Slack

1. Setup sonar
2. Setup slack

pipeline {

agent {

label 'slave'

}

environment {

SONARQUBE\_URL = 'http://192.168.111.130:9000' /\* Adjust SonarQube server URL if needed \*/

SONARQUBE\_PROJECT\_KEY = 'myproject'

}

stages {

stage('Checkout SCM') {

steps {

script {

checkoutSCM()

}

slackNotification('Checkout SCM')

}

}

stage('Compiling and Running Test Cases') {

steps {

script {

compileAndRunTests()

}

slackNotification('Compiling and Running Test Cases')

}

}

stage('Generating Cucumber Reports') {

steps {

script {

generateCucumberReports()

}

slackNotification('Generating Cucumber Reports')

}

}

stage('Creating Package') {

steps {

script {

createPackage()

}

slackNotification('Creating Package')

}

}

stage('Adding Generate Report') {

steps {

script {

addGenerateReport()

}

slackNotification('Adding Generate Report')

}

}

stage('Analyzing Code Quality') {

steps {

script {

analyzeCodeQuality()

}

slackNotification('Analyzing Code Quality')

}

}

}

}

def checkoutSCM() {

checkout scm: [

$class: 'GitSCM',

branches: [[name: '\*/master']],

userRemoteConfigs: [[url: 'https://github.com/hellokaton/java11-examples.git']]

]

}

def compileAndRunTests() {

sh 'mvn clean'

sh 'mvn compile'

sh 'mvn test'

}

def generateCucumberReports() {

sh 'mvn verify'

}

def createPackage() {

sh 'mvn package'

}

def addGenerateReport() {

sh 'mvn verify'

}

def analyzeCodeQuality() {

withCredentials([string(credentialsId: '1116c32e-5de3-43cb-9c18-e704c00d238a', variable: 'SONAR\_TOKEN')]) {

sh '''

mvn sonar:sonar \

-Dsonar.projectKey="$SONARQUBE\_PROJECT\_KEY" \

-Dsonar.host.url="$SONARQUBE\_URL" \

-Dsonar.login="$SONAR\_TOKEN"

'''

}

}

def slackNotification(stageName) {

slackSend channel: 'proj-amitowjenkins',

message: "Stage '${stageName}' completed",

tokenCredentialId: 'a2a014af-280c-4393-9f3d-d15aed59bd0c',

username: 'bot'

}