Git Basics and Troubleshooting

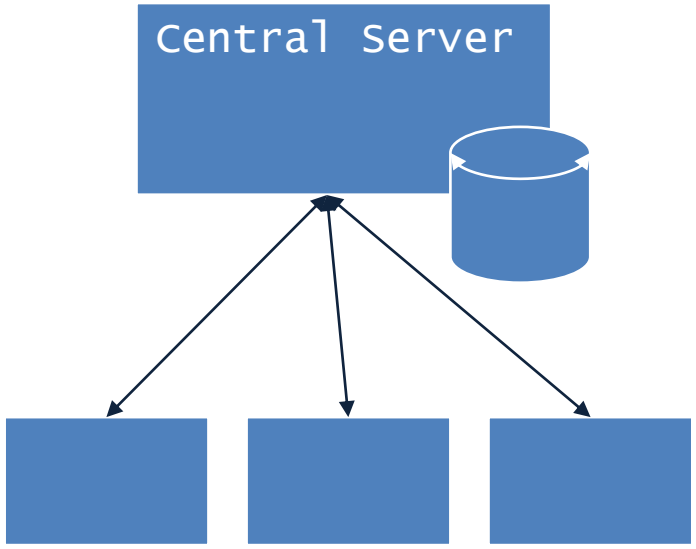
Frank Walsh

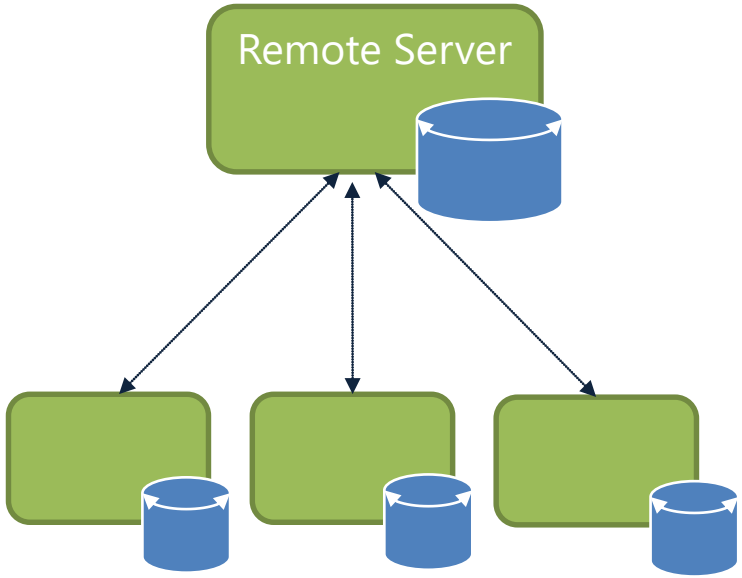
This Photo by Unknown Author is licensed under CC BY-SA-NC

What's Git

- Distributed Version control
- Directory content management
- Use it to keep track of your stuff
  - Really useful if you break your code and can't get back to when it was working!
- Not just for code…
  - Documents, diagrams, configs
- Sometimes I get confused!

Git is “distributed”





Initialising a Repo locally

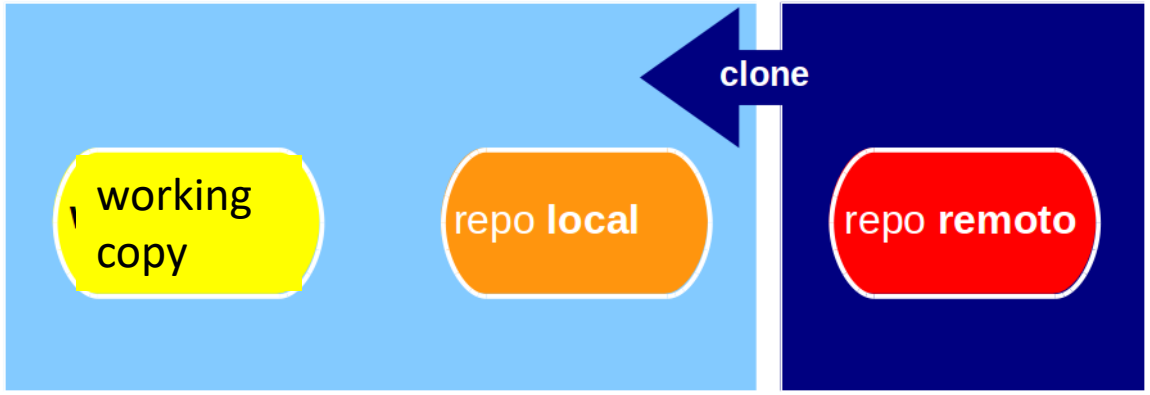
```
$ mkdir myproject
$ cd myproject
$ git init
Initialized empty Git repository in C:/Users/Frank/myproject/.git/
$ git config --global user.name “fxwalsh“
$ git config --global user.email fxwalsh@wit.com
$ touch README.md
$ git add .
$ git commit -m 'initial commit'
[master (root-commit) 7d738f4] initial commit
1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

This will create a .git directory (your local repo) and the current directory will be your “working directory”

Alternative – clone a remote

- `git clone url someLocalDirectory`

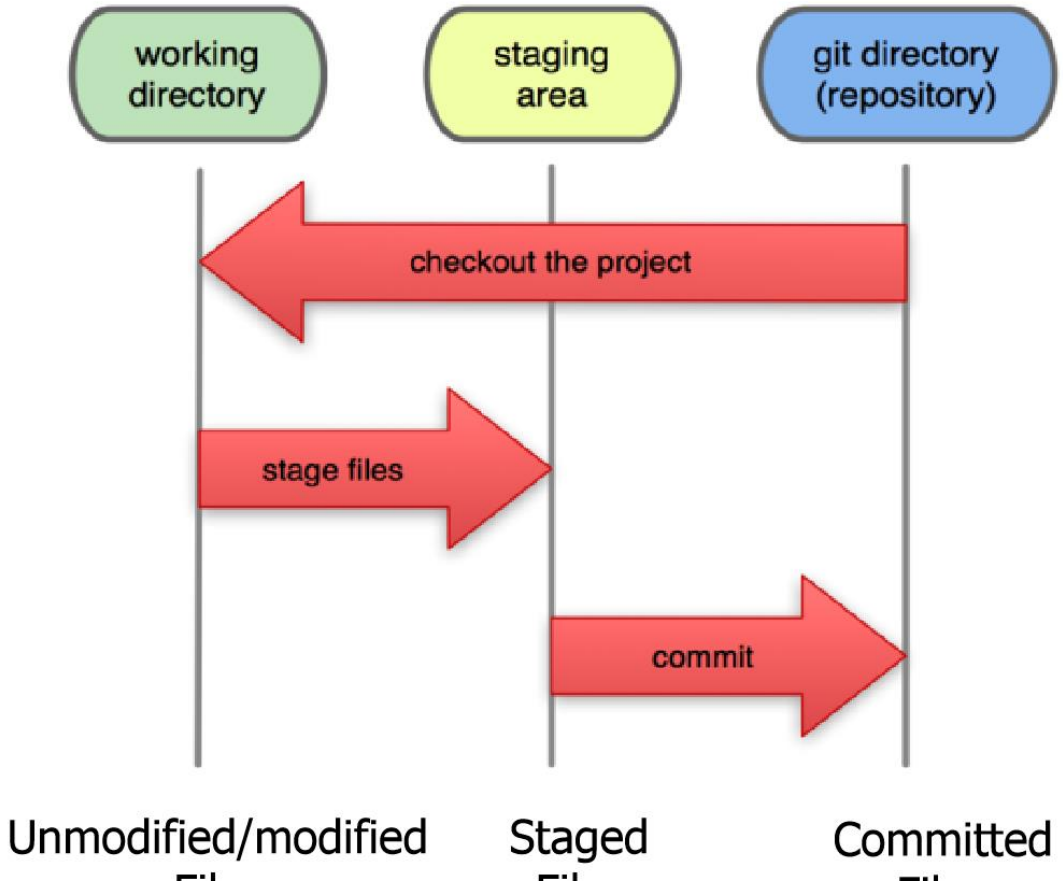
This will create the given local directory with a working copy of the files from the repo, and a .git directory (your local repo)



This Photo by Unknown Author is licensed under CC BY-SA-NC

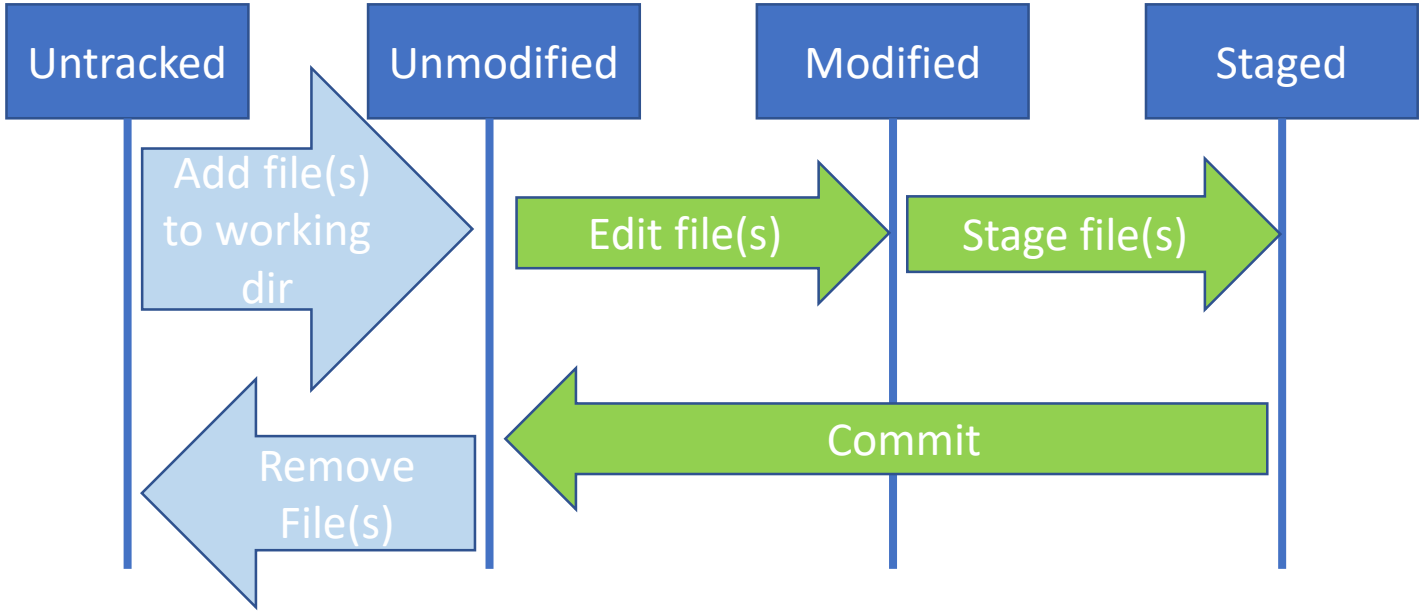
Local Git States

- Files can be
  - In your local repository
    - once committed
  - Checked out and/or modified
    - In your working directory(working copy)
  - Staged (using git add)
    - Ready to be committed
- A Commit saves a “snapshot” of (and only of) staged changes.
  - REMEMBER: JUST BECAUSE YOU SAVED A FILE DOESN’T MEAN GIT WILL PAY IT ANY ATTENTION.

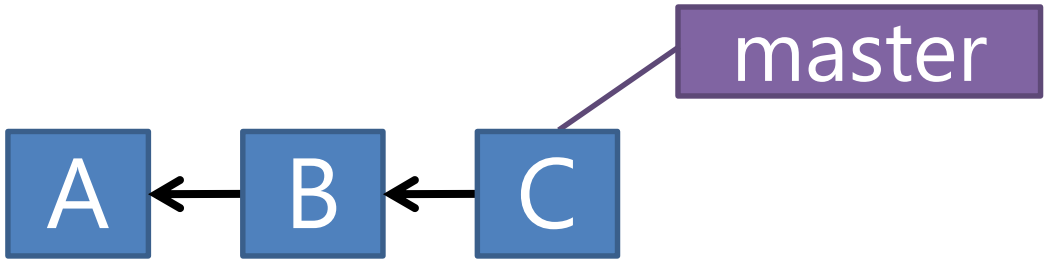


Typical Git Workflow

- Edit your files in the working directory
- Stage files once your happy (e.g. the script works!)
- Commit at significant junctures (e.g. functionality finished, embarking on significant code changes/refactoring)



Multiple Commits



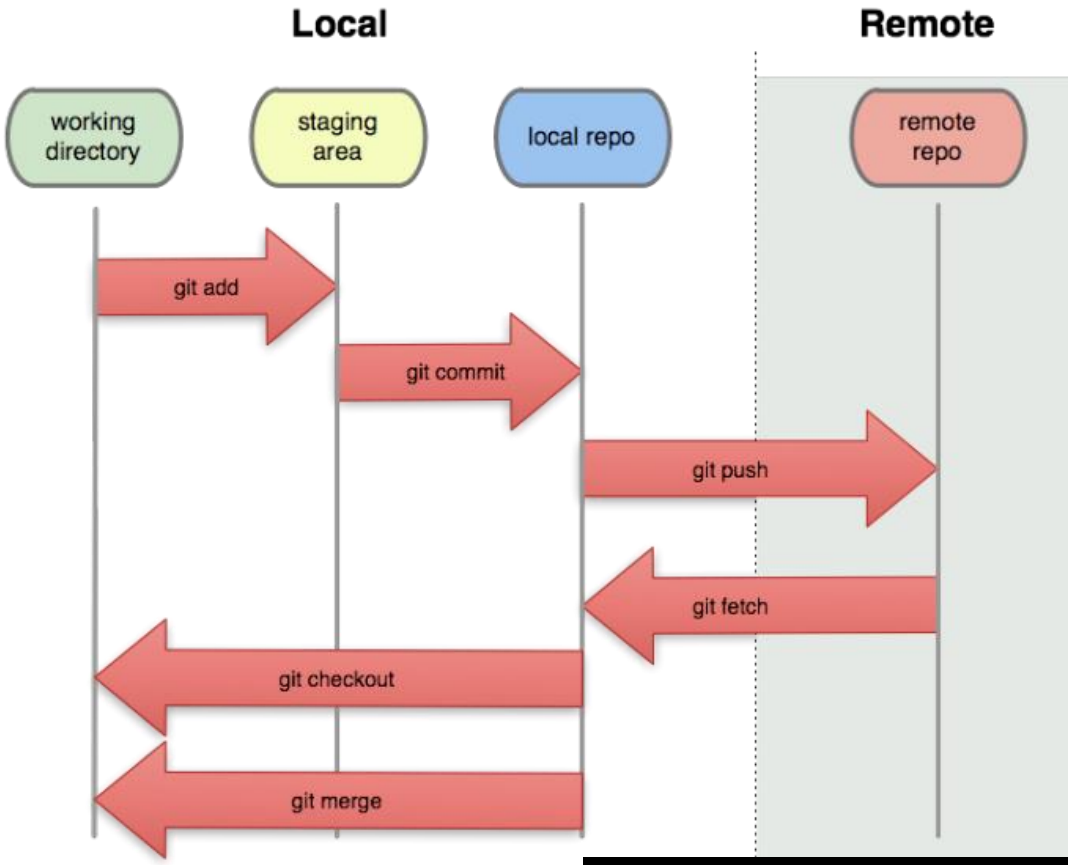
git add README.md
git commit –m “updated read me”
[master (root-commit) 7d738f4] ….
git add --all
git commit –m “updated read me again”
[master (root-commit) 627d8h4] ….

Popular Git Commands

command	description
`git clone` ***`url [dir]`***	copy a Git repository so you can add to it
`git add` ***`file`***	adds file contents to the staging area
`git commit`	records a snapshot of the staging area
`git status`	view the status of your files in the working directory and staging area
`git diff`	shows diff of what is staged and what is modified but unstaged
`git help` ***`[command]`***	get help info about a particular command
`git pull`	fetch from a remote repo and try to merge into the current branch
`git push`	push your new branches and data to a remote repository
others: `init, reset, branch, checkout, merge, log, tag`	

Remote Repositories

- Online storage providers of Git repositories include Github and Bitbucket.
  - You can create a remote repo there and push code to it.
  - Many open source projects use it, such as the Linux kernel.
  - You can get free space for open source and private projects

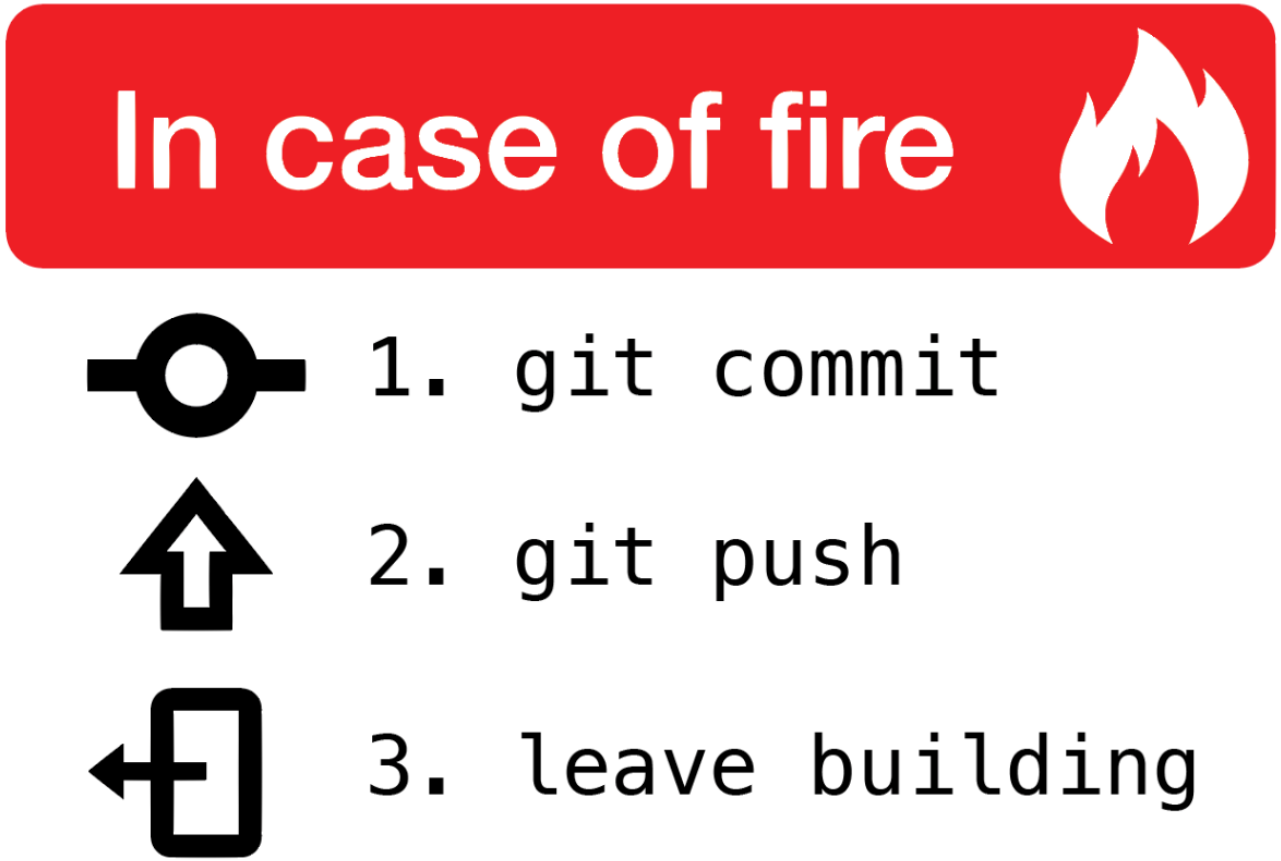


This Photo by Unknown Author is licensed under CC BY-SA

Adding a Remote Repo to Existing Project

```
 git remote add origin https://github.com/fxwalsh/msc-2019.git
 git remote -v
origin  https://github.com/fxwalsh/msc-2019.git(fetch)
origin  https://github.com/fxwalsh/msc-2019.git(push)
```

git push



- Pushes your changes to remote
- Changes will be rejected if newer changes exist on remote
- Good to pull then push
  - merge locally, then push the results.

Pushing to remote

By default, remote repo is labelled **origin.**
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
```

Potential situation 1

I can't push to origin, it's saying "! [rejected] master -> master (fetch first)"

- Probably changes on the remote repo that you don't have locally.
- Solution: Do a **pull** then **push.**
- Should work as long and there's no conflict

```
$ git push
   To https://github.com/fxwalsh/project2.git
   ! [rejected]        master -> master (fetch
   first) error: failed to push some refs
….
$git pull
…..
   Merge made by the 'recursive' strategy.
   README.md | 2 +-
   1 file changed,1 insertion(+), 1 deletion(-)
$git push
Counting objects: 5, done.
…
```

Potential situation 2

I want to replace local repo with remote, don't care about losing changes

- Solution: fetch from the default remote, origin
- Reset your current branch (master) to origin's master

```
$ git fetch
…
$ git reset --hard origin/master
```

Potential situation 3

I want to replace local repo with remote, don't care about losing changes on remote

- Solution: force the push

```
$ git push -f origin master
```

Potential situation 4

Trying to pull but getting "CONFLICT (content): Merge conflict in ...."

- Two different versions of the same file(s)
- Decide which one you want to keep manually.
- Perhaps force push or hard reset.

```
$ git push -f origin master
```

Potential Situation 5

I committed stuff I really didn't want to. I want to revert to the previous commit.

Do **git reset HEAD~1**

Moves Head (pointer to latest commit) back one

Leaves files alone (but not staged)

Do **git reset –hard HEAD~1**

Gets rid of changes completely

Do **git reset –soft HEAD~1**

Leaves changes staged