pipeline {

agent any

stages {

stage ('Clone') {

steps {

git branch: 'main', credentialsId: '59397ac7-59e9-456a-ad83-4752f8059d24', url: 'https://github.com/7gowda/new\_repo.git'

}

}

stage ("build") {

steps {

sh "mvn clean install"

}

}

stage ("code\_quality") {

steps {

withSonarQubeEnv('SonarQube') {

sh 'mvn clean package sonar:sonar -Dsonar.projectKey=maven\_03 -Dsonar.host.url=http://13.59.229.253:9000 -Dsonar.login=f7987e163ead4a90dba74ddc92c3d65ecf08a4ec'

}

}

}

stage ('Upload file') {

steps {

rtUpload (

// Obtain an Artifactory server instance, defined in Jenkins --> Manage Jenkins --> Configure System:

serverId: "artifactory-ID",

spec: """{

"files": [

{

"pattern": "/var/lib/jenkins/workspace/jfrog\_artifactory/target/hello-world-war-1.0.0.war",

"target": "libs-snapshot-local"

}

]

}"""

)

}

}

stage ('Publish build info') {

steps {

rtPublishBuildInfo (

serverId: "artifactory-ID",

)

}

}

stage ('Set output resources') {

steps {

// 'jfPipelines' step will be skipped if the build is not triggered by JFrog Pipelines.

jfPipelines(

/\*\*

\* Sets the output resources to send to JFrog Pipelines.

\* 'pipelinesBuildInfo' is the build-info resource defined in JFrog Pipelines.

\*/

outputResources: """[

{

"name": "pipelinesBuildInfo",

"content": {

"buildName": "${env.JOB\_NAME}",

"buildNumber": "${env.BUILD\_NUMBER}"

}

}

]"""

)

}

}

stage ('tomat\_deploy') {

steps {

sshagent(credentials: ['07tomcat'], ignoreMissing: true) {

sh 'scp -o StrictHostKeyChecking=no target/hello-world-war-1.0.0.war ubuntu@3.131.90.203:/opt/ubuntu/webapps'

}

}

}

}

}