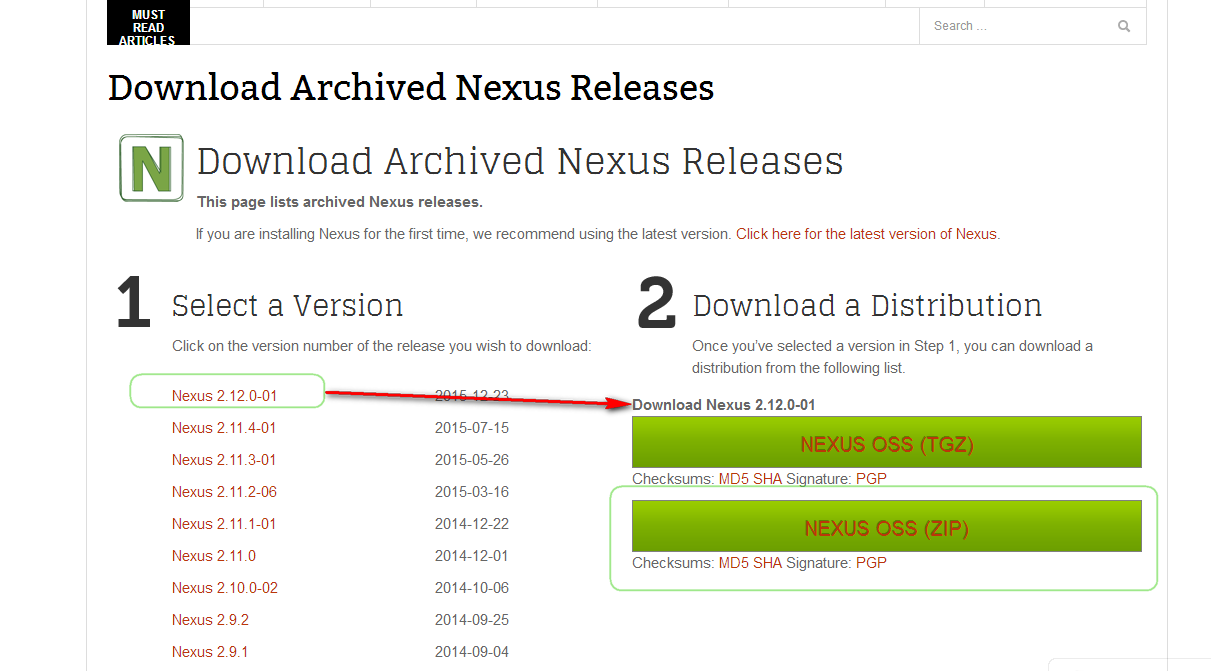
**1 访问nexus官方网站**

http://www.sonatype.org/nexus/，

下载Nexus 2.12.0-01-bundle.zip。

<http://www.sonatype.org/nexus/archived>

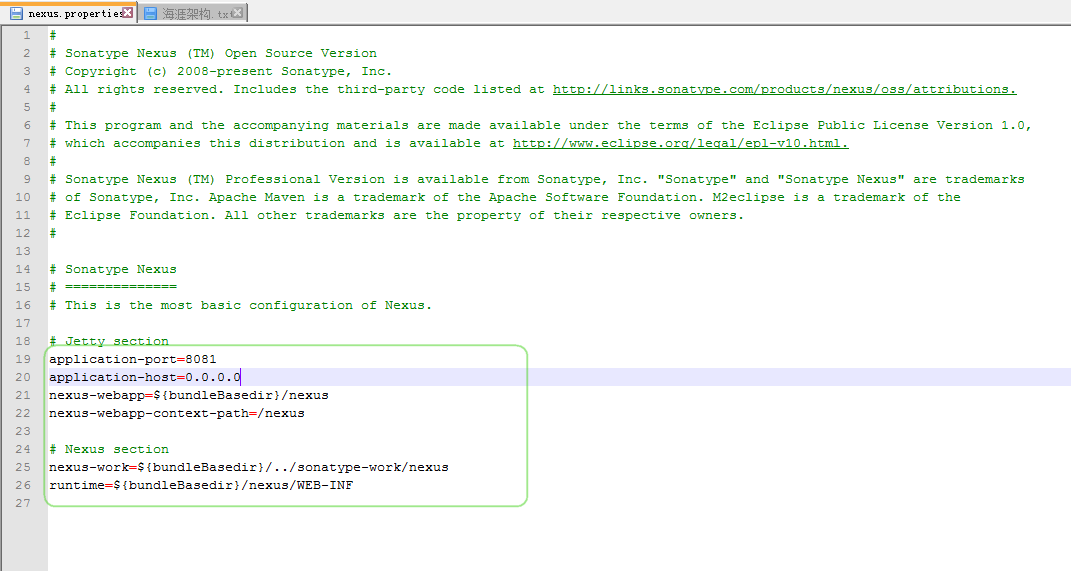
有两种安装包：一种是自带 jetty 容器的 Bundle 包，另一种是不带 web 容器的 war 包。



**2 开启jetty服务器**

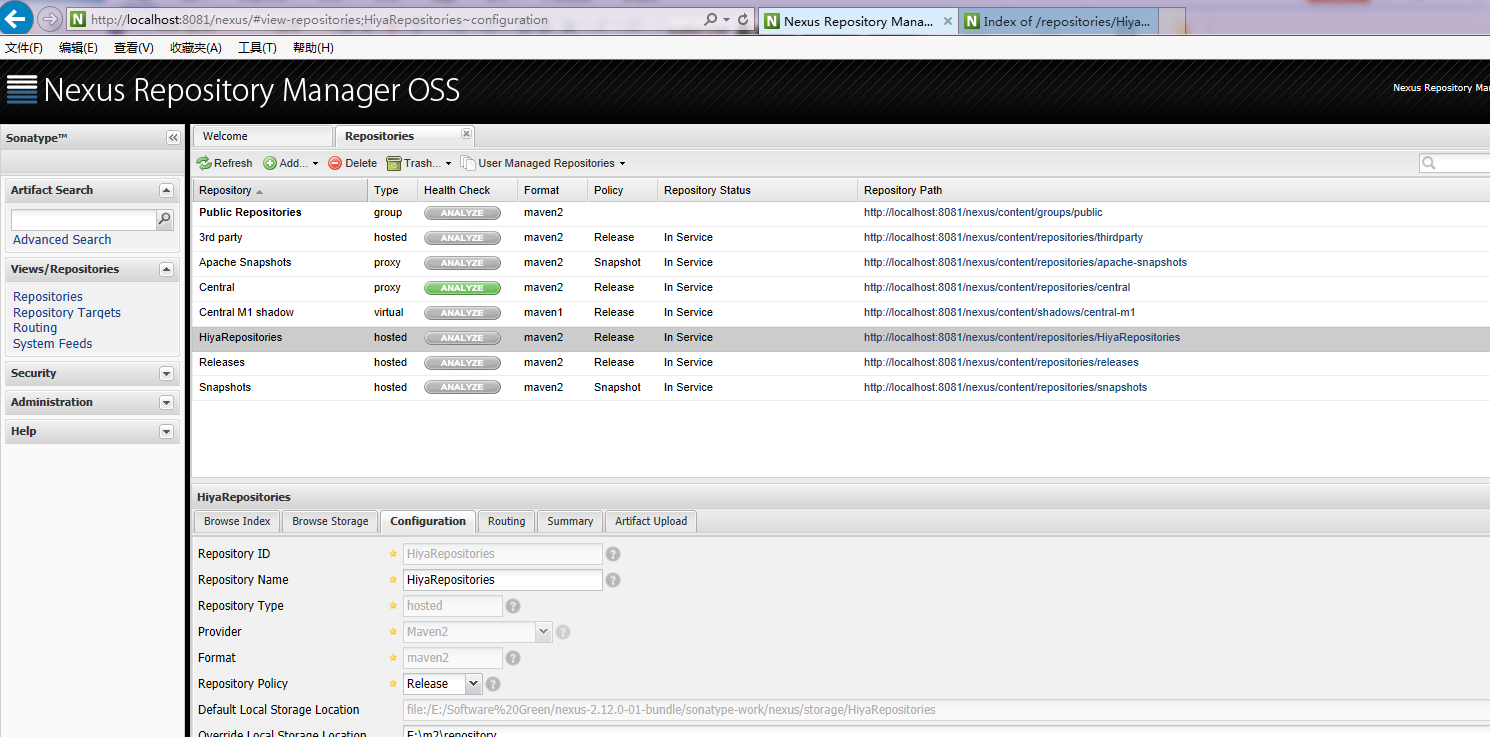
找到E:\SoftwareGreen\nexus-2.12.0-01-bundle\nexus-2.12.0-01\bin

\jsw\windows-x86-64，执行console-nexus.bat批处理文件，等待命令行执行到如下情况



**3 nexus管理页面**

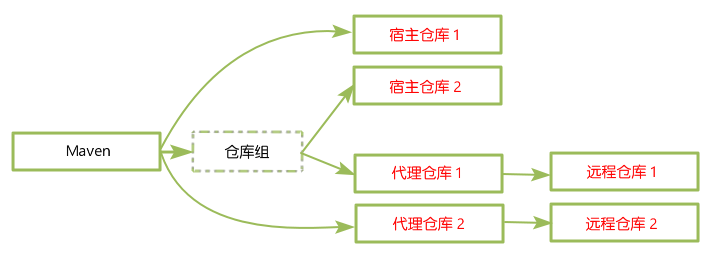
打开浏览器，输入http://localhost:8081/nexus/，点击页面右上方的Log in超链接，输入默认用户名密码admin/admin123，登录nexus管理页面。

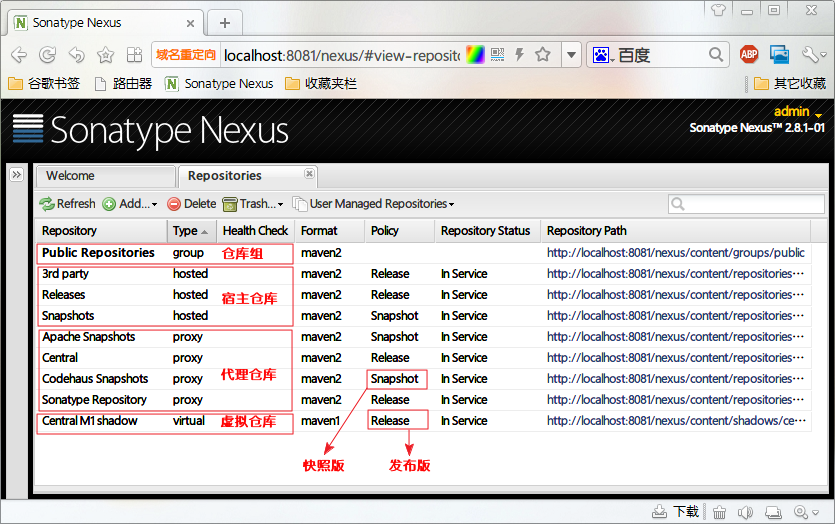


**4 仓库类型**

* hosted，本地仓库。通常我们会部署自己的构件到这一类型的仓库。比如公司/组织开发的项目
* proxy，代理仓库，它们被用来代理远程的公共仓库，如maven中央仓库或一些第三方公共仓库。
* group，仓库组，用来合并多个hosted/proxy仓库，当你的项目希望在多个repository使用依赖项时，无需引用多个仓库了，只需要引用一个类型为group的repository即可。

public repository默认包含本地仓库的Releases, snapshots和3rd party以及代理仓库的Maven Central. 你可以在Configuration配置页添加其他仓库到这个仓库组。



****

**5 仓库版本**

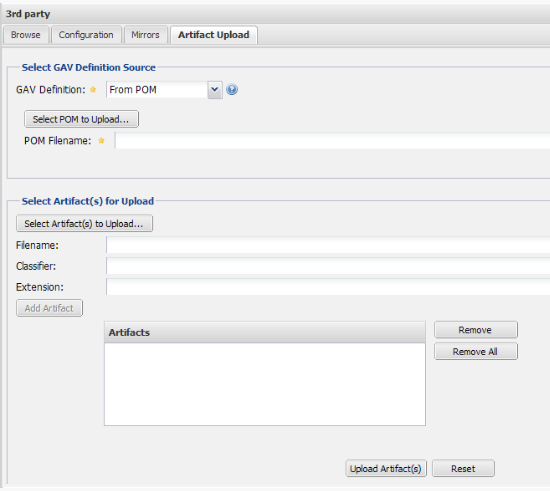
Nexus预定义了三种本地仓库，分别是**Releases, Snapshots, 3rd Party**. 下面分别介绍一下这三种类型仓库的作用

• Releases: 存放稳定版本的构件。比如我们完成了一个版本的下数组件的开发,就可以把它发布到这里。

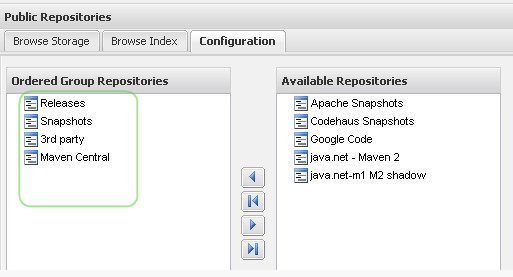
• Snapshots: 存放快照版本的构件。 比如一个下数组件在完成所有开发和测试工作之前，是不应该发布到release仓库的，但可能其他项目只需用到这个组件的某些接口，只要这些接口完成了开发并通过测试，就可以拿来使用, 从而实现多个项目并行开发。

• 3rd Party: 存放其他第三方构件。你可能会问， 不是有中央仓库和其他第三方公共仓库来管理这些依赖项了吗？没错。但由于某些开源项目出现的时间比maven要早，因此他们大都没有采用maven方式进行构建。 其他还包括包括一些非开源或使用maven 1.x构建的组件. 我们可以把这类组件通通添加到这里。

**6 上传构件**



**7 仓库组顺序**



这个是正确的配置顺序。其中左边的是正在使用的定义好的仓库，右边的是候选的仓库。在这个例子中，nexus下载构件的顺序是Releases -->Snapshots --> 3rd party --> Maven Central。

**8 什么是私服**

