



git

&



GitHub

Système de Gestion de Version

Version Control System



Qu'est-ce donc ?

C'est un système qui permet de stocker du code source, des fichiers de configuration, des documents, ... tout en sauvegardant les étapes d'avancement (versions) c'est à dire les modifications apportées.

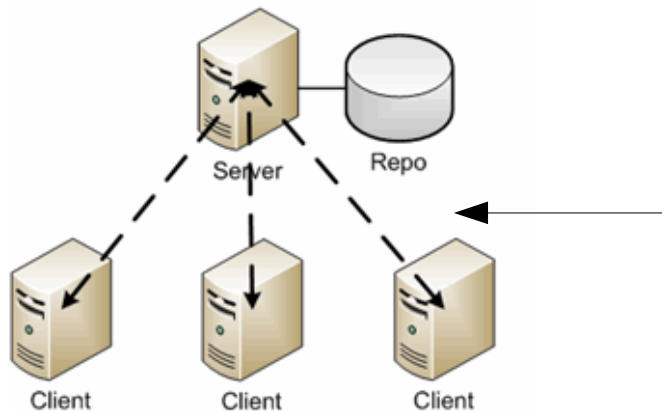
⇒ Nécessite de remonter très régulièrement ces éléments au système de gestion de version au cour du développement.

Systeme de Gestion de Version

Version Control System

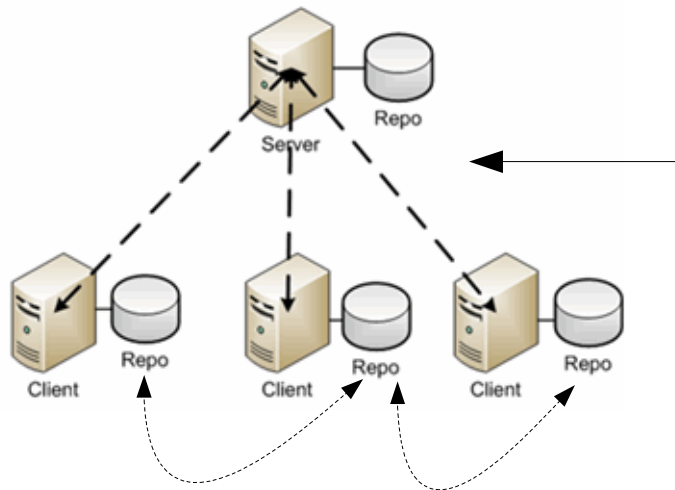


2 types existent :



Gestionnaire centralisé : le stockage de référence se fait sur un serveur
→ dépôt distant.

Subversion (SVN), CVS



Gestionnaire distribué : chaque dépôt est réparti et peut-être considéré comme une dépôt de référence au même titre que le dépôt distant.

Git, Mercurial, Bazaar

Repo = dépôt

Apports

- Avantages

- Facilite grandement le travail en équipe :
 - Travail simultané
 - Fusion des modifications
- Trace le travail effectué, permet de revenir en arrière, permet de travailler sur plusieurs versions en même temps.
- Évite les erreurs liées à la recopie de code

- Inconvénients

- Faut s'y mettre !
- Ça prend du temps sur le développement logiciel, non ?
 - Oui mais au final on y gagne !

Fonctionnement (1)

- Les branches → Exemple : git



Adopter une stratégie pour les branches :

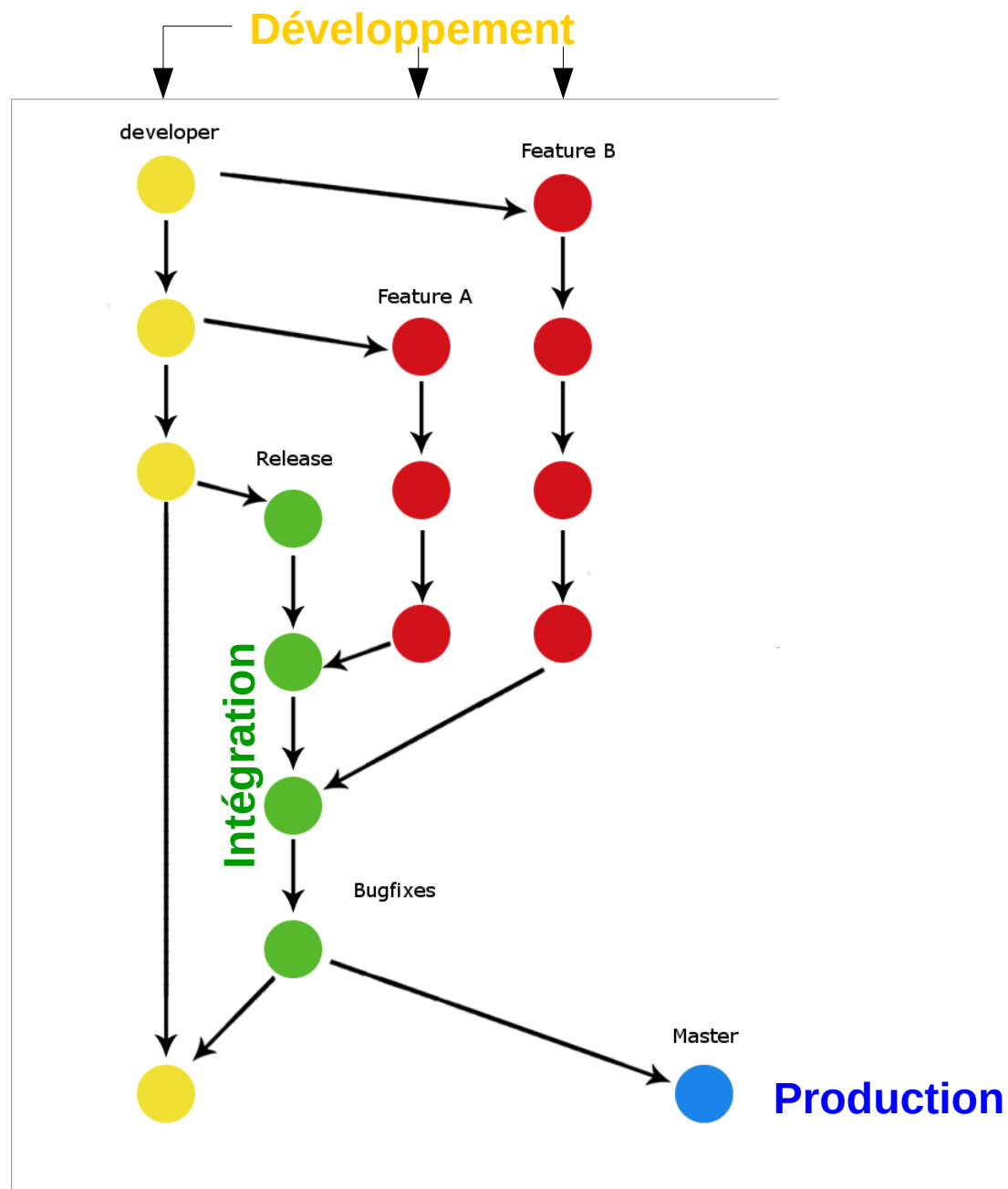
Développement / Intégration / Production (master)

et en fonction de :

- *La taille de l'équipe*
- *La complexité des fonctionnalités à développer*

Fonctionnement (2)

Exemple :



Fonctionnement (3)



- Concurrence de modifications

Par un système de gestion des conflits

Par verrouillage : fichiers

- Trace des modifications

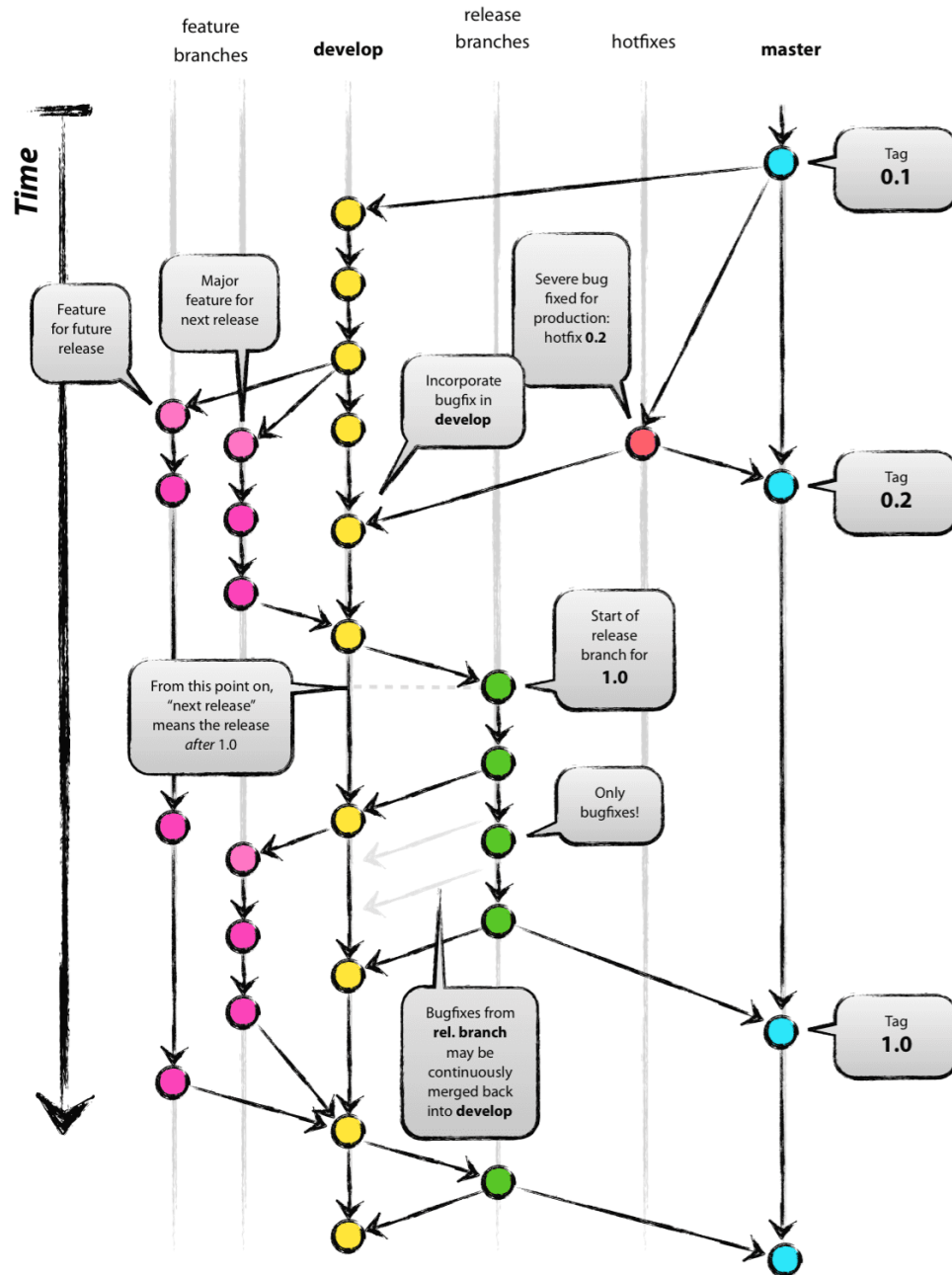
Code source des versions + Historique avec messages du développeur

- Étiquetage (tag) → Pour les livrer en production

version x.y.z (x: nouvelles fonctionnalités, y: évolutions, z: correction de bugs)

- Travail avec une copie locale du dépôt

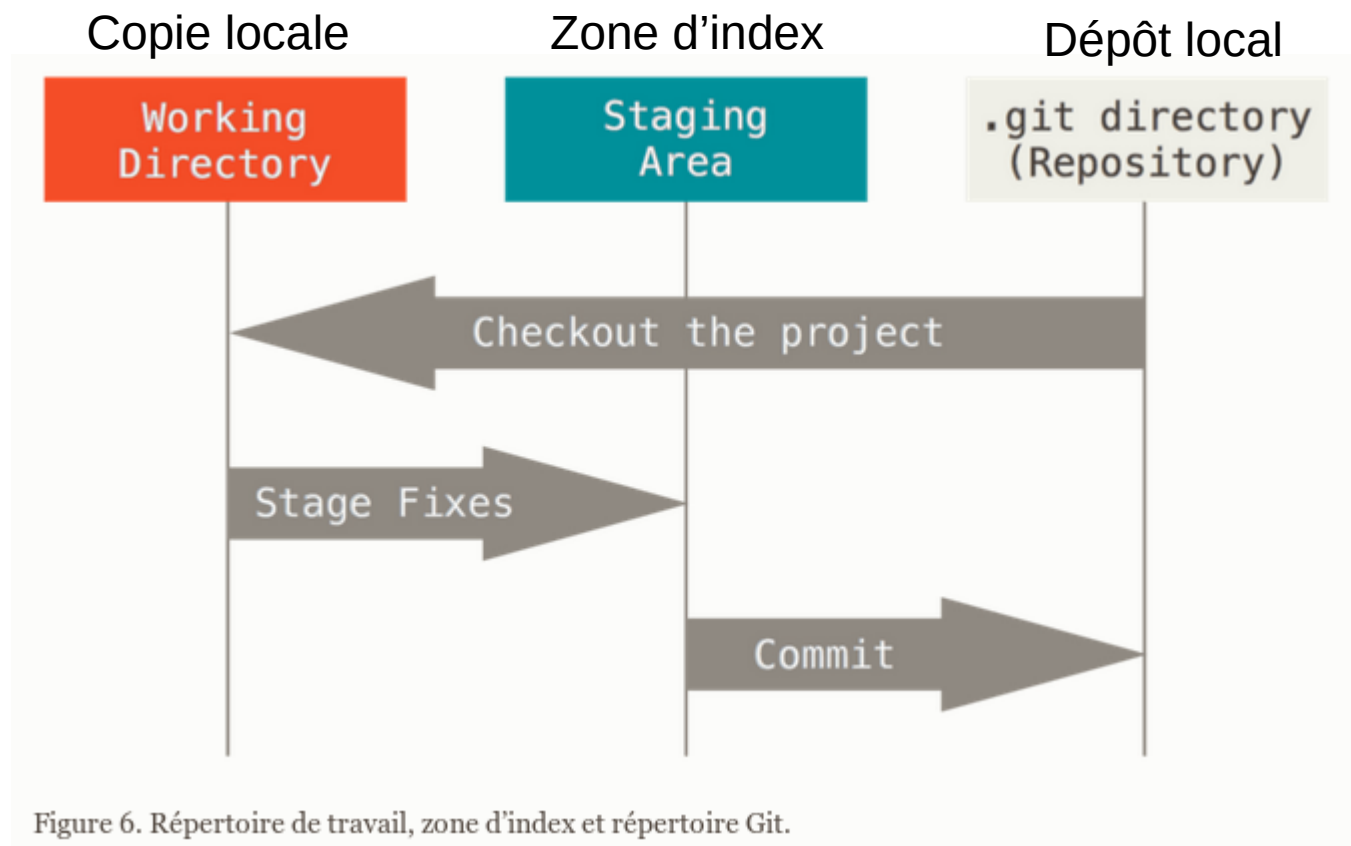
Fonctionnement (4)



*Exemple
d'arborescence de
branches très
complet :*

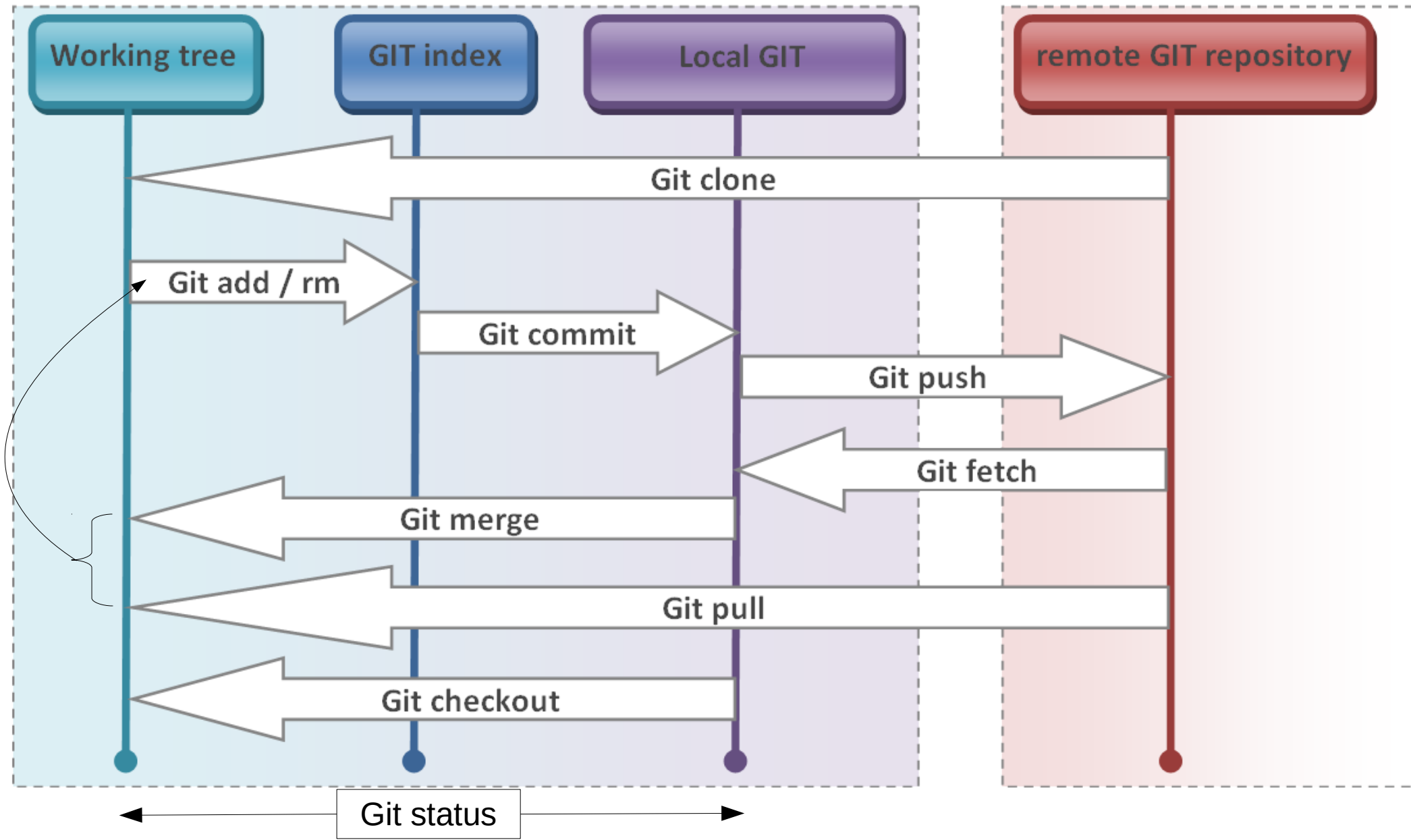


Sur ma machine



⇒ ~ 150 commandes !!!

Git log



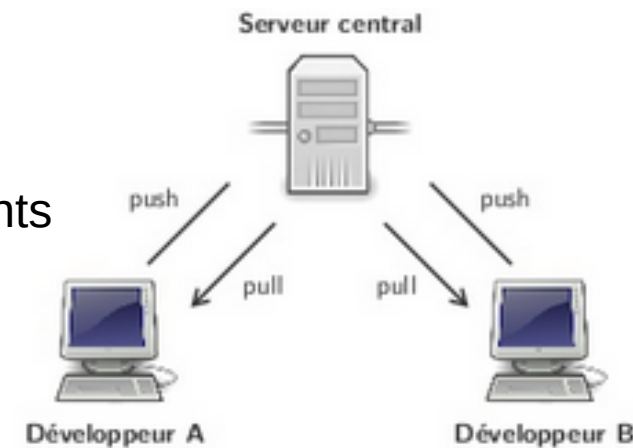


- Clone => Récupération du dépôt distant, création du dépôt local.
- Add => Ajout de fichiers/répertoires (modifications) à la zone d'index.
- Commit => Remontée des modifications au dépôt local depuis
la zone d'index.
- Push => Remontée des modifications au dépôt distant depuis
l'espace de travail.
- Fetch => Récupération des modif. du dépôt distant dans dépôt local.
- Merge => Récupération des modif. du dépôt distant depuis le dépôt local.
- Pull => Fetch + Merge
- Status => État des fichiers/répertoires
- Log => Historique des opérations.

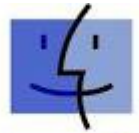


- Créé en 2008, racheté par Microsoft au printemps 2018
- 28 Millions d'utilisateurs, 80 millions de projets
- Développement collaboratif
 - Outils de communication
 - Wiki
 - Travail en équipe
 - gestion de version avec Git
 - tableau de bord, gestion d'anomalies, etc
 - Gestion de communauté
 - open source
 - Accès public gratuit
 - l'accès privé est payant sauf pour les étudiants

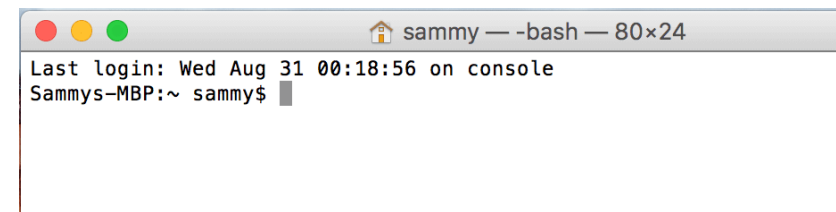
⇒ DL2018: Utilisation de GitHub comme
dépôt de référence distant !



Projets DL / CDI



Mac OS



TortoiseGit
Windows Shell Interface to Git



Dépôt

Gestion des Branches

Current Repository
mytestrepo2

Current Branch
master

Fetch origin
Last fetched 5 minutes ago

Changes

History

README.md

2 changed files

anotherfile.md

README.md

1

2

3

4

1

2

3

4

@@ -1,2 +1,4 @@

mytestrepo2

a test repo

+

+this is another test repository.

Summary

Description

Commit to master

Actions entre dépôt local et dépôt distant

Commit

Fichiers modifiés
(Add automatique)



Au démarrage :

- Créer un dépôt sur GitHub
- Récupérer le dépôt (clone)
- Créer 2 branches :
 - 'Release' depuis 'master'
 - 'Develop' depuis 'Release'
- Créer le projet avec le code source et l'environnement de développement (projet NetBeans) dans la branche 'Develop'
- 'Committer'
- Remonter sur le dépôt distant (push)

⇒ NB : 'Origin' dans Git et GitHub = dépôt distant



Au cour du projet:

- Continuer le développement sur la branche 'Develop'
 - 'Commit' et 'Push' réguliers !!!
- A chaque fonctionnalité développée, créer une 'Pull Request' (depuis GitHub) vers la branche 'Release'
- Tester le fonctionnement dans la branche 'Release' après chaque développement : existe-t-il des régressions ?
 - ⇒ Capital si l'on travaille en équipe !
- Lorsqu'une version est stable, créer une 'Pull Request' (depuis GitHub) vers la branche 'master'
- En profiter pour créer un tag.



Pull Request



Search or jump to...



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



[AFPA38](#) / [testgithub](#)

Watch ▾

0

Star

0

Fork

0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

No description, website, or topics provided.

Edit

[Add topics](#)

1 commit

1 branch

0 releases

1 contributor

Your recently pushed branches:

develop (less than a minute ago)

Compare & pull request

Branch: **master** ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

AFPA38 Initial commit

Latest commit afa3a13 4 minutes ago

[.gitignore](#)

Initial commit

4 minutes ago

[README.md](#)


Initial commit

4 minutes ago



Pull Request (2)



 Search or jump to... / Pull requests Issues Marketplace Explore


AFPA38 / testgithub


Watch 0 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#) [Settings](#)

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

 base: master compare: develop ✓ Able to merge. These branches can be automatically merged.




1er commit bidon

Write Preview

AA B i “ < > ↻ ⋮ ⋮ ⋮ @ 📌 ↶

Leave a comment

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

 Styling with Markdown is supported

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Create pull request

1 commit

2 files changed

0 commit comments

1 contributor





Pull Request (3)




1er commit bidon a985f6e

Add more commits by pushing to the **develop** branch on **AFPA38/testgithub**.




**Continuous integration has not been set up**
Several apps [are available](#) to automatically catch bugs and enforce style.

**This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request

 or view [command line instructions](#).




Write Preview

AA B i “ < > 🔗 ☰ ☷ ✓ @ 📌 ↶

Leave a comment

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

 Styling with Markdown is supported

Close pull request Comment

Références



<https://git-scm.com/book/fr/v2/D%C3%A9marrage-rapide-%C3%80-propos-de-la-gestion-de-version>

<https://guides.github.com/activities/hello-world/>

<https://nvie.com/posts/a-successful-git-branching-model/>

<https://gist.github.com/aquelito/8596717>

Pour en savoir un peu plus :

<http://igm.univ-mlv.fr/~dr/XPOSE2010/gestiondeversiondecentralisee/dvcs-dvcs.html>