



链滴

git 相关学习笔记

作者: [wanmisy](#)

原文链接: <https://ld246.com/article/1530170539844>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

GIT

分布式版本控制系统,跟踪并管理修改而非文件

1.配置

```
git config --global user.name "your name"
```

```
git config --global user.email "email@example.com"
```

- 文件添加到本地仓库git add (工作区添加到暂存区)

```
git add readme.txt
```

- 文件提交到本地仓库git commit (暂存区提交到版本库)

```
git commit -m "wrote a readme file"
```

- 仓库当前状态

```
git status
```

- 查看不同git diff

```
git diff readme.txt
```

- 查看提交日志git log

```
git log (--pretty=online)
```

- 回退 (未push到远程仓库, 只针对本地仓库)

```
git reset -- hard HEAD^
```

回退到上一个版本, HEAD为当前版本, 上一版本则是HEAD^,上上一版本则是HEAD^^

```
git reset --hard 3628165
```

3628165为你需要回退的版本号, 可通过git log查看, 版本号不需要写全, 只需前几位即可, 但一定唯一

```
git checkout -- [文件]
```

说明: git checkout readme.txt,将readme.txt在工作区的修改全部撤销

情况一: 还未git add,则回退到版本库一模一样

情况二: 已经git add,则回退到暂存区的状态

针对二: git reset HEAD readme.txt可以把暂存区的修改撤销掉 (unstage) , 重新放回工作区

- 查看自己的每一次命令

```
git reflog
```

- 删除

git rm

2. github建立远程仓库

- 第1步: 创建SSH Key。在用户主目录下, 看看有没有.ssh目录, 如果有, 再看看这个目录下有没有id_rsa和id_rsa.pub这两个文件, 如果已经有了, 可直接跳到下一步。如果没有, 打开Shell (Windows下打开Git Bash), 创建SSH Key:

```
$ ssh-keygen -t rsa -C "youremail@example.com"
```

- 第2步: 登陆GitHub, 打开 "Account settings", "SSH Keys" 页面

然后, 点 "Add SSH Key", 填上任意Title, 在Key文本框里粘贴id_rsa.pub文件的内容

- 第3步: 建立本地库和github的连接

新建库

```
echo "# smart4j" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/admin-wl/smart4j.git
git push -u origin master
```

已有库和github的连接

```
git remote add origin https://github.com/admin-wl/smart4j.git
git push -u origin master
```

使用命令git push -u origin master第一次推送master分支的所有内容

使用命令git push -u origin master第一次推送master分支的所有内容

- 克隆

git clone 地址

3. 分支

- 创建并切换分支 (git checkout -b [分支名])

```
git checkout -b dev
```

-b 参数表示创建并切换, 相当于以下两条命令

```
git branch dev
git checkout dev
```

- 创建分支

```
git branch [name]
```

- 切换分支

git checkout [name]

- 查看当前分支

git branch

- 合并指定分支到当前分支 (git merge [指定分支名])

git merge dev

说明：为fast forward合并，看不出曾经做过合并。

git merge --no-ff -m "test" dev

说明：合并分支时，加上--no-ff参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而fast forward合并就看不出曾经做过合并。

- 删除分支

git branch -d dev

- 查看分支合并图

git log --graph

4. 解决bug方法

git stash当前工作现场“储藏”起来，等以后恢复现场后继续工作

恢复stash

一是用git stash apply恢复，但是恢复后，stash内容并不删除，你需要用git stash drop来删除；

另一种方式是用git stash pop，恢复的同时把stash内容也删了

- 强行删除未合并的分支

git branch -D <name>

- 查看远程仓库信息

git remote

- 创建远程origin的dev分支到本地

git checkout -b dev origin/dev

- 把最新的提交从origin/dev抓下来

git pull

5. 多人合作

- 查看远程库信息，使用git remote -v;
- 本地新建的分支如果不推送到远程，对其他人就是不可见的;
- 从本地推送分支，使用git push origin branch-name，如果推送失败，先用git pull抓取远程的新交;
- 在本地创建和远程分支对应的分支，使用git checkout -b branch-name origin/branch-name，地和远程分支的名称最好一致;
- 建立本地分支和远程分支的关联，使用git branch --set-upstream branch-name origin/branch-name;
- 从远程抓取分支，使用git pull，如果有冲突，要先处理冲突。

6. 标签

- 命令git tag <name>用于新建一个标签，默认为HEAD，也可以指定一个commit id;
- git tag -a <tagname> -m "blablabla..."可以指定标签信息;
- git tag -s <tagname> -m "blablabla..."可以用PGP签名标签;
- 命令git tag可以查看所有标签。
- 命令git push origin <tagname>可以推送一个本地标签;
- 命令git push origin --tags可以推送全部未推送过的本地标签;
- 命令git tag -d <tagname>可以删除一个本地标签;
- 命令git push origin :refs/tags/ <tagname>可以删除一个远程标签。

7. 搭建git服务器

- 第一步，安装git:

```
$ sudo apt-get install git
```

- 第二步，创建一个git用户，用来运行git服务:

```
$ sudo adduser git
```

- 第三步，创建证书登录:

收集所有需要登录的用户的公钥，就是他们自己的id_rsa.pub文件，把所有公钥导入到/home/git/.ssh/authorized_keys文件里，一行一个。

- 第四步，初始化Git仓库:

先选定一个目录作为Git仓库，假定是/srv/sample.git，在/srv目录下输入命令:

```
$ sudo git init --bare sample.git
```

- 第五步，禁用shell登录

编辑/etc/passwd文件完成。找到类似下面的一行：

```
git:x1001:1001:,,,:/home/git:/bin/bash
```

改为：

```
git:x1001:1001:,,,:/home/git:/usr/bin/git-shell
```

这样，git用户可以正常通过ssh使用git，但无法登录shell，因为我们为git用户指定的git-shell每次登录就自动退出。

- 第六步，克隆远程仓库