git

Duncan Nicholson & Jeremiah Ballard

Contents

What is it?

- From the Git User’s manual:

  “Git is a free and open source distributed version control system”

What does it allow us to do?

- Collaboratively write software
  - Track every individual change by each collaborator
- Keep precious “source code” safe

Git Components

- Code (Repositories or repos)
  - Contains all files (including documentation) and revisions
- Branch/ Branching
  - A parallel version of the repository
  - Allows user to work on code while program is running without interrupting
- Issues
  - Suggested improvements with code or questions
  - Created by anyone if public

Prerequisite: Bash shell & Command language

Open a terminal	CTRL+ALT+T
Print working directory	`$ pwd`
Change directory	`$ cd path/to/project`
List directory contents	`$ ls [-la]`
Make directories	`$ mkdir <pathname>`
Create files	`$ touch [filename.ext]`
Delete files	`$ rm [-rf] <file>`
Text editors:	nano, gedit, etc

Prerequisite: Bash shell & Command language

Install & Update Software*:

```
$ sudo apt-get [install | update] <package>
```

This is a tiny fraction of all the useful commands!

You can try them out for yourself (Windows SubSystem for linux)

Cloning an existing Repository

```
$ git clone https://github.com/JerryBBallard/Aero-Drone-2018.git
```

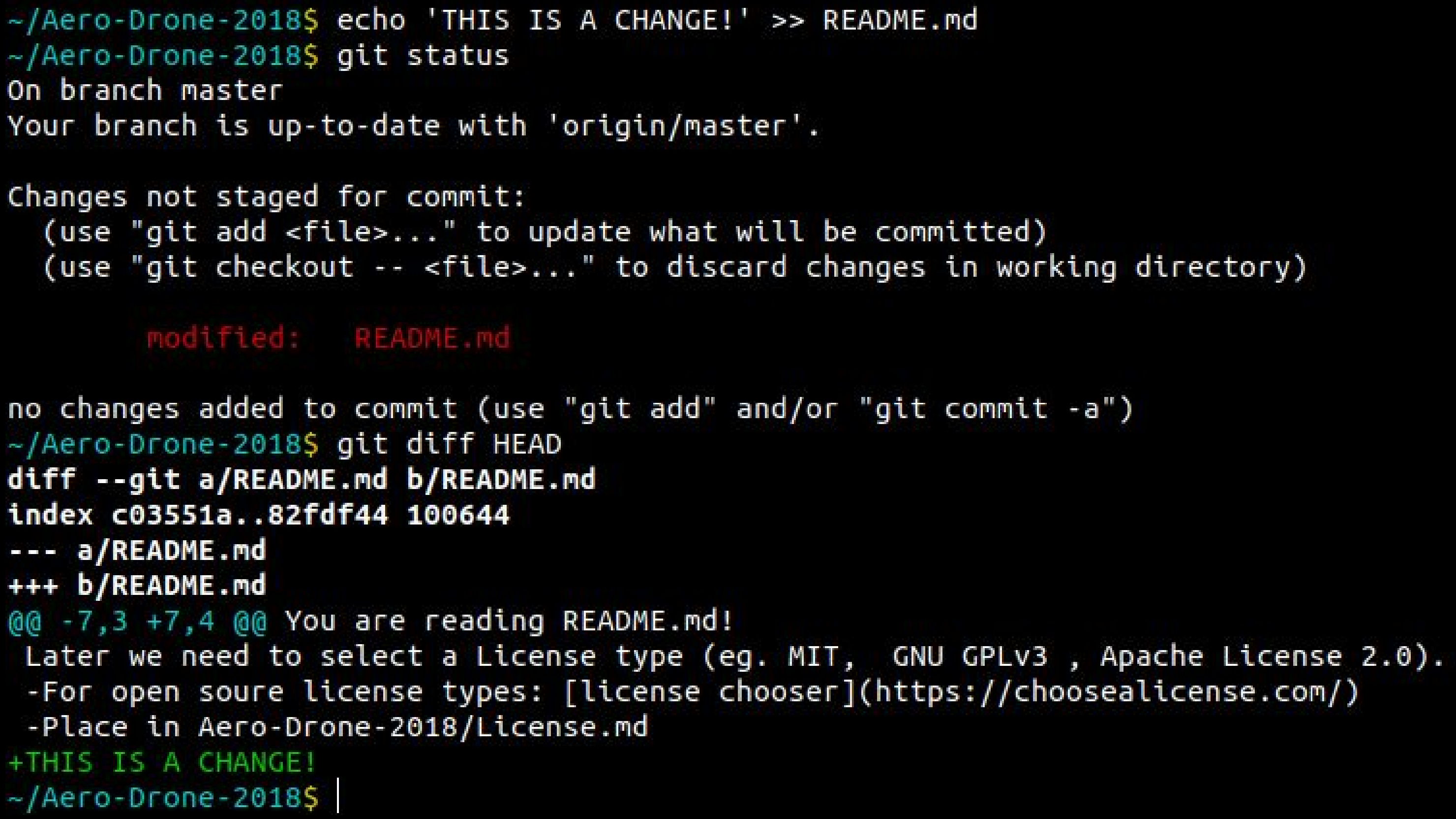
- Runs `git init`, then copies repository contents to local machine
- Changes made will be tracked by Git (even offline)
- But it is connected to a remote version of Git so the changes can be synced
- Then just push these changes to the repository

Visualizing Differences

```
$ Git diff HEAD
```

- Visually describe the changes changes between commits and saved changes
- Git tracks file *contents*

Example



```
~/Aero-Drone-2018$ echo 'THIS IS A CHANGE!' >> README.md
~/Aero-Drone-2018$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
~/Aero-Drone-2018$ git diff HEAD
diff --git a/README.md b/README.md
index c03551a..82fdf44 100644
--- a/README.md
+++ b/README.md
@@ -7,3 +7,4 @@ You are reading README.md!
 Later we need to select a License type (eg. MIT,  GNU GPLv3 , Apache License 2.0).
 -For open soure license types: [license chooser](https://choosealicense.com/)
 -Place in Aero-Drone-2018/License.md
+THIS IS A CHANGE!
~/Aero-Drone-2018$ 
```

Saving Changes

```
$ git add <file>
```

- Adds files to staging area
- Buffer between working directory and project history
- Use `git add -A` to add all

```
$ git commit -m “<message>”
```

- Commit a staged snapshot to your *local* repo
  - Version control is based on snapshots
- Buffer between your changes and the central repo
  - Option to accumulate commits in local repo

Saving Changes

```
$ git push [-u] <remote> <branch>

$ git push origin master
```

- Finally uploading to the repository
- Enter credentials

Syncing

```
$ Git Pull <repo_url>
```

- Fetch new changes and merge them with another repository or local branch
- First time? -> initialize any submodules
  - `$ git submodule update --init --recursive`
  - Afterwords update submodules as needed
    - `git pull --recurse-submodules`

Extras

- GIt Fork
  - A personal copy of another user’s repository that lives on your computer
  - Allows to make changes without affecting original.
  - Remain attached to original, and allows to submit a pull request to the author
- Git Permission
  - Public Repository
  - Private Repository
- Git Ignore
  - Specifies intentionally untracked files that Git should ignore
  - Can manually edit .gitignore

Getting Help

- Github
    - Git Cheat Sheet
    - Git User’s manual
    - Interactive tutorial
- Bash
    - Extremely comprehensive guide
    - ROS command line tools
    - https://askubuntu.com/
    - https://explainshell.com/

    - **Manual pager** `$ man <command>`
    - **Usage help** `$ <command> [arguments] [-h | --help]`

Questions?