
Project-7
Name:[N Narendra]
Project:[FlaskApp-Jenkins-Docker-ECR]

Project: **Build and Deploy of Python Flask APP and Push dockerized Flask app image to AWS ECR by Jenkins Pipeline** [Docker + Jenkins + ECR]
Reference Video: https://youtu.be/Q5sTM5LuhuI [Devops by Meenu]

Step 1: Application code files in Github
Step 2: Create Dockerfile to create docker image
Step 3: Create Jenkins Pipeline Script in scriptive (within Job dashboard) or declarative (Jenkinsfile)
Step 4: Launch Amazon Linux2 EC2 and 22, 80, 8080, 8096 ports opened
Step 5: Docker & Jenkins both installed same EC2 and AWS CLI also to be installed & configured on same EC2
Step 6: Create ECR Repo
Step 7: Create IAM user with AmazonEC2ContainerRegistryFullAccess and EC2Fullaccess & save access its keys
Step 8: Install and configure Git, docker Plug-ins, AWS IAM user credentials in Jenkins dashboard
Step 9: Create Pipeline Job and build it to push image to ECR and check pushed or not

Required files: Dockerfile, Jenkinsfile, app.py, requirements.txt, templates folder

Task-1 (Git & Docker) on EC2

```
1. sudo yum update -y
2. Sudo yum install git -y
3. sudo amazon-linux-extras install docker -y
4. sudo systemctl start docker, sudo systemctl enable docker, sudo systemctl status docker
5. sudo usermod -a -G docker ec2-user
6. Sudo usermod -a -G docker jenkins
7. /var/run/docker.sock
```

Task-2 (Jenkins) on EC2

```
1. amazon-linux-extras install java-openjdk11 -y
2. sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
3. sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
4. yum install fontconfig java-11-openjdk
5. yum install jenkins
6. systemctl start jenkins
7. systemctl enable jenkins
8. systemctl status jenkins
9. cat /var/lib/jenkins/secrets/initialAdminPassword [login into Jenkins dashboard public IP:8080]
```

Create Pipeline Job and run the build in Jenkins dashboard and check the dockerised Flask App Image in ECR and access the Python based Flask App in browser PublicIP:8096

Task-3 Dockerfile

```
# this is my base image
FROM alpine:3.5
# Install python and pip
RUN apk add --update py2-pip
# install Python modules needed by the Python app
COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt
# copy files required for the app to run
COPY app.py /usr/src/app/
COPY templates/index.html /usr/src/app/templates/
# tell the port number the container should expose
EXPOSE 5000
# run the application
CMD ["python", "/usr/src/app/app.py"]
```

Task-4 requirements.txt

```
Flask==0.10.1
```

Task-5 Jenkinsfile

```
pipeline {
    agent any
     environment {
        registry = "167613117387.dkr.ecr.us-east-1.amazonaws.com/nn-ecr-jenkins"
    }

    stages {
          stage('Checkout') {
            steps {
                git branch: 'main', url: 'https://github.com/Narian318/nn-ECR-Jenkins.git'
            }
        }
           stage('Building image') {
             steps{
                  script {
                   dockerImage = docker.build registry
                   }
      }
           }

            stage('Pushing to ECR') {
             steps{
                  script {
               withCredentials([[$class:   'AmazonWebServicesCredentialsBinding',   credentialsId:
'aws_cred', accessKeyVariable: 'AWS_ACCESS_KEY_ID', secretKeyVariable: 'AWS_SECRET_ACCESS_KEY']])
{
    sh 'aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-
stdin 167613117387.dkr.ecr.us-east-1.amazonaws.com'
     sh 'docker push 167613117387.dkr.ecr.us-east-1.amazonaws.com/nn-ecr-jenkins:latest'
}

}
                   }
            }
             stage('stop previous containers') {
               steps {
      sh 'docker ps -f name=mypythonContainer -q | xargs --no-run-if-empty docker container stop'
      sh 'docker container ls -a -fname=mypythonContainer -q | xargs -r docker container rm'
         }
       }
            stage('Docker Run') {
              steps{
                   script {
      sh 'docker run -d -p 8096:5000 --rm --name mypythonContainer 167613117387.dkr.ecr.us-east-
1.amazonaws.com/nn-ecr-jenkins:latest'
      }
    }
        }
    }
  }
```

Task-6 app.py

```
from flask import Flask, render_template
import random

app = Flask(__name__)

# list of cat images
images = [
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr06/15/9/anigif_enhanced-buzz-25158-
1381844793-0.gif",
    "http://ak-hdl.buzzfed.com/static/2013-10/enhanced/webdr03/15/10/anigif_enhanced-buzz-11980-
1381846269-1.gif"
]
```

```
@app.route('/')
def index():
    url = random.choice(images)
    return render_template('index.html', url=url)

if __name__ == "__main__":
    app.run(host="0.0.0.0")
```