



FORMA GIT

Un outil nécessaire pour tous
vos projets



“Mais pourquoi j’aurais
besoin de tout ça moi ?”



Loko a définitivement besoin de mieux s'organiser...

Moi c'est Loko



FlappyOzgur



fichier.py



fichier_Final.py



fichier_v2.py



fichier_final_final.py



fichier_final.py



fichier_vrai_final.py

Git est avant tout un logiciel de gestion de versions

1 version = 1 commit



fichier.py

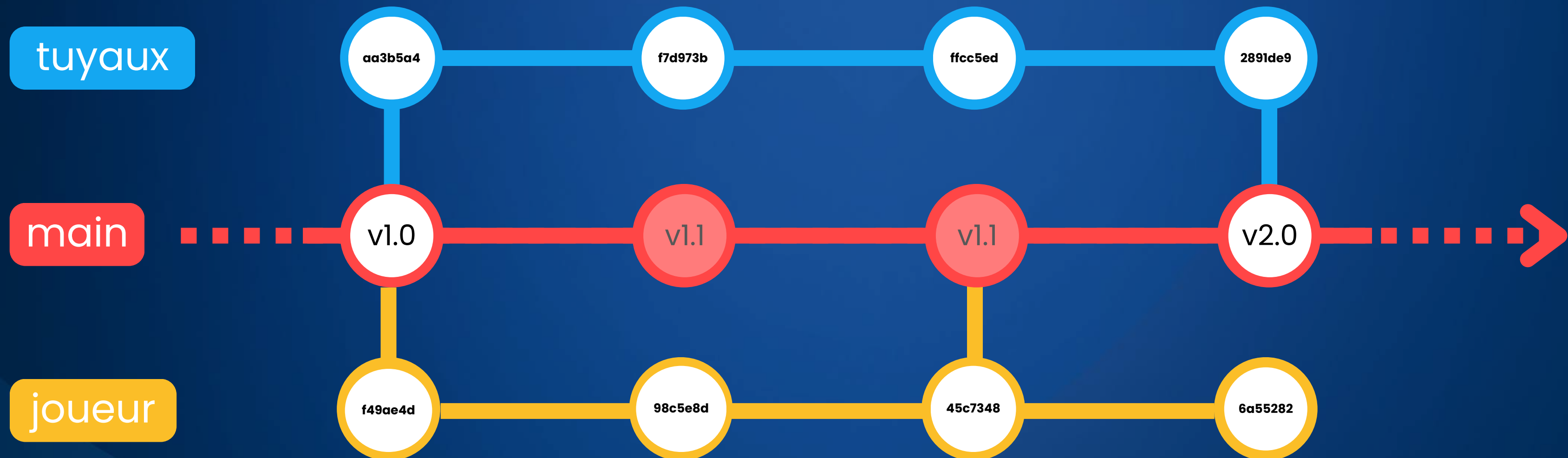
Ouah c'est trop bien !

- ...
- Ajout d'une fonction
- Correction de la fonction
- En fait elle sert à rien je l'enlève
- P*n y a plus rien qui marche
- fjegfpqjduéfôzqdpf

Exemple type d'un projet git

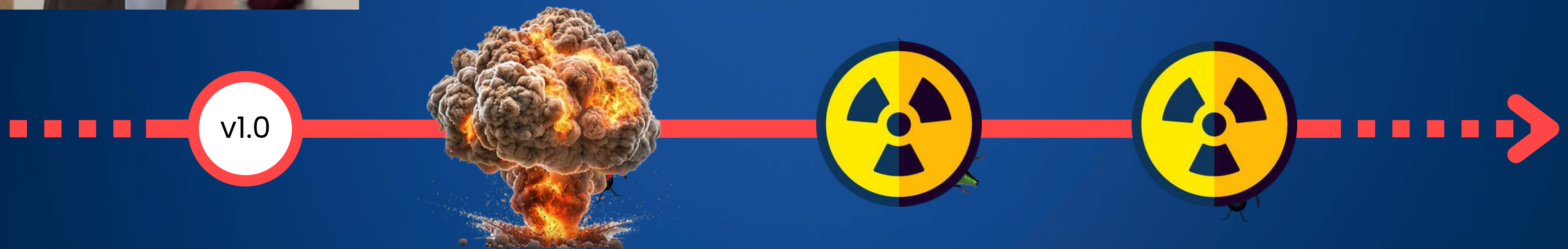


En vrai il y a plusieurs branches



En cas de problème

On peut revenir aux versions précédentes !



Git pour collaborer

Moi c'est Elysée

Et elle casse tout
le projet



Loko  ClubCode 18:48

Voilà le début de backend pour notre projet PacDev !



pacdev.zip

2.67 MB



Elysée  MRTC 19:13

Super merci, j'ai codé les fruits mais ça marche pas (modifié)



pacdev.zip

2.76 MB



Loko  ClubCode 23:12

J'ai tout recodé en C



pacdev.zip

3.14 TB



Elysée  MRTC 23:16

Je venais de corriger les fruits #@^% 🤯



pacdev.zip

2.77 MB

```
fan2git@fedora:~$ sudo apt install git
Updating and loading repositories:
Repositories loaded.
Package "git-2.55.0-1.fc44.x86_64" is already installed.

Nothing to do.
fan2git@fedora:~$ git init vive_git
Initialized empty Git repository in /home/fan2git/vive_git/.git/
fan2git@fedora:~$ cd vive_git/
fan2git@fedora:~/vive_git$ ls -la
.  ..  .git
fan2git@fedora:~/vive_git$
```

Installer git avec **sudo apt install git**

mkdir crée un dossier

cd pour s'y déplacer

ls -la liste les fichiers et dossiers

Initialiser le répo git avec **git init**

git init permet aussi de créer le dossier directement

cheat sheet à la fin

"Carrément je suis pas le fou du bus je suis le fou de l'avion"



Bon on va utiliser git en local ce sera mieux



```
fan2git@fedora:~/vive_git$ git config user.name "CCRTC"  
fan2git@fedora:~/vive_git$ git config user.email "ccrtc@hackademint.org"  
fan2git@fedora:~/vive_git$
```

On va configurer
git pour être
surveillé identifié

Configuration avec **git config**

user.name change le nom

user.email pour changer d'email

Utilisez **--global** pour appliquer la
config à tous vos projets



Exemple concret



Mais comment on fait ensuite ?



> écraser le contenu d'un fichier

git status pour voir l'état actuel

git add ajoute les changements
préparer le futur commit

Vous maintenant ↓



```
fan2git@fedora:~/vive_git$ echo 'print("Hello World!")' > script1.py
fan2git@fedora:~/vive_git$ python3 script1.py
Hello World!
fan2git@fedora:~/vive_git$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        script1.py

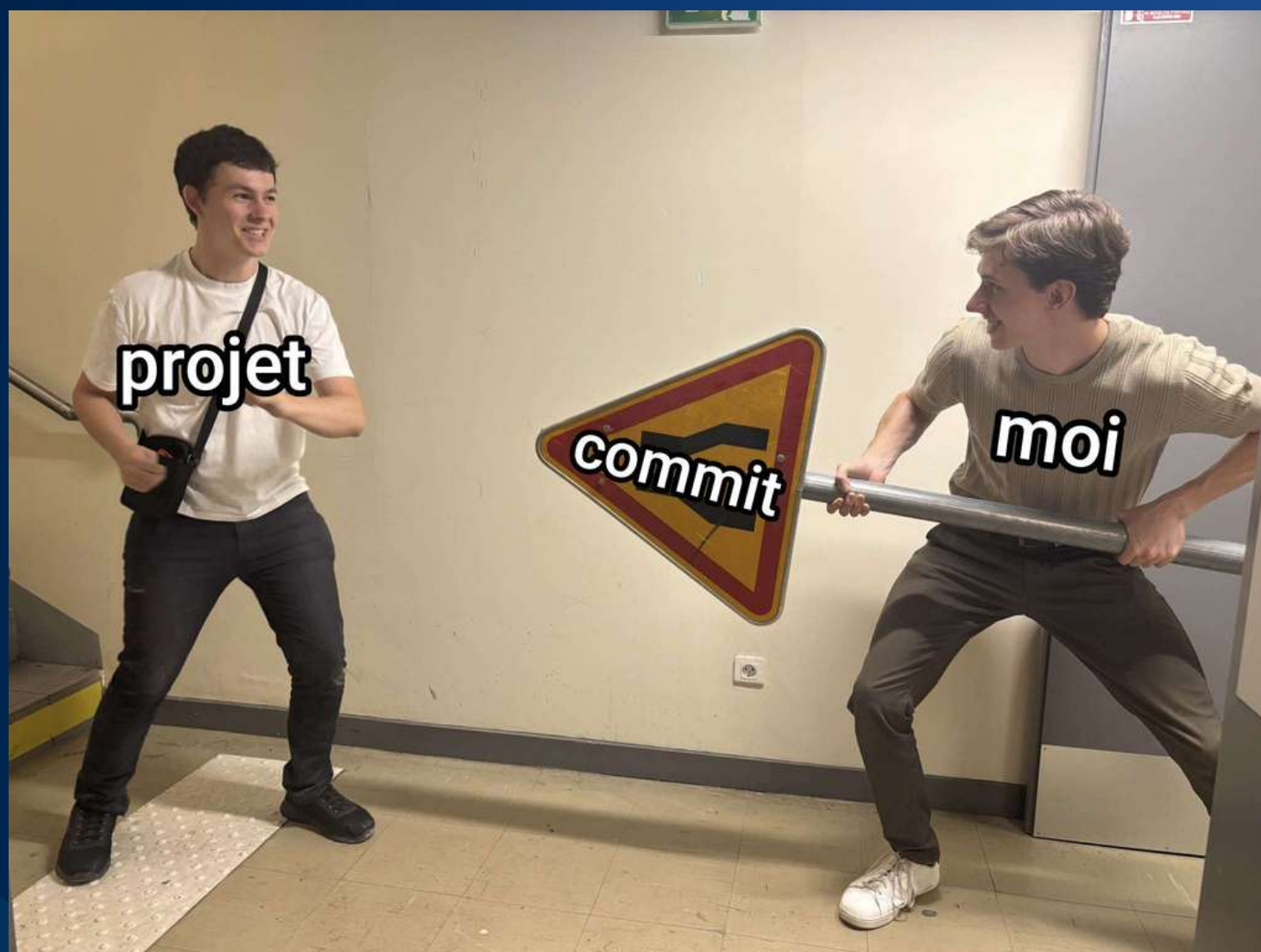
nothing added to commit but untracked files present (use "git add" to track)
fan2git@fedora:~/vive_git$ git add script1.py
fan2git@fedora:~/vive_git$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   script1.py

fan2git@fedora:~/vive_git$
```

```
> ✂ -: bash — Konsole
fan2git@fedora:~/vive_git$ git commit -m "Ajout de mon premier script python"
[master (root-commit) ab7f6f4] Ajout de mon premier script python
1 file changed, 1 insertion(+)
create mode 100644 script1.py
fan2git@fedora:~/vive_git$
```



git commit pour commiter

On peut ensuite commit avec un message



Le message du commit est **obligatoire** et il est conseillé de le faire commencer par **fix, chore, feat ou docs** pour la clarté



Fichier_1.py



script1.py



fichier_temporaire



README.md

```
~: bash — Konsole
fan2git@fedora:~/vive_git$ echo HRTC > fichier_temporaire
fan2git@fedora:~/vive_git$ git add fichier_temporaire
fan2git@fedora:~/vive_git$ git commit -m "Ajout d'un message important
> "
[master ad69045] Ajout d'un message important
1 file changed, 1 insertion(+)
create mode 100644 fichier_temporaire
fan2git@fedora:~/vive_git$ echo 'print("HELL000 WORLD")' > script1.py
fan2git@fedora:~/vive_git$ git rm fichier_temporaire
rm 'fichier_temporaire'
fan2git@fedora:~/vive_git$ echo 'Mon magnifique projet' > README.md
fan2git@fedora:~/vive_git$ git add script1.py README.md
fan2git@fedora:~/vive_git$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   README.md
    deleted:    fichier_temporaire
    modified:   script1.py
fan2git@fedora:~/vive_git$
```

Je peux
consulter
mes modifs





Ou bien
l'historique

```

~ : bash — Konsole
fan2git@fedora:~/vive_git$ echo HRTC > fichier_temporaire
fan2git@fedora:~/vive_git$ git add fichier_temporaire
fan2git@fedora:~/vive_git$ git commit -m "Ajout d'un message important"
> "
[master ad69045] Ajout d'un message important
1 file changed, 1 insertion(+)
create mode 100644 fichier_temporaire
fan2git@fedora:~/vive_git$ echo 'print("HELLO00 WORLD")' > script1.py
fan2git@fedora:~/vive_git$ git rm fichier_temporaire
rm 'fichier_temporaire'
fan2git@fedora:~/vive_git$ echo 'Mon magnifique projet' > README.md
fan2git@fedora:~/vive_git$ git add script1.py README.md
fan2git@fedora:~/vive_git$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   README.md
    deleted:    fichier_temporaire
    modified:   script1.py

fan2git@fedora:~/vive_git$ git commit -m "Plusieurs modifs"
[master 220aa52] Plusieurs modifs
3 files changed, 2 insertions(+), 2 deletions(-)
create mode 100644 README.md
delete mode 100644 fichier_temporaire

fan2git@fedora:~/vive_git$ git log
commit 220aa52806fab46cc57220f61b6b33c03961e503 (HEAD -> master)
Author: CCRTC <ccrtc@hackademint.org>
Date:   Mon Sep 21 00:32:14 2026 +0200

    Plusieurs modifs

commit ad69045498cdebb54598eccf91ba534ee61d9a9c
Author: CCRTC <ccrtc@hackademint.org>
Date:   Mon Sep 21 00:28:10 2026 +0200

    Ajout d'un message important

commit d2dbc2ca559ca6b9b44e2c13f5f63886fb0abb9d
Author: CCRTC <ccrtc@hackademint.org>
Date:   Sun Sep 20 23:25:02 2026 +0200

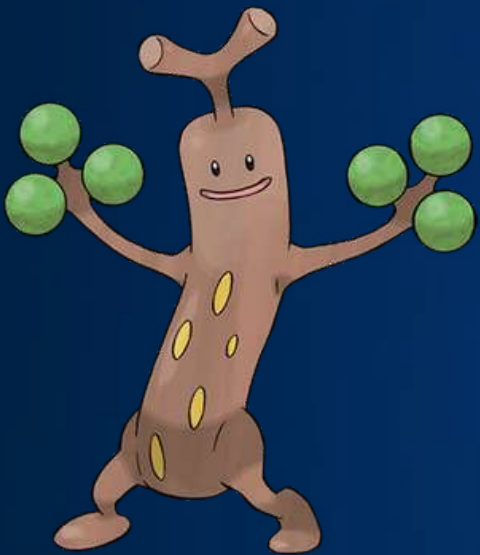
    Ajout de mon premier script python

```

Créer une nouvelle branche

`git branch` crée une branche

`git checkout` pour changer



`git branch`

main

aa3b5a4

f7d973b

ffcc5ed

2891de9

dev

f49ae4d

98c5e8d

45c7348

```
fan2git@fedora:~/vive_git$ git branch
* master
fan2git@fedora:~/vive_git$ git branch reponse-gen-z
fan2git@fedora:~/vive_git$ git branch
* master
  reponse-gen-z
fan2git@fedora:~/vive_git$ git checkout reponse-gen-z
Switched to branch 'reponse-gen-z'
fan2git@fedora:~/vive_git$ echo 'coubeh' > quoi
fan2git@fedora:~/vive_git$ git add quoi
fan2git@fedora:~/vive_git$ git commit -m "j'ai mis une blague trop drôle"
[reponse-gen-z 8a60989] j'ai mis une blague trop drôle
1 file changed, 1 insertion(+)
create mode 100644 quoi
fan2git@fedora:~/vive_git$ ls
quoi README.md script1.py
fan2git@fedora:~/vive_git$ git checkout master
Switched to branch 'master'
fan2git@fedora:~/vive_git$ ls
README.md script1.py
fan2git@fedora:~/vive_git$
```

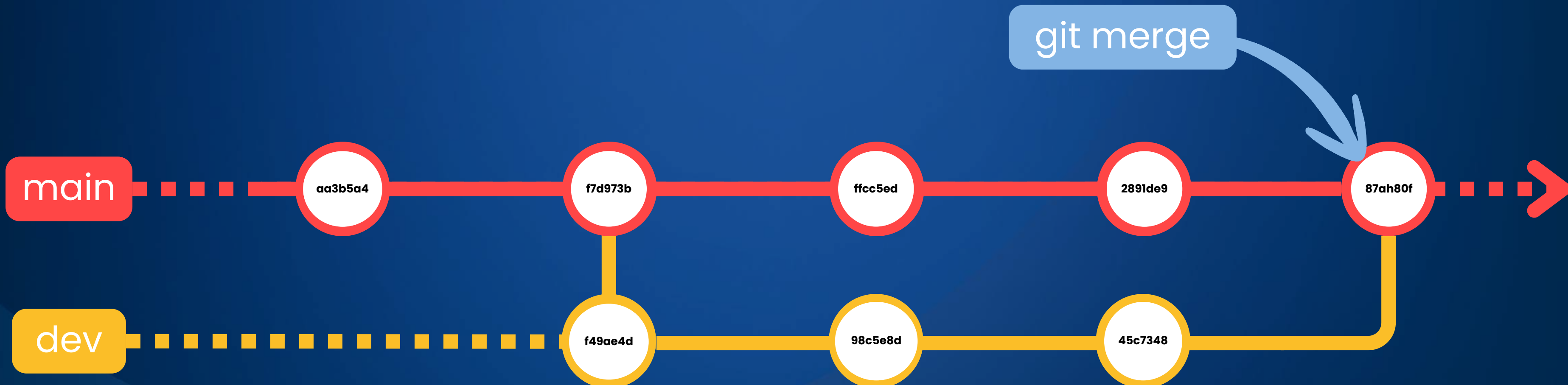
Fusionner les branches

```
fan2git@fedora:~/vive_git$ echo 'feur' > quoi
fan2git@fedora:~/vive_git$ git add quoi
fan2git@fedora:~/vive_git$ git commit -m "J'ai mis un message encore plus marrant"
[master 9a89fc9] J'ai mis un message encore plus marrant
 1 file changed, 1 insertion(+)
 create mode 100644 quoi
fan2git@fedora:~/vive_git$ git merge
HEAD          master          ORIG_HEAD      reponse-gen-z
fan2git@fedora:~/vive_git$ git merge reponse-gen-z
Auto-merging quoi
```

git branch crée une branche

git checkout pour changer

git merge pour fusionner



Gérer les conflits

```
fan2git@fedora:~/vive_git$ echo 'feur' > quoi
fan2git@fedora:~/vive_git$ git add quoi
fan2git@fedora:~/vive_git$ git commit -m "J'ai mis un message encore plus marrant"
[master 9a89fc9] J'ai mis un message encore plus marrant
1 file changed, 1 insertion(+)
create mode 100644 quoi
fan2git@fedora:~/vive_git$ git merge
HEAD          master          ORIG_HEAD     reponse-gen-z
fan2git@fedora:~/vive_git$ git merge reponse-gen-z
Auto-merging quoi
CONFLICT (add/add): Merge conflict in quoi
Automatic merge failed; fix conflicts and then commit the result.
fan2git@fedora:~/vive_git$ cat quoi
<<<<<<< HEAD
feur
=====
coubeh
>>>>>>> reponse-gen-z
fan2git@fedora:~/vive_git$ nano quoi
fan2git@fedora:~/vive_git$ git add quoi
fan2git@fedora:~/vive_git$ git commit -m "J'ai gagné la bagarre (contre moi-même)"
[master 9c7ffd0] J'ai gagné la bagarre (contre moi-même)
fan2git@fedora:~/vive_git$
```



La commande **nano** permet d'ouvrir un éditeur très simple sur le terminal, c'est également le nom d'un super mandat.





C'est bon le projet
va tout bien

Bon par contre
nos fichiers
sont pas
synchro là





Git en local c'est
pas safe



OH NON CHARLES
COMMUNISTE EST
REVENU !!!



On fait tout
exploser

Git en réseau : les forges



Forge Git : plateforme en ligne qui héberge
des dépôts Git + outils de collaboration
(suivi des issues, demandes de fusion, gestion de projets...)



Synchroniser son repo au serveur

```
fan2git@fedora:~/vive_git$ git remote
fan2git@fedora:~/vive_git$ git remote add origin git@gitlab.minet.net:fan2git/vive-git.git
fan2git@fedora:~/vive_git$ git push --set-upstream origin --all
branch 'master' set up to track 'origin/master'.
branch 'reponse-gen-z' set up to track 'origin/reponse-gen-z'.
Everything up-to-date
fan2git@fedora:~/vive_git$
```



Il faut préalablement avoir créé son répertoire sur le serveur !

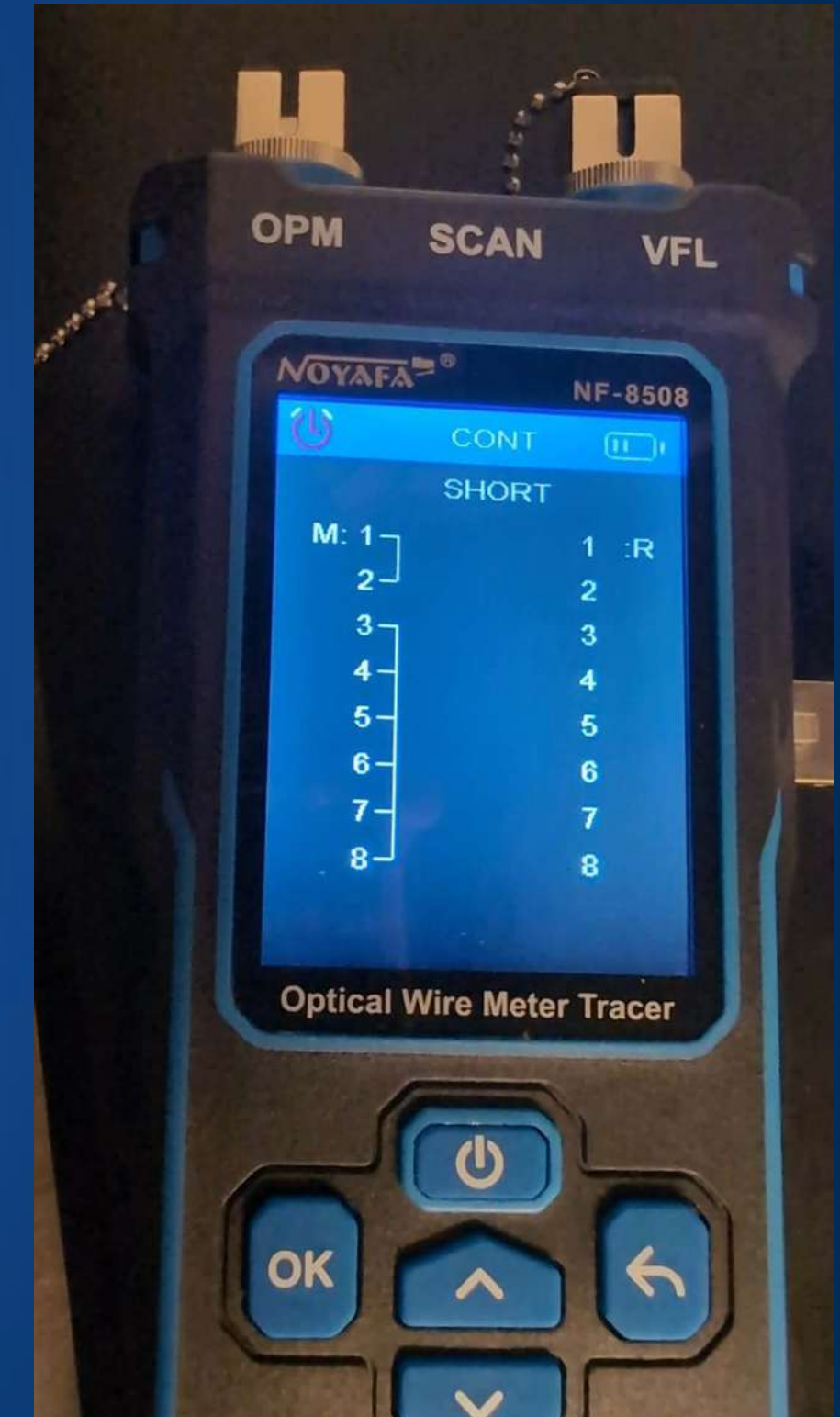


Consulter ses branches

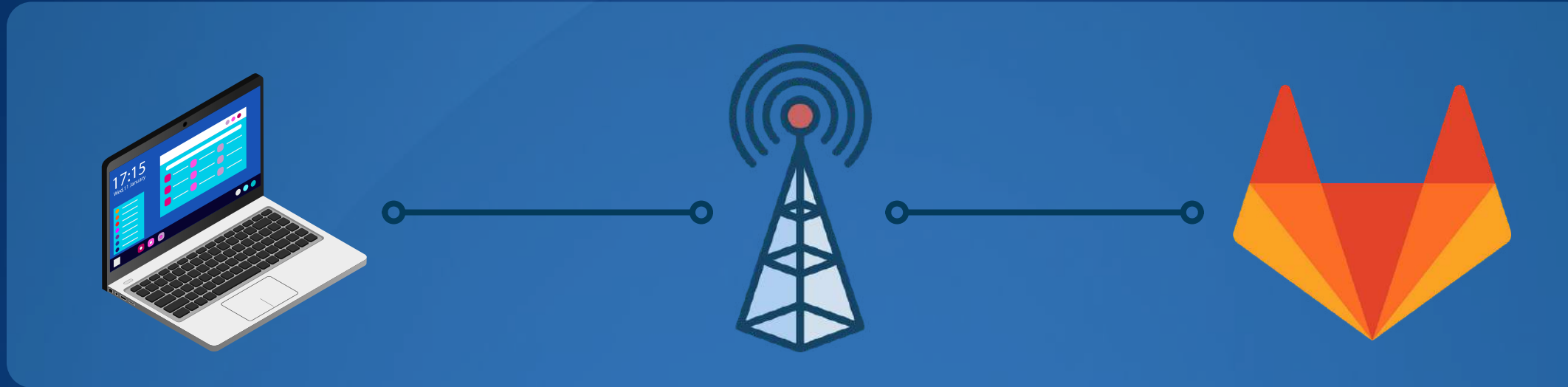
```
fan2git@fedora:~/vive_git$ git branch -a
* master
  reponse-gen-z
  remotes/origin/master
  remotes/origin/reponse-gen-z
fan2git@fedora:~/vive_git$
```



On remarque que les branches distantes sont séparées des branches locales.



Récupérer un repo sur le serveur



Je peux ensuite
récupérer
le repo

```
fan2git@fedora:~/vive_git$ cd
fan2git@fedora:~$ mkdir repo-git && cd repo-git
fan2git@fedora:~/repo-git$ git clone "https://gitlab.minet.net/$name/vive-git.git"
Cloning into 'vive-git'...
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 17 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (17/17), done.
fan2git@fedora:~/repo-git$ cd vive-git/
fan2git@fedora:~/repo-git/vive-git$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/reponse-gen-z
fan2git@fedora:~/repo-git/vive-git$
```

Mettre à jour le serveur

```
fan2git@fedora:~/repo-git/vive-git$ git config user.name HRTC && git config user.email "HRTC@clubcode.fr"
fan2git@fedora:~/repo-git/vive-git$ echo "J'ai réussi à synchroniser mon projet git grâce à GitLab !" >> README.md
fan2git@fedora:~/repo-git/vive-git$ git add README.md
fan2git@fedora:~/repo-git/vive-git$ git commit -m "qsdckflhqdh1"
[master ad4a33c] qsdckflhqdh1
1 file changed, 1 insertion(+)
fan2git@fedora:~/repo-git/vive-git$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
    (use "git push" to publish your local commits)

nothing to commit, working tree clean
fan2git@fedora:~/repo-git/vive-git$ git push
```

git push

pousser vers le serveur

Se mettre à jour

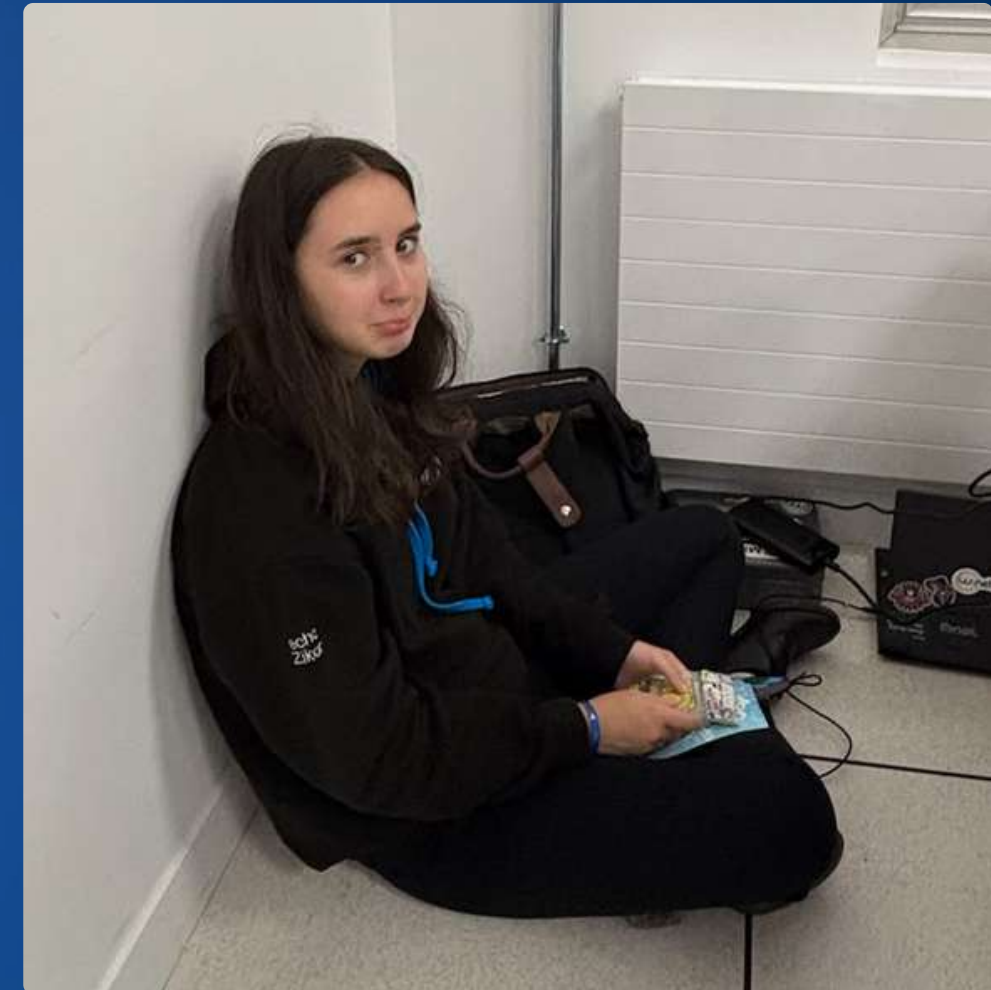
```
fan2git@fedora:~/repo-git/vive-git$ cd ~/vive_git/
fan2git@fedora:~/vive_git$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
fan2git@fedora:~/vive_git$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 373 bytes | 373.00 KiB/s, done.
From gitlab.minet.net: {{user}} /vive-git
   9c7ffd0..24804fe  master    -> origin/master
fan2git@fedora:~/vive_git$ git status
On branch master
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working tree clean
fan2git@fedora:~/vive_git$ git pull
Updating 9c7ffd0..24804fe
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
fan2git@fedora:~/vive_git$ cat README.md
Mon magnifique projet
J'ai réussi à synchroniser mon projet git grâce à GitLab !
fan2git@fedora:~/vive_git$
```

git pull

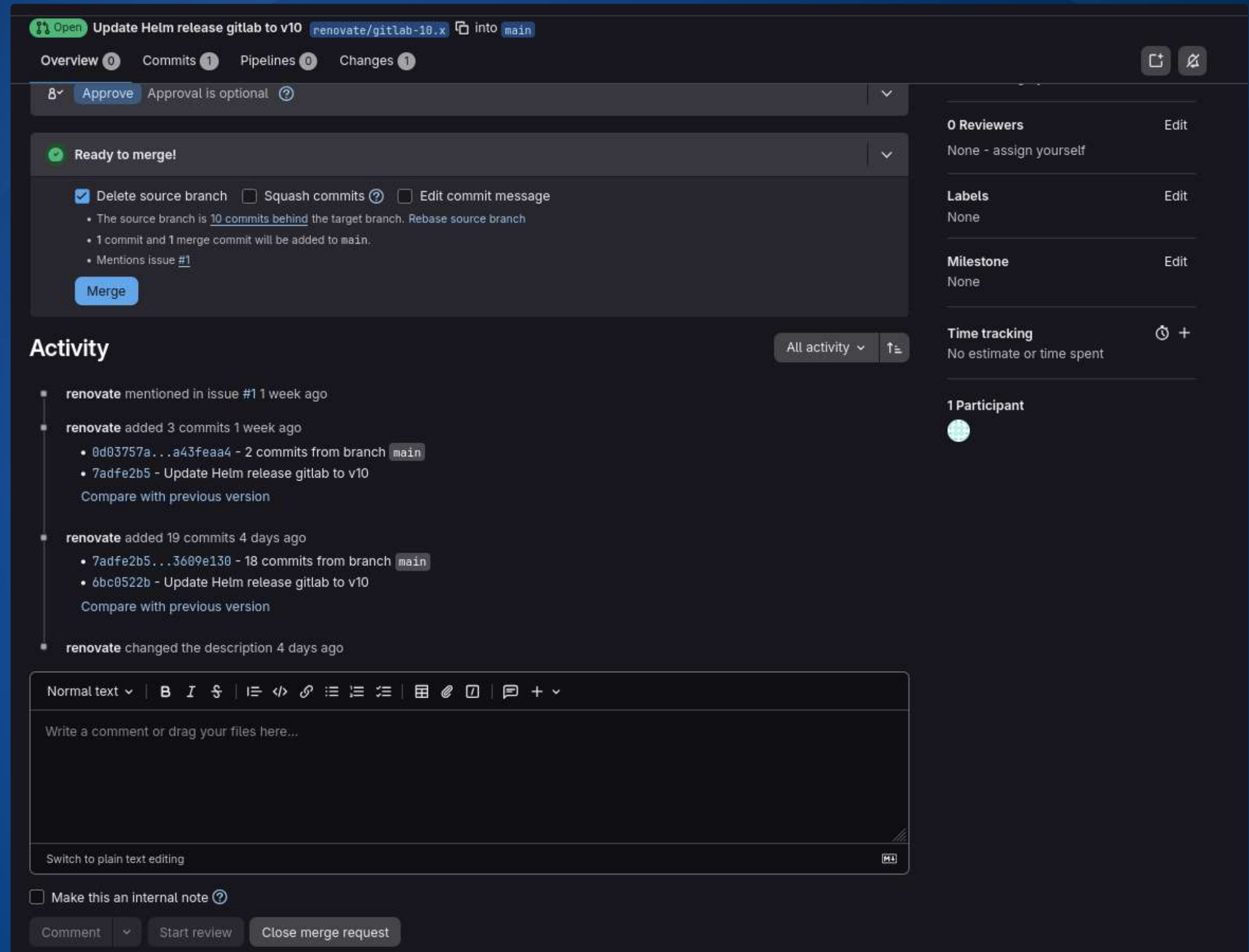
mettre à jour sa branche locale



“J’ai commit avant de pull”

Et pour un projet public ?

J'ai pas envie qu'on sabote mon projet dev !



The screenshot shows a GitLab merge request (MR) titled "Update Helm release gitlab to v10" from the branch "renovate/gitlab-10.x" into "main". The interface includes tabs for Overview, Commits (1), Pipelines (0), and Changes (1). A dropdown menu shows "8" and "Approve" with a note "Approval is optional". A green status bar indicates "Ready to merge!". Below this, there are checkboxes for "Delete source branch" (checked), "Squash commits" (unchecked), and "Edit commit message" (unchecked). A list of details follows: "The source branch is 10 commits behind the target branch. Rebase source branch", "1 commit and 1 merge commit will be added to main.", and "Mentions issue #1". A blue "Merge" button is present. The "Activity" section shows a timeline of actions by "renovate", including mentioning in issue #1, adding commits, and changing the description. A rich text editor at the bottom allows for comments, with a "Switch to plain text editing" option. At the very bottom, there are buttons for "Comment", "Start review", and "Close merge request". On the right sidebar, there are sections for "0 Reviewers", "Labels" (None), "Milestone" (None), "Time tracking" (No estimate or time spent), and "1 Participant".

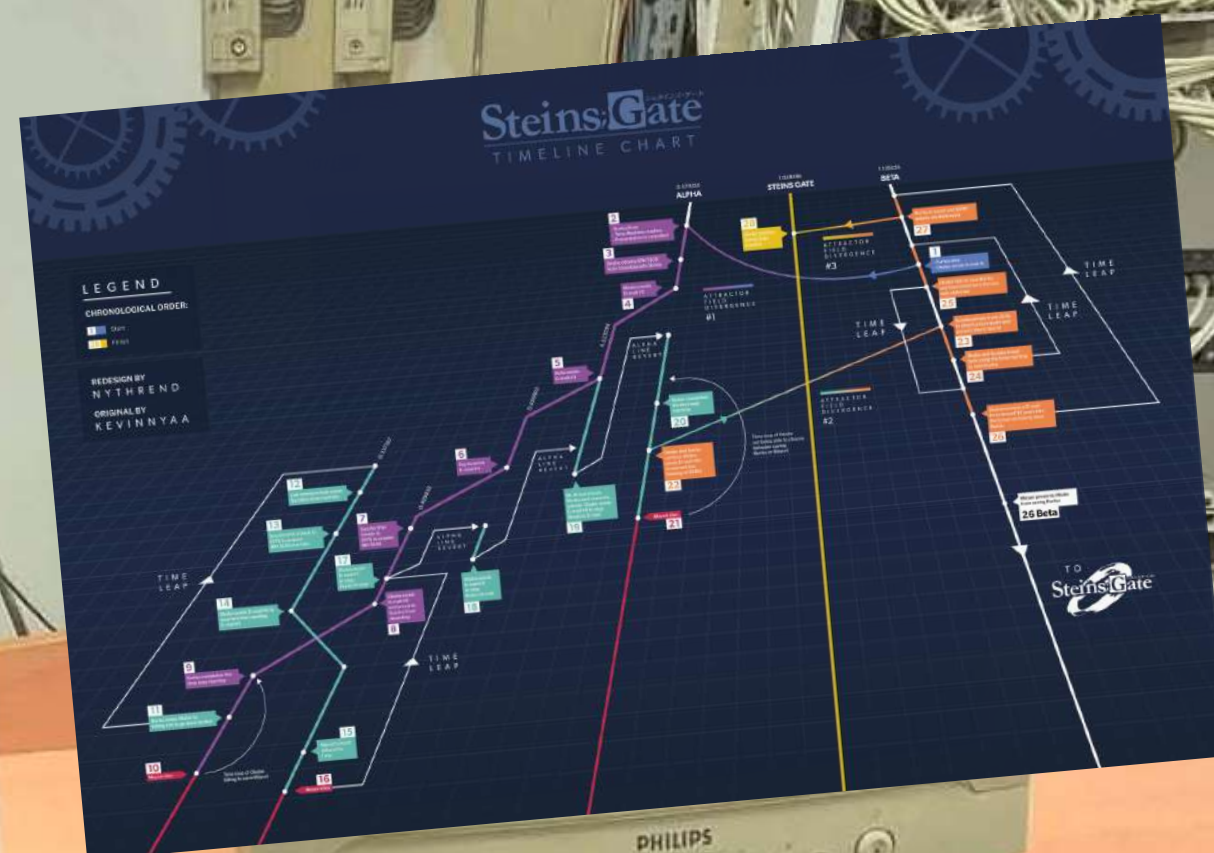
On utilise les PR/MR

C'est l'heure du bloocket !





PLACE AU TP !



Cheat sheet des commandes git

git init Initialiser le répo git

git config Configurer git *user.name*
user.email

git status Afficher l'état

git add <file> Ajouter les changements

git commit -m <message> Commiter

git fetch Récupérer la version distante

git pull Mettre à jour la version locale

git push Envoyer la version locale sur le serveur

sudo apt install git Installer git avec apt

mkdir <dir> créer un dossier

cd <dir> se déplacer dans un dossier

ls -a <dir> liste les fichiers et dossiers cachés

git branch <name> Créer une branche

git checkout <name> Pour changer de branche

git merge <name> Fusionner avec une branche

git remote add origin git@github.com:User/UserRepo.git Ajouter un repo distant