Nexus Repository Manager

Nexus

Repository Manager:

Repository manager allows stores and retrieves the build artifact (Packages and Libraries).

Ex: Sonatype Nexus
Jfrog Artifactory

Nexus:

Nexus is a repository/Artifactory manager, it allows stores and retrieves the build artifacts. A Nexus installation brings you a repository for your company. So we can host own repositories and also use Nexus as a proxy for public repositories. It's allows to host private build artifacts.
Nexus is available as commercial and open source distribution.

Advantages:

- Ability to Deploy 3rd-party Artifacts
- Ability to Host Internal Repositories
- Ability to Host Public Repositories
- Speedier Builds
- Control and Auditing
- Dead simple install (and since 1.2, dead simple upgrade, too)
- Very good web UI
- Faster and more reliable builds
- Improved collaboration
- Component usage visibility
- Enforce components standards
- Controlled sharing with partners
- Ideal repository for robust governance
- Easy to maintain, almost no administrative overhead
- Provides you with RSS feeds of recently installed, broken artifacts and errors
- It can group several repositories so you can mirror several sources but need only one or two entries in your settings.xml
- Deploying from Maven works out of the box (no need for WebDAV hacks, etc).
- You can redirect access paths (i.e. some broken pom.xml requires "a.b.c" from "xxx"). Instead of patching the POM, you can fix the bug in Nexus and redirect the request to the place where the artifact really is.

Disadvantages

- The dependency is lost in case the link changes (if no version is specified within the POM). You are forced to adapt to all changes that are made to the dependency.
- We are not using the latest version of dependencies. You have to manually update dependencies.
- You have to implement the automated update of the repositories.

Types Repositories

1. Proxy Repository
2. Hosted Repository
3. Group Repository

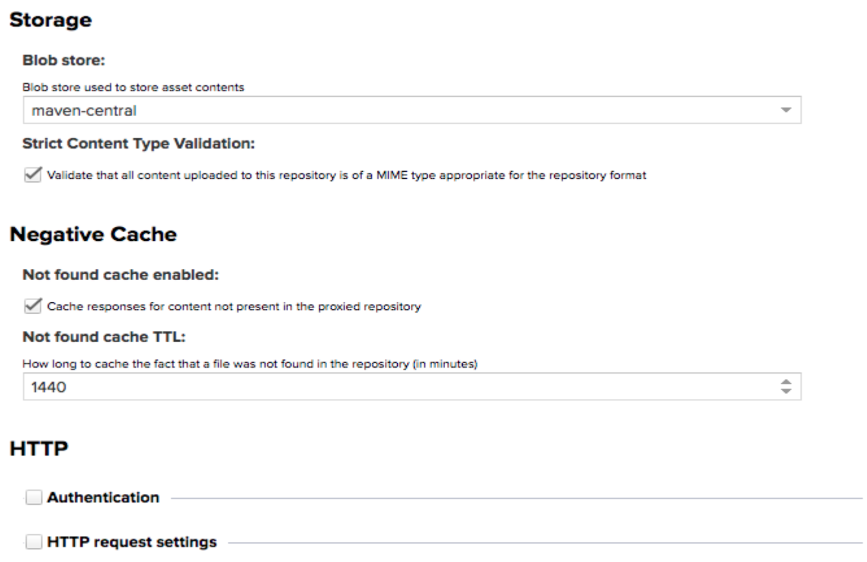
Proxy Repository

- Proxy Repository is a repository that is linked to a remote repository.
- A proxy repository refers to a remote repository. The initial request for a component is forwarded to the remote repository. The component is then retrieved and stored locally in the repository manager, which acts as a cache. Further requests for the same component are fulfilled from the local storage.

- Any request for a component is verified against the local content of the proxy repository. **If no local component is found, the request is forwarded to the remote repository.** The component is then retrieved and stored locally in the repository manager, which acts as a cache.
- Subsequent requests for the same component are then fulfilled from the local storage, therefore eliminating the network bandwidth and time overhead of retrieving the component from the remote repository again.
- By default, the repository manager ships with the following configured proxy repositories:

  a). maven-central - Proxy repository accesses the Central Repository, formerly known as Maven Central.
  b). nuget.org-proxy - This proxy repository accesses the NuGet Gallery. It is the default component repository used by the nuget package management tool used for .Net development.

-



Storage

Blob store:

Blob store used to store asset contents

maven-central

Strict Content Type Validation:

✓ Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format

Negative Cache

Not found cache enabled:

✓ Cache responses for content not present in the proxied repository

Not found cache TTL:

How long to cache the fact that a file was not found in the repository (in minutes)

1440

HTTP

Authentication

HTTP request settings

Hosted Repository

- Hosted Repository is a repository that stores components in the repository manager as the authorative location for those components.
- A repository with the type hosted, also known as a hosted repository, is a repository that **stores components in the repository manager as the authoritative location for these components.**
- By default, the repository manager ships with the following configured hosted repositories:

  a). maven-releases - This hosted repository uses the maven2 repository format with a release version policy. It is **intended to be the repository where your organization publishes internal releases**. You can also use this repository for third-party components that are not available in external repositories and can therefore not be retrieved via a configured proxy repository.
  b). maven-snapshots - Uses the maven2 repository format with a snapshot version policy. It is intended to be the the repository where your organization publishes internal development versions, also known as snapshots.
  c). nuget-hosted - It is intended to be the the repository where your organization can publish internal releases in repository using the NuGet repository format.

3rd Party

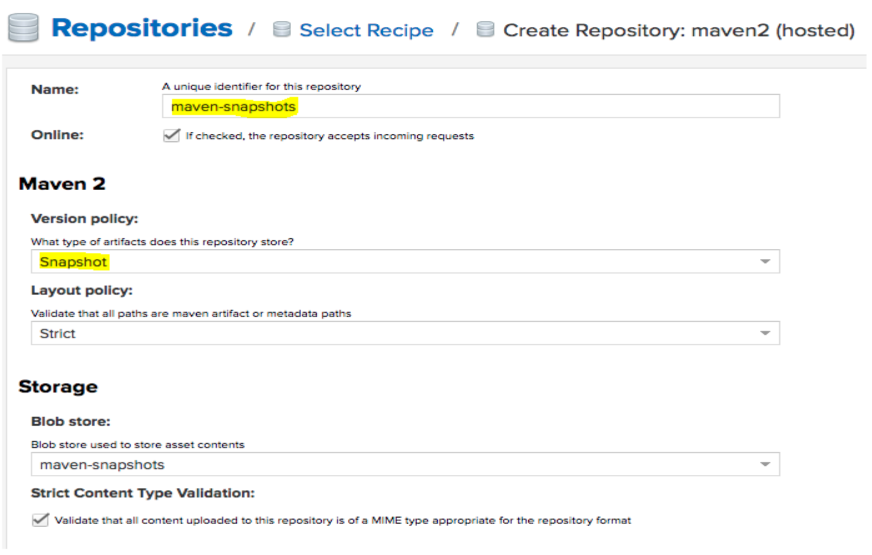
This hosted repository should be used for third-party dependencies not available in the public Maven repositories. Examples of these dependencies could be commercial, proprietary libraries such as an Oracle JDBC driver that may be referenced by your organization.

Releases

This hosted repository is where your organization will publish internal releases.

Snapshots

This hosted repository is where your organization will publish internal snapshots



Repositories / Select Recipe / Create Repository: maven2 (hosted)

Name: A unique identifier for this repository
maven-snapshots

Online: ✓ If checked, the repository accepts incoming requests

Maven 2

Version policy:
What type of artifacts does this repository store?
Snapshot

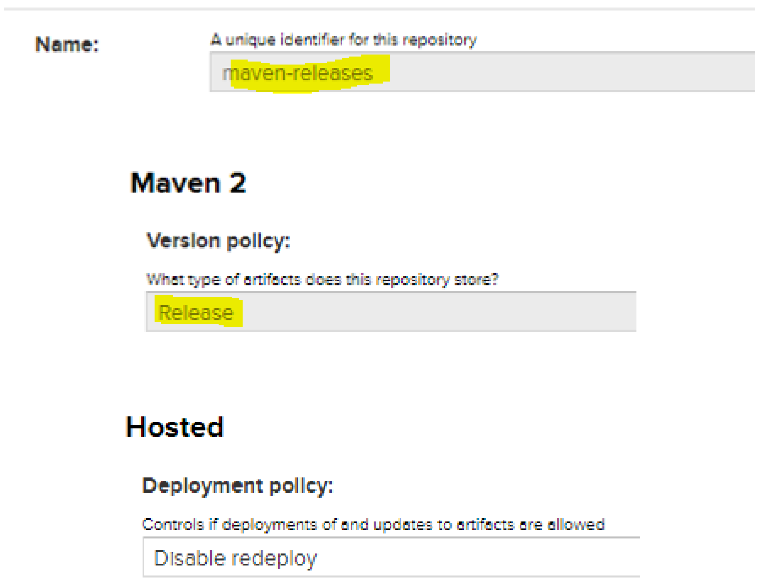
Layout policy:
Validate that all paths are maven artifact or metadata paths
Strict

Storage

Blob store:
Blob store used to store asset contents
maven-snapshots

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

For release, follow the same configuration done for snapshots except Name, Version policy and Deployment policy.



Name: A unique identifier for this repository
maven-releases

Maven 2

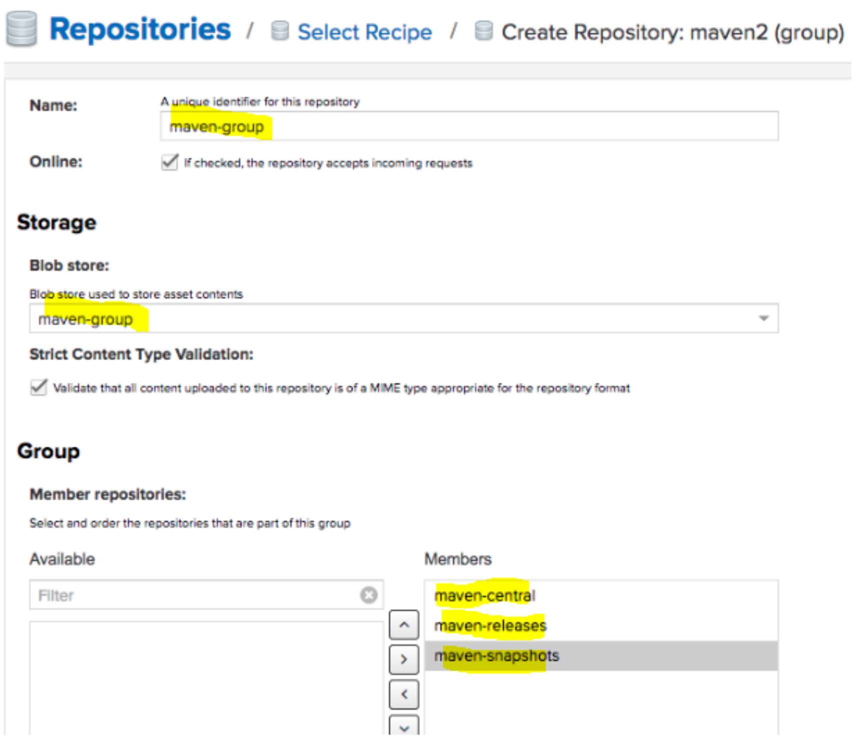
Version policy:
What type of artifacts does this repository store?
Release

Hosted

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

Group Repository

- Group Repository is a collection of other repositories, Where we can combine multiple repositories of the same format into a single item.



Repositories / Select Recipe / Create Repository: maven2 (group)

Name: A unique identifier for this repository
maven-group

Online: ✓ If checked, the repository accepts incoming requests

Storage

Blob store:
Blob store used to store asset contents
maven-group

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

Group

Member repositories:
Select and order the repositories that are part of this group

Available | Members
Filter | maven-central, maven-releases, maven-snapshots

Installation

Ubuntu

1. Install the java 1.8+ version

```
sudo apt-get update
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get install openjdk-8-jdk
/usr/lib/jvm/java-8-openjdk-amd64/bin
/usr/lib/jvm/java-8-openjdk-amd64
```

2. Download the maven 3.6+ version

```
cd /opt
sudo wget https://www-eu.apache.org/dist/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz
sudo tar -xvf apache-maven-3.6.3-bin.tar.gz
```

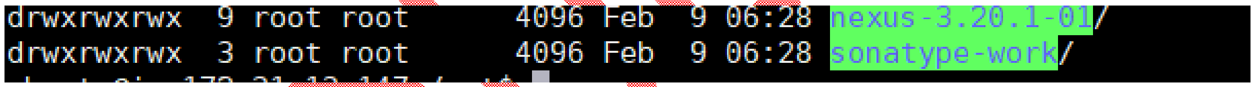
sudo mv apache-maven-3.6.3 maven
sudo chmod 777 -R maven

3. Set the path in root level

```
cd /etc/profile.d
vi paths.sh
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export JAVA_HOME
MAVEN_HOME=/opt/maven
export MAVEN_HOME
PATH=$PATH:$JAVA_HOME/bin:$MAVEN_HOME/bin
export PATH
```

4. Download the sonatype binaries in below url.

```
cd /opt
sudo wget https://download.sonatype.com/nexus/3/latest-unix.tar.gz
sudo tar -xvf latest-unix.tar.gz
sudo chmod 777 -R nexus-3.20.1-01/
sudo chmod 777 -R sonatype-work/
```



```
drwxrwxrwx  9 root root      4096 Feb  9 06:28 nexus-3.20.1-01/
drwxrwxrwx  3 root root      4096 Feb  9 06:28 sonatype-work/
```

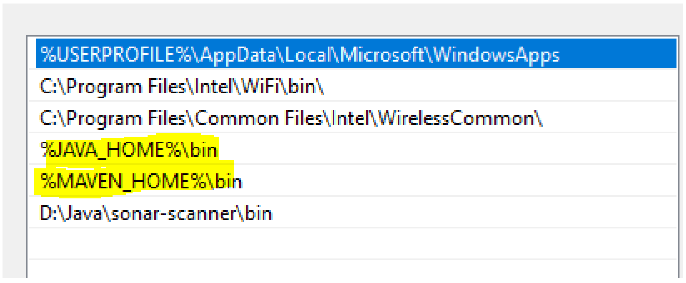
5. Start the sonar server

```
cd /opt/nexus-3.20.1-01/bin
./nexus start (OR) ./nexus run (OR) ./nexus \run
```

6. Test the server
   http://<IP-Address>:8081/

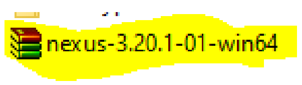
Windows

1. Install java and maven set to the path.



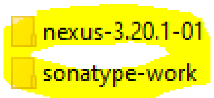
```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Program Files\Intel\WiFi\bin\
C:\Program Files\Common Files\Intel\WirelessCommon\
%JAVA_HOME%\bin
%MAVEN_HOME%\bin
D:\Java\sonar-scanner\bin
```

2. Download and extract the sonatype binaries in below url.
   https://download.sonatype.com/nexus/3/latest-win64.zip



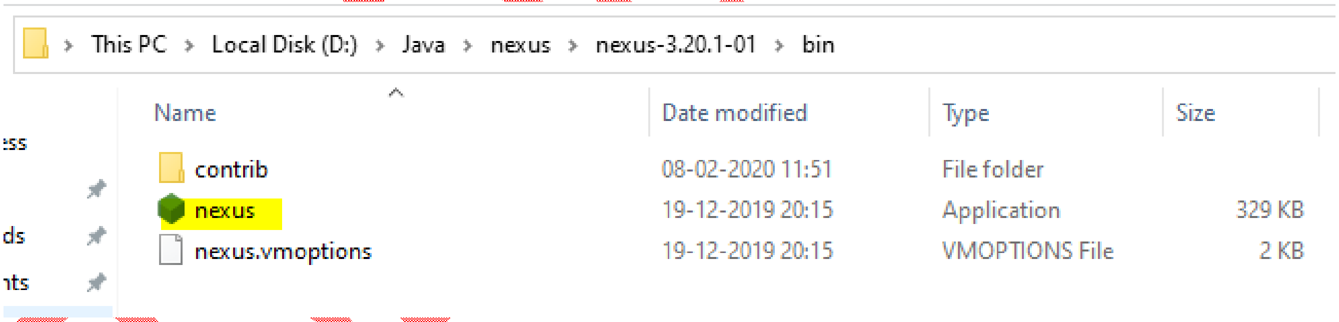
nexus-3.20.1-01-win64

Extract:



nexus-3.20.1-01
sonatype-work

3. Start the server

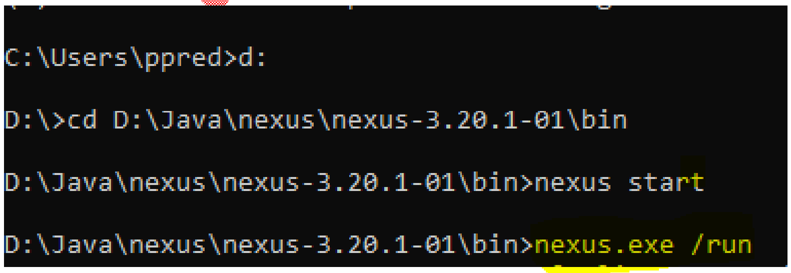


This PC > Local Disk (D:) > Java > nexus > nexus-3.20.1-01 > bin

Name	Date modified	Type	Size
contrib	08-02-2020 11:51	File folder	
nexus	19-12-2019 20:15	Application	329 KB
nexus.vmoptions	19-12-2019 20:15	VMOPTIONS File	2 KB

Open the command prompt run the below command.

➢ nexus.exe /run



```
C:\Users\ppred>d:

D:\>cd D:\Java\nexus\nexus-3.20.1-01\bin

D:\Java\nexus\nexus-3.20.1-01\bin>nexus start

D:\Java\nexus\nexus-3.20.1-01\bin>nexus.exe /run
```

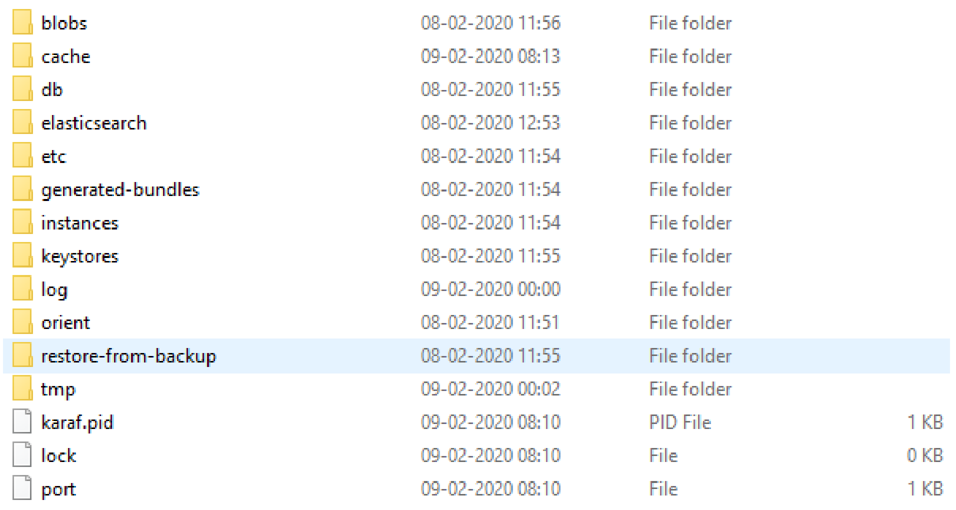
4. Test the server

   http://localhost:8081/

Nexus Directory Structure

The installation directory having below directories:
bin
deploy
etc
lib
public
system
bin: Contains the nexus startup script itself as well as startup-related configuration files.
etc: Contains configuration files.
lib: Contains binary libraries related to Apache Karaf.
public: Contains public resources of the application.
system: Contains all components and plugins that constitute the application.

Data Directory
The data directory, found by default at ../sonatype-work/nexus3, includes subdirectories that contain all the components, repositories, configurations and other data presented by the repository manager. The subdirectories are listed as:

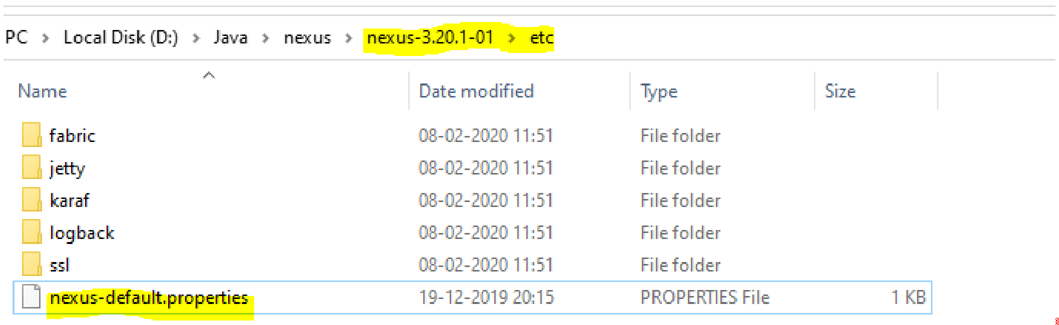


blobs	08-02-2020 11:56	File folder	
cache	09-02-2020 08:13	File folder	
db	08-02-2020 11:55	File folder	
elasticsearch	08-02-2020 12:53	File folder	
etc	08-02-2020 11:54	File folder	
generated-bundles	08-02-2020 11:54	File folder	
instances	08-02-2020 11:54	File folder	
keystores	08-02-2020 11:55	File folder	
log	09-02-2020 00:00	File folder	
orient	08-02-2020 11:51	File folder	
restore-from-backup	08-02-2020 11:55	File folder	
tmp	09-02-2020 00:02	File folder	
karaf.pid	09-02-2020 08:10	PID File	1 KB
lock	09-02-2020 08:10	File	0 KB
port	09-02-2020 08:10	File	1 KB

blobs/
This is the default location of the blob store. If you provided a fully qualified path when creating a new blob store, it may not end up in this directory.
cache/
This directory contains information on currently cached Karaf bundles
db/
This directory contains the OrientDB databases which are the primary storage for your repository manager's metadata
elasticsearch/
This directory contains the currently configured state of Elasticsearch
etc/
This directory contains the main runtime configuration and customization of the repository manager.

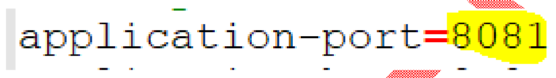
health-check/
This directory contains cached reports from the Repository Health Check feature
keystores/
This contains the automatically generated key used to identify your repository manager
log/
This directory contains several log files that capture information about various aspects of the running repository manager. The nexus.log and request.log files are rotated every day so this directory also contains archived copies of these files. To reclaim disk space, you can delete old log files from the logs directory. Log files found in this directory include:
nexus.log - The main repository manager application log. Log messages contain standard log output fields including date/time, log level, the associated thread, class and message.
request.log - Used to log http access requests to a running repository manager. Log messages contain information such as client host, user and HTTP request attributes including status code, bytes, and user-agent header.
jvm.log - Contains JVM stdout, stderr and thread dump messages
karaf.log - This is the Apache Karaf container log file which contains messages specific to the repository manager startup
The **log** directory also contains tasks subdirectory which contains separate, uniquely named (by date, time and task name) log output files for each task that is run. See Task Logging for more details concerning naming strategy and content of these files.
tmp/
This directory is used for temporary storage

Port number changing:


PC › Local Disk (D:) › Java › nexus › nexus-3.20.1-01 › etc

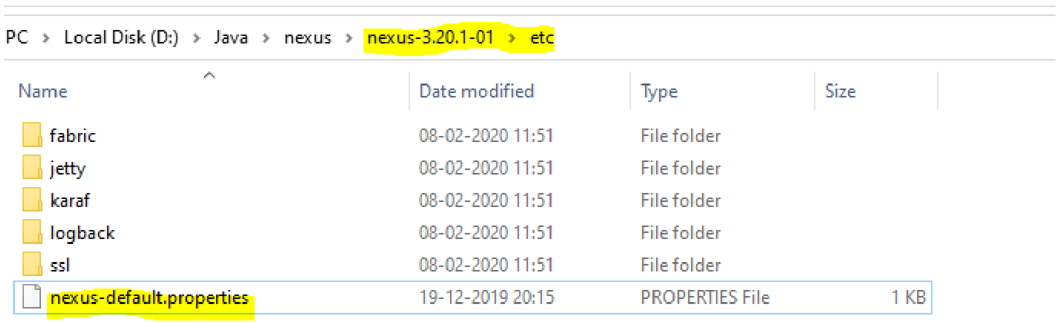
Name	Date modified	Type	Size
fabric	08-02-2020 11:51	File folder	
jetty	08-02-2020 11:51	File folder	
karaf	08-02-2020 11:51	File folder	
logback	08-02-2020 11:51	File folder	
ssl	08-02-2020 11:51	File folder	
nexus-default.properties	19-12-2019 20:15	PROPERTIES File	1 KB

To change the default HTTP port from 8081 to custom port, follow the below steps.
Go to the etc directory and open the **nexus-default.properties** file and update the port number from 8081 to your custom port.
application-port=**8081** ---> **By default port number**.
application-port=**2020** ---> **Customised port number**.


```
application-port=8081
```

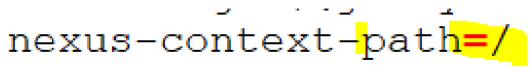
Context root changing:

Go to the etc directory and open the **nexus-default.properties** file and update the context root as follows.


PC › Local Disk (D:) › Java › nexus › nexus-3.20.1-01 › etc

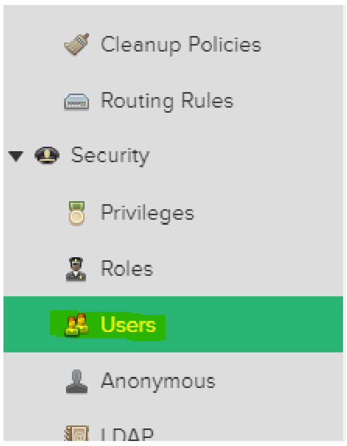
Name	Date modified	Type	Size
fabric	08-02-2020 11:51	File folder	
jetty	08-02-2020 11:51	File folder	
karaf	08-02-2020 11:51	File folder	
logback	08-02-2020 11:51	File folder	
ssl	08-02-2020 11:51	File folder	
nexus-default.properties	19-12-2019 20:15	PROPERTIES File	1 KB

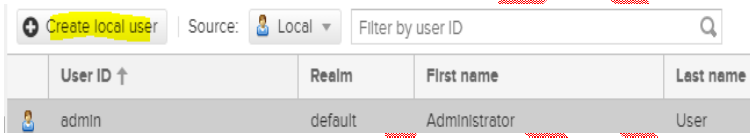
nexus-context-path=/ ---> **By default context root**
nexus-context-path=**/psddevops ---> Customised context root**


```
nexus-context-path=/
```

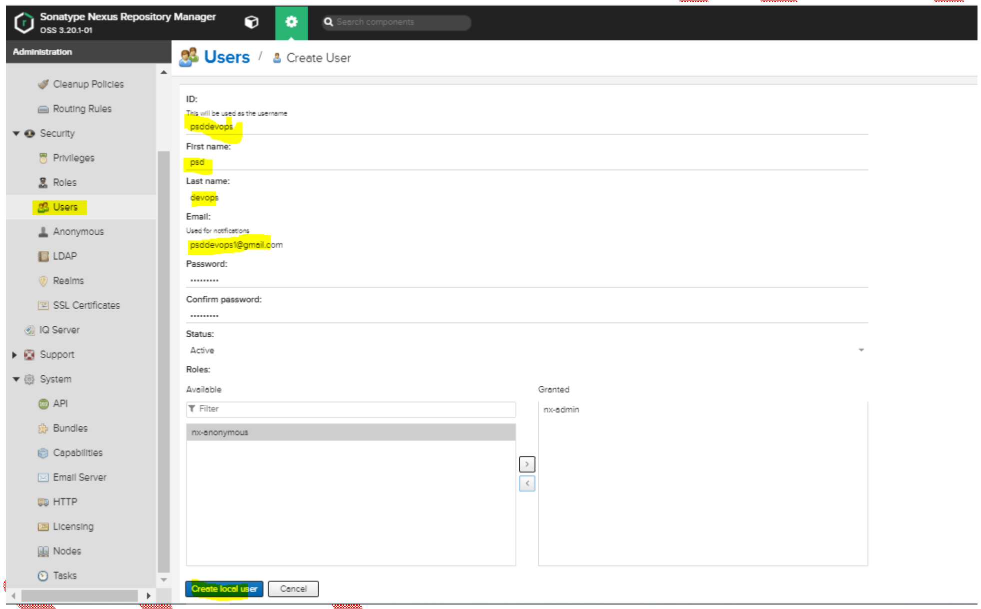
Creating new user

Go to Security → Users → Create local user





User ID	Realm	First name	Last name
admin	default	Administrator	User



ID:
This will be used as the username
psddevops

First name:
psd

Last name:
devops

Email:
Used for notifications
psddevops1@gmail.com

Password:
•••••••••

Confirm password:
•••••••••

Status:
Active

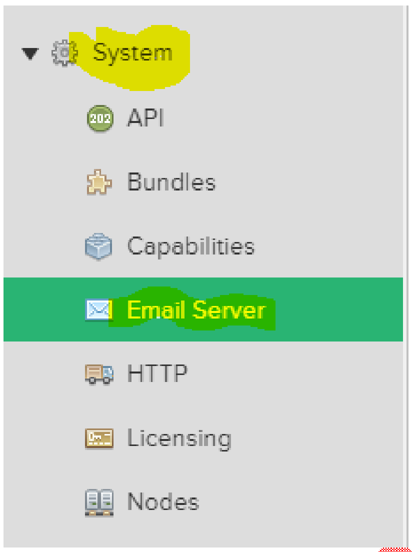
Roles:

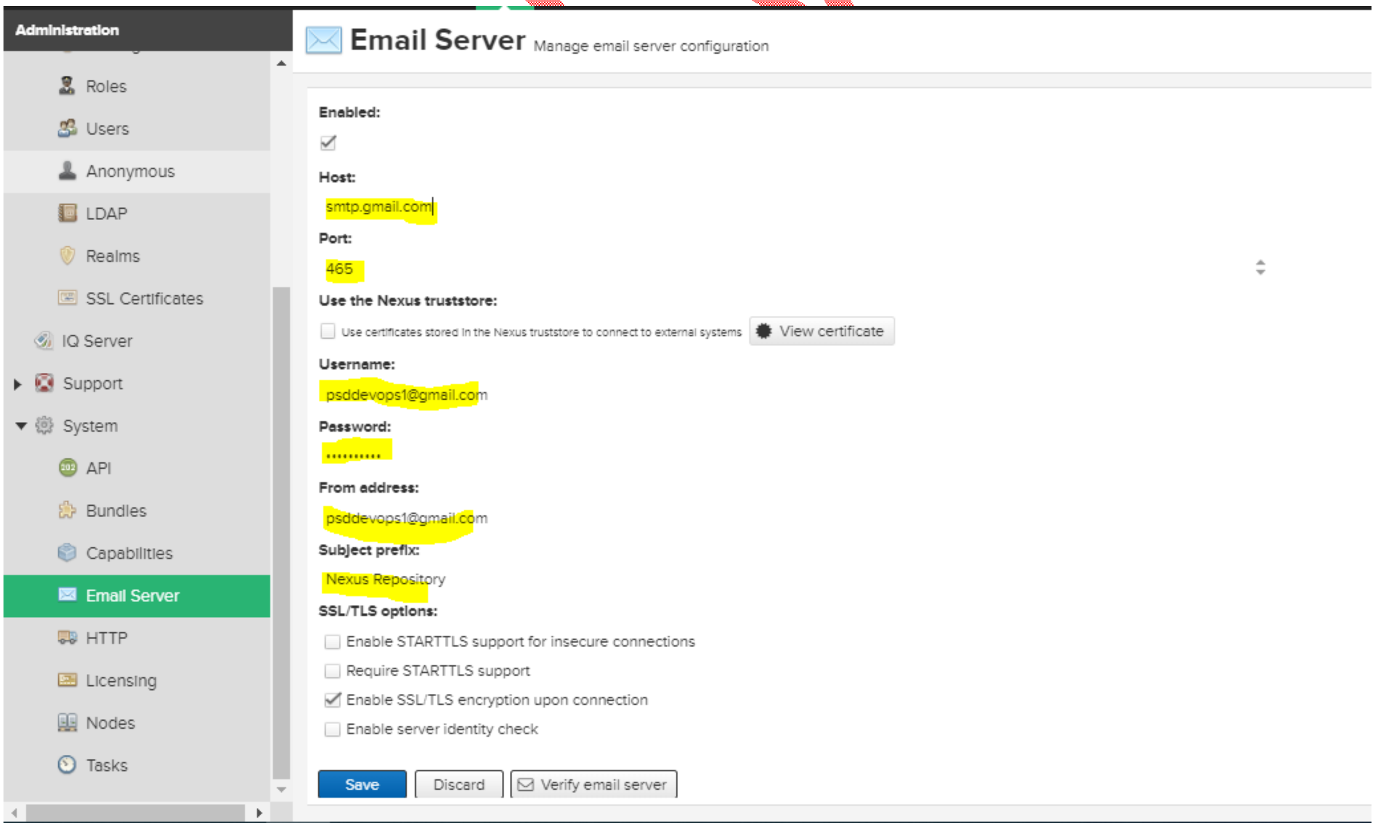
Available	Granted
nx-anonymous	nx-admin

Then click on Save button.

Configure the SMTP Settings

Go to System → Email Server





Email Server Manage email server configuration

Enabled: ✓

Host: smtp.gmail.com

Port: 465

Use the Nexus truststore:

☐ Use certificates stored in the Nexus truststore to connect to external systems View certificate

Username: psddevops1@gmail.com

Password: ••••••••••

From address: psddevops1@gmail.com

Subject prefix: Nexus Repository

SSL/TLS options:

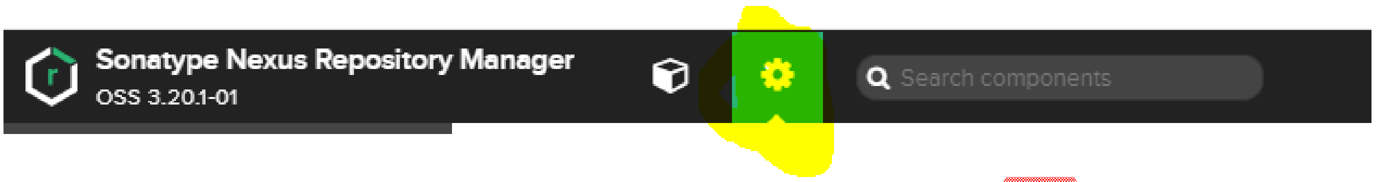
- ☐ Enable STARTTLS support for insecure connections
- ☐ Require STARTTLS support
- ☑ Enable SSL/TLS encryption upon connection
- ☐ Enable server identity check

Save | Discard | Verify email server

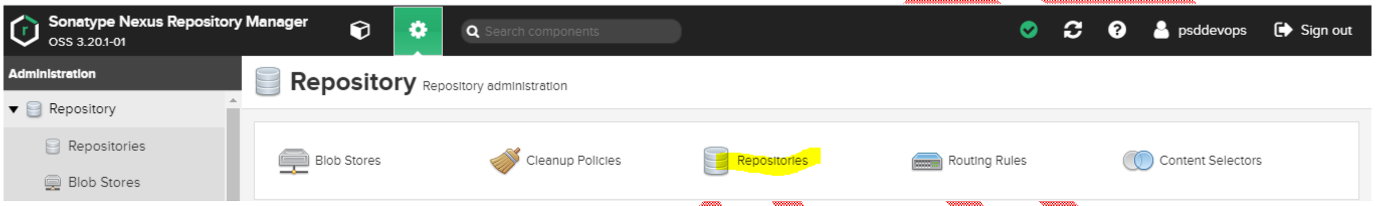
Check Verify email server and save.

Creating the Repos in Nexus

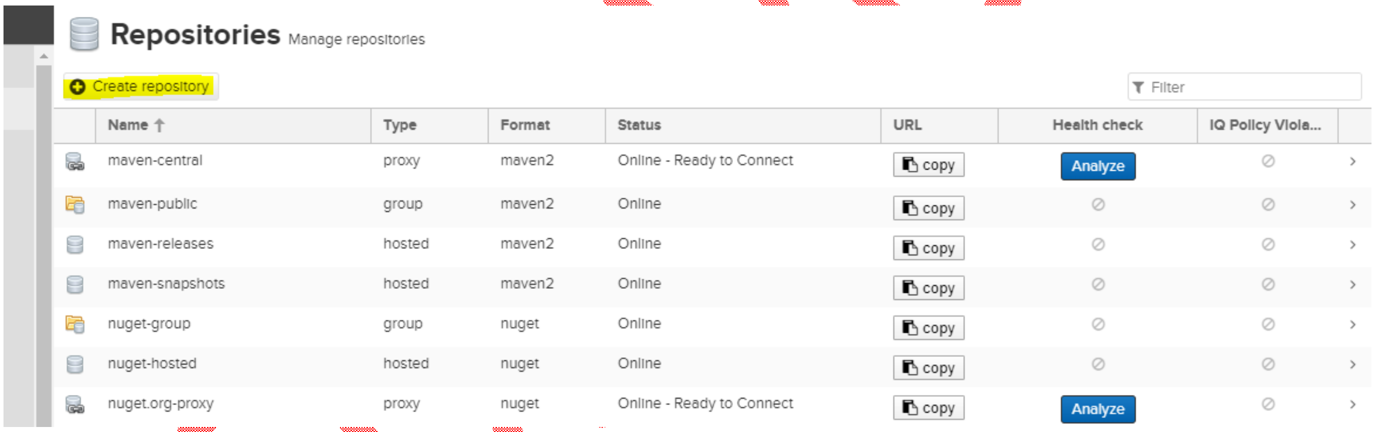
Go to Server **Administration and Configuration**



Click on Repositories

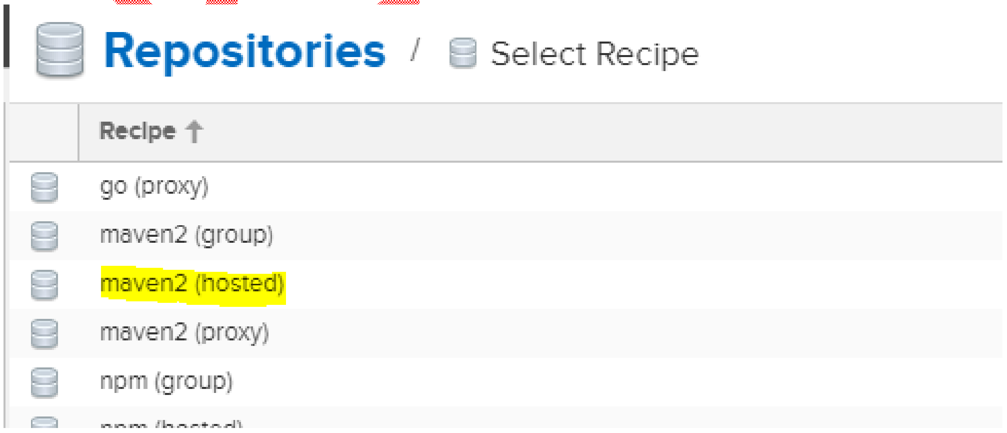


Click on create repository



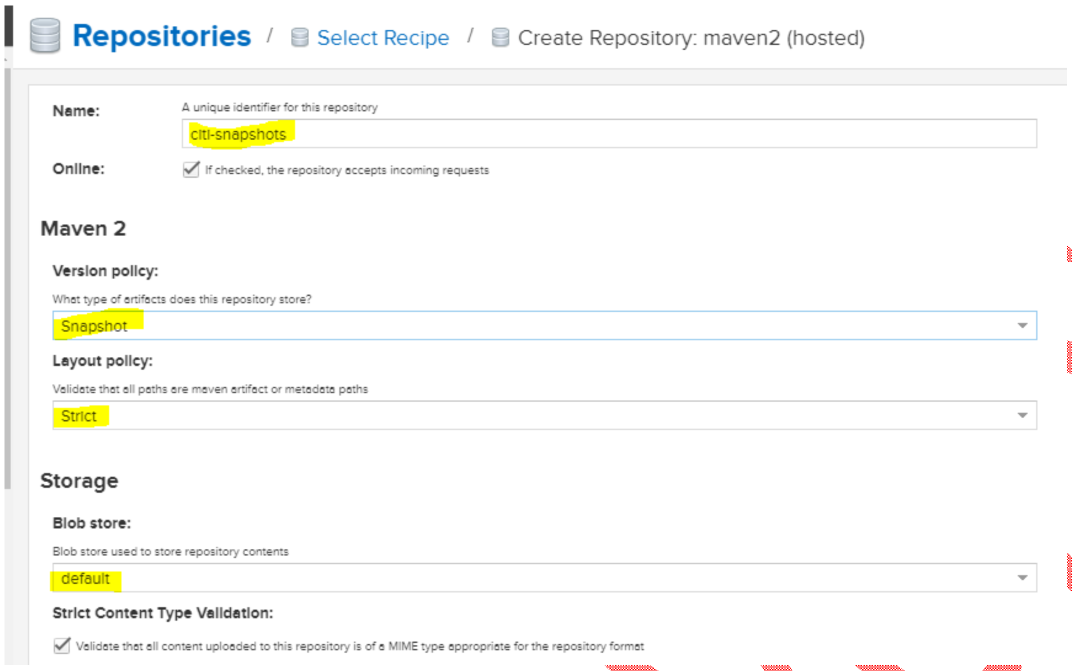
Name ↑	Type	Format	Status	URL	Health check	IQ Policy Viola...
maven-central	proxy	maven2	Online - Ready to Connect	copy	Analyze	⊘
maven-public	group	maven2	Online	copy	⊘	⊘
maven-releases	hosted	maven2	Online	copy	⊘	⊘
maven-snapshots	hosted	maven2	Online	copy	⊘	⊘
nuget-group	group	nuget	Online	copy	⊘	⊘
nuget-hosted	hosted	nuget	Online	copy	⊘	⊘
nuget.org-proxy	proxy	nuget	Online - Ready to Connect	copy	Analyze	⊘

Select maven2(hosted)



Recipe ↑
go (proxy)
maven2 (group)
maven2 (hosted)
maven2 (proxy)
npm (group)
npm (hosted)

Enter the below details for snapshot repository creation.



Repositories / Select Recipe / Create Repository: maven2 (hosted)

Name: A unique identifier for this repository
citi-snapshots

Online: ✓ If checked, the repository accepts incoming requests

Maven 2

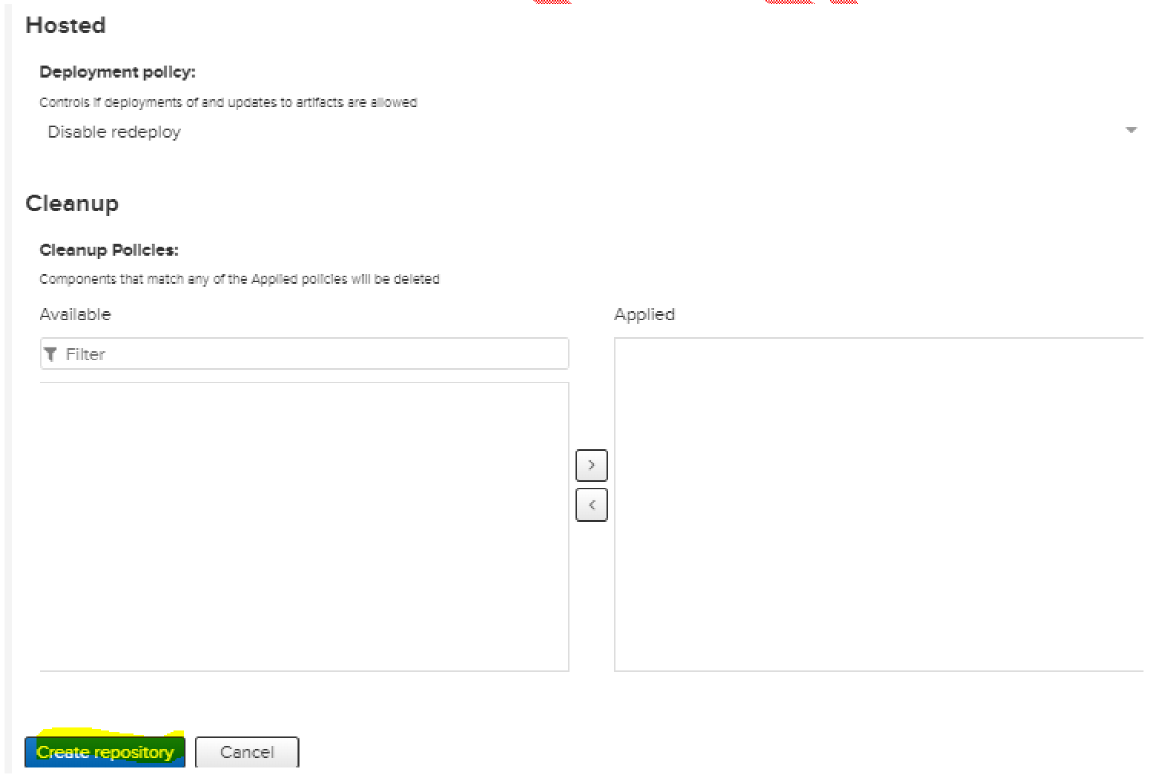
Version policy:
What type of artifacts does this repository store?
Snapshot

Layout policy:
Validate that all paths are maven artifact or metadata paths
Strict

Storage

Blob store:
Blob store used to store repository 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

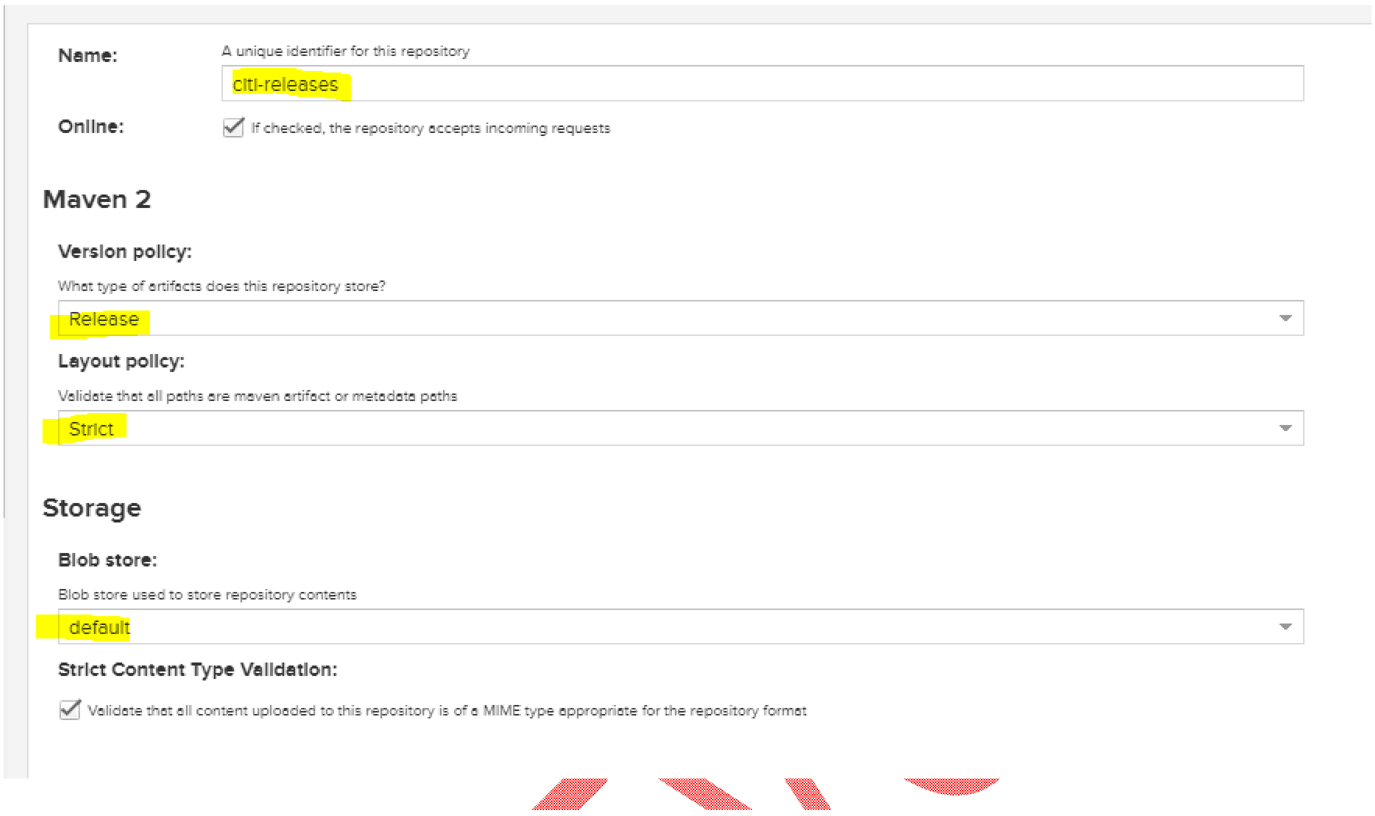
Deployment policy:
Controls if deployments of and updates to artifacts are allowed
Disable redeploy

Cleanup

Cleanup Policies:
Components that match any of the Applied policies will be deleted

Create repository

Enter the below details for release repository creation



Name: A unique identifier for this repository

citi-releases

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

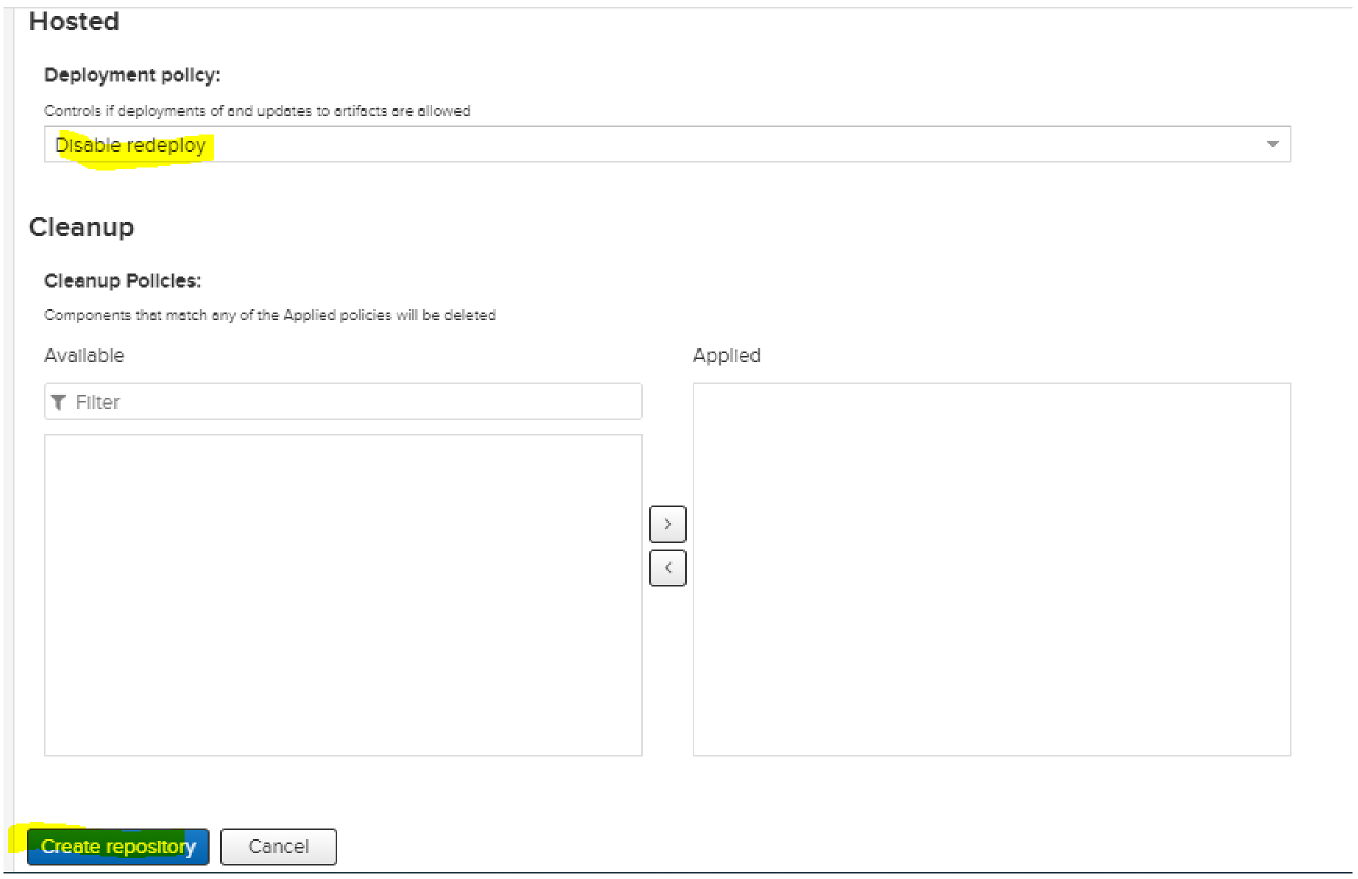
Blob store:

Blob store used to store repository 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

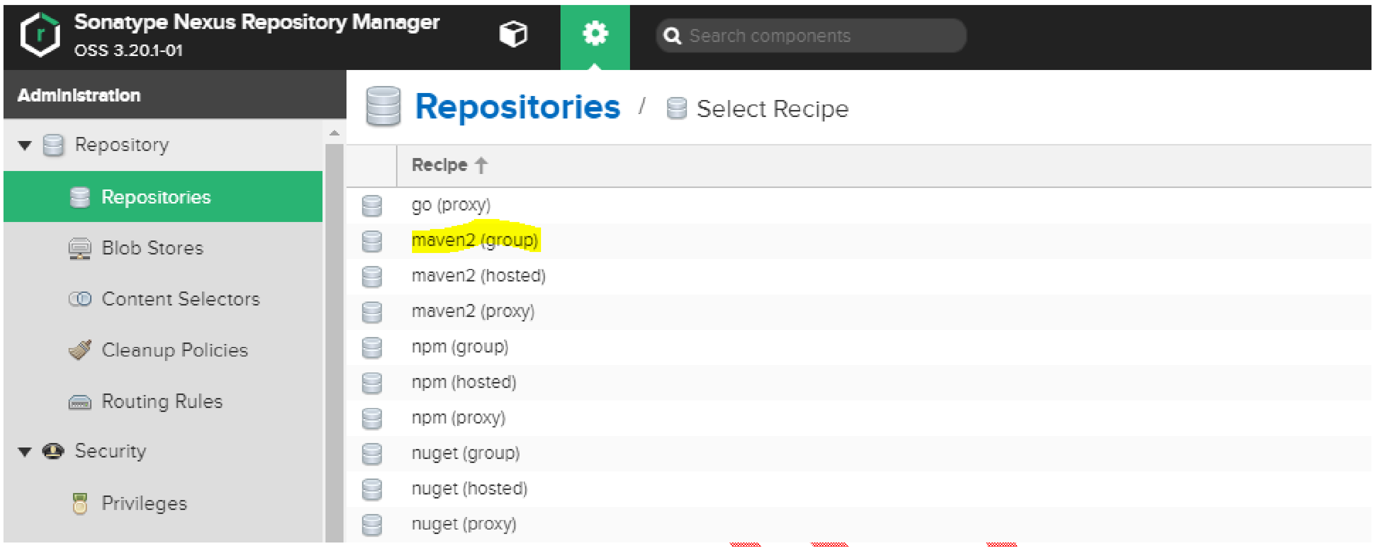
Cleanup Policies:

Components that match any of the Applied policies will be deleted

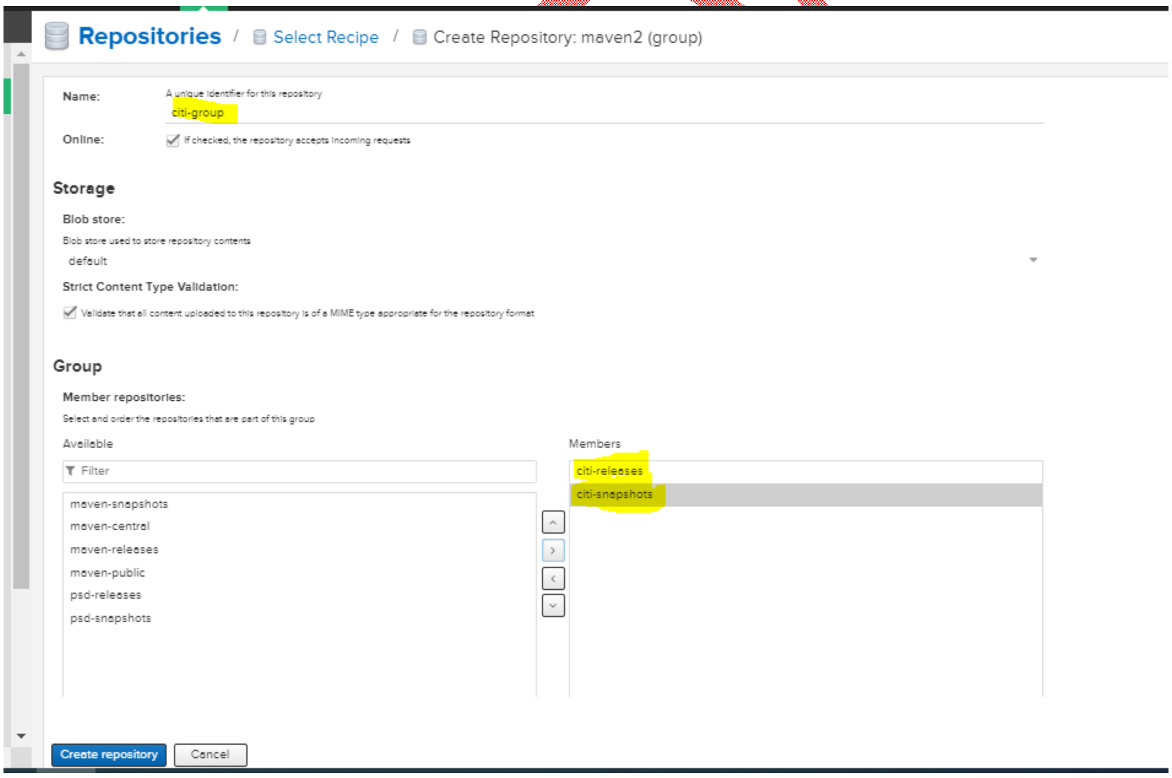
Create repository

Creating group repositories

Select maven2 (group)



Recipe
go (proxy)
maven2 (group)
maven2 (hosted)
maven2 (proxy)
npm (group)
npm (hosted)
npm (proxy)
nuget (group)
nuget (hosted)
nuget (proxy)


Repositories / Select Recipe / Create Repository: maven2 (group)

Name: A unique identifier for this repository
citi-group

Online: ✓ If checked, the repository accepts incoming requests

Storage

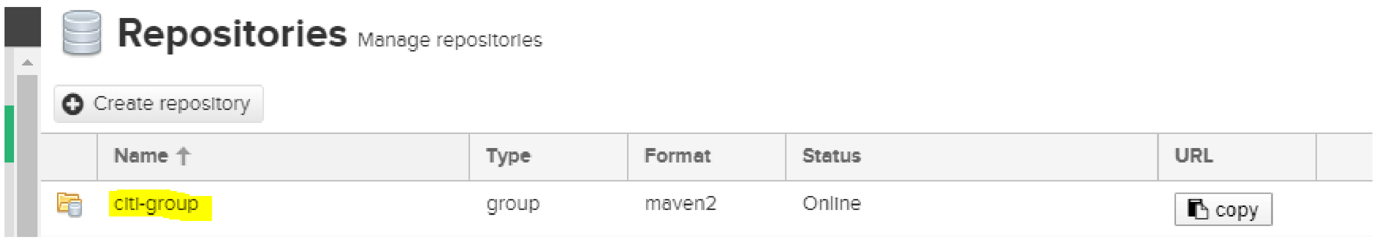
Blob store:
Blob store used to store repository contents
default

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

Group

Member repositories:
Select and order the repositories that are part of this group

Available	Members
maven-snapshots	citi-releases
maven-central	citi-snapshots
maven-releases	
maven-public	
psd-releases	
psd-snapshots	

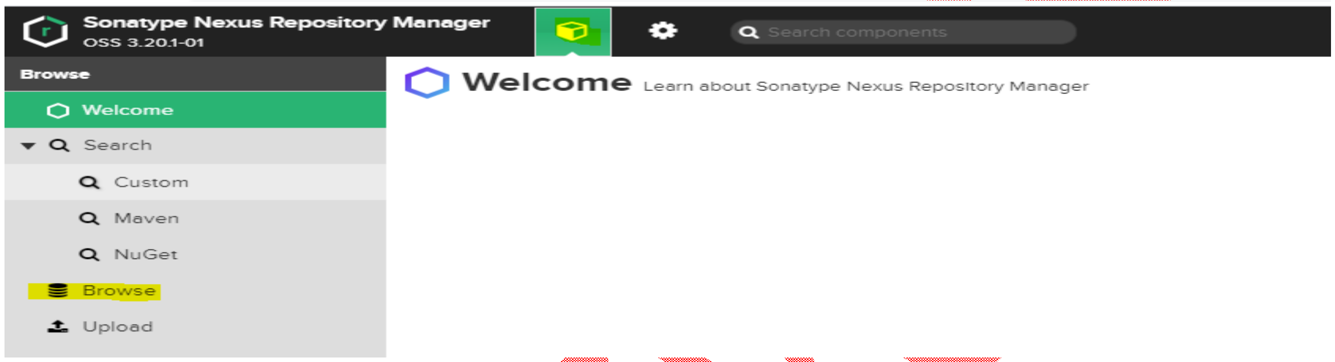

Repositories Manage repositories

Create repository

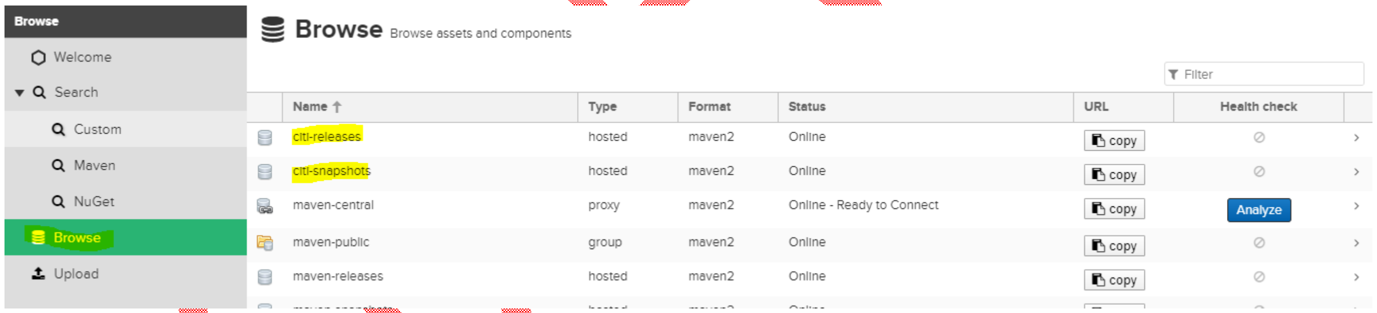
	Name	Type	Format	Status	URL
	citi-group	group	maven2	Online	copy

Browse the repositories

Go to home page click on browse icon

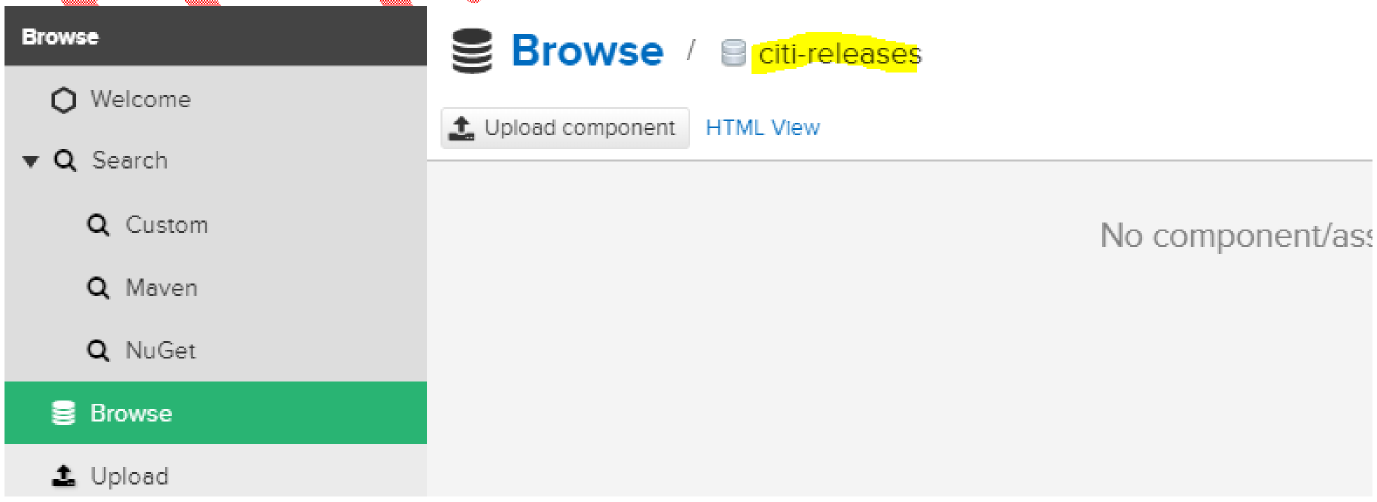


Welcome Learn about Sonatype Nexus Repository Manager



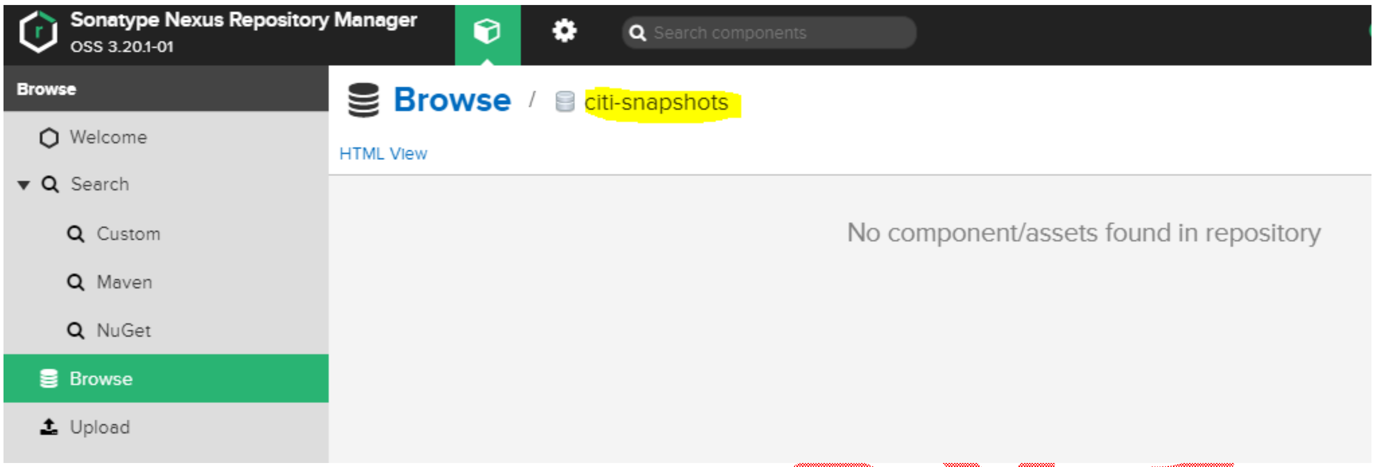
Browse Browse assets and components

	Name	Type	Format	Status	URL	Health check
	citi-releases	hosted	maven2	Online	copy	
	citi-snapshots	hosted	maven2	Online	copy	
	maven-central	proxy	maven2	Online - Ready to Connect	copy	Analyze
	maven-public	group	maven2	Online	copy	
	maven-releases	hosted	maven2	Online	copy	



Browse / citi-releases

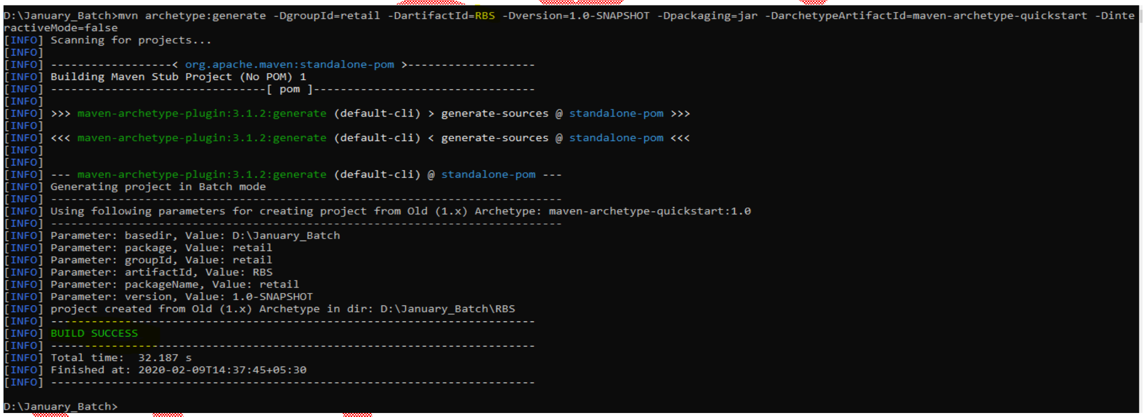
No component/ass



No component/assets found in repository

Nexus integration with maven

1. Create the maven based application
   mvn archetype:generate -DgroupId=retail -DartifactId=RBS -Dversion=1.0-SNAPSHOT -Dpackaging=jar -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false



```
D:\January_Batch>mvn archetype:generate -DgroupId=retail -DartifactId=RBS -Dversion=1.0-SNAPSHOT -Dpackaging=jar -DarchetypeArtifactId=maven-archetype-quickstart -Dinte
ractiveMode=false
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< org.apache.maven:standalone-pom >-------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] --------------------------------[ pom ]---------------------------------
[INFO]
[INFO] >>> maven-archetype-plugin:3.1.2:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:3.1.2:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO]
[INFO] --- maven-archetype-plugin:3.1.2:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.0
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: basedir, Value: D:\January_Batch
[INFO] Parameter: package, Value: retail
[INFO] Parameter: groupId, Value: retail
[INFO] Parameter: artifactId, Value: RBS
[INFO] Parameter: packageName, Value: retail
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: D:\January_Batch\RBS
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  32.187 s
[INFO] Finished at: 2020-02-09T14:37:45+05:30
[INFO] ------------------------------------------------------------------------

D:\January_Batch>
```

2. Update the pom.xml and settings.xml below configurarions

   Pom.xml

   Add below lines after </dependency> tag

   <distributionManagement>
   <repository>
   <id>releases</id>

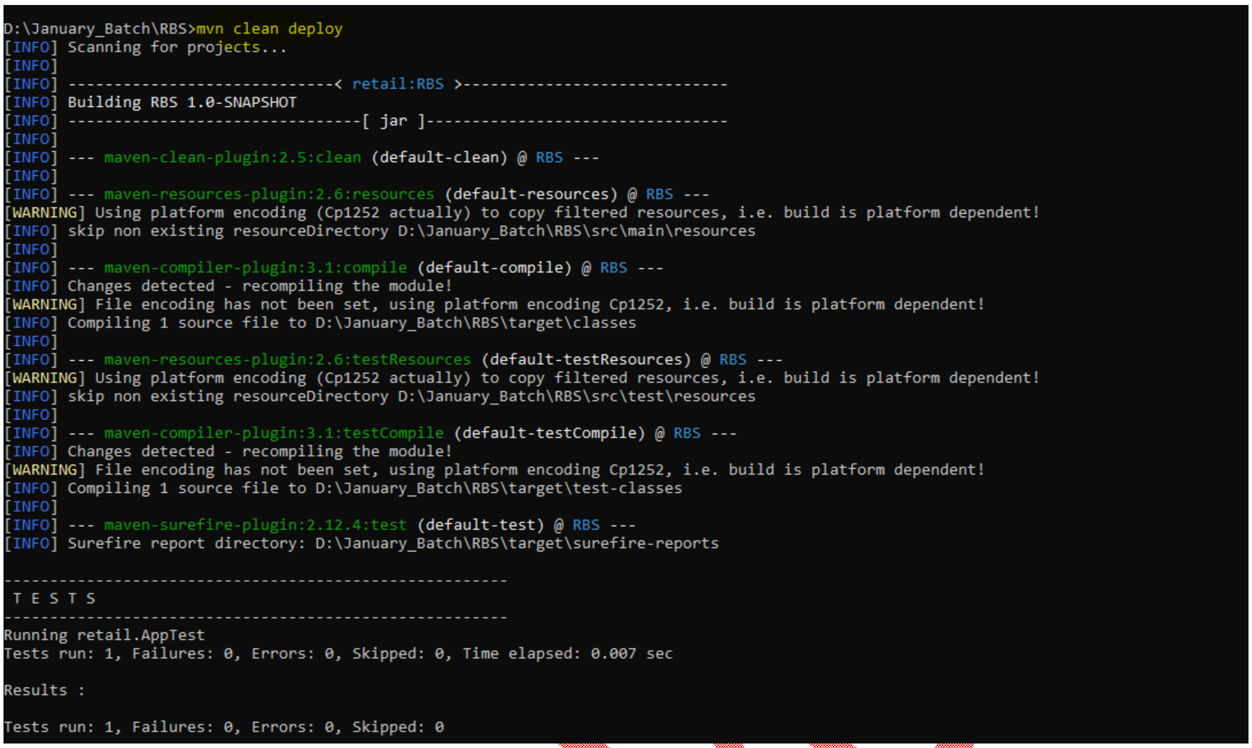
```
<name>releases</name>
<url>http://localhost:8081/repository/citi-releases/</url>
</repository>
<snapshotRepository>
<id>snapshots</id>
<name>snapshots</name>
<url>http://localhost:8081/repository/citi-snapshots/</url>
</snapshotRepository>
</distributionManagement>
```

Settings.xml

Add the below lines inside <servers> tag

```
<server>
     <id>snapshots</id>
     <username>psddevops</username>
     <password>psddevops</password>
   </server>
    <server>
     <id>releases</id>
     <username>psddevops</username>
     <password>psddevops</password>
   </server>
```

Run the application **mvn clean deploy**

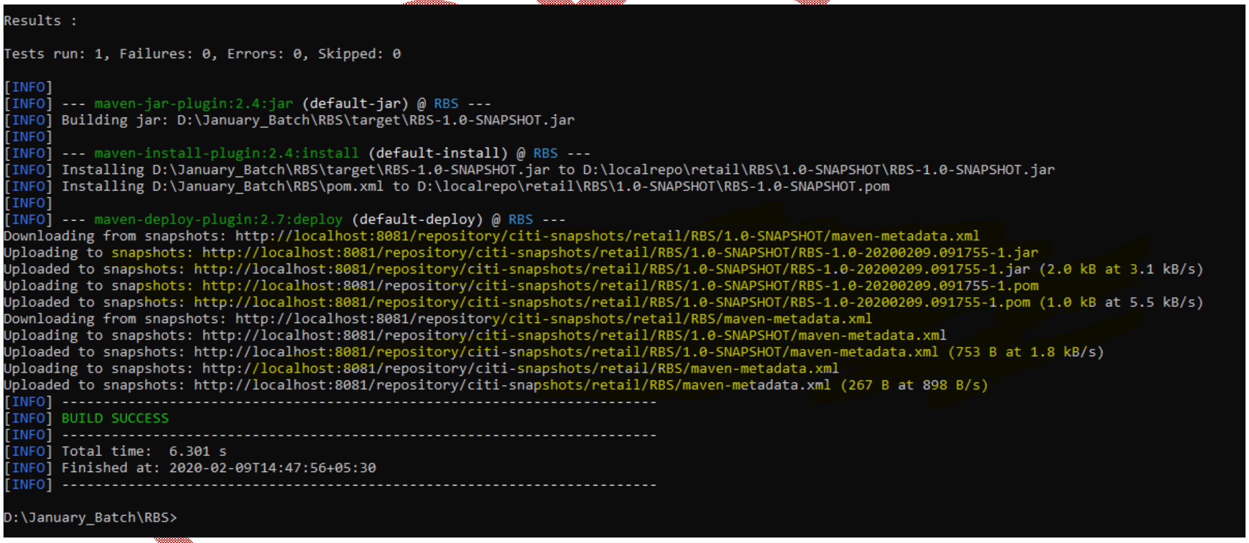


```
D:\January_Batch\RBS>mvn clean deploy
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------------< retail:RBS >----------------------------
[INFO] Building RBS 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ RBS ---
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ RBS ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory D:\January_Batch\RBS\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ RBS ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 1 source file to D:\January_Batch\RBS\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ RBS ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory D:\January_Batch\RBS\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ RBS ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 1 source file to D:\January_Batch\RBS\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ RBS ---
[INFO] Surefire report directory: D:\January_Batch\RBS\target\surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running retail.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.007 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```



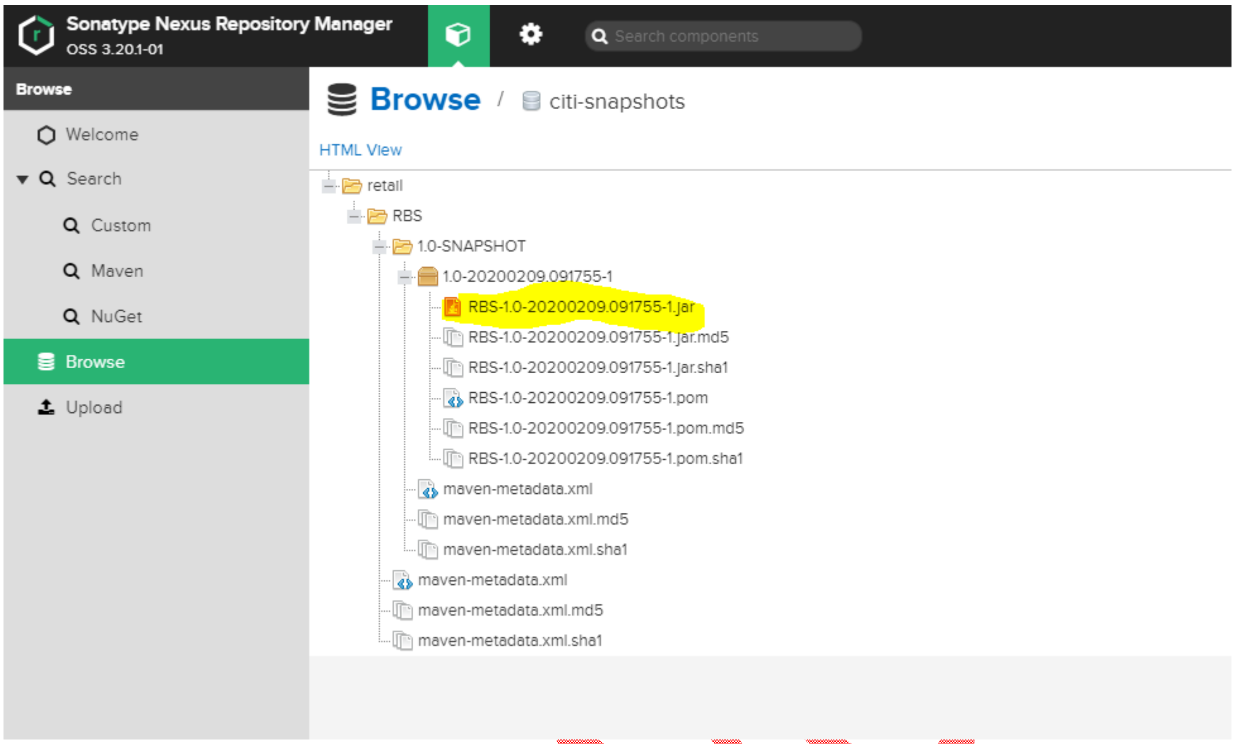
```
Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ RBS ---
[INFO] Building jar: D:\January_Batch\RBS\target\RBS-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ RBS ---
[INFO] Installing D:\January_Batch\RBS\target\RBS-1.0-SNAPSHOT.jar to D:\localrepo\retail\RBS\1.0-SNAPSHOT\RBS-1.0-SNAPSHOT.jar
[INFO] Installing D:\January_Batch\RBS\pom.xml to D:\localrepo\retail\RBS\1.0-SNAPSHOT\RBS-1.0-SNAPSHOT.pom
[INFO]
[INFO] --- maven-deploy-plugin:2.7:deploy (default-deploy) @ RBS ---
Downloading from snapshots: http://localhost:8081/repository/citi-snapshots/retail/RBS/1.0-SNAPSHOT/maven-metadata.xml
Uploading to snapshots: http://localhost:8081/repository/citi-snapshots/retail/RBS/1.0-SNAPSHOT/RBS-1.0-20200209.091755-1.jar
Uploaded to snapshots: http://localhost:8081/repository/citi-snapshots/retail/RBS/1.0-SNAPSHOT/RBS-1.0-20200209.091755-1.jar (2.0 kB at 3.1 kB/s)
Uploading to snapshots: http://localhost:8081/repository/citi-snapshots/retail/RBS/1.0-SNAPSHOT/RBS-1.0-20200209.091755-1.pom
Uploaded to snapshots: http://localhost:8081/repository/citi-snapshots/retail/RBS/1.0-SNAPSHOT/RBS-1.0-20200209.091755-1.pom (1.0 kB at 5.5 kB/s)
Downloading from snapshots: http://localhost:8081/repository/citi-snapshots/retail/RBS/maven-metadata.xml
Uploading to snapshots: http://localhost:8081/repository/citi-snapshots/retail/RBS/1.0-SNAPSHOT/maven-metadata.xml
Uploaded to snapshots: http://localhost:8081/repository/citi-snapshots/retail/RBS/1.0-SNAPSHOT/maven-metadata.xml (753 B at 1.8 kB/s)
Uploading to snapshots: http://localhost:8081/repository/citi-snapshots/retail/RBS/maven-metadata.xml
Uploaded to snapshots: http://localhost:8081/repository/citi-snapshots/retail/RBS/maven-metadata.xml (267 B at 898 B/s)
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  6.301 s
[INFO] Finished at: 2020-02-09T14:47:56+05:30
[INFO] ------------------------------------------------------------------------

D:\January_Batch\RBS>
```

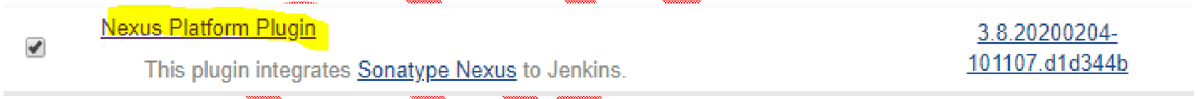
Verify the nexus repository



```
retail
  RBS
    1.0-SNAPSHOT
      1.0-20200209.091755-1
        RBS-1.0-20200209.091755-1.jar
        RBS-1.0-20200209.091755-1.jar.md5
        RBS-1.0-20200209.091755-1.jar.sha1
        RBS-1.0-20200209.091755-1.pom
        RBS-1.0-20200209.091755-1.pom.md5
        RBS-1.0-20200209.091755-1.pom.sha1
      maven-metadata.xml
      maven-metadata.xml.md5
      maven-metadata.xml.sha1
    maven-metadata.xml
    maven-metadata.xml.md5
    maven-metadata.xml.sha1
```

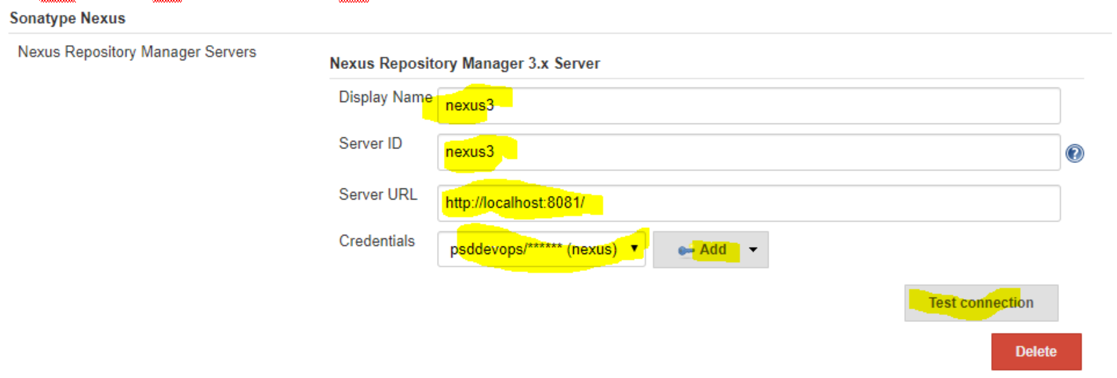
Nexus Integration with Jenkins

Install the Nexus Platform Plugin



✓	Nexus Platform Plugin This plugin integrates Sonatype Nexus to Jenkins.	3.8.20200204-101107.d1d344b

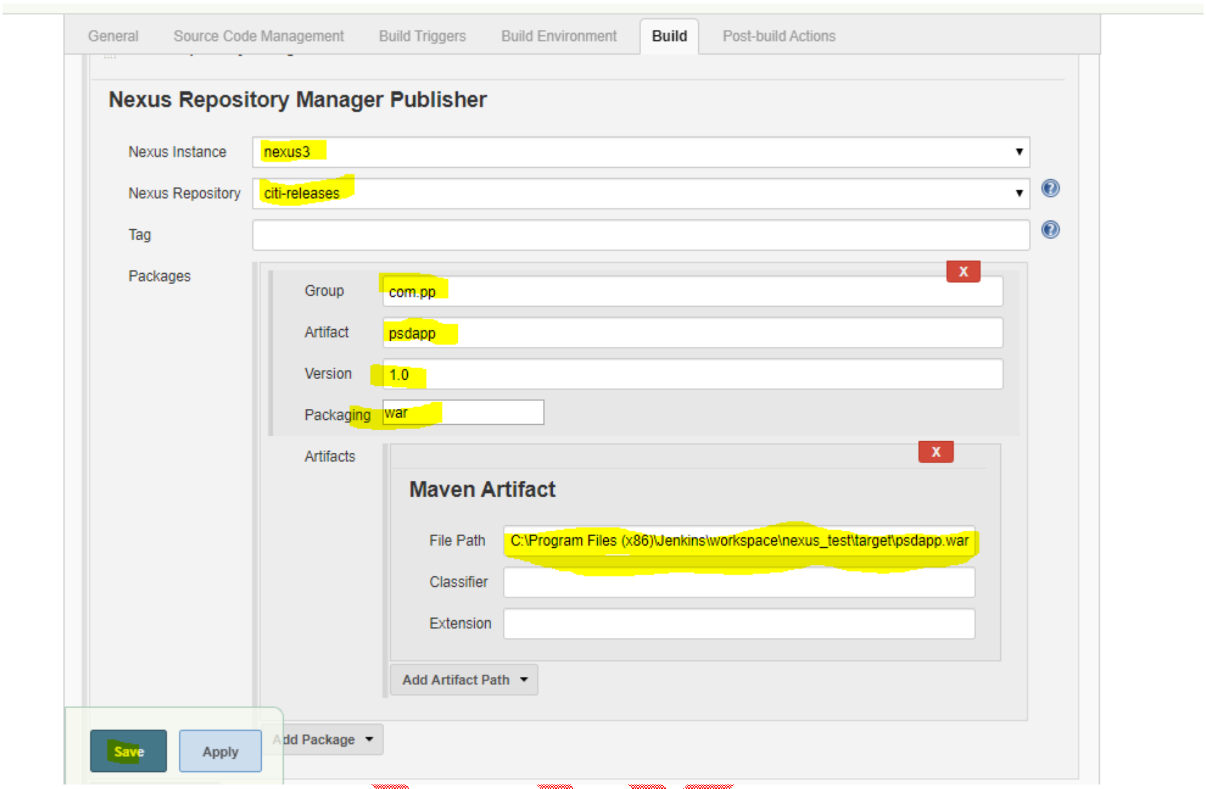
Configure and test the nexus server in Jenkins .
Manage Jenkins → Configure System


Sonatype Nexus
Nexus Repository Manager Servers
Nexus Repository Manager 3.x Server
Add
Test connection
Delete

Display Name	nexus3
Server ID	nexus3
Server URL	http://localhost:8081/
Credentials	psddevops/****** (nexus)

Go to our job and configure

Open the job → configure →Build →Add Build step → **Nexus Repository Manager Publisher**

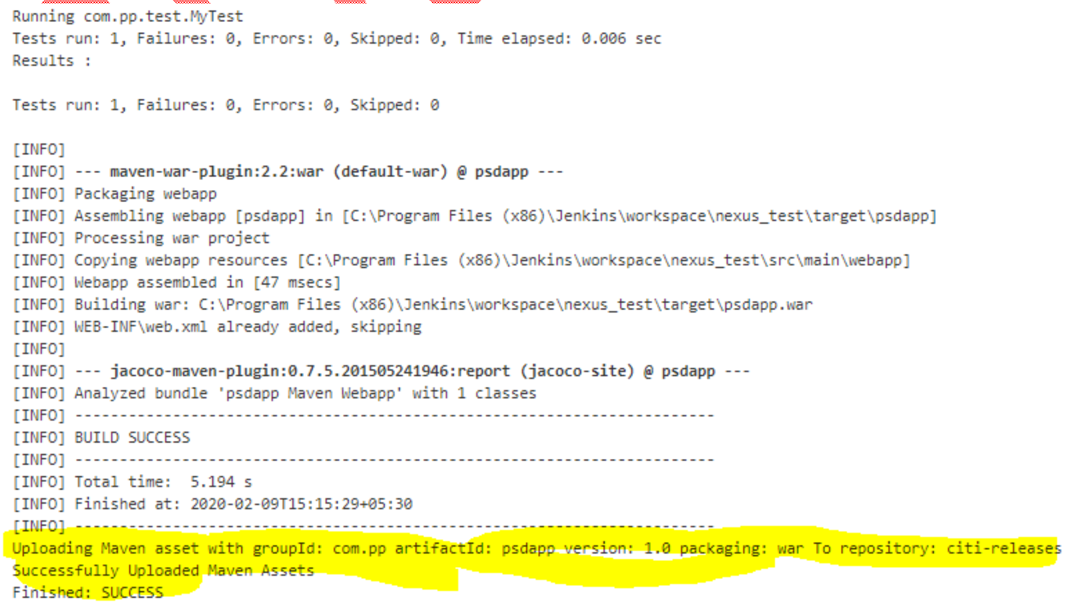


Nexus Repository Manager Publisher

Nexus Instance: nexus3
Nexus Repository: citi-releases
Tag:
Packages
Group: com.pp
Artifact: psdapp
Version: 1.0
Packaging: war
Artifacts
Maven Artifact
File Path: C:\Program Files (x86)\Jenkins\workspace\nexus_test\target\psdapp.war
Classifier:
Extension:

Build the job

Job-Log

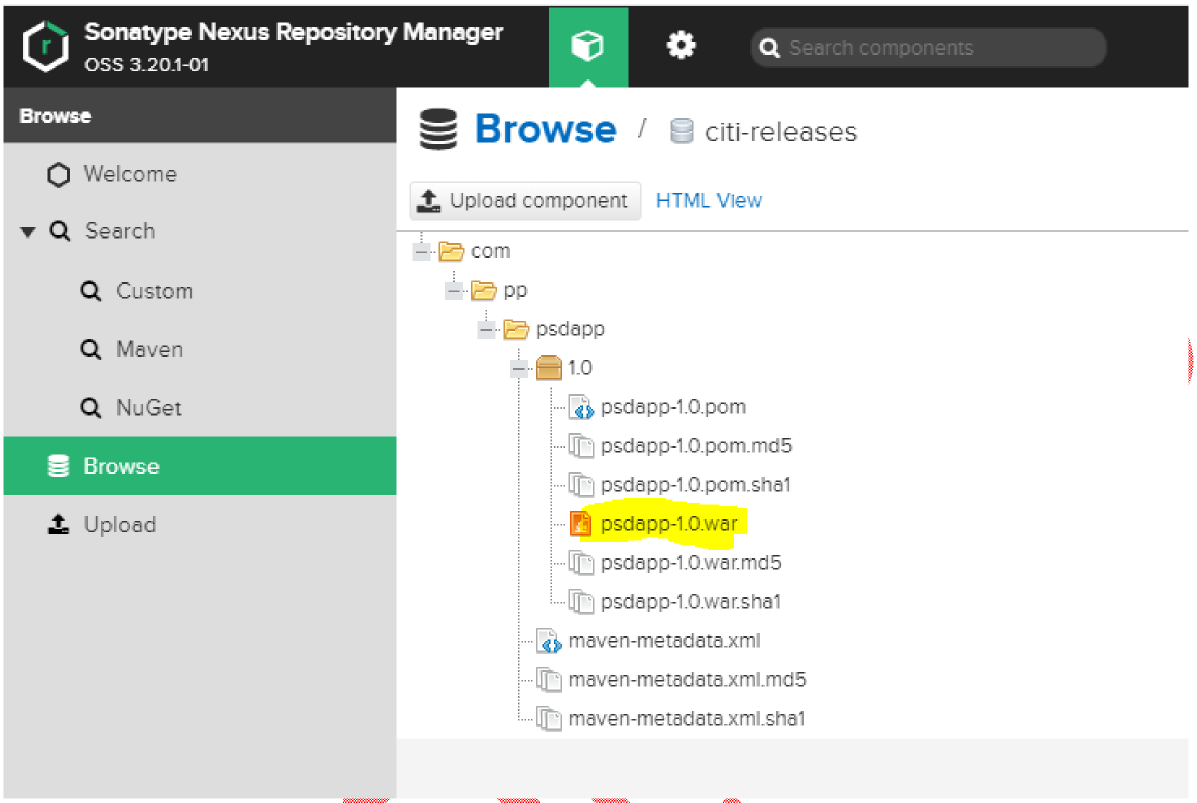


```
Running com.pp.test.MyTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 sec
Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-war-plugin:2.2:war (default-war) @ psdapp ---
[INFO] Packaging webapp
[INFO] Assembling webapp [psdapp] in [C:\Program Files (x86)\Jenkins\workspace\nexus_test\target\psdapp]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\Program Files (x86)\Jenkins\workspace\nexus_test\src\main\webapp]
[INFO] Webapp assembled in [47 msecs]
[INFO] Building war: C:\Program Files (x86)\Jenkins\workspace\nexus_test\target\psdapp.war
[INFO] WEB-INF\web.xml already added, skipping
[INFO]
[INFO] --- jacoco-maven-plugin:0.7.5.201505241946:report (jacoco-site) @ psdapp ---
[INFO] Analyzed bundle 'psdapp Maven Webapp' with 1 classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.194 s
[INFO] Finished at: 2020-02-09T15:15:29+05:30
[INFO] ------------------------------------------------------------------------
Uploading Maven asset with groupId: com.pp artifactId: psdapp version: 1.0 packaging: war To repository: citi-releases
Successfully Uploaded Maven Assets
Finished: SUCCESS
```

Verify the nexus citi-releases repo



Sonatype Nexus Repository Manager
OSS 3.20.1-01

Browse / citi-releases

```
com
  pp
    psdapp
      1.0
        psdapp-1.0.pom
        psdapp-1.0.pom.md5
        psdapp-1.0.pom.sha1
        psdapp-1.0.war
        psdapp-1.0.war.md5
        psdapp-1.0.war.sha1
      maven-metadata.xml
      maven-metadata.xml.md5
      maven-metadata.xml.sha1
```