# Maven-Nexus私服

maven中的仓库分为两种，snapshot快照仓库和release发布仓库。snapshot快照仓库用于保存开发过程中的不稳定版本，release正式仓库则是用来保存稳定的发行版本。定义一个组件/模块为快照版本

maven会根据模块的版本号(pom文件中的version)中是否带有-SNAPSHOT来判断是快照版本还是正式版本。如果是快照版本，那么在mvn deploy时会自动发布到快照版本库中，而使用快照版本的模块，在不更改版本号的情况下，直接编译打包时，maven会自动从镜像服务器上下载最新的快照版本。如果是正式发布版本，那么在mvn deploy时会自动发布到正式版本库中，而使用正式版本的模块，在不更改版本号的情况下，编译打包时如果本地已经存在该版本的模块则不会主动去镜像服务器上下载。

所以，我们在开发阶段，可以将公用库的版本设置为快照版本，而被依赖组件则引用快照版本进行开发，在公用库的快照版本更新后，我们也不需要修改pom文件提示版本号来下载新的版本，直接mvn执行相关编译、打包命令即可重新下载最新的快照库了，从而也方便了我们进行开发。

决定发布的是正式版本release还是快照snapshot的是

<!--<version>1.0-SNAPSHOT</version>-->

<version>1.0.1-RELEASE</version>

版本号及发布类型，有一点也需要注意，snapshot快照允许重复发布更新同一个版本，而release是不允许的，release每次发布版本号都要加的。

### IntelliJ IDE使用私服配置

##### 一.修改setting.xml 添加如下：

<!-修改镜像地址,本地Maven直接连接私服Nexus,从私服获取jar包-->

<mirror>

<id>gme-group</id>

<mirrorOf>\*</mirrorOf>

<name>Central Repository</name>

<url>http://172.16.3.17:9090/repository/gme-group/</url>

</mirror>

<servers>

<!-- 发布版 -->

<server>

<id>gme-release</id>

<username>admin</username>

<password>admin123</password>

</server>

<!-- 测试版 -->

<server>

<id>gme-snapshot</id>

<username>admin</username>

<password>admin123</password>

</server>

</servers>

##### 工程pom.xml 添加如下配置：

发布到私服配置:

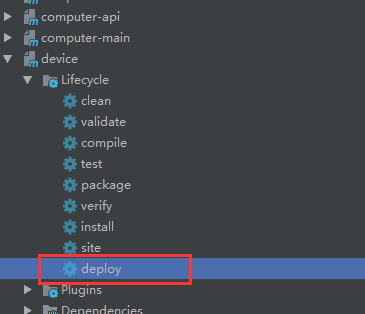
<distributionManagement>  
 <!--正式版本-->  
 <repository>  
 <id>gme-release</id>  
 <name>user release version</name>  
 <url>http://172.16.3.17:9090/repository/gme-release/</url>  
 </repository>  
 <!--快照-->  
 <snapshotRepository>  
 <id>gme-snapshot</id>  
 <name>user snapshots version</name>  
 <url>http://172.16.3.17:9090/repository/gme-snapshot/</url>  
 </snapshotRepository>  
</distributionManagement>

##### 上传jar到Nexus私服

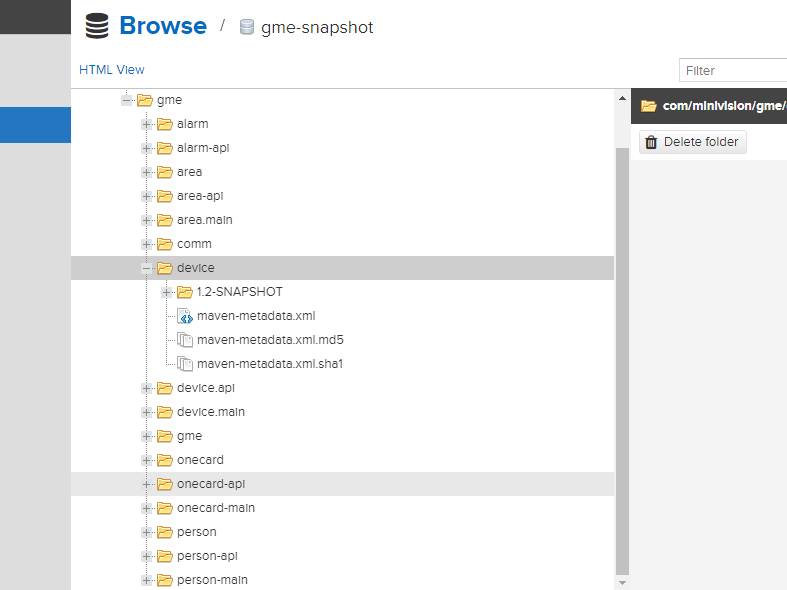
maven package：打包到本项目，一般是在项目target目录下。

maven install：打包会安装到本地仓库。

maven deploy：将打包的文件发布到远程仓库（私服）,提供其他人员进行下载依赖。这个在本地仓库也会生成







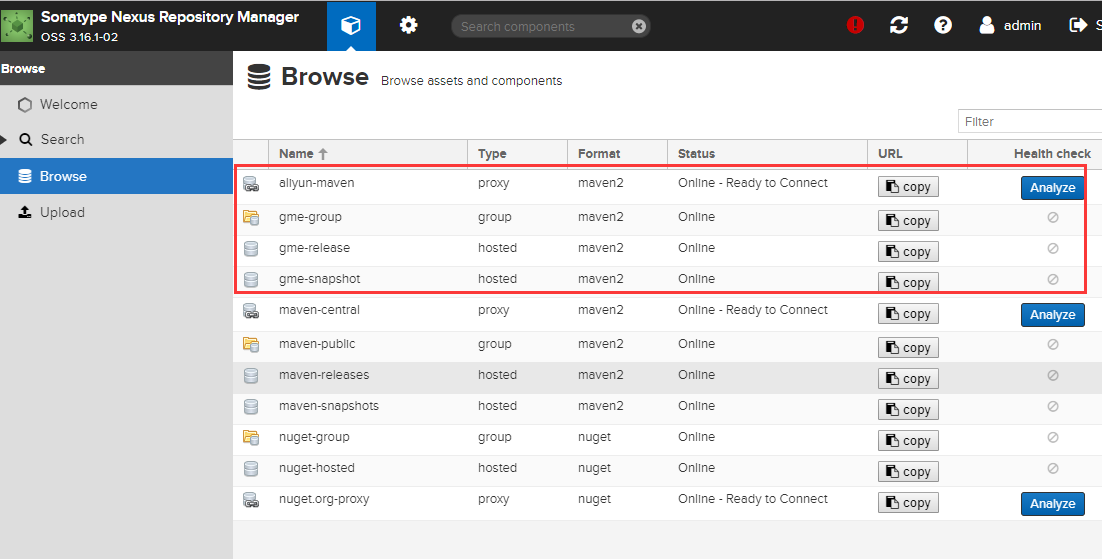
##### 私服地址

地址:http://172.16.3.17:9090/

账号:admin

密码:admin123

Nexus安装地址:/data/nexus-3.16.1-02



hosted，本地仓库，通常我们会部署自己的构件到这一类型的仓库。比如公司的第二方库。

proxy，代理仓库，它们被用来代理远程的公共仓库，如maven中央仓库。

group，仓库组，用来合并多个hosted/proxy仓库，当你的项目希望在多个repository使用资源时就不需要多次引用了，只需要引用一个group即可。

创建的仓库组没有Release和Snapshot之分，不同于宿主仓库和代理仓库，

在配置界面，用户可以选择Nexus中的仓库，聚合成一个虚拟仓库组。配置的顺序，

决定了仓库组遍历的顺序，所以，要把常用的配置在前面，保证当用户从仓库组下载

构件的时候，能够尽快的访问到包含构件的仓库，提高maven构件项目的效率。

