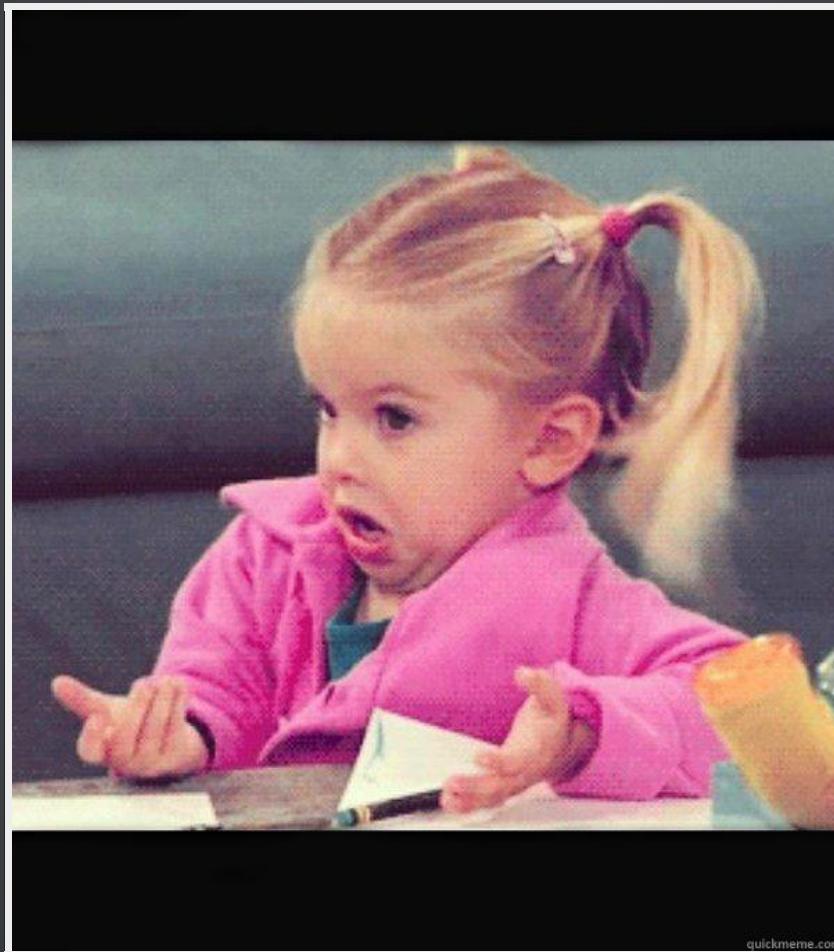


# GIT POUR LES NULS

## RETOUR VERS LE FUTUR POUR TON CODE

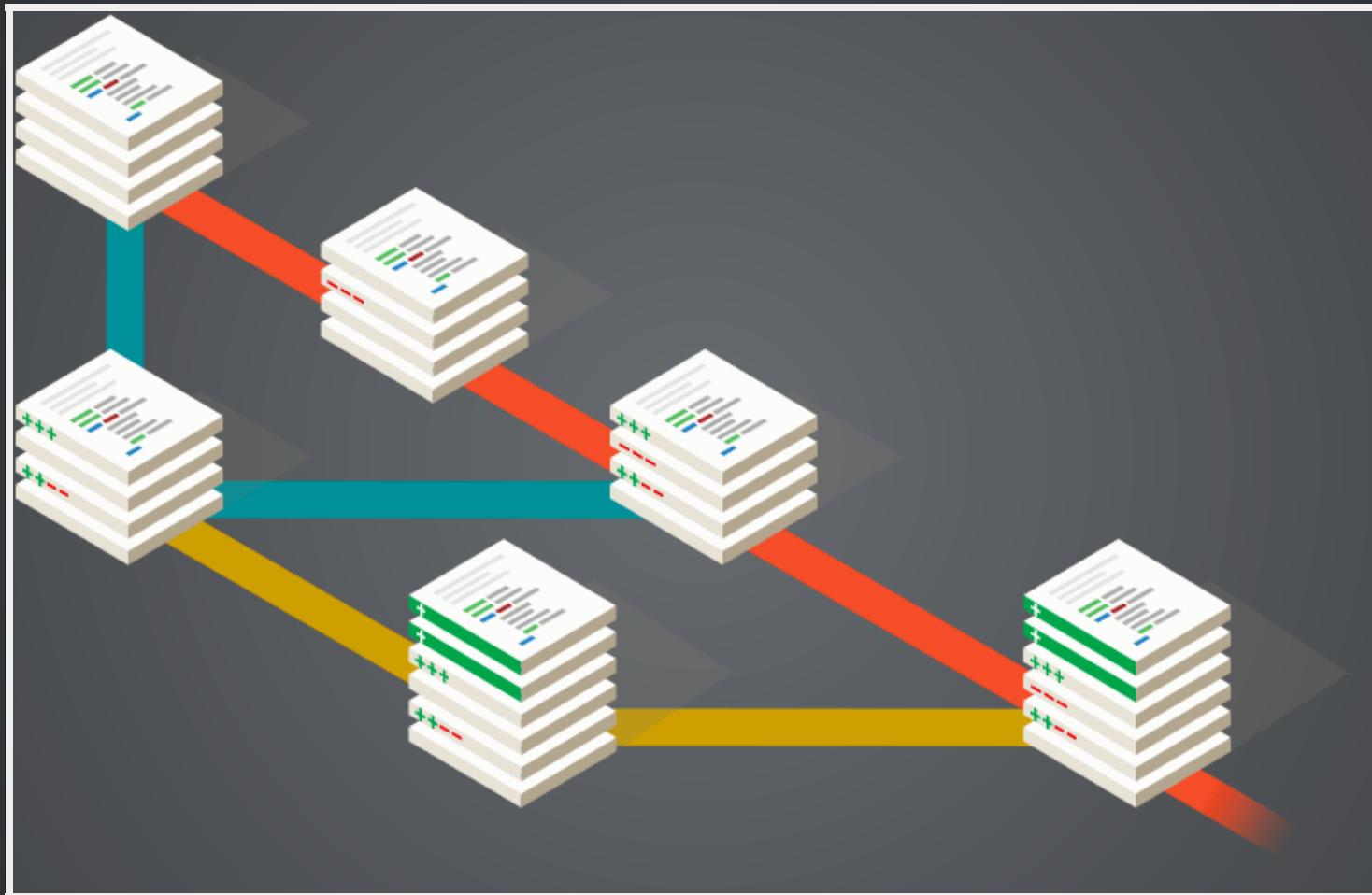


# GIT C'EST QUOI ?

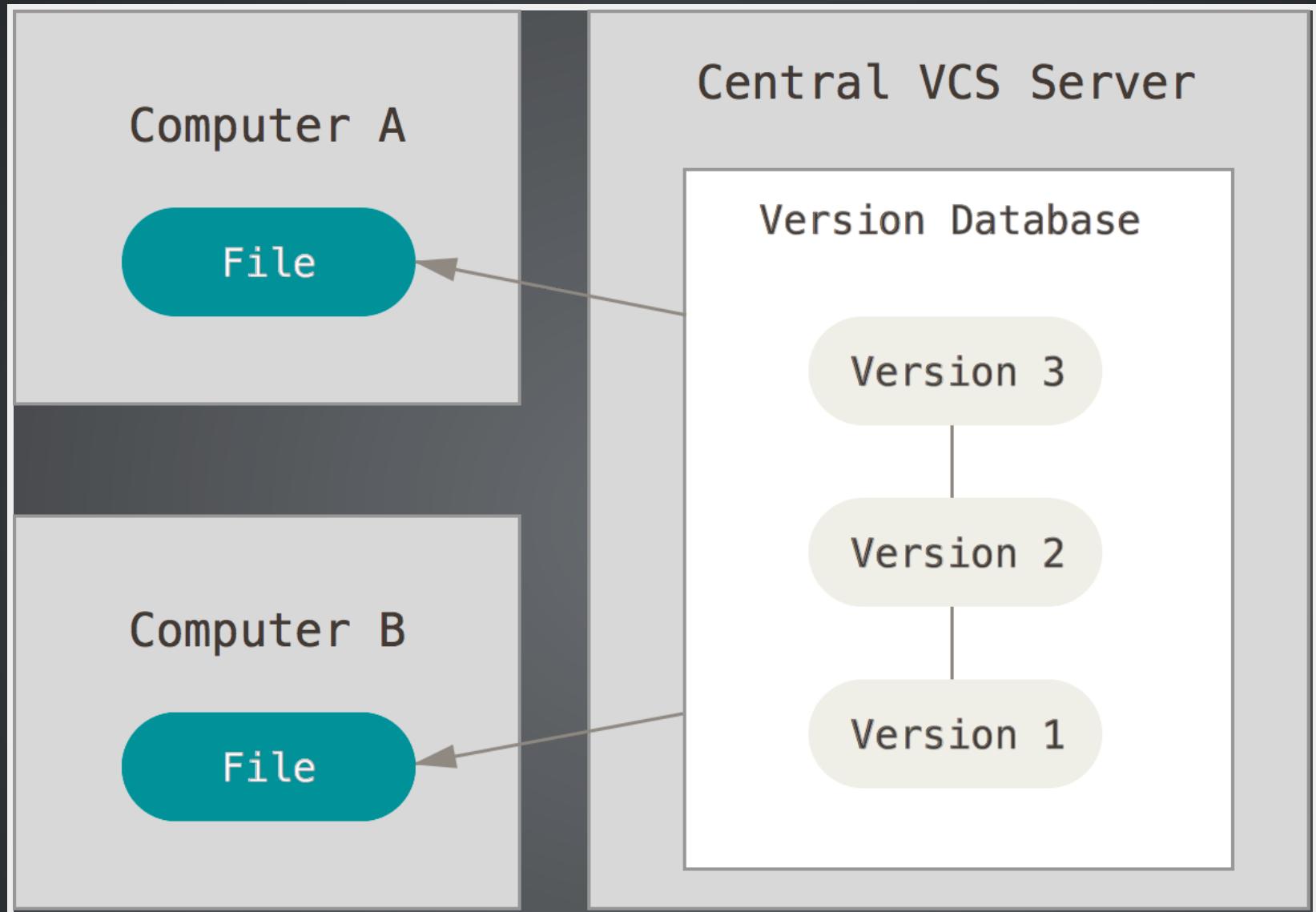


quickmeme.com

# GESTIONNAIRE DE VERSION



**AVANT  
LE DÉPÔT CENTRALISÉ**

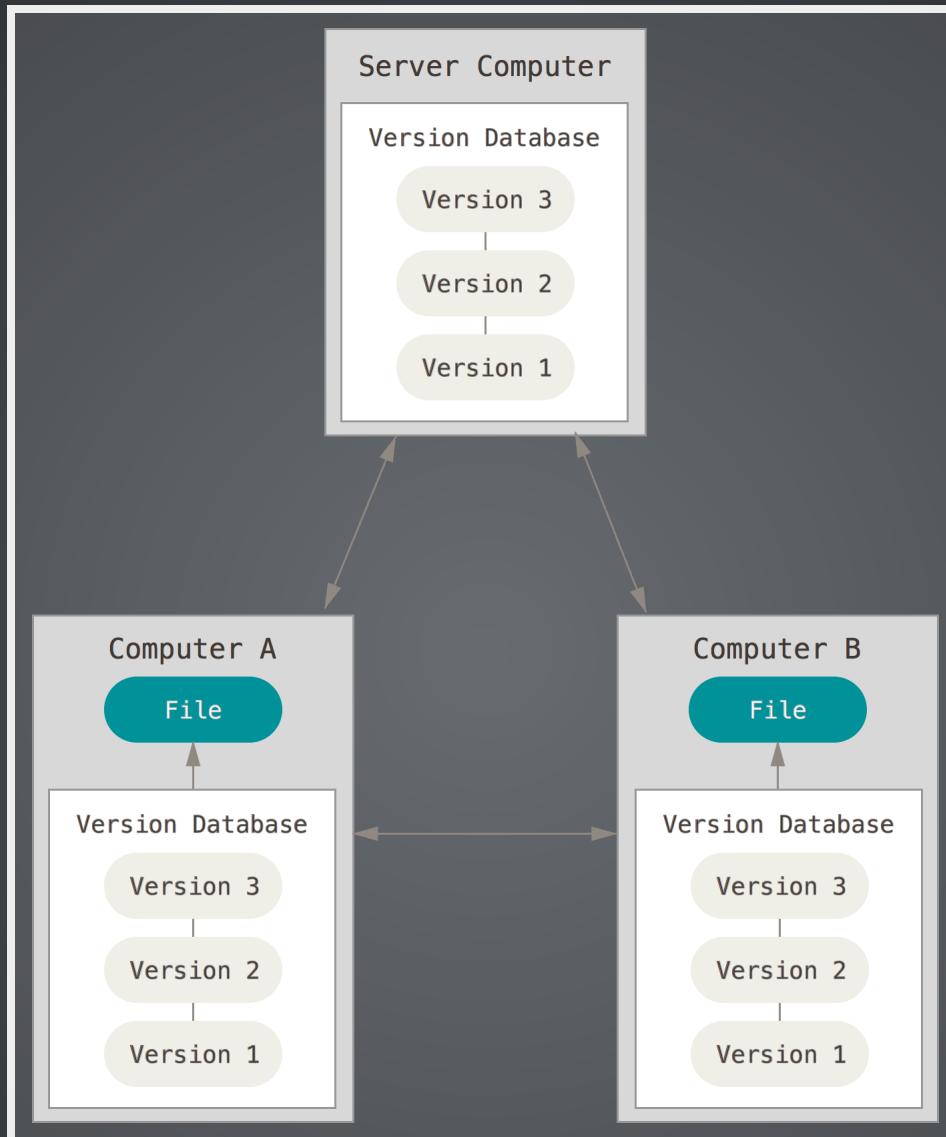


# THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:

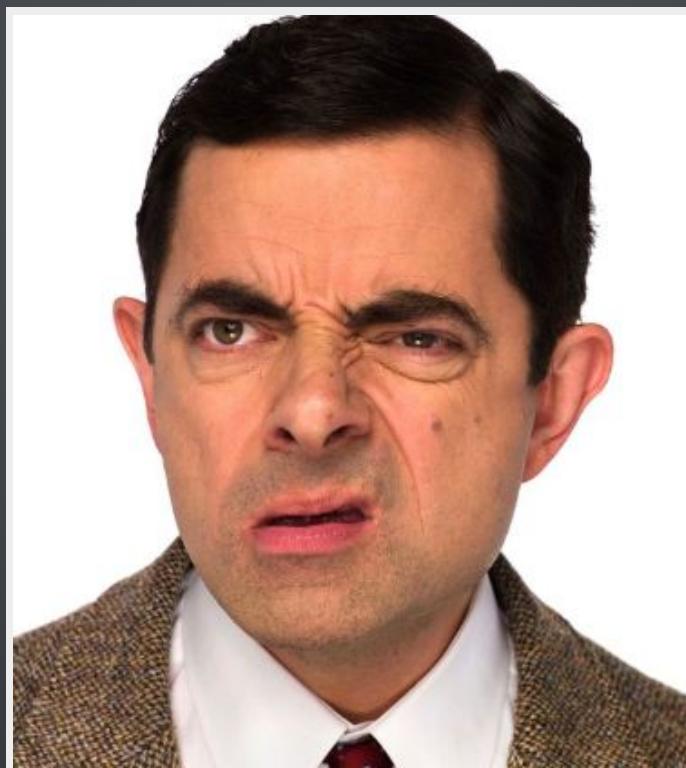
**"The Subversion server's down"**



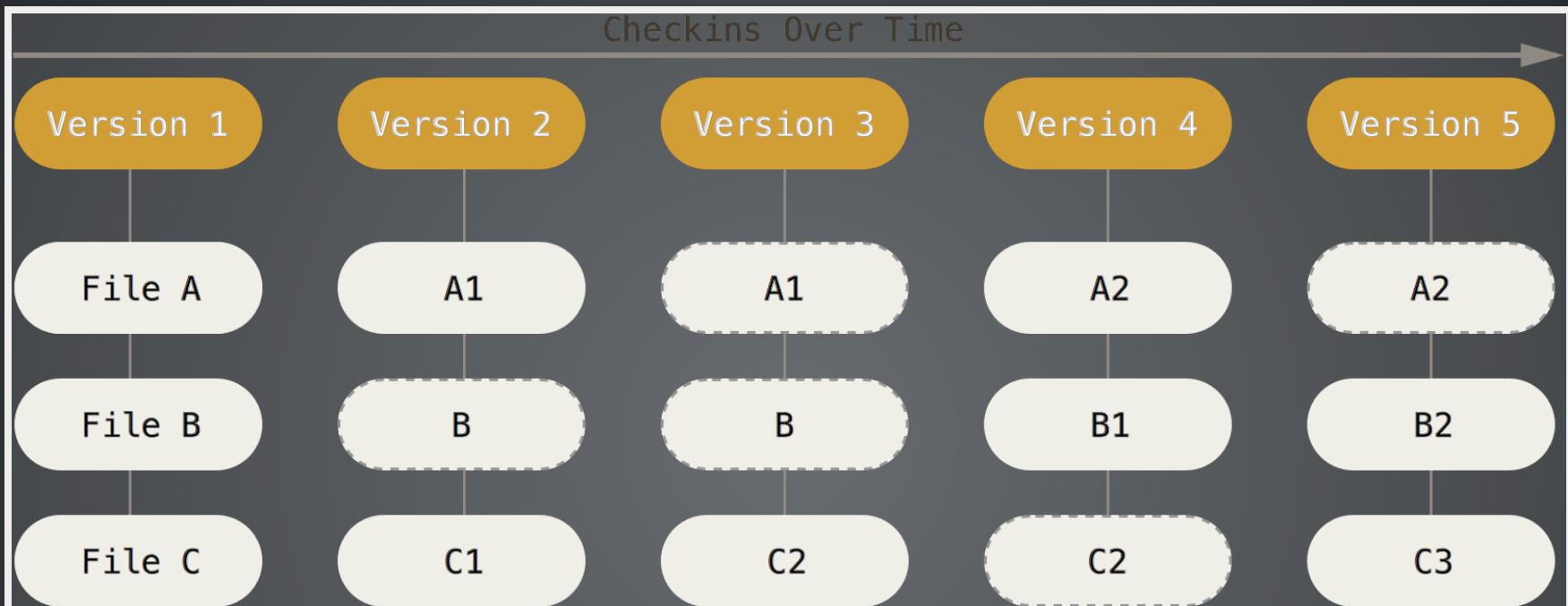
**MAINTENANT  
VOUS ÊTES LE DÉPÔT!**



OKAY ... MAIS EN VRAI  
ÇA MARCHE COMMENT ?

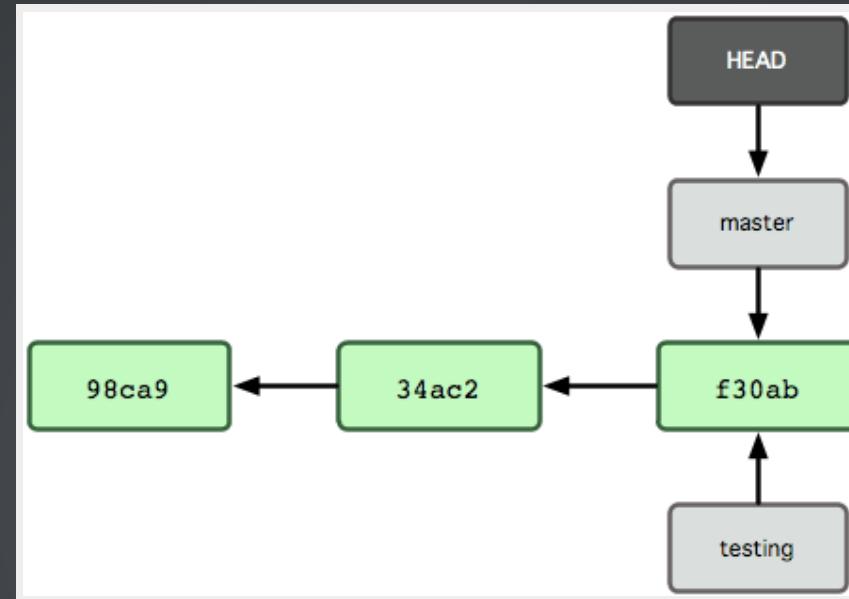


# UN SYSTÈME D'INSTANTANÉS (SNAPSHOTS)



*Un snapshot est créé à chaque  
git commit*

# LE POINTEUR HEAD



# LES DIFFÉRENTS ÉTATS D'UN FICHIER

Un fichier est forcément dans un de ces états

- **MODIFIÉ**

- Le fichier a changé depuis le dernier snapshot

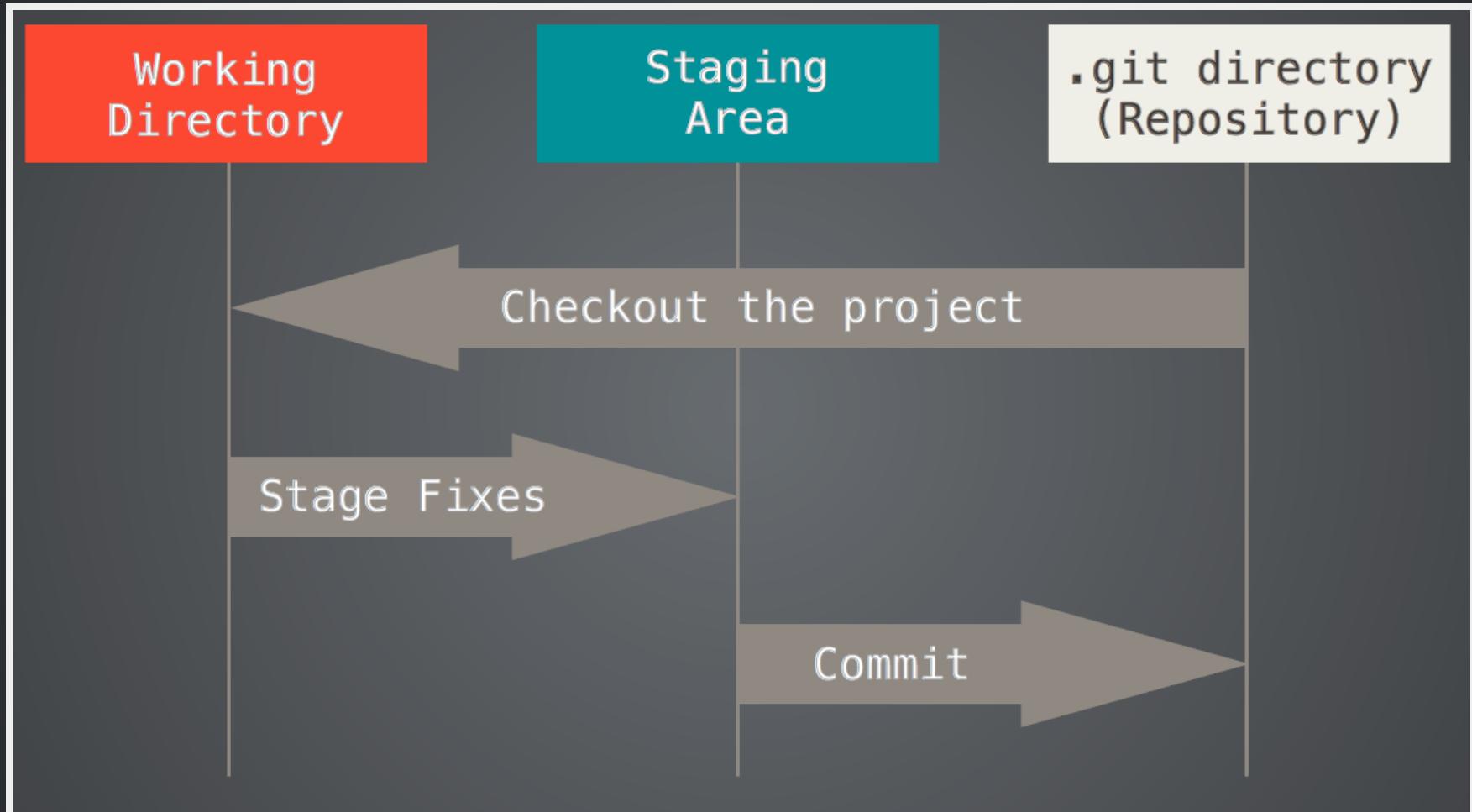
- **INDEXÉ**

- La modification est confirmée et sera intégrée dans le prochain snapshot

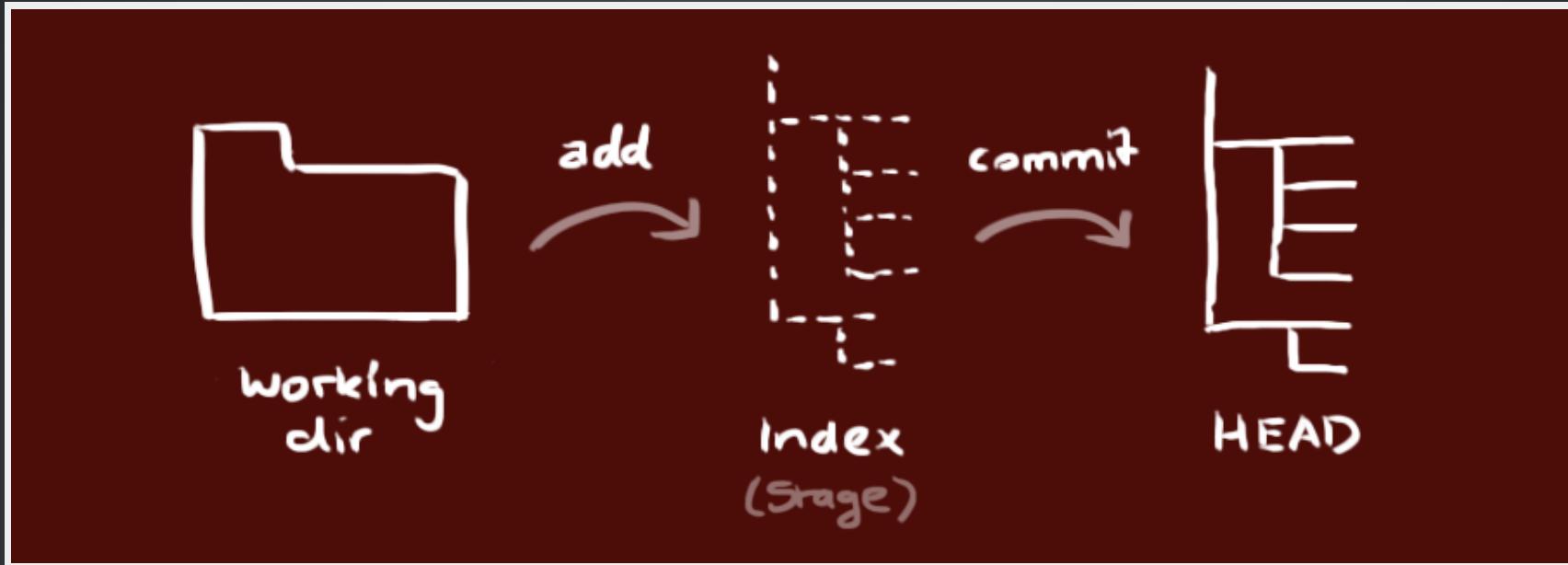
- **VALIDÉ**

- Le fichier a été intégré dans le snapshot

# 3 ÉTATS = 3 ZONES



# ENCHAINEMENT



```
git add .
```

```
git commit -m "Message"
```

# RTFM

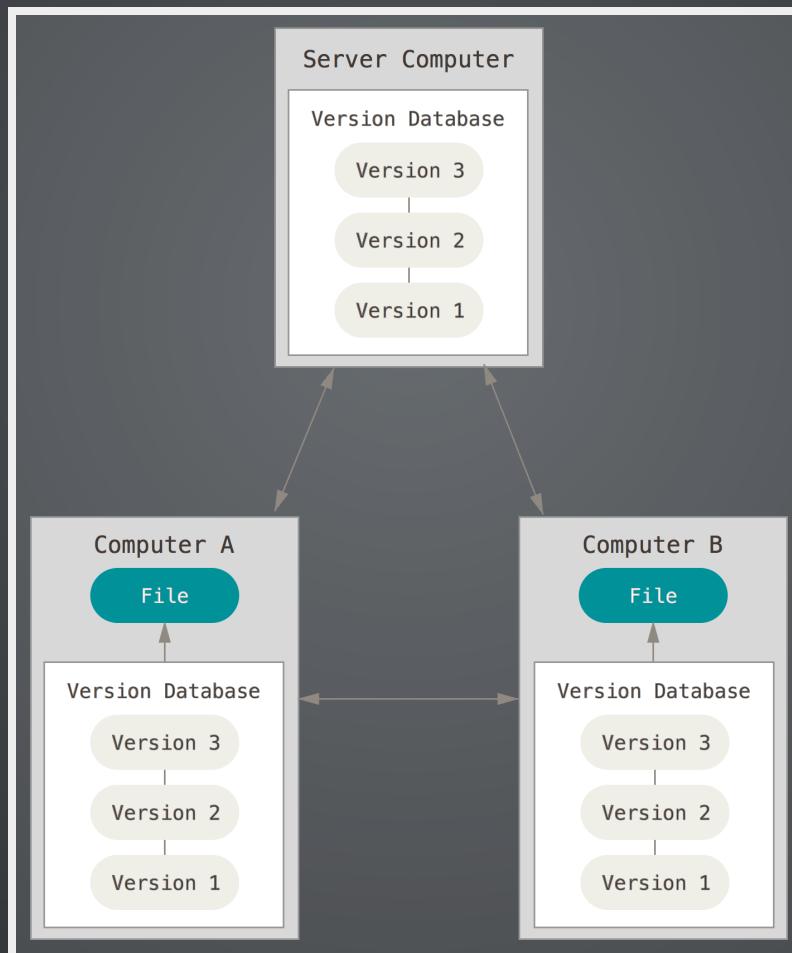
L'aide de Git (<http://git-scm.com/book/fr/v2/>)

# TRAVAIL EN ÉQUIPE



# LES REMOTES

Une remote est un dépôt distant accessible par tous les membres de l'équipe.



# LES REMOTES

Ajouter une remote à son dépôt :

```
git remote add <nom remote> <url remote>
```

*L'url peut être au format http(s) ou SSH*

# PUSH

Publier ses modifications :

```
git push <nom remote> <branche locale>:<branche distante>
```

# PULL

Récupérer les modifications de ses collègues adorés:

```
git pull [<nom remote>] [<branche distante>]
```

# HOOKS

*Un hook (littéralement « crochet » ou « hameçon ») permet à l'utilisateur d'un logiciel de personnaliser le fonctionnement de ce dernier, en lui faisant réaliser des actions supplémentaires à des moments déterminés.*

*– Wikipedia*

# HOOKS

Un hook c'est un endroit dans le processus Git où vous pouvez exécuter votre propre code.

# HOOKS

Deux sortes de hooks git :

Les Hooks Locaux & Les Hooks Distants

- Hooks Locaux

- pre-commit

*Exécuté avant le commit*

- post-commit

*Exécuté après le commit*

- Hooks Distants

- pre-receive

*Exécuté sur le serveur distant avant la réception d'un push*

- update

*Comme pre-receive mais lancé une fois pour chaque branche poussée*

- post-receive

*Exécuté sur le serveur distant après la réception d'un push*

# HOOKS

RTFM

Plus d'infos sur [githooks.com](https://githooks.com)

# GITHUB

# LES FORKS



# LES FORKS

## Copie d'un dépôt

The screenshot shows a GitHub repository page for the project 'thelia / thelia'. The repository has 5,773 commits, 2 branches, 15 releases, and 37 contributors. The current branch is 'master'. A merge pull request #1003 from 'roadster31/toto' has been merged. The latest commit was made 8 hours ago by 'lunika'. Other recent commits include 'add end line on text files' by 'cache' 3 months ago, 'Merge pull request #1003 from roadster31/toto' by 'core' 8 hours ago, and 'Force virtual document download. Fix #974' by 'local' a day ago. The repository has 49 watchers, 319 stars, and 95 forks. The sidebar on the right provides links to 'Code', 'Issues' (67), 'Pull Requests' (6), 'Pulse', 'Graphs', and an 'SSH clone URL'.

thelia / thelia

Watch 49 Unstar 319 Fork 95

Thelia is an open source tool for creating e-business websites and managing online content. Repo containing the new major version (v2) <http://thelia.net>

5,773 commits 2 branches 15 releases 37 contributors

branch: master thelia / +

Merge pull request #1003 from roadster31/toto ...

lunika authored 8 hours ago latest commit 5c46bad790

cache add end line on text files 3 months ago

core Merge pull request #1003 from roadster31/toto 8 hours ago

local Force virtual document download. Fix #974 a day ago

Code Issues 67 Pull Requests 6 Pulse Graphs SSH clone URL

git@github.com:thelia / thelia

# LES FORKS

 **Id4v / thelia**  
forked from [thelia/thelia](#)

[Unwatch](#) 1   [Star](#) 0   [Fork](#) 96

Thelia is an open source tool for creating e-business websites and managing online content. Repo containing the new major version (v2) <http://thelia.net> — [Edit](#)

5,773 commits   2 branches   15 releases   37 contributors

 [branch: master](#) [thelia](#) / + 

This branch is even with [thelia:master](#) [Pull Request](#) [Compare](#)

Merge pull request #1003 from roadster31/toto ...

 **iunika** authored 8 hours ago   latest commit [5c46bad790](#) 

|   |   |              |
|---|---|--------------|
|  cache   | add end line on text files                                | 3 months ago |
|  core    | Merge pull request #1003 from roadster31/toto             | 8 hours ago  |
|  local  | Force virtual document download. Fix #974                 | a day ago    |
|  log   | add end line on text files                                | 3 months ago |
|  setup | Updated hook module.config.js to be specific to a module. | a day ago    |

**Code**

**Pull Requests** 0

**Pulse**

**Graphs**

**Settings**

**SSH clone URL**  
`git@github.com:Id4v/` 

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#). 

 [Clone in Desktop](#)

# LES FORKS

2 remotes:

- origin : Mon fork
- upstream : Le dépôt de base

# LES FORKS

Synchroniser

**git fetch upstream**

*Récupère les modifications du dépôt de base, mais ne les applique pas*

**git checkout master**

*On s'assure de se mettre sur la branche master*

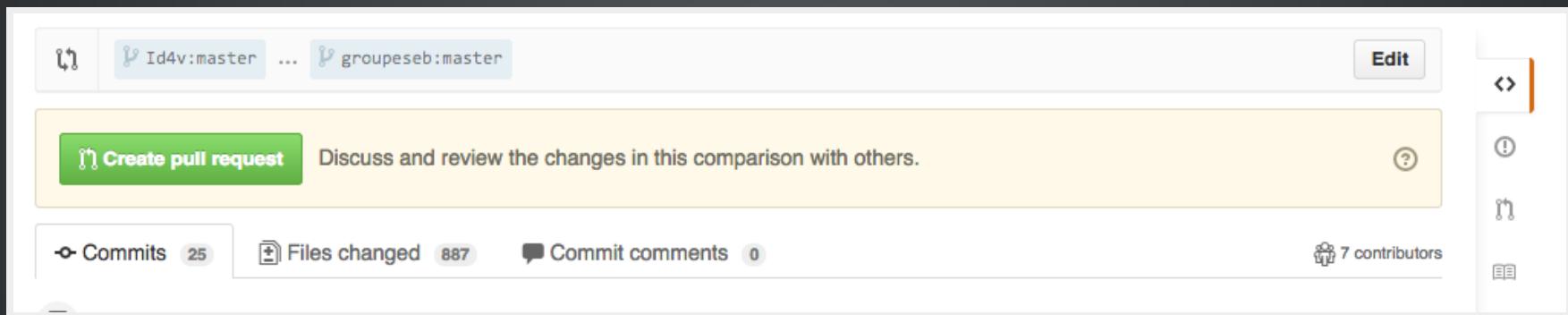
**git merge upstream/master**

*On fusionne les modifs avec les nôtres*

# LES FORKS

Intégrer son code

# LES PULL-REQUESTS



# LES FORKS (& GITHUB)

RTFM

Plus d'infos [help.github.com/](http://help.github.com/)

A black silhouette of a large, gnarled tree against a light blue sky. The branches are intricate and twisted, creating a complex web against the background.

LES BRANCHES

# LES BRANCHES

/!\ Attention travaux! Déviation de code /!\



# LES BRANCHES

Créer une branche

```
git branch <branch name>
```

Créer une branche et switcher direct dessus

```
git checkout -b <branch name>
```

Lister les branches existantes

```
git branch
```

# LES BRANCHES

*Comment switcher de branche avec du code en suspend ?*

## LE STASH

`git stash save`

*Stocke vos modifs à part et remet la branche actuelle au dernier commit*

# LES BRANCHES

## LE STASH

`git stash pop <stash name>`

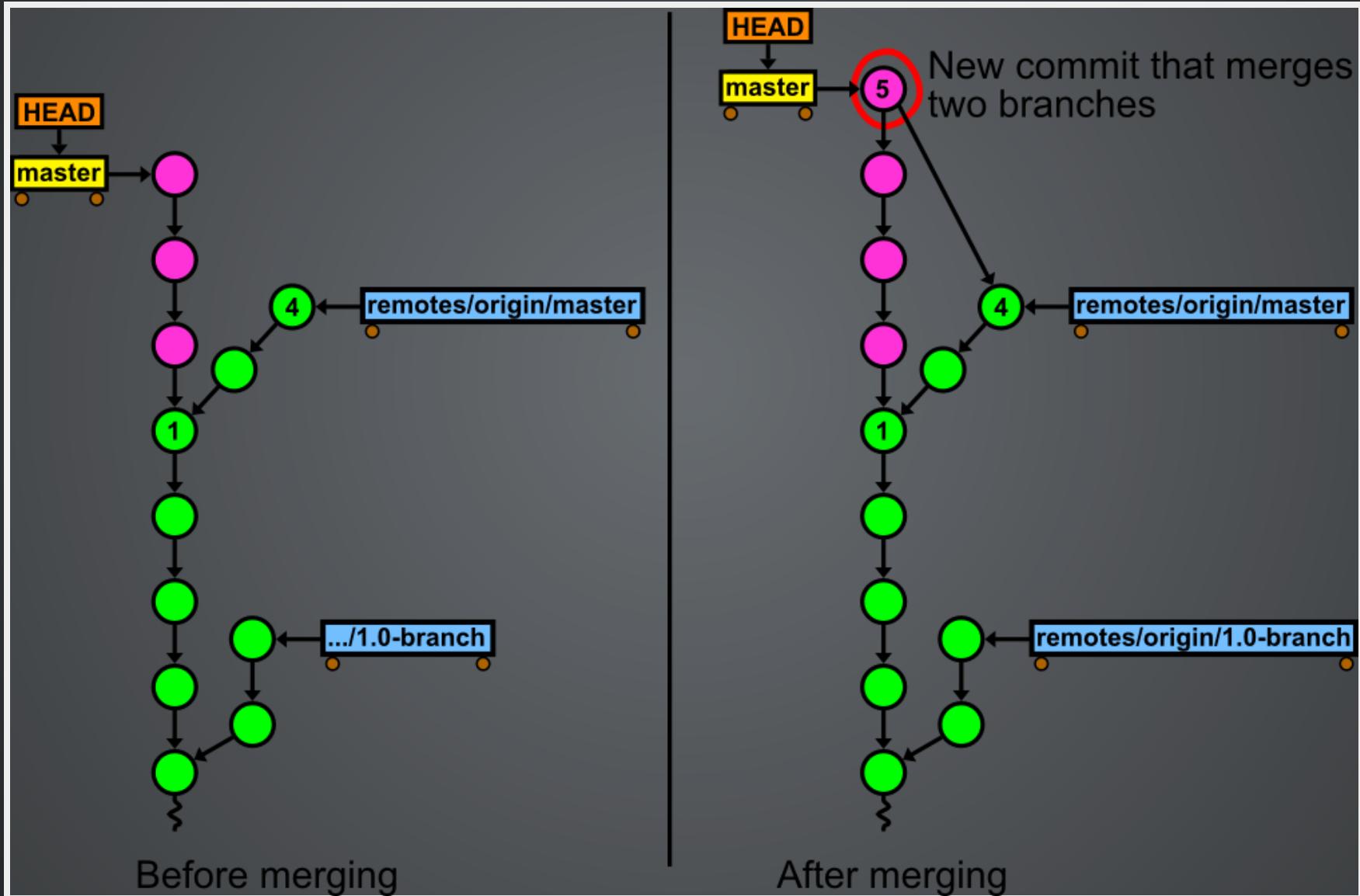
*Stocke vos modifs à part et remet la branche actuelle au dernier commit*

# LES BRANCHES

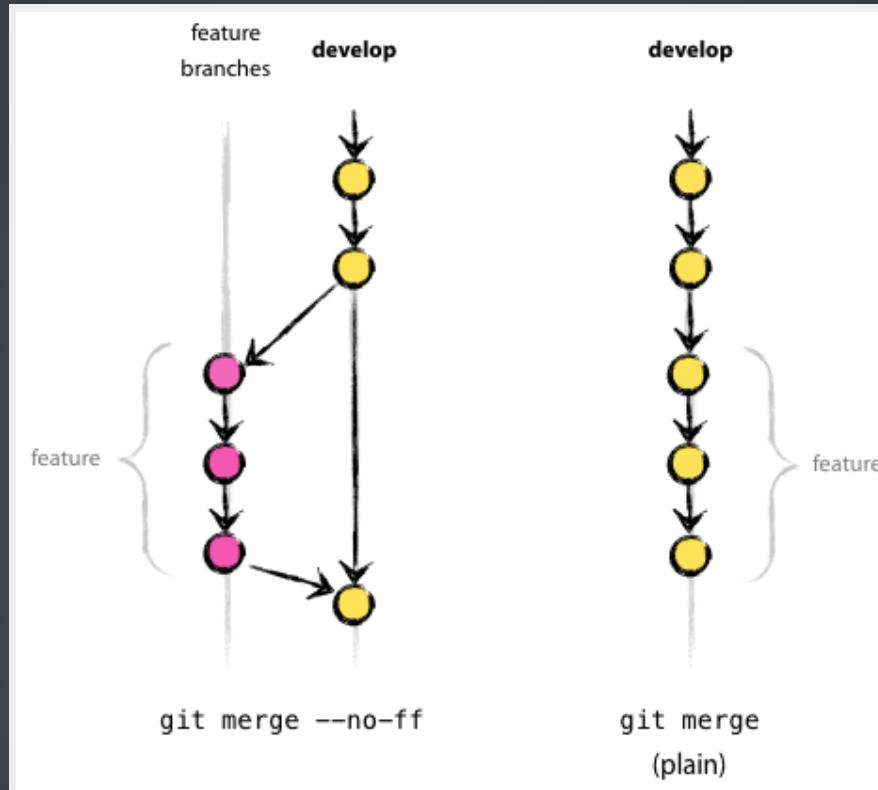
## MERGE

git merge <branch name>

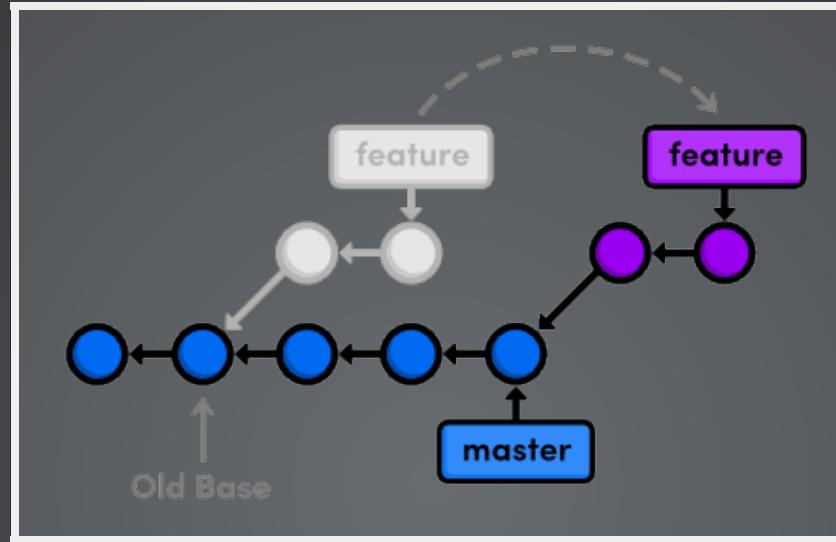
# MERGE



# MERGE



# REBASE



# MERGE VS REBASE

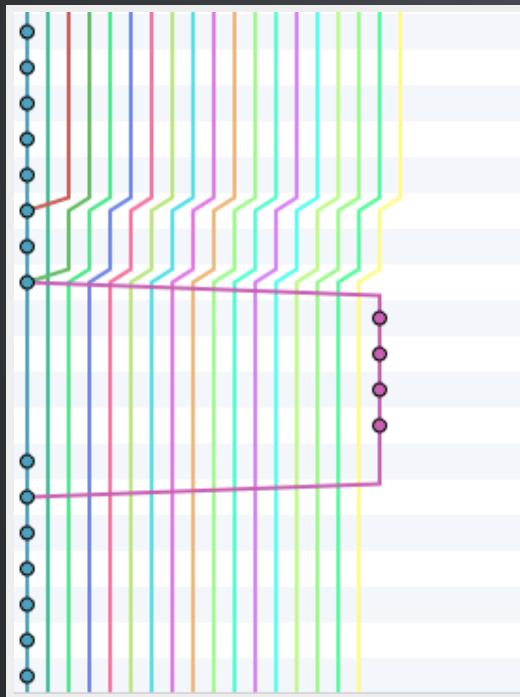
Résultat identique, but différent.

## MERGE

- Branche publique (utilisée par l'équipe)
- Si la branche représente un Sprint Agile ou une fonctionnalité qui pourra évoluer par la suite
- Il faut que l'historique de cette branche reste distinct et visible.

# REBASE

- Branche locale, pas partagée
- Pas besoin de garder l'historique
- Eviter des commits inutiles, ou un historique de branche du style



# MERCI



Des questions ?