Introduction to Revision Control with git

Afif Elghraoui
NIH HPC staff
staff@hpc.nih.gov

Why?

- Sanity
  - No need to manually maintain multiple copies of the same file.
  - No need to preserve commented out sections of old code in your working copy.
- History
- Annotations
- Undoability
- Collaboration

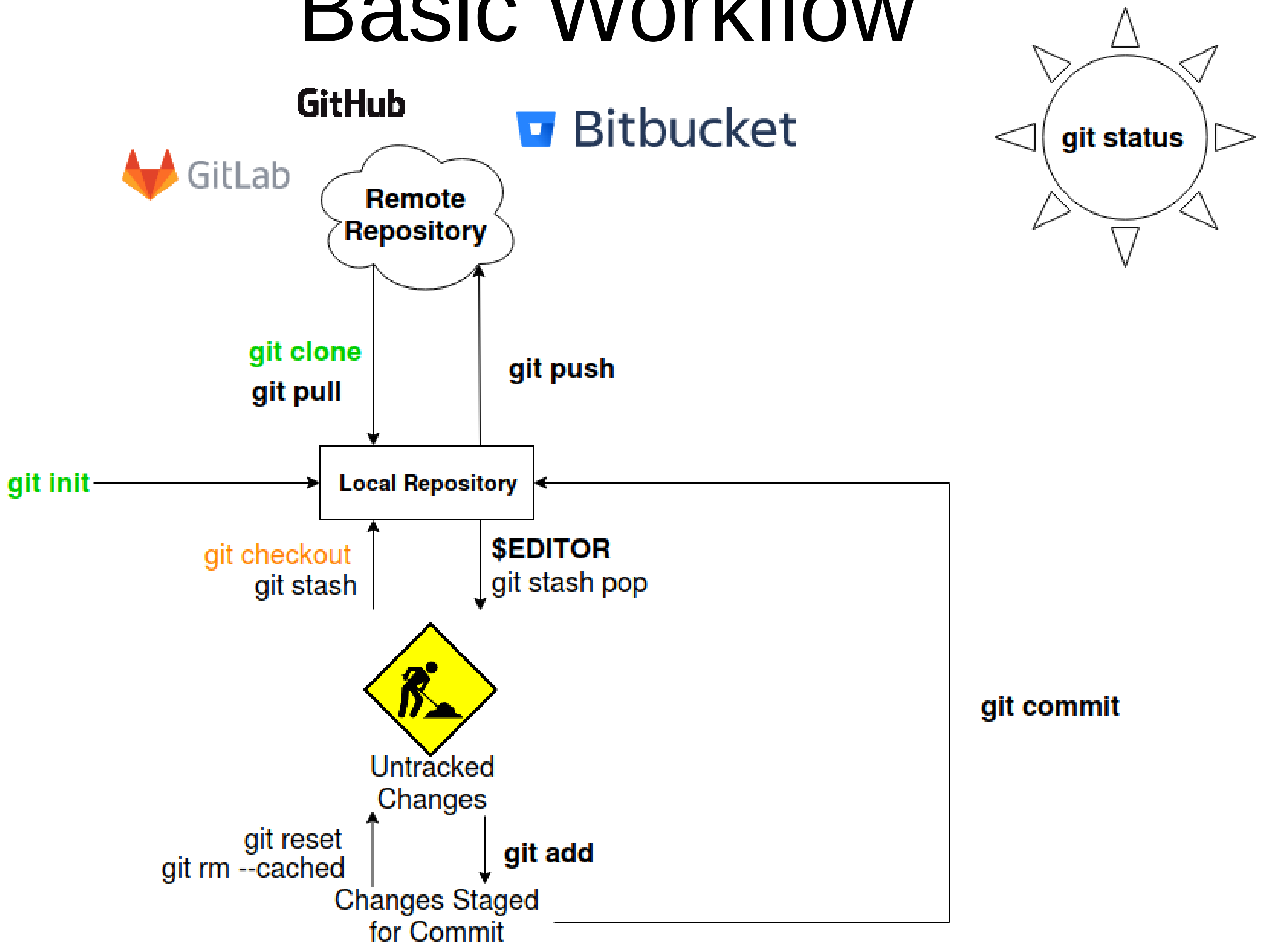
Getting started

- `git config --global user.name “Fulan bin Fulan”`
- `git config --global user.email fulan@example.com`
- `export EDITOR=nano`

  `echo “!!” >> ~/.bashrc`

Global configurations go into `~/.gitconfig` by default. Local configurations are per-repository.

Basic Workflow



🟩 initialization

🟧 destructive operation

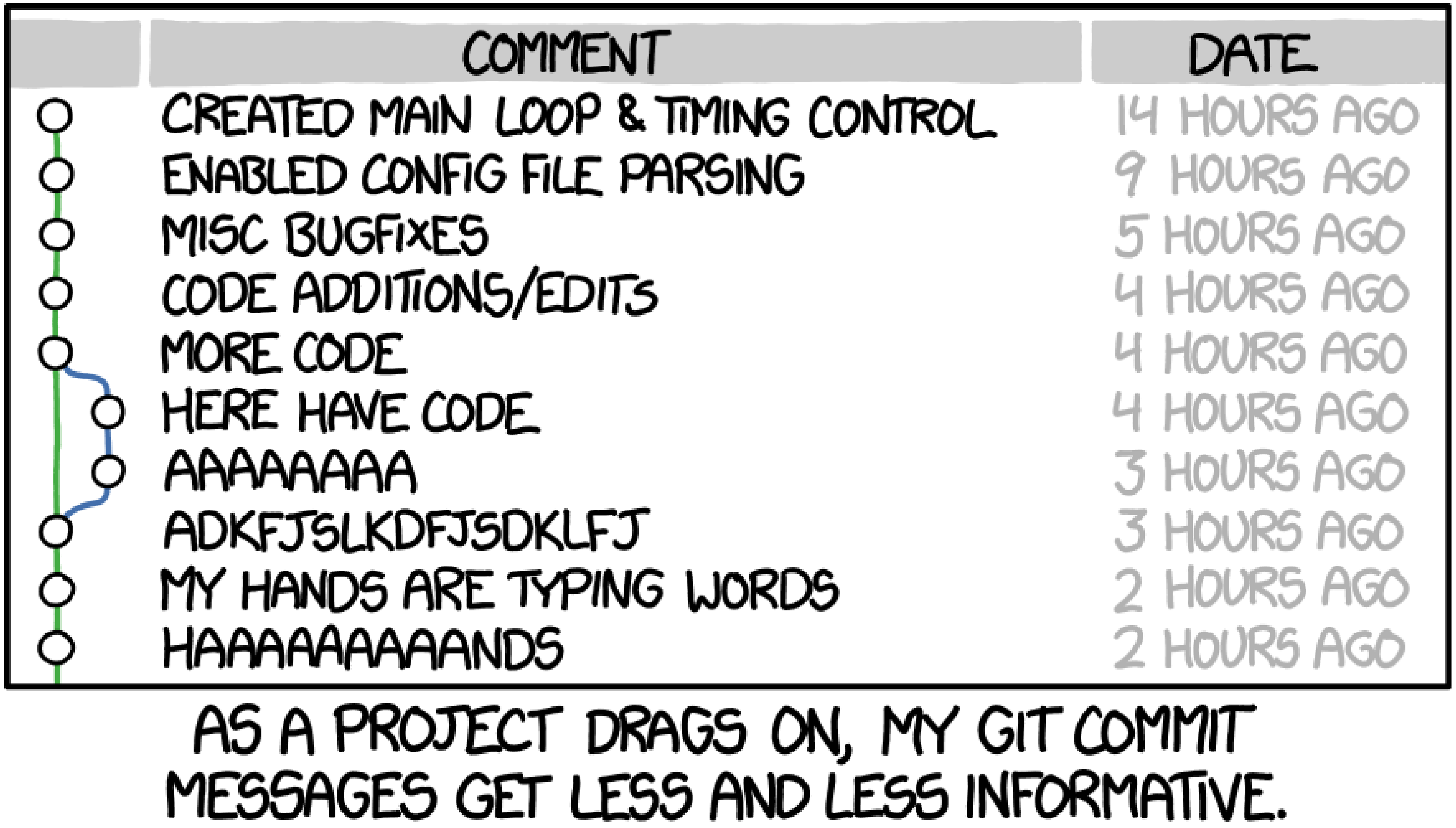
What Goes in Git?

- Content that is manually generated and maintained.
- Primarily text files. Don’t commit large binary files because Git can’t compare and track them efficiently. If you need to track them, consider tools like git-lfs or git-annex.

Ignoring files: `.gitignore`

- Specify file names or patterns in a file named `.gitignore` in your repository to avoid accidentally committing unwanted files like
  - Editor backups
  - Program outputs
  - Sensitive information

Commit Messages



https://xkcd.com/1296/

BIOWULF
HIGH PERFORMANCE COMPUTING AT THE NIH

Commit Messages

- Bad:

  Updated README

- Good:

  README: update installation steps due to os upgrade

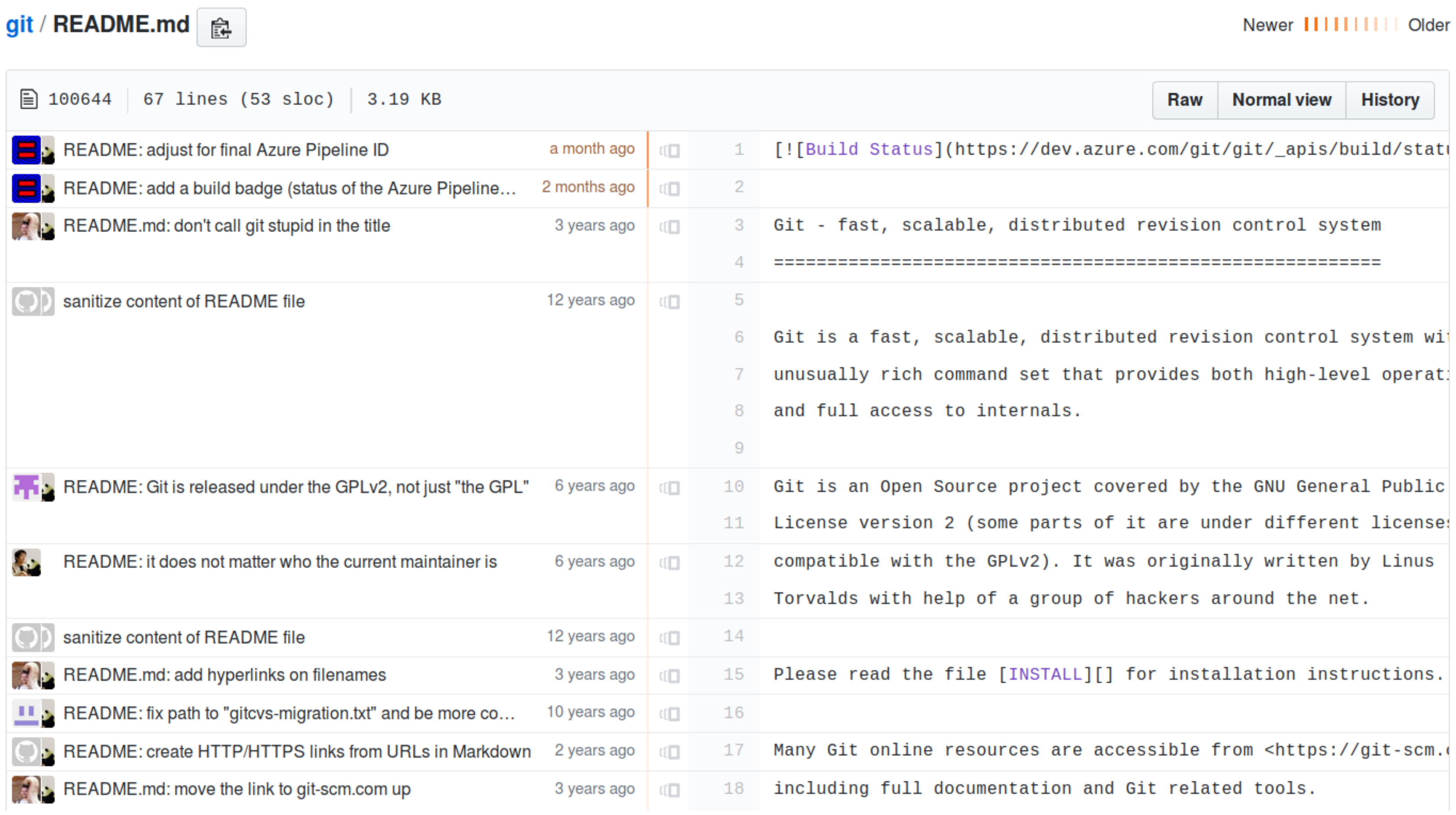
  The new version of the OS no longer comes with support for foo, which we were relying on.

Inspecting the Repository

- `git log`
  - See the change history of a file: `git log -p filename`
- `git blame`
- `git diff`
- A graphical utility: `gitk(1)`
- Use `git checkout` to move to different points in history

Why? Annotations

`git blame`



git / README.md

Newer Older

100644 | 67 lines (53 sloc) | 3.19 KB

Raw | Normal view | History

README: adjust for final Azure Pipeline ID	a month ago	`[![Build Status](https://dev.azure.com/git/git/_apis/build/stat`
README: add a build badge (status of the Azure Pipeline…	2 months ago	

README.md: don't call git stupid in the title	3 years ago	`Git - fast, scalable, distributed revision control system`
		`=========================================================`

sanitize content of README file	12 years ago	
		`Git is a fast, scalable, distributed revision control system wi`
		`unusually rich command set that provides both high-level operat`
		`and full access to internals.`

README: Git is released under the GPLv2, not just "the GPL"	6 years ago	`Git is an Open Source project covered by the GNU General Public`
		`License version 2 (some parts of it are under different license`

README: it does not matter who the current maintainer is	6 years ago	`compatible with the GPLv2). It was originally written by Linus`
		`Torvalds with help of a group of hackers around the net.`

sanitize content of README file	12 years ago	
README.md: add hyperlinks on filenames	3 years ago	`Please read the file [INSTALL][] for installation instructions.`
README: fix path to "gitcvs-migration.txt" and be more co…	10 years ago	
README: create HTTP/HTTPS links from URLs in Markdown	2 years ago	`Many Git online resources are accessible from <https://git-scm.`
README.md: move the link to git-scm.com up	3 years ago	`including full documentation and Git related tools.`

https://github.com/git/git/blame/master/README.md

Marking Important Snapshots: versioning and `git tag`

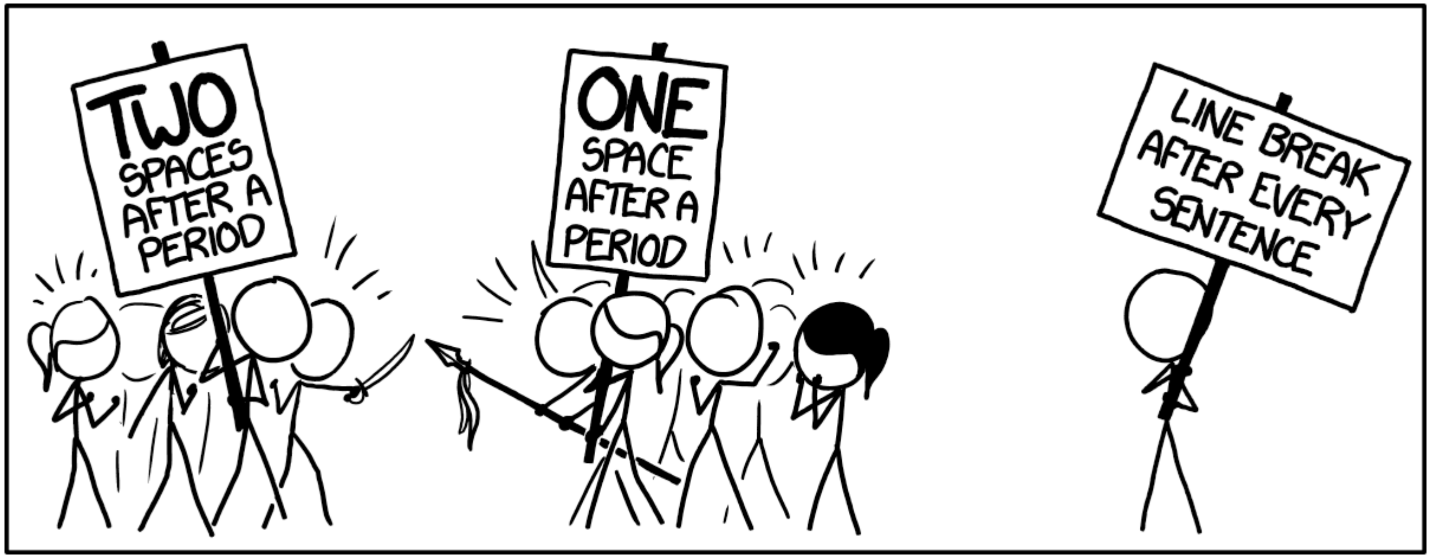
- Example: create an annotated tag marking the current snapshot as version 0.1.0:

  ```
  git tag -a 0.1.0
  ```

  a commit id can be specified at the end if you don’t want the tag pointing to the current HEAD.
- Semantic versioning - https://semver.org/

Helpful Conventions

- Put meaningful components on their own line to be able to get more useful comparisons
  - When editing prose in a markup language, put a line break after every sentence to get this effect.



https://xkcd.com/1285/

BIOWULF
HIGH PERFORMANCE COMPUTING AT THE NIH

Remote Repositories

- Common Git hosting providers are GitHub, GitLab, and Bitbucket
- Use `git remote` to manage remote repositories from your working directory.

Branching and Collaboration

- Create a new branch:

  `git branch` ***`branchname`***

- Switch to branch:

  `git checkout` ***`branchname`***

  ...or create the branch and switch to it in one step:

  `git checkout -b` ***`branchname`***

- Make your changes (edit, `git add`, `git commit`)
- Push your branch to the remote repository

  `git push origin` ***`branchname`*** **`--set-upstream`**

- Switch back to the default branch

  `git checkout master`

`git merge`

- Git can automatically merge branches if there are no conflicting changes.
- Merge conflicts are presented to you to resolve:
  - Both versions of the file are available for comparing.
  - Use a merge tool to make your life easier. See `git mergetool --tool-help` for a list of options.
- Fun fact: `git pull` is actually a shorthand for `git fetch` followed by `git merge`

Resources

- reference book: https://git-scm.com/book/en/v2
- man pages:
  - `git(1)`
  - **`man git-`*`subcommand`*** or **`git help`** ***`subcommand`***
- more man pages:
  - `giteveryday(7)`
  - `gittutorial(7)`