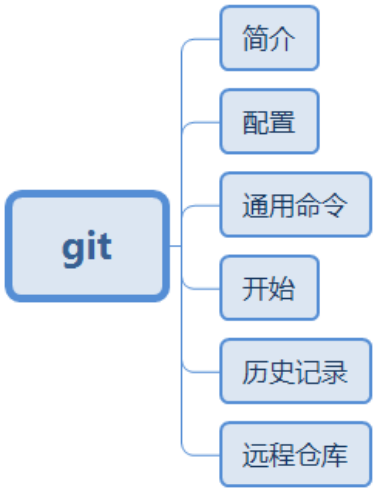
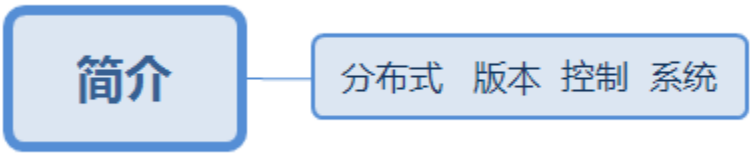
git

对stage的理解:工作区和暂存区是一个公开的工作台，任何分支都会用到，并能看到工作台上最新的内容，只要在工作区、暂存区的改动未能够提

集中式版本控制系统



1. 简介



1.1. 分布式 版本 控制 系统

参见: 集中式版本控制系统 (联网 安全)

1.1.1. git(linux)

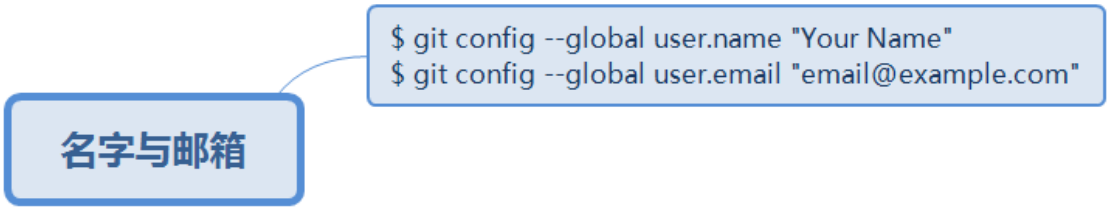
1.1.2. bitkeeper(Andrew)

1.1.3. ClearCase(IBM)

1.1.4. VSS(微软)

2. 配置

2.1. 名字与邮箱

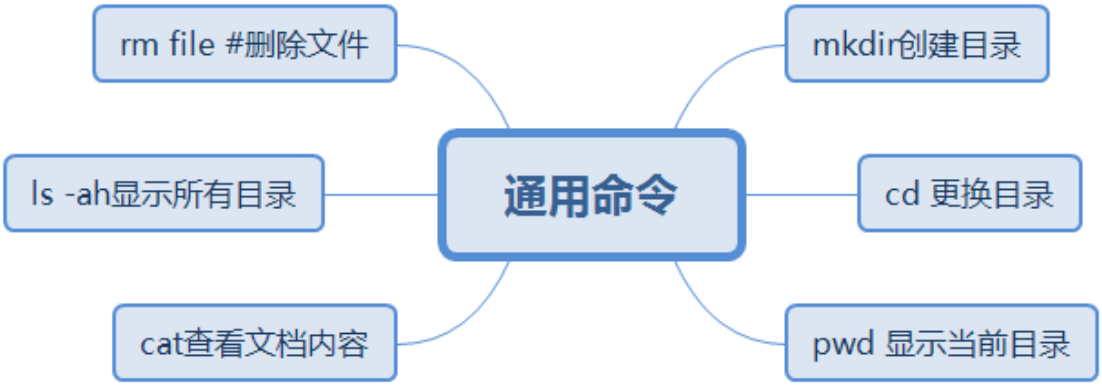


2.1.1. $ git config --global user.name "Your Name"

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

2.2. git config --global color.ui true#让Git显示颜色

3. 通用命令



3.1. mkdir创建目录

3.2. cd 更换目录

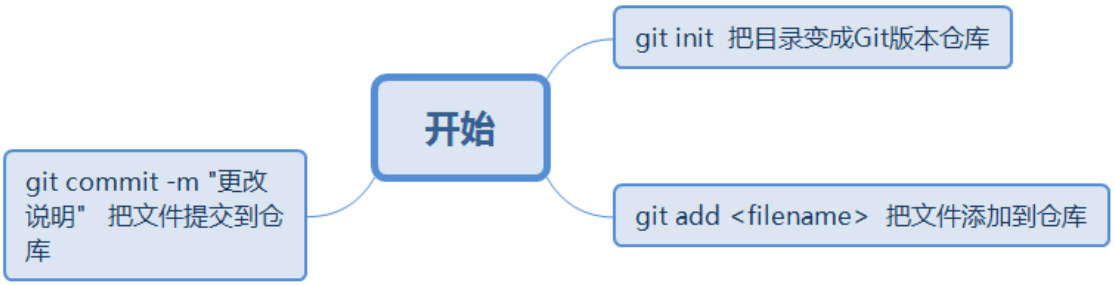
3.3. pwd 显示当前目录

3.4. cat查看文档内容

3.5. ls -ah显示所有目录

3.6. rm file #删除文件

4. 开始

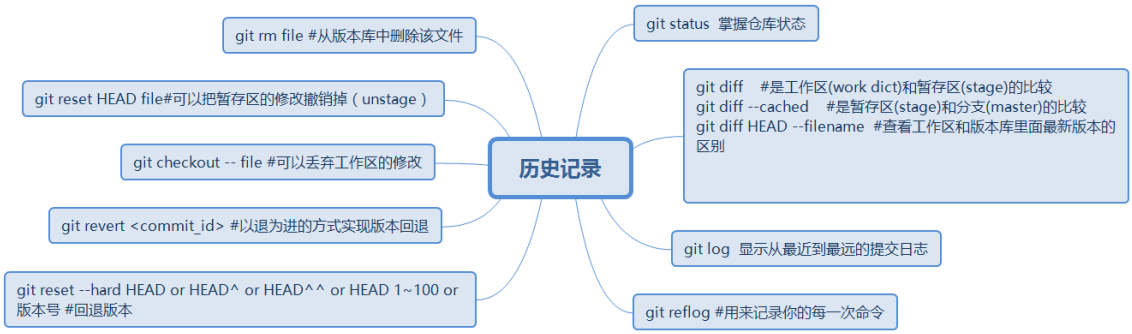


4.1. git init 把目录变成Git版本仓库

4.2. git add <filename> 把文件添加到仓库

4.3. git commit -m "更改说明" 把文件提交到仓库

5. 历史记录



5.1. git status 掌握仓库状态

5.2. git diff #是工作区(work dict)和暂存区(stage)的比较

git diff --cached #是暂存区(stage)和分支(master)的比较

git diff HEAD --filename #查看工作区和版本库里面最新版本的区别

5.3. git log 显示从最近到最远的提交日志

5.3.1. --pretty=oneline 显示一行信息

5.4. git reflog #用来记录你的每一次命令

5.5. git reset --hard HEAD or HEAD^ or HEAD^^ or HEAD 1~100 or 版本号 #回退版本

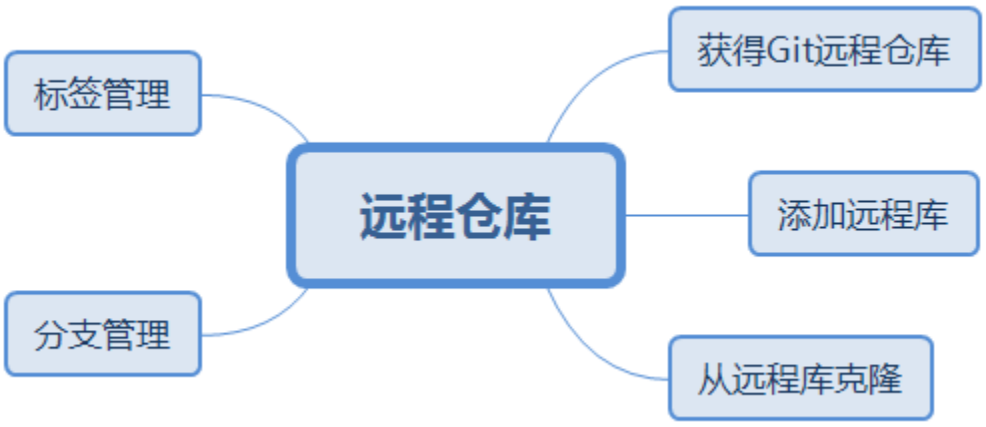
5.6. git revert <commit_id> #以退为进的方式实现版本回退

5.7. git checkout -- file #可以丢弃工作区的修改

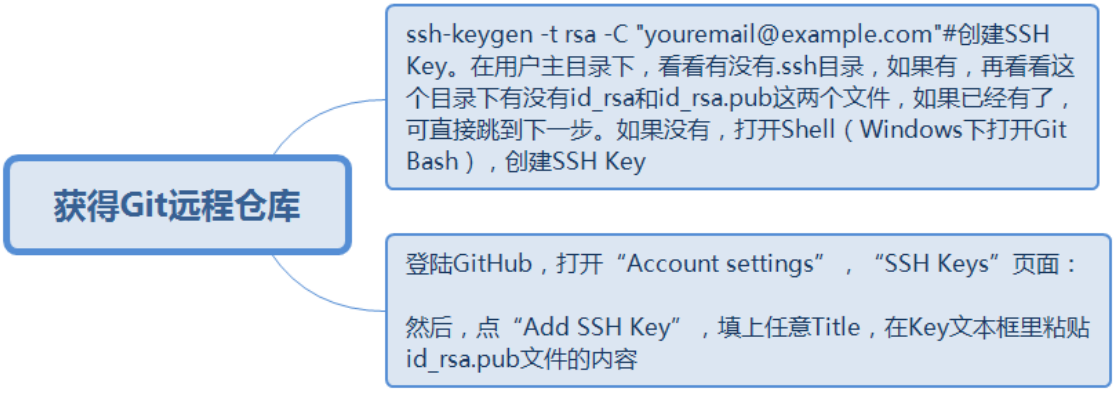
5.8. git reset HEAD file#可以把暂存区的修改撤销掉（unstage）

5.9. git rm file #从版本库中删除该文件

6. 远程仓库



6.1. 获得Git远程仓库

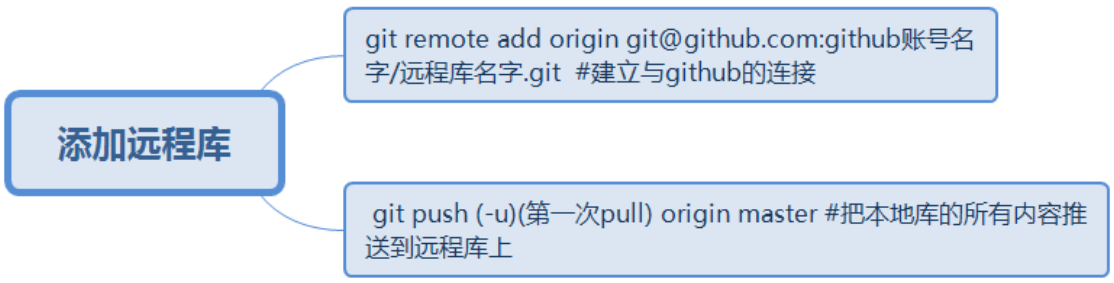


6.1.1. ssh-keygen -t rsa -C "youremail@example.com"#创建SSH Key。在用户主目录下，看看有没有.ssh目录，如果有，再看看这个目录下有没有id_rsa和id_rsa.pub这两个文件，如果已经有了，可直接跳到下一步。如果没有，打开Shell（Windows下打开Git Bash），创建SSH Key

6.1.2. 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

6.2. 添加远程库

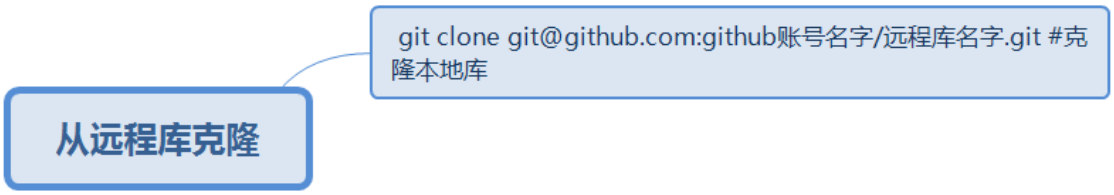


6.2.1. git remote add origin git@github.com:github账号名字/远程库名字.git #建立与github的连接

6.2.2. git push (-u)(第一次pull) origin master

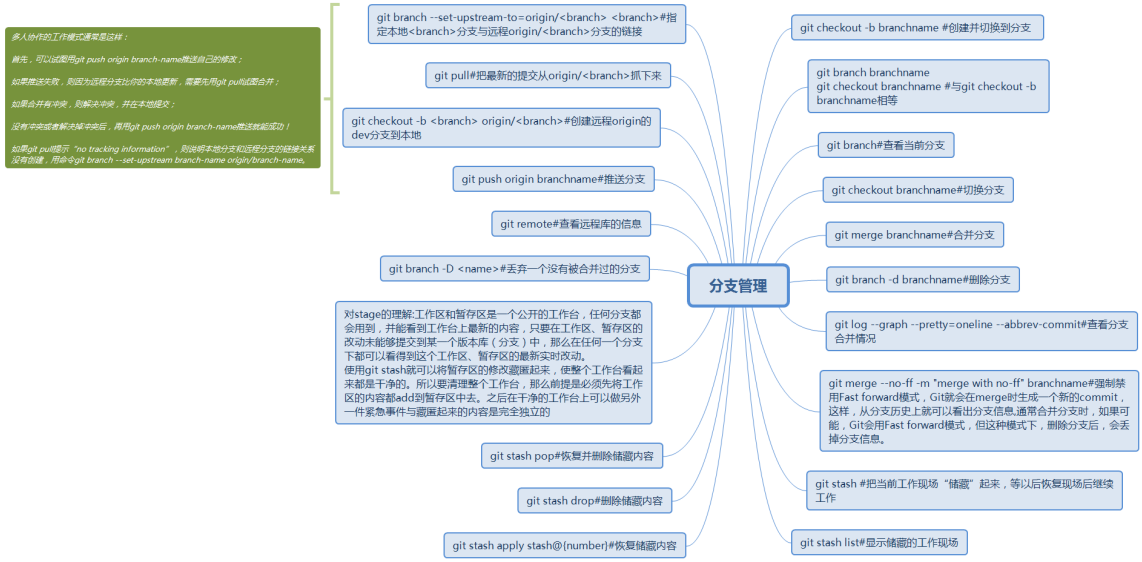
#把本地库的所有内容推送到远程库上

6.3. 从远程库克隆



6.3.1. git clone git@github.com:github账号名字/远程库名字.git #克隆本地库

6.4. 分支管理



6.4.1. git checkout -b branchname #创建并切换到分支

6.4.2. git branch branchname

git checkout branchname #与git checkout -b branchname相等

6.4.3. git branch#查看当前分支

6.4.4. git checkout branchname#切换分支

6.4.5. git merge branchname#合并分支

6.4.6. git branch -d branchname#删除分支

6.4.7. git log --graph --pretty=oneline --abbrev-commit#查看分支合并情况

6.4.8. git merge --no-ff -m "merge with no-ff" branchname#强制禁用Fast forward模式，Git就会在merge时生成一个新的commit，这样，从分支历史上就可以看出分支信息,通常合并分支时，如果可能，Git会用Fast forward模式，但这种模式下，删除分支后，会丢掉分支信息。

6.4.9. git stash #把当前工作现场“储藏”起来，等以后恢复现场后继续工作

6.4.10. git stash list#显示储藏的工作现场

6.4.11. git stash apply stash@{number}#恢复储藏内容

6.4.12. git stash drop#删除储藏内容

6.4.13. git stash pop#恢复并删除储藏内容

6.4.14. 对stage的理解:工作区和暂存区是一个公开的工作台，任何分支都会用到，并能看到工作台上最新的内容，只要在工作区、暂存区的改动未能够提交到某一个版本库（分支）中，那么在任何一个分支下都可以看得到这个工作区、暂存区的最新实时改动。
使用git
stash就可以将暂存区的修改藏匿起来，使整个工作台看起来都是干净的。所以要清理整个工作台，那么前提是必须先将工作区的内容都add到暂存区中去。之后在干净的工作台上可以做另外一件紧急事件与藏匿起来的内容是完全独立的

6.4.15. git branch -D <name>#丢弃一个没有被合并过的分支

6.4.16. git remote#查看远程库的信息

-v显示更详细的信息

6.4.17. git push origin branchname#推送分支

6.4.18. git checkout -b <branch>

origin/<branch>#创建远程origin的dev分支到本地

6.4.19. git pull#把最新的提交从origin/<branch>抓下来

6.4.20. git branch --set-upstream-to=origin/<branch>

<branch>#指定本地<branch>分支与远程origin/<branch>分支的链接

多人协作的工作模式通常是这样：

首先，可以试图用git push origin branch-name推送自己的修改；

如果推送失败，则因为远程分支比你的本地更新，需要先用git pull试图合并；

如果合并有冲突，则解决冲突，并在本地提交；

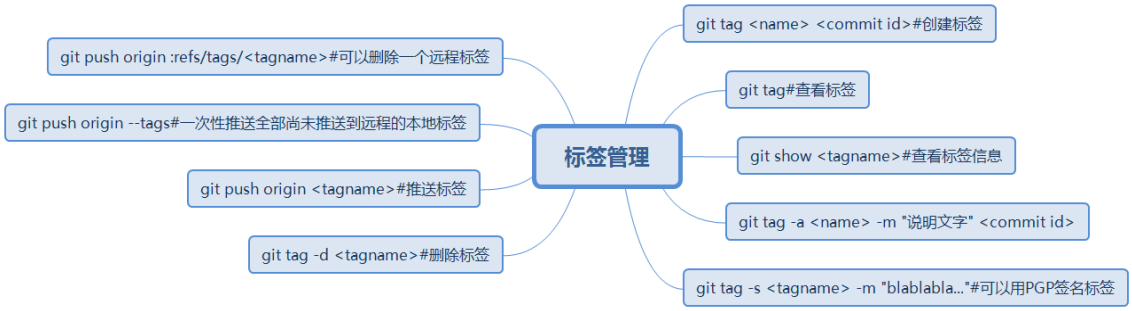
没有冲突或者解决掉冲突后，再用git push origin branch-name推送就能成功！

如果git pull提示“no tracking

information”，则说明本地分支和远程分支的链接关系没有创建，用命令git

branch --set-upstream branch-name origin/branch-name。(git push origin branchname#推送分支, git checkout -b <branch> origin/<branch>#创建远程origin的dev分支到本地, git pull#把最新的提交从origin/<branch>抓下来, git branch --set-upstream-to=origin/<branch> <branch>#指定本地<branch>分支与远程origin/<branch>分支的链接)

6.5. 标签管理



6.5.1. git tag <name> <commit id>#创建标签

6.5.2. git tag#查看标签

6.5.3. git show <tagname>#查看标签信息

6.5.4. git tag -a <name> -m "说明文字" <commit id>

6.5.5. git tag -s <tagname> -m "blablabla..."#可以用PGP签名标签

6.5.6. git tag -d <tagname>#删除标签

6.5.7. git push origin <tagname>#推送标签

6.5.8. git push origin --tags#一次性推送全部尚未推送到远程的本地标签

6.5.9. git push origin :refs/tags/<tagname>#可以删除一个远程标签

集中式版本控制系统

参见: 分布式 版本 控制 系统 (联网 安全)

CVS

SVN