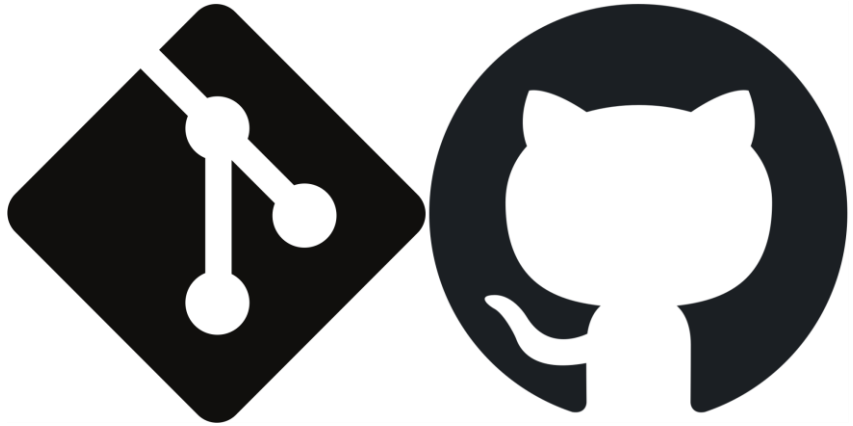
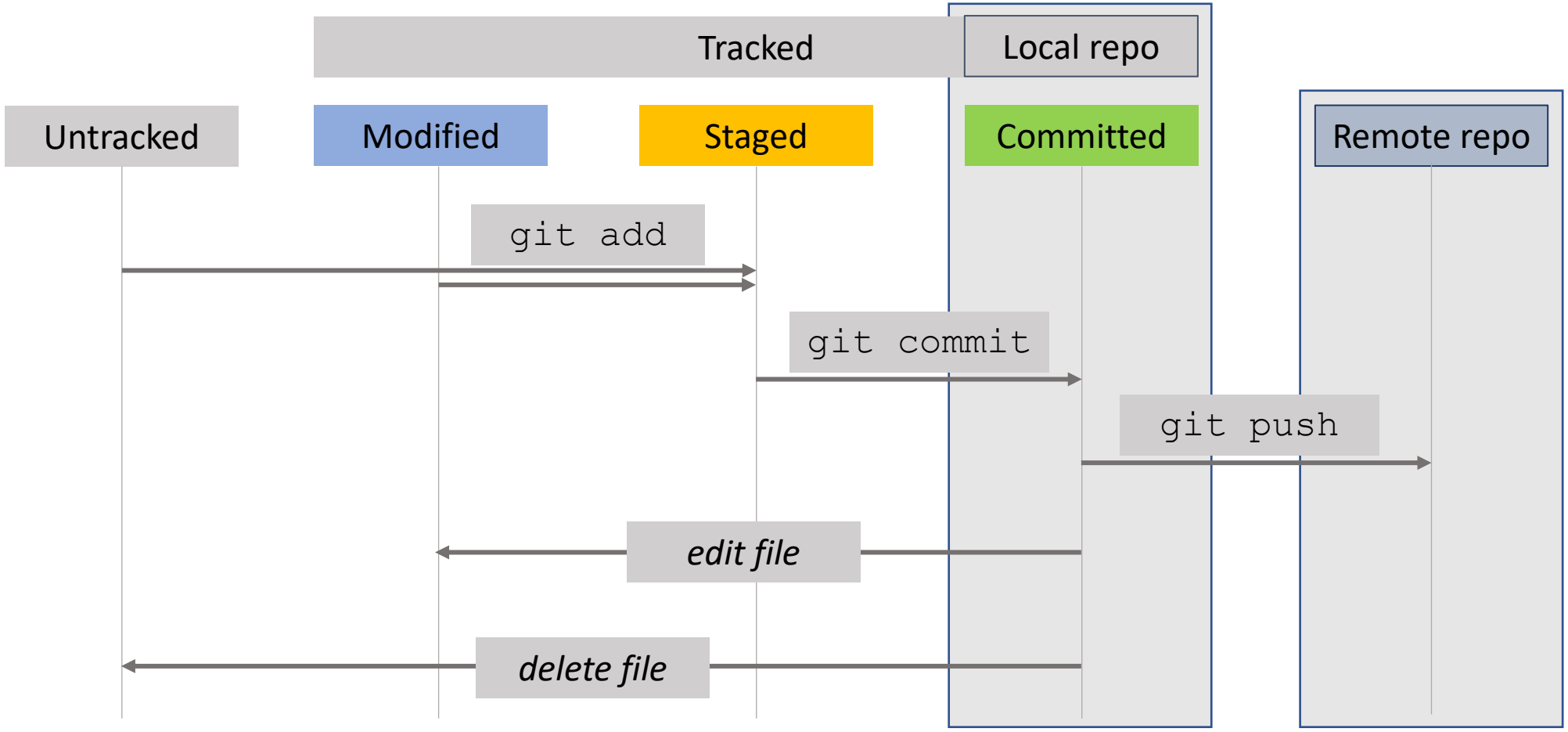


Introduction to Git & GitHub

Jennifer Graham
Tiago Silva
David Ryder
Stephen Gregory
Joe Ribeiro

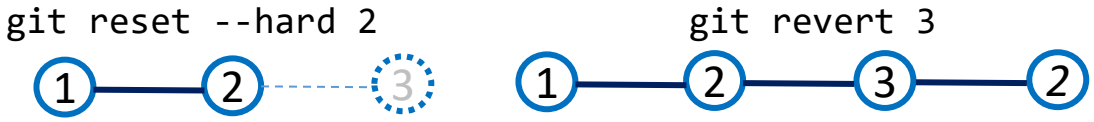
Session 3
16th May 2025

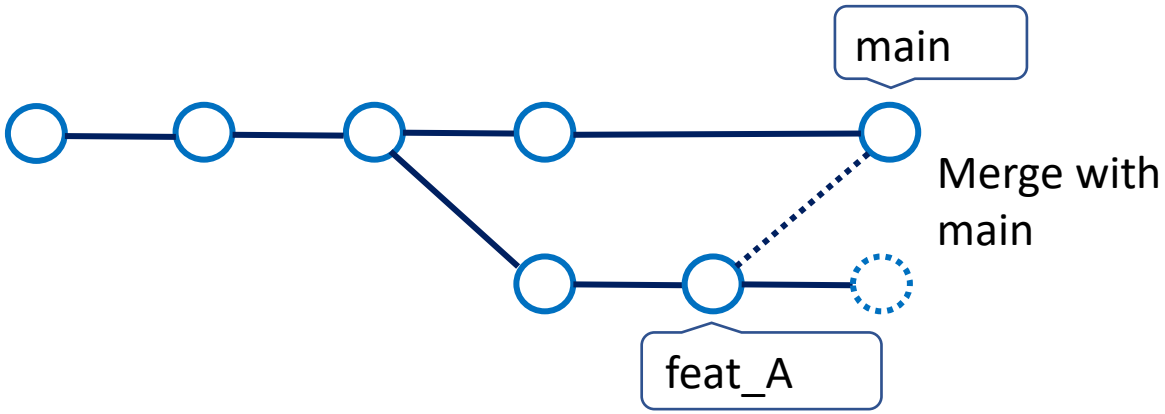
Recap from Sessions 1 & 2 …


Tracked
Local repo
Untracked
Modified
Staged
Committed
Remote repo
git add
git commit
git push
edit file
delete file

```
git init                :: turn current directory into git repository.
git add <file>          :: stage the file for commit i.e. track changes.
git commit              :: confirm changes with message/explanation.
git push                :: publish/back-up changes on GitHub (remote server).


git status              :: show which files have been tracked or modified.
git diff <file>         :: show difference between current file and last commit.
git log <file>          :: show commit history [for file]
```

last commit
1
2
3


git reset --hard 2
1
2
3
git revert 3
1
2
3
2


main
Merge with
main
feat_A

Clearing up mistakes

Restore previous version of file

```
git checkout  [<commit>] -- <path>
```

Go back to last commit, discarding all changes in working directory
Discard all commits after [<commit>]

```
git reset --hard HEAD
git reset --hard [<commit>]
```

Undo git add (undo staging or tracking file)

```
git reset [HEAD] <path>
```

Undo changes introduced by single commit, recording this in history

```
git revert <commit>
```

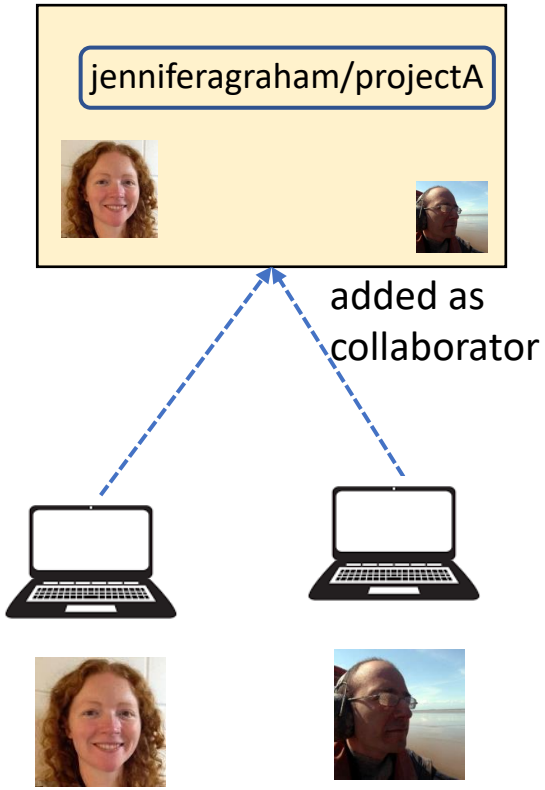
```
git branch                                  :: List local branches
                                                (* = you are here).

git branch <branch-name>                    :: Create new branch
git checkout <branch-name>                  :: Change to <branch-name>
                                                (updates local directory)

git diff <branch1>..<branch2> <path>
                                            :: Check difference between two branches
                                                (in optional <path>)

git merge <branch-name>                     :: Merge changes from <branch-name>
                                                into current branch
```

How to collaborate?

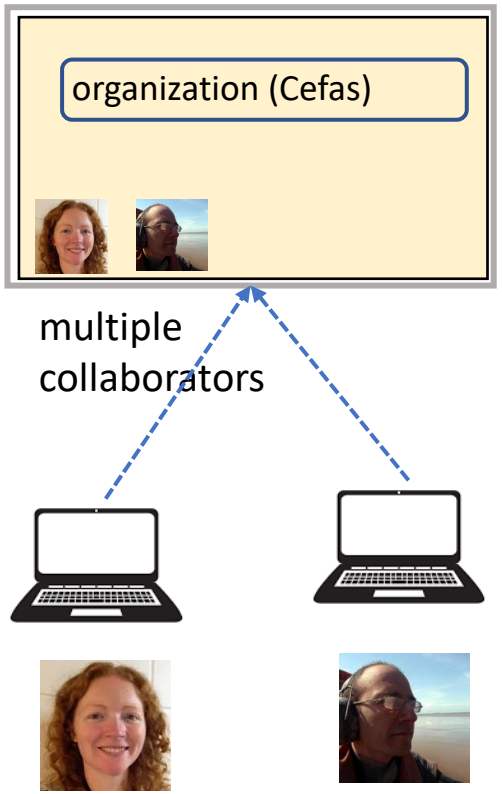
Collaboration models

GitHub

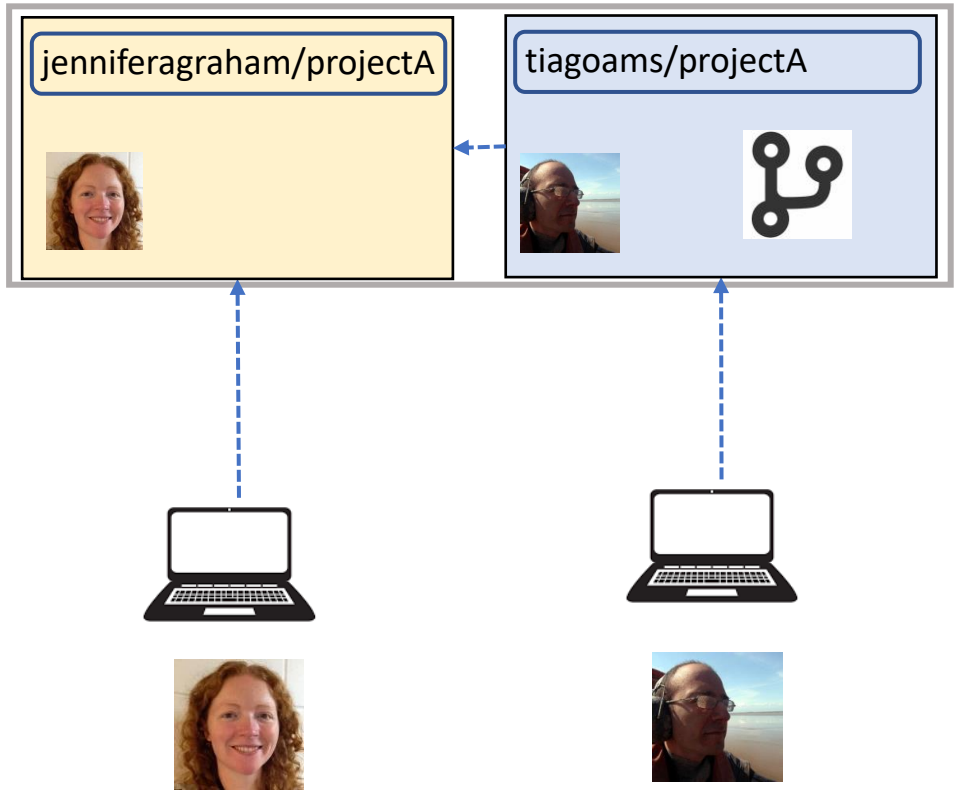
Personal account + add collaborator



GitHub Organization (CefasRepRes, UEABIO, UeaPy)

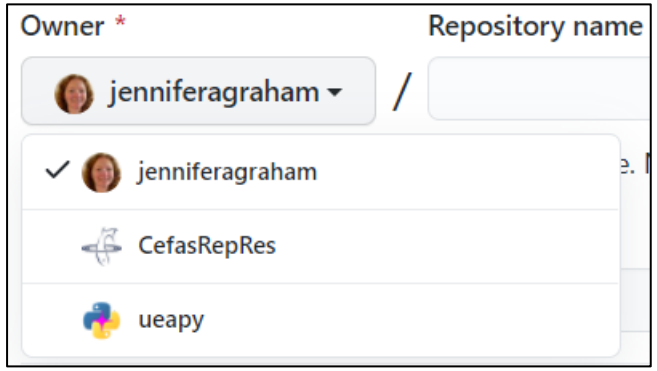


Forked + pull request



Organisations

- Organisations can add further functionality on GitHub.
  - For those at Cefas, we have CefasRepRes.
    - Further guidance here: https://www.github.com/CefasRepRes/cefas-guides/
  - For those elsewhere, if you don't have one already, you can always create your own…
- Develop private repositories that are still visible to all members.
- Easily share and organise work within the organisation.
- Repositories are technically "owned" by the organisation rather than person creating them.
  - NB. Select the correct "owner" on creation.



Centre for Environment Fisherie

github.com/CefasRepRes

CefasRepRes

Type / to search

Overview | Repositories 504 | Projects 26 | Packages | Teams 20 | People 116

Centre for Environment Fisheries and Aquaculture Science (Cefas)

Cefas is a world leader in marine science and technology, providing innovative solutions for the aquatic environment, biodiversity and food security

Follow

51 followers | UK | https://www.cefas.co.uk | github@cefas.gov.uk

Pinned

cefas-guides Private

Everything you need to know about how to use GitHub and CefasRepRes

☆ 2

wiki Private

Experimental Cefas wide wiki documentation

☆ 1

Repositories

Find a repository... | Type | Language | Sort | New

gbelsham-shellfish-classification-application Private

A tkinter Python GUI script that brings together numerous scripts used by the Shellfish Hygiene Classification team in a user friendly application

Jupyter Notebook ☆ 0 0 0 0 Updated 37 minutes ago

gbelsham-shellfish-classification-carmarthan-labs-results-import-checks Private

Used to check Carmarthan lab uploads against data uploaded to the SHS database

View as: Member

You are viewing the README and pinned repositories as a member of the Centre for Environment Fisheries and Aquaculture Science (Cefas) organization.

You can create a README file visible only to members of the organization.

People

View all

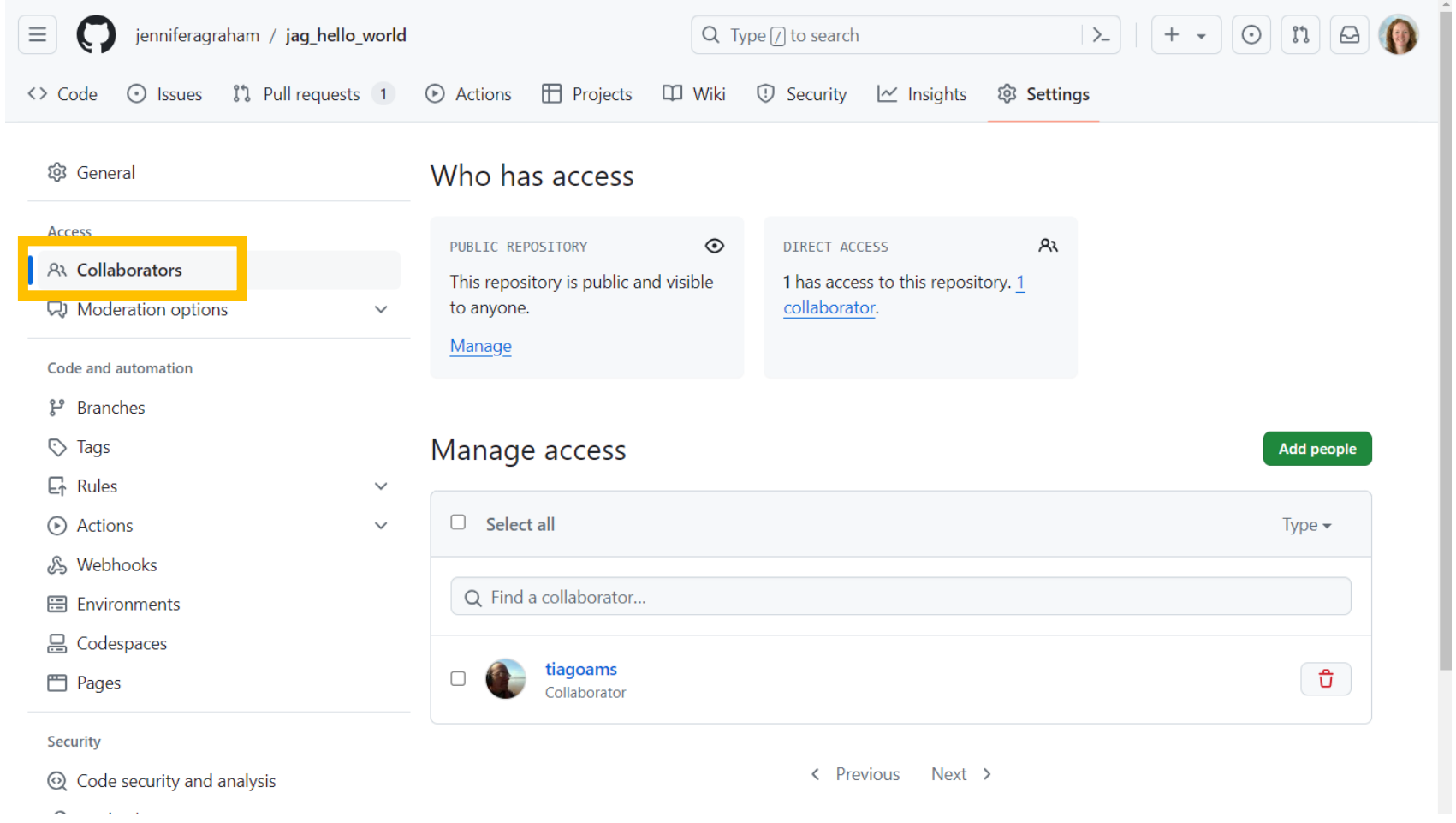
Top languages

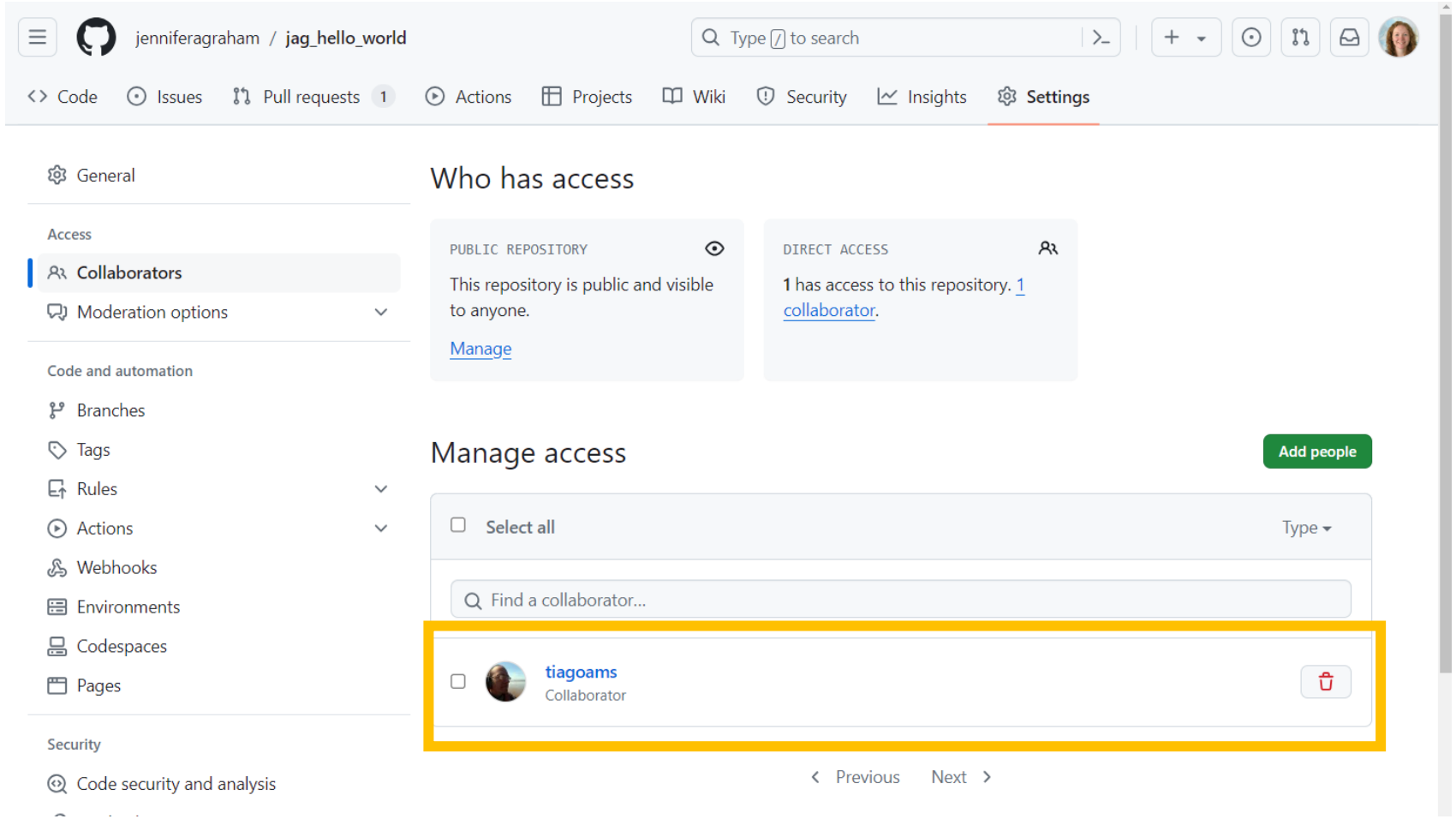
R Python Jupyter Notebook

HTML Fortran

Controlling repository access…

- Collaborators can be added to each repository.





- For personal repositories, there are two levels of permission:
  - **Repository owners** (Admin permissions)
  - **Collaborators** (aka Write access etc.)

- Only the owner can invite new collaborators.

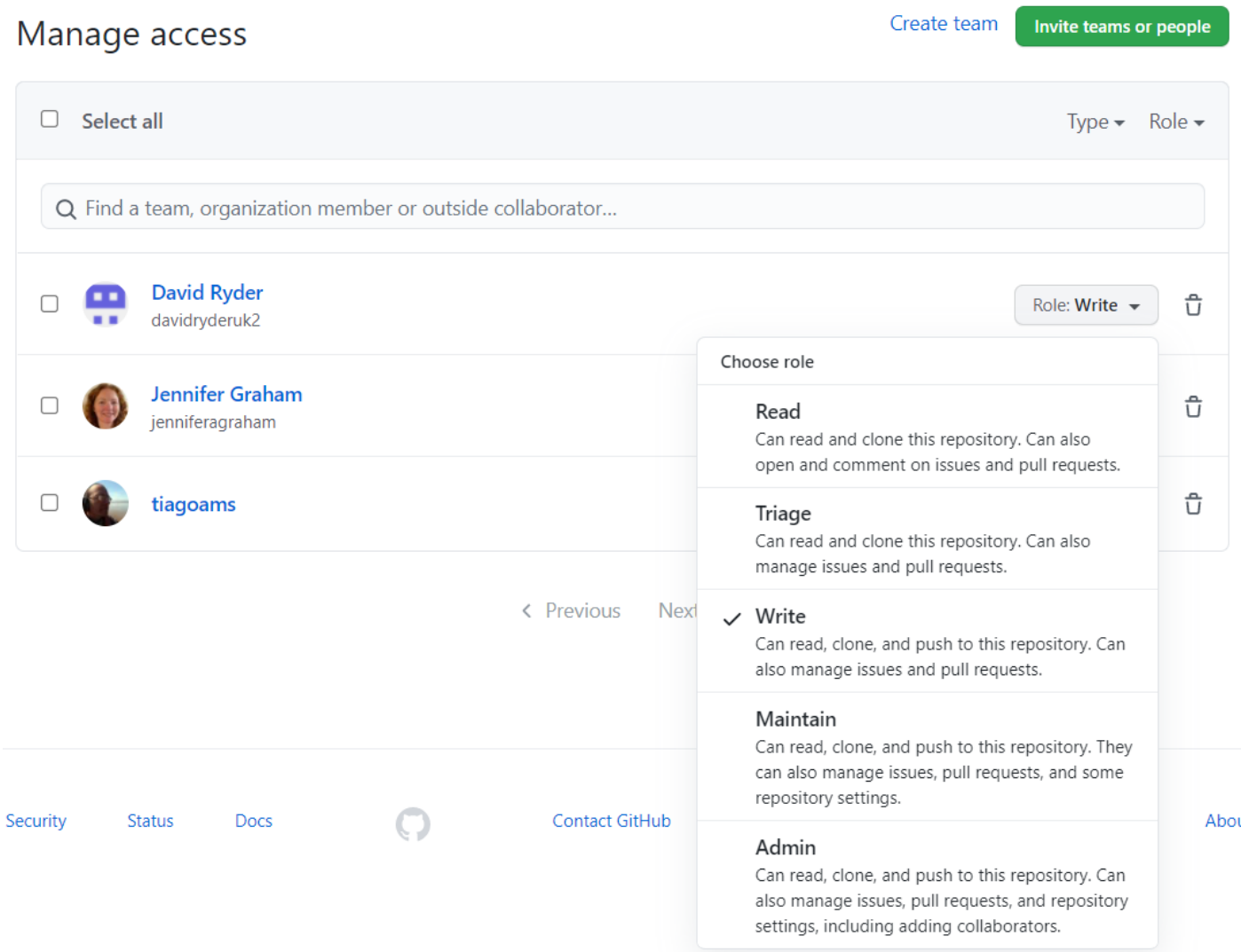
- For organisation repositories, there are many levels available:

Role	Access
Admin	Full access i.e. repository owners
Maintain	Manage the repository with access to all but most sensitive settings.
Write	Can push changes to repo
Triage	Can manage issues but no write access.
Read	View code, make comments, but not change

- Only those with **Admin** access can invite new collaborators.

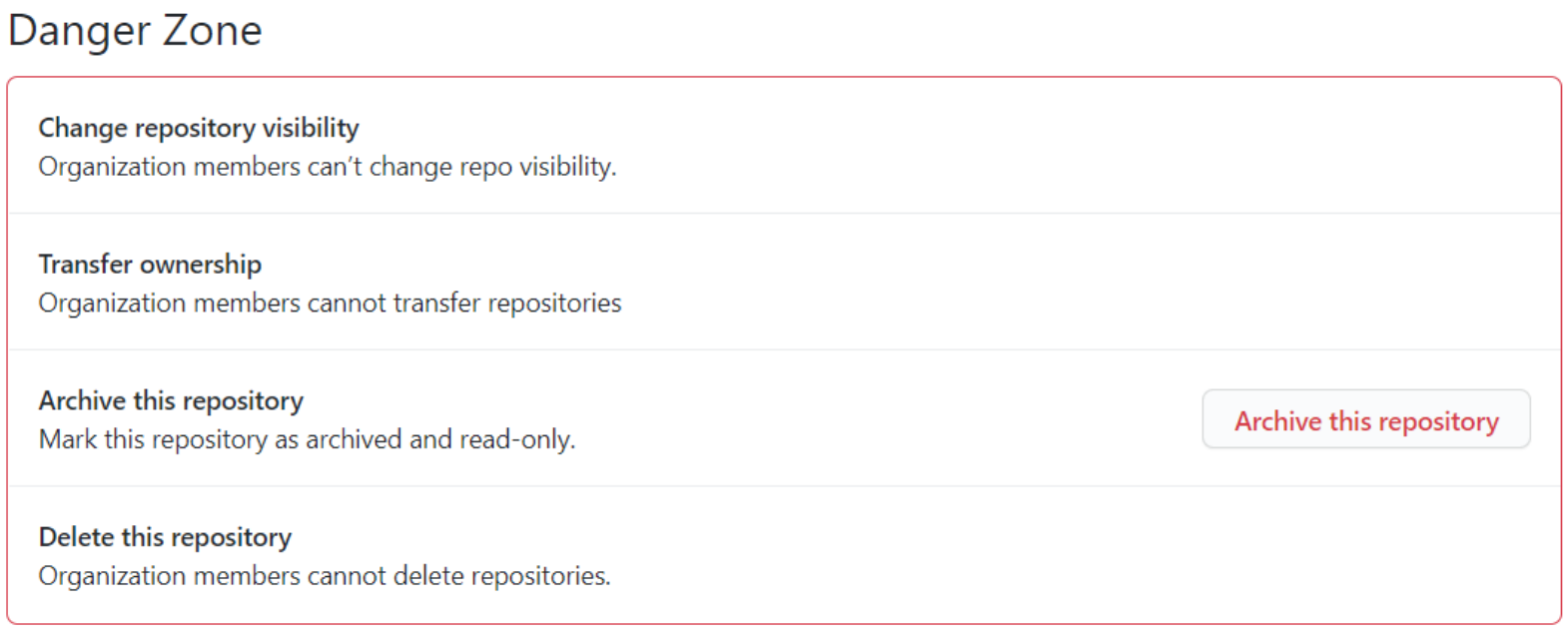
- Access can be grouped via “teams” as well as individuals.

- NB. Outside collaborators can be invited (don’t need to be a member to collaborate)



Restricted admin access

- Within an organisation, repositories are technically owned by the organisation, not individual.
  - Some settings can only be changed/managed by the **organisation owner** e.g.



- Organisation owners can decide on the level of access for members.
- Only owners of the organisation can invite/approve new members.

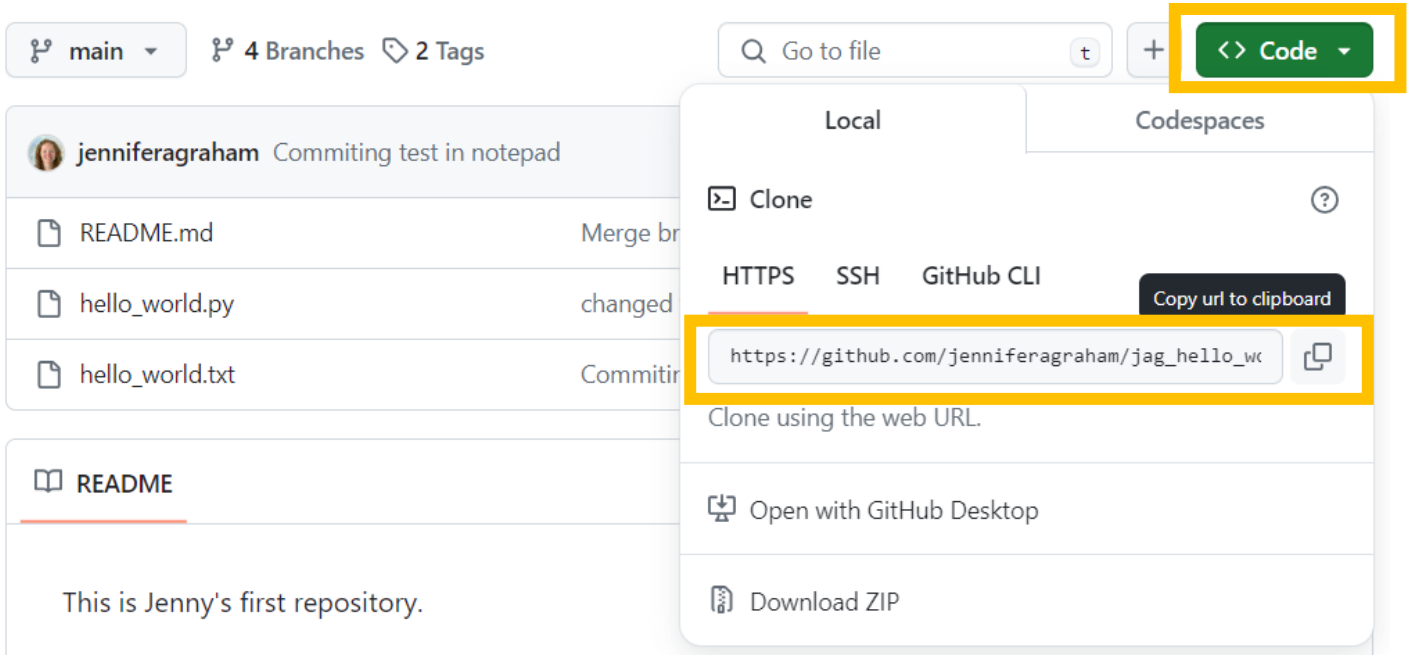
(How) can I modify someone else's repository?

It depends on your access…

Admin, write or collaborator access…

- Use the same workflow discussed already.
  - You can push changes to main, and create branches for developments.

- Use `git clone` to make a local copy.




```
git clone <copied-url>

download into new local repository
```

fetch, merge (or pull)

- Once you start collaborating, you need to be aware that others may (will) commit and push changes.
  - If you try to push changes, you may see an error message…
- To bring your local repository up to date with remote changes:

```
git fetch          :: brings in information on changes from the remote
git merge          :: merge those changes into your repository

--

git pull           :: short cut to fetch and merge in one step (if confident
                   that there are no conflicts)
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ 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

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$
```

```
MINGW64:/c/Users/JG10/OneDrive - CEFAS/GitHub/Repos/jag_hello_world
JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ 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

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ git push
To https://github.com/jenniferagraham/jag_hello_world.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/jenniferagraham/jag_hello_world.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$
```

```
git pull
:: fetch updates from remote and merge in one step
```

MINGW64:/c/Users/JG10/OneDrive - CEFAS/GitHub/Repos/jag_hello_world

```
JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ git fetch
```

git fetch

:: fetch retrieves updates from remote server, but does not change your local repository files.

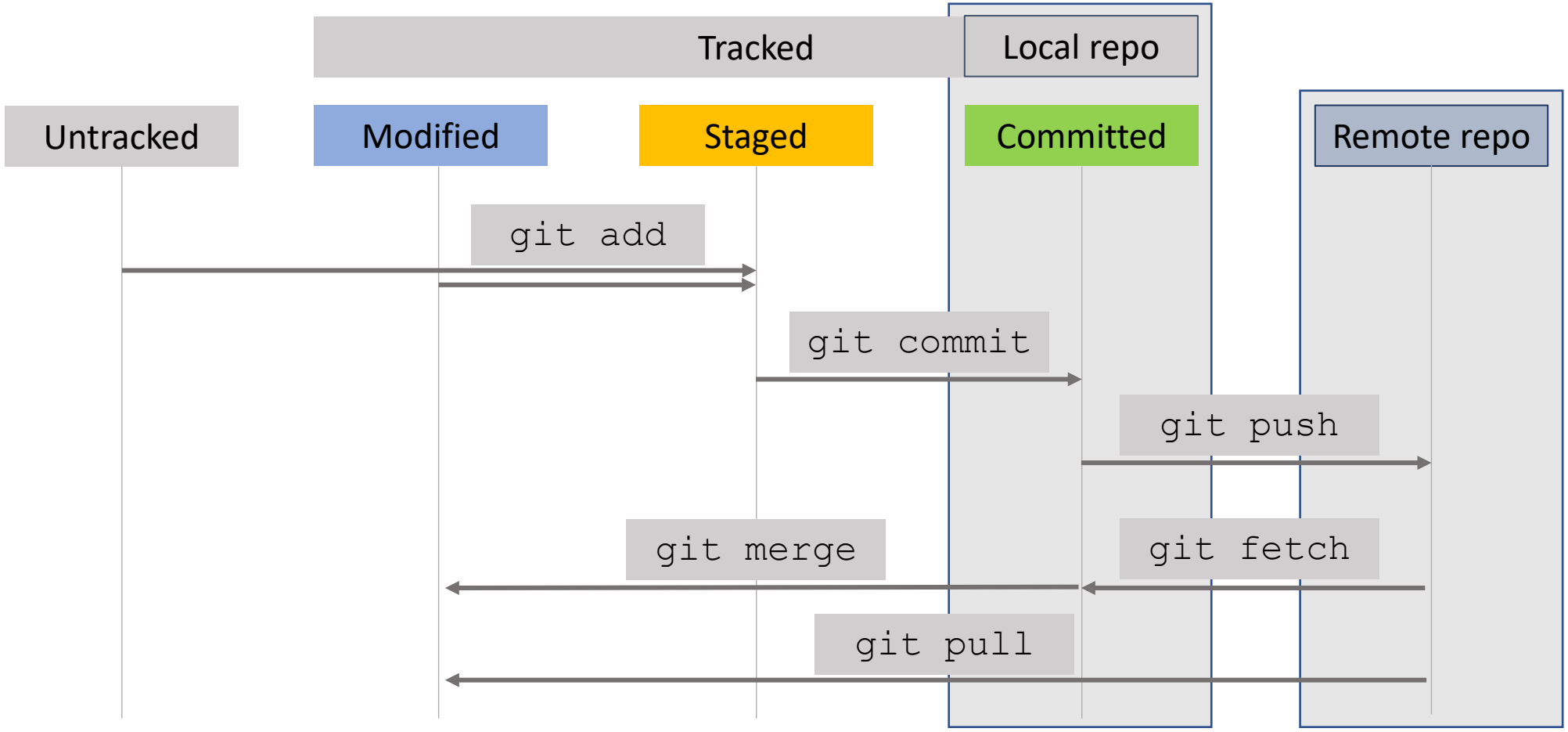
```
JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/jenniferagraham/jag_hello_world
   ddd7a97..c07fb9d  master     -> origin/master

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ 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
Unpacking objects: 100% (3/3), done.
From https://github.com/jenniferagraham/jag_hello_world
   ddd7a97..c07fb9d  master     -> origin/master

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ git merge
```

```
git merge
:: merge retrieved commits into local repository.
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ 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
Unpacking objects: 100% (3/3), done.
From https://github.com/jenniferagraham/jag_hello_world
   ddd7a97..c07fb9d  master     -> origin/master

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$ git merge
Auto-merging README.md
Merge made by the 'recursive' strategy.
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag_hello_world (master)
$
```

```
git clone <copied-url>    :: download into new local repository

git fetch         :: brings in information on changes from the remote
git merge         :: merge changes into your local repository

git pull          :: short cut to fetch and merge in one step
```


Tracked
Local repo
Untracked
Modified
Staged
Committed
Remote repo
git add
git commit
git push
git merge
git fetch
git pull

If you have only read access…

e.g. For public repositories, or those within your organisation.

- You can still clone the repository, then track locally as normal.
- However, if you modify the repository, you won't be able to push changes back to the remote repository.

- If you want to make a copy and intend to make developments, consider forking…

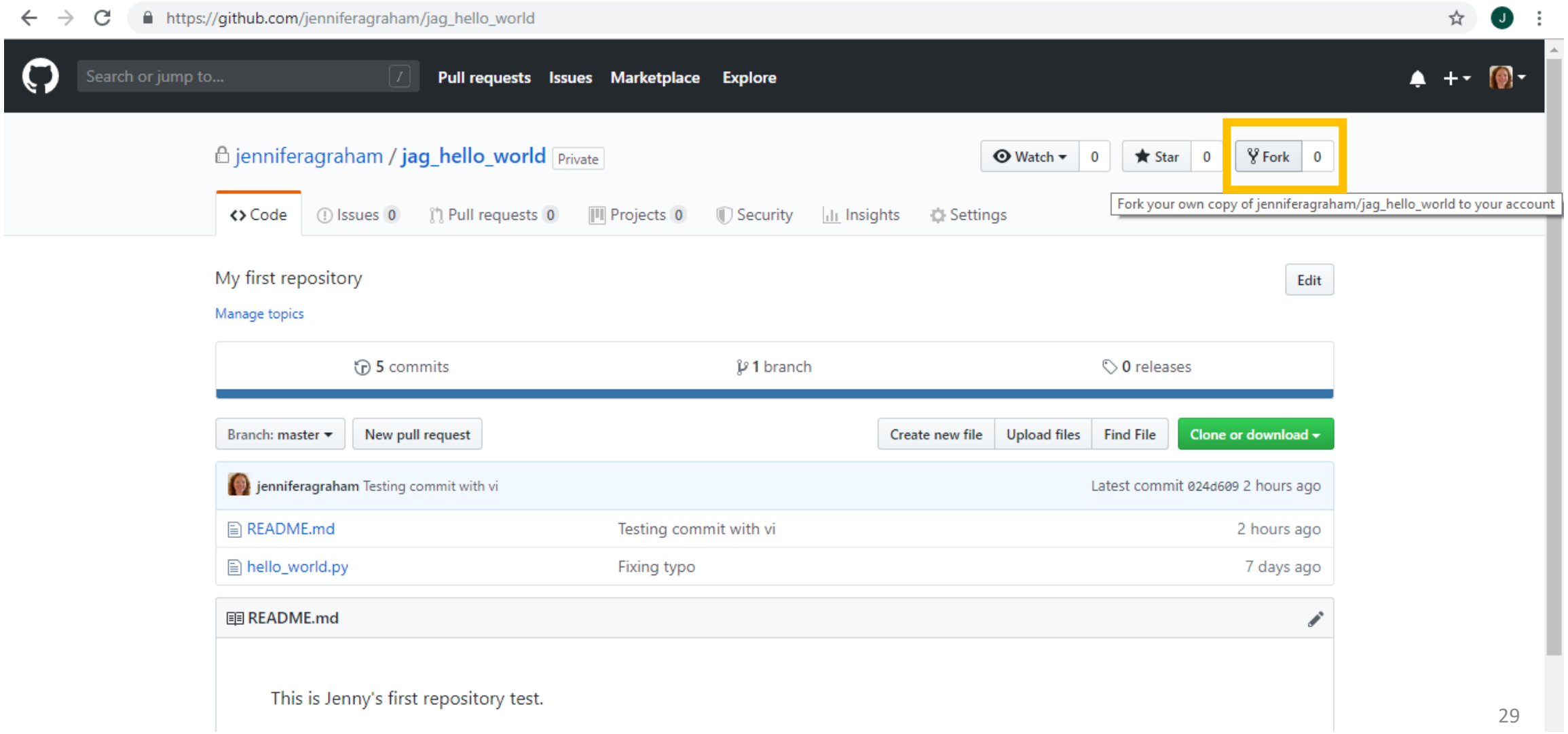
What is a fork?

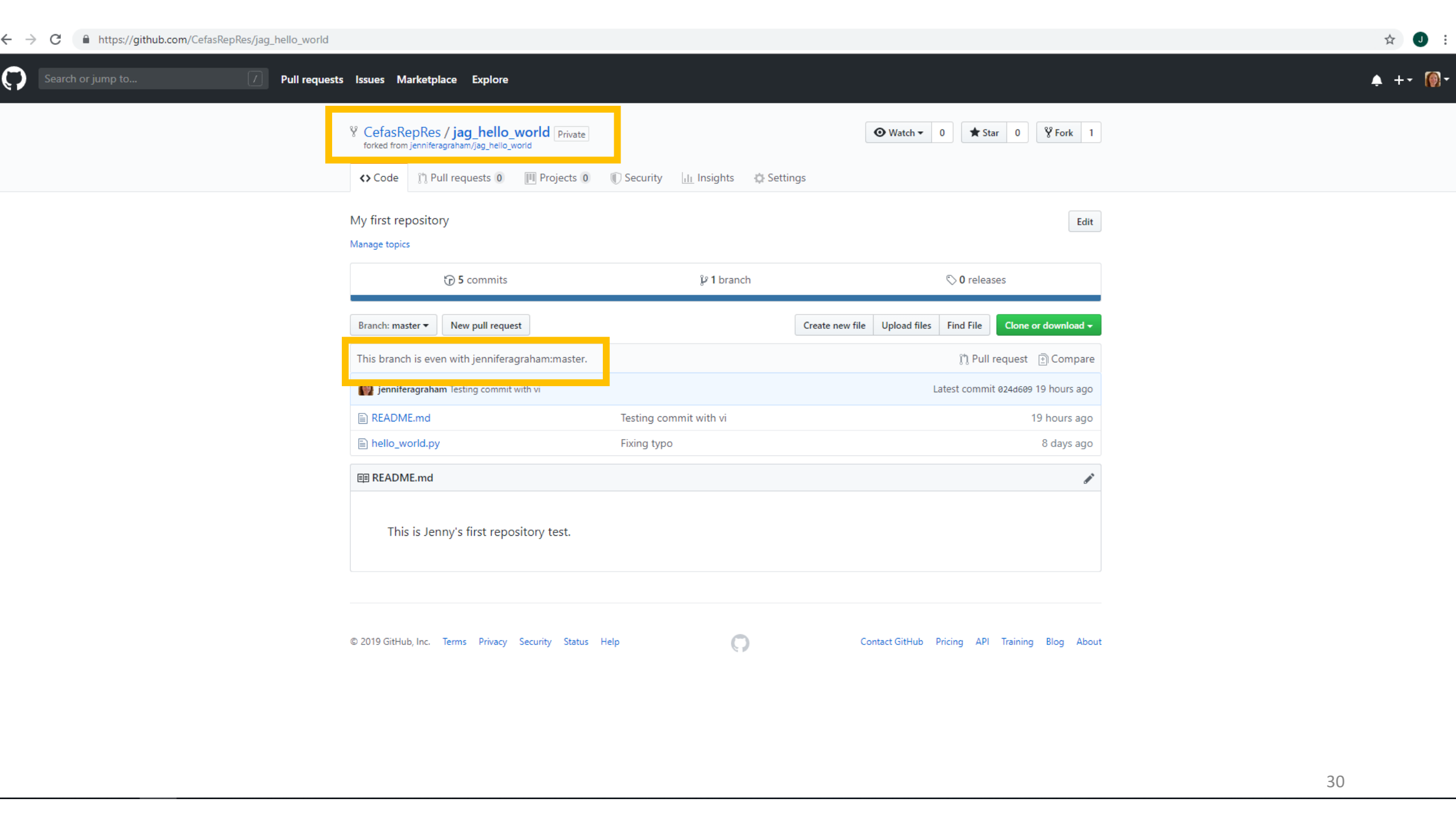
- Similar to a branch, in that both can be used to develop new features, away from the main branch.
- In basic terms, the difference is:
  - If you have write access for the repository -> branch.
  - If you don’t have write access -> fork.

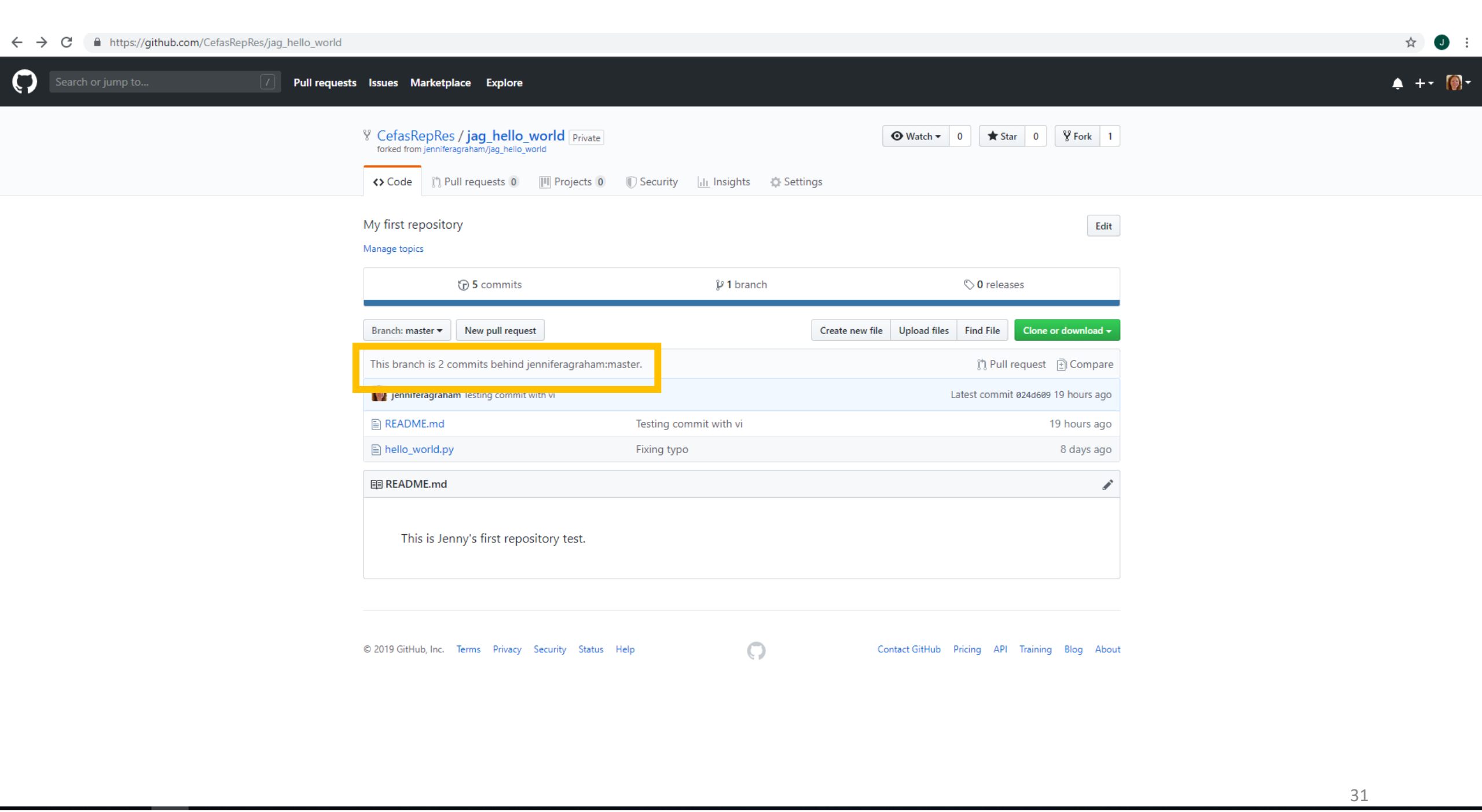
- Forking creates a copy of the original repository in your ownership (including its history).
- Repository remains linked to the original (main) so you can see how the developments diverge.
- From a fork, you must create a “pull request” to merge changes.
  - This will notify the code owner, and allow them to review changes before deciding whether or not to accept/merge (i.e. request them to pull changes into their repository).

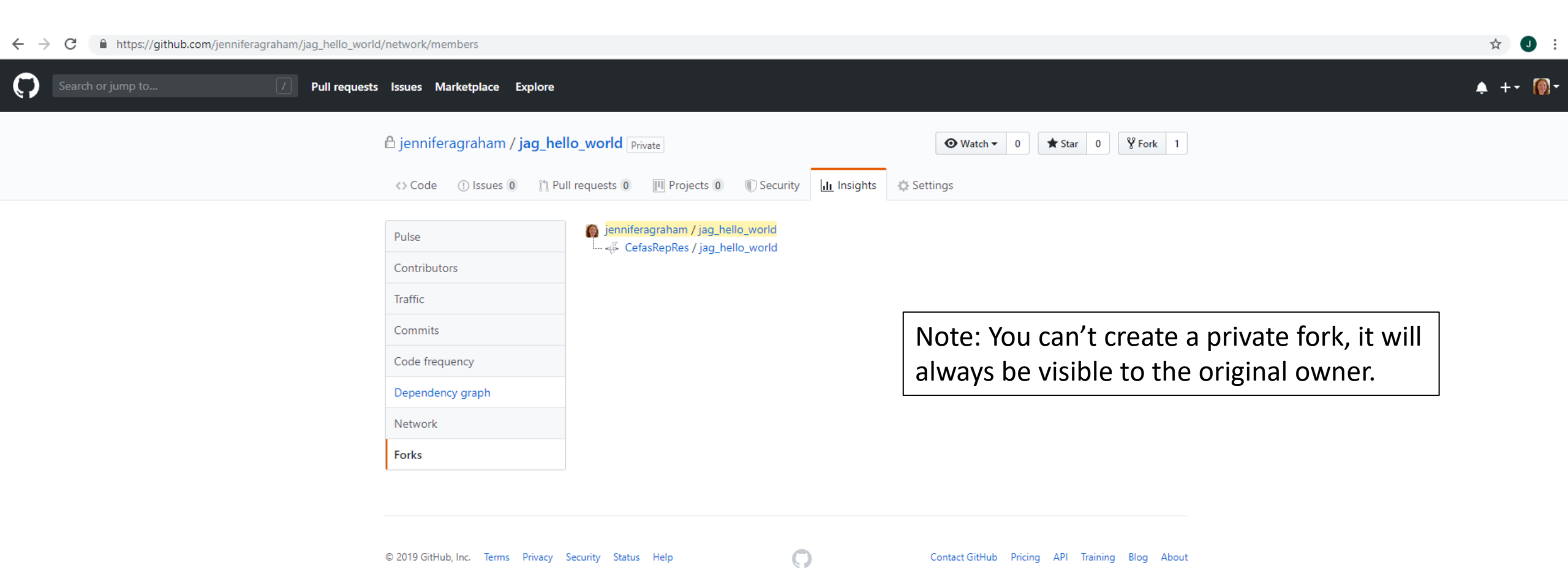
How to create a fork?

Note: you can't create a fork owned by the same individual or organisation as the original.




https://github.com/CefasRepRes/jag_hello_world
Search or jump to...
Pull requests
Issues
Marketplace
Explore
CefasRepRes / jag_hello_world
Private
forked from jenniferagraham/jag_hello_world
Watch 0
Star 0
Fork 1
Code
Pull requests 0
Projects 0
Security
Insights
Settings
My first repository
Edit
Manage topics
5 commits
1 branch
0 releases
Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download
This branch is even with jenniferagraham:master.
Pull request
Compare
jenniferagraham Testing commit with vi
Latest commit 024d609 19 hours ago
README.md
Testing commit with vi
19 hours ago
hello_world.py
Fixing typo
8 days ago
README.md
This is Jenny's first repository test.
© 2019 GitHub, Inc.
Terms
Privacy
Security
Status
Help
Contact GitHub
Pricing
API
Training
Blog
About


https://github.com/CefasRepRes/jag_hello_world
Search or jump to...
Pull requests
Issues
Marketplace
Explore
CefasRepRes / jag_hello_world
Private
forked from jenniferagraham/jag_hello_world
Watch 0
Star 0
Fork 1
Code
Pull requests 0
Projects 0
Security
Insights
Settings
My first repository
Edit
Manage topics
5 commits
1 branch
0 releases
Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download
This branch is 2 commits behind jenniferagraham:master.
Pull request
Compare
jenniferagraham Testing commit with vi
Latest commit 024d609 19 hours ago
README.md
Testing commit with vi
19 hours ago
hello_world.py
Fixing typo
8 days ago
README.md
This is Jenny's first repository test.
© 2019 GitHub, Inc.
Terms
Privacy
Security
Status
Help
Contact GitHub
Pricing
API
Training
Blog
About



Note: You can't create a private fork, it will always be visible to the original owner.

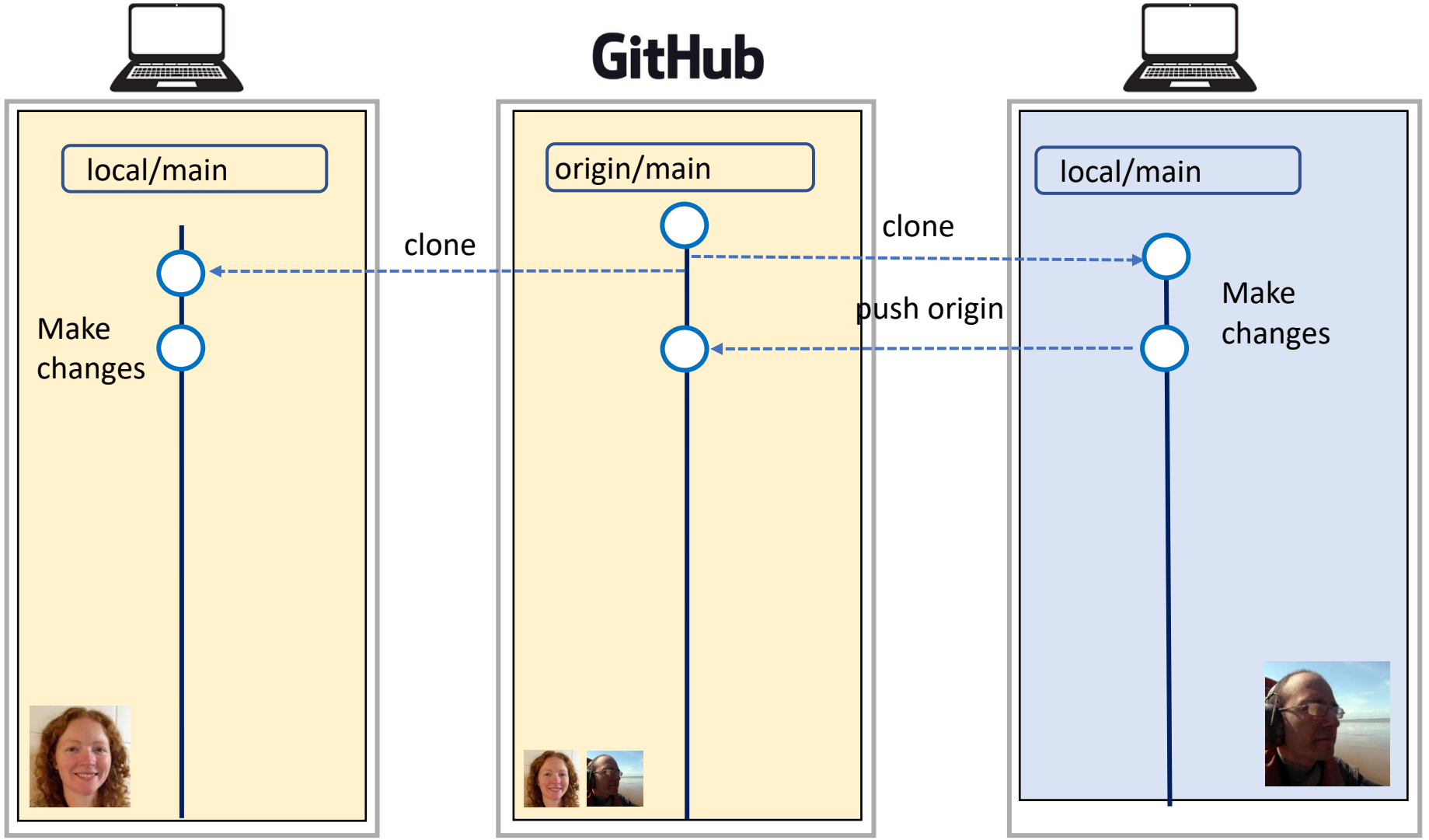
Branch	Fork
• Created in your repository • You have total control over what you merge • Pull requests are sent to your own repository • Always intended to merge successful branches into master	• Creates an independent copy of another person's repository on your account so you can make changes • You can suggest changes to the original project but they don't have to merge them • Pull requests automatically go back to the original project • Allows you to take a GitHub project in a different direction - may never intend to merge with original project

Collaboration workflows…

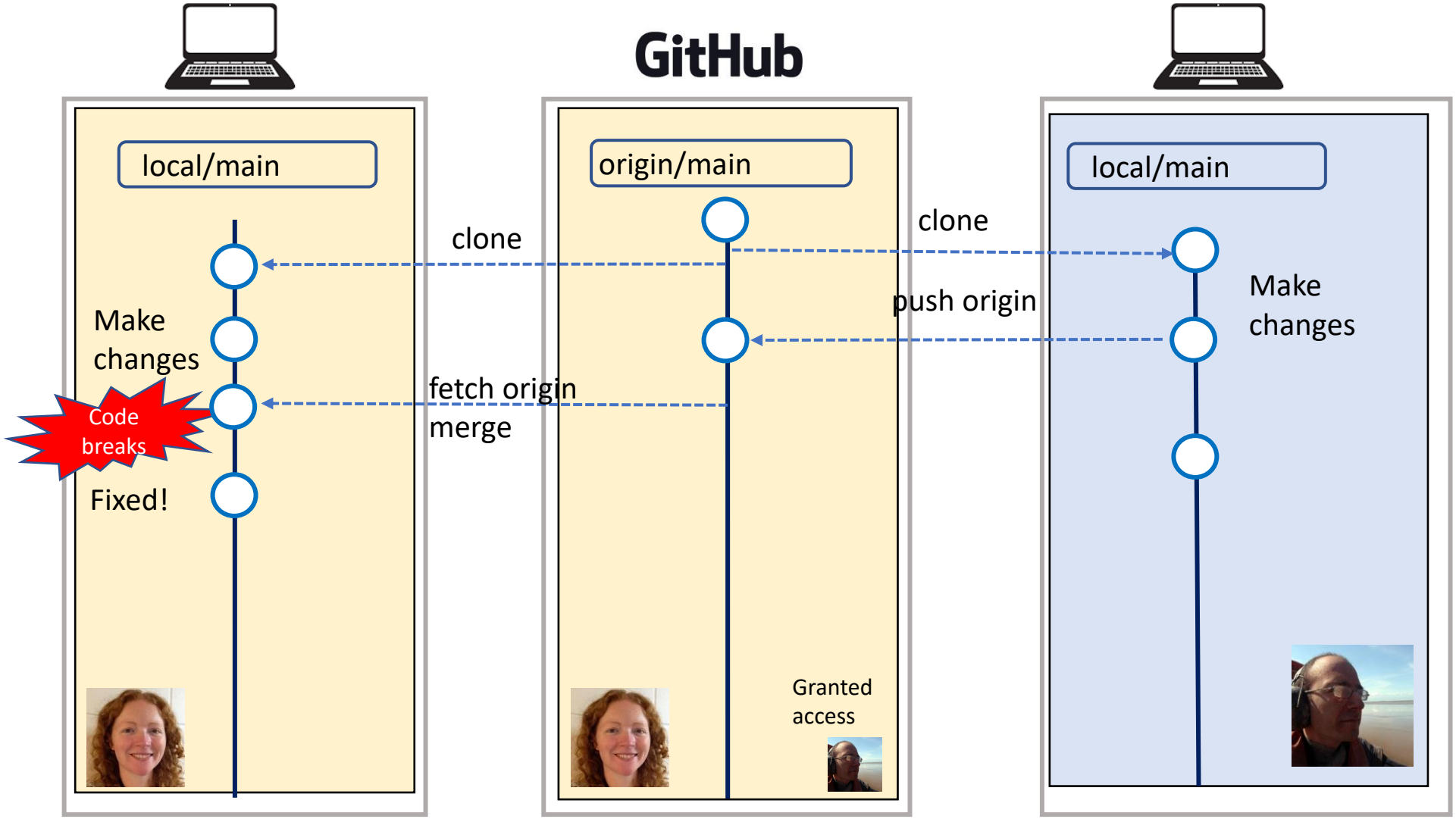
Collaboration workflows…

- git is just the tool
- Many workflows can be used, making use of various features available, and will depend on:
  - number of collaborators
  - frequency code is written
  - type of organisation (community, corporation)
  - criticality of the code

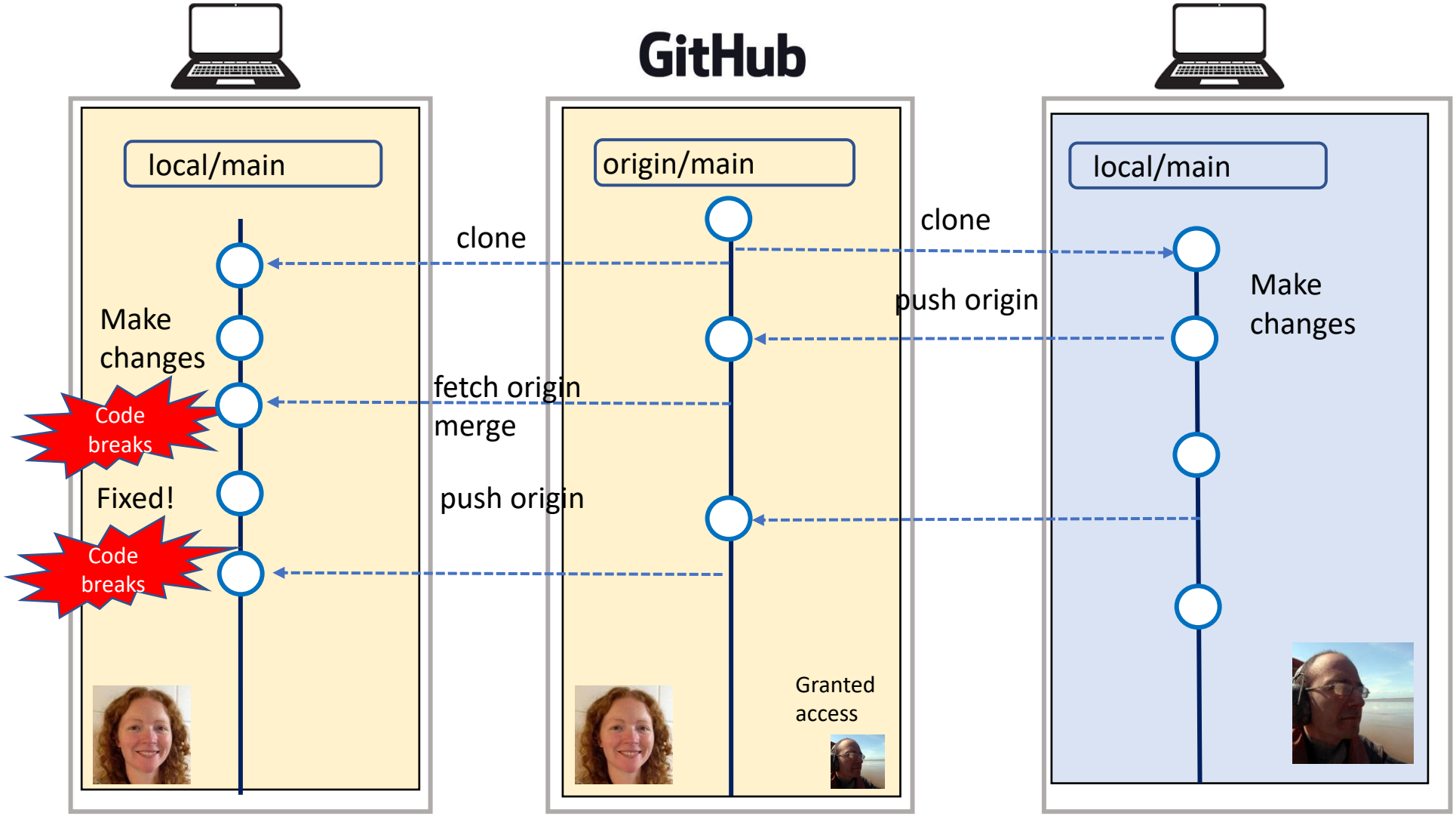
Bad workflow: competitors vs. collaborators



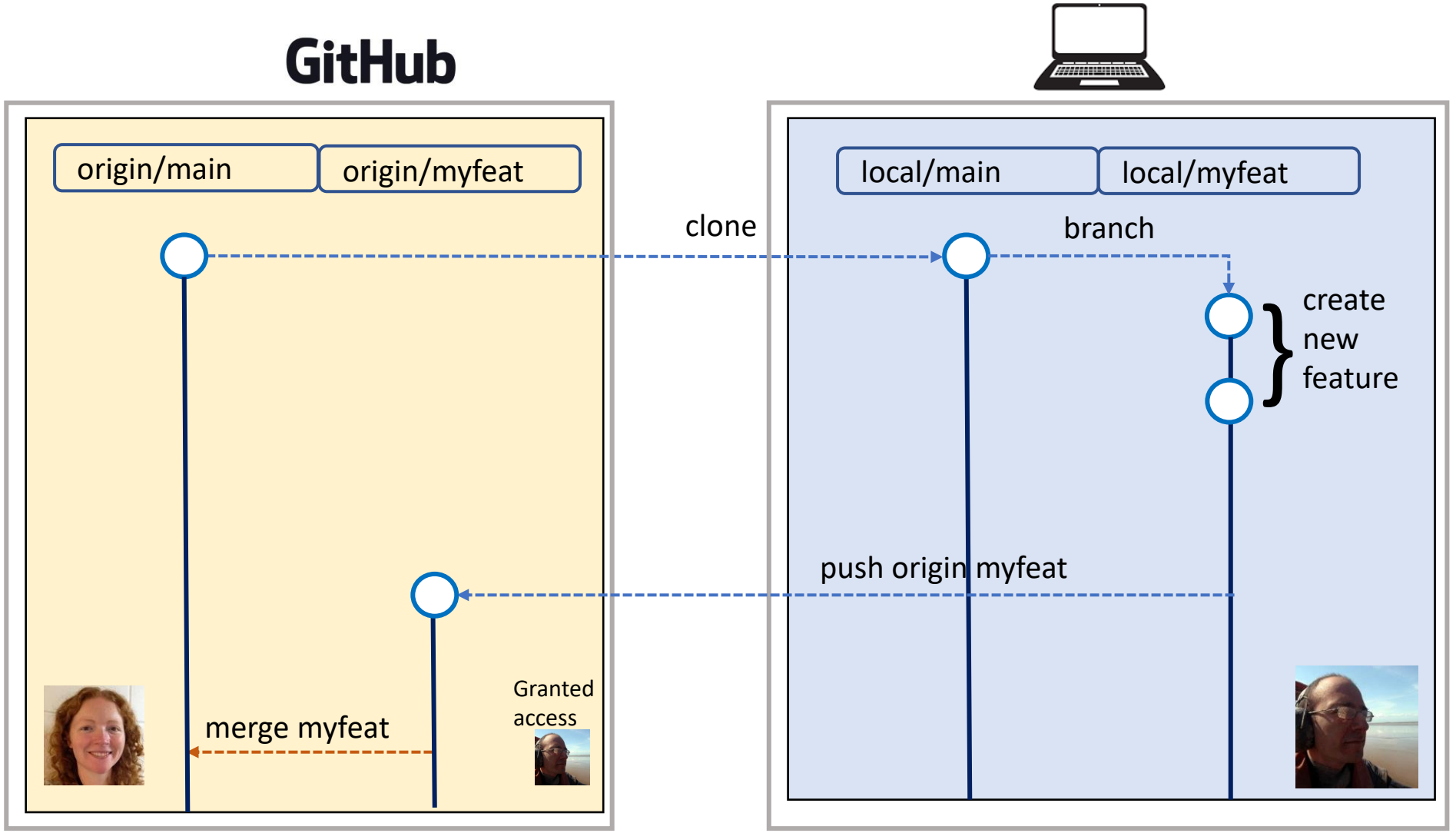
Bad workflow: competitors vs. collaborators



Bad workflow: competitors vs. collaborators



Shared access workflow: feature branches



Shared access workflow: creating and pushing a branch

1. Make a local repo

```
$ git clone https://github.com/jenniferagraham/jag_hello_world.git
Cloning into 'jag_hello_world'...
$ cd jag_hello_world/
$
```

Shared access workflow: creating and pushing a branch

1. Make a local repo
2. Create branch

```
$ git clone https://github.com/jenniferagraham/jag_hello_world.git
Cloning into 'jag_hello_world'...
$ cd jag_hello_world/
$ git branch fixreadme
$ git checkout fixreadme
$ git branch
* fixreadme
  main
$
```

Shared access workflow: creating and pushing a branch

1. Make a local repo
2. Create branch
3. Modify

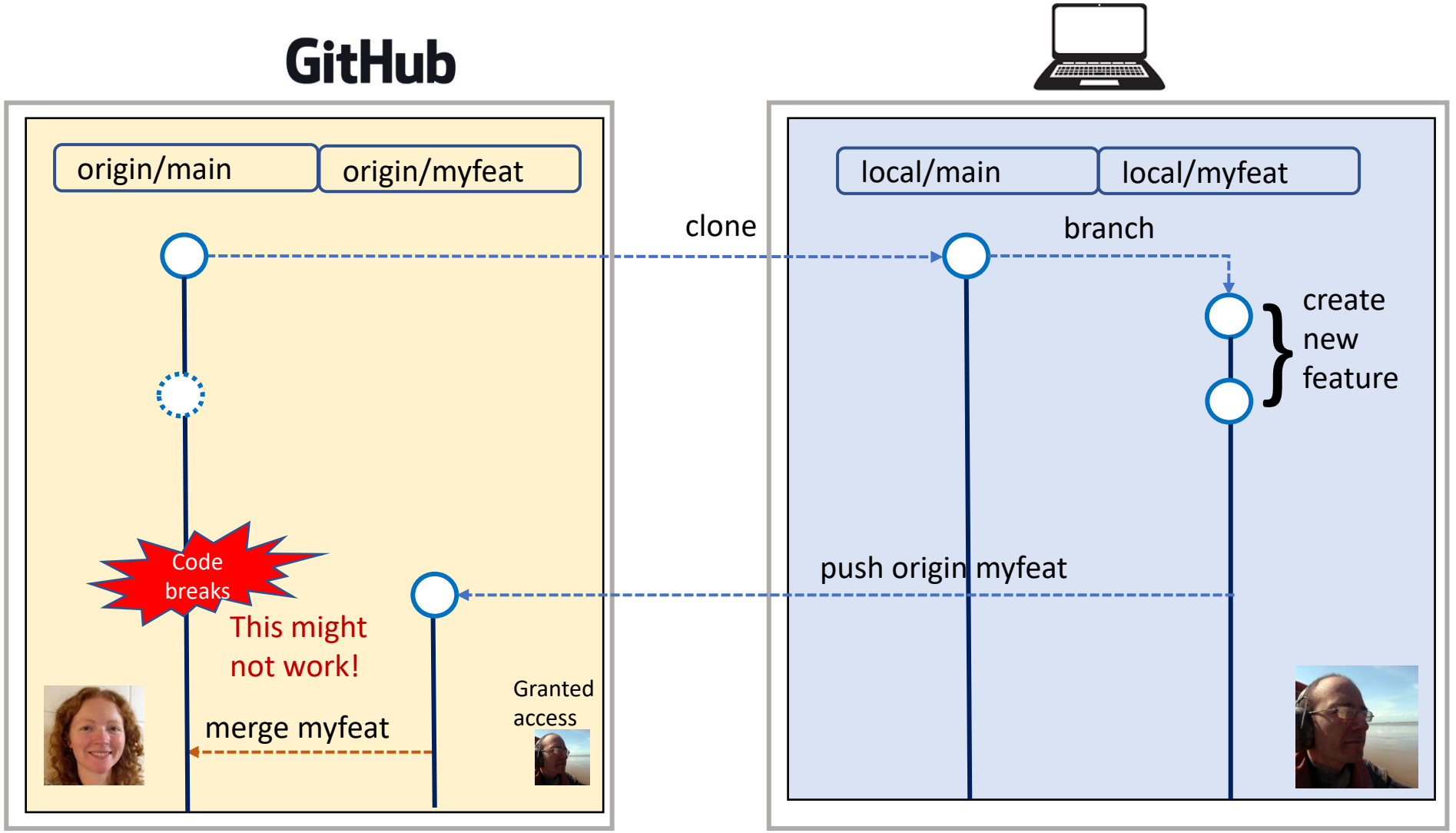
```
$ git clone https://github.com/jenniferagraham/jag_hello_world.git
Cloning into 'jag_hello_world'...
$ cd jag_hello_world/
$ git branch fixreadme
$ git checkout fixreadme
$ git branch
* fixreadme
  main
$ notepad README.md
$ git commit -am "Update name of UEA cluster"
[main 1ba2db8] Update name of UEA cluster
1 file changed, 1 insertion(+), 1 deletion(-)
$
```

Shared access workflow: creating and pushing a branch

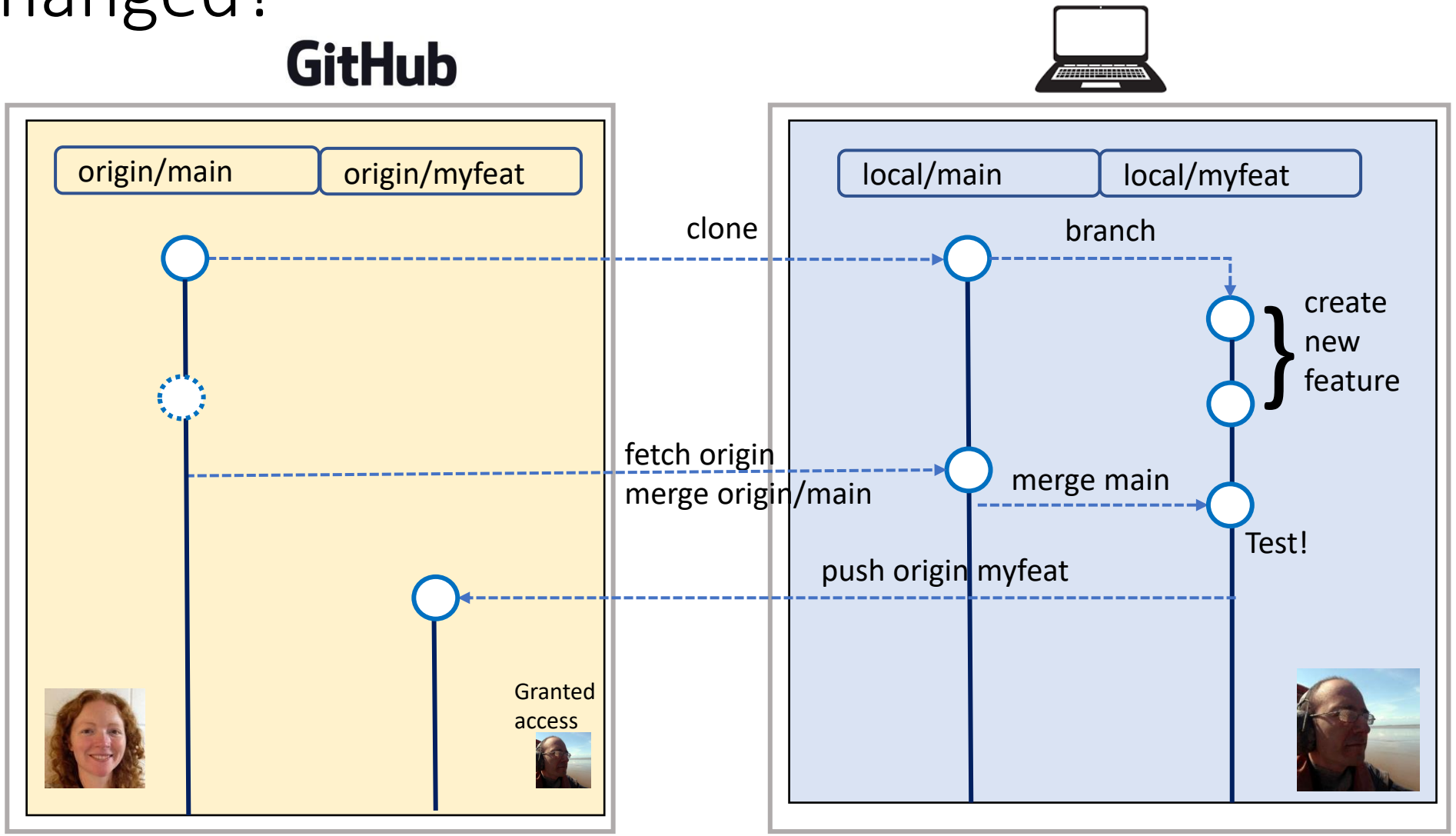
1. Make a local repo
2. Create branch
3. Modify
4. Share branch

```
$ git clone https://github.com/jenniferagraham/jag_hello_world.git
Cloning into 'jag_hello_world'...
$ cd jag_hello_world/
$ git branch fixreadme
$ git checkout fixreadme
$ git branch
* fixreadme
  main
$ notepad README.md
$ git commit -am "Update name of UEA cluster"
[main 1ba2db8] Update name of UEA cluster
1 file changed, 1 insertion(+), 1 deletion(-)
$ git push origin
Enumerating objects: 5, done.
(…)
* [new branch]      fixreadme -> fixreadme
```

What if main branch has changed?



Shared access workflow: what if origin/main has changed?



Shared access workflow: merging with changes on origin

3. Modify

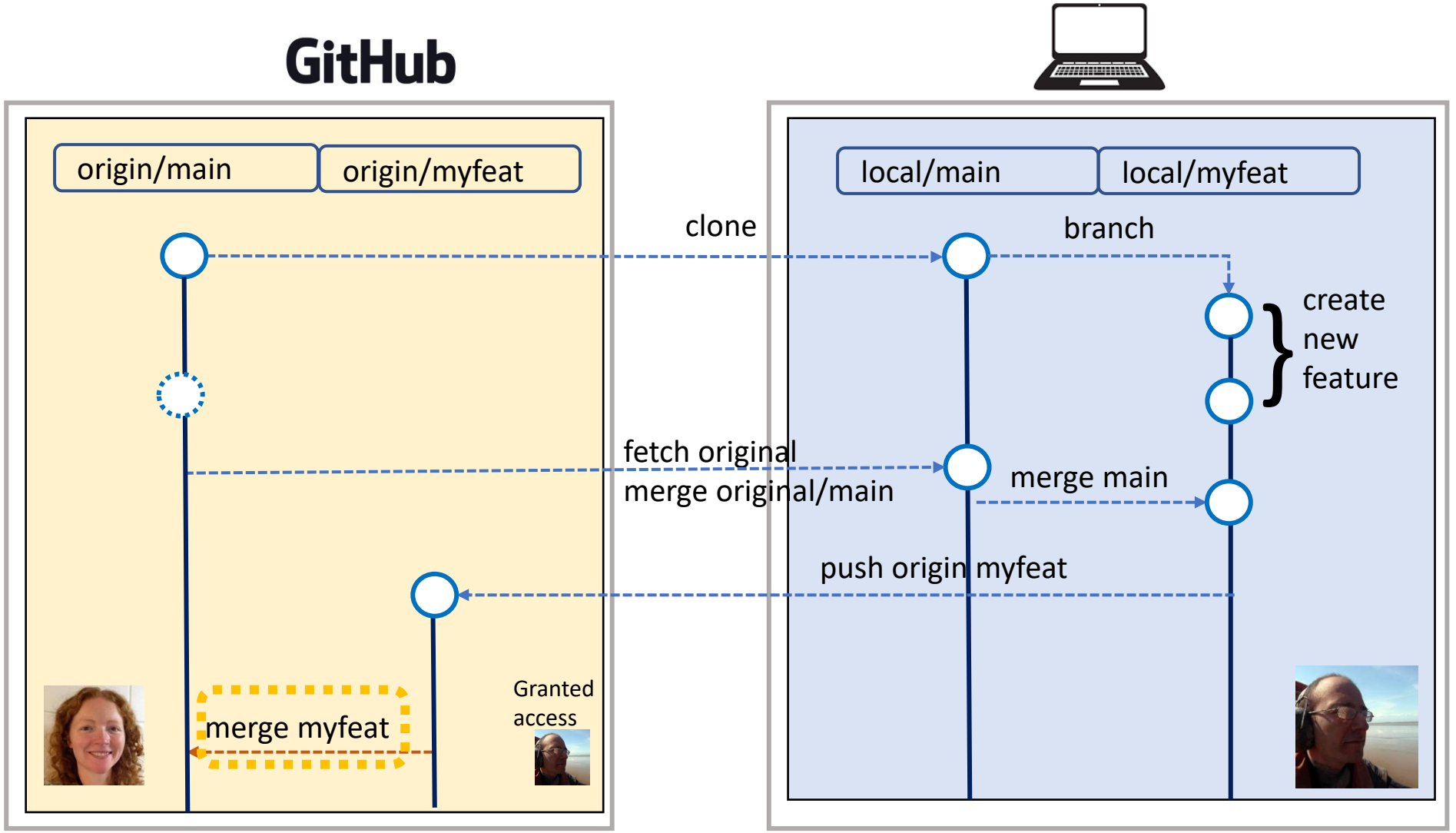
4. update local main

5. merge to new branch

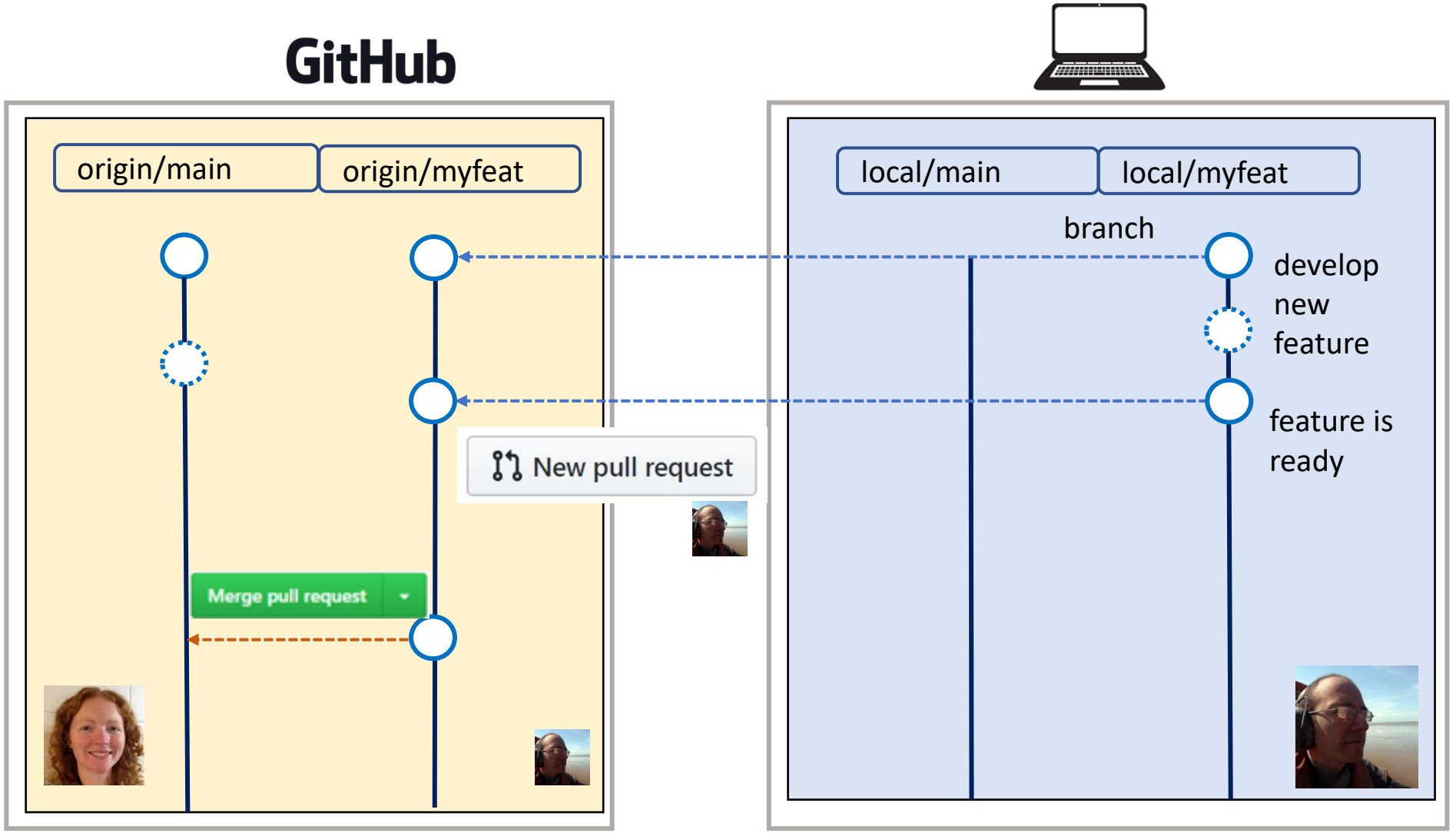
6. Share branch

```
$ git commit -am "Update name of UEA cluster"
[main 1ba2db8] Update name of UEA cluster
1 file changed, 1 insertion(+), 1 deletion(-)
$ git pull
$ git pull
(…)
$ git merge main
Merge made by the 'recursive' strategy.
README.md | 4 +++-
fileC.txt | 1 +
2 files changed, 4 insertions(+), 1 deletion(-)
create mode 100644 fileC.txt
$ git push origin
Enumerating objects: 5, done.
(…)
* [new branch]      fixreadme -> fixreadme
```

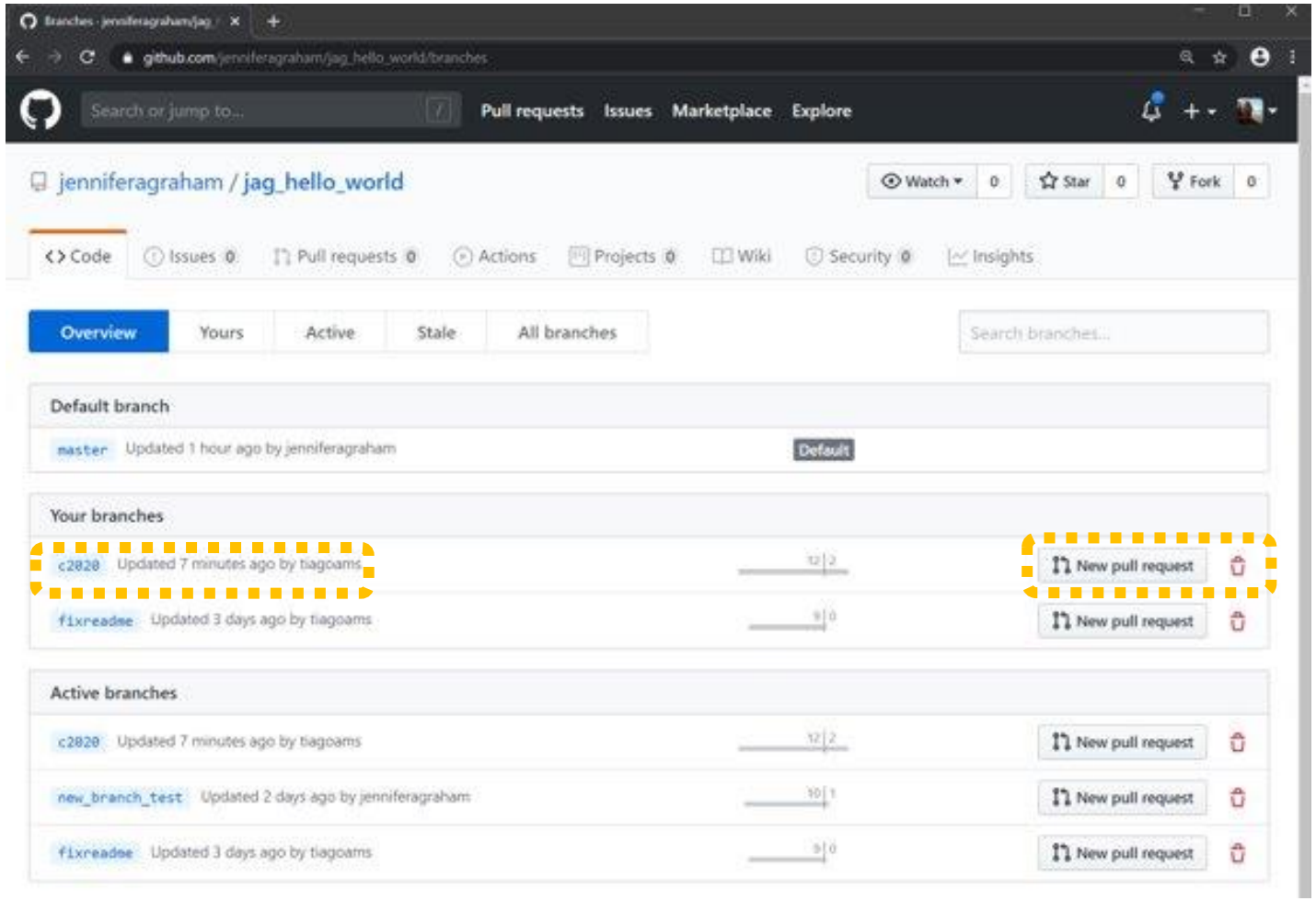
Shared access workflow: how is the merging done?



Recommended option: create a pull request



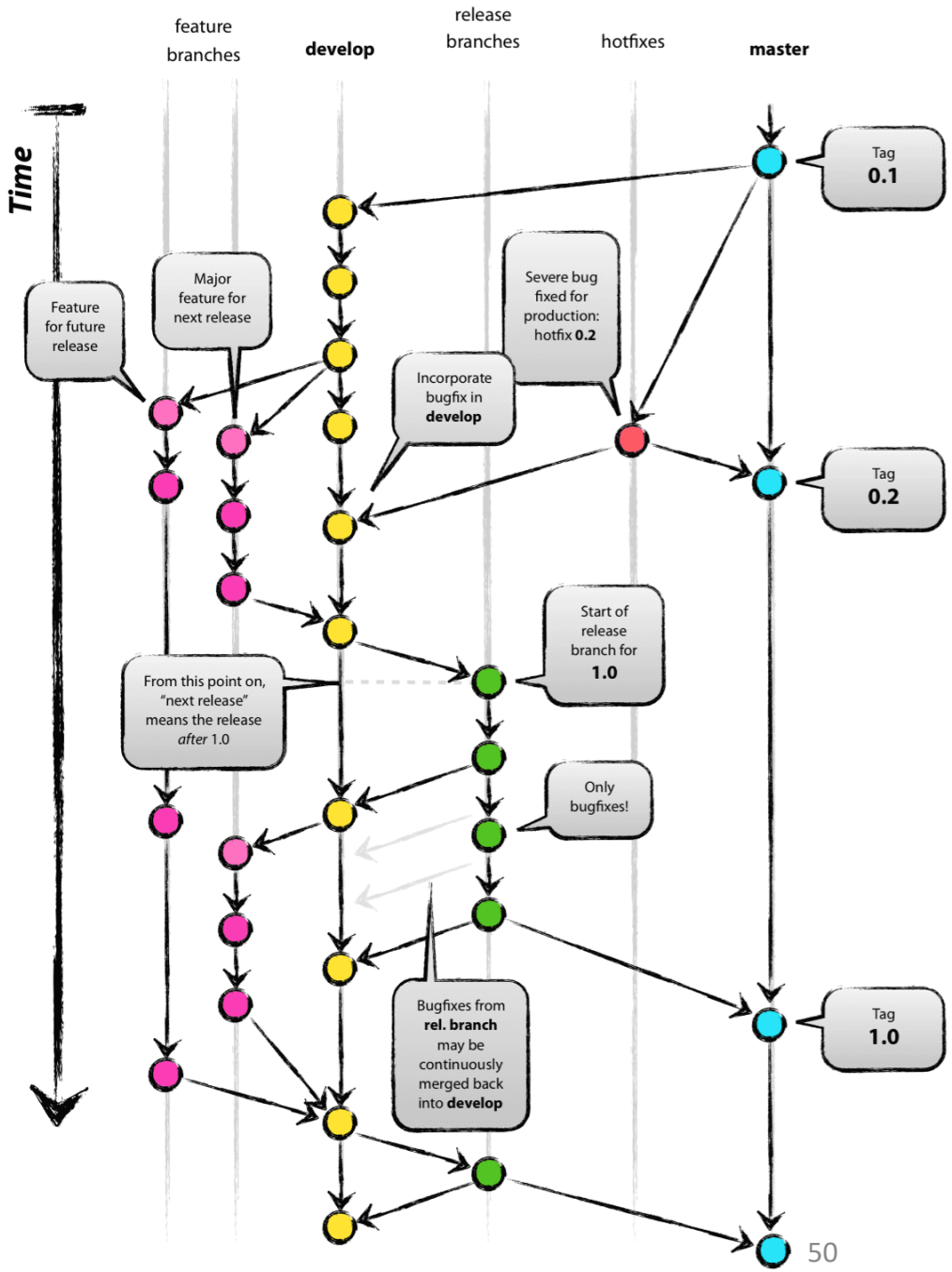
Shared access workflow: the pull request can be inspected and accepted/rejected on GitHub



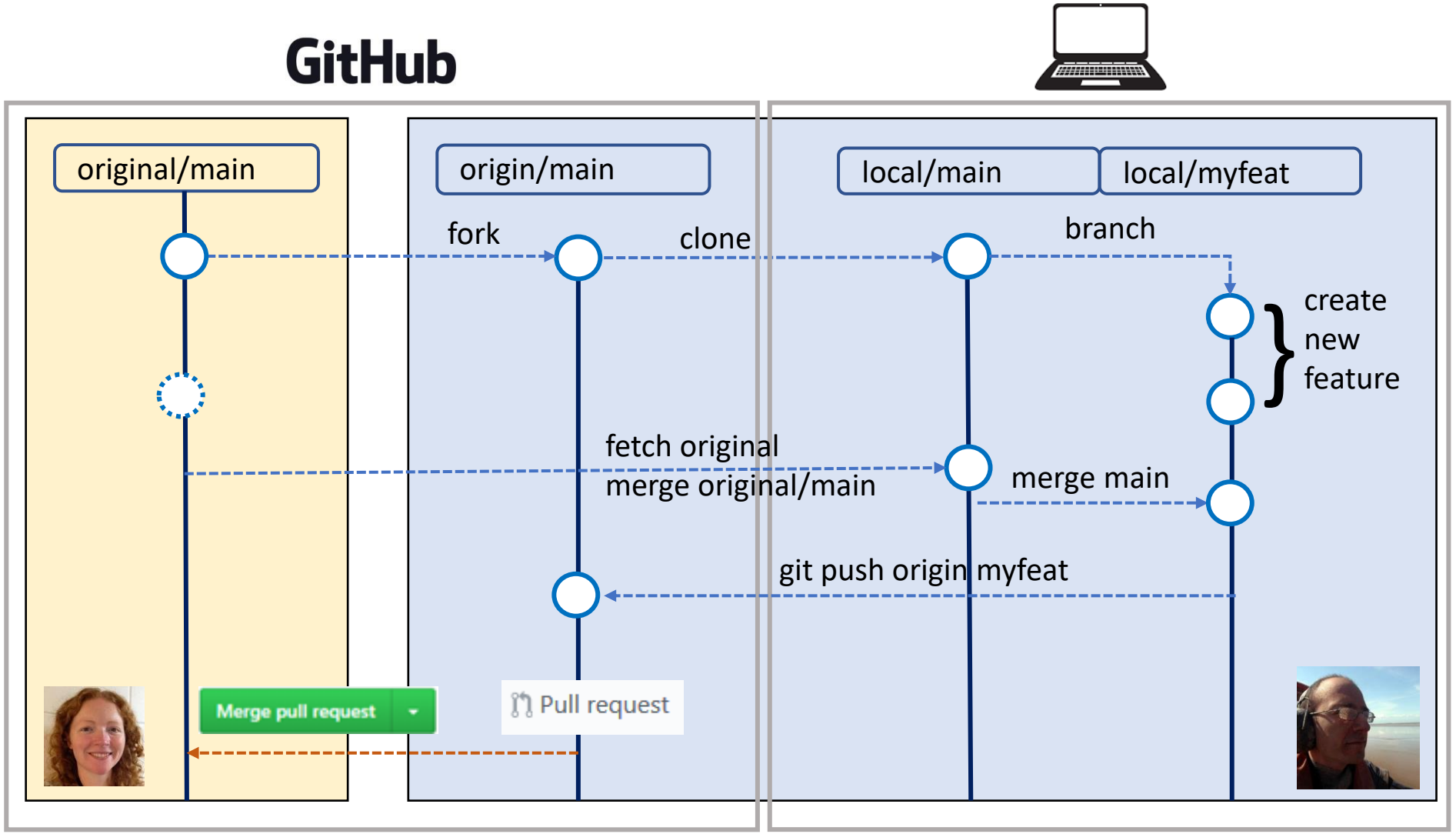
Shared access workflow: Gitflow

https://nvie.com/posts/a-successful-git-branching-model/

- Many software developers use strict branching systems to ensure “the main is always shippable”
  - i.e. changes always tested on development branch, so they don’t break the code!

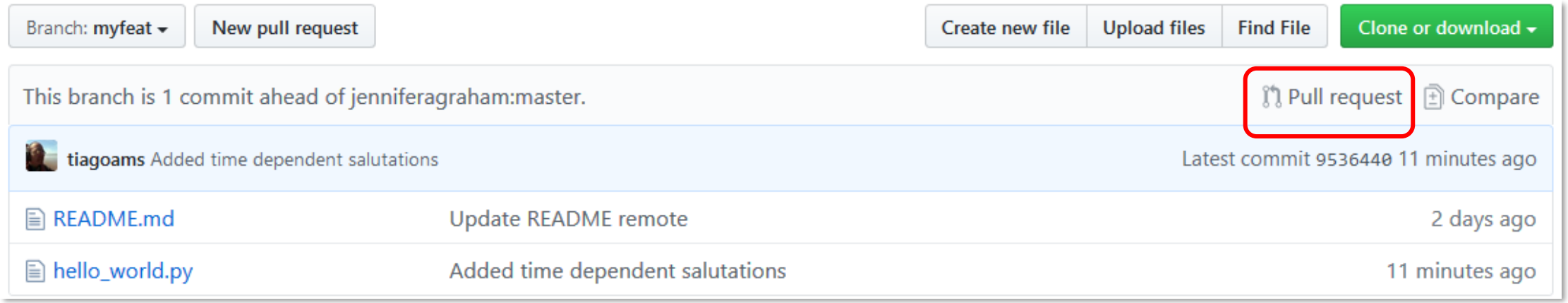


Fork workflow: no privileged access required

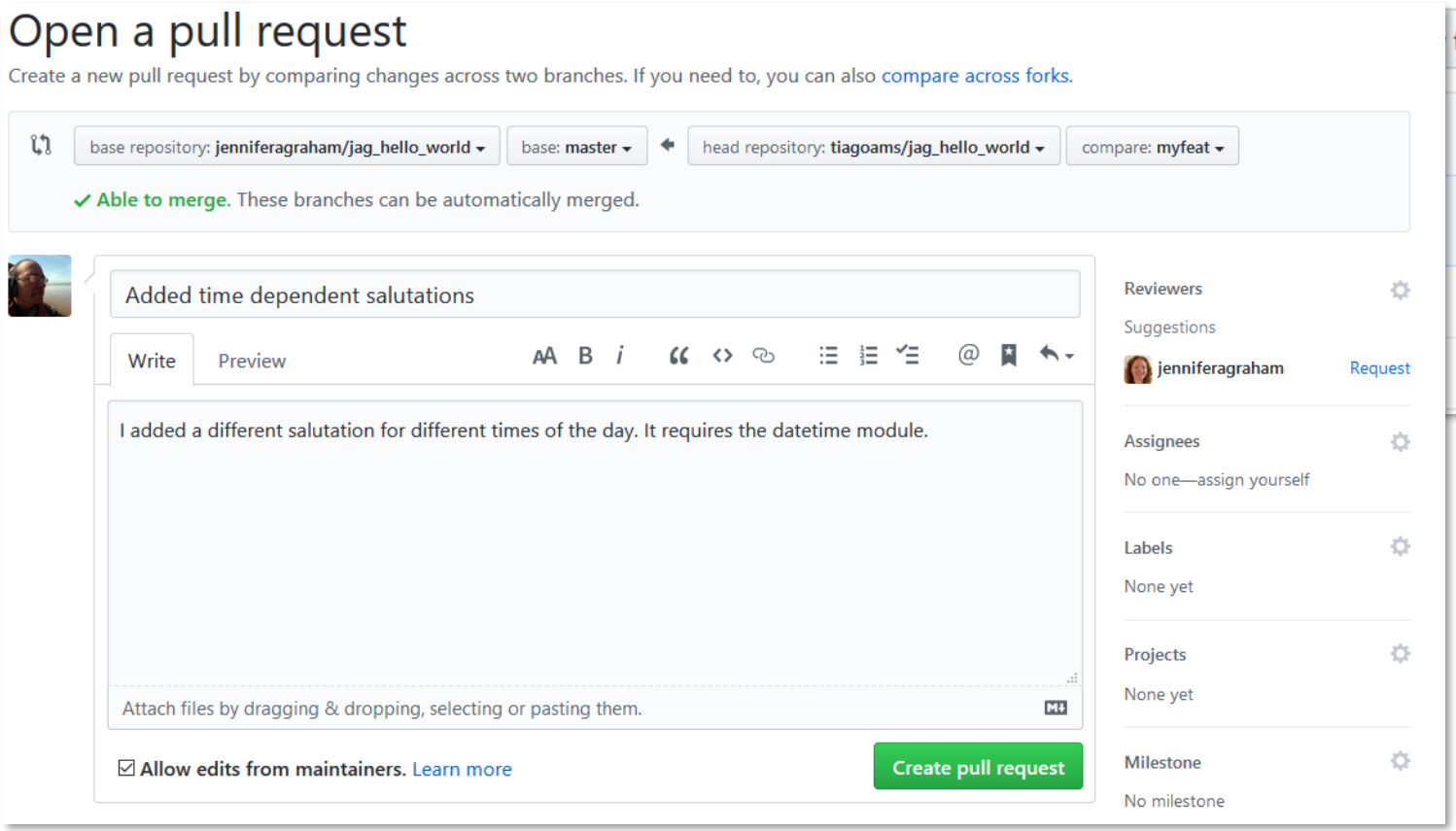


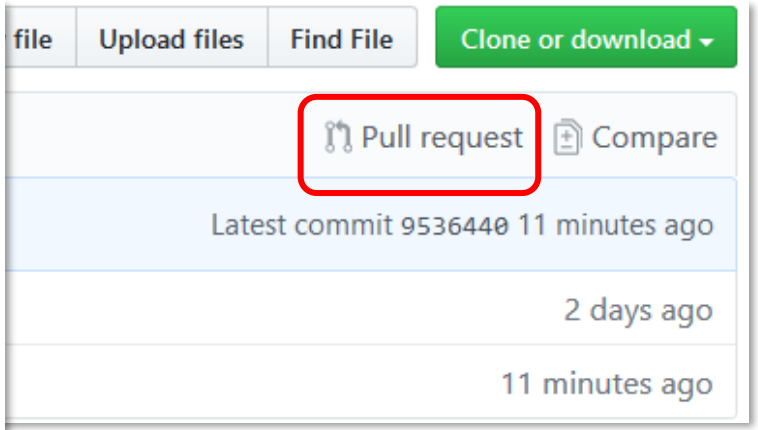
Pull requests: the contributor vs owners view

No write access: create a pull request

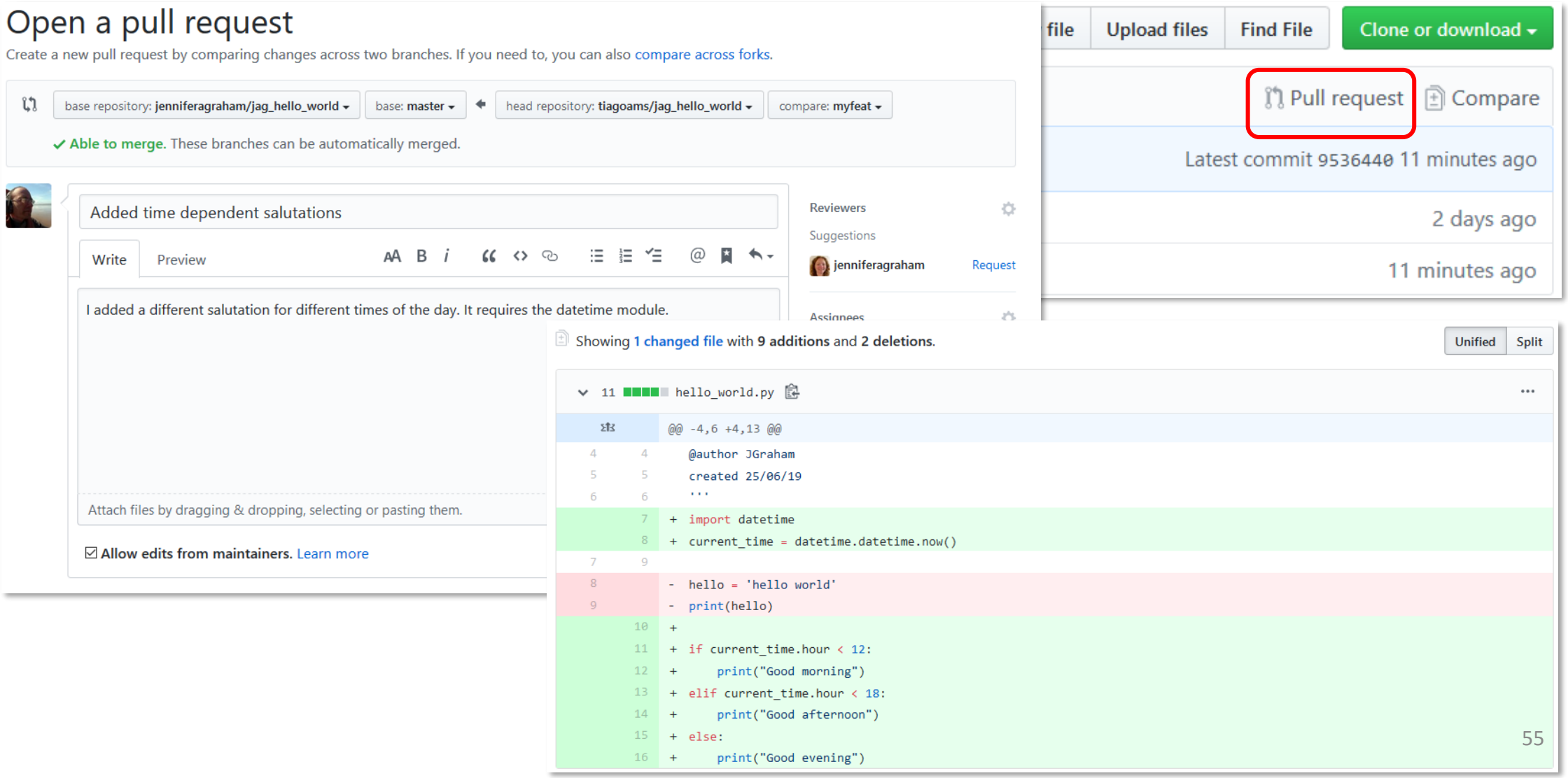


No write access: create a pull request

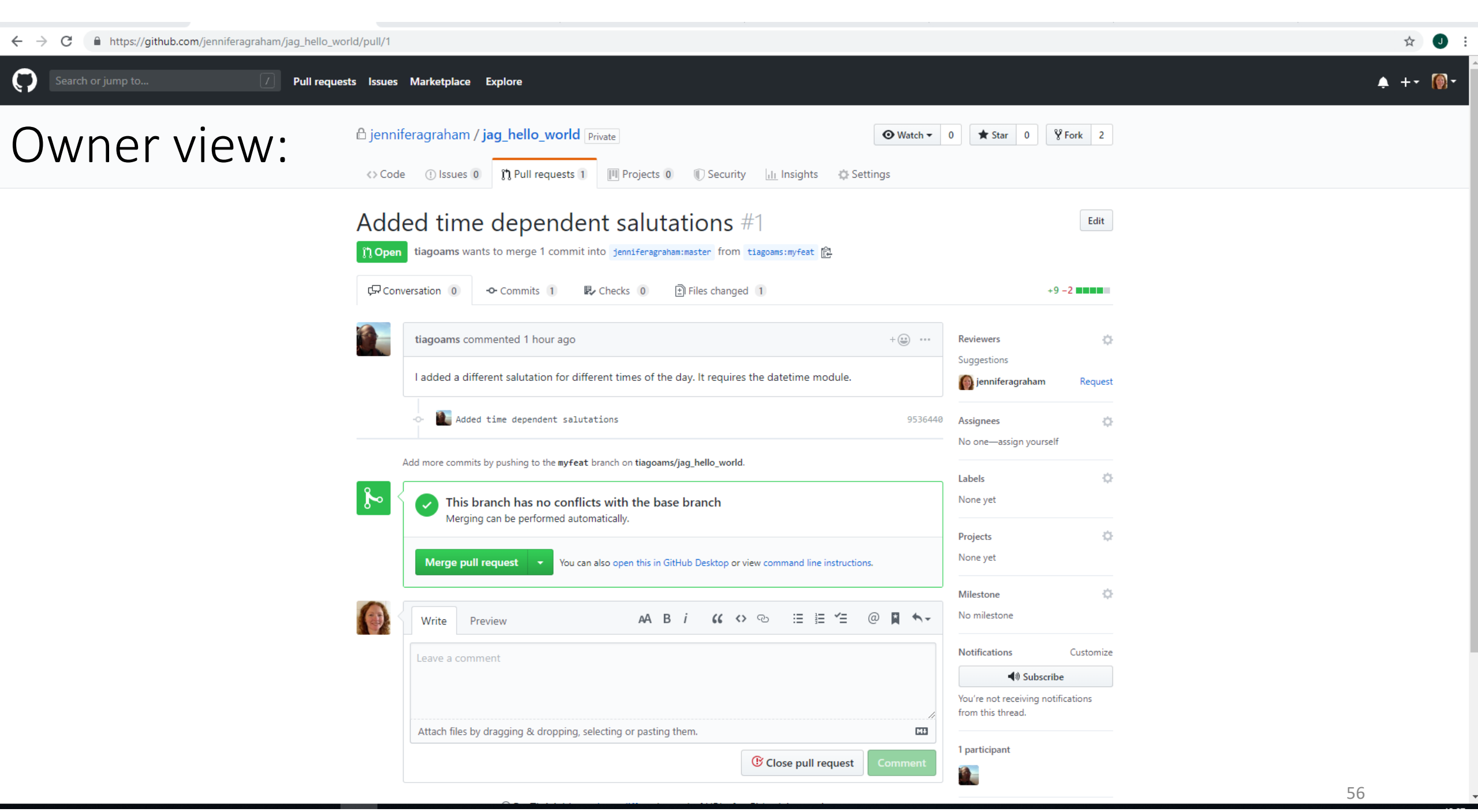




No write access: create a pull request



Owner view:


https://github.com/jenniferagraham/jag_hello_world/pull/1
Search or jump to...
Pull requests
Issues
Marketplace
Explore
jenniferagraham / jag_hello_world
Private
Watch 0
Star 0
Fork 2
Code
Issues 0
Pull requests 1
Projects 0
Security
Insights
Settings
Added time dependent salutations #1
Edit
Open
tiagoams wants to merge 1 commit into jenniferagraham:master from tiagoams:myfeat
Conversation 0
Commits 1
Checks 0
Files changed 1
+9 −2
tiagoams commented 1 hour ago
I added a different salutation for different times of the day. It requires the datetime module.
Added time dependent salutations
9536440
Add more commits by pushing to the myfeat branch on tiagoams/jag_hello_world.
This branch has no conflicts with the base branch
Merging can be performed automatically.
Merge pull request
You can also open this in GitHub Desktop or view command line instructions.
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Close pull request
Comment
Reviewers
Suggestions
jenniferagraham
Request
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Notifications
Customize
Subscribe
You're not receiving notifications from this thread.
1 participant

https://github.com/jenniferagraham/jag_hello_world/pull/1/commits/9536440b9aa74386a5fa9fbf7814ed1363ba24de

jenniferagraham / jag_hello_world Private

Watch 0 Star 0 Fork 2

Code Issues 0 Pull requests 1 Projects 0 Security Insights Settings

Added time dependent salutations #1

Edit

Open tiagoams wants to merge 1 commit into jenniferagraham:master from tiagoams:myfeat

Conversation 0 Commits 1 Checks 0 Files changed 1 +9 −2

Changes from all commits File filter... Jump to...

Review changes

Write Preview

This is a nice change!

Attach files by dragging & dropping, selecting or pasting them.

- Comment
  Submit general feedback without explicit approval.
- Approve
  Submit feedback and approve merging these changes.
- Request changes
  Submit feedback that must be addressed before merging.

Submit review

Added time dependent salutations

tiagoams committed 1 hour ago

11 hello_world.py

```
@@ -4,6 +4,13 @@
 @author JGraham
 created 25/06/19
 '''
+import datetime
+current_time = datetime.dat

-hello = 'hello world'
-print(hello)
+
+if current_time.hour < 12:
+    print("Good morning")
+elif current_time.hour < 18:
+    print("Good afternoon")
+else:
+    print("Good evening")
```

ProTip! Use n and p to navigate between commits in a pull request.

© 2019 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub Pricing API Training Blog About

https://github.com/jenniferagraham/jag_hello_world/pull/1#pullrequestreview-258371608

Open **Added time dependent salutations** #1

tiagoams wants to merge 1 commit into `jenniferagraham:master` from `tiagoams:myfeat`

Added time dependent salutations 9536440

jenniferagraham approved these changes now — View changes

jenniferagraham left a comment

This is a nice change!

Add more commits by pushing to the **myfeat** branch on **tiagoams/jag_hello_world**.

Changes approved — Show all reviewers

1 approving review Learn more.

1 approval

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request — You can also open this in GitHub Desktop or view command line instructions.

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close pull request | Comment

ProTip! Add .patch or .diff to the end of URLs for Git's plaintext views.

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications — Customize

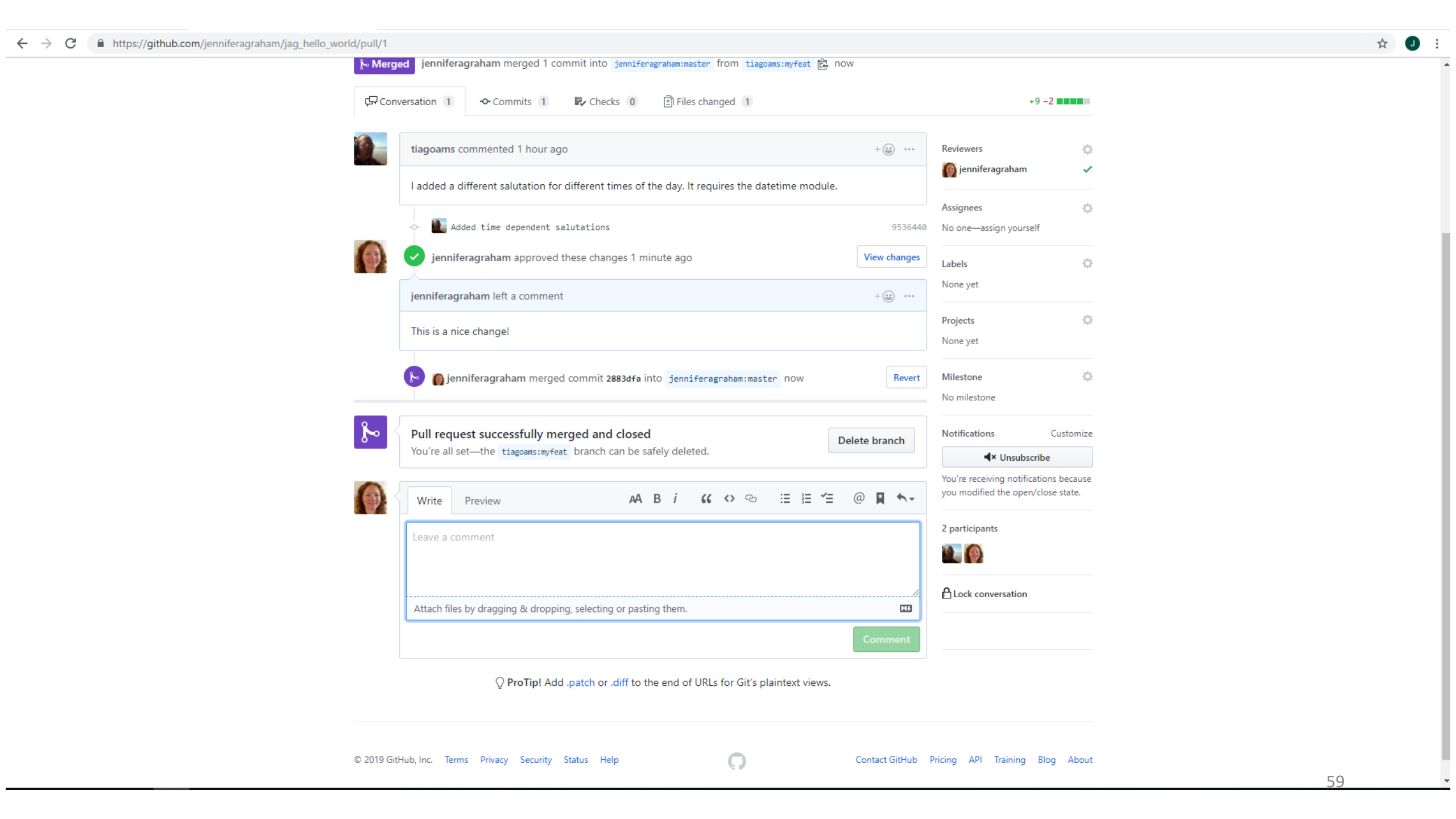
Unsubscribe

You're receiving notifications because you commented.

2 participants

Lock conversation

© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About


https://github.com/jenniferagraham/jag_hello_world/pull/1
Merged
jenniferagraham merged 1 commit into jenniferagraham:master from tiagoams:myfeat now
Conversation 1
Commits 1
Checks 0
Files changed 1
+9 −2
tiagoams commented 1 hour ago
I added a different salutation for different times of the day. It requires the datetime module.
Added time dependent salutations
9536440
jenniferagraham approved these changes 1 minute ago
View changes
jenniferagraham left a comment
This is a nice change!
jenniferagraham merged commit 2883dfa into jenniferagraham:master now
Revert
Pull request successfully merged and closed
You're all set—the tiagoams:myfeat branch can be safely deleted.
Delete branch
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Comment
ProTip! Add .patch or .diff to the end of URLs for Git's plaintext views.
Reviewers
jenniferagraham
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Notifications
Customize
Unsubscribe
You're receiving notifications because you modified the open/close state.
2 participants
Lock conversation
© 2019 GitHub, Inc.
Terms
Privacy
Security
Status
Help
Contact GitHub
Pricing
API
Training
Blog
About

Your turn:

Option 1 – Shared repository

- Pair up and find each other's test repositories
- Add write acess to colleague on test repository
- Clone, create branch, make a commit and push changes
- Check if changes were made in the original repo whilst you were editing (there might!) 😈
  - Pull changes and merge to new branch
  - Push to shared repo

Option 2 – Fork Jenny's jag_hello_world

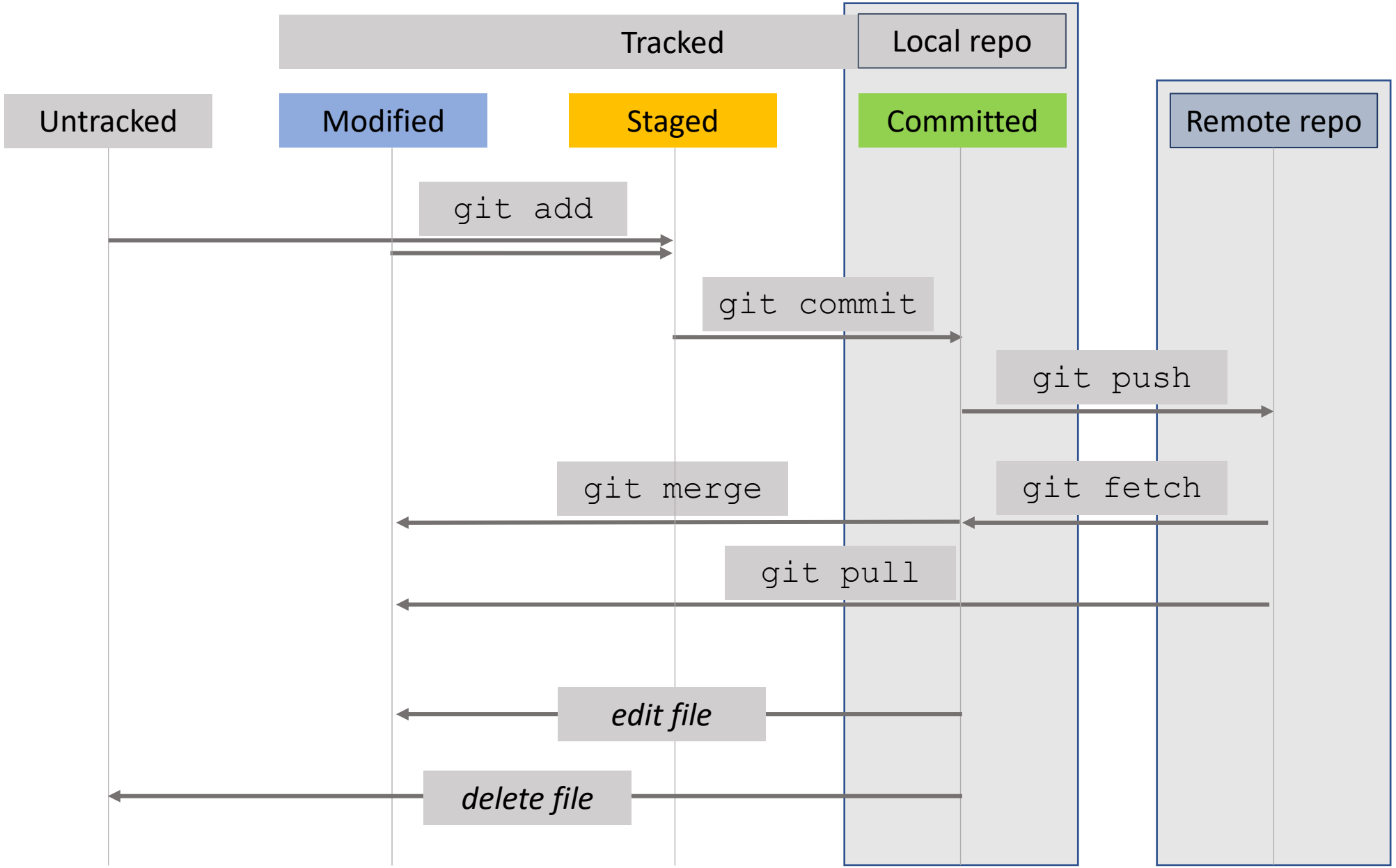
https://github.com/jenniferagraham/jag_hello_world.git

- Clone your own forked copy to your PC
- Create branch, make a commit and push to your remote
- Make a pull request on GitHub

Please provide feedback!

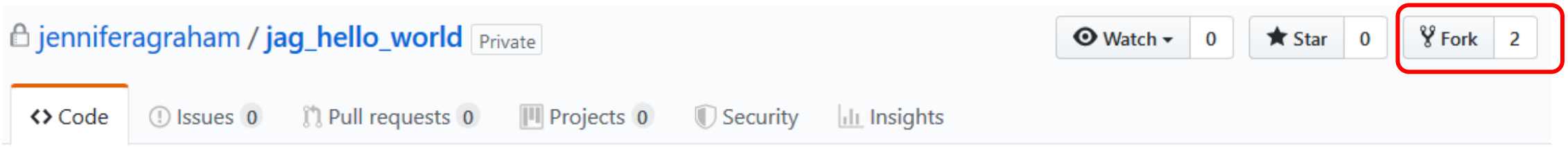
Anonymous comments can be provided here:

https://forms.gle/UTXXAwzq1qEFFx829

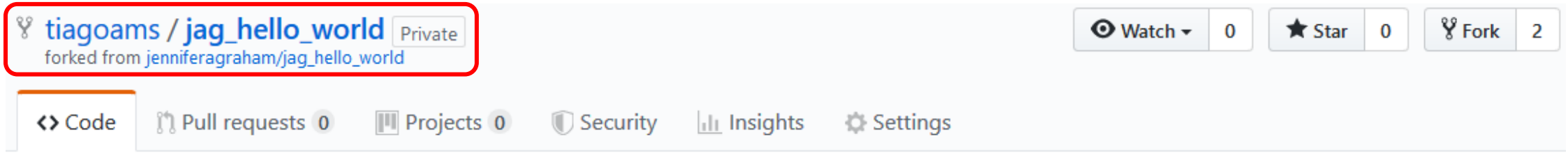

Tracked
Local repo
Untracked
Modified
Staged
Committed
Remote repo
git add
git commit
git push
git merge
git fetch
git pull
edit file
delete file

Fork workflow: step by step…

1. Fork other user's repo on GitHub

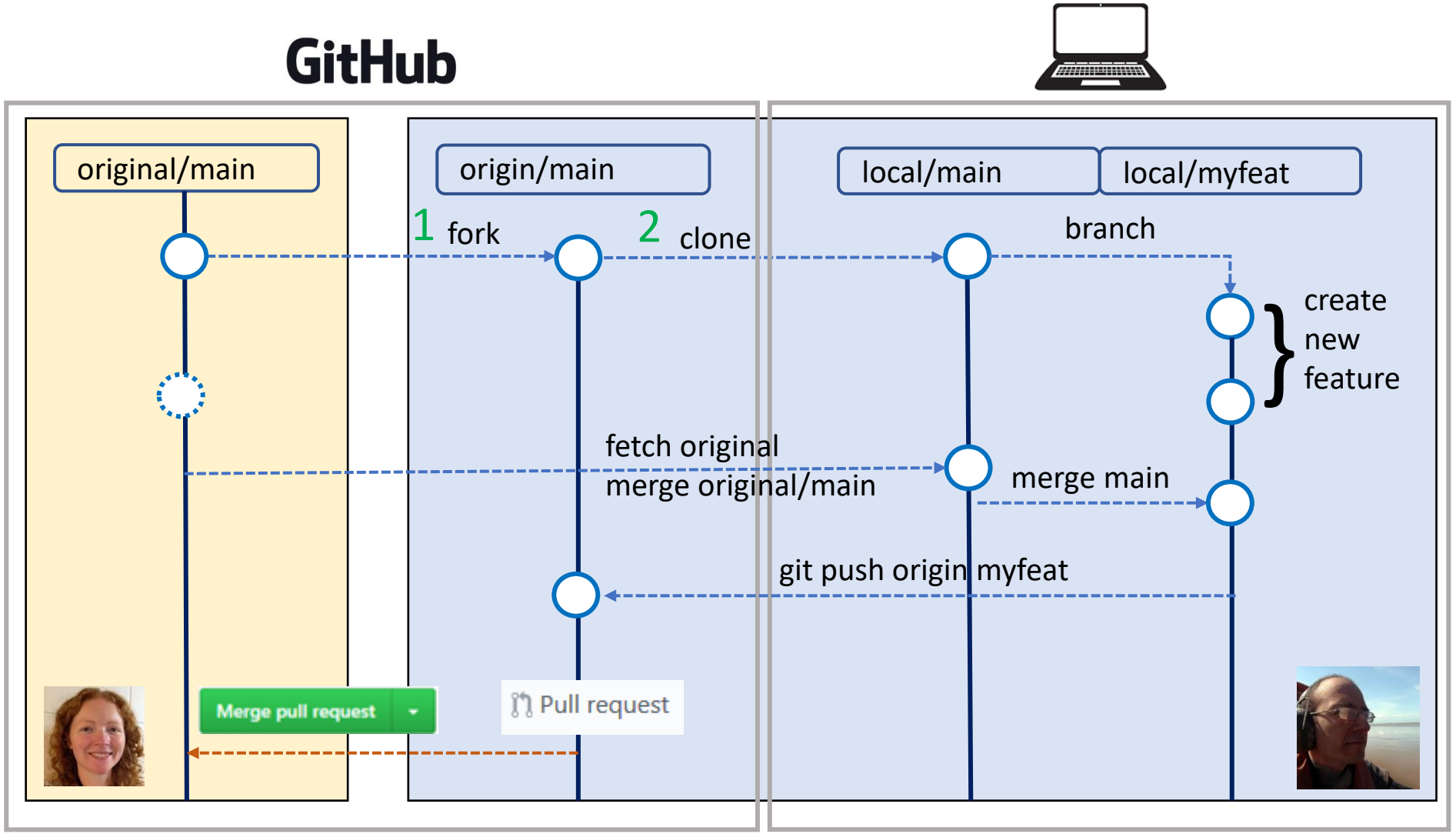


This is how it looks in your account




```
$ git clone git@github.com:tiagoams/jag_hello_world.git
```

Fork workflow: no privileged access required



Fork workflow: working locally

hello_world.py

```
'''
My first code.

@author JGraham
created 25/06/19
'''

hello = 'hello world'
print(hello)
```

3. Create new local branch

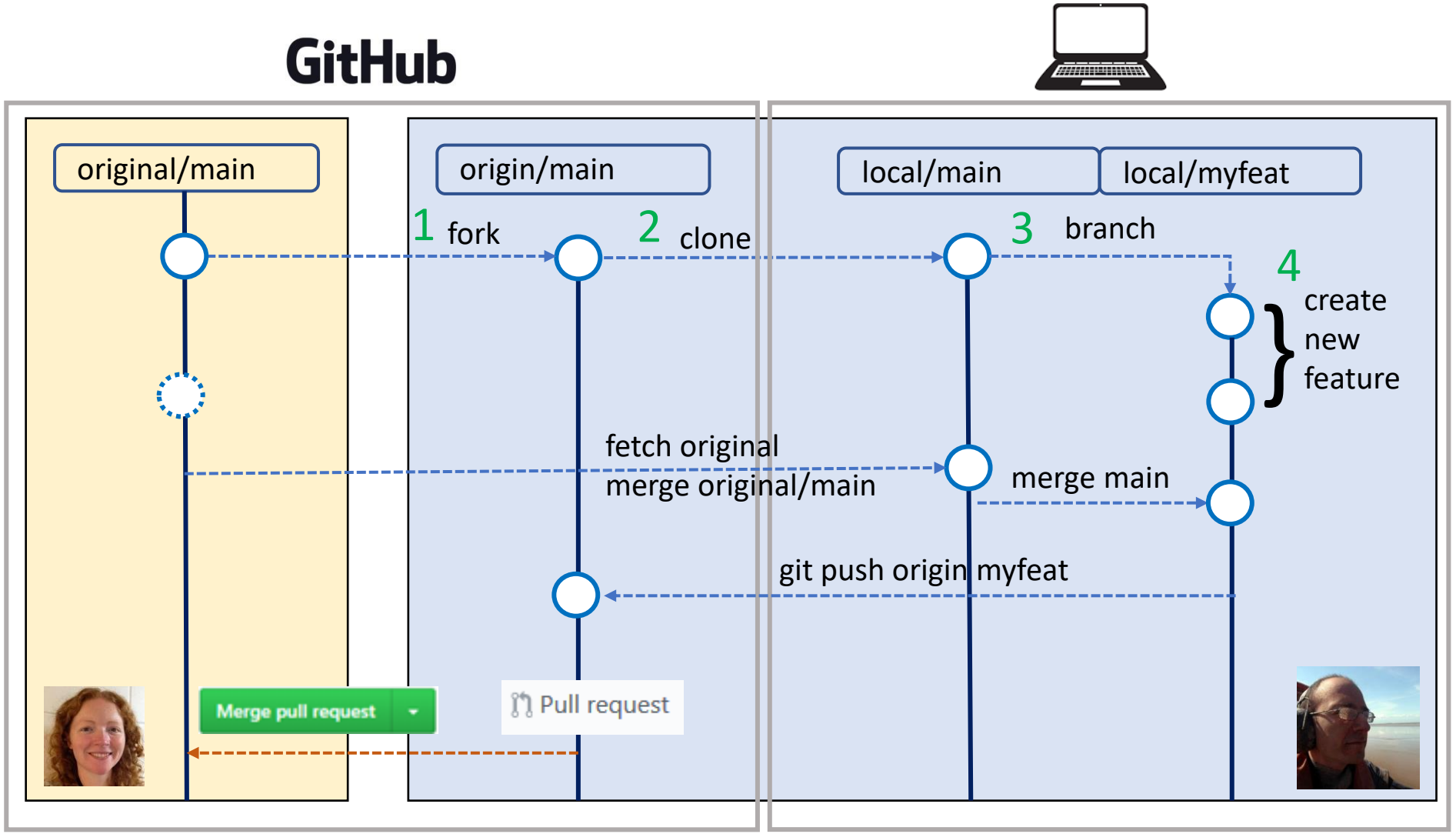
`checkout -b <branch>` = `branch <branch>` `checkout <branch>`

4. Modify and commit

`commit -a -m <msg>` = `add *` `commit -m <msg>`

```
$ git checkout -b myfeat

Switched to a new branch 'myfeat'
$ cat hello_world.py
(...)
import datetime
current_time = datetime.datetime.now()


if current_time.hour < 12:
    print("Good morning")
elif current_time.hour < 18:
    print("Good afternoon")
else:
    print("Good evening")

$ git commit -am "Added time dependent salutations"

[myfeat 9536440] Added time dependent salutations

 1 file changed, 9 insertions(+), 2 deletions(-)
```

Fork workflow: no privileged access required



Fork workflow: check for changes since fork

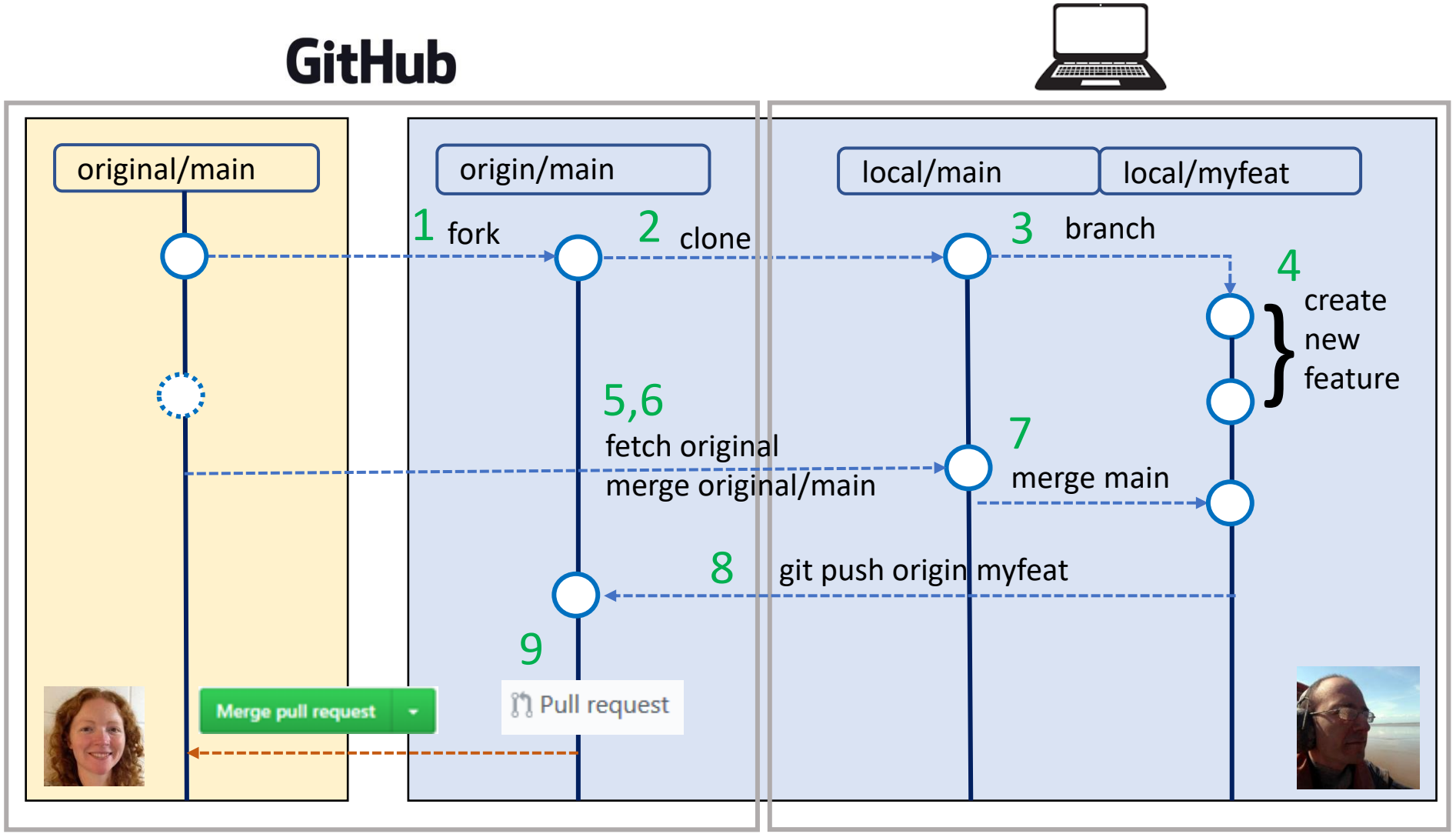
5. Add a new remote *original*

6. Fetch and merge to local main branch

7. Merge/rebase to *myfeat* branch

```
$ git remote add original git@github.com:jenniferagraham/jag_hello_world.git
$ git remote -v
origin  git@github.com:tiagoams/jag_hello_world.git (fetch)
origin  git@github.com:tiagoams/jag_hello_world.git (push)
original        git@github.com:jenniferagraham/jag_hello_world.git (fetch)
original        git@github.com:jenniferagraham/jag_hello_world.git (push)
$ git fetch original
From github.com:jenniferagraham/jag_hello_world
 * [new branch]      main     -> original/main
$ git checkout main
Switched to branch ‘main’
$ git merge original/main
Already up to date.
$ git checkout myfeat
Switched to branch 'myfeat'
$ git rebase main
Current branch myfeat is up to date.
$
```

Fork workflow: no privileged access required

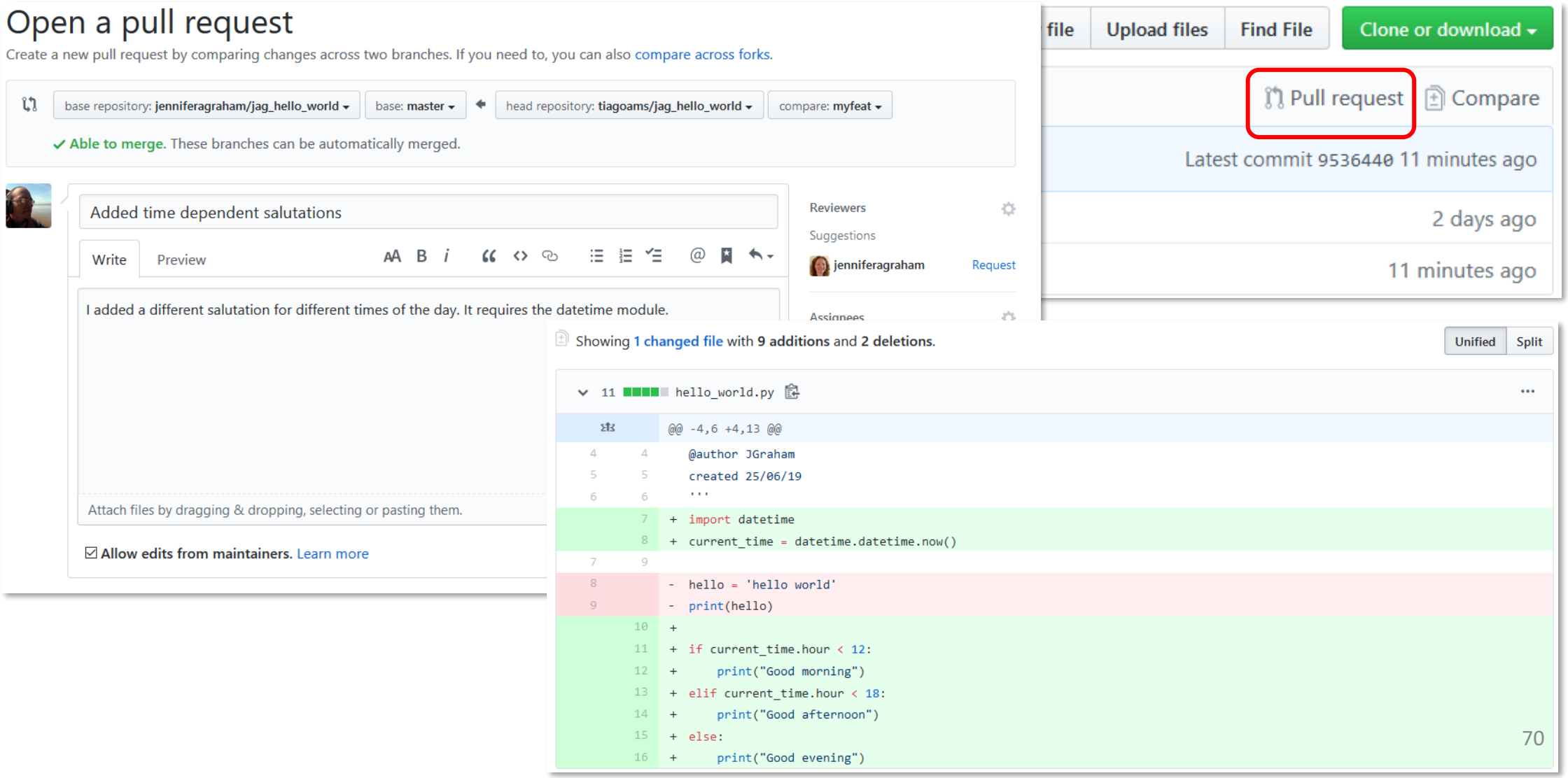


Fork workflow: push to *origin*

8. Push changes to your GitHub repo

```
$ git push origin myfeat
Counting objects: 3, done.
Delta compression using up to 12 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 447 bytes | 223.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'myfeat' on GitHub by visiting:
remote:
https://github.com/tiagoams/jag_hello_world/pull/new/myfeat
remote:
To github.com:tiagoams/jag_hello_world.git
 * [new branch]      myfeat -> myfeat
$
```

Fork workflow: 9. create a pull request



Fork workflow: the owners view

https://github.com/jenniferagraham/jag_hello_world/pull/1

Search or jump to... / Pull requests Issues Marketplace Explore

jenniferagraham / jag_hello_world Private

Watch 0 Star 0 Fork 2

Code Issues 0 Pull requests 1 Projects 0 Security Insights Settings

Added time dependent salutations #1

Edit

Open tiagoams wants to merge 1 commit into `jenniferagraham:master` from `tiagoams:myfeat`

Conversation 0 Commits 1 Checks 0 Files changed 1 +9 −2

tiagoams commented 1 hour ago

I added a different salutation for different times of the day. It requires the datetime module.

Added time dependent salutations 9536440

Add more commits by pushing to the **myfeat** branch on **tiagoams/jag_hello_world**.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close pull request Comment

Reviewers
Suggestions
jenniferagraham Request

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications Customize
Subscribe
You're not receiving notifications from this thread.

1 participant

https://github.com/jenniferagraham/jag_hello_world/pull/1/commits/9536440b9aa74386a5fa9fbf7814ed1363ba24de

jenniferagraham / jag_hello_world Private

Watch 0 | Star 0 | Fork 2

Code | Issues 0 | Pull requests 1 | Projects 0 | Security | Insights | Settings

Added time dependent salutations #1

Edit

Open tiagoams wants to merge 1 commit into `jenniferagraham:master` from `tiagoams:myfeat`

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

+9 −2

Changes from all commits | File filter... | Jump to...

Review changes

Added time dependent salutations

tiagoams committed 1 hour ago

11 hello_world.py

```
@@ -4,6 +4,13 @@
 @author JGraham
 created 25/06/19
 '''

-hello = 'hello world'
-print(hello)
```

```
 @author JGraham
 created 25/06/19
 '''
+import datetime
+current_time = datetime.dat

+
+if current_time.hour < 12:
+    print("Good morning")
+elif current_time.hour < 18:
+    print("Good afternoon")
+else:
+    print("Good evening")
```

Write | Preview

This is a nice change!

Attach files by dragging & dropping, selecting or pasting them.

- Comment
  Submit general feedback without explicit approval.
- Approve
  Submit feedback and approve merging these changes.
- Request changes
  Submit feedback that must be addressed before merging.

Submit review

ProTip! Use n and p to navigate between commits in a pull request.

© 2019 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub Pricing API Training Blog About

https://github.com/jenniferagraham/jag_hello_world/pull/1#pullrequestreview-258371608

Open **Added time dependent salutations** #1

tiagoams wants to merge 1 commit into `jenniferagraham:master` from `tiagoams:myfeat`

Added time dependent salutations 9536440

jenniferagraham approved these changes now — View changes

jenniferagraham left a comment

This is a nice change!

Add more commits by pushing to the **myfeat** branch on **tiagoams/jag_hello_world**.

Changes approved — Show all reviewers

1 approving review Learn more.

1 approval

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request — You can also open this in GitHub Desktop or view command line instructions.

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close pull request | Comment

ProTip! Add .patch or .diff to the end of URLs for Git's plaintext views.

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications — Customize

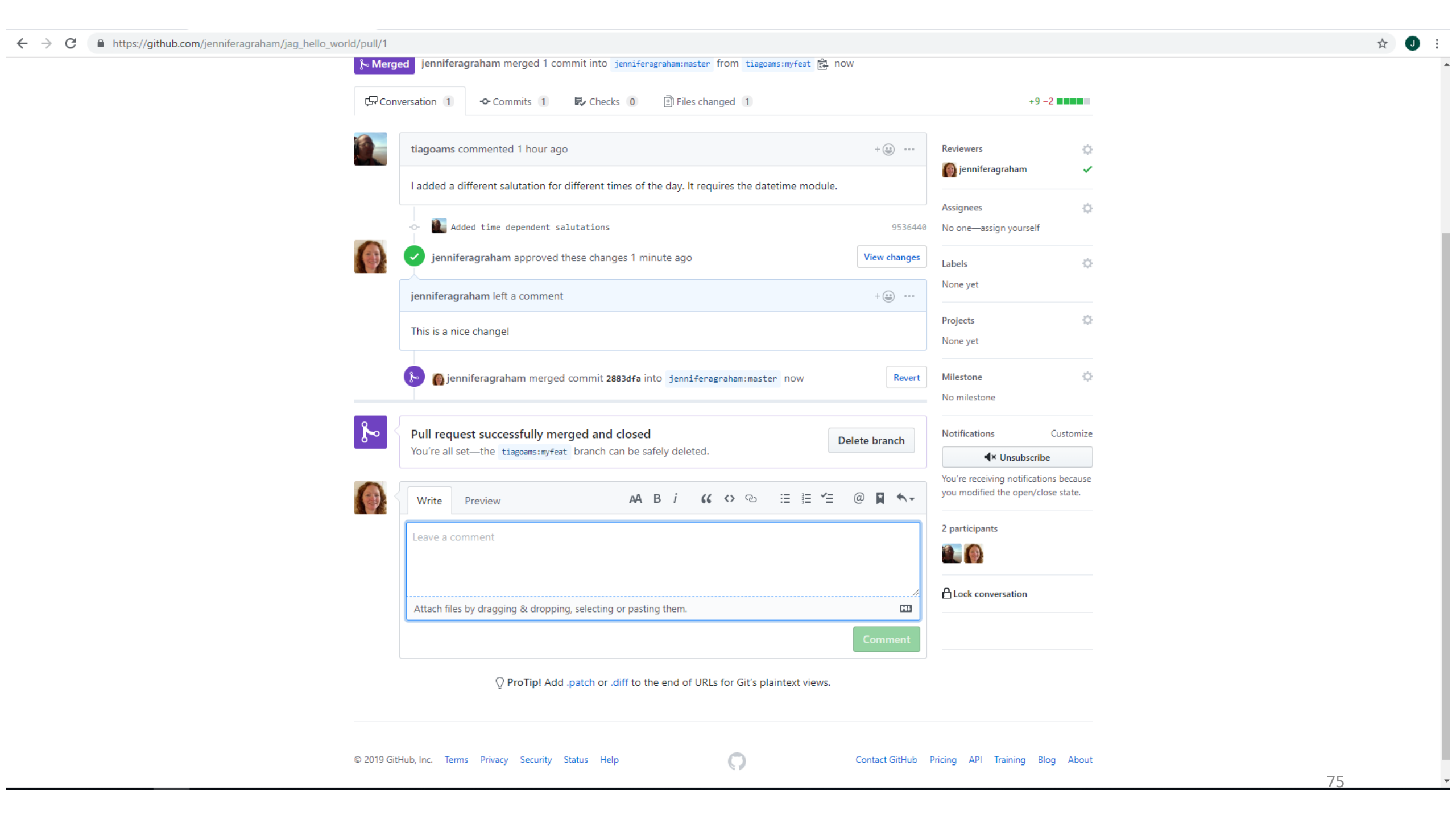
Unsubscribe

You're receiving notifications because you commented.

2 participants

Lock conversation

© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About


https://github.com/jenniferagraham/jag_hello_world/pull/1
Merged
jenniferagraham merged 1 commit into jenniferagraham:master from tiagoams:myfeat now
Conversation 1
Commits 1
Checks 0
Files changed 1
+9 −2
tiagoams commented 1 hour ago
I added a different salutation for different times of the day. It requires the datetime module.
Added time dependent salutations
9536440
jenniferagraham approved these changes 1 minute ago
View changes
jenniferagraham left a comment
This is a nice change!
jenniferagraham merged commit 2883dfa into jenniferagraham:master now
Revert
Pull request successfully merged and closed
You're all set—the tiagoams:myfeat branch can be safely deleted.
Delete branch
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Comment
ProTip! Add .patch or .diff to the end of URLs for Git's plaintext views.
Reviewers
jenniferagraham
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Notifications
Customize
Unsubscribe
You're receiving notifications because you modified the open/close state.
2 participants
Lock conversation
© 2019 GitHub, Inc.
Terms
Privacy
Security
Status
Help
Contact GitHub
Pricing
API
Training
Blog
About