

前提准备:

1. 更改主机名为你的个人姓名;
2. git 工具的下载和安装(<https://gitforwindows.org/>);
3. github 上进行注册;
4. Git 开始连接 github 远程仓库。

一、Git 的配置

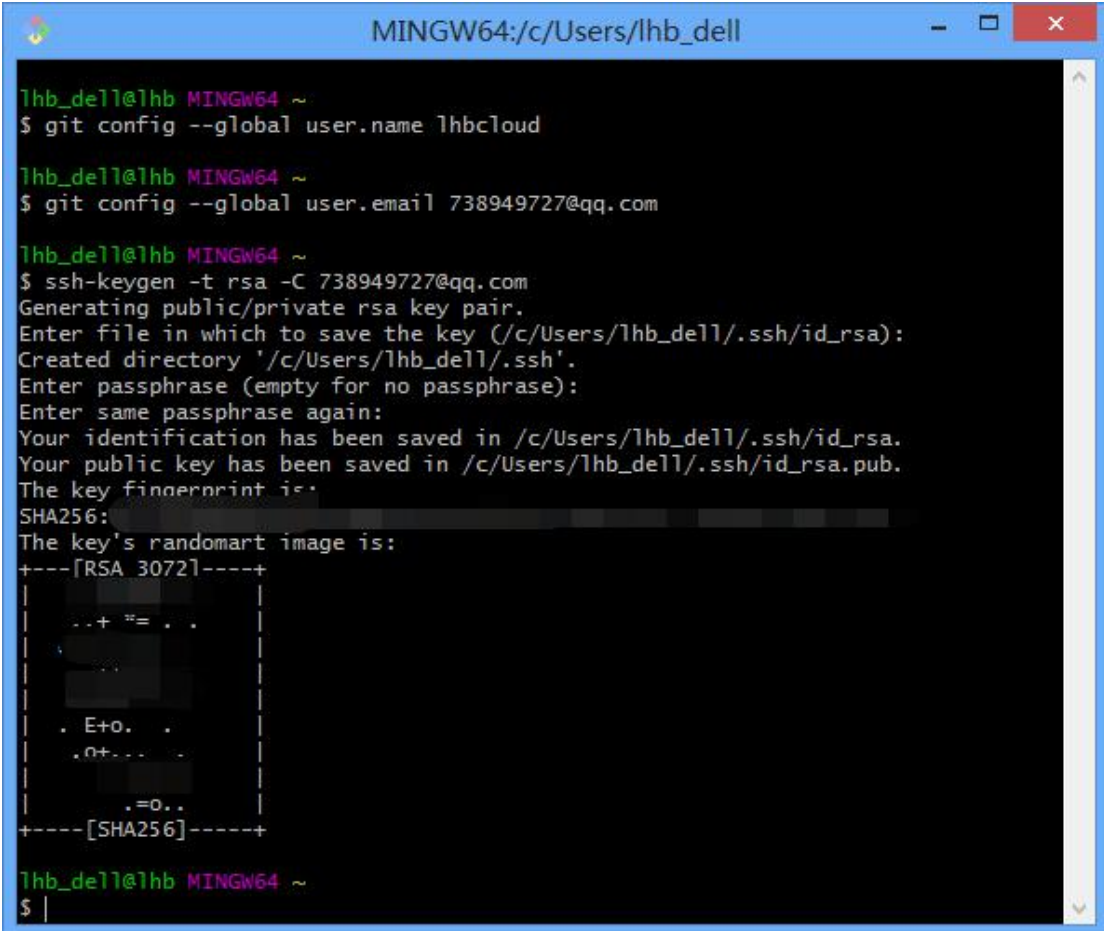
- 1.设置用户名和邮箱(--global 为全局参数, 表明本地所有 Git 仓库都会使用这个配置)

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

```
git config --global user.email "your_email@youremail.com"
```

- 2.生成密钥(SSH key)

```
ssh-keygen -t rsa -C "your_email@youremail.com"
```



```
lhb_dell@lhb MINGW64 ~
$ git config --global user.name lhbcloud

lhb_dell@lhb MINGW64 ~
$ git config --global user.email 738949727@qq.com

lhb_dell@lhb MINGW64 ~
$ ssh-keygen -t rsa -C 738949727@qq.com
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/lhb_dell/.ssh/id_rsa):
Created directory '/c/Users/lhb_dell/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/lhb_dell/.ssh/id_rsa.
Your public key has been saved in /c/Users/lhb_dell/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:
The key's randomart image is:
+---[RSA 3072]---+
|  ..+  =  .  |
|  .    .    |
|  .  E+o.   |
|  .  o+...   |
|  .  =o..   |
+---[SHA256]---+

lhb_dell@lhb MINGW64 ~
$ |
```

- 3.添加密钥(SSH key), 并验证是否成功

添加密钥：将上一步骤生成的密钥即.ssh/id_rsa.pub 中内容全部复制。在 github 的 Settings-->SSH and GPG keys-->New SSH key, key 中粘贴复制的内容(Title 自定义)。

验证：github 输入第一条的命令，码云输入第二条

a. `ssh -T git@github.com`

```
lhb_dell@lhb MINGW64 ~
$ ssh -T git@github.com
The authenticity of host 'github.com (13.250.177.223)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,13.250.177.223' (RSA) to the list of known hosts.
Hi lhbcloud! You've successfully authenticated, but GitHub does not provide shell access.

lhb_dell@lhb MINGW64 ~
$ |
```

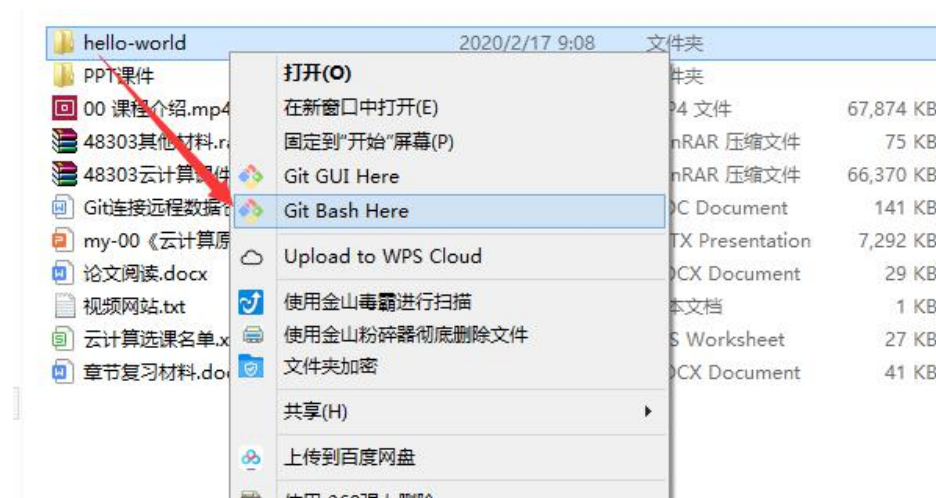
b. `ssh -T git@gitee.com`

二、创建项目工程

1.远程仓库：在 github 中 New repository 输入 Repository name。[例如：hello-world]

2.项目工程：在自己本地电脑上新建一个与 github 新项目工程同名的文件夹：hello-world

三、创建版本库



进入步骤二中的文件夹下，输入以下命令初始化仓库：

```
git init
```

```

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world
$ git init
Initialized empty Git repository in F:/教学文档/教学课件/云计算/云计算原理与实践/hello-world/.git/

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$ |

```

出现: Initialized empty Git repository in E:/** /**/.git/ 则表示创建成功[注意: 此时会生成一个.git 目录(隐藏目录)]



四、连接远程仓库(下面两种方式都可以)

```
git remote add origin git@github.com:yourName/repositoryname.git
```

```
git remote add origin https://github.com/yourName/repositoryname.git
```

```

MINGW64:/f/教学文档/教学课件/云计算/云计算原理与实践/hello-...
lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world
$ git init
Initialized empty Git repository in F:/教学文档/教学课件/云计算/云计算原理与实践/hello-world/.git/

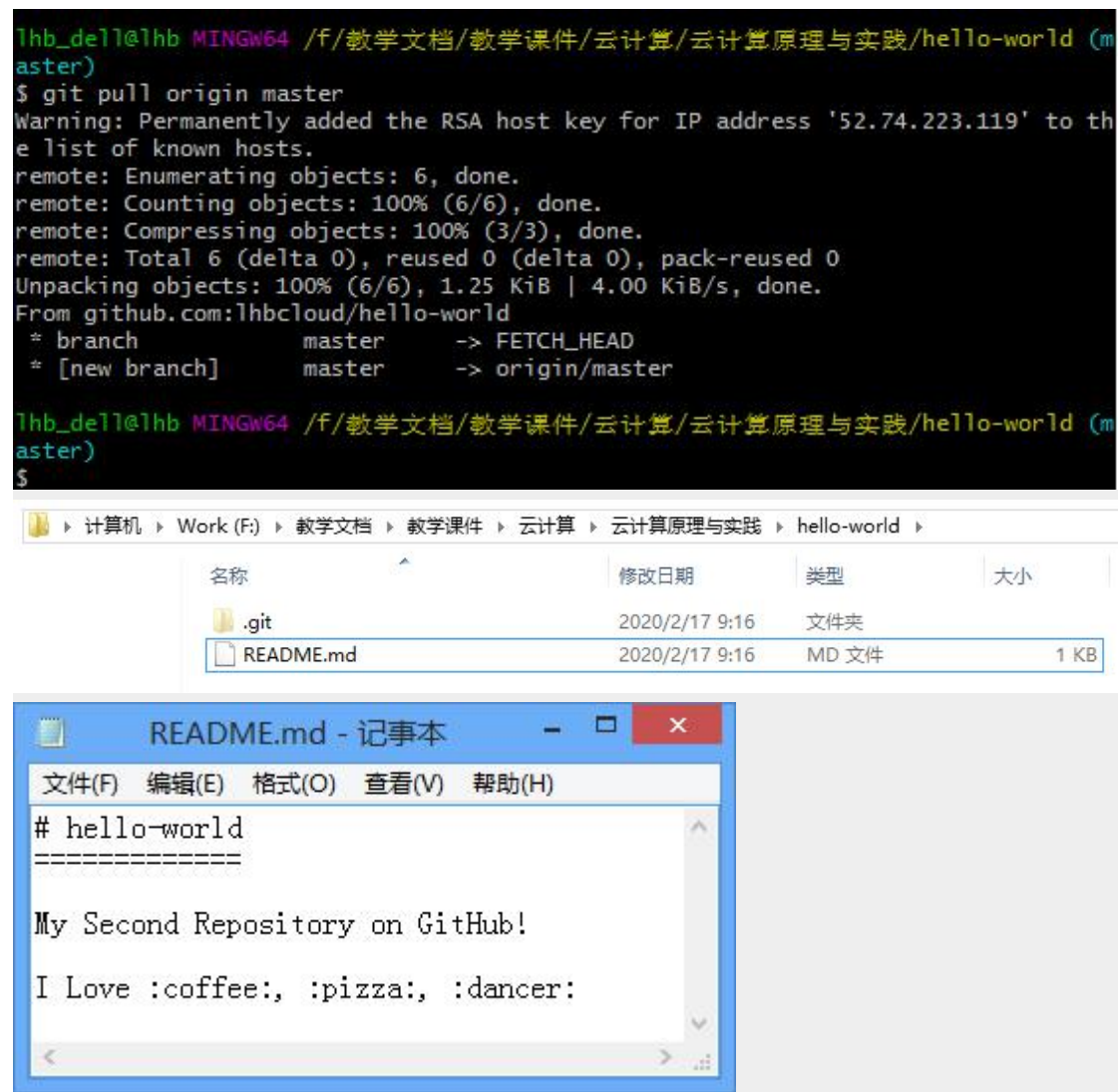
lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$ git remote add origin git@github.com:lhcloud/hello-world.git

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$ |

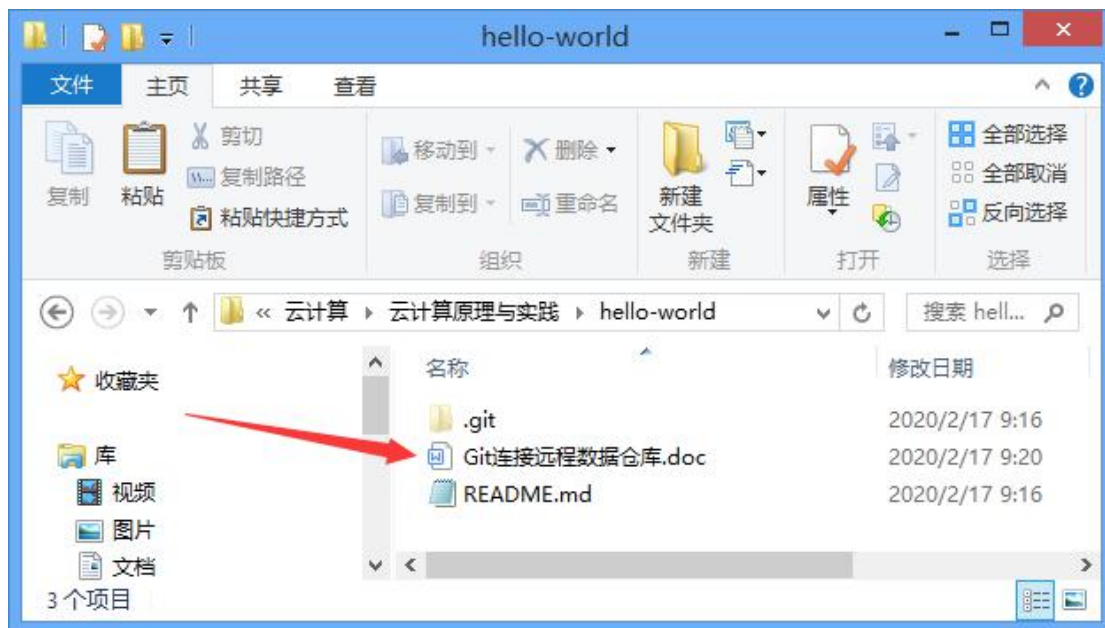
```

五、从远程仓库 pull 文件(若远程仓库没有文件, 直接执行步骤六)

```
git pull origin master
```



六、将本地文件 **push** 到远程仓库(若没有文件则手动创建)



1. `git status` 查看工作目录的状态

```

MINGW64:/f/教学文档/教学课件/云计算/云计算原理与实践/hello-...
git: 'ststatus' is not a git command. See 'git --help'.

The most similar command is
    status

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        "Git\350\277\236\346\216\245\350\277\234\347\250\213\346\225\260\346\215\256\344\273\223\345\272\223.doc"

nothing added to commit but untracked files present (use "git add" to track)

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$ |
  
```

2. `git add <file>` 将文件添加到暂存区


```
MINGW64:/f/教学文档/教学课件/云计算/云计算原理与实践/hello-... - □ ×

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    "Git\350\277\236\346\216\245\350\277\234\347\250\213\346\225\260\346\215\256\344\273\223\345\272\223.doc"

nothing added to commit but untracked files present (use "git add" to track)

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$ git add Git连接远程数据仓库.doc

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$
```

```
MINGW64:/f/教学文档/教学课件/云计算/云计算原理与实践/hello-... - □ ×

nothing added to commit but untracked files present (use "git add" to track)

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$ git add Git连接远程数据仓库.doc

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   "Git\350\277\236\346\216\245\350\277\234\347\250\213\346\225\260\346\215\256\344\273\223\345\272\223.doc"

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$
```

3. `git commit -m "commnet"` 提交更改, 添加备注信息 (此时将暂存区的信息提交到本地仓库)

```
MINGW64:/f/教学文档/教学课件/云计算/云计算原理与实践/hello-... - □ ×

$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   "Git\350\277\236\346\216\245\350\277\234\347\250\213\346\225\260\346\215\256\344\273\223\345\272\223.doc"

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$ git commit -m 测试提交新文件
[master c3f2f52] 测试提交新文件
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "Git\350\277\236\346\216\245\350\277\234\347\250\213\346\225\260\346\215\256\344\273\223\345\272\223.doc"

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$
```

4. `git push origin master` 将本地仓库的文件 push 到远程仓库 (若 push 不成功, 可加 `-f` 进行强推操作)

The image shows a terminal window and the GitHub web interface. The terminal window displays the output of a `git push origin master` command, indicating a successful push to the remote repository. The GitHub interface shows the user's profile, the repository list, and the details of the 'hello-world' repository. A red arrow points from the 'hello-world' repository in the list to its details page, and another red arrow points from the 'Clone or download' button to the file list.

```
lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 208.35 KiB | 826.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:lhbcloud/hello-world.git
3ca4032..c3f2f52 master -> master

lhb_dell@lhb MINGW64 /f/教学文档/教学课件/云计算/云计算原理与实践/hello-world (master)
$
```

GitHub interface details:

- User: lhbcloud
- Repositories: 2
- Repository: hello-world (Updated 1 minute ago)
- Repository type: Cloud Private
- Repository description: No description, website, or topics provided.
- Repository stats: 3 commits, 1 branch, 0 packages, 0 releases, 1 contributor
- Files: lhbcloud 测试提交新文件, Git连接远程数据库.doc, README.md

注: 至此已经完成了 远程与本地仓库的配置, 若需要单独配置可见以下操作

七、所遇问题及解决办法

Q1. `git pull origin master` 无法进行 pull, 出现如下提示:

```
git pull origin master
```

```
fatal: unable to access 'https://github.com/yourName/Demo.git': error
setting certificate verify locations:
  CAfile: G:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
  CPath: none
```

分析: ca-bundle.crt 文件是证书文件。根据提示 CPath: none 没有该文件，所以无法访问远程仓库

解决: 修改为正确路径 或者 将证书验证设置 false

```
git config --system http.sslcainfo E:/Program
Files/Git/mingw64/ssl/certs/ca-bundle.crt

git config --system http.sslverify false
```

Q2.git pull origin master 出现如下提示:

```
fatal: refusing to merge unrelated histories
```

解决: 如下操作即可解决

```
git pull origin master --allow-unrelated-histories
```

Q3.每次 git push origin master 时都需要输入用户名和密码:

因为配置的时候使用的是 https 协议，所以每次都需要输入

```
git remote -v 查看远程连接

git remote rm origin 删除远程连接

git remote add origin git@github.com:yourName/repositoryname.git
```

**Q4: git 报错\$ git push -u origin master ERROR: Repository not found. fatal:
Could not read from remote repository**

```
git remote set-url origin https://github.com/lhb738441242/Test.git
```