

## 1、GitLab 安装

```
// 安装GitLab依赖包
[root@localhost home]# yum install -y curl policycoreutils-python openssh-server
openssh-clients

// 查看centos的版本，本教程的是7.5，是centos7
[root@localhost home]# cat /etc/centos-release
CentOS Linux release 7.5.1804 (Core)

下载对应操作系统的gitlab的rpm包，用浏览器打开（本教程用gitlab-ce-11.5.7）
centos6系统的下载地址：https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el6
centos7系统的下载地址：https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el7

// centos7的执行这个（本教程用的centos7）
[root@localhost home]# wget https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el7/gitlab-ce-11.5.7-ce.0.el7.x86_64.rpm

// centos6的执行这个
[root@localhost home]# wget https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el6/gitlab-ce-11.5.7-ce.0.el6.x86_64.rpm

// 这里我先进来home目录，然后执行了wget
[root@localhost home]# ll
total 454180
-rw-r--r--. 1 root root 465080251 Jan 16 10:36 gitlab-ce-11.5.7-ce.0.el7.x86_64.rpm
drwx-----. 3 houtai houtai 78 Jan 28 22:37 houtai
drwxr-xr-x. 6 root root 86 Jan 25 01:37 svnProject

// 如果没有wget命令，则安装一下，然后再下载rpm
[root@localhost home]# yum install wget

// 通过yum安装本地GitLab的rpm包
[root@localhost home]# yum -y localinstall gitlab-ce-11.5.7-ce.0.el7.x86_64.rpm
```

安装完成有个大大的 logo

55

## 2、修改 GitLab 配置文件

```
// 修改GitLab的主配置文件
[root@localhost home]# vi /etc/gitlab/gitlab.rb
```

这里配置 `external_url` 为 <http://192.168.0.21:9999> (指定 gitlab 运行端口, 后续访问 gitlab、代码提交都是这个 url), 如果用域名则直接用域名, 但保证域名是可用的。

```
## GitLab configuration settings
##! This file is generated during initial installation and is not modified
##! during upgrades.
##! Check out the latest version of this file to know about the different
##! settings that can be configured by this file, which may be found at:
##! https://gitlab.com/gitlab-org/omnibus-gitlab/raw/master/files/gitlab-config-template/gitlab.rb.template

## GitLab URL
##! URL on which GitLab will be reachable.
##! For more details on configuring external_url see:
##! https://docs.gitlab.com/omnibus/settings/configuration.html#configuring-the-external-url-for-gitlab
external_url 'http://gitlab.example.com'
external_url 'http://192.168.0.21:9999'

## Roles for multi-instance GitLab
##! The default is to have no roles enabled, which results in GitLab running as an all-in-one instance.
##! Options:
##!   redis_sentinel_role redis_master_role redis_slave_role geo_primary_role geo_secondary_role
##! For more details on each role, see:
##! https://docs.gitlab.com/omnibus/roles/README.html#roles
##!
# roles ['redis_sentinel_role', 'redis_master_role']

## Legend
##! The following notations at the beginning of each line may be used to
##! differentiate between components of this file and to easily select them using
##! a regex.
##! ## Titles, subtitles etc
##! ##! More information - Description, Docs, Links, Issues etc.
##! Configuration settings have a single # followed by a single space at the
##! beginning; Remove them to enable the setting.

##! **Configuration settings below are optional.**
##! **The values currently assigned are only examples and ARE NOT the default
##! values.**
```

### 3、初始化 GitLab

#### 初始化

```
// 初始化GitLab，要等待挺久的，不断的输出  
[root@localhost home]# gitlab-ctl reconfigure
```

最后一定要看到这句 Chef Client finished，如果是 Chef Client failed 就要检查哪里报错了，一般错误信息会有红色字体提示。

```
- execute the ruby block restart redis-exporter svlogd configuration  
* ruby_block[reload redis-exporter svlogd configuration] action create  
- execute the ruby block reload redis-exporter svlogd configuration  
Recipe: gitlab::prometheus  
* service[prometheus] action restart  
- restart service service[prometheus]  
* execute[reload prometheus] action run  
- execute /opt/gitlab/bin/gitlab-ctl hup prometheus  
* ruby_block[restart prometheus svlogd configuration] action create  
- execute the ruby block restart prometheus svlogd configuration  
* ruby_block[reload prometheus svlogd configuration] action create  
- execute the ruby block reload prometheus svlogd configuration  
Recipe: gitlab::alertmanager  
* service[alertmanager] action restart  
- restart service service[alertmanager]  
* ruby_block[restart alertmanager svlogd configuration] action create  
- execute the ruby block restart alertmanager svlogd configuration  
* ruby_block[reload alertmanager svlogd configuration] action create  
- execute the ruby block reload alertmanager svlogd configuration  
Recipe: gitlab::postgres-exporter  
* service[postgres-exporter] action restart  
- restart service service[postgres-exporter]  
* ruby_block[restart postgres-exporter svlogd configuration] action create  
- execute the ruby block restart postgres-exporter svlogd configuration  
* ruby_block[reload postgres-exporter svlogd configuration] action create  
- execute the ruby block reload postgres-exporter svlogd configuration  
  
Running handlers:  
Running handlers complete  
Chef Client finished, 434/644 resources updated in 02 minutes 16 seconds  
gitlab Reconfigured!  
[root@localhost home]#
```

比如我在另外的机器就有报错，这里看不到报错信息，去看日志文件会有提示的

```
executables=/opt/gitlab/embedded/bin/chef-client  
  
Recipe: gitlab::gitlab-rails  
* execute[clear the gitlab-rails cache] action run  
- execute /opt/gitlab/bin/gitlab-rake cache:clear  
  
Running handlers:  
There was an error running gitlab-ctl reconfigure:  
  
bash[migrate gitlab-rails database] (gitlab::database_migrations line 50) had an error: Mixlib::ShellOut::ShellCommandFailed: Expected process to exit with [0], but received '1'  
---- Begin output of "bash" /tmp/chef-script20190129-21778-lmbf6zf" ----  
STDOUT: rake aborted!  
PG::ConnectionBad: could not connect to server: No such file or directory  
Is the server running locally and accepting  
connections on Unix domain socket "/var/opt/gitlab/postgresql/.s.PGSQL.5432"?  
/opt/gitlab/embedded/service/gitlab-rails/lib/tasks/gitlab/db.rake:51:in 'block (3 levels) in <top (required)>'  
/opt/gitlab/embedded/bin/bundle:23:in 'load'  
/opt/gitlab/embedded/bin/bundle:23:in '<main>'  
Tasks: TOP => gitlab:db:configure  
(See full trace by running task with --trace)  
STDERR:  
---- End output of "bash" /tmp/chef-script20190129-21778-lmbf6zf" ----  
Ran "bash" /tmp/chef-script20190129-21778-lmbf6zf" returned 1  
  
Running handlers complete  
Chef Client failed, 9 resources updated in 45 seconds
```

# 页面访问初始化 root 用户密码

访问 <http://192.168.0.21:9999>，就是那个 external\_url，能访问则表示成功，然后修改默认 root 用户(管理员)的密码，不需要旧的密码（也没有），直接修改即可。

Please create a password for your new account.

## GitLab Community Edition

### Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

#### Change your password

New password

.....

Confirm new password

.....

Change your password

[Didn't receive a confirmation email? Request a new one](#)  
[Already have login and password? Sign in](#)

通过浏览器访问页面，设置初始密码，默认管理用户是root

如果不能访问，可能是端口没有开放（gitlab 在我的 CentOS 上，浏览器访问在 win10 上，故要开放端口）  
确定是否是端口问题，可以在 gitlab 所在的主机，使用命令：

// 如果能返回设置初始化密码的 HTML 页面信息，则表示是端口问题。

```
[root@localhost ~]# curl http://192.168.0.21:9999
```

CentOS7（注意 CentOS7 默认使用 firewall 作为防火墙，6 的可能不适用）对外开放端口：

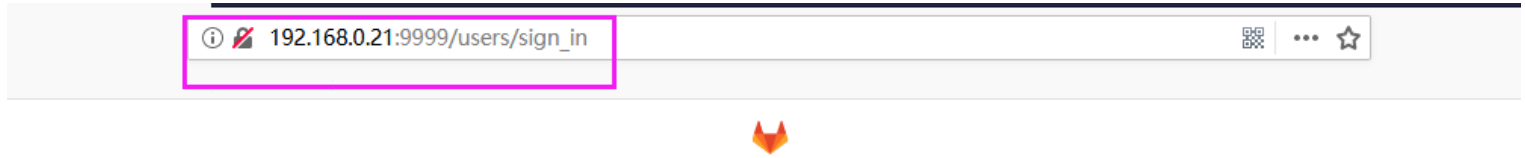
// --permanent 永久生效，没有此参数重启后失效

```
[root@localhost ~]# firewall-cmd --zone=public --add-port=9999/tcp --permanent
```

// 重新载入配置

```
[root@localhost ~]# firewall-cmd --reload
```

修改成功就会自动跳转到



# GitLab Community Edition

## Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in

Register

Username or email

root

管理员默认用户

Password

.....

就是之前设置修改的密码

☐ Remember me

[Forgot your password?](#)

Sign in

登录成功页面



## Welcome to GitLab

Code, test, and deploy together

**Create a project**  
Projects are where you store your code, access issues, wiki and other features of GitLab.

**Create a group**  
Groups are a great way to organize projects and people.

**Add people**  
Add your team members and others to GitLab.

**Configure GitLab**  
Make adjustments to how your GitLab instance is set up.

## 温馨提示:

如果安装错误了，需要重新安装，则

```
# rpm -e gitlab-ce 卸载
```

```
# cd / 去到根目录
```

```
# find -name gitlab* 把 gitlab 所有的都找出来，然后
```

```
# rm -rf .... 全部删除，就可以重新安装了，不然删除干净就重新安装可能会有问题。
```

## 管理 GitLab 服务的常用命令：

```
// 如果修改了external_url，重启一下就好了
[root@localhost home]# gitlab-ctl restart

// 如果不行，可能要用
[root@localhost home]# gitlab-ctl reconfigure

启动GitLab服务：
[root@localhost home]# gitlab-ctl start

查看GitLab的状态：
[root@localhost home]# gitlab-ctl status

停止GitLab服务：
[root@localhost home]# gitlab-ctl stop
```

## 穿插清理缓存的知识点：

```
// 查看运行内存
[root@localhost home]# free -m

// free就是剩余多少；cached就是缓存，centos会缓存大文件什么的

      total        used        free      shared    buffers     cached
Mem:    3961        3749         212           0         225         560
-/+ buffers/cache:    2963         998
Swap:          0           0           0

// 想清除缓存的操作

[root@localhost home]# sync
[root@localhost home]# echo 3 > /proc/sys/vm/drop_caches

// 然后再看一下，就cached就很少了，可用运行内存就多了

[root@localhost home]# free -m

      total        used        free      shared    buffers     cached
Mem:    3961        2919        1042           0           2         18
-/+ buffers/cache:    2898        1062
Swap:          0           0           0
[root@localhost home]#
```

## GitLab 感觉挺耗内存的

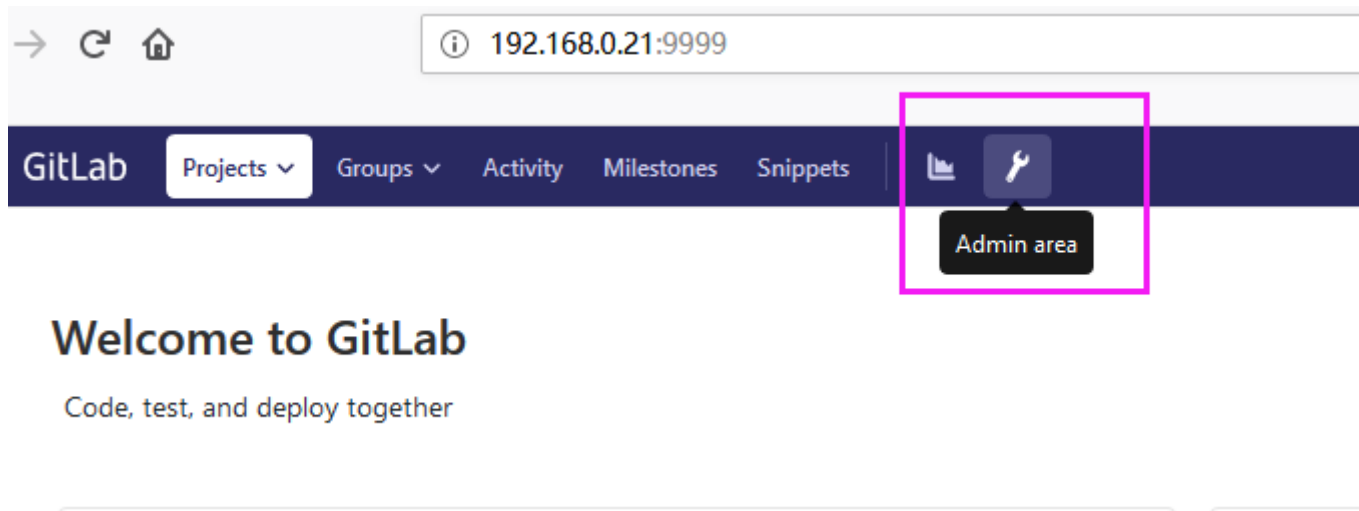
```
// 安装、运行GitLab之前的内存
[root@localhost home]# free -m

      total        used        free      shared  buff/cache   available
Mem:    15828         255       13120           8        2452       15088
```

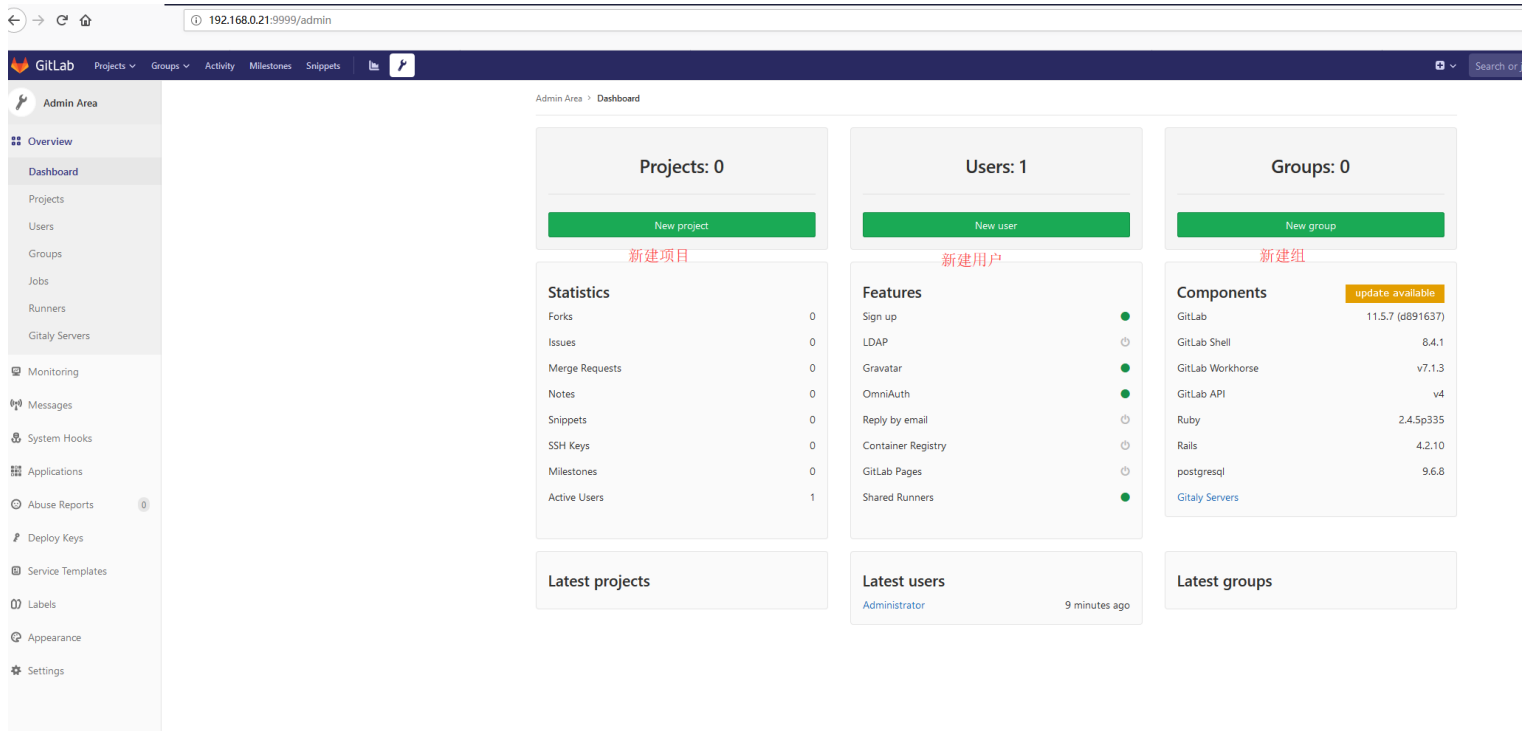
```
Swap:          12031          0        12031
// 安装、运行GitLab之后的内存
[root@localhost home]# free -m
              total          used         free       shared    buff/cache   available
Mem:          15828          5946          7082          130         2798         9267
Swap:          12031           0         12031
[root@localhost home]#
```

## 4、GitLab 的使用

### 管理区域



新建项目、新建用户、新建组 都在这个页面



# 新建项目

### New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), [among other things](#).

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

**Tip:** You can also create a project from the command line. [Show command](#)

Blank project空白项目

Create from template

Import project

Project name

MyFirst名称

Project URL

http://192.168.0.21:9999/

root这项目挂载到哪个用户上

Project slug

myfirst路径

Want to house several dependent projects under the same namespace? [Create a group](#).

Project description (optional)

Description format描述

Visibility Level

☒ Private权限，私人表示自己、授权给对应的用户，能访问  
Project access must be granted explicitly to each user.

☐ Internal只要是登录的用户都可以访问  
The project can be accessed by any logged in user.

☐ PublicThe project can be accessed without any authentication.不登录也可以访问

☐ Initialize repository with a README

Allows you to immediately done this project's repository. Skip this if you plan to push up an existing repository.

Create project

Cancel

GitLab

ProjectsGroupsActivityMilestonesSnippets

MyFirst

You won't be able to pull or push project code via SSH until you **add an SSH key** to your profile. 这需要使用公钥密钥等操作，反正我不需要ssh登录

The Auto DevOps pipeline has been enabled and will be used if no alternative CI configuration file is found. [More information](#)

Project

DetailsActivityCycle Analytics

Issues0

Merge Requests0

CI / CD

Operations

Wiki

Snippets

Settings

AdministratorMyFirstDetails

Project 'MyFirst' was successfully created.

MyFirstPrivateAdd license

Project ID: 1

0StarHTTPhttp://192.168.0.21:9999/复制该链接就可以用来代码管理了，后面说

+Global

The repository for this project is empty

If you already have files you can push them using the [command line instructions](#) below.

Note that the master branch is automatically protected. [Learn more about protected branches](#)

You can automatically build and test your application if you [enable Auto DevOps](#) for this project. You can automatically deploy it as well, if you [add a Kubernetes cluster](#).

Otherwise it is recommended you start with one of the options below.

Files (0 Bytes)Commits (0)Branches (0)Tags (0)Auto DevOps enabled

New fileAdd ReadmeAdd Kubernetes cluster

Command line instructions

Git init

看下这个链接，myfirst 就是项目路径或者说是项目仓库名称  
http://192.168.0.21:9999/root/myfirst.git



# 新建用户

新用户的密码设置有两种，下图是一种

GitLab

Projects Groups Activity Milestones Snippets

Admin Area

Overview

Dashboard

Projects

Users

Groups

Jobs

Runners

GitLab Servers

Monitoring

Messages

System Hooks

Applications

Abuse Reports

Deploy Keys

Service Templates

Labels

Appearance

Settings

Admin Area > New User

新建用户

New user

Account

Name

测试用户 可以说是用户描述

\* required

Username

TestUser 用来登录的用户名

\* required

Email

123456@163.com gitlab会发邮箱过去，哪怕是本地私有仓库也会发，收到这个邮件，点开链接，就会有设置密码的界面，与root用户的密码设置一致。当然也有其他方法，后面讲。

\* required

Password

Password

Reset link will be generated and sent to the user. User will be forced to set the password on first sign in.

Access

Projects limit

100000 项目限制

Can create group

☒ 他能否创建组

Access level

☒ Regular 普通用户

Regular users have access to their groups and projects

☐ Admin 管理员

Administrators have access to all groups, projects and users and can manage all features in this installation

External

☐ External users cannot see internal or private projects unless access is explicitly granted. Also, external users cannot create projects or groups.

Profile

Avatar

浏览... 未选择文件。

Skype

Linkedin

Twitter

Website

Create user

Cancel

创建完成，点击 Edit

User was successfully created.

测试用户

Impersonate

Edit

Account Groups and projects SSH keys Identities Impersonation Tokens

测试用户



Profile page: **TestUser**

Profile

Member since **Jan 30, 2019 3:46am**

Account:

Name: **测试用户**

Username: **TestUser**

Email: **123456@163.com**

ID: **2**

Two-factor Authentication: **Disabled**

External User: **No**

Can create groups: **Yes**

Personal projects limit: **100000**

Member since: **Jan 30, 2019 3:46am**

Confirmed at: **Jan 30, 2019 3:46am**

Current sign-in IP: **never**

Block this user

Blocking user has the following effects:

- User will not be able to login
- User will not be able to access git repositories
- Personal projects will be left
- Owned groups will be left

Block user

Delete user

Deleting a user has the following effects:

- Certain user content will be moved to a system-wide "Ghost User" in order to maintain content for posterity. For further information, please refer to the [user account deletion documentation](#).

Delete user

Delete user and contributions

This option deletes the user and any contributions that would usually be moved to the [system ghost user](#). As well as the user's personal projects, groups owned solely by the user, and projects in them, will also be removed. Commits to other projects are unaffected.

Delete user and contributions

第二种方法设置密码，上面点击完 Edit 之后，会有下图

Password

这里就可以设置密码了，不需要经过邮箱也是可以的，注意密码要8位以上

## Profile

Save changes

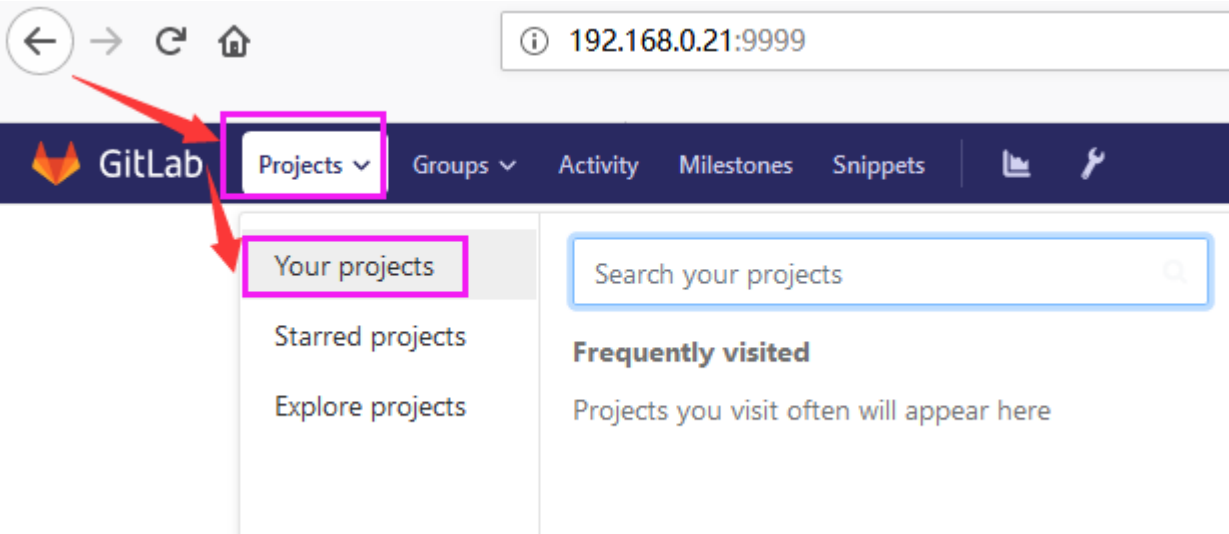
Cancel

← → ↺ 🏠 ⓘ 192.168.0.21:9999/admin

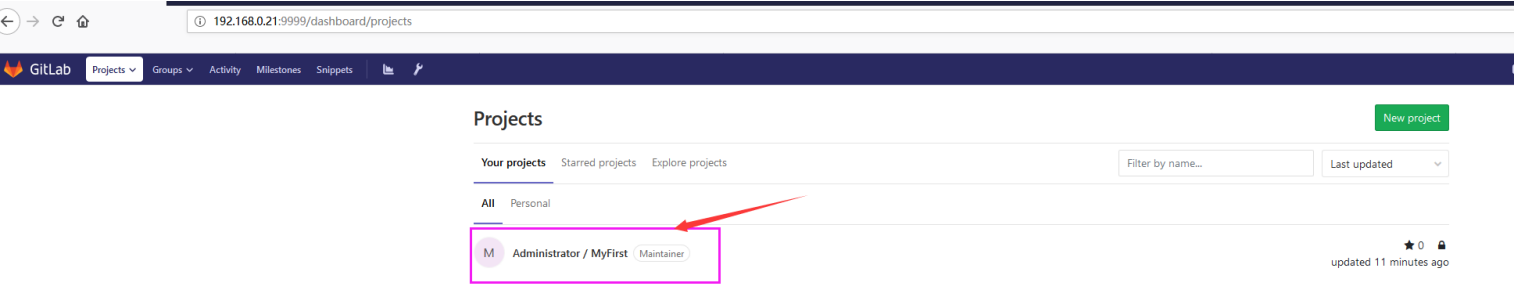
The screenshot displays the GitLab Admin Area Dashboard. The left sidebar contains navigation links for Overview, Dashboard, Projects, Users, Groups, Jobs, Runners, and GitLab Servers. The main content area is divided into several sections:

- Projects: 1**: A green button labeled "New project".
- Users: 2**: A green button labeled "New user".
- Groups: 0**: A green button labeled "New group".
- Statistics**: A table showing counts for Forks, Issues, Merge Requests, Notes, Snippets, SSH Keys, Milestones, and Active Users.
- Features**: A list of features with status indicators (green dot for enabled, grey dot for disabled).
- Components**: A table showing the versions of various components, including GitLab, GitLab Shell, GitLab Workhorse, GitLab API, Ruby, Rails, postgresql, and GitLab Servers.
- Latest projects**: A table showing the latest project, "Administrator / MyFirst", created 9 minutes ago.
- Latest users**: A table showing the latest users, "Administrator" (created 24 minutes ago) and "Administrator" (created 49 seconds ago).
- Latest groups**: A table showing the latest groups.

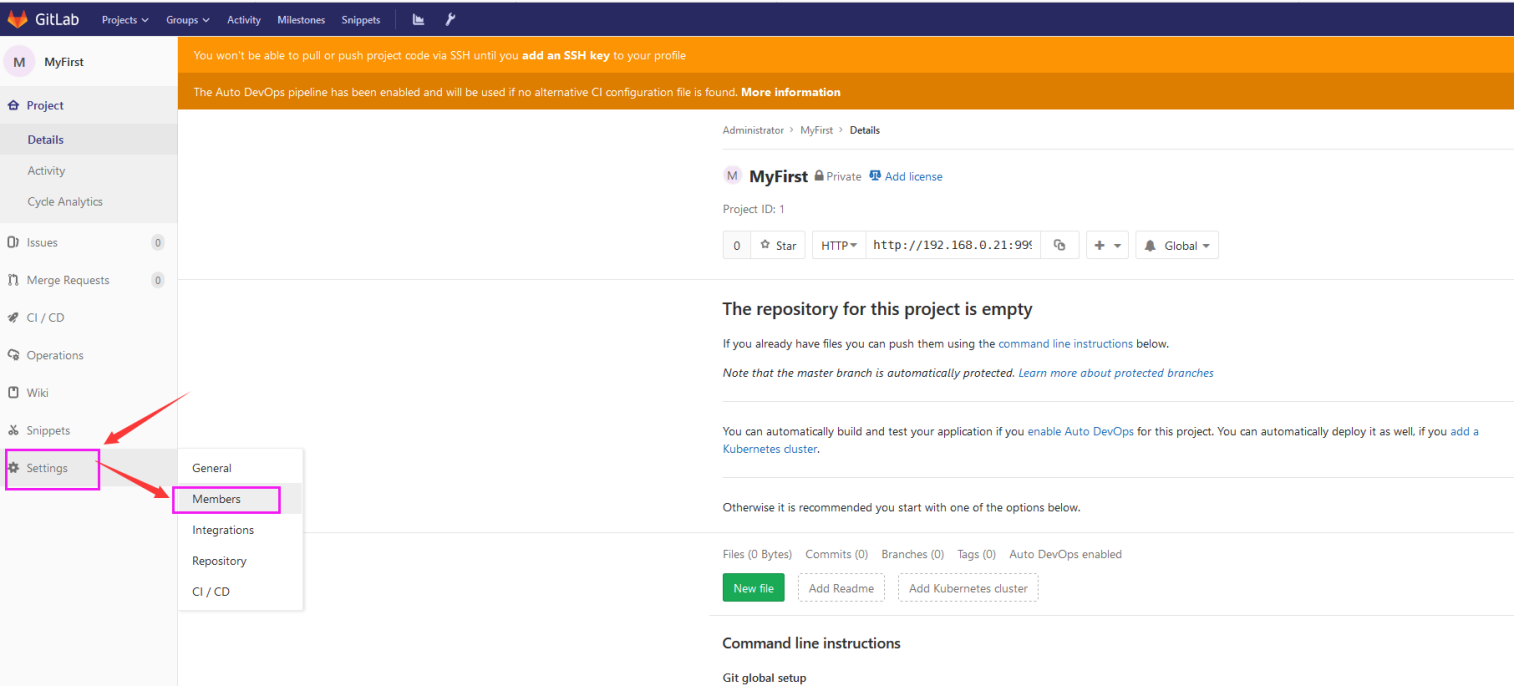
# 项目添加用户



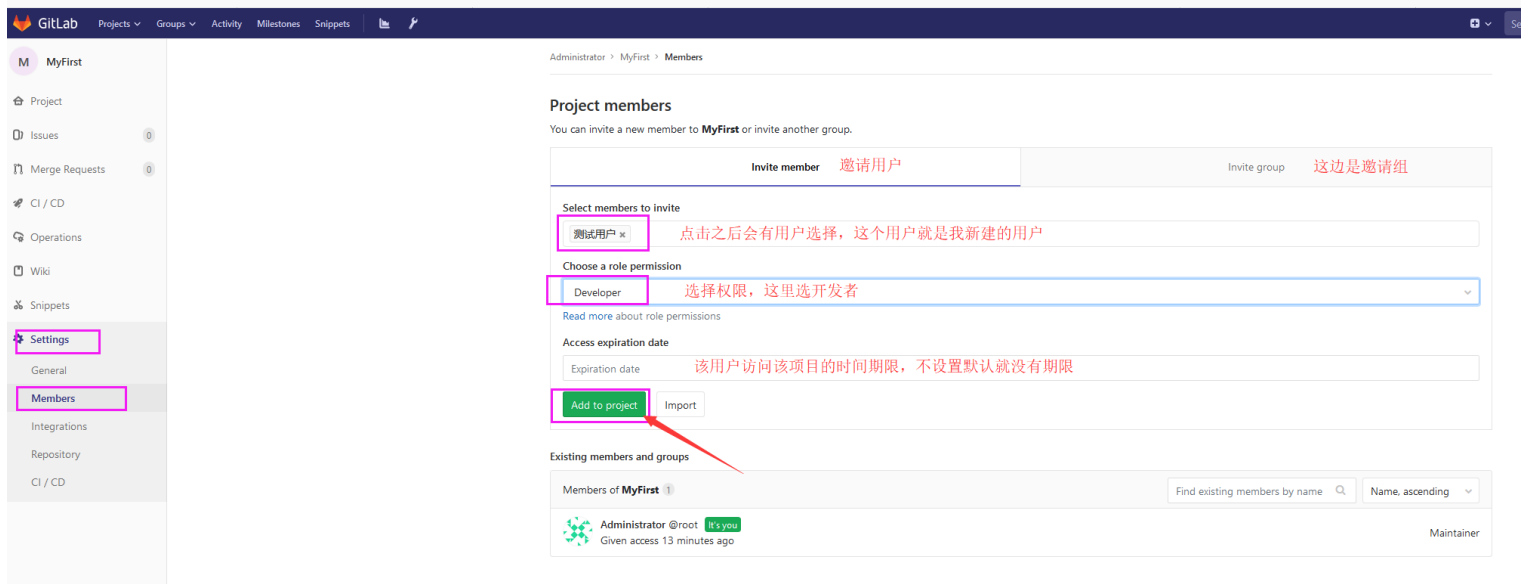
点击



点击设置，然后成员



就可以邀请用户了，还可以邀请组，组我们还没有创建（需要注意的是 Developer 权限不能 pushmaster 主分支，需要更高的权限）



然后就能看到有新的成员的了，这样该项目 TestUser 用户就能管理了

### Project members

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

Invite member

Invite group

Select members to invite

Search for members to update or invite

Choose a role permission

Guest

Read more about role permissions

Access expiration date

Expiration date

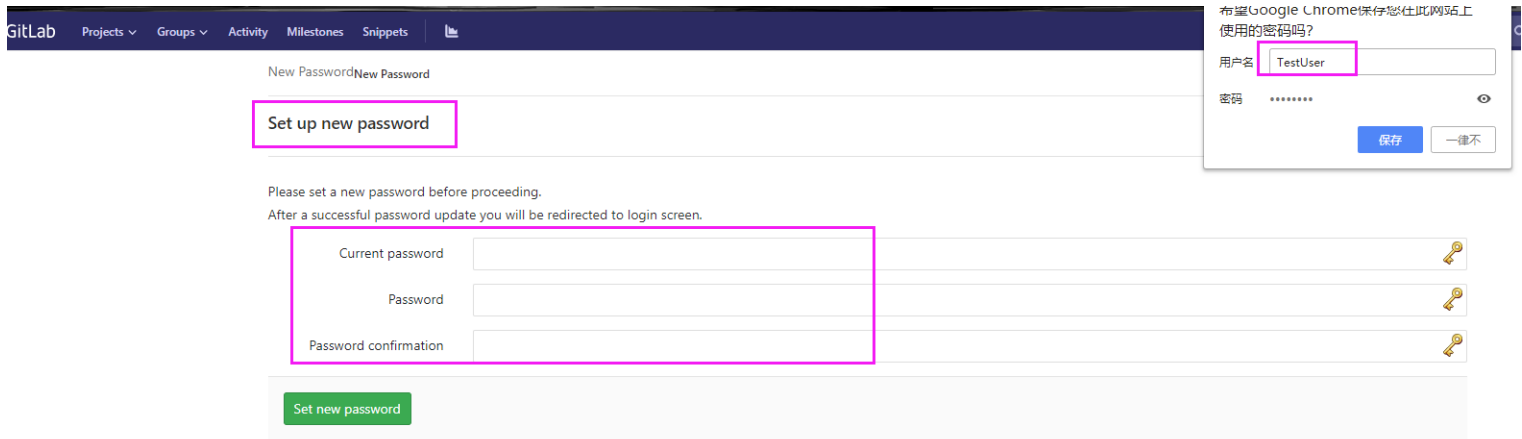
Add to project

Import

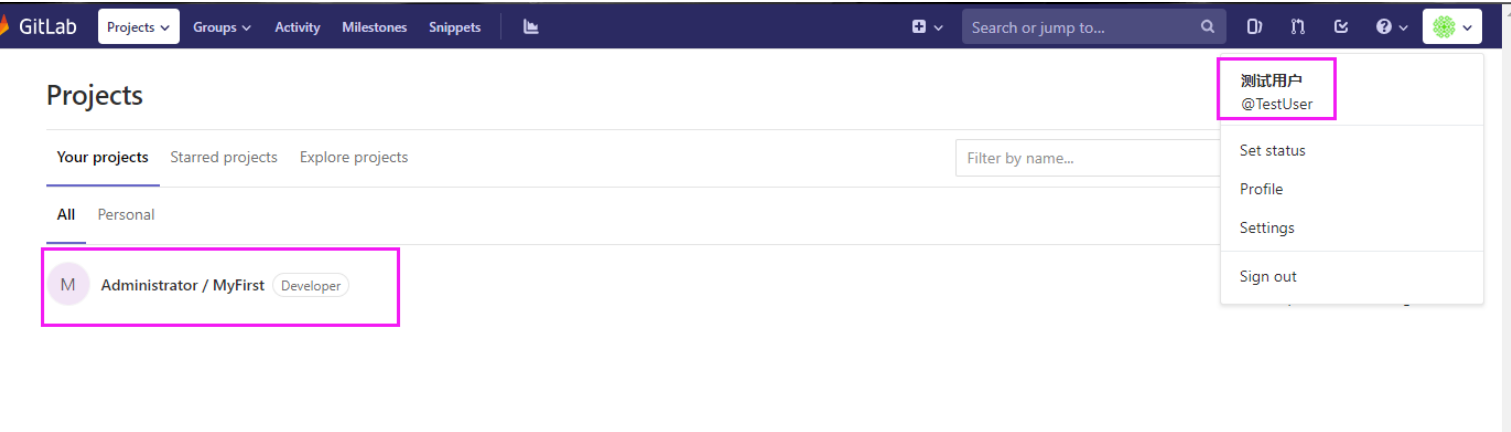
#### Existing members and groups

Members of <b>MyFirst</b> 2		Find existing members by name	Name, ascending
	Administrator @root <span>It's you</span> Given access 15 minutes ago		Maintainer
	测试用户 @TestUser Given access in 1 minute	Developer	Expiration date

登录 TestUser 测试用户（换一个浏览器登录，root 用户就不用退出了，后面 root 用户还会用到），会强制要求修改密码

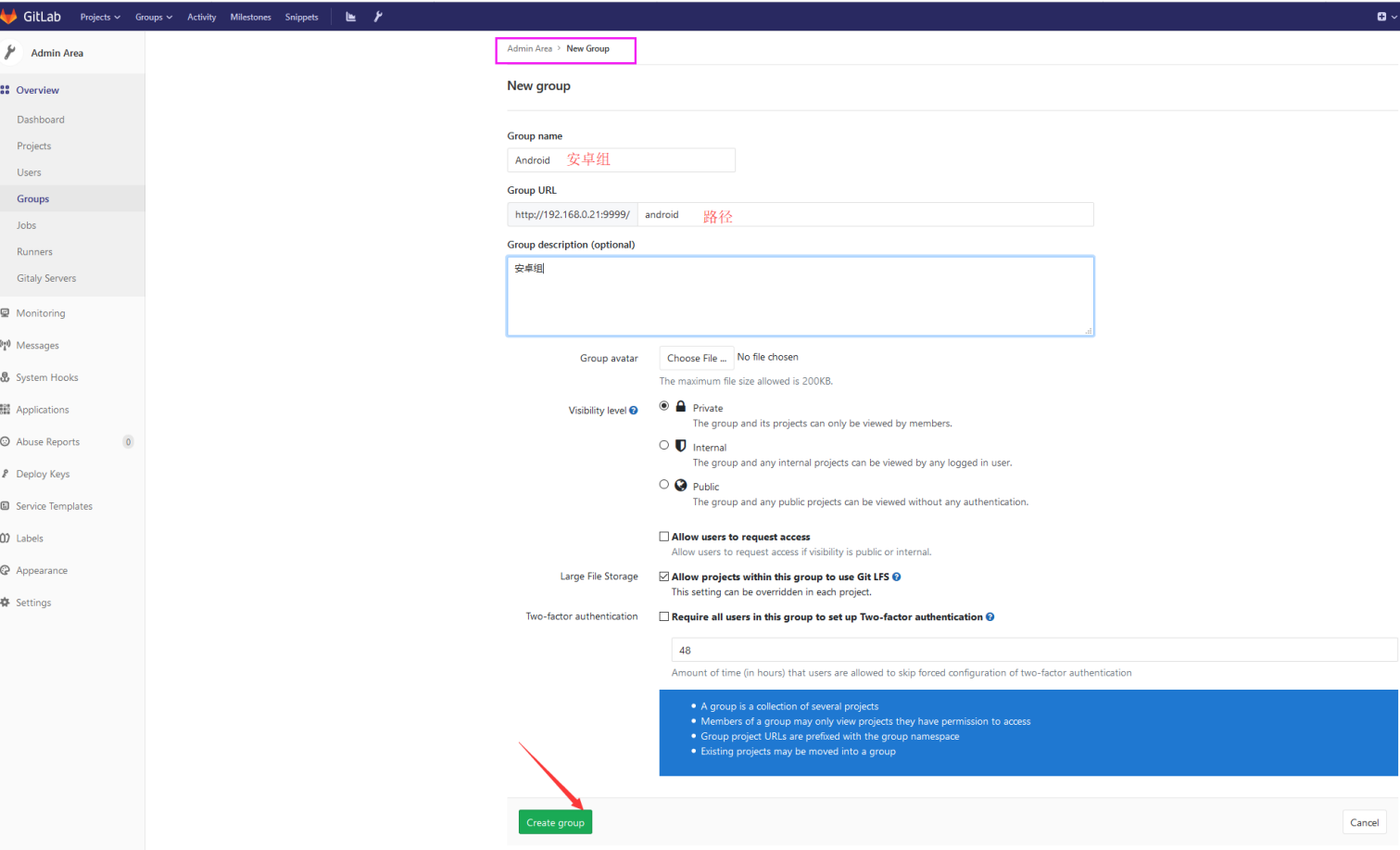


测试用户就有了新添加的项目



## 创建组

回到 root 用户，看图



创建组完成之后，该界面就能添加用户进 组 了

Group 'Android' was successfully created.

Group: Android

Edit

Group info:

A

Name: **Android**

Path: **android**

Description: 安卓组

Visibility level: **Private**

Created on: Jan 30, 2019 3:56am

Storage: 0 Bytes (0 Bytes repositories, 0 Bytes build artifacts, 0 Bytes LFS)

Group Git LFS status: Enabled for all projects

Projects 0

Add user(s) to the group: 给组添加用户

Read more about project permissions [here](#)

测试用户 x

Developer

Add users to group

Android group members 1

Manage access

Administrator @root It's you

Given access 1 hour ago

Owner

添加好了

Group: Android

Edit

Group info:

A

Name: **Android**

Path: **android**

Description: 安卓组

Visibility level: **Private**

Created on: Jan 30, 2019 3:56am

Storage: 0 Bytes (0 Bytes repositories, 0 Bytes build artifacts, 0 Bytes LFS)

Group Git LFS status: Enabled for all projects

Projects 0

Add user(s) to the group:

Read more about project permissions [here](#)

Search for a user

Guest

Add users to group

Android group members 2

Manage access

Administrator @root It's you

Given access 1 hour ago

Owner

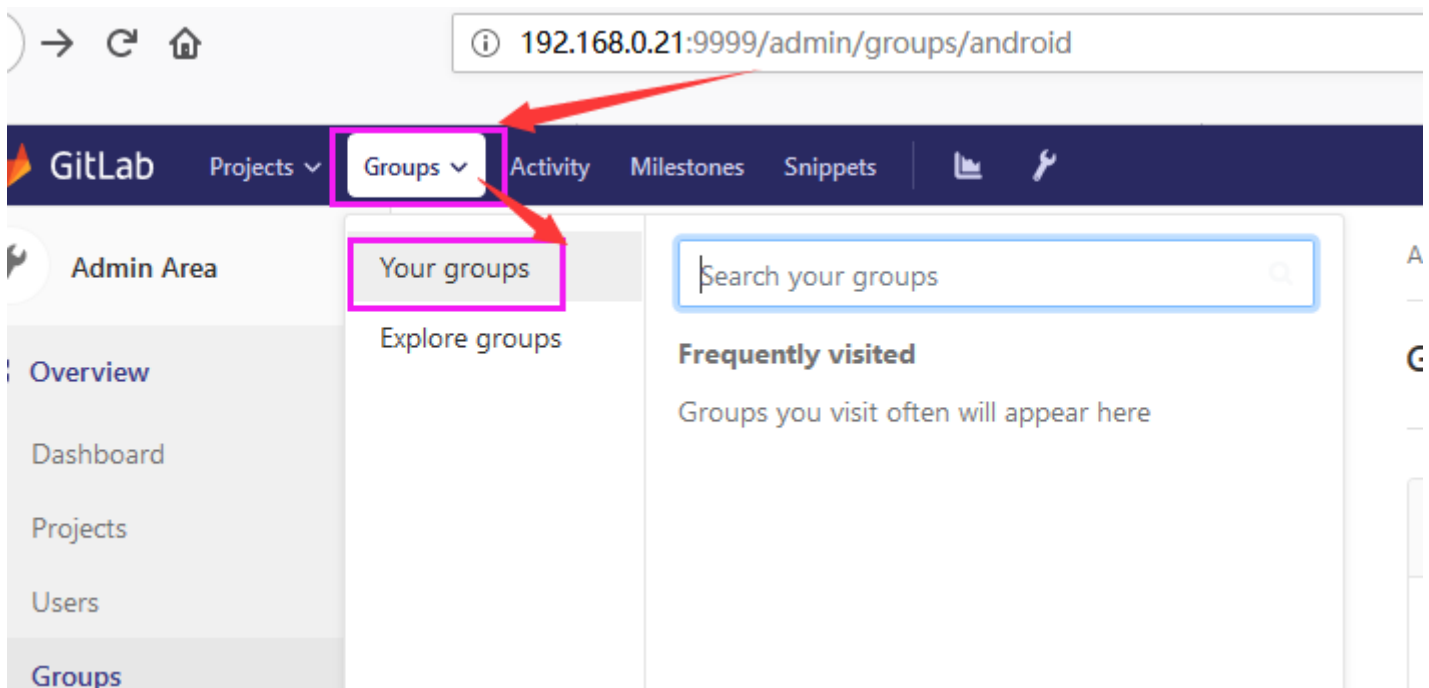
测试用户 @TestUser

Given access in 1 minute

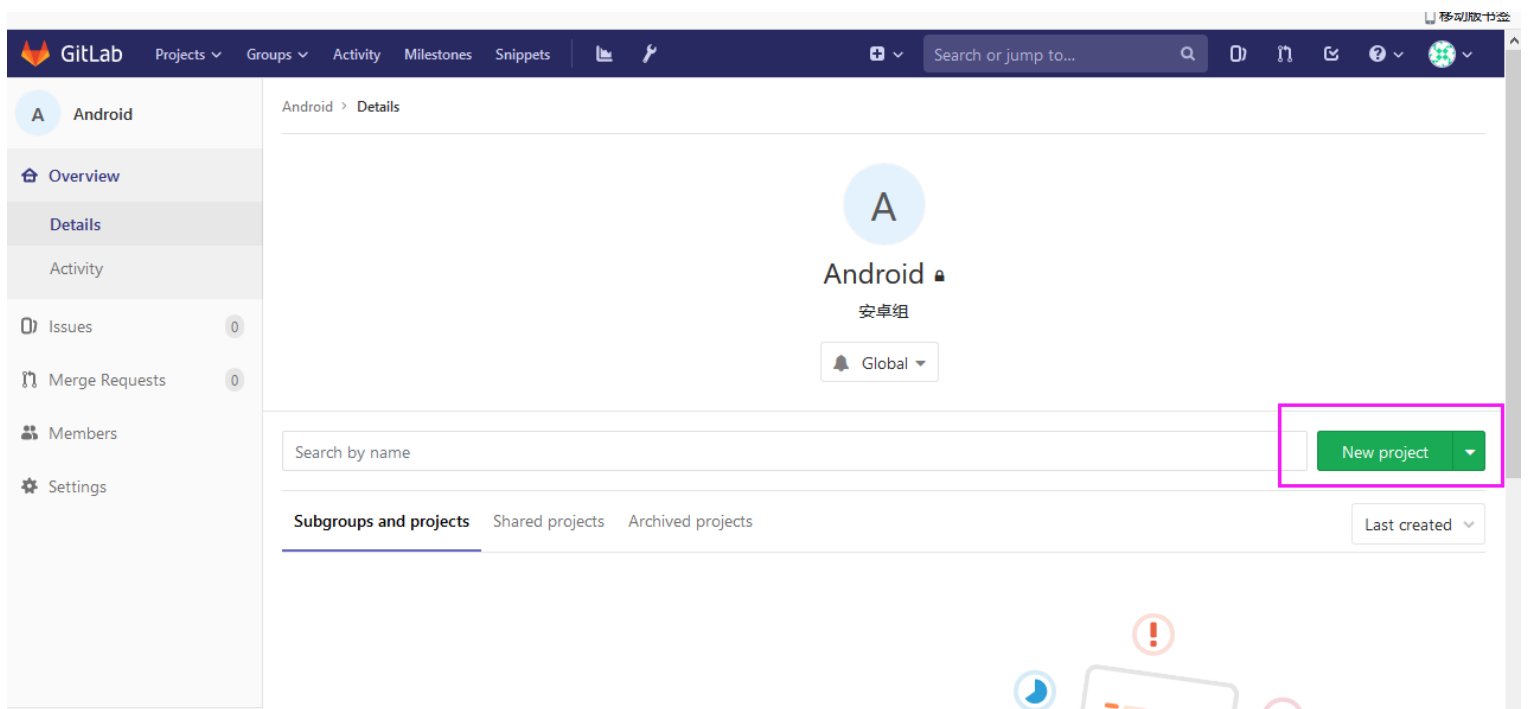
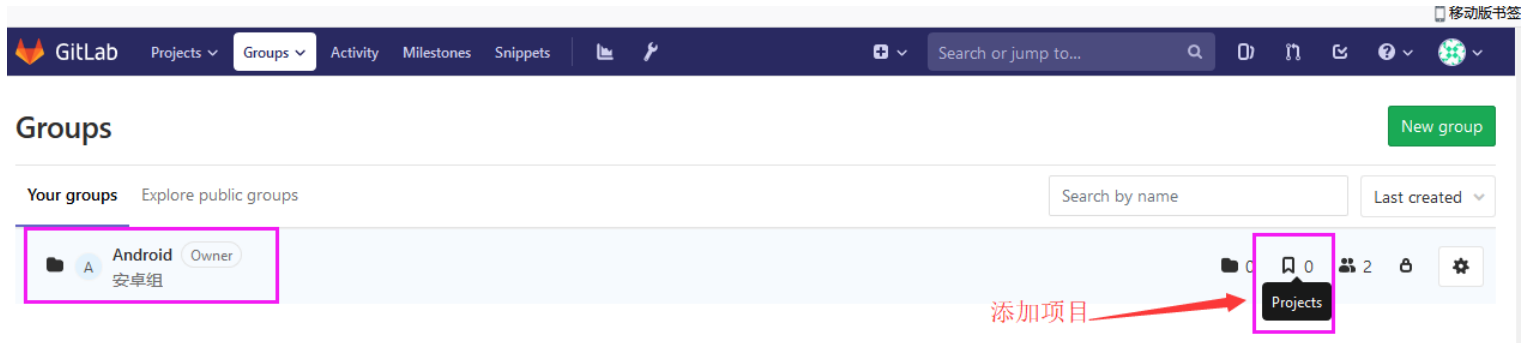
Developer

给组新建项目

给组添加项目，这样整个组的成员都可以管理该项目



给组添加项目，可以从下图小图标进去



也可以，点击 组名称，然后选择 设置 进去添加项目



# Groups

New group

Your groups

Explore public groups

Search by name

Last created

A

Android

Owner

安卓组

002

GitLab

ProjectsGroupsActivityMilestonesSnippets

Search or jump to...

AAndroid

OverviewIssuesMerge RequestsMembersSettingsGeneralProjectsCI / CD

Android > Projects

Android projects:

This group has no projects yet

New project

GitLab

ProjectsGroupsActivityMilestonesSnippets

Search or jump to...

new project

Project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), among other things.

When features are enabled for blank projects, templates, or when importing, but you can disable them afterward in the project settings.

You can also create a project from the command line. [Show command](#)

Blank project

Create from template

Import project

Project name

AndroidFirst

Project URL

http://192.168.0.21:9999/ root

Project slug

androidfirst

Want to house several dependencies? [Create a group.](#)

Project description (optional)

Description format

Groups

android

Users

root

Visibility Level

☒ Private

Project access must be granted explicitly to each user.

☐ Internal

The project can be accessed by any logged in user.

☐ Public

The project can be accessed without any authentication.

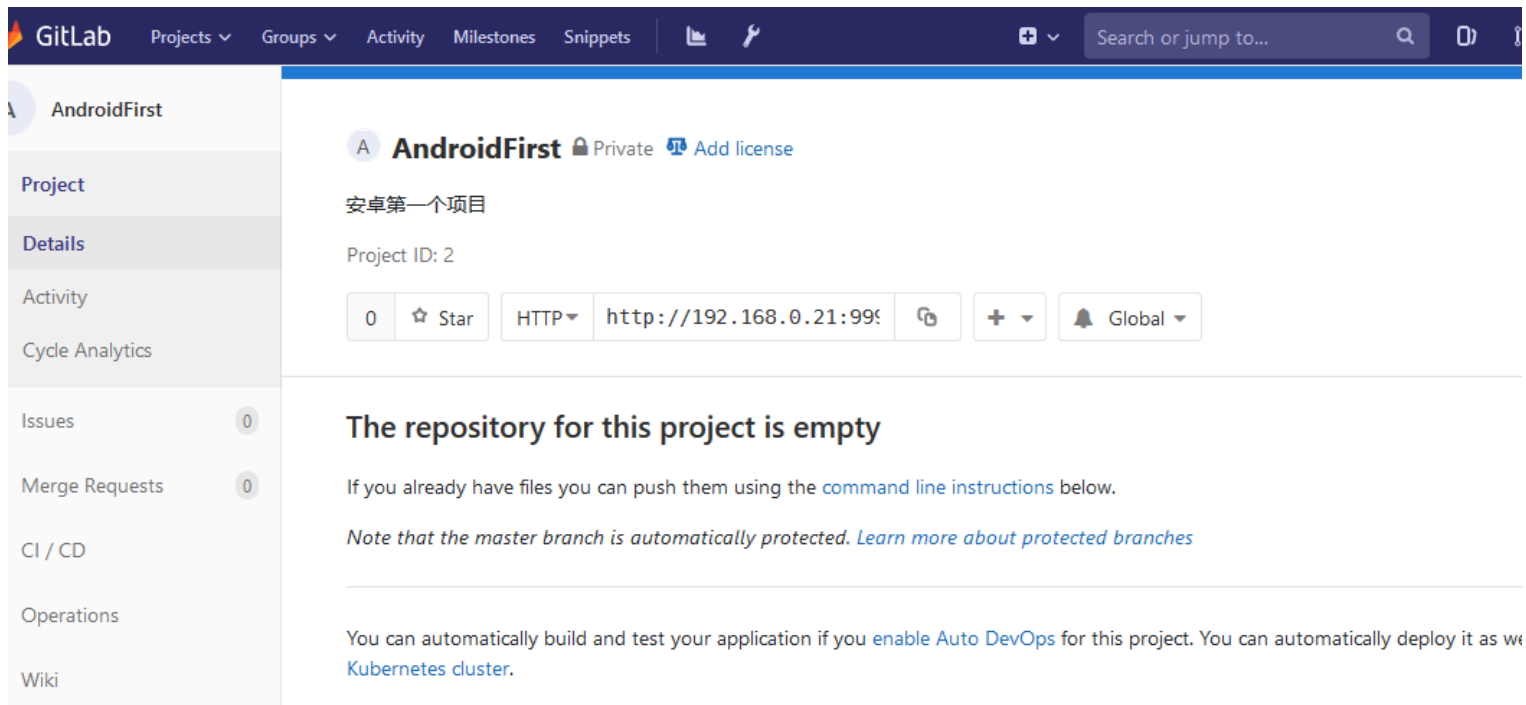
☐ Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project

Cancel

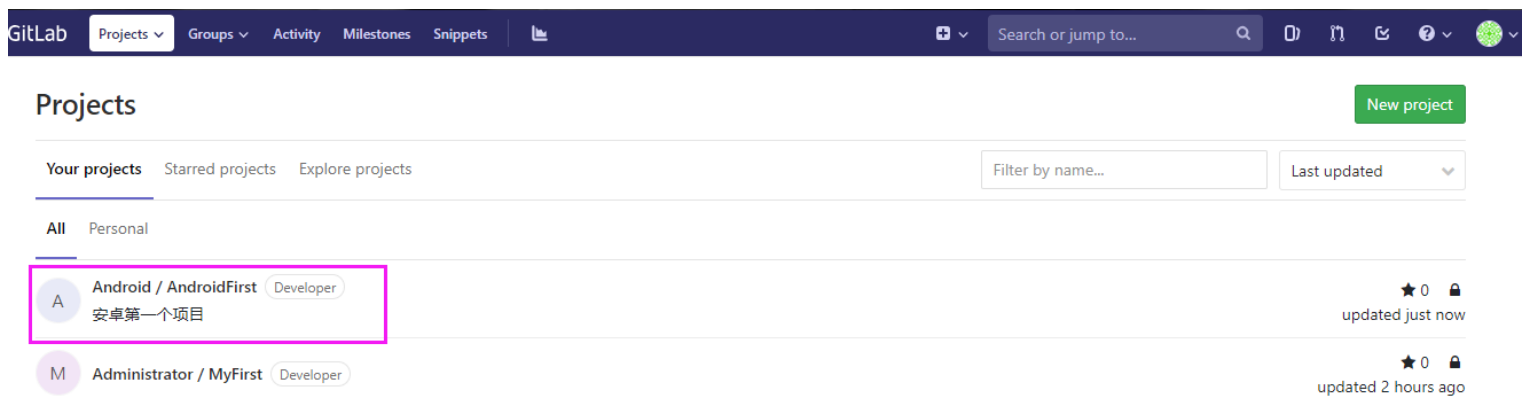
创建完成



看下仓库路径

<http://192.168.0.21:9999/android/androidfirst.git>

登录 TestUser 用户，可以看到，又多了一个项目



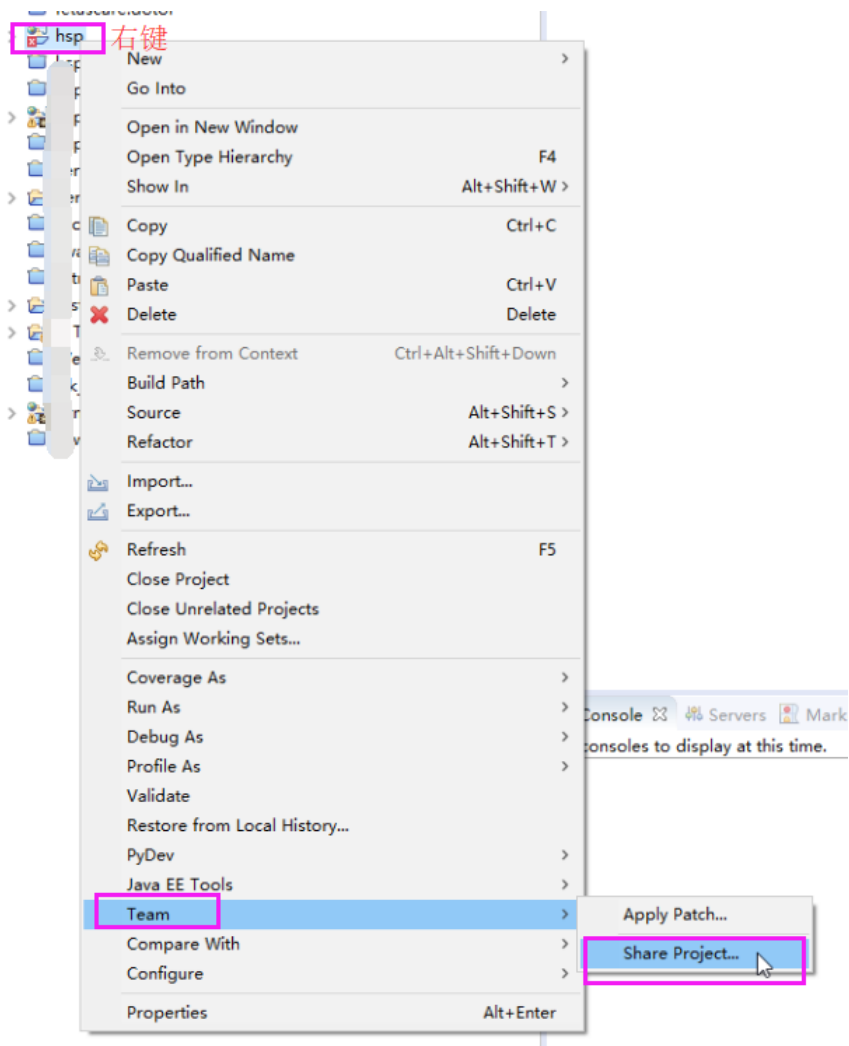
这样子项目、用户、组，就都说了，对于团队的项目，推荐采用分组，然后组添加人员，然后在组里面创建项目（注意，组里面不能添加用户下的已有项目，只能新建项目，或者导入，起码我没有找到能添加的方法）。

## 5、Eclipse 提交代码到本地仓库、私有远程 GitLab 仓库

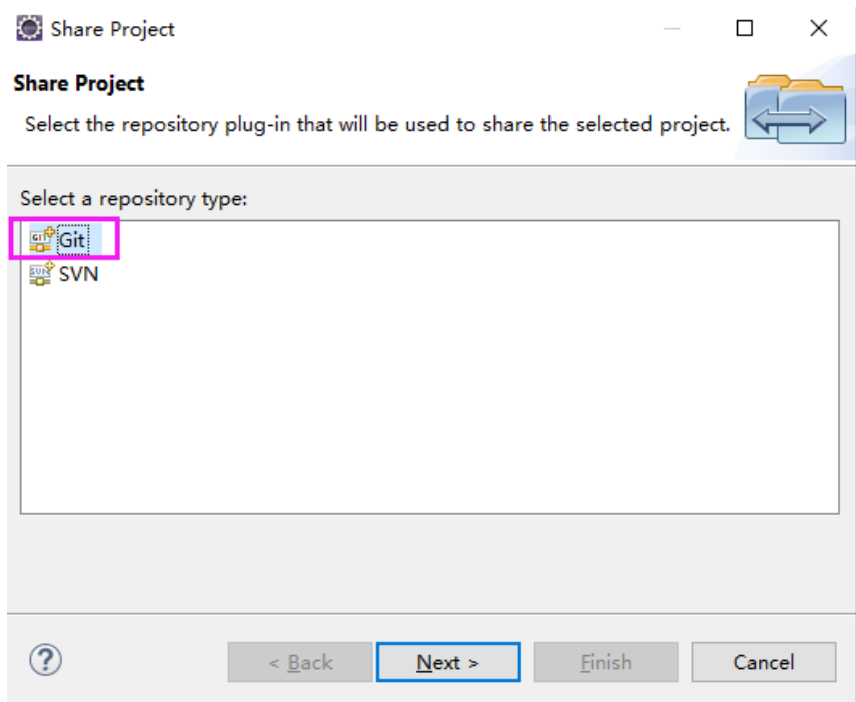
先在 GitLab 上创建项目 <http://192.168.0.21:9999/backend/hsp.git>，我这里是分组，组里面创建项目。都没有关系。

### 创建本地仓库

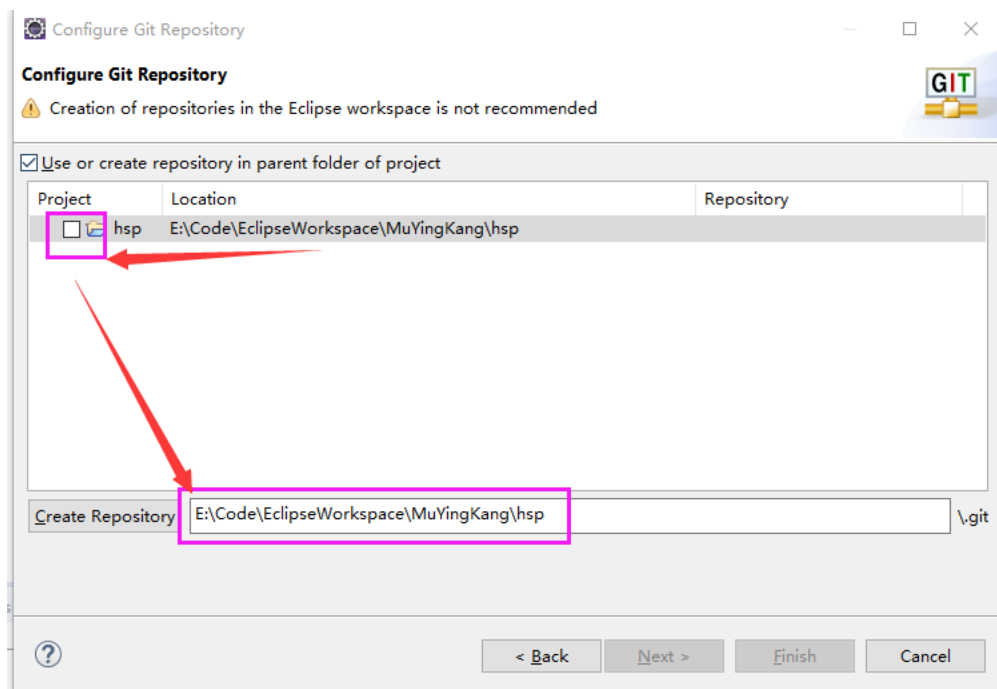
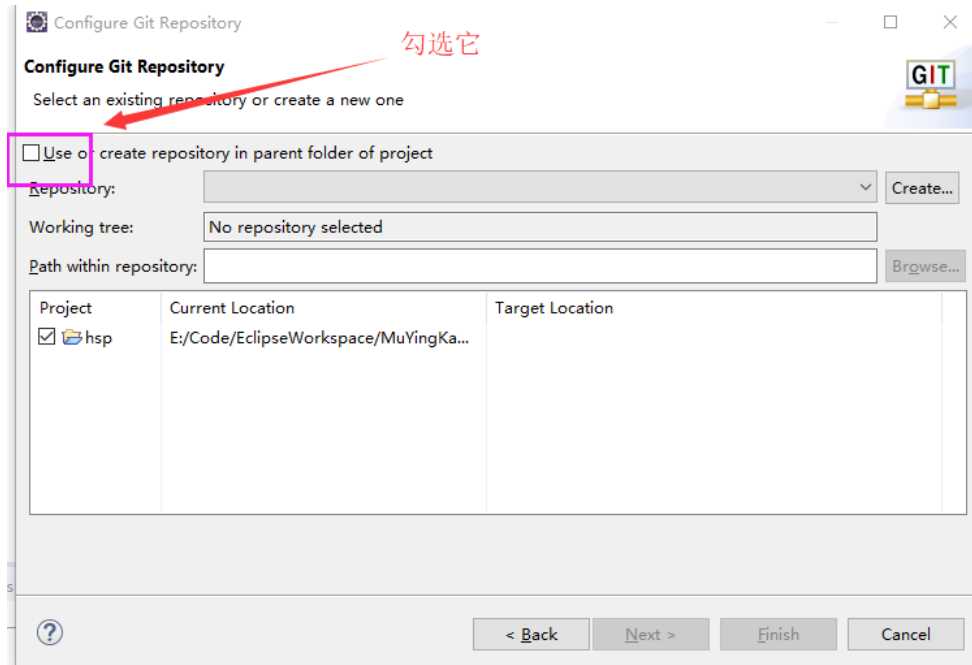
右键需要提交的项目



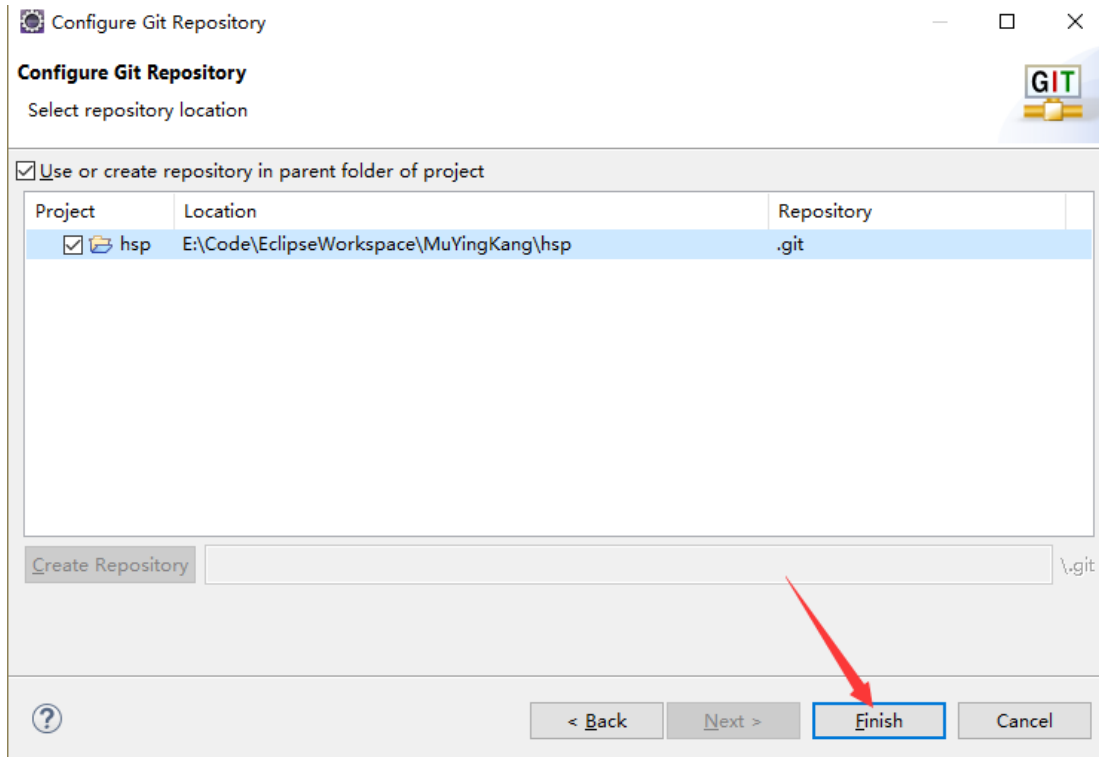
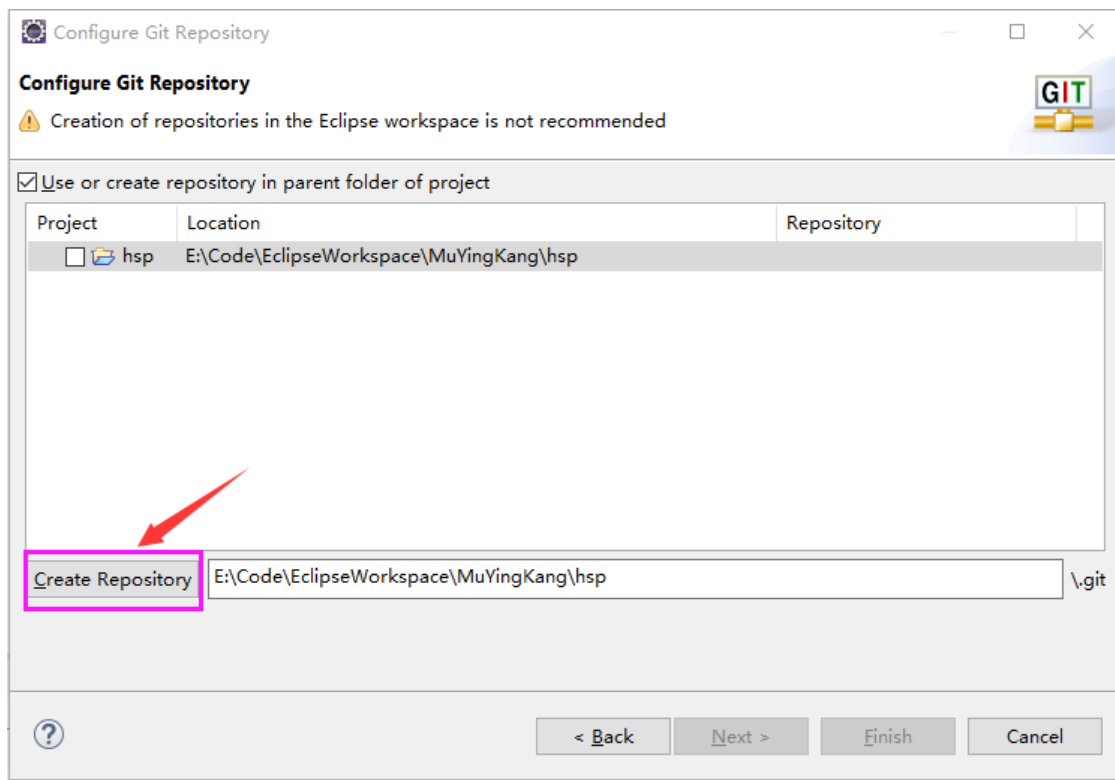
选 Git



配置本地 Git 仓库



git 提交代码一般都是先提交到 git 本地的仓库，然后再提交到远程的仓库（即类似于 GitLab、GitHub 这样的），这里应该就是创建本地的仓库，这是我用大腿猜的

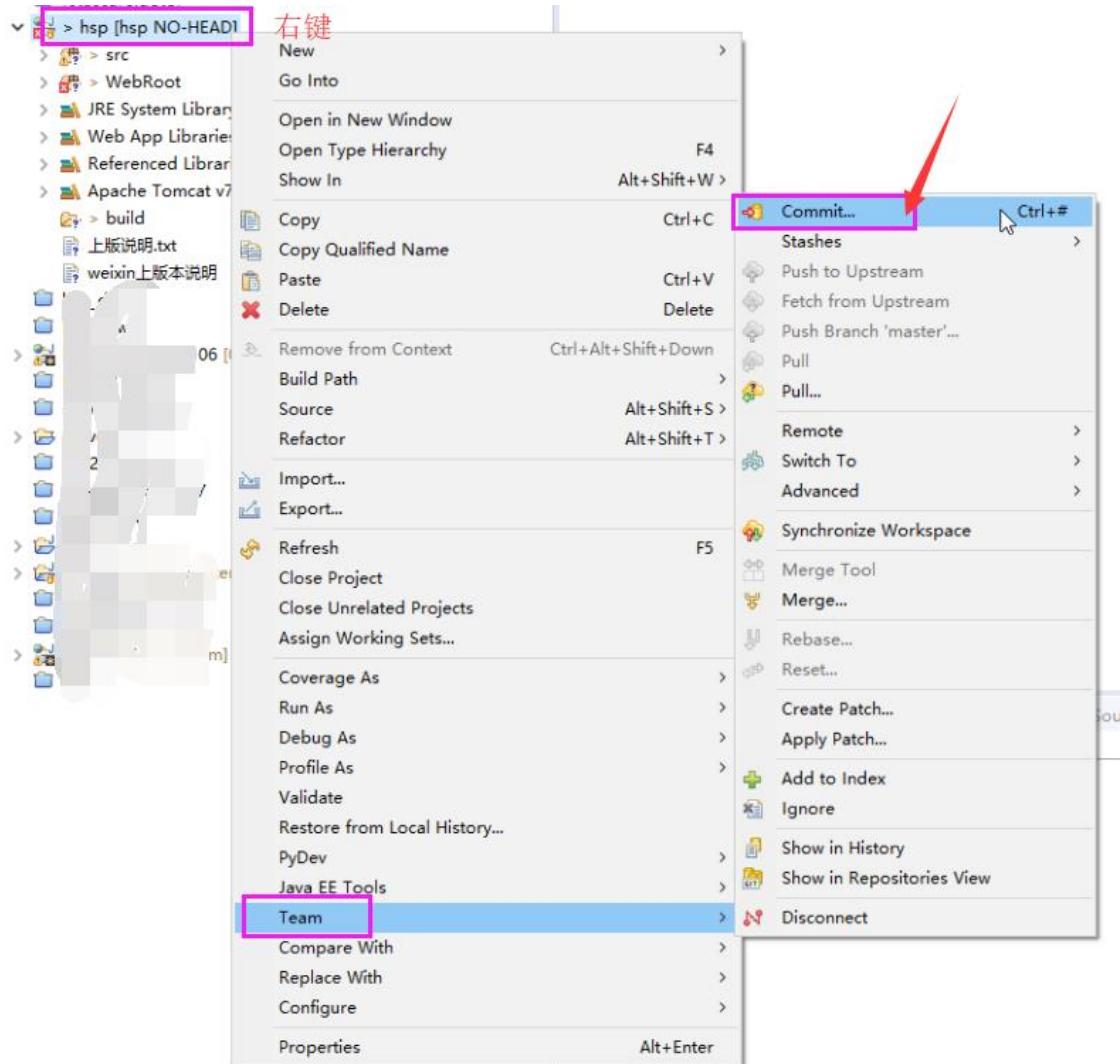


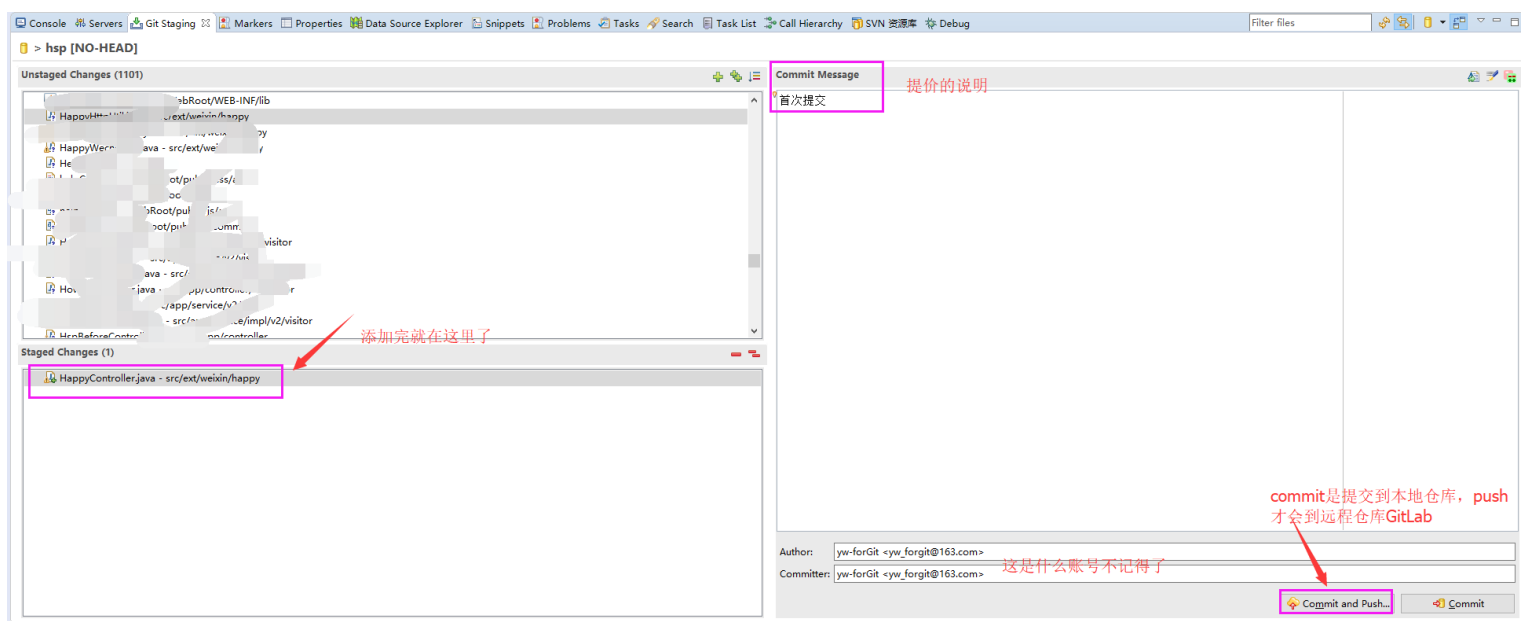
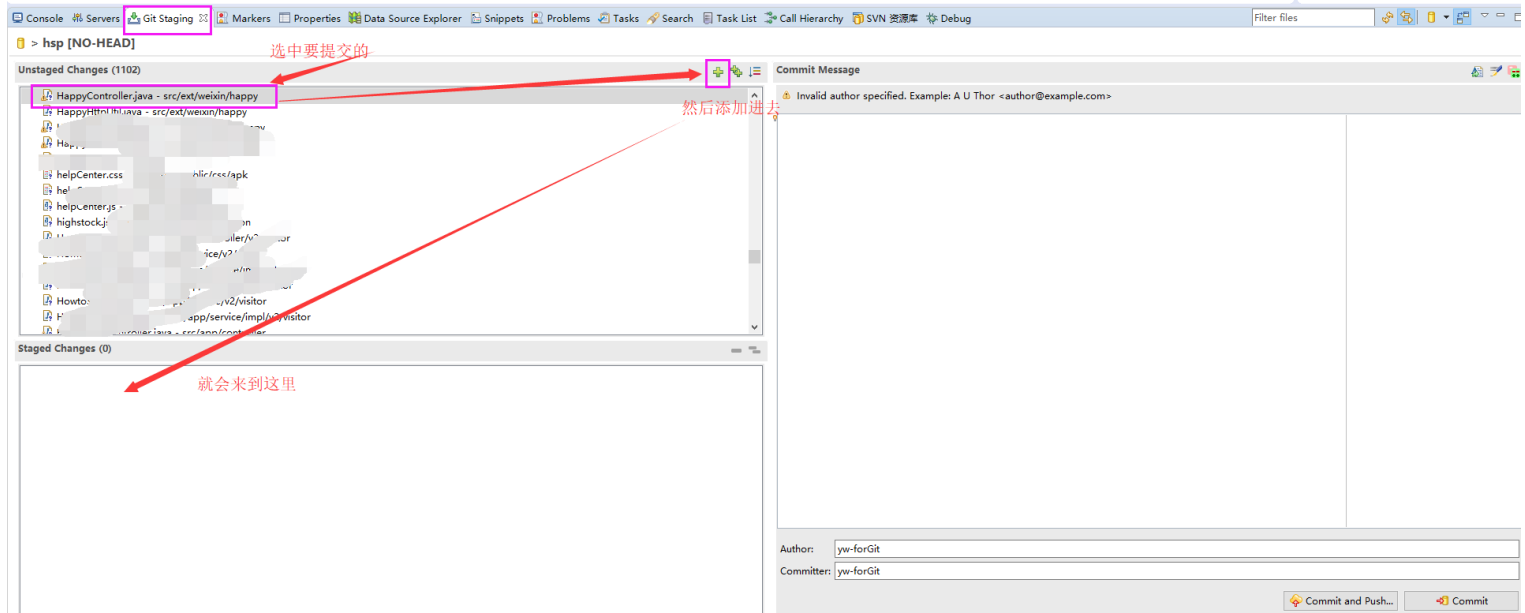
## 预 Commit

完成之后，什么弹框都没有，不要慌，看项目，多了 NO-HEAD 标志

```
retuscare.dotor
> hsp [hsp NO-HEAD]
> src
> WebRoot
> JRE System Library [jre1.8.0_161]
> Web App Libraries
> Referenced Libraries
> Apache Tomcat v7.0 [Apache Tomcat v7.0]
> build
```

右键项目，最后点击 Commit 提交，也不要慌，还不会提交的

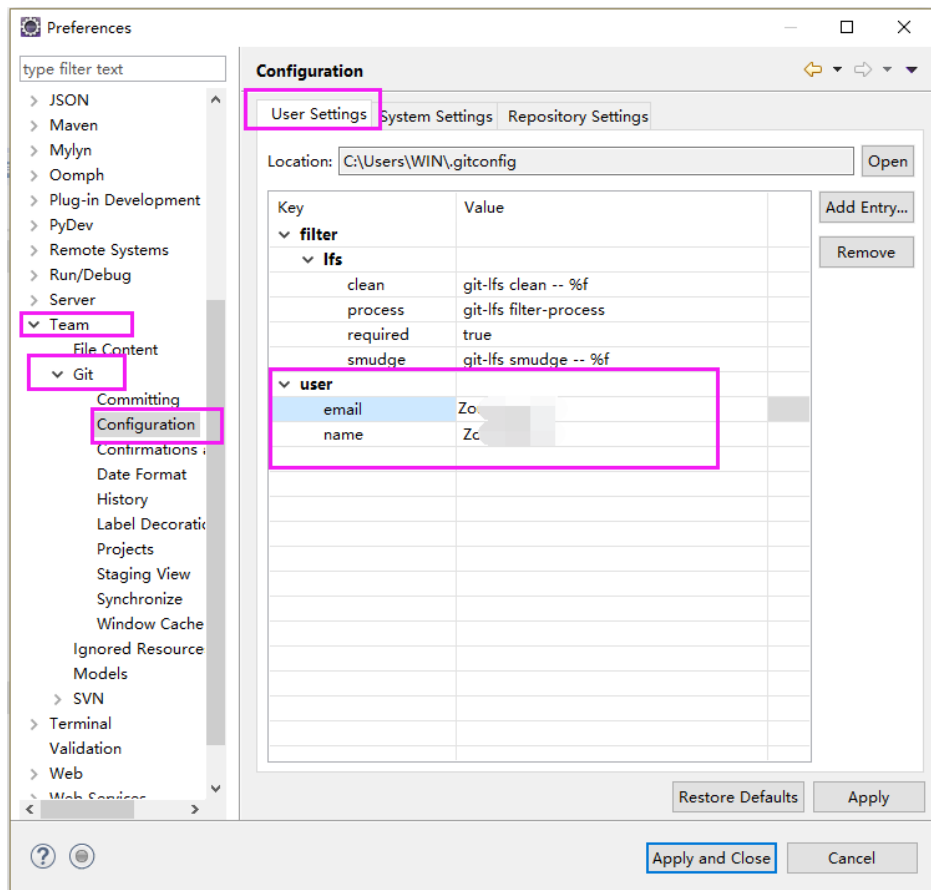




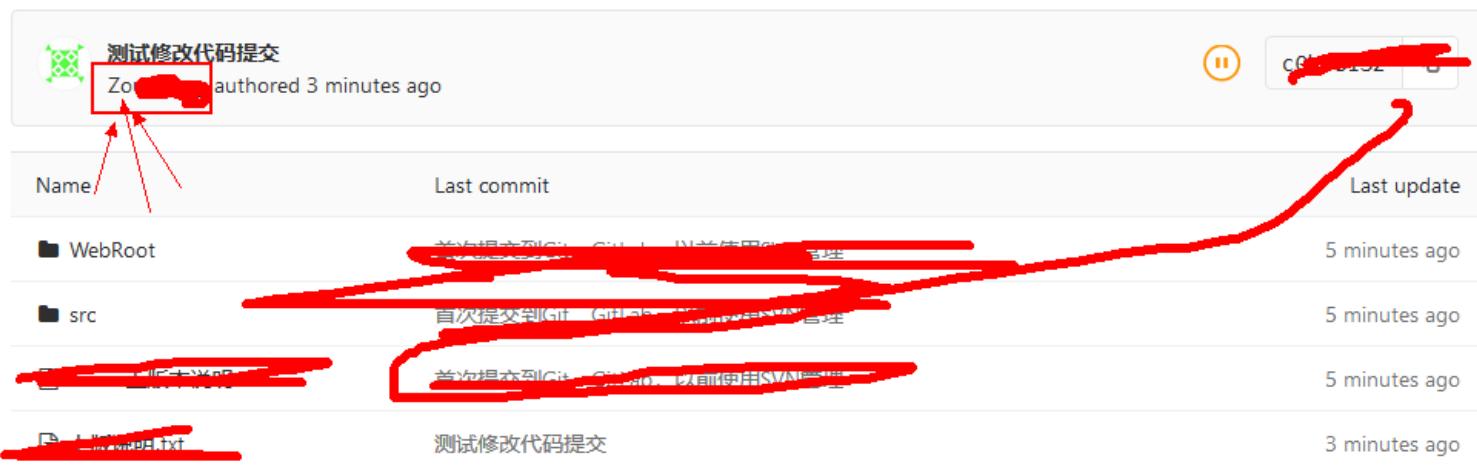
## 插播 修改提交者的名称

（eclipse 通过 Git 提交到 GitLab，提交代码的作者不是用 GitLab 的账号记录的）

（插播）去设置发现，上图那个 Author 是在这里设置的，要修改，不然提交的时候在 GitLab 那边是以这个名称显示提交者的，注意，修改之后如果不生效，可能要重启 eclipse。



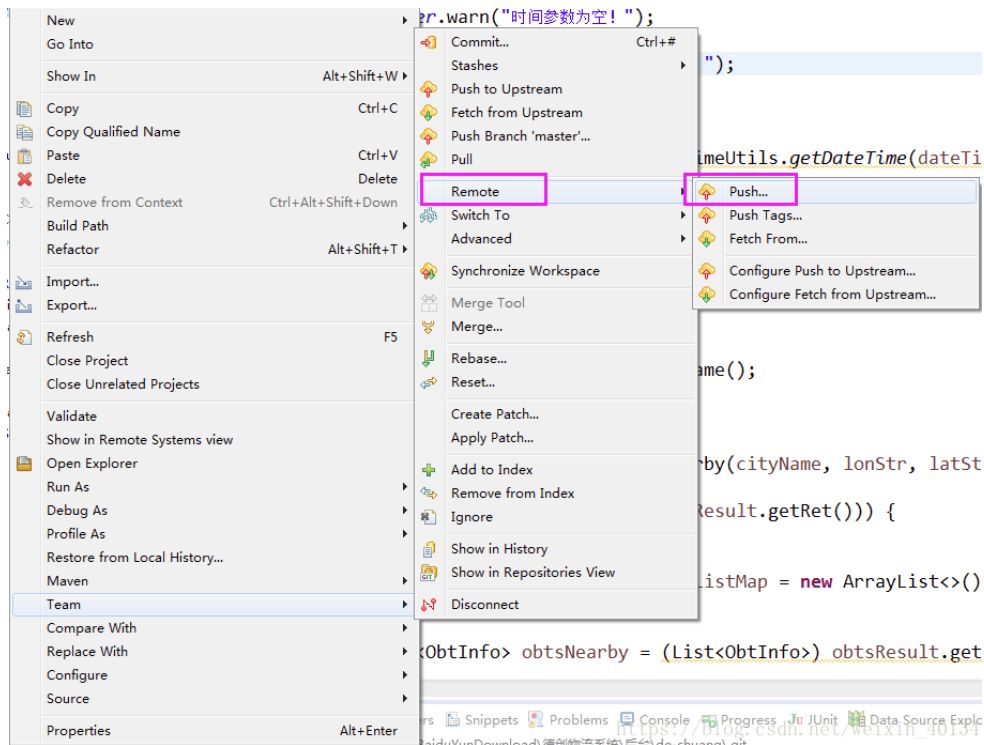
提交完成后，GitLab 仓库查看到



## Commit 与 Push 分开

当然也可以先 commit 到本地仓库，最后再 push 到远程仓库（别人的截图）





## GitLab 仓库配置

预 Commit、Commit 与 Push 分开 最后都会来到这里。这里的用户名、密码才是 GitLab 的密码。

Push Branch master

push

Destination Git Repository

Enter the location of the destination repository.

Remote name:

GitLab

随便写

Location

URI:

http://192.168.0.21:9999/backend/hsp.git

这个就是那个复制的仓库地址

Host:

192.168.0.21

Repository path:

/backend/hsp.git

Connection

Protocol:

http

Port:

9999

Authentication

User:

ng

用户密码

Password:

●●●●●●●●

☐ Store in Secure Store

?

< Back

Next >


Finish

Cancel


Push Branch master


Push to branch in remote

Select a remote and the name the branch should have in the remote.



Source:

 master

 e03c13b 首次提交

Destination:

Remote: GitLab: http://192.168.0.21:9999/backend/hsp.git


Branch: master

☒ Configure upstream for push and pull

When pulling: Merge

☐ Force overwrite branch in remote if it exists and has diverged

Show [advanced push](#) dialog



< Back

Next >

Finish


Cancel


注意，如果提交失败，则会有 Rejected....

Push Branch master

Push Confirmation

Confirm following expected push result.



 master → master [new branch]


Message Details

Repository

<http://192.168.0.21:9999/backend/hsp.git>

☐ Cancel push if result would be different than above because of changes on remote

☐ Show dialog with result only when it is different from the confirmed result above

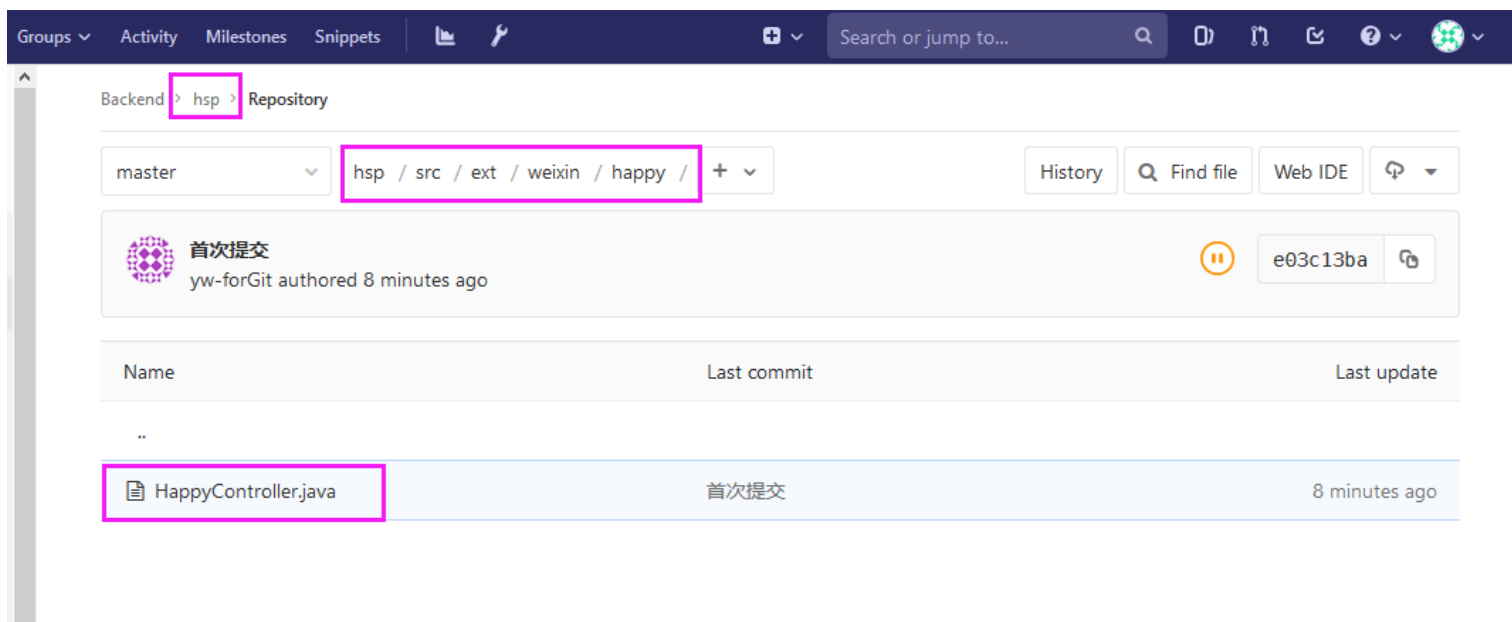
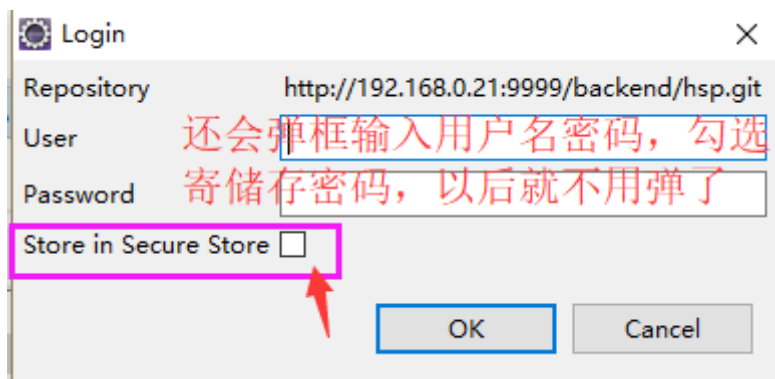


< Back

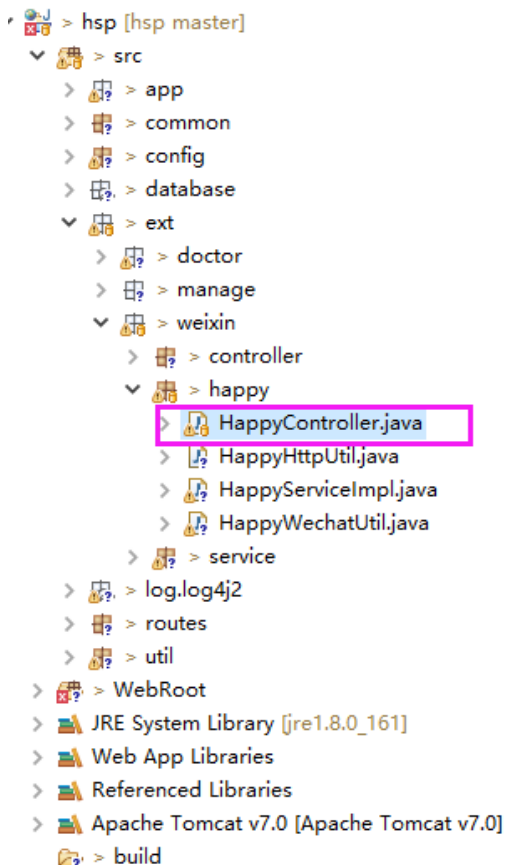
Next >

Finish

Cancel



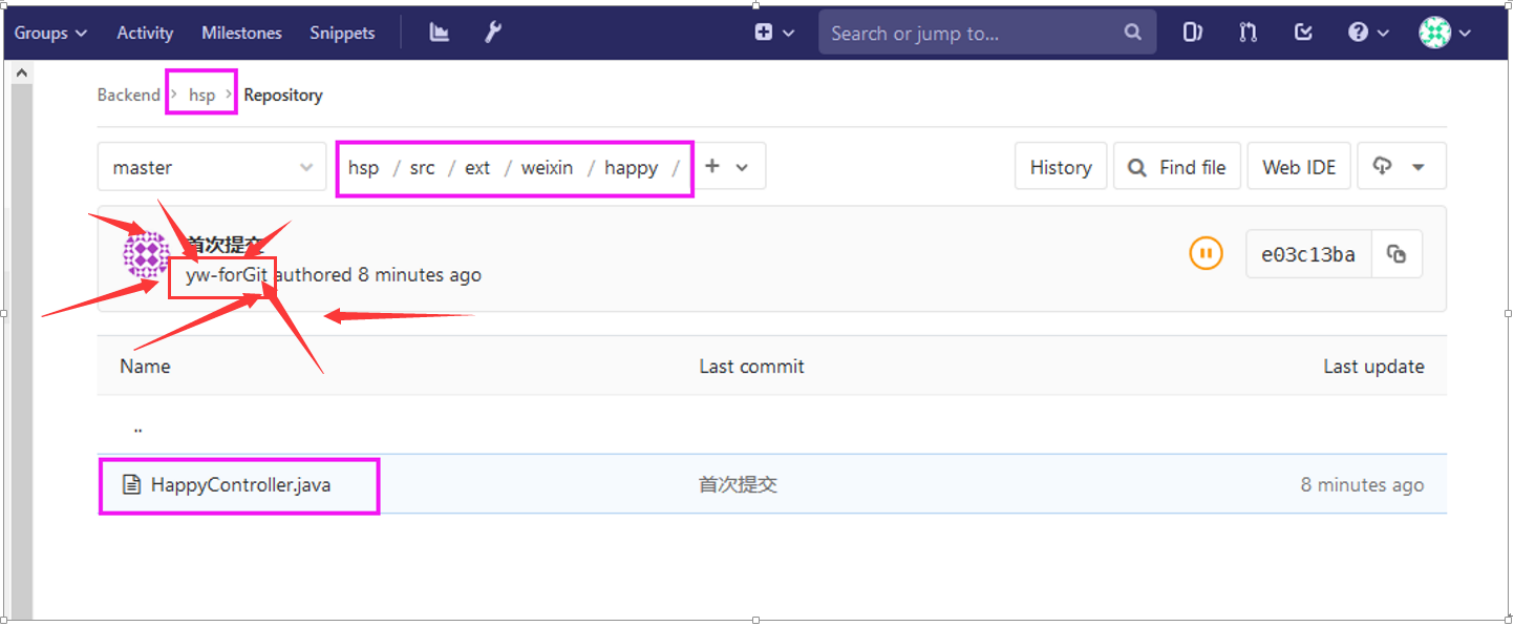
对应一下项目



注意，提交的项目最好要与 GitLab 创建的项目的名称一致，因为提交整个项目，其项目名称这个路径不会提交到 GitLab 上，

只会提交该项目目录下的东西到 GitLab。

注意点：  
可以看到 yw-forGit 名称就是我 eclipse 设置的 user 的 name。



（好像 master 分支要一次性提交完成代码，如果第一次只提交了一个文件，后面想一次性提交整个项目，会被 rejected，好像是这样的，然后我删除整个本地 git 仓库提交记录、GitLab 私有仓库的对应项目，重新提交就好了）

结束日期：2019-02-01