Hello, Git!

Voula Troulaki

What is Git

- Git is a free Version Control System to keep track of changes to files and projects over time
- If you never used it, you really don't know what you miss
- Before-after

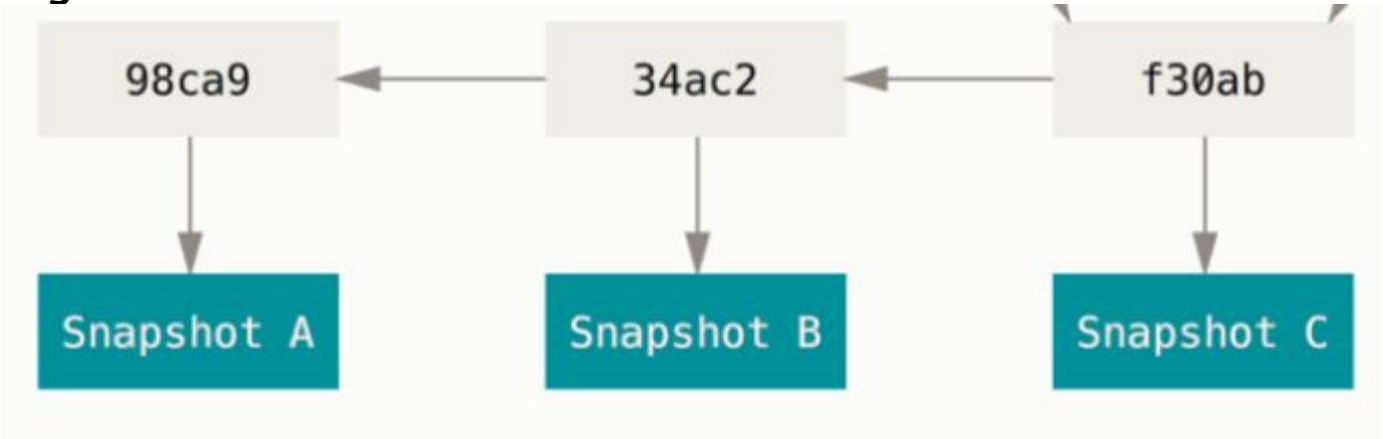
Why Git

- One project one repository
- Feature Branch Workflow
- Distributed Development
- Pull Requests/Code Reviews
- Community
- Faster Release Cycle
- Git hooks CI/CD

Commits

Repos contain a set o commit objects and references to commits (heads)

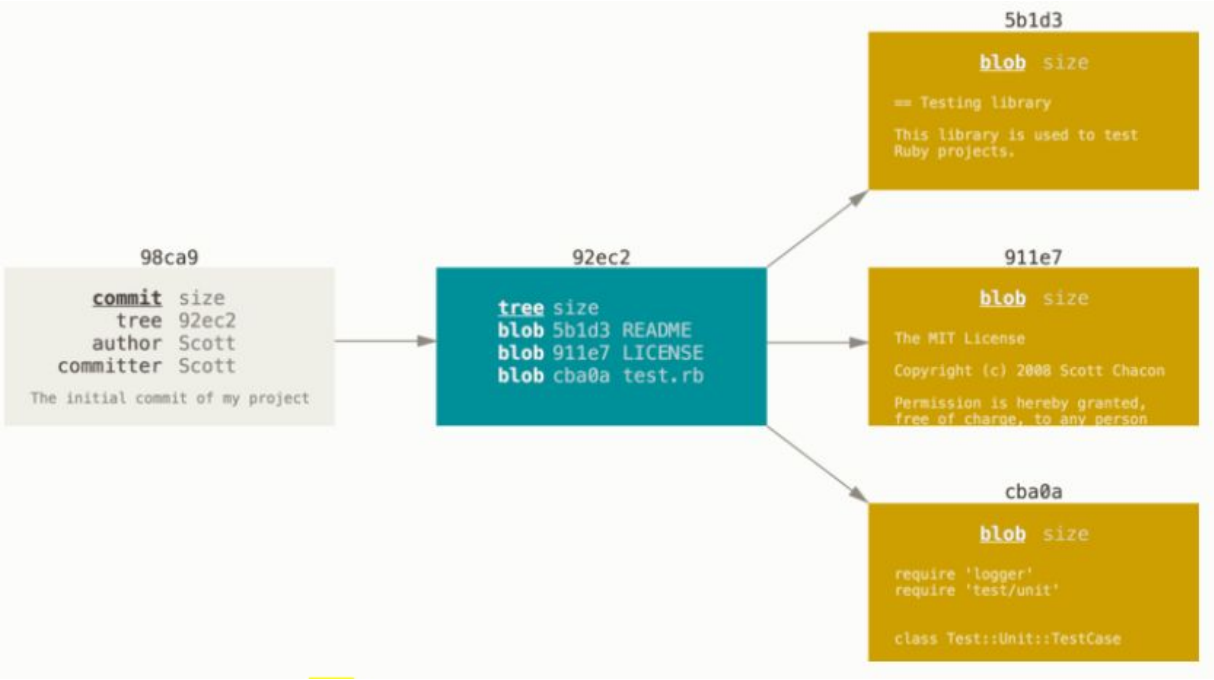
A commit object contains a pointer to the snapshot of the content you staged.



Snapshots

A snapshot contains

- Blobs (files)
- Trees (subdirectories) that list the contents of the directory and specify which file names are stored as which blobs



Git Branches - Pointers :)

`$ git branch testing`

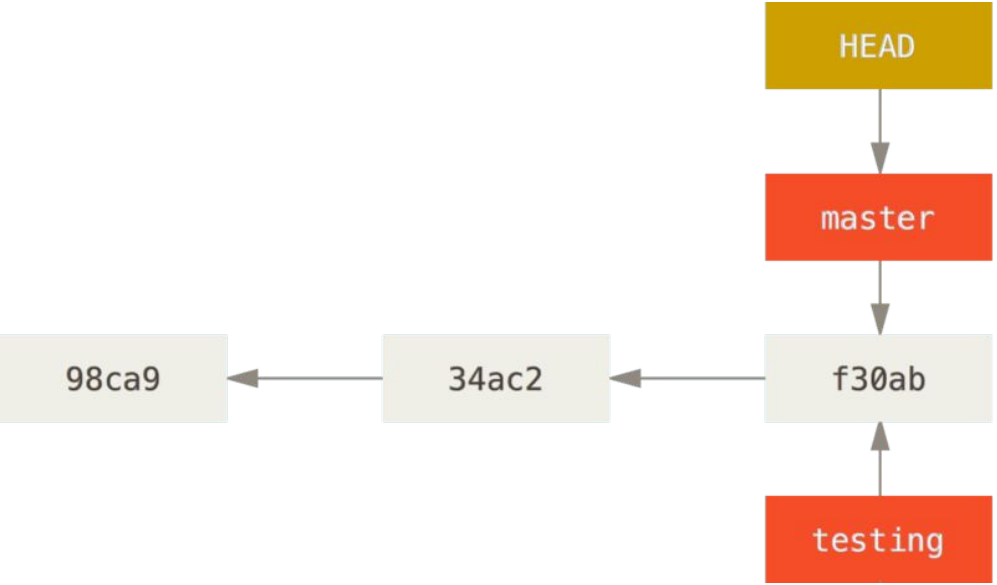
This creates a new pointer to the same commit you're currently on.

`$ git checkout testing`

This moves `HEAD` to point to the `testing` branch.

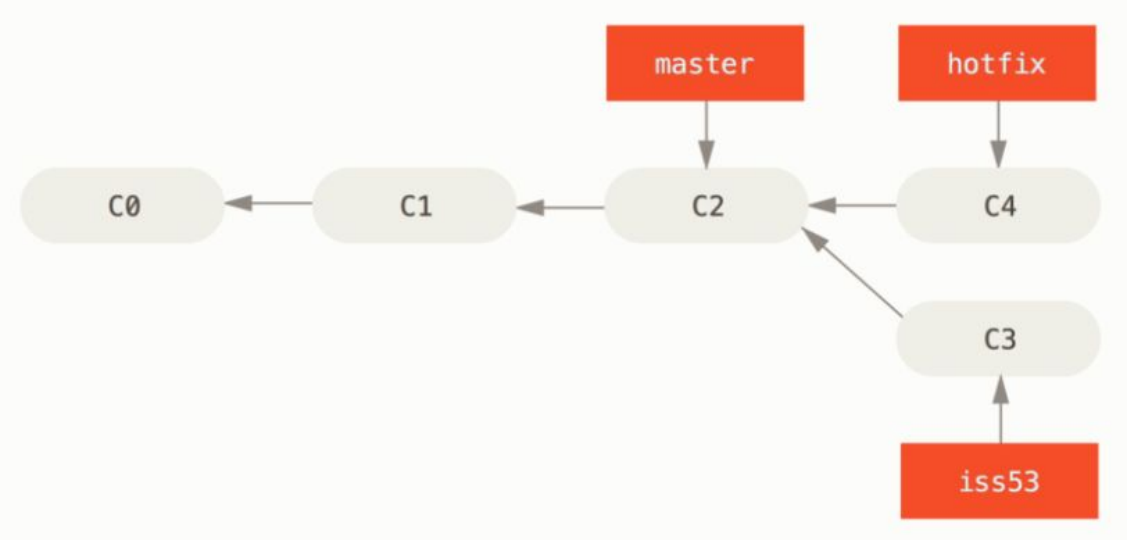
HEAD is a special pointer to the local branch you're currently on
The default branch name in Git is `master`

When you make a commit,
Git stores a commit object that contains a pointer to the snapshot of the content you staged.



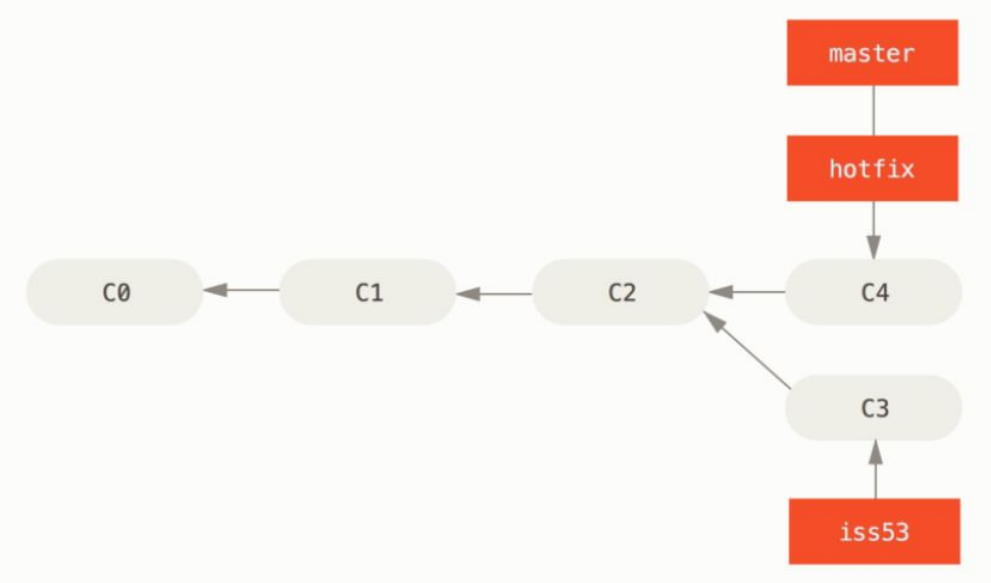
Basic Branching

```
$ git branch hotfix

$ git checkout -b iss53

$ git commit -m 'Create new footer [issue 53]'

$ git checkout hotfix

$ vim index.html

$ git commit -m 'Fix logo'
```



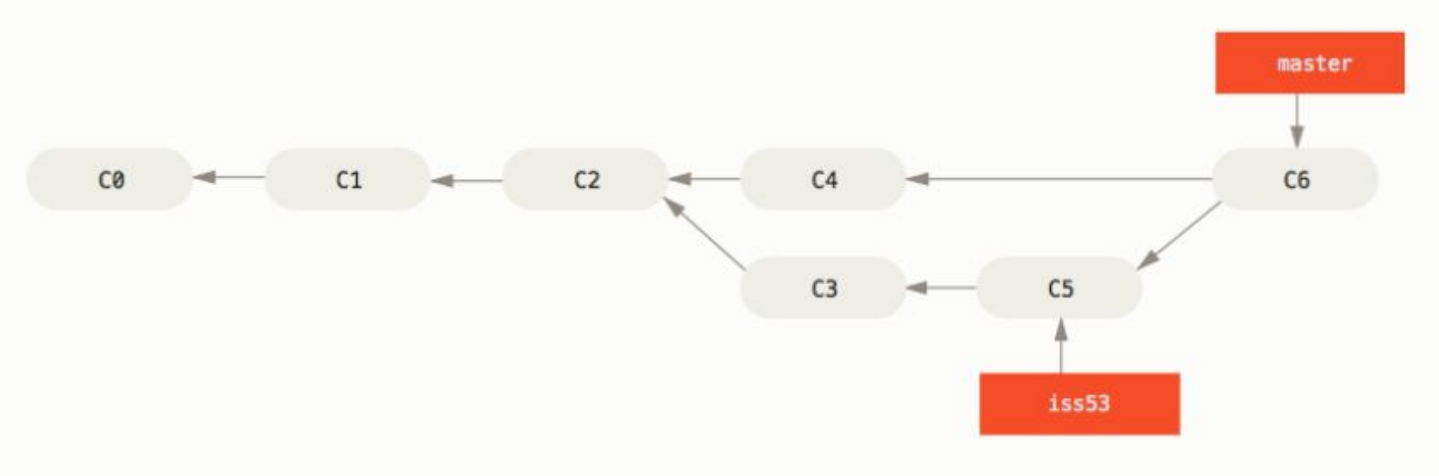
Fast Forward merging with master

```
$ git checkout master

$ git merge hotfix
```



Merging with Recursive strategy

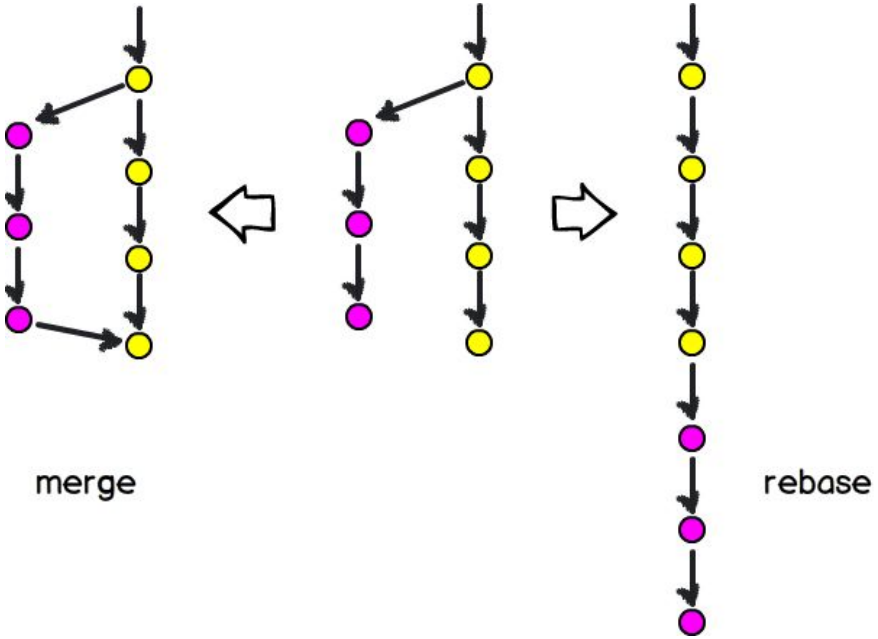
```
$ git checkout iss53

$ vim index.html

$ git commit  -m 'Finish the new footer [issue 53]'

$ git checkout master

$ git merge iss53
```



Rebase master to feature

```
$ git checkout master

$ git rebase -i master [feature-branch-name]

# CONFLICT RESOLUTION

git rebase --continue

# The same until nothing to rebase any more.

git push origin -f [feature-branch-name]
```

Rebase vs Merge



Rebase vs Merge

Comparison Chart

Merge	Rebase
It integrates changes while preserving the ancestry of each commit history.	It rewrites history by creating new commits for each commit in the source branch.
First you switch to the branch to be merged and then use the merge command to select a branch to merge in.	First you select a branch to rebase and then use the rebase command to select where to put it.
It is a one-step operation with one place to resolve merge conflicts.	It is a multi-step operation meaning the steps are smaller, but there are more of them.
Commits remain reachable from the branch.	Commits once reachable are no longer reachable.

DB Difference Between.net

Rebase or merge?

Rebase changes history.

Should we lie about our project history?

Does anyone care if i tried one thing and then another on my code?

You can get the best of both worlds: rebase local changes before pushing to clean up your work, but never rebase anything that you've pushed somewhere.

Common git commands

git clone username@host:/path/to/repository: Create a working copy of a remote server

git add *: Add all files to staging (index):

git commit -m "Commit message": Commit changes to head

git push origin <branchname>: Push the branch to your remote repository, so others can use it

git merge <branchname>: To merge a different branch into your active branch

git push origin master: Send changes to the master branch of your remote repository

git status : List the files you've changed and those you still need to add or commit

git checkout -b <branchname>: Create a new branch and switch to it

git branch -d <branchname>: Delete the feature branch

git push origin :<branchname>: Delete a branch on your remote repository

Git reflog - git Reset

If Something bad happens and you want to go back to find a commit to revert you can use

git reflog

git reflog

git reflog with date

git reflog --relative-date

#reset to the commit f682c8f

git reset --hard f682c8f

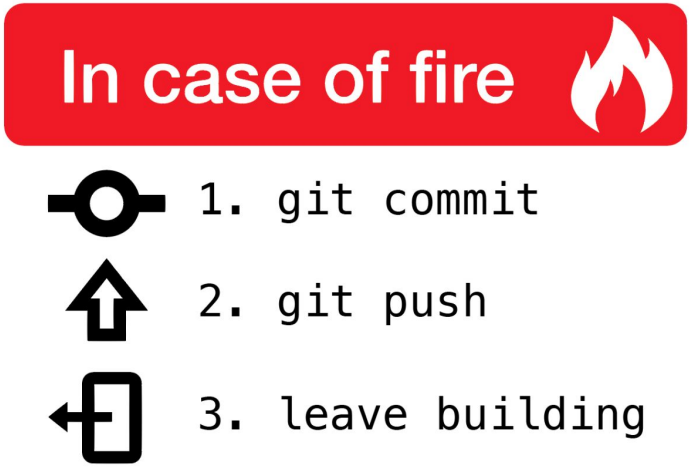
#force push

git push origin -f feature-branch

```
jekyll master $ git reflog
6703083... HEAD@{0}: pull origin master: Fast forward
fe71d2b... HEAD@{1}: commit: Updating README with added
eac6b03... HEAD@{2}: commit: Making sure that posts fla
f682c8f... HEAD@{3}: commit: Added publish flag to post
9478f1b... HEAD@{4}: rebase: Modifying the README a bit
7e178ff... HEAD@{5}: checkout: moving from master to 7e
cda3b9e... HEAD@{6}: HEAD~1: updating HEAD
3b75036... HEAD@{7}: merge newfeature: Merge made by re
cda3b9e... HEAD@{8}: commit: Modifying the README a bit
6981921... HEAD@{9}: checkout: moving from newfeature t
7e178ff... HEAD@{10}: commit: Rewriting history is fun
4a97573... HEAD@{11}: commit: all done with TODOs
eb60603... HEAD@{12}: commit: README
```

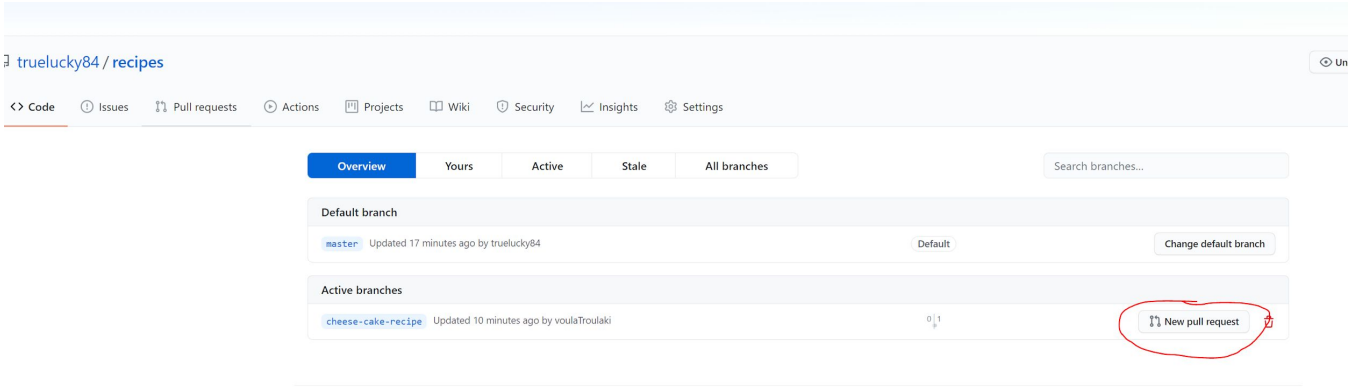
Github

Github is a website that hosts Git Repositories online, making it easy for developers to share code



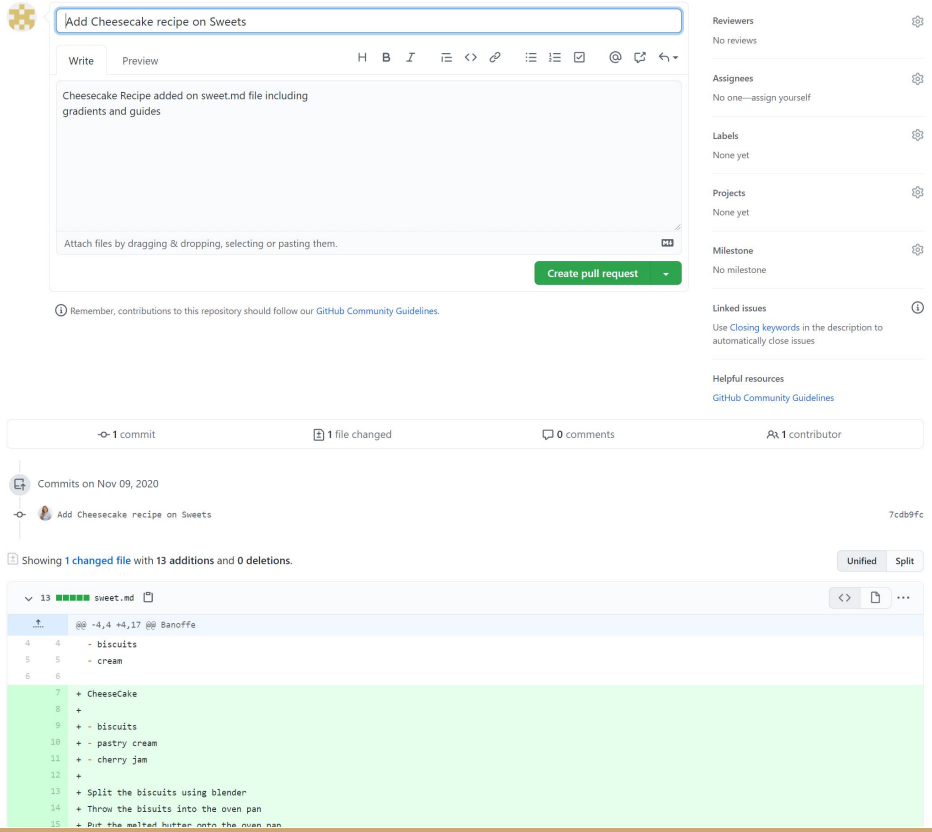
Github Pull Requests 1/2

Create New Pull Request



Github Pull Requests 2/2

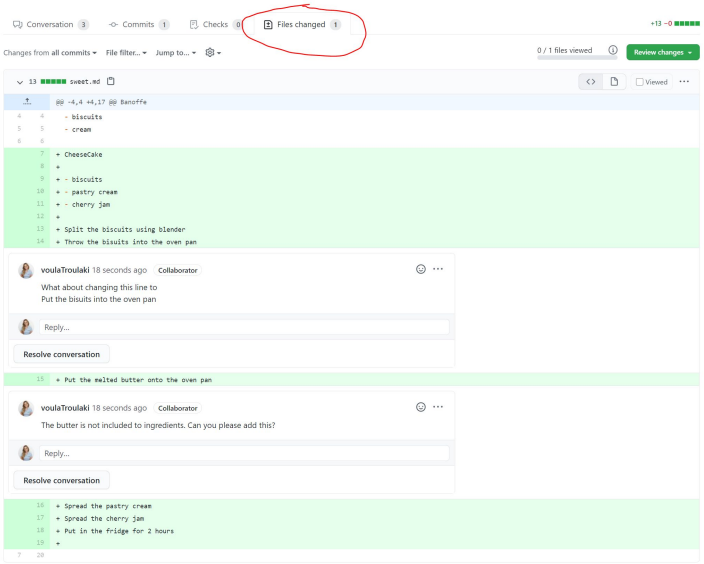
- Authors/Reviewers
- The Smaller the better for Everyone
- Agile and Pull Requests
- Clean Commit history
- Detailed Description
- Please review your PR First
- CI / CD tools
- Style guides

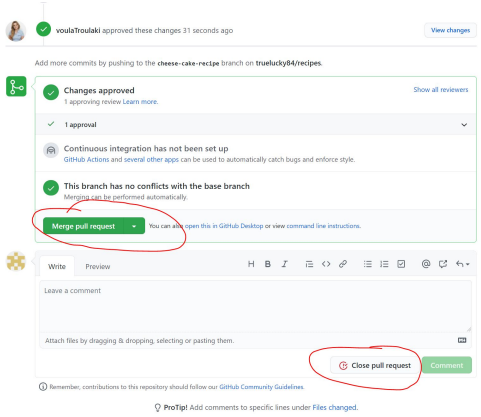


Code Reviews

When you are a Reviewer

- Be specific
- don't be rude
- Read the file changes line line
- Give code examples
- Explain why
- Suggest another way with links to principles
- Check this
  - https://mtlynch.io/human-code-reviews-1/





References

- https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell#ch03-git-branching
- https://www.atlassian.com/git/tutorials/why-git#git-for-developers
- https://www.slideshare.net/qrush/git-started-with-git
- http://www.differencebetween.net/technology/difference-between-git-rebase-and-merge/
- https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html
- http://gitready.com/intermediate/2009/02/09/reflog-your-safety-net.html
- https://mtlynch.io/human-code-reviews-1/

Questions