**开机启动**

**sudo ln -s /opt/nexus-2.6.4/nexus-2.6.4-02/bin/nexus /etc/init.d/nexus**

**使用  service nexus status/start/stop  查看nexus服务状态、启动服务、停止服务等**

**使用 chkconfig nexus on/off 设置nexus服务开机自启动或者开机不启动**

[**centos7搭建nexus maven私服**](http://www.cnblogs.com/grey-wolf/p/6481166.html)

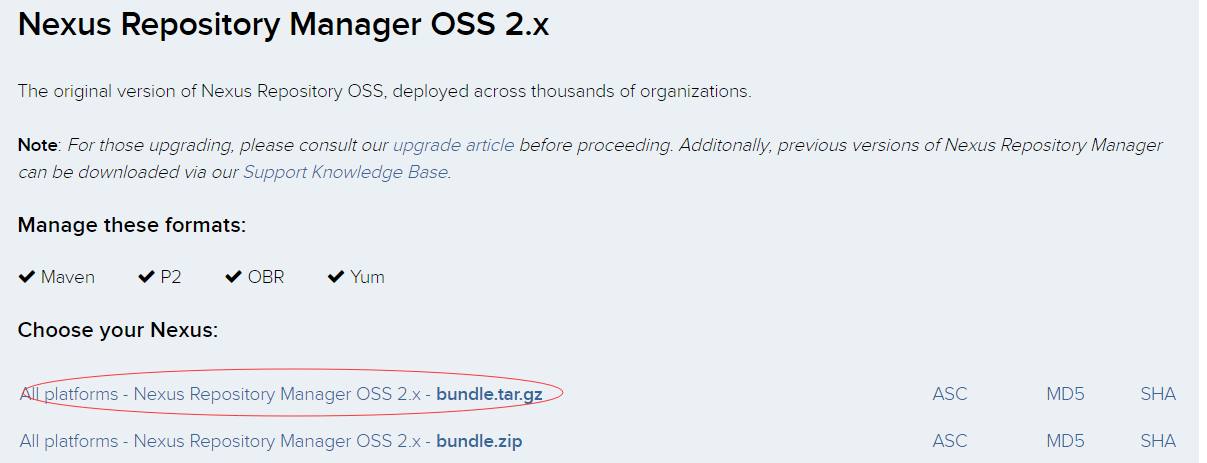
前置条件：

1.安装jdk，可参考 http://www.cnblogs.com/grey-wolf/p/6480489.html

2.nexus仓库管理器，分为两个版本，Nexus Repository Manager OSS 和 Nexus Repository Manager Pro。前者可以免费使用，相比后者，功能缺少一些，但是不影响我们搭建maven私服。

所以就选择OSS版本。

下载地址：https://www.sonatype.com/download-oss-sonatype，下载不下来可以去这里 http://download.csdn.net/detail/cctvckl/9766638



文档下载地址：

http://books.sonatype.com/nexus-book/pdf/nxbook-pdf.pdf

好了，让我们正式开始吧。

1. 安装nexus服务器

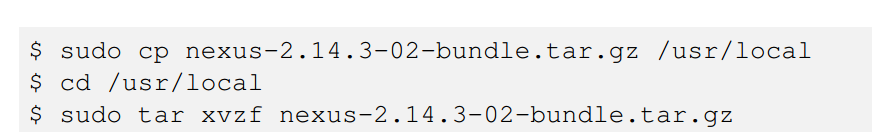
* 上传tar.gz包到指定目录（我这边假定是/home/ckl/upload）

　　rz  
　　mv nexus-2.14.3-02-bundle.tar.gz /home/ckl/upload/

* 解压到/usr/local  
  tar -zxvf /home/ckl/upload/nexus-2.14.3-02-bundle.tar.gz -C /usr/local/

**注意：**

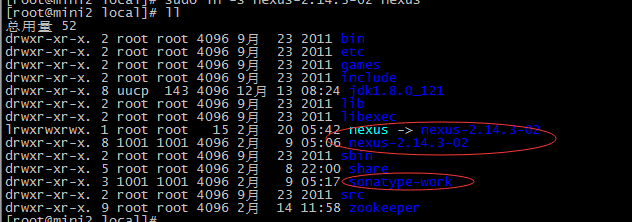
　　一定要解压到这个目录下，具体原因我也不知道，文档里是这么做的，我们也没必要自己找麻烦。



* 建立软链接

　　sudo ln -s nexus-2.14.3-02 nexus

* 查看/usr/local目录，结果如下：



* 设置环境变量（按官网文档的说法是可选，不过还是配上吧）

　　vim /etc/profile,在最后加上下面这一句,:wq保存退出

https://images2015.cnblogs.com/blog/519126/201702/519126-20170228203220563-27304370.png

* source /etc/profile 刷新刚设置的环境变量，使之生效
* 查看结果

https://images2015.cnblogs.com/blog/519126/201702/519126-20170228203408016-1955589360.png

1. 启动服务器

（博客园的这个数字序号不知道怎么才能设成2.有知道的朋友麻烦告知下）

　　前台启动：

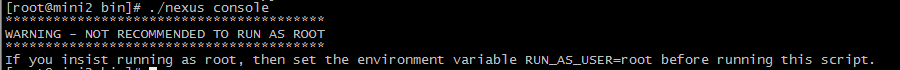
　　cd /usr/local/nexus  
　　./bin/nexus console

　　以后台进程启动：

　　./bin/nexus start  
　　查看输出的日志：

　　tail -f logs/wrapper.log

　　好了，我们来前台启动一下。咦，没起来，提示：



　　意思是坚持使用root用户运行的话，需要设置一个环境变量.

　　vim /etc/profile

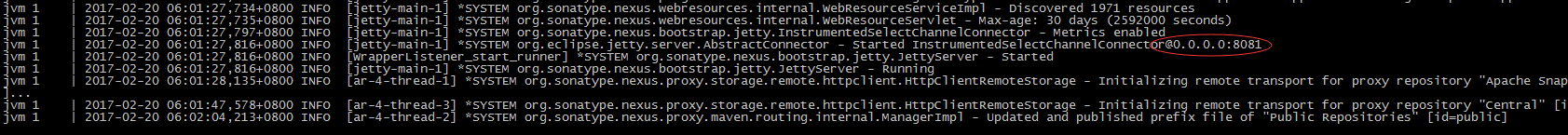
　　结尾加上

　　export RUN\_AS\_USER=root

　　保存退出。

　　刷新：source /etc/profile

　　再次运行，结果如下：

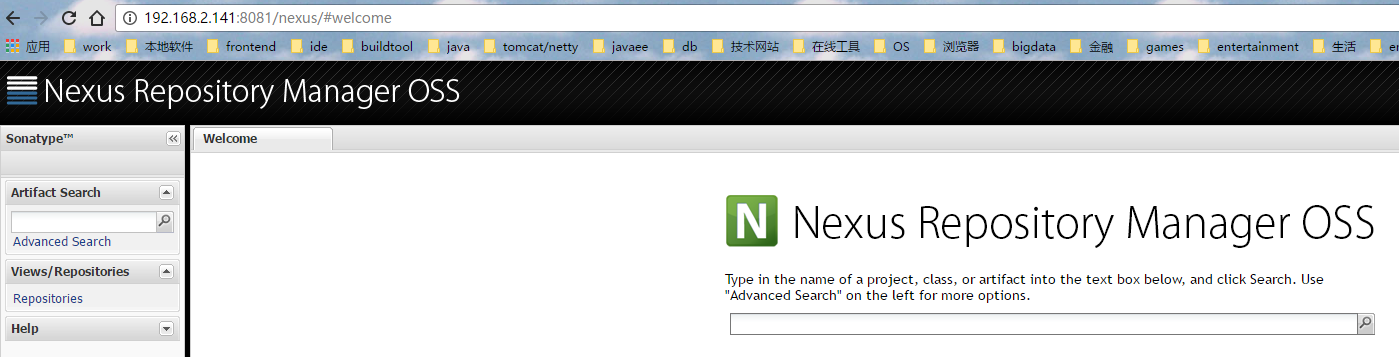


从红框标识及相关文档，可以知道，进程在8081端口启动。

好了，我们在浏览器上看看能不能访问吧，路径：

http://192.168.2.141（换成你的ip）:8081/nexus.

 结果如图所示：

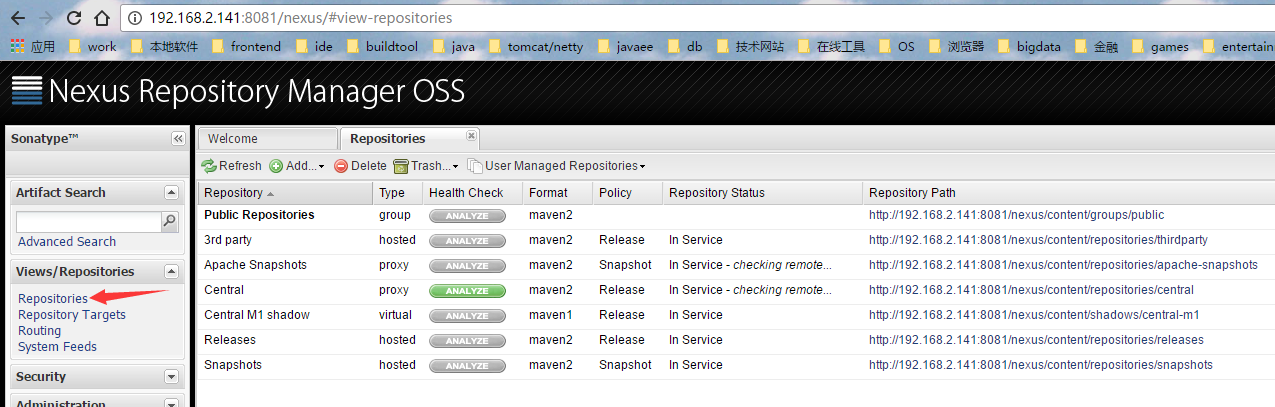


点击右上角登录：

默认用户名为admin，密码admin123

**3.搭建私服**

**3.1**界面元素介绍

****

登录后，点击左侧Repositories，界面如上图所示。

右侧的列表中，可以看到nexus预设的几个仓库。

第一个public Repositories，类型为group，这个简单理解为仓库的集合，下面的仓库就是可以加入到这个集合的元素。

对外可以只提供一个单独的url，如上图显示为：http://192.168.2.141:8081/nexus/content/groups/public/

大部分的终端用户，只需要配置上面这一个单独的聚合后的url，而不用单独配置多个仓库。用户也不需要知道某个jar包具体来源于maven 中央仓库，或者是Apache Snapshots，或者是我们自己添加的其他仓库。

这样的好处在于：如果我们要新增某个仓库（如开源中国、阿里云），客户端不需要做任何更改，只需要在nexus上将要新增的仓库加入到对外提供服务的仓库 group中就可以。

第二个3rd party，与倒数第一个和第二个仓库，Releases和Snapshots一样，类型为hosted，意思是由nexus本机管理的仓库。该仓库用于商业化的，第三方提供的非开源的依赖仓库，如oracle jdbc driver。

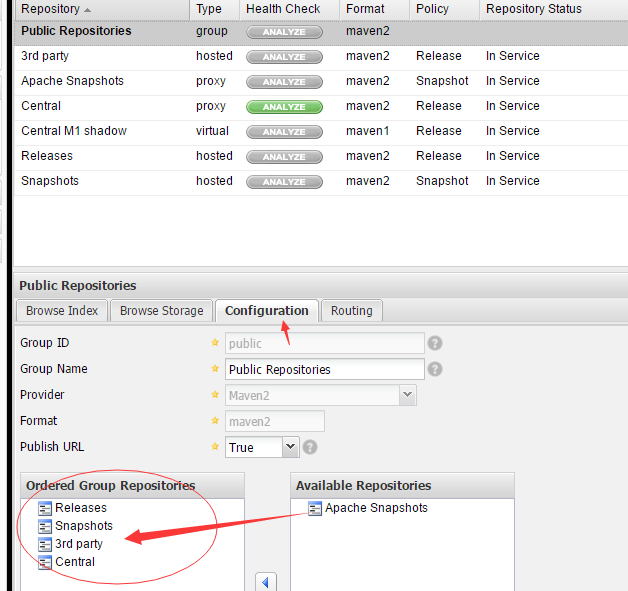
倒数第二个Releases，用于存放开发团队内部用的正式版的依赖。

倒数第一个Snapshots，用于存放开发团队内部日常构建的频率更新较快的依赖包。

Apache Snapshots和Central类型都是proxy，意思是远端仓库的代理。前者包含了Apache Software Foundation 发布的快照版本（这么翻译不知道对不对），后者为Maven中央仓库，我们平时maven默认就是连接该仓库。

Central M1 Shadow类型为virtual，按官方文档的意思是，只是已有仓库的一个不同呈现方式的映射。有需要可以参考官方手册6.2.3节。

3.2 仓库集合的界面

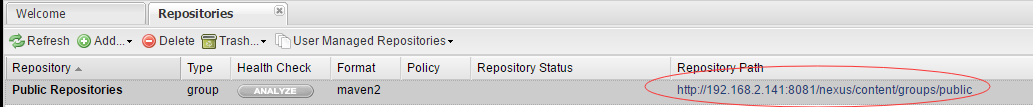


点击Configuration，可以看到当前添加到该集合的仓库列表及顺序（优先级高的在上面，可用鼠标拖拽），当我们新增了仓库，将会出现在右侧的available Repository，添加到左边即可。

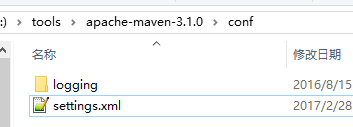
如果有哪个仓库不想加入到该集合，也可以拖拽到右边来。

3.3使用本地maven客户端测试nexus是否成功部署

首先复制仓库集合的repository url，客户端配置需要用到。



配置maven客户端：



打开settings.xml，按照如下步骤修改：（该部分参考文档4.2节）

[复制代码](javascript:void(0);)

<mirrors>

<mirror>

<!--This sends everything else to /public -->

<id>nexus</id>

<mirrorOf>\*</mirrorOf>

<url>http://192.168.2.140:8081/nexus/content/groups/public</url>

</mirror>

</mirrors>

[复制代码](javascript:void(0);)

<profiles>

<profile>

<id>nexus</id>

<!--Enable snapshots for the built in central repo to direct -->

<!--all requests to nexus via the mirror -->

<repositories>

<repository>

<id>central</id>

<url>http://central</url>

<releases><enabled>true</enabled></releases>

<snapshots><enabled>true</enabled></snapshots>

</repository>

</repositories>

<pluginRepositories>

<pluginRepository>

<id>central</id>

<url>http://central</url>

<releases><enabled>true</enabled></releases>

<snapshots><enabled>true</enabled></snapshots>

</pluginRepository>

</pluginRepositories>

</profile>

</profiles>

<activeProfiles>

<activeProfile>nexus</activeProfile>

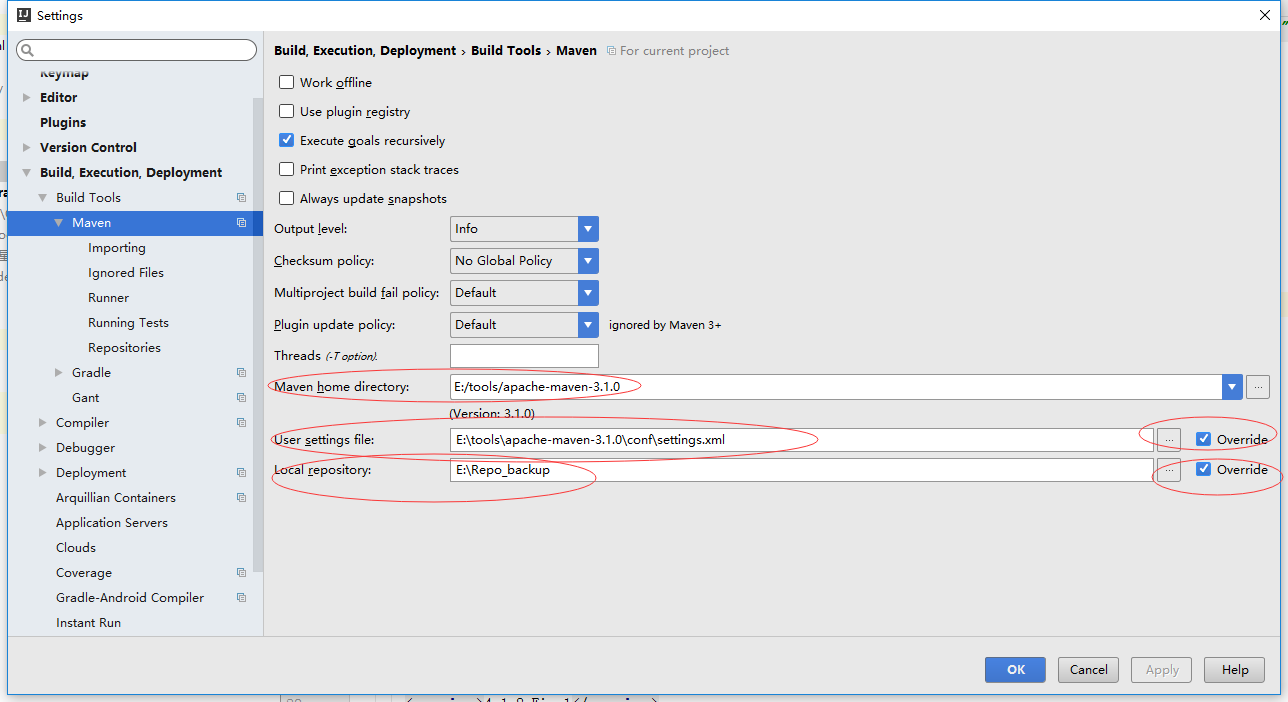
</activeProfiles>

[复制代码](javascript:void(0);)

[复制代码](javascript:void(0);)

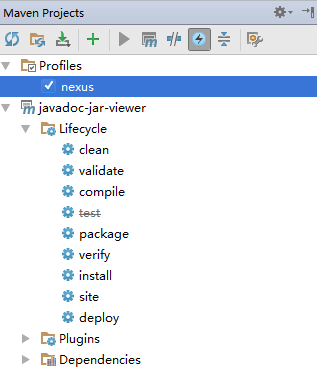
接下来在ide中测试下（我个人用maven命令行不多，不熟悉，还是用intelj idea演示吧）

在idea中依次打开File--Settings--搜索Maven--出现如下界面：

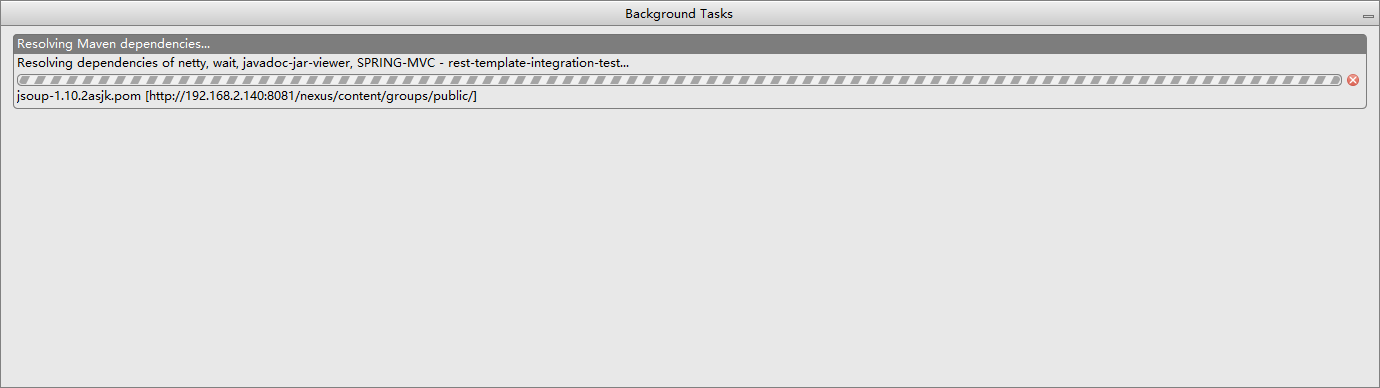


保证上述几项指向正在用的maven目录和配置文件即可。点击ok，退出。

 打开Maven Projects窗口，勾选Profiles中的nexus。



随便新建一个maven工程，在pom中添加一项本地仓库中没有的依赖，应该出现如下图示：（下图可以看出，正在去我们配置的私服下载maven依赖）



至此。大体配置完成。

上传本地内部依赖到私服，可以参考：

http://jingyan.baidu.com/article/a948d6517b72eb0a2dcd2e2a.html

ps：有问题欢迎大家一起讨论，也可以直接参考文章开头发的pdf文档，非常详细。

使用方式

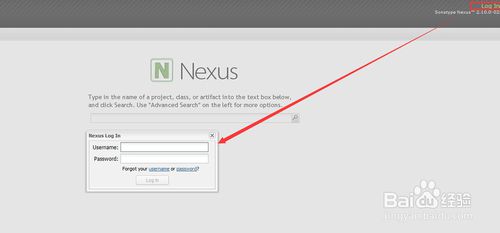
## -------------------------------------------------------------------------------工具/原料

* nexus私有库

## 方法/步骤

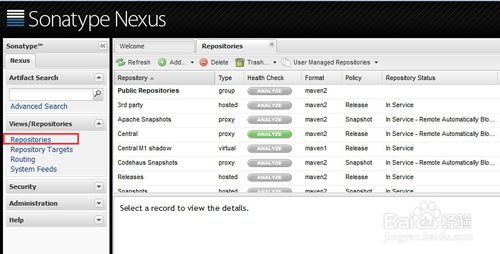
1. 1

登陆已经安装好的nexus私有仓库，如图：

[](http://jingyan.baidu.com/album/a948d6517b72eb0a2dcd2e2a.html?picindex=1)

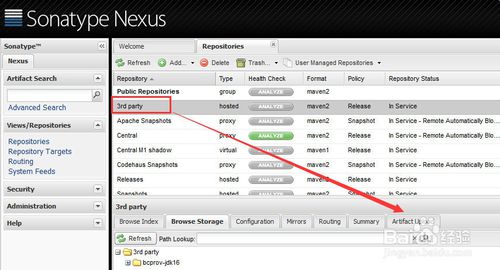
1. 2

点击左边菜单“Repositories”，选择右边列表“3rd party“

[](http://jingyan.baidu.com/album/a948d6517b72eb0a2dcd2e2a.html?picindex=2)

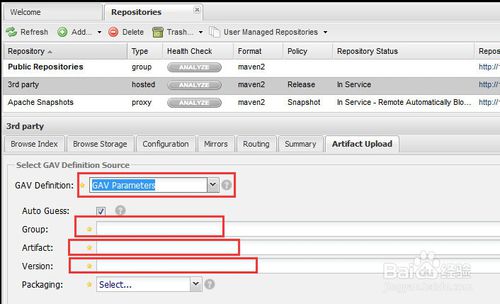
1. 3

点击“3rd party”,选择artifact Upload,如下图：

[](http://jingyan.baidu.com/album/a948d6517b72eb0a2dcd2e2a.html?picindex=3)

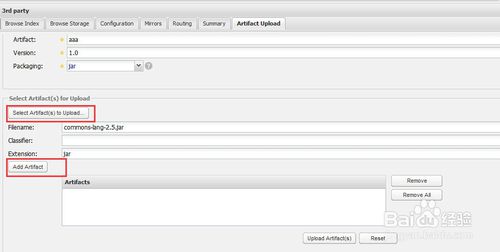
1. 4

在artifact Upload界面显示的有GAV Parameter,groupId,artifactId,Version,Packaging等等参数，界面如图：

[](http://jingyan.baidu.com/album/a948d6517b72eb0a2dcd2e2a.html?picindex=4)

1. 5

将以上参数填完之后，选择要上传的jar包，点击”add artifact“,最后点击”save“保存按钮即可，如下图所示：

[](http://jingyan.baidu.com/album/a948d6517b72eb0a2dcd2e2a.html?picindex=5)

END

经验内容仅供参考，如果您需解决具体问题(尤其法律、医学等领域)，建议您详细咨询相关领域专业人士