

Installation et utilisation de Git et GitHub sur Ubuntu Linux : Un guide du débutant

GitHub est un trésor de certains des meilleurs projets au monde, construit grâce aux contributions de développeurs du monde entier. Cette plateforme simple mais extrêmement puissante aide toute personne intéressée à créer ou à développer quelque chose de grand à contribuer et à être reconnu dans la communauté open source.

Ce didacticiel est un guide d'installation rapide pour installer Git et utiliser GitHub et comment exécuter ses différentes fonctions de créer un référentiel localement, en connectant ce référentiel à l'hôte distant qui contient votre projet (où tout le monde peut voir), valider les modifications et enfin transmettre tout le contenu du système local vers GitHub.

Veuillez noter que ce tutoriel suppose que vous ayez une connaissance de base des termes utilisés dans Git tels que push, pull, requêtes, commit, référentiel, etc. Cela vous oblige également à vous inscrire sur GitHub [ici](#) et [notez](#) votre GitHub nom d'utilisateur. Alors commençons :

1 Installer Git pour Linux

Téléchargez et installez Git pour Linux :

```
sudo apt install git
```

La commande ci-dessus est destinée à Ubuntu et fonctionne sur toutes les versions récentes d'Ubuntu, testées d'Ubuntu 16.04 à Ubuntu 22.04, et il est probable que cela fonctionnera de la même manière sur les versions futures.

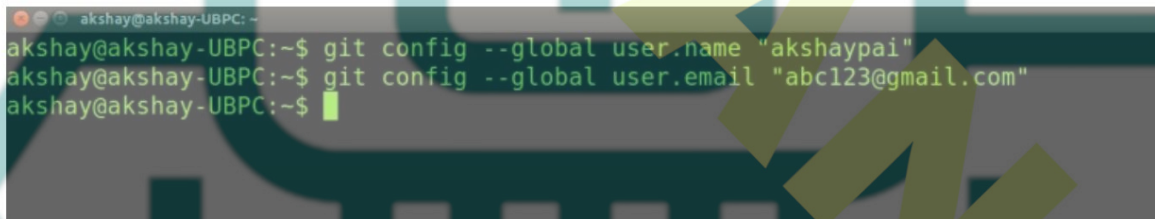
2 Configuration de GitHub

Une fois l'installation terminée avec succès, la prochaine chose à faire est de configurer les détails de configuration de GitHub. utilisateur. Pour ce faire, utilisez les deux commandes suivantes en remplaçant "user_name" par votre nom d'utilisateur GitHub et en remplaçant "email_id" avec votre identifiant de messagerie que vous avez utilisé pour créer votre compte GitHub.

```
git config --global user.name "user_name"
```

```
git config --global user.email "email_id"
```

L'image suivante montre un exemple de ma configuration avec mon "user_name" étant "akshaypai" et mon "email_id" étant "abc123@gmail.com"



```
akshay@akshay-UBPC:~$ git config --global user.name "akshaypai"
akshay@akshay-UBPC:~$ git config --global user.email "abc123@gmail.com"
akshay@akshay-UBPC:~$
```

3 Création d'un référentiel local

Créez un dossier sur votre système. Cela servira de référentiel local qui sera ensuite transféré sur le site Web GitHub. Utilisez la commande suivante :

```
git init Mytest
```

Si le référentiel est créé avec succès, vous obtiendrez la ligne suivante :

Dépôt Git vide initialisé dans /home/akshay/Mytest/.git/

Cette ligne peut varier en fonction de votre système.

Donc ici, Mytest est le dossier créé et "init" fait du dossier un référentiel GitHub. Changez le répertoire en ce dossier nouvellement créé :

```
cd Mytest
```

4 Créez un fichier README pour décrire le référentiel

Créez maintenant un fichier README et ntrez du texte comme " ceci est une configuration git sous Linux ". Le fichier README est généralement utilisé pour décrivez ce que contient le référentiel ou en quoi consiste le projet. Exemple:

```
gedit README
```

Vous pouvez utiliser n'importe quel autre éditeur de texte. J'utilise gedit. Le contenu du fichier README sera :

```
This is a git repo
```

5 Ajout de fichiers de référentiel à un index

Ceci est une étape importante. Ici, nous ajoutons tout ce qui doit être placé sur le site Web dans un index. Ces choses il peut s'agir des fichiers texte ou des programmes que vous pourriez ajouter pour la première fois dans le référentiel ou il peut s'agir de l'ajout d'un fichier qui existe déjà mais avec quelques modifications (une version plus récente/version mise à jour).

Ici, nous avons déjà le fichier README. Alors, créons un autre fichier contenant un simple programme C et appelons-le échantillon.c. Son contenu sera :

```
#include<stdio.h>
int main()
{
printf("hello world");
return 0;
}
```

Donc maintenant que nous avons 2 fichiers

README and sample.c

ajoutez-le à l'index en utilisant les 2 commandes suivantes :

```
git add README
```

```
git add sample.c
```

Notez que la commande "git add" peut être utilisée pour ajouter n'importe quel nombre de fichiers et de dossiers à l'index. Ici, quand je dis index, ce à quoi je fais référence est un espace de type tampon qui stocke les fichiers/dossiers qui doivent être ajoutés dans Git. dépôt.

6 Validation des modifications apportées à l'index

Une fois tous les fichiers ajoutés, nous pouvons le valider. Cela signifie que nous avons finalisé les ajouts et/ou modifications à apporter. être réalisés et ils sont maintenant prêts à être téléchargés dans notre référentiel. Utilisez la commande :

```
git commit -m "some_message"
```

"some_message" dans la commande ci-dessus peut être n'importe quel message simple comme "mon premier commit" ou "modifier dans le fichier readme", etc.

7 Créer un référentiel sur GitHub

Créez un référentiel sur GitHub. Notez que le nom du référentiel doit être le même que celui du référentiel sur le système local. Dans ce cas, ce sera « Mytest ». Pour ce faire, connectez-vous à votre compte sur <https://github.com>. Cliquez ensuite sur le Symbole "plus(+)" dans le coin supérieur droit de la page et sélectionnez "créer un nouveau référentiel". Remplissez les détails comme indiqué dans le image ci-dessous et cliquez sur le bouton "Créer un référentiel".

Owner akshaypai / **Repository name** Mytest ✓

Great repository names are short and memorable. Need inspiration? How about **petulant-ironman**.

Description (optional)

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

Une fois celui-ci créé, nous pouvons transférer le contenu du référentiel local sur le référentiel GitHub de votre profil. Connectez-vous au référentiel sur GitHub à l'aide de la commande :

Remarque importante : assurez-vous de remplacer « nom_utilisateur » et « Mytest » dans le chemin par votre nom d'utilisateur et votre dossier Github avant d'exécuter la commande !

```
git remote add origin https://github.com/user_name/Mytest.git
```

8 Transférer des fichiers d'un référentiel local vers un référentiel GitHub

La dernière étape consiste à transférer le contenu du référentiel local vers le référentiel hôte distant (GitHub), en utilisant la commande :

```
git push origin master
```

Entrez les informations de connexion [nom d'utilisateur et mot de passe].

L'image suivante montre la procédure de l'étape 5 à l'étape 8

```
akshay@akshay-UBPC: ~/Mytest
akshay@akshay-UBPC:~/Mytest$ git add README
akshay@akshay-UBPC:~/Mytest$ git add sample.c
akshay@akshay-UBPC:~/Mytest$ git commit -m "first edit"
[master (root-commit) 62c032f] first edit
 2 files changed, 7 insertions(+)
 create mode 100644 README
 create mode 100644 sample.c
akshay@akshay-UBPC:~/Mytest$ git remote add origin https://github.com/akshaypai/Mytest.git
akshay@akshay-UBPC:~/Mytest$ git push origin master
Username for 'https://github.com': akshaypai
Password for 'https://akshaypai@github.com':
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 337 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/akshaypai/Mytest.git
 * [new branch]      master -> master
akshay@akshay-UBPC:~/Mytest$
```

Cela ajoute donc tout le contenu du dossier « Mytest » (mon référentiel local) à GitHub. Pour les projets ultérieurs ou pour la création de référentiels, vous pouvez commencer directement par l'étape 3. Enfin, si vous vous connectez à votre compte GitHub et cliquez sur votre dépôt Mytest, vous pouvez constater que les 2 fichiers README et sample.c ont été téléchargés et sont visibles par tous comme

montré dans l'image suivante.

akshaypai / Mytest

Unwatch 1

Description Website

Short description of this repository Website for this repository (optional) Save or Cancel

1 commit 1 branch 0 releases 1 contributor

branch: master Mytest / +

first edit

akshaypai authored 4 minutes ago latest commit 62c032f076

README first edit 4 minutes ago

sample.c first edit 4 minutes ago

README

this is a sample file

Questions fréquemment posées

Comment puis-je vérifier si Git est correctement installé ?

Après l'installation, vous pouvez vérifier que Git est installé en tapant :

```
git --version
```

dans la borne. Cela affichera la version installée de Git.

Comment configurer Git avec mon nom d'utilisateur et mon email ?

Configurez votre nom d'utilisateur et votre adresse e-mail Git à l'aide des commandes :

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

Remplacez « Votre nom » et « votre.email@example.com » par vos informations personnelles.

Quelles sont les commandes Git de base que je devrais connaître ?

Les commandes GIT les plus fréquemment utilisées sont :

- git init : initialise un nouveau référentiel Git.
- git clone [URL] : clone un référentiel à partir d'une source distante.
- git add [file] : ajoute des fichiers à la zone de transit.
- git commit -m "[commit message]": valide vos modifications avec un message.
- git push : pousse vos modifications vers le référentiel distant.
- git pull : extrait les mises à jour du référentiel distant.

Comment créer un nouveau référentiel sur GitHub ?

Accédez à GitHub, connectez-vous et cliquez sur l'icône « + » dans le coin supérieur droit, puis sélectionnez « détails Nouveau référentiel ». Remplissez le du référentiel et cliquez sur »Créer un référentiel «.

Comment puis-je transférer mon référentiel local vers GitHub ?

Tout d'abord, ajoutez le référentiel distant en utilisant :

```
git remote add origin [repository URL]
```

Ensuite, poussez votre code avec :

```
git push -u origin master
```

(pour la première pression) ou

```
git push
```

(pour les poussées ultérieures).

Que dois-je faire si j'obtiens une erreur « autorisation refusée » lors de la transmission vers GitHub ?

Cette erreur se produit généralement lorsque les clés SSH ne sont pas correctement configurées. Générez une clé SSH avec :

```
ssh-keygen
```

et ajoutez-le à votre compte GitHub sous " Paramètres " -> " Clés SSH et GPG " .

Comment extraire les modifications d'un référentiel GitHub ?

Utiliser git HUB pour récupérer et fusionner les modifications du référentiel distant vers votre référentiel local.

Qu'est-ce qu'un fichier .gitignore et comment l'utiliser ?

Un fichier .gitignore spécifie les fichiers intentionnellement non suivis que Git doit ignorer. Ajoutez des noms de fichiers ou des modèles à ce fichier pour les exclure du suivi.

Comment puis-je résoudre les conflits de fusion dans Git ?

Lorsqu'un conflit de fusion se produit, modifiez manuellement les fichiers pour résoudre les conflits, puis utilisez :

```
git add [file]
```

pour les marquer comme résolus et valider les modifications.

Comment organiser le développement avec plusieurs développeurs

La fonctionnalité de branche dans GIT peut être utilisée pour organiser les validations et le développement lorsque plusieurs développeurs travaillent sur le même projet. Consultez ce guide pour savoir comment créer [des branches dans GIT](#).