Git and Github

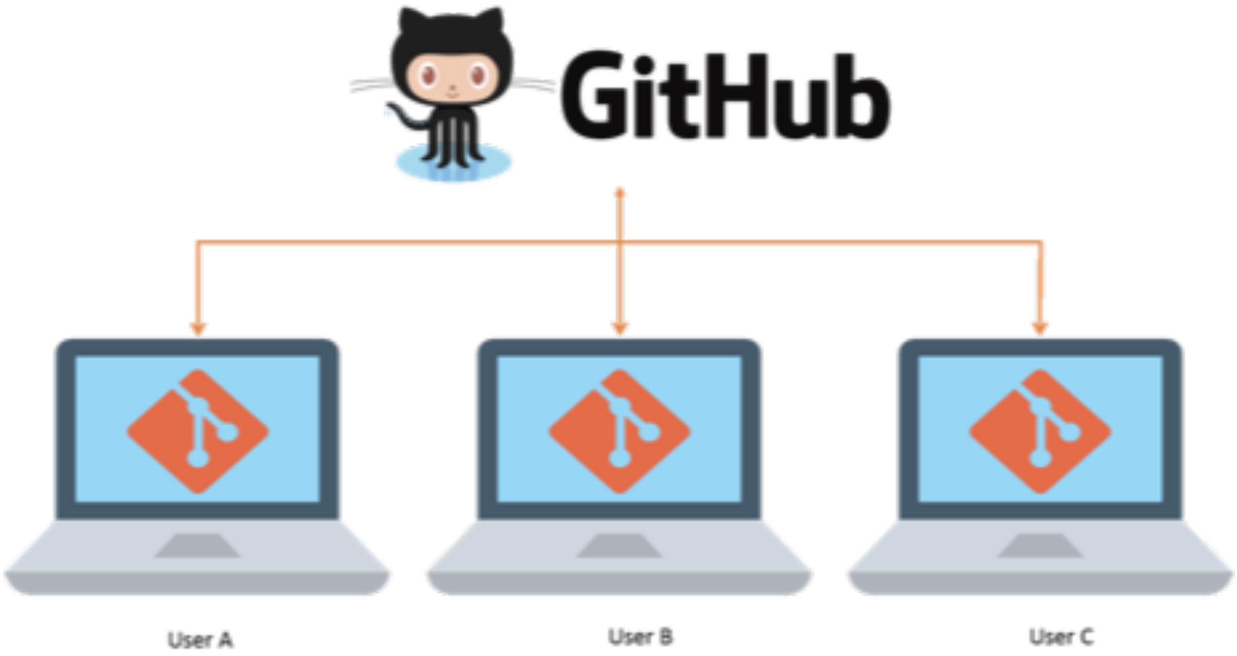
R-Ladies Boston

October 23, 2018

Abi Dawson

Git vs GitHub

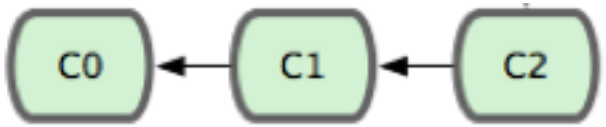
GitHub: Remote hosting and collaboration



Git: Local version control system

Git

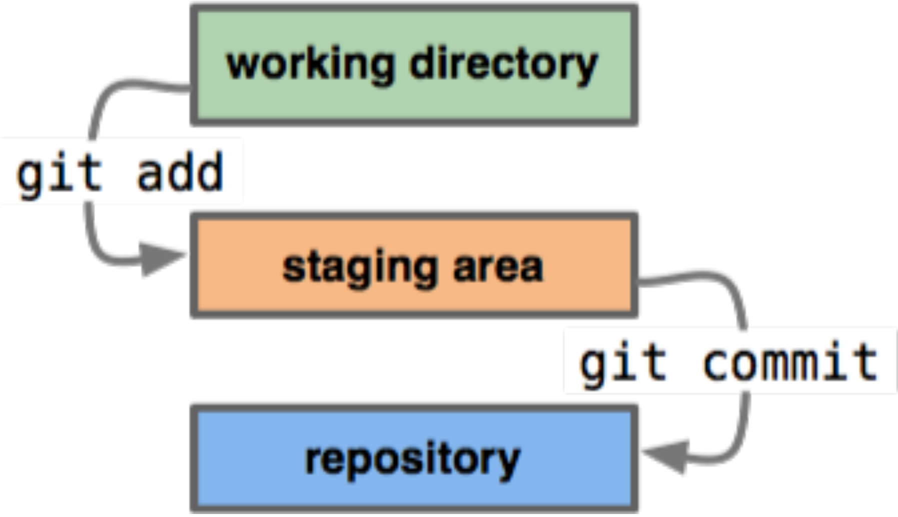
- Version control system for tracking changes to code, data, or files more generally
- A project’s history consists of snapshots called **commits**



- Key differences from other version control methods such as a revision history
  - user decides when to create a commit
  - user decides what files to include
  - leaves a helpful message describing the changes that were made (**commit message**)

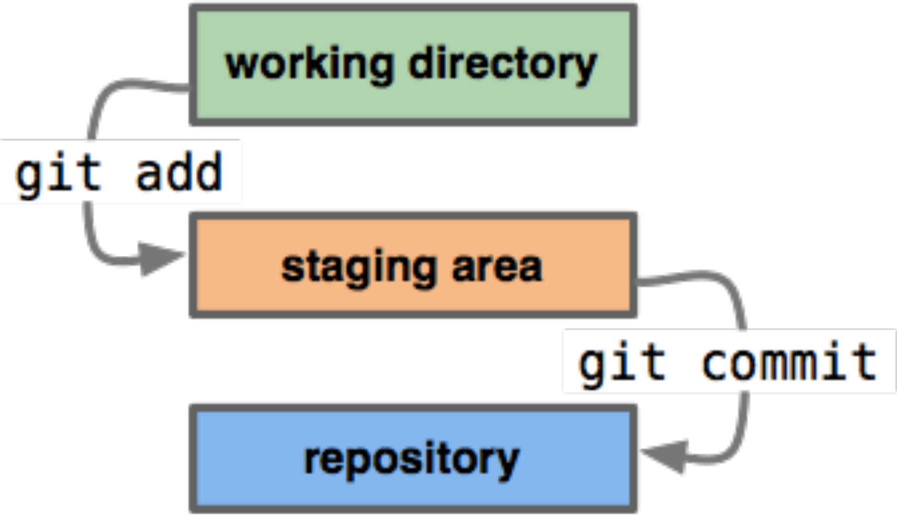
Repository

- Set of files you want to track / collaborate
- Initialize a repository in an existing directory:
  - `git init`
- Make a local copy of a remote repository (e.g. GitHub):
  - `git clone <address>`

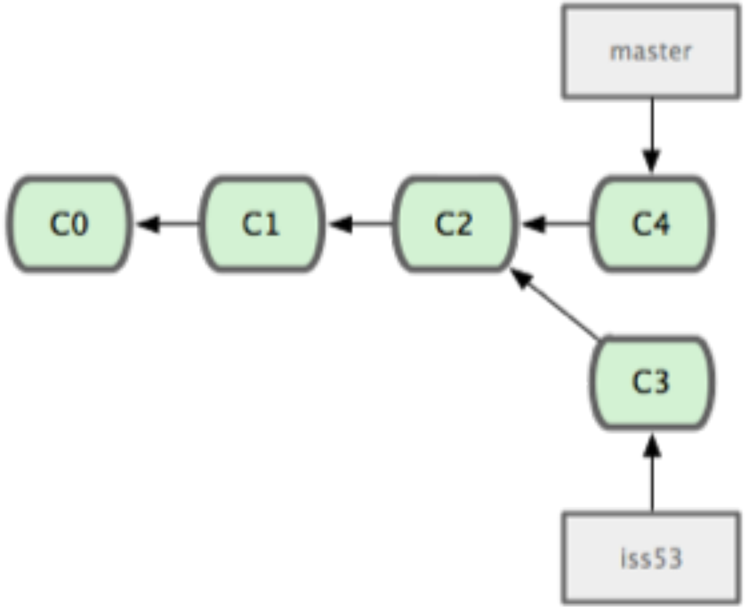


Repository

- Make changes within your **working directory**
- When those changes are ready, **add to the staging area**
  - `git add <filename>`
  - `git add .`
- **Commit** changes from the staging area to the **repository's history**
  - `git commit –m "Fix pesky bug"`
- To see your previous commits, type `git log`



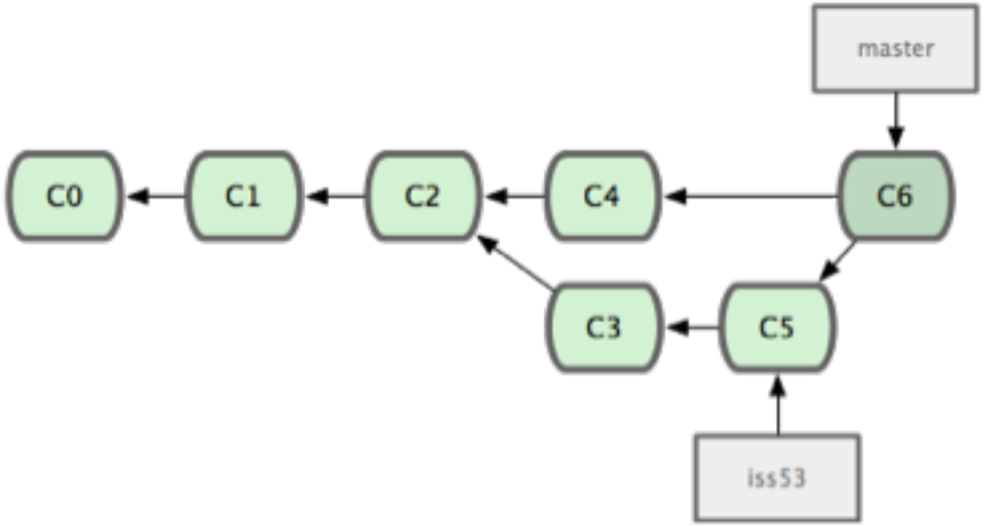
Branches



- A repository can have many **branches**
- The main branch with the production version of your code is called **master**
- To work on a new feature without causing an interruption in production use, create a **branch**
  - `git branch iss53`
  - `git checkout iss53`

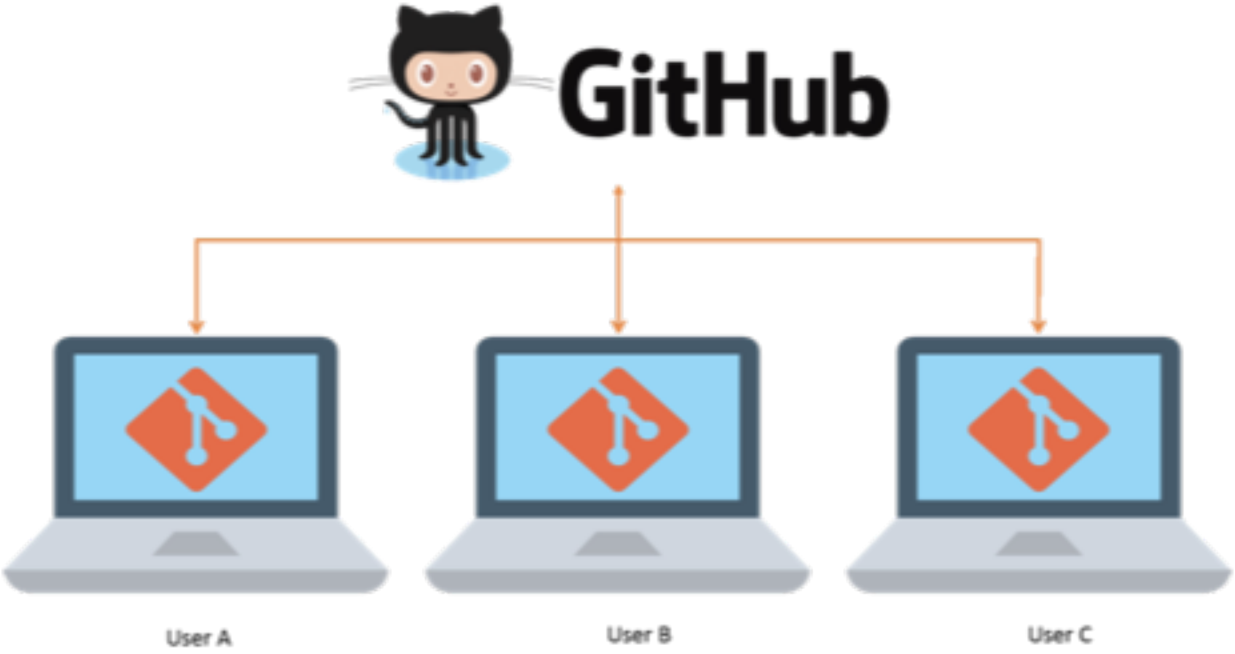
Merging

- When a feature is ready to be included in the master branch, it's time to **merge**
- Move to the master branch: `git checkout master`
- Merge the feature branch into the master branch: `git merge iss53`



Git vs GitHub

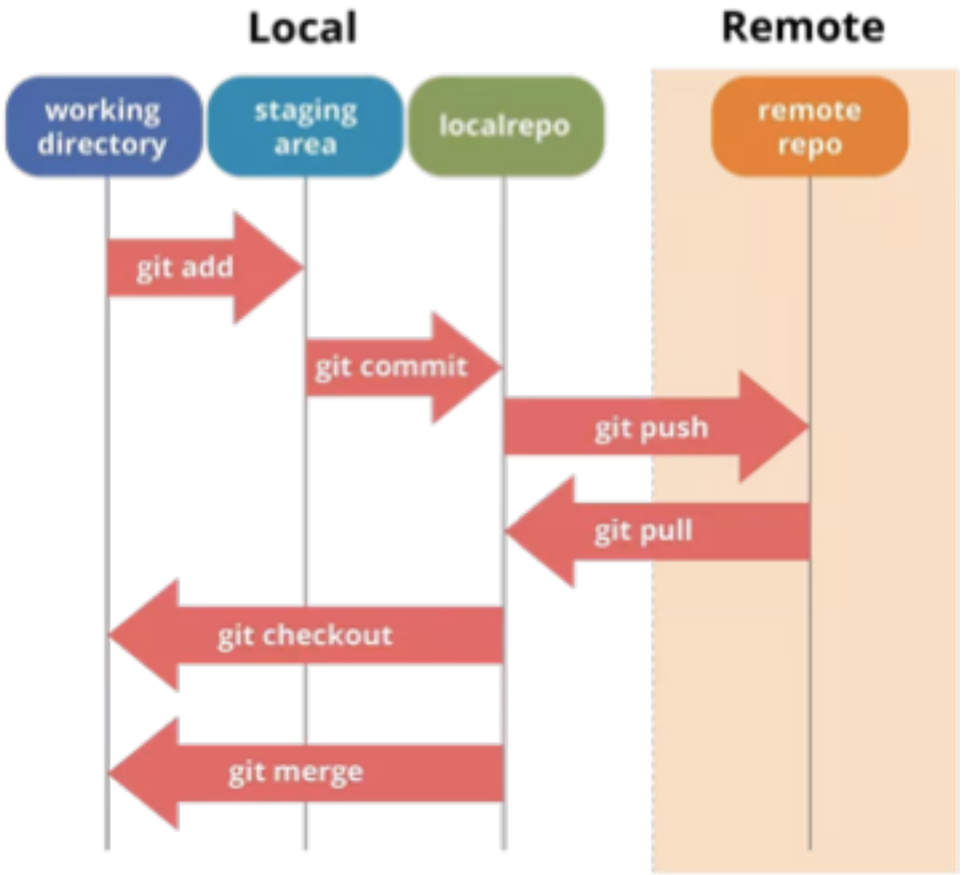
GitHub: Remote hosting and collaboration



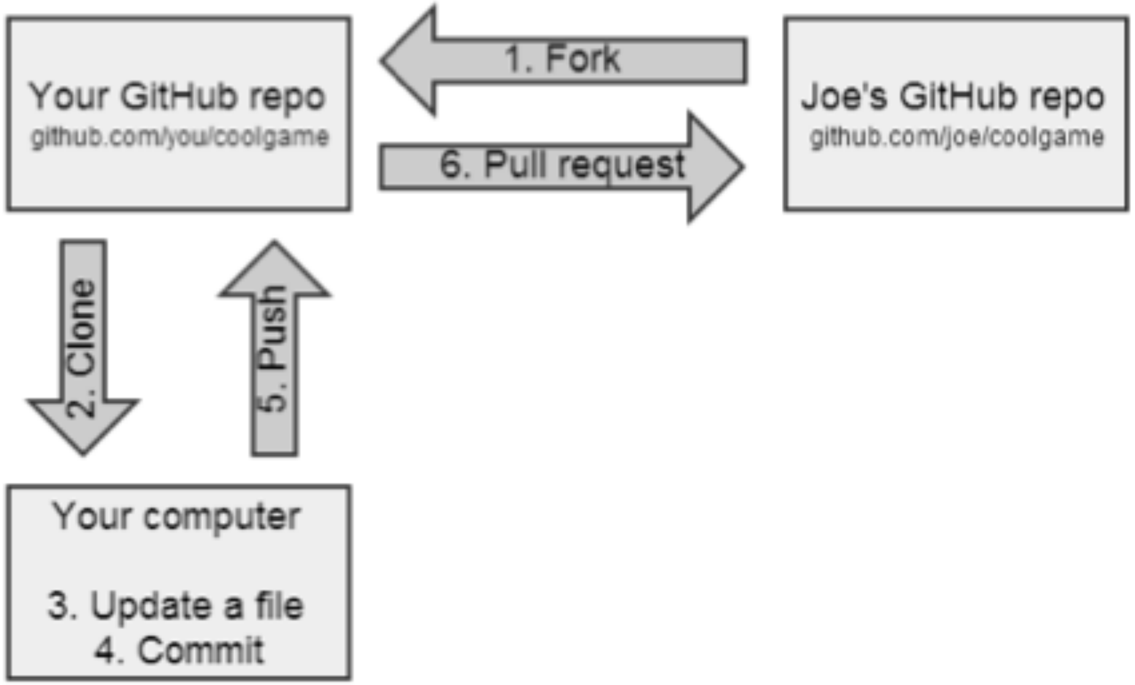
Git: Local version control system

Remote

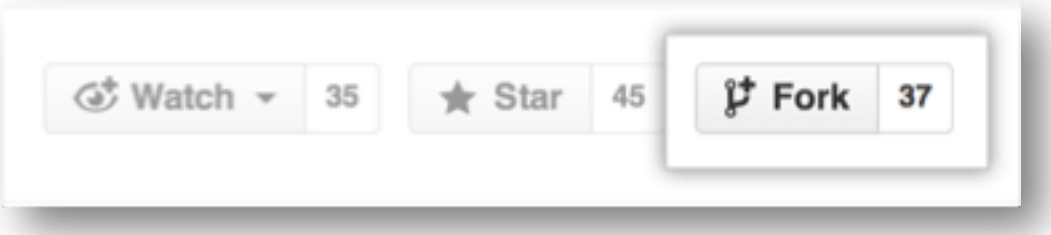
- When you work on a project locally that is also hosted on GitHub, the version on GitHub is your **remote** version
- To send your local changes to the remote, use `git push`
- To get changes on the remote in your local version use `git pull`



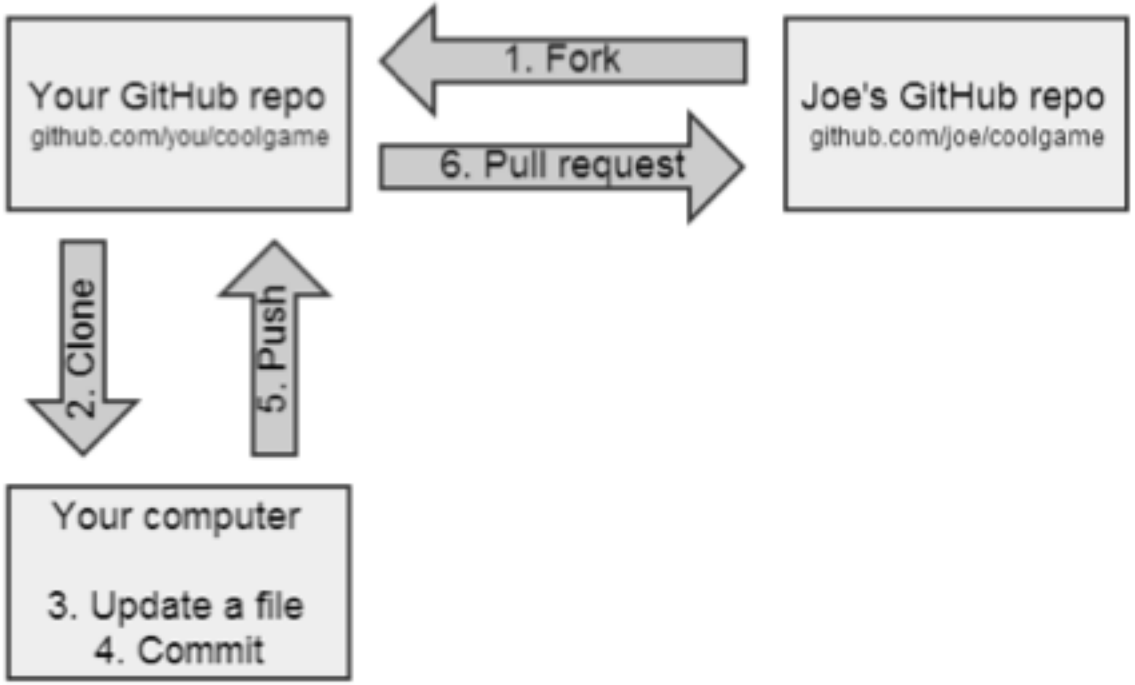
Forking



- If you want to make changes to someone else’s repository, you need to **fork** their project on GitHub
- Then **clone** your fork and make changes



Pull Request



- If you want to share your changes with the owner of the original repository, you need to submit a **pull request**
  - You are requesting that they pull your branch's changes into their master branch

Installing Git

- Check if git is already installed
  - Terminal / Command Prompt: `git –-version`
- If not…
  - Mac/Windows: Download installer from git
    - https://git-scm.com/download/mac
    - https://git-scm.com/download/win
  - Linux:
    - Install with apt/yum/dnf depending on your OS
    - `sudo apt[yum][dnf] install git`

Setting Up Git

- Configuration

```
git config --global user.name 'My_Name’
git config --global user.email 'myEmail@wherever.com’
git config --global color.ui 'auto’
```

- Ignore certain files
  - Add a file called .gitignore to your repository containing files or directories to ignore (such as .DS_Store for Mac)
  - Can also configure this globally with `git config --global core.excludesfile ~/.gitignore_global`

Additional Resources

- http://happygitwithr.com/
- https://www.udacity.com/course/how-to-use-git-and-github--ud775
- http://codingdomain.com/git/branches/ (image credit)
- https://git-scm.com/book/en/v2/
- https://www.git-tower.com/blog/git-cheat-sheet
- https://ohshitgit.com/