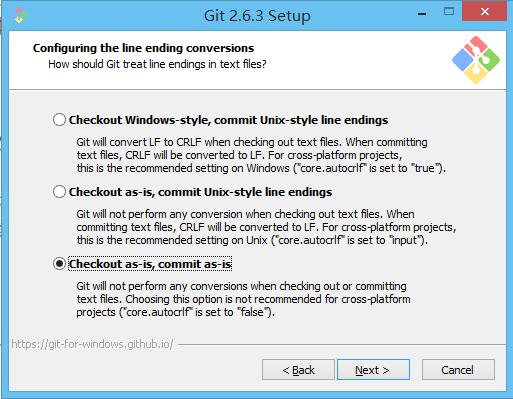
**安装**   
安装的过程没有什么好说的，一般是开始安装后，一路的点击“下一步”。由于windows平台的换行符（CRLF）和Linux(\*nix)平台的换行符（LF）不同，那么在windows下开发其它平台软件的朋友有一个地方要注意（见下图)：   


在这里最好选“Checkout as-is, commit as-is”这个选项，这样，Git就不会修改你代码的换行符风格。

配置Git

在Linux下和Windows下配置Git的方法差不多，只是在Linux下，可以在命令行里直接使用git config进行配置, 而在Windows下则要先打开“Git Bash”，进入msysGit命令行界面，再用git config命令进行相应的配置操作。   
好了，前面安装好了Git，现在我们开始配置   
1）第一个需要配置的就是用户的用户名和email，因为这些内容会出现在你的每一个提交（commit）里面的，

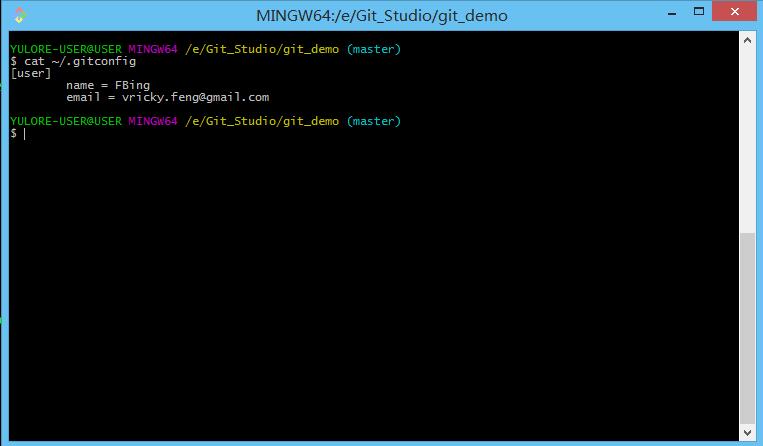
$ git config --global user.name "your name"

$ git config --global user.email "your\_email"

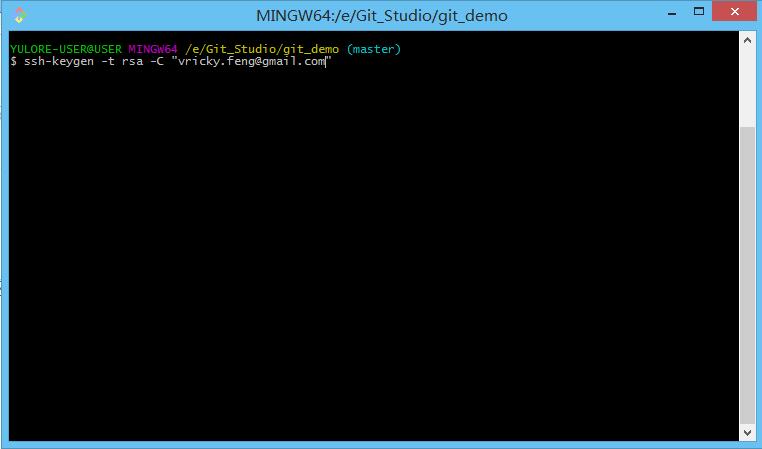
Git的配置信息分为全局和项目两种，上面命令中带了“–global”参数，这就意味是在进行全局配置，它会影响本机上的每个一个Git项目。   
Git 可以为每个项目设定不同的配置信息，在命令行环境，进入Git项目所在目录，执行下面的命令：

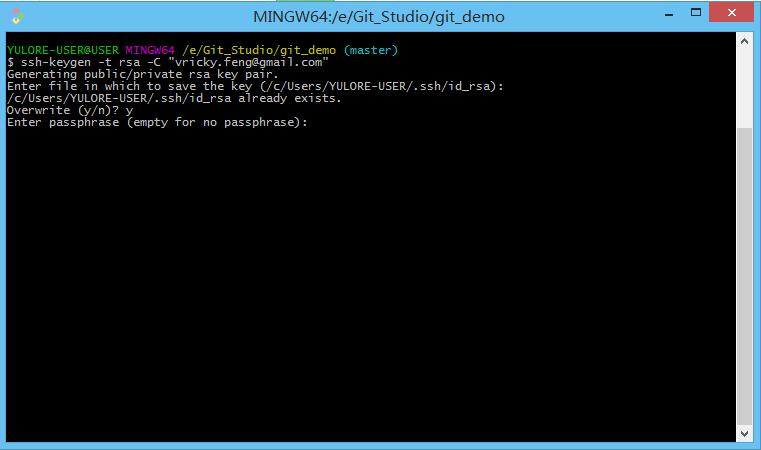
$ git config user.name "your name"

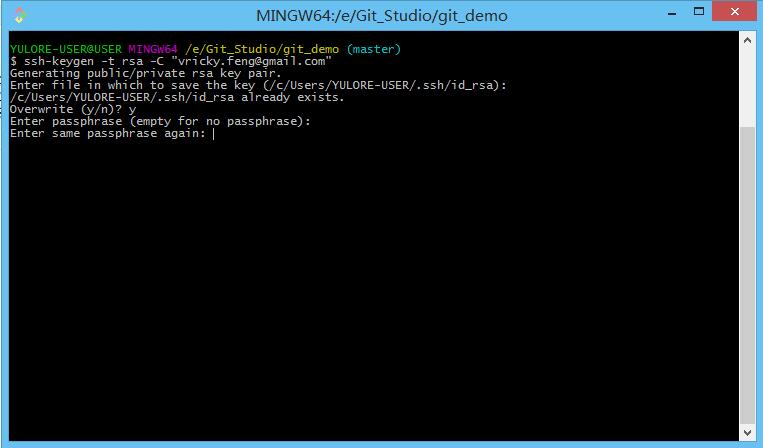
$ git config user.email "your\_email"

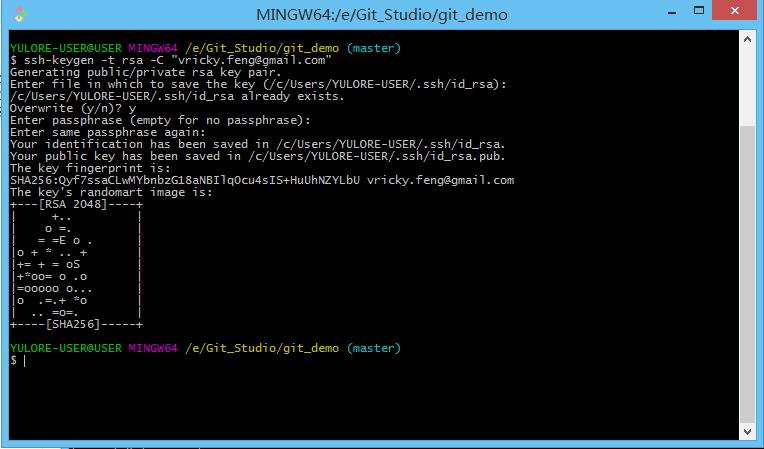
Git的设计哲学和Linux（\*nix）一样，尽量的使用“文本化”（Textuality）；它里面尽量用文本化的形式存储信息，对于配置信息也更是如此，用户的这些配置信息全部是存储在文本文件中。Git的全局配置文件是存放在”~/.gitconfig”（用户目录下的.gitconfig）文件中：   
我们用cat、head命令查看全局配置信息文件   


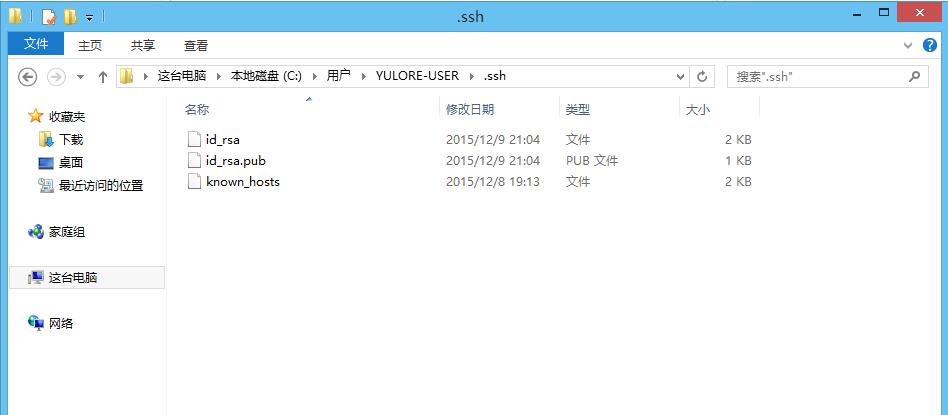
如果对于Git熟悉后，可以直修改”~/.gitconfig”,”.git/config”这两个文件进行配置。

2）在本地创建ssh key   
$ ssh-keygen -t rsa -C “your\_email”   
将后面的your\_email改为你的邮箱，也是你在github上注册的那个邮箱：   


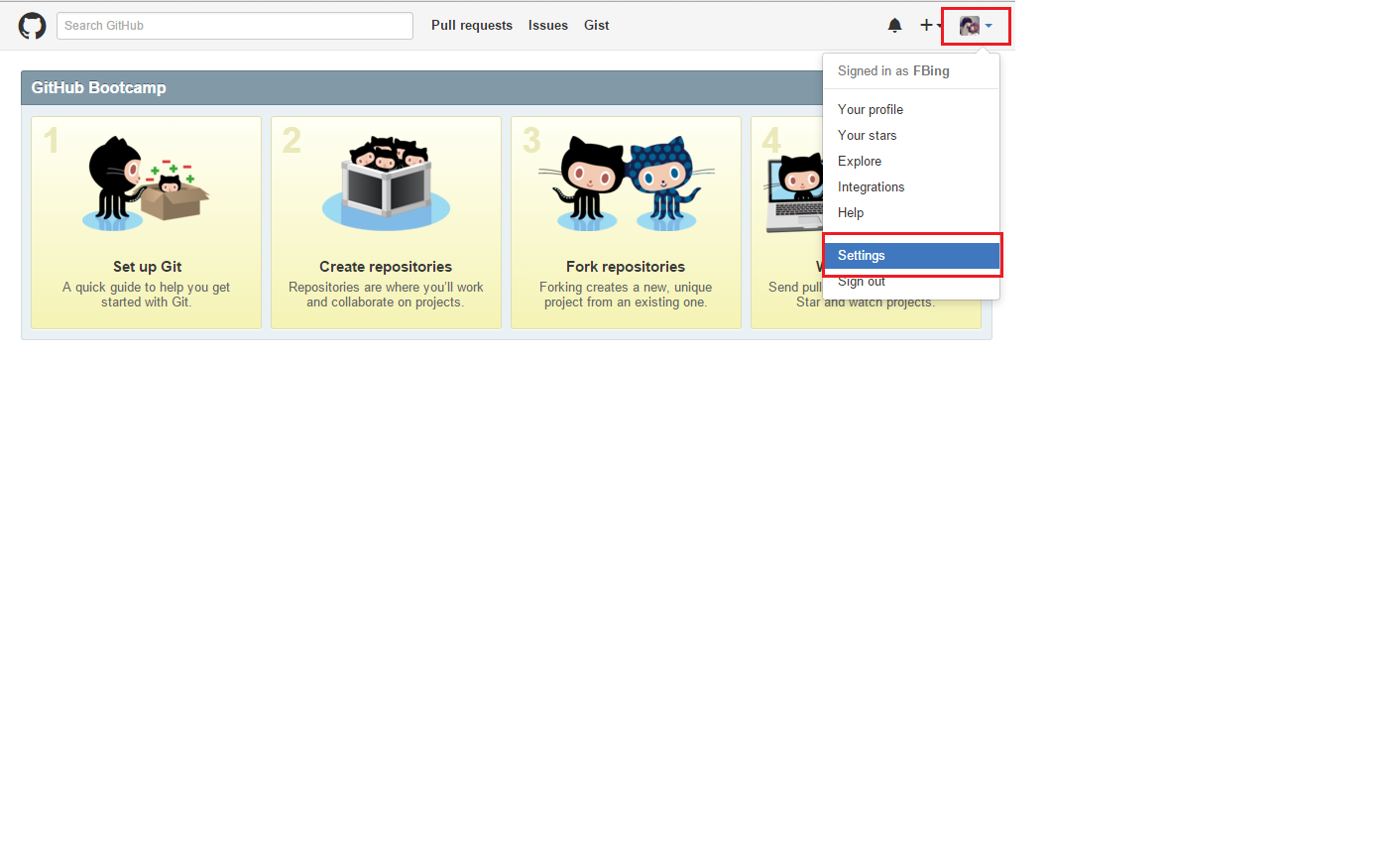
直接点回车，说明会在默认文件id\_rsa上生成ssh key。   


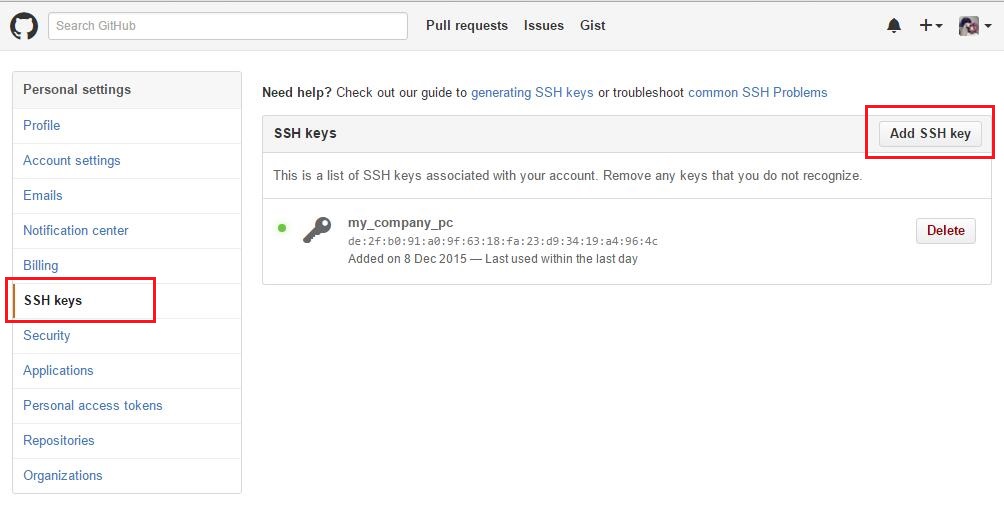
因我之前已经生成过了，所以会提示我已经rsa key已经存在，直接输y选择覆盖即可。   
然后系统要求输入密码，直接按回车表示不设密码   


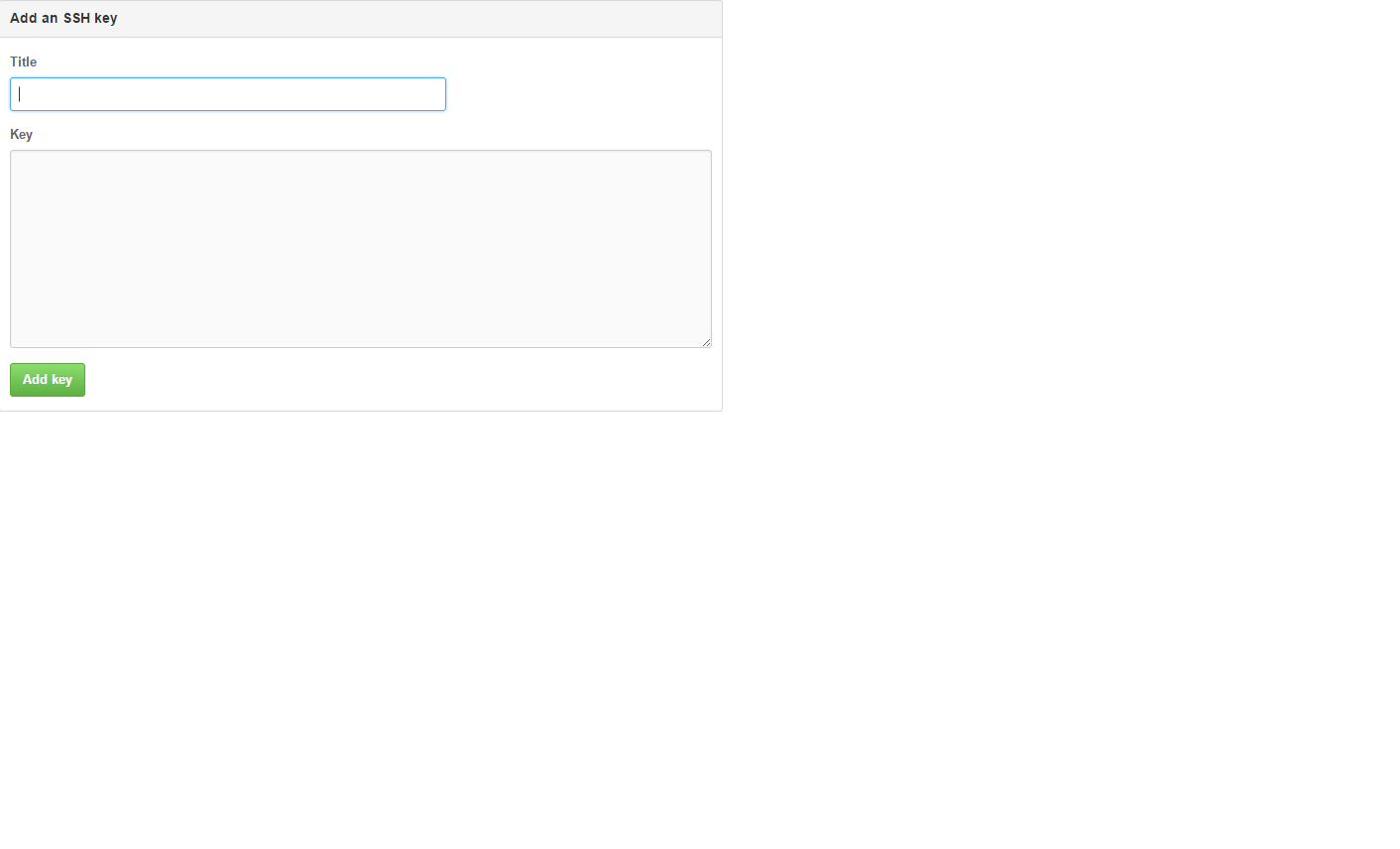
重复密码时也是直接回车，之后提示你shh key已经生成成功。   


然后我们进入提示的地址下查看ssh key文件。 我的电脑的地址是C:\Users\YULORE-USER.ssh ，其中YULORE-USER是我的电脑的名称   


打开id\_rsa.pub，复制里面的key。里面的key是一对看不懂的字符数字组合，不用管它，直接复制。

然后回到github网站，进入Settings   


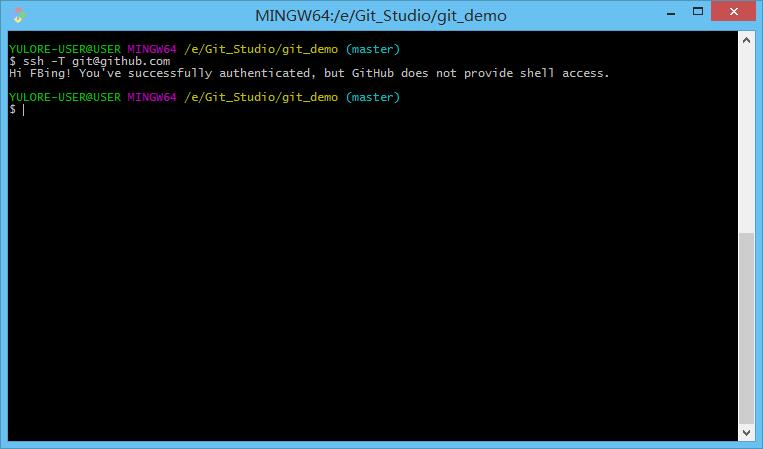
左边选择SSH Keys，Add SSH Key   




title随便填，粘贴key，然后点击 Add key 按钮。

3）验证是否成功，在git bash下输入

$ ssh -T git@github.com

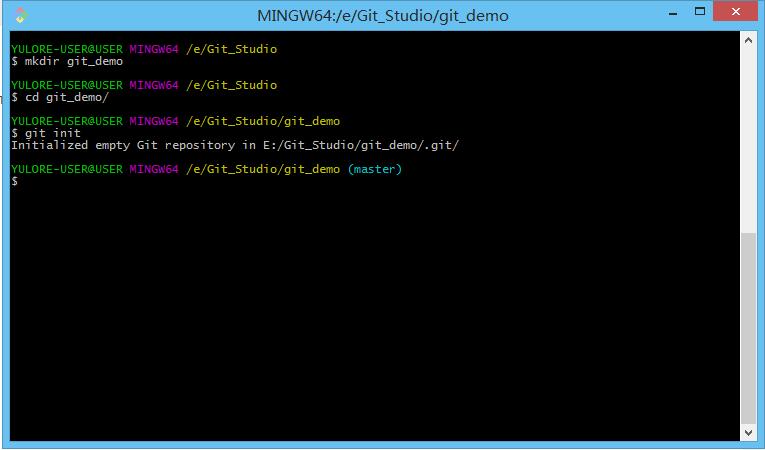
回车就会看到：You’ve successfully authenticated, but GitHub does not provide shell access 。这就表示已成功连上github。   


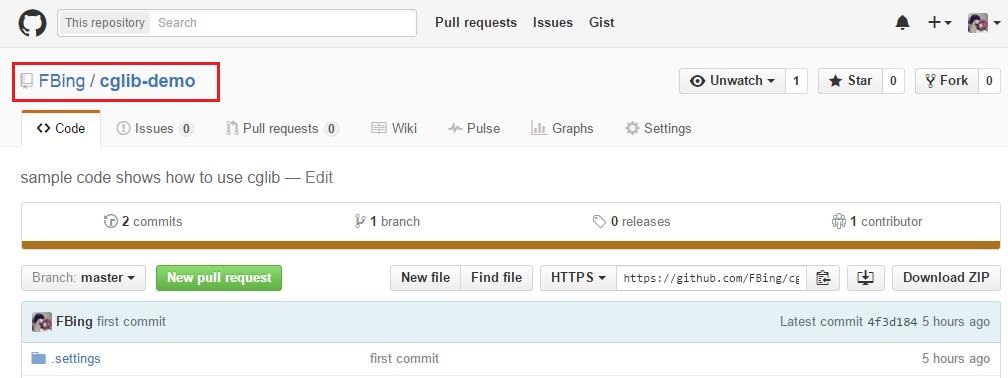
4）接下来我们要做的就是在本地电脑上新建一个仓库

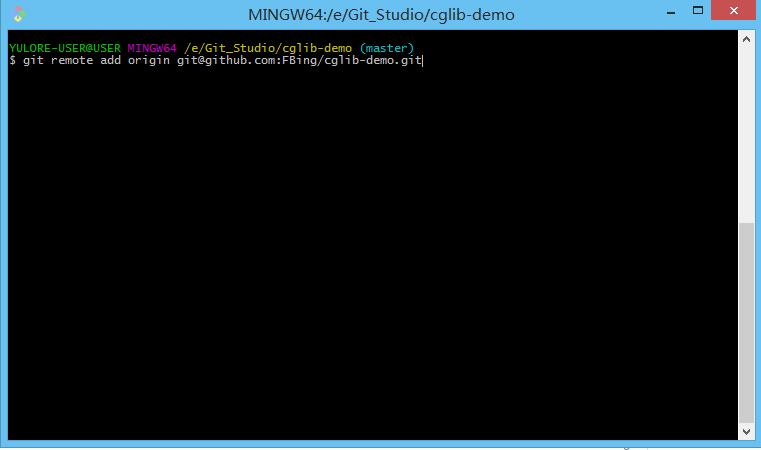
$ mkdir git\_demo

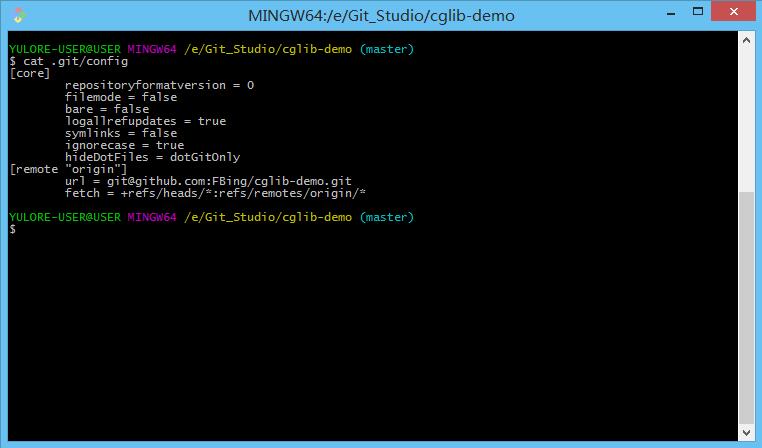
$cd git\_demo

$ git init



5）添加远程地址   
$ git remote add origin git@github.com:yourName/yourRepo.git   
后面的yourName和yourRepo表示你再github的用户名和刚才新建的仓库   


执行命令   


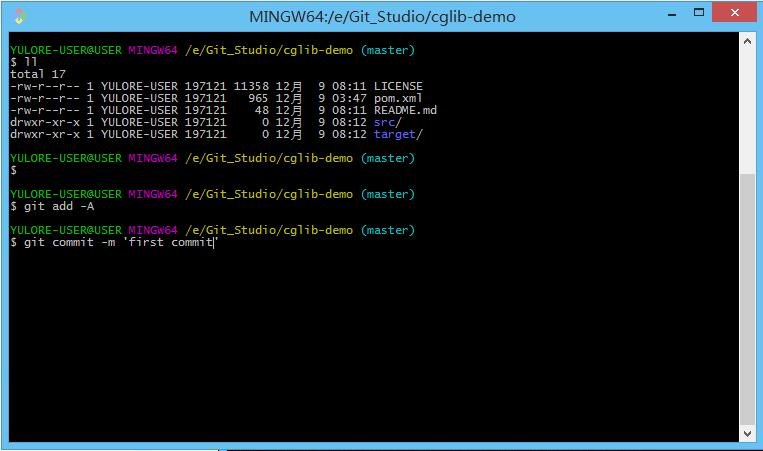
加完之后cat 查看 .git/config，这里会多出一个remote “origin”内容，这就是刚才添加的远程地址，也可以直接修改config来配置远程地址   


**6）**提交上传*\*\**   
6.1）接下来在本地仓库里添加一些文件，然后在命令行输入一下命令

$ git add -A

$ git commit -m "first commit"

* 1

我的执行界面如下   


6.2）push本地文件到github   
执行命令

$ git push origin master

* 1
* 1

git push命令会将本地仓库推送到远程服务器。git pull 命令则相反。   
注：首次提交，先git pull下，修改完代码后，使用git status可以查看文件的差别，使用git add 添加要commit的文件。

大功告成，现在你的本地的项目已经提交到github上了。