

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



Natalie Perlin, NOAA/EPIC, natalie.perlin@noaa.gov

<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.



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)



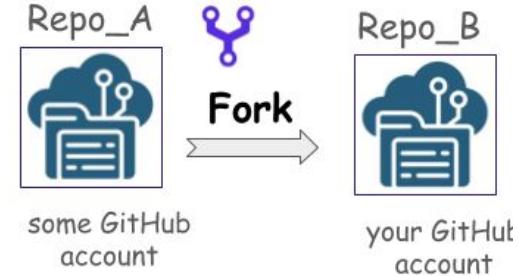
Common GitHub repositories used by UFS community

- <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/uwtools>
- https://github.com/ufs-community/UFS_UTILS
- [https://github.com/ufs-community/ccpp-physics \(forked from NCAR/ccpp-physics\)](https://github.com/ufs-community/ccpp-physics)
- [https://github.com/hafs-community/HAFS \(forked from hafs-communityHAFS\)](https://github.com/hafs-community/HAFS)
- [https://github.com/ufs-community/global-workflow-AR \(NOAA-EMC/global-workflow\)](https://github.com/ufs-community/global-workflow-AR)
- <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



EARTH PREDICTION INNOVATION CENTER (EPIC)

GitHub Repositories: Forks and Clones

A screenshot of a web browser showing the GitHub fork interface for the repository `NOAA-EPIC/training-github`. The URL in the address bar is `github.com/NOAA-EPIC/training-github`. The main repository page is visible, showing the README.md file. A modal window titled "Existing forks" is open, displaying the message "You don't have any forks of this repository." with a red circle around the "Fork 0" button. Below this button is a dashed red circle around the "+ Create a new fork" link.

A screenshot of a web browser showing the GitHub repository for `natalie-perlin/training-github`. The URL in the address bar is `github.com/natalie-perlin/training-github`. The repository page shows the README.md file. A red circle highlights the "training-github" repository card, which includes the text "forked from `NOAA-EPIC/training-github`". A large blue arrow points from the left screenshot towards this forked repository card.

EARTH PREDICTION INNOVATION CENTER (EPIC)



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



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 will 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”



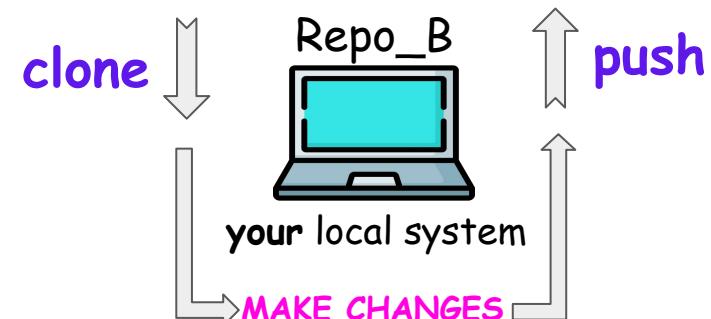
EARTH PREDICTION INNOVATION CENTER (EPIC)



Contributing to an open-source GitHub repository



pull request
←————→
fork



EARTH PREDICTION INNOVATION CENTER (EPIC)

GitHub Repositories: Branches and Tags

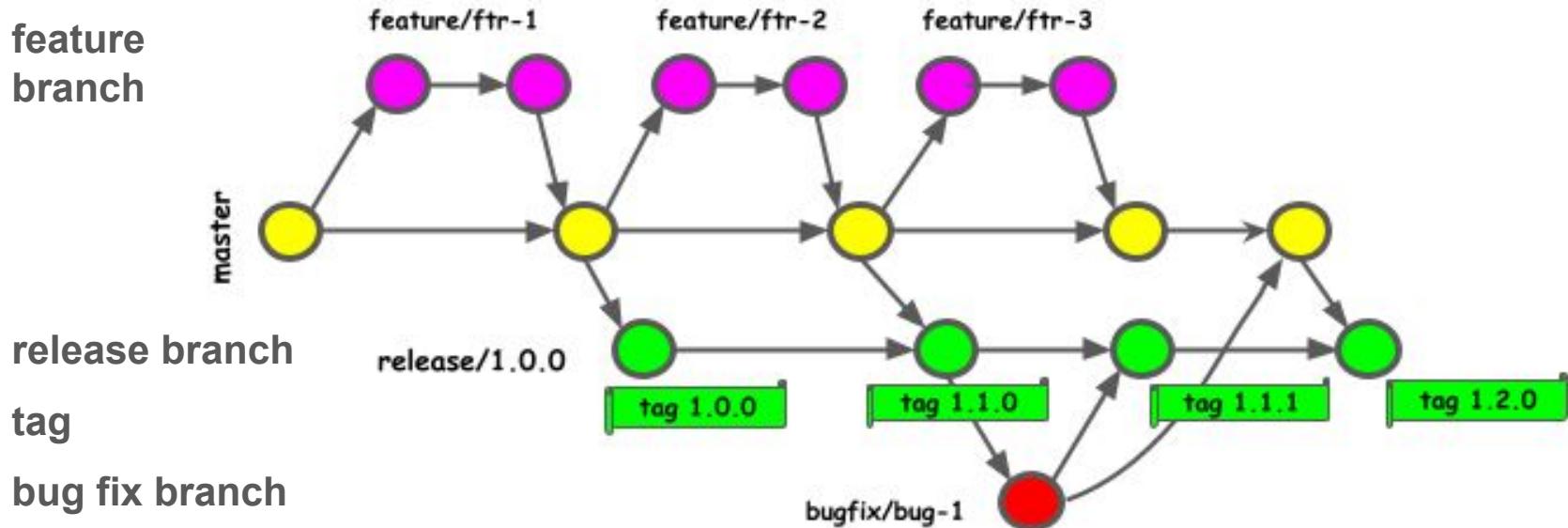
Branches:

- 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

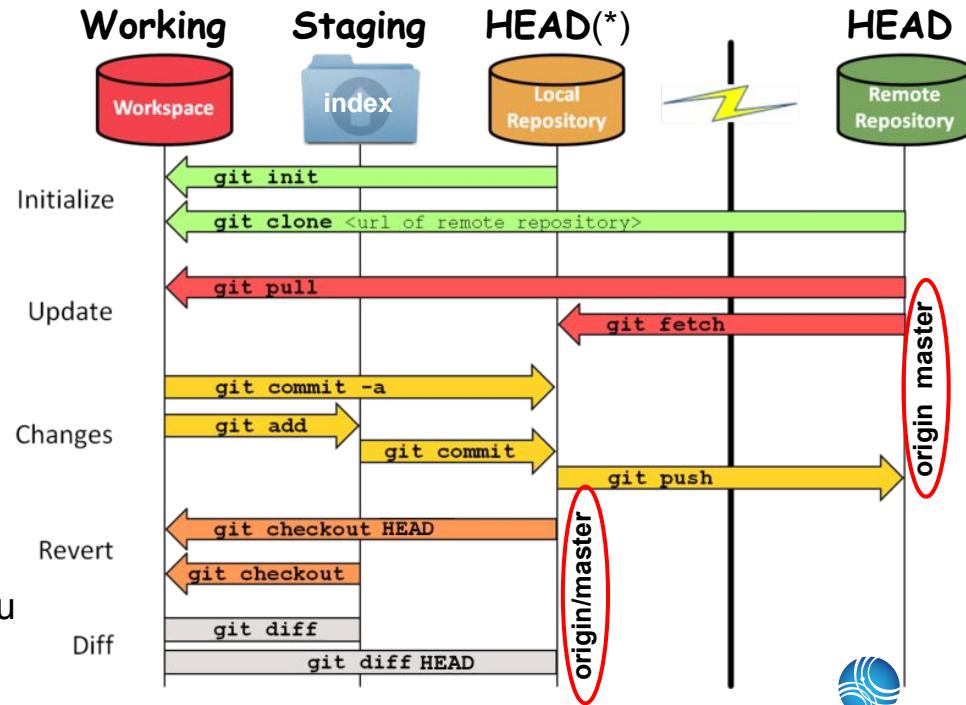


Git workflow: from local spaces to remote repos

Local Git has **three** stages:

- **Working area** -
your local directory,
all untracked, unstaged,
locally deleted files
- **Staging area:**
what needs to be committed, what
does not need to be committed,
any new files to be added/removed
- **Local repository (HEAD*):**
committed changes,
latest checked out version of the
remote repository

(*) HEAD - symbolic reference to the branch you
are working on, or rather, most recent commit



Git workflow: from local spaces to remote repos

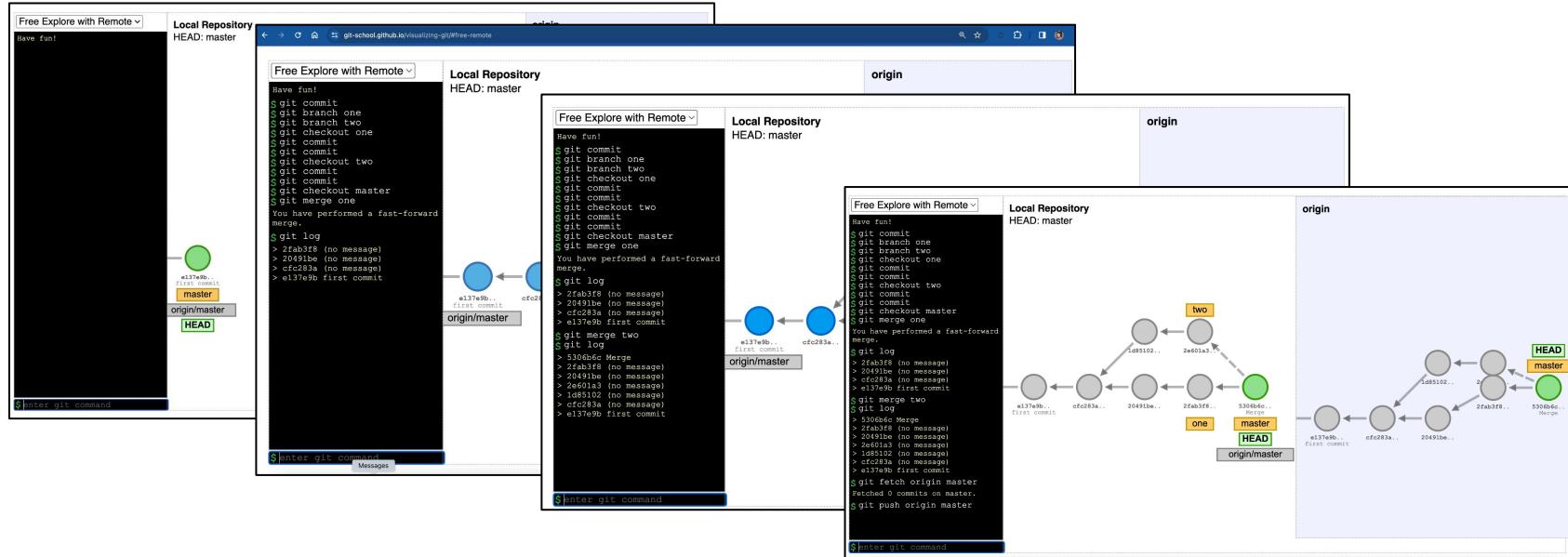
- Use an online workflow visualizer tool: <https://git-school.github.io/visualizing-git>:
 - **case 1:** divergent local branches, merge, update the remote repo
 - **case 2:** divergent local branches, detached HEAD, rebase, merge, bring upstream branch changes, update the remote repo
- Use an existing 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>)

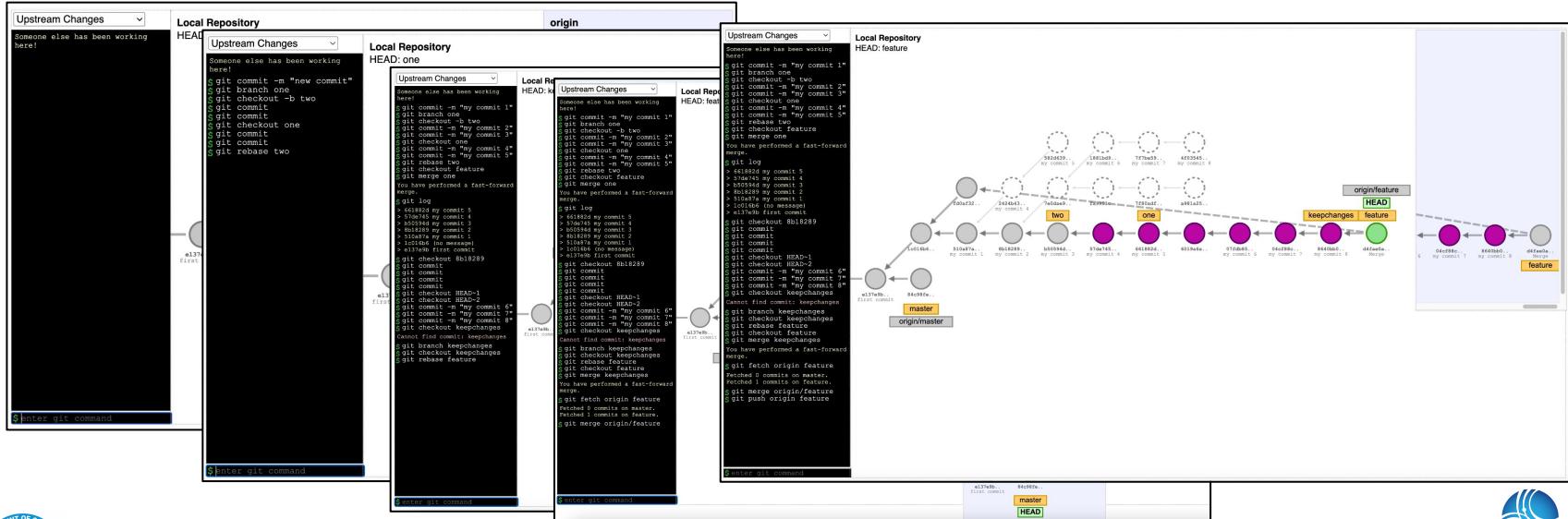
case 1: divergent local branches, merge, update the remote repo



EARTH PREDICTION INNOVATION CENTER (EPIC)

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

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



Steps to contribute to public remote repositories

1. Create a fork of the authoritative repo in your GitHub account (**dashboard**)
2. Clone the fork repository to your local machine
`git clone git@github.com<your-fork-repo>`
3. Add the authoritative repository for the reference and updates
`git remote add upstream <url-of-the-authoritative-repo>`
4. Checkout a branch to make changes: `git checkout master`
5. **MAKE CHANGES, MAKE A DIFFERENCE!**
6. Fetch any changes from the authoritative repository:
`git fetch upstream master`
7. Merge changes from the updated remote reference into the your branch
`git merge upstream/master`
8. Push your changes back to your fork on GitHub: `git push origin master`
9. Create a pull request (PR) into the authoritative repository (**dashboard**)



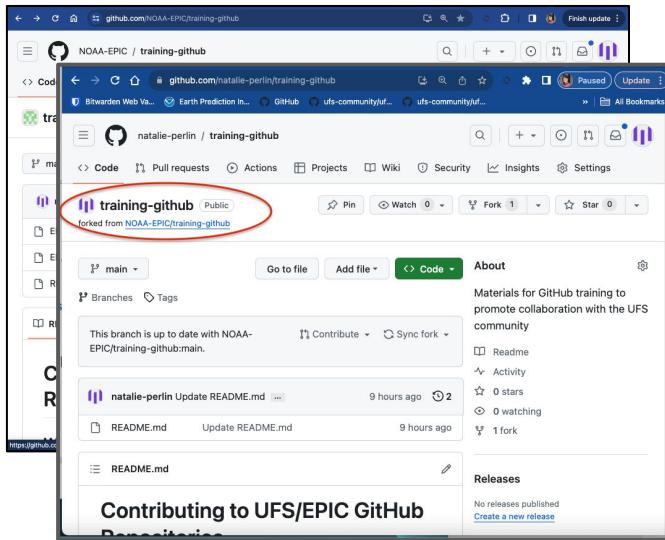
EARTH PREDICTION INNOVATION CENTER (EPIC)



Git workflow example with an existing authoritative repo

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

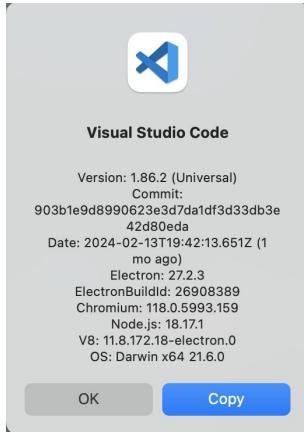
- o fetch and merge remote branches, update a fork repository



```
git clone git@github.com:<repo-fork>
cd <training-github>
git remote add upstream <authoritative-repo>
git remote -v
git checkout master
# make changes, commit them to a master branch
# fetch any updates from the authoritative repository:
git fetch upstream
git merge upstream/master
git log --oneline
git push origin master
```

Git workflow example with an existing authoritative repo

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



The screenshot shows a Mac OS X desktop environment. In the foreground, a terminal window is open with the command 'git log' or a similar command, displaying a list of commits from a GitHub repository named 'training-github'. The commits are listed in reverse chronological order, starting with 'Initial commit' and ending with the most recent commit. In the background, a code editor window for Visual Studio Code is visible, showing the same 'README2' file with the same five commits. The code editor's status bar at the bottom shows the path '/Users/natalie/Downloads/training-github' and other standard code editor settings.

```
git log
dc91e01 Initial commit
6851ca Update README.md
27d98dd Update README.md
daaa25d Added GitHub Tutorial Part 1 files, pdf and pptx
4cba347 (origin/master, origin/HEAD, master) Update README.md
d3c850d second commit
175c441 first commit
d5735b5 (two) third commit
eefd59d fourth commit
fb61b36 (one) Merge branch 'two' into one
42b2f4d fifth commit
762ef34 (upstream/master) Create README2
903b1e9d8990623e3d7da1d3d33db3e
42d80eda
Date: 2024-02-13T19:42:13.651Z (1
mo ago)
Electron: 27.2.3
ElectronBuildId: 26908389
Chromium: 118.0.5993.159
Node.js: 18.17.1
V8: 11.8.172.18-electron.0
OS: Darwin x64 21.6.0
```



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
```



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 (*)

The screenshot shows the GitHub interface for the repository 'nathalie-perlin / training-github'. The repository is a fork of 'NOAA-EPIC/training-github'. The main page displays recent activity, including a push from 'nathalie-perlin' and several commits from 'EPIC_GitHub_Tutorial'. The 'Compare & pull request' button is highlighted with a red circle.

The screenshot shows the 'Open a pull request' dialog box. It allows users to compare changes across two branches. The 'base repository' is set to 'NOAA-EPIC/training-github' and the 'head repository' is set to 'nathalie-perlin/training-github'. The 'compare' dropdown is also highlighted with a red circle.

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



EARTH PREDICTION INNOVATION CENTER (EPIC)

Local repository: keep it synched with the remote!

- An existing authoritative repository: <https://github.com/NOAA-EPIC/training-github>
- Remember to sync your primary (master) branch with that from the original repo!

The figure consists of three side-by-side screenshots of a GitHub repository interface. All three screenshots show the same repository: `natalie-perlin / training-github`, which is a fork of `NOAA-EPIC/training-github`.

- Screenshot 1:** Shows the master branch status as "This branch is 1 commit behind NOAA-EPIC/training-github:master". A red circle highlights this message.
- Screenshot 2:** Shows the master branch status as "This branch is 1 commit behind NOAA-EPIC/training-github:master". A red circle highlights this message. Below it, a callout box says "This branch is out-of-date. Update branch to keep this branch up-to-date by syncing a commit from the upstream repository." A red circle highlights the "Sync fork" button, and another red circle highlights the "Update branch" button at the bottom of the callout.
- Screenshot 3:** Shows the master branch status as "This branch is up to date with NOAA-EPIC/training-github:master". A red circle highlights this message.



EARTH PREDICTION INNOVATION CENTER (EPIC)

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

```
git clone git@github.com:<user-fork>/ufs-srweather-app.git  
cd ufs-srweather-app  
git remote add upstream git@github.com:ufs-community/ufs-srweather-app.git  
# checkout a specific PR from the upstream branch:  
  
git pull upstream pull/<PR#>/head:PRtest  
git checkout PRtest  
  
# Test the PR, build, run the code  
# Comment on the PR on specific issues, files, or specific lines  
# Make your own pull request into original PR-author's repository to suggesting your own changes
```

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

Edit <> Code ▾

Open

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



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

```
git clone git@github.com:<user-fork>/ufs-srweather-app.git  
cd ufs-srweather-app  
git remote add upstream git@github.com:ufs-community/ufs-srweather-app.git  
# checkout a specific PR from the upstream branch:  
  
git pull upstream pull/<PR#>/head:PRtest  
git checkout PRtest  
  
# Test the PR, build, run the code  
# Comment on the PR on specific issues, files, or specific lines  
# Make your own pull request into original PR-author's repository to suggesting your own changes
```

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

Edit <> Code ▾

Open

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

Repositories with submodules or external dependencies

- 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
```

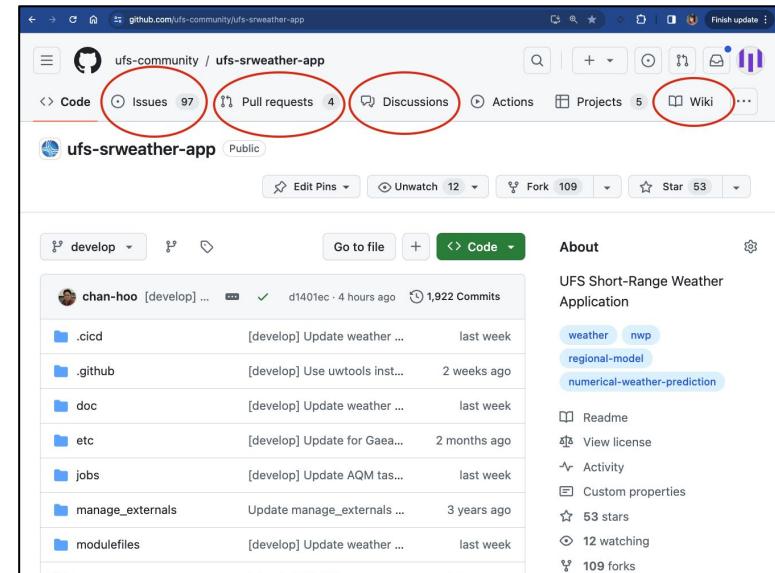
```
git clone --recurse-submodules https://github.com/jcsda/spack-stack.git
```

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



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.

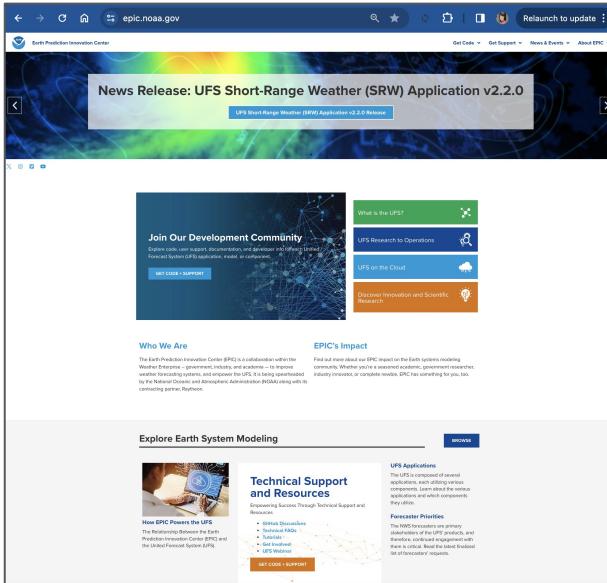
- Screenshot 1:** Shows the Issues page with 97 open issues. A search bar at the top has the query `is:issue is:open`. On the left, a sidebar lists several pull requests with green checkmarks, such as `create_symlink_to`, `Users guide needs t`, and `[SRW-AQM] Need to`.
- Screenshot 2:** Shows the Pull 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: "Announcement: UFS Short-Range Weather Application 10/31/2023". Below the announcement, there's a "Categories" section with links like "View all discussions", "Announcements", "Enhancements", "General", and "Ideas". To the right, there's a "Discussions" section with a post titled "Gaea C4/F2 Dec Friday 8 March" and another titled "Methods for better utilizing cores and nodes on RDHPCS".

EARTH PREDICTION INNOVATION CENTER (EPIC)



Happy Modeling and Contributing to UFS!

- <https://epic.noaa.gov/>
- <https://github.com/ufs-community>



A screenshot of the GitHub organization page for "ufs-community" at https://github.com/ufs-community. The page shows several repositories: "ufs-weather-model" (UFS Weather Model, Fortran, 126 stars), "ccpp-physics" (UFS fork for CCPP, Fortran, 233 stars), "uwtools" (Workflow tools for UFS, Python, 143 stars), and "ufs-srweather-app" (UFS Short-Range Weather Application, Python, 53 stars). The page also displays discussions and a list of contributors.



EARTH PREDICTION INNOVATION CENTER (EPIC)

