

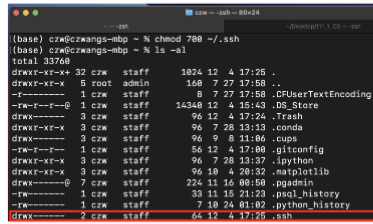
## 金鑰與GitHub連線

- 金鑰對(公鑰&私鑰)
- 將公鑰放在遠端伺服器(GitHub)

### 使用 SSH 金鑰與 GitHub 連線

→ 建立 .ssh 目錄

1. **MacOS:** `mkdir ~/.ssh`  
**Windows:** `mkdir c:/Users/username/.ssh`
2. `chmod 700 ~/.ssh`



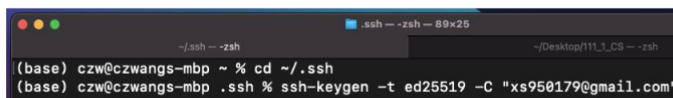
```
(base) czw@czwangs-mbp ~ % chmod 700 ~/.ssh
(base) czw@czwangs-mbp ~ % ls -al
total 33768
drwxr-xr-x  32 czw  staff   1024 12  4 17:55 .
drwxr-xr-x  5 root  admin   160  7 27 17:58 ..
-r-----  1 czw  staff    8  7 27 17:58 .CFUserTextEncoding
-rw-r--r--  1 czw  staff  14340 12  4 15:43 .DS_Store
drwxr-xr-x  3 czw  staff   96 12  4 17:54 .Trash
drwxr-xr-x  3 czw  staff   96  7 28 13:13 .conda
drwxr-xr-x  3 czw  staff   96  9  8 11:06 .cups
-rw-r--r--  1 czw  staff   56 12  4 17:09 .gitconfig
drwxr-xr-x  3 czw  staff   96  7 28 13:37 .ipython
drwxr-xr-x  3 czw  staff   96 10  4 20:32 .matplotlib
drwxr-xr-x  7 czw  staff  224 11 16 00:50 .ngadmin
-rw-r--r--  1 czw  staff   33 11 15 21:23 .psql_history
-rw-r--r--  1 czw  staff   7 10 24 01:02 .python_history
drwxr-xr-x  2 czw  staff   64 12  4 17:55 .ssh
```

9

### 使用 SSH 金鑰與 GitHub 連線

→ 產生金鑰對

1. `cd ~/.ssh`
2. `ssh-keygen -t ed25519 -C "your_email@example.com"`



```
~/ssh -- zsh
(base) czw@czwangs-mbp ~ % cd ~/.ssh
(base) czw@czwangs-mbp .ssh % ssh-keygen -t ed25519 -C "xs950179@gmail.com"
```

10

## 使用 SSH 金鑰與 GitHub 連線

→ 產生金鑰對(過程中會問三個問題)

1. Enter file in which to save the key (/Users/czw/.ssh/id\_rsa): **/Users/username/.ssh/github\_key**  
# 詢問儲存檔案的位置和檔名，預設檔名是id\_ed25519。
2. Enter passphrase (empty for no passphrase):
3. Enter same passphrase again:  
# 詢問是否設定密碼來保護金鑰。  

Generating public/private ed25519 key pair.  
Enter file in which to save the key (/Users/czw/.ssh/id\_ed25519): /Users/czw/.ssh/id\_rsa  
Enter passphrase (empty for no passphrase):

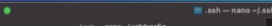
```
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/Users/czwz/cwzg-sah/id_ed25519): /Users/czwz/sah/github/  
pub.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /Users/czwz/sah/github/key.  
Your public key has been saved in /Users/czwz/sah/github/key.pub.  
The key fingerprint is:  
SHA256:f7f82B6A03c9kuzqWZS5G0UmpM7ZeHMAIjN3jPTaXy9S0179@gmail.com  
The key's randomart image is:  
+-----+  
|          |  
|.=====|. |  
|O..cccc+.||  
|o .m+ = o ||  
|..m+.   ||  
|..m+...||  
|o m+ S +||  
|=d= o o +||  
| ..o . o ||  
|          |  
+-----+  
SHA256:  
(base) Czwzcwzg-maps - sah N
```

## 使用 SSH 金鑰與 GitHub 連線

→ 本機設定 ssh config

```
nano ~/.ssh/config
```

```
Host github.com
  HostName github.com
  User github_username
  IdentityFile ~/.ssh/github_key
```



```
ssh - nano - /ssh/config - 89x25
~/.ssh - nano - /ssh/config
GNU nano 2.0.6      File: /Users/czw/.ssh/config

Host github.com
HostName github.com
User ChangZaiWang
IdentityFile ~/.ssh/github_key
```

1. control + o 儲存檔案
2. enter
3. control + x 離開 nano

12

## 使用 SSH 金鑰與 GitHub 連線

## ➔ 新增公鑰至 GitHub 遠端倉庫

1. `ls`
2. `cat github_key.pub`

```
(base) czw@czwangs-mbp: ~ % ssh -t -Z -R 89x25 -- /Desktop/r11_C3 -- -Zsh
```

```
-Zsh -- -Zsh                                     ~/Desktop/r11_C3 -- -Zsh
```

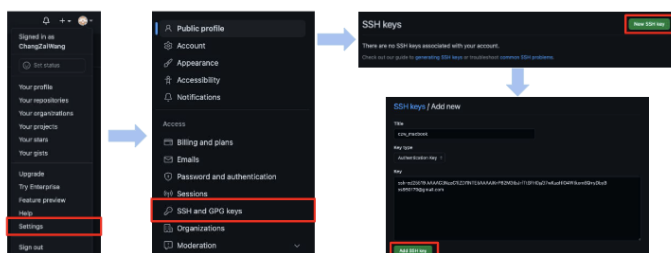
```
(base) czw@czwangs-mbp: ~ % ssh -t -Z -R 89x25 -- /Desktop/r11_C3 -- -Zsh
```

```
github_key      github_key.pub  
[base] czw@czwangs-mbp: ~ % cat github_key.pub  
ssh-ed25519 AAAAC3NzaC1lZD01INTE5AAAIAK+p82M3btfrjITSFH0y/37wKuahI04nk0m6Q+yObYB X 9508179  
@gmail.com  
[base] czw@czwangs-mbp: ~ % ssh -t -Z -R 89x25 -- /Desktop/r11_C3 -- -Zsh
```

13

## 使用 SSH 金鑰與 GitHub 連線

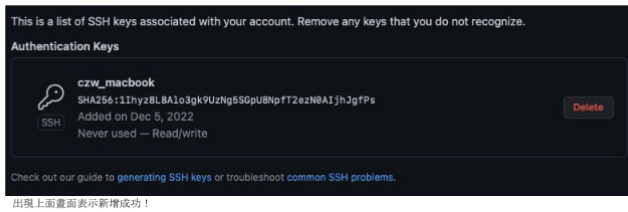
## ➔ 新增公鑰至 GitHub 遠端倉庫



14

## 使用 SSH 金鑰與 GitHub 連線

→ 新增公鑰至 GitHub 遠端倉庫

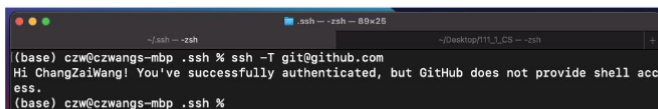


15

## 使用 SSH 金鑰與 GitHub 連線

→ 測試連線

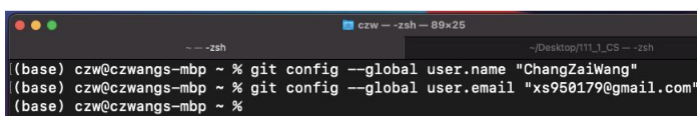
```
ssh -T git@github.com
```



16

## Git 連結帳號至 Github

1. \$ git config --global user.name "Your Name"
2. \$ git config --global user.email "your@gmail.com"



18

## Git 指令

- 初始化專案: git init
- 單一檔案加入索引(暫存區): git add <檔案名稱>
- 所有檔案加入索引(暫存區): git add . # "." 的意思代表全部
- 觀看當前狀態: git status
- 提交版本: git commit -m "修改紀錄"
- 瀏覽歷史紀錄: git log

## Git-使用方法 Local(本地端)



### I. 新增Working directory(工作目錄)

1. 建立資料夾(ex: mkdir cs6\_git)
2. 移動到資料夾: `$ cd cs6_git`
3. 將專案資料夾建立成git repository: `$ git init` (初始化資料夾)  
# 此時會產生隱藏檔(.git), 而這個隱藏檔會追蹤修改
4. 新增檔案(ex: index.html)
5. 查看資料夾內檔案變化: `$ git status`  
# 此時檔案尚未被追蹤(Untracked files)

20

## Git-使用方法 Local(本地端)



### 二. 進入Staging area(暫存區)

1. 將新增或變更的檔案加入追蹤: `$ git add index.html`
2. 查看資料夾內檔案變化: `$ git status`  
# 此時檔案已被追蹤
3. 再新增一個檔案(ex: README.md)
4. 查看資料夾內檔案的變化: `$ git status`
5. 將所有的檔案加入追蹤: `$ git add --all`

21

## Git-使用方法 Local(本地端)



### 三. 進入Local repository(本地數據庫)

1. 將檔案移入本地repo, 提交新版本:  
`$ git commit -m "First release of Hello"`  
# -m: 填寫版本資訊、修改紀錄。方便日後查找
2. 使用`$ git status`觀察: `$ git status`  
# 出現"nothing to commit, working tree clean", 因為暫存區的檔案已被提交成新的版本了
3. 查看新增的版本(歷史紀錄): `$ git log`  
# 會看到版本更新紀錄

22

## Git- 取消追蹤檔案 Local(本地端)



四. 檔案從Staging area(暫存區)退回Working directory工作目錄

(不是真的想把這個檔案刪掉, 只是不想讓這個檔案再被 Git 控管)

- 修改index.html: nano index.html
- 查看狀態: \$ git status
- 將檔案加入索引: \$ git add index.html
- 查看狀態: \$ git status
- 取消追蹤檔案: \$ git rm --cached index.html  
#此時檔案變回尚未被追蹤(Untracked files)
- 查看狀態: \$ git status

## Git- 取消commit Local(本地端)



五. 將commit拆掉

- 新增檔案: nano hello.py
- 將檔案加入索引: \$ git add .
- 提交版本: \$ git commit -m "print method"
- 查看歷史紀錄: \$ git log
- 取消commit: \$ git reset --mixed HEAD^  
# ^符號表示「前一次」的意思, 回到前n個 HEAD~n
- 查看歷史紀錄: \$ git log  
# 會發現最後一次commit的紀錄不見了
- 查看狀態: \$ git status  
# 會出現hello.py尚未被追蹤, 代表此時檔案在工作目錄

## Git-使用方法 Remote(雲端)



七. Remote repository(遠端數據庫)

1. 到github上新增或編輯README.md
2. 把遠端東西拉回來: \$ git fetch # 執行 Fetch 指令後, Git 看了一下線上版本的内容後, 將目前線上有但本地這邊沒有的内容抓了一份下來
3. 查看狀態: \$ git status
4. 查看歷史紀錄: \$ git log # 會發現在github上發送的commit被抓下來了
5. 比較本地分支和遠端分支之内容的不同: \$ git diff origin/main
6. 將遠端和本地端的内容做合併: \$ git merge origin/main
7. 查看狀態: \$ git status
8. Pull下載更新: \$ git pull origin # 將本地main的資料更新, 以防與Remote端不同

