So git ?

Source code control and data repositories

Suzanne Little

What is it?

Git is an open source, distributed, version control system designed for speed and efficiency.

What is it?

Git is an open source, **distributed, version control system** designed for speed and efficiency.

- fast (built for linux) [8,196 loc in 1991, now?]
- local - able to work offline
- harder to lose changes
- easier to collaborate on code

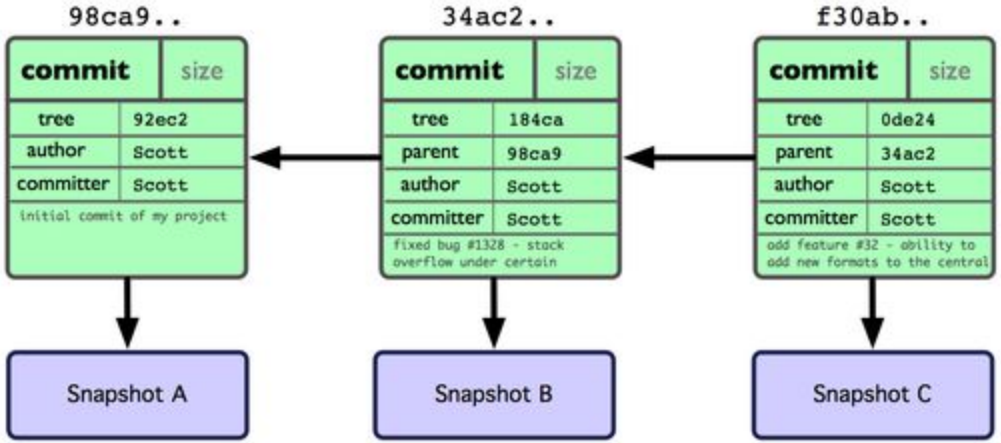
Terminology

Repo (or repository): collection of files

Git vs Gitlab vs Github: hosts of repositories

Source or source code: text of programs

What is git?



Stores ‘snapshots’ of your work

And the relationship between those snapshots

Why?

- Record of changes (date, comment, diff)
- Ability to ‘rollback’ to a previous version
- Have multiple ‘branches’ of software (public, beta, functionX, etc.)
- Collaboration with team members and others (merge)
- DS: Source of datasets https://github.com/datasets

More terminology (check your understanding)

local vs remote

master

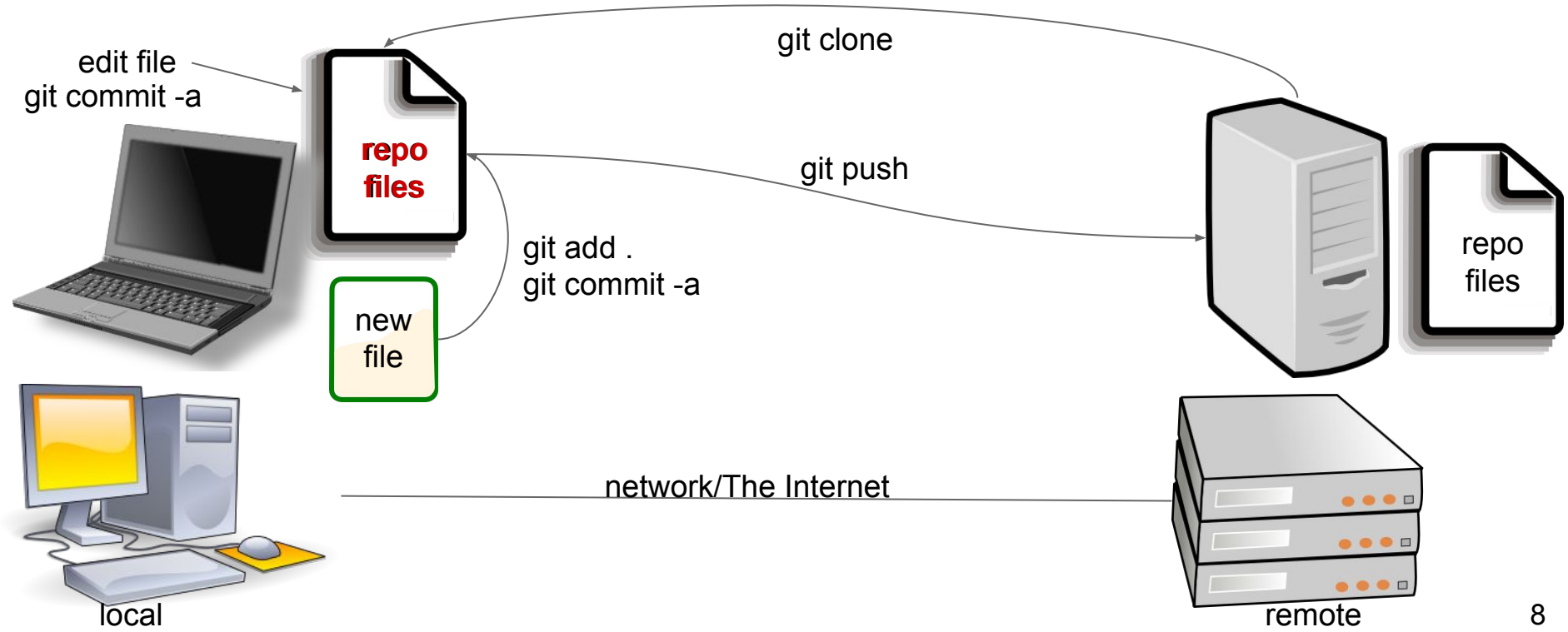
clone or fork & branch

staging

commit & merge

pull & push & fetch & upstream

local vs remote



Getting started

- Version Control Vocab - https://en.wikipedia.org/wiki/Version_control
- What about CVS, SVN, Bazaar, Mercurial?
- git vs github vs gitlab vs bitbucket?
- install git
  - https://git-scm.com/downloads
  - https://www.atlassian.com/git/tutorials/install-git
  - 1st config (set name, email, editor preference)

My first repo

Log in to https://gitlab.computing.dcu.ie → New Project

Name the project: `202409-csc1158-<username>-first`

```
$ git clone <url to repo>
$ cd 202409-csc1158-<username>-first
$ ls .git
```

edit README.md (don't forget to save the file!)

```
$ git status
$ git commit -a
```

write a useful update message

```
$ git push
```

these commands (`$ cmd`) happen on the local computer

My first repo

Look at 202409-csc1158-<username>-first on gitlab.computing.dcu.ie, then

```
$ mkdir 01
$ cd 01
$ touch gitnotes.md
```
← what does this do?

edit gitnotes.md

```
$ git status
$ git add .
```
← the . is important (but don't over use it)

```
$ git commit -am "added note page for git"
$ git status
$ git push
```

Reflection on first steps

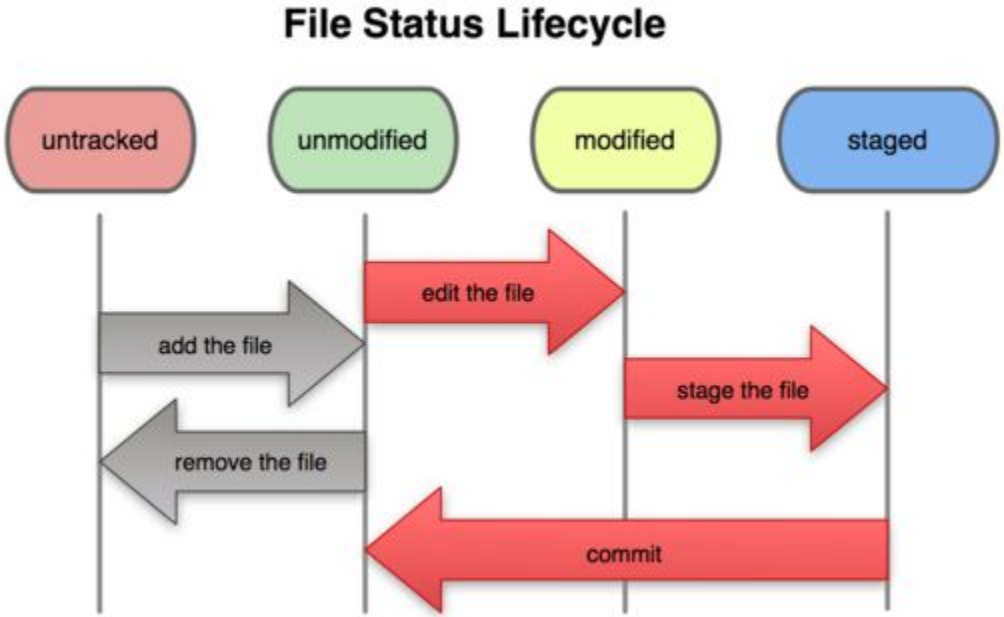
On windows use gitbash to run your commands

On linux use the terminal (git already installed)

There are multiple ways to do things

Working on your local “clone” or in the browser? Using a code editor?

`git commit -a`



Untracked
Git doesn't know about these files

Tracked, unmodified
Git knows about them, and there have been no changes.

Tracked, modified
File is in git, but changes have been made that git doesn't know about.

Staged
Changes to this file will be committed next time you commit.

`git status` is your best friend!

A basic workflow

Edit files	vim / vs code / etc
Stage the changes	git add (file)
Review your changes	git status
Commit the changes	git commit -am “note”
Upload to remote	git push

Code Editor?

What do you use to edit Python code?

Text editors

- Notepad++ https://notepad-plus-plus.org/download/v7.5.8.html
- Visual Studio Code (Anaconda Navigator → Install VS Code)
- Sublime http://www.sublimetext.com/
- Emacs (on Linux)
  https://www.masteringemacs.org/article/beginners-guide-to-emacs
- Nano (menu on the bottom of the screen, e.g. Ctrl-X = exit)
- vi or vim https://www.linux.com/learn/vim-101-beginners-guide-vim

`git commit` → vi or vim

When you run `git commit -a` you may end up in a strange environment trying to add your comments. This is mostly likely the vi/vim editor. See here for more information about this text editing programme. It's very powerful but takes some practise to use. You should know **three** commands for vi - exit without saving, save and exit and edit mode.

1. Exit without changing → `Esc :q!`
2. Edit mode → `i` then type your changes, `Esc` will exit edit mode
3. Save changes and exit → `Esc :wq`

You can change the default editor for your system to something like gedit on linux or notepad on windows (git config --global core.editor "gedit") - don't forget to **save** your comments before quitting (don't just click the x in gedit!).

md? MarkDown

A really simple way to format a plain text file (headings, bold, bulleted lists, etc.)

Used in git for documentation

Can include hyperlinks, images, code and more

Reference: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

Try it out: https://markdown-it.github.io/

Practise

1. Make sure you have an up-to-date local copy of your 2024-ca273-<username>-first repo (`git clone <url>` or `git pull`)
2. Make a new file called `ca273-assignments.md`
3. Add this file to your repo
4. Create a list with the details of the assignments for ca273 (use markdown format)
5. commit and push the changes to your remote repo

Resources

- https://gitlab.computing.dcu.ie/sblott/local-gitlab-documentation
- http://nyuccl.org/pages/GitTutorial/ - basics
- http://vimeo.com/14629850 - overview video
- http://try.github.io - interactive tutorial
- http://git-scm.com/videos - intro videos
- https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf - cheat sheet (useful overview)
- https://www.datacamp.com/courses/introduction-to-git-for-data-science

Sick of typing your password? Setup ssh

```
$ ssh-keygen -t rsa -b 4096 -C "titleforkey"
```

copy contents of .ssh/id_rsa.pub to gitlab