Git学习笔记

1. Git是一个分布式版本控制系统，而CVS和SVN是集中式版本控制系统区别如下：

|  |  |  |
| --- | --- | --- |
|  | 优点 | 缺点 |
| 集中式 | 将开发资源统一管理，一致性保持很容易做到。 | 必须联网才能工作，提交/取回受网络环境影响大。使用自己的电脑开发时必须先从中央服务器获取最新版本，然后修改，修改完后再推回中央服务器。 |
| 分布式 | 每个人的电脑都是一个完整的版本库，只有需要和别人同步的时候才需要联网。安全性也高，当一个节点宕掉后其他节点还有副本。 | 无法保证所有节点的版本内容完全一致，但是可以根据情况选择github或gitlab作为中心节点同步所有人的工作。 |

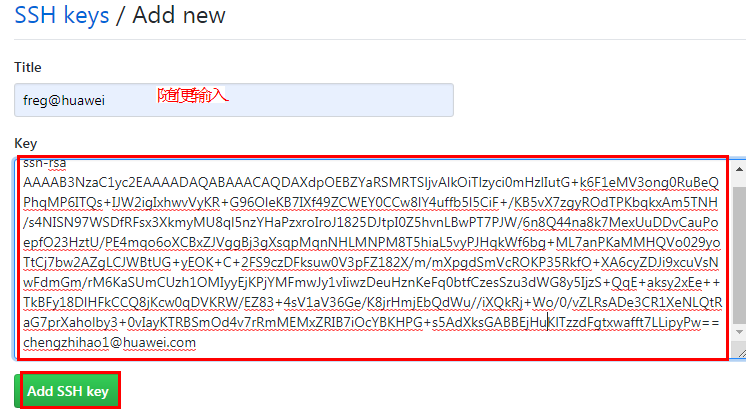
1. 安装后需要配置个人信息，例如姓名、email等，用于区分不同节点的修改提交

可以git config --global user.name "Your Name"，也可以直接去修改~/.gitconfig文件，如下：



1. 创建ssh秘钥。创建ssh秘钥主要用于和远端gitlab或github进行认证，认证之后每次提交/更新都不需要输入用户名和密码。
2. ssh-keygen -t rsa，之后会在/home/freg/.ssh生成id\_rsa.pub
3. 以github举例，选择“SSH and GPG keys”并添加新的ssh key，将id\_rsa.pub里面的内容拷贝至key中即可





1. 设置提交过滤。有些时候我们并不想把项目中的所有文件一并同步到git中，例如\*.pyc，\*.swp，\*~等临时文件，需要在git仓库下的.gitignore设置



1. 拉取和更新

使用git clone remote\_url拉取远程仓库

更新本地的remote branch，使得本地的remote branch和远程服务器保持一致，再同步本地分支和远程分支，使得两者内容保持一致（包括本地修改和远程修改内容）

git fetch origin

git rebase origin/master

1. 推送

完成本地作业后，需要将本地修改推送到远程服务器，在推送之前需要更新。推送操作：

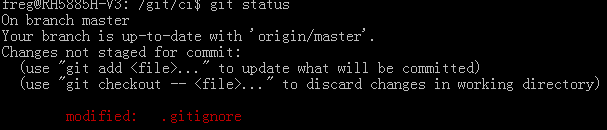
git add FILE // 修改的文件或文件夹

git status // 可省略，主要用于查看本次提交做的修改

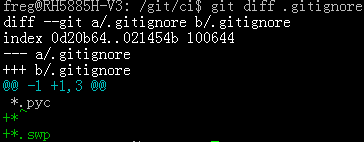
git commit –m ‘modify operation’ // 本次推送的主要修改内容，给团队中别的人看

1. 如何处理本地仓库和远程仓库的不一致

使用git status查看本地和远程仓库的差异，发现有一个叫.gitignore的文件被修改却没有同步。



可以通过git diff详细查看修改内容（修改后文件和远程仓库对应文件的具体差异）



这时候可以有两个选择，一个是add/commit将文件同步到远程仓库，另一个则是使用git checkout放弃本地修改来同步远程仓库。使用git checkout后发现，本地修改被放弃，已完成和远程仓库同步。



1. Git版本控制
2. 回退至历史某一版本

git log可以查到所有历史提交记录（根据commit内容可以对每个版本有个大致的了解，所以commit内容很重要，不要乱填）

