**GitLab使用指南**

说明：阅读本文档需了解一些git工具使用的基础知识，本文档使用Git bash窗口进行git命令演示。

# 一、简介

### 1.1 Git

一个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

### 1.2 Gitlab

与github类似，提供代码托管服务，目前已经在内网搭建了一套gitlab环境，地址为[http://128.160.184.30/users/sign\_in](http://128.160.184.161/users/sign_in)

# 二、使用前准备

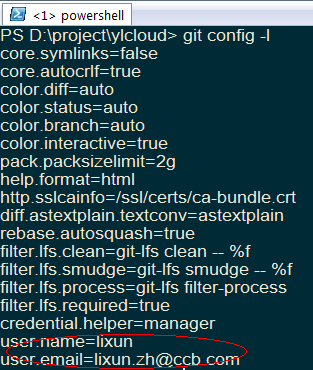
### 2.1 安装git工具

使用下列命令配置自己的用户名与邮箱，如不配置此项，那么git就无法知道每次提交代码的人是谁

git config --global user.name “username”

git config –global user.email “useremail”

配置后输入git config –l查看是否配置成功



### 2.2 注册gitlab账号

大家去[http://128.160.184.30/users/sign\_in](http://128.160.184.161/users/sign_in)自行注册，建议使用名字全拼与工作邮箱，gitlab支持http与ssh两种方式提交代码：

如果使用http，则每次需输入用户名与密码

如果使用ssh，则不需要每次输入用户名与密码，下面说说配置ssh：

（1）使用命令ssh-keygen -t rsa -C “youremail”生成ssh公钥

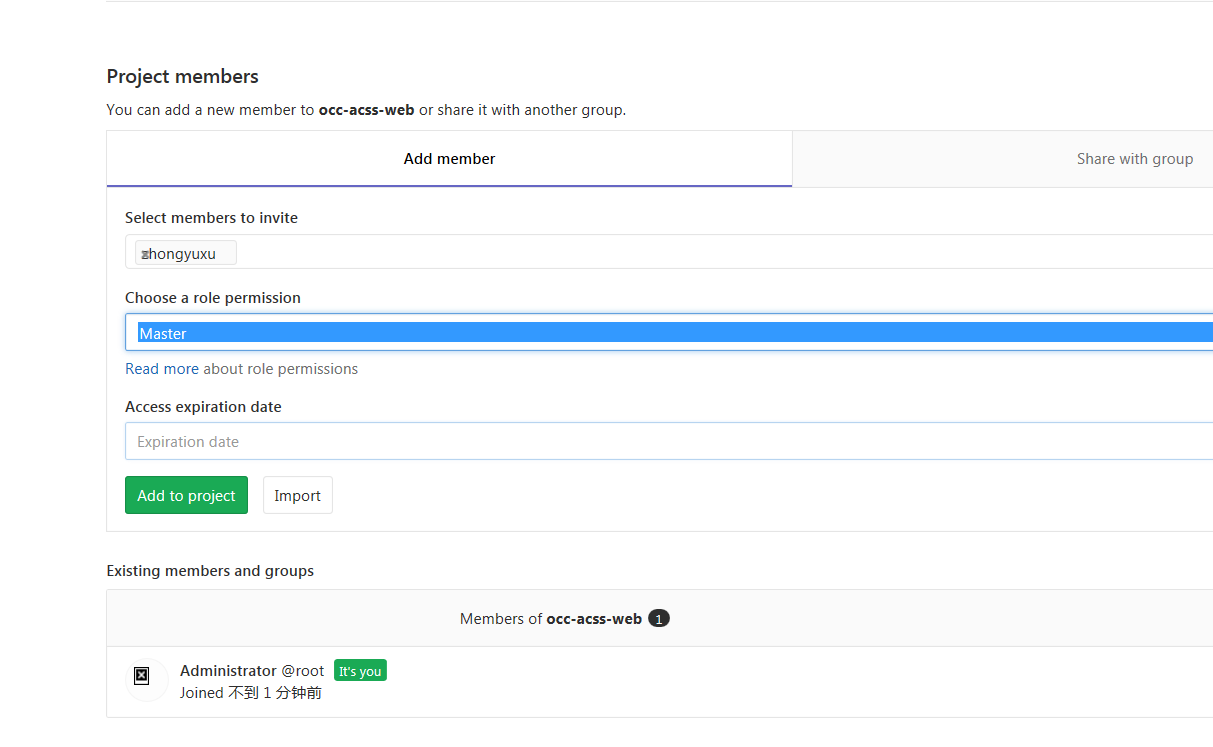
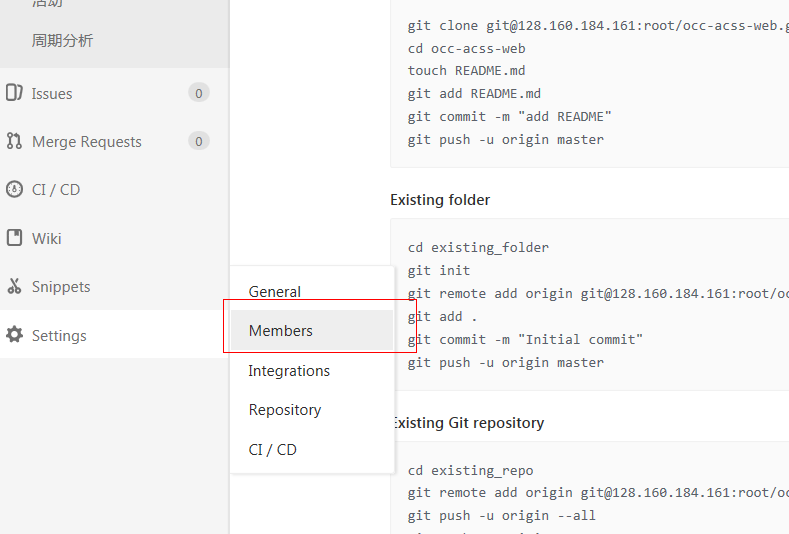
（2）去用户目录下找到.ssh文件夹（例如我的是C:\Users\lixun\.ssh），复制id\_rsa.pub文件里面的内容

（3）登陆gitlab，点击右上角图像下面的setting，然后点击SSH Keys，将自己的ssh公钥填加进来

# 三、使用步骤

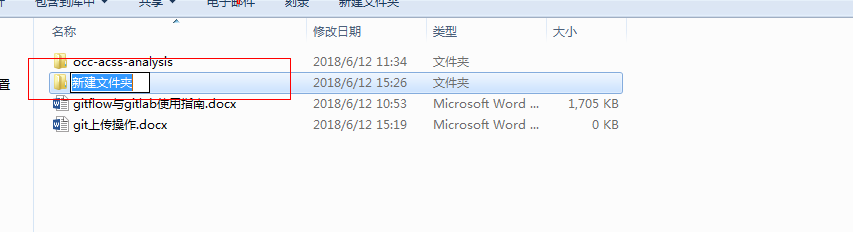
### 3.1 上传本地工程到远端gitlab服务器

1. 登录http://128.160.184.30/root/ ，选择newproject；设置member权限。

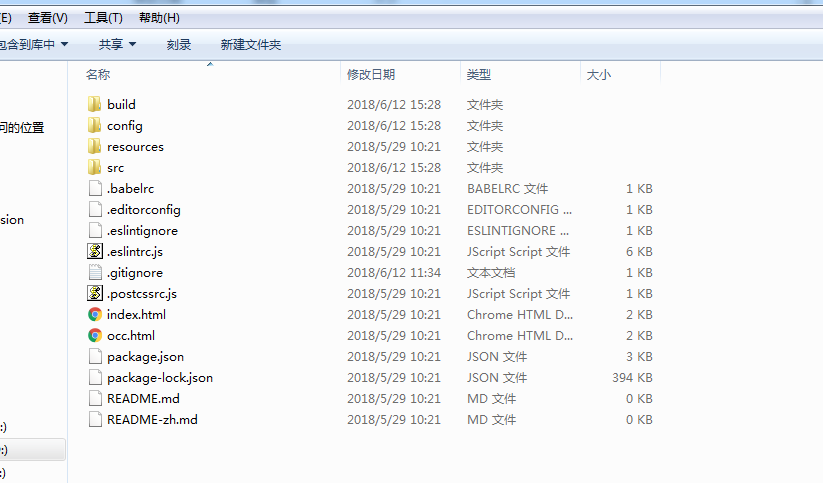


或者选择groups目录下某个group新建工程，这样新建的工程会初始化该group的属性作为工程的默认属性，包括工程的member、Visibility Level等属性。

1. 本机新建工程目录。



1. 拷贝.gitignore文件及代码到工程目录。



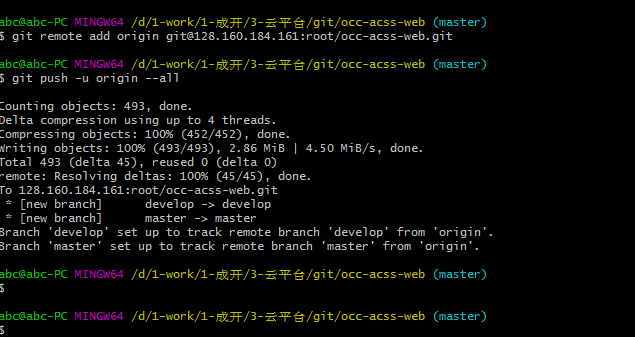
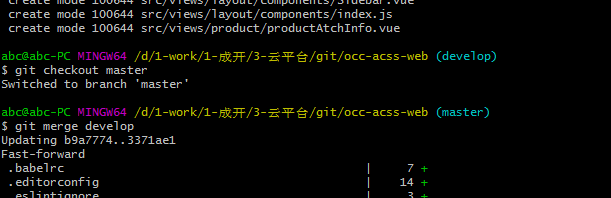
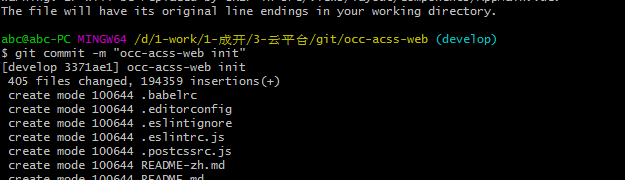
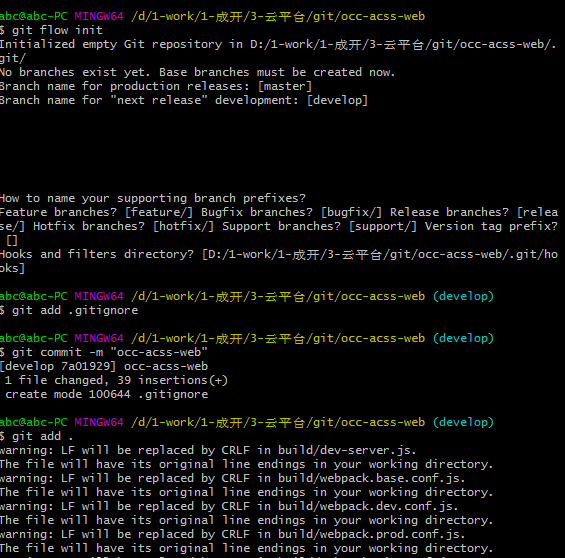
1. 在工程目录下右键打开git Bash。
2. 执行git add .gitignore ; git commit -m “add projectname .gitignore file”上传.gitignore文件到本地仓库。
3. 执行git add . ; git commit –m “projectname init” 上传工程代码到本地仓库。
4. 执行git checkout master 切换到master分支。
5. 执行git merge develop 同步develop到master分支。
6. 回到[http://128.160.184.30/root/occ-acss-web查看工程overview/](http://128.160.184.161/root/occ-acss-web查看工程overview/)详情，选择ssh然后

Copy“Existing Git repository”命令到gitbash执行：

git remote add origin [git@128.160.184.30:root/occ-acss-web.git](mailto:git@128.160.184.161:root/occ-acss-web.git)

git push –u origin –all

1. 最后刷新[http://128.160.184.30/root/occ-acss-web](http://128.160.184.161/root/occ-acss-web)，可以看到新上传的工程代码。
2. setting的member菜单可以管理该工程的用户权限（如果工程属于某个group则初始化该group的属性作为工程的默认属性）。



### 3.2 初始化分支

本地工程上传到git服务器默认只有develop分支，如果需要把develop分支初始化到master分支，可以在本地工程的git-bash执行命令：git checkout master; git merge develop；git push;

### 3.3 管理分支

1. 使用命令git branch -a检查所在分支



如上图所示，目前处于master分支

1. 使用命令git checkout 切换分支

（1）使用下面命令切换到<branch name>分支

git checkout <branch name>

（2）使用下面命令使<branch name>分支与远程<branch name>分支关联

git branch –set-upstream-to=origin/develop <branch name>

（3）使用下面命令同步本地和远程服务器分支

git fetch –prune origin

1. 新增删除分支
2. 删除本地分支命令

git branch –d <branch name>

1. 强制删除本地分支命令（有风险）

git branch –D <branch name>

1. 删除远程分支命令

git pull origin –delete <branch name>

1. 分支代码管理：
2. add分支代码命令，以develop分支为例：

git checkout develop;

git add .;(所以未上传代码)

git add <文件名>；

1. commit代码到本地仓库；

git commit –m <commit内容简要说明>；

1. 上传代码到远程分支命令：

git checkout develop;

git push;

git push origin develop;

1. 更新某分支代码命令：

git checkout develop;

git pull;

1. 查看分支状态命令

git status;

分支处于clean状态为正常，如有未上传代码可使用（1）（2）（3）上传代码.