

# Deploying Maven Project to Nexus

Nexus is a common repo manager. It maintains proxy to repo like maven central, maven releases, maven snapshots and maven public (combination of all 3), etc.

Go to settings on nexus server (localhost:8081) and click on create repository under repositories section.

There select the recipe -> maven hosted.

First create repo with name nexus-snapshot and under Maven 2 section give layout policy -> strict and version policy -> Snapshot.

For default blob store use default. Under hosted deployment policy select disable redeploy.

Select none under cleanup policy.

Take the url from the repo that was build.

In the pom add:

```
<repositories>
  <repository>
    <id>nexus-snapshot</id>
    <url>http://localhost:8081/repository/nexus-snapshot</url>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
</repositories>
```

```
<distributionManagement>
  <snapshotRepository>
    <id>nexus-snapshot</id>
    <url>http://localhost:8081/repository/nexus-snapshot</url>
  </snapshotRepository>
</distributionManagement>
```

```
<server>
  <id>nexus-snapshot</id>
  <username>admin</username>
  <password>admin123</password>
</server>
```

Check if id matches.

Repositories element allows resolving artifacts and distributionManagement is gonna allow to publish artifacts.

2 steps we are handling:

1. Repositories element is saying to go ahead and resolve artifacts from there. Any additional repo can be resolved from there.
2. distributionManagement gonna allow artifacts be published

To create release one:

```
<repositories>
  <repository>
    <id>nexus-snapshot</id>
    <url>http://localhost:8081/repository/nexus-snapshot/</url>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
  <repository>
    <id>nexus-release</id>
    <url>http://localhost:8081/repository/nexus-release/</url>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
</repositories>

<distributionManagement>
  <snapshotRepository>
    <id>nexus-snapshot</id>
    <url>http://localhost:8081/repository/nexus-snapshot/</url>
  </snapshotRepository>
  <repository>
    <id>nexus-release</id>
    <url>http://localhost:8081/repository/nexus-release/</url>
  </repository>
</distributionManagement>
```

Also add server

```
<servers>
  <server>
    <id>nexus-snapshot</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
  <server>
    <id>nexus-release</id>
    <username>admin</username>
    <password>admin123</password>
```

```
</server>
</servers>
```

Nexus repo groups:

The repo build can have virtual other repos to connect and user. Select the recipe maven group.

Under group select the repos you want.

This helps in making the nexus repo manager as our primary repo. Nexus group repo will be going to be our virtual repo (e.g. adding maven central, maven release and snapshot which we have made).

We can setup a mirror for our project

```
<mirrors>
  <mirror>
    <id>central</id>
    <name>central</name>
    <url>http://localhost:8081/repository/nexus-group</url>
    <mirrorOf>*</mirrorOf>
  </mirror>
</mirrors>
```

```
<dependencies>
  <dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <version>1.3.166</version>
  </dependency>
</dependencies>
```

What happens nexus group first seeks the repo we are searching in their local cache but if it is not found it will go into the remote plus our own repos we linked into the group to find the repo and save it in cache and then return it.