

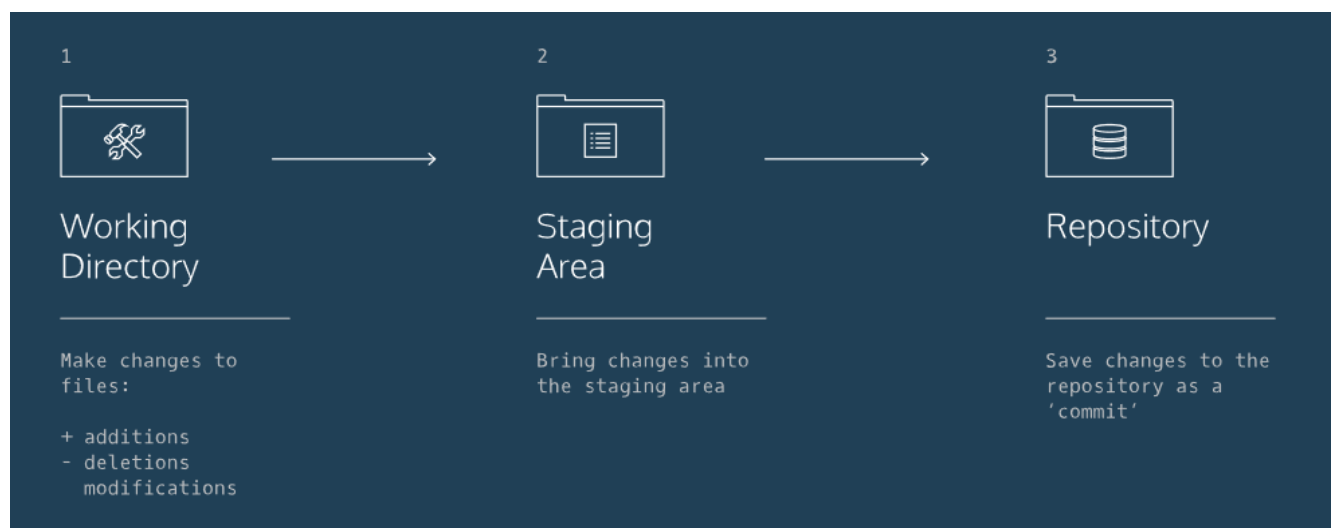


## 1. LE FONCTIONNEMENT

**Github** est une plate-forme payante pour héberger et partager des projets **Git**.

**Git** est un logiciel libre (créé par l'auteur du noyau Linux) pour gérer les versions de projets partagés entre plusieurs utilisateurs/créateurs distants. Il est autonome et ne dépend pas du réseau.

A noter : Il existe d'autres plates-formes ayant les mêmes fonctionnalités que Github tels que Gitlab ou Bitbucket . Sur cette dernière, il est possible de créer 5 projets gratuitement.



Le « **Working Directory** » est le répertoire de travail local de git.

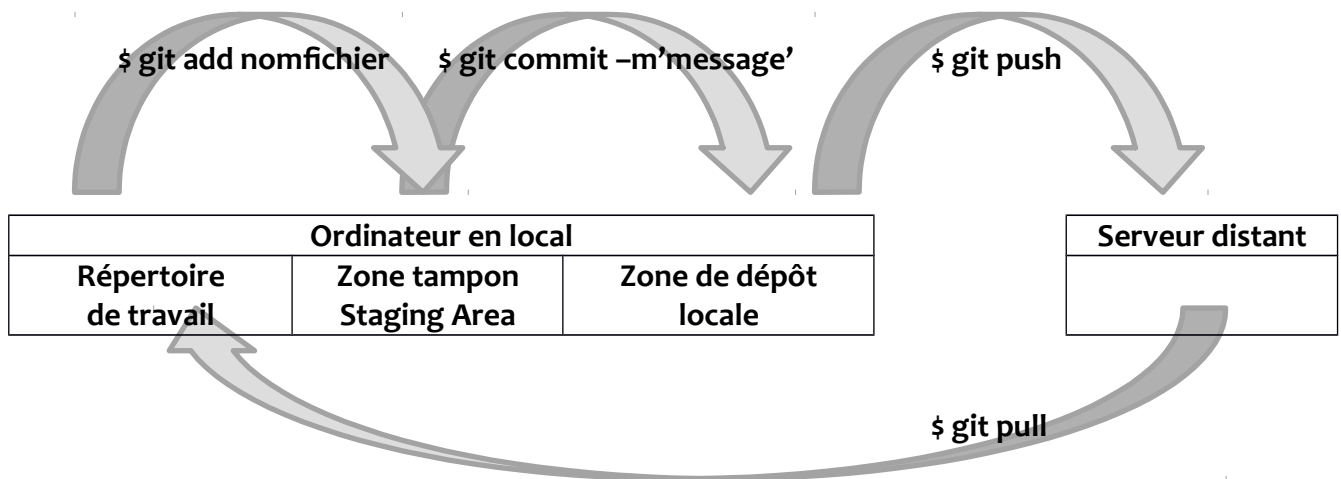
La « **Staging Area** » est une zone tampon entre les fichiers modifiés et le dossier git localisé.

Le « **Repository** » est la zone de dépôt locale avant transfert sur la plate-forme de partage.



**On ne doit jamais toucher au répertoire /.git**  
**Il est « l'entrepôt » qui contient le dossier caché de l'historique**

## 2. LES DIFFÉRENTES ÉTAPES POUR METTRE A JOUR UN FICHIER



## 3. LES COMMANDES PRINCIPALES

Après avoir installé Git en local, lancer Git bash pour travailler sur la console de commande.

Commandes	Action	Remarques
Travail en local		
<code>\$ pwd</code>	Nous indique où l'on se trouve dans le répertoire	
<code>\$ cd +nom-fichier</code>	Change de répertoire	va sur « nom-fichier »
<code>\$ git init</code>	Initialise un projet git en local en créant un dossier <code>/.git</code>	À faire la première fois que l'on crée un projet en local
<code>\$ git status</code>	Inspecte le contenu actuel du fichier du répertoire de travail et de la zone de transit	quand il est modifié le fichier est rouge / quand il passe dans la zone de transit il devient vert
<code>\$ git add +nom-fichier</code>	Ajoute des fichiers du répertoire de travail à la zone de transit	Prépare le prochain « commit » <u>Raccourci</u> : Écrire les premières lettres du nom du fichier + tab
<code>\$ git add '*.txt'</code>	Ajoute tous les fichiers <code>.txt</code> du répertoire de travail à la zone de transit	
<code>\$ git diff</code>	Permet de comparer 2 versions et voir ainsi les changements effectués	En rouge ce qui a été enlevé En vert ce qui a été ajouté
<code>\$ git log</code>	Affiche la liste de toutes les actions effectuées dans le dépôt avec les modifications effectuées.	Pour sortir de l'affichage : « Q »

\$ git commit -m « message »	Stocke de manière permanente les modifications de fichiers dans la zone de transit en lui attribuant un numéro unique de version en hexadécimal	Le commentaire est nécessaire et doit être explicite
<b>Relations avec le serveur distant</b>		
\$ git clone +adresse-sur-github	Crée un clone du document présent sur github	À faire pour importer un nouveau projet distant
\$ git pull	Récupère une nouvelle version du projet sur la plate-forme de partage	
\$ git push	Envoie la nouvelle version du projet présente dans la zone de dépôt locale vers la plate-forme de partage distante	Des droits sont nécessaires pour accéder au serveur distant

#### 4. LA GESTION DES CONFLITS EN 3 ETAPES

Quand il y a un conflit, le « **\$ git status** » indique CONFLICT

<b>Étape 1</b>	« <b>\$ git pull</b> »	pour récupérer la version en conflit
	« <b>\$ git status</b> »	pour savoir ce qui s'est passé et repérer les indications « unmerged paths »
	« <b>\$ git diff</b> »	pour afficher les différences en conflit
<b>Étape 2</b>	Ouvrir le fichier dans l'éditeur	
	Repérer les zones de conflit indiquées entre chevrons  <<<<HEAD >>>>>>> N°schar	
	Enlever la version qui n'est pas bonne + les marqueurs MERGE sur tout le fichier et enregistrer le fichier	
	« <b>\$ git status</b> » pour avoir des indications sur ce que propose le logiciel	
<b>Étape 3</b>	« <b>\$ git add</b> »	pour préparer le MERGE
	« <b>\$ git status</b> »	pour savoir si le conflit est réglé (optionnel)
	« <b>\$ git commit</b> »	pour mettre dans la zone de dépôt et régler le MERGE (NB le message n'est pas nécessaire car la manip ne fait que régler le MERGE et n'a aucune nouvelle modification)

	« \$ git push »	pour le transférer sur le serveur distant
	« \$ git pull »	Vérifie si le conflit est réglé où si un nouveau conflit est détecté