



Introduction to Git and GitHub

GIS714

Spring 2024



Introduction to Git and GitHub

Part 1: Git concepts and vocabulary

Part 2: Collaborating with GitHub

Part 3: Hands-on activity - Open your own Pull Request



vs.



Open-source version-control software
on your computer

Probably already installed on your OS:

```
git --version
```

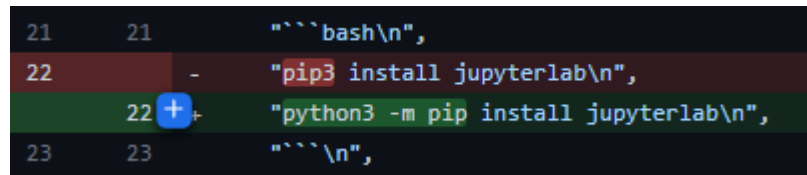
Company owned by Microsoft that
hosts Git Repositories in the cloud

Alternatives: Bitbucket, GitLab, GitHub
Enterprise

Provides features for sharing code,
collaborating, raising issues, project
tracking and more

What is “version-control” any ways?

- “a system that records changes to a file or set of files over time so that you can recall specific versions later.” - git-scm.com

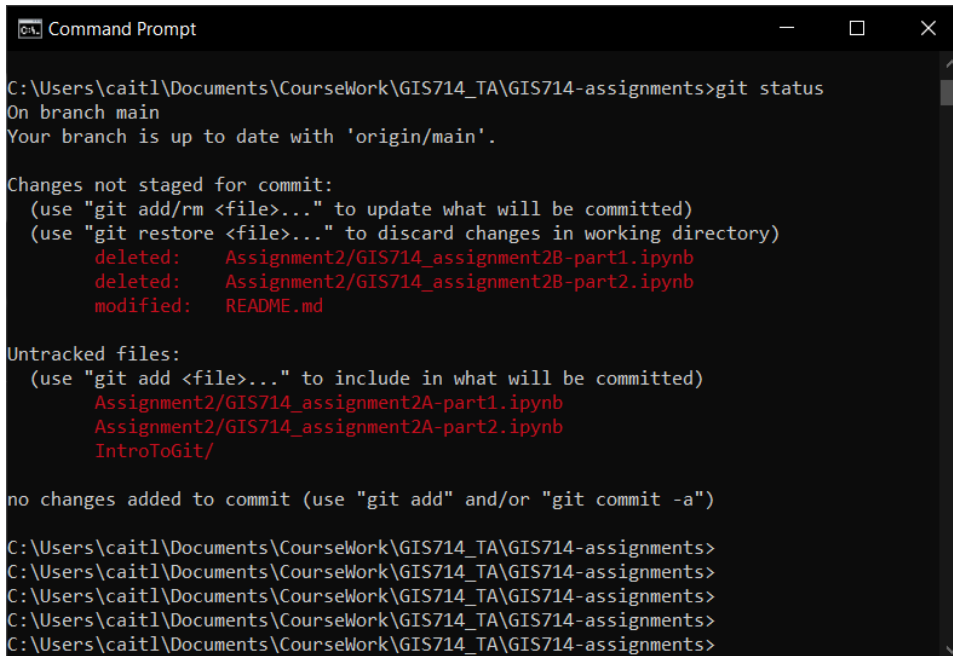


```
21      21      """bash\n",\n22      -      "pip3 install jupyterlab\n",\n22      +      "python3 -m pip install jupyterlab\n",\n23      23      """\n",
```

- Keeps track of multiple versions of a set of files (branches)
- Allows you to move back to older versions without deleting current work (revert/reset/checkout)

How do I use Git on my local machine?

- Command Line Interface (CLI)
 - [Install Git](#)
 - [Configure SSH Key Authentication with GitHub](#)
- Graphical User Interfaces
 - GitHub Desktop, GitKraken, SourceTree



```
Command Prompt

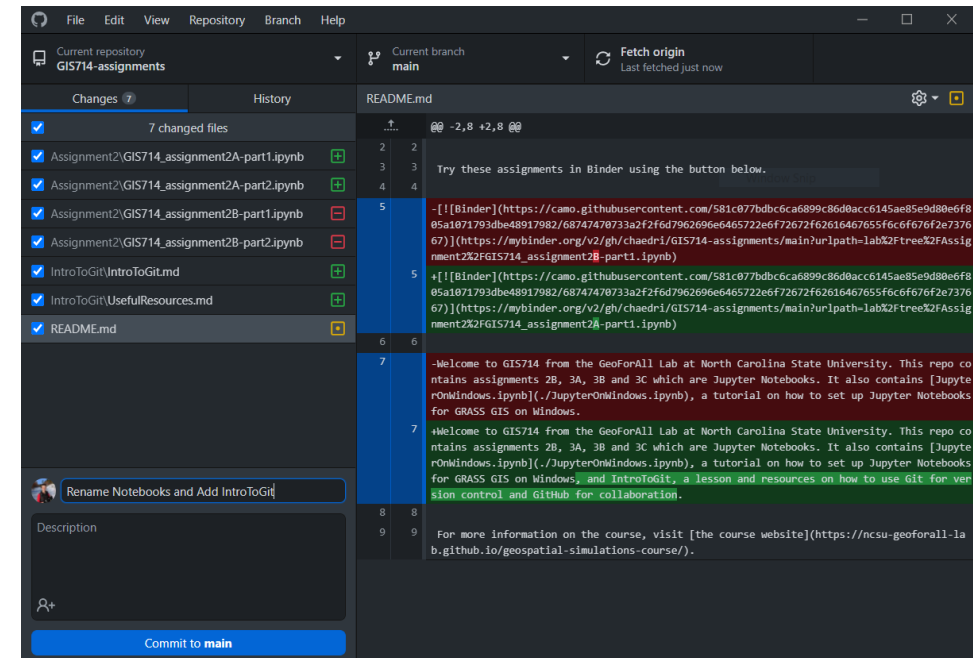
C:\Users\caitl\Documents\CourseWork\GIS714_TA\GIS714-assignments>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
       deleted:    Assignment2\GIS714_assignment2B-part1.ipynb
       deleted:    Assignment2\GIS714_assignment2B-part2.ipynb
       modified:   README.md

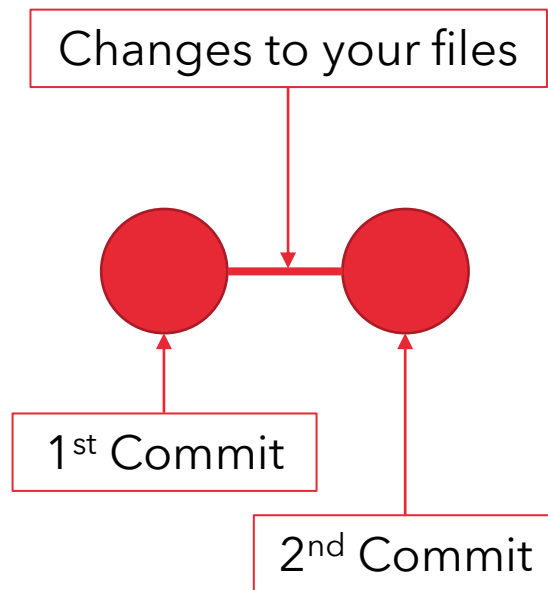
Untracked files:
  (use "git add <file>..." to include in what will be committed)
       Assignment2\GIS714_assignment2A-part1.ipynb
       Assignment2\GIS714_assignment2A-part2.ipynb
       IntroToGit/

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\caitl\Documents\CourseWork\GIS714_TA\GIS714-assignments>
C:\Users\caitl\Documents\CourseWork\GIS714_TA\GIS714-assignments>
C:\Users\caitl\Documents\CourseWork\GIS714_TA\GIS714-assignments>
C:\Users\caitl\Documents\CourseWork\GIS714_TA\GIS714-assignments>
C:\Users\caitl\Documents\CourseWork\GIS714_TA\GIS714-assignments>
```



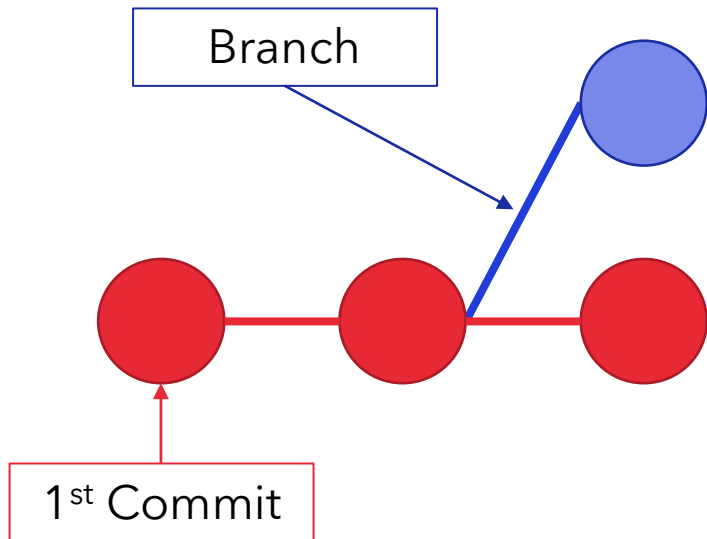
Basic Git: Commit



Commit your changes:

1. `git status` to see current state of directory
2. `git add` files that you want to commit
3. `git commit` changes with descriptions ("-m" flag)

Basic Git: Branch



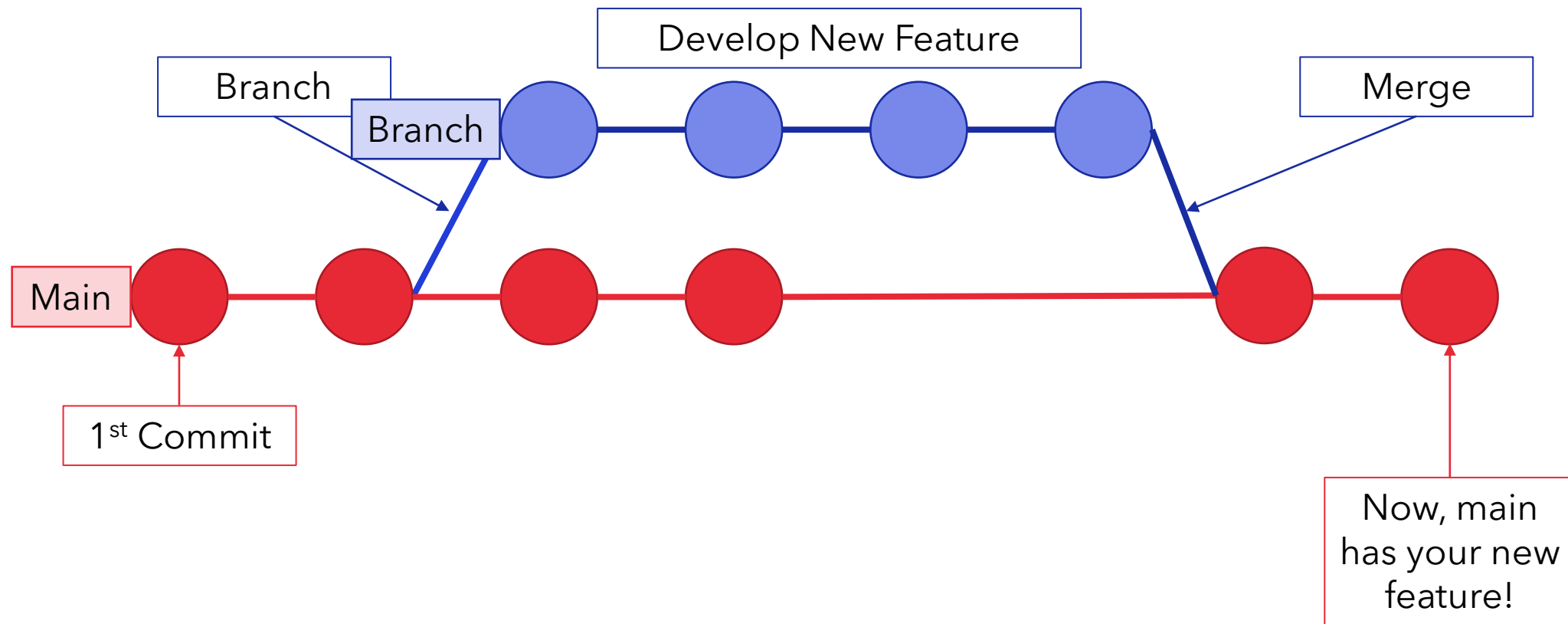
Make a branch

1. `git commit`
2. `git branch new-branch-name`
3. `git checkout new-branch-name`

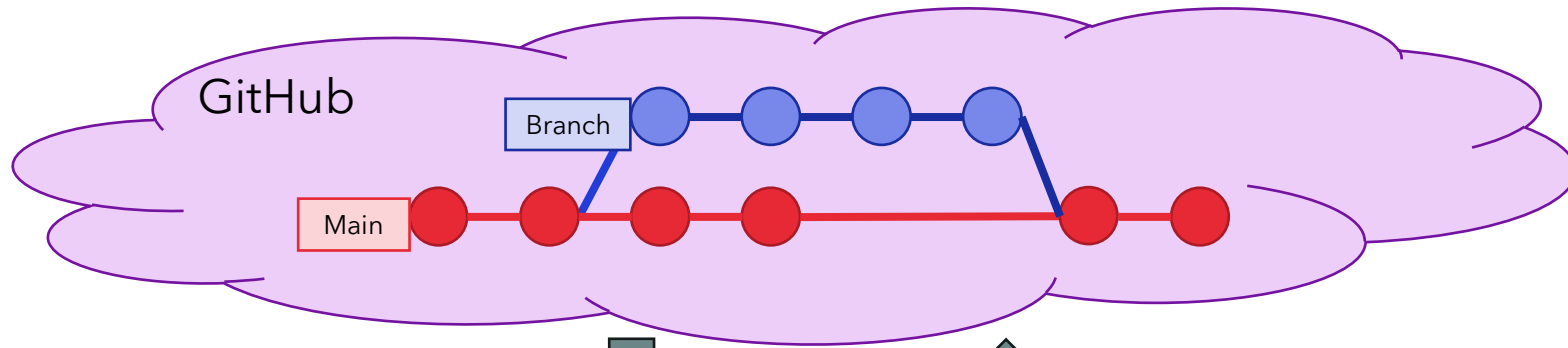
... shortcut for 2 and 3:

`git checkout -b new-branch-name`

Basic Git: Branch and Merge



So, what about GitHub?



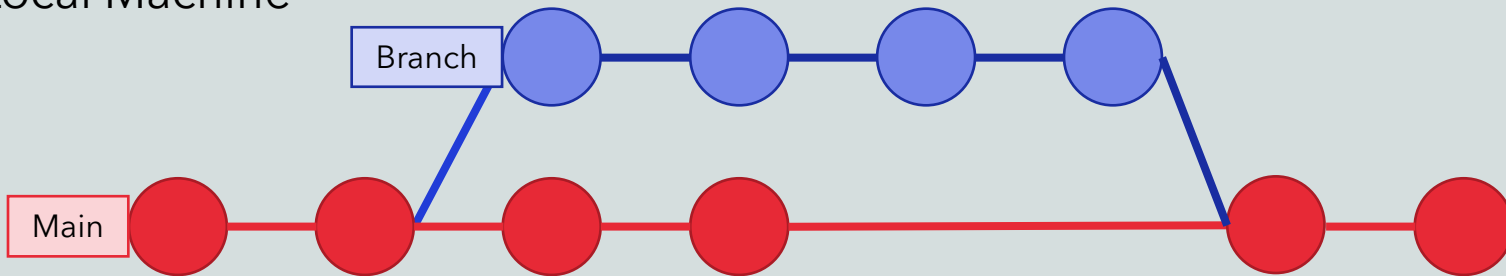
Pull



Push



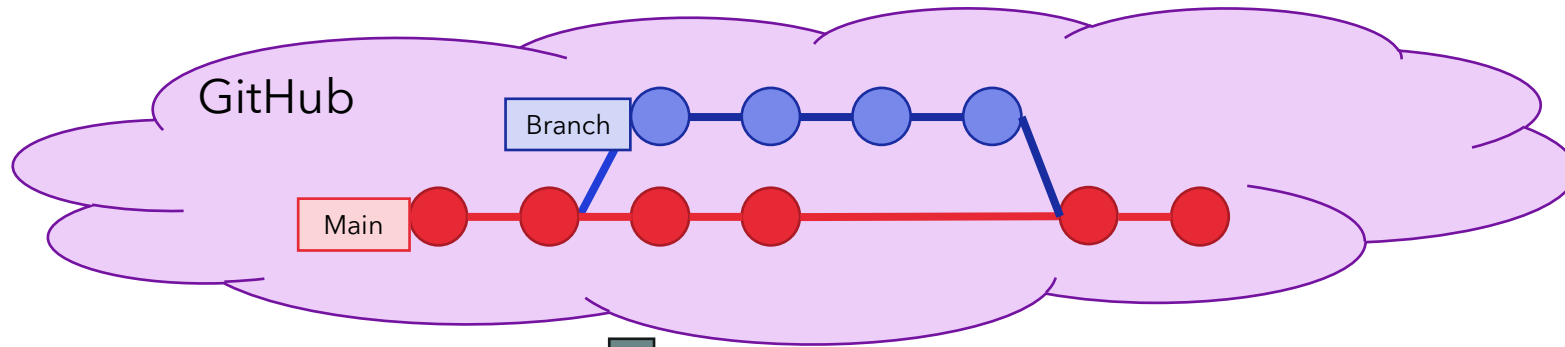
Local Machine



"I just pushed some changes"

"Let me pull the changes you just made so I can test them!"

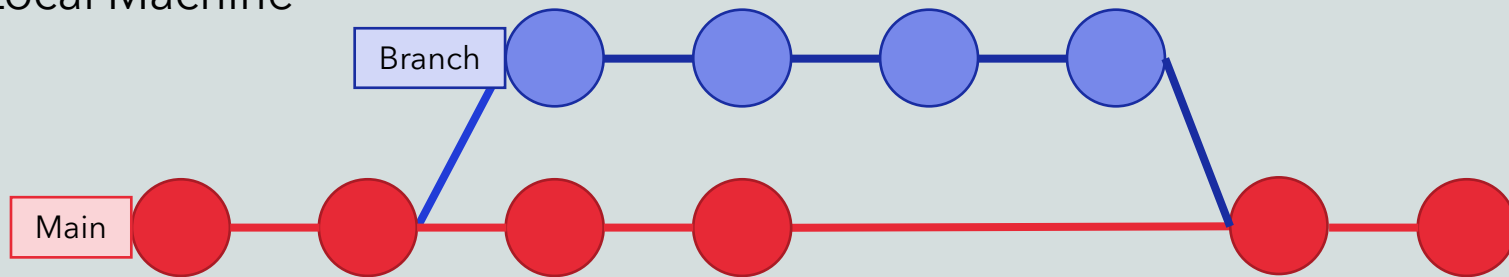
So, what about GitHub?



Clone



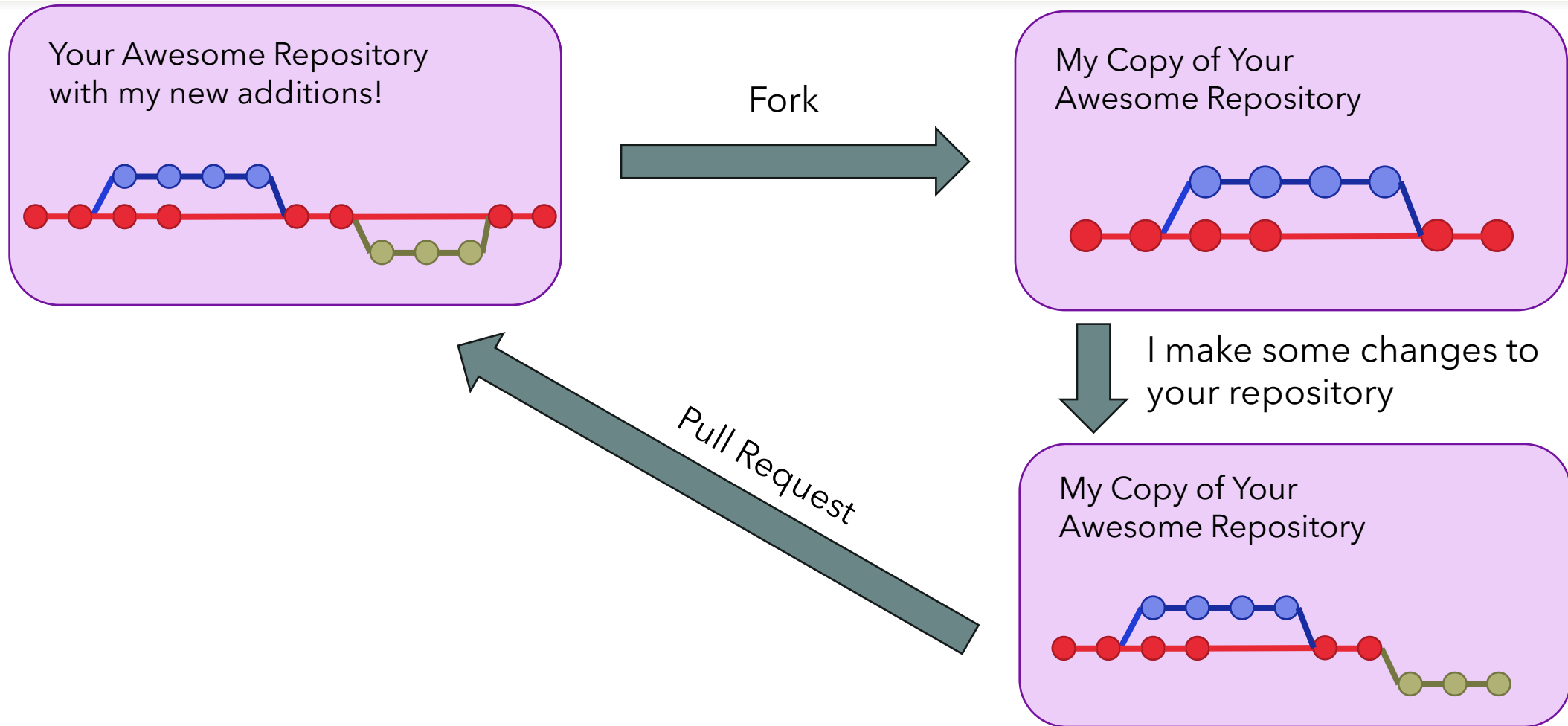
Local Machine



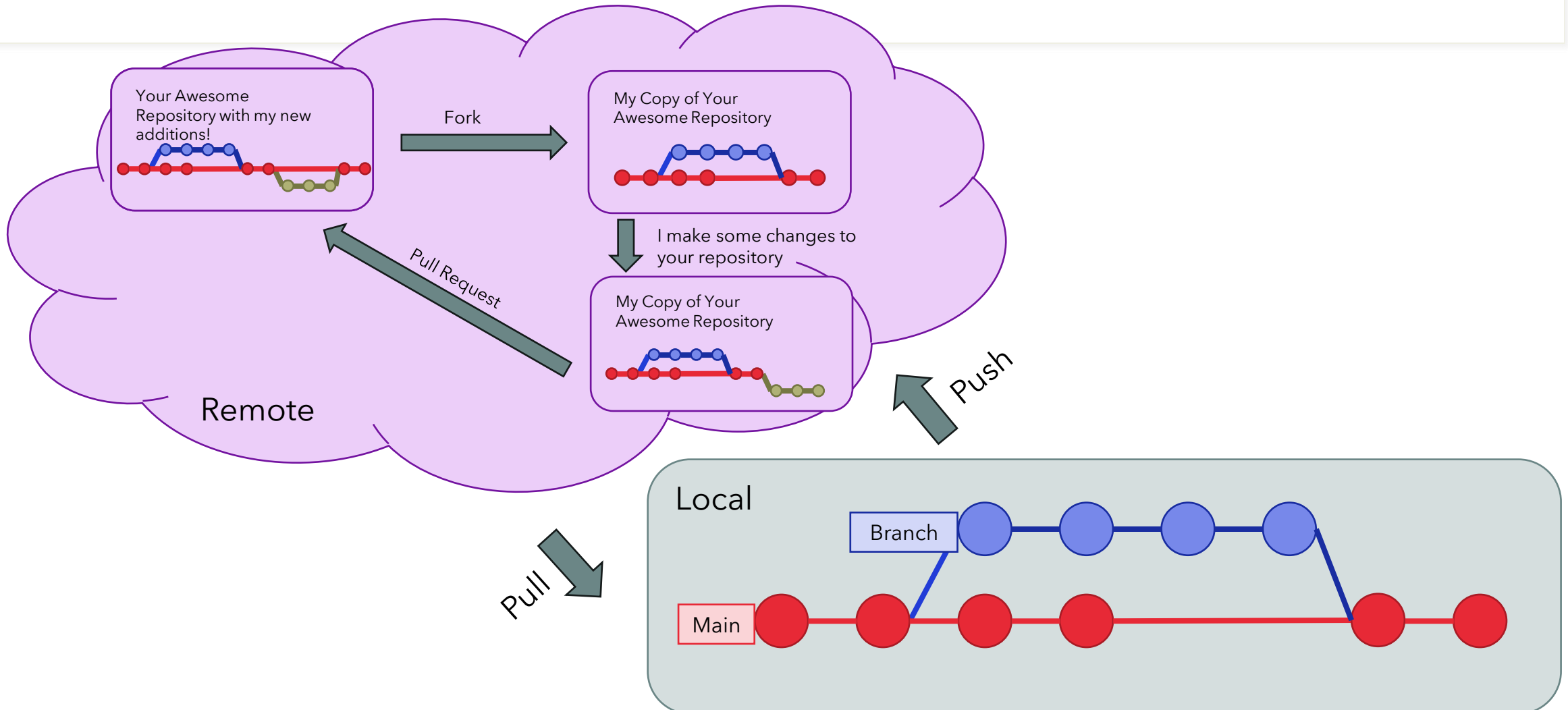
"I just pushed some changes"

"Let me pull the changes you just made so I can test them!"

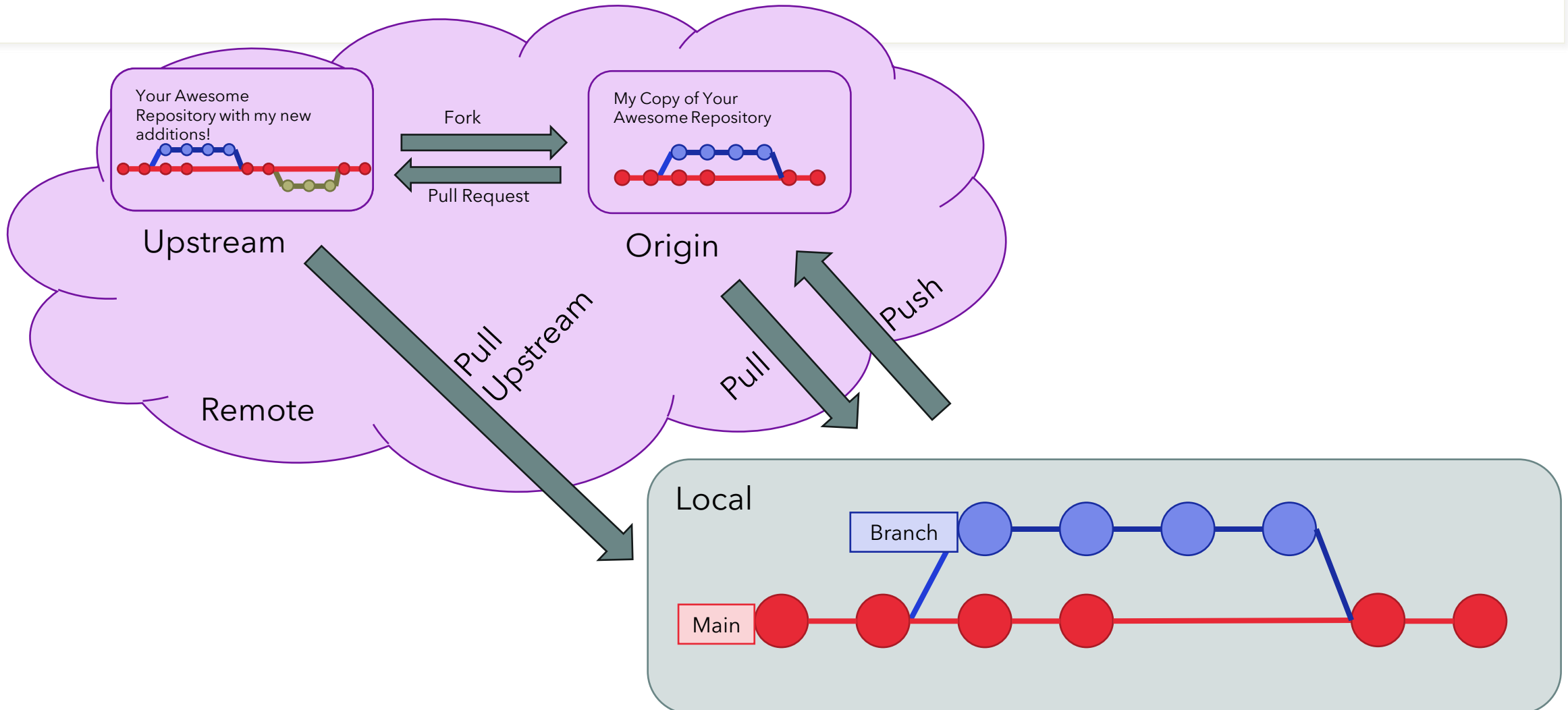
Collaborating with GitHub



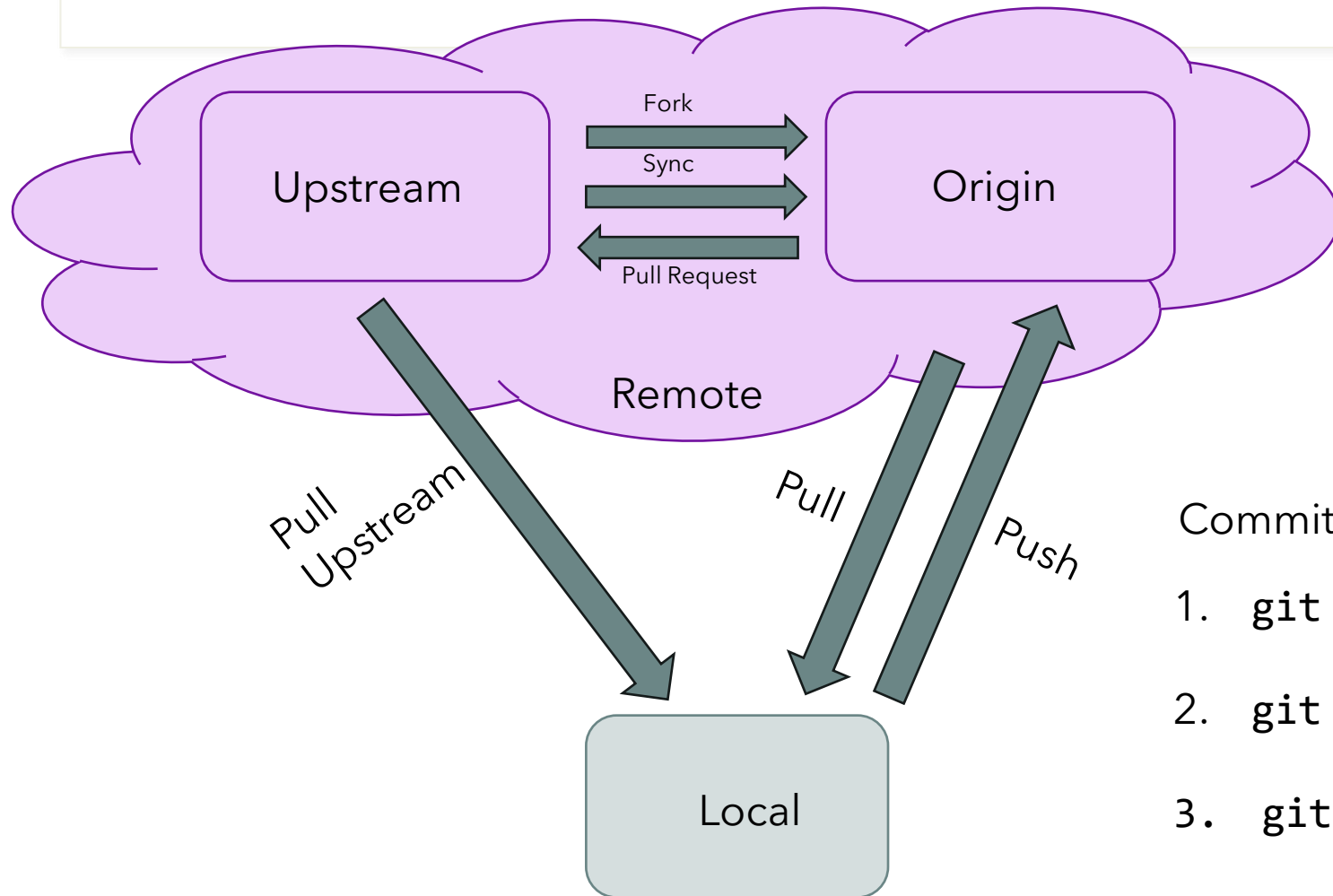
Collaborating with GitHub



Collaborating with GitHub



In Summary



Commit your changes:

1. `git status` to see current state of directory
2. `git add` files that you want to commit
3. `git commit -m "{description}"`

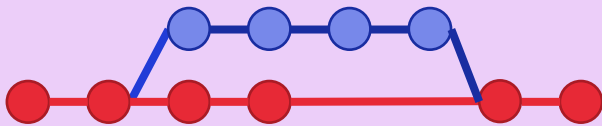


Collaborating with GitHub

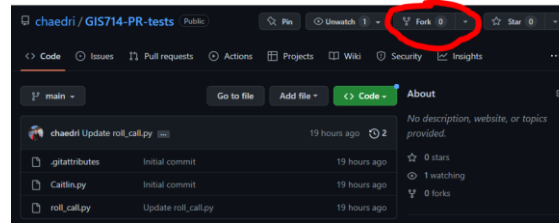
chaedri/GIS714-PR-tests

Collaborating with GitHub

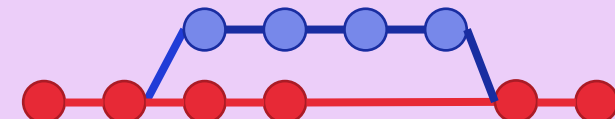
chaedri/GIS714-PR-tests



Fork

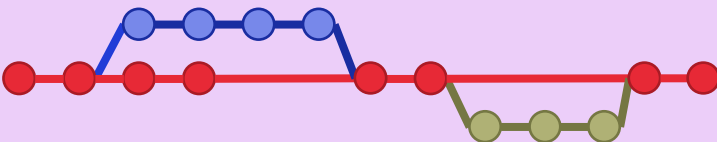


your-name/GIS714-PR-tests

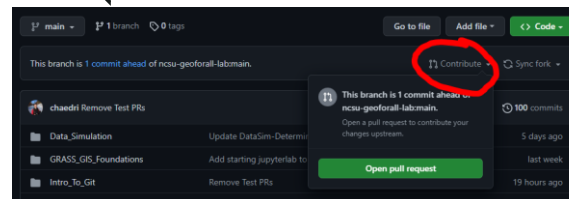


You add your file (clone, add, commit, push)

chaedri/GIS714-PR-tests
with your new additions!



Pull Request



your-name/GIS714-PR-tests

