

Git

陳肇廷

Installation

Mac

`brew install git`

Windows

`winget install -e --id Git.Git`

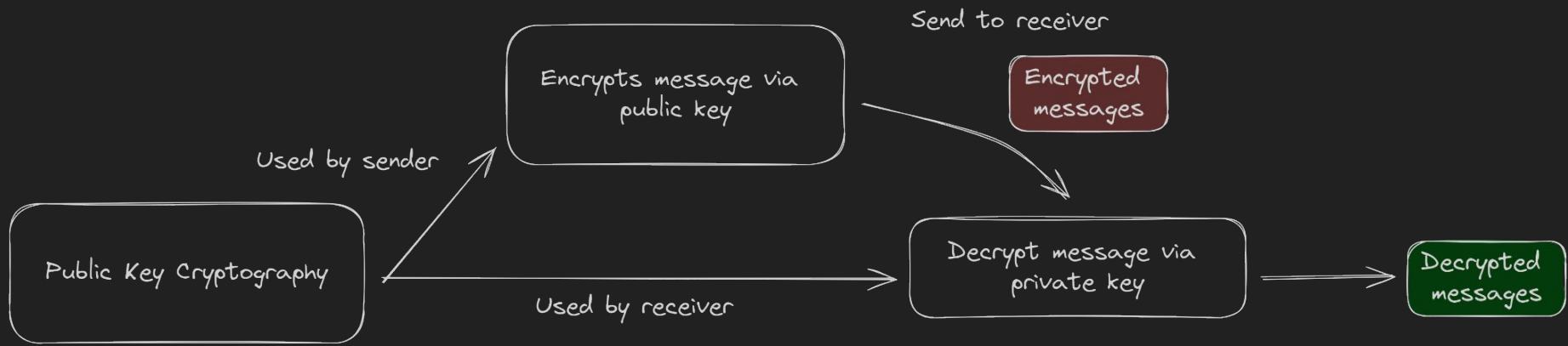
Linux

`sudo apt install git`

SSH

Secure Shell

在不安全的網路中提供安全的傳輸環境



```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

將 .ssh/xxx.pub 的內容複製下來

cd ~/.ssh

Mac

pbcopy < xxx.pub

Windows

cat xxx.pub | Set-Clipboard

GitHub

原始碼代管服務平台, 許多企業原始碼都儲存在上面
e.g. Minecraft, Meta UI lib - StyleX

sign up *github.com*

Settings > SSH and GPG keys

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

SSH keys / Add new

Title

Key type

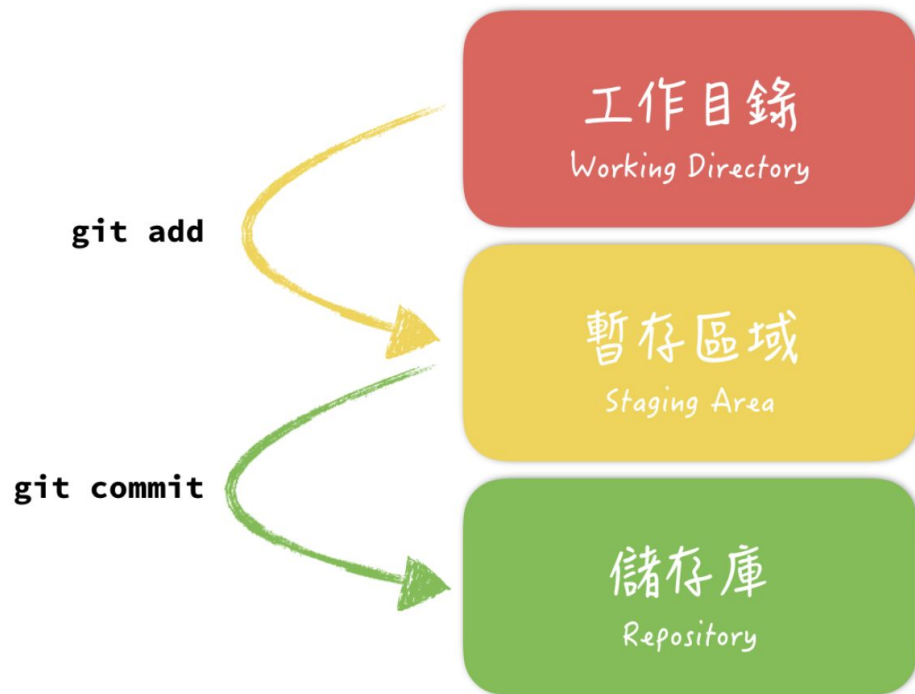
Authentication Key ▾

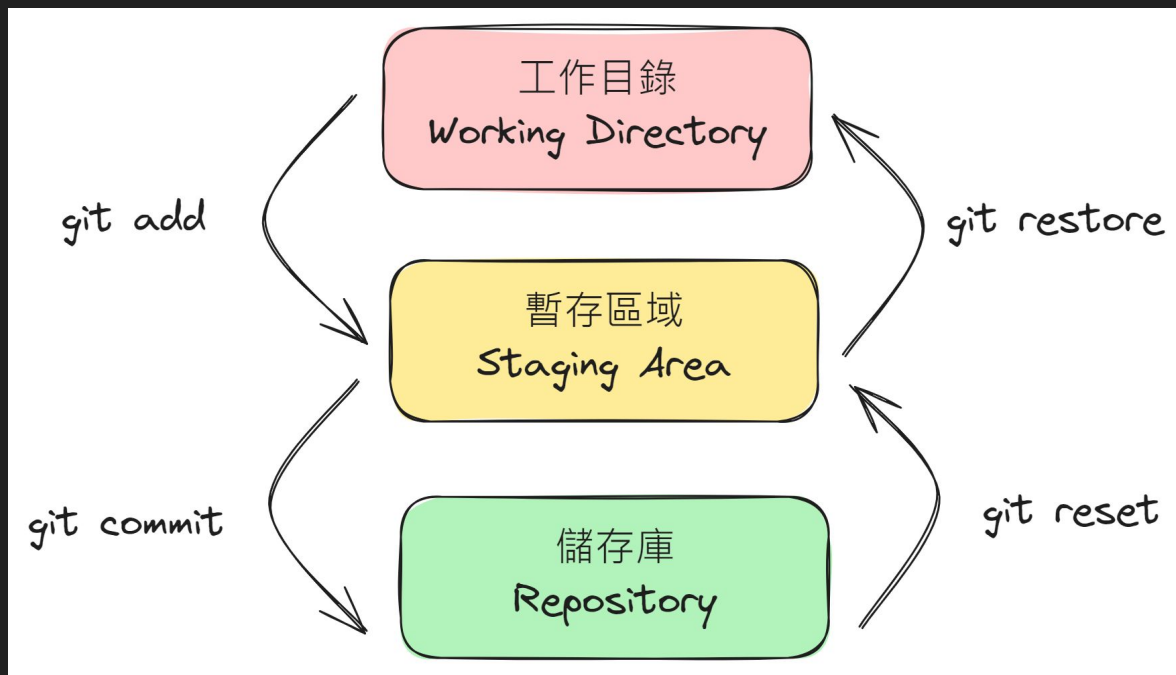
Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

Concept





Status

git status

位於分支 main

您的分支與上游分支 'origin/main' 一致。

要提交的變更：

(使用 "git restore --staged <檔案>..." 以取消暫存)

新檔案: README.md <- *Staging Area*

尚未暫存以備提交的變更：

(使用 "git add <檔案>..." 更新要提交的內容)

(使用 "git restore <檔案>..." 捨棄工作區的改動)

修改: ../README.md <- *Working directory*

修改: ../fish/config.fish

修改: ../format/.clang_format

修改: lazy-lock.json

未追蹤的檔案：

(使用 "git add <檔案>..." 以包含要提交的內容)

../fish/README.md <- *Working directory*

../powershell/README.md

../tmux/README.md

Adding to Repository

```
> git commit -m "docs: Add readme"
```

```
[main 3eb0463] docs: Add readme
```

```
1 file changed, 37 insertions(+)
```

```
create mode 100644 nvim/README.md
```

位於分支 main

您的分支與上游分支 'origin/main' 一致。

尚未暫存以備提交的變更：

(使用 "git add <檔案>..." 更新要提交的內容)

(使用 "git restore <檔案>..." 捨棄工作區的改動)

修改: ../README.md <- *Working directory*

修改: ../fish/config.fish

修改: ../format/.clang_format

修改: lazy-lock.json

未追蹤的檔案：

(使用 "git add <檔案>..." 以包含要提交的內容)

../fish/README.md <- *Working directory*

../powershell/README.md

../tmux/README.md

Log

git log

```
> git log
```

```
commit ad8b01c440261ecc628944fc6c87d04341423b21 (HEAD -> main, origin/main)
```

```
Author: GINTOAH <chaotingchen10@gmail.com>
```

```
Date: Tue Aug 22 14:04:10 2023 +0800
```

```
docs: Add readme <- Your commit message
```

```
> git log --oneline
```

```
3eb0463 (HEAD -> main) docs: Add readme
```

```
ad8b01c (origin/main) fix: nvim-ts-rainbow archived cause tree-sitter tsx break
```

```
4ffbbf5 fix: Fix c/c++ formatter config
```

```
...
```



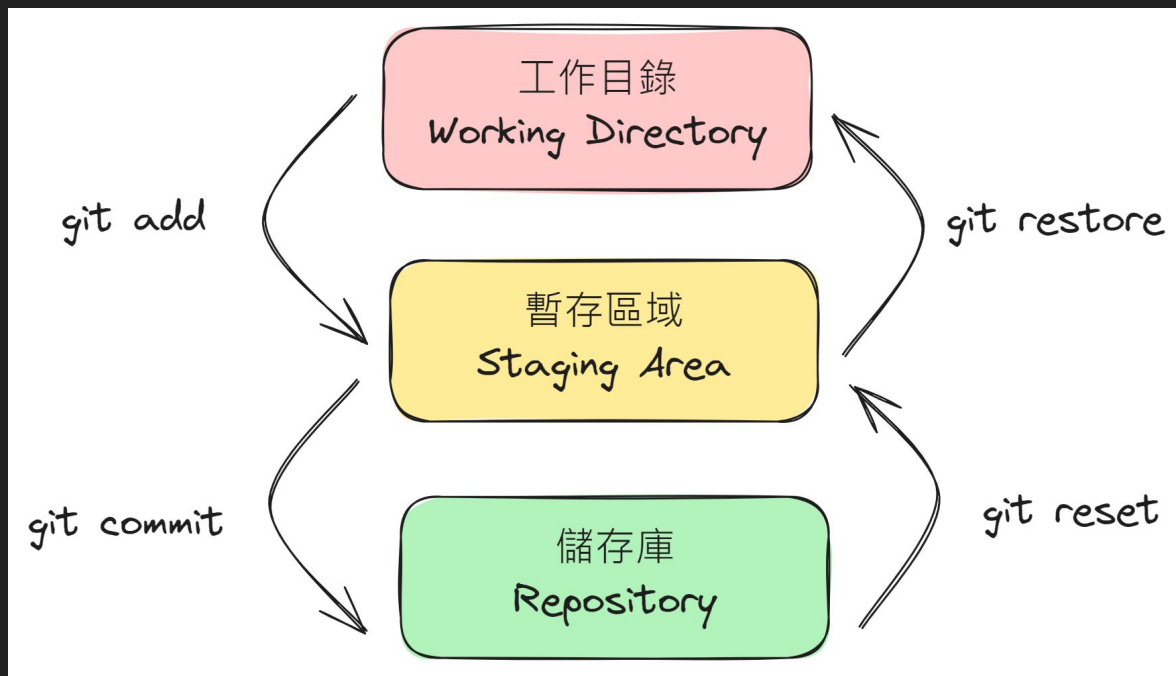
```
> git log --oneline -n 2
```

```
3eb0463 (HEAD -> main) docs: Add readme
```

```
ad8b01c (origin/main) fix: nvim-ts-rainbow archived cause tree-sitter tsx break
```

Reset / Restore

`git reset / git restore`



位於分支 main

您的分支與上游分支 'origin/main' 一致。

尚未暫存以備提交的變更：

(使用 "git add <檔案>..." 更新要提交的內容)

(使用 "git restore <檔案>..." 捨棄工作區的改動)

修改: ../README.md <- *Working directory*

修改: ../fish/config.fish

修改: ../format/.clang_format

修改: lazy-lock.json

未追蹤的檔案：

(使用 "git add <檔案>..." 以包含要提交的內容)

../fish/README.md <- *Working directory*

../powershell/README.md

../tmux/README.md

往前回覆一次 commit 但保持變更

> git reset --soft HEAD^

> git log --oneline -n 2

ad8b01c (origin/main) fix: nvim-ts-rainbow archived ...

4ffbbf5 fix: Fix c/c++ formatter config

位於分支 main

您的分支與上游分支 'origin/main' 一致。

要提交的變更：

(使用 "git restore --staged <檔案>..." 以取消暫存)

新檔案: README.md <- *Staging Area*

尚未暫存以備提交的變更：

(使用 "git add <檔案>..." 更新要提交的內容)

(使用 "git restore <檔案>..." 捨棄工作區的改動)

修改: ../README.md <- *Working directory*

修改: ../fish/config.fish

修改: ../format/.clang_format

修改: lazy-lock.json

未追蹤的檔案：

(使用 "git add <檔案>..." 以包含要提交的內容)

../fish/README.md <- *Working directory*

../powershell/README.md

../tmux/README.md

取消 *stage* 的狀態

> `git restore --staged README.md`

`ad8b01c` (`origin/main`) fix: nvim-ts-rainbow archived ...

`4ffbbf5` fix: Fix c/c++ formatter config

位於分支 main

您的分支與上游分支 'origin/main' 一致。

尚未暫存以備提交的變更：

（使用 "git add <檔案>..." 更新要提交的內容）

（使用 "git restore <檔案>..." 捨棄工作區的改動）

修改: ../README.md <- *Working directory*

修改: ../fish/config.fish

修改: ../format/.clang_format

修改: lazy-lock.json

未追蹤的檔案：

（使用 "git add <檔案>..." 以包含要提交的內容）

../fish/README.md <- *Working directory*

README.md <- *回到 unstaged 狀態*

../powershell/README.md

../tmux/README.md

取消變更, 類似 *ctrl-z*

> **git** restore lazy-lock.json

ad8b01c (**origin/main**) fix: nvim-ts-rainbow archived ...

4ffbbf5 fix: Fix c/c++ formatter config

位於分支 main

您的分支與上游分支 'origin/main' 一致。

尚未暫存以備提交的變更：

（使用 "git add <檔案>..." 更新要提交的內容）

（使用 "git restore <檔案>..." 捨棄工作區的改動）

修改: ../README.md <- *Working directory*

修改: ../fish/config.fish

修改: ../format/.clang_format

修改: lazy-lock.json <- 這行會消失

未追蹤的檔案：

（使用 "git add <檔案>..." 以包含要提交的內容）

../fish/README.md <- *Working directory*

README.md

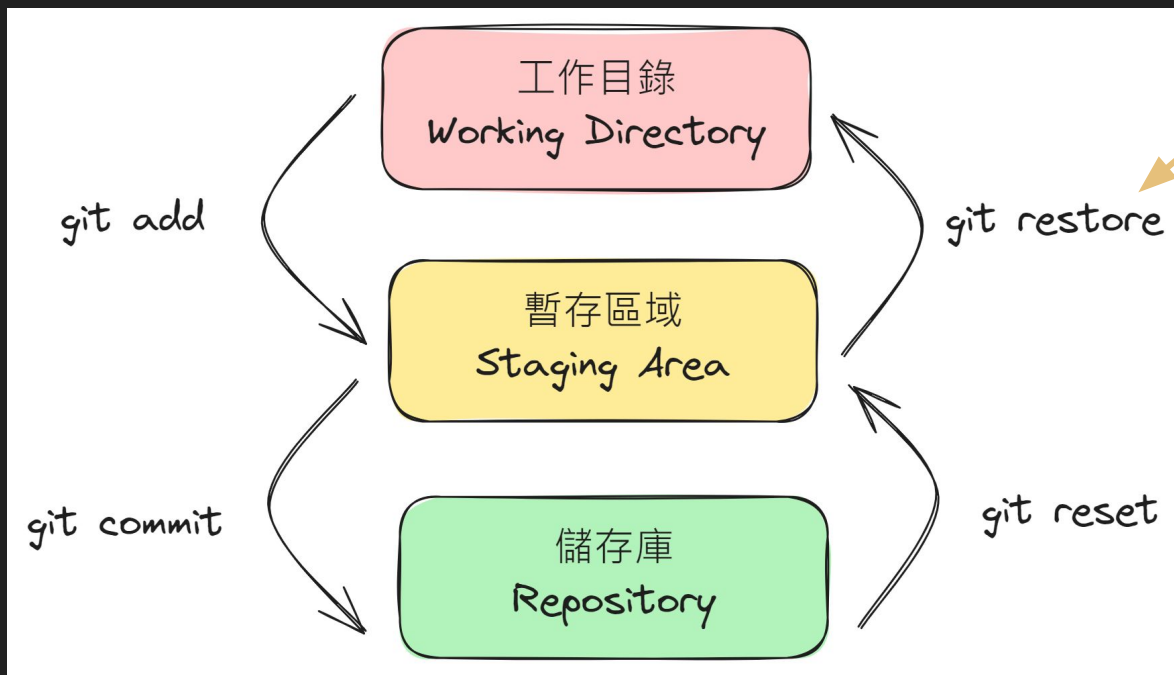
../powershell/README.md

../tmux/README.md

```
# With & without `--staged`
```

```
> git restore --staged
```

```
> git restore
```



With stage

位於分支 main

您的分支與上游分支 'origin/main' 一致。

尚未暫存以備提交的變更：

(使用 "git add <檔案>..." 更新要提交的內容)

(使用 "git restore <檔案>..." 捨棄工作區的改動)

修改: ../README.md <- *Working directory*

修改: ../fish/config.fish

修改: ../format/.clang_format

修改: lazy-lock.json <- 這行會消失

Without stage

“捨棄工作區的改動”

未追蹤的檔案：

(使用 "git add <檔案>..." 以包含要提交的內容)

../fish/README.md <- *Working directory*

README.md

../powershell/README.md

../tmux/README.md

Dotfiles

```
# Create your first git repo
```

```
> cd ~
```

```
> mkdir .dotfiles
```

```
# Go into the repo and initialize  
> cd .dotfiles  
> git init
```


Dotfiles

linking config file

```
# Move your nvim config to .dotfiles/
```

```
# Mac & Linux
```

```
> cd ~/.config/
```

```
> mv nvim ~/.dotfiles
```

```
> ln ~/.dotfiles/nvim .
```

```
# Move your nvim config to .dotfiles/  
# Windows  
> cd ~\.AppData\Local  
> Move-Item  
    -Path "$HOME\AppData\Local\nvim"  
    -Destination "$HOME\.dotfiles\nvim"  
> New-Item -ItemType SymbolicLink  
    -Path "$HOME\AppData\Local\nvim"  
    -Target "$HOME\.dotfiles\nvim"
```

Dotfiles

Staging config file

```
> cd ~/.dotfiles # 進入 .dotfiles 資料夾  
> git add . # working directory -> staging area
```

Dotfiles

Committing config file

```
> git commit -m <commit message>
```

Dotfiles

Push to remote


```
> git branch -m main
```

```
> git branch -m main  
> git remote add origin git@...
```

```
> git branch -m main  
> git remote add origin git@...  
> git push -u origin main
```

Contact me

chaotingchen10@gmail.com