A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

PIPELINE CODE:

pipeline {

agent any

environment {

DOCKER\_IMAGE = "kishoreg12/maven"

DOCKER\_TAG = "latest"

DOCKER\_CREDENTIALS\_ID = "docker"

GITHUB\_CREDENTIALS\_ID = "github\_seccred"

KUBECONFIG = "/var/lib/jenkins/.kube/config"

MAVEN\_HOME = "/opt/maven"

}

stages {

stage('Checkout Code') {

steps {

script {

retry(3) {

try {

git credentialsId: GITHUB\_CREDENTIALS\_ID, url: 'https://github.com/kishoregunasekaran/spring-boot-apps.git', branch: 'main'

} catch (Exception e) {

echo "HTTPS clone failed, retrying with SSH..."

git credentialsId: GITHUB\_CREDENTIALS\_ID, url: 'git@github.com:kishoregunasekaran/spring-boot-apps.git', branch: 'main'

}

}

}

}

}

stage('Build Application') {

steps {

script {

try {

sh "${MAVEN\_HOME}/bin/mvn clean package -DskipTests"

} catch (Exception e) {

error "❌ Build failed: ${e.getMessage()}"

}

}

}

}

stage('Run Maven Tests') {

steps {

script {

catchError(buildResult: 'SUCCESS', stageResult: 'FAILURE') {

try {

sh "${MAVEN\_HOME}/bin/mvn test"

} catch (Exception e) {

error "⚠️ Tests failed but proceeding: ${e.getMessage()}"

}

}

}

}

}

stage('Build Docker Image') {

steps {

script {

try {

sh "docker build -t ${DOCKER\_IMAGE}:${DOCKER\_TAG} ."

} catch (Exception e) {

error "❌ Docker build failed: ${e.getMessage()}"

}

}

}

}

stage('Push Docker Image') {

steps {

withDockerRegistry([credentialsId: DOCKER\_CREDENTIALS\_ID, url: '']) {

script {

retry(3) {

try {

sh "docker push ${DOCKER\_IMAGE}:${DOCKER\_TAG}"

} catch (Exception e) {

error "❌ Docker push failed: ${e.getMessage()}"

}

}

}

}

}

}

stage('Deploy to Kubernetes') {

when {

expression { return fileExists('scripts/deploy.sh') }

}

steps {

script {

try {

sh '''

chmod +x scripts/deploy.sh

./scripts/deploy.sh

'''

} catch (Exception e) {

error "❌ Kubernetes deployment failed: ${e.getMessage()}"

}

}

}

}

}

post {

success {

echo "✅ Deployment Successful!"

}

failure {

echo "❌ Deployment Failed!"

}

}

}