从零搭建 jenkins 自动化打包部署

|  |  |  |
| --- | --- | --- |
| 编写 | Zy | 2019.06.19 |
| 审核 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 变更人 | 变更日期 | 变更项 |
| V1.0 |  |  |  |

**目录**

1 搭建所需准备的工具及环境 1

1.1 linux 系统 1

1.2 java 环境 1

1.3 git 1

1.4 maven 1

1.5 tomcat 1

2 Jenkins 的安装及相关配置 3

2.1 下载及安装 jenkins 3

2.2 运行 jenkins 3

2.3 jenkins 的基础配置 6

3 服务器配置 7

3.1 git 配置 7

搭建所需准备的工具及环境

1.1 linux 系统

要确定以下几点

1. 确定 linux 的 ip 地址，通过 ifconfig 查询。
2. 确定登陆 linux 系统的用户名及密码。
3. 确定可以远程连接 ssh 服务，通过 netstat -anp| grep 22 查询，若出现类似下面的情况，说明已经开启

tcp 0 0 192.168.1.104:22 192.168.1.111:58653 ESTABLISHED 1706/sshd: root@pts

tcp 0 0 192.168.1.104:22 192.168.1.111:58662 ESTABLISHED 1853/sshd: root@pts

若没有开启，则通过 service sshd start 启动服务。

1. 确定防火墙处于关闭状态，通过 systemctl status firewalld 查询，若没有关闭，则通过 systemctl stop firewalld.service 关闭。

1.2 java 环境

自行安装，若没有，则通过 yum install java 安装，通过 java -version 检验。

1.3 git

自行安装，若没有，则通过 yum install git 安装，通过 git version 检验。

1.4 maven

有些项目在 maven 项目下的，因此也可以安装 maven。进入一个好找的文件夹，通过命令 wget http://mirror.bit.edu.cn/apache/maven/maven-3/3.6.1/binaries/apache-maven-3.6.1-bin.tar.gz 进行下载。通过 “tar -zxvf apache-maven-3.6.1-bin.tar.gz -C .” 进行解压到当前文件夹。进入解压后的文件夹，通过 pwd 查询当前目录的路径并进行复制。之后通过 vim /etc/profile 在最后添加以下内容

# maven path， M2\_HOME 之间有下划线，格式原因没有显示出来。

export M2\_HOME=/usr/local/maven

export PATH=$PATH:$M2\_HOME/bin

export MAVEN\_OPTS='-XX:PermSize=1024m -XX:MaxPermSize=1024m'

/usr/local/maven 为刚复制的路径。添加并保存之后，通过 source /etc/profile 将文件重新生效。最后通过 mvn -version 进行验证，有版本号出现则说明已经安装成功。

1.5 tomcat

部署的程序是部署在tomcat 上的，通过命令 wget http://mirror.bit.edu.cn/apache/tomcat/tomcat-9/v9.0.21/bin/apache-tomcat-9.0.21.tar.gz 进行下载。通过命令 “tar -zxvf apache-tomcat-9.0.21.tar.gz -C .” 进行解压到当前文件夹。

进入解压后的文件夹，输入命令 chmod a+x -R \* ，将该目录下的文件赋予可执行权限。Tomcat 的默认端口为 8080，但这容易和别的应用发生冲突。因此修改 tomcat 配置文件的端口。打开当前解压后的文件夹下的配置文件 vim conf/server.xml，找到 Connector port="8080" ，将其修改为 8090。

现在尝试运行 tomcat，依旧在解压后的文件夹中输入 bin/startup.sh， 稍等一会，输入命令 ps -ef | grep tomcat 查看是否有启动，最后在浏览器中输入虚拟机的IP + 在 tomcat配置文件中修改后的端口，查看是否能够打开页面。

Jenkins 的安装及相关配置

2.1 下载及安装 jenkins

登录网址 <https://jenkins.io/zh/>，点击下载：

图2.1 jenkins 主界面

下载 war:

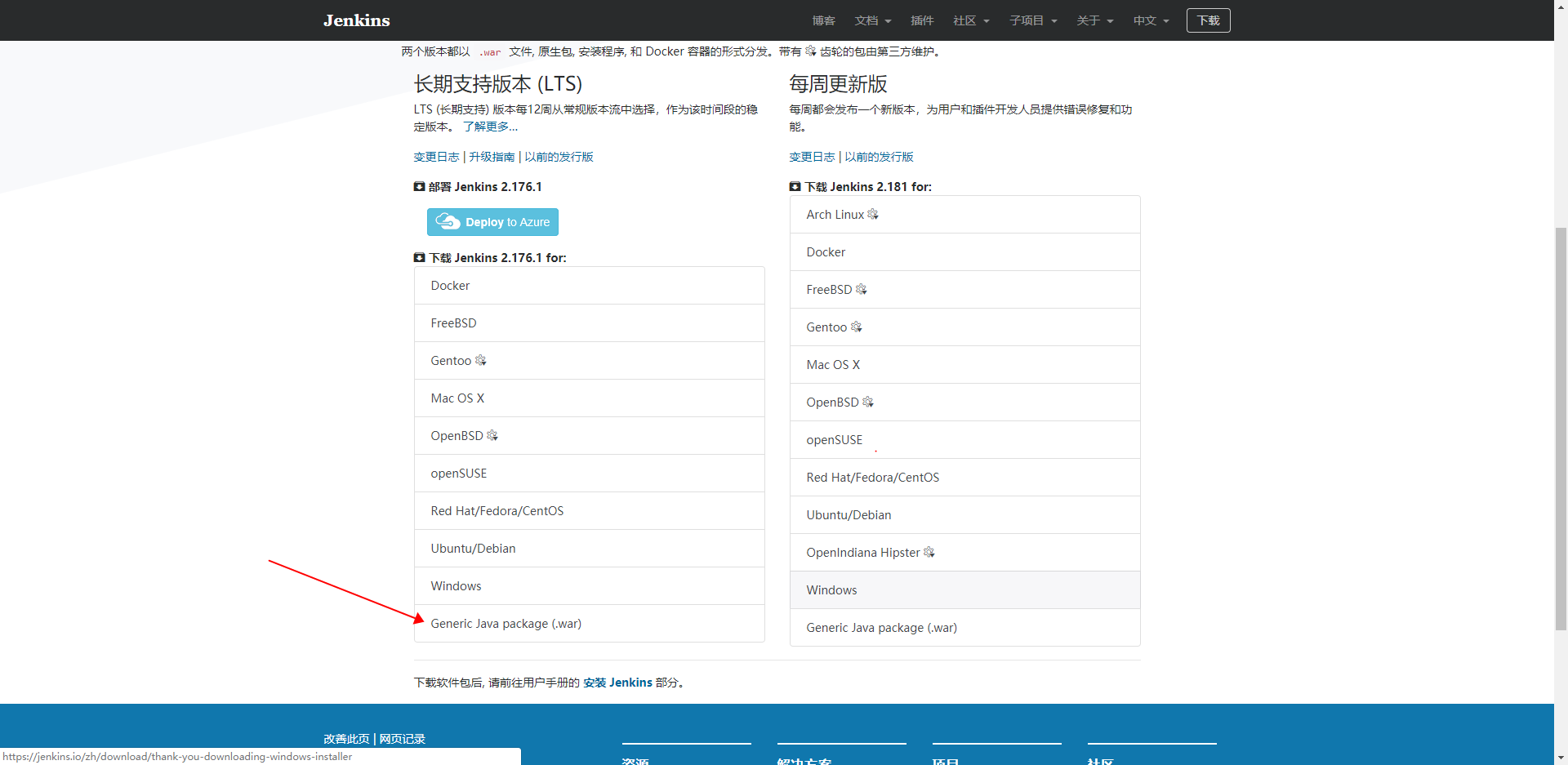


图2.2 下载 war 界面

2.2 运行 jenkins

在 linux 系统中，通过命令 java -jar jenkins.war 直接启动，如果出现插件下载失败相关的报错信息，可以采取一个暴力的方法，将 /root/.jenkins 下的所有文件全部删掉，当再次启动 jenkins 时会自动重新下载刚被删掉的文件的原始文件。

启动成功之后，可以打开浏览器，输入 [loacalhost:8](http://192.168.0.124:9999/login?from=/)080，有些虚拟机需要输入虚拟机的 ip + 端口号才能打开。

打开后为该界面：

图2.3 输入管理员密码



查看 /root/.jenkins/secrets/initialAdminPassword 下的信息，如果不存在，通过命令 find / -name initialAdminPassword，找到文件对应的位置，然后查看内容。把查看到的内容复制到页面上管理员的密码，然后点击“继续”。

进入页面，点击“安装推荐的插件”：

图2.4 安装插件

会出现以下界面：



图2.5 自动安装插件

安装插件完成之后，会进入到创建管理员界面：

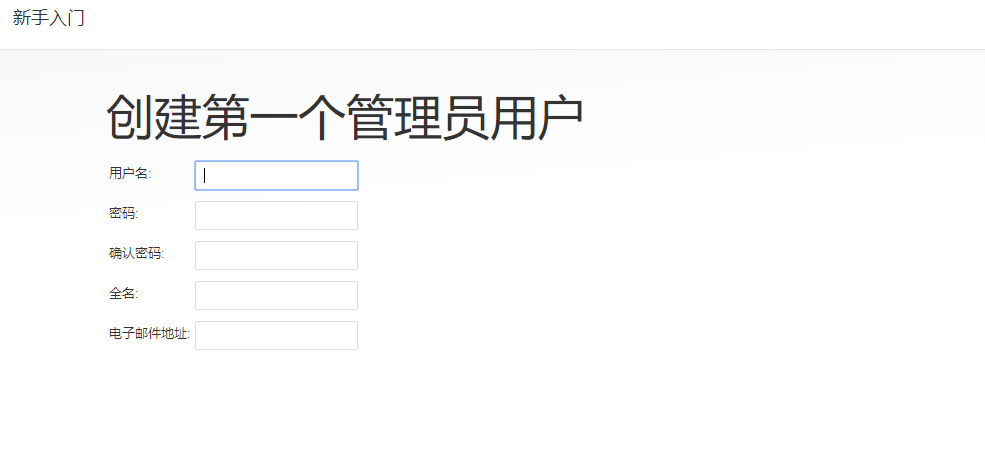


图2.6 创建管理员

用户名和密码需要记住，以后都是以此进行登录的。

之后配置 url，如图2.7所示。



图2.7 配置 url

2.3 jenkins 的基础配置

2.3.1 安装插件



图2.8 主界面

图2.8 为主界面，依次点击系统管理、插件管理、可选插件、在过滤出输入 rebuilder、勾选之后点击“直接安装”，同理安装第二个插件 safe restart plugin。当第一次进入该界面时，可能存在一些标题为英文的情况，原因是缺少插件，因此可以先安装 Maven Intrgration plugin 插件。安装完毕之后，点击“重启”。等待 jenkins 重启即可生效插件，可能存在停留在插件界面并提示在重启中，若出现该情况，则点击“返回首页”，即可跳转到登陆界面。

服务器配置

3.1 git 配置

3.1.1 配置 git 的用户名和 email

在linux 中通过命令 git config --global user.name ["xxxx"](mailto:\"z648883413@163.com\") 和 git config --global user.email ["xxxx@163.com"，不做强制要求为真实信息，存在即可。](mailto:\"xxxx@163.com\"，不做强制要求为真实信息，存在即可。)

3.1.2 授权证书

通过命令 ssh-keygen -t rsa -C ["xxxx@163.com"，email](mailto:\"xxxx@163.com\"，email) 地址为你之前填写的 email 一致。随之系统提示输入用户名、密码、确认密码，这时直接回车，不需要输入。

生成之后确认一下是否有生成，进入 ~/.ssh/ 目录，查看当前目录文件，会出现 id\_rsa 和 id\_rsa.pub，前者为私钥，后者为公钥。有则证明已经生成成功。

3.1.3 linux 与 github 相通

登录 github，点击右上角的头像之后，再点击 settings

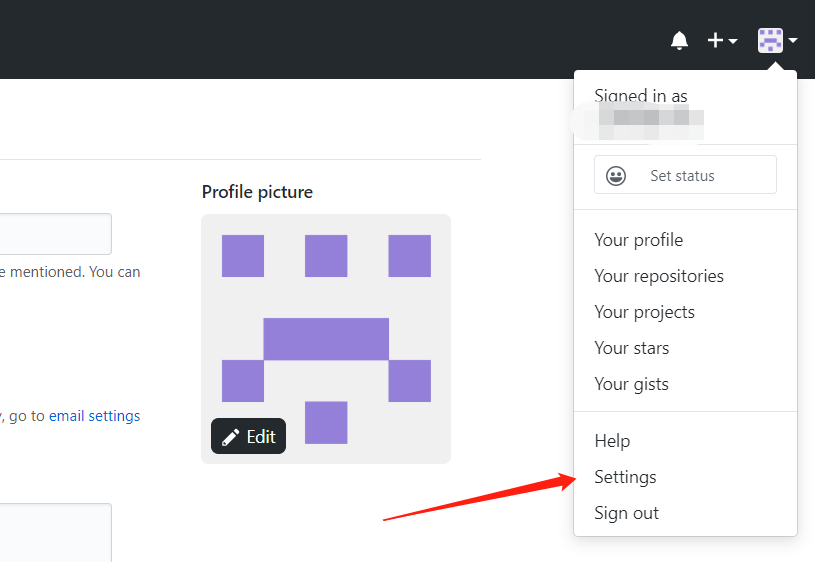


图3.1 进入配置

之后点击“SSH and GPG keys”，再点击“New SSH key”

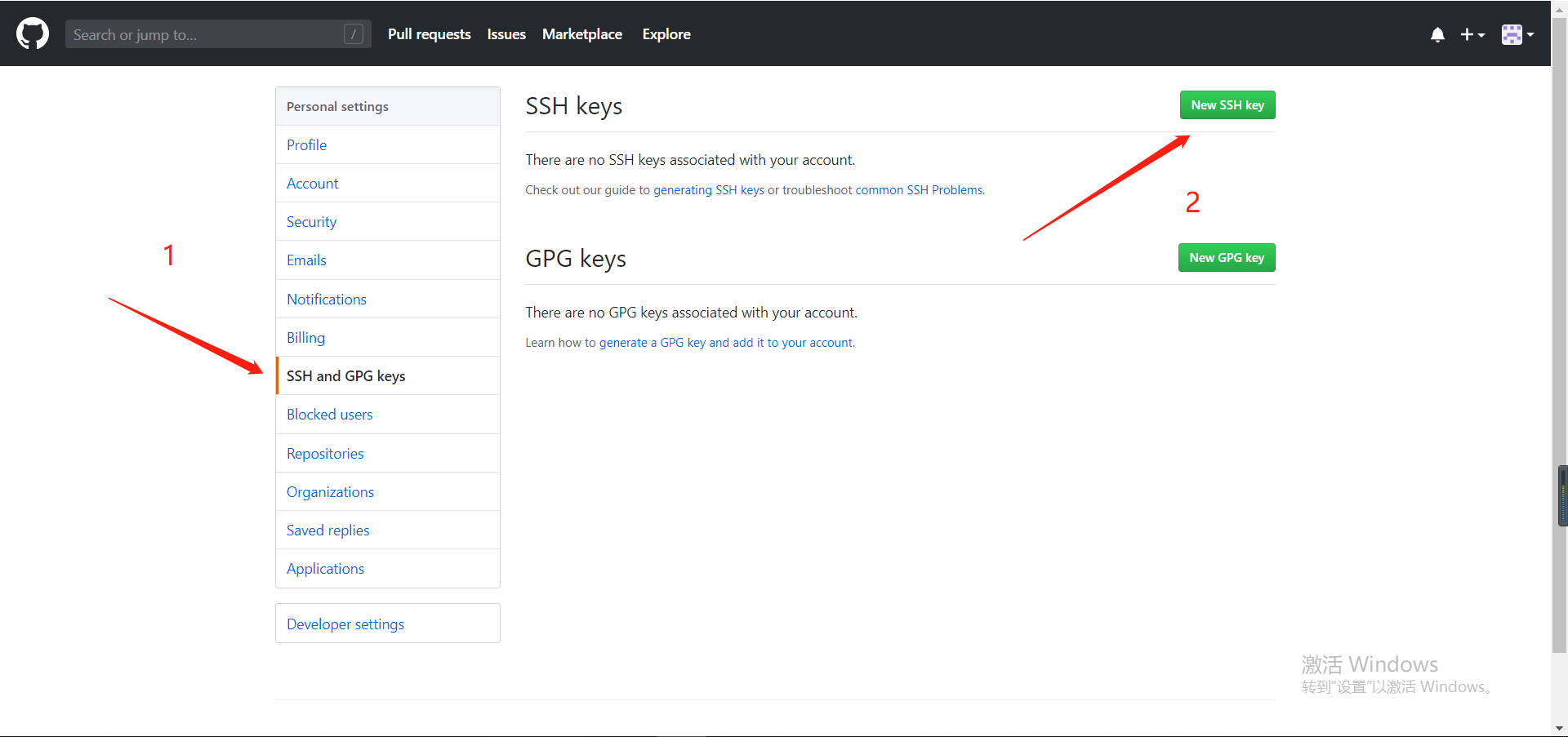


图3.2 进入 ssh 密钥添加

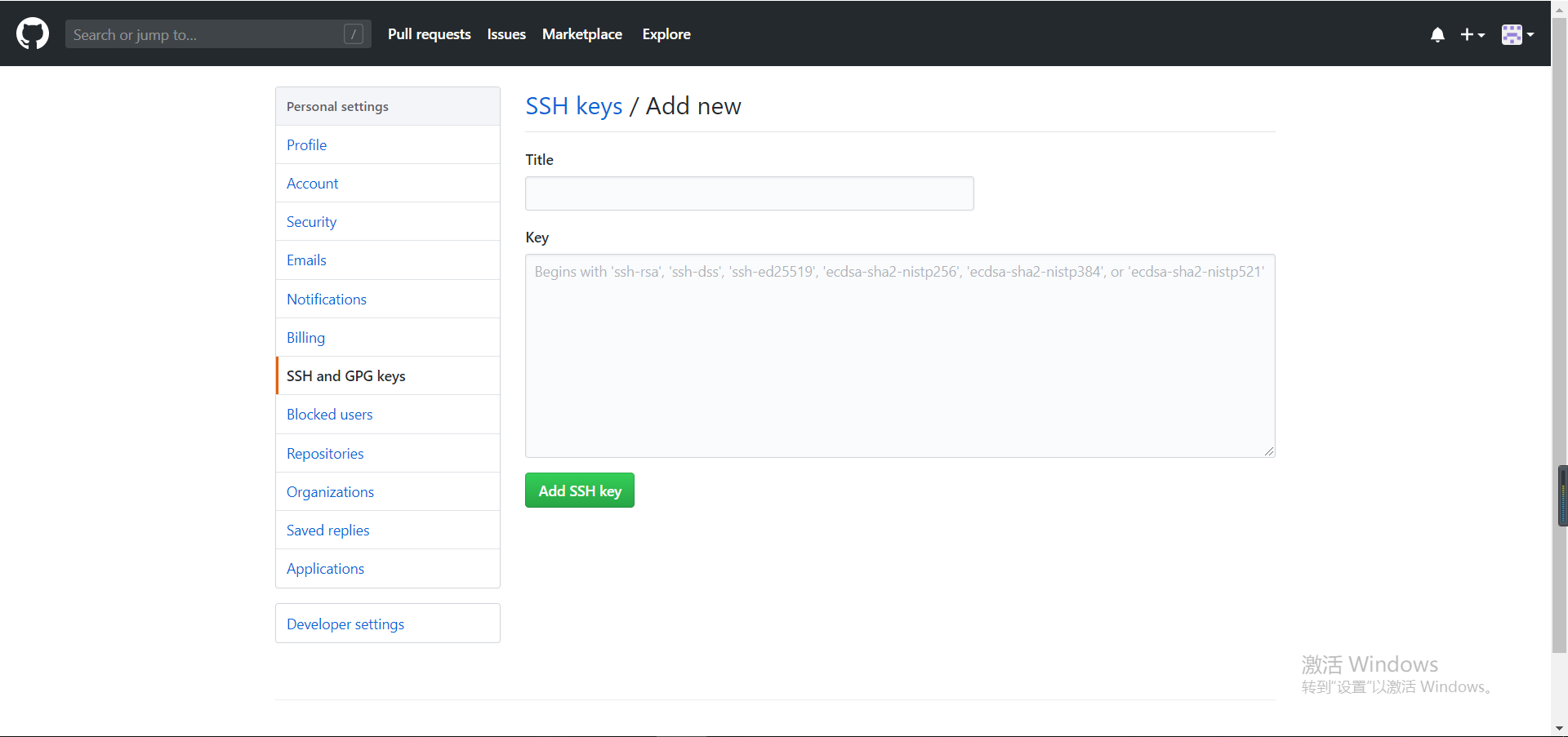


图3.3 添加密钥

进入图3.3 界面，title 自行输入，然后将刚 linux 下生成的公钥中的所有信息复制之后粘贴到 key 中。最后点击“Add SSH key”。

添加完毕之后，要验证 linux 是否与 github 相通。在 linux 中输入 ssh [git@github.com，之后会跳出提示语询问填写](mailto:git@github.com，之后会跳出提示语询问填写) yes or no，这时填写 yes。当出现“Hi xxxx! You've successfully authenticated, but GitHub does not provide shell access.”时说明已经连接成功。