

# Nexus 3 搭建 Maven 私服

Name : 曲中岭

Email:[zlingqu@126.com](mailto:zlingqu@126.com)

Q Q :441869115

## 一 Nexus 入门

### 1.1 简介

Nexus 是一个强大的 Maven 仓库管理器，它极大地简化了自己内部仓库的维护和外部仓库的访问。利用 Nexus 你可以只在一个地方就能够完全控制访问和部署在你所维护仓库中的每个 Artifact。

Nexus 是一套“开箱即用”的系统不需要数据库，它使用文件系统加 Lucene 来组织数据。

### 1.2 为何要使用 nexus

传统方法，研发人所有构件都需要通过 maven 的中央仓库和第三方的 Maven 仓库下载到本地，而一个团队中的所有人都重复的从 maven 仓库下载构件，而这些仓库往往都位于境外站点，这样的后果是：

- 1) 重复下载，加大了仓库的负载、浪费了外网带宽，如果网速慢或者不稳定的话，还会影响项目的进程
- 2) 如果项目是在内网进行的，无法连接到外网
- 3) 开发的公共构建共享比较麻烦

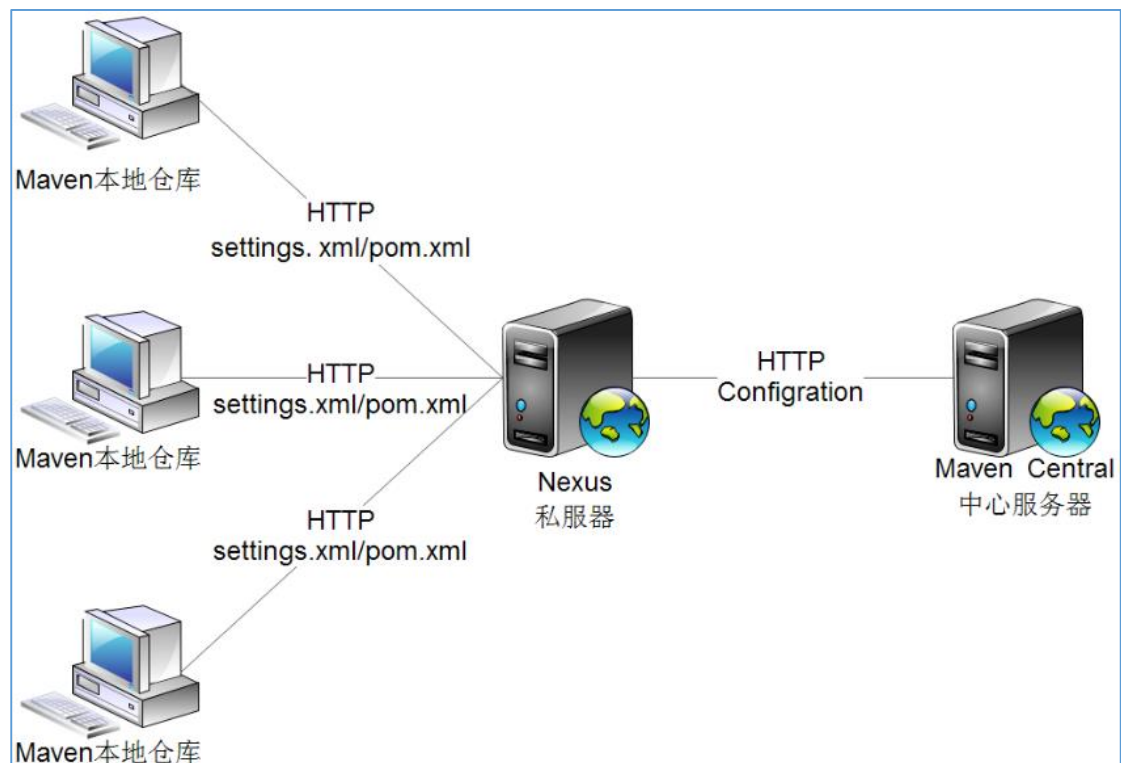
这个时候如果有一个内网的私有仓库就方便多了。nexus 就是做这个事情的，使用 nexus 可以搭建一个私有仓库，能自动从远程仓库下载构件，同时也给给开发人员提供了一个公共资源池。

## 1.3 nexus 工作原理

私服是一种特殊的远程仓库，它是架设在局域网内的仓库服务，私服代理广域网上的远程仓库，供局域网内的 Maven 用户使用。

当 Maven 需要下载构件的时候，它从私服请求，如果私服上不存在该构件，则从外部的远程仓库下载，缓存在私服上之后，再为 Maven 的下载请求提供服务。所以，只要有一个人请求了某个构件，私服上就有了副本，其他人就可以方便的使用。

此外我们还可以把一些无法从外部仓库下载到的构件上传到私服上，供其他人员使用。



## 二 部署环境准备

### 2.1 环境要求

#### 2.1.1 JDK

JDK: oracle 的 jdk8,

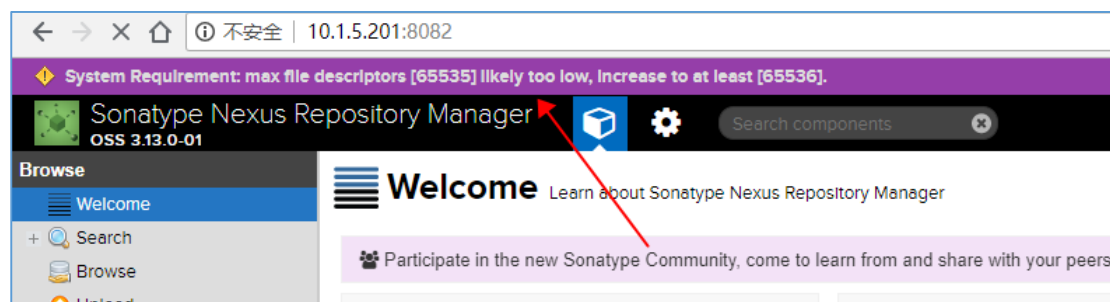
官方明确说明不支持 OpenJDK 或其他 Java 发行版或较旧的 Java 版本。

见: <https://help.sonatype.com/repomanager3/installation/java-runtime-environment>

#### 2.1.2 文件句柄

```
cat "nexus - nofile 65536" >> /etc/security/limits.conf
```

若不修改, web 登陆侯会有以下警告:



另外, 修改

```
[root@hadoop01 ~]# grep limit /etc/profile
```

```
ulimit -HSn 65536
```

里面的配置, 这是会话级别的限制, 若不修改, 上面的修改不生效。

#### 2.1.3 其他要求

参考:

<https://help.sonatype.com/repomanager3/system-requirements>

### 2.2 版本发布查询

2.x 版本:

<https://help.sonatype.com/repomanager2/release-notes/older-release-notes>

3.x 版本:

<https://help.sonatype.com/repomanager3/release-notes>

本文档演示 3.13.0-01 版本，日期 2018-08-10

## 2.3 配置 jdk

下载地址：<http://www.oracle.com/technetwork/cn/java/archive-139210-zhs.html>

找到自己需要的版本，上传到服务器

```
tar xf jdk-8u171-linux-x64.tar -C /usr/local
```

```
mv /usr/local/jdk1.8.0_171 /usr/local/jdk #也可以创建软连接，看个人习惯
```

保证/etc/profile 中有以下内容：

```
export JAVA_HOME=/usr/local/jdk
```

```
export PATH=$JAVA_HOME/bin:$PATH # $JAVA_HOME/bin 一定要放到$PATH 前面，
```

防止识别到系统环境中其他版本的 jdk

```
source /etc/profile 使配置生效
```

```
java -version 查看是否配置成功
```

## 2.4 添加用户

官方强烈不建议使用 root 运行，这里添加程序用户 nexus，也是为了安全

```
useradd nexus
```

```
mkdir /data
```

```
chown -R nexus.nexus /data
```

## 2.5 规划

OS：CentOS 6.9 x64

JDK：1.8.0\_171

安装目录：/data/nexus-3.13.0-01

端口：8081 （默认）

## 三 安装

### 3.1 下载和解压

```
su - nexus
cd /data/
wget https://download.sonatype.com/nexus/3/latest-unix.tar.gz
可到如下连接查看其他信息：
https://help.sonatype.com/repomanager3/download
```

```
tar xf latest-unix.tar.gz
解压后会出现两个目录：
nexus-3.13.0-01    #程序目录
sonatype-work     #数据目录，默认的，可修改
```

修改目录权限

```
chown -R nexus:nexus /data/nexus-3.13.0-01 /data/sonatype-work
```

### 3.2 配置文件

修改程序运行用户：

```
cat /data/nexus-3.13.0-01/bin/nexus.rc
run_as_user="nexus"
```

### 3.3 启停

使用：/data/nexus-3.13.0-01/bin/nexus start  
启动，启动需要大概十多秒，耐心等待，观察 8081 端口是否监听，来确定是否启动，当然也可以通过观察日志。  
也可接受其他参数  
/data/nexus-3.13.0-01/bin/nexus {start|stop|run|run-redirect|status|restart|force-reload}  
启动后监听 8081 端口，运行用户是 nexus

```

[root@hadoop01 ~]# netstat -tnlp|grep java
tcp        0      0 0.0.0.0:8081          0.0.0.0:*           LISTEN     22395/java
tcp        0      0 0.0.0.0:40328       0.0.0.0:*           LISTEN     22395/java
[root@hadoop01 ~]#
[root@hadoop01 ~]#
[root@hadoop01 ~]# ps -ef|grep nexus
nexus      22395      1 33 09:52 pts/0    00:01:16 /usr/local/jdk/bin/java -server -Dinstall4j.jvmDir=/
X:+UnlockDiagnosticVMOptions -Dinstall4j.launcherId=245 -Dinstall4j.swt=false -Di4jv=0 -Di4jv=0 -Di4
2G -XX:+UnlockDiagnosticVMOptions -XX:+UnsyncloadClass -XX:+LogVMOutput -XX:LogFile=./sonatype-work
rIPv4Stack=true -Dkaraf.home=. -Dkaraf.base=. -Dkaraf.etc=etc/karaf -Djava.util.logging.config.file=
k/nexus3 -Djava.io.tmpdir=./sonatype-work/nexus3/tmp -Dkaraf.startLocalConsole=false -Di4j.vpt=true
exus-3.13.0-01/lib/boot/nexus-main.jar:/data/nexus-3.13.0-01/lib/boot/org.apache.karaf.main-4.0.9.jar
s-3.13.0-01/lib/boot/org.apache.karaf.diagnostic.boot-4.0.9.jar:/data/nexus-3.13.0-01/lib/boot/org.a
Launcher start 9d17dc87 org.sonatype.nexus.karaf.NexusMain
root       22810  21802  0 09:56 pts/0    00:00:00 grep nexus
[root@hadoop01 ~]#

```

## 3.4 添加为系统服务

```

[root@hadoop01 bin]# ln -sv /data/nexus-3.13.0-01/bin/nexus /etc/init.d/nexusr
"/etc/init.d/nexusr" -> "/data/nexus-3.13.0-01/bin/nexus"

```

我这里将系统服务名改成了 nexusr，多加了一个 d。

```

[root@hadoop01 bin]#
[root@hadoop01 bin]# chkconfig --add nexusr
[root@hadoop01 bin]#
[root@hadoop01 bin]# chkconfig --list nexusr
nexusr      0:关闭 1:关闭 2:启用 3:启用 4:启用 5:启用 6:关闭

```

```

[root@hadoop01 bin]# ln -sv /data/nexus-3.13.0-01/bin/nexus /etc/init.d/nexusr
"/etc/init.d/nexusr" -> "/data/nexus-3.13.0-01/bin/nexus"
[root@hadoop01 bin]#
[root@hadoop01 bin]# chkconfig --add nexusr
[root@hadoop01 bin]#
[root@hadoop01 bin]# chkconfig --list nexusr
nexusr      0:关闭 1:关闭 2:启用 3:启用 4:启用 5:启用 6:关闭
[root@hadoop01 bin]#

```

此后可以使用 service nexusr start|stop|restart 等操作

同样，在 Ubuntu 平台，使用 rc.d 进行服务的添加等操作，可实现同样的效果。

## 3.5 其他配置修改

以下内容，自己决定是否是否需要修改，使用默认的配置也可以运行。

### 3.5.1 修改监听的 IP 和端口

修改

/data/nexus-3.13.0-01/etc/nexus-default.properties

中

application-port=8082

application-host=0.0.0.0

这两行即可

### 3.5.2 修改默认路径

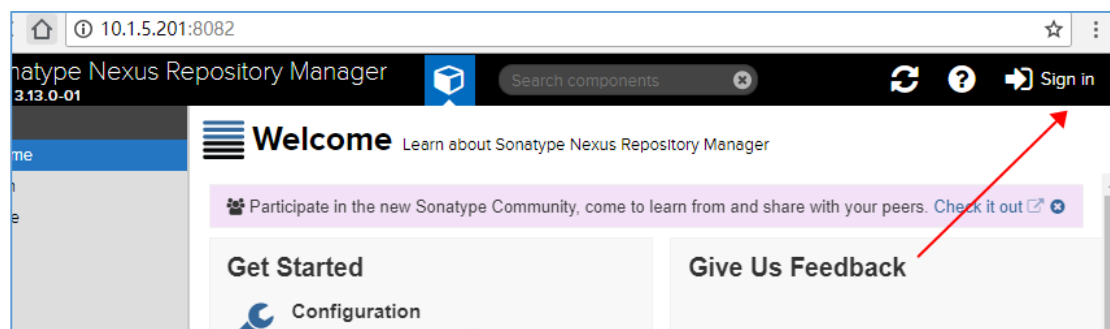
如下所示，可修改启动的内存大小，log/tmp/data 等路径

```
[root@hadoop01 bin]# cat /data/nexus-3.13.0-01/bin/nexus.vmoptions
-Xms1200M
-Xmx1200M
-XX:MaxDirectMemorySize=2G
-XX:+UnlockDiagnosticVMOptions
-XX:+UnsyncloadClass
-XX:+LogVMOutput
-XX:LogFile=../sonatype-work/nexus3/log/jvm.log
-XX:-OmitStackTraceInFastThrow
-Djava.net.preferIPv4Stack=true
-Dkaraf.home=.
-Dkaraf.base=.
-Dkaraf.etc=etc/karaf
-Djava.util.logging.config.file=etc/karaf/java.util.logging.properties
-Dkaraf.data=../sonatype-work/nexus3
-Djava.io.tmpdir=../sonatype-work/nexus3/tmp
-Dkaraf.startLocalConsole=false
```

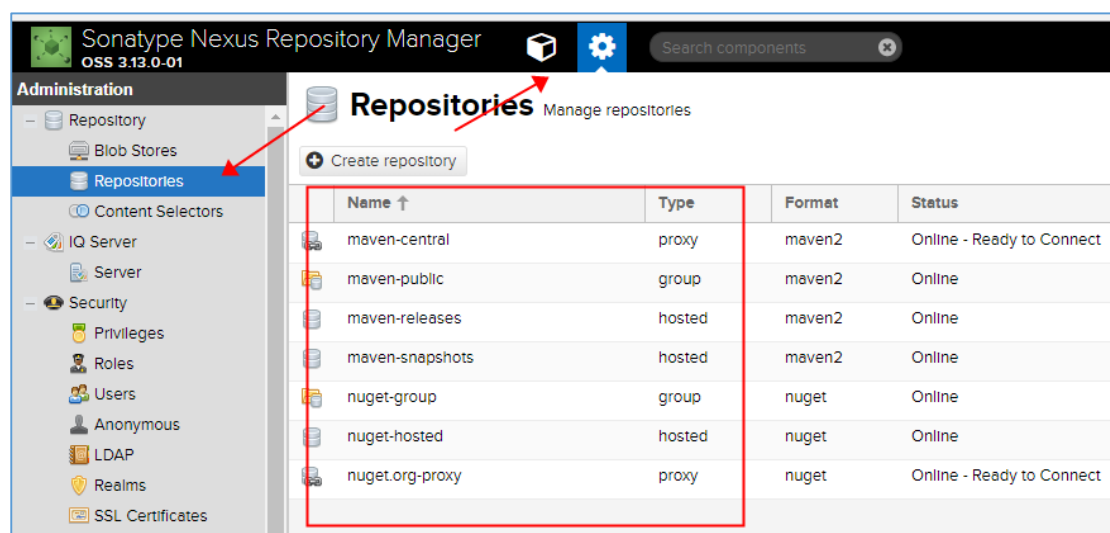
## 四 使用

### 4.1 登陆

使用 <http://10.1.5.201:8082/> (已修改端口) 登陆 web, 点击 sign in 登陆, 默认账号密码是 admin/admin123



登陆之后, 可以看下有如下几个默认仓库:



### 4.2 仓库分类

Nexus 默认的仓库类型有以下四种:

- 1) group(仓库组类型): 又叫仓库组, 用于方便开发人员自己设定的仓库;
  - 2) hosted(宿主类型): 内部项目的发布仓库 (内部开发人员, 发布上去存放的仓库);
  - 3) proxy(代理类型): 从远程中央仓库中寻找数据的仓库 (可以点击对应的仓库的 Configuration 页签下 Remote Storage Location 属性的值即被代理的远程仓库的路径);
  - 4) virtual(虚拟类型): 虚拟仓库 (这个基本用不到, 重点关注上面三个仓库的使用);
- aven 可直接从宿主仓库、仓库组或者代理仓库下载构件, 而代理仓库间接的从远程仓库



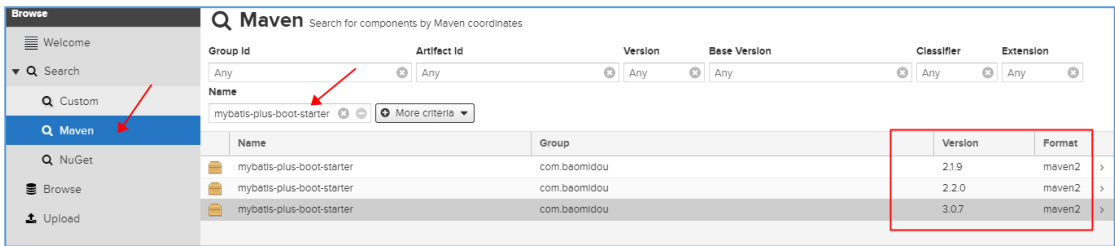
下载并缓存构件

为了方便,Maven 可以从仓库组下载构件,而仓库组并没有时间的内容(下图中用虚线表示,它会转向包含的宿主仓库或者代理仓库获得实际构件的内容)。

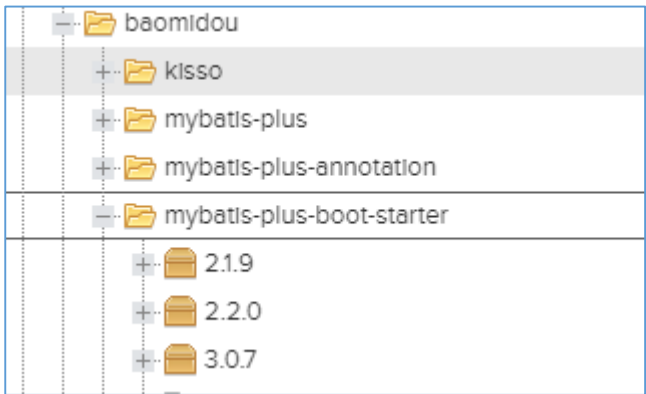


### 4.3 搜索包

如下图, Search→Maven→按照名字 mybatis-plus-boot-starter 搜索, 搜索出来的有三个版本, 这既是当前私仓中存在的。

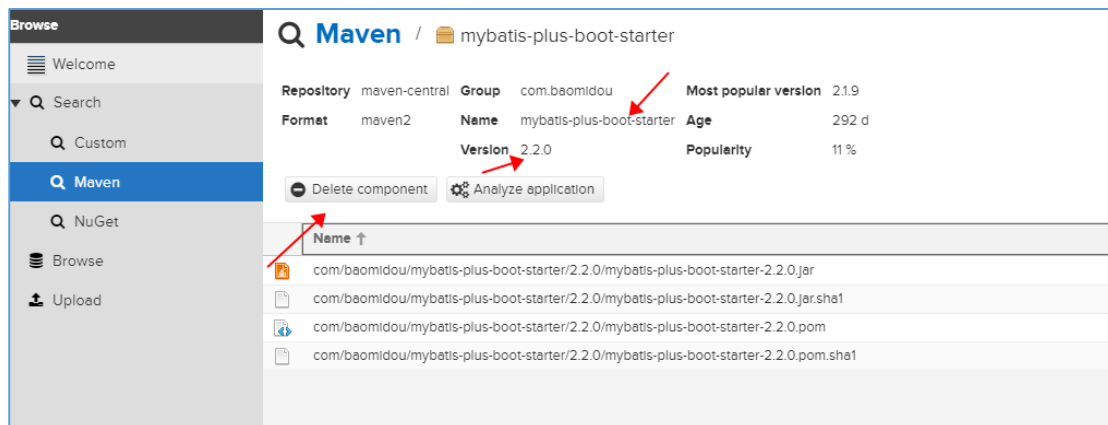


使用 Browse 浏览, 与上述结果一致。



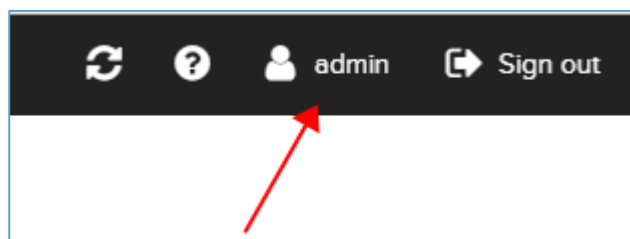
### 4.4 删除包

接上一步, 搜索出来后, 进去某一个版本, 如下图点击 Delete 删除即可。

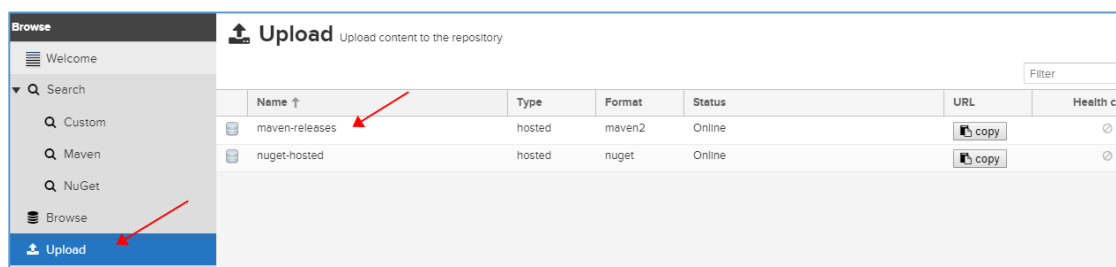


## 4.5 上传包

使用管理员登陆



点击 Upload，选择 maven-releases



**Choose assets for this component**

File: C:\fakepath\mybatis-plus-boot-starter.jar    Classifier:    Extension: jar

Add another asset

**Component coordinates**

Group ID: com.baomidou

Artifact ID: mybatis-plus-boot-starter

Version: 3.0.7

☒ Generate a POM file with these coordinates

Packaging:

Upload Cancel

file 处选择本地的 jar 包，Extension 自动识别为 jar，填写 Group ID(路径)、Artifact ID (包名)、Version (版本)，这三个参数也可以在官方文档中看到，如下

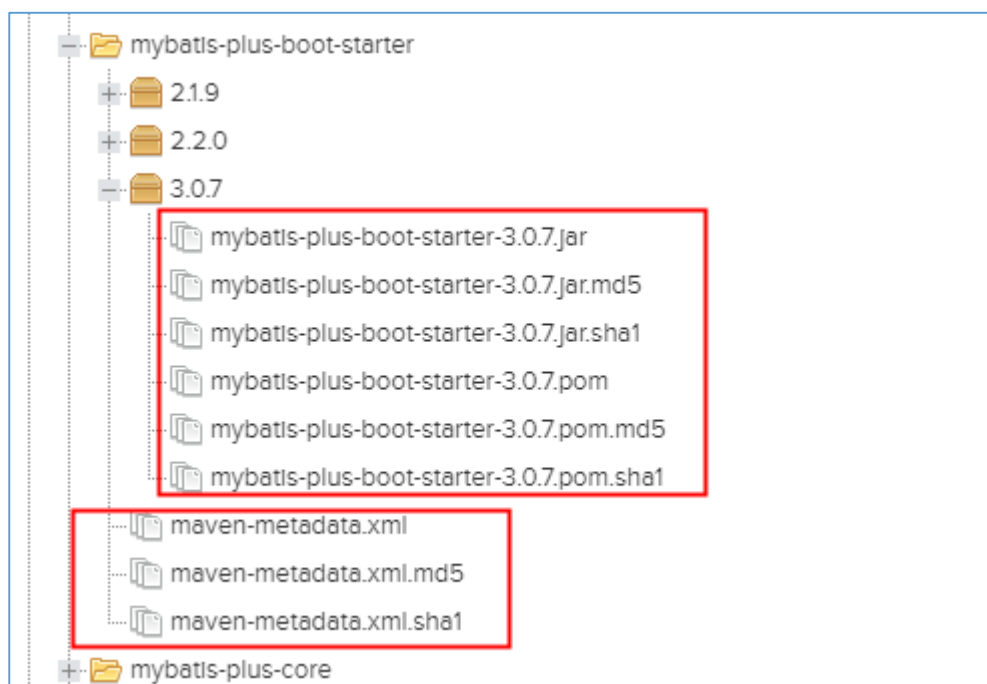
<https://mvnrepository.com/artifact/com.baomidou/mybatis-plus-boot-starter/3.0.7>

Maven    Gradle    SBT    Ivy    Grape    Leiningen    Buildr

```
<!-- https://mvnrepository.com/artifact/com.baomidou/mybatis-plus-boot-starter -->
<dependency>
  <groupId>com.baomidou</groupId>
  <artifactId>mybatis-plus-boot-starter</artifactId>
  <version>3.0.7</version>
</dependency>
```

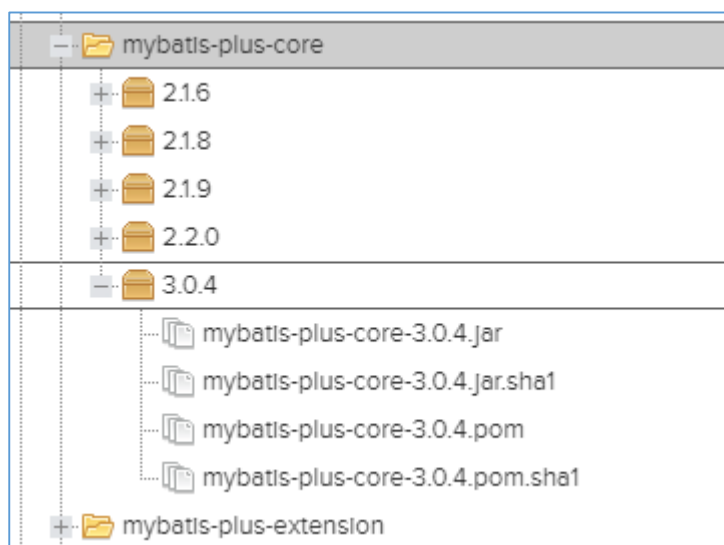
勾选"Generate a POM file with these coordinates"，将自动生产 pom 文件。

上传后的效果如下图：



3.0.7 下面有 6 个文件，和 3.0.7 并列的有 3 个文件，这和 nexus 自动从网上下载的不一样，属于正常现象。

如下是互联网上自动下载的目录结构。



## 4.6 其他内容

待补充，也可参考 <https://www.cnblogs.com/kevingrace/p/6201984.html>