

# Introduction à Git

Anne Cadiou

Laboratoire de Mécanique des Fluides et d'Acoustique

Informatique scientifique pour le calcul  
École doctorale  
2015-2016



# Table des matières

- 1 Contexte
- 2 Systèmes de gestion de versions
- 3 Git
- 4 Exemple local
- 5 Exemple collaboratif
- 6 Résumé

# Table des matières

- 1 Contexte
- 2 Systèmes de gestion de versions
- 3 Git
- 4 Exemple local
- 5 Exemple collaboratif
- 6 Résumé

# Méthodologie

Mise en place de **bonnes pratiques** pour le développement

Adopter une méthodologie d'**intégration continue**

Vise à l'automatisation des tâches (compilation, tests unitaires et fonctionnels, tests de performances, validation, documentation ...)

Adopter une méthodologie **agile**

(d'après Wikipédia)

- *les individus et leurs interactions*, plus que les processus et les outils
- *des logiciels opérationnels*, plus qu'une documentation exhaustive
- *la collaboration* avec les clients, plus que la négociation contractuelle
- *l'adaptation* au changement, plus que le suivi d'un plan

## ... et outils

### En pratique

- Code partagé et versionné
- Modifications fréquentes, testées automatiquement
- Souplesse dans les développements
- Version fonctionnelle toujours disponible  
(tests, documentation, démonstration ou production)

Nécessite l'utilisation d'un minimum d'**outils de génie logiciel**

# Outils de génie logiciel (1/2)

- Atelier de génie logiciel, environnement de développement
  - Eclipse
  - Visual Studio (propriétaire)

## Principe (*d'après Wikipédia*)

- conception générale du projet, étapes ou phases de réalisation
- composition et organisation de l'équipe projet
- calendrier, charges de travail, moyens et budgets
- conventions de nommage des données et des programmes
- structuration des données
- aide à l'édition de programmes dans différents langages
- compilation
- génération de code optimisé
- édition de liens
- aide aux tests et suivi des corrections
- bibliothèques pouvant être réutilisées dans plusieurs projets
- documentations
- gestion des versions successives des programmes

## Outils de génie logiciel (2/2)

- Forge [logicielle](#), gestion de projets de développement collaboratifs
  - Redmine
  - Tuleap
  - Trac
  - GitLab
  - FusionForge (reprise opensource de GForge, devenue propriétaire)

Fonctions disponibles d'une forge (*d'après Wikipédia*)

- système de gestion des versions (par exemple, via Git ou Mercurial)
- gestionnaire de listes de discussion (et/ou de forums)
- outil de suivi des bugs
- gestionnaire de documentation (souvent sur le principe du wiki)
- gestion des tâches

Dénominateur commun :

[système de gestion de version](#)

# Table des matières

1 Contexte

2 Systèmes de gestion de versions

3 Git

4 Exemple local

5 Exemple collaboratif

6 Résumé



# À quoi ça sert ?

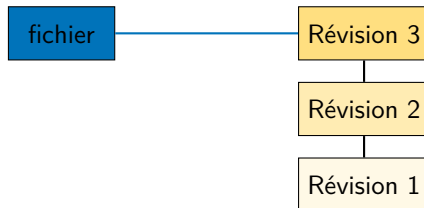
Systèmes de gestion de version (*en anglais Version System Control*)

Sert à :

- enregistrer et sauvegarder les développements au cours du temps
- revenir en arrière sur une version spécifique
- collaborer sur un même document
- enregistrer les modifications avec auteur et date

## Modes de travail (1/3)

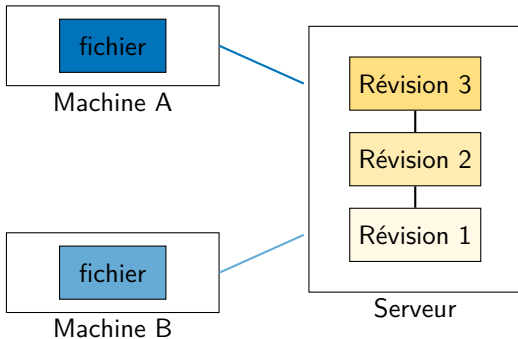
Projet local



Généralement facile d'utilisation, léger à configurer  
Collaboration pas vraiment prévue

## Modes de travail (2/3)

### Projet centralisé



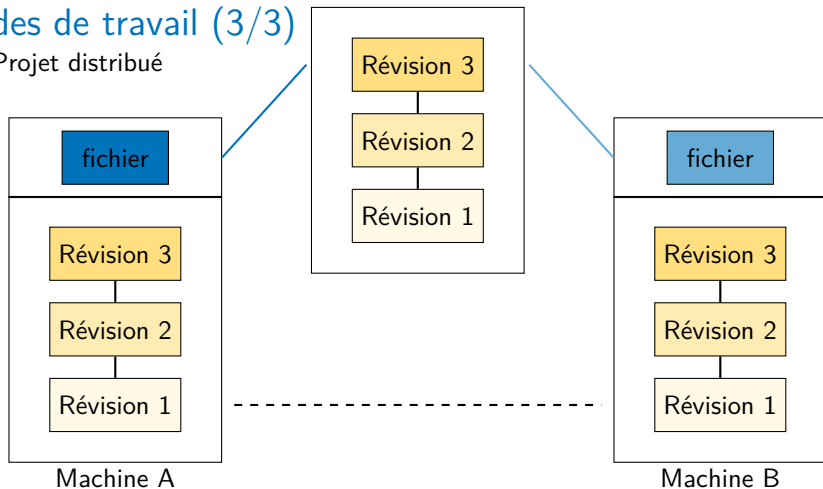
Partage d'une archive sur un serveur unique

Accès distant au serveur

Nécessite une connexion internet

## Modes de travail (3/3)

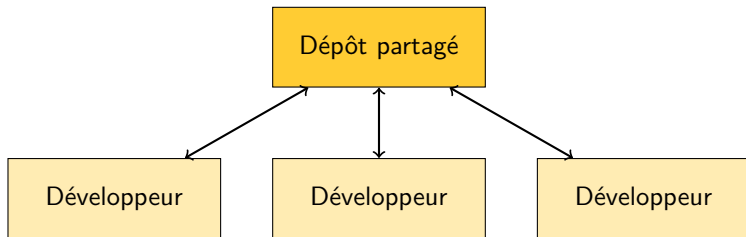
### Projet distribué



Chacun travaille sur un clone du dépôt, gère des dépôts public/privé  
Permet de collaborer de différentes manières avec différentes personnes  
sur un même projet  
Pas besoin d'accès internet

# Modes d'organisation (1/3)

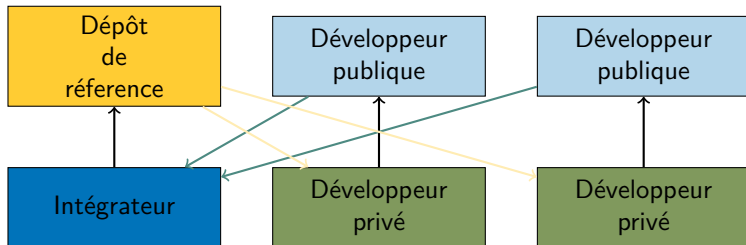
## Organisation centralisée



Chacun échange avec le dépôt de référence.  
Les développeurs doivent mettre à jour leurs clones  
avant de modifier le dépôt.

## Modes d'organisation (2/3)

### Organisation via un intégrateur



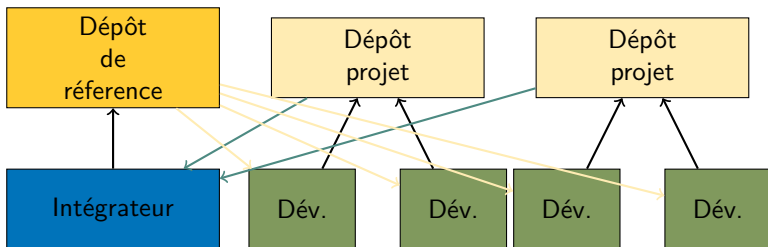
Chacun travaille avec un clone du dépôt de référence.

Chacun dépose ses développements dans un dépôt privé.

Chacun demande à l'intégrateur de déposer dans la référence.

## Modes d'organisation (3/3)

### Organisation en sous-projet



Adapté à des gros projets, avec sous-parties de développement.

# Historique des principaux opensources (1/2)

- **Système local**

- **RCS** (Revision Control System)

- les développeurs partagent le même système de fichiers

- **Système client-serveur**

- **CVS** (Concurrent Versions System)

- dérivé de RCS, les développeurs partagent un dépôt unique

- **Subversion** (SVN)

- dérivé de CVS, fusions facilitées



# Historique des principaux opensources (2/2)

## Distributed Version Control System (DVCS)

- **Système décentralisé**

- **Git**

- conçu par Linus Torvalds pour le projet du noyau Linux en 2005

- **Mercurial**

- en Python, évolution depuis 2005 de l'opensource de BitKeeper, anciennement utilisé pour le développement du noyau Linux (2002-2005)

- **Bazaar**

- en Python, depuis 2005

- **Fossil**

- en C, depuis 2006

- **Veracity**

- depuis 2011

- intègre le suivi des bug et les outils de développement agile pas de rebase

# Table des matières

- 1 Contexte
- 2 Systèmes de gestion de versions
- 3 Git**
- 4 Exemple local
- 5 Exemple collaboratif
- 6 Résumé

# Apprendre git

<https://www.atlassian.com/git/tutorials/>

<http://www.le-fab-lab.com/memo-git.html>

<http://www.le-fab-lab.com/git-travail-en-local.html>

<http://www.le-fab-lab.com/git-depot-distant.html>

<https://git-scm.com/>

# Principales commandes

## Commandes de base

```
$ git status; git log; git diff  
$ git add; git commit
```

## Commandes de branches

```
$ git branch  
$ git checkout  
$ git merge
```

## Commandes de partage

```
$ git push  
$ git fetch  
$ git pull
```

# Configurer git

```
$ git config --global user.name "Anne Cadiou"
$ git config --global user.email "anne.cadiou@ec-lyon.fr"
```

```
$ git config --global core.editor vim
$ git config --global diff.external vimdiff
$ git config --global color.ui true
$ git config --global merge.tool vimdiff
...
$ git config --list
```

Stocké dans

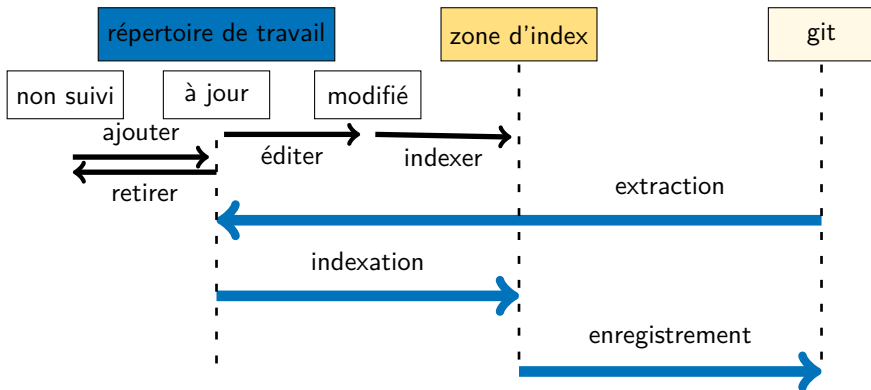
```
$ ls ~/.gitconfig
```

## Exemple de configuration

```
[user]
  name = Anne Cadiou
  email = anne.cadiou@ec-lyon.fr
[core]
  editor = vim
[color]
  ui = true
[merge]
  tool = vimdiff
[alias]
  lg = log --pretty=format:@"%h%x09%an%x09%ad%x09%s"
```

# Principe de base

## Cycle de vie d'un fichier



# Table des matières

1 Contexte

2 Systèmes de gestion de versions

3 Git

**4 Exemple local**

5 Exemple collaboratif

6 Résumé



# Projet de développement local

## Répertoire du projet

```
cadiou@local:~$ cd demo
cadiou@local:~/demo$ ls
logocalcul.png  projet.tex
```

## Initialiser le dépôt

```
cadiou@local:~/demo$ git init
```

# Projet de développement local

## Répertoire du projet

```
cadiou@local:~$ cd demo
cadiou@local:~/demo$ ls
logocalcul.png  projet.tex
```

## Initialiser le dépôt

```
cadiou@local:~/demo$ git init

Initialized empty Git repository in
/home/cadiou/demo/.git/
```

# État du projet

```
cadiou@local:~/demo$ git status
```

# État du projet

```
cadiou@local:~/demo$ git status
```

```
On branch master
```

```
Initial commit
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be  
committed)
```

```
logocalcul.png
```

```
projet.tex
```

```
nothing added to commit but untracked files present (use "  
git add" to track)
```

# Modification du projet

Ajout des fichiers dans la liste des fichiers à suivre

```
cadiou@local:~/demo$ git add -all  
cadiou@local:~/demo$ git status
```

# Modification du projet

Ajout des fichiers dans la liste des fichiers à suivre

```
cadiou@local:~/demo$ git add -all
cadiou@local:~/demo$ git status
```

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: logocalcul.png

new file: projet.tex

# Révision les fichiers

Confier une révision au dépôt

```
cadiou@local:~/demo$ git commit -m "dépôt initial"
```

Consulter le journal de bord

```
cadiou@local:~/demo$ git log
```

# Révision les fichiers

Confier une révision au dépôt

```
cadiou@local:~/demo$ git commit -m "dépôt initial"

[master (root-commit) 35dbd34] dépôt initial
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 logocalcul.png
 create mode 100644 projet.tex
```

Consulter le journal de bord

```
cadiou@local:~/demo$ git log
```



# Révision les fichiers

Confier une révision au dépôt

```
cadiou@local:~/demo$ git commit -m "dépôt initial"

[master (root-commit) 35dbd34] dépôt initial
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 logocalcul.png
 create mode 100644 projet.tex
```

Consulter le journal de bord

```
cadiou@local:~/demo$ git log

commit 35dbd347c4264548fe2140bc36a49d5b08416fc9
Author: Anne Cadiou <anne.cadiou@ec-lyon.fr>
Date: Thu Jan 14 15:18:41 2016 +0100

    dépôt initial
```

# État du suivi

```
cadiou@local:~/demo$ git status
```

Après modification d'un fichier

# État du suivi

```
cadiou@local:~/demo$ git status
```

```
On branch master
```

```
nothing to commit, working directory clean
```

Après modification d'un fichier

# État du suivi

```
cadiou@local:~/demo$ git status
```

```
On branch master
```

```
nothing to commit, working directory clean
```

Après modification d'un fichier

```
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:   projet.tex
```

```
no changes added to commit (use "git add" and/or "git commit  
-a")
```

# Enregistrement des modifications

```
cadiou@local:~/demo$ git commit -m "ajout texte" projet.tex
```

```
cadiou@local:~/demo$ git log
```

# Enregistrement des modifications

```
cadiou@local:~/demo$ git commit -m "ajout texte" projet.tex
```

```
[master 9002ccb] ajout texte  
1 file changed, 2 insertions(+)
```

```
cadiou@local:~/demo$ git log
```

# Enregistrement des modifications

```
cadiou@local:~/demo$ git commit -m "ajout texte" projet.tex
```

```
[master 9002ccb] ajout texte  
1 file changed, 2 insertions(+)
```

```
cadiou@local:~/demo$ git log
```

```
commit 9002ccbab0a0cca2941cec9ee3182defe787a4b7  
Author: Anne Cadiou <anne.cadiou@ec-lyon.fr>  
Date: Thu Jan 14 17:38:04 2016 +0100
```

```
    ajout texte
```

```
commit 35dbd347c4264548fe2140bc36a49d5b08416fc9  
Author: Anne Cadiou <anne.cadiou@ec-lyon.fr>  
Date: Thu Jan 14 15:18:41 2016 +0100
```

```
    dépôt initial
```

# Comparaison entre les révisions

```
cadiou@local:~/demo$ git diff 9002ccb 35dbd34
```

```
cadiou@local:~/demo$ git diff HEAD 35dbd34
```

```
cadiou@local:~/demo$ git difftool HEAD 35dbd34
```



# Comparaison entre les révisions

```
cadiou@local:~/demo$ git diff 9002ccb 35dbd34
```

```
diff --git a/projet.tex b/projet.tex
index c50c265..e69de29 100644
--- a/projet.tex
+++ b/projet.tex
@@ -1,2 +0,0 @@
-% Ajout d'un texte
-
```

```
cadiou@local:~/demo$ git diff HEAD 35dbd34
```

```
cadiou@local:~/demo$ git difftool HEAD 35dbd34
```

# Emploi de raccourcis

## Modification et ajout de fichiers

```
cadiou@local:~/demo$ git status -s
```

```
cadiou@local:~/demo$ git add README.txt  
cadiou@local:~/demo$ git status -s
```

# Emploi de raccourcis

## Modification et ajout de fichiers

```
cadiou@local:~/demo$ git status -s
```

```
M projet.tex
?? README.txt
```

```
cadiou@local:~/demo$ git add README.txt
cadiou@local:~/demo$ git status -s
```

# Emploi de raccourcis

## Modification et ajout de fichiers

```
cadiou@local:~/demo$ git status -s
```

```
M projet.tex  
?? README.txt
```

```
cadiou@local:~/demo$ git add README.txt  
cadiou@local:~/demo$ git status -s
```

```
A README.txt  
M projet.tex
```

# Journal de bord

```
cadiou@local:~/demo$ git commit -m "ajout titre" -a
```

```
cadiou@local:~/demo$ git log --oneline
```

# Journal de bord

```
cadiou@local:~/demo$ git commit -m "ajout titre" -a
```

```
[master 145b9b5] ajout titre  
2 files changed, 2 insertions(+)  
create mode 100644 README.txt
```

```
cadiou@local:~/demo$ git log --oneline
```

# Journal de bord

```
cadiou@local:~/demo$ git commit -m "ajout titre" -a
```

```
[master 145b9b5] ajout titre  
2 files changed, 2 insertions(+)  
create mode 100644 README.txt
```

```
cadiou@local:~/demo$ git log --oneline
```

```
145b9b5 ajout titre  
9002ccb ajout texte  
35dbd34 dépôt initial
```

# Alias

```
cadiou@local:~/demo$ git log --pretty=format:@"%h%x09%an%x09%ad%x09%s\"
```

```
145b9b5 Anne Cadiou      Thu Jan 14 18:23:21 2016 +0100
    ajout titre
9002ccb Anne Cadiou      Thu Jan 14 17:38:04 2016 +0100
    ajout texte
35dbd34 Anne Cadiou      Thu Jan 14 15:18:41 2016 +0100
    dépôt initial
```

```
git config --global alias.lg ' log --pretty=format:@"%h%x09%an%x09%ad%x09%s"'
```



## Retour en arrière

```
145b9b5 ajout titre
9002ccb ajout texte
35dbd34 dépôt initial
```

```
cadiou@local:~/demo$ git checkout 9002ccb projet.tex
cadiou@local:~/demo$ git status -s
```

```
cadiou@local:~/demo$ git commit -m "retour vers 9002ccb" -a
```

## Retour en arrière

```
145b9b5 ajout titre
9002ccb ajout texte
35dbd34 dépôt initial
```

```
cadiou@local:~/demo$ git checkout 9002ccb projet.tex
cadiou@local:~/demo$ git status -s
```

```
M  projet.tex
```

Le fichier est comme dans 9002ccb

```
cadiou@local:~/demo$ git commit -m "retour vers 9002ccb" -a
```

## Retour en arrière

```
145b9b5 ajout titre
9002ccb ajout texte
35dbd34 dépôt initial
```

```
cadiou@local:~/demo$ git checkout 9002ccb projet.tex
cadiou@local:~/demo$ git status -s
```

```
M  projet.tex
```

Le fichier est comme dans 9002ccb

```
cadiou@local:~/demo$ git commit -m "retour vers 9002ccb" -a

[master b1b2c39] retour vers 9002ccb
1 file changed, 1 deletion(-)
```

# Annulation

```
cadiou@local:~/demo$ git revert HEAD
```

```
cadiou@local:~/demo$ git log --oneline
```

Le fichier est comme [145b9b5](#)

# Annulation

```
cadiou@local:~/demo$ git revert HEAD
```

```
[master e03841b] Revert "retour vers 9002ccb"  
1 file changed, 1 insertion(+)
```

```
cadiou@local:~/demo$ git log --oneline
```

Le fichier est comme [145b9b5](#)

# Annulation

```
cadiou@local:~/demo$ git revert HEAD
```

```
[master e03841b] Revert "retour vers 9002ccb"  
1 file changed, 1 insertion(+)
```

```
cadiou@local:~/demo$ git log --oneline
```

```
e03841b Revert "retour vers 9002ccb"  
b1b2c39 retour vers 9002ccb  
145b9b5 ajout titre  
9002ccb ajout texte  
35dbd34 dépôt initial
```

Le fichier est comme **145b9b5**

# Branches

## Branches

```
cadiou@local:~/demo$ git branch  
  
* master
```

# Configuration

Ajouter à son environnement (.bashrc)

```
function parse_git_branch {
git branch --no-color 2> /dev/null | sed -e '/^[^*]/d' -e 's
/* \(.*\)/\1/'
}
function git-track {
CURRENT_BRANCH=$(parse_git_branch)
git-config branch.$CURRENT_BRANCH.remote $1
git-config branch.$CURRENT_BRANCH.merge refs/heads/
$CURRENT_BRANCH
}
function parse_git_branch_and_add_brackets {
git branch --no-color 2> /dev/null | sed -e '/^[^*]/d' -e 's
/* \(.*\)/\ \[\1\]/'
}

PS1="\u@h:\$(parse_git_branch_and_add_brackets) \w\$ "
```



# Création d'une nouvelle branche

```
cadiou@moon: [master] ~/demo$ git branch -a
* master

cadiou@moon: [master] ~/demo$ git branch develop
cadiou@moon: [master] ~/demo$ git branch
develop
* master

cadiou@moon: [master] ~/demo$ git checkout develop
* develop
master
cadiou@moon: [develop] ~/demo$
```

Tout en une seule commande

```
cadiou@moon: [master] ~/demo$ git checkout -b develop
Switched to a new branch 'develop'
cadiou@moon: [develop] ~/demo$
```

# Fusion

```
cadiou@moon: [master] ~/demo$ git merge develop
cadiou@moon: [master] ~/demo$ git log --oneline --graph --
color
```

# Fusion

```
cadiou@moon: [master] ~/demo$ git merge develop
cadiou@moon: [master] ~/demo$ git log --oneline --graph --
color

*    2136676  après fusion
|\
| * 457c155  ajout références
| * 4ba2d09  ajout bibliographie
* | fc6cc31  après branche
|/
* e03841b Revert "retour vers 9002ccb"
* b1b2c39 retour vers 9002ccb
* 145b9b5 ajout titre
* 9002ccb ajout texte
* 35dbd34 dépôt initial
```

# Outils d'arbitrage

```
cadiou@moon: [master] ~/demo$ git difftool 9002ccb 145b9b5
    projet.tex
cadiou@moon: [master] ~/demo$ git mergetool develop
cadiou@moon: [master] ~/demo$ git mergetool -t meld
cadiou@moon: [master] ~/demo$ git config --global merge.tool
    kdiff3
cadiou@moon: [master] ~/demo$ git config --global merge.tool
    vimdiff
```

dans .gitconfig

```
[merge]
tool = vimdiff
```

# Destruction des branches

```
cadiou@moon: [master] ~/demo$ git branch -d develop  
Deleted branch develop (was 387526b).
```

```
cadiou@moon: [master] ~/demo$ git branch -a  
* master
```

# Structure locale

```
cadiou@moon: [master] ~/demo$ ls -a
```

# Structure locale

```
cadiou@moon: [master] ~/demo$ ls -a  
.  
..  
bib.bib  
.git  
logocalcul.png  
projet.tex  
README.txt
```

```
cadiou@moon: [master] ~/demo$ cat .git/config
```

# Structure locale

```
cadiou@moon: [master] ~/demo$ ls -a
```

```
.  ..  bib.bib  .git  logocalcul.png  projet.tex  README.txt
```

```
cadiou@moon: [master] ~/demo$ cat .git/config
```

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
```



# Exclure des fichiers

```
cadiou@moon: [master] ~/demo$ vi .gitignore
```

# Exclure des fichiers

```
cadiou@moon: [master] ~/demo$ vi .gitignore
```

```
.*.swp  
*.out  
*.log  
*.vrb  
*.nav  
*.aux  
*.toc  
*.blg
```

# Table des matières

- 1 Contexte
- 2 Systèmes de gestion de versions
- 3 Git
- 4 Exemple local
- 5 Exemple collaboratif**
- 6 Résumé

# Initialiser le dépôt

Bonne pratique (extension .git)

```
tamia@distant:~$ mkdir demo.git
```

Initialiser le dépôt

```
tamia@distant:~/demo.git$ git init --bare
```

# Initialiser le dépôt

Bonne pratique (extension .git)

```
tamia@distant:~$ mkdir demo.git
```

Initialiser le dépôt

```
tamia@distant:~/demo.git$ git init --bare
```

```
Initialized empty Git repository in /home/tamia/demo.git/
```

# Structure

```
tamia@distant:~/demo.git$ ls
```

```
tamia@distant:~/demo.git$ cat config
```

# Structure

```
tamia@distant:~/demo.git$ ls
```

```
branches  config  description  HEAD  hooks  info  objects
refs
```

```
tamia@distant:~/demo.git$ cat config
```

# Structure

```
tamia@distant:~/demo.git$ ls
```

```
branches  config  description  HEAD  hooks  info  objects
refs
```

```
tamia@distant:~/demo.git$ cat config
```



# Structure

```
tamia@distant:~/demo.git$ ls
```

```
branches  config  description  HEAD  hooks  info  objects
refs
```

```
tamia@distant:~/demo.git$ cat config
```

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = true
```

# Cloner

## Machine locale de Tic

```
tic@pctic:~$ git clone tamia@distant:~/demo.git
```

# Cloner

## Machine locale de Tic

```
tic@pctic:~$ git clone tamia@distant:~/demo.git
```

```
Cloning into 'demo'...
```

```
warning: You appear to have cloned an empty repository.  
done.
```

```
tic@pctic:~$ cd demo
```

```
tic@pctic:~/demo$ ls -a
```

# Cloner

## Machine locale de Tic

```
tic@pctic:~$ git clone tamia@distant:~/demo.git
```

```
Cloning into 'demo'...
```

```
warning: You appear to have cloned an empty repository.  
done.
```

```
tic@pctic:~$ cd demo
```

```
tic@pctic:~/demo$ ls -a
```

```
.  ..  .git
```

# Structure du clone

```
tic@pctic:~/demo$ cat .git/config
```

# Structure du clone

```
tic@pctic:~/demo$ cat .git/config
```

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[remote "origin"]
  url = tamia@distant:demo.git
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
  remote = origin
  merge = refs/heads/master
```

# Échanges avec le dépôt

```
tic@pctic:~/demo$ vi fichier.tex
tic@pctic:~/demo$ git add fichier.tex
tic@pctic:~/demo$ git commit -m "Ajout de fichier" -a
```

# Échanges avec le dépôt

```
tic@pctic:~/demo$ vi fichier.tex
tic@pctic:~/demo$ git add fichier.tex
tic@pctic:~/demo$ git commit -m "Ajout de fichier" -a
```

```
[master (root-commit) 4ba89d5] Ajout de fichier
1 file changed, 2 insertions(+)
create mode 100644 fichier.tex
```



# Échanges avec le dépôt

```
tic@pctic:~/demo$ vi fichier.tex
tic@pctic:~/demo$ git add fichier.tex
tic@pctic:~/demo$ git commit -m "Ajout de fichier" -a
```

```
[master (root-commit) 4ba89d5] Ajout de fichier
 1 file changed, 2 insertions(+)
 create mode 100644 fichier.tex
```

```
tic@pctic:~/demo$ git push origin master
```

# Échanges avec le dépôt

```
tic@pctic:~/demo$ vi fichier.tex
tic@pctic:~/demo$ git add fichier.tex
tic@pctic:~/demo$ git commit -m "Ajout de fichier" -a
```

```
[master (root-commit) 4ba89d5] Ajout de fichier
 1 file changed, 2 insertions(+)
 create mode 100644 fichier.tex
```

```
tic@pctic:~/demo$ git push origin master
```

```
Counting objects: 3, done.
Writing objects: 100% (3/3), 240 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To tamia@distant:TEST/demo.git
 * [new branch]      master -> master
```

## Tac veut participer...

```
tac@pctac:~$ git clone tamia@distant:~/demo.git
```

# Tac veut participer...

```
tac@pctac:~$ git clone tamia@distant:~/demo.git
```

```
Cloning into 'demo'...
```

```
remote: Counting objects: 3, done.
```

```
remote: Total 3 (delta 0), reused 0 (delta 0)
```

```
Receiving objects: 100% (3/3), done.
```

```
Checking connectivity... done.
```

# Tac veut participer...

```
tac@pctac:~$ git clone tamia@distant:~/demo.git
```

```
Cloning into 'demo'...
```

```
remote: Counting objects: 3, done.
```

```
remote: Total 3 (delta 0), reused 0 (delta 0)
```

```
Receiving objects: 100% (3/3), done.
```

```
Checking connectivity... done.
```

```
tac@pctac:~$ cd demo
```

```
tac@pctac:~/demo$ ls
```

# Tac veut participer...

```
tac@pctac:~$ git clone tamia@distant:~/demo.git
```

```
Cloning into 'demo'...
```

```
remote: Counting objects: 3, done.
```

```
remote: Total 3 (delta 0), reused 0 (delta 0)
```

```
Receiving objects: 100% (3/3), done.
```

```
Checking connectivity... done.
```

```
tac@pctac:~$ cd demo
```

```
tac@pctac:~/demo$ ls
```

```
fichier.tex
```

# Tac veut participer...

```
tac@pctac:~$ git clone tamia@distant:~/demo.git
```

```
Cloning into 'demo'...
```

```
remote: Counting objects: 3, done.
```

```
remote: Total 3 (delta 0), reused 0 (delta 0)
```

```
Receiving objects: 100% (3/3), done.
```

```
Checking connectivity... done.
```

```
tac@pctac:~$ cd demo
```

```
tac@pctac:~/demo$ ls
```

```
fichier.tex
```

```
tac@pctac:~/demo$ git lg (alias de log --pretty=format:@"%h%  
x09%an%x09%ad%x09%s\"")
```

# Tac veut participer...

```
tac@pctac:~$ git clone tamia@distant:~/demo.git
```

```
Cloning into 'demo'...
```

```
remote: Counting objects: 3, done.
```

```
remote: Total 3 (delta 0), reused 0 (delta 0)
```

```
Receiving objects: 100% (3/3), done.
```

```
Checking connectivity... done.
```

```
tac@pctac:~$ cd demo
```

```
tac@pctac:~/demo$ ls
```

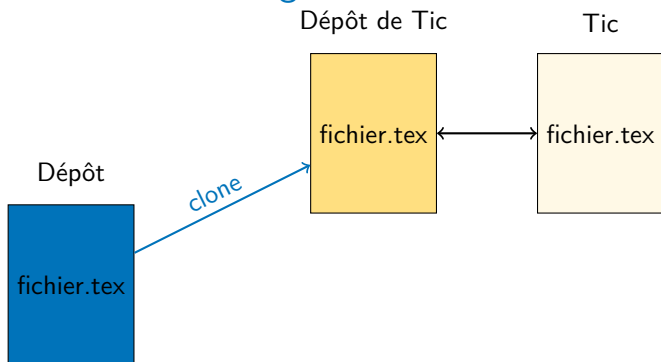
```
fichier.tex
```

```
tac@pctac:~/demo$ git lg (alias de log --pretty=format:@"%h%  
x09%an%x09%ad%x09%s\\")
```

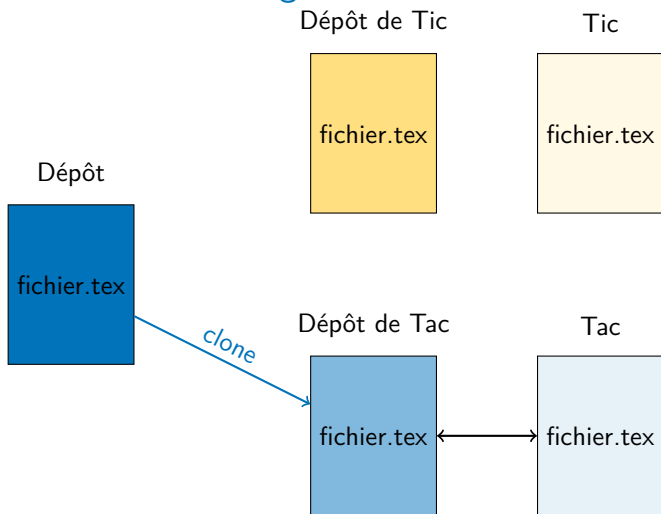
```
ba89d5 Tic Squirrel      Sun Jan 17 18:10:43 2016 +0100  Ajout  
de fichier
```



# Collaboration en image



# Collaboration en image



# Collaboration en image

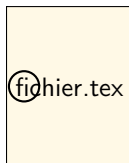
Dépôt



Dépôt de Tic



Tic



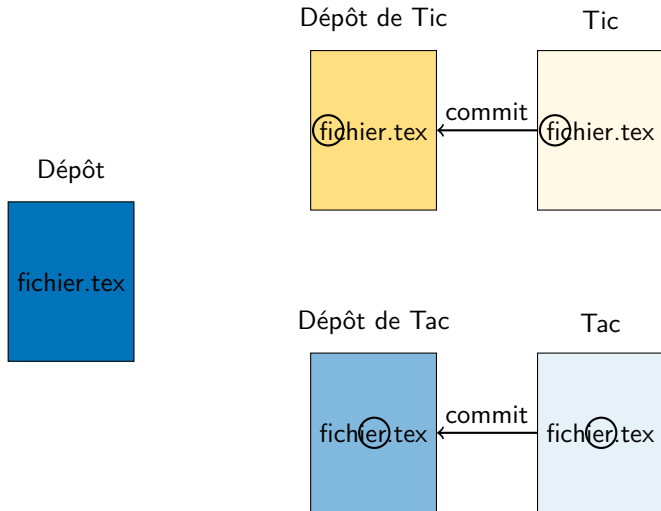
Dépôt de Tac



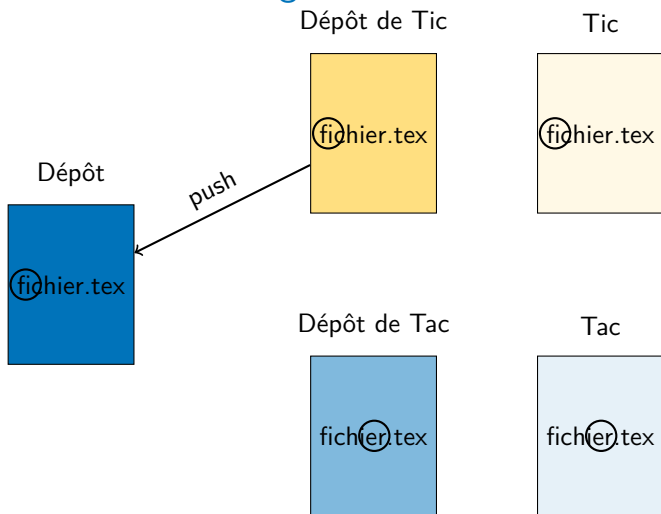
Tac



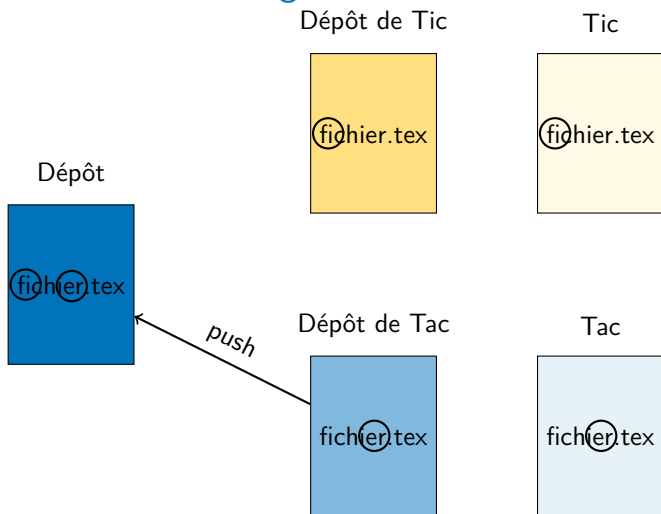
# Collaboration en image



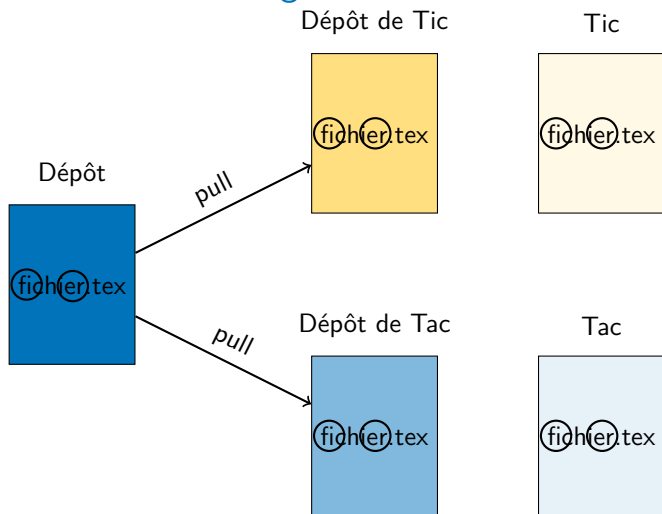
# Collaboration en image



# Collaboration en image



# Collaboration en image



# Travail simultané de Tic et Tac

```
tac@pctac:~/demo$ vi fichier.tex  
tac@pctac:~/demo$ git commit -m 'modification de Tac' -a
```

```
tic@pctic:~/demo$ vi fichier.tex  
tic@pctic:~/demo$ git commit -m 'modification de Tic' -a
```



# Travail simultané de Tic et Tac

```
tac@pctac:~/demo$ vi fichier.tex
tac@pctac:~/demo$ git commit -m 'modification de Tac' -a
```

```
[master 94ae224] modification de Tac
1 file changed, 1 insertion(+)
```

```
tic@pctic:~/demo$ vi fichier.tex
tic@pctic:~/demo$ git commit -m 'modification de Tic' -a
```

# Travail simultané de Tic et Tac

```
tac@pctac:~/demo$ vi fichier.tex
tac@pctac:~/demo$ git commit -m 'modification de Tac' -a

[master 94ae224] modification de Tac
1 file changed, 1 insertion(+)
```

```
tic@pctic:~/demo$ vi fichier.tex
tic@pctic:~/demo$ git commit -m 'modification de Tic' -a
```

# Travail simultané de Tic et Tac

```
tac@pctac:~/demo$ vi fichier.tex
tac@pctac:~/demo$ git commit -m 'modification de Tac' -a

[master 94ae224] modification de Tac
1 file changed, 1 insertion(+)
```

```
tic@pctic:~/demo$ vi fichier.tex
tic@pctic:~/demo$ git commit -m 'modification de Tic' -a

[master 52d0dce] modification de Tic
1 file changed, 1 insertion(+)
```

# Tac dépose ses modifications

```
tac@pctac:~/demo$ git push origin master
```

# Tac dépose ses modifications

```
tac@pctac:~/demo$ git push origin master
```

```
Counting objects: 5, done.
```

```
Delta compression using up to 4 threads.
```

```
Compressing objects: 100% (2/2), done.
```

```
Writing objects: 100% (3/3), 299 bytes | 0 bytes/s, done.
```

```
Total 3 (delta 0), reused 0 (delta 0)
```

```
To tamia@distant:~/demo.git
```

```
4ba89d5..94ae224 master -> master
```

# Tic échoue

```
tic@pctic:~/demo$ git push origin master
```

# Tic échoue

```
tic@pctic:~/demo$ git push origin master
```

```
To tamia@distant:~/demo.git
```

```
! [rejected]          master -> master (fetch first)
```

```
error: failed to push some refs to 'tamia@distant:~/demo.git',
```

```
hint: Updates were rejected because the remote contains work  
      that you do
```

```
hint: not have locally. This is usually caused by another  
      repository pushing
```

```
hint: to the same ref. You may want to first integrate the  
      remote changes
```

```
hint: (e.g., 'git pull ...') before pushing again.
```

```
hint: See the 'Note about fast-forwards' in 'git push --help  
      ' for details.
```

# Conflit et résolution

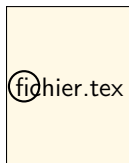
Dépôt



Dépôt de Tic



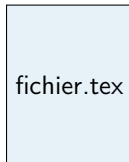
Tic



Dépôt de Tac



Tac





# Conflit et résolution

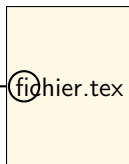
Dépôt



Dépôt de Tic



Tic



commit



Dépôt de Tac



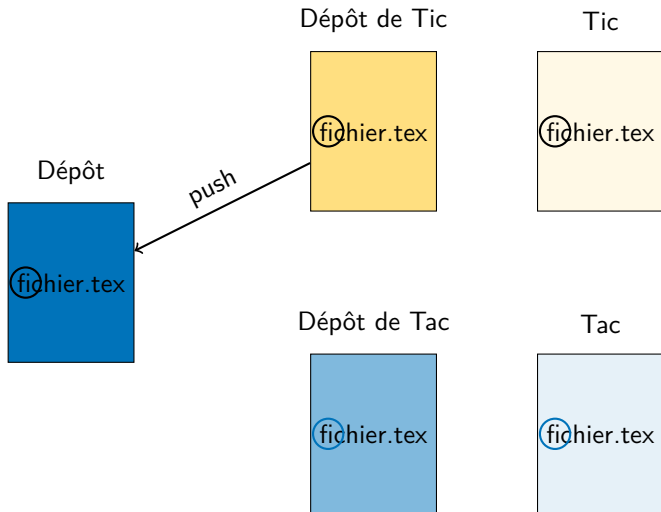
Tac



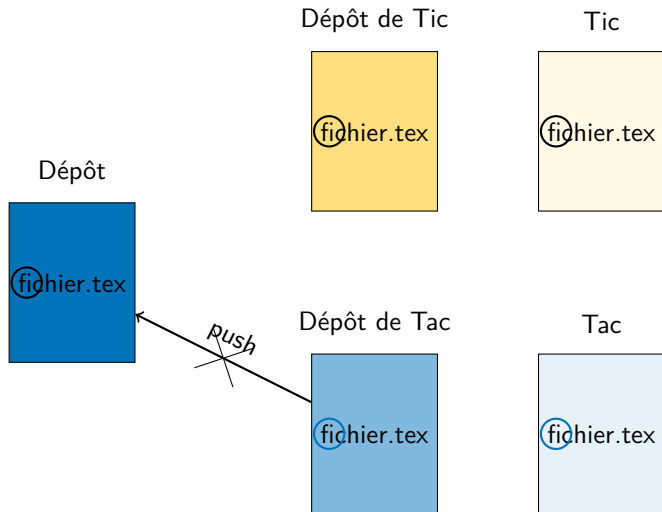
commit



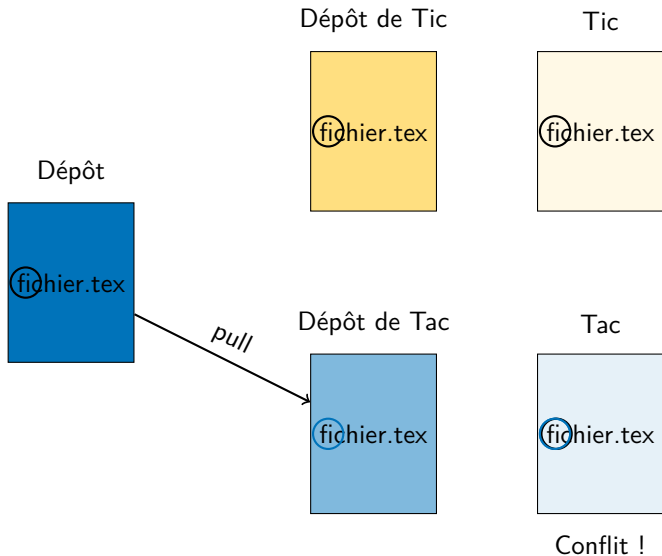
# Conflit et résolution



# Conflit et résolution



# Conflit et résolution



# Conflit et résolution

Dépôt



Dépôt de Tic



Tic



Dépôt de Tac



Tac



arbitrage/fusion effective

# Conflit et résolution



Dépôt de Tic



Tic



Dépôt de Tac



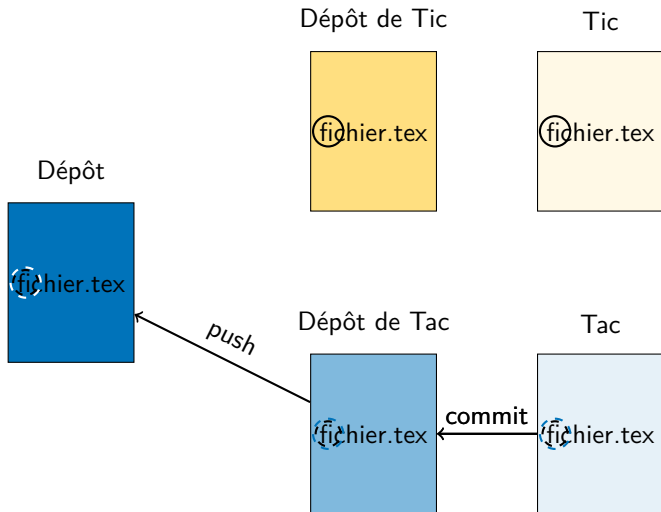
Tac



commit



# Conflit et résolution



# Gestion des conflits entre Tic et Tac

```
tic@pctic:~/demo$ git pull
```



# Gestion des conflits entre Tic et Tac

```
tic@pctic:~/demo$ git pull

remote: Counting objects: 5, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From p2chpd-visu7.univ-lyon1.fr:TEST/demo
   4ba89d5..94ae224  master      -> origin/master
Auto-merging fichier.tex
CONFLICT (content): Merge conflict in fichier.tex
Automatic merge failed; fix conflicts and then commit the
result.
```

# Fusion manuelle

```
tic@pctic:~/demo$ vi fichier.tex
```

# Fusion manuelle

```
tic@pctic:~/demo$ vi fichier.tex
```

```
1 %commentaire de Tic
2 <<<<<< HEAD
3 %commentaire de Tic
4 =====
5 %commentaire de Tac au même endroit
6 >>>>>> 94ae22421e3bf6d329bdf1e13bcbc77a850ed258
```

# Fusion avec un outil

```
tic@pctic:~/demo$ git mergetool fichier.tex
tic@pctic:~/demo$ git checkout --ours fichier.tex
tic@pctic:~/demo$ git checkout --theirs fichier.tex
tic@pctic:~/demo$ git commit -m "fusion avec theirs"
```

## Fusion avec un outil

```
tic@pctic:~/demo$ git mergetool fichier.tex
tic@pctic:~/demo$ git checkout --ours fichier.tex
tic@pctic:~/demo$ git checkout --theirs fichier.tex
tic@pctic:~/demo$ git commit -m "fusion avec theirs"
```

Normal merge conflict for 'fichier.tex':

{local}: modified file

{remote}: modified file

Hit return to start merge resolution tool (vimdiff):

4 files to edit

# Fusion en instance

```
tic@pctic:~/demo$ git status
```

# Fusion en instance

```
tic@pctic:~/demo$ git status
```

```
On branch master
```

```
Your branch and 'origin/master' have diverged,  
and have 1 and 1 different commit each, respectively.  
(use "git pull" to merge the remote branch into yours)
```

```
All conflicts fixed but you are still merging.  
(use "git commit" to conclude merge)
```

```
Changes to be committed:
```

```
modified:   fichier.tex
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be  
committed)
```

```
fichier.tex.orig
```

## Fusion effective

```
tic@pctic:~/demo$ git commit -m "fusion avec theirs"
[master c198c63] fusion avec theirs
tic@pctic:~/demo$ git push origin master
```

```
tic@pctic:~/demo$ git pull origin master
```



## Fusion effective

```
tic@pctic:~/demo$ git commit -m "fusion avec theirs"
[master c198c63] fusion avec theirs
tic@pctic:~/demo$ git push origin master

Counting objects: 10, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 556 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To tamia@distant:~/demo.git
   94ae224..c198c63  master -> master
```

```
tic@pctic:~/demo$ git pull origin master
```

## Fusion effective

```
tic@pctic:~/demo$ git commit -m "fusion avec theirs"
[master c198c63] fusion avec theirs
tic@pctic:~/demo$ git push origin master
```

```
Counting objects: 10, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 556 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To tamia@distant:~/demo.git
   94ae224..c198c63  master -> master
```

```
tic@pctic:~/demo$ git pull origin master
```

```
From distant:~/demo
```

```
* branch                master      -> FETCH_HEAD
```

```
Already up-to-date.
```

## Bonne pratique : utiliser les branches

```
tic@pctic:~$ git clone tamia@distant:~/demo.git
tic@pctic:~$ git checkout -b ticbranche
```

... et utiliser les différentes commandes

## Bonne pratique : utiliser les branches

```
tic@pctic:~$ git clone tamia@distant:~/demo.git
tic@pctic:~$ git checkout -b ticbranche
```

```
git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
```

```
tac@pctac:~$ git clone tamia@distant:~/demo.git
tac@pctac:~$ git checkout -b tacbranche
```

... et utiliser les différentes commandes

# Table des matières

- 1 Contexte
- 2 Systèmes de gestion de versions
- 3 Git
- 4 Exemple local
- 5 Exemple collaboratif
- 6 Résumé**

# Résumé

- Travailler sur des clones de dépôt (bare)
- Travailler sur des branches
- Gestion des conflits
  - lors d'une commande git pull  
fusionner les fichiers (merge)
  - lors d'une commande git push  
récupérer (pull) la dernière modification et  
fusionner les fichiers (merge)

Merci à **Geoffroy Berret** pour sa relecture,

à **Claire Mouton** et **Vincent Miele** pour l'intégration dans leurs interventions sur les bonnes pratiques,

à **Bastien Di Pierro** pour l'organisation.