**Assignment:**

Create a CICD pipeline in Jenkins to fetch Java code from GitHub repository, containerize using Docker, push to ECR and deploy to EKS so that users can access the application viz URL.

---------------------------------------------------------------------------------------------------------------------------------------

**Technology Used:**

Docker, ECR, EKS, GitHub Actions

---------------------------------------------------------------------------------------------------------------------------------------

**Folder structure:**

**|\_** src

**|\_** .github

**|\_**workflows

**|\_** deploy.yml

**|\_** Dockerfile

**|\_** pom.xml

|\_ k8s/

**|\_** deployment.yaml

**|\_** service.yaml

---------------------------------------------------------------------------------------------------------------------------------------

**Setup**

* Create new repository in GitHub.
* Git Configuration

1. Configure Git user info in local

$ git config --global user.name "shivam"

$ git config --global user.email [shivamthakur0567@gmail.com](mailto:shivamthakur0567@gmail.com)

1. Initialize Git Repository

$ git init

1. Add files to Git:

$ git add .

1. Commit changes:

$ git commit -m "Initial commit"

1. Check current local branch

$ git branch

Master

1. Rename local branch to main

$ git branch -m main

1. Add our remote repository

$ git remote add origin https://github.com/shivam-th/java-hello-world.git

1. Push the code from local to remote repository

$ git push -u origin main --force

* AWS Configuration

$ aws configure

AWS Access Key ID [None]:

AWS Secret Access Key [None]:

Default region name [None]:

Default output format [None]:

---------------------------------------------------------------------------------------------------------------------------------------

Jenkins Instance Setup

* Visit

EC2 > Instances > Launch instances

* Give name for instance

Names and tags > Name – jenkins-server

* Select AMI

Application and OS Images > Quick Start > Select – Ubuntu (Ubuntu Server 22.04)

* Select Instance type

Instance type > Instance type – t2.small

* Create or choose existing key

Key pair > Create new key pair > Key pair name – jenkins-key > Type – RSA > File Format – .pem > Create Key pair

* Create security group

Network settings > Choose – Create security group > Edit

Security group name – jenkins-sg > Description – jenkins-sg

Type – ssh > Source type – MyIP > Add security group

Type – Custom TCP > Port range – 8080 > Source type – Anywhere > Description – allow port 8080

* Add provisioner

Advanced details > User data

#!/bin/bash

sudo apt update

sudo apt install openjdk-11-jdk -y

sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

<https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key>

echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \

<https://pkg.jenkins.io/debian-stable> binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update

sudo apt-get install jenkins -y

* Verify

Browse – <instance public IP>:8080

* Login to Jenkins server

$ ssh -i <key-pair-file-name> ubuntu@<public-p>

* Home directory

$ ls /var/lib/jenkins

* Check service

$ system status jenkins

* Copy password

$ cat /var/lib/jenkins/secrets/initialAdminPassword

12345678

* Unlock Jenkins

Administrator password – 12345678 > Continue

* Customize Jenkins

Select plugins to install > Uncheck plugins if not required (ant) > Install

* Create First Admin User

Username – admin

Password – admin123

Fullname – Admin

E-mail address – [admin@gmail.com](mailto:admin@gmail.com)

* Save & Continue
* If your server IP is dynamic, then give any domain

<http://jenkins.xyz> > Save and Finish