

1、git 用来进行版本控制，是一个版本控制工具

2、git config -l 查看所有的配置。git config --global --list 查看当前用户的配置。git config --system --list 查看系统配置

3、必须要配置用户信息

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

```
git config --global user.email "1964316456@qq.com"
```

4、git基本理论

workspace：工作区，存放代码的地方。其实就是一个目录

index/stage：缓存区，用于临时存放改动，其实就是一个文件，保存即将保存到文件的列表信息

repository：仓库区，也就是本地仓库，安全存放数据的位置，这里有提交的所有版本的信息，其中，HEAD指向最新放入仓库的版本

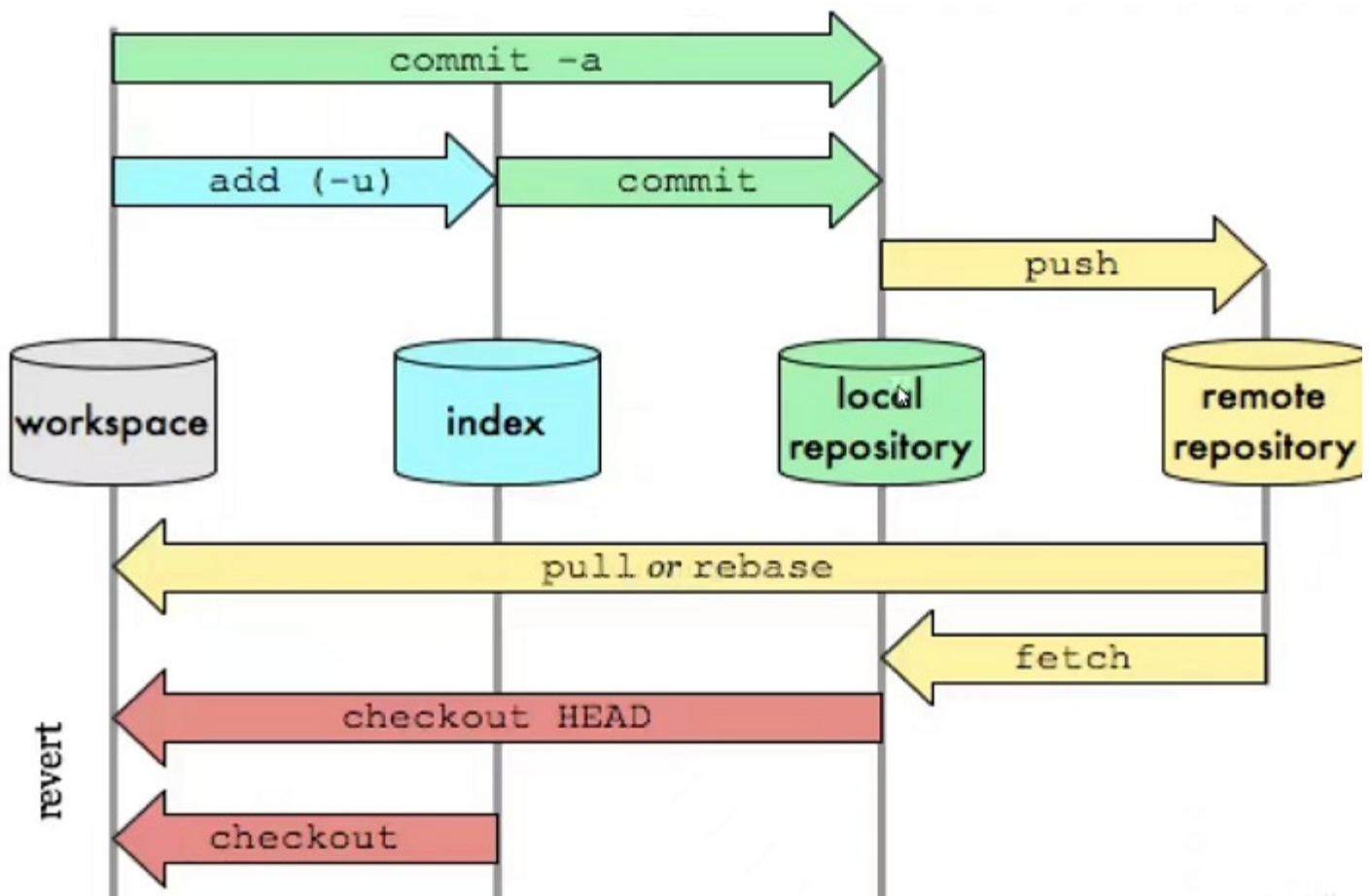
remote：远程仓库，也就是服务器

平时较多的操作工作区和远程仓库，缓存区和本地仓库一般较少操作

5、git的工作流程一般是这样的：

- 在工作目录中添加，修改文件：
- 将需要进行版本管理的文件放入暂存区域：
- 将暂存区的文件提交到本地仓库

因此，git管理的文件有三种状态，已修改 (modified) 已暂存 (staged) 已提交 (committed)



6、git项目搭建

创建本地仓库的两种方式：一种是创建全新的仓库。命令是git init。另一种是克隆远程的仓库，如果是克隆远程仓库，其命令是git clone [url]

7、文件四种状态

版本控制就是对文件的版本进行控制, 要对文件进行修改或者是提交等工作。首先要知道文件目前处于什么状态, 不然可能会是交现在还不想提交的文件, 或者要是交的文件没提交上。

- untracked 未跟踪, 该文件在文件夹中, 也就是在工作区中, 但是还没有提交到本地仓库中, 不参与版本控制, 需要通过git add 变成staged状态
- unmodify 文件已经入库, 未修改, 即本地仓库的文件内容和工作区文件夹中的完全一致, 这种类型的文件有两种去出, 如果它被修改, 而变成modified。或者使用git rm 移除本地仓库, 从而变成untracked状态
- modified 文件已修改, 这个文件也有两种去出, 一种是通过git add 进入暂存区, 变成staged状态。一种是通过git checkout 则丢弃修改过, 返回到unmodify状态, 这个git checkout 即从本地仓库中去出文件, 覆盖当前修改
- staged 暂存状态, 执行git commit同步到库中, 这时库中的文件和工作区文件变成一致, 文件为unmodify状态, 执行git reset head filename 取消暂存, 文件变成modified

使用git status 查看文件当前的状态。

8、git commit -m "注释" 将暂存区的文件添加到本地仓库。

9、有些工作区的文件并不是所有文件都需要提交到仓库中, 因此, 有了一个文件.gitignore文件, 该文件可以配置不需要添加到仓库的文件

10、设置本机绑定SSH公钥, 实现免密码登录 (由于我们工作的是本地仓库, 需要与远程到远程仓库中, 每一次我们并不想都进行登录操作)

安装了git 的电脑, 在电脑中会有一个.ssh文件, 在git 命令行中, 输入ssh -keygen -t rsa 生成一个公钥然后一路空格。然后将公钥信息添加到对应账户中。

11、git branch 查看本地的分支, git branch -r 查看远程的分支

12、git branch dev 新建dev这个分支

13、切换到另一个分支, git checkout -b 分支名

14、撤销更改的两种方式, 一种是使用git reset master^ (这里是回到哪一步), 但是这个命令只能更改本地代码库, 不能对远程库造成影响。另外一个命令是git revert master^ 可以撤销更改并分享给别人。

15、建立远程链接

git remote add 远程仓库名称 远程仓库地址。如此就建立起了连接

git push 远程仓库名称 远程仓库的分支名称 master or main

16、如果遇到更新被拒绝的情况, 可以先把远程仓库的内容先给拉下来, 使用git pull origin(给远程仓库起的名字) master(从远程仓库的哪个分支拉) --allow-unrelated-histories

17、OpenSSL SSL_read: Connection was aborted, errno 10053报错

解决方案链接: https://blog.csdn.net/m0_57135756/article/details/124739591

18、Failed to connect to github.com port 443: Timed out

两行命令:

```
git config --global --unset http.proxy  
git config --global --unset https.proxy
```