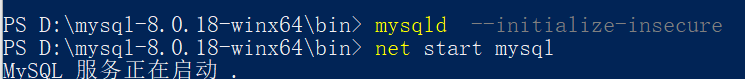
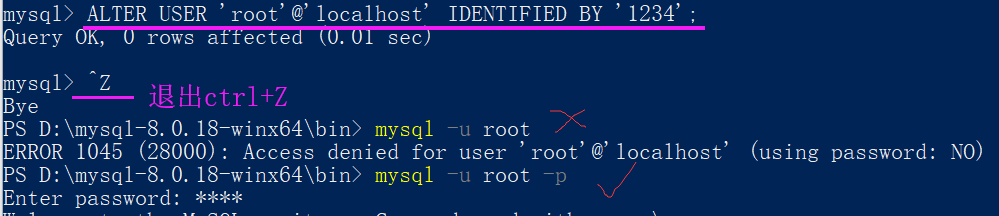
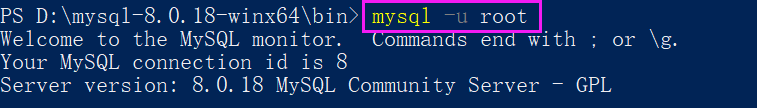
Mysql免安装版本安装

1.注意my.ini的跳过权限，不注释。不然会出现无法启动。

2.注意要进入\bin目录，进行初始化，或者安全模式初始化。注意配置my.ini后，初始化之前，删掉目录下的data文件。

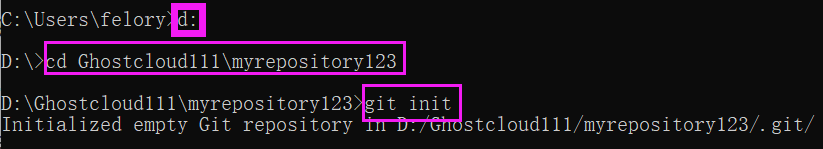


3．以root登录不需要密码，修改root的密码。



shadowsocks包括两种代理模式：**全局**（访问所有网站都通过Shadowsocks），**PAC**（通过PAC文件里的规则列表控制哪些流量走SS，哪些不走（比如国内流量不走）在Windows版本的SSR客户端，这个功能已经可以被“代理规则”（Proxy Rule）设置代替（除非使用gfwlist），因此在Windows版本的SSR客户端一般不建议用这一模式。）**直连模式**（Disable System Proxy）：基本可以理解为关掉Shadowsocks。

创建新文件夹

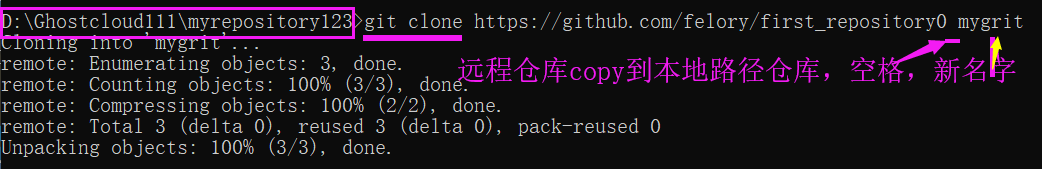


【windows cmd先进入D:再CD 文件夹名字】【cd.>a.txt】【md d:\a\b\test】【删除文夹下所有文件夹，/s以询问是否删除rd /s test】

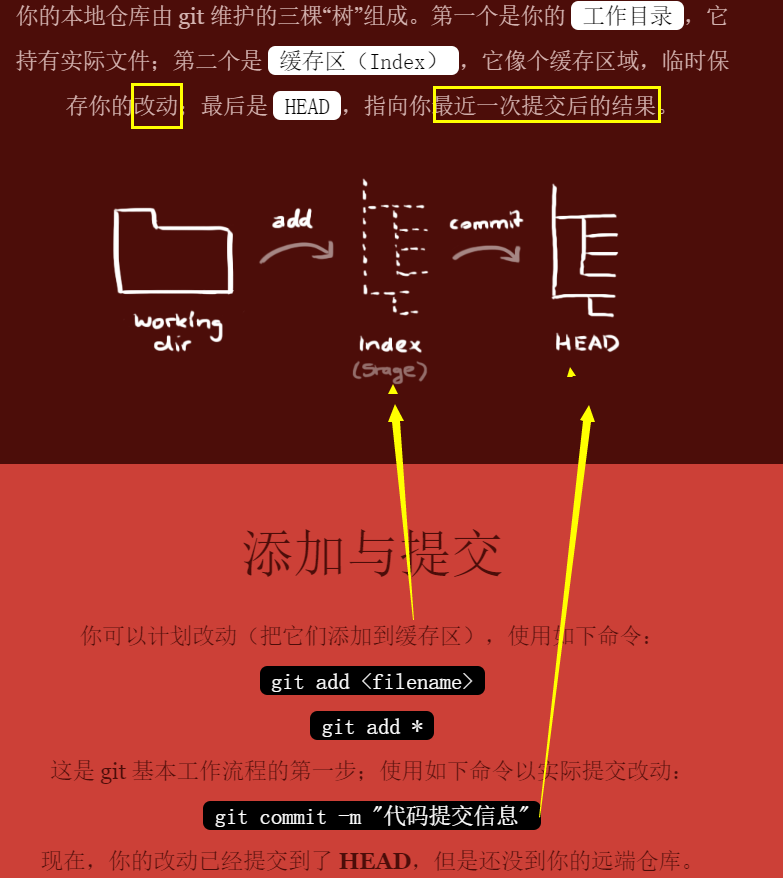
git init (内部会生成.git文件)

克隆本地仓库到新仓库 git/clone /path

克隆远程仓库到新仓库 git clone felory@github.com:/ first\_repository0.git



Git 支持许多数据传输协议。之前的例子使用的是 git:// 协议【$ git clone git://github.com/felory/myfirstgrit.git mygrit】，不过你也可以用 http(s):// 或者 user@server:/path.git 表示的 SSH 传输协议。【git@github.com:felory/learnrep1.git】

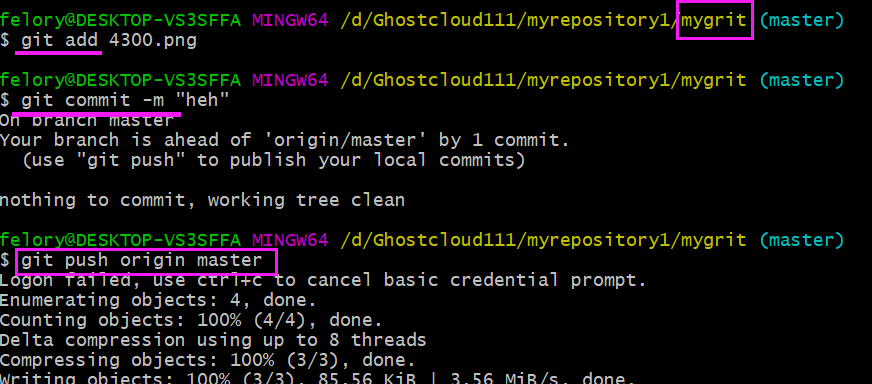


在本地仓库中(克隆下来的mygrit中)添加文件4300.png

git add 4300.png 进行追踪【注，进入mygrit中】

git commit –m “describe” 在本地仓库提交修改、描述 【双引号】

git push origin master 上传新修改的到远程仓库 【需要个人验证】



远程的空仓库，可以克隆，或者与本地关联，从本地push上去。

1.在本地目录下关联远程repository ：

git remote add origin git@github.com:felory/learnrep1.git 【git@github.com:git\_username/repository\_name.git】

2.取消本地目录下关联的远程库：

git remote remove origin

创建一个叫做“feature\_x”的分支，并切换过去：

git checkout **-b** feature\_x 相当于【$ git branch iss53 创建

$ git checkout iss53切换过去】

切换回主分支：

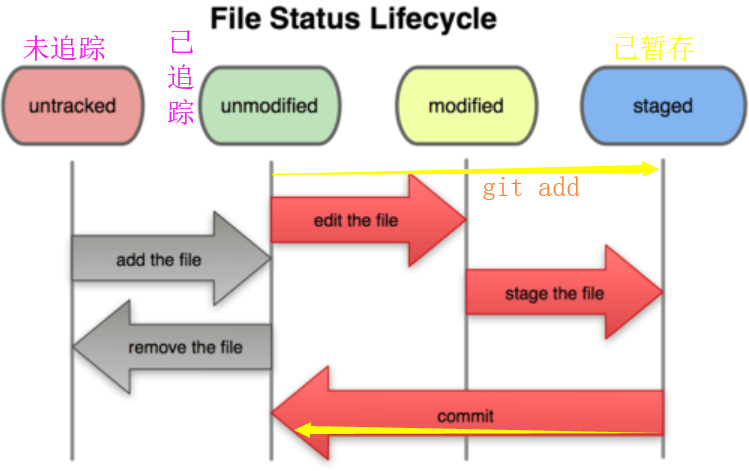
git checkout master

再把新建的分支删掉：

git branch **-d** feature\_x

除非你将分支推送到远端仓库，不然该分支就是不为他人所见的：

git push origin <branch>



ps:

工作目录下的文档只有两种状态：已跟踪---》（未更改、已修改、放入暂存区）

git status

查看状态，若是nothing to commit就是都已被跟踪，且未修改。

若有“Changes not staged for commit”则是已修改，但未放到暂存区。要暂存这次更新，就需要运行git add，用它开始跟踪新文件，或者把已跟踪的放到暂存区，或者合并时把有冲突的文件标记为已解决。

若是Changes to be committed:则文件已暂存，需commit来留存到历史记录中。

 Git 只不过暂存了你运行 git add 命令时的版本，如果现在提交，那么提交的那个历史版本。

git log 查看提交历史

gitk 可视化查看log

重新提交

git commit –m “” 提交到历史记录

git add file\_forgotten 新追踪，到暂存区

git commit –amend 修正第一次提交的内容，重新提交