How To Git

Matt McCarthy

12 March, 2016

1 Version Control

2 Git 101

To start, we need to configure `git`. To do so, we run the following two commands.

```
git config --global user.name "Your Name Here"
git config --global user.email "your.email@host.domain"
```

These commands tell our `git` installation who we are and how to contact us so that other users know who is responsible for each commit.

Our first task is to make a repository on our local machine. Next we want a directory in which we will store our `git` repositories. For the sake of simplicity, let's just make a new folder in the home directory called `git` (e.g. run `mkdir ~/git`). We now need to `cd` into our new directory, so we run `cd ~/git`. Now run `mkdir my-git-repo` and then `cd my-git-repo`. We will now turn this folder into a `git` repository by running `git init`. And now we have a `git` repository.

We're now going to start making changes, tracking them, and committing them. Let's begin by creating a file and telling `git` to track it. Run `touch file.txt`, this will create a file called `file.txt`. If we run `git status`, it will list `file.txt` as an untracked file. We now need to run `git add .`. The previous command tracks all untracked files and tracks any changes you made. If we run `git status` again, we will see that `new file: file.txt` is in the list of changes to be committed. Lastly, to commit our changes, we run `git commit -m "Added file.txt"`. This logs our changes and gives us a point to which we can revert. If we run `git status` once more, it will report that there is nothing to commit and that the working directory is clean. The `git add .` and `git commit -m "message here"` commands define the workflow on a single machine, that is these commands track and log each change you make to your project.

3 Git Demonstration

4 Advanced Git