**Forking V/S Cloning**

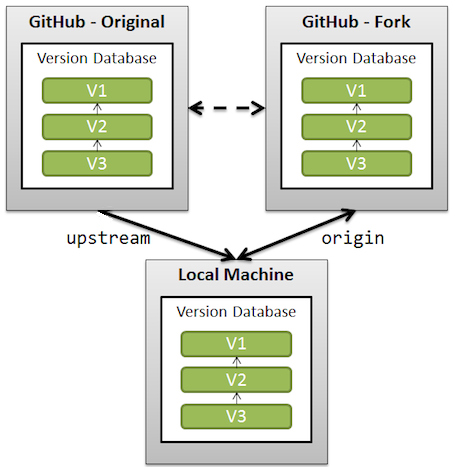
**Forking a repository:**

A fork is a copy of a repository. Forking a repository allows us to freely experiment with changes without affecting the original project. In other words, a fork is a personal copy of a repository that we can use as we wish.

Basically, this is doing a **cloning a GitHub repo at GitHub** before cloning that fork locally to our machine.

The difference between clone and fork will become clear if we understand what the fork is.

Most commonly, forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for our own.



Propose changes to someone else's project

A great example of using forks to propose changes is for bug fixes. Rather than logging an issue for a bug we have found, we can:

1. Fork the repository.
2. Make the fix.
3. Submit a pull request to the project owner.

If the project owner likes our work, they might pull our fix into the original repository!

Use someone else's project as a starting point for our own idea

At the heart of open source is the idea that by sharing code, we can make better, more reliable software. In fact, when we create a repository on GitHub, we have a choice of automatically including a license file, which determines how we want our project to be shared with others.

Note: when we are cloning a GitHub repo on our local workstation, we cannot contribute back to the upstream repo unless we are explicitly declared as "contributor". So, when we clone a repository to our local workstation, we are not doing a fork. It is just a clone.

**Origin vs Upstream:**

1. **upstream** generally refers to the original repo that we have forked.
2. **origin** is our fork: our own repo on GitHub, clone of the original repo of GitHub.

According to GitHub page:

"When a repo is cloned, it has a default remote called **origin** that points to your fork on GitHub, not the original repo it was forked from. To keep track of the original repo, you need to add another remote named upstream":

$ git remote add upstream git://github.com/user/repo.git

1. We will use **upstream** to fetch from the original repo (in order to keep our local copy in sync with the project we want to contribute to).
2. We will use **origin** to pull and push since we can contribute to our own repo.
3. When we're doing git push origin branchname, we're pushing to the **origin** repository. There is no requirement to name the remote repository origin, and there can be multiple remote repositories. Remotes are simply an alias that store the url of repositories. We can see what url belongs to each remote by using git remote -v. In the push command we can use remotes, or we can simply use a url directly.
4. We will contribute back to the **upstream** repo by making a pull request.