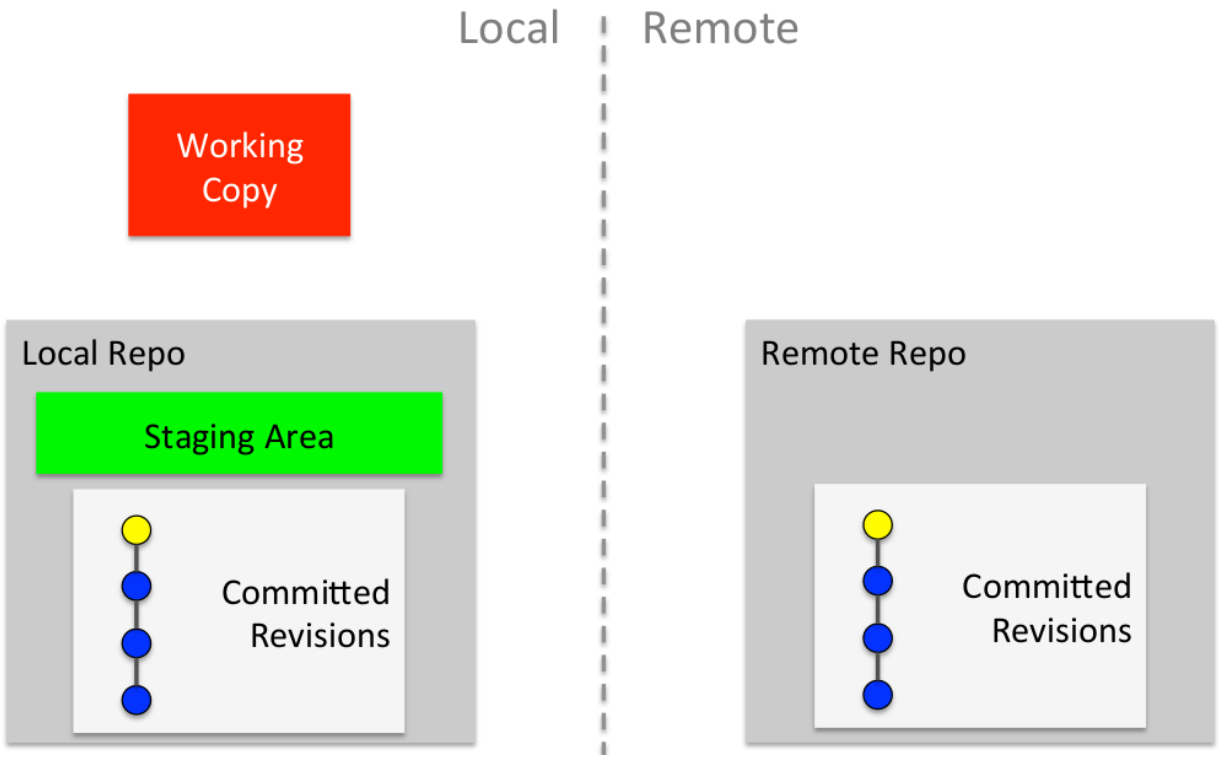
Introduction to Git

Dr. Xiaodong LIU

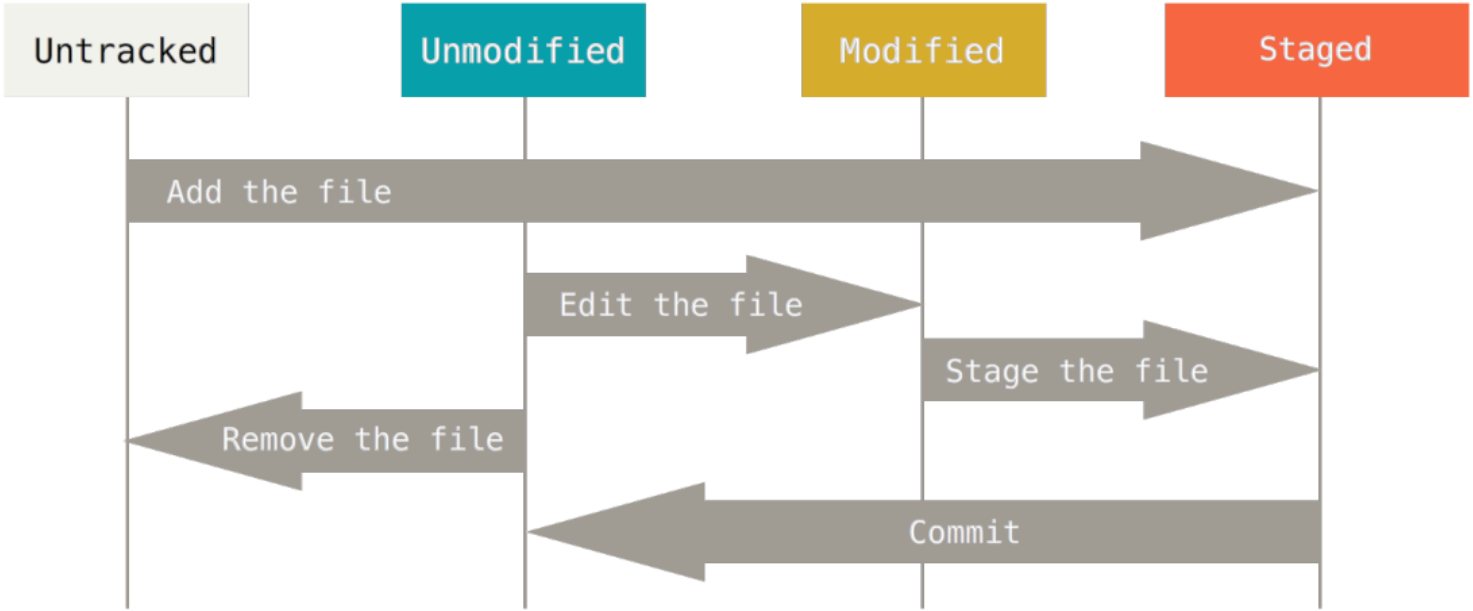
GeM cnrs Nantes Université CENTRALE NANTES

What is git?



Picture from DAVID PARSONS (Inria) - INTRODUCTION TO GIT

What is git?



Picture from DAVID PARSONS (Inria) - INTRODUCTION TO GIT

Why git?

version 1 : by me

version 2 : by c1

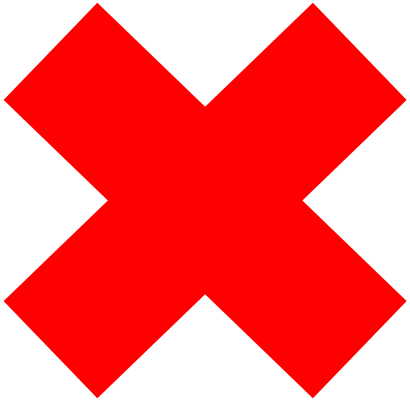
version 3 : by c2

version 4 : by c3

version 5 : by c4

...

Final version



Parallel working
Histrory tracking

1

Configuration

Basic local configuration

git config --global user.name “Jean-Jean“

git config --global user.email jean.jean@ec-nantes.fr

Editor

git config –global core.editor vim (The default Git editor is **Nano**)

Git Alias

git config --global alias.co checkout

git config --global alias.br branch

git config --global alias.ci commit

git config --global alias.st status

git config --global alias.loggraph ‘log --all --decorate --oneline --graph’

git config --global alias.log1 ‘log --oneline’

error ??? git config --global --edit

Github

local username and email adress shall be exactly the same as online one
Configue the ssh key locally (generation and edit config file)
Put your ssh key in Github

```
user.name=Xiao-dong-LIU
user.email=xiaodong.liu@cnrs.fr
```

```
xiliu2016@pc-gem120:~/.ssh$ ls
config   git_gem.pub  id_ed25519.pub  id_fixe.pub  id_github.pub  id_rsa_ccrt      id_rsa.pub
git_gem  id_ed25519   id_fixe         id_github    id_rsa         id_rsa_ccrt.pub  known_hosts
```

config file in .ssh

```
# Private GitLab instance
Host github
   HostName github.com
   User Xiao-dong-LIU
   IdentityFile ~/.ssh/id_github
```

koda

Koda : Forge des laboratoires

Gitlab

Gitlab est une plate-forme de développement collaboratif permettant de gérer le code source avec git. Ce service est proposé aux laboratoires du CNRS. Il est administré par la Direction des Systèmes d'Information du CNRS.

Vous pouvez signaler un incident ou nous contacter au travers du portail e-dem.

L'authentification à la forge se fait uniquement via Janus. Un compte dans le SI du CNRS est donc nécessaire pour pouvoir y accéder.

Documentation | Mentions légales

By signing in you accept the Terms of Use and acknowledge the Privacy Policy and Cookie Policy.

Janus

Remember me

koda

Site : https://src.koda.cnrs.fr/
Configue the PAT (Personal Access Tokens)

Linux case (if gpg does not work)

vim ~/.netrc

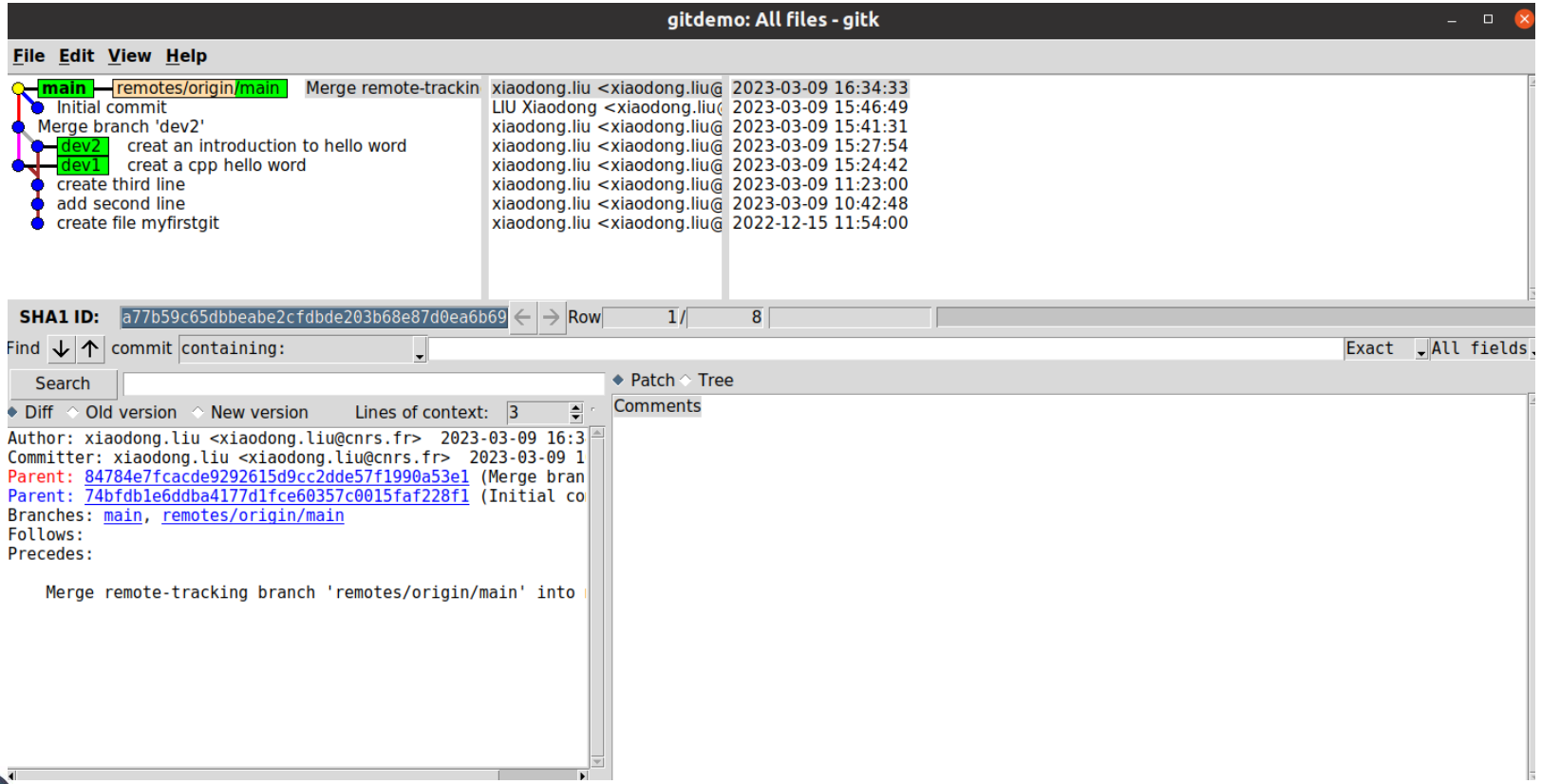
machine src.koda.cnrs.fr
protocol https
login oauth2 (not important for koda)
password Your_PAT

chmod 600 ~/.netrc

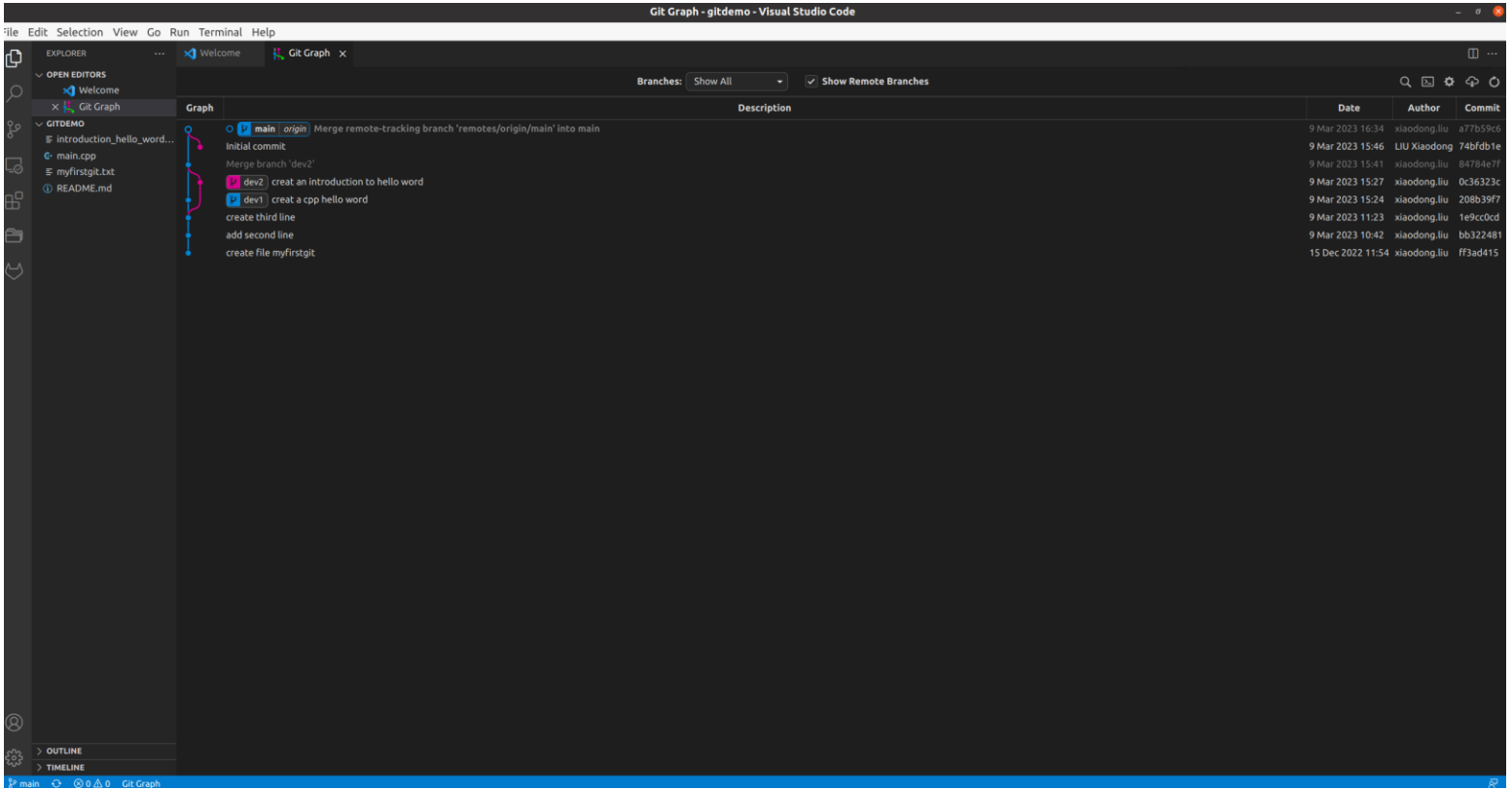
2

Graphical User Interfaces

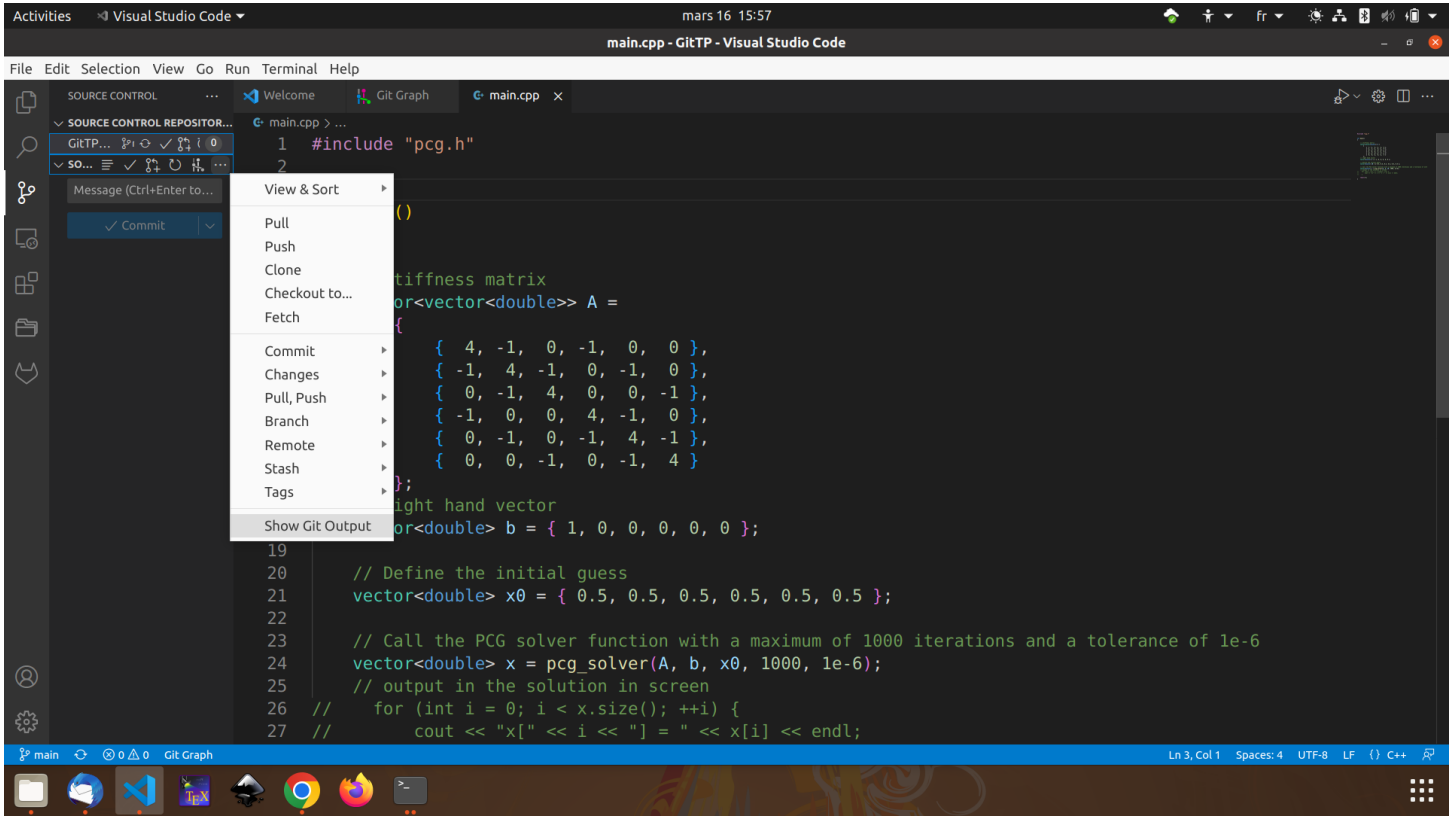
gitk


gitdemo: All files - gitk
File Edit View Help
main
remotes/origin/main
Merge remote-trackin
Initial commit
Merge branch 'dev2'
dev2
creat an introduction to hello word
dev1
creat a cpp hello word
create third line
add second line
create file myfirstgit
xiaodong.liu <xiaodong.liu@ 2023-03-09 16:34:33
LIU Xiaodong <xiaodong.liu(2023-03-09 15:46:49
xiaodong.liu <xiaodong.liu@ 2023-03-09 15:41:31
xiaodong.liu <xiaodong.liu@ 2023-03-09 15:27:54
xiaodong.liu <xiaodong.liu@ 2023-03-09 15:24:42
xiaodong.liu <xiaodong.liu@ 2023-03-09 11:23:00
xiaodong.liu <xiaodong.liu@ 2023-03-09 10:42:48
xiaodong.liu <xiaodong.liu@ 2022-12-15 11:54:00
SHA1 ID: a77b59c65dbbeabe2cfdbde203b68e87d0ea6b69
Row 1/ 8
Find commit containing:
Exact
All fields
Search
Patch
Tree
Diff
Old version
New version
Lines of context: 3
Comments
Author: xiaodong.liu <xiaodong.liu@cnrs.fr> 2023-03-09 16:3
Committer: xiaodong.liu <xiaodong.liu@cnrs.fr> 2023-03-09 1
Parent: 84784e7fcacde9292615d9cc2dde57f1990a53e1 (Merge bran
Parent: 74bfdb1e6ddba4177d1fce60357c0015faf228f1 (Initial co
Branches: main, remotes/origin/main
Follows:
Precedes:
Merge remote-tracking branch 'remotes/origin/main' into

Git Graph in vscode



Git Graph in vscode



Others?

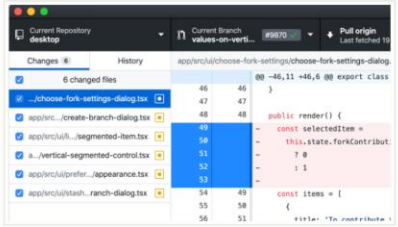
https://git-scm.com/downloads/guis/

GUI Clients

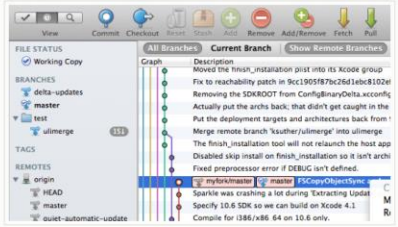
Git comes with built-in GUI tools for committing (git-gui) and browsing (gitk), but there are several third-party tools for users looking for platform-specific experience.

If you want to add another GUI tool to this list, just follow the instructions.

All | Windows | Mac | Linux | Android | iOS

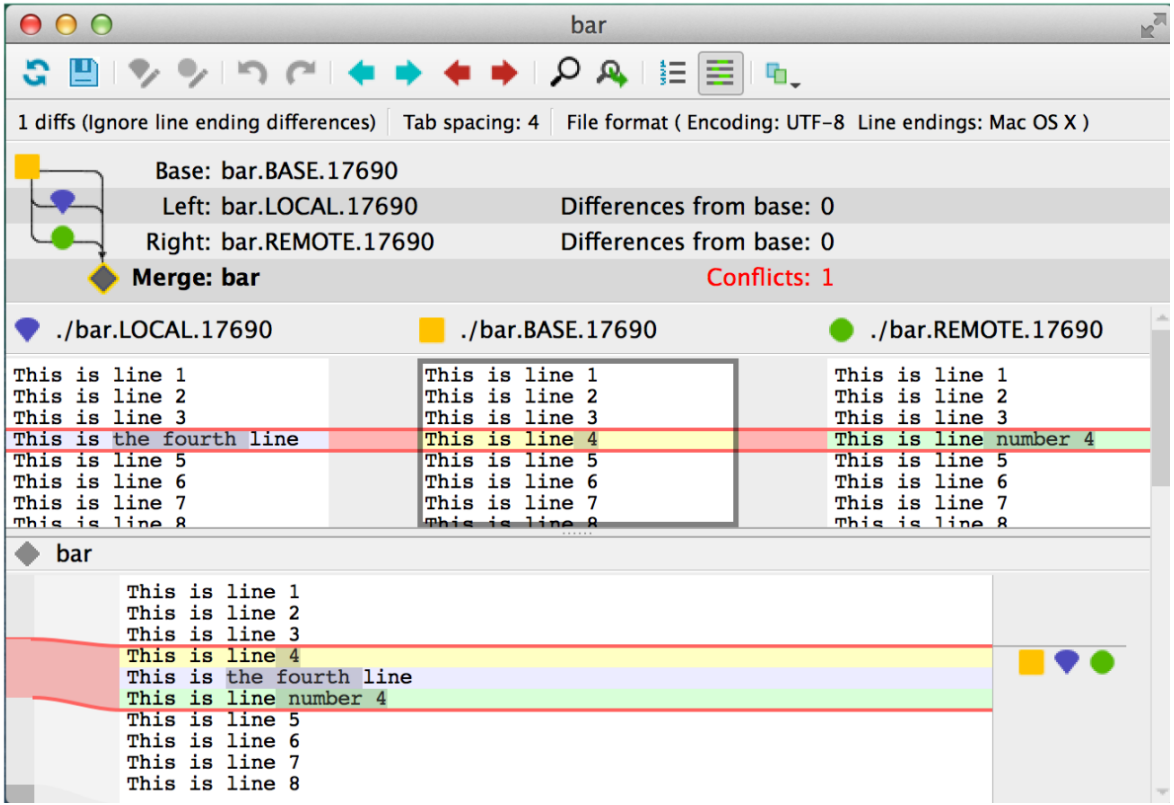


GitHub Desktop
Platforms: Mac, Windows
Price: Free
License: MIT

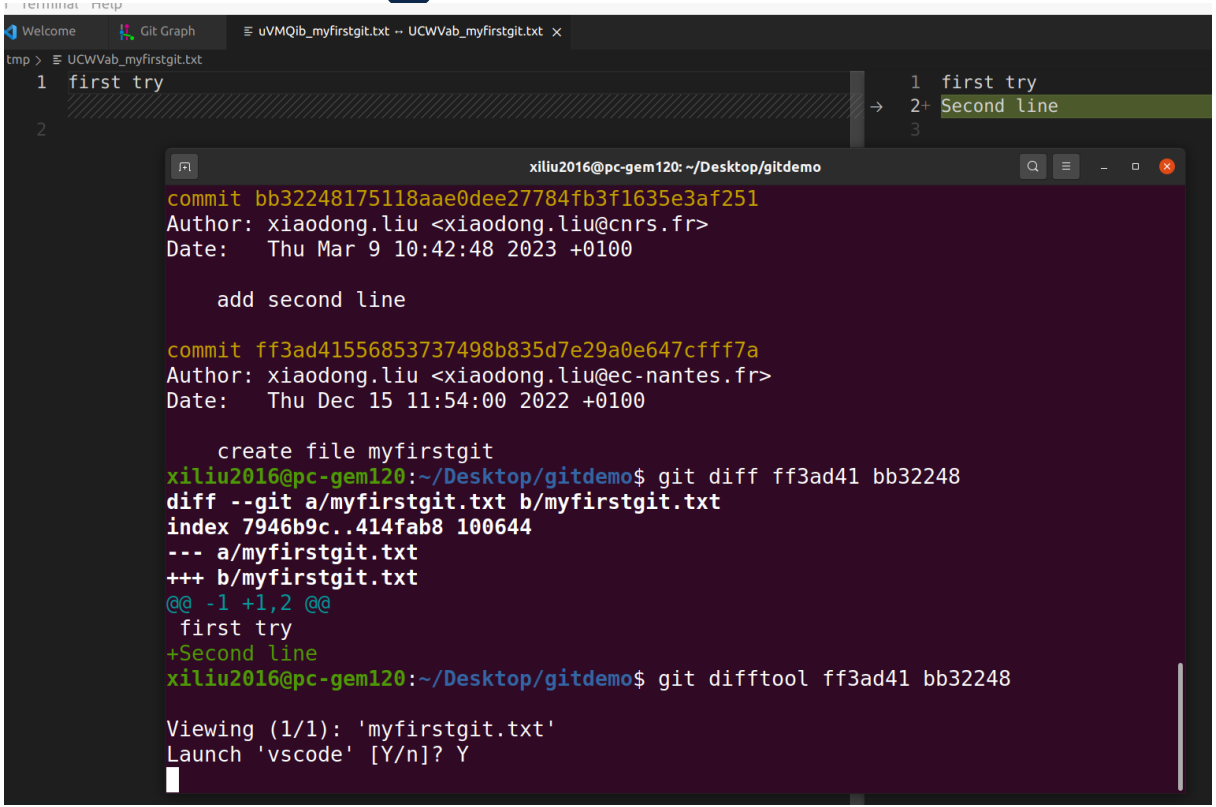


SourceTree
Platforms: Mac, Windows
Price: Free
License: Proprietary

Git diff/merge tool - p4merge



Git diff/merge tool - vscode



Tool configurations : vscode case

diff tool

```
git config --global diff.tool vscode
git config --global difftool.vscode.cmd 'code --wait --diff $LOCAL $REMOTE'
```

merge tool

```
git config --global merge.tool vscode
git config --global mergetool.vscode.cmd 'code --wait $MERGED'
```

check with : git config --global --list

3

Main Commands

First git repository

git init

Initialized empty Git repository in /home/xiliu2016/Desktop/Git/.git/

ls -a

.git

git st / git status

```
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

First commit

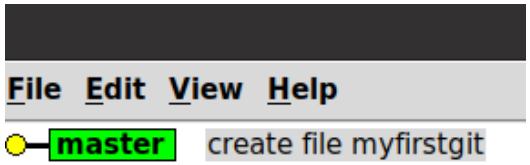
Create a txt file

git add

git commit

logs

```
create file myfirstgit
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   myfirstgit.txt
#
```




```
xiliu2016@pc-str106:~/Desktop/gitdemo$ vim myfirstgit.txt
xiliu2016@pc-str106:~/Desktop/gitdemo$ git st
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        myfirstgit.txt

nothing added to commit but untracked files present (use "git add" to track)
xiliu2016@pc-str106:~/Desktop/gitdemo$ git add .
xiliu2016@pc-str106:~/Desktop/gitdemo$ git st
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   myfirstgit.txt
xiliu2016@pc-str106:~/Desktop/gitdemo$ git commit
[master (root-commit) ff3ad41] create file myfirstgit
 1 file changed, 1 insertion(+)
 create mode 100644 myfirstgit.txt
xiliu2016@pc-str106:~/Desktop/gitdemo$ git st
On branch master
nothing to commit, working tree clean
```

First commit

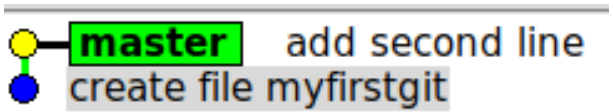
Modify the txt by adding a second line

```
first try
Second line
```

```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ vim myfirstgit.txt
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git st
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed
)
  (use "git restore <file>..." to discard changes in workin
g directory)
        modified:   myfirstgit.txt

no changes added to commit (use "git add" and/or "git commi
t -a")
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git add .
```

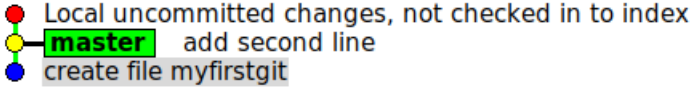
- Commit often
- Clear and informative logs




```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git ci
[master bb32248] add second line
 1 file changed, 1 insertion(+)
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git st
On branch master
nothing to commit, working tree clean
```

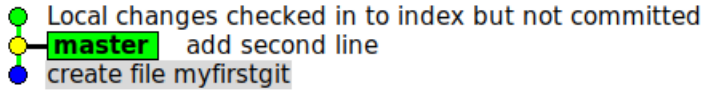
Go back to unstage area

Make some mistake in the txt



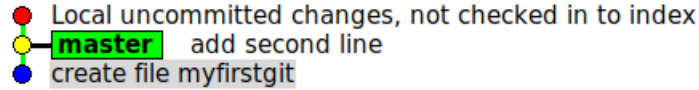

```
first try
Second line
error line
```

git add .



git config --global alias.unstage ”restore --staged”

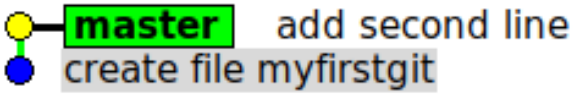
git unstage .




```
first try
Second line
error line
```

Restore

git restore .



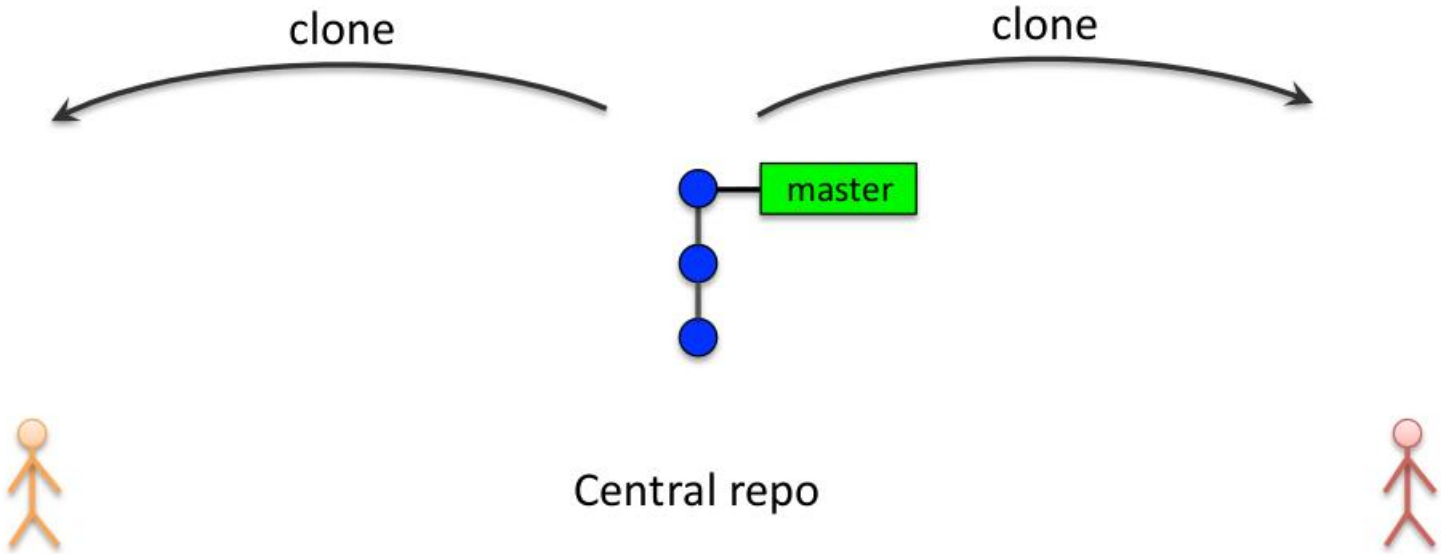

```
first try
Second line
~
```

Change is lost

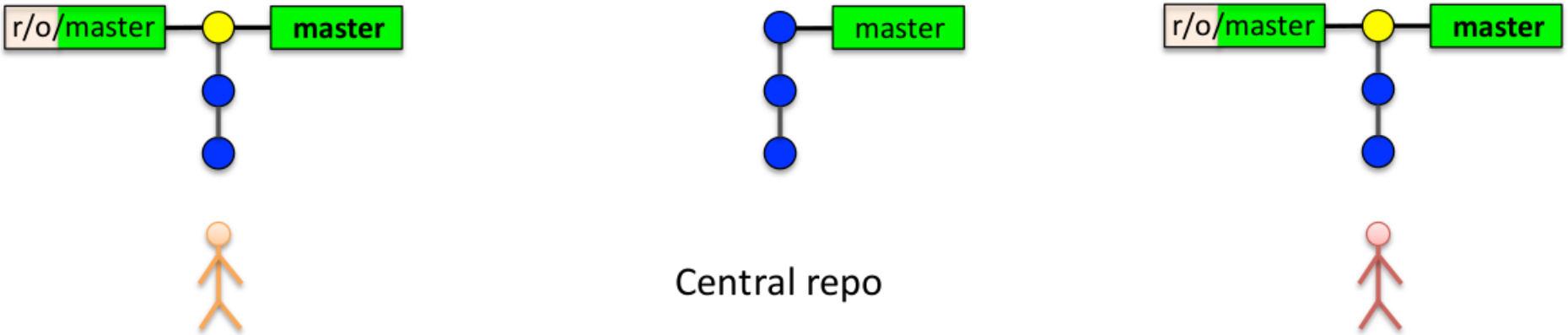
4

Remote

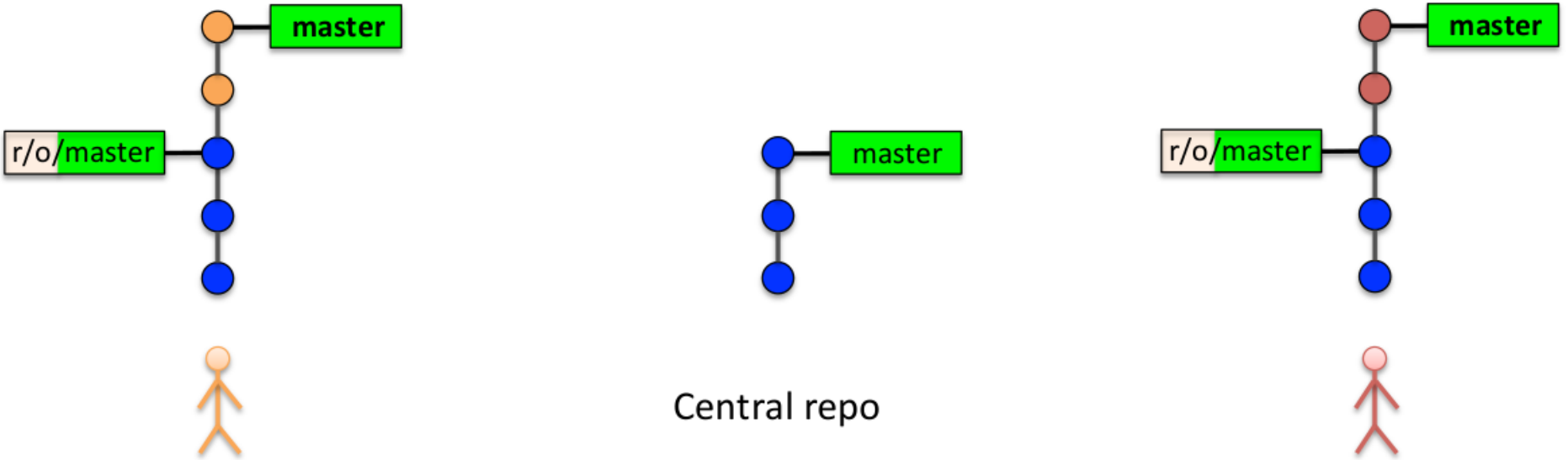
Work with remote



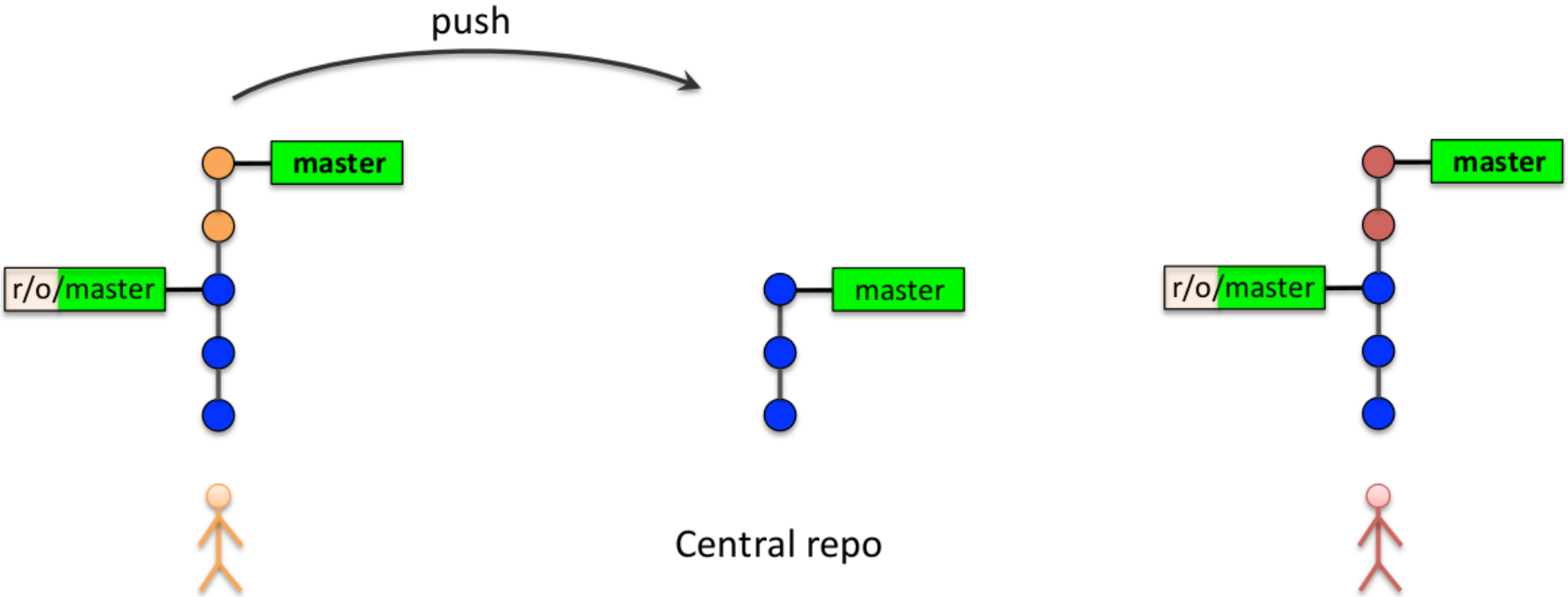
Work with remote



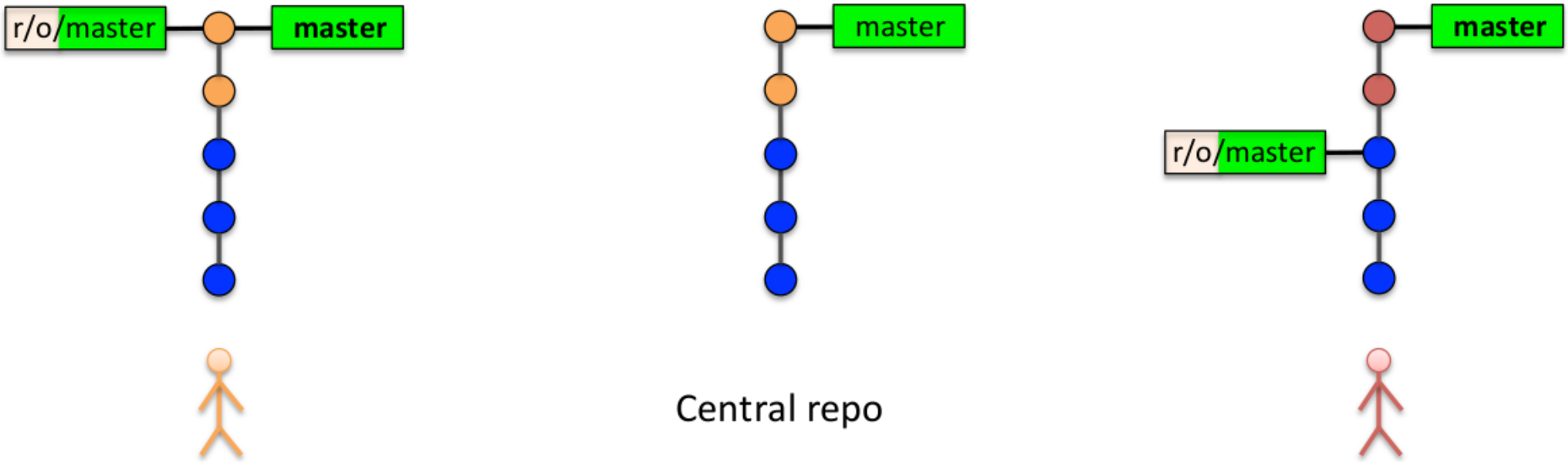
Work with remote



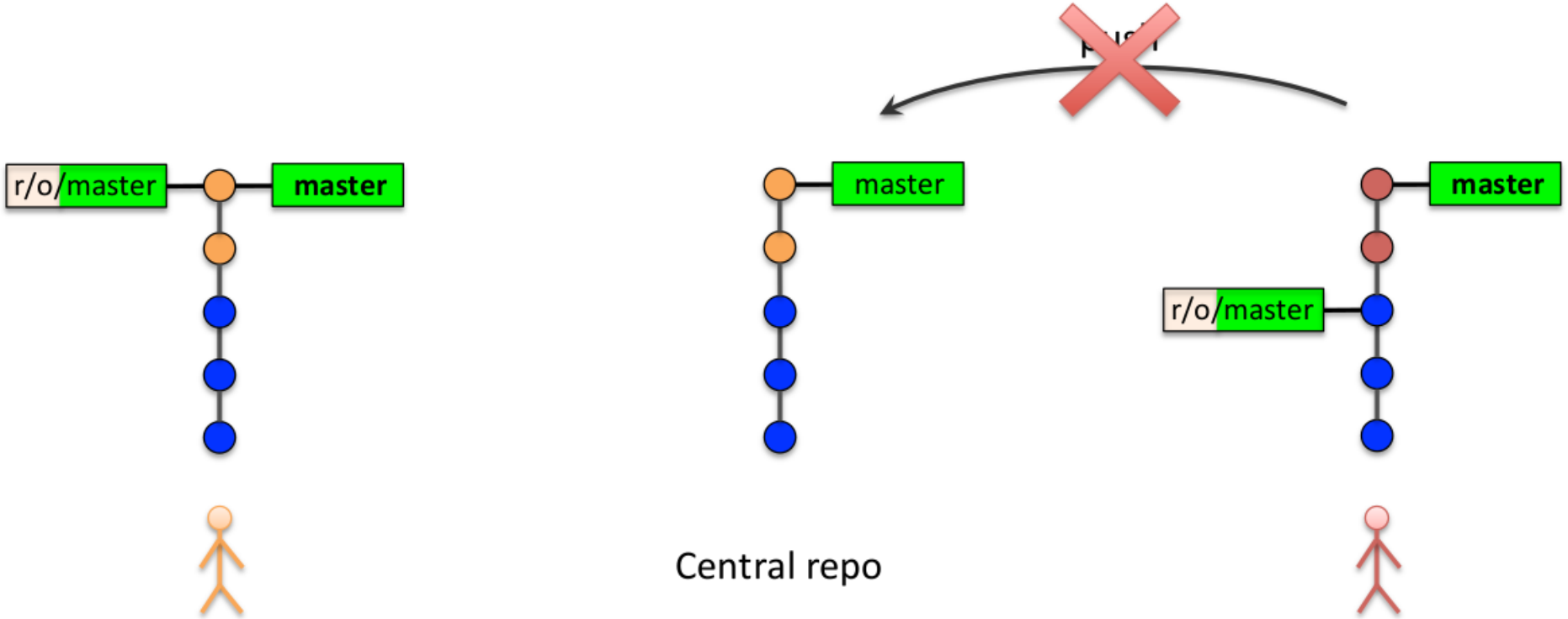
Work with remote



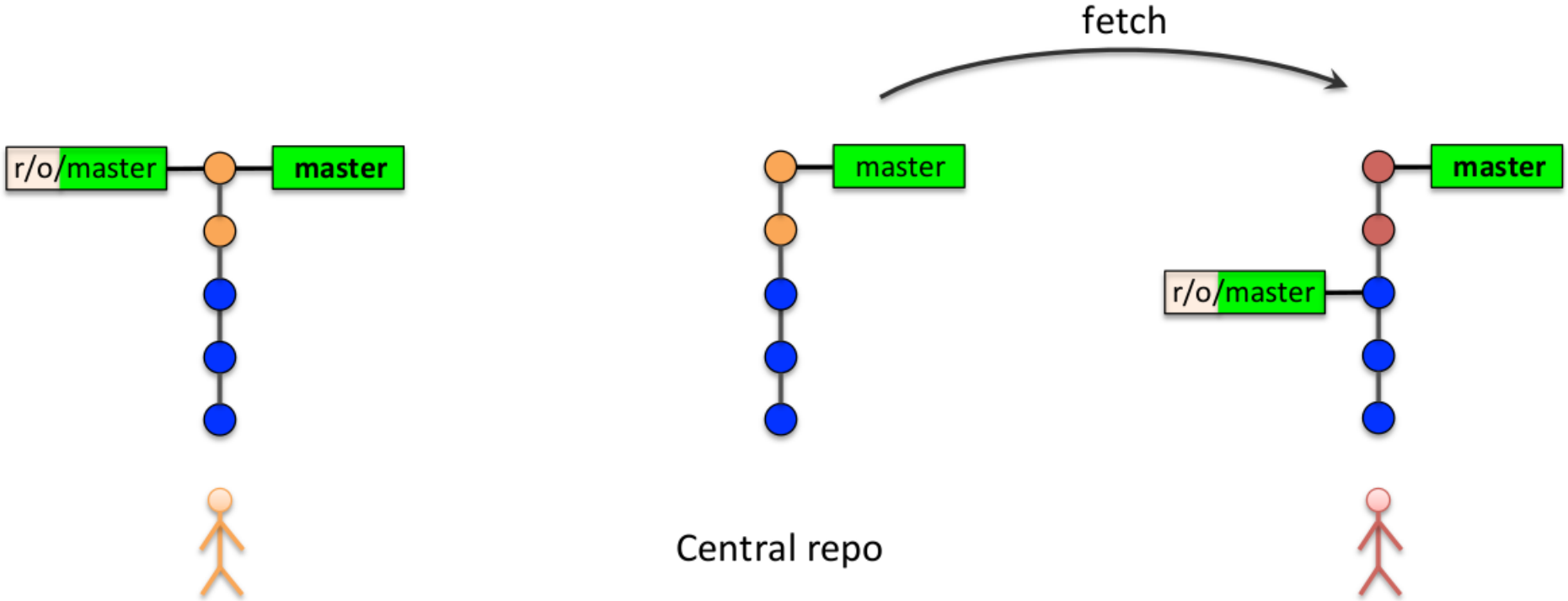
Work with remote



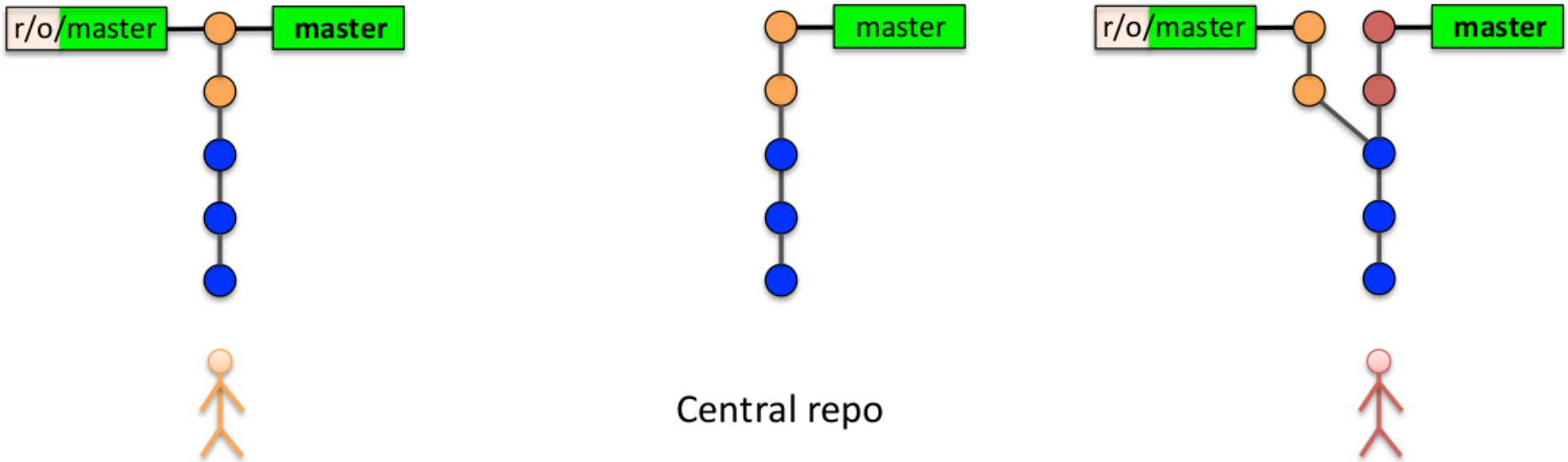
Work with remote



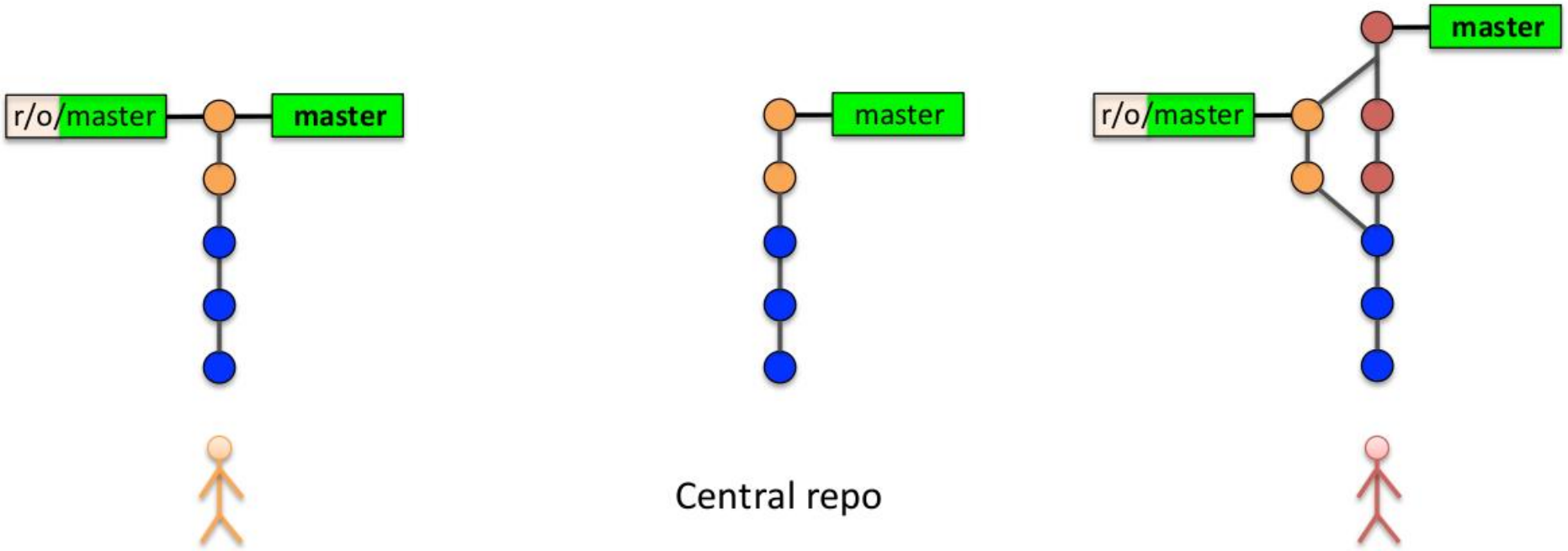
Work with remote



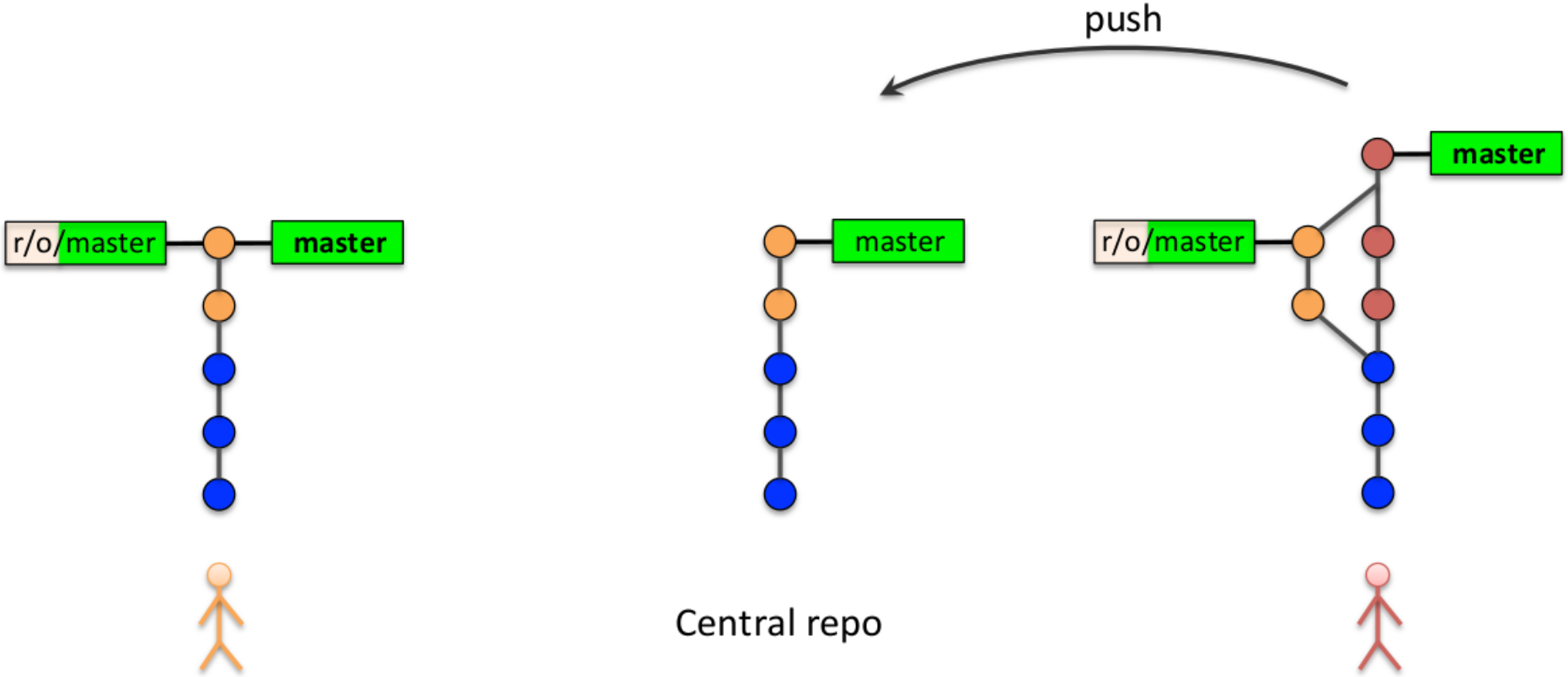
Work with remote



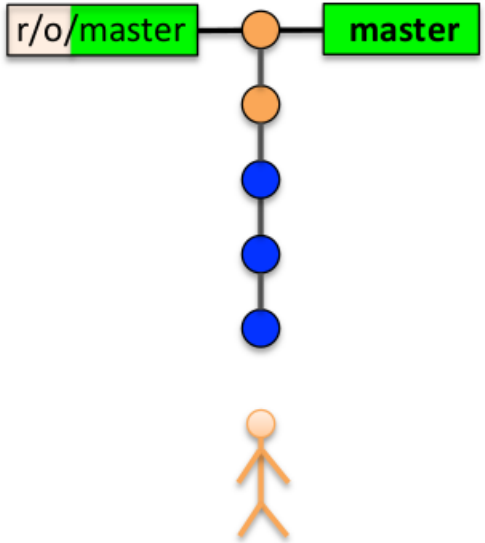
Work with remote

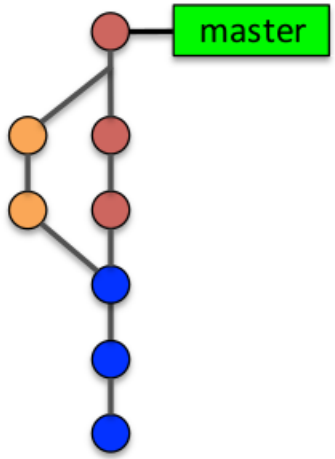


Work with remote

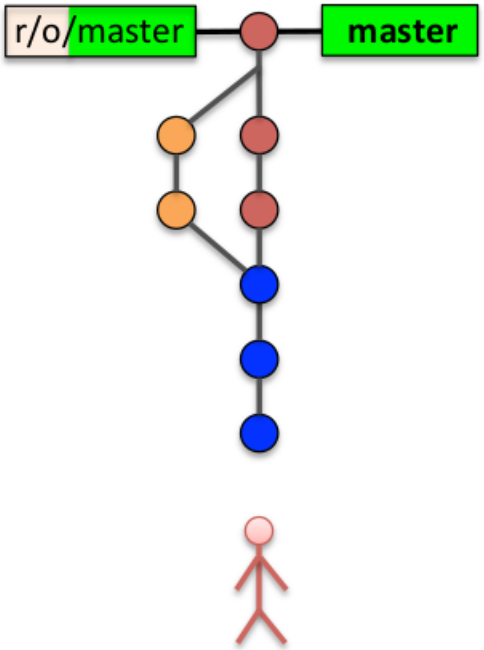


Work with remote

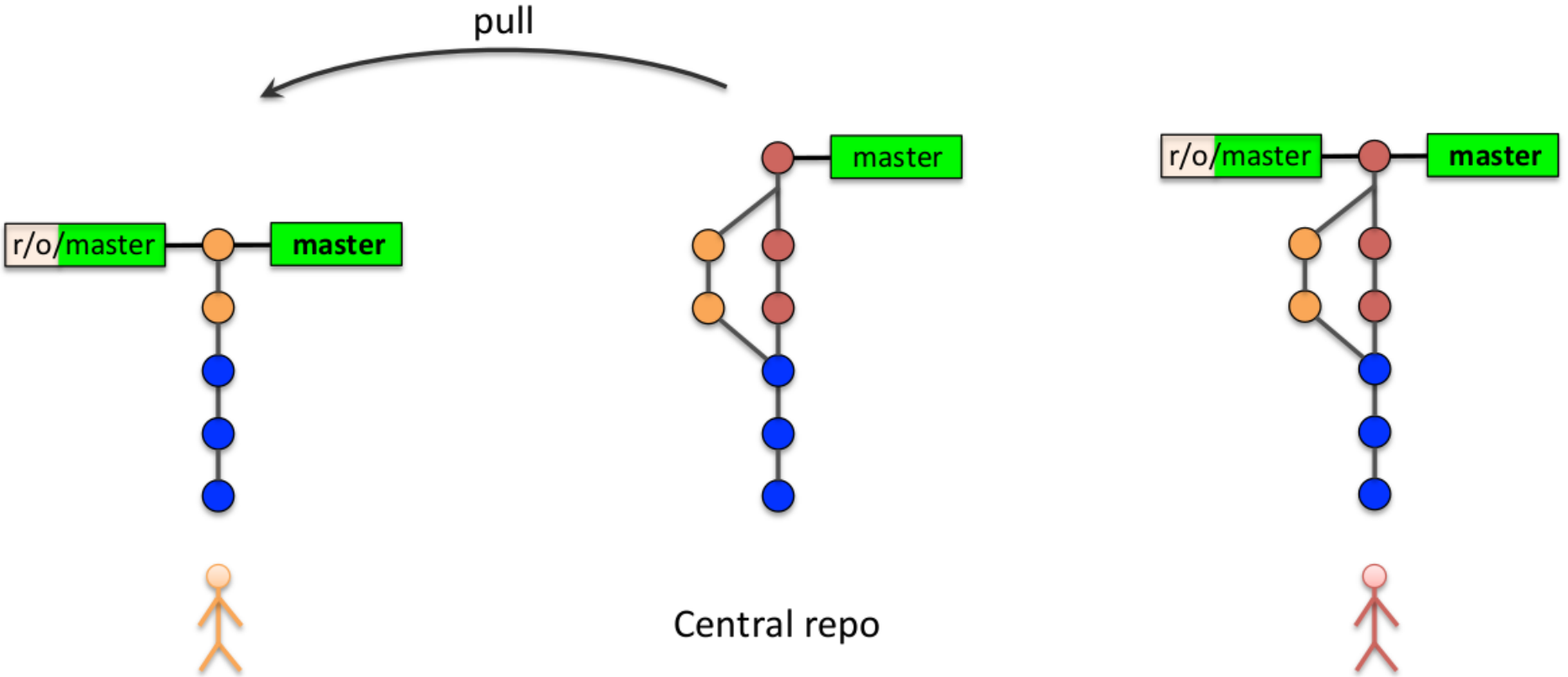




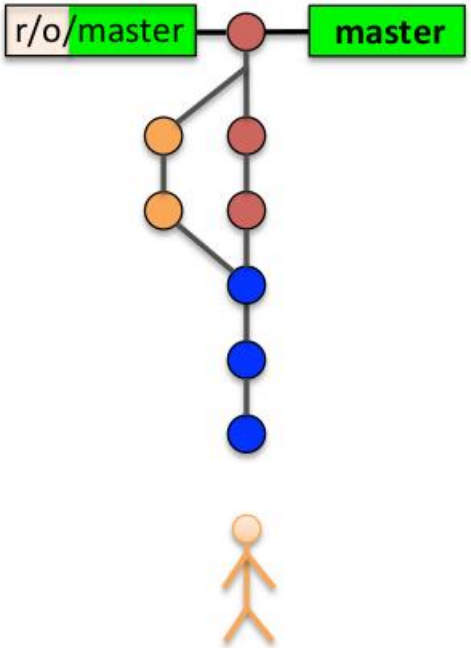
Central repo

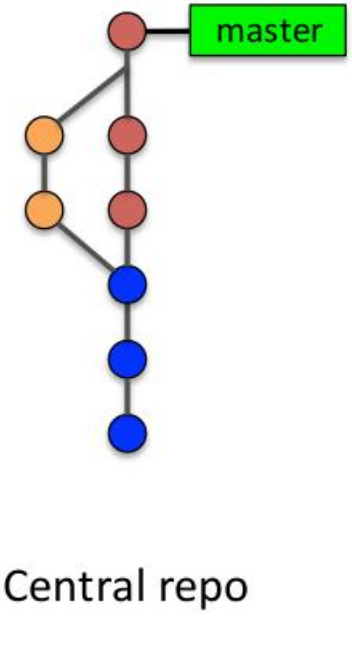


Work with remote

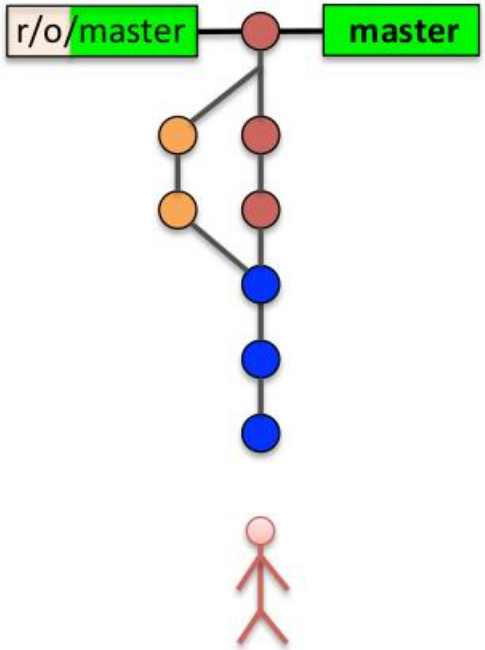


Work with remote





Central repo



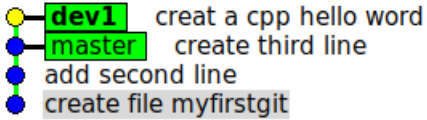
5

Branch

Branch

git br dev1

git checkout dev1

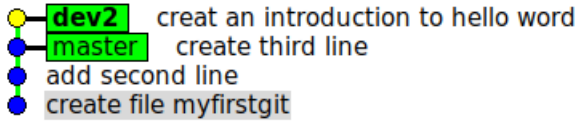



```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ ls
main.cpp  myfirstgit.txt
```

git checkout master

git br dev2

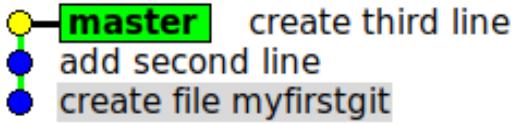
git checkout dev1




```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ dir
introduction_hello_word.txt  myfirstgit.txt
```

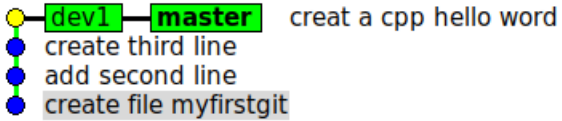
Branch

git co master



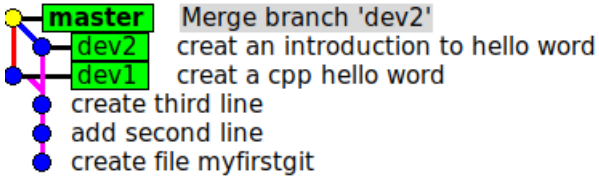

```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ dir
myfirstgit.txt
```

git merge dev1




```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ ls
main.cpp  myfirstgit.txt
```

git merge dev2




```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ dir
introduction_hello_word.txt  main.cpp  myfirstgit.txt
```

6

Specifique Scenarios

Checkout without commit

What is it?

Stash the changes in a dirty working directory away

Why?

A new idea?

git stash

```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ vim myfirstgit.txt
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git stash
Saved working directory and index state WIP on main: a77b59c Merge remote-track
ing branch 'remotes/origin/main' into main
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git br
  dev1
  dev2
* main
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git co dev1
Switched to branch 'dev1'
```

```
first try
Second line
third line : 15151ds
git stash
```

Checkout without commit

Hashes

```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git st
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git loggraph
*   8c27f8d (refs/stash) WIP on main: a77b59c Merge remote-tracking branch 'remotes/origin/main' into main
|\
| * beea817 index on main: a77b59c Merge remote-tracking branch 'remotes/origin/main' into main
|/
*   a77b59c (HEAD -> main, origin/main) Merge remote-tracking branch 'remotes/origin/main' into main
|\
| * 74bfdb1 Initial commit
*   84784e7 Merge branch 'dev2'
|\
| * 0c36323 (dev2) creat an introduction to hello word
* | 208b39f (dev1) creat a cpp hello word
|/
* 1e9cc0c create third line
* bb32248 add second line
* ff3ad41 create file myfirstgit
```

```
first try
Second line
third line : 15151ds
~
```

git stash applay

```
first try
Second line
third line : 15151ds
git stash
```

git stash pop

Change the last Commit

Why?

You made a mistake in the log?

git commit --amend

```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git ci -m "add one blank line"
[main 7178584] add one blank line
 1 file changed, 1 insertion(+)
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git log1
7178584 (HEAD -> main) add one blank line
b2a0307 new ideas
5caeb59 fixed all bugs
618908f add gitignore
fad9e36 git stash test
cabf76e creat a cpp hello word
0c36323 (dev2) creat an introduction to hello word
1e9cc0c create third line
bb32248 add second line
ff3ad41 create file myfirstgit
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git ci --amend  -m "add one blank line in README.md
[main f2db5f0] add one blank line in README.md
 Date: Thu Oct 5 10:35:44 2023 +0200
 1 file changed, 1 insertion(+)
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git log1
f2db5f0 (HEAD -> main) add one blank line in README.md
b2a0307 new ideas
5caeb59 fixed all bugs
618908f add gitignore
fad9e36 git stash test
cabf76e creat a cpp hello word
0c36323 (dev2) creat an introduction to hello word
1e9cc0c create third line
bb32248 add second line
ff3ad41 create file myfirstgit
```

Useless commits

What is it?

Reapply commits on top of another base tip

```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git log1
9a44775 (HEAD -> main) git stash test
406c4d6 Initial commit
e7ee3a5 creat a cpp hello word
0c36323 (dev2) creat an introduction to hello word
1e9cc0c create third line
bb32248 add second line
ff3ad41 create file myfirstgit
```

Useless commits

git rebase -i 0c36323

```
pick e7ee3a5 creat a cpp hello word
pick 406c4d6 Initial commit
pick 9a44775 git stash test

# Rebase 0c36323..9a44775 onto 0c36323 (3 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup <commit> = like "squash", but discard this commit's log message
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
# .       create a merge commit using the original merge commit's
# .       message (or the oneline, if no original merge commit was
# .       specified). Use -c <commit> to reword the commit message.
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
```

Useless commits

```
pick e7ee3a5 creat a cpp hello word
f 406c4d6 Initial commit
pick 9a44775 git stash test
```

```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git log1
70ae790 (HEAD -> main) git stash test
b697a90 creat a cpp hello word
0c36323 (dev2) creat an introduction to hello word
1e9cc0c create third line
bb32248 add second line
ff3ad41 create file myfirstgit
```

Debug

A new solver is developed, but needs to be compared with the old one to debug

new solver

old solver

git stash?

checkout
stash applay
complie

git worktree

Debug

git worktree add newfeature

```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git loggraph
* 483fbb6 (HEAD -> main, origin/main, newfeature) add gitignore
* 5f5548d git stash test
*   a77b59c Merge remote-tracking branch 'remotes/origin/main' into main
|\
| * 74bfdb1 Initial commit
*   84784e7 Merge branch 'dev2'
|\
| * 0c36323 (dev2) creat an introduction to hello word
* | 208b39f (dev1) creat a cpp hello word
|/
* 1e9cc0c create third line
* bb32248 add second line
* ff3ad41 create file myfirstgit
xiliu2016@pc-gem120:~/Desktop/gitdemo$
```

```
xiliu2016@pc-gem120:~/Desktop/gitdemo/newfeature$ git loggraph
* 483fbb6 (HEAD -> newfeature, origin/main, main) add gitignore
* 5f5548d git stash test
*   a77b59c Merge remote-tracking branch 'remotes/origin/main' into main
|\
| * 74bfdb1 Initial commit
*   84784e7 Merge branch 'dev2'
|\
| * 0c36323 (dev2) creat an introduction to hello word
* | 208b39f (dev1) creat a cpp hello word
|/
* 1e9cc0c create third line
* bb32248 add second line
* ff3ad41 create file myfirstgit
xiliu2016@pc-gem120:~/Desktop/gitdemo/newfeature$
```

.gitignore

```
newfeature
.
..
```

Debug

```
iliu2016@pc-gem120:~/Desktop/gitdemo$ git loggraph
 cd5818d (newfeature) new ideas
 * 58ba9aa (HEAD -> main) fixed all bugs
/
 483fbb6 (origin/main) add gitignore
 5f5548d git stash test
   a77b59c Merge remote-tracking branch 'remotes/origin/main' into main
\
 * 74bfdb1 Initial commit
   84784e7 Merge branch 'dev2'
\
 * 0c36323 (dev2) creat an introduction to hello word
 | 208b39f (dev1) creat a cpp hello word
/
 1e9cc0c create third line
 bb32248 add second line
 ff3ad41 create file myfirstgit
iliu2016@pc-gem120:~/Desktop/gitdemo$ dir
ixbugs.txt                main.cpp        newfeature
ntroduction_hello_word.txt  myfirstgit.txt  README.md
iliu2016@pc-gem120:~/Desktop/gitdemo$
```

Fix bugs

```
 create mode 100644 newideas.txt
xiliu2016@pc-gem120:~/Desktop/gitdemo/newfeature$ git loggraph
* cd5818d (HEAD -> newfeature) new ideas
| * 58ba9aa (main) fixed all bugs
|/
* 483fbb6 (origin/main) add gitignore
* 5f5548d git stash test
*   a77b59c Merge remote-tracking branch 'remotes/origin/main' into main
|\
| * 74bfdb1 Initial commit
*   84784e7 Merge branch 'dev2'
|\
| * 0c36323 (dev2) creat an introduction to hello word
* | 208b39f (dev1) creat a cpp hello word
|/
* 1e9cc0c create third line
* bb32248 add second line
* ff3ad41 create file myfirstgit
xiliu2016@pc-gem120:~/Desktop/gitdemo/newfeature$ dir
introduction_hello_word.txt  main.cpp  myfirstgit.txt  newideas.txt  README.md
xiliu2016@pc-gem120:~/Desktop/gitdemo/newfeature$
```

add new ideas

Debug

```
introduction_hello_word.txt  myfirstgit.txt  README.md
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git merge newfeature
Merge made by the 'recursive' strategy.
 newideas.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 newideas.txt
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git loggraph
*   e2c5e5e (HEAD -> main, newfeature) Merge branch 'newfeature' into main
|\
| * cd5818d new ideas
* | 58ba9aa fixed all bugs
|/
* 483fbb6 (origin/main) add gitignore
* 5f5548d git stash test
*   a77b59c Merge remote-tracking branch 'remotes/origin/main' into main
|\
| * 74bfdb1 Initial commit
*   84784e7 Merge branch 'dev2'
|\
| * 0c36323 (dev2) creat an introduction to hello word
* | 208b39f (dev1) creat a cpp hello word
|/
* 1e9cc0c create third line
* bb32248 add second line
* ff3ad41 create file myfirstgit
xiliu2016@pc-gem120:~/Desktop/gitdemo$
```

```
xiliu2016@pc-gem120:~/Desktop/gitdemo/newfeature$ git merge main
Updating cd5818d..e2c5e5e
Fast-forward
 fixbugs.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 fixbugs.txt
xiliu2016@pc-gem120:~/Desktop/gitdemo/newfeature$ git loggraph
*   e2c5e5e (HEAD -> newfeature, main) Merge branch 'newfeature' into main
|\
| * cd5818d new ideas
* | 58ba9aa fixed all bugs
|/
* 483fbb6 (origin/main) add gitignore
* 5f5548d git stash test
*   a77b59c Merge remote-tracking branch 'remotes/origin/main' into main
|\
| * 74bfdb1 Initial commit
*   84784e7 Merge branch 'dev2'
|\
| * 0c36323 (dev2) creat an introduction to hello word
* | 208b39f (dev1) creat a cpp hello word
|/
* 1e9cc0c create third line
* bb32248 add second line
* ff3ad41 create file myfirstgit
xiliu2016@pc-gem120:~/Desktop/gitdemo/newfeature$
```

Debug

```
xiliu2016@pc-gem120:~/Desktop/gitdemo$ git loggraph
*   e2c5e5e (HEAD -> main, origin/main, newfeature) Merge branch 'newfeature' i
nto main
|\
| * cd5818d new ideas
* | 58ba9aa fixed all bugs
|/
* 483fbb6 add gitignore
* 5f5548d git stash test
*   a77b59c Merge remote-tracking branch 'remotes/origin/main' into main
|\
| * 74bfdb1 Initial commit
*   84784e7 Merge branch 'dev2'
|\
| * 0c36323 (dev2) creat an introduction to hello word
* | 208b39f (dev1) creat a cpp hello word
|/
* 1e9cc0c create third line
* bb32248 add second line
* ff3ad41 create file myfirstgit
```

Go back

git br newbranch

git reset ‘Hashes’

git checkout -b newbranch ‘Hashes’

Never do it in main

You are thirsty?

Documentation : https://git-scm.com/book/en/v2

Exercises : https://sed.inrialpes.fr/advancedgit-tuto/

7

TP

Work with remote -- Github

https://github.com/Xiao-dong-LIU/GitTP.git

git clone

git add

git commit

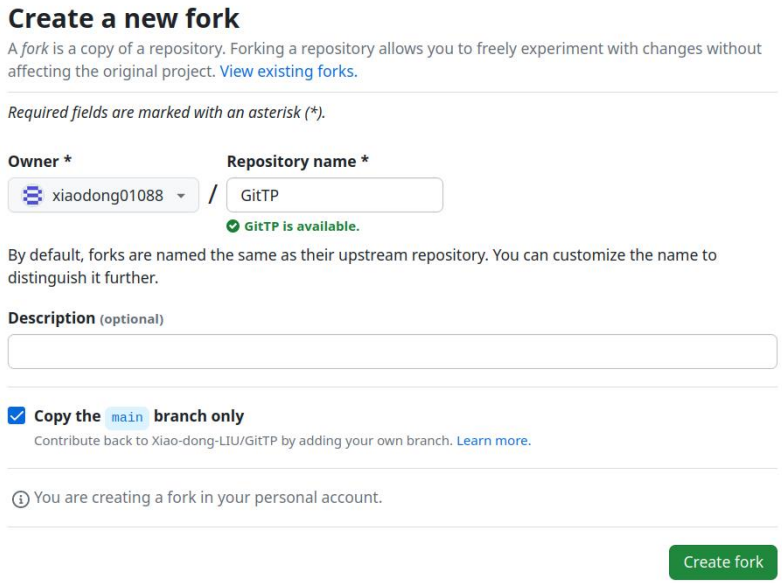
git push

git fetch

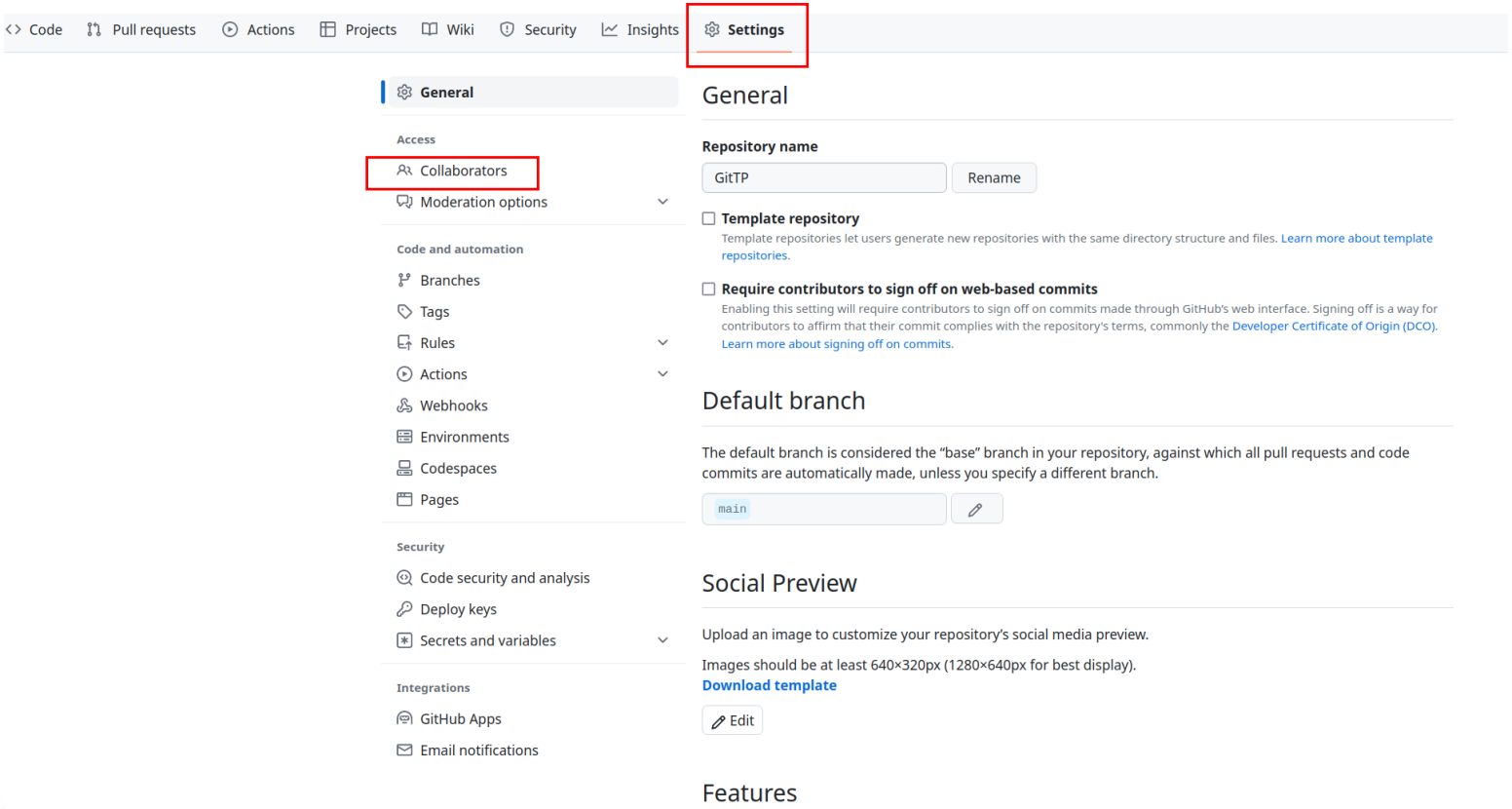
git merge

git pull

git log



Work with remote -- Github



Work with remote -- Github

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

Manage

DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet

Add people

Merci !