Git

Les différentes commandes GIT

bstorm TRAINING

Table des matières

Qu'est-ce que GIT ?

Git est un système de contrôle de version distribué utilisé dans le développement de logiciels.

Il permet aux développeurs de suivre et de gérer les modifications apportées à leur code source au fil du temps.

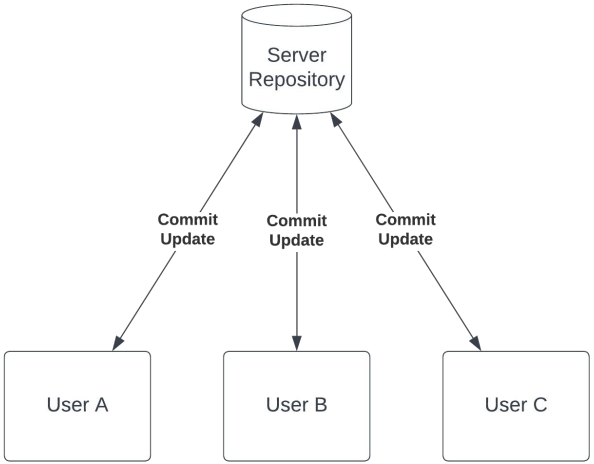
Git est compatible avec de nombreuses plateformes d'hébergement de code, telles que Github, GitLab, BitBucket, ... ce qui facilite la collaboration et le partage du code avec d'autres développeurs.

https://git-scm.com/

Qu'est-ce que GIT ?

Centralisé

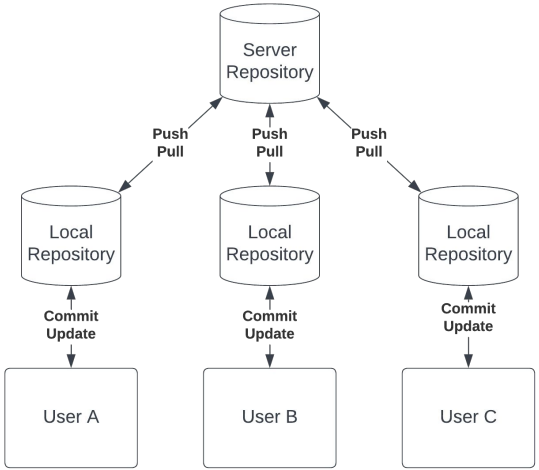
Tous les utilisateurs utilisent le même dépôt distant.



Risqué

Distribué (décentralisé)

Chaque utilisateur possède son propre dépôt local. Il valide ses changements et les synchronise avec le dépôt distant.



Moins risqué

Configuration de GIT

Configuration des informations de l'utilisateur

Configuration de GIT

Il existe 3 niveaux de configuration :

- **--system**

Appliquer à tous les utilisateurs sur l'ordinateur.

- **--global**

Appliquer à un compte utilisateur courant.

- **--local**

Appliquer à un dépôt local.

Configuration de GIT

- **git --version**

Permet d'afficher la version de GIT installée.

- **git config --global --list**

Permet d'afficher la liste des paramètres globaux.

- **git config --global --edit**

Permet d'éditer dans un éditeur de texte la configuration pour l'utilisateur courant.

Configuration de GIT

Configuration des informations de l'utilisateur pour tous les dépôts locaux.

- **git config --global user.name [name]**

Configure le nom de l'utilisateur qui sera utilisé pour tous les commits.

- **git config --global user.email [email]**

Configure l'adresse e-mail de l'utilisateur qui sera utilisé pour tous les commits.

- **git config --global color.ui auto**

Configure la colorisation automatique des lignes de commandes.

Initialisation d’un projet

Initialisation et clônage de répertoires

Initialisation d’un projet

Initialisation d’un dépôt local existant depuis un dépôt vierge ou en ligne.

- **git init [project-name]**

Si le project-name est fourni, crée un dossier avec ce nom et initialise un dépôt local.

Sinon initialise un dépôt local dans le dossier courant.

- **git clone [project-url]**

Récupération d’un répertoire avec l’historique complet depuis un répertoire en ligne.

Stage & Snapshot

Stage & Snapshot

- **git status**

Permet d'afficher les différents changements de statut des fichiers contenus dans le dépôt.

- **git add [file]**

Permet d'ajouter les fichiers à l'index de suivi des modifications.

- **git reset [file]**

Permet d'enlever le fichier de l'index, mais conserve le contenu.

Stage & Snapshot

- **git diff**

Permet d'afficher les modifications de fichier qui ne sont pas encore indexées.

- **git diff --staged**

Permet d'afficher les différences de modifications entre la version du fichier indexée et la dernière version.

- **git commit -m "message descriptif"**

Permet de créer un nouveau commit en enregistrant les modifications de l'index, avec un message de validation décrivant les changements effectués.

L'option **-a** permet d'ajouter directement les fichiers au commit créé.

Stage & Snapshot

- **git restore [file-name]**

Permet de restaurer l'état d'origine du fichier.

Branches & Fusion

Branches & Fusion

- git branch
- git branch [branch-name]
- git checkout

Branches & Fusion

- git merge [branch]
- git log

Inspecter & Comparer

Examiner les informations de changements

Inspector & Comparer

- git log
- git log branchB..branchA
- git log --follow [file]

Inspecter & Comparer

- git diff branchB...branchA
- git show [SHA]

Traquer les changements de chemin

Traquer les changements de chemin

- git rm [file]
- git mv [existing-path] [new-path]
- git log --stat -M

Partage & Mise à jour

Récupération des mises à jour depuis d'autres répertoires

Partage & Mise à jour

- git remote add [alias] [url]
- git fetch [alias]
- git merge [alias]/branch

Partage & Mise à jour

- git push [alias] [branch]
- git pull

Réécriture de l’historique

Récupération des mises à jour depuis d’autres répertoires

Réécriture de l'historique

- git rebase [branch]
- git reset --hard [commit]

Commits temporaires

Enregistrer les modifications temporaires

Commits temporaires

- git stash
- git stash list

Commits temporaires

- git stash pop
- git stash drop

Merci pour votre attention.

bstorm
TRAINING