Commandes de base de Git

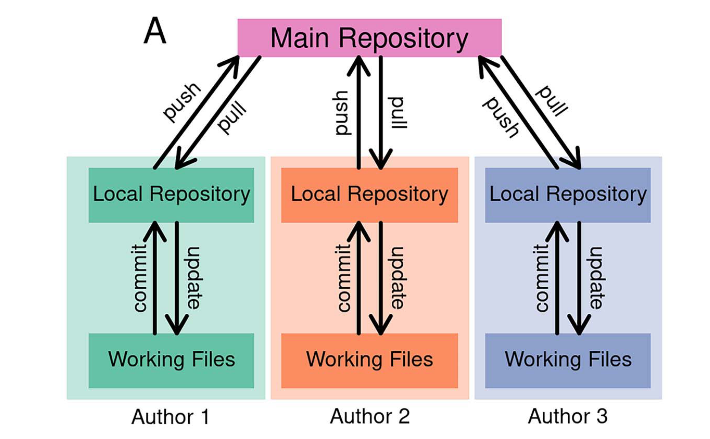
Damien Belvèze

18-11-2021

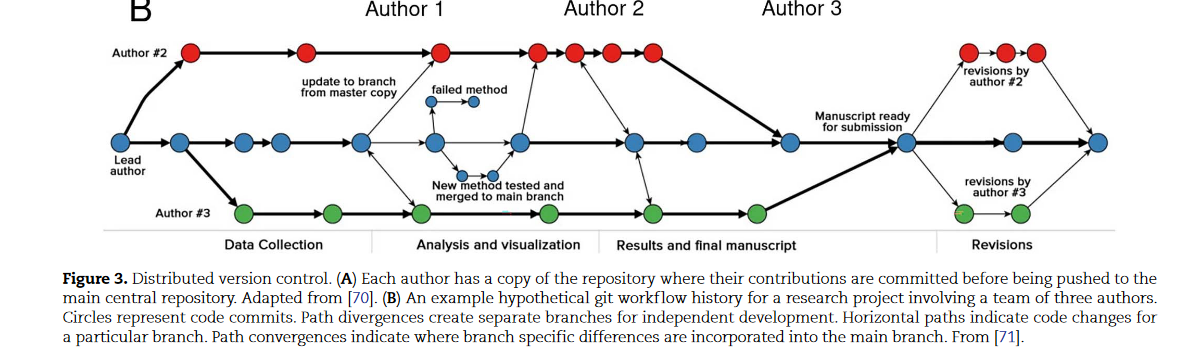
Git est un outil de développement incorporant un contrôle des versions (Distributed Version Control System). Le développeur possède en local une copie du code qui est accessible depuis un serveur central (un hub comme github, propriété de Microsoft, ou Gitlab)

Git se présente généralement sans interface graphique, en lignes de commandes ou bien peut être actionné à travers divers logiciels comme Rstudio, Jupyter, ou l’éditeur VS Code.

Chaque utilisateur récupère le code partagé, le modifie et le renvoie au moyen de commandes simples (pull, push).



Différents embranchements du code peuvent être prévus dans le processus. A la fin, on ne fait qu’un seul code de ces différents embranchements (merge)

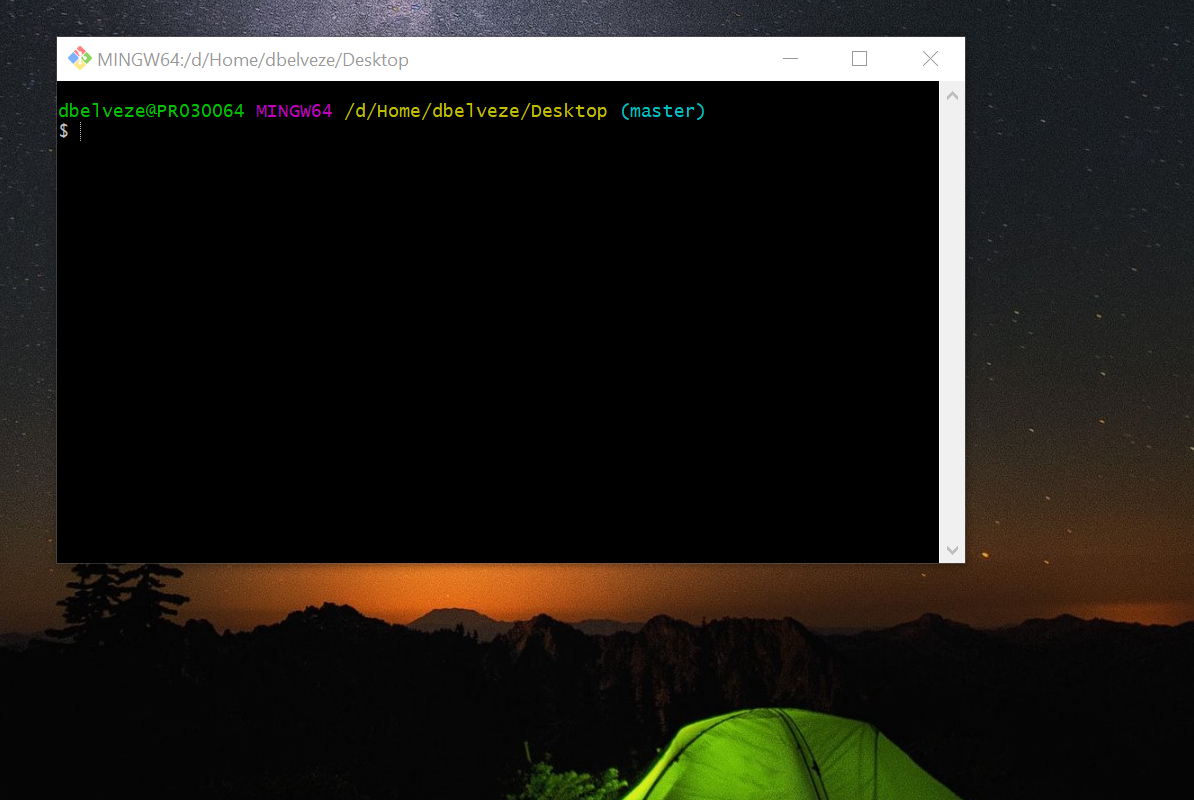
 Git peut également être utilisé pour mener et écrire des études reproductibles

Un DVCS identifie chaque version d’un code et chaque changement apporté à ce code (commit)

Chaque commit comporte la motivation du changement dans le code, donc on peut mieux en comprendre le sens

Les plateformes Git permettent de rendre le code visible, de le partager mais aussi de le discuter avec d’autres et d’obtenir de l’aide pour résoudre les bugs

Télécharger Git Créer un répertoire (repo(sitory)) sous github ou gitlab (on fera ici référence à Github) aller à l’endroit où l’on veut télécharger le répertoire, MAJ+clic droit > ouvrir git bash here



dans ce terminal, pour coller des expressions, utiliser la touche inser

# gérer les fichiers d’un repo

## cloner un repo

Le Ctrl+V ne fonctionne pas dans Git bash, on utilise Shift+Inser. Les nouveaux ordis portables ont deux touches inser, la bonne est le 0 du clavier numérique (quand celui-ci est désactivé)

télécharger le répertoire vide créé

git clone repertoire.git

## initialiser le repo et ajouter des fichiers au track

pour initialiser ce répertoire comme étant un répertoire git (va ajouter un fichier caché .git) :

git init

créer, modifier les fichiers. Pour visualiser les fichiers qui ont été créés/modifiés et ne sont pas encore dans le track de git :

git status

ces fichiers à ajouter apparaissent en rouge

pour ajouter un fichier

git add nomdufichier

pour ajouter tous les fichiers créés ou toutes les modifications de fichiers

git add -A

## commettre et pousser (commit & pull)

Pour préparer les fichier à l’envoi dans le répertoire distant (commettre)

git commit -m "message indiquant la création ou la modification réalisée"

Il est nécessaire de commettre avant d’envoyer des fichiers / modifications vers le répertoire

pour envoyer les modifs vers le répertoire

git push -u origin

pour lier un répertoire git à un répertoire

git remote add origin liendurepository.git

pour supprimer un fichier du track

git rm fichier

## gestion des branches

pour créer une branche à partir d’une branche existante :

git branch nomdelabrancheàcréer nomdelabrancheexistante

une fois cette branche créée, l’envoyer sur le répertoire

git push -u origin branchecréée

## gestion des accès

Depuis 2021, l’accès en mot de passe seul depuis un repo en local n’est plus disponible pour des raisons de sécurité. Les deux modes d’accès possibles sont les suivants :

### ssh

en ssh (se créer une paire de clés ssh et envoyer la clé publique sur le repo)

### avec un jeton personnel

voir documentation [ici](https://docs.github.com/en/get-started/getting-started-with-git/managing-remote-repositories#switching-remote-urls-from-ssh-to-https)

### changer de mode

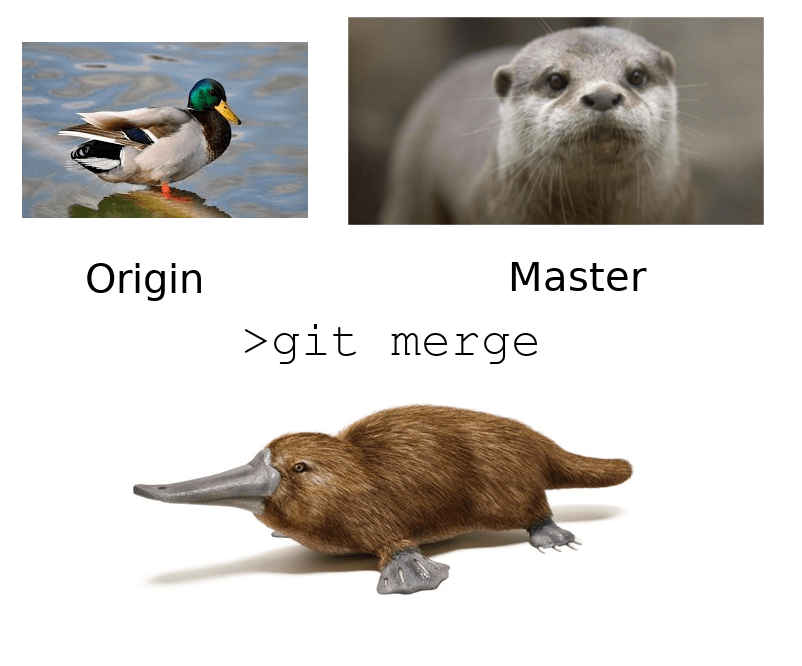
git remote -v

affiche le mode d’accès (ssh ou https (jeton personnel))

Pour passer en ssh :

git remote set-url origin git@github.com:USERNAME/REPOSITORY.git

# Fusionner deux branches



Mode d’emploi sur ces fonctionnalités de base sur Github : https://docs.github.com/fr/get-started/quickstart/hello-world

Gestion des fichiers lourds (par exemple données de la recherche avec git-annex

# Les commandes qu’on utilise toujours sur Git quand on travaille avec un le repo de quelqu’un :

Dans son dossier sur son poste de travail, passer en revue ce qu’on a modifié :

git status

Ajouter les modifications dans le fichier git du répertoire (= ici ajouter tous les fichiers)

git add \*

Committer

git commit -m "mettre dans le message quelques mots-clé sur ce qui a été modifié"

Envoyer vers github

git push -u origin nomdelabranche