



Qu'est ce que Git ?



git

Version Control System

Solution Open Source

Depuis 2005

Linus Torvald

Innovation

Souplesse

Performance

Pourquoi versionner ?

Enregistrer / stocker la solution de manière à garantir sa pérennité et son intégrité.

Récupérer un état antérieur du projet.

Visualiser les changements dans le temps.

Identifier la source d'un problème.

Les avantages de Git

Il a été pensé nativement pour être **décentralisé** sur un modèle de partage **Peer to Peer**. (chaque Client et aussi Serveur)

Presque toutes **les opérations sont locales**

Il indexe les fichiers d'après leur somme de contrôle (**checksum**) ce qui permet également de **les crypter**.

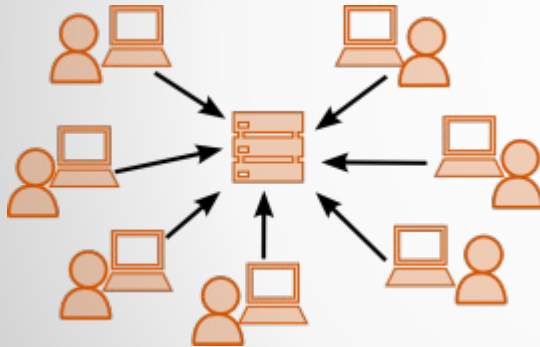
Le **système de branches est plus facile** à employer dans GIT, en particulier les fusions de branches

VCS 2 modèles

Modèle Centralisé

SVN (subversion)

CVS (concurrent version system)

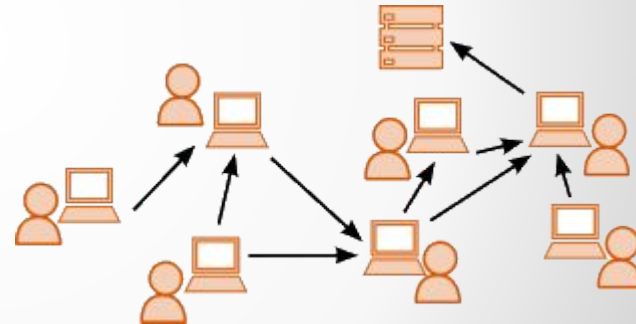


Modèle Distribué

Git

Mercurial

Bazaar



Git : Principe de base

Initialiser un repository local

git init

Pour travailler dans un répertoire avec git, il faut l'initialiser.

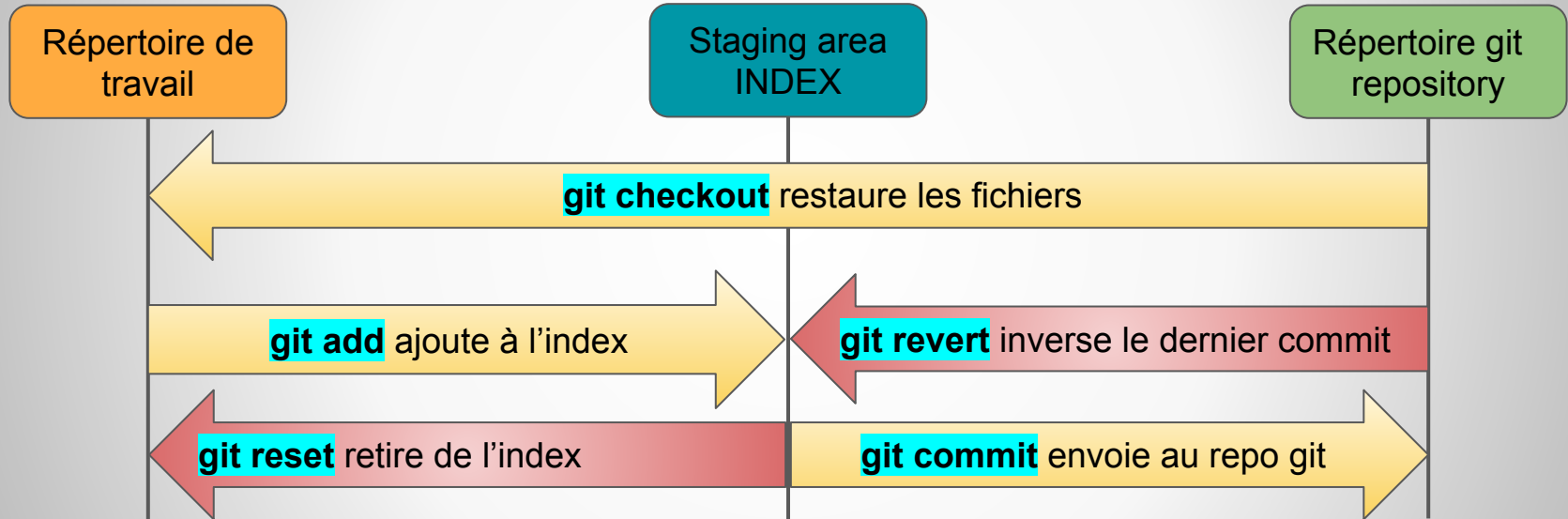
Un dossier .git sera créé.

Il contiendra les fichiers du projet compressé ainsi que la zone de staging, Index.

Cette commande doit être exécutée une fois la console positionnée dans le répertoire cible.

Git : Principes de base

Opérations locales



Git : Principe de base

Commandes de base

git status

Untracked file : indique les fichiers non versionné

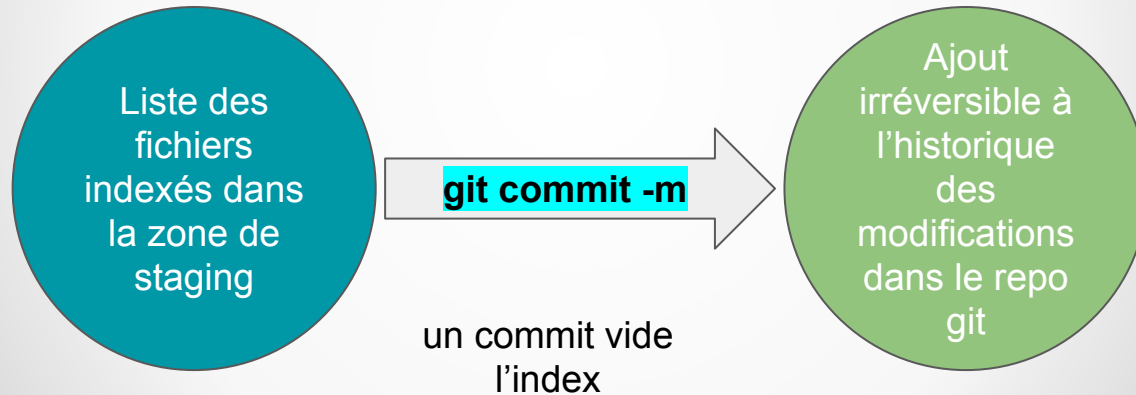
change to be committed : indique les fichiers modifiés et en attentes de commit dans la zone de staging

change not to be committed : indique les fichiers modifiés et non indexé dans la zone de staging

Git : Principes de base

Commit

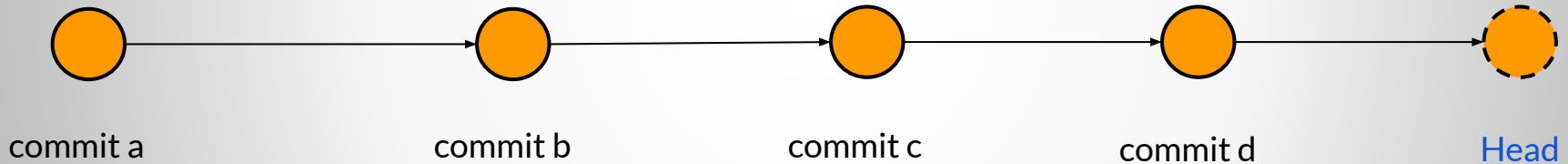
- Enregistrement du travail réalisé ET indexé
 - se réalise sur la branche courante
- un commit est obligatoirement accompagné d'un message grâce à l'option -m



Git : Principes de base

Les Branches

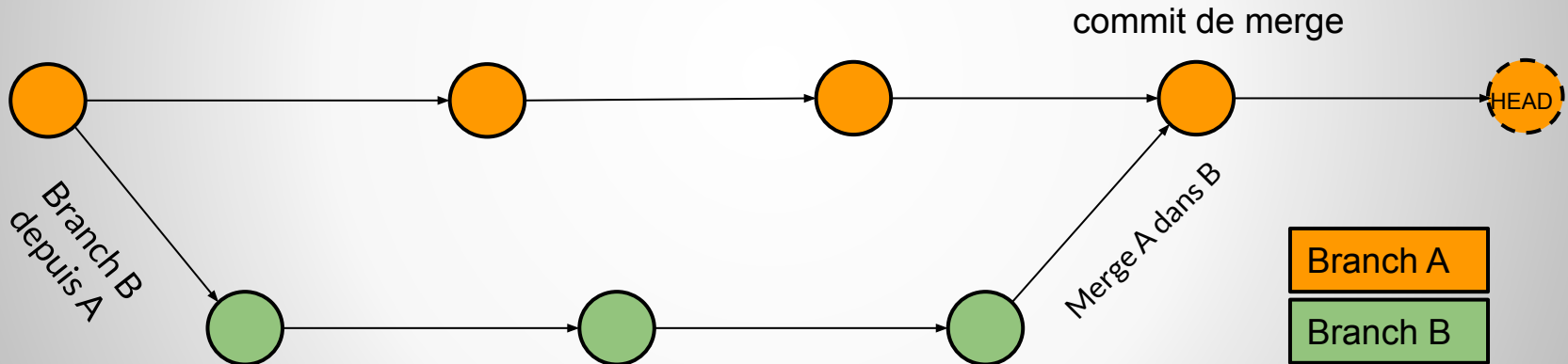
- Suite de commit
- Matérilise le développement d'une feature, fix, ou tâche clairement définie.
- A chaque commit on peut réaliser une nouvelle branch, ou en merger une autre.



Git : Principes de base

Les Branches

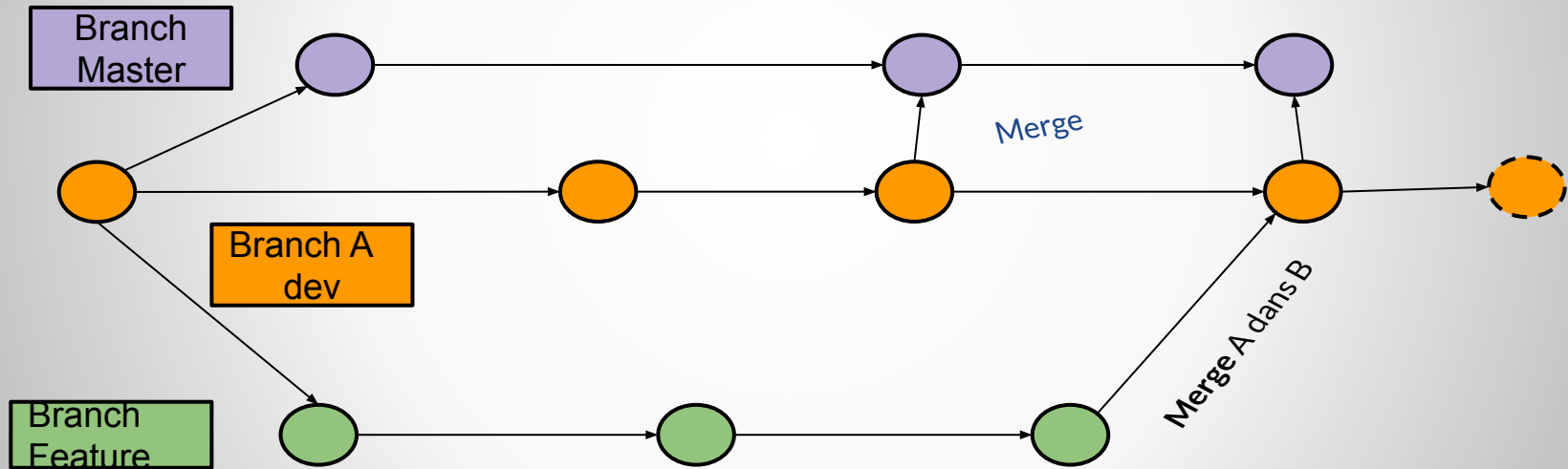
- Le commit de départ d'une nouvelle branche correspondra à l'état initial de la nouvelle branche
 - Pour merger une branche on se place sur la branche cible A puis on ramène la branche B
 - Un commit de merge sera créer sur la branche qui réceptionne.



Git : Principes de base

La branche master

- Branche principale: master. Elle représente l'état fini du projet, aucune modif direct (commit) ne doit être faite sur cette branche.



Git : Principe de base

Commande de gestion de branches

git branch : afficher l'ensemble des branches existantes sur le projet, la branche courante est marquée d'une étoile.

git branch nom_branch : créer une nouvelle branche à partir du dernier commit de la branch courante.

git checkout nom_branch : Changer de branche courante. Aller sur la branch nom_branch

GitHub



GitHub est une plateforme d'hébergement de projet Git

C'est là que l'on va pouvoir créer le dépôt distant.

C'est ce qui va permettre de réellement collaborer.

Il permet d'avoir une vision global du projet de manière plus graphique.

Si le projet n'est pas hébergé en privée, le code source est visible par tous.

L'hébergement privée sur git est payant au delà de 3 collaborateurs

Warning: Exposed Git Repo

Les repos sur Git peuvent être Public ou Privé.

Depuis 2019 et le rachat de GitHub par Microsoft. Les dépôts privés sont gratuits mais limités à 3 collaborateurs. Pour plus il faut passer à GitHub Pro

Il est évident que les **dépôts Public exposent** le code source à la vu de tous, augmentant le **risque piratage...**

!!!! Certaines informations comme les mots de passe de base de données et les clés API ne doivent pas être stockées dans un CVS, même en Privé !!!!

<https://blog.nexcess.net/the-dangers-of-exposed-git-repos/>

GitHub : Principes de base

Dépôt distant

- La collaboration passe par les dépôts distants
- Les dépôts distant sont des versions hébergés sur internet ou le réseau d'entreprise.
- Les droits d'accès sont soit en lecture seule, soit en lecture/écriture.
- il est possible d'avoir plusieurs dépôts distants.
- On peut cloner le dépôt distant dans son dépôt local : **git clone**
- On peut ajouter des modifications depuis notre dépôt local vers le dépôt distant : **git push**
- On peut aussi récupérer, mettre à jour son dépôt local : **git fetch**

GitHub : Principes de base

créer un dépôt distant

Aller sur Github, remplir le formulaire suivant

- Nom de votre dépôt
- Description (en option)
- Public ou Privé (L'option privé est payante)

cliquez sur « Create Repository. »

Il n'y a plus qu'à récupérer l'url du dépôt et la donner à git.

GitHub : Principes de base

Ajouter et cloner un dépôt

Ajouter des dépôts distants

Pour ajouter un nouveau dépôt distant à git

git remote add [nomcourt] [url]

Pour visualiser les dépôt disponible dans git bash:

git remote

l'option -v affiche aussi les url

GitHub : Principes de base

Mettre à jour le dépôt

Il existe deux manières de mettre à jour son dépôt local depuis un dépôt distant.

git fetch [remote-name] : va récupérer les commit de la branch courante que vous ne possédez pas en local.

Sans les merger à votre branch. Il faudra faire un merge pour cela.

L'intérêt est d'éviter d'éventuels conflits.

git pull : va réaliser à la fois la récupération et le merge.

GitHub : Principes de base

Envoyer ses commits sur un dépôt distant

git push [nom-distant] [nom-de-branche]

Cette commande ne fonctionne que sur un serveur sur lequel on a des droits d'accès en écriture et si personne n'a poussé dans l'intervalle. Dans ce cas, il faut mettre à jour son dépôt local avant de pusher.

git request-pull

Si on a pas les droits en écriture sur le serveur, il faut envoyer une requête auprès d'un admin, pour qu'il autorise, ou refuse, nos modifications.