

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

Nexus IQ Server

☐ Hide messages about what's coming to the Nexus Platform Plugin

Add Nexus Repository Manager Server ▼

- Nexus Repository Manager 2.x Server
- Nexus Repository Manager 3.x Server

Select the Nexus Repository Manager Version

Nexus IQ Server

Add Nexus IQ Server ▼

☐ Hide messages about what's coming to the Nexus Platform Plugin

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

The screenshot shows the 'Sonatype Nexus' interface with a tab for 'Nexus Repository Manager Servers'. A form titled 'Nexus Repository Manager 3.x Server' is displayed with the following fields: 'Display Name' (nexus_3), 'Server ID' (123456), 'Server URL' (http://localhost:8081), and 'Credentials' (admin/***** (Nexus_credential)). There is an 'Add' button next to the credentials field. Below the form are 'Test connection' and 'Delete' buttons. At the bottom, there is a button labeled 'Add Nexus Repository Manager Server'.

Enter credentials for nexus server.

To confirm, if its correctly configured click Test connection.

This screenshot shows a zoomed-in view of the 'Test connection' button and the success message: 'Nexus Repository Manager 3.x connection succeeded'. Below the message is a warning icon and text: 'NXRM OSS 3.16.1-02 found. Some operations require Nexus Repository Manager Professional server version 3.13.0 or newer; use of an incompatible server could result in failed builds.' The 'Delete' button is also visible at the bottom right.

It should show as shown above.

Step-3 Create new job,

The screenshot shows the 'Create new job' configuration page in Jenkins, specifically the 'General' tab. The tabs at the top are: General, Source Code Management, Build Triggers, Build Environment, Build, and Post-build Actions. The 'Description' field is empty. Below it, there is a '[Plain text] Preview' link. The 'Discard old builds' checkbox is unchecked. The 'GitHub project' checkbox is checked. The 'Project url' field contains the text 'https://github.com/topgun93/spring-project/'. To the right of this field is an 'Advanced...' button. The 'GitLab Connection' dropdown menu is empty. The 'Permission to Copy Artifact' checkbox is unchecked.

Select GitHub project and enter the url there.

The screenshot shows the 'String Parameter' section of the Jenkins configuration page. The 'This build requires lockable resources' checkbox is unchecked. The 'This project is parameterized' checkbox is checked. The 'String Parameter' section has a red 'X' icon in the top right corner. It contains three fields: 'Name' with the value 'VERSION', 'Default Value' which is empty, and 'Description' which is empty. Below these fields is a '[Plain text] Preview' link. At the bottom of the section is a 'Trim the string' checkbox which is unchecked.

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

General **Source Code Management** Build Triggers Build Environment Build Post-build Actions

Source Code Management

☐ None
☒ Git

Repositories

Repository URL

Credentials

Branches to build

Branch Specifier (blank for 'any')

Repository browser

Additional Behaviours

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

Build

☒ Invoke top-level Maven targets

Maven Version

Goals

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,

Nexus Repository Manager

localhost:8081/#admin/repository/repositories

Repositories Manage repositories

Name	Type	Format	Status ↑	URL	Health check
gameoflife_release	hosted	maven2	Online	<input type="button" value="copy"/>	<input type="button" value="refresh"/>

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)	>
gitifs (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,

Storage

Blob store:
Blob store used to store asset contents
default

Strict Content Type Validation:
☒ Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format

Hosted

Deployment policy:
Controls if deployments of and updates to artifacts are allowed
Disable redeploy

Cleanup Policy

Available cleanup policies:
Select a cleanup policy
None

Create repository Cancel

Click on Create Repository.

Create repository Filter

	Name	Type	Format	Status ↑	URL	Health check	IQ Policy Violat...
	gameoflife_release	hosted	maven2	Online	copy		
	nexus-nexus	nexus	nexus	Online			

You can see the name of the repository.

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

Nexus artifact uploader

Nexus Details

Nexus Version: NEXUS3

Protocol: HTTP

Nexus URL: localhost:8081/repository/mvn_release/

Credentials: admin/***** (Nexus credential) Add

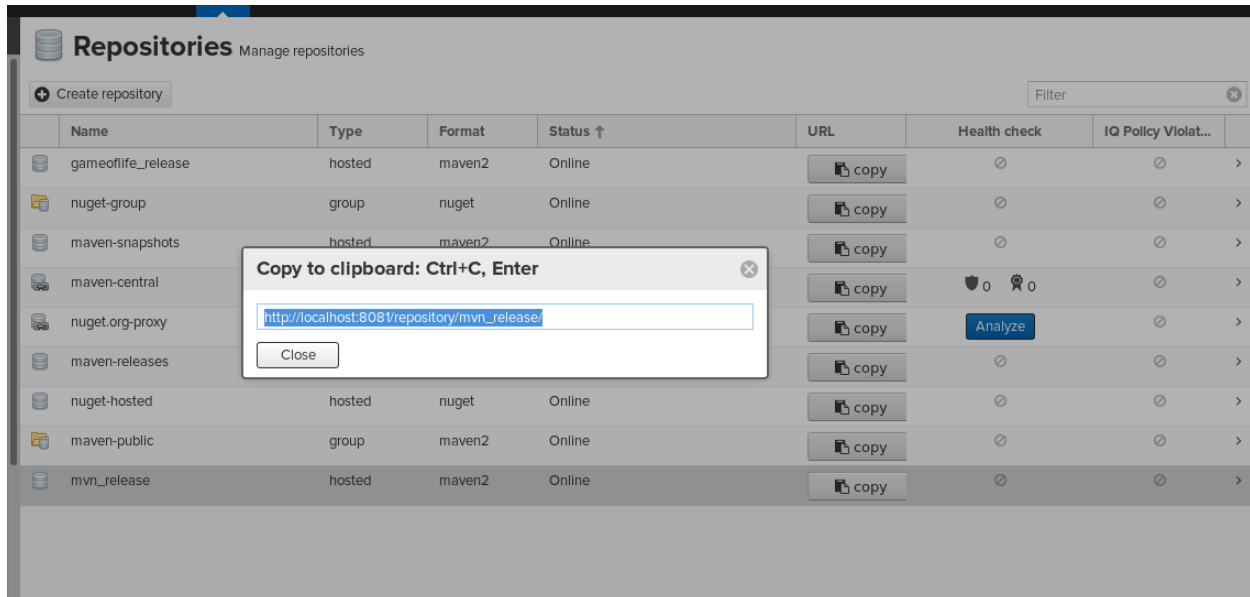
GroupId: com.mkyyong

Version: \${VERSION}

Repository: mvn_release

Artifacts: Artifact

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 previously 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

Repository: mvn_release

Artifacts

Artifact

ArtifactId: spring-hello-world

Type: war

Classifier:

File: target/spring-hello-world-1.0.war

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.

Post-build Actions

Trigger parameterized build on other projects

Build Triggers

Projects to build: nexus_deploy

Trigger when build is: Stable

Trigger build without parameters: ☐

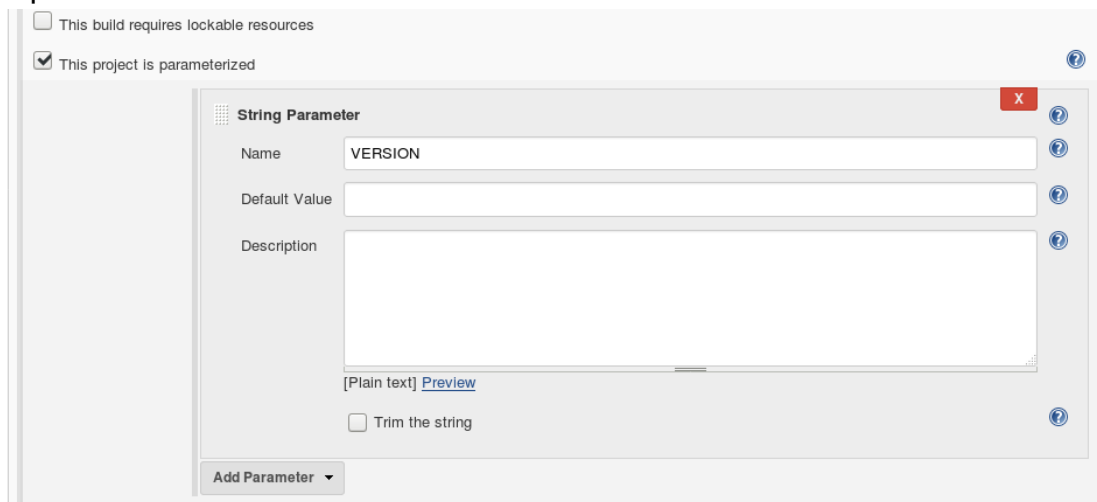
Current build parameters

Add Parameters

Add trigger...

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.

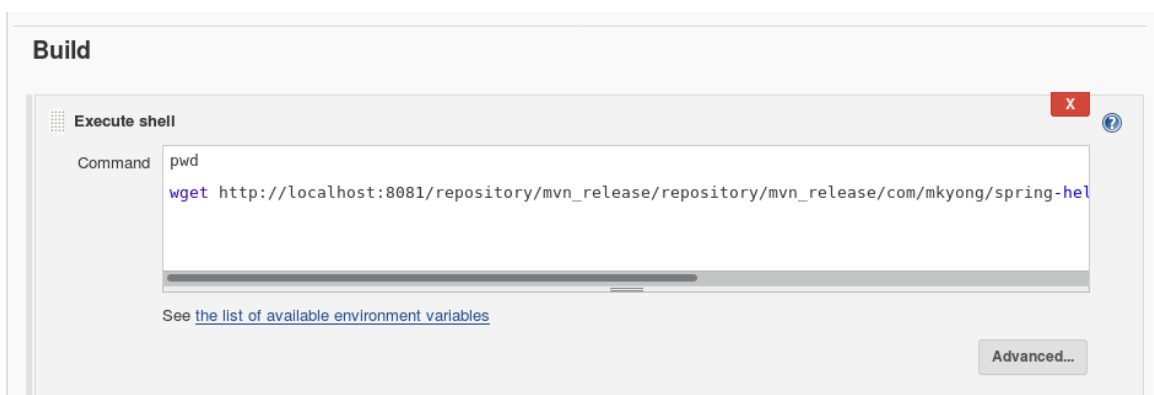
Step-6 Create another job



The screenshot shows the Jenkins 'String Parameter' configuration interface. At the top, there are two checkboxes: 'This build requires lockable resources' (unchecked) and 'This project is parameterized' (checked). Below these is a 'String Parameter' section with a red 'X' icon and a help icon. It contains three input fields: 'Name' with the value 'VERSION', 'Default Value' (empty), and 'Description' (empty). Below the description field is a '[Plain text] Preview' link and a 'Trim the string' checkbox (unchecked). At the bottom left is an 'Add Parameter' dropdown button.

Select this project is parameterized

In build select execute shell

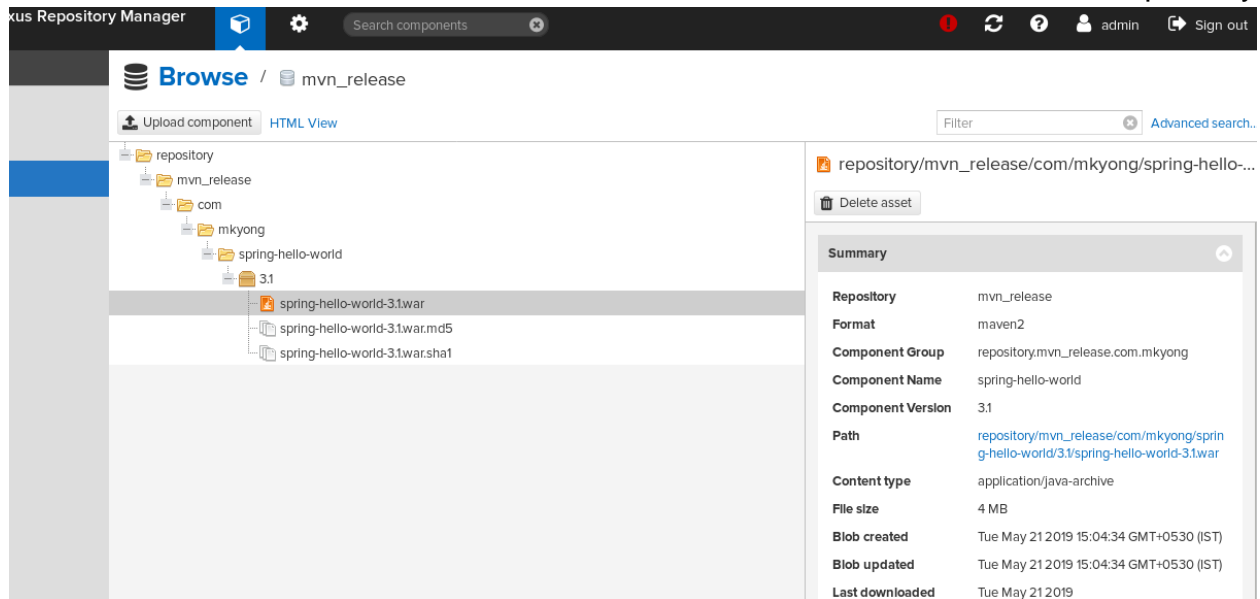


The screenshot shows the Jenkins 'Build' configuration interface. It features a 'Build' section with a red 'X' icon and a help icon. Inside, there is an 'Execute shell' step with a 'Command' input field. The command is: `pwd`
`wget http://localhost:8081/repository/mvn_release/repository/mvn_release/com/mkyong/spring-hello-world/${VERSION}/spring-hello-world-${VERSION}.war`. Below the command field is a link to 'See the list of available environment variables' and an 'Advanced...' button.

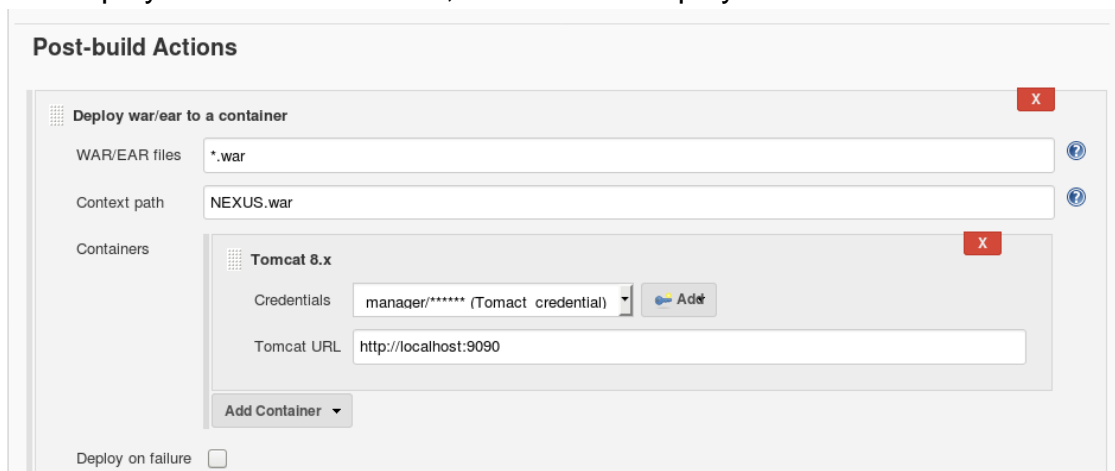
Run the command `wget http://localhost:8081/repository/mvn_release/repository/mvn_release/com/mkyong/spring-hello-world/${VERSION}/spring-hello-world-${VERSION}.war`

This job is parameterized so based on the given version 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,



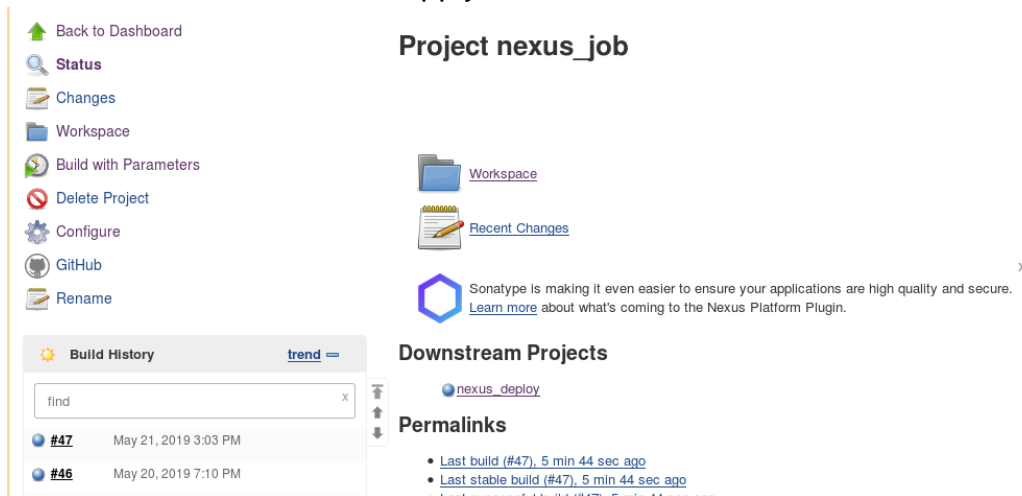
Enter the war/ear file, in container select the tomcat version its credential and tomcat url.

Click

Apply

and

save.



The screenshot shows the Sonatype Nexus web interface for a project named **Project nexus_job**. On the left is a sidebar with navigation links: [Back to Dashboard](#), [Status](#), [Changes](#), [Workspace](#), [Build with Parameters](#), [Delete Project](#), [Configure](#), [GitHub](#), and [Rename](#). The main content area has a header with the project name and a **Workspace** link. Below this is a **Recent Changes** section with a Sonatype logo and a message: "Sonatype is making it even easier to ensure your applications are high quality and secure. [Learn more](#) about what's coming to the Nexus Platform Plugin." There is a close button (X) next to this message. The **Build History** section is visible, showing a table with two entries: build #47 on May 21, 2019 at 3:03 PM, and build #46 on May 20, 2019 at 7:10 PM. Below the build history is a **Downstream Projects** section showing a link to [nexus_deploy](#). At the bottom is a **Permalinks** section with a list of links: [Last build \(#47\), 5 min 44 sec ago](#), [Last stable build \(#47\), 5 min 44 sec ago](#), and [Last successful build \(#47\), 5 min 44 sec ago](#).

Run the job.

```

[INFO] Total time: 12.502 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	

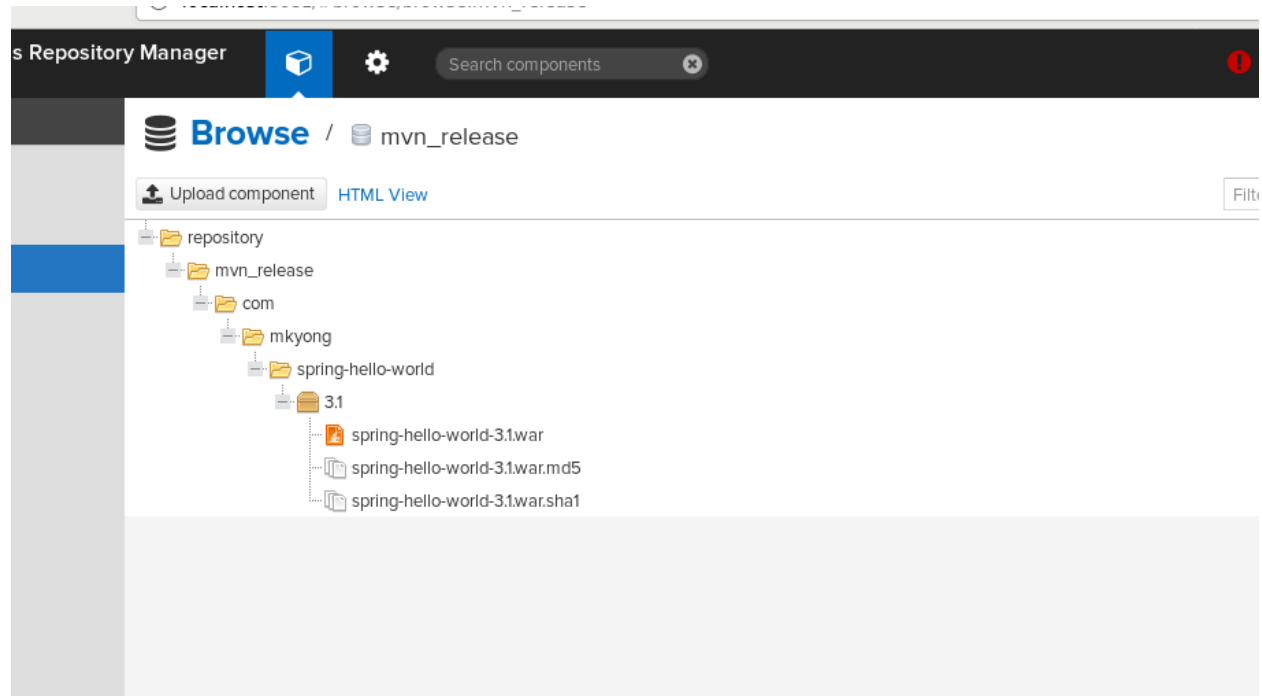
2019-05-21 15:04:43 (76.3 MB/s) - "spring-hello-world-3.1.war" saved [4166010/4166010]

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



Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle <input type="text" value="30"/> minutes</div>
/NEXUS.war	None specified	Spring3 MVC Application	true	0	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle <input type="text" value="30"/> minutes</div>
/docs	None specified	Tomcat Documentation	true	0	<div>Start Stop Reload Undeploy</div>

You will see the output as above.