



GitHub

Skylar Gering and Julia Sloan

Part 1

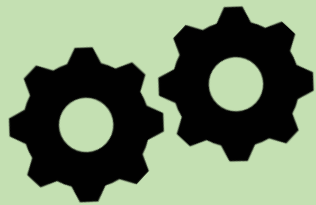
Setup Steps

1. VSCode installed
2. Julia installed
3. Git installed
4. Github account
5. Send us the email account associated with your Github

What are Git and GitHub?

Git

- Software that runs on your computer
- Version control system
 - Let's you track different versions of your code/project



GitHub

- Hosting service for Git projects
- Exclusively cloud-based
- Share your code with others



Why use Git and GitHub?

- **For yourself! (Part 1)**

- Amazing code backup
- Easy to test out new things without ruining the code you have
- Your code can be private if you're anxious about people seeing it

- **For your team! (Part 2)**

- Easy collaboration
- Easy to share code
- Made sure you don't have huge amounts of the same, slightly modified script floating around
- Reproducibility for publications

Ways to use Git and GitHub

GitHub Desktop

Focus on what matters instead of fighting with Git. Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

[Download for macOS](#)

More flexible. Can use on computers without the app.
No GUI!

Uses GUI!

Has an easy tutorial once you download.

Terminal

```
Subzero — -zsh — 80x24
Last login: Thu Nov 24 12:17:34 on ttys000
skylargering@dhcp-103-224 ~ % cd .julia/dev/Subzero
skylargering@dhcp-103-224 Subzero % git status
On branch refactor_domain
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   examples/basic_horizontal_flow.jl
    modified:   examples/example_sims.jl
    modified:   src/Subzero.jl
    modified:   src/collisions.jl
    modified:   src/coupling.jl
    modified:   src/model.jl
    modified:   src/plotting.jl
    modified:   src/simulation.jl

no changes added to commit (use "git add" and/or "git commit -a")
skylargering@dhcp-103-224 Subzero %
```

THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.

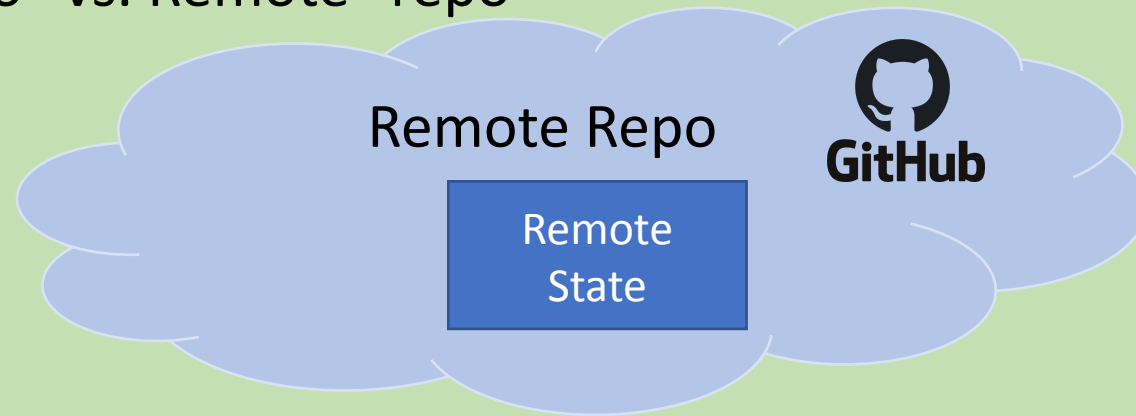


What is a repository?

- All files from the project and the history of those files
 - Local "repo" vs. Remote "repo"

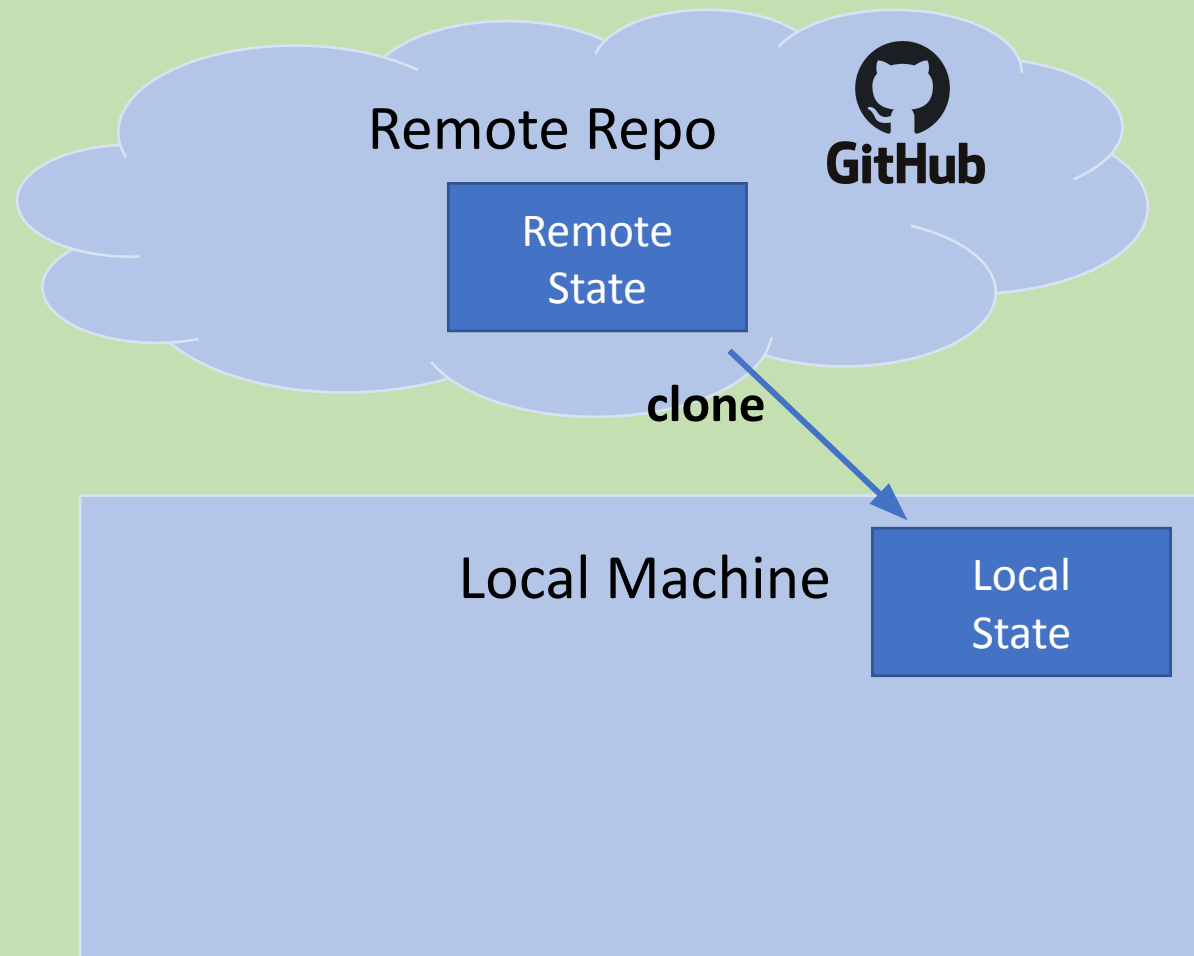
What is a repository?

- All files from the project and the history of those files
 - Local "repo" vs. Remote "repo"



1. Go to the repository link.

Clone





1. Open VSCode

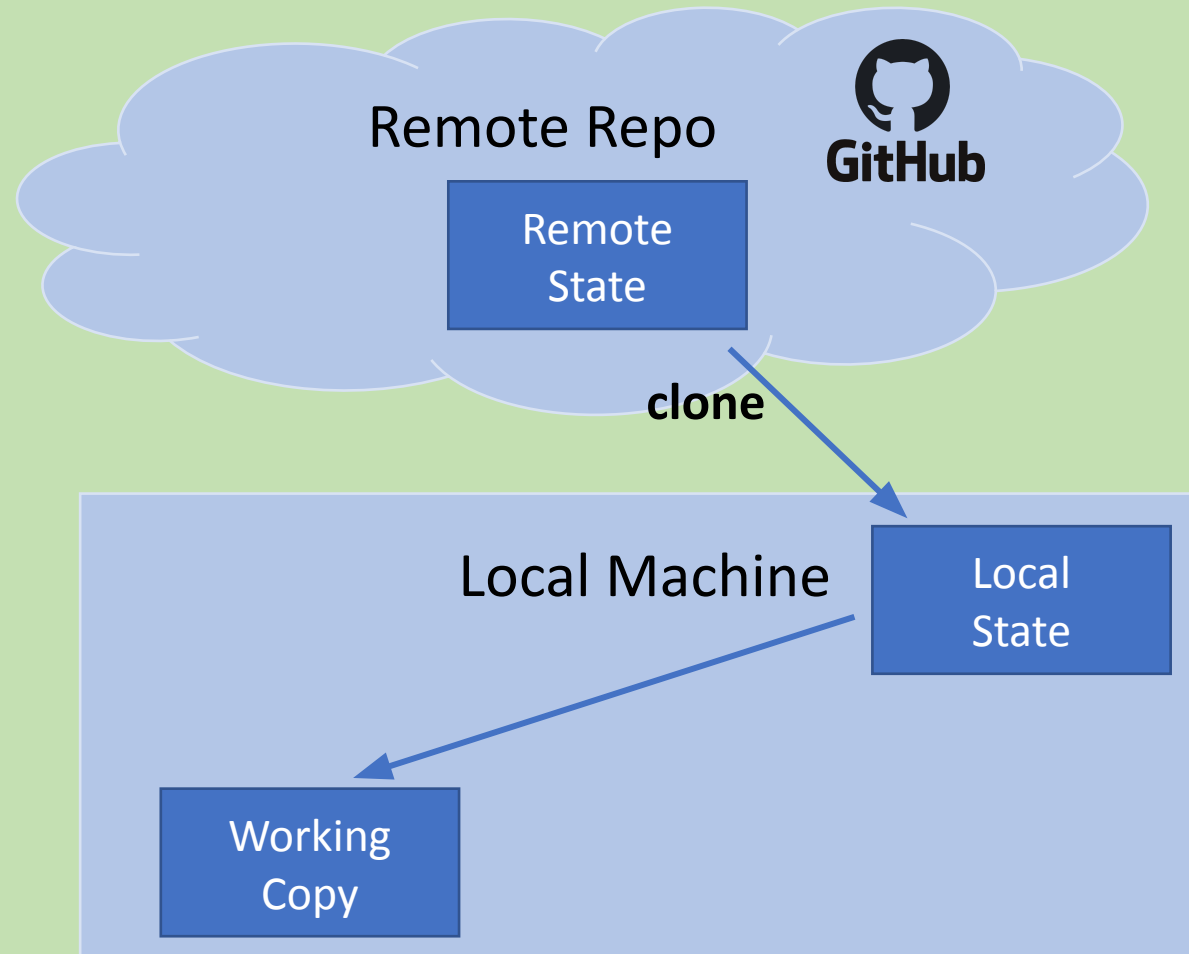
2. Open terminal in VSCode

3. In the repo, click the Code button and copy the HTTPs URL

4. Run `git clone `url`` from the terminal

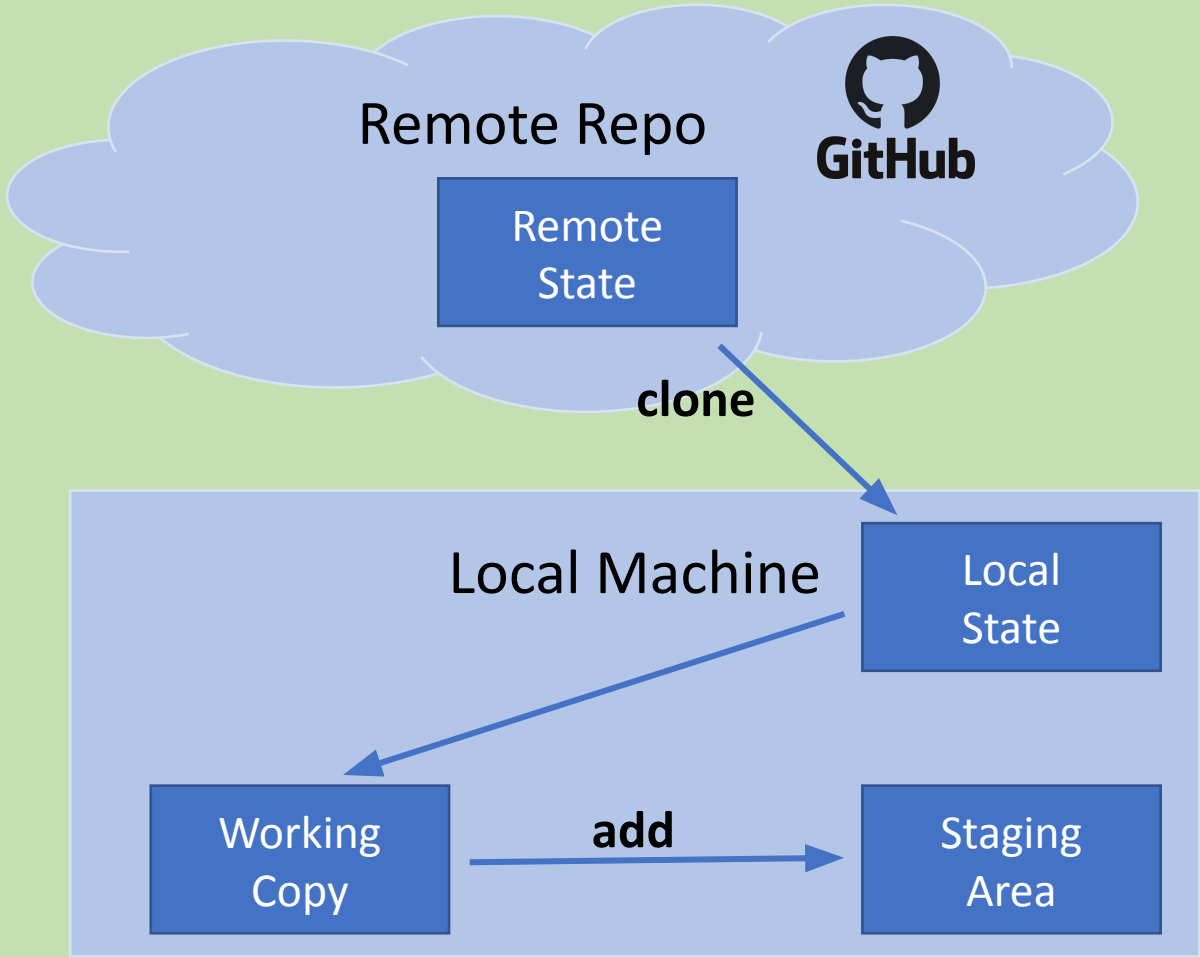
5. Run `git status` ... what do you see?

Working on a repo



1. Run `cd madlibs`
2. Make a copy of ``sample_inputs.txt``
3. Rename your file `'your_name_inputs.txt'`
4. Fill in the blanks
5. Run `git status` ... what do you see?

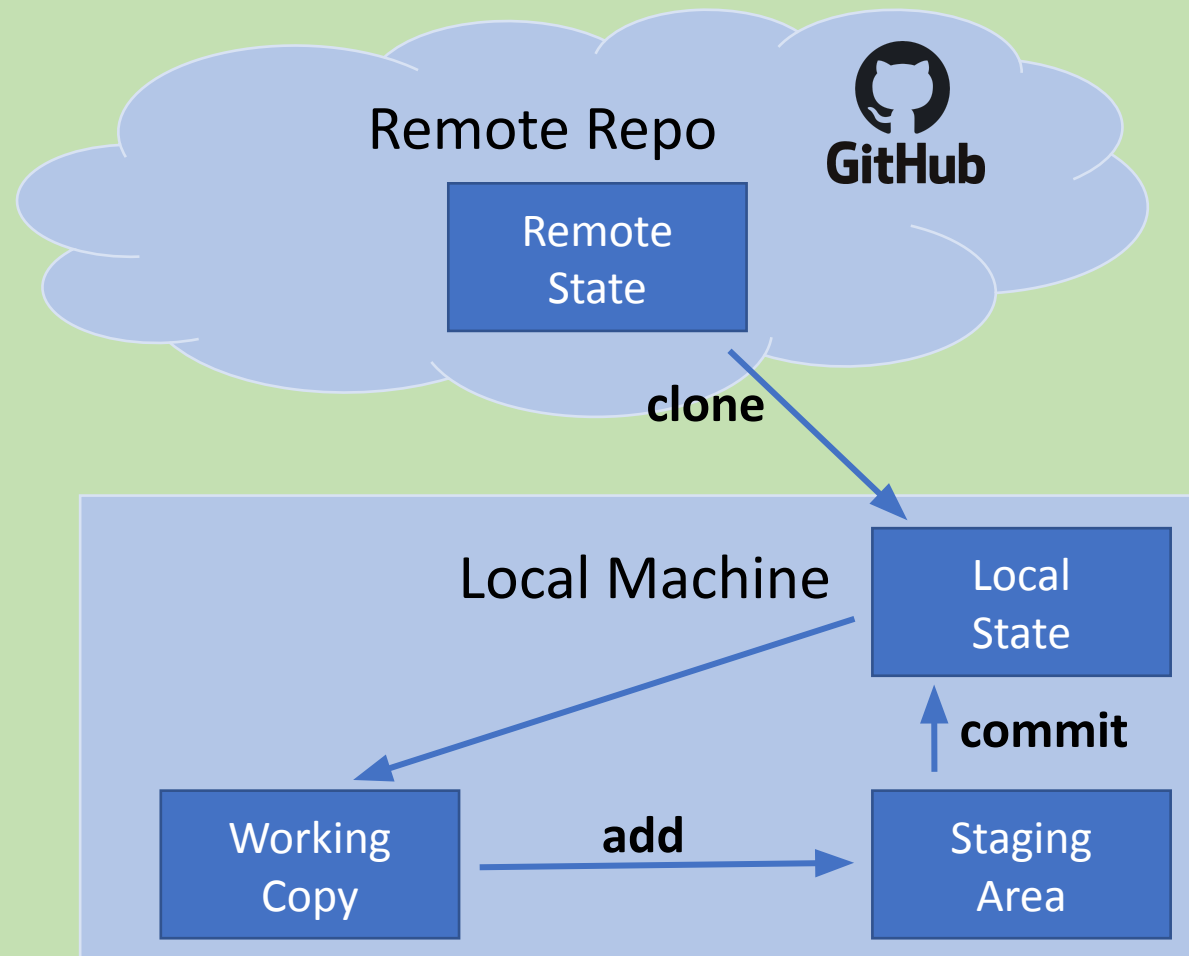
Adding Changes



1. Run `git add .`

2. Run `git status` ... what do you see?

Committing Changes



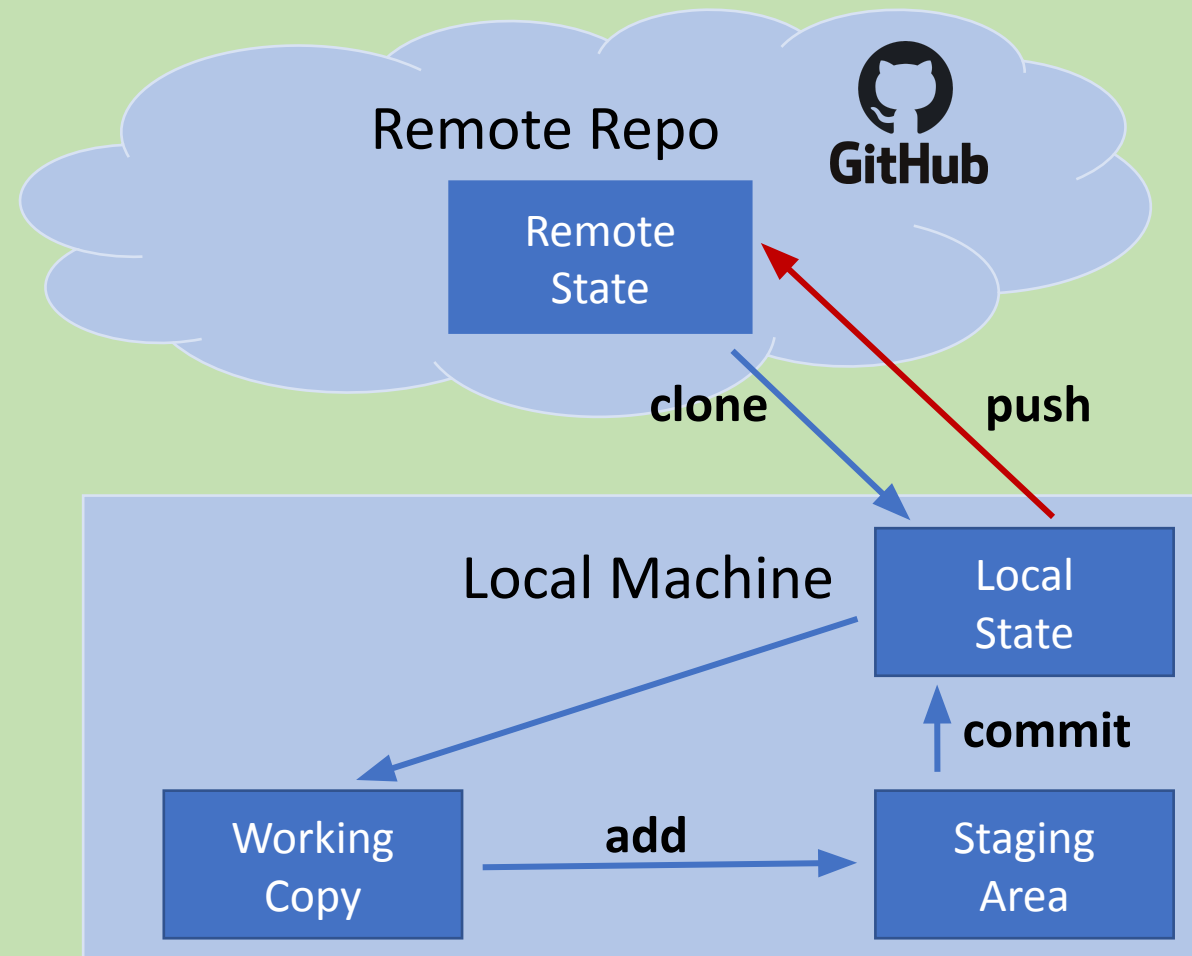
Commit Messages

Writing a good commit message:

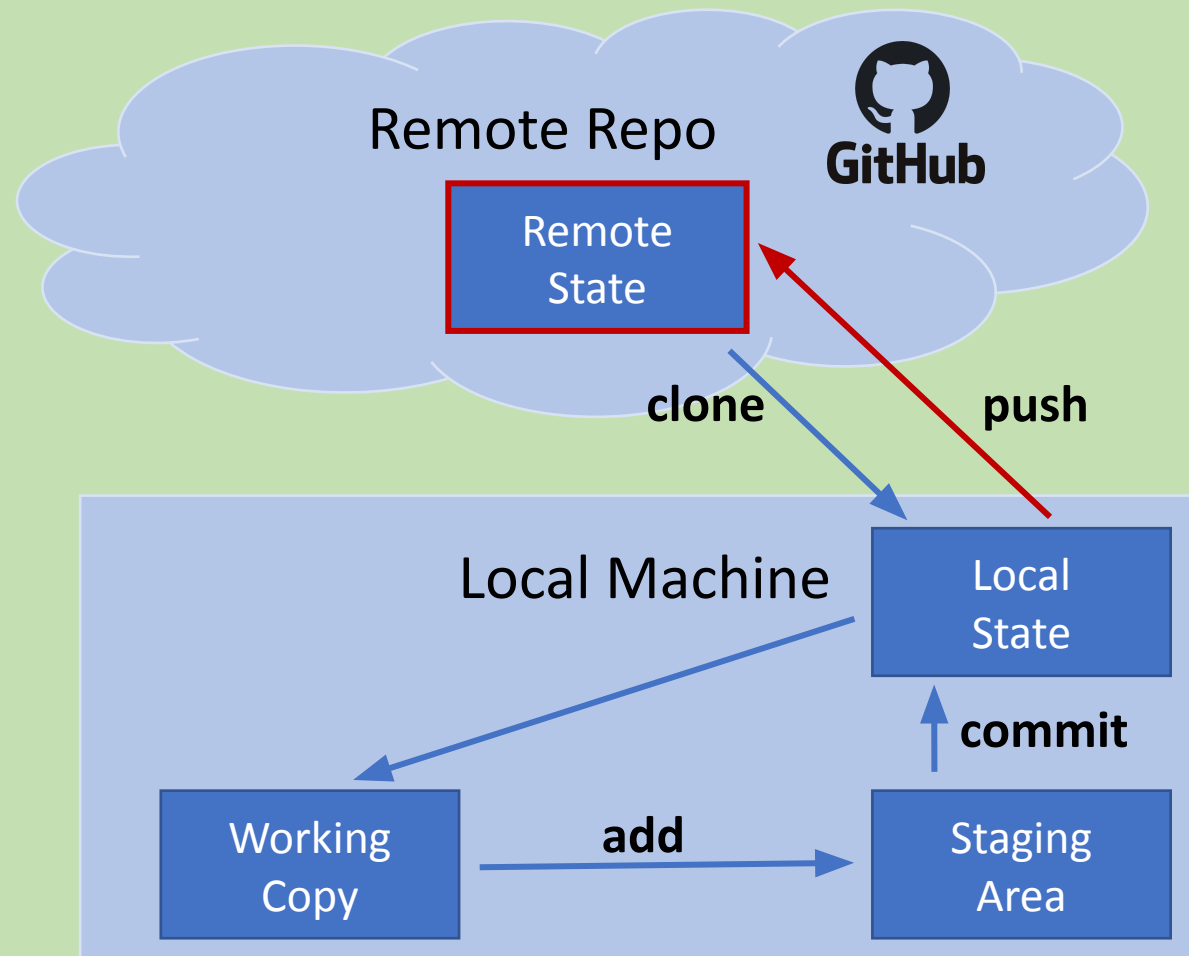
- Add code changes you want in the commit
- Write a short commit message
 - This commit will...(Fill in the blank)
 - Includes the What – Not the How
- Examples:
 - "Add normal force calculation function"
 - "Fix bug in geostrophic flow calculations"

1. Run `git commit -m "Your message here"`
2. Run `git status` ... what do you see?
3. Run `git log` ... what is there?

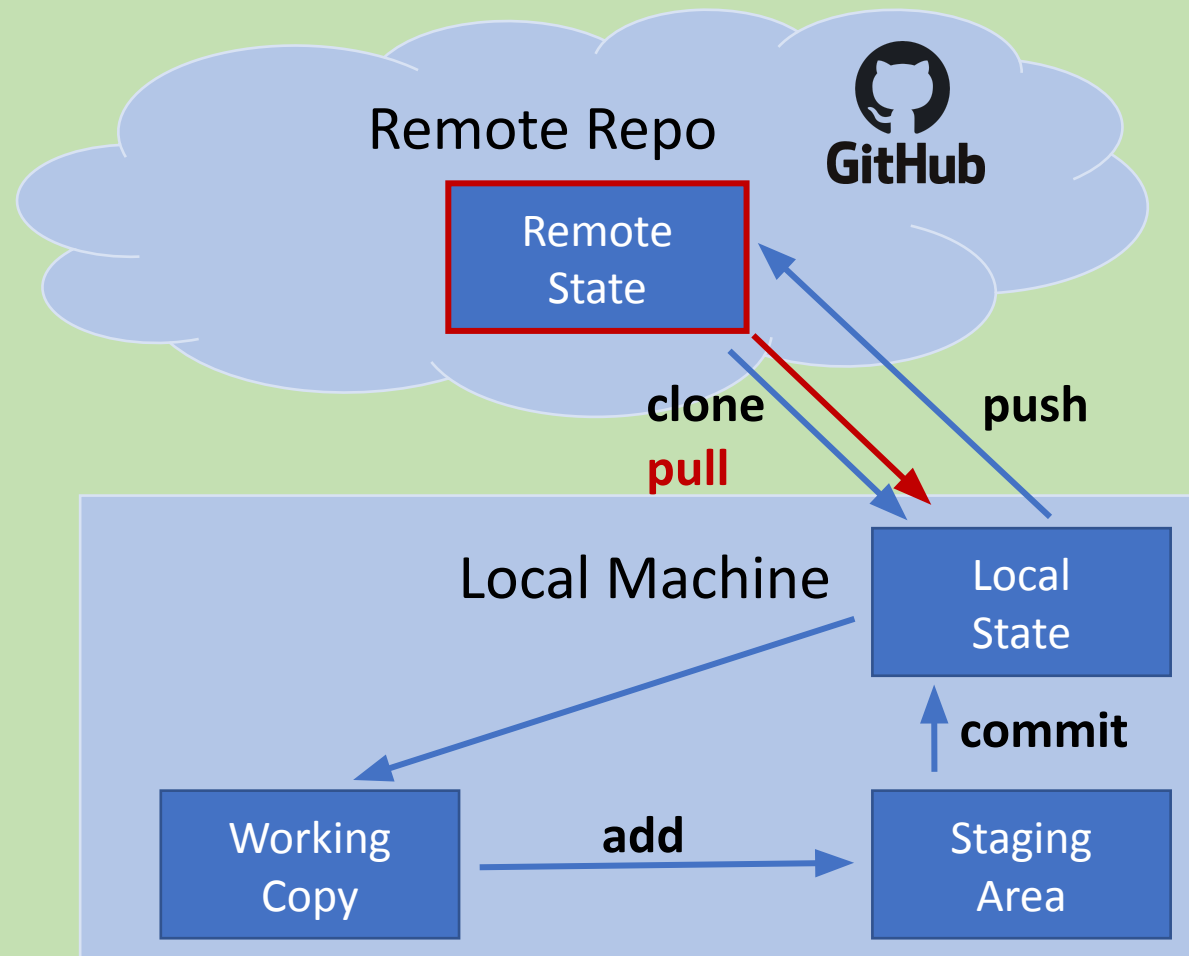
Pushing Changes



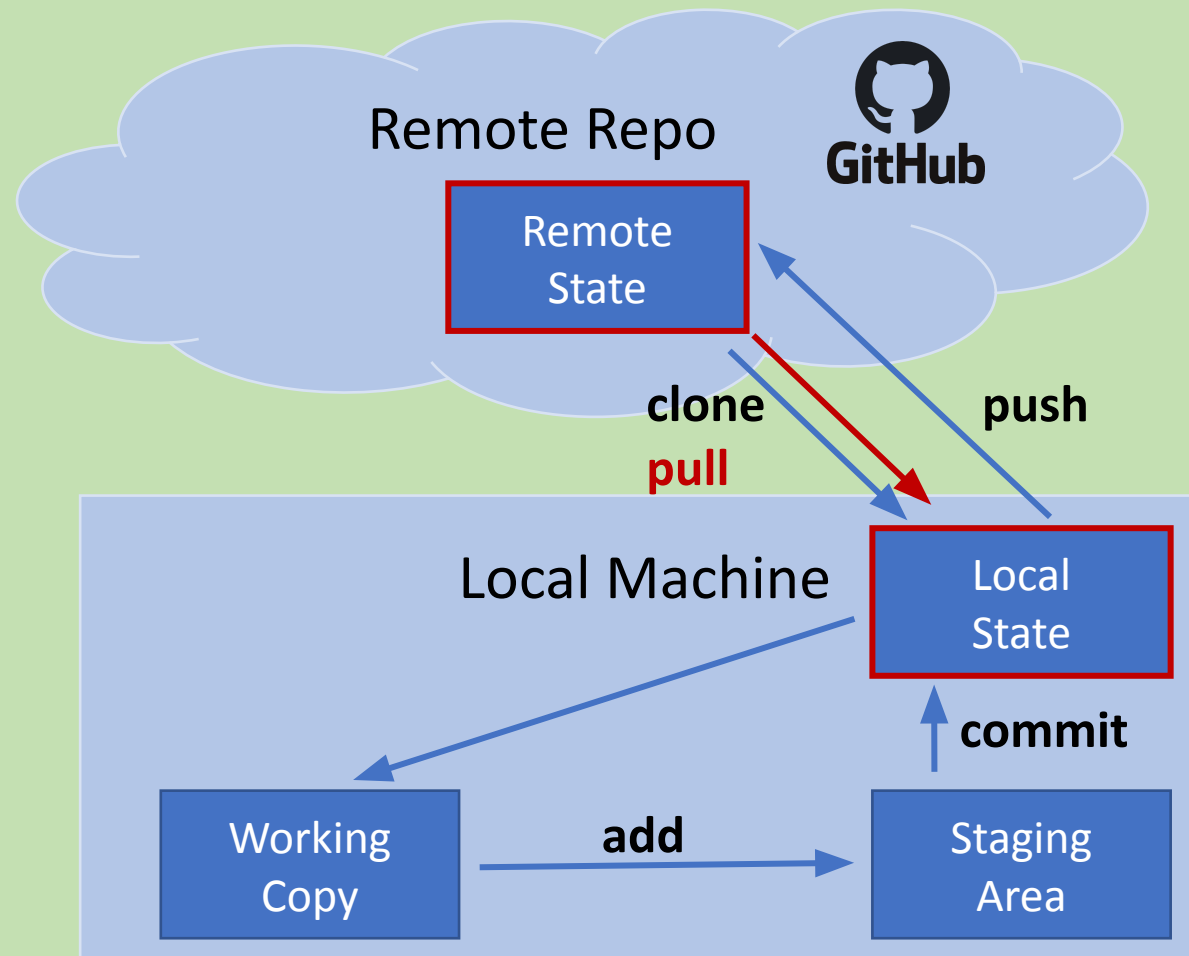
Pushing Changes



Pulling Changes



Pulling Changes



1. ONE PERSON AT A TIME

a. Run `git pull`

b. Run `git push`

2. Look at the repository online now!

1. Run `git pull`

2. What is different about your working directory?

3. Run `git log ...` what changed?

1. Run `generate_madlibs.jl`

2. Share your results :)

Git Commands Review

- `git add` – add changes to the staging area
- `git commit` – move changes from the staging area to local state
- `git push` – move changes from local state to remote state
- `git pull` – move changes from remote state to local state
- `git clone` – create local repo from remote repo
- `git status` – see what files are being tracked/have changes
- `git log` – see last few commit messages

Any questions?



GitHub