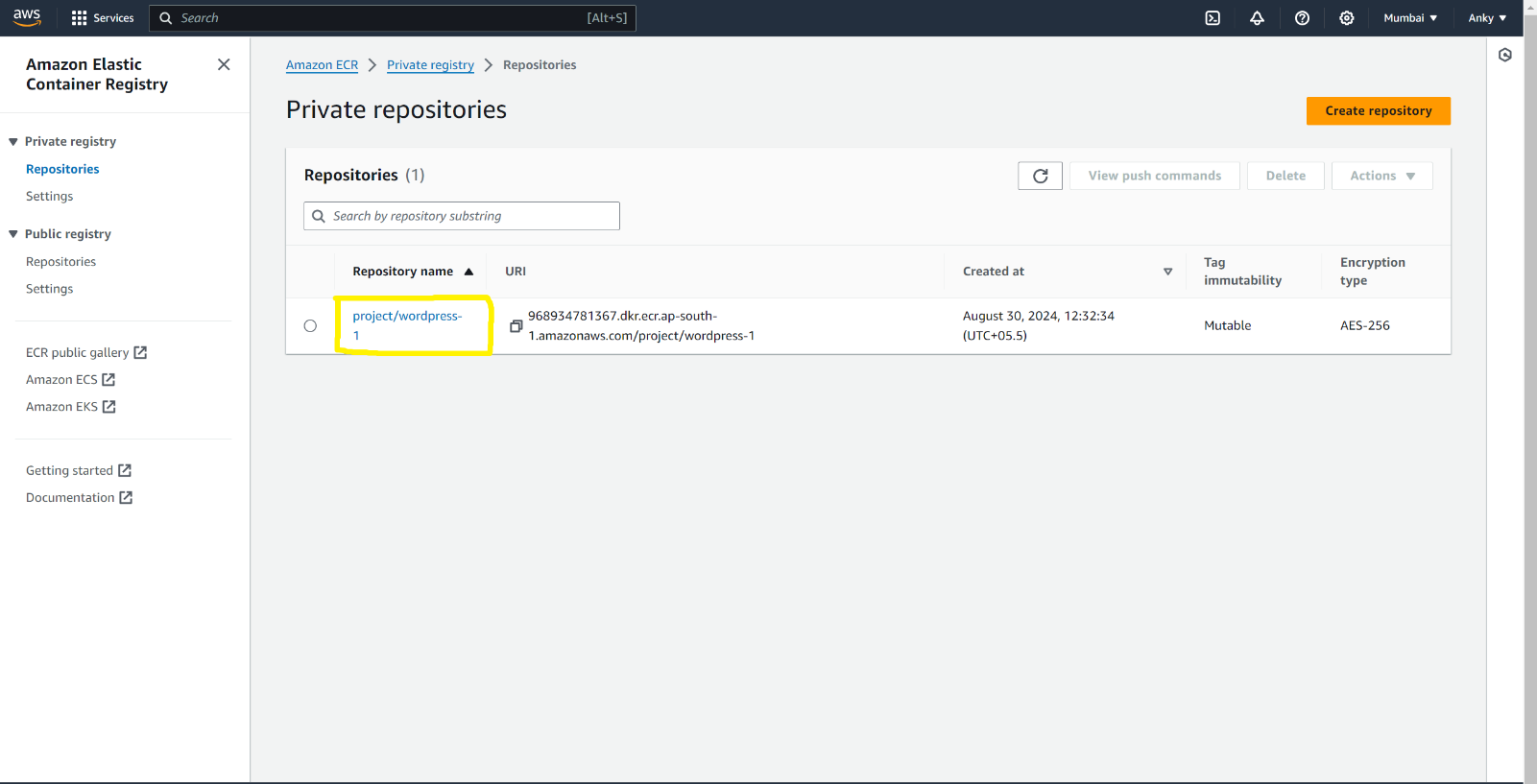
* ***STEP - 5***



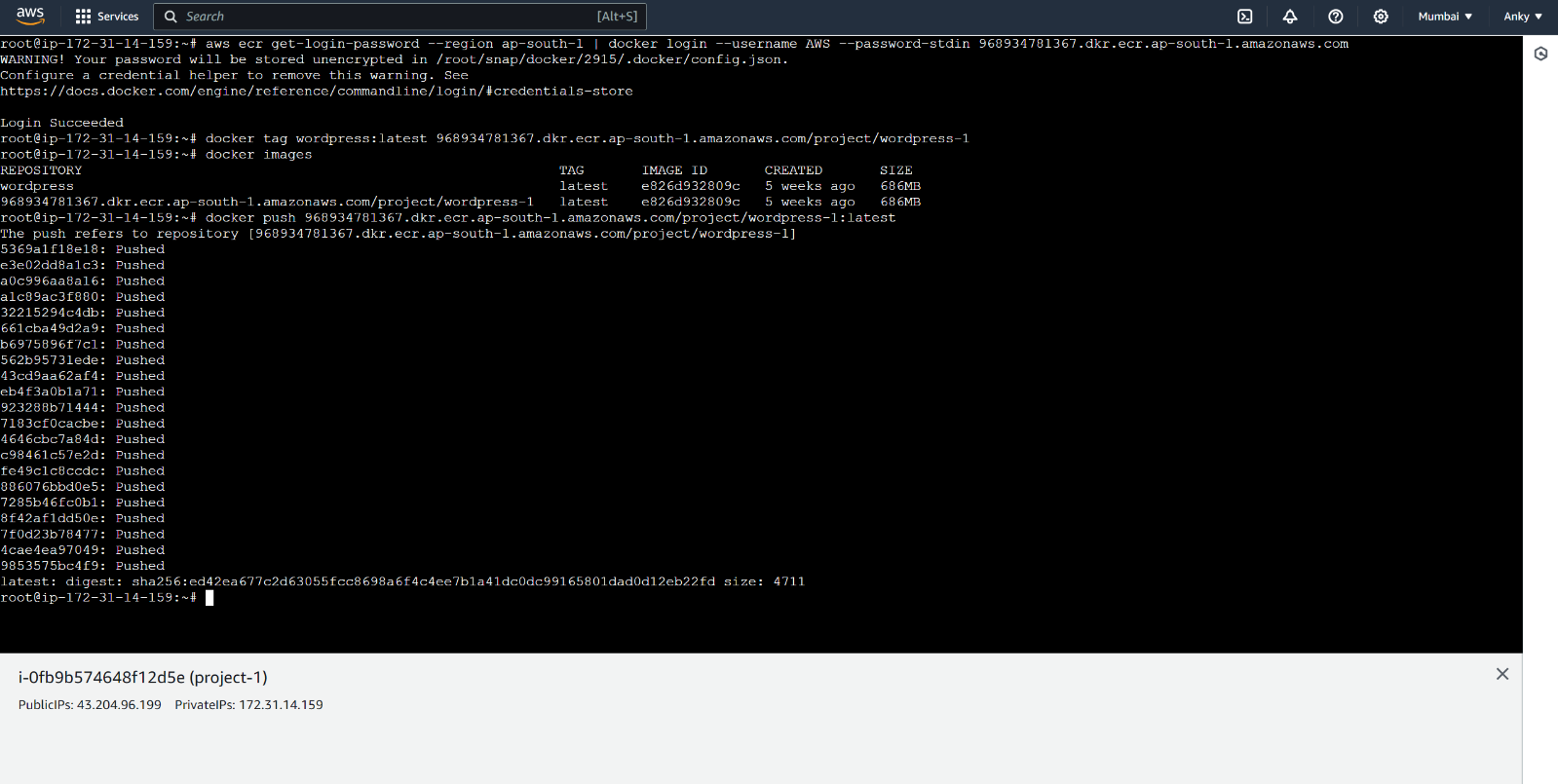
* **Create a Repository**
* ***STEP - 6***



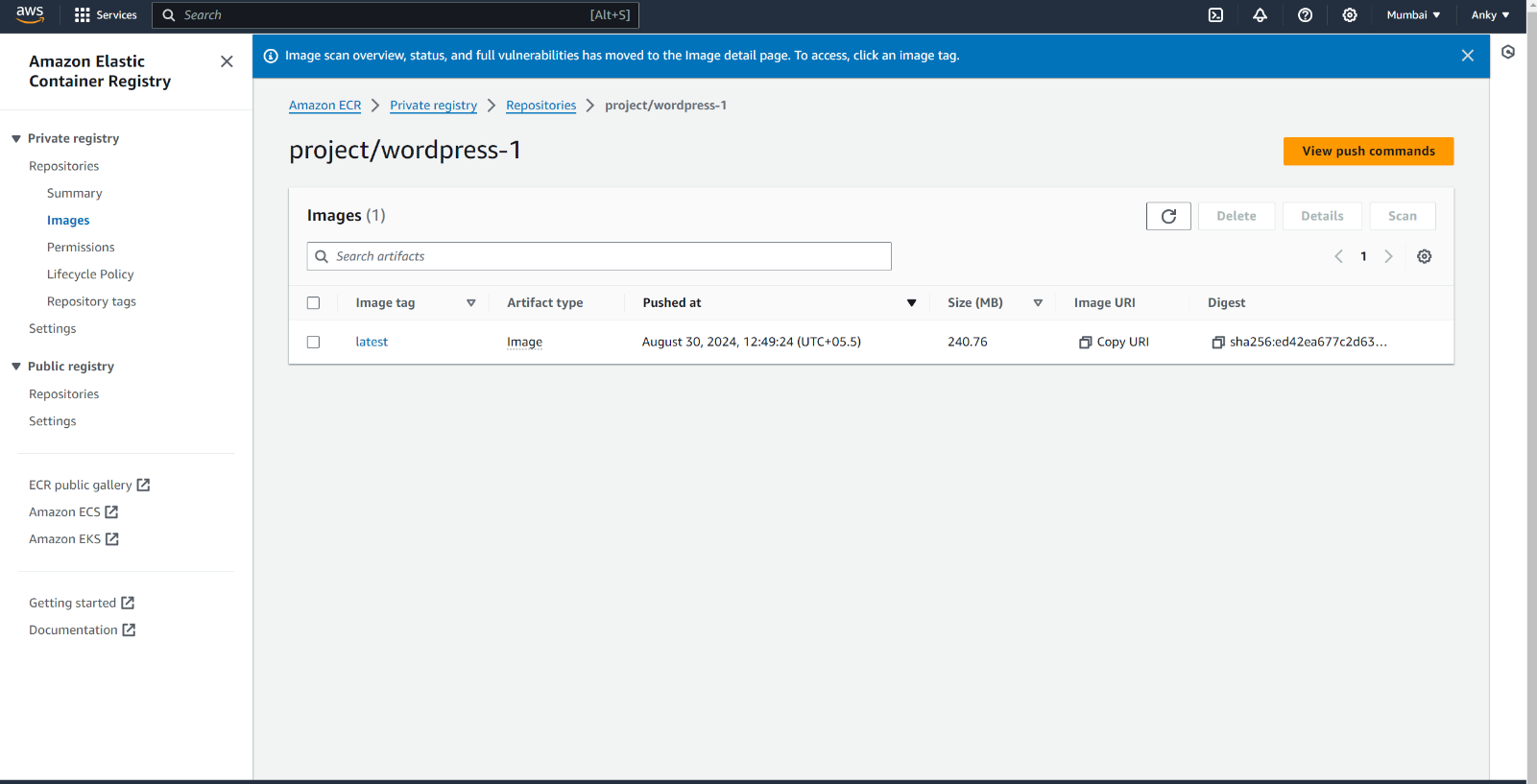
* Mention Repository Name
* Click on – Create
* ***STEP - 7***



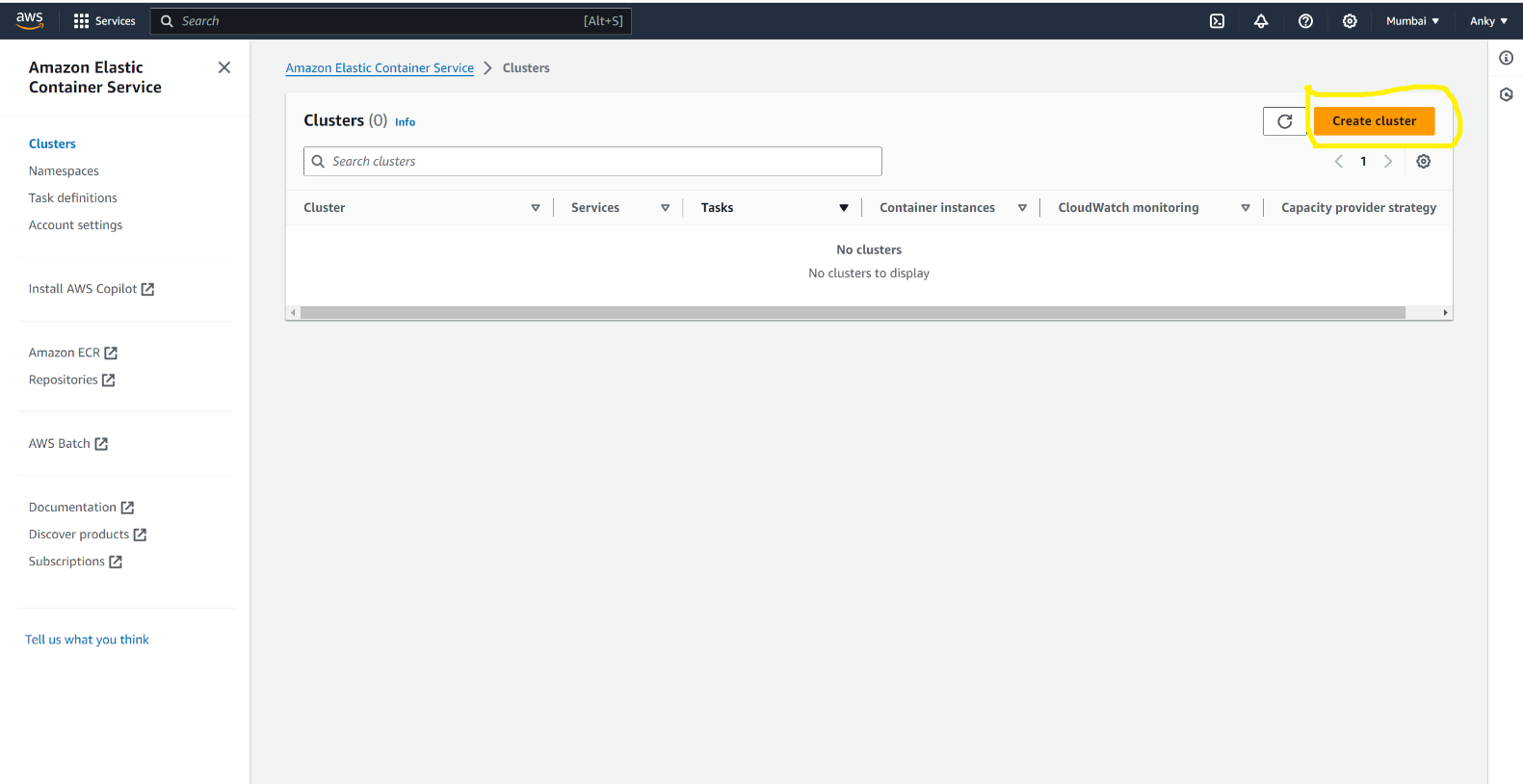
* Click on Repository Name to See the Image
* ***STEP - 8***



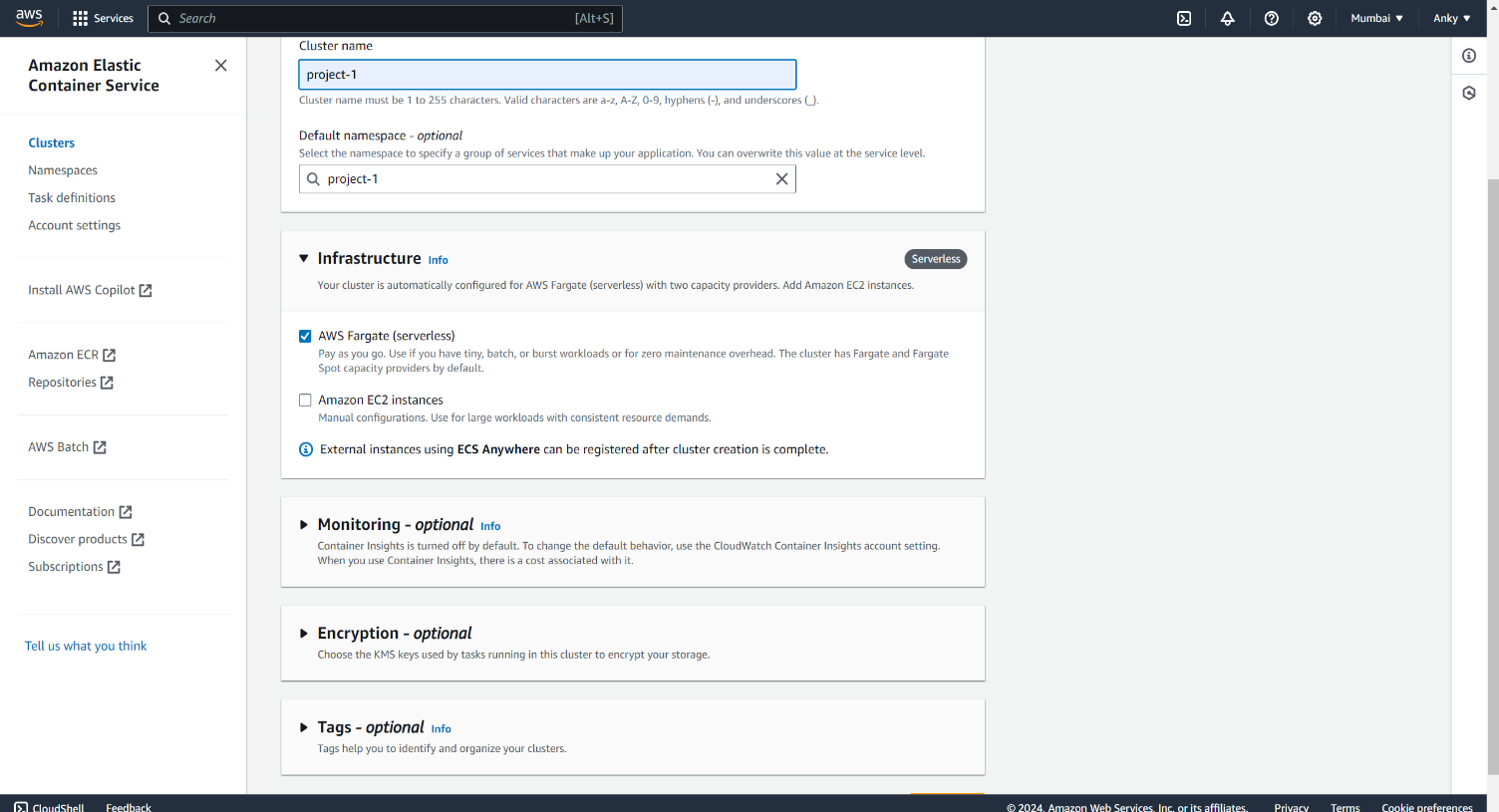
* IMAGE PUSH COMMANDS: -
* aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 968934781367.dkr.ecr.ap-south-1.amazonaws.com
* docker images
* docker tag “image old name:tag” 968934781367.dkr.ecr.ap-south-1.amazonaws.com/image new name:tag (as similar to ECR repository)
* docker push 968934781367.dkr.ecr.ap-south-1.amazonaws.com/ “image new name “:tag
* ***STEP - 9***



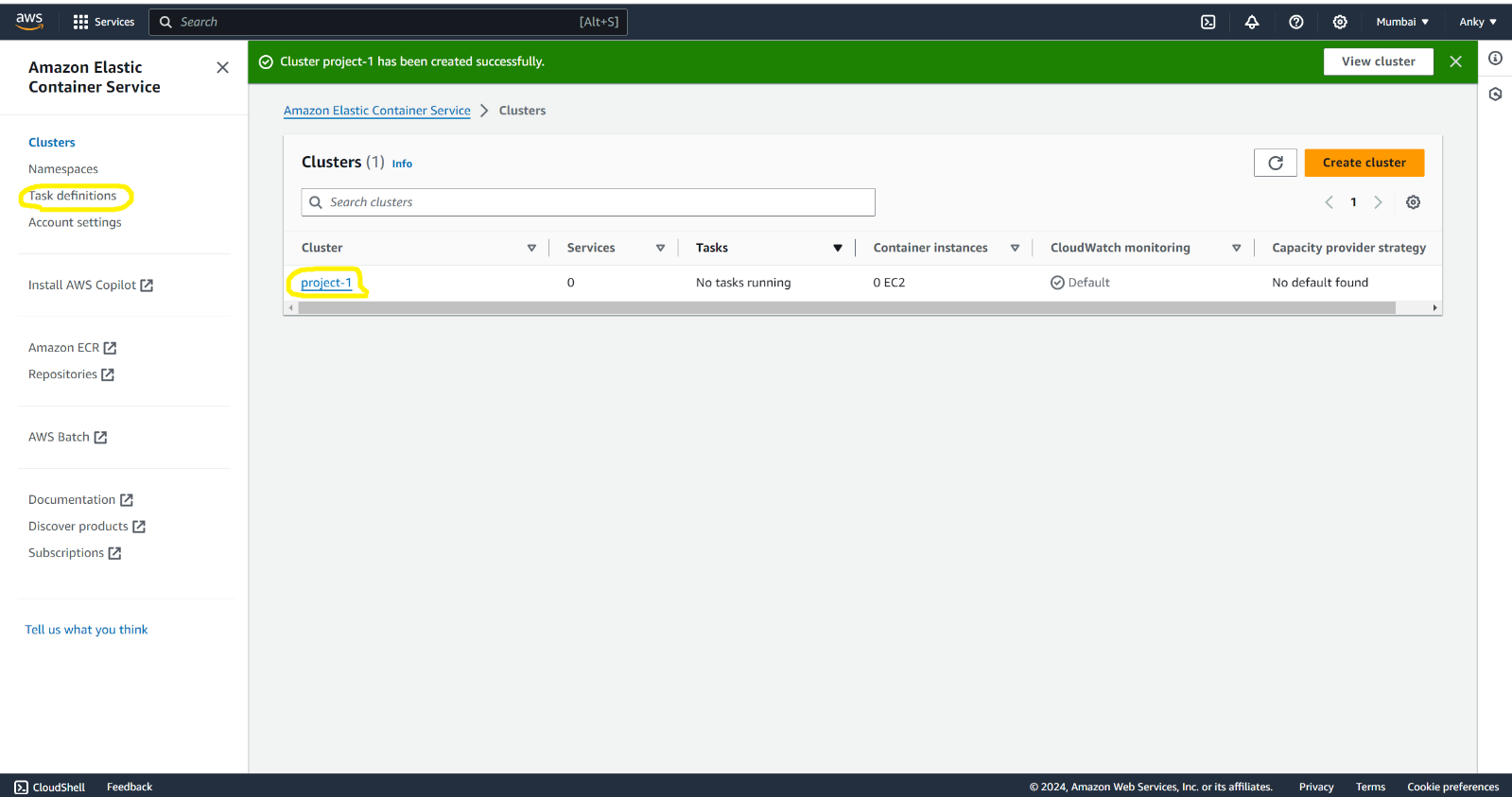
* Image visible in ECR repositroy
* ***STEP - 10***



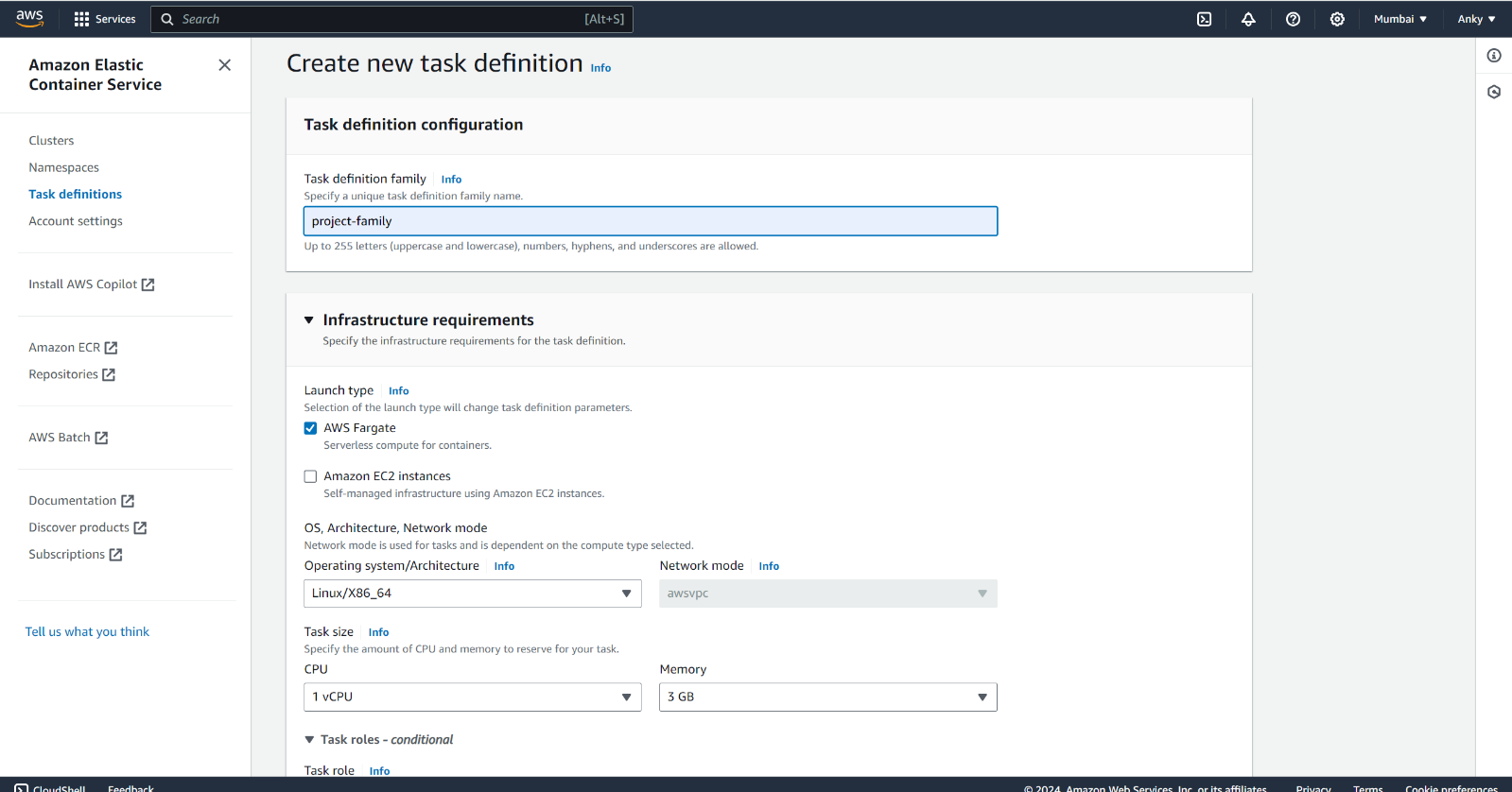
* Create a cluster
* Click on – Create Cluster
* ***STEP – 11***



* Mention Cluster Name
* Click on – Create
* ***STEP - 12***

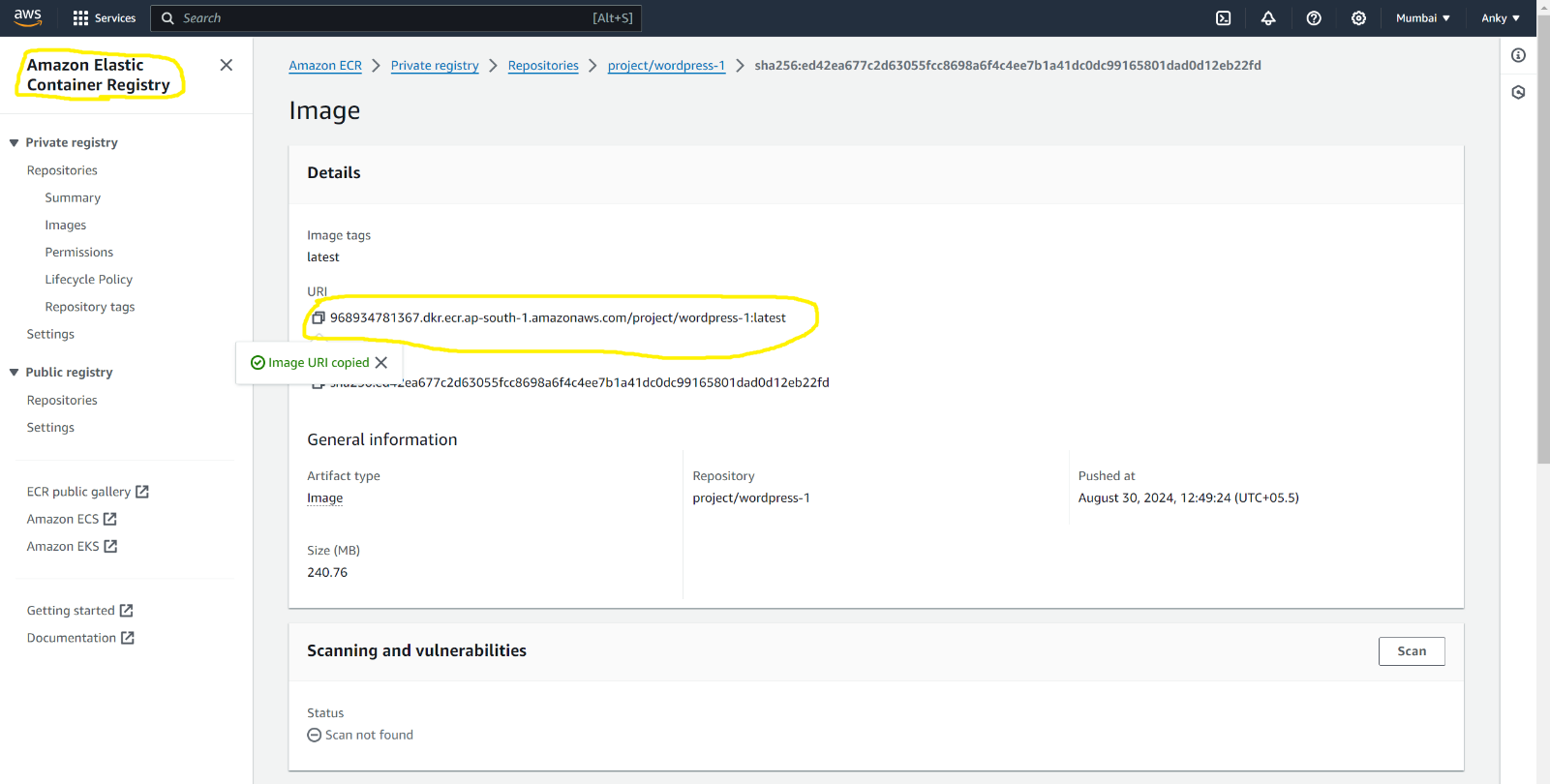


* Create a TASK Definitions
* Click on – Task Definitions
* ***STEP - 13***

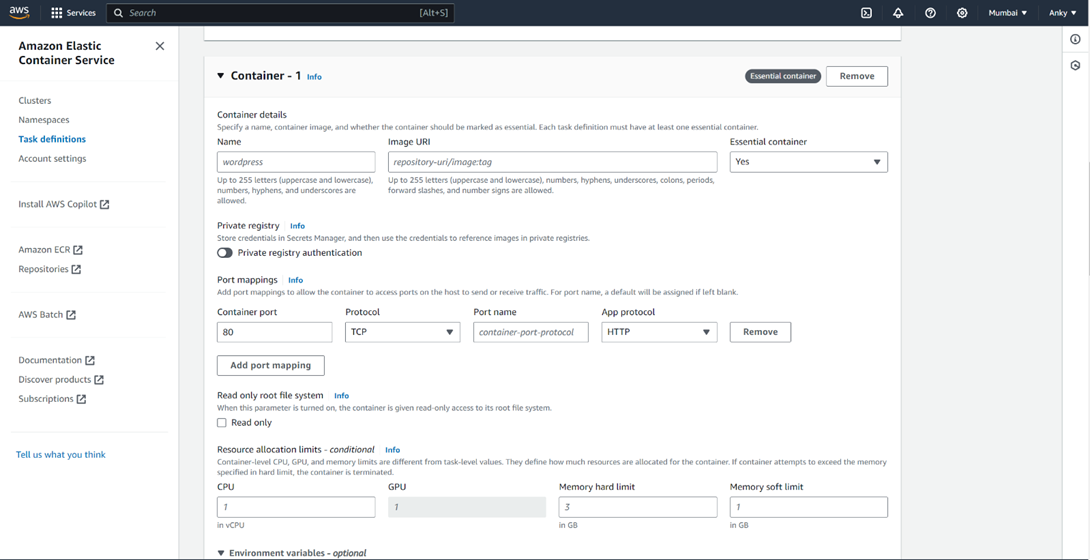


* Specify a Unique Task definition family name
* Select AWS Fargate

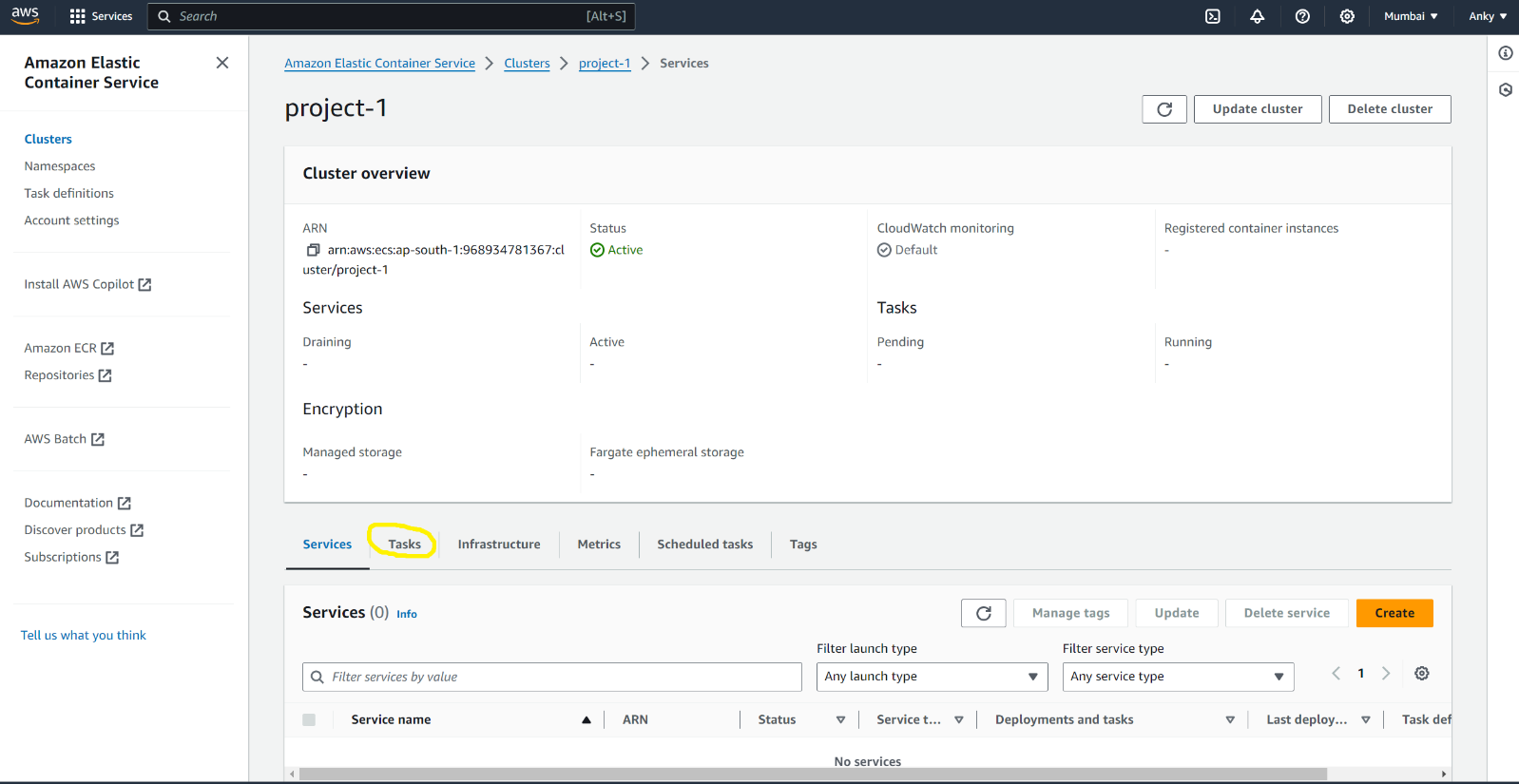
* ***STEP – 14***



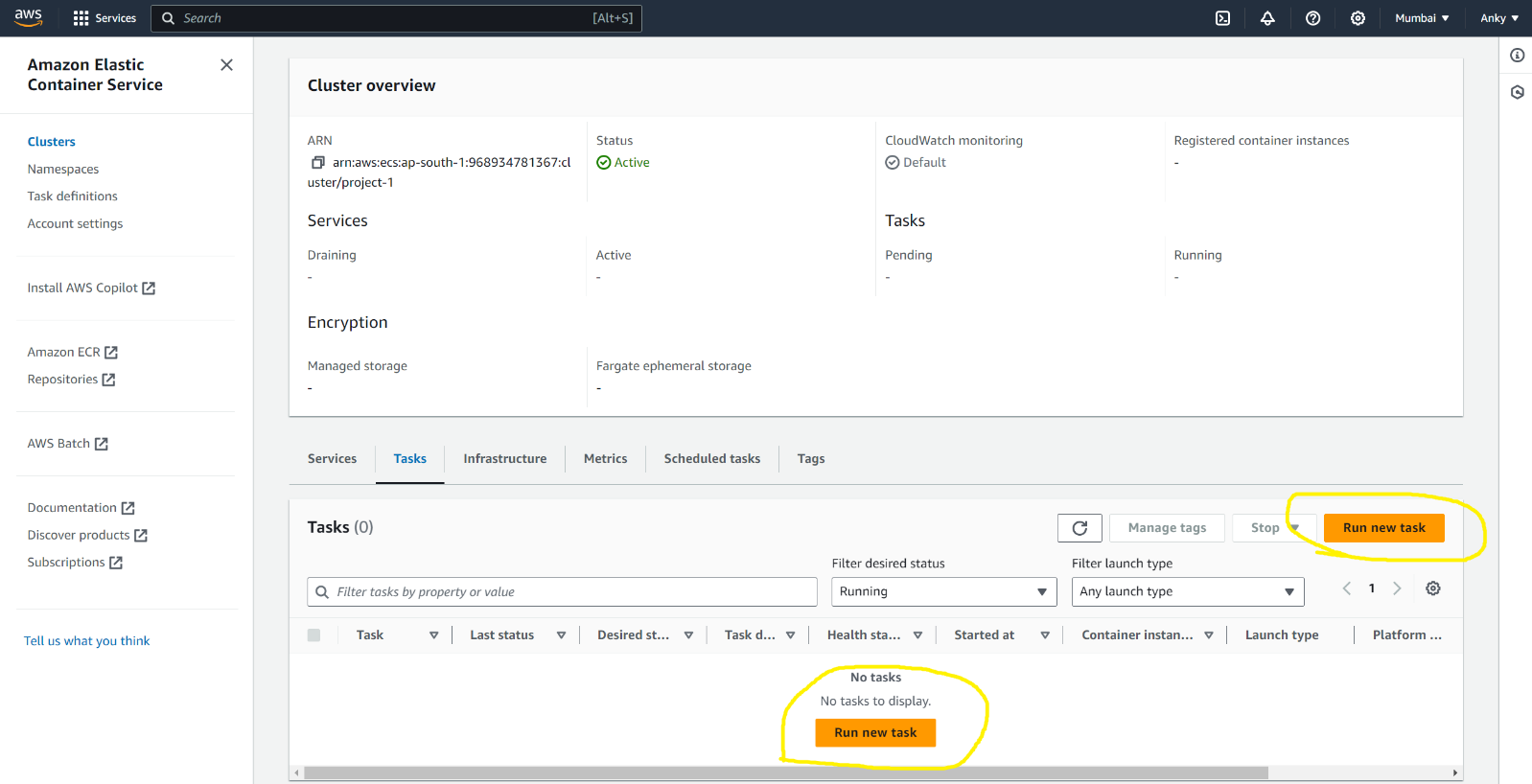
* Go to the ECR – click on Image name
* Copy Image URL
* ***STEP – 15***



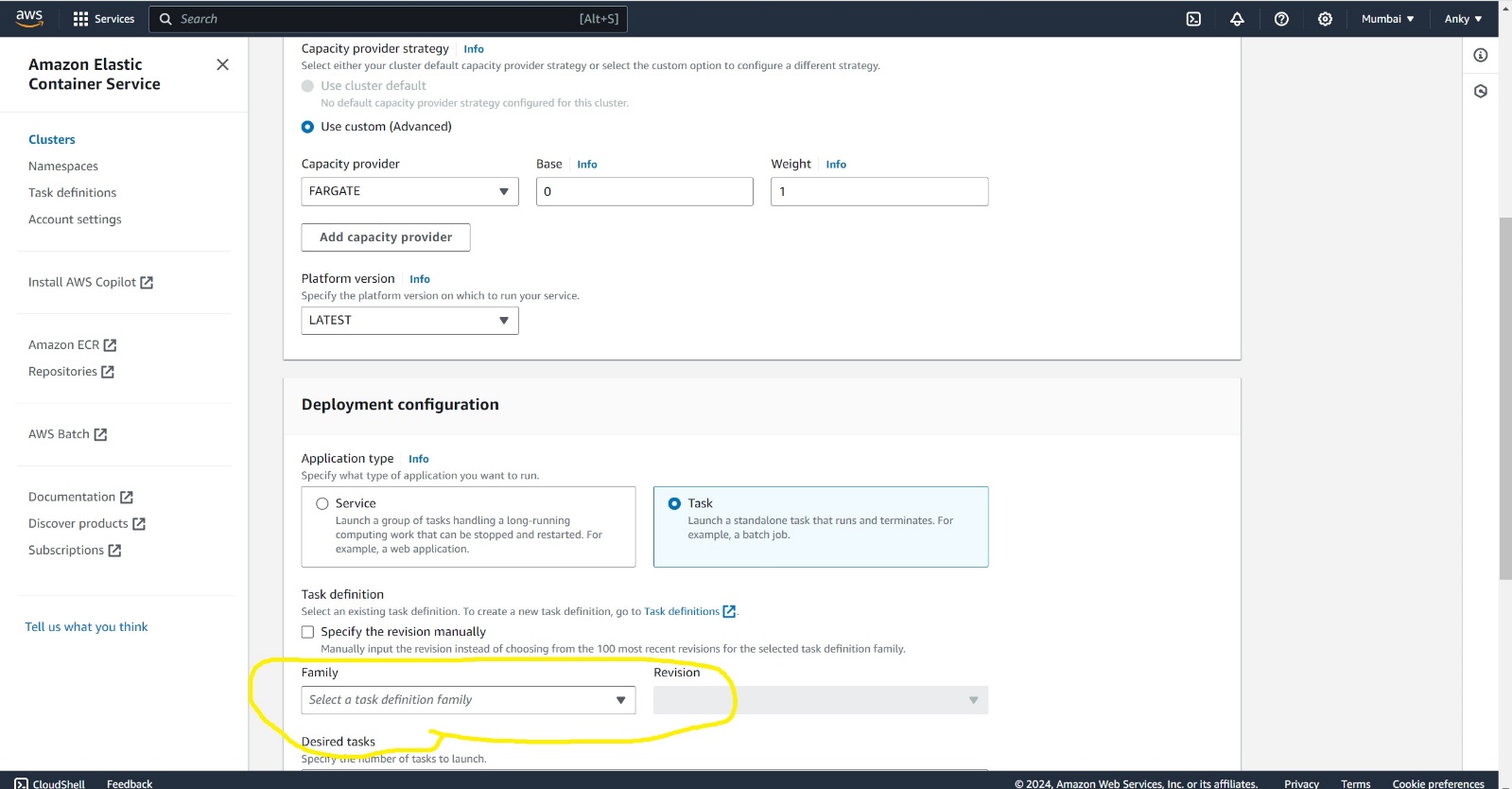
* Mention Container name
* Paste image URL
* Click on – Create
* ***STEP - 16***



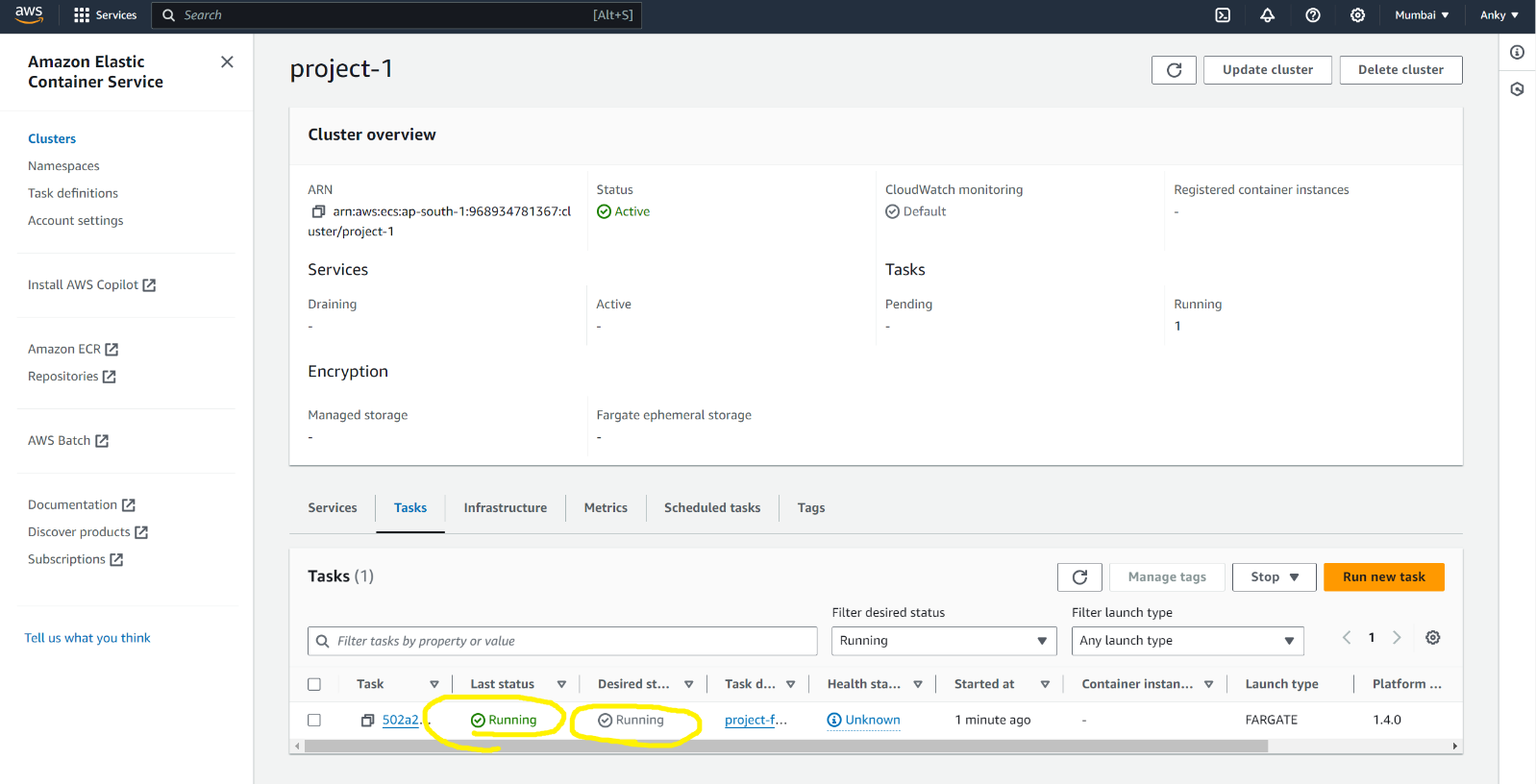
* Clusters >> Cluster overview >> click on Task
* ***STEP - 17***



* Run New Task
* ***STEP - 18***



* Select a Task Definition family
* Go to NETWORKING Section >> check VPC >> SUBNETS >> SECURITY GROUP
* Click On – Create
* ***FINAL STEP -***



* Now task is Running >>> for check
* Click on TASK ID >> Copy Public IP >> Search On Browser
* ***Conclusion****: In this project we successfully deploying WordPress using ECS and Fargate, the project has demonstrated the effectiveness of modern containerized solutions for scalable, secure, and efficient web hosting. The approach ensures high availability and performance while reducing operational complexity.*

ANKIT SHARMA