

Architecture web

Git

Qu'est-ce que git ?

- Un logiciel de gestion de sources
 - Suivi des changements d'un fichier/dossier ou d'un ensemble de fichiers/dossiers
 - Navigation dans l'historique des modifications
 - Partage des changements
- Décentralisé
 - Pas besoin d'un serveur
- Et distribué
 - Multi-utilisateurs

D'où vient git ?

- Git a été inventé en 2005 par Linus Torvalds.
- Il succède à de nombreux outils similaires comme cvs ou subversion.
- git signifie « sale pourri » ou « connard » en argot britannique

Pourquoi ?

- Quelques exemples
 - Suivre les étapes de modification d'un programme
 - Tester un changement complexe et pouvoir revenir facilement en arrière
 - Travailler à plusieurs sur un projet
 - Inviter des collaborateurs sur un projet
 - Contribuer à un projet OpenSource

Les notions clés ?

- Repository
 - Un repository ou dépôt git correspond à l'ensemble de l'historique de votre projet.
- Changeset / Commit
 - Un changeset ou commit est une collection de changement d'un ensemble de fichiers dans un dépôt
- Revision
 - Une révision correspond à un changeset donné dans un dépôt.
- Version
 - Souvent on parle d'une « revision » pour une « file revision » qui désigne une certaine version d'un fichier

Comment utiliser git ?

- git est un outil fonctionnant en ligne de commande
- La commande s'appelle git
- Elle est utilisable depuis un terminal ou une console (shell)

Installer git ?

- GIT Source Code Management
 - <https://git-scm.com/>
 - <https://gitforwindows.org>

Comment utiliser git ?

- Le programme git permet de lancer des commandes pour gérer votre dépôt git
 - git <commande> <arguments>

Premiers pas avec git

- Définir son identité
- Votre identité sera associée aux modifications que vous enregistrerez dans vos dépôts
- Elle est défini dans le fichier ~/.gitconfig ou %USERPROFILE%\ .gitconfig
 - \$ git config --global user.name "Votre nom"
 - \$ git config --global user.email votre@mail

Créer votre premier dépôt

- Placez-vous dans votre dossier personnel et lancer la commande git init burgers
 - \$ git init burgers
 - Initialized empty Git repository in ~/burgers/.git/
 - \$ ls -a burgers
 -git

Dossier de travail vs dépôt

- Dossier de travail
 - Votre dossier de travail contient vos fichiers et dossiers composant votre projet.
 - C'est votre cuisine !
 - Git peut modifier ces fichiers pour les mettre à jour ou vous les présenter à différentes versions du projet grâce à votre dépôt.
- Dépôt Git
 - Votre dépôt git contient tout l'historique de votre projet.
 - Toutes les versions des fichiers, toutes les modifications, etc.
 - C'est le dossier .git dans votre dossier de travail.

git init <dossier de travail>

- Permet de créer un dépôt git.
 - Si le dossier de travail indiqué n'existe pas, elle va le créer.
- Sans paramètre, la commande crée un dépôt git pour le dossier courant.

Ajouter un fichier à son dépôt

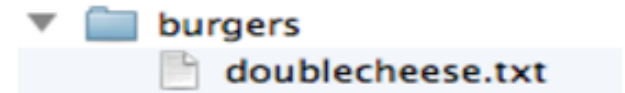
- Créons un burger
 - Créez le fichier doublecheese.txt dans le dossier burgers
 - Notez-y la liste des ingrédients pour faire un double cheese.
- steak
 - tomates
 - cheese
 - oignons
 - cornichons
 - ketchup
 - moutarde

Ajouter un fichier à son dépôt

Mon dépôt git



Mon dossier de travail



Ajouter un fichier à son dépôt

- Indexer un fichier dans votre dépôt git
- Placez vous dans votre dossier burgers
 - \$ cd burgers
- Indexez le fichier doublecheese.txt dans votre dépôt avec la commande git add doublecheese.txt
 - \$ git add doublecheese.txt

git add <fichier(s) ou dossier(s)>

- Permet de marquer un ou plusieurs fichiers/dossiers comme devant être ajoutés au dépôt git.
- Attention : les fichiers/dossiers sont juste marqués et ne sont pas encore ajoutés

Ajouter un fichier à son dépôt

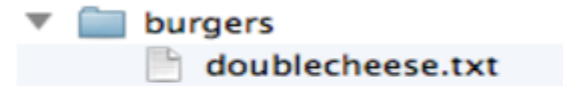
- Enregistrer les changements
- Validez cette modification afin de l'enregistrer dans votre dépôt avec la commande git commit
 - `$ git commit doublecheese.txt -m "Création du doublecheese"`
 - `[master (root-commit) c17b141] Création du doublecheese`
 - 1 file changed, 7 insertions(+)
 - create mode 100644 doublecheese.txt

Ajouter un fichier à son dépôt

Mon dépôt git



Mon projet



Git commit

- `git commit <fichier(s) ou dossier(s)> -m "commentaire"`
- Permet d'enregistrer les modifications concernant les fichiers ou dossiers passés en paramètre sur votre dépôt (en créant une nouvelle révision)
- `git commit -a` permet d'enregistrer toutes modifications en cours même non indexées

Modifier un fichier

- Il n'y a pas de tomates dans le doublecheese !
- Modifiez le fichier doublecheese.txt

Un nouveau burger

- Ajoutez le fichier bigmac.txt à votre projet burgers
- steak
- salade
- tomates
- oignons
- cornichons
- ketchup
- moutarde

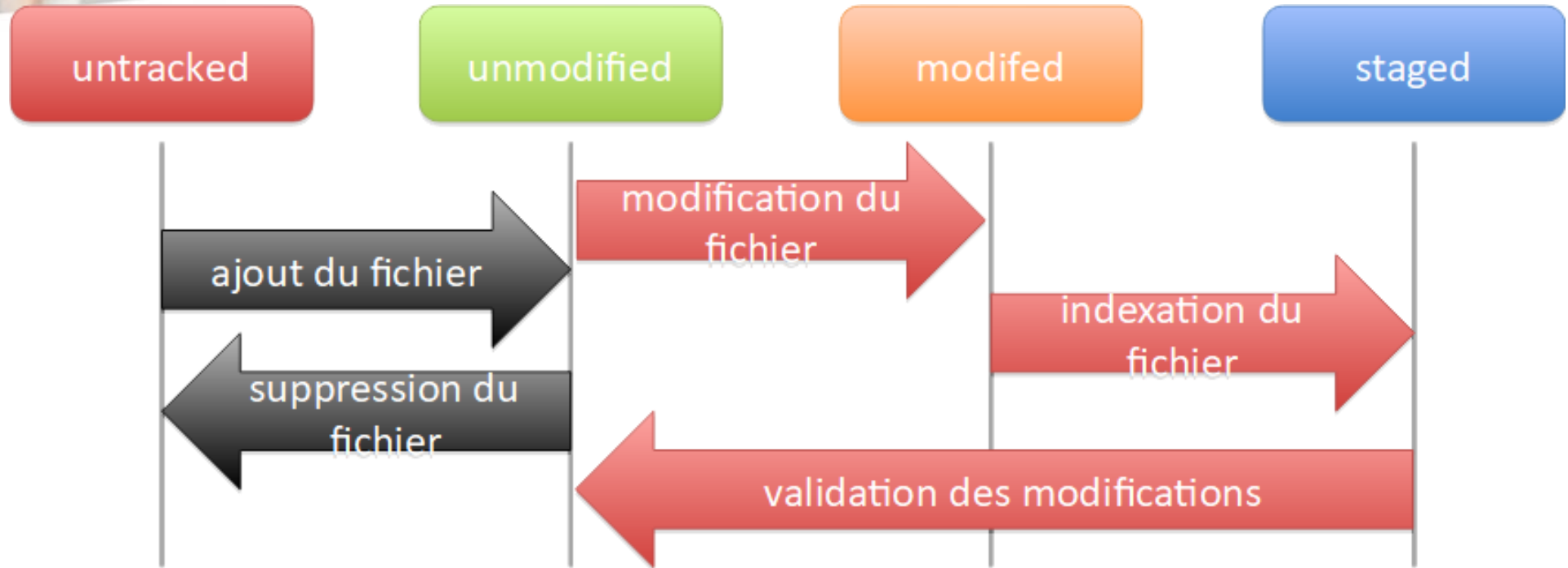
Vérifier l'état de son projet

- Où en sommes-nous ?
- Vérifiez ce qui a changé dans votre projet avec la commande `git status`
 - `$ git status`
 - On branch master
 - Changes not staged for commit:
 - (use "`git add <file>...`" to update what will be committed)
 - (use "`git checkout -- <file>...`" to discard changes in working directory)
 - modified: doublecheese.txt
 - Untracked files:
 - (use "`git add <file>...`" to include in what will be committed)
 - bigmac.txt
 - no changes added to commit (use "`git add`" and/or "`git commit -a`")

git status

- Permet de lister les modifications en cours (non enregistrées dans votre dépôt) dans votre dossier de travail.
- On retrouve 4 états possibles pour chaque fichier :
 - untracked : fichier non suivi
 - unmodified : fichier non modifié
 - modified : fichier modifié
 - staged : modification du fichier indexée

Les statuts des fichiers dans un dépôt git



Indexer les modifications

- Indexer la modification sur le fichier doublecheese.txt avec la commande git add
 - \$ git add doublecheese.txt

Vérifier l'état de son projet

- Où en sommes-nous ?
- Vérifiez ce qui a changé dans votre projet avec la commande `git status`
 - `$ git status`
 - On branch master
 - Changes to be committed:
 - (use "`git reset HEAD <file>...`" to unstage)
 - modified: doublecheese.txt
 - Untracked files:
 - (use "`git add <file>...`" to include in what will be committed)
 - bigmac.txt

Ajouter un nouveau fichier au dépôt

- Ajout le nouveau fichier bigmac.txt avec la commande git add
 - \$ git add bigmac.txt

Vérifier l'état de son projet

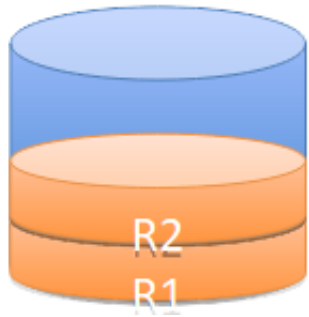
- Où en sommes-nous ?
- Vérifiez ce qui a changé dans votre projet avec la commande git status
 - \$ git status
 - On branch master
 - Changes to be committed:
 - (use "git reset HEAD <file>..." to unstage)
 - new file: bigmac.txt
 - modified: doublecheese.txt

Valider les modifications

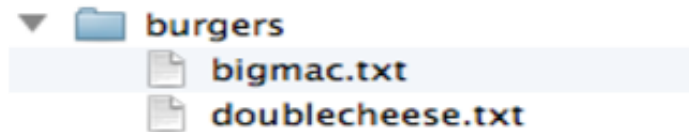
- Lancez la commande git commit
 - \$ git commit -m "Ajout du bigmac et correction du double cheese"
 - [master afb67b0] Ajout du bigmac et correction du double cheese
 - 2 files changed, 7 insertions(+), 1 deletion(-)
 - create mode 100644 bigmac.txt

Premiers pas avec git

Mon dépôt git



Mon projet



Introduisons une erreur

- Modifiez le fichier bigmac.txt afin d'y introduire un mauvais ingrédient.

Version précédente d'un fichier

- Restaurer la dernier version valide du fichier à l'aide de la commande git checkout
 - \$ git checkout bigmac.txt

git checkout <fichier ou dossier>

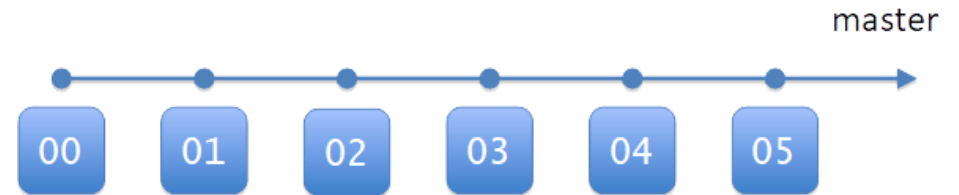
- Permet de restaurer un fichier ou un dossier à sa dernière version tel qu'il est enregistré dans votre dépôt.
- Il est possible d'indiquer un numéro de révision pour restaurer le fichier ou le dossier à une version antérieur.

Les révisions

- Vous avez dit numéro de révision ?
- Chaque commit sur votre dépôt crée une nouvelle révision ou version de votre projet.
- Chaque révision correspond en fait à un ensemble de changement que vous avez « commité »

Les révisions

- Chaque commit crée une nouvelle revision.



Consulter les révisions de votre dépôt local

- Utilisez la commande git log pour consulter les révisions de votre dépôt local
 - \$ git log
 - commit afb67b0177d49235f43737ee72cc76bccd58e5b0
 - Author: toto tutu <toto.tutu@gmail.com>
 - Date: Tue Apr 7 15:56:07 2019 +0200
 - Ajout du bigmac et correction du double cheese
 - commit c17b1418035b6ae2ad2d38b3b089df27130eea9c
 - Author: toto tutu <toto.tutu@gmail.com>
 - Date: Tue Apr 7 15:40:01 2019 +0200
 - Création du doublecheese

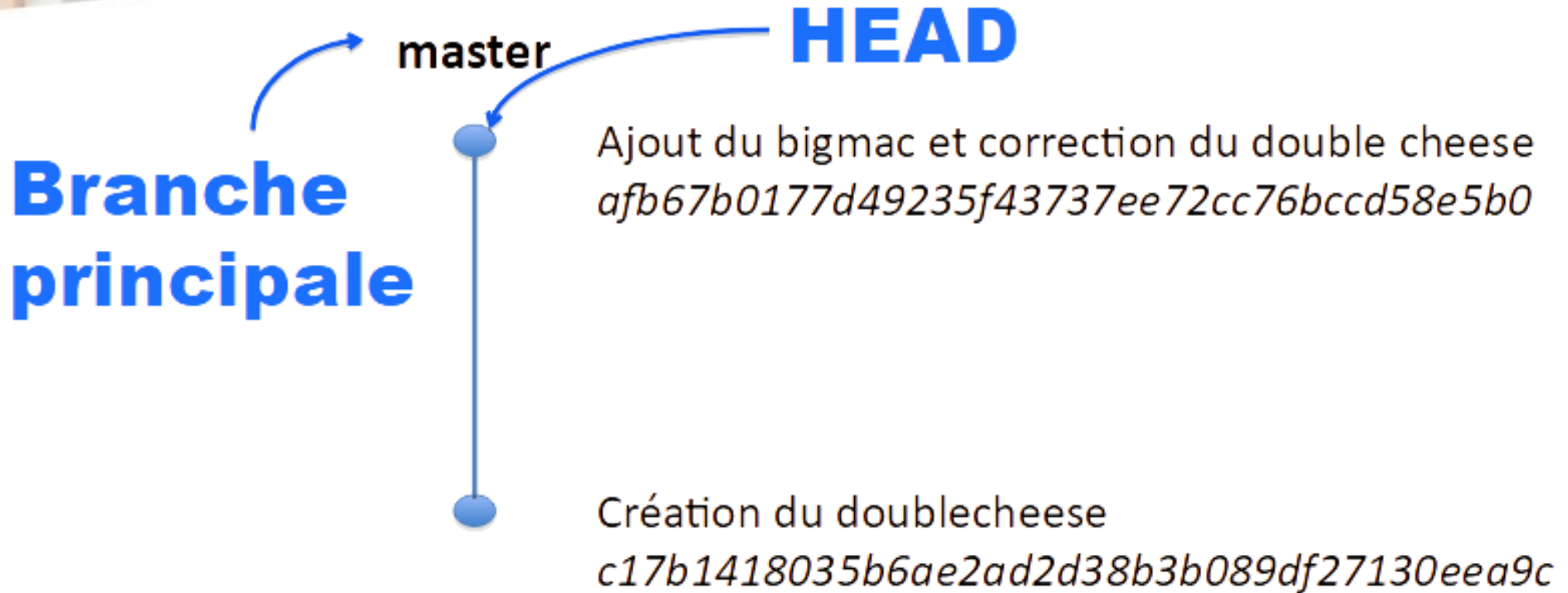
git log

- permet de consulter l'ensemble des révisions stockées dans votre dépôt local
- Pour chaque révision, on peut consulter les informations suivantes :
 - commit : numéro de révision
 - Author : utilisateur ayant enregistré la révision
 - Date : date de la création de la révision
 - Commentaire inscrit par l'utilisateur lors du commit

Révision précédente de votre projet

- Revenez à la version initiale de votre projet (juste le doublecheese.txt) en utilisant la commande git checkout <id rev>
 - \$ git checkout
c17b1418035b6ae2ad2d38b3b089df27130eea9c
 - \$ ls
 - doublecheese.txt

Introduction à la notion de branche



Introduction à la notion de branche

master



**DETACHED
HEAD**

Création du doublecheese

`c17b1418035b6ae2ad2d38b3b089df27130eea9c`

Introduction à la notion de branche

master



dev

**On peut démarrer
une nouvelle
branche**

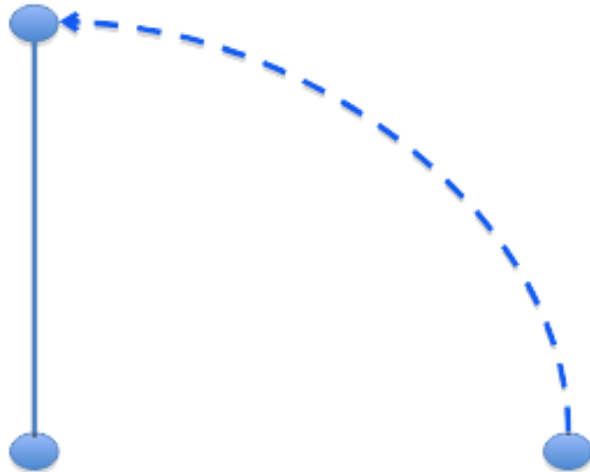


Création du doublecheese

c17b1418035b6ae2ad2d38b3b089df27130eea9c

Introduction à la notion de branche

master



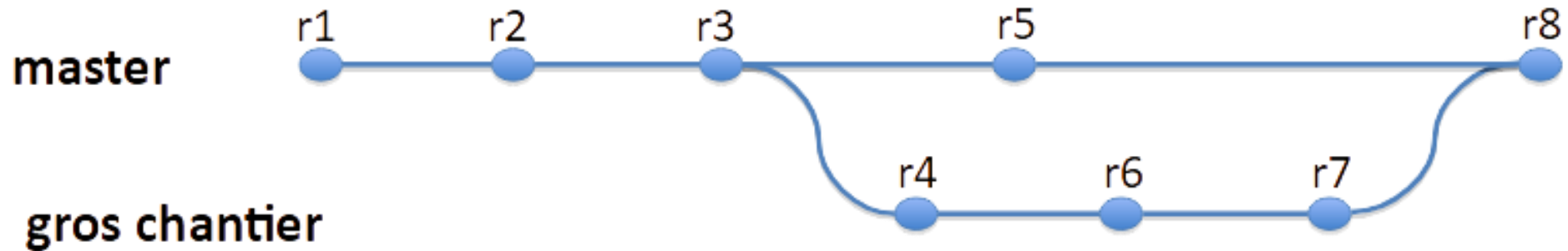
**On peut revenir à la
version courante
git checkout master**

Création du doublecheese
c17b1418035b6ae2ad2d38b3b089df27130eea9c

Revenir à la dernière version du projet

- Revenez à la version initiale de votre projet
 - \$ git checkout master
 - \$ ls
 - bigmac.txt doublecheese.txt

Aller plus loin avec les branches



Aller plus loin avec les branches

- Notre fast-food veut passer au BIO.
- Il faut changer toutes nos recettes !!!
- C'est un « gros chantier »...



Créer une nouvelle branche

- Créer une nouvelle branche bio sur votre dépôt
 - \$ git branch bio

Créer une nouvelle branche

- Consultez les branches disponibles sur votre dépôt
 - \$ git branch
 - bio
 - * master

git branch

- Permet de gérer les branches de votre dépôt

Changer de branche

- Utilisez la commande git checkout pour changer de branche.
 - \$ git checkout bio
 - Switched to branch 'bio'

Aller plus loin avec les branches

master



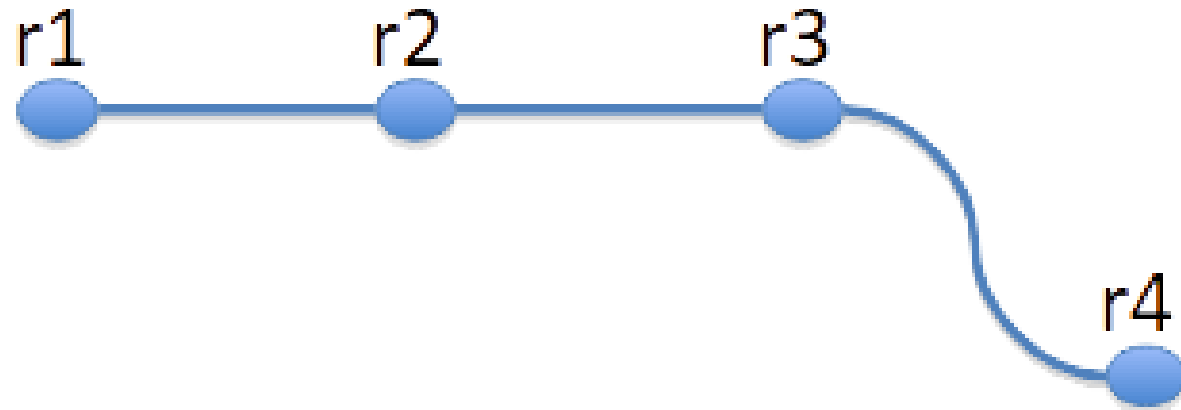
bio

On passe au bio !

- Modifiez la recette du Big Mac et commité la modification
 - \$ git commit -a -m "Bio Mac"
- steak bio
- salade bio
- tomates bio
- oignons bio
- cornichons bio
- ketchup bio
- moutarde bio

Aller plus loin avec les branches

master



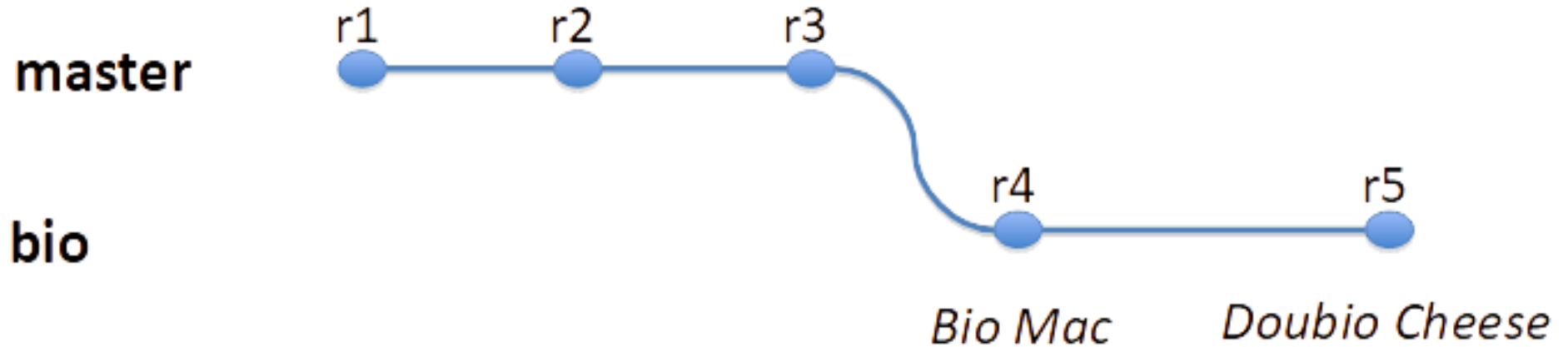
bio

Bio Mac

On passe au bio !

- Modifiez la recette du Double Cheese et commité la modification
 - \$ git commit -a -m « Doubio Cheese"
- steak bio
- cheese bio
- oignons bio
- cornichons bio
- ketchup bio
- moutarde bio

Aller plus loin avec les branches



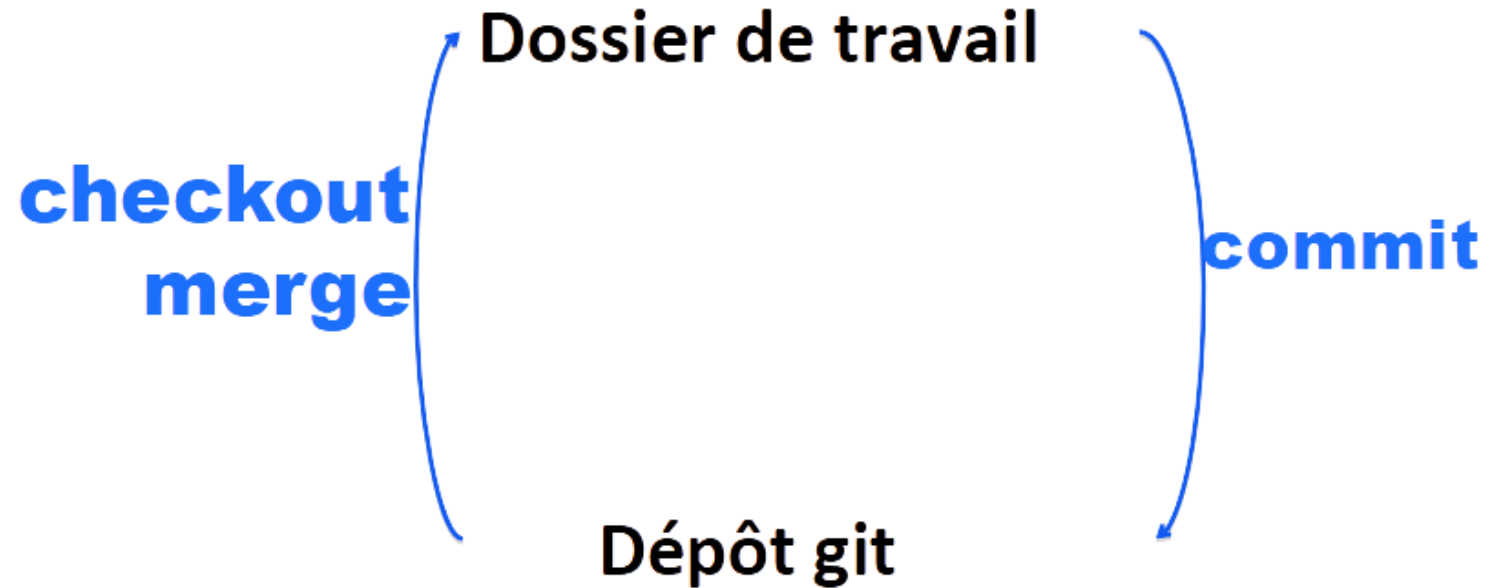
Revenons sur notre branche principale

- Revenez sur la branche principale avec la commande git checkout
 - \$ git checkout master

Fusion des branches

- On peut récupérer les modifications enregistrées sur la branche bio avec la commande git merge
 - \$ git merge bio
 - Updating 92afcc5..71c36de
 - Fast-forward
 - bigmac.txt | 14 ++++++-----
 - doublecheese.txt | 12 ++++++-----
 - 2 files changed, 13 insertions(+), 13 deletions(-)

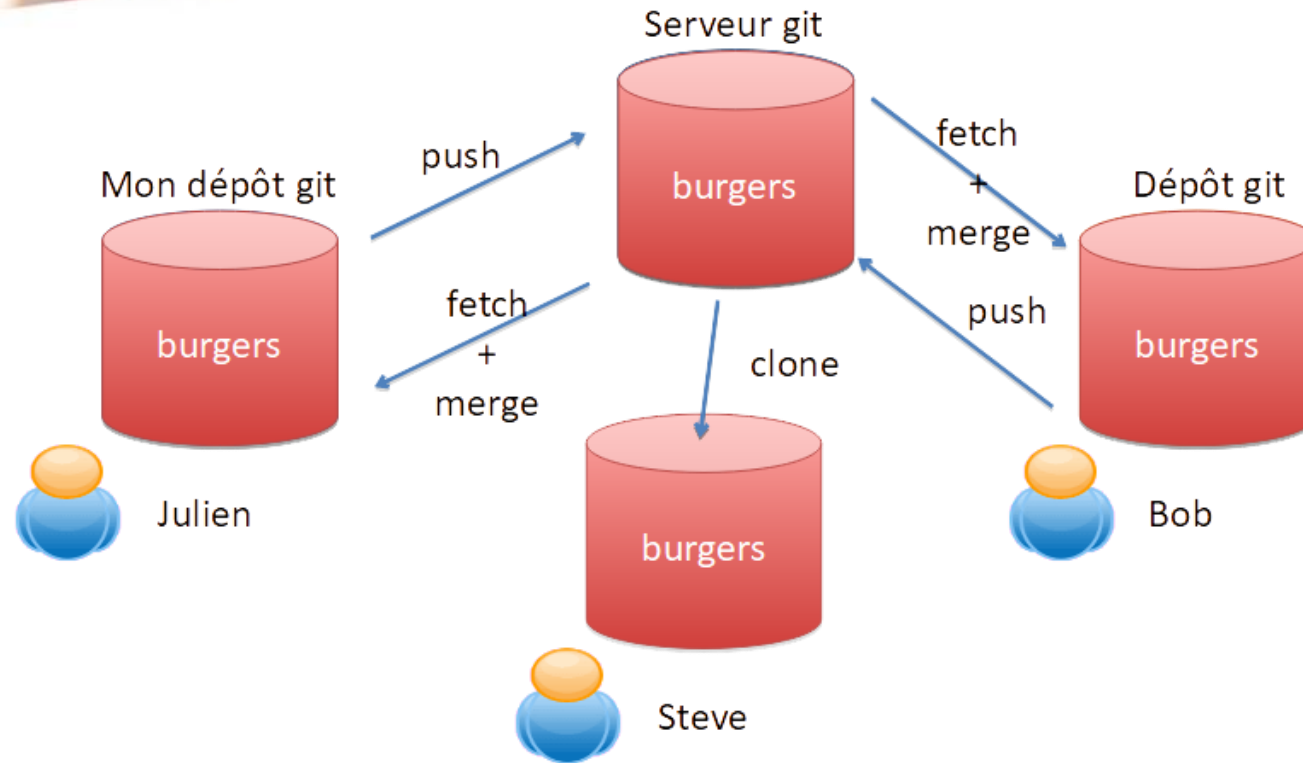
Le cycle git



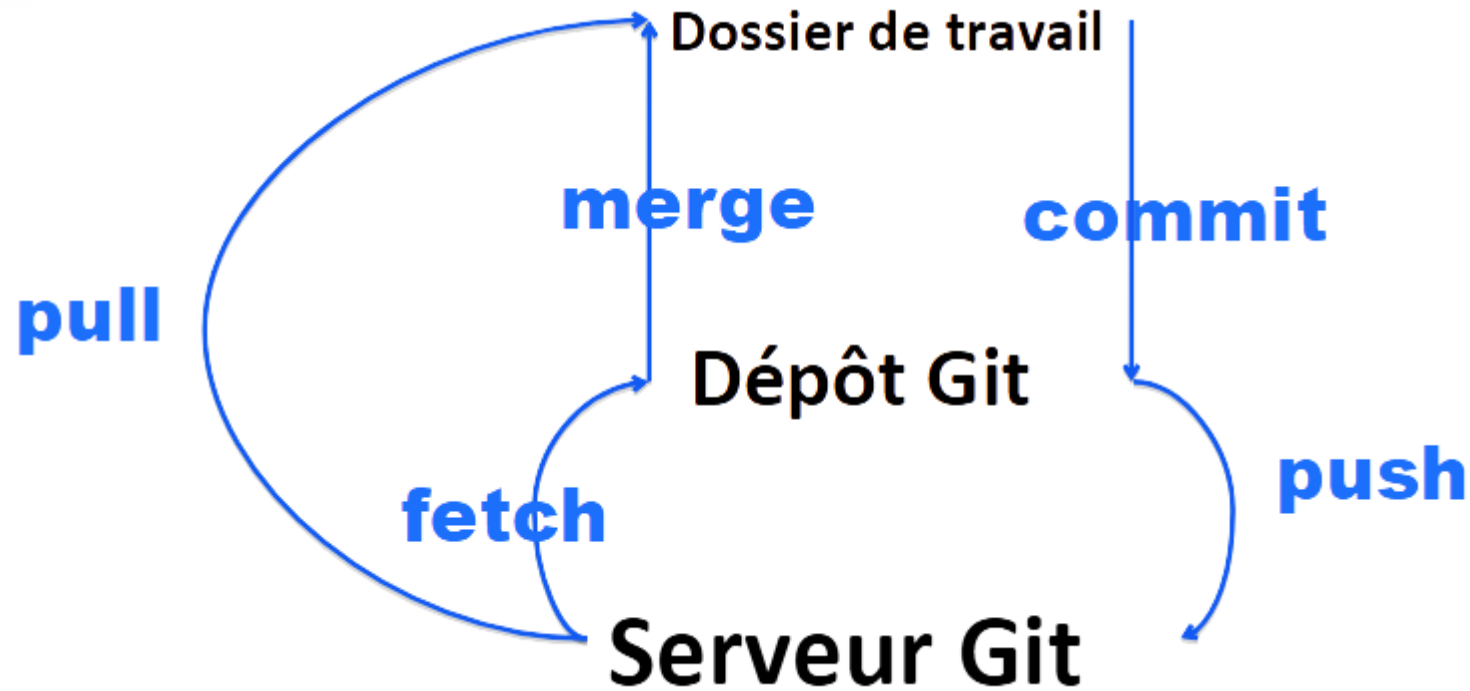
Les commandes git pour gérer votre dépôt local

Commande git	Description
init	Création d'un dépôt pour un projet/dossier
add	Indexation d'une modification ou de l'ajout d'un fichier ou dossier
rm	Suppression d'un fichier ou dossier
mv	Déplacement d'un fichier ou dossier
status	Visualisation de l'état du dépôt
diff	Visualisation des modifications entre deux révisions ou entre une révision et la version courante
checkout	Récupération d'un fichier à partir du dépôt
log	Consultation de la liste des validations (commits) enregistrées sur le dépôt

Travailler à plusieurs



Le cycle git



Travailler à plusieurs

- Déclarer serveur distant
- Déclarez le serveur gitlab dans votre dépôt local avec la commande git remote add
 - \$ git remote add origin
<http://xxxxxxxxxx/yyyyyyyyyy/burgers-tutu.git>

Envoyer les révisions de votre dépôt sur le serveur

- `git push -u origin master`
- Vous pouvez également « pusher » votre branche

```
$ git push -u origin master
Counting objects: 12, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (12/12), 1.14 KiB | 0 bytes/s, done.
Total 12 (delta 0), reused 0 (delta 0)
To http://xxxxxxxxx/yyyyyyyyyy/burgers-tutu.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

git push <remote> <branch>

- Permet de copier les changeset de votre dépôt local sur le dépôt d'un serveur distant

Cloner un dépôt

- Créer une copie locale du dépôt de votre voisin
- Placez-vous dans un dossier en dehors de votre dépôt et utilisez la commande
- `git clone <adresse du dépôt de votre voisin>`
 - `$ cd /tmp`
 - `$ git clone http://xxxxxxxxxxxxx/yyyyyyyyyyy/burgers-toto.git`
 - Cloning into 'burgers-toto'...
 - remote: Counting objects: 12, done.
 - remote: Compressing objects: 100% (9/9), done.
 - remote: Total 12 (delta 0), reused 0 (delta 0)
 - Unpacking objects: 100% (12/12), done.
 - Checking connectivity... done.

git clone

- Permet de créer une copie locale d'un dépôt du serveur sur votre poste de travail.
- Le dossier de travail est positionner automatiquement sur la dernière révision du dépôt.

Travailler à plusieurs

- Créer un nouveau burger sur le dépôt de votre voisin et mettez à jour le serveur
 - \$ cd burgers-slegras
 - \$ git add smallburger.txt
 - \$ git commit -m "Ajout d'un petit burger"
 - \$ git push

Travailler à plusieurs

- Mettez à jour votre dépôt local pour récupérer les modifications apportés par votre voisin
 - \$ cd ../burgers
 - \$ git pull

git pull

- Permet de récupérer les nouveaux changements enregistrés sur le dépôt du serveur pour les intégrer dans votre dépôt local.
- Votre dossier de travail est automatiquement mis à jour

Les commandes git pour collaborer

Commande git	Description
clone	Créer un dépôt local et un dossier de travail à partir d'un dépôt distant
fetch	Récupération les modifications d'un dépôt distant
merge	Fusionner les modifications du dépôt avec les modifications de votre dossier de travail
push	Envoyer vos modifications validés vers un dépôt distant
pull	fetch + merge

Serveur distant github.com

- Github est un serveur Git accessible à tous depuis une simple connexion Internet.
- Il est gratuit pour les projets public (open source) et payant pour les projets privés.

Git dans le développement web

- Transformer le répertoire de votre site web en un repository Git connecté à un serveur distant (remote) tel que GitHub ou Bitbucket !
- Voici ce que vous pouvez maintenant en faire

Git dans le développement web

- Déployer votre site web sur un autre serveur.
 - Il suffit de cloner le dépôt avec la commande git clone.
- Développer avec l'esprit tranquille.
 - Créer facilement une version de votre site web avec « commit » et procéder à vos modifications.
 - En cas de soucis, il faudra juste retourner au dernier « commit » avec un rapide « checkout »
 - C'est nettement plus pratique que d'effectuer régulièrement des sauvegardes physiques du répertoire entier.

Git dans le développement web

- Envoyer vos modifications locales sur votre site en ligne (hébergé sur un serveur dédié).
 - Vous avez travaillé en local et vous souhaitez maintenant mettre à jour votre site en ligne avec vos derniers développements.

GIT dans le développement web

Sans GIT

- Soit reporter fichier par fichier vos modifications, soit envoyer l'intégralité de votre site local vers votre site en ligne.
 - La première méthode est lente et fastidieuse,
 - La seconde vous fera perdre les derniers ajouts de votre site en ligne.

Avec GIT

- Déployer automatiquement des « patches » contenant les dernières lignes de code modifiées.
- Vous mettez à jour
 - facilement votre site internet avec vos récents développements
 - intelligemment, sans écraser les fichiers existants.

Git dans le développement web

- Travailler en équipe sur le même site internet.
 - Si vous travaillez à plusieurs sur le même site internet, vous allez devoir déployer les modifications de chaque développeur sans que celles-ci s'écrasent mutuellement.
 - Sans système de versionning, c'est une étape risquée et lente, qui demande une grande concertation de la part de toute l'équipe.

Git dans le développement web

- Avec Git, c'est beaucoup plus simple.
 - À la condition que les développeurs n'aient pas modifié les mêmes lignes des mêmes fichiers, les modifications seront toutes incorporées.
 - Et en cas de conflit, Git indiquera les lignes posant problème pour facilement le régler.

GIT dans le développement web

- Synchroniser deux serveurs hébergeant deux versions d'un site.
- Un site web hébergé sur deux serveur
 - Un serveur de dev pour les développeurs
 - Un serveur de production ouvert au public