



BINV2190-B

DevOps : les bases

Git les bases

Sébastien de Beaufort – Olivier Choquet



Sommaire

- Introduction Git
- Commandes de base
- Conflits
- .gitignore
- Installation
- Démo



Git - Sondage

- Utilisation de Git dans le projet Web au Q2 l'année académique passée ?
 - <https://www.wooclap.com/DEVOPSS1>

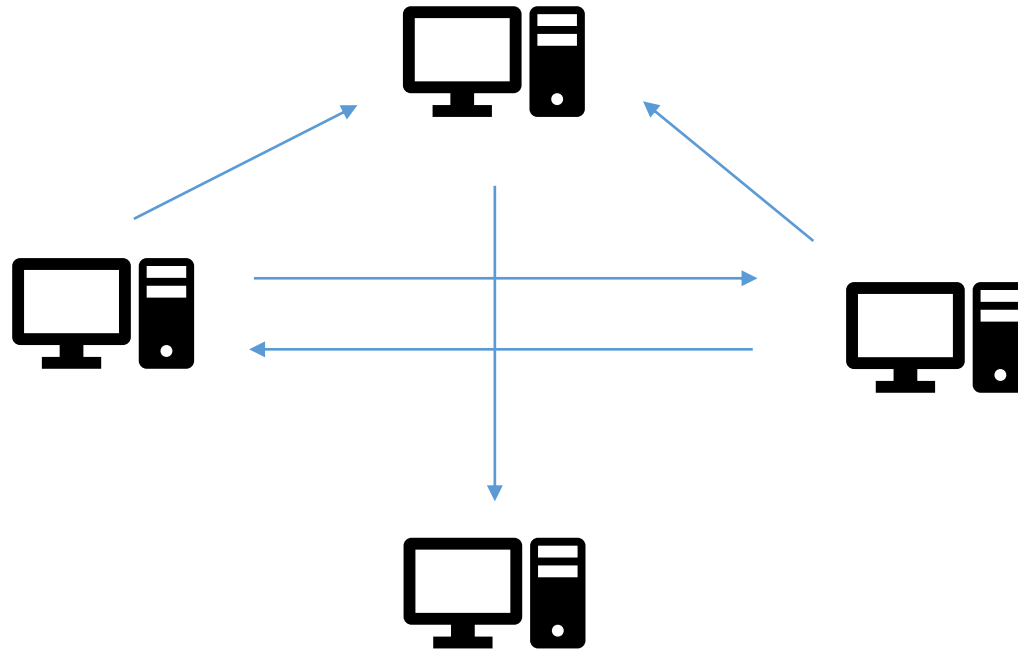


Introduction Git

- Git est un système de contrôle de version distribué
 - gratuit
 - open source
 - conçu pour tout gérer, des petits aux très grands projets, avec rapidité et efficacité.
- Git permet notamment de :
 - partager facilement du code entre plusieurs développeurs
 - intégrer les changements de plusieurs développeurs sur un même projet



Git - système décentralisé



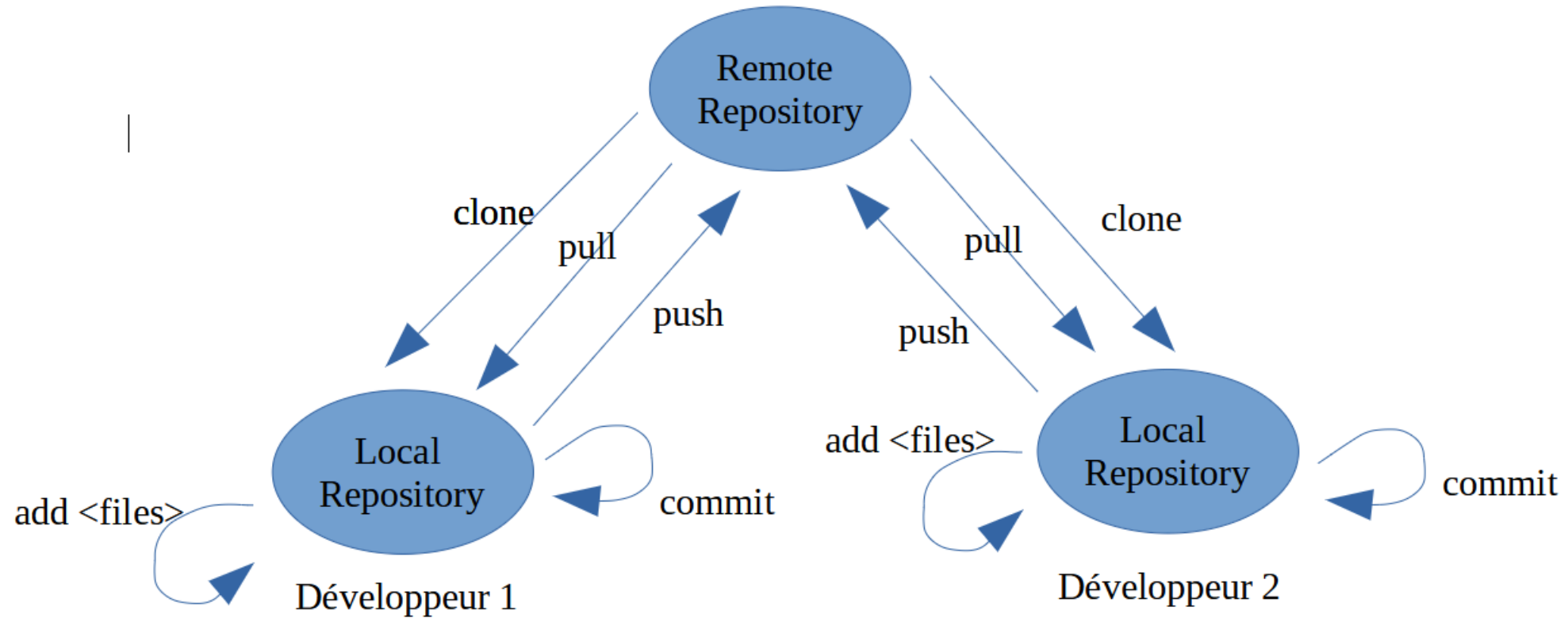


Dans la pratique

- Un dépôt central (Remote repository) créé par projet sur GitHub, GitLab,
- Un dépôt local (Local Repository) présent sur la machine de chaque développeur. Ce dépôt peut s'obtenir par clonage(**clone**) du dépôt distant
- Chaque développeur valide ses changements en local (**commit**) et envoie(**push**) ensuite ses changements sur le remote repository
- Chaque développeur peut récupérer les changements présents sur le remote repository (**pull**)



Git - Utilisation





Git – Commandes de base

- `git init`
 - création d'un dépôt (souvent en local)
- `git clone <url repository>`
 - cloner un dépôt distant
- `git pull`
 - récupérer les changements du remote repository en local
- `git push <name of remote repository> < name of branch>`
 - envoyer les changements validés (commits) sur le remote repository.
 - `git push` prend en paramètre le nom d'un repository(remote) et d'une branche.
 - Exemple : `git push origin main`



Git – Commandes de base

- `git add` : ajouter des fichiers à Git
 - A : all
- `git commit` : valider les changements dans un dépôt
 - a : ajouter les fichiers
 - m : message
- Le message du commit sera présent visible sur le repository. Il est vivement recommandé de structurer ce message
- La recommandation est d'utiliser un verbe d'action (init, add, delete, modify, refactor, ...) décrivant l'opération faite / la fonctionnalité
- Exemple : `git commit -a -m "init repository"`



Git – Commandes de base

- Git est un système de gestion de version
- On peut donc revenir à une version précédente, un commit précédent
- Il est donc vivement conseillé de valider (commit) souvent et des petits changements plutôt que d'envoyer des gros changements
- De gros changements ont également plus de chance de générer des conflits (voir slide suivant)



Git - Conflits

- Git fusionnera le code entre le dépôt local et le dépôt distant
- Cependant il peut rester des conflits c'est-à-dire des morceaux de code que Git n'arrive pas à fusionner
- Typiquement cela survient quand la même ligne d'un fichier a été modifiée différemment par 2 développeurs

```
<<<<<< HEAD
```

```
$message = "Coucou"; #Le code que vous avez écrit (Head)
```

```
=====
```

```
$message = "Bonjour"; #Le code qui vient du repo distant (master)
```

```
>>>>>> master
```



Git - Résolution des conflits

- 3 possibilités
 - Accepter les modifications du dépôt distant
 - Accepter les modifications du dépôt local
 - Modifier les fichiers en conflit, valider(commit) et envoyer(push)
- Souvent un outil de présentation des conflits Git est présent dans les IDE



.gitignore

- Git offre la possibilité d'ignorer des fichiers. Pq ?
 - Fichiers avec données confidentielles (mot de passe, ...)
 - Fichiers propres à l'IDE
 - Librairies
 - ...

- Exemple :

```
# ignorer le répertoire .idea qui contient les fichiers de PhpStorm
```

```
/.idea/
```

```
# ignorer le fichier thumbs.db
```

```
/thumbs.db
```

```
# ignorer tous les fichiers avec pour extension .txt
```

```
*.txt
```



.gitignore

- Important de penser au .gitignore au début du projet avant le premier commit !!!
- Le .gitignore (et les règles de chemins qu'il contient) s'applique par rapport à l'endroit dans l'arborescence où il se trouve
 - On le place donc le plus souvent à la racine du projet



Installation Git

- <https://git-scm.com/downloads>
- Ubuntu : `sudo apt install git`
- Mac OS : `brew install git`



Démo

- Création d'un dépôt Git sur GitHub
- Cloner le dépôt Git
- Création d'un projet PHP avec PhpStorm
 - JetBrains Account educative licence
- .gitignore
- Modification d'un fichier
- Envoi des modifications sur le dépôt distant
- Gestion d'un conflit



Exercice

- Créer un dépôt sur GitHub
- Ajouter dans ce dépôt le code du site des bonnes nouvelles
- Modifier le footer en y ajoutant votre nom



Devoirs pour séance prochaine

- Installer/Mettre à jour sur votre machine
 - WampServer
 - PhpStorm
 - Git
- Trouver un partenaire
 - Lien MooVin