

# **INSTITUT UNIVERSITAIRE DES SCIENCES (IUS)**

**FST**

*TD No1- Système*

Nom : EXUME

Prénom : Billy Rolph

Faculté : FST-INFORMATIQUE-L3

DATE : 26/10/2025

# Description des résultats de la tâche

L'objectif de ce TD était de comprendre et de manipuler les bases du système de gestion de **Git**, ainsi que d'apprendre à utiliser **GitHub** pour héberger un projet en ligne.

Plus précisément, il fallait :

- Installer et vérifier le bon fonctionnement de **Git** sur Windows.
- Créer et configurer un compte **GitHub**.
- Générer une **clé SSH** pour une connexion sécurisée entre la machine locale et GitHub.
- Initialiser un dépôt local, effectuer des **commits**, puis **pousser (push)** le projet sur GitHub.
- Créer la structure de dossiers demandée (`système`, `réseau`, etc.) et héberger le **rapport PDF** et les **images** sur le dépôt distant.

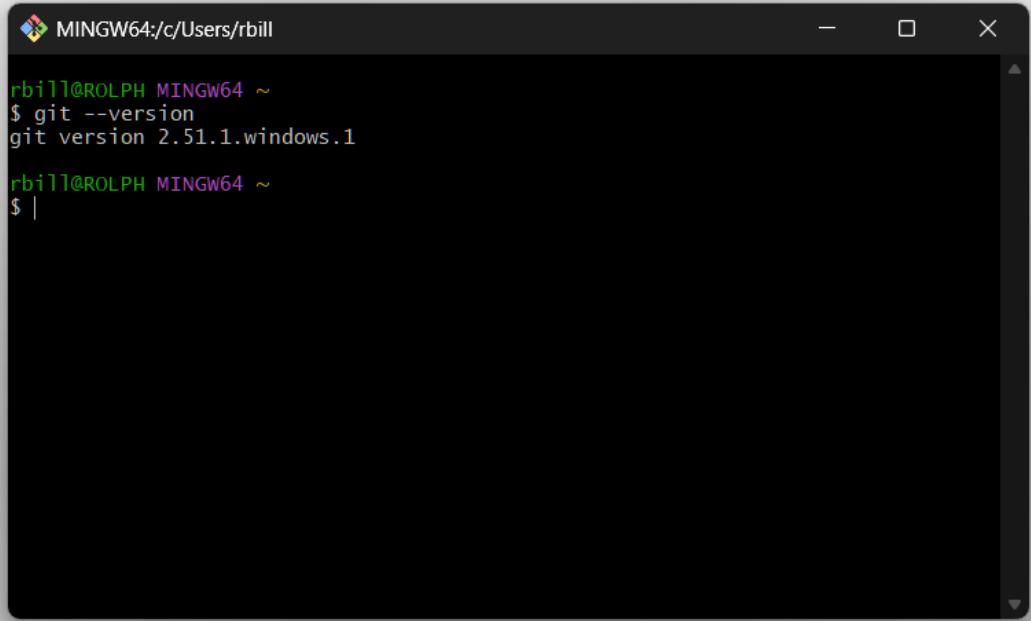
## Démarche suivie

### 1. Installation et vérification de Git

- J'ai installé Git sur Windows, puis j'ai vérifié l'installation avec la commande :

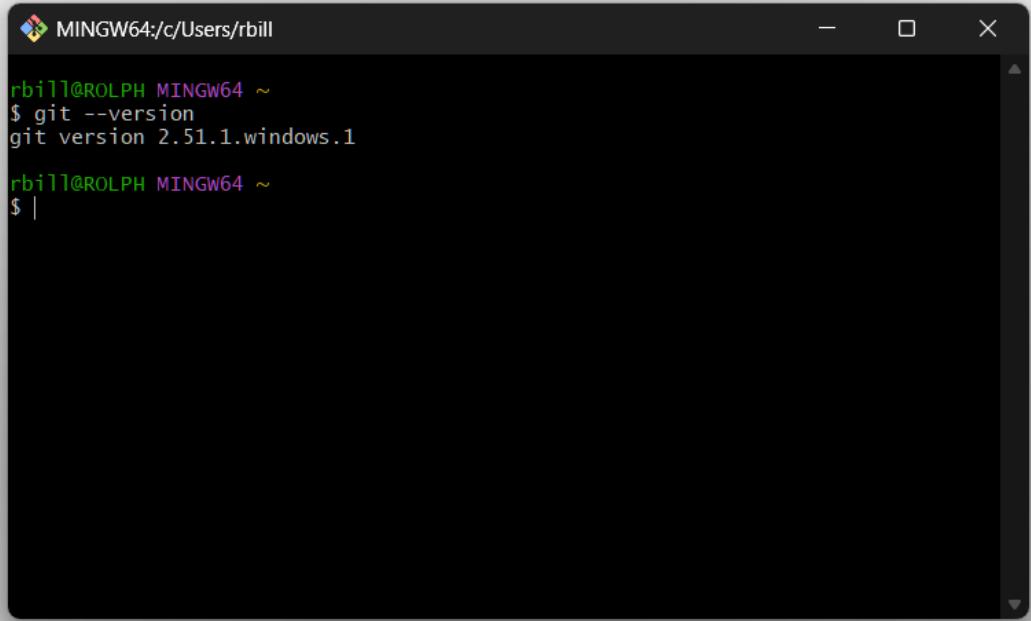
```
git -version
```

Le terminal a bien affiché la version installée.



```
MINGW64:/c/Users/rbill
rbill@ROLPH MINGW64 ~
$ git --version
git version 2.51.1.windows.1
rbill@ROLPH MINGW64 ~
$ |
```

Figure 1: vérification de l'instalation de Git sur mon pc



```
MINGW64:/c/Users/rbill
rbill@ROLPH MINGW64 ~
$ git --version
git version 2.51.1.windows.1
rbill@ROLPH MINGW64 ~
$ |
```

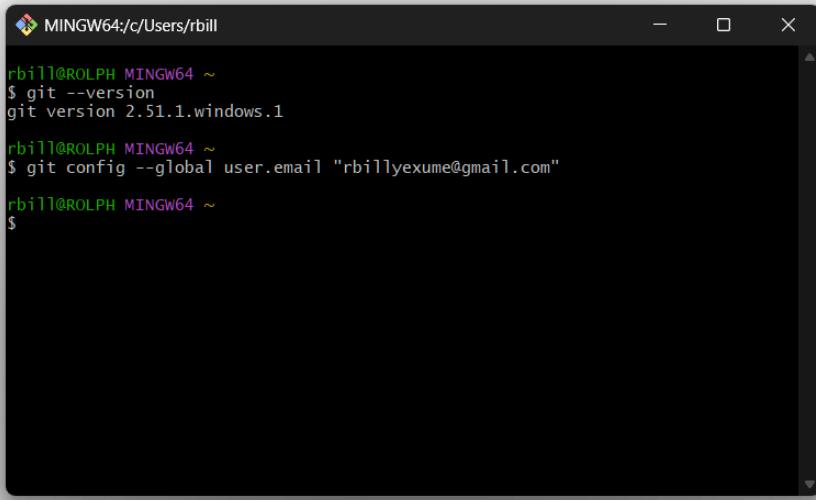
Figure 2 : vérification de l'instalation de Git sur mon pc

## 2. Compte GitHub

- J'avais déjà un compte github donc je l'ai utilisé.

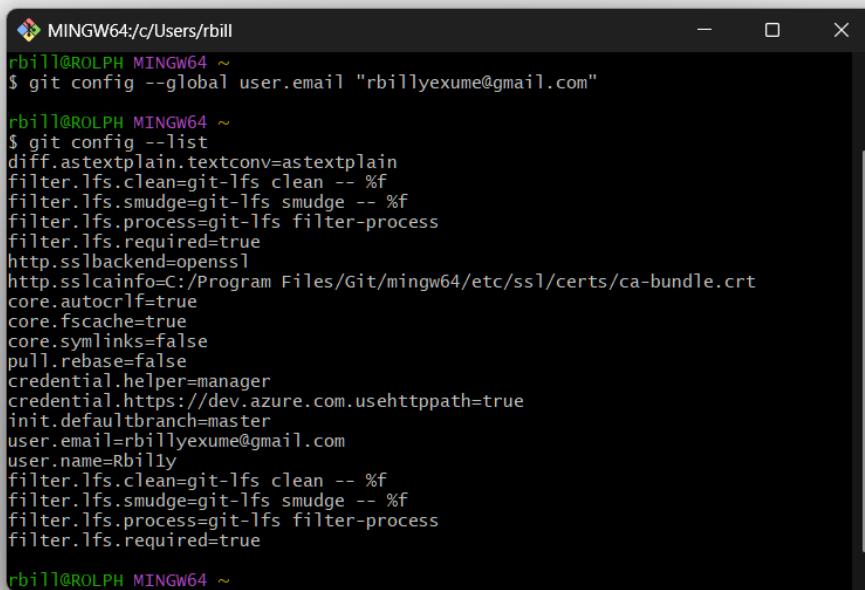
## 3. Configuration de Git

- J'ai configuré mon adresse e-mail pour identifier mes futurs commits :
- git config --global user.email "rbillyexume@gmail.com"
- Puis j'ai vérifié la configuration avec :
- git config -list



```
MINGW64:/c/Users/rbill
rbill@ROLPH MINGW64 ~
$ git --version
git version 2.51.1.windows.1
rbill@ROLPH MINGW64 ~
$ git config --global user.email "rbillyexume@gmail.com"
rbill@ROLPH MINGW64 ~
$
```

Figure 3: configuré mon adresse e-mail pour identifier mes futurs commits

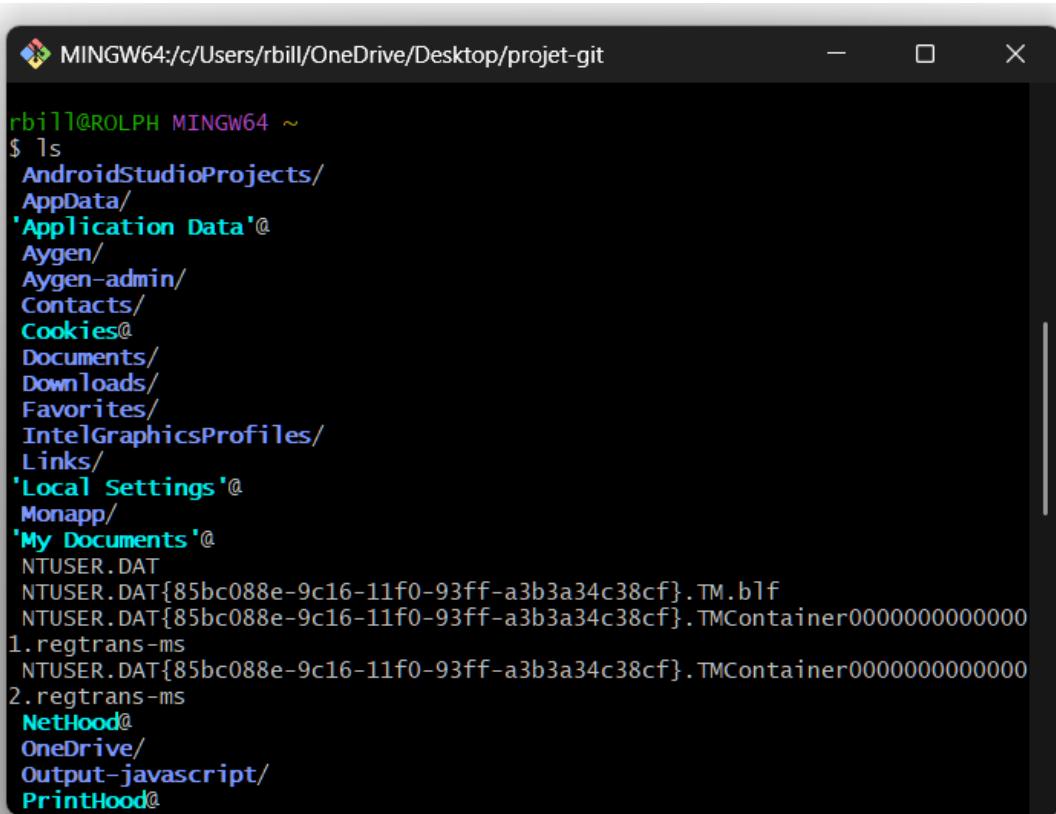


```
MINGW64:/c/Users/rbill
rbill@ROLPH MINGW64 ~
$ git config --global user.email "rbillyexume@gmail.com"
rbill@ROLPH MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=rbillyexume@gmail.com
user.name=Rbilly
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
rbill@ROLPH MINGW64 ~
```

Figure 4: je vérifie la configuration git config --list

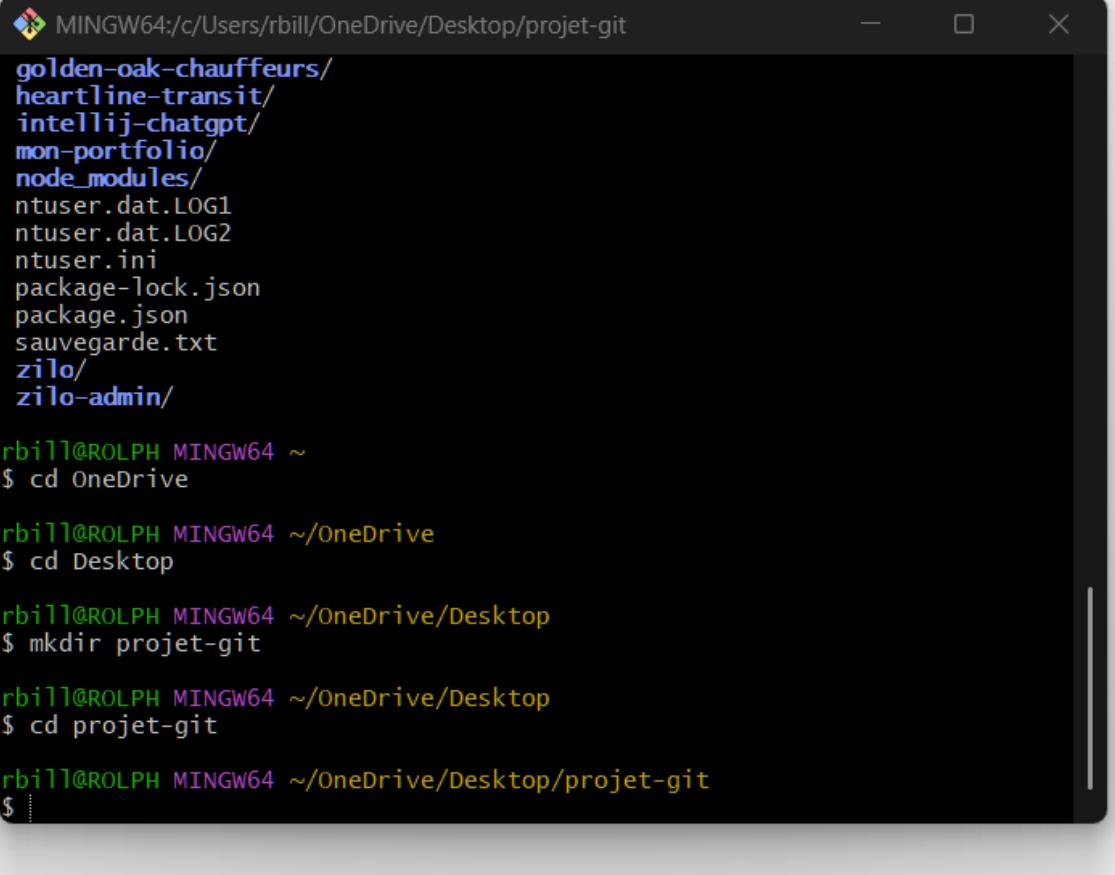
#### 4. Création d'un dépôt de test et premiers commits

- Dans PowerShell, j'ai créé un dossier de travail :
- cd Desktop
- mkdir projet-git
- cd projet-git
- J'ai initialisé le dépôt :
- git init
- J'ai créé un fichier README.md, ajouté et validé mon premier commit :
- echo "Hello Git!" > README.md
- git add README.md
- git commit -m "Premier commit avec PowerShell"
- git log



```
rbill@ROLPH MINGW64 ~
$ ls
AndroidStudioProjects/
AppData/
'Application Data'@
Agen/
Agen-admin/
Contacts/
Cookies@
Documents/
Downloads/
Favorites/
IntelGraphicsProfiles/
Links/
'Local Settings'@
Monapp/
'My Documents'@
NTUSER.DAT
NTUSER.DAT{85bc088e-9c16-11f0-93ff-a3b3a34c38cf}.TM.blf
NTUSER.DAT{85bc088e-9c16-11f0-93ff-a3b3a34c38cf}.TMContainer0000000000000000
1.regtrans-ms
NTUSER.DAT{85bc088e-9c16-11f0-93ff-a3b3a34c38cf}.TMContainer0000000000000000
2.regtrans-ms
NetHood@
OneDrive/
Output-javascript/
PrintHood@
```

Figure 5: liste de mes fichiers



The screenshot shows a terminal window titled "MINGW64:/c/Users/rbill/OneDrive/Desktop/projet-git". The window contains the following text:

```
golden-oak-chauffeurs/
heartline-transit/
intellij-chatgpt/
mon-portfolio/
node_modules/
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
package-lock.json
package.json
sauvegarde.txt
zilo/
zilo-admin/
rbill@ROLPH MINGW64 ~
$ cd OneDrive
rbill@ROLPH MINGW64 ~/OneDrive
$ cd Desktop
rbill@ROLPH MINGW64 ~/OneDrive/Desktop
$ mkdir projet-git
rbill@ROLPH MINGW64 ~/OneDrive/Desktop
$ cd projet-git
rbill@ROLPH MINGW64 ~/OneDrive/Desktop/projet-git
$ ..
```

Figure 6: Creation du dossier projet-git sur mon bureau

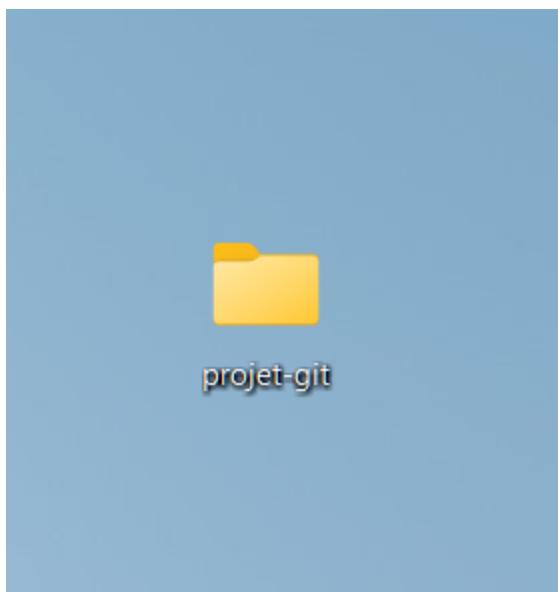


Figure 7: Le dossier est bien créer

```
MINGW64:/c/Users/rbill/OneDrive/Desktop/projet-git
$ cd Desktop

rbill@ROLPH MINGW64 ~/OneDrive/Desktop
$ mkdir projet-git

rbill@ROLPH MINGW64 ~/OneDrive/Desktop
$ cd projet-git

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/projet-git
$ git init
Initialized empty Git repository in C:/Users/rbill/OneDrive/Desktop/projet-git/.git/

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/projet-git (master)
$ echo "Hello Git">> README.md

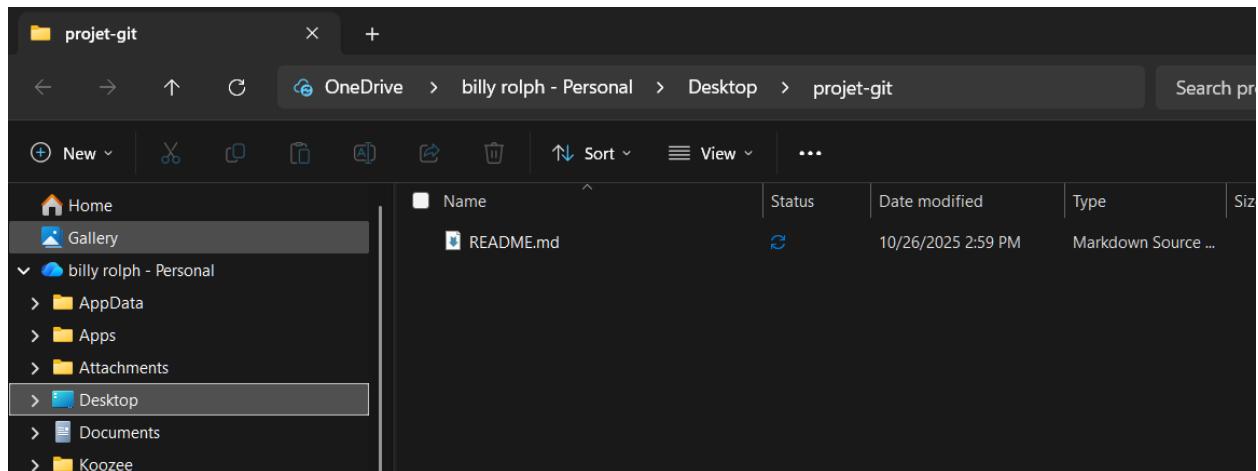
rbill@ROLPH MINGW64 ~/OneDrive/Desktop/projet-git (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

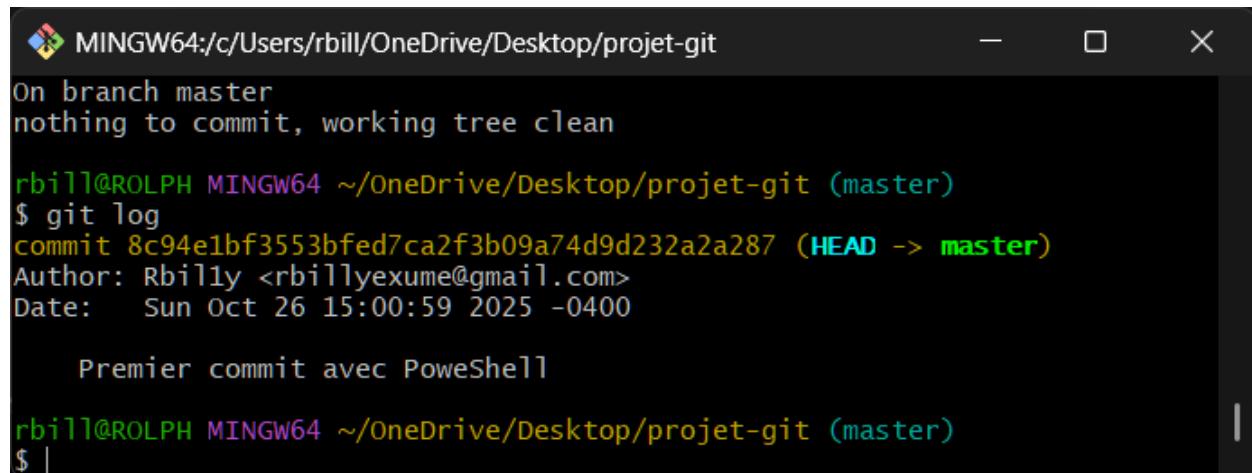
rbill@ROLPH MINGW64 ~/OneDrive/Desktop/projet-git (master)
$ git commit -m "Premier commit avec PowerShell"
[master (root-commit) 8c94e1b] Premier commit avec PowerShell
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/projet-git (master)
$ git status
On branch master
nothing to commit, working tree clean

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/projet-git (master)
$
```

Figure 8:initialisation du depo et creation d'un fichier README.md





```
MINGW64:/c/Users/rbill/OneDrive/Desktop/projet-git
On branch master
nothing to commit, working tree clean

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/projet-git (master)
$ git log
commit 8c94e1bf3553bfed7ca2f3b09a74d9d232a2a287 (HEAD -> master)
Author: RbillY <rbillyexume@gmail.com>
Date:   Sun Oct 26 15:00:59 2025 -0400

    Premier commit avec PowerShell

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/projet-git (master)
$ |
```

Figure 9: log de mon git , les ajouts ou modification repertorier dans le temps

## 5. Génération d'une clé SSH

- J'ai générée une paire de clés SSH :
- ssh-keygen -t ed25519 -C "rbillyexume@gmail.com"
- J'ai copié la clé publique dans les paramètres de mon compte GitHub (Settings → SSH and GPG keys).

```
rbill@ROLPH MINGW64 ~
$ ssh-keygen -t ed25519 -C "rbillyexume@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/rbill/.ssh/id_ed25519):
Created directory '/c/Users/rbill/.ssh'.
Enter passphrase for "/c/Users/rbill/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/rbill/.ssh/id_ed25519
Your public key has been saved in /c/Users/rbill/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:7sPmpSxB9vLgeMiddtB/wmA72fHETu4KjoZRFKpztUo rbillyexume@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|          ..        |
|          ..        |
|          ...       |
|          .o...     . |
|         ooE+S    . . |
|        +*o= o o o   |
|       . =%.X.= +   |
|      + Oo%o* + .   |
|      o *=o o       |
+---[SHA256]-----+
rbill@ROLPH MINGW64 ~
$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDp0j8HuggVm8n22PAhB69yHn5U/sNdJYcJHq
fTjFSQ rbillyexume@gmail.com

rbill@ROLPH MINGW64 ~
```

Figure 10:generer une paire de cle keygen et affichage de mon cle public pour le coller dans Github et avoir une connexion sécurisé

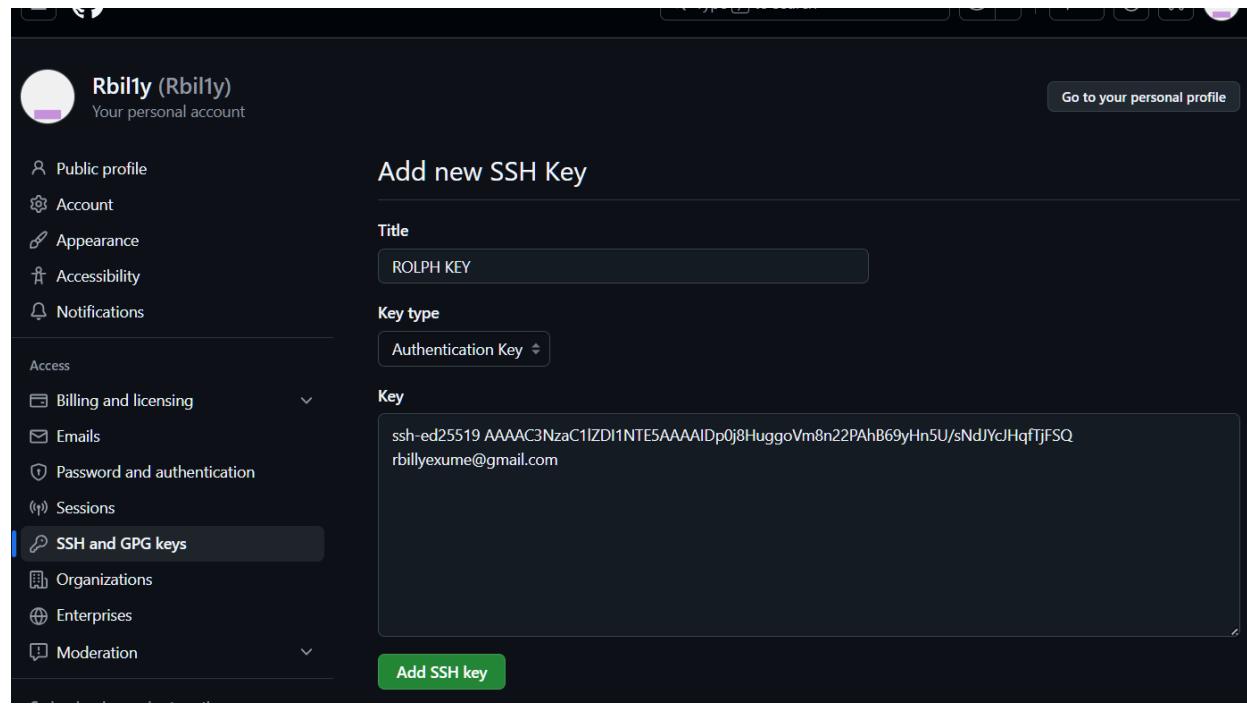


Figure 11: coller mon cle public dans github

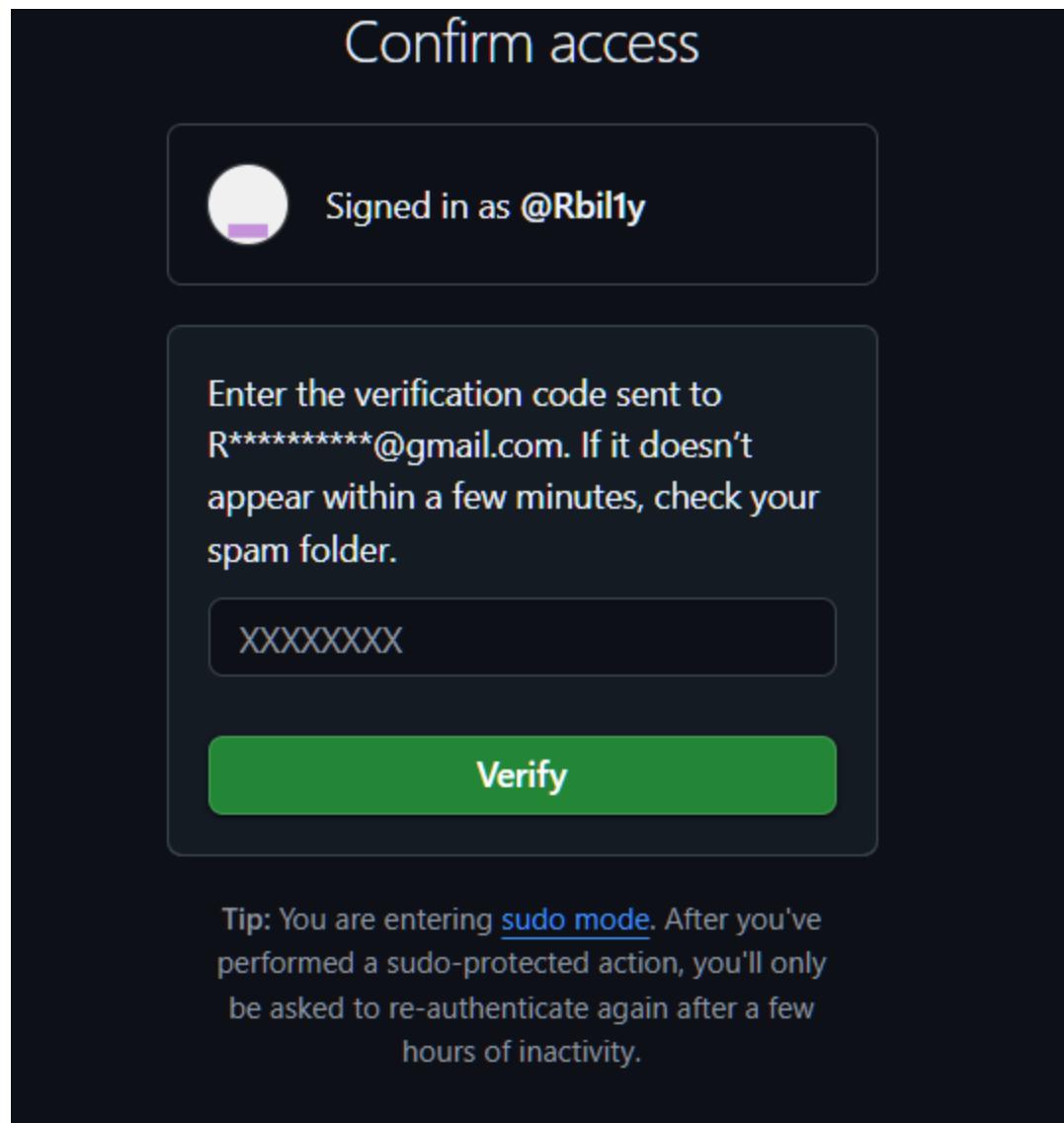


Figure 12: double verification

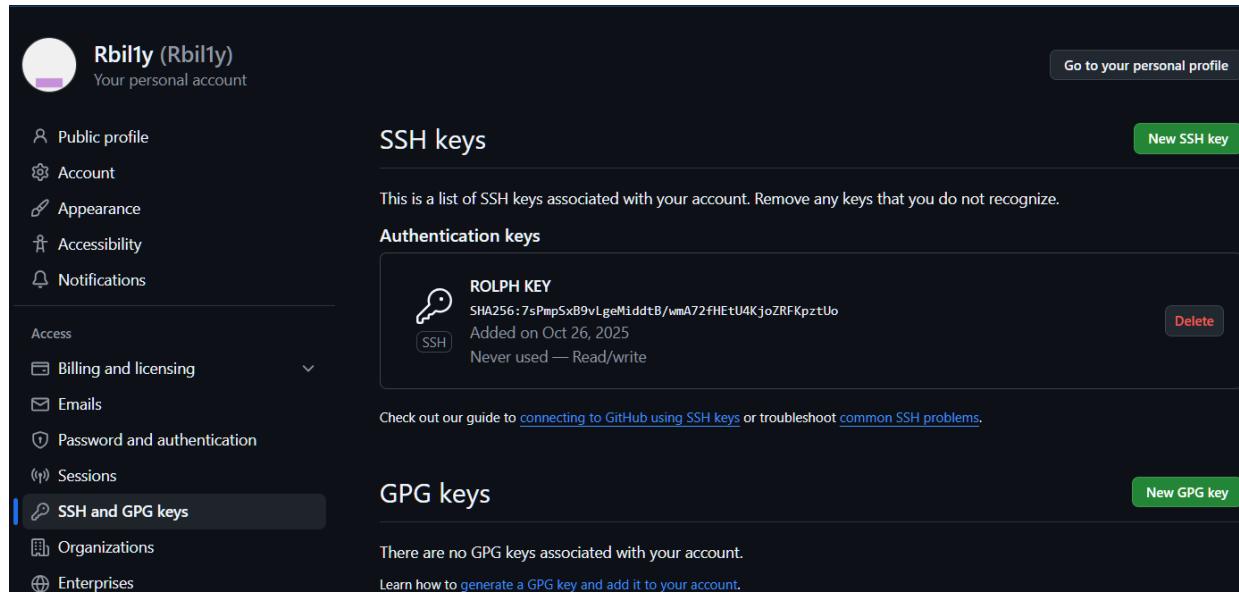
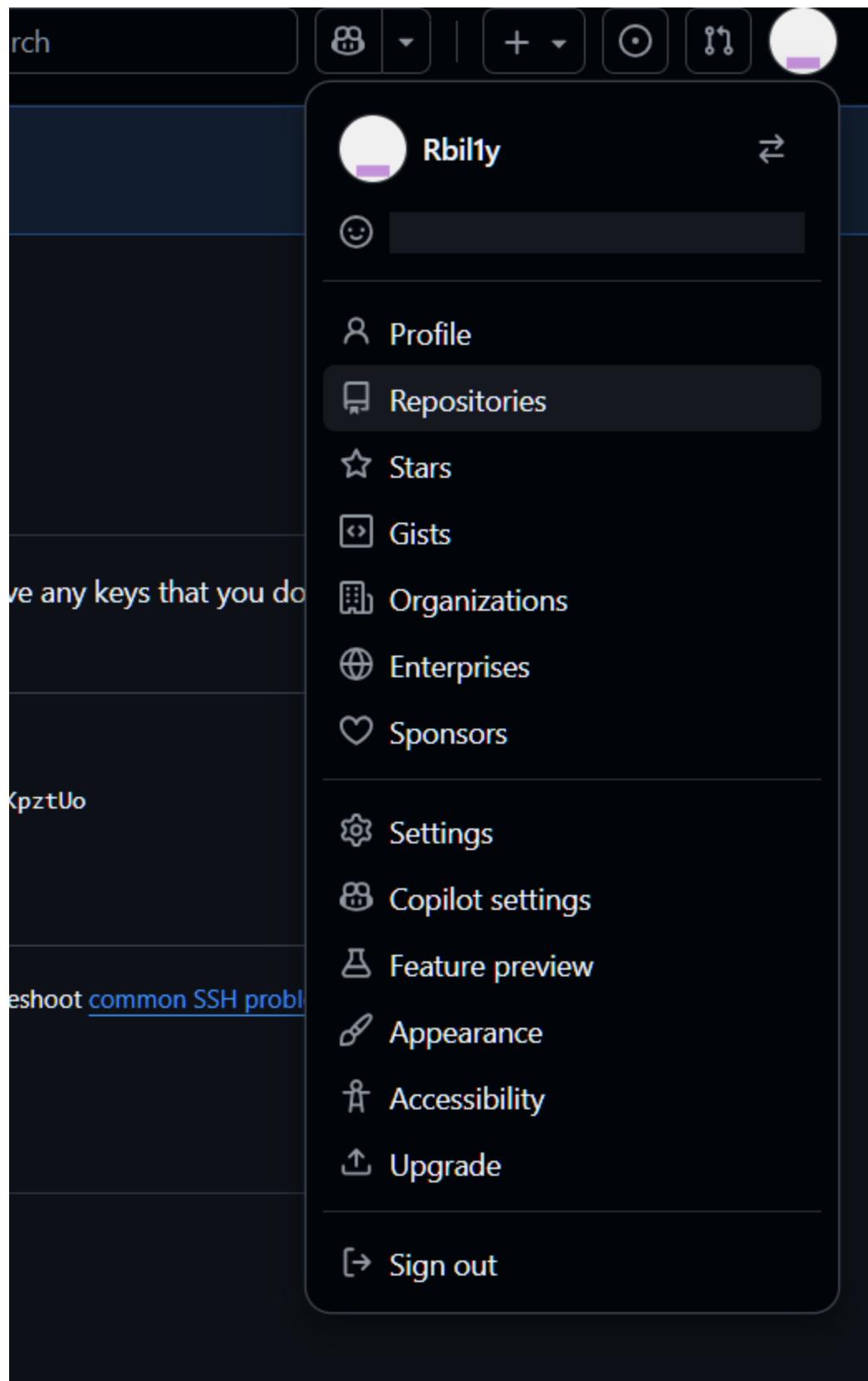


Figure 13: clé creer et connexion sécurisé

## 6. Création d'un repo mon-projet et clonage du repo GitHub

- Sur GitHub, j'ai créé un repo nommé **mon-projet**.
- J'ai ensuite cloné ce repo sur mon bureau via ssh
- `git clone git@github.com:monusername/mon-projet.git`
- j'ai créer un fichier `Module.txt` dans mon fichier sur mon pc
- puis j'ai pousser ce fichier sur Github
- `git add .`
- `git commit -m "Ajout rapport et images TD1"`
- `git push -u origin main`

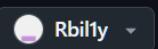


## Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).  
Required fields are marked with an asterisk (\*).

### 1 General

Owner \*



Rbilly

Repository name \*

mon-projet

✓ mon-projet is available.

Great repository names are short and memorable. How about [legendary-waddle](#)?

#### Description

0 / 350 characters

### 2 Configuration

Choose visibility \*

Choose who can see and commit to this repository

Public

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

On

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

Add license

Licenses explain how others can use your code. [About licenses](#)

No license

**Create repository**

The screenshot shows a GitHub repository page for 'mon-projet'. At the top, there's a header with navigation links: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the header, the repository name 'mon-projet' is displayed, along with a 'Public' badge. To the right of the repository name are buttons for Pin, Watch (0), Fork (0), Star (0), and a dropdown menu.

The main content area shows a file tree with a single file, 'README.md'. A modal window is open over the file tree, titled 'Clone', showing cloning options via HTTPS, SSH, or GitHub CLI. The GitHub CLI option is selected, displaying the command: `git@github.com:Rbilly/mon-projet.git`. Below the cloning options, there are links to 'Open with GitHub Desktop' and 'Download ZIP'.

On the right side of the page, there are sections for 'About', 'Releases', and 'Packages'. The 'About' section notes 'No description, website, or topics provided.' It includes links for Readme, Activity, Stars (0), Watching (0), and Forks (0). The 'Releases' section indicates 'No releases published' and provides a link to 'Create a new release'. The 'Packages' section shows 'No packages published' and a link to 'Publish your first package'.

At the bottom of the page, there's a footer with copyright information: '© 2025 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do not share my personal information'.

The screenshot shows a Windows File Explorer window with the path 'OneDrive > billy rolph - Personal > Desktop > mon-projet'. The current folder is 'mon-projet', which contains two files: 'Module.txt' and 'README.md'. The 'Module.txt' file was modified on 10/26/2025 at 3:51 PM and is a 1 KB Text Document. The 'README.md' file was modified on 10/26/2025 at 3:50 PM and is a 1 KB Markdown Source file. The window also shows a sidebar with folder icons and a preview pane on the right.

```
MINGW64:/c/Users/rbill/OneDrive/Desktop/mon-projet
rbill@ROLPH MINGW64 ~/OneDrive
$ cd Desktop

rbill@ROLPH MINGW64 ~/OneDrive/Desktop
$ git clone git@github.com:Rbillly/mon-projet.git
Cloning into 'mon-projet'...
The authenticity of host 'github.com (140.82.112.3)' can't be established.
ED25519 key fingerprint is: SHA256:+DiY3wvvV6TuJJhbZisF/zLDA0zPMSvHdkr4Uv
COqU
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

rbill@ROLPH MINGW64 ~/OneDrive/Desktop
$ ls
0124.mp4
0208.mp4
0227.mp4
0702-F4_BILLYLAB-INTRO.mp4
1003.mp4
1119.mp4
'1208(1).WAV'
1208.mp4
2025-04-09-114537227.mp4
AUD-20231006-WA0002.m4a
'Adobe Audition 2023.Lnk'*
'Adobe Premiere Pro 2023.Lnk'*
'Android Studio.Lnk'*
'Arc.Lnk'*
'Billy Lab'/
Billy-Edit.mp4
BossEsdras/
```

```
MINGW64:/c/Users/rbill/OneDrive/Desktop/mon-projet
$ cd mon-projet

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/mon-projet (main)
$ echo "Systeme d'exploitation!" > Module.txt

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/mon-projet (main)
$ git add .
warning: in the working copy of 'Module.txt', LF will be replaced by CRLF
the next time Git touches it

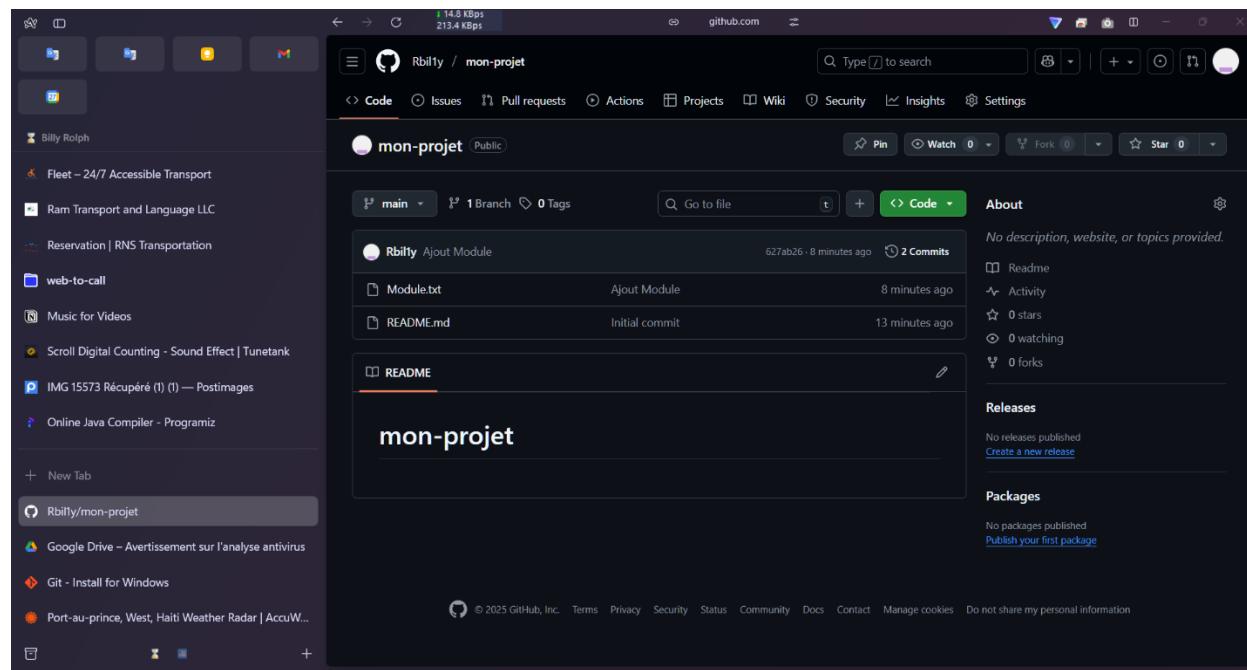
rbill@ROLPH MINGW64 ~/OneDrive/Desktop/mon-projet (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Module.txt

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/mon-projet (main)
$ git commit -m "Ajout Module"
[main 627ab26] Ajout Module
 1 file changed, 1 insertion(+)
 create mode 100644 Module.txt

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/mon-projet (main)
$ git branch -M main

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/mon-projet (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 303.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Rbillly/mon-projet.git
  ed3a38f..627ab26 main -> main
branch 'main' set up to track 'origin/main'.
```



## 7. Création et clonage d'un dépôt GitHub

- Sur GitHub, j'ai créé un dépôt nommé **TD**.
- J'ai ensuite cloné ce dépôt sur mon bureau via SSH :
- `git clone git@github.com:monusername/ TD.git`
- J'ai créé la structure demandée :  
○ `TD/`
  - `|-- systeme/`
    - `|-- image/`
    - `|-- projet/`
    - `|-- presentation/`
  - `|-- reseauI/`
    - `|-- image/`
    - `|-- projet/`
    - `|-- presentation/`
- J'ai ajouté et poussé les fichiers sur GitHub :
  - `git add .`
  - `git commit -m "Ajout rapport et images TD1"`
  - `git push -u origin main`

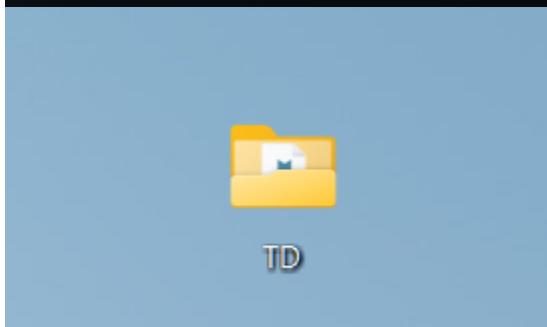
The screenshot shows a GitHub repository named 'TD' owned by user 'Rbil1y'. The repository is public and contains one branch ('main') and one commit ('Initial commit'). The README file is present and displays the text 'TD'. The repository has 0 forks and 0 stars. The 'About' section indicates no description, website, or topics provided.

The screenshot shows the same GitHub repository 'TD' with the cloning interface open. The 'Local' tab is selected, showing the clone URL 'git@github.com:Rbil1y/TD.git'. Other options include 'SSH' and 'GitHub CLI'. Below the URL, there is a note about using a password-protected SSH key. The rest of the repository page is identical to the first screenshot, showing the README file content and the 'About' section.

```
rbill@ROLPH MINGW64 ~/OneDrive/Desktop/mon-projet (main)
$ cd ..

rbill@ROLPH MINGW64 ~/OneDrive/Desktop
$ git clone git@github.com:Rbill1y/TD.git
Cloning into 'TD'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

rbill@ROLPH MINGW64 ~/OneDrive/Desktop
$
```



```
MINGW64:/c/Users/rbill/OneDrive/Desktop/TD
rbill@ROLPH MINGW64 ~/OneDrive/Desktop
$ cd TD

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD (main)
$ mkdir systeme

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD (main)
$ cd systeme

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD/systeme (main)
$ mkdir image

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD/systeme (main)
$ mkdir projet

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD/systeme (main)
$ mkdir presentation

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD/systeme (main)
$ cd ..

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD (main)
$ mkdir reseauI

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD (main)
$ cd reseauI

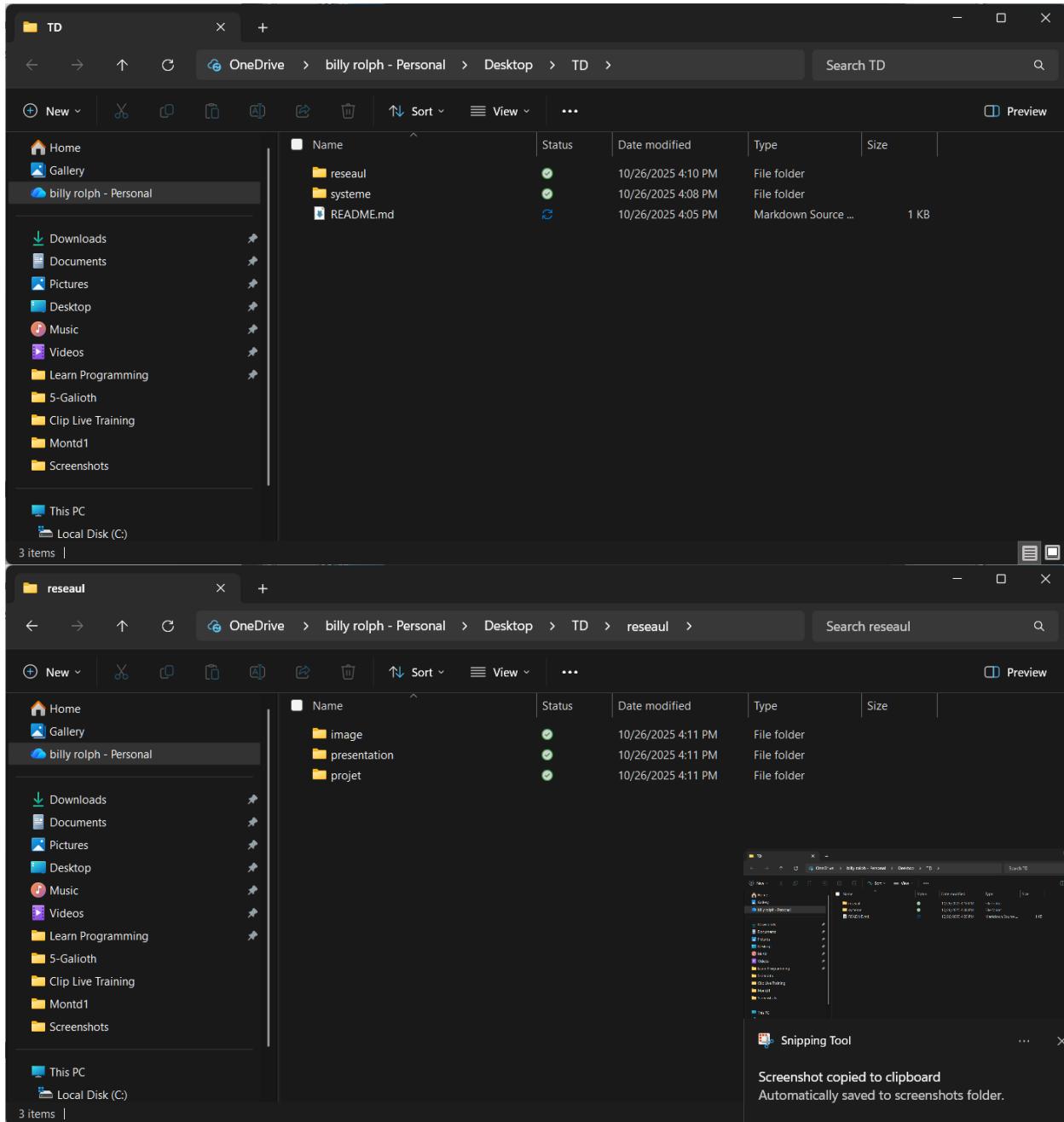
rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD/reseauI (main)
$ mkdir image

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD/reseauI (main)
$ mkdir projet

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD/reseauI (main)
$ mkdir presentation

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD/reseauI (main)
$ cd ..

rbill@ROLPH MINGW64 ~/OneDrive/Desktop/TD (main)
$ |
```



# Conclusion

## Résumé de ce que j'ai appris

Au cours de ce TD, j'ai appris :

- Le fonctionnement de Git et Github
- Utiliser les commandes de base : `git init, git add, git commit, git status, git push, ect.`
- Générer une clé SSH et l'utiliser pour sécuriser la connexion à GitHub.
- Créer et gérer un dépôt distant sur GitHub.
- Organiser un projet complet et y héberger mon rapport PDF avec ses fichiers associés.

## Résultat de la tâche

La tâche est **réussie**. 

Tous les fichiers ont été créés, validés et poussés sur GitHub avec succès.

## Difficultés rencontrées et solutions

Au début, j'ai eu une difficulté liée à **OneDrive**, car mon dossier *Desktop* se trouvait dans un répertoire synchronisé avec OneDrive.

J'ai d'abord paniqué en pensant que cela allait empêcher Git de fonctionner, mais finalement j'ai compris que je pouvais continuer à travailler sans problème.

Je pouvais aussi désactiver OneDrive, mais j'ai préféré le laisser activé, car j'ai plusieurs fichiers sur mon bureau que je veux garder sauvegardés en ligne.

Mon compte OneDrive ayant une capacité étendue à **1 To**, je ne crains pas de manquer d'espace de stockage.