

Maven 私服 Nexus3.x 环境构建操作记录

Maven 介绍

Apache Maven 是一个创新的软件项目管理和综合工具。

Maven 提供了一个基于项目对象模型（POM）文件的新概念来管理项目的构建，可以从一个中心资料片管理项目构建，报告和文件。

Maven 最强大的功能就是能够自动下载项目依赖库。

Maven 提供了开发人员构建一个完整的生命周期框架。开发团队可以自动完成项目的基础工具建设，Maven 使用标准的目录结构和默认构建生命周期。

在多个开发团队环境时，Maven 可以设置按标准在非常短的时间里完成配置工作。由于大部分项目的设置都很简单，并且可重复使用，Maven 让开发人员的工作更轻松，同时创建报表，检查，构建和测试自动化设置。

Maven 项目的结构和内容在一个 XML 文件中声明，pom.xml 项目对象模型（POM），这是整个 Maven 系统的基本单元。

Maven 提供了开发人员的方式来管理：

- 1) Builds
- 2) Documentation
- 3) Reporting
- 4) Dependencies
- 5) SCMs
- 6) Releases
- 7) Distribution
- 8) mailing list

概括地说，Maven 简化和标准化项目建设过程。处理编译，分配，文档，团队协作和其他任务的无缝连接。

Maven 增加可重用性并负责建立相关的任务。

Maven 最初设计，是以简化 Jakarta Turbine 项目的建设。在几个项目，每个项目包含了不同的 Ant 构建文件。JAR 检查到 CVS。

Apache 组织开发 Maven 可以建立多个项目，发布项目信息，项目部署，在几个项目中 JAR 文件提供团队合作和帮助。

Maven 主要目标是提供给开发人员：

- 1) 项目是可重复使用，易维护，更容易理解的一个综合模型。
- 2) 插件或交互的工具，这种声明性的模式。

私服介绍

私服是指私有服务器，是架设在局域网的一种特殊的远程仓库，目的是代理远程仓库及部署第三方构建。有了私服之后，当 **Maven** 需要下载构件时，直接请求私服，私服上存在则下载到本地仓库；否则，私服请求外部的远程仓库，将构件下载到私服，再提供给本地仓库下载。



Nexus 介绍

Nexus 是一个强大的 **Maven** 仓库管理器，它极大地简化了本地内部仓库的维护和外部仓库的访问。

如果使用了公共的 **Maven** 仓库服务器，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。

正常做法是在本地架设一个 **Maven** 仓库服务器，即利用 **Nexus** 私服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 在代理远程仓库的同时维护本地仓库，以降低中央仓库的负荷,节省外网带宽和时间，**Nexus** 私服就可以满足这样的需要。

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。

Nexus 使用 **ExtJS** 来开发界面，利用 **Restlet** 来提供完整的 **REST APIs**，通过 **m2eclipse** 与 **Eclipse** 集成使用。

Nexus 支持 WebDAV 与 LDAP 安全身份认证。

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。

为什么要构建 Nexus 私服？

如果没有 Nexus 私服，我们所需的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 仓库下载构件无疑加大了仓库的负载和浪费了外网带宽，如果网速慢的话，还会影响项目的进程。很多情况下项目的开发都是在内网进行的，连接不到 maven 仓库怎么办呢？开发的公共构件怎么让其它项目使用？这个时候我们不得不为自己的团队搭建属于自己的 maven 私服，这样既节省了网络带宽也会加速项目搭建的进程，当然前提条件就是你的私服中拥有项目所需的所有构件。

总之，在本地构建 nexus 私服的好处有：

- 1) 加速构建；
- 2) 节省带宽；
- 3) 节省中央 maven 仓库的带宽；
- 4) 稳定（应付一旦中央服务器出问题的情况）；
- 5) 控制和审计；
- 6) 能够部署第三方构件；
- 7) 可以建立本地内部仓库；
- 8) 可以建立公共仓库

这些优点使得 Nexus 日趋成为最流行的 Maven 仓库管理器。

Maven 的安装

下载地址：<http://maven.apache.org/download.cgi>

提前在服务器上安装 jdk 环境（参考：[Centos 中 yum 方式安装 java](#)）

```
[root@master-node ~]# cd /usr/local/src/
```

```
[root@master-node src]# wget http://mirrors.hust.edu.cn/apache/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.tar.gz
```

```
[root@master-node src]# tar -zxvf apache-maven-3.3.9-bin.tar.gz
```

```
[root@master-node src]# mv apache-maven-3.3.9 /usr/local/maven
```

接着配置系统环境变量，在/etc/profile 文件底部添加如下内容：

```
[root@master-node src]# java -version
openjdk version "1.8.0_111"
OpenJDK Runtime Environment (build 1.8.0_111-b15)
OpenJDK 64-Bit Server VM (build 25.111-b15, mixed mode)
[root@master-node src]# vim /etc/profile
.....
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk           //java 的环境变量设置
export CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export PATH=$PATH:$JAVA_HOME/bin

export MAVEN_HOME=/usr/local/maven                        //maven 的环境变量设置
export PATH=$PATH:$MAVEN_HOME/bin
[root@master-node src]# source /etc/profile
```

最后验证是否安装成功，出现如下信息，说明安装成功

```
[root@master-node src]# mvn --version           # 最好按照 java jdk
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)
Maven home: /usr/local/maven
Java version: 1.8.0_111, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.111-2.b15.el7_3.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "3.10.0-327.el7.x86_64", arch: "amd64", family: "unix"
```

Nexus 安装

Nexus 的安装有两种实现方式：

1) war 包安装方式

下载地址: <https://sonatype-download.global.ssl.fastly.net/nexus/oss/nexus-2.14.2-01.war>

直接将 war 包放在 tomcat 的根目录下, 启动 tomcat 就可以用了

2) 源码安装方式 (之前在用的是 2.14.4 版本, 这里是新版本)

下载地址: <https://www.sonatype.com/download-oss-sonatype>

```
[root@master-node ~]# cd /usr/local/src/
```

```
[root@master-node src]# wget https://sonatype-download.global.ssl.fastly.net/nexus/3/nexus-3.2.0-01-unix.tar.gz
```

```
[root@master-node src]# tar -zxvf nexus-3.2.0-01-unix.tar.gz
```

```
[root@master-node src]# mv nexus-3.2.0-01 /usr/local/nexus
```

启动 nexus (默认端口是 8081)

```
[root@master-node src]# /usr/local/nexus/bin/nexus
```

```
WARNING: *****
```

```
WARNING: Detected execution as "root" user. This is NOT recommended!
```

```
WARNING: *****
```

```
Usage: /usr/local/nexus/bin/nexus {start|stop|run|run-redirect|status|restart|force-reload}
```

```
[root@master-node src]# /usr/local/nexus/bin/nexus start
```

```
WARNING: *****
```

```
WARNING: Detected execution as "root" user. This is NOT recommended!
```

```
WARNING: *****
```

Starting nexus

上面在启动过程中出现告警: 不推荐使用 root 用户启动。这个告警不影响 nexus 的正常访问和使用。

去掉上面 WARNING 的办法:

```
[root@master-node src]# vim /etc/profile
```

```
.....
```

```
export RUN_AS_USER=root
```

```
[root@master-node src]# source /etc/profile
```

```
[root@master-node src]# lsof -i:8081 //nexus 服务启动成功后，需要稍等一段时间，8081 端口才起来
```

```
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
```

```
java 1486 root 859u IPv4 23504303 0t0 TCP *:tproxy (LISTEN)
```

在部署机上的 iptables 里打开 8081 端口

```
[root@master-node src]# vim /etc/sysconfig/iptables
```

```
....
```

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 8081 -j ACCEPT
```

```
[root@master-node src]# /etc/init.d/iptables restart
```

访问 nexus，即 <http://localhost:8081> （如果出现 404，就访问 <http://localhost:8081/nexus>）



Browse



Welcome



+ Search



- Browse



Assets



Components



Components

Browse components and assets

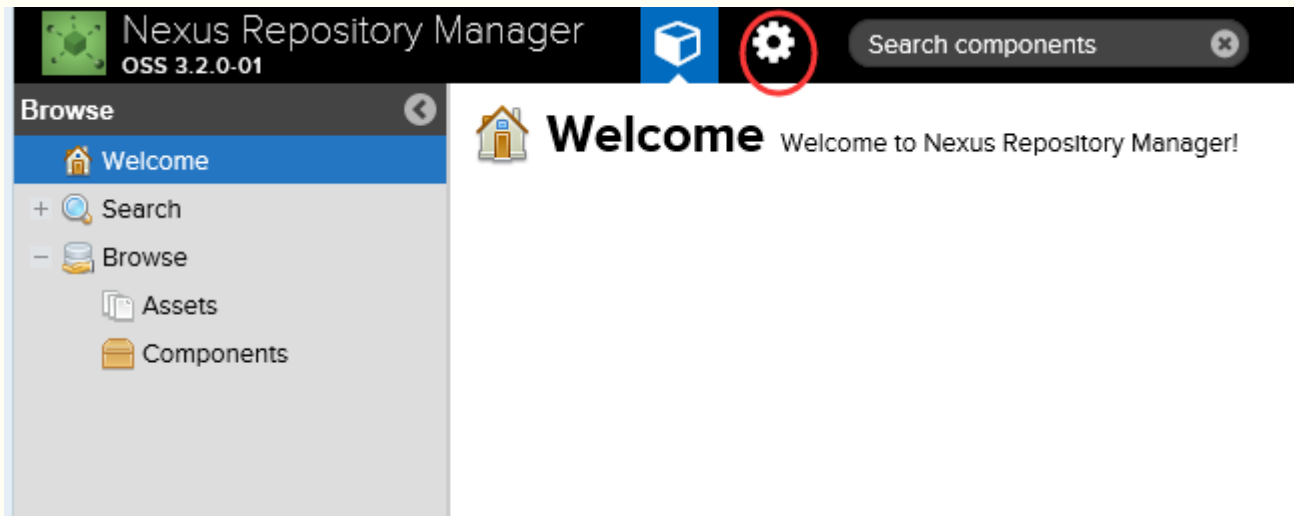
	Name ↑	Type	Format	Status
	maven-central	proxy	maven2	Online - R
	maven-public	group	maven2	Online
	maven-releases	hosted	maven2	Online
	maven-snapshots	hosted	maven2	Online
	nuget-group	group	nuget	Online
	nuget-hosted	hosted	nuget	Online
	nuget.org-proxy	proxy	nuget	Online - R


出现上述页面，说明配置 nexus 成功！

点击右上角“Log in”，



输入默认用户名(admin)和默认密码（admin123）登录

可以点击上面的“设置”图标，在“设置”里可以添加用户、角色，对接 LDAP 等的设置，如下：





Nexus Repository Manager
OSS 3.2.0-01



Search components

Administration

Repository

Blob Stores

Repositories

Content Selectors

Security

Privileges

Roles

Users

Anonymous

LDAP

Realms

SSL Certificates

Support

Analytics

Logging

Log Viewer

Metrics

Support ZIP

System Information

System

Bundles

Capabilities

Email Server

HTTP

Licensing



Users Manage users

 Create user

Source:  Default

Filter by user ID

User ID ↑	Realm	First name
 admin	default	Administrator
 anonymous	default	Anonymous
 wangshiibo	default	wang

可以在“管理”里查看 **nexus** 的系统信息



Administration

- Repository
 - Blob Stores
 - Repositories
 - Content Selectors
- Security
 - Privileges
 - Roles
 - Users
 - Anonymous
 - LDAP
 - Realms
 - SSL Certificates
- Support
 - Analytics
 - Logging
 - Log Viewer
 - Metrics
 - Support ZIP
- System Information
- System
 - Bundles
 - Capabilities
 - Email Server
 - HTTP
 - Licensing



System Information

Shows system information



Download

nexus-properties

application-host	0.0.0.0
application-port	8081
fabric.etc	/usr/local/nexus/etc/fabric
jetty.etc	/usr/local/nexus/etc/jetty
karaf.base	/usr/local/nexus
karaf.data	/usr/local/sonatype-work/nexus3
karaf.etc	/usr/local/nexus/etc/karaf
karaf.home	/usr/local/nexus
karaf.instances	/usr/local/sonatype-work/nexus3/instances
logback.etc	/usr/local/nexus/etc/logback
nexus-args	/usr/local/nexus/etc/jetty/jetty.xml,/usr/local/nexus/etc/jetty/jetty-ht
nexus-context-path	/
nexus-edition	nexus-oss-edition
nexus-features	nexus-oss-feature
ssl.etc	/usr/local/nexus/etc/ssl

nexus-license

licenseRequired	false
licenseValid	false
licenseInstalled	false

注意下面几点说明：

1.component name 的一些说明：

- 1) maven-central: maven 中央库，默认从 <https://repo1.maven.org/maven2/> 拉取 jar
- 2) maven-releases: 私库发行版 jar
- 3) maven-snapshots: 私库快照（调试版本）jar
- 4) maven-public: 仓库分组，把上面三个仓库组合在一起对外提供服务，在本地 maven 基础配置 settings.xml 中使用。

2.Nexus 默认的仓库类型有以下四种：

- 1) group(仓库组类型): 又叫组仓库，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）；

3.Policy(策略):表示该仓库为发布(Release)版本仓库还是快照(Snapshot)版本仓库；

4.Public Repositories 下的仓库

- 1) 3rd party: 无法从公共仓库获得的第三方发布版本的构件仓库，即第三方依赖的仓库，这个数据通常是由内部人员自行下载之后发布上去；
 - 2) Apache Snapshots: 用了代理 ApacheMaven 仓库快照版本的构件仓库
 - 3) Central: 用来代理 maven 中央仓库中发布版本构件的仓库
 - 4) Central M1 shadow: 用于提供中央仓库中 M1 格式的发布版本的构件镜像仓库
 - 5) Codehaus Snapshots: 用来代理 CodehausMaven 仓库的快照版本构件的仓库
 - 6) Releases: 内部的模块中 release 模块的发布仓库，用来部署管理内部的发布版本构件的宿主类型仓库；release 是发布版本；
 - 7) Snapshots: 发布内部的 SNAPSHOT 模块的仓库，用来部署管理内部的快照版本构件的宿主类型仓库；snapshots 是快照版本，也就是不稳定版本
- 所以自定义构建的仓库组代理仓库的顺序为：Releases, Snapshots, 3rd party, Central。也可以使用 oschina 放到 Central 前面，下载包会更快。

5.Nexus 默认的端口是 8081，可以在 [etc/nexus-default.properties](#) 配置中修改。

6.Nexus 默认的用户名密码是 [admin/admin123](#)

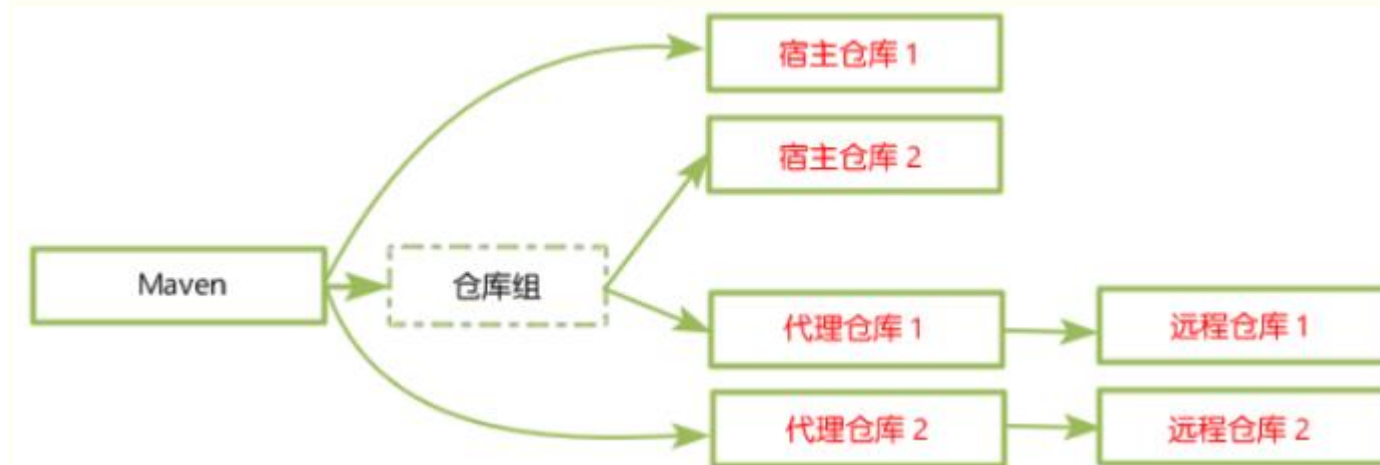
7.当遇到奇怪问题时，重启 nexus，重启后 web 界面要 1 分钟左右后才能访问。

8.Nexus 的工作目录是 [sonatype-work](#)（路径一般在 nexus 同级目录下）

```
[root@master-node local]# pwd
/usr/local
[root@master-node local]# ls nexus/
bin deploy etc lib LICENSE.txt NOTICE.txt public system
[root@master-node local]# ls sonatype-work/
nexus3
[root@master-node local]# ls sonatype-work/nexus3/
backup blobs cache db elasticsearch etc generated-bundles health-check instances keystores lock log orient port tmp
```

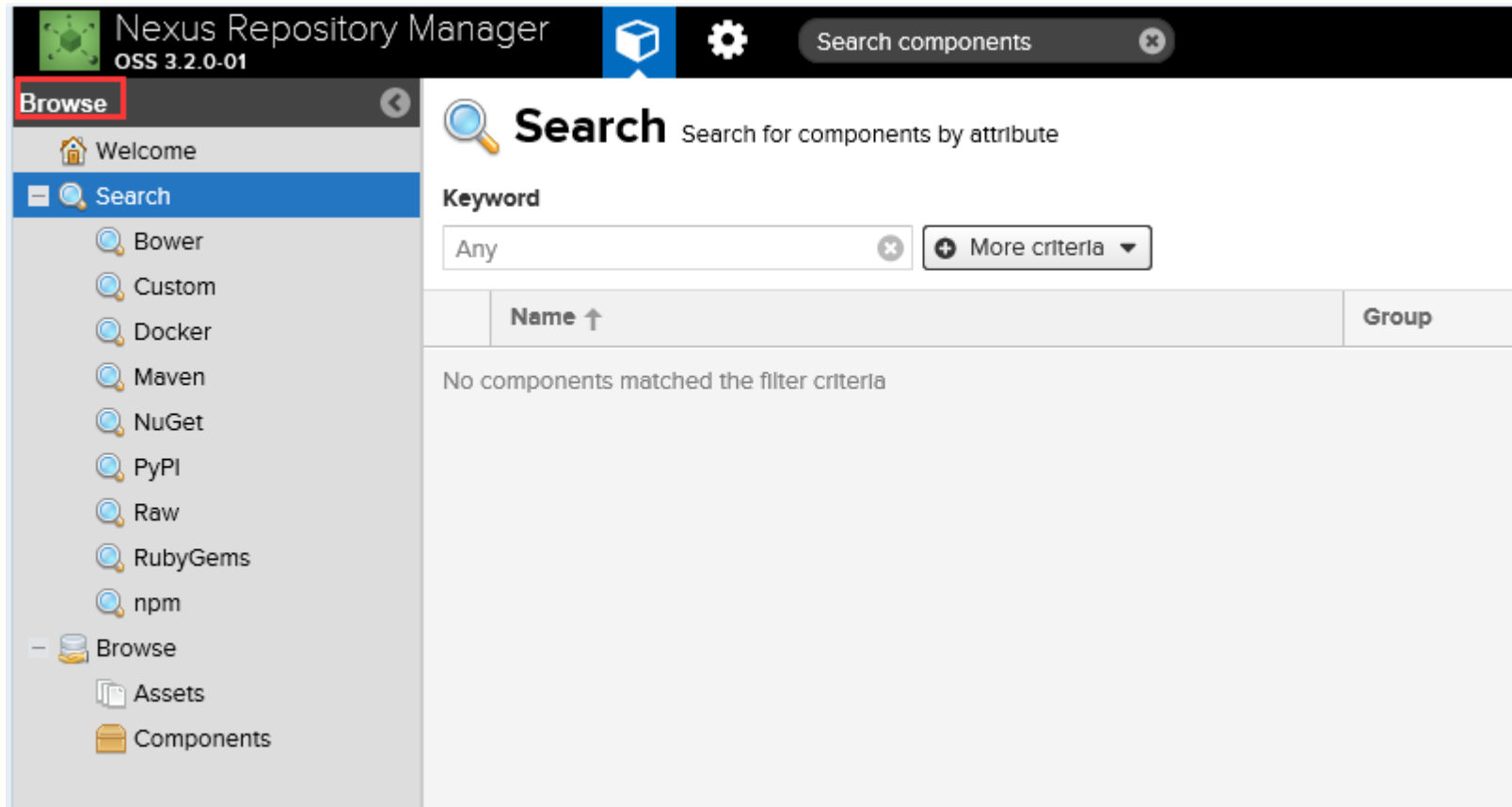
Nexus 仓库分类的概念:

- 1) **Maven** 可直接从宿主仓库下载构件,也可以从代理仓库下载构件,而代理仓库间接的从远程仓库下载并缓存构件
- 2) 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).



Nexus 的 web 界面功能介绍

1. Browse Server Content

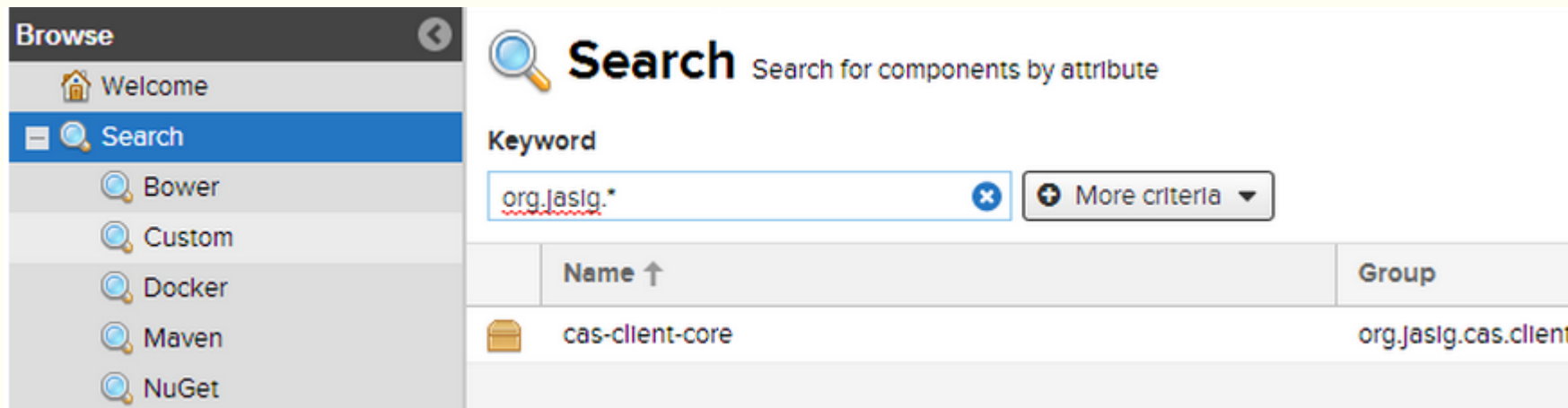


1.1 Search

这个就是类似 Maven 仓库上的搜索功能，就是从私服上查找是否有哪些包。

注意：

1) 在 Search 这级是支持模糊搜索的，如图所示：



2) 如果进入具体的目录，好像不支持模糊搜索，如图所示：

Browse

Welcome

Search

Bower


Custom

Docker

Maven

NuGet

Raw

 **Maven** Search for components by Maven coordinates

Group Id

Artifact Id

Version

org.jasig.cas.*

Any

Any

Extension

Any

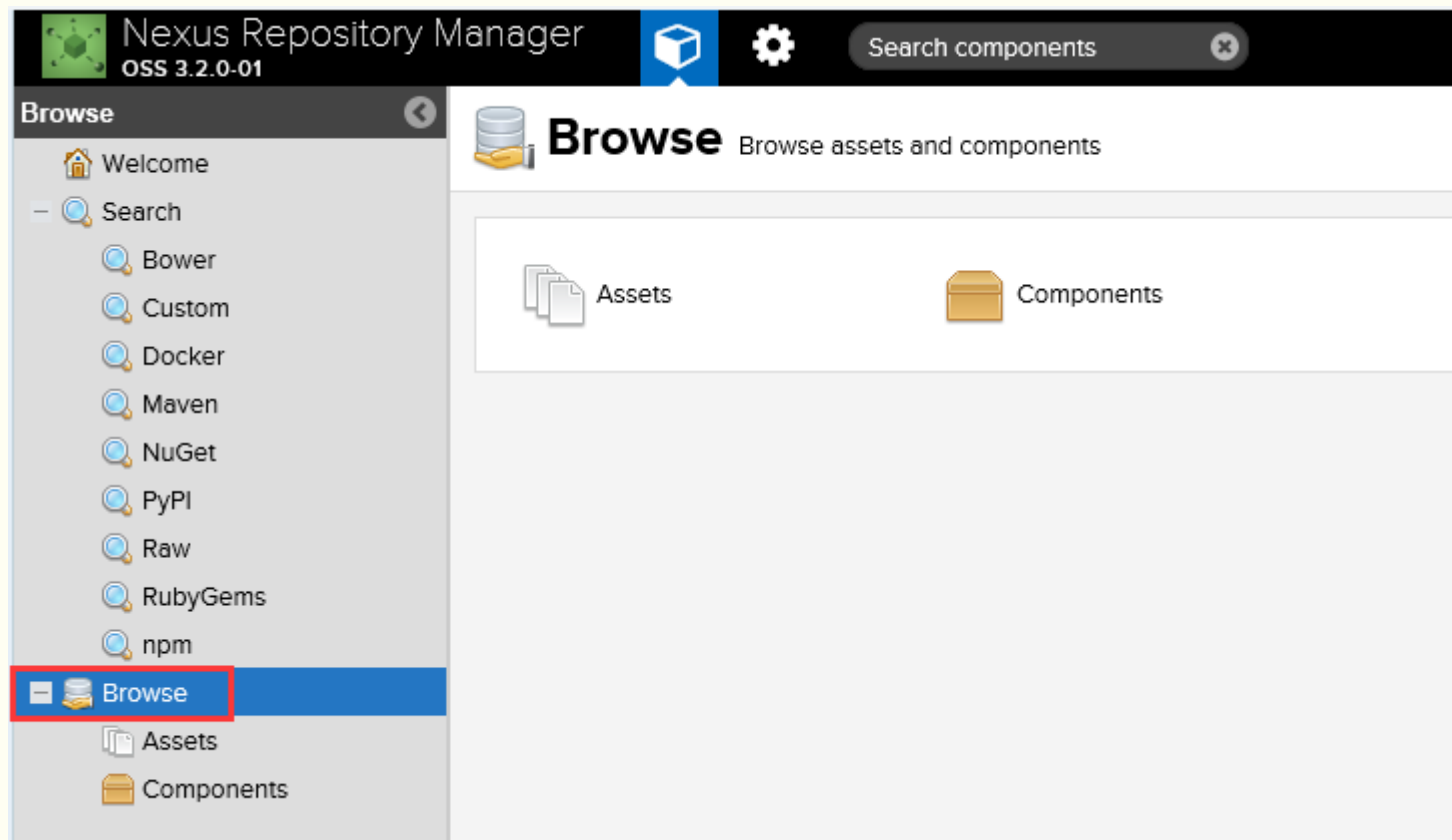
More criteria

Name ↑

Group

No components matched the filter criteria

1.2 Browse



1) Assets

这是能看到所有的资源，包含 Jar，已经对 Jar 的一些描述信息。

2) Components

这里只能看到 Jar 包。

2.Server Adminstration And configuration

看到这个选项的前提是要进行登录的，如上面已经介绍登陆方法，右上角点击“Sign In”的登录按钮，输入 admin/admin123,登录成功之后，即可看到此功能，如图所示：



Administration

Repository

Blob Stores

Repositories

Content Selectors

Security

Privileges

Roles

Users

Anonymous

LDAP

Realms

SSL Certificates

Support

Analytics

Logging

Log Viewer

Metrics

Support ZIP

System Information

System

Bundles

Capabilities

Email Server

HTTP

Licensing



Repository

Repository administration



Blob Stores



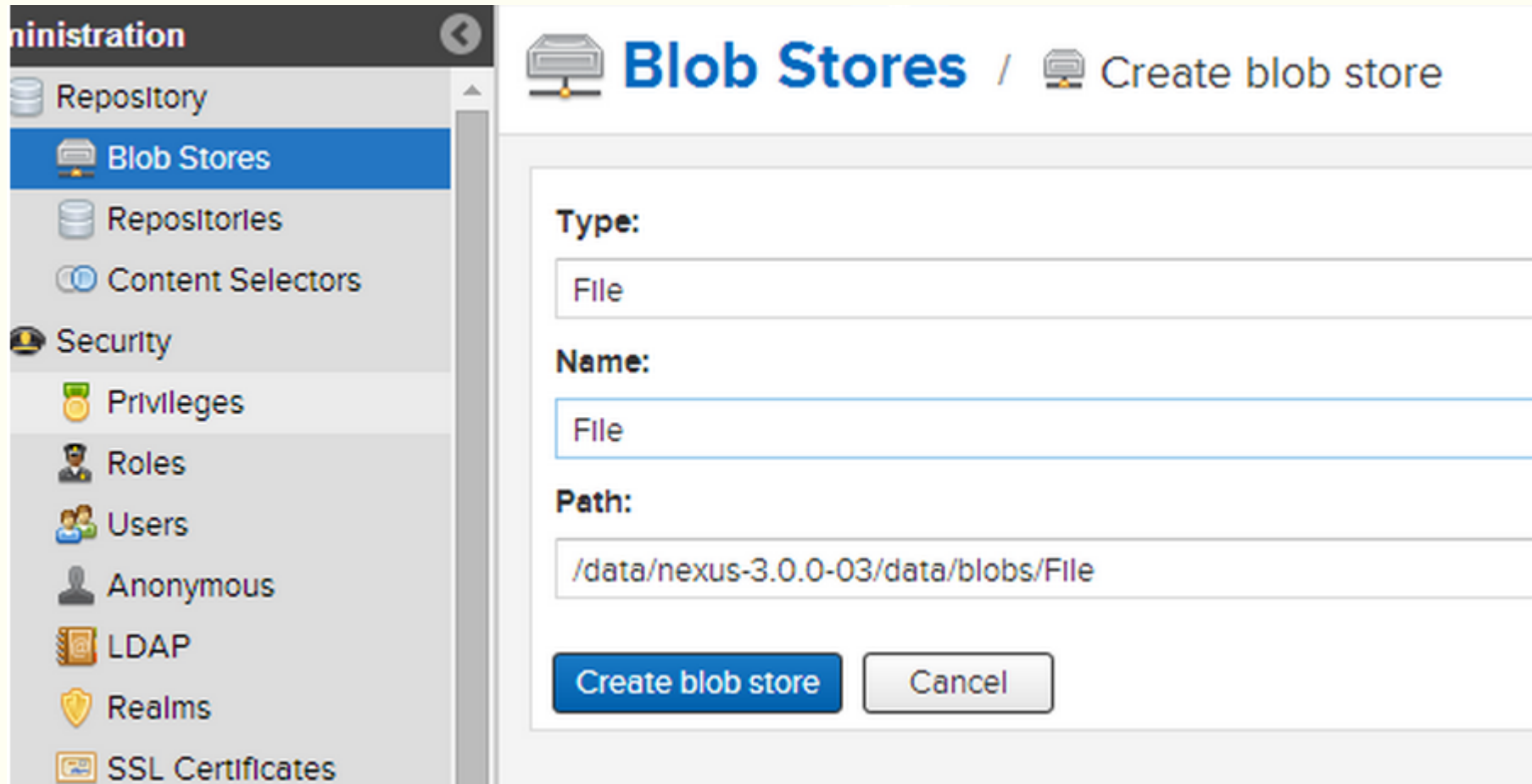
Repositories



Content Selectors

2.1 Blob Stores

文件存储的地方，创建一个目录的话，对应文件系统的一个目录，如图所示：



Administration

- Repository
- Blob Stores**
- Repositories
- Content Selectors
- Security
 - Privileges
 - Roles
 - Users
 - Anonymous
 - LDAP
 - Realms
 - SSL Certificates

Blob Stores / Create blob store

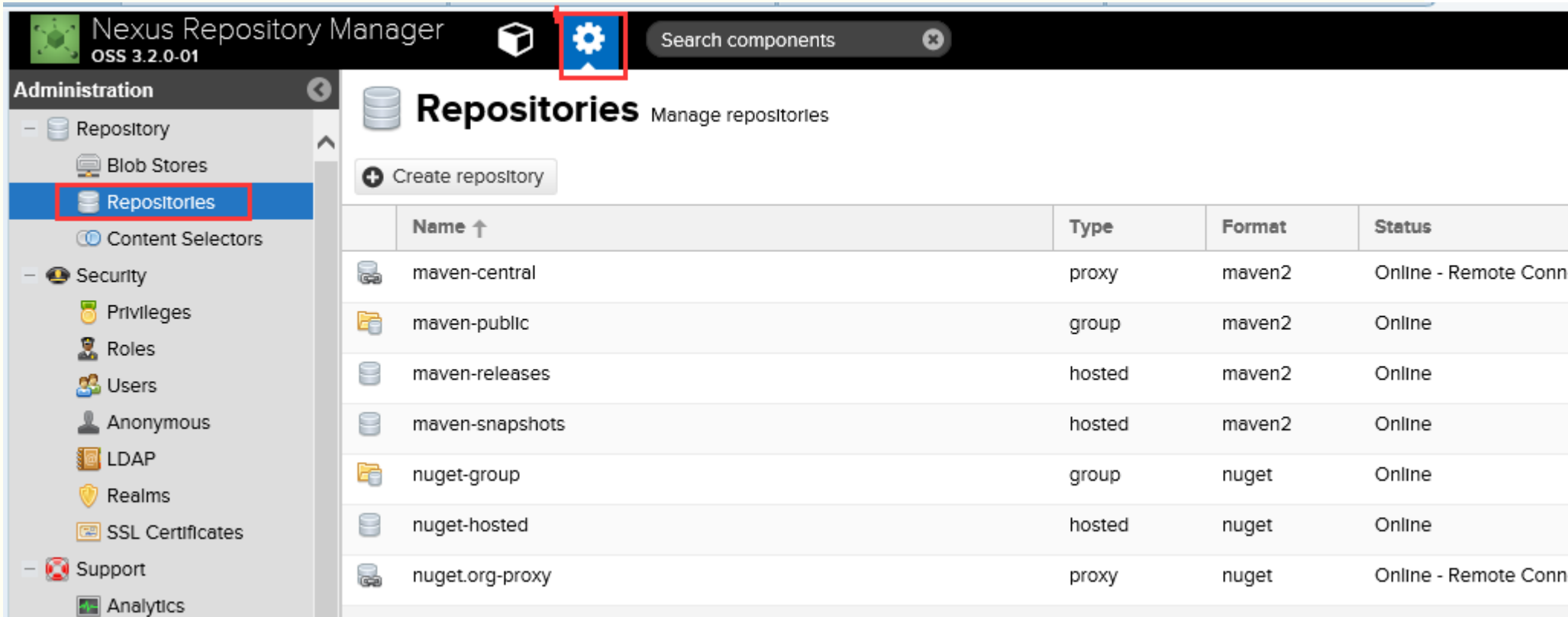
Type:
File

Name:
File

Path:
/data/nexus-3.0.0-03/data/blobs/File

Create blob store Cancel

2.2 Repositories



Nexus Repository Manager OSS 3.2.0-01

Administration

- Repository
- Blob Stores
- Repositories**
- Content Selectors
- Security
 - Privileges
 - Roles
 - Users
 - Anonymous
 - LDAP
 - Realms
 - SSL Certificates
- Support
- Analytics

Repositories Manage repositories

+ Create repository

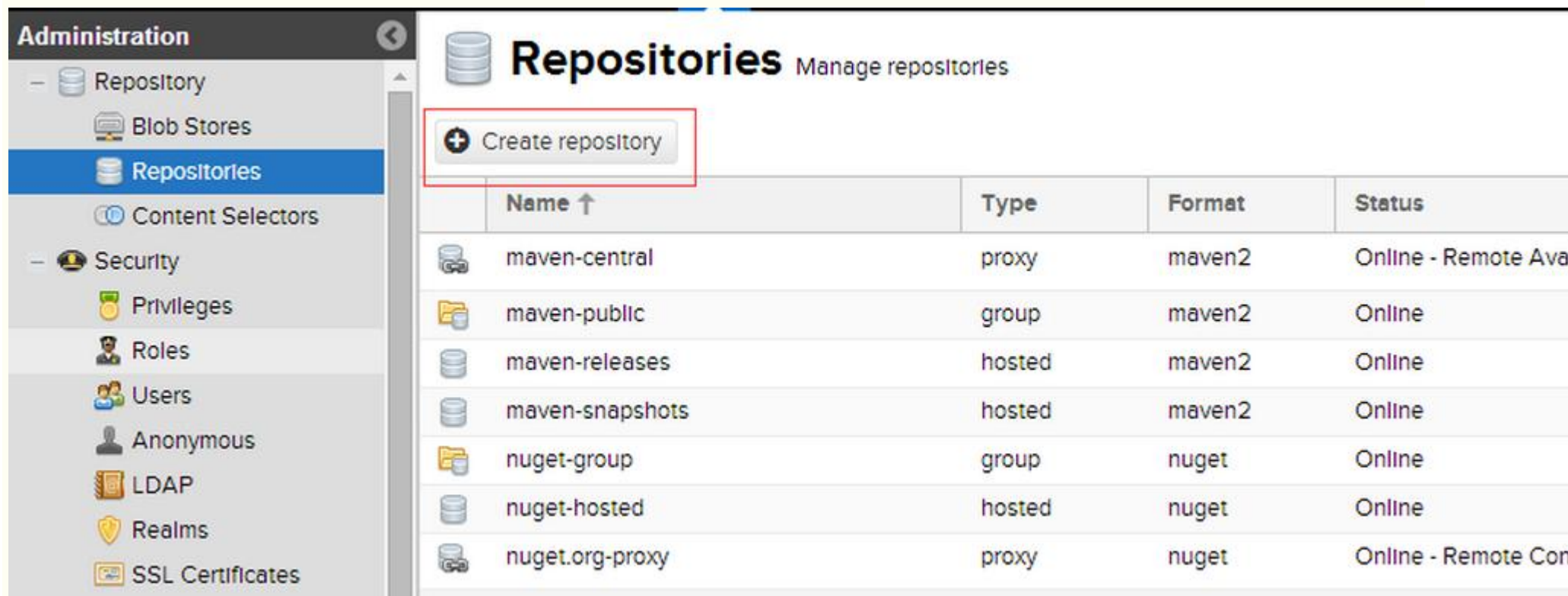
	Name ↑	Type	Format	Status
	maven-central	proxy	maven2	Online - Remote Conn
	maven-public	group	maven2	Online
	maven-releases	hosted	maven2	Online
	maven-snapshots	hosted	maven2	Online
	nuget-group	group	nuget	Online
	nuget-hosted	hosted	nuget	Online
	nuget.org-proxy	proxy	nuget	Online - Remote Conn

1) Proxy








这里就是代理的意思，代理中央 Maven 仓库，当 PC 访问中央库的时候，先通过 Proxy 下载到 Nexus 仓库，然后再从 Nexus 仓库下载到 PC 本地。这样的优势只要其中一个人从中央库下来了，以后大家都是从 Nexus 私服上进行下来，私服一般部署在内网，这样大大节约的宽带。

创建 Proxy 的具体步骤


1-- 点击“Create Repositories”按钮





The screenshot shows the JFrog Artifactory Administration interface. On the left is a sidebar with the 'Administration' menu, where 'Repositories' is selected. The main panel is titled 'Repositories' with a subtitle 'Manage repositories'. A red box highlights the '+ Create repository' button. Below this is a table of existing repositories.


	Name ↑	Type	Format	Status
	maven-central	proxy	maven2	Online - Remote Available
	maven-public	group	maven2	Online
	maven-releases	hosted	maven2	Online
	maven-snapshots	hosted	maven2	Online
	nuget-group	group	nuget	Online
	nuget-hosted	hosted	nuget	Online
	nuget.org-proxy	proxy	nuget	Online - Remote Connected

2-- 选择要创建的类型



Nexus Repository Manager
OSS 3.2.0-01



Search components 

Administration

Repository

Blob Stores

Repositories

Content Selectors

Security

Privileges

Roles

Users

Anonymous

LDAP

Realms

SSL Certificates

Support

Analytics

Logging

Log Viewer



Metrics

Support ZIP


System Information


System


Bundles


**Repositories** /  Select Recipe


Recipe ↑


 bower (group)


 bower (hosted)


 bower (proxy)


 docker (group)


 docker (hosted)


 docker (proxy)


 maven2 (group)


 maven2 (hosted)


 **maven2 (proxy)**


 npm (group)


 npm (hosted)


 npm (proxy)

 nuget (group)

 nuget (hosted)

 nuget (proxy)

 pypl (group)

 pypl (hosted)

3--填写详细信息

Name: 就是为代理起个名字

Remote Storage: 代理的地址，Maven 的地址为: <https://repo1.maven.org/maven2/>

Blob Store: 选择代理下载包的存放路径



微博

最新微博即时通知

- Repository
 - Blob Stores
 - Repositories**
 - Content Selectors
- Security
 - Privileges
 - Roles
 - Users
 - Anonymous
 - LDAP
 - Realms
 - SSL Certificates
- Support
 - Analytics
 - Logging
 - Log Viewer
 - Metrics
 - Support ZIP
 - System Information



Repositories



Select Recipe



Create Repository:

Name:

A unique Identifier for this repository

This field is required

Online:

☒ If checked, the repository accepts Incoming requests

Maven 2

Version policy:

What type of artifacts does this repository store?

Release

Layout policy:

Validate that all paths are maven artifact or metadata paths

Strict

Proxy

Remote storage:

Location of the remote repository being proxied

Enter a URL

2) Hosted

Hosted 是宿主机的意思，就是怎么把第三方的 Jar 放到私服上。

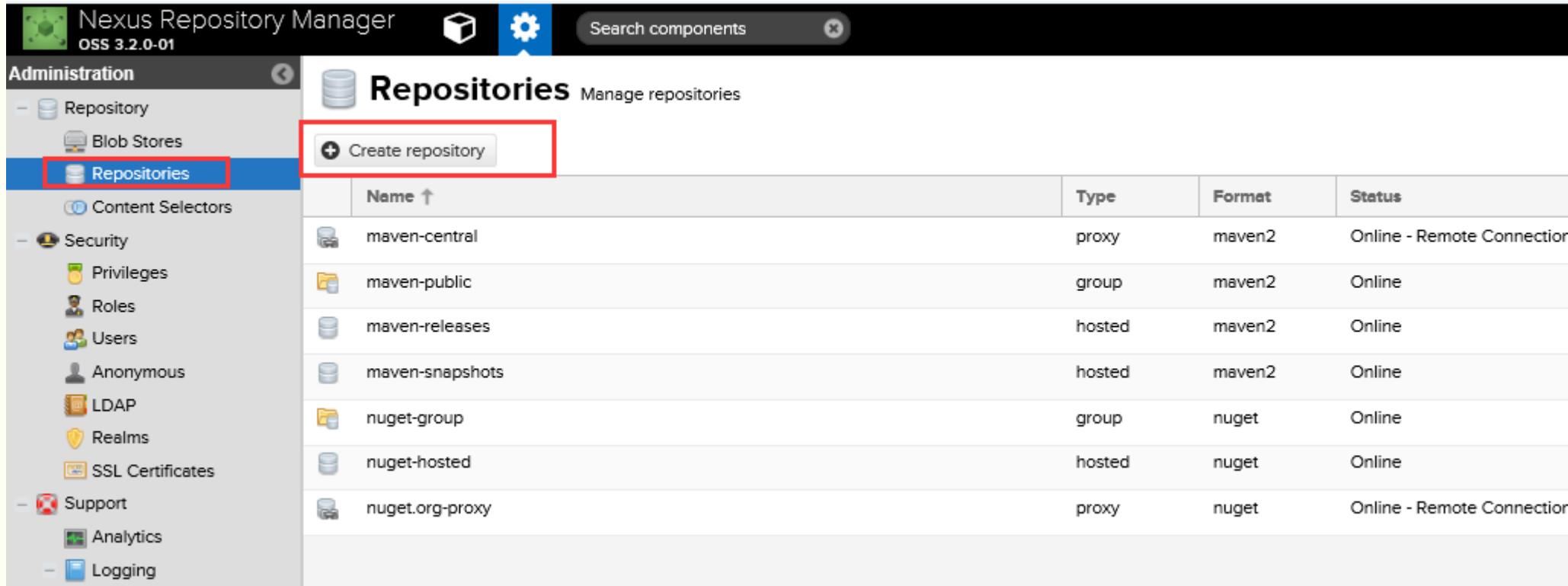
Hosted 有三种方式，Releases、SNAPSHOT、Mixed

Releases: 一般是已经发布的 Jar 包

Snapshot: 未发布的版本

Mixed: 混合的

Hosted 的创建和 Proxy 是一致的，具体步骤和上面基本一致。如下：



Nexus Repository Manager OSS 3.2.0-01


Administration

- Repository
- Blob Stores
- Repositories**
- Content Selectors
- Security
 - Privileges
 - Roles
 - Users
 - Anonymous
 - LDAP
 - Realms
 - SSL Certificates
- Support
 - Analytics
 - Logging

Repositories Manage repositories



+ Create repository

Name ↑	Type	Format	Status
maven-central	proxy	maven2	Online - Remote Connection
maven-public	group	maven2	Online
maven-releases	hosted	maven2	Online
maven-snapshots	hosted	maven2	Online
nuget-group	group	nuget	Online
nuget-hosted	hosted	nuget	Online
nuget.org-proxy	proxy	nuget	Online - Remote Connection



Nexus Repository Manager

OSS 3.2.0-01



Search components

Administration

Repository

Blob Stores

Repositories

Content Selectors

Security

Privileges

Roles

Users

Anonymous

LDAP

Realms

SSL Certificates

Support

Analytics

Logging

Log Viewer

Metrics

Support ZIP

System Information

System

Bundles

Capabilities

Email Server

HTTP

Repositories

Select Recipe

Recipe ↑

bower (group)

bower (hosted)

bower (proxy)

docker (group)

docker (hosted)

docker (proxy)

maven2 (group)

maven2 (hosted)

maven2 (proxy)

npm (group)

npm (hosted)

npm (proxy)

nuget (group)

nuget (hosted)

nuget (proxy)


pypi (group)



pypi (hosted)


pypi (proxy)

raw (group)

raw (hosted)

Nexus Repository Manager
OSS 3.2.0-01



Search components 

Administration

Repository

Blob Stores

Repositories

Content Selectors

Security

Privileges

Roles

Users

Anonymous

LDAP

Realms

SSL Certificates

Support

Analytics

Logging

Log Viewer

Metrics

Support ZIP

System Information

System

Bundles

Capabilities

Email Server




HTTP

Licensing

Recent Connections


Nodes

Tasks

Repositories /  Select Recipe /  Create Repository: maven2 (hoste

Name:

A unique Identifier for this repository

 This field is required

Online:

☒ If checked, the repository accepts Incoming requests

Maven 2

Version policy:

What type of artifacts does this repository store?

Release

Layout policy:

Validate that all paths are maven artifact or metadata paths

Strict

Storage

Blob store:

Blob store used to store asset contents

default

Strict Content Type Validation:

☒ Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format

Hosted

Deployment policy:

Controls if deployments of and updates to artifacts are allowed

Disable redeploy

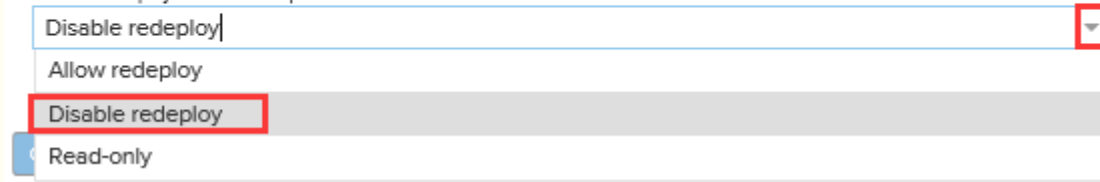
注意事项:

Deployment Policy: 需要把策略改成“Allow redeploy”。

Hosted

Deployment policy:

Controls if deployments of and updates to artifacts are allowed



Disable redeploy

Allow redeploy

Disable redeploy

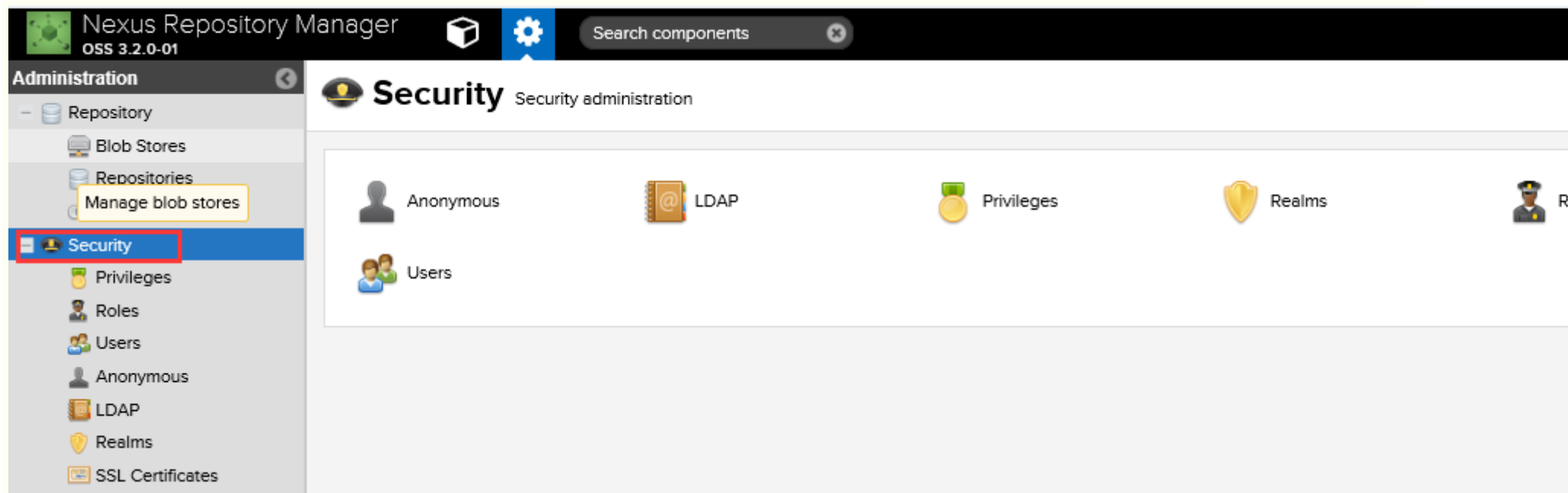
Read-only

3) Group

能把两个仓库合成一个仓库来使用，目前没使用过，所以没做详细的研究。

2.3 Security

这里主要是用户、角色、权限的配置（上面已经提到了在这里添加用户和角色等）



2.4 Support

包含日志及数据分析。

收藏夹
快速打开收藏夹

Repository Manager

Search components

Repository

Repository administration

Repositories

Content Selectors

Security

Privileges

Roles

Users

Anonymous

LDAP

Realms

SSL Certificates

Support

Analytics

Logging

Log Viewer

Metrics

Support ZIP

System Information

System

Bundles

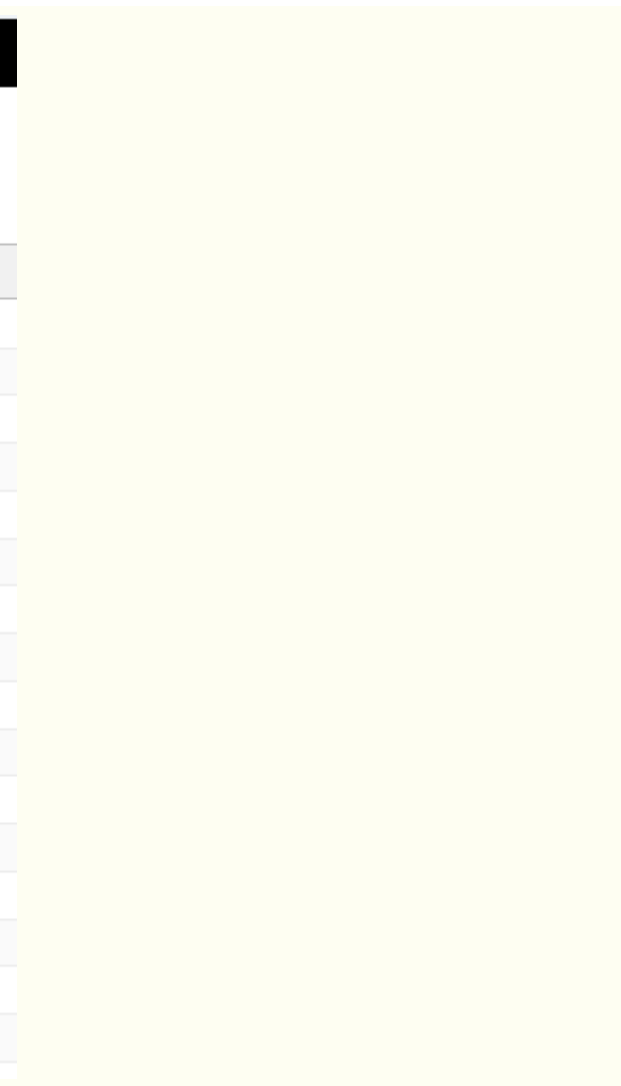
Capabilities

Logging

Control logging verbosity levels

Create loggerDelete loggerReset to default levels

Name ↑
ROOT
com.google.inject.internal.util.Stopwatch
com.orienttechnologies
com.softwarementors.extjs.djn
com.sonatype.nexus.ahc
com.sonatype.nexus.ahc.internal.scan
com.sonatype.nexus.analytics
com.sonatype.nexus.migration
com.sonatype.nexus.plugins.healthcheck
com.sonatype.nexus.ssl
eu.medsea.mimeutil.MimeUtil2
javax.management
org.apache.aries
org.apache.commons
org.apache.felix
org.apache.http



2.5 System

主要是邮件服务器，调度的设置地方

这部分主要讲怎么和 **Maven** 做集成,集成的方式主要分以下种情况：代理中央仓库、**Snapshot** 包的管理、**Release** 包的管理、第三方 **Jar** 上传到 **Nexus** 上。

代理中央仓库

只要在 **PMO** 文件中配置私服的地址（比如 <http://192.168.1.14:8081>）即可，配置如下：

```
1 <repositories>
2     <repository>
3         <id>maven-central</id>
4         <name>maven-central</name>
5         <url>http://192.168.1.14:8081/repository/maven-central/</url>
6         <snapshots>
7             <enabled>true</enabled>
8         </snapshots>
9         <releases>
10             <enabled>true</enabled>
11         </releases>
12     </repository>
13 </repositories>
```

Snapshot 包的管理

1) 修改 **Maven** 的 **settings.xml** 文件，加入认证机制

1	<servers>
2	<server>
3	<id>nexus</id>
4	<username>admin</username>
5	<password>admin123</password>
6	</server>

2) 修改工程的 Pom 文件

1	<distributionManagement>
2	<snapshotRepository>
3	<id>nexus</id>
4	<name>Nexus Snapshot</name>
5	<url>http://192.168.1.14:8081/repository/maven-snapshots/</url>
6	</snapshotRepository>
7	<site>
8	<id>nexus</id>
9	<name>Nexus Sites</name>
10	<url>dav:http://192.168.1.14:8081/repository/maven-snapshots/</url>
11	</site>
12	</distributionManagement>

注意事项:

```
<snapshotRepository>
  <id>nexus</id>
  <name>Nexus Snapshot</name>
  <url>http://192.168.1.14:8081/repository/maven-snapshots/</url>
</snapshotRepository>
```

上面修改的 Pom 文件如截图中的名字要跟/usr/local/maven/conf/settings.xml 文件中的名字一定要对应上。

3) 上传到 Nexus 上

1--项目编译成的 jar 是 Snapshot(POM 文件的头部)

1	<groupId>com.woasis</groupId>
2	<artifactId>test-nexus</artifactId>
3	<version>1.0.0- SHAPSHOT </version>
4	<packaging>jar</packaging>

2--使用 mvn deploy 命令运行即可（运行结果在此略过）

3--因为 Snapshot 是快照版本，默认他每次会把 Jar 加一个时间戳，做为历史备份版本。

Releases 包的管理

1) 与 Snapshot 大同小异，只是上传到私服上的 Jar 包不会自动带时间戳

2) 与 Snapshot 配置不同的地方，就是工程的 PMO 文件，加入 repository 配置

1	<distributionManagement>
2	<repository>
3	<id>nexus</id>

4	<code><name>Nexus Snapshot</name></code>
5	<code><url>http://192.168.1.14:8081/repository/maven-releases/</url></code>
6	<code></repository></code>

3) 打包的时候需要把 **Snapshot** 去掉

1	<code><groupId>com.woasis</groupId></code>
2	<code><artifactId>test-nexus</artifactId></code>
3	<code><version>1.0.0</version></code>
4	<code><packaging>jar</packaging></code>

第三方 **Jar** 上传到 **Nexus**

```
[root@master-node src]# mvn deploy:deploy-file -DgroupId=org.jasig.cas.client -DartifactId=cas-client-core -Dversion=3.1.3 -Dpackaging
```

注意事项:

-DrepositoryId=nexus 对应的就是 Maven 中 settings.xml 的认证配的名字。

*****当你发现自己的才华撑不起野心时，就请安静下来学习吧*****