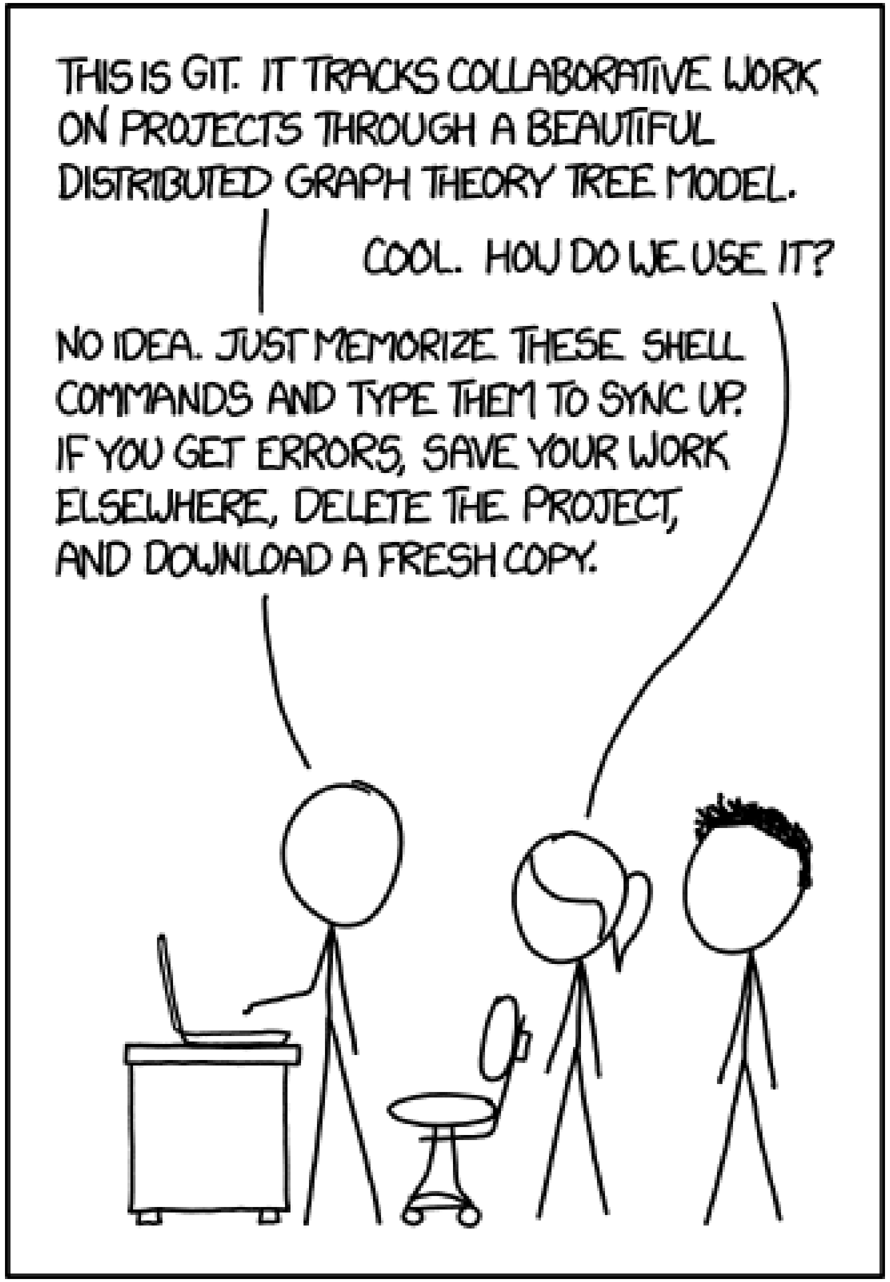

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

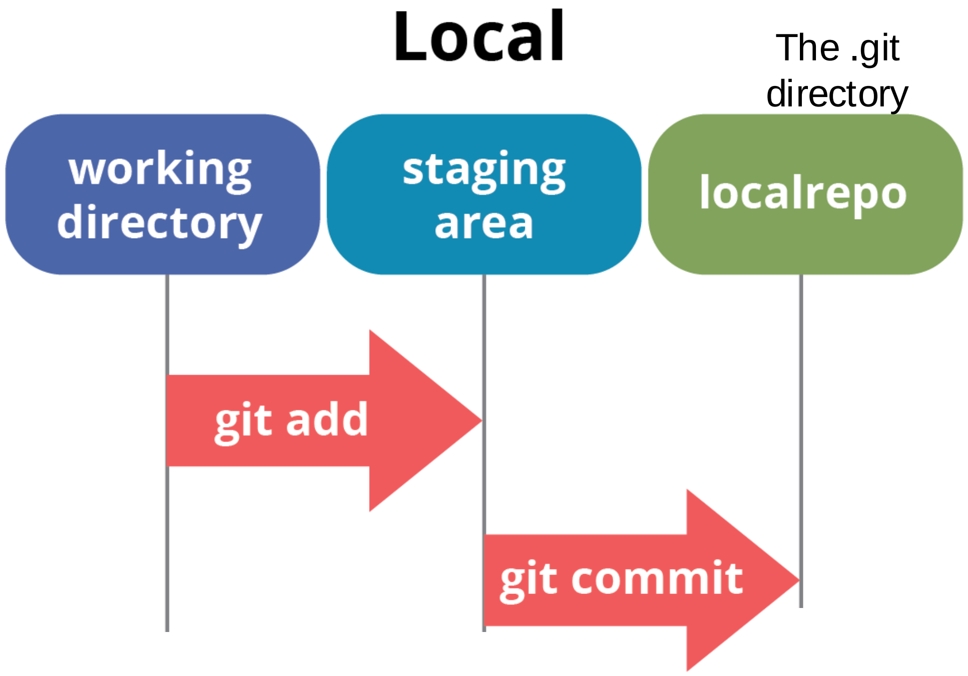
https://xkcd.com/1597/

A very short introduction to Git and GitHub

Why?

- Keep a history of changes in a project.
- Revert to previous versions if needed.
- Work collaboratively.

The Basic Git Workflow

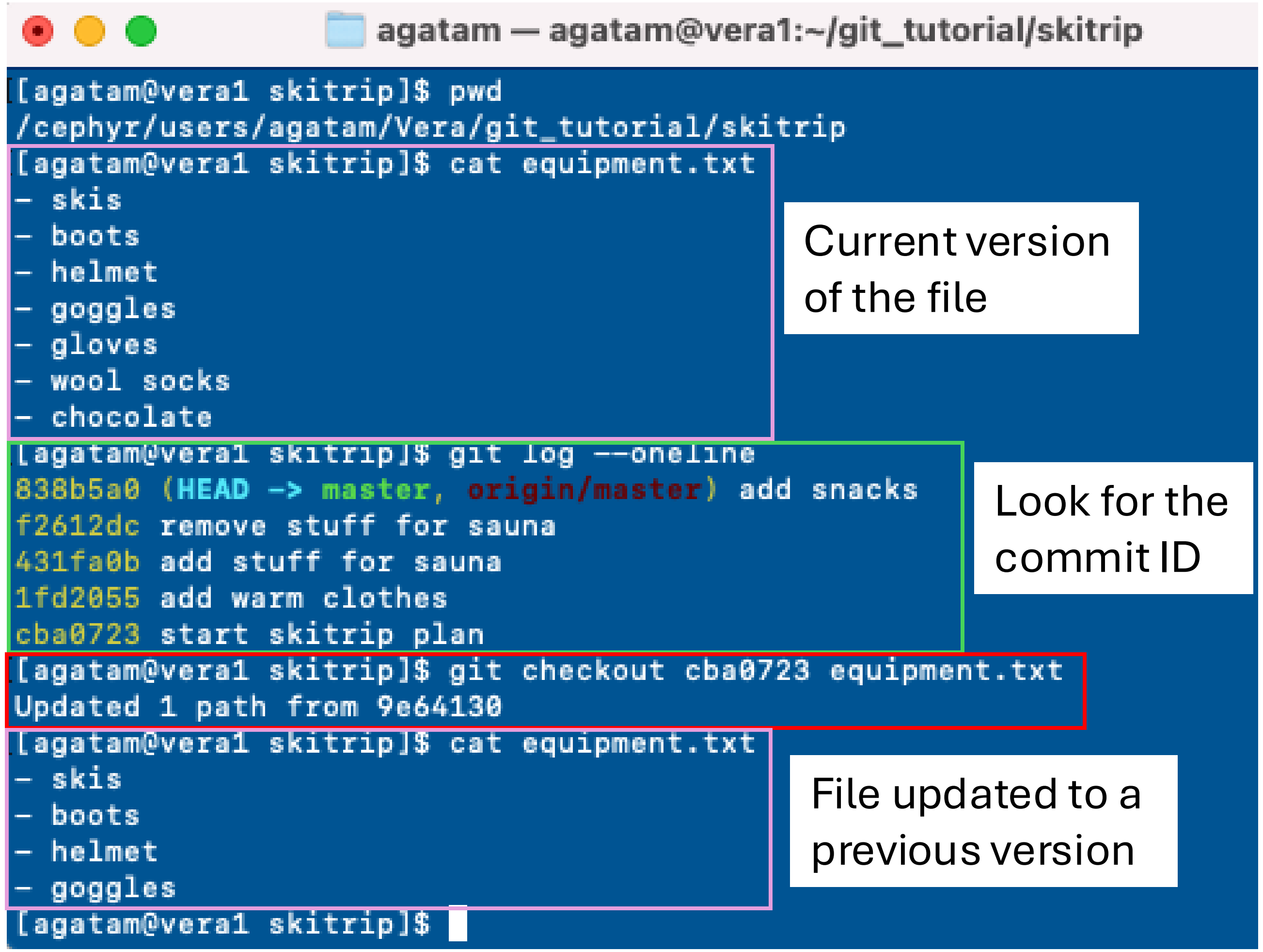


1. Modify a file in your local working directory.
2. Add changes to the staging area (or *index*)
3. Commit the changes

The **staging area** is a space where Git temporarily holds changes before they are committed. It acts as a middle step between modifying files and saving them to the repository history. This allows you to carefully choose which changes to commit, instead of committing everything at once.

When you commit, the changes from the staging area are saved as a **commit.** Git records a snapshot of the project at that point in time.

Restore a file from a previous commit



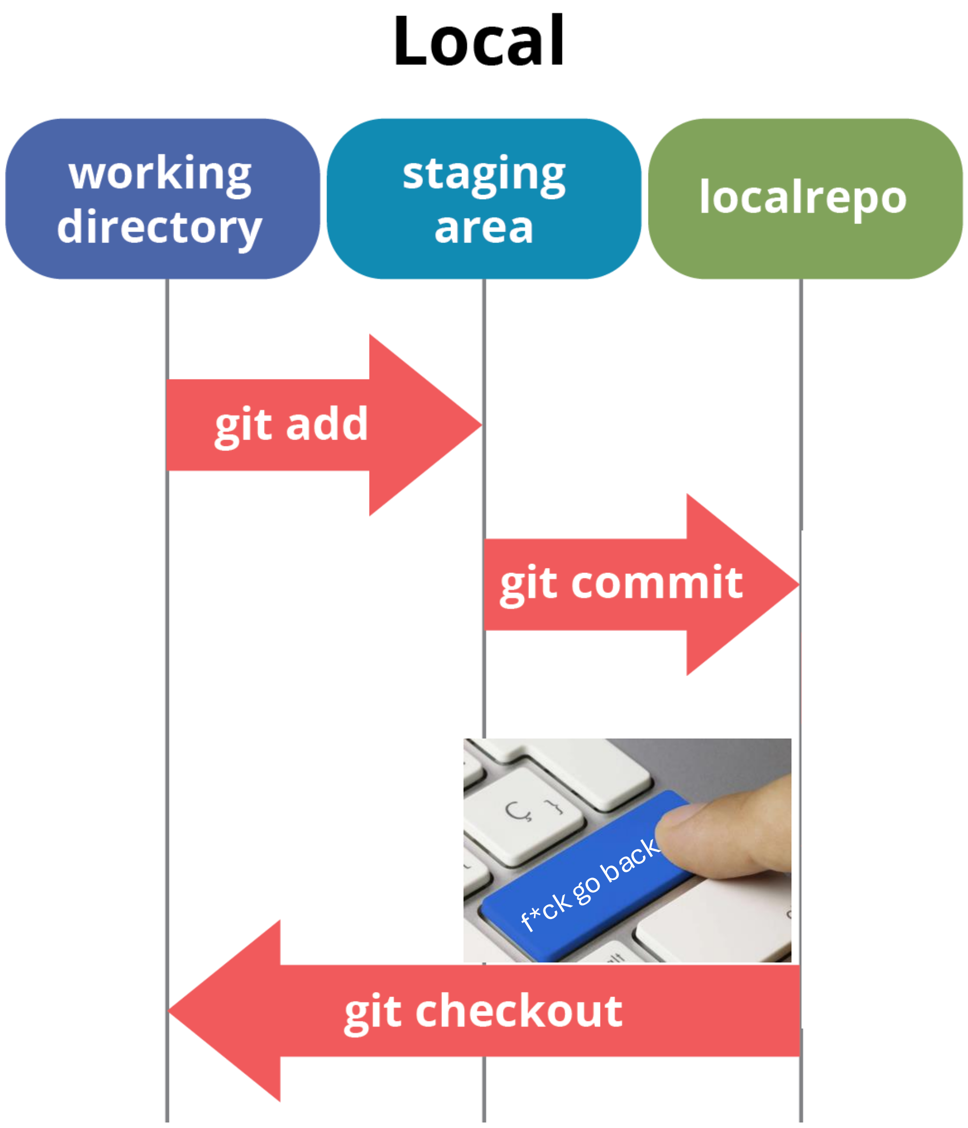
agatam — agatam@vera1:~/git_tutorial/skitrip

```
[agatam@vera1 skitrip]$ pwd
/cephyr/users/agatam/Vera/git_tutorial/skitrip
[agatam@vera1 skitrip]$ cat equipment.txt
- skis
- boots
- helmet
- goggles
- gloves
- wool socks
- chocolate
[agatam@vera1 skitrip]$ git log --oneline
838b5a0 (HEAD -> master, origin/master) add snacks
f2612dc remove stuff for sauna
431fa0b add stuff for sauna
1fd2055 add warm clothes
cba0723 start skitrip plan
[agatam@vera1 skitrip]$ git checkout cba0723 equipment.txt
Updated 1 path from 9e64130
[agatam@vera1 skitrip]$ cat equipment.txt
- skis
- boots
- helmet
- goggles
[agatam@vera1 skitrip]$ 
```

Current version of the file

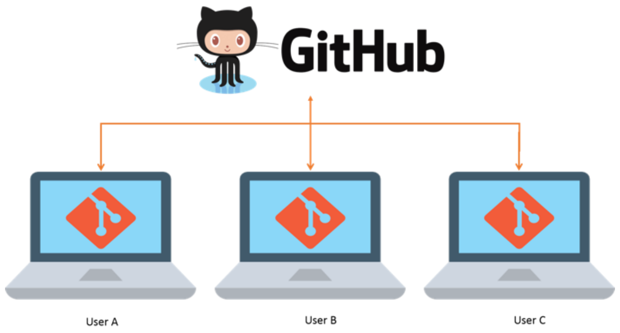
Look for the commit ID

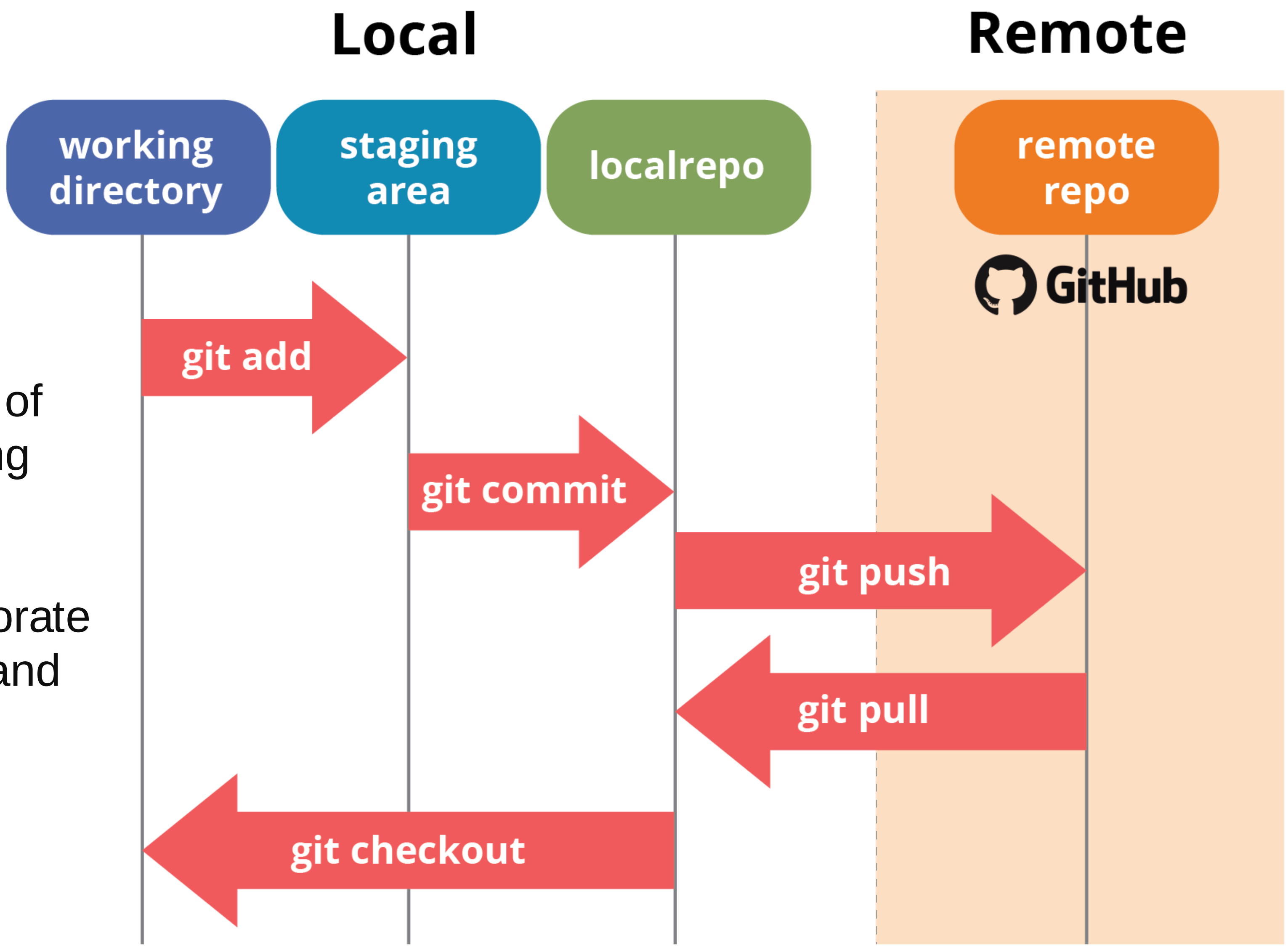
File updated to a previous version



Collaborating on the same project

- A **remote repository** is a version of your Git project stored on a hosting service like GitHub.
- It allows multiple people to collaborate on the same project by **pushing** and **pulling** changes.

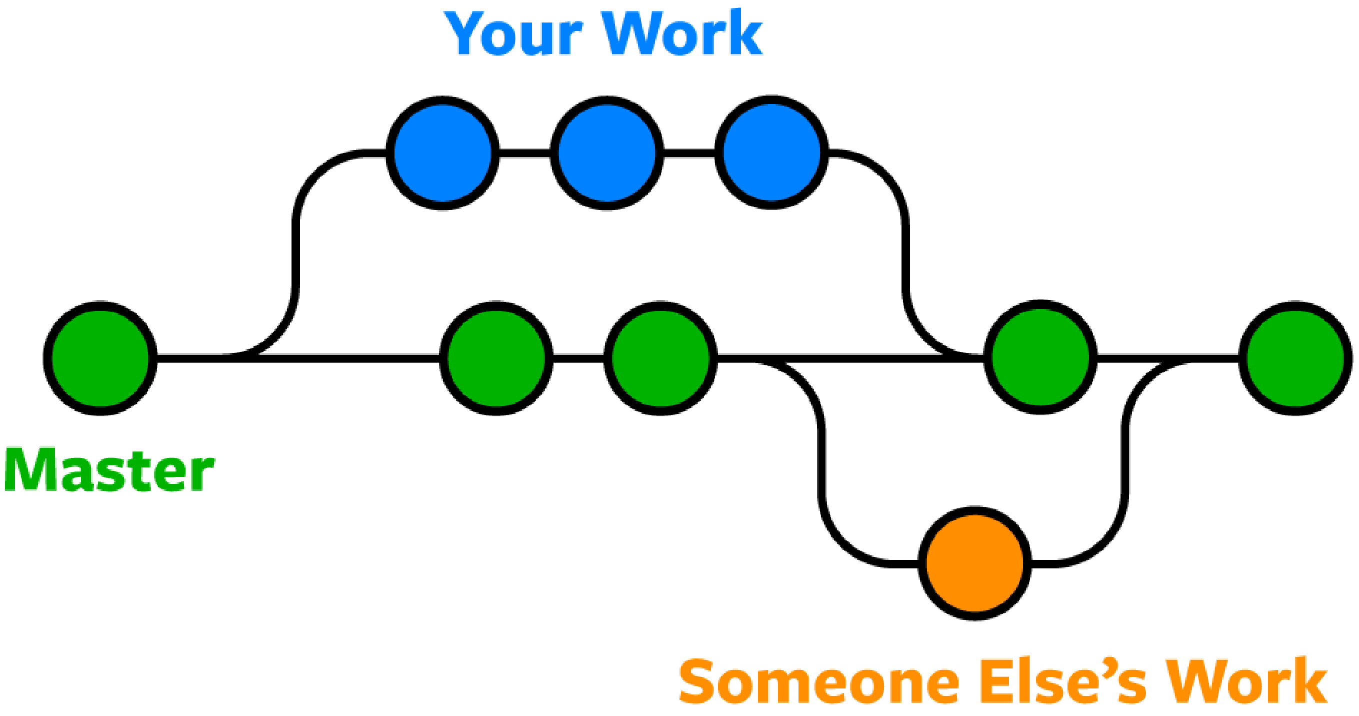


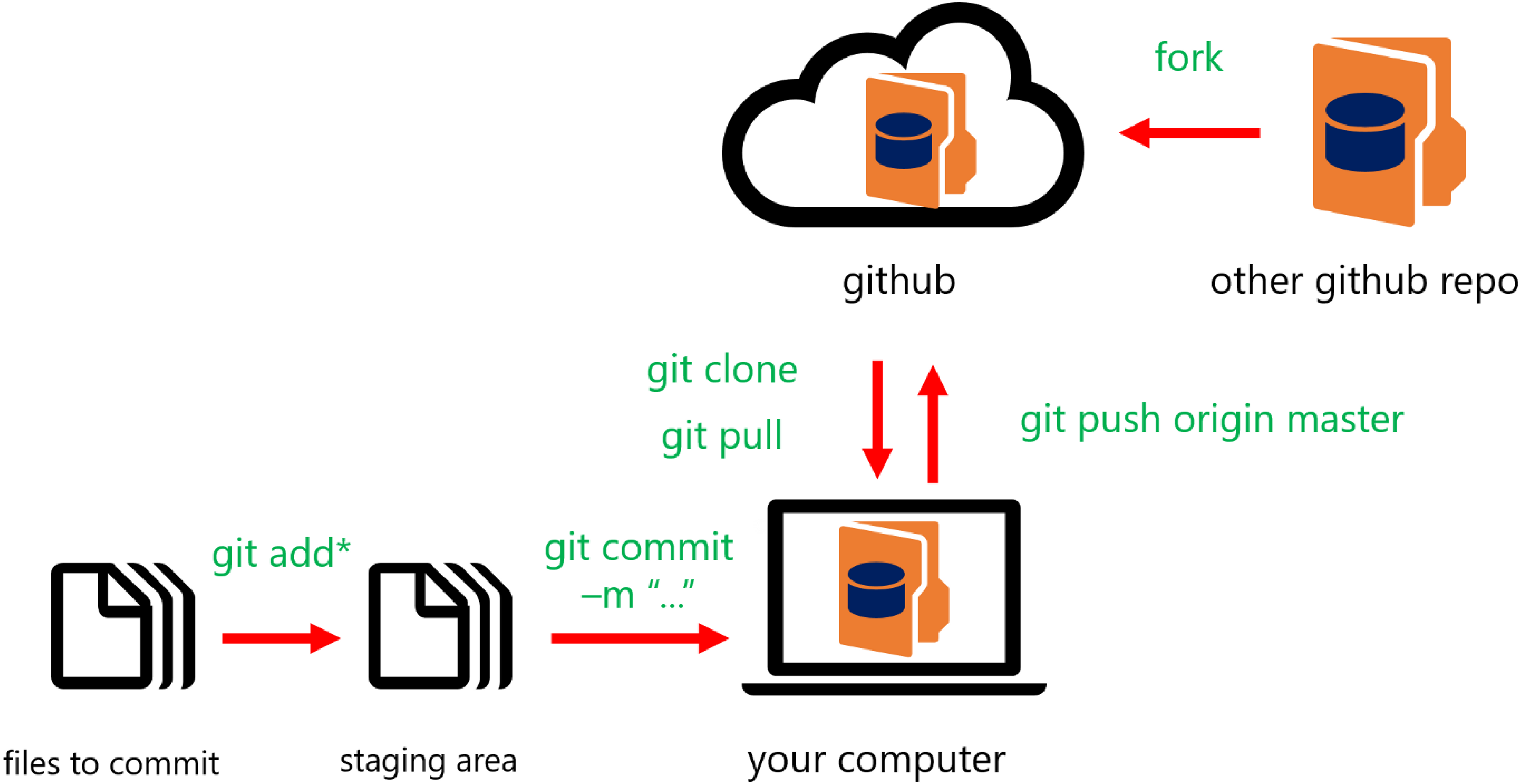


Branches

- **Isolate work** → Keep feature development separate from the stable code = Try new ideas without breaking the project.

- **Parallel and collaborative development** → Multiple people can work on different branches at the same time then share and merge changes.





fork

github

other github repo

git clone

git pull

git push origin master

git add*

git commit –m "..."

files to commit

staging area

your computer