# Jenkins使用手册

Jenkins 是一款流行的开源持续集成（Continuous Integration）工具，广泛用于项目开发，具有自动化构建、测试和部署等功能。本文以 CentOS7 环境为例，介绍Jenkins 的安装与使用。

操作环境：Windows

一、环境准备

1 安装JDK

  本文采用jdk-8u111-windows-x64.exe；

2 配置tomcat

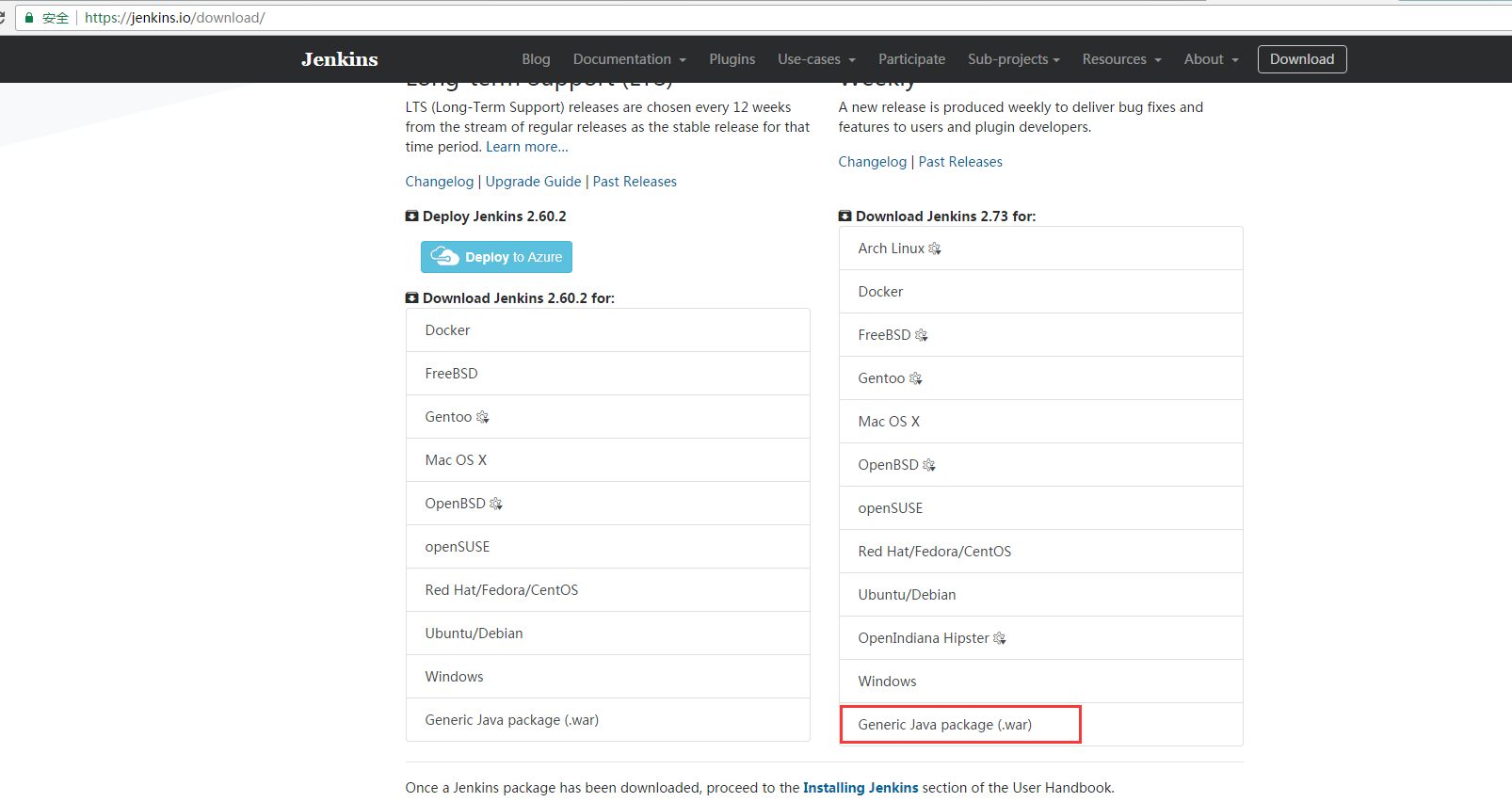
  本文采用tomcat8，无需安装，配置JAVA\_HOME及JRE\_HOME环境变量即可；

3 安装maven

  本文采用maven3.3.9，无需安装；

4 安装Jenkins

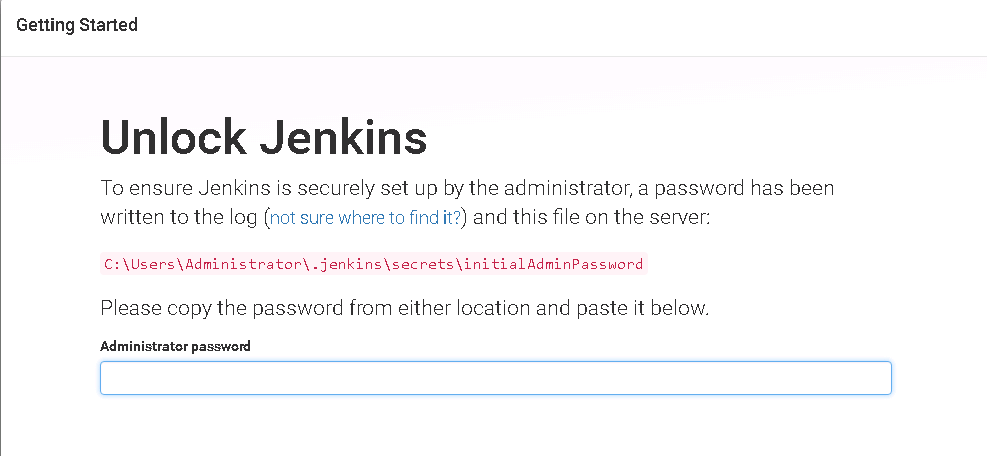
  下载地址https://jenkins.io/download/，仅下载war包，如下图：



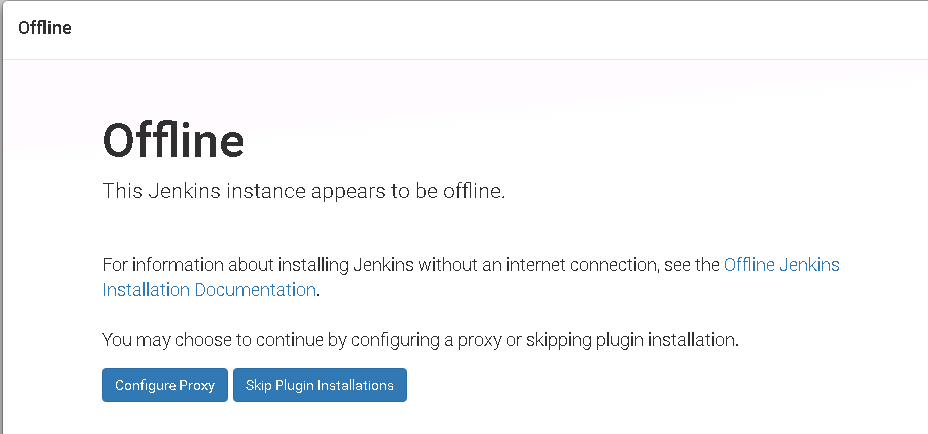
将下载好的jenkins.war放进tomcat/webapps目录下。运行Tomcat，默认访问地址：<http://127.0.0.1:8080/jenkins/>。

二、相关配置

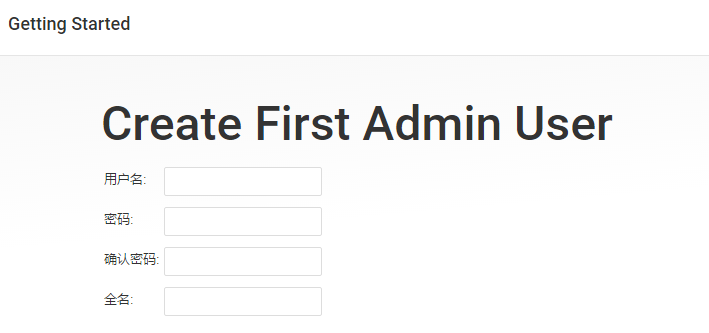
1 登入http://localhost:8080/jenkins，进入Jenkins初始化页面，第一次启动时间可能有点长，耐心等待。进入成功后会看到如下画面，按提示路径打开密码文件，输入密码：



解锁后又是一长段时间等待，此后可能出现如下图所示界面：



表示无法下载Jenkins插件，可能是因为防火墙导致，而Jenkins插件的安装非常重要，建议翻墙。如无法翻墙，则选择Skip Plugin Installations跳过插件安装。进入以下页面，设置登陆用户：



2 设置成功后即进入Jenkins主界面：

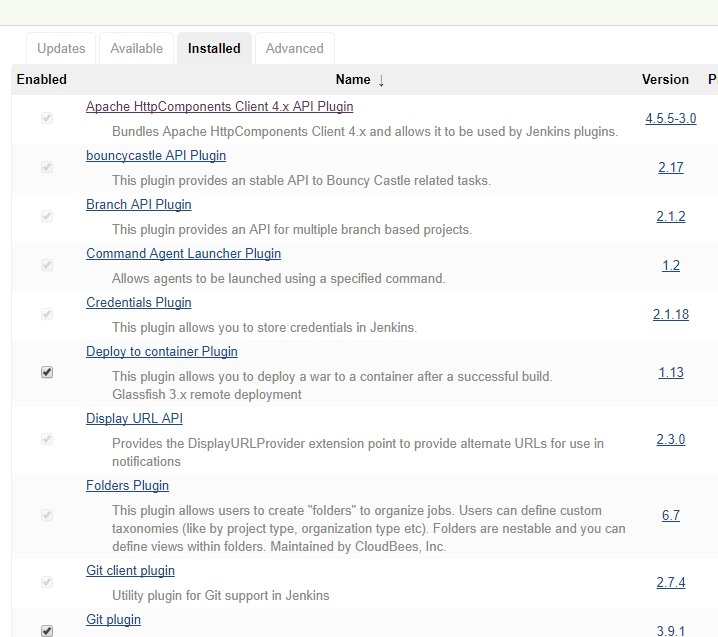


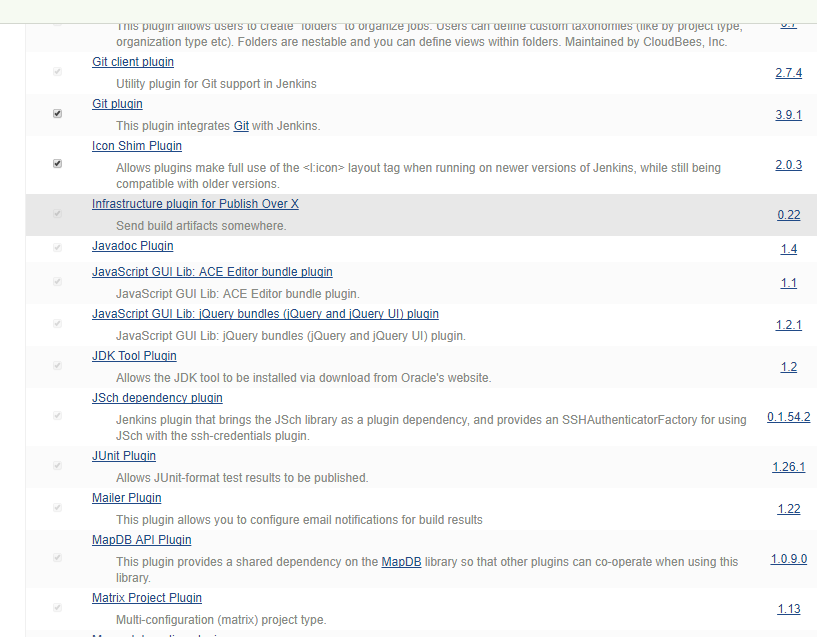
点击左上侧系统管理，进入Jenkins基本系统设置(主要是以下三块):

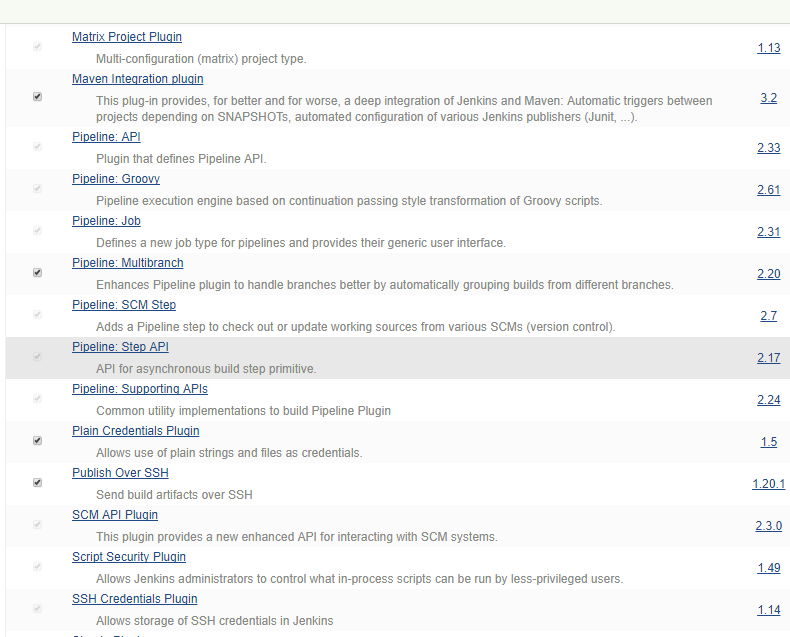


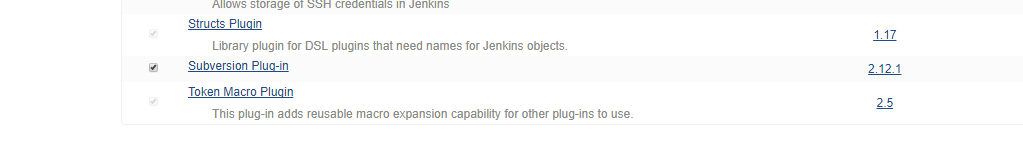
3 先进入“管理插件”模块安装必需的插件，以下是本机已安装列表，只安装了部分插件，如果后续有需要，请在Avaliable列表下载自己所需要的插件。

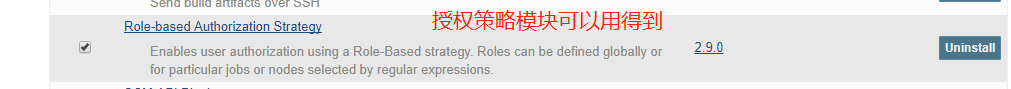
Publish Over SSH插件比较重要，用于发送war包等到Linux服务器，可以先只下载这一个插件。







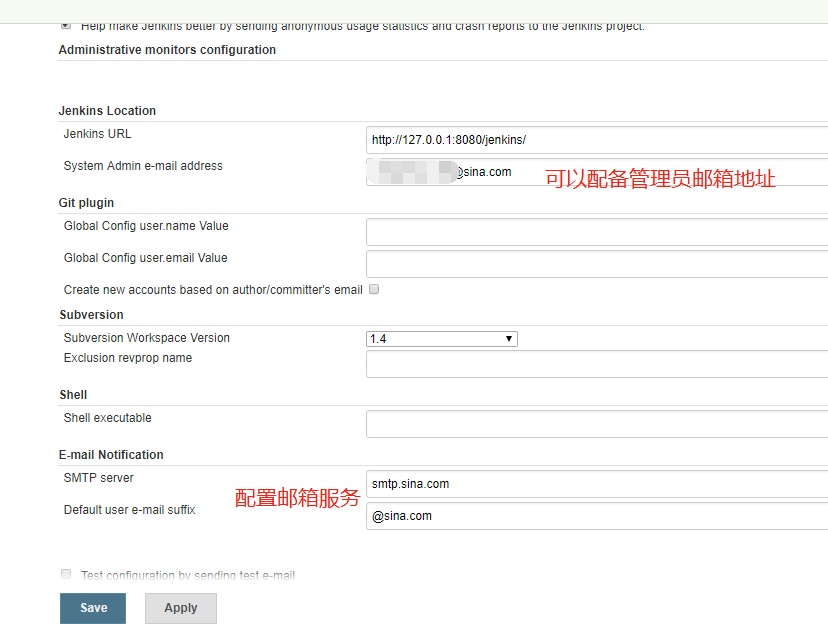


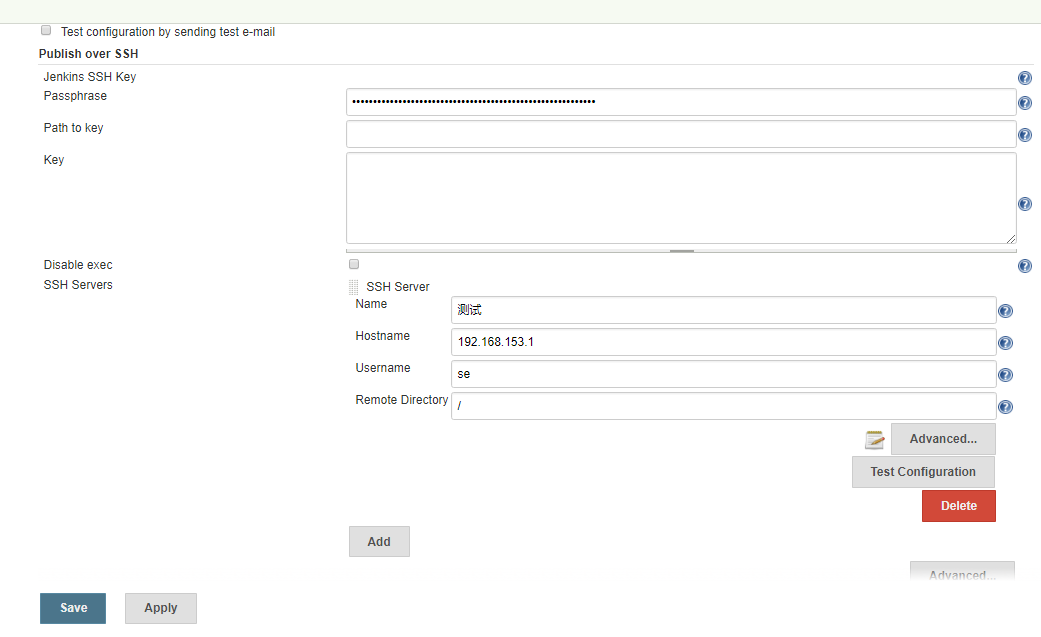


4 配置系统设置



增加系统管理员邮件地址：

  
下面是ssh信息，可以配置Linux服务器的信息，用于后面使用



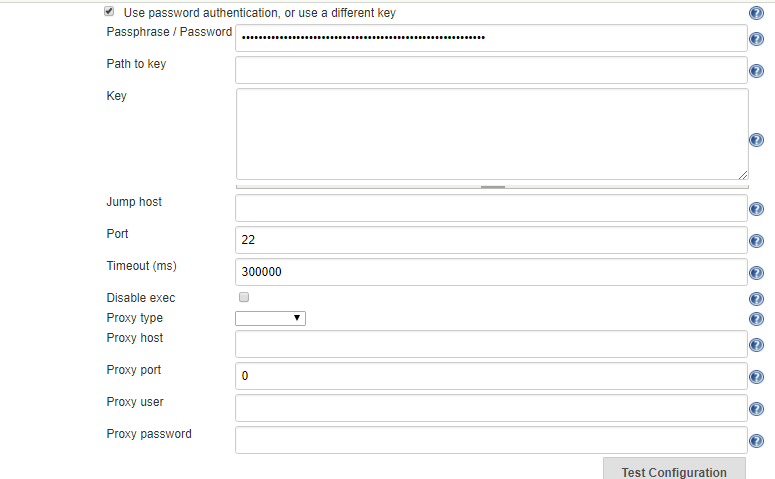
Name:ssh 服务名称，自己起一个

Hostname:服务器地址

Username:服务器用户名

|  |
| --- |
| Remote Directory:填写 / 就行 |

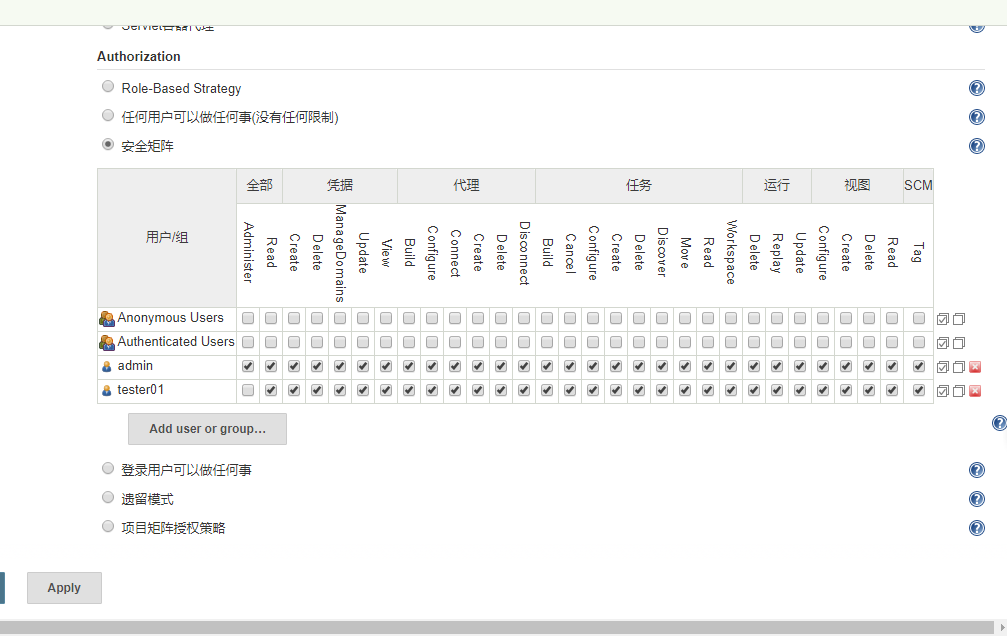
接下来点击advanced 高级选项，选中Use password authentication, or use a different key， 在Passphrase / Password输入服务器用户名的密码



其他的可用默认配置，保存后退出。

5 ：添加全局配置[Global ToolConfiguration](http://localhost:8080/jenkins/configureTools)

5.1：全局安全配置：对安全性有要求的话，这个地方就配置下，不配置也可以使用jenkins，只是安全性差点。下图中，添加了两个用户到组里（这两个用户已经存在或已经由你创建了），分配勾选不同的权限，不同用户登录后，jenkins管理界面或操作权限应该就变了。



5.2全局工具配置：



配置JDK，不采用自动安装：



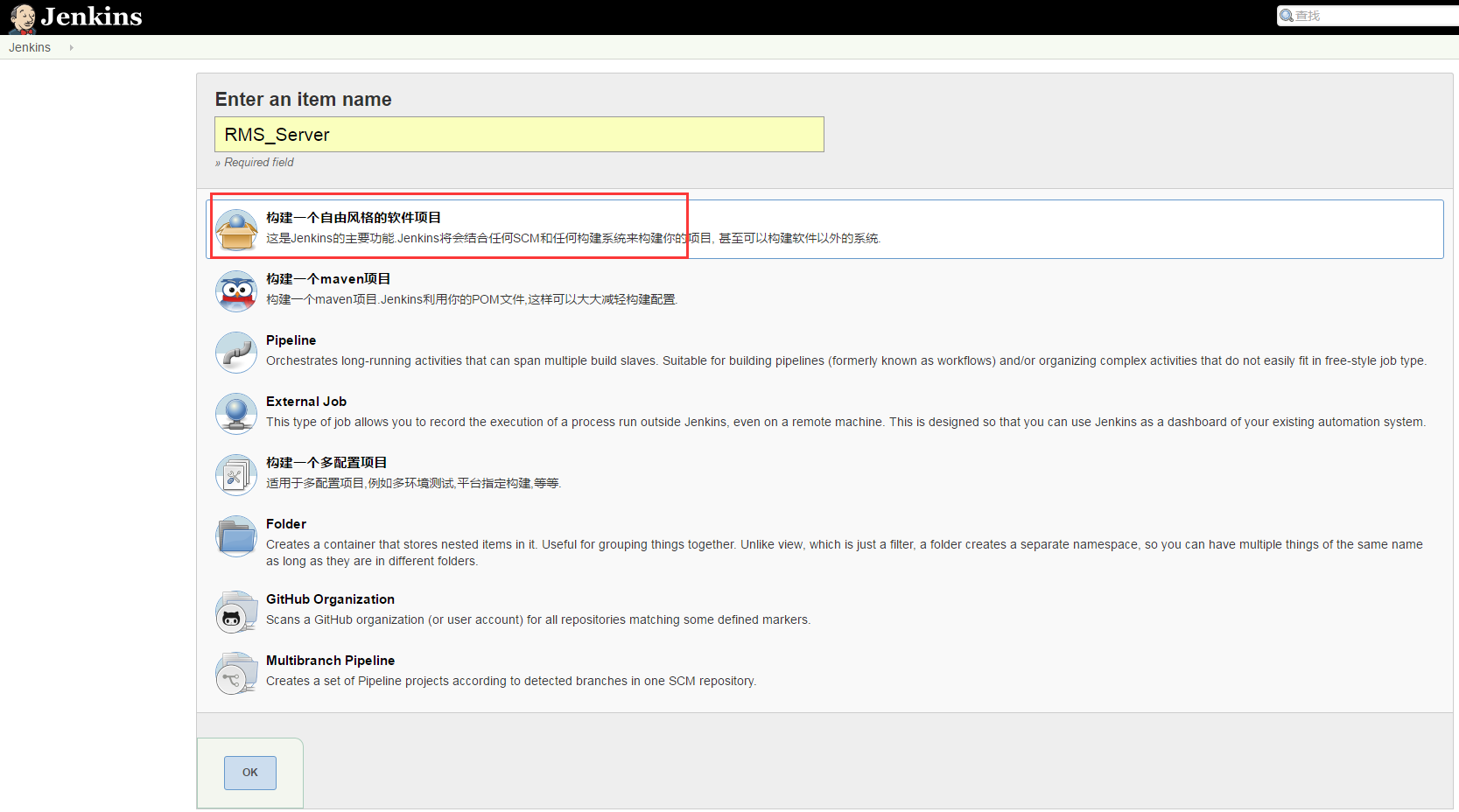
配置maven，不采用自动安装：



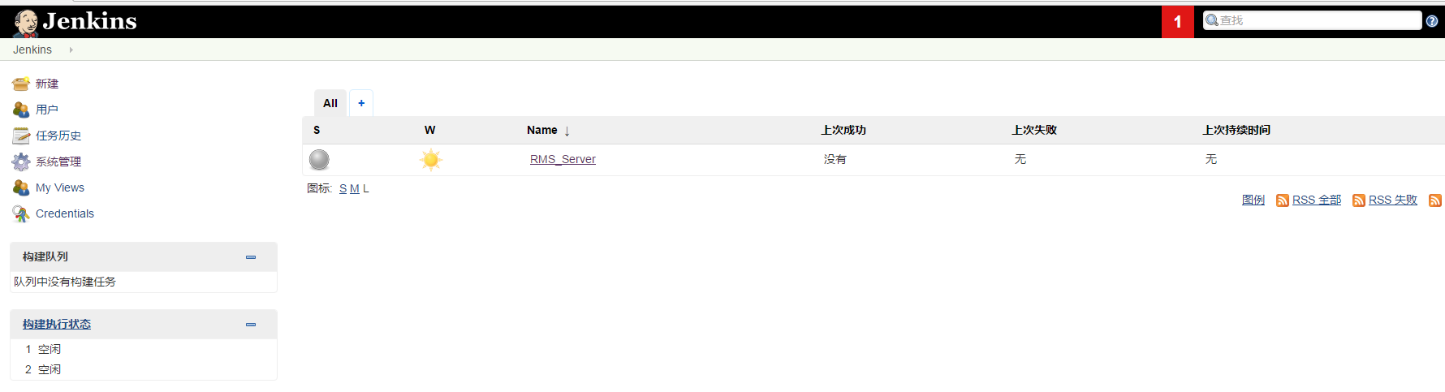
以上即为需要设置的系统配置。

三、创建任务和配置任务

1：系统设置完成后开始添加任务，任务类型选择自由风格：

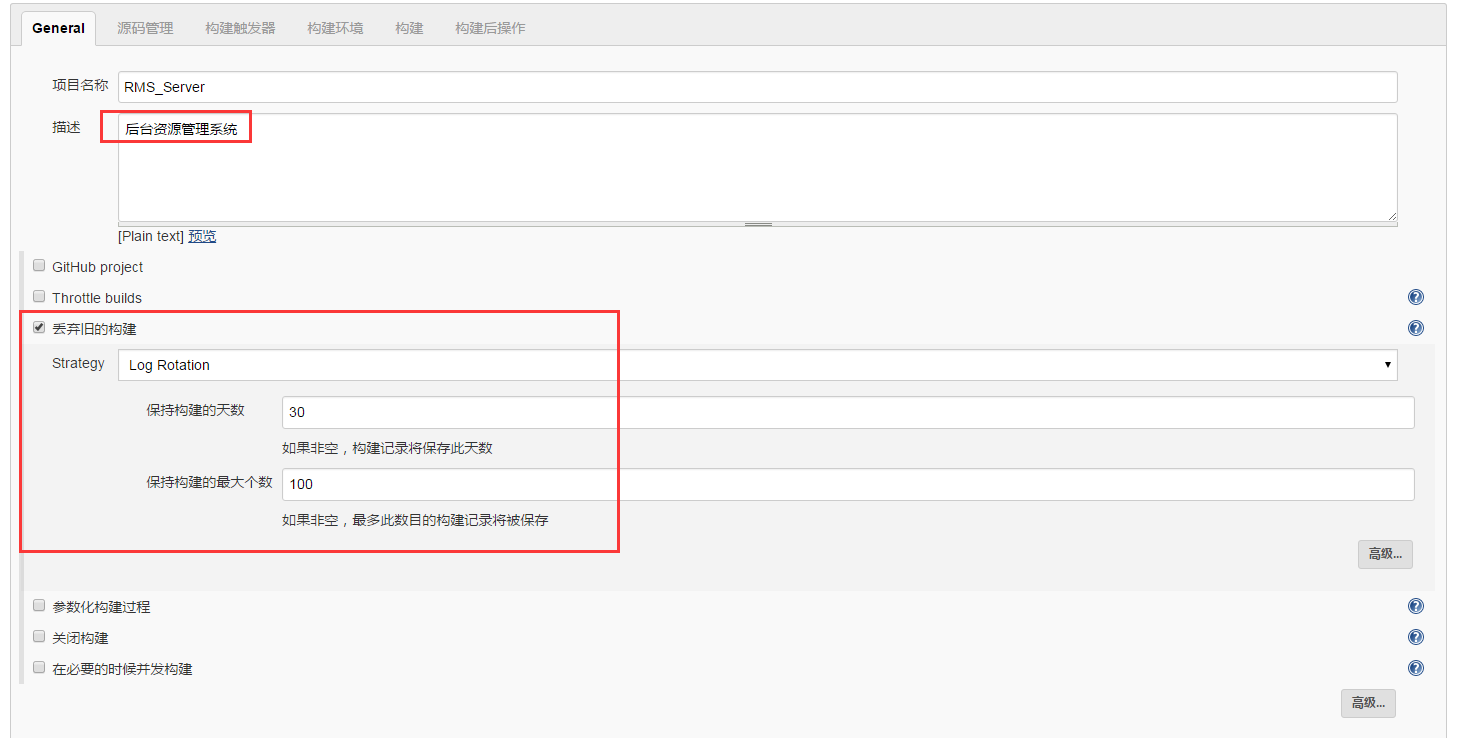


创建完成后可在主页看到如下画面：

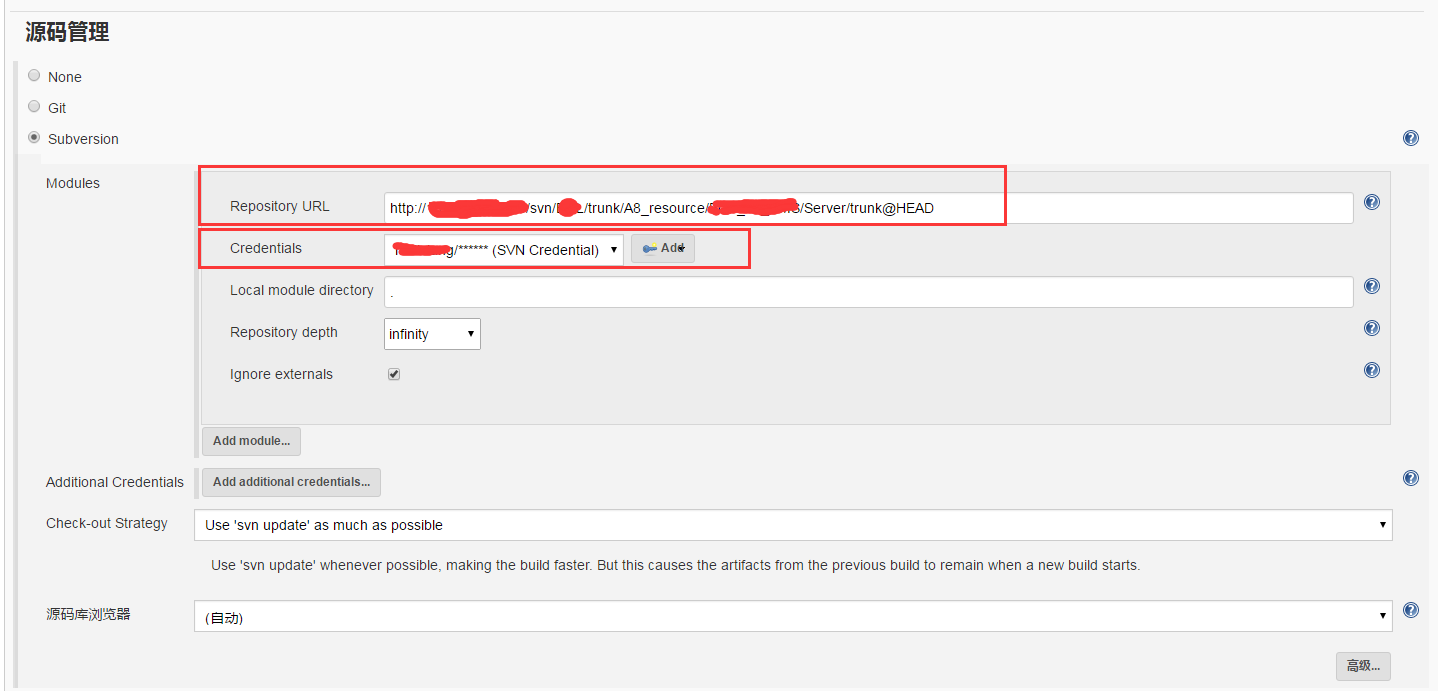


在”All” tab下能看到新建的任务，点击该任务，进入该任务的配置页面：

  
2：设置项目备注及构建规则：

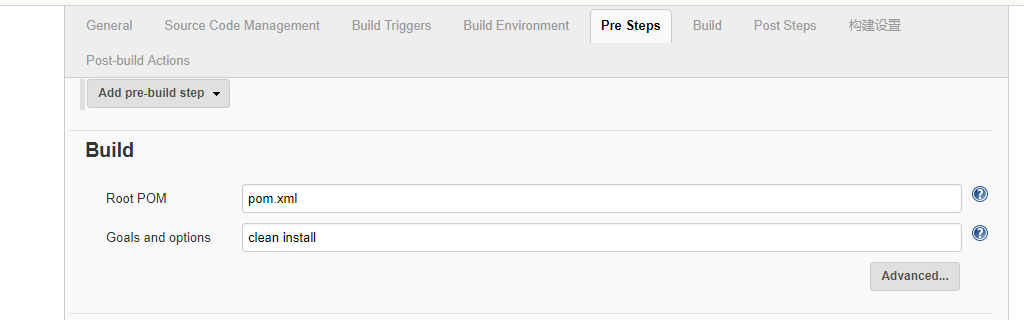


3：配置项目轮询的源码位置(@HEAD表示构建最新的代码)并配置代码访问密码：

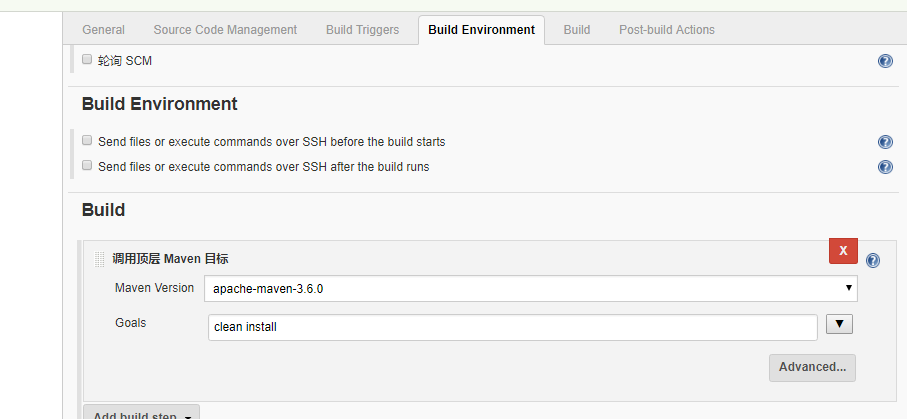
  
4：配置构建触发器，如下图配置为每天晚上9：30开始构建（Cron表达式）：



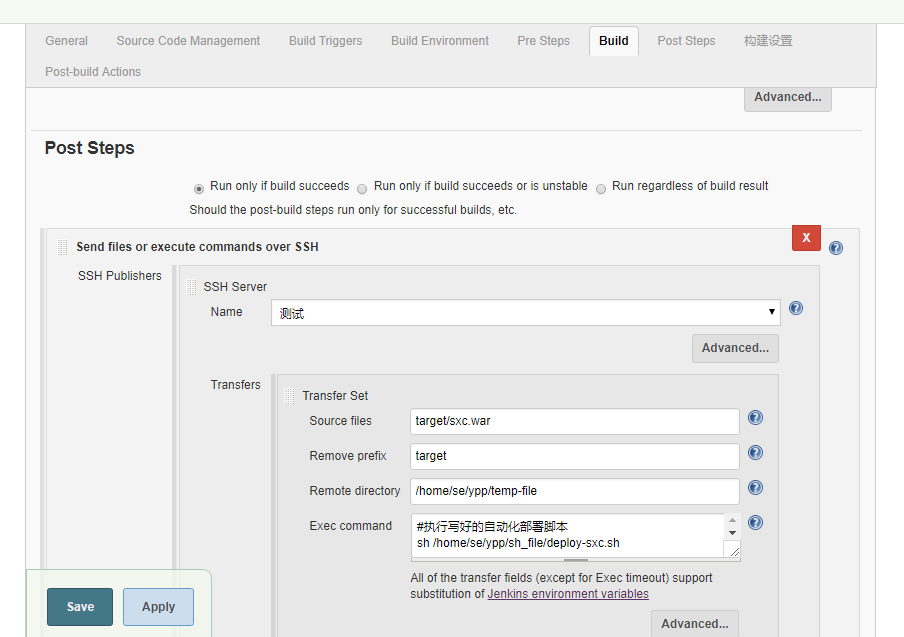
5：构建



或者



6：构建后后续步骤



Name：之前配置的ssh server名称，下拉选择

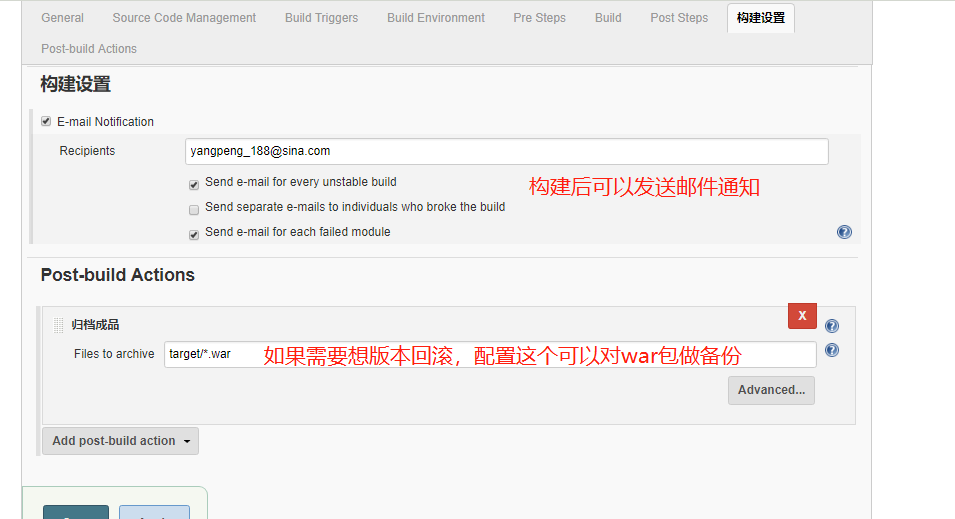
Source files：target/项目名.war

Remote prefix:target

Remote directory:linux服务器接收war包等的位置

Exec command：sh /home/se/ypp/sh\_file/deploy-sxc.sh（这个shell脚本在服务器上已经存储）

7：归档和邮件通知

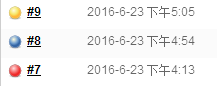


以上即为本项目的所有配置，完成后应用（或保存）并退出。

四：配置完成后即可开始构建

1：执行构建

### https://img-blog.csdn.net/20170822111443945 2：构建结果：在系统构建之后，在结果栏会显示构建状态、构建编号、构建时间。构建状态是用原型图片显示的



灰色：从未执行构建过的项目。

蓝色：成功，表示项目打包成功，执行过程中没有出现错误。

黄色：异常，表示项目打包成功，但执行过程中有异常出现。

红色：失败，表示项目打包失败。

**项目打包目录：**想成在打包成功后，生成的war或jar包，放会在Jenkins的warkspace目录下。

Windows项目为：$JENKINS\_HOME$\jobs\$your\_job\_name$\workspace\target\

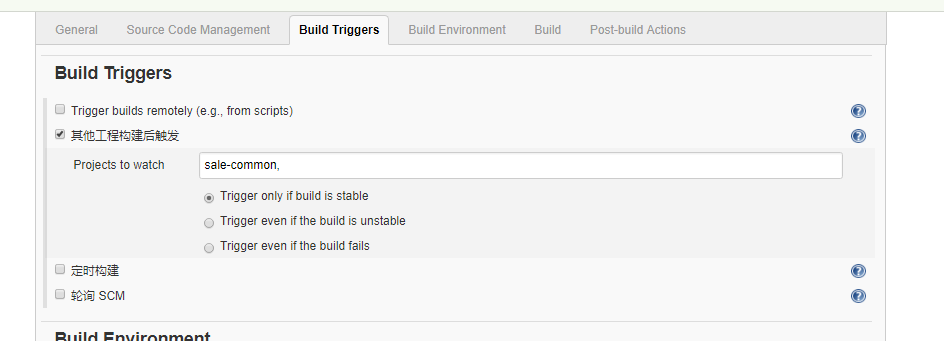
Linux系统为：user/.jenkins/workspace/$your\_job\_name$/target/

点击某个构建记录，如上图中的#31，即可查看构建日志、SVN代码提交日志及bugs分析结果：

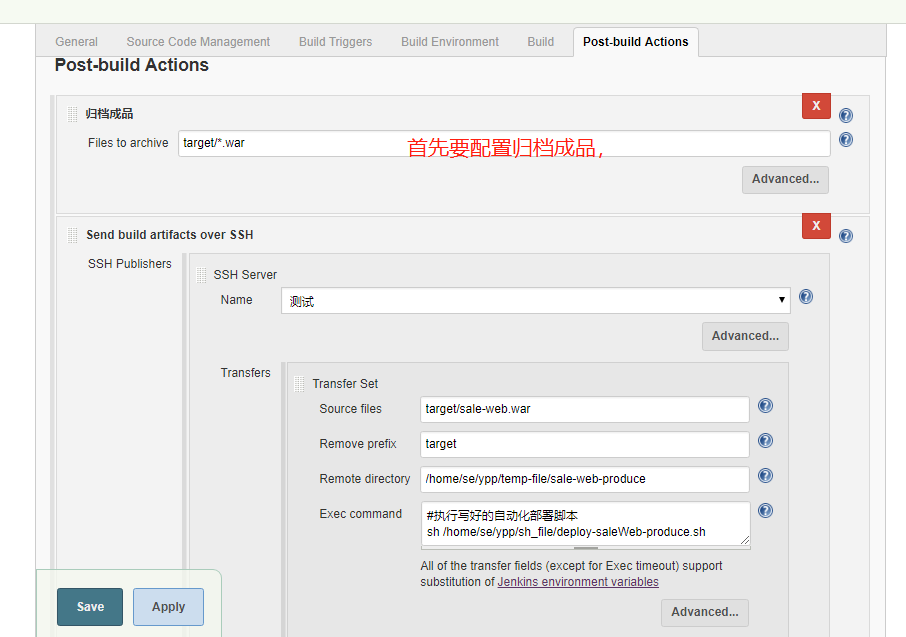


五、多项目依赖构建和参数化构建(用于版本回滚)

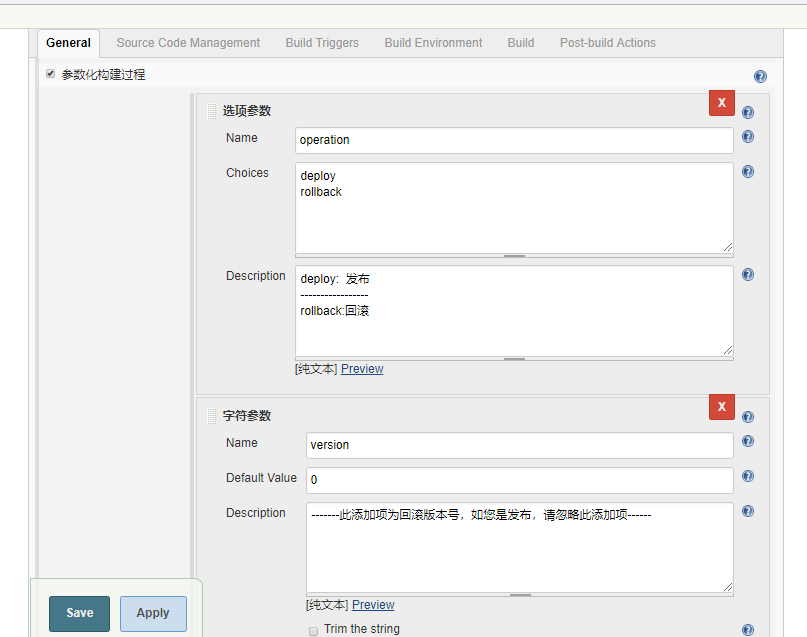
1：当前项目A依赖项目B的jar包，可以配置构建触发器，每次B构建完，A就会自动构建一次。



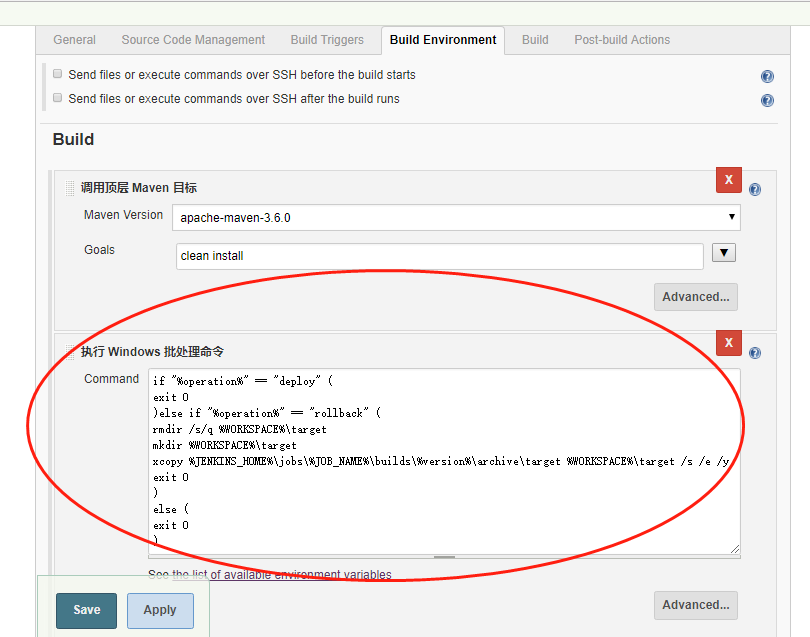
### 2：版本回滚



配置参数化构建



批处理命令



开始构建发布或者回滚

