

企业级GitLab在Docker部署使用 - 包子wxi - 博客园

 [cnblogs.com/wei325/p/17106525.html](https://www.cnblogs.com/wei325/p/17106525.html)

企业级GitLab在Docker部署使用

[回到顶部](#)

一、部署gitlab

这里使用的是Centos8，安装Docker环境，这里不说了，参考：

<https://www.cnblogs.com/wei325/p/15139701.html>

gitlab有ce版和ee版，ce版为免费版本；ee版为企业版本，需要收费；企业使用ce版足够了，这里用ce版。

1)拉取Gitlab镜像

```
docker pull gitlab/gitlab-ce
```

```
last login: Thu Feb  9 17:36:10 2023 from 113.113.99.142
[root@VM-0-6-centos ~]# docker pull gitlab/gitlab-ce
Using default tag: latest
latest: Pulling from gitlab/gitlab-ce
846c0b181fff: Pull complete
749a4cfa2e97: Pull complete
019ffbe53fe6: Pull complete
1ea525f9dd74: Pull complete
3538a5d4f3f1: Pull complete
0de093023137: Pull complete
1a4090c1b369: Pull complete
3726ecd45362: Pull complete
Digest: sha256:2daa368303a130daf89fbc20b6529f918931ddf70c3e8fb8854efe84ecc4503c
Status: Downloaded newer image for gitlab/gitlab-ce:latest
docker.io/gitlab/gitlab-ce:latest
[root@VM-0-6-centos ~]#
```

<https://www.cnblogs.com/wei325>

2)启动Docker

先建3个目录用于把gitlab的文件挂载出来



#安装程序目录

```
mkdir -p /home/gitlab/etc
```

#数据目录

```
mkdir -p /home/gitlab/opt
```

#日志目录

```
mkdir -p /home/gitlab/log
```



启动docker



```
docker run \
  -itd \
  -p 8180:80 \
  -p 8122:22 \
  -v /home/gitlab/etc:/etc/gitlab \
  -v /home/gitlab/log:/var/log/gitlab \
  -v /home/gitlab/opt:/var/opt/gitlab \
  --restart always \
  --privileged=true \
  --name gitlab \
  gitlab/gitlab-ce
```



```
[root@VM-0-6-centos home]# mkdir -p /home/gitlab/etc
[root@VM-0-6-centos home]# ^C
[root@VM-0-6-centos home]# mkdir -p /home/gitlab/opt
[root@VM-0-6-centos home]# mkdir -p /home/gitlab/log
[root@VM-0-6-centos home]# docker run \
  -itd \
  -p 8180:80 \
  -p 8122:22 \
  -v /home/gitlab/etc:/etc/gitlab \
  -v /home/gitlab/log:/var/log/gitlab \
  -v /home/gitlab/opt:/var/opt/gitlab \
  --restart always \
  --privileged=true \
  --name gitlab \
  gitlab/gitlab-ce
e9905e7f0e43bcb18845355724390f6a8ef48b3ac72a167cf637d04edf3c5a49
[root@VM-0-6-centos home]#
```

8180端口：gitlab管理页面端口

8122端口：SSH端口，拉代码时的端口

-v xxx：把Docker里面的目录映射到本机上

查看启动的gitlab

```
[root@VM-0-6-centos ~]# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS	PORTS
e9905e7f0e43	gitlab/gitlab-ce	"/assets/wrapper"	gitlab	7 hours ago	Up 3 minutes (healthy)	443/tcp, 0.0.0.0:8122→22/tcp, :::8122→22/tcp, 0.0.0.0:8180→80/tcp, :::8180→80/tcp

3)修改配置

docker安装gitlab，创建项目时，默认项目URL访问地址显示的是容器主机地址(容器id)，所以我们需要修改为一个固定可以外部访问的地址及宿主机地址，这样才可以进行pull/push代码



```
vi /home/gitlab/etc/gitlab.rb

# 取消external_url注释，地址为宿主机地址，不需要设置端口
external_url 'http://1.15.63.120'

#ssh主机ip
gitlab_rails['gitlab_ssh_host'] = '1.15.63.120'
#ssh连接端口
gitlab_rails['gitlab_shell_ssh_port'] =8122


# 重载服务
[root@vanje-dev01 ~]# docker exec -t gitlab gitlab-ctl reconfigure
[root@vanje-dev01 ~]# docker exec -t gitlab gitlab-ctl restart
```



4)修改root密码

启动docker后，gitlab已经能访问了，但还没有设置密码

← → ↻ ⚠ 不安全 | 1.15.63.120:8180/users/sign_in 🔗 ☆ ⚙ □ 👤 更新 ⋮



GitLab Community Edition

Username or email

Password

☐ Remember me [Forgot your password?](#)

[Sign in](#)

<https://www.cnblogs.com/wei325/> Don't have an account yet? [Register now](#)



```
# 进入容器内部
docker exec -it gitlab /bin/bash
# 进入控制台
gitlab-rails console -e production
# 查询id为1的用户，id为1的用户是超级管理员
user = User.where(id:1).first
# 修改密码为root123456
user.password='root123456'
# 保存
user.save!
# 退出
exit
```



```
[root@VM-0-6-centos ~]# docker exec -it gitlab /bin/bash
root@e9905e7f0e43:/# gitlab-rails console -e production
-----
Ruby:      ruby 2.7.7p221 (2022-11-24 revision 168ec2b1e5) [x86_64-linux]
GitLab:    15.8.1 (383efe57adf) FOSS
GitLab Shell: 14.15.0
PostgreSQL: 13.8
-----[ booted in 55.68s ]
Loading production environment (Rails 6.1.6.1)
irb(main):001:0> user = User.where(id:1).first
=> #<User id:1 @root>
irb(main):002:0> user.password='root123456'
=> "root123456"
irb(main):003:0> user.save!
=> true
irb(main):004:0> exit
```

设置完再去登录，就能登录成功了！

← → ↻ ⚠ 不安全 | 1.15.63.120:8180

Search GitLab

Check your sign-up restrictions

Your GitLab instance allows anyone to register for an account, which is a security risk on public-facing GitLab instances. You should deactivate new sign ups if public users aren't expected to register for an account.

Deactivate Acknowledge

Projects New project

Yours 1 Starred 0 Explore Topics

Filter by name Name

All Personal

<https://www.cnblogs.com/wei325/>

M GitLab Instance / Monitoring Owner

This project is automatically generated and helps monitor this GitLab instance. [Learn more.](#)

☆ 0 🗑 0 📄 0 Updated 8 hours ago

5)服务器卡顿问题解决

这里用的是腾讯云服务器，2核4G，因为gitlab会用比较多的内存和cpu，这个配置装了后服务器变得卡顿，解决办法。

1、创建swap分区



```
#查看swap分区情况，如果看到没有东西就是没有swap分区
cat /proc/swaps
#创建swap分区，注意：创建swap大小为bs*count=4294971392(4G)，/data/swap目录若无则
找/mnt/swap
dd if=/dev/zero of=/data/swap bs=512 count=8388616
#把上文件的文件做成swap分区
mkswap /data/swap
#cat /proc/sys/vm/swappiness，如果不是60，则设置
sysctl -w vm.swappiness=60
#启用swap分区
swapon /data/swap
echo "/data/swap swap swap defaults 0 0" >> /etc/fstab
#再次查看swap分区
cat /proc/swaps
```



```
[root@VM-0-6-centos ~]# cat /proc/swaps
Filename                                Type              Size              Used              Priority
[root@VM-0-6-centos ~]# dd if=/dev/zero of=/data/swap bs=512 count=8388616
8388616+0 records in
8388616+0 records out
4294971392 bytes (4.3 GB, 4.0 GiB) copied, 26.2734 s, 163 MB/s
[root@VM-0-6-centos ~]# mkswap /data/swap
mkswap: /data/swap: insecure permissions 0644, fix with: chmod 0600 /data/swap
Setting up swapspace version 1, size = 4 GiB (4294967296 bytes)
no label, UUID=8b06a4cc-2501-4ca0-a04b-d93aca3f0646
[root@VM-0-6-centos ~]# cat /proc/sys/vm/swappiness
60
[root@VM-0-6-centos ~]# swapon /data/swap
swapon: /data/swap: insecure permissions 0644, 0600 suggested.
[root@VM-0-6-centos ~]# echo "/data/swap swap swap defaults 0 0" >> /etc/fstab
[root@VM-0-6-centos ~]# cat /proc/swaps
Filename                                Type              Size              Used              Priority
/data/swap                             file              4194304           4184              -2
[root@VM-0-6-centos ~]#
```

<https://www.cnblogs.com/wei325/>

2、修改配置把进程数据变少

在挂载出来的本机文件上改，目录为/home/gitlab/etc/gitlab.rb，修改完重启容器生效



```
vi /home/gitlab/etc/gitlab.rb
#把下面3个取消注释，工作进程减少，最低为2
puma['worker_processes'] = 2
puma['min_threads'] = 2
puma['max_threads'] = 2

# 重载服务
[root@vanje-dev01 ~]# docker exec -t gitlab gitlab-ctl reconfigure
[root@vanje-dev01 ~]# docker exec -t gitlab gitlab-ctl restart
```



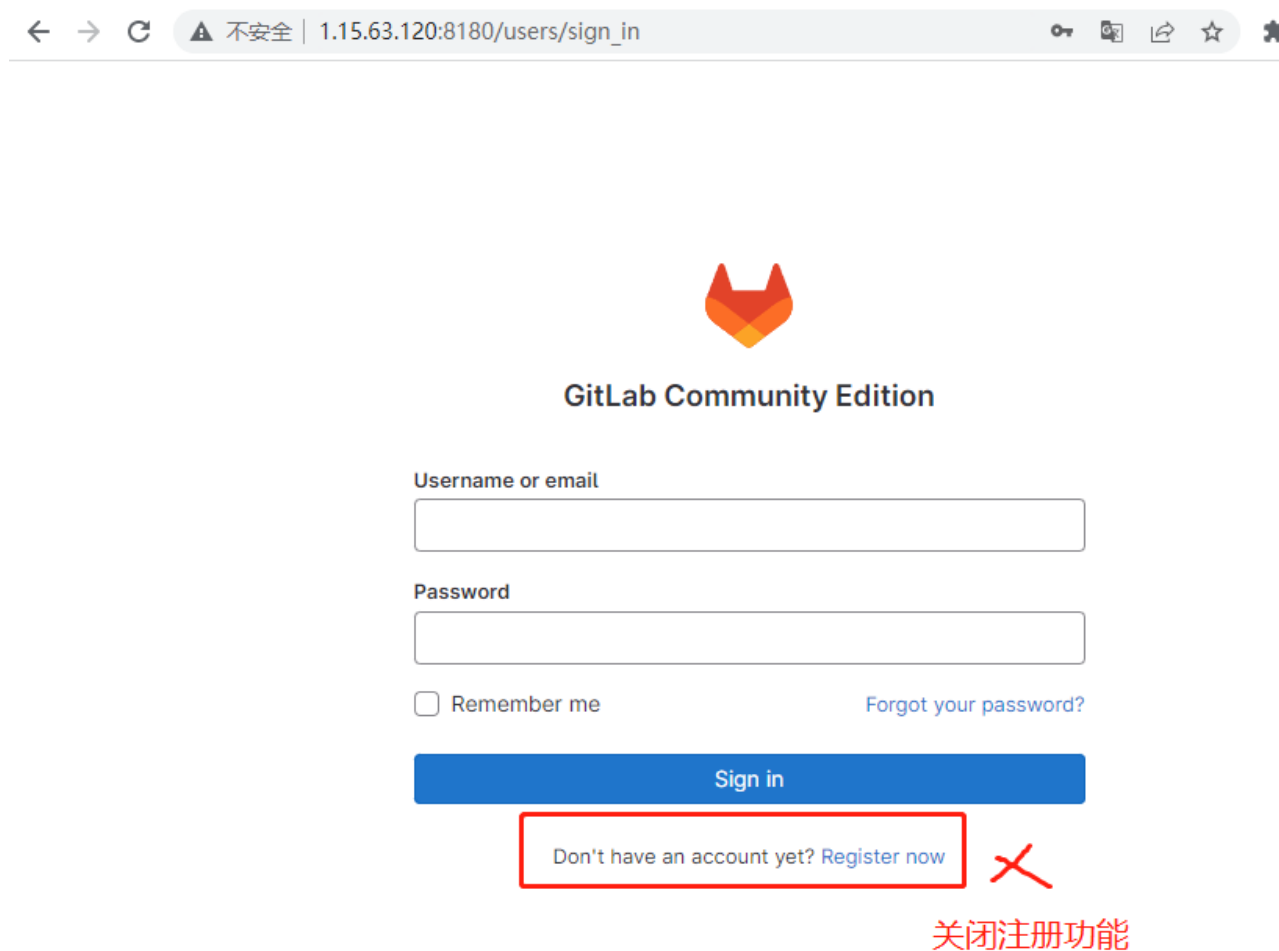
做完这些，服务器占用资源就降下来了，也不会卡顿了。

[回到顶部](#)


二、使用gitlab

1)设置关闭自动创建账号功能

gitlab默认是开放注册账号功能的，在企业里面使用是不允许的，用户的账号是通过管理员创建出来的



← → ↻ ⚠ 不安全 | 1.15.63.120:8180/users/sign_in 🔑 📄 📁 ☆ 🏠



GitLab Community Edition

Username or email

Password

☐ Remember me [Forgot your password?](#)

[Sign in](#)

Don't have an account yet? [Register now](#)

关闭注册功能

管理员登录进后台

← → ↻ 1.15.63.120:8180/admin/application_settings/general#js-signup-settings

Search GitLab

Admin Area

- Overview
- CI/CD
- Analytics
- Monitoring
- Messages
- System Hooks
- Applications
- Abuse Reports
- Deploy Keys
- Labels
- Settings**
 - General**
 - Integrations
 - Repository
 - CI/CD

Account and limit Expand

Set projects and maximum size limits, session duration, user options, and check feature availability for namespace plan.

Diff limits Expand

Set size limits for displaying diffs in the browser.

Sign-up restrictions Collapse

Configure the way a user creates a new account.

☒ Sign-up enabled
Any user that visits http://e9905e7f0e43:8180/users/sign_in can create an account.

☒ Require admin approval for new sign-ups
Any user that visits http://e9905e7f0e43:8180/users/sign_in and creates an account must be explicitly approved by an administrator before they can sign in. Only effective if sign-ups are enabled.

Email confirmation settings

☐ Hard
Send a confirmation email during sign up. New users must confirm their email address before they can log in.

☒ Off
New users can sign up without confirming their email address.

去掉勾选

<https://home.cnblogs.com/u/wei325/>

2)创建项目

← → ↻ 1.15.63.120:8180

Search GitLab

Projects

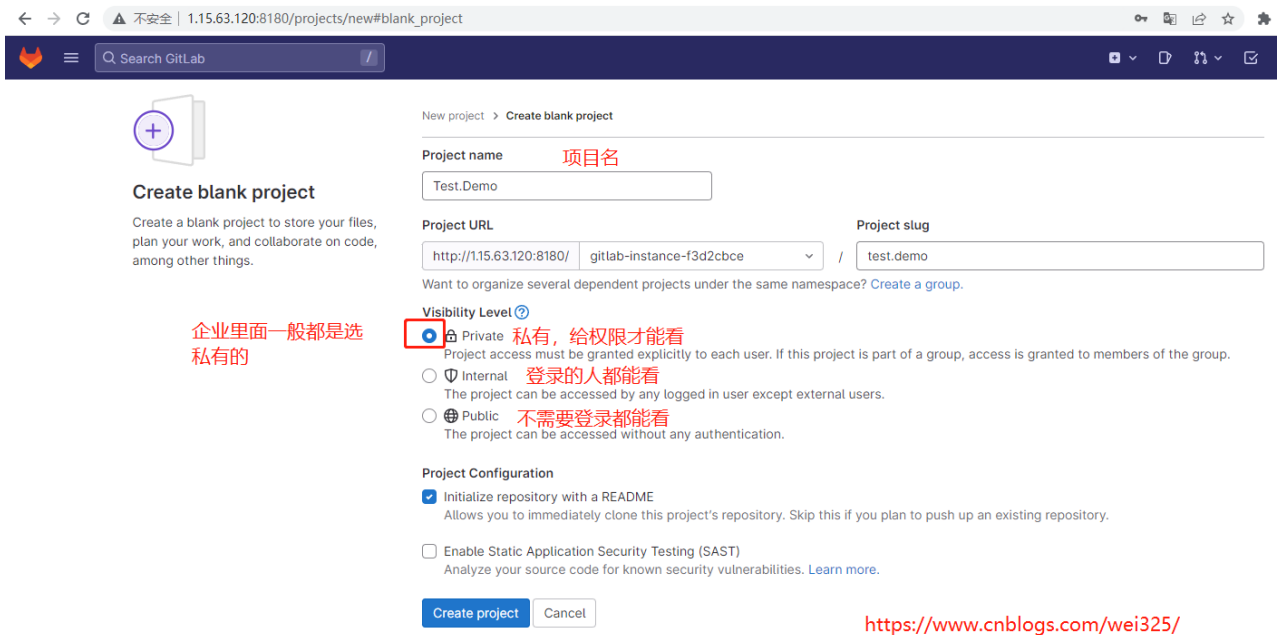
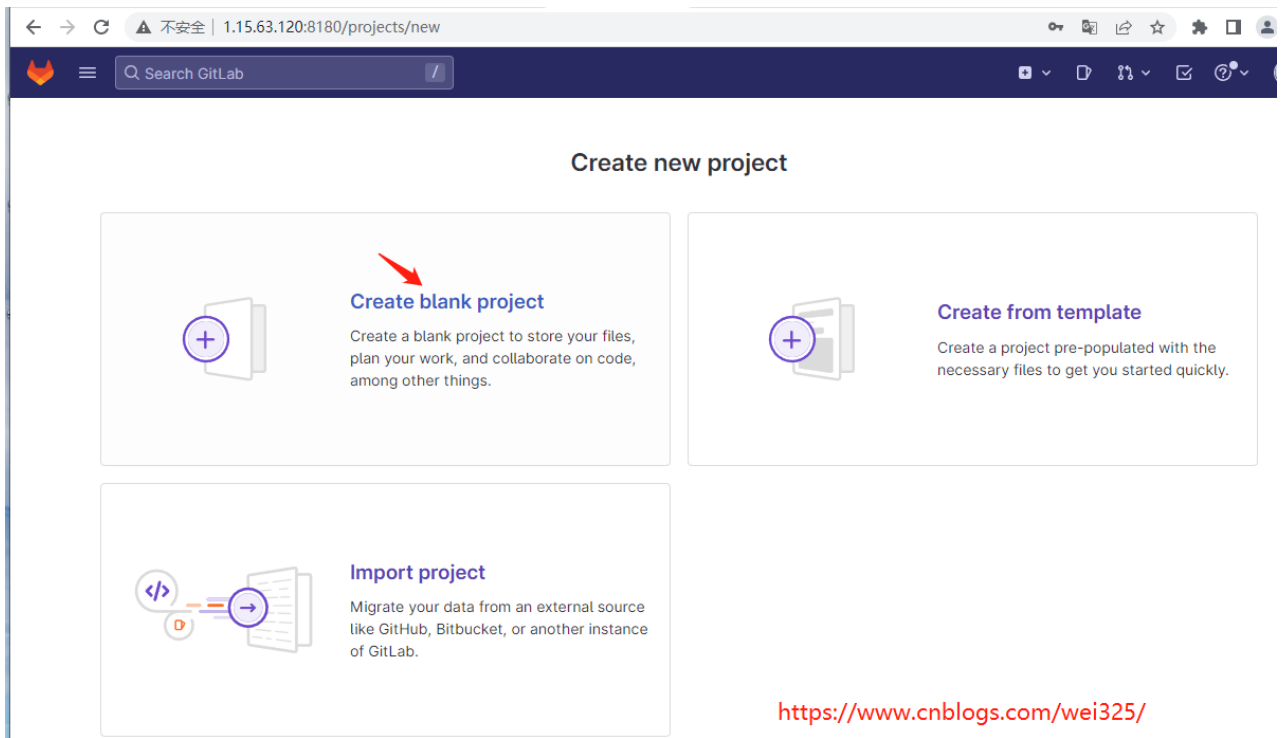
[New project](#)

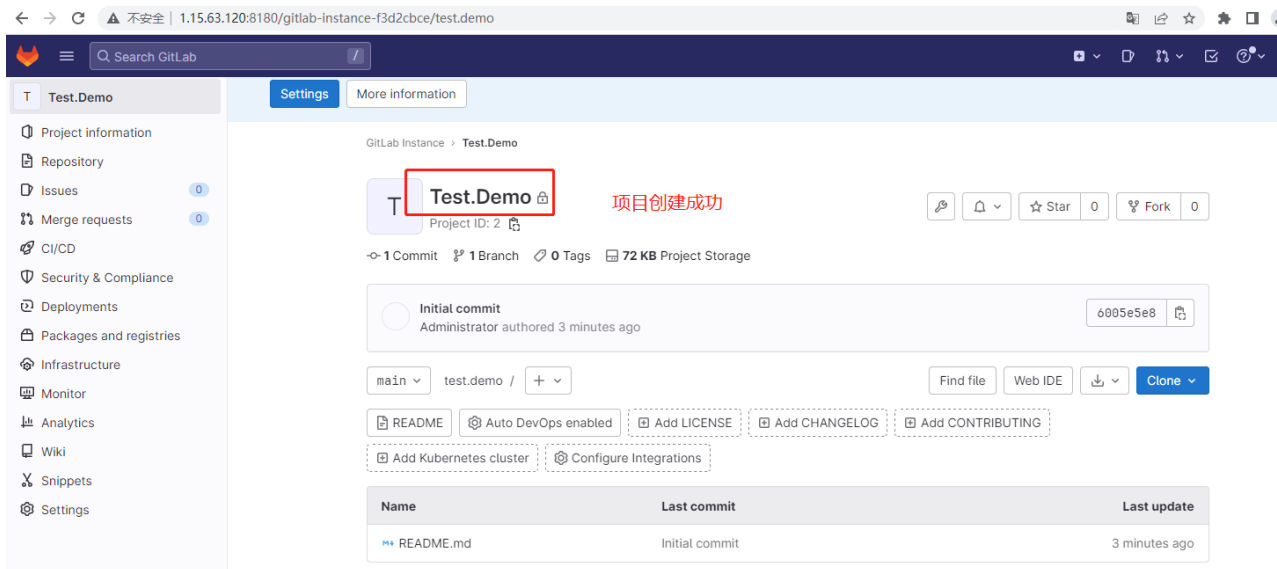
Yours 1 Starred 0 Explore Topics Filter by name Name

All Personal

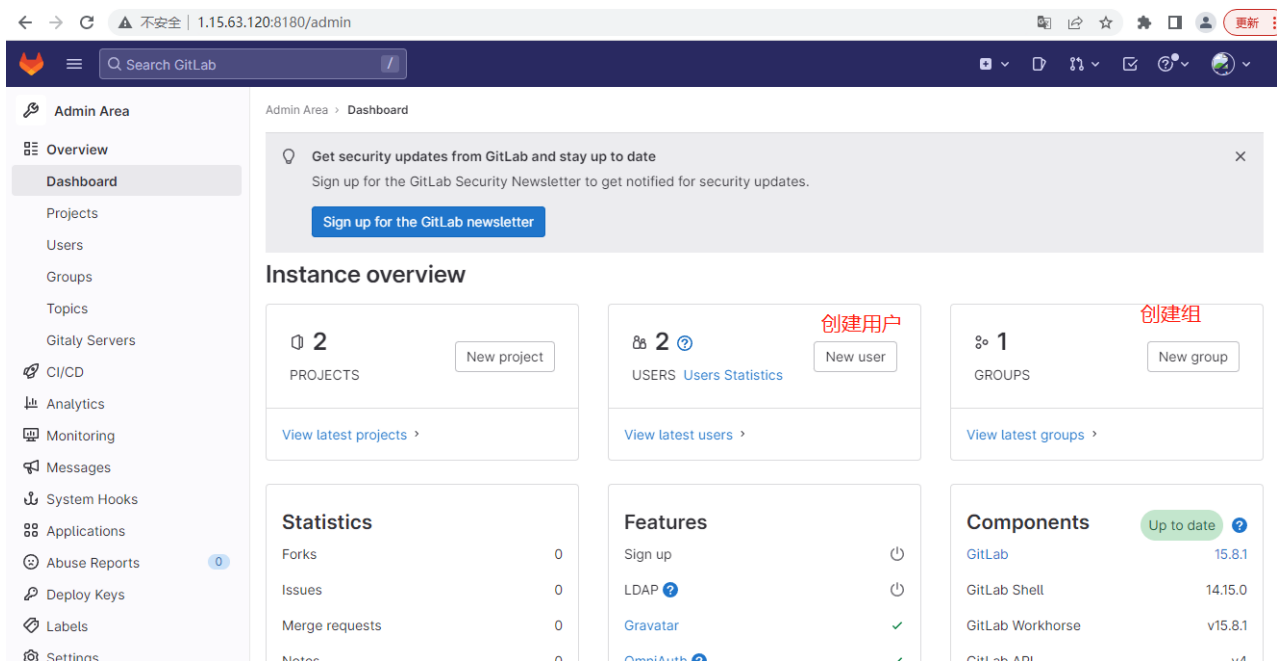
M GitLab Instance / **Monitoring** Owner ☆ 0 🍴 0 📄 0 Updated 13 hours ago

This project is automatically generated and helps monitor this GitLab instance. [Learn more.](#)





3)创建用户，组群，给项目权限



→ ↻ ⚠ 不安全 | 1.15.63.120:8180/admin/users/test/edit

Search GitLab

Admin Area

Overview

Dashboard

Projects

Users

Groups

Topics

GitLab Servers

CI/CD

Analytics

Monitoring

Messages

System Hooks

Applications

Abuse Reports 0

Deploy Keys

Labels

Settings

Admin Area > Users

Edit user: test

Account

Name
test

Username
test

Email
test@qq.com

Password

Password
.....

Password confirmation
.....

Access

Projects limit
100000

☒ Can create group

☐ Private profile

Access level

把用户test加入到Test.Demo的项目里面

← → ↻ ⚠ 不安全 | 1.15.63.120:8180/gitlab-instance-f3d2cbce/test.demo/-/project_members

Search GitLab

Test.Demo

Project information

Activity

Labels

Members

Repository

Issues 0

Merge requests 0

CI/CD

Security & Compliance

Deployments

Packages and registries

Infrastructure

Monitor

Analytics

Wiki

Snippets

GitLab Instance > Test.Demo > Members


Project members

You can invite a new member to **Test.Demo** or invite another group.

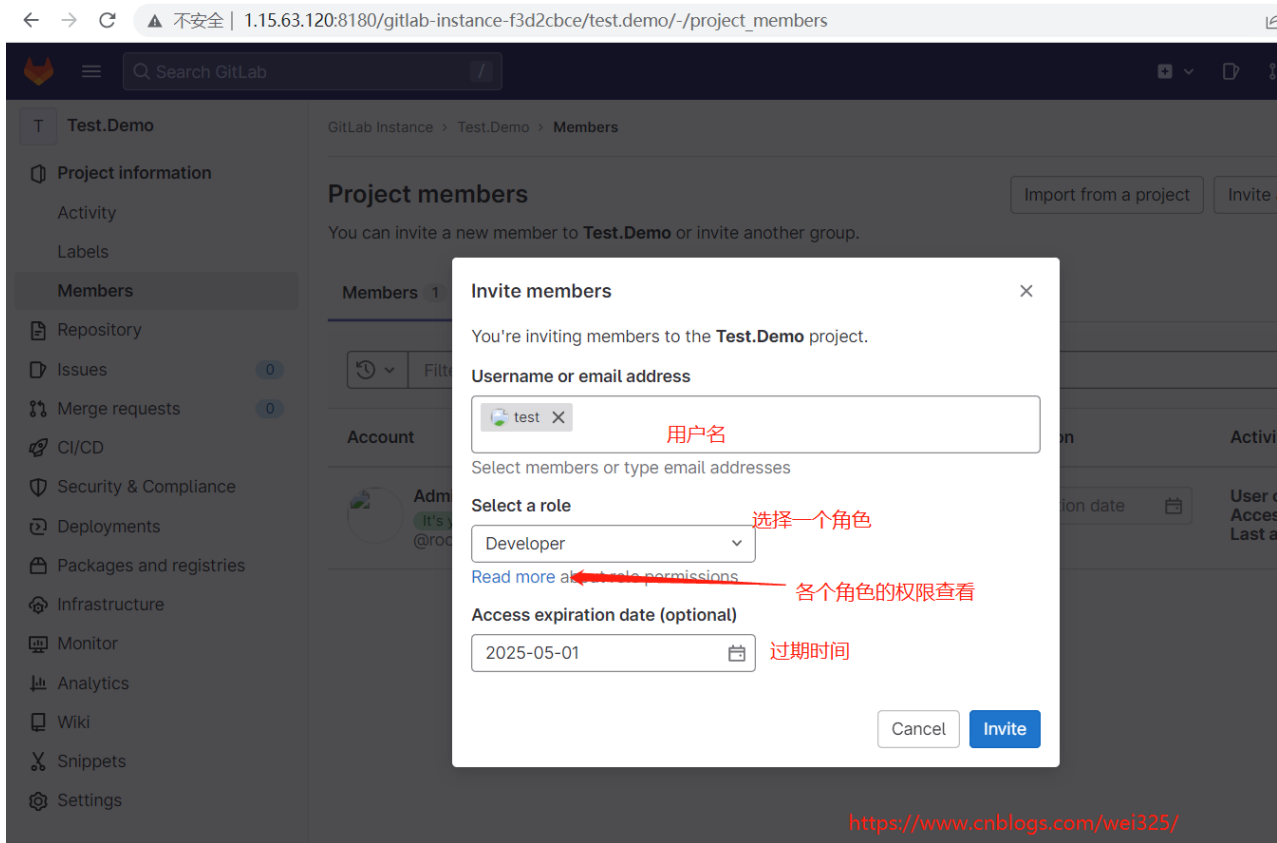
Import from a project Invite a group **Invite members**

Members 1

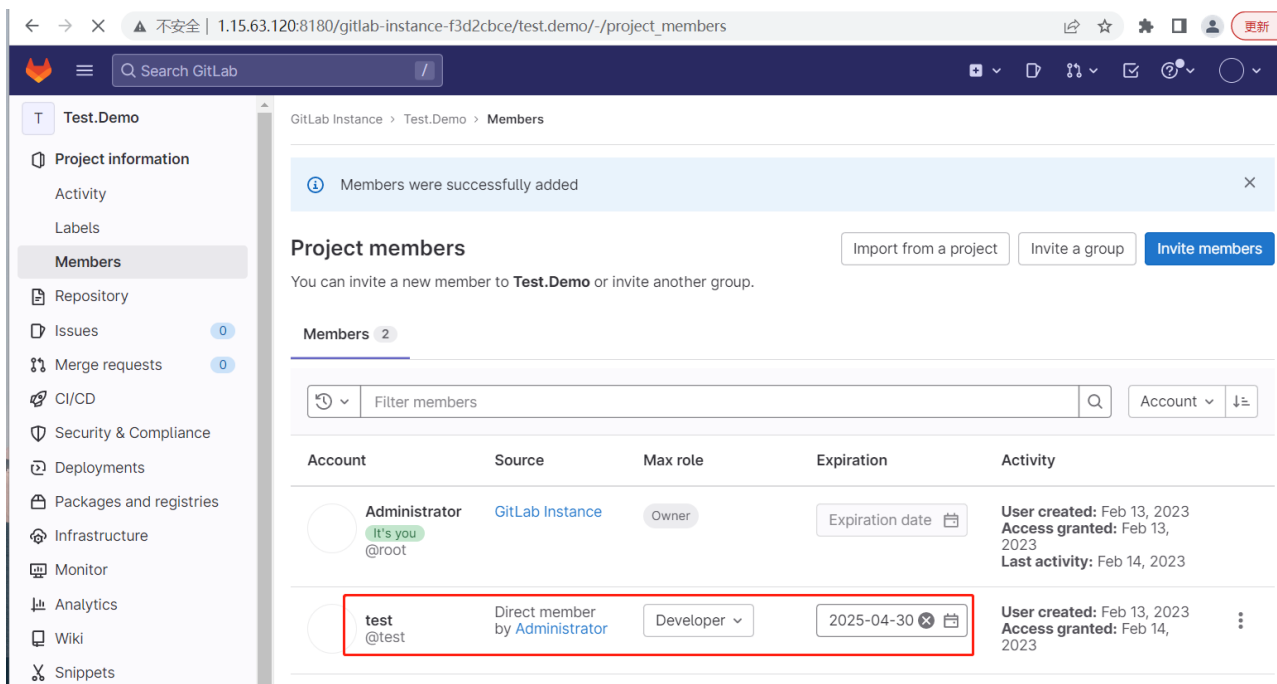
Filter members

Account	Source	Max role	Expiration	Activity
 Administrator It's you @root	GitLab Instance	Owner	Expiration date	User created: Feb 13, 2023 Access granted: Feb 13, 2023 Last activity: Feb 14, 2023

<https://www.cnblogs.com/wei325/>



成功把用户加了进来





GitLab用户权限管理：

GitLab用户在组中有五种权限：Guest、Reporter、Developer、Maintainer、Owner

1.Guest：可以创建issue、发表评论、不能读写版本库

2.Reporter：可以克隆代码，不能提交，QA、PM可以赋予这个权限

3.Developer：可以克隆代码、开发、提交、push、研发人员可以赋予这个权限

4.Maintainer：可以创建项目、添加 tag、保护分支、添加项目成员、编辑项目、核心研发负责人可以赋予这个权限

5.Owner：可以设置项目的访问权限-Visibility Level、删除项目、迁移项目、管理组成员、项目经理，部门经理可以赋予这个权限

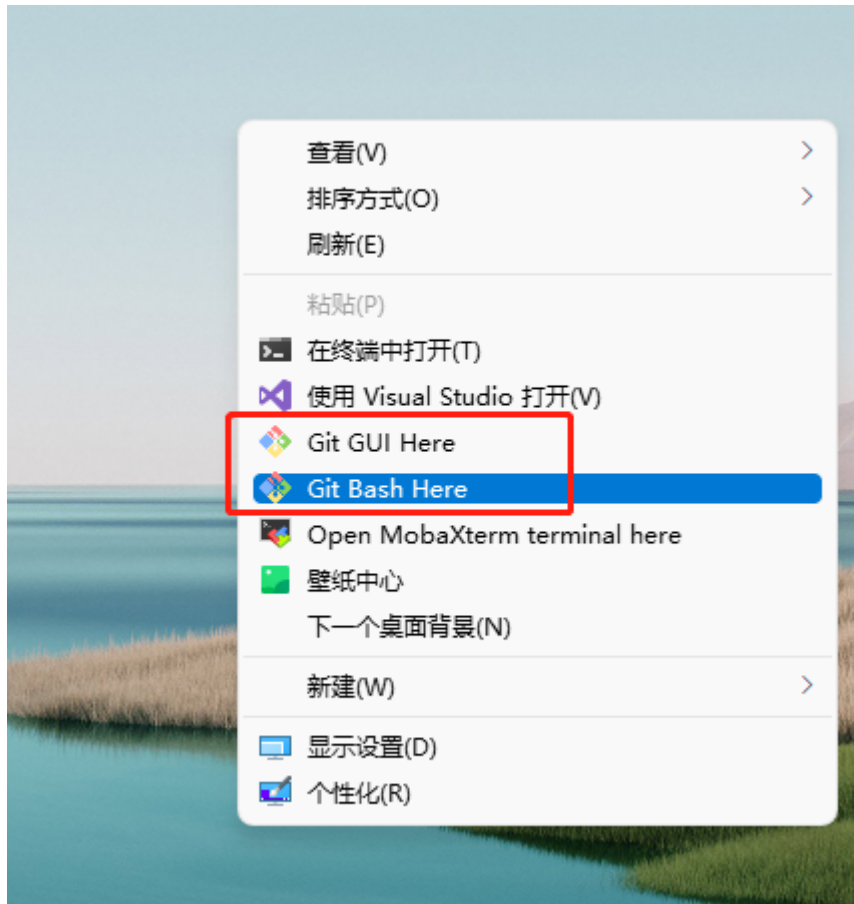
[回到顶部](#)

三、Git拉取和提交GitLab代码

1)安装git客户端

官网地址：<https://git-scm.com/downloads/>

下载后，一直下一步安装即可。安装完成后，桌面鼠标右键能看到如下图的两个菜单



2)git生成SSH Key

git客户端要连接仓库，就要先生成ssh key，ssh key有公钥和私钥，生成后把公钥填到git站点上，现在生成ssh key

点击Git Bash Here

2.1设置用户名

```
git config --global user.name 'test'
```

test为自己的用户名，可以随便填，注意提交记录的时候会显示这个用户名，所以一般填员工的姓名好辨认谁提交了代码



```
MINGW64:/c/Users/28474/Desktop
28474@wxlpc MINGW64 ~/Desktop
$ git config --global user.name 'test'
28474@wxlpc MINGW64 ~/Desktop
$ |
```

<https://www.cnblogs.com/wei325/>

2.2设置邮箱

```
git config --global user.email 'test@qq.com'
```

这个邮箱也是可以随便填，一般填员公在公司的邮箱

上两步生成的信息会在C:\Users\账号 下生成.gitconfig配置文件，特别注意此文件不能删除，此文件记录了账号和邮箱。



2.3生成ssh key

```
ssh-keygen -t rsa -C "test@qq.com"
```

填刚才的邮箱，输入后一直回车即可

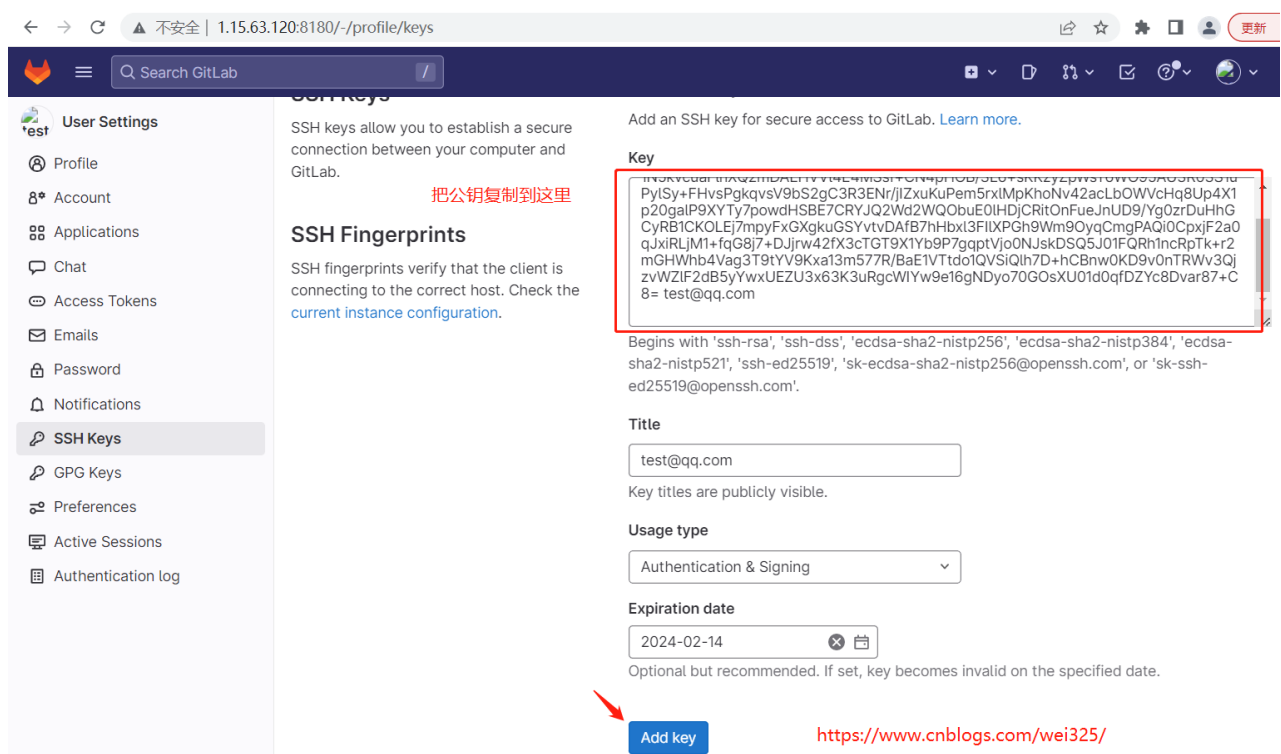
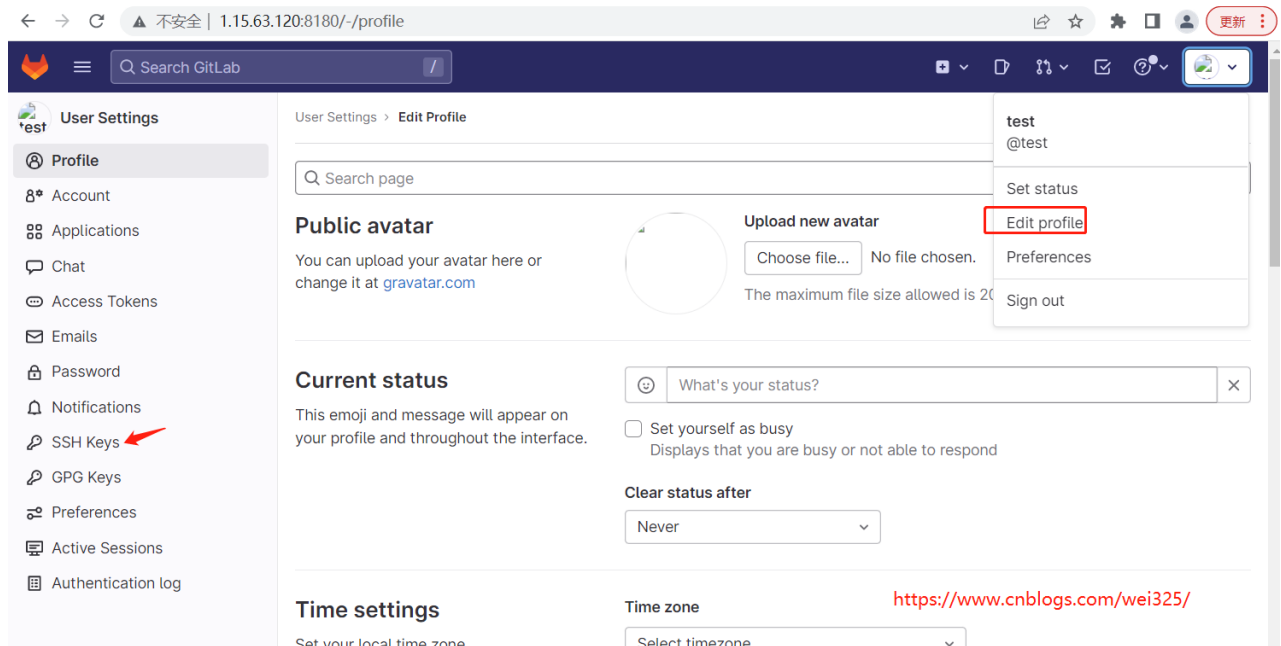
生成的ssh key信息保存在 C:\Users\账号下的.ssh文件夹里



3)把生成的公钥填到gitlab

git客户端要和gitlab连接，需要把公钥里面的内容填到gitlab上

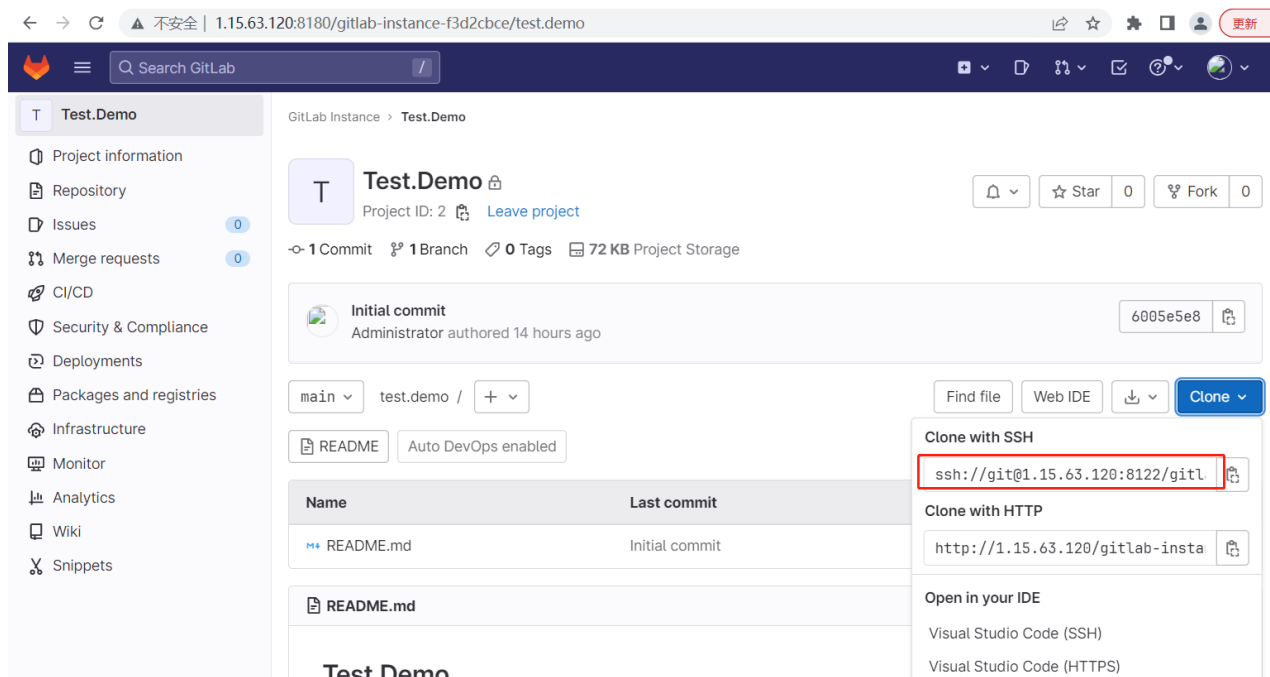
回到gitlab，每个员工都会有独立的ssh key，这里我登录test账号，表示把本机的git和test账号关联



4) 拉取项目和提交文件

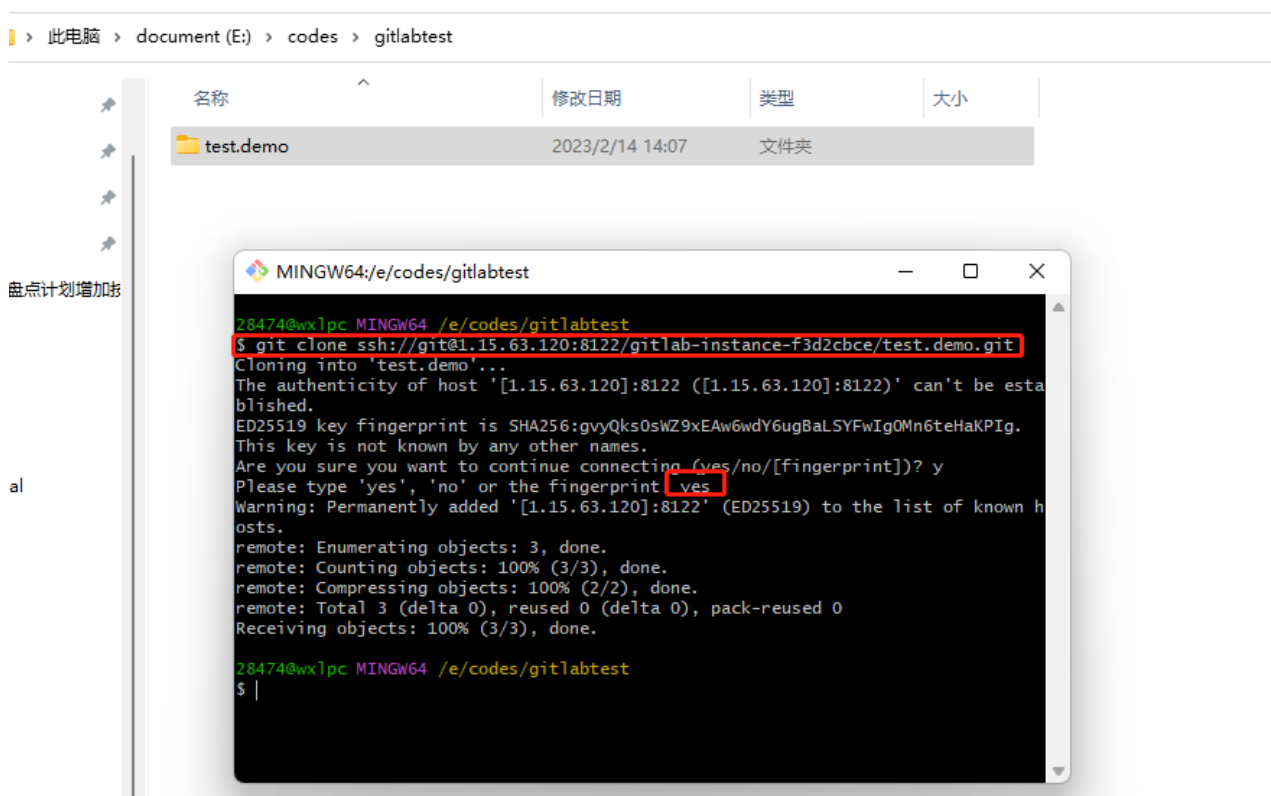
4.1 克隆项目

找到项目的ssh地址，也可以通过http地址

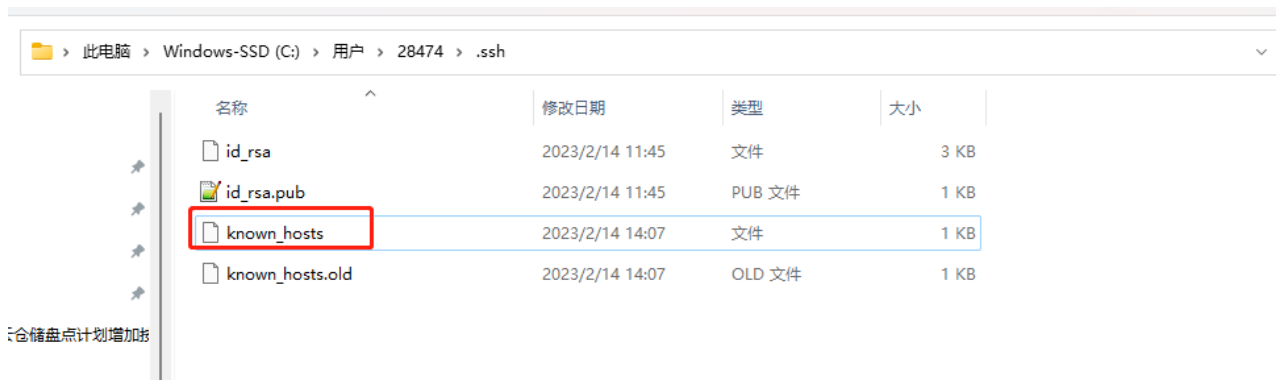


在本机上建一个文件夹放拉取的项目，在文件夹的路径里打开 Git Bash

git clone ssh地址



第一次拉取时会报一个无法验证主机真实性，输入yes，然后会在 ssh key的文件夹下的 known_hosts里面生成gitlab信息，下次就不会询问了。



克隆完成后，可以看到项目已经被拉下来了，因为是一个空项目，所以还没有东西，只有一份描述文档



4.2提交文件

我们新增加一个文件 abc.txt



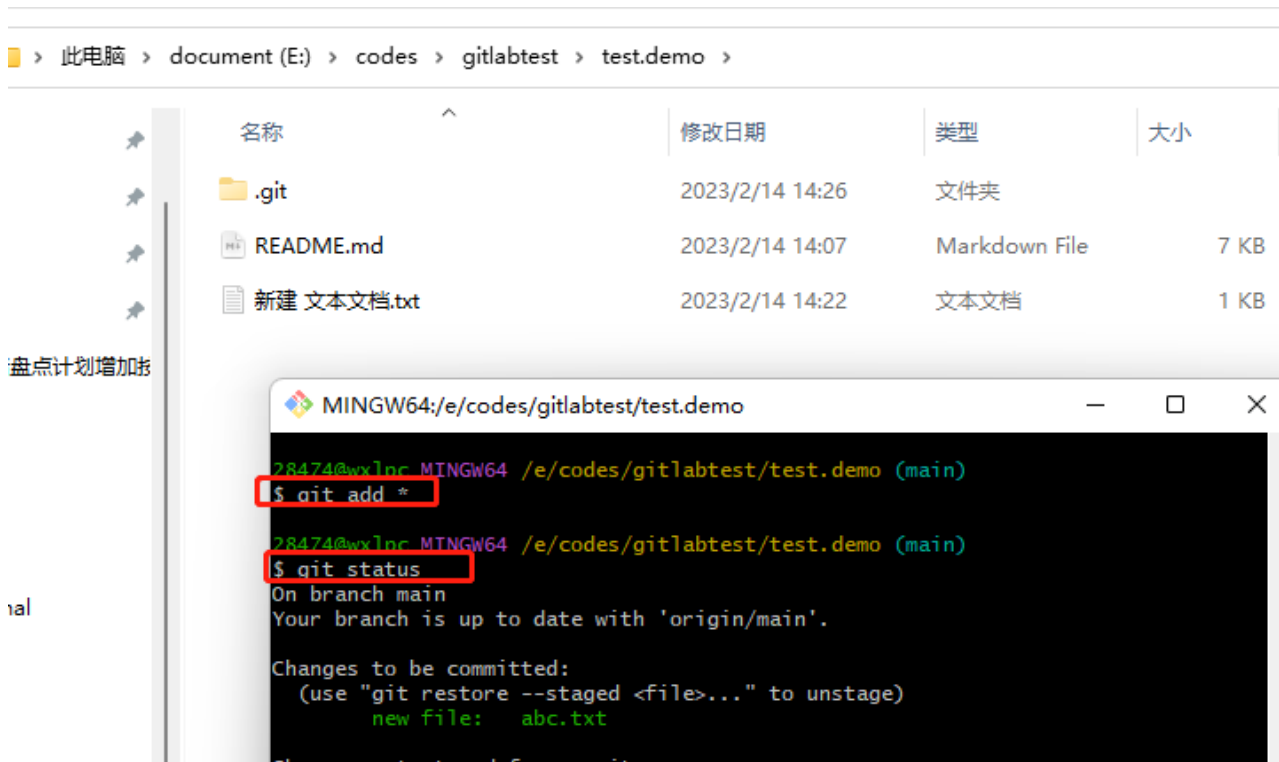
现在的文件是在工作区，然后把文件add到暂存区

#把文件添加到暂存区

```
git add *
```

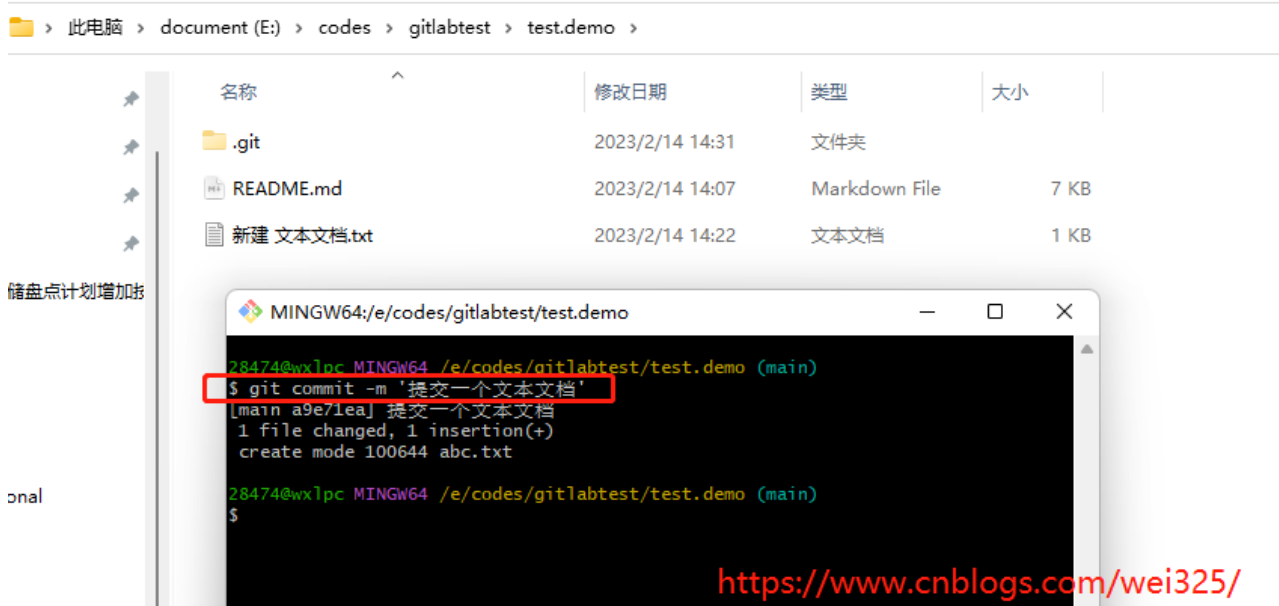
#查看文件变化状态

```
git status
```



4.3提交文件到本地仓库

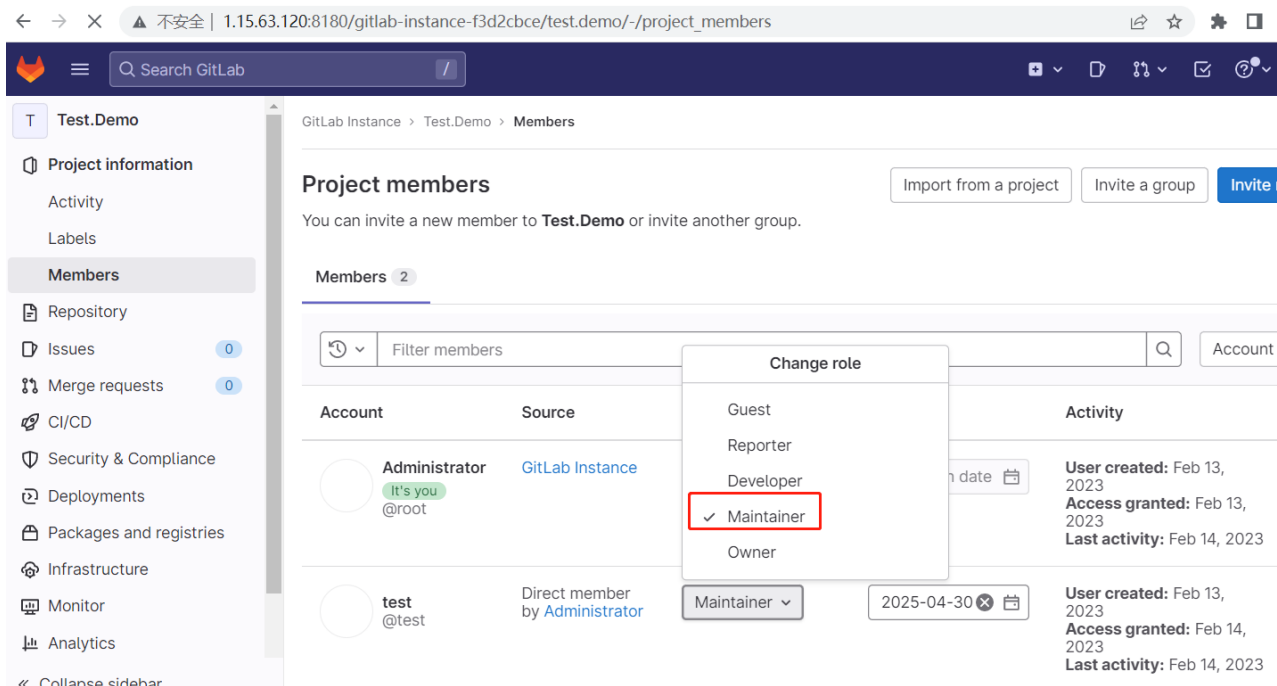
#提交修改到本地仓库
git commit -m '提交一个文本文档'



4.4把本地仓库同步到远程仓库

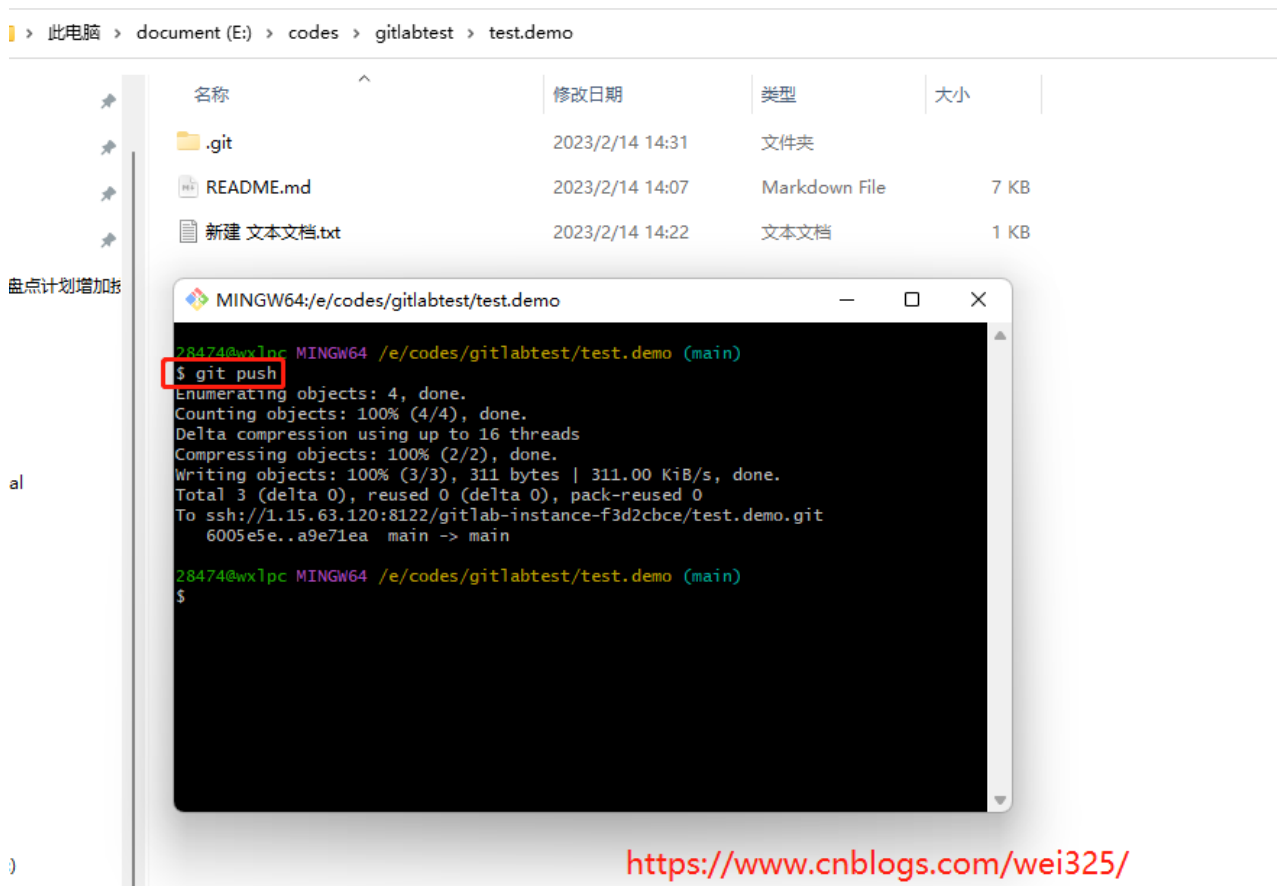
因为这里是一个main分支，developer不能推送到main分支，所以这里给一下maintainer权限给test用户。正常企业里面开发，开发人员只有开发分支(dev)，

测试分支(test)的权限，main分支只能看，不能提交代码的，因为main分支要保持着随时能发布，不能签提交代码，要由项目负责人从测试分支(test)合并 代码到main分支。



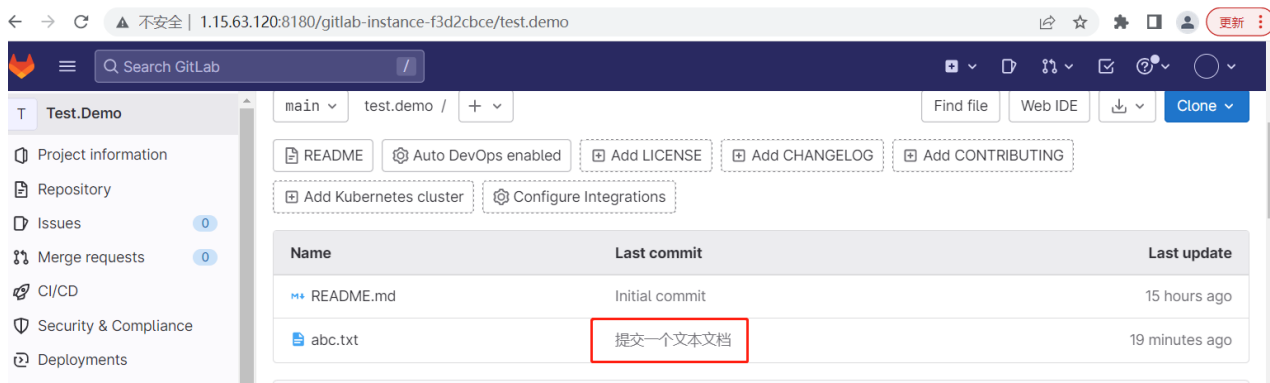
推送代码

git push



<https://www.cnblogs.com/wei325/>

这里把本地仓库同步到远程仓库，到远程仓库看一下就能看到提交记录了



如果想拉取最新的代码，则用git pull

到这里gitlab部署和使用就完成了，git如果命令行方式觉得麻烦，则可以用小乌龟git的界面可视化操作

小乌龟git下载地址为：<https://tortoisegit.org/download>

其余的开发IDE也都会自带有git插件，这个就不多说啦。

标签: [DevOps](#), [Docker](#), [Git](#)

好文要顶 已关注 收藏该文



包子wxi

粉丝 - 246 关注 - 41

我在关注他 取消关注

8 K

0

[关注](#) | [顶部](#) | [评论](#)

« 上一篇: [SQL Server、MySQL主从搭建，EF Core读写分离代码实现](#)