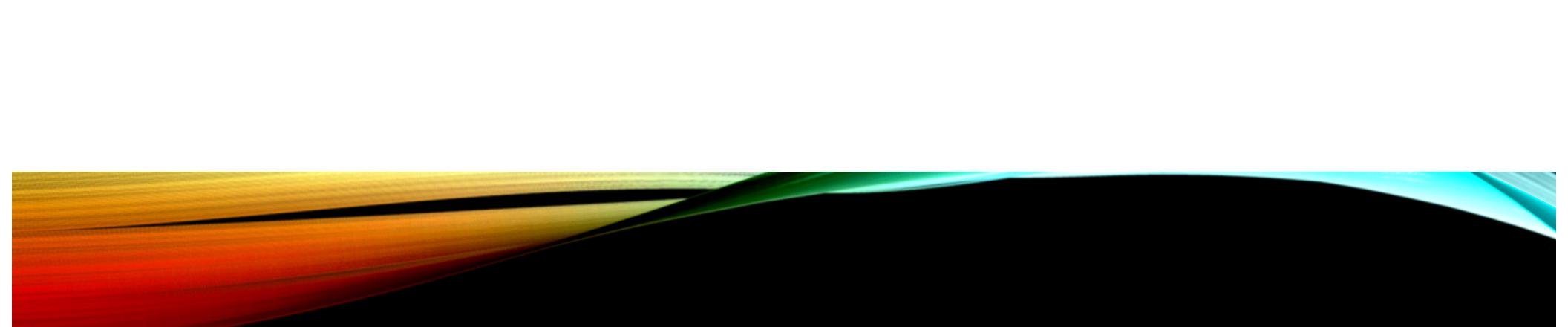


# GIT AND GITHUB

Matty B



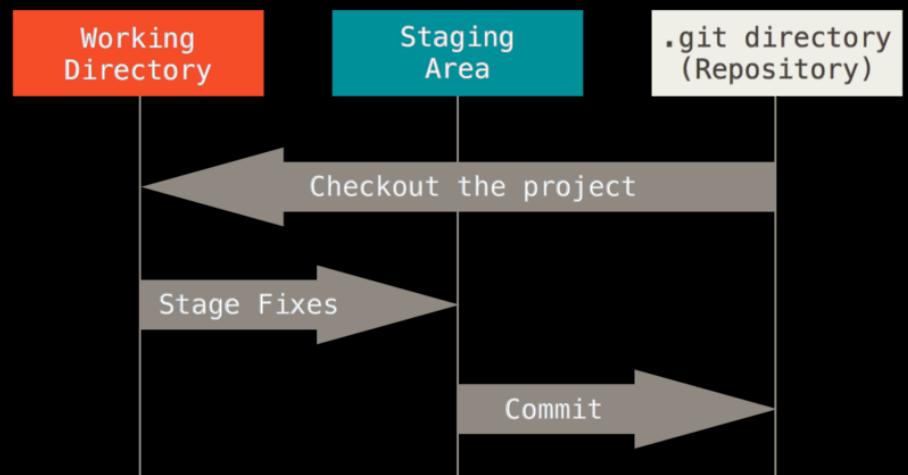


# GITTING YOU TO UNDERSTAND GIT

- VERSION CONTROL
- Git is here to help you monitor the changes you make to files
- Takes a “snapshot” of your file and stores a reference
- These “snapshots” show the differences you make to a file
- Works locally! You can check the history of your “snapshots” instantly and revert to old code.

# HOW GIT WORKS

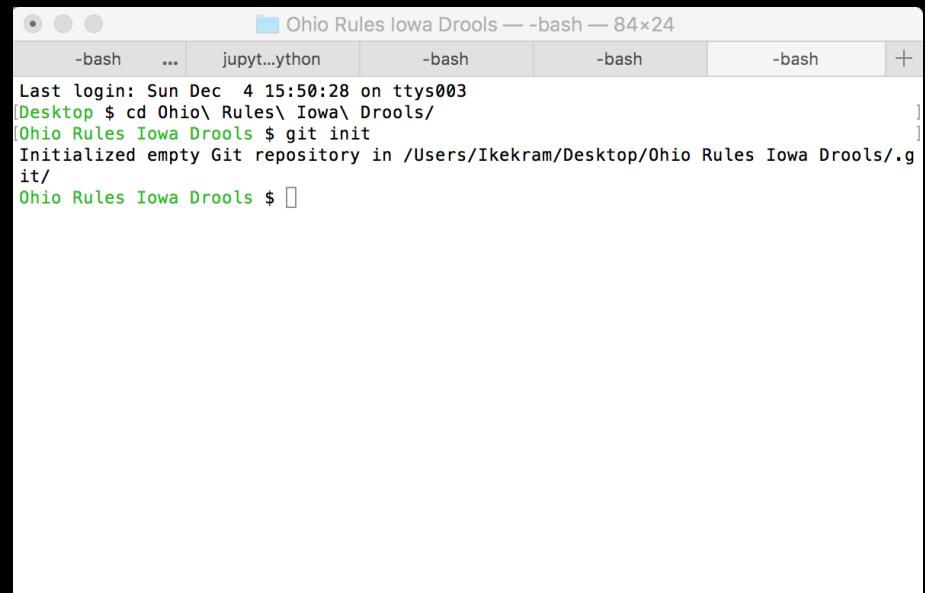
1. Do work or modify files from the working directory.
2. Take “snapshots” of the files and add them to the staging area.
3. Store the “snapshots” of the files by committing them.



# HOW TO GIT THIS TO WORK

## Initialize the Repository

1. From the terminal navigate to the folder you would like to be your repository
2. From here type 'git init'
3. This folder has now been initialized as a local repository



A screenshot of a terminal window titled "Ohio Rules Iowa Drools — -bash — 84x24". The window shows a command-line interface with the following text:

```
Last login: Sun Dec  4 15:50:28 on ttys003
[Desktop $ cd Ohio\ Rules\ Iowa\ Drools/
[Ohio Rules Iowa Drools $ git init
Initialized empty Git repository in /Users/Ikekram/Desktop/Ohio Rules Iowa Drools/.git/
Ohio Rules Iowa Drools $ ]
```

# HOW TO GIT THIS TO WORK

## Add File to the Staging Area

1. Create a file in our local repository
2. It is currently not being tracked as it has never been committed
3. Then add it to the staging area with '**git add .**' (the '.' adds all the files in the repo to the staging area)

```
Ohio Rules Iowa Drools — bash — 84x24
-bash ... jupyter...python -bash -bash -bash +
Ohio Rules Iowa Drools $ echo 'Ohio Make Me Happy' > README
Ohio Rules Iowa Drools $ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    README

nothing added to commit but untracked files present (use "git add" to track)
Ohio Rules Iowa Drools $ git add .
Ohio Rules Iowa Drools $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

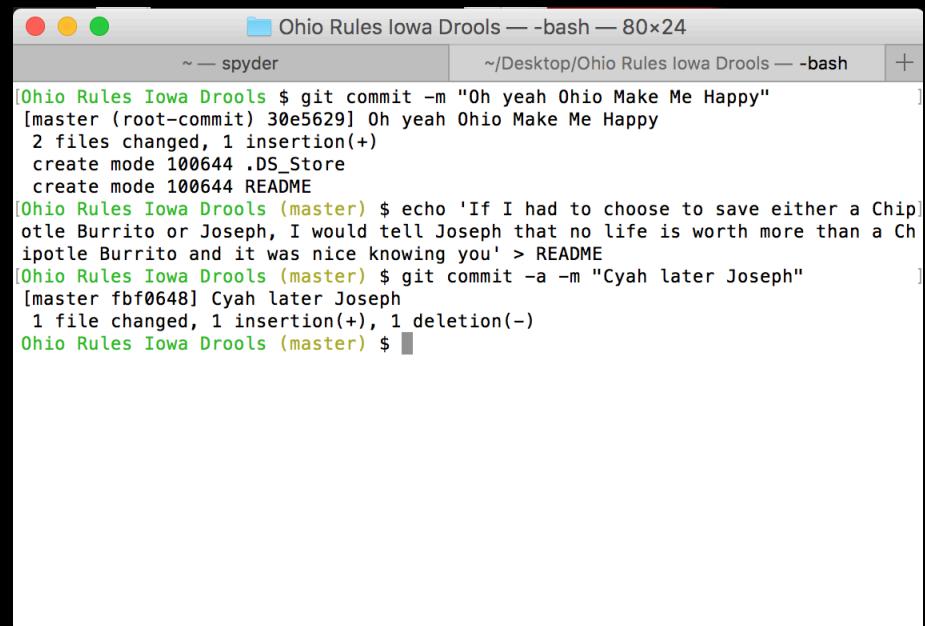
    new file: README

Ohio Rules Iowa Drools $
```

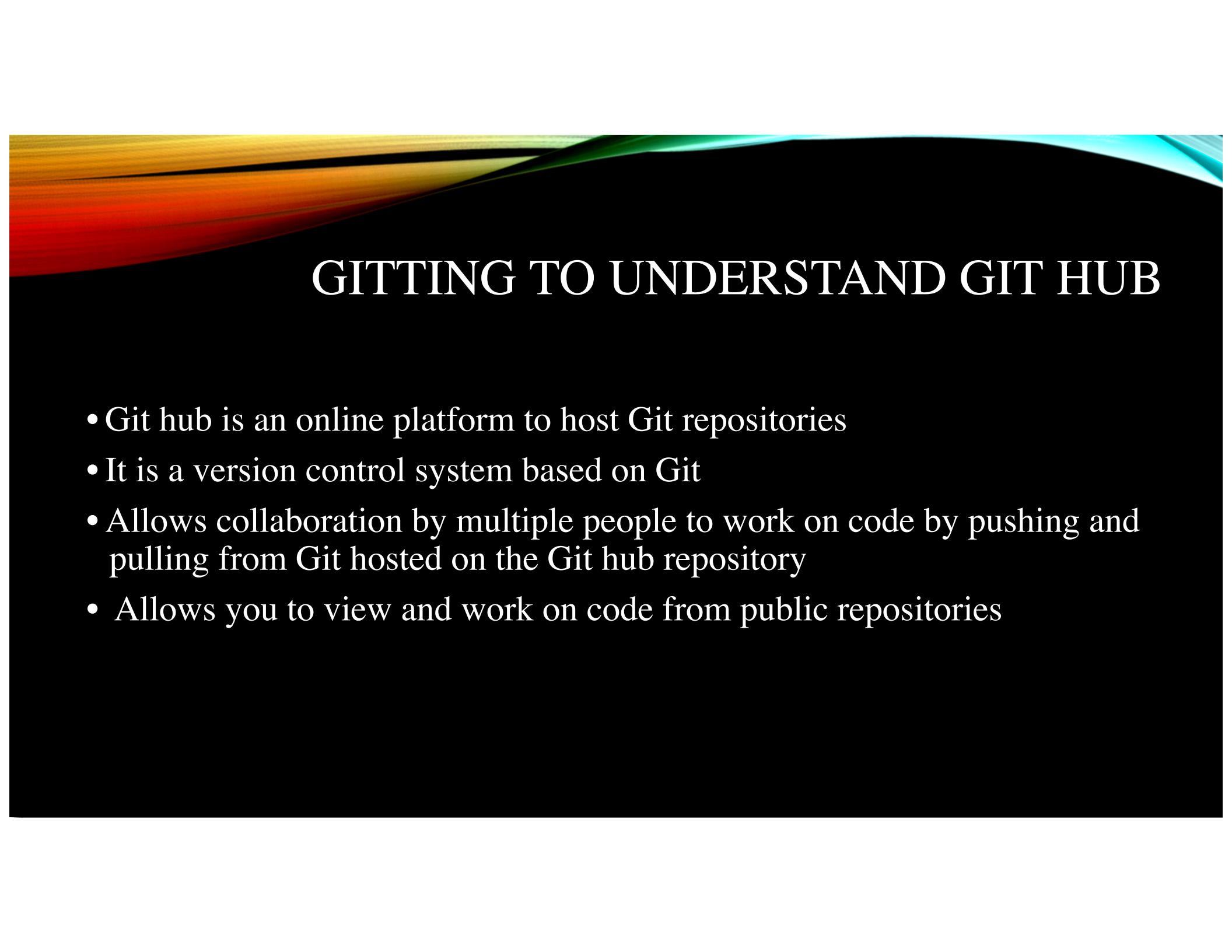
# HOW TO GIT THIS TO WORK

## Committing your files

- Use the 'git commit -m "insert comment here"' command
- This stores a "snapshot" of your file
- You can use the 'git commit -a -m "insert comment here"' to add and commit in the same command



```
Ohio Rules Iowa Drools $ git commit -m "Oh yeah Ohio Make Me Happy"
[master (root-commit) 30e5629] Oh yeah Ohio Make Me Happy
 2 files changed, 1 insertion(+)
  create mode 100644 .DS_Store
  create mode 100644 README
Ohio Rules Iowa Drools (master) $ echo 'If I had to choose to save either a Chipotle Burrito or Joseph, I would tell Joseph that no life is worth more than a Chipotle Burrito and it was nice knowing you' > README
Ohio Rules Iowa Drools (master) $ git commit -a -m "Cyah later Joseph"
[master fbf0648] Cyah later Joseph
 1 file changed, 1 insertion(+), 1 deletion(-)
Ohio Rules Iowa Drools (master) $
```

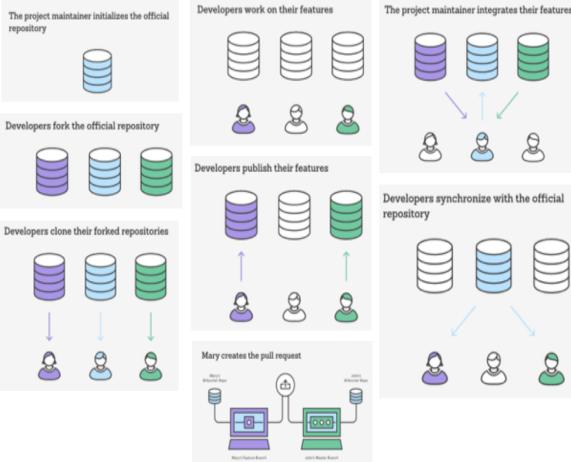


# GITTING TO UNDERSTAND GIT HUB

- Git hub is an online platform to host Git repositories
- It is a version control system based on Git
- Allows collaboration by multiple people to work on code by pushing and pulling from Git hosted on the Git hub repository
- Allows you to view and work on code from public repositories

# THE GITHUB WORKFLOW

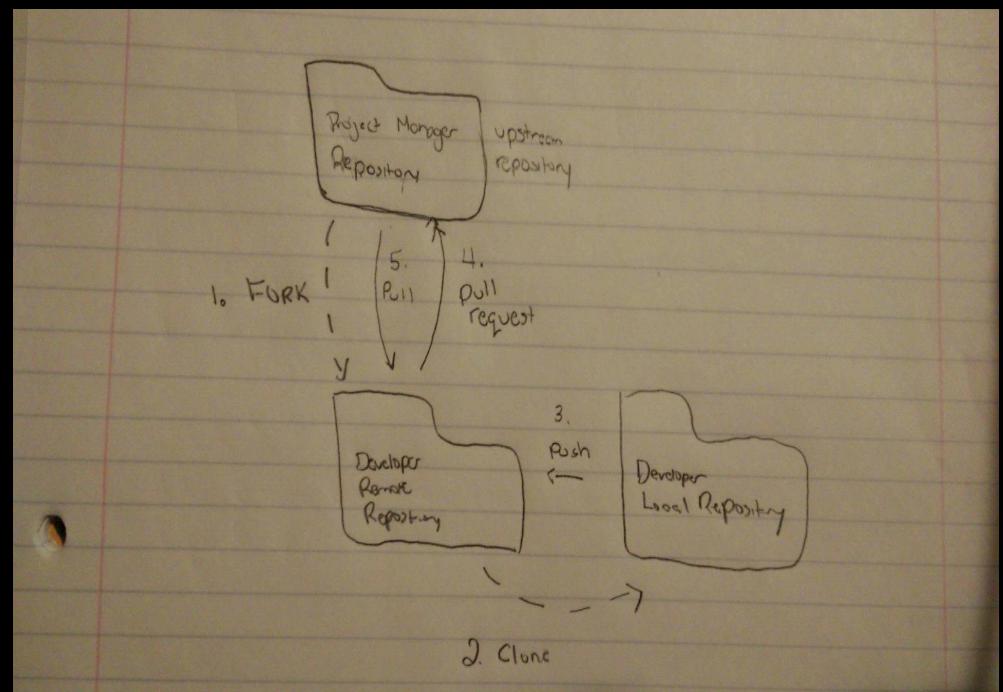
## FORKING WORKFLOW



The Forking Workflow is fundamentally different than other workflows.

Instead of using a single server-side repository to act as the “central” codebase, it gives every developer a server-side repository.

This means that each contributor has not one, but two Git repositories: a private local one and a public server-side one.



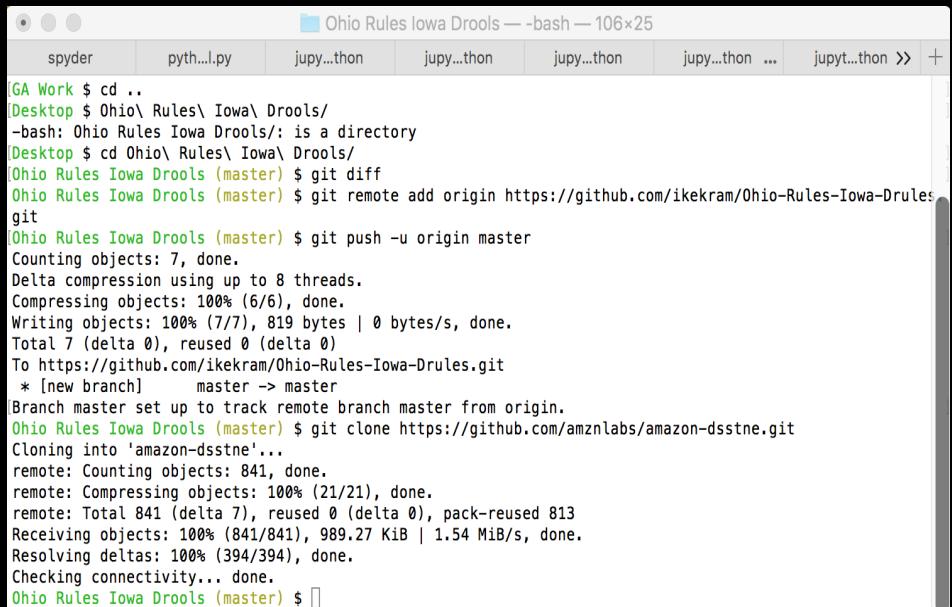


# CREATING A NEW REMOTE REPOSITORY

1. Go to GITHUB.com
2. Select 'NEW REPOSITORY'
3. Create a repository name
4. Select 'CREATE REPOSITORY'
5. Copy and paste info into terminal from inside location of your local repository:
  - \$ git remote add origin https://github.com/ikekram/Ohio-Rules-Iowa-Drules.git
  - \$ git push -u origin master

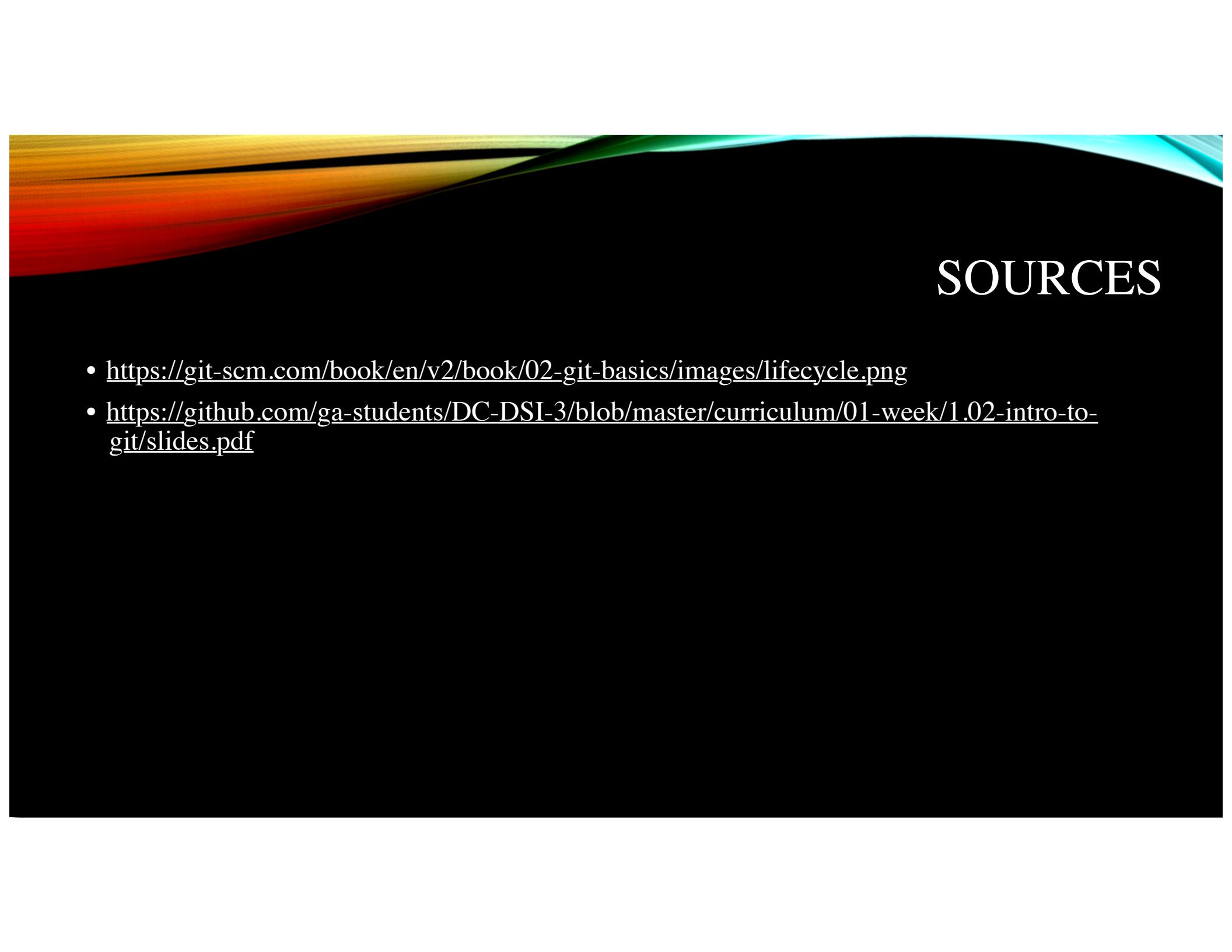
# CLONING PUBLIC REPOS

- Lets clone the Amazon Deep Learning Machine Learning model repo
- From inside repo type:
- `git clone https://github.com/amznlabs/amazon-dsstne.git`
- Allows us to work on this public code



The screenshot shows a terminal window titled "Ohio Rules Iowa Drools — bash — 106x25". The terminal is displaying a command-line session for cloning a GitHub repository. The session starts with the user navigating to a directory, then cloning the "amazon-dsstne" repository from the "ikekram" user. The output shows the progress of the cloning process, including the number of objects, compression levels, and the final status of the cloned repository.

```
GA Work $ cd ..
/Desktop $ Ohio\ Rules\ Iowa\ Drools/
-bash: Ohio Rules Iowa Drools/: is a directory
/Desktop $ cd Ohio\ Rules\ Iowa\ Drools/
Ohio Rules Iowa Drools (master) $ git diff
Ohio Rules Iowa Drools (master) $ git remote add origin https://github.com/ikekram/Ohio-Rules-Iowa-Drules.git
Ohio Rules Iowa Drools (master) $ git push -u origin master
Counting objects: 7, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 819 bytes | 0 bytes/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/ikekram/Ohio-Rules-Iowa-Drules.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
Ohio Rules Iowa Drools (master) $ git clone https://github.com/amznlabs/amazon-dsstne.git
Cloning into 'amazon-dsstne'...
remote: Counting objects: 841, done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 841 (delta 7), reused 0 (delta 0), pack-reused 813
Receiving objects: 100% (841/841), 989.27 KiB | 1.54 MiB/s, done.
Resolving deltas: 100% (394/394), done.
Checking connectivity... done.
Ohio Rules Iowa Drools (master) $
```



# SOURCES

- <https://git-scm.com/book/en/v2/book/02-git-basics/images/lifecycle.png>
- <https://github.com/ga-students/DC-DSI-3/blob/master/curriculum/01-week/1.02-intro-to-git/slides.pdf>