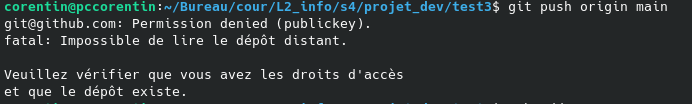
## Créer sa clée ssh et l’ajouter à github:

https://docs.github.com/fr/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account

## Blocage de permission :

Si vous avez rentré votre clé ssh dans github mais bloqué lors d’un push.



### Récupérer le chemin de votre clé privé sur votre pc

Quand vous avez créé votre clé ssh vous l’avez mis dans un dossier, il faut connaitre le nom de votre clé.

(Dossier par défaut)

(Fichiers présents)

Le .pud est la clé publique que vous avez entrée dans github .

L’autre est la clé privé.

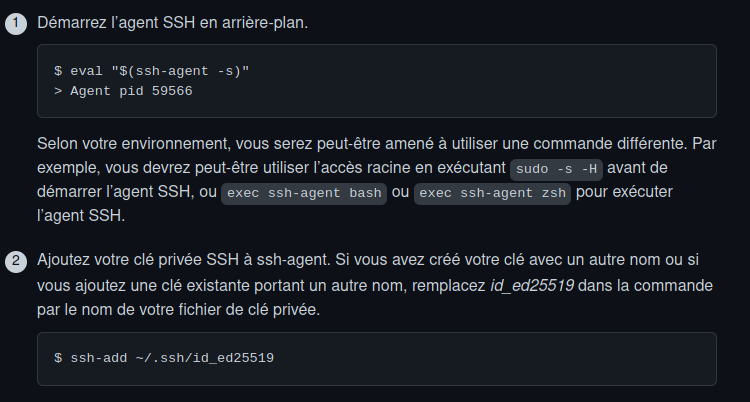
(Vos clés ont surement un nom blizzard genre id\_rsa ou e2423...

### Ajout de votre clé SSH à ssh-agent

A chaque ouverture d’un terminal (on automatisera plus tard)

Effectuer les commandes dans les carrées suivant :

* Utilisation 2 : ssh-add [chemin de la clé privé]
* -> vous devez entrée votre “passphrase” le mot de passe de vous avez mis lors de la création de votre clé.



MAGIQUE vous pouvez maintenant faire vos push.

## Les problèmes :

Si vous n’arriviez toujours pas à vous connecter.

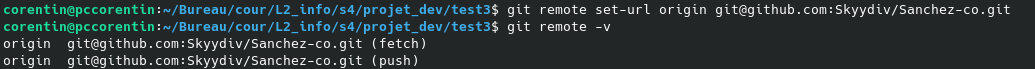
### HTTPS -> SSH:

Si on vous demande un username et un mot de passe.

Faire la commande et vérifier en face de origin ça soit bien git@github.... et non https....



Si c’est https: **git remote set-url origin git@github.com:Skyydiv/Sanchez-co.git**



## Crée son dépôt local :

Crée un dossier projet, et se déplacer dedans.

Transformer le dossier en dépôt git: git init

## Ajouter un fichier / dossier sur la branche main (local) :

-Crée un fichier :

ex : **echo “test” > test.txt**

-Ajoute le fichier au dépôt (zone temporaire ou on met tous les fichiers qu’on veut modifier) (le fichier est maintenant indexé) :

**git add test.txt**

-Vérifie que tout est vert avec un status:

**git status**

-Fait un commit (envoie tout ce qui est dans le dépôt sur la branche actuelle (local) ):

**git commit –m “message”**

-refaire un git status et observer le message, il faudra faire un push plus tard pour enregistrer nos motifs sur notre branche mais continué le tuto.

## Créer une branche :

Chaque branche permet d’avoir le même code à différents états du développement tout en permettant à tous les participant d’avancer différentes fonctionnalités simultanément.

-Regarder la liste des branches présentes

**git branch**

-> Vert avec étoile : la branche dans laquelle vous vous trouvez.

-> Blanc: autres branches locales

-Créer une nouvelle branche name:

git branch name

-Vérifier que la création à marcher avec **git branch**

-Se déplacer dans cette branche, tjr à faire avant de faire les add/commit pour sauvegarder les fichiers :

**git checkout Prenom**

(-modifier le fichier / add /commit)

## Configuré la connexion au github:

-Se connecter au dépôt distant avec ssh:

**git remote add origin git@github.com:Skyydiv/Sanchez-co.git**

-Faire son 1er push: (lie la branche local dans laquelle vous vous trouvez à la branche distante Name du github)

**git push –u origin Prenom**

## Le push :

-Plus généralement le push est pour mettre accessible / sauvegarder tous vos fichiers de votre branche (local) sur le github (branche distante)

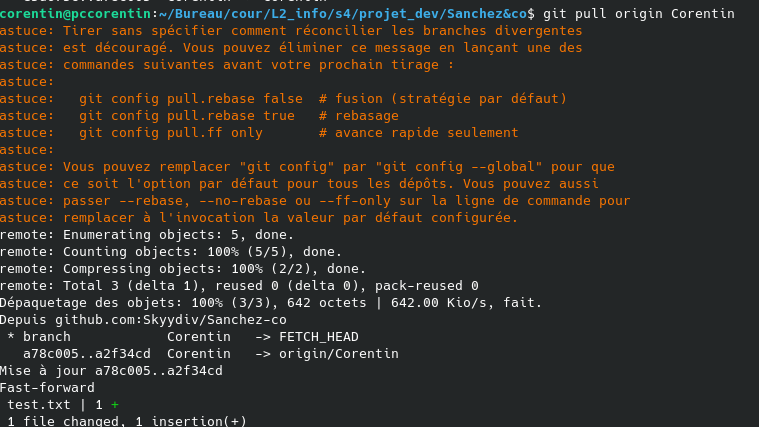
**git push origin Name**

## Le pull :

En va récupérer les dossiers d’une branche du github (distant)

**git pull origin main**

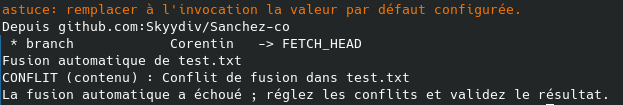
-> si tout se passe bien



-> Si fatal: **unrelated\_historie**  ou **historique sans relation** alors

**git pull origin Nom\_de\_branche\_a\_pull --allow-unrelated-histories**

-> Il y a des conflits, vous avez un fichier qui est sur votre dépôt local et le même sur le dépôt distant et ils ont tous les 2 été modifier



il faut le fichier en question, l’ouvrir garde le bout de code qui nous intéresse.

Puis faire un add fichier puis commit

## Repartir sur un espace propre:

* Allez où vous voulez crée votre projet
* [Ajout la clée ssh aux ssh\_agent](#_Ajout_de_votre)
* Git clone git@github.com:Skyydiv/Sanchez-co.git -> copie le dépôt distant
* Se déplacer dans le fichier
* Git branch dev -> créer une branche dev
* (si on je se séplacer dans dev -> **git checkout dev**)
* Faire ses modifications
* Add
* Commit
* Push dans main si la version du code modifié est montrable en démonstration, sinon dans dev.