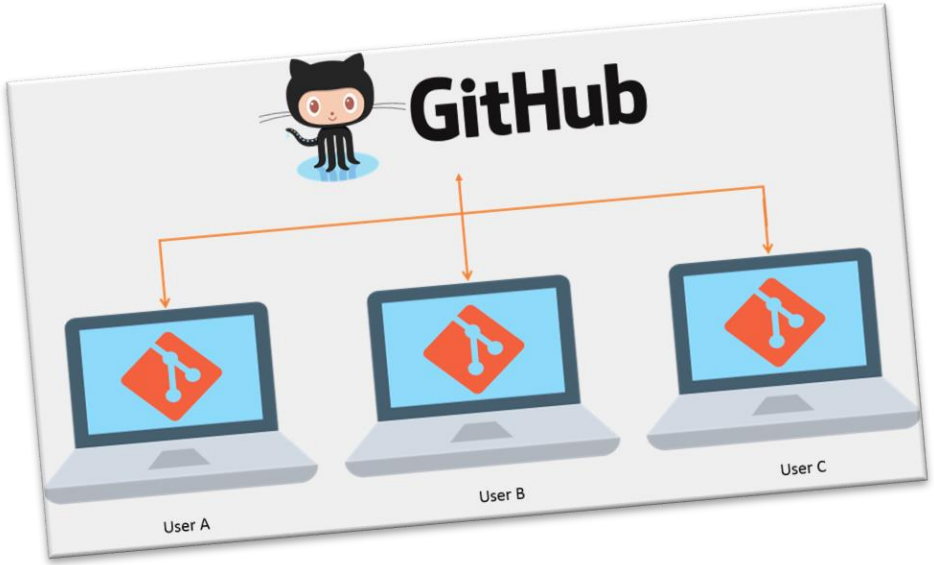


Version Control: Using Git and GitHub

Chen Chiu

Website: dataservices.library.jhu.edu

Email: dataservices@jhu.edu

Johns Hopkins Research Data Repository: archive.data.jhu.edu

JOHNS HOPKINS LIBRARIES

Data Services

What Do You Need?

GitHub account

- Create a free GitHub account

Software installation

- GitHub Desktop https://desktop.github.com/

JOHNS HOPKINS
LIBRARIES

Data Services

Workshop Topics

- Version control (What is it and Why is it important?)
- Git terminology and basic concepts
- Set up GitHub Desktop
- GitHub Desktop demo
- Resources

git + github SOCIAL CODING

What is Version Control

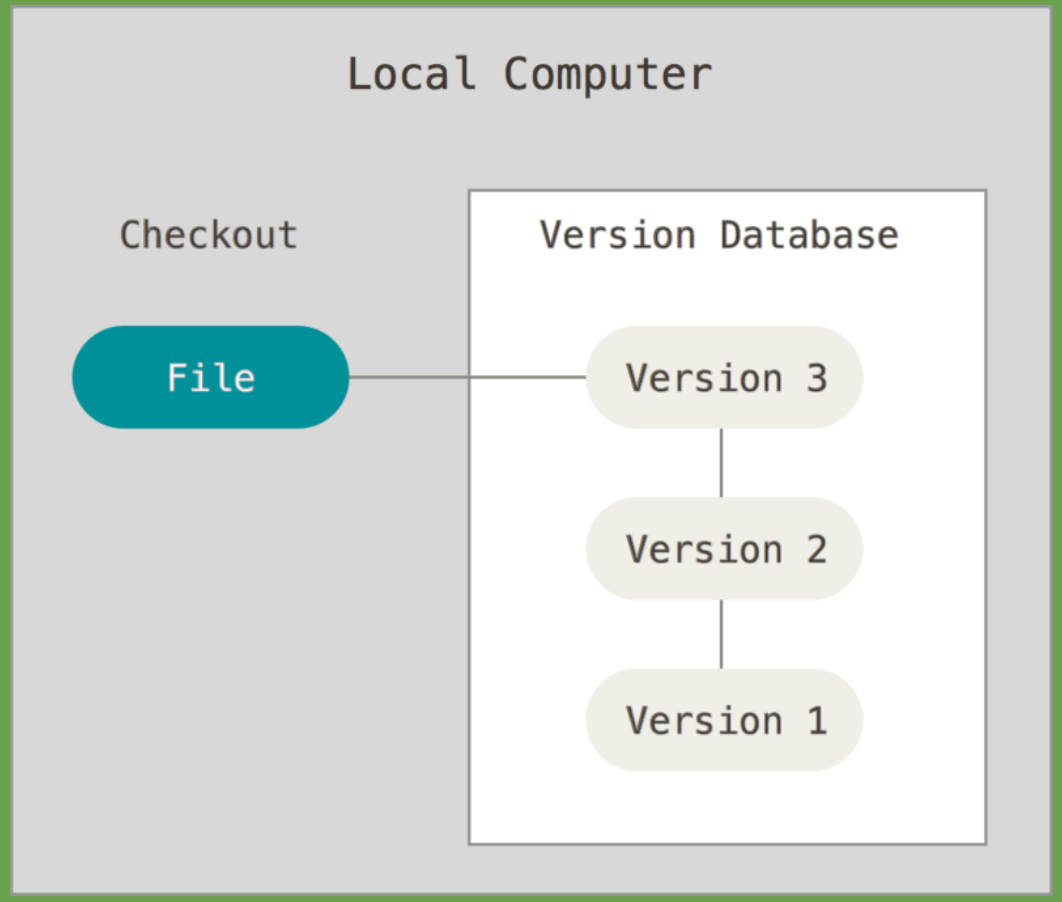
Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

TOP VERSION CONTROL SYSTEMS

https://www.linuxnix.com/what-are-the-top-version-control-systems/

Local Version Control Systems



How it works:

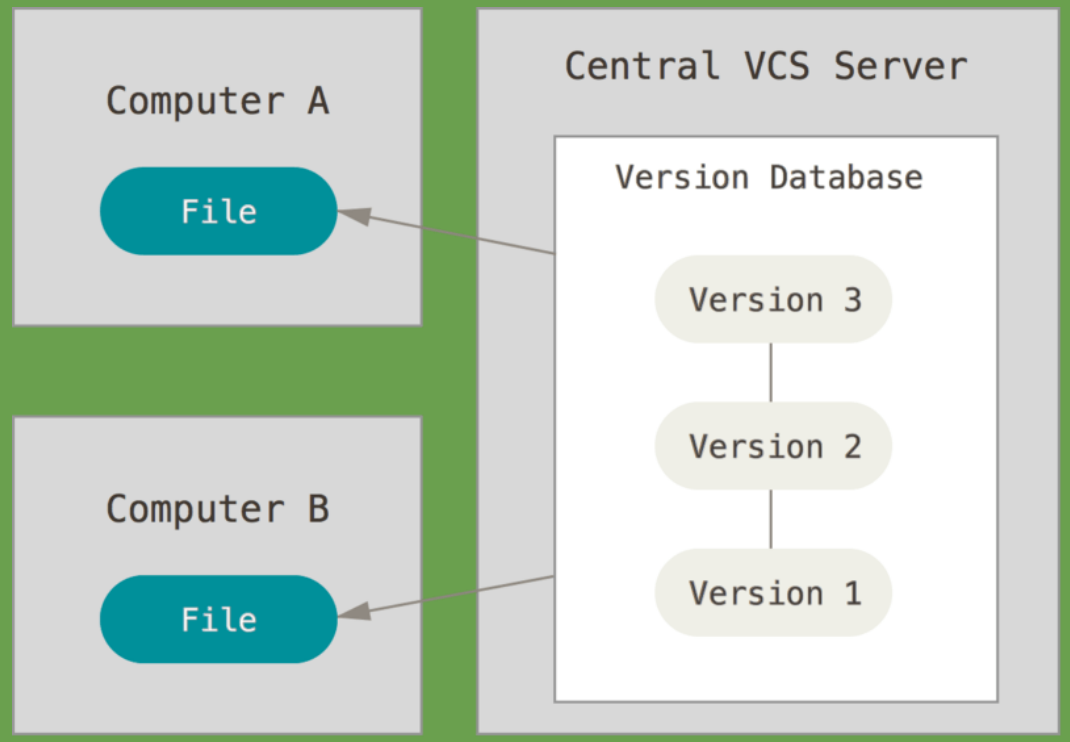
- Keep a local database of versions
- Check out the latest version to work on

Possible issues:

- It is hard for more than one person to work on a file
- All versions are stored locally

https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Centralized Version Control Systems



How it works:

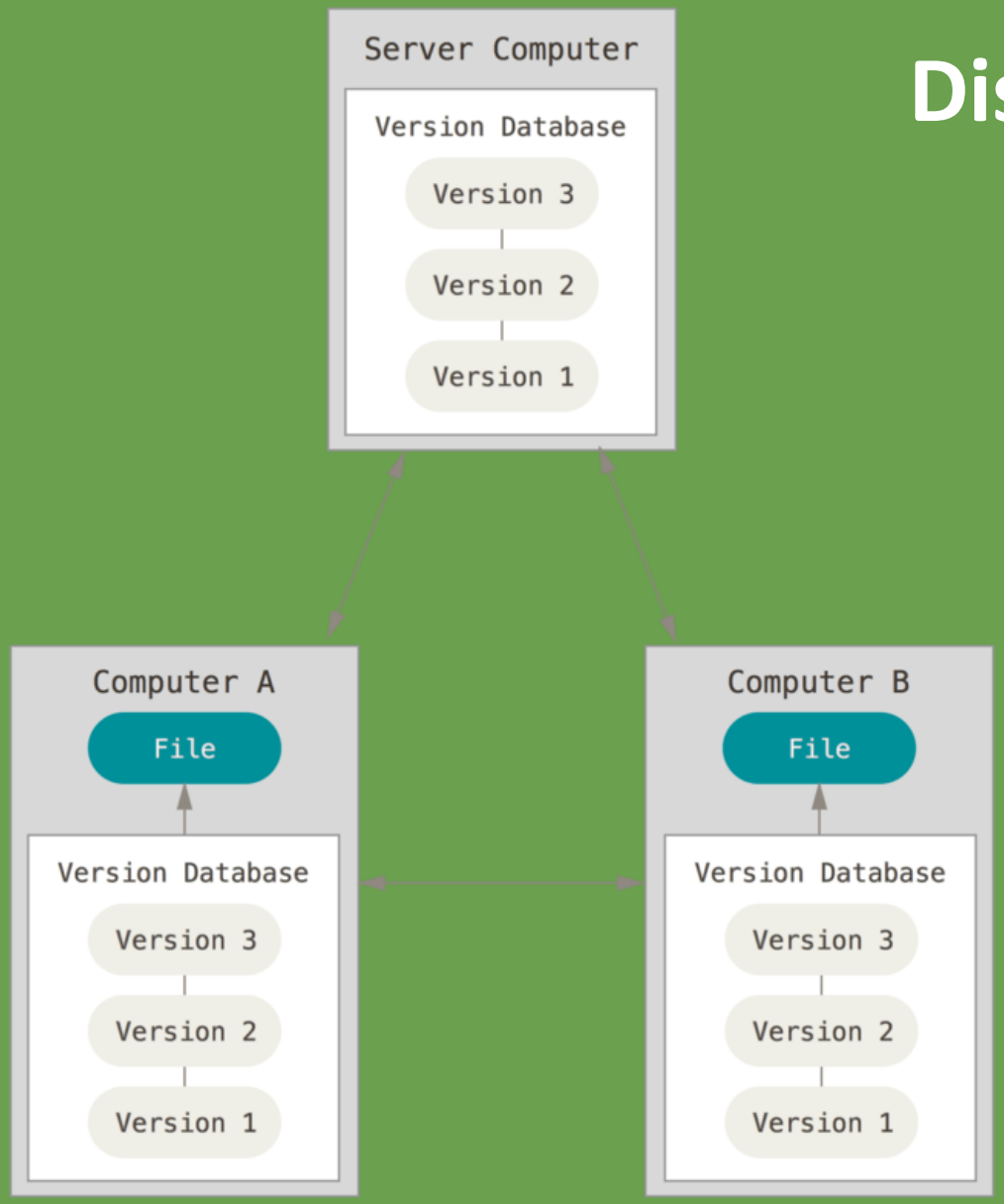
- All versions are stored in a central server
- You can check out files from the server to work on them
- Has been around for many years
- Examples: Subversion, CVS

Possible issues:

- When the server is down, you cannot work on the files
- If there is no proper backup for a central server, you may lose everything if a server's hard drive is corrupted

https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Distributed Version Control Systems



How it works:

- You don't just check out a snapshot of a version, you mirror the whole version database
- Examples: Git, Mercurial, Bazaar
- Good for collaboration and for keeping multiple backups

https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Why it is Important to have Version Control

- Keep track of changes:
  - Who, when, what, and (sometimes) why
- Easy to work collaboratively:
  - Different lab member(s) work on the same file
- Backup:
  - Get the previous version back if you mess up something

Note: You can use version control for any type of files, not just code

Scenarios

- You saw some changes in a file, but don’t remember if your collaborator made these changes or you did it yourself
- You and your collaborators want to work on the same file
- You accidently deleted an important chunk of code while debugging and there is no way to undo the deletion

What do Other Researchers Think about Git/GitHub?

- A survey, conducted by Investigating & Archiving the Scholarly Git Experience (IASGE), targeting scholars who use Git
- Preliminary results show that
  - **Git** is the most used version control system
  - Reasons for using git hosting platform (such as GitHub) are
    - **Collaboration** (primary)
    - **Openness** (secondary)

What are Git and GitHub?

- Git is
  - A version control system
  - Free and open source
  - https://git-scm.com/

- GitHub is
  - A free git hosting platform
  - Hosting software development and version control using Git
  - Bought by Microsoft in 2018
  - https://github.com/

Git is like taking a snapshot of your files, at different moments, and you can go back to previous versions

Git Terminology

- Repository (Repo): Local and Remote
  - A place to store all files, contents, folders, versions, etc.
  - Local repo: in your own computer
  - Remote repo: GitHub (this workshop) or other git hosting platforms

≠

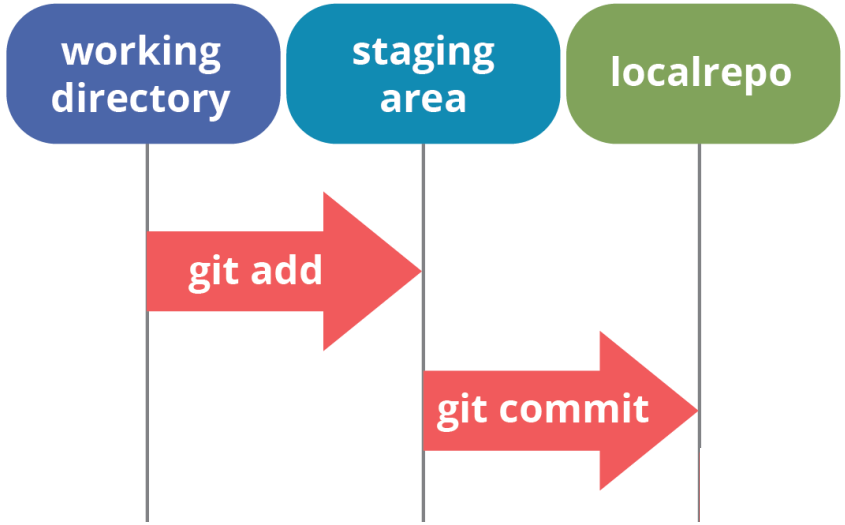
Local **Remote**

Git Terminology

When working on your own computer (local):

- Working directory
  - A directory with your files in a local computer
- Staging area
  - An area to store changed files, but are not yet committed
- Commit
  - (verb): save change(s) to a repo
  - (noun): change(s) to a file

Local



Git Terminology

Commit hash

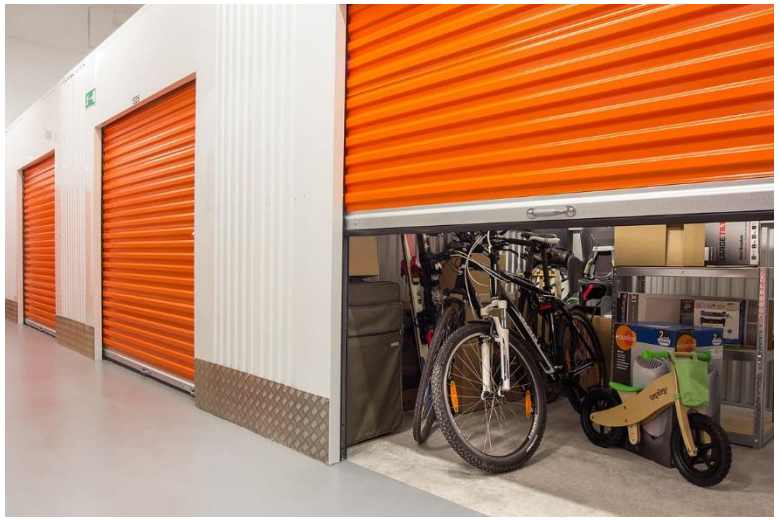
- A unique identifier for a commit
- SHA-1 hashes: An algorithm takes some data as input and generates a unique 40-character string from it
- GitHub commit hash: usually it only takes the first 7 characters

Imagine that You are Moving…

working directory → git add → staging area → git commit → localrepo

Imagine that You are Moving…

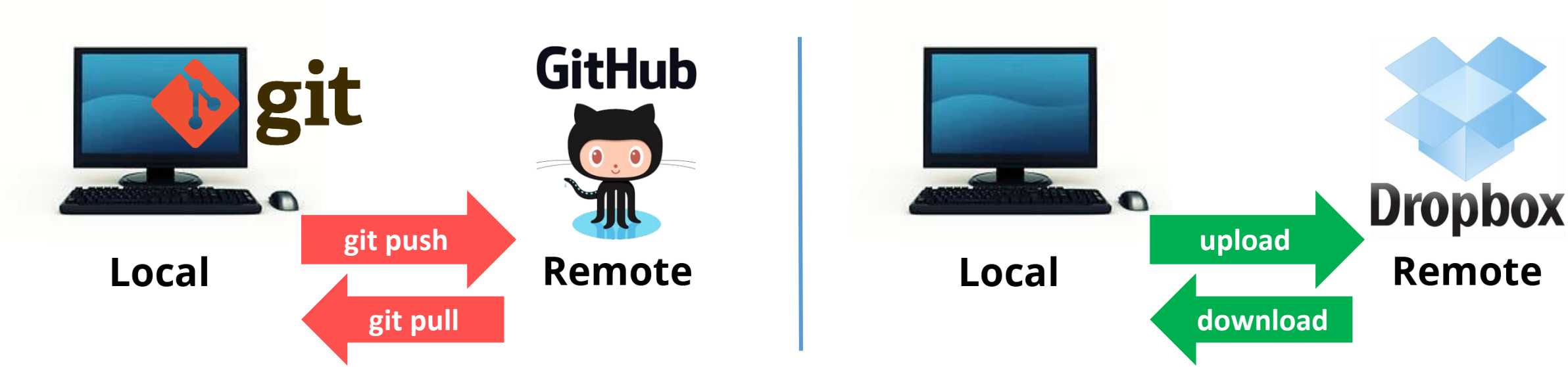
Retrieve old items (versions)
from the storage unit (version database)

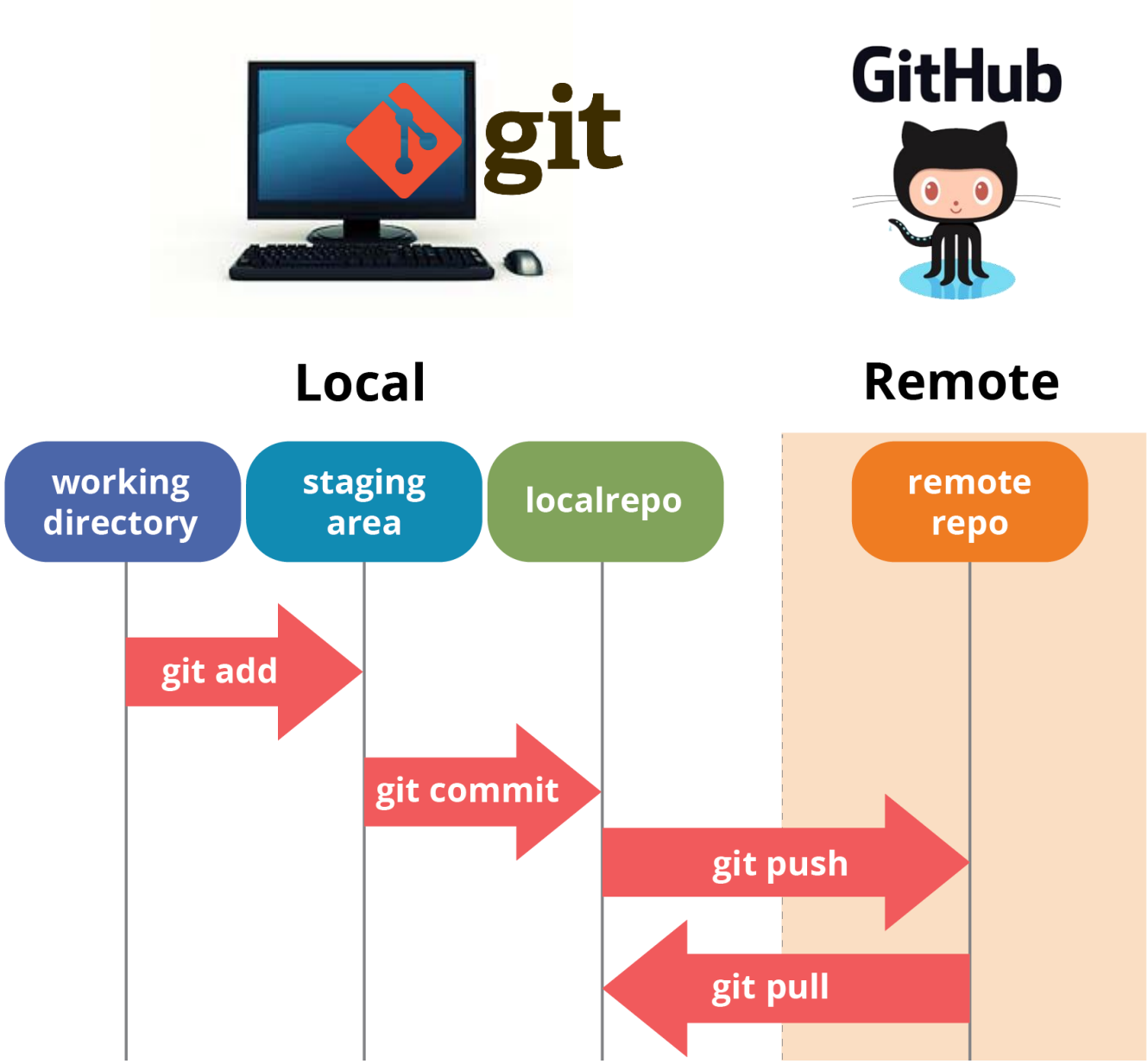


Git Terminology

When communicating with a remote repo (like GitHub)

- Push and Pull (git push and git pull)
  - Push: upload contents from local to remote repo
  - Pull: download contents from remote to local repo




git
GitHub
Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull

Git and GitHub Desktop: Installation

- Git
  - A command line tool
  - Download and install Git (You can skip this if you already installed GitHub Desktop)
  - Documentation
- Git client
  - Graphical User Interface (GUI) tools for committing and browsing (examples)
  - We will demo GitHub Desktop here
  - GitHub Desktop documentation
- Create a GitHub account

Authenticating and Configuring GitHub Desktop

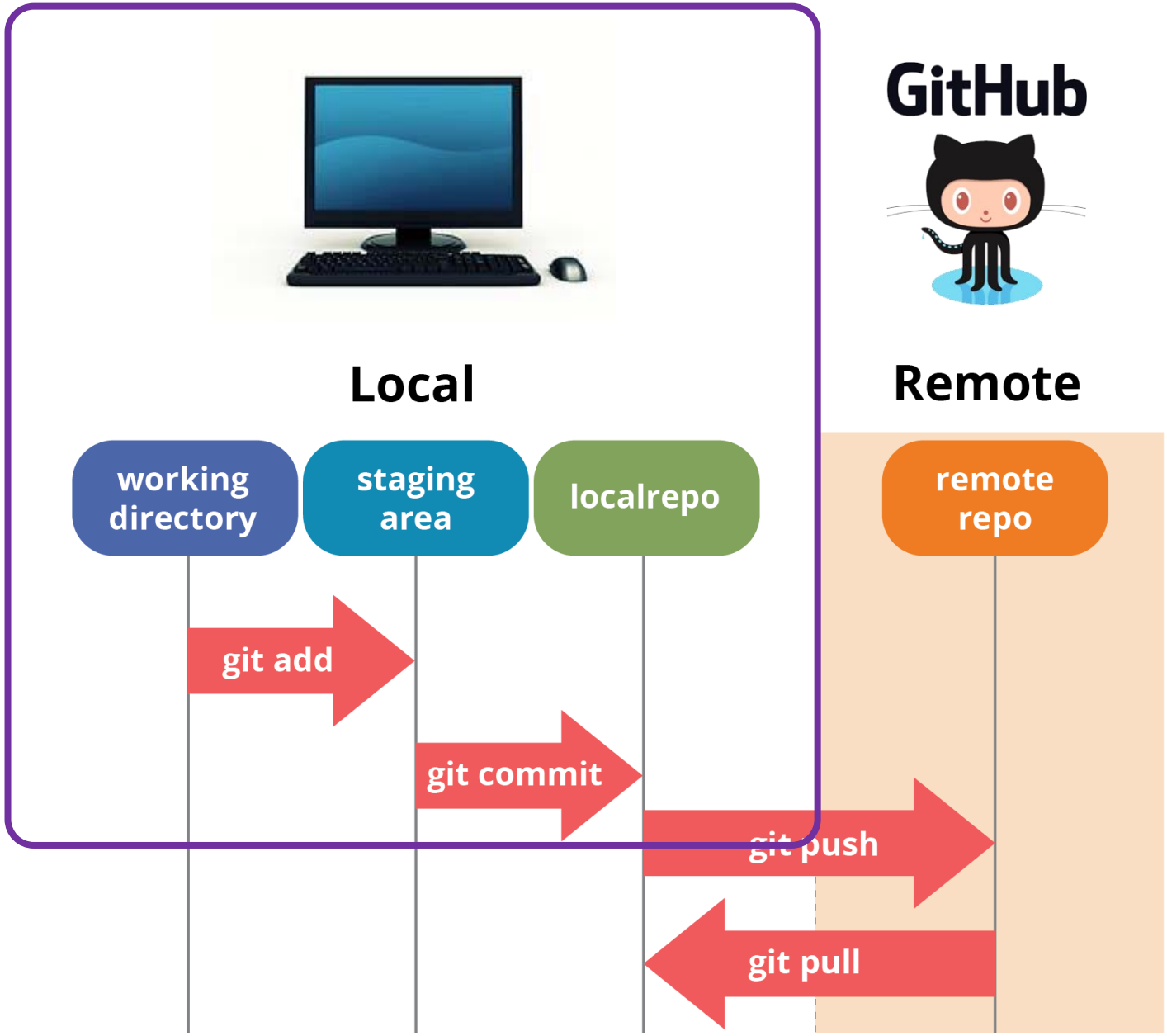
- You only need to do this once
- Authenticating to GitHub using the browser (instruction)
  - File -> Options -> Accounts -> GitHub.com 

  - Click 

  - Type in your GitHub username and password
- Configuring GitHub Desktop (instruction)
  - File -> Options -> Git
  - Enter your username and email
    - Commit email address

GitHub Desktop Demo

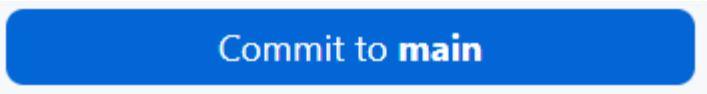
- Create a repo (local and remote)
- Make change to a file and do commits
- Communicate with GitHub
  - Push and pull
  - Ignore files
- Branches
  - Merge and manage conflicts
  - Mange conflicts: pull request
- Clone and fork


GitHub
Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull

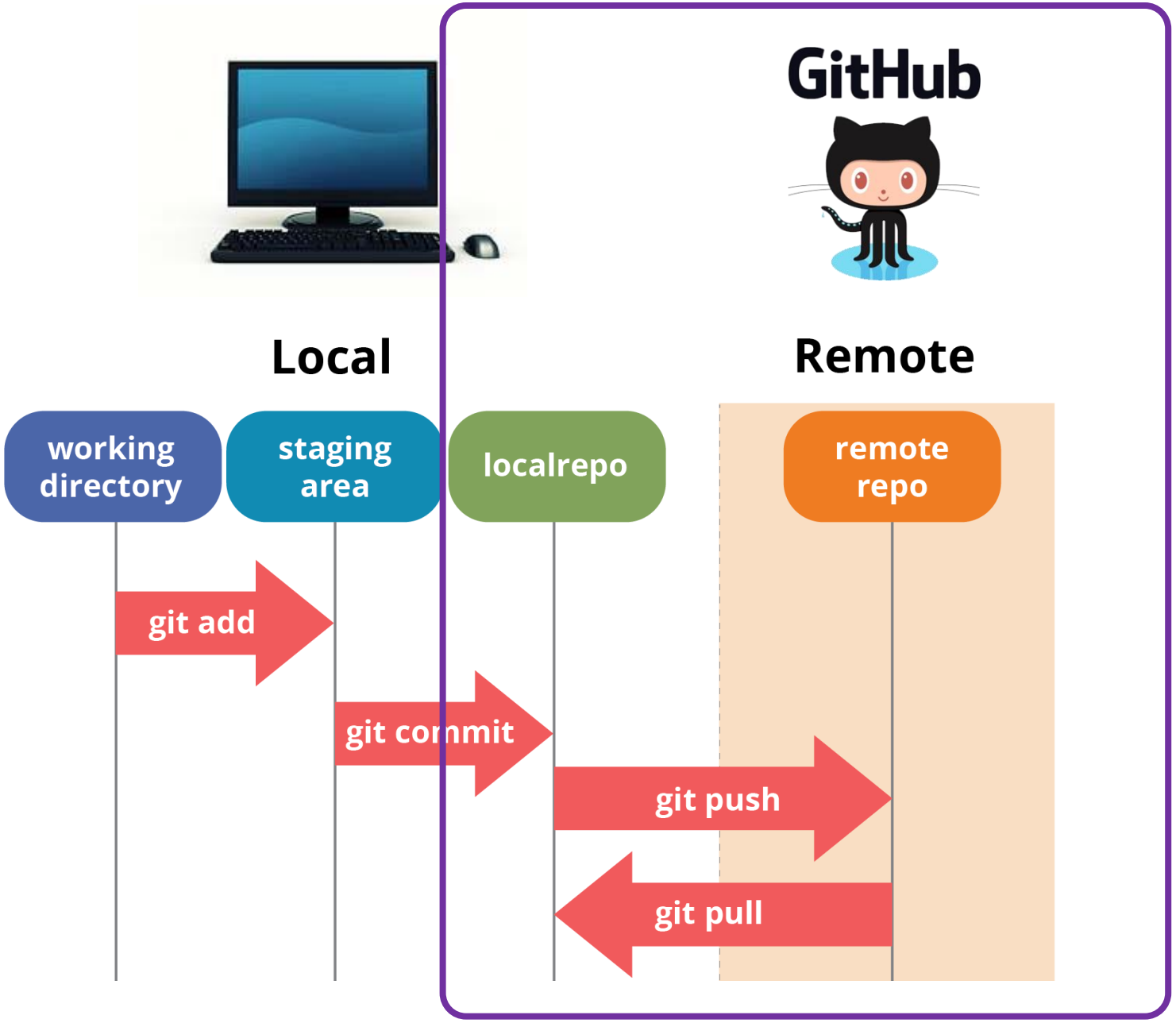
Local Workflow

- Create a new local repo on your computer
  - Check the hidden folders
- Add and edit files in this folder (working directory)
  - Add a file to this folder
  - Write something in the ReadMe file and save
- Commit these changes to the local repo

Local Workflow

- Move to the staging area and commit changes to local repo
  - Checkmark the file(s) you want to commit
  - Write notes in the commit message box
  - Click 

- Commit hash
  - A unique identifier for a commit
  - GitHub commit hash: usually it only takes the first 7 characters
  - Can you find where is the commit hash in your GitHub Desktop?


GitHub
Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull

Communicate with GitHub

- Publish this local repo to GitHub
- Push contents to GitHub
  - This step will update files on GitHub
- Pull contents from GitHub
  - Update the local repo with GitHub files
- Ignore file(s) or folder(s)
  - Choose not to upload certain file(s) or folder(s) to GitHub repo

gitignore: Ignore File/Folder

- What to ignore?
  - The file is not used by your project
  - The file is not used by anyone else in your team
  - The file is generated by another process
  - The file has personal or sensitive information
- Examples
  - .Rproj.user folder or .Rhistory file that auto-generated
  - Personal notes
  - Research data with PII/PHI
  - API keys

https://zellwk.com/blog/gitignore/

gitignore: Ignore File/Folder

- Create a file that we want to ignore (log.txt)
- Start a .gitignore file
  - Do this step **BEFORE** you commit anything
  - Go to Repository tab and select Repository Settings
  - List the file(s) or folder(s) you want to ignore
  - Click Save

gitignore: Ignore File/Folder

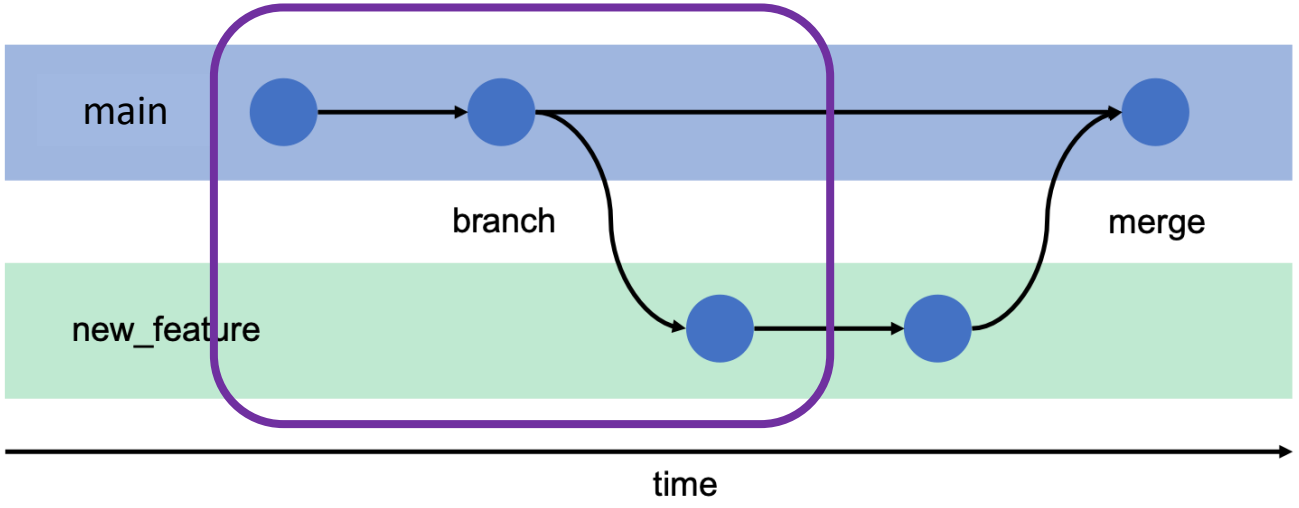
- You can go to your local git repo and will find
  - A .gitignore file with a list of files to be ignored
  - None of the files listed in the .gitignore file will show up in your GitHub repo
  - You can add file/folder names directly into this file and save or repeat the above steps in GitHub Desktop

Branch

- You want to make changes, fix bugs, etc., but don't want to mess up the main copy
- You and your collaborator want to work on the same file

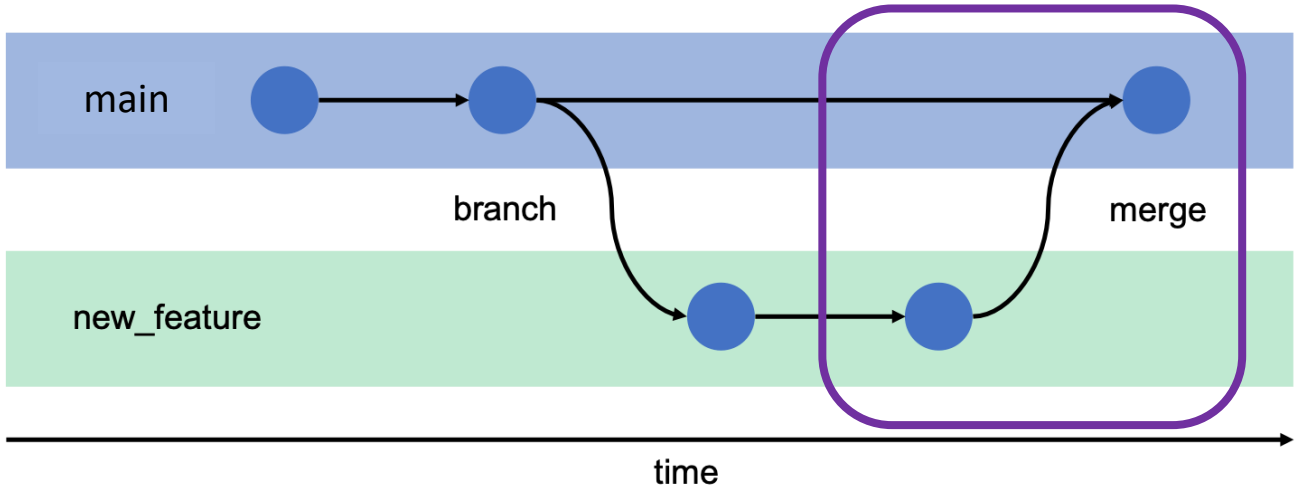
Solution: Create a branch and make changes in that branch.

Once you feel comfortable with your changes, you can merge your changes back to the main branch



Merge a Branch

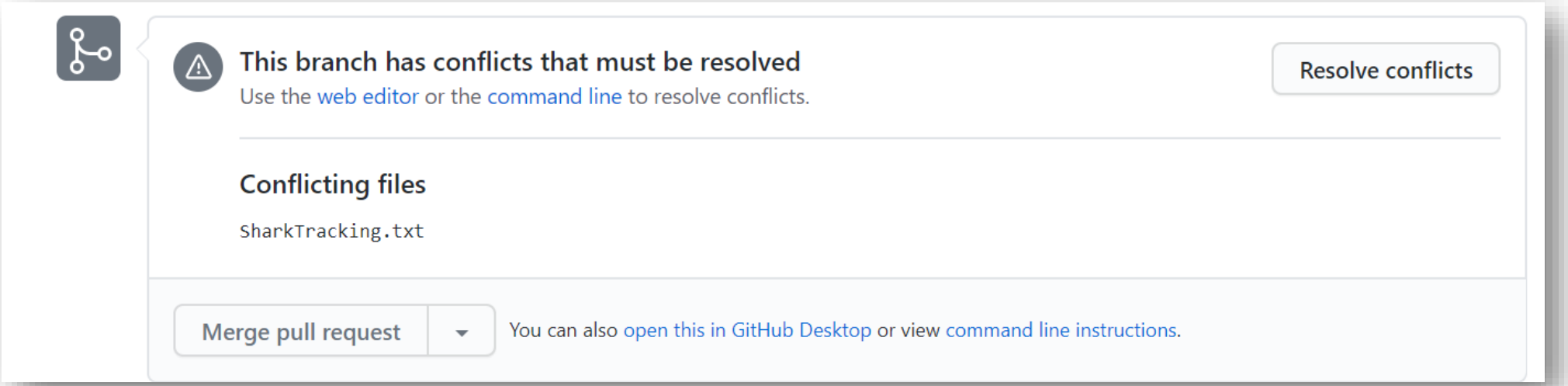
- Send a “pull request” to merge branch to the main branch
  - Can only have one pull request each time
  - Can create another pull request once the previous pull request is merged



Potential Problem for Multiple Branches?

How to Resolve a Conflict?

- GitHub will show you the conflict and you need to decide which change you want to keep
- GitHub demo: resolve conflicts

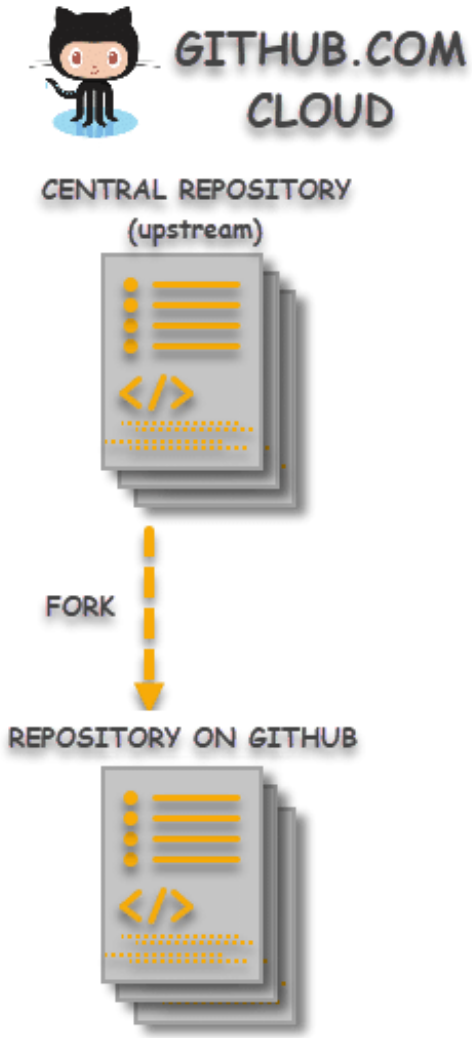


Learn more: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/resolving-a-merge-conflict-on-github

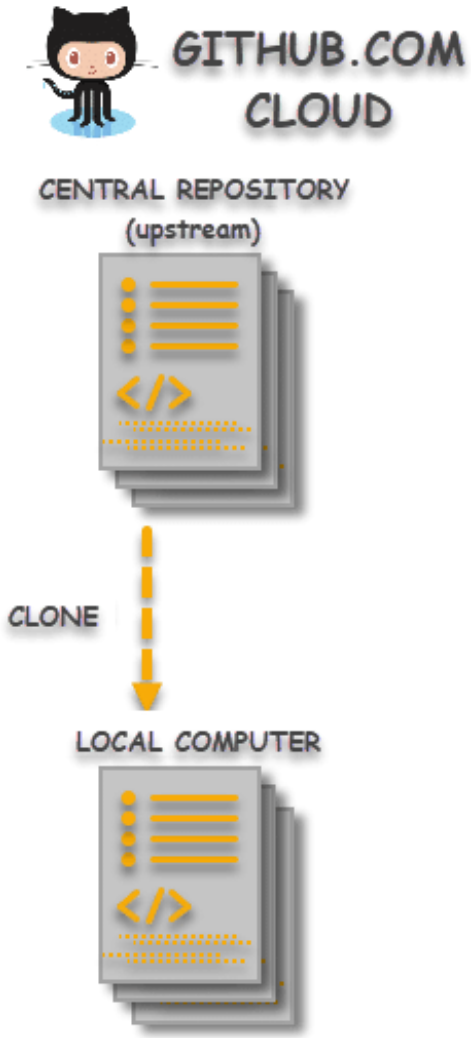
Clone and Fork a GitHub Repository

- Clone a repository from GitHub (to local machine)
  - Clone other people’s repo
  - Clone your own
- Fork a repository from GitHub (to your GitHub account)
  - Fork a repository that you don’t have write access
- What’s the difference?
  - Fork (GitHub), Clone (local computer)
  - Make changes: Fork (Pull request), Clone (Push)

FORK



CLONE



https://www.toolsqa.com/git/difference-between-git-clone-and-git-fork/

Resources: Learning Git and GitHub

- GitHub Skills
- Resources to learn Git
- Git and GitHub learning resources
- Learn Git Branching: An interactive guide
- Version control with Git by Coursera
- Reproducible Research Toolkit by coding2share

Resources: GitHub Desktop

- GitHub document
- GitHub Desktop document
- Branches in GitHub using GitHub Desktop
- GitHub Tutorial 2020 - Beginner's Training Guide
- How to resolve a Merge Conflict in GitHub Desktop

Desktop

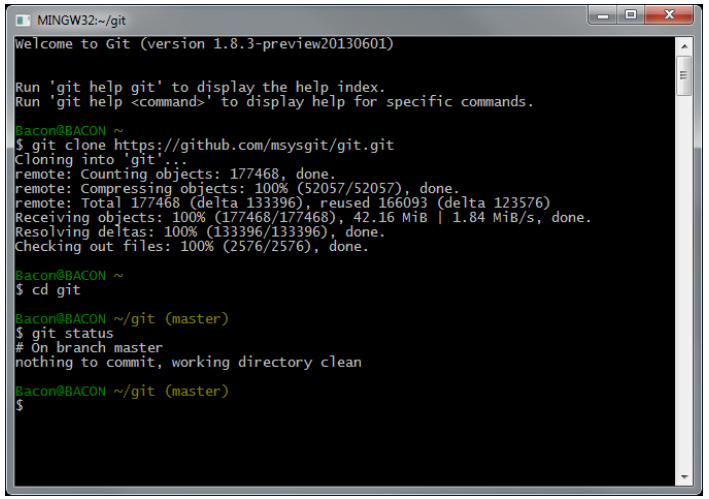
Resources: GitHub and R Studio

- Happy Git and GitHub for the useR
- Creating R Studio projects from GitHub Repositories:
  - https://www.youtube.com/watch?v=YxZ8J2rqhEM
  - https://happygitwithr.com/new-github-first.html
- Link an existing R project to GitHub:
  - https://happygitwithr.com/existing-github-first.html
  - https://hansenjohnson.org/post/sync-github-repository-with-existing-r-project/
- RMarkdown and GitHub
- Using the ATOM editor with R

Resources: Command Lines

You can use Git command lines (without using GitHub Desktop) for everything we talked about in this workshop

- Git & GitHub crash course for beginners
- Git cheatsheet
- Git documentation



https://gitforwindows.org/

Resources: Other than Coding

GitHub is not only for code management, you can use it for other purposes.

For example:

- Manuscript revision: Organizing a Paper Revision with GitHub
- Group project: Data Curation Network Primers
- Share a Template: Best-README-Template
- Any collaborative projects you can think of!

Resources: Other Git Hosting Platforms

Here are a few other Git hosting platforms (other than GitHub):

GitLab

https://about.gitlab.com/

SOURCEFORGE

https://sourceforge.net/

Bitbucket

https://bitbucket.org/

beanstalk

https://beanstalkapp.com/

Resources: JHU GitHub Enterprise Account

- GitHub Plans and Pricing: Free, Team, Enterprise
- JHU GitHub Enterprise account
  - Single sign-on with JHED
  - Create a team GitHub space for your lab
  - Up to 50 GB storage space
  - Contact us if you are interested in getting one

JOHNS HOPKINS UNIVERSITY

Contact JHU Data Services

GO TO
dataservices.library.jhu.edu

EMAIL
dataservices@jhu.edu

SHARE DATA AT
archive.data.jhu.edu

Helping you

FIND

USE

MANAGE

VISUALIZE

SHARE

DATA

JOHNS HOPKINS LIBRARIES | Data Services