**Configuring Nexus as a Maven repo**

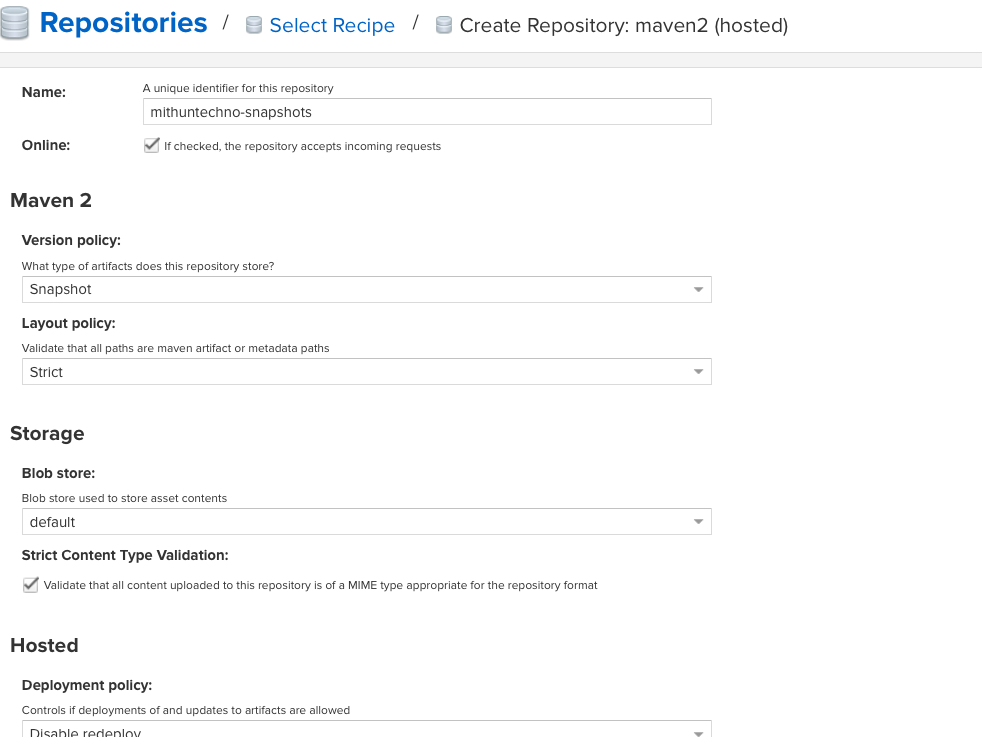
What we will do:  
– create a private (hosted) repository for our snapshots  
– create a private (hosted) repository for our releases  
– create a proxy repository pointing to Maven Central

**snapshots repo**

A repository for Maven artifacts that you deploy **with** -SNAPSHOT in the end of the version tag of your pom.xml:

<version>1.0.0-SNAPSHOT</version>

Create a new maven (hosted) repository and configure it like:

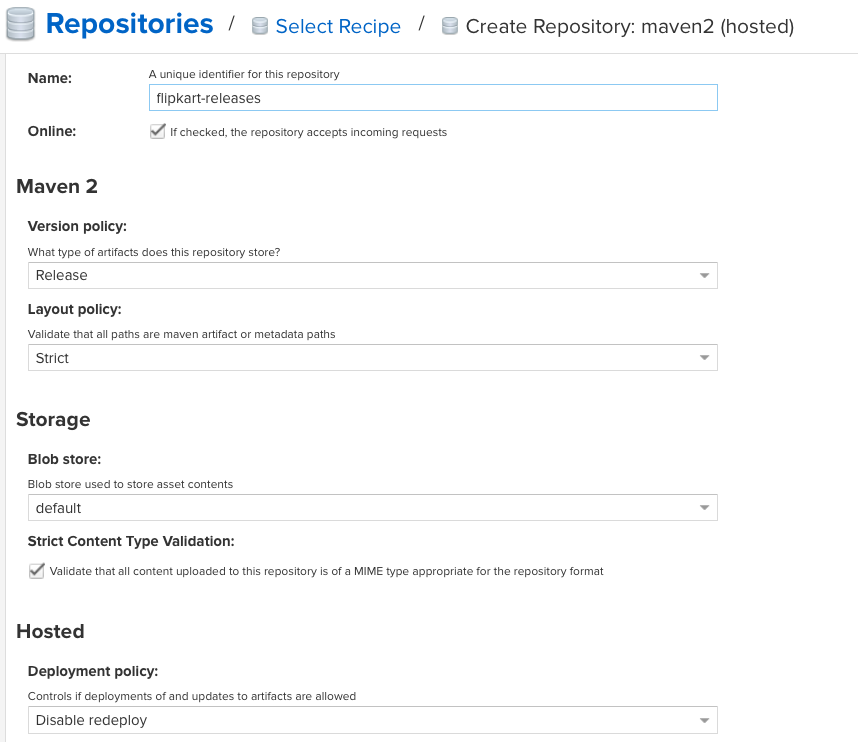


**releases repo**

A repository for Maven artifact that you deploy **without** -SNAPSHOT in the end of the version tag of your pom.xml:

<version>1.0.0</version>

Create a new maven (hosted) repository and configure it like:

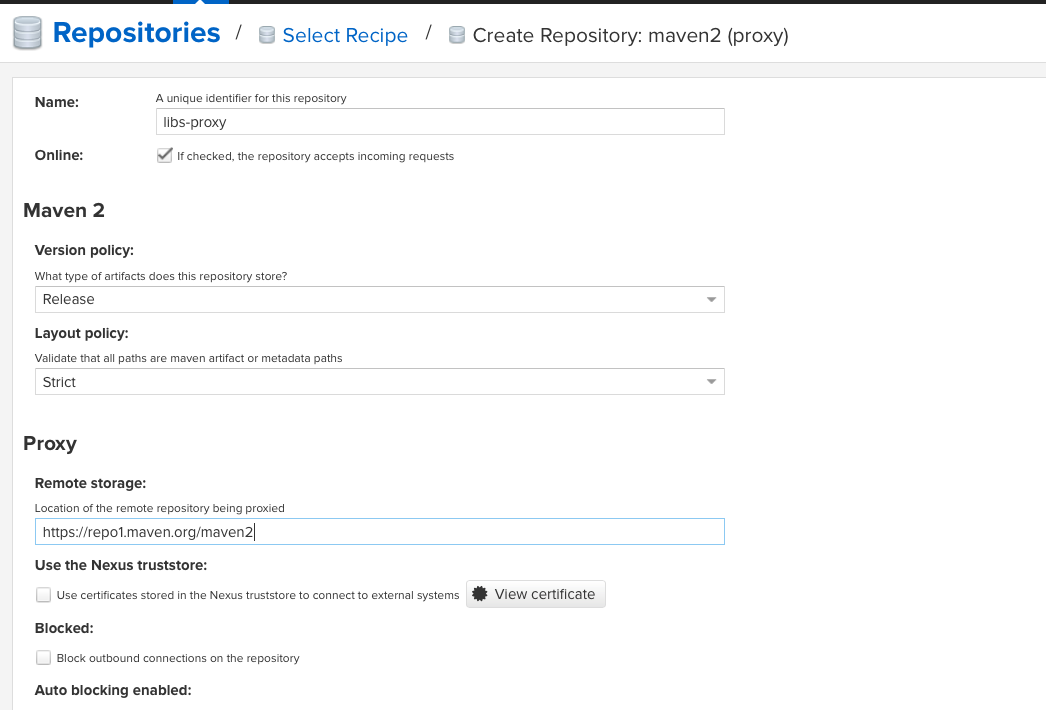
**  
proxy to Maven Central repo**

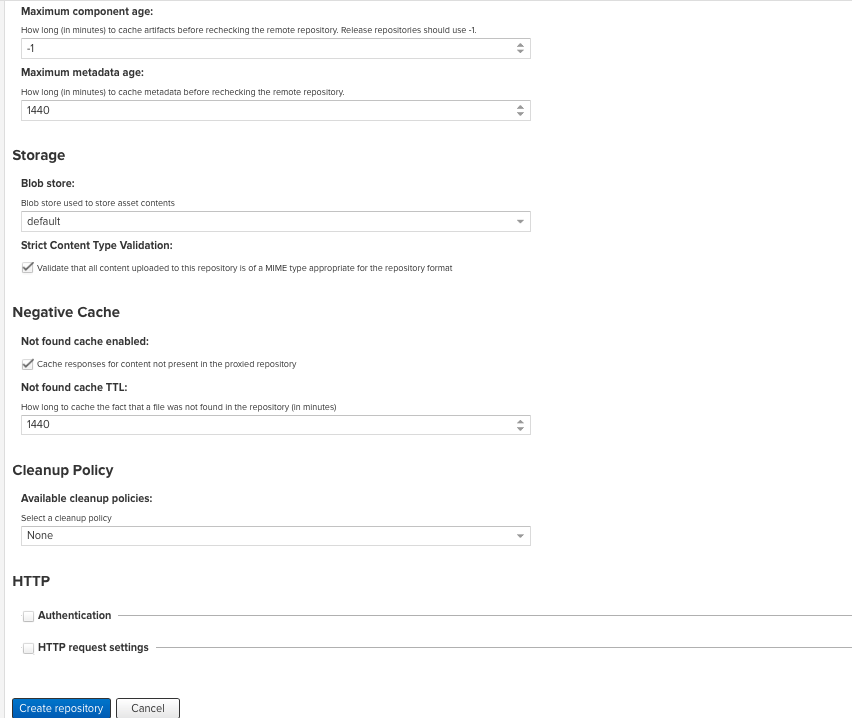
A repository that proxies everything you download from Maven Central. Next time you download the same dependency, it will be cached in your Nexus.

Create a new maven (proxy) repository and configure it like:

We’ll create a new Maven proxy in the Nexus Repository Manager UI.

1. Open the Nexus Repository Manager user interface.
2. Click Administration /var/folders/mn/1ssdrg7172d358h5xw6k94pm0000gn/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/admin_icon.pngin the top navigation menu, and then select Repositories.
3. Click Create repository, and then choose the maven2 (proxy) recipe from the list.
4. Add the following text in the required fields:
   1. Name: maven-proxy-test
   2. Remote storage URL: <https://repo1.maven.org/maven2>
5. Click Create repository to complete the form.





You can create as many repos as you need (like proxies to other public repos) and group them all in the group repo.

**Configuring your clients and projects to use your Nexus repos**

Put this in your ~/.m2/settings.xml file. This will configure the credentials to publish to your hosted repos, and will tell your mvn to use your repo as a mirror of central:

<?xml version="1.0" encoding="UTF-8"?>

<settings xmlns="http://maven.apache.org/SETTINGS/1.1.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.1.0 http://maven.apache.org/xsd/settings-1.1.0.xsd">

<servers>

<server>

<id>central</id>

<username>admin</username>

<password>admin123</password>

</server>

<server>

<id>snapshots</id>

<username>admin</username>

<password>admin123</password>

</server>

</servers>

<mirrors>

<mirror>

<id>central</id>

<name>central</name>

<url>http://localhost:8081/repository/libs-proxy/</url>

<mirrorOf>\*</mirrorOf>

</mirror>

</mirrors>

</settings>

And now configure your projects.

If you want only to download dependencies from Nexus, put this in the pom.xml:

<repositories>

<repository>

<id>central</id>

<url>http://localhost:8081/repository/libs-proxy/</url>

</repository>

</repositories>

And if you want also to publish your project, add:

<distributionManagement>

<snapshotRepository>

<id>snapshots</id>

<url>http://localhost:8081/repository/mithuntechno-snapshots/</url>

</snapshotRepository>

<repository>

<id>central</id>

<url>http://localhost:8081/repository/mithuntechno-releases/</url>

</repository>

</distributionManagement>

Now if you run in your projects:

mvn clean package

# or

mvn clean deploy

Deploy third party library to Remote Repo

mvn deploy:deploy-file -DgroupId=com.oracle -DartifactId=ojdbc14 -Dversion=ojdbc14 -DgeneratePom=true -Dpackaging=jar -DrepositoryId=central -Durl=http://ec2-35-154-213-177.ap-south-1.compute.amazonaws.com:8081/repository/libs-release/ -Dfile=ojdbc14.jar

**Note:** The "repositoryId" property value is *not* a Nexus repository ID. it is the <id> element value of a <server> section in your settings.xml file which has the credentials needed for deployment. If these credentials are not configured you will see an error like: "Return code is: 401, ReasonPhrase: Unauthorized."

<servers>  
...  
  <server>  
      <id>central</id>  
      <username>admin</username>  
      <password>admin123</password>  
    </server>  
  </servers>