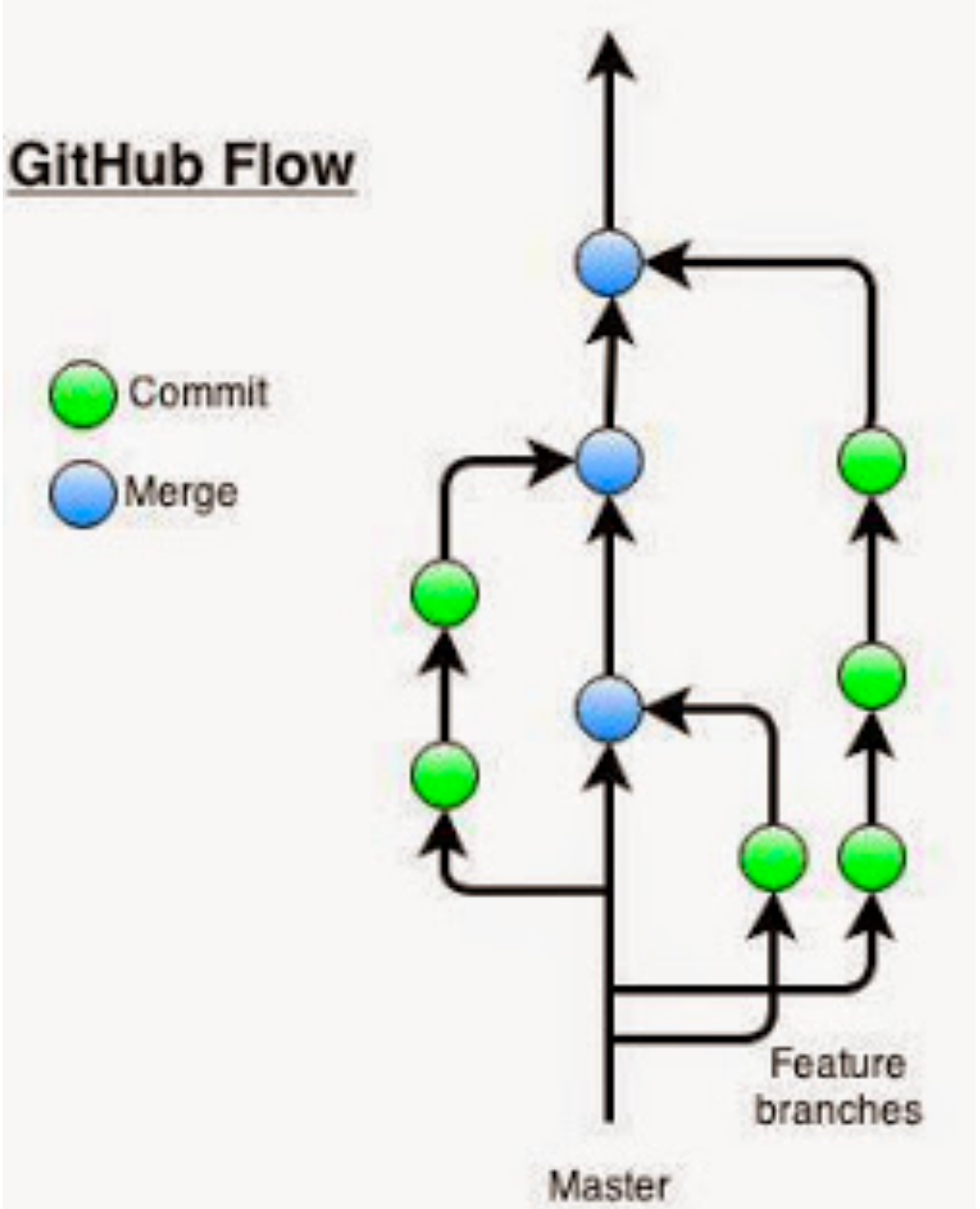
Intro to Git

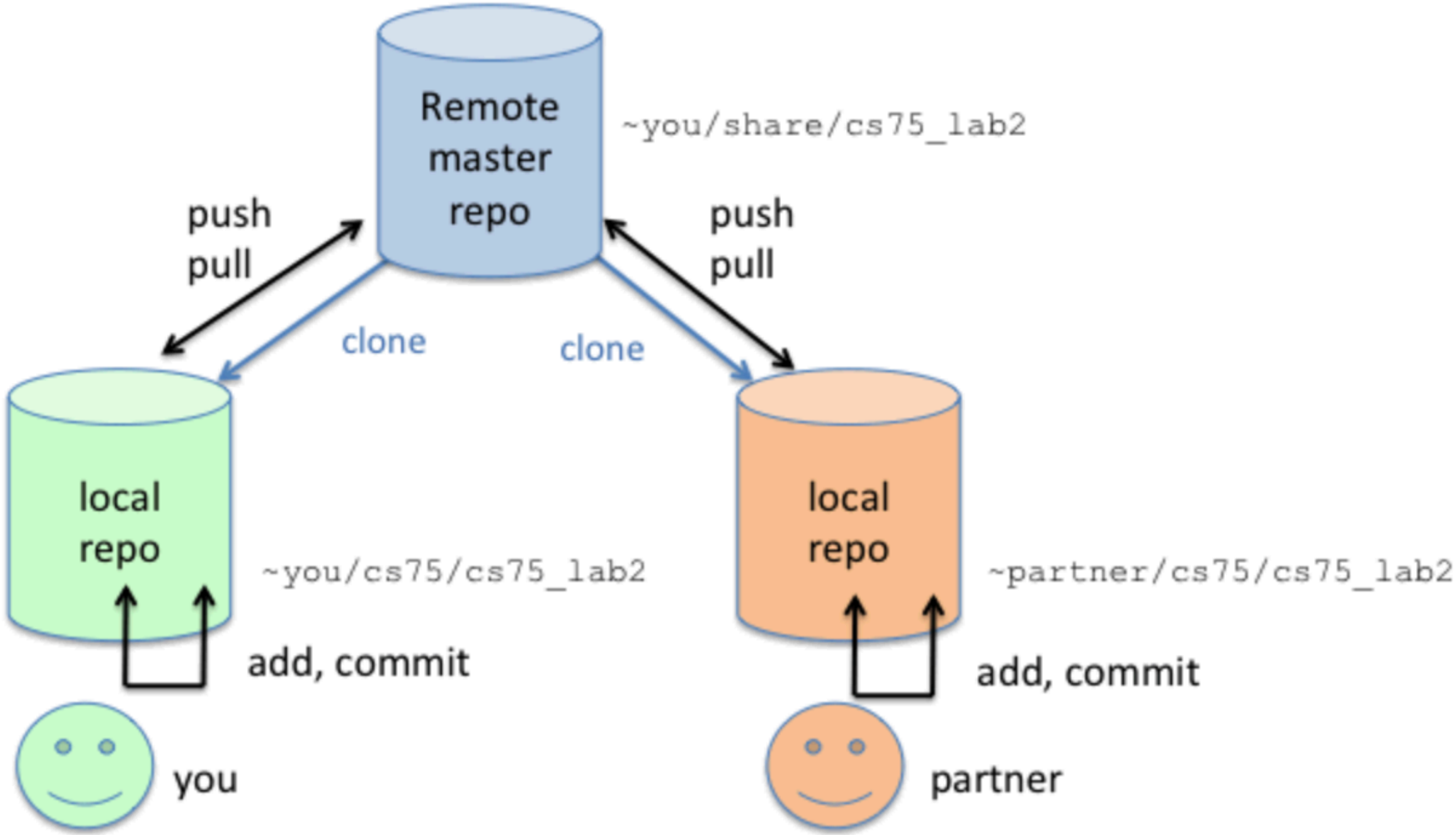
HWRS 482/582

GitHub is a version control system. It allows teams to work collaboratively on the same pieces of code (like track changes for word but much more sophisticated)



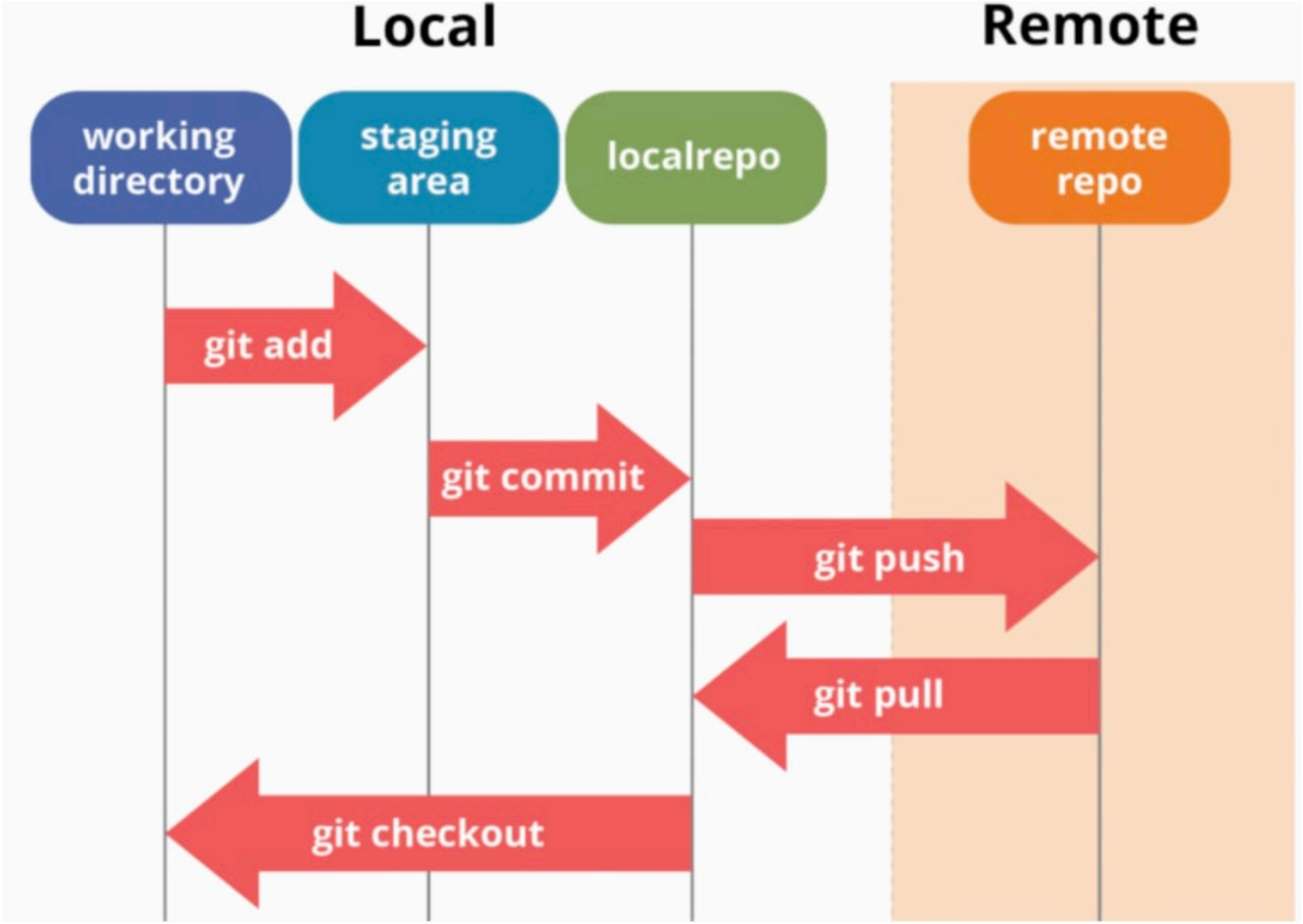
https://www.endpoint.com/blog/2014/05/02/git-workflows-that-work

Local and remote version-controlled repositories



https://www.cs.swarthmore.edu/~newhall/unixhelp/git_create.php

GitHub workflow



https://dev.to/mollynem/git-github--workflow-fundamentals-5496

https://www.reddit.com/r/git/comments/99ul9f/git_workflow_diagram_showcasing_the_role_of/

checkout a repository

create a working copy of a local repository by running the command

```
git clone /path/to/repository
```

when using a remote server, your command will be

```
git clone username@host:/path/to/repository
```

Copied from: http://rogerdudler.github.io/git-guide/

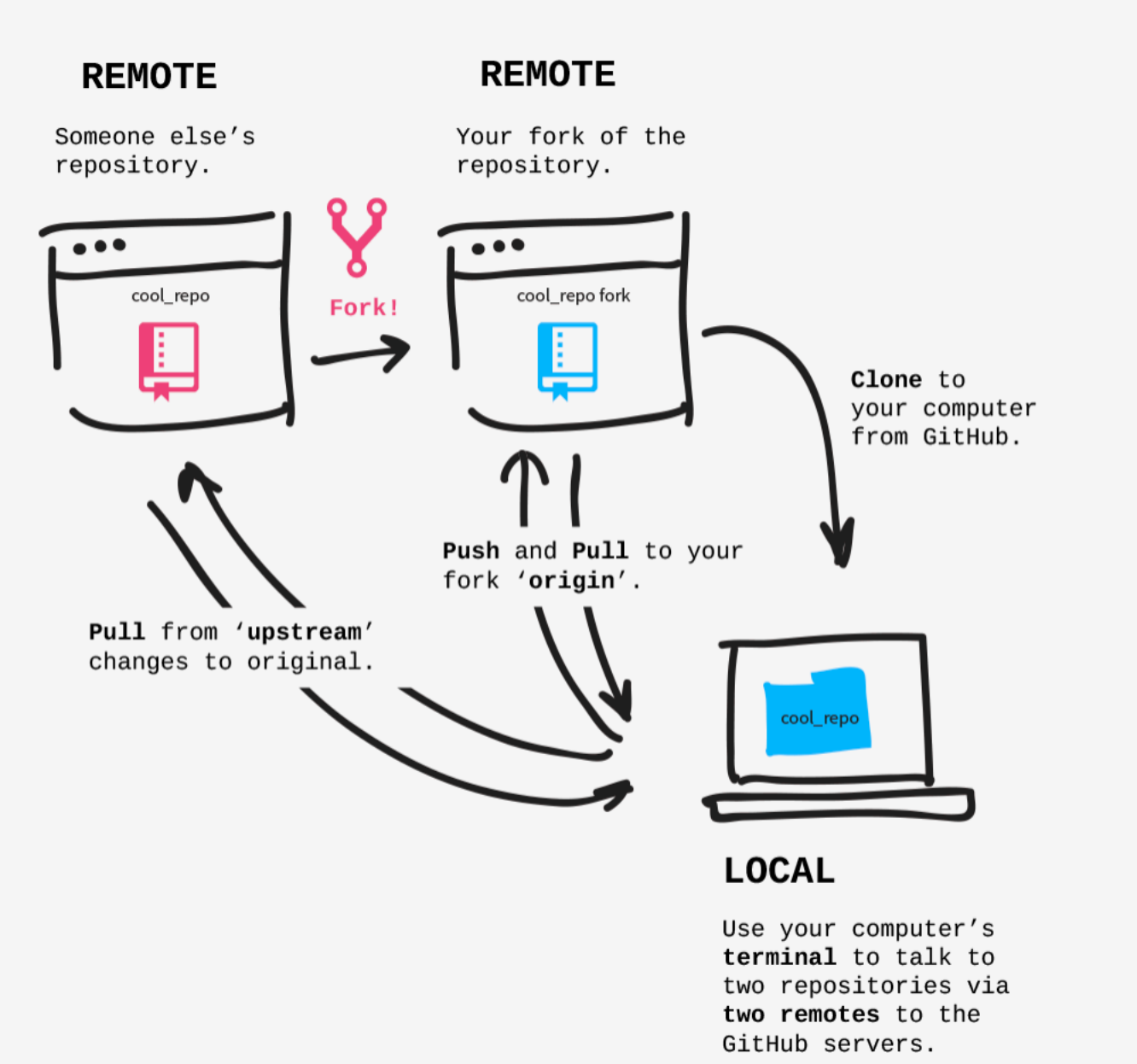
Check your setup

(Note you must be inside your repo for these to work)

- *git status* : to see whether your local repo is up to date and what branch you are on
- *git branch* : shows you want branch you are currently working on
- *git remote*: Shows the remote repose you are connected to
- *git remote show origin*: To show the url for a given remote (in this case origin)

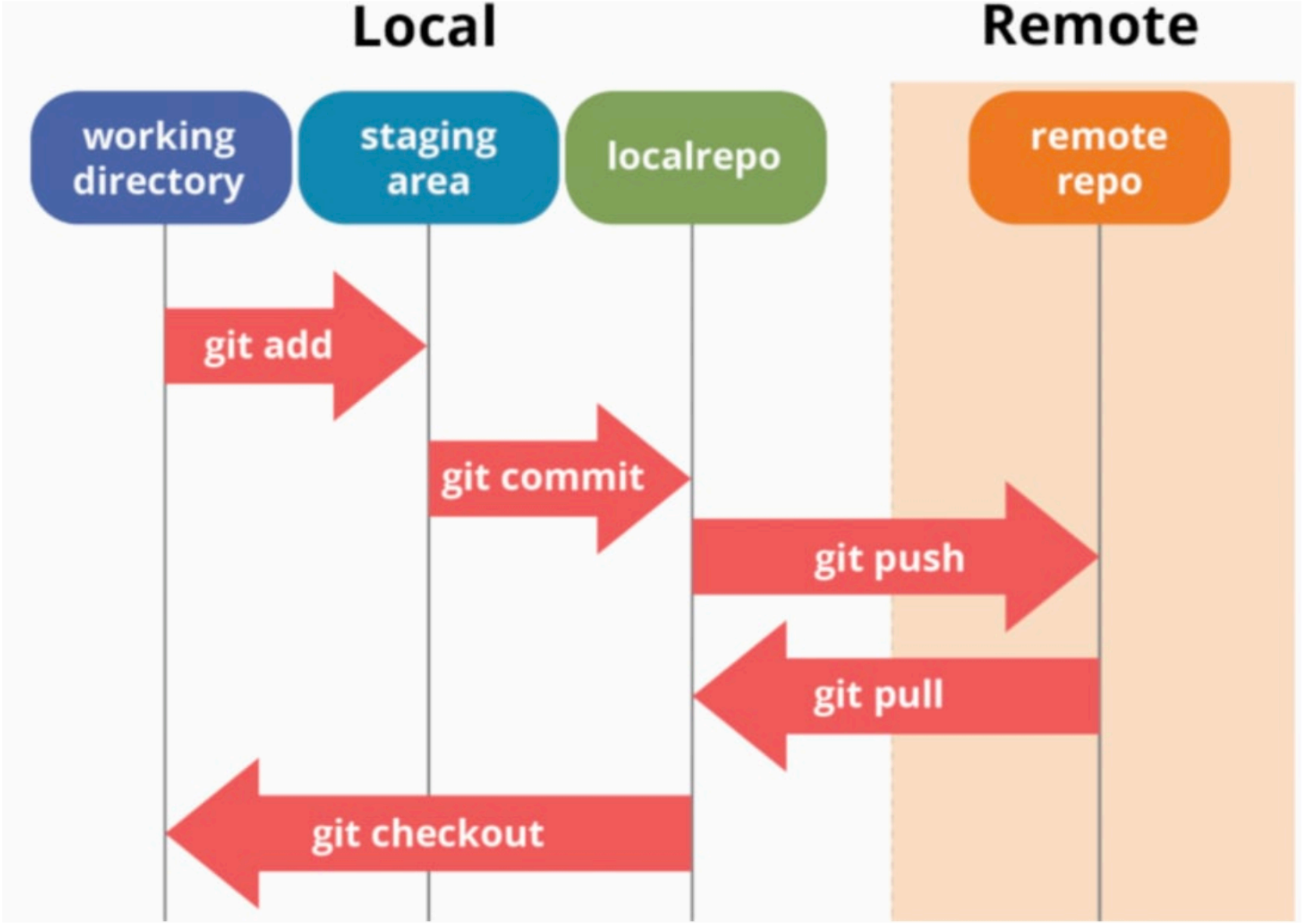
Cloning vs. Forking

- Cloning is you making a local copy of a repository
- If you Fork first then you have your own version of the repository remotely that you can pull and push changes to
- For this class we will just clone but for future work you may want to fork if there is a repo you want to make changes to without changing someone else's repository



http://jlord.us/git-it/challenges/forks_and_clones.html

GitHub workflow



https://dev.to/mollynem/git-github--workflow-fundamentals-5496

https://www.reddit.com/r/git/comments/99ul9f/git_workflow_diagram_showcasing_the_role_of/

workflow

your local repository consists of three "trees" maintained by git. the first one is your `Working Directory` which holds the actual files. the second one is the `Index` which acts as a staging area and finally the `HEAD` which points to the last commit you've made.

Copied from: http://rogerdudler.github.io/git-guide/

add & commit

You can propose changes (add it to the **Index**) using

```
git add <filename>
```

```
git add *
```

This is the first step in the basic git workflow. To actually commit these changes use

```
git commit -m "Commit message"
```

Now the file is committed to the **HEAD**, but not in your remote repository yet.

Copied from: http://rogerdudler.github.io/git-guide/

pushing changes

Your changes are now in the **HEAD** of your local working copy. To send those changes to your remote repository, execute

```
git push origin master
```

Change *master* to whatever branch you want to push your changes to.

If you have not cloned an existing repository and want to connect your repository to a remote server, you need to add it with

```
git remote add origin <server>
```

Now you are able to push your changes to the selected remote server

Copied from: http://rogerdudler.github.io/git-guide/