pipeline {

agent any

environment {

// Define paths for your test results

LOCAL\_TEST\_RESULT\_FILE = 'D:\\JavaFiles\\task\_automation\_main\\target\\surefire-reports\\TEST-org.deloitte.runner.TestRunner.xml'

WORKSPACE\_TEST\_RESULT\_FILE = 'target\\surefire-reports\\TEST-org.deloitte.runner.TestRunner.xml'

}

stages {

stage('Run Tests') {

steps {

script {

// Run your tests and capture the exit code

dir('D:/JavaFiles/task\_automation\_main') { // Directory where your pom.xml is located

def testResult = bat(returnStatus: true, script: 'mvn clean test')

// Save the exit code to the environment variable

env.TEST\_RESULT = testResult

}

}

}

}

stage('Prepare Directories') {

steps {

script {

// Create directories if they don't exist

bat 'if not exist "target\\surefire-reports" mkdir "target\\surefire-reports"'

}

}

}

stage('Copy Test Results') {

steps {

script {

// Copy the test result file to the Jenkins workspace

bat "copy \"${env.LOCAL\_TEST\_RESULT\_FILE}\" \"${env.WORKSPACE\_TEST\_RESULT\_FILE}\""

}

}

}

stage('Archive Test Results') {

steps {

// Archive XML test results

archiveArtifacts artifacts: "${env.WORKSPACE\_TEST\_RESULT\_FILE}", allowEmptyArchive: true

}

}

stage('Publish Test Results') {

steps {

junit '\*\*/target/surefire-reports/TEST-org.deloitte.runner.TestRunner.xml'

}

}

stage('Calculate Pass Percentage') {

steps {

script {

// Calculate pass percentage from test results

def testResultAction = currentBuild.rawBuild.getAction(hudson.tasks.junit.TestResultAction.class)

def totalTests = testResultAction?.totalCount ?: 0

def failedTests = testResultAction?.failCount ?: 0

def passedTests = totalTests - failedTests

def passPercentage = totalTests > 0 ? (passedTests / totalTests) \* 100 : 0

echo "Total Tests: ${totalTests}"

echo "Passed Tests: ${passedTests}"

echo "Failed Tests: ${failedTests}"

echo "Pass Percentage: ${passPercentage}%"

def passThreshold = 90

if (passPercentage < passThreshold) {

echo "Pass percentage is below ${passThreshold}%. Triggering revert action..."

bat 'scripts\\revert-build.bat'

} else {

echo "Pass percentage is above ${passThreshold}%. No revert action needed."

}

// Save passPercentage to the environment

env.PASS\_PERCENTAGE = passPercentage.toString()

}

}

}

stage('Notify') {

steps {

script {

echo "Sending notification to the team..."

def passPercentage = env.PASS\_PERCENTAGE

// Example: using an email plugin or other notification mechanism

// mail to: 'srihari18892@gmail.com', subject: "Build Result", body: "Pass Percentage: ${passPercentage}%"

}

}

}

}

post {

always {

echo "Pipeline execution completed."

script {

if (env.TEST\_RESULT != '0') {

currentBuild.result = 'FAILURE'

}

}

}

}

}

method hudson.model.Actionable getAction java.lang.Class

method hudson.tasks.test.AbstractTestResultAction getFailCount

method hudson.tasks.test.AbstractTestResultAction getTotalCount

method org.jenkinsci.plugins.workflow.support.steps.build.RunWrapper getRawBuild

method hudson.model.Actionable getAction java.lang.Class

method org.jenkinsci.plugins.workflow.support.steps.build.RunWrapper getRawBuild

method org.jenkinsci.plugins.workflow.support.steps.build.RunWrapper getRawBuild