1. **上传jar到私服仓库**

**方式1**：**直接通过操作nexus页面完成：**

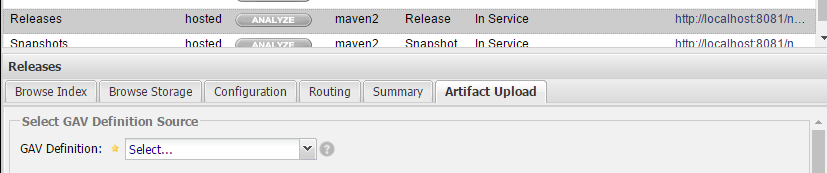
步骤：

1. 登录nexus仓库(用户名：admin，密码：admin123)

网页地址如：我本机的为http://192.168.7.52:8081/nexus/

2.点击左边Views/Repositorties菜单的Repositories选项

3.单击右侧仓库列表中Releases或3rd party仓库，看到下方页面上有Artifact Upload选项



4.通过下方的GAV Definition下拉列表 选择 GAV Parameters选项,然后可以设置要上传的jar的Group,Artifact,Version,Packaging,仓库用它们来生成具体的dependency

1. 点击Select Artifact(s) to Upload... 选择要上传的jar文件,会自动补全上面的group等信息，

不过可能不是我们需要的

1. 之后点击Add Artifact，最后点击Upload Artifact(s)进行上传

方式1好像不能把snapshots版本的jar上传到snapshot仓库

**方式2**：**通过mvn deploy命令进行上传jar**

步骤：

1. 在maven的settings.xml文件中加入认证信息,maven在上传jar到私服时需要提供认证，否则上传会被拒绝

<servers>

<server>

<id>nexus-releases</id>

<name>release reap</name>

<username>admin</username>

<password>admin123</password>

</server>

<server>

<id>nexus-snapshots</id>

<name>snapshot reap</name>

<username>admin</username>

<password>admin123</password>

</server>

</servers>

用户名和密码可以自己在nexus私服的网站上面创建，然后赋予权限，这里是默认的

2.在需要上传的jar对应的项目的pom.xml中增加以下内容：

<distributionManagement>

<repository>

<id>nexus-releases</id>

<name>release reap</name>

<url>http://localhost:8081/nexus/content/repositories/releases/</url>

</repository>

<snapshotRepository>

<id>nexus-snapshots</id>

<name>snapshot reap</name>

<url>http://localhost:8081/nexus/content/repositories/snapshots</url>

</snapshotRepository>

</distributionManagement>

注：id需要和前面settings.xml文件中的一致，url为上传仓库的具体地址



Maven会根据id中的releases,snapshots这些关键字，把release或snapshot版本的jar上传到对应的仓库中

3. 执行mvn clean deploy命令，这时就会把jar上传到私服仓库了

以上就是两种jar上传方式

**注：如果项目的pom.xml文件中配置了parent,则还需要把parent项目也上传到私服，否则可能在下载时失败**

1. **从仓库中下载jar**

**方式1：**

在需要该jar的项目的pom.xml文件中配置私服仓库

<repositories>

<repository>

<id>my-repo</id>

<name>Nexus Local</name>

<url>http://localhost:8081/nexus/content/groups/public/</url>

<releases>

<enabled>true</enabled>

</releases>

<snapshots>

<enabled>true</enabled>

</snapshots>

</repository>

</repositories>

然后就可以添加相应的dependency了

**注意：需要确定配置的这个仓库没有被mirror拦截，如果settings.xml中配置了mirror拦截的地址为<mirrorOf>\*</mirrorOf>，则需要使用<mirrorOf>\*,!my-repo</mirrorOf>代替,配置成<mirrorOf>central</mirrorOf>则最好了，所有自己配置的远程仓库都不用担心被拦截**

1. **配置好maven的jar包的下载顺序**

**如：先从阿里云下载，阿里云没有再从私服仓库下载**

**方式1：**

经过我测试和网上的资料，初步得出结论：

Maven搜索仓库的优先级为：本地仓库 > 在settings.xml中<profiles>里配置的仓库 > 在

pom.xml中配置的仓库

在pom.xml中配置的仓库，越上方的越优先被搜索

Maven在搜索远程仓库前会去判断该仓库有没有mirror，如果有则转向mirror中的url对应的仓库

所以我们可以在maven的settings.xml中配置:

<profiles>

<profile>

<id>dev</id>

<repositories>

<repository>

<id>aliyun</id>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

<releases>

<enabled>true</enabled>

</releases>

<snapshots>

<enabled>true</enabled>

</snapshots>

</repository>

</repositories>

<pluginRepositories>

<pluginRepository>

<id>aliyun</id>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

<releases>

<enabled>true</enabled>

</releases>

<snapshots>

<enabled>true</enabled>

</snapshots>

</pluginRepository>

</pluginRepositories>

</profile>

</profiles>

<!--激活配置-->

<activeProfiles>

<activeProfile>dev</activeProfile>

</activeProfiles>

来让maven优先从阿里云的仓库下载jar,然后在pom.xml中配置：

<repositories>

<repository>

<id>my-repo</id>

<name>Nexus Repository</name>

<url>http://localhost:8081/nexus/content/groups/public/</url>

<releases>

<enabled>true</enabled>

</releases>

<!--snapshots默认是关闭的,需要开启 -->

<snapshots>

<enabled>true</enabled>

</snapshots>

</repository>

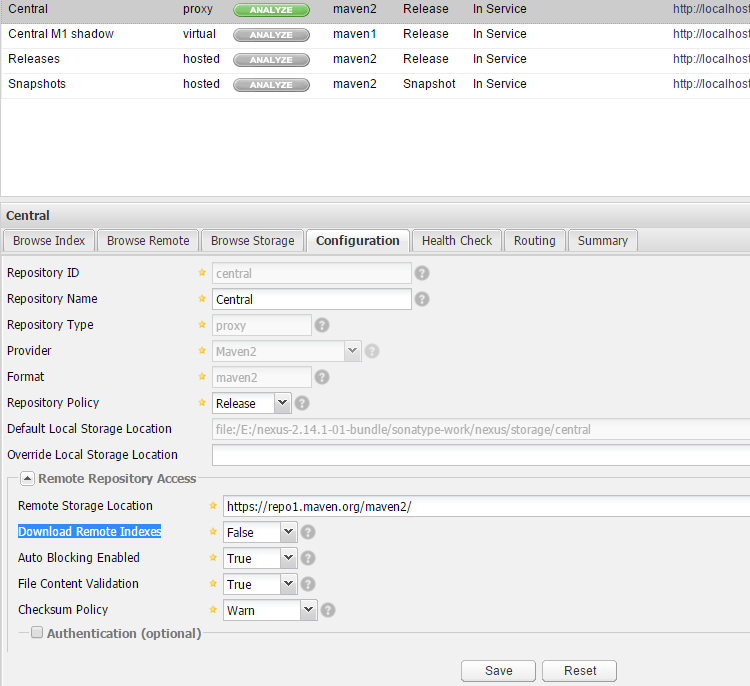
</repositories>

来下载我们自己上传到私服的jar

**方式2：**

还有种更好的办法，就是让我们的私服仓库只用来保存我们自己的jar,而不包括其他的jar,这样就算先从私服仓库搜索也不能够搜索到我们上传以外的jar,就不会出问题了。

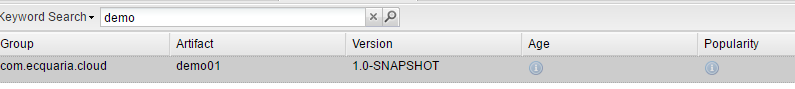
具体配置就是在搭建私服仓库时，不设置central仓库的Download Remote Indexes为true



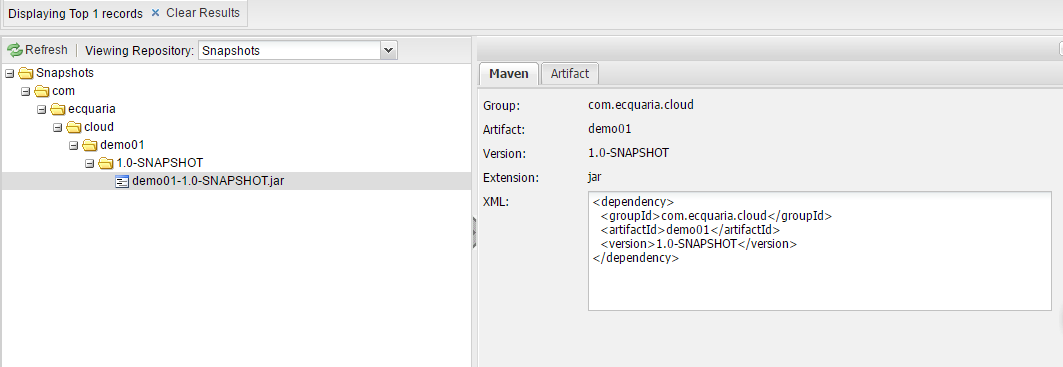
这样如果jar在私服仓库中没有，就不会去中央仓库下载下来，保存到私服仓库了,这样私服仓库里就只剩下我们自己的jar

1. 从私服仓库删除jar

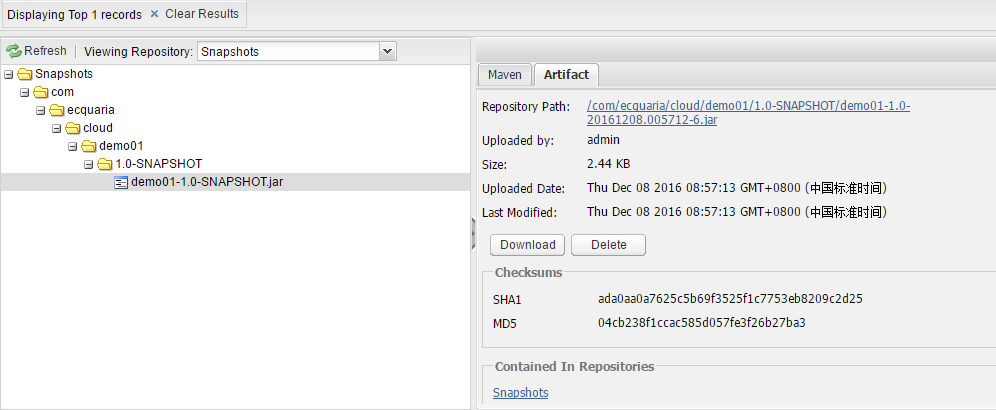
操作： 首先搜索找到该jar



点击具体要删除的那个，然后下方会出现如下界面：

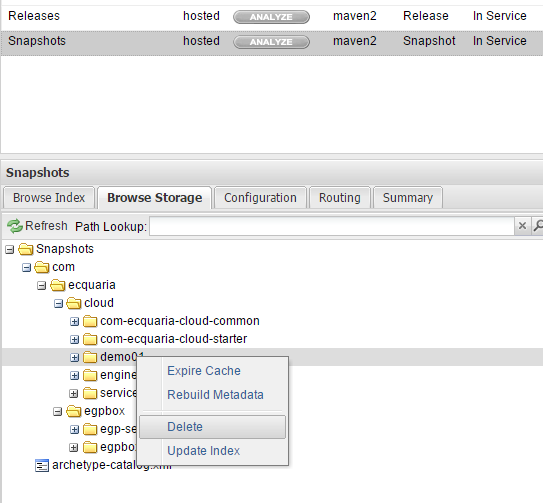


点击右边的Artifact按钮，就可以看到删除按钮了



点击delete，这时jar就删除成功了，如需删除该jar的所有信息，则还需执行以下操作：

然后去该jar所在仓库里的browse storage里删除其他相关信息，如下所示：



点击expire cache再点击delete 最后右击仓库，更新下索引就可以了

