

# EPIC GitHub Tutorial (Part 2): Contributing to UFS/EPIC Repositories



<https://github.com/NOAA-EPIC/training-github>

EARTH PREDICTION INNOVATION CENTER (EPIC)



# Welcome!

- This is Part 2 of the EPIC GitHub Tutorial on Contributing to UFS/EPIC Repositories
- View Part 1: <https://epic.noaa.gov/tutorials/?playlist=afde205&video=9e65c0d>
- All the materials and tutorial presentations (Part 1 and Part 2) could be found at  
<https://github.com/NOAA-EPIC/training-github>
- Get all the materials and tutorial presentations (Part 1, Part 2):

```
git clone https://github.com/NOAA-EPIC/training-github.git
cd ./training-github
```

- The directory ./training-github/ will contain all of the materials for the tutorial.



# Welcome!

- This is the second part of the GitHub Tutorial on Contributing to UFS/EPIC Repositories.
- All the materials and tutorial presentations (Part 1 and Part 2) could be found at <https://github.com/NOAA-EPIC/training-github>
- Complete all prerequisite steps as listed in a README.md :
  - git installed (via Homebrew for Mac OS, “git-bash” for Windows)
  - basic command-line interface (CLI) commands
  - opened a GitHub account



EARTH PREDICTION INNOVATION CENTER (EPIC)



# Recap of the Part 1: Git and GitHub Basics

- Basic Git and GitHub terms and concepts introduced
- Basic Git configurations and commands discussed
- Example given on setting up a SSH key pair for GitHub authentication
- A new local repository was created, initialized, pushed to GitHub
- Modifications made, committed, local changes tracked, logs viewed
- A new branch made, different branches compared
- Started topic: public GitHub repositories, forks and clones



EARTH PREDICTION INNOVATION CENTER (EPIC)



# GitHub Tutorial Part 2: Working with Remote Repositories

- UFS/EPIC Public Repositories, Forks and Clones
- Branches and Tags
- Git Workflow: from Local Spaces to Remote Repos
- Fetching and Merging Remote Branches
- Resolve Merge conflicts
- Making Pull Requests
- GitHub Issues and Discussions



EARTH PREDICTION INNOVATION CENTER (EPIC)



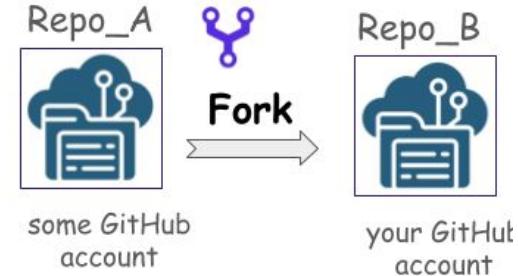
# Common GitHub repositories used by UFS community

- <https://github.com/ufs-community/ufs-weather-model>
- <https://github.com/ufs-community/ufs-srweather-app>
- [https://github.com/ufs-community/land-DA\\_workflow](https://github.com/ufs-community/land-DA_workflow)
- <https://github.com/ufs-community/uwtools>
- [https://github.com/ufs-community/UFS\\_UTILS](https://github.com/ufs-community/UFS_UTILS)
- <https://github.com/ufs-community/ccpp-physics> (forked from [NCAR/ccpp-physics](https://github.com/NCAR/ccpp-physics))
- <https://github.com/hafs-community/HAFS> (forked from [hafs-communityHAFS](https://github.com/hafs-communityHAFS))
- <https://github.com/ufs-community/global-workflow-AR> ([NOAA-EMC/global-workflow](https://github.com/NOAA-EMC/global-workflow))
- <https://github.com/NOAA-EMC/UPP>
- <https://github.com/noaa-oar-arl/NEXUS>
- <https://github.com/NOAA-EMC/AQM-utils>
- <https://github.com/JCSDA/spack-stack>



# GitHub Repositories: Forks and Clones

- **fork:** a separate copy of another existing repository
  - created in a remote location, e.g., GitHub account
  - could be synced with the primary repository
  - contribute to your fork not affecting the original repo
- **clone:** a linked local copy of an existing repository
  - a repository code is downloaded to your local machine
  - done through the command ‘git clone’
  - references to an original target/remote repository



# GitHub Repositories: Forks and Clones

This screenshot shows the GitHub fork interface for the `NOAA-EPIC/training-github` repository. The main repository page is visible, showing the README.md file. A red circle highlights the 'Fork' button in the top right corner of the repository card. A red dashed box highlights the 'Create a new fork' link in the 'Existing forks' dropdown menu.

Creating a fork of a repository, e.g., GitHub, creates a copy of the repository for your own use.

This screenshot shows the GitHub repository for `natalie-perlin/training-github`. The repository card indicates it is a fork of `NOAA-EPIC/training-github`, with a red circle highlighting this information. A large blue arrow points from the left screenshot to this forked repository card.

# GitHub Repositories: Forks and Clones

- **fork:** a separate copy of another existing repository
  - created in a remote location, e.g., GitHub account
  - could be synced with the primary repository
  - contribute to your fork not affecting the original repo
- **clone:** a linked local copy of an existing repository
  - a repository code is downloaded to your local machine
  - done through the command 'git clone'
  - references to an original target/remote repository



# GitHub Repositories: Forks and Clones

- **fork:** a separate copy of another existing repository

- **create**  
[Natalie@MacbookPro:~]\$ git clone https://github.com/NOAA-EPIC/training-github.git  
Cloning into 'training-github'...  
remote: Enumerating objects: 16, done.  
remote: Counting objects: 100% (16/16), done.  
remote: Compressing objects: 100% (13/13), done.  
remote: Total 16 (delta 3), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (16/16), 9.95 MiB | 1.52 MiB/s, done.
- **clone**  
Resolving deltas: 100% (3/3), done.

- **clone:** a copy of a repository
  - **a remote**  
[Natalie@MacbookPro:~/training-github]\$ git remote -v  
origin https://github.com/NOAA-EPIC/training-github.git (fetch)  
origin https://github.com/NOAA-EPIC/training-github.git (push)
  - **done**  
[Natalie@MacbookPro:~/training-github]\$ git branch  
\* main
  - **refer**  
[Natalie@MacbookPro:~/training-github]\$



# Contributing to an open-source GitHub repository

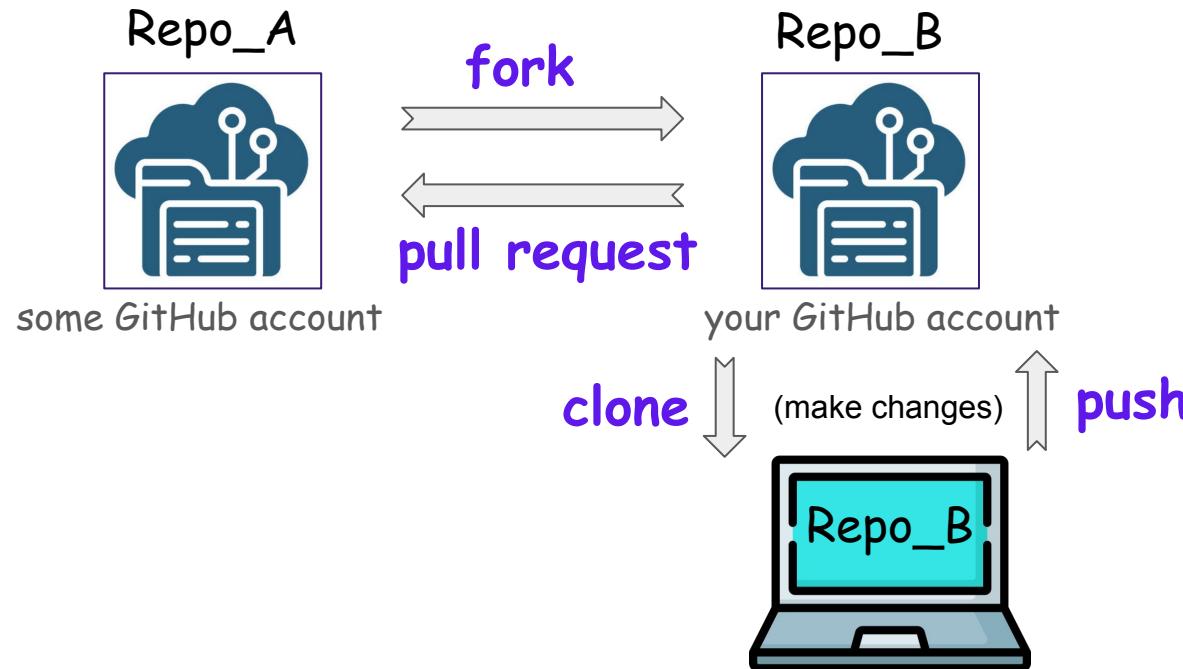
1. **FORK** the original repository to your GitHub account
2. **CLONE** your repository to a local machine
3. Make changes to your cloned repository
4. **PUSH** them to your GitHub repository: it then be synced to your changes
5. Create a **PULL REQUEST** (PR) to merge your changes with the original repository: it means making a request to the repository owner and telling them “Please check my changes and merge them if you like it”



EARTH PREDICTION INNOVATION CENTER (EPIC)



# Contributing to an open-source GitHub repository



EARTH PREDICTION INNOVATION CENTER (EPIC)



# GitHub Repositories: Branches and Tags

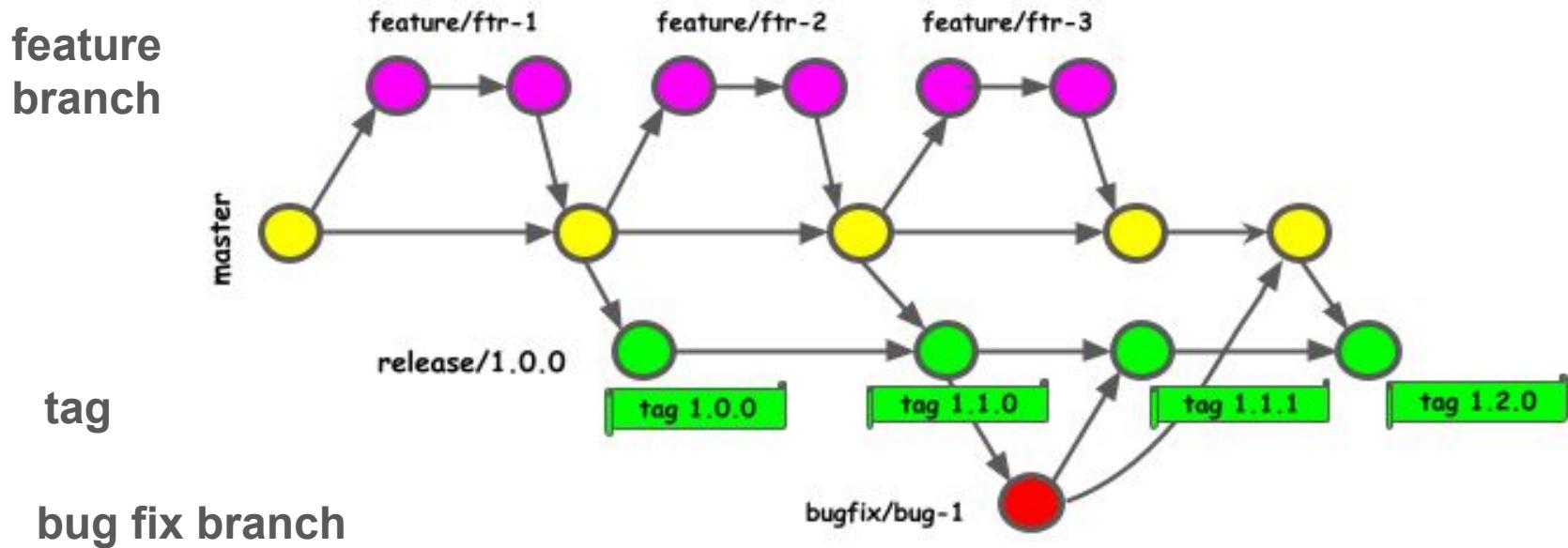
## Branches:

- used for concurrent development, bug fixes, implementation *features*
- can be moved or deleted
- often expected to be merged back into original branch
- *feature branches* could have a long span

## Tags:

- mark specific points in history to capture software versions
- typically used to mark stable points that correspond to specific releases
- *permanent* and don't change after they are created
- used to label a specific *commit*

# GitHub Repositories: Branches and Tags



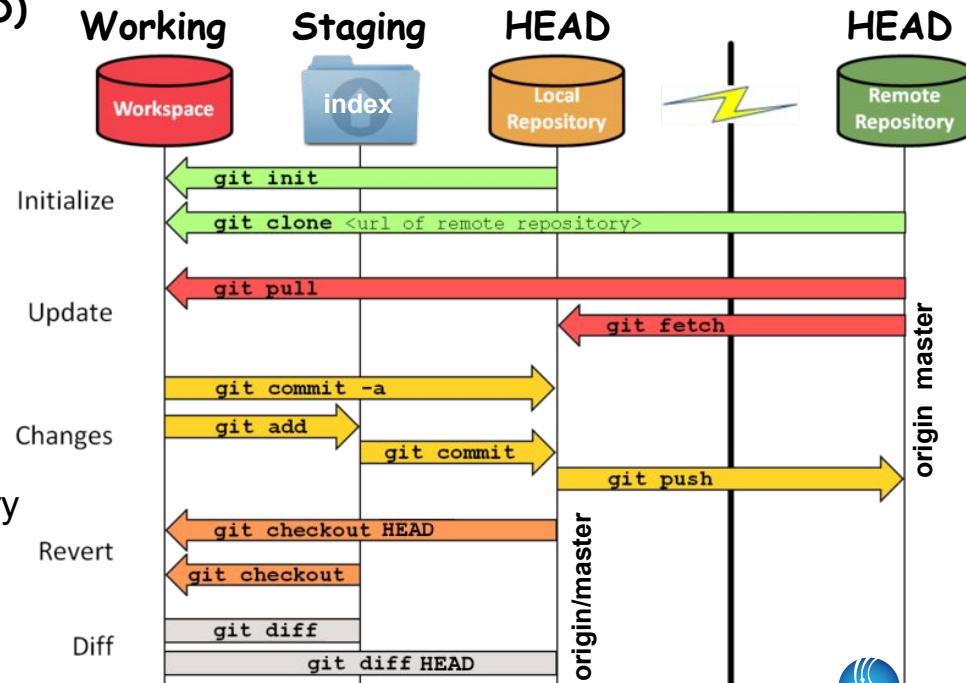
# Git workflow: from local spaces to remote repos

Local Git has three stages:

**Work area - Staging area - Local repo (HEAD)**

- Working area - your local directory
  - git clone/init
  - git checkout
  - git pull
- Staging area: what is to be committed
  - git add/mv/rm/
  - git checkout
  - git reset <file>/>commit>
- Local repository (HEAD\*):
  - default branch of the remote repository
  - fetch updates from a remote repo
  - push changes to a remote repo

\* - HEAD term also has a wider use in Git



# Git workflow: from local spaces to remote repos

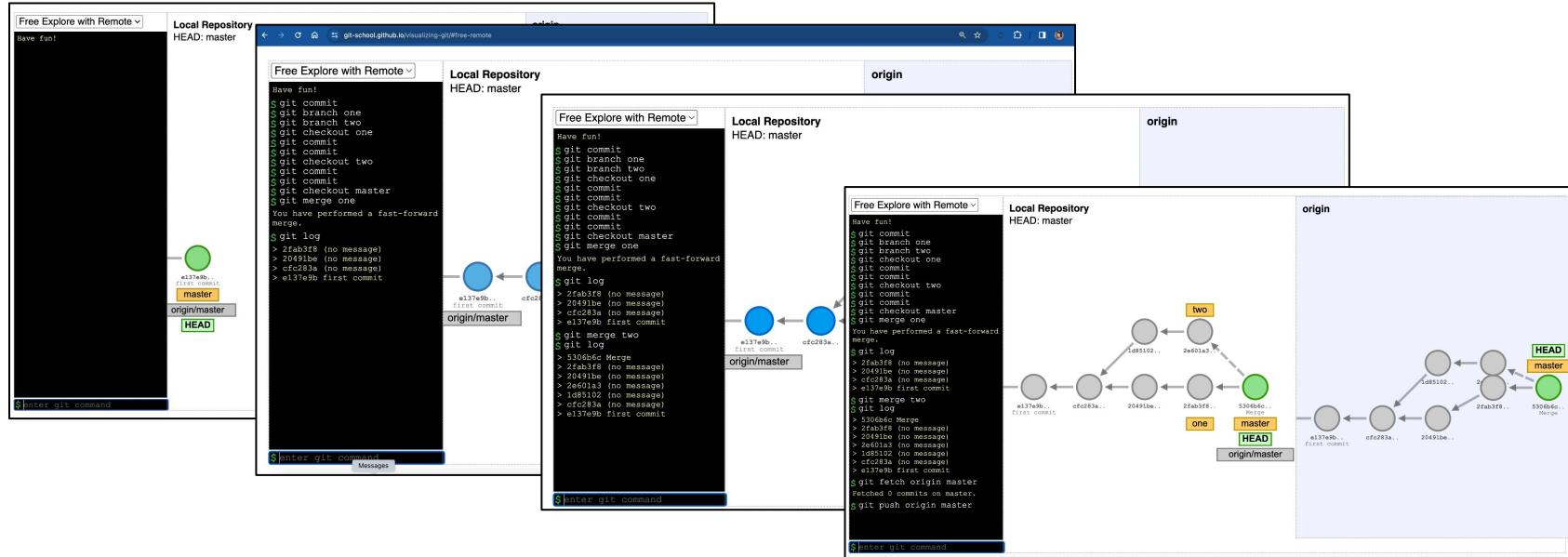
- Use an online workflow visualizer tool: <https://git-school.github.io/visualizing-git>:
  - case 1: divergent local branches, merge, update the remote repo
  - case 2: divergent local branches, detached HEAD, rebase, merge, bring upstream branch changes, update the remote repo
- Use an existing authoritative repository to test real-case scenarios
  - fetch and merge remote branches, update a fork repository
  - introduce your changes, merge, resolve merge conflicts
  - make a pull request (PR) into the authoritative repository
- Test out a PR, give feedback, make a PR into another developer's code



EARTH PREDICTION INNOVATION CENTER (EPIC)

# Visualization cases: case 1 (<https://git-school.github.io/visualizing-git>)

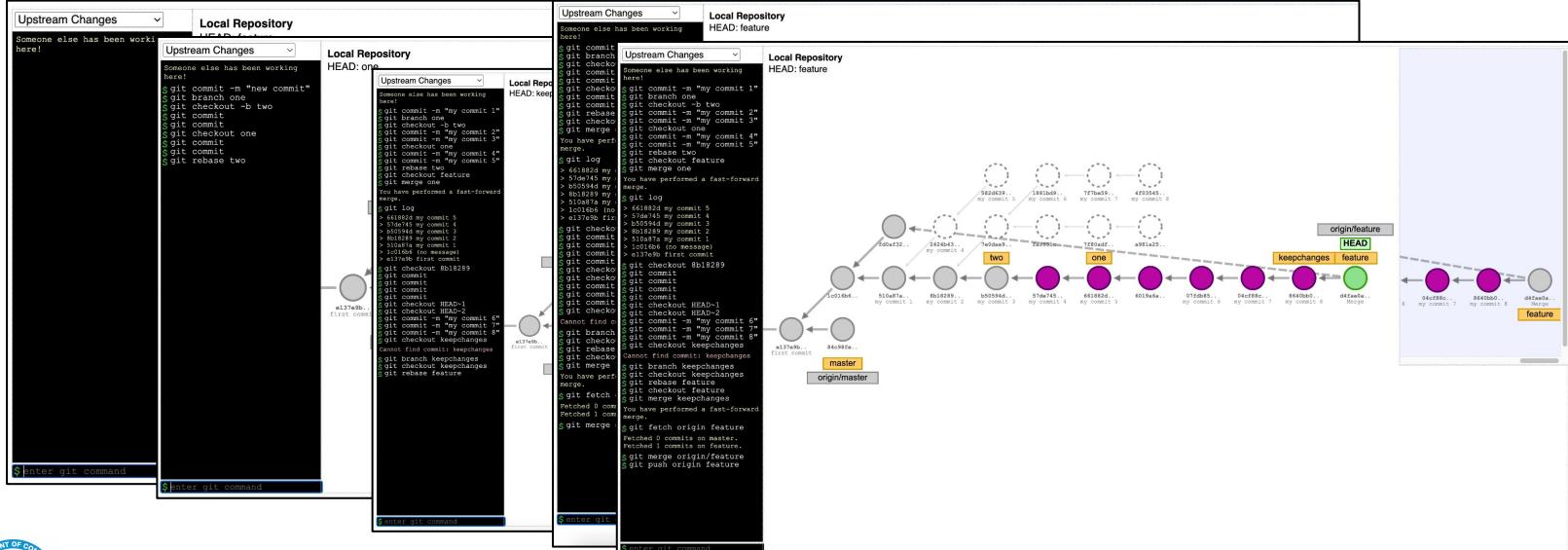
case 1: divergent local branches, merge, update the remote repo



EARTH PREDICTION INNOVATION CENTER (EPIC)

## Visualization cases: case 2 (<https://git-school.github.io/visualizing-git>)

case 2: divergent local branches, detached HEAD, rebase, merge, bring upstream branch changes, update the remote repo



# General steps to contribute to remote repos

1. Create a fork of the authoritative repo in your GitHub account (dashboard)
2. Clone the repository into your local machine

```
git clone <url-of-the-repo-fork>
```

3. Add an upstream authoritative repository for the reference and updates

```
git remote add upstream <url-of-the-authoritative-repo>
```

4. Checkout your repository's main branch: git checkout master")

5. Fetch any changes from the authoritative repository:

```
git fetch upstream
```

6. Merge in the changes to the master branch into your working reference

```
git merge upstream/master
```

7. Push your changes back to GitHub: git push origin master

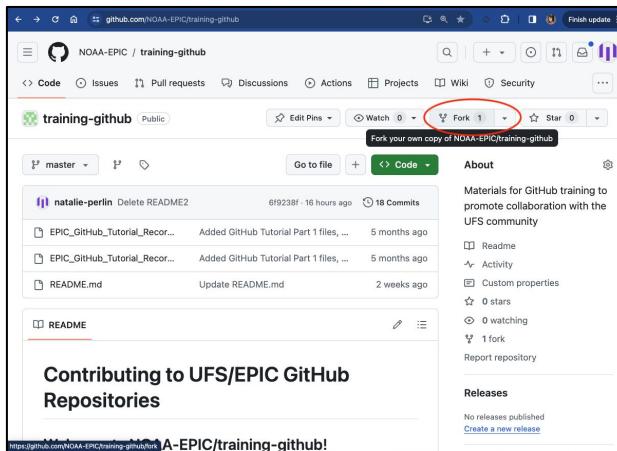
8. Create a pull request (PR) into the authoritative repository (dashboard)



# Git workflow example with an existing authoritative repo

An existing authoritative repository: <https://github.com/NOAA-EPIC/training-github>

- o fetch and merge remote branches, update a fork repository



```
git clone git@github.com:<repo-fork>
git remote add upstream <authoritative-repo>
git remote -v
git checkout master
git fetch upstream
git merge upstream/master
git log --oneline
git push origin master
```

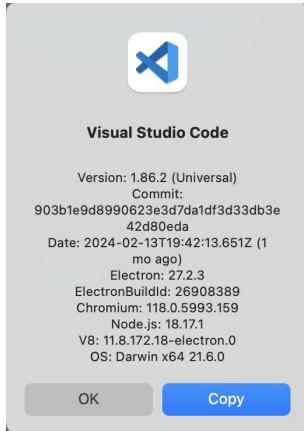


EARTH PREDICTION INNOVATION CENTER (EPIC)



# Git workflow example with an existing authoritative repo

We could use a GUI code editor, such as Visual Studio Code

A screenshot of the Visual Studio Code interface. The title bar says "training-github". The left sidebar shows the file tree with "EXPLORER", "OPEN EDITORS", and a folder named "TRAINING-GITHUB" containing files like "README2", "EPIC\_GitHub\_Tutorial\_Re...", "README.md", and another "README2". The main right pane shows the content of "README2" with several commit messages. Below the code editor, the "TERMINAL" tab is active, showing a command-line session with the following output:

```
91f65b9 (HEAD -> feature, origin/feature) merged with upstream/master
762ef34 (upstream/master) Create README2
fb61b36 (one) Merge branch 'two' into one
42b2f4d fifth commit
eefd59d fourth commit
d5735b5 (two) third commit
d3c850d second commit
175c441 first commit
4cba347 (origin/master, origin/HEAD, master) Update README.md
daaa25d Added GitHub Tutorial Part 1 files, pdf and pptx
27d98dd Update README.md
68512ca Update README.md
dc91e01 Initial commit
[Natalie@MacbookPro:~/GIT/training-github]$
```

The status bar at the bottom shows "Cloud Code - Sign in", "Ln 5, Col 16", "Spaces: 4", "UTF-8", "LF", "Markdown", and a few other icons.

EARTH PREDICTION INNOVATION CENTER (EPIC)

## Git workflow example 2 with an existing authoritative repo

- An existing authoritative repository: <https://github.com/NOAA-EPIC/training-github>
  - introduce your changes, merge, resolve merge conflicts

```
git clone git@github.com:<repo-fork>
git remote add upstream <authoritative-repo>
git remote -v
git checkout master
git checkout -b feature
# create a file README2, line: # first commit
git add README2
git commit -m "first commit"
git branch one
git checkout -b two
```



EARTH PREDICTION INNOVATION CENTER (EPIC)



## Git workflow example 2 (continued)

- An existing authoritative repository: <https://github.com/NOAA-EPIC/training-github>
  - introduce your changes, merge, resolve merge conflicts

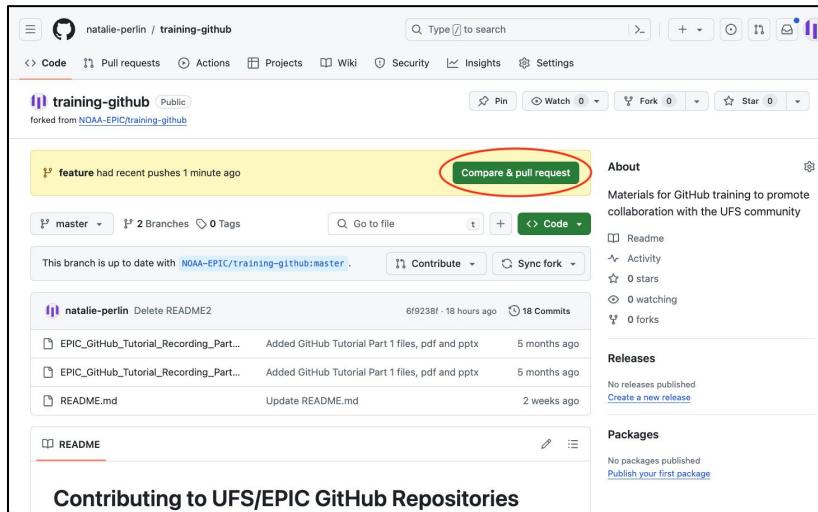
```
git branch  
# Add a line to README2 file: # second commit  
git commit -am "second commit"  
# Add a line to README2 file: # third commit  
git commit -am "third commit"  
git checkout one  
# Add a line to README2 file: # fourth commit  
git commit -am "fourth commit"  
# Add a line to README2 file: # fifth commit  
git commit -am "fifth commit"  
git branch  
git merge two
```

```
# solve merge conflicts using a text editor!!  
git commit -am "merge with two"  
git branch  
git checkout feature  
git merge one  
git log --oneline  
git fetch upstream master  
git merge upstream/master  
git checkout upstream/master  
git commit -am "merge with upstream"  
git push origin feature
```

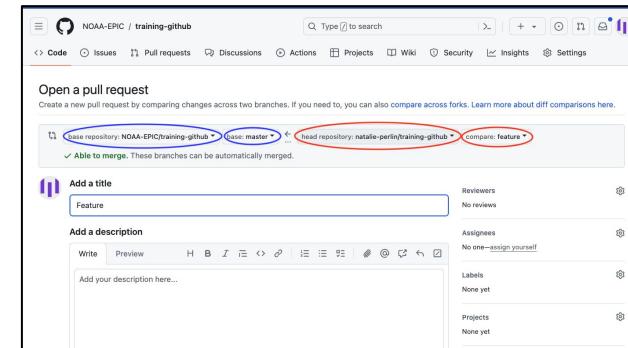


# Git workflow example: submit a pull request

- An existing authoritative repository: <https://github.com/NOAA-EPIC/training-github>
  - make a pull request (PR) into the authoritative repository (\*)



This screenshot shows the GitHub interface for the repository 'natalie-perlin / training-github'. The main header bar includes links for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the header, the repository name 'training-github' is shown as public, forked from 'NOAA-EPIC/training-github'. A yellow banner at the top indicates 'feature had recent pushes 1 minute ago'. A prominent green button labeled 'Compare & pull request' is circled in red. The repository structure shows branches 'master' and 'feature', and several commits from 'natalie-perlin'. The 'About' section describes the repository as materials for GitHub training to promote collaboration with the UFS community. The 'Contributing' section links to 'Contributing to UFS/EPIC GitHub Repositories'.



This screenshot shows the 'Open a pull request' dialog box. It displays the 'base repository: NOAA-EPIC/training-github' and 'head repository: natalie-perlin/training-github' fields, both of which are circled in red. The dialog box also shows fields for 'Add a title' (set to 'Feature'), 'Add a description' (with a rich text editor), and various review and project settings on the right side.

(\*) - you may need to address any comments, resolve conversations, complete requested changes by PR Reviewers!



## Git workflow example (3): testing another user's pull request

- Test out a PR, give feedback, make a PR into another developer's code

```
git clone git@github.com:ufs-community/ufs-srweather-app.git
cd ufs-srweather-app
git pull origin pull/<PR#>/head:PRtest
git checkout PRtest
# test the PR, build, run the code, comment on the PR on specific
# issues, files, or lines, i.e.:
# Make your own PR into original PR author's repository:
# get author's repository location and branch from the PR's page, i.e:
# https://github.com/ufs-community/ufs-srweather-app/pull/1005
```

[develop] Streamline SRW App's interface to  
MET/METplus #1005

Open

gsketefian wants to merge 105 commits into [ufs-community:develop](#) from [gsketefian:feature/metplus\\_conf\\_templates](#)

Edit

< > Code ▾

# Perusing Public Repositories

- Check out a few different repositories with submodules:

```
git clone --recursive <url-repo>
```

```
git clone --recurse-submodules <url-repo>
```

```
git clone <url-repo>
```

```
cd <repo>
```

```
git submodule update -remote
```

```
git clone --recurse-submodules https://github.com/jcsda/spack-stack.git
```

```
git clone --recursive https://github.com/ufs-community/ufs-weather-model.git
```

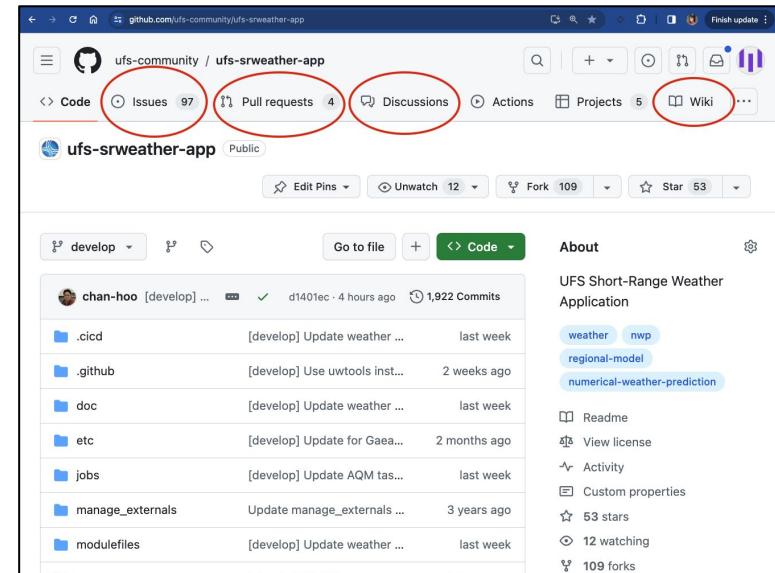


EARTH PREDICTION INNOVATION CENTER (EPIC)



# Perusing Public Repositories

- Submitting issues - “**Issues**”
- Reaching out for help through discussions - “**Discussions**”
- Checking out and testing existing PRs - “**Pull Requests**”
- Navigating documentation - “**Wiki**”, look for links to **readthedocs.io**, such as  
<https://ufs-weather-model.readthedocs.io/>  
<https://ufs-srweather-app.readthedocs.io/>



# Perusing Public Repositories

The image shows three screenshots of a GitHub repository interface for the `ufs-community / ufs-srweather-app` repository.

- Screenshot 1:** Shows the Issues page with 97 open issues. A search bar at the top has the query `is:issue is:open`. On the left, a sidebar lists several pull requests with green checkmarks, such as `[SRW-AQM] Need to`, `[SRW-AQM] Need to`, and `[develop] Port SRW-AQM to C`.
- Screenshot 2:** Shows the Pull requests page with 4 open pull requests. A search bar at the top has the query `is:pr is:open`. A red circle highlights the "New discussion" button in the top right corner of the pull request list.
- Screenshot 3:** Shows the Home page of the repository. It features a prominent announcement banner: "Announcement: UFS Short-Range Weather Application 10/31/2023". Below the banner, there's a "Welcome to the UFS Short-Range Weather (SRW) Application Wiki!" section. This section includes a paragraph about the SRW App v2.2.0 release, links to the User's Guide and NOAA EPIC website, and a detailed description of the repository's purpose and the forecast model used. To the right, there's a sidebar with navigation links like "Home", "Supported Platforms and Compilers", "Getting Started", "Releases", and "Announcements".

EARTH PREDICTION INNOVATION CENTER (EPIC)

