git

Intro

- Developers collaborate
  - Source control systems help teams collaborate

- Git has had a major impact on how this is achieved.

- Agenda:
  - Introduction to Git basics.
 - Collaborating with git
  - Branching
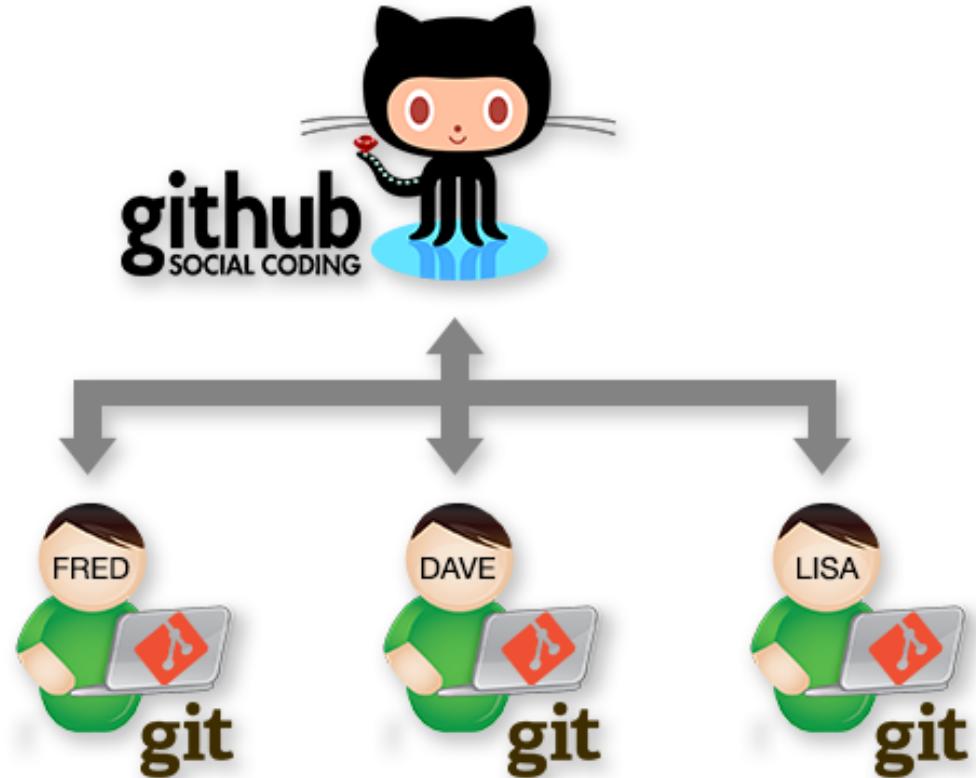
Some History

- Created by Linus Torvalds for work on the Linux kernel ~2005

- Used by:

  - Nearly everybody at this stage…

What is Git

- Distributed Version Control

- Directory Content Management

- Tree Based History

- Everybody has complete history

- **I get confused sometimes…**
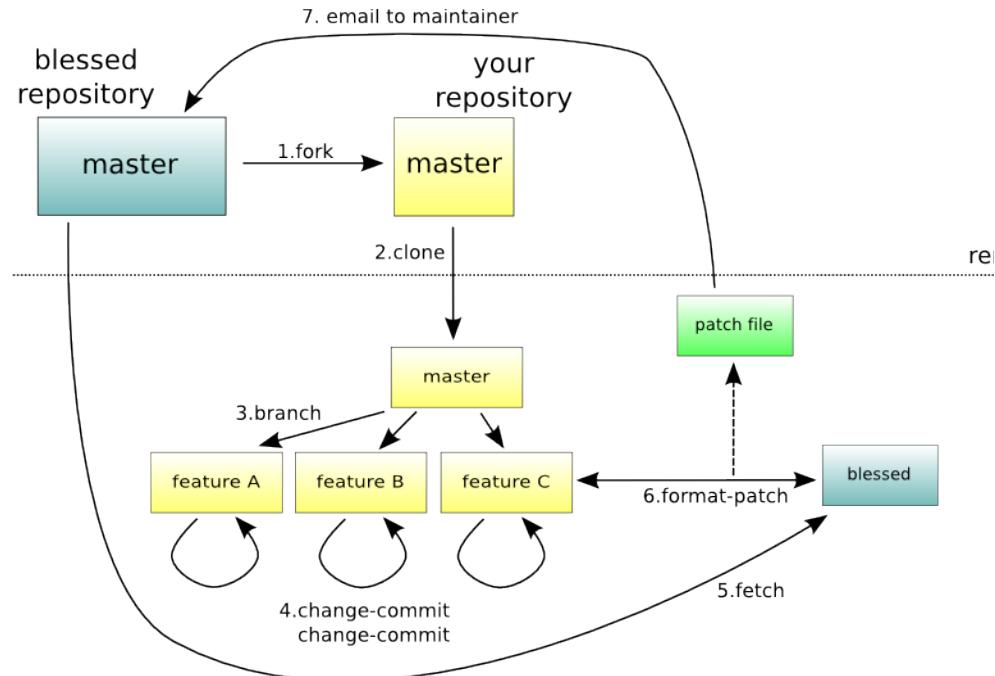
Centralised vs Distributed
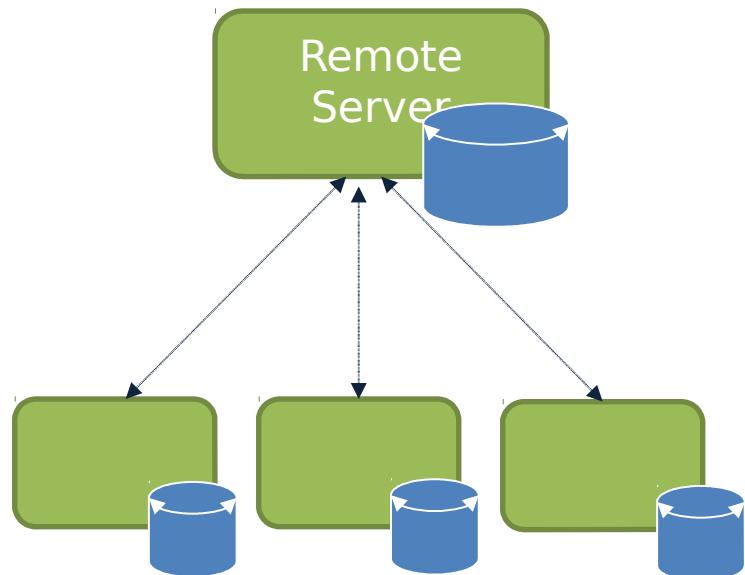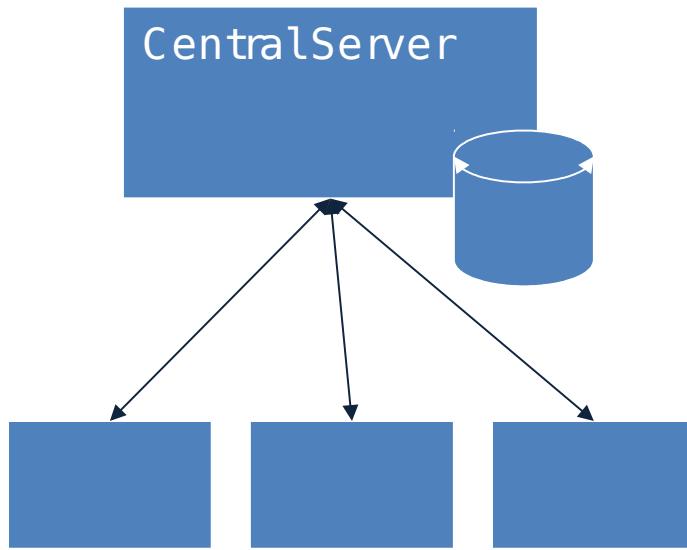
Initialising a repo…

$ mkdir myproject
$ cd myproject
$ git init
Initialized empty Git repository in /home/ec2-user/myproject/.git/
$ git config --global user.name "fxwalsh"
$ git config --global user.email fxwalsh@wit.com
$ vi README.txt
$ git add .
$ git commit -m 'initial commit'
[master (root-commit) 7d738f4] initial commit
1 file changed, 1 insertion(+)
 create mode 100644 README.txt

master

A
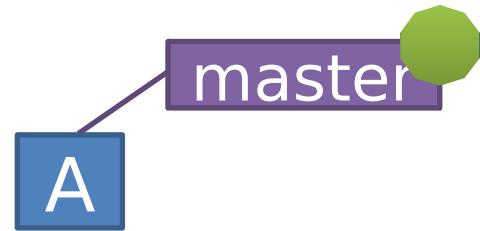
Multiple Commits
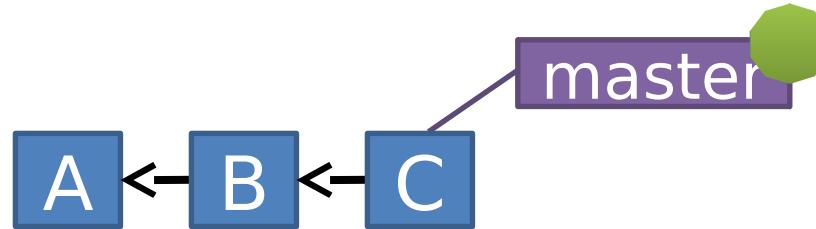
git add --all
git commit –m "updated text file"
git add --all
git commit –m "updated text file
again"

Github

Adding a Remote Repo to Existing Project

```
git remote add origin https://github.com/fxwalsh/BSc4Repo.git
git remote -v
origin  https://github.com/fxwalsh/BSc4Repo.git (fetch)
origin  https://github.com/fxwalsh/BSc4Repo.git (push)
```

Setting up Remote via Cloning

```
 git clone
… …
 git remote -v
origin https://github.com /fxwalsh/BSc4Repo.git (fetch)
origin https://github.com /fxwalsh/BSc4Repo.git (push)
```

git push
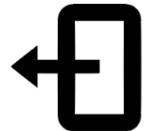
- Pushes your changes to remote
- Changes will be rejected if newer changes exist on remote
  - If this is the case, do a **git pull**
  - merge locally, then push the results.

Pushing to remote

- By default, remote repo is labelled origin.
- Say you want to stage, commit, and push all your latest changes to origin…

```
git add --all
git commit –m "some important update to important stuff"
git push origin master
```

Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. This cheat sheet summarizes commonly used Git command line instructions for quick reference.

INSTALL GIT

GitHub provides desktop clients that include a graphical user interface for the most common repository actions and an automatically updating command line edition of Git for advanced scenarios.

GitHub for Windows
https://windows.github.com

GitHub for Mac
https://mac.github.com

Git distributions for Linux and POSIX systems are available on the official Git SCM web site.

Git for All Platforms
http://git-scm.com

CONFIGURE TOOLING

Configure user information for all local repositories

```
$ git config --global user.name "[name]"
```
Sets the name you want attached to your commit transactions

```
$ git config --global user.email "[email address]"
```
Sets the email you want attached to your commit transactions

```
$ git config --global color.ui auto
```
Enables helpful colorization of command line output

CREATE REPOSITORIES

Start a new repository or obtain one from an existing URL

MAKE CHANGES

Review edits and craft a commit transaction

```
$ git status
```
Lists all new or modified files to be committed

```
$ git diff
```
Shows file differences not yet staged

```
$ git add [file]
```
Snapshots the file in preparation for versioning

```
$ git diff --staged
```
Shows file differences between staging and the last file version

```
$ git reset [file]
```
Unstages the file, but preserve its contents

```
$ git commit -m "[descriptive message]"
```
Records file snapshots permanently in version history

GROUP CHANGES

Name a series of commits and combine completed efforts

```
$ git branch
```
Lists all local branches in the current repository

```
$ git branch [branch-name]
```
Creates a new branch

```
$ git checkout [branch-name]
```
Switches to the specified branch and updates the working directory

Final Word

- Cheat sheet is always handy…