

Nexus3.x安装及配置

若排版紊乱可查看[我的个人博客原文地址](#)

搭建maven私服主要是为了方便在一个团队中使用maven中央仓库没有的jar包，这些非中央仓库的jar包可能是自己团队开发的公共依赖jar包，可能是第三方私有的认证相关jar包，特殊的还有oracle的驱动jar包，下面是使用Nexus搭建maven私服及私服使用

Nexus3.x安装及配置

前置条件要求存在jdk1.8的环境

下载

这里之所以要特地提下下载是因为这个东西是真难下，我的电脑本身一直没有安装过专门的下载工具(度盘不算)，一般下载都是走网盘或者浏览器直接下载，这次无论度盘的离线下载还是浏览器下载都非常慢甚至下载失败(100M光纤)，在服务器上使用wget同样很慢，最后安装了迅雷使用迅雷下载后速度快很多，看来下载确实需要专业的下载工具，下载完成后上传到服务器上即可，当然也可以直接在服务器上使用wget下载，如果不嫌弃下载的慢的话：

官方下载页：

```
https://www.sonatype.com/download-oss-sonatype
```

实际资源地址

```
wget https://sonatype-download.global.ssl.fastly.net/repository/repositoryManager/3/nexus-3.10.0-04-unix.tar.gz
```

安装及配置

```
#解压
tar -zxvf nexus-3.10.0-04-unix.tar.gz
#解压的文件夹移到nexus (自动创建)
mv nexus-3.10.0-04/ /usr/local/nexus

#防火墙开启8081端口，该端口是nexus的默认端口 (在/usr/local/nexus/etc/nexus-default.properties文件中)
vi /etc/sysconfig/iptables
# 在打开的文件中加入如下内容
-A INPUT -p tcp -m state --state NEW -m tcp --dport 8081 -j ACCEPT

#启动 ps 我安装时第一次启动使用./nexus start 访问不到web页面，使用run命令可以，之后使用start也可以了
./nexus run &

#启动成功会有如下信息
WARNING: *****
WARNING: Detected execution as "root" user. This is NOT recommended!
WARNING: *****
Starting nexus
```

上面启动成功后会警告不要使用root用户启动，这里可以新建一个用户，也可以指定root用户启动，使他不出现警告，下面配置指定root用户启动

```
#该文件在bin目录下和启动程序nexus同级
vi nexus.rc
run_as_user=root
```

< 2018年5月 >						
日	一	二	三	四	五	六
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

导航

- 博客园
- 首页
- 新随笔
- 联系
- 订阅
- 管理

统计

- 随笔 - 7
- 文章 - 0
- 评论 - 0
- 引用 - 0

公告

昵称：[柠檬请问](#)
园龄：[1个月](#)
粉丝：[0](#)
关注：[0](#)
[+加关注](#)

搜索

找找看

谷歌搜索

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论
- 我的标签

我的标签

- linux(2)
- maven(2)
- mysql(1)
- nexus(1)

手动更新索引(可选)

我这里没有下载索引，以前配置过nexus2.x时下载过全部索引，很大很慢，下载了一晚上，不下载索引并不影响使用，这里会同步当前项目使用的依赖到私服的索引中，有新的依赖从私服中获取也会更新对应的索引，下面介绍一种离线安装的方式

- 1. 前往maven中央仓库下载 **indexer-cli-5.1.1.jar** 解压工具包
- 2. 下载离线**索引文件**

```
nexus-maven-repository-index.gz
nexus-maven-repository-index.properties
```

- 3. 将3个文件放在同一目录下解压

```
java -jar indexer-cli-5.1.1.jar -u nexus-maven-repository-index.gz -d indexer
```

- 4. 拷贝索引
nexus2.x拷贝到{nexus-home}/sonatype-work/nexus/indexer/central-ctx
nexus3.x拷贝到/sonatype-work/nexus3/blobs/default，3.x我没尝试过参考**这篇博客**

web界面

- 1. 访问web界面，使用默认管理账号admin/admin123登录
http://192.168.16.30:8081/
- 2. 仓库浏览在左侧菜单栏Browse,这里有多种仓库，相比于nexus2.x精简了很多仓库

```
nexus仓库类型
Group：这是一个仓库聚合的概念，用户仓库地址选择Group的地址，即可访问Group中配置的所有仓库资源，访问顺序取决于配置顺序3.x默认Releases，Snapshots，Central，可在web页面配置在web界面点开

Hosted:私有仓库，专门用来存储我们自己生成的jar文件
  3rd party：未发布到公网的第三方jar（3.x去除了）
  Snapshots：本地项目的快照仓库
  Releases： 本地项目发布的正式版本

Proxy:公网上发布的jar 例如：spring
  Central：中央仓库
  Apache Snapshots：Apache专用快照仓库（3.x去除了）
```

- 3. 配置Releases版本可重复上传

点击图片画框部分配置Releases: Deployment polcy --> Allow redeploy

环境搭建(1)
日志(1)
源码解析(1)
log4j(1)
APM(1)
javaagent(1)
更多

随笔档案

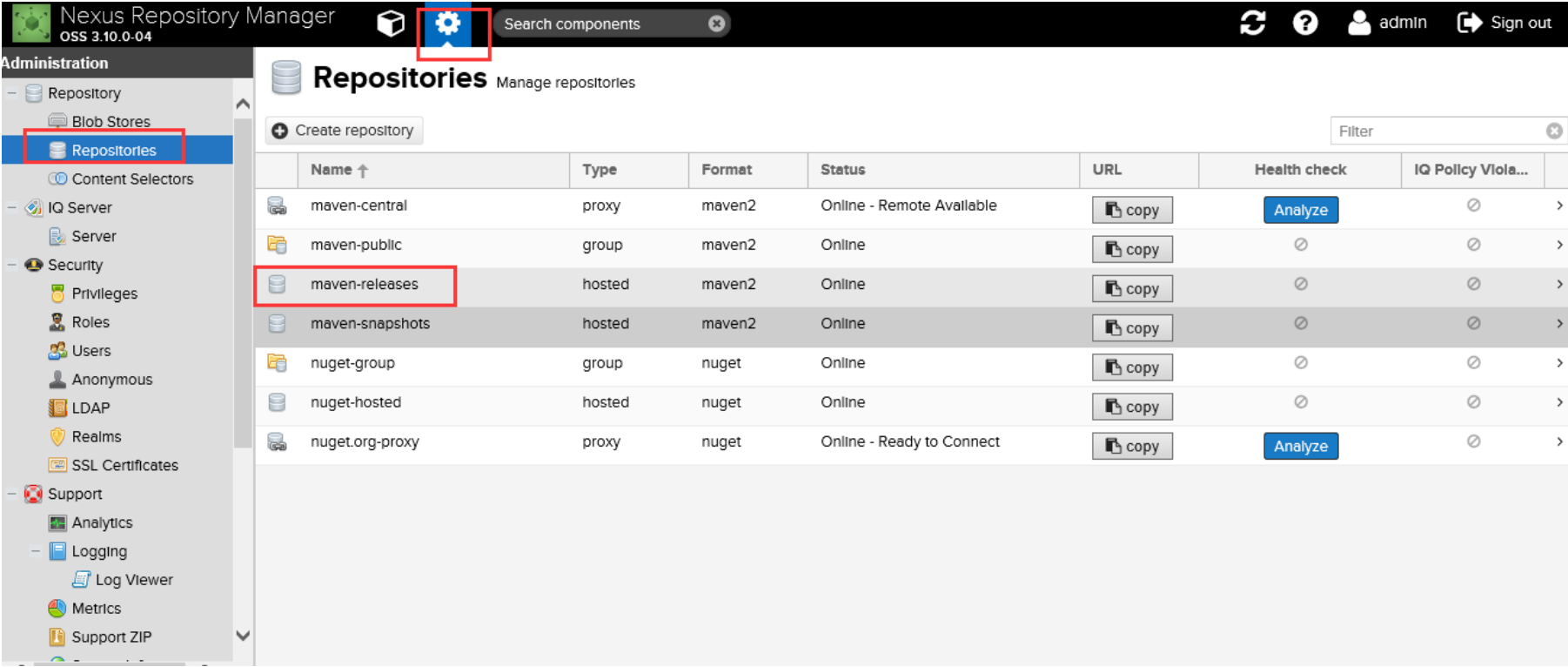
2018年4月（7）

阅读排行榜

- 1. Nexus3.x安装及配置(281)
- 2. linux链接及文件互相上传下载(33)
- 3. java后台常用json解析工具问题小结(29)
- 4. maven的依赖特性(8)
- 5. ThreadLocal解析(8)

推荐排行榜

- 1. Nexus3.x安装及配置(1)



使用nexus本地maven配置

全局配置下载依赖(即项目pom无需配置)

在maven的setting.xml文件中配置私服配置，这种方式配置后所有本地使用该配置的maven项目的pom文件都无需配置私服下载相关配置

```
<profiles>
  <profile>
    <id>mycof</id>
    <repositories>
      <!-- 私有库地址-->
      <repository>
        <id>nexus</id>
        <url>http://192.168.16.30:8081/repository/maven-public/</url>
        <releases>
          <enabled>true</enabled>
        </releases>
        <snapshots>
          <enabled>true</enabled>
        </snapshots>
      </repository>
    </repositories>
    <pluginRepositories>
      <!-- 插件库地址-->
      <pluginRepository>
        <id>nexus</id>
        <url>http://192.168.16.30:8081/repository/maven-public/</url>
        <releases>
          <enabled>true</enabled>
        </releases>
        <snapshots>
          <enabled>true</enabled>
        </snapshots>
      </pluginRepository>
    </pluginRepositories>
  </profile>
</profiles>
```

```
        </snapshots>
    </pluginRepository>
</pluginRepositories>
</profile>
```

激活使用上面的配置

```
<!--激活profile-->
<activeProfiles>
    <activeProfile>mycof</activeProfile>
</activeProfiles>
```

指定镜像代理为我们的私服

```
<mirror>
    <id>nexus-myself</id>
    <!--*指的是访问任何仓库都使用我们的私服-->
    <mirrorOf>*</mirrorOf>
    <name>Nexus myself</name>
    <url>http://192.168.16.30:8081/repository/maven-public/</url>
</mirror>
```

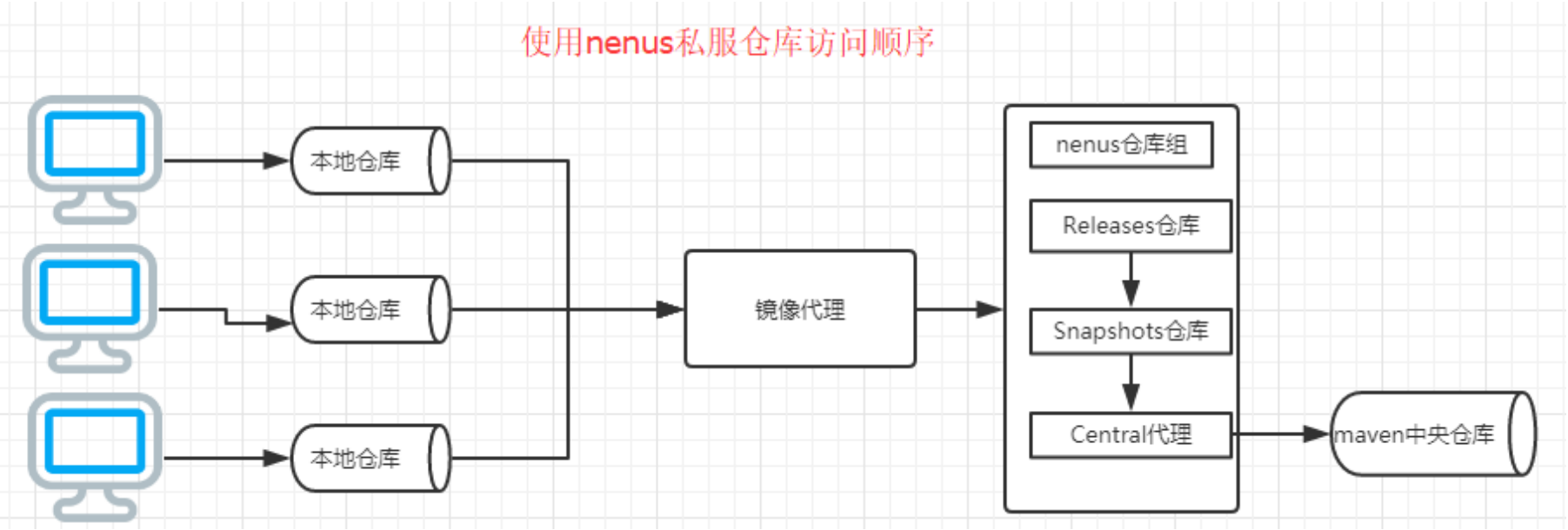
单独项目下载依赖(即项目pom文件中配置)

这种配置是修改单个项目的pom文件，无需修改maven的setting配置

```
<repositories>
    <repository>
        <id>nexus</id>
        <url>http://192.168.16.30:8081/repository/maven-public/</url>
        <releases>
            <enabled>true</enabled>
        </releases>
        <snapshots>
            <enabled>true</enabled>
        </snapshots>
    </repository>
</repositories>
```

建议使用全局的setting配置，一个项目组共用一个maven的setting配置，项目中就无需配置下载相关的私服仓库配置

使用私服后仓库访问顺序



上传jar包配置

对于中央仓库没有的jar包，需要我们自己将jar包发布到私服中去，其中jar包主要分为两类，一类是本地自己开发供给项目组其余同事使用，这种直接配置项目的pom文件和maven的setting文件，之后deploy发布即可发布到，另一类是第三方jar包，可以直接使用web页面上传并设置对应GAV即可

本地maaven开发的项目上传配置

1. maven的setting文件配置

这里要配置配置上传用户及仓库信息，我这里直接使用了admin用户，在nexus2.x中还内置了一个deployment/deployment123用户，在3.x该用户被移除了，这里可以自己配置角色和用户，我就直接使用admin用户了

```
<servers>
  <server>
    <id>nexus-releases</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
  <server>
    <id>nexus-snapshots</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
</servers>
```

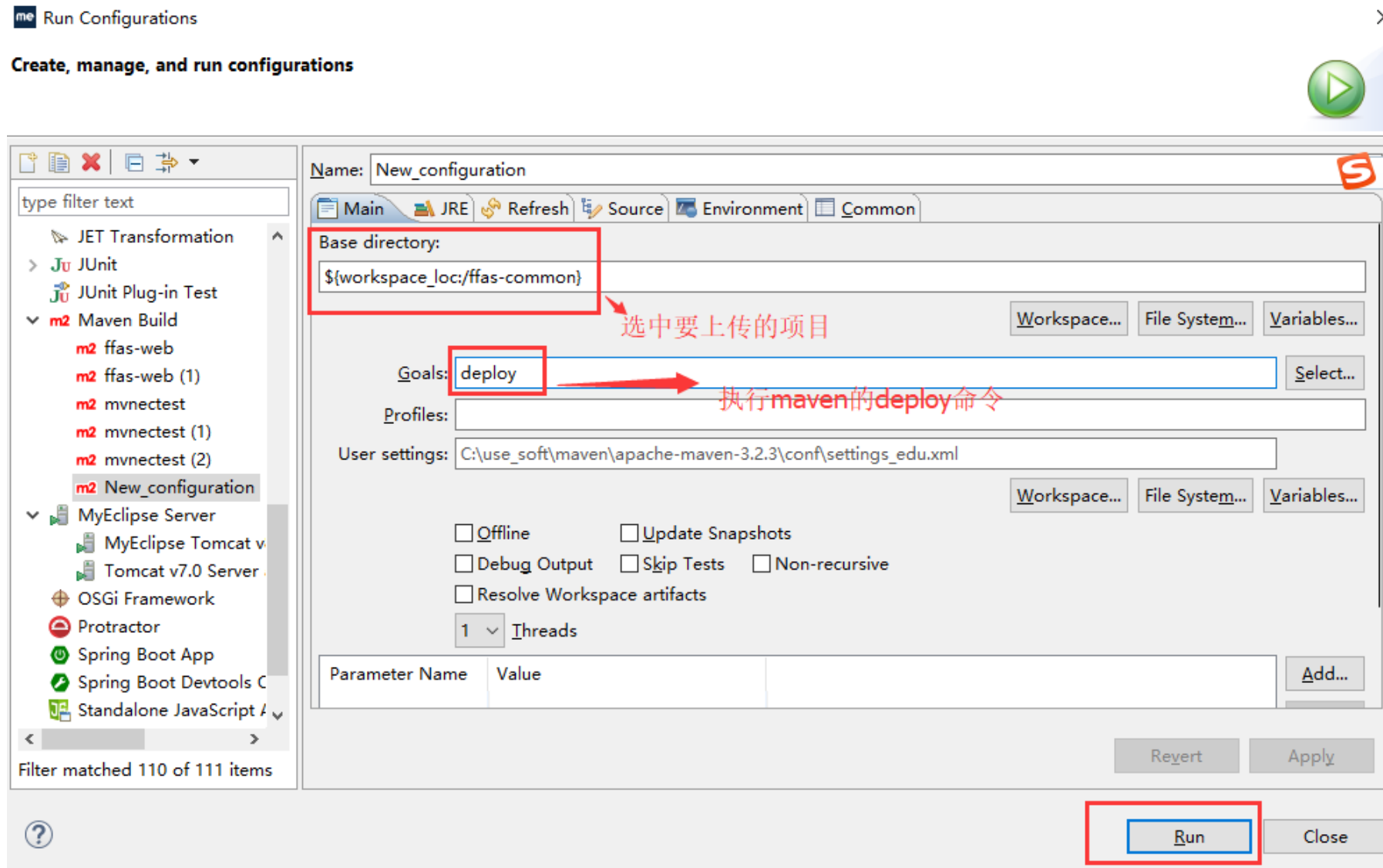
2. 项目中的pom文件配置

```
<distributionManagement>
  <repository>
    <id>nexus-releases</id>
    <name>Nexus Release Repository</name>
    <url>http://192.168.16.30:8081/repository/maven-releases/</url>
  </repository>
  <snapshotRepository>
    <id>nexus-snapshots</id>
    <name>Nexus Snapshot Repository</name>
    <url>http://192.168.16.30:8081/repository/maven-snapshots/</url>
  </snapshotRepository>
</distributionManagement>
```

```
</snapshotRepository>  
</distributionManagement>
```

3. 执行maven的deploy命令

这是myeclipse中上传本地项目jar到私服，IDEA中更为简单直接点击maven周期中的deploy即可



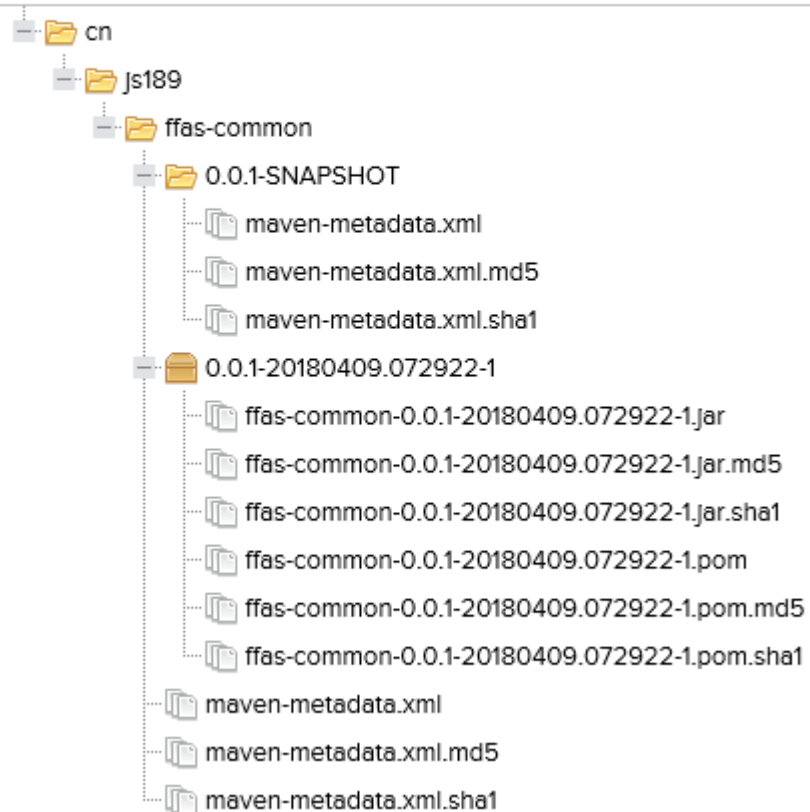
出现uploading信息并且没报错说明上传成功

```
[INFO] Downloading: http://192.168.16.30:8081/repository/maven-snapshots/cn/js
[INFO] Uploading: http://192.168.16.30:8081/repository/maven-snapshots/cn/js18
[INFO] Uploaded: http://192.168.16.30:8081/repository/maven-snapshots/cn/js18
[INFO] Uploading: http://192.168.16.30:8081/repository/maven-snapshots/cn/js18
[INFO] Uploaded: http://192.168.16.30:8081/repository/maven-snapshots/cn/js18
[INFO] Downloading: http://192.168.16.30:8081/repository/maven-snapshots/cn/js
[INFO] Uploading: http://192.168.16.30:8081/repository/maven-snapshots/cn/js18
[INFO] Uploaded: http://192.168.16.30:8081/repository/maven-snapshots/cn/js18
[INFO] Uploading: http://192.168.16.30:8081/repository/maven-snapshots/cn/js18
[INFO] Uploaded: http://192.168.16.30:8081/repository/maven-snapshots/cn/js18
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 11.394 s
[INFO] Finished at: 2018-04-09T15:29:23+08:00
[INFO] Final Memory: 23M/176M
[INFO] -----
```

到对应仓库查看

 **Browse** /  maven-snapshots

[HTML View](#)

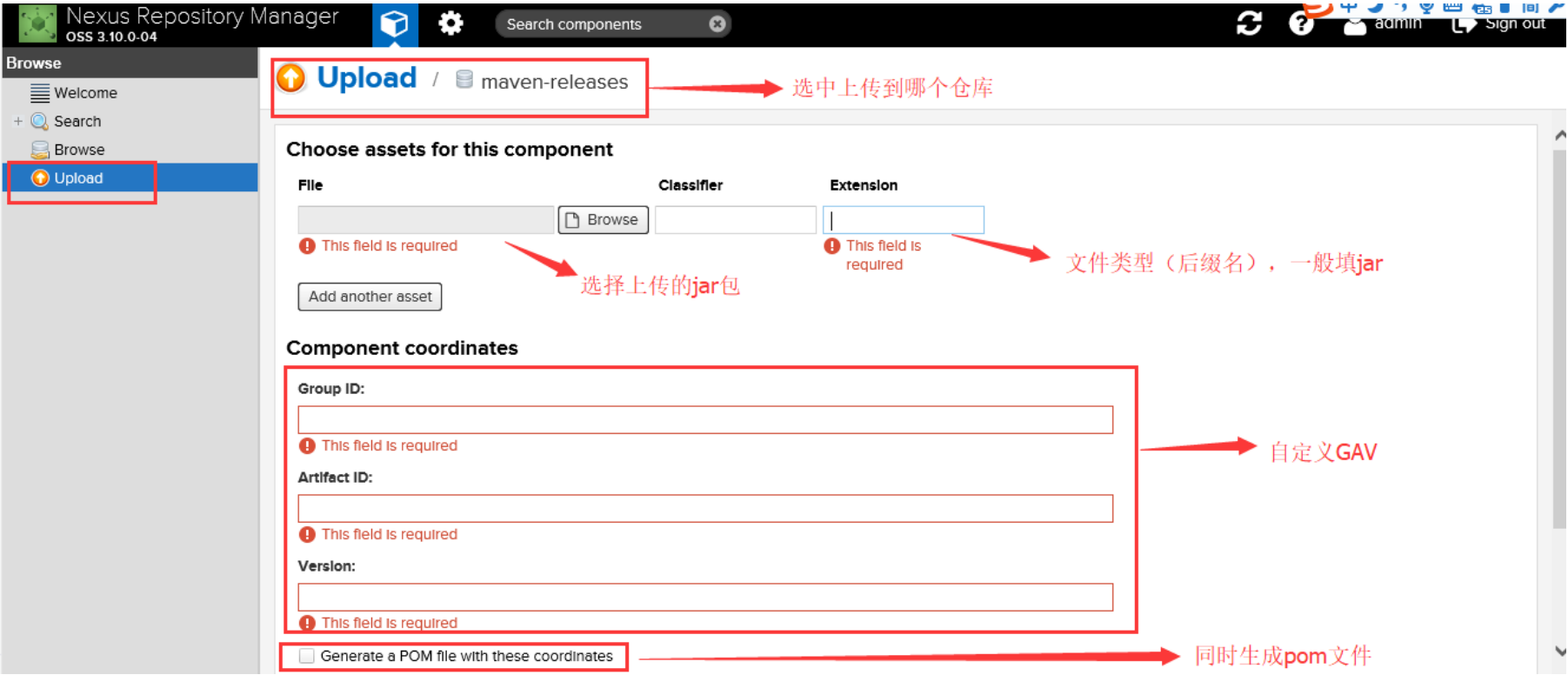


注意：

1. 这里只有Snapshot的仓库上传成功，Release仓库并没有上传成功，这是因为版本号名称结尾有SNAPSHOT，默认以这个结尾是不会上传到Release仓库去的，去除就会上传到Release仓库且不会上传到快照仓库
2. Release仓库默认不支持重复发布，可配置
3. 注意setting中server标签下的id要和pom文件中仓库配置的id相同，即指定上传到哪个仓库

第三方jar包上传

对于第三方jar包的上传采用nexus提供的web界面上传，上传成功后需要使用该jar包的话，依赖中填写自定义的GAV即可



标签: [maven](#), [环境搭建](#), [nexus](#)

好文要顶

关注我

收藏该文

柠檬请问

关注 - 0

粉丝 - 0

+加关注

« 上一篇：[linux链接及文件互相上传下载](#)
» 下一篇：[maven的依赖特性](#)

posted on 2018-04-12 14:47 [柠檬请问](#) 阅读(280) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)[网站](#)首页。



最新IT新闻:

- [方兴东评马云演讲：真正具备演出级水平 孤独求败](#)
 - [腾讯五月初已成“差评”工商股东 若退股怎么办？](#)
 - [七问“真假三文鱼”：有无淡水三文鱼？能生吃吗？](#)
 - [携程：经调查不存在“大数据杀熟” 将优化页面展示](#)
 - [马云：脱贫是给人以希望，而不仅仅是给人以钱财](#)
- » [更多新闻...](#)



最新知识库文章:

- [你可以把编程当做一项托付终身的职业](#)
 - [评审的艺术——谈谈现实中的代码评审](#)
 - [如何高效学习](#)
 - [如何成为优秀的程序员？](#)
 - [菜鸟工程师的超神之路 -- 从校园到职场](#)
- » [更多知识库文章...](#)