Prise en main git

Qu’est ce que git:

Git est un projet open source de versioning très utilisé dans le monde du développement.
Il permet grâce aux « commits » d’avoir un historique des modifications apportées au projet à chaque étapes (on peut ainsi savoir qui a modifié, quand ça a été modifié, pourquoi ….).

Le contenu des fichiers, les liens entre les fichiers et les répertoires, les versions, les tags, les commits : tous ces éléments du dépôt Git sont sécurisés à l'aide d'un algorithme de hachage sécurisé cryptographiquement, appelé SHA1. Celui-ci protège le code et l'historique des changements contre toute modification accidentelle ou malveillante, tout en assurant une traçabilité complète de l’historique.

(Les commandes git sont a effectuées dans un terminal)

Principales commandes :

Git config

Commande pratique utilisée pour définir des valeurs de configuration Git au niveau global ou local d'un projet
Exemple:

```
git config --global user.email "votre_email@exemple.com"
```

Git init

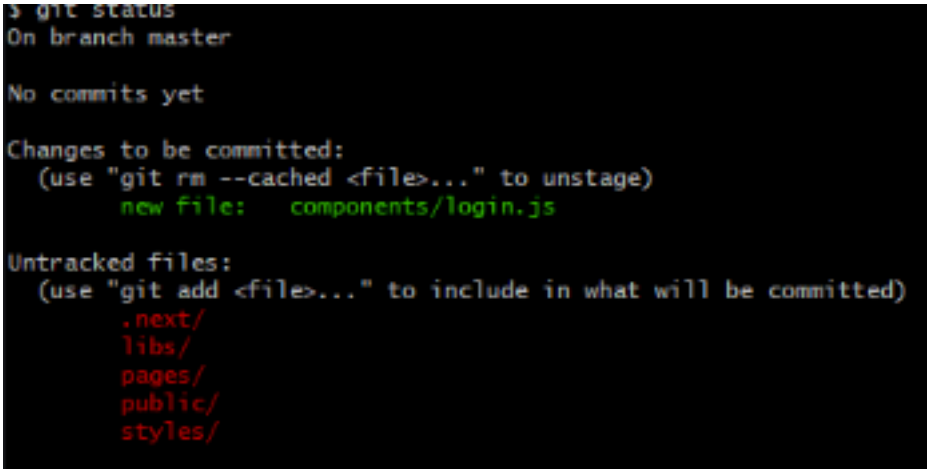
La commande git init crée un nouveau dépôt Git. Permet de convertir un projet existant, sans version en un dépôt Git ou d'initialiser un nouveau dépôt vide.

Git clone

Clone un dépôt dans un répertoire nouvellement créé.
Commande très utile quand vous souhaitez cloner un dépôt git en local.
Exemple:
Git clone /chemin/vers/dépôt

Git status

Renvoi le statut de l'arbre de travail.
Exemple:



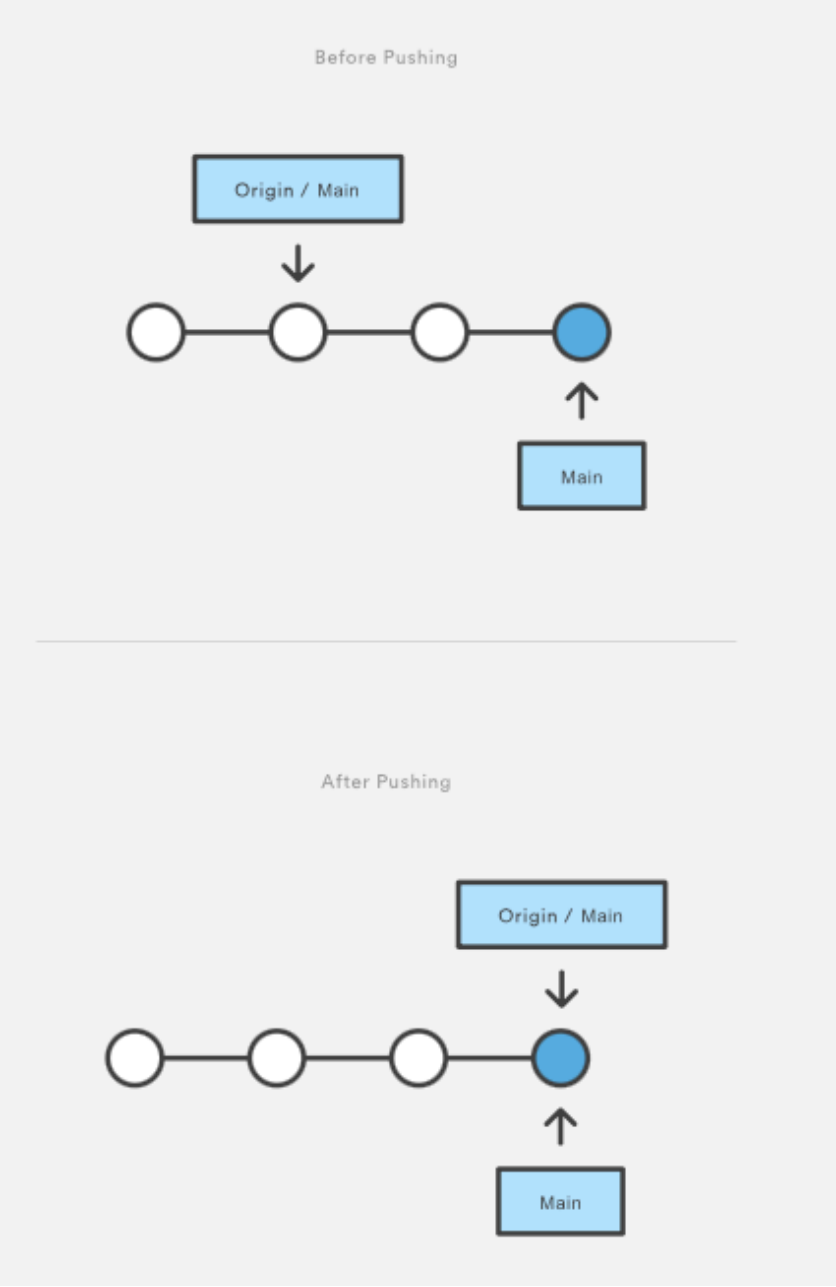
Git push

git push est la commande la plus couramment utilisée pour publier des changements locaux et les charger vers un dépôt centralisé. Après qu'un dépôt local a été modifié, un push est exécuté pour partager les changements avec les membres de l'équipe distants.

```
git push <remote> <branch>
```

Le schéma ci-dessous montre ce qui arrive lorsque votre branche main locale évolue par rapport à la branche main du dépôt centralisé et que vous publiez des changements en exécutant git push origin main.

Vous n'effectuerez généralement pas de push directement sur la branche main d'un projet (hormis si vous êtes lead dev du dit projet) mais sur votre branche de travail.

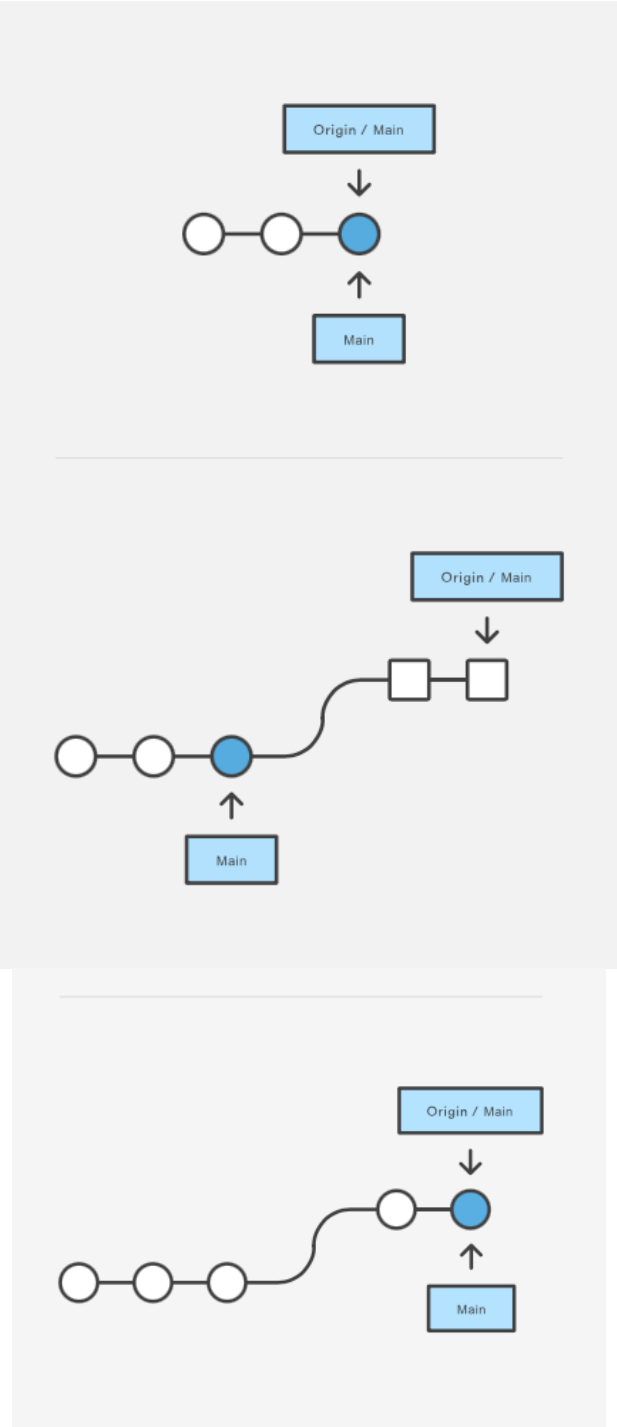


Git pull

La commande git pull est utilisée pour faire un fetch du contenu d'un dépôt distant et pour le télécharger, puis pour mettre à jour immédiatement le dépôt local qui correspond à ce contenu.

```
git pull <remote>
```

Cela permet de synchroniser facilement votre dépôt local avec les changements en amont. Le schéma suivant explique toutes les étapes du processus de pull.



Git checkout

La commande git checkout vous permet de basculer entre les branches créées au moyen de git branch. Le check-out d'une branche entraîne une mise à jour des fichiers contenus dans le répertoire de travail, qui s'alignent sur la version stockée dans cette branche. En outre, Git reçoit l'ordre d'enregistrer tous les nouveaux commits sur cette branche. Considérez cette opération comme une manière de sélectionner la ligne de développement sur laquelle vous travaillez.

```
git checkout <branchname>
```

De plus, la commande git checkout accepte un argument -b qui agit comme une méthode pratique pour créer la branche et basculer vers elle immédiatement.

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

Git branch

Permet de répertorier toutes les branches de votre dépôt. Cette commande est synonyme de git branch —list.

Git add

Elle informe Git que vous voulez inclure les mises à jour dans un fichier du prochain commit.
Exemple:
Git add . //Inclut tous les fichiers du répertoire dans le prochain commit (hormis ceux spécifiés dans un fichier .gitignore)

Git commit

Les commits constituent les piliers d'une chronologie de projet Git. Les commits peuvent être considérés comme des instantanés ou des étapes importantes dans la chronologie d'un projet Git. Ils sont créés grâce à la commande git commit pour capturer l'état d'un projet à un point dans le temps.

Git merge

La commande git merge vous permet de sélectionner les lignes de développement indépendantes créées avec git branch et de les intégrer à une seule branche.

Sources:
https://www.atlassian.com
https://git-scm.com/docs