

Useful links:

<https://github.com/aws-actions/amazon-ecr-login>

Github actions will do the building and testing while githubs will utilize code deploy to do the deploy part of CI/CD.

Start by making a yaml file for your workflow that looks something like below

name: CI

on:

push:

branches: [mafia-dev]

pull_request:

branches: [mafia-dev]

workflow_dispatch:

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Run a one-line script

run: echo Hello, world!

- name: Run a multi-line script

run: |

echo Add other actions to build,

echo test, and deploy your project.

test:

needs: build

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Echo

run: echo This is a test.

Deploy:

needs: test

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Echo

run: echo this is a deployment

- name: Configure AWS credentials

uses: aws-actions/configure-aws-credentials@v1

with:

aws-access-key-id: \${{ secrets.AWS_ACCESS_KEY_ID }}

aws-secret-access-key: \${{ secrets.AWS_SECRET_ACCESS_KEY }}

aws-region: us-east-1

- name: Login to Amazon ECR

id: login-ecr

uses: aws-actions/amazon-ecr-login@v1

- name: Build, tag, and push the image to Amazon ECR

id: build-image

env:

ECR_REGISTRY: \${{ steps.login-ecr.outputs.registry }}

ECR_REPOSITORY: \${{ secrets.REPO_NAME }}

IMAGE_TAG: 1.0

run: |

docker build -t \$ECR_REGISTRY/\$ECR_REPOSITORY:\$IMAGE_TAG .

echo "Pushing image to ECR..."

docker push \$ECR_REGISTRY/\$ECR_REPOSITORY:\$IMAGE_TAG

echo "::set-output name=image::\$ECR_REGISTRY/\$ECR_REPOSITORY:\$IMAGE_TAG"

Create a private ecr repository

Create an aws user

Add permissions and attach existing policy **AmazonEC2ContainerRegistryFullAccess**

Create the user

Copy the access and secret access id

On github click secrets

Click on add new repository secrets

Add REPO_NAME as the environment name and for the value add your ECR name

Add AWS_ACCESS_KEY_ID as name and paste the value

ADD AWS_SECRET_ACCESS_KEY_ID as name and paste value

Push some code and try the workflow