

Installation et configuration de git sur Linux et Windows

[Retour à toute les documentations](#)

Règles

- "Saisie utilisateur"
- 'Élément cliquable/sélectionnable'
- Nom de fichier, dossier ou autre
- <Élément à remplacer>

lien, raccourci clavier et phrase de demande de saisie

commande, extrait code et extrait de fichier

Table des matières

- [Installation et configuration de git sur Linux et Windows](#)
 - [Règles](#)
 - [Table des matières](#)
 - [Installation](#)
 - [Installation de Git sur Ubuntu Desktop et Debian](#)
 - [Installation de Git sur Windows 10 et 11](#)
 - [Configuration](#)
 - [Configuration pour Linux et Windows](#)
 - [Liaison avec Github sur Linux et Windows](#)
 - [Utilisation](#)
 - [Utilisation de plusieurs comptes Github sur le même ordinateur - Ubuntu](#)
 - [Remplacer la branch master par une autre branch](#)
 - [Erreur](#)
 - [Erreur d'authentification HTTPS](#)

Installation

Installation de Git sur Ubuntu Desktop et Debian

- Installer le paquet `git` depuis le dépôt `apt` :

```
sudo apt install git
```

- Pour vérifier l'installation ouvrez un terminal et lancer la commande :

```
git --version
```

- Résultat attendu :

```
git version 2.34.1
```

Installation de Git sur Windows 10 et 11

- Télécharger le fichier d'installation sur le site officiel :

<https://git-scm.com/download/win>

- Ouvrir le fichier d'installation
- Suivre les instructions d'installation
- Sélectionner l'option '`Add a Git Bash Profile to Windows Terminal`'
- Ne pas sélectionner l'option '`Use Git from Git Bash only`' pour pouvoir utiliser git depuis le terminal windows
- Changer l'éditeur de texte par défaut si vous en avez envie
- Laisser toutes les autres options par défaut
- Cliquer sur '`Install`'
- Pour vérifier l'installation ouvrez un terminal et lancer la commande :

```
git --version
```

- Résultat attendu :

```
git version 2.41.0.windows.1
```

Configuration

Configuration pour Linux et Windows

- Si vous avez besoins d'aide pour la configuration de git, lancer la commande :

```
git help config
```

- Configurer l'adresse mail :

```
git config --global user.email <adresse mail>
```

- Configurer l'adresse mail :

```
git config --global user.name <votre prenom>
```

Liaison avec Github sur Linux et Windows

- Générer une clé SSH :

```
ssh-keygen -t rsa -b 4096 -C "votre adresse mail"
```

- laisser vide les trois champs suivant :

Enter file in which to save the key (/home/\$USER/.ssh/id_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

- Ajouter la clé public (présente dans le fichier `~/.ssh/id_rsa.pub`) à Github dans 'Settings' > 'SSH and GPG keys' > 'New SSH key'

Utilisation

Utilisation de plusieurs comptes Github sur le même ordinateur - Ubuntu

- Source

<https://gist.github.com/bonnopc/c78920431284ce3fc2a5270016205116>

La documentation ci-dessous utilise deux comptes Github, avec deux clés SSH différentes, mais vous pouvez l'adapter pour autant de comptes que vous voulez.

- Générer une nouvelle clé SSH comme indiqué dans la section [Liaison avec Github sur Linux et Windows](#)
- Effacer les clés mises en cache précédentes

```
ssh-add -D
```

- Si vous obtenez l'erreur ci-dessous :

```
Error connecting to agent: Connection refused
```

- Activer l'agent SSH

```
eval "$(ssh-agent)"
```

- Réessayer d'effacer les clés mises en cache

```
ssh-add -D
```

- Ajouter vos clés SSH à l'agent SSH

```
ssh-add ~/.ssh/id_rsa
ssh-add ~/.ssh/id_rsa_2
...
```

- Si vous le souhaitez, vous pouvez vérifier les clés ajoutées à l'agent SSH avec la commande

```
ssh-add -l
```

- Si vous n'en avez pas déjà un, créer un fichier de configuration SSH

```
touch ~/.ssh/config
```

- Ouvrir le fichier de configuration

```
open ~/.ssh/config
```

- Modifier le fichier de configuration pour ajouter les configurations suivantes

```
# Compte github 1
Host <NOM>.github.com
HostName github.com
IdentitiesOnly yes
IdentityFile <~/.ssh/id_rsa_1>

# Compte github 2
Host <NOM>.github.com
HostName github.com
IdentitiesOnly yes
IdentityFile <~/.ssh/id_rsa_rsa_2>
```

- Remplacer <NOM> par le nom de votre compte Github
 - Remplacer ~/.ssh/id_rsa_1 et ~/.ssh/id_rsa_2 par le chemin complet de vos clés SSH
- Cloner un dépôt Github

```
git clone git@<NOM>.github.com:exemple/exemple-repo.git
```

- Remplacer <NOM> par le nom du compte Github que vous voulez utiliser
- Modifier un dépôt Github existant

- Aller dans le dossier du dépôt
 - Modifier le fichier `.git/config`

```
open .git/config
```

- Modifier l'URL du dépôt

```
[remote "origin"]
url = git@<NOM>.github.com:exemple/exemple-repo.git
```

- Vérifier que tout fonctionne correctement en poussant un changement

```
git add .  
git commit -m "Message de commit"  
git push
```

Remplacer la branch master par une autre branch

- Source

<https://stackoverflow.com/questions/2862590/how-to-replace-master-branch-in-git-entirely-from-another-branch>

- Vérifier que la branch master est bien à jour

```
git checkout master  
git pull
```

- Passer sur la branch que vous voulez mettre à la place de la branch master que nous appelons `<new_branch>` et vérifier qu'elle est bien à jour

```
git checkout <new_branch>  
git pull
```

- Fusionner la branch `<new_branch>` avec la branch master en écrasant la branch master

```
git merge -s ours master
```

- Passer sur la branch master

```
git checkout master
```

- Fusionner la branch `<new_branch>` avec la branch master

```
git merge <new_branch>
```

- Mettre à jour la branch master sur Github

```
git push origin master
```

Erreur

Erreur d'authentification HTTPS

- Ne pas passer par HTTPS, il faut utiliser le ssh pour communiquer avec Github. Si votre clé ssh est bien configuré comme indiqué au dessus vous n'aurez normalement pas de problème.
- Sinon vous pouvez aussi mettre un token à la place du mot de passe mais il faudra le remettre à chaque action que vous effectuer sur Github
 - Générer un token d'authentification sur Github dans 'Settings' > 'Developer settings' > 'Personal access tokens' > 'Fine-grained personal access tokens' > 'Generate new token'
 - Sélectionner quel repo vous voulez accéder avec votre token, dans mon cas j'ai sélectionné 'All repositories'

- Donner les autorisations que vous voulez, dans mon cas j'ai sélectionné la plus haute autorisation de chaque élément de la catégorie **Repository permissions** et j'ai rien touché à l'autre catégorie.
- Cliquer sur '**Generate token**'

[Retour à toutes les documentations](#)