Git版本控制

Git是一个分布式的版本控制工具。所谓版本控制(Revision Control)是工程维护的标准方法，通过该方法能追踪工程从诞生一直到定案的整个过程。所谓分布式是工程代码通常会分布在本地和远程的不同仓库节点。接下来，本文档主要围绕本地仓库和远程仓库的管理展开，本地仓库如何管理？远程仓库如何管理？本地仓库如何与远程仓库交互？

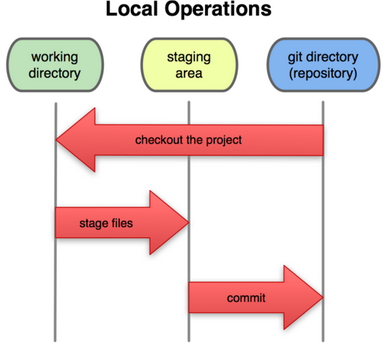
# 本地仓库管理

本地开发基本流程可以概括为：修改文件 => 加入暂存区 => 提交 ，在这一过程中本地仓库存在三种状态：

1. 已提交(committed)：表示文档已被安全保存在本地repository中；
2. 已修改(modified)：表示修改某些文档，但还没有提交；
3. 已暂存(staged)：表示已修改文档已放在下次要提交的清单中；

这三种状态分别对应不同的工作区：

1. Working directory：工作区，所有的修改、变动都发生在这个阶段；
2. Staging area：暂存区，提交文档前供确认及检查；
3. Repository：本地仓库，存储更新后的提交；



常用本地仓库的管理大致如下：

1. 初始化本地repository

git init

1. 复制服务器端repository

git clone <resource\_url>

1. 查看repository状态

git status

1. 添加修改到暂存区

git add <modified\_file> /\*添加文件\*/

git add . /\*添加目录下所有修改\*/

1. 提交修改到仓库

git commit /\*with comments\*/

git commit –m “comments information”

1. 补充修改到上一次提交

git commit --amend

1. 查看提交记录

git log

1. 查看分支

git branch

1. 创建分支

git branch <branch\_name>

1. 分支切换

git checkout <branch\_name>

1. 更新主分支

git rebase

1. 分支合并

git merge

1. 查看分支差别

git diff <branch\_1> <branch\_2>

1. 取消上一次操作

git reset HEAD <file> /\*移除暂存区文档\*/

git reset HEAD^ --hard /\*移除上一次提交\*/

git reset HEAD^ --soft /\*移除上一次提交，但保留文件修改\*/

# 远程仓库

远程仓库通常由git管理员管理，相对于git管理员而言就是本地仓库，操作与上一节内容相同，这里不再赘述。

# 本地与远程仓库交互

Git可以使用四种协议来传输资料：本地协议、http协议、ssh协议以及git协议。目前常使用的协议为ssh协议，本地计算机在安装git后，需要配置git环境并将配置过程中生成的公钥上传到服务器端。

1. Git环境配置

git config配置信息会存储在系统中三个不同位置：--system配置存放/etc/gitconfig文件；--global配置存放在~/.gitconfig或~/.config/git/config文件；当前仓库配置存放在.git/config文件中。

git config –global user.name “<username>” /\*配置用户名\*/

git config –global user.email <user email assress> /\*配置用户邮箱\*/

git config list /\*查看配置信息\*/

1. 生成ssh公钥

ssh-keygen

生成的密钥会存放在~/.ssh目录下，进入该目录可以看到一对id\_dsa或id\_rsa命名的文件，其中一个带有.pub扩展名为公钥，另一个为私钥。(注：在生成密钥时会要求输入两次密钥口令，如果不想再使用密钥时输入口令可以留空。)

1. 上传公钥给git服务器管理员

管理员在得到公钥后腰配置服务器的ssh访问，将公钥内容添加至git的.ssh目录下authorized\_keys文件结尾。

到此，本地仓库已于服务器端建立联系。接下来介绍一下本地与远程仓库的交互。

1. 查看远程仓库

git remote [-v] /\*-v 查看对应url\*/

1. 添加远程仓库

git remote –add <shortname> <url>

1. 从远程仓库抓取或拉取

git fetch [remote name] /\*将数据拉到本地但不会自动合并\*/

1. 提交本地修改到服务器

git push <remote name> <branch name>

1. 远程仓库的移除与重命名

git remote rename <origin name> <new name>

git remote rm <name>