Accéder en SSH à vos dépôts Git distants

I. Fonctionnement du protocole SSH

Secure Shell (**SSH**) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments **TCP** sont authentifiés et chiffrés.

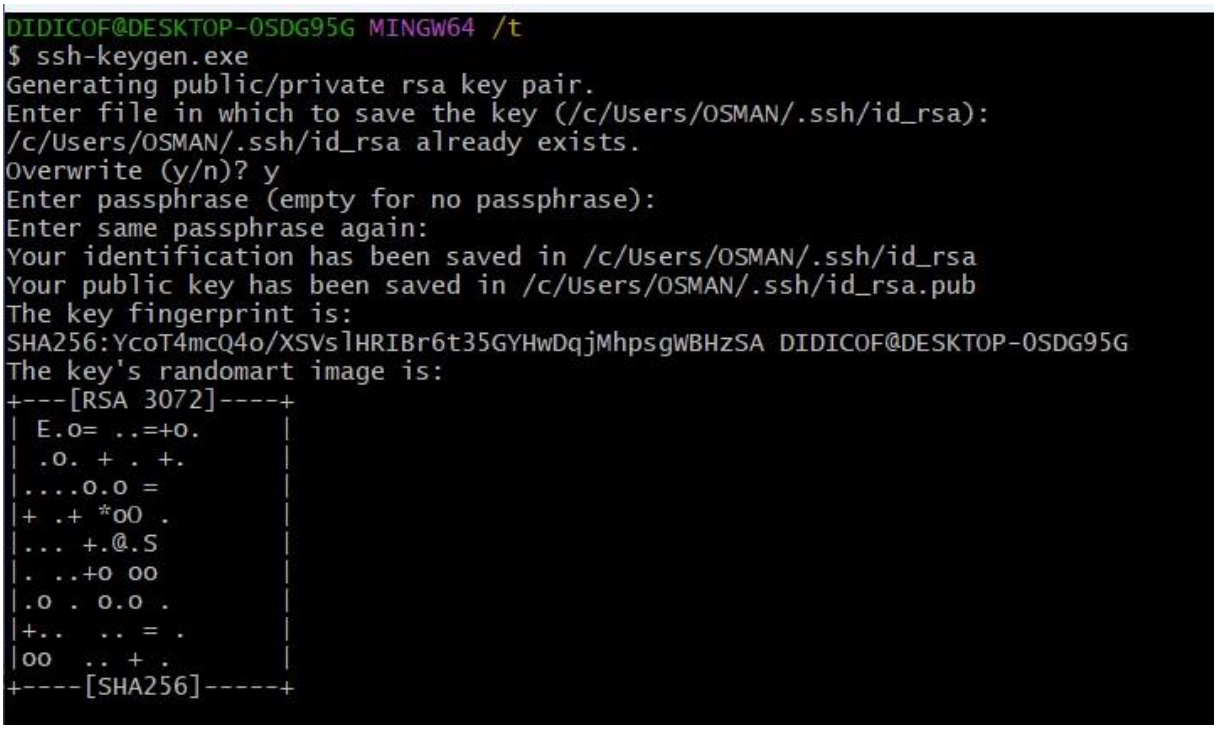
TCP/IP est un protocole de liaison de données utilisé sur Internet. Il est divisé en **4 couches distinctes**. Les quatre couches du modèle **TCP/IP** sont les suivantes :

1. **Couche d'application** : Cette couche est responsable de la communication entre les applications et les utilisateurs. Elle fournit des services de haut niveau tels que le courrier électronique, le transfert de fichiers et la navigation Web.
2. **Couche de transport** : Cette couche est responsable de la communication entre les programmes sur différents appareils. Elle fournit des services de transport de bout en bout tels que TCP (Transmission Control Protocol) et UDP (User Datagram Protocol).
3. **Couche Internet** : Cette couche est responsable de la communication entre les réseaux. Elle fournit des services de routage et de transmission de paquets tels que IP (Internet Protocol).
4. **Couche de liaison de données :** Cette couche est responsable de la communication entre les appareils sur le même réseau. Elle fournit des services de liaison de données tels qu'Ethernet.

TCP/IP est généralement intégré aux ordinateurs et est largement automatisé, mais il peut être utile d'en comprendre le modèle, en particulier lorsque vous configurez un ordinateur pour vous connecter à d'autres systèmes.

II. Génération des clés SSH

Il faut alors générer une paire de clé SSH via la commande : **ssh-keygen**



Info : Vous pouvez ne pas indiquez de mot de passe et ainsi avoir une communication SSH qui sera tout de même sécurisé. L'intérêt ici, c'est que vous n'aurez pas à écrire ce mot de passe lors d'un **git clone** ou **git pull/push**.

A ce moment-là, deux clés vont être générés dans votre dossier 'home' par défaut :

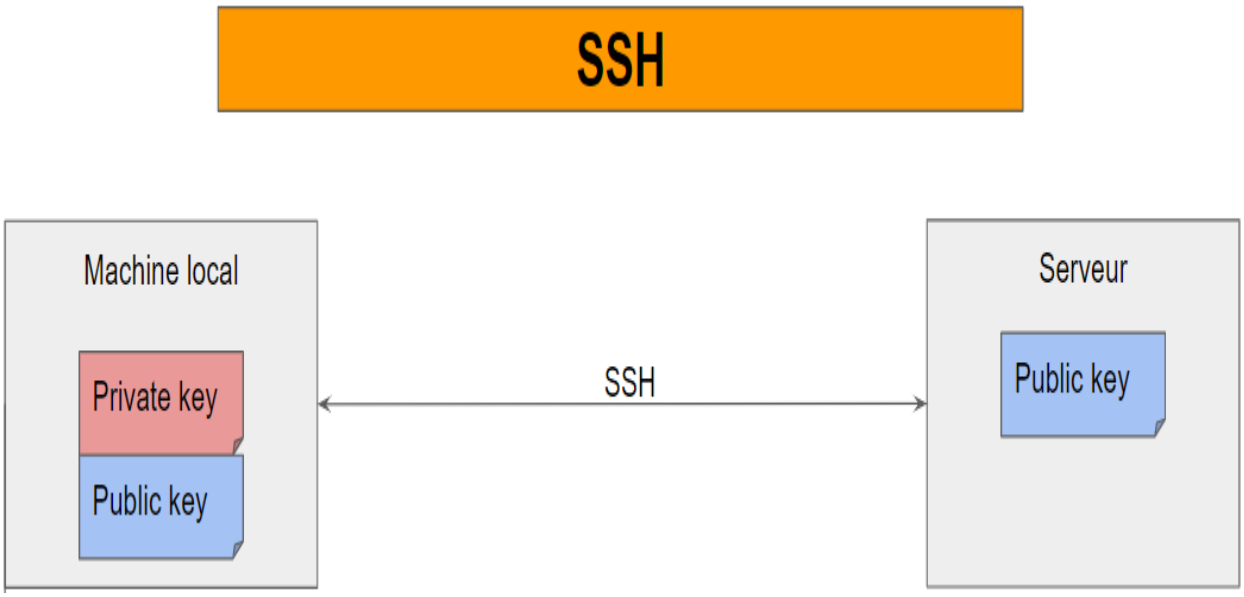
- **Linux** : **/home/[user]/.ssh**
- **Windows** : **C:\Utilisateurs\[user]\.ssh**

(Attention le dossier **.ssh** est un dossier **caché)**

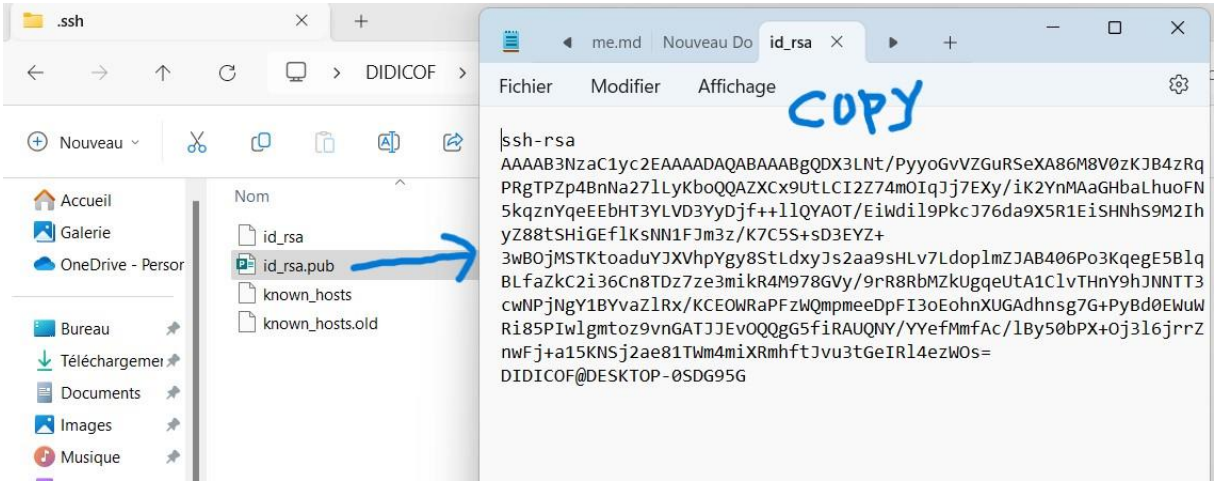
On retrouve alors dans le dossier **.ssh** les fichiers suivants :

- **id_rsa**: **clé privé** à conserver sur son PC et à ne surtout pas partager
- **id_rsa.pub** : **clé publique** à envoyer sur les machines avec lesquels vous voulez communiquer en **SSH**

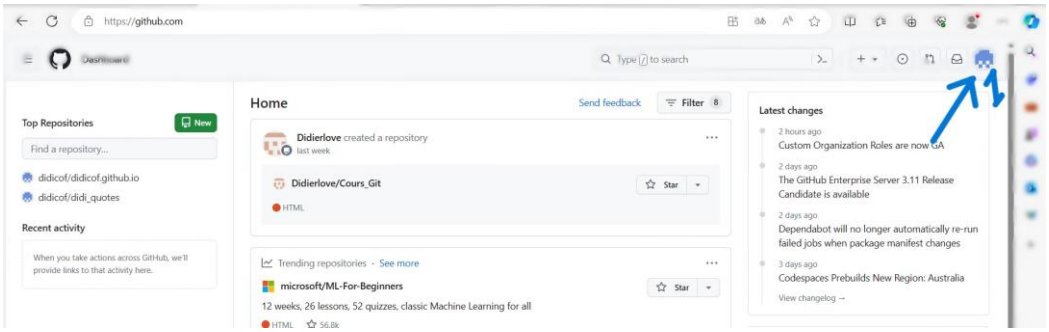
III. Mise en place du SSH sur GitHub

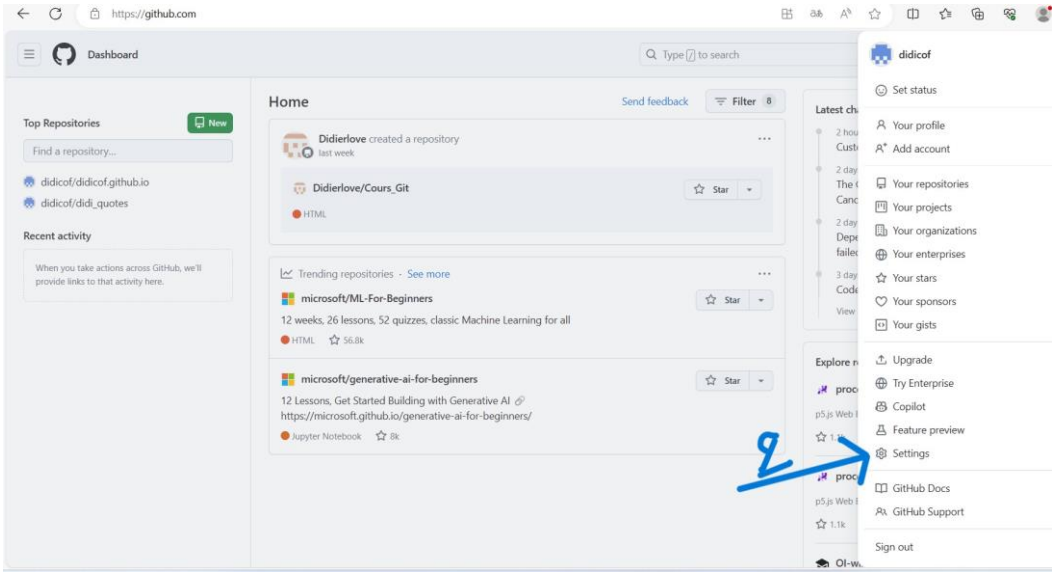


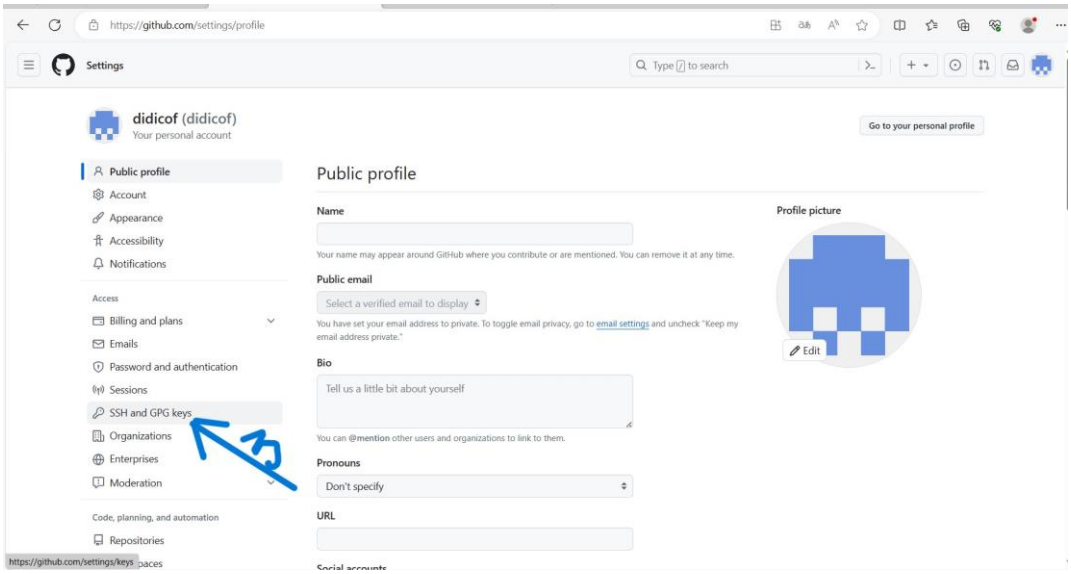
Pour commencer, il faut copier le contenu de **la clé SSH publique**. Pour cela, il faut se rendre dans le dossier où **ssh-keygen** a généré les clés. Puis, éditez le fichier **id_rsa.pub** et enfin copier l'intégralité de son contenu.



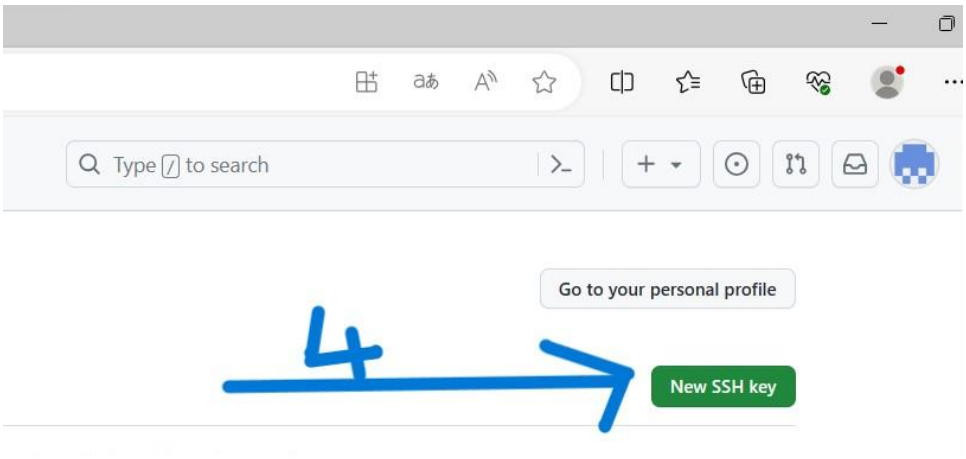
Vous pouvez alors vous rendre dans **les settings de votre compte** sur le site de **GitHub**.



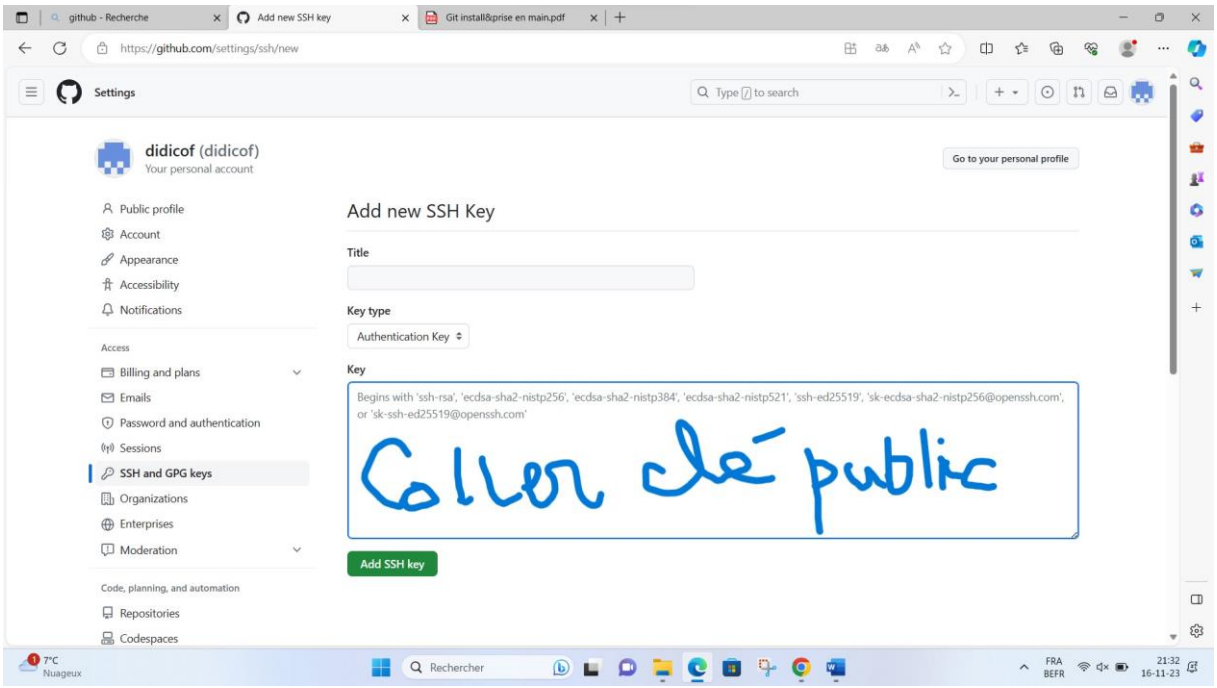


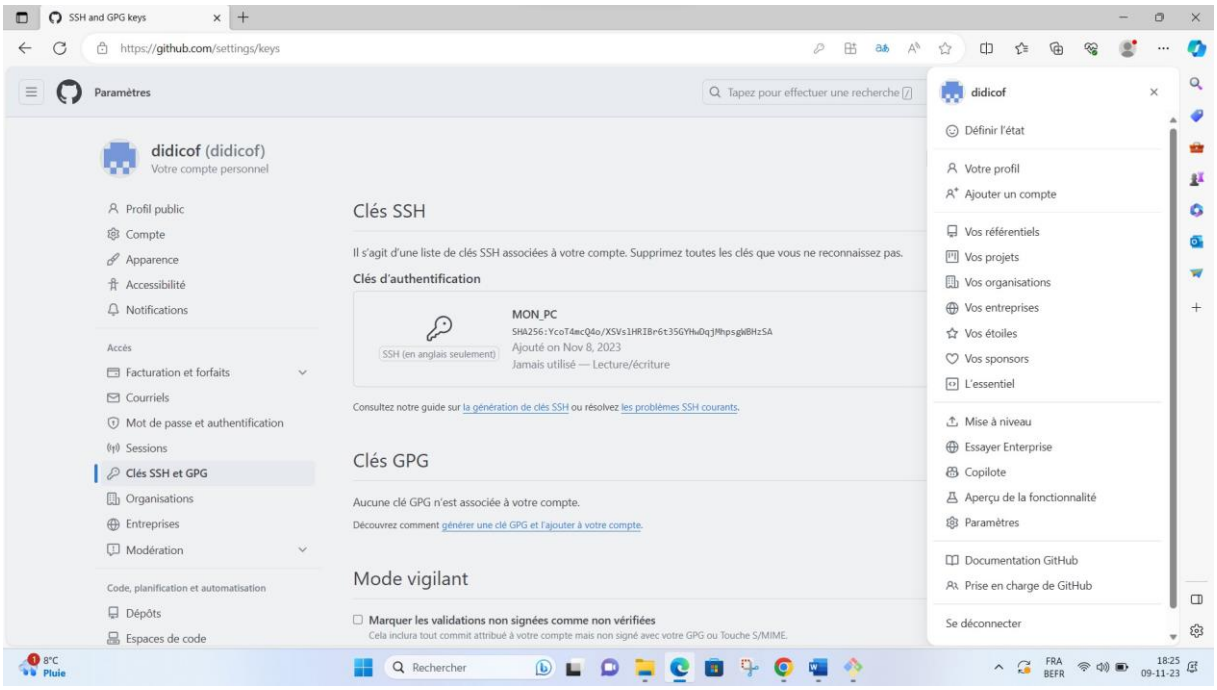


Rendez-vous dans l'onglet **SSH and GPG keys**, puis cliquer sur le bouton "**new SSH key"**.

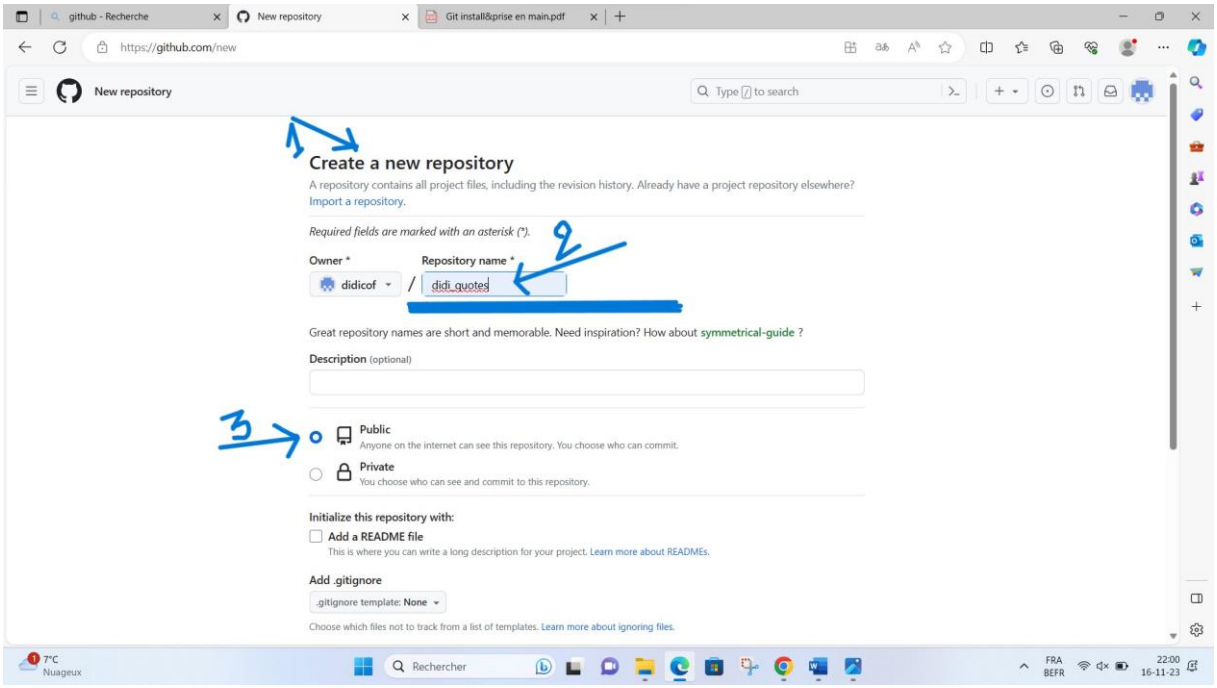


Il faut alors simplement coller le contenu du fichier **id_rsa.pub** que l'on a copier depuis notre PC, puis valider.





Il faut maintenant créer **un nouveau dépôt (repository)**, le donnez **un nom** et bien cochez **public** si vous voulez être vu par les autres. Cas contraire est **private.**

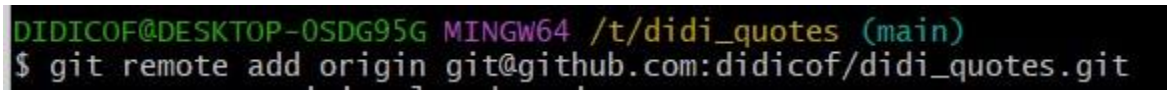


```
DIDICOF@DESKTOP-0SDG95G MINGW64 /t/didi_quotes (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello.html
        open_excel.bat
        quotes.txt
        style.css
        test.txt

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

DIDICOF@DESKTOP-0SDG95G MINGW64 /t/didi_quotes (main)
$ git add .
warning: in the working copy of 'hello.html', LF will be replaced by CRLF the ne
xt time Git touches it
warning: in the working copy of 'quotes.txt', LF will be replaced by CRLF the ne
xt time Git touches it
warning: in the working copy of 'style.css', LF will be replaced by CRLF the nex
t time Git touches it
warning: in the working copy of 'test.txt', LF will be replaced by CRLF the next
```

```
DIDICOF@DESKTOP-0SDG95G MINGW64 /t/didi_quotes (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   hello.html
        new file:   open_excel.bat
        new file:   quotes.txt
        new file:   style.css
        new file:   test.txt


DIDICOF@DESKTOP-0SDG95G MINGW64 /t/didi_quotes (main)
$ git commit -m "premier envoi sur internet"
[main (root-commit) d1ea6ec] premier envoi sur internet
 5 files changed, 109 insertions(+)
 create mode 100644 hello.html
 create mode 100644 open_excel.bat
 create mode 100644 quotes.txt
 create mode 100644 style.css
 create mode 100644 test.txt

DIDICOF@DESKTOP-0SDG95G MINGW64 /t/didi_quotes (main)
```

IV. Clone ou remote le dépôt en SSH

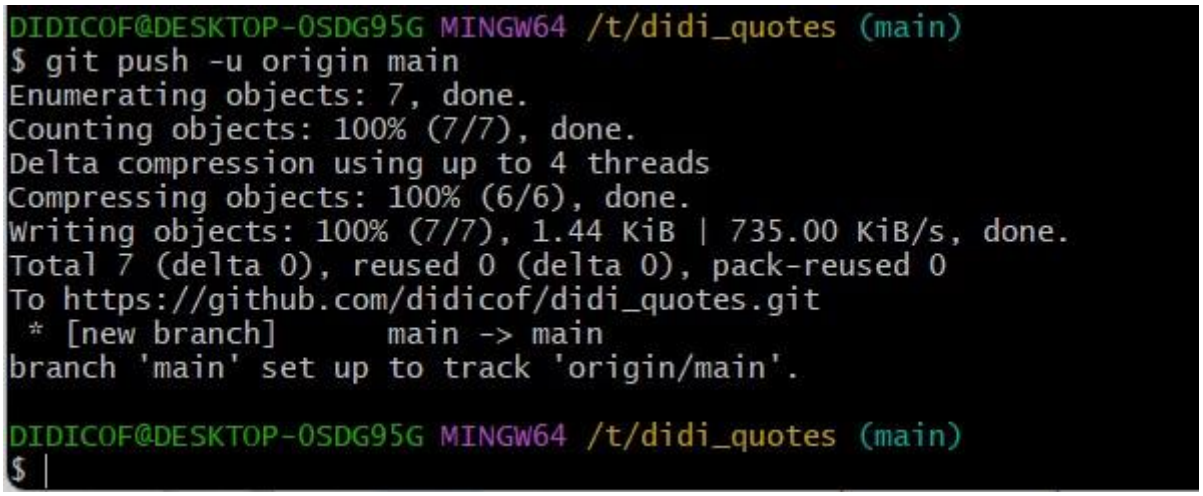
Maintenant que votre communication **SSH** est en place, vous pouvez taper, dans le terminal, la commande : **git remote add origin [url_ssh] ou git clone [url_ssh]**



Vous pouvez changer le protocole que vous utilisez sur un dépôt déjà cloné et ainsi passer de https à ssh et inversement.

V. Push le dépôt en SSH

Enfin, tapez la commande **git push -u origin [nom de la branche]**



VI. Un résumé :


GitHub Cheat Sheet
Versionner son travail
Versionner en local
git init
initialise le dépôt (se mettre sur le bon dossier), mieux à faire depuis Github.com
git add .
ajoute toutes les modifications (le . symbolise tout)
git commit -m "explication"
créer un nouveau commit. git add pousse les fichiers en zone d'index, git commit les sauvegarde réellement dans un nouveau commit
Gérer les commits
git log
liste des commits
git log -n2
affiche les 2 derniers commits
git show sha-1
voir commit spécifique (cliquer molette souris pour coller)
git checkout sha-1
remettre la version du sha-1
git checkout master
remettre version la plus récente
Versionner sur un dépôt distant
git clone lien-github.com
récupérer travail depuis dépôt distant
git push -u origin master
pousse les modifications vers serveur
git push -f origin master
pousse de force des modifications (à manipuler avec précaution)
Naviguer dans Git Bash
pwd
savoir dans quel dossier je suis
mkdir "dossier"
créer un dossier (Make Directory)
touch fichier.txt
créer fichier
ls
liste le dossier courant
ls -la
liste tout plus précisément que ls
cd dossier
aller dans le dossier (Change Directory)
cd ..
Remonter d'un dossier
Initialisation de Git
git config --global user.name
"Mon Nom"
git config --global user.email
mon@mail.com
git config --global --list
Affiche nom et mail
Autres commandes
git status
état du fichier
git diff
affiche les modifs avant commit
Commit son projet sur Github
git add .
git commit -m "message"
git push -u origin master