私服

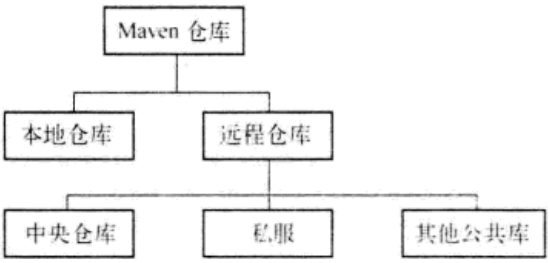
1.专用词：

1.构件： 指依赖，插件，jar，项目输出

2.快照版本： 不是稳定版本的构件，好比每天修改运行成功的

3.发布版本： 稳定版

2.maven仓库：



大概分为：

1.本地仓库

2.远程仓库

- **maven默认的中央仓库**
- **自建私服**
- **其它公开库**

maven根据坐标寻找构件的流程：

本地——》远程（私服——》中央仓库或者其它公共库，要看自己的配置指向）

3.本地仓库：

默认仓库：

windows在c:\users\juven\.m2\repository

linux在/home/juven/.m2/repository/

windows:

1.更改仓库：D:\java\repository

2.复制全局配置文件config/settings.xml 到 D:\java\repository/目录下

注意：不要更改全局settings.xml文件，因为将来升级带来麻烦

刚安装好maven，如果不执行maven命令，本地仓库目录是不存在的，当用户输入第一条maven命令，maven才会创建本地仓库，然后根据配置和需要，从远处仓库下载构件至本地仓库

3.安装

4.远程仓库：

1.中央仓库：

- **是maven的默认远程仓库，刚安装好maven时，由于本地仓库是空的，maven会去默认的远程仓库（即中央仓库）下载需要的构件**
- **maven中央仓库的配置路径：**

/lib/maven-model-builder-3.0.jar中org/apache/maven/model/pom-4.0.0.xml

```
<repositories>
 <repository>
  <id>central</id>
            <name>Maven Repository Switchboard</name>
  <url>http://repol.maven.org/maven2</url>
  <snapshots>
   <enabled>false</enabled>
  </snapshots>
```

</repository>
</repositories>
这是所有maven都会继承的超级POM

id: 是中央仓库的唯一标示，如果自定义仓库的id和中央仓库第id一样，会引用中央仓库的配置

name: 中央仓库的名称，随便取

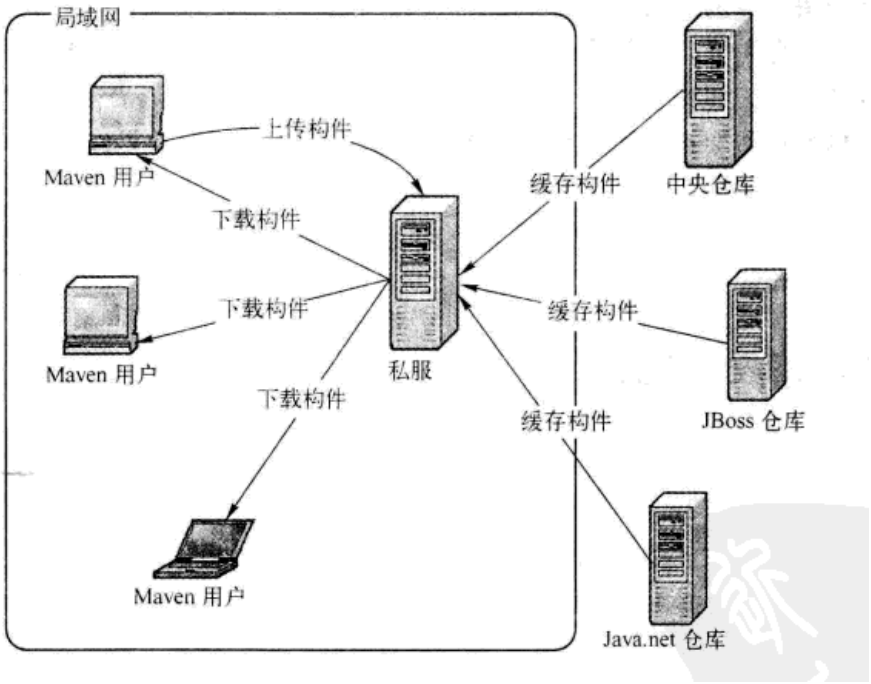
url: 中央仓库的url路径

snapshots:<enabled>元素为false代表不从该中央仓库下载快照版本的构件

2.私服：

- **作用： 架设在局域网内，代理广域网上的远程仓库，供局域网内的maven用户使用**
- **访问流程：先从本地仓库，没有找到，进入私服从私服请求，私服没有该构件，私服从外部**

远程仓库下载，先缓存在私服上，在存入本地仓库，最后为maven的下载请求提供服务，

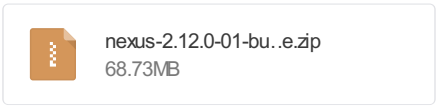


5.nexus：是私服工具

bundle方式安装nexus

1.下载：

windows: 下载nexus-webapp-1.7-bundle.zip



2.解压：

解压出两个目录：

1.nexus-webapp-1.7.2/: 该目录包含nexus运行所需要的文件，启动脚本，依赖jar包

2.sonatype-work/: 该目录包含nexus生成的配置文件，仓库文件，日志文件

3.运行：

进入nexus-web-1.7.2/bin/jsw/windows-x86-32/目录（或者x86-64），双击nexus.bat启动nesus，原理是nexus自带 Jetty容器

4.访问：

http://localhost:8081/nexus/

5.登录：

单击界面右上角Log In登录

默认管理员用户名：admin

密码： admin123

6.nexus内置 仓库

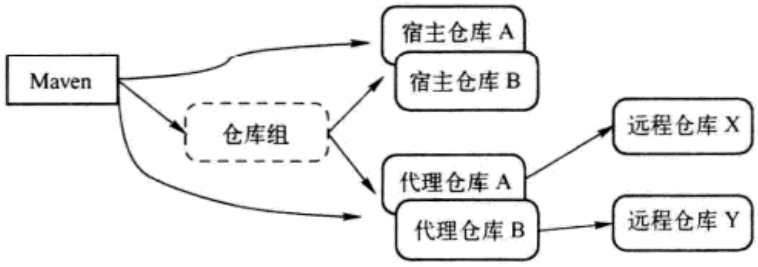
- 单击nexus左边pepositories，显示列表仓库：

group: 仓库组
hosted： 宿主仓库
proxy: 代理仓库
virtual 虚拟仓库
release：发布版本仓库
snapshot：快照版本仓库

- 仓库格式为maven1，maven2（包括maven3）
- policy策略表示该仓库为release发布版本仓库还是snapshot快照版本仓库
- 最后两列是仓库的状态和路径

Maven central: 该仓库代理maven中央仓库，只会下载和缓存中央仓库中的发布版本构件
3rd party： 用来部署 无法从公共仓库获得的第三方发布版本构件
Apache Snapshots: 用来代理Apache Maven仓库的快照版本构件
Codehaus Snapshots： 用来代理Codehaus maven仓库的快照版本构件
Google Code: 用来代理Google Code Maven仓库的发布版本构件
java.net-Maven2: 用来代理java.net Maven仓库的发布版本构件
Public Repositories: 该仓库组将上述所有策略为release的仓库聚合并通过一致的地址提供服务
PublicSnapshot Repositories: 该仓库组将上述所有策略为snapshot的仓库聚合并通过一致的地址提供服务
Release： 用来部署组织内部的发布版本构件
Snapshots： 用来部署组织内部快照版本构件

总结：在本地配置Public Repositories和PublicSnapshot Repositories仓库就能得到中央仓库构件，第三方构件，组织内部构件（各仓库的发布版本和快照版本）



6.nexus创建宿主仓库

1.单击界面左边导航栏Reposutories链接，在右边面板选择Add,然后在下拉菜单中选择hosted Repository

2，填上

- **ID: 仓库id**
- **Repository type：仓库类型**
- **Provider: 仓库格式（默认maven2 Repository）**
- **Repository Policy： 是发布版构件仓库还是快照版构件仓库**
- **Default Local Storage Location：仓库默认存储目录**

（默认目录：sonatype-work/nexus/storage/repository-id）

Deployment Policy: 部署策略

1.只读：禁止部署
2.关闭重新部署：同一构件只能部署一次
3.允许重新部署

Allow File Browsing: 是否允许浏览仓库内容，一般选true，（以树形结构浏览仓库存储文件）
Include insearch：该仓库进行索引并提供搜索
Publish URL: 控制是否通过URL提供服务
Not Found Cache TTL：当一个文件不存在时，缓存这一不存在信息的时间（默认1440分钟）

7.nexus索引与搜索

1.开启nexus central 索引（默认是关闭）点击central——》configuration——》

Download Remode Indexes——》true

由于maven中央仓库索引较多，需要很长时间下载，可以在界面左边Administration ——》Scheduled Tasks中看到有一条下载任务

8.nexus部署第三方构件

1.项目pom配置：

```
<distributionManagement>
 <repository><!-- 发布版本构件的仓库 -->
  <id>nexus-releases</id><!-- 私服中发布版本构件的宿主仓库id -->
  <name>Nexus Releases Repository</name><!-- 发布版本构件的宿主仓库名 -->
  <url>http://localhost:8081/nexus/content/repositories/releases</url><!-- 私服中发布版本构件的宿主仓库url -->
 </repository>
 <snapshotRepository><!-- 私服中快件版本的宿主仓库 -->
  <id>nexus-snapshots</id><!-- 私服中快件版本的宿主仓库 id -->
  <name>Nexus Snapshot Repository</name><!-- 私服中快件版本的宿主仓库名称 -->
  <url>http://localhost:8081/nexus/comtent/repositories/snapshots</url><!-- 私服中快件版本的宿主仓库 url -->
 </snapshotRepository>
</distributionManagement>
```

2.Settings.xml配置（本地仓库中的settings.xml，不要配全局的）

认证信息：

```
<server><!-- 往私服发布版本宿主库 部署构件的认证信息-->
 <id>nexus-releases</id><!--id必须 与 pom中需要认证的<distributionManagement>发布版本宿主库的id完全一致，正是这个id将认证信息与仓库配置联系在了一起-->
 <username>admin</username>
 <password>admin123</password>
</server>
<server><!-- 往私服快照版本宿主库 部署构件的认证信息-->
 <id>nexus-snapshots</id><!--id必须 与 pom中需要认证的<distributionManagement>快照版本宿主库的id完全一致，正是这个id将认证信息与仓库配置联系在了一起-->
 <username>admin</username>
 <password>admin123</password>
</server>
```

1.选择宿主仓库Releases——》Artifact——》GAV Definition(parameters)——》maven坐标***注意版本0.0.1不要后缀 ——》Select Artifact(s) to Upload——》Add Artifact——》Upload Artifact(s)

9.从nexus下载构件

1.在本地仓库目录settings.xml全局配置（也可以在pom中配置，但作用域只限于本项目）

镜像：

```
<mirror>
  <!-- 此镜像指向私服，任何远程仓库的请求都会转至此私服,此私服如果需要验证配置一个<server>即可,注意：镜像仓库屏蔽了被镜像仓库，当镜像仓库不稳定或者停止服务，maven将无法访问被镜像仓库-->
   <id>nexus</id>
  <name>Nexus</name>
   <mirrorOf>*</mirrorOf><!-- 匹配所有远程仓库-->
   <url>http://localhost:8081/nexus/content/groups/public</url><!--私服仓库组url，仓库组的好处是私服中所有仓库构件都能得到-->
  </mirror>
```

仓库：

```
<profile>
 <id>nexus</id><!-- 配置仓库和下面的插件的目的是：开启对快照版本下载的支持，当maven需要下载快照版和发布版构件的时候，它首先检查central,看该类型的构件是否支持，得到正面回答后，再根据镜像匹配规则转而访问私服仓库地址 -->
 <repositories>
  <repository>
   <id>central</id><!-- id为central是覆盖了超级pom中央仓库的配置，他们的url已无关紧要，因为所有请求都会通过镜像访问私服地址-->
```

```
<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>
```

激活：

```
<activeProfiles>
<activeProfile>nexus</activeProfile><!-- 将nexus这个profile激活，当执行maven构建的时候，激活的profile会将仓库配置应用到项目中去-->
</activeProfiles>
```

所有请求都会通过私服访问地址，配置仓库和插件的主要目的是开启对快照版本下载的支持，当maven需要下载发布版或快照版构件的时候，它首先检查central，看该类型的构件是否支持，得到正面的回答后，再根据镜像匹配规则转而访问私服仓库地址

10.nexus权限

默认：

1.admin:对nexus的完全控制

2.deployment：访问浏览仓库，搜索，上传部署构件，但无法对nexus进行任何配置

3.anonymous：对应了未登录的匿名用户，可以浏览仓库并进行搜索

自定义：

1.在左边界面点击Security——》Users——》右边界面Add——》Nexus User填写用户信息(在Role Management中add用户权限)——》点击Save

2.可以单击任何一个用户——》点击Role Tree查看以树形结构详细查看包含角色及进一步的权限

UI：Basic UI Privileges: 界面必须最基本的权限

UI:Repository Browser: 浏览仓库页面所需要的权限

UI:Basic UI Privileges: 访问nexus快速搜索栏及搜索页面所需要的权限

Repo: All Repositoriey(Read):给予用户读取所有仓库内容的权限（没有仓库的读清晰，用户无法看到实际仓库的内容，和从仓库下载构件）

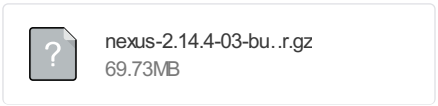
Repo:All Repositories(Full Control): 给予用户完全控制所有仓库内容的权限，用户不仅可以浏览，下载构件，还可以部署构件及删除仓库内容

3.一个特殊的匿名用户角色

Nexus Anonymous Role:默认配置下没有登录的用户都会拥有该匿名角色的权限，包含了除Repo：AllRepositories(Full Control)之外的所有角色所包含的权限

11.linux安装nexus

1.下载nexus-2.14.4-03-bundle.tar.gz



2.解压到/usr/local/目录

3.解压出两个目录nexus-*和sonatype-work

4.进入nexus-*/bin/,运行./nexus start（这样运行，应为我下载的是公用nexus）

会出现问题：

If you insist running as root, then set the environment variable RUN_AS_USER=root before running this script.

解决：

- 一种是临时解决： 输入：export RUN_AS_USER=root 后在执行 ./nexus start 运行
- 一种是永久方法：在系统用配置即可，输入：vi /etc/profile向其exportRUN_AS_USER=root，

修改后保存退出

访问：http://192.168.190.3:8081/nexus/

12.修改nexus端口：

默认端口是：**8081**

可以更改**/conf/plexus.properties,**找到**application-port**更改其端口，重启**nexus**