

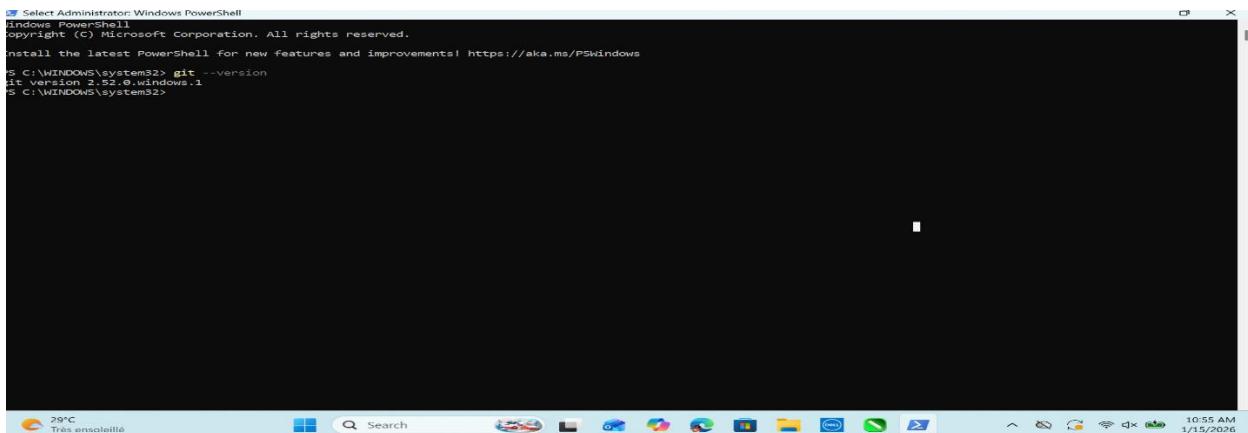


# UNIVERSITÉ DE TÉCHNOLOGIE D'HAÏTI

## (UNITECH)

# Faculté des Sciences de Génie, et d'Architecture

NOM.....MAITRE  
PRÉNOM.....Amourana  
NIVEAU.....2  
DATE.....le/20/01/25

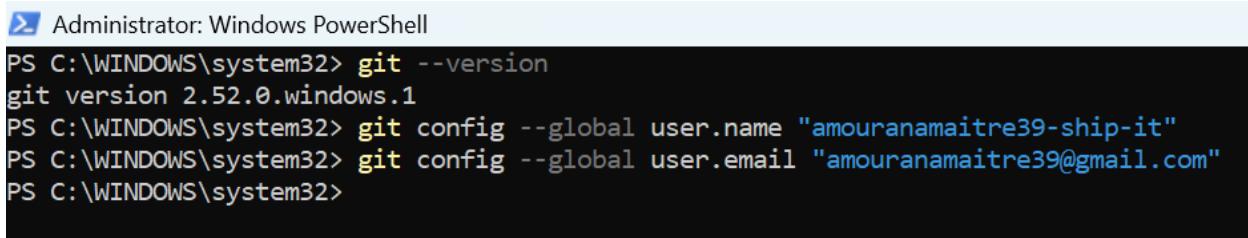


```
PS Select Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

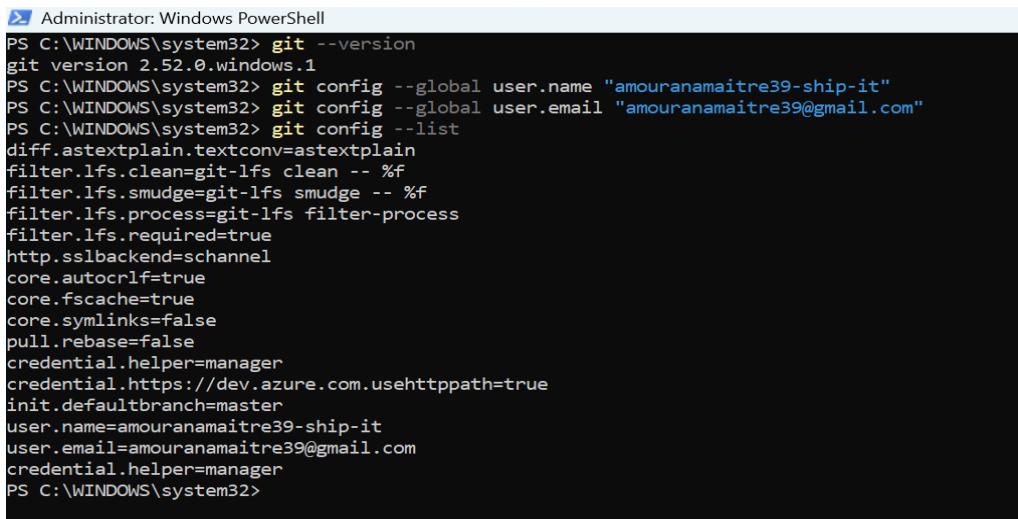
PS C:\WINDOWS\system32> git --version
git version 2.52.0.windows.1
PS C:\WINDOWS\system32>
```

La commande `git --version`, permet de vérifier la version de Git installée sur votre ordinateur.



```
PS Select Administrator: Windows PowerShell
PS C:\WINDOWS\system32> git --version
git version 2.52.0.windows.1
PS C:\WINDOWS\system32> git config --global user.name "amouranamaitre39-ship-it"
PS C:\WINDOWS\system32> git config --global user.email "amouranamaitre39@gmail.com"
PS C:\WINDOWS\system32>
```

Les commandes `git config --user.name` et `git config --global user.email` servent à configurer l'identité de l'utilisateur afin d'identifier les auteurs des commits sur GitHub.



```
PS Select Administrator: Windows PowerShell
PS C:\WINDOWS\system32> git --version
git version 2.52.0.windows.1
PS C:\WINDOWS\system32> git config --global user.name "amouranamaitre39-ship-it"
PS C:\WINDOWS\system32> git config --global user.email "amouranamaitre39@gmail.com"
PS C:\WINDOWS\system32> 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=amouranamaitre39-ship-it
user.email=amouranamaitre39@gmail.com
credential.helper=manager
PS C:\WINDOWS\system32>
```

La commande `git config --list` affiche la liste complète des paramètres de configuration Git actifs sur la machine ou dans le dépôt.

```

Windows PowerShell
credential.helper=manager
PS C:\Users\amour> ls

Directory: C:\Users\amour

Mode                LastWriteTime         Length Name
—
d----        12/3/2025   5:42 PM          .android
d----        12/12/2025  8:12 AM          .dotnet
d----        11/21/2025  8:29 AM          .gradle
d----        11/15/2025  8:08 AM          .idlerc
d----        12/12/2025  9:07 AM          .nuget
d----        11/28/2025  1:31 PM          .skiko
d----        12/8/2025   8:54 AM          .templateengine
d----        1/9/2026    5:12 PM          .VirtualBox
d----        8/11/2025   8:54 AM          .vscode
d----        11/28/2025  12:26 PM          AndroidStudioProjects
d----        11/14/2025  1:33 PM          Cisco Packet Tracer 8.2.0
d-r--      11/4/2025   1:12 PM          Contracts
d-r--      1/15/2026  9:55 AM          Desktop
d-r--      1/9/2026   4:32 PM          Documents
d-r--      1/14/2026  10:21 AM          Downloads
d-r--      11/4/2025  1:12 PM          Favorites
d-r--      11/4/2025  1:12 PM          Links
d-r--      11/4/2025  1:12 PM          Music
d-r--      8/8/2024   1:04 PM          OneDrive

```

La commande Ls affiche le contenu(fichier et dossiers)du dossier où l'on trouve.

```

Windows PowerShell
PS C:\Users\amour> cd Desktop
PS C:\Users\amour\Desktop> mkdir projet-git

Directory: C:\Users\amour\Desktop

Mode                LastWriteTime         Length Name
—
d----        1/15/2026  11:24 AM          projet-git

PS C:\Users\amour\Desktop> cd projet-git
PS C:\Users\amour\Desktop\projet-git>

```

La commande cd Desktop permet de se déplacer dans le dossier Desktop(Bureau)

Mkdir projet-git;permet de créer un nouveau dossier nommé projet git.

Cd projet-git;permet d'entrer dans le dossier projet-git.

```

PS C:\Users\amour\Desktop> cd projet-git
PS C:\Users\amour\Desktop\projet-git> git init
Initialized empty Git repository in C:/Users/amour/Desktop/projet-git/.git/
PS C:\Users\amour\Desktop\projet-git> echo "Hello Git!" > README.MD
PS C:\Users\amour\Desktop\projet-git> git add README.MD
PS C:\Users\amour\Desktop\projet-git> git commit -m "Premier commit avec powershell"
[master (root-commit) b379140] Premier commit avec powershell
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.MD
PS C:\Users\amour\Desktop\projet-git> git status

```

Git init: initializer un depot Git

Git add:prepare les fichier pour le commit

Git commit:enregistre les modification

Git status:affiche l'état du depot.

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> git --version
git version 2.52.0.windows.1
PS C:\WINDOWS\system32> git config --global user.name "amouranamaitre39-ship-it"
PS C:\WINDOWS\system32> git config --global user.email "amouranamaitre39@gmail.com"
PS C:\WINDOWS\system32> 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.httpss://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=amouranamaitre39-ship-it
user.email=amouranamaitre39@gmail.com
credential.helper=manager
PS C:\WINDOWS\system32>
```

Git config –global user.name: Elle permet de configurer le nom de l'utilisateur Git.

Git config –global user,email: Elle permet de configurer l'adresse email associée à Git.

Git commit -m: Enregistre les modifications ajoutées dans Git avec un message descriptif.

Git init: Initialise un nouveau dépôt Git dans le dossier courant.

Git add README.md: Ajoute le fichier README.md à la Zone de Préparation pour le commit.

```
nothing to commit, working tree clean
PS C:\Users\amour\Desktop\projet-git> git log
commit b379140783fd6fcfaa19ee12003d38ce03a63882 (HEAD -> master)
Author: amouranamaitre39-ship-it <amouranamaitre39@gmail.com>
Date:   Thu Jan 15 11:35:44 2026 -0500

    Premier commit avec powershell
PS C:\Users\amour\Desktop\projet-git> |
```

La commande git log: Permet d'afficher l'historique des commits d'un dépôt Git.

```

PS C:\Users\amour> ssh-keygen -t ed25519 -C "amouranamaitre39@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:/Users/amour/.ssh/id_ed25519):
Created directory 'C:\\Users\\amour/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:/Users/amour/.ssh/id_ed25519
Your public key has been saved in C:/Users/amour/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:LHLFugrFLbfTlf0vAtedLCBD02BLH6E3/UqNbKgtA0 amouranamaitre39@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|          o =+.. |
|         . o * ... |
|         . o. + +... |
|         . . +E o.+.= . |
|= *.S=++oo+ o |
| . = ==o..=oo. |
| . . + . . *.. |
| . . . . + o |
+---[SHA256]----+
PS C:\Users\amour>

```

ss-keygen crée une clé privée et une clé publique pour s'authentifier de manière sécurisée (par exemple avec GitHub), sans utiliser de mot passe à chaque connexion.

```

PS C:\Users\amour> cat ~/.ssh/id_ed25519
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktkjEAAAAABG5vbmuAAAAEBm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACAnvupvdqvfrtrjA2dmqNrLQ52UcPnpj8+gjTCbrWUBcQAAAKDa0tIb2tLS
GwAAAAtzc2gtZWQyNTUxOQAAACAnvupvdqvfrtrjA2dmqNrLQ52UcPnpj8+gjTCbrWUBcQ
AAAEBcZ55/Wf10CgnRwkUiKdHPEckuEE8PsERNqqc70dGvtye+6m92q99G1GMDZ2ao2stD
nZrw+emPz6CNMJutZQFxAAAAGmFtb3VYw5hbWFpdHJlMzIAZ21haWwuY29tAQID
-----END OPENSSH PRIVATE KEY-----
PS C:\Users\amour>

```

La commande cat~/.ssh/id\_ed25519 permet d'afficher le contenu de la clé privée SSH dans le terminal.

The screenshot shows the GitHub Settings interface for the user 'amouranamaitre39-ship-it'. The user is adding a new SSH key. The 'Title' field contains 'key Win'. The 'Key type' dropdown is set to 'Authentication Key'. The 'Key' field displays the content of the SSH private key generated earlier, starting with '-----BEGIN OPENSSH PRIVATE KEY-----' and ending with '-----END OPENSSH PRIVATE KEY-----'. The left sidebar shows various account settings like Public profile, Account, Appearance, Accessibility, Notifications, Billing and licensing, Emails, Password and authentication, Sessions, and SSH and GPG keys, with 'SSH and GPG keys' currently selected.

```
PS C:\Users\amour> ssh -T git@github.com
The authenticity of host 'github.com (140.82.112.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
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.
Hi amouranamaitre39-ship-it! You've successfully authenticated, but GitHub does not provide shell access.
PS C:\Users\amour>
```



La commande ssh permet de se connecter de manière sécurisée à une machine distante via le protocole SSH.

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 \*  / Repository name \*   
 Your new repository will be created as Projet-Intra.  
 The repository name can only contain ASCII letters, digits, and the characters ., -, and \_.

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

Description  
  
 18 / 350 characters

**2 Configuration**

Choose visibility \*  Choose who can see and commit to this repository

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

```
PS C:\Users\amour\Desktop> git clone git@github.com:amouranamaitre39-ship-it/Projet-intra.git
Cloning into 'Projet-intra'...
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.
PS C:\Users\amour\Desktop>
```

La commande git clone permet de copier ou télécharger un dépôt Git distant sur son ordinateur.

```
PS C:\Users\amour\Desktop> ls

Directory: C:\Users\amour\Desktop

Mode                LastWriteTime         Length Name
----                -----        ----
d----           1/27/2025  12:35 PM          all_vc_redist_x86_x64
d----           11/30/2025  8:27 AM          Android
d----           11/5/2025   11:18 AM          CorelDRAW Graphics Suite 2024 v25.0.0.230 (x64) + Fix {CracksHash}
d----           7/14/2025   10:30 AM          cours_bootstrap
d----           12/18/2024  10:19 AM          Cours MySql
d----           11/15/2025  9:15 AM          CoursPython
d----           8/12/2025   5:06 PM          Devoir AIME
d----           1/31/2025  11:24 AM          footer_foto
d----           12/11/2024  6:11 PM          jjj
d----           12/11/2024  12:06 PM          Liste
d----           2/12/2025  10:56 AM          Liste des monuments
d----           1/7/2025    8:04 PM          Liste des Monuments.html
d----           1/7/2025    7:36 PM          New folder
d----           11/10/2025  12:29 PM          New folder (2)
d----           1/15/2026   11:33 AM          projet-git
d----           1/15/2026   2:02 PM          Projet-intra
d----           2/12/2025  10:43 AM          Td formulaire
d----           12/9/2025   9:43 AM          TD SITE WEB
d----           11/10/2025  12:16 PM          TP
```

La commande ls permet d'afficher la liste des fichiers et dossiers présent dans le répertoire courant.

```
PS C:\Users\amour\Desktop> cd Projet-intra
PS C:\Users\amour\Desktop\Projet-intra> echo "Système d'exploitation!" > Module.txt
PS C:\Users\amour\Desktop\Projet-intra> git add .
PS C:\Users\amour\Desktop\Projet-intra> 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

PS C:\Users\amour\Desktop\Projet-intra> |
```

Cd Projet-intra: Permet de se déplacer dans le dossier Projet-intra.

Echo “système d’exploitation!”>Module.txt: Crée un fichier Module.txt et y écrit le texte “système d’exploitation”.

Git add: Préparer les fichiers pour le prochain commit en les ajoutant à la zone de staging.

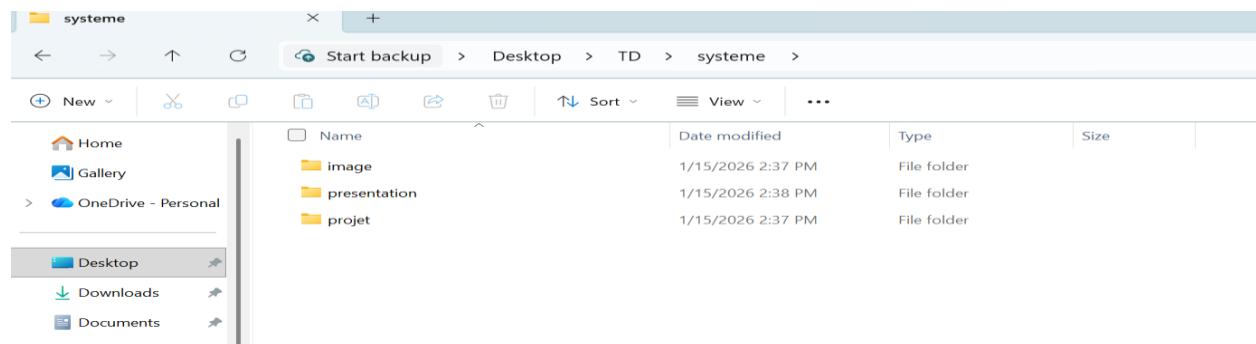
Git status: Affiche l’état actuel du dépôt Git, montrant les fichiers.

```
PS C:\Users\amour\Desktop\Projet-intra> git commit -m "Ajout module"
[main a55988c] Ajout module
 1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 Module.txt
PS C:\Users\amour\Desktop\Projet-intra> git branch -M main
PS C:\Users\amour\Desktop\Projet-intra> git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 329 bytes | 164.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:amouranamaitre39-ship-it/Projet-intra.git
  e42e161..a55988c main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\amour\Desktop\Projet-intra> |
```

Git commit -m: Enregistre définitivement les modifications ajoutées dans la git avec le message “Ajout module”.

Git branch-M main: Renomme la branche courante en main.

Git push-u origin main : Envoie les commits locaux vers le dépôts GitHub sur la branche main.



```
PS C:\Users\amour\Desktop\TD\ReseauI> ls

Directory: C:\Users\amour\Desktop\TD\ReseauI

Mode                LastWriteTime         Length Name
----                -----                  ---- 
d-----        1/15/2026   2:40 PM            image
d-----        1/15/2026   2:41 PM            presentation
d-----        1/15/2026   2:40 PM            projet

PS C:\Users\amour\Desktop\TD\ReseauI> |
```

La commande ls permet d'afficher la liste des fichiers et dossiers contenus dans le répertoire courant.

```
PS C:\Users\amour\Desktop\TD> cd systeme
PS C:\Users\amour\Desktop\TD\systeme> ls

Directory: C:\Users\amour\Desktop\TD\systeme

Mode                LastWriteTime         Length Name
----                -----                  ---- 
d-----        1/15/2026   2:37 PM            image
d-----        1/15/2026   2:38 PM            presentation
d-----        1/15/2026   2:37 PM            projet

PS C:\Users\amour\Desktop\TD\systeme> |
```

La commande cd permet de changer de répertoire.

CONCLUSION: Au cours de ce travail, j'ai appris à utiliser Git et GitHub pour gérer les versions d'un projet. J'ai appris à créer un dépôt, initialiser un projet avec git init, ajouter des fichiers avec git commit, et envoyer le projet sur GitHub avec git push.

Résultat de la tache: La tache a été réussie le dépôt GitHub est créé, les fichiers sont bien organisés et les commits sont visibles en ligne.

Dificultés et solutions: Les principaux difficultés rencontrées concernaient la configuration de Git et l'utilisation des commandes de base. Ces problèmes ont été résolus en configurant correctement le nom et l'email Git et en vérifiant l'état du dépôt avec git status.