Git + Github

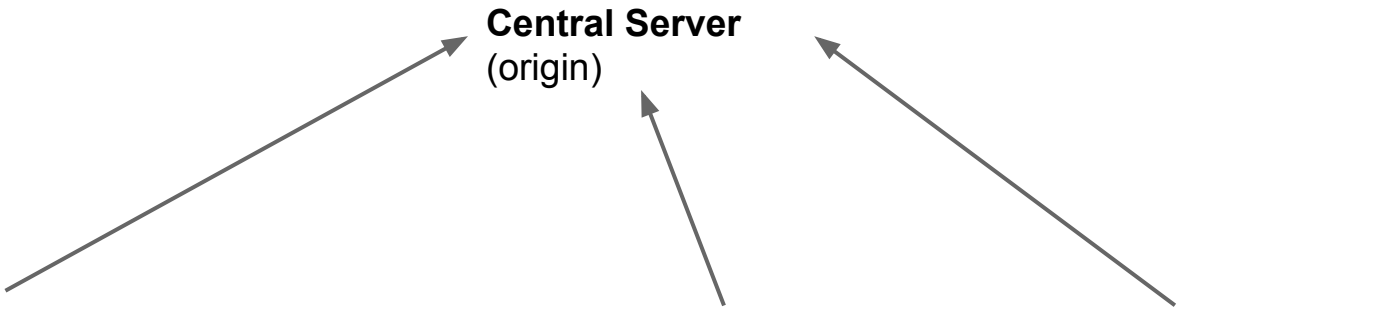
What is git?

Distributed Version Control System (DVCS)

- Central version of repository on server
- Local version of repository on your machine
- Commit changes to local repository and push changes to central repository

Stores history as a mini-file system with references to files for each commit (each commit is a snapshot)

git



Person 1
2 commits ahead of origin/master

Person 2
0 commits ahead of origin/master + local changes

Person 3
10 commits behind origin/master

What is Github?

- Remote
- Nice visual UI

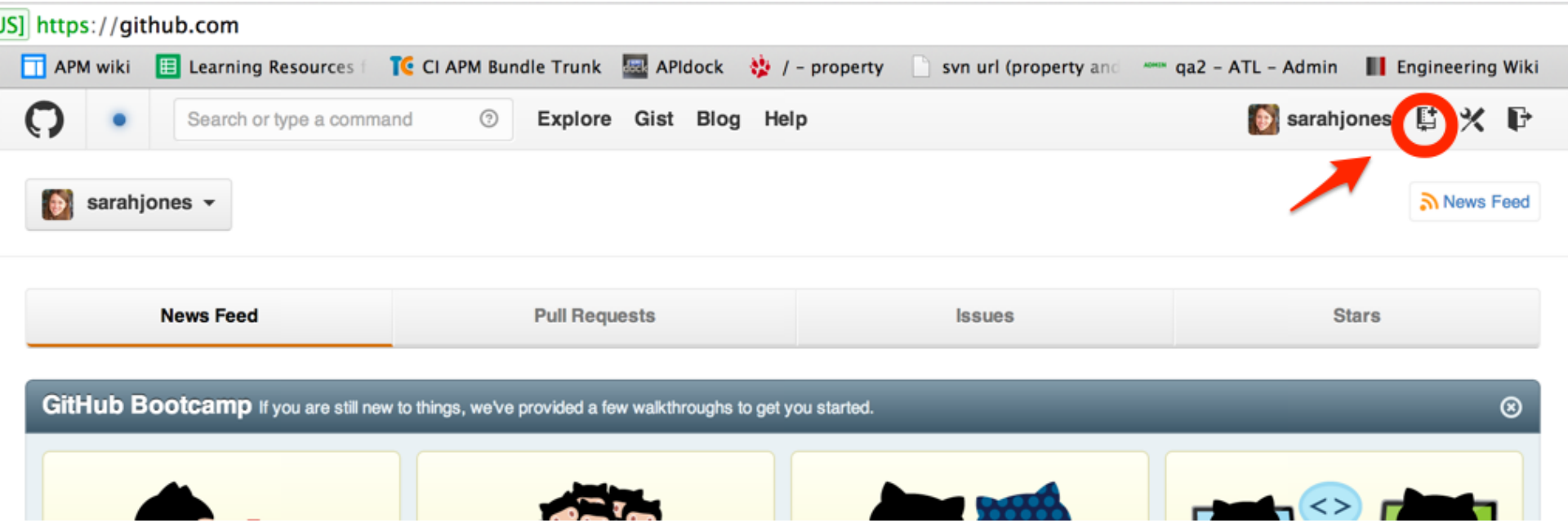
Setup SSH Keys

1. Account Settings (in the top right corner)
2. SSH Keys
3. Add SSH key
    a. copy-paste in your public key

Alternatively

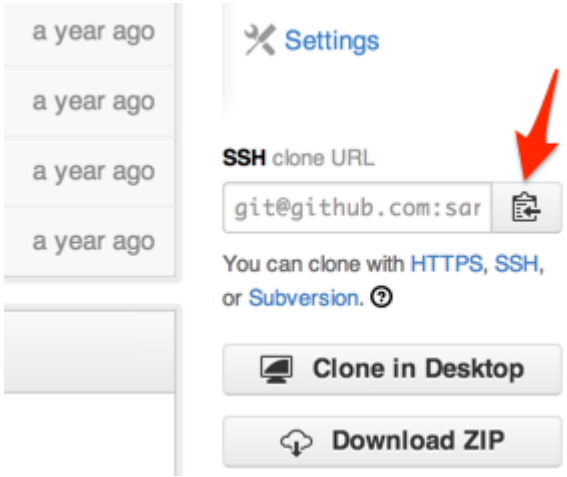
- Type in username and password every time you push

Create a Repo



Get your git repo locally

- On command line, in the directory you want
- git clone <repo url>



git status

staged (changes):

- these changes will be committed with the commit command

tracked (changes):

- git knows about this file, there are changes, but if you commit now, they won't be saved in that commit

untracked (files):

- git does not know anything about this file, but it is located on your filesystem in one of the directories of your repo

Commit

- Add a untracked or unstaged file to stage

git add <filepath>

- Unstage a file

git reset <filepath>

- Make a local commit

git commit -m “My descriptive commit message”

Other helpful commands

- See diff of unstaged work

git diff

- See diff of staged work

git diff --cached

- Throw away all unstaged/staged changes since last local commit

git reset --hard HEAD

Push (commit remotely)

Any time after making a local commit:

- if there are changes that are not committed

git stash #stash away the changes

- get latest changes

git pull --rebase

- push work to remote

git push

- if you stashed, get your changes back

git stash pop

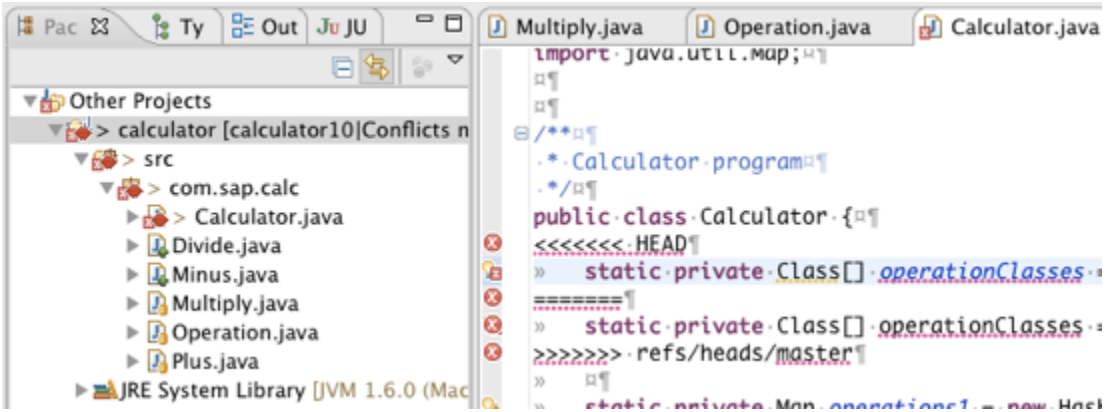
Conflicts

- I want to push my changes, but I know I need the latest code first: git pull --rebase
- Git says I have a conflict

1. Open files with conflicts
2. Resolve conflicts
3. git add <filepath> #for each of those files to mark conflict as resolved
4. git rebase --continue # (if this is necessary the context will indicate it)

Note: if you stashed code, sometimes unstashing causes conflicts. These conflicts should be treated the same way

Resolving conflict in a file




```
<<<<<<<<<< label1
    //the code that label1 thinks is here
==========
    //the code that label2 thinks is here
>>>>>>>>>> label2
```

Resolving conflict in a file

- Git isn't smart enough to figure out what the code should look like
- You manually choose what the code should look like (label1, label2, or a combination of the two)
- Remove <<<<<, >>>>>, ===== lines
- Once the file looks like you want it to be, save the file

Conflicts

- I want to push my changes, but I know I need the latest code first: git up
- Git says I have a conflict

1. Open files with conflicts
2. Resolve conflicts
3. git add <filepath> #for each of those files to mark conflict as resolved
4. git rebase --continue #(if this is necessary the context will indicate it)

Note: if you stashed code, sometimes unstashing causes conflicts. These conflicts should be treated the same way

Branches

- Main branch = master
- Branch = Pointer to a series of commits
- Recommendation for this class: only use the master branch for development (could use branches to save working code for demos)
  - Class is fast paced and want to be on same page
  - If you haven't used git before merging branches can be challenging

Resources

- Create a Github account - github.com
- Free **private** student repos - https://github.com/edu
- Documentaiton - http://git-scm.com/documentation

References

http://git-scm.com/documentation

Merge conflict pic: http://wiki.eclipse.org/images/b/b9/Egit-0.10-merge-conflict.png