Maven私服搭建

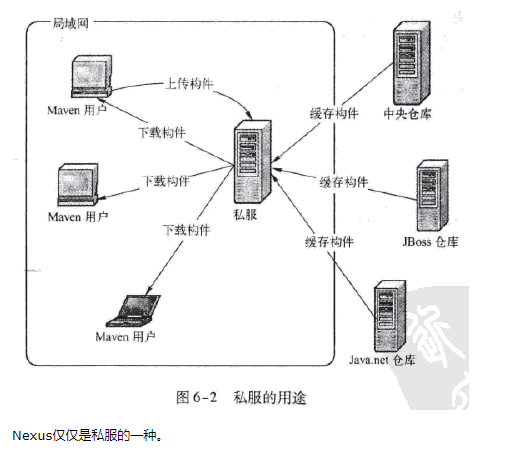
# 一、准备软件包

1、Nexus介绍以及下载：

1.1Nexus介绍

Nexus是Maven仓库管理器。

一些无法从外部仓库下载的构件，例如内部的项目还能部署到私服上，以便供其他依赖项目使用。为了节省带宽和时间，在局域网内架设一个私有的仓库服务器，用其代理所有外部的远程仓库。当本地Maven项目需要下载构件时，先去私服请求，如果私服没有，则再去远程仓库请求，从远程仓库下载构件后，把构件缓存在私服上。这样，即使暂时没有Internet链接，由于私服已经缓存了大量构件，整个项目还是可以正常使用的。同时，也降低了中央仓库的负荷。



更细致的了解可以访问该网页：<https://blog.sonatype.com/>

1.2Nexus下载

下载地址：<https://www.sonatype.com/download-oss-sonatype>



2、Maven介绍及下载：

2.1maven介绍

（1）、优秀的构建工具

通过简单的命令，能够完成清理、编译、测试、打包、部署等一系列过程。同时，不得不提的是，Maven是跨平台的，无论是在Windows、还是在Linux或Mac上，都可以使用同样的命令。

（2）、依赖管理工具

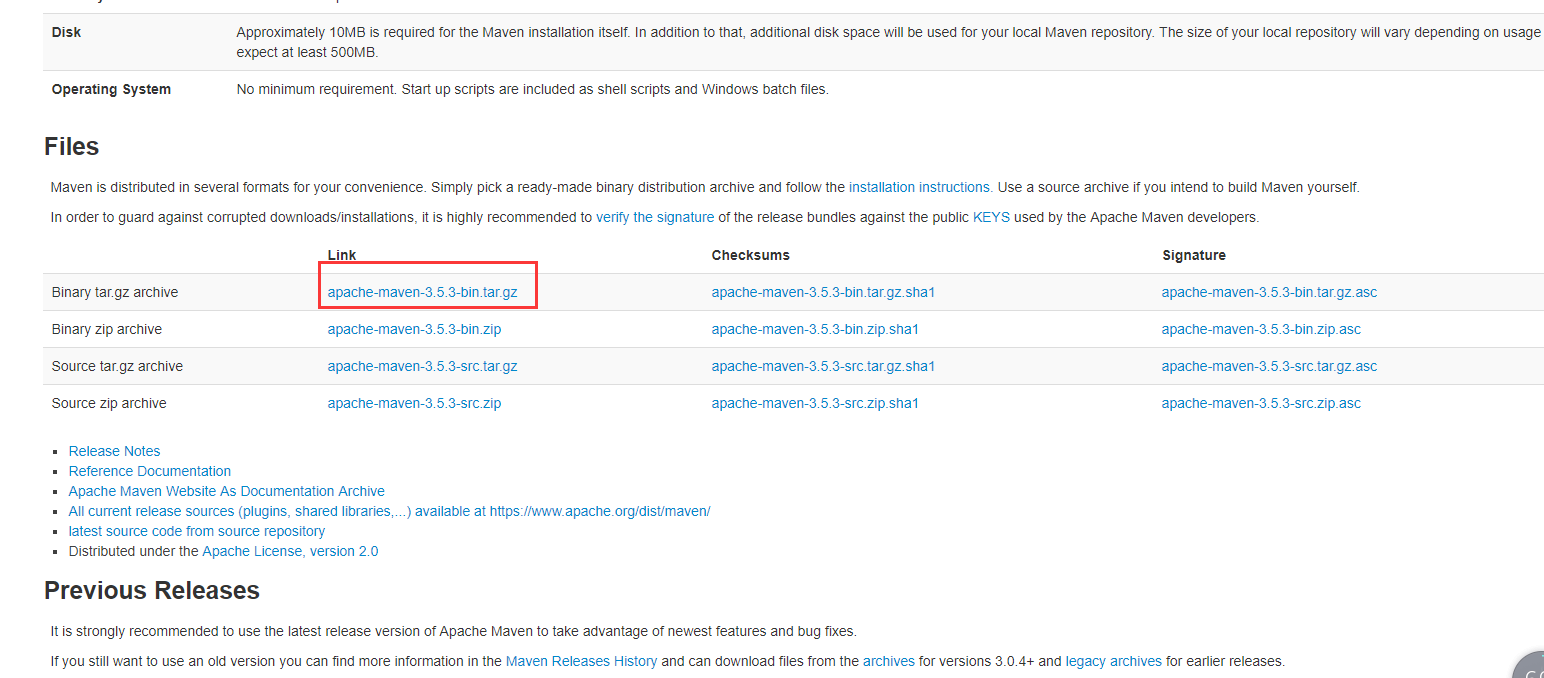
项目依赖的第三方开源类库，都可以通过依赖的引入到项目中来。代替了原来需要首先下载的第三方jar，再加入到项目中的方式。从而更好的解决了合作开发中依赖增多、版本不一致、版本冲突、依赖臃肿的问题。Maven通过坐标系统准确的定位每一个构件，即通过坐标找到对应的java类库。

（3）、项目信息管理工具

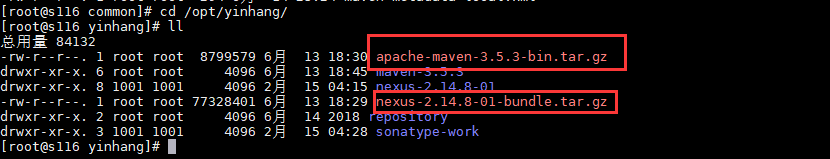
能够管理项目描述、开发者列表、版本控制系统地址、许可证等一些比较零散的项目信息。除了直接的项目信息，通过Maven自动生成的站点，以及一些已有的插件，还能够轻松获得项目文档、测试报告、静态分析报告、源码版本、日志报告等非常具有价值的项目信息。

2.2maven下载

下载地址：<http://maven.apache.org/download.cgi>



3、将下载的包传到linux服务器上



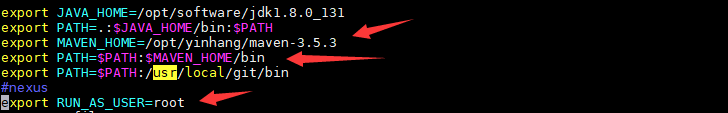
4、解压

tar -zxvf apache-maven-3.5.3-bin.tar.gz

tar -zxvf nexus-2.14.8-01-bundle.tar.gz

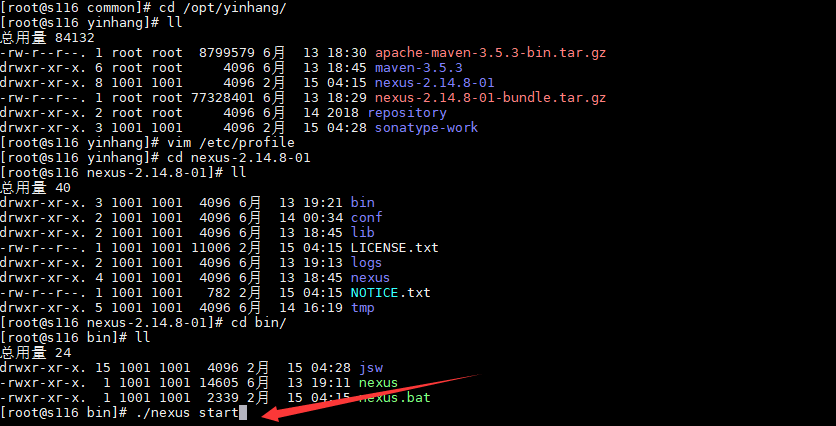
5、添加环境变量

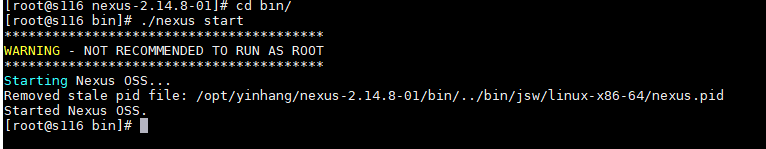
vim /etc/profile



# 二、启动Nexus

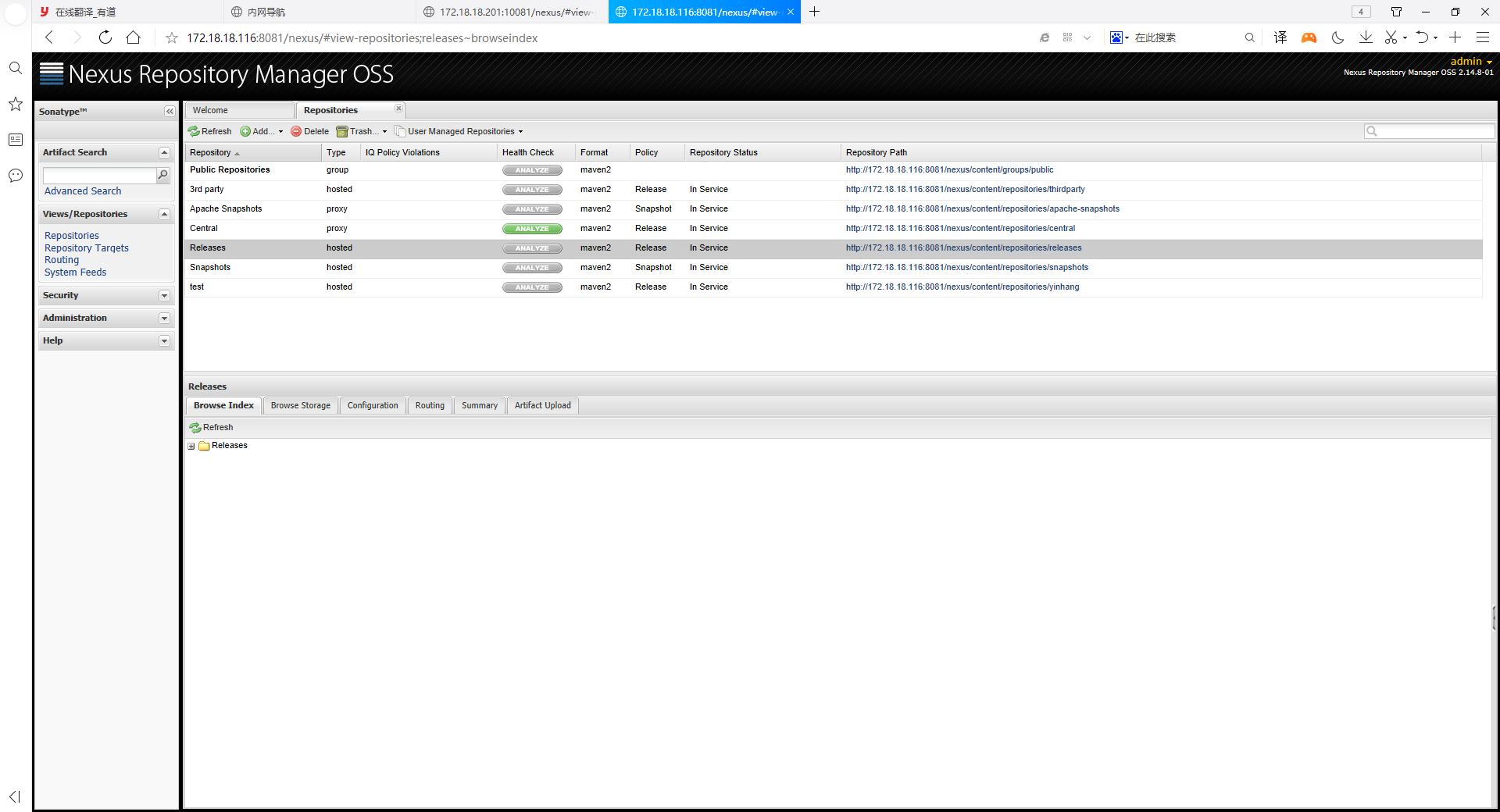
1、执行命令启动服务





2、进入Nexus浏览

启动服务之后，浏览器中输入：<http://172.18.18.116:8081/nexus/，默认用户名和密码是admin和admin123>



**Nexus仓库类型介绍**

nexus的仓库类型分为以下四种：

        group:仓库组，用了合并多个hosted/proxy仓库，当你的项目希望多个repository使用资源时就不需要多次引用了，只需要引用一个group即可。

        hosted：宿主，本地仓库，通常我们会部署自己的构建到这一类仓库。

        proxy：代理仓库，通常被用来代理远程的公共仓库，如maven中央仓库。

        virtual：虚拟

主要介绍一下三个本地仓库：

**Releases:**用来部署管理内部的发布版本构件的宿主类型仓库，这里存放我们自己项目中发布的构建,通常是Release版本的, 比如我们自己做了一个FTP Server的项目, 生成的构件为ftpserver.war,我们就可以把这个构建发布到Nexus的Releases本地仓库。

**Snapshots:**用来部署管理内部的快照版本构件的宿主类型仓库，它的目的是让我们可以发布那些非release版本, 非稳定版本,比如我们在trunk下开发一个项目,在正式release之前你可能需要临时发布一个版本给你的同伴使用, 因为你的同伴正在依赖你的模块开发,那么这个时候我们就可以发布Snapshot版本到这个仓库, 你的同伴就可以通过简单的命令来获取和使用这个临时版本。

**3rd Party:**无法从公共仓库获得的第三方发布版本的构件仓库，比如有些构件在中央仓库是不存在的.比如你在中央仓库找不到Oracle 的JDBC驱动, 这个时候我们就需要自己添加到3rdparty仓库。

# 三、修改配置文件

1、修改settings.xml



2、修改IDEA的pom文件，添加distributionManagement

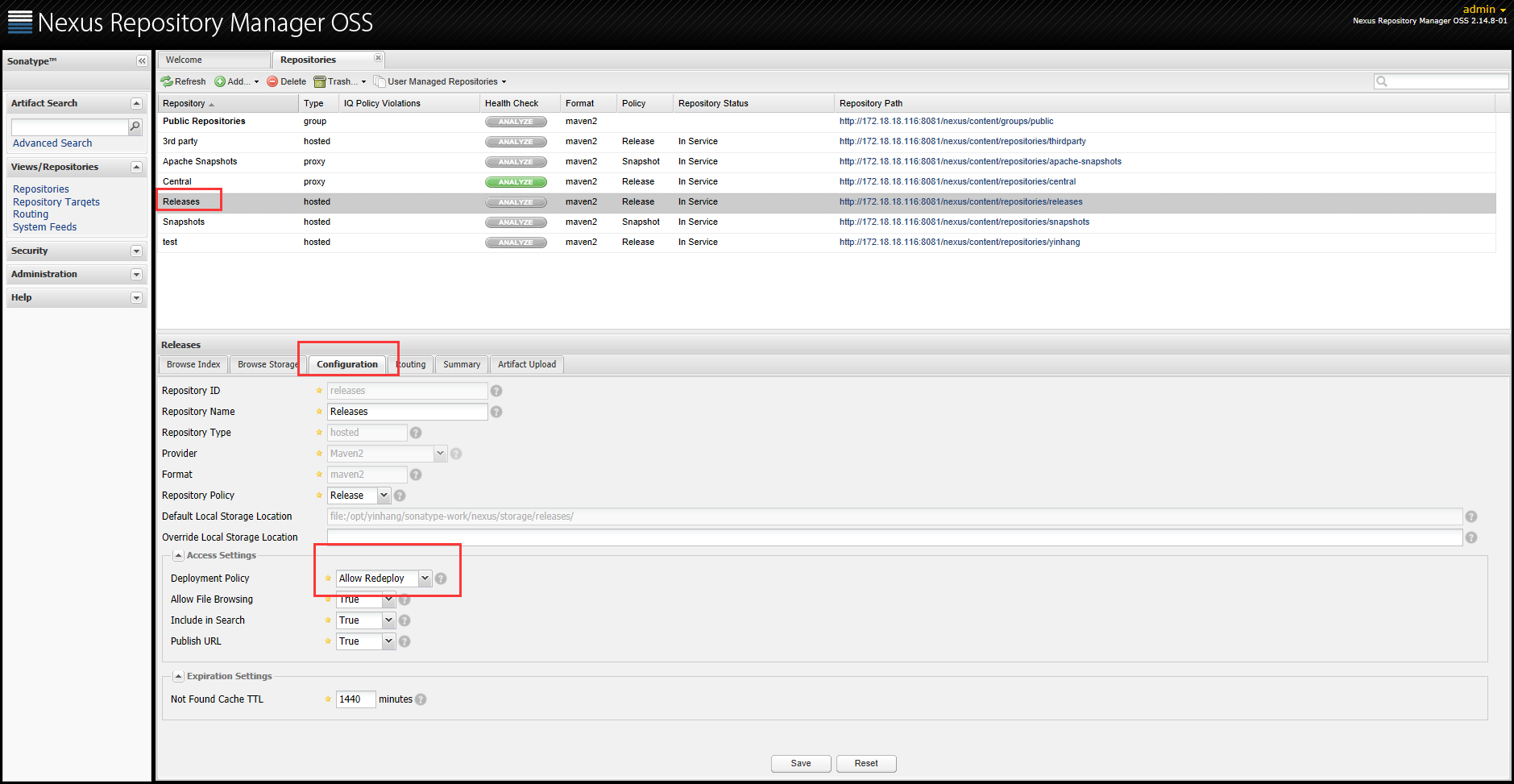
作用是可以将包上传到maven私服库

pom文件中的id要和settings中的id相同



3、修改web页面设置

将Disable Redeploy 改成Allow Redeploy

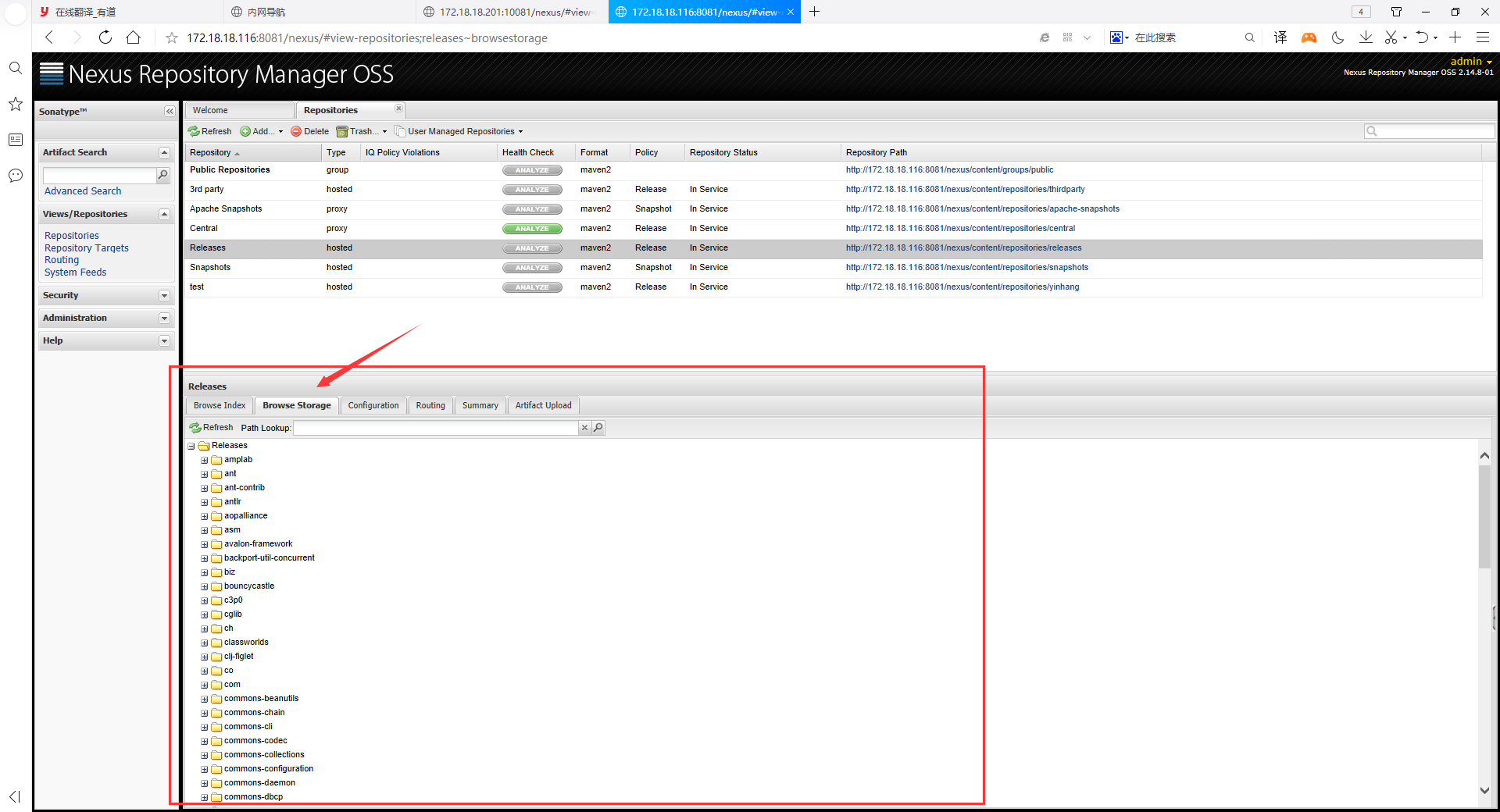


# 四、使用maven私服

1、加载maven私服里面的releases仓库，在项目文件夹里使用git命令



这样maven私服仓库里面的包就会下载到你的本地库，因为你的本地库在你的settings文件里面已经配置好了。



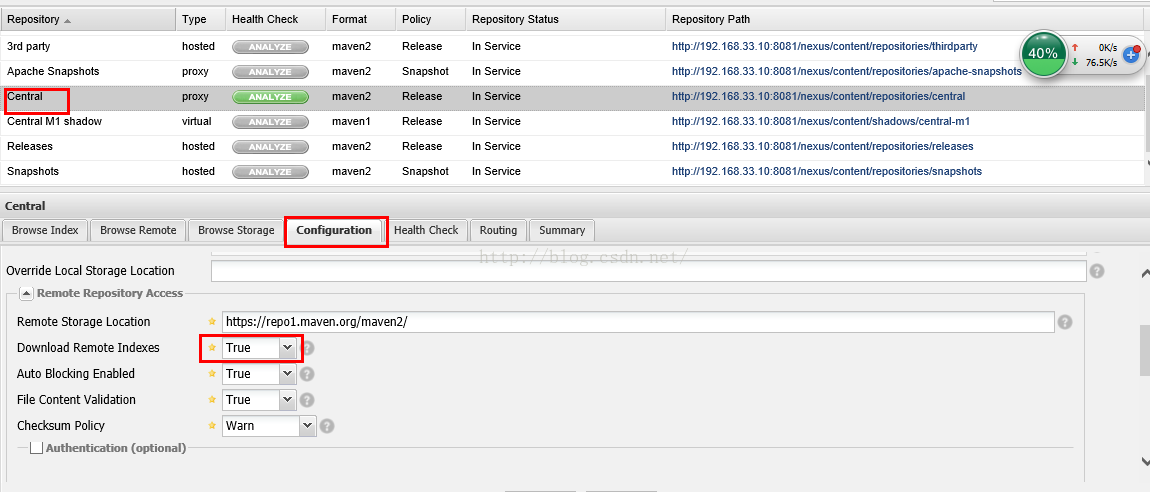
2、将项目的包上传到maven私服releases仓库



上传之后在去releases仓库看一下包的更新时间有没有上传成功就OK了

3、下载和更新索引（选用）

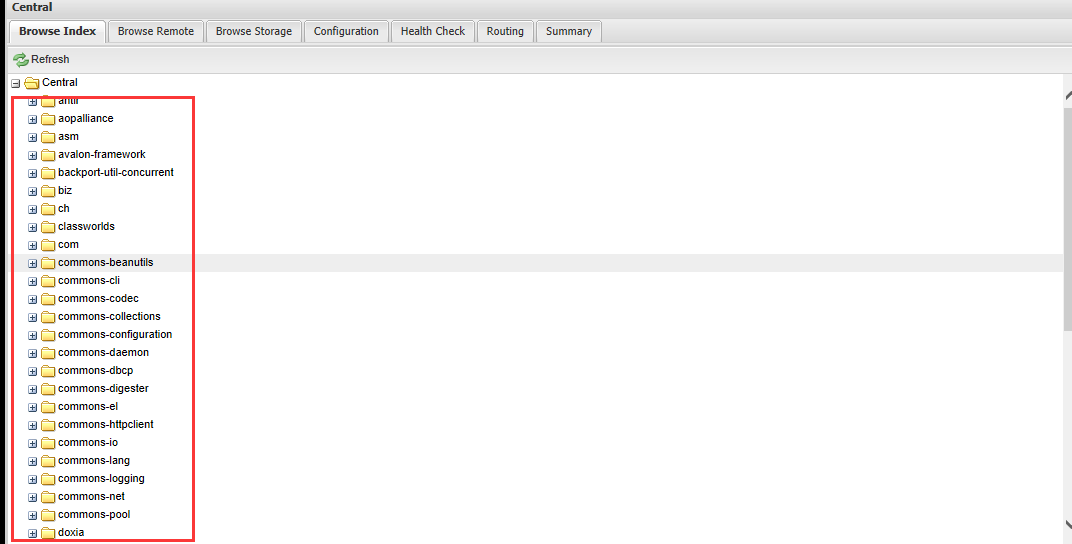
把Centra仓库，点击仓库下面的configuration下把DownloadRemote Indexes修改为true。如下图：



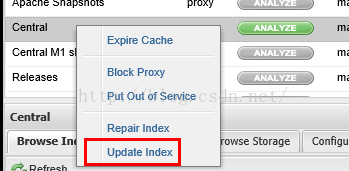
然后在Central仓库上右击，选择Pepari Index，这样Nexus就会去下载索引文件。

重新启动一下nexus。

能够很快的把索引下载过来。如图：

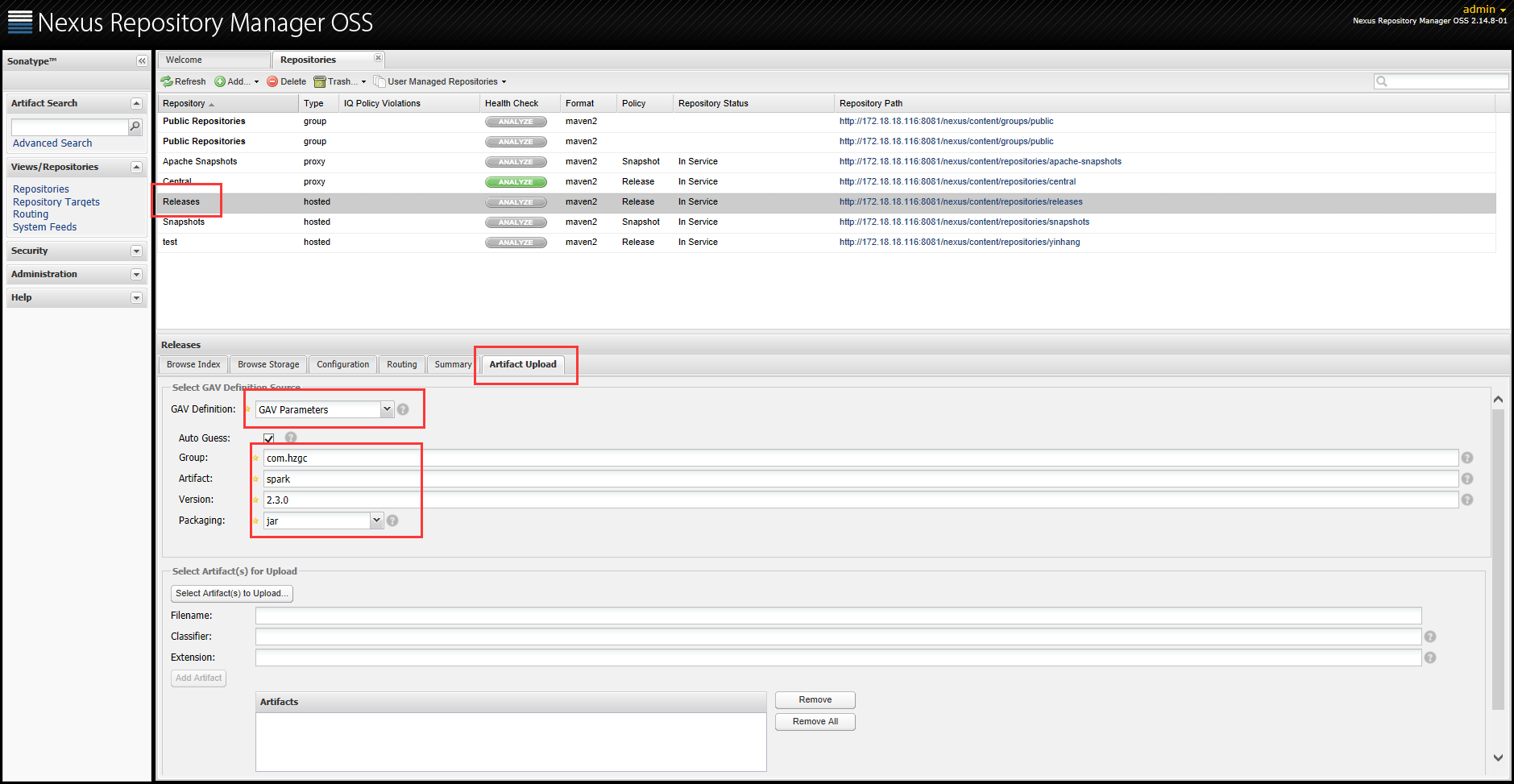


如果中间添加了新的jar包想更新索引，点中右击update index就行了。如图：

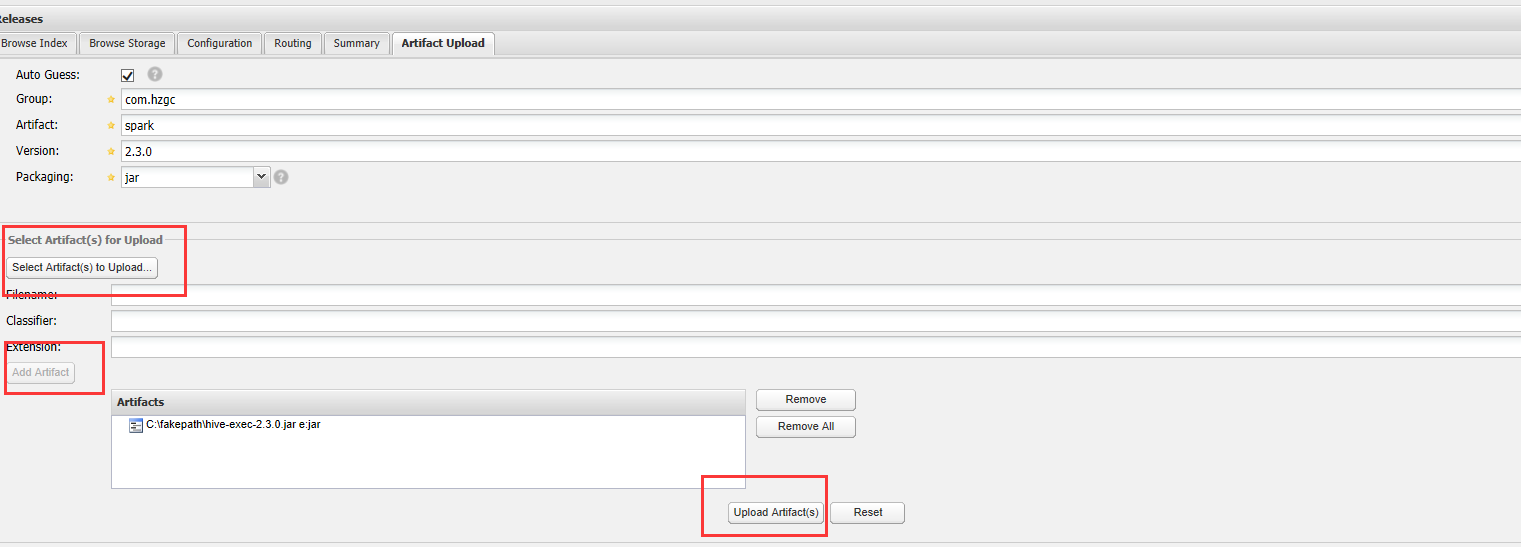


4、上传jar包到maven私服

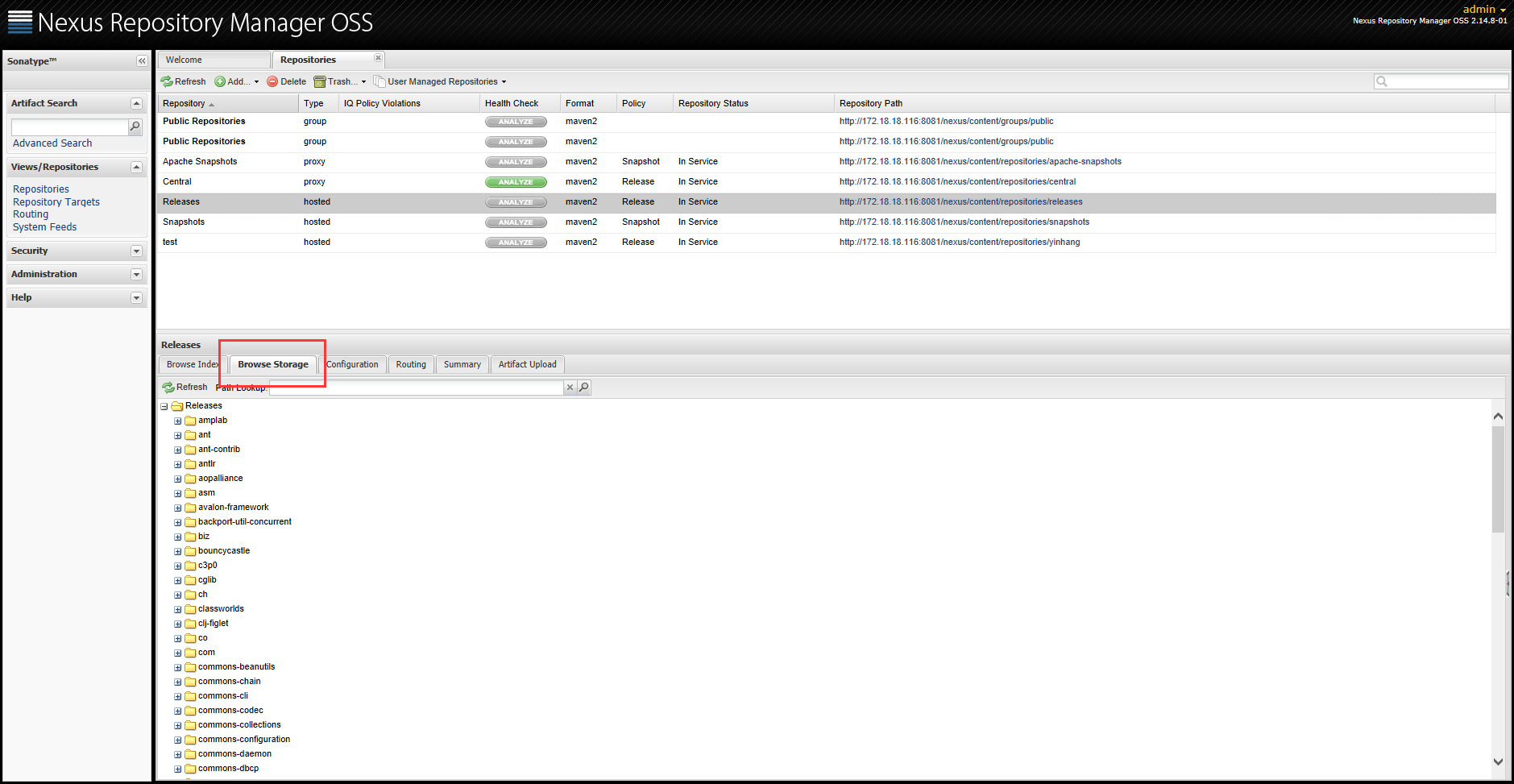
在打开的标签页中，选择releases，我们自己的jar包都要上传到releases中。GAV Definit ion 选择：GAVParameters。   
依次填写Group、Artifact、Version，分别对应pom.xml文件中的groupId、artifactId、version。Packaging选择jar



填写完成之后，点击 selectArtifact(s) to upload选择需要上传的jar，下方的Filename、Extension会自动填充, 然后点add Artifact。最后点Upload Artifact(s)。



上传成功之后，可以在BrowseStorage中，查看到刚刚上传的jar包。



# 附件：

mvn命令补充：

* **不要忘了clean：** clean能够保证上一次构建的输出不会影响到本次构建。
* **使用deploy而不是install：** 构建的SNAPSHOT输出应当被自动部署到私有Maven仓库供他人使用。
* **使用-U参数：** 该参数能强制让Maven检查所有SNAPSHOT依赖更新，确保集成基于最新的状态，如果没有该参数，Maven默认以天为单位检查更新，而持续集成的频率应该比这高很多。
* **使用-e参数：**如果构建出现异常，方便分析错误原因。
* **使用-Dmaven.repo.local参数：**如果持续集成服务器有很多任务，每个任务都会使用本地仓库，下载依赖至本地仓库，为了避免这种多线程使用本地仓库可能会引起的冲突，可以使用**-Dmaven.repo.local=/home/juven/ci/foo-repo/**这样的参数为每个任务分配本地仓库。
* **使用-B参数：**该参数表示让Maven使用批处理模式构建项目，能够避免一些需要人工参与交互而造成的挂起状态。

附赠 myeclipse配置和用法

https://www.cnblogs.com/zhongshengzhen/p/nexus\_maven.html