

GitLab用户权限管理详述

在安装完毕 **GitLab** 后，登录到页面后，我们可以看到是没有任何项目，而且只有一个用户。在使用的时候，需要创建项目，而且会有对应的用户在项目中。下面练习这个操作

- 新建用户组

- 1、点击右上角的扳手按钮
- 2、在弹出的对话框中选择 **new group**
- 3、**Private**:只有授权的用户才可以看到

Internal: 只要是登录上来的用户就可以看到

Public: 任何组和项目都可以看到

The screenshot shows the GitLab dashboard interface. At the top, there's a navigation bar with the GitLab logo and a search bar. Below the navigation bar, there's a table with various system components and their versions. The table has columns for component name, status, and version. The components listed are GitLab API, Git, Ruby, Rails, and PostgreSQL.

Component	Status	Version
GitLab API	●	v3
Git	⚙	2.7.4
Ruby	⚙	2.1.8p440
Rails		4.2.6
PostgreSQL		9.2.17

Below the table, there are three summary cards: Projects, Users, and Groups. Each card shows a large number representing the count of items and a 'New' button. The 'New Group' button in the Groups card is highlighted with a red box and a red arrow.

Projects

1

New Project

Latest projects

Users

4

New User

Latest users

Groups

1

New Group

Latest groups

Admin Area

Overview Monitoring Messages System Hooks Applications Abuse Reports

New group

Group path: **Group1**

Description:

Group avatar:
The maximum file size allowed is 200KB.

Visibility Level (?): ☒ **Private**
The group and its projects can only be viewed by members.

☐ Internal
The group and any internal projects can be viewed by any logged in user.

☐ Public
The group and any public projects can be viewed without any authentication.

- A group is a collection of several projects
- Members of a group may only view projects they have permission to access
- Group project URLs are prefixed with the group namespace
- Existing projects may be moved into a group

Admin Area

Overview Monitoring Messages System Hooks Applications Abuse Reports

Group was successfully created.

Group: Group1

Group info:

Name: **Group1**

Path: **Group1**

Description:

Visibility level: **Private**

Created on: **May 14, 2017 1:07pm**

Projects **0**

Add user(s) to the group:

[Read more about project permissions here](#)

- Administrator root
- dev1
- dev2
- pm

更多 **GitLab** 相关教程见以下内容:

Ubuntu 14.04 下安装 GitLab 指南 <http://www.linuxidc.com/Linux/2015-12/126876.htm>

如何在 Ubuntu Server 14.04 下安装 Gitlab 中文版 <http://www.linuxidc.com/Linux/2015-12/126875.htm>

CentOS 源码安装 GitLab 汉化版 <http://www.linuxidc.com/Linux/2015-10/124648.htm>

CentOS 7 下 GitLab 9.1.0 安装及汉化 <http://www.linuxidc.com/Linux/2017-04/143240.htm>

CentOS 7.2 安装 GitLab CE 图文详解 <http://www.linuxidc.com/Linux/2017-05/143538.htm>

搭建属于自己的 GitLab <http://www.linuxidc.com/Linux/2017-04/142665.htm>

Ubuntu 14.04 搭建 GitLab 服务器 <http://www.linuxidc.com/Linux/2017-02/140959.htm>

CentOS 6.5 安装 GitLab 教程及相关问题解决 <http://www.linuxidc.com/Linux/2014-05/101526.htm>

GitLab 的详细介绍: [请点击这里](#)

GitLab 的下载地址: [请点击这里](#)

- 组授权

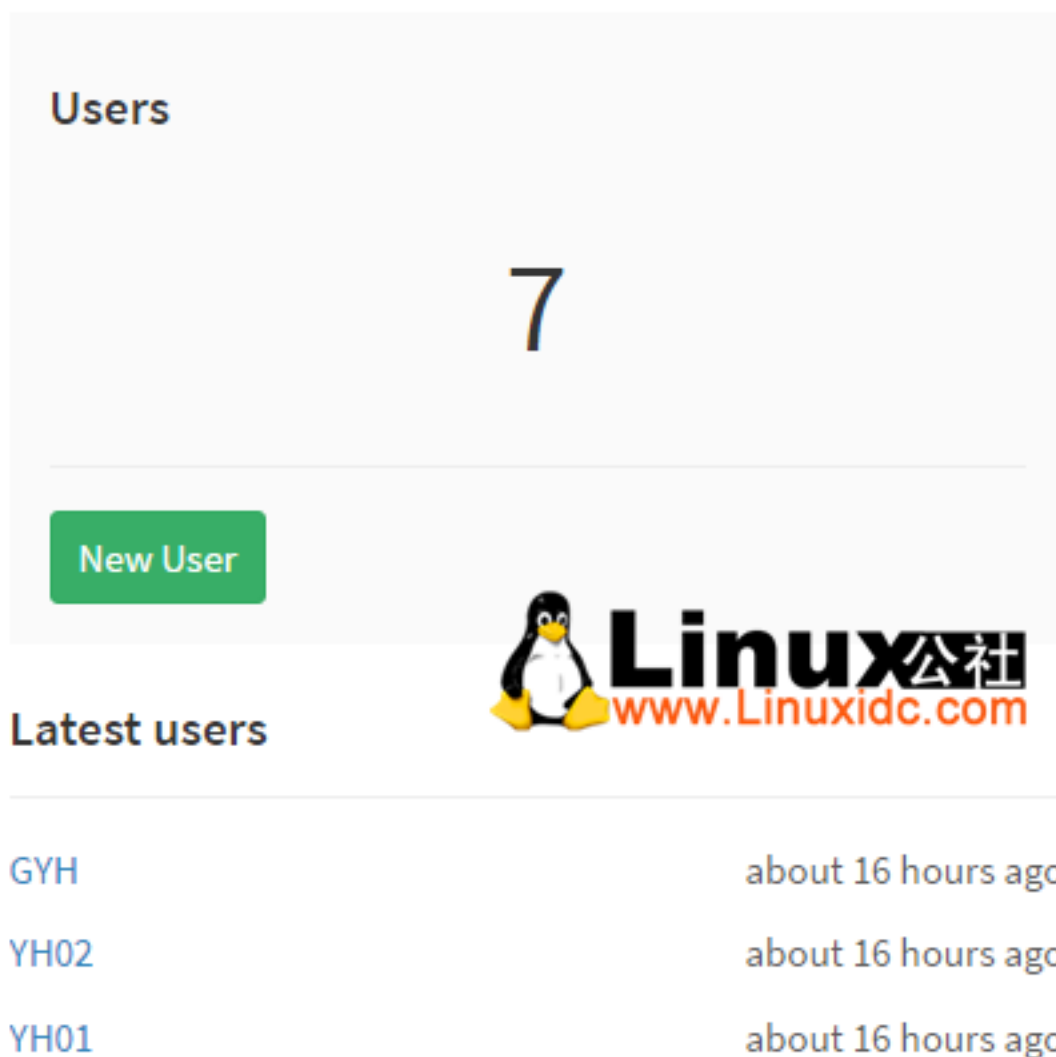
1、可以看到创建完组以后，可以给组内添加成员。下面创建几个用户

tips: 一个组可以有多个项目

- 用户创建

1、点击右上角的扳手按钮

2、在弹出的对话框中选择 **new user**，下面创建了一个管理用户 **GYH**，两个普通用户 **YH01** 和 **YH02**



The screenshot shows the GitLab 'Users' interface. At the top, the word 'Users' is displayed. Below it, a large number '7' indicates the total number of users. A green button labeled 'New User' is visible. At the bottom, there is a section titled 'Latest users' which lists three users: GYH, YH02, and YH01, each with a timestamp of 'about 16 hours ago'. A watermark for 'Linux公社' and 'www.Linuxidc.com' is present in the lower right area of the screenshot.

Username	Time
GYH	about 16 hours ago
YH02	about 16 hours ago
YH01	about 16 hours ago

给用户添加到授权组中，并修改权限

如图，这样我们就把 GYH 用户添加到 Group1 组中，并给予了 master 权限

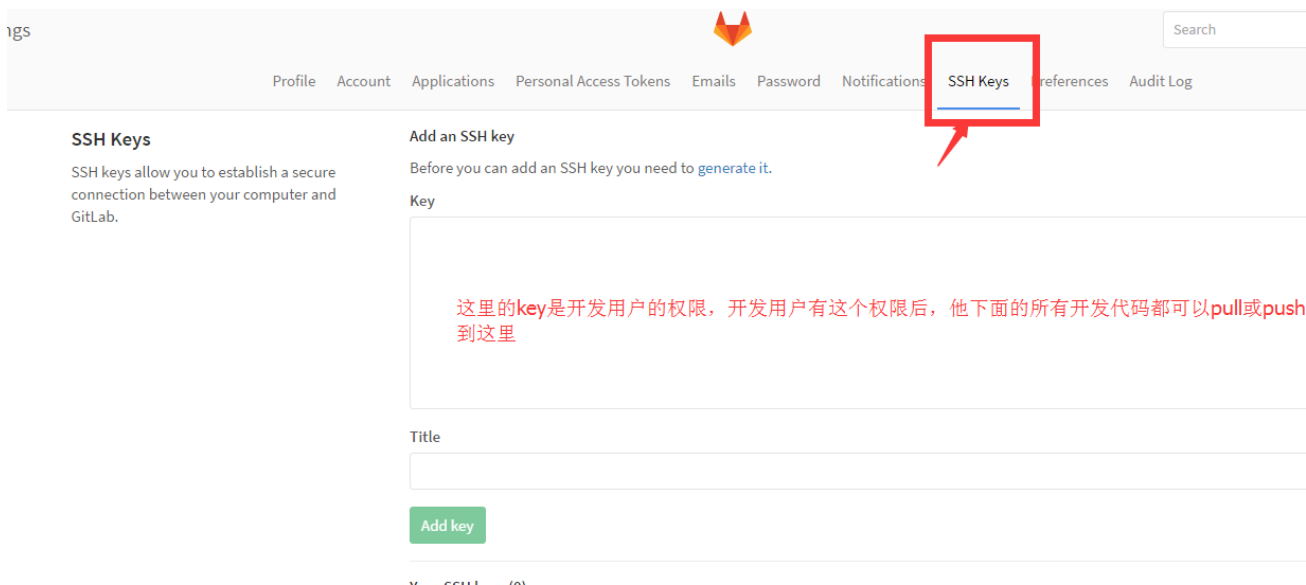
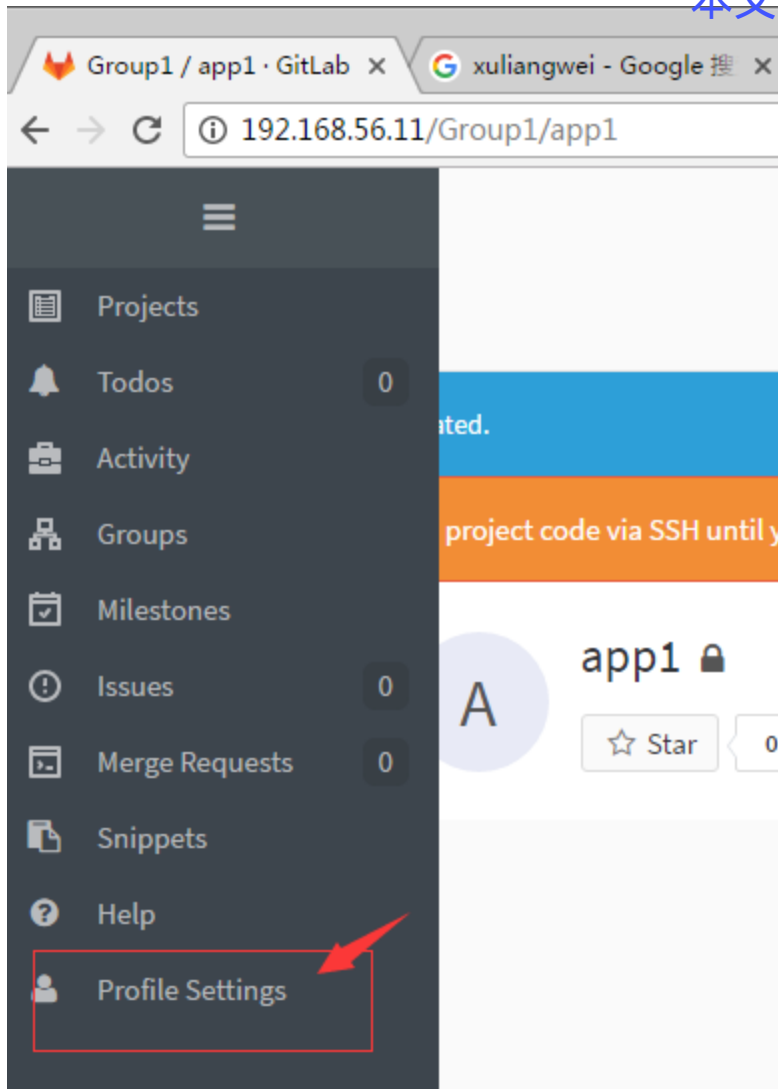
- 创建项目

- 1、点击右上角的扳手按钮
- 2、在弹出的对话框中选择 new project
- 3、选择项目拥有者
- 4、创建完项目，要授权项目

当我们创建完毕后，会有一个提示信息，如下，告知我们需要配置了 ssh 才能使用 ssh 协议来 push 或者 pull 这个项目

You won't be able to pull or push project code via SSH until you add an SSH key to your profile

下面我们就要对这个项目做权限处理，让它可以通过 ssh 来 pull 或者 push 动作



点击到上面的位置后，我们需要登录到需要授权连接到 gitlab 的设备上面。来复制密钥到 gitlab 上面。

下面以 Linux 设备 ip 为 56.11 为例，登录到此设备上

通过 ssh-keygen 一路回车来生成密钥，然后拷贝 pub 公钥到 gitlab 上去

```
[root@linux-node1 .ssh]# ll
total 16
-rw----- 1 root root 794 May 14 11:15 authorized_keys
-rw----- 1 root root 1679 May 7 14:11 id_rsa
-rw-r--r-- 1 root root 398 May 7 14:11 id_rsa.pub
-rw-r--r-- 1 root root 175 Feb 28 19:09 known_hosts
[root@linux-node1 .ssh]# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQ2isoX9dmXqj/SqD6zq8QNUTFWunEkzKcNZ5fdM/w9r53Tud2HW2rQk
4awnxji8rQSu1FijjZofEXr6D8fyjnYRWKsQzNZPxDsao+FtWoAdZAYgDjz5W2afTmiyXdexGPsFO0+ts73WGLaqvVS0Q
[root@linux-node1 .ssh]#
```

gs

Profile Account Applications Personal Access Tokens Emails Password Notifications SSH Keys Preferences Audit Log

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

Before you can add an SSH key you need to [generate it](#).

Key

ssh-rsa

AAAAB3NzaC1yc2EAAAADAQABAAQ2isoX9dmXqj/SqD6zq8QNUTFWunEkzKcNZ5fdM/w9r53Tud2HW2rQkxdL38aL/PJ9wDyrWAbkK600eS9KwE RCMnXIXOFjAxxO74ejhd2CP48o/dw91hWTtbtj/Noj2MlsVbmHj4P/+MDDkJe1owLkwk74W+mpoltcGQhCyg/4awnxji8rQSu1FijjZofEXr6D8fyjnYRW KsQzNZPxDsao+FtWoAdZAYgDjz5W2afTmiyXdexGPsFO0+ts73WGLaqvVS0Qw51H5E8SnOp9nkvx2MMk43ndCUj9J5vCQQZ19yg/coE/DQ2Wwgjqlc Ubx/wbdBVfbTzlgzbFoF13 root@linux-node1

Title

root@linux-node1

Add key

Your SSH keys (0)

点击 Addkey 按钮后，就添加成功！！

这里需要注意的是，**gitlab** 上面登录的账号是 **YH01** 是在这个账号下添加到 **SSH key**

如果不添加上面的 ssh 的话，你 clone 的时候会提示错误：错误提示如下图

```
[root@linux-node1 ~]# git clone git@192.168.56.11:java/app1.git
Cloning into 'app1'...
The authenticity of host '192.168.56.11 (192.168.56.11)' can't be established.
ECDSA key fingerprint is b0:b0:28:c4:05:d1:7a:92:90:9d:ed:21:4a:55:f7:0c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.11' (ECDSA) to the list of known hosts.
git@192.168.56.11's password:
```

给项目添加授权用户

Project: Group1 / app1

Edit

Project info:

Name: **app1**

Namespace: **Group1**

Owned by: **Group1**

Created by: **Administrator**

Created on: **May 14, 2017 1:19pm**

http: <http://192.168.56.11/Group1/app1.git>

ssh: <ssh://192.168.56.11:22/Group1/app1.git>



Group1 group members 2

Manage Access

GYH GYH

Master

Administrator root It's you

Owner

app1 project members 0

Manage Access

添加完毕后，这个项目拥有了 4 个用户，其中两个是组里面继承过来的，另外两个是这个项目的用户，如图

Add new user to project Import members

Users with access to this project are listed below.

People

Search for users by name, username, or email, or invite new ones using their email address.

Project Access Guest

[Read more about role permissions here](#)

Add users to project

app1 project members 2

Find existing member by name

Q

YH02 YH02

Developer

YH01 YH01

Developer

Group1 group members 2

Manage group members

GYH GYH

Master

Administrator root It's you

Owner



当完成上面的设置后，使用 YH01 账号登录后，在登录界面就能看到他所拥有的项目了

Projects Signed in successfully. YH01

Your Projects Starred Projects Explore Projects

Filter by name... Last updated New Project

A Group1 / app1

您查看的内容来自
Linux公社 www.linuxidc.com

这个时候就可以从客户通过 git clone 来下载项目了

```
[root@linux-node1 ~]# git clone git@192.168.56.11:Group1/app1.git
```

Cloning into 'app1'...

The authenticity of host '192.168.56.11 (192.168.56.11)' can't be established.
ECDSA key fingerprint is 28:3e:e9:97:c9:ff:33:16:41:23:0a:14:30:7c:5c:7f.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.56.11' (ECDSA) to the list of known hosts.

```
warning: You appear to have cloned an empty repository.
Checking connectivity... done.
[root@linux-node1 ~]# ll
total 277436
-rw-----. 1 root root 990 Dec 26 18:21 anaconda-ks.cfg
drwxr-xr-x 3 root root 17 May 14 22:30 appl
```

这里由于 **appl** 项目是一个空的目录，下面我们在这个目录上面新建一个文件 **readme**

然后使用 **git pull** 来测试拉取

```
[root@linux-node1 appl]# git pull
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From 192.168.56.11:Group1/appl
 * [new branch] master -> origin/master
[root@linux-node1 appl]# ll
total 4
-rw-r--r-- 1 root root 47 May 14 22:37 readme
[root@linux-node1 appl]# cat readme
readme 创建一个床主主峰之
#这里可以看到，把 master 上面的 readme 文件拉取下来了
```

二 Windows 客户端建立 ssh 免密钥登录

1、安装工具

Git-2.7.2-32-bit_setup.1457942412.exe

2、在一个目录里面选择右键，选择“Git Bash Here”，会弹出一个 shell 窗口，这个窗口里面执行和 Linux 一样的 ssh 密钥生成命令。结果也会生成一个密钥

```
Tony@Tony-PC MINGW32 ~/.ssh
```

```
$ cat id_rsa.pub
```

```
ssh-
```

```
rsa AAAAB3NzaC1yc2EAAAADAQABAAQACxafaZna8H+91hB4+GYPI2RcOzZhD1eHx8sK8po5bq5wwk+tgRUSG
QiLg8mA+HAXSIueylsKA0sfBCajP9k2hJ05va0xr+U0qZ0yEytYq1QsOWIgX+VOAwkJDwGYgeZ7jZ3N0knem4GK
8NRSWeoSxseoZs9ddeFk3YVAaVykLDDngZ0YKsBkP7rjPn0JnFau8Me75cfIjuTnMVYPGY5YPJnN9MULjQvZB9
mJ34E05a/4cAXZBSVKtumasBnIjTtIbRFNVxcRjDohNdak1kZgdmOwMSyeCX+ADcV5vwgGZNxxWkRpOFV+W9vI/N
qIpGktj7HD12XXtzxi8nnuozGe0SL Tony@Tony-PC
```

```
Tony@Tony-PC MINGW32 ~/.ssh
```

3、把 windows 上面这个加到 YH02 用户上面

4、在 gitlab 上用 YH02 登录上去，在 profiles setting 设置里面添加 SSH Key 添加上去，

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

Before you can add an SSH key you need to generate it.

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAxZna8H+91hB4+GYPI2RcOzZhDleHx8sK8po5bq5wwk+tgRUSGQilG8mA+HAXSlueylsKA0sfBCajP9k2hJ0
5vaOxr+UOqZ0yEytYqIQs0WlgX+VOAwkJDwGYgeZTjZ3N0knem4GK8NRSWesoSwXseoZs9ddeFk3YVaaVykLDDngZ0YKsBkP7rjPn0JnFau8Me75cfjuT
nMVYPGY5YPJnN9MULjQvZB9mJ34E05a/4cAXZBSVKtumasBnljTlbRFNVxcRjDohNdak1kZgdmOwMSyeCX+ADcV5vwgGZNXxWkRpOFV+W9vl/NqlpG
ktj7HD12XXtzi8nnuozGe0SL Tony@Tony-PC
```

YH02用户

Title

Tony@Tony-PC

Add key

您查看的内容来自
Linux公社 www.linuxidc.com

5、从 windows 客户端上 pull 代码下来。

```
Tony@Tony-PC MINGW32 ~/.ssh
$ git clone git@192.168.56.11:Group1/appl.git
Cloning into 'appl'...
The authenticity of host '192.168.56.11 (192.168.56.11)' can't be established.
ECDSA key fingerprint is SHA256:l8F70nGgtB8t1VEWnooakloCP6UsZ3GJx1MPhssMv04.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.11' (ECDSA) to the list of known hosts.
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.
Checking connectivity... done.
```

```
Tony@Tony-PC MINGW32 ~/.ssh
$ ll
total 6
drwxr-xr-x 1 Tony 197121 0 五月 15 14:36 appl/
-rw-r--r-- 1 Tony 197121 1675 五月 14 11:34 id_rsa
-rw-r--r-- 1 Tony 197121 394 五月 14 11:34 id_rsa.pub
-rw-r--r-- 1 Tony 197121 175 五月 15 14:36 known_hosts
```

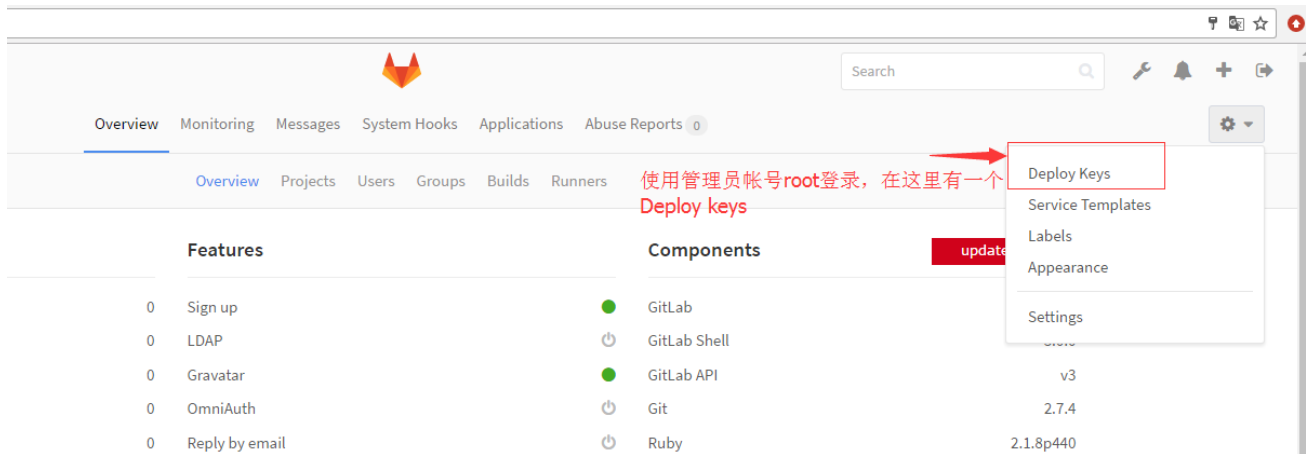
```
Tony@Tony-PC MINGW32 ~/.ssh
$ cd appl/
```

```
Tony@Tony-PC MINGW32 ~/.ssh/appl (master)
$ ll
total 1
-rw-r--r-- 1 Tony 197121 49 五月 15 14:36 readme
```

```
Tony@Tony-PC MINGW32 ~/.ssh/appl (master)
$ cat readme
readme 创建一个床主主峰之
```

```
master brach
Tony@Tony-PC MINGW32 ~/.ssh/appl (master)
$
```

Deploy Key



每个项目都有个 deploy key，项目部署公钥（Deploy Key）允许通过 SSH 协议以只读的方式访问项目，不需要输入密码，而且数据是使用你上传的公钥加密传输的。与 HTTPS 协议相比，SSH 协议的数据传输效率要更高和稳定些，支持超大项目数据的传输。使用 deploy key 后发现只能 pull 不能 push

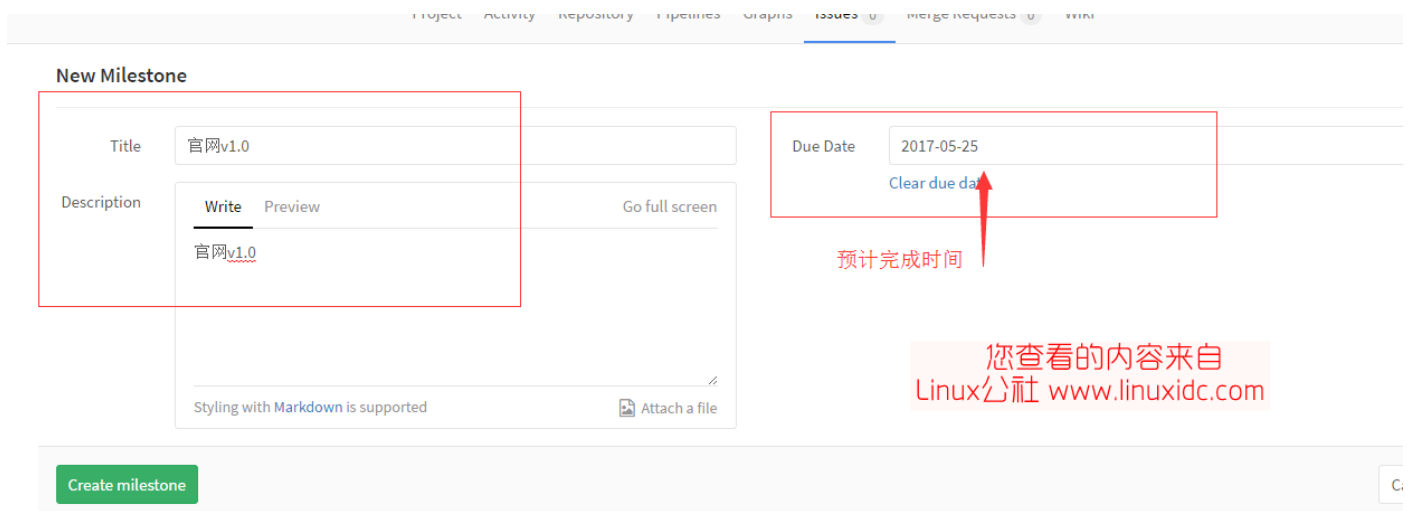
这个一般是给 jenkins 使用的。

三 GitLab 开发计划实施

假设现在已经有了一个开发计划，怎么做 gitlab 上实施这个开发计划那？？？

这里我们使用管理账号 GYH 来当项目经理，登录到 gitlab 上面

在左面的菜单中选择 Milestones 里程碑，--》new milestones



新建新的 issue 来分配任务

app1 ▾

This project Search

Project Activity Repository Pipelines Graphs **Issues 0** Merge Requests 0 Wiki

New Issue

Title 首页

Description **Write** Preview Go full screen

为什么开发

您查看的内容来自
Linux公社 www.linuxidc.com

Styling with [Markdown](#) is supported Attach a file

☐ This issue is confidential and should only be visible to team members with at least Reporter access.

指派给谁开发

Assignee YH01 Assign to me

Due date 2017-05-17 完成时间

Milestone 官网v1.0 属于哪个里程碑

[Create new milestone](#)

Labels No labels yet. [Create new label](#)

[Submit issue](#) [Cancel](#)

依次创建 4 个 issue

Project Activity Repository Pipelines Graphs **Issues 4** Merge Requests 0 Wiki

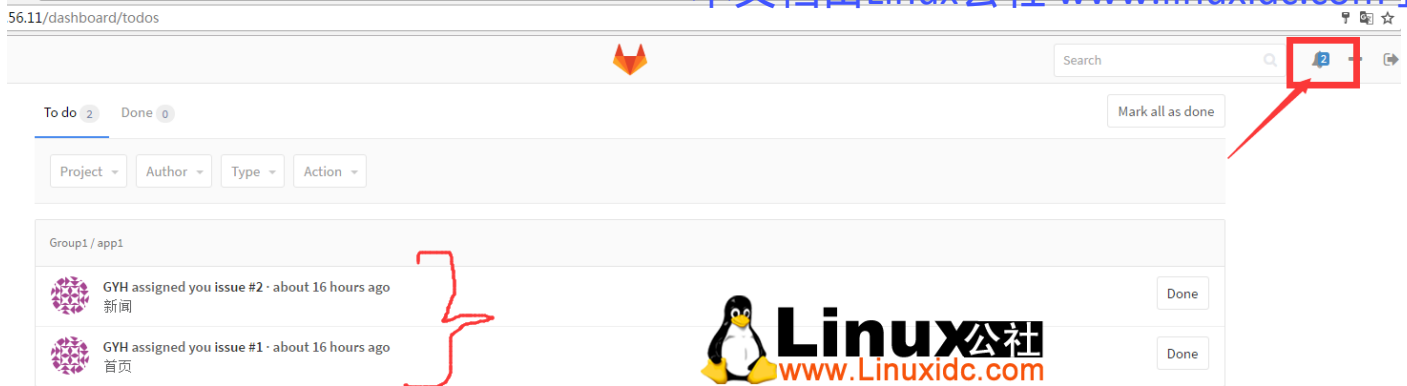
Issues Labels Milestones

Open 4 Closed 0 All 4 Filter by name ... [New Issue](#)

<input type="checkbox"/>	Author ▾	Assignee ▾	Milestone ▾	Label ▾	Last created ▾
<input type="checkbox"/>	关于				
	#4 · opened about 16 hours ago by GYH	官网v1.0	May 21, 2017		updated about 16 hours ago 0
<input type="checkbox"/>	支付				
	#3 · opened about 16 hours ago by GYH	官网v1.0	May 20, 2017		updated about 16 hours ago 0
<input type="checkbox"/>	新闻				
	#2 · opened about 16 hours ago by GYH	官网v1.0	May 19, 2017		updated about 16 hours ago 0
<input type="checkbox"/>	首页				
	#1 · opened about 16 hours ago by GYH	官网v1.0	May 17, 2017		updated about 16 hours ago 0

您查看的内容来自
Linux公社 www.linuxidc.com

当我们使用 YH01 或者 YH02 登录到 gitlab 时候就能看到分配给自己的任务



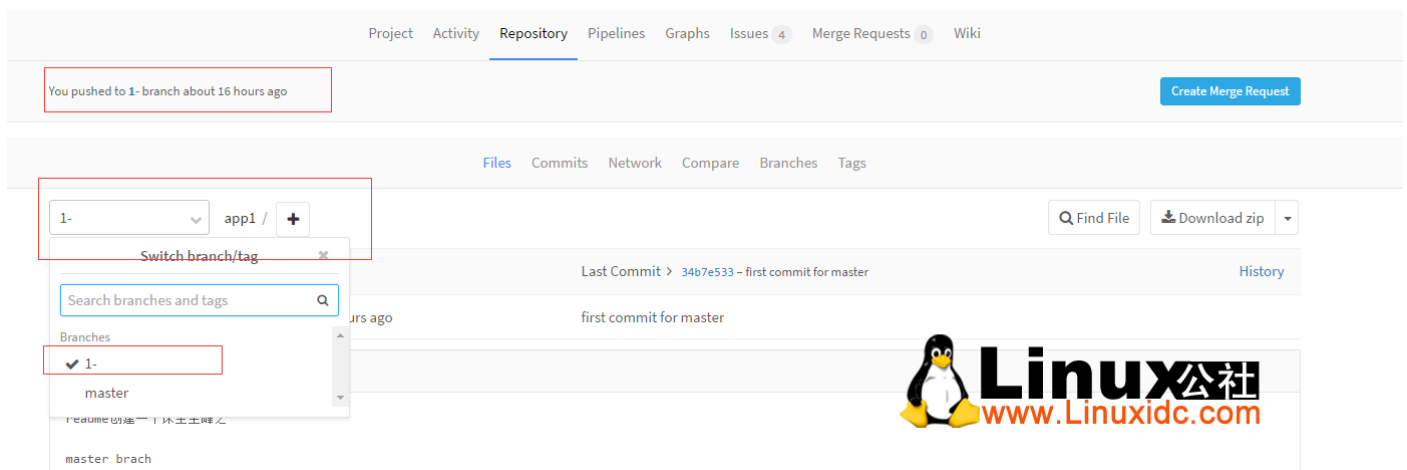
开发者开始开发的流程

1、首先要创建一个新的分支

(1) 可以通过命令创建分支

```
[root@linux-node1 appl]# git checkout -b shouye
Switched to a new branch 'shouye'
[root@linux-node1 appl]# git status
On branch shouye
nothing to commit, working directory clean
```

(2) 在页面直接创建分支，并且会以当前 issue 的 ID 来命名分支



2、在分支上开发程序，并推送到分支上

```
[root@linux-node1 appl]# echo "shou ye wen jian" >index.html
[root@linux-node1 appl]# git add .
[root@linux-node1 appl]# git commit -m "shouye"
[shouye 1f9cf77] shouye
1 file changed, 1 insertion(+)
create mode 100644 index.html
```

```
[root@linux-node1 appl]# git log
commit 1f9cf77118cc8c8ec704bc1458db128fc25f39d2
Author: kesungang <605937099@qq.com>
Date: Sun May 14 23:25:17 2017 +0800
```

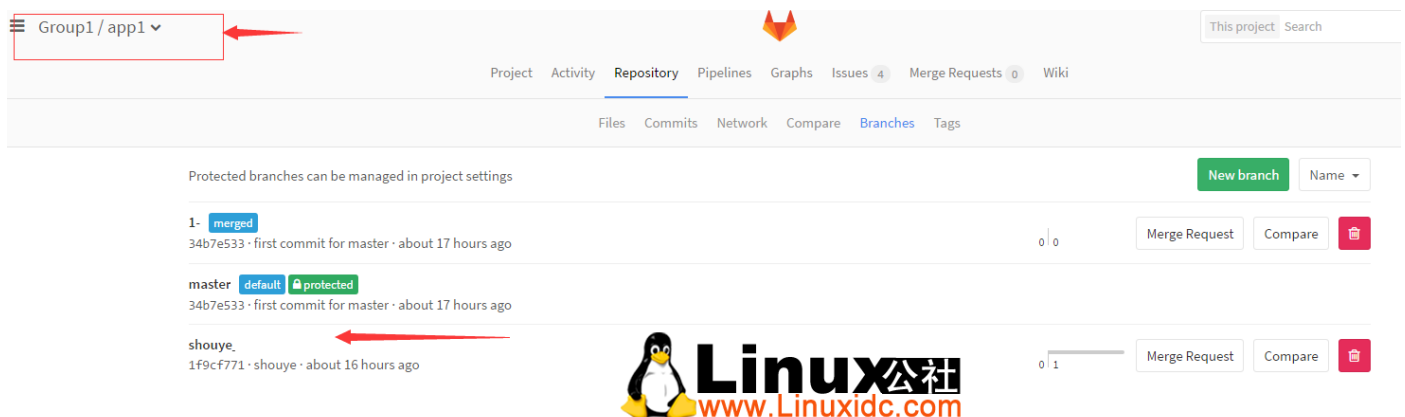
shouye

```
commit 34b7e533c414b0f27ab2d1e209e145bc5d0afd1b
Author: YH01 <yh01@126.com>
Date: Sun May 14 22:35:42 2017 +0800
```

first commit for master

```
[root@linux-node1 appl]# git push origin shouye
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 284 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@192.168.56.11:Group1/appl.git
 * [new branch] shouye -> shouye
```

gitlab 上在用户的项目里面可以查看到分支，并且点击分支，可以看到你的 commit 记录



3、开发完毕，分支请求合并到 master 分支上去

(1)、可以从页面上的 shouye 分支，点击右边的 Merge Request

Project Activity Repository Pipelines Graphs Issues 4 Merge

Title shouye

Start the title with **WIP:** to prevent a **Work In Progress** merge request from being merged b

Description

Write Preview

开发完毕合并到主分支

Styling with **Markdown** is supported

Assignee GYH

Assign to me

Milestone 官网v1.0

Labels No labels yet.
Create new label

Source branch shouye

Target branch master Change branches

☐ Remove source branch when merge request is accepted.

由于是项目经理创建的里程碑，所以GYH有权限关闭里程碑，我们要指派给GYH



(2) GYH 项目经理登录到 gitlab 上，可以从有上面的小铃铛，提示有任务，如果检查没有问题，接受合并请求

Group1/app1

Project Activity Repository Pipelines Graphs Issues 4 Merge Requests 1 Wiki

Open Merge Request !1 opened about 16 hours ago by YH01 Close Edit

shouye

开发完毕合并到主分支

Edited about 16 hours ago

Request to merge shouye into master

Accept Merge Request Remove source branch Modify commit message

You can also accept this merge request manually using the **command line**.

Discussion 1 Commits 1 Changes 1

Linux公社 www.Linuxidc.com

Todo Mark Done

Assignee GYH @GYH

Milestone 官网v1.0

2 participants

Notifications Unsubscribe

You're receiving notifications because you're subscribed to this thread.

(3) 测试正常合并的，可以关闭分支了

(4) 分支更新 (其他人就可以从 master 上面拉下最新内容了)

```
[root@linux-node1 appl]# git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.
[root@linux-node1 appl]# ll
total 4
-rw-r--r-- 1 root root 47 May 14 22:37 readme
[root@linux-node1 appl]# git pull
remote: Counting objects: 1, done.
remote: Total 1 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (1/1), done.
From 192.168.56.11:Group1/appl
   34b7e53..1f84445  master           -> origin/master
   * [new branch]      1-             -> origin/1-
Updating 34b7e53..1f84445
Fast-forward
 index.html | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 index.html
[root@linux-node1 appl]# ll
total 8
-rw-r--r-- 1 root root 17 May 15 00:13 index.html
-rw-r--r-- 1 root root 47 May 14 22:37 readme
```

另外一种 commit 自动关闭分支任务

```
-rw-r--r-- 1 root root 47 May 14 22:37 readme
[root@linux-node1 appl]#
[root@linux-node1 appl]# git checkout -b news
Switched to a new branch 'news'
[root@linux-node1 appl]# git status
On branch news
nothing to commit, working directory clean
[root@linux-node1 appl]# echo "xinwen" >news.html
[root@linux-node1 appl]# ll
total 12
-rw-r--r-- 1 root root 17 May 15 00:13 index.html
-rw-r--r-- 1 root root  7 May 15 00:16 news.html
-rw-r--r-- 1 root root 47 May 14 22:37 readme
[root@linux-node1 appl]# git add .
[root@linux-node1 appl]# git status
On branch news
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   news.html

[root@linux-node1 appl]# git commit -m "close #2"#使用#2 格式来提交
```

```
[news 9def595] close #2
 1 file changed, 1 insertion(+)
 create mode 100644 news.html
[root@linux-node1 appl]# git status
On branch news
nothing to commit, working directory clean
[root@linux-node1 appl]# git push origin news
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@192.168.56.11:Group1/appl.git
 * [new branch]      news -> news
```


本文档由Linux公社 www.linuxidc.com 整理
欢迎点击这里的链接进入精彩的[Linux公社](http://www.linuxidc.com)网站

Linux公社（www.Linuxidc.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

[Linux公社](http://www.linuxidc.com)是专业的Linux系统门户网站，实时发布最新Linux资讯，包括Linux、Ubuntu、Fedora、RedHat、红旗Linux、Linux教程、Linux认证、SUSE Linux、Android、Oracle、Hadoop、CentOS、MySQL、Apache、Nginx、Tomcat、Python、Java、C语言、OpenStack、集群等技术。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

Linux公社 主站网址：www.linuxidc.com 旗下网站：www.linuxidc.net

包括：[Ubuntu 专题](#) [Fedora 专题](#) [Android 专题](#) [Oracle 专题](#) [Hadoop 专题](#)
[RedHat 专题](#) [SUSE 专题](#) [红旗 Linux 专题](#) [CentOS 专题](#)



Linux 公社微信公众号：[linuxidc_com](#)



微信扫一扫

Linuxidc.com

订阅专业的最新Linux资讯及开源技术教程。

搜索微信公众号：[linuxidc_com](#)