

A Brief Introduction to Git and Github

Michael Luu

May 11, 2023

Slides are publicly available at:

<https://mluu921.github.io/git-github-slides/>

"FINAL".doc



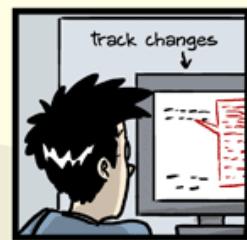
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRAD SCHOOL????.doc

JORGE CHAM © 2012



What is git?

Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.¹



What is GitHub?

GitHub, Inc. is an Internet hosting service for software development and version control using Git. It provides the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project.¹





git



GitHub

1. It is a software	1. It is a service
2. It is installed locally on the system	2. It is hosted on Web
3. It is a command line tool	3. It provides a graphical interface
4. It is a tool to manage different versions of edits, made to files in a git repository	4. It is a space to upload a copy of the Git repository
5. It provides functionalities like Version Control System Source Code Management	5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features

Why...?

SCIENTIFIC POSTER



Reproducibility Crisis

- The reproducibility crisis refers to the fact that many scientific findings in biomedical research cannot be replicated by independent researchers.
 - The crisis has led to calls for increased transparency and data sharing in scientific research, as well as greater emphasis on replication studies to confirm the validity of scientific findings.
-
- Peng, R. D. (2011). Reproducible research in computational science. *Science*, 334(6060), 1226-1227. <https://doi.org/10.1126/science.1213847>

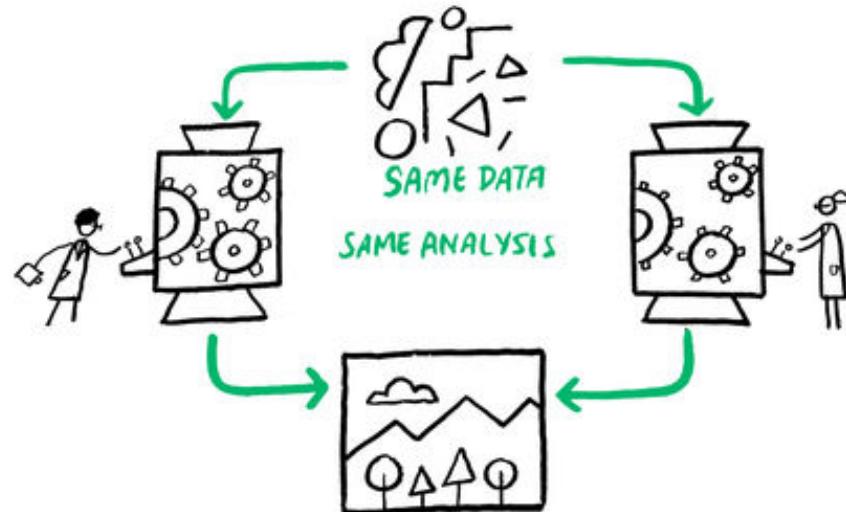
Reproducibility Crisis

An analysis of past studies indicates that the cumulative (total) prevalence of irreproducible preclinical research exceeds 50%, resulting in approximately US\$28,000,000,000 (US\$28B)/year spent on preclinical research that is not reproducible—in the United States alone.

- Freedman LP, Cockburn IM, Simcoe TS (2015) The Economics of Reproducibility in Preclinical Research. PLoS Biol 13(6): e1002165.
<https://doi.org/10.1371/journal.pbio.1002165>

Reproducible Research

REPRODUCIBLE



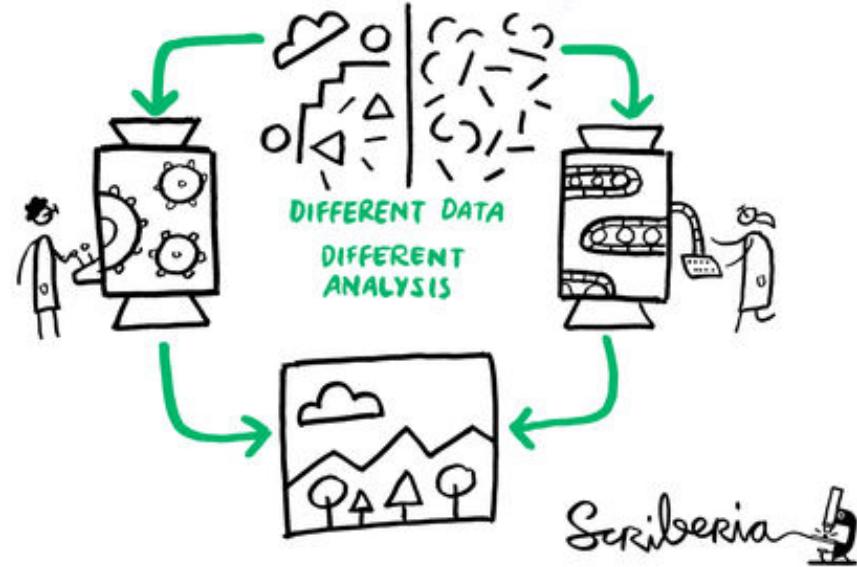
REPLICABLE



ROBUST



GENERALISABLE



Important Git Jargon

- **Repo (Repository)** - A repo or repository is generally a folder in which git manages and tracks the files that are contained within.
- **Staging** - This is a ‘holding area’ of the files that you would like to version control. We need to ‘stage’ a file before we can ‘commit’ it.
- **Commit** - A commit is a ‘snapshot’ of current changes among the files that have been ‘staged’ in your repository. A commit also requires a small message or text description, in which you can describe the changes.

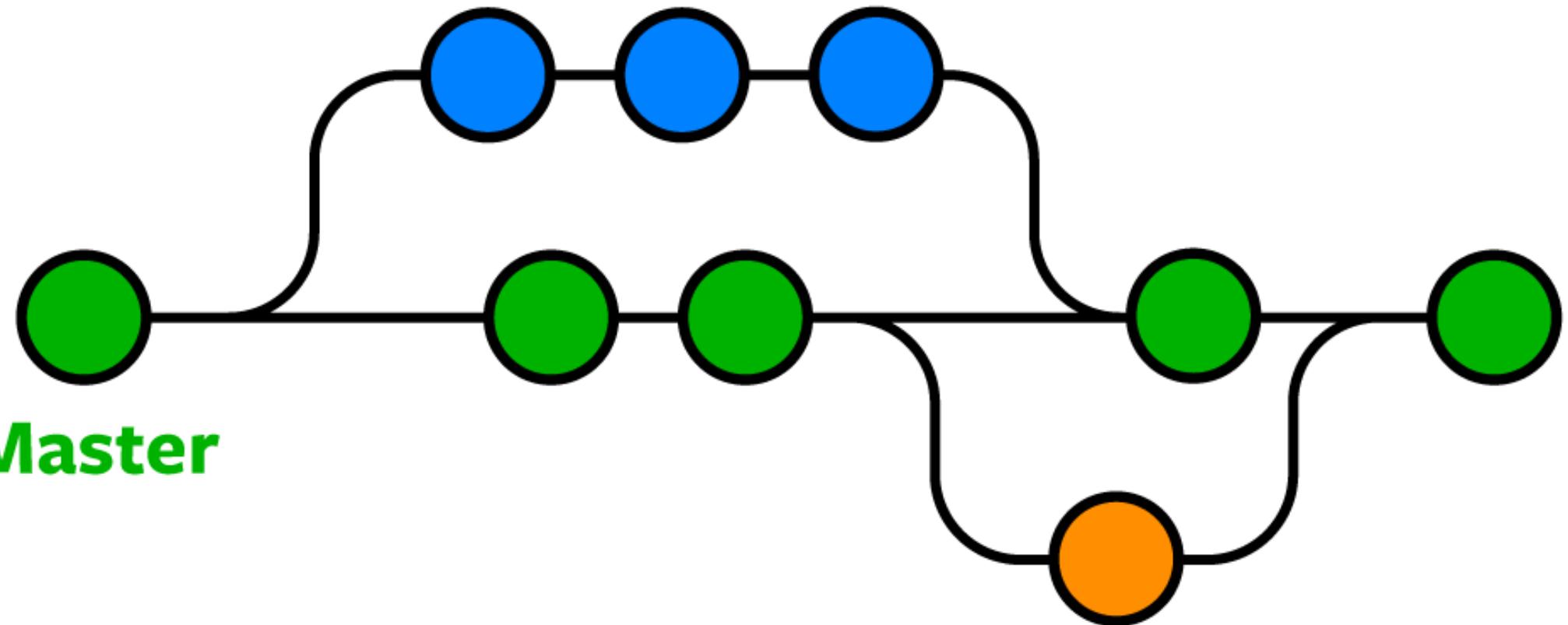
Important Git Jargon

- **Branching & Merging** - Git allows for a non-linear workflow, in which we can ‘branch’ away from the current source. With Git we can create infinite number of branches in which we can experiment without affecting the main source code. If content, we can merge those changes from various branches back to the main source.

Important Git Jargon

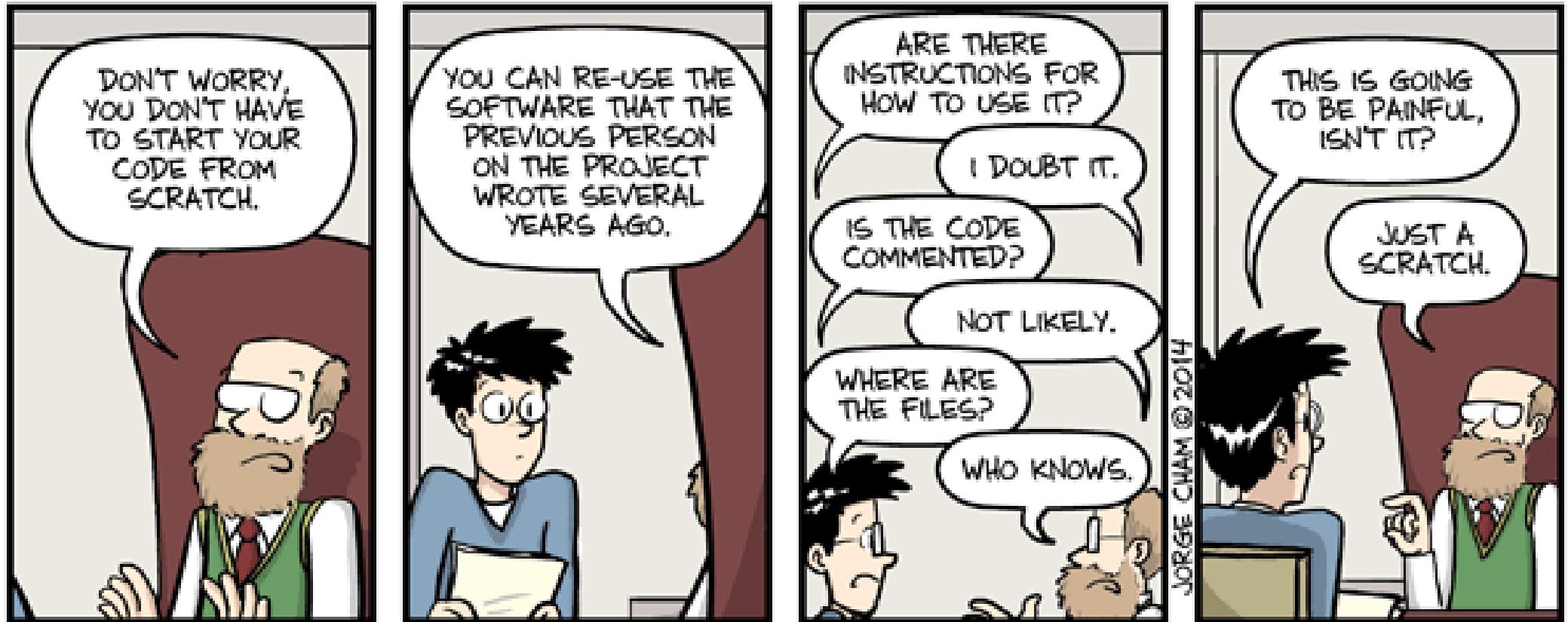
- **Push & Pull** - If git is configured to work with Github (or another remote repository), **push** allows us to push the changes to a remote repository (Github), and **pull** allows us to pull in the changes from Github
- **Cloning** - Allows the user to create **clone** of the Github repository, including the files, history, and branches onto the local machine

Your Work



Master

Someone Else's Work



JORGE CHAM © 2014

WWW.PHDCOMICS.COM

Why you should use it?

- git provides a highly structured workflow that promotes reproducibility and transparency
- git provides a continuous log of the changes you have done on your project / analysis (via **Commit Messages**)
- git provides the ability to travel through time with the files in your project - easily jump back to previous versions of files (via **Commits**)
- git provides the freedom to explore new ideas/analysis without the fear of affecting your primary analysis (via **Branching**)

Why you should use it?

- Github provides a remote backup of your **git repository**
- Github provides a highly structured method of sharing your **repository** with other collaborators
- Github provides you with an online presence (think of this as an online portfolio of your work)
- Github provides you with a free online hosting / website for your project
- Git and Github is a industry standard version control system
 - this skill is transferable to many industries

However...

git is complicated

- git is designed as a command line tool (e.g. to take full advantage of git, you will have to learn commands to enter in the terminal)
- There is a barrier to entry on getting your local git repository to ‘talk’ with Github
- Although git can version control any type of files, it is best used in conjunction with text files (e.g. source code)

git is complicated

- There's additional work to consider in your current workflow
 - Storing your project into a “project folder”
 - Making the folder a git repository
 - Making commits at logical points in time to snapshot your project
 - Writing **commit** messages to describe the changes
 - **Pushing** your work to Github

Setting up Git and Github



--distributed-is-the-new-centralized



Search entire site...

Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is [easy to learn](#) and has a [tiny footprint with lightning fast performance](#). It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like [cheap local branching](#), [convenient staging areas](#), and [multiple workflows](#).



About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

GUI clients and binary releases for all major platforms.



Community

Get involved! Bug reporting, mailing list, chat, development and more.



[Pro Git](#) by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).



[Windows GUIs](#)



[Tarballs](#)



[Mac Build](#)



[Source Code](#)



Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change

<https://gitforwindows.org/>



Only show new options

Next

Cancel



Product ▾ Solutions ▾ Open Source ▾ Pricing

Search GitHub

/

Sign in

Sign up



Introducing GitHub Copilot X

Your AI pair programmer is leveling up



Let's build from here

Harnessed for productivity. Designed for collaboration.



Search or jump to...



Pull requests Issues Codespaces Marketplace Explore



Pinned

Customize your pins

[mluu921.github.io](#) Public

Personal website and portfolio for Michael Luu

JavaScript

mluu921

Edit profile

3 followers · 4 following

Achievements



Beta Send feedback

Highlights

★ PRO

Contribution activity

2023

April 2023

2022

Created 100 commits in 6 repositories



2021

[mluu921/zumsteg-multicenter-oral-cavity-ln-staging](#) 42 commits

2020

[mluu921/mluu921.github.io](#) 37 commits

2019

[mluu921/git-github-slides](#) 10 commits

2018

[mluu921/daskivich-nlp-model-patient-facing-output](#) 7 commits

2017

[mluu921/biostats_mrm615_descriptive](#) 2 commits

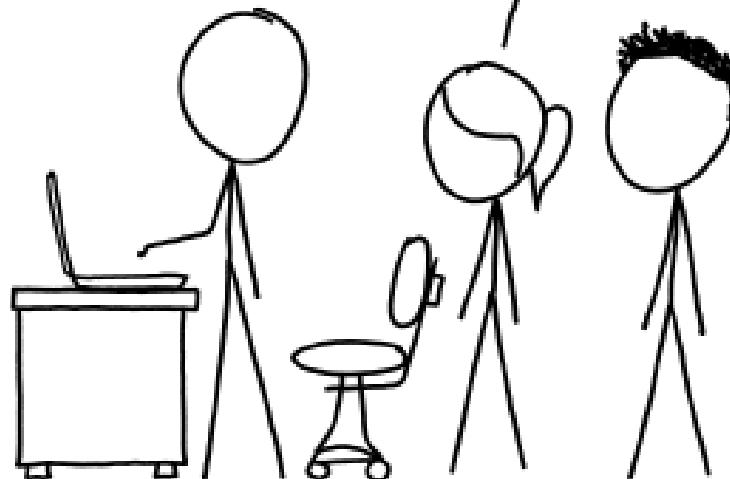
2017

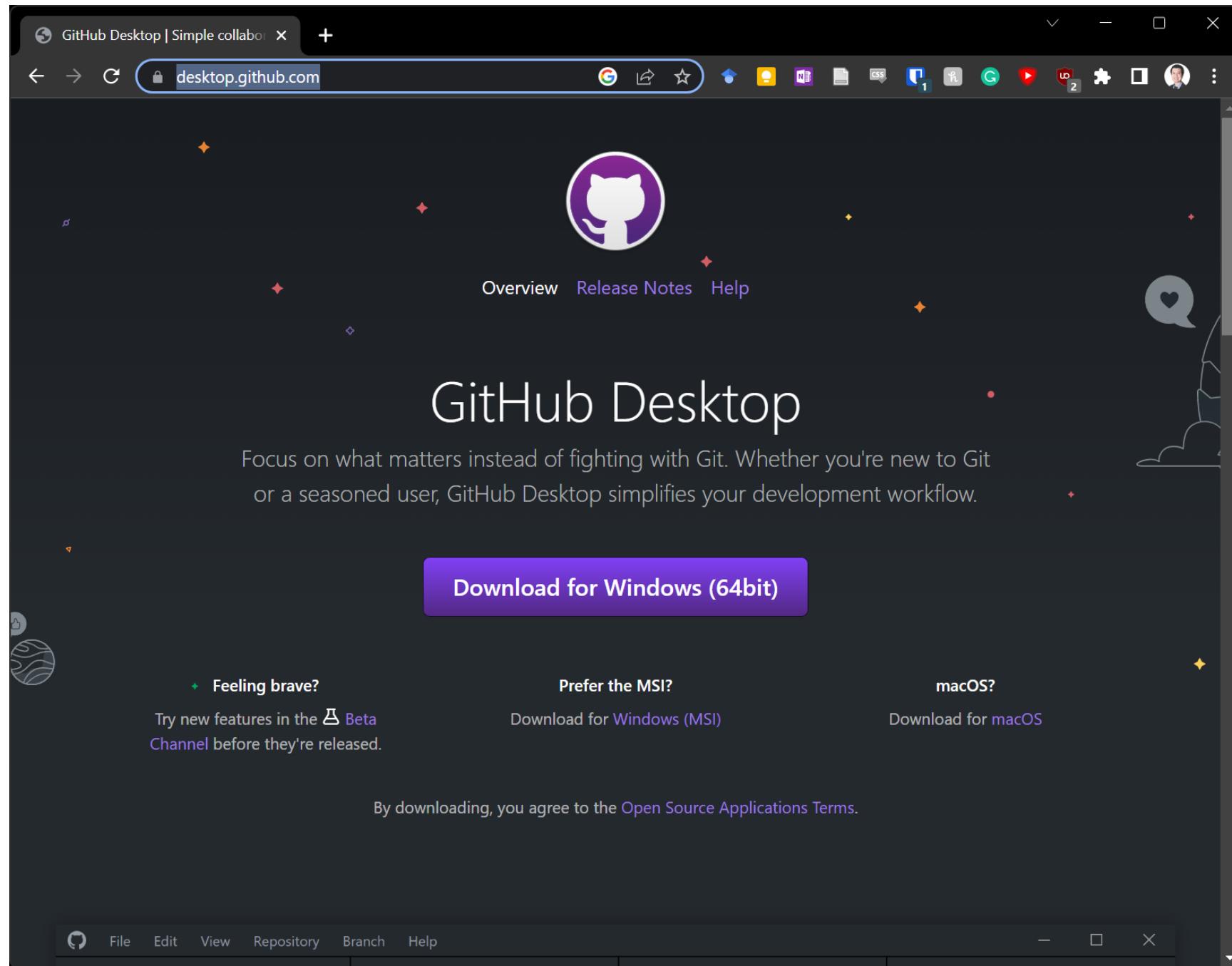
[mluu921/zumsteg-meta-analysis-chemo-nochemo](#) 2 commits

THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.





Creating a Git Repository

File Edit View Repository Branch Help

Current repository git-github-slides

Current branch master

Fetch origin Last fetched 13 minutes ago

Changes 3 History index.html

3 changed files

images\github-desktop-repository.gif

index.html

index.qmd

327 327 @@ -327,7 +327,7 @@

327 327 <section id="a-brief-introduction-to-git-and-github" class="title-slide slide level1 center" data-background-image="https://images.unsplash.com/photo-1618401479427-c8ef9465fbe?ixlib=rb-4.0.3&ixid=MnwXMjA3fDB8MHxwaG90by1wyWdlfHx8fGVufD88fHx8&auto=format&fit=crop&w=1686&q=80" data-background-color="black">

328 328 <h1>A Brief Introduction to Git and Github</h1>

329 329 <p>Michael Luu</p>

330 330 -<p>May 09, 2023</p>

330 330 +<p>May 10, 2023</p>

331 331 </section>

332 332

333 333 <section>

333 333 @@ -525,36 +525,37 @@

525 525 <section id="section-8" class="slide level2">

526 526 <h2></h2>

527 527

528 528 -</section></section>

529 529 -<section>

530 530 -<section id="creating-a-git-repository" class="title-slide slide level1 center">

531 531 -<h1>Creating a Git Repository</h1>

532 532 -

528 528 +<div class="footer">

529 529 +<p>https://github.com/mluu921/</p>

530 530 +</div>

531 531 </section>

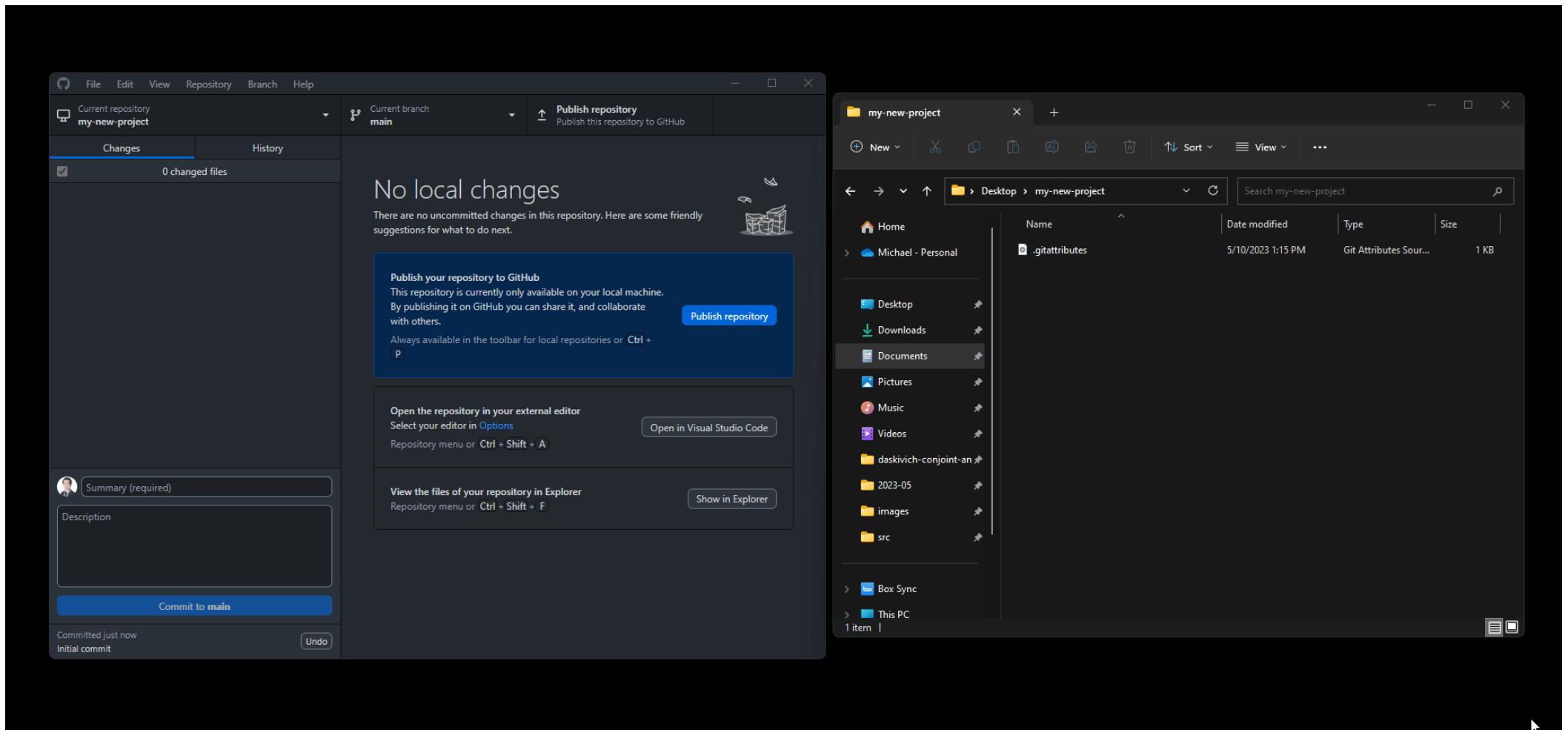
532 532 <section id="section-9" class="slide level2">

533 533 <h2></h2>

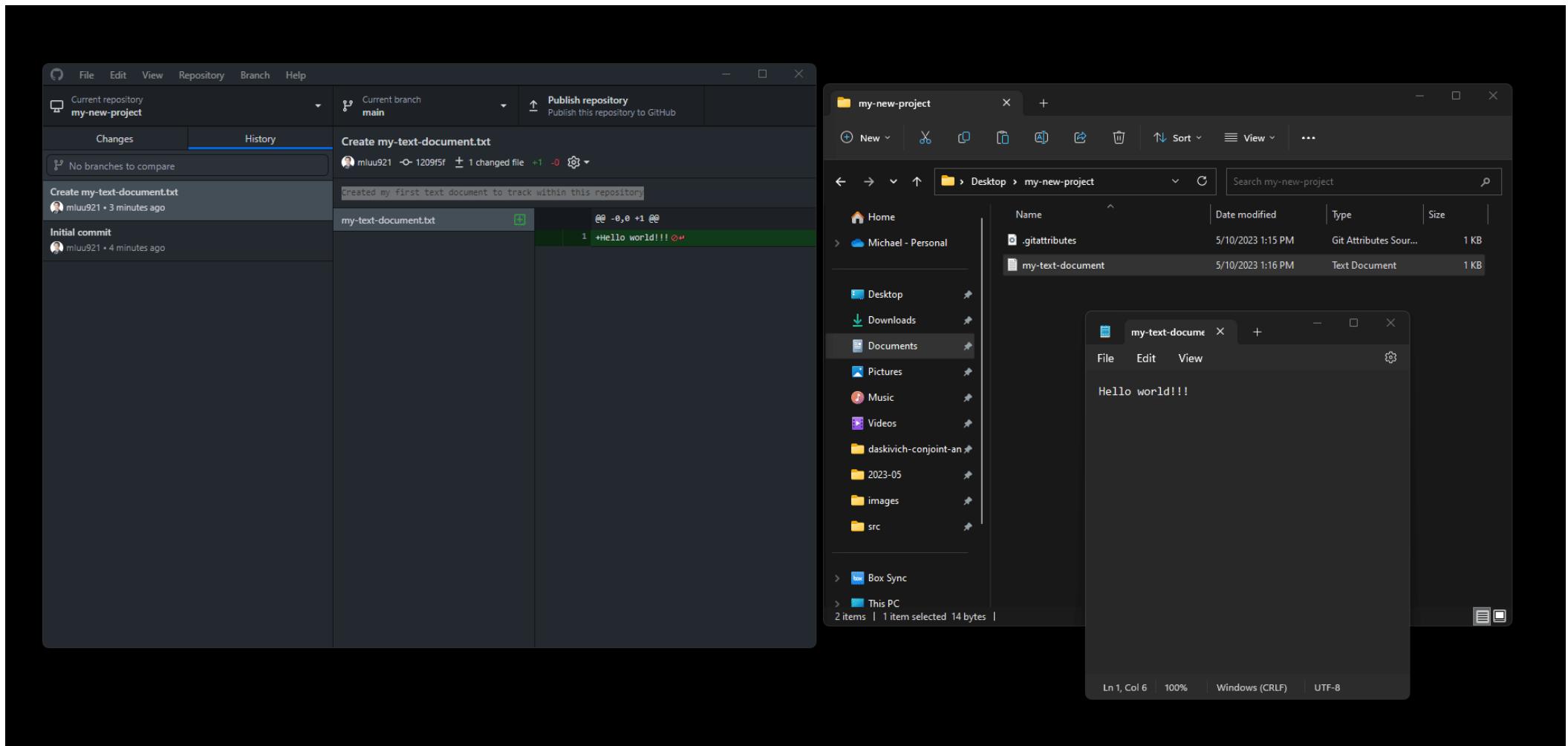
Commit to master

Staging and Committing

Making a commit

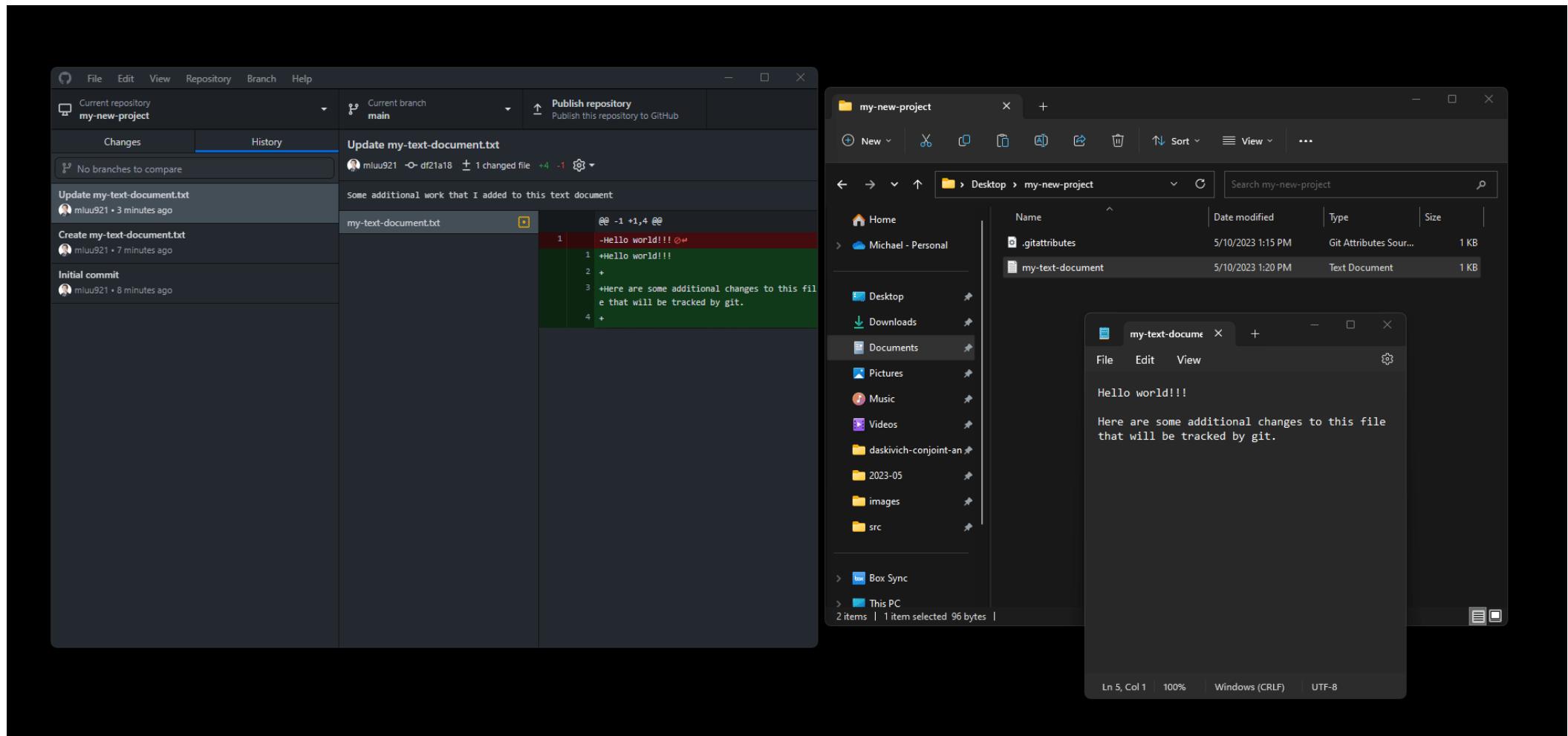


Making additional commits

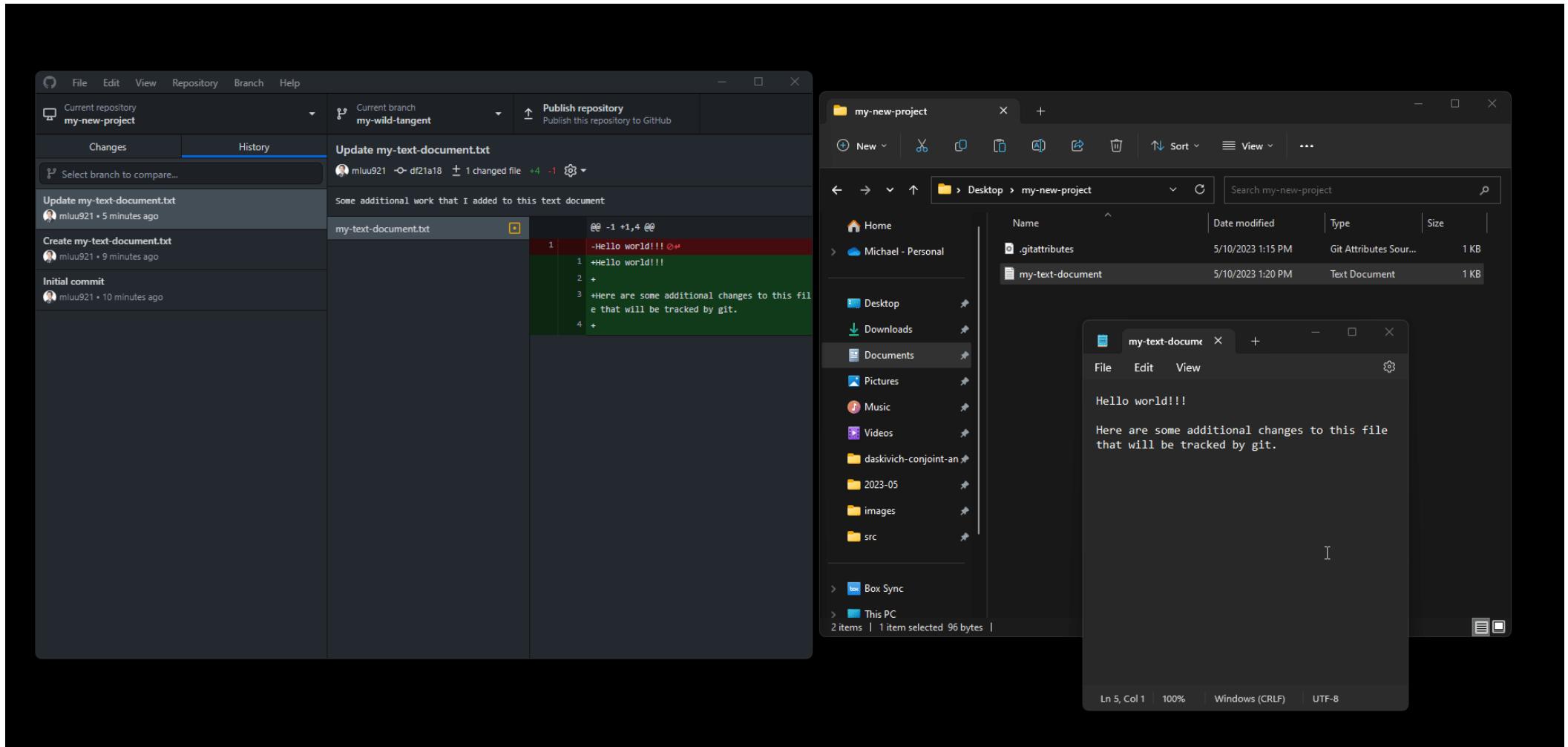


Branching and Merging

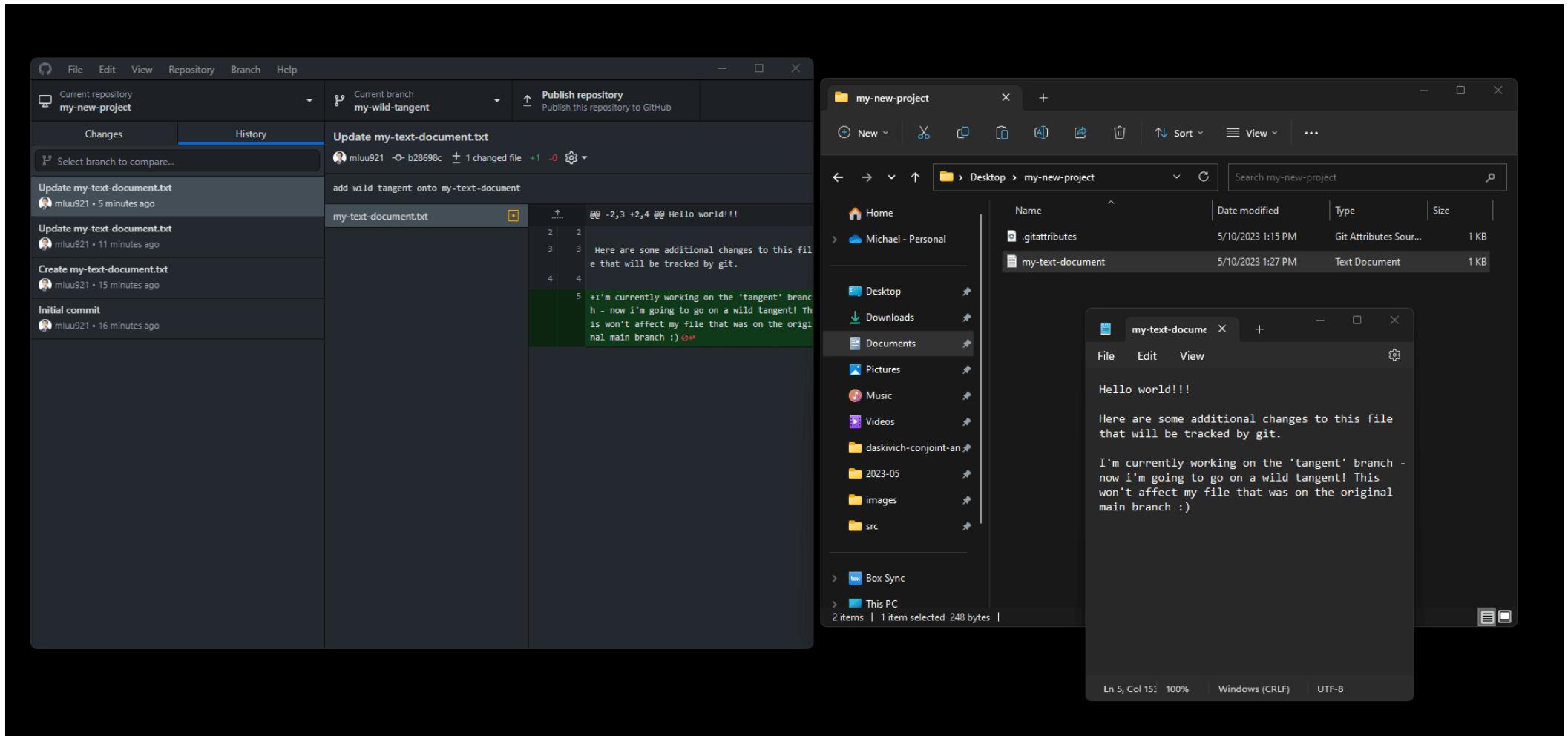
Creating and switching branch



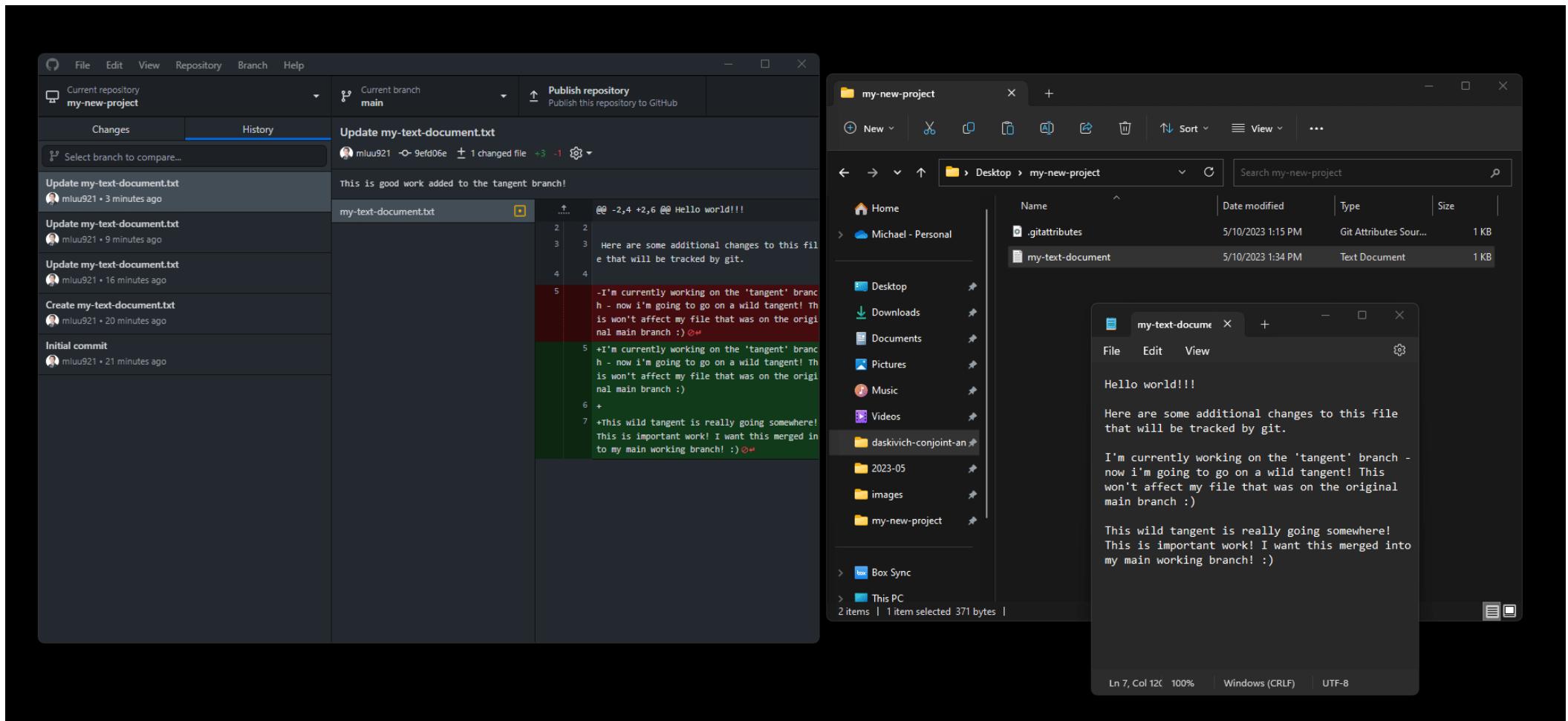
Making a commit on a branch



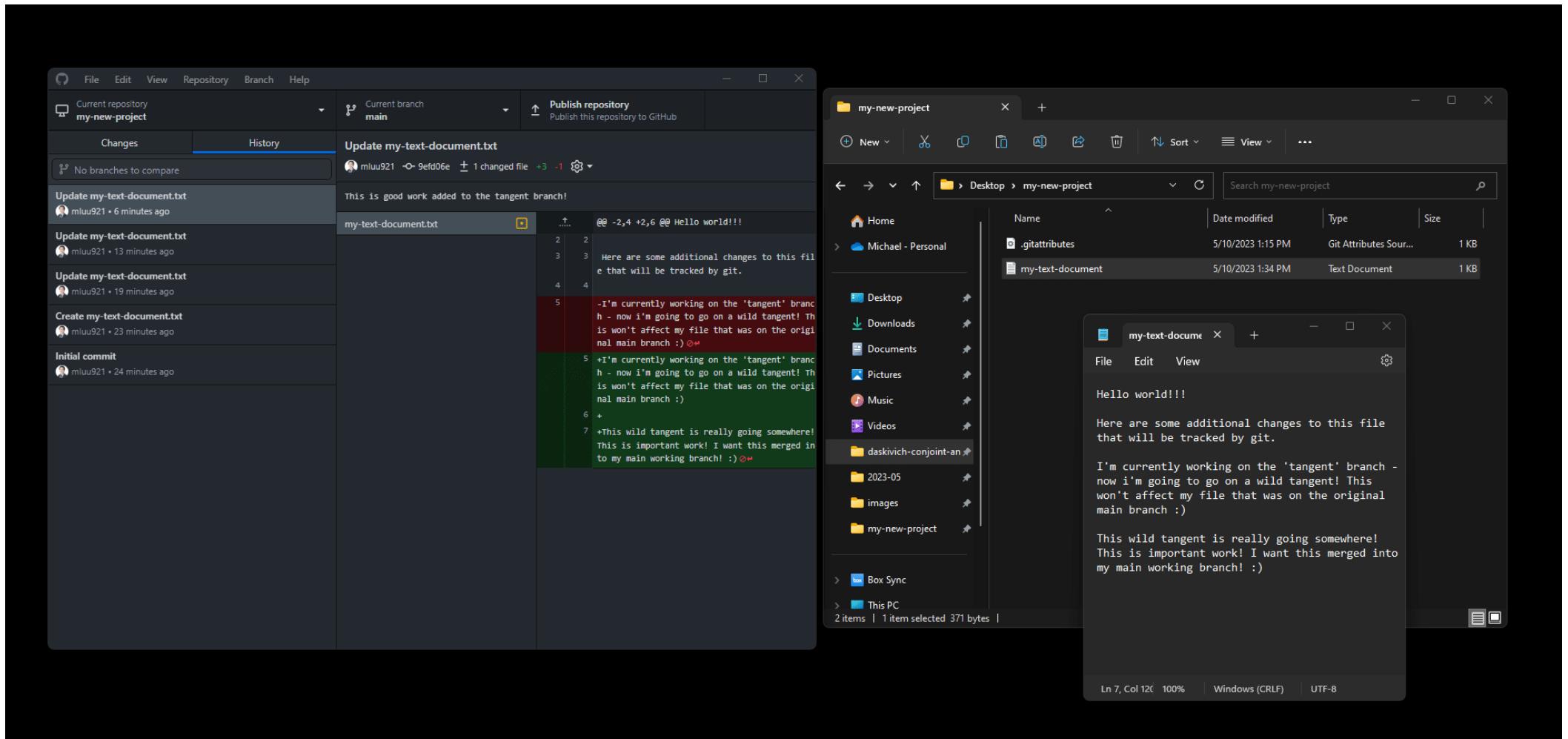
Merging branches



Deleting branches

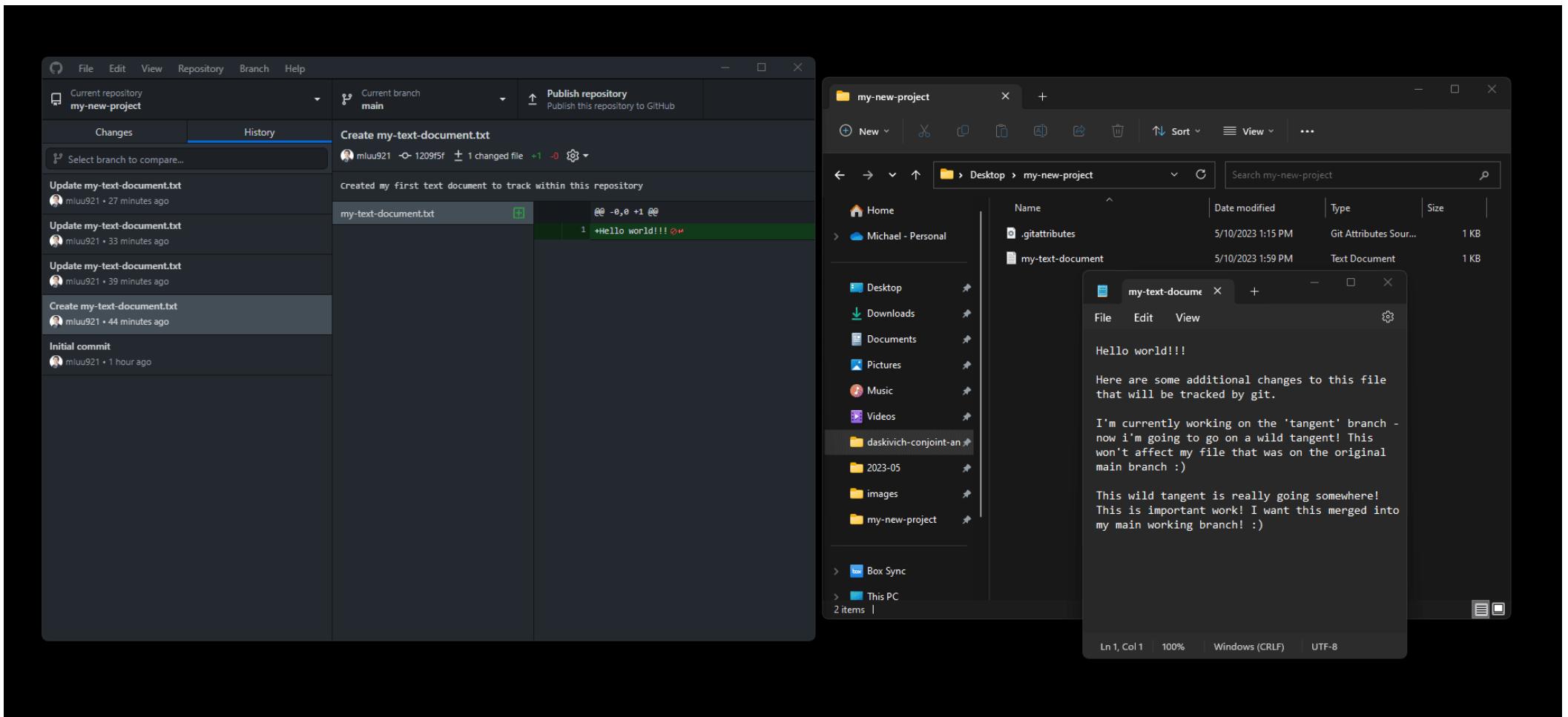


Creating a branch from a commit



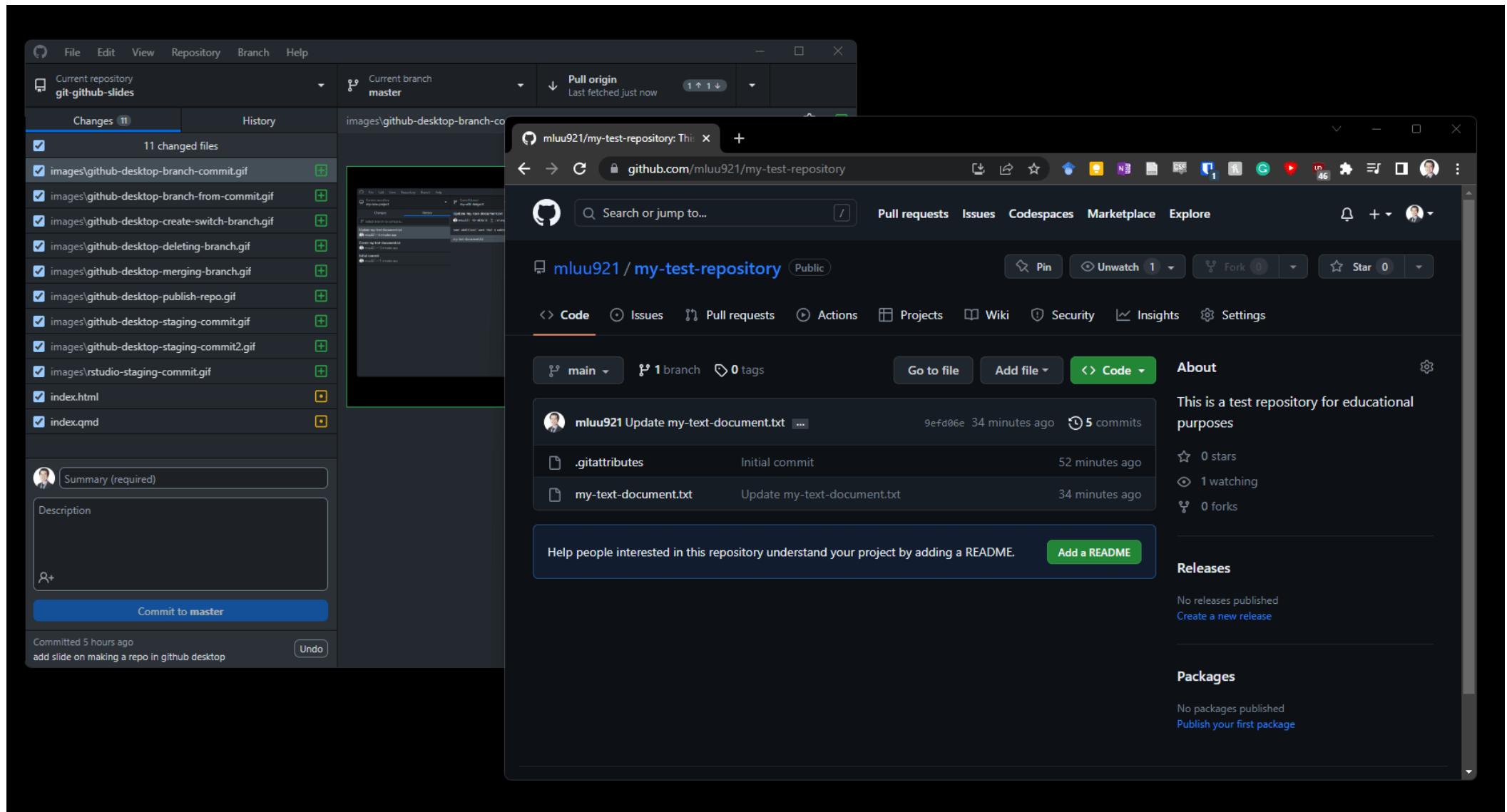
Publishing a repo

Publishing a repo



Cloning a repo

Cloning a repo



Cloning a repo

- Git repository of test repository
 - <https://github.com/mluu921/my-test-repository>
- Git repository of current slides
 - <https://github.com/mluu921/git-github-slides>

Additional Resources

- <https://happygitwithr.com/> (Git / Github Integration with Rstudio)
- <https://git-scm.com/book/en/v2> (Definitive Git Book)

