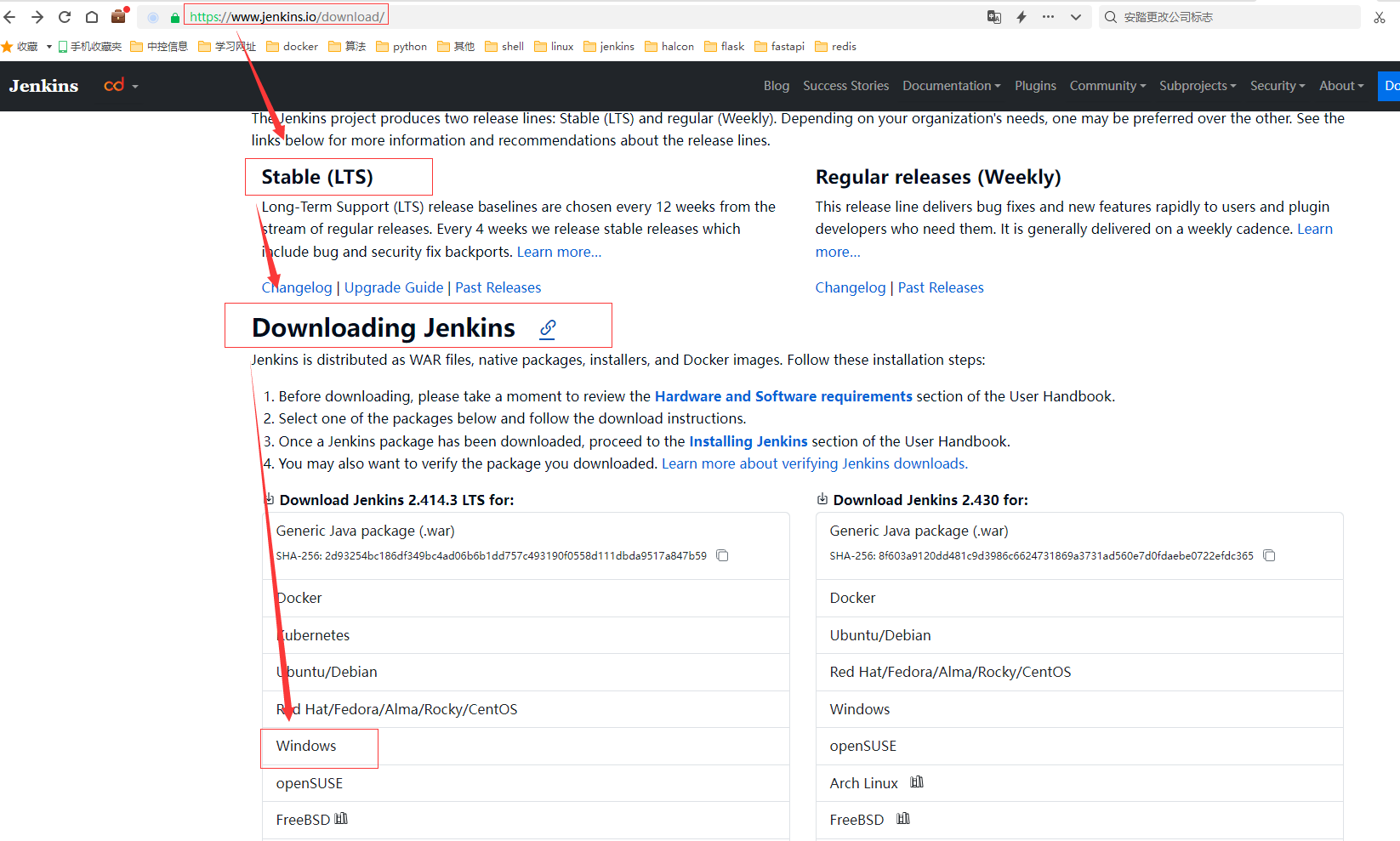
# 一、下载软件：

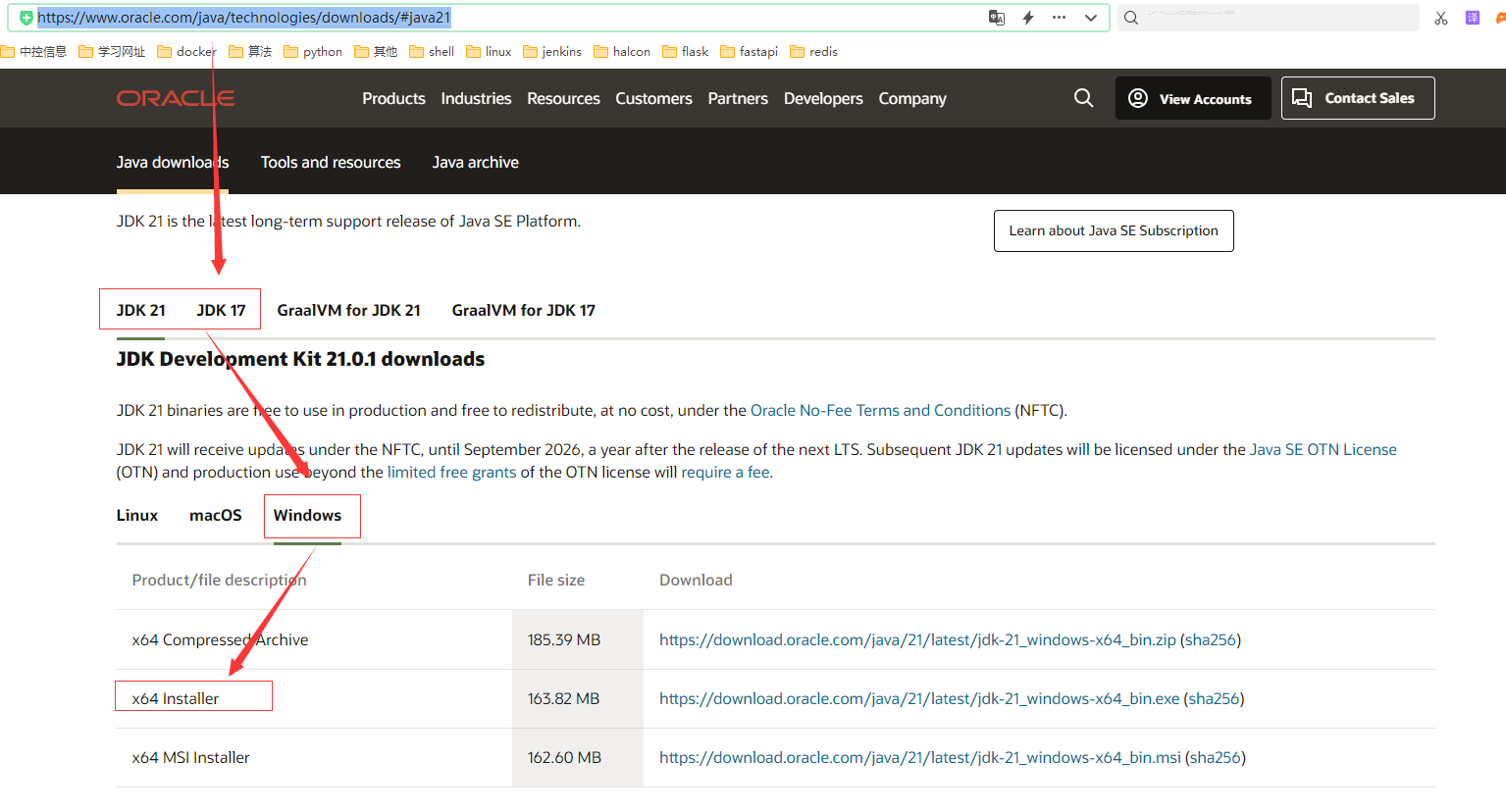
1）jenkins软件：

https://www.jenkins.io/download/



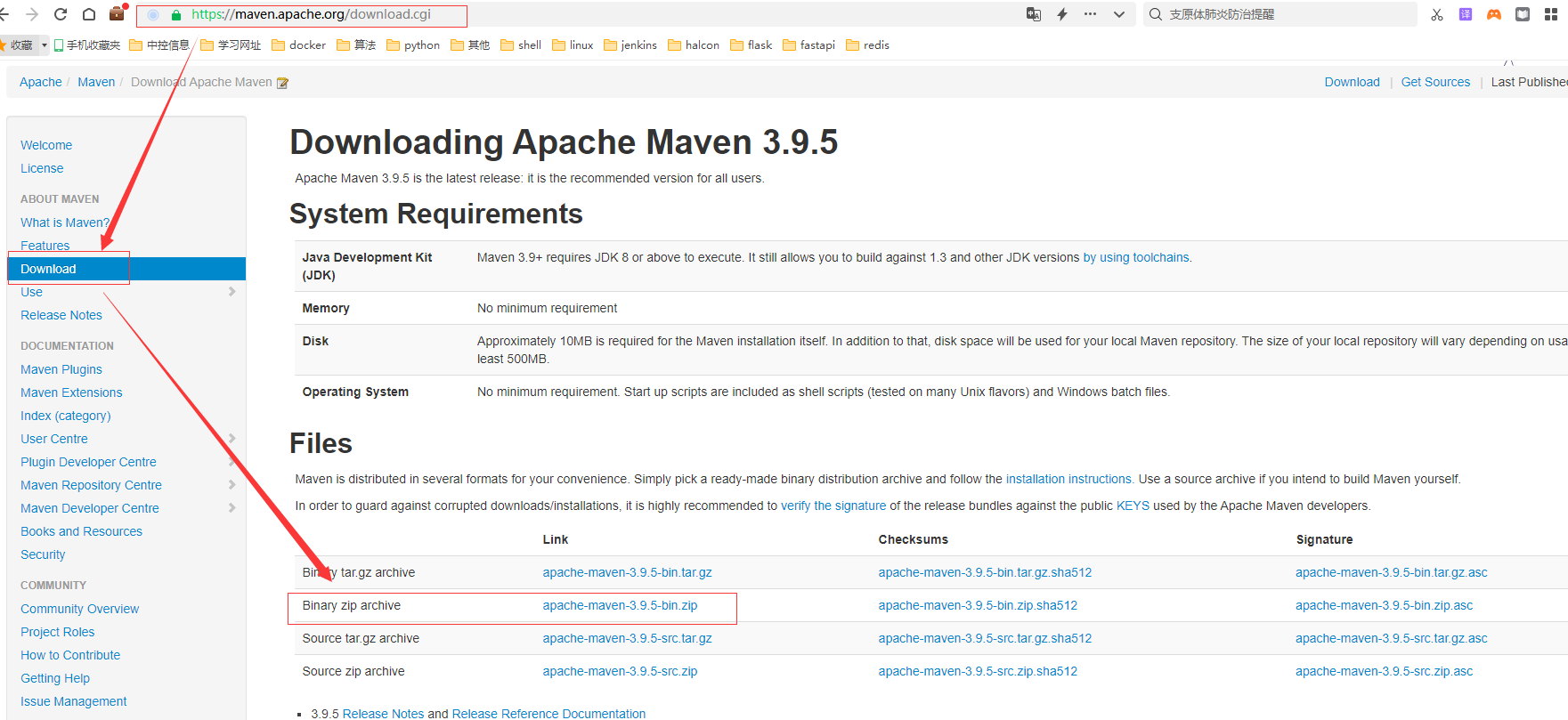
2）jdk软件（安装过程省略）：

<https://www.oracle.com/java/technologies/downloads/#java21>



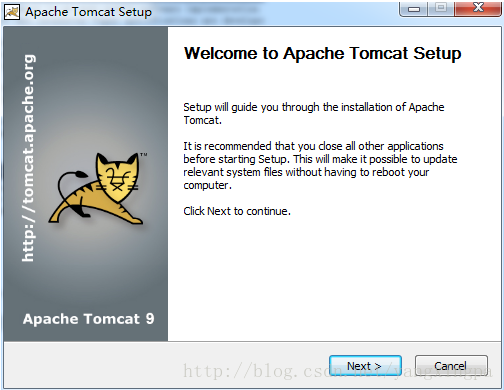
3）maven软件（安装过程省略）：

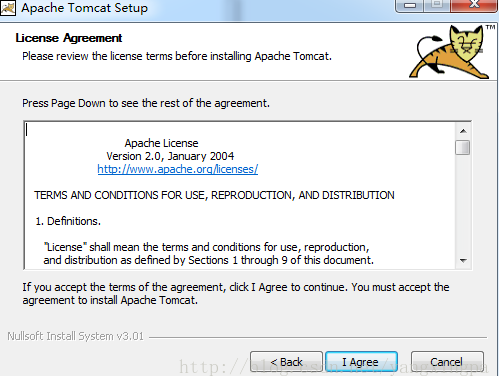
<https://maven.apache.org/download.cgi>

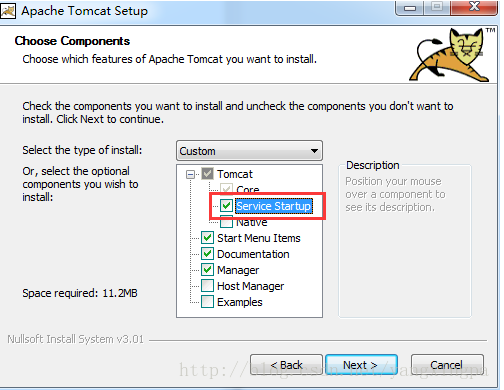


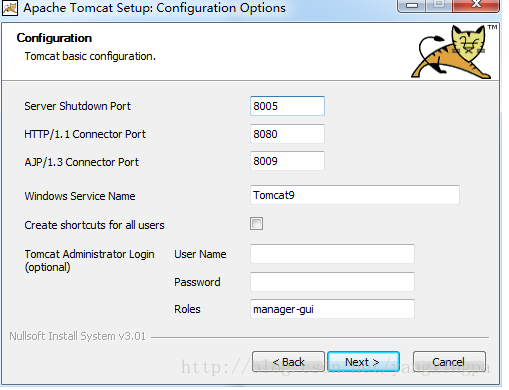
4）Tomcat安装：

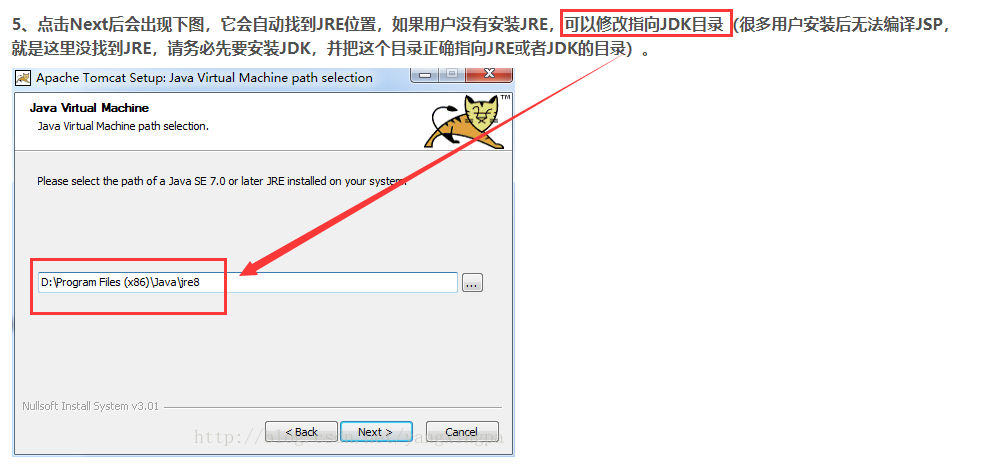
<https://tomcat.apache.org/download-10.cgi>

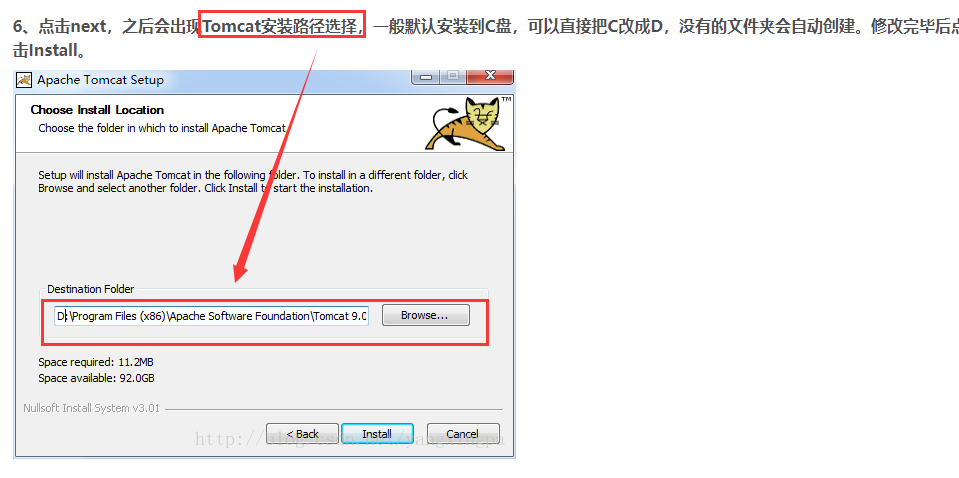


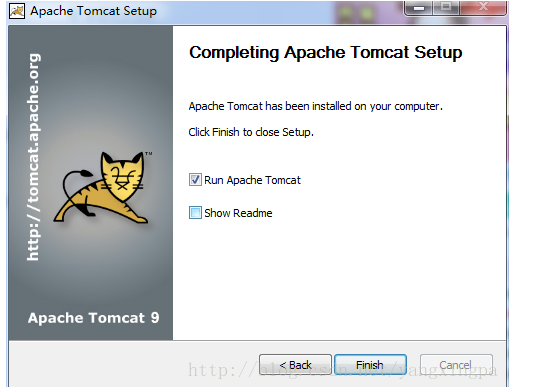










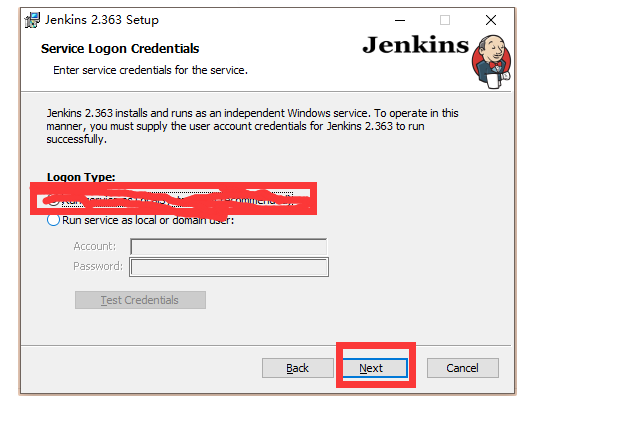




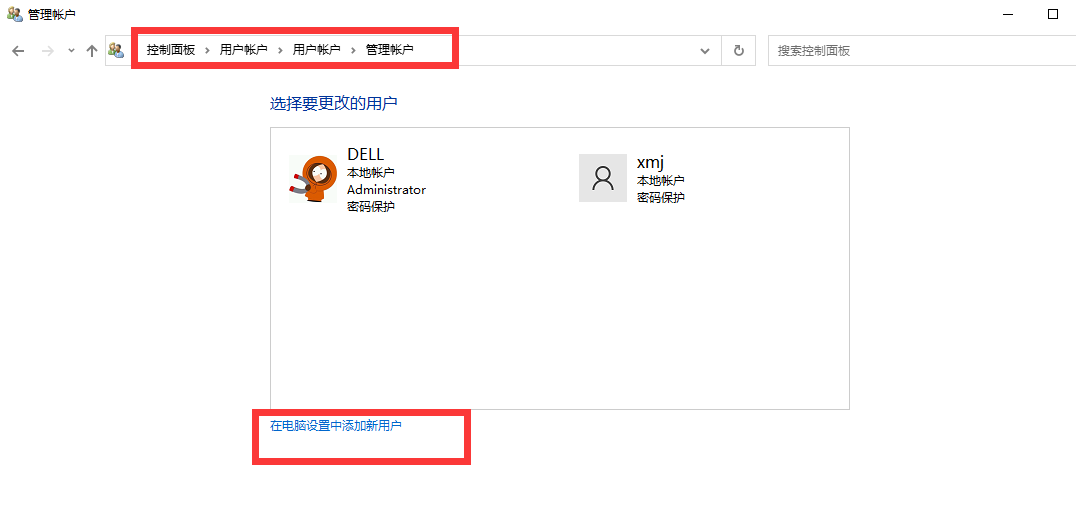
注意：验证是否配置成功，那就运行Tomcat,点击启动（或Start service）后，打开浏览器，敲入http://localhost:8080,如果出现页面，那么配置成功。

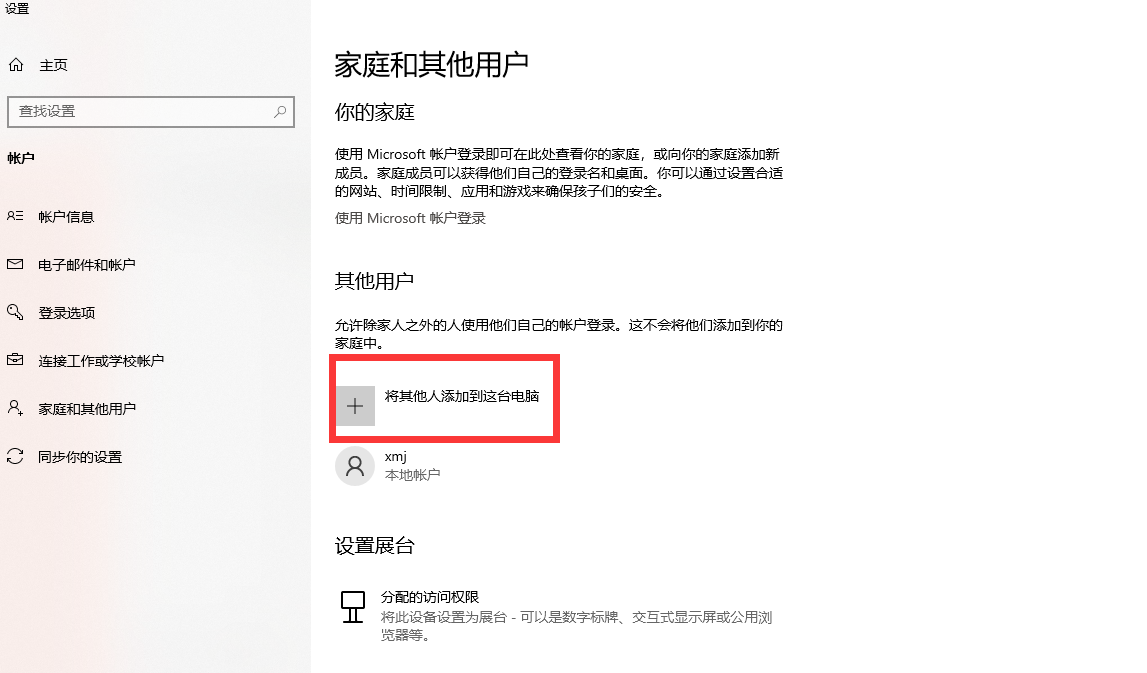
# 二、安装Jenkins

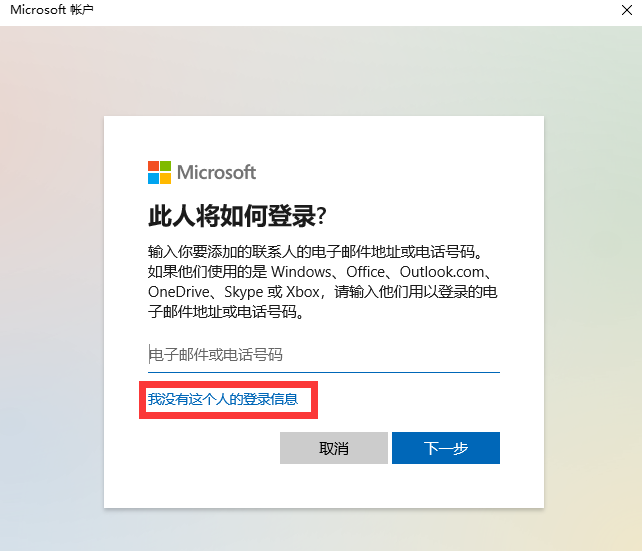
选择“Run service as local or domain user”，账号密码填写电脑登录密码，如果测试失败，按一下方式进行操作：

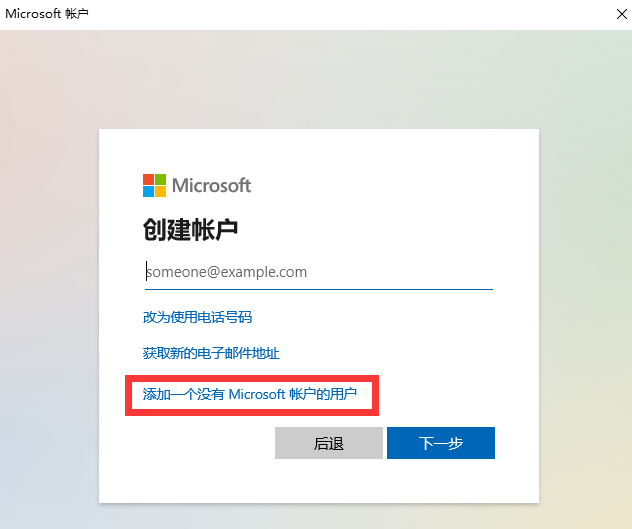


1）如果是账号密码测试不通过，重新申请一个电脑账号：



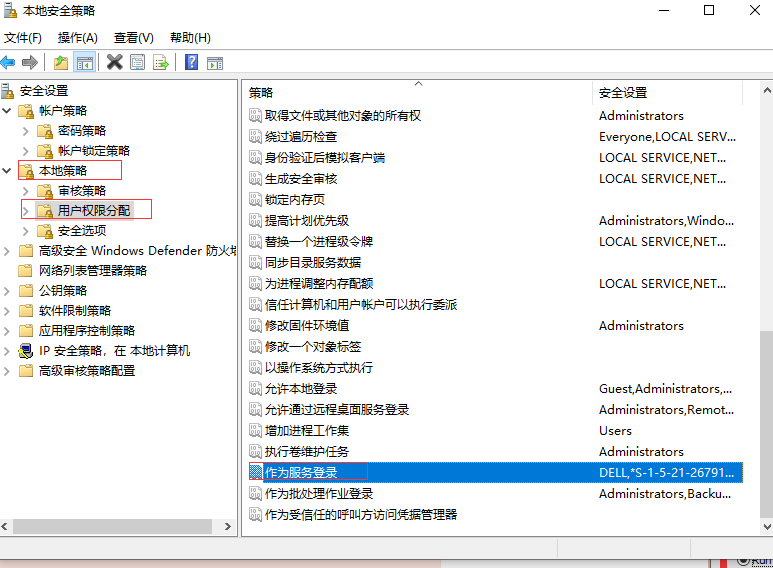


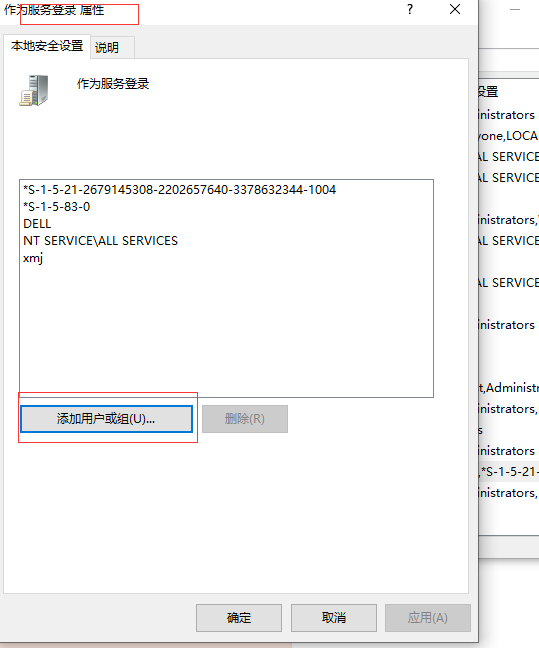


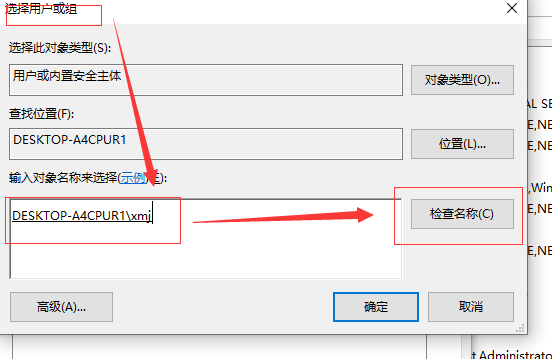




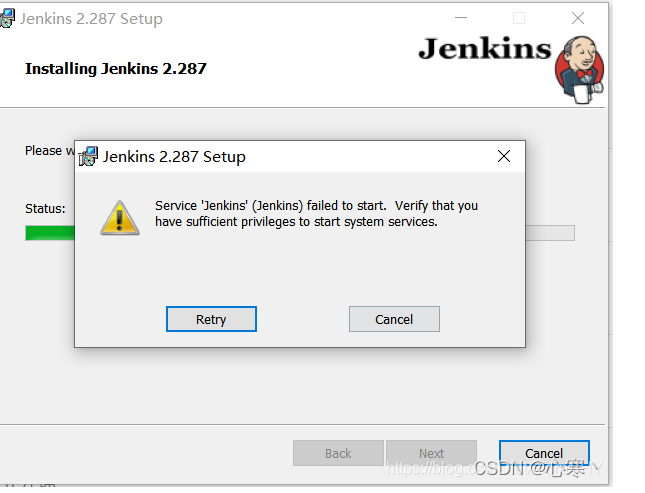
2）按一下步骤进行操作，最后如果显示用户名下有下划线，表示成功了，可以再对jenkins进行安装。

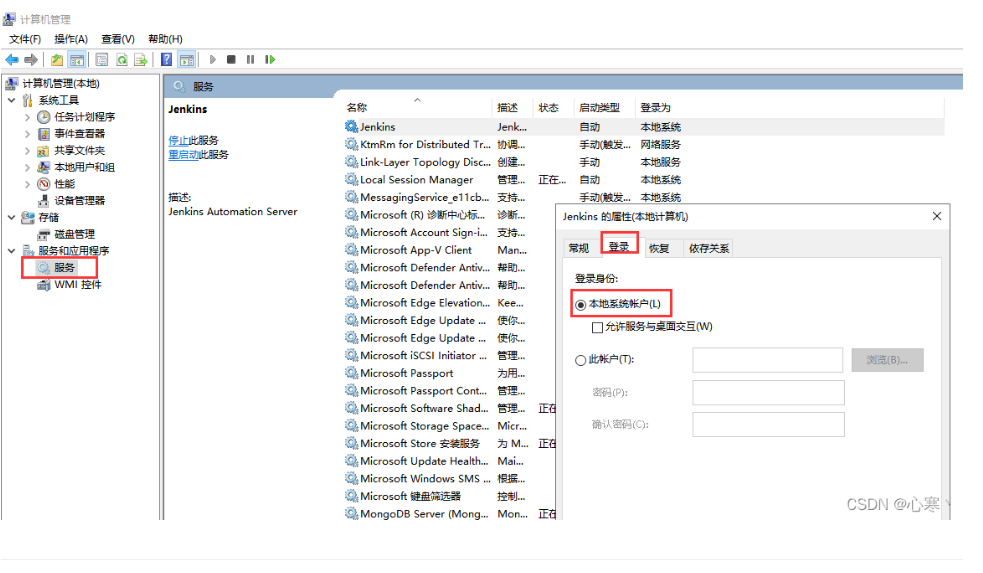






**注意：**如果安装过程中出现Retry, 打开 计算机管理-服务-jenkins右键属性-登录，选择本地系统用户登录即可.





# 三、启动Jenkins

1）进入jenkins：<http://127.0.0.1:9999>  （由于之前安装jenkins时候咱们设置的端口是9999，这个别忘了哟！）

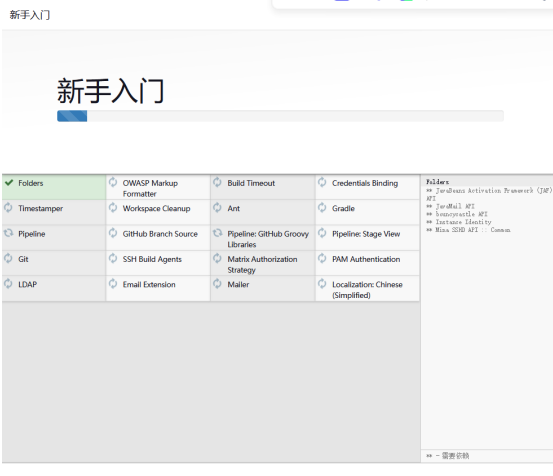


**注意：**密码所在位置



2）以下是jenkins的使用步骤：



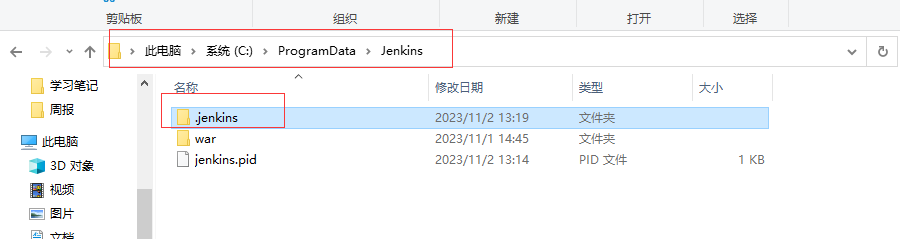






# 四、jenkins的卸载和重启

**卸载：**方法：windows部署的war包，在C盘搜索.jenkins，一般在C盘》用户下面，删除.jenkins文件夹即可



# 重启：http://127.0.0.1:9999/restart

# 五、jenkins安装完成后，插件屡次失败

原因：主要是jdk有问题，重新下载其他版本即可。

# 六、在jenkins中，创建一个自由风格的项目，配置的详细介绍

## 1．参数General

**1) Discard old builds：**

通过时间/次数来清理历史构造数据, 减少磁盘占用。

a.保留构建的天数：超过设置的天数，运行的任务自动删除；

b.保留构建的最大个数：超过设置的任务个数，会优先删除最早创建的任务；

**2) This project is parameterized:**

工程添加自定义参数, 支持多种类型: boolean, Choice, File, Password, Text, String等**。**

**3) Throttle builds：**

限制工程构造频率, 如每分钟仅构建1次。

a．构建数：

b．Time period：

**4) Execute concurrent builds if necessary：**

默认情况下, 工程在某个时刻只能执行1个构建，勾选后, 支持并行构建。

**5) GitHub 项目：**

代表从github仓库获取代码，设置 github 的项目地址。

1. Description；
2. Discard old builds——忽略较早的构建（选择保存的天数或者个数）；
3. This project is parameterized——给项目配置参数，构建时需要输入参数；
4. Disable this project——禁用该项目；
5. Execute concurrent builds if necessary——允许项目并行执行；
6. Quiet period——静默周期，触发下一次构建的CD时间；
7. Retry Count——当从SCM检入失败后，重试的次数；
8. Block build when upstream project is building——当上游项目正在构建时，不允许构建；
9. Block build when downstream project is building——当下游项目正在构建时，不允许构建；
10. Use custom workspace——手工指定编译位置；
11. Keep the build logs of dependencies——保留构建日志；

## 2．源码管理

**1) 无：**

没有任何操作；

**2) Git：**

从代码库获取代码；

a. Repository URL：填写代码块地址；

b. Credentials：添加凭证，包含代码库的账号和密码；

## 3．构建触发器

**1) 触发远程构建 (例如,使用脚本)：**

通过登录远程一个地址，就触发一个任务执行。

身份验证令牌：

http://localhost:8848/job/FlashRegistration/build??token=口令&cause=书写构建原因；

**2) Build after other projects are built：**

在其他项目触发的时候触发；里面有分为三种情况，也就是其他项目构建成功、失败、或者不稳定的时候触发项目。

**3) Build periodically：**

隔一段时间build一次，不管版本库代码是否发生变化，比如说填好日程表，每天晚上12:00构建一次，通常不会采用此种方式。

**4) GitHub hook trigger for GITScm polling：**

如果装了GitLab Plugin插件就会出现，不详解了主要用于Jenkins与GitLab相结合来在提交代码到GitLab后，触发构建。

**5) Poll SCM：**

按照设定的时间规则，先比较一次源代码是否发生变更，如果发生变更，那么就按设定的时间规则（例如：每天晚上12:00）build。否则，不进行build，通常采用这种方式。

## 4．构建环境

**1) Delete workspace before build starts：**

在构建之前清空工作空间；

**2) Use secret text(s) or file(s)：**

使用加密文件或者文本；

**3) Add timestamps to the Console Output：**

给控制台输出增加时间戳；

**4) Inspect build log for published build scans：**

**无；**

**5) Terminate a build if it's stuck：**

如果构建出现问题则终止构建；

**6) With Ant：**

ant相关；

## 5．构建

**1) Execute Windows batch command：**

执行Windows批处理命令；

**2) Execute shell：**

执行shell；

**3) Invoke Ant：**

调用ant；

**4) Invoke Gradle script：**

调用Gradle脚本；

**5) Invoke top-level Maven targets：**

调用top-level Maven目标；

**6) Run with timeout：**

运行超时；

**7) Set build status to "pending" on GitHub commit：**

在github上设置“等待”的构建状态；

## 6．构建后操作

**1) Aggregate downstream test results：**

综合下游测试结果；

**2) Archive the artifacts：**

归档文物；

**3) Build other projects：**

构建其他项目；

**4) Publish JUnit test result report：**

发布JUnit测试结果报告；

**5) Record fingerprints of files to track usage：**

技术文件中的指纹跟踪使用情况；

**6) Git Publisher：**

Git发行商；

**7) E-mail Notification：**

电子邮箱通知配置（邮件格式只支持jenkins固定格式）；

**8) Editable E-mail Notification：**

可编辑电子邮箱通知（发送格式可配）；

**9) Set GitHub commit status(universal)：**

设置github提交状态(通用)；

**10) Set build status on GitHub commit[deprecated] ：**

在github上设置构建状态[已弃用]；

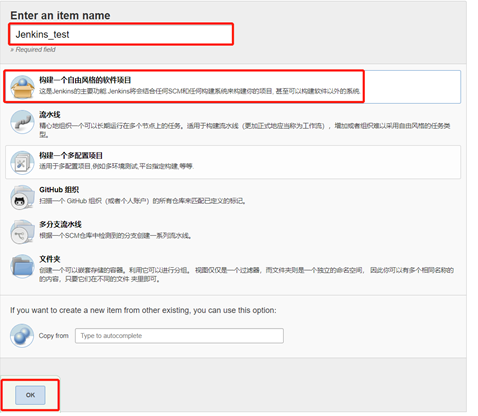
**11) Delete workspace when build is done：**

在完成构建时删除workspace；

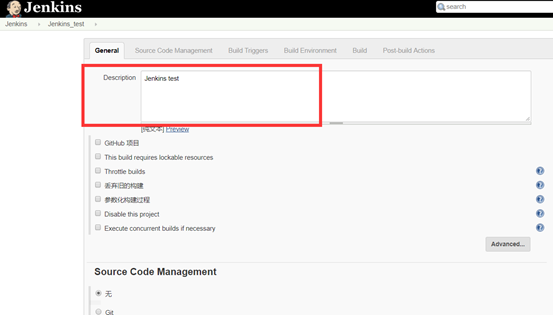
# 七、jenkins项目部署案例

1）一个简单的python脚本执行案例：

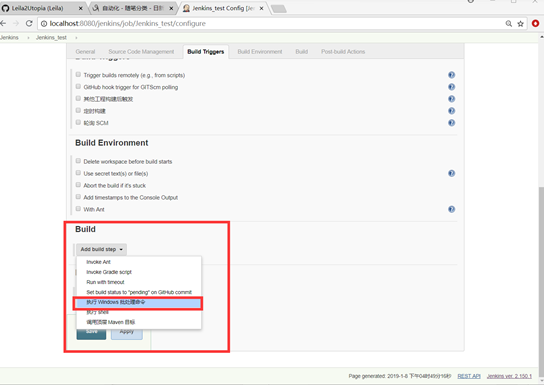
第一步：在Jenkins首页点击‘New 任务’进入创建任务页面，在‘Enter an item name’输入框内输入项目名称，选择Jenkins任务类型，这里我们选择'构建一个自由风格的软件项目',点击'OK'。



第二步：跳转至配置页面，添加项目描述；



第三步：滚动页面至Build选项，点击下拉框，选择'执行Windows批处理命令；



第四步：在输入框中输入命令如图所示命令，点击'save'

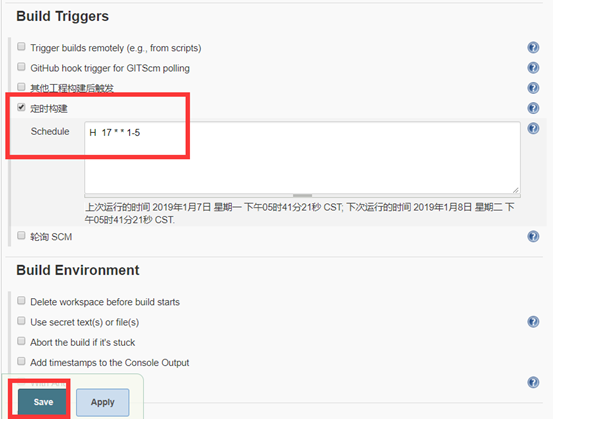


第五步：在Jenkins首页点击项目Jenkins\_test，进入项目管理页面，点击左侧的'立即构建',在'Bulid History'列表，你会得到一个蓝色的小圆点，表示构建成功，点击该小圆点，查看控制台输出，python脚本执行成功。





第六步：每次都手动的构建项目不方便，因此我们可以在Jenkins中设置定时任务，在项目管理页面点击'Configure'，修改项目的配置，找到'Build Triggers'，勾选‘定时构建’,输入Schedule(此处格式的语法可以点击后面的帮助进行查看)，点击'save'即可



# 八、Jenkins 持续集成环境构建四（Jenkins 不同项目构建类型：自由风格、Maven、Pipeline）

## 1．自由风格的jenkins 项目

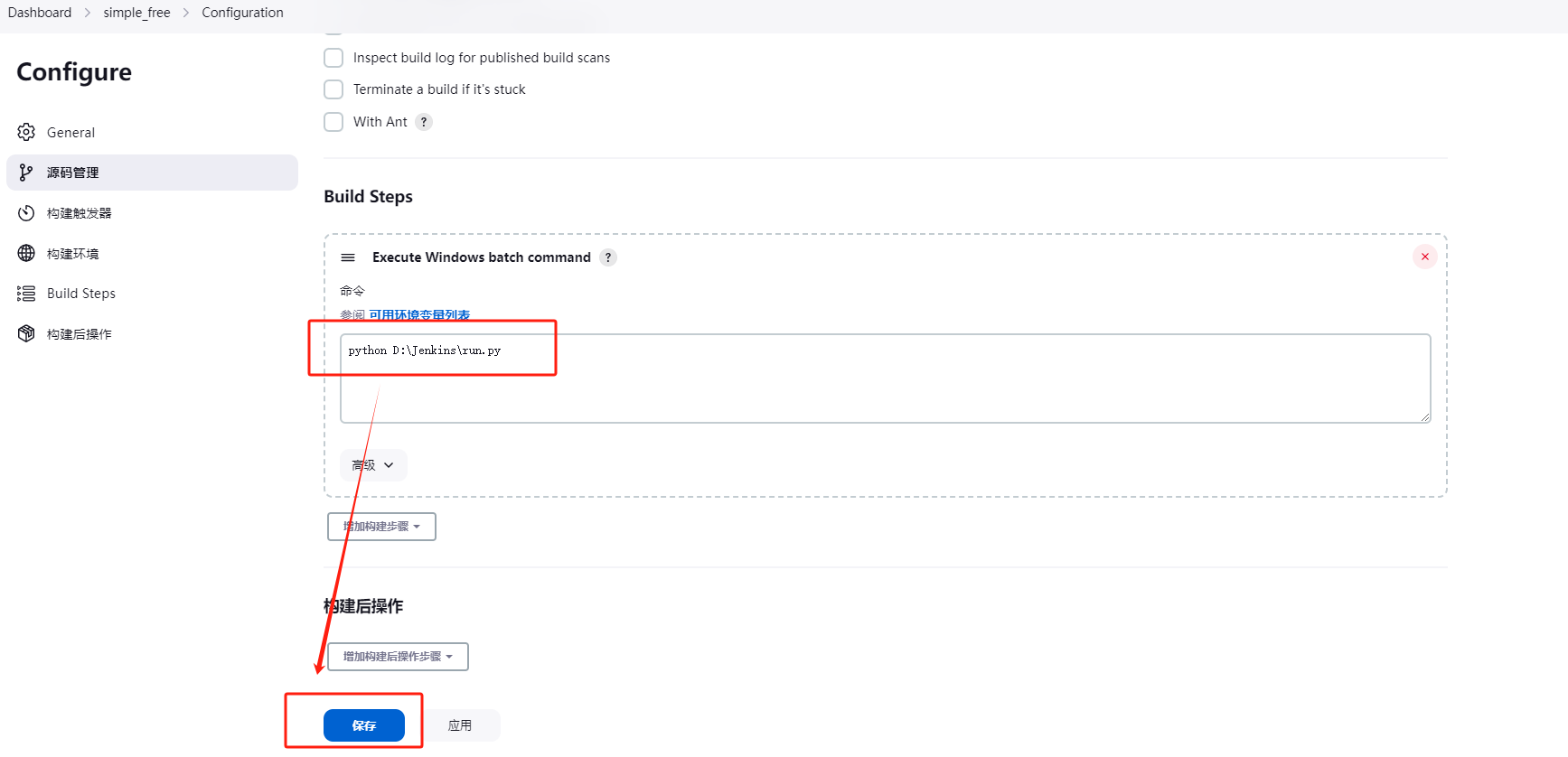
### 案例一：

有2处修改：

1)选择Discard old builds：jenkins的创建后保留期限和保留个数；

2)Build Steps：选择“Execute Windows batch command”，通过运行脚本命令；





### 案例二：

有3处修改：

1)选择Discard old builds：jenkins的创建后保留期限和保留个数；

2)Build Steps：选择“Execute Windows batch command”，通过运行脚本命令；

3)构建触发器：选择“Build periodically”，定时运行jenkins；

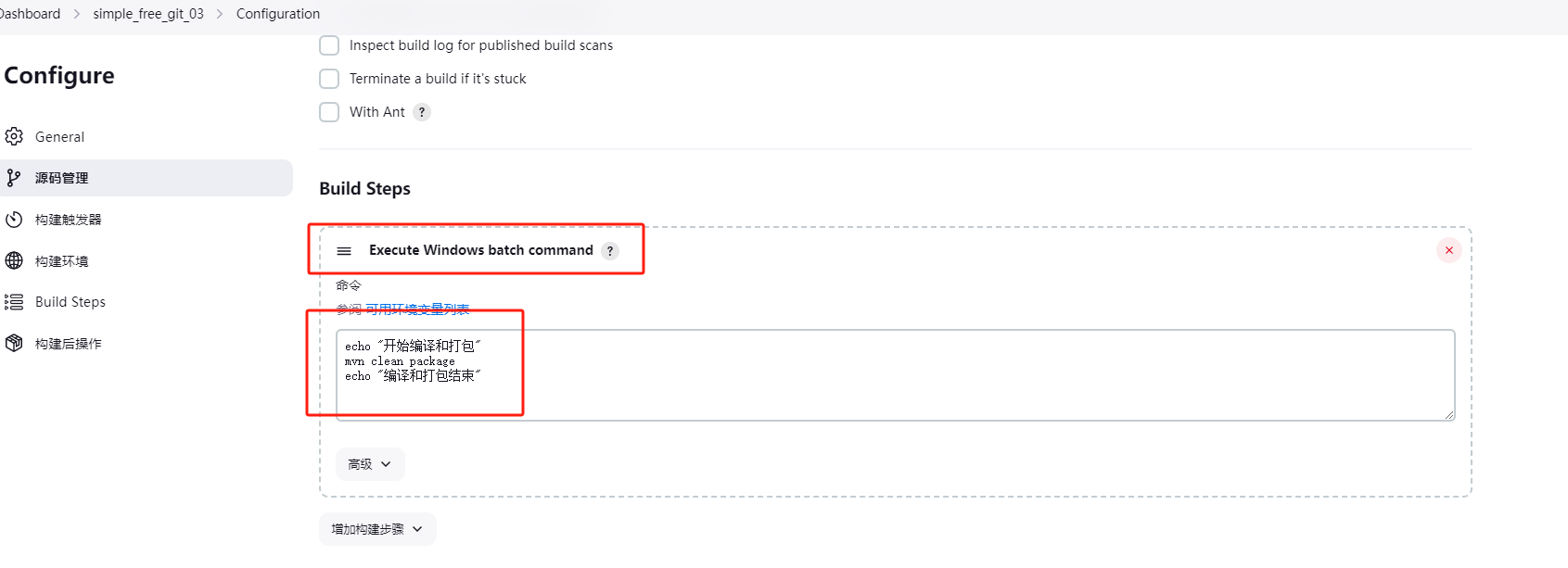


### 案例三：

有6处修改：

1)选择Discard old builds：jenkins的创建后保留期限和保留个数；

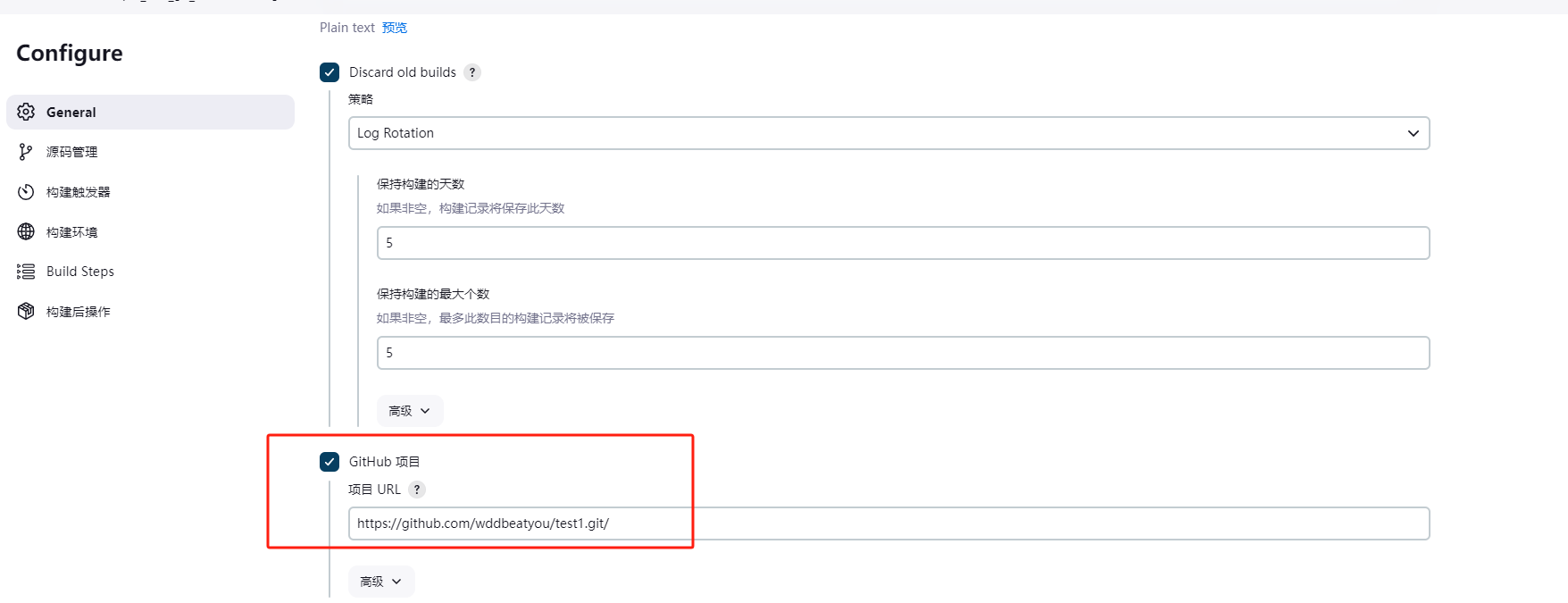
2)Build Steps：选择“Execute Windows batch command”，通过运行脚本命令；



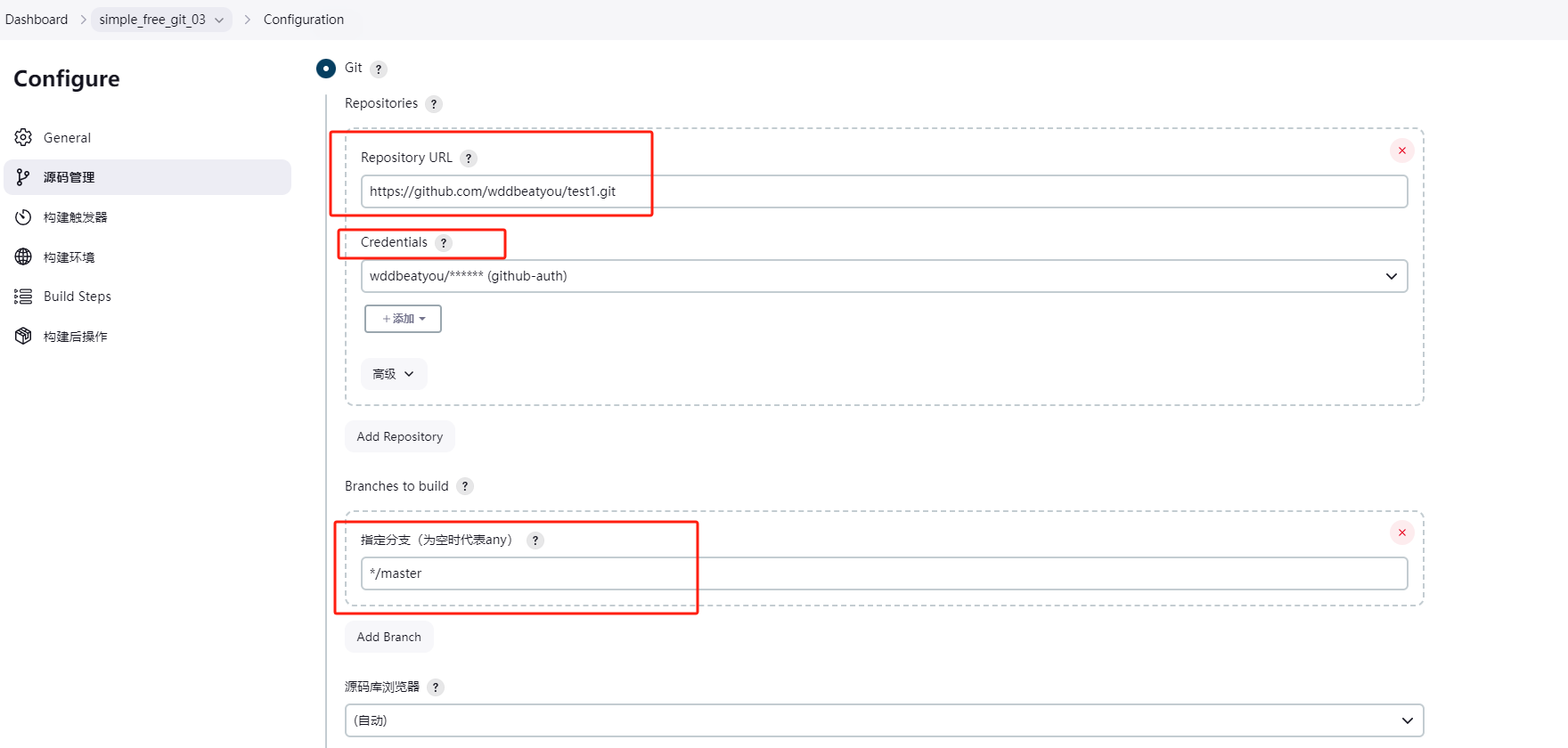
3)构建触发器：选择“Build periodically”，定时运行jenkins；

4)选择GitHub 项目：填写git仓库地址，如：

<https://github.com/wddbeatyou/test1.git>；



5)源码管理：选择Git,填写git项目地址，“Credentials”凭证填git仓库的账号和密码；



6)构建后操作：选择“**Deploy war/ear to a container**”，“WAR/EAR files”填写固定格式，如：\*/target/\*.war，“Containers”选择Tomcat 9.x Remote，“Credentials”凭证tomcat账号和密码，“Tomcat URL”填写打包的地址；

