Intro to Git

Shyam Srinivasan

October 5, 2017

Learning expectations

This won't even touch the surface of git... But! Hopefully, you:

- Recognize the power of using version control
- Know the basic tools to get started using git
- Where to go for help

But! No expectation to actually start using it :)

What is Git?

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Figure 1:

Getting started with Git and GitHub

Create an account on GitHub

- You set your username, email and password here

Windows/MacOS

`https://desktop.github.com/`

- Make sure you install posh-git along with GitHub

Linux

- Download and install Git for Ubuntu/Debian/Mint Linux `sudo apt-get install git`
- Download and install Git for Fedora Linux `sudo dnf install git`

Configuring your setup:

```
git config --global user.name "Your Name"
git config --global user.email "you@some.domain"
```

Optional Configurations

```
git config --global color.ui "auto"
```

Setup Editor for Text Files

```
git config --global core.editor "your_editor"
git config --list # To confirm
```

Creating a Repo on Github

The cloud (GitHub) serves as an online repo where files are accessible 24x7

- Go to GitHub
- Create a new repo by pressing the `New Repository` button
- Let us keep the name short and self explanatory and call it 'git-training'
- We will call this the `remote` repo from here on

Creating a Git Repo

- change directory (move to another folder) using

```
cd
```

- to move back one folder use

```
cd ..
```

- to make a new directory

```
mkdir
```

Let us create a local Repo

```
cd ~/Documents
# for windows users with full path
cd C:/Users/user_name/Documents
# create a new folder
# use the same name as your remote repo on GitHub
mkdir git-training
# move to the new folder
cd git-training
```

Convert folder to **git** repo

```
git init
```

check your folder to find git configuration files/folders

This folder is now a git repo (short for repository)

Using GitHub with Git

If you have GitHub (on Windows/MacOS), you can add the repository you cloned/created to it and visualize the changes

Remember, you installed this earlier?

Configuring your repo for use with GitHub on the cloud

In order to use this local repo with the one you created earlier we need to set the remote configuration

```
git remote -v
```

will tell you what is the remote corresponding to your local repo

If it is empty, you can setup a remote using

```
git remote add remote_name remote_url
```

Before you do this, you need to have an account on GitHub and a repo on it

Configuring your repo for use with GitHub on the cloud

- Copy the repo url from the repo creation page and use it in

```
git remote add origin
https://github.com/UofTCoders/git-training.git
```

- Check your remote configuration again

```
git remote -v
```

Adding new files to a repo

- create a text file with
  - your name
  - program
  - research area
- save the file as `my_cred` in the git repo you just created

Visualize changes to files

You can see the changes you made to this file in GitHub

New files

on GitHub can be seen with a green '+' to their right

This indicates the file is new and is `untracked` by git

Old modified files

can be seen with a gray circle to their right

This indicates the file is already `tracked` and there are `uncommitted` changes

```
git status
```

can give information on `tracked`, `untracked` and `indexed` files

Adding Files to git index

- add the file to the git index

```
git add my_cred.txt
```

- check file status again

```
git status
```

  Notice the file has changed from red to green

- let us `commit` this change to repo history

```
git commit -m "initial commit with my info"
```

- Now your file is `tracked` by git

push to your new repo on GitHub

You can move all your **committed** changes to the remote repo

```
git push origin
```

Since this is the first time we are doing this, we need to establish a link between your local repo and remote repo.

you can do this using

```
git push -u origin master
```

Make sure your `remote` is set before you do this

Cloning an Existing git repo

We can copy or clone existing repositories from GitHub using

```
git clone remote_repo_url
```

Let us do this for the sandbox repo I have created

Select the green `Clone or Download` button and select `copy to clipboard`

In your shell/bash move to a directory of your choice and clone the repo

```
cd ~/Documents
# if you are using git posh (in powershell)
# use the right click to paste the copied url
git clone
https://github.com/UofTCoders/sandbox-2017.git
```

Modify a existing file and commit changes

Go to the directory containing the example file

```
cd sandbox-2017
# see all files in the directory
ls
```

Open file `example_file.md` in any text editor

To open files in bash/shell

```
editor_name file_name
notepad example_file.md
```

add a new line to this file starting with an asterisk(*)

save file and close

Add changes to index and commit changes to repo

We will use the previously used commands to `add` and `commit` files

```
git add example_file.md
git commit -m "add one new line to file"
```

Visualize changes to file

You can use

```
git diff example_file.md
```

in the command line

Note the file has to be `tracked` and the changes have to be `committed` to use `git diff`

You can also visualize all commits to the repo using

```
git log
```

commit information can be obtained in a cleaner format using

```
git log --online
```

push commits to the remote repo

Let us first check the remote configuration

```
git remote -v
```

Notice that the remote is already set

```
git push origin
```

will push your local changes to the remote repo

Discarding changes in your repo

Discard unstaged Tracked files using

```
git checkout .
```

Discard unstaged unTracked files using

```
git clean -f
```

Use this command with caution:

Discard staged and unstaged Tracked changes using

```
git reset --hard
```

Summary of Commands covered today

```
git init
git clone
git config
git remote
git status
git diff
git add
git commit
git push
git pull
git checkout
git clean
```