How To - Nexus Jenkins Integration

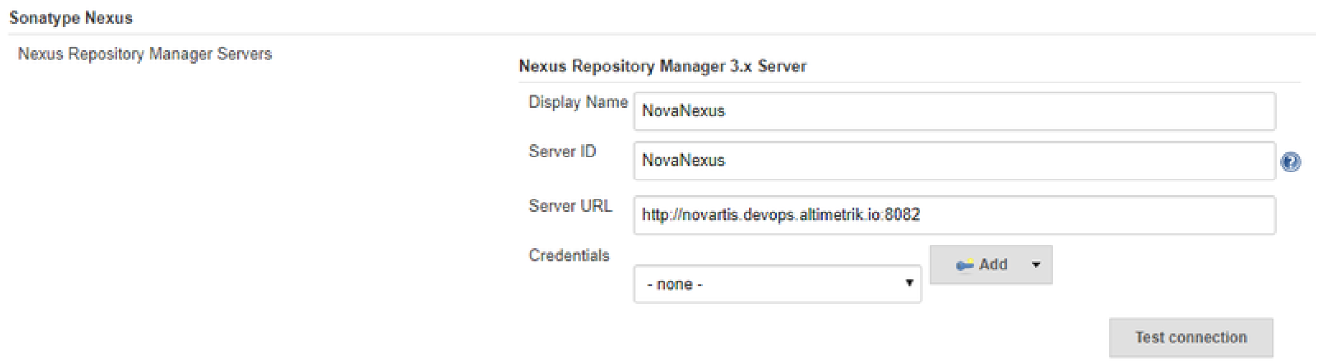
1. Jenkins Plugin Installation -

There are few plugins need to install in Jenkins to use Docker within Jenkins jobs and Pipeline. Go to Manage Jenkins Manage Plugins Available tab search for Nexus then install Nexus Platform Plugin and Nexus Artifact Uploader Plugin.

2. Setting Nexus configuration in Jenkins -

To use the Nexus within Jenkins jobs and pipeline; need to configure Nexus in Jenkins as other tools.

Go to Manage Jenkins Configure System Sonatype Nexus Add Nexus Repository Manager Server Nexus Repository Manager 3.x Server.



Sonatype Nexus

Nexus Repository Manager Servers

Nexus Repository Manager 3.x Server

Display Name: NovaNexus

Server ID: NovaNexus

Server URL: http://novartis.devops.altimetrik.io:8082

Credentials: - none -

Display Name: Mention any name for Nexus server

Server ID: Mention any name for Nexus server to use as a Server ID

Server URL: Nexus Server URL

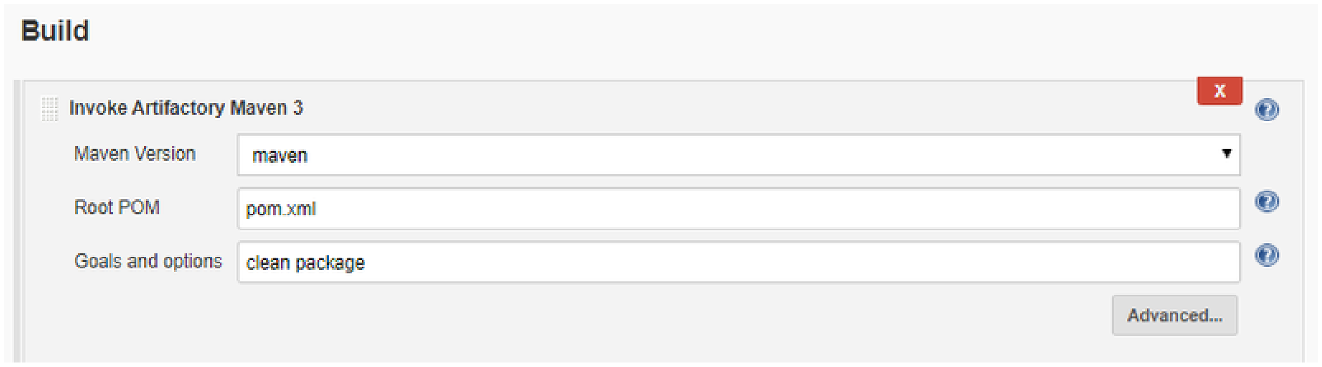
Credentials: Mention Credentials to authenticate

Click on "Test Connection" and it will show connectivity with Nexus server.

3. Jenkins Job Execution -

3.1 Free Style Job -

When working for Jenkins Freestyle job then need to provide the option and make some configuration. However this particular example is fetching the code from GitLab or Bitbucket and building with the maven build tool (pom.xml) with goal package. It will create artifacts like war/jar/ear etc.



Build

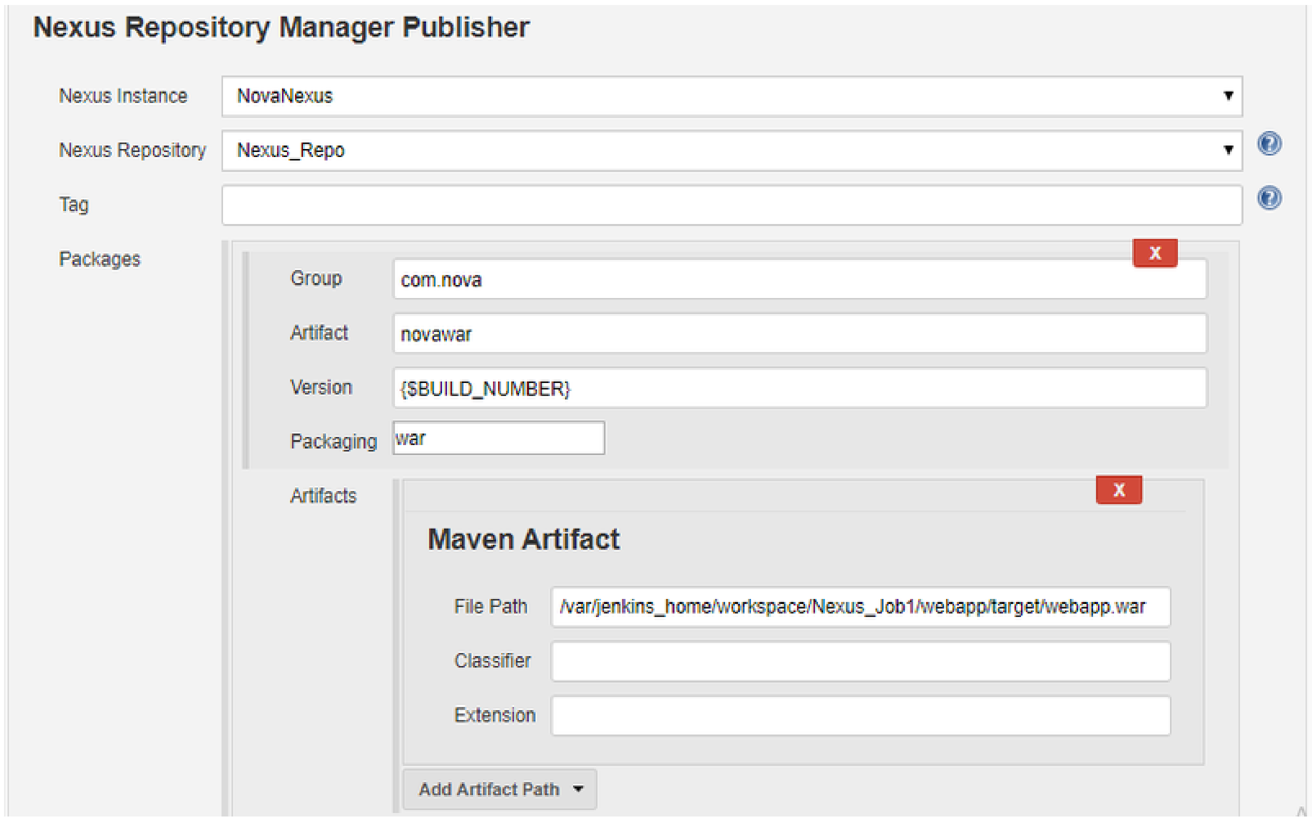
Invoke Artifactory Maven 3

Maven Version: maven

Root POM: pom.xml

Goals and options: clean package

Below mentioned screen shot is showing how to push artifacts on Nexus server.



Nexus Repository Manager Publisher

Nexus Instance	NovaNexus
Nexus Repository	Nexus_Repo
Tag	
Packages	
Group	com.nova
Artifact	novawar
Version	{$BUILD_NUMBER}
Packaging	war
Artifacts	Maven Artifact
File Path	/var/jenkins_home/workspace/Nexus_Job1/webapp/target/webapp.war
Classifier	
Extension	

Nexus Instance: This is Sever ID which has been configured in step 2.

Nexus Repository: Name of the repository created in Nexus.

Package Group: Group needs to mention like com.nova; so it will create the folder structure in Nexus like com > nova.

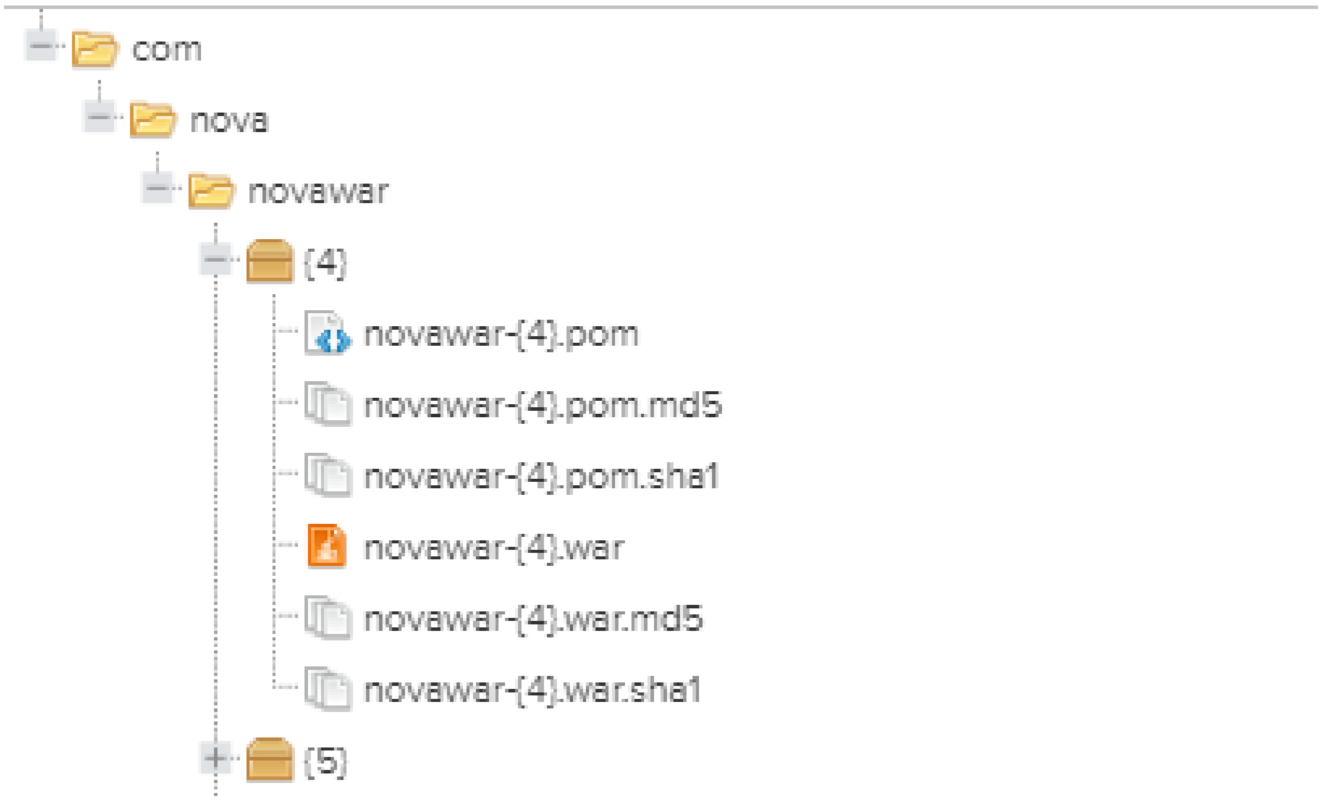
Package Artifact: Name of the artifacts for logical segregation for packages.

Package Version: Version strategy can be given accordingly however {$BUILD_NUMBER} given to upload the artifacts with all the Jenkins Builds.

Package Packaging: Type of artifacts can be mentioned here like war, tar, jar or ear.

Package Artifacts File Path: Need to mention the path; where artifact is generated after successful build. This same will be uploaded in Nexus server.

Above configuration or setting will create the folder structure in Nexus as below-



3.2 Through Jenkins File -

Below mentioned Jenkinsfile is an example which can be used for Pipeline.

3.2.1 Scripted JenkinsFile -

```
node ('slave1') {

  stage("POLL SCM") {
            git credentialsId: 'fefe1963-b4e0-4d09-bd5b-f8a95b934ce0',
url: 'giturl'
        }

        stage("Build") {
            withMaven(maven: 'maven-3',mavenSettingsConfig:
'my-maven-settings') {
         sh "mvn clean package
   }
        }

        stage("Push To Nexus"){
            nexusArtifactUploader artifacts: [[artifactId: 'com.nova',
classifier: '', file: '/home/jenkins/workspace/Nexus_Job1/webapp.war',
            type: 'war']], credentialsId: 'nexus', groupId: 'com.nova',
nexusUrl: 'novartis.devops.altimetrik.io:8082', nexusVersion: 'nexus3',
protocol:
           'http', repository: 'NexusRepo', version: '{$BUILD_NUMBER}'
        }
}
```

3.2.2 Declarative JenkinsFile -

```
pipeline {
    agent {label 'slave1'}
    stages {
        stage("POLL SCM") {
            git credentialsId: 'fefe1963-b4e0-4d09-bd5b-f8a95b934ce0',
url: 'giturl'
        }

        stage("Build") {
            withMaven(maven: 'maven-3',mavenSettingsConfig:
'my-maven-settings') {
         sh "mvn clean package
   }
        }

        stage("Push To Nexus"){
            nexusArtifactUploader artifacts: [[artifactId: 'com.nova',
classifier: '', file: '/home/jenkins/workspace/Nexus_Job1/webapp.war',
            type: 'war']], credentialsId: 'nexus', groupId: 'com.nova',
nexusUrl: 'novartis.devops.altimetrik.io:8082', nexusVersion: 'nexus3',
protocol:
           'http', repository: 'NexusRepo', version: '{$BUILD_NUMBER}'
        }
    }
}
```

3.2.3 Working Example with R-

```
node ('slave1') {

  stage("POLL SCM") {
            git credentialsId: 'fefe1963-b4e0-4d09-bd5b-f8a95b934ce0',
url: 'http://novartis.devops.altimetrik.io:7990/scm/rtes/r-mul.git'
        }

        stage("compile") {
            sh label: '', script: 'Rscript
/home/jenkins/workspace/Jenkins_pipeline_R/Rmul/R/test.R'
        }

        stage("Package"){
            sh label: '', script: 'R CMD build
/home/jenkins/workspace/Jenkins_pipeline_R/Rmul/R --save'
        }
        stage("Deploy"){
            nexusArtifactUploader artifacts: [[artifactId: 'com.nova',
classifier: '', file:
'/home/jenkins/workspace/Jenkins_pipeline_R/Rmul_0.1.0.tar.gz',
            type: 'tar.gz']], credentialsId: 'nexus', groupId:
'com.nova', nexusUrl: 'novartis.devops.altimetrik.io:8082',
nexusVersion: 'nexus3', protocol:
            'http', repository: 'Rproject', version: '{$BUILD_NUMBER}'
        }
}
```