Create an IAM user with administrator access and generate secret access keys for aws configure

Create an IAM role with administrator access and add it to your Jenkins machine, in case when you use slave attach it to slave machine

INSTALLATIONS

Install awscli, eksctl, kubectl on the jenkins server

awscli =>

=========

sudo apt install awscli

aws configure => give the security credentials

eksctl

=======

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp

sudo mv /tmp/eksctl /usr/local/bin

eksctl version

kubectl

=======

sudo curl --silent --location -o /usr/local/bin/kubectl https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.6/2022-03-09/bin/linux/amd64/kubectl

sudo chmod +x /usr/local/bin/kubectl

kubectl version --short --client

eksctl create cluster --name cloud-eks --region ap-south-1 --nodegroup-name my-nodes --node-type t3.small --managed --nodes 1

aws eks update-kubeconfig --name cloud-eks --region ap-south-1

eksctl delete cluster --name cloud-eks --region ap-south-1

USE JAVA 11

USE MAVEN 3.8.1 for this project petclinic

|  |  |  |  |
| --- | --- | --- | --- |
| [Kubernetes :: Pipeline :: DevOps StepsVersion1.6](https://plugins.jenkins.io/kubernetes-pipeline-devops-steps)  [Report an issue with this plugin](https://www.jenkins.io/participate/report-issue/redirect/#21630) |  |  | Top of Form  Bottom of Form |
| [Kubernetes CLI PluginVersion1.12.0](https://plugins.jenkins.io/kubernetes-cli)  Configure kubectl for Kubernetes  [Report an issue with this plugin](https://github.com/jenkinsci/kubernetes-cli-plugin/issues/new/choose) |  | Top of Form  **1.12.0**  Bottom of Form | Top of Form  Bottom of Form |
| [Kubernetes Client API PluginVersion6.4.1-215.v2ed17097a\_8e9](https://plugins.jenkins.io/kubernetes-client-api)  Kubernetes Client API plugin for use by other Jenkins plugins.  [Report an issue with this plugin](https://www.jenkins.io/participate/report-issue/redirect/#25921) |  |  | Top of Form  Bottom of Form |
| [Kubernetes Credentials PluginVersion0.10.0](https://plugins.jenkins.io/kubernetes-credentials)  Common classes for Kubernetes credentials  [Report an issue with this plugin](https://www.jenkins.io/participate/report-issue/redirect/#23230) |  |  | Top of Form  Bottom of Form |
| [Kubernetes Credentials ProviderVersion1.225.v14f9e6b\_28f53](https://plugins.jenkins.io/kubernetes-credentials-provider)  Provides a read only credentials store backed by Kubernetes.  [Report an issue with this plugin](https://www.jenkins.io/participate/report-issue/redirect/#23531) |  |  | Top of Form  Bottom of Form |
| [Kubernetes pluginVersion3950.v581298fa\_e4e7](https://plugins.jenkins.io/kubernetes)  This plugin integrates Jenkins with [Kubernetes](https://github.com/GoogleCloudPlatform/kubernetes/) |  |  |  |

pipeline {

    agent {

        label "Agent1"

    }

    tools {

        jdk "java17"

        maven "M3"

    }

    environment {

        APP\_NAME = "petclinic"

        RELEASE = "1.0.0"

        DOCKER\_USER = "sevenajay"

        DOCKER\_PASS = "dockerhub"

        IMAGE\_NAME = "${DOCKER\_USER}" + "/" + "${APP\_NAME}"

        IMAGE\_TAG = "${RELEASE}-${BUILD\_NUMBER}"

        DEPLOYMENT\_FILE = "kubernetes/petclinic.yaml"

    }

    stages {

        stage("clean workspace"){

            steps {

                cleanWs()

            }

        }

         stage("checkout scm"){

            steps {

                checkout scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/Aj7Ay/amazon-eks-jenkins-terraform-aj7.git']])

            }

        }

         stage("build stage"){

            steps {

                sh "mvn clean package"

            }

        }

         stage("test code"){

            steps {

                sh "mvn test"

            }

        }

         stage("sonar checks"){

            steps {

                script {

                withSonarQubeEnv(credentialsId: 'sonar') {

                     sh "mvn sonar:sonar"

                }

            }

        }

        }

        stage("quality gate"){

            steps {

                script {

                waitForQualityGate abortPipeline: false, credentialsId: 'sonar'

            }

        }

        }

        stage ("docker build") {

            steps {

                script {

                    docker.withRegistry('',DOCKER\_PASS) {

                        // Remove old images from docker repository

                        sh "docker images --format '{{.Repository}}:{{.Tag}}' | grep ${IMAGE\_NAME} | grep -v ${RELEASE}-${BUILD\_NUMBER} | grep -v latest | xargs -I {} docker rmi {} || true"

                        docker\_image = docker.build "${IMAGE\_NAME}"

                    }

                }

            }

        }

        stage ("Trivy image scan") {

            steps {

                script {

                    sh "trivy image ${docker\_image.id} > trivy.txt"

                }

            }

        }

        stage ("PUSH docker image") {

            steps {

                script {

                    docker.withRegistry('',DOCKER\_PASS) {

                        docker\_image.push("${IMAGE\_TAG}")

                        docker\_image.push('latest')

                    }

                }

            }

        }

        stage ("deployment to k8s") {

            steps {

                script {

                    withKubeConfig(caCertificate: '', clusterName: '', contextName: '', credentialsId: 'k8s', namespace: '', restrictKubeConfigAccess: false, serverUrl: '') {

                    //UPDATE THE IMAGE NAME IN DEPLOYMENT YAML FILE

                    sh "sed -i '/^\\s\*image:/ s|.\*|        image: ${IMAGE\_NAME}|' ${DEPLOYMENT\_FILE}"

                    //APPLY THE UPDATED YAML FILE

                    sh "kubectl apply -f ${DEPLOYMENT\_FILE}"

                    }

                }

            }

        }

    }

    post {

    always {

        emailext attachLog: true,

            subject: "'${currentBuild.result}'",

            body: "Project: ${env.JOB\_NAME}<br/>" +

                "Build Number: ${env.BUILD\_NUMBER}<br/>" +

                "URL: ${env.BUILD\_URL}<br/>",

            to: 'postbox.aj99@gmail.com',

            attachmentsPattern: 'trivy.txt'

        }

    }

}

FROM anapsix/alpine-java

VOLUME /tmp

ADD target/spring-petclinic-2.1.0.war app.war

EXPOSE 8080

ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.war"]

If you want with tag name also in K8S DEPLOYMENT USE THIS STAGE

stage("k8s deployment") {

steps {

script {

// Update the image in the deployment YAML file

sh "sed -i '/^\\s\*image:/ s|.\*| image: ${IMAGE\_NAME}:${IMAGE\_TAG}|' ${DEPLOYMENT\_FILE}"

// Apply the updated YAML file

sh "kubectl apply -f ${DEPLOYMENT\_FILE}"

}

}

}