ڤوليتيكنيك بروني
POLITEKNIK BRUNEI

WD4307
Web Application and Development Tools

Topic 02 - Git Training 2

Last Revised: 21-01-2024

Table of Contents

ڤوليتيكنيك بروني
POLITEKNIK BRUNEI

Individual work

GitHub

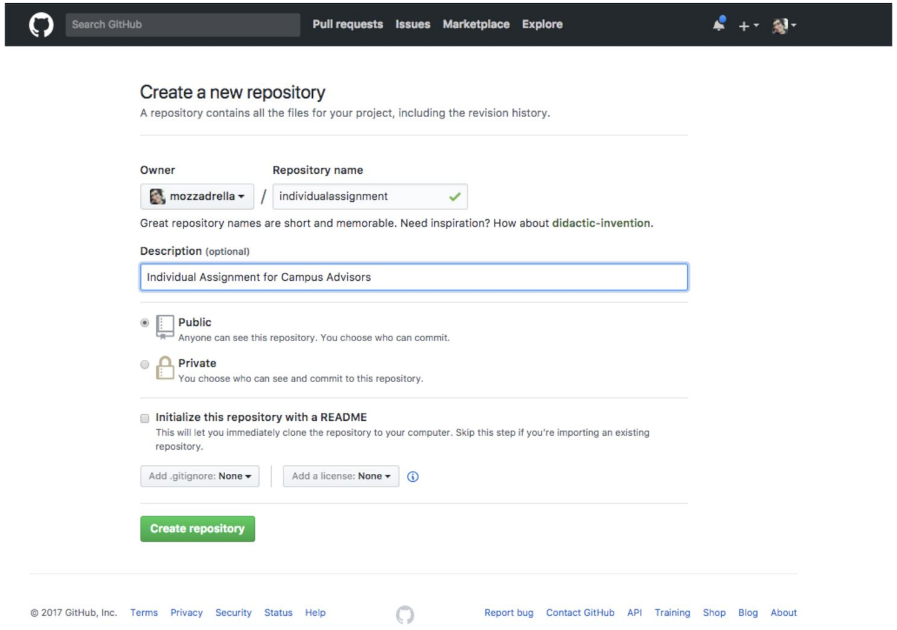
- Hosts your repositories
- Track student progress
- Social features to enable collabora

Let's create a repository in GitHub!

Let's set up a place to host your code

POLITEKNIK BRUNEI

- A repository on GitHub!
- https://github.com/new
- You have access to free private repositories, but let's choose public for now



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: mozzadrella / Repository name: individualassignment ✓

Great repository names are short and memorable. Need inspiration? How about didactic-invention.

Description (optional): Individual Assignment for Campus Advisors

[✓ Public] Anyone can see this repository. You choose who can commit.

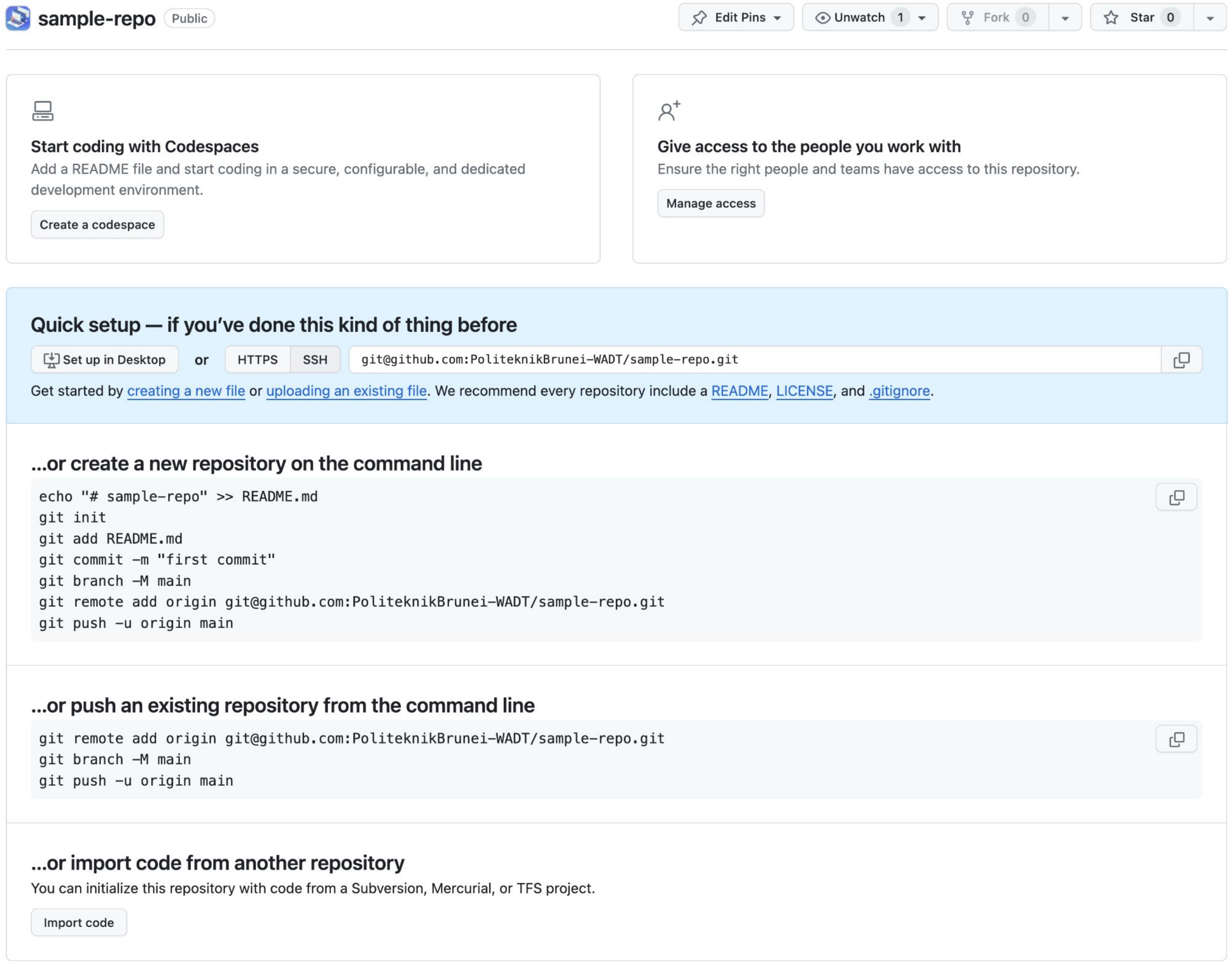
Private — You choose who can see and commit to this repository.

☐ Initialize this repository with a README — This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

Create repository

© 2017 GitHub, Inc. Terms Privacy Security Status Help Report bug Contact GitHub API Training Shop Blog About



Start coding with Codespaces

Add a README file and start coding in a secure, configurable, and dedicated development environment.

Give access to the people you work with

Ensure the right people and teams have access to this repository.

Quick setup — if you've done this kind of thing before

```
git@github.com:PolitekniKBrunei-WADT/sample-repo.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# sample-repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:PoliteknikBrunei-WADT/sample-repo.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:PoliteknikBrunei-WADT/sample-repo.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

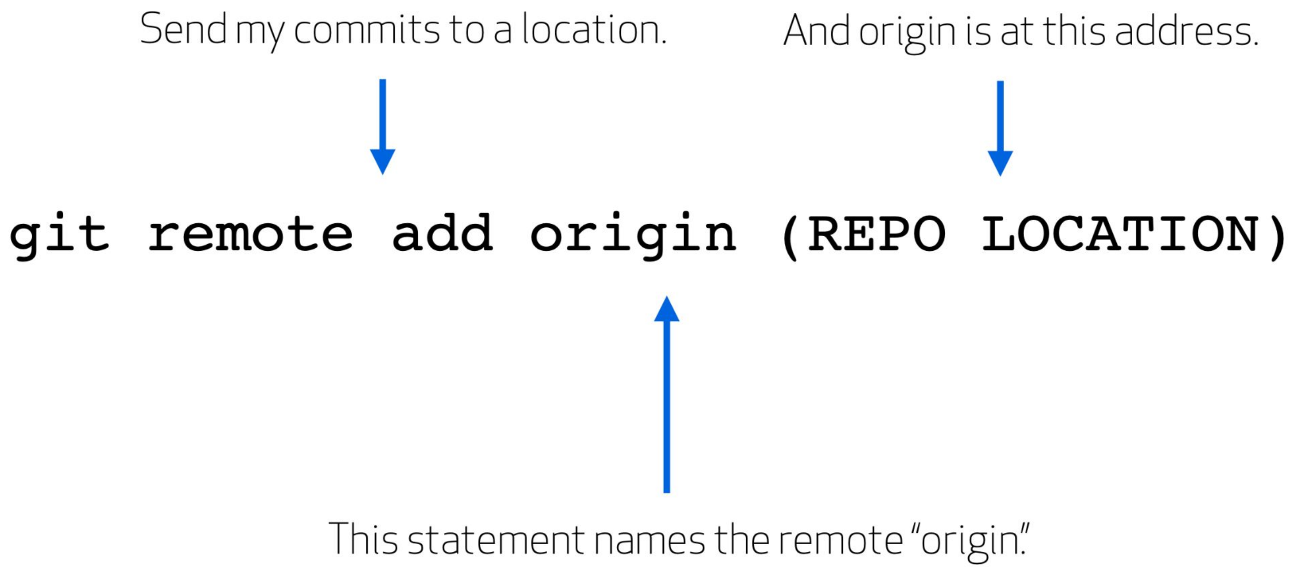
Remote

- Adding a remote allows the transfer of your commits to another machine.

```
git remote add origin (REPO LOCATION)
```

- The bookmarked location is referred to as a "remote"

Add origin



Send my commits to a location.

And origin is at this address.

```
git remote add origin (REPO LOCATION)
```

This statement names the remote "origin."

ڤوليتيكنيك بروني
POLITEKNIK BRUNEI

Back to the terminal!

Pushing to a remote

POLITEKNIK BRUNEI

- How do you get your commits up to the remote?

Link remote with local.

-u is short for --set-upstream

```
git push –u origin master
```

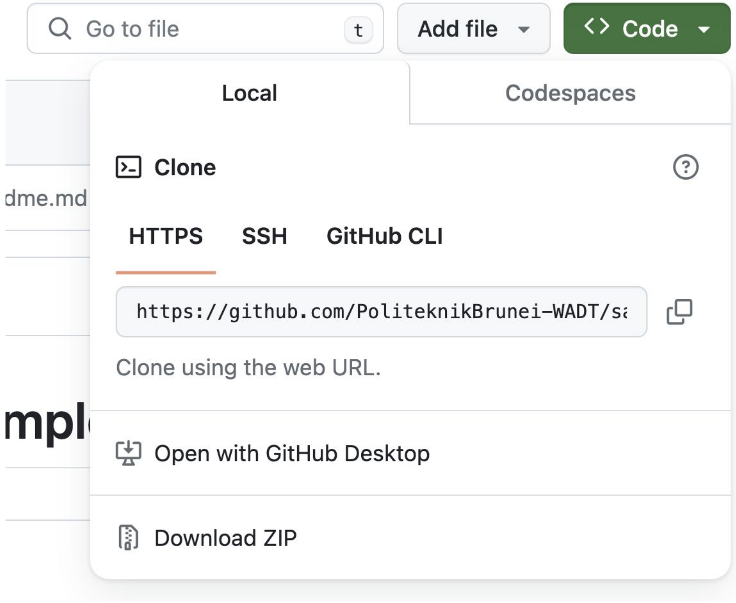
Useful because you can just write "git push" when you want to push future commits.

Types of remote addresses

- HTTP/HTTP Surls
- Git protocol over SSH and use the file path
- GitHub Desktop client (clone repository and openi n Desktop)

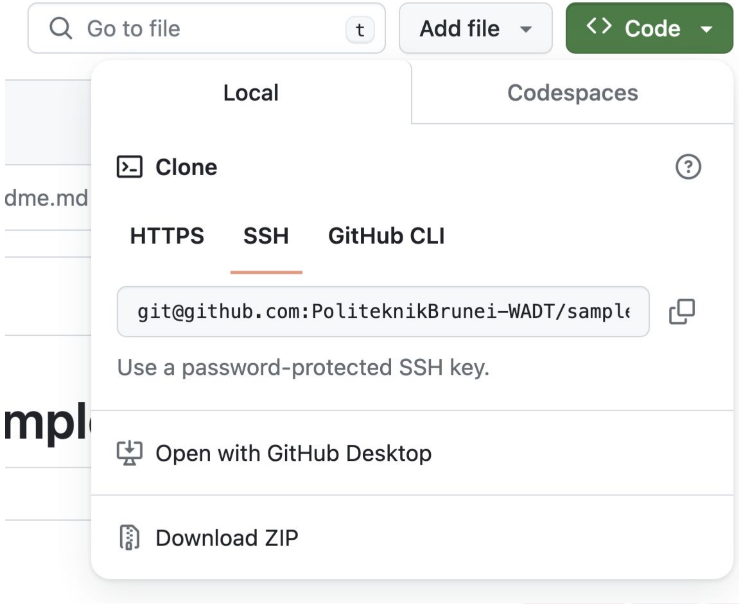
Using HTTPS

Using SSH



```
https://github.com/PoliteknikBrunei-WADT/sa
```

Clone using the web URL.



```
git@github.com:PoliteknikBrunei-WADT/sample
```

Use a password-protected SSH key.

https://github.com/PoliteknikBrunei-WADT/sample-repo.git

git@github.com:PoliteknikBrunei-WADT/sample-repo.git

Checking remote with -v

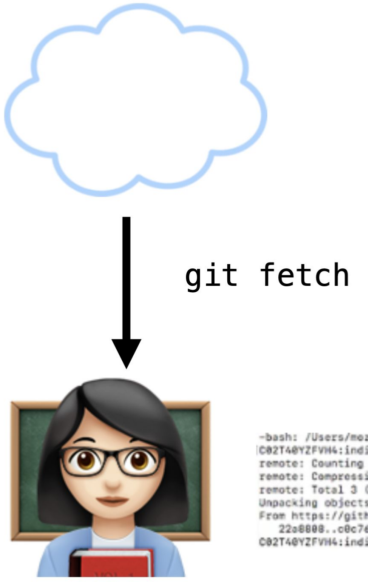
Using SSH

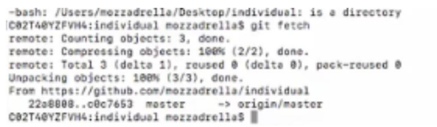
```
$ git remote -v
$ origin  git@github.com:PoliteknikBrunei-WADT/sample-repo.git (fetch)
$ origin  git@github.com:PoliteknikBrunei-WADT/sample-repo.git (push)
```

Using HTTPS

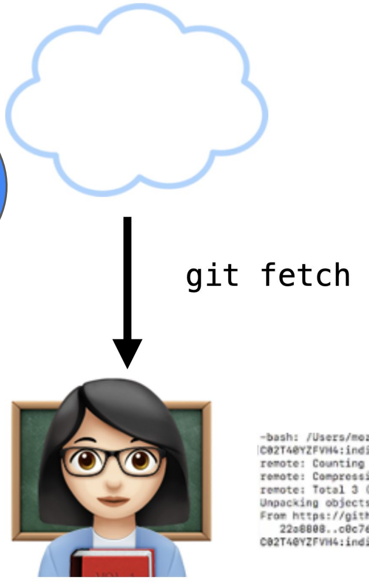
```
$ git remote -v
$ origin  https://github.com/PoliteknikBrunei-WADT/sample-repo.git (fetch)
$ origin  https://github.com/PoliteknikBrunei-WADT/sample-repo.git (push)
```

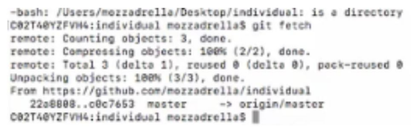
Fetch




```
-bash: /Users/mozzadrella/Desktop/individual: is a directory
C02T40YZFVH4:individual mozzadrella$ git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/mozzadrella/individual
   22a0808..c0c7653  master     -> origin/master
C02T40YZFVH4:individual mozzadrella$ 
```

Hmm, when I run `git log` I can't see these commits in my local repo

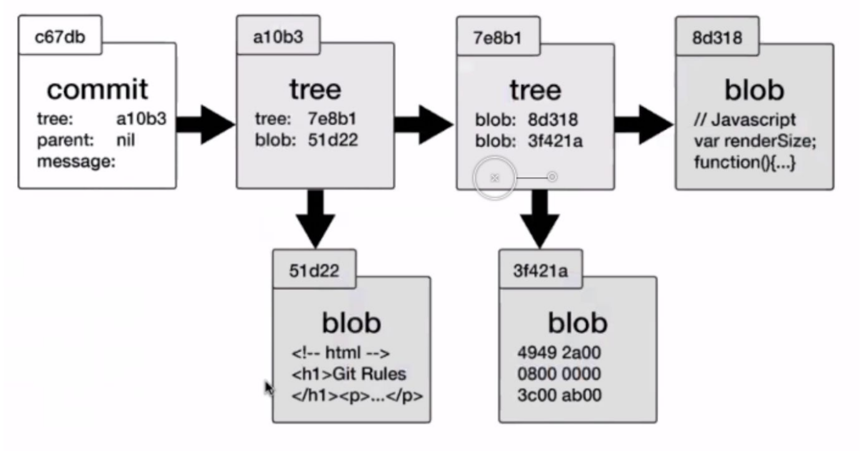




```
-bash: /Users/mozzadrella/Desktop/individual: is a directory
C02T40YZFVH4:individual mozzadrella$ git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/mozzadrella/individual
   22a0808..c0c7653  master     -> origin/master
C02T40YZFVH4:individual mozzadrella$ 
```

Counting Objects

- Git **only** transmits the necessary objects.
- **Push**: sends objects the remote doesn't have.
- **Fetch**: receives objects we don't have locally.



Activity!

Work with remotes

1. On the command line: create a repository from the command line called “individual-work”
2. On GitHub.com, create a repository.
3. On GitHub.com, upload your last slide’s (Git Training 01) activity to the “Individual” repository.
4. Use the command line to bring the commits back down to your local repository.

POLITEKNIK BRUNEI

Let’s talk about this command

Which branch do you want to push?

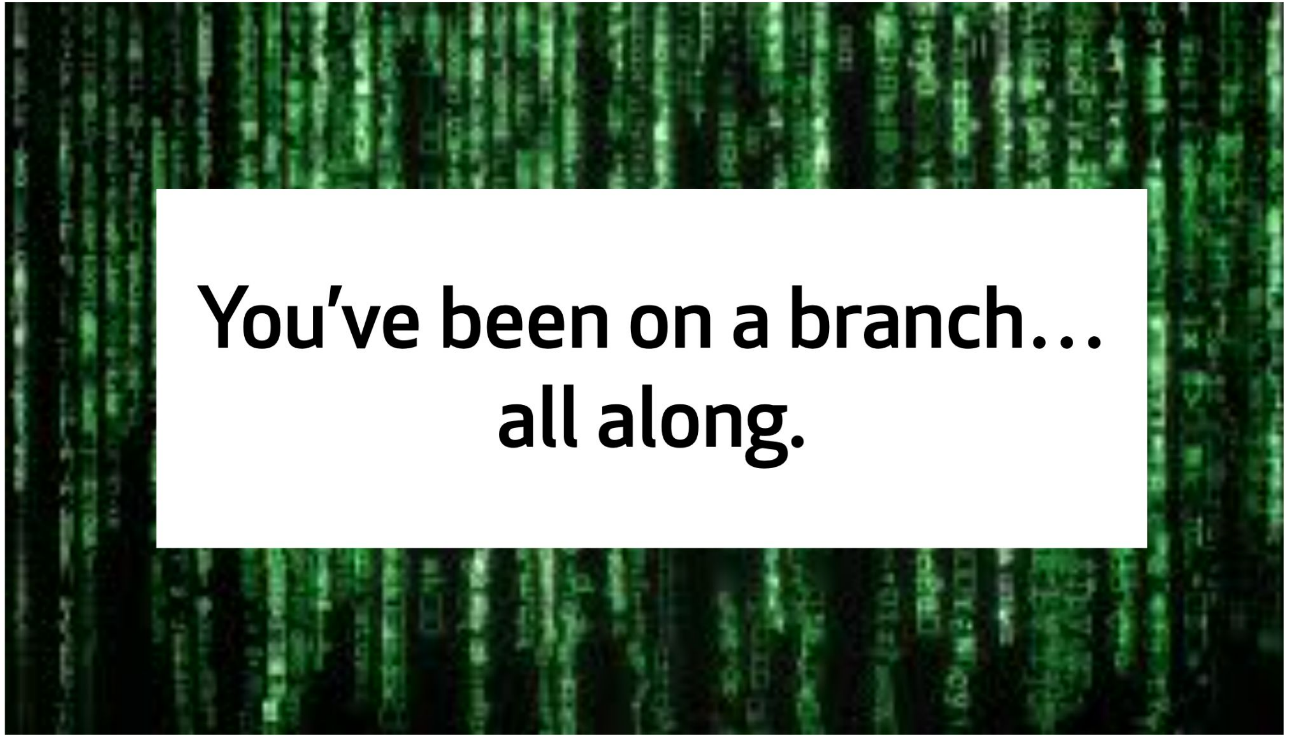
```
git push —u origin master
```

You want to push master.
To origin, the remote.

Let's talk about this command

But what is "master"?

```
git push –u origin master
```



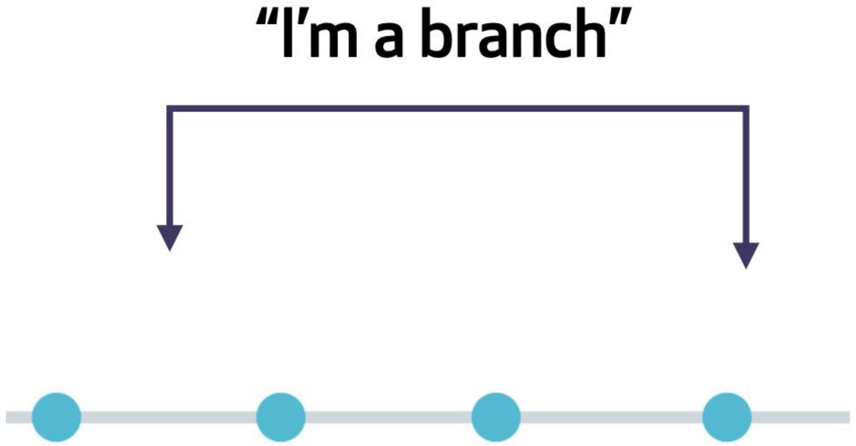
You've been on a branch… all along.

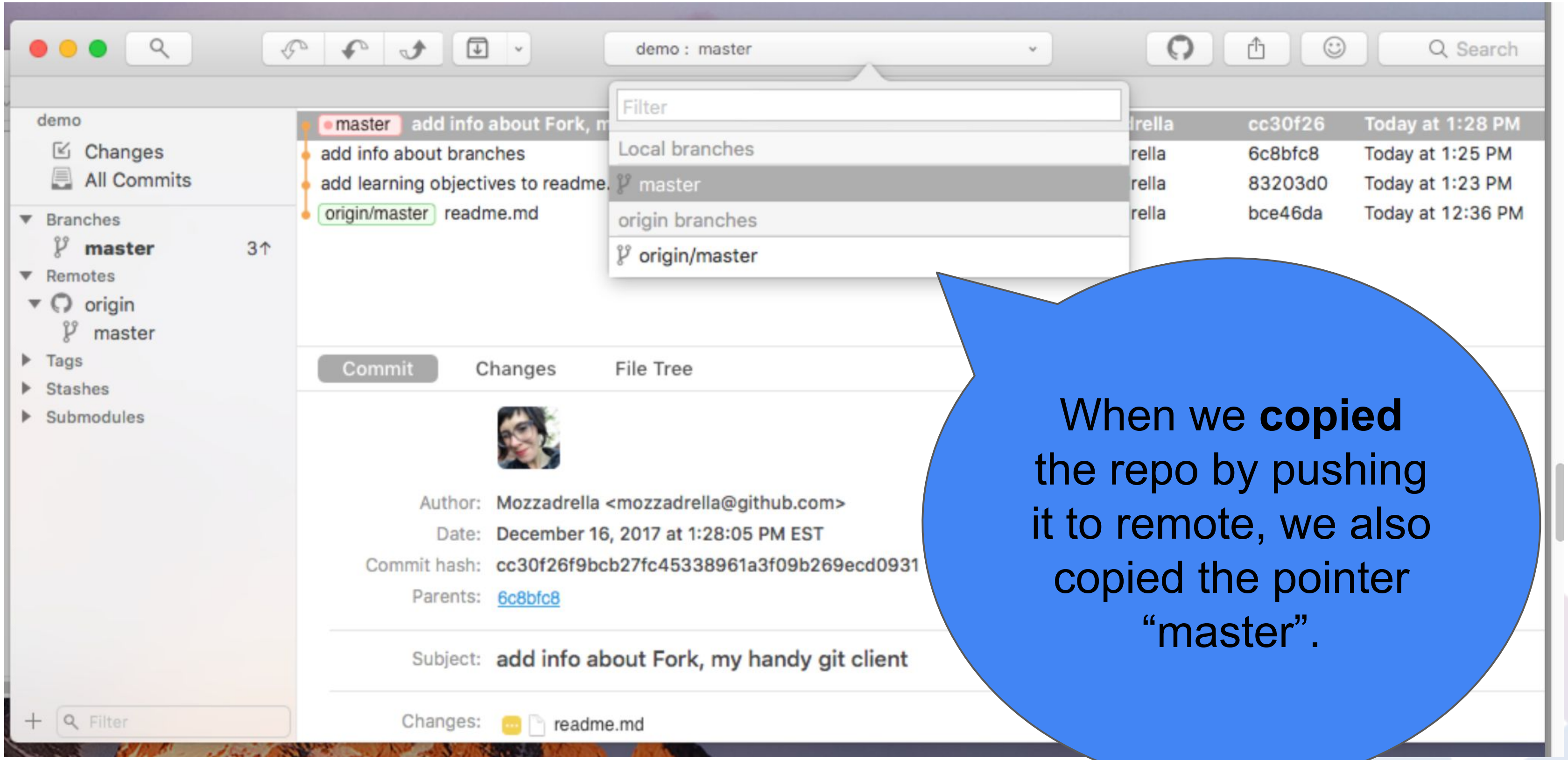
ڤوليتيكنيك بروني
POLITEKNIK BRUNEI

An aside to discuss about branches

Branches are bookmarks to commits

- “Master” is the default, it’s a naming convention. But few years back GitHub wants to change this to “Main”*
- Can think about branches as either a bookmark or a pointer for commits.
- As we add commits, the active branch updates to point to the newest commit (HEAD).

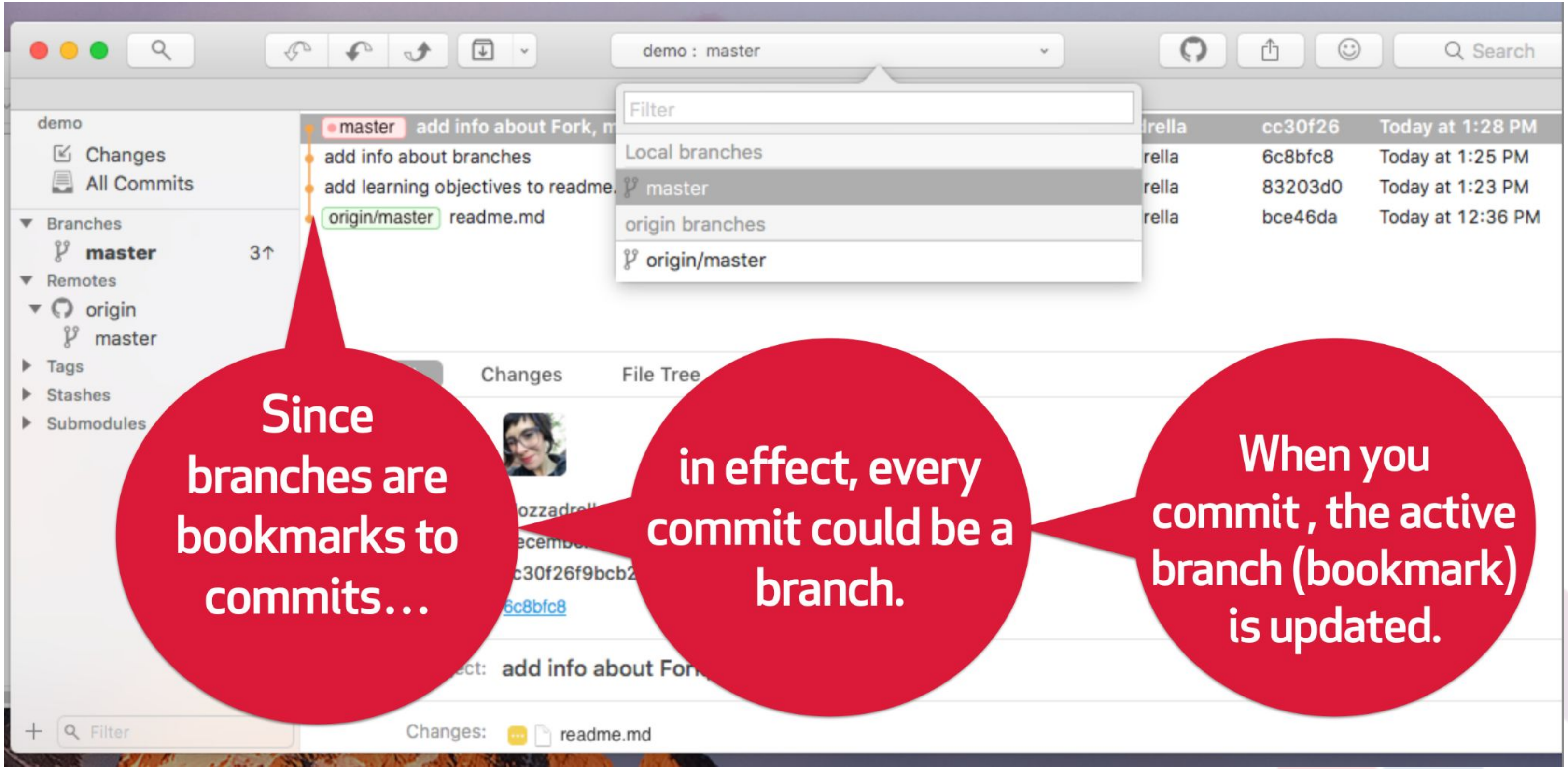




master add info about Fork, m	rella	cc30f26	Today at 1:28 PM
add info about branches	rella	6c8bfc8	Today at 1:25 PM
add learning objectives to readme.	rella	83203d0	Today at 1:23 PM
origin/master readme.md	rella	bce46da	Today at 12:36 PM

Author: Mozzadrella <mozzadrella@github.com>
Date: December 16, 2017 at 1:28:05 PM EST
Commit hash: cc30f26f9bcb27fc45338961a3f09b269ecd0931
Parents: 6c8bfc8
Subject: add info about Fork, my handy git client
Changes: readme.md

When we **copied** the repo by pushing it to remote, we also copied the pointer “master”.



master add info about Fork, m	rella	cc30f26	Today at 1:28 PM
add info about branches	rella	6c8bfc8	Today at 1:25 PM
add learning objectives to readme.	rella	83203d0	Today at 1:23 PM
origin/master readme.md	rella	bce46da	Today at 12:36 PM

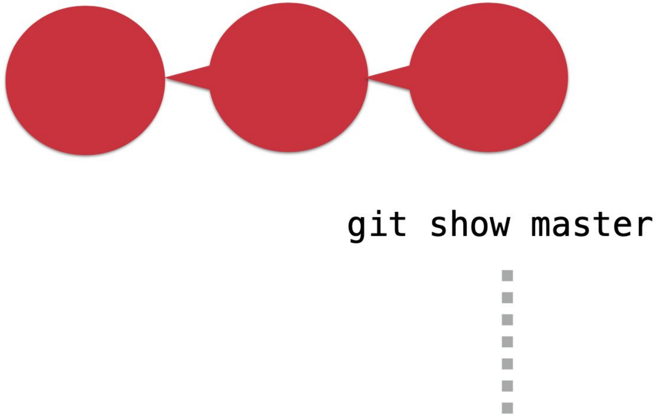
Since branches are bookmarks to commits…

in effect, every commit could be a branch.

When you commit , the active branch (bookmark) is updated.

Using branches in your terminal

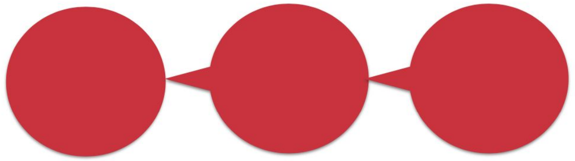
- Remember, branches are pointers to commits.
- If we say 'git show master' we'll see the commit master points to.



```
commit cc30f26f9bcb27fc45338961a3f09b269ecd0931
(HEAD -> master)
Author: Mozzadrella <mozzadrella@github.com>
Date:   Sat Dec 16 13:28:05 2017 -0500
```

Using branches in your terminal

- Remember, branches are pointers to commits.
- If we say 'git show master' we'll see the commit master points to.



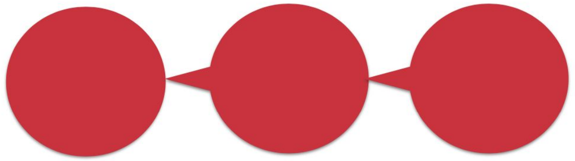
```
git show master
```

```
commit cc30f26f9bcb27fc45338961a3f09b269ecd0931
(HEAD -> master)
Author: Mozzadrella <mozzadrella@github.com>
Date:   Sat Dec 16 13:28:05 2017 -0500
```

Using branches in your terminal

POLITEKNIK BRUNEI

- Remember, branches are pointers to commits.
- If we say ‘git show master’ we’ll see the commit master points to.



```
git show master
```

```
commit cc30f26f9bcb27fc45338961a3f09b269ecd0931
(HEAD -> master)
Author: Mozzadrella <mozzadrella@github.com>
Date:   Sat Dec 16 13:28:05 2017 -0500
```

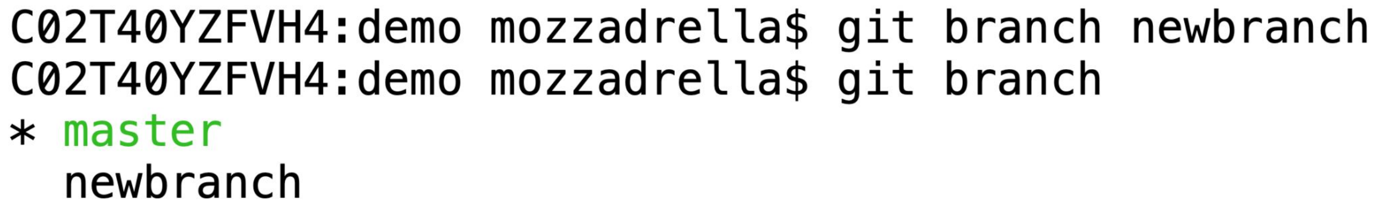
Finding the active branch

- 'Git branch' will show you the branches in your project…
- and the "*" indicates your currently active branch.
- If you made commits at that moment, the active branch would be updated to point to the new commit.

```
C02T40YZFVH4:demo
mozzadrella$ git branch
* master
```

Creating a new branch

- To summon a new branch, use ‘`git branch`’ and the new branch name. We’ll call ours ‘newbranch’

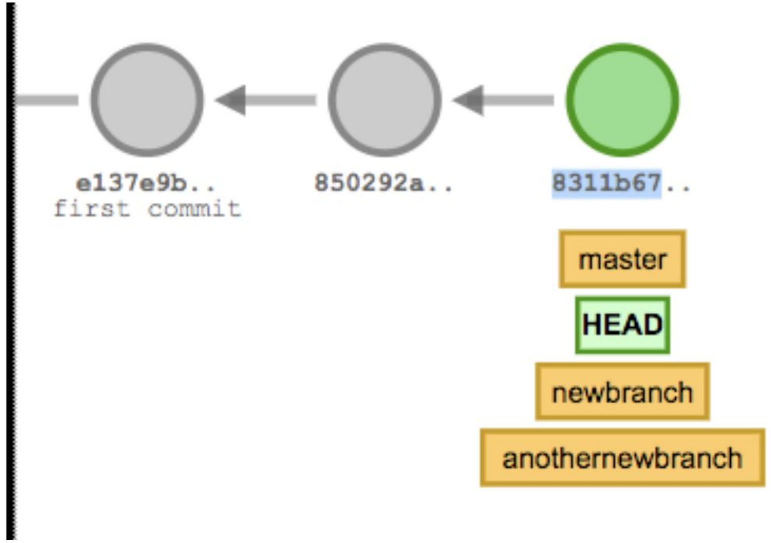


```
C02T40YZFVH4:demo mozzadrella$ git branch newbranch
C02T40YZFVH4:demo mozzadrella$ git branch
* master
  newbranch
```

Branches point back to the currently active commit

- If we created 2 new branches from

8311b67



Branches point back to the currently active commit

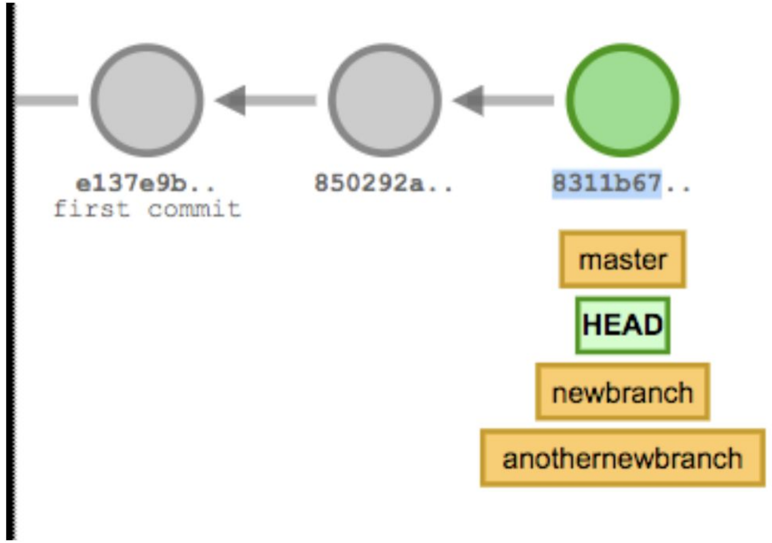
POLITEKNIK BRUNEI

- If we created 2 new branches from

  8311b67

- they would both point to

  8311b67



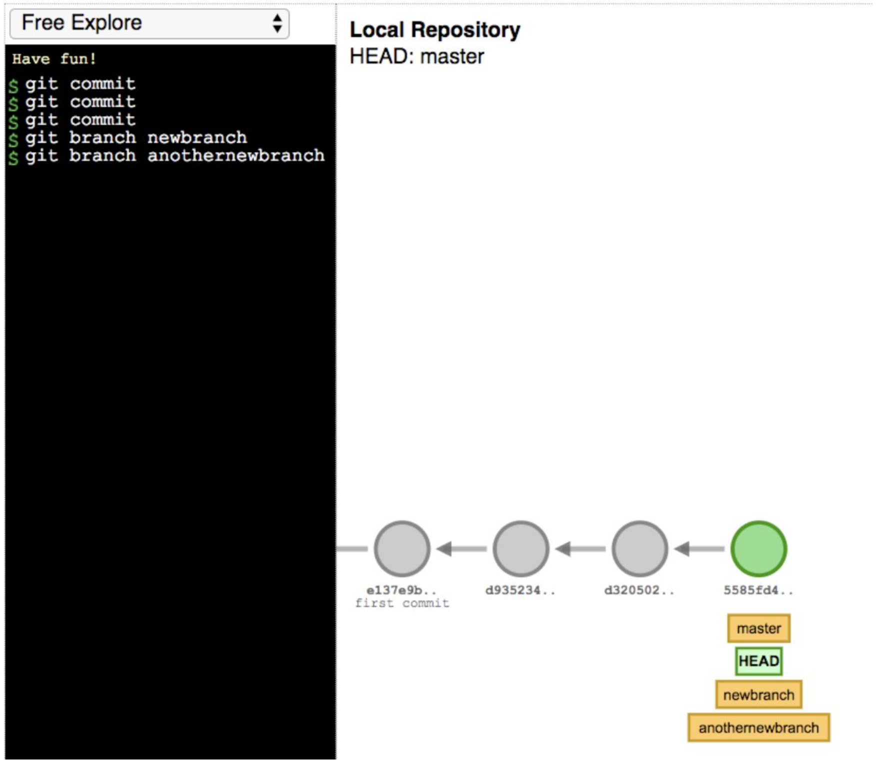
Branching from commits using references

POLITEKNIK BRUNEI

git branch <name> createsa branch at HEAD

git branch <name> <ref> creates a branch at <ref>

<ref> can be HEAD, a branch name, a commit, or a commit-ish (e.g. HEAD^or master~3)

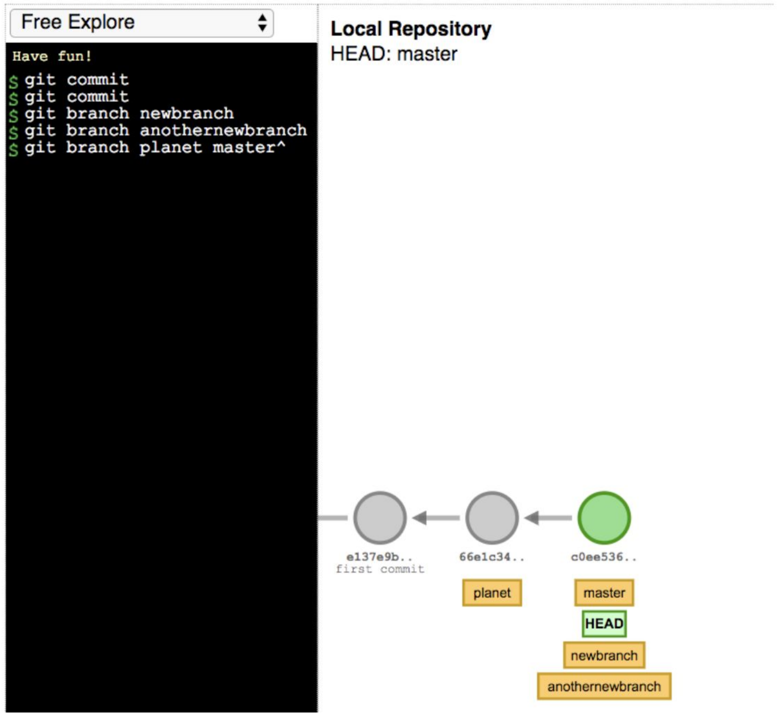


```
Have fun!
$ git commit
$ git commit
$ git commit
$ git branch newbranch
$ git branch anothernewbranch
```

Or branch from previous commits

POLITEKNIK BRUNEI

```
git branch planet master^
```



Free Explore

```
Have fun!
$ git commit
$ git commit
$ git branch newbranch
$ git branch anothernewbranch
$ git branch planet master^
```

Local Repository
HEAD: master

POLITEKNIK BRUNEI

A final word on branches...

```
tree .git/refs
```

```
└── refs
    ├── heads
    │   ├── master
    │   ├── newbranch
    │   └── planet
    ├── remotes
    │   └── origin
    │       └── master
    └── tags
```

Note: ‘tree’ command only available on Linux Operating System

Those files contain the commit ID...

```
C02T40YZFVH4:demo mozzadrella$ cat .git/refs/heads/planet
6c8bfc88bb440844f18a5e0a6ca885998b461bb7
```

ڤوليتيکنيک بروني
POLITEKNIK BRUNEI

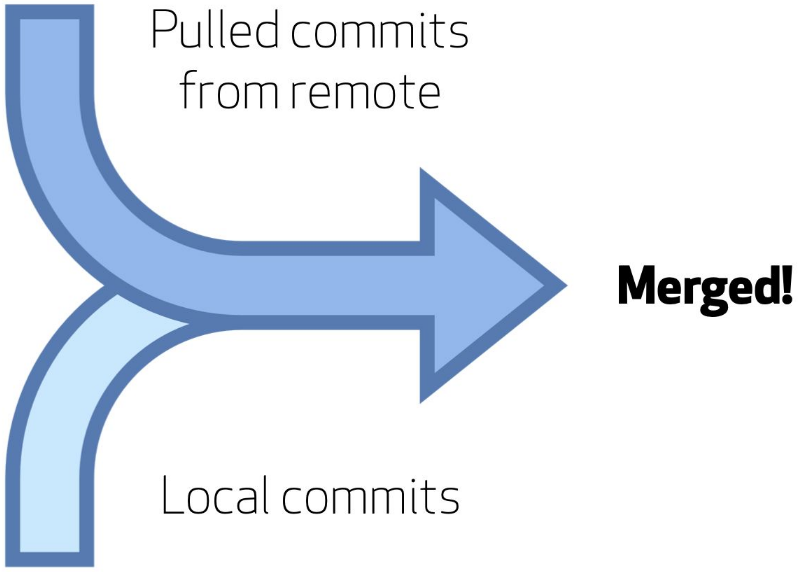
So the implementation for branches is a file with a hash in it.

Back to the world of network activity

Pull = fetch + merge

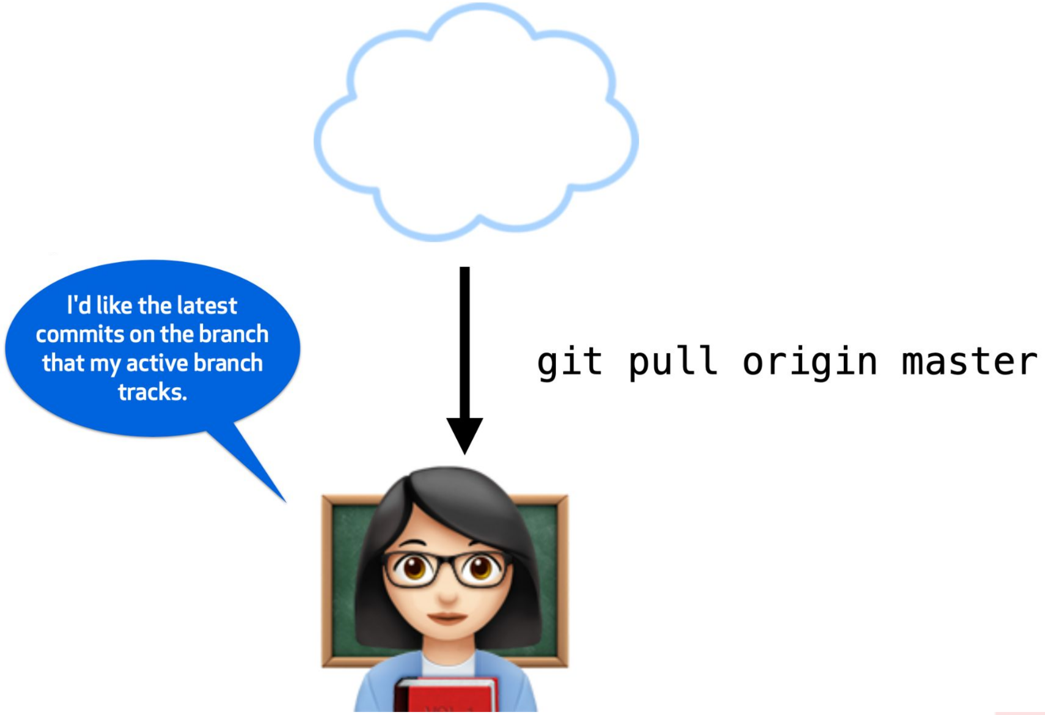
POLITEKNIK BRUNEI

- "`pull`" first fetches the commits and stores them locally.
- Merge takes the two divergent commits, puts them together in the staging area and makes a new commit with two parents.
- Merge updates the active branch to point to the new merge commit
- You'll see the new commits reflected in your local project when you run "`git log`"

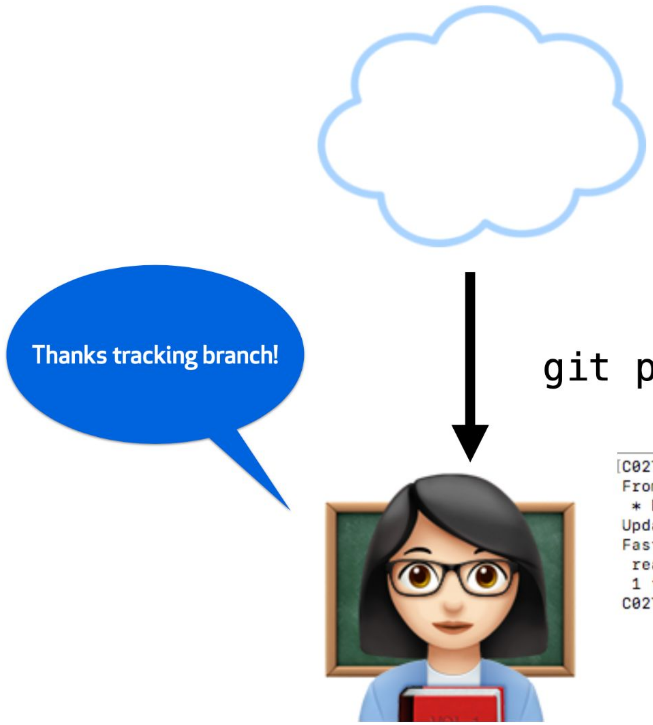


POLITEKNIK BRUNEI

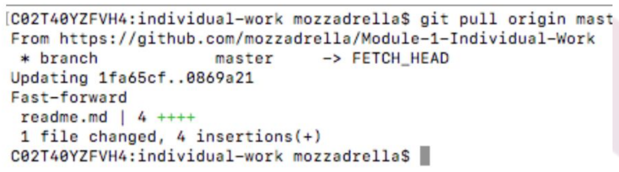
Watch what happens when we run “`pull`”


I'd like the latest commits on the branch that my active branch tracks.
git pull origin master

Watch what happens when we run “`pull`”



git pull origin master



```
[C02T40YZFVH4:individual-work mozzadrella$ git pull origin mast
From https://github.com/mozzadrella/Module-1-Individual-Work
 * branch            master     -> FETCH_HEAD
Updating 1fa65cf..0869a21
Fast-forward
 readme.md | 4 ++++
 1 file changed, 4 insertions(+)
C02T40YZFVH4:individual-work mozzadrella$ 
```

To sum up, here are the commands with network activity:

```
git push
git fetch
git pull
```

Note: You can’t do a `git pull` if your local repository contains uncommitted code or changes that are not ‘saved/committed. You’ll need to either commit your changes first or “`stash`” it

ڤوليتيكنيك بروني
POLITEKNIK BRUNEI

Thank you