

# Université de Technologie d'Haïti

## UNITECH

Facultés des sciences, de génie et d'architecture

---

TD N1-Git et GitHub

**Nom et prénom :** Shamaya LOUIS

**Niveau :** 4em année

**Date :** 19/01/2026

# Description des résultats de la tâche

## Objectif du TD

L'objectif de ce TD est de maîtriser les concepts de base de Git, de configurer un environnement de travail local, d'utiliser GitHub comme dépôt distant, et de réaliser une série d'opérations courantes (initialisation, commit, clonage, push, pull) via PowerShell.

## Démarche suivie

### 1. Installation de Git

- Téléchargement et installation de Git pour Windows.
- Vérification de la version via `git --version`.

### 2. Création d'un compte GitHub

- Inscription sur GitHub avec adresse email valide, mot de passe sécurisé et nom d'utilisateur.

### 3. Configuration de Git

- Configuration des identifiants globaux :  
`git config --global user.name "MonNom"`  
`git config --global user.email "mon.email@example.com"`

### 4. Test de Git avec PowerShell

- Création d'un dossier projet-git sur le Bureau.
- Initialisation d'un dépôt Git : `git init`.
- Création d'un fichier README.md et premier commit.

### 5. Génération d'une clé SSH

- Génération d'une paire de clés SSH avec l'algorithme ed25519 :  
`ssh-keygen -t ed25519 -C "mon.email@example.com"`
- Ajout de la clé publique au compte GitHub (Settings → SSH and GPG keys).

### 6. Création d'un dépôt GitHub

- Création d'un dépôt nommé mon-projet avec README initial.

### 7. Clonage du dépôt via SSH

- Clonage du dépôt sur le Bureau :

```
git clone git@github.com:username/mon-projet.git
```

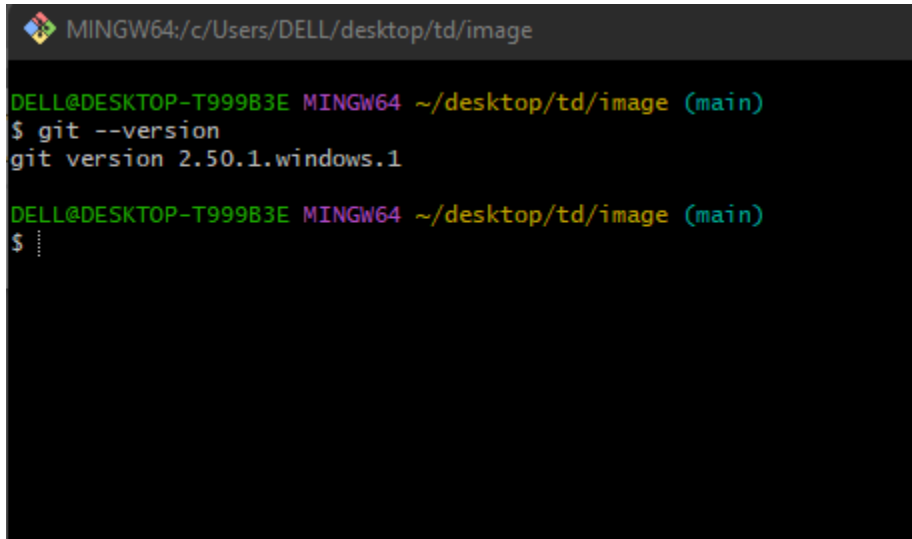
- Ajout d'un fichier Module.txt, commit et push vers main.

## 8. Création d'une structure pour le rapport

- Création des dossiers systeme, reseauI avec sous-dossiers image, projet, presentation.
- Ajout des fichiers (PDF et images) et push vers GitHub.

### Captures d'écran et explications

Vérification de l'installation de Git. La commande `git --version` affiche la version installée (2.46.0), confirmant que Git est opérationnel.

A screenshot of a terminal window with a dark background. The title bar at the top shows a Windows icon and the path 'MINGW64:/c/Users/DELL/desktop/td/image'. The terminal text shows a user prompt 'DELL@DESKTOP-T999B3E MINGW64 ~/desktop/td/image (main)' followed by the command '\$ git --version' and its output 'git version 2.50.1.windows.1'. Below this, the prompt is repeated, followed by '\$' and a vertical ellipsis '...'.

Premier commit local. Après `git init` et création du README, les commandes `git add` et `git commit` enregistrent le premier snapshot du projet.

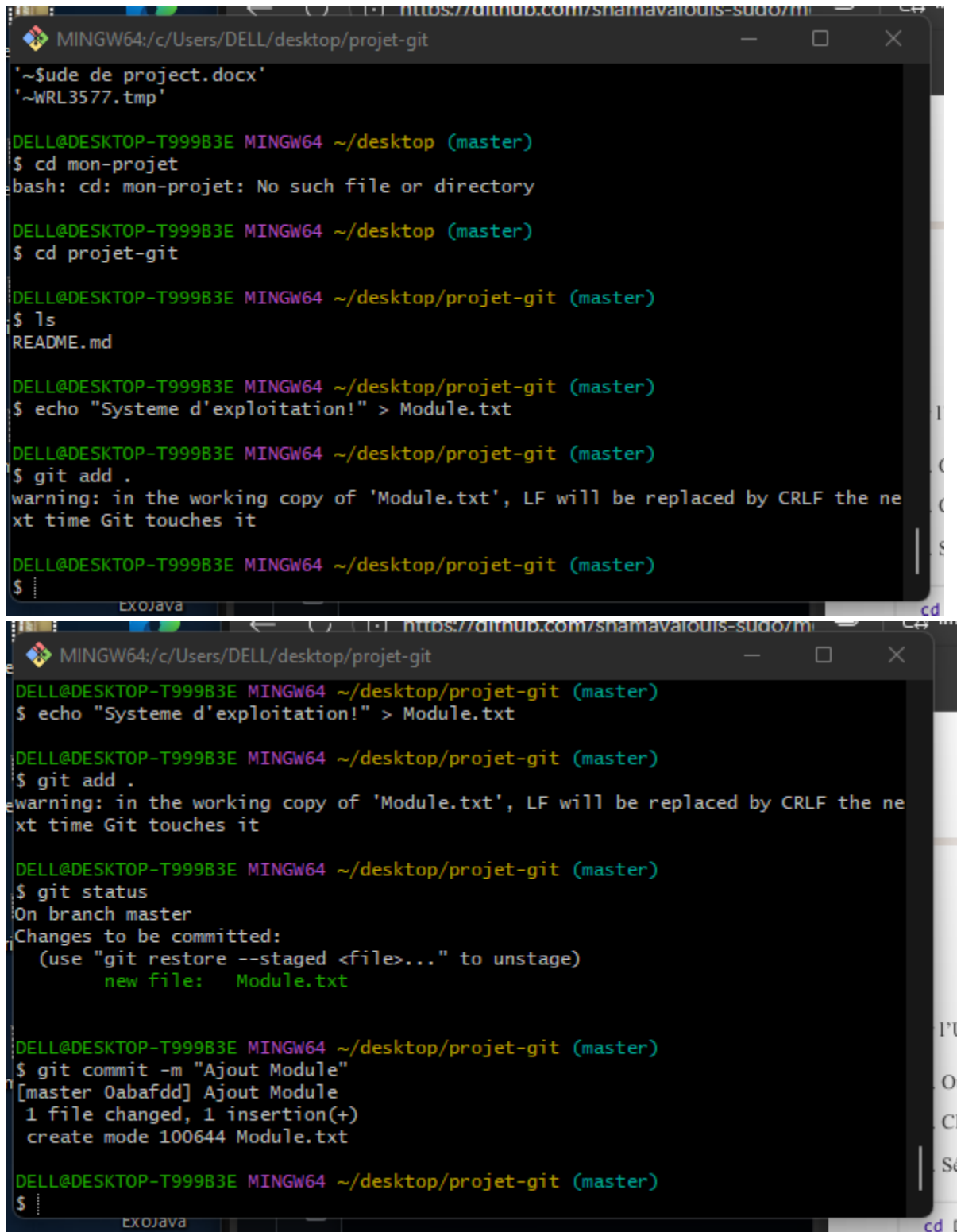
```
MINGW64:/c/Users/DELL/desktop/projet-git
DELL@DESKTOP-T999B3E MINGW64 ~
$ cd desktop
DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ mkdir projet-git
DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ cd projet-git
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$ git init
Initialized empty Git repository in C:/Users/DELL/Desktop/projet-git/.git/
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$ echo "Hello git!" > README.md
DELL@DESKTOP-T999B3E MINGW64 ~/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
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$
```

Clé SSH générée. La génération de clé SSH sécurisée permet une authentification sans mot de passe pour les opérations distantes.

```
MINGW64:/c/Users/DELL
$ cat ~/.ssh/id_ed25519
cat: /c/Users/DELL/.ssh/id_ed25519: No such file or directory
DELL@DESKTOP-T999B3E MINGW64 ~
$ ssh-keygen -t rsa -b 4096 -c "shamayalouis@gmail.com"
Too many arguments.
usage: ssh-keygen [-q] [-a rounds] [-b bits] [-C comment] [-f output_keyfile]
                 [-m format] [-N new_passphrase] [-O option]
                 [-t dsa | ecdsa | ecdsa-sk | ed25519 | ed25519-sk | rsa]
                 [-w provider] [-Z cipher]
ssh-keygen -p [-a rounds] [-f keyfile] [-m format] [-N new_passphrase]
            [-P old_passphrase] [-Z cipher]
ssh-keygen -i [-f input_keyfile] [-m key_format]
ssh-keygen -e [-f input_keyfile] [-m key_format]
ssh-keygen -y [-f input_keyfile]
ssh-keygen -c [-a rounds] [-C comment] [-f keyfile] [-P passphrase]
ssh-keygen -l [-v] [-E fingerprint_hash] [-f input_keyfile]
ssh-keygen -B [-f input_keyfile]
ssh-keygen -D pkcs11
ssh-keygen -F hostname [-lv] [-f known_hosts_file]
ssh-keygen -H [-f known_hosts_file]
ssh-keygen -K [-a rounds] [-w provider]
ssh-keygen -R hostname [-f known_hosts_file]
ssh-keygen -r hostname [-g] [-f input_keyfile]
```

Dépôt cloné et premier push. Le dépôt GitHub a été cloné en SSH, modifié localement

Structure du dépôt TD sur GitHub. Organisation des dossiers systeme et reseauI avec sous-dossiers pour images, projet et présentation.



```
MINGW64:/c/Users/DELL/desktop/projet-git
'$ude de project.docx'
'~WRL3577.tmp'

DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ cd mon-projet
bash: cd: mon-projet: No such file or directory

DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ cd projet-git

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$ ls
README.md

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$ echo "Systeme d'exploitation!" > Module.txt

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$ git add .
warning: in the working copy of 'Module.txt', LF will be replaced by CRLF the next time Git touches it

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$ echo "Systeme d'exploitation!" > Module.txt

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$ git add .
warning: in the working copy of 'Module.txt', LF will be replaced by CRLF the next time Git touches it

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Module.txt

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$ git commit -m "Ajout Module"
[master 0abafdd] Ajout Module
1 file changed, 1 insertion(+)
create mode 100644 Module.txt

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$
```

```
MINGW64:/c/Users/DELL/desktop/projet-git
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Module.txt

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$ git commit -m "Ajout Module"
[master 0abafdd] Ajout Module
1 file changed, 1 insertion(+)
create mode 100644 Module.txt

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (master)
$ git branch -M main

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (main)
$ git push -u origin main
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (main)
$
```

```
MINGW64:/c/Users/DELL/desktop/td
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (main)
$ cd desktop
bash: cd: desktop: No such file or directory

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/projet-git (main)
$ cd ..

DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ cd ..

DELL@DESKTOP-T999B3E MINGW64 ~
$ cd desktop

DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ mkdir td

DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ cd td

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/td (master)
$ mkdir systeme

DELL@DESKTOP-T999B3E MINGW64 ~/desktop/td (master)
$
```

```
MINGW64:/c/Users/DELL/desktop
DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ mkdir td
DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ cd td
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/td (master)
$ mkdir systeme
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/td (master)
$ mkdir image
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/td (master)
$ mkdir projet
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/td (master)
$ mkdir presentation
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/td (master)
$ cd ..
DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$
```

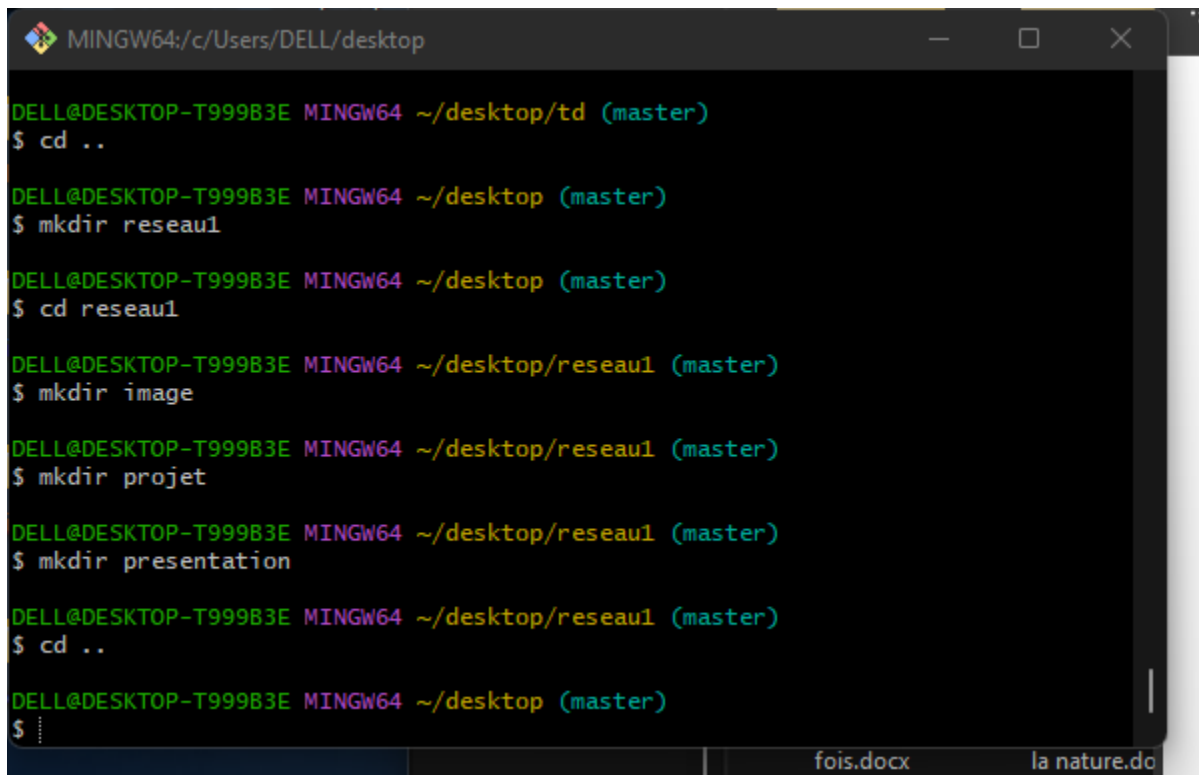
*P Sudo apt install traceroute* mise a du paquets et pouvoir utiliser traceroute

*du -sh* affiche le nombre de blocs utilises pour les fichiers

*free -h* affiche des informations de disponible sur la mémoire vive

*Ls cybersec* verifier que le contenu soit vide

*ip a* pour afficher les interfaces réseau, l'adresse IP et l'état de la connexion



```
MINGW64; c:/Users/DELL/desktop
DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ cd ..
DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ mkdir reseau1
DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ cd reseau1
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/reseau1 (master)
$ mkdir image
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/reseau1 (master)
$ mkdir projet
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/reseau1 (master)
$ mkdir presentation
DELL@DESKTOP-T999B3E MINGW64 ~/desktop/reseau1 (master)
$ cd ..
DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ ..
DELL@DESKTOP-T999B3E MINGW64 ~/desktop (master)
$ ..
```

## Conclusion

Je suis arrivé à maitriser les concepts de base de Git (working directory, staging area, repository) ainsi que la gestion des dépôts locaux et distants via PowerShell. J'ai utilisation de SSH pour une connexion sécurisée à GitHub, une structuration d'un projet avec branches, commits

**Difficultés rencontrées** avec la commande :

```
sh-keygen -t ed25519 -C "ton.email@example.com"
```

```
cat ~/.ssh/id_ed25519
```