

Introduction to Git



Agenda

- Git
- Git Hub
- Git CLI
- Git GUI
- Git Sign



Git

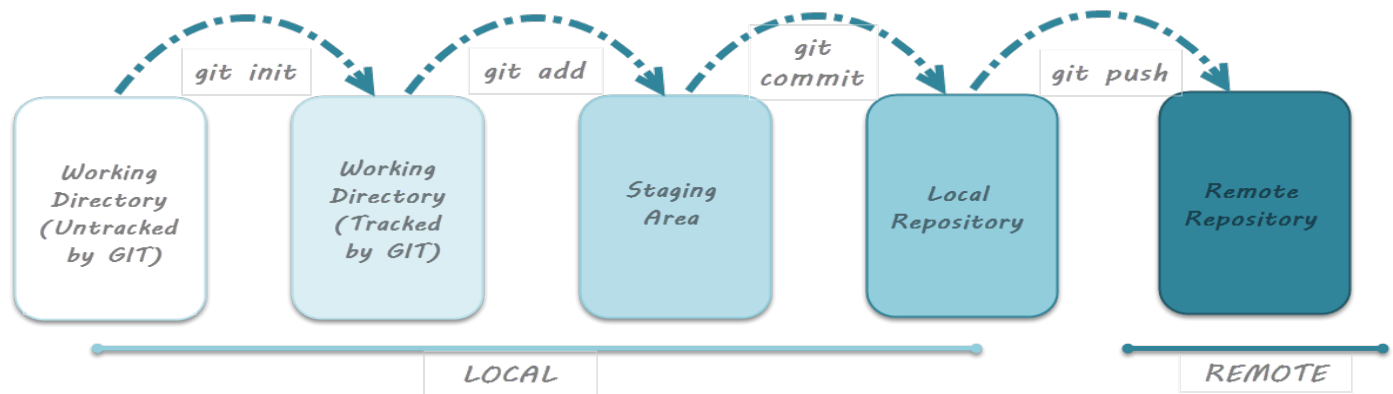
- What is Git?
 - Version Control System

Git

- Why use Git?
 - Source Code Tracking
 - Team Work Development

Git

- Git State
 - Untracked
 - Working Directory
 - Staged
 - Local Repository
 - Remote Repository

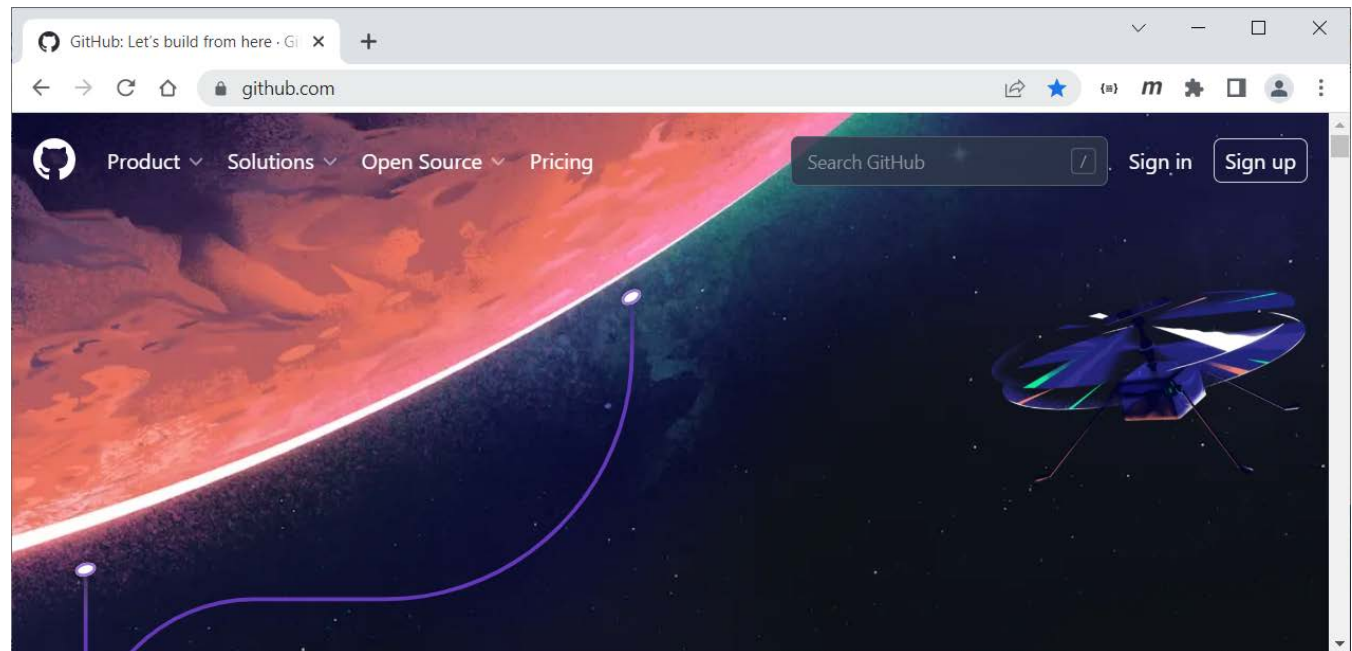


Git Hub

- What is Git Hub?
 - Web site for Git
 - Control version repository

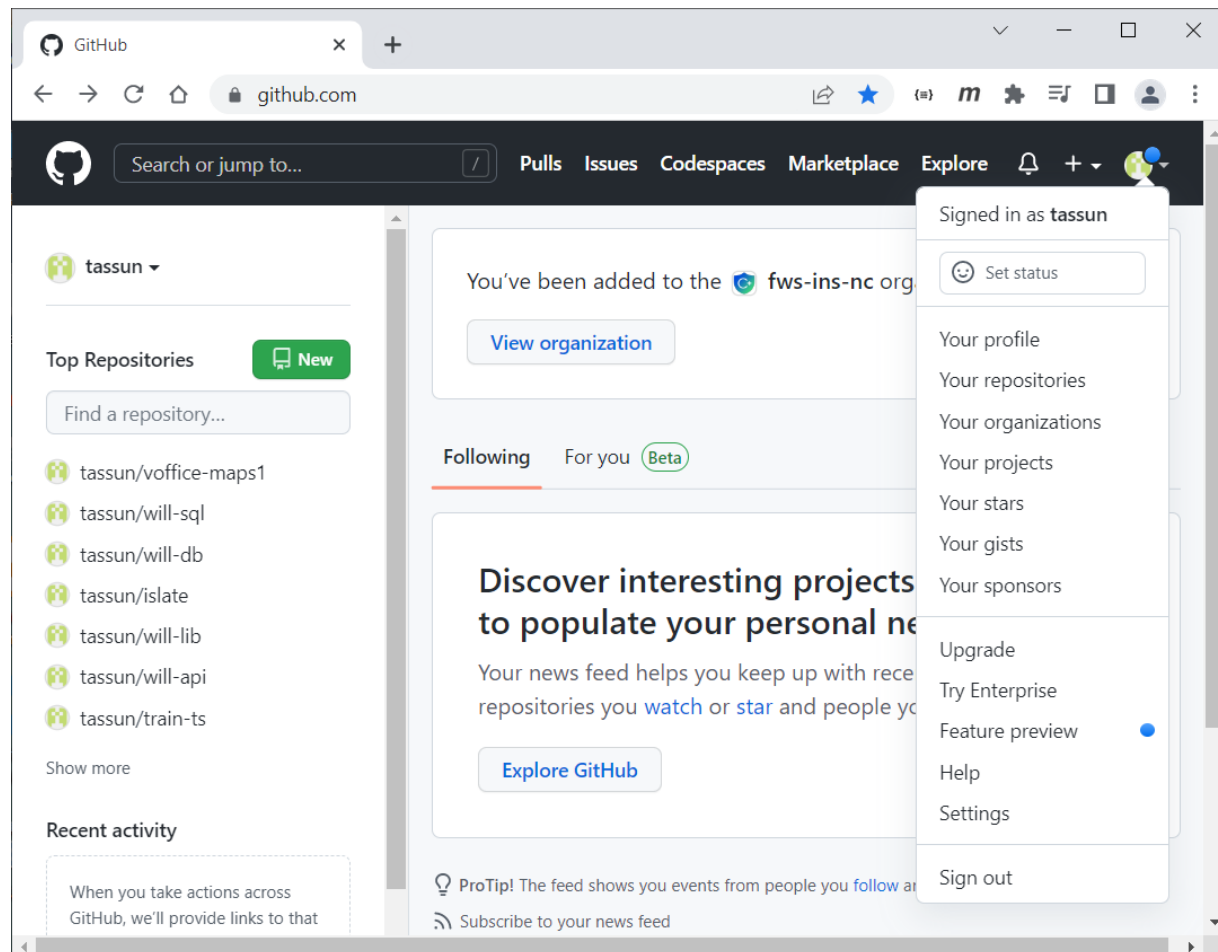
Git Hub

- How to use Git Hub?
 - Go to <https://github.com>
 - Create account (Sign up)



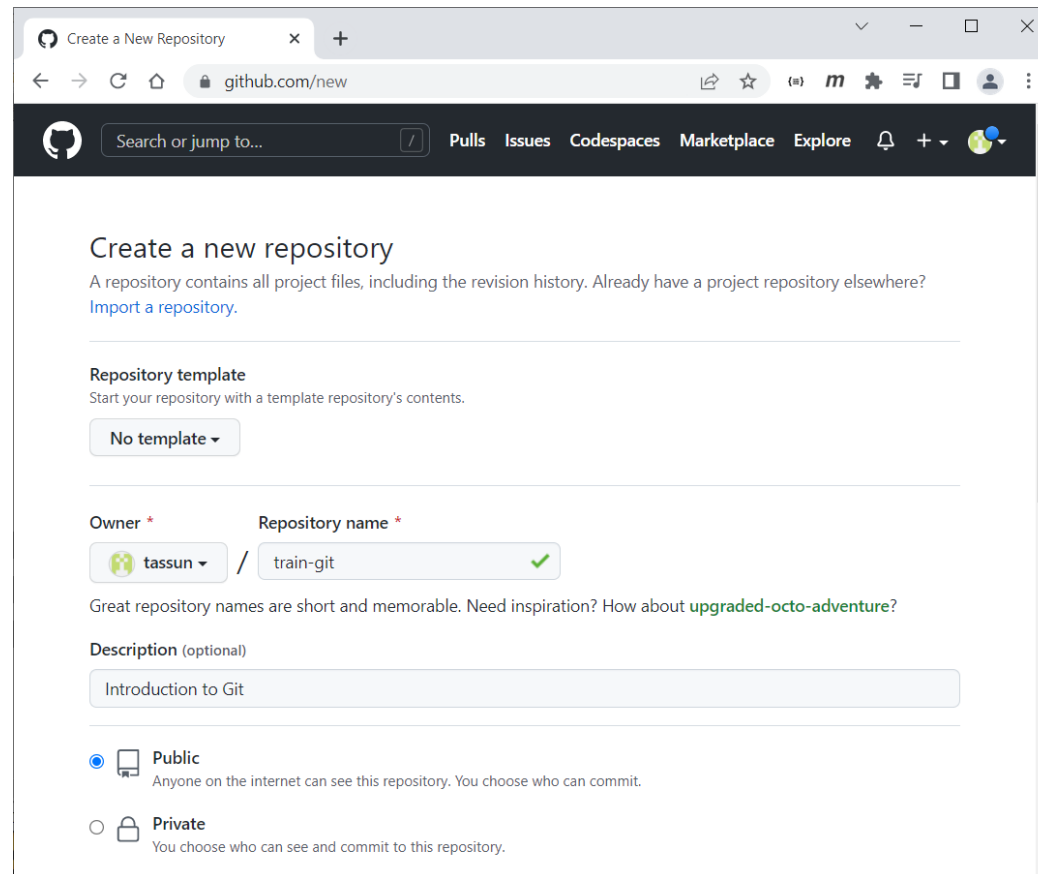
Git Hub

- How to use Git Hub?



Git Hub

- How to use Git Hub?
 - Create repository



The screenshot shows the GitHub 'Create a New Repository' page. The browser address bar shows 'github.com/new'. The page has a dark header with the GitHub logo, a search bar, and navigation links: Pulls, Issues, Codespaces, Marketplace, Explore, and a user profile icon. The main content area is titled 'Create a new repository' and includes a sub-header 'Repository template' with a 'No template' dropdown. Below this are fields for 'Owner' (tassun) and 'Repository name' (train-git), which is marked with a green checkmark. A description field contains 'Introduction to Git'. At the bottom, there are radio buttons for 'Public' (selected) and 'Private' repository visibility.

Create a New Repository

github.com/new

Search or jump to...

Pulls Issues Codespaces Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner * Repository name *

tassun / train-git ✓

Great repository names are short and memorable. Need inspiration? How about [upgraded-octo-adventure?](#)

Description (optional)

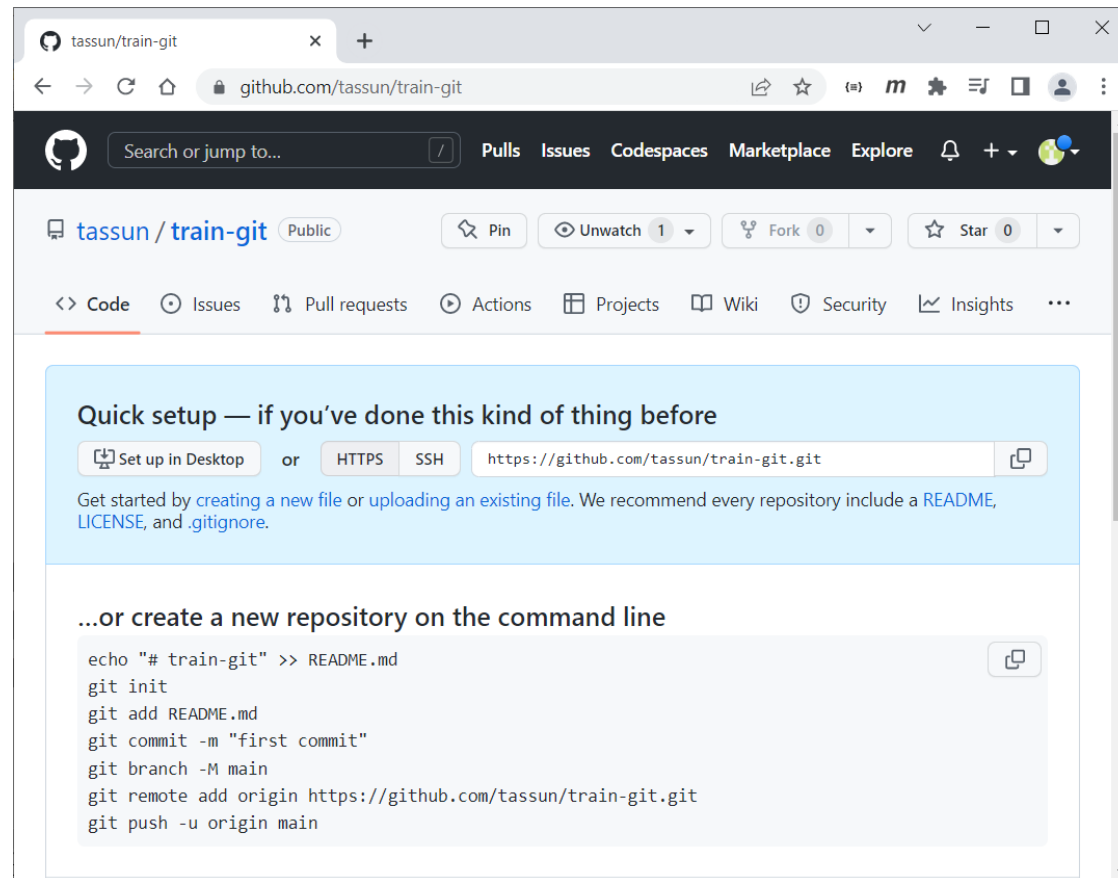
Introduction to Git

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

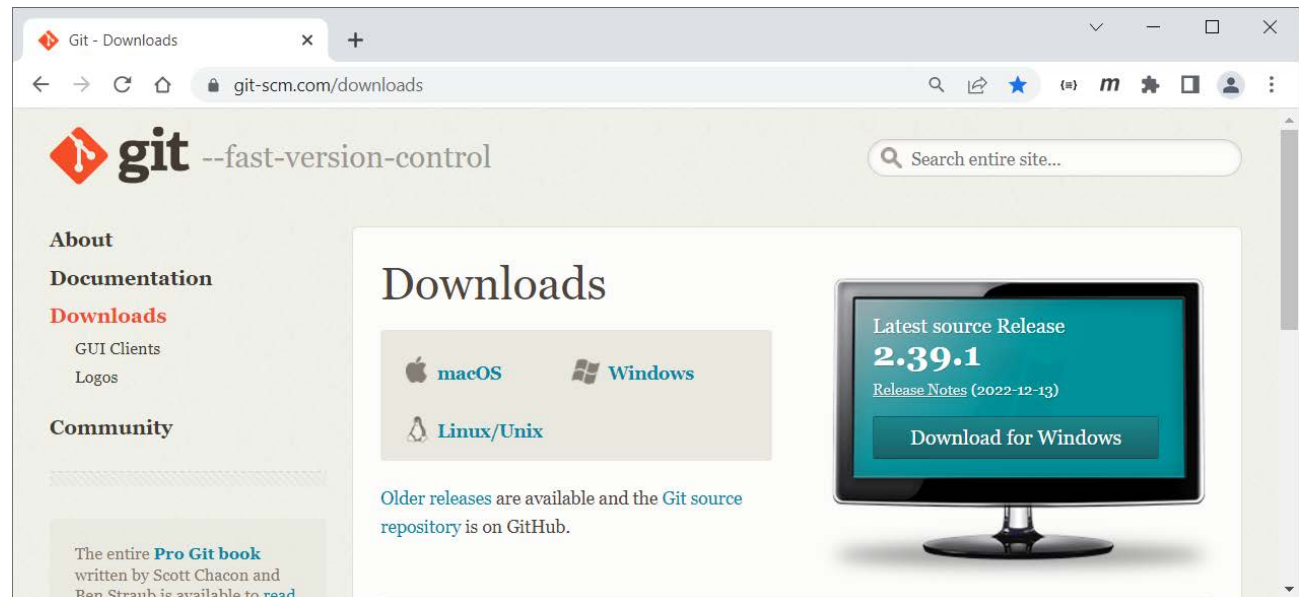
Git Hub

- How to use Git Hub?
 - Create repository



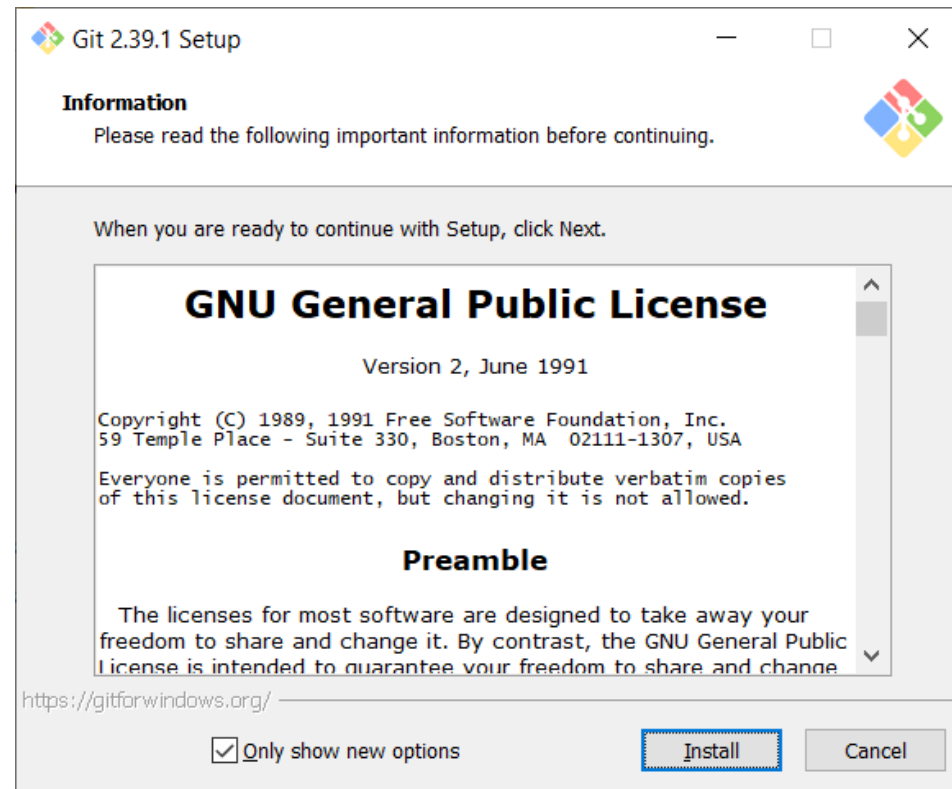
Git CLI

- How to use Git?
 - Go to <https://git-scm.com/>
 - Download & Install
 - <https://git-scm.com/downloads>



Git CLI

- Git Installation
 - windows



Git CLI

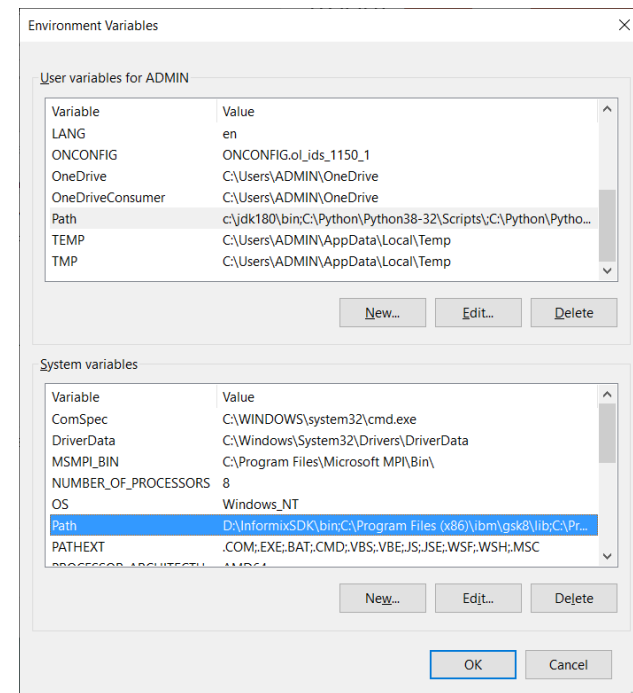
- Git Installation

- windows portable

- Settings -> Advanced system settings

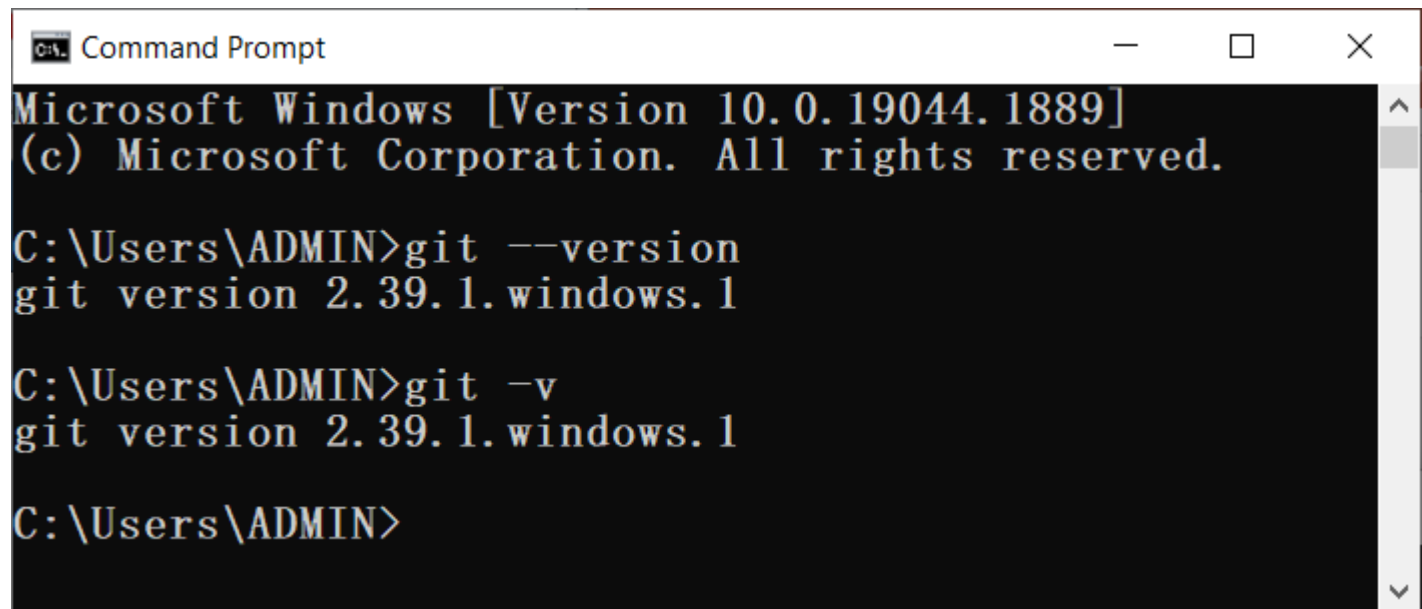
- System Properties -> Advanced -> Environment Variables

- set PATH



Git CLI

- Git Command
 - `git --version` or `git -v`
 - `git help` , `git help <command>`



```
Command Prompt
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>git --version
git version 2.39.1.windows.1

C:\Users\ADMIN>git -v
git version 2.39.1.windows.1

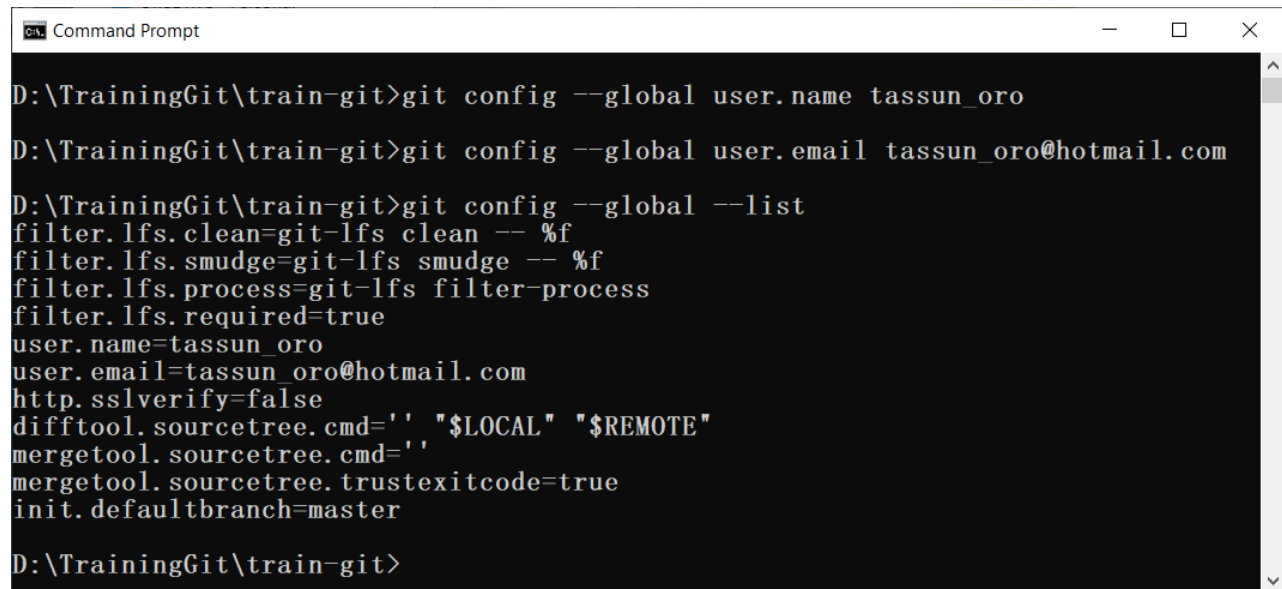
C:\Users\ADMIN>
```

Git CLI

- Git Command

- git config

- git config --global user.name your-name
 - git config --global user.email your-email
 - git config --global --list



```
Command Prompt

D:\TrainingGit\train-git>git config --global user.name tassun_oro

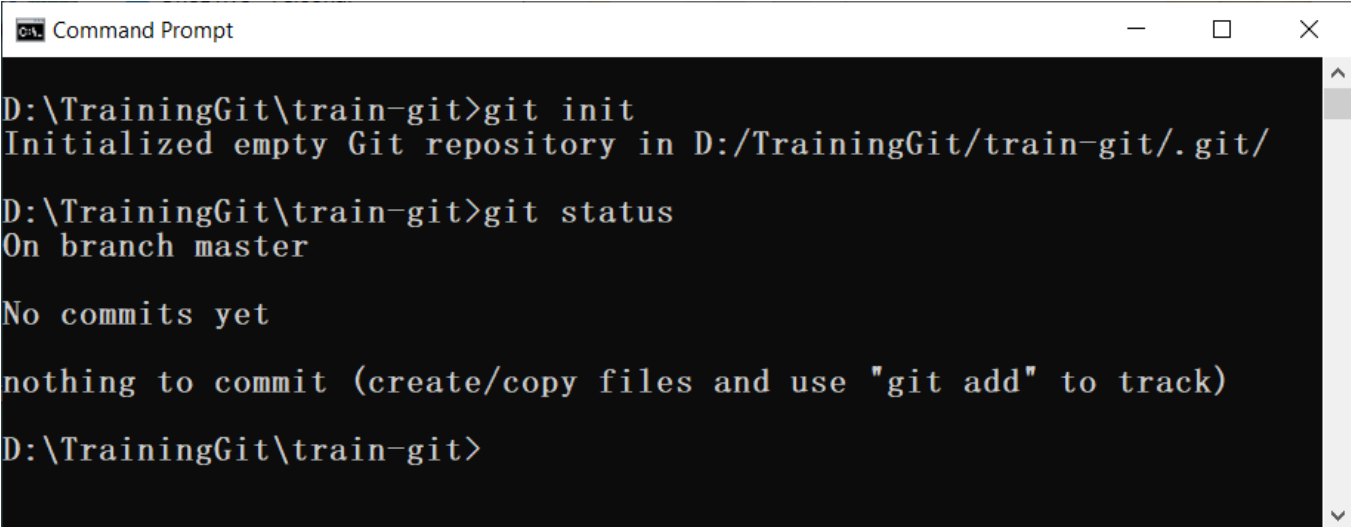
D:\TrainingGit\train-git>git config --global user.email tassun_oro@hotmail.com

D:\TrainingGit\train-git>git config --global --list
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=tassun_oro
user.email=tassun_oro@hotmail.com
http.sslverify=false
difftool.sourcetree.cmd='' "$LOCAL" "$REMOTE"
mergetool.sourcetree.cmd=''
mergetool.sourcetree.trustexitcode=true
init.defaultbranch=master

D:\TrainingGit\train-git>
```

Git CLI

- Git Command
 - git init
 - create .git folder store local changed
 - git status



```
Command Prompt

D:\TrainingGit\train-git>git init
Initialized empty Git repository in D:/TrainingGit/train-git/.git/

D:\TrainingGit\train-git>git status
On branch master

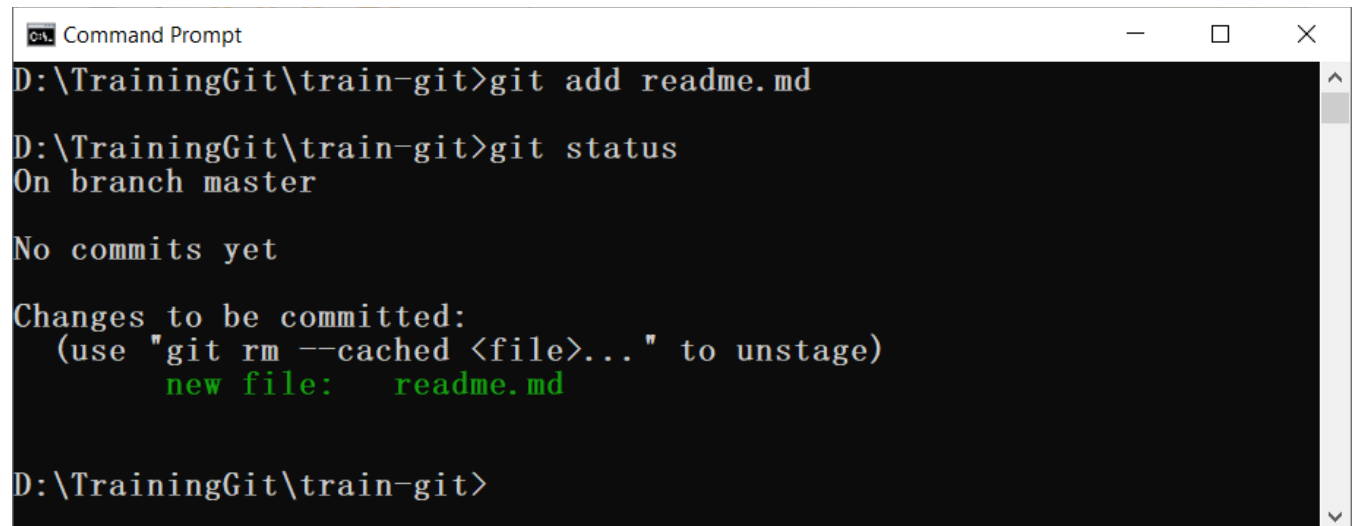
No commits yet

nothing to commit (create/copy files and use "git add" to track)

D:\TrainingGit\train-git>
```


Git CLI

- Git Command
 - git add
 - git add file-name
 - git add . or git add *.txt
 - git status



```
Command Prompt
D:\TrainingGit\train-git>git add readme.md

D:\TrainingGit\train-git>git status
On branch master

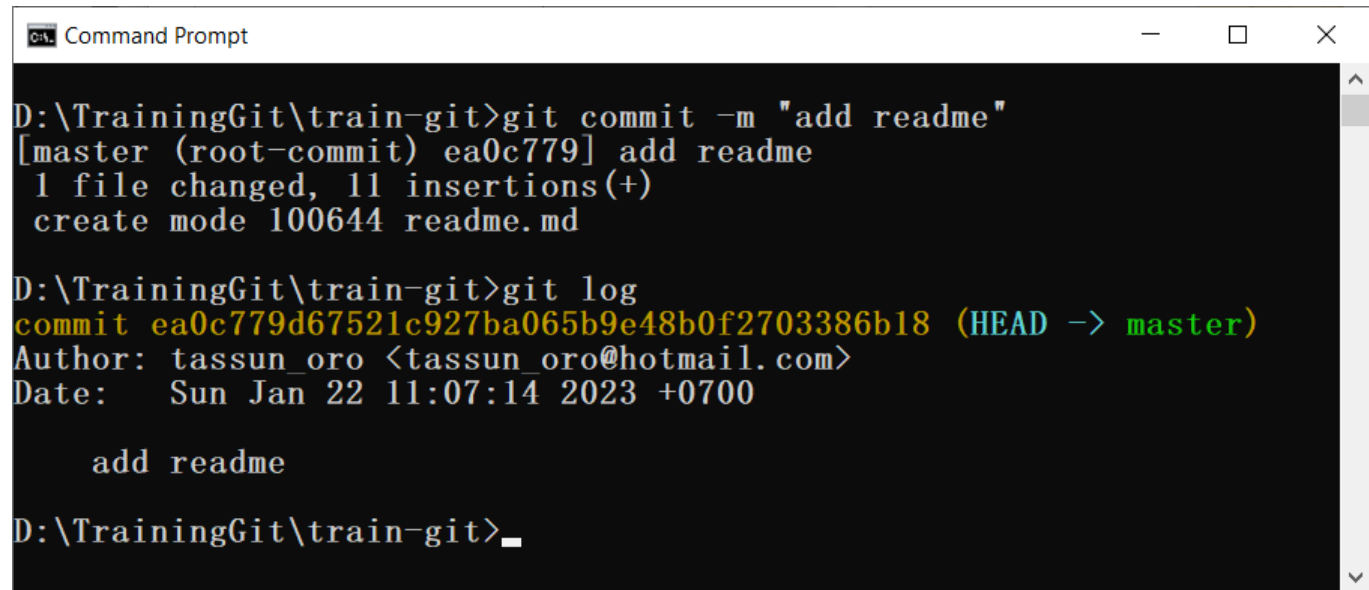
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   readme.md

D:\TrainingGit\train-git>
```

Git CLI

- Git Command
 - git commit
 - git commit -m your-messages
 - git log



```
Command Prompt

D:\TrainingGit\train-git>git commit -m "add readme"
[master (root-commit) ea0c779] add readme
1 file changed, 11 insertions(+)
create mode 100644 readme.md

D:\TrainingGit\train-git>git log
commit ea0c779d67521c927ba065b9e48b0f2703386b18 (HEAD -> master)
Author: tassun_oro <tassun_oro@hotmail.com>
Date:   Sun Jan 22 11:07:14 2023 +0700

    add readme

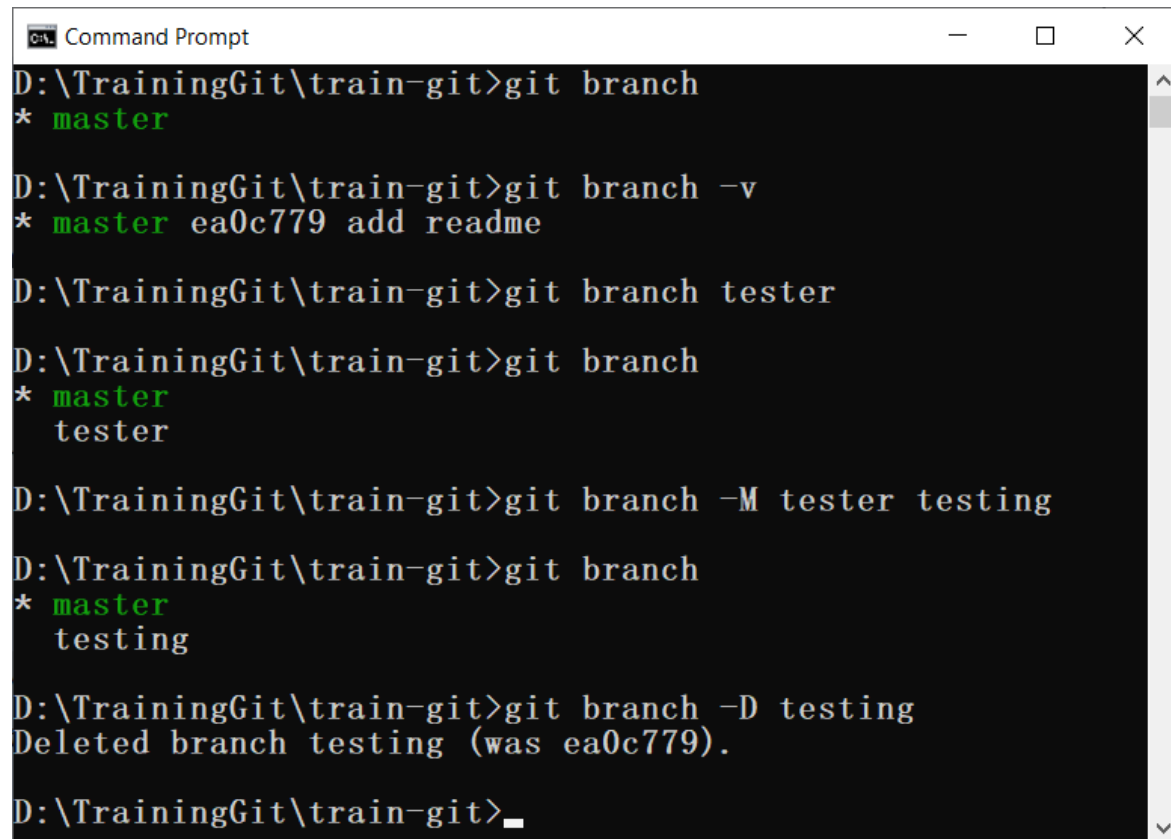
D:\TrainingGit\train-git>_
```

Git CLI

- Git Command
 - git branch
 - create
 - git branch branch-name
 - move/rename
 - git branch -M new-branch-name
 - delete
 - git branch -D branch-name
 - switch/change
 - git checkout branch-name

Git CLI

- Git Command
 - git branch



```
Command Prompt
D:\TrainingGit\train-git>git branch
* master

D:\TrainingGit\train-git>git branch -v
* master ea0c779 add readme

D:\TrainingGit\train-git>git branch tester

D:\TrainingGit\train-git>git branch
* master
  tester

D:\TrainingGit\train-git>git branch -M tester testing

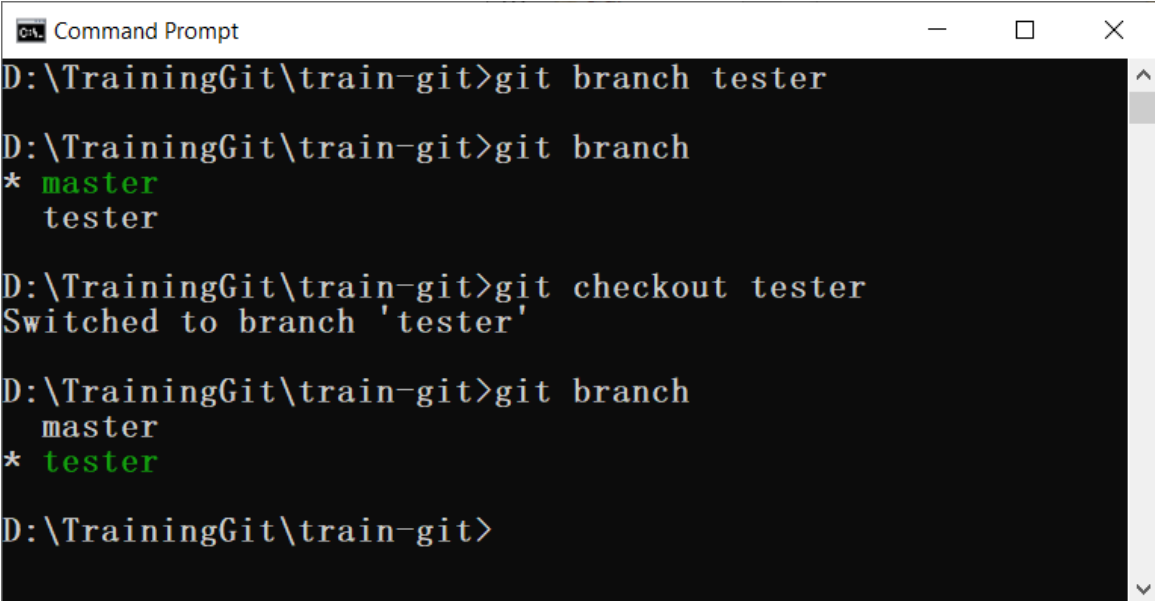
D:\TrainingGit\train-git>git branch
* master
  testing

D:\TrainingGit\train-git>git branch -D testing
Deleted branch testing (was ea0c779).

D:\TrainingGit\train-git>
```

Git CLI

- Git Command
 - git branch



```
Git_ Command Prompt
D:\TrainingGit\train-git>git branch tester

D:\TrainingGit\train-git>git branch
* master
  tester

D:\TrainingGit\train-git>git checkout tester
Switched to branch 'tester'

D:\TrainingGit\train-git>git branch
  master
* tester

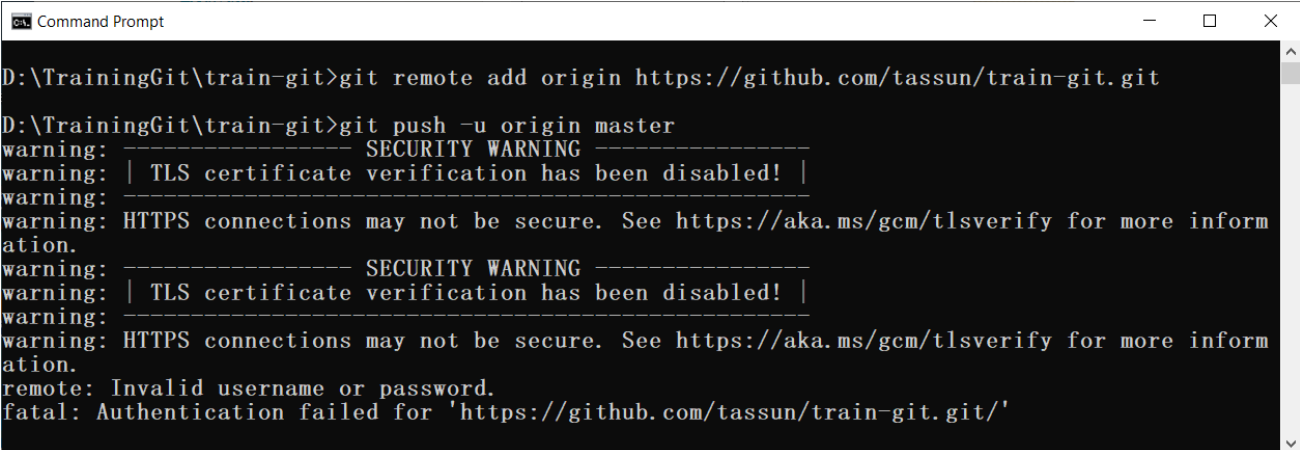
D:\TrainingGit\train-git>
```

Git CLI

- Git Command

- git push

- git remote add origin <https://github.com/username/repo-name.git>
 - git push -u origin master



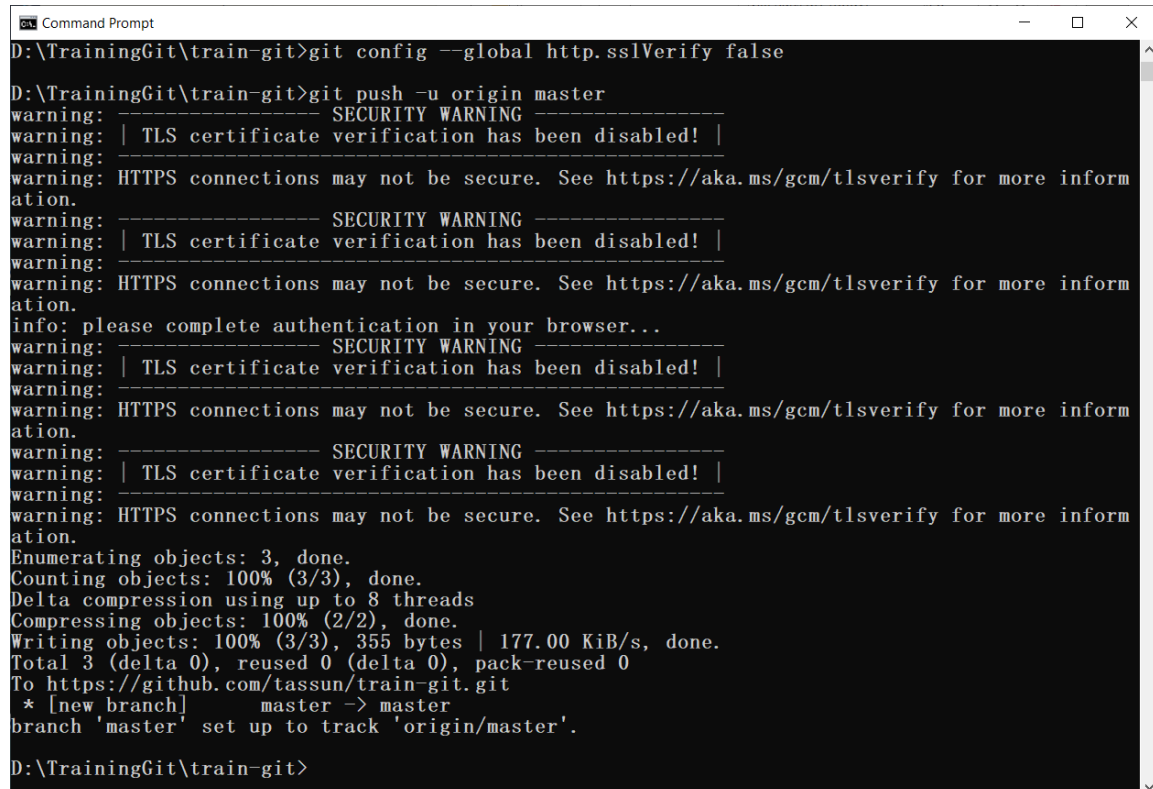
```
Command Prompt
D:\TrainingGit\train-git>git remote add origin https://github.com/tassun/train-git.git
D:\TrainingGit\train-git>git push -u origin master
warning: SECURITY WARNING
warning: | TLS certificate verification has been disabled! |
warning: -----
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more information.
warning: SECURITY WARNING
warning: | TLS certificate verification has been disabled! |
warning: -----
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more information.
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/tassun/train-git.git/'
```

Git CLI

- Git Command

- git push

- git config --global http.sslVerify false



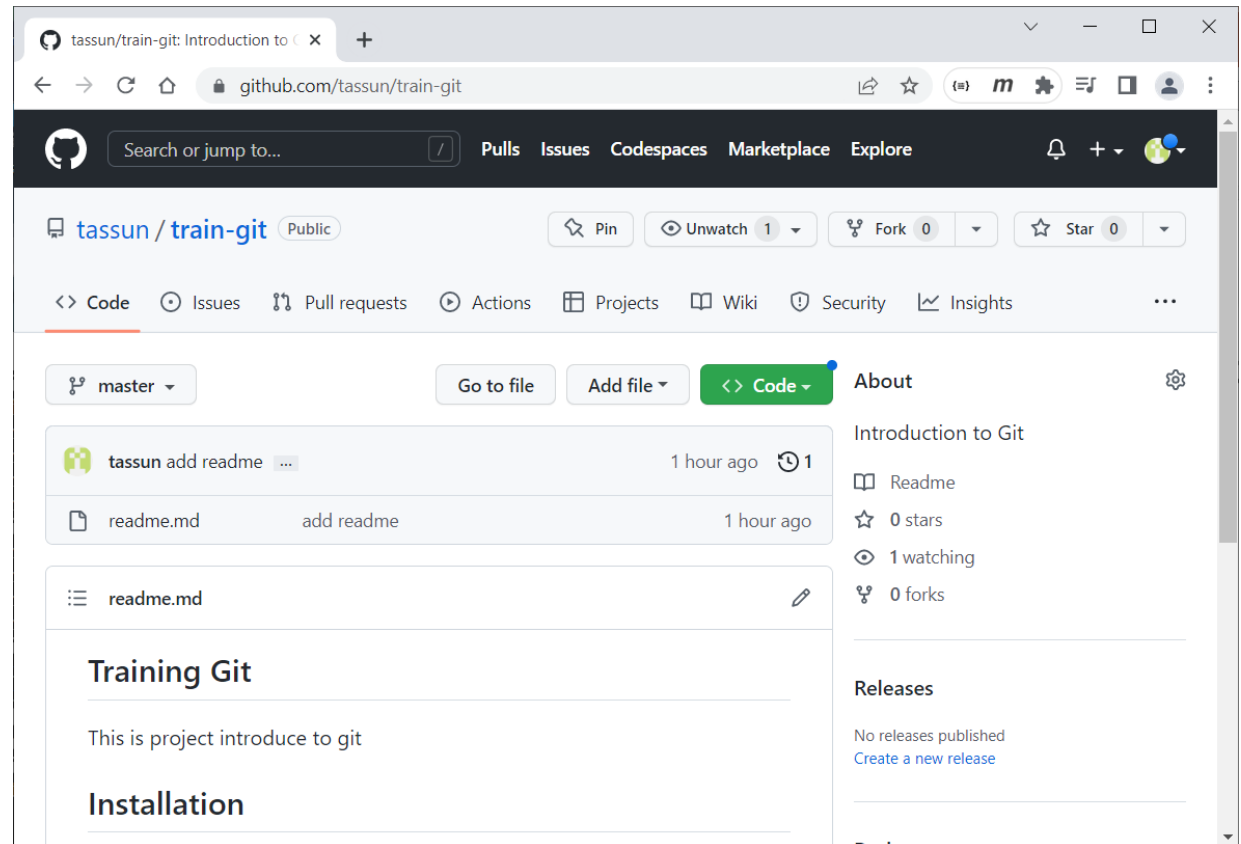
```
Command Prompt
D:\TrainingGit\train-git>git config --global http.sslVerify false

D:\TrainingGit\train-git>git push -u origin master
warning: SECURITY WARNING
warning: | TLS certificate verification has been disabled! |
warning: -----
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more inform
ation.
warning: SECURITY WARNING
warning: | TLS certificate verification has been disabled! |
warning: -----
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more inform
ation.
info: please complete authentication in your browser...
warning: SECURITY WARNING
warning: | TLS certificate verification has been disabled! |
warning: -----
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more inform
ation.
warning: SECURITY WARNING
warning: | TLS certificate verification has been disabled! |
warning: -----
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more inform
ation.
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 355 bytes | 177.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/tassun/train-git.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

D:\TrainingGit\train-git>
```

Git CLI

- Git Command
 - git push




Git CLI

- Git Command

- git clone

- git clone <https://github.com/user-name/repo-name.git>

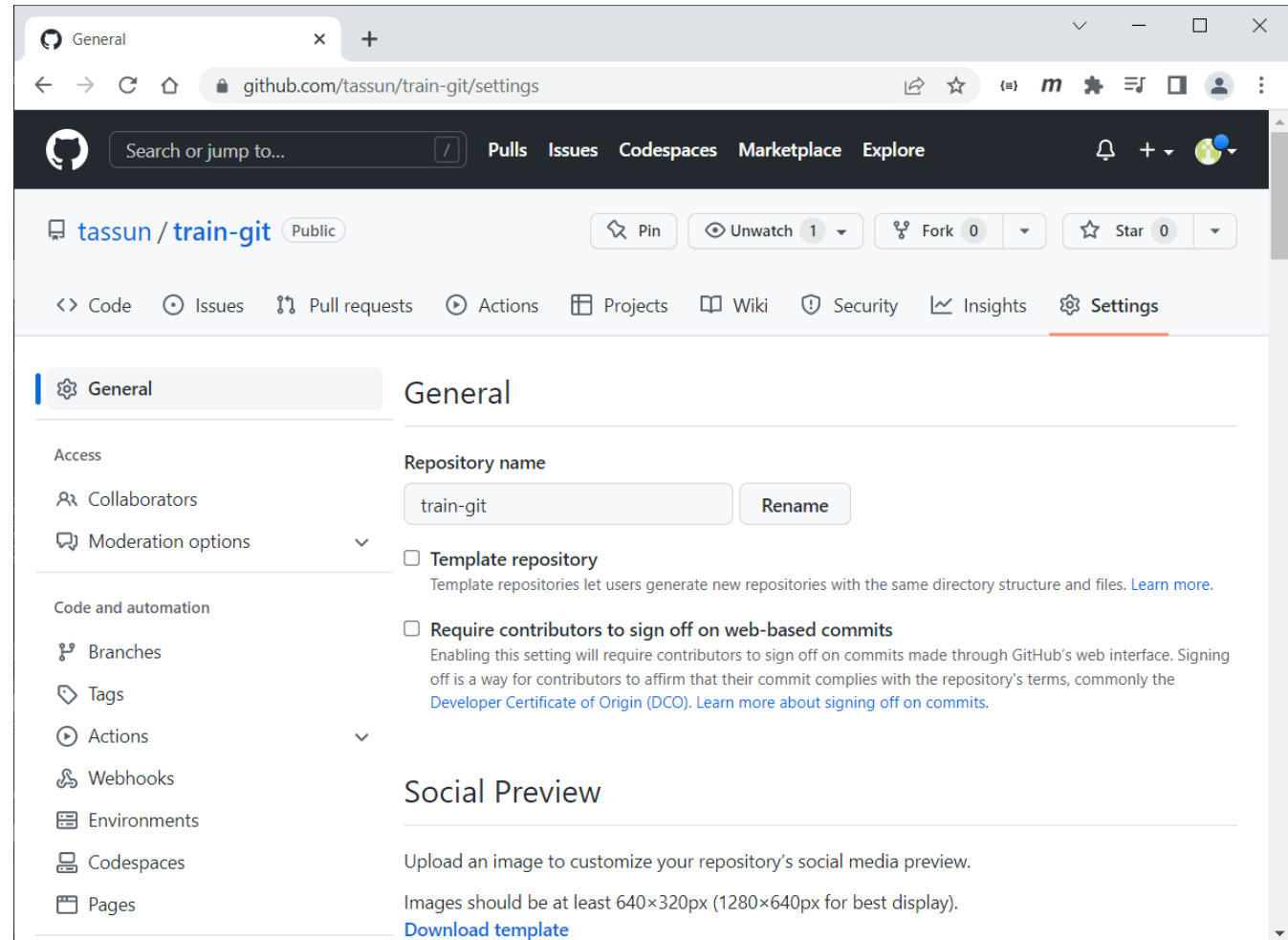


```
Git Command Prompt
D:\TrainingGit\clone>git clone https://github.com/tassun/train-git.git
Cloning into 'train-git'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

D:\TrainingGit\clone>_
```

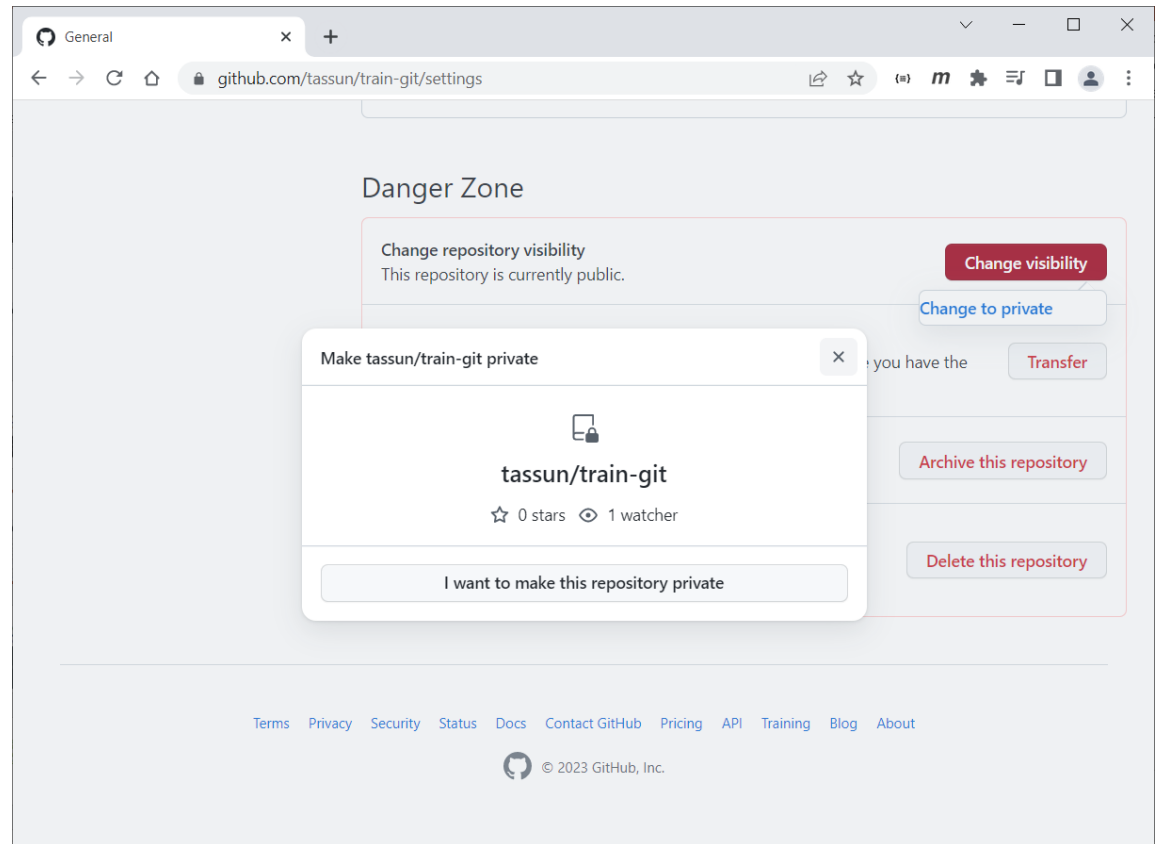
Git CLI

- GitHub Setting



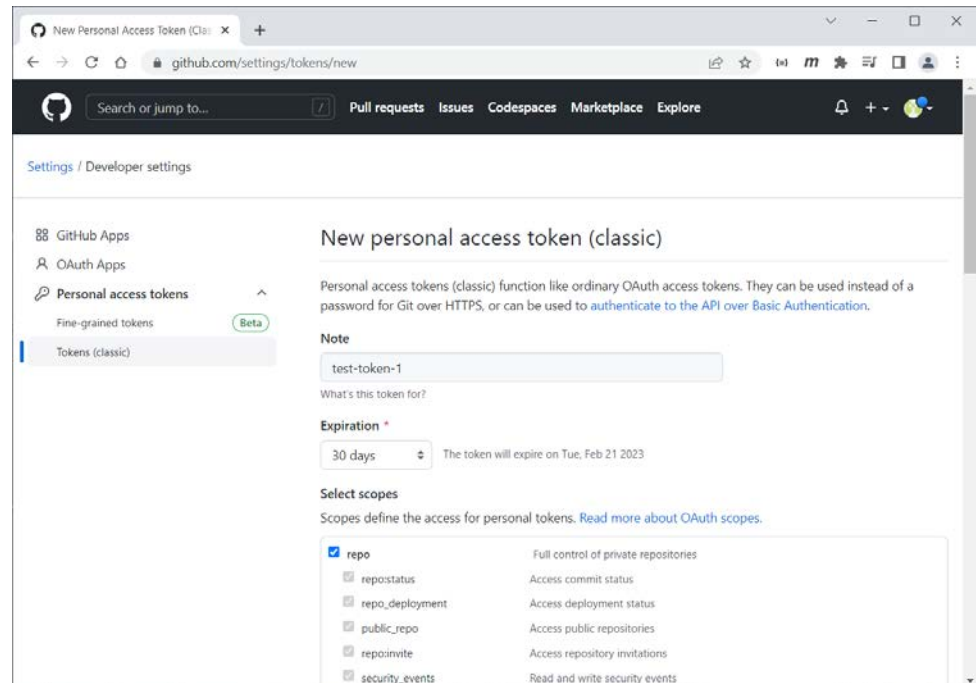
Git CLI

- GitHub Setting
 - try to change repository private/public



Git CLI

- Create Personal Access Token on GitHub
 - Settings -> Developer Settings
 - Personal access tokens -> Tokens (classic)
 - Generate new token



The screenshot shows the GitHub web interface for creating a new personal access token. The browser address bar shows 'github.com/settings/tokens/new'. The page title is 'New personal access token (classic)'. The left sidebar shows the navigation menu with 'Personal access tokens' selected, and 'Tokens (classic)' highlighted. The main content area includes a text input for the token name (containing 'test-token-1'), an 'Expiration' dropdown set to '30 days' (with a note that the token will expire on Tue, Feb 21 2023), and a 'Select scopes' section. The 'repo' scope is checked, and its sub-scopes are listed: 'repo:status' (Access commit status), 'repo:deployment' (Access deployment status), 'public_repo' (Access public repositories), 'repo:invite' (Access repository invitations), and 'security_events' (Read and write security events).

New Personal Access Token (Classic) x +

github.com/settings/tokens/new

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

Settings / Developer settings

GitHub Apps OAuth Apps

Personal access tokens

Fine-grained tokens (Beta)

Tokens (classic)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

test-token-1

What's this token for?

Expiration *

30 days The token will expire on Tue, Feb 21 2023

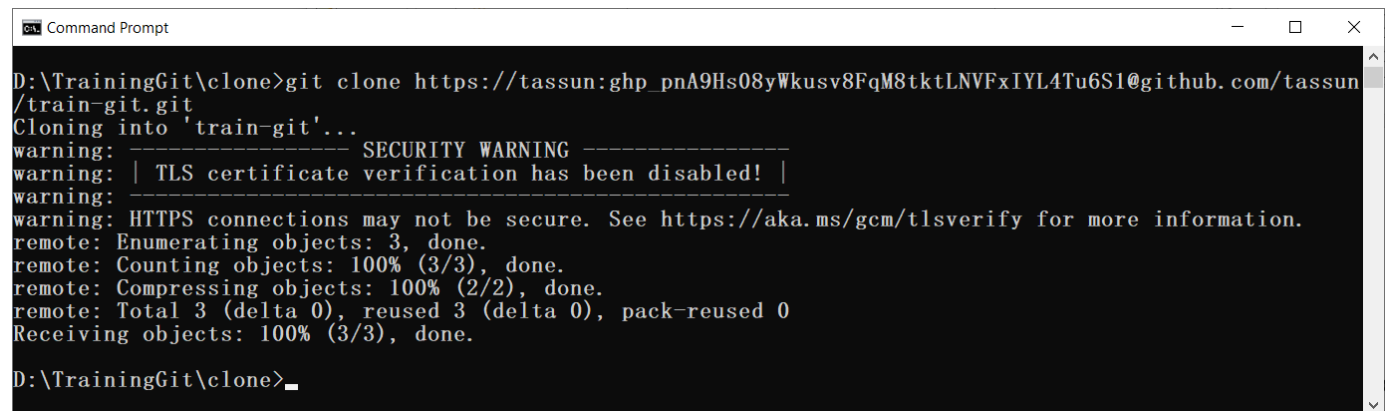
Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo:deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events

Git CLI

- Git Command
 - git hub token
 - git clone https://user-name:github-token@github.com/user-name/repo-name.git



```
Command Prompt
D:\TrainingGit\clone>git clone https://tassun:ghp_pnA9Hs08yWkusv8FqM8tkLNVFxiYL4Tu6S1@github.com/tassun
/train-git.git
Cloning into 'train-git'...
warning: ----- SECURITY WARNING -----
warning: | TLS certificate verification has been disabled! |
warning: -----
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more information.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

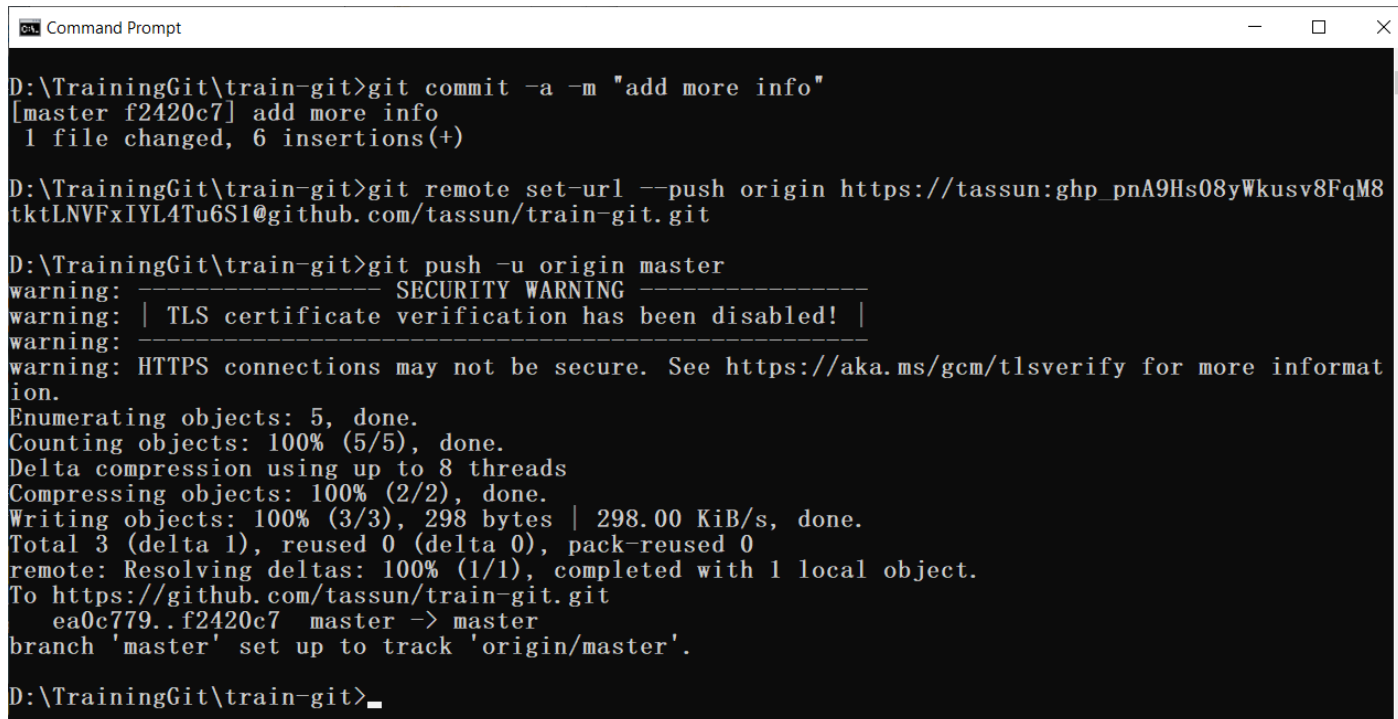
D:\TrainingGit\clone>_
```

Git CLI

- Git Command

- git hub token

- try to edit readme.md then commit & push changed



```
Command Prompt

D:\TrainingGit\train-git>git commit -a -m "add more info"
[master f2420c7] add more info
1 file changed, 6 insertions(+)

D:\TrainingGit\train-git>git remote set-url --push origin https://tassun:ghp_pnA9Hs08yWkusv8FqM8
tktLNVFxiYL4Tu6S1@github.com/tassun/train-git.git

D:\TrainingGit\train-git>git push -u origin master
warning: ----- SECURITY WARNING -----
warning: | TLS certificate verification has been disabled! |
warning: -----
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more informat
ion.
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 298.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/tassun/train-git.git
   ea0c779..f2420c7  master -> master
branch 'master' set up to track 'origin/master'.

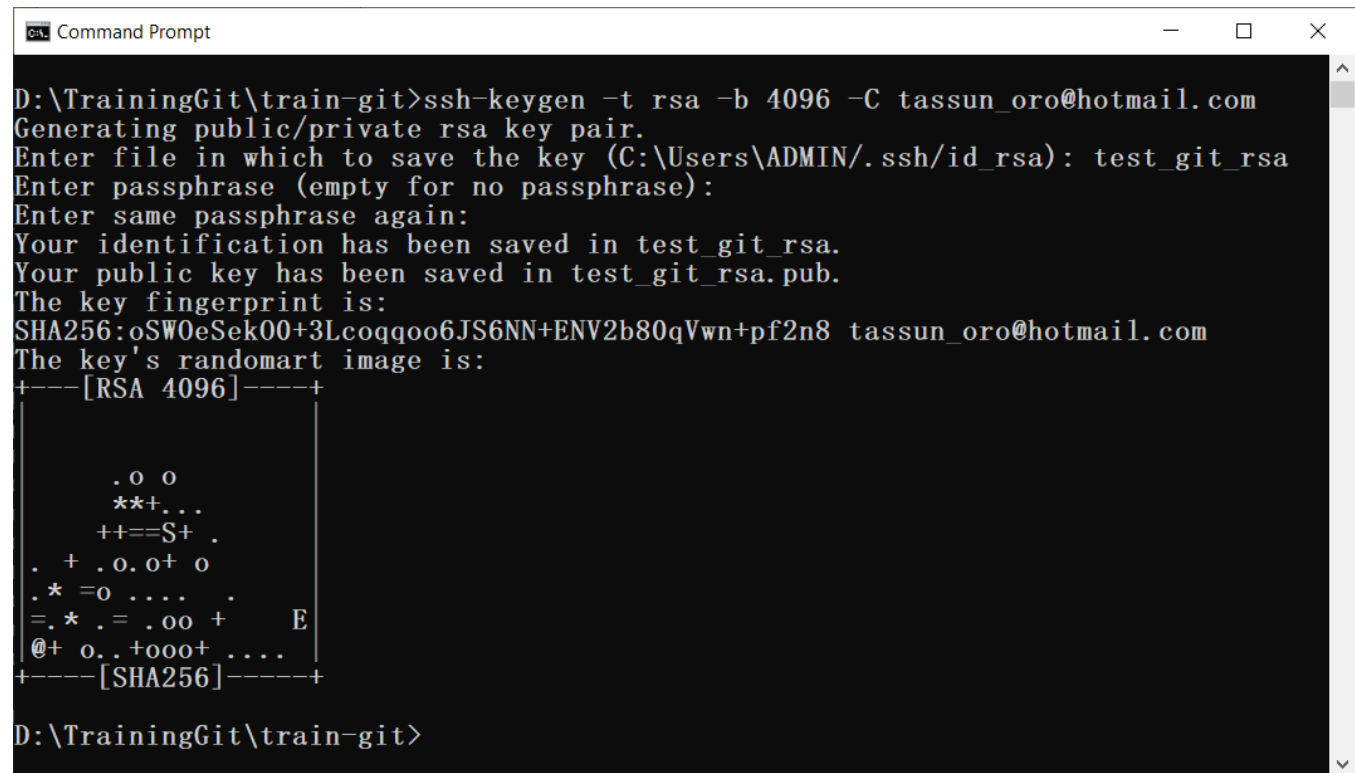
D:\TrainingGit\train-git>_
```

Git CLI

- Git Command

- git ssh key

- ssh-keygen -t rsa -b 4096 -C "email@example.com"

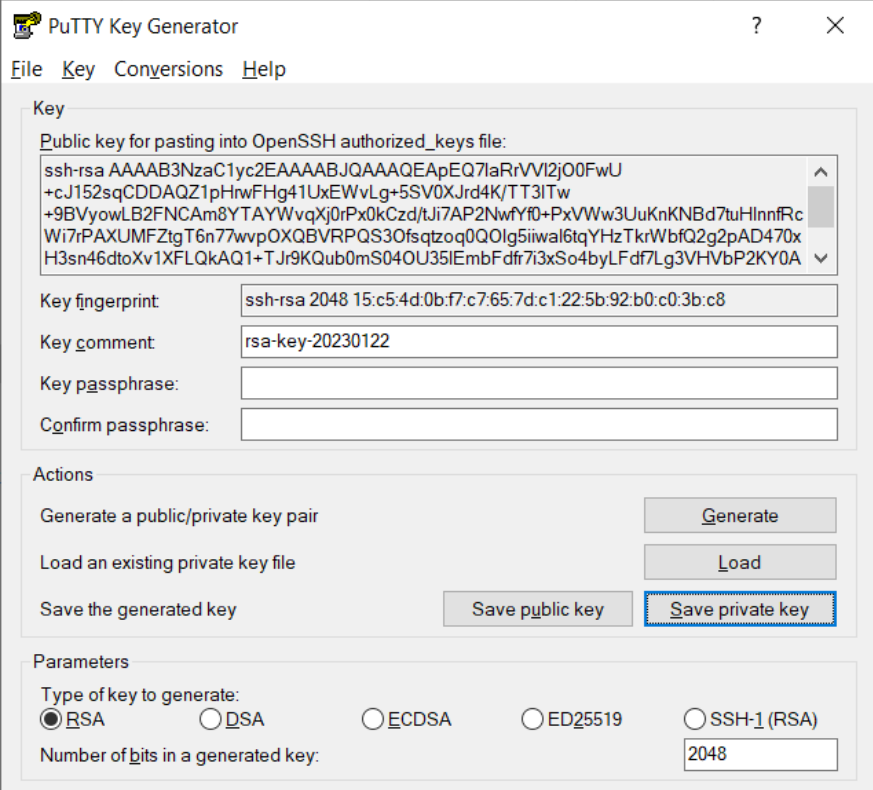


```
Command Prompt
D:\TrainingGit\train-git>ssh-keygen -t rsa -b 4096 -C tassun_oro@hotmail.com
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\ADMIN/.ssh/id_rsa): test_git_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in test_git_rsa.
Your public key has been saved in test_git_rsa.pub.
The key fingerprint is:
SHA256:oSW0eSek00+3Lcoqqoo6JS6NN+ENV2b80qVwn+pf2n8 tassun_oro@hotmail.com
The key's randomart image is:
+---[RSA 4096]---+
  . o o
    **+...
  ++==S+ .
. + .o.o+ o
. * =o ....
=. * . = .oo +   E
@+ o..+ooo+ ....
+---[SHA256]---+

D:\TrainingGit\train-git>
```

Git CLI

- Git Command
 - git ssh key
 - putty key generator



The screenshot shows the PuTTY Key Generator application window. The 'Key' section displays a generated SSH key in the 'Public key for pasting into OpenSSH authorized_keys file:' field. The key fingerprint is shown as 'ssh-rsa 2048 15:c5:4d:0b:f7:c7:65:7d:c1:22:5b:92:b0:c0:3b:c8'. The key comment is 'rsa-key-20230122'. The 'Actions' section includes buttons for 'Generate', 'Load', 'Save public key', and 'Save private key'. The 'Parameters' section shows the 'Type of key to generate' set to 'RSA' and the 'Number of bits in a generated key' set to '2048'.

PuTTY Key Generator

File Key Conversions Help

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAAQEApcEQ7laRrVVI2jO0FwU
+cJ152sqCDDAQZ1pHrwFHg41UxEWvLg+5SV0XJrd4K/TT3ITw
+9BVyowLB2FNCAm8YTAYWvqXj0rPx0kCzd/tJi7AP2NwfYf0+PxVWw3UuKnKNBd7tuHlnnfRc
Wi7rPAXUMFZtgT6n77wvpOXQBVPRQS3Ofsqtoq0Q0Ig5iwal6tqYHzTkrWbfQ2g2pAD470x
H3sn46dtoXv1XFLQkAQ1+TJr9KQub0mS04OU35IEmbFdf7i3xSo4byLFdf7Lg3VHVbP2KY0A
```

Key fingerprint: ssh-rsa 2048 15:c5:4d:0b:f7:c7:65:7d:c1:22:5b:92:b0:c0:3b:c8

Key comment: rsa-key-20230122

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair Generate

Load an existing private key file Load

Save the generated key Save public key Save private key

Parameters

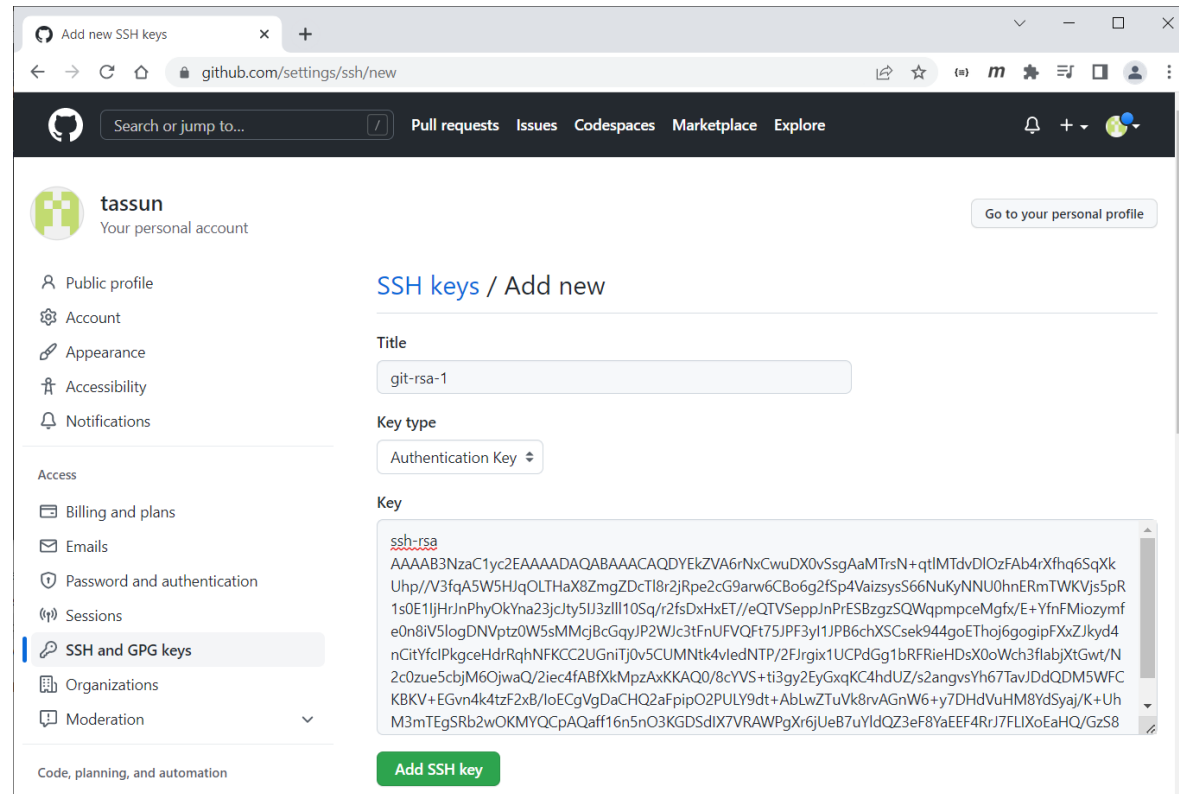
Type of key to generate:

☒ RSA ☐ DSA ☐ ECDSA ☐ ED25519 ☐ SSH-1 (RSA)

Number of bits in a generated key: 2048

Git CLI

- Add new SSH keys on GitHub
 - Settings -> SSH and GPG keys
 - SSH keys -> New SSH key



The screenshot shows the GitHub web interface for adding a new SSH key. The browser address bar shows 'github.com/settings/ssh/new'. The page title is 'Add new SSH keys'. The user's profile is 'tassun'. The left sidebar shows the 'SSH and GPG keys' section selected. The main content area has the heading 'SSH keys / Add new'. There are three input fields: 'Title' with the value 'git-rsa-1', 'Key type' set to 'Authentication Key', and a large 'Key' text area containing a long RSA key string. A green 'Add SSH key' button is at the bottom right.

Add new SSH keys

github.com/settings/ssh/new

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

tassun
Your personal account

Go to your personal profile

Public profile
Account
Appearance
Accessibility
Notifications

Access

Billing and plans
Emails
Password and authentication
Sessions

SSH and GPG keys

Organizations
Moderation

Code, planning, and automation

SSH keys / Add new

Title
git-rsa-1

Key type
Authentication Key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQADYEkZVA6rNxCwuDX0vSsgAaMTTrN+qtIMTdvdIOzFAB4rXfhq6SqXk
Uhp//V3fqA5W5HJqOLTHaX8ZmgZDcTl8r2jRpe2cG9arw6CBo6g2fSp4VaizsysS66NuKyNNU0hnERmTWKVjs5pR
1s0E1ljHrJnPhyOkYna23jcTy5lJ3zlll10Sqr/2fsDxHxET//eQTVSeppJnPrESBzgzSQWqmpceMgFx/E+YfnFMiozymf
e0n8iV5logDNVptz0W5sMMcJBcGqyJP2WJc3tFnUFVQFt75JPF3y11JPB6chXSCsek944goEThoj6gogipFXxZJkyd4
nCitYfcIPkgceHdRqhNFKCC2UGniTj0v5CUMNtk4vledNTP/2FJrgix1UCPdGg1bRFRieHDSX0oWch3flabjXtGwt/N
2c0zue5cbjM6OjwaQ/2iec4fABfXkMpzAxKKAQ0/8cYVS+ti3gy2EyGxqKC4hdUZ/s2angvsYh67TavjDdQDM5WFC
KBKV+EGvn4k4tzF2xB/loECgVgDaCHQ2aFpipO2PULY9dt+AbLwZTuVk8rvAGnW6+y7DHDdVuHM8YdSyaj/K+Uh
M3mTEgSRb2wOKMYQCpAQaff16n5nO3KGDSDlX7VRAWPgXr6jUeB7uYldQZ3eF8YaEEF4Rrj7FLIXoEaHQ/GzS8
```

Add SSH key

Git CLI

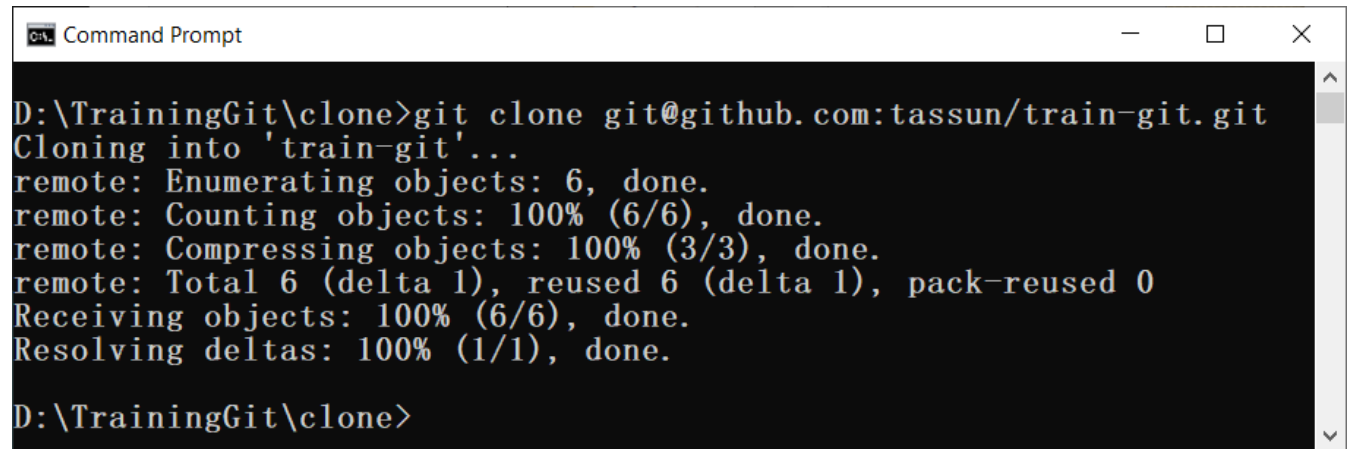
- Git Command
 - git ssh key testing



```
Git Command Prompt
D:\TrainingGit\train-git>ssh -T git@github.com
Hi tassun! You've successfully authenticated, but GitHub does not
provide shell access.
D:\TrainingGit\train-git>_
```

Git CLI

- Git Command
 - git ssh key
 - try to clone with ssh



```
Command Prompt
D:\TrainingGit\clone>git clone git@github.com:tassun/train-git.git
Cloning into 'train-git'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 1), reused 6 (delta 1), pack-reused 0
Receiving objects: 100% (6/6), done.
Resolving deltas: 100% (1/1), done.

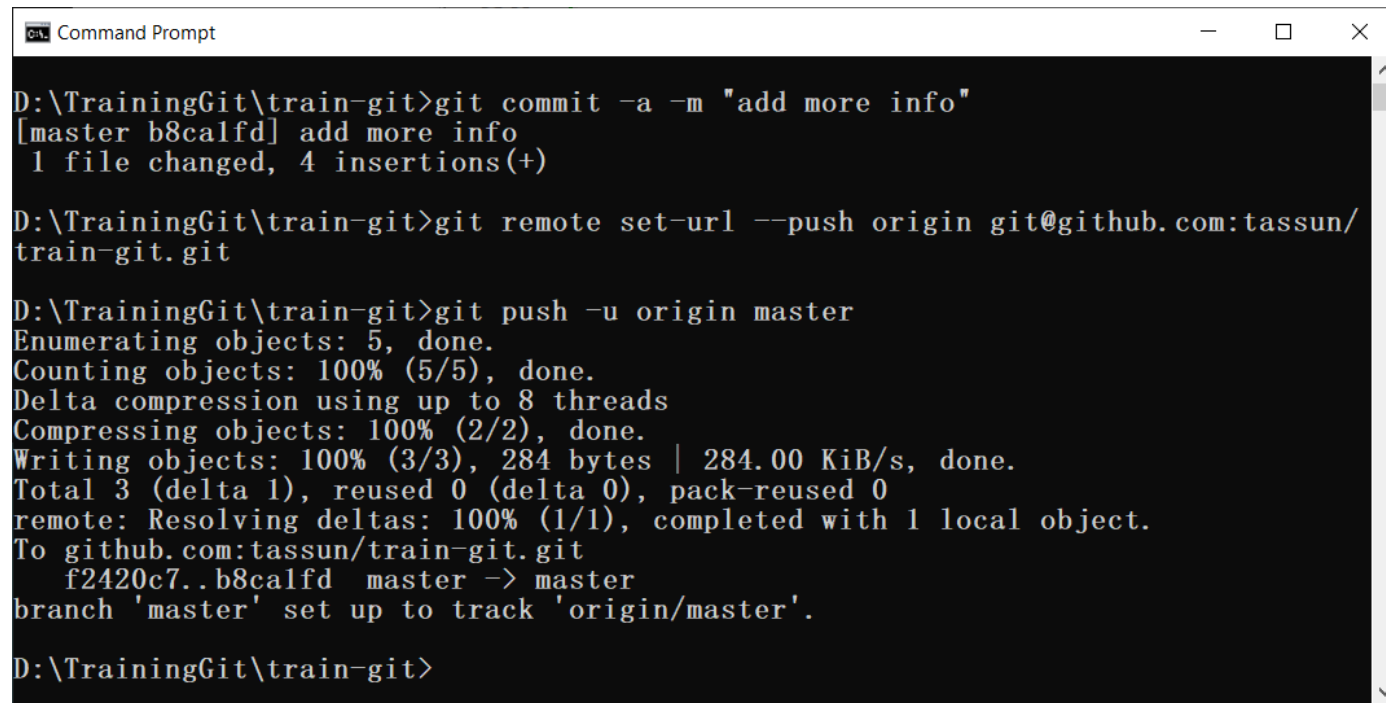
D:\TrainingGit\clone>
```

Git CLI

- Git Command

- git ssh key

- try to edit readme.md then commit & push changed



```
C:\> Command Prompt

D:\TrainingGit\train-git>git commit -a -m "add more info"
[master b8calfd] add more info
1 file changed, 4 insertions(+)

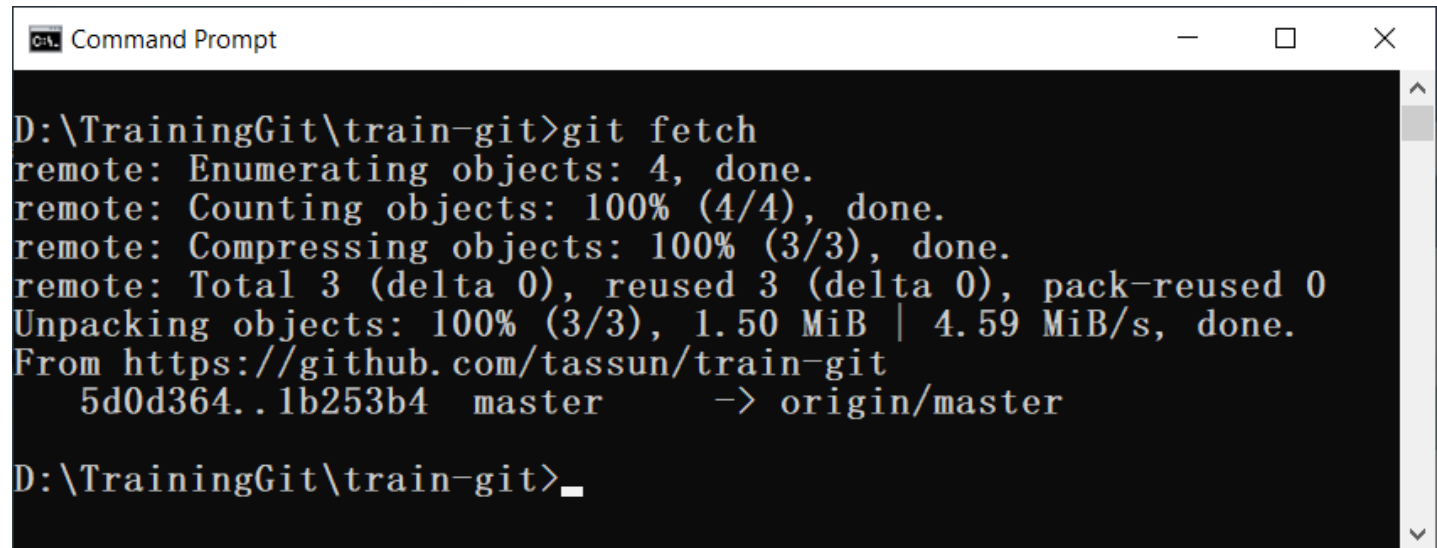
D:\TrainingGit\train-git>git remote set-url --push origin git@github.com:tassun/
train-git.git

D:\TrainingGit\train-git>git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 284 bytes | 284.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:tassun/train-git.git
 f2420c7..b8calfd master -> master
branch 'master' set up to track 'origin/master'.

D:\TrainingGit\train-git>
```

Git CLI

- Git Command
 - git fetch



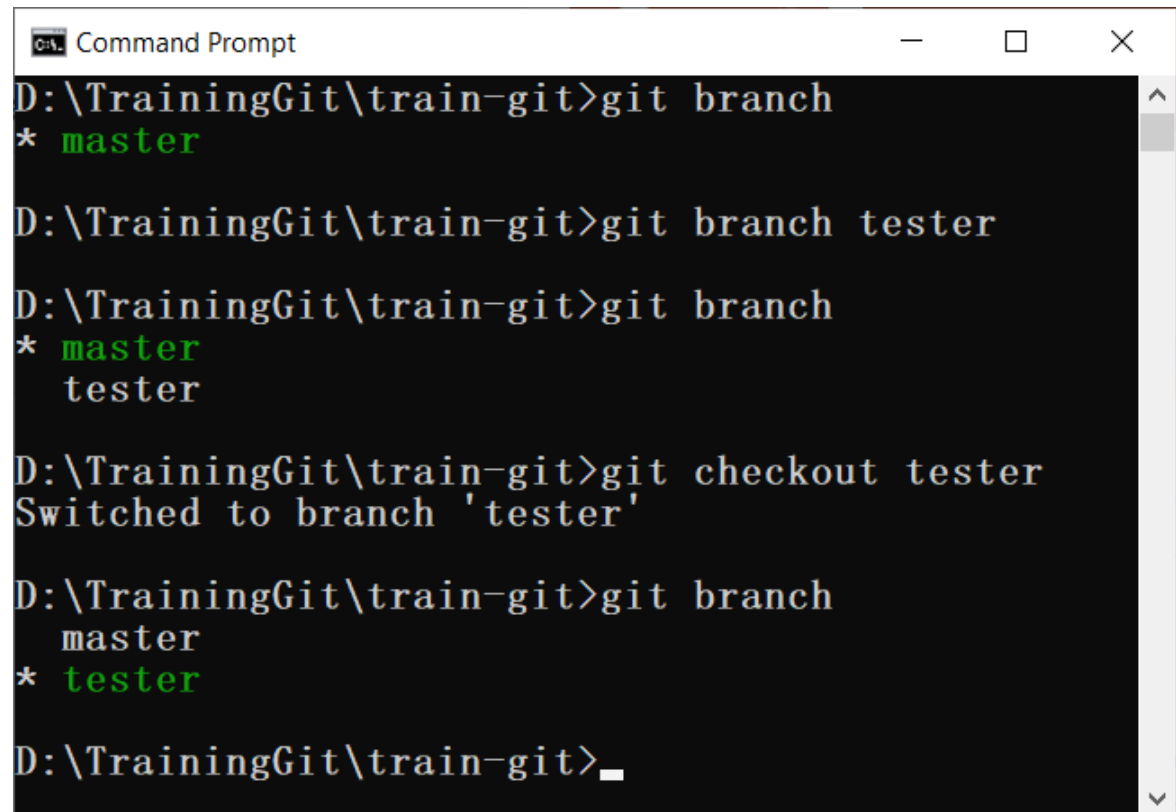
```
Command Prompt

D:\TrainingGit\train-git>git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.50 MiB | 4.59 MiB/s, done.
From https://github.com/tassun/train-git
    5d0d364..1b253b4  master    -> origin/master

D:\TrainingGit\train-git>
```

Git CLI

- Git Command
 - git merge



```
Command Prompt
D:\TrainingGit\train-git>git branch
* master

D:\TrainingGit\train-git>git branch tester

D:\TrainingGit\train-git>git branch
* master
  tester

D:\TrainingGit\train-git>git checkout tester
Switched to branch 'tester'

D:\TrainingGit\train-git>git branch
  master
* tester

D:\TrainingGit\train-git>_
```

Git CLI

- Git Command
 - git merge

```
Command Prompt
D:\TrainingGit\train-git>git add testme.txt

D:\TrainingGit\train-git>git commit -m "add test me"
[tester 8010551] add test me
1 file changed, 2 insertions(+)
create mode 100644 testme.txt

D:\TrainingGit\train-git>
```

```
Command Prompt
D:\TrainingGit\train-git>git branch
  master
* tester

D:\TrainingGit\train-git>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

D:\TrainingGit\train-git>git branch
* master
  tester

D:\TrainingGit\train-git>git merge tester
Updating 67bedb6..8010551
Fast-forward
 testme.txt | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 testme.txt

D:\TrainingGit\train-git>
```

Git CLI

- Git Command
 - git pull (fetch + merge)

```
Command Prompt

D:\TrainingGit\train-git>git status
On branch master
Your branch is behind 'origin/master' by 2 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

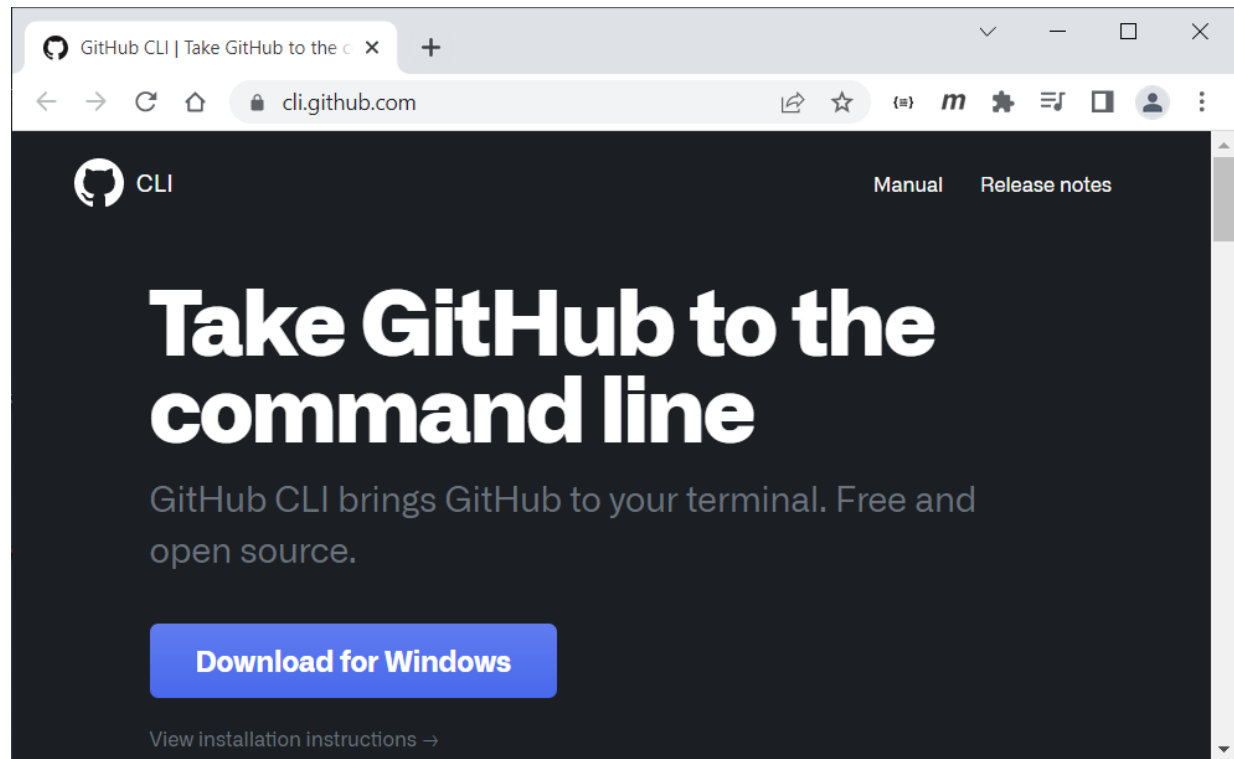
nothing to commit, working tree clean

D:\TrainingGit\train-git>git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), 94.53 KiB | 620.00 KiB/s, done.
From https://github.com/tassun/train-git
   1b253b4..2ea9ed7  master    -> origin/master
Updating b8calfd..2ea9ed7
Fast-forward
 Introduction to Git.pdf | Bin 0 -> 1767462 bytes
 readme.md               | 3 +++
2 files changed, 3 insertions(+)
create mode 100644 Introduction to Git.pdf

D:\TrainingGit\train-git>
```

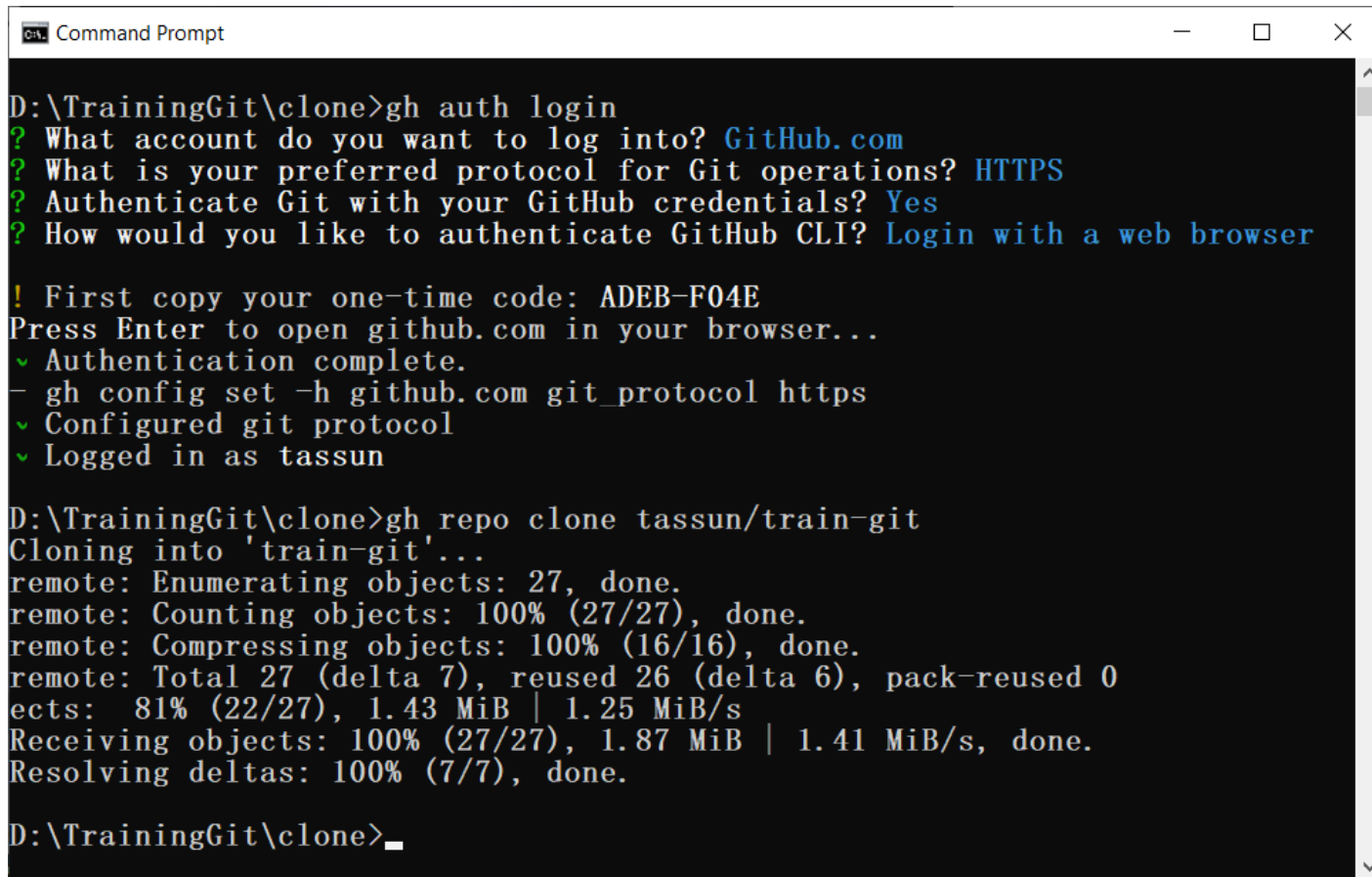

Git CLI

- GitHub CLI
 - Go to <https://cli.github.com/>
 - Download & Install



Git CLI

- GitHub CLI
 - try to login and clone



```
Command Prompt

D:\TrainingGit\clone>gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: ADEB-F04E
Press Enter to open github.com in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as tassun

D:\TrainingGit\clone>gh repo clone tassun/train-git
Cloning into 'train-git'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 27 (delta 7), reused 26 (delta 6), pack-reused 0
objects: 81% (22/27), 1.43 MiB | 1.25 MiB/s
Receiving objects: 100% (27/27), 1.87 MiB | 1.41 MiB/s, done.
Resolving deltas: 100% (7/7), done.

D:\TrainingGit\clone>
```

Git CLI

- GitHub CLI
 - list repository

```
Git Command Prompt
D:\TrainingGit\clone\train-git>gh repo list

Showing 14 of 14 repositories in @tassun

tassun/train-git      Introduction to Git      public      1d
tassun/will-db        Introduction to TypeScript public      3d
tassun/train-ts       Library classes         public      4d
tassun/will-lib        Simple SQL by place hold... public      5d
tassun/will-sql       Runner for moleculer ser... public      5d
tassun/will-run        slate setting            public      6d
tassun/will-api        Express DH Project       public      6d
tassun/islate          Diffie-Hellman Utilities private     Dec 22, 2022
tassun/express-dh      Utility functional       public     Nov 21, 2022
tassun/will-dh         Mar 8, 2022
tassun/will-util       Aug 9, 2021
tassun/voffice-maps1   public      Oct 5, 2017
tassun/chat
tassun/mypurse

D:\TrainingGit\clone\train-git>
```

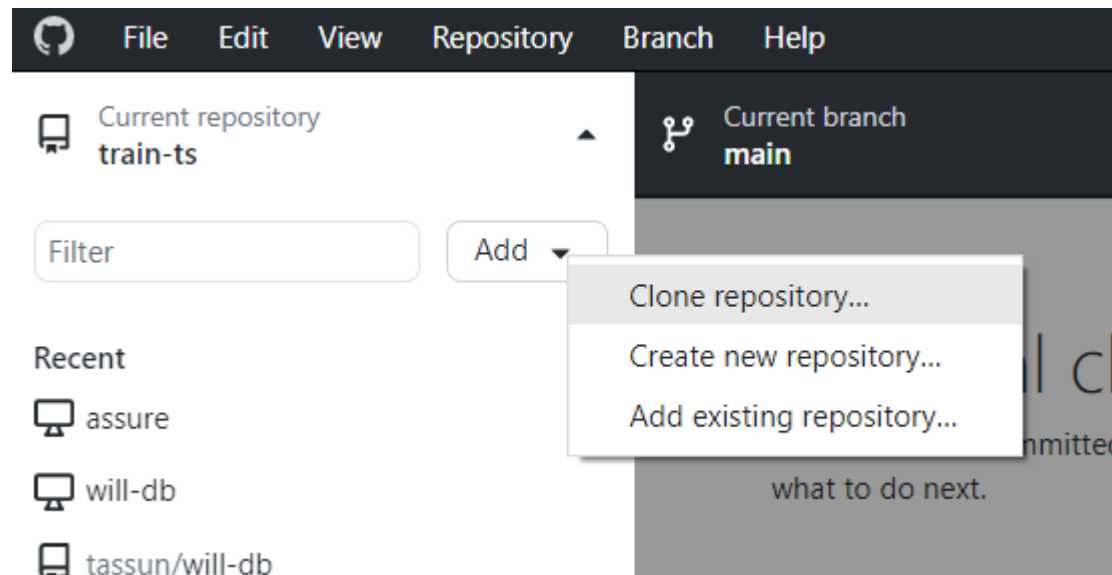
Git GUI

- Git Desktop
 - Go to <https://desktop.github.com/>
 - Download & Install



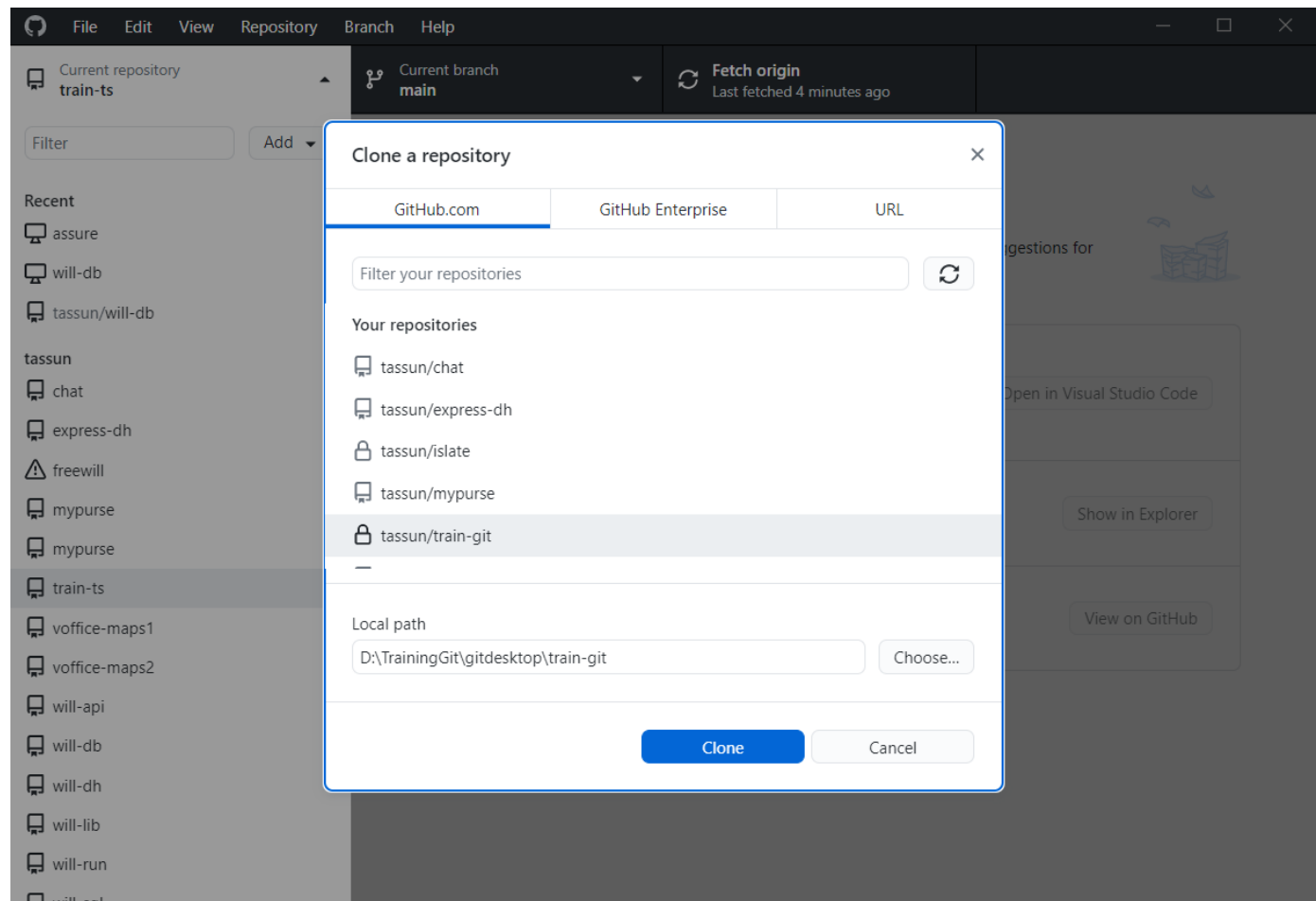
Git GUI

- Git Desktop



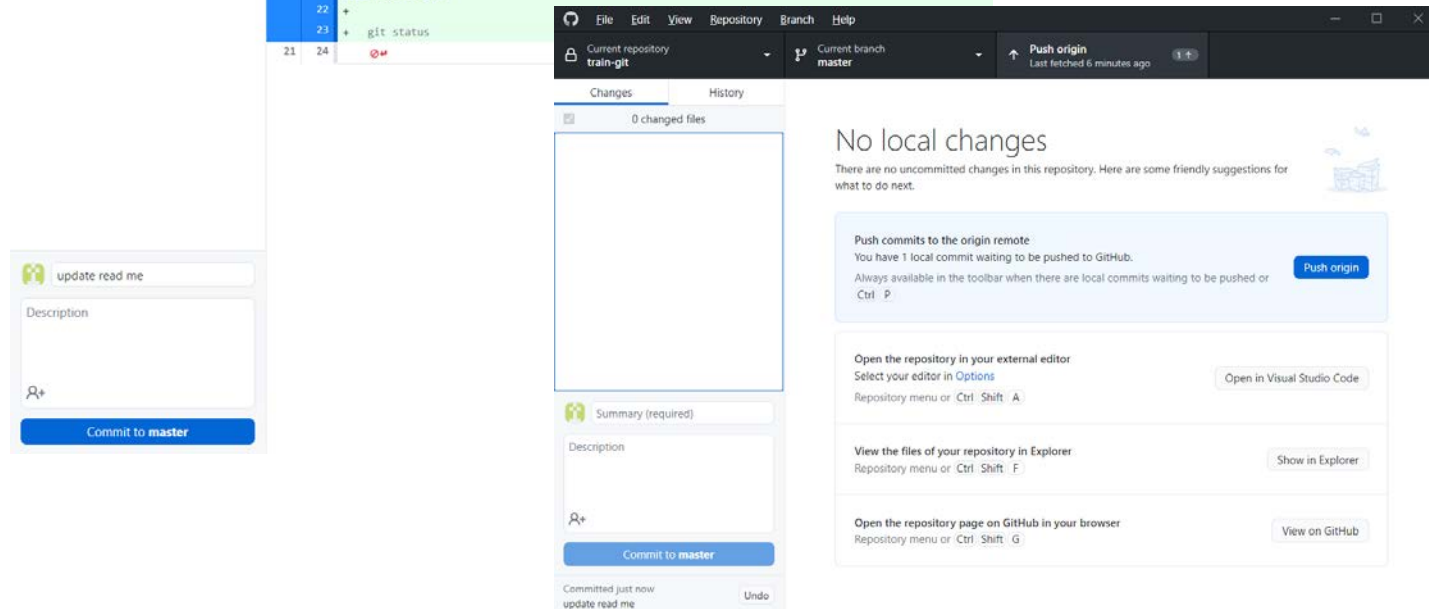
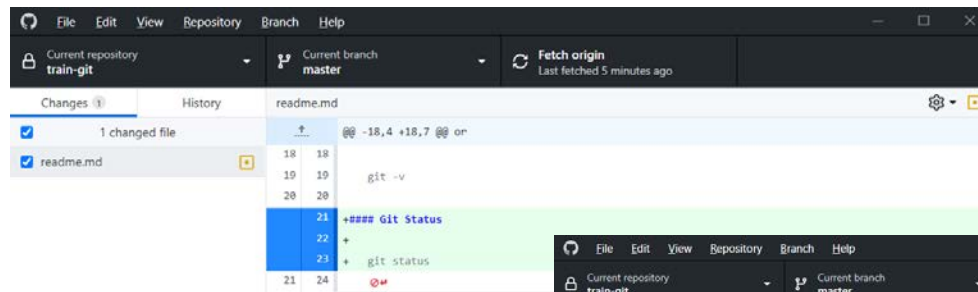
Git GUI

- Git Desktop



Git GUI

- Git Desktop
 - try to edit readme.md then commit & push

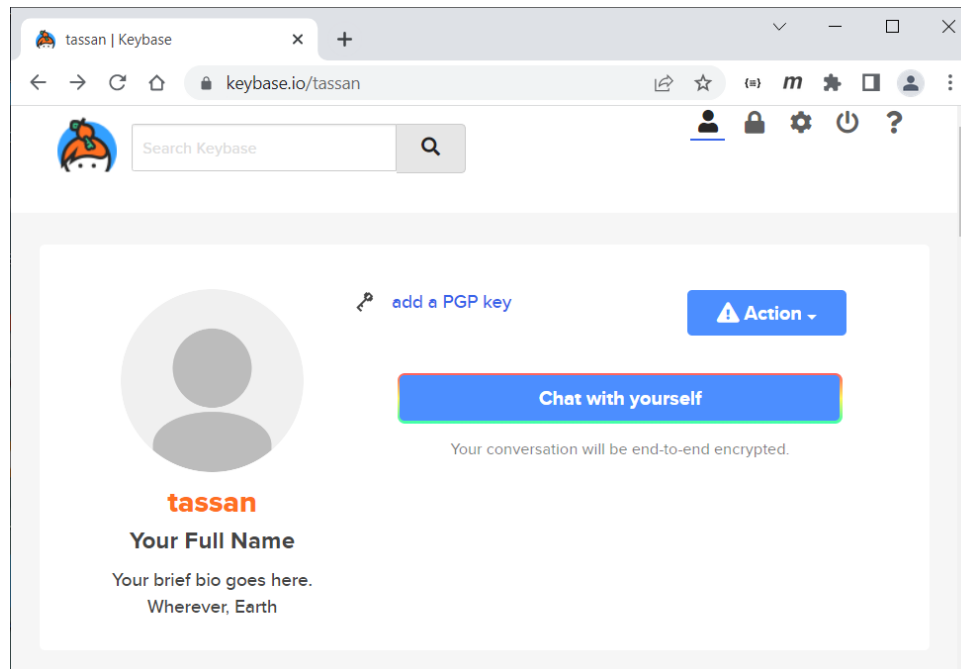


Git Sign

- GPG Key
 - Commit signature verification
 - can sign tags and commits locally
 - tags or commits are marked as verified
 - can be confident that the changes come from a trusted source

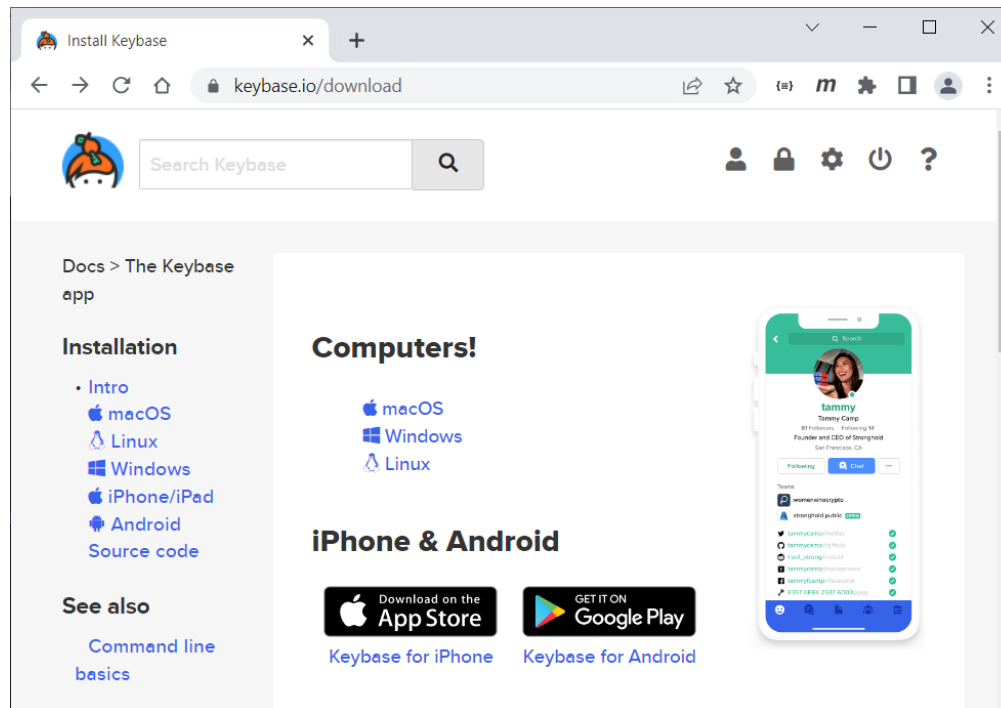
Git Sign

- GPG Key
 - Go to <https://keybase.io/> register and add PGP key



Git Sign

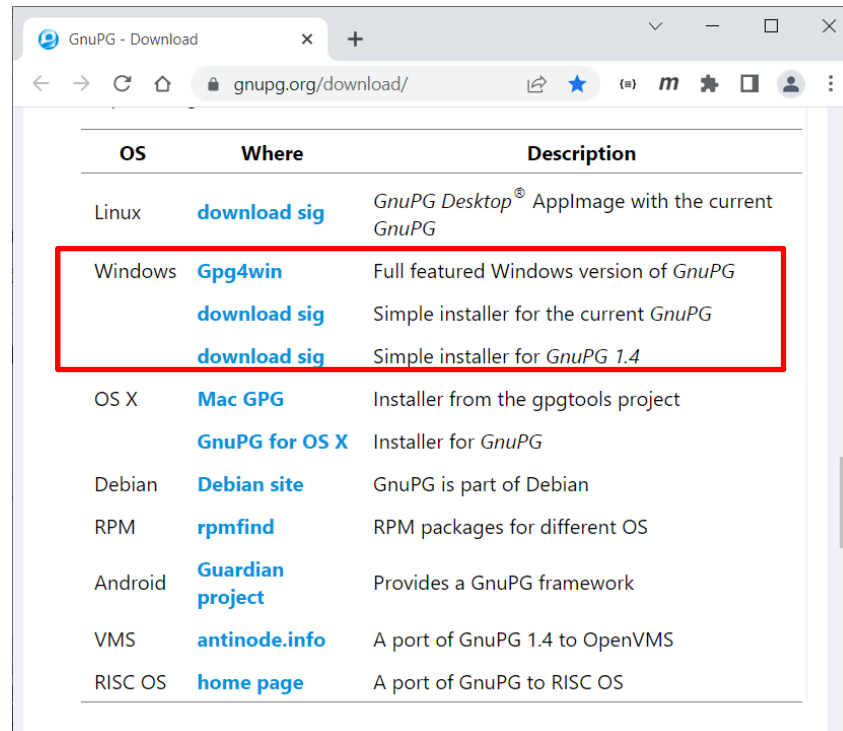
- GPG Key
 - Go to <https://keybase.io/download/>
 - Try to download and install



Git Sign

- GPG Key

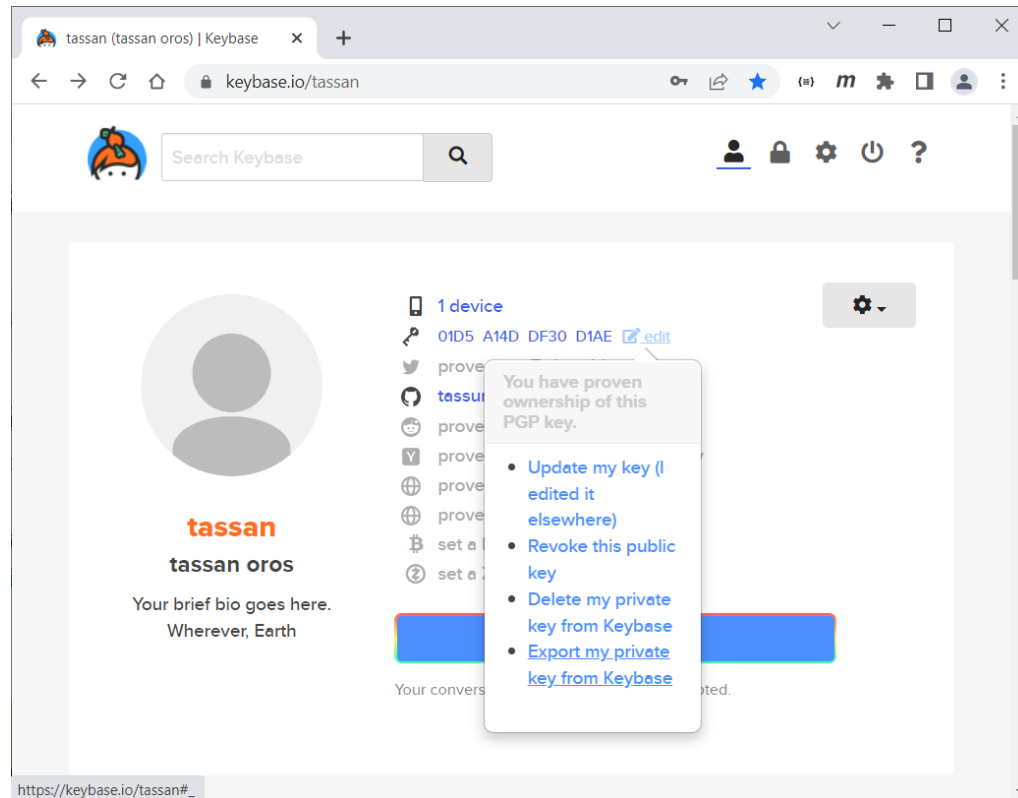
- Go to <https://www.gnupg.org/download/>
- Try to download and install

A screenshot of a web browser window showing the GnuPG download page. The browser's address bar displays 'gnupg.org/download/'. The page content is a table with three columns: 'OS', 'Where', and 'Description'. The 'Windows' section is highlighted with a red rectangular box. This section contains three links: 'Gpg4win', 'download sig', and 'download sig', each with a corresponding description. The other sections include Linux, OS X, Debian, RPM, Android, VMS, and RISC OS, each with a link and description.

OS	Where	Description
Linux	download sig	<i>GnuPG Desktop</i> ® Applimage with the current <i>GnuPG</i>
Windows	Gpg4win	Full featured Windows version of <i>GnuPG</i>
	download sig	Simple installer for the current <i>GnuPG</i>
	download sig	Simple installer for <i>GnuPG</i> 1.4
OS X	Mac GPG	Installer from the gpgtools project
	GnuPG for OS X	Installer for <i>GnuPG</i>
Debian	Debian site	GnuPG is part of Debian
RPM	rpmfind	RPM packages for different OS
Android	Guardian project	Provides a GnuPG framework
VMS	antinode.info	A port of GnuPG 1.4 to OpenVMS
RISC OS	home page	A port of GnuPG to RISC OS

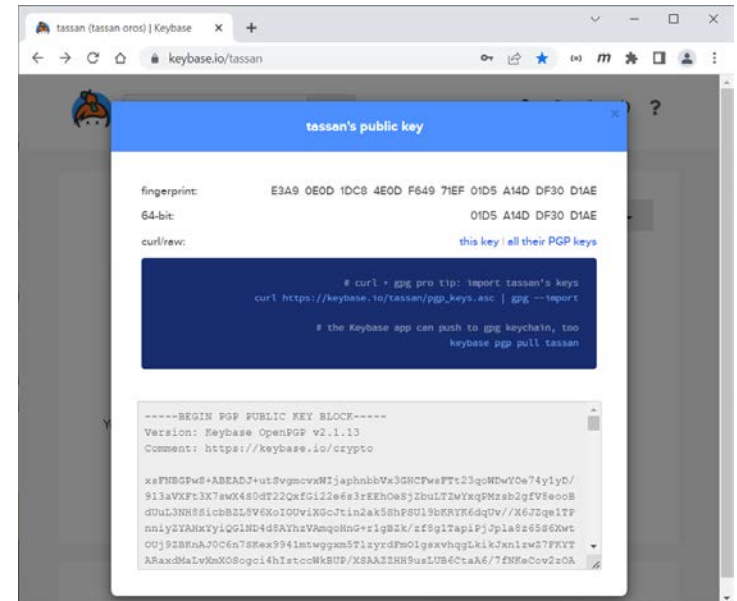
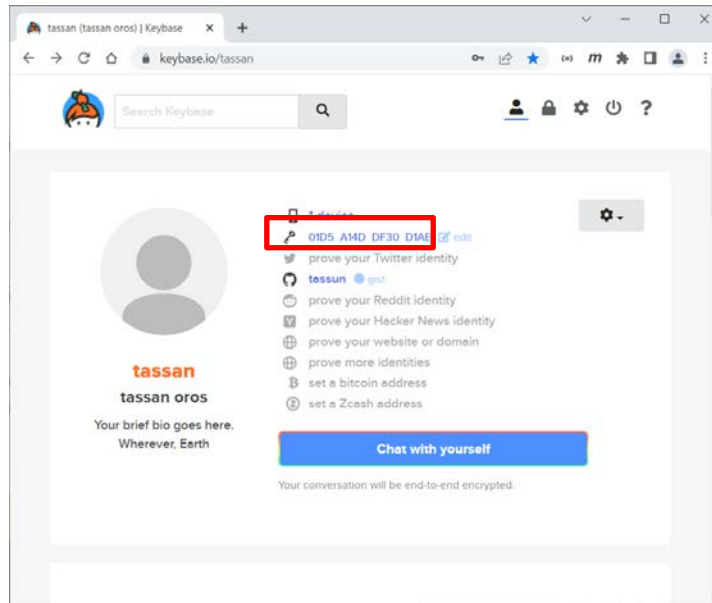
Git Sign

- GPG Key
 - Export private key from key base



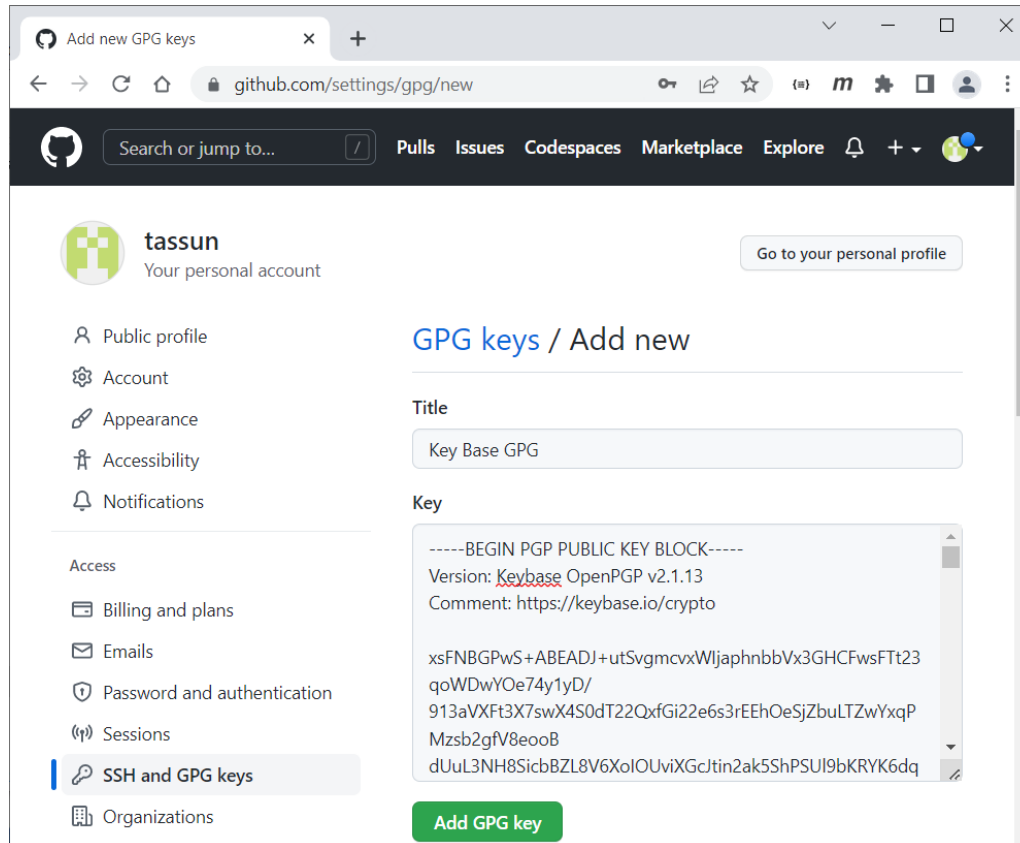
Git Sign

- GPG Key
 - Export public key from key base



Git Sign

- GPG Key
 - Add public key from key base to GitHub



Add new GPG keys

github.com/settings/gpg/new

Search or jump to...

Pulls Issues Codespaces Marketplace Explore

tassun
Your personal account

Go to your personal profile

Public profile
Account
Appearance
Accessibility
Notifications

Access

Billing and plans
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations

GPG keys / Add new

Title

Key Base GPG

Key

```
-----BEGIN PGP PUBLIC KEY BLOCK-----  
Version: Keybase OpenPGP v2.1.13  
Comment: https://keybase.io/crypto  
  
xsFNBGPwS+ABEADJ+utSvgrmcvxWljaphnbbVx3GHCFwsFTt23  
qoWDwYOe74y1yD/  
913aVXFt3X7swX4S0dT22QxfGi22e6s3rEEhOeSjZbuLTzwYxqP  
Mzsb2gfV8eooB  
dUuL3NH8SicbBZL8V6XoIOUviXGcJtin2ak5ShPSUI9bKRYK6dq
```

Add GPG key

Git Sign

- GPG Key
 - Config git – using fingerprint as sign key

```
Command Prompt
C:\Users\ADMIN>keybase id tassan
- INFO Identifying tassan
OK public key fingerprint: E3A9 0E0D 1DC8 4E0D F649 71EF 01D5
A14D DF30 D1AE
OK "tassun" on github: https://gist.github.com/32b0f6cb23672ac
2a3a50e3828d8ef96 [cached 2023-02-18 11:46:42 +07]
C:\Users\ADMIN>
```

```
Command Prompt
D:\TrainingGit\train-git>git config --global user.signingkey E3A90E0D1DC84E0DF64971EF01D5
A14DDF30D1AE
D:\TrainingGit\train-git>git config --global commit.gpgsign true
D:\TrainingGit\train-git>
```

Git Sign

- GPG Key
 - Import public & private key on local host

```
Command Prompt

D:\TrainingGit\keybase>gpg -q --import keybase_public.key

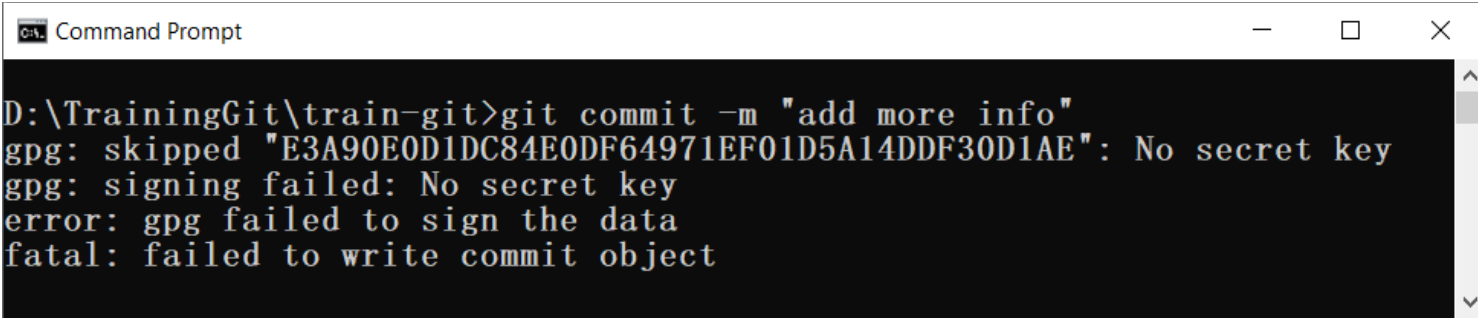
D:\TrainingGit\keybase>gpg -q --allow-secret-key-import --import keybase_private.key

D:\TrainingGit\keybase>gpg --list-keys
C:\Users\ADMIN\AppData\Roaming\gnupg\pubring.kbx
-----
pub   rsa4096 2023-02-18 [SC]
      E3A90E0D1DC84E0DF64971EF01D5A14DDF30D1AE
uid           [ unknown] tassan oros <tassun_oro@hotmail.com>
uid           [ unknown] tassan oros <tassunoros@gmail.com>
uid           [ unknown] tassan oros <tassan_oro@freewillsolutions.com>
sub   rsa2048 2023-02-18 [E] [expires: 2031-02-16]
sub   rsa2048 2023-02-18 [SA] [expires: 2031-02-16]

D:\TrainingGit\keybase>_
```


Git Sign

- GPG Key
 - Try to add, commit and push

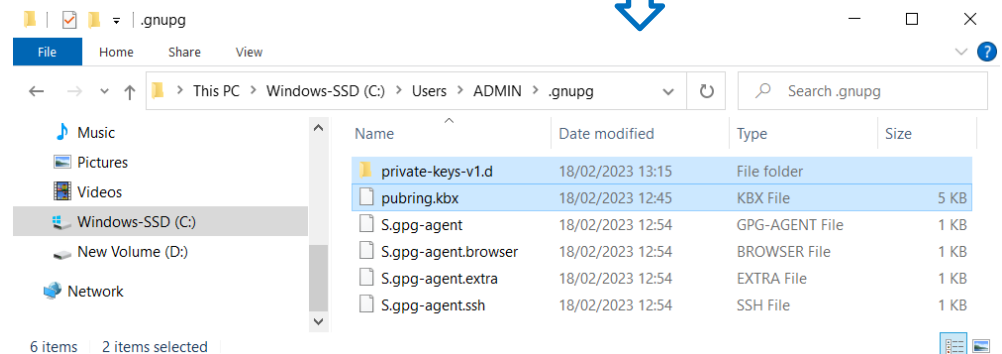
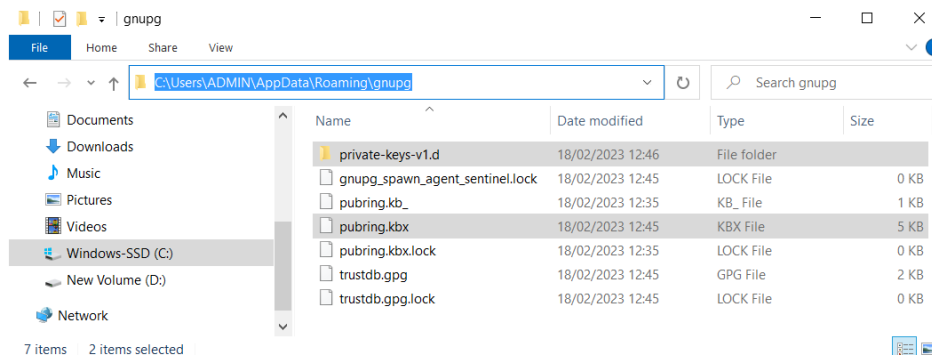


```
Command Prompt

D:\TrainingGit\train-git>git commit -m "add more info"
gpg: skipped "E3A90E0D1DC84E0DF64971EF01D5A14DDF30D1AE": No secret key
gpg: signing failed: No secret key
error: gpg failed to sign the data
fatal: failed to write commit object
```

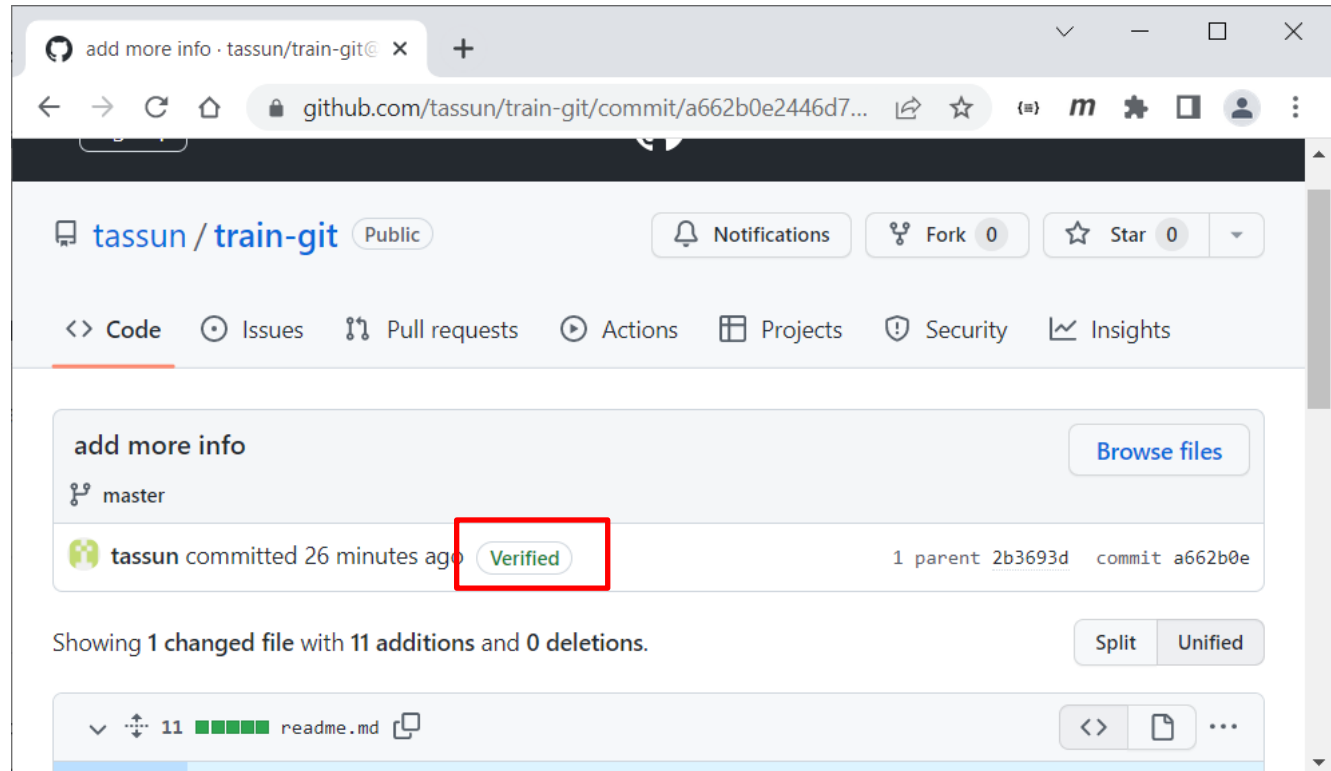
Git Sign

- GPG Key
 - Try to add, commit and push
 - Ensure pubring.kbx & private-keys-v1.d



Git Sign

- GPG Key
 - After push



Reference

- <https://devahoy.com/blog/2015/08/introduction-to-git-and-github>
- <https://km.cc.swu.ac.th/archives/3606>
- <https://saixiii.com/what-is-github/>
- <https://devahoy.com/blog/2017/12/how-to-show-verify-sign-with-gpg>
- <https://docs.github.com/en/authentication/connecting-to-github-with-ssh>
- <https://mahasak.com/keybase-pgp-ssh-exports/>
- <https://docs.github.com/en/authentication/managing-commit-signature-verification>



Q & A