# 1.下载安装包[jenkins.war](http://mirrors.jenkins.io/war-stable/latest/jenkins.war)；

# 2.在安装包根路径下，运行命令 java -jar

# jenkins.war --httpPort=8080，（linux环境、Windows环境都一样）；

# 3.打开浏览器进入链接 [http://localhost:8080](http://localhost:8080/).

# 4.填写初始密码，激活系统





安装完成



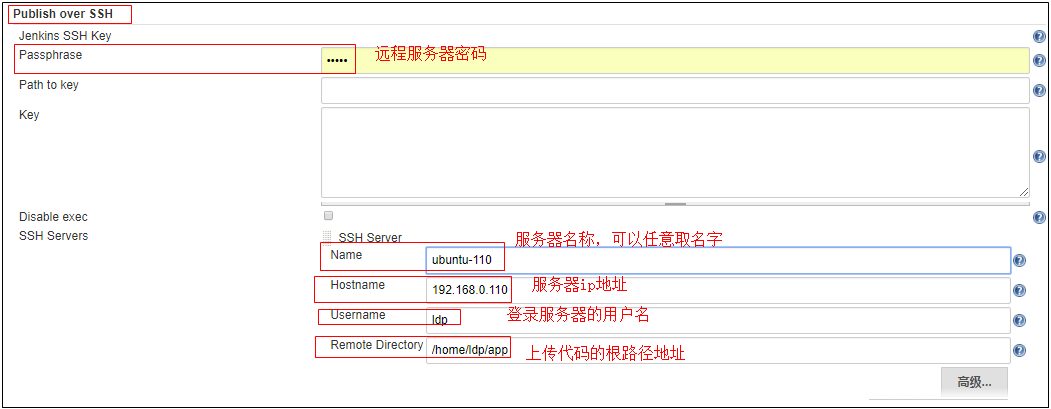
**1.Configure System (系统设置)**

在系统设置这里，我们只需要设置最后面的一项，配置远程服务器地址，

即我们代码最终运行的服务器地址信息，就像我们之前手动部署时使用xshell登录Linux服务器一样，

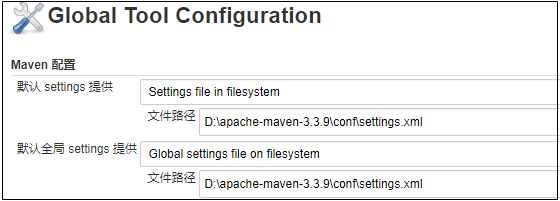
当然这里是可以配置多台远程Linux服务器的，配置完成后点击保存即可，为后面我们配置自动化部署做准备，配置如下图

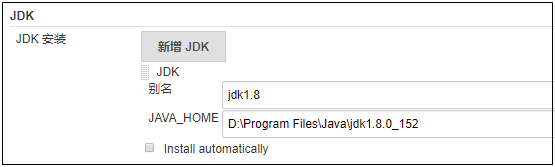


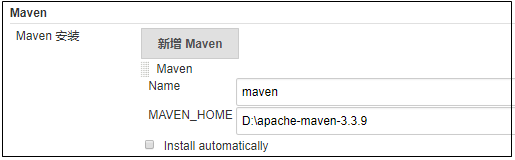


**3.Global Tool Configuration (全局工具配置 )**

　　a.配置maven的全局settings路径



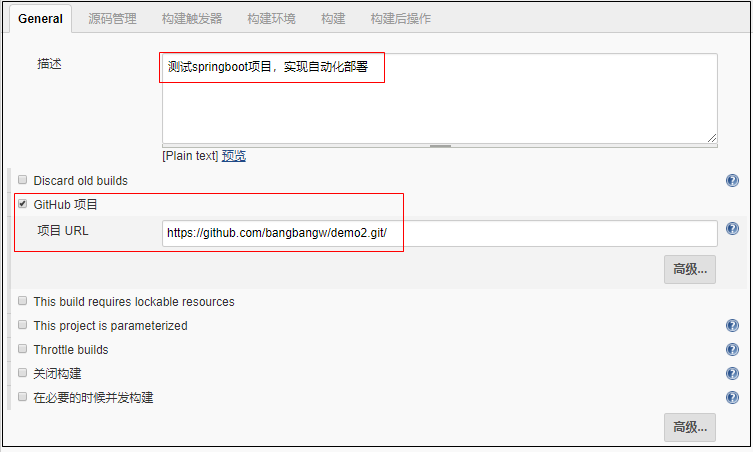


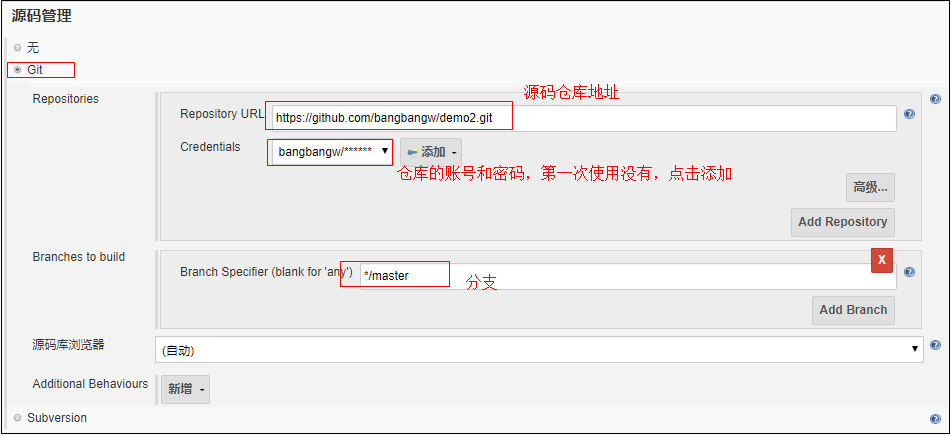


# 5.Jenkins自动化部署（springBoot+maven+gitHub）项目

大体步骤：General(基础配置)--》源码管理--》构建触发器--》构建环境--》构建--》构建后操作

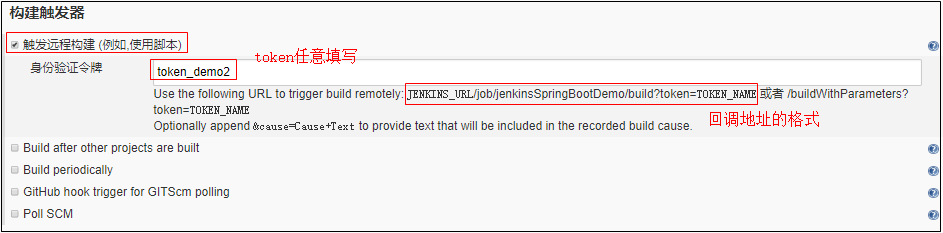








4.构建触发器



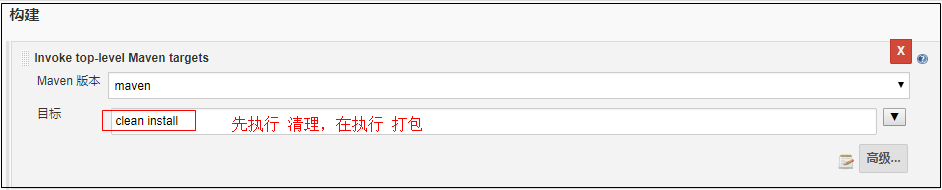
如上图：当前项目的回调地址为：

http://localhost:8080/job/jenkinsSpringBootDemo/build?token=token\_demo2

只要执行这个地址（在浏览器上访问改地址），该项目就会发起一次构建项目，即拉取代码打包部署操作，

在实际中，是由git服务器回调改地址，在后面讲git服务器配置时详细讲解 ，

值得注意的是，如果你是学习测试，使用的是GitHub服务，要确保你的回调地址已经试外网映射过的，即往外网可以访问

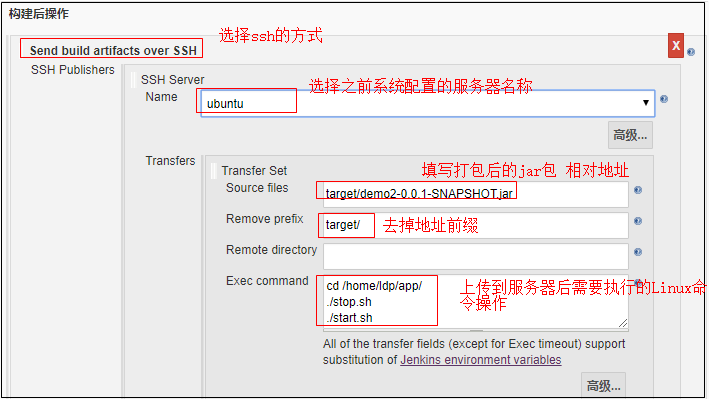


5.构建后操作

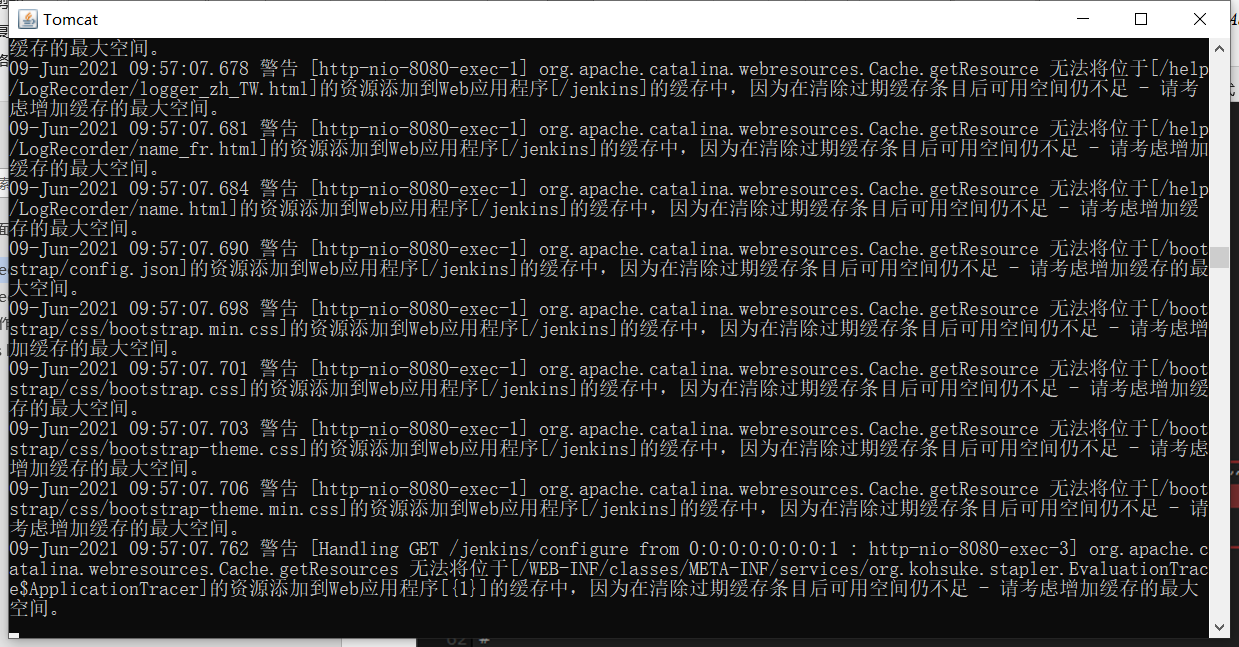
构建后操作的意思是，jar打包好后，要将jar发送到哪里去，发送后去和启动等

这里需要提前在需要部署的服务器上配置好路径，写好启动和停止项目的脚本，并设置为可以执行的脚本，

其实就是我们平时在Linux上手动部署项目操作的脚本



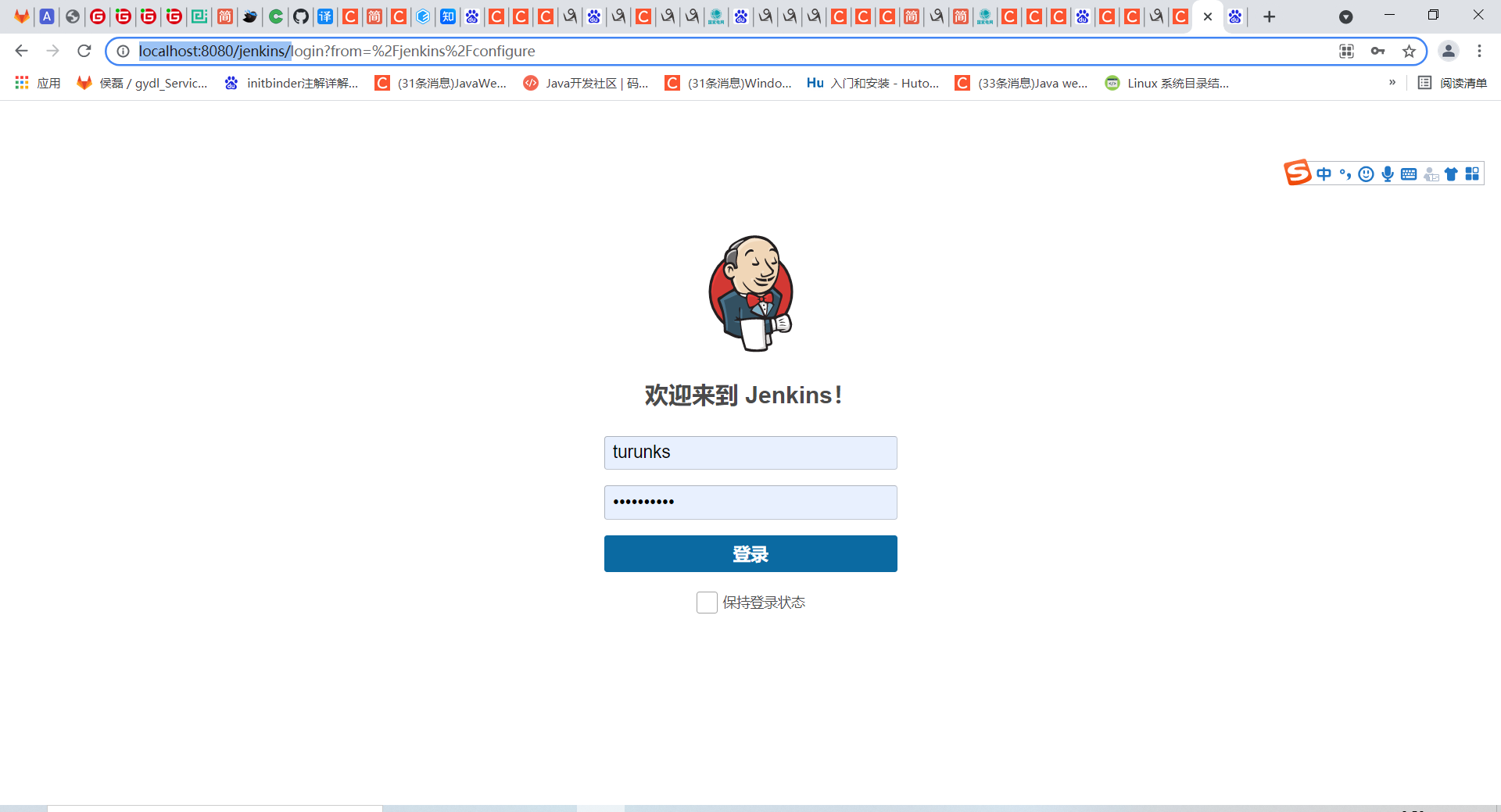
# 6.实操环节

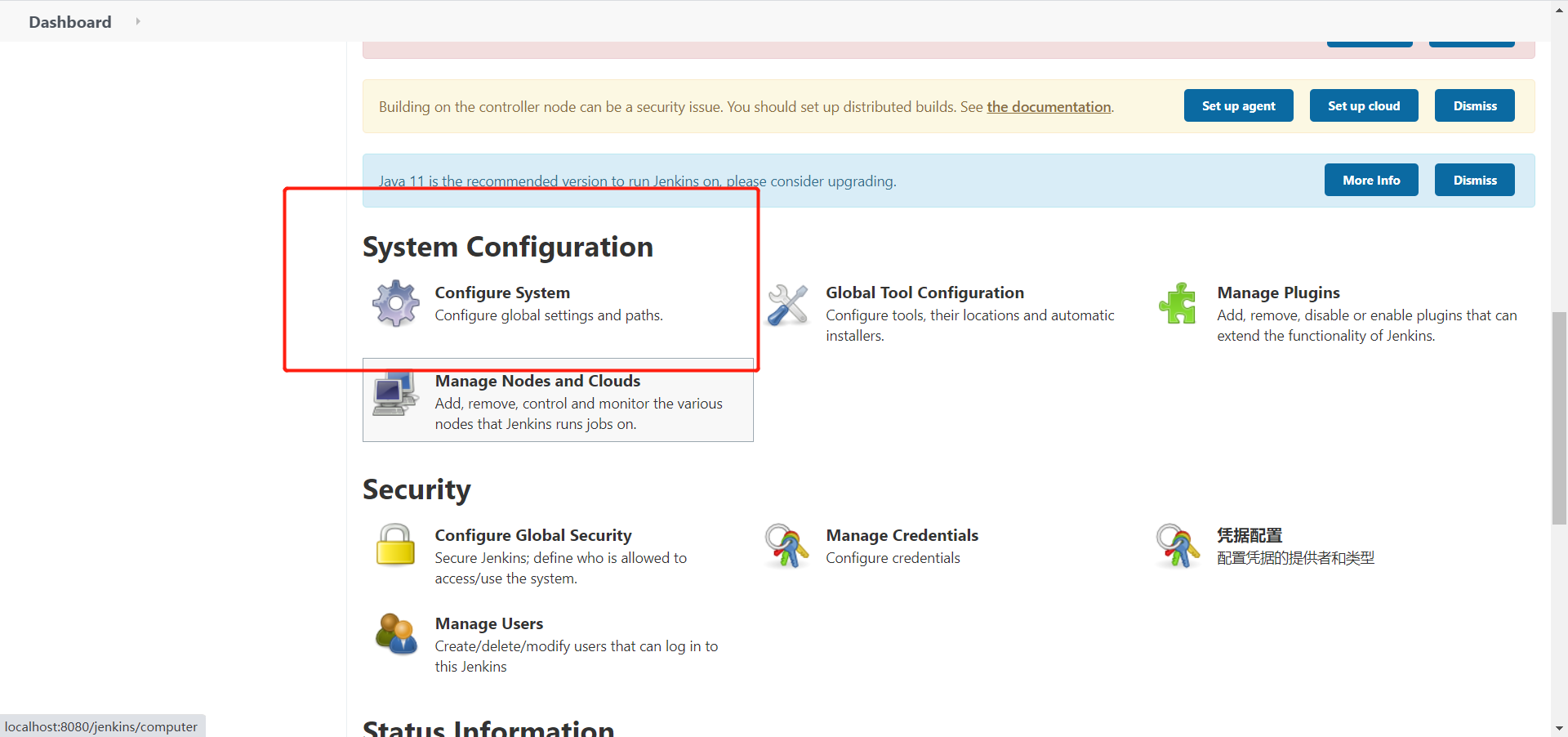


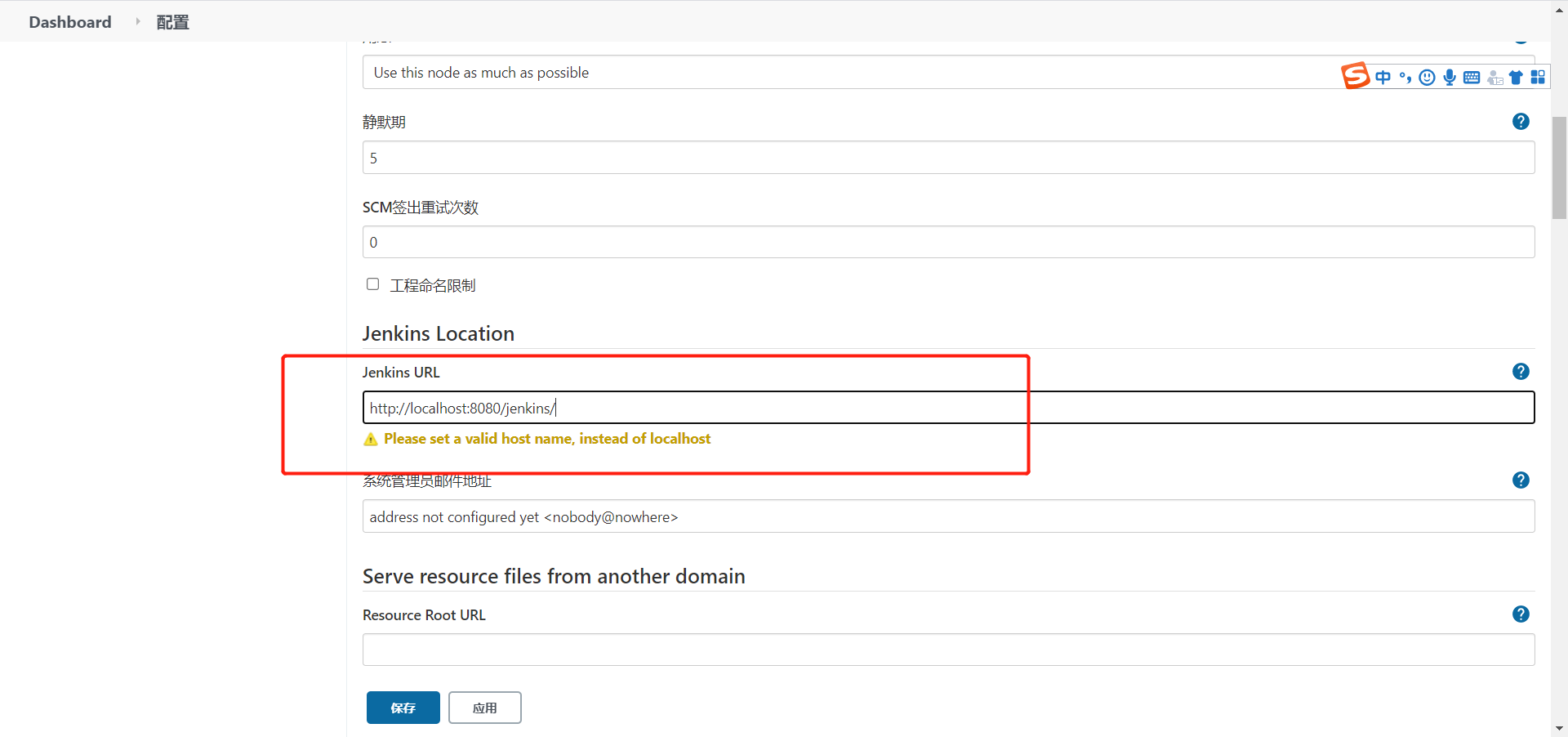
一．War放入tomcat启动

二．War放入tomcat启动

访问 <http://localhost:8080/jenkins/>





tomcat启动后访问地址为系统设置里的Jenkins Url

## 创建Job

1、创建项目

点击登录之后，会看到下边的界面中：My Views



