



🚀 Project Mission Deploy a Scalable PHP Web Application Using Amazon ECS

- 1 To deploy a PHP-based web application that simulates a production-ready workload
- 2 The application will run on Amazon ECS using both Fargate and EC2 launch types
- 3 It will be exposed to the internet through an Application Load Balancer (ALB)
- 4 The app will support image uploads, which will be stored in Amazon S3
- 5 The container image for the app will be built using Docker and stored in Amazon ECR
- 6 The deployment process will follow AWS best practices, including proper IAM roles and environment configuration
- 7 This project reflects a real-world cloud deployment scenario, similar to what is used in modern production environments

🎯 Objective

- 1 Containerize a web application
- 2 Push Docker images to ECR
- 3 Run tasks and services in ECS
- 4 Attach load balancers
- 5 Scale the application
- 6 Handle network and IAM configurations

🎯 Goal

- 1 Create a single ECS cluster that supports both Fargate (serverless containers) and EC2 , so we can
- 2 🛠 Practice and compare both launch types
- 3 📦 Unlock all ECS Task Definition configuration options
- 4 🚀 Deploy App to either Fargate or EC2

1 Go to the ECS Console Click "Clusters" → "Create Cluster"

2 ⚙ Section 1: Cluster Configuration Cluster name cloudfolks-ecs-cluster

1 Option 1: AWS Fargate (Serverless) ⚙ Just keep Fargate enabled (checked). No other config needed

1 Auto Scaling group (ASG) Create new ASG
2 Provisioning model On-demand
3 Container instance Amazon Machine Image (AMI) Use default: Amazon ECS-Optimized Amazon Linux 2023

3 🏗 Section 2: Infrastructure

2 Option 2: Amazon EC2 Instances

4 Instance type t2.micro (Free tier)
5 EC2 instance role Create new role
6 Desired capacity 1 Minimum 1
2 Maximum 2
7 SSH Key pair Select your existing key or Create new
8 Root EBS volume size 30 GB

🛠 Implementation Plan

1 ✅ Step 1: Create ECS Cluster

📋 Steps

4 Create

1 VPC Choose the default VPC already selected

1 Just keep the default selected subnet as-is
2 ☒ No need to create or modify subnets

2 Subnets

1 If you're deploying in Mumbai (ap-south-1) region
2 ⚙ Sometimes, t2.micro (free tier) instances are not available in ap-south-1c
3 ☒ To avoid instance launch failures in the future Deselect the ap-south-1c subnet

3 🌐 Section 3: Network Settings for Amazon EC2 Instances

1 Select Create a new security group
2 Name cloudfolks-ecs-sg

3 Inbound Rules to Add 1 Type SSH From 0.0.0.0/0
2 Custom TCP 8080 From 0.0.0.0

4 Auto-assign public IP Use subnet Setting