# Maven 私有库和本地库的安装与配置

# Sonatype Nexus + Maven

前提：已安装 JDK7 并配置好了环境变量

## 一 安装私服与配置

1、下载最新版 Nexus（本教程使用的是：nexus-2.11.2-03-bundle.tar.gz）,下载地址： http://www.sonatype.org/nexus/go/

# wget https://sonatype-download.global.ssl.fastly.net/nexus/oss/nexus-2.11.203-bundle.tar.gz

1. 解压

# mkdir nexus

# tar -zxvf nexus-2.11.2-03-bundle.tar.gz -C nexus

# cd nexus

# ls

nexus-2.11.2-03 sonatype-wor

1. 编辑 Nexus 的 nexus.properties 文件,配置端口和 work 目录信息（保留默认）

# cd nexus-2.11.2-03

# ls

bin conf lib LICENSE.txt logs nexus NOTICE.txt tmp

查看目录结构，jetty 运行

# cd conf

# vi nexus.properties

# Jetty section

application-port=8081

application-host=0.0.0.0

nexus-webapp=${bundleBasedir}/nexus nexus-webapp-context-path=/nexus

# Nexus section

nexus-work=${bundleBasedir}/../sonatype-work/nexus runtime=${bundleBasedir}/nexus/WEB-INF

1. 编辑 nexus 脚本, 配置 RUN\_AS\_USER 参数

# vi /root/nexus/nexus-2.11.2-03/bin/nexus

#RUN\_AS\_USER=

改为：

RUN\_AS\_USER=root

1. 防火墙中打开 8081 端口

# vi /etc/sysconfig/iptables

添加：

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

保存后重启防火墙

# service iptables restart

1. 启动 nexus

# /root/nexus/nexus-2.11.2-03/bin/nexus start

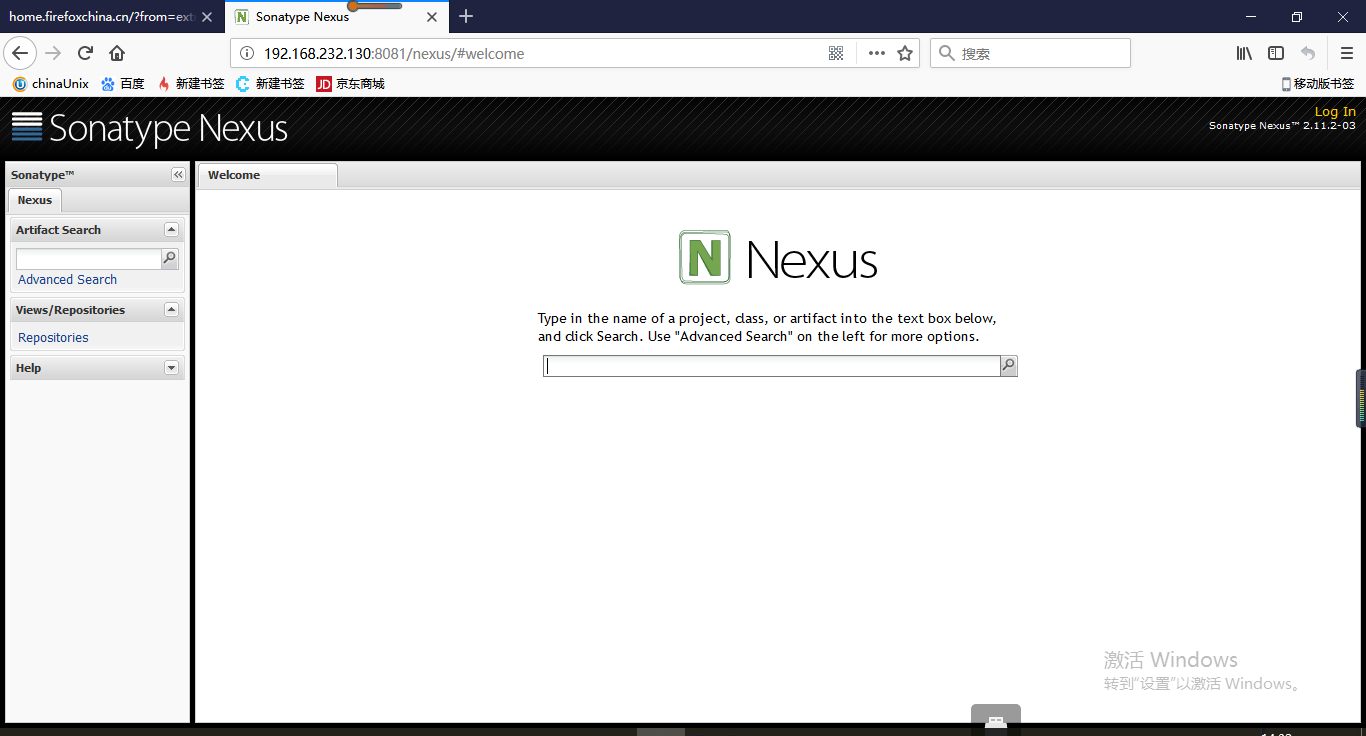
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WARNING - NOT RECOMMENDED TO RUN AS ROOT

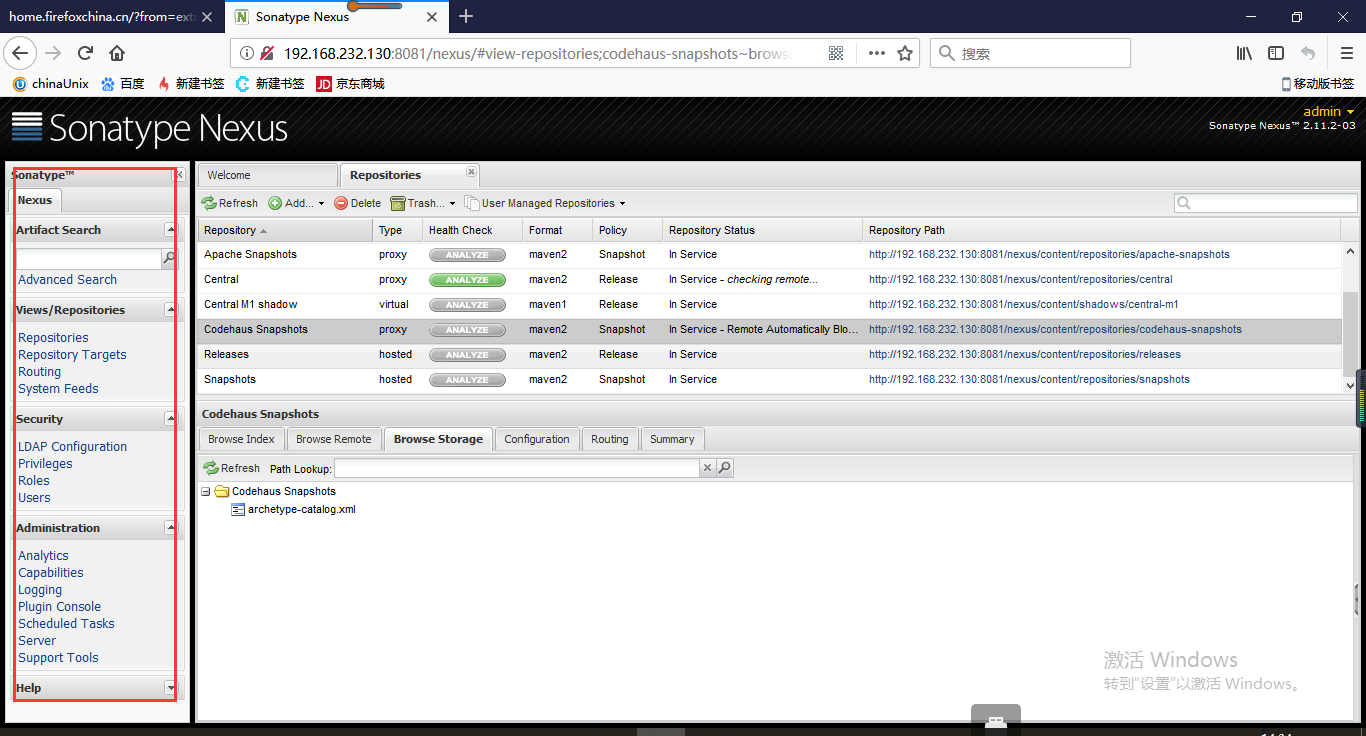
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Starting Nexus OSS... Started Nexus OSS.

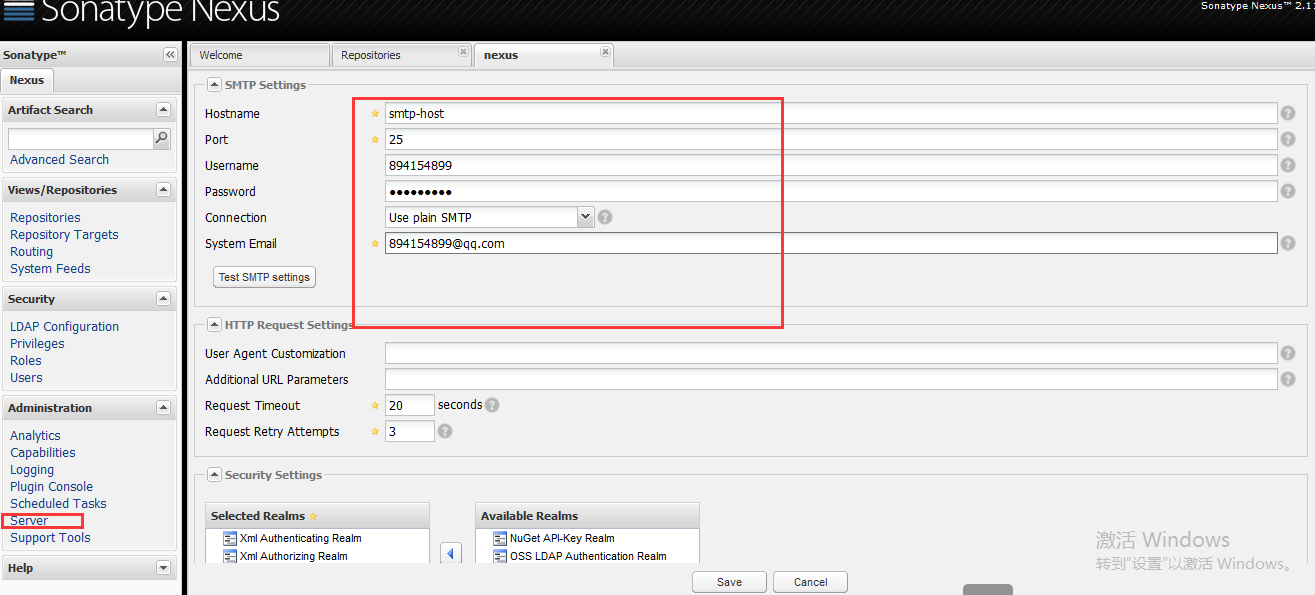
http://192.168.232.130:8081/nexus/#welcome输入如下网址便可以访问私服了。



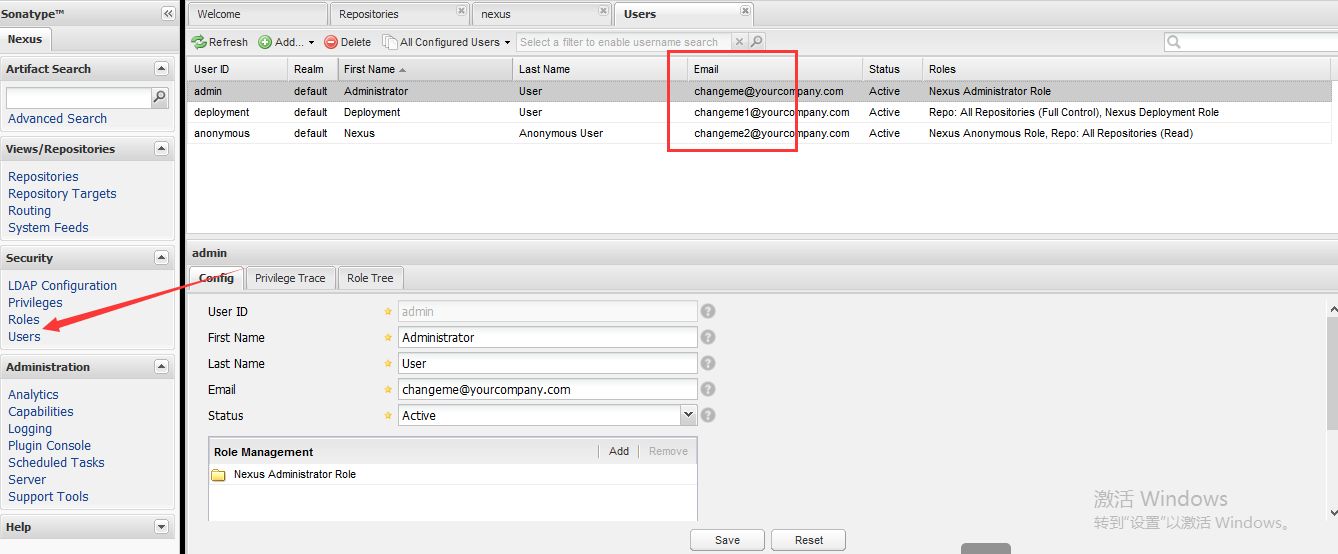
1. 登录，默认用户名 admin,默认密码 admin123:



nexus已经完成，接下来就可以进行nexus的配置了。

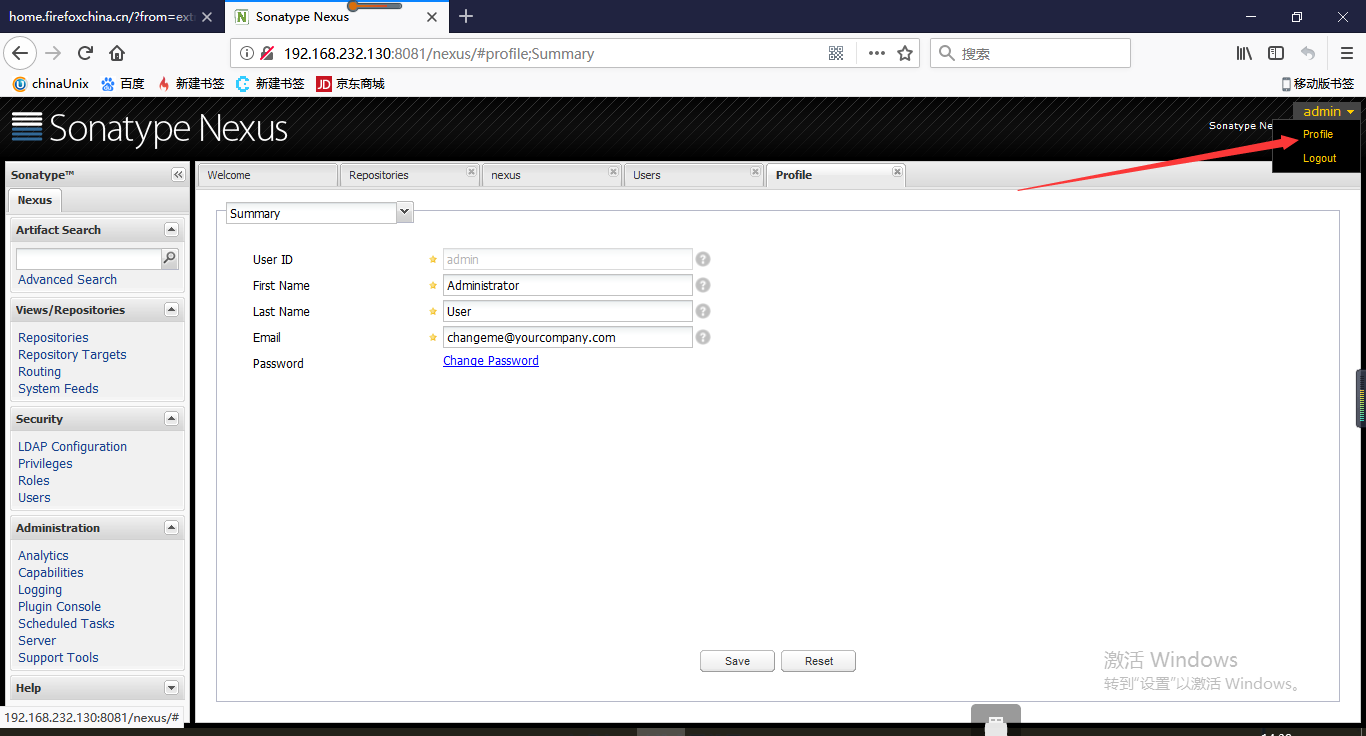


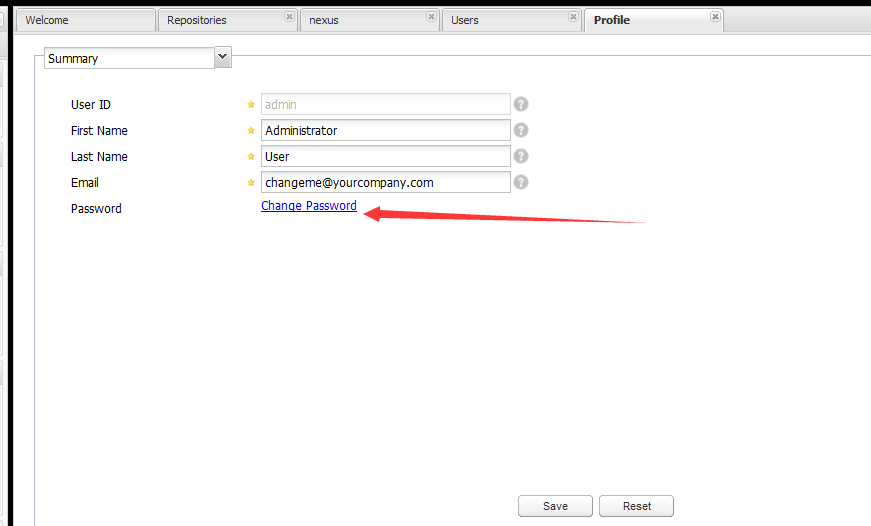
配置邮箱密码。并做保存，方便后面找回邮箱密码。



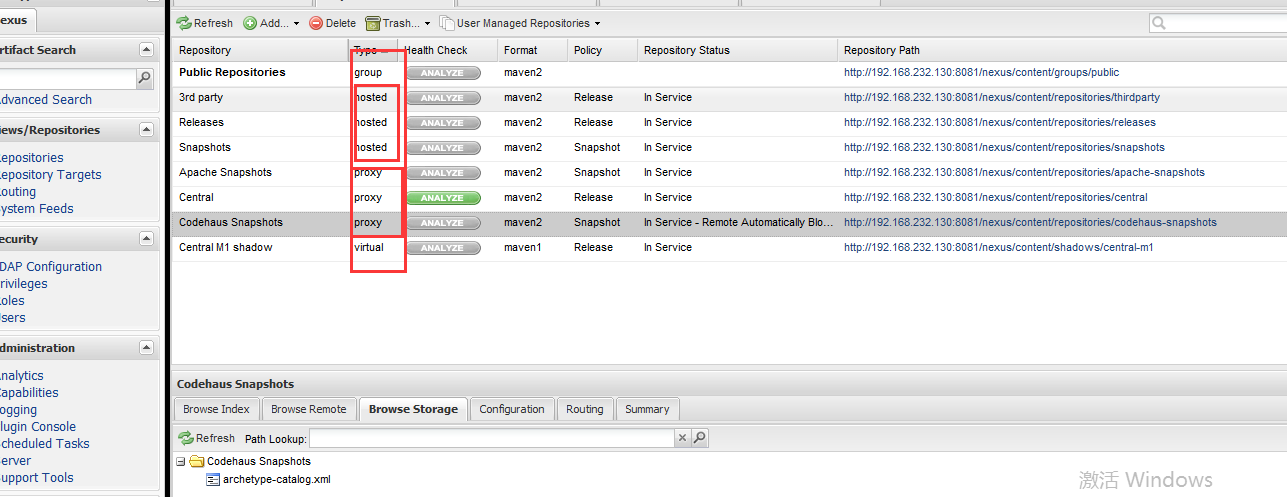
配置用户的邮箱地址，方便修改变更。

用户修改密码





仓库



group 仓库组：Nexus 通过仓库组的概念统一管理多个仓库，这样我们在项目中直

接请求仓库组即可请求到仓库组管理的多个仓库；

hosted 宿主仓库：主要用于发布内部项目构件或第三方的项目构件（如购买商业的构件） 以及无法从公共仓库获取的构件（如 oracle 的 JDBC 驱动）

proxy 代理仓库：代理公共的远程仓库；

virtual 虚拟仓库：用于适配 Maven 1； 一般用到的仓库种类是 hosted、proxy

Hosted 仓库常用类型说明：

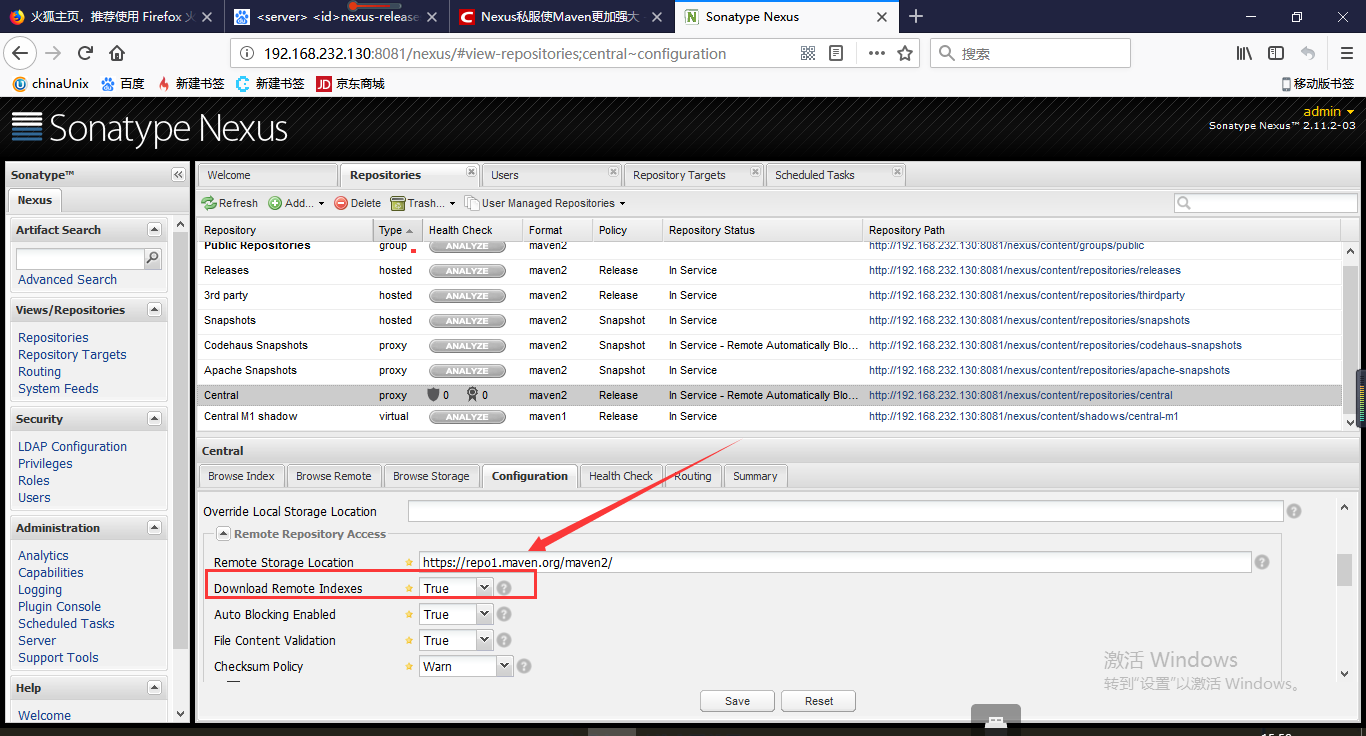
releases 内部的模块中 release 模块的发布仓库

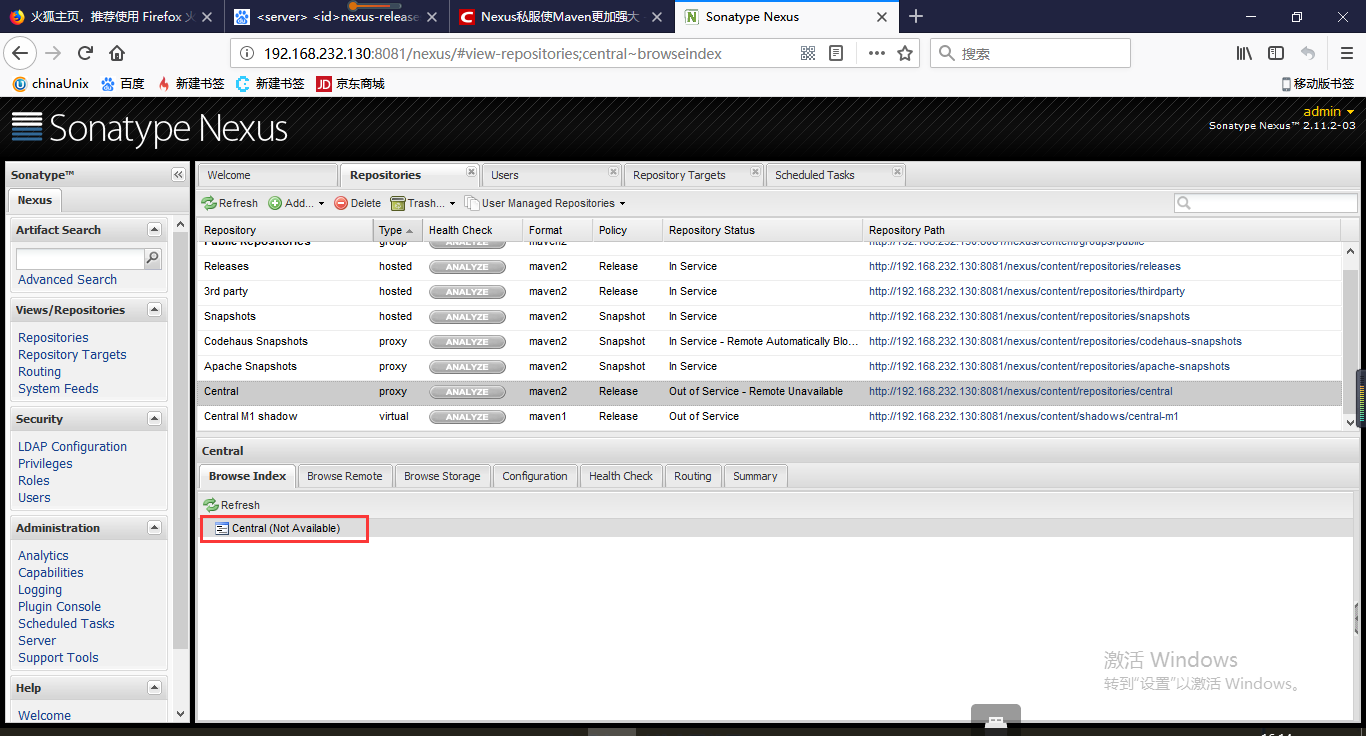
snapshots 发布内部的 SNAPSHOT 模块的仓库

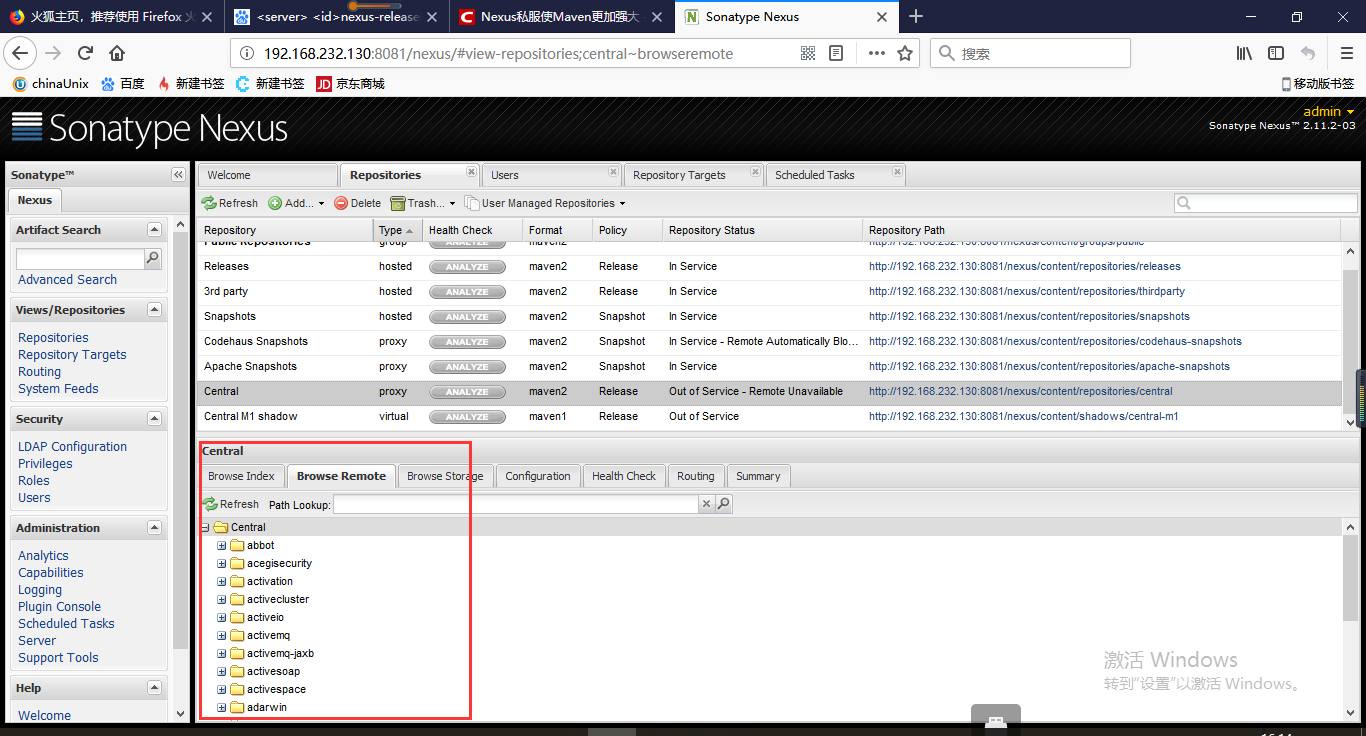
3rd party 第三方依赖的仓库，这个数据通常是由内部人员自行下载之后发布上去

如果构建的 Maven 项目本地仓库没有对应的依赖包，那么就会去 Nexus 私服去下载， 如果Nexus私服也没有此依赖包，就回去远程中央仓库下载依赖，这些中央仓库就是proxy。 Nexus 私服下载成功后再下载至本地 Maven 库供项目引用。

1. 设置 proxy 代理仓库(Apache Snapshots/Central/Codehaus Snapshots)准许远程下载,也就是依赖的一些spring的包等等的， 如：







## 二 安装本地与配置

本地私有文件POM.xml文件配置

<distributionManagement>

<repository>

<id>nexus-releases</id>

<name>Nexus Release Repository</name>

<url>http://192.168.232.130:8081/nexus/content/repositories/releases/</url>

</repository>

<snapshotRepository>

<id>nexus-snapshots</id>

<name>Nexus Snapshot Repository</name>

<url>http://192.168.232.130:8081/nexus/content/repositories/snapshots/</url>

</snapshotRepository>

</distributionManagement>

本地私有文件setting.xml文件的配置

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<settings xmlns=*"http://maven.apache.org/SETTINGS/1.0.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd"*>

<localRepository>E:/apache-maven-3.1.0/.m2/repository</localRepository>

<interactiveMode>true</interactiveMode>

<offline>false</offline>

<pluginGroups>

<pluginGroup>org.mortbay.jetty</pluginGroup>

<pluginGroup>org.jenkins-ci.tools</pluginGroup>

</pluginGroups>

<!--配置权限,使用默认用户 -->

<servers>

<server>

<id>nexus-releases</id>

<username>deployment</username>

<password>deployment123</password>

</server>

<server>

<id>nexus-snapshots</id>

<username>deployment</username>

<password>deployment123</password>

</server>

</servers>

<profiles>

<profile>

<id>edu</id>

<activation>

<activeByDefault>false</activeByDefault>

<jdk>1.6</jdk>

</activation>

<repositories>

<!-- 私有库地址 -->

<repository>

<id>nexus</id>

<url>http://192.168.232.130:8081/nexus/content/groups/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://192.168.232.130:8081/nexus/content/groups/public/</url>

<releases>

<enabled>true</enabled>

</releases>

<snapshots>

<enabled>true</enabled>

</snapshots>

</pluginRepository>

</pluginRepositories>

</profile>

</profiles>

<!--激活profile -->

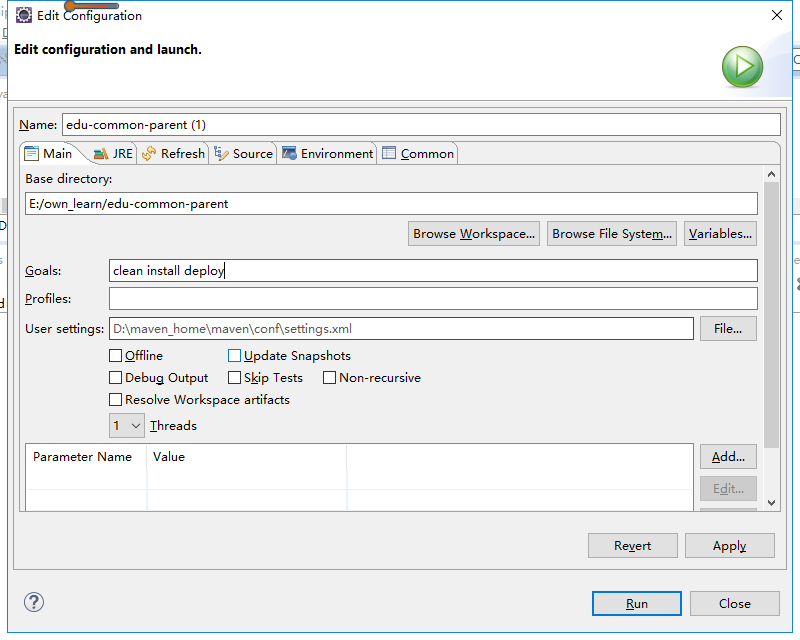
<activeProfiles>

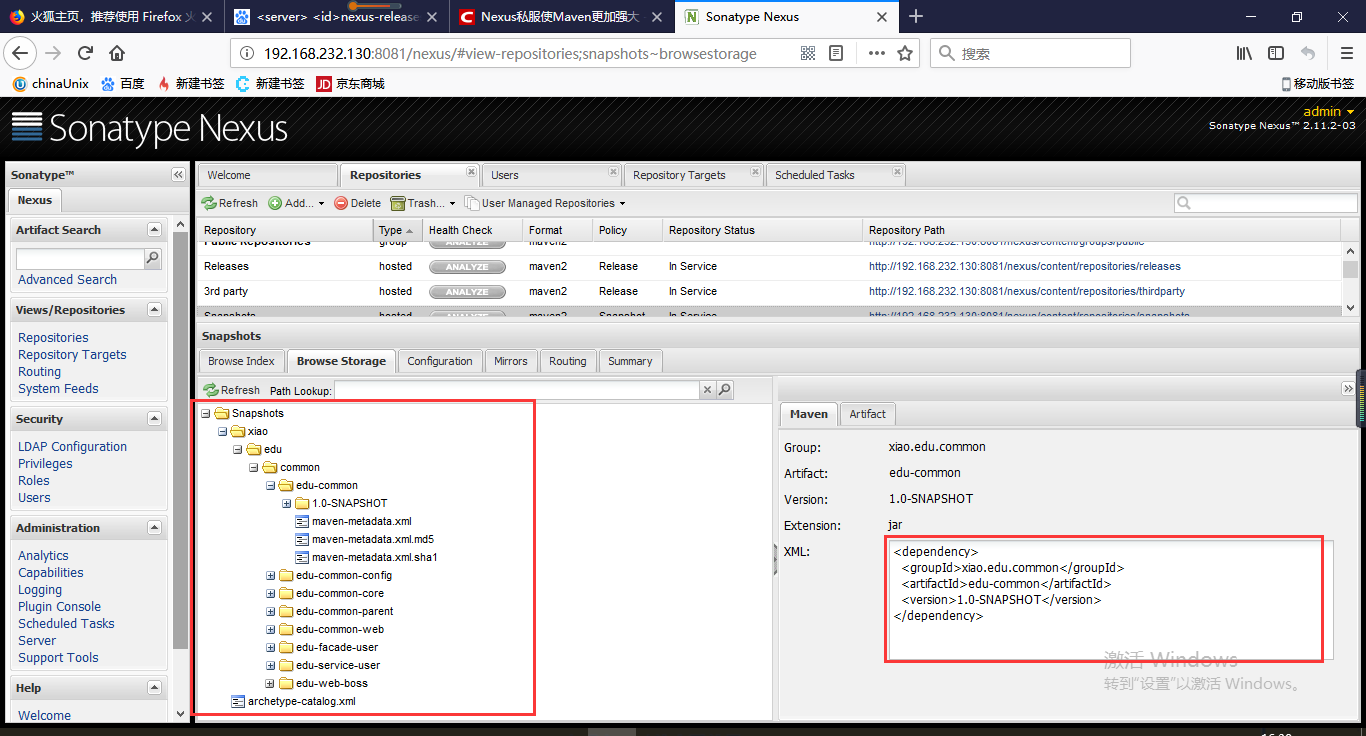
<activeProfile>edu</activeProfile>

</activeProfiles>

</settings>

部署构建





如上图所示，通过上面的部署构建的命令，就可以把我们本地私有的构建部署到私有库，这样所有的开发团队就可以共同下载引用构建进行协作开发了。