**DevOps Project**

**Problem Statement:**

Create an end-to-end CI/CD pipeline in AWS platform using Jenkins as the orchestration tool, GitHub as scm, maven as the build tool, deploy in a docker instance and create a docker image, store the docker image in ECR, Kubernetes deployment using ECR image. Build sample java web app using maven.

**Approach:**

**Requirements:**

* CI/CD pipeline System
* Git - local version control system.
* GitHub - As Distributed version control system.
* Jenkins - Continuous Integration and Orchestration tool.
* Maven - As a Build Tool.
* ECR – AWS Container Registry
* docker -Containerization
* Kubernetes - As Container Orchestration Tool

**Step-1:**

* Install Jenkins, Maven, Git.
* Configure Jenkins global tools and install necessary plugins.
* Setup Tomcat server.
* Create a pipeline job in Jenkins.
* Create Jenkins file on pipeline job.
* Test deployment by accessing Tomcat URL.

**Step-2:**

* Create a Docker Host and Install Docker on it.
* Create a repo in ECR.
* Update Jenkins file to Integrate with Docker Host.
* Use declarative pipeline to build and push image to ECR.

**Step-3:**

* Setup EKS Host.
* Install AWSCLI, Kubectl, Eksctl.
* Configure AWS.
* Create cluster.
* Create deployment.yaml, service.yaml.
* Update Jenkins file to Integrate with EKS Host.
* Use declarative pipeline to deploy**.**

**Step-4:**

➢ Deploy artifacts on the Kubernetes

➢ Write codes in the artifacts of docker and Kubernetes which we want to run.

➢ Now build the code in Jenkins.

➢ Check in Kubernetes the pods are getting created or not.

➢ Now copy the service IP and paste it in the browser and check the output.

**Solution:**

An EC2 instance is created for git operation

A screenshot of a computer

AI-generated content may be incorrect.

Repository has been cloned

A computer screen shot of a black screen

AI-generated content may be incorrect.

SSH key is generated and added in GitHub for connection with GitHub to push it to my GitHub.

A screenshot of a computer

AI-generated content may be incorrect.

Created a repository name java\_project and pushed it to the GitHub.

A computer screen with white text

AI-generated content may be incorrect.

Added in repository in GitHub

A screenshot of a computer

AI-generated content may be incorrect.

Created Jenkins and Docker Instance

A screenshot of a computer

AI-generated content may be incorrect.

Creating tomcat instance

A screenshot of a computer

AI-generated content may be incorrect.

Attaching roles in tomcat-users.xml in tomcat instance

A computer screen with text

AI-generated content may be incorrect.

starting tomcat server

A computer screen shot of a black screen

AI-generated content may be incorrect.

Paste tomcat public ip in browser

A screenshot of a computer

AI-generated content may be incorrect.

Shell script for installing Jenkins

A computer screen shot of a black screen

AI-generated content may be incorrect.

Java, Git, Maven, Jenkins is installed in Jenkins instance and started Jenkins

A screenshot of a computer

AI-generated content may be incorrect.

Installing plugins

A screenshot of a computer

AI-generated content may be incorrect.

Configuring Java path in tools in Jenkins

A blue line on a white background

AI-generated content may be incorrect.

Configuring Maven path in tools in Jenkins

A blue line on a white background

AI-generated content may be incorrect.

Adding webhook to my GitHub repository to establish connection with Jenkins

A screenshot of a computer

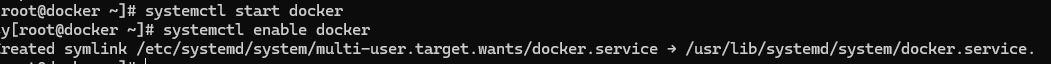
AI-generated content may be incorrect.

Creating a new pipeline Job for maven project

A screenshot of a computer

AI-generated content may be incorrect.

Now in Docker instance install Docker and start Docker



Generate SSH-KEY in docker

A computer screen with white text

AI-generated content may be incorrect.

Generate SSH-KEY in Jenkins

A computer screen with white text

AI-generated content may be incorrect.

In both Jenkins and Docker go to /etc/ssh/sshd\_config file and enable Password authentication and permit root login

A black background with white text

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

Now restart sshd and give password in both Docker and Jenkins

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

In both Docker and Jenkins, add private ip address and hostname of both docker and jenkins in /etc/hosts in both instances

A computer screen with white text

AI-generated content may be incorrect.

Copy the ssh key from Jenkins to docker

A computer screen with white text

AI-generated content may be incorrect.

In Jenkins go to system, in that SSH Server add docker private ip and password

A screenshot of a notebook

AI-generated content may be incorrect.

Now create an IAM user and allow administrator access

A screenshot of a computer

AI-generated content may be incorrect.

Create credentials like access key and secret key for that IAM user and give them in docker instance for accessing of ECR

A black background with white text

AI-generated content may be incorrect.

Now create public repository in ECR for storing images

A screen shot of a computer

AI-generated content may be incorrect.

Now in Jenkins pipeline, writing stages in those steps in Jenkins file for cloning repository

A screenshot of a computer

AI-generated content may be incorrect.

Writing Jenkins file for building stage using maven and deploying stage war files to tomcat server

A screenshot of a computer

AI-generated content may be incorrect.

Now writing a stage in Jenkins file to transfer java project folder into images folder in docker and also executing commands that create latest image and push into public repository in ECR

A screenshot of a computer

AI-generated content may be incorrect.

Pasting those push commands in the Jenkins file

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Creating an instance for EKS node for creation of EKS cluster

A screenshot of a computer

AI-generated content may be incorrect.

Adding IAM role to EKS-node instance which has access to IAMfull access, EKS full access, ECR full access

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Install kubectl, eksctl, aws cli in eks-node instance and create cluster

A screenshot of a computer screen

AI-generated content may be incorrect.

Cluster devops created in AWS EKS

A screenshot of a computer

AI-generated content may be incorrect.

We get two target nodes

A screen shot of a computer program

AI-generated content may be incorrect.

Create deployment file in eks-node instance

A screen shot of a computer

AI-generated content may be incorrect.

Create service file in eks-node instance

A screen shot of a computer

AI-generated content may be incorrect.

Now writing a stage in Jenkins file to apply deployment and service file in eks-node



A screenshot of a computer

AI-generated content may be incorrect.

Now for post build stage print success for pipeline works fine

A screenshot of a computer

AI-generated content may be incorrect.

Selecting github scm trigger checkbox to check for changes

A screenshot of a computer

AI-generated content may be incorrect.

Making changes in github repository and committing changes

A screenshot of a computer

AI-generated content may be incorrect.

Build automatically triggers

A screenshot of a computer

AI-generated content may be incorrect.

Build success

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Groovy script:-

pipeline {  
    agent any  
       environment {  
        TOMCAT\_USER = 'deployer'  
        TOMCAT\_PASS = 'deployer'  
        TOMCAT\_HOST = '34.226.247.144'  
        TOMCAT\_PORT = '8080'  
    }  
    stages {  
        stage('Clone Repository') {  
            steps {  
                git branch: 'main', url: '[https://github.com/tonybabu2004-eng/java-project.git'](https://github.com/tonybabu2004-eng/java-project.git%27)  
            }  
        }  
        stage('Build with Maven') {  
            steps {  
                sh 'mvn clean package -Dmaven.test.failure.ignore=true'  
            }  
        }  
        stage('Deploy to Tomcat') {  
            steps {  
                sh '''  
                    curl -u $TOMCAT\_USER:$TOMCAT\_PASS \  
                    --upload-file webapp/target/webapp.war \  
                    "http://$TOMCAT\_HOST:$TOMCAT\_PORT/manager/text/deploy?path=/webapp&update=true"  
                '''  
            }  
        }  
        stage('Deploy to Docker Host') {  
            steps {  
                sshPublisher(publishers: [  
                    sshPublisherDesc(  
                        configName: 'docker-host',   
                        transfers: [  
                            sshTransfer(  
                                sourceFiles: 'java-deployment.yaml,java-service.yaml',  
                                removePrefix: '',  
                                remoteDirectory: 'images',  
                                execCommand: '''  
                                    cd images  
                                    aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/m2r5y6g7  
                                    docker build -t project .  
                                    docker tag project:latest public.ecr.aws/m2r5y6g7/project:latest  
                                    docker push public.ecr.aws/m2r5y6g7/project:latest  
                                '''  
                            )  
                        ]  
                    )  
                ])  
            }  
        }  
    }  
    post {  
        success {  
            junit '\*\*/target/surefire-reports/TEST-\*.xml'  
            archiveArtifacts artifacts: '\*\*/target/\*.war', fingerprint: true  
        }  
    }  
}

Image created and pushed to ECR

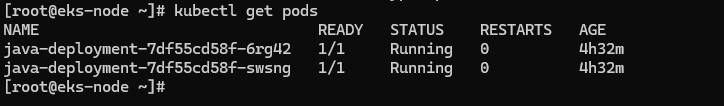
A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Deployment file in eks-node pulls latest image and creates deployment and container in pods with latest image



Paste service Ip in browser and changes reflected

A screenshot of a computer

AI-generated content may be incorrect.

Done By –

Yelakapati Tony babu

Ps No. 10844611