**Level 2: Staging and Remotes**

**Git Diff**

git diff

Shows unstagged differences from something that was last committed to something recently worked on.

Red is line removed

Green is line added

git diff --staged

View staged differences

**Unstaged files**

git reset HEAD filename

Head refers to the most recent commit on the timeline for the branch.

This command removes a file from the staging area so it won’t be committed and needs to be re added.

**Discard Stages**

git checkout -- filename

If you aren’t happy with the file’s current changes this will revert the file to its previously committed state.

**Skip Staging and Commit**

git commit -a -m “commit message”

This adds and commits all tracked files.

If a file was created and not added to the staging area or tracked the new file will not be committed.

**Undoing a Commit**

git reset --soft HEAD^

This resets back into staging

The HEAD^ means move back one on the timeline or move one under the HEAD

**Adding To A Commit**

Maybe we forgot a file

git add filename

git commit --amend -m “commit message”

This adds to the previous commit

**Useful Commands**

git reset --soft HEAD^

undo the last commit, put changes into staging

git commit --amend -m “commit message”

Amends the last commit

git reset --hard HEAD^

undo last commit and all changes

git reset --hard HEAD^^

undo the last two commits and all changes

DON’T DO THESE AFTER YOU PUSH

**How to Share**

Git doesn’t take care of access control. That’s where services like Github come into play.

Hosted Solution

Github

bitbucket

Self managed

gitosis

gitorious

**Adding a Remote**

git remote add origin <https://github.com/Gregg/git-real.git>

add is adding the new remote

origin is the alias we call our remote server

and the url is the link to the repo

git remote -v

shows all remotes

**Pushing to a remote**

git push -u origin master

origin is the remote repo name

master is the local branch to push

It will ask for your github user name and password to authorize the push.

**Pulling From Remote**

To pull changes from the remote

git pull

This pulls the remote repo to the local and syncs them.

**Having Multiple Remotes**

Test remote

Hosting remote

Production

git remote add remoteName remoteURL

This adds a new remote

git remote rm remoteName

This removes a remote

git push -u remoteName branchName

the -u this makes it so that the next time you do a push you don’t have to specify the remote or the branch unless your are working in a different remote and/or branch.

**Heroku**

To create a repo with heroku

heroku create

This creates the remote automatically with an ssh address

To push to heroku

git push heroku master

when heroku gets the code it deploys it