UNIVERSITÉ TECHNOLOGIE D'HAÏTI

(UNITECH)

Faculte de Genie et d'architechture

Departement des Sciences Informatiques

Niveau: L4

Prepare par: Jacquesima Jean Raynold

Propose par: Ismael Saint Amour

Dans le cadre du cours sécurité et cybersécurite Informatique

Date: 21-12-25

I..Description des résultats de la tâche

a) Expliquer en quelques lignes ce que tu devais faire (objetif du TD)

L'objectif de ce travail dirigé était d'apprendre à utiliser **Git** et **GitHub** pour gérer un projet informatique. Il fallait comprendre comment suivre les modifications d'un projet, enregistrer les versions et héberger les fichiers en ligne sur GitHub afin de pouvoir les partager et les sauvegarder..

. b) Décrire la démarche suivie

Pour réaliser ce TD, nous avons d'abord installé et configuré Git sur l'ordinateur. Ensuite, nous avons créé un compte GitHub et généré une clé SSH pour connecter notre machine à GitHub. Après cela, nous avons créé un dépôt GitHub nommé TD, que nous avons cloné sur le bureau. Nous avons ensuite créé les dossiers demandés, ajouté le rapport en PDF et les images. Enfin, nous avons utilisé les commandes git add, git commit et git push pour envoyer tous les fichiers vers GitHub.II.. Résultat de l'éxécution des commande

```
$ git config --global user.email "jacquesimajeanraynold@gmail.com"
```

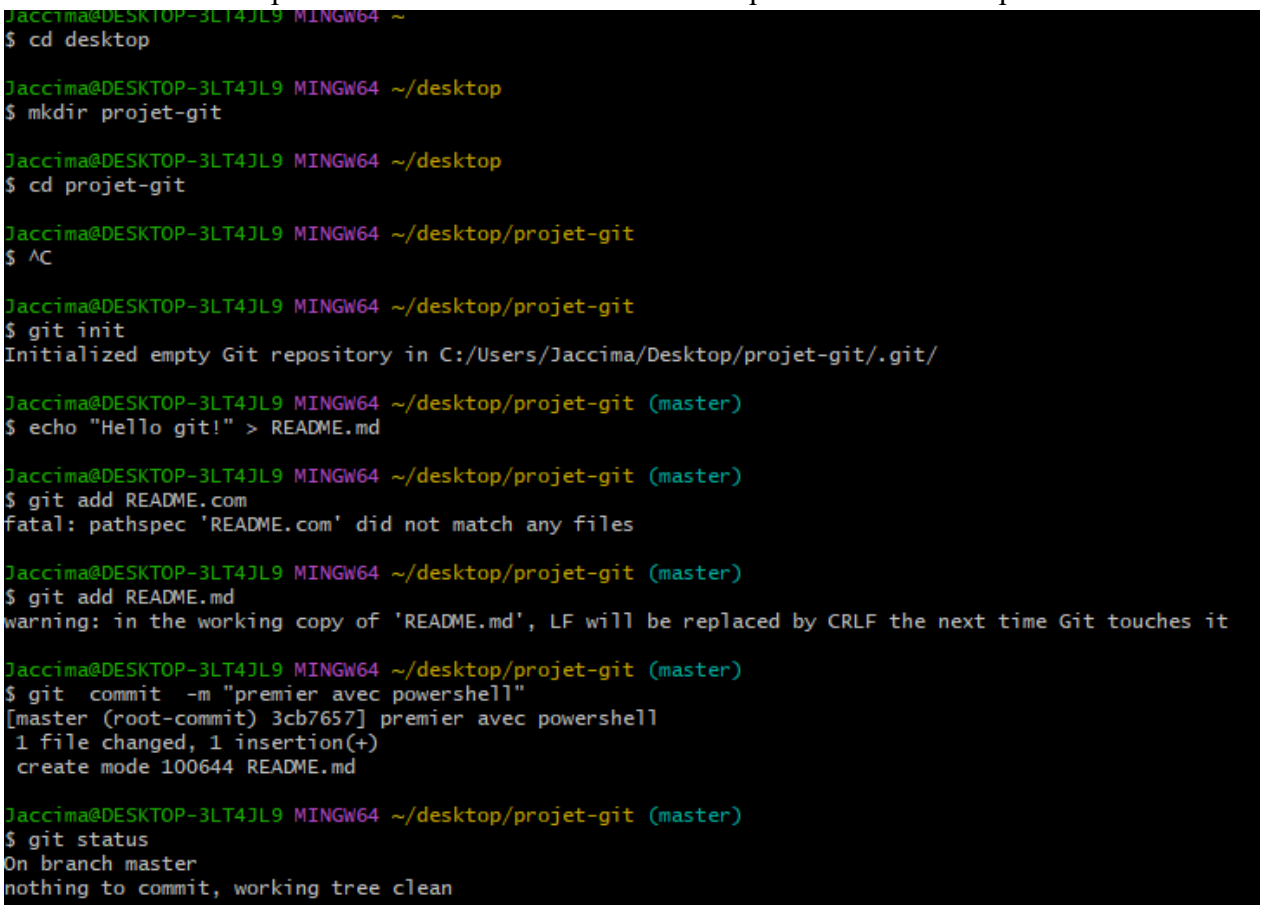
Elle permet de configure l'identite de l'utilisateur dans le Git.

```
$ git config --list
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
http.sslbackend=openssl
diff.astextplain.textconv=astextplain
credential.helper=manager
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=jacquesimajeanraynold@gmail.com
```

Elle permet d'afficher tous les configurations Git actives sur la machine .

```
$ ls
'3D Objects'/                Favorites/              NTUSER.DAT                                                                      SendTo@
 AppData/                    IntelGraphicsProfiles/  NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.blf                        Videos/
 Application Data'@          Links/                  NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000001.regtrans-ms  "Voisinage d'impression"@
 Cisco Packet Tracer 9.0.0'/ 'Local Settings'@       NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000002.regtrans-ms  'Voisinage réseau'@
 Contacts/                   'Menu Démarrer'@        OneDrive/                                                                       gestionEtudiant.txt
 Cookies@                    'Mes documents'@        Pictures/                                                                       ntuser.dat.LOG1
 Desktop/                    Modèles@                Recent@                                                                         ntuser.dat.LOG2
 Documents/                  Music/                  'Saved Games'/                                                                  ntuser.ini
 Downloads/                  Music.rar               Searches/
```

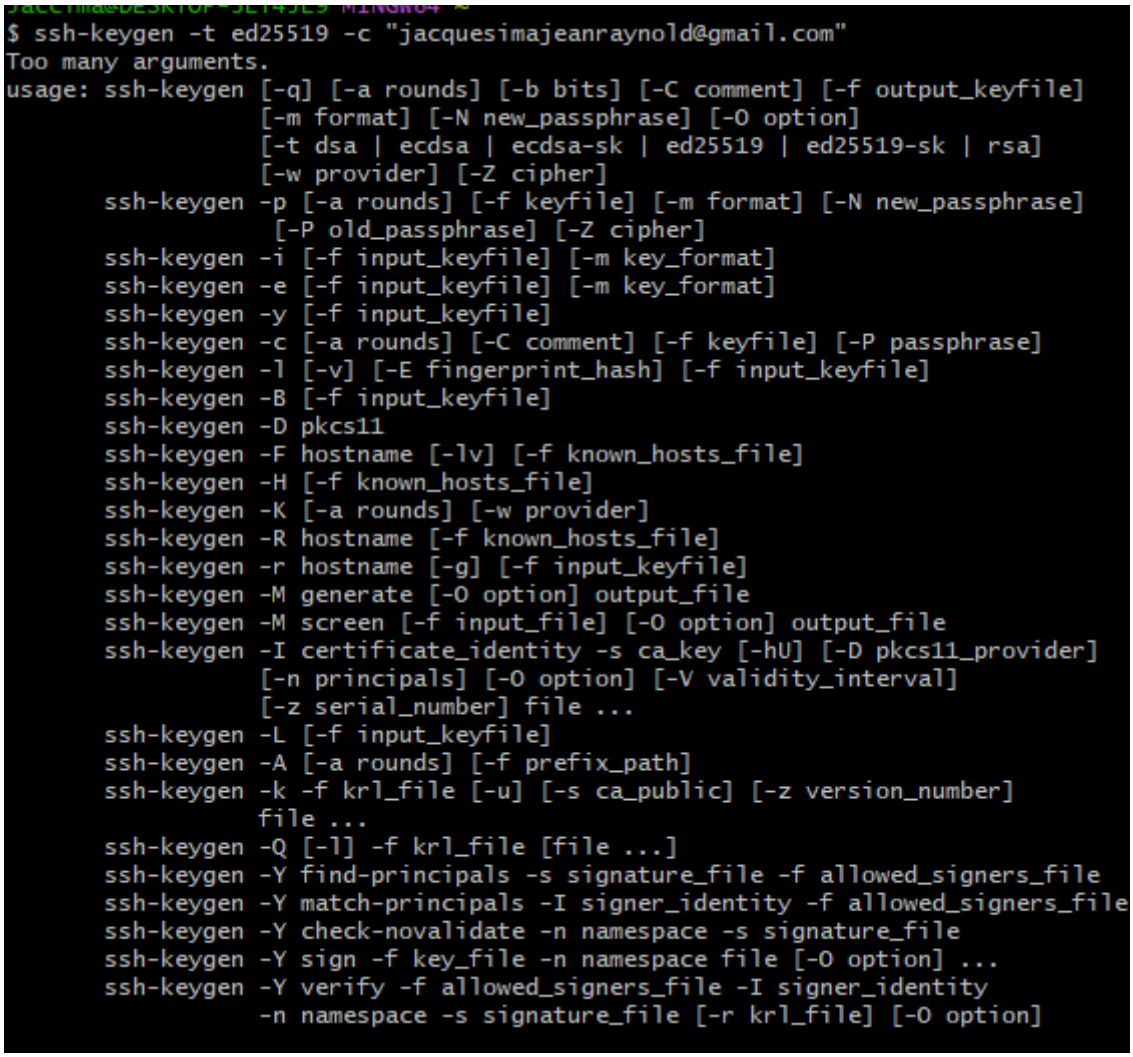
La commande ls permet d'afficher la liste des fichiers présentes dans le repertoire courant.



À travers ce travail, nous avons utilisé les commandes de base de Git afin de gérer un projet de manière efficace. Nous avons commencé par nous déplacer dans les répertoires à l'aide de la commande cd, puis nous avons initialisé un dépôt Git avec la commande git init. Ensuite, un fichier README.md a été créé à l'aide de la commande echo, puis ajouté au dépôt grâce à la commande git add.

La commande git commit nous a permis d'enregistrer le premier état du projet avec un message descriptif, tandis que la commande git status a servi à vérifier l'état du dépôt et à s'assurer que toutes les modifications ont bien été prises en

compte. Ce travail nous a permis de comprendre les principes fondamentaux du contrôle de version et l'importance de Git dans la gestion des projets informatiques.



La commande ssh-keygen -t ed25519 -C permet de générer une paire de clés SSH sécurisées afin d'authentifier l'utilisateur lors de connexions distantes, notamment pour l'accès à GitHub sans utilisation de mot de passe.

```
Jaccima@DESKTOP-3LT4JL9 MINGW64 ~
$ ssh-keygen -t ed25519 -C "jacquesimajeanraynold@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Jaccima/.ssh/id_ed25519):
Created directory '/c/Users/Jaccima/.ssh'.
Enter passphrase for "/c/Users/Jaccima/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Jaccima/.ssh/id_ed25519
Your public key has been saved in /c/Users/Jaccima/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:Y2l9BdCwBs+kkerG242YUC9CAJqulFF9KKNS4qMkzXo jacquesimajeanraynold@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|.. .. . o.++.    |
|o.+o o ..B ...   |
|+*..o ... =   .  |
|+== . o  +   .   |
|==.. + .S . .    |
|= E o =o.. .     |
|..   + * o       |
|      + o .      |
|                 |
+----[SHA256]-----+
```

La commande ssh-keygen -t ed25519 -C permet de générer une paire de clés SSH sécurisées associées à une adresse e-mail, afin de permettre une authentification sécurisée lors des échanges avec des plateformes comme GitHub.

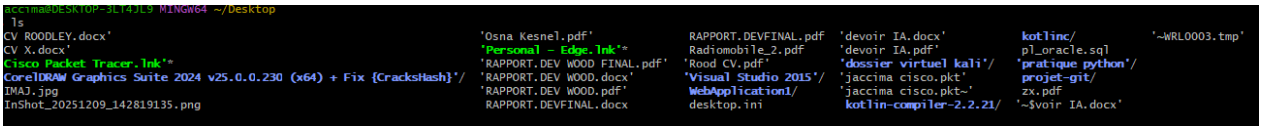
```
Jaccima@DESKTOP-3LT4JL9 MINGW64 ~
$

Jaccima@DESKTOP-3LT4JL9 MINGW64 ~
$ cat ~/.ssh/id_ed25519
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACBg8xgD7Ak/HRxKFhNCM3PSXFyGwa1QKCFqiwXReAtOgwAAAKiscj5zrHI+
cwAAAAtzc2gtZWQyNTUxOQAAACBg8xgD7Ak/HRxKFhNCM3PSXFyGwa1QKCFqiwXReAtOgw
AAAECHexyfOA2BY2eO/VKtBf55ER+xLRnGwEaEwVFkuyRCgWDzGAPsCT8dHEoWEOIzc9Jc
XIbBrVAoIWqLBdF4C3SDAAAAH2phY3F1ZXNpbWFqZWFucmF5bm9sZEBnbWFpbC5jb2OBAg
MEBQY=
-----END OPENSSH PRIVATE KEY-----
```

La commande $ cat ~/.ssh/id_ed25519 affiche **la clé privée SSH**, qui permet de s'authentifier sur un serveur ou GitHub. Cette clé est très sensible et ne doit jamais être partagée. Pour se connecter ou configurer GitHub, on utilise plutôt **la clé publique** (id_ed25519.pub).

```
$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGDzGAPsCT8dHEoWEOIzc9JcXIbBrVAoIWqLBdF4C3SD jacquesimajeanraynold@gmail.com
```

$ cat ~/.ssh/id_ed25519.pub affiche **la clé publique SSH**, que l'on peut partager pour s'authentifier sur un serveur ou GitHub. Contrairement à la clé privée, elle n'est pas secrète.



La commande ls permet d'afficher la liste des fichiers présentes dans le repertoire courant.

```
Jaccima@DESKTOP-3LT4JL9 MINGW64 ~/Desktop
$ cd mon-projet
bash: cd: mon-projet: No such file or directory

Jaccima@DESKTOP-3LT4JL9 MINGW64 ~/Desktop
$ echo "Systeme d'exploitation!" > module.text

Jaccima@DESKTOP-3LT4JL9 MINGW64 ~/Desktop
$ git add .
fatal: not a git repository (or any of the parent directories): .git

Jaccima@DESKTOP-3LT4JL9 MINGW64 ~/Desktop
$ git status
fatal: not a git repository (or any of the parent directories): .git

Jaccima@DESKTOP-3LT4JL9 MINGW64 ~/Desktop
$ git commit -m "Ajouter module"
fatal: not a git repository (or any of the parent directories): .git

Jaccima@DESKTOP-3LT4JL9 MINGW64 ~/Desktop
$ git branch -M main
fatal: not a git repository (or any of the parent directories): .git

Jaccima@DESKTOP-3LT4JL9 MINGW64 ~/Desktop
$ git push -u origin main
fatal: not a git repository (or any of the parent directories): .git

Jaccima@DESKTOP-3LT4JL9 MINGW64 ~/Desktop
$ cd TD
bash: cd: TD: No such file or directory

Jaccima@DESKTOP-3LT4JL9 MINGW64 ~/Desktop
$ mkdir systeme

Jaccima@DESKTOP-3LT4JL9 MINGW64 ~/Desktop
$ cd systeme
```

Ces commandes ont permis de :

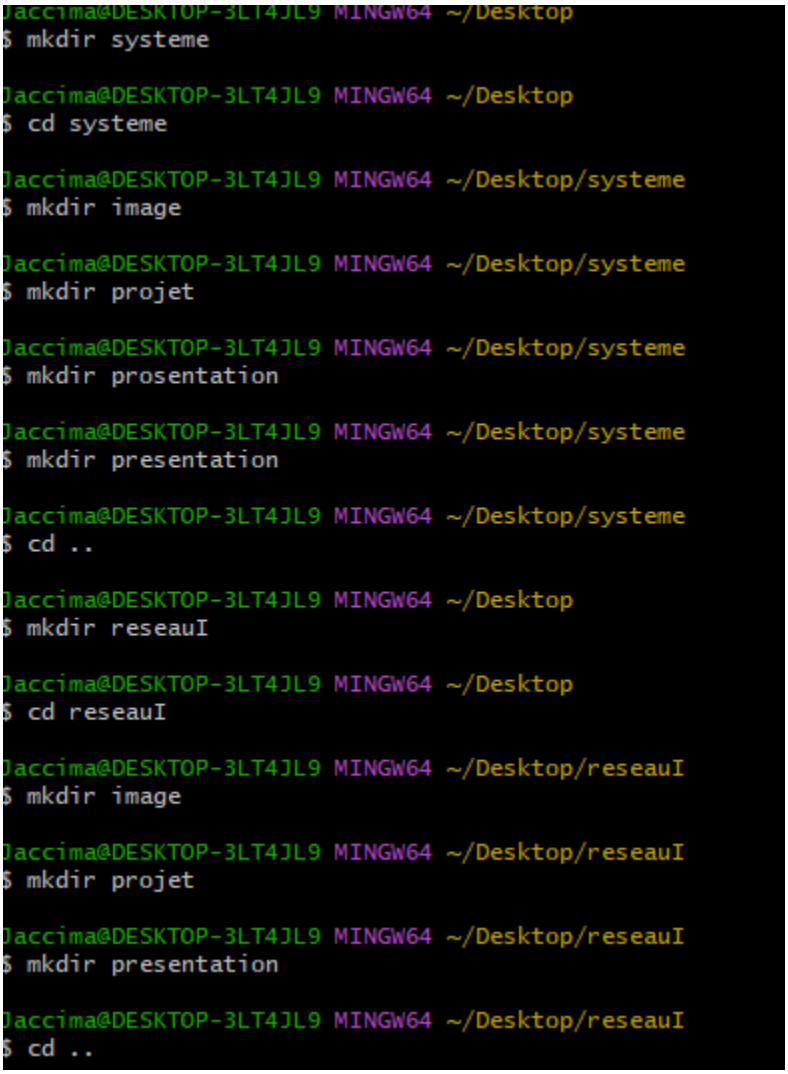
- Créer une arborescence de dossiers (systeme, projet, presentation, reseauI, image)

- Se déplacer entre les dossiers avec cd

- Créer un fichier texte Module.text avec du contenu

- Ajouter les fichiers au dépôt Git avec git add

- Vérifier l'état du dépôt avec git status



Conclusion

Ce travail dirigé nous a permis de découvrir et de pratiquer l'utilisation de **Git** et **GitHub** pour la gestion de projets informatiques. Nous avons appris à créer un dépôt local, à suivre les modifications grâce aux commits, et à envoyer nos fichiers vers un dépôt distant sur GitHub.

.La tâche est **réussie**, car tous les fichiers demandés ont été correctement ajoutés et envoyés sur GitHub. Les principales difficultés rencontrées concernaient la configuration de Git et l'utilisation de certaines commandes, mais elles ont été surmontées en suivant les étapes fournies.

Ce TD nous a donné une base solide pour utiliser Git et GitHub dans nos futurs projets informatiques.