

文一实验室git服务器使用说明

2021/01/14 韩鑫焱

简介

文一实验室git服务器现位于文一校区新教楼404，主要用于代码、文档及资料托管和归档，现可在杭州电子科技大学内网下使用，外网使用需要借助公有IP及外部服务器（购买云服务器可实现）。

此外，另开设Github organization用于实验室项目协作研发和代码托管，目前Github大部分功能免费开放，团队协作不限制公开/私有库和协作人数，组织名称：hduorg。

文一实验室git服务器

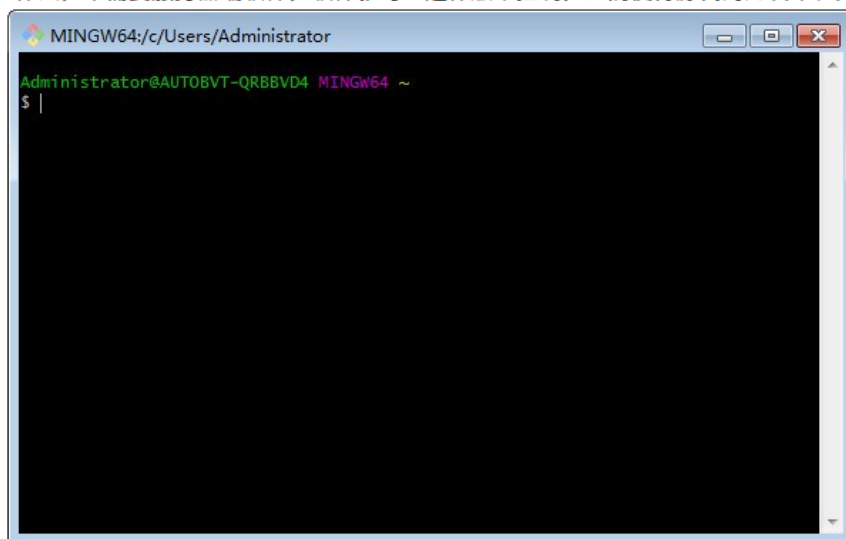
文一实验室服务器IP地址：192.168.1.52，用户名：git，密码：123（需连接文一404路由器）

校园网（下沙、文一）内网IP地址：10.63.110.231，转发端口：6000

下面所有操作在Linux和Windows下都可执行，Windows建议安装Git客户端。



Git客户端官方版是一款非常专业的开放源码分布式版本控制系统，Git客户端官方版具有闪电般快速的性能，并且拥有便利的分段区域、多个工作流等功能，Git客户端官方版采用了分布式版本库的作法，不需要服务器端软件，软件就可以运作版本控制，从而使得源代码的发布和交流极其方便。



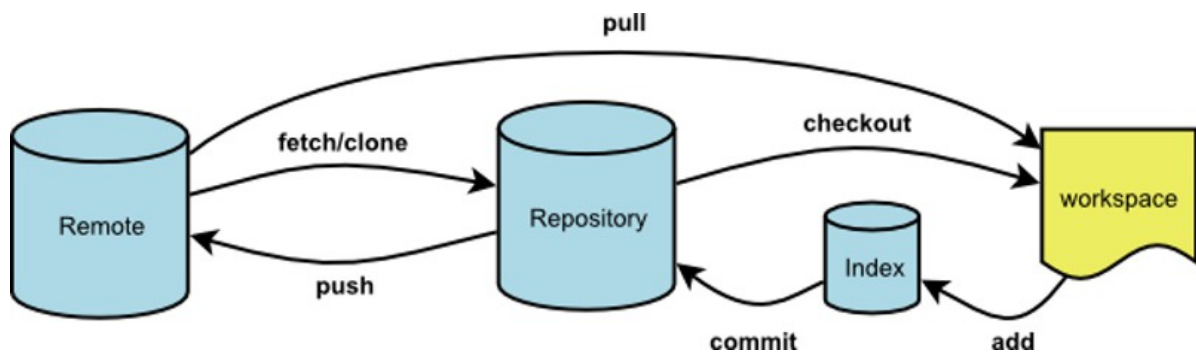
本服务器中项目全部使用git管理，项目本地仓库和远程仓库尽量同名，项目进行需要遵从git使用规范流程。

git基本介绍：<https://www.runoob.com/git/git-tutorial.html>

git使用规范流程：<http://www.ruanyifeng.com/blog/2015/08/git-use-process.html>（Github与私有服务器同原理）

git官网：<https://git-scm.com/book/zh/v2>

git简明指南：<https://www.runoob.com/manual/git-guide/>



git常用操作

- git clone
- git init
- git remote
- git fetch
- git pull
- git commit
- git push

上述常用操作包含了git本地及远程操作大部分内容，下面说明git操作基本流程。

1.新建本地仓库和远程仓库

在本地创建一个项目文件夹，进入文件夹，然后执行

```
git init
```

文一远程服务器开放ssh密码登录，可以内网ssh连接git服务器新建远程库，执行

```
# 连接远程服务器
ssh git@192.168.1.52
# 新建文件夹 所有远程托管项目均放在/home/git/repository/姓名全拼
mkdir /home/git/repository/姓名全拼/xxxx.git
# 初始化仓库
cd /home/git/repository/姓名全拼/xxxx.git
git init --bare ./xxxx.git
```

当前服务器/home/git/repository目录下已创建好每个人的工作路径（git用户权限），工作路径内需要大家自行建立项目远程仓库。

```
caishanglei fanyujie fengshanchu hanxinyan jinjianbo lipeijie lulongfei ouyangyang runoob.git
wangchenchen wangweifan yanghaijian zhangnanlong zhenglincong
chengzishuai fengrui gitManual.git houshiwei linyanjuan liyudi maxiao qianyingyang songyude
wangmingjun wangzijing yuchangjun zhangyu zhongshenyou
$ pwd
/home/git/repository
```

Github远程仓库直接在网页上创建即可。

2.克隆远程仓库

其他人的仓库可以克隆到本地进行修改，而不需要新建本地仓库，如果是Github上的远程仓库，执行

```
git clone git@github.com:xxxx/xxxx.git
```

如果是其他人Github上的仓库，建议使用fork，后续可以修改原作者源代码。

如果是文一实验室git服务器，所有远程托管项目均放在/home/git/repository，所有远程归档项目均放在/home/git/archive，执行

```
git clone git@192.168.1.52:/home/git/repository/姓名全拼/xxxx.git
```

3.添加和提交

若对项目进行更改，使用如下命令，将更改内容添加到暂存区

```
git add <filename>
```

使用如下命令将更改内容进行提交

```
git commit -m "代码提交相关信息"
```

若对其他人的仓库内容进行更改，建议开辟新的分支，使仓库所有者了解到你所做的贡献，并决定是否合并分支

```
# 开辟新的分支，并切换至新分支
git checkout -b dev
# 删除分支
git branch -d dev
```

4.推送

若不是克隆来的仓库，需将你的仓库连接到远程服务器，执行

```
git remote add <仓库名比如origin> <server>
```

Github上的在网页上复制得到，文一服务器为git@192.168.1.52:/home/git/repository/姓名全拼/xxxx.git。

之后将更改提交推送到远程仓库

```
git push origin master
```

5.更新与合并

从远程仓库拉取最新的更改，执行

```
git fetch
```

如果觉得某分支的更改可以接受，决定与当前分支合并，执行

```
git merge dev
```

遗憾的是，git会尝试进行分支合并操作，但并不是每次合并都能成功，需要手动对分支中的冲突进行消除，这一部分操作建议在编辑器上完成，多数编辑器冲突消除都比较直观，如vscode，idea等。

6.创建ssh公钥和私钥

使用对称密钥连接，可以不再需要ssh密码登录（也更安全），执行

```
ssh-keygen -C "邮箱"
```

```
# 此时windows用户C:\Users\用户名\.ssh 或者 Linux用户 /home/用户名/.ssh 下会多出两个文件  
id_rsa和id_rsa.pub, 前者是私钥, 后者是公钥
```

```
# 将该公钥导入git服务器
```

```
ssh git@192.68.1.52 'cat >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
```

7.下沙校园网连接

目前文一git服务器可通过下沙校园网连接，连接IP地址为：10.63.110.231，转发端口号：6000。校网内网连接时，由于使用非标准ssh端口（非22），部分命令不同。

```
# ssh连接
```

```
ssh -p 6000 git@10.63.110.231
```

```
# git clone
```

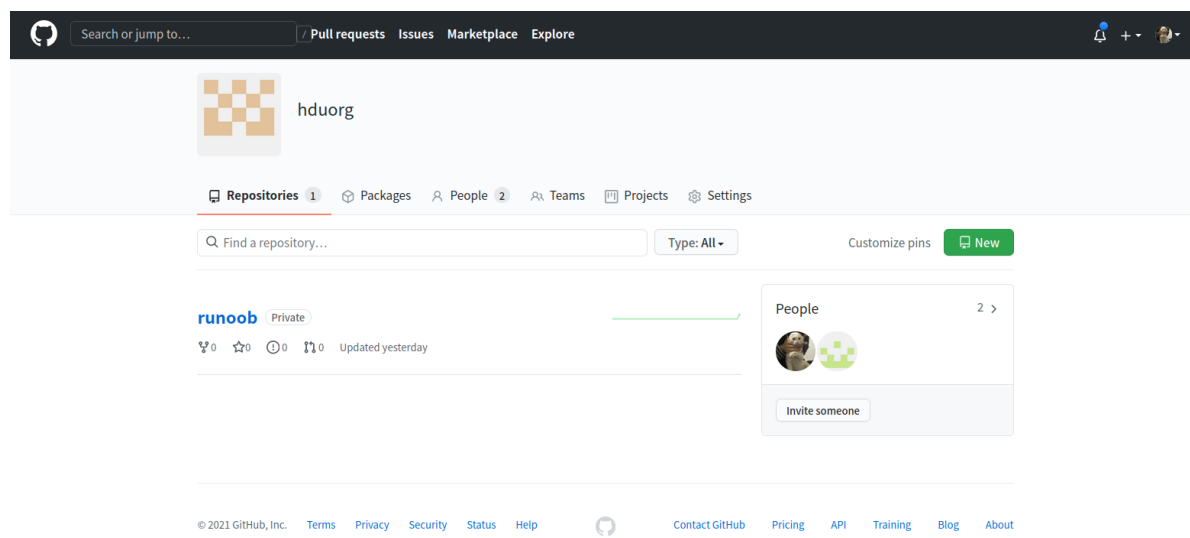
```
git clone ssh://git@10.63.110.231:6000/home/git/repository/姓名全拼/xxxx.git
```

```
# git remote
```

```
git remote add <仓库名比如origin>
```

```
ssh://git@10.63.110.231:6000/home/git/repository/姓名全拼/xxxx.git
```

Github organization



Github organization现已全面开放团队功能，无项目协作人数和私有库限制，可以在组织中建立私有库进行多人协作项目研发，其pull request功能使得协作更为便利，项目管理者能够直观看见更新的提交并决定是否合并提交。

chengTest #1

Edit Open with ▾

Merged kbbkbs merged 1 commit into `naster` from `dev` yesterday

Conversation 0 Commits 1 Checks 0 Files changed 1 +2 -0

@pythontester

python-faker commented yesterday

加一条

chengTest

3cffc4

 kbbks self-requested a review yesterday

 @ **kbkbks** merged commit **0480703** into **master** yesterday

Revert

You're all set—the `dev` branch can be safely deleted.

Delete branch

@kbw
bks

Write

Preview

H B I \equiv $\langle \rangle$ \circ \equiv \equiv \square @ \rightarrow \leftarrow

Reviewers
@k kbkbks

@k kbkbks

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

Successfully merging this pull request may close these issues.