FAIR Bioinfo 2022

Introduction to Git, GitHub/Gitlab

P. Marin, Mateo Hiriart, P. Ruiz, N. Goué

aubi@uca.fr

Université Clermont Auvergne, AuBi, Mésocentre

du 28 novembre au 02 décembre 2022

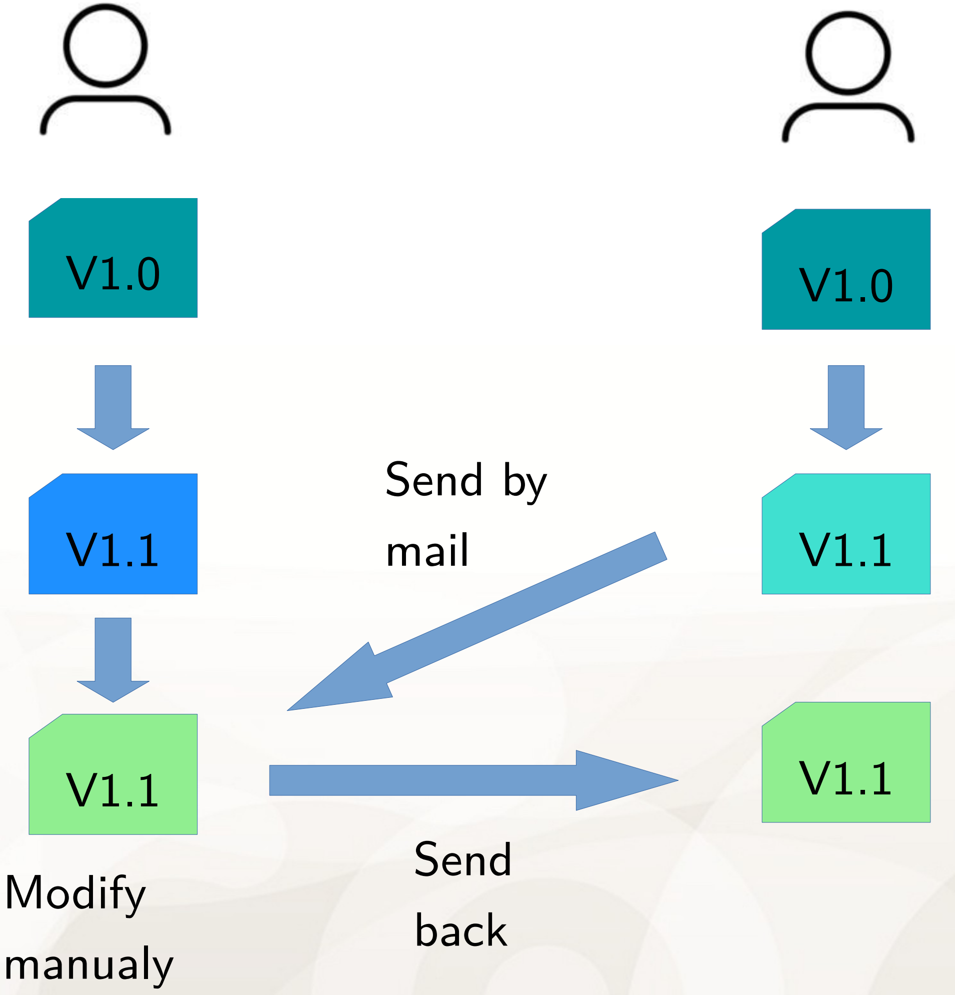
Git, GitHub, Gitlab

Version control system

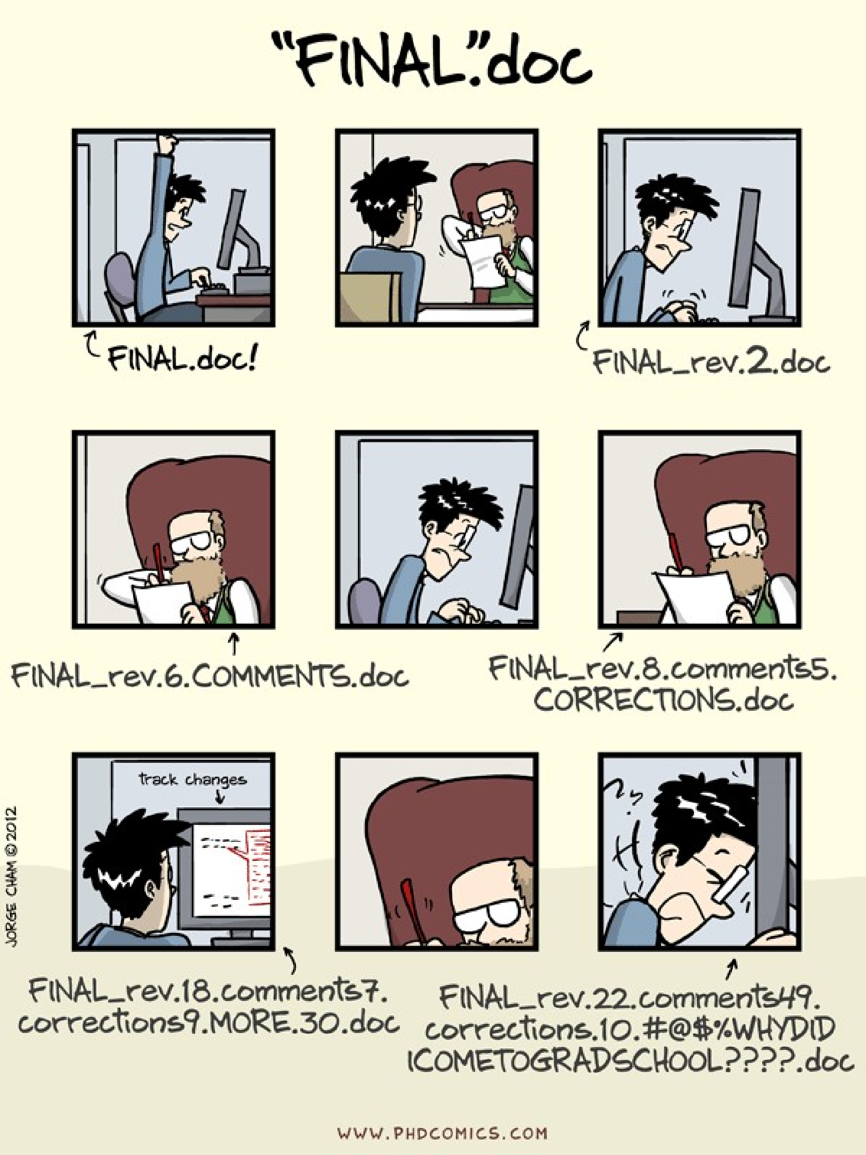
Online hosting service

for software base on git

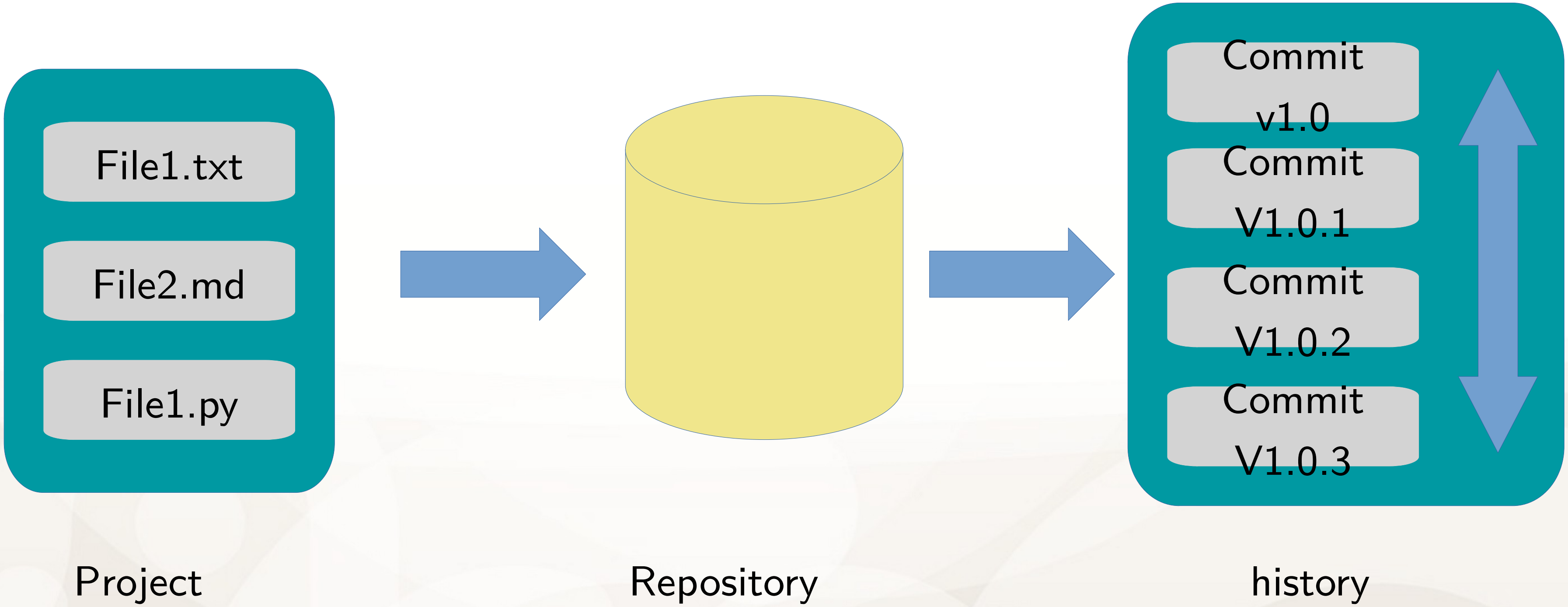
Sharing code



Sharing code



Git a version control system



File1.txt

File2.md

File1.py

Commit v1.0

Commit V1.0.1

Commit V1.0.2

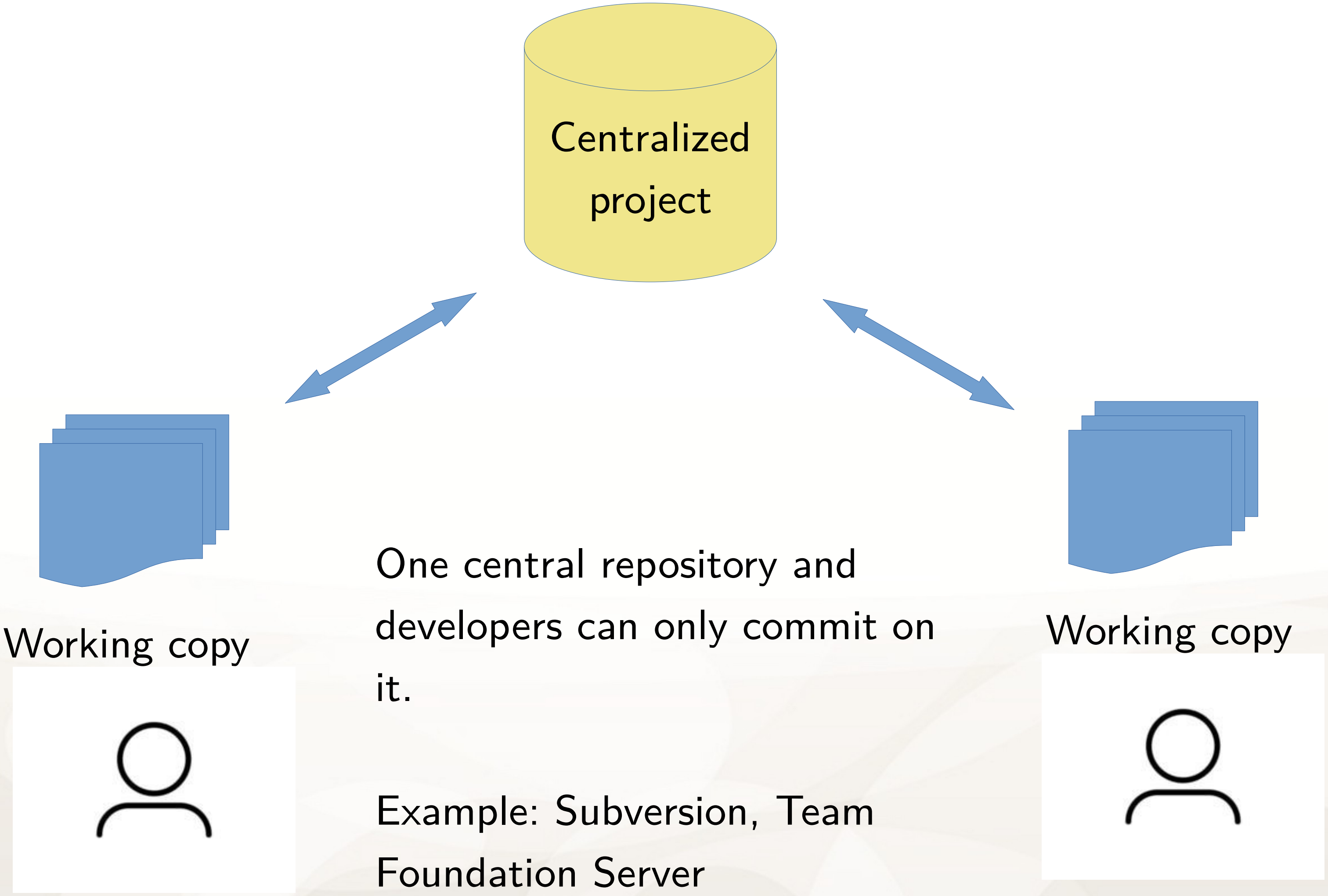
Commit V1.0.3

Project

Repository

history

Centralized version control



Centralized project

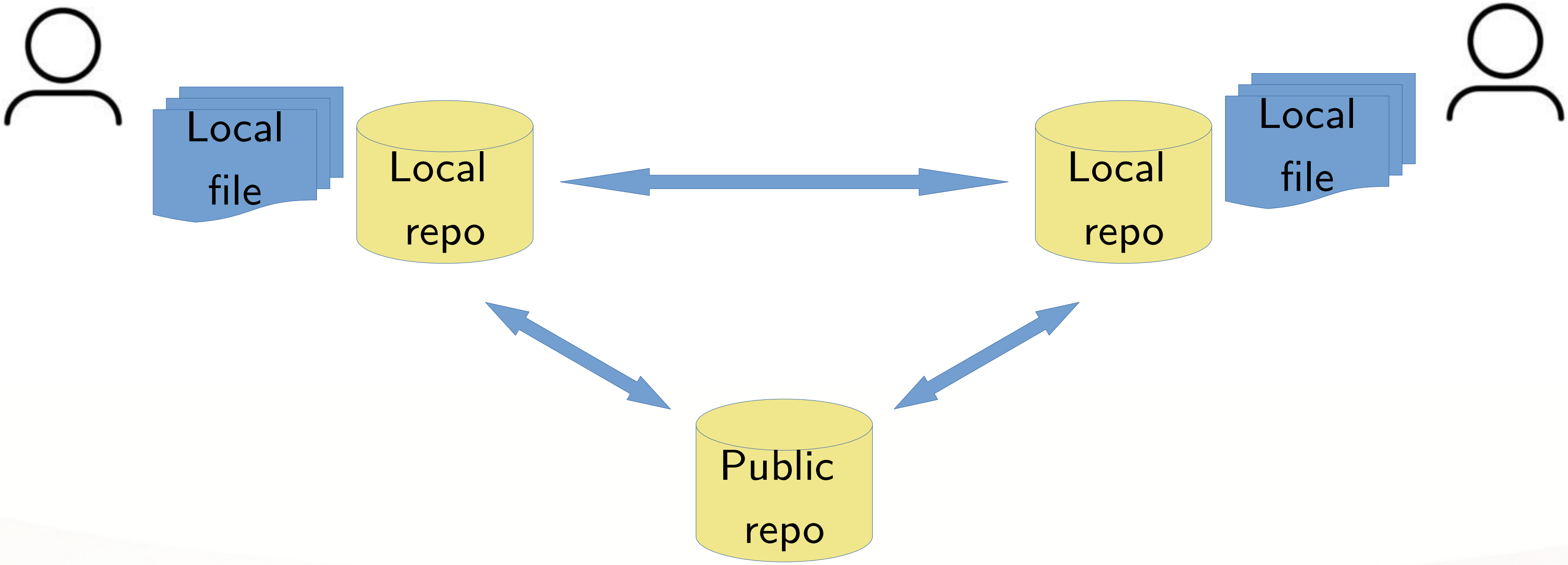
Working copy

Working copy

One central repository and developers can only commit on it.

Example: Subversion, Team Foundation Server

Distributed version control



All repository are interconnected and can be edited at same time.

Example: Git, Mercurial

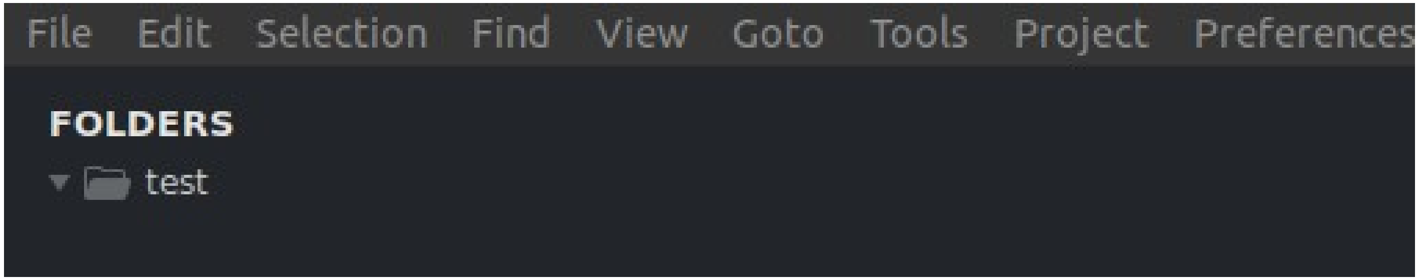
Git

Git is a free and open source distributed version control system.

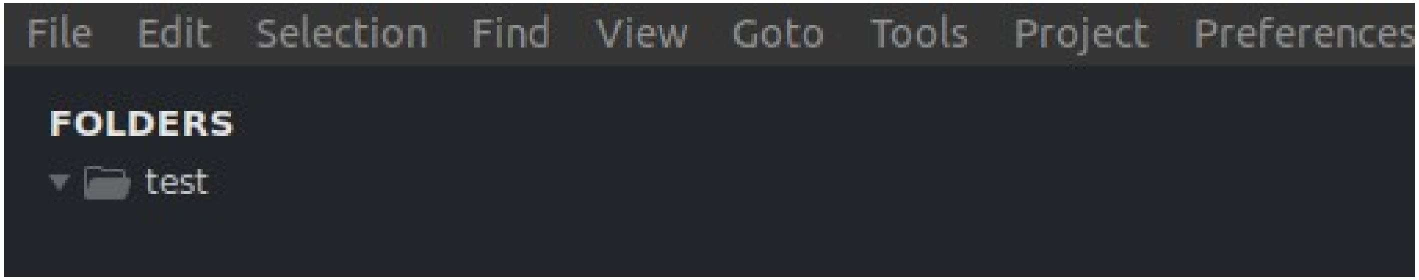
Handle any type of project, fast and efficiency.

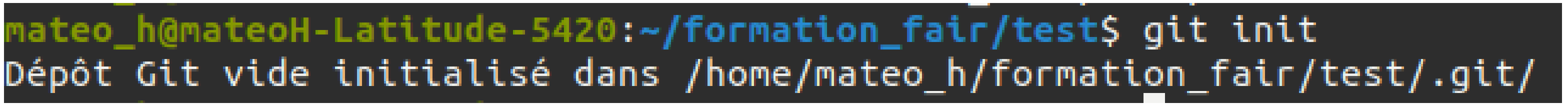
Created in 2005 by Linus Torvalds

Git: Initialisation



Git: Initialisation



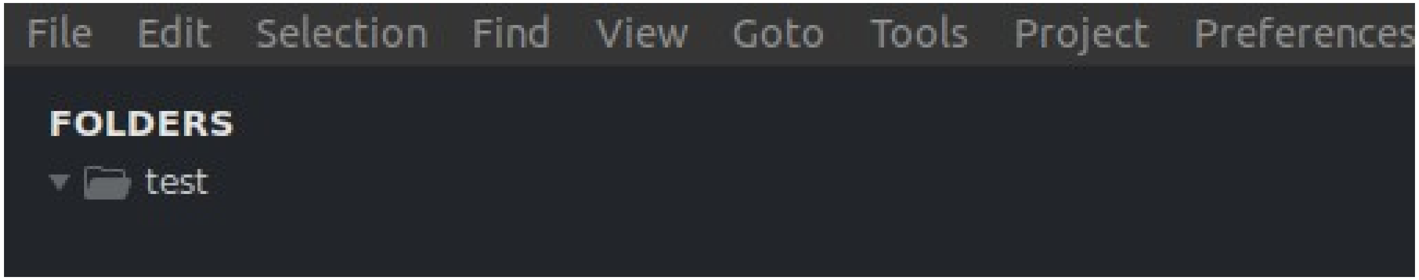


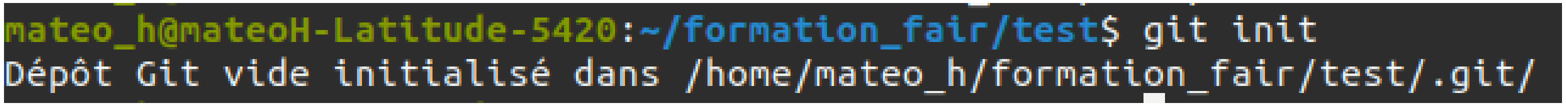
```
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git init
Dépôt Git vide initialisé dans /home/mateo_h/formation_fair/test/.git/
```

“Git init “ to initialize a git repository in any folder

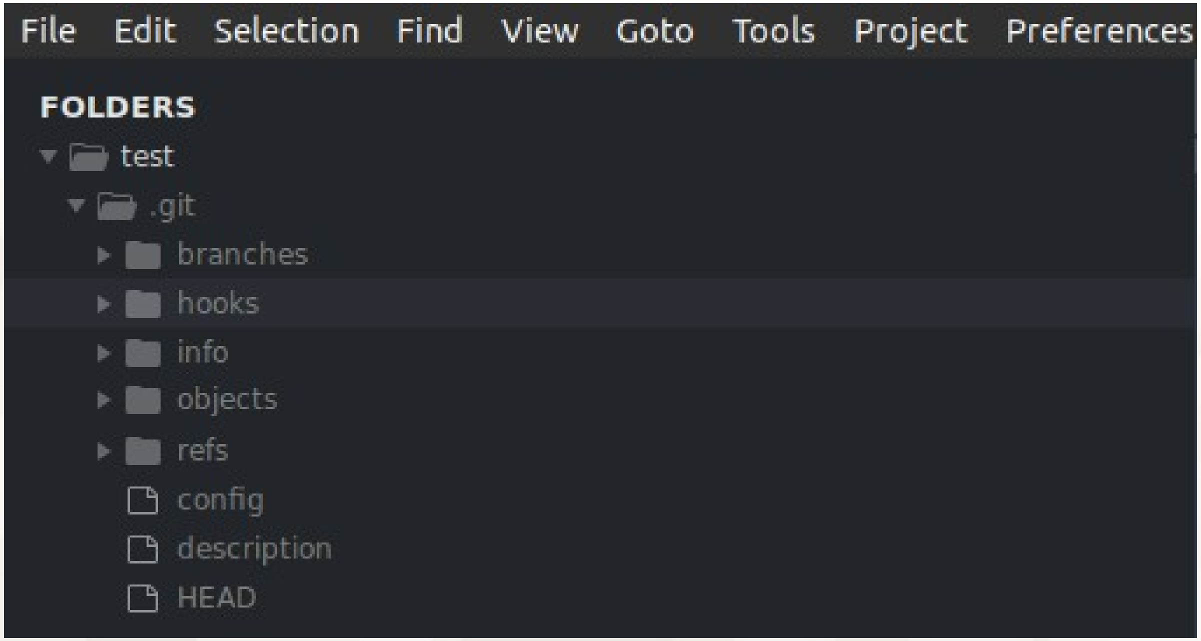
“Git clone” can also be used to import an existing repository

Git: Initialisation





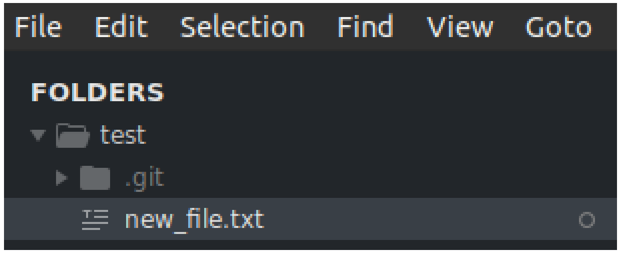
```
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git init
Dépôt Git vide initialisé dans /home/mateo_h/formation_fair/test/.git/
```



“Git init “ to initialize a git repository in any folder

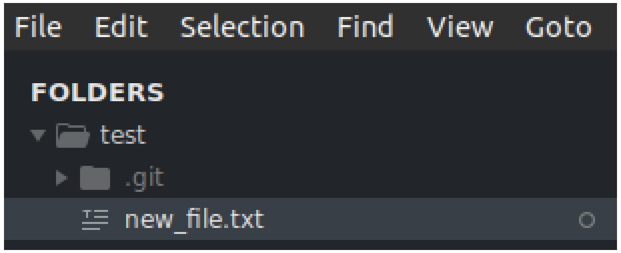
“Git clone” can also be used to import an existing repository

Git: Commit

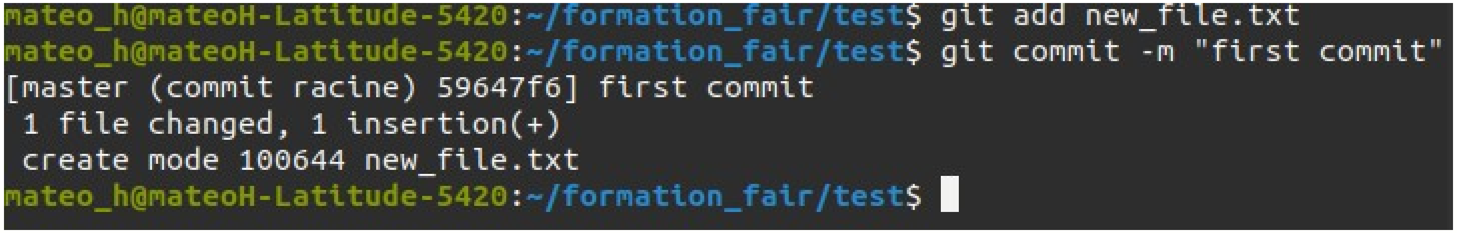


Create new file in project

Git: Commit

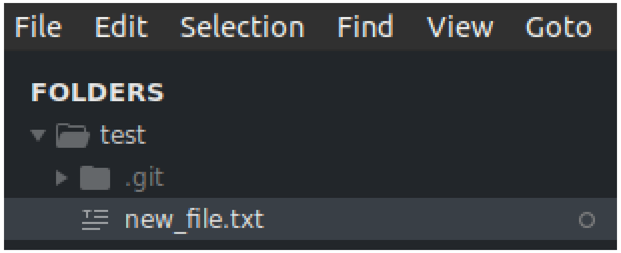


Create new file in project

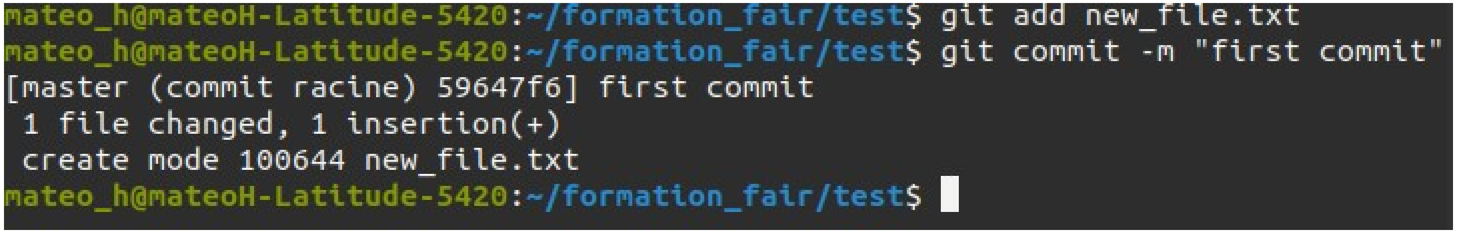


```
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git add new_file.txt
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git commit -m "first commit"
[master (commit racine) 59647f6] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 new_file.txt
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ 
```

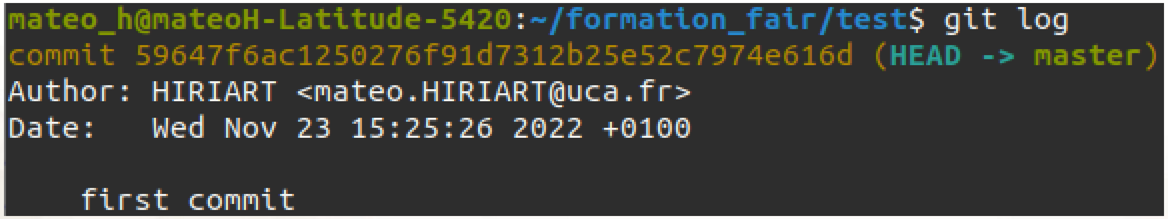
Git: Commit



Create new file in project

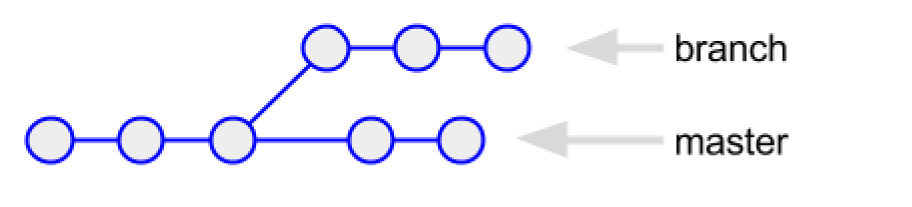


```
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git add new_file.txt
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git commit -m "first commit"
[master (commit racine) 59647f6] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 new_file.txt
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ 
```



```
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git log
commit 59647f6ac1250276f91d7312b25e52c7974e616d (HEAD -> master)
Author: HIRIART <mateo.HIRIART@uca.fr>
Date:   Wed Nov 23 15:25:26 2022 +0100

    first commit
```



Commit are snapshot of your files (code)

They all have a unique id

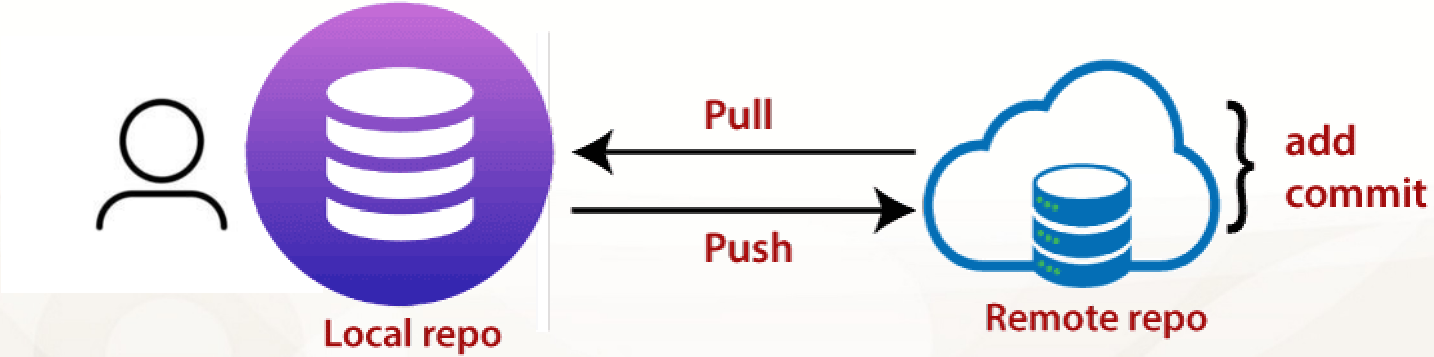
And are linked to their parent

Can always go back to previous commit

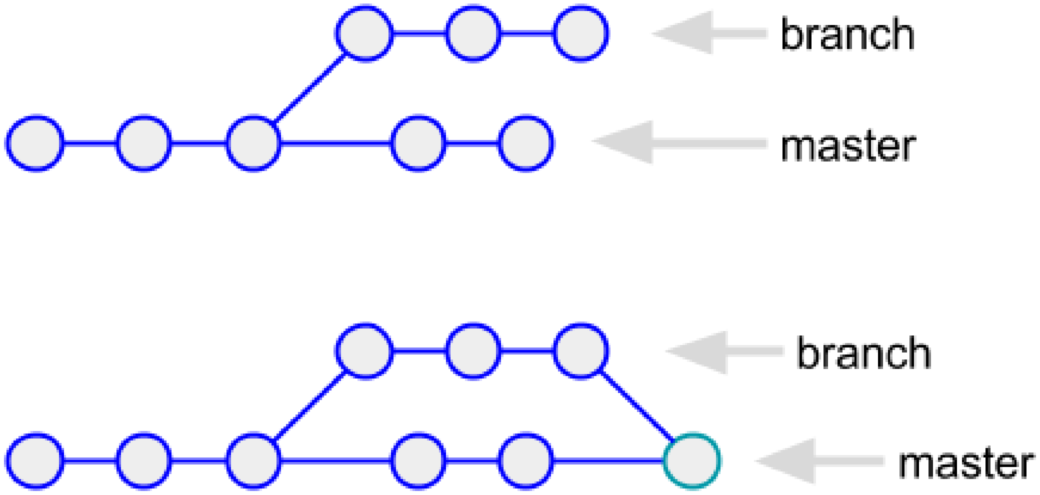
Git: Push and Pull

Git pull import modification form another repository if it have same origin.

Git push export modification to another repository if it have same origin.



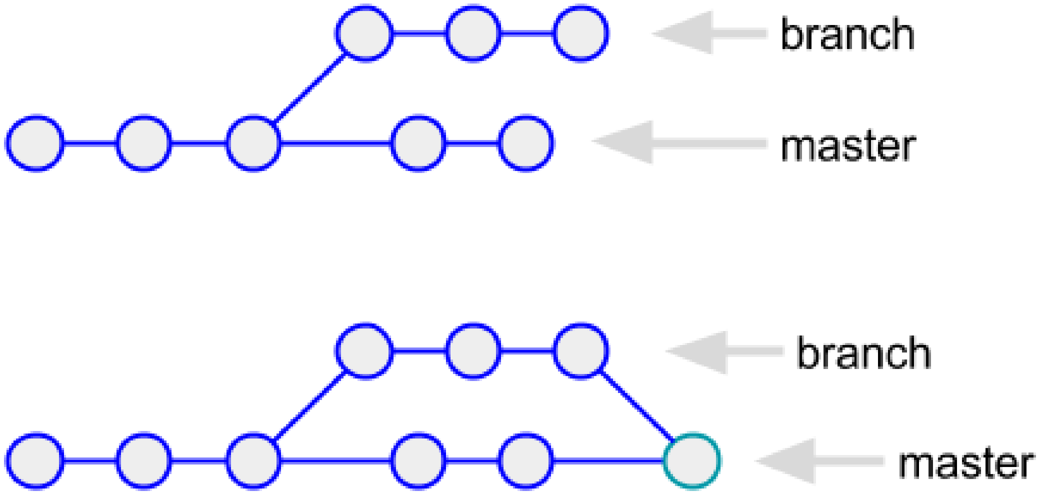
Git: branches



Git branch allow parallel development

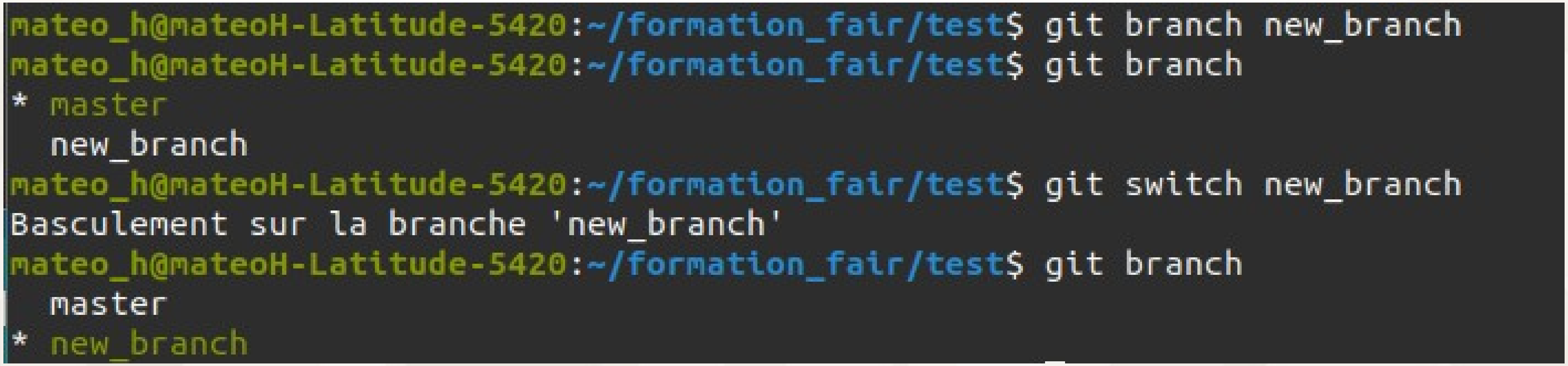
And still apply it latter with code control

Git: branches



Git branch allow parallel development

And still apply it latter with code control

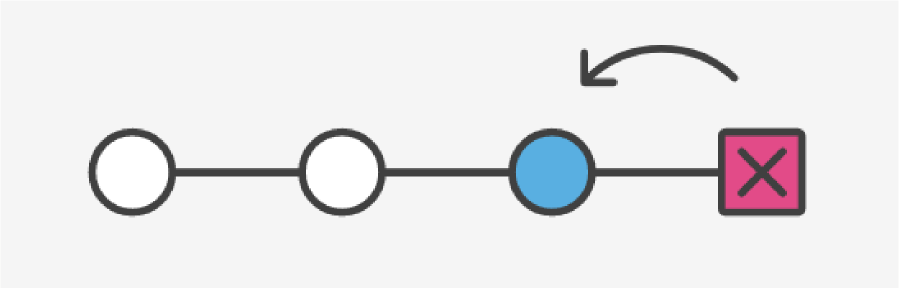


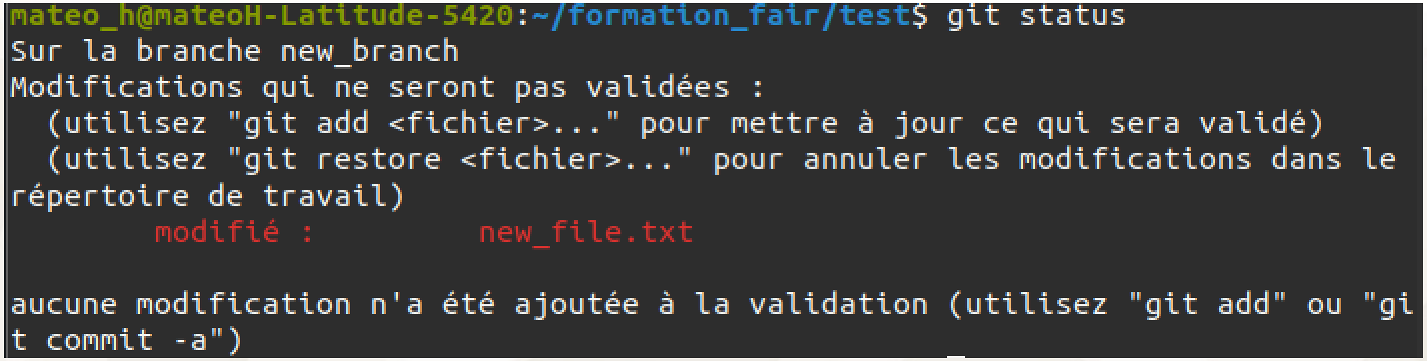
```
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git branch new_branch
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git branch
* master
  new_branch
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git switch new_branch
Basculement sur la branche 'new_branch'
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git branch
  master
* new_branch
```

Git: Revert

Git reset: reset repository back to the specified HEAD

Git restore: reset current modification





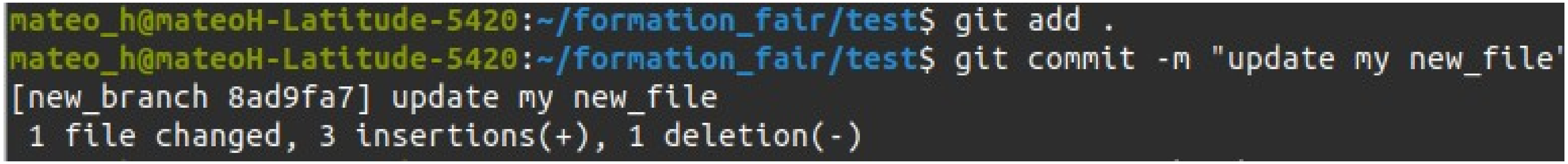
```
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git status
Sur la branche new_branch
Modifications qui ne seront pas validées :
  (utilisez "git add <fichier>..." pour mettre à jour ce qui sera validé)
  (utilisez "git restore <fichier>..." pour annuler les modifications dans le
répertoire de travail)
        modifié :         new_file.txt

aucune modification n'a été ajoutée à la validation (utilisez "git add" ou "gi
t commit -a")
```

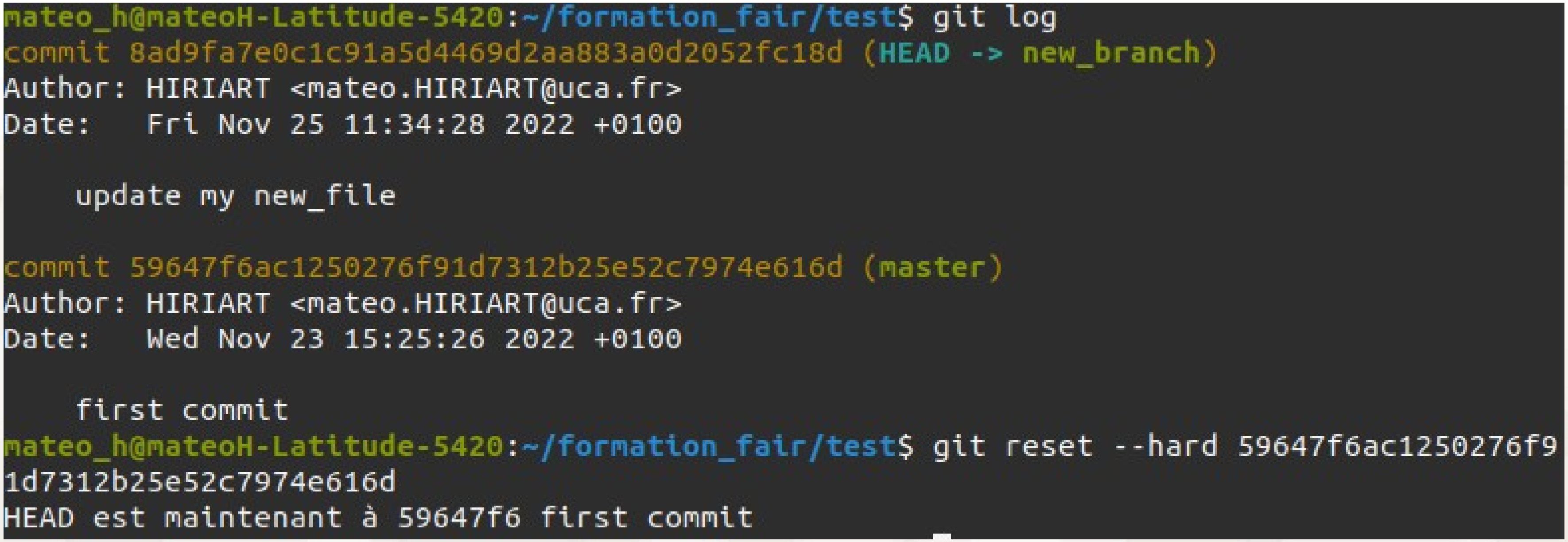
Git: Revert

Git reset: reset repository back to the specified commit

Git restore: reset current modification



```
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git add .
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git commit -m "update my new_file'
[new_branch 8ad9fa7] update my new_file
 1 file changed, 3 insertions(+), 1 deletion(-)
```



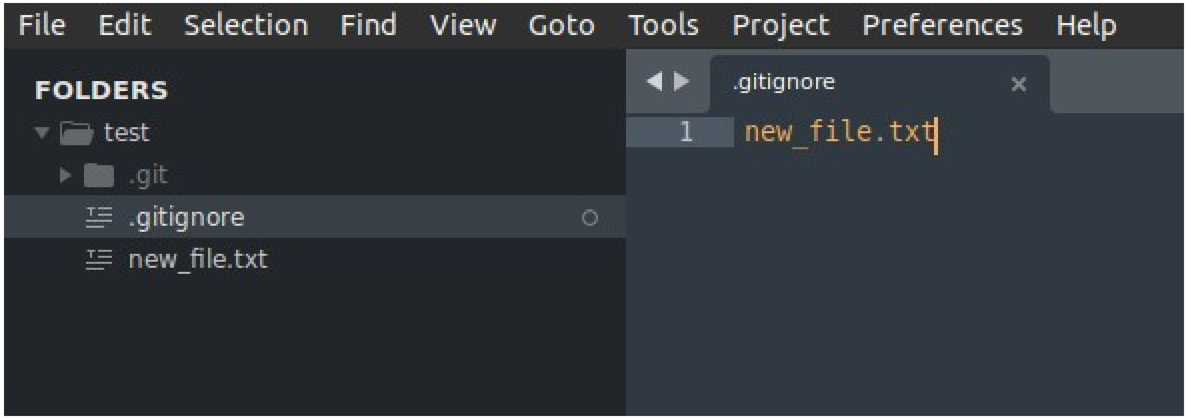
```
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git log
commit 8ad9fa7e0c1c91a5d4469d2aa883a0d2052fc18d (HEAD -> new_branch)
Author: HIRIART <mateo.HIRIART@uca.fr>
Date:   Fri Nov 25 11:34:28 2022 +0100

    update my new_file

commit 59647f6ac1250276f91d7312b25e52c7974e616d (master)
Author: HIRIART <mateo.HIRIART@uca.fr>
Date:   Wed Nov 23 15:25:26 2022 +0100

    first commit
mateo_h@mateoH-Latitude-5420:~/formation_fair/test$ git reset --hard 59647f6ac1250276f9
1d7312b25e52c7974e616d
HEAD est maintenant à 59647f6 first commit
```

Git: .gitignore



```
new_file.txt
```

Specific file that tell git to untrack any files listed inside.

Very useful especially with project that required other packages or if you have any file that store sensitive data.

Git: Useful command

Git help: display multiple git command with a description

Git log: display commit history of the current branch

Git status: display current state off your local repository

Git diff: display code difference between current local repository and HEAD

Documentation link for git command:

https://git-scm.com/doc

https://docs.github.com/fr

Github and Gitlab

Public repository hosting server.

Both free version with pricing for enhance features.

GitHub

Released in 2008

Owned by microsoft

Fast, more popular

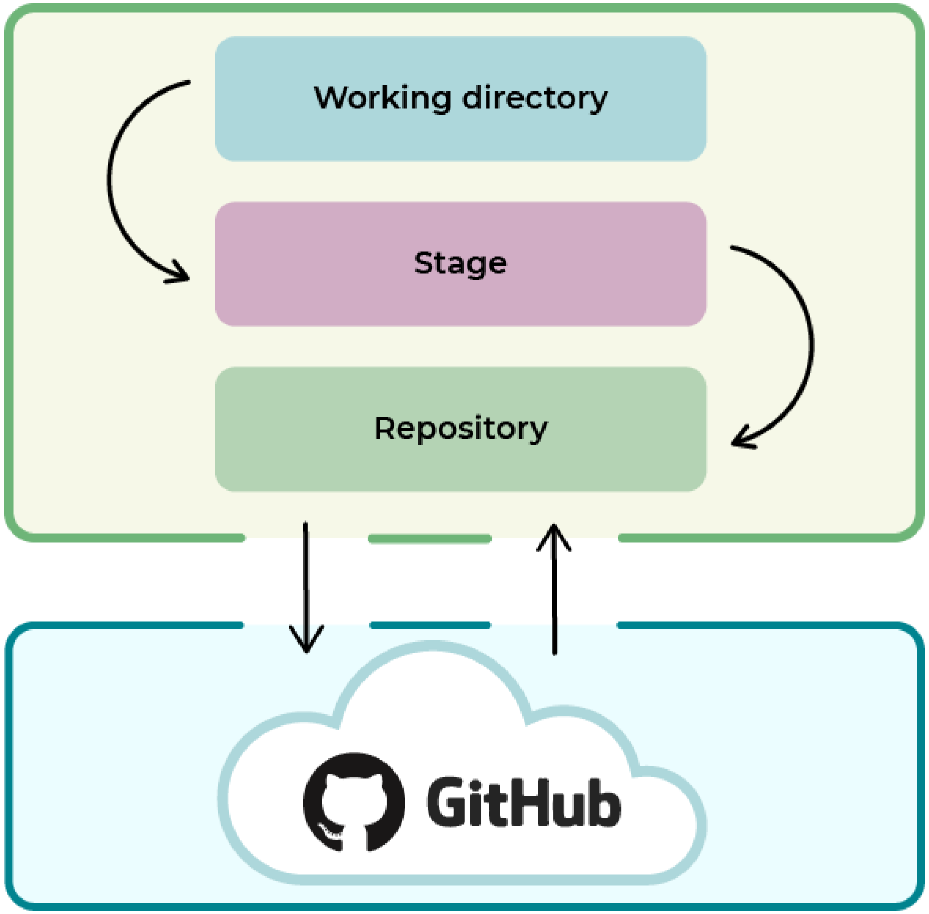
GitLab

Released in 2014

Open source

Built-in CI/CD, Less restriction in free version

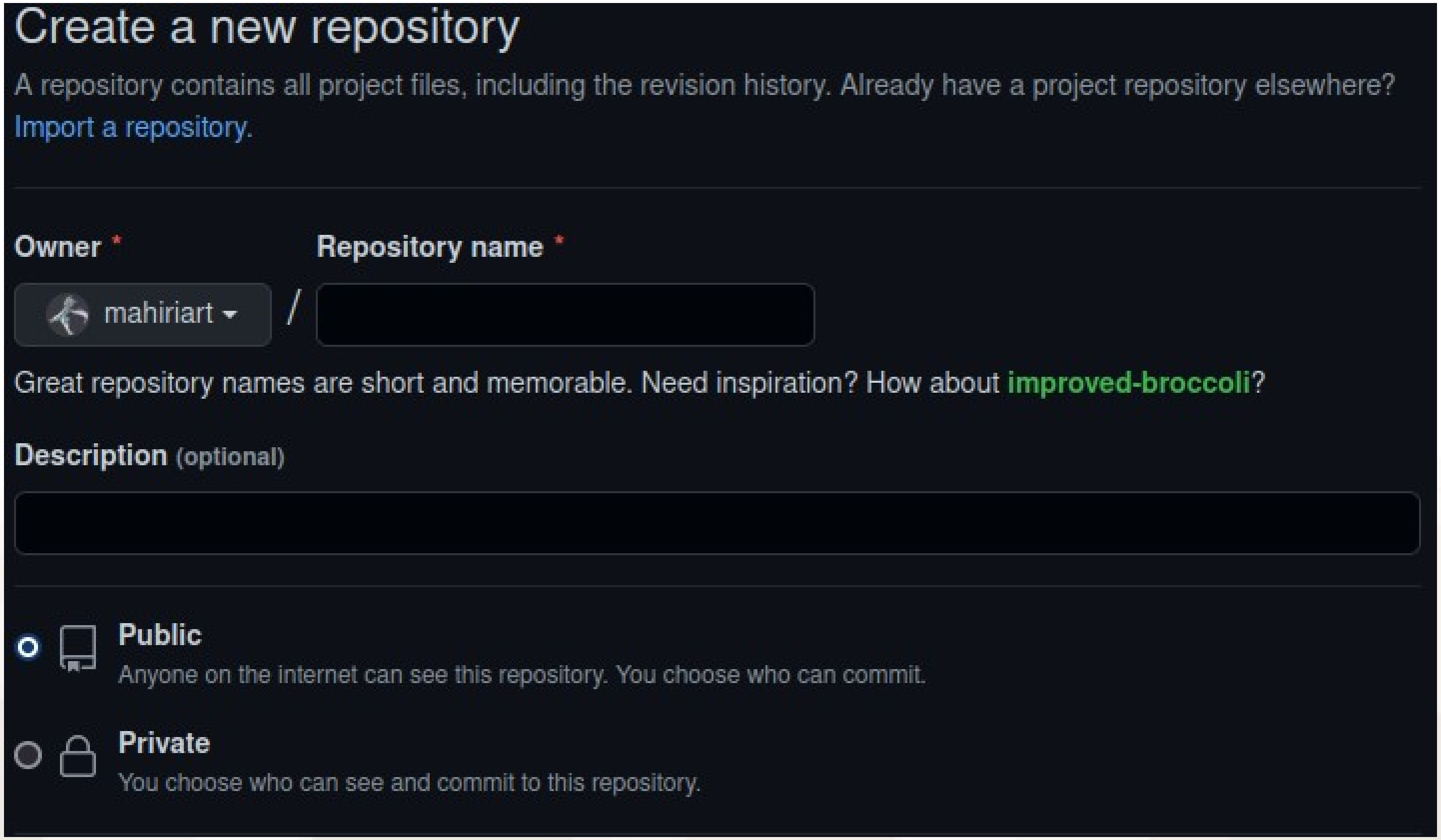
Github and Gitlab



Dépôt local

Dépôt distant

Create your own repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Repository name *

mahiriart ▾ /

Great repository names are short and memorable. Need inspiration? How about improved-broccoli?

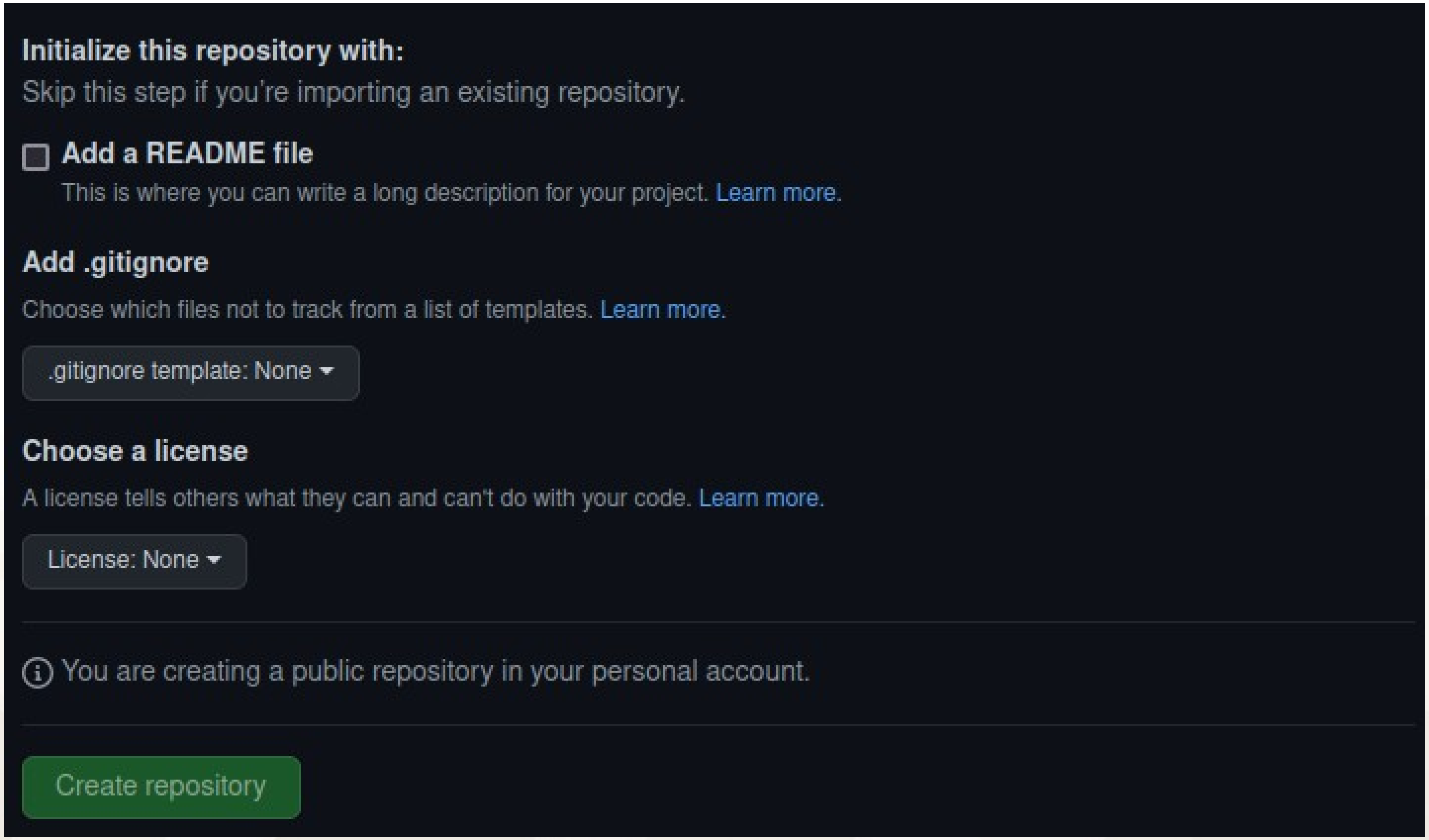
Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Github/Gitlab: Basics

Create your own repository, with base features



Initialize this repository with:

Skip this step if you're importing an existing repository.

□ **Add a README file**

This is where you can write a long description for your project. Learn more.

Add .gitignore

Choose which files not to track from a list of templates. Learn more.

.gitignore template: None ▾

Choose a license

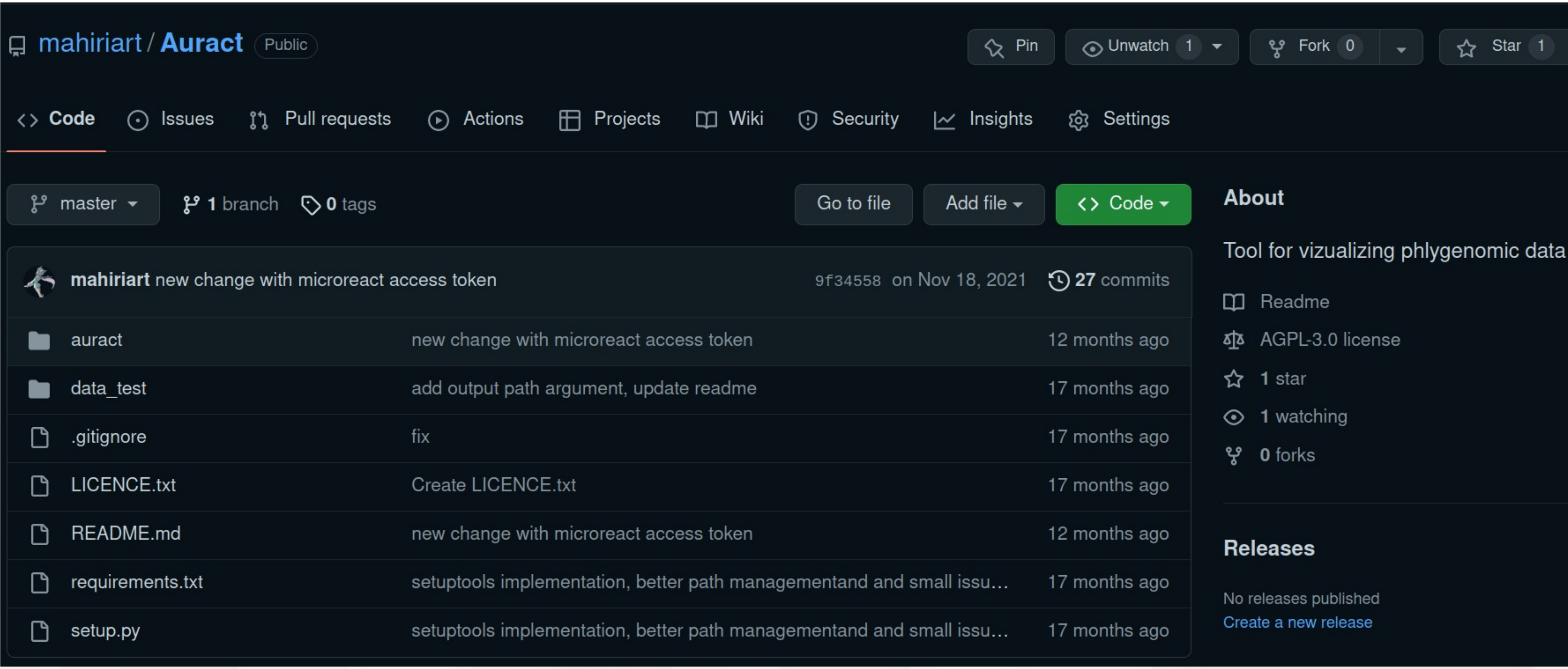
A license tells others what they can and can't do with your code. Learn more.

License: None ▾

ⓘ You are creating a public repository in your personal account.

Github/Gitlab: Basics

Main page of a repository



mahiriart new change with microreact access token 9f34558 on Nov 18, 2021 27 commits

auract	new change with microreact access token	12 months ago
data_test	add output path argument, update readme	17 months ago
.gitignore	fix	17 months ago
LICENCE.txt	Create LICENCE.txt	17 months ago
README.md	new change with microreact access token	12 months ago
requirements.txt	setuptools implementation, better path managementand and small issu…	17 months ago
setup.py	setuptools implementation, better path managementand and small issu…	17 months ago

About

Tool for vizualizing phylgenomic data

- Readme
- AGPL-3.0 license
- 1 star
- 1 watching
- 0 forks

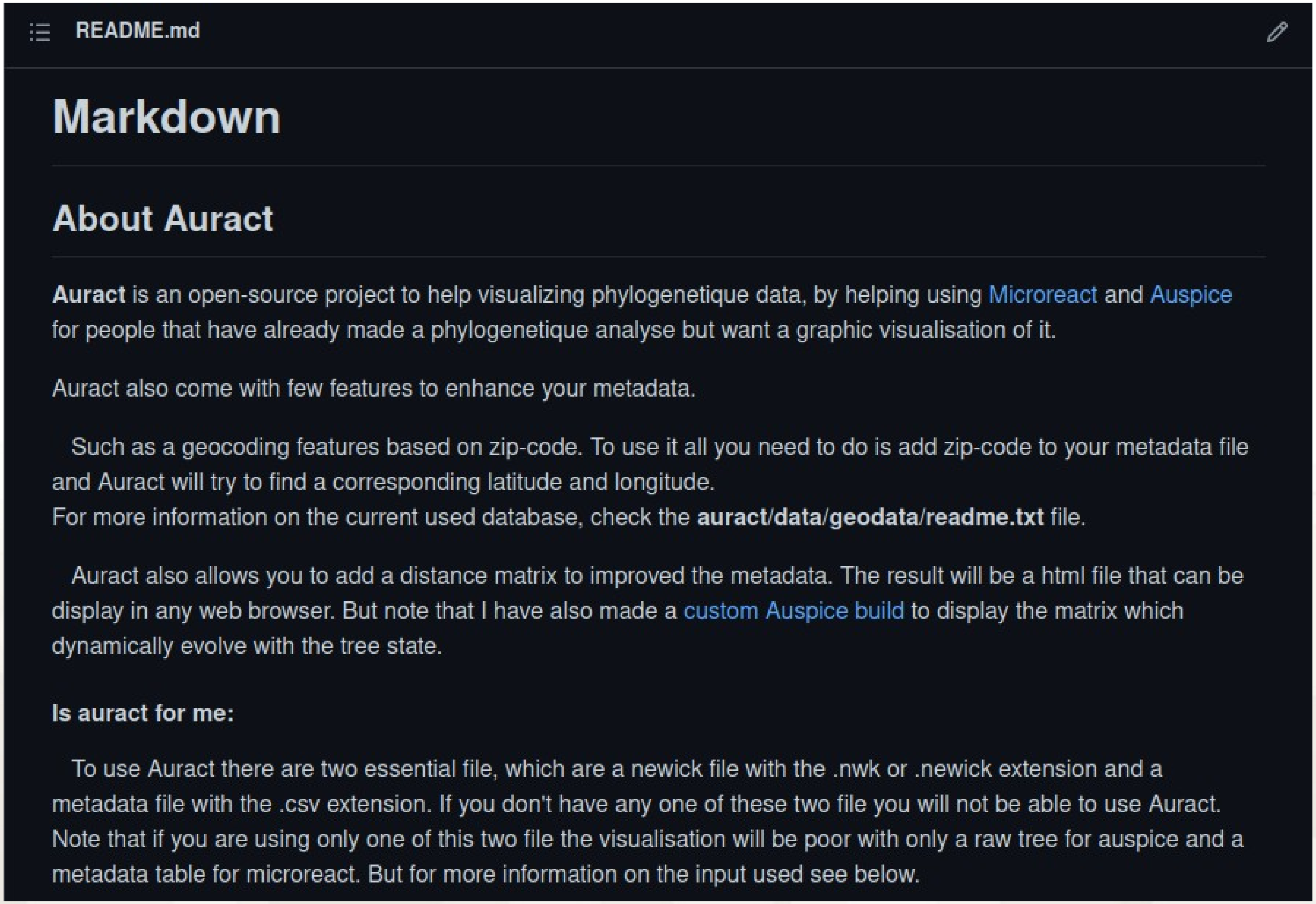
Releases

No releases published

Create a new release

Github/Gitlab: Improvement

Readme usually present the repository and also have documentation or guide on installation and usage.



README.md

Markdown

About Auract

Auract is an open-source project to help visualizing phylogenetique data, by helping using Microreact and Auspice for people that have already made a phylogenetique analyse but want a graphic visualisation of it.

Auract also come with few features to enhance your metadata.

Such as a geocoding features based on zip-code. To use it all you need to do is add zip-code to your metadata file and Auract will try to find a corresponding latitude and longitude.
For more information on the current used database, check the **auract/data/geodata/readme.txt** file.

Auract also allows you to add a distance matrix to improved the metadata. The result will be a html file that can be display in any web browser. But note that I have also made a custom Auspice build to display the matrix which dynamically evolve with the tree state.

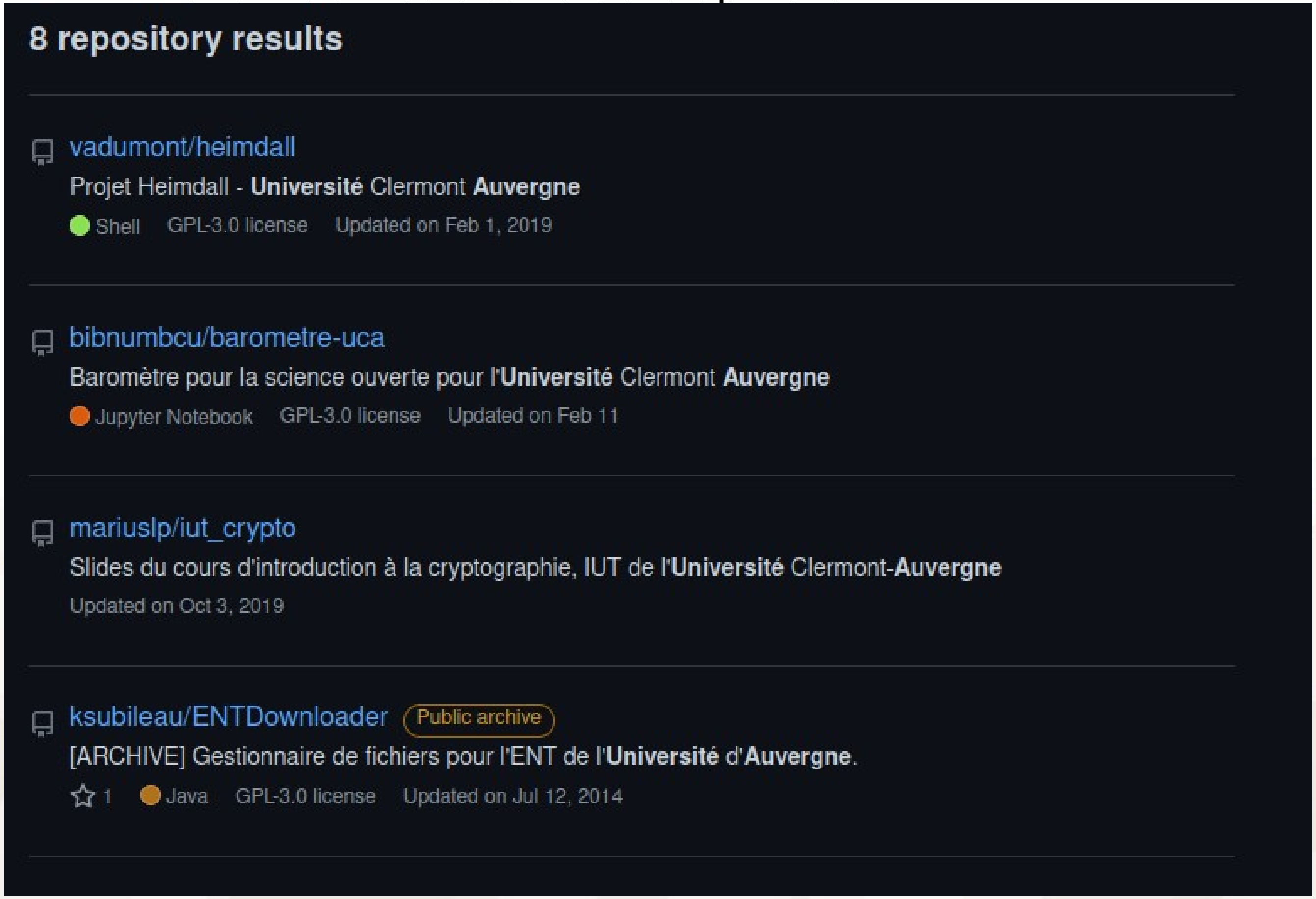
Is auract for me:

To use Auract there are two essential file, which are a newick file with the .nwk or .newick extension and a metadata file with the .csv extension. If you don't have any one of these two file you will not be able to use Auract. Note that if you are using only one of this two file the visualisation will be poor with only a raw tree for auspice and a metadata table for microreact. But for more information on the input used see below.

Github/Gitlab: Open source project

Find open source project and publish yours

Worldwide interactive development



8 repository results

vadumont/heimdall
Projet Heimdall - **Université** Clermont **Auvergne**
● Shell GPL-3.0 license Updated on Feb 1, 2019

bibnumbcu/barometre-uca
Baromètre pour la science ouverte pour l'**Université** Clermont **Auvergne**
● Jupyter Notebook GPL-3.0 license Updated on Feb 11

mariuslp/iut_crypto
Slides du cours d'introduction à la cryptographie, IUT de l'**Université** Clermont-**Auvergne**
Updated on Oct 3, 2019

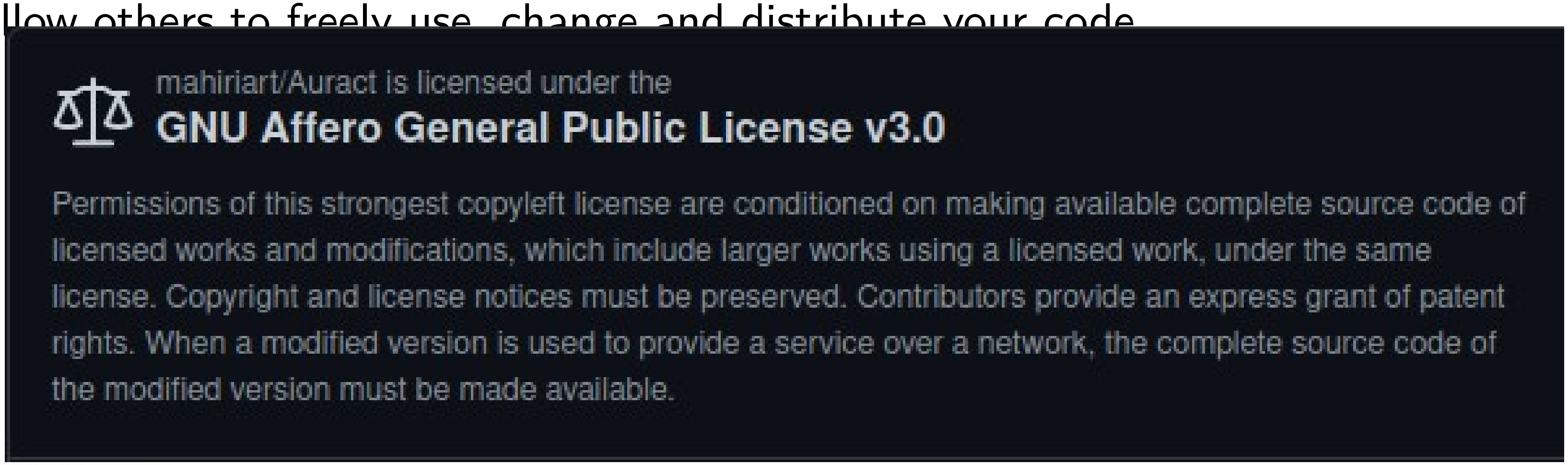
ksubileau/ENTDownloader Public archive
[ARCHIVE] Gestionnaire de fichiers pour l'ENT de l'**Université** d'**Auvergne**.
☆ 1 ● Java GPL-3.0 license Updated on Jul 12, 2014

UNIVERSITÉ Clermont Auvergne

Github/Gitlab: Licenses

Key element for public repository, allow then to truly be open source.

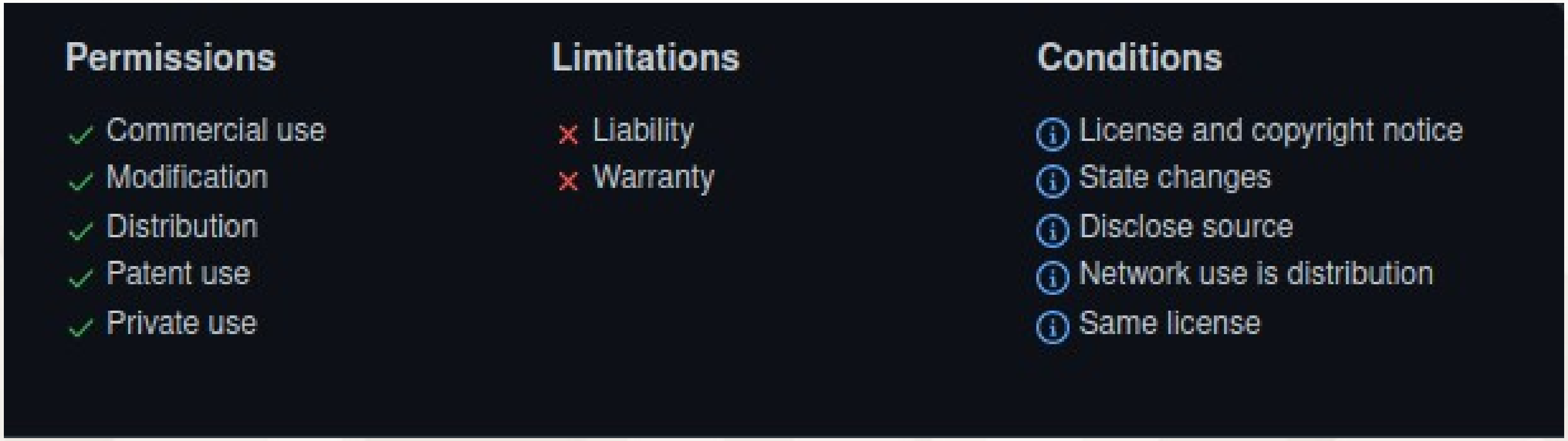
Allow others to freely use, change and distribute your code.



mahiriart/Auract is licensed under the

GNU Affero General Public License v3.0

Permissions of this strongest copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. When a modified version is used to provide a service over a network, the complete source code of the modified version must be made available.



Permissions

- ✓ Commercial use
- ✓ Modification
- ✓ Distribution
- ✓ Patent use
- ✓ Private use

Limitations

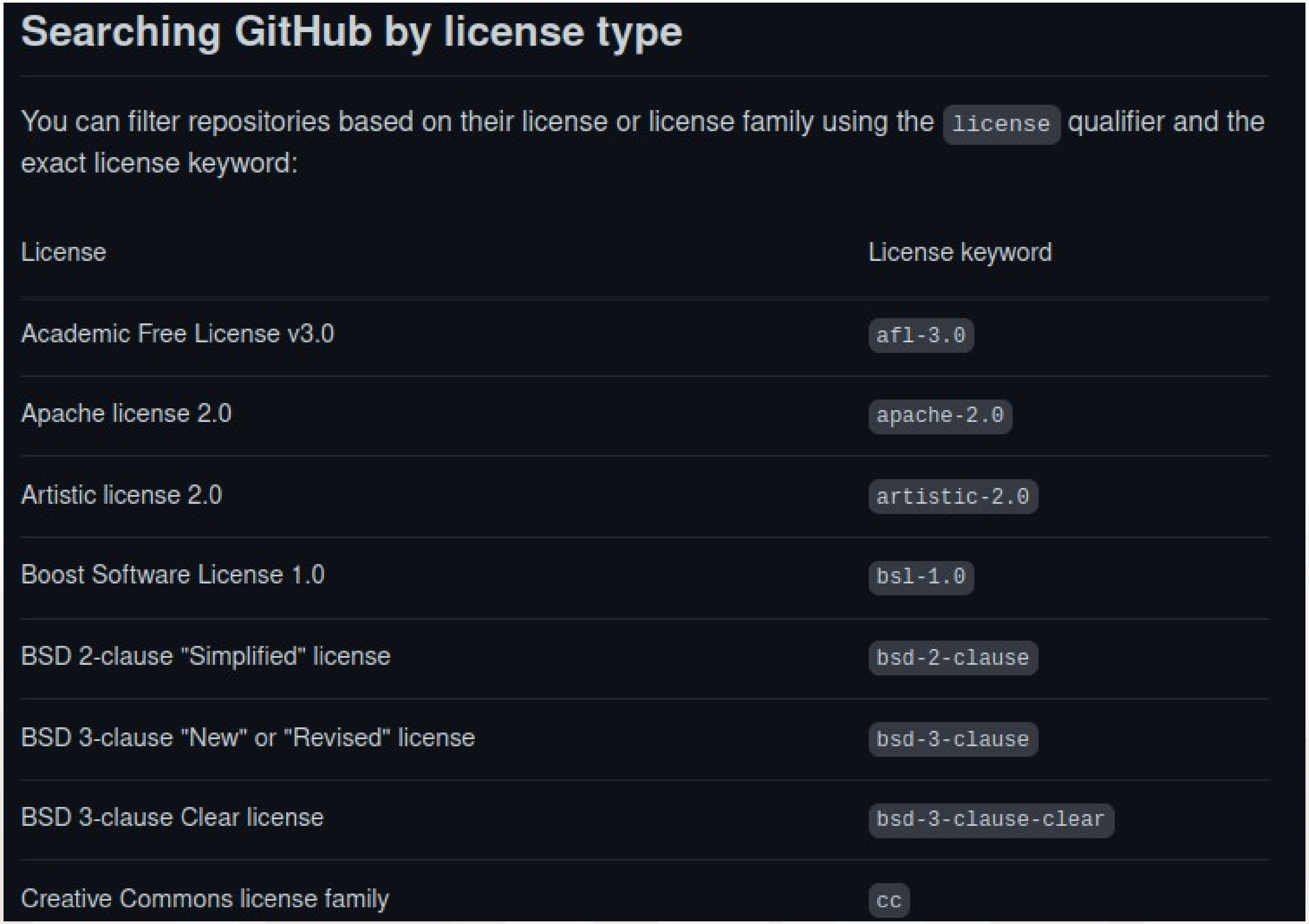
- ✗ Liability
- ✗ Warranty

Conditions

- ⓘ License and copyright notice
- ⓘ State changes
- ⓘ Disclose source
- ⓘ Network use is distribution
- ⓘ Same license

Github/Gitlab: Licenses

Many licences available



Searching GitHub by license type

You can filter repositories based on their license or license family using the `license` qualifier and the exact license keyword:

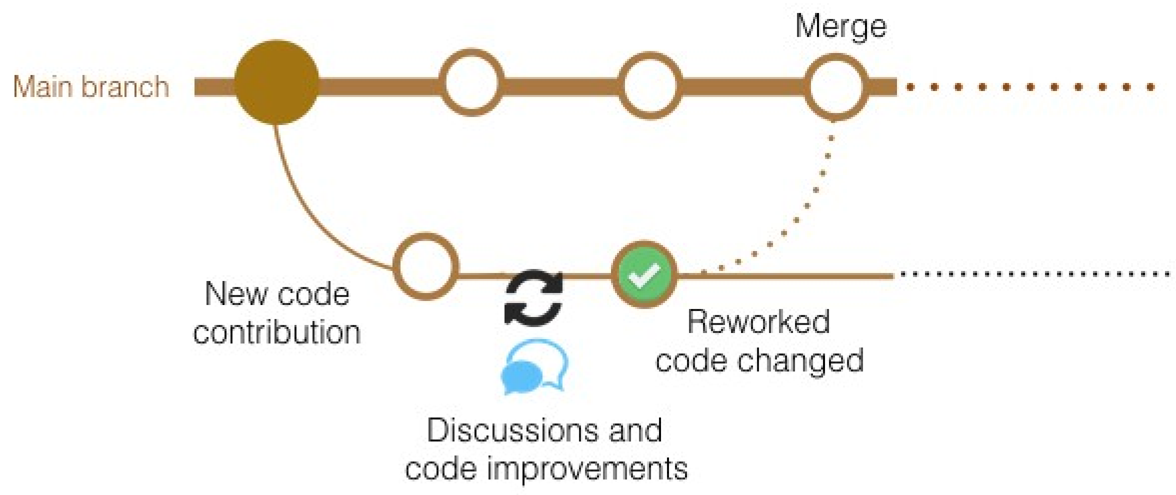
License	License keyword
Academic Free License v3.0	`afl-3.0`
Apache license 2.0	`apache-2.0`
Artistic license 2.0	`artistic-2.0`
Boost Software License 1.0	`bsl-1.0`
BSD 2-clause "Simplified" license	`bsd-2-clause`
BSD 3-clause "New" or "Revised" license	`bsd-3-clause`
BSD 3-clause Clear license	`bsd-3-clause-clear`
Creative Commons license family	`cc`

Github/Gitlab: Issues and pull request

Issues are features to point out problem in code or possible enhancement.

Usually solved in a pull request.

Pull request are code contribution in a review state.



Simplified Pull Request process

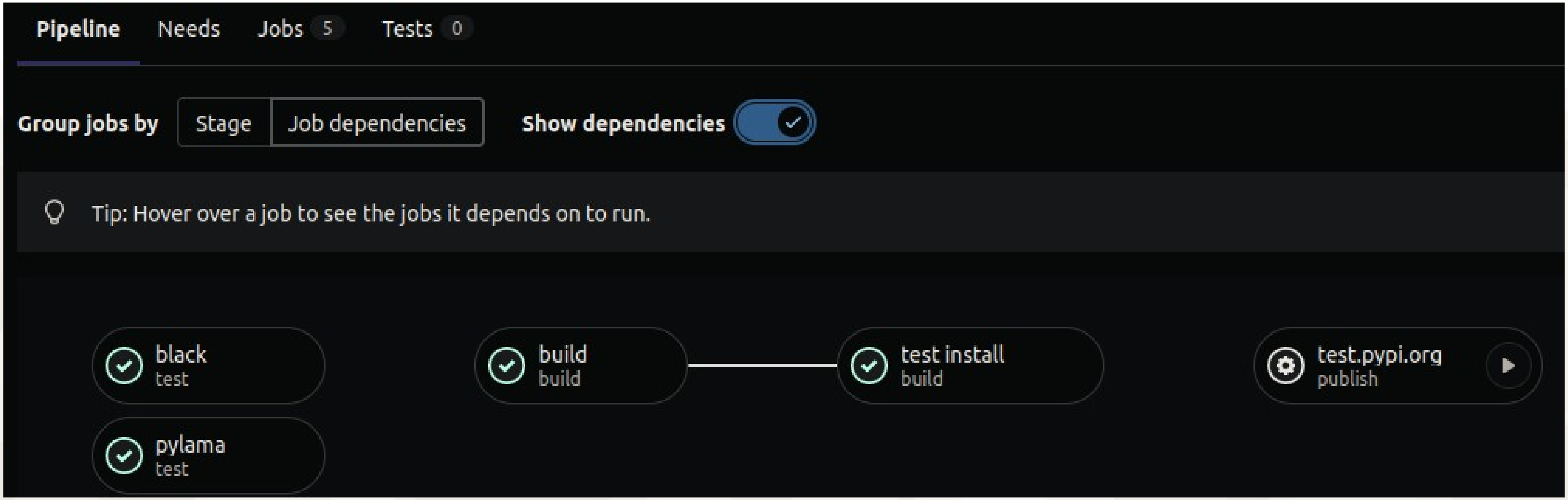
Github/Gitlab: Pipeline CI/CD

Continuous Integration (CI)

Continuous Delivery (CD)

Can automatically build, test, deploy, and monitor your applications

Example of CI pipeline



Group jobs by: Job dependencies

Show dependencies: ✓

Tip: Hover over a job to see the jobs it depends on to run.

Github/Gitlab: Pages

Easily build own web site for your project. To highlight your work.

Hosted by Github or Gitlab.

Multiple template can be found.

https://pages.gitlab.io/hugo/

End

https://github.com/mesocentre-clermont-auvergne/formation_fair_2022/tree/main/fair_git