Contenu

[a. Installation Windows 2](#_Toc496103228)

[b. Installation Linux 3](#_Toc496103229)

[c. S’identifier 3](#_Toc496103230)

[d. Mise en place du dépôt 4](#_Toc496103231)

[e. Commit 5](#_Toc496103232)

[f. Push 6](#_Toc496103233)

[g. Pull 7](#_Toc496103234)

[h. Gestion des branches 8](#_Toc496103235)

[i. Stash 10](#_Toc496103236)

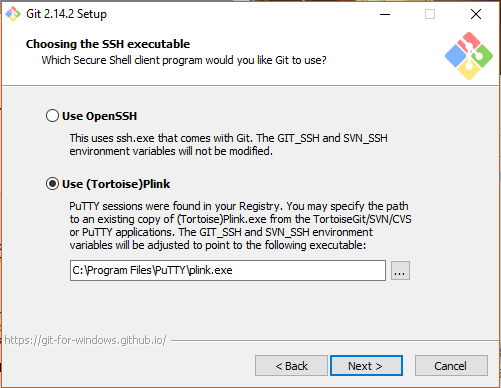
[Note : 10](#_Toc496103237)

# Installation Windows

Télécharger git *:* [*https://git-scm.com/downloads*](https://git-scm.com/downloads)

Suivez les étapes en laissant les choix par défaut, à une exception près :

si vous avez Putty installé, une page supplémentaire apparaitra, elle permet de choisir entre l’option utilisée de base (OpenSSH) et l’option Plink. Cette dernière est préférable.

Ce choix entraine une différence dans l’installation de tortoiseGit ! 

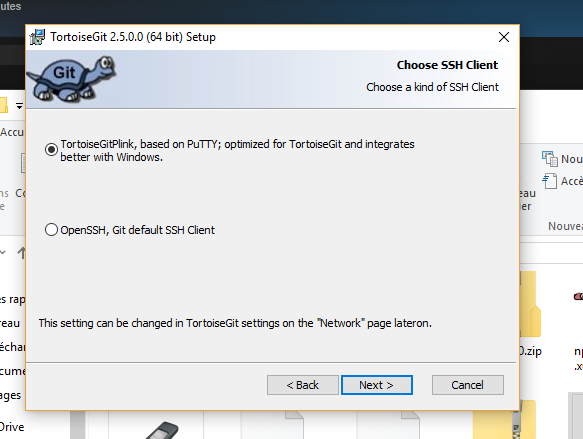
Télécharger tortoiseGit : [*https://tortoisegit.org/download/*](https://tortoisegit.org/download/)

*Attention : L’installation peut-être gêné par un anti-virus!!!*

De même pour git, suivez les étapes en laissant les choix par défaut.

Si vous n’avez pas changé l’option OpenSSH une fenêtre supplémentaire demande le client SSH à utiliser.

Prenez celui défini correspondant à votre précédente installation (la fenêtre n’apparaissant pas si l’option Plink a été choisie ce sera OpenSSH)



# Installation Linux

Dans un terminal :

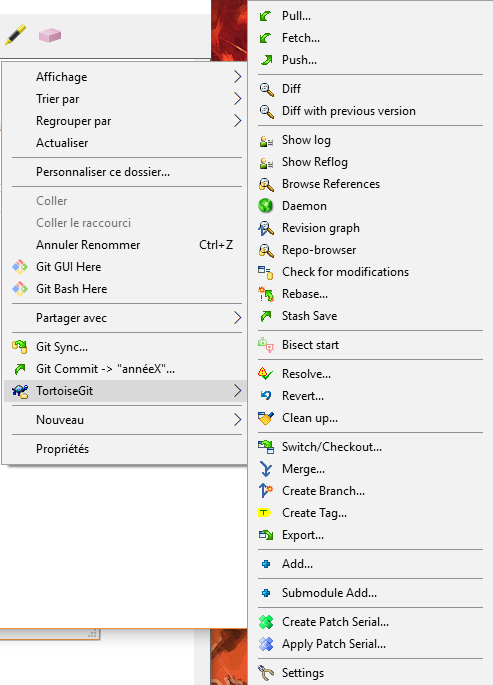
*Sudo apt-get install git*

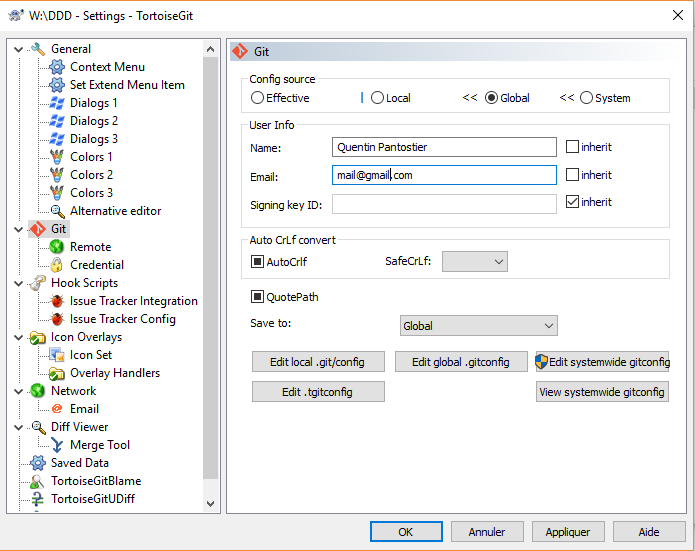
# S’identifier

**Windows :**

Il est important de s’identifier pour savoir qui a fait un commit.

*Clic droit dans le dépôt-> TortoiseGit->setings*

**



Dans la section Git ( ), vous pouvez définir vos informations (Name /Email : ) dans local/Global/System ( ) et vérifier celle active dans effective ( ).

**Linux** :

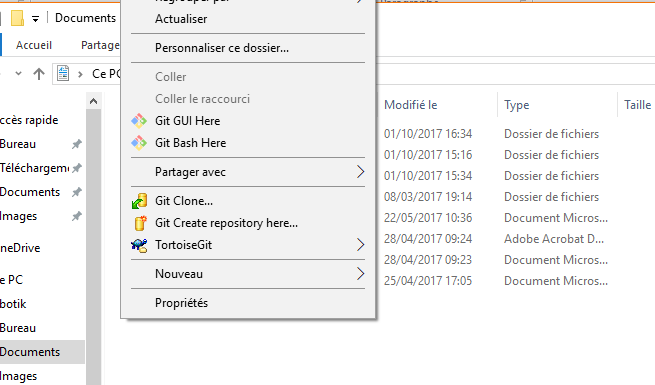
*Git config --global user.name « nom »*

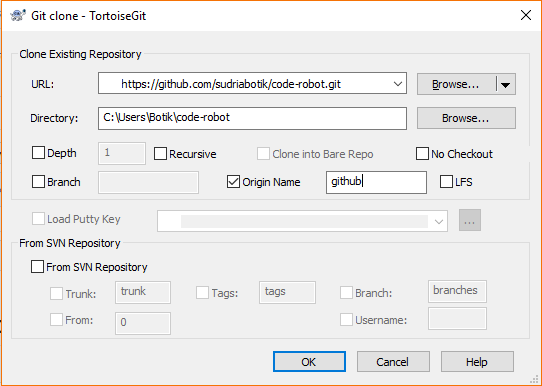
*Git config --global user.email  « email »*

# Mise en place du dépôt

**Windows :**

*Clic droit dans l’explorateur de fichier -> git clone*



Copier l’url du dépôt distant dans URL ( ) et le chemin du dépôt local dans Directory ( ) 

**Linux** :

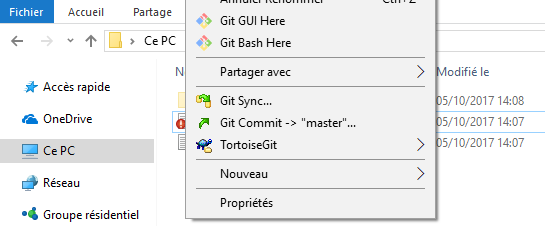
*Git clone URL*

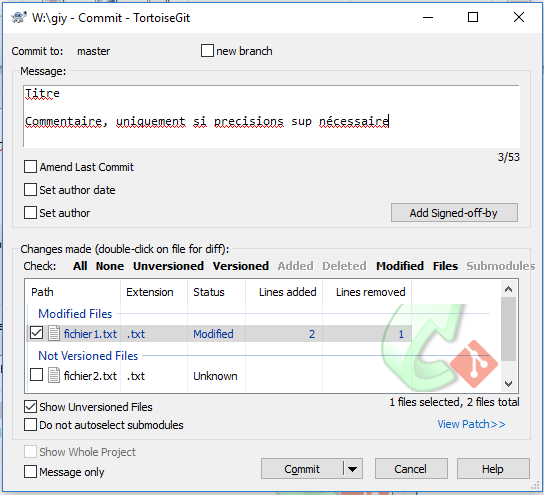
# Commit

**Windows :**

Action de base de git, permet d’ajouter les modifications dans la table de suivi de git.

*Clic droit dans le dépôt-> Git commit -> …*





* La branche modifiée par le commit
* Zone de texte pour le titre, indique de manière claire et concise les changements apportés.
* Une ligne d’espace, **OBLIGATOIRE** si un commentaire suit
* La zone de commentaire, pas obligatoire
* Zone d’affichage des modifications apportées aux fichiers déjà suivis par git
* Zone d’affichage des nouveaux fichiers

= Option Spéciale qui permet la correction du dernier commit. **Cette option ne doit être utilisée quand local**, si un commit a déjà étè push sur le serveur github il ne doit pas être modifié !

**Linux** :

Dans le dépôt :

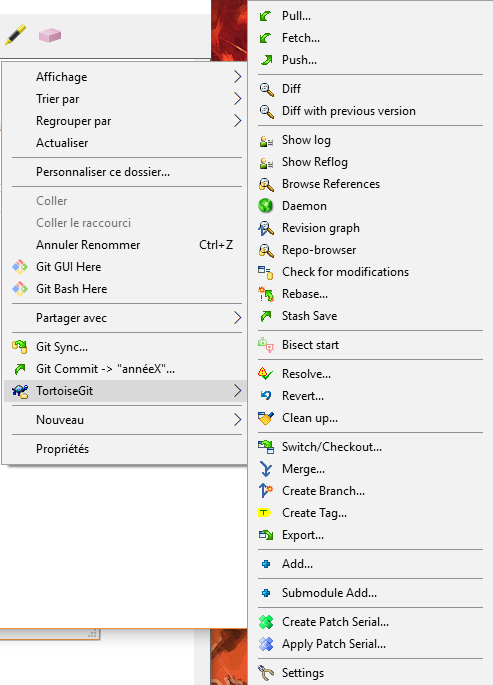
*git add \chemin\fichier-modif*

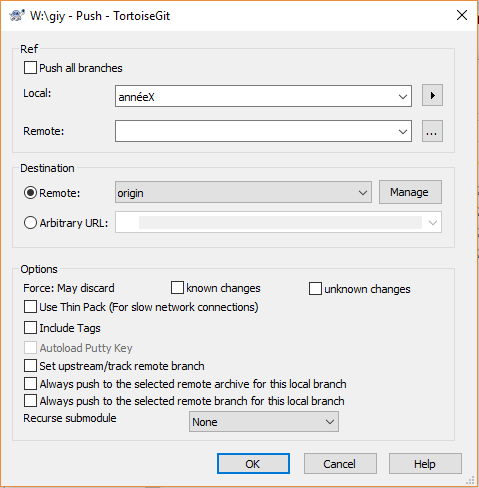
*git commit –m «titre commit »*

# Push

**Windows :**

*Clic droit dans le dépôt-> TortoiseGit->Push*

****



* Branche du commit à push
* Dépôts où se trouve la branche

**Linux** :

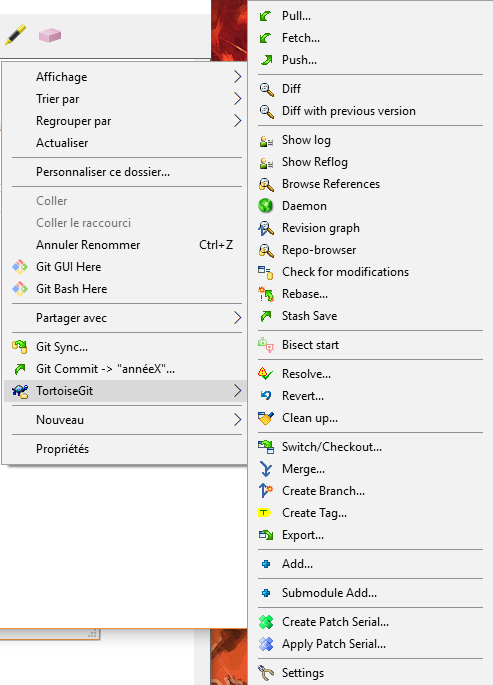
*git push origin nombranch*

# Pull

**Windows :**

Permet de mettre à jour le dépôt local par rapport au dépôt distant.

*Clic droit dans le dépôt-> TortoiseGit->Pull*

****

**Linux** :

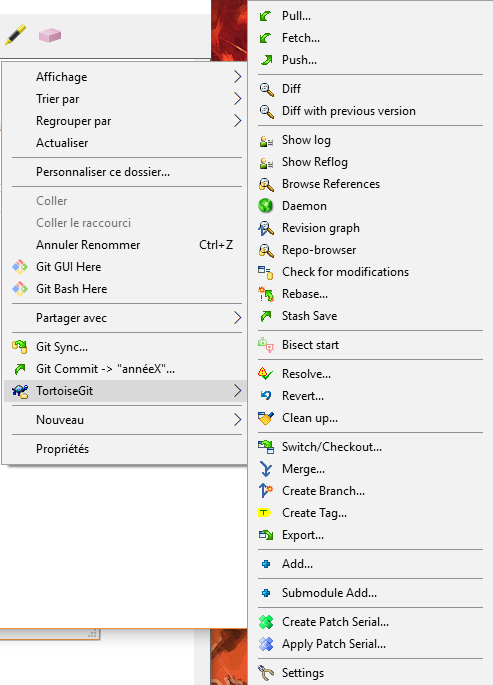
*git pull origin nombranch*

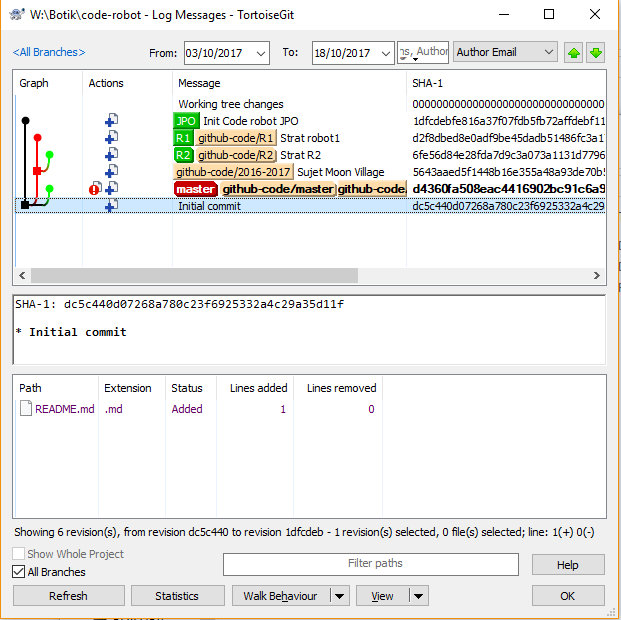
# Gestion des branches

**Windows :**

Faire apparaitre les logs du dépôt :

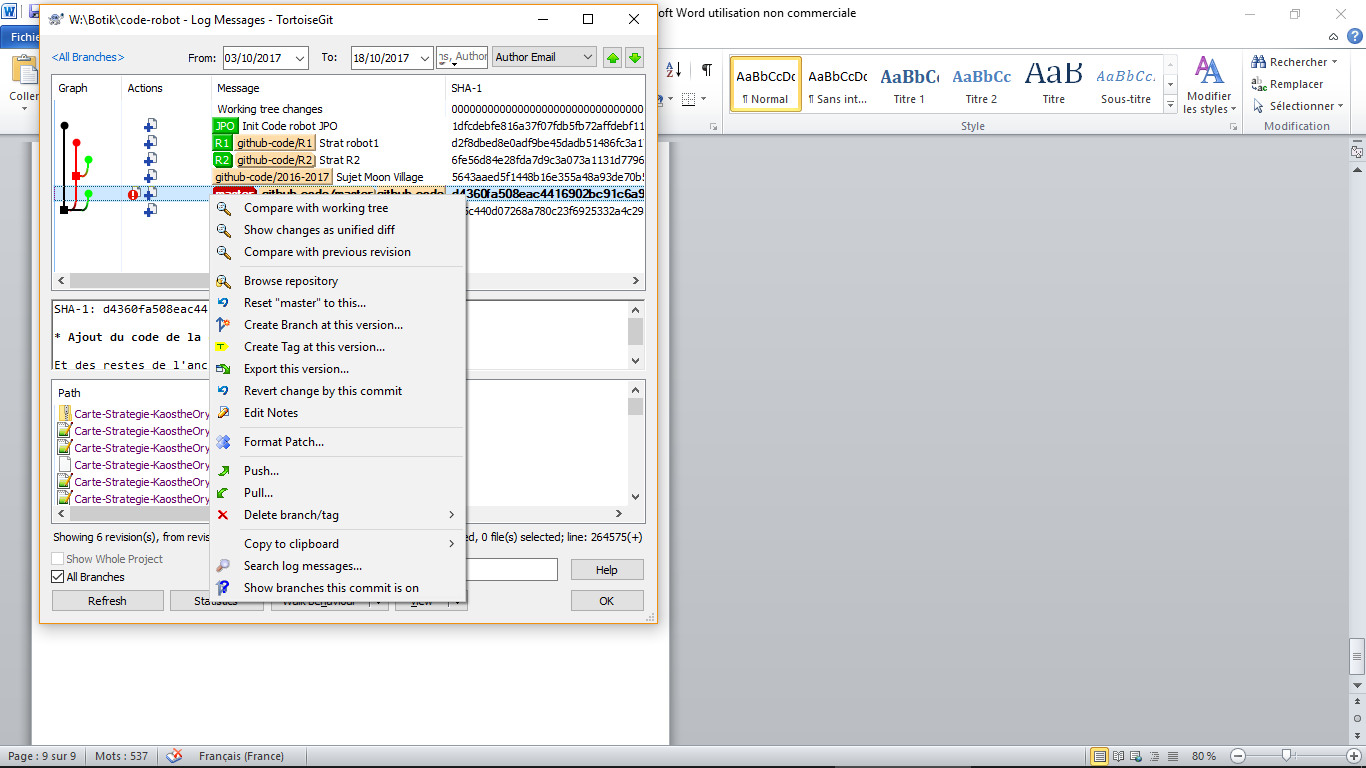
*Clic droit dans le dépôt-> TortoiseGit->Show log*

****



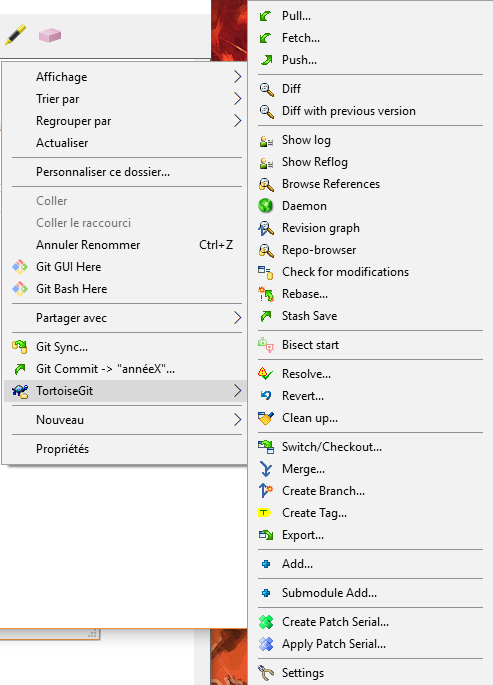
* Graph structure du dépots
* Zone de titre correspondant au commit
* Zone de Clés SHA-1, identifiant des commit
* Tête de branche (rouge branche actuel)
* Tête de branche Serveur
* Zone détaillant les fichiers modifé/ajouté par un commit

A partir de cette page il est possible d’un simple clique droit sur le commit voulu de :

- Revenir en arriére à partir du commit selectionner

- Créer une nouvelle branche

Ect…

****

Permet de changer de branches

**Linux :**

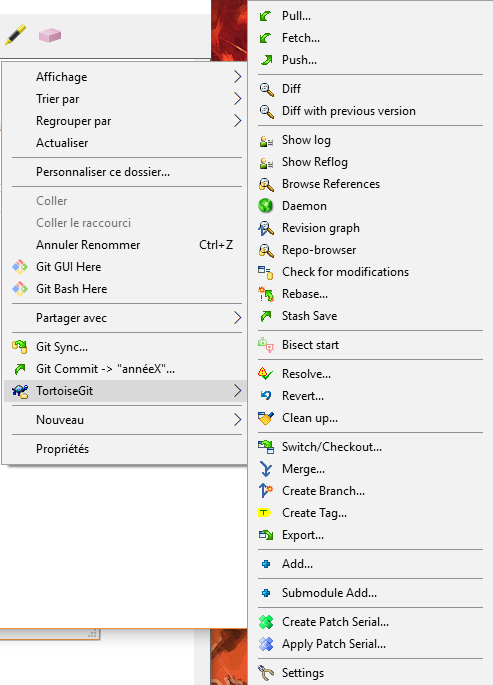
*git checkout -b nom\_branche*

*git push origin nom\_branche*

# Stash

**Windows :**

Permet d’enregistrer tous les changements dans un coin sans altérer le dépôt.

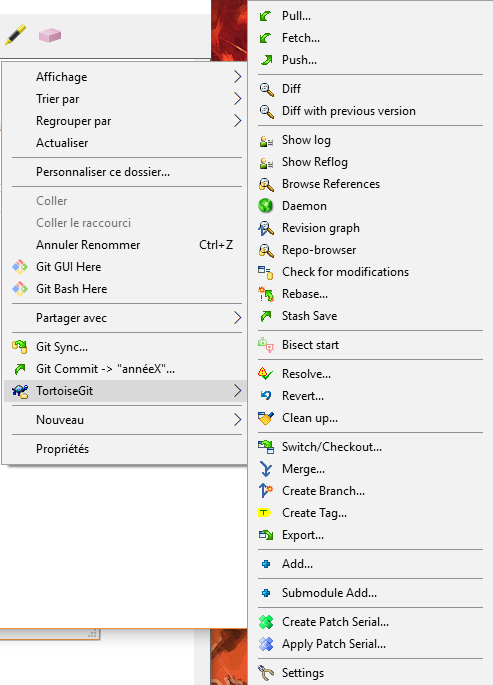
****

**Linux :**

git stash

# Note :

Les commandes Linux sont aussi utilisable sur Windows via l’invité de commande Git Bash

****

De plus les commandes linux permette de faire beaucoup plus de choses grâce options qui ne sont pas détaillé ici, voir la doc sur internet ou le man pour plus de précision.