What about GitHub?

Now that you have some understanding of Git, you will learn about “GitHub.”

We'll cover the following

- GitHub
  - Upstream? Downstream?
- Centralized vs. distributed VCS workflow

Earlier it was noted that:

ALL GIT REPOSITORIES ARE BORN EQUAL!

In practice, some repositories are more equal than others (i.e., GitHub). This is a matter of convention within a project.

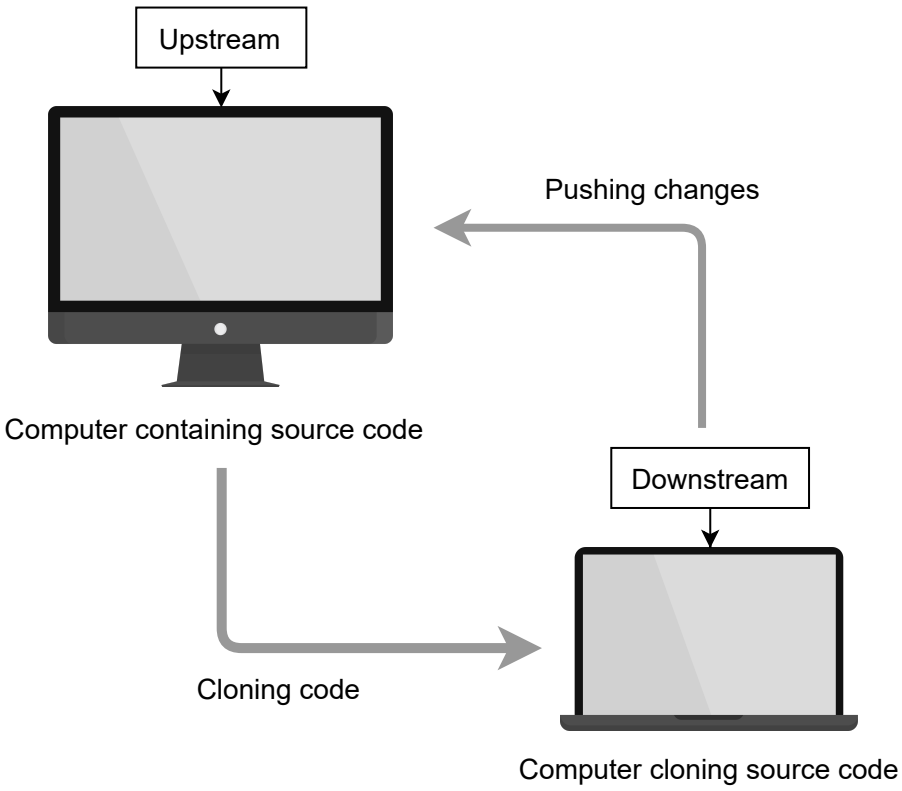
GitHub

Most people use GitHub as their *reference* or *upstream* repository (i.e., the “primary” one), but it could easily be used as a *secondary* or *downstream* repository for a workflow. It’s up to me (indeed, I do this for some of my private repositories).

Upstream? Downstream?

The phrases such as “upstream” and “downstream” are not always well-known (but many assume they are). In coding terms, upstream means the repository (or repositories) from which the source code is taken, and

downstream means the repository (or repositories) that takes changes to the code from the upstream.



Upstream and downstream

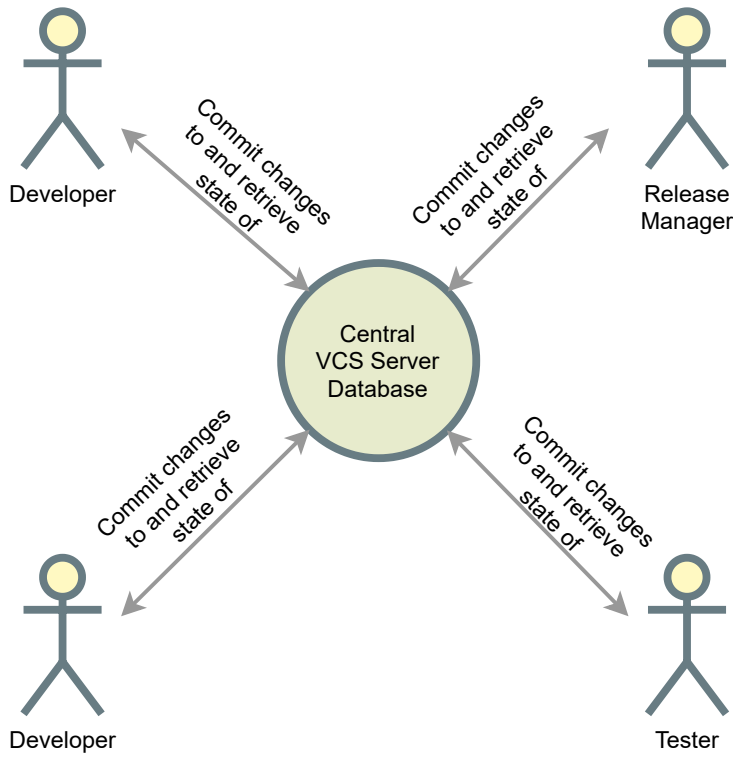
As with a stream of water, if you are downstream, you receive whatever floats along the water, and you have to “push” changes upstream.

GitHub’s *de facto* status as a centralized repository (and all the machinery that assumes its existence and continuous uptime) is the reason every GitHub outage causes a flurry of smart-alec comments about Git supposedly being a *decentralized* source control tool that relies on one *central* system.

More seriously, being a distributed source control tool makes Git more challenging to understand than the traditional version control system (VCS) (a.k.a. source control management (SCM)) tools, which is one of the reasons why services like GitHub become central references. It is to keep things simple for the typical user.

Centralized vs. distributed VCS workflow

Here's some diagrams that might help describe the differences between the decentralized and server-based VCS tools:



Central VCS workflow

1 of 2

In a distributed source control, rather than everyone talking to one central source code "server," the tool operates as a set of separate repositories that can pull and push changes to each other if the "owner" of the repository agrees.

This is why "pull request" exists. When you make a pull request, you are saying to the owner of the other repository: "please take a change I've made in my repository and apply it to yours." The operation could just as

easily go the other way!

No GUIs here

This course focuses on *core* Git rather than GitHub and the command line rather than GUIs (Graphical User Interfaces). This is for a few reasons. One reason is that GUIs differ and can mislead you about what is going on under the hood. In turn, this can be confusing when you are forced to use (for example) BitBucket instead of GitHub. Finally, it is easier to understand *core* Git and then map that to GUIs rather than the reverse.

The first step to mastering Git is understanding this equality of repositories.

Back

Branches

Next

How Git Differs from Other Version Co...

Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/what-about-github__introduction-to-git__learn-git-the-hard-way)