**gitlab版本控制系统**

一介绍

git Git是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。

Git是一个开源的分布式版本控制系统，用以有效、高速的处理从很小到非常大的项目版本管理。

从一般开发者的角度来看，git有以下功能：

1、从服务器上克隆完整的Git仓库（包括代码和版本信息）到单机上。

2、在自己的机器上根据不同的开发目的，创建分支，修改代码。

3、在单机上自己创建的分支上提交代码。

4、在单机上合并分支。

5、把服务器上最新版的代码fetch下来，然后跟自己的主分支合并。

6、生成补丁（patch），把补丁发送给主开发者。

7、看主开发者的反馈，如果主开发者发现两个一般开发者之间有冲突（他们之间可以合作解决的冲突），就会要求他们先解决冲突，然后再由其中一个人提交。如果主开发者可以自己解决，或者没有冲突，就通过。

8、一般开发者之间解决冲突的方法，开发者之间可以使用pull 命令解决冲突，解决完冲突之后再向主开发者提交补丁。

优点：

适合[分布式开发](http://baike.baidu.com/view/2370062.htm)，强调个体。

公共服务器压力和数据量都不会太大。

速度快、灵活。

任意两个开发者之间可以很容易的解决冲突。

离线工作。

GitLab 是一个用于仓库管理系统的开源项目。使用[Git](http://baike.baidu.com/view/1531489.htm)作为代码管理工具，并在此基础上搭建起来的web服务。我们也是通过gitlab进行管理。

## 文档修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 文档版本 | 修改日期 | 维护人 | 修改内容 |
| 1.0 | 20150818 | 王盛 | 初次建立 |
|  |  |  |  |

目录

## 一介绍

## 二gitlab安装与使用

2.1手工安装

2.2 通过rpm安装

2.3 初始化操作

2.4 登陆gitlab系统,修改root密码

2.5 通过gitlab创建用户

2.6 创建一个仓库

2.7提交ssh-key到gitlab

2.8提交master

2.9 创建分支branch

2.10 模拟多人开发

2.11合并分支

2.12 将本地主线master推送到gitlab中

## 三 git常用操作

目前移动在上已经使用了gitlab，地址是http://192.168.5.227/

用户是 root 密码是com.jiuxian@admin

### 二gitlab安装与使用

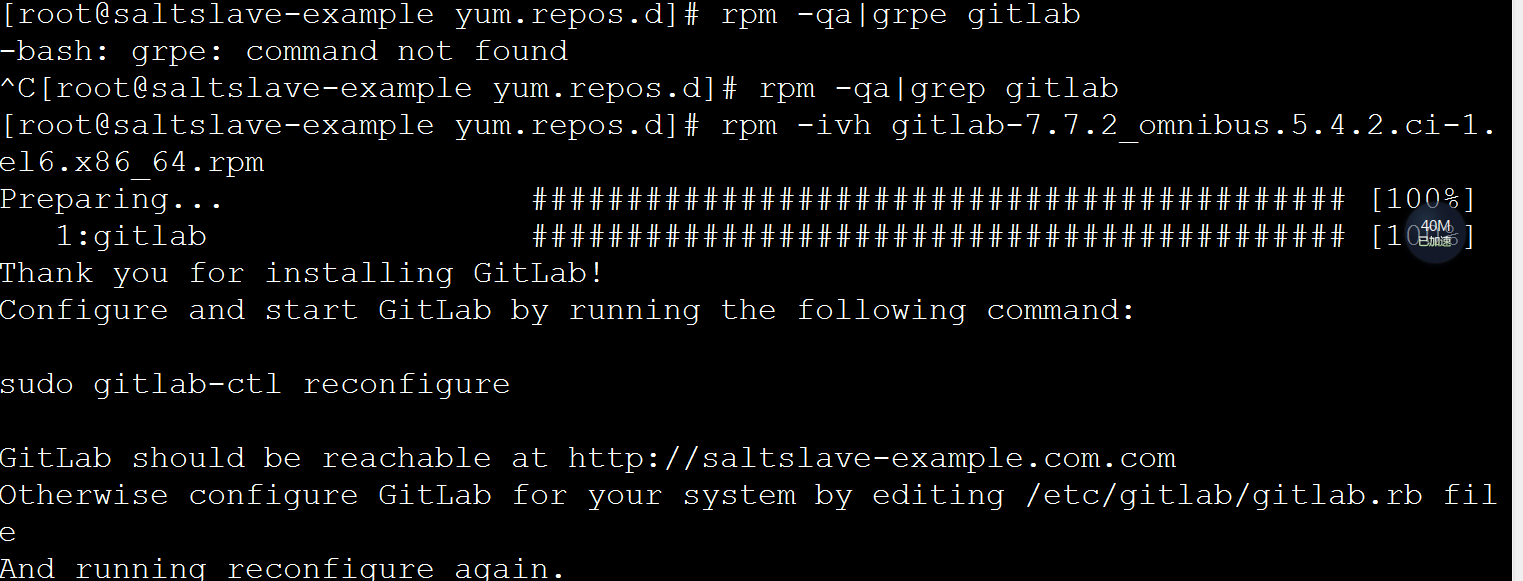
#### 2.1 手工安装

详见 https://gitlab.com/gitlab-org/gitlab-ci/tree/master/doc/install

#### 2.2 通过rpm安装

检查本机是否安装gitcli

|  |
| --- |
| [root@saltslave-example ~]# rpm -qa|grep gitcli  增加163源  cd /etc/yum.repos.d  mkdir bak  mv \*.repo bak/  增加163的源  [root@saltslave-exampleyum.repos.d]#wget http://mirrors.163.com/.help/CentOS6-Base-163.repo  [root@saltslave-example yum.repos.d]# yum makecache上传软件  yum install -y lrzsz  上传文件 gitlab-7.7.2\_omnibus.5.4.2.ci-1.el6.x86\_64.rpm  安装  rpm -ivh gitlab-7.7.2\_omnibus.5.4.2.ci-1.el6.x86\_64.rpm |



#### 2.3初始化操作

|  |
| --- |
| gitlab-ctl reconfigure |

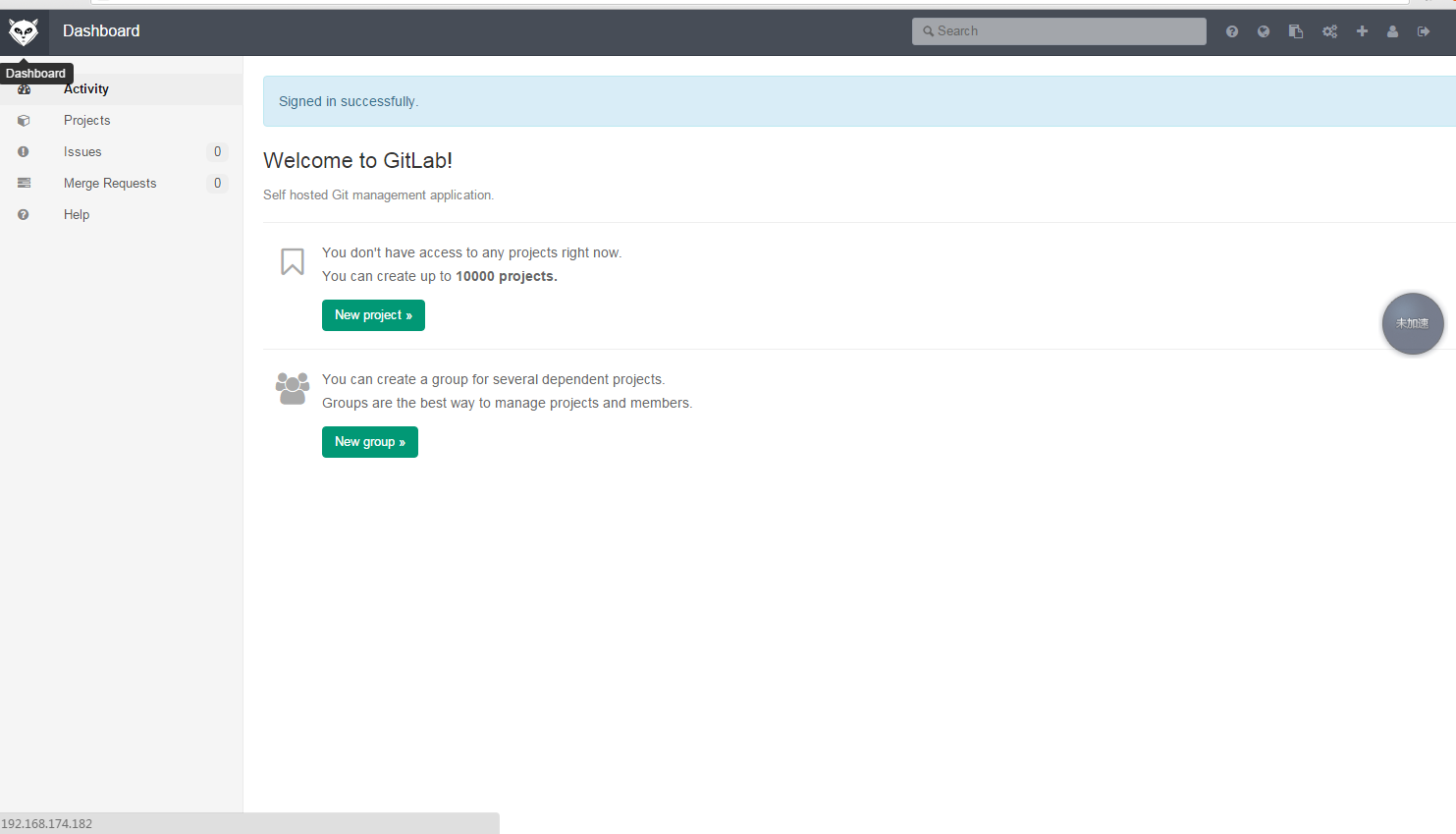
此操作为初始化gitlab的操作,会自动的安装gitlab-ctl,nginx,redis,postfix.

2.4 登陆gitlab系统

登陆的地址为本机的Ip eg:http:// http://192.168.174.182/

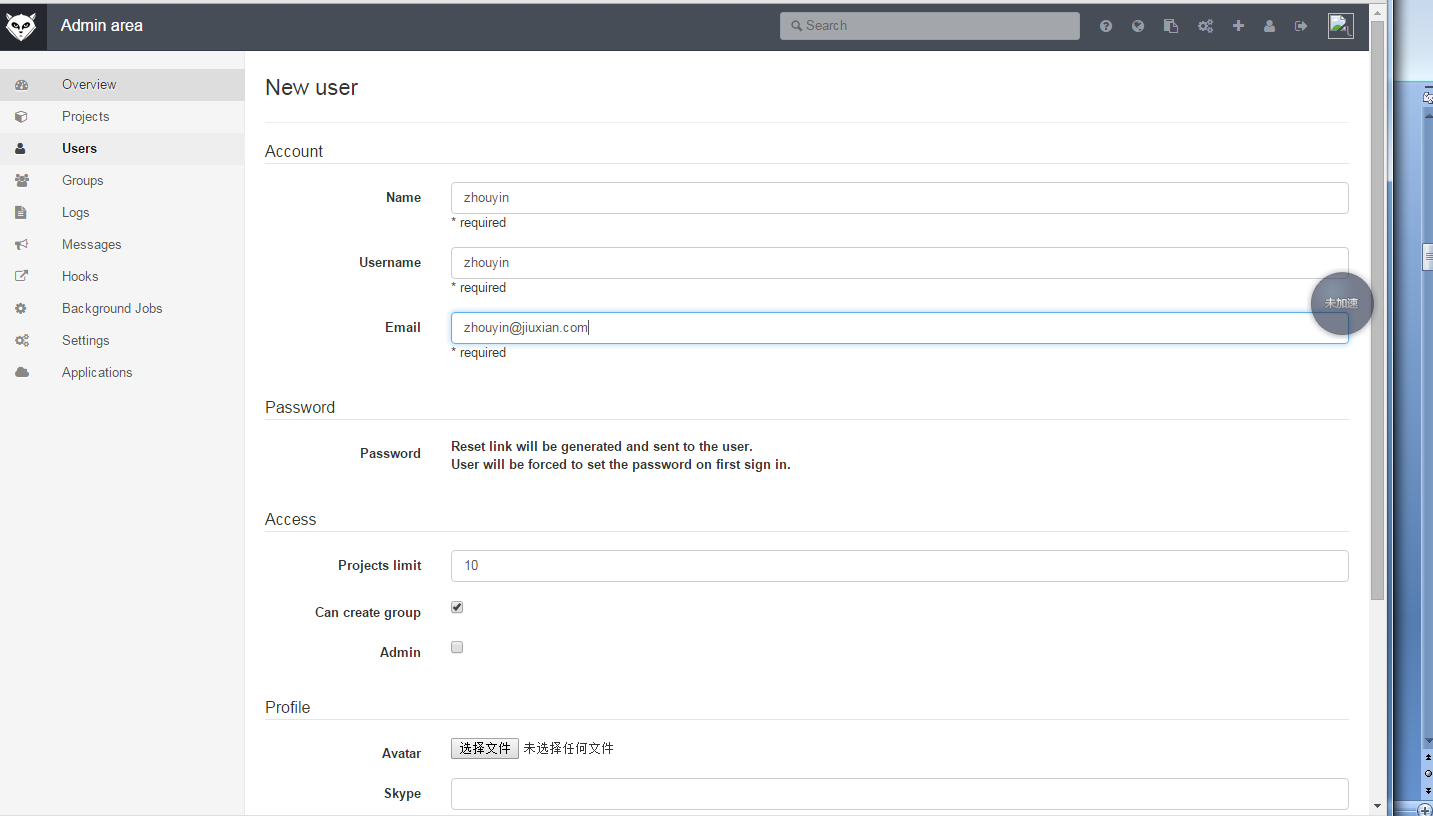
**默认账号和密码, 默认帐号root, 密码 5iveL!fe**

root 登陆系统后提示要重新更改密码需要修改默认密码,统一修改成jx.123456，修改成功后,登陆系统。

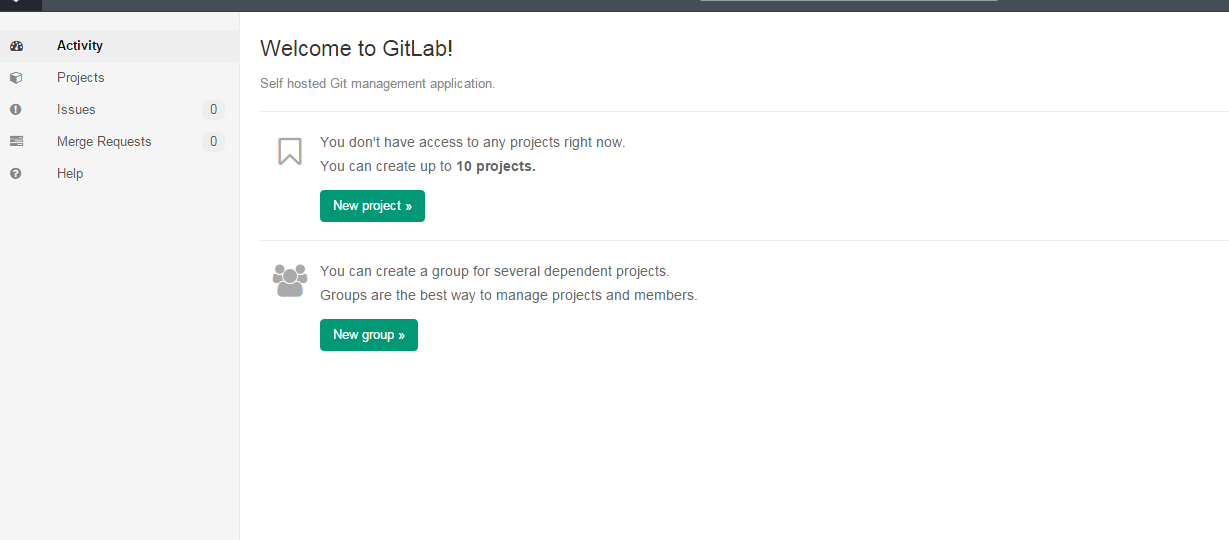


#### 2.5 创建一个用户

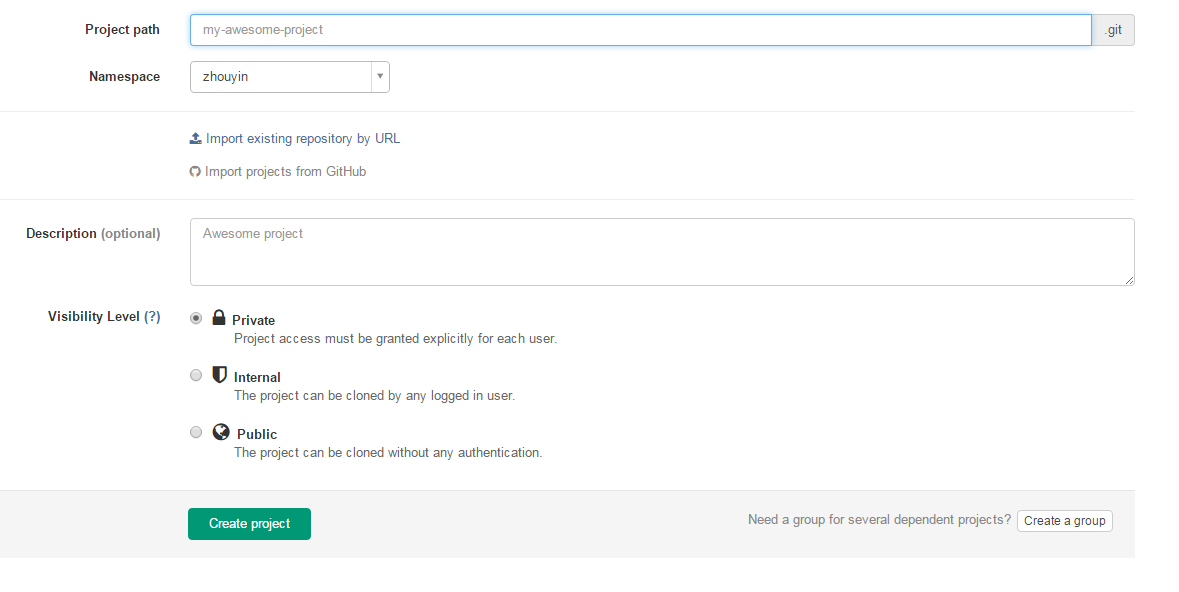
管理员创建一个新的用户



#### 2.6 创建仓库

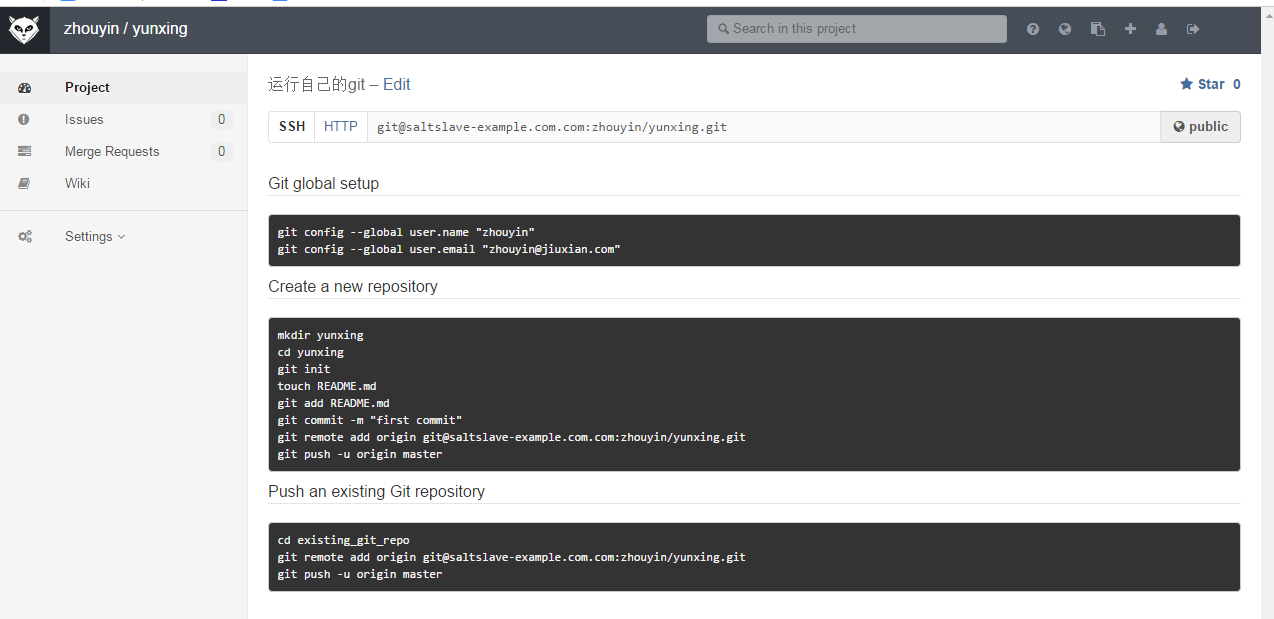


点击 New project,是创建一个新的项目。



填写项目的名称,描述,项目的权限

eg:我新建了一个项目yunxing,权限是public,描述是运行组的git仓库.



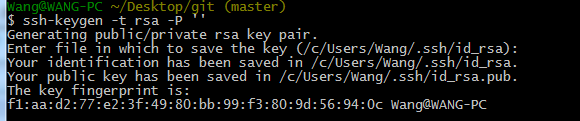
在gitlab上面已经将项目创建完成,

在本地建立一个仓库

|  |
| --- |
| eg:以一台机器做一个示范  [root@test-5-248 ~]# git config --global user.name "zhouyin"  [root@test-5-248 ~]# git config --global user.email "zhouyin@jiuxian.com"  [root@test-5-248 ~]# mkdir yunxing  [root@test-5-248 ~]# cd yunxing  [root@test-5-248 yunxing]# git init //初始化仓库  Initialized empty Git repository in /root/yunxing/.git/  [root@test-5-248 yunxing]# touch 1.php  [root@test-5-248 yunxing]# git add 1.php //将文件提交到本地仓库  [root@test-5-248 yunxing]# git commit -m "first commit"  [master (root-commit) ff87c72] first commit  0 files changed, 0 insertions(+), 0 deletions(-)  create mode 100644 1.php  create mode 100644 README.md  第一次提交,只是提交本机仓库 |

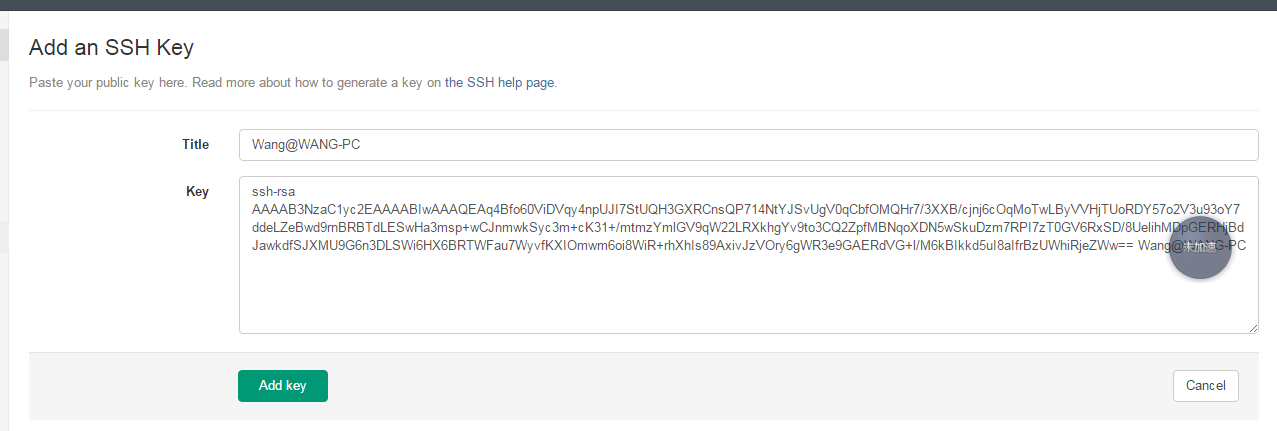
#### 2.7 提交ssh-key到gitlab

将开发者的ssh-key添加到gitlab上面,

创建了一个windows 本机的key

id\_rsa文件在C:\Users\Wang\.ssh下面

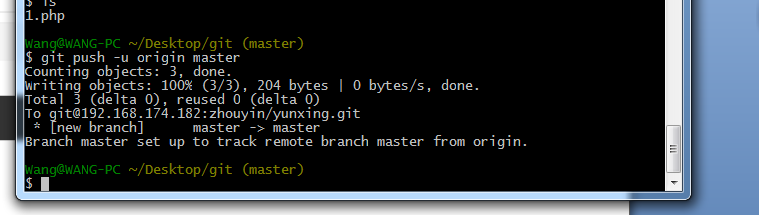
将key的内容复制到



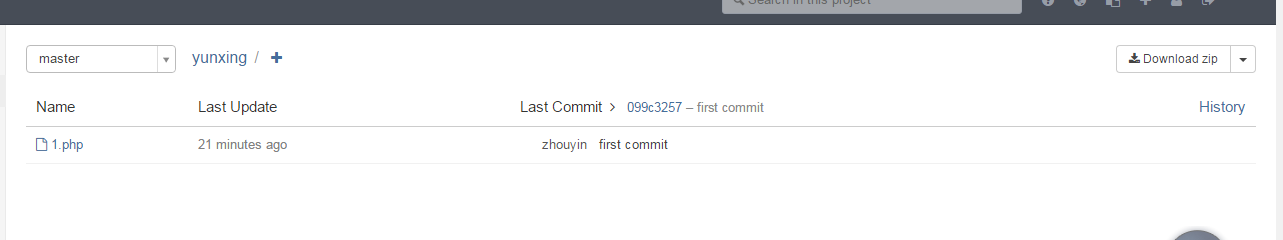
将文件提交到gitlab

#### 2.8 将文件提交到master

|  |
| --- |
| git push -u origin master |



提交后 gitlab出现你刚才的文件信息

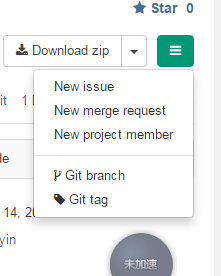


#### 2.9 创建一个分支 branch

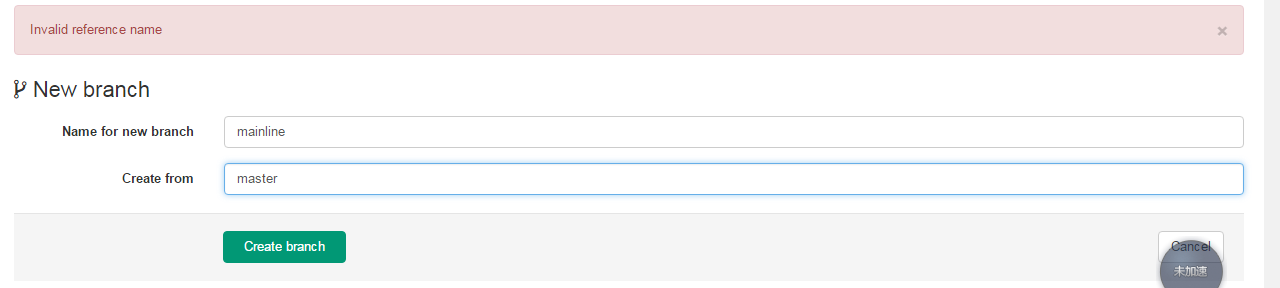
git中分支的概念:

git中分支相当于给仓库做了一次拷贝,拷贝仓库中的代码。开发者一般在master保留了完全的代码,与此同时还有一个develop专门用于后面的开发，当开发版本功能完善后,便可以将分支合并到master中。

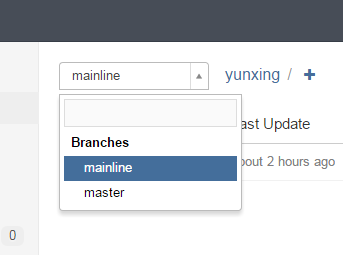
通过gitlab创建分支



填写分支的信息



从master 里面创建一个分支mainline,此时从master主线里面创建了一个分支mainline



#### 2.10 模拟多人协同开发

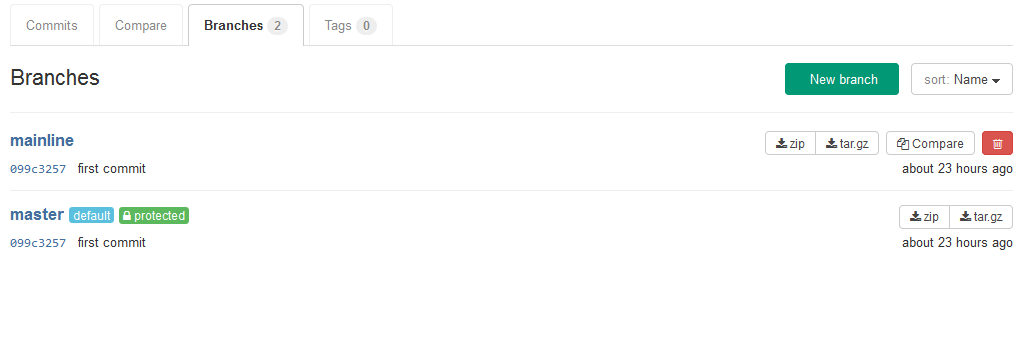
增加wangsheng这个用户，通过这个用户将项目克隆到本机

|  |
| --- |
| [root@saltslave-example ~]# git clone git@192.168.174.182:zhouyin/yunxing.git  Initialized empty Git repository in /root/yunxing/.git/  remote: Counting objects: 3, done.  remote: Total 3 (delta 0), reused 0 (delta 0)  Receiving objects: 100% (3/3), 204 bytes, done.  [root@saltslave-example ~]# ls -rlt  总用量 120  -rw-r--r--. 1 root root 11949 5月 18 01:33 install.log.syslog  -rw-r--r--. 1 root root 62826 5月 18 01:37 install.log  -rw-------. 1 root root 2193 5月 18 01:38 anaconda-ks.cfg  drwxr-xr-x. 2 root root 4096 5月 18 01:45 桌面  drwxr-xr-x. 2 root root 4096 5月 18 01:45 音乐  drwxr-xr-x. 2 root root 4096 5月 18 01:45 下载  drwxr-xr-x. 2 root root 4096 5月 18 01:45 文档  drwxr-xr-x. 2 root root 4096 5月 18 01:45 图片  drwxr-xr-x. 2 root root 4096 5月 18 01:45 视频  克隆了yungxing这个项目  drwxr-xr-x. 2 root root 4096 5月 18 01:45 模板  drwxr-xr-x. 2 root root 4096 5月 18 01:45 公共的  drwxr-xr-x. 3 root root 4096 9月 15 21:49 yunxing  [root@saltslave-example ~]# cd yunxing/  [root@saltslave-example yunxing]# ls  1.php |

程序员只能在自己的分支上面工作,不能修改主分支的代码。

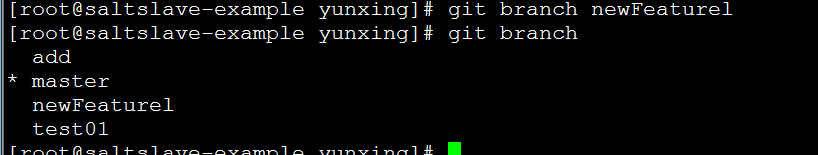
developer 的权限

master 设置为受保护的分支,mainline主线是另外的一个分支。



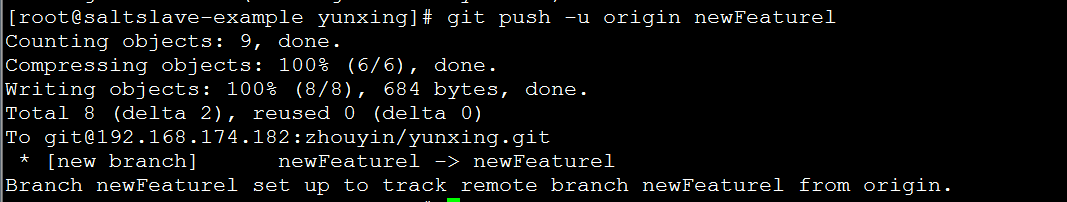
在自己的仓库里面创建自己的分支,并且切换自己分支中

|  |
| --- |
| git branch newFeaturel //创建分支 |
| git checkout -b newFeaturel //切换到newFeaturel 这个分支中 |

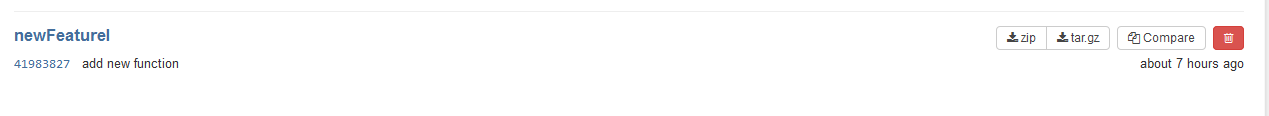




|  |
| --- |
| [root@saltslave-example yunxing]# touch 2.php  [root@saltslave-example yunxing]# git add 2.php  [root@saltslave-example yunxing]# git commit "add 2.php"  [root@saltslave-example yunxing]# git commit -m "add 2.php"  [newFeaturel ba414d7] add 2.php  0 files changed, 0 insertions(+), 0 deletions(-)  create mode 100644 2.php  git push -u origin newFeaturel |

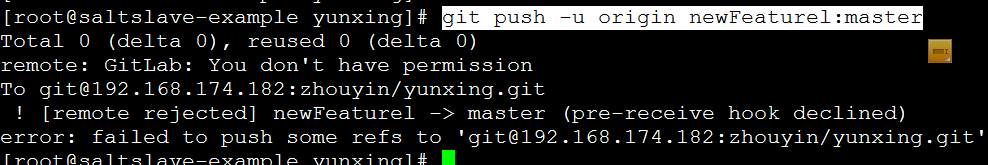


此时gitlab 出现newFeaturel

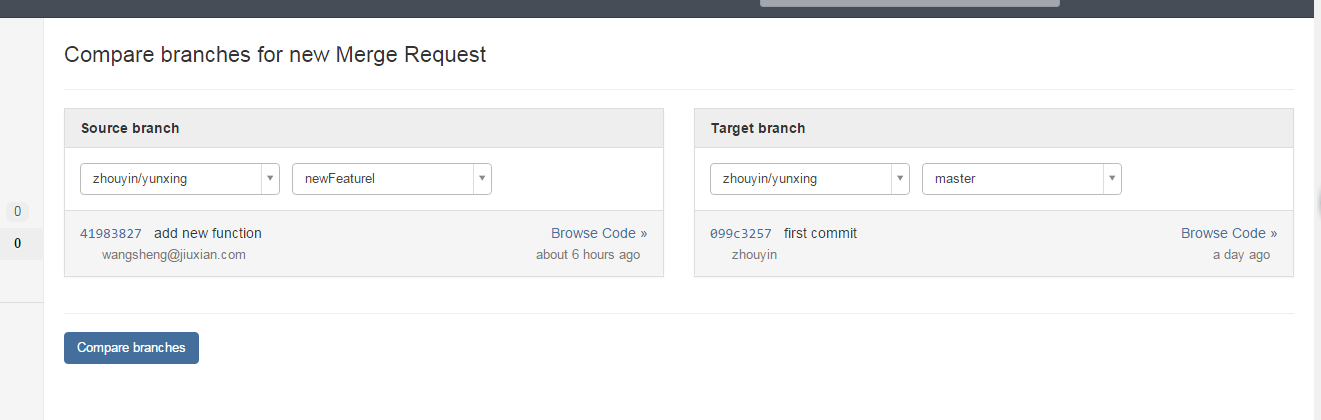


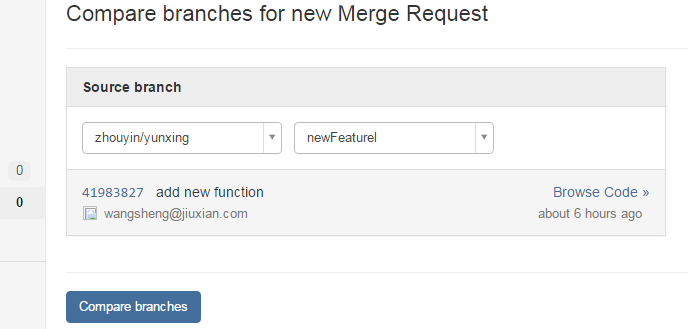
如果将 newFeaturel 推送到主线git push -u origin newFeaturel:master

如果将 newFeaturel 推送到master则会发生错误。

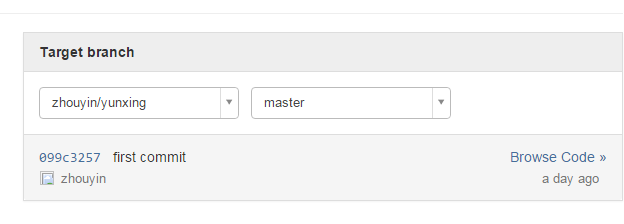


#### 2.11 分支合并

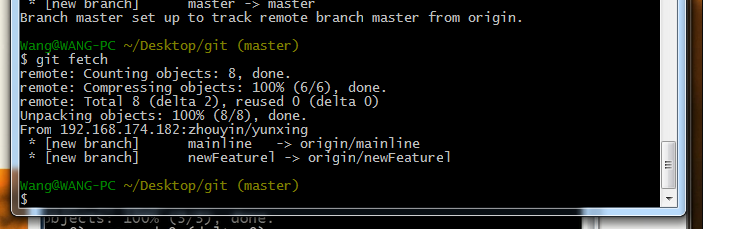




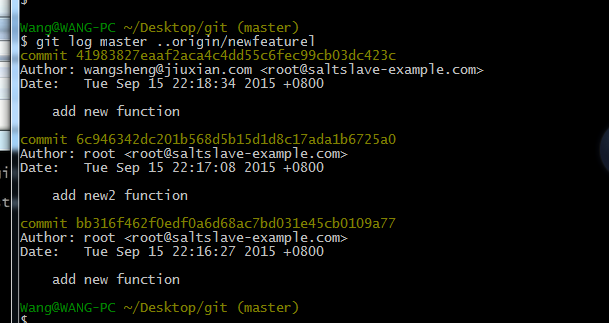
从newFeaturel 分支 合并到主分支上面



|  |
| --- |
| 管理员通过get fetch 获取所有的分支 |

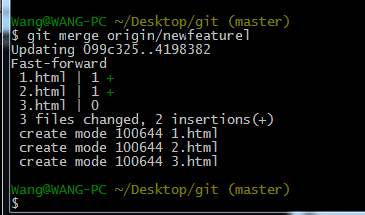


|  |
| --- |
| 查看日志 git log master ..origin/newFeaturel |



命令可以看出,我们在这个分支中提交了什么？

|  |
| --- |
| 仓库管理员的在本地仓库中，合并分支，将newfeaturel合并到master中, 执行命令 git merge origin/newfeaturel |

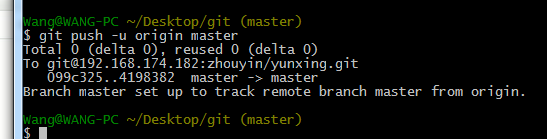


如果出现代码不一致的状况

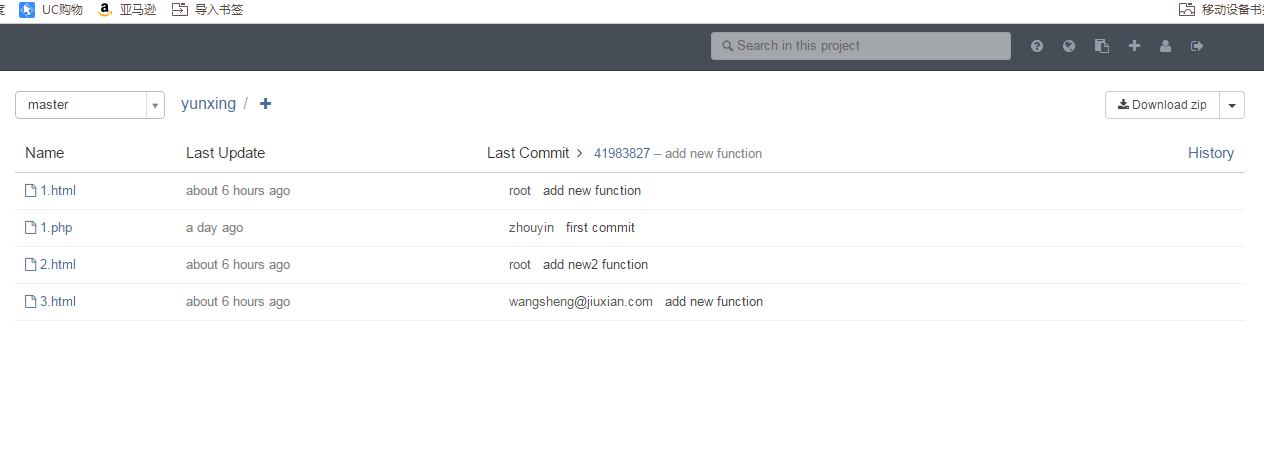
管理员检查代码有没有问题,则将master推送到gitlab中的主线中

#### 2.12 将本地主线master推送到gitlab中

|  |
| --- |
| git push -u origin master |



|  |
| --- |
| 将本地的仓库推送到远程服务器上  检查gitlab上面的文件 |



master 分支上面就出现了newFeaturel分支的文件

### 三 git常用操作

切换主线

git checkout master 切换到master 分支

删除本地分支

[root@saltslave-example yunxing]# git branch -d newFeaturel

warning: deleting branch 'newFeaturel' that has been merged to

'refs/remotes/origin/newFeaturel', but it is not yet merged to HEAD.

Deleted branch newFeaturel (was 4198382).