



链滴

# 简明 SONATYPE NEXUS 私有库搭建配置说明

作者: [flhuoshan](#)

原文链接: <https://ld246.com/article/1496463966423>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 缘由

- 持续集成jenkins框架需要配合
- 私库上传自有包
- 避免中央仓库网络条件不行包下载失败，且外网中断时也可以正常运行
- 节约公司网络带宽
- 加速包下载

## 步骤

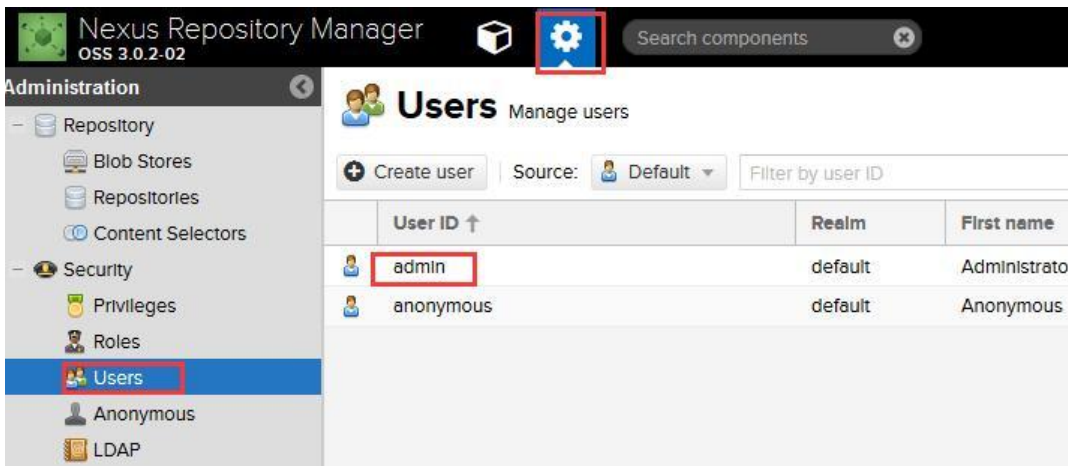
- 先安装jdk8，配置环境变量
- 下载程序包 nexus-3.3.1-01-unix.tar.gz 【nexus oss版本】
- 上传到/usr/local目录，解压,进入到bin目录执行启动脚本，这里的&意思是作为后台进程运行

```
tar -cxvf nexus-3.3.1-01-unix.tar.gz
mv nexus-3.3.1-01-unix.tar.gz nexus331
cd nexus331/bin
./nexus run &
```

- 执行后，会打印很多启动信息。启动完成后访问：服务器地址:8081,nexus的默认端口是8081，有个默认用户admin（密码admin123）、anonymous（匿名用户，只能查看）。
- 关闭nexus

```
./nexus stop[关闭nexus]
```

## 登录，大概是这样：



The screenshot shows the Nexus Repository Manager Administration interface. The top navigation bar includes the Nexus logo, version 'OSS 3.0.2-02', a search bar for components, and a gear icon for settings. The left sidebar shows the 'Administration' menu with options like Repository, Security, and Users. The 'Users' option is highlighted. The main content area is titled 'Users Manage users' and contains a table of users. The 'admin' user is highlighted with a red box.

User ID ↑	Realm	First name
admin	default	Administrator
anonymous	default	Anonymous

## 本地设置

- 安装好以后，需要与本地做关联。那么，这个关联要怎样体现呢？一是本地maven的settings.xml文件，二是工程的pom.xml文件。
- 一、maven配置文件：settings.xml,该文件作用在于定位主库地址，如果仅仅需要从私库download资源，将使用的镜像替换为主库地址即可。

```

<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <!-- 定义本地仓库代码 -->
  <localRepository>D:\userfulsoftware\mnrepository</localRepository>

  <!--使用用户名、密码进行权限认证 -->
  <servers>
  <server>
    <id>nexus-snapshots</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
  <server>
    <id>nexus-releases</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
  </servers>
  <!--私有仓库主库地址 -->
  <mirrors>
  <mirror>
    <id>nexus</id>
    <mirrorOf>*</mirrorOf>
    <url>http://xx.xx.xx.xx:8081/repository/maven-public/</url>
  </mirror>
  </mirrors>
</settings>

```

- 二、工程的pom.xml,在文件内加入，作用是有jar包源码，或者是自己开发的私有jar包，可以在pom.xml中添加如下配置，加完之后可以将本地的jar包部署到maven库中，当然，当有多个repository时可以在mvn命令参数中设定，在后面会提到。

```

<distributionManagement>
  <repository>
    <id>nexus-releases</id>
    <name>Releases</name>
    <url>http://xx.xx.xx.xx:8081/repository/maven-releases</url>
  </repository>
  <snapshotRepository>
    <id>nexus-snapshots</id>
    <name>Snapshot</name>
    <url>http://xx.xx.xx.xx:8081/repository/maven-snapshots</url>
  </snapshotRepository>
</distributionManagement>

```

## 上传jar包到NEXUS私服

- 以idea为例，在下方terminal选项卡中，跳转到工程后，有两种情况：  
上传本工程到NEXUS私服，键入命令

```
mvn clean deploy -Dmaven.test.skip=true
```

- 将本工程打包，分发至本地仓库及NEXUS私服。
- 上传已有jar包，如oracle的ojdbc6.jar，此包在中央仓库是没有的，因此需要作为私有包上传到NEXUS私服，键入命令：

```
mvn deploy:deploy-file -DgroupId=com.oracle -DartifactId=ojdbc6 -Dversion=11.2.0.2.0 -Dpackaging=jar -Dfile=lib\ojdbc6.jar -Durl=http://xx.xx.xx.xx:8081/repository/maven-releases -DrepositoryId=nexus-releases
```

- 其中-Dfile=lib\ojdbc6.jar这个是本地的文件，lib\ojdbc6.jar是文件路径，上传私库后本地的文件可以删除了。
- -Durl=http://xx.xx.xx.xx:8081/repository/maven-releases这个路径，是私库中用来存放稳定版本jar包的路径。
- -DrepositoryId=nexus-releases这个参数对应pom.xml里的repository

## 结语

- 在进行开发工作的时候，因为墙的存在，很多人会将maven中央仓库换为国内的镜像仓库，以避免依赖下载不了，网络时断时续。然而，更方便的做法是在内网架构一台私有仓库服务器，既去除了断的风险，又可以轻松的上传自己的私有包，同时可以和其他的一些工具结合使用【比如CI工具】，大提升工作效率，减少重复性劳动。