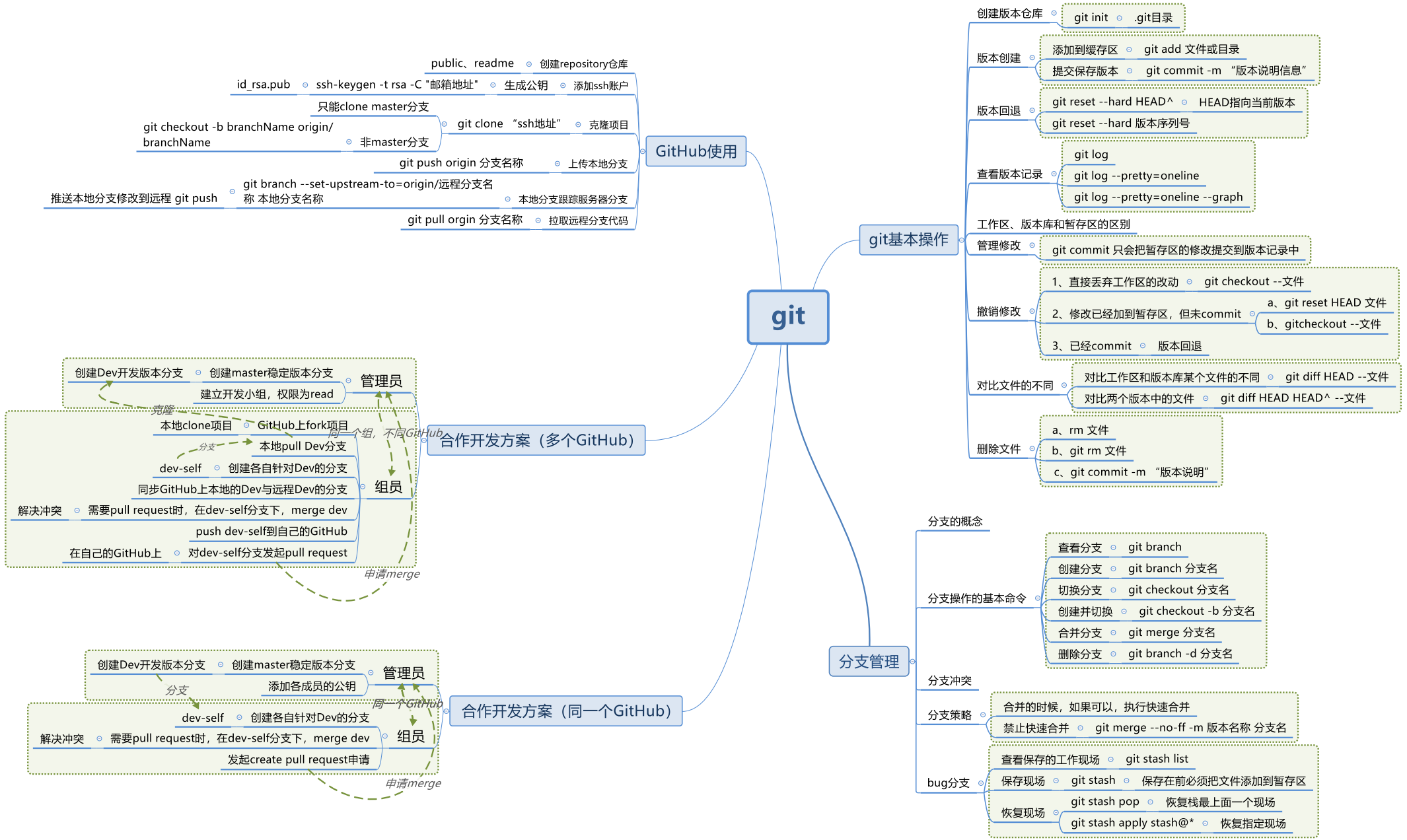
git
GitHub使用
public、readme
创建repository仓库
id_rsa.pub
ssh-keygen -t rsa -C "邮箱地址"
生成公钥
添加ssh账户
只能clone master分支
git checkout -b branchName origin/branchName
非master分支
git clone “ssh地址”
克隆项目
git push origin 分支名称
上传本地分支
推送本地分支修改到远程 git push
git branch --set-upstream-to=origin/远程分支名称 本地分支名称
本地分支跟踪服务器分支
git pull orgin 分支名称
拉取远程分支代码
git基本操作
创建版本仓库
git init
.git目录
版本创建
添加到缓存区
git add 文件或目录
提交保存版本
git commit -m “版本说明信息”
版本回退
git reset --hard HEAD^
HEAD指向当前版本
git reset --hard 版本序列号
查看版本记录
git log
git log --pretty=oneline
git log --pretty=oneline --graph
工作区、版本库和暂存区的区别
管理修改
git commit 只会把暂存区的修改提交到版本记录中
撤销修改
1、直接丢弃工作区的改动
git checkout --文件
2、修改已经加到暂存区，但未commit
a、git reset HEAD 文件
b、gitcheckout --文件
3、已经commit
版本回退
对比文件的不同
对比工作区和版本库某个文件的不同
git diff HEAD --文件
对比两个版本中的文件
git diff HEAD HEAD^ --文件
删除文件
a、rm 文件
b、git rm 文件
c、git commit -m “版本说明”
分支管理
分支的概念
分支操作的基本命令
查看分支
git branch
创建分支
git branch 分支名
切换分支
git checkout 分支名
创建并切换
git checkout -b 分支名
合并分支
git merge 分支名
删除分支
git branch -d 分支名
分支冲突
分支策略
合并的时候，如果可以，执行快速合并
禁止快速合并
git merge --no-ff -m 版本名称 分支名
bug分支
查看保存的工作现场
git stash list
保存现场
git stash
保存在前必须把文件添加到暂存区
恢复现场
git stash pop
恢复栈最上面一个现场
git stash apply stash@*
恢复指定现场
合作开发方案（多个GitHub）
创建Dev开发版本分支
创建master稳定版本分支
管理员
建立开发小组，权限为read
克隆
本地clone项目
GitHub上fork项目
同一个组，不同GitHub
分支
本地pull Dev分支
dev-self
创建各自针对Dev的分支
组员
同步GitHub上本地的Dev与远程Dev的分支
解决冲突
需要pull request时，在dev-self分支下，merge dev
push dev-self到自己的GitHub
在自己的GitHub上
对dev-self分支发起pull request
申请merge
合作开发方案（同一个GitHub）
创建Dev开发版本分支
创建master稳定版本分支
管理员
分支
添加各成员的公钥
同一个GitHub
dev-self
创建各自针对Dev的分支
解决冲突
需要pull request时，在dev-self分支下，merge dev
组员
发起create pull request申请
申请merge