Source and Version Control

Control Your Destiny

Christian Grewell Follow
Aug 13, 2018 · 3 min read

What is Source Control?

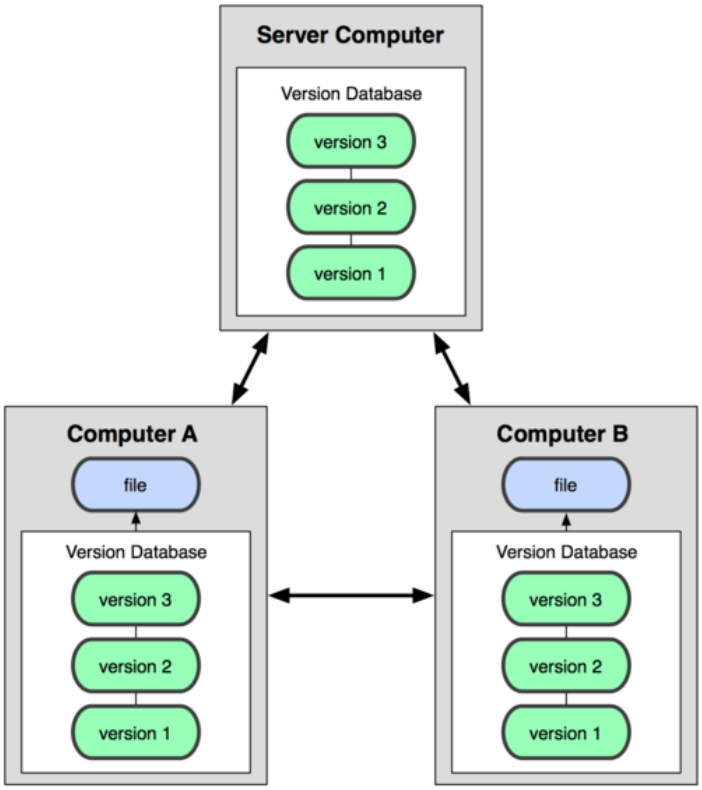
Source control is a way to control changes to files and directories, so that you can keep a record of changes and recall specific versions of a file in the event you'd like to back up to an earlier time.

You probably perform some version of *local source control*, that is, you copy files from one place and put them in other (maybe you 'backup' your files to a USB drive). This approach is quite common, but also extremely error prone. It is very easy to overwrite a file, or accidentally copy files you never mean to.

In the past, programmers developed local version control systems that were essentially databases that stored numbered versions of files that needed to have tracked changes.

Distributed Version Control

In this course, we're going to use distributed version control, specifically, we're using Git + Github.com. Your local computer has git installed, but there's a challenge if you want to share your files with others. In a distributed source control system, *each user* has an entire copy of the version database, and essentially synchronizes changes among two or more of those repositories. Think if it like Google Docs with versioning.



https://git-scm.com/book/en/v1/Getting-Started-About-Version-Control

Still confused? Ok. Try this video!

Benefits of Source Control

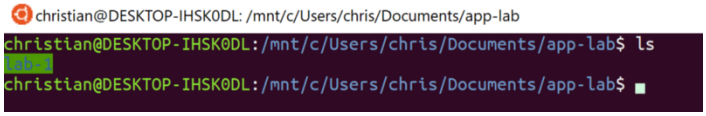
There are a few big benefits to using source control. First, I'm going to know where to go to find your code, and others can help you if you run into trouble (just send us your repo URL). There's never a disagreement about what the correct code is because it's all right there in the repo.

Second is that it will make your final and group projects incredibly simple to coordinate activities and changes. You'll avoid the horror show where everyone is using different code because everyone is now sharing the same code through source control during development. Third, and most important, with source control there's a **log of every change made to the code**. If something is broken you can go back and fix it. You also know who broke it :)

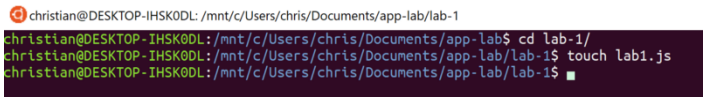
Source Control in This Course

So in terms of how we'll use GIT + Github in the course, complete the following tasks, and then watch the video below to further familiarize yourself with how to use Git and Github for code versioning and collaboration.

- Sign up for a Github.com account (www.github.com)
- Create a repository named **app-lab**
- Initialize an empty git repository on your local machine (it'd be helpful if you named it app-lab as well)
- Create a folder for our first lab



- Use the **touch** command to create a lab1.js file in app-lab/lab-1/



- Enter the following commands in your CLI from the app-lab directory

```
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/clg236/applab.git
git push -u origin master
```

- Post your Github repository URL to the **#lab-1** channel in Slack

Having Trouble? Check out this video below where I illustrate the process of creating a simple program, initializing a local git repository, and pushing changes to Github.

Sorry, this is 15 minutes :(