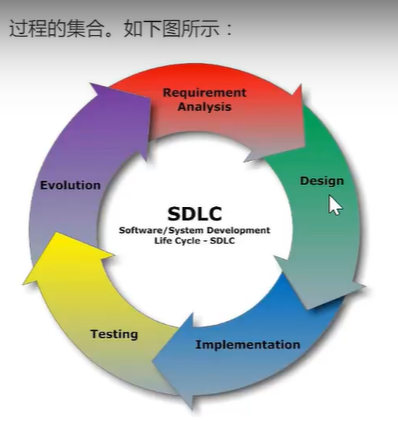
持续集成



软件开发生命周期：需求分析、设计、实现、测试、进化

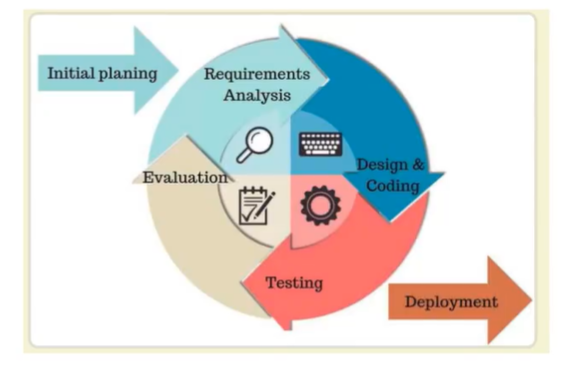
软件开发瀑布模型：最著名和最常使用的软件开发模型，是由制造业繁衍出来的。

优势：简单易用和理解，当完成一阶段后，只需要关注后续阶段，为项目提供阶段划分的检查节点

劣势：各个阶段的划分完全固定，阶段之间产生大量的文档，极大地增加了工作量；

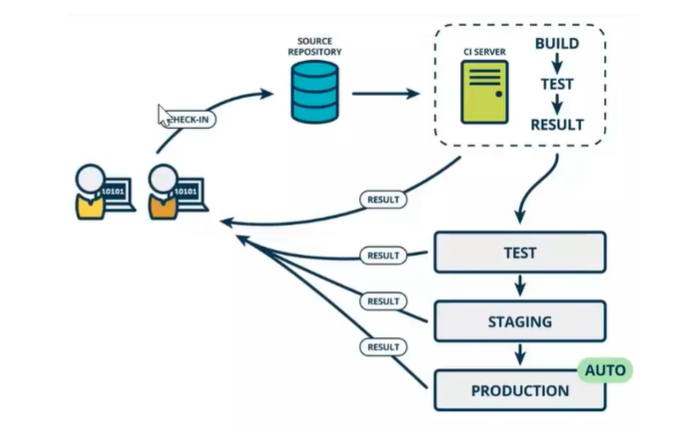
用户只能等到开发末期才能看到成果，风险较大；不适应用户需求的变化

敏捷开发：核心是迭代开发和增量开发



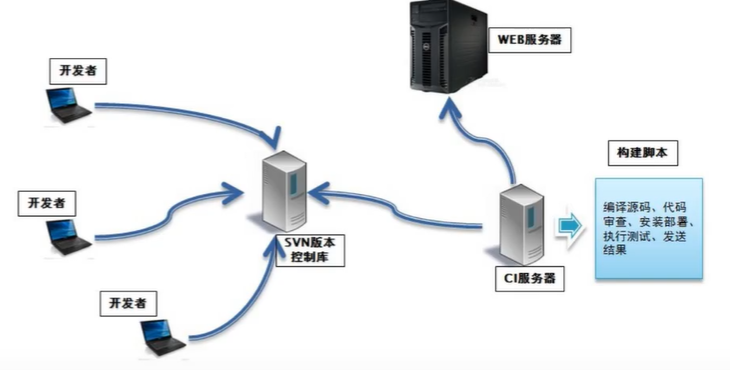
优势：早期交付、降低风险

持续集成：频繁地将代码集成到主干，目的是让产品可以快速迭代，同时还能保持高质量。



环节：提交、测试（第一轮）、构建、测试（第二轮）、部署、回滚

组成要素：构建过程完全自动化、代码存储库、持续集成服务器



优势：降低风险、减少发布风险、减少重复性工作、持续部署、持续交付可供使用的版本、增强团队信心

Jenkins特征：开源、易于安装、信息通知及测试报告、分布式构建、文件识别、丰富的插件、基于java编写的软件

Jenkins安装和持续集成环境配置

部署安装分别gitlab一台机器，jenkins一台机器，测试与用户一台机器

如果是虚拟机，则尽量分配高一点的配置，不然gitlab等会很慢

使用gitlab作为代码管理仓库

安装相关依赖：

yum -y install policycoreutils openssh-server openssh-clients postfix

启动ssh服务&设置为开机启动：

systemctl enable sshd && sudo systemctl start sshd

设置postfix开机自启，postfix支持gitlab发信功能：

systemctl enable postfix && sudo systemctl start postfix

开放ssh以及http服务，然后重载防火墙列表

firewall-cmd --add-service=ssh –permanent

firewall-cmd --add-service=http –permanent

firewall-cmd –reload

下载gitlab包，并安装

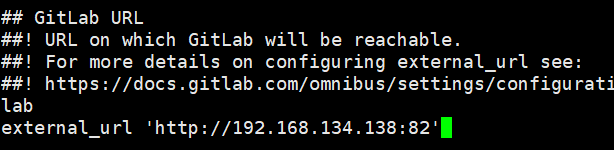
Wget <https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el6/gitlab-ce-12.4.2-ce.0.el6.x86_64.rpm>

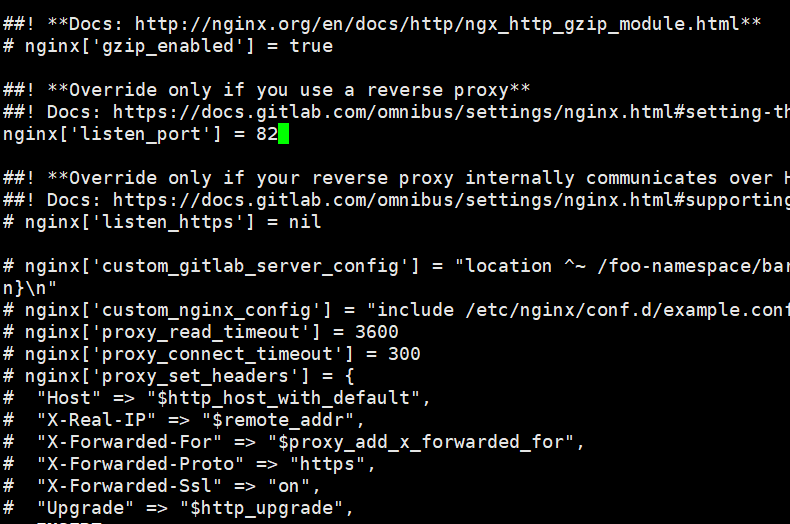
Rpm –I [下载好的gitlab.rpm]

修改gitlab配置

Vim /etc/gitlab/gitlab.rb

修改gitlab访问地址和端口，默认为80，改为82





重载配置和启动gitlab

Gitlab-ctl reconfigure

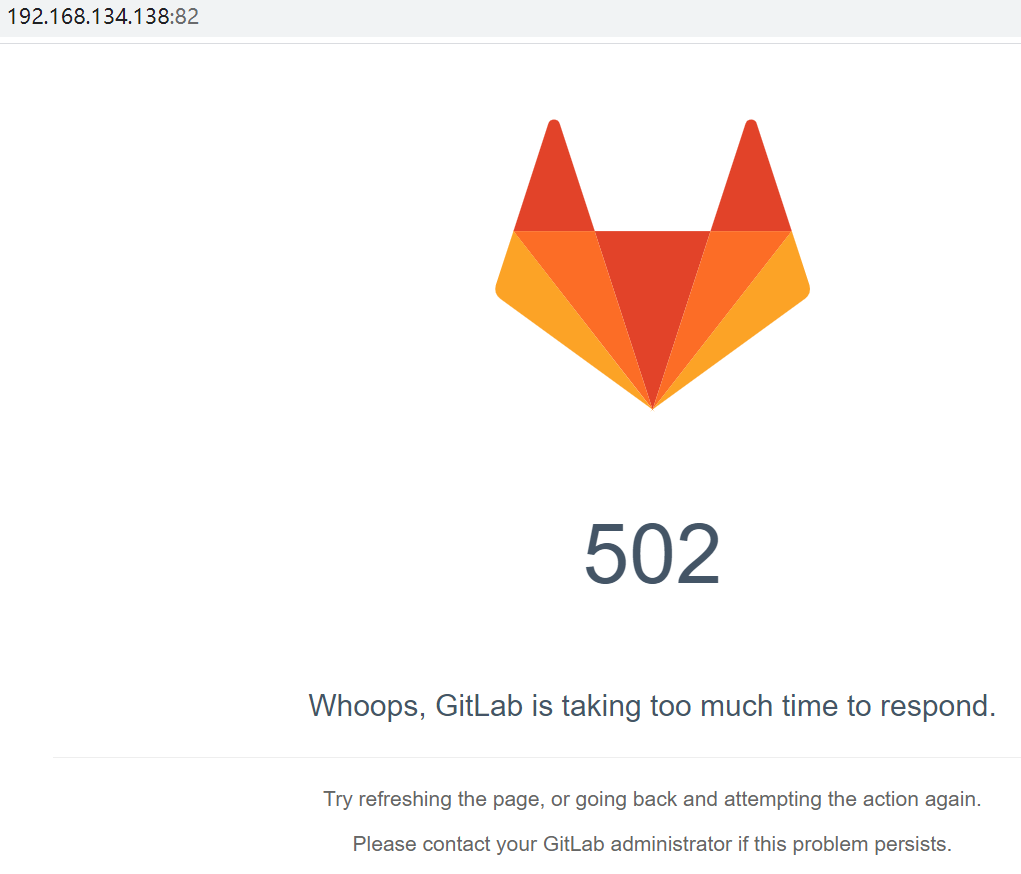
Gitlab-ctl restart

把端口添加到防火墙

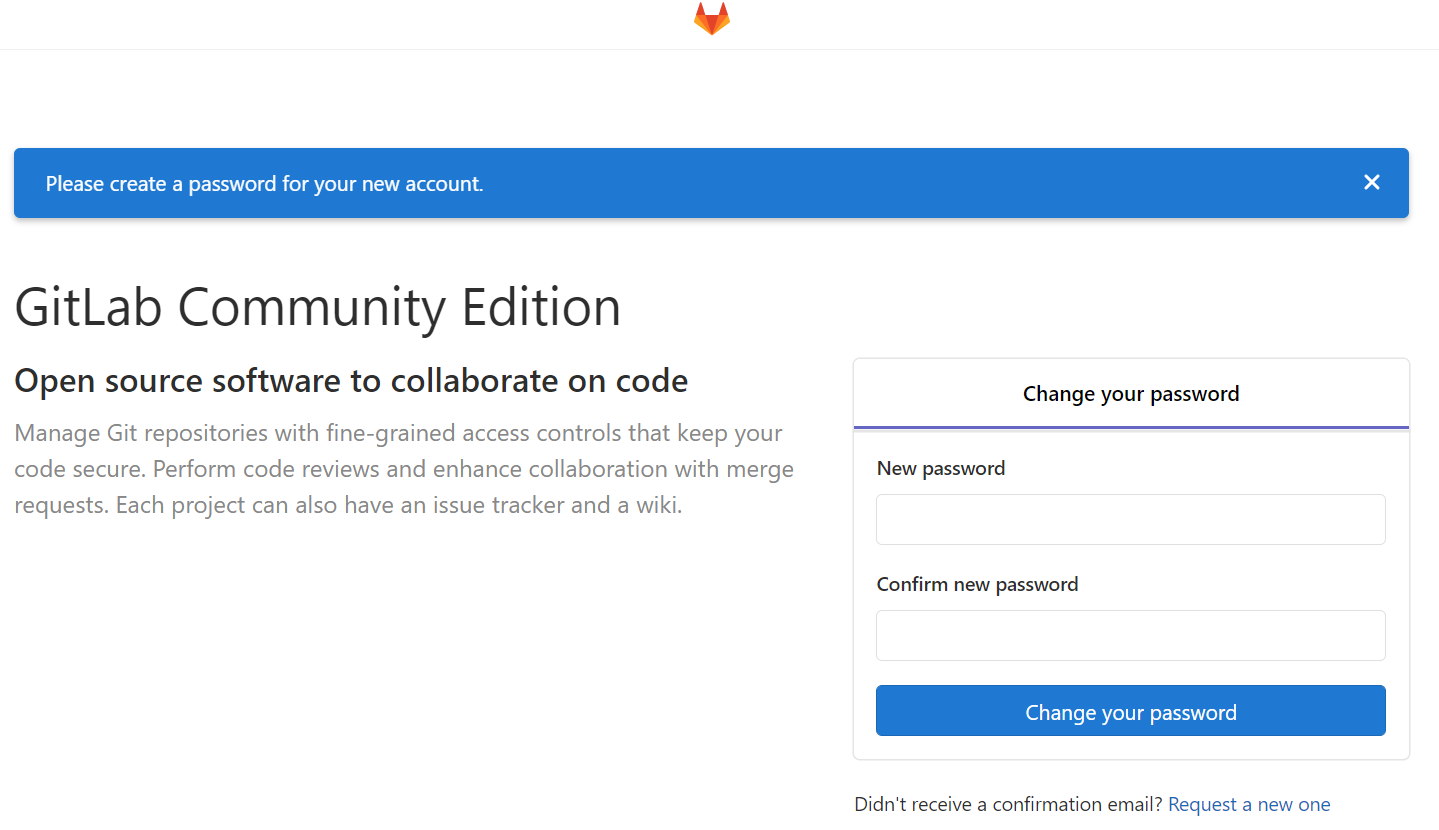
firewall-cmd --zone=public --add-port=82/tcp –permanent

firewall-cmd --reload

准备完成，在宿主机访问

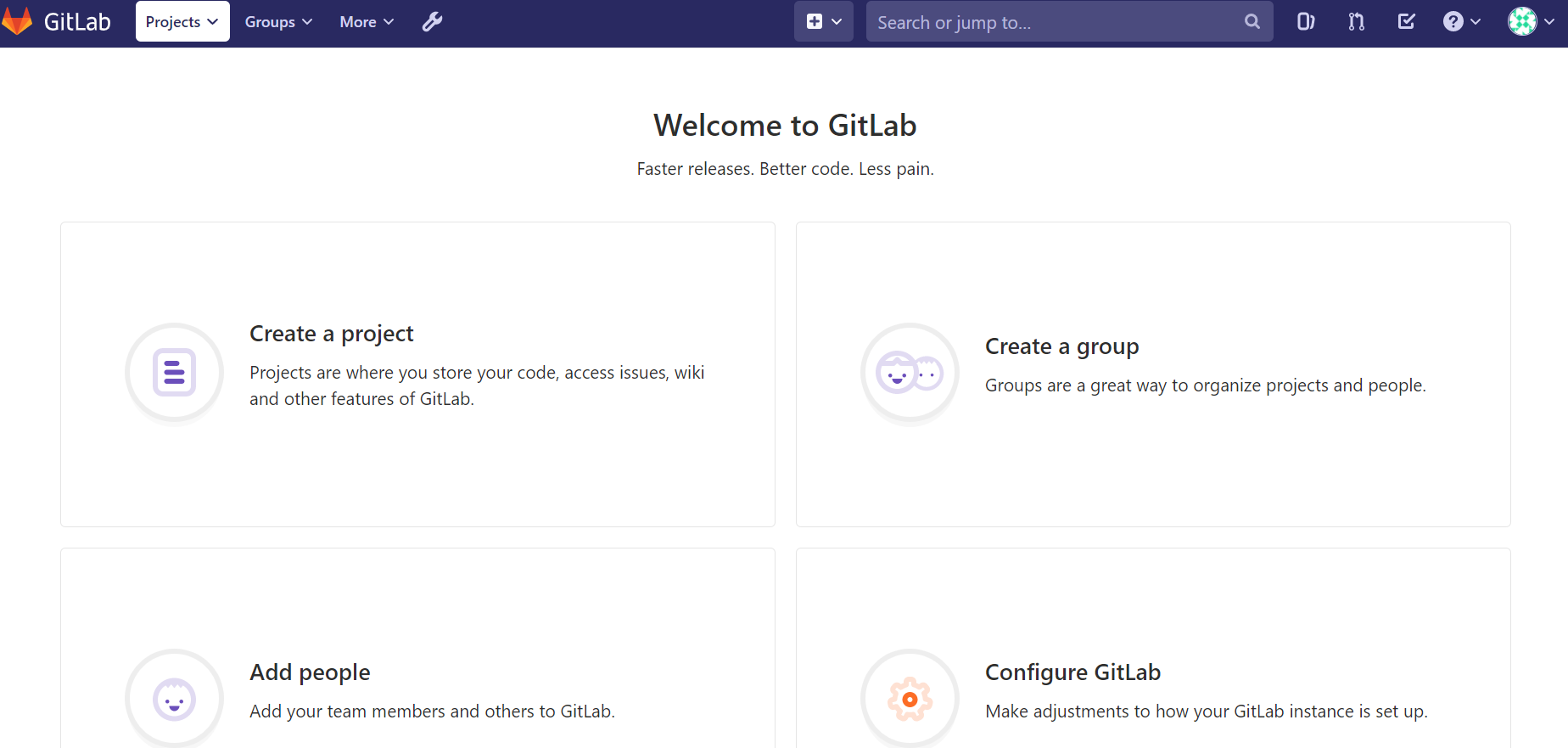


等待一会刷新页面



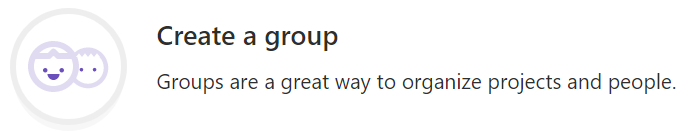
Password：123456789

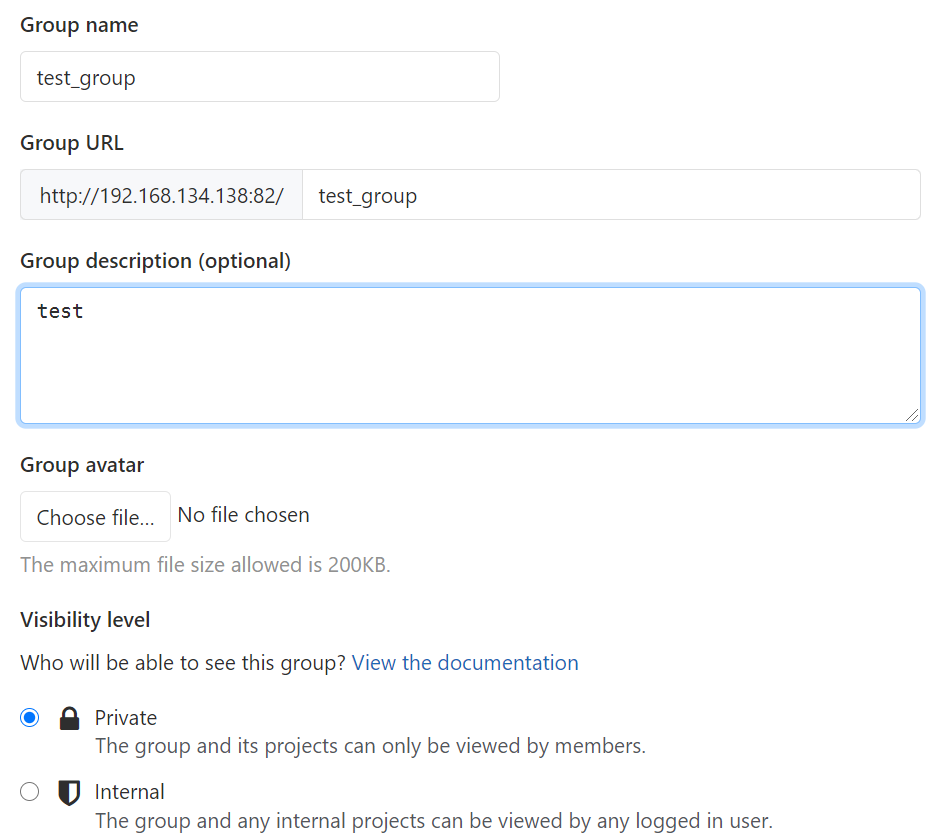
登录成功·

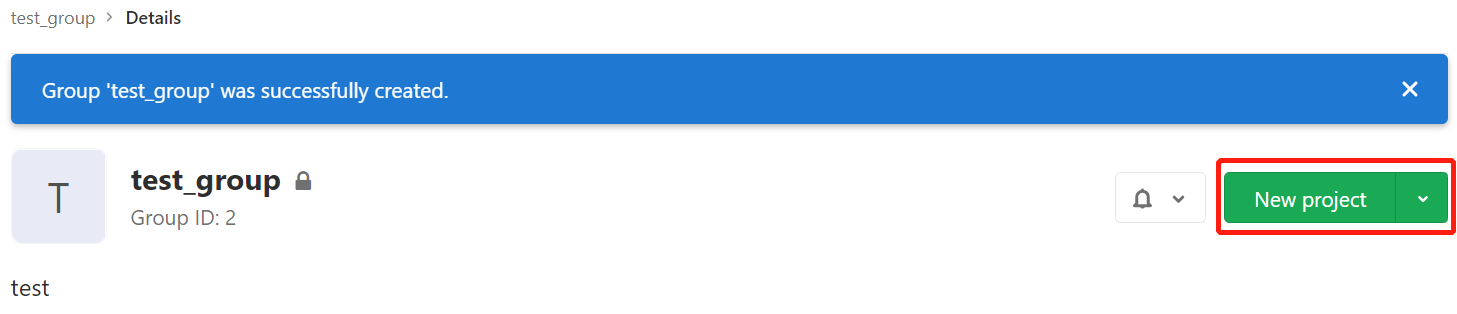


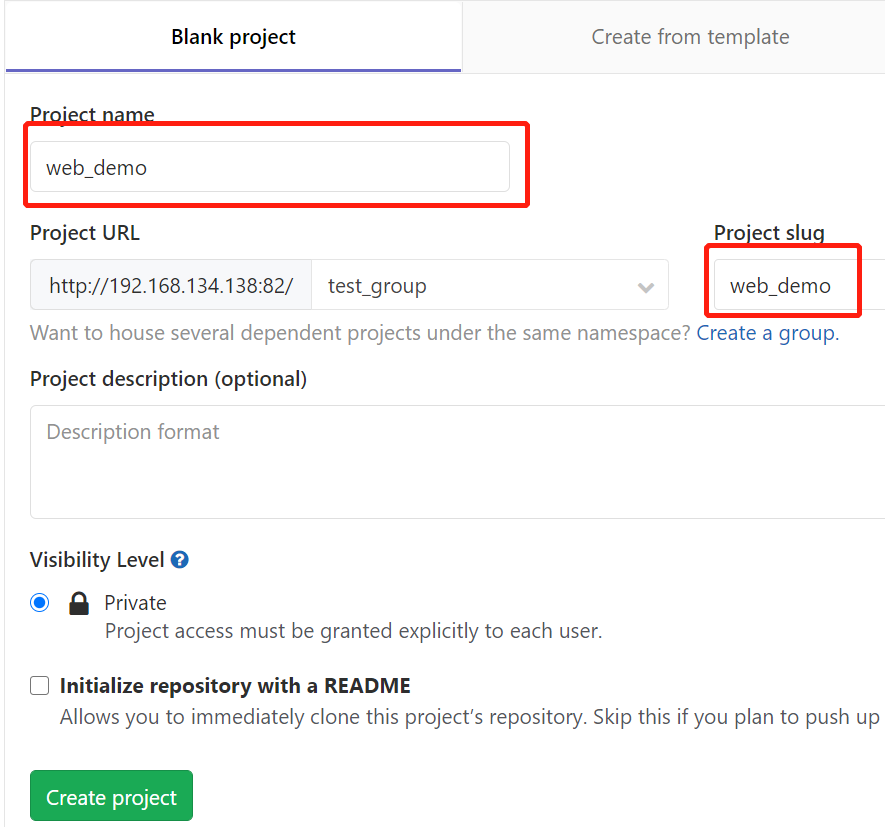
Gitlab添加组、创建用户、创建项目

Root登录，点击create a group

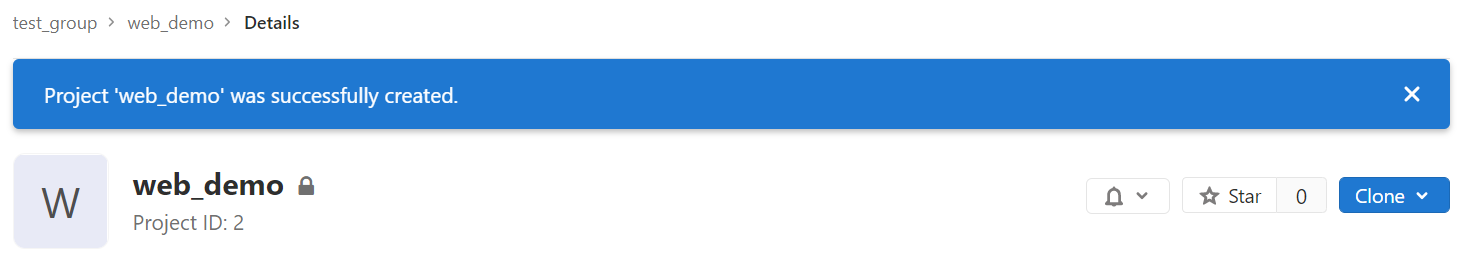




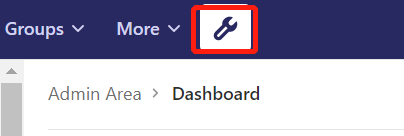


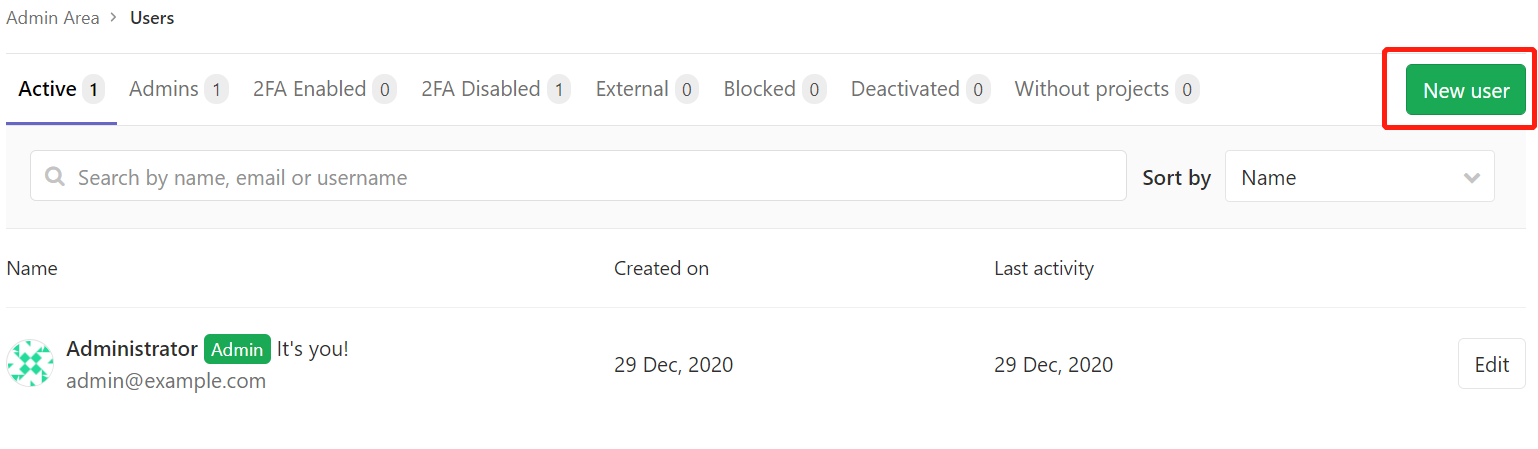


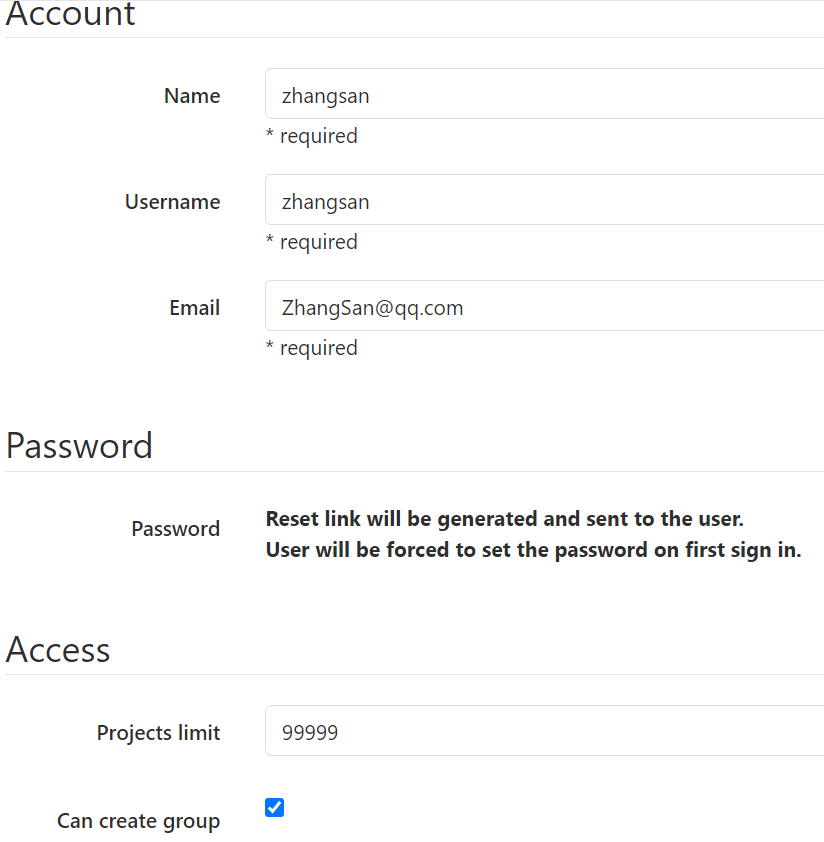
创建project成功



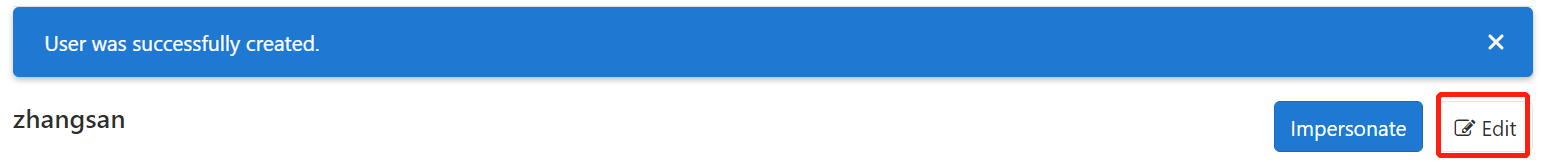
创建用户，点击users

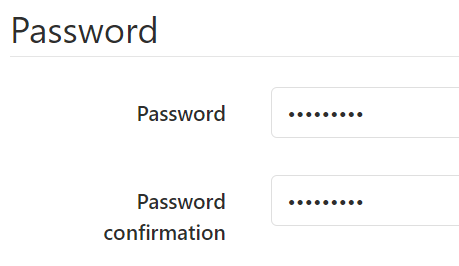






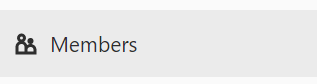
添加密码

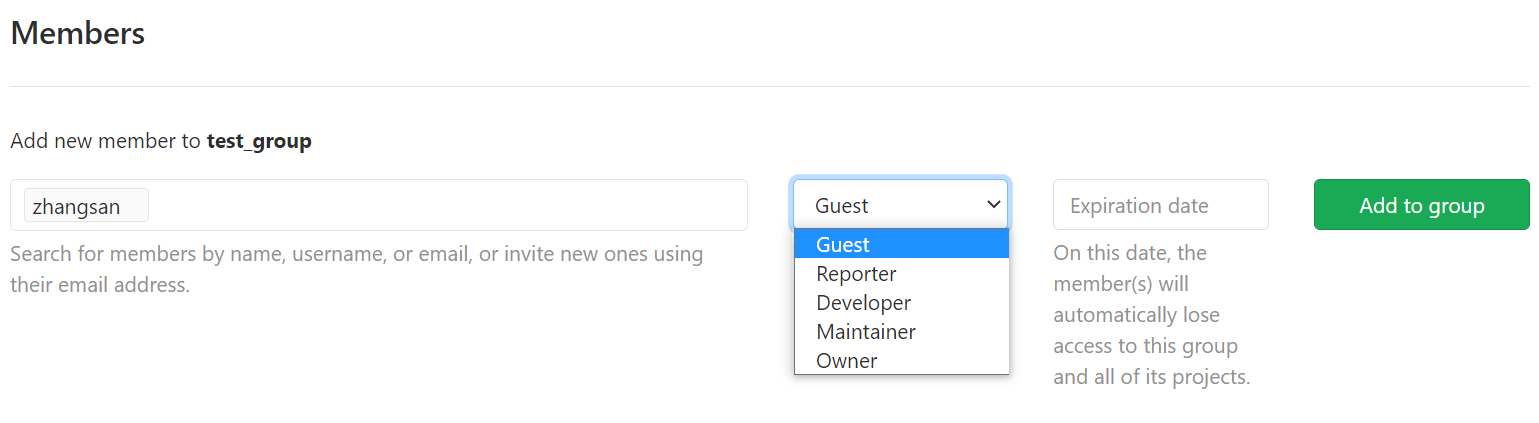




123456789

添加用户到组





Guest：可以创建issue、发表评论、不能读写版本库

Reporter：可以克隆代码、不能提交、QA、PM可以赋予这个权限

Developer：可以克隆代码、开发、提交、push、普通开发可以给这个权限

Maintainer：可以创建项目、添加tag、保护分支、添加项目成员、编辑项目、核心开发可以给这个权限

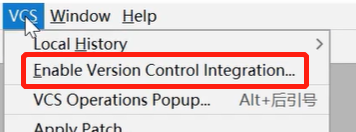
Owner：可以设置项目访问权限-Visibility Level、删除项目、迁移项目、管理组成员、开发组组长可以给这个权限

然后即可使用zhangsan这个账号登录

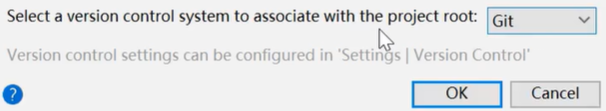
上传源码到gitlab

打开idea或pycharm等软件

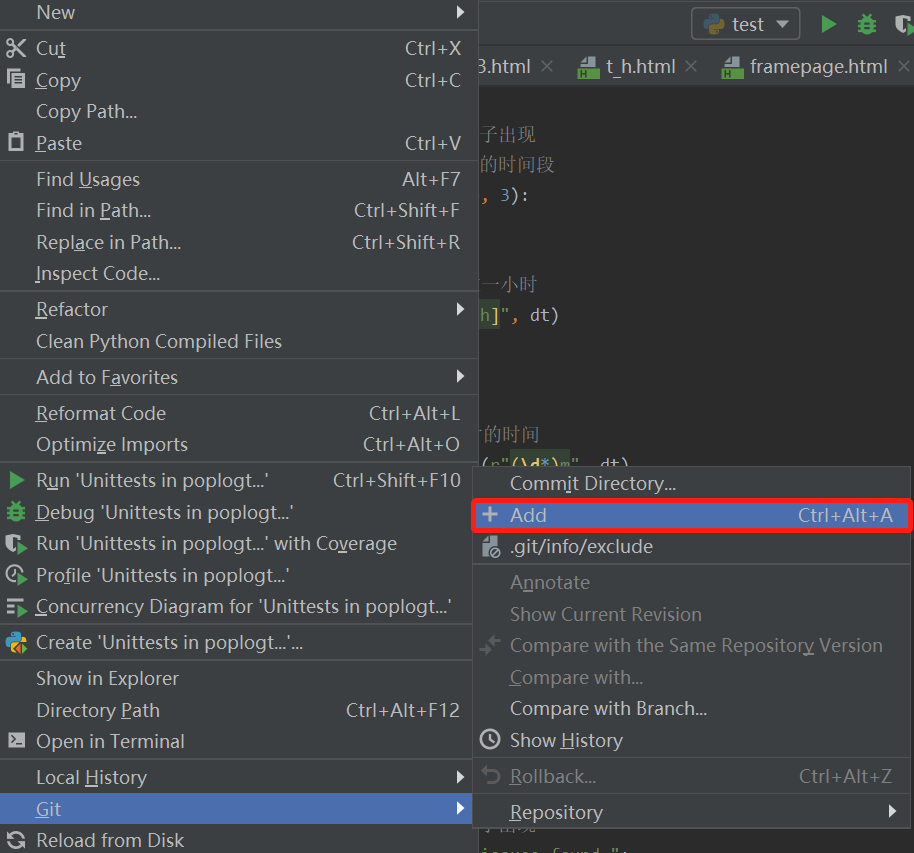




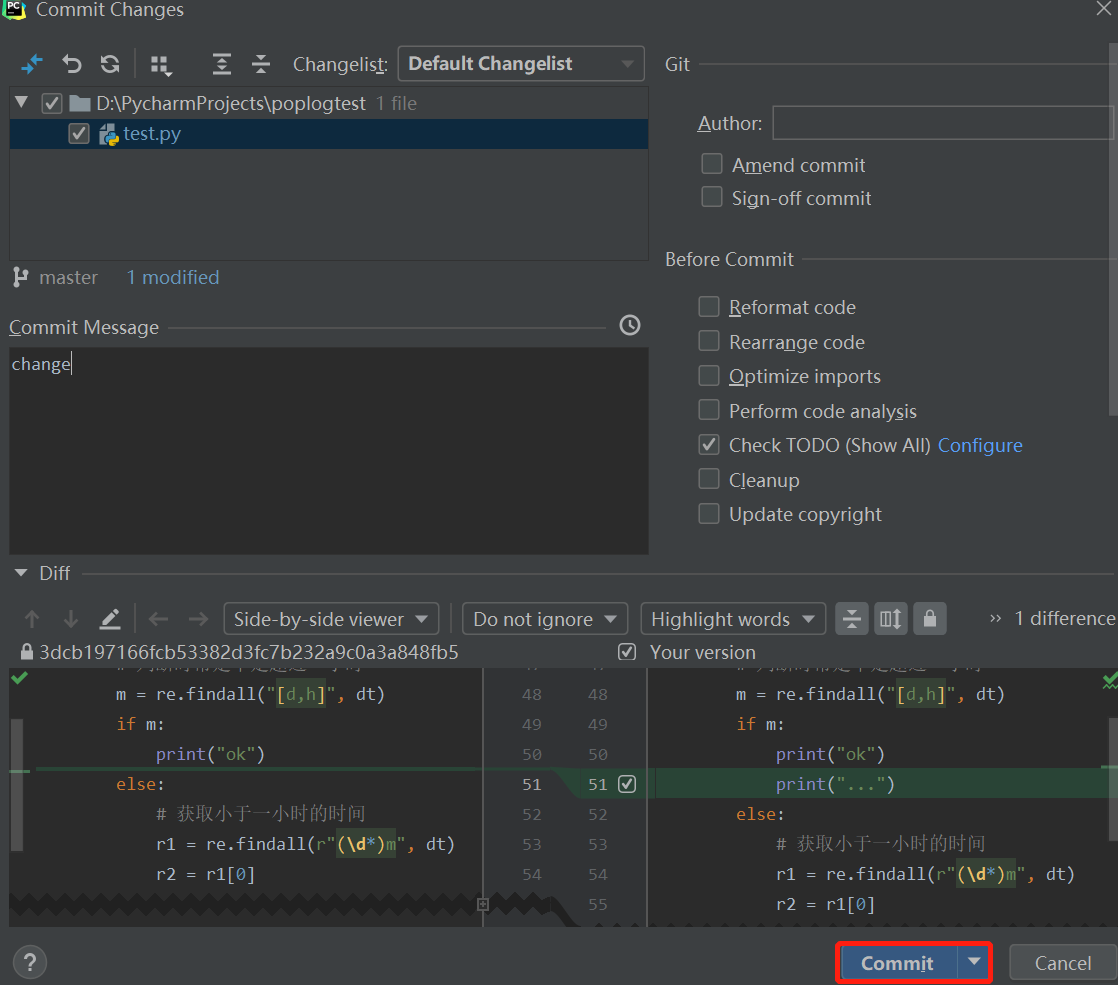
选择git类型



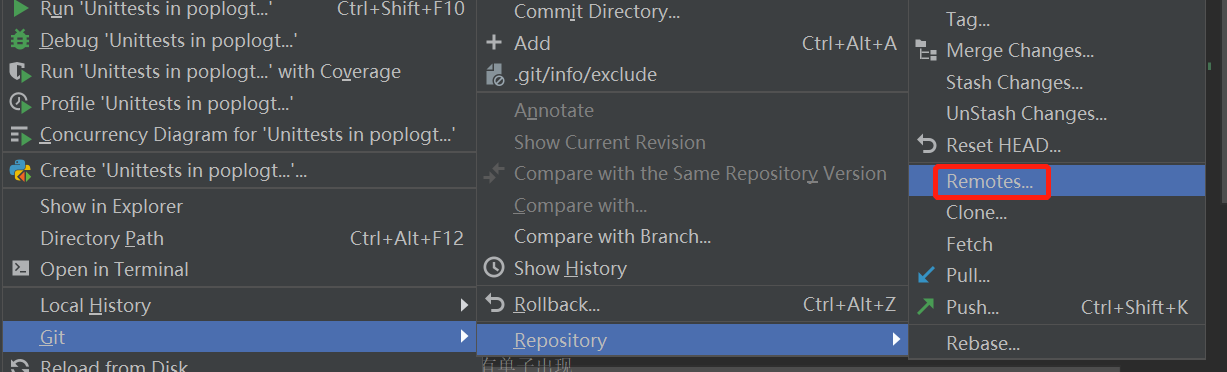
添加项目缓存

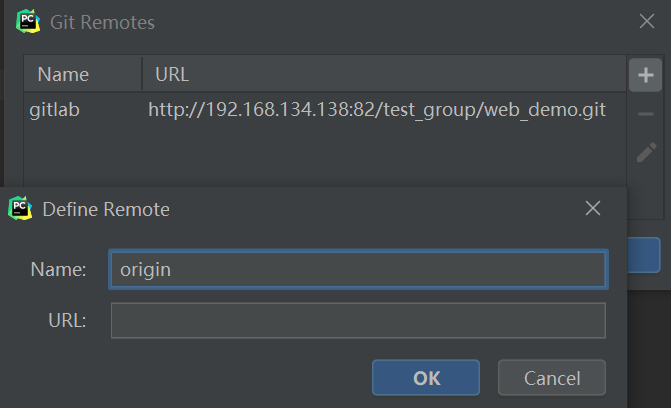


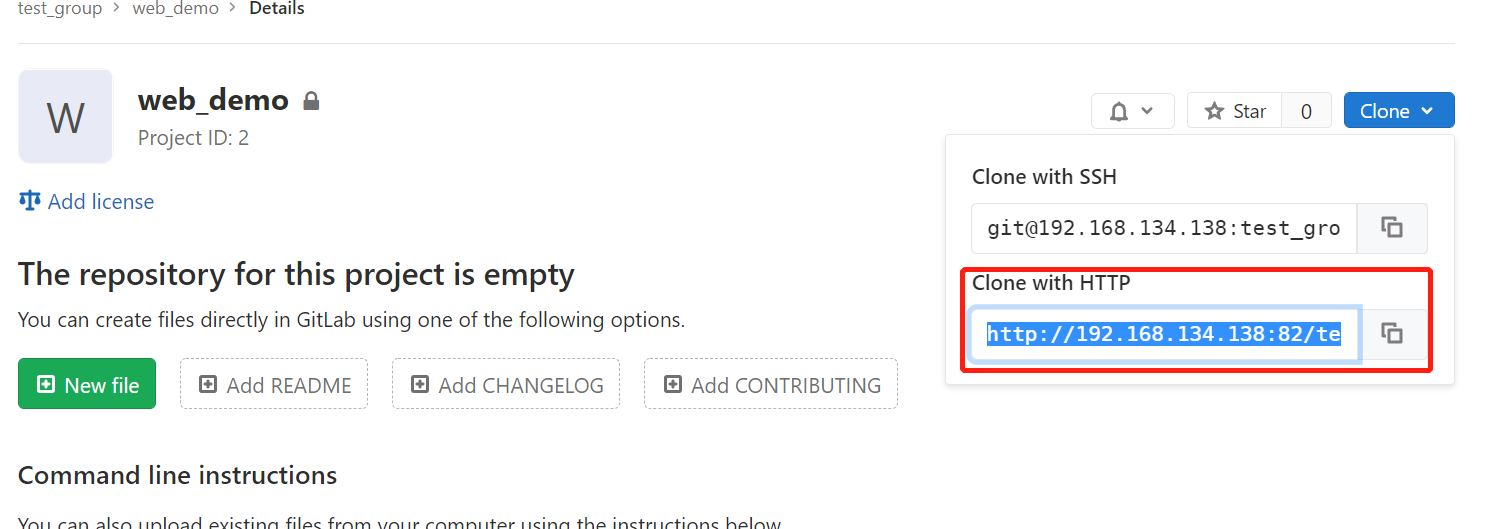
提交更改到本地项目



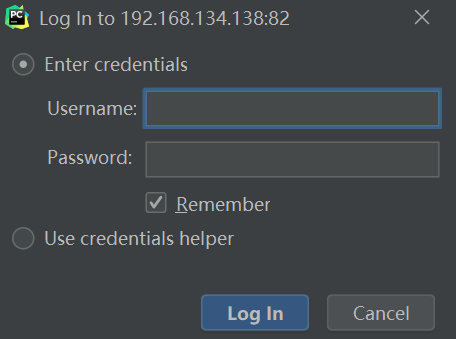
配置仓库url



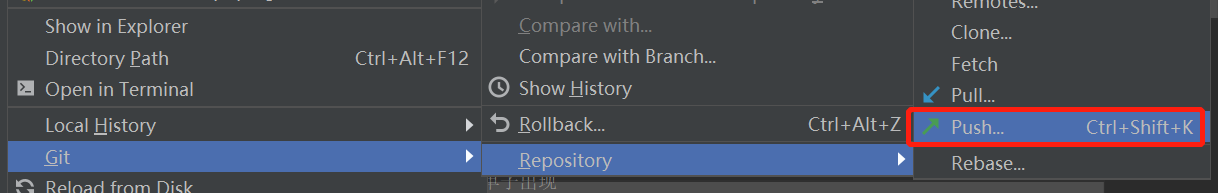


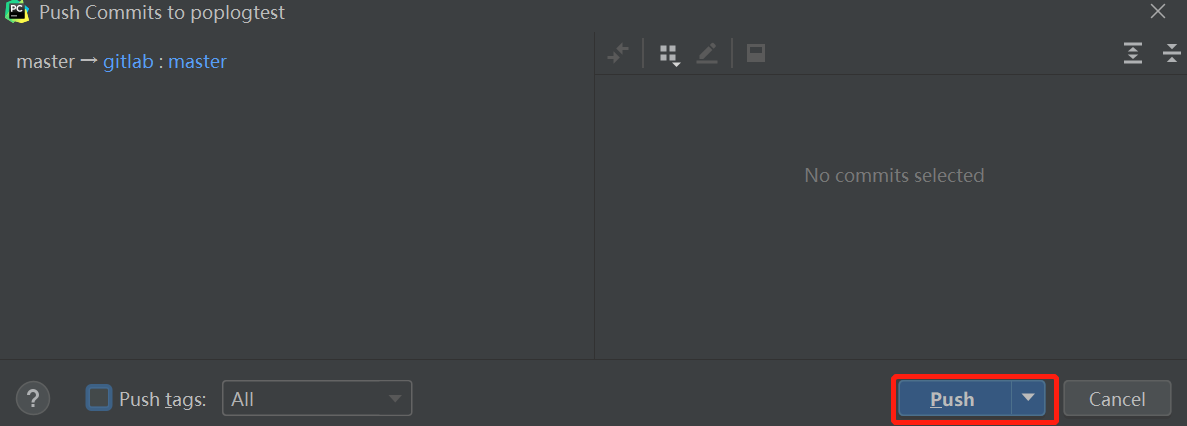


输入gitlab上的用户名和密码

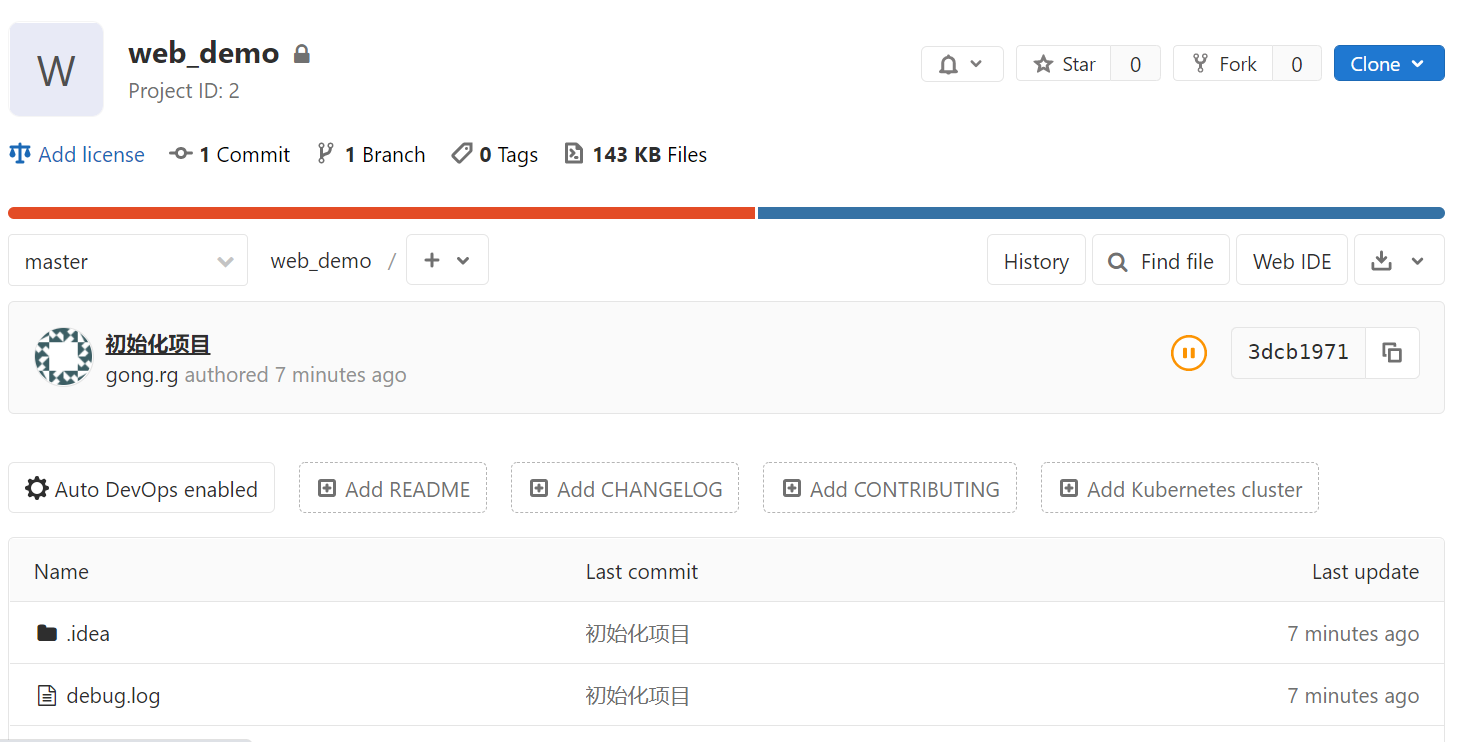


最后push上去





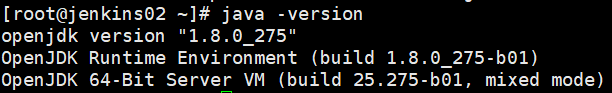
刷新gitlab项目即可看到跟新



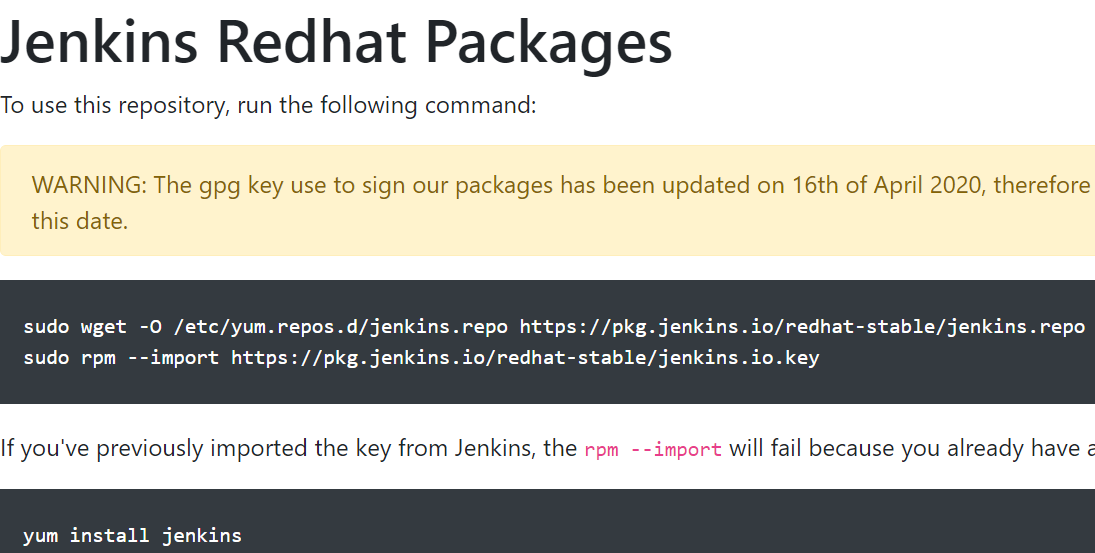
Jenkins配置安装

在另外一台192.168.134.139的主机上安装

安装java包：Yum –y install java-1.8.0-openjdk\*

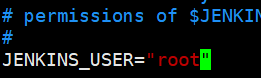


centos安装jenkins >>><https://pkg.jenkins.io/redhat-stable/>



跟随教程安装jenkins

修改vim /etc/sysconfig/jenkins





启动Jenkins

systemctl start Jenkins

关闭防火墙

Systemctl stop Jenkins

Systemctl disable Jenkins

登录192.168.134.139:8888，第一次登录需要默认管理员密码

Cat /var/lib/jenkins/secrets/initialAdminPassword



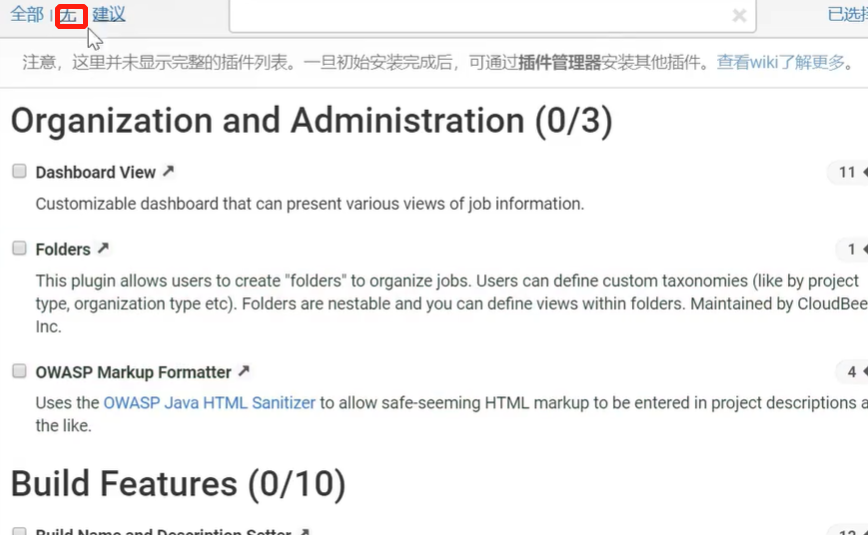
填入管理员密码中

进入安装界面

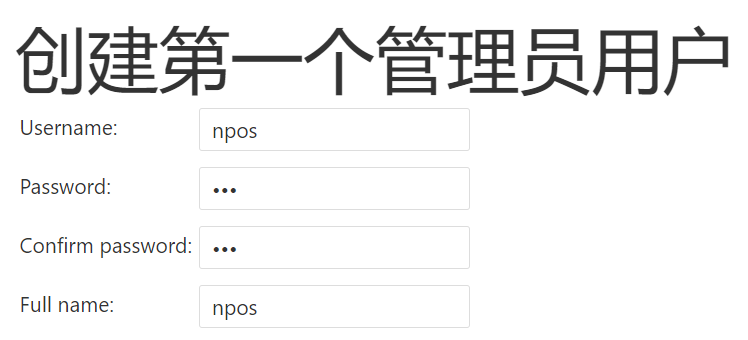
选择第二个

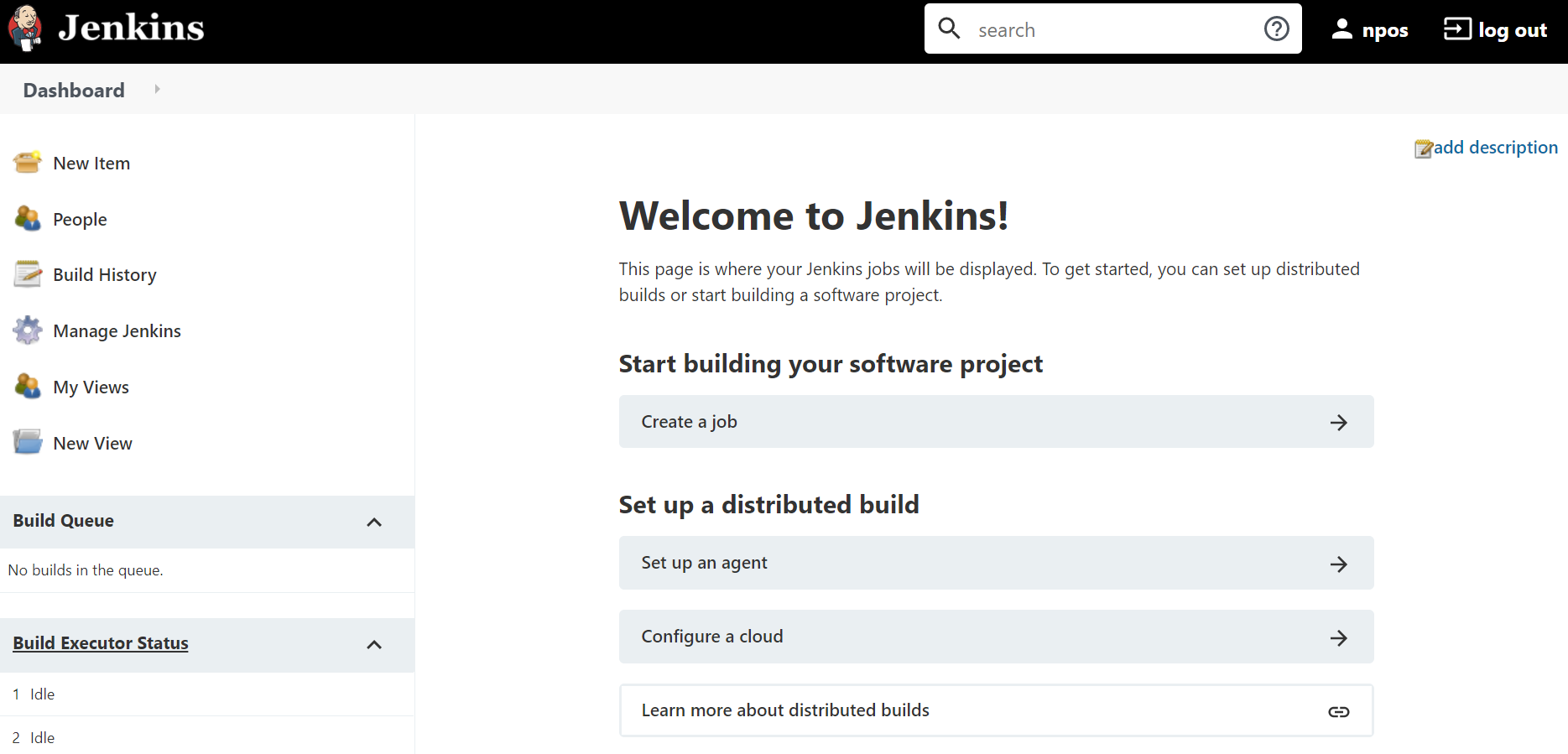


选择全无安装

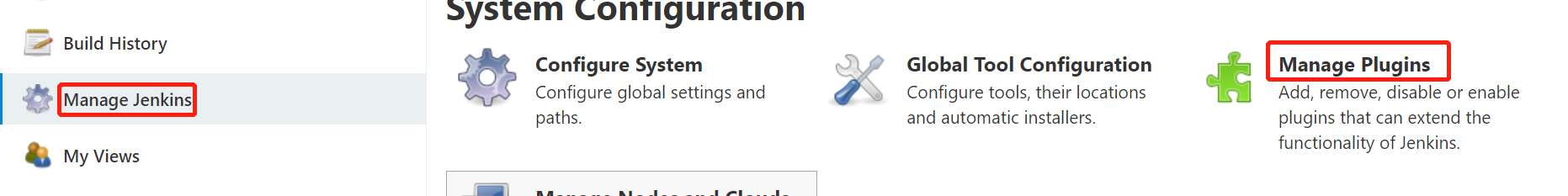


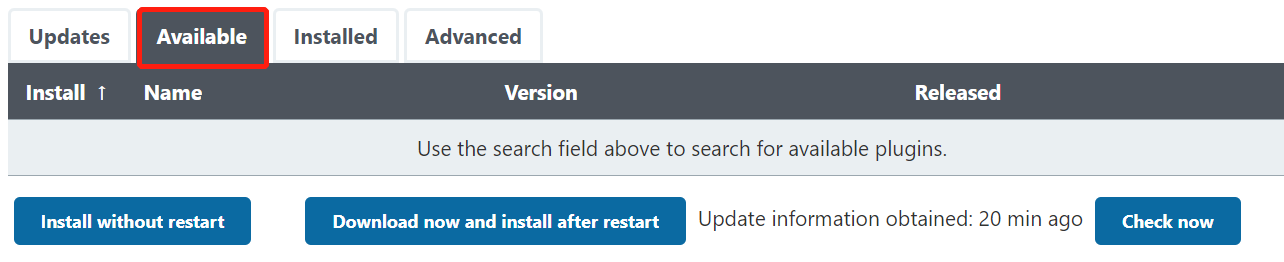
自定义一个管理员





修改jenkins插件下载配置



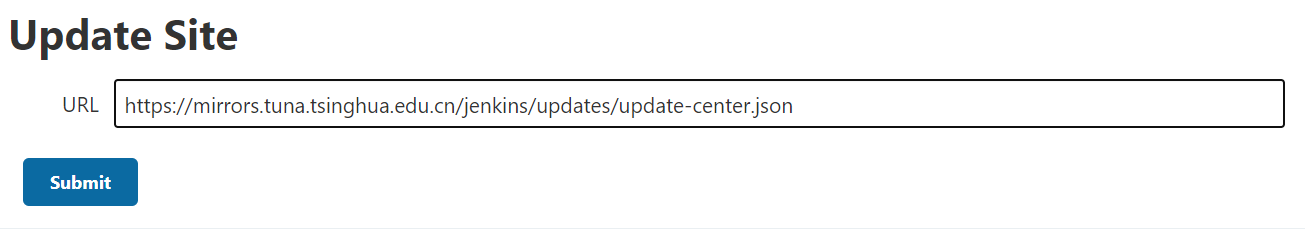


修改 /var/lib/Jenkins/updates/default.json

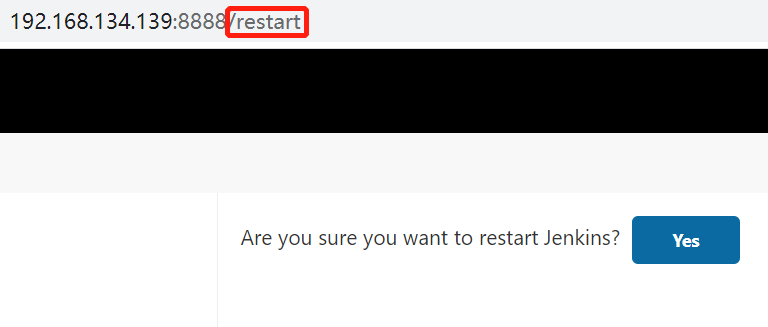
sed -i 's/http:\/\/updates.jenkinsci.org\/download/https:\/\/mirrors.tuna.tsinghua.edu.cn\/jenkins/g' default.json && sed -i 's/http:\/\/www.google.com/https:\/\/www.baidu.com/g' default.json

然后与点击advance高级选项，填入对应地址，点击submit

https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json

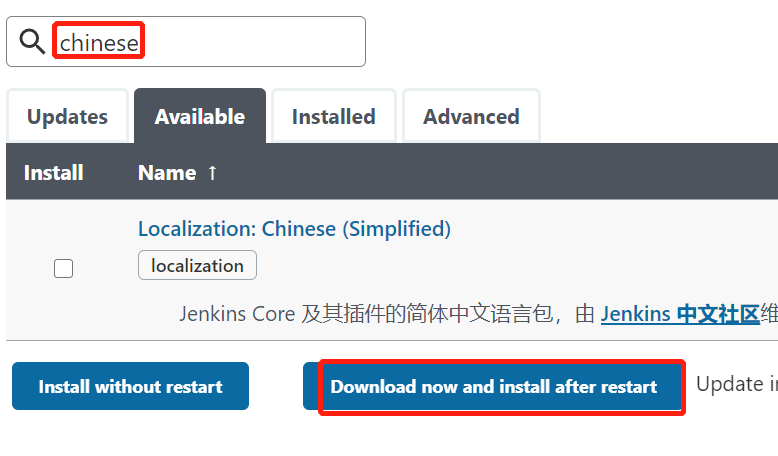


然后重启jenkins

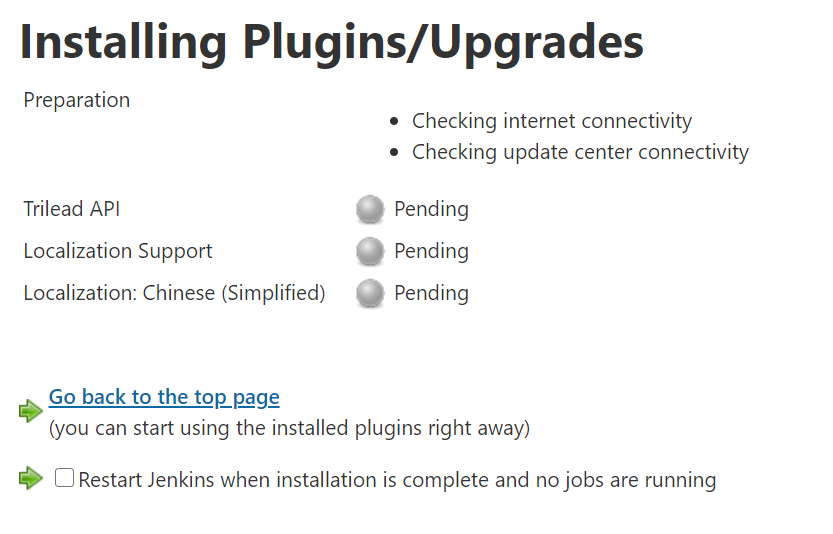


然后需重新登录

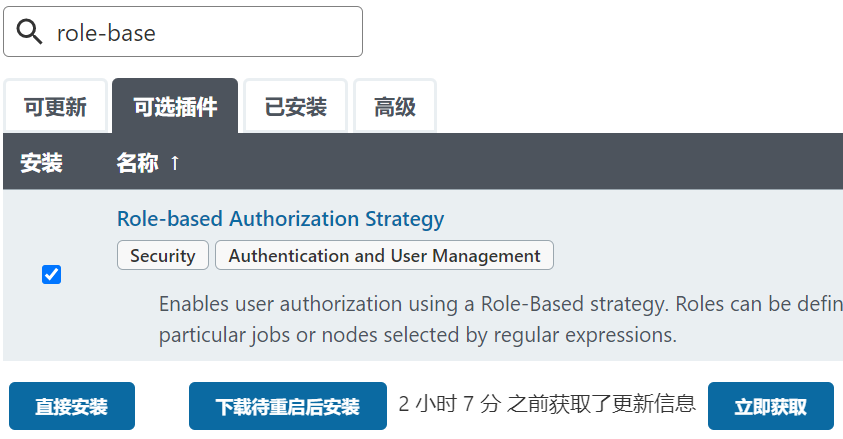
安装汉化包



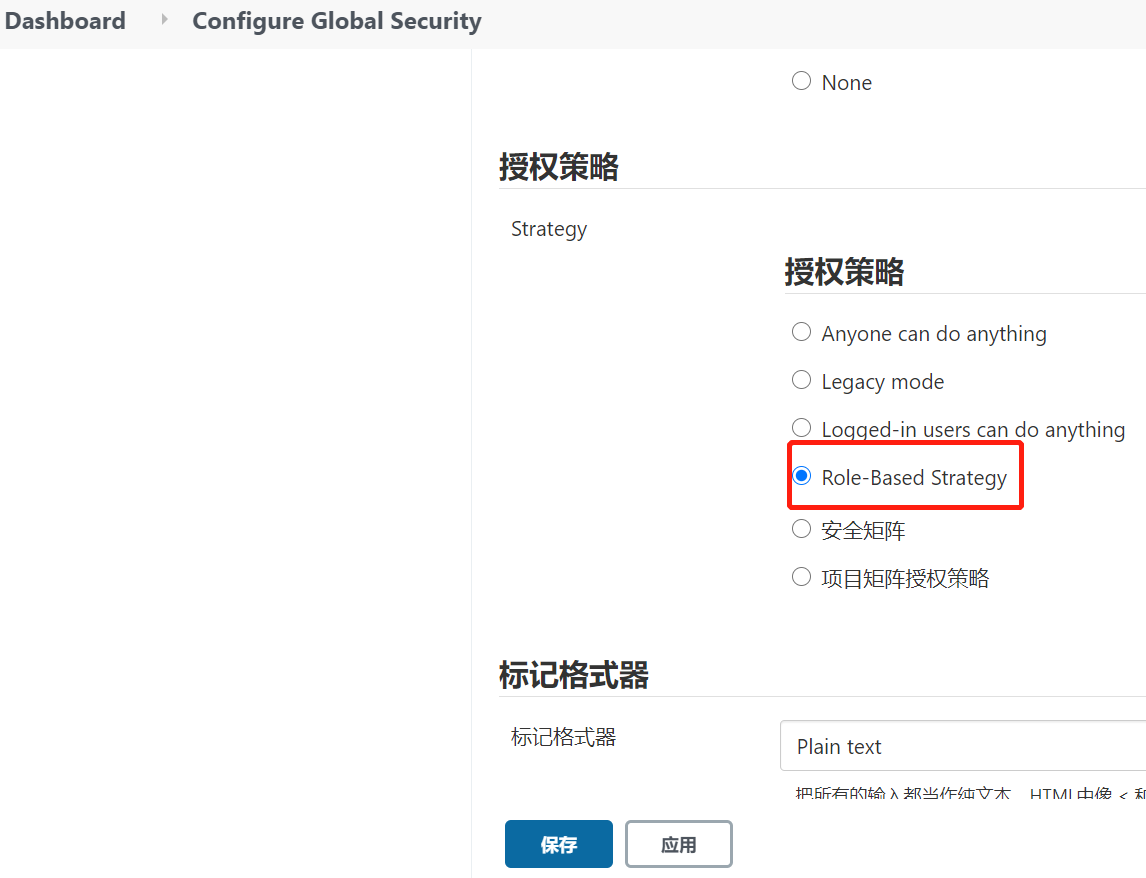
安装完包后，需要重启才能生效



安装role-based authorization strategy，提供role管理服务

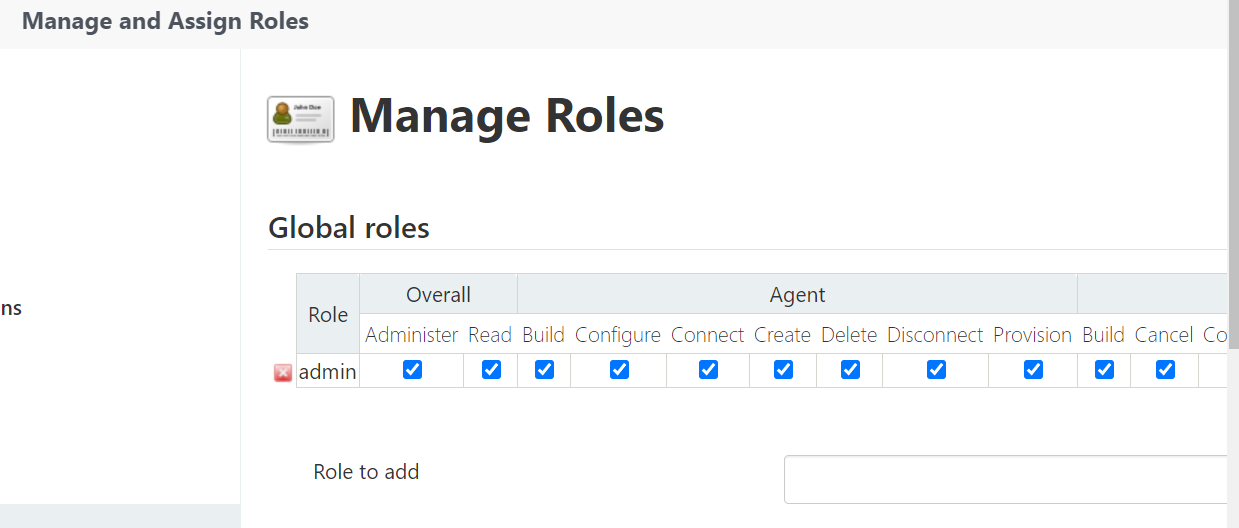


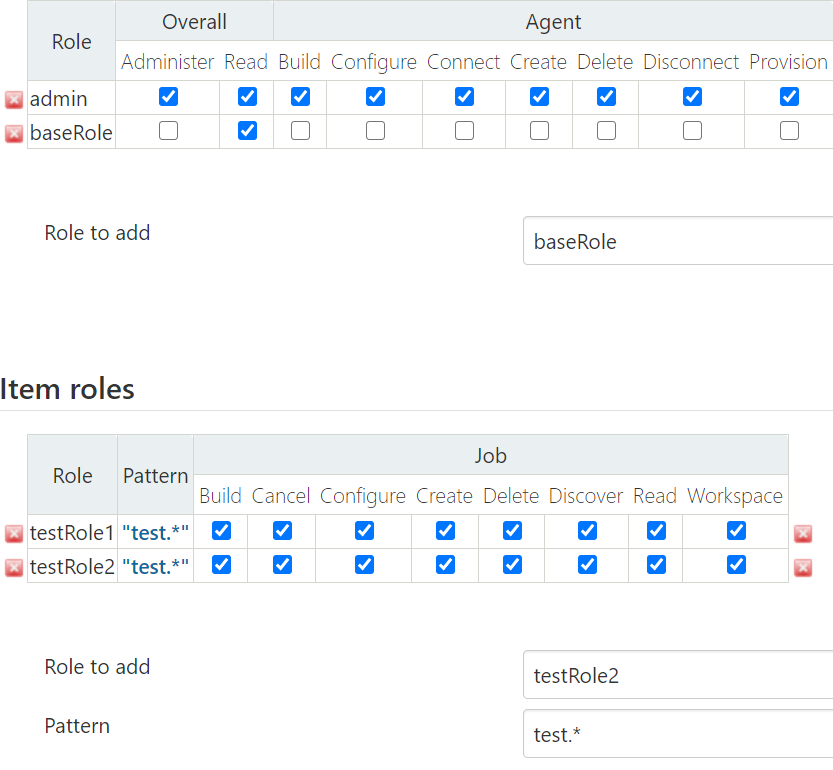
不需要重启



点击保存

添加新的global role与project role

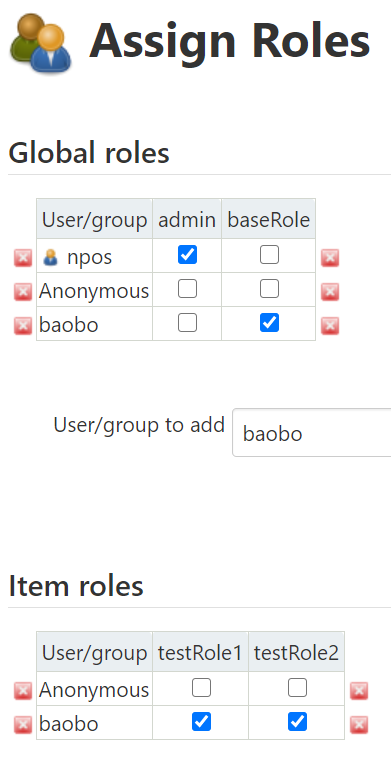




新建用户

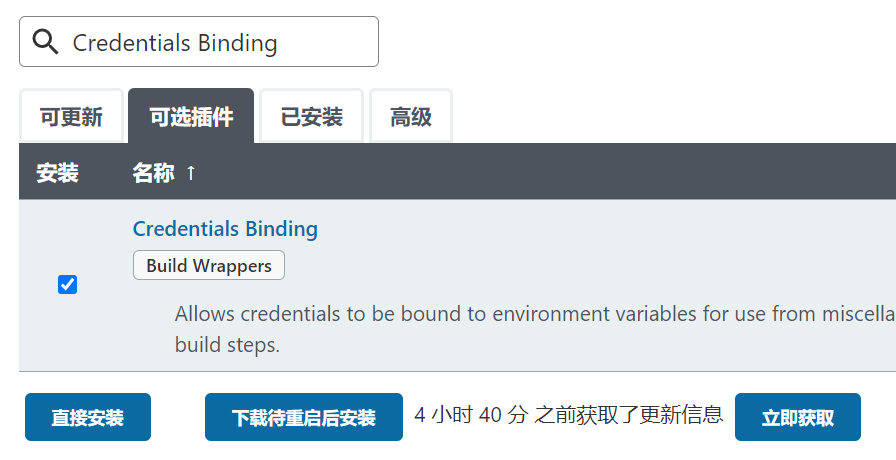


给用户添加角色

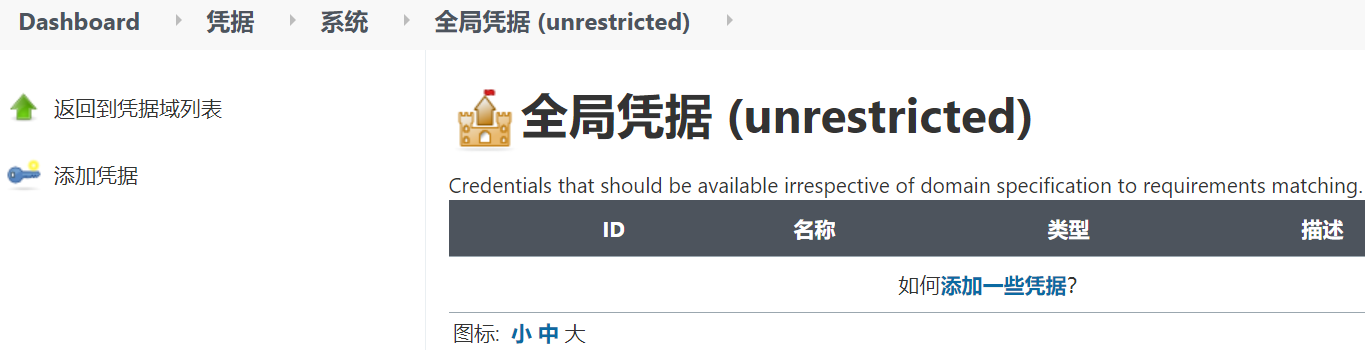


Jenkins凭证管理

jenkins凭证管理功能，需要安装凭证管理功能



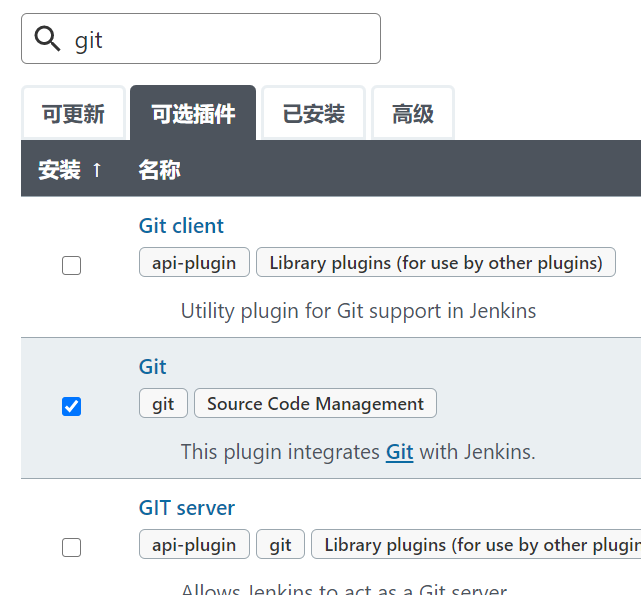
安装好后再manage Jenkins里找到凭据，点击全局



点击添加凭据



安装git插件

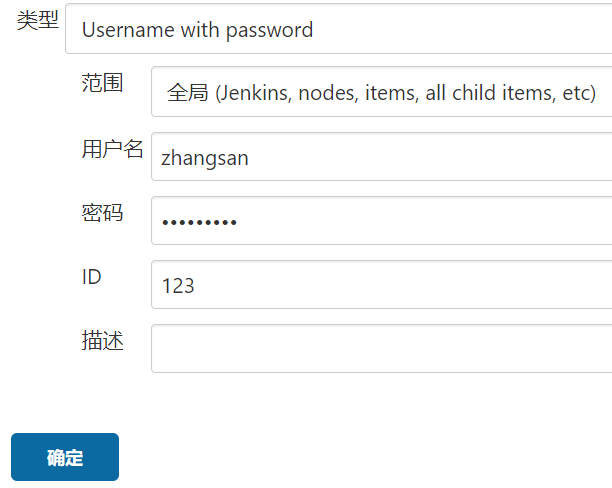


在192.168.134.139虚拟机上安装git



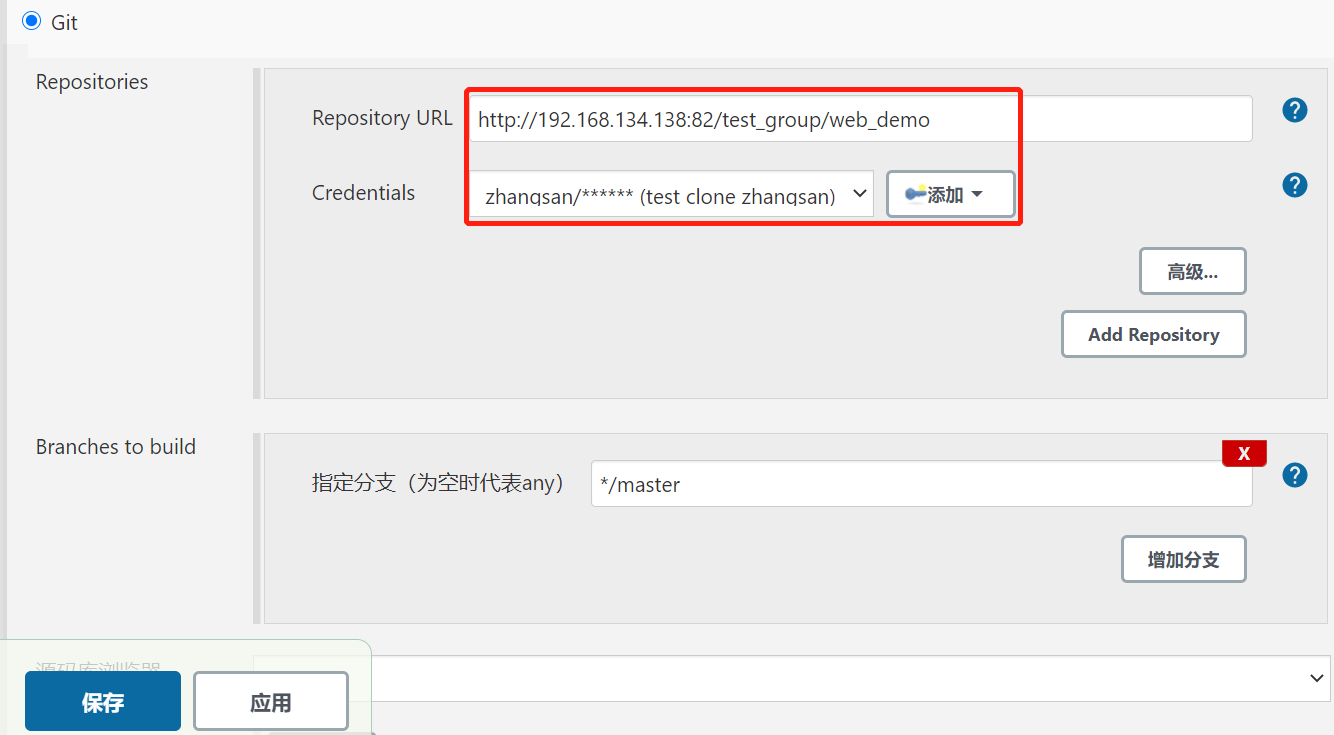


添加gitlab上的zhangsan用户名与密码，作为一个全局凭证



为后面拉取gitlab zhangsan的project上的代码做准备

点击配置项目，添加gitlab仓库地址和凭证



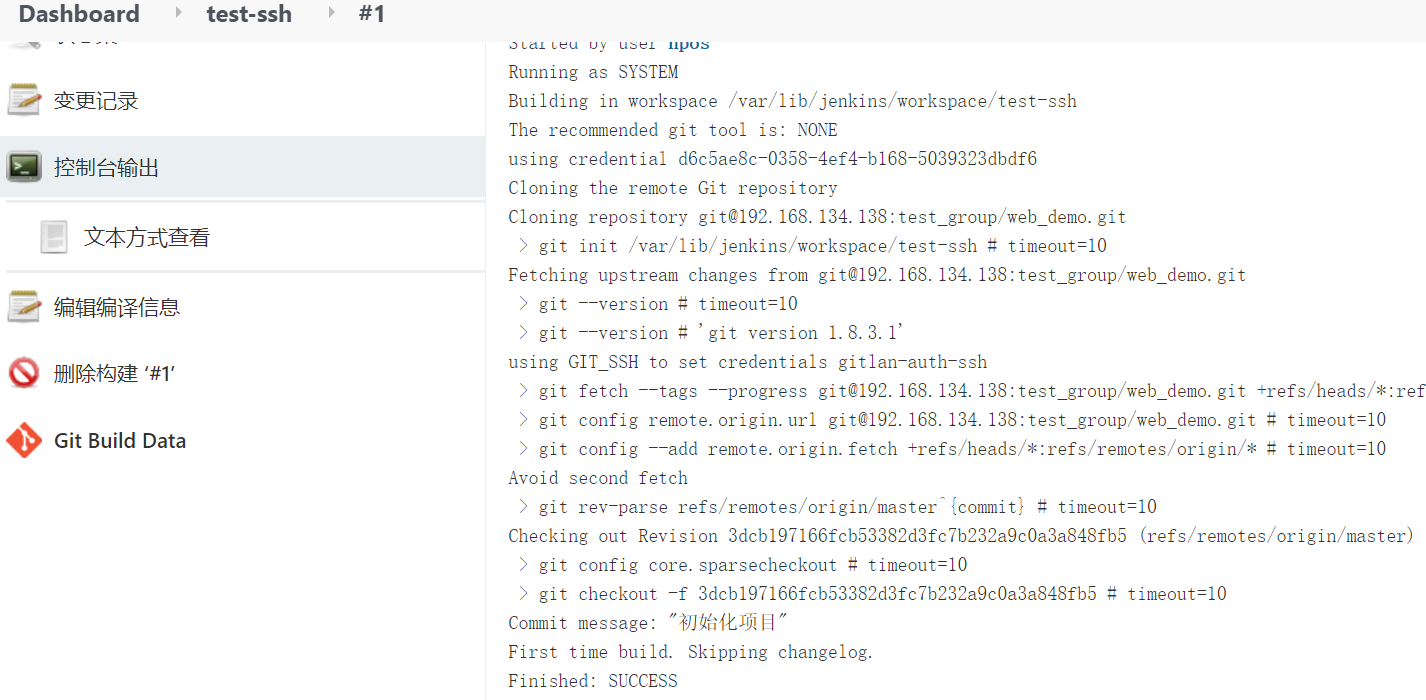
点击build now，再点击test02



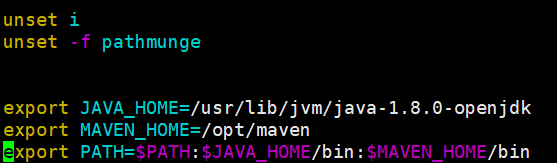
在jenkins02里的机子就能看到clone下来的项目



使用ssh拉取代码不记录

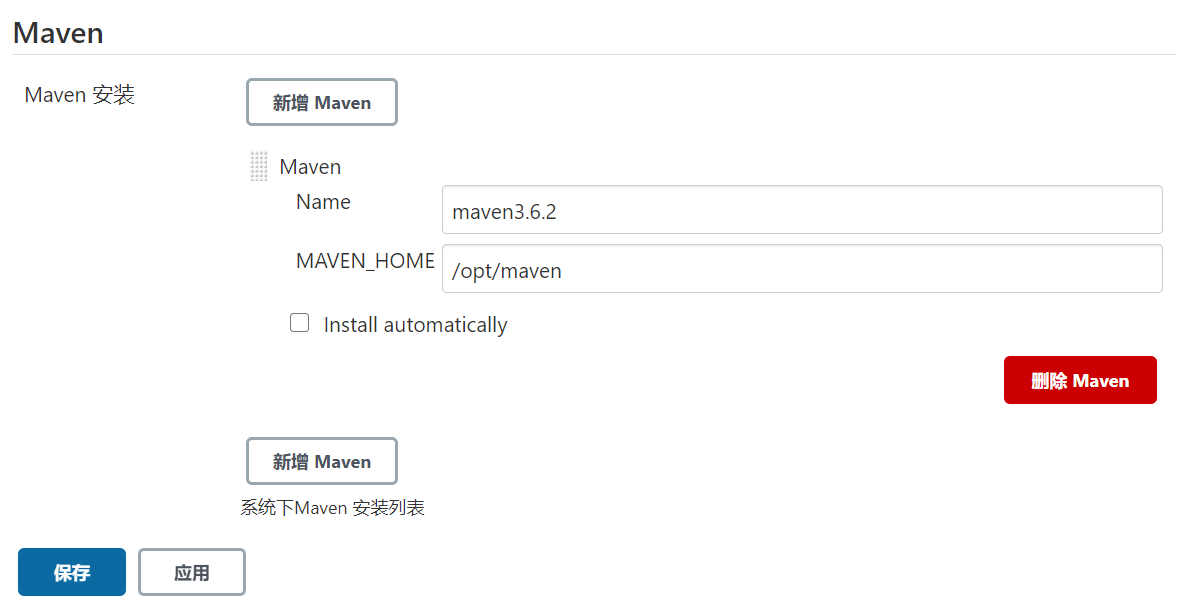


安装maven软件打包

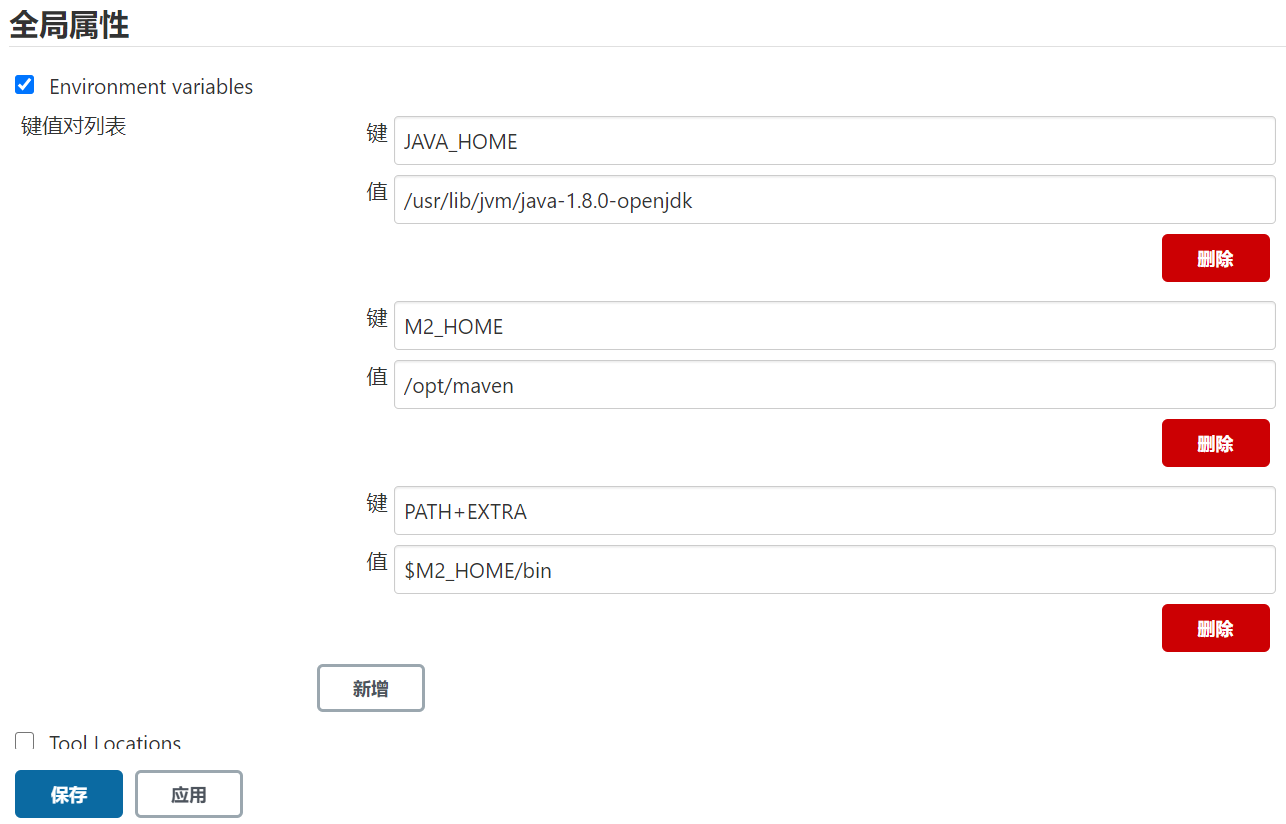


在jenkins global tool configuration里配置





增加系统配置



新增本地仓库

Mkdir /root/maven-repo

修改vim /opt/maven/conf/settings.xml

<localRepository>/root/maven-repo</localRepository>

<mirror>

<id>alimaven</id>

<mirrorOf>central</mirrorOf>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

</mirror>

