Create and build jobs using Jenkins as well as upload artifacts to nexus and deploy that artifact to tomcat server

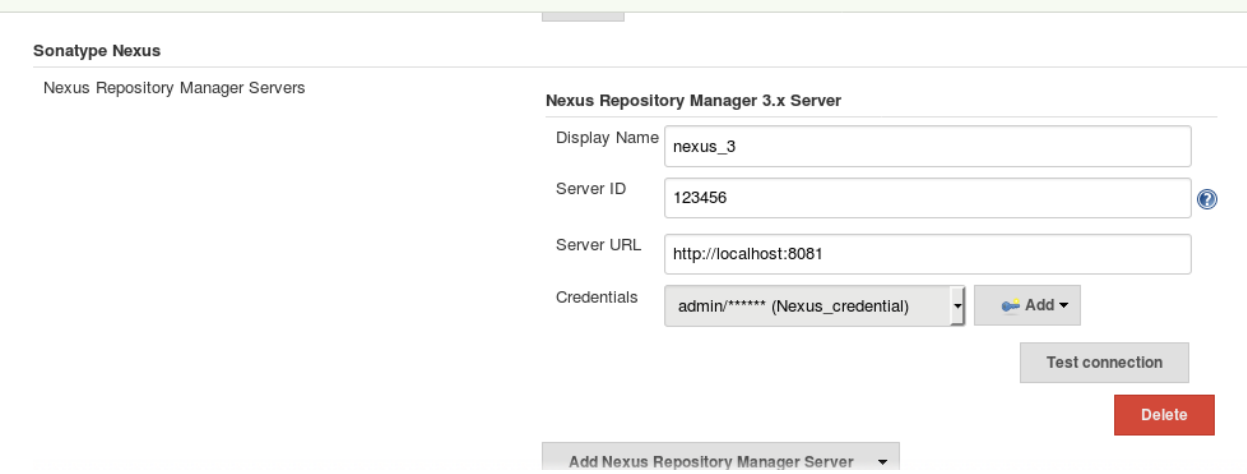
Step-1 Configure Jenkins with nexus. Plugins needed –

- ✓ Nexus Artifact Uploader
- ✓ Nexus Platform Plugin
- ✓ Build with parameters
- ✓ Copy artifact plugin
- ✓ Deploy to container plugin
- ✓ JUnit plugin
- ✓ Parameterized trigger plugin

Step-2 Configure nexus in Jenkins configure system, search for sonatype nexus

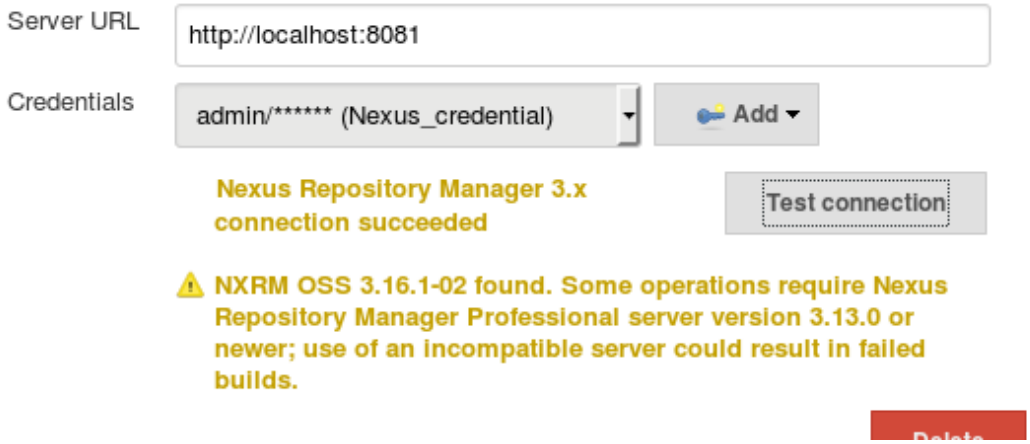
Select the Nexus Repository Manager Version

Then click on Add Nexus IQ Server, Enter the display name, ServerID, Server URL as http://<nexus-server-url>:8081



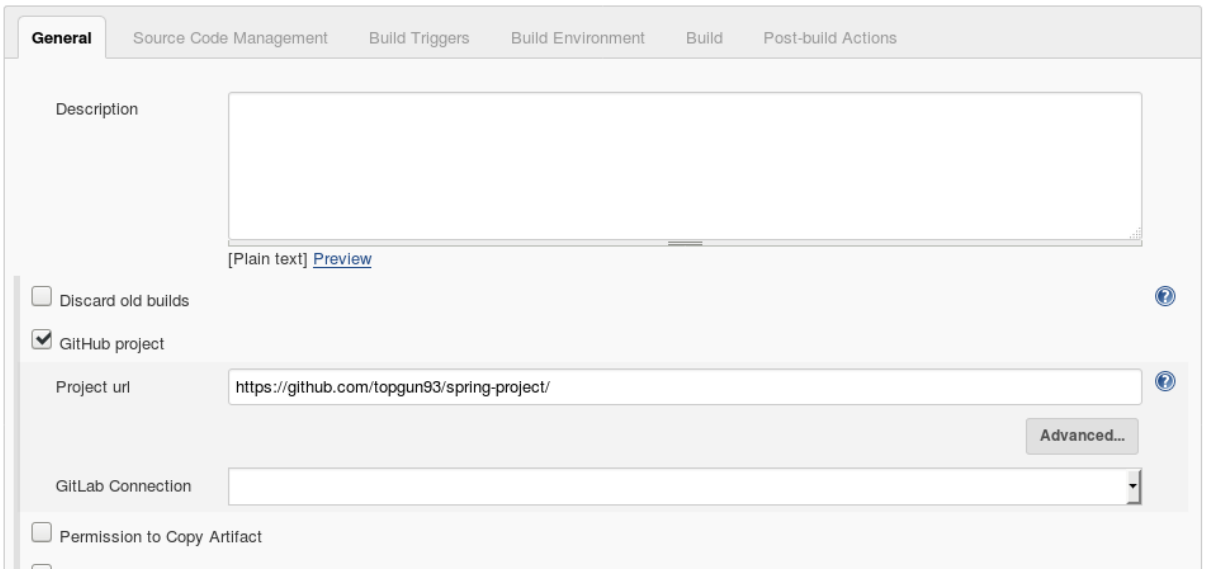
Enter credentials for nexus server.

To confirm, if its correctly configured click Test connection.

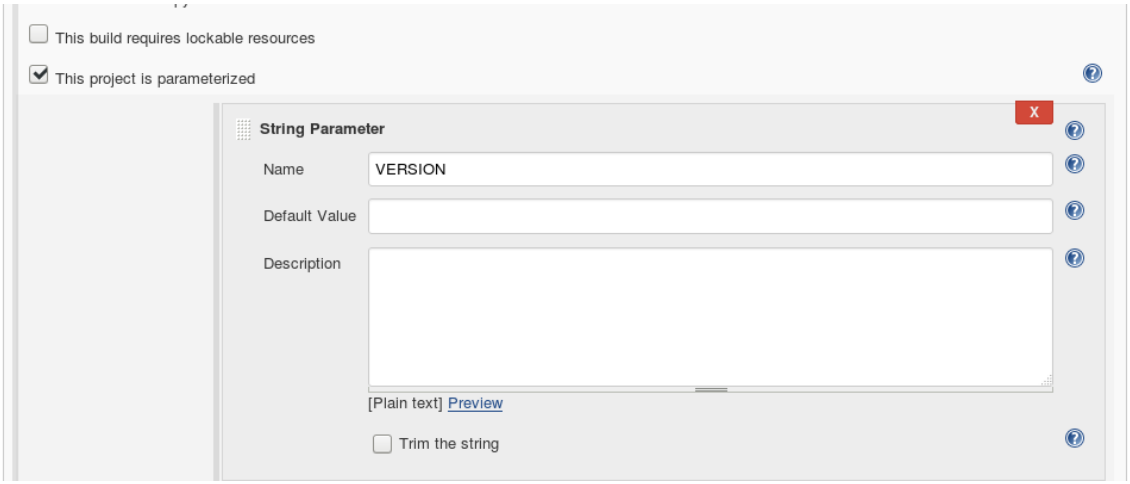


It should show as shown above.

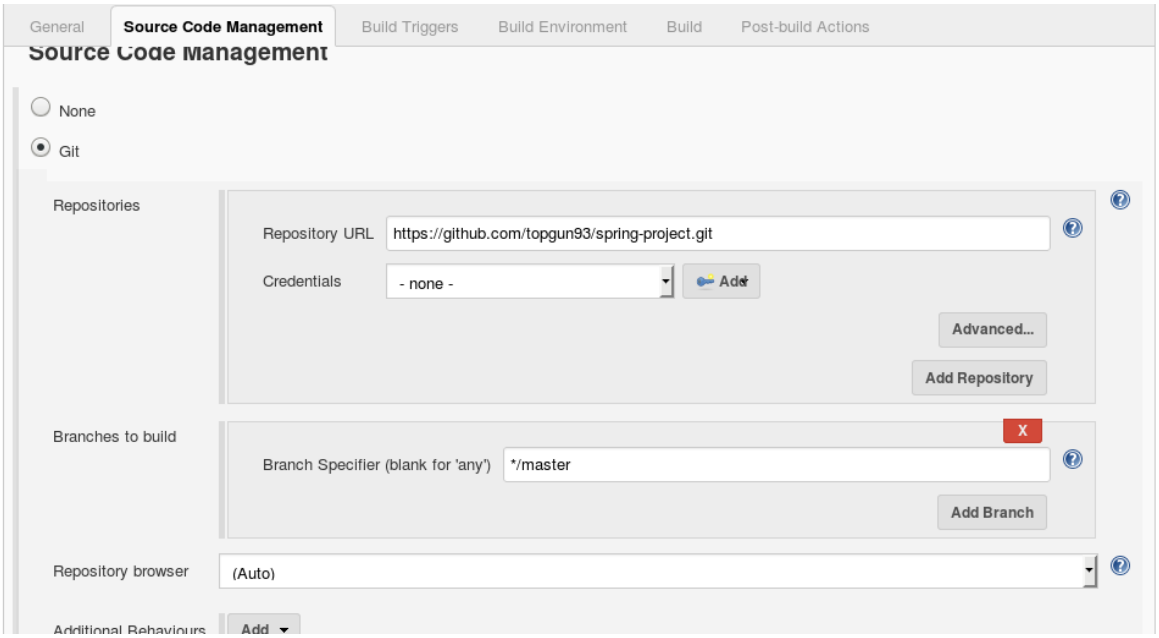
Step-3 Create new job,



Select GitHub project and enter the url there.



Since, we are going to pass parameters to the project based on the version numbers, select This project is parameterized, select string parameter.

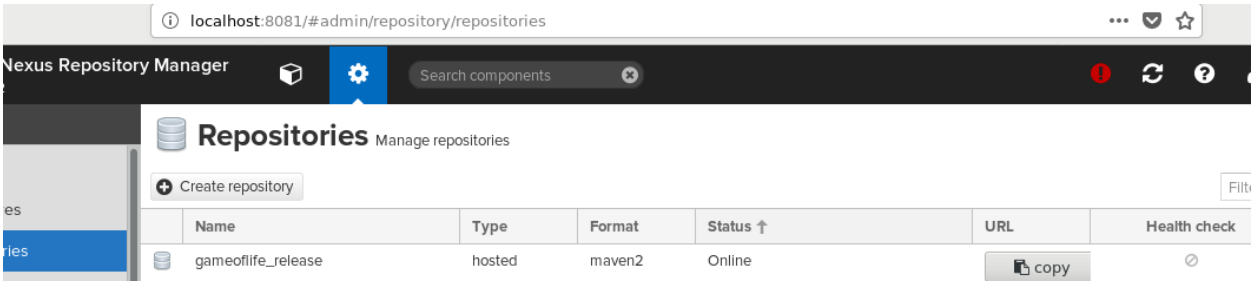


in Source code management, select git and enter .git url.

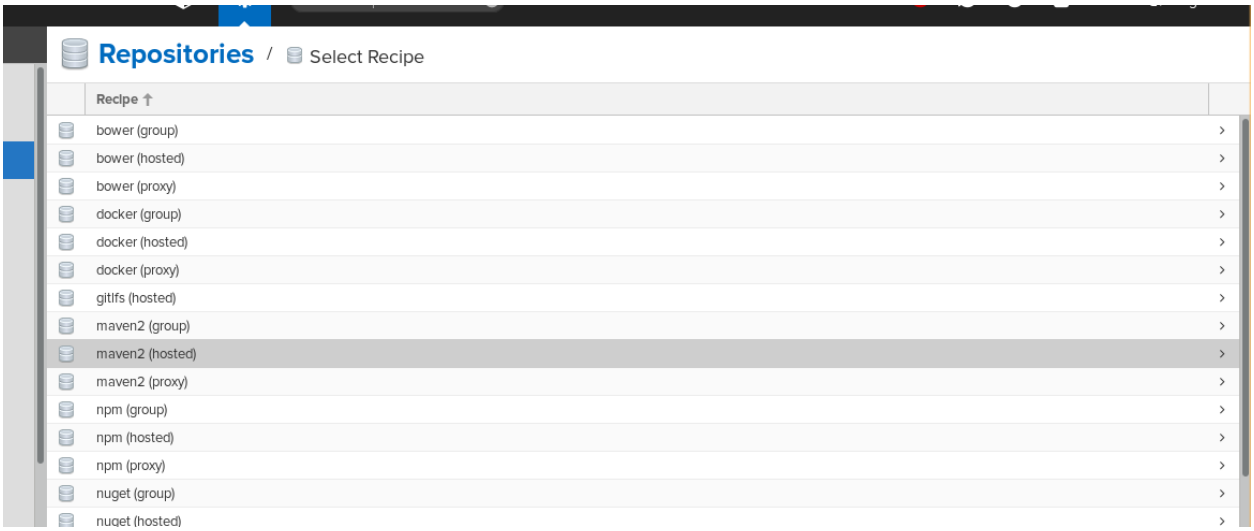
In build select Invoke top level Maven Targets, select the Maven Version and enter the Goals.

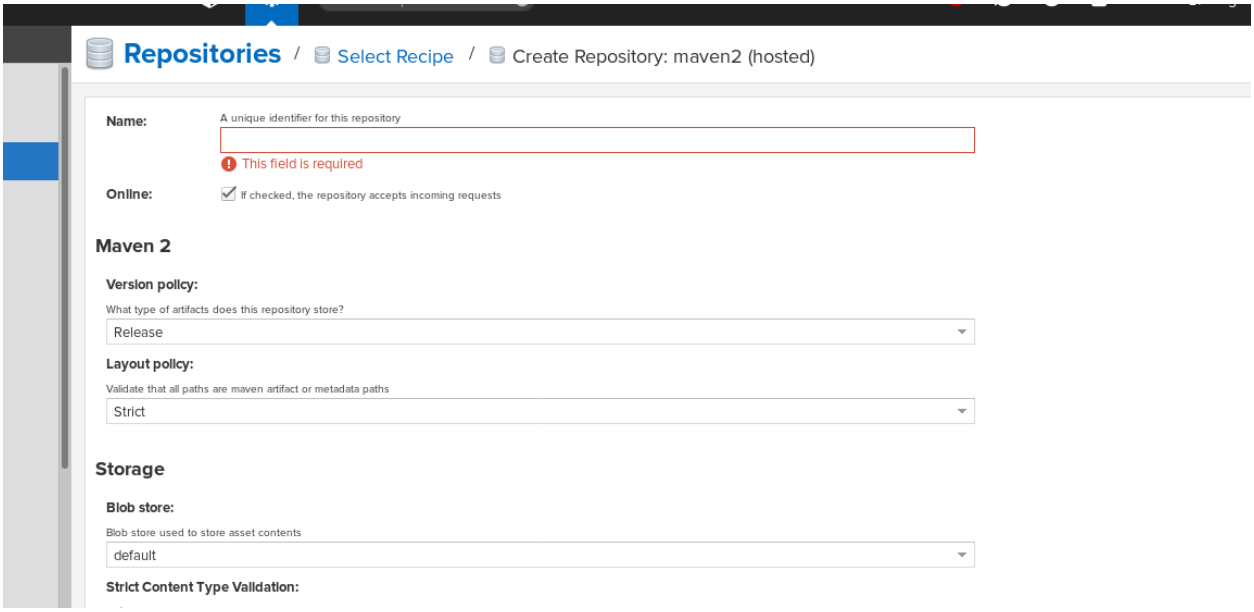
Step-4 Upload artifacts to Nexus server,

In nexus Create new repository,

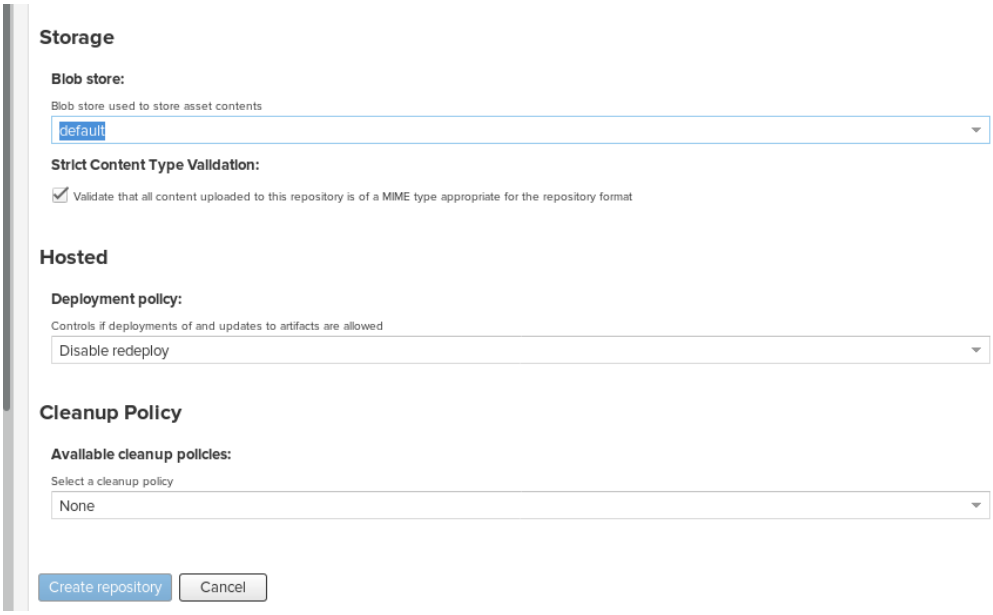


Select Recipe, since we are going to release artifact and upload it, select maven2(hosted).

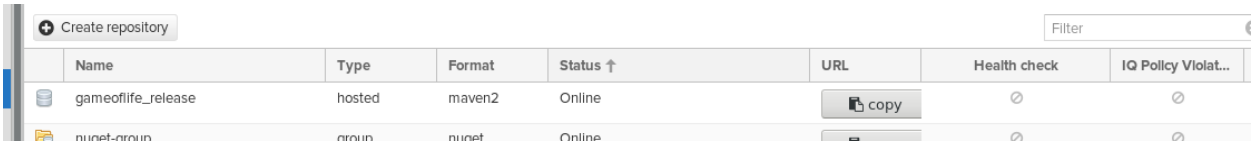

Repositories / Select Recipe
Recipe
bower (group)
bower (hosted)
bower (proxy)
docker (group)
docker (hosted)
docker (proxy)
gitlfs (hosted)
maven2 (group)
maven2 (hosted)
maven2 (proxy)
npm (group)
npm (hosted)
npm (proxy)
nuget (group)
nuget (hosted)


Repositories / Select Recipe / Create Repository: maven2 (hosted)
Name:
A unique identifier for this repository
This field is required
Online:
If checked, the repository accepts incoming requests
Maven 2
Version policy:
What type of artifacts does this repository store?
Release
Layout policy:
Validate that all paths are maven artifact or metadata paths
Strict
Storage
Blob store:
Blob store used to store asset contents
default
Strict Content Type Validation:

Enter the name of repository, select version policy as Release/Snapshot/mixed,

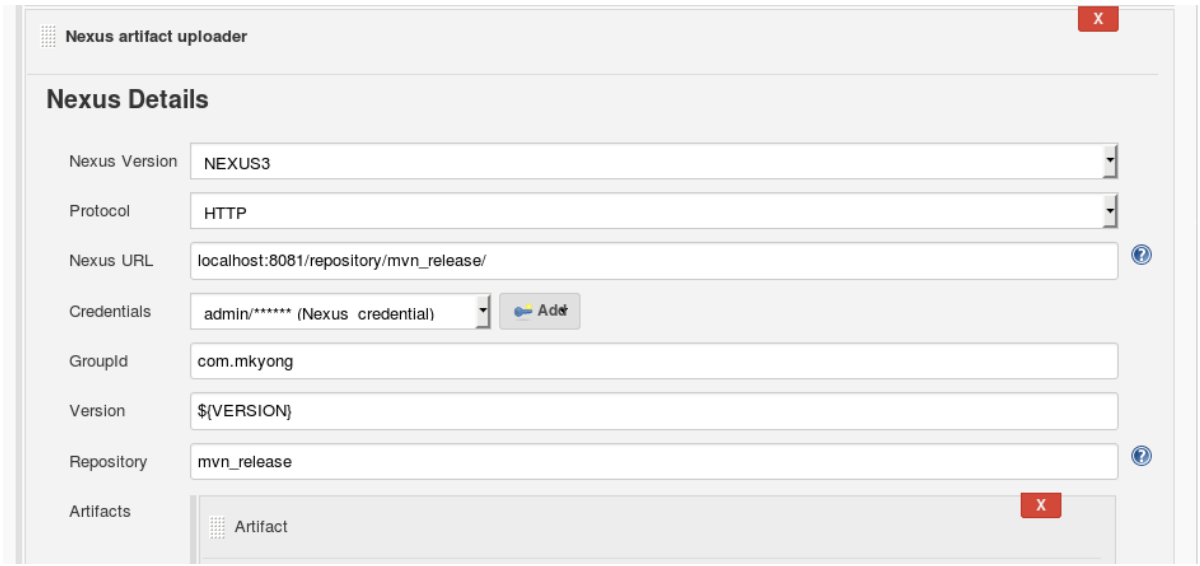


Click on Create Repository.

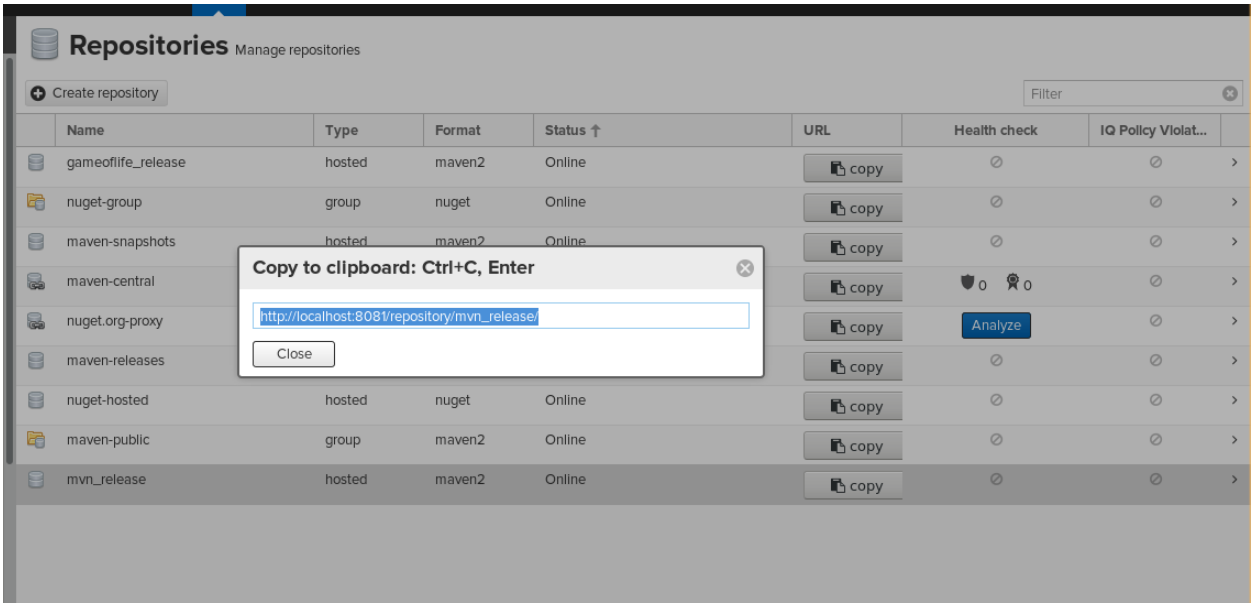


You can see the name of the repository.

Step-5 In Jenkins , in build section select Nexus artifact Uploader,



Here, select the Nexus Version, Enter the Nexus URL which you can get from nexus repository, in nexus, select the repository which you have created



In the same line you will find copy button, click on that and you will see a popup with the url of that repo, here I have selected mvn_release, copy the url.

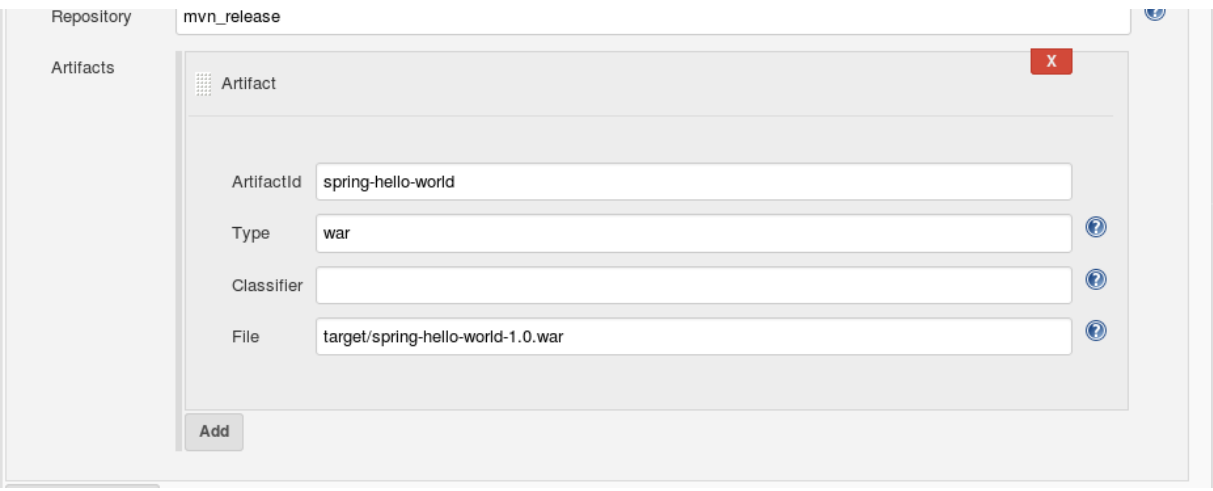
In jenkins prviously shown build step enter the copied nexus url and select credentials.

Enter the groupid which you can get from pom.xml of that repo

```
<modelVersion>4.0.0</modelVersion>
<groupId>com.mkyong</groupId>
<artifactId>spring-hello-world</artifactId>
<packaging>war</packaging>
<version>1.0</version>
<name>spring3 mvc maven</name>
```

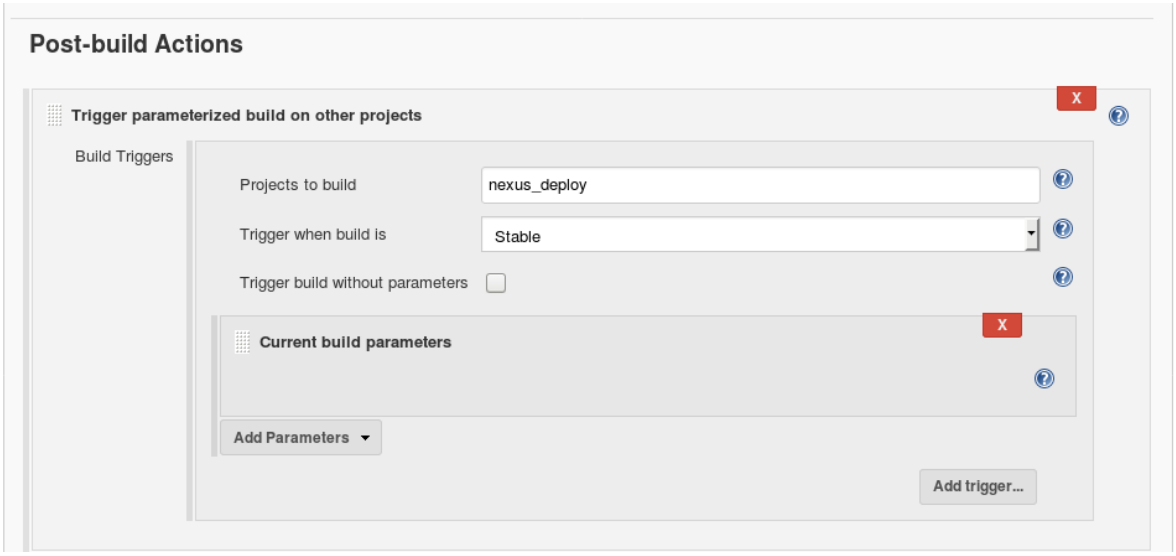
In Version enter the parameter variable as we are going to give parameter while building the job, enter the repository name mvn_release as I have created repo with that name.

In artifact select add



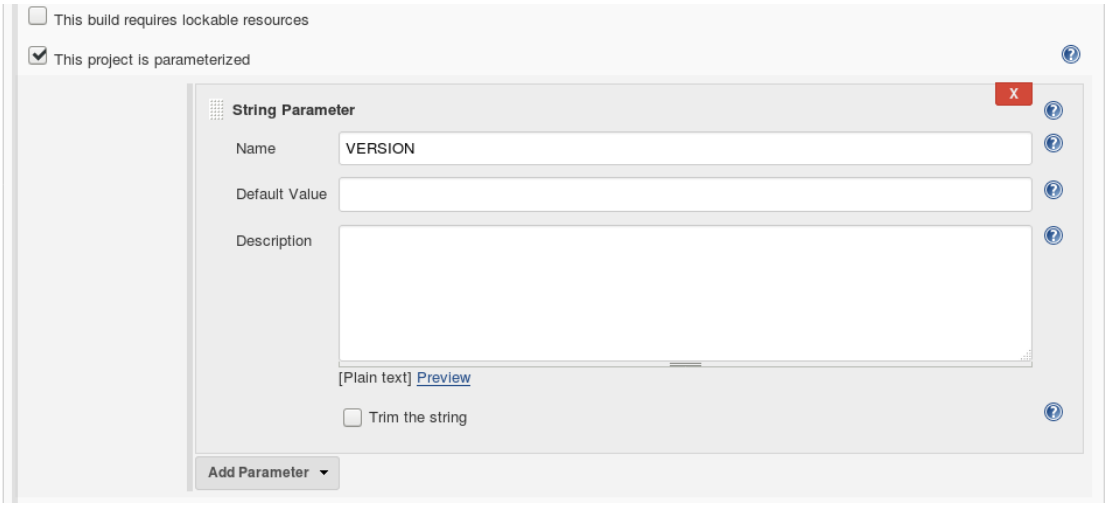
enter artifact id from pom.xml, enter the type of artifact such as war/jar, enter the File location where that extension file is present.

Now, to deploy that repository, we will pass that parameter to other job which will run in pipeline.



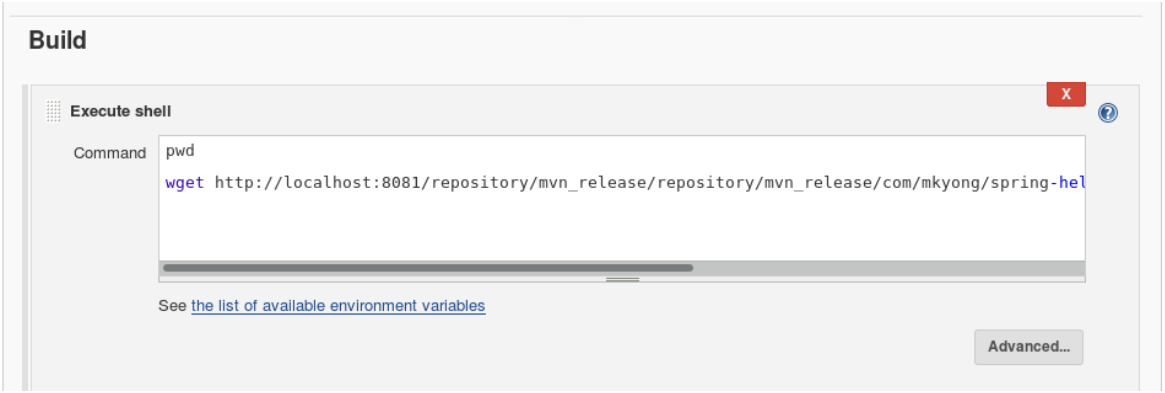
In post build actions, select Trigger parameterized build on other projects, Enter the name of project to build, select the parameter type as Current build parameters. Click apply and save.

Step-6 Create another job



Select this project is parameterized

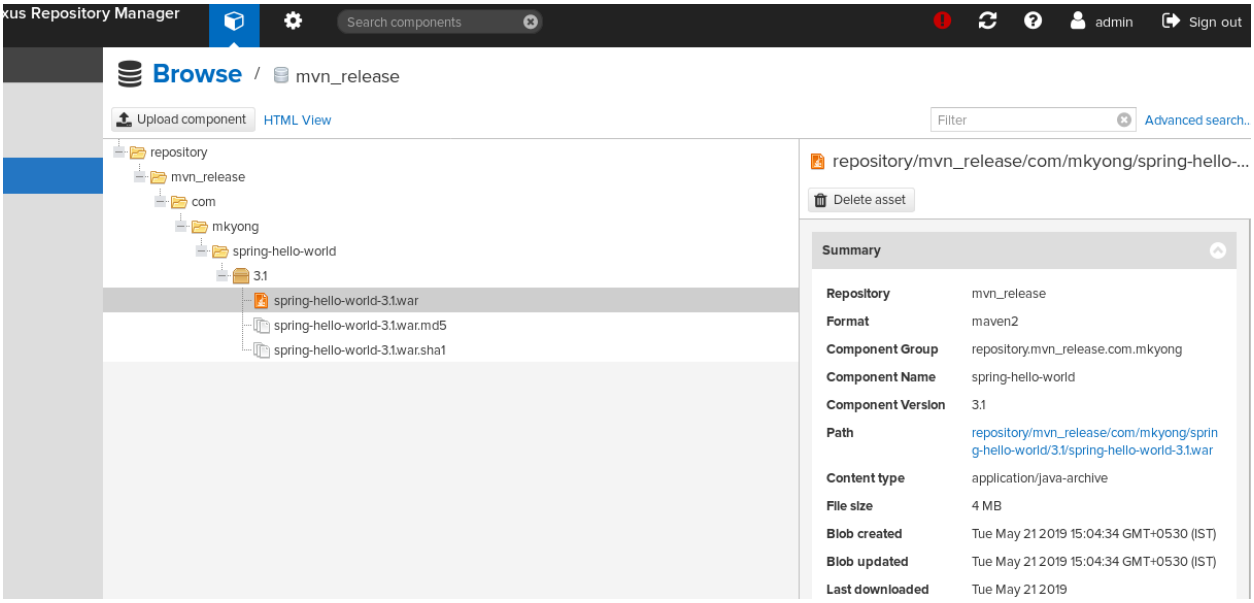
In build select execute shell



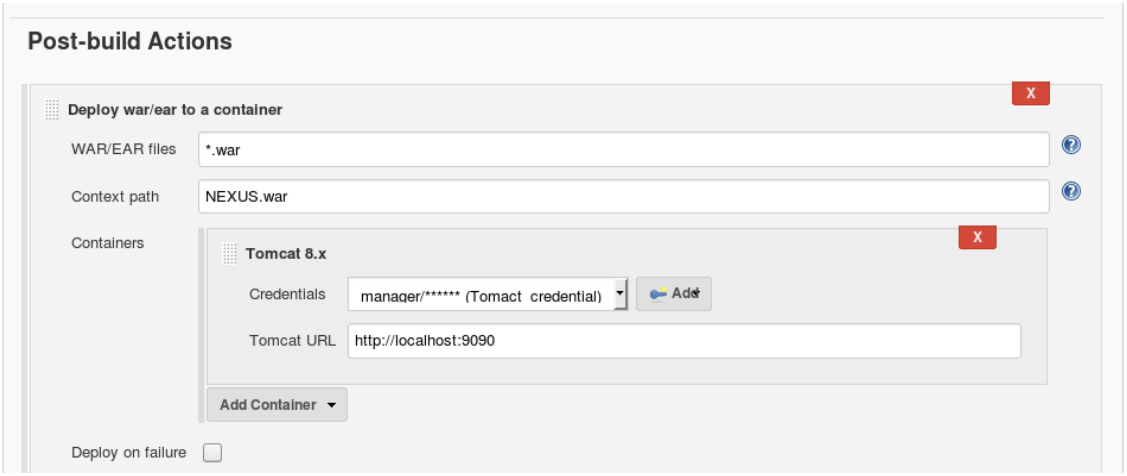
Run the command wget http://localhost:8081/repository/mvn_release/repository/mvn_release/com/mkyong/spring-hello-world/${VERSION}/spring-hello-world-${VERSION}.war

This job is parameterized so based on the given versio number it will download the war file.

This link can be found in the repository



To deploy it in tomcat, select Deploy war/ear to a container,



Enter the war/ear file, in container select the tomcat version its credential and tomcat url.

Click Apply and save.

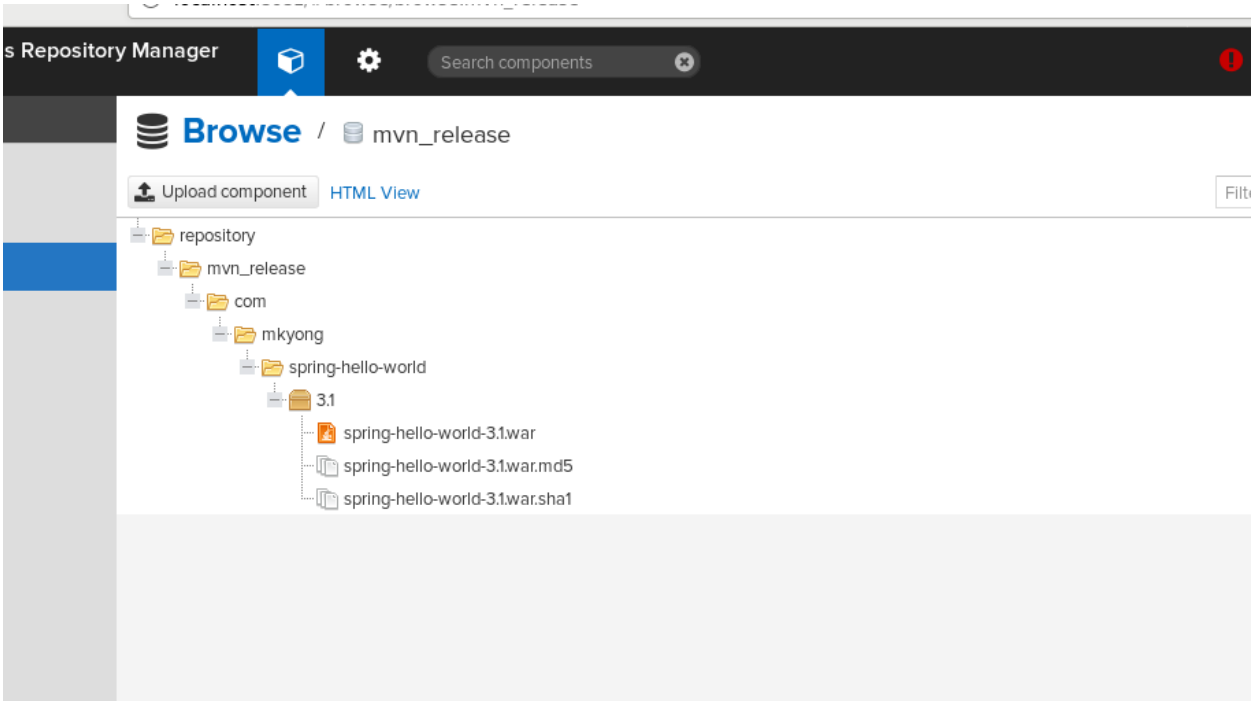
Run the job.

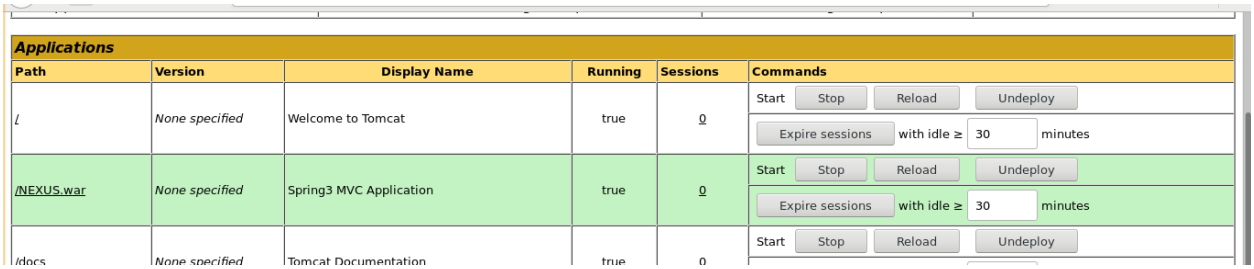
```
[INFO] Total time:  12.582 s
[INFO] Finished at: 2019-05-21T15:04:31+05:30
[INFO] ------------------------------------------------------------------------
Uploading artifact spring-hello-world-1.0.war started....
GroupId: com.mkyong
ArtifactId: spring-hello-world
Classifier:
Type: war
Version: 3.1
File: spring-hello-world-1.0.war
Repository:mvn_release
Uploading: http://localhost:8081/repository/mvn_release//repository/mvn_release/com/mkyong/spring-hello-world/3.1
/spring-hello-world-3.1.war
10 % completed (418 kB / 4.2 MB).
20 % completed (836 kB / 4.2 MB).
30 % completed (1.3 MB / 4.2 MB).
40 % completed (1.7 MB / 4.2 MB).
50 % completed (2.1 MB / 4.2 MB).
60 % completed (2.5 MB / 4.2 MB).
70 % completed (2.9 MB / 4.2 MB).
80 % completed (3.3 MB / 4.2 MB).
90 % completed (3.8 MB / 4.2 MB).
100 % completed (4.2 MB / 4.2 MB).
Uploaded: http://localhost:8081/repository/mvn_release//repository/mvn_release/com/mkyong/spring-hello-world/3.1/spring-
hello-world-3.1.war (4.2 MB at 885 kB/s)
Uploading artifact spring-hello-world-1.0.war completed.
Triggering a new build of nexus_deploy
Finished: SUCCESS
```

```
3100K .......... .......... .......... .......... .......... 77% 10.4M 0s
3150K .......... .......... .......... .......... .......... 78% 76.9M 0s
3200K .......... .......... .......... .......... .......... 79% 58.1M 0s
3250K .......... .......... .......... .......... .......... 81%  661M 0s
3300K .......... .......... .......... .......... .......... 82% 66.9M 0s
3350K .......... .......... .......... .......... .......... 83%  669M 0s
3400K .......... .......... .......... .......... .......... 84%  692M 0s
3450K .......... .......... .......... .......... .......... 86%  679M 0s
3500K .......... .......... .......... .......... .......... 87%  697M 0s
3550K .......... .......... .......... .......... .......... 88%  731M 0s
3600K .......... .......... .......... .......... .......... 89%  523M 0s
3650K .......... .......... .......... .......... .......... 90%  716M 0s
3700K .......... .......... .......... .......... .......... 92%  653M 0s
3750K .......... .......... .......... .......... .......... 93%  669M 0s
3800K .......... .......... .......... .......... .......... 94%  719M 0s
3850K .......... .......... .......... .......... .......... 95%  674M 0s
3900K .......... .......... .......... .......... .......... 97%  678M 0s
3950K .......... .......... .......... .......... .......... 98%  617M 0s
4000K .......... .......... .......... .......... .......... 99%  503M 0s
4050K .......... ........                                  100%  647M=0.05s

2019-05-21 15:04:43 (76.3 MB/s) - “spring-hello-world-3.1.war” saved [4166010/4166010]

Deploying /var/lib/jenkins/workspace/nexus_deploy/spring-hello-world-3.1.war to container Tomcat 8.x Remote with context
NEXUS.war
  [/var/lib/jenkins/workspace/nexus_deploy/spring-hello-world-3.1.war] is not deployed. Doing a fresh deployment.
  Deploying [/var/lib/jenkins/workspace/nexus_deploy/spring-hello-world-3.1.war]
Finished: SUCCESS
```





You will see the output as above.