## 一、Jenkins的下载安装

### 1、安装Jenkins

在基于Debian的发行版（如Ubuntu）上，您可通过`apt`安装Jenkins

在[an apt repository](https://pkg.jenkins.io/debian/)可获得最新版本，较老但稳定的LTS版本在[this apt repository](https://pkg.jenkins.io/debian-stable/)这里可获得

wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt-get update

sudo apt-get install jenkins

安装这个软件包将会：

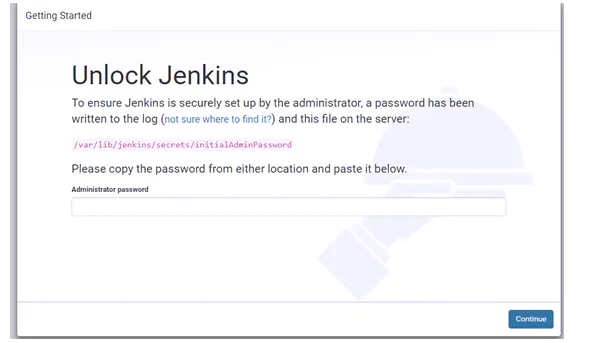
* 将Jenkins设置为启动时启动的守护进程。查看`/etc/init.d/jenkins`获取更多细节
* 创建一个'jenkins'用户来运行此服务
* 直接将控制台日志输出到文件`/var/log/jenkins/jenkins.log`。如果您正在解决Jenkins问题，请检查此文件
* /etc/default/jenkins`为启动填充配置参数，例如JENKINS\_HOME
* 将Jenkins设置为在端口8080上进行监听。使用浏览器访问此端口以开始配置

如果你的`/etc/init.d/jenkins`文件无法启动Jenkins，编辑`/etc/default/jenkins`， 修改 ----HTTP\_PORT=8080----`为----HTTP\_PORT=8081----` 在这里，“8081”也可被换为其他可用端口。

这个安装过程可能会比较长，中途会失败，应该是网络问题，失败之后再试就行，如果安装启动失败吗，检查一下有没有正在运行的服务端口是8080，出现了端口号冲突

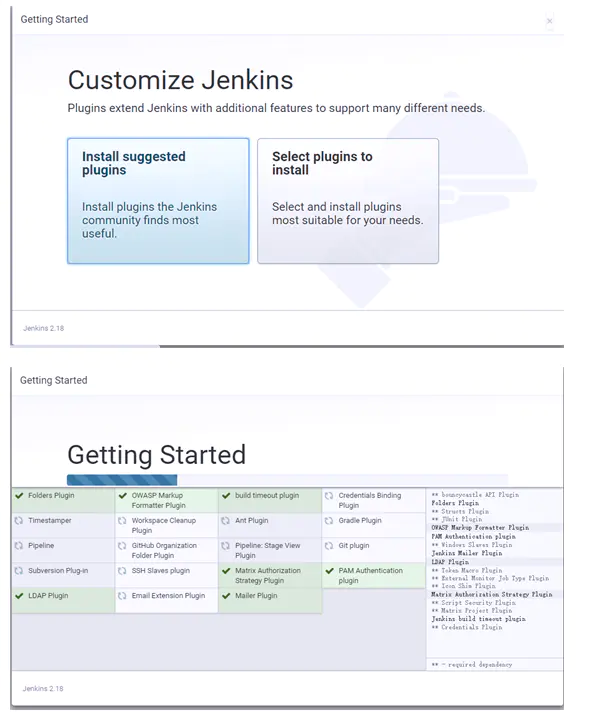
### 2、安装插件

访问ip:8080/进入jenkins界面，第一次进入需要输入初始化密码，密码在你挂载Jenkins目录下的/secrets/initialAdminPassword中可以查看到，输入之后到下一步。



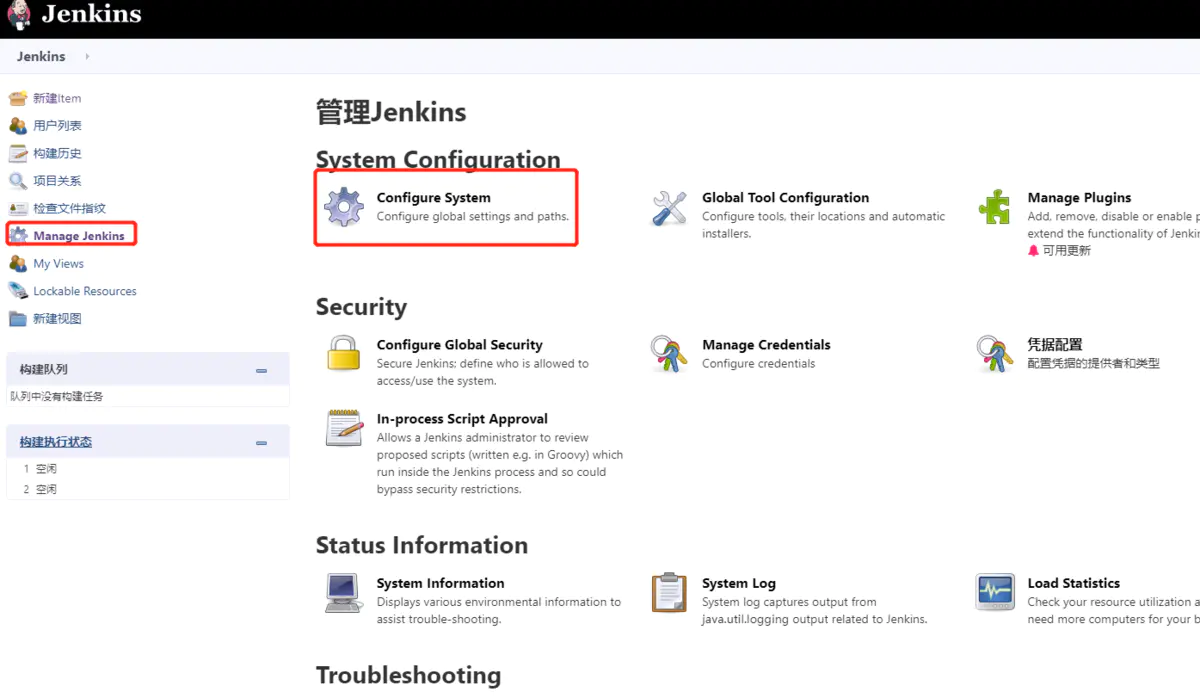
选择“Install suggested plugins”安装默认的插件，下面Jenkins就会自己去下载相关的插件进行安装。

下载完成之后会进入一个创建账号的界面，这里自己创建一个就行，后续登录就用这个账号。



下载完成之后会进入一个创建账号的界面，这里自己创建一个就行，后续登录就用这个账号。

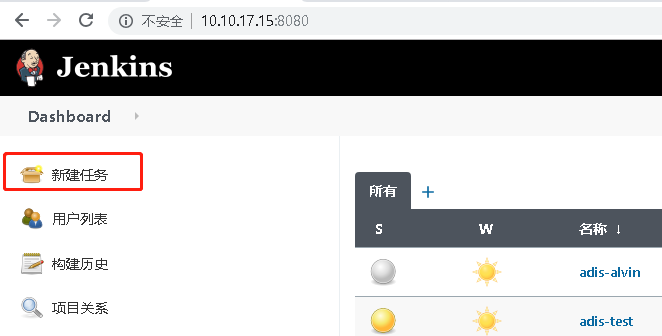
然后进入管理Jenkins-->系统配置



添加插件，重启jenkins服务

### 3、创建任务并配置

首先，登录Jenkins，点击新建任务



进入新的页面，输入任务名称，选择“构建一个风格的软件项目”，最后点确定

点击最下面的“确定”按钮



这样一个Jenkins软件项目被创建了，并且现在页面跳转到了Jenkins软件项目的配置页面

先不管General，后面要设置tag再来看他

点击“源码管理”，选择git，下面会弹出两个框需要填写



URL输入框输入git仓库

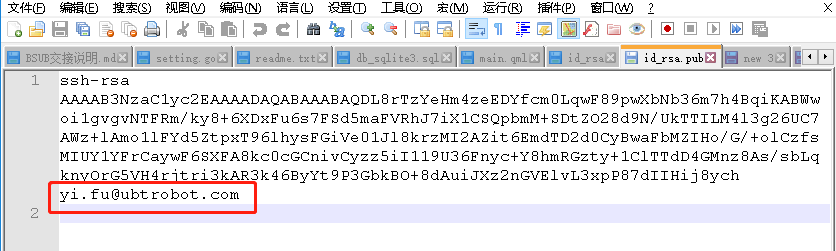


注意：Git仓库路径也要用IP，域名无法识别

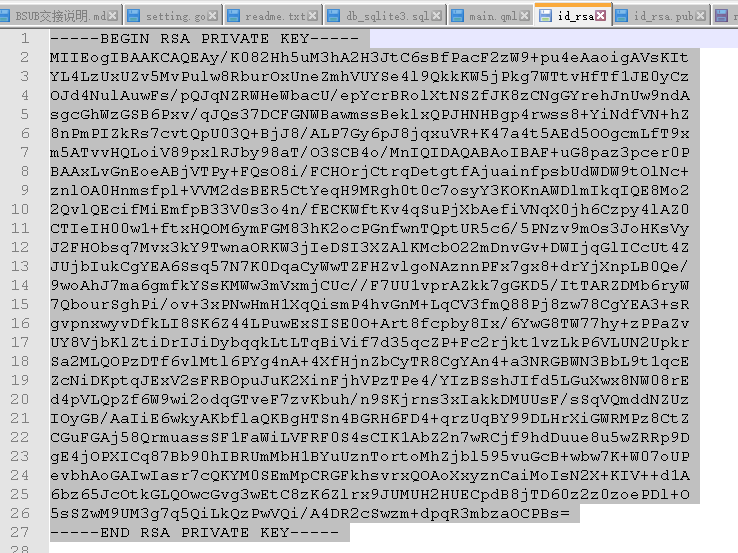
点击“添加”-“Jenkins”，类型选择“SSH Username with private key”



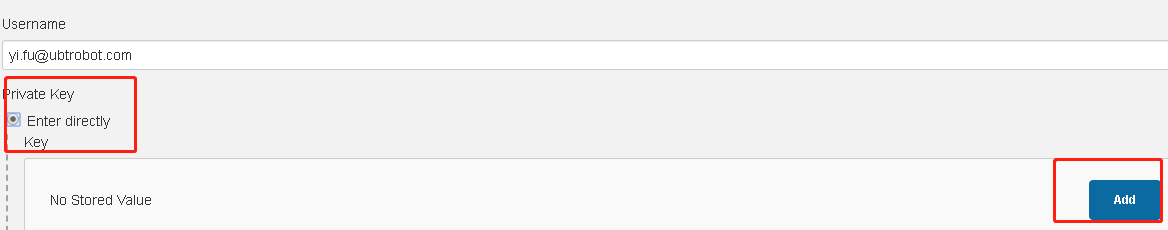
打开本机的公钥，选择最后的一行的邮箱，[yi.fu@ubtrobot.com](mailto:yi.fu@ubtrobot.com)，填写在Username的输入框



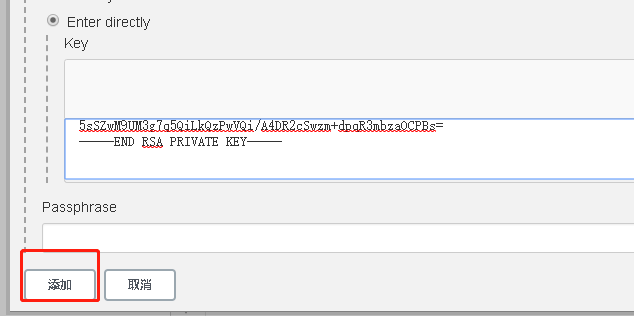
发开私钥，复制



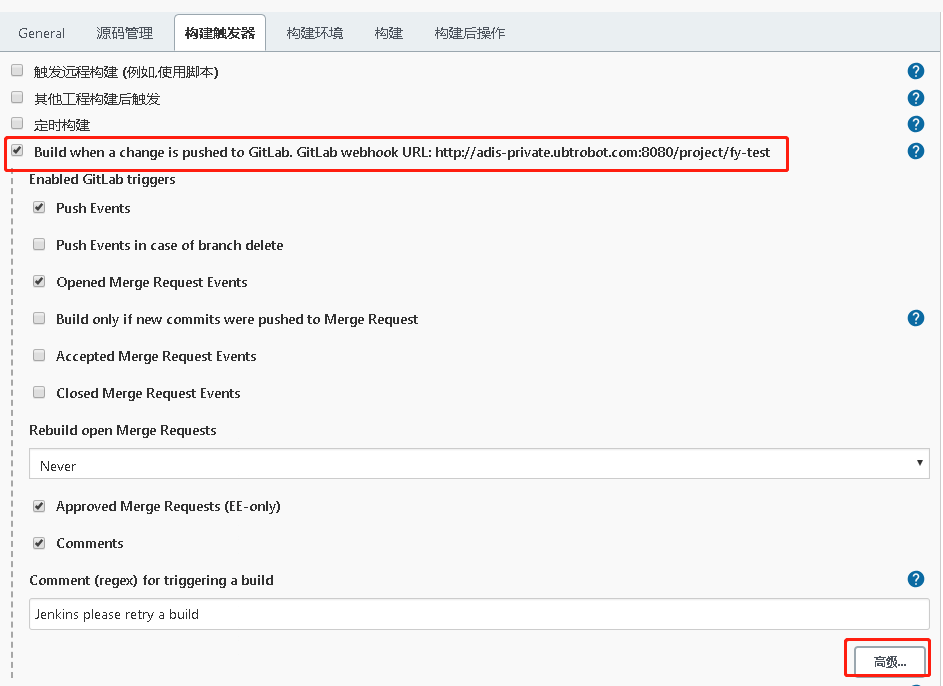
在private key 下面选择Enter directly,再点击add，把私钥粘贴在下面的输入框



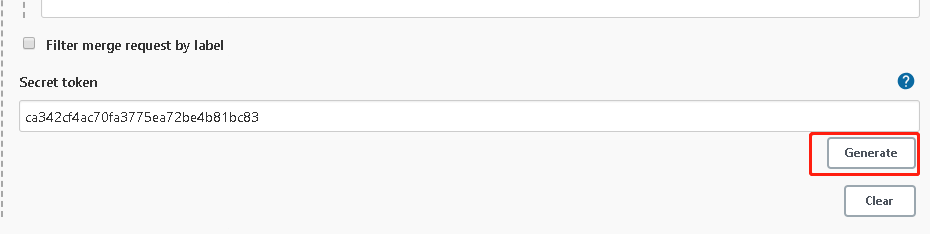
最后点击“添加”，在Credentials的下拉框选择公钥中的邮箱，如“[yi.fu@ubtrobot.com](mailto:yi.fu@ubtrobot.com)”



接着点击“构建触发器”，勾选“Build when a change is pushed to GitLab”，保存URL后面的路径，在gitlab 上设置webhook要用到，然后点击最下面的按钮“高级”



点击“高级”后下面会弹出很多可以设置的选项，其他的不管，点击generate可以生成token，这个token要保存一下，后面设置webhook要用到。



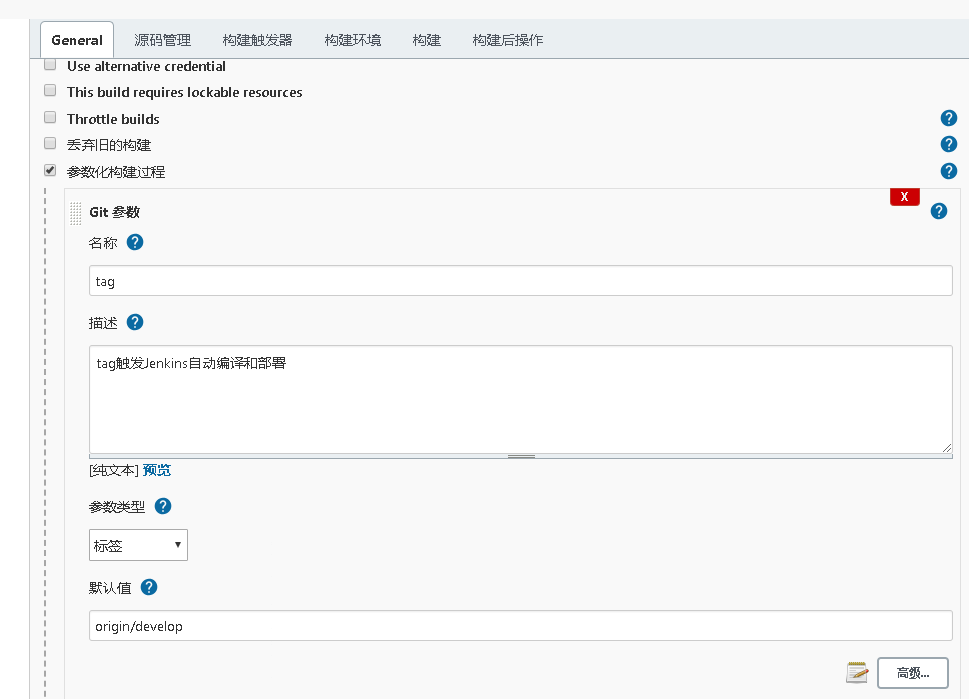
最后点击“应用”-“保存”按钮退出界面。

**4、Go项目的推送与自动拉取**

**webhook实现git提交代码，触发Jenkins编译代码**

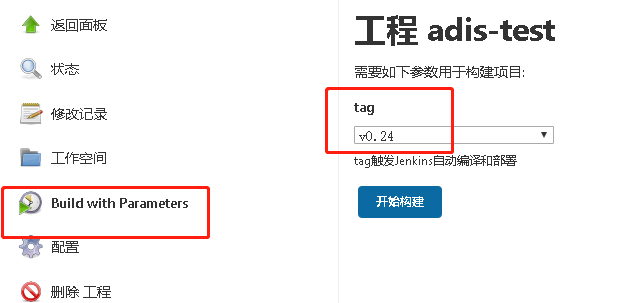
**设置jenkins**

前面**“**创建任务并配置”中已经设置好了jenkins，后面设置好webhook后，只要git提交代码，jenkins就会拉取代码，进行build，但是一般提交代码次数太多，最好是提交的代码有Tag才触发构建，要实现这个功能需要继续设置jenkins。

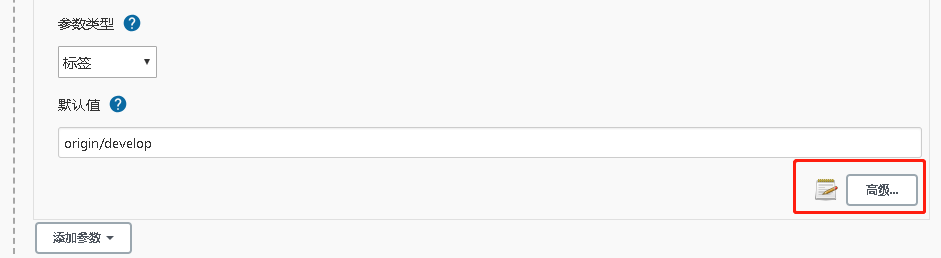


点击General，勾选参数化构建过程，在名称下面填一个字段，我这里写的是tag，描述是选填项，参数类型选标签，默认值写你要拉取的分支，可以填origin/master。

这样设置之后，保存退出，会发现左侧列表出现了一个Build with parameter，点击它会出现右边的tag选择下拉框，和开始构建按钮，可能会有多个tag，想要自动触发，还要回到配置页面



点默认值下面的高级按钮

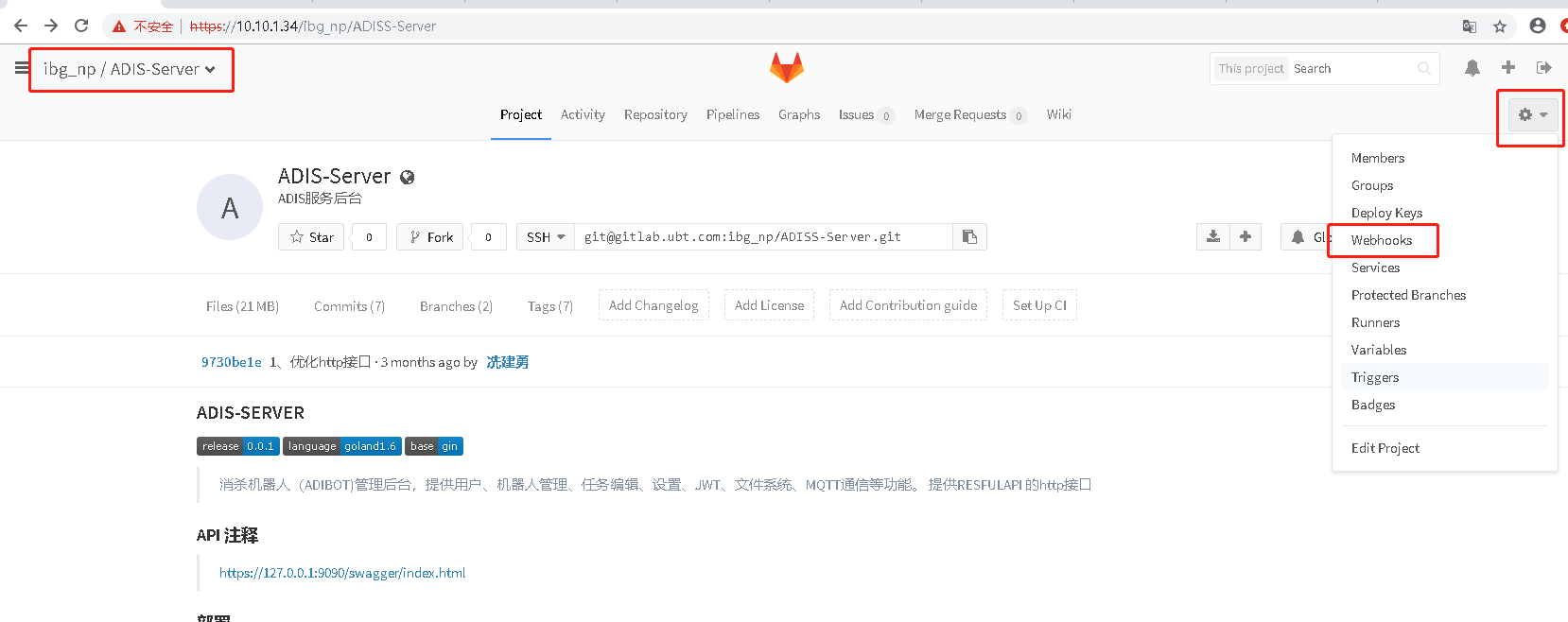


进行如下设置即可

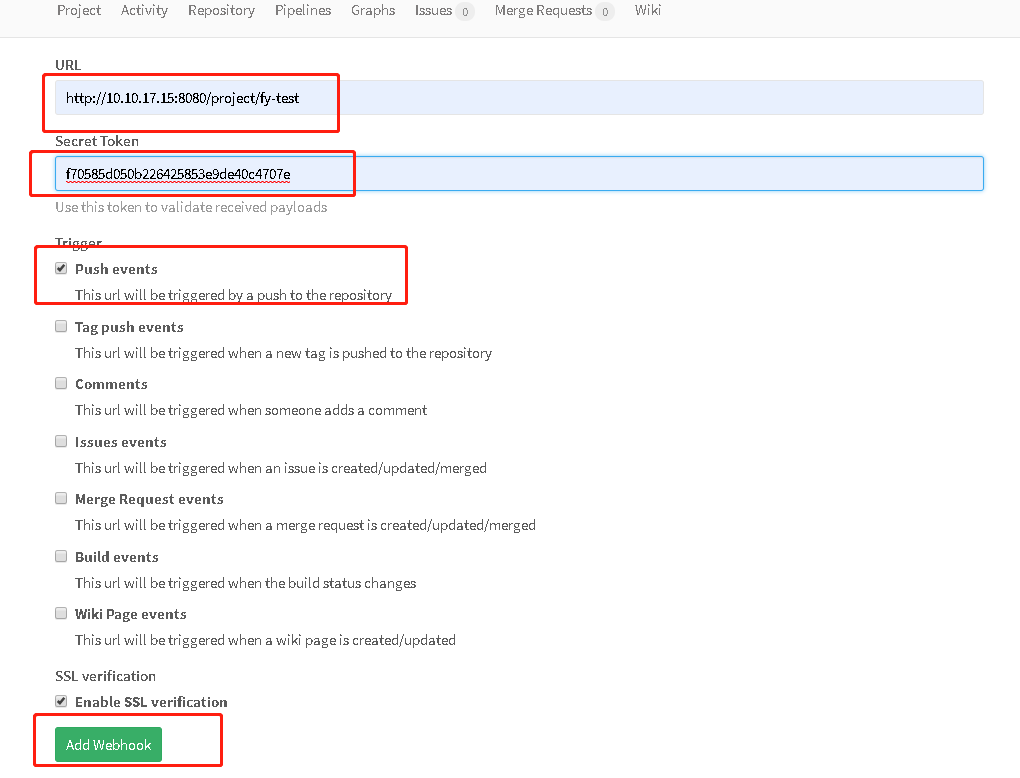


**设置gitlab**

登录gitlab，选择要编译部署的工程，点击右上角齿轮图标，选择webhooks

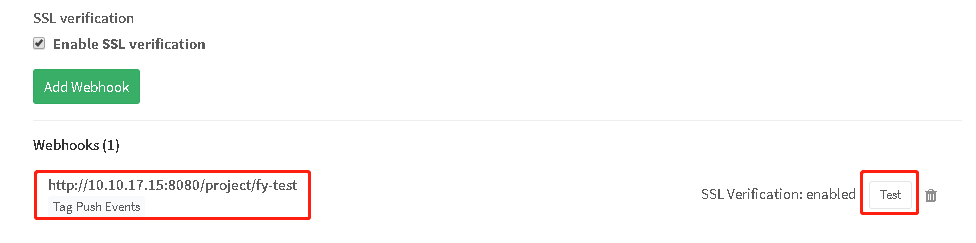


然后在URL填上jenkins拉取工程代码的路径和token，并点击Add Webhook按钮，URL和token这两个在之前设置Jenkins是提示要保存一下的。

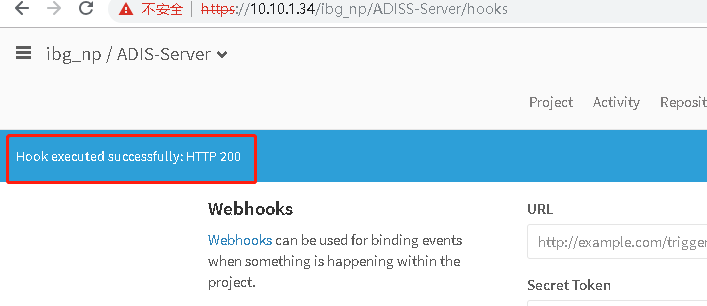


注意：路径中端口号前一定要是IP，不能是域名，如果是域名，gitlab可能识别不了

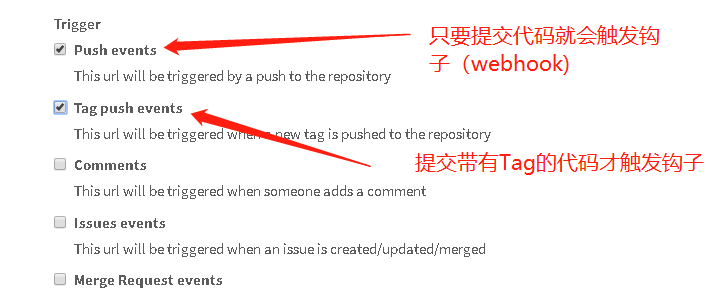
最后，验证一下，webhook是否已经设置成功，在Add Webhook之后，最下面会有你添加webhook 的记录，可以Test，也可以删除，



点击Test,如果界面出现执行成功的提示，表示webhook已经设置成功，可以出发Jenkins



另外，还可以设置出发的具体方式



## 二、持续集成和自动部署

#自动编译部署脚本

echo "starting..."

pwd

go mod tidy

go build

sudo systemctl stop adis

cp -r /var/lib/jenkins/workspace/fy-test/adis-server /usr/bin/adis/adis-server

sudo systemctl restart adis

echo "ending..."

#自动编译打包文件

echo "starting..."

pwd

go mod tidy

go build

comitId=`git rev-parse --short HEAD`

echo "comitId is：" $comitId

tagName=`git describe --abbrev=0 --tags`

echo "tag name is：" $tagName

cp -r /var/lib/jenkins/workspace/fy-test/adis-server /home/adis/upgrade\_adis/adis-server

cp -r /var/lib/jenkins/workspace/fy-test/dist /home/adis/upgrade\_adis/dist

cd /home/adis

tar -zcvf ./$tagName-$comitId.tar.gz -C ./upgrade\_adis/ .

echo "ending..."

## 三、问题及解决方案

### 1、Jenkins 构建 Go 项目时 $GOPATH 的处理

编译 Golang 程序需要在 $GOPATH 下面，但是 Jenkins 拉取 Go 项目的代码以后，默认是放在 Jenkins 当前构建进程的 $WORKSPACE 下面的，而这个 $WORKSPACE 并非 GOPATH，此时直接 go build 是行不通的。需要设置系统环境变量GOPATH和GOROOT

cd到etc目录下:

cd /etc

打开peofile文件

sudo vim profile

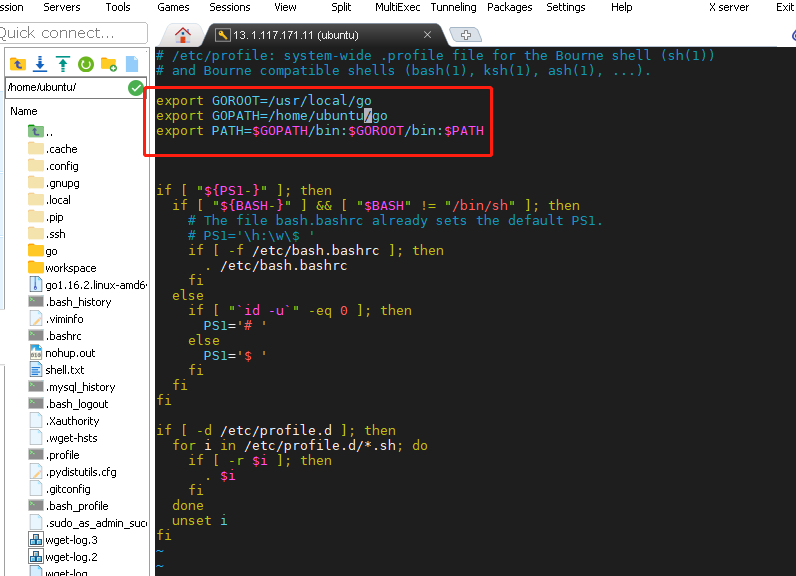
在文件里面添加

export GOROOT=/usr/local/go

export GOPATH=/home/adis/go

export PATH=$GOPATH/bin:$GOROOT/bin:$PATH

保存后退出即可



### 2、Jenkins 权限问题

由于需要在jenkins中执行shell脚本，但是有些目录下没有权限，下面的操作为jenkins用户授予root权限。

jenkins用户加入到root组:

sudo gpasswd -a root Jenkins

编辑文件/etc/sysconfig/Jenkins:

sudo vim /etc/default/jenkins

JENKINS\_USER="root"

JENKINS\_GROUP="root"

