# Git and GitHub/GitLab

*Version control and collaboration*

Jason DeBacker

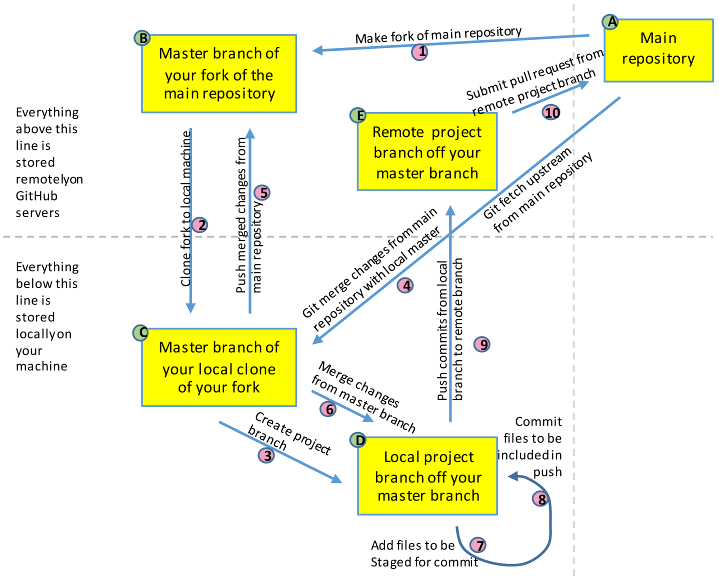
January 16, 2025

## Roadmap

## Words of wisdom(?) at the outset

- The ability to effectively and efficiently scale up collaboration is one of the most important drivers of productivity growth.
- Two warnings that a seasoned Git and GitHub user should always give a new entrant to this type of version control and code collaboration are the following.
  - → The learning curve is steep.
  - → The workflow is initially not intuitive.

# Git and GitHub workflow


A
Main repository
B
Master branch of your fork of the main repository
1
Make fork of main repository
Everything above this line is stored remotelyon GitHub servers
Everything below this line is stored locallyon your machine
2
Clone fork to local machine
5
Push merged changes from main repository
E
Remote project branch off your master branch
10
Submit pull request from remote project branch
Git fetch upstream from main repository
4
Git merge changes from main repository with local master
C
Master branch of your local clone of your fork
9
Push commits from local branch to remote branch
6
Merge changes from master branch
3
Create project branch
D
Local project branch off your master branch
8
Commit files to be included in push
7
Add files to be Staged for commit

## What are Git and GitHub?

**Definition (Git)**

**Git** is an open source distributed version control system (DVCS) software that resides on your local computer and tracks changes and the history of changes to all the files in a directory or repository. See the Git website https://git-scm.com/ and the Git Wikipedia entry for more information.

## What are Git and GitHub?

**Definition (GitHub)**

**GitHub** or **GitHub.com** is a cloud source code management service platform designed to enable scalable, efficient, and secure version controlled collaboration by linking local Git version controlled software development by users. GitHub's main business footprint is hosting a collection of millions of version controlled code repositories. In addition to being a platform for distributed version control system (DVCS), GitHub's primary features include code review, project management, continuous integration unit testing, GitHub actions, and associated web page (GitHub pages) and documentation hosting and deployment.

## What are Git and GitHub?

**Definition (repository or repo)**

A **repository** or **repo** is a directory containing files that are tracked by a version control system. A local repository resides on a local machine. A remote repository resides in the cloud.

## Wide usage

“Octoverse 2023: The state of open source software”. In 2023:

- “93% of developers use [GitHub] to build and deploy software everywhere”
- 420 million total projects
- 284 million public repositories
- 65 thousand public generative AI projects
- 4.5 billion total contributions to all projects on GitHub
- 98 million new projects started on GitHub

## Wide usage

*"The best indication of Git's market dominance is a survey of developers by Stack Overflow. This found that 88.4% of 74,298 respondents in 2018 used Git (up from 69.3% in 2015). The nearest competitors were Subversion, with 16.6% penetration (down from 36.9%); Team Foundation Version Control, with 11.3% (down from 12.2%); and Mercurial, with 3.7% (down from 7.9%). In fact, so dominant has Git become that the data scientists at Stack Overflow didn't bother to ask the question in their 2019 survey."*

*Andy Favel, "The history of Git," Feb. 2020*

## Git and GitHub brought us Linux

*"The development of Git began on 3 April 2005. Torvalds announced the project on 6 April and became self-hosting the next day. The first merge of multiple branches took place on 18 April. Torvalds achieved his performance goals; on 29 April, the nascent Git was benchmarked recording patches to the Linux kernel tree at the rate of 6.7 patches per second. On 16 June, Git managed the kernel 2.6.12 release.*

*Wikipedia, "Git," Aug. 2024*

## Git and GitHub brought us Linux

Show Linux contributors page: https://github.com/torvalds/linux

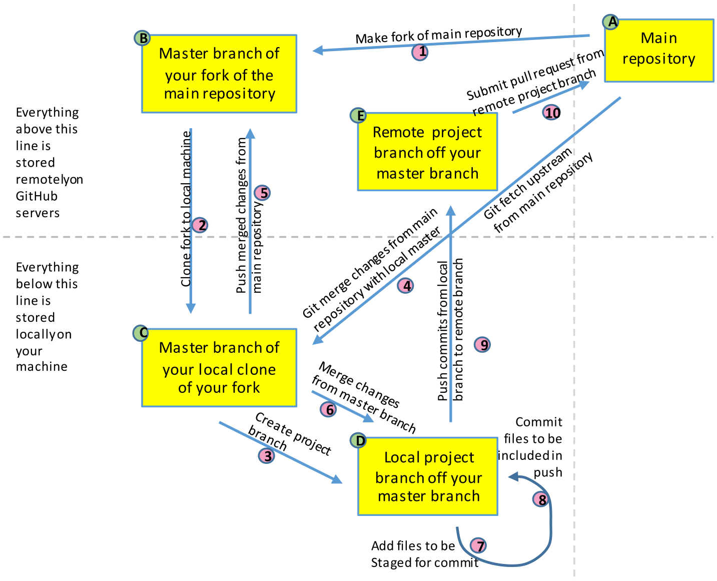
## Alternatives to Git and GitHub

Alternatives to GitHub include GitLab and Bitbucket. Other alternatives are documented in this March 2024 post by Software Testing Help.

But GitHub has the largest user base and largest number of repositories.

## Roadmap

# Git and GitHub workflow


A
Main repository
1
Make fork of main repository
B
Master branch of your fork of the main repository
Everything above this line is stored remotelyon GitHub servers
Everything below this line is stored locallyon your machine
2
Clone fork to local machine
5
Push merged changes from main repository
10
Submit pull request from remote project branch
E
Remote project branch off your master branch
Git fetch upstream from main repository
4
Git merge changes from main repository with local master
C
Master branch of your local clone of your fork
9
Push commits from local branch to remote branch
6
Merge changes from master branch
3
Create project branch
D
Local project branch off your master branch
Commit files to be included in push
8
7
Add files to be Staged for commit

## Git vs. Google Docs and Dropbox

- Google Docs and Dropbox can allow changes with no outside confirmation
- Google Docs and Dropbox comments and suggesting mode can pile up in a difficult way
- Google Docs history is difficult to order and search
- Dropbox can create version conflicts
- Cannot set hierarchical permissions easily

## Some git definitions

- clone: a copy of a repository
- fork: a copy of a repository that you manage
- branch: a parallel version of a repository
  - → main/master: name of fork with the main version of the code
  - → Other forks will have names specific to development on that branch
- commit: a snapshot of a repository
- upstream: typical name used to reference the original repository
- origin: typical name used to reference your fork of the original repository
- fetch: get the latest changes from an online repository

## Git workflow example in the terminal

Show what workflow looks like in the terminal.

## Most commonly used git commands

| **Functionality** | **Git Command** |
|---|---|
| See active branch and uncommitted changes for tracked files | `git status -uno` |
| See branches (with note about active branch) | `git branch` |
| Change branch | `git checkout <branch name>` |
| Create new branch and change to it | `git checkout -b <new branch name>` |
| Track file or latest changes to file | `git add <filename>` |
| Commit changes to branch | `git commit -m "message describing changes"` |
| Stash changes (ignore them from history) | `git stash <filename>` |

## Most commonly used git commands (cont'd)

| **Functionality** | **Git Command** |
|---|---|
| Push committed changes to remote branch | `git push origin <branch name>` |
| Merge changes from master into development branch | `(change working branch to master, then...) git merge <branch name>` |
| Merge changes from development branch into master | `(change to development branch, then...) git merge master` |
| List current tags | `git tag` |
| Create a new tag | `git tag -a v<version number> -m "message with new tag"` |

## Most commonly used git commands (cont'd)

| **Functionality** | **Git Command** |
|---|---|
| Pull changes from remote repo onto local machine | `git fetch upstream` |
| Merge changes from remote into active local branch | `git merge upstream/<branch name>` |
| Clone a remote repository | `git clone <url to remote repo>` |

## Roadmap

## GitHub Issues

GitHub issues are a great place to make lists of things to update and fix and a great forum for discussing problems.

### GitHub Issue AllStars

- OG-ZAF Issue #18, “Compute lifetime earnings profiles”
- OG-USA Issue #20, “Values of dataframe in psid_data_setup.py”
- OG-Core Issue #828, “Generalize spline tax functions to 2D”
- OG-Core, Issue #574, “New monotone tax function estimation and Euler equation solution”
- OG-Core, Issue #234, “Estimate tax functions from microdata that are monotonic in labor and capital income”

## Roadmap

## Pull requests (PRs)

Pull requests are the submissions of code changes that await approval, testing, and merging.

### Components of a PR

- Description of what the PR does
- Show exactly what changed
- Continuous integration (CI) automatic testing
- PR thread
  - → OG-ZAF, PR #42, "UN data portal ssl fix"
  - → OG-ZAF, PR #31, "Calibrate IO matrix"
  - → OG-ZAF, PR #28 "Update get_e_orig() with an adjustment by age calibration for ZAF"

## Roadmap

## Attribution

- `git blame`: OG-Core/ogcore/txfunc.py
- GitHub Activity

  → https://github.com/jdebacker

## Roadmap

Intro

Workflow

Issues

Pull requests (PRs)

Attribution

GitHub vs GitLab

## GitHub vs GitLab

- GitHub is the most popular platform for open-source projects
- GitHub is hosted on the Internet
- GitLab is a popular alternative
- GitLab is hosted on a private server
- Workflow is similar between the two
- GitLab has more features for security and compliance
- There are some differences with continuous intergration (CI) tools (I'll point these out later in our workflow topic)