

Git un système de gestion de versions

ALBERT SHIH

Direction Informatique
Observatoire de Paris

13 octobre 2017



Gestionnaire de version

- Wikipedia : *La gestion de versions (en anglais version control ou revision control) consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers (généralement en texte).*

Gestionnaire de version

- Wikipedia : *La gestion de versions (en anglais version control ou revision control) consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers (généralement en texte).*
- Gestion des versions de fichiers.

Gestionnaire de version

- Wikipedia : *La gestion de versions (en anglais version control ou revision control) consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers (généralement en texte).*
- Gestion des versions de fichiers.
- Revenir en arrière.

Gestionnaire de version

- Wikipedia : *La gestion de versions (en anglais version control ou revision control) consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers (généralement en texte).*
- Gestion des versions de fichiers.
- Revenir en arrière.
- Méta-données, date, auteurs, historiques des fichiers, etc.

Gestionnaire de version

- Wikipedia : *La gestion de versions (en anglais version control ou revision control) consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers (généralement en texte).*
- Gestion des versions de fichiers.
- Revenir en arrière.
- Méta-données, date, auteurs, historiques des fichiers, etc.
- SCCS (1972), RCS (1982).

Gestionnaire de version

- Wikipedia : *La gestion de versions (en anglais version control ou revision control) consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers (généralement en texte).*
- Gestion des versions de fichiers.
- Revenir en arrière.
- Méta-données, date, auteurs, historiques des fichiers, etc.
- SCCS (1972), RCS (1982).
- CVS (1990), subversion (2000).

Gestionnaire de version

- Wikipedia : *La gestion de versions (en anglais version control ou revision control) consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers (généralement en texte).*
- Gestion des versions de fichiers.
- Revenir en arrière.
- Méta-données, date, auteurs, historiques des fichiers, etc.
- SCCS (1972), RCS (1982).
- CVS (1990), subversion (2000).
- git (2005), mercurial (2005).

Centralisé



Centralisé

Serveur

Utilisateur 1

Centralisé

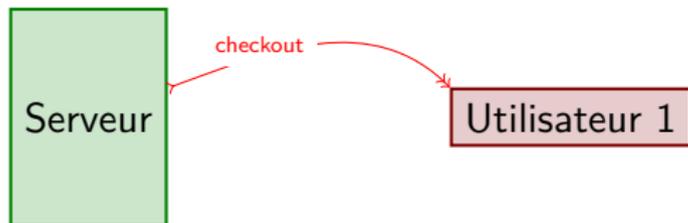
Serveur

Utilisateur 1

Utilisateur 2

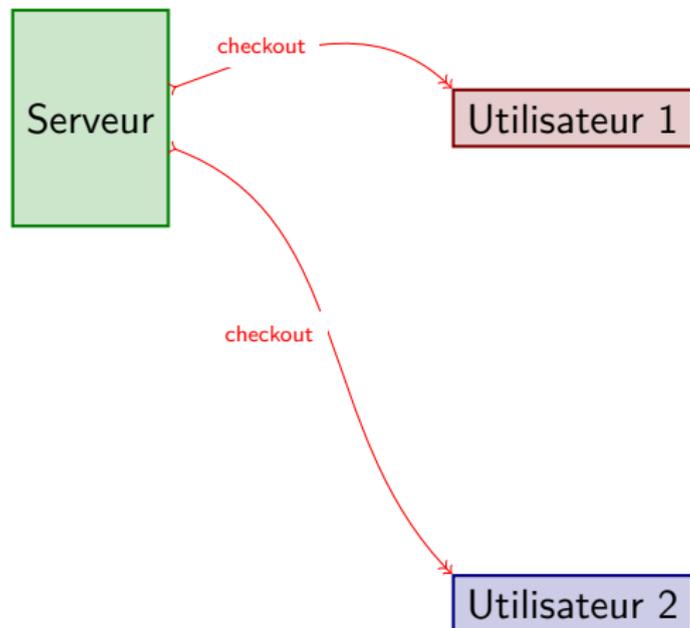
The diagram illustrates a centralized system. On the left, a light green square represents the 'Serveur' (Server). To its right, a red-bordered rectangle represents 'Utilisateur 1' (User 1). Below 'Utilisateur 1', a blue-bordered rectangle represents 'Utilisateur 2' (User 2). There are no lines or arrows connecting the server to the users, suggesting a simple, non-distributed architecture.

Centralisé

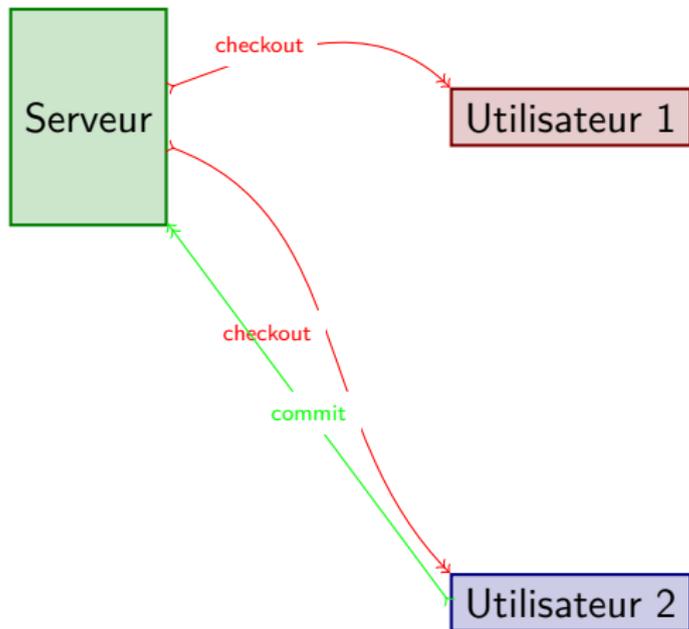


Utilisateur 2

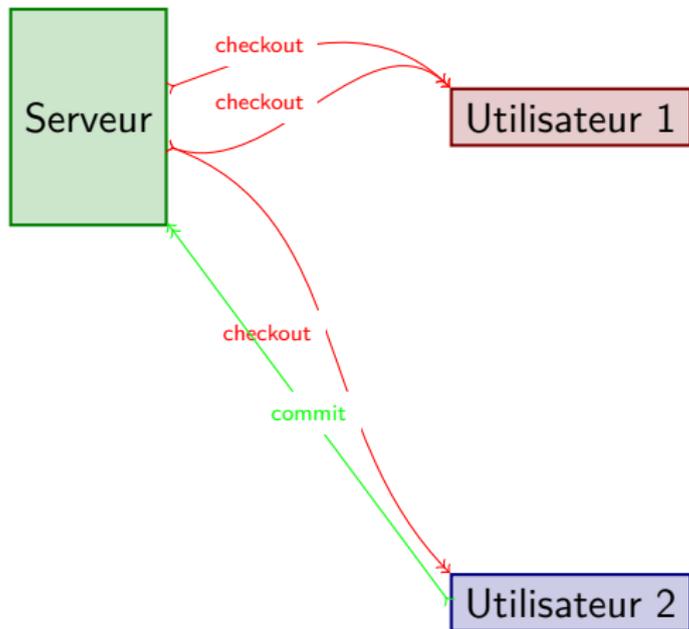
Centralisé



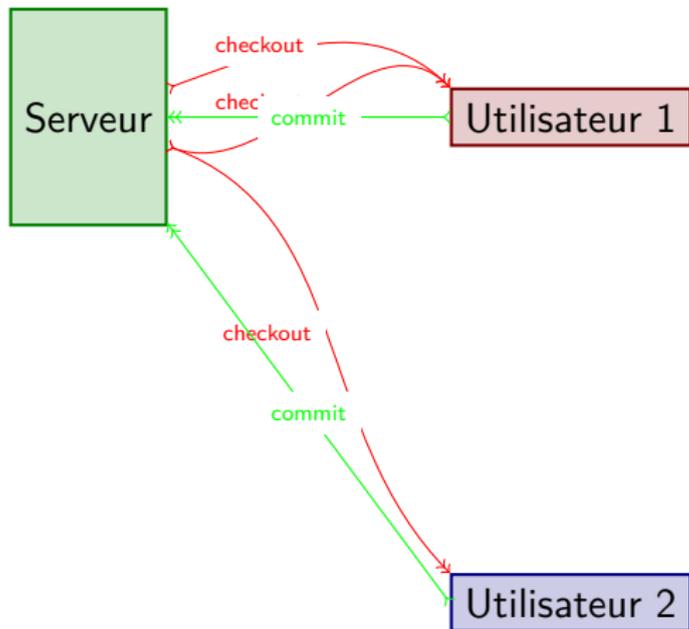
Centralisé



Centralisé



Centralisé



Décentralisé

- Avantages
 - Compréhension simple
- Désavantages
 - Gestion de large projet.
 - Gestion de nombreux contributeurs.
 - Historique propres et cohérent.

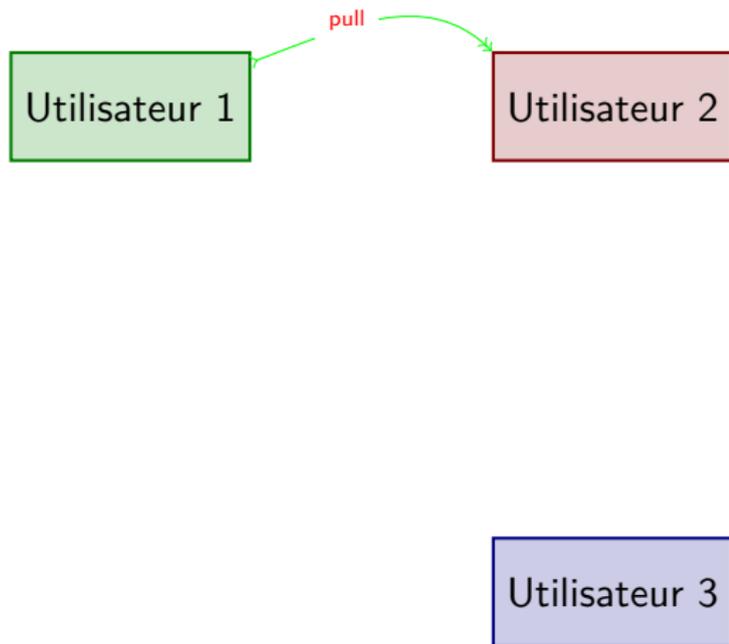
Décentralisé

Utilisateur 1

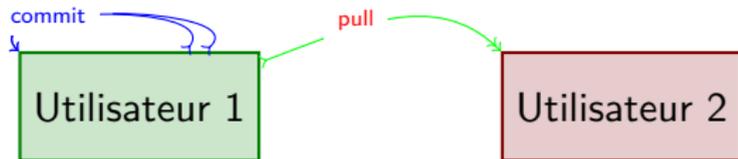
Utilisateur 2

Utilisateur 3

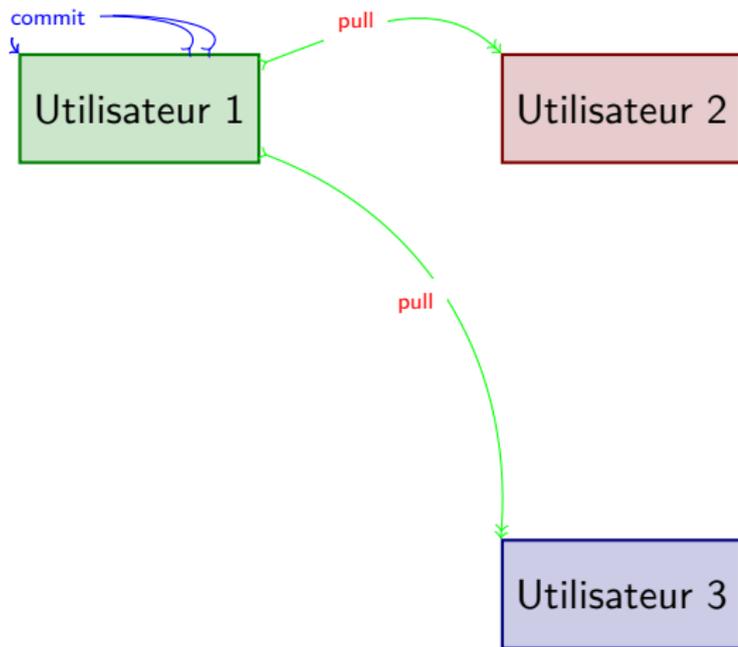
Décentralisé



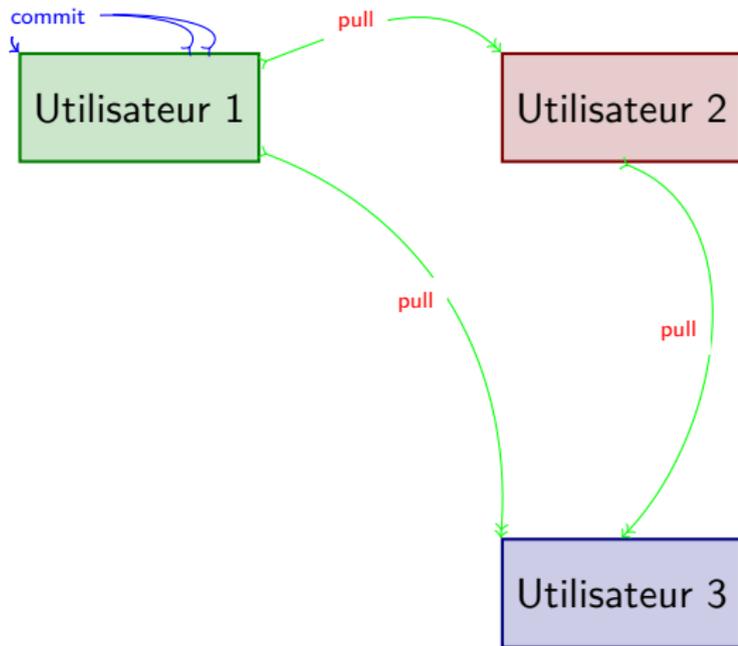
Décentralisé



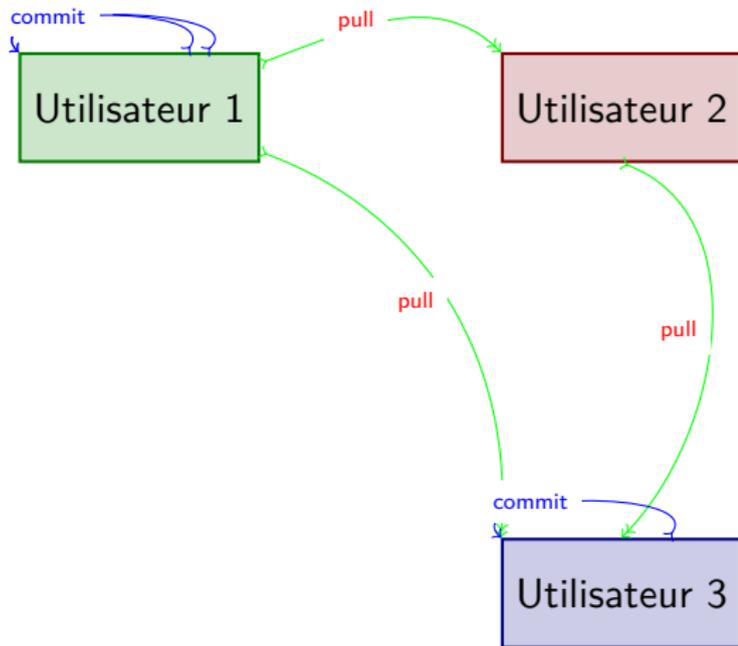
Décentralisé



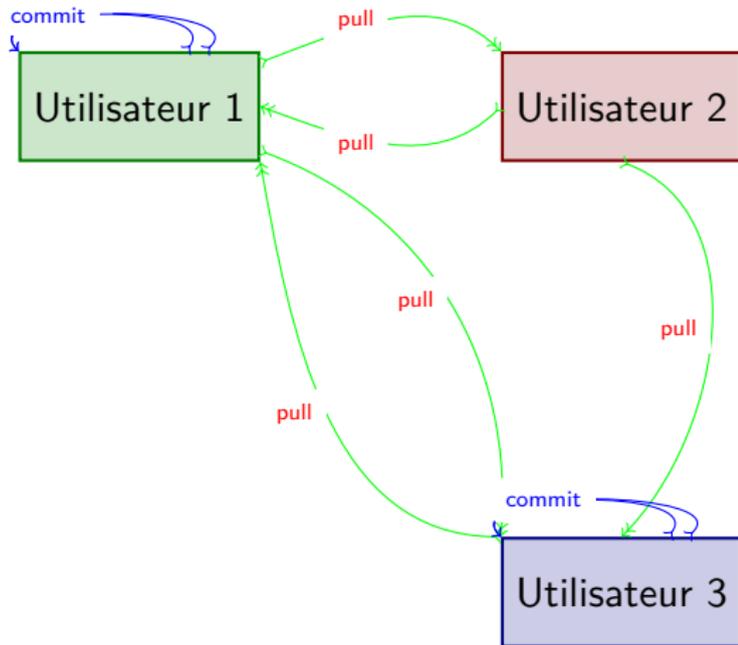
Décentralisé



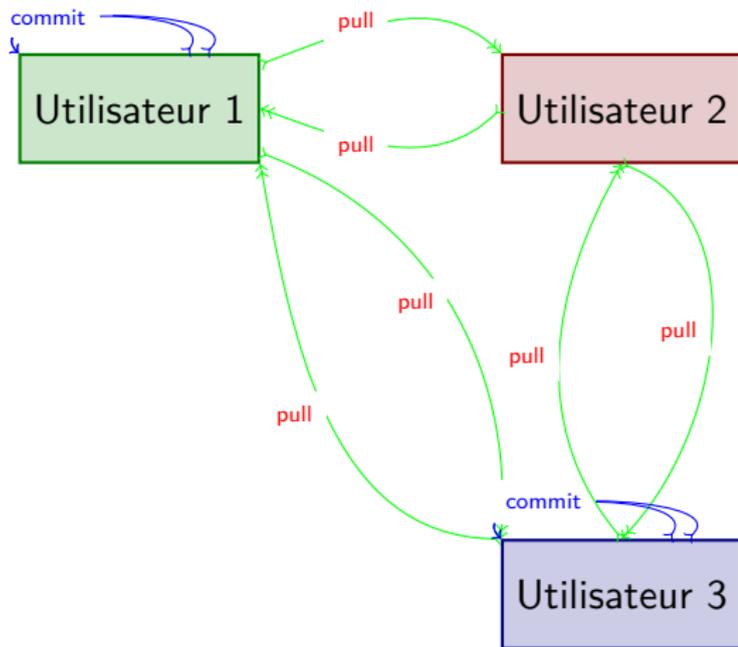
Décentralisé



Décentralisé



Décentralisé



Décentralisé

- Avantages
 - Gestion de large projet.

Décentralisé

- Avantages

- Gestion de large projet.
- Gestion de nombreux contributeur.

Décentralisé

- Avantages

- Gestion de large projet.
- Gestion de nombreux contributeur.
- Historique propres et cohérent.

Décentralisé

- Avantages

- Gestion de large projet.
- Gestion de nombreux contributeur.
- Historique propres et cohérent.
- Efficacité.

Décentralisé

- Avantages

- Gestion de large projet.
- Gestion de nombreux contributeur.
- Historique propres et cohérent.
- Efficacité.
- Indépendance au réseau.

Décentralisé

- Avantages

- Gestion de large projet.
- Gestion de nombreux contributeur.
- Historique propres et cohérent.
- Efficacité.
- Indépendance au réseau.

- Désavantages

Décentralisé

- Avantages

- Gestion de large projet.
- Gestion de nombreux contributeur.
- Historique propres et cohérent.
- Efficacité.
- Indépendance au réseau.

- Désavantages

- Plus complexe.

Décentralisé

- Avantages

- Gestion de large projet.
- Gestion de nombreux contributeur.
- Historique propres et cohérent.
- Efficacité.
- Indépendance au réseau.

- Désavantages

- Plus complexe.
- Consommation espace disque en local.

Git

- Système de gestion de versions.

Git

- Système de gestion de versions.
- Lancé par Linus Torvalds.

Git

- Système de gestion de versions.
- Lancé par Linus Torvalds.
- Largement utilisé, GitHub.

Git

- Système de gestion de versions.
- Lancé par Linus Torvalds.
- Largement utilisé, GitHub.
- Gestion de version centralisé ou dé-centralisé.

Git

- Système de gestion de versions.
- Lancé par Linus Torvalds.
- Largement utilisé, GitHub.
- Gestion de version centralisé ou dé-centralisé.
- git \neq subversion.

Git

- Système de gestion de versions.
- Lancé par Linus Torvalds.
- Largement utilisé, GitHub.
- Gestion de version centralisé ou dé-centralisé.
- git \neq subversion.
- La doc `man git-XXXX` par exemple `man git-add`.

Git

- Système de gestion de versions.
- Lancé par Linus Torvalds.
- Largement utilisé, GitHub.
- Gestion de version centralisé ou dé-centralisé.
- `git` \neq subversion.
- La doc `man git-XXXX` par exemple `man git-add`.
- `git` est bavard.

Git

- Système de gestion de versions.
- Lancé par Linus Torvalds.
- Largement utilisé, GitHub.
- Gestion de version centralisé ou dé-centralisé.
- git \neq subversion.
- La doc `man git-XXXX` par exemple `man git-add`.
- git est bavard.
- Intégration dans le shell (bash/zsh).

Git

- Système de gestion de versions.
- Lancé par Linus Torvalds.
- Largement utilisé, GitHub.
- Gestion de version centralisé ou dé-centralisé.
- git \neq subversion.
- La doc `man git-XXXX` par exemple `man git-add`.
- git est bavard.
- Intégration dans le shell (bash/zsh).
- Intégration dans emacs/vi/eclipse/etc. . . .

Git

- Système de gestion de versions.
- Lancé par Linus Torvalds.
- Largement utilisé, GitHub.
- Gestion de version centralisé ou dé-centralisé.
- `git` \neq subversion.
- La doc `man git-XXXX` par exemple `man git-add`.
- `git` est bavard.
- Intégration dans le shell (`bash/zsh`).
- Intégration dans `emacs/vi/eclipse/etc...`
- Un dépôt `git` \rightarrow `.git`.

Git

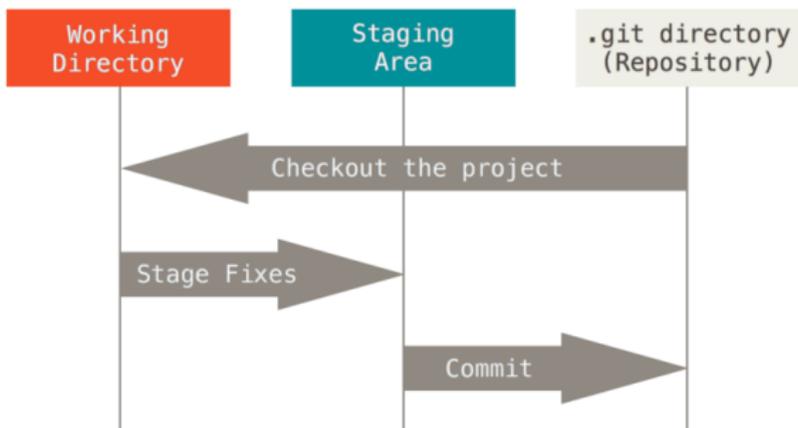
- Système de gestion de versions.
- Lancé par Linus Torvalds.
- Largement utilisé, GitHub.
- Gestion de version centralisé ou dé-centralisé.
- `git` \neq subversion.
- La doc `man git-XXXX` par exemple `man git-add`.
- `git` est bavard.
- Intégration dans le shell (`bash/zsh`).
- Intégration dans `emacs/vi/eclipse/etc...`
- Un dépôt `git` \rightarrow `.git`.
- Beaucoup de documentation.

Git

- Système de gestion de versions.
- Lancé par Linus Torvalds.
- Largement utilisé, GitHub.
- Gestion de version centralisé ou dé-centralisé.
- `git` \neq subversion.
- La doc `man git-XXXX` par exemple `man git-add`.
- `git` est bavard.
- Intégration dans le shell (`bash/zsh`).
- Intégration dans `emacs/vi/eclipse/etc...`
- Un dépôt `git` \rightarrow `.git`.
- Beaucoup de documentation.
- *Truc de geek.*

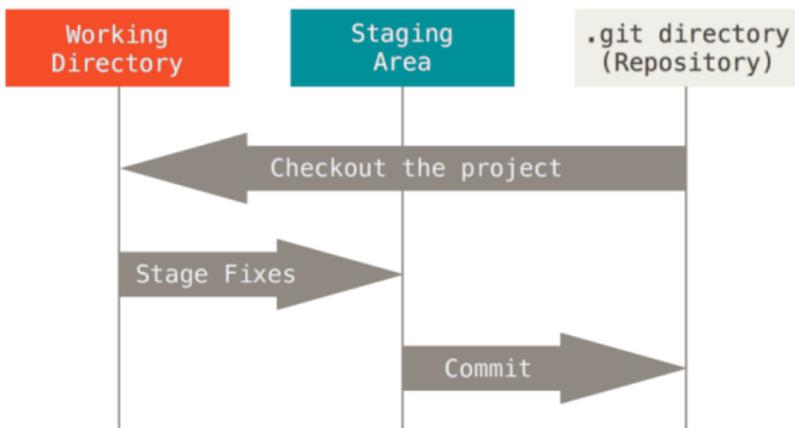
Bases

- Trois états possible pour un fichiers.



Bases

- Trois états possible pour un fichiers.



- Les révision sont des sha-1.

git init

Start your engine..

- `export $MES_REPOS='CEQUEVOUSVOULEZ'`

git init

Start your engine..

- `export $MES_REPOS='CEQUEVOUSVOULEZ'`
- `mkdir $MES_REPOS`



git init

Start your engine..

- `export $MES_REPOS='CEQUEVOUSVOULEZ'`
- `mkdir $MES_REPOS`
- `cd $MES_REPOS`

git init

Start your engine..

- `export $MES_REPOS='CEQUEVOUSVOULEZ'`
- `mkdir $MES_REPOS`
- `cd $MES_REPOS`
- `git clone https://github.com/git/git.git > /dev/null 2>/dev/null &`

Start your engine..

- `export $MES_REPOS='CEQUEVOUSVOULEZ'`
- `mkdir $MES_REPOS`
- `cd $MES_REPOS`
- `git clone https://github.com/git/git.git > /dev/null 2>/dev/null &`
- `mkdir repos_1`

Start your engine..

- `export $MES_REPOS='CEQUEVOUSVOULEZ'`
- `mkdir $MES_REPOS`
- `cd $MES_REPOS`
- `git clone https://github.com/git/git.git > /dev/null 2>/dev/null &`
- `mkdir repos_1`
- `cd repos_1`

Start your engine..

- `export $MES_REPOS='CEQUEVOUSVOULEZ'`
- `mkdir $MES_REPOS`
- `cd $MES_REPOS`
- `git clone https://github.com/git/git.git > /dev/null 2>/dev/null &`
- `mkdir repos_1`
- `cd repos_1`
- `git init`

Start your engine..

- `export $MES_REPOS='CEQUEVOUSVOULEZ'`
- `mkdir $MES_REPOS`
- `cd $MES_REPOS`
- `git clone https://github.com/git/git.git > /dev/null 2>/dev/null &`
- `mkdir repos_1`
- `cd repos_1`
- `git init`
- `ls -al`

Start your engine..

- `export $MES_REPOS='CEQUEVOUSVOULEZ'`
- `mkdir $MES_REPOS`
- `cd $MES_REPOS`
- `git clone https://github.com/git/git.git > /dev/null 2>/dev/null &`
- `mkdir repos_1`
- `cd repos_1`
- `git init`
- `ls -al`
- `git --help`

Configuration

- Une série de configuration global ou local.

Configuration

- Une série de configuration global ou local.
 - `git config --global --list`

Configuration

- Une série de configuration global ou local.
 - `git config --global --list`
 - `git config --global --add user.name "VOTRE NOM"`
 - `git config --global --add user.email VOTRE_EMAIL`

Configuration

- Une série de configuration global ou local.
 - `git config --global --list`
 - `git config --global --add user.name "VOTRE NOM"`
 - `git config --global --add user.email VOTRE_EMAIL`
 - `git config --global --add color.ui auto`
 - `git config --global --add core.editor
/usr/local/bin/vim`

Configuration

- Une série de configuration global ou local.
 - `git config --global --list`
 - `git config --global --add user.name "VOTRE NOM"`
 - `git config --global --add user.email VOTRE_EMAIL`
 - `git config --global --add color.ui auto`
 - `git config --global --add core.editor
/usr/local/bin/vim`
 - `git config --global --list`

Configuration

- Une série de configuration global ou local.
 - `git config --global --list`
 - `git config --global --add user.name "VOTRE NOM"`
 - `git config --global --add user.email VOTRE_EMAIL`
 - `git config --global --add color.ui auto`
 - `git config --global --add core.editor
/usr/local/bin/vim`
 - `git config --global --list`
- Si besoin changer `git config --global --replace-all`

Configuration

- Une série de configuration global ou local.
 - `git config --global --list`
 - `git config --global --add user.name "VOTRE NOM"`
 - `git config --global --add user.email VOTRE_EMAIL`
 - `git config --global --add color.ui auto`
 - `git config --global --add core.editor
/usr/local/bin/vim`
 - `git config --global --list`
- Si besoin changer `git config --global --replace-all`
- Possibilité `git config --local`

Configuration

- Une série de configuration global ou local.
 - `git config --global --list`
 - `git config --global --add user.name "VOTRE NOM"`
 - `git config --global --add user.email VOTRE_EMAIL`
 - `git config --global --add color.ui auto`
 - `git config --global --add core.editor
/usr/local/bin/vim`
 - `git config --global --list`
- Si besoin changer `git config --global --replace-all`
- Possibilité `git config --local`
- Possibilité de faire `vi .git/config`

Ajout fichiers

- `cd $MES_REPOS/repos_1`
- Faites un `cp /etc/[a-e]* .` ou `cp` de ce que vous voulez (≥ 2)

Ajout fichiers

- `cd $MES_REPOS/repos_1`
- Faites un `cp /etc/[a-e]* .` ou `cp` de ce que vous voulez (≥ 2)
- `git status`

Ajout fichiers

- `cd $MES_REPOS/repos_1`
- Faites un `cp /etc/[a-e]* .` ou `cp` de ce que vous voulez (≥ 2)
- `git status`
- `git add *`

Ajout fichiers

- `cd $MES_REPOS/repos_1`
- Faites un `cp /etc/[a-e]*` . ou `cp` de ce que vous voulez (≥ 2)
- `git status`
- `git add *`
- `git commit`

Ajout fichiers

- `cd $MES_REPOS/repos_1`
- Faites un `cp /etc/[a-e]* .` ou `cp` de ce que vous voulez (≥ 2)
- `git status`
- `git add *`
- `git commit`
- `vi/emacs unfichier`

Ajout fichiers

- `cd $MES_REPOS/repos_1`
- Faites un `cp /etc/[a-e]*` . ou `cp` de ce que vous voulez (≥ 2)
- `git status`
- `git add *`
- `git commit`
- `vi/emacs unfichier`
- `git status`

Ajout fichiers

- `cd $MES_REPOS/repos_1`
- Faites un `cp /etc/[a-e]*` . ou `cp` de ce que vous voulez (≥ 2)
- `git status`
- `git add *`
- `git commit`
- `vi/emacs unfichier`
- `git status`
- `git add unfichier`

Ajout fichiers

- `cd $MES_REPOS/repos_1`
- Faites un `cp /etc/[a-e]*` . ou `cp` de ce que vous voulez (≥ 2)
- `git status`
- `git add *`
- `git commit`
- `vi/emacs unfichier`
- `git status`
- `git add unfichier`
- `git commit`
Mettre un **VRAI** commentaire

Ajout fichiers

- `cd $MES_REPOS/repos_1`
- Faites un `cp /etc/[a-e]* .` ou `cp` de ce que vous voulez (≥ 2)
- `git status`
- `git add *`
- `git commit`
- `vi/emacs unfichier`
- `git status`
- `git add unfichier`
- `git commit`
Mettre un **VRAI** commentaire
- Faire plusieurs fois les `vi`, `git add`, `git commit`



add

Staged

- vi/emacs unfichier

add

Staged

- vi/emacs unfichier
- git status

add

Staged

- vi/emacs unfichier
- git status
- git add unfichier

add

Staged

- vi/emacs unfichier
- git status
- git add unfichier
- vi/emacs unfichier

add

Staged

- vi/emacs unfichier
- git status
- git add unfichier
- vi/emacs unfichier
- git status

add

Staged

- vi/emacs unfichier
- git status
- git add unfichier
- vi/emacs unfichier
- git status
- git commit

add

Staged

- vi/emacs unfichier
- git status
- git add unfichier
- vi/emacs unfichier
- git status
- git commit
- git status

add

Staged

- vi/emacs unfichier
- git status
- git add unfichier
- vi/emacs unfichier
- git status
- git commit
- git status
- git add unfichier

add

Staged

- vi/emacs unfichier
- git status
- git add unfichier
- vi/emacs unfichier
- git status
- git commit
- git status
- git add unfichier
- git commit

Ajout fichiers partiel

- `vi/emacs unfichier`
Faire ≥ 2 groupes de modifications

Ajout fichiers partiel

- `vi/emacs unfichier`
Faire ≥ 2 groupes de modifications
- `git status`

Ajout fichiers partiel

- `vi/emacs unfichier`
Faire ≥ 2 groupes de modifications
- `git status`
- `git add -p unfichier`

Ajout fichiers partiel

- `vi/emacs unfichier`
Faire ≥ 2 groupes de modifications
- `git status`
- `git add -p unfichier`
- `git commit`

Ajout fichiers partiel

- `vi/emacs unfichier`
Faire ≥ 2 groupes de modifications
- `git status`
- `git add -p unfichier`
- `git commit`
- `git status`

Ajout fichiers partiel

- `vi/emacs unfichier`
Faire ≥ 2 groupes de modifications
- `git status`
- `git add -p unfichier`
- `git commit`
- `git status`
- `git add -p unfichier`

Ajout fichiers partiel

- `vi/emacs unfichier`
Faire ≥ 2 groupes de modifications
- `git status`
- `git add -p unfichier`
- `git commit`
- `git status`
- `git add -p unfichier`
- `git commit`

Ajout fichiers partiel

- `vi/emacs unfichier`
Faire ≥ 2 groupes de modifications
- `git status`
- `git add -p unfichier`
- `git commit`
- `git status`
- `git add -p unfichier`
- `git commit`
- `git status`

amend

- vi/emacs unfichier

amend

- vi/emacs unfichier
- git add; git commit

amend

- vi/emacs unfichier
- git add; git commit
- git log

amend

- vi/emacs unfichier
- git add; git commit
- git log
- git commit -amend

amend

- vi/emacs unfichier
- git add; git commit
- git log
- git commit -amend
- git log

amend

- `vi/emacs unfichier`
- `git add; git commit`
- `git log`
- `git commit -amend`
- `git log`
- Pour faire plus `git rebase`

diff

Différence

- vi/emacs unfichier

Différence

- vi/emacs unfichier
- git diff

Différence

- vi/emacs unfichier
- git diff
- git add unfichier
- git diff --staged (ou git diff --cached)

Parcours logs

- `cd $MES_REPOS/git`
- `git log`

Parcours logs

- `cd $MES_REPOS/git`
- `git log`
- `git log --oneline`

Parcours logs

- `cd $MES_REPOS/git`
- `git log`
- `git log --oneline`
- `git log -p`
- `git log -p XXXX...XXXX`

Parcours logs

- `cd $MES_REPOS/git`
- `git log`
- `git log --oneline`
- `git log -p`
- `git log -p XXXX...XXXX`
- `git log --author="Hamano"`

Parcours logs

- `cd $MES_REPOS/git`
- `git log`
- `git log --oneline`
- `git log -p`
- `git log -p XXXX...XXXX`
- `git log --author="Hamano"`
- `git log --grep="forget"`

Parcours logs

- `cd $MES_REPOS/git`
- `git log`
- `git log --oneline`
- `git log -p`
- `git log -p XXXX...XXXX`
- `git log --author="Hamano"`
- `git log --grep="forget"`
- `git log -S"function"`

Parcours logs

- `cd $MES_REPOS/git`
- `git log`
- `git log --oneline`
- `git log -p`
- `git log -p XXXX...XXXX`
- `git log --author="Hamano"`
- `git log --grep="forget"`
- `git log -S"function"`
- `git diff XXXX XXXX`

Parcours logs

- `cd $MES_REPOS/git`
- `git log`
- `git log --oneline`
- `git log -p`
- `git log -p XXXX...XXXX`
- `git log --author="Hamano"`
- `git log --grep="forget"`
- `git log -S"function"`
- `git diff XXXX XXXX`
- `git diff XXXX XXXX -- FICHER`

Divers

- `cd $MES_REPOS/git`

Divers

- `cd $MES_REPOS/git`
- `git blame utf8.c`

Divers

- `cd $MES_REPOS/git`
- `git blame utf8.c`
- `tig` ou `gitk`

Divers

- `cd $MES_REPOS/git`
- `git blame utf8.c`
- `tig` ou `gitk`
- `more .gitignore`

Divers

- `cd $MES_REPOS/git`
- `git blame utf8.c`
- `tig` ou `gitk`
- `more .gitignore`
- `git checkout HEAD~`

Divers

- `cd $MES_REPOS/git`
- `git blame utf8.c`
- `tig` ou `gitk`
- `more .gitignore`
- `git checkout HEAD~`
- `git checkout master`

Voir en arrière

- `cd $MES_REPOS/repos_1`

Voir en arrière

- `cd $MES_REPOS/repos_1`
- `git log --oneline`

Voir en arrière

- `cd $MES_REPOS/repos_1`
- `git log --oneline`
- `git checkout XXXXX`

Voir en arrière

- `cd $MES_REPOS/repos_1`
- `git log --oneline`
- `git checkout XXXXX`
- `git checkout master`

Voir en arrière

- `cd $MES_REPOS/repos_1`
- `git log --oneline`
- `git checkout XXXXX`
- `git checkout master`
- `git checkout XXXXX FICHIER`

Voir en arrière

- `cd $MES_REPOS/repos_1`
- `git log --oneline`
- `git checkout XXXXX`
- `git checkout master`
- `git checkout XXXXX FICHER`
- `git reset HEAD -- FICHER`
- `git checkout -- FICHER`

revert

Revenir en arrière

- `cd $MES_REPOS/repos_1`

Revenir en arrière

- `cd $MES_REPOS/repos_1`
- `git log --oneline`

Revenir en arrière

- `cd $MES_REPOS/repos_1`
- `git log --oneline`
- Annuler **un** commit `git revert XXXXX`.

Revenir en arrière

- `cd $MES_REPOS/repos_1`
- `git log --oneline`
- Annuler **un** commit `git revert XXXXX`.
- Revenir à une version `git revert HEAD...XXXX`.

Revenir en arrière

- `cd $MES_REPOS/repos_1`
- `git log --oneline`
- Annuler **un** commit `git revert XXXXX`.
- Revenir à une version `git revert HEAD...XXXX`.
- Option `git revert -no-edit HEAD...XXXX` :

Revenir en arrière

- `cd $MES_REPOS/repos_1`
- `git log --oneline`
- Annuler **un** commit `git revert XXXXX`.
- Revenir à une version `git revert HEAD...XXXX`.
- Option `git revert -no-edit HEAD...XXXX` :
- `git log`

Revenir en arrière avec reset

- `cd $MES_REPOS/repos_1`

Revenir en arrière avec reset

- `cd $MES_REPOS/repos_1`
- `git log --oneline`

Revenir en arrière avec reset

- `cd $MES_REPOS/repos_1`
- `git log --oneline`
- `git reset XXXXX`

Revenir en arrière avec reset

- `cd $MES_REPOS/repos_1`
- `git log --oneline`
- `git reset XXXXX`
- `git log`

Revenir en arrière avec reset

- `cd $MES_REPOS/repos_1`
- `git log --oneline`
- `git reset XXXXX`
- `git log`
- Ne pas faire sur des données pusher.

Exemples

- Un code source.

Exemples

- Un code source.
- Vos fichiers $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ etc.

Exemples

- Un code source.
- Vos fichiers $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ etc.
- Tous type de fichiers textes.

Conseils

- Un repos par contexte.

Conseils

- Un repos par contexte.
- Commiter régulièrement.

Conseils

- Un repos par contexte.
- Commiter régulièrement.
- Mettre de vrais commentaires (éviter `git commit -m`).

Conseils

- Un repos par contexte.
- Commiter régulièrement.
- Mettre de vrais commentaires (éviter `git commit -m`).
- Commiter régulièrement.

Conseils

- Un repos par contexte.
- Commiter régulièrement.
- Mettre de vrais commentaires (éviter `git commit -m`).
- Commiter régulièrement.
- Mettre de vrais commentaires (éviter `git commit -m`).

À plusieurs

- `cd $MES_REPOS/`

À plusieurs

- `cd $MES_REPOS/`
- `git clone repos_1 repos_2`

À plusieurs

- `cd $MES_REPOS/`
- `git clone repos_1 repos_2`
- `cd $MES_REPOS/repos_1`

À plusieurs

- `cd $MES_REPOS/`
- `git clone repos_1 repos_2`
- `cd $MES_REPOS/repos_1`
- Faites quelques modifs dans `repos_1` et commiter

À plusieurs

- `cd $MES_REPOS/`
- `git clone repos_1 repos_2`
- `cd $MES_REPOS/repos_1`
- Faites quelques modifs dans `repos_1` et commiter
- `cd $MES_REPOS/repos_2`

À plusieurs

- `cd $MES_REPOS/`
- `git clone repos_1 repos_2`
- `cd $MES_REPOS/repos_1`
- Faites quelques modifs dans `repos_1` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull`

À plusieurs

- `cd $MES_REPOS/`
- `git clone repos_1 repos_2`
- `cd $MES_REPOS/repos_1`
- Faites quelques modifs dans `repos_1` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull`
- `git log`

À plusieurs

- `cd $MES_REPOS/`
- `git clone repos_1 repos_2`
- `cd $MES_REPOS/repos_1`
- Faites quelques modifs dans `repos_1` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull`
- `git log`
- Faites quelques modifs dans `repos_2` et commiter

À plusieurs

- `cd $MES_REPOS/`
- `git clone repos_1 repos_2`
- `cd $MES_REPOS/repos_1`
- Faites quelques modifs dans `repos_1` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull`
- `git log`
- Faites quelques modifs dans `repos_2` et commiter
- `git remote -v`

- `cd $MES_REPOS/repos_1`

- `cd $MES_REPOS/repos_1`
- `git remote add repos2 $MES_REPOS/repos_2`

- `cd $MES_REPOS/repos_1`
- `git remote add repos2 $MES_REPOS/repos_2`
- `git pull`

- `cd $MES_REPOS/repos_1`
- `git remote add repos2 $MES_REPOS/repos_2`
- `git pull`
- `git pull repos2 master`

- `cd $MES_REPOS/repos_1`
- `git remote add repos2 $MES_REPOS/repos_2`
- `git pull`
- `git pull repos2 master`
- Faites quelques modifs dans `repos_2` et commiter

- `cd $MES_REPOS/repos_1`
- `git remote add repos2 $MES_REPOS/repos_2`
- `git pull`
- `git pull repos2 master`
- Faites quelques modifs dans `repos_2` et commiter
- `cd $MES_REPOS/repos_1`

- `cd $MES_REPOS/repos_1`
- `git remote add repos2 $MES_REPOS/repos_2`
- `git pull`
- `git pull repos2 master`
- Faites quelques modifs dans `repos_2` et commiter
- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`

- Connectez vous sur `https://gitlab.obspm.fr`

- Connectez vous sur `https://gitlab.obspm.fr`
- Créez un projet test

- Connectez vous sur `https://gitlab.obspm.fr`
- Créez un projet test
- `cd $MES_REPOS/repos_1`

- Connectez vous sur `https://gitlab.obspm.fr`
- Créez un projet test
- `cd $MES_REPOS/repos_1`
- `git remote add origin
git@gitlab.obspm.fr:jas/test.git`

- Connectez vous sur `https://gitlab.obspm.fr`
- Créez un projet test
- `cd $MES_REPOS/repos_1`
- `git remote add origin
git@gitlab.obspm.fr:jas/test.git`
- `git push -u origin -all`

- Connectez vous sur `https://gitlab.obspm.fr`
- Créez un projet test
- `cd $MES_REPOS/repos_1`
- `git remote add origin
git@gitlab.obspm.fr:jas/test.git`
- `git push -u origin -all`
- `git push -u origin -tags`

- `cd $MES_REPOS/repos_2`

- `cd $MES_REPOS/repos_2`
- `git remote add repos1 $MES_REPOS/repos_1`

- `cd $MES_REPOS/repos_2`
- `git remote add repos1 $MES_REPOS/repos_1`
- `git remote add distant`
`git@gitlab.obspm.fr:jas/test.git`

- `cd $MES_REPOS/repos_2`
- `git remote add repos1 $MES_REPOS/repos_1`
- `git remote add distant`
`git@gitlab.obspm.fr:jas/test.git`
- `git remote -v`

- `cd $MES_REPOS/repos_2`
- `git remote add repos1 $MES_REPOS/repos_1`
- `git remote add distant`
`git@gitlab.obspm.fr:jas/test.git`
- `git remote -v`
- Faites quelques modifs dans `repos_2` et commiter
- `git push distant`

- `cd $MES_REPOS/repos_2`
- `git remote add repos1 $MES_REPOS/repos_1`
- `git remote add distant`
`git@gitlab.obspm.fr:jas/test.git`
- `git remote -v`
- Faites quelques modifs dans `repos_2` et commiter
- `git push distant`
- `git push repos1`

- `cd $MES_REPOS/repos_2`
- `git remote add repos1 $MES_REPOS/repos_1`
- `git remote add distant`
`git@gitlab.obspm.fr:jas/test.git`
- `git remote -v`
- Faites quelques modifs dans `repos_2` et commiter
- `git push distant`
- `git push repos1`
- `cd $MES_REPOS/repos_1`

- `cd $MES_REPOS/repos_2`
- `git remote add repos1 $MES_REPOS/repos_1`
- `git remote add distant`
`git@gitlab.obspm.fr:jas/test.git`
- `git remote -v`
- Faites quelques modifs dans `repos_2` et commiter
- `git push distant`
- `git push repos1`
- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`

Conflits

- `cd $MES_REPOS/repos_1`

Conflits

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`

Conflits

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`

Conflits

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`

Conflits

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter

Conflits

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites similaires modifs dans `repos_2` et commiter

Conflits

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites similaires modifs dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`

Conflits

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites similaires modifs dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`

Conflits

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites similaires modifs dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- `git status`

Conflits

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites similaires modifs dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- `git status`
- Résoudre les conflits

Conflits

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites similaires modifs dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- `git status`
- Résoudre les conflits
- `git add FICHER`

Conflits

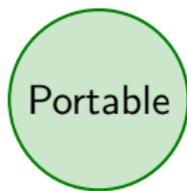
- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites similaires modifs dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- `git status`
- Résoudre les conflits
- `git add FICHER`
- `git commit`

Conflits

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites similaires modifs dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- `git status`
- Résoudre les conflits
- `git add FICHER`
- `git commit`
- `git status`

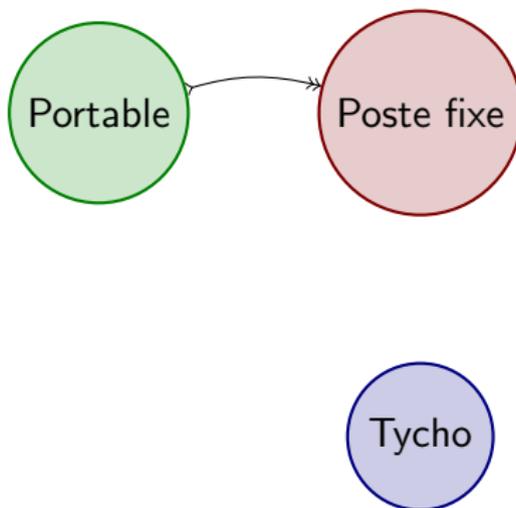
Mon code

Votre code



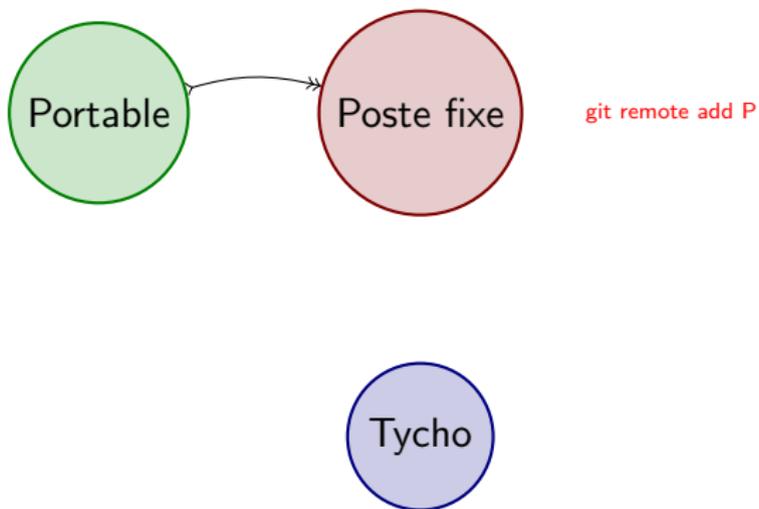
Mon code

Votre code



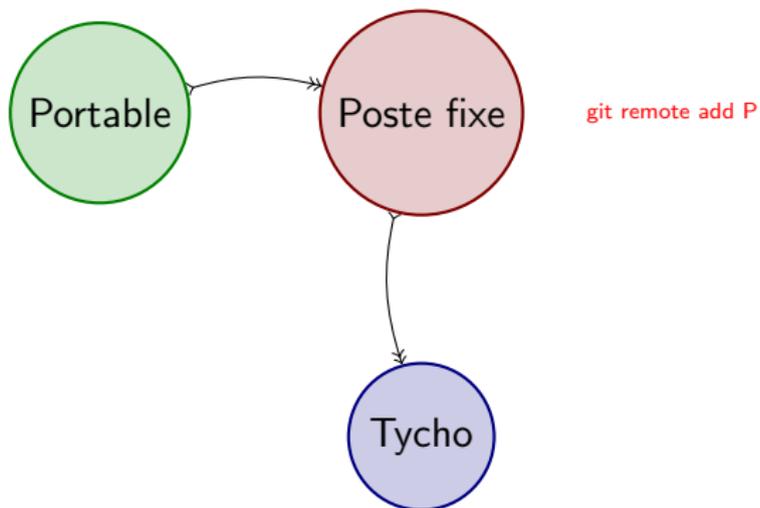
Mon code

Votre code



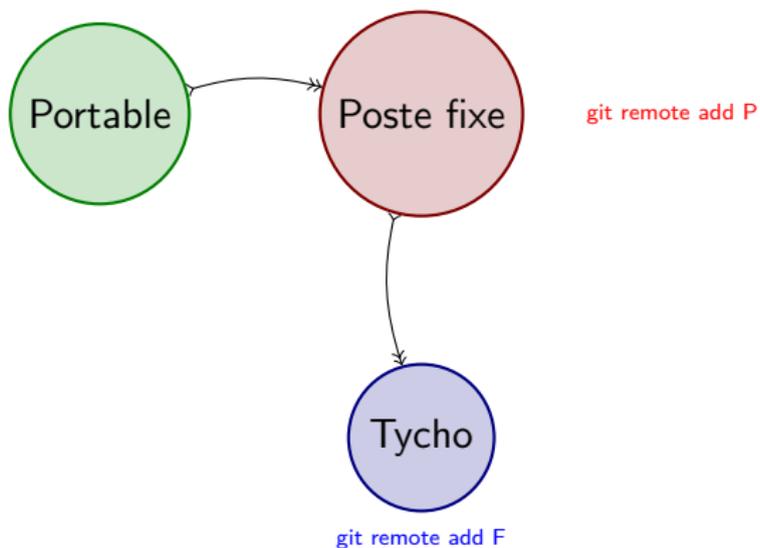
Mon code

Votre code



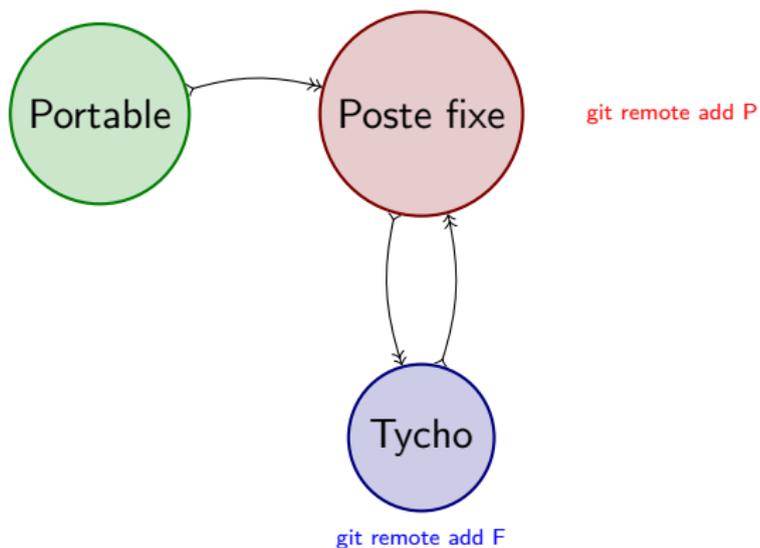
Mon code

Votre code



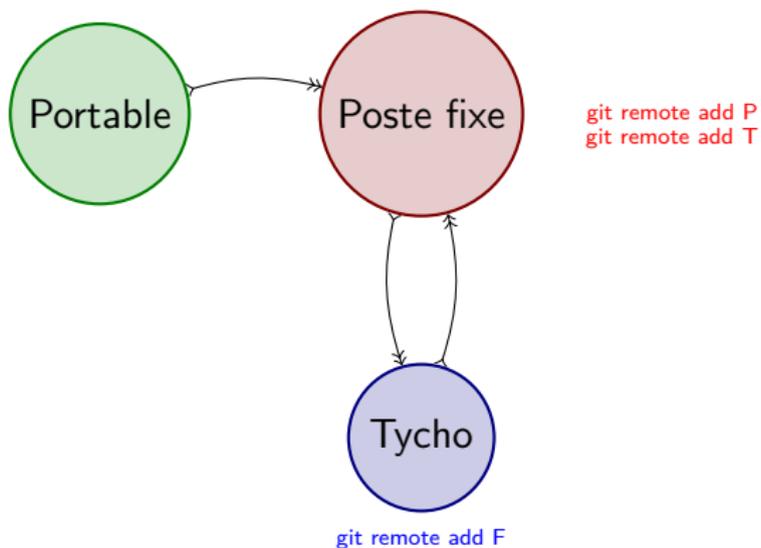
Mon code

Votre code



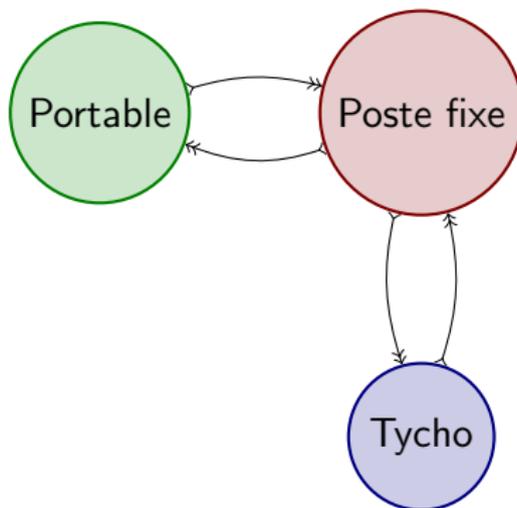
Mon code

Votre code



Mon code

Votre code



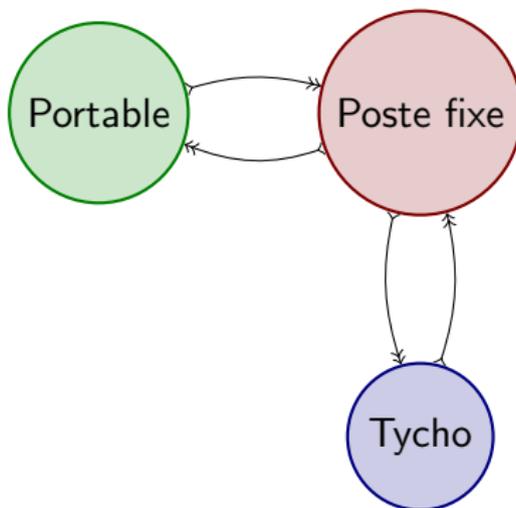
`git remote add P`
`git remote add T`

`git remote add F`

Mon code

Votre code

`git remote add F`



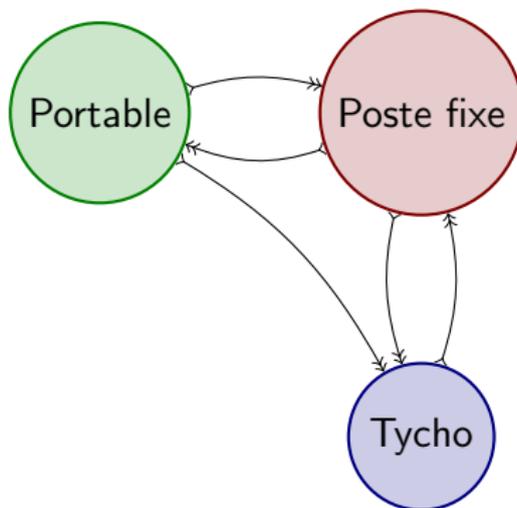
`git remote add P`
`git remote add T`

`git remote add F`

Mon code

Votre code

`git remote add F`



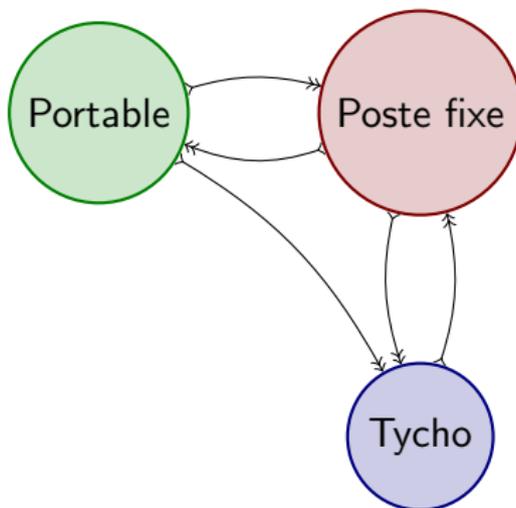
`git remote add P`
`git remote add T`

`git remote add F`

Mon code

Votre code

`git remote add F`



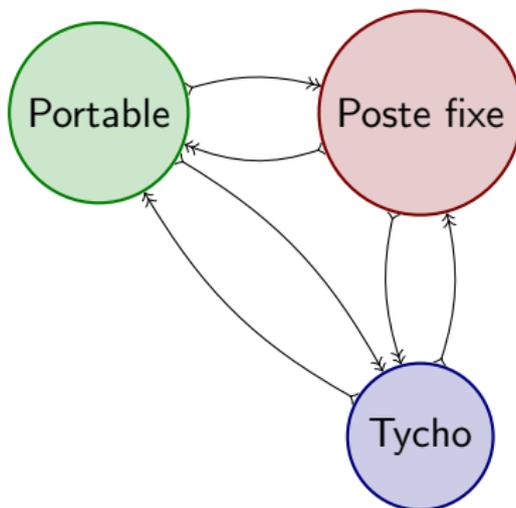
`git remote add P`
`git remote add T`

`git remote add F`
`git remote add P`

Mon code

Votre code

`git remote add F`



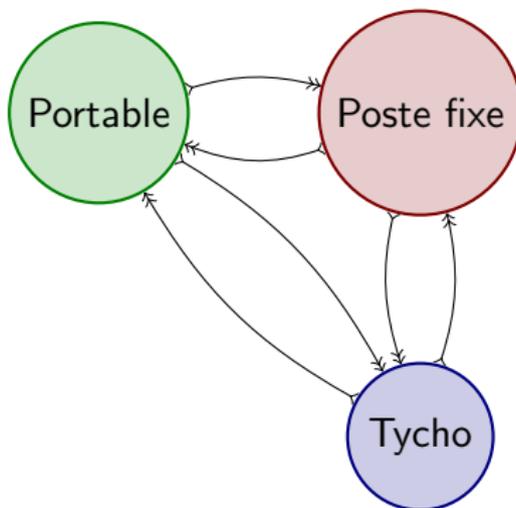
`git remote add P`
`git remote add T`

`git remote add F`
`git remote add P`

Mon code

Votre code

```
git remote add F  
git remote add T
```

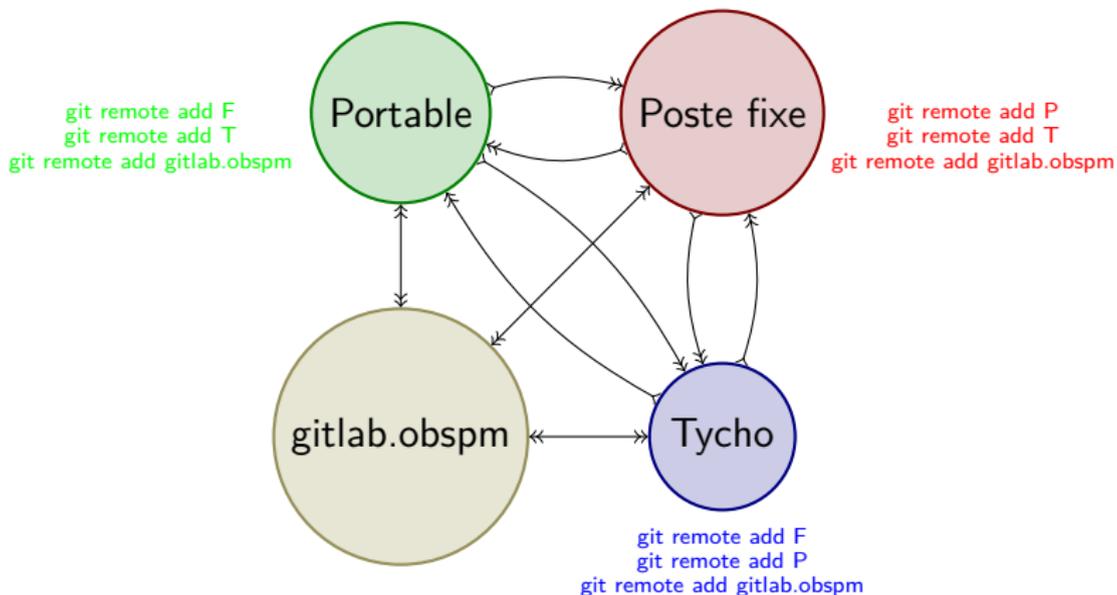


```
git remote add P  
git remote add T
```

```
git remote add F  
git remote add P
```

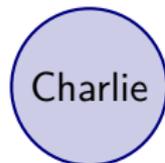
Mon code

Votre code



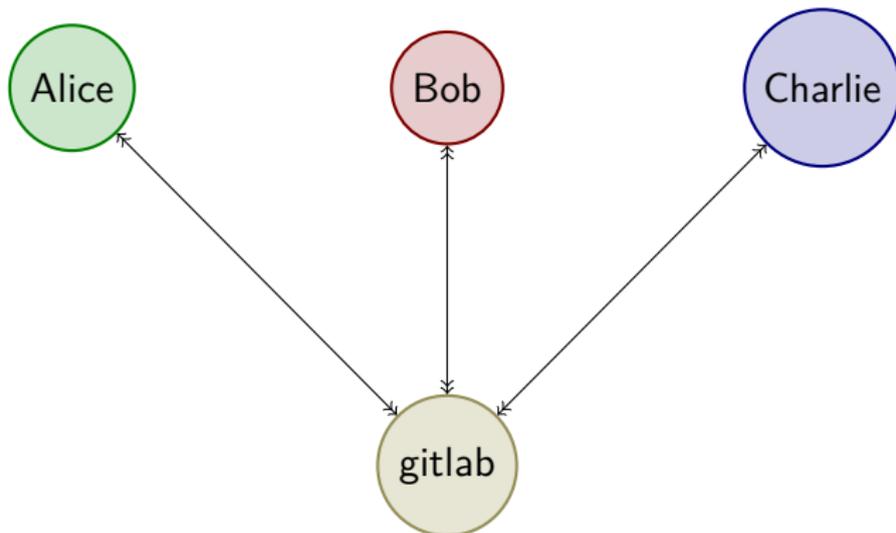
Travail à plusieurs

Projet avec Alice, Bob et Charlie.



Travail à plusieurs

Projet avec Alice, Bob et Charlie.



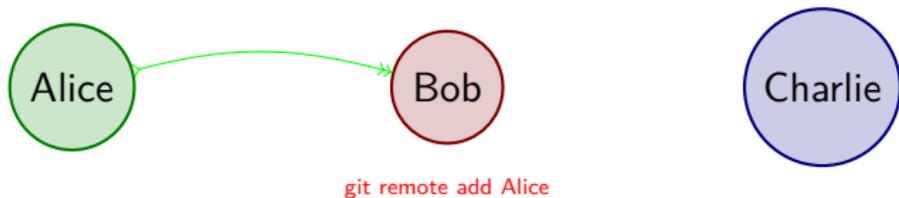
Travail à plusieurs

Projet avec Alice, Bob et Charlie.



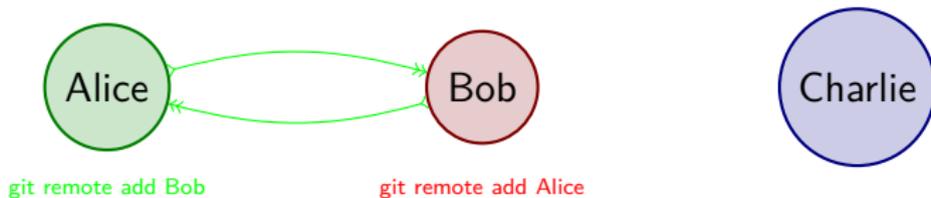
Travail à plusieurs

Projet avec Alice, Bob et Charlie.



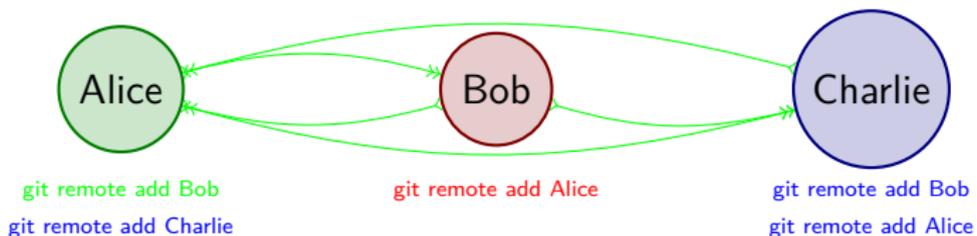
Travail à plusieurs

Projet avec Alice, Bob et Charlie.



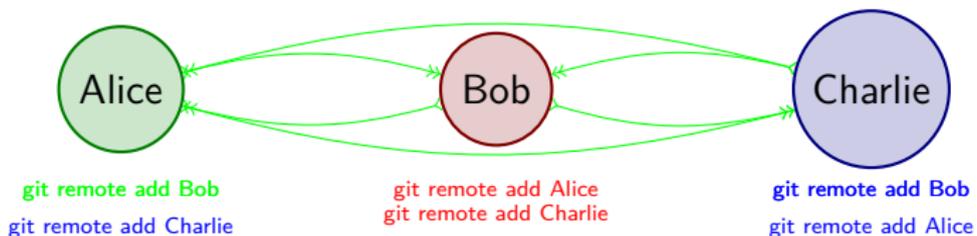
Travail à plusieurs

Projet avec Alice, Bob et Charlie.



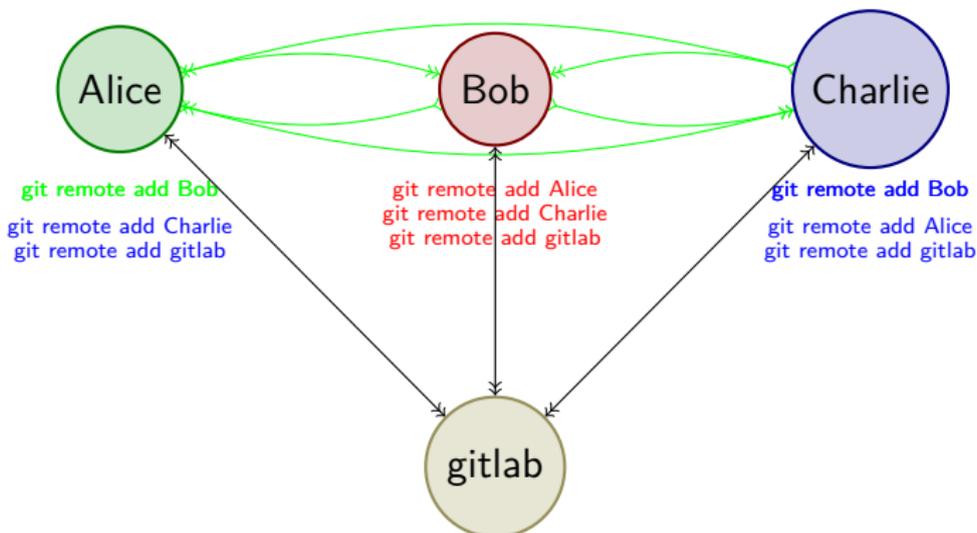
Travail à plusieurs

Projet avec Alice, Bob et Charlie.



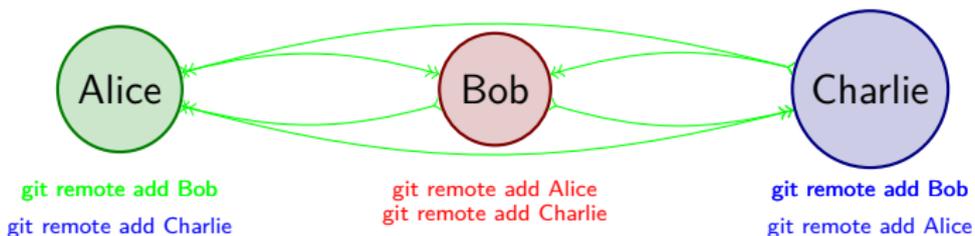
Travail à plusieurs

Projet avec Alice, Bob et Charlie.



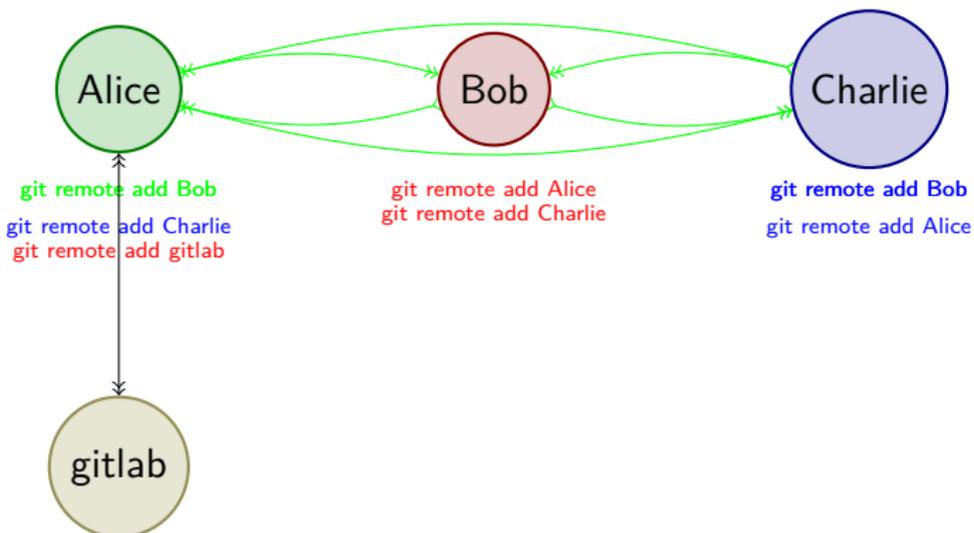
Travail à plusieurs

Projet avec Alice, Bob et Charlie.



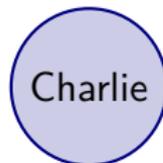
Travail à plusieurs

Projet avec Alice, Bob et Charlie.



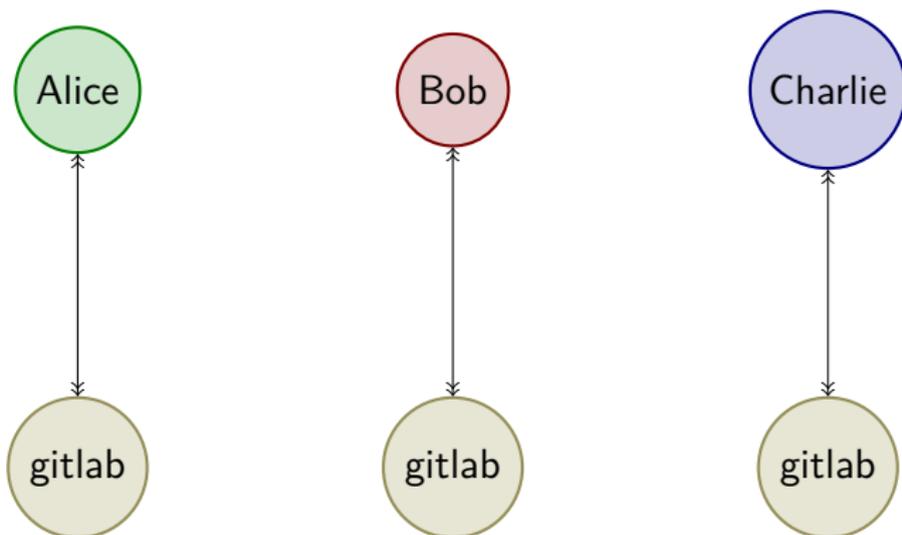
Travail à plusieurs

Projet avec Alice, Bob et Charlie.



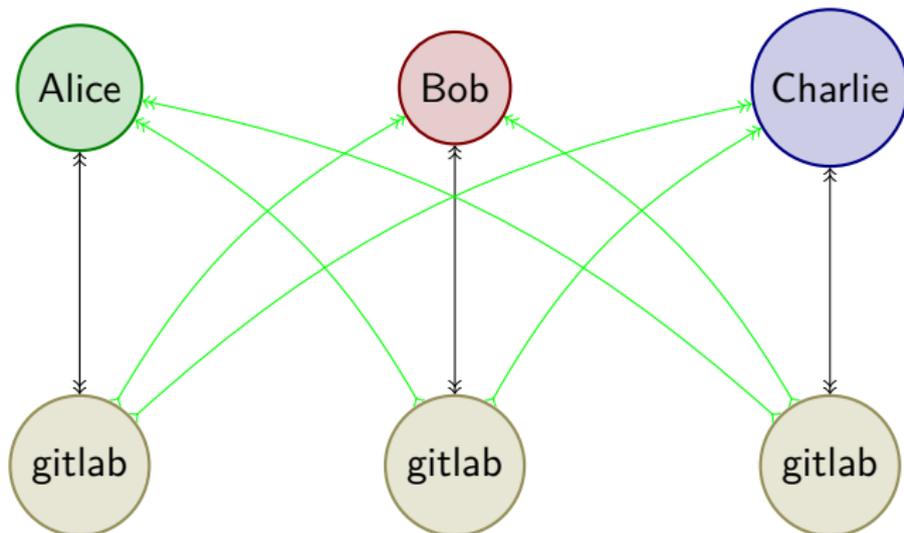
Travail à plusieurs

Projet avec Alice, Bob et Charlie.

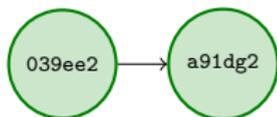


Travail à plusieurs

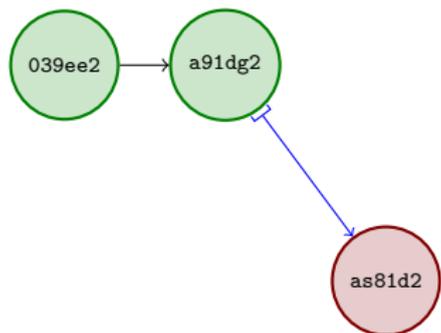
Projet avec Alice, Bob et Charlie.



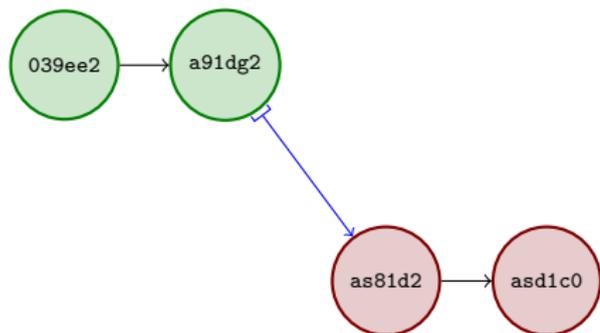
Branches



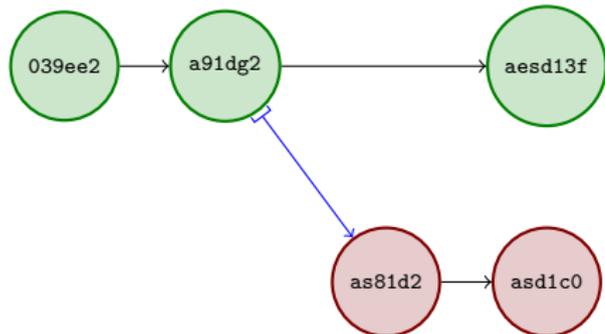
Branches



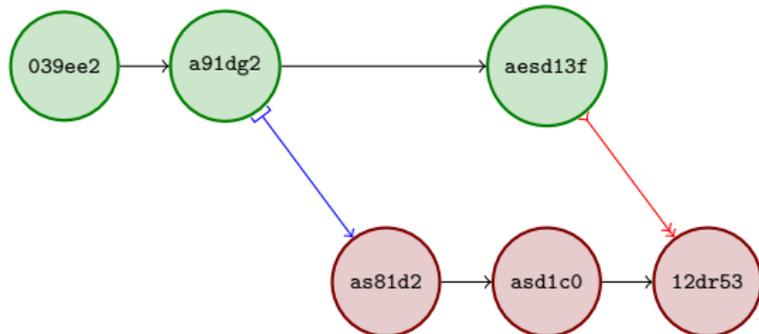
Branches



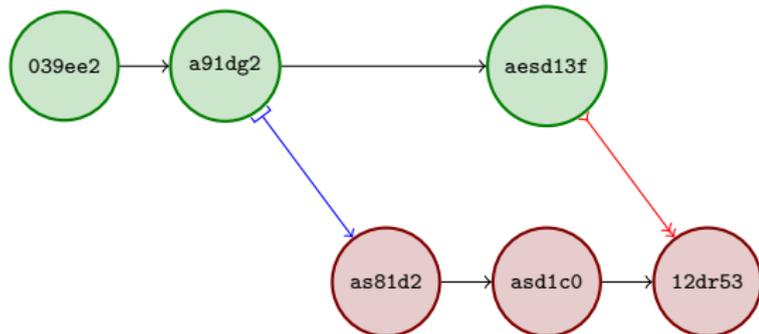
Branches



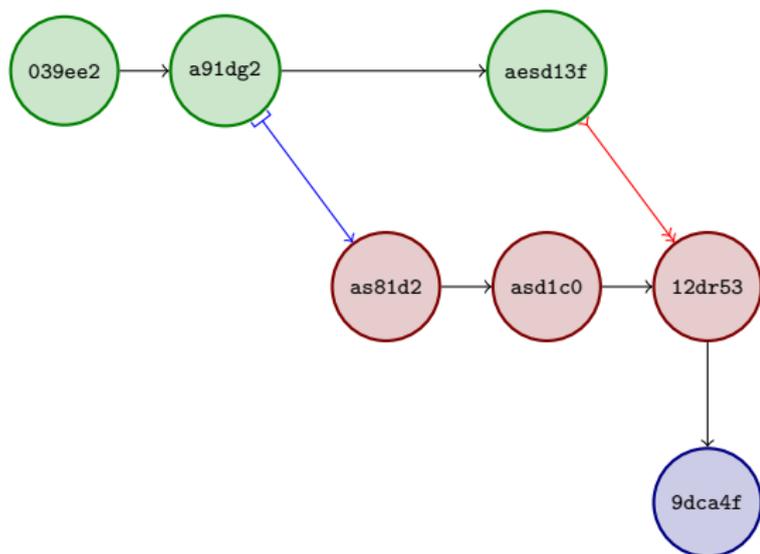
Branches



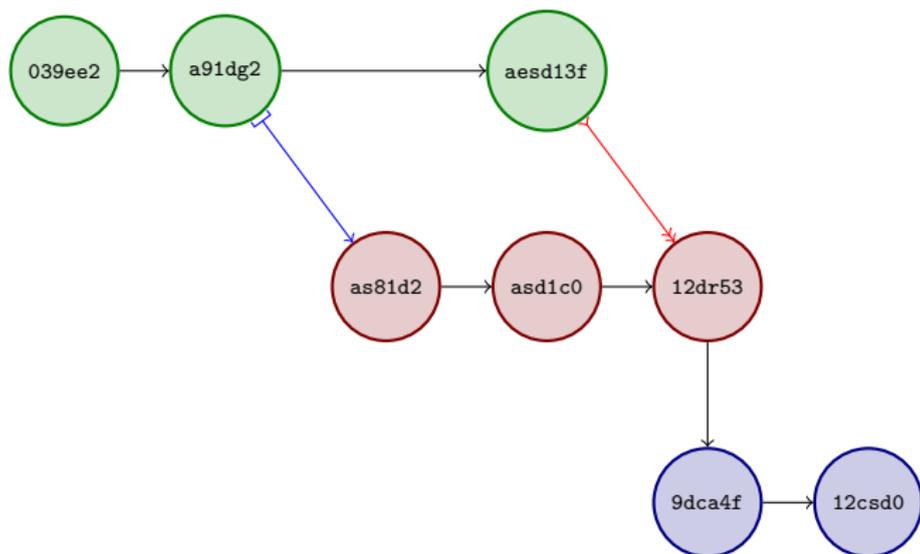
Branches



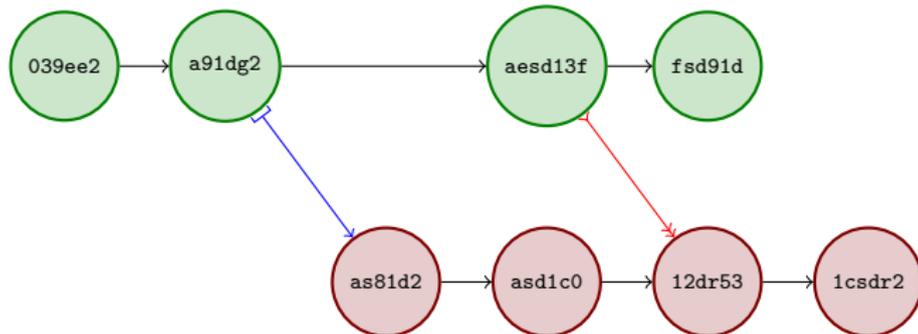
Branches



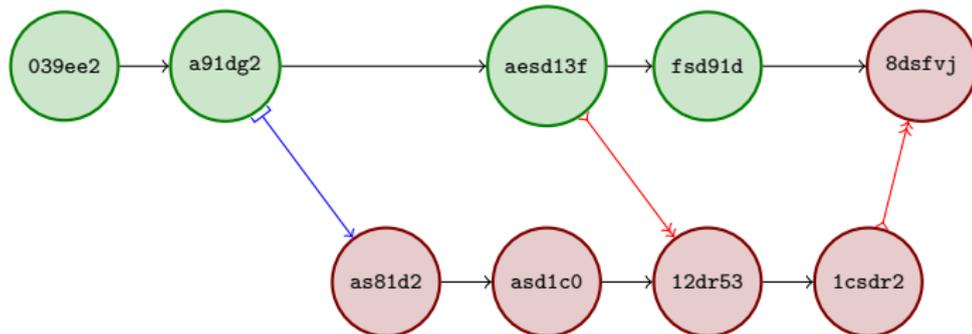
Branches



Branches



Branches



Branches

- `cd $MES_REPOS/repos_1`

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout -b essai`

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout -b essai`
- `git branch`

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout -b essai`
- `git branch`
- Faites des modifs dans et commiter plusieurs fois

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout -b essai`
- `git branch`
- Faites des modifs dans et commiter plusieurs fois
- `git checkout master`

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout -b essai`
- `git branch`
- Faites des modifs dans et commiter plusieurs fois
- `git checkout master`
- `git merge essai`

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout -b essai`
- `git branch`
- Faites des modifs dans et commiter plusieurs fois
- `git checkout master`
- `git merge essai`
- `git log --oneline -graph -decorate`
- Utiliser soit `tig` soit `gitk` pour visualiser les branches

Branches

- `cd $MES_REPOS/repos_1`

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout essai`

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout essai`
- Faites des modifs et commiter plusieurs fois

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout essai`
- Faites des modifs et commiter plusieurs fois
- `git checkout master`

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout essai`
- Faites des modifs et commiter plusieurs fois
- `git checkout master`
- Faites des modifs dans un fichier différent et commiter plusieurs fois

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout essai`
- Faites des modifs et commiter plusieurs fois
- `git checkout master`
- Faites des modifs dans un fichier différent et commiter plusieurs fois
- `git merge essai`

Branches

- `cd $MES_REPOS/repos_1`
- `git branch`
- `git checkout essai`
- Faites des modifs et commiter plusieurs fois
- `git checkout master`
- Faites des modifs dans un fichier différent et commiter plusieurs fois
- `git merge essai`
- `git log --oneline -graph -decorate`
- Utiliser soit `tig` soit `gitk` pour visualiser les branches

Les branches et les remotes

- `cd $MES_REPOS/repos_2`

Les branches et les remotes

- `cd $MES_REPOS/repos_2`
- `git fetch`

Les branches et les remotes

- `cd $MES_REPOS/repos_2`
- `git fetch`
- `git branch`

Les branches et les remotes

- `cd $MES_REPOS/repos_2`
- `git fetch`
- `git branch`
- `git checkout -b monessai origin/essai`

Les branches et les remotes

- `cd $MES_REPOS/repos_2`
- `git fetch`
- `git branch`
- `git checkout -b monessai origin/essai`
- `cd $MES_REPOS/repos_1`

Les branches et les remotes

- `cd $MES_REPOS/repos_2`
- `git fetch`
- `git branch`
- `git checkout -b monessai origin/essai`
- `cd $MES_REPOS/repos_1`
- `git checkout essai`

Les branches et les remotes

- `cd $MES_REPOS/repos_2`
- `git fetch`
- `git branch`
- `git checkout -b monessai origin/essai`
- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter plusieurs fois

Les branches et les remotes

- `cd $MES_REPOS/repos_2`
- `git fetch`
- `git branch`
- `git checkout -b monessai origin/essai`
- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter plusieurs fois
- `cd $MES_REPOS/repos_2`

Les branches et les remotes

- `cd $MES_REPOS/repos_2`
- `git fetch`
- `git branch`
- `git checkout -b monessai origin/essai`
- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter plusieurs fois
- `cd $MES_REPOS/repos_2`
- `git pull`

Les branches et les remotes

- `cd $MES_REPOS/repos_2`
- `git fetch`
- `git branch`
- `git checkout -b monessai origin/essai`
- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter plusieurs fois
- `cd $MES_REPOS/repos_2`
- `git pull`
- Faites des modifs et commiter plusieurs fois

Les branches et les remotes

- `cd $MES_REPOS/repos_1`

Les branches et les remotes

- `cd $MES_REPOS/repos_1`
- `git branch -a`

Les branches et les remotes

- `cd $MES_REPOS/repos_1`
- `git branch -a`
- `git fetch repos2`

Les branches et les remotes

- `cd $MES_REPOS/repos_1`
- `git branch -a`
- `git fetch repos2`
- `git branch -a`

Les branches et les remotes

- `cd $MES_REPOS/repos_1`
- `git branch -a`
- `git fetch repos2`
- `git branch -a`
- `git checkout essai`

Les branches et les remotes

- `cd $MES_REPOS/repos_1`
- `git branch -a`
- `git fetch repos2`
- `git branch -a`
- `git checkout essai`
- `git branch --set-upstream-to=repos2/monessai`
`essai`

Les branches et les remotes

- `cd $MES_REPOS/repos_1`
- `git branch -a`
- `git fetch repos2`
- `git branch -a`
- `git checkout essai`
- `git branch --set-upstream-to=repos2/monessai
essai`
- `git pull`

Les branches et les remotes

- `cd $MES_REPOS/repos_1`
- `git branch -a`
- `git fetch repos2`
- `git branch -a`
- `git checkout essai`
- `git branch --set-upstream-to=repos2/monessai
essai`
- `git pull`
- `git log --oneline --decorate --graph ou tig ou gitk`

Branche vs revert

- Beaucoup plus souple pour basculer.

Branche vs revert

- Beaucoup plus souple pour basculer.
- Plus facile pour les diff.

Branche vs revert

- Beaucoup plus souple pour basculer.
- Plus facile pour les diff.
- Historique plus propre.

Branche vs revert

- Beaucoup plus souple pour basculer.
- Plus facile pour les diff.
- Historique plus propre.
- Plus facile à mettre en place sur différentes machines.

Mon code

prod

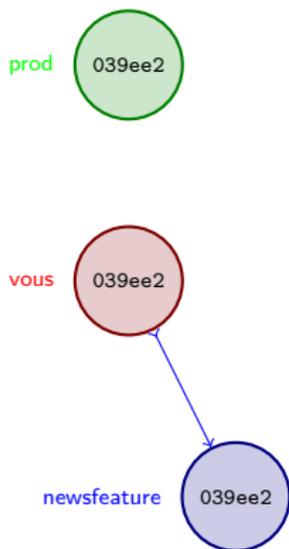


vous



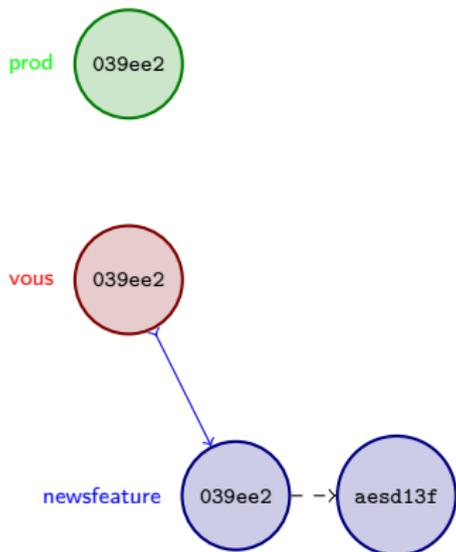
Mon code

```
git checkout -b newfeature
```



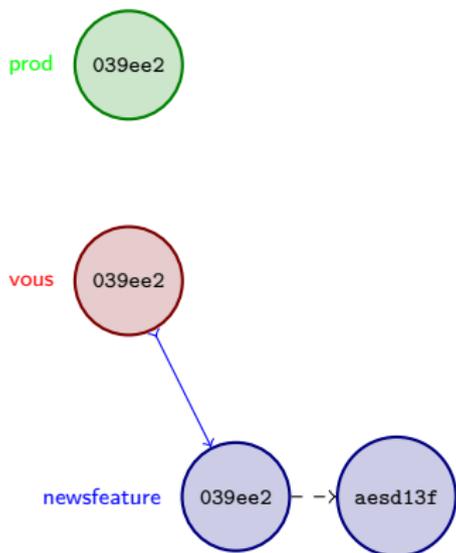
Mon code

```
vi;git commit;vi;git commit;etc...
```



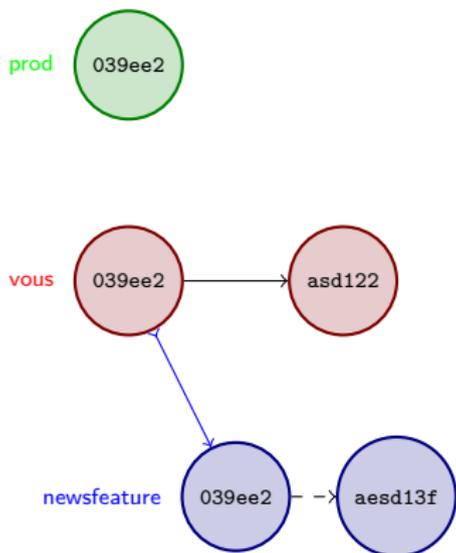
Mon code

```
git checkout master;vi;git commit;etc...
```



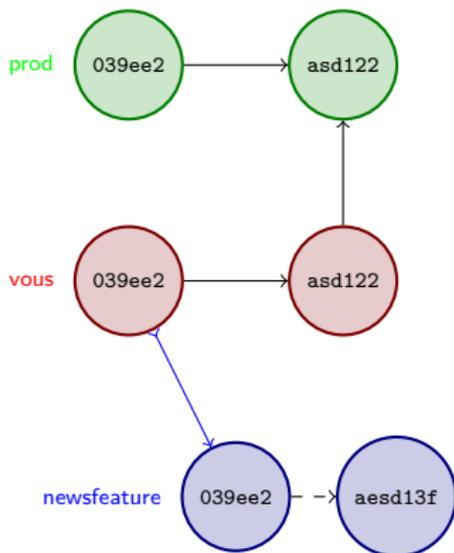
Mon code

```
git checkout master;vi;git commit;etc...
```

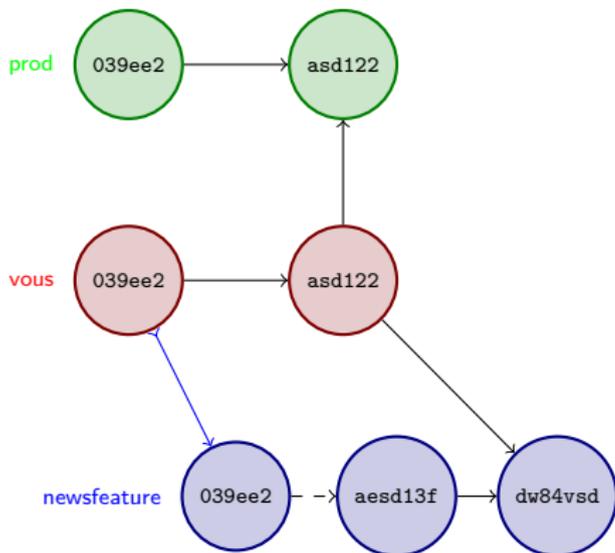


Mon code

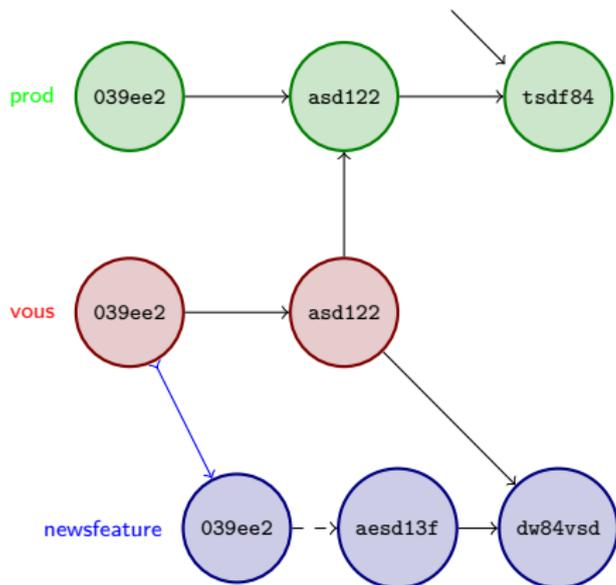
```
ssh prod;git pull vous master...
```



Mon code

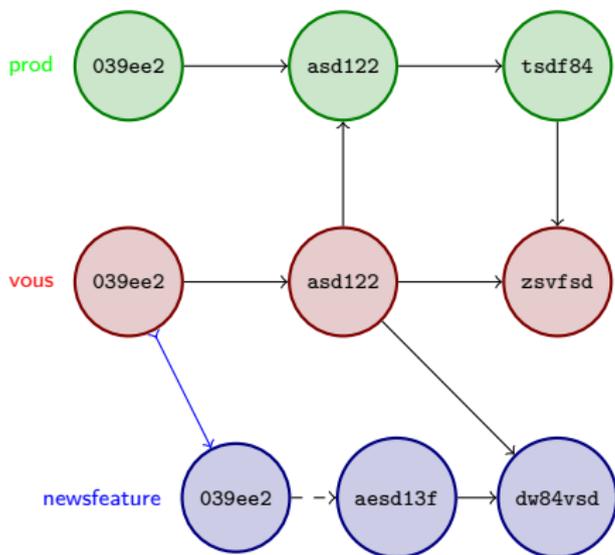


Mon code



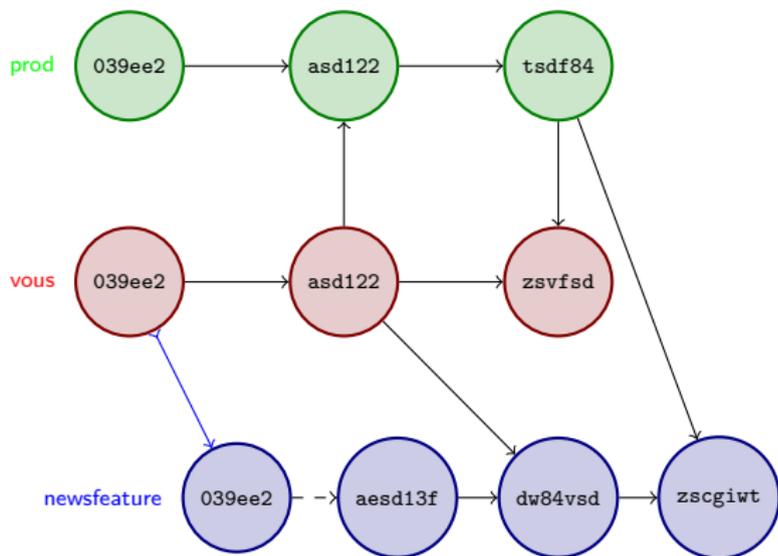
Mon code

```
ssh vous;git checkout master;git pull
```

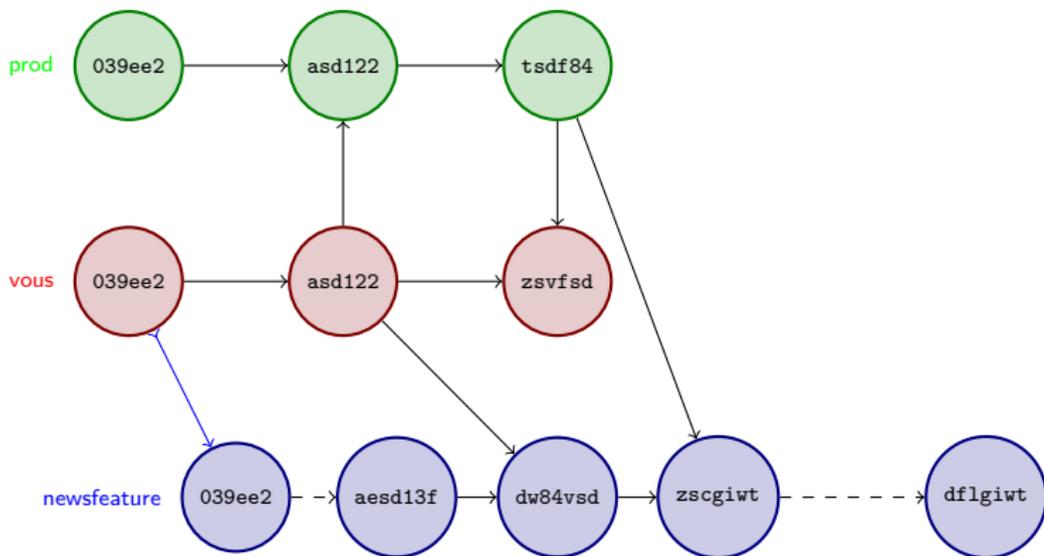


Mon code

```
git checkout newfeature;git pull origin newfeature
```

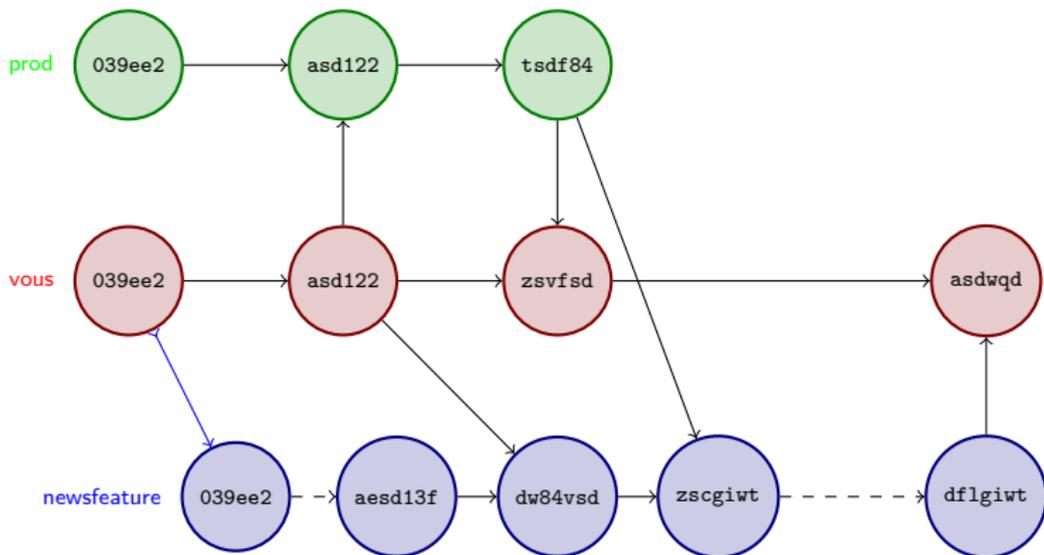


Mon code

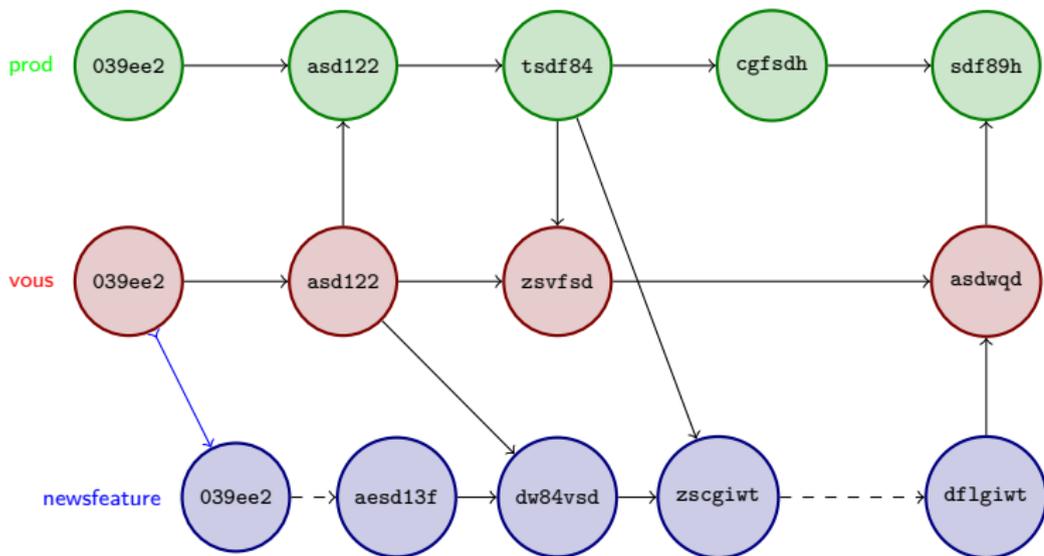


Mon code

```
git checkout master;git merge newfeature
```



Mon code



Historique

- **Très important**

Historique

- **Très important**
- `git amend`

Historique

- **Très** important
- `git amend`
- Ré-écriture ou pas ?

Historique

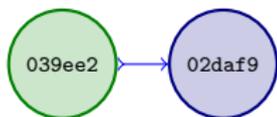
- **Très** important
- `git amend`
- Ré-écriture ou pas ?
- Historique propre.

Historique

- **Très** important
- `git amend`
- Ré-écriture ou pas ?
- Historique propre.
- Historique lisible.

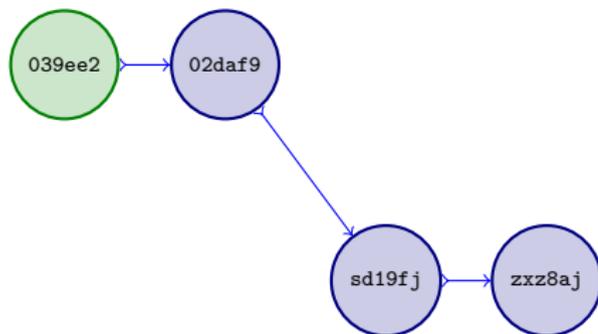
Merge en fast-forward

Fast forward



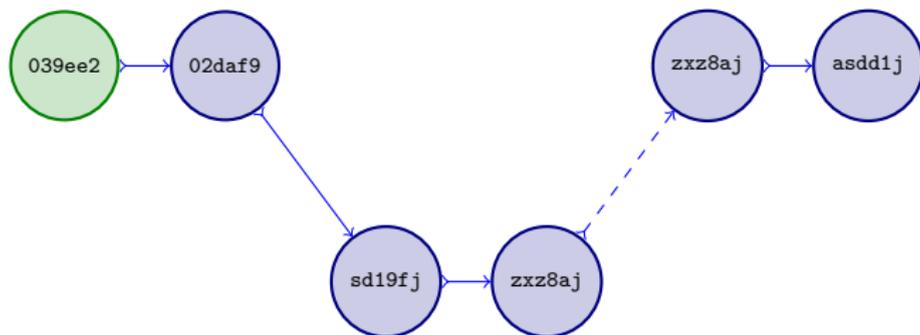
Merge en fast-forward

Fast forward



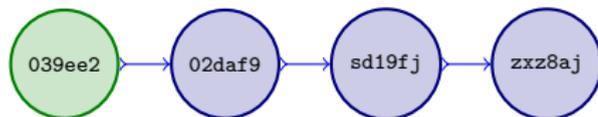
Merge en fast-forward

Fast forward



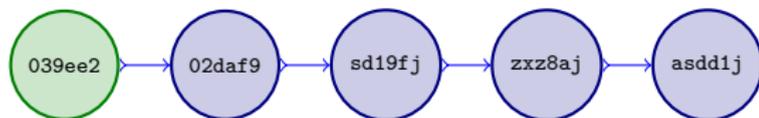
Merge en fast-forward

Fast forward



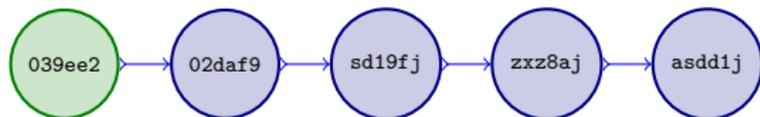
Merge en fast-forward

Fast forward



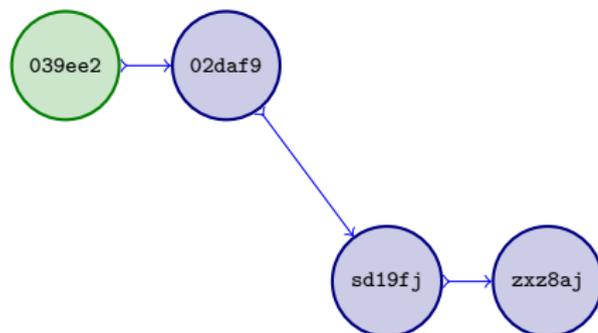
Fast forward

fast forward



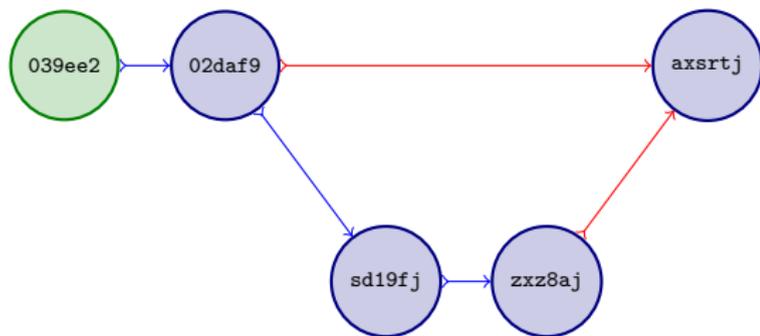
Merge en fast-forward

-no-ff



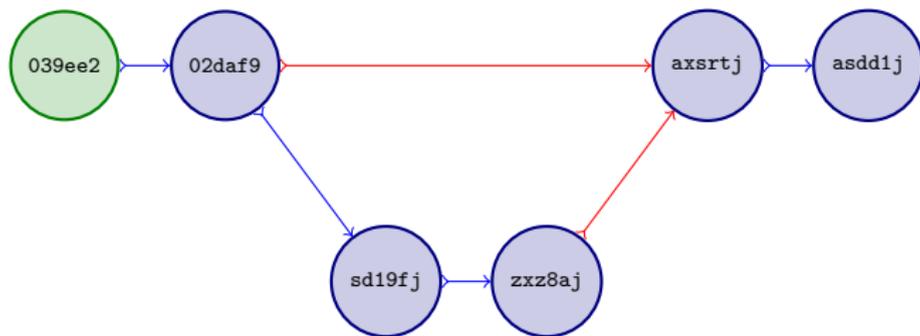
Merge en fast-forward

-no-ff



Merge en fast-forward

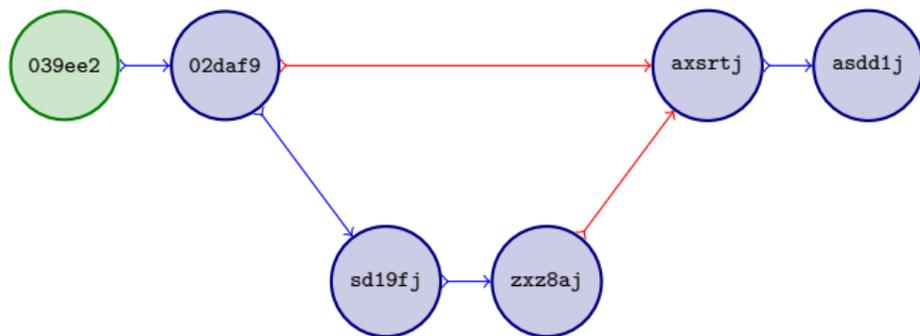
-no-ff



Merge en fast-forward

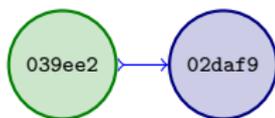
-no-ff

```
git merge --no-ff
```



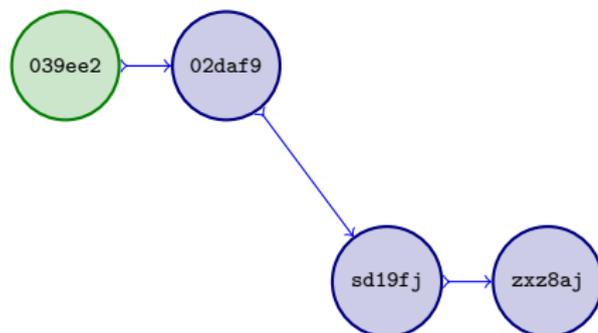
Merge sans fast forward

No fast forward



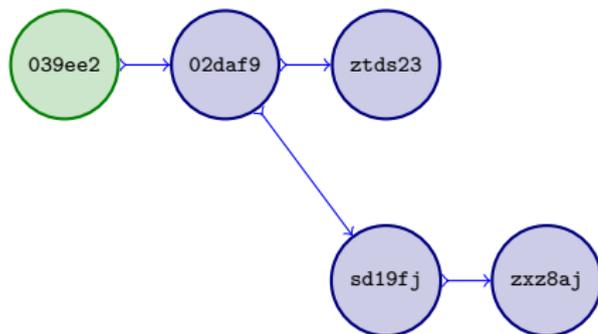
Merge sans fast forward

No fast forward



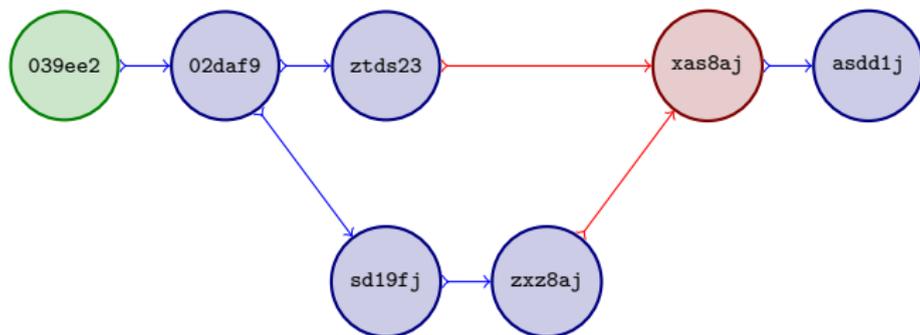
Merge sans fast forward

No fast forward



Merge sans fast forward

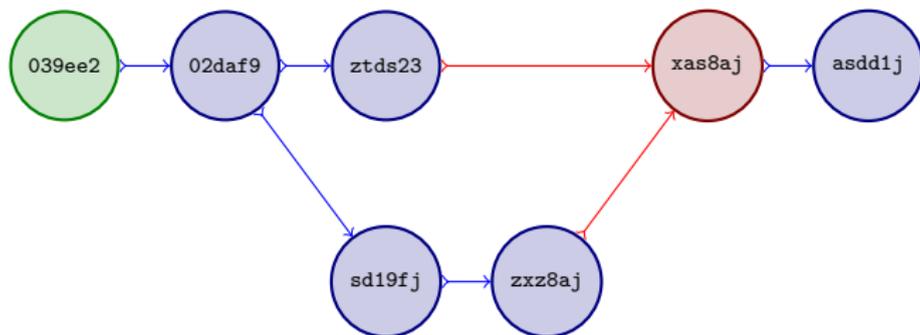
No fast forward



Merge sans fast forward

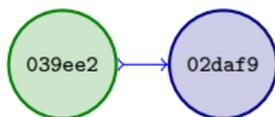
No fast forward

no fast forward



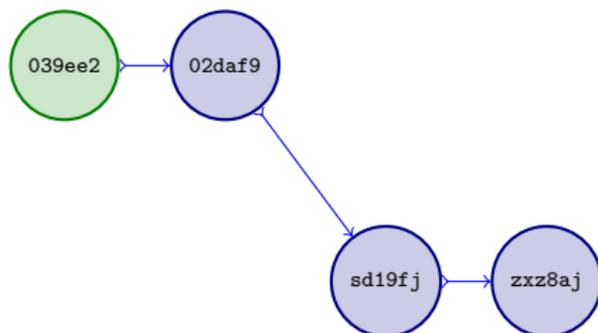
Merge sans fast forward

rebase + ff



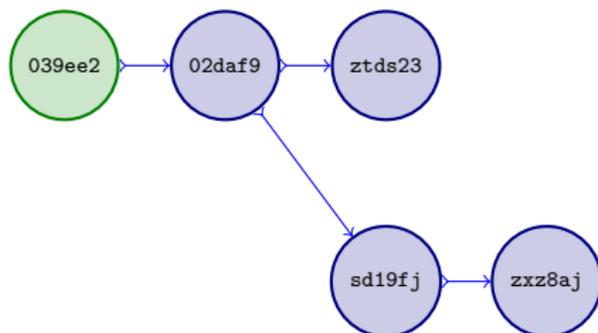
Merge sans fast forward

rebase + ff



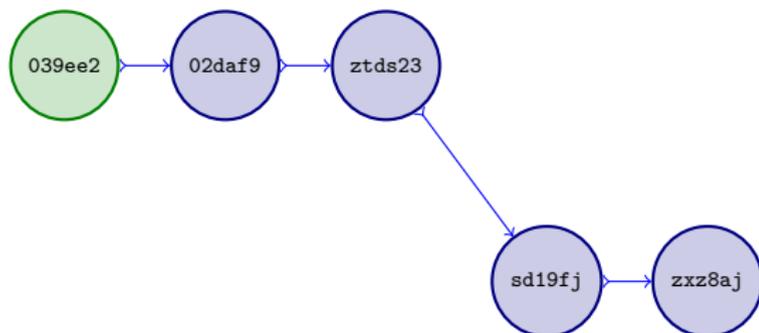
Merge sans fast forward

rebase + ff

`git rebase`

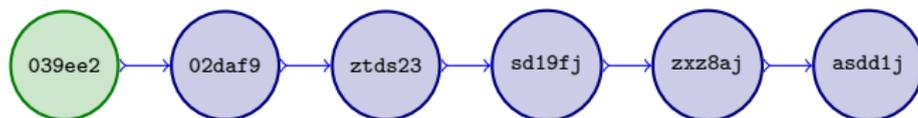
Merge sans fast forward

rebase + ff

`git rebase`

Merge sans fast forward

rebase + ff

`git rebase`

Tout droit ou pas (Fast forward)

- `cd $MES_REPOS/repos_1`

Tout droit ou pas (Fast forward)

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`

Tout droit ou pas (Fast forward)

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`

Tout droit ou pas (Fast forward)

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`

Tout droit ou pas (Fast forward)

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter

Tout droit ou pas (Fast forward)

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter

Tout droit ou pas (Fast forward)

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- `cd $MES_REPOS/repos_2`

Tout droit ou pas (Fast forward)

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`

Tout droit ou pas (Fast forward)

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Utiliser soit `tig` soit `gitk` soit `git log --oneline --graph --decorate` pour visualiser les branches

Tout droit ou pas

- `cd $MES_REPOS/repos_1`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites quelques modifs (sans conflit) dans `repos_2` et commiter

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites quelques modifs (sans conflit) dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites quelques modifs (sans conflit) dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites quelques modifs (sans conflit) dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Utiliser soit `tig` soit `gitk` soit `git log --oneline --graph --decorate` pour visualiser les branches

Tout droit ou pas

- `cd $MES_REPOS/repos_1`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites quelques modifs (sans conflit) dans `repos_2` et commiter

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites quelques modifs (sans conflit) dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites quelques modifs (sans conflit) dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull --rebase repos1 master`

Tout droit ou pas

- `cd $MES_REPOS/repos_1`
- `git pull repos2 master`
- `cd $MES_REPOS/repos_2`
- `git pull repos1 master`
- Faites quelques modifs dans `repos_1` et commiter
- Faites quelques modifs (sans conflit) dans `repos_2` et commiter
- `cd $MES_REPOS/repos_2`
- `git pull --rebase repos1 master`
- Utiliser soit `tig` soit `gitk` soit `git log --oneline --graph --decorate` pour visualiser les branches

- Ré-écriture de l'histoire.

- Ré-écriture de l'histoire.
- Historique propre vs. *honnête*.

- Ré-écriture de l'histoire.
- Historique propre vs. *honnête*.
- Utilisation de rebase.
- Ne jamais faire après un **push**.

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`
- `git checkout essai`

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter ≥ 3 fois

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter ≥ 3 fois
- `git log`

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter ≥ 3 fois
- `git log`
- `git checkout nettoyage`

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter ≥ 3 fois
- `git log`
- `git checkout nettoyage`
- `git rebase -i HEAD~3`

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter ≥ 3 fois
- `git log`
- `git checkout nettoyage`
- `git rebase -i HEAD~3`
- `git log`

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter ≥ 3 fois
- `git log`
- `git checkout nettoyage`
- `git rebase -i HEAD~3`
- `git log`
- `git checkout essai`

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter ≥ 3 fois
- `git log`
- `git checkout nettoyage`
- `git rebase -i HEAD~3`
- `git log`
- `git checkout essai`
- `git merge nettoyage`

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter ≥ 3 fois
- `git log`
- `git checkout nettoyage`
- `git rebase -i HEAD~3`
- `git log`
- `git checkout essai`
- `git merge nettoyage`
- `git checkout master`

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter ≥ 3 fois
- `git log`
- `git checkout nettoyage`
- `git rebase -i HEAD~3`
- `git log`
- `git checkout essai`
- `git merge nettoyage`
- `git checkout master`
- `git merge essai`

Ré-écriture

Ne pas faire si vous avez déjà fait git push Ne pas faire dans master

- `cd $MES_REPOS/repos_1`
- `git checkout essai`
- Faites des modifs et commiter ≥ 3 fois
- `git log`
- `git checkout nettoyage`
- `git rebase -i HEAD~3`
- `git log`
- `git checkout essai`
- `git merge nettoyage`
- `git checkout master`
- `git merge essai`
- `git log`

À faire

- Un commit = petit diff.

À faire

- Un commit = petit diff.
- Commiter, commiter, ...

À faire

- Un commit = petit diff.
- Committer, commiter, ...
- Utiliser les branches.

À faire

- Un commit = petit diff.
- Commiter, commiter, ...
- Utiliser les branches.
- Faites des **vrai** message de log.

À faire

- Un commit = petit diff.
- Commiter, commiter, ...
- Utiliser les branches.
- Faites des **vrai** message de log.
- Ré-écrire l'historique ou pas.

À faire

- Un commit = petit diff.
- Committer, commiter, ...
- Utiliser les branches.
- Faites des **vrai** message de log.
- Ré-écrire l'historique ou pas.
- Faites des historiques propres.

À ne pas faire

- Beaucoup de modifications dans un commit.

À ne pas faire

- Beaucoup de modifications dans un commit.
- `git commit -m`

À ne pas faire

- Beaucoup de modifications dans un commit.
- `git commit -m`
- Ne **jamais** faire rebase sur des données déjà pusher/puller.

Pour aller plus loin

Cerises

- tags

Pour aller plus loin

Cerises

- tags
- stash
- autosquash

Pour aller plus loin

Cerises

- tags
- stash
- autosquash
- cherry-pick

Pour aller plus loin

Cerises

- tags
- stash
- autosquash
- cherry-pick
- rerere

Pour aller plus loin

Cerises

- tags
- stash
- autosquash
- cherry-pick
- rerere
- bisect

Pour aller plus loin

Cerises

- tags
- stash
- autosquash
- cherry-pick
- rerere
- bisect
- subtree

Pour aller plus loin

Cerises

- tags
- stash
- autosquash
- cherry-pick
- rerere
- bisect
- subtree
- submodules

Pour aller plus loin

Merci pour votre attention