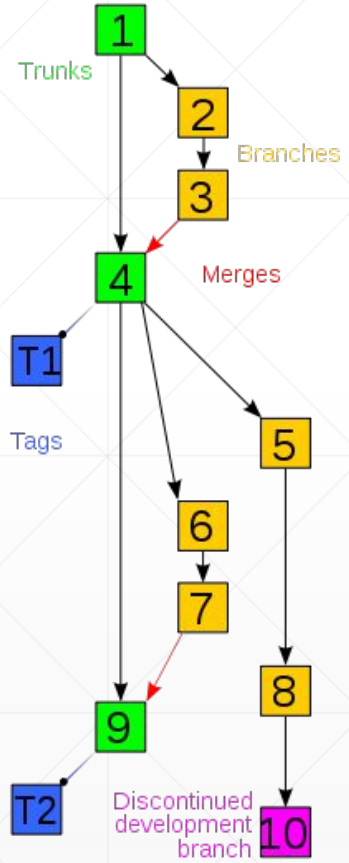
GEOG-312
First Steps in GIS Programming

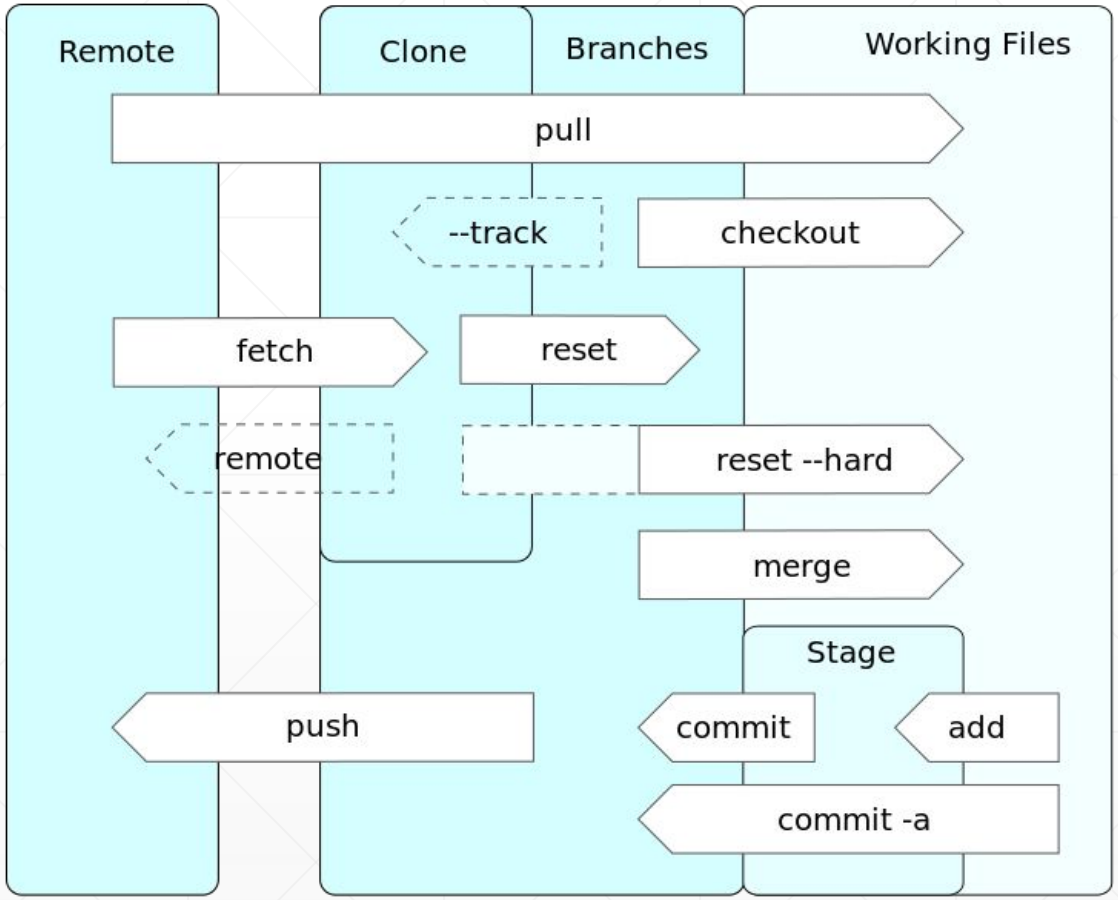
Introduction to GitHub

Instructor: Qiusheng Wu, Ph.D.
Department of Geography
University of Tennessee
https://wetlands.io

What is Git?

- Git is a distributed version control system (VCS)

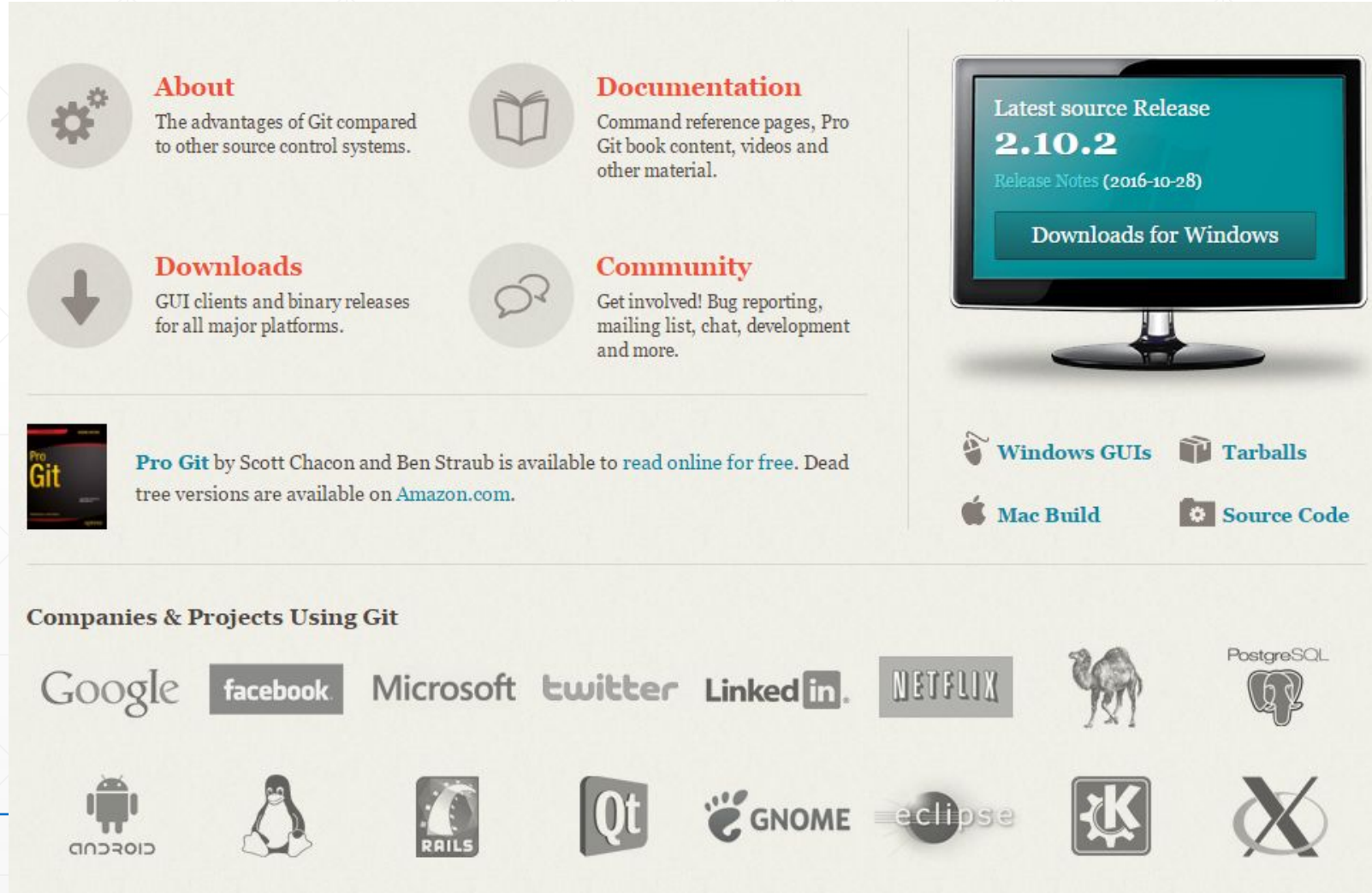




https://en.wikipedia.org/wiki/Git

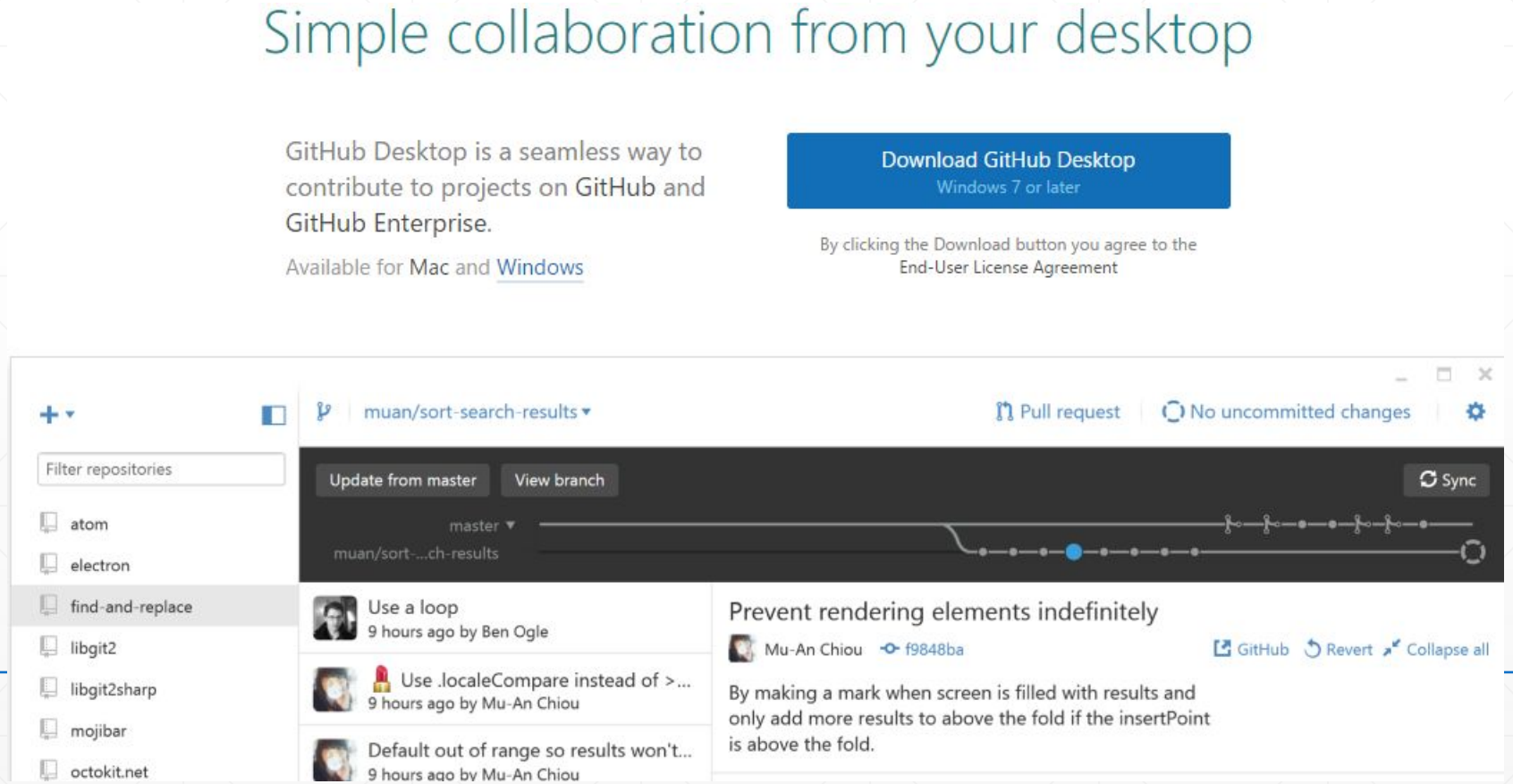
Install Git

- Download Git from https://git-scm.com/



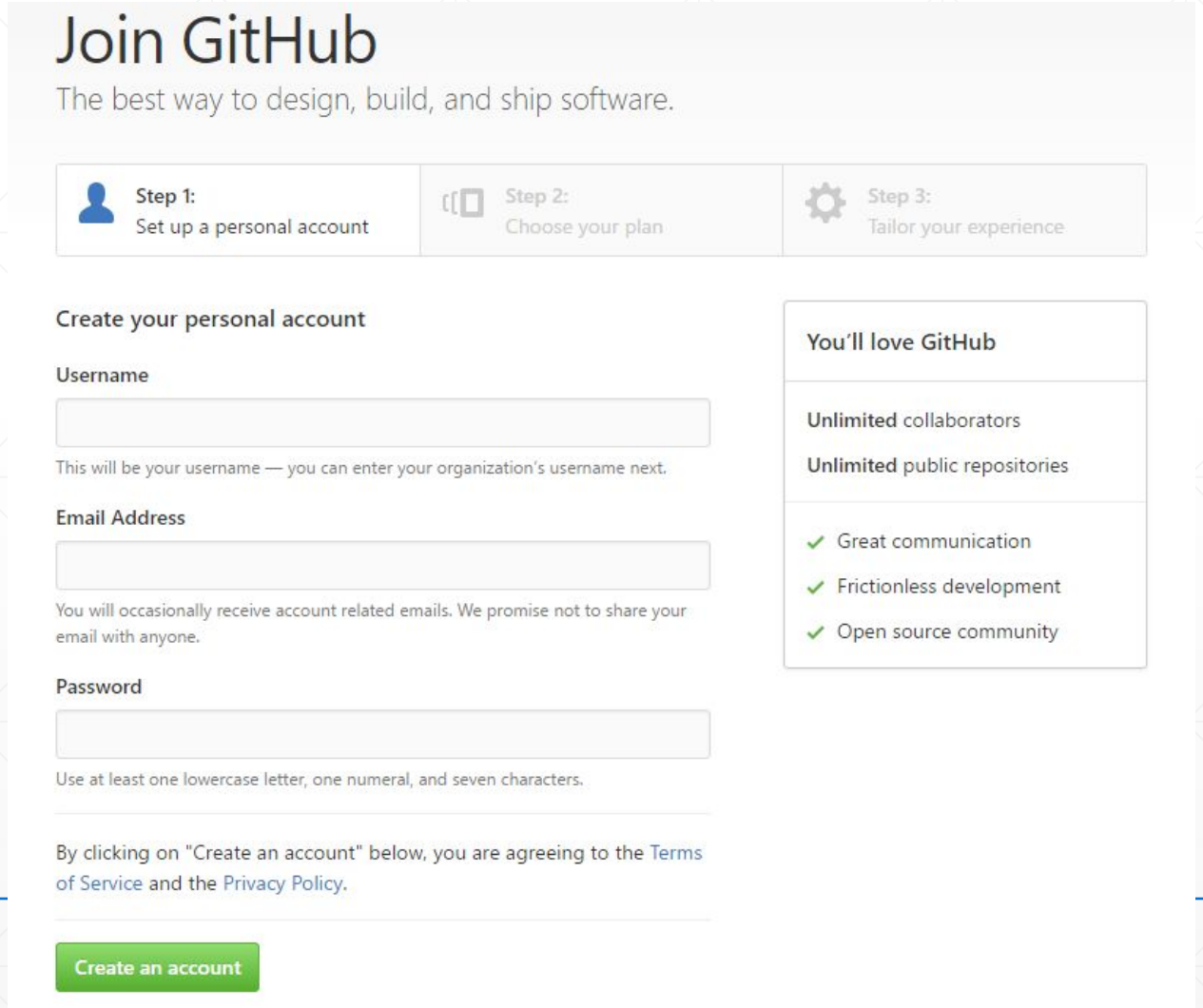
What is GitHub?

- GitHub is a web-based Git repository hosting service.
  - GitHub: https://github.com/
  - GitHub Desktop: https://desktop.github.com/



Sign up a GitHub account

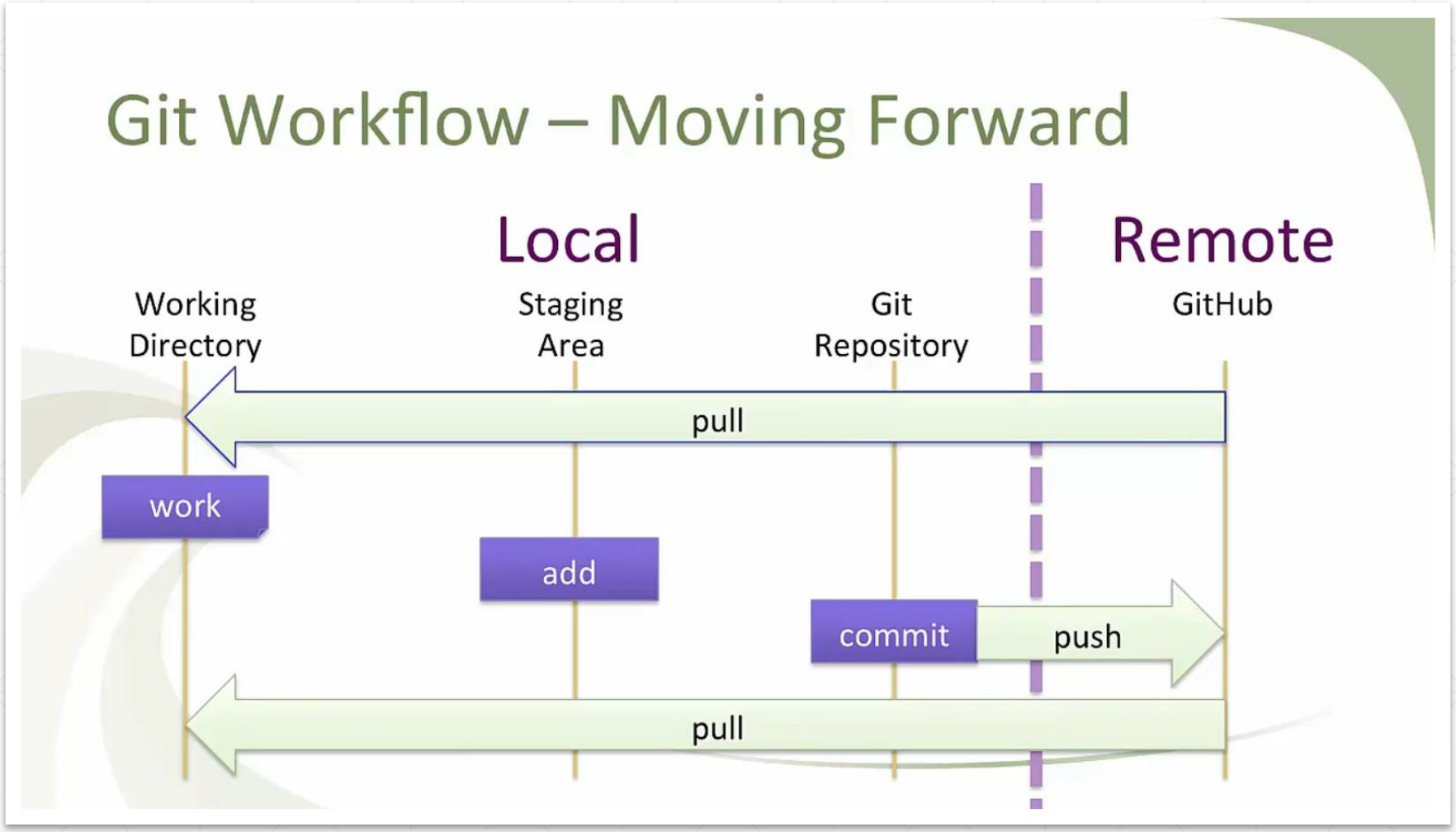
- https://github.com/join



Key Git concepts

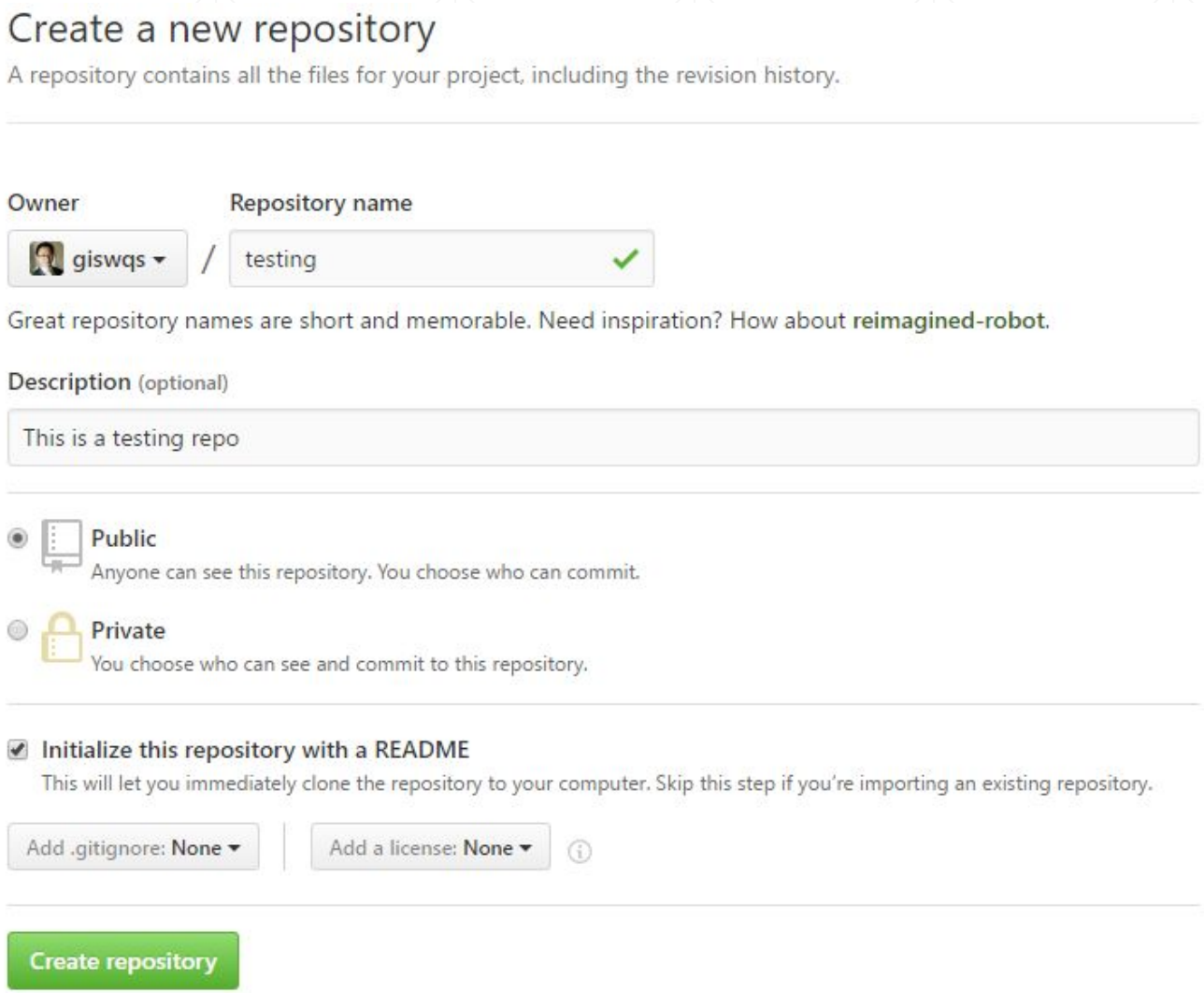
- Repository contains files, history, config managed by Git
- Three States of Git
  - Working directory
  - Staging area – pre-commit holding area
  - Commit – Git Repository (history)
- Remote repository (GitHub)
- Master branch

Git Workflow



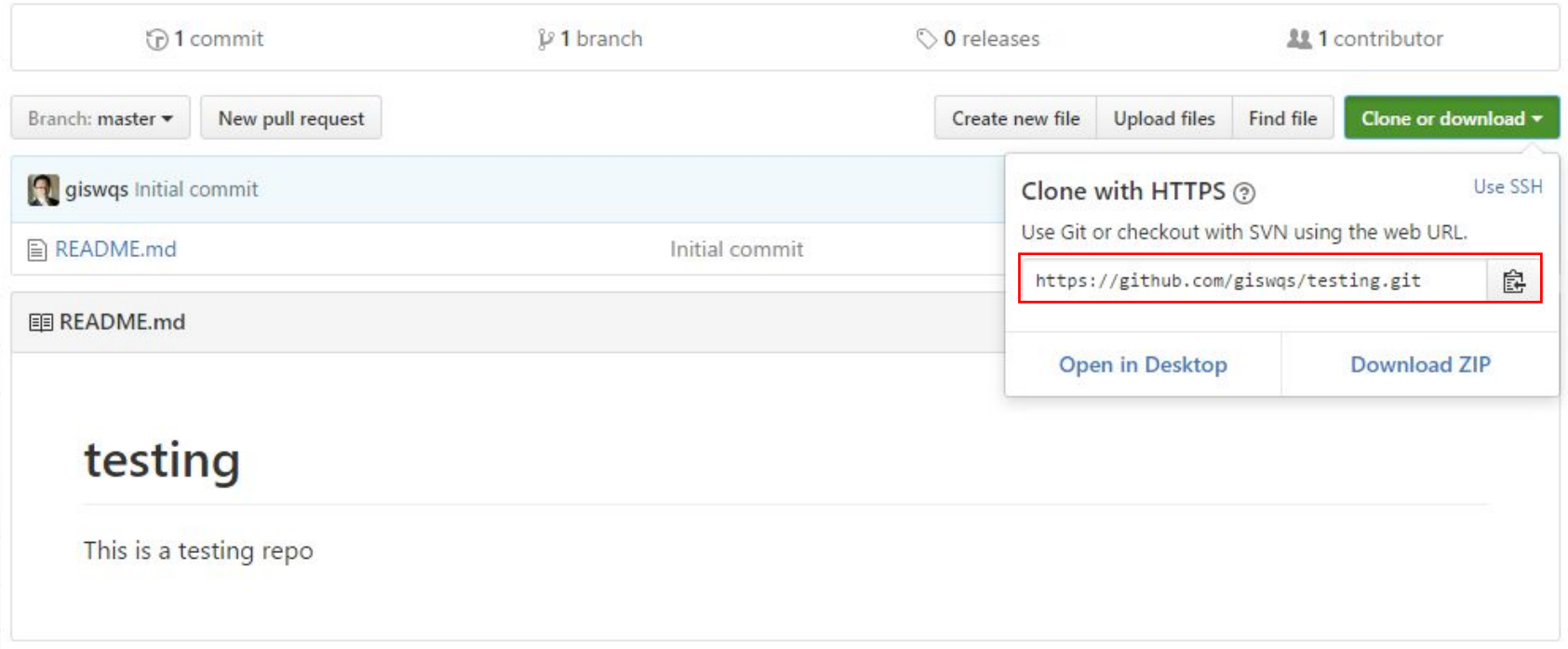
Create a new repository

- https://github.com/new



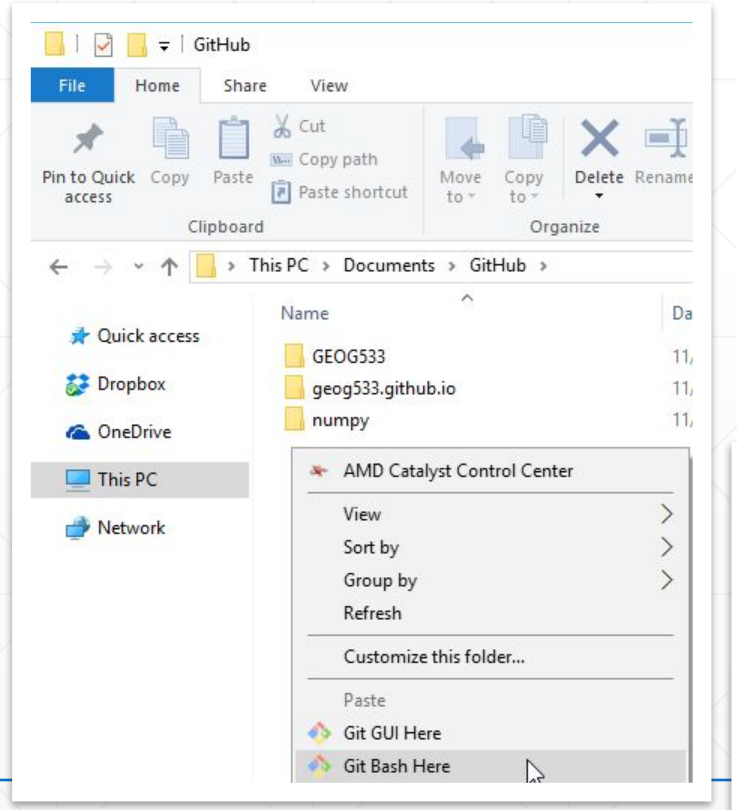
Clone a repository

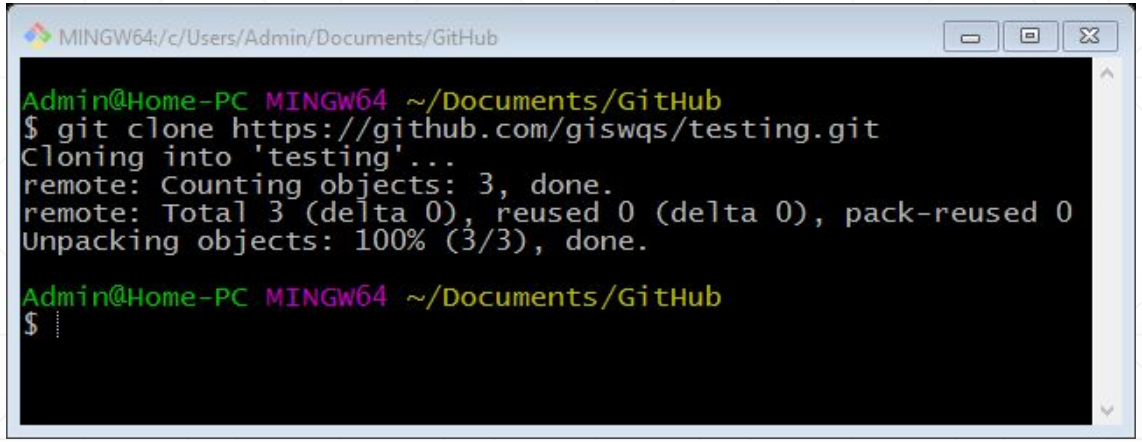
- Clone a repo from GitHub to your local computer

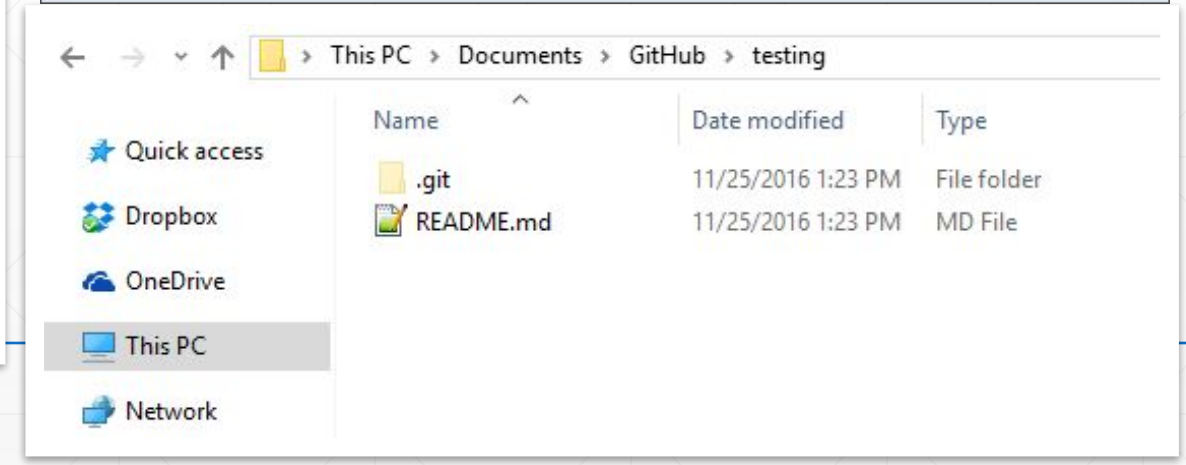


Clone a repository

- Steps
  - Open **Git Bash**
  - Enter command: ***git clone https://github.com/giswqs/testing.git***

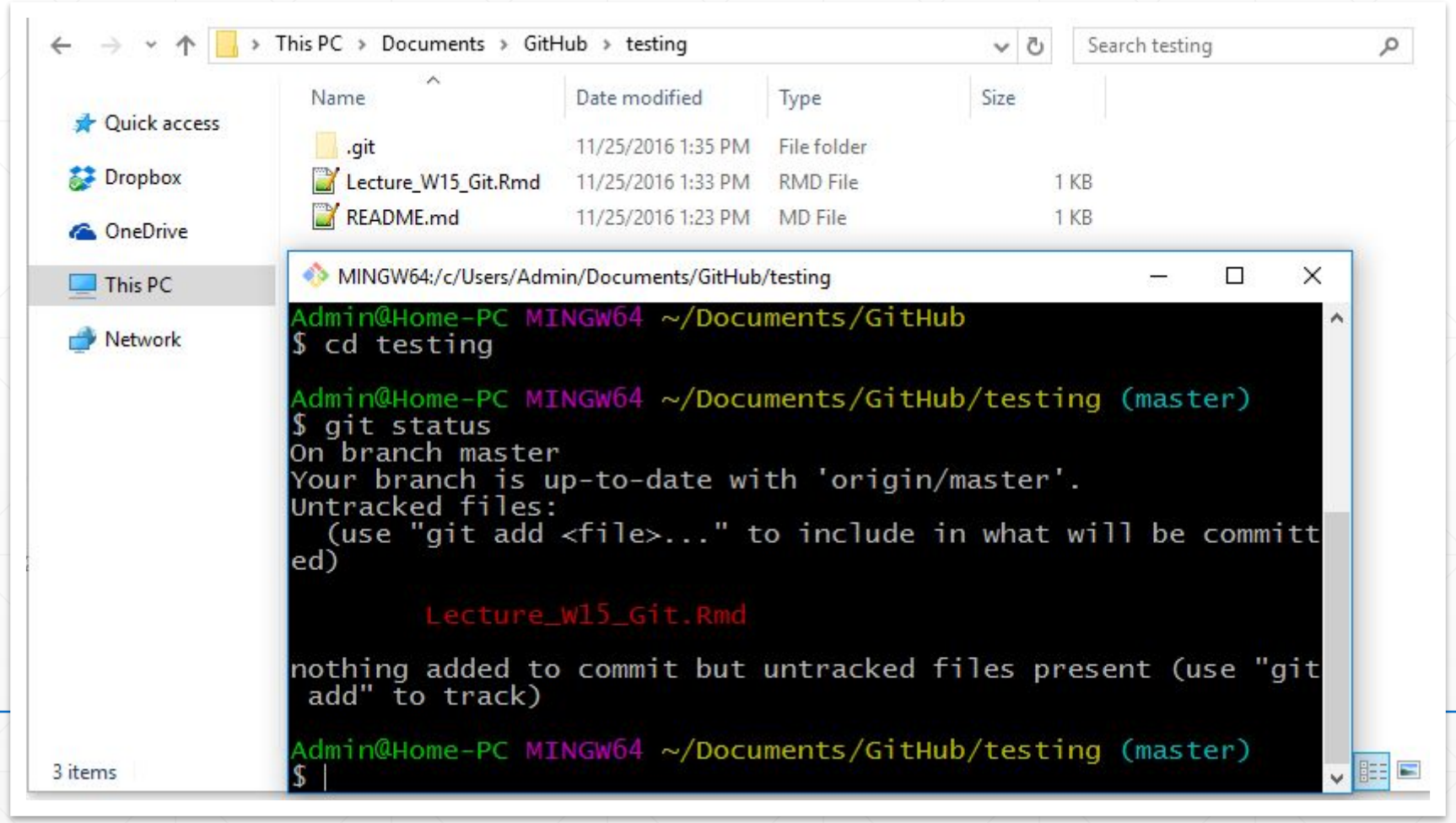






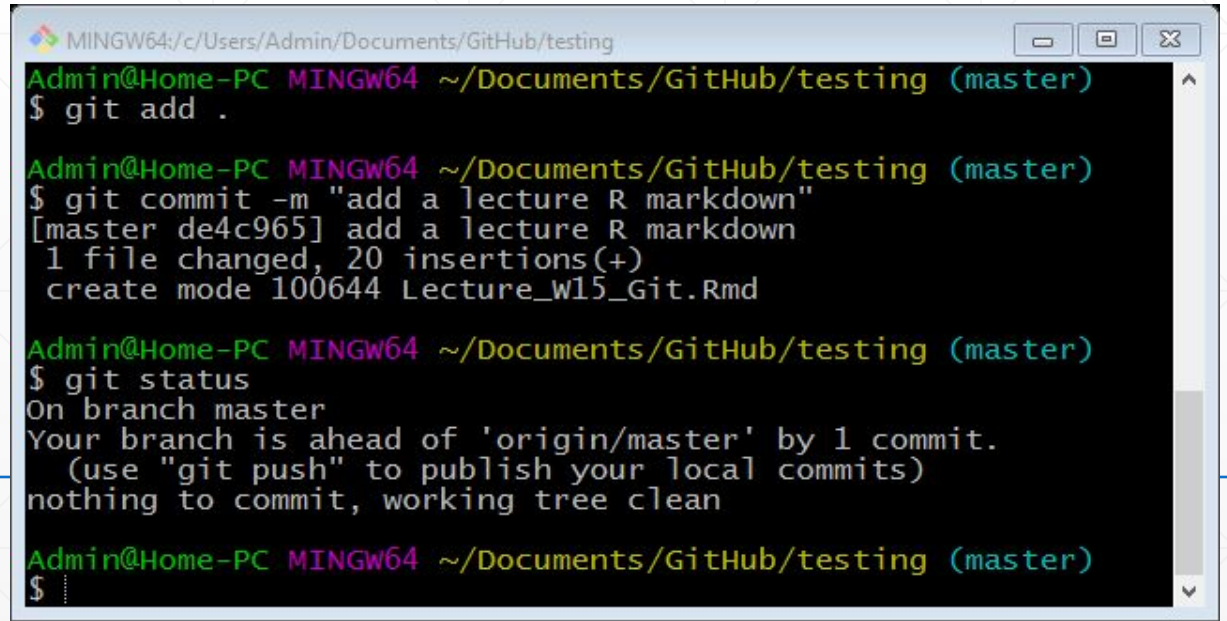
Add files to local repo

- Add any files to the local repo folder
- Use command ***git status*** to check changes



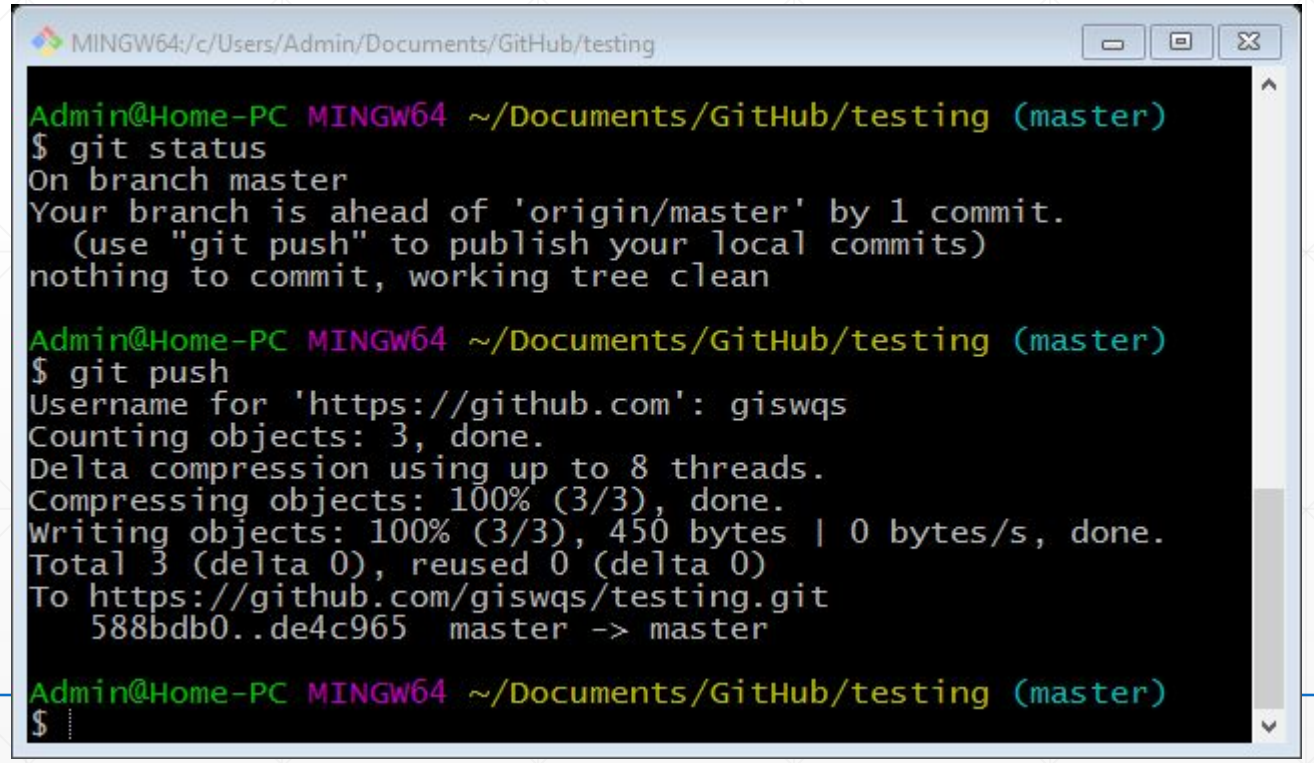
Commit the changes

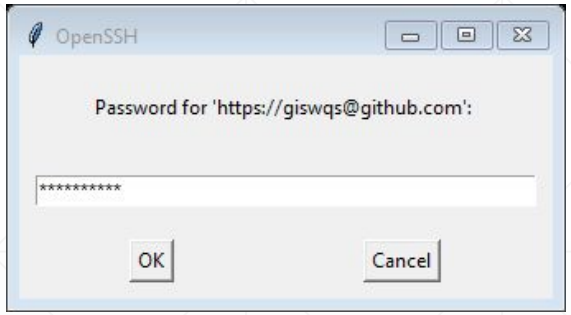
- Add files to Git staging area
  - ***git add .*** or ***git add "[filename]"***
- Commit the staged content as a new commit snapshot
  - ***git commit -m "[descriptive message]"***
- Check status
  - ***git status***



Push changes to GitHub

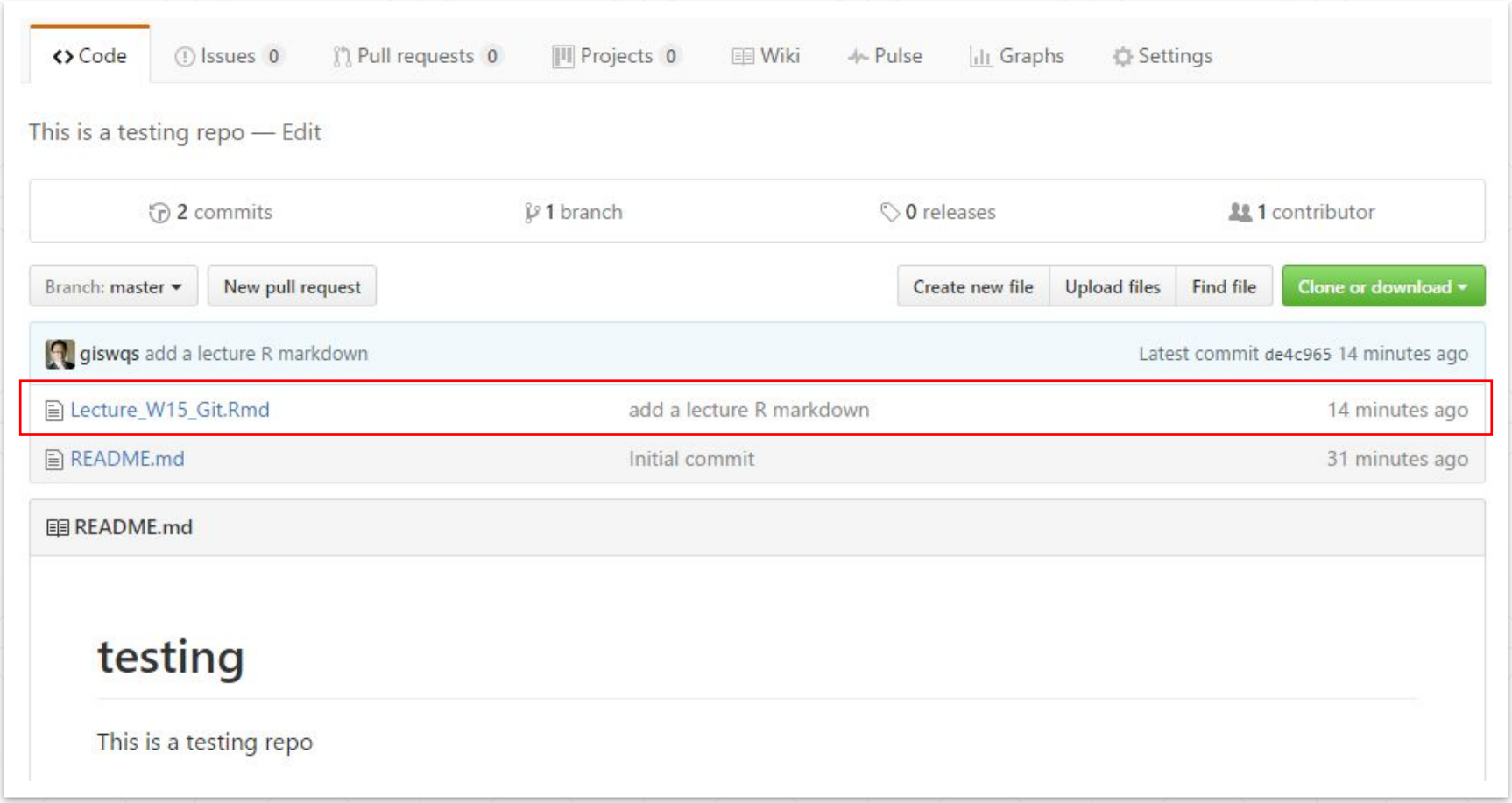
- Push changes to GitHub
  - ***git push***
  - Optional: ***git config credential.helper store*** (no need to enter GitHub username and password every time)





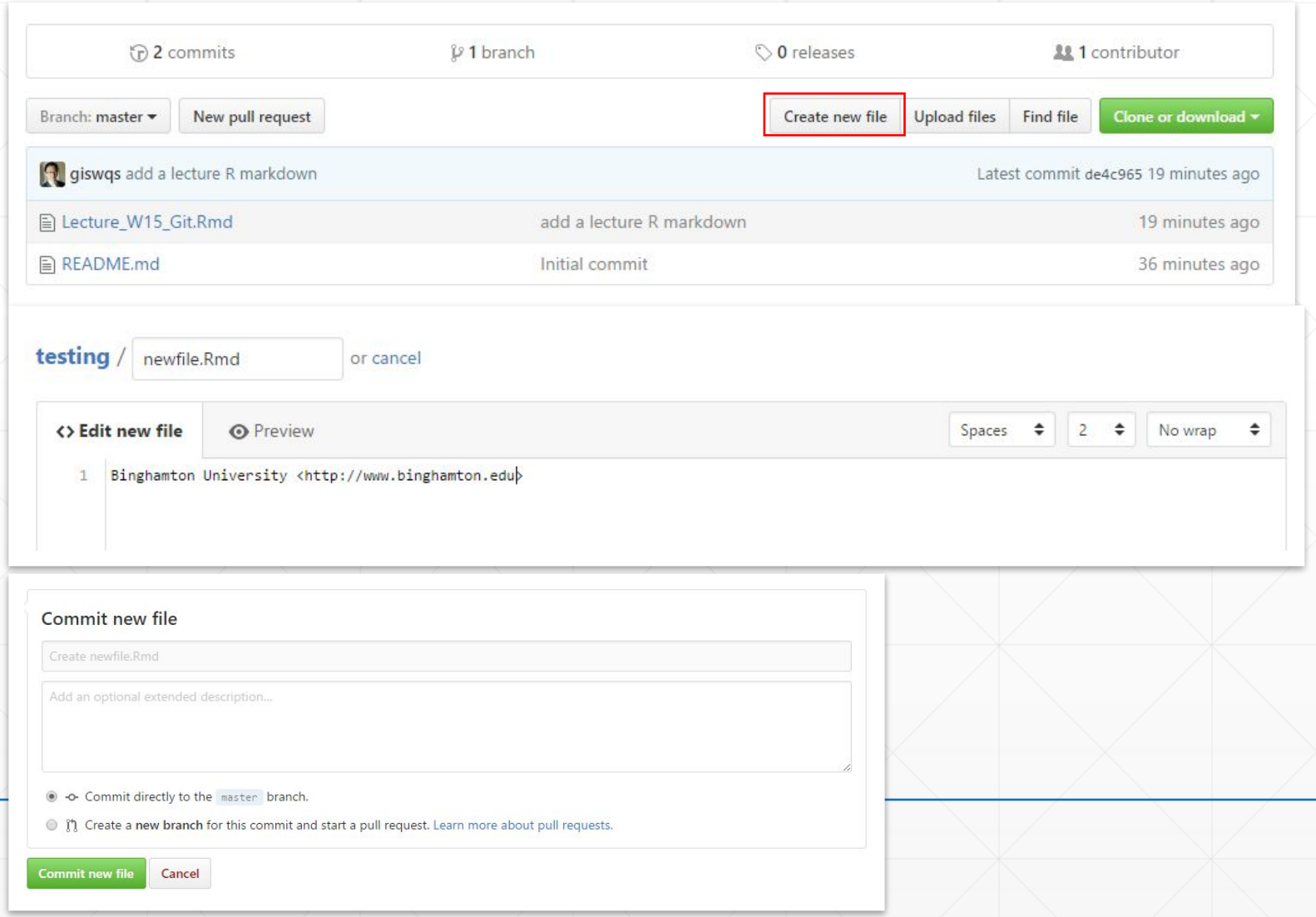
Check changes on GitHub

- https://github.com/giswqs/testing



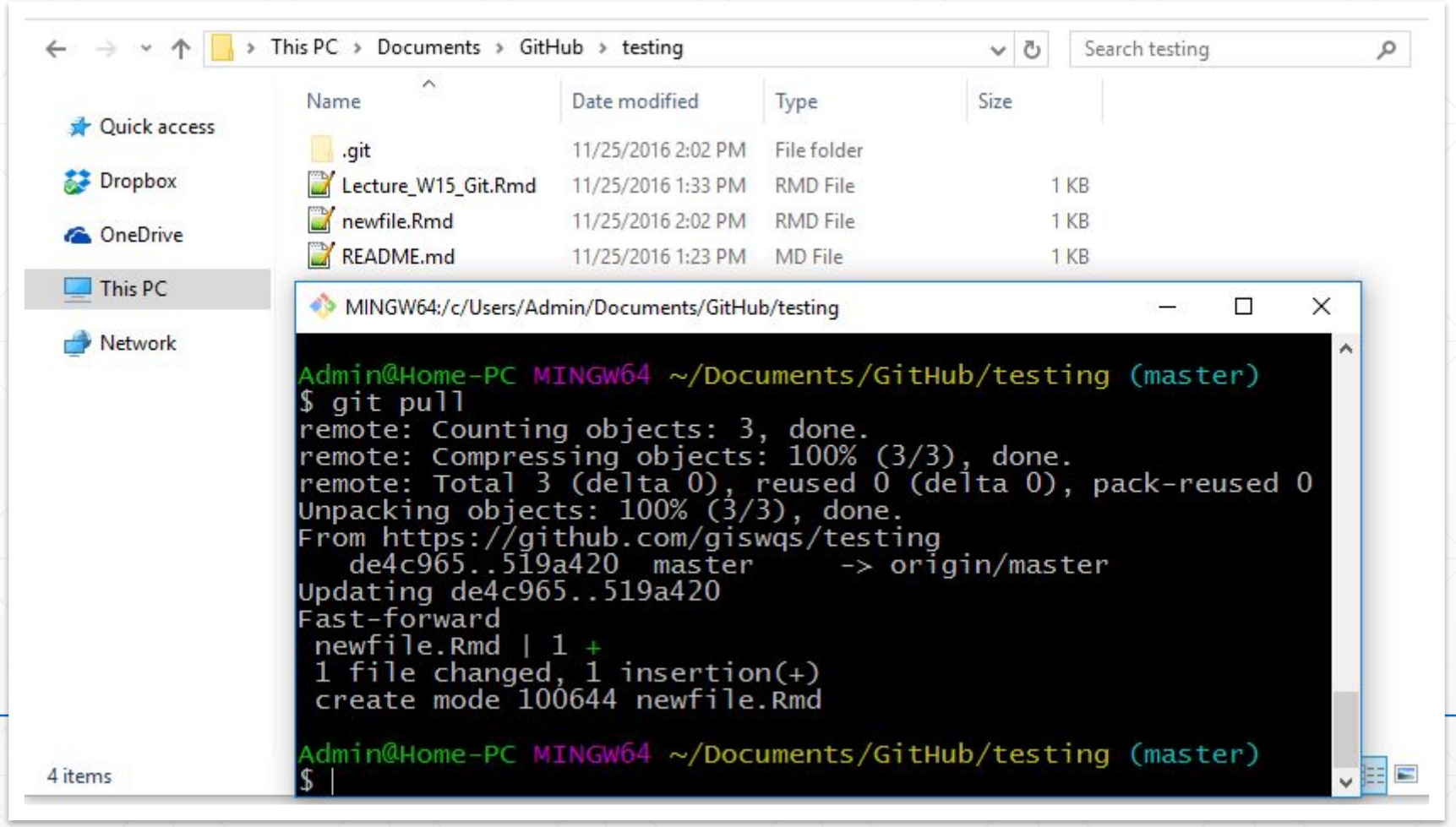
Make changes on GitHub

- Create a new file



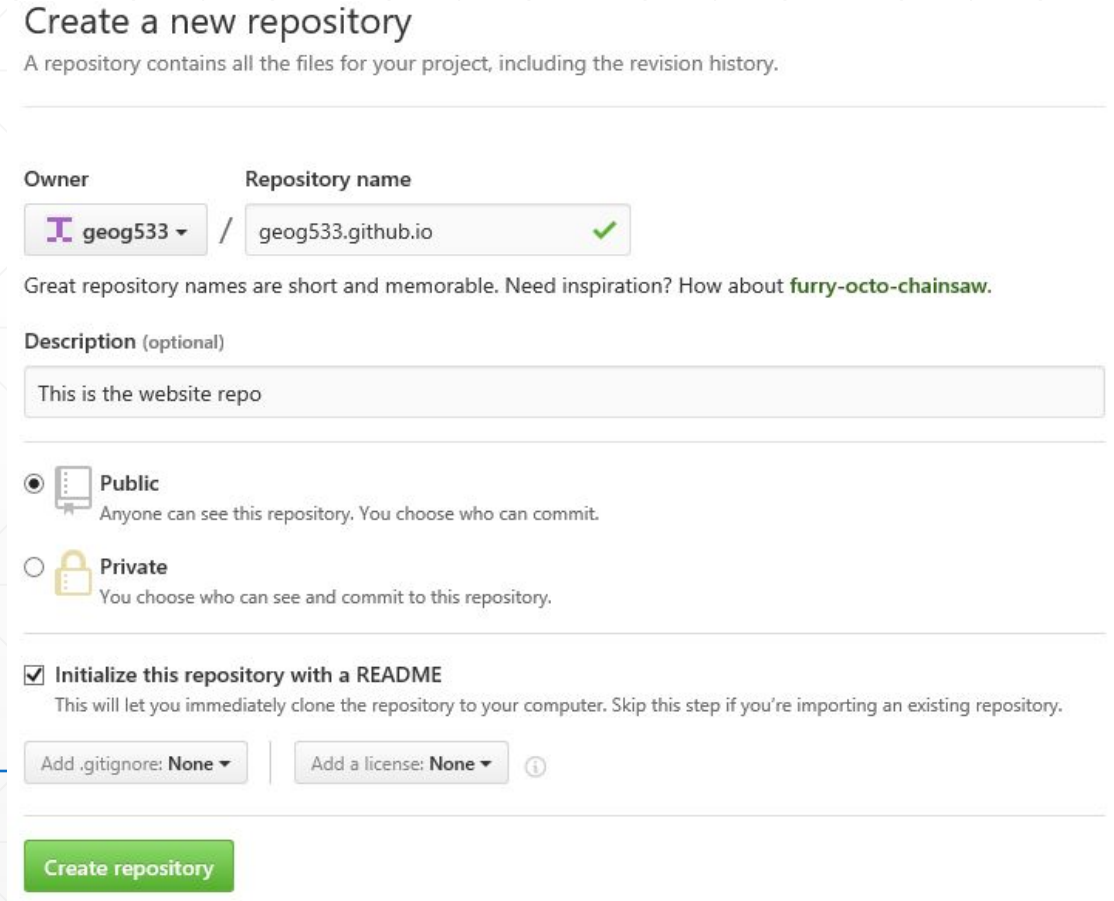
Pull GitHub changes to local repo

- Pull GitHub changes to local repo
  - ***git pull***



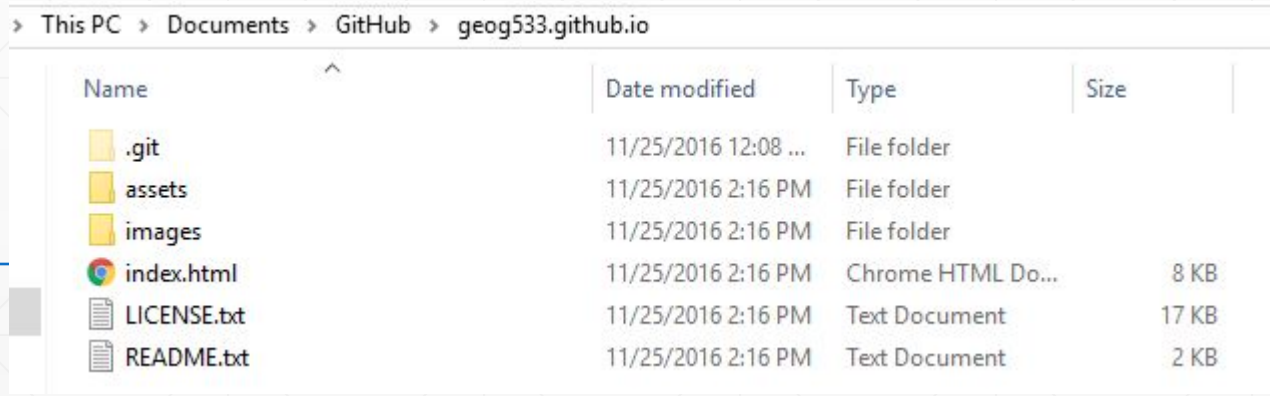
GitHub Pages

- Create your website and host it on GitHub
  - https://pages.github.com/
- Create a new repo: ***[username].github.io***



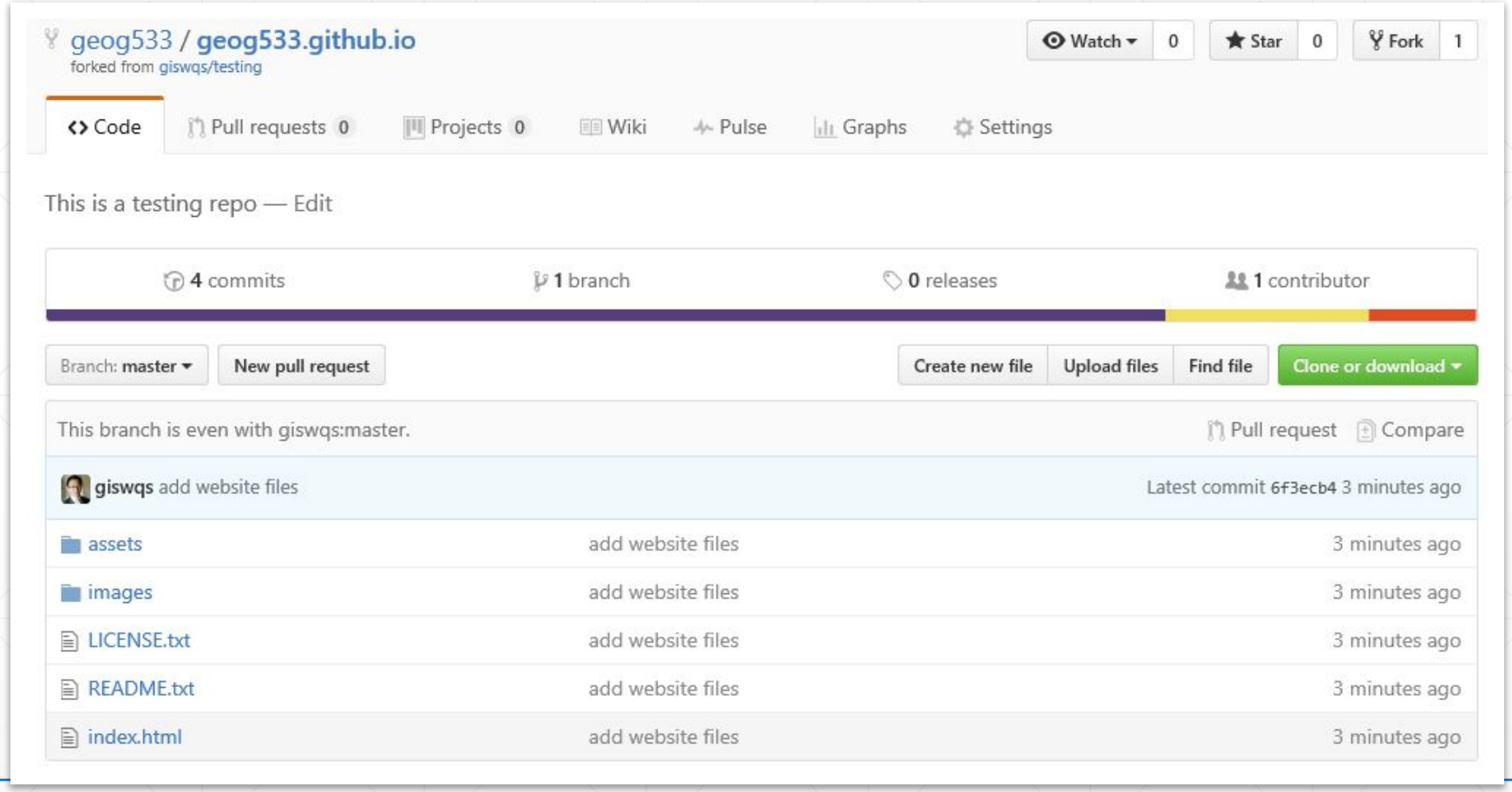
Website template

- Clone the GitHub repo to local PC
  - ***git clone https://github.com/geog-312/geog-312.github.io.git***
- Download a website template and copy files to the repo
  - https://html5up.net/
- Commit changes
  - ***git add .***
  - ***git commit -m "add website files"***
- Push changes to GitHub
  - ***git push***



Preview the website

- https://giswqs.github.io/geog-312



Git Cheat Sheet

➢ https://education.github.com/git-cheat-sheet-education.pdf

SETUP

Configuring user information used across all local repositories

`git config --global user.name "[firstname lastname]"`
set a name that is identifiable for credit when review version history
`git config --global user.email "[valid-email]"`
set an email address that will be associated with each history marker
`git config --global color.ui auto`
set automatic command line coloring for Git for easy reviewing

SETUP & INIT

Configuring user information, initializing and cloning repositories

`git init`
initialize an existing directory as a Git repository
`git clone [url]`
retrieve an entire repository from a hosted location via URL

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

`git status`
show modified files in working directory, staged for your next commit
`git add [file]`
add a file as it looks now to your next commit (stage)
`git reset [file]`
unstage a file while retaining the changes in working directory
`git diff`
diff of what is changed but not staged
`git diff --staged`
diff of what is staged but not yet committed
`git commit -m "[descriptive message]"`
commit your staged content as a new commit snapshot

Git Cheat Sheet

➢ https://education.github.com/git-cheat-sheet-education.pdf

BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

`git branch`
list your branches. a * will appear next to the currently active branch
`git branch [branch-name]`
create a new branch at the current commit
`git checkout`
switch to another branch and check it out into your working directory
`git merge [branch]`
merge the specified branch's history into the current one
`git log`
show all commits in the current branch's history

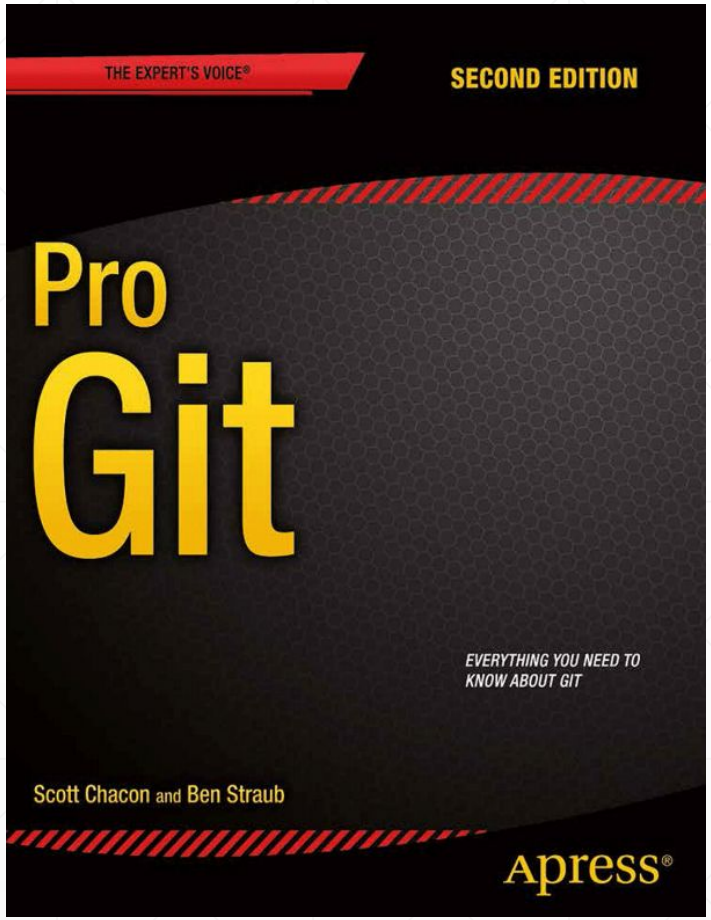
SHARE & UPDATE

Retrieving updates from another repository and updating local repos

`git remote add [alias] [url]`
add a git URL as an alias
`git fetch [alias]`
fetch down all the branches from that Git remote
`git merge [alias]/[branch]`
merge a remote branch into your current branch to bring it up to date
`git push [alias] [branch]`
Transmit local branch commits to the remote repository branch
`git pull`
fetch and merge any commits from the tracking remote branch

Git book

- https://git-scm.com/book/en/v2



Try Git

- https://try.github.io/