# Docker and AWS ECS Deployment Guide

## Prerequisites

- Docker installed on your local machine
- AWS CLI configured with appropriate permissions
- AWS account with ECS permissions

## Part 1: Docker Setup and Local Testing

### Step 1: Create Project Structure

```shell
mkdir my-web-app
cd my-web-app
```

### Step 2: Create the index.html file

Save the HTML content provided in the artifact as `index.html`

### Step 3: Create the Dockerfile

Save the Dockerfile content provided in the artifact as `Dockerfile`

### Step 4: Build the Docker Image

```shell
docker build -t my-web-app .
```

### Step 5: Run the Container Locally

```shell
docker run -d -p 8080:80 --name my-web-app-container my-web-app
```

### Step 6: Test Locally

Open your browser and navigate to `http://localhost:8080`

**Take a screenshot of your application running on localhost:8080**

### Step 7: Stop the Container (when done testing)

Shell
```
docker stop my-web-app-container
docker rm my-web-app-container
```

## Part 2: AWS ECS Deployment (Using AWS Console)

### Step 1: Create ECR Repository

1. Go to AWS Console → **Elastic Container Registry (ECR)**
2. Click **"Create repository"**
3. Set repository name: `my-web-app`
4. Leave other settings as default
5. Click **"Create repository"**
6. Click on your repository name and then **"View push commands"**
7. Follow the commands to push your Docker image:

Shell
```
# Get login token (from push commands)
aws ecr get-login-password --region us-east-1 | docker login
--username AWS --password-stdin
YOUR_ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com

# Tag the image
docker tag my-web-app:latest
YOUR_ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com/my-web-app:latest

# Push the image
docker push
YOUR_ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com/my-web-app:latest
```

### Step 2: Create Security Group

1. Go to **EC2 Console → Security Groups**
2. Click **"Create security group"**
3. Name: `my-web-app-sg`
4. Description: `Security group for my web app`
5. VPC: Select default VPC
6. **Inbound Rules**: Click **"Add rule"**
    - Type: `HTTP`
    - Protocol: `TCP`
    - Port: `80`
    - Source: `Anywhere-IPv4` (0.0.0.0/0)
7. Click **"Create security group"**

### Step 3: Create ECS Cluster

1. Go to **ECS Console**
2. Click **"Clusters" → "Create Cluster"**
3. Cluster name: `my-web-app-cluster`
4. Infrastructure: Select **"AWS Fargate (serverless)"**
5. Leave other settings as default
6. Click **"Create"**

### Step 4: Create Task Definition

1. In ECS Console, click **"Task definitions" → "Create new Task Definition"**
2. Task definition family: `my-web-app-task`
3. Launch type: **"AWS Fargate"**
4. Operating system: **"Linux/X86_64"**
5. CPU: `0.25 vCPU`
6. Memory: `0.5 GB`
7. Task execution role: Select `ecsTaskExecutionRole` (if it doesn't exist, it will be created automatically)

### Step 5: Add Container Definition

1. In the same task definition creation page, scroll to **"Container definitions"**
2. Click **"Add container"**
3. Container details:
    - **Name**: `web-app-container`
    - **Image URI**: Your ECR image URI (e.g., `YOUR_ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com/my-web-app:latest`)
    - **Port mappings**:

        - Container port: `80`
        - Protocol: `TCP`
        - Port name: `web-80-tcp`
        - App protocol: `HTTP`
4. **Logging** (optional but recommended):
    - Log driver: `awslogs`
    - awslogs-group: `/ecs/my-web-app` (will be created automatically)
    - awslogs-region: `us-east-1`
    - awslogs-stream-prefix: `ecs`
5. Click **"Create"**

## Step 6: Create ECS Service

1. Go to your cluster (`my-web-app-cluster`)
2. In the **Services** tab, click **"Create"**
3. **Compute configuration**:
    - Launch type: **"Fargate"**
    - Platform version: `LATEST`
4. **Deployment configuration**:
    - Application type: **"Service"**
    - Task definition: Select `my-web-app-task` (latest revision)
    - Service name: `my-web-app-service`
    - Desired tasks: `1`
5. **Networking**:
    - VPC: Select your default VPC
    - Subnets: Select at least 2 public subnets
    - Security group: Select the `my-web-app-sg` you created
    - **Public IP**: **"Enabled"** (Very important!)
6. Click **"Create"**

## Step 7: Wait for Deployment

1. The service will take a few minutes to deploy
2. Go to the **Tasks** tab in your service
3. Wait until the task status shows **"RUNNING"**

## Step 8: Get Public IP Address

1. Click on the running task
2. In the task details, find the **"Configuration"** section
3. Look for **"Public IP"** - this is your application's public IP address
4. Copy this IP address

### Step 9: Access Your Application

1. Open your browser
2. Navigate to `http://YOUR_PUBLIC_IP` (the IP from step 8)
3. **Take a screenshot of your application running on the public IP address**

## Alternative: Using Application Load Balancer (Optional)

If you want a permanent domain name instead of changing IP addresses:

1. **Create Application Load Balancer**:
    - Go to **EC2 Console** → **Load Balancers** → **Create Load Balancer**
    - Choose **"Application Load Balancer"**
    - Name: `my-web-app-alb`
    - Scheme: **"Internet-facing"**
    - IP address type: **"IPv4"**
    - VPC: Select your default VPC
    - Mappings: Select at least 2 public subnets
    - Security groups: Select the same security group you created
2. **Create Target Group**:
    - Target type: **"IP addresses"**
    - Target group name: `my-web-app-targets`
    - Protocol: `HTTP`, Port: `80`
    - VPC: Select your default VPC
    - Health check path: `/`
3. **Update ECS Service**:
    - Edit your ECS service
    - In **Load balancing**, choose **"Application Load Balancer"**
    - Select your load balancer and target group
    - Update the service

## Cleanup (Using Console)

1. **ECS Service**: Go to your cluster → Services → Select service → **"Update"** → Set desired tasks to 0 → **"Update"** → Then **"Delete"**
2. **ECS Cluster**: Delete cluster after service is removed
3. **Task Definition**: Deregister all revisions
4. **ECR Repository**: Delete repository and images
5. **Security Group**: Delete the security group

6. **Load Balancer** (if created): Delete ALB and target group

## Important Notes

1. **Replace Placeholders**: Make sure to replace `YOUR_ACCOUNT_ID` with your actual AWS account ID in all commands and configuration files.

2. **Region**: This guide uses `us-east-1`. Change the region if you prefer a different one.

3. **Costs**: Running ECS Fargate tasks incurs costs. Make sure to clean up resources when done.

4. **Screenshots**: Take screenshots at the specified steps:

    - Local application running on localhost:8080
    - Application running on the public IP address from ECS
5. **Permissions**: Ensure your AWS CLI is configured with sufficient permissions for ECS, ECR, EC2, IAM, and CloudWatch operations.