Git

From the ground up

Git for Windows

- You can grab git (and other linux utils) from: https://git-scm.com/
- In linux its even easier:

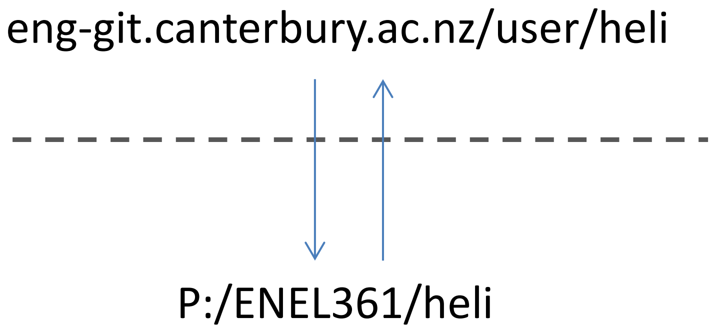
```
$ sudo apt-get install git
```

eng-git.canterbury.ac.nz

- Repository hosting service
- The ‘real’ version of your code
- Secure access via. HTTPS or SSH

git clone <url>

- Download a local copy of the repository
- Points back to the remote repository
- These are two separate repositories that you need to synchronize



Types of Operations

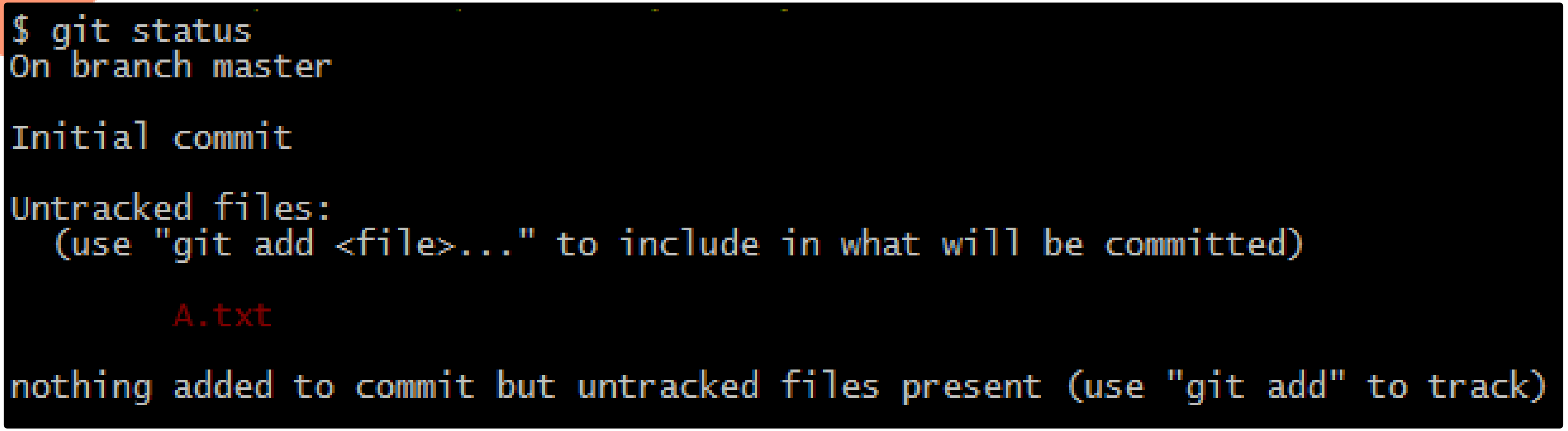
- Modify the local repository
  - git commit
  - git branch
- Synchronize the local and remote repositories
  - git push
  - git pull

Useful Information

- You can also ask git questions
  - git status   Tells you about general repo info
  - git log [-n #]   Lists previous commit tree
  - git log --graph [-n #]   Like above but pretty!
  - git log --graph --all

Lets Add a File!

```
$ touch A.txt
$ echo ‘Hello World!’ > A.txt
$ git status
```



```
$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        A.txt

nothing added to commit but untracked files present (use "git add" to track)
```

git add

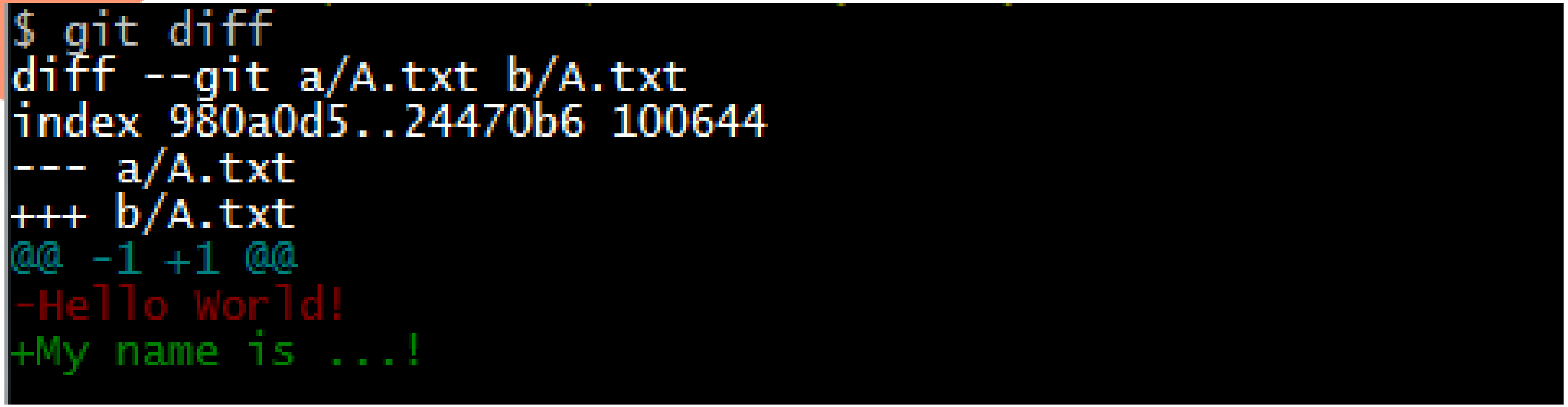
- git add .
- git add -A
- git add A.txt
- These are three ways to start tracking changes to the *A.txt* file. It also adds the file to the next commit.

git commit

- git commit -m ‘Added A.txt’
- This commits your changes (adding a file in this case) with the message “…”.
- A commit records the changes you made to get from the previous commit, to the current state of your working directory (changeset).

Making a Change!

```
$ echo ‘My name is …!’ > A.txt
$ git diff
```



```
$ git diff
diff --git a/A.txt b/A.txt
index 980a0d5..24470b6 100644
--- a/A.txt
+++ b/A.txt
@@ -1 +1 @@
-Hello World!
+My name is ...!
```

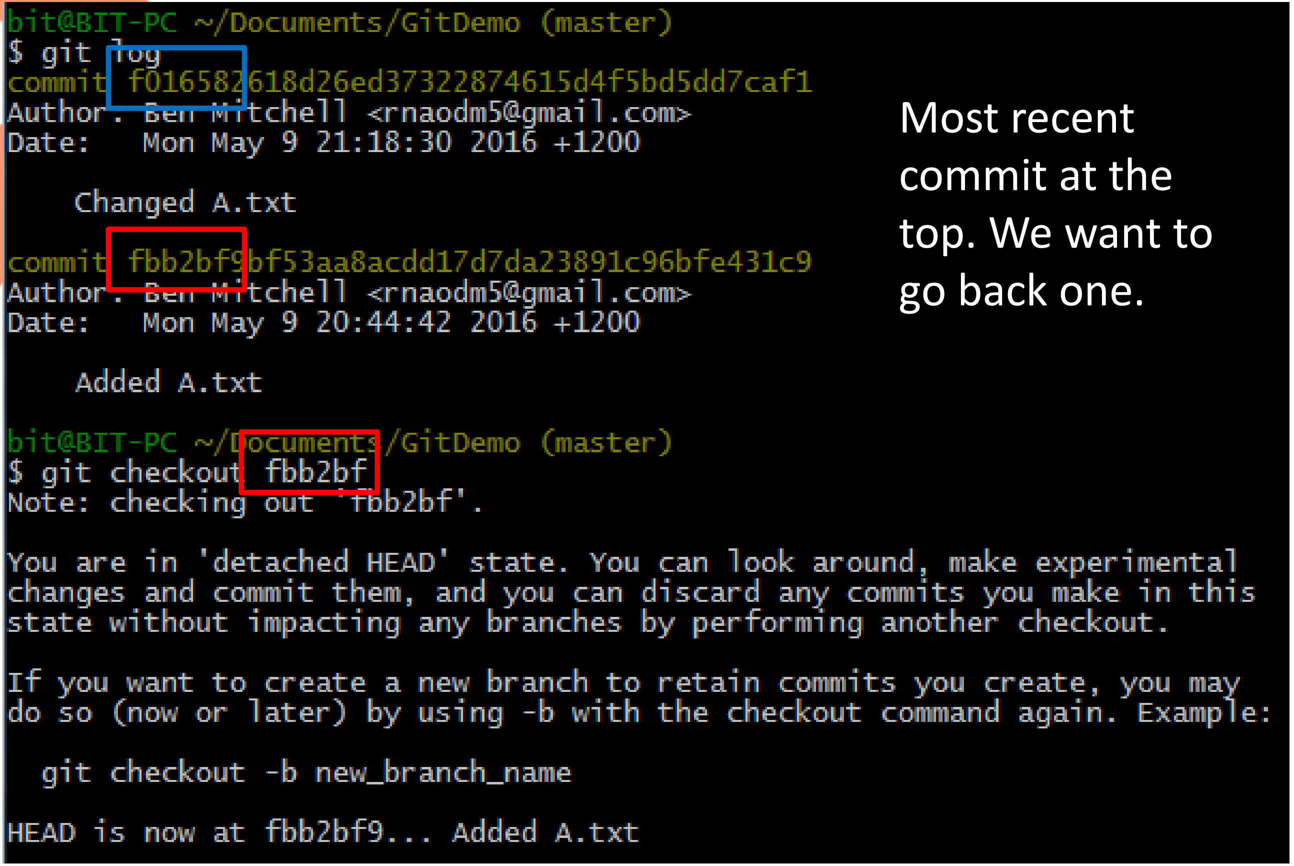
Checking Git State

- git diff
  - shows what has changed for each file (insertions and deletions).
- git status
  - shows which files have been changes, which are new/deleted, and which are staged.

Staging Area

- git status   will tell you there are changes not staged for commit.
- git add   adds these files to the *staging area*. This is a half-way point for a commit.
- git commit   then turns these staged changes into a changeset and commits it to history

Going back in Time



```
bit@BIT-PC ~/Documents/GitDemo (master)
$ git log
commit f016582618d26ed37322874615d4f5bd5dd7caf1
Author: Ben Mitchell <rnaodm5@gmail.com>
Date:   Mon May 9 21:18:30 2016 +1200

    Changed A.txt

commit fbb2bf9bf53aa8acdd17d7da23891c96bfe431c9
Author: Ben Mitchell <rnaodm5@gmail.com>
Date:   Mon May 9 20:44:42 2016 +1200

    Added A.txt

bit@BIT-PC ~/Documents/GitDemo (master)
$ git checkout fbb2bf
Note: checking out 'fbb2bf'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b new_branch_name

HEAD is now at fbb2bf9... Added A.txt
```

Most recent commit at the top. We want to go back one.

Pushing and Pulling

- git push   Pushes new commits to the remote repository. Always pull before writing code.
- git pull   Pulls new commits from the remote

- If git tells you something went wrong during a push or a pull. Ask! DO NOT USE -f (force).

Branching

- So far we have just used one branch, ‘master’
- It is very hard to coordinate multiple people on one branch without causing a fuss.
- Branching allows multiple versions of the code to exist simultaneously. These can be edited separately then the changes ‘merged’ later.

Questions?