Project Title:

Containerized Node.js Application Deployment on AWS ECS using Docker ,ECR and ECS.

1. Introduction

This project demonstrates the complete procedure of deploying a Node.js application using Docker containers on AWS ECS. The workflow includes:

- Local development and testing
- Docker containerization
- Push to AWS ECR
- ECS cluster, task, and service configuration
- Health check and verification

The goal is to deploy a fully functional containerized application accessible via a public IP.

2. Project Overview

- **Application:** Node.js with Express framework
- **Container Port:** 8000
- **AWS Services Used:**
    - ECR (Elastic Container Registry)
    - ECS (Elastic Container Service) with Fargate
    - Security Group (port 8000 open)

3. Procedure

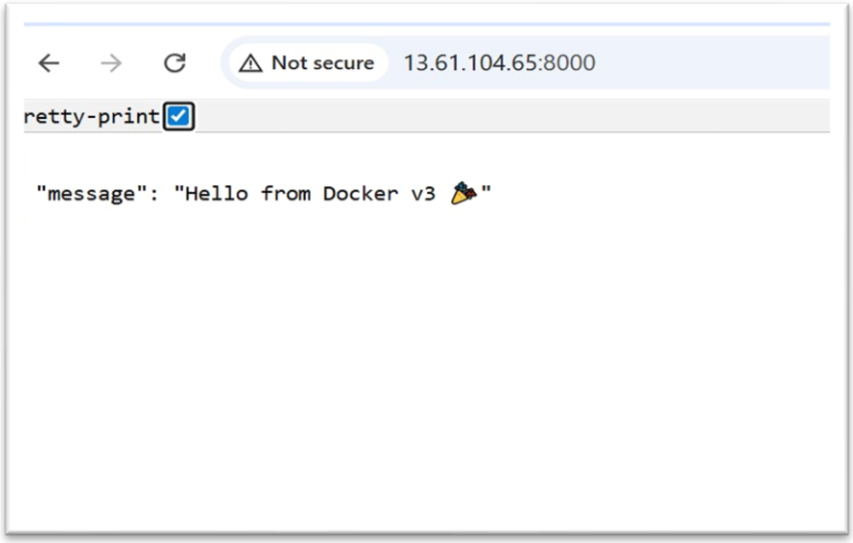
Step 1: Code Setup and Local Testing

- git clone github repo Ali
- cd Ali

Step 2: Docker Image Creation

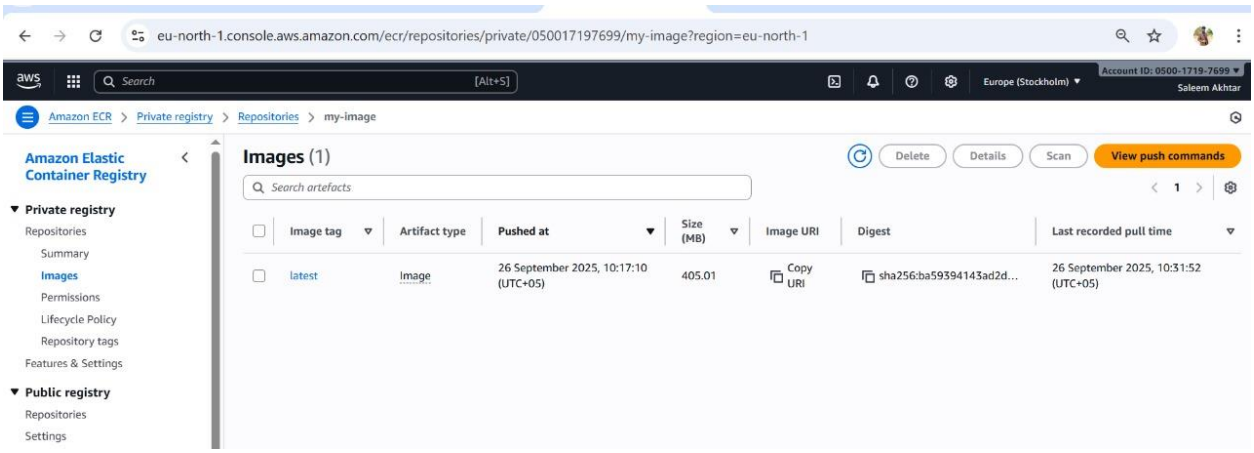
- docker build -t my-image .
- docker run -p 8000:8000 my-image

Step 3:Verify the app via browser



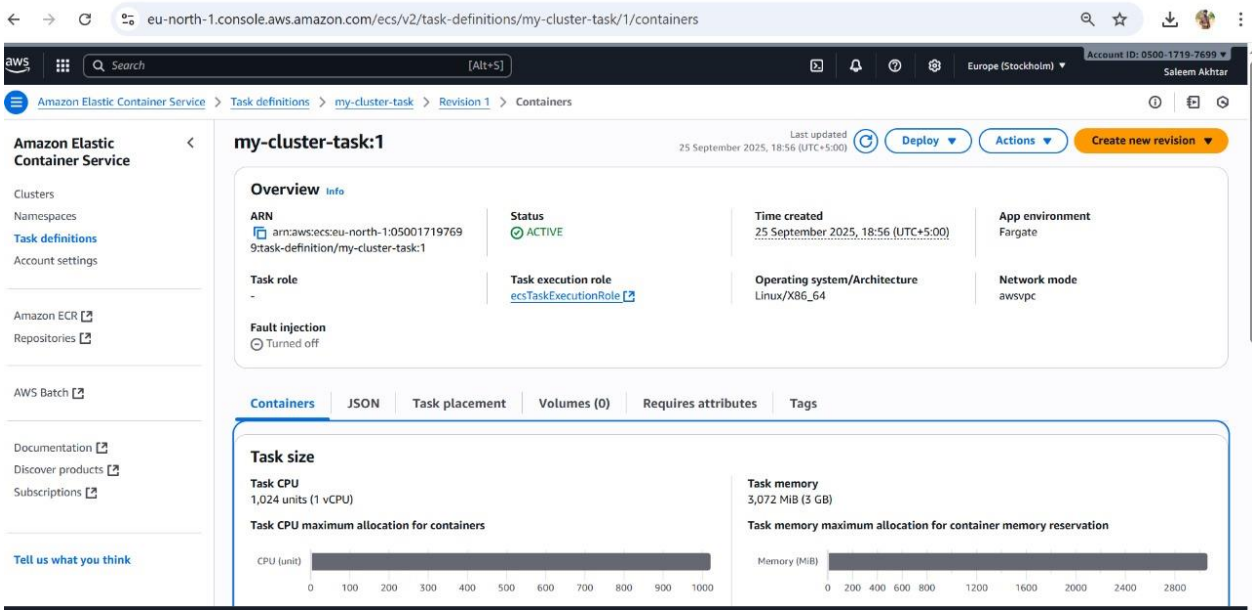
Step 3: Push Docker Image to AWS ECR

- Create an **ECR repository** in AWS Console.
- Go to repo name and view push commands on terminal



Step 4: ECS Cluster Creation

- Create a new cluster in ECS (Fargate, serverless).
- Select **VPC, subnet**, and security group allowing **port 8000**.

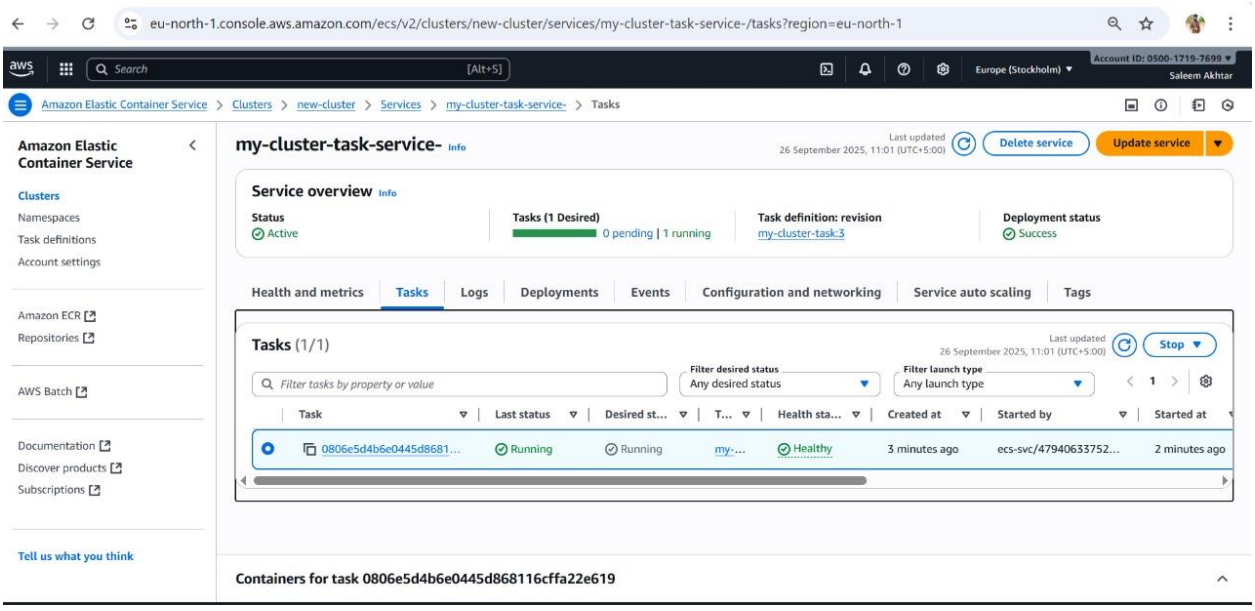


Step 5: Task Definition

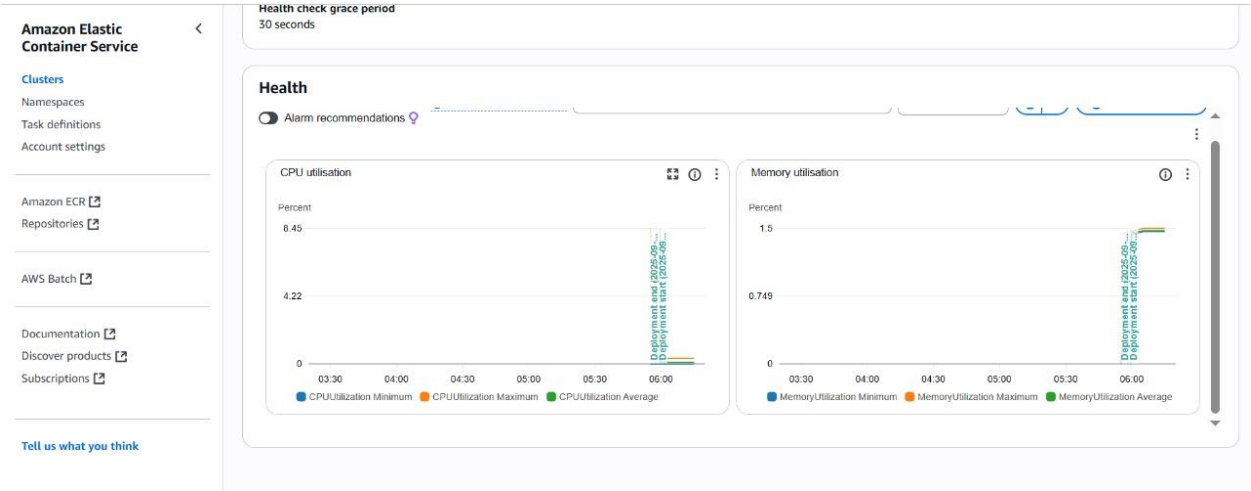
- Define a new task using the ECR image.
- Set **container port 8000**.
- Enable **Health Check:**

Step 6: Service Creation

- Create an ECS service using the task definition.
- Set desired tasks = 1, launch type = FARGATE, platform version = LATEST.
- Assign public IP for external access.
- Wait for task status = HEALTHY.

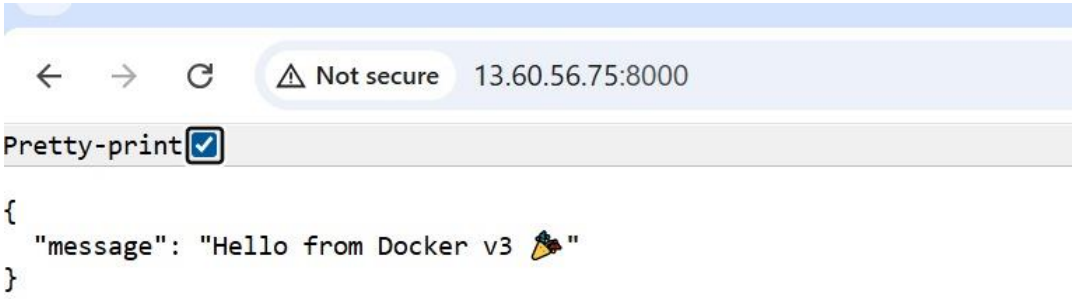


ECS Cluster Health.



Step 7: Verification

- Access the app using **task public IP** on port 8000.
- Confirm ECS service **desired tasks = running**, and container health is **HEALTHY**.



5.Conclusion

This project demonstrates the complete end-to-end deployment of a Node.js application using Docker containers on AWS ECS. It highlights the process of containerizing an application, pushing Docker images to Amazon ECR, creating an ECS cluster along with task definitions and services, configuring health checks and security groups, and successfully deploying the application for external access. The workflow presented in this project serves as a reusable blueprint for deploying other containerized applications on AWS.