git features 1

Web Development 2

john.rellis@setu.ie

Git Features

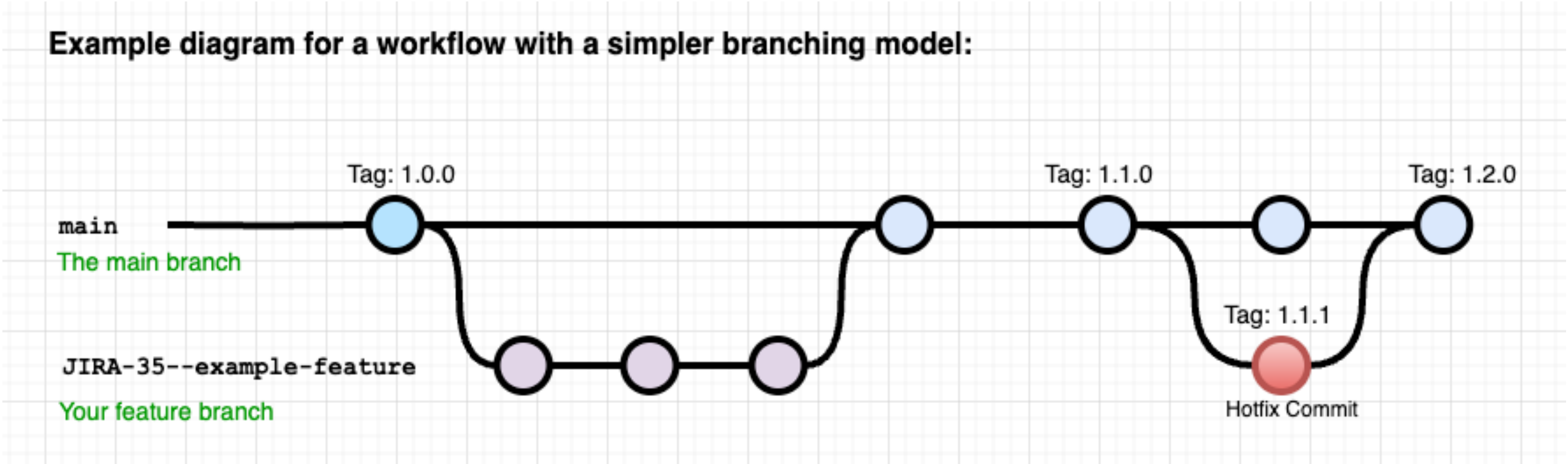
- Branching and Merging
- Remote Branches and Pull Requests (in brief)

Branching and Merging

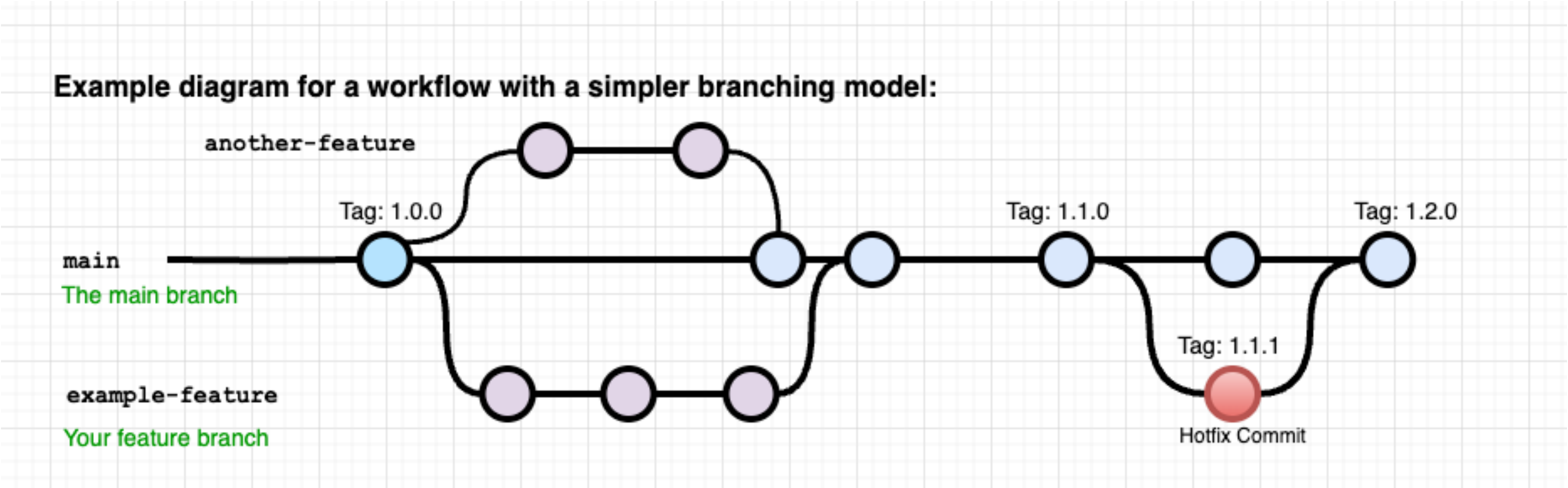
- In git, all work is done on a branch
- When you create a repository, you create the “main” branch
  - This “main” branch was once called ”master” but it is now less common
- Up to now you have seen staging and then committing all on the main branch
- Typically, a branch is created from a commit on main to, for example, create a new feature or fix a bug. This is often referred to as “feature branching”
- Once the feature is complete/bug fixed, we then merge back to main, between such time, others may have augmented main. It is at this time merge conflicts are managed, if they exist

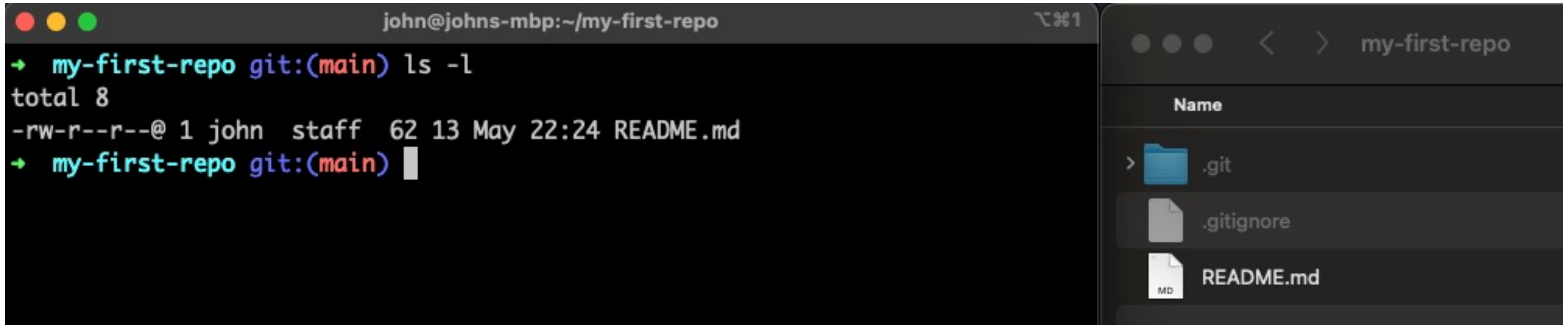
Simple Branching Model

- Below we can see a typical way of visualising git branches
- The dots are commits
- A release is tagged on the main branch and the next feature is started via a feature branch, when done, we merge back and delete our branch

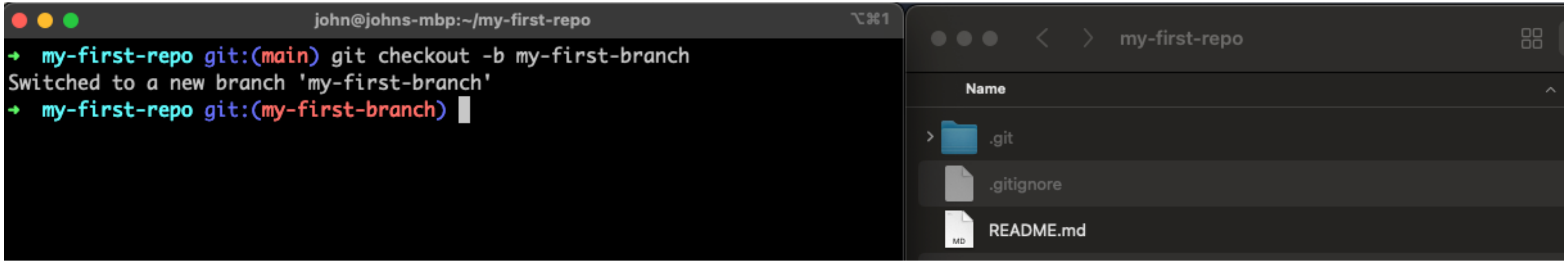


Extended example with 2 features

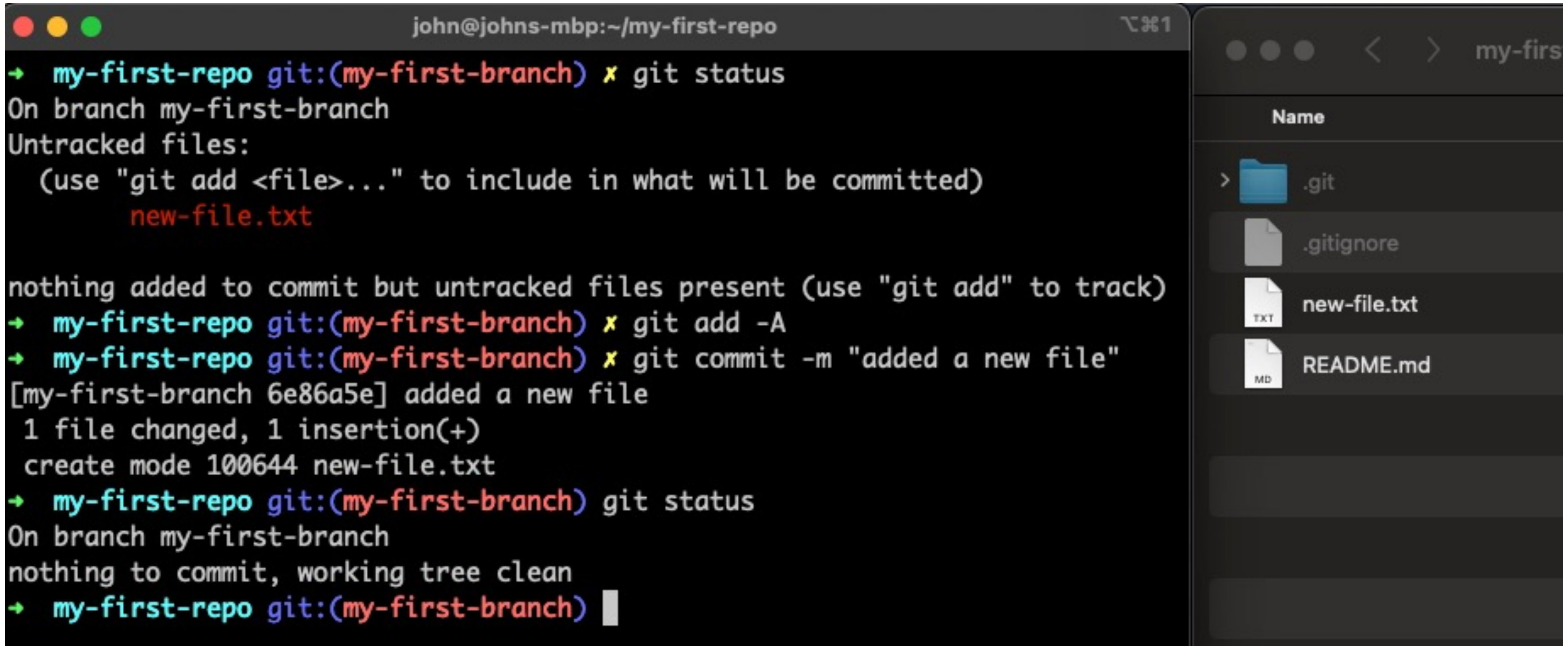




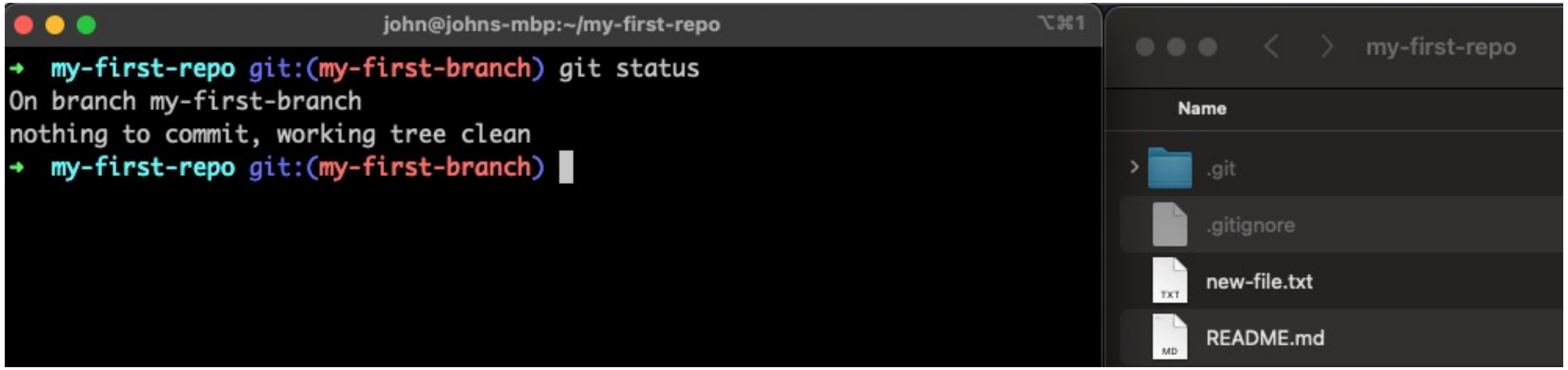
```
➜  my-first-repo git:(main) ls -l
total 8
-rw-r--r--@ 1 john  staff  62 13 May 22:24 README.md
➜  my-first-repo git:(main) 
```



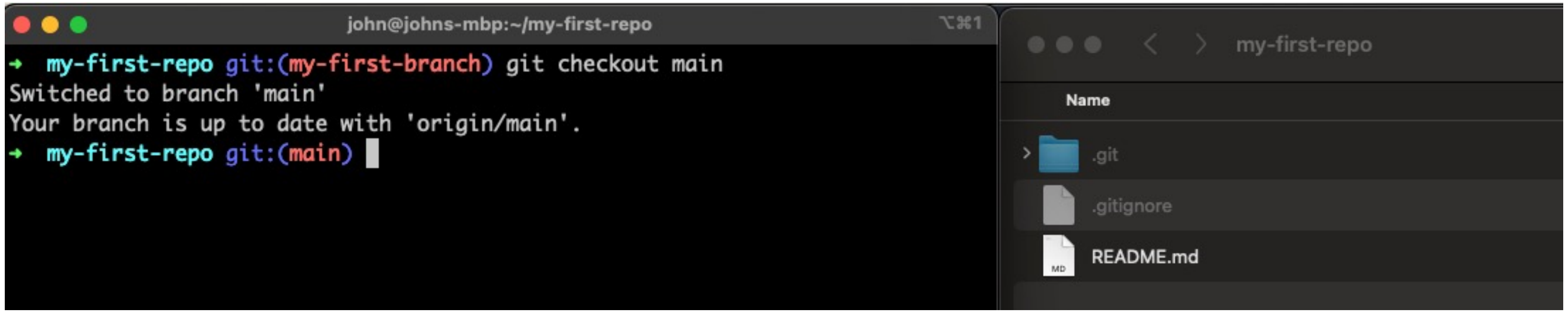
```
➜  my-first-repo git:(main) git checkout -b my-first-branch
Switched to a new branch 'my-first-branch'
➜  my-first-repo git:(my-first-branch) 
```



```
➜  my-first-repo git:(my-first-branch) ✗ git status
On branch my-first-branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        new-file.txt

nothing added to commit but untracked files present (use "git add" to track)
➜  my-first-repo git:(my-first-branch) ✗ git add -A
➜  my-first-repo git:(my-first-branch) ✗ git commit -m "added a new file"
[my-first-branch 6e86a5e] added a new file
 1 file changed, 1 insertion(+)
 create mode 100644 new-file.txt
➜  my-first-repo git:(my-first-branch) git status
On branch my-first-branch
nothing to commit, working tree clean
➜  my-first-repo git:(my-first-branch) 
```

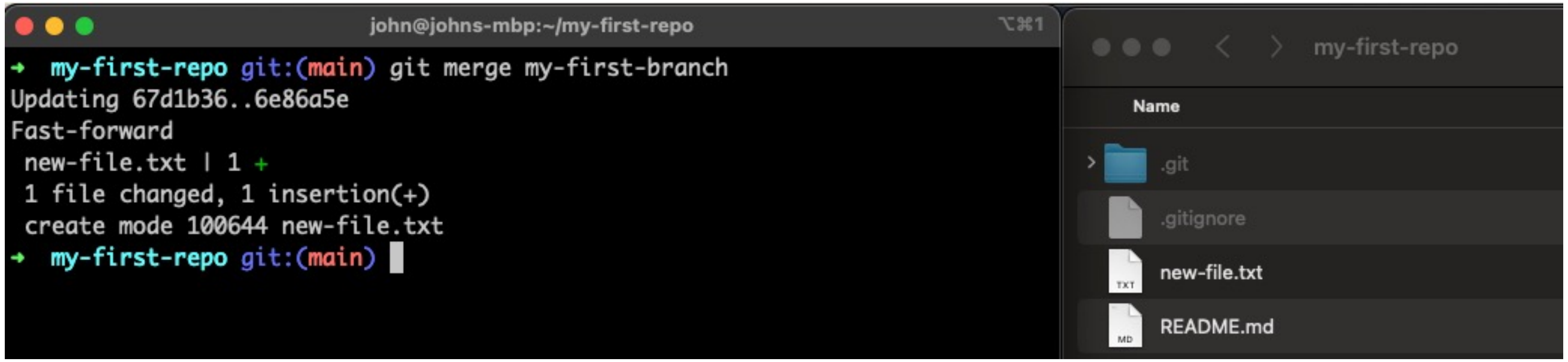


```
➜  my-first-repo git:(my-first-branch) git status
On branch my-first-branch
nothing to commit, working tree clean
➜  my-first-repo git:(my-first-branch) 
```



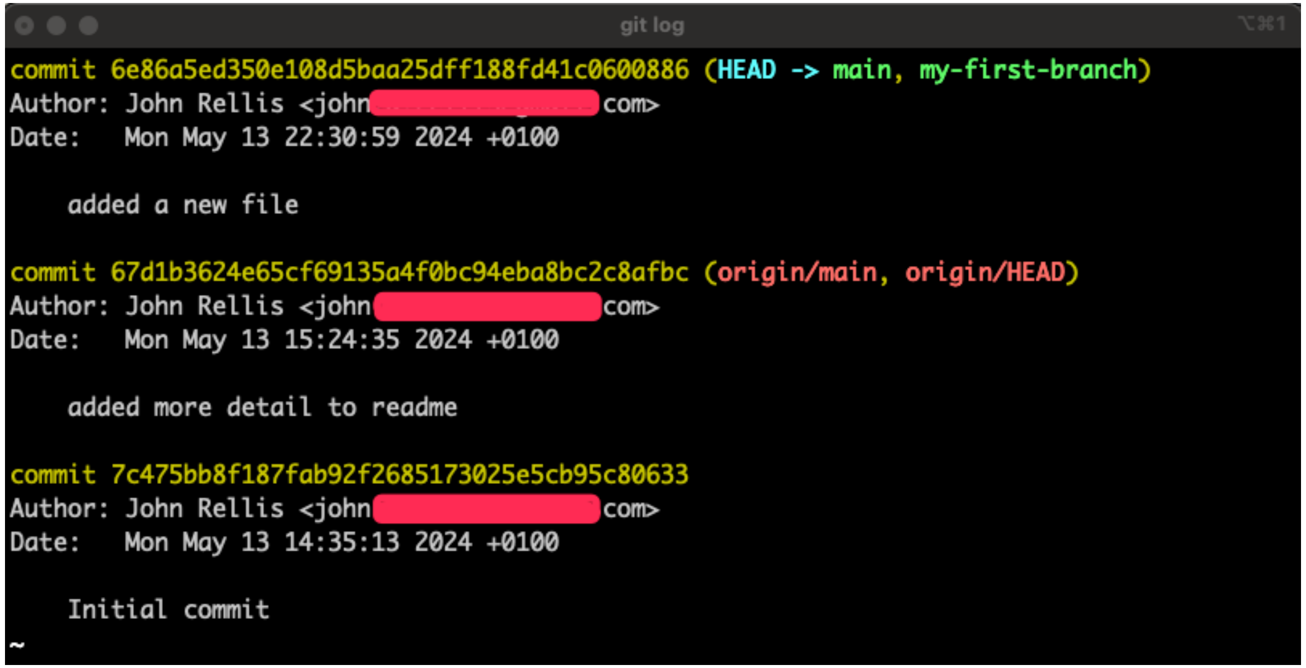
```
➜  my-first-repo git:(my-first-branch) git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
➜  my-first-repo git:(main) 
```

Notice, new-file.text is gone on the file system as we changed branch. When we merge below, it reappears.



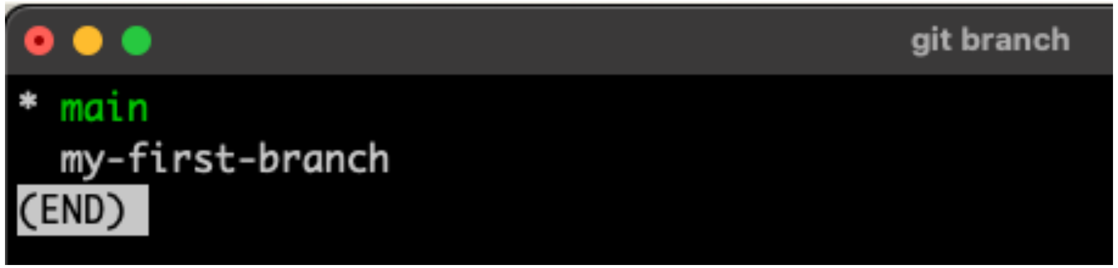
```
➜  my-first-repo git:(main) git merge my-first-branch
Updating 67d1b36..6e86a5e
Fast-forward
 new-file.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 new-file.txt
➜  my-first-repo git:(main) 
```

git log

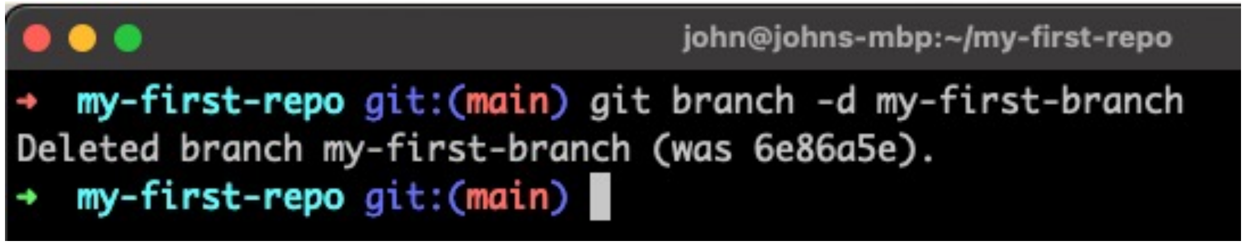


```
commit 6e86a5ed350e108d5baa25dff188fd41c0600886 (HEAD -> main, my-first-branch)
Author: John Rellis <john[redacted]com>
Date:   Mon May 13 22:30:59 2024 +0100

    added a new file

commit 67d1b3624e65cf69135a4f0bc94eba8bc2c8afbc (origin/main, origin/HEAD)
Author: John Rellis <john[redacted]com>
Date:   Mon May 13 15:24:35 2024 +0100

    added more detail to readme

commit 7c475bb8f187fab92f2685173025e5cb95c80633
Author: John Rellis <john[redacted]com>
Date:   Mon May 13 14:35:13 2024 +0100

    Initial commit
~
```

git branch

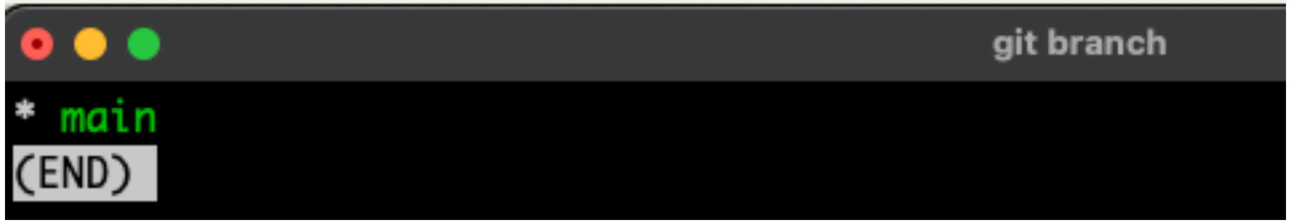


```
* main
  my-first-branch
(END)
```



```
➜  my-first-repo git:(main) git branch -d my-first-branch
Deleted branch my-first-branch (was 6e86a5e).
➜  my-first-repo git:(main) 
```

git branch

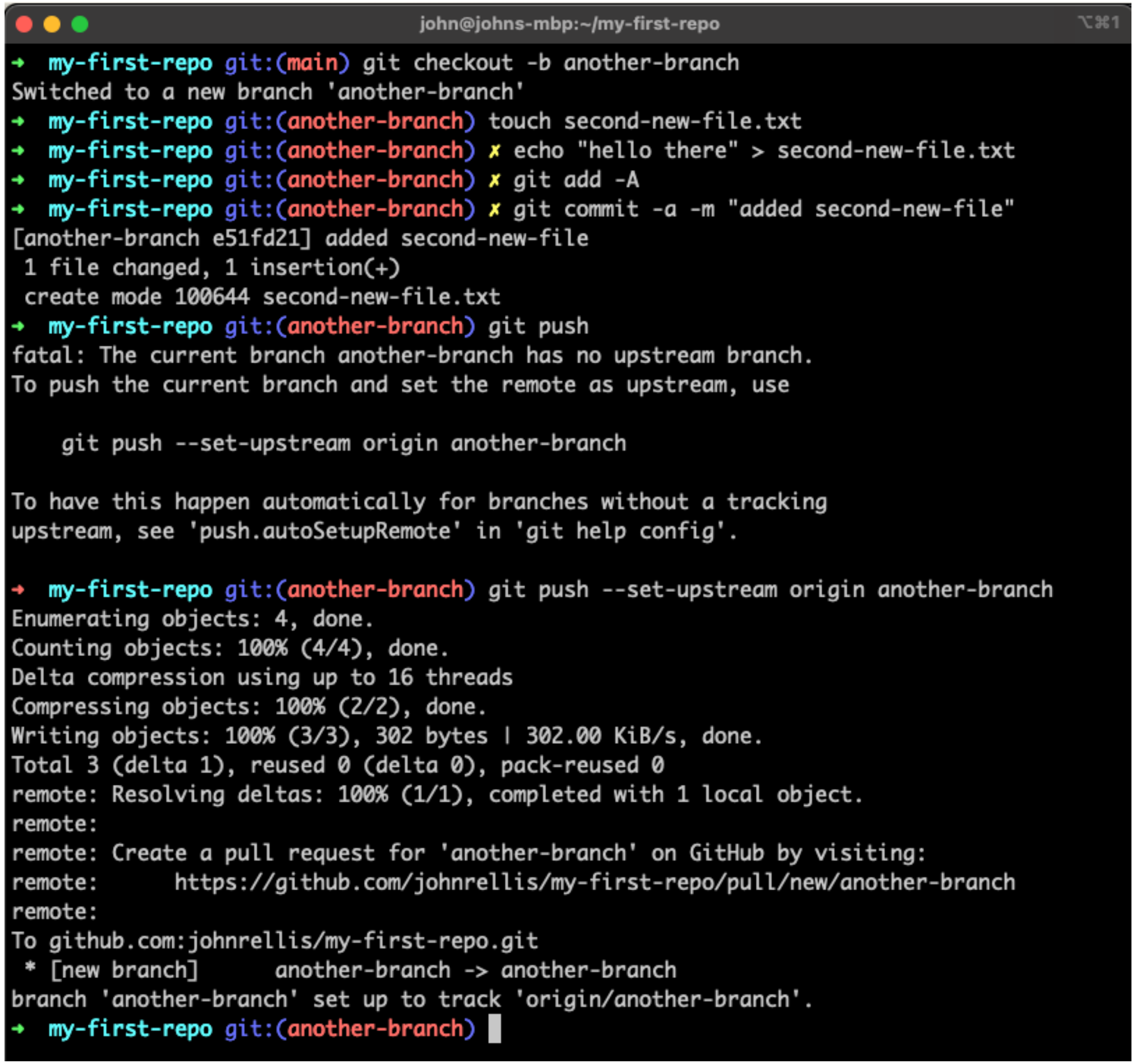


```
* main
(END)
```

Recap

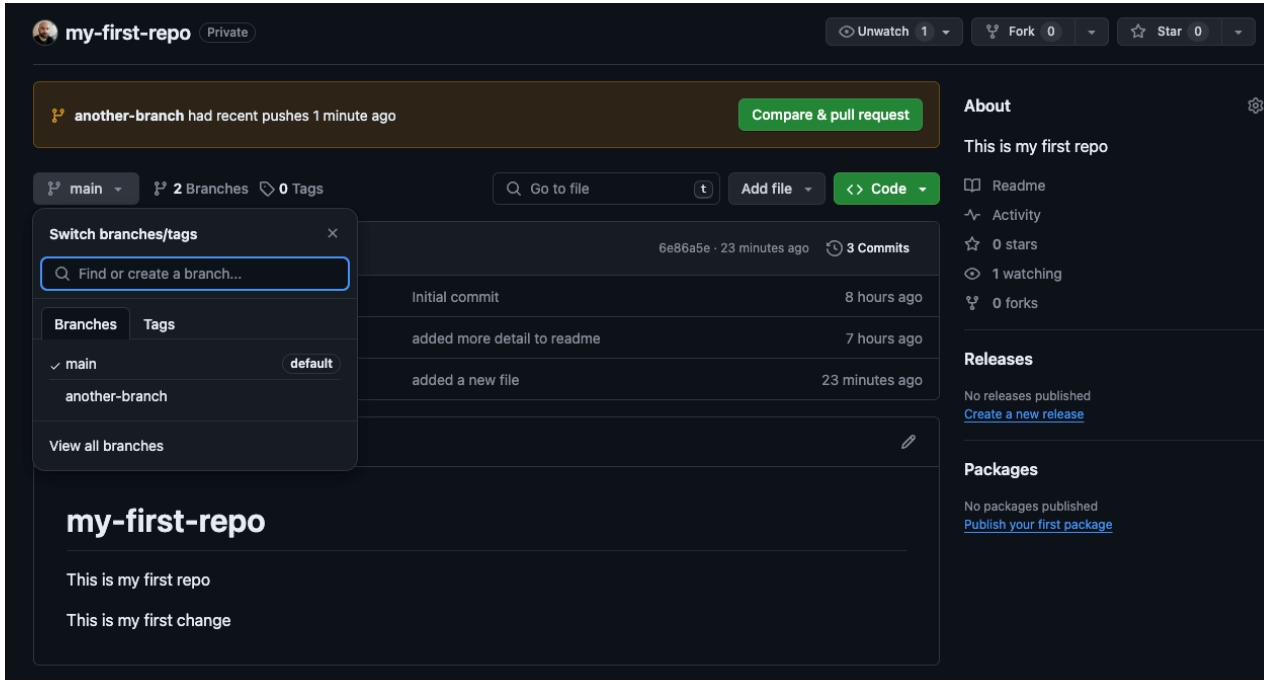
- We created a new branch called my-first-branch - *git checkout –b my-first-branch*
  - -b creates the new branch and switches you to it
- git status will tell you what branch you are on
- We created a new file, added it and committed it to my-first-branch
  - Create file
  - *git add new-file.txt*
  - *git commit –m “added a new file”*
- We switched back to main
  - *git checkout main* (or in newer versions of git, *git switch main*)
- We noted the file system changed when switching branches, not something hidden in the depths of git
  - Personally, this was a big understanding moment for me with branches, nothing is hidden, you can see the file system change as you switch between branches.
- We then merged our branch into main - *git merge my-first-branch*
- Delete the branch - *git branch –d my-first-branch*
- To view available branches – *git branch*
- To view the git log – *git log*
- Use q to quit from the log and the branch view

Git and Remote Branches



john@johns-mbp:~/my-first-repo ⌥⌘1

```
➜  my-first-repo git:(main) git checkout -b another-branch
Switched to a new branch 'another-branch'
➜  my-first-repo git:(another-branch) touch second-new-file.txt
➜  my-first-repo git:(another-branch) ✗ echo "hello there" > second-new-file.txt
➜  my-first-repo git:(another-branch) ✗ git add -A
➜  my-first-repo git:(another-branch) ✗ git commit -a -m "added second-new-file"
[another-branch e51fd21] added second-new-file
 1 file changed, 1 insertion(+)
 create mode 100644 second-new-file.txt
➜  my-first-repo git:(another-branch) git push
fatal: The current branch another-branch has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin another-branch

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

➜  my-first-repo git:(another-branch) git push --set-upstream origin another-branch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 302 bytes | 302.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'another-branch' on GitHub by visiting:
remote:      https://github.com/johnrellis/my-first-repo/pull/new/another-branch
remote:
To github.com:johnrellis/my-first-repo.git
 * [new branch]      another-branch -> another-branch
branch 'another-branch' set up to track 'origin/another-branch'.
➜  my-first-repo git:(another-branch) 
```



my-first-repo Private

another-branch had recent pushes 1 minute ago

main · 2 Branches · 0 Tags

Switch branches/tags

Find or create a branch...

Branches | Tags

- ✓ main default
- another-branch

View all branches

6e86a5e · 23 minutes ago · 3 Commits

	Initial commit	8 hours ago
	added more detail to readme	7 hours ago
	added a new file	23 minutes ago

my-first-repo

This is my first repo

This is my first change

About

This is my first repo

Releases

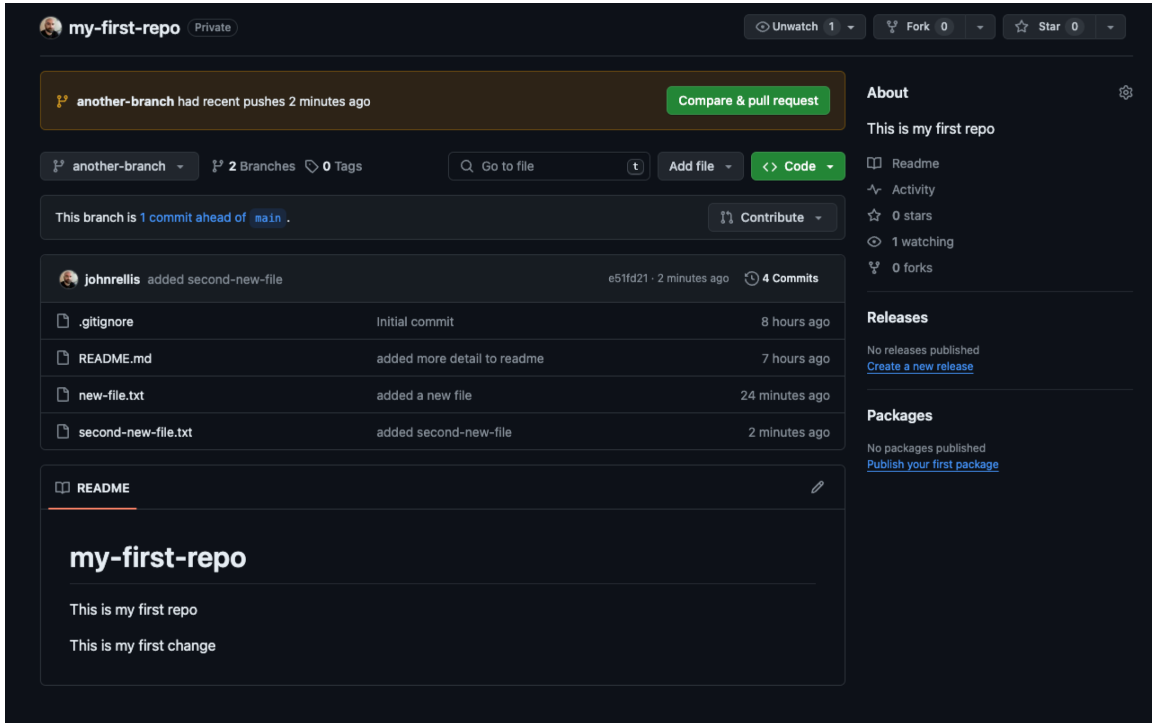
No releases published

Create a new release

Packages

No packages published

Publish your first package



my-first-repo Private

another-branch had recent pushes 2 minutes ago

another-branch · 2 Branches · 0 Tags

This branch is 1 commit ahead of `main`.

johnrellis added second-new-file · e51fd21 · 2 minutes ago · 4 Commits

.gitignore	Initial commit	8 hours ago
README.md	added more detail to readme	7 hours ago
new-file.txt	added a new file	24 minutes ago
second-new-file.txt	added second-new-file	2 minutes ago

README

my-first-repo

This is my first repo

This is my first change

About

This is my first repo

Releases

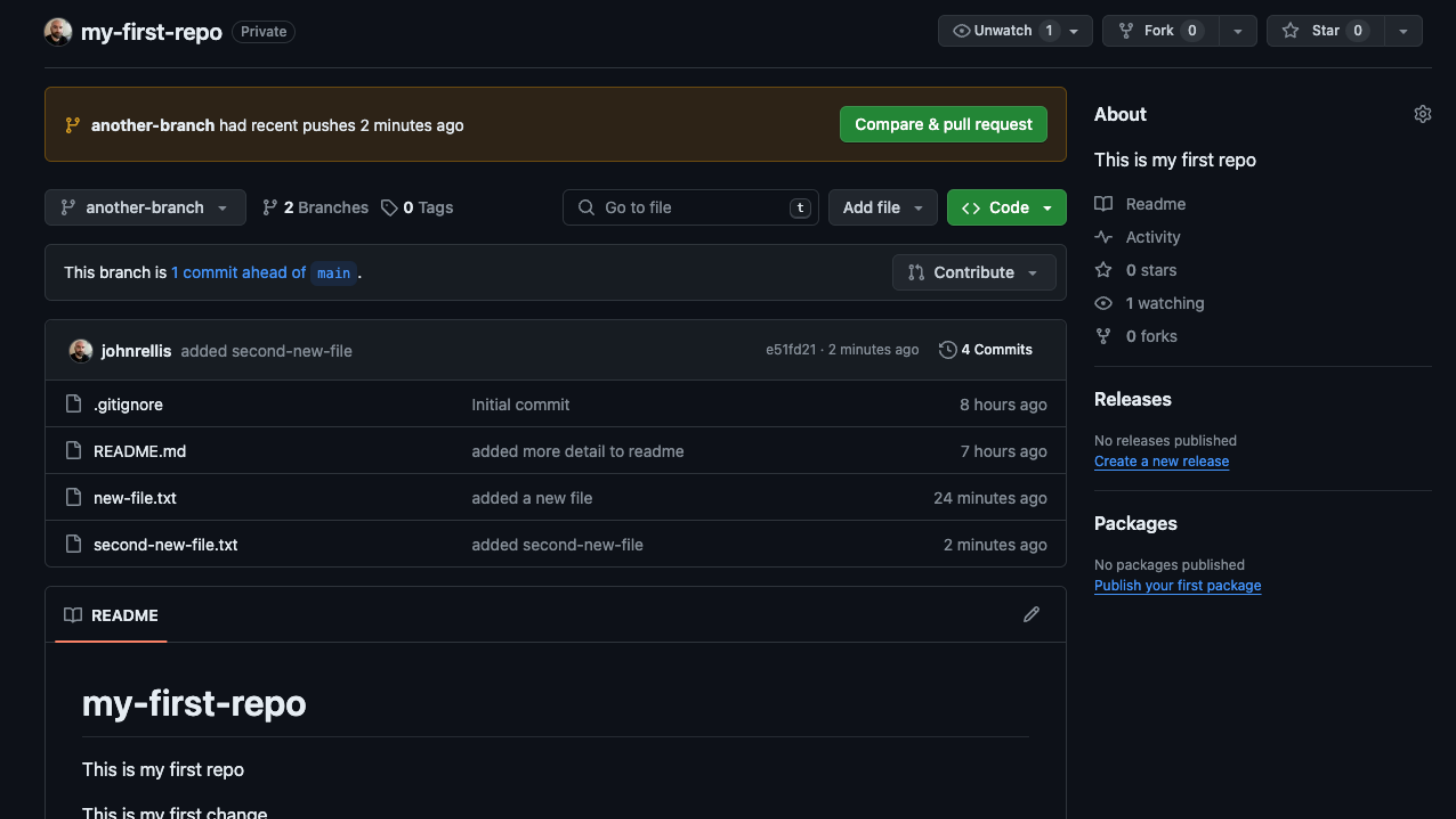
No releases published

Create a new release

Packages

No packages published

Publish your first package



another-branch had recent pushes 2 minutes ago

This branch is 1 commit ahead of `main`.

johnrellis added second-new-file e51fd21 · 2 minutes ago

.gitignore	Initial commit	8 hours ago
README.md	added more detail to readme	7 hours ago
new-file.txt	added a new file	24 minutes ago
second-new-file.txt	added second-new-file	2 minutes ago

This is my first repo

No releases published
Create a new release

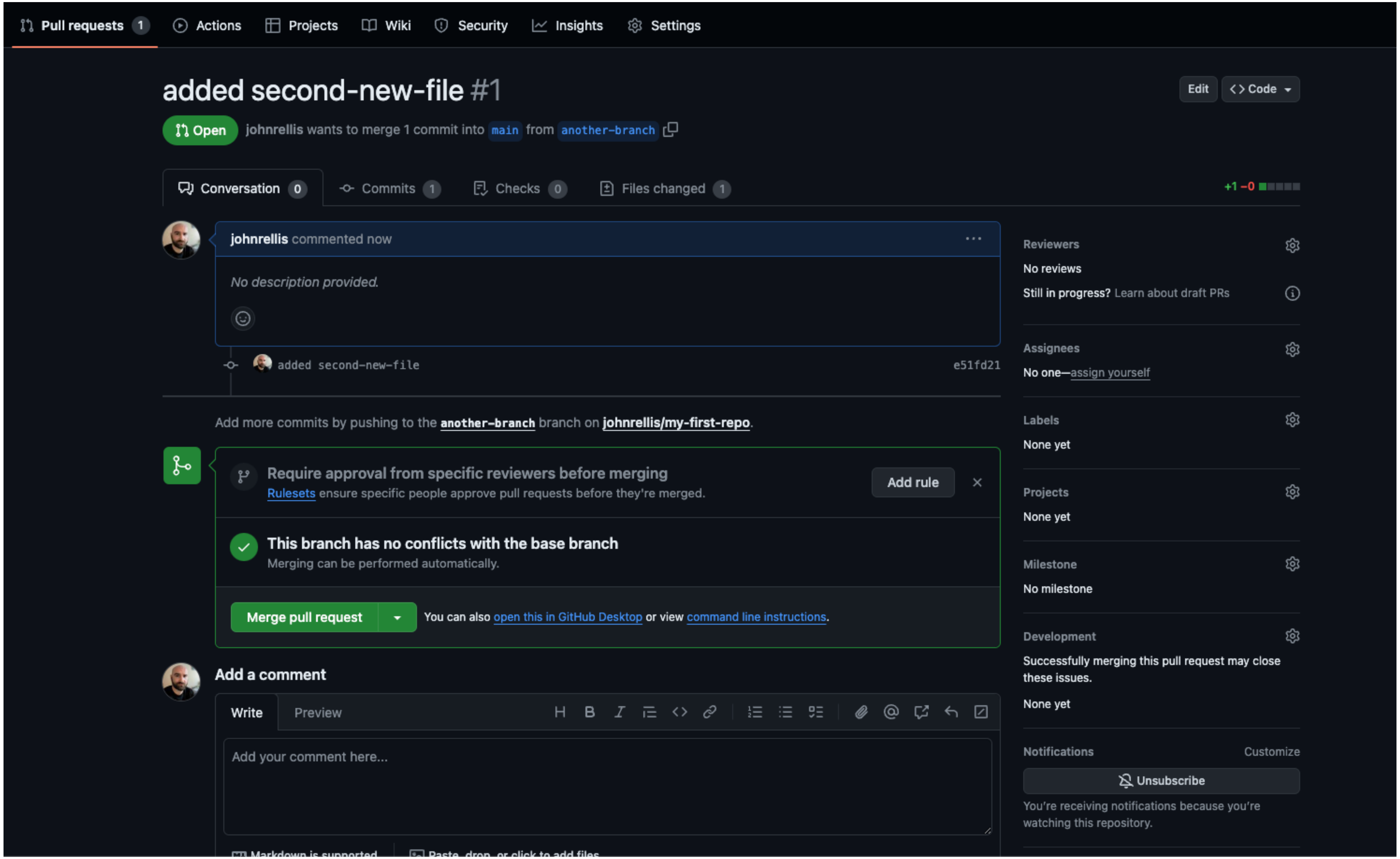
No packages published
Publish your first package

my-first-repo

This is my first repo

This is my first change

Compare and Pull Request



Pull requests 1 · Actions · Projects · Wiki · Security · Insights · Settings

added second-new-file #1

Edit · <> Code

Open · johnrellis wants to merge 1 commit into `main` from `another-branch`

Conversation 0 · Commits 1 · Checks 0 · Files changed 1 · +1 −0

johnrellis commented now

No description provided.

`added second-new-file` `e51fd21`

Add more commits by pushing to the **another-branch** branch on **johnrellis/my-first-repo**.

Require approval from specific reviewers before merging
Rulesets ensure specific people approve pull requests before they're merged.

Add rule

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.

Add a comment

Write · Preview

Add your comment here...

Markdown is supported · Paste, drop, or click to add files

Reviewers: No reviews
Still in progress? Learn about draft PRs

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Successfully merging this pull request may close these issues.
None yet

Notifications · Customize
Unsubscribe
You're receiving notifications because you're watching this repository.

Where to next?

- Check out - https://git-scm.com/about
- Free ebook - https://git-scm.com/book/en/v2
- Discover what is a git tag?
- https://code.visualstudio.com/docs/sourcecontrol/overview
  - Videos worth watching
- https://gist.github.com/bryanbraun/8c93e154a93a08794291df1fcdce6918
- https://github.com/nicoespeon/gitgraph.js/tree/master/packages/gitgraph-js