Git/github tutorial:

BrainHack Marseille 28/11/2022

David Meunier,
Institut de Neurosciences de la Timone, Marseille
david.meunier@univ-amu.fr

Context

When you code with someone else (your PhD director, a collaborator, a student) :

Context

When you code with someone else (your PhD director, a collaborator, a student) :

A single place where the « good » code is stored

Git

What is git :

→ successor of previous versioning system : CVS and SVN

→ git allows for multiple people to work together on a project : collaborative development

→ Allows for **versionning** = the successive version of a project can be retrieved from a single place.

not restricted to source code : anything that can be shared can be ‘gitified’ (website, tutorial)

BUT not data (other solutions) and article (with the restriction maybe to latex versions) → to be discussed

→ git also allows to make branch (i.e. each developer works on a subpart of the project) and the allows ‘automatic’ merging of all the branches

Git

What is git :

→ successor of previous versioning system : CVS and SVN

→ git allows for multiple people to work together on a project : collaborative development

→ Allows for **versionning** = the successive version of a project can be retrieved from a single place.

not restricted to source code : anything that can be shared can be ‘gitified’ (website, tutorial)

BUT not data (other solutions) and article (with the restriction maybe to latex versions) → to be discussed

→ git also allows to make branch (i.e. each developer works on a subpart of the project) and the allows ‘automatic’ merging of all the branches

Why use git ?

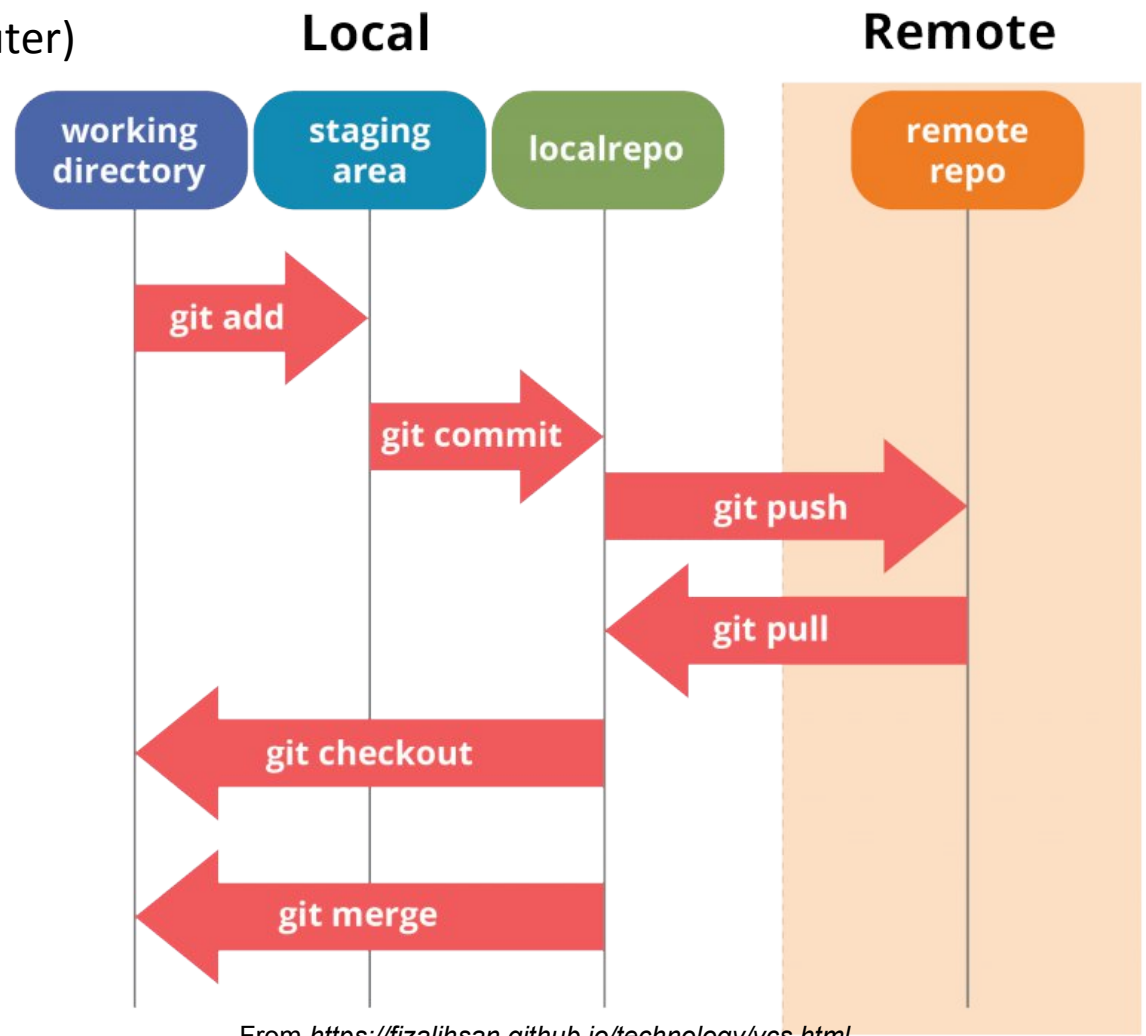
Reproducible science: from the data (**open data**) and the code (**open source**), then you should *in principle* be able to reproduce the results (**open science**)

Git

Basic principles:

→ a repository (a repo) is a place on a **remote** (i.e. not on your computer) where the « project » is stored.

→ The project on the server is the « good » version, and the « **local** » copy to be modified and ongoing work



From *https://fizalihsan.github.io/technology/vcs.html*

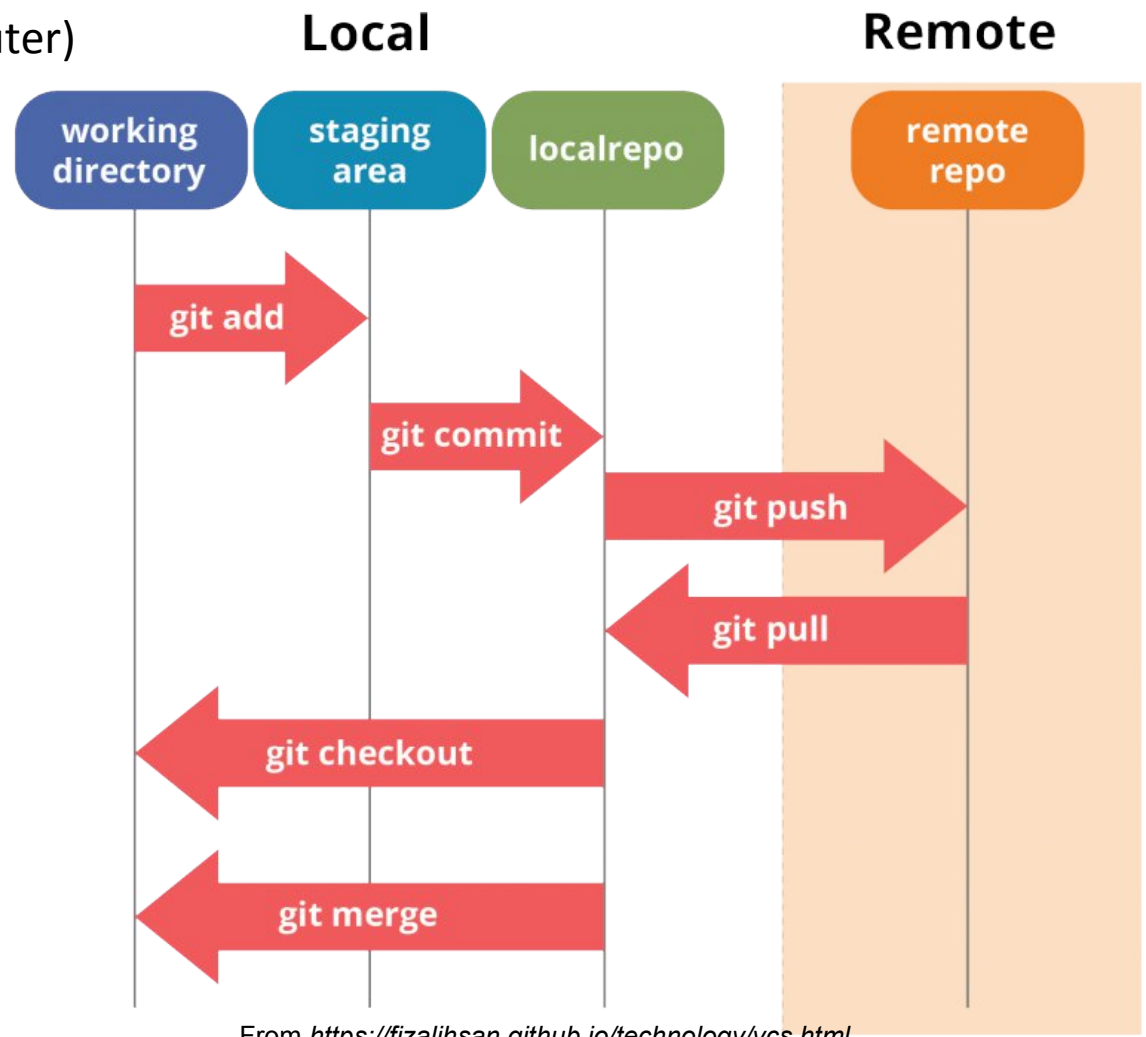
Git

Basic principles:

→ a repository (a repo) is a place on a **remote** (i.e. not on your computer) where the « project » is stored.

→ The project on the server is the « good » version, and the « **local** » copy to be modified and ongoing work

By default, the remote is called « origin »



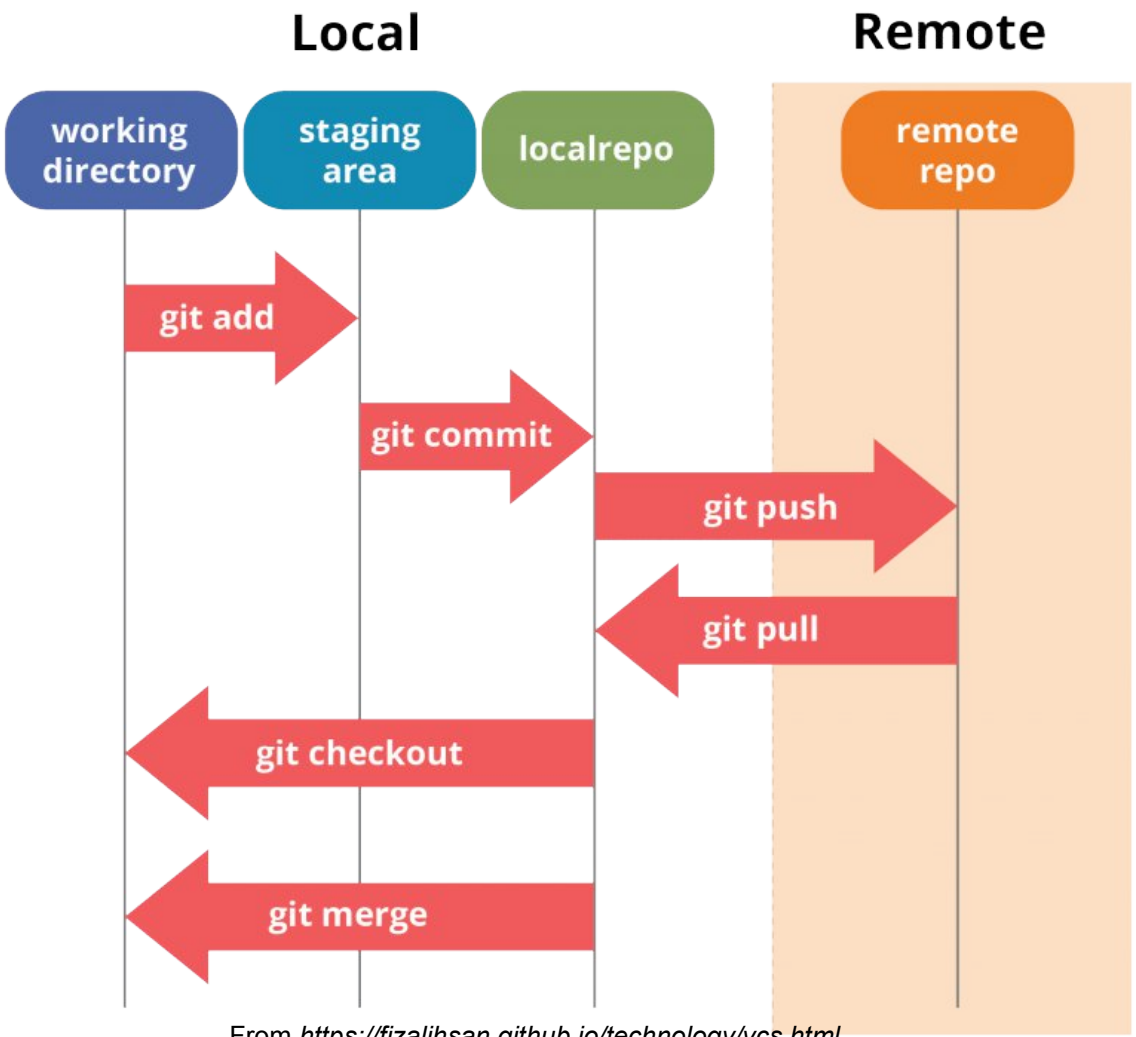
From *https://fizalihsan.github.io/technology/vcs.html*

Git

Basic operations:

```
$ git clone repo_adress/my_project.git
```

You have now a local version of the project



From *https://fizalihsan.github.io/technology/vcs.html*

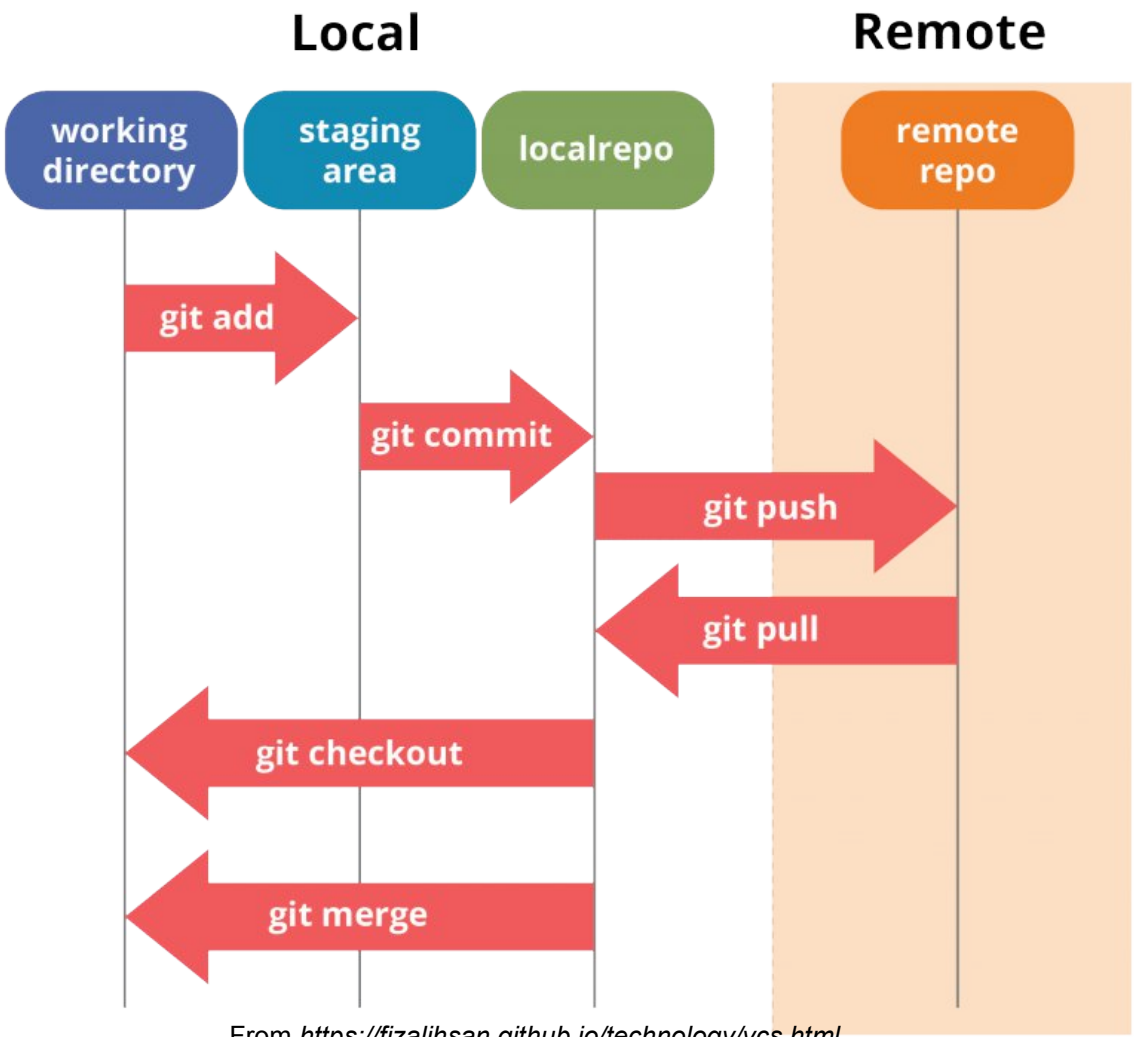
Git

Basic operations:

```
$ git clone repo_adress/my_project.git
```

You have now a local version of the project

(side note : be careful, on github now, *https* will require 2 Factor Authentification 2-FA, *ssh* with public/private key)



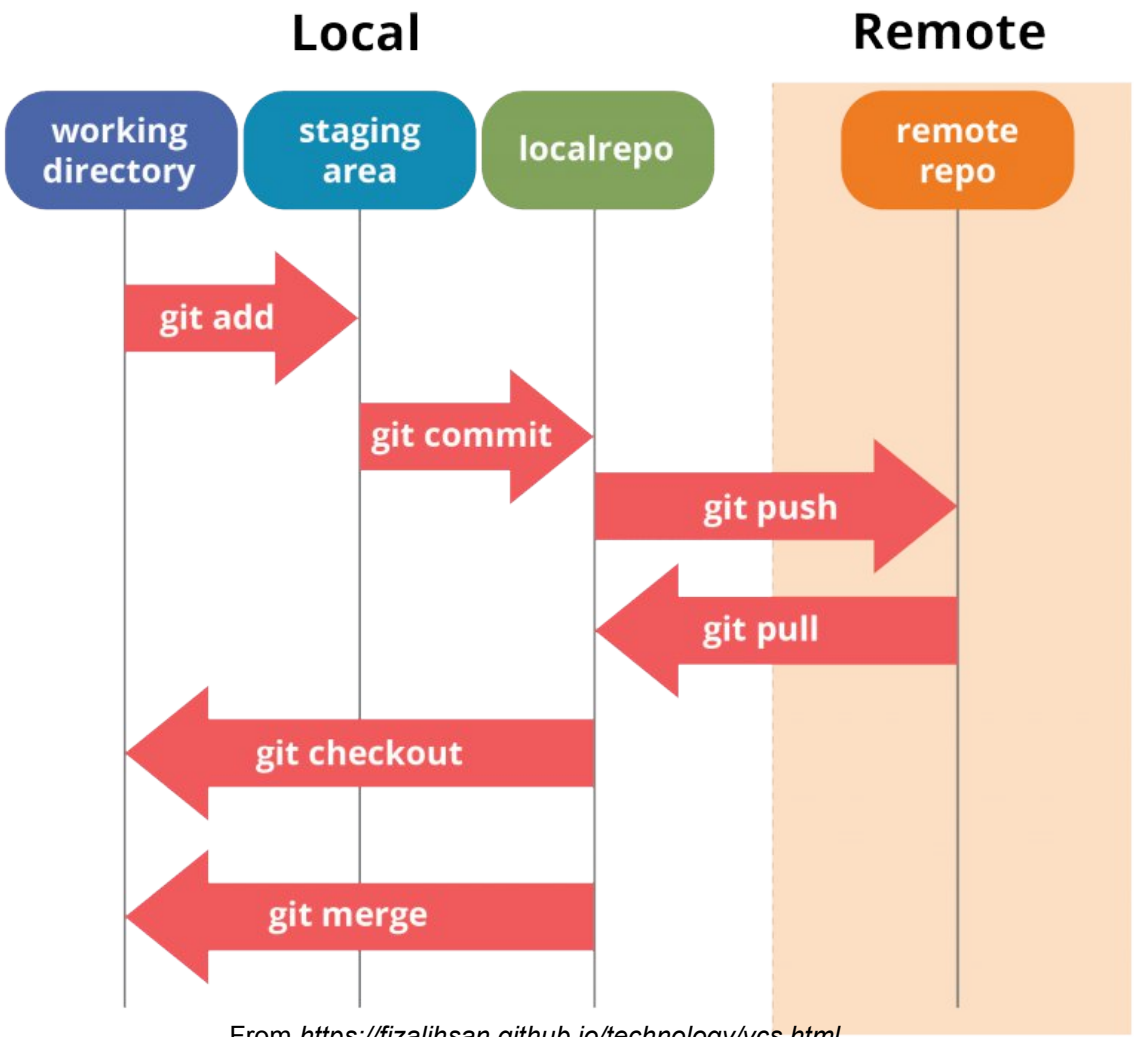
From *https://fizalihsan.github.io/technology/vcs.html*

Git

Basic operations:

```
$ git clone repo_adress/my_project.git
```

You have now a local version of the project



From *https://fizalihsan.github.io/technology/vcs.html*

Git

Basic operations:

`$ git clone` *`repo_adress/my_project.git`*

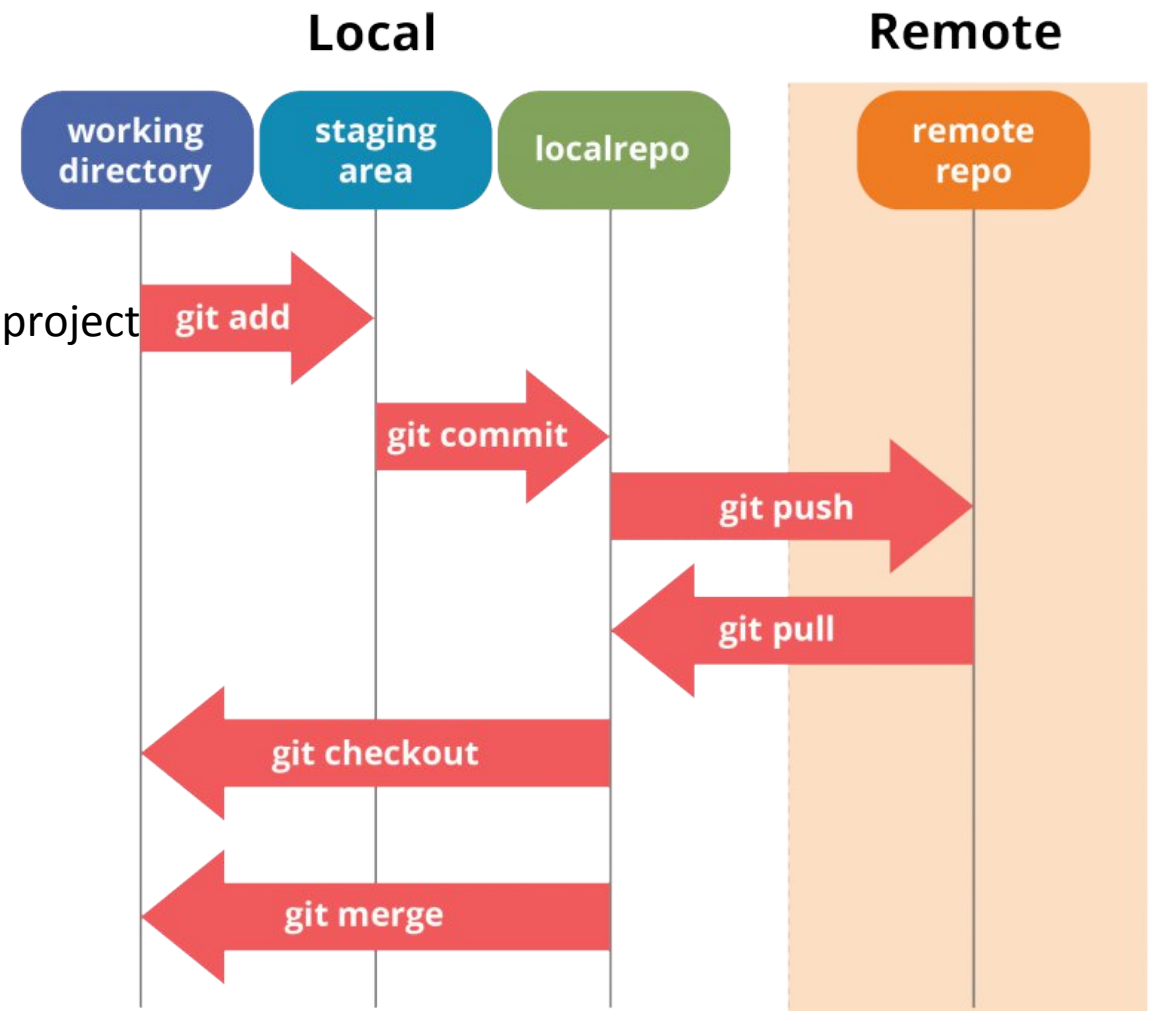
You have now a local version of *my_project*

→ do some modifications on the project ;

→ once you are happy with the modifications you made locally :

`$ git add new_file.py` (only when a new file is added to the project

not needed if an existing file is modified)



From *https://fizalihsan.github.io/technology/vcs.html*

Git

Basic operations:

```
$ git clone repo_adress/my_project.git
```

You have now a local version of *my_project*

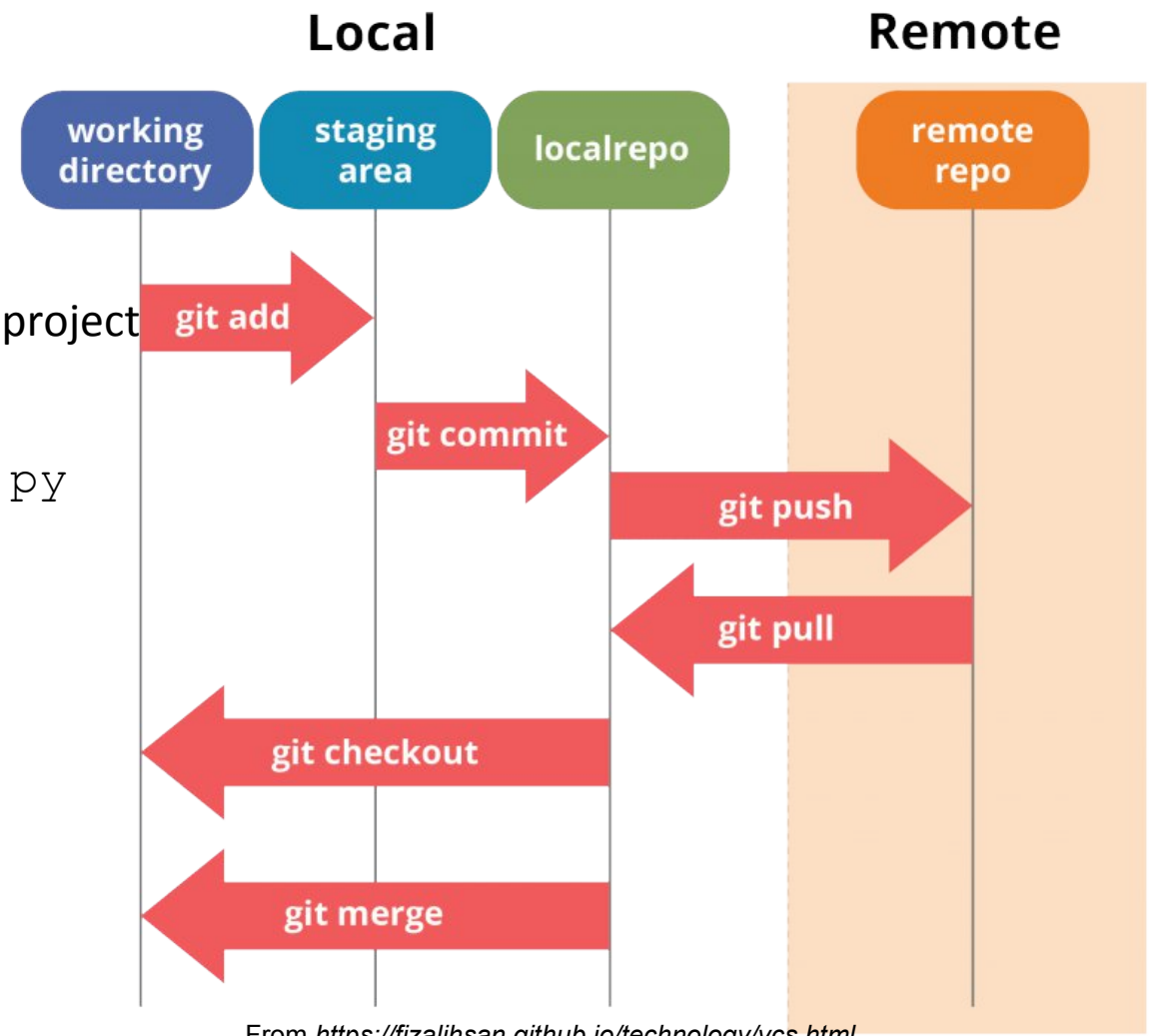
→ do some modifications on the project ;

→ once you are happy with the modifications you made locally :

`$ git add new_file.py` (only when a new file is added to the project not needed if an existing file is modified)

```
$ git commit -m 'my new modifications' my_file.py
```

(describe what you did to the project)



From *https://fizalihsan.github.io/technology/vcs.html*

Git

Basic operations:

```
$ git clone repo_adress/my_project.git
```

You have now a local version of *my_project*

→ do some modifications on the project ;

→ once you are happy with the modifications you made locally :

`$ git add new_file.py` (only when a new file is added to the project not needed if an existing file is modified)

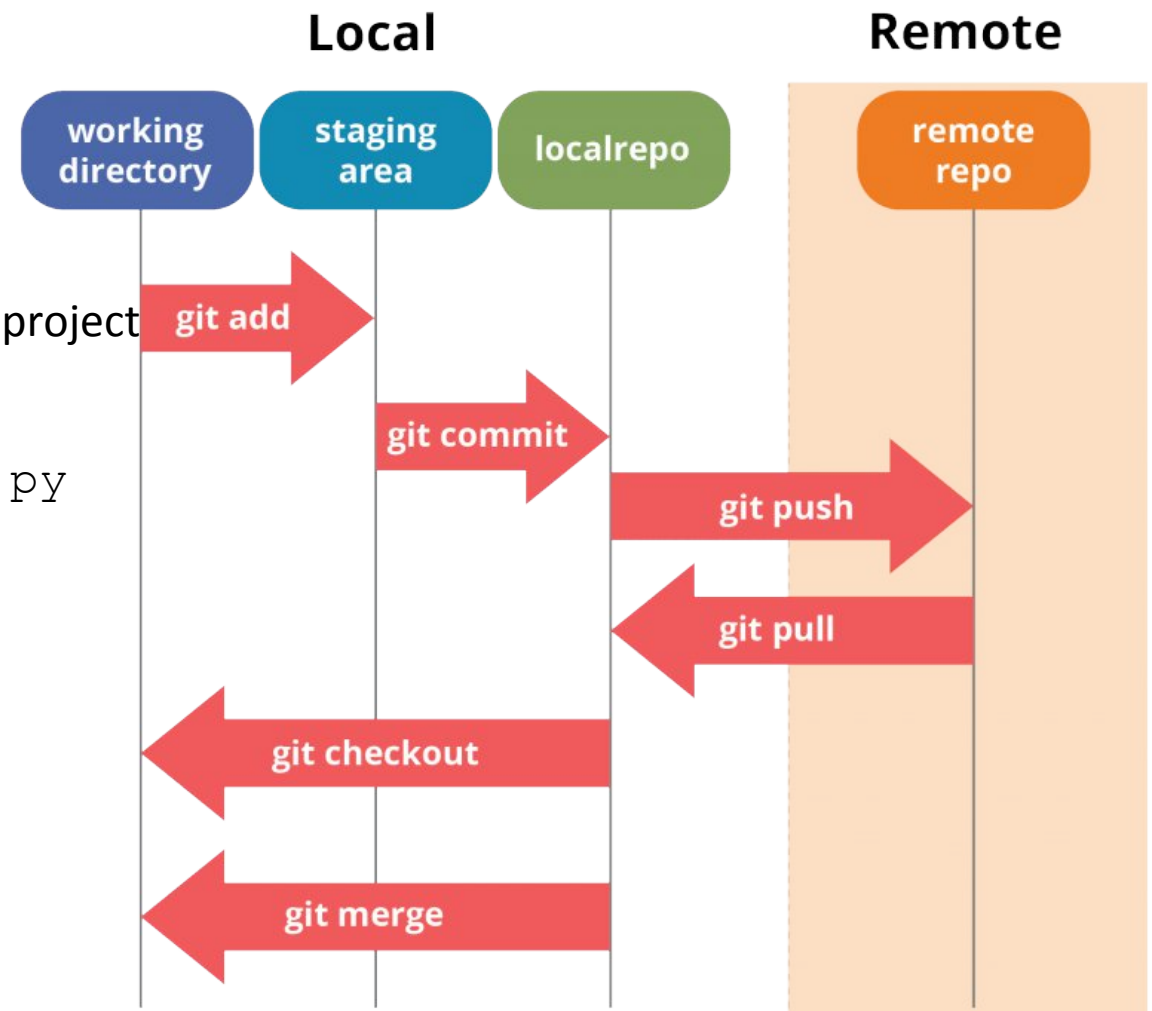
```
$ git commit -m 'my new modifications' my_file.py
```

(describe what you did to the project)

```
$ git push origin master
```

(push your version to become the new standard on the repo)

NB : default branch is named *master*



From *https://fizalihsan.github.io/technology/vcs.html*

Git

Basic operations:

```
$ git clone repo_adress/my_project.git
```

You have now a local version of *my_project*

→ do some modifications on the project ;

→ once you are happy with the modifications you made locally :

`$ git add new_file.py` (only when a new file is added to the project not needed if an existing file is modified)

```
$ git commit -m 'my new modifications' my_file.py
```

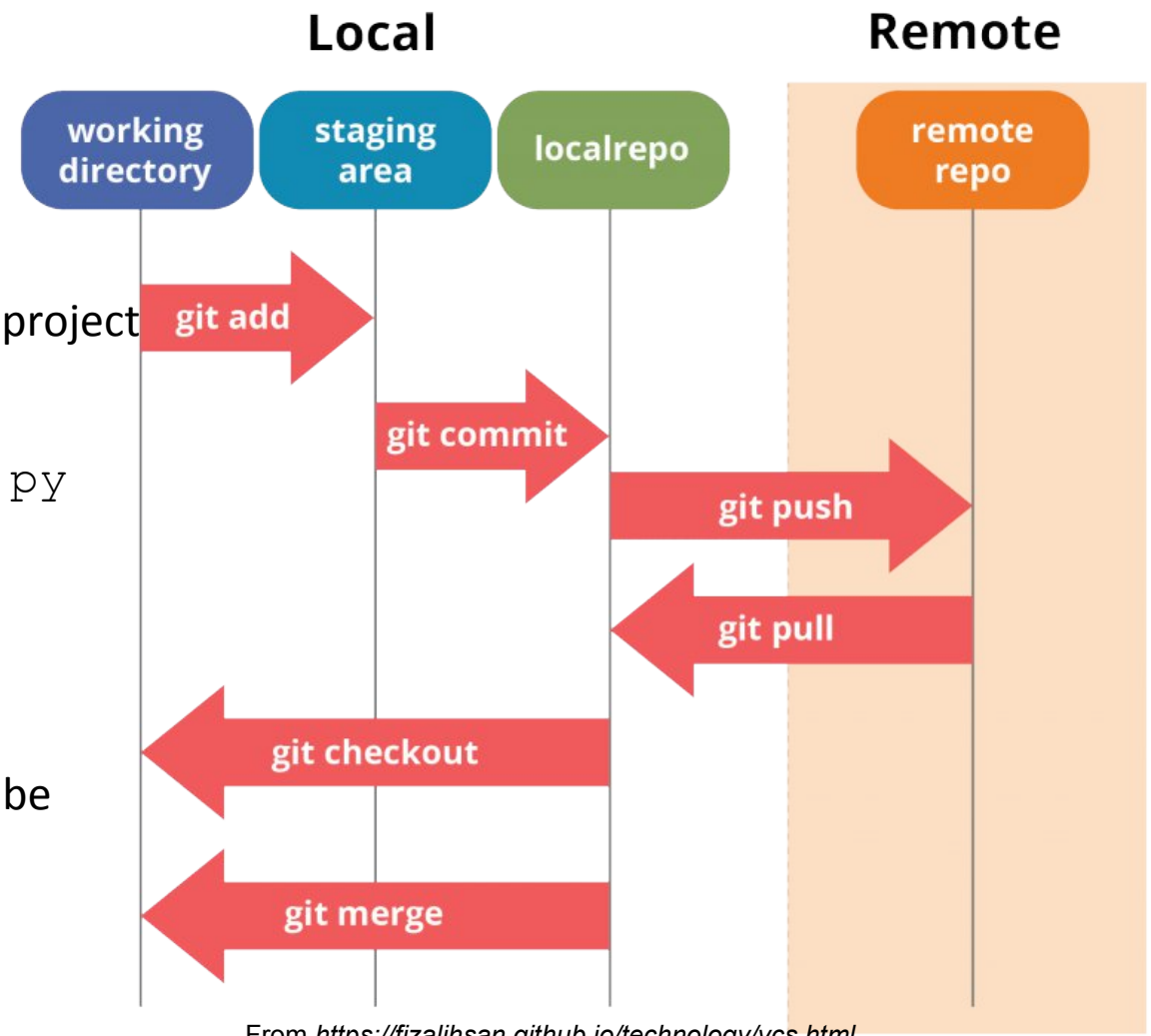
(describe what you did to the project)

```
$ git push origin master
```

(push your version to become the new standard on the repo)

Before pushing your version, the newest version on the repo should be retrieved by

$ git pull origin master



From *https://fizalihsan.github.io/technology/vcs.html*

Git

I made a mistake:

$ git status

Will show the « status » of the files in a local git directory

```
(base) 2 [15:24:46] Je n'ai pas de nom !@CRI-103-DE01: ~/miniconda3/Packages/neuropycon_demo/OpenfMRI_ds000117  $ git status
Sur la branche add_pybids
Fichiers non suivis:
  (utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)

        02-plot_preprocessing_pybids.py
        10_events_inverse_stc_old.py
        10_events_inverse_stc_old2.py
        __pycache__/
        params_testDM.json
        ../OpenfMRI_ds000117_pybids/log.txt
        ../OpenfMRI_ds000117_pybids/params_nocli.json
        ../OpenfMRI_ds000117_pybids/prep_trans.py
        ../OpenfMRI_ds000117_pybids/trns.txt

aucune modification ajoutée à la validation mais des fichiers non suivis sont présents (utilisez "git add" pour les suivre)
```

Git

I made a mistake:

I make some modifications on a « followed » file (i.e. a file that have been added to the git project)

```
(base) 3 [15:24:52] Je n'ai pas de nom !@CRI-103-DE01: ~/miniconda3/Packages/neuropycon_demo/OpenfMRI_ds000117  $ git status
Sur la branche add_pybids
Modifications qui ne seront pas validées :
  (utilisez "git add <fichier>..." pour mettre à jour ce qui sera validé)
  (utilisez "git checkout -- <fichier>..." pour annuler les modifications dans la copie de travail)

        modifié :         ../OpenfMRI_ds000117_pybids/02-03-plot_preprocessing_inverse_pybids.py

Fichiers non suivis:
  (utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)

        02-plot_preprocessing_pybids.py
        10_events_inverse_stc_old.py
        10_events_inverse_stc_old2.py
        __pycache__/
        params_testDM.json
        ../OpenfMRI_ds000117_pybids/log.txt
        ../OpenfMRI_ds000117_pybids/params_nocli.json
        ../OpenfMRI_ds000117_pybids/prep_trans.py
        ../OpenfMRI_ds000117_pybids/trns.txt

aucune modification n'a été ajoutée à la validation (utilisez "git add" ou "git commit -a")
```

Git

I made a mistake:

`$ git stash`

You erase all the modications that have NOT been commited so far (i.e. you go back to the latest pulled version)

```
(base) 4 [15:26:04] Je n'ai pas de nom !@CRI-103-DE01: ~/miniconda3/Packages/neuropycon_demo/OpenfMRI_ds000117  $ git stash
Copie de travail et état de l'index sauvegardés dans WIP on add_pybids: a7c6046 bug
HEAD est maintenant à a7c6046 bug
(base) 5 [15:27:04] Je n'ai pas de nom !@CRI-103-DE01: ~/miniconda3/Packages/neuropycon_demo/OpenfMRI_ds000117  $ git status
Sur la branche add_pybids
Fichiers non suivis:
  (utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)

        02-plot_preprocessing_pybids.py
        10_events_inverse_stc_old.py
        10_events_inverse_stc_old2.py
        __pycache__/
        params_testDM.json
        ../OpenfMRI_ds000117_pybids/log.txt
        ../OpenfMRI_ds000117_pybids/params_nocli.json
        ../OpenfMRI_ds000117_pybids/prep_trans.py
        ../OpenfMRI_ds000117_pybids/trns.txt

aucune modification ajoutée à la validation mais des fichiers non suivis sont présents (utilisez "git add" pour les suivre)
```

This will modify the version of the file on the computer IRREVERSIBLY !!!!!!

Git

I made a mistake:

```
$ git stash
```

You erase all the modications that have NOT been commited so far (i.e. you go back to the latest pulled version)

```
$ git restore my/file.py
```

It is also possible to « restore » one single file instead of reloading all the modified files

This will modify the version of the file on the computer IRREVERSIBLY !!!!!!

Git

Sidenote : Github desktop

Most of the commands that were presented are available in a GUI called « github desktop »

Much easier to handle for beginners

Only available if your code is stored on github....

N.B. github can be used with or without github desktop

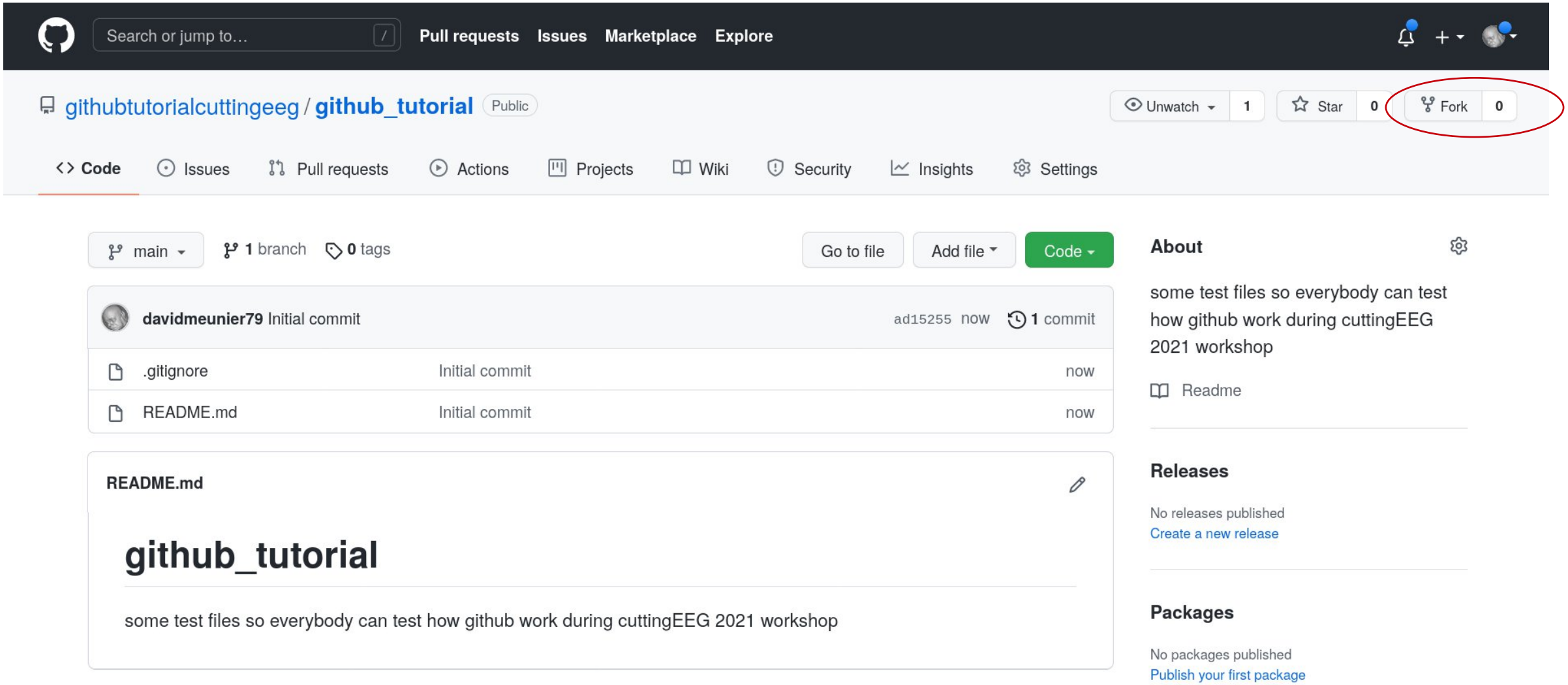
Git on Github

Git on Github

Advanced operations:

A better way of working :

- ***fork*** a version of the original project (remote = *upstream*) on your (github or else) account


Search or jump to…
Pull requests
Issues
Marketplace
Explore
githubtutorialcuttingeeg / github_tutorial
Public
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
davidmeunier79 Initial commit
ad15255 now
1 commit
.gitignore
Initial commit
now
README.md
Initial commit
now
README.md
github_tutorial
some test files so everybody can test how github work during cuttingEEG 2021 workshop
About
some test files so everybody can test how github work during cuttingEEG 2021 workshop
Readme
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package

davidmeunier79 / **github_tutorial** Public

forked from githubtutorialcuttingeeg/github_tutorial

Watch 0 | Star 0 | Fork 1

Code | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

main | 1 branch | 0 tags | Go to file | Add file | Code

This branch is even with githubtutorialcuttingeeg:main. | Contribute | Fetch upstream

davidmeunier79 Initial commit `ad15255` 1 minute ago | 1 commit

.gitignore	Initial commit	1 minute ago
README.md	Initial commit	1 minute ago

README.md

github_tutorial

some test files so everybody can test how github work during cuttingEEG 2021 workshop

About

some test files so everybody can test how github work during cuttingEEG 2021 workshop

Readme

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

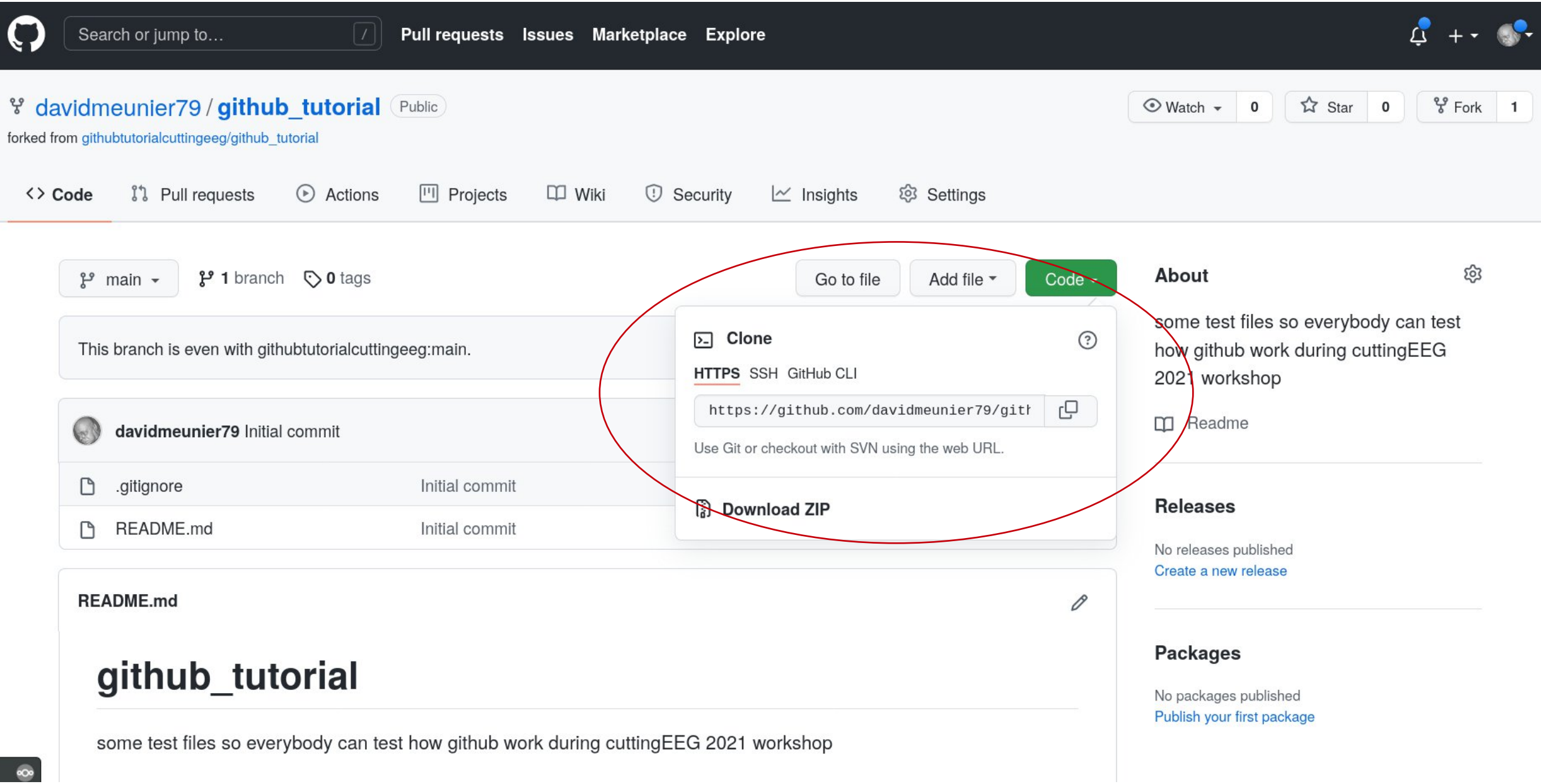
Git on Github

Advanced operations:

A better way of working :

- ***fork*** a version of the original project (remote = *upstream*) on your (github or else) account

- **clone** your local version from your forked remote (now named *origin*)


Search or jump to…
Pull requests
Issues
Marketplace
Explore
davidmeunier79 / github_tutorial
Public
forked from githubtutorialcuttingeeg/github_tutorial
Watch 0
Star 0
Fork 1
Code
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
This branch is even with githubtutorialcuttingeeg:main.
Clone
HTTPS SSH GitHub CLI
https://github.com/davidmeunier79/gith
Use Git or checkout with SVN using the web URL.
Download ZIP
davidmeunier79 Initial commit
.gitignore Initial commit
README.md Initial commit
About
some test files so everybody can test how github work during cuttingEEG 2021 workshop
Readme
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
README.md
github_tutorial
some test files so everybody can test how github work during cuttingEEG 2021 workshop

Git on Github

Advanced operations:

A better way of working :

- ***fork*** a version of the original project (remote = *upstream*) on your (github or else) account

- **clone** your local version from your forked remote (now named *origin*)

```
$ git clone my_repo/my_project
```

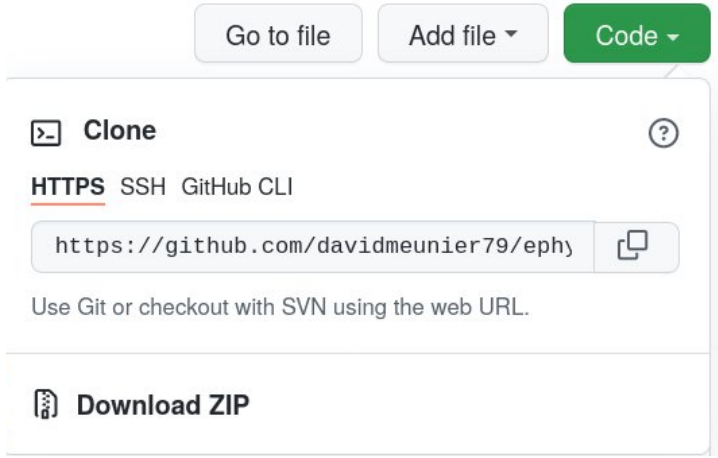
Git on Github

Advanced operations:

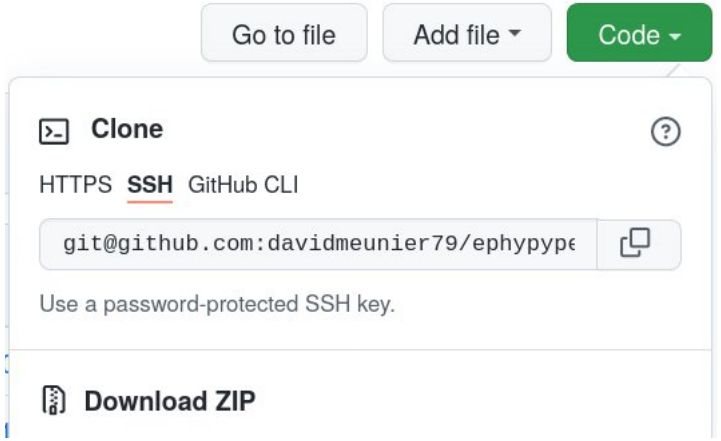
A better way of working :

- ***fork*** a version of the original project (remote = *upstream*) on your (github or else) account
- **clone** your local version from your forked remote (now named *origin*)

Side note : be careful, on github since a few months :



https requires 2 Factor Authentification 2-FA to push, but is ok if you clone only



ssh with public/private key is still OK

Git on Github

Advanced operations:

A better way of working :

- ***fork*** a version of the original project (remote = *upstream*) on your (github or else) account

- **clone** your local version from your forked remote (now named *origin*)

- create a new **branch** (*my_new_feature*)

```
$ git branch my_new_feature
$ git checkout my_new_feature
```

Git on Github

Advanced operations:

A better way of working :

- ***fork*** a version of the original project (remote = *upstream*) on your (github or else) account
- **clone** your local version from your forked remote (now named *origin*)
- create a new **branch** (*my_new_feature*)
- work on your local version, and make commit/push on *my_new_feature*

Git on Github

Advanced operations:

A better way of working :

- ***fork*** a version of the original project (remote = *upstream*) on your (github or else) account

- **clone** your local version from your forked remote (now named *origin*)

- create a new **branch** (*my_new_feature*)

- work on your local version, and make commit/push on *my_new_feature*

- propose a **Pull Request (PR)** when a « real » improvement have been added to the project.

This step involves both automatic and « human » evaluation of the quality of the code (formatting, naming, etc) and checks if the modification do not disrupt the project as a whole **(**called **continuous integration)**.

Git on Github

Advanced operations:

A better way of working :

- ***fork*** a version of the original project (remote = *upstream*) on your (github or else) account

- **clone** your local version from your forked remote (now named *origin*)

- create a new **branch** (*my_new_feature*)

- work on your local version, and make commit/push on *my_new_feature*

- propose a **Pull Request (PR)** when a « real » improvement have been added to the project.

This step involves both automatic and « human » evaluation of the quality of the code (formatting, naming, etc) and checks if the modification do not disrupt the project as a whole **(**called **continuous integration)**.

A good practice is to **retrieve** the last version of the original project before :

```
$ git remote add upstream https://github.com/neuropycon/ephypype.git
$ git pull upstream master
```

Pull Request on github

[MRG] Hotfix #47

Open **davidmeunier79** wants to merge 3 commits into `neuropycon:master` from `davidmeunier79:hotfix`

Conversation 1 | Commits 3 | Checks 0 | Files changed 15

davidmeunier79 commented 10 days ago — Member

No description provided.

`started example with dyn graph, ok till series of conf cormat` `99ef953`

davidmeunier79 force-pushed the `davidmeunier79:hotfix` branch from **`19f834a`** to **`bfbcd76`** 10 days ago

Pull Request on github

[MRG] Hotfix #47

Open **davidmeunier79** wants to merge 3 commits into `neuropycon:master` from `davidmeunier79:hotfix`

Conversation 1 | Commits 3 | Checks 0 | Files changed 15

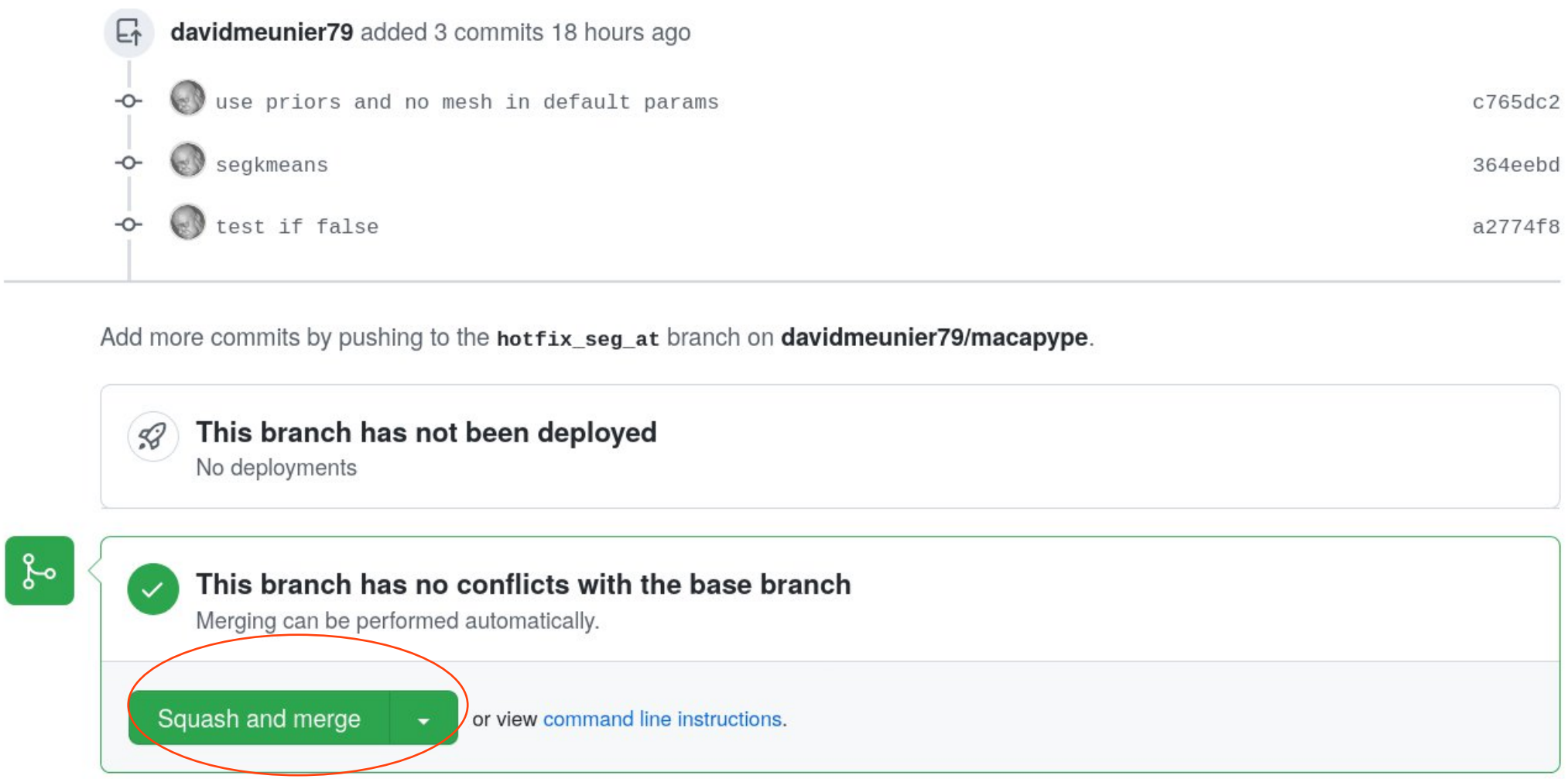
davidmeunier79 commented 10 days ago — Member

No description provided.

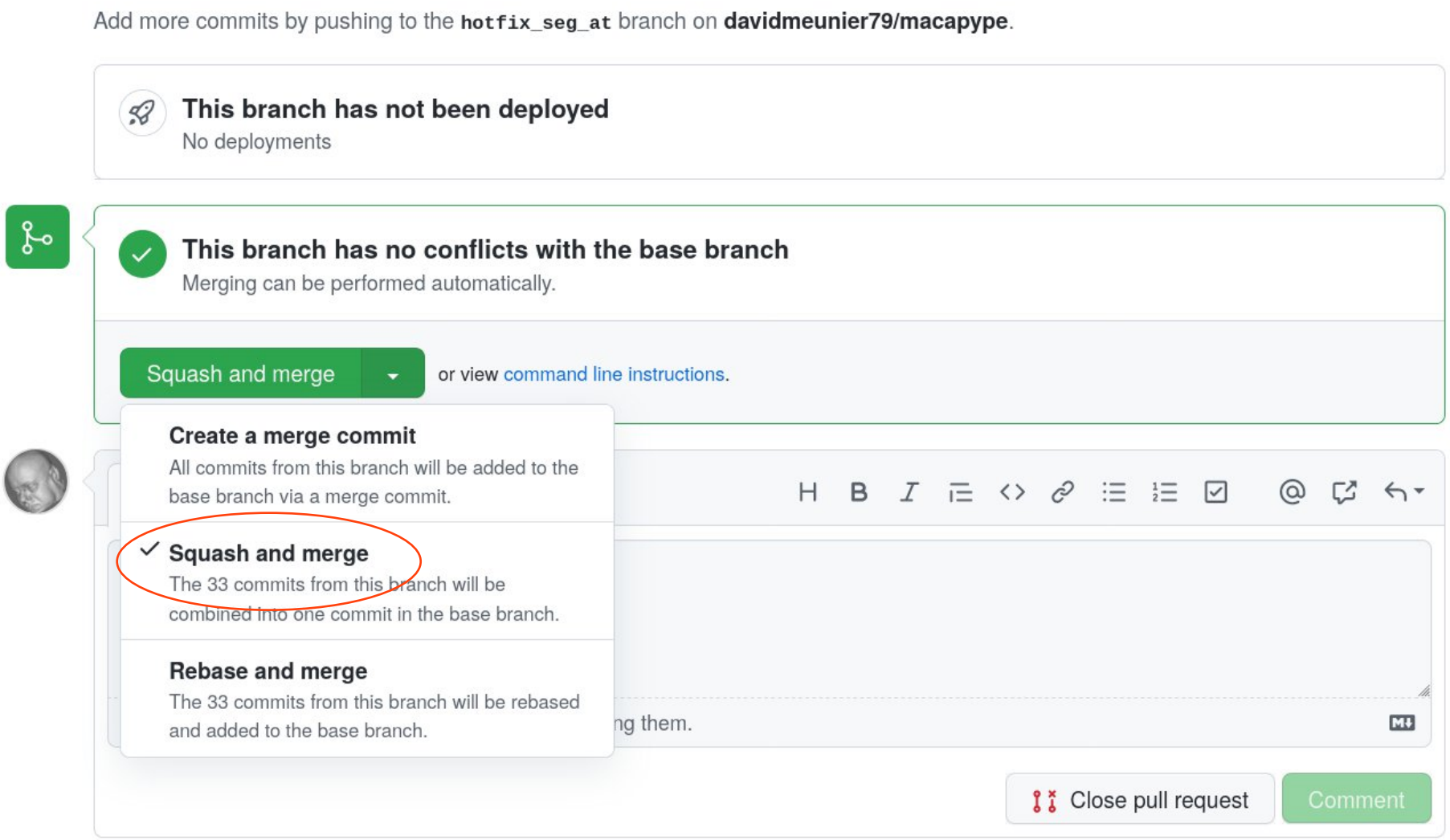
`started example with dyn graph, ok till series of conf cormat` `99ef953`

davidmeunier79 force-pushed the `davidmeunier79:hotfix` branch from **`19f834a`** to **`bfbcd76`** 10 days ago

Pull Request on github



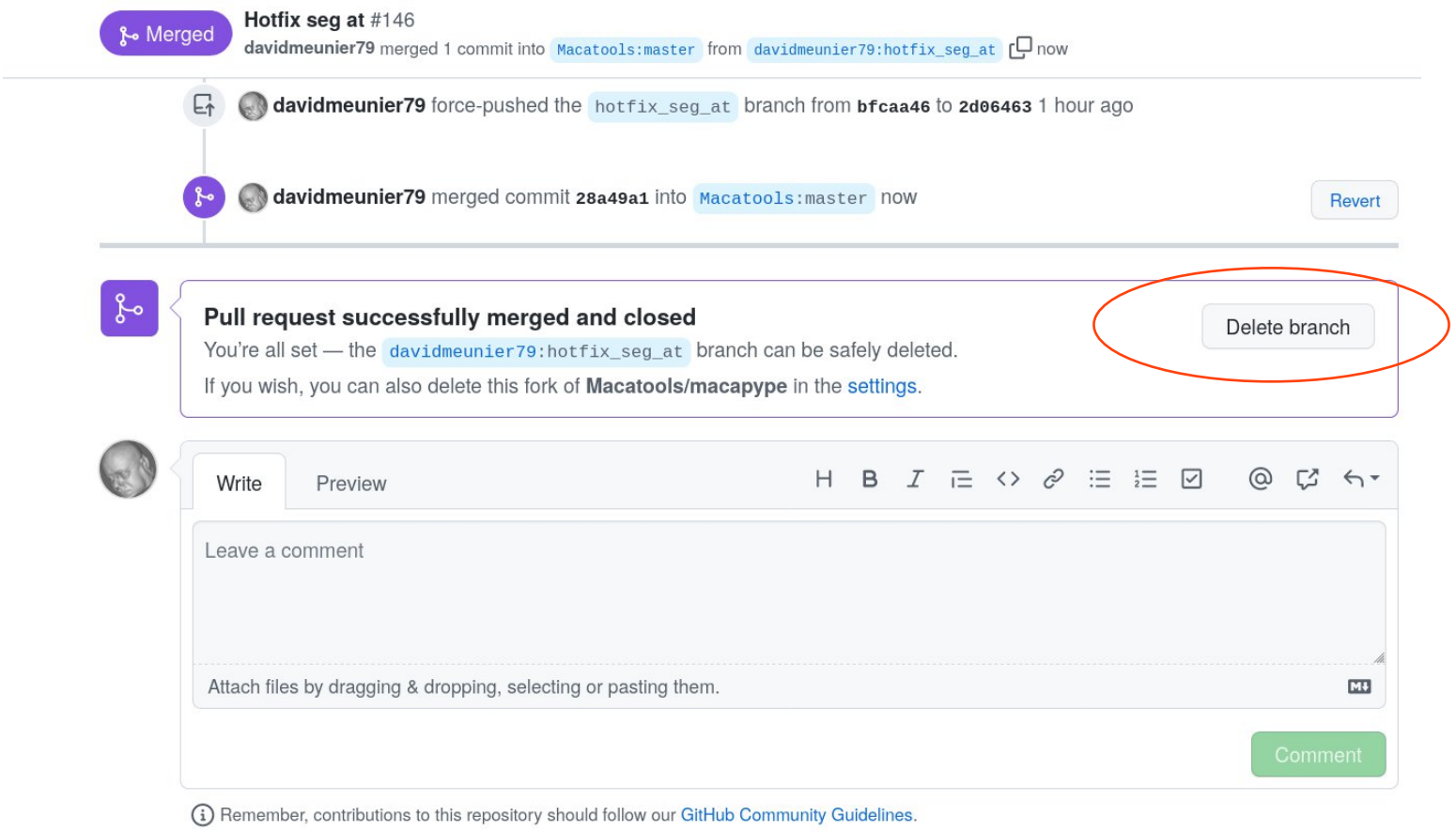
Pull Request on github



After a Pull Request

Once a PR is accepted, the good practice is to :

- delete the corresponding branch on your github fork (eg davidmeunier79:my_feature)



After a Pull Request

Once a PR is accepted, the good practice is to :

- delete the corresponding branch on your github fork (eg davidmeunier79:my_feature)

- on your computer, in your local version :

```
$ git checkout master # get back to the master/main branch
$ git pull upstream master  # get the master verion updated with the « new » clean version of the package
$ git branch -D my_feature # DELETE the branch you were working on
```

More advanced topics

More advanced topics

Configuring CI:

- What is required for a PR before event having a human contributor looking at it...
- Typically include the following steps :
- unit-tests should all pass
- formatting of code (e.g PEP8 for python)
- generating documentation from examples using package code

Typical CI providers are CircleCI and TravisCI .

Update: now everything can done with **github actions,** a service provided by github on their own servers

It requires a config file, where you basically build an OS from scratch, install all your package dependancies and your package, and make it run

More advanced topics

Configuring CI:

- Example file of a github action config file for a python packag

(.github/workflows/check_on_PR.yml in your package)

```yaml
name: Checking unit-tests, PEP8 and standard pipeline congruency

on:

  pull_request:
    branches:
      - master


jobs:
  build:

    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@master
      - name: Set up Python 3.7
        uses: actions/setup-python@v1
        with:
          python-version: 3.7

      - name: Install dependencies
        run: |
            pip install codecov pytest pytest-cov
            pip install -e .
            pip install flake8
            sudo apt-get install build-essential  graphviz libgraphviz-dev
            pip install --upgrade pygraphviz graphviz

      - name: Test with pytest
        run:
            py.test --cov macapype --ignore=examples/ --ignore=run_examples/

      - name: Flake8
        run:
            flake8 --count macapype
```

More advanced topics

Configuring CI:

- Example file of a github action config file for a python packag

(.github/workflows/check_on_PR.yml in your package)

- on : event triggering the launch on the action
  - can be : pull_request, release, etc...
  - possibly on specific branches (e.g master, dev, etc.)

- runs-on : the OS (windows, ubuntu, etc.) and the python vers

```yaml
name: Checking unit-tests, PEP8 and standard pipeline congruency

on:

  pull_request:
    branches:
      - master


jobs:
  build:

    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@master
      - name: Set up Python 3.7
        uses: actions/setup-python@v1
        with:
          python-version: 3.7

      - name: Install dependencies
        run: |
            pip install codecov pytest pytest-cov
            pip install -e .
            pip install flake8
            sudo apt-get install build-essential  graphviz libgraphviz-dev
            pip install --upgrade pygraphviz graphviz

      - name: Test with pytest
        run:
            py.test --cov macapype --ignore=examples/ --ignore=run_examples/

      - name: Flake8
        run:
            flake8 --count macapype
```

More advanced topics

Configuring CI:

- Example file of a github action config file for a python packag

(.github/workflows/check_on_PR.yml in your package)

- on : event triggering the launch on the action
  - can be : pull_request, release, etc...
  - possibly on specific branches (e.g master, dev, etc.)
- runs-on : the OS (windows, ubuntu, etc.) and the python vers
- install all the depencies of your package

```yaml
name: Checking unit-tests, PEP8 and standard pipeline congruency

on:

  pull_request:
    branches:
      - master


jobs:
  build:

    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@master
      - name: Set up Python 3.7
        uses: actions/setup-python@v1
        with:
          python-version: 3.7

      - name: Install dependencies
        run: |
            pip install codecov pytest pytest-cov
            pip install -e .
            pip install flake8
            sudo apt-get install build-essential  graphviz libgraphviz-dev
            pip install --upgrade pygraphviz graphviz

      - name: Test with pytest
        run:
            py.test --cov macapype --ignore=examples/ --ignore=run_examples/

      - name: Flake8
        run:
            flake8 --count macapype
```

More advanced topics

Configuring CI:

- Example file of a github action config file for a python packag

(.github/workflows/check_on_PR.yml in your package)

```yaml
name: Checking unit-tests, PEP8 and standard pipeline congruency

on:

  pull_request:
    branches:
      - master


jobs:
  build:

    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@master
      - name: Set up Python 3.7
        uses: actions/setup-python@v1
        with:
          python-version: 3.7

      - name: Install dependencies
        run: |
            pip install codecov pytest pytest-cov
            pip install -e .
            pip install flake8
            sudo apt-get install build-essential  graphviz libgraphviz-dev
            pip install --upgrade pygraphviz graphviz

      - name: Test with pytest
        run:
            py.test --cov macapype --ignore=examples/ --ignore=run_examples/

      - name: Flake8
        run:
            flake8 --count macapype
```

- on : event triggering the launch on the action
  - can be : pull_request, release, etc...
  - possibly on specific branches (e.g master, dev, etc.)
- runs-on : the OS (windows, ubuntu, etc.) and the python vers
- install all the depencies of your package
- run unit-tests (pytest) and formatting (flake8)

More advanced topics

Configuring CI:

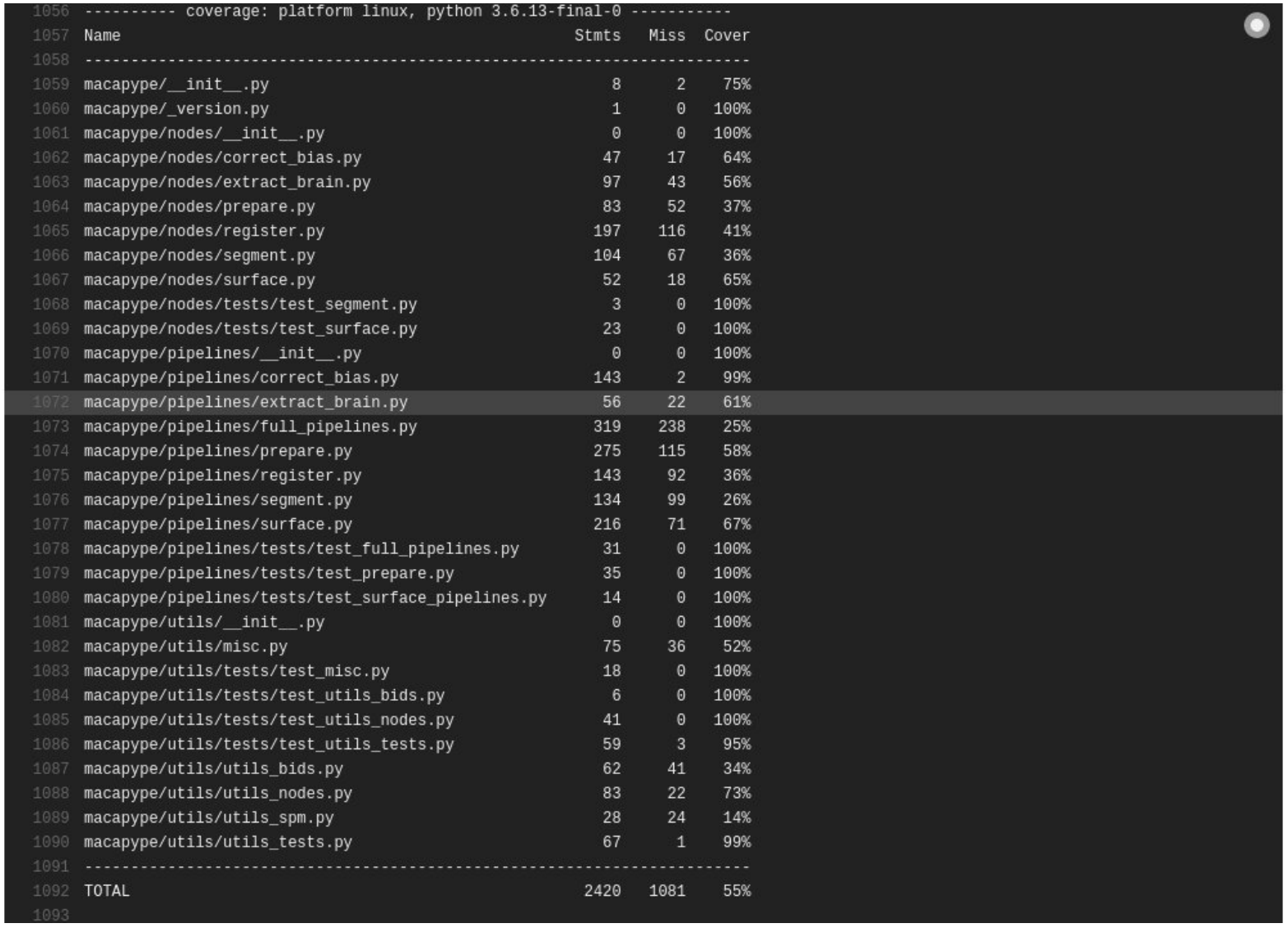
- **Running unittest,**
- testing if coverage is always increasing
- checking if code follows PEP8 standards

```
1007 ============================= test session starts ==============================
1008 platform linux -- Python 3.6.13, pytest-6.2.4, py-1.10.0, pluggy-0.13.1
1009 rootdir: /home/travis/build/Macatools/macapype
1010 plugins: cov-2.11.1
1011 collected 27 items
1012
1013 macapype/nodes/tests/test_segment.py .                                  [  3%]
1014 macapype/nodes/tests/test_surface.py .                                  [  7%]
1015 macapype/pipelines/tests/test_full_pipelines.py ...                     [ 18%]
1016 macapype/pipelines/tests/test_prepare.py .....                         [ 37%]
1017 macapype/pipelines/tests/test_surface_pipelines.py .                   [ 40%]
1018 macapype/utils/tests/test_misc.py ...                                  [ 51%]
1019 macapype/utils/tests/test_utils_bids.py .                              [ 55%]
1020 macapype/utils/tests/test_utils_nodes.py ....                          [ 70%]
1021 macapype/utils/tests/test_utils_tests.py ........                      [100%]
1022
1023 ============================== warnings summary ===============================
1024 macapype/nodes/tests/test_surface.py::test_meshify
1025 macapype/nodes/tests/test_surface.py::test_meshify
                                                                          Top ▲
```

More advanced topics

Configuring CI:

- Running unittest,
- **testing if coverage is always increasing**
- checking if code follows PEP8 standards



```
---------- coverage: platform linux, python 3.6.13-final-0 -----------
Name                                                    Stmts   Miss  Cover
---------------------------------------------------------------------------
macapype/__init__.py                                        8      2    75%
macapype/_version.py                                        1      0   100%
macapype/nodes/__init__.py                                  0      0   100%
macapype/nodes/correct_bias.py                             47     17    64%
macapype/nodes/extract_brain.py                            97     43    56%
macapype/nodes/prepare.py                                  83     52    37%
macapype/nodes/register.py                                197    116    41%
macapype/nodes/segment.py                                 104     67    36%
macapype/nodes/surface.py                                  52     18    65%
macapype/nodes/tests/test_segment.py                        3      0   100%
macapype/nodes/tests/test_surface.py                       23      0   100%
macapype/pipelines/__init__.py                              0      0   100%
macapype/pipelines/correct_bias.py                        143      2    99%
macapype/pipelines/extract_brain.py                        56     22    61%
macapype/pipelines/full_pipelines.py                      319    238    25%
macapype/pipelines/prepare.py                             275    115    58%
macapype/pipelines/register.py                            143     92    36%
macapype/pipelines/segment.py                             134     99    26%
macapype/pipelines/surface.py                             216     71    67%
macapype/pipelines/tests/test_full_pipelines.py            31      0   100%
macapype/pipelines/tests/test_prepare.py                   35      0   100%
macapype/pipelines/tests/test_surface_pipelines.py         14      0   100%
macapype/utils/__init__.py                                  0      0   100%
macapype/utils/misc.py                                     75     36    52%
macapype/utils/tests/test_misc.py                          18      0   100%
macapype/utils/tests/test_utils_bids.py                     6      0   100%
macapype/utils/tests/test_utils_nodes.py                   41      0   100%
macapype/utils/tests/test_utils_tests.py                   59      3    95%
macapype/utils/utils_bids.py                               62     41    34%
macapype/utils/utils_nodes.py                              83     22    73%
macapype/utils/utils_spm.py                                28     24    14%
macapype/utils/utils_tests.py                              67      1    99%
---------------------------------------------------------------------------
TOTAL                                                    2420   1081    55%
```

More advanced topics

Configuring CI:

- Running unittest,

- testing if coverage is always increasing

- checking if code follows PEP8 standards

```
1093
1094 ================== 27 passed, 6 warnings in 146.98s (0:02:26) ==================
1095 The command "if [[ "${TEST}" == "standard" ]]; then py.test --cov macapype --ignore=examples/ --ignore=run_examples/; fi;"
     exited with 0.
1096 $ flake8 --count macapype                                                    1.59s
1097 0
1098 The command "flake8 --count macapype" exited with 0.
▶1099 store build cache                                                  cache.2
1114
▶1115 $ if [ "${TEST}" == "standard" ]; then codecov; fi                 after_success  1.41s
1145
1146 Done. Your build exited with 0.
                                                                              Top ▲
```

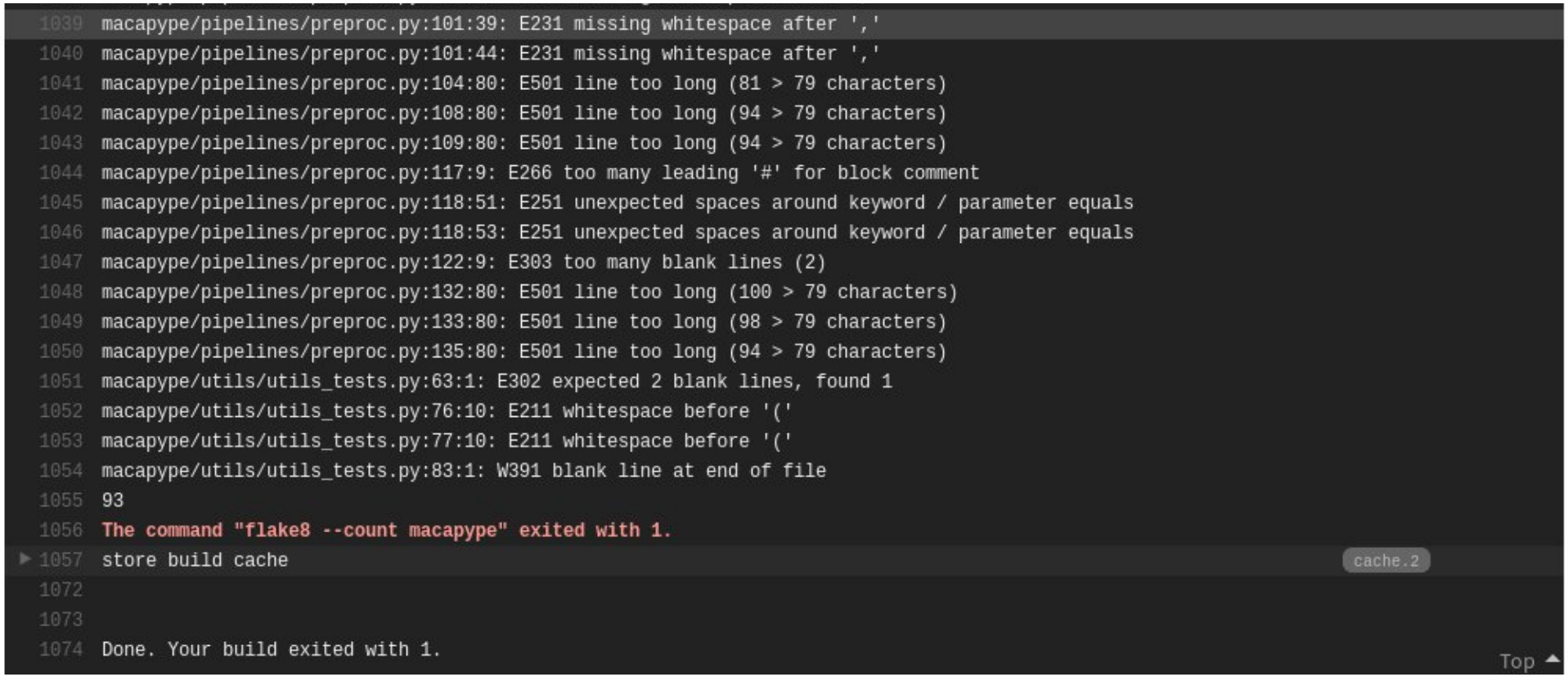
More advanced topics

Configuring CI:

- Running unittest,

- testing if coverage is always increasing

- checking if code follows PEP8 standards (example when it fails)



More advanced topics

Configuring CI:

- Example file of a github action config file for a python package (.github/workflows/check_on_PR.yml in your package)

- it is also possible to run low level full testing on dataset (for PR), but also full analysis for release

```
- name: Download dataset
  run: |
      wget --no-check-certificate --content-disposition  "https://amubox.univ-amu.fr/public.php?
      service=files&t=KJ2L5j6L6orPXxM&download" -O macapype_CI.zip
      unzip -o macapype_CI.zip -d macapype_CI

- name: Running all test pipelines
  run: |
      python workflows/segment_pnh.py -data /home/runner/work/macapype/macapype/macapype_CI/macaque_prime-de -
      out /home/runner/work/macapype/macapype/macapype_CI/macaque_prime-de/results -soft ANTS_test -species
      macaque -sub 032140 -ses 001 -deriv
```

More advanced topics

Configuring CI:

- Example file of a github action config file for a python package (.github/workflows/check_on_PR.yml in your package)

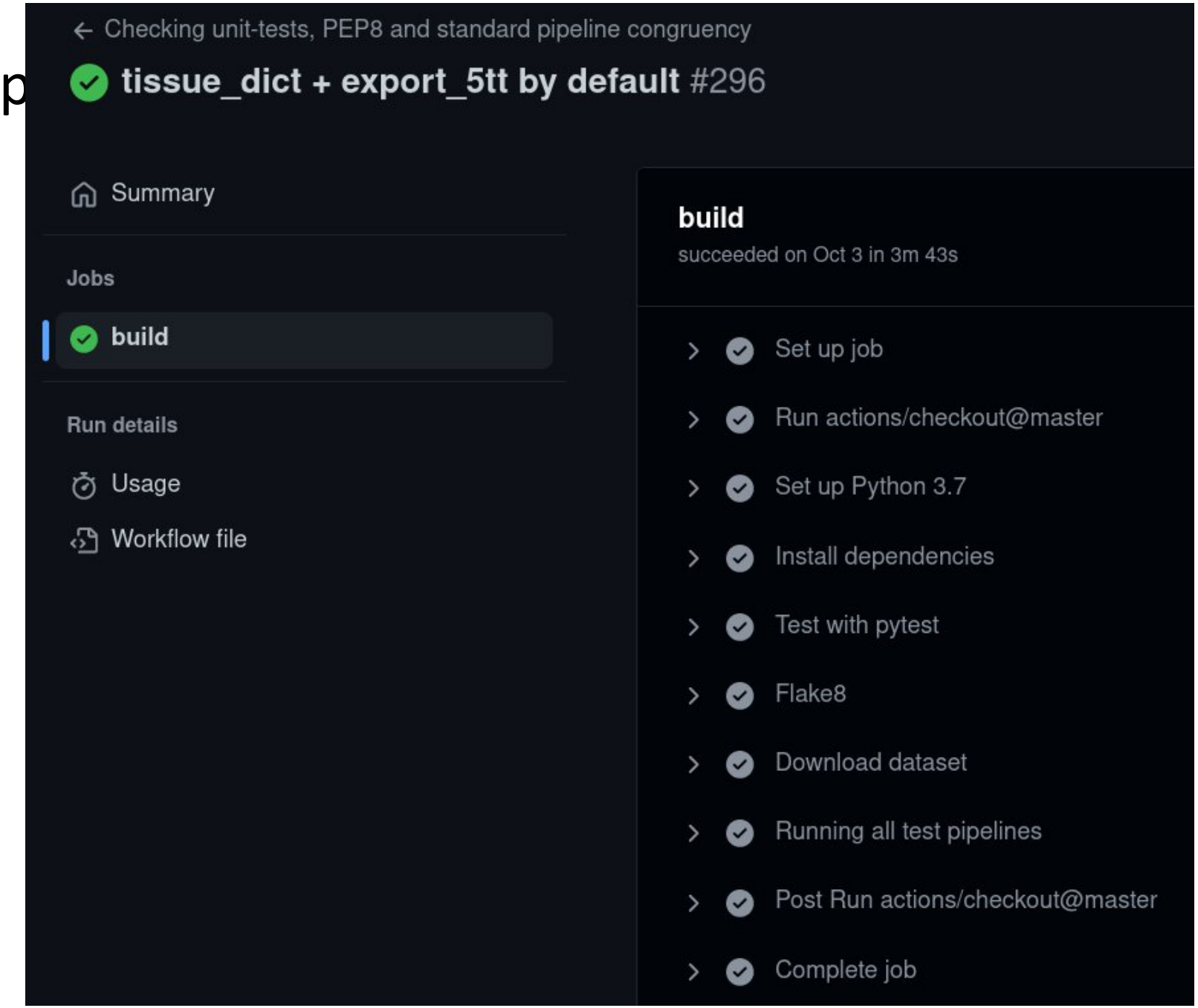
- it is also possible to run low level full testing on dataset (for PR), but also full analysis for release

```
- name: Download dataset
  run: |
      wget --no-check-certificate --content-disposition  "https://amubox.univ-amu.fr/public.php?
      service=files&t=KJ2L5j6L6orPXxM&download" -O macapype_CI.zip
      unzip -o macapype_CI.zip -d macapype_CI

- name: Running all test pipelines
  run: |
      python workflows/segment_pnh.py -data /home/runner/work/macapype/macapype/macapype_CI/macaque_prime-de -
      out /home/runner/work/macapype/macapype/macapype_CI/macaque_prime-de/results -soft ANTS_test -species
      macaque -sub 032140 -ses 001 -deriv
```

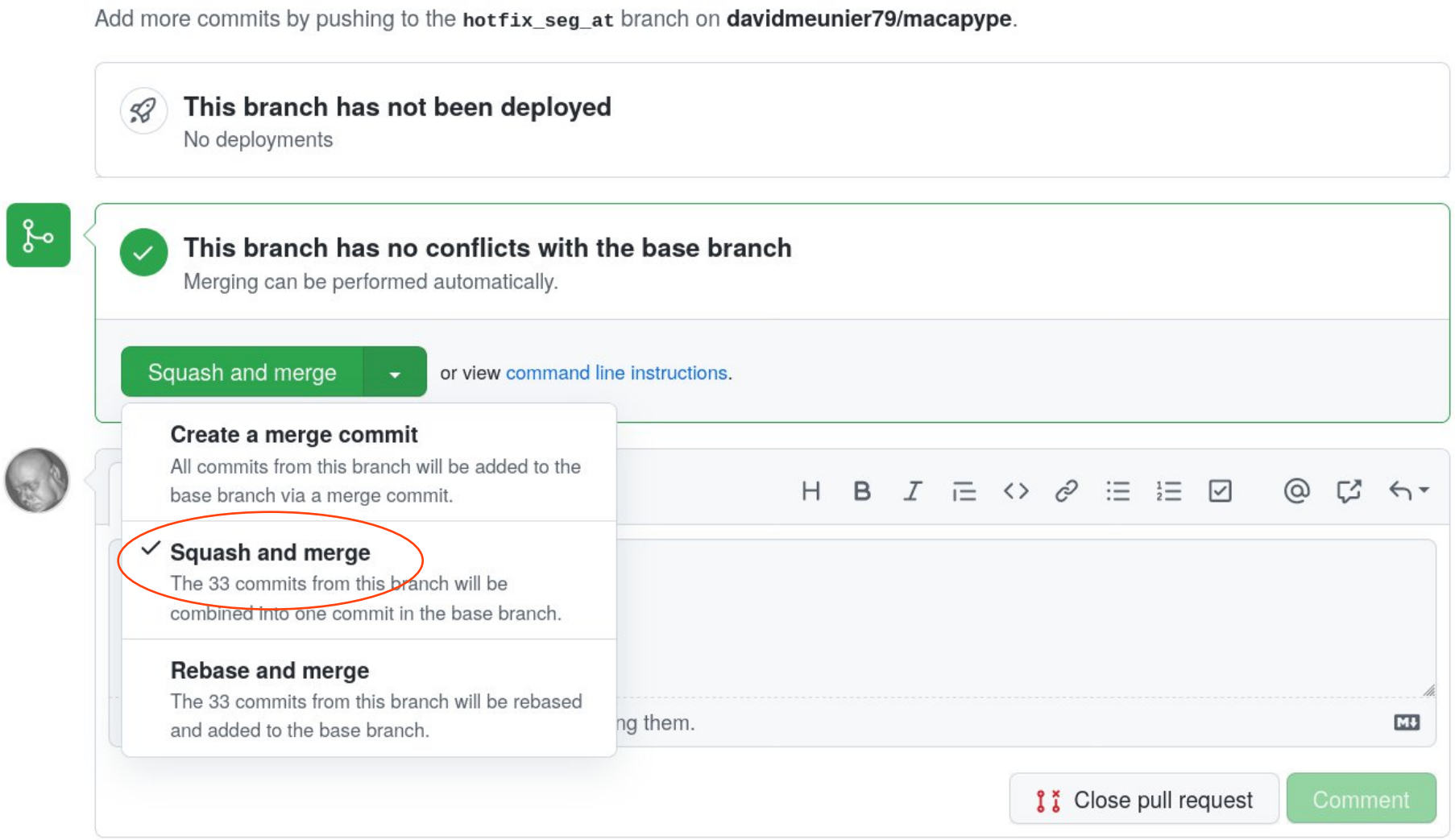
More advanced top

Accepting PR:



More advanced topics

Accepting PR:



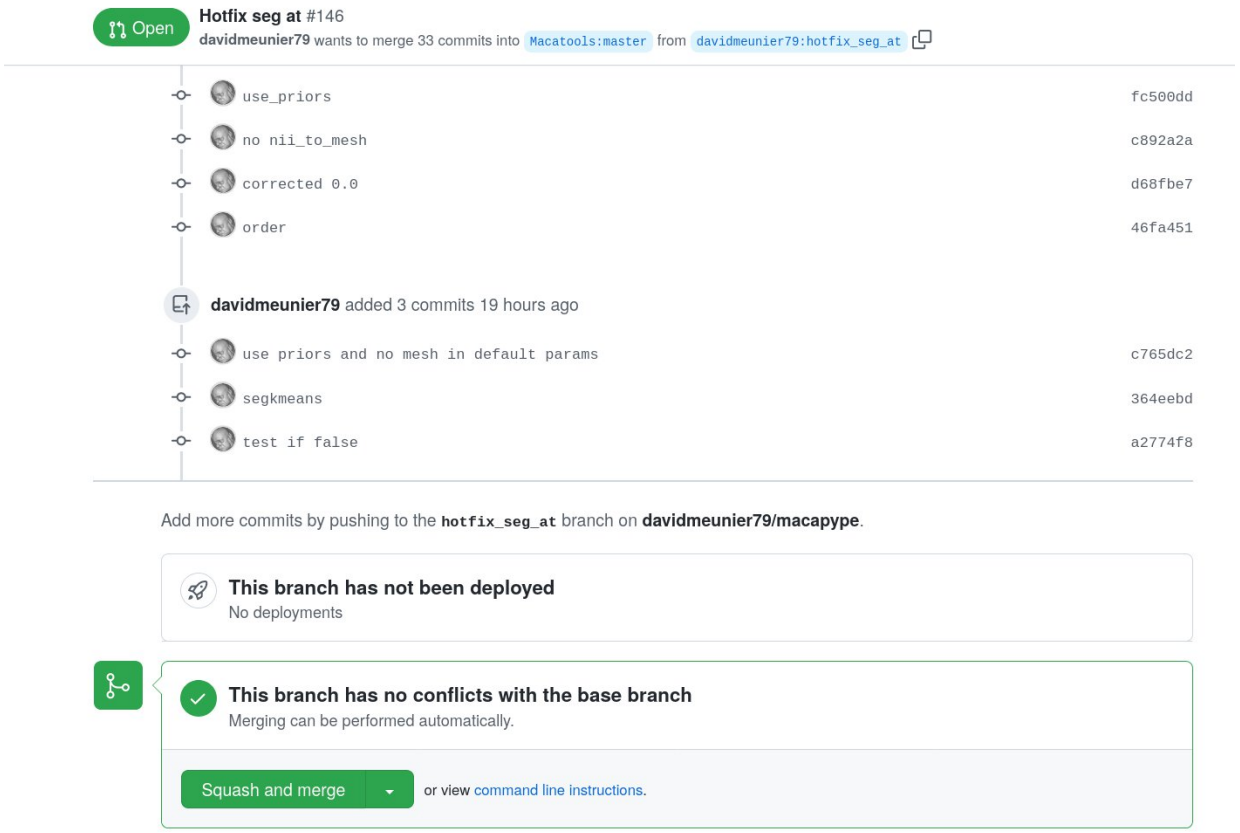
More advanced topics

Rebase:

Git rebase command have two different purposes :

- Remove all the little unrelvant commits you do (back and forth) when coding everyday between your local version and your fork, that do not need to appear in the commit history of the « clean project)

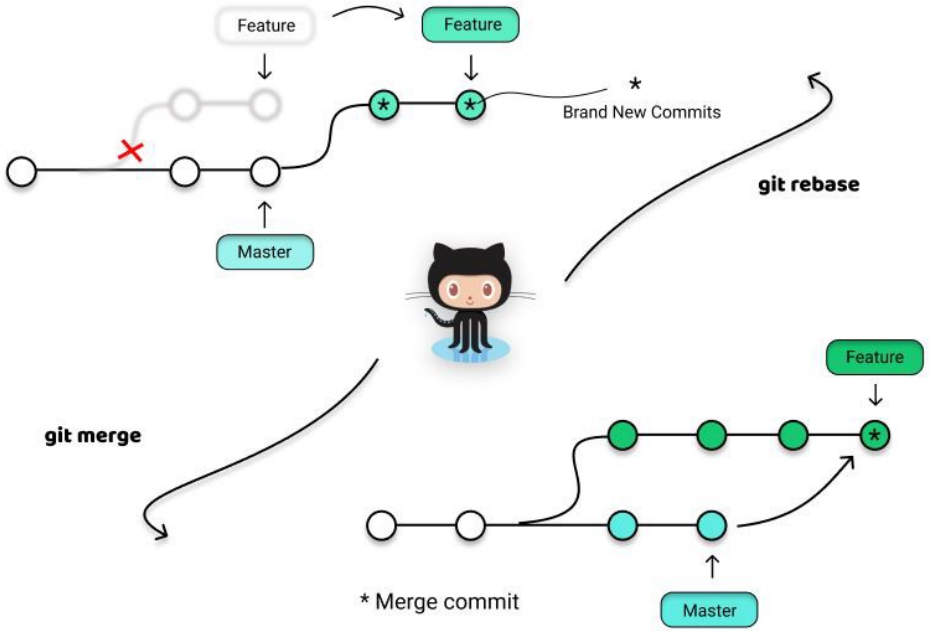
→ this is called **« squashing »**in git terms



More advanced topics

Rebase:

- « git merge » will merge the two branches in there current states
- « git rebase » will move all modifications from a previous bifurcaction to the present state



From *www.medium.com*

More advanced topics

Rebase:

- « git merge » will merge the two branches in there current states

- « git rebase » will move all modifications from a previous bifurcaction to the present state

```
$ git fetch upstream
$ git rebase -i upstream/master
```
(-i for interactive)

More advanced topics

```
pick 58721c1 suppressing all old files (only the new ones are present now)
pick 2ab0122 print register
pick c505aec align_masks
pick 0c84ccf self.new_value
pick a136825 print
pick a89b9bf print debut
pick 9a15ba8 essai list
pick 25b139f test priors
pick 916ee4a modified main
pick ac4df2d bug
pick 2b71b26 error argstr
pick ad7f737 removed template file
pick 0beaf4b removed checked on priors
pick 70c4ce2 Now a list or string
pick fb23012 bug
pick 5b46efe traits
pick afa7c2e added -w in atropos and in macaque/ants params + removed mesh by default
pick bff44c3 priors1.0
pick 715e90d testing prior_weight = -1.0
pick f108299 no return
pick a2f29bb testing use priors
pick 7088b06 outputnode
pick 4cf413e no default weight
pick c9b5c79 adding use_priors to macaque_ants
pick 7944ece using use_priors for passing prior_weight
pick fc500dd use_priors
pick c892a2a no nii_to_mesh
pick d68fbe7 corrected 0.0
pick 46fa451 order
pick c765dc2 use priors and no mesh in default params
pick 364eebd segkmeans
pick a2774f8 test if false

# Rebasage de 7776796..a2774f8 sur 7776796 (32 commandes)
#
# Commandes :
# p, pick = picorer le commit
# r, reword = picorer le commit, mais reformuler son message
# e, edit = picorer le commit, mais s'arrêter pour le modifier
# s, squash = prendre le commit, mais le fusionner avec le précédent
# f, fixup = comme "squash", mais en éliminant son message
# x, exec = lancer la commande (reste de la ligne) dans un shell
# d, drop = supprimer le commit
#
# Vous pouvez réordonner ces lignes ; elles sont exécutées de haut en bas.
#
# Si vous éliminez une ligne ici, LE COMMIT CORRESPONDANT SERA PERDU.
#
# Cependant, si vous effacez tout, le rebasage sera annulé.
#
# Veuillez noter que les commits vides sont en commentaire
```

More advanced topics

```
pick 58721c1 suppressing all old files (only the new ones are present now)
pick 2ab0122 print register
pick c505aec align_masks
pick 0c84ccf self.new_value
pick a136825 print
pick a89b9bf print debut
pick 9a15ba8 essai list
pick 25b139f test priors
pick 916ee4a modified main
pick ac4df2d bug
pick 2b71b26 error argstr
pick ad7f737 removed template file
pick 0beaf4b removed checked on priors
pick 70c4ce2 Now a list or string
pick fb23012 bug
pick 5b46efe traits
pick afa7c2e added -w in atropos and in macaque/ants params + removed mesh by default
pick bff44c3 priors1.0
pick 715e90d testing prior_weight = -1.0
pick f108299 no return
pick a2f29bb testing use priors
pick 7088b06 outputnode
pick 4cf413e no default weight
pick c9b5c79 adding use_priors to macaque_ants
pick 7944ece using use_priors for passing prior_weight
pick fc500dd use_priors
pick c892a2a no nii_to_mesh
pick d68fbe7 corrected 0.0
pick 46fa451 order
pick c765dc2 use priors and no mesh in default params
pick 364eebd segkmeans
pick a2774f8 test if false

# Rebasage de 7776796..a2774f8 sur 7776796 (32 commandes)
#
# Commandes :
# p, pick = picorer le commit
# r, reword = picorer le commit, mais reformuler son message
# e, edit = picorer le commit, mais s'arrêter pour le modifier
# s, squash = prendre le commit, mais le fusionner avec le précédent
# f, fixup = comme "squash", mais en éliminant son message
# x, exec = lancer la commande (reste de la ligne) dans un shell
# d, drop = supprimer le commit
#
# Vous pouvez réordonner ces lignes ; elles sont exécutées de haut en bas.
#
# Si vous éliminez une ligne ici, LE COMMIT CORRESPONDANT SERA PERDU.
#
# Cependant, si vous effacez tout, le rebasage sera annulé.
#
# Veuillez noter que les commits vides sont en commentaire
```

```
r 58721c1 suppressing all old files (only the new ones are present now)
f 2ab0122 print register
f c505aec align_masks
f 0c84ccf self.new_value
f a136825 print
f a89b9bf print debut
f 9a15ba8 essai list
f 25b139f test priors
f 916ee4a modified main
f ac4df2d bug
f 2b71b26 error argstr
f ad7f737 removed template file
f 0beaf4b removed checked on priors
f 70c4ce2 Now a list or string
f fb23012 bug
f 5b46efe traits
f afa7c2e added -w in atropos and in macaque/ants params + removed mesh by default
f bff44c3 priors1.0
f 715e90d testing prior_weight = -1.0
f f108299 no return
f a2f29bb testing use priors
f 7088b06 outputnode
f 4cf413e no default weight
f c9b5c79 adding use_priors to macaque_ants
f 7944ece using use_priors for passing prior_weight
f fc500dd use_priors
f c892a2a no nii_to_mesh
f d68fbe7 corrected 0.0
f 46fa451 order
f c765dc2 use priors and no mesh in default params
f 364eebd segkmeans
f a2774f8 test if false

# Rebasage de 7776796..a2774f8 sur 7776796 (32 commandes)
#
# Commandes :
# p, f = picorer le commit
# r, reword = picorer le commit, mais reformuler son message
# e, edit = picorer le commit, mais s'arrêter pour le modifier
# s, squash = prendre le commit, mais le fusionner avec le précédent
# f, fixup = comme "squash", mais en éliminant son message
# x, exec = lancer la commande (reste de la ligne) dans un shell
# d, drop = supprimer le commit
#
# Vous pouvez réordonner ces lignes ; elles sont exécutées de haut en bas.
#
# Si vous éliminez une ligne ici, LE COMMIT CORRESPONDANT SERA PERDU.
#
# Cependant, si vous effacez tout, le rebasage sera annulé.
#
# Veuillez noter que les commits vides sont en commentaire
```

More advanced topics

Rebase:

- « git merge » will merge the two branches in there current states

- « git rebase » will move all modifications from a previous bifurcaction to the present state

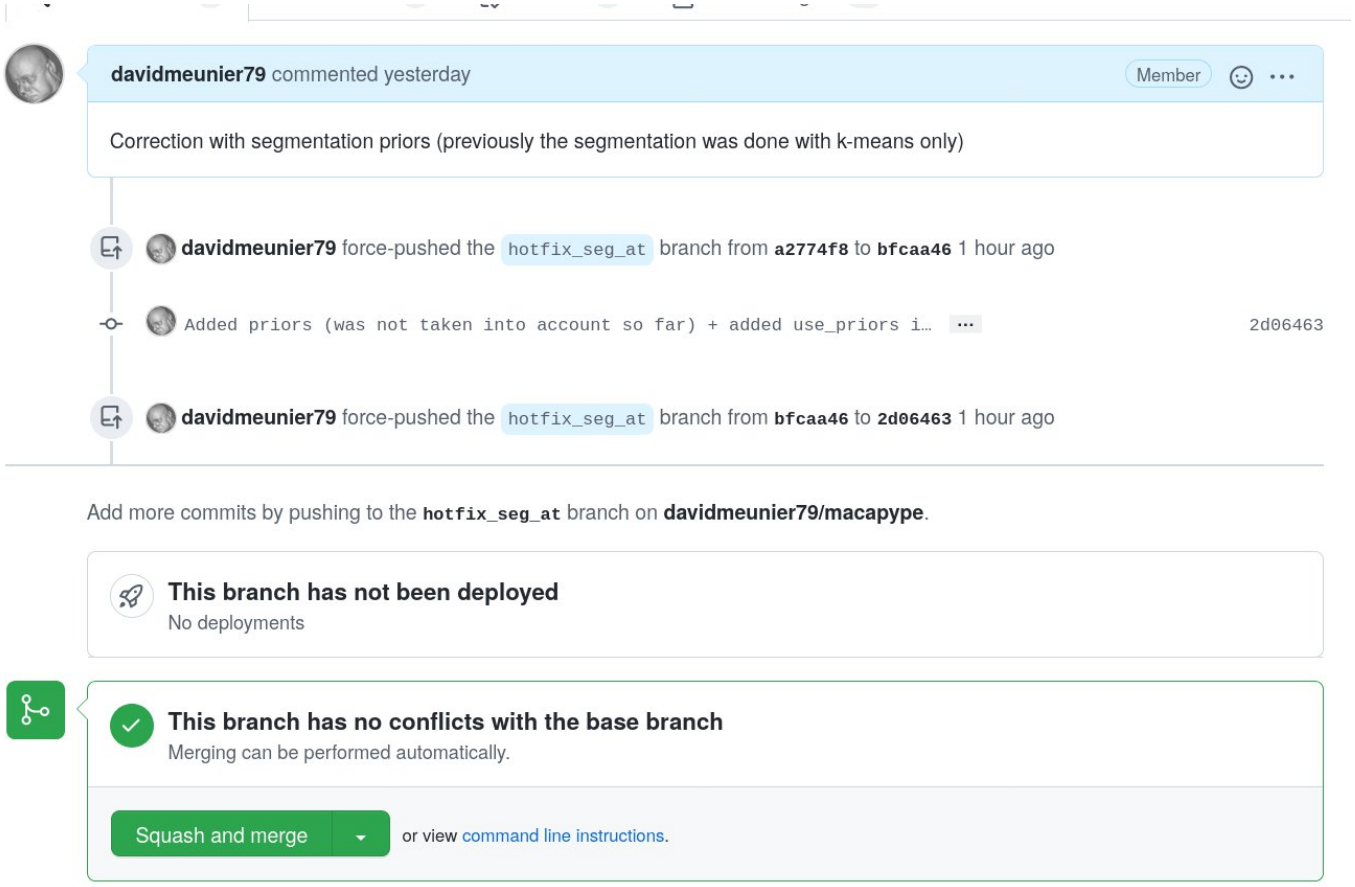
$ git fetch upstream

$ git rebase -i upstream/master

$ git pull -f origin my_feature

(needs to be force pushed, as

the commit history is modified)



More advanced topics

WHY Rebase ?

- Rebase allows all commits to be linear (increasing the readabability of the history of modifications)
- It is however complicated if some merge/branching by other people have happened inbetween the previous and current steps
- Requires a way of coding will small modifications (i.e. A PR is done very quickly, every few days, not few weeks/months)

→ For security, before rebase, always make a copy of the directory …

Thanks for your attention !