

“ Par bien des aspects, vous pouvez considérer Git comme un simple système de fichiers. Il est adressé par contenu, et possède la notion de versionnement, mais je l’ai vraiment conçu en prenant le point de vue d’un spécialiste des systèmes de fichiers (après tout, j’ai l’habitude de travailler sur des noyaux) et je n’avais absolument aucune envie de créer un système de gestion de version traditionnel. ”

Linus Torvald



### Qui ?

Git a été créé par Linus Torvald en 2006, l’auteur du noyau Linux (élément fondamental de systèmes d’exploitation de type Unix qui gère les ressources de l’ordinateur et permet la communication entre hardware et software).

### Quoi ?

Git est un logiciel de gestion de versions, c’est-à-dire qu’il permet de gérer l’évolution du contenu d’une arborescence de fichiers afin de conserver toutes les versions de ces fichiers ainsi que leurs différences, ce qui est très utile dans le développement d’applications.

### Où ?

Git est multiplateforme, il peut être utilisé sur différentes interfaces telles que GitHub (Microsoft), GitLab (libre) ou Bitbucket (Atlassian), des services web d’hébergement et de gestion de développement logiciel.

### Combien ?

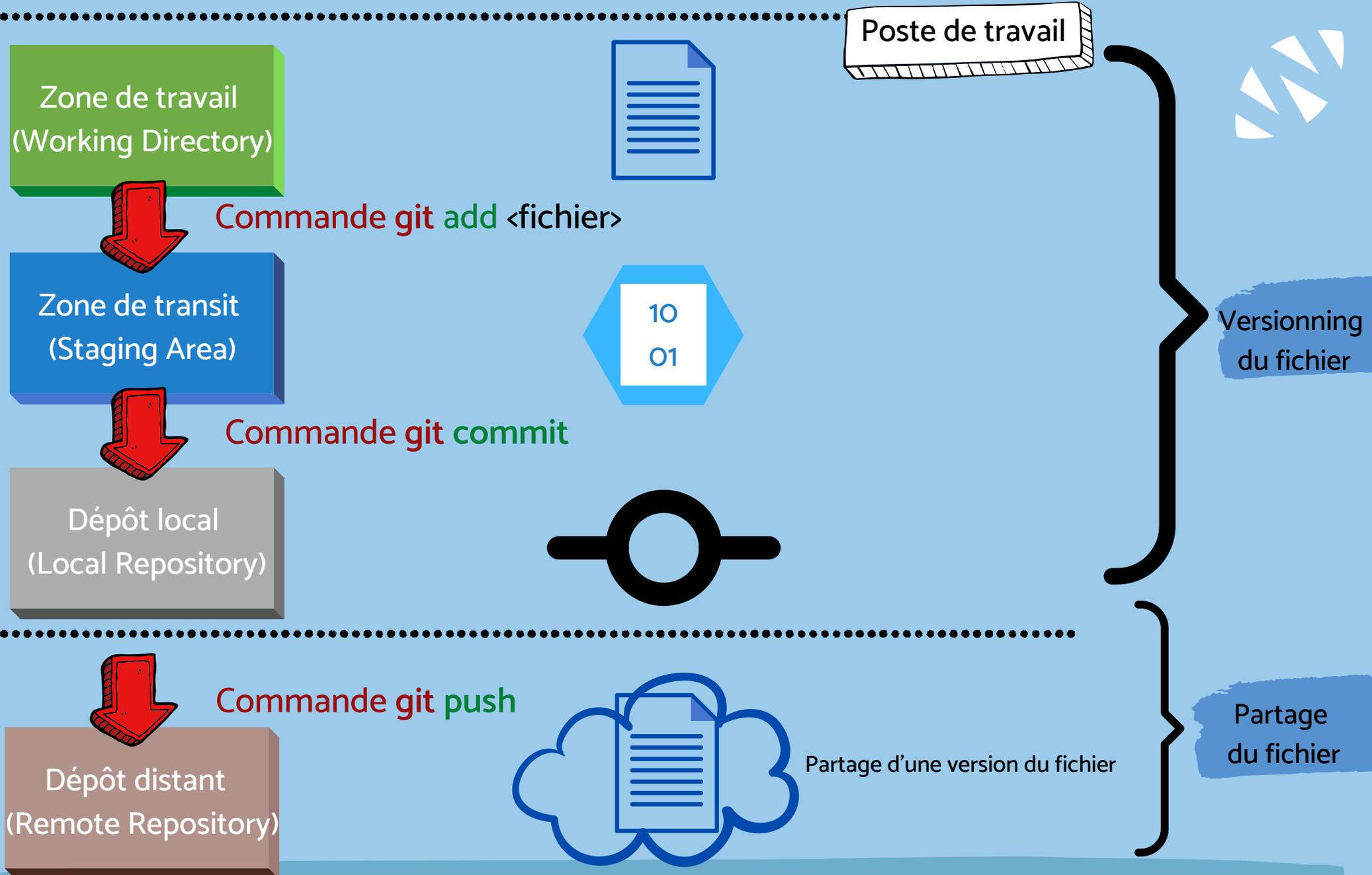
En 2016, 12 millions de personnes utilisaient Git, aujourd’hui 73 millions.

### Comment ?

Git est décentralisé, autrement dit il permet de mutualiser un développement entre plusieurs personnes sur différents postes de travail, et il est plus performant car de ce fait la puissance de calcul est partagée entre ces différents lieux de travail.

### Pourquoi ?

Contrairement à d’autres outils de versionning, Git est libre, distribué selon les termes de la licence publique générale GNU, ce qui permet une utilisation pour tout type de projets, l’étude et la modification du code source, rendu alors disponible. Il est ainsi évolutif et modulaire et permet de répondre aux nouveaux besoins émergents.



L'objectif est de partager, sur une plateforme utilisant Git, une nouvelle version d'un fichier.

Pour ce faire il faut :

- 1 Cloner le dépôt distant en local sur son poste de travail : `git clone`
- 2 Synchroniser le dépôt local avec les dernières mises à jour du dépôt distant : `git pull`
- 3 Ajouter le contenu du fichier, ou blob, dans une zone de transit : `git add`
- 4 Créer une nouvelle version unique du fichier, ou nouvel objet commit, sur le dépôt local : `git commit`
- 5 Pousser ce commit sur le dépôt distant : `git push`

# QU'EN PENSENT NOS EXPERTS ?

Développeur  
full-stack

## CHRISTOPHE

Passions : football, tennis, ping-pong, badminton

Utilisateur quotidien de Git pour du dépôt  
et du partage de sources



“ Là où GIT apporte une vraie plus value par rapport à ses concurrents (SVN, CVS), c'est qu'il est **décentralisé**, c'est-à-dire qu'il est installé en local sur la machine de son utilisateur, qui peut alors travailler comme il l'entend avant de pousser son travail sur un serveur distant. Les opérations réalisées sont également plus rapides car locales. ”

### MES CONSEILS POUR L'UTILISER

- Utiliser correctement le système de branches GIT (isolation du code source d'une nouvelle fonctionnalité avec le code source stable existant en créant une branche dédiée pour cette fonctionnalité).
- Protéger les branches sensibles (code source en version stable) et empêcher les utilisateurs GIT de pousser du code sans passer par une étape de code review (revue de code par un ou plusieurs membres du projet, avec en général au moins un référent technique).

De ce fait, on limite très fortement les mauvaises surprises et l'introduction de bugs.



# DIDIER

## Architecte logiciels



Passions : plongée sous-marine, tennis, randonnées, photographie

Utilisateur quotidien de Git pour des projets d'informatique de gestion, de démonstration et de bases de connaissances

J'utilise Git au quotidien sur l'ensemble des projets que je peux mener chez les différents clients ainsi que sur des projets personnels, essentiellement pour la gestion de code source. Git est un outil assez central dans mes activités de développement de projets, que ce soit en ligne de commande, dans sa version intégrée aux environnements intégrés de développement ou des outils tiers, Git Kraken par exemple.

“ Git est selon moi devenu un standard de fait de ses fonctionnalités, de ses performances, notamment grâce à son système de stockage, de l'historique des modifications et de son approche décentralisée. ”

### MES CONSEILS POUR L'UTILISER

Cela vaut pour n'importe quel outil de gestion de codes sources :

- Une définition précise du modèle de branches utilisé et de manière associée.
- Le workflow de travail, en particulier pour permettre l'automatisation de certaines opérations dans le cadre de l'intégration continue et éventuellement de déploiement continu.