

# Nexus+Maven 安装配置手册

构架组:张栋芳([zhangdongfang@knet.cn](mailto:zhangdongfang@knet.cn))

2011-8-30

简介 .....	2
安装 .....	2
前提条件.....	2
安装步骤.....	2
配置 .....	4
安全管理.....	4
仓库类型.....	4
仓库策略.....	4
管理仓库.....	5
创建仓库.....	5
修改仓库.....	5
删除仓库.....	6
管理仓库组.....	6
创建仓库组.....	6
修改仓库组.....	7
删除仓库组.....	7
Nexus 内置仓库 .....	7
Nexus 索引设置 .....	8
部署构件到 Nexus .....	9
通过 Nexus UI 部署.....	9
通过 Maven 部署.....	11
搜索构件.....	12
Nexus 监听端口配置 .....	13
Nexus 调度任务 .....	13
整合 Maven .....	14
安装 Maven .....	14
设置 Maven 环境变量.....	15
设置 Maven 仓库地址指向 Nexus .....	15

# 简介

[Nexus](#) 是 Maven 仓库管理器，使用 Maven，你可以从 [Maven 中央仓库](#) 下载所需要的构件（artifact），但这通常不是一个好的做法，你应该在本地架设一个 Maven 仓库服务器，在代理远程仓库的同时维护本地仓库，以节省带宽和时间，Nexus 就可以满足这样的需要。此外，Nexus 还提供了强大的仓库管理功能，构件搜索功能，它基于 REST，友好的 UI 是一个 extjs 的 REST 客户端，它占用较少的内存，基于简单文件系统而非数据库。这些优点使其日趋成为最流行的 Maven 仓库管理器。

## 安装

### 前提条件

下载 nexus-oss-webapp-1.9.2.2-bundle.tar.gz 最新包,上传到服务器  
<http://nexus.sonatype.org/downloads>

### 安装步骤

1. 将 nexus-oss-webapp-1.9.2.2-bundle.tar.gz 解压到指定目录

```
[root@knet-dp04 nexus]# tar -zxvf nexus-oss-webapp-1.9.2.2-bundle.tar.gz
nexus-oss-webapp-1.9.2.2/
```

2. 进入 `/workspace/zhangdongfang/nexus/nexus-oss-webapp-1.9.2.2/bin/jsv` 目录

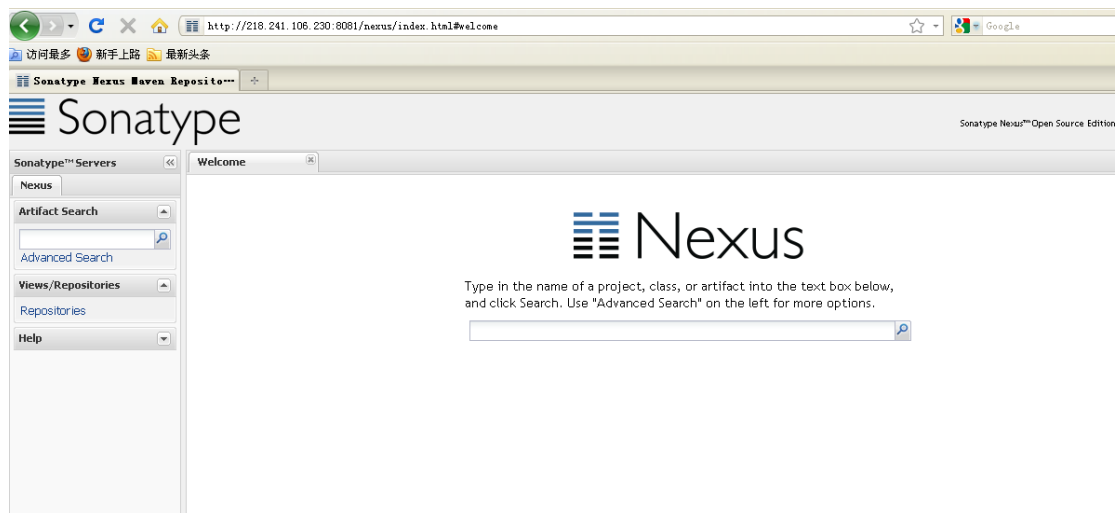
查看 Nexus 支持的 OS 版本信息

```
drwxr-xr-x 3 root root 4096 Aug 30 09:45 linux-ppc-64
drwxr-xr-x 3 root root 4096 Aug 30 09:45 linux-x86-32
drwxr-xr-x 3 root root 4096 Aug 30 09:45 linux-x86-64
drwxr-xr-x 3 root root 4096 Aug 30 09:45 macosx-universal-32
drwxr-xr-x 3 root root 4096 Aug 30 09:45 macosx-universal-64
drwxr-xr-x 3 root root 4096 Aug 30 09:45 solaris-sparc-32
drwxr-xr-x 3 root root 4096 Aug 30 09:45 solaris-sparc-64
drwxr-xr-x 4 root root 4096 Aug 30 09:45 solaris-x86-32
drwxr-xr-x 3 root root 4096 Aug 30 09:45 windows-x86-32
drwxr-xr-x 3 root root 4096 Aug 30 09:45 windows-x86-64
```

我们的服务器版本为 Linux-x86-64,选择对应的目录。进入 Linux-x86-64 目录来启动 Nexus

```
[root@knet-dp04 linux-x86-64]# ./nexus start
Starting Nexus OSS...
Started Nexus OSS.
```

通过访问 <http://218.241.106.230:8081/nexus> 地址出现如下图，表示 Nexus 安装成功

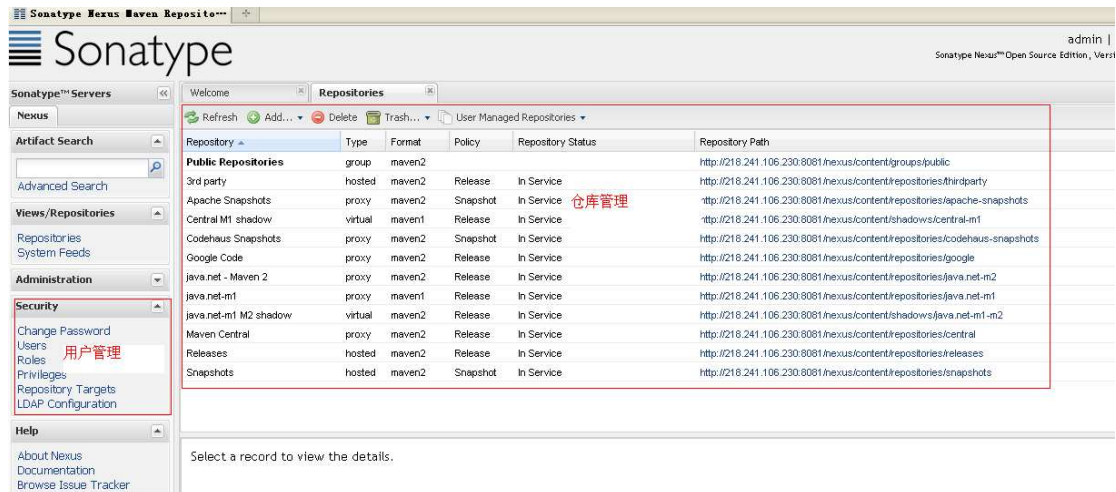


运行后会自动生成一个 nexus 工作目录 sonatype-work，nexus 下载的 jar 包会存放在 sonatype-work/nexus/storage 目录下。

# 配置

## 安全管理

使用 admin/admin123 用户登录 nexus 系统，登录后视图如下：



在 Security 选项中，用户可以修改密码、用户管理(创建、删除、修改)、角色管理、权限管理等一系列安全管理信息。

## 仓库类型

- group 仓库组
- proxy 代理仓库
- hosted 本地仓库
- virtual 虚拟仓库

## 仓库策略

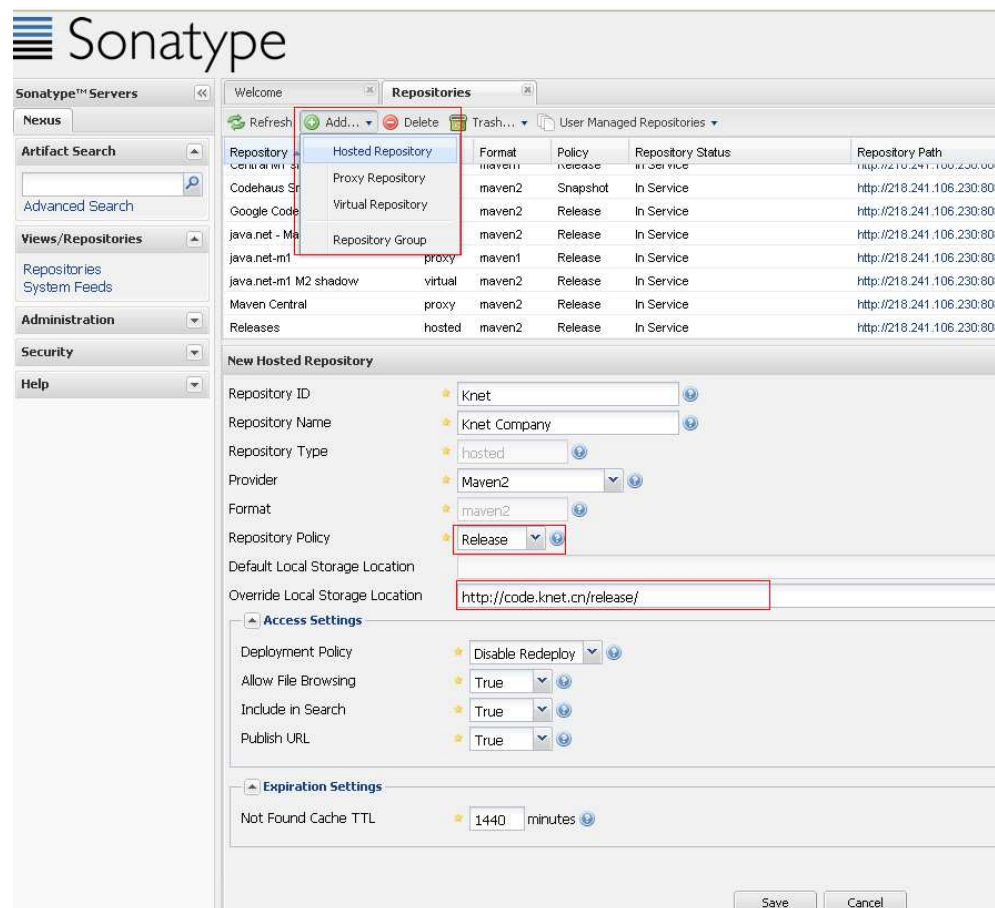
- Release 发布版本
- Snapshot 快照版本

## 管理仓库

Nexus 预定义了 3 个本地仓库，分别为 Releases，Snapshots，和 3rd Party。这三个仓库都有各自明确的目的。Releases 用于部署我们自己的 release 构件，Snapshots 用于部署我们自己的 snapshot 构件，而 3rd Party 用于部署第三方构件。

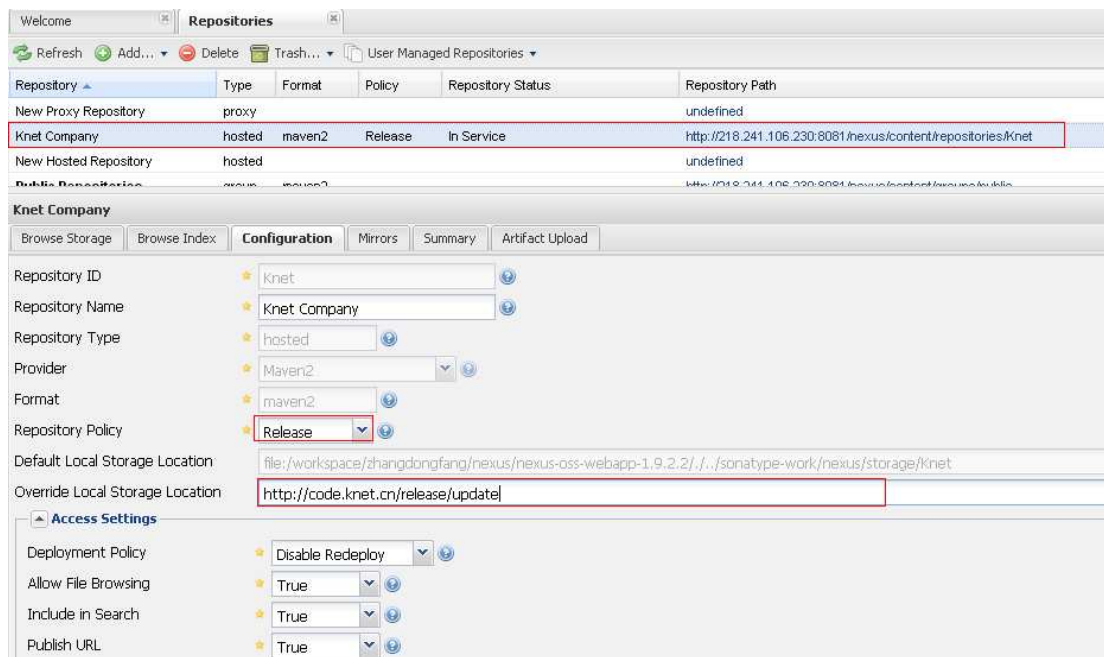
## 创建仓库

点击 Add...选择 Hosted Repository 如图所示:



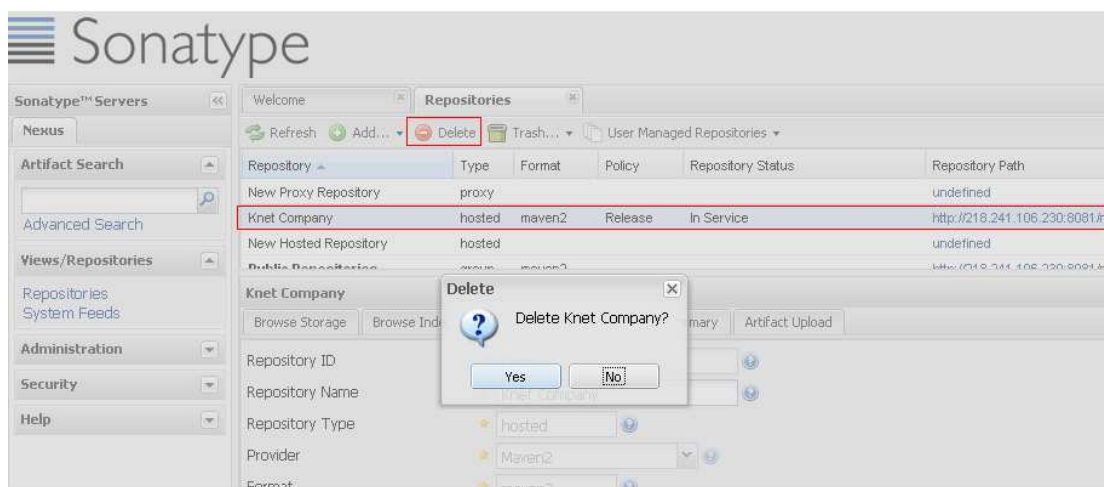
## 修改仓库

选择一条记录，再选择 Configuration Tab 页进行相关修改。



## 删除仓库

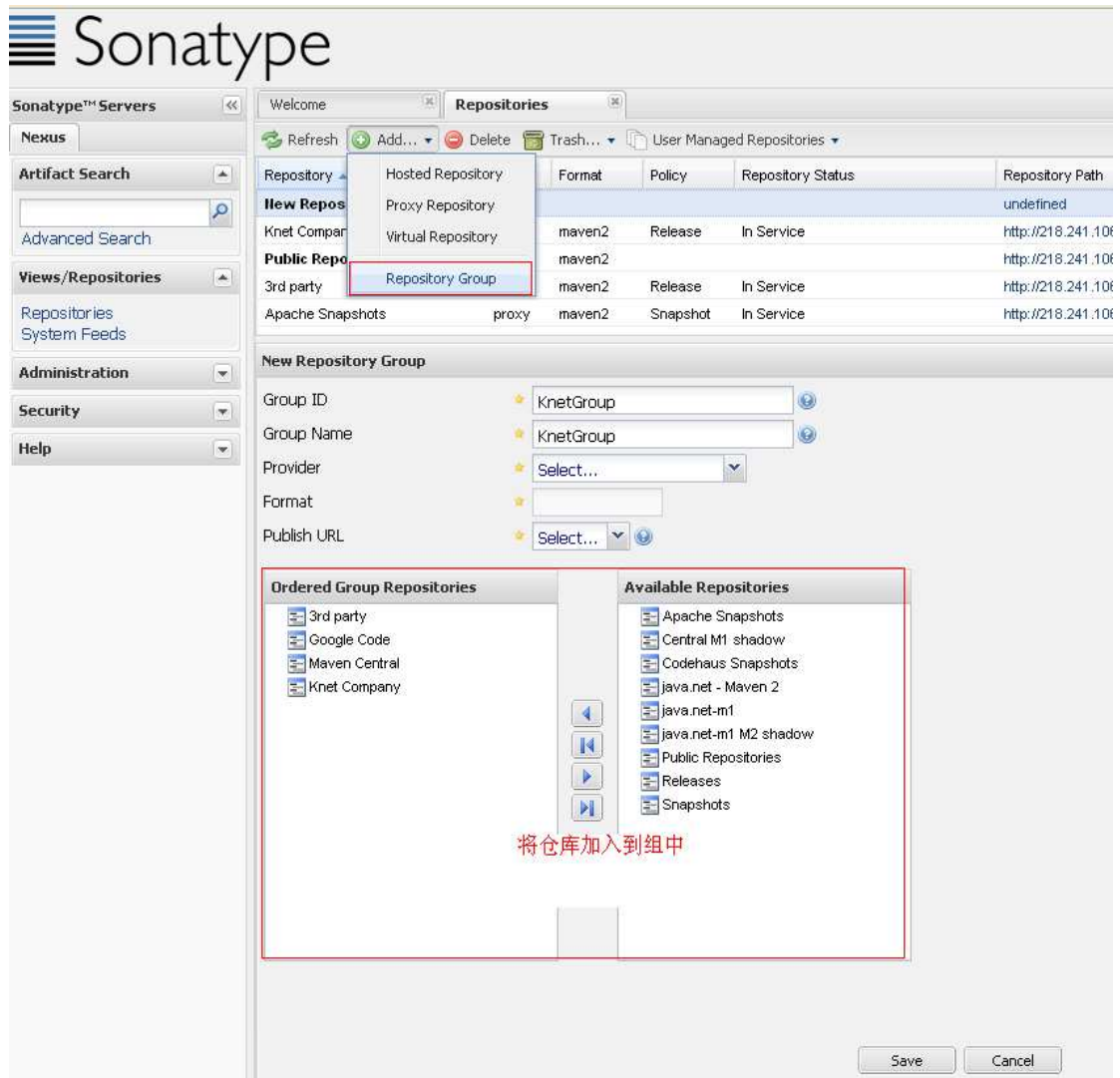
选择一条记录,点击 **Delete** 按钮。如图所示



## 管理仓库组

### 创建仓库组

点击 **Add...** 按钮,列表中选择 **Repository Group**,如图所示



## 修改仓库组

(同修改仓库)

## 删除仓库组

(同删除仓库)

## Nexus 内置仓库

➤ Maven Central Maven 中央仓库,策略为 Release,用于下载和缓存

中央仓库中发布的版本构件

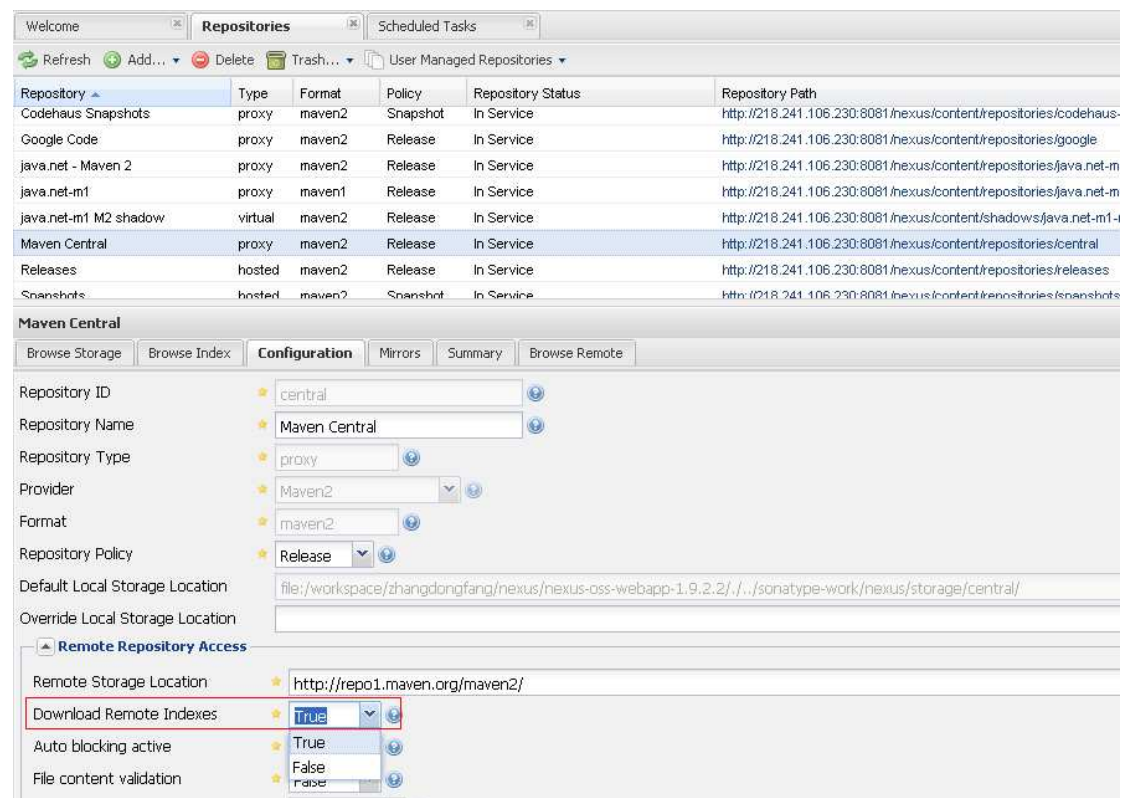
- Release 策略为 Release 的本地仓库，用于部署组织内部的发布版本构件
- Snapshots 策略为 Snapshot 的本地仓库,部署快照版本构件
- 3rd Party 策略为 Release 的本地仓库，部署第三方发布的版本构件
- Apache Snapshots 策略为 Snapshot 的代理仓库,代理 Apache Maven 仓库的快照版本构件
- Codehaus Snapshots 策略为 Snapshot 的代理仓库,代理 Codehaus Maven 仓库的快照版本构件
- Google Code 策略为 Release 的代理仓库,代理 Google Code Maven 仓库发布版本构件
- Java.net-Maven 2 策略为 Release 的代理仓库，代理 java.net Maven 仓库发布版本构件
- Public Repository 仓库中所有策略为 Release 仓库聚合构件
- Public Snapshots Repository 仓库中所有策略为 Snapshot 仓库聚合构件

## **Nexus 索引设置**

为了能够快速搜索构件的 Jar 文件，要设置 Nexus 中代理仓库下载远程索引，默认配置是关闭的。下面我们将开启 Maven 中央仓库的索引。选择 Maven Central 在页面中选择 Configuration 页签，设置



Download Remote Indexes 为 ”True” 点击 Save，其它代理仓库索引设置都一致。当设置 Download Remote Indexes 为 ”True”时，Nexus 会自动创建一个 Scheduled Tasks 将远程索引下载到本地。

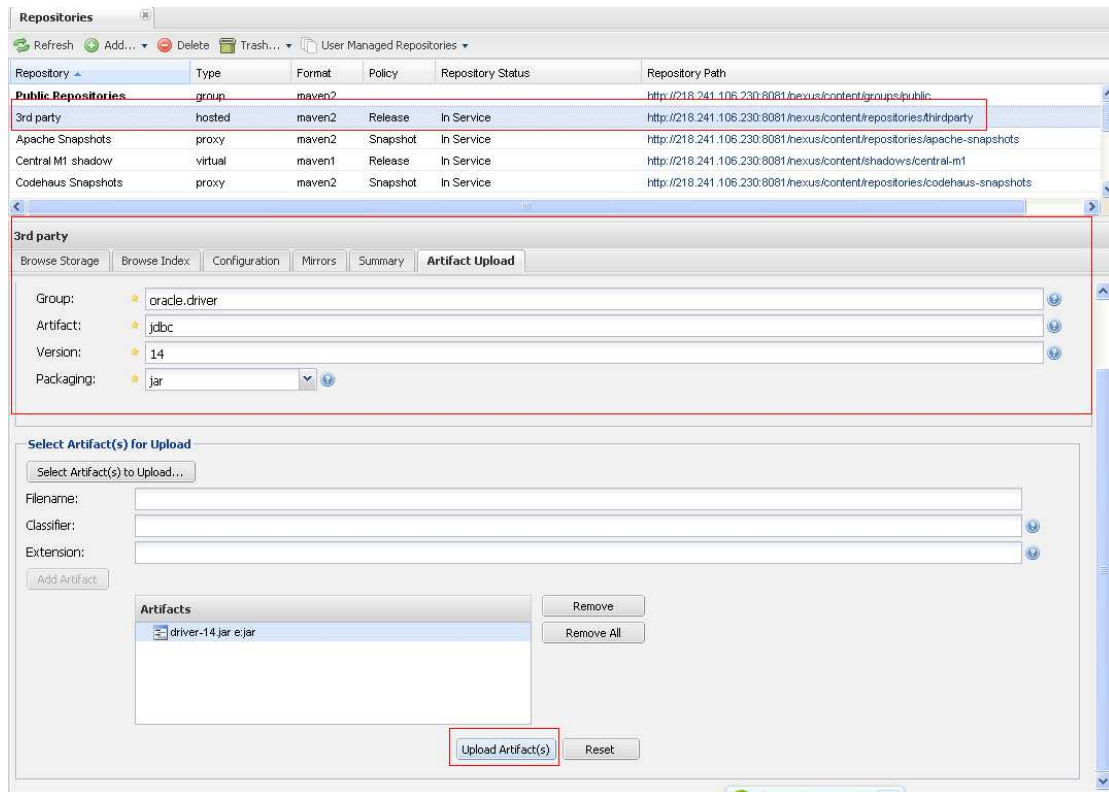


## 部署构件到 Nexus

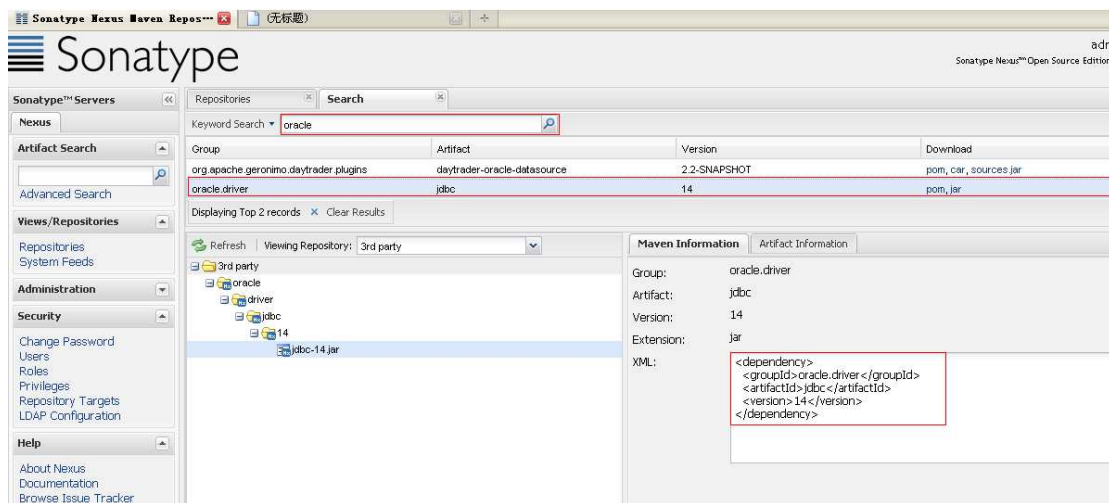
Nexus 提供了两种方式来部署构件，从 UI 直接上传或通过配置 Maven 部署构件。

### 通过 Nexus UI 部署

在右边的仓库列表选择一个仓库，如“3rd Party”，然后会看到页面下方有四个 tab，选择最后一个“Artifact Upload”如图所示：



上传完后,使用搜索查询部署结果



## 通过Maven 部署

### 1. 修改 maven/config/.setting.xml 文件

```
<servers>
  <server>
    <id>Releases</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
  <server>
    <id>Snapshots</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
</servers>
-----
```

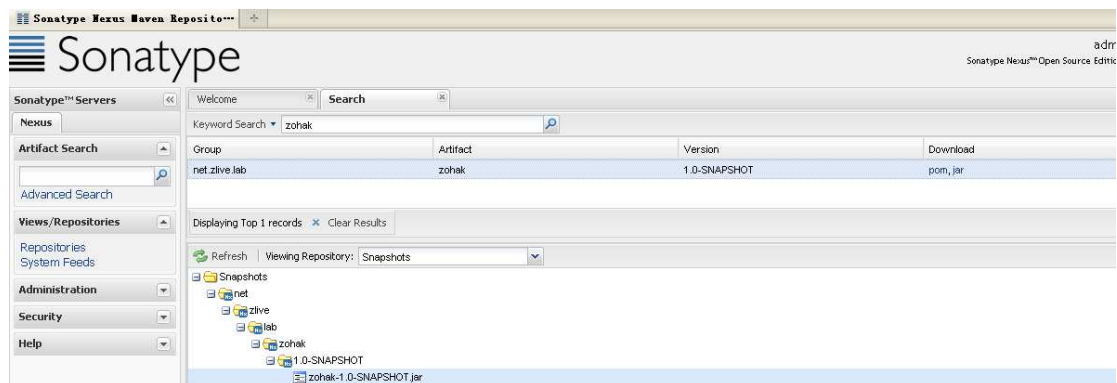
### 2. 修改 pom.xml 文件

```
<!-- 增加部署信息 -->
<distributionManagement>
  <!-- Release 版本 -->
  <repository>
    <!-- ID要与Maven中setting.xml中的ID要一致 -->
    <id>Releases</id>
    <name>Nexus Release Respository</name>
    <url>http://218.241.106.230:8081/nexus/content/repositories/releases/</url>
  </repository>
  <!-- Snapshot 版本 -->
  <snapshotRepository>
    <!-- ID要与Maven中setting.xml中的ID要一致 -->
    <id>Snapshots</id>
    <name>Nexus Snapshot Repository</name>
    <url>http://218.241.106.230:8081/nexus/content/repositories/snapshots/</url>
  </snapshotRepository>
</distributionManagement>
```

### 3. 在项目中执行部署 mvn deploy 命令

```
E:\static>mvn deploy
[INFO] Scanning for projects...
[INFO] -----
[INFO] Building zohak for bbking project
[INFO] Uploading repository metadata for: 'artifact net.zlive.lab:zohak'
[INFO] Uploading project information for zohak 1.0-20110830.034137-1
[INFO] Retrieving previous metadata from Snapshots
[INFO] repository metadata for: 'snapshot net.zlive.lab:zohak:1.0-SNAPSHOT'
not be found on repository: Snapshots, so will be created
[INFO] Uploading repository metadata for: 'snapshot net.zlive.lab:zohak:1.0-
HOT'
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
```

## 4. 使用搜索查询部署结果

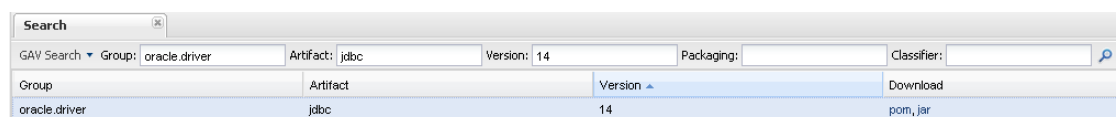


我们配置所有的 snapshot 版本构件部署到 Nexus 的 Snapshots 仓库中，所有的 release 构件部署到 Nexus 的 Releases 仓库中。由于部署需要登陆，因为我们在 settings.xml 中配置对应 Repository id 的用户名和密码。

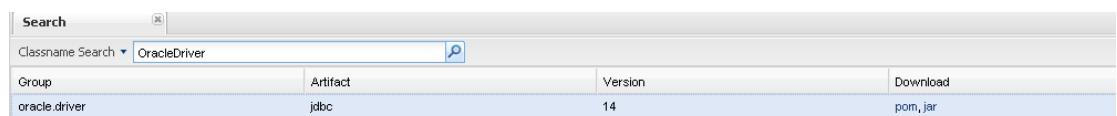
然后，在项目目录中执行 **mvn deploy**，你会看到 maven 将项目构件部署到 Nexus 中，浏览 Nexus 对应的仓库，就可以看到刚才部署的构件。当其他人构建其项目时，Maven 就会从 Nexus 寻找依赖并下载。

## 搜索构件

### 1. GAV 搜索;通过 groupId,artifactId,version 来搜索



### 2. 类名搜索;通过包含 Java 类名来搜索



### 3. 校验搜索

Search			
Checksum Search <input type="text"/> <a href="#">Browse...</a>			
Group	Artifact	Version	Download

### 4. 关键字搜索

Search			
Keyword Search <input type="text" value="oracle.driver"/> <a href="#">Browse...</a>			
Group	Artifact	Version	Download
oracle.driver	jdbc	14	<a href="#">pom, jar</a>

## Nexus 监听端口配置

默认情况下，Nexus 监听端口 8081。你可以更改这个端口，通过更改 `${NEXUS_HOME}/conf/plexus.properties` 的值，为此，停止 Nexus，更改文件中 `applicationPort` 的值，然后重启 Nexus

## Nexus 调度任务

Nexus 提供了一系列可配置的调度任务来方便用户管理系统，用户可以设定这些任务运行方式，例如每天、每周、手动等，Nexus 包含了以下几种类型的调度任务：

- **Download Indexs** 为代理仓库下载远程索引
- **Empty Trash** 清空 Nexus 的回收站
- **Evict Unused Proxied Items From Respository Caches** 清空代理仓库长期未使用的缓存。
- **Expire Repository Caches** 清空仓库中过期的缓存。
- **Pushlish Indexs** 发布仓库索引。
- **Purge Nexus Timeline** 删除 Nexus 时间线文件。

- Repair Repository Index 为仓库重组索引
- Remove Snapshots From Repository 删除仓库快照。
- Synchronized Shadow Repository 同步虚拟仓库内容。
- Update Repository Index 更新仓库索引

创建新任务如下:

The screenshot shows the 'Scheduled Tasks' configuration window. The 'Add' button in the top toolbar is highlighted with a red box. Below the toolbar, the 'Task Type' dropdown is set to 'Download Indexes' and is also highlighted with a red box. The 'Recurrence' dropdown is set to 'Hourly' and is highlighted with a red box. The 'Name' field is set to 'MavenCentral'. The 'Repository/Group' field is set to 'Knet Company (Repo)'. The 'Alert Email' field is set to 'zhangdongfang@knet.cn'. The 'Start Date' field is set to '08/30/2011' and the 'Start Time' field is set to '14:19'.

## 整合 Maven

### 安装 Maven

1. 下载 apache-maven-2.2.1-bin.tar.gz
2. 解压 apache-maven-2.2.1-bin.tar.gz 到指定目录

## 设置 Maven 环境变量

```
export MAVEN_HOME=/workspace/zhangdongfang/maven/apache-maven-3.0.3
PATH=.:$PATH:$MYSQL_HOME/bin:$HOME/bin:$MAVEN_HOME/bin
```

## 设置 Maven 仓库地址指向 Nexus

配置 AVEN\_HOME/conf/setting.xml 文件如下:

```
<!--配置本地jar存放位置-->
<localRepository>/workspace/zhangdongfang/maven/mavenRepo</localRepository>
<!--配置代理信息-->
<proxies>
  <proxy>
    <id>normal</id>
    <active>true</active>
    <protocol>http</protocol>
    <username>admin</username>
    <password>admin123</password>
    <host>218.241.106.230:8081/nexus</host>
    <port>8081</port>
    <nonProxyHosts>218.241.106.230:8081/nexus</nonProxyHosts>
  </proxy>
</proxies>
<!--配置镜像-->
<mirrors>
  <mirror>
    <id>nexus-public-snapshots</id>
    <mirrorOf>public-snapshots</mirrorOf>
    <url>
      http://218.241.106.230:8081/nexus/content/groups/public-snapshots
    </url>
  </mirror>
  <mirror>    <!--This sends everything else to /public -->
    <id>nexus</id>
    <mirrorOf>*</mirrorOf>
    <url>http://218.241.106.230:8081/nexus/content/groups/public
    </url>
  </mirror>
</mirrors>

<profiles>
  <profile>
    <id>development</id>
    <repositories>
      <repository>
        <id>central</id>
        <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>
<profile>
    <id>public-snapshots</id>
    <repositories>
        <repository>
            <id>public-snapshots</id>
            <url>http://public-snapshots</url>
            <releases><enabled>false</enabled></releases>
            <snapshots>
                <enabled>true</enabled>
            </snapshots>
        </repository>
    </repositories>
    <pluginRepositories>
        <pluginRepository>
            <id>public-snapshots</id>
            <url>http://public-snapshots</url>
            <releases><enabled>false</enabled></releases>
            <snapshots>
                <enabled>true</enabled>
            </snapshots>
        </pluginRepository>
    </pluginRepositories>
</profile>
</profiles>
<activeProfiles>
    <activeProfile>development</activeProfile>
</activeProfiles>
<!--服务器端Releases与Snapshots访问设置用于Maven部署构件文件到仓库-->
<servers>
    <server>
        <id>Releases</id>
        <username>admin</username>
        <password>admin123</password>
    </server>
    <server>
        <id>Snapshots</id>
        <username>admin</username>
        <password>admin123</password>
    </server>
</servers>

```