Online “Remote” Repositories

GitHub and Bitbucket centralized Git repositories for dissemination and collaboration

Barry Grant
bjgrant@umich.edu
http://thegrantlab.org
Recap: Client-Server vs Distributed VCS

For distributed version control systems like Git a “remote repository” (e.g. an online Git repo at GitHub or Bitbucket) promotes further dissemination and collaboration.
Your Directory — 'Staging Area' — Local Repository

- status
- add
- commit
- checkout
GitHub & Bitbucket

GitHub and Bitbucket are two popular hosting services for Git repositories. These services allow you to share your projects and collaborate with others using both ‘public’ and ‘private’ repositories.

https://github.com  https://bitbucket.org
GitHub Raises $250 Million at $2 Billion Valuation

Capital raise puts company’s total funding at $350 million
What is the big deal?

• At the simplest level GitHub and Bitbucket offer backup of your projects history and a centralized mechanism for sharing with others by putting your Git repo online.

• GitHub in particular is often referred to as the “nerds FaceBook and LinkedIn combined”.

• At their core both services offer a new paradigm for open collaborative project development, particularly for software.

• In essence they allow anybody to contribute to any public project and get acknowledgment. [We will demo this later!]
Public contributing by ‘fork & pull’

For any public project on GitHub or BitBucket you can make any change you like - that is you don’t first need permissions to contribute your improvements/bug-fixes/ideas etc.

- There are two mechanisms for doing this:
  1. For trusted “collaborators” (via a shared repository and regular commit & push steps)
  2. Joe public (via a different ‘fork & pull request’ approach)

You don't have to beg for permission (just submit a pull request) on GitHub or BitBucket and your changes with attribution will be in the project and its history once approved!
First sign up for a GitHub account

https://github.com
Pick the FREE plan!
Your GitHub homepage

Check your email for verification request
Skip the hello-world tutorial

https://guides.github.com/activities/hello-world/
Name your repo
demo1_github
Let's push an existing repository

Change back in your Terminal/MobilXterm App

> cd ~/Desktop/git_class    # Your local repo

> git remote add origin https://github.com/YourGitHubUserName/demo1_github.git

> git push -u origin master

(Tip: you can get the long URL in step2 here from your GitHub page)
Lets push an existing repository

Change back in your Terminal/MobilXterm App

> cd ~/Desktop/git_class  # Your local repo

> git remote add origin https://github.com/YourGitHubUserName/demo1_github.git

> git push -u origin master

Congratulations!
You just pushed your local repo to GitHub!!
Check it out in your web browser…
Let's edit **README** online.

Specifically, let's add some Markdown content.
Let's also edit locally…

And rename README to README.md

> mv README README.md  # Move to Markdown
> git status
> git add README.md README
> git push -u origin master

# What happened and why?
We need to **pull** changes first!

```bash
> git pull origin master  # Sync from GitHub
> git status             # What does the msg mean?
> git push -u origin master  # Sync to GitHub
```
Success!

Let's look at how GitHub presents your commit history.
Demo 2

https://github.com/bioboot/demos2-github
Summary: **Forking, Pull requests & code review**

- Using these three steps you can contribute any public project even though you don’t have write access.

- You first “fork” the repo you are interested in. This creates a completely separate copy of the repo by cloning it and adding a copy to YOUR GitHub (or Bitbucket) account.

- You then make your changes (in your forked repo) and submit a pull request back to the original repo.

- These undergo code review and, if approved, subsequent merging into the original repo.
Side-Note: Keeping your fork up to date

- When the central repository is updated with someone else’s code (after your fork was created), these new commits do not magically appear on your fork.

- You will need to add a link to the original upstream central repository to be able to pull changes.

```
> git remote add upstream https://github.com/bioboot/demo2-github.git
> git remote -v
> git pull upstream master
```

# Can you now push to ‘upstream’?
Optional: Further self exploration of BitBucket/GitHub/Git features

• **Issues** are integrated into repos and enable bug tracking, feature requests, to-do items, questions etc. on a per project basis. E.G.
  
  https://bitbucket.org/Grantlab/bio3d/issues

• **Websites** and Wikis. For example I have course websites hosted on GitHub and authored collaboratively via git. See:
  
  http://w16.bioinfquiz.org

• Some **git commands** to play with:
  
  > git show
  > man git-blame
  > git log --stat
  > git log --graph --oneline
Side-Note: Identifying how to contribute to an open source project

- Oftentimes open source projects place a `CONTRIBUTING.md` file in the root directory.

- It explains how a potential contributor should format code and submit patches etc. Here is a fine example from the `ggplot2` R package.

- From a maintainer's point of view, the document succinctly communicates how best to collaborate.

- And for a contributor, one quick check of this file verifies their submission follows the maintainer's guidelines.
Summary

• Git is a popular ‘distributed’ version control system that is lightweight and free.

• GitHub and BitBucket are popular hosting services for git repositories that have changed the way people contribute to open source projects.

• Introduced basic git and GitHub usage and encouraged you to adopt these ‘best practices’ for your future projects.
Learning Resources

• **Set up Git.** If you will be using Git mostly or entirely via GitHub, look at these how-tos.
  <https://help.github.com/categories/bootcamp/>

• **Getting Git Right.** Excellent Bitbucket git tutorials
  <https://www.atlassian.com/git/>

• **Pro Git.** A complete, book-length guide and reference to Git, by Scott Chacon and Ben Straub.

• **StackOverflow.** Excellent programming and developer Q&A.
  <http://stackoverflow.com/questions/tagged/git>
Learning git can be painful!

However in practice it is not nearly as crazy-making as the alternatives:

- Documents as email attachments
- Hair-raising ZIP archives containing file salad
- Am I working with the most recent data?
- Archaeological “digs” on old email threads and uncertainty about how/if certain changes have been made or issues solved
Finally Please remember that GitHub and BitBucket are PUBLIC and that you should cultivate your professional and scholarly profile with intention!