ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY

Approved by AICTE, Permanently affiliated to JNTUK & Accredited by NAAC
Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956 Aditya Nagar, ADB Road –Surampalem 533437, E.G. Dist., A.P.

Presentation on Internship

Deploying a Web Application on Amazon ECS

Submitted by

NAME: R.MADHAVI
ROLL-NUMBER:23MH1A4453
SECTION:III DS-C

Internship Coordinator

Mrs. K. Roja Rani

Department of CSE(DS & IOT)

Deploying a web applications on Amazon ECS

Prepared in the partial fulfillment of Summer Internship Program on AWS

AT

Under the guidance of

Mrs. Sumana Bethala, APSSDC

Mr. Anil Kumar, APSSDC

Deploying a Web Application on Amazon ECS

- Amazon EC2 (Elastic Compute Cloud)
- Amazon ECR (Elastic Container Registry)
- Amazon ECS (Elastic Container Service)
- AWS Fargate (within ECS)
- Elastic Load Balancing (Application Load Balancer)
- Amazon VPC (Virtual Private Cloud)
- AWS IAM (Identity and Access Management)

Problem Statement

- Modern web applications require reliable, scalable, and highly available infrastructure.
- Manual provisioning and deployment are error-prone and time-consuming.

Challenges:

- - Managing compute resources dynamically
- - Packaging and containerizing applications correctly
- - Scaling with demand
- - Simplifying deployment workflows
- - Monitoring and maintaining high availability

Goal:

- Deploy a web application in a containerized form on AWS, using services like ECS, EC2, ECR, Fargate, and ALB, so that the application is scalable, reliable, and easy to manage.

Solution Overview

- Implemented a fully containerized web application hosted on Amazon ECS.

Architecture Highlights:

- Docker to containerize the app
- Amazon ECR is to Securely stored container images
- Amazon ECS to Managed and orchestrated containers
- EC2 & Fargate to Ran containers on managed and serverless compute
- Application Load Balancer for routing
- IAM Roles, VPC, Security Groups ensure secure access and networking

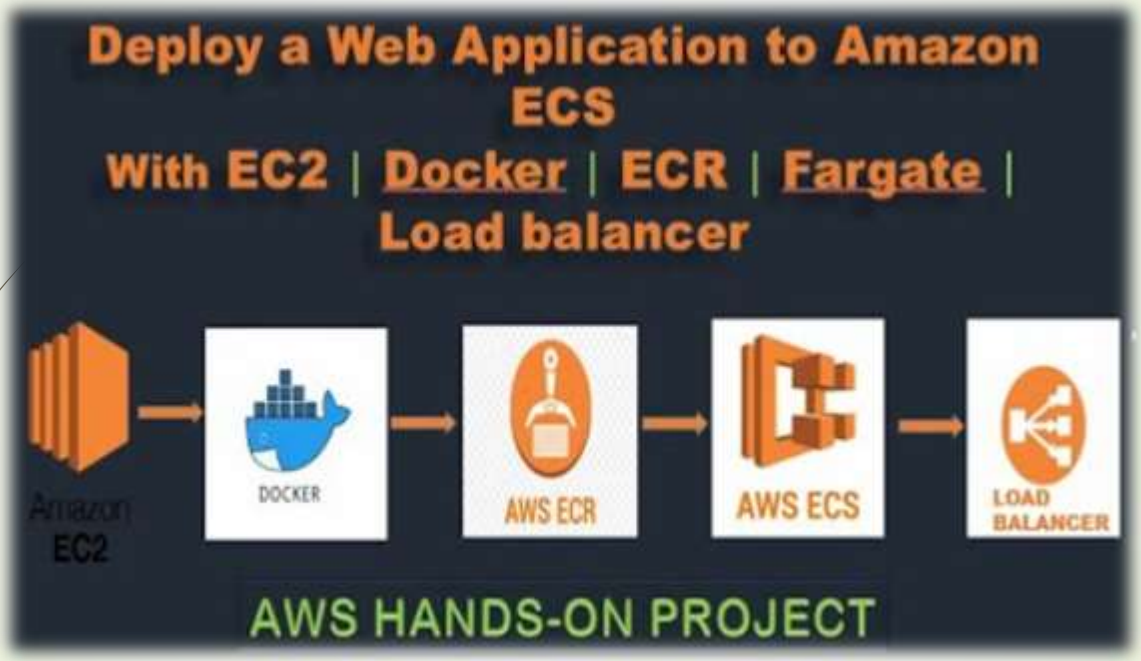
Key Features

1. Containerization: Portable and reproducible builds using Docker
2. Scalability: ECS with Fargate scales based on CPU/memory
3. High Availability: Load balancer distributes traffic
4. Security: IAM, VPC, and security group controls
5. CI/CD Friendly: Easily pluggable with Code Pipeline/Code Build
6. Cost-Effective: Pay for compute only when used (Fargate)

Tools & Technologies

- AWS ECS (EC2 + Fargate)
- Docker
- Amazon ECR
- Application Load Balancer
- IAM, Security Groups, VPC
- EC2 Instances

Architecture Diagram



1. Amazon EC2 (Elastic Compute Cloud)

•**Purpose**: Acts as a development/CI environment where you create and build your Docker image.

•**What Happens Here**:

- Launch an EC2 instance (Linux/Ubuntu).
- Install Docker on EC2.
- Write your application code and a Dockerfile.
- Build the Docker image locally on EC2.

2. Docker

•**Purpose**: Containerizes your web application for easy deployment.

•**What Happens Here**:

- Use Docker to build an image from the application code.
- Tag the Docker image for upload to ECR.

3. AWS ECR (Elastic Container Registry)

•**Purpose**: A fully managed Docker container registry.

•**What Happens Here**:

- Push the Docker image from EC2 to ECR.
- ECR stores and version controls your image for ECS deployments.

4. AWS ECS (Elastic Container Service)

•**Purpose**: Orchestrates the deployment of containerized applications.

•**What Happens Here**:

- Create a **Task Definition** using the image from ECR.
- Choose **Fargate** as the launch type (serverless deployment).
- ECS runs your container with the required CPU, memory, and networking settings.

5. Fargate (Launch Type)

•**Purpose**: Serverless compute engine for containers.

•**What Happens Here**:

- You don’t manage servers — Fargate provisions infrastructure automatically.
- ECS service ensures your application stays running and scalable.

6. Load Balancer (Elastic Load Balancer - ELB)

•**Purpose**: Distributes incoming traffic to ECS tasks.

•**What Happens Here**:

- Set up an Application Load Balancer (ALB).
- ALB routes traffic to your running containers behind the scenes.
- Ensures high availability and fault tolerance.

Learnings and Outcomes

- Understood ECS launch types: EC2 vs Fargate
- Hands-on with Docker, ECR, and ALB configuration
- Importance of IAM roles and networking setup
- Troubleshooting deployment issues
- Learned how to automate multi-step cloud deployments

Conclusion & Next Steps

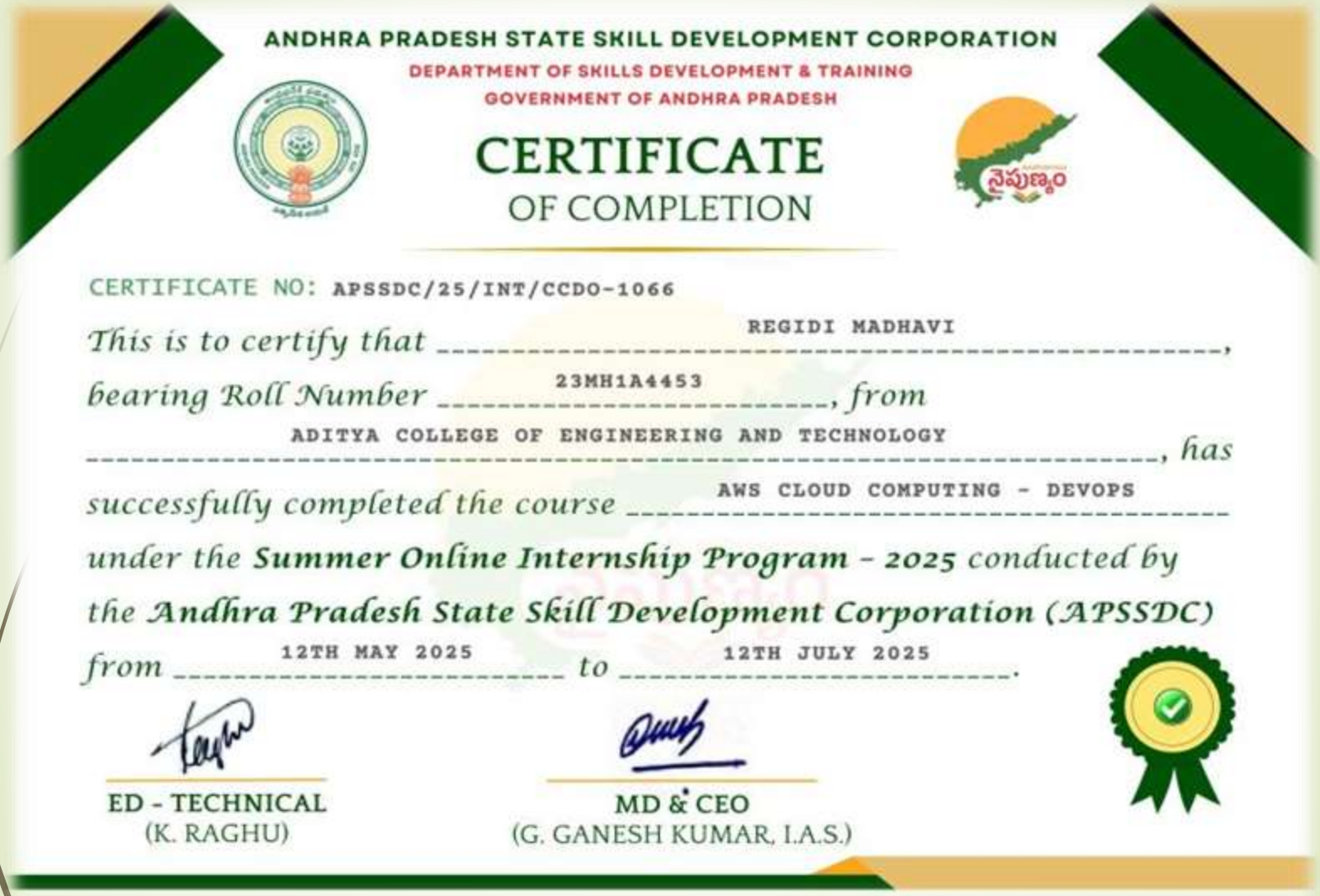
Conclusion:

- Successfully deployed a production-grade web application on AWS ECS with full container orchestration and traffic management.

Next Steps:

- Automate using Code Pipeline
- Implement HTTPS with ACM
- Explore service discovery and auto-scaling

INTERNSHIP COMPLETION CERTIFICATE



ANDHRA PRADESH STATE SKILL DEVELOPMENT CORPORATION

DEPARTMENT OF SKILLS DEVELOPMENT & TRAINING

GOVERNMENT OF ANDHRA PRADESH

CERTIFICATE

OF COMPLETION

CERTIFICATE NO: APSSDC/25/INT/CCDO-1066

This is to certify that REGIDI MADHAVI,

bearing Roll Number 23MH1A4453, *from*

ADITYA COLLEGE OF ENGINEERING AND TECHNOLOGY, *has*

successfully completed the course AWS CLOUD COMPUTING - DEVOPS

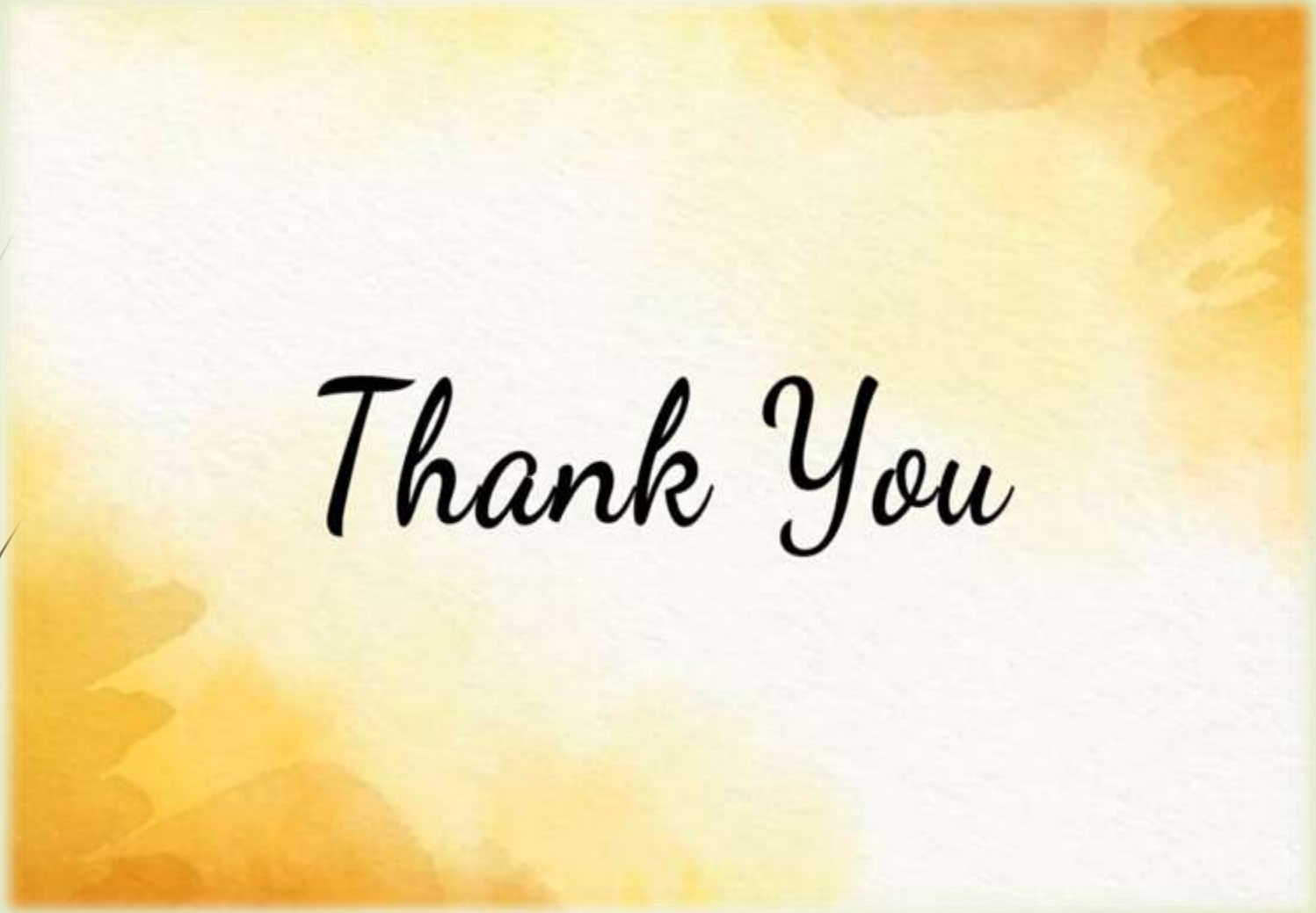
under the ***Summer Online Internship Program - 2025*** *conducted by*

the ***Andhra Pradesh State Skill Development Corporation (APSSDC)***

from 12TH MAY 2025 *to* 12TH JULY 2025.

ED - TECHNICAL
(K. RAGHU)

MD & CEO
(G. GANESH KUMAR, I.A.S.)


Thank You