GitHub Actions + AWS ECS Deployment

Step 1: Prerequisites

- ● An AWS account with ECS and IAM access
- ● A GitHub repository with your application
- ● AWS CLI installed and configured (aws configure)
- ● Docker installed for building and pushing images
- ● An ECS Cluster and ECS Service set up

Step 2: Create AWS Resources

Create an ECS Cluster aws ecs create-cluster --cluster-name my-cluster

- Create an ECS Task Definition (Fargate example)
  - Define a task-definition.json file with the container details.

Json

```
{
 "family": "my-task",
"networkMode": "awsvpc",
"containerDefinitions": [
{
 "name": "my-container",
"image": ".dkr.ecr..amazonaws.com/my-repo:lat est",
 "memory": 512,
 "cpu": 256,
"essential": true
}
]
 }
```

2. Register the task:
   - aws ecs register-task-definition --cli-input-json file://task-definition.json

3. Create an ECS Service

- aws ecs create-service --cluster my-cluster --service-name my-service --task-definition my-task --desired-count 1 --launch-type FARGATE

Step 3: Set Up GitHub Secrets

- Go to your GitHub repository → Settings → Secrets and variables → Actions → New repository secret, and add:
- AWS_ACCESS_KEY_ID
- AWS_SECRET_ACCESS_KEY
- AWS_REGION
- ECR_REPOSITORY_NAME
- ECS_CLUSTER
- ECS_SERVICE
- TASK_DEFINITION

Step 4: Configure GitHub Actions Workflow

Create .github/workflows/deploy.yml in your repo:

yaml

```
name: Deploy to AWS ECS
 on:
   push:
     branches:
          - main
jobs:
   deploy:
       runs-on: ubuntu-latest
   steps:
 - name: Checkout Code
uses: actions/checkout@v4
 - name: Login to AWS ECR
id: login-ecr
```

```
uses: aws-actions/amazon-ecr-login@v1
- name: Build, Tag, and Push Docker Image
env:
    ECR_REGISTRY: ${{ steps.login-ecr.outputs.registry }}
    IMAGE_TAG: ${{ github.sha }}
 run:
   docker build -t
$ECR_REGISTRY/$ECR_REPOSITORY_NAME:$IMAGE_TAG
   . docker push
$ECR_REGISTRY/$ECR_REPOSITORY_NAME:$IMAGE_TAG
      - name: Update ECS Task Definition
        id: task-def
         uses: aws-actions/amazon-ecs-render-task-definition@v1
         with:
          task-definition: task-definition.json
             container-name: my-container
             image: ${{ steps.login-ecr.outputs.registry }}/${{
secrets.ECR_REPOSITORY_NAME }}:${{ github.sha }}
    - name: Deploy to ECS
      uses: aws-actions/amazon-ecs-deploy-task-definition@v1
      with:
     cluster: ${{ secrets.ECS_CLUSTER }}
     service: ${{ secrets.ECS_SERVICE }}
     task-definition: ${{ steps.task-def.outputs.task-definition }}
     wait-for-service-stability: true
```

Step 5: Commit and Push Code

```
git add .
git commit -m "Added GitHub Actions for ECS deployment"
```

```
git push origin main
```

This triggers the GitHub Actions workflow, building the image, pushing it to ECR, and updating ECS.

Now, your GitHub Actions pipeline will automatically deploy to AWS ECS whenever you push changes to the main branch.