



IUS
INSTITUT
UNIVERSITAIRE
DES SCIENCES

Faculté: Sciences Informatiques

TD No 1: Systèmes

Nom et Prénom: SAINT-JEAN Marc-Evenort

Professeur: Ismael Saint Amour

Niveau: 3ème Année

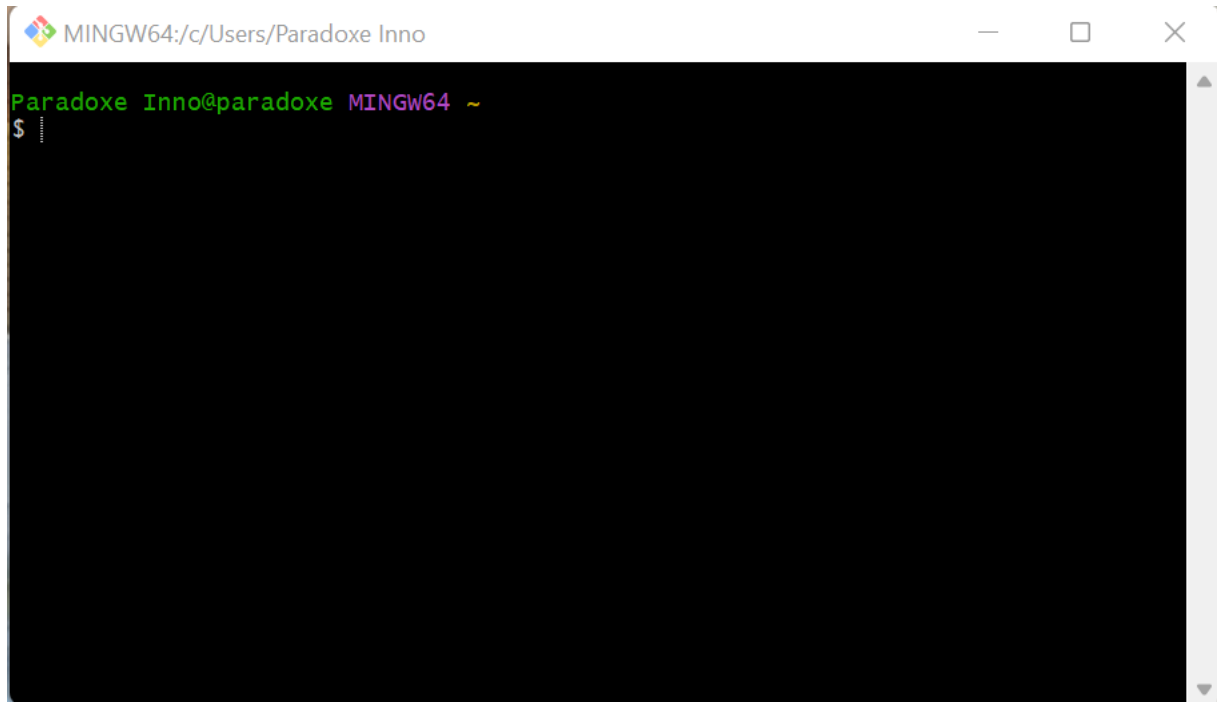
Date: Novembre 2025

Introduction:

Ce travail dirigé a pour objectifs de présenter et mettre en pratique les notions essentielles liées à l'utilisation de Git et GitHub.

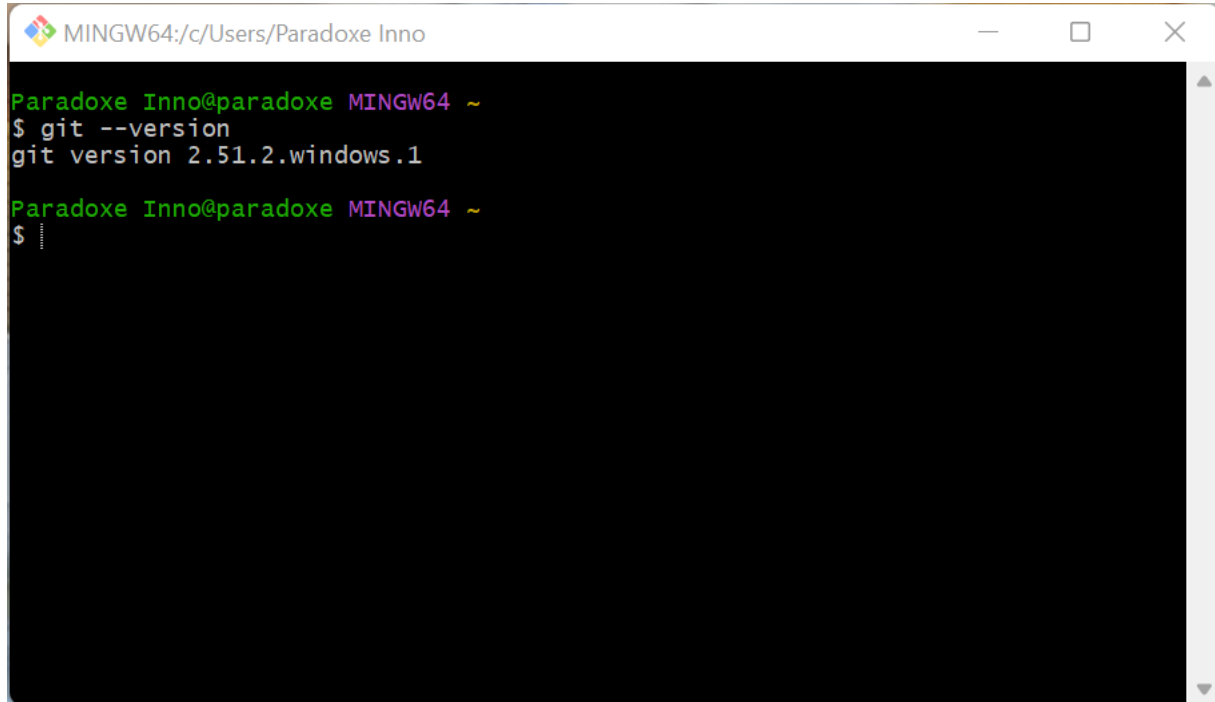
- Comprendre les concepts fondamentaux de Git (Utiliser Git en local).
- Créer un compte GitHub et configurer une connexion SSH.
- Créer et cloner un dépôt distant.
- Organiser un projet dans un dépôt et héberger un rapport PDF.

A) Installation et Configuration de Git



- Après l'installation de Git sur mon ordinateur (windows), j'ai fait un double-clic sur Git bash (bascule) pour ouvrir le terminal.

B) Vérifier la version de Git

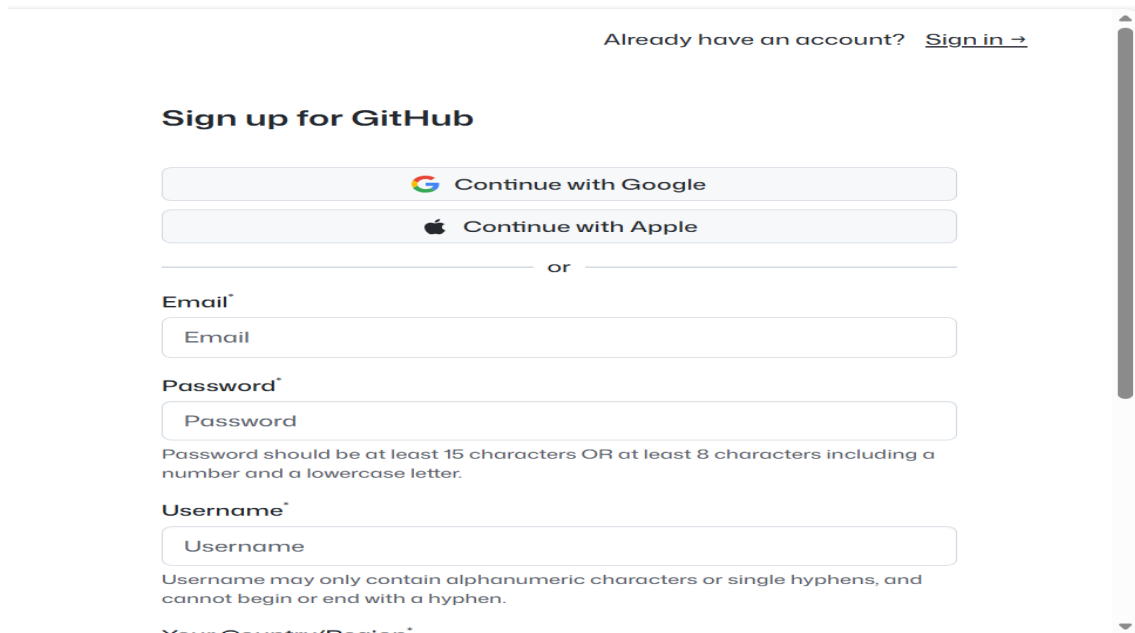


```
MINGW64:/c/Users/Paradoxe Inno
Paradoxe Inno@paradoxe MINGW64 ~
$ git --version
git version 2.51.2.windows.1
Paradoxe Inno@paradoxe MINGW64 ~
$
```

- La commande: **git --version** permet de vérifier la version de Git installée.


C) Création du compte Github


a) j'ai enregistré mes informations personnelles



Already have an account? [Sign in →](#)

Sign up for GitHub

 Continue with Google

 Continue with Apple

or

Email*

Email

Password*

Password

Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.

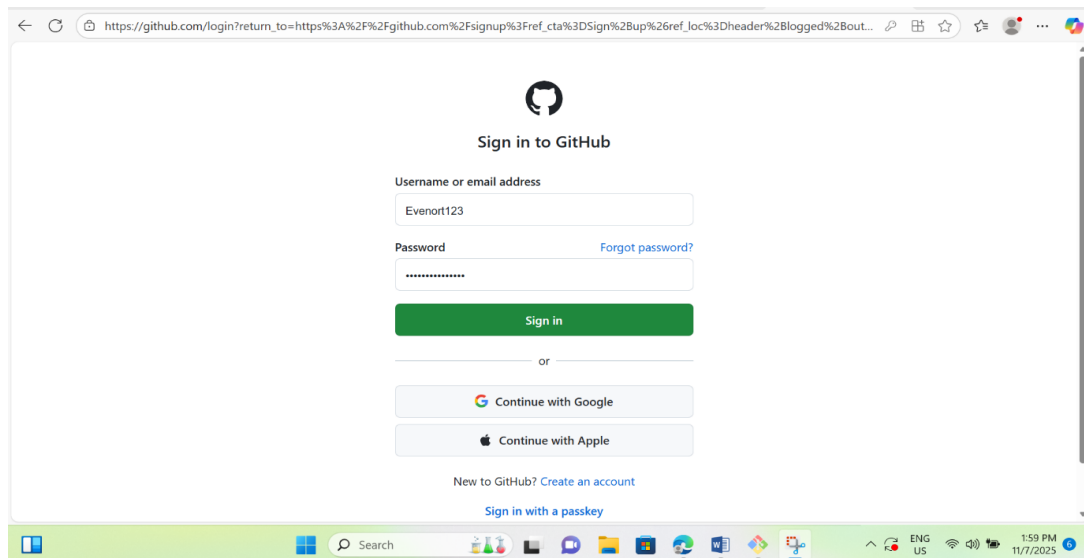
Username*

Username

Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.

Your Country/Region*

b) Connexion à GitHub



https://github.com/login?return_to=https%3A%2F%2Fgithub.com%2Fsignup%3Fref_cta%3DSign%2Bup%26ref_loc%3Dheader%2Blogged%2Bout...

Sign in to GitHub


Username or email address


Evenort123

Password [Forgot password?](#)

Sign in

or

 Continue with Google

 Continue with Apple

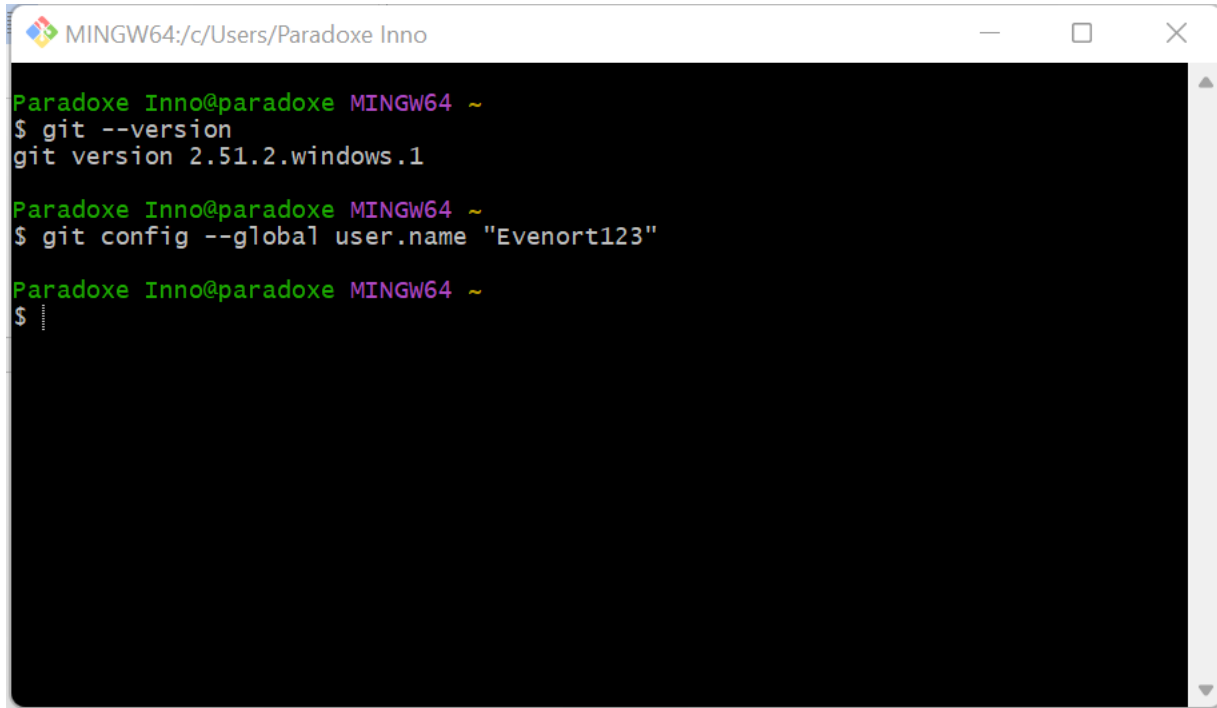
New to GitHub? [Create an account](#)

[Sign in with a passkey](#)

- Je me suis connecté(e) à mon compte GitHub en saisissant mon nom d'utilisateur (ou mon adresse e-mail) et mon mot de passe sur la page de connexion.

Tout ces démarches nous permettent d'avoir un compte GitHub pleinement fonctionnel.

D) Configuration de Git

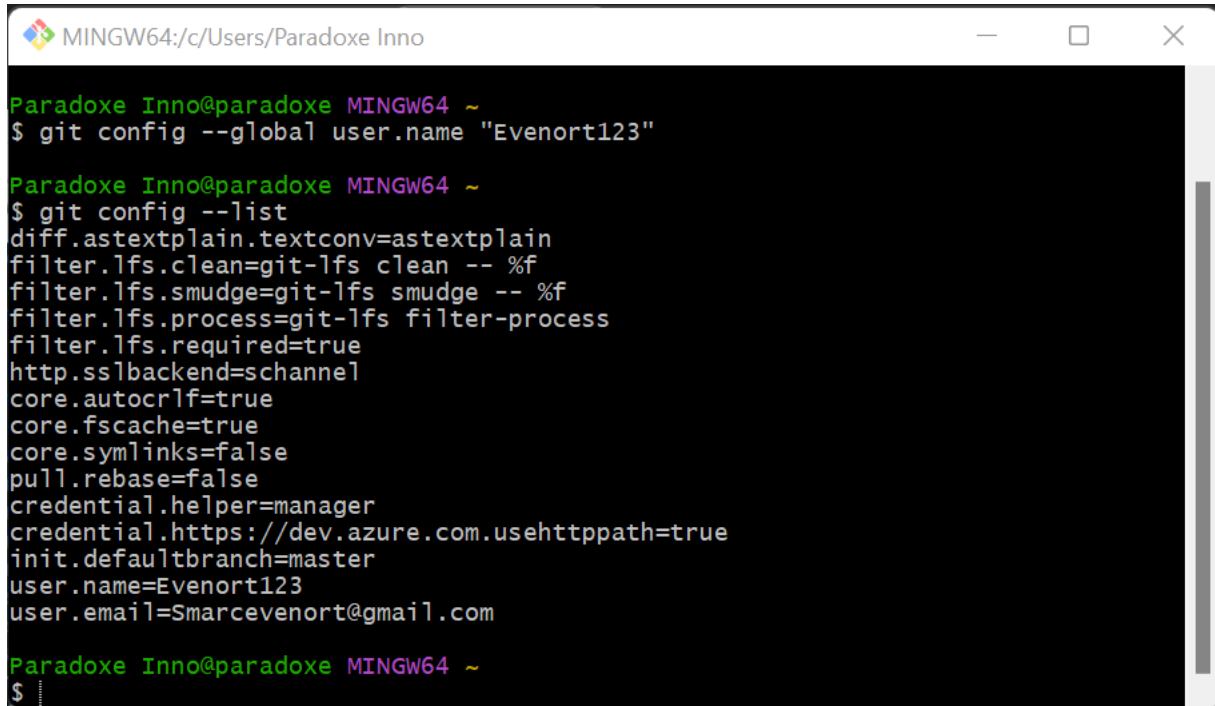
A screenshot of a MINGW64 terminal window. The title bar shows the path 'MINGW64:/c/Users/Paradoxe Inno'. The terminal content shows three lines of text: the first line is the prompt 'Paradoxe Inno@paradoxe MINGW64 ~' followed by the command '\$ git --version' and its output 'git version 2.51.2.windows.1'; the second line is the prompt 'Paradoxe Inno@paradoxe MINGW64 ~' followed by the command '\$ git config --global user.name "Evenort123"'; the third line is the prompt 'Paradoxe Inno@paradoxe MINGW64 ~' followed by '\$ |' and a cursor.

```
Paradoxe Inno@paradoxe MINGW64 ~  
$ git --version  
git version 2.51.2.windows.1  
  
Paradoxe Inno@paradoxe MINGW64 ~  
$ git config --global user.name "Evenort123"  
  
Paradoxe Inno@paradoxe MINGW64 ~  
$ |
```

- Dans cette étape, nous configurons Git avec notre nom d'utilisateur à l'aide de la commande: **git config --global user.name "Evenort123"**.

Cette image illustre la façon dont nous configurons Git dans le terminale.

E) Vérifier la Configuration



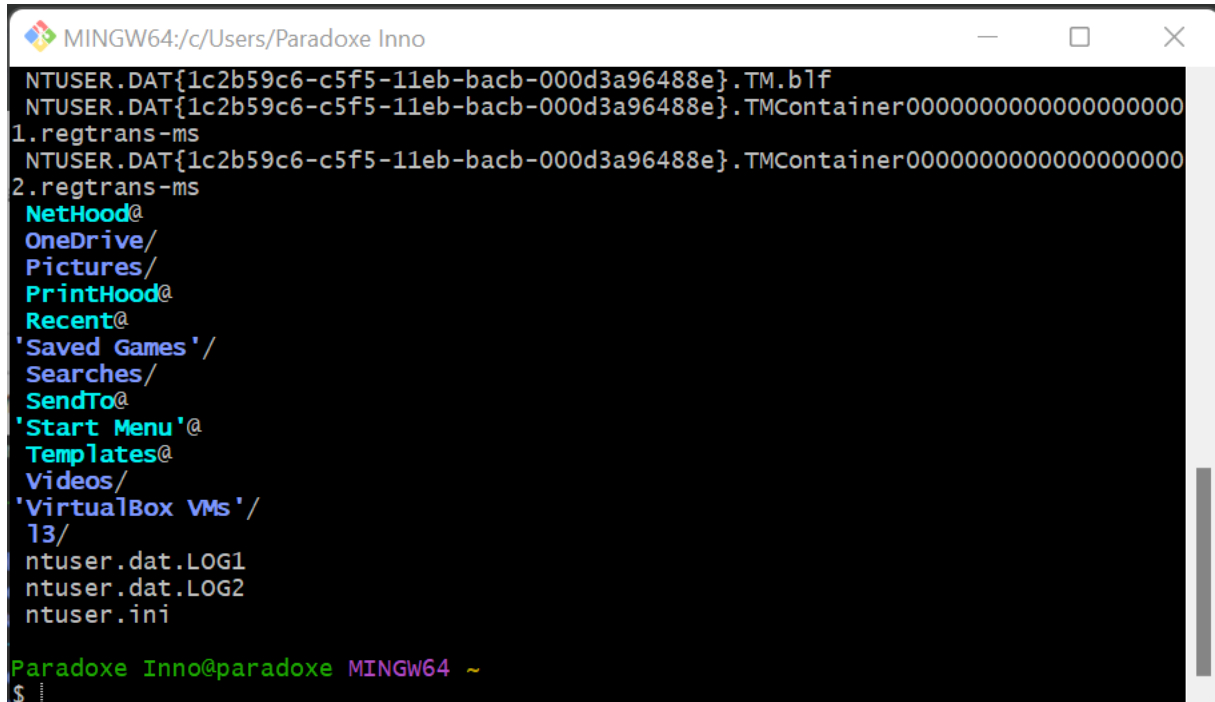
```
MINGW64:/c/Users/Paradoxe Inno
Paradoxe Inno@paradoxe MINGW64 ~
$ git config --global user.name "Evenort123"

Paradoxe Inno@paradoxe 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=schannel
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.name=Evenort123
user.email=Smarcevenort@gmail.com

Paradoxe Inno@paradoxe MINGW64 ~
$
```

- Pour vérifier la configuration de Git, on utilise la commande **git config --list**
Cette commande affiche les informations de configuration de l'application Git dans le terminal (Comme nom d'utilisateur et l'adresse e-mail).

F) Créer un dossier de test

A terminal window titled 'MINGW64:/c/Users/Paradoxe Inno' with standard window controls. It displays the output of the 'ls' command, listing files and directories in the user's home directory. The output includes system files like NTUSER.DAT, OneDrive, Pictures, and various user-specific folders like NetHood, PrintHood, Recent, and VirtualBox VMs. It also lists files like ntuser.dat.LOG1, ntuser.dat.LOG2, and ntuser.ini.

```
MINGW64:/c/Users/Paradoxe Inno
NTUSER.DAT{1c2b59c6-c5f5-11eb-bacb-000d3a96488e}.TM.b1f
NTUSER.DAT{1c2b59c6-c5f5-11eb-bacb-000d3a96488e}.TMContainer00000000000000000000
1.regtrans-ms
NTUSER.DAT{1c2b59c6-c5f5-11eb-bacb-000d3a96488e}.TMContainer00000000000000000000
2.regtrans-ms
NetHood@
OneDrive/
Pictures/
PrintHood@
Recent@
'Saved Games'/
Searches/
SendTo@
'Start Menu'@
Templates@
Videos/
'VirtualBox VMs'/
13/
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini

Paradoxe Inno@paradoxe MINGW64 ~
$
```

a) Pour afficher le contenu du répertoire dans le terminal, j'ai utilisé la commande **ls**

b) Créer des dossiers dans le terminal

A terminal window titled 'MINGW64:/c/Users/Paradoxe Inno/Desktop/TD/systeme' with standard window controls. It shows a sequence of terminal commands used to create a directory structure: navigating to the Desktop, creating 'TD', navigating to 'TD', creating 'systeme', navigating to 'systeme', creating 'reseauI', navigating to 'reseauI', creating 'image', and finally navigating to 'image'.

```
MINGW64:/c/Users/Paradoxe Inno/Desktop/TD/systeme
Paradoxe Inno@paradoxe MINGW64 ~
$ cd Desktop

Paradoxe Inno@paradoxe MINGW64 ~/Desktop
$ mkdir TD

Paradoxe Inno@paradoxe MINGW64 ~/Desktop
$ cd TD

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD
$ mkdir systeme

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD
$ mkdir reseauI

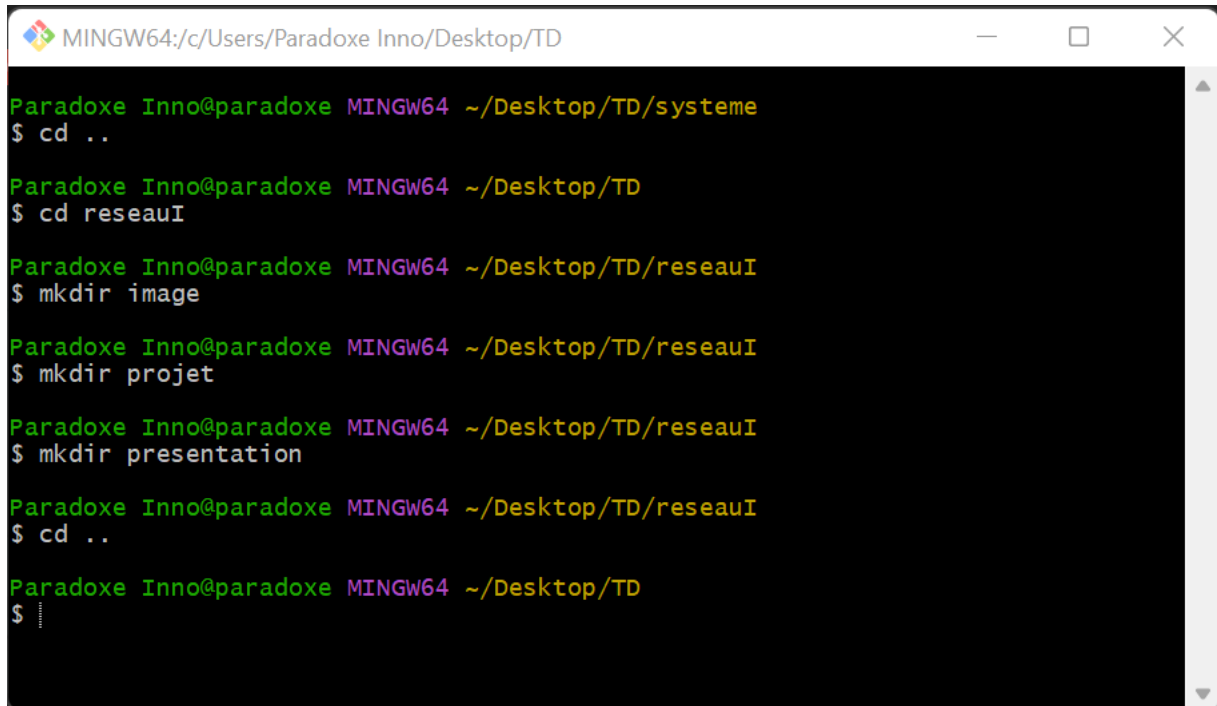
Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD
$ cd systeme

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD/systeme
$ mkdir image

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD/systeme
$
```

Pour accéder au bureau de l'ordinateur, j'ai utilisé la commande **cd Desktop** et avec la commande **mkadir**, j'ai créé différents dossiers (TD,systeme,reseauI,image). Ces commandes permettent de créer plusieurs dossiers et sous-dossiers directement dans le terminal.

c) Utiliser la commande `cd ..`



The screenshot shows a terminal window titled "MINGW64:/c/Users/Paradoxe Inno/Desktop/TD". The user is navigating through directories. The prompt is "Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD/systeme". The user enters "\$ cd ..", and the prompt changes to "Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD". The user then enters "\$ cd reseauI", and the prompt changes to "Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD/reseauI". The user enters "\$ mkdir image", and the prompt remains the same. The user enters "\$ mkdir projet", and the prompt remains the same. The user enters "\$ mkdir presentation", and the prompt remains the same. The user enters "\$ cd ..", and the prompt changes to "Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD". The user enters "\$", and the prompt remains the same.

```
Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD/systeme
$ cd ..

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD
$ cd reseauI

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD/reseauI
$ mkdir image

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD/reseauI
$ mkdir projet

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD/reseauI
$ mkdir presentation

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD/reseauI
$ cd ..

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD
$
```

- Elle permet de revenir au dossier précédent, dans cette image la commande `cd ..` est utilisée pour naviguer du dossier `systeme` vers le dossier `TD`.

```
MINGW64:/c/Users/Paradoxe Inno/Desktop/TD

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD
$ git init
Initialized empty Git repository in C:/Users/Paradoxe Inno/Desktop/TD/.git/

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (master)
$ echo "bonjour" > README.md

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md

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

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (master)
$ git add .
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git
touches it

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (master)
$ git status
On branch master

No commits yet

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

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (master)
$ git commit -m "premier commit"
[master (root-commit) fbb54c9] premier commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (master)
$ |

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (master)
$ git log
commit fbb54c96f5f287706c7be9ee32873c533bfe76ee (HEAD -> master)
Author: Evenort123 <Smarcevenort@gmail.com>
Date:   Fri Nov 7 17:13:37 2025 -0500

    premier commit

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (master)
$ |
```

- Ces deux images nous indiquent la façon dont on peut initialiser Git dans un dossier.
- Elle montrent aussi comment créer différents fichiers et ajouter au suivi de Git.
- En fin la commande git log permet de voir le commit Head (master), l'Auteur, la Date et vérifier que le travail est bien démarré.

G) Générer la la clés SSH

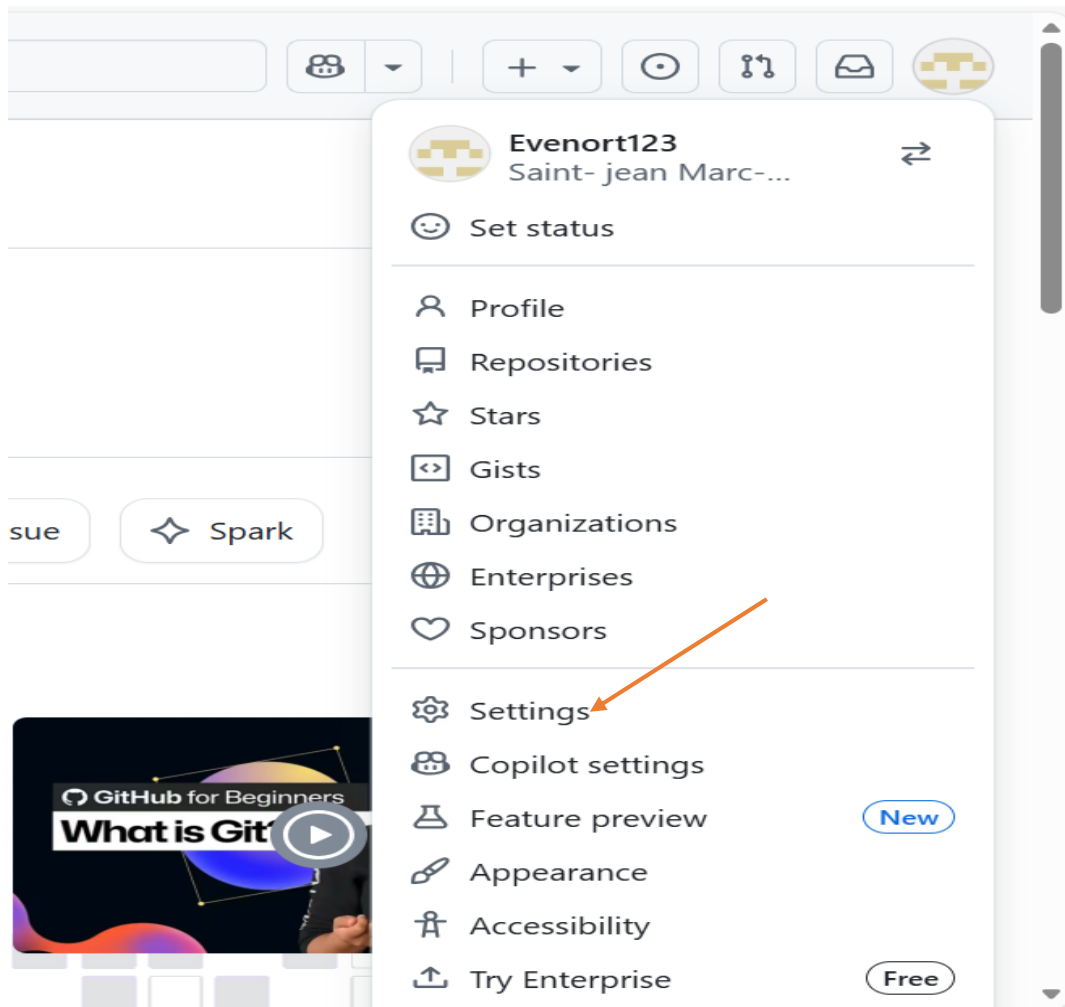
```
MINGW64:/c/Users/Paradoxe Inno/Desktop/TD
Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (master)
$ ssh-keygen -t rsa -b 4096 -C "Smarcevenort@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Paradoxe Inno/.ssh/id_rsa):
/c/Users/Paradoxe Inno/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase for "/c/Users/Paradoxe Inno/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Paradoxe Inno/.ssh/id_rsa
Your public key has been saved in /c/Users/Paradoxe Inno/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:OC0i+eW7Rxf4PWgKXfmFnAJYKjNz7TaSBHP2nM5B2vg Smarcevenort@gmail.com
The key's randomart image is:
+---[RSA 4096]---+
|      o.      |
|     .o.     |
|    = o .o o o|
|   B o . = + .|
|  . B.XoS* .  |
| o o.B@=.+.  |
|o  .oo+* .   |
| . o  o+ .   |
|  . ++ E     |
+-----[SHA256]-----+

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (master)
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADEPTz3IEFnEudo5QRCUDPos01NstYxT7X02D+Nb72u
gUAsx30M3quEdihFm5g6XRzdodEpxFET0o1zph5Xm6IdpPEywmYYaX5Cv9s3wfi6f1CnMsKS+LXveeaB
BcVGmX9BApjOLTU/yhD5rD8Gw+EHa/LaQo1kwdGJeDRwjzrceToHSzjdP0eh2QyRhwJhG8E1c/yGM0Gx
qAYacPetWN9YUHSBDMNiQTHJXFV75nD1MpLHXGnlb3w62/9za5BuUFv8YhtJy5ze6OI0ju050Y0+Ug2J
eY40w4+IQ7+dsJa2hw2Ohg+we2jjuOW7B+nQrt1ofEmz6Xmy0rqGyRJa8Zc++XrtEtW/dS0VOzWcB3
p5FIwZAhSPwHbzVfay5Uk+isjvmm+8lFP+ILH3/uEYy1rSk292FpgBjcwQC33a/3Axx+HYFoPDm/O9yr
SeNb23/aaQH20CPSrYHw2A3HdekLtYfONf8V613Man8m1lud/VcMDrt5A2UmvoqCHa8reLgzkkB+mv8
k0HqXCj1l6Y/bsyikjGn2AE0h4fcEUqtp2KDdcupkSDmlo/soXjprQIWT+f1EC8f3AFbazRyQC0qzeFd
HCpPWOGWvaCUwqxpq9sNwy4DRhjuOLjsO/a7tEUjYOCNxbofyCzWu/ysVD+tdjGZj3FFEgYl+ZoTX3xG
8w== Smarcevenort@gmail.com

Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (master)
$
```

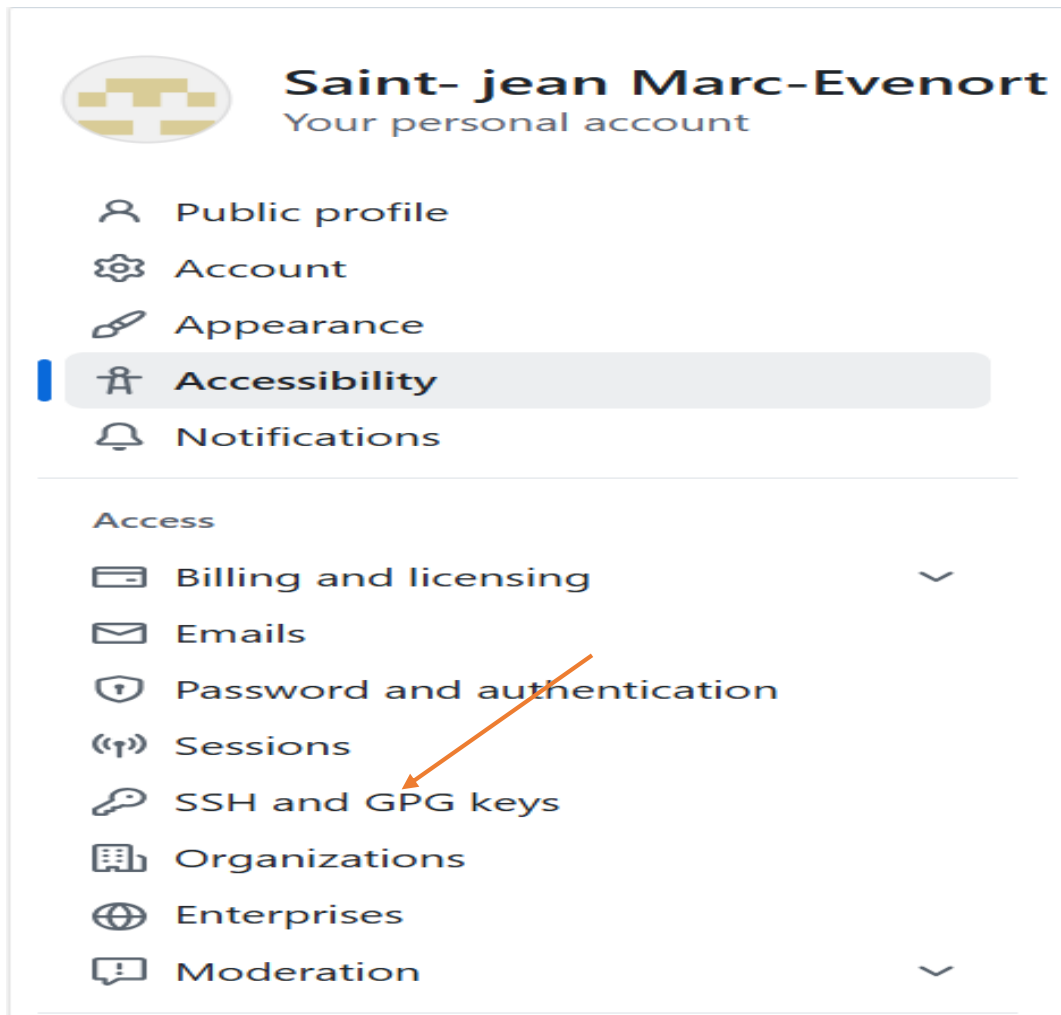
- Cette image nous montre la clé [SHA256] et la clé publique qui permettent de créer une connexion SSH sécurisée entre l'ordinateur et un serveur distant. La commande `cat ~/.ssh/id_rsa.pub` génère une paire de clés (une clé privée et une clé publique).

H) Accéder aux paramètres de GitHub



- En cliquant sur le profil de l'utilisateur, puis sur settings (paramètres), on accède au paramètres GitHub. dans cet endroit on trouve **"SSH et GPG keys"**, qui permet d'ajouter la clé SSH générée précédemment afin d'établir une connexion sécurisée entre Git et GitHub.

I) Accéder à la section SSH and GPG keys



- Dans cette image, on voit la zone fléchée indiquant la section **“SSH and GPG keys”**. En cliquant sur cette option, on accède à la page permettant d’ajouter la clé SSH générée précédemment dans le terminal. Elle permet aussi d’établir une connexion sécurisée entre Git et GitHub.

J) Ajouter la clé SSH sur GitHub

Saint- jean Marc-Evenort (Evenort123)
Your personal account

Go to your personal profile

Public profile
Account
Appearance
Accessibility
Notifications

Access

Billing and licensing
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations
Enterprises
Moderation

Add new SSH Key

Title
Mon Projet

Key type
Authentication Key

Key

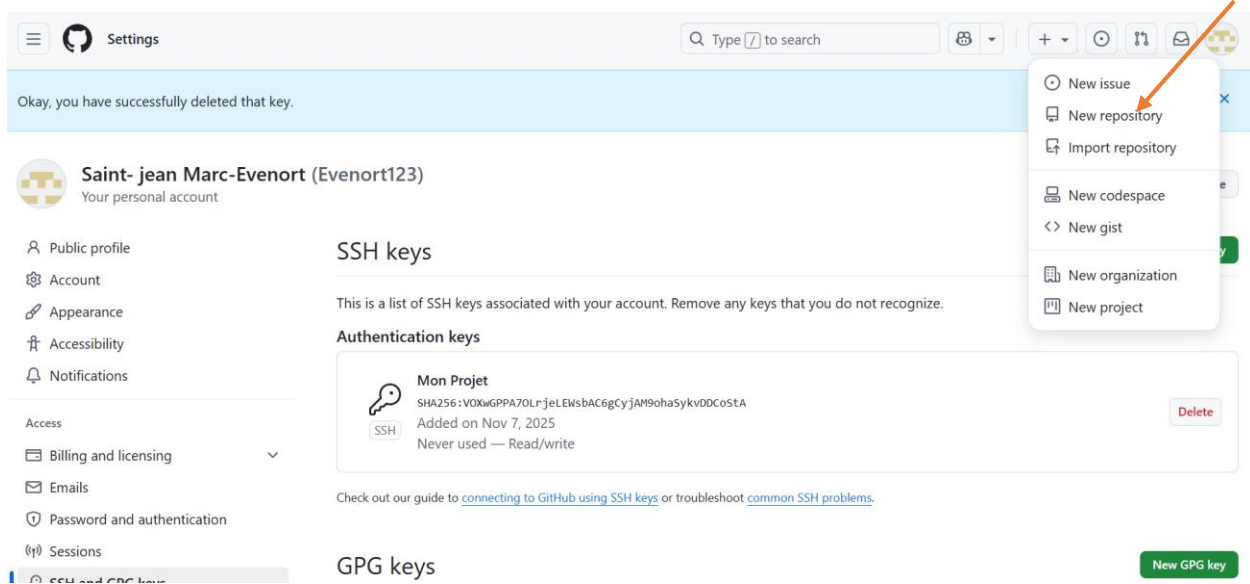
```
ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQADEptz3IEFnEudo5QRCUDPos01NstYxT7X02D+Nb72ugUAsx3OM3quEdihFm5g6XRzdodEpxFET0o1  
zph5Xm6ldpPEyWmYYaX5Cv9s3wfi6fCnMsKS+LXveeaBBcVGmX9BApjOLTU/yhD5rD8Gw+EHa/LaQolkwdGJeDRwjzrceToHSzjdP0eh2QyRhwJ  
hG8E1c/yGM0GxqAYacPetWN9YUHS8DMNiQTHJXFV75nDlMpLHXGnlb3w62/9za5BuUFv8Yhtly5ze6OI0ju050YO+Ug2JeY4Qw4+IQ7+dsJa2h  
w2Ohg+we2juOW7B+nQrt1ofEmz6Xmy0rqGyRJa8Zc++XrttEtW/dS0VozWcB3p5FlwZAhsPWHbzVfay5Uk+isjvmm+8IFP+ILH3/uEYy1rSk292  
FpgBjcWQC33a/3Axx+HYFoPDm/O9yrSeNb23/aaQH20CPSrYHw2A3HdekLTYfONf8V613Man8m1lcud/VCMdrt5A2UmvoqCHa8reLgzkB+mv  
8k0HqXCj116Y/bsyikjGn2AEOh4fcEUqtp2KDcupkSDm1o/soXjpRQIWT+fIEC8f3AFbazRyQC0qzeFdHCpPWOGWvaCUwqxqq9sNwy4DRhjuOLj  
sO/a7tEUjYOCNxbofyCzWu/ysVD+tdjGZj3FFEgYl+ZoTX3xG8w== Smarcevenort@gmail.com
```

Add SSH key

- Dans cette image, j'ai créé une nouvelle clé SSH sous le nom "mon Projet". J'ai ensuite collé la clé publique générée dans le champ prévu à cet effet, puis cliqué sur le bouton "Add SSH key" pour l'enregistrer.

Cette étape permet à GitHub de reconnaître automatiquement mon ordinateur grâce à la clé SSH, et ainsi d'établir une connexion sécurisée sans mot de pass.

K) Créer un dépôt GitHub



- Dans Cette image, on voit le cheminement à suivre pour créer un nouveau dépôt sur GitHub.

L) Création du dépôt et configuration

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 *

 Evenort123 ▾

Repository name *

"TD"

✓ Your new repository will be created as -TD-.

The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. How about **potential-octo-happiness**?

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](#)

Off ☐


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](#)


 MIT License ▾

 -TD- Public

 Pin

 Watch 0 ▾

 Fork 0 ▾

 Star 0 ▾

 main ▾  1 Branch  0 Tags

 Go to file





 Code ▾



About



 Evenort123 Initial commit 9a04326 · 2 minutes ago 1 Commit

 LICENSE Initial commit 2 minutes ago

 README.md Initial commit 2 minutes ago

 README  MIT license

-TD-

No description, website, or topics provided.

 Readme

 MIT license

 Activity

 0 stars

 0 watching

 0 forks

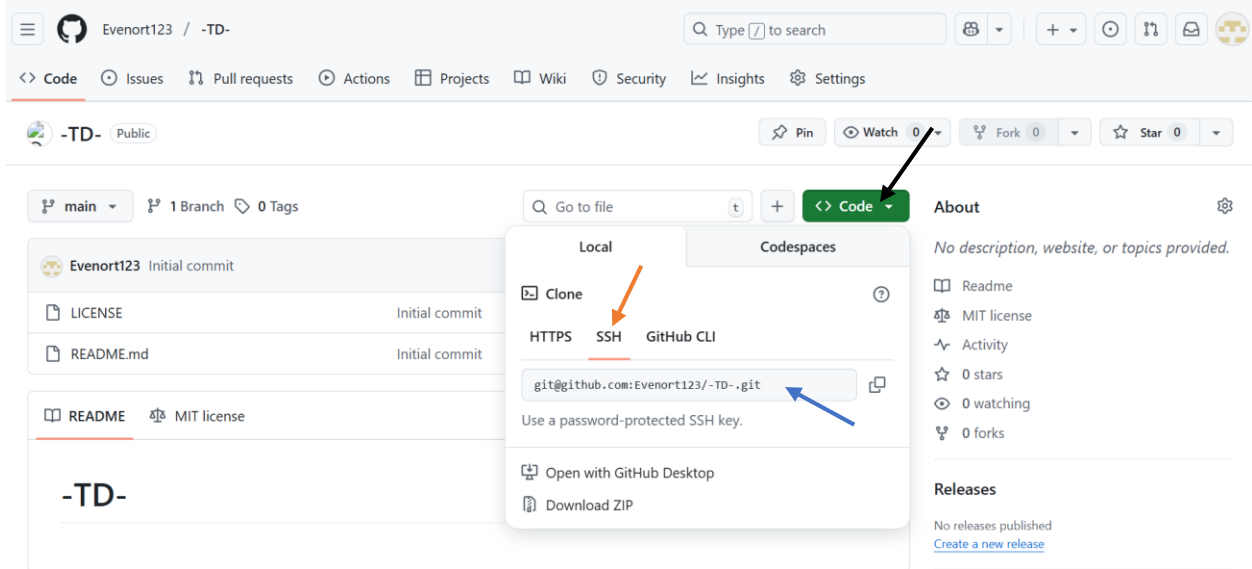
Releases

No releases published

[Create a new release](#)

- ces images nous montre le nom du dépôt (-TD-) ainsi la configuration utilisée lors de la création

8- Cloner un dépôt GitHub via SSH



- Cette image montre comment copier le code de clonage SSH afin de l'utiliser dans le terminal pour cloner le dépôt sur l'ordinateur local.

```
MINGW64:/c/Users/Paradoxe Inno/Desktop/TD
Paradoxe Inno@paradoxe MINGW64 ~/Desktop/TD (main)
$ cd ..

Paradoxe Inno@paradoxe MINGW64 ~/Desktop
$ git clone git@github.com:Evenort123/-TD-.git
Cloning into '-TD-'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
```

Ainsi, cette image montre clairement que le projet a été cloné avec succès sur la machine locale.

Conclusion

Ce TD m'a permis de comprendre et d'appliquer les bases de Git et GitHub.

Grâce à ce devoir j'ai appris comment manipuler le terminal sans utiliser l'interface graphique de mon ordinateur telles que:

Initialiser un dépôt Git, Créer un dépôt GitHub et le cloner, j'ai découvert les différentes commandes, ainsi que la manière de cloner et d'héberger un document sur GitHub.

Ce devoir m'a donné une source de motivation pour ce cours de système. malgré les difficultés rencontrées au cours de ce travail, je veux travailler davantage afin de mieux comprendre le reste du cours.

Même si j'ai rencontré quelques difficultés, je compte travailler encore plus, car dans la vie rien n'est facile.

Pour moi, la tâche est réussie et cette expérience m'a permis de progresser.

Dans ce travail dirigé, j'ai rencontré des difficultés énormes qui m'ont permis de mieux maîtriser le laboratoire.

Pour l'objectif je peux dire qu'il a été atteint.