Nexus安装与使用指南

# Nexus下载、安装与启动

Nexus提供两种安装模式，一种是WAR包部署，一种是Bundle安装，这里主要采用Bundle安装模式。

你可以直接使用附件中的nexus-2.6.2-01-bundle.zip（这是编写改文档时，官方网站的最新版本），或者到官方网站下载最新版本。地址：<http://www.sonatype.org/nexus/>

获得zip安装文件后，将其解压，会发现它包含后两个目录。如下图1.1所示：



图1.1

上图1.1中，目录sonatype-work为nexus的相关工作目录，里面包含了nexus运行时的各种参数配置（用户、权限、repository proxy配置信息等）和相关repository目录空间。该目录底下此时只有一个nexus目录和README.txt文件，后续nexus启动成功后，会生成刚才提到的相应文件及子目录。

上图1.1中，目录nexus-2.6.2-01为nexus的home目录，后续统称为%NEXUS-HOME%，内嵌jetty-server容器，包含了相关lib的Dependencies，以及日志文件。如下图图1.2所示

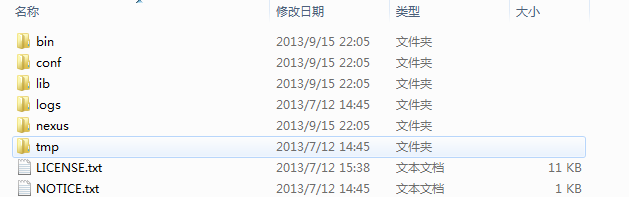


图1.2

进入bin子目录，nexus.bat包含相关的启动参数命令（start、stop、install as service等）录如下图1.3所示：



图1.3

进入命令行窗口，执行cd {NEXUS-HOME}/bin跳转到nexus的bin目录，执行nexus命令。Windows环境执行nexus.bat，linux环境输入./nexus执行。命令行显示如下图1.4：



图1.4

通过提示可以知道nexus命令需要带有相关的参数才有效果。

启动请输入：nexus start

关闭请输入：nexus stop

重启请输入：nexus restart

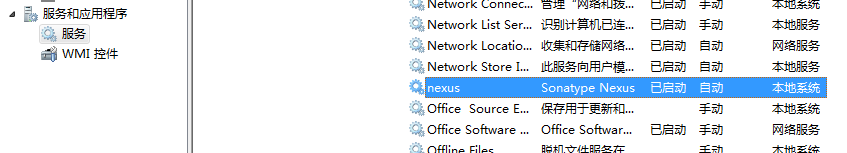
安装为自启动服务请输入：nexus install

卸载自启动服务请输入：nexus uninstall

作者本次安装环境是win7x64操作系统，通过 nexus install为系统自启动服务了。

如下图1.5所示：

图1.5



nexus服务启动成功后，在浏览器输入：<http://192.168.1.100:8081/nexus>。进入Sonatype Nexus的Welcome页面。这里IP地址192.168.1.100是作者的IP 地址，请根据实际情况输入安装机器的IP，如果你是本机安装也可以用192.168.1.100代替IP地址。如下图1.6所示：

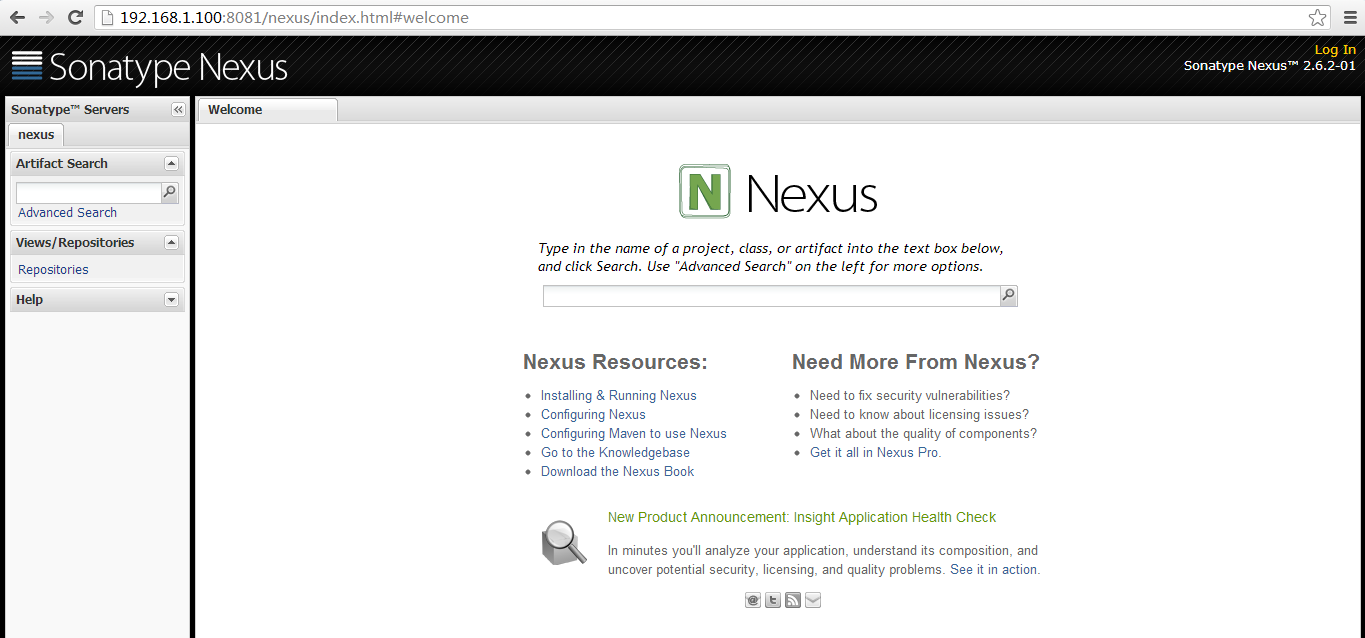


图1.6

端口号8081是Nexus的内嵌Jetty-Server的默认端口。若有其他需要可以自行更改%NEXUS-HOME%/conf/nexus.properties文件里面application-port属性。重启后生效。

# Nexus配置

在Nexus主页面的右上角，点击Log In，通过Nexus的默认管理员用户：admin，默认密码：admin123，进入Nexus的管理界面。如下图2.1所示：

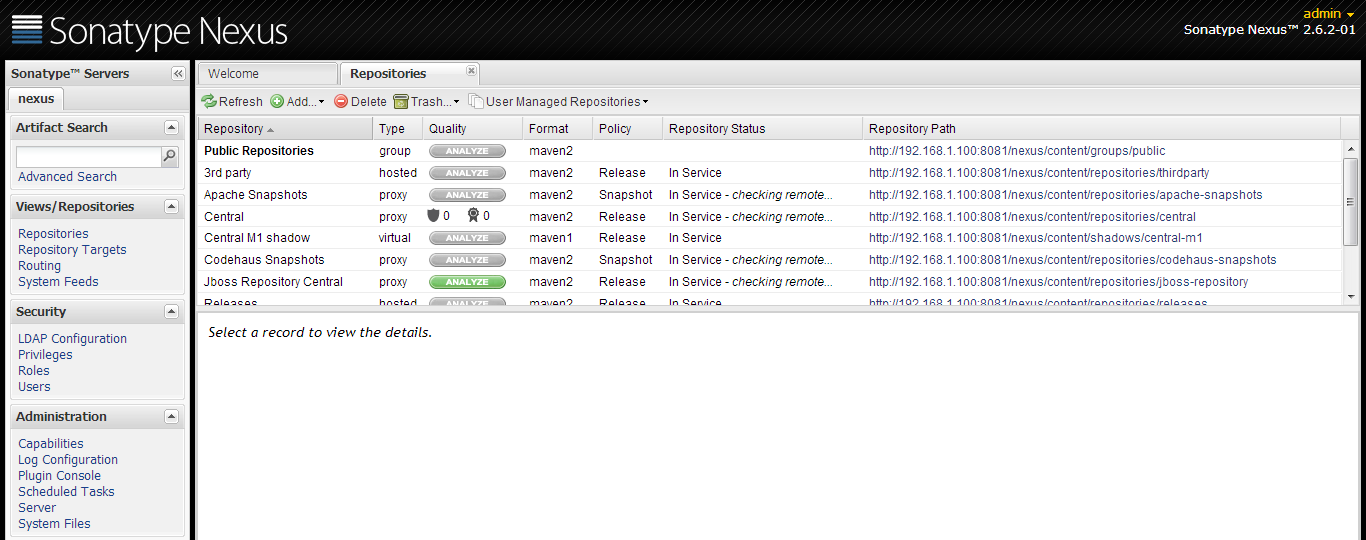


图2.1

## 配置Internet代理（可选）

如当前安装机器，不能直接连接Internet，可以通过菜单栏的Server进入nexus的Server配置，勾选Default HTTP Proxy Setting(optional)。并进行HTTP代理服务器配置如下图2.1.1所示：

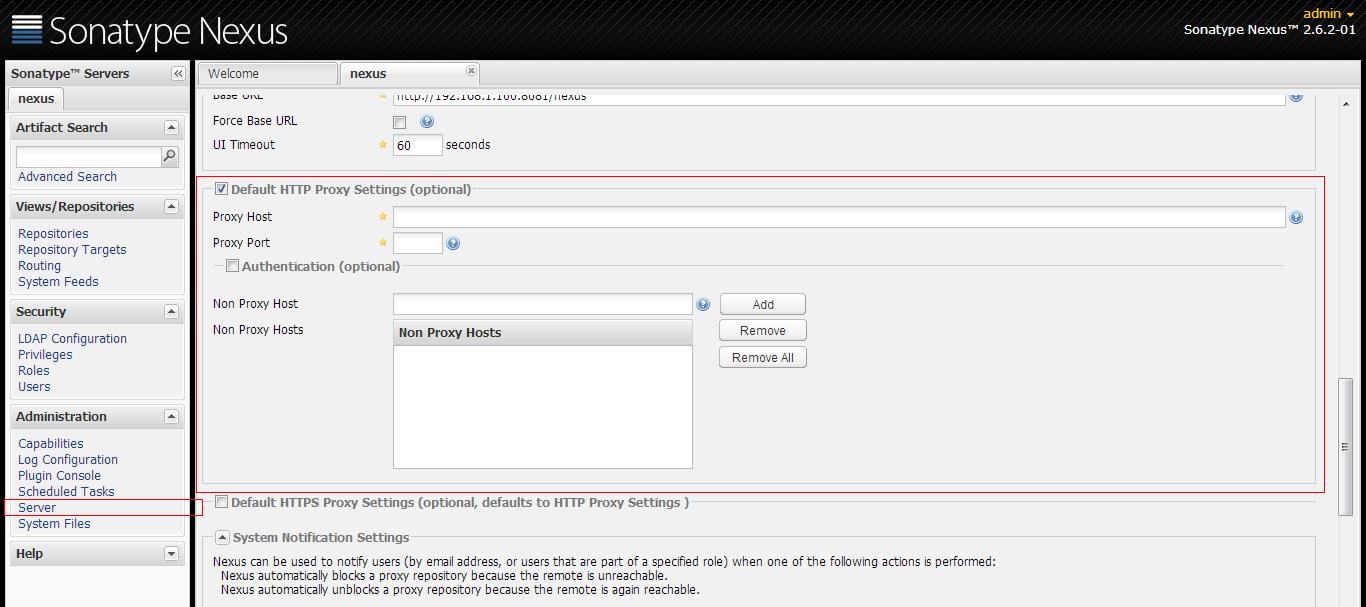


图2.1.1

Proxy Host输入代理服务器地址，Proxy Port输入代理服务器端口，如果代理服务器需要账号密码，勾选Authentication(optional)，输入账号密码，根据需要添加不需要代理的网络地址。

## 配置Repository

Nexus支持3种仓库类型：

* **Proxy：**代理Internet上开放的Repository。maven用户通过nexus访问下载一个internet上的依赖包时，先检查本地是否存在，如果存在直接从nexus本地下载，如果不存在先从远程下载，并缓存在nexus服务器本地，也就是说所有依赖包，只需要向远程Maven库下载一次，之后就直接从Nexus服务器本地下载，这样大大节省带宽，并提高依赖包的下载速度。例如代理Central Repository：<http://repo1.maven.org/maven2/>等。
* **Hosted：**是宿主仓库，用户可以把自己的一些构件，deploy到hosted中，也可以手工上传构件到hosted里。比如说oracle的驱动程序，ojdbc6.jar，在Central Repository是获取不到的，就需要手工上传到hosted里。在比如说：公司内部发布的依赖包，也是发布到hosted的Repository里面的，公司内部发布的依赖包，在多个team协同开发的情况下，特别常见。
* **Group：**Nexus特别提供的仓库组。目的是将多个仓库聚合，对用户暴露一个统一的地址，这样用户就不需要在pom中配置多个地址，只要配置统一group的地址就可以了

### 配置Proxy仓库

Nexus安装成功，自动带了一些Proxy仓库配置，

比如：

<http://repo1.maven.org/maven2/>；

<http://nexus.codehaus.org/snapshots/>；

等。

根据开发需要可以添加下面几个internet的开发性Repository。

例如：

<http://repo.springsource.org/release/>

<http://repository.sonatype.org/content/groups/public/>

<https://repository.jboss.org/nexus/content/repositories/>

<http://google-maven-repository.googlecode.com/svn/repository/>

<http://download.java.net/maven/glassfish/>

<http://download.java.net/maven/2/>

<http://download.java.net/maven/1/>

<http://people.apache.org/repo/m2-incubating-repository/>

<http://mirrors.ibiblio.org/maven2/>

请注意在配置Proxy仓库时候，勾选Download Remote Indexes，这样将会自动下载远程的服务器的依赖包索引。其他的默认即可，远程库地址请在Remote Storage Location栏输入。详细如下图2.2.1.1所示：

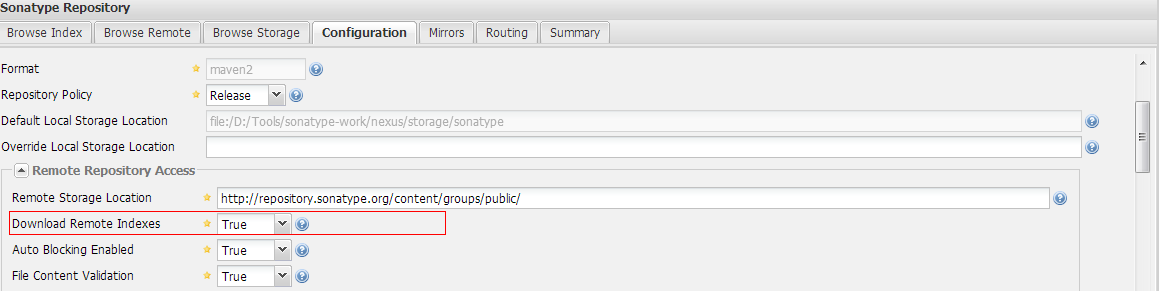


图2.2.1.1

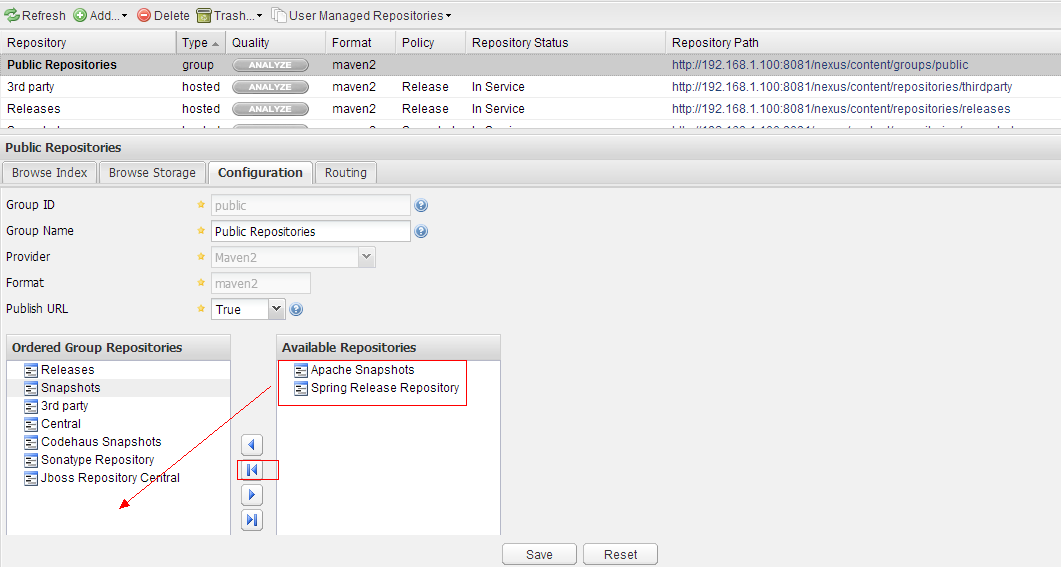
### 配置Hosted库

在Nexus第一次安装后，自动配置了3个Hosted库，分别是3rd party、Releases、Snapshots。其中3rd party主要用来存放和管理第三方的并且在internet开放库里面不存在的依赖包。Releases主要用来存储公司内部正式发布的依赖包，Snapshots主要来存储公司内部发布快照依赖包。这些主要是用来公司内部多个team之间协同开发用的。

当然如果是需要将多个team之间的发布的正式包和快照包，分别存储的话。就可以根据需要新建与team对应的Hosted库。

### 配置Group库

在Nexus第一次安装后，自动配置一个Group库，叫Public Repository。由于我们在前面配置了一些Proxy库和Hosted库。这里需要将Group库，进行一些修改。将我们前面新加的库，加入到该Group库中来。进入配置页面，将右边Available Repositories中的有效库，添加到左边Ordered Group Repositories来，点击Save保存即可。如下图2.2.3.1所示：

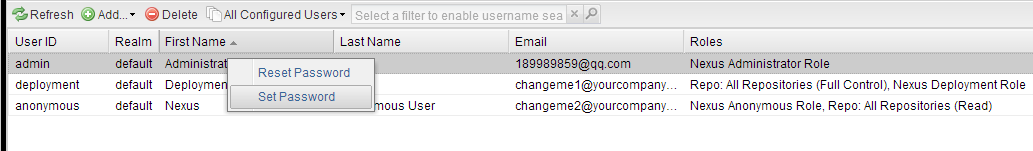


下图2.2.3.1

## 用户配置

Nexus安装后，默认包含3个用户，分别是admin、deployment、anonymous。admin拥有所有的管理权限，deployment拥有发布相关Hosted库的权限，anonymous只查看和下载依赖包的权限。admin用户的默认密码是：admin123，deployment的默认密码是：deployment123。

如果需要更改用户密码，选择用户记录点击右键，弹出菜单选择Set Password。



# Maven配置

Nexus的一切都配置好，开发人员要想使用Nexus库，可以通过两种方式进行配置。①配置工程的pom.xml文件，添加repository地址。②配置setting.xml，添加mirror。为了方便起见，防止每个工程都要去配置pom.xml。这里采用方式②。

settings.xml分别会在两个地方存在，一个是Maven的安装目录，一个是在用户目录，例如：%M2\_HOME%/conf/settings.xml和%USER\_HOME%/.m2/settings.xml。

在settings.xml中添加如下信息：

<mirrors>

<mirror>

<id>nexus</id>

<mirrorOf>\*</mirrorOf>

<name>central nexus repository</name>

<url>http://192.168.1.100:8081/nexus/content/groups/public/</url>

</mirror>

</mirrors>

和

<servers>

<server>

<id>nexus-snapshots</id>

<username>deployment</username>

<password>deployment</password>

</server>

</servers>

mirrors中的url为nexus的库地址，由于作者是在本机安装的nexus所以这里填的是192.168.1.100。开发人员请根据自己的实际情况，填写nexus的服务器ip地址。

至于<servers>元素，是因为我们把项目内部的构件上传到nexus的仓库中时，nexus会进行权限控制，所以这里需要设置权限相关的信息。server中username和 password根据实际情况配置。注意这里的<id>nexus-snapshots<id>，和开发时的maven工程里的pom设置是一致的 。

Maven工程需要发布包，还需要进行distrubutionManagement配置。如下所示：

<distributionManagement>

<snapshotRepository>

<id>nexus-snapshots</id>

<name>nexus distribution snapshot repository</name>

<url>http://192.168.1.100:8081/nexus/content/repositories/snapshots/</url>

</snapshotRepository>

</distributionManagement>

请根据实际情况配置url中的IP地址。

这里配置了<distributionManagement>元素，其中的<id>nexus-snapshots</id>，与前面说的settings.xml中的<servers>元素中的配置必须一致

配置这个的目的，是当执行maven deploy时，才知道要将生成的构件部署到哪个远程仓库上，注意这里的URL填的就不是public group的地址：

<http://192.168.1.100:8081/nexus-2.1.1/content/groups/public/>

而是snapshots的地址：

<http://192.168.1.100:8081/nexus-2.1.1/content/repositories/snapshots/>

由于在nexus中，snapshots也是聚合到public group里的，所以开发人员A提交到snapshots的构件，开发人员B也可以从public group里获取到 。

# Eclipse配置

以上都配置好，需要在Eclipse里面进行一个小小的操作。通过Windows🡪Preferences🡪Maven🡪User Settings进入Maven的用户设置，点击里面的Update Settings按钮，更新加载刚才的Maven配置。如下图4.1所示：

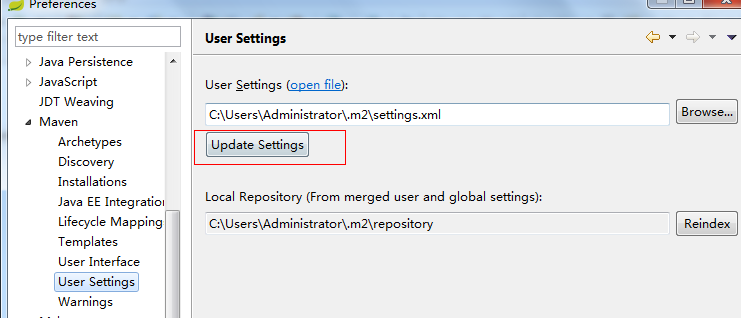


图4.1

打开Maven Repositories视图可以发现nexus镜像库以及代理远程库成功。如下图4.1所示：

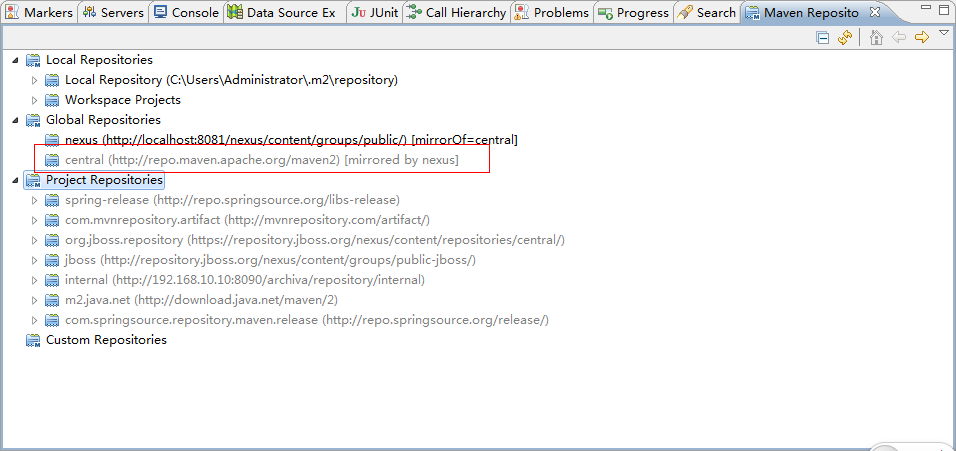


图4.1