

EPIC GitHub Tutorial: Contributing to UFS/EPIC Repositories



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

EARTH PREDICTION INNOVATION CENTER (EPIC)



Welcome!

- Tutorial materials and a training session presentation could be found at
<https://github.com/NOAA-EPIC/training-github>
- The tutorial presents general directions for Mac OS or Linux* system (Terminal)
- 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)



Outline of the GitHub Tutorial

Part 1. Git/GitHub Basics

- Cover some Git and GitHub terms and concepts
- Basic Git commands to *git* you up
- Set up a SSH key pair for easy GitHub authentication
- Create your own repository
- Track your local changes
- Push your local changes to your GitHub



EARTH PREDICTION INNOVATION CENTER (EPIC)



Outline of the GitHub Tutorial

Part 2. Working with Remote Repositories

- Checkout remote repositories: forks and clones
- Creating and managing GitHub forks, branches
- Submitting pull requests
- Fetching and merging remote branches
- Resolve merge conflicts
- Keeping local repositories in sync with remote ones



Part 1.

Basic Git Terms and Concepts

- **Git:** A program; a free, open-source version control system
- **Version control system (VCS):** software providing management of changes to computer programs, documentation, web sites, etc.
- **GitHub** (github.com): The host website for many free and open-source repositories, including numerous ones by NOAA
- **Repository (repo):** a set of code, documents, website(s), etc., that are version controlled
- **Branch:** a working version of a repository with its own change history



EARTH PREDICTION INNOVATION CENTER (EPIC)



Basic Local Git Commands (`git + <command>`)

- **config:** configure your git experience
- **init:** create a new local repository in the existing directory tracked by git
- **add:** add files to be tracked by git
- **commit:** save changes you have made to the repository
- **branch:** verify a current branch, copy the current branch to create a new one
- **checkout:** check out a specific branch within a repository
- **log:** show recorded (committed) change history to the repository
- **status:** show changes to the repository since the recent commit
- **diff:** show local file/line-by-line changes to the repository since the recent commit



Common Local Git Commands

- **config:** configure your git experience
- **init:** create a new local repository in the existing directory tracked by git
- **add:** add files to be tracked by git
- **commit:** save changes made to the repo
- **branch:** verify a current branch, copy the current branch to create a new one
- **checkout:** check out a specific branch
- **log:** show recorded (committed) change history
- **status:** show changes to repository since the recent commit
- **diff:** show local file/line-by-line changes to the repository since the recent commit

```
> git config --global user.name  
Natalie Perlin  
> git config --global user.name "Natalie Perlin"  
> git config --global user.email natalie.perlin@noaa.gov  
> git config --global user.email  
natalie.perlin@noaa.gov  
> git config --global core.editor  
> git config --global core.editor "vim"  
> git config --global core.editor  
vim  
> █
```

```
> cat ~/.gitconfig  
# This is Git's per-user configuration file.  
[user]  
    email = natalie.perlin@noaa.gov  
    name = Natalie Perlin  
[core]  
    editor = vim  
> █
```



Common Local Git Commands

- **config**: configure your git experience
- **init**: create a new local repository in the existing directory tracked by git
- **add**: add files to be tracked by git
- **commit**: save changes made to the repo
- **branch**: verify a current branch, copy the current branch to create a new one
- **checkout**: check out a specific branch
- **log**: show recorded (committed) change history
- **status**: show changes to repository since the recent commit
- **diff**: show local file/line-by-line changes to the repository since the recent commit

```
[Natalie@Mac:~]$ mkdir my_new_repo
[Natalie@Mac:~]$ cd my_new_repo/
[Natalie@Mac:~/my_new_repo]$ git init
hint: Using 'master' as the name for the initial branch. This default br
anch name
hint: is subject to change. To configure the initial branch name to use
in all
hint: of your new repositories, which will suppress this warning, call
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this com
mand:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/Natalie/my_new_repo/.git/
[Natalie@Mac:~/my_new_repo]$ ls -d .git
.git
[Natalie@Mac:~/my_new_repo]$ █
```



Common Local Git Commands

- **config**: configure your git experience
- **init**: create a new local repository in the existing directory tracked by git
- **add**: add files to be tracked by git
- **commit**: save changes made to the repo
- **branch**: verify a current branch, copy the current branch to create a new one
- **checkout**: check out a specific branch
- **log**: show recorded (committed) change history
- **status**: show changes to repository since the recent commit
- **diff**: show local file/line-by-line changes to the repository since the recent commit

```
[Natalie@Mac:~/my_new_repo]$ echo "This is my new repository." > README
[Natalie@Mac:~/my_new_repo]$ ls
README
[Natalie@Mac:~/my_new_repo]$ git add README
```



Common Local Git Commands

- **config**: configure your git experience
- **init**: create a new local repository in the existing directory tracked by git
- **add**: add files to be tracked by git
- **commit**: save changes made to the repo
- **branch**: verify a current branch, copy the current branch to create a new one
- **checkout**: check out a specific branch
- **log**: show recorded (committed) change history
- **status**: show changes to repository since the recent commit
- **diff**: show local file/line-by-line changes to the repository since the recent commit

```
[Natalie@Mac:~/my_new_repo]$ echo "This is my new repository." > README
[Natalie@Mac:~/my_new_repo]$ ls
README
[Natalie@Mac:~/my_new_repo]$ git add README
[Natalie@Mac:~/my_new_repo]$ git commit -m "First commit to my new repository"
[master (root-commit) 5e8b26e] First commit to my new repository
 1 file changed, 1 insertion(+)
 create mode 100644 README
```



Common Local Git Commands

- **config**: configure your git experience
- **init**: create a new local repository in the existing directory tracked by git
- **add**: add files to be tracked by git
- **commit**: save changes made to the repo
- **branch**: verify a current branch, copy the current branch to create a new one
- **checkout**: check out a specific branch
- **log**: show recorded (committed) change history
- **status**: show changes to repository since the recent commit
- **diff**: show local file/line-by-line changes to the repository since the recent commit

```
[Natalie@Mac:~/my_new_repo]$ echo "This is my new repository." > README
[Natalie@Mac:~/my_new_repo]$ ls
README
[Natalie@Mac:~/my_new_repo]$ git add README
[Natalie@Mac:~/my_new_repo]$ git commit -m "First commit to my new repository"
[master (root-commit) 5e8b26e] First commit to my new repository
 1 file changed, 1 insertion(+)
 create mode 100644 README
[Natalie@Mac:~/my_new_repo]$ git branch
* master
[Natalie@Mac:~/my_new_repo]$ git branch new_branch
[Natalie@Mac:~/my_new_repo]$ git branch
* master
  new_branch
```



Common Local Git Commands

- **config**: configure your git experience
- **init**: create a new local repository in the existing directory tracked by git
- **add**: add files to be tracked by git
- **commit**: save changes made to the repo
- **branch**: verify a current branch, copy the current branch to create a new one
- **checkout**: check out a specific branch
- **log**: show recorded (committed) change history
- **status**: show changes to repository since the recent commit
- **diff**: show local file/line-by-line changes to the repository since the recent commit

```
[Natalie@Mac:~/my_new_repo]$ echo "This is my new repository." > README
[Natalie@Mac:~/my_new_repo]$ ls
README
[Natalie@Mac:~/my_new_repo]$ git add README
[Natalie@Mac:~/my_new_repo]$ git commit -m "First commit to my new repository"
[master (root-commit) 5e8b26e] First commit to my new repository
  1 file changed, 1 insertion(+)
   create mode 100644 README
[Natalie@Mac:~/my_new_repo]$ git branch
* master
[Natalie@Mac:~/my_new_repo]$ git branch new_branch
[Natalie@Mac:~/my_new_repo]$ git branch
* master
  new_branch
[Natalie@Mac:~/my_new_repo]$ git checkout new_branch
Switched to branch 'new_branch'
[Natalie@Mac:~/my_new_repo]$ █
```



Common Local Git Commands

- **config:** configure your git experience
- **init:** create a new local repository in the existing directory tracked by git
- **add:** add files to be tracked by git
- **commit:** save changes made to the repo
- **branch:** verify a current branch, copy the current branch to create a new one
- **checkout:** check out a specific branch
- **log:** show recorded (committed) change history
- **status:** show changes to repository since the recent commit
- **diff:** show local file/line-by-line changes to the repository since the recent commit

```
[Natalie@Mac:~/my_new_repo]$ touch a_new_file.txt
[Natalie@Mac:~/my_new_repo]$ echo "This is a new file" > a_new_file.txt
[Natalie@Mac:~/my_new_repo]$ git add a_new_file.txt
[Natalie@Mac:~/my_new_repo]$ ls
README
    a_new_file.txt
[Natalie@Mac:~/my_new_repo]$ git commit a_new_file.txt -m "Adding a new
file"
[new_branch 169f89a] Adding a new file
 1 file changed, 1 insertion(+)
  create mode 100644 a_new_file.txt
[Natalie@Mac:~/my_new_repo]$ git log
commit 169f89aba6294630d19e57b9c02cab7c5ae8ed43 (HEAD -> new_branch)
Author: Natalie Perlin <natalie.perlin@noaa.gov>
Date:   Thu Nov 16 04:38:25 2023 -0500

    Adding a new file

commit 5e8b26ef670a45c94849746cd165a5b6cf9eea17 (master)
Author: Natalie Perlin <natalie.perlin@noaa.gov>
Date:   Thu Nov 16 04:26:05 2023 -0500

    First commit to my new repository
[Natalie@Mac:~/my_new_repo]$
```



Common Local Git Commands

- **config:** configure your git experience
- **init:** create a new local repository in the existing directory tracked by git
- **add:** add files to be tracked by git
- **commit:** save changes made to the repo
- **branch:** verify a current branch, copy the current branch to create a new one
- **checkout:** check out a specific branch
- **log:** show recorded (committed) change history
- **status:** show changes to repository since the recent commit
- **diff:** show local file/line-by-line changes to the repository since the recent commit

```
[Natalie@Mac:~/my_new_repo]$ git status
On branch new_branch
nothing to commit, working tree clean
[Natalie@Mac:~/my_new_repo]$ echo "This is an important README" >> README
[Natalie@Mac:~/my_new_repo]$ git status
On branch new_branch
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:   README

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



Common Local Git Commands

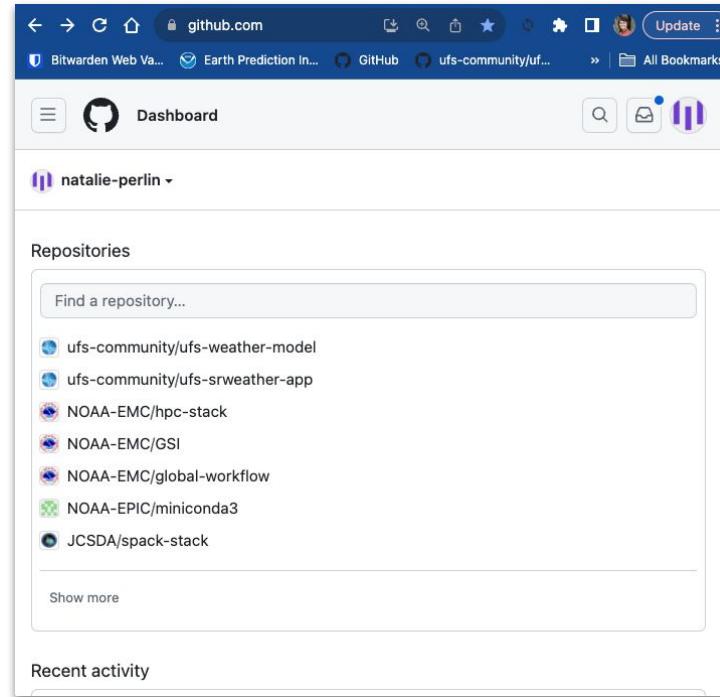
- **config:** configure your git experience
- **init:** create a new local repository in the existing directory tracked by git
- **add:** add files to be tracked by git
- **commit:** save changes made to the repo
- **branch:** verify a current branch, copy the current branch to create a new one
- **checkout:** check out a specific branch
- **log:** show recorded (committed) change history
- **status:** show changes to repository since the recent commit
- **diff:** show local file/line-by-line changes to the repository since the recent commit

```
[Natalie@Mac:~/my_new_repo]$ git diff
diff --git a/README b/README
index 5cc31d9..b93aa9a 100644
--- a/README
+++ b/README
@@ -1 +1,2 @@
This is my new repository.
+This is an important README
[Natalie@Mac:~/my_new_repo]$ git diff master
diff --git a/README b/README
index 5cc31d9..b93aa9a 100644
--- a/README
+++ b/README
@@ -1 +1,2 @@
This is my new repository.
+This is an important README
diff --git a/a_new_file.txt b/a_new_file.txt
new file mode 100644
index 0000000..6dfa057
--- /dev/null
+++ b/a_new_file.txt
@@ -0,0 +1 @@
+This is a new file
[Natalie@Mac:~/my_new_repo]$ git diff master -- README
diff --git a/README b/README
index 5cc31d9..b93aa9a 100644
--- a/README
+++ b/README
@@ -1 +1,2 @@
This is my new repository.
+This is an important README
[Natalie@Mac:~/my_new_repo]$
```



Creating a public GitHub profile

- Navigate to <https://github.com> dashboard
 - click on your user picture
- Choose Settings from the dropdown menu
- Fill out your public profile

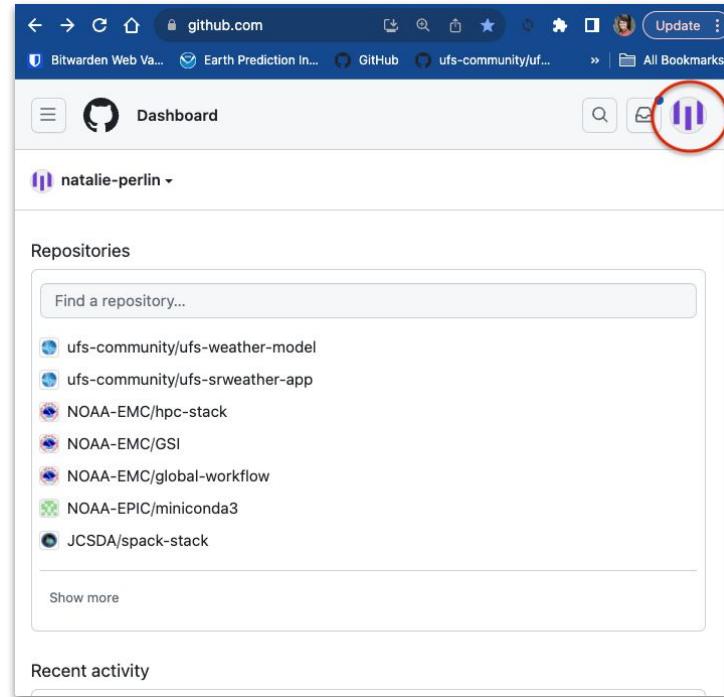


EARTH PREDICTION INNOVATION CENTER (EPIC)



Creating a public GitHub profile

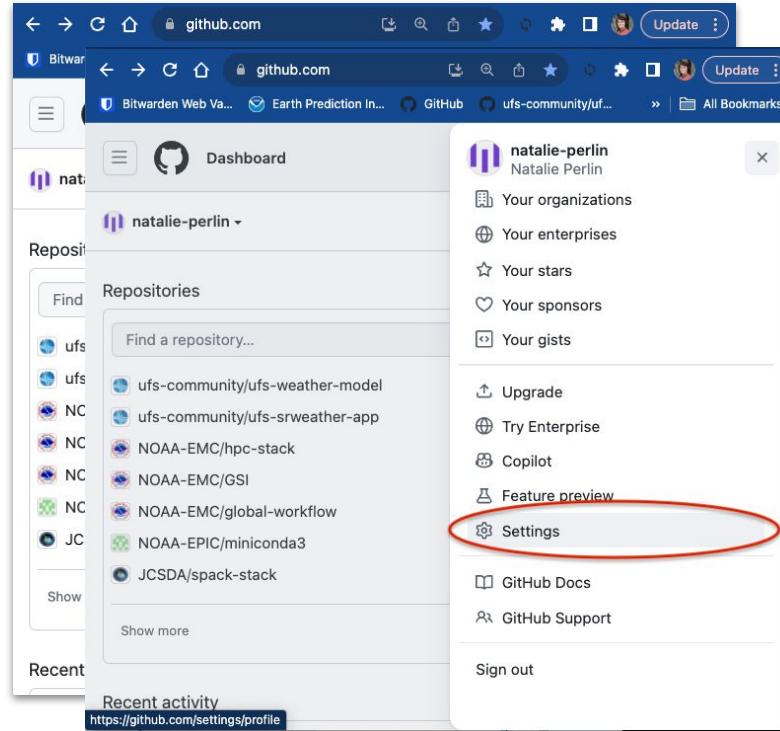
- Navigate to <https://github.com> dashboard
 - click on your user picture
- Choose Settings from the dropdown menu
- Fill out your public profile



EARTH PREDICTION INNOVATION CENTER (EPIC)

Creating a public GitHub profile

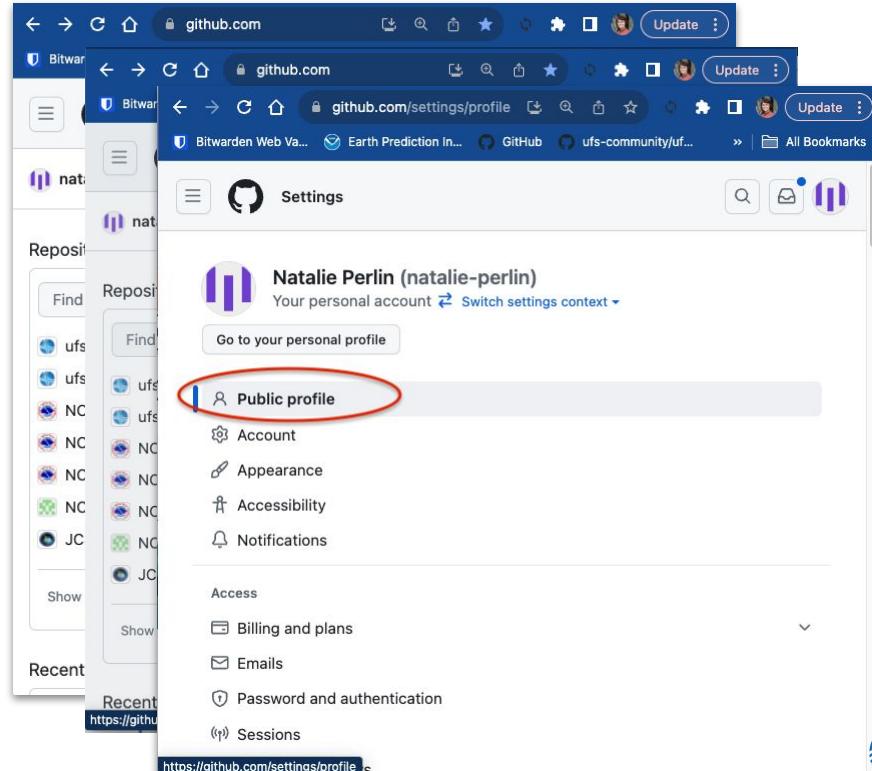
- Navigate to <https://github.com> dashboard
 - click on your user picture
- Choose Settings from the dropdown menu
- Fill out your public profile



EARTH PREDICTION INNOVATION CENTER (EPIC)

Creating a public GitHub profile

- Navigate to <https://github.com> dashboard
 - click on your user picture
- Choose Settings from the dropdown menu
- Fill out your public profile



EARTH PREDICTION INNOVATION CENTER (EPIC)

Generate SSH key pair for easy Git/GitHub communication

- From Settings, on the left, click on “SSH and GPG keys”, then “New SSH key”

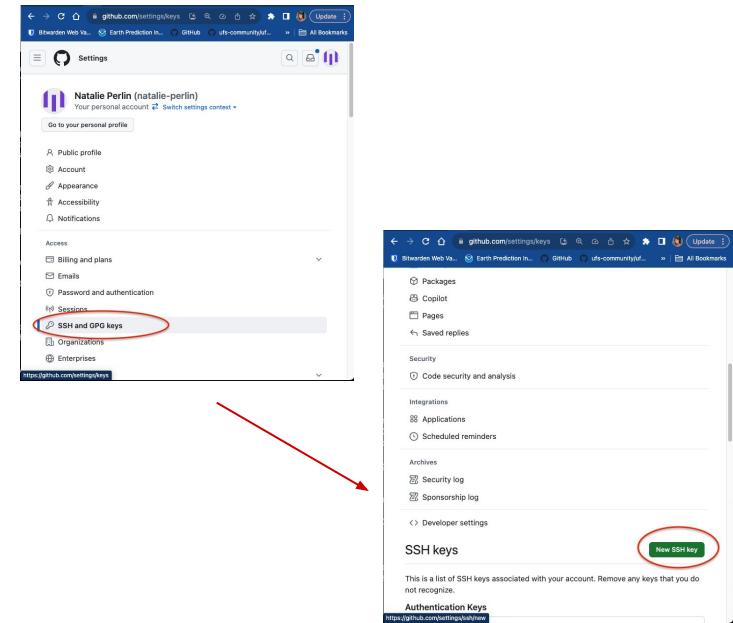
- In a local terminal, go to `~/.ssh` and generate a key pair:

```
ssh-keygen -t rsa -b 4096
```

- Name your key (full path) or use a default `id_rsa` name
- Copy the public key from the output:

```
cat id_rsa.pub
```

- Paste the output into the “Key” portion of the GitHub page
- Give it a title and click “Add SSH key”



more help: <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>



EARTH PREDICTION INNOVATION CENTER (EPIC)



Generate SSH key pair for easy Git/GitHub communication

- From Settings, on the left, click on “SSH and GPG keys”, then “New SSH key”
- In a local terminal, go to ~/.ssh and generate a key pair:

```
ssh-keygen -t rsa -b 4096
```

- Name your key (full path) or use a default *id_rsa* name
- Copy the public key from the output:

```
cat id_rsa.pub
```

- Paste the output into the “Key” portion of the GitHub page
- Give it a title and click “Add SSH key”

```
[Natalie@Mac:~/my_new_repo]$ cd ~/.ssh
[Natalie@Mac:~/.ssh]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/Natalie/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/Natalie/.ssh/id_rsa
Your public key has been saved in /Users/Natalie/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:AWDLDFN6QNAGGgD1T2d686elsYnF76UYKeuKx4OD9WI Natalie@MacBook-Pro.local
The key's randomart image is:
+---[RSA 4096]---+
|O*oo...
|...+*...
|.. o= . +
| .. o +
| .. o S
| .. + .
| o + . B o .
| . E. = % o
| ..+.*+
+---[SHA256]---+
[Natalie@Mac:~/.ssh]$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yC2EAAAQABAAACQChGSs49zKLPOduAnn206s5nb58Y4CZXVc
53p5jm/zyb568savB1/j19Db2VcaomkVHPF2Zox/W12SsD5oBJabIxguvh16Mv6DXHRsq
GD1zdrhDzJxfdqtuFha/KJC97cjqwz8Z5/u3aRvUsjn2ILRygnAyfE8xc64q4f1aB+mc8
D7LzKTtdZgUjrrrbQMxpBo16qouBXudiLF9K+eBoAsJaVGCAbs6G0JnDs5Bkst1I405anuQ
Xq4PhVhFnkzMhW0mRjPFBEIaq5c80XoYGuCgr5ac9I0iqVmaIy/76GRYrNgFlab19w6kJ
4npPLWbxPtaGfmpriMIUNKUdeVqkZmxInaPV6xKiTvE8VxKKPFEbkPoF5cCtCXYz1paNRDS
SHfmhaw1idBdrv/Mk6B8puIk+PEtARBPoRdp9TkY93oRQJ0/XMC02p1fJLxbI7mdXY4Yku
OtPgTEw1k0CsRcxkWTU6HFhvlfbz/lj1RhZwVG9csXB815gp/h5zqEBkZ22QVDFY0TC9TE
7wQ5ZJePrSSScrxEqSmPxsr88dSORauHURw6eguIbgKgwXboYlx1JfbGvu63px6pcdux
tCr4QwoTeS/Dax0AbmkBN91w3AVD+tC6pUfRxu00CC+VqzWx7cwh4zV1RPeabRUaWv8Vaf
+bMURKp7k3Nnw== Natalie@MacBook-Pro.local
[Natalie@Mac:~/.ssh]$
```

more help: <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>



EARTH PREDICTION INNOVATION CENTER (EPIC)



Generate SSH key pair for easy Git/GitHub communication

- From Settings, on the left, click on “SSH and GPG keys”, then “New SSH key”
- In a local terminal, go to `~/.ssh` and generate a key pair:

```
ssh-keygen -t rsa -b 4096
```

- Name your key (full path) or use a default `id_rsa` name
- Copy the public key from the output:

```
cat id_rsa.pub
```

- Paste the output into the “Key” portion of the GitHub page
- Give it a title and click “Add SSH key”

```
[Natalie@Mac:~/my_new_repo]$ cd ~/.ssh
[Natalie@Mac:~/.ssh]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/Natalie/.ssh/id_rsa): ○
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/Natalie/.ssh/id_rsa
Your public key has been saved in /Users/Natalie/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:AWLDNF6QNAGG9D1T2d686elsYnF76UYKeuKx4OD9WI Natalie@MacBook-Pro.local
The key's randomart image is:
+---[RSA 4096]---+
|O*oo...
|...+*...
|.. o= . +
| . . +
| . o S |
| . . + .
| o + . B o .
| . E. = % o |
| ..+*+ *+
+---[SHA256]---+
[Natalie@Mac:~/.ssh]$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yC2EAAAQABAAACQChGSs49zKLPOduAnn206s5nb58Y4CZXVc
53p5jm/zyb568savB1/j19Db2VcaomkVHPF2Zox/W12SsD5oBJabIXgUsvh16Mv6DXHRsq
GD1zdrhDzJxfdqtuFha/KJC9s7cjqwz8Z5/u3aRvUsjh2ILRygnAyfE8xc64q4f1aB+mc8
D7LzKTtdZgUjrrrbQMxpBo16qouBXudiLF9K+eBoAsJaVGCAbs6G0JnDs5Bkst1I405anuQ
Xq4PhVhFnkzMhW0mRjPFBEIaq5c80XoYGuCgr5ac9IOiqVmaIy/76GRYrNgFlab19w6kJ
4npPLWbxPtaGfmpriMIUNKUdeVqkZmxInaPV6xKiTvE8VxKKPFEbkPoF5cCtCXy1paNRDS
SHfmhaw1idBdrv/Mk6B8puIk+PEtARBPoRdp9TkY93oRQJ0/XMC02p1fJLxbI7mdXY4Yku
OtPgTEw1k0CsRcxkWTU6HFhvlfbz/lj1RhZwVG9csXB815gp/h5zqEBkZ22QVDFY0TC9TE
7wQ5ZjePrSSScrxPsxr88dSORauHURw6eguIbgKgwXboYlx1JfbGvUw63px6pcdux
tCr4QwoTeS/Dax0AbmkBN91w3AVD+tC6pUfRxu00CC+VqzWx7cwh4zV1RPeabRuWv8Vaf
+bMURKp7k3Nnw== Natalie@MacBook-Pro.local
[Natalie@Mac:~/.ssh]$
```

more help: <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>



EARTH PREDICTION INNOVATION CENTER (EPIC)



Generate SSH key pair for easy Git/GitHub communication

- From Settings, on the left, click on “SSH and GPG keys”, then “New SSH key”
- In a local terminal, go to `~/.ssh` and generate a key pair:

```
ssh-keygen -t rsa -b 4096
```

- Name your key (full path) or use a default `id_rsa` name
- Copy the public key from the output:

```
cat id_rsa.pub
```

- Paste the output into the “Key” portion of the GitHub page
- Give it a title and click “Add SSH key”

```
[Natalie@Mac:~/my_new_repo]$ cd ~/.ssh
[Natalie@Mac:~/.ssh]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/Natalie/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/Natalie/.ssh/id_rsa
Your public key has been saved in /Users/Natalie/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:AWLDNF6QNAGGgD1T2d686elsYnF76UYKeuKx4OD9WI Natalie@MacBook-Pro.local
The key's randomart image is:
+---[RSA 4096]---+
|O*oo...
|...+*...
|.. o= . +
| .. o +
| . o S
| . . +
| o + . B o .
| . E. = % o
| ..+.*+
+---[SHA256]---+
[Natalie@Mac:~/.ssh]$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yC2EAAAQABAAQChGSs49zKLPOduAnn206s5nb58Y4CZXVc
53p5jm/zyb568savB1/j19Db2VcaomkVHPF2Zox/W12SsD5oBJabIXgUsvh16Mv6DXHrsq
GD1zdrDzJxfdqtuFha/KJC9s7cjqwz8Z5/u3aRvUsjh2ILRygnAyfE8xc64q4f1aB+mc8
D7LzKTtdZgUjrrbQMxpBo16qouBXudiLF9K+eBoAsJaVGCAbs6G0JnDs5Bkst1I405anuQ
Xq4PhVhFnkzMhW0mRjPFBEIaq5c80XoYGuCgr5ac9I0iqVmaIy/76GRYrNgFlab19w6kJ
4npPLWbxPtagfmpRIUNKUdeVqkZmxInaPV6xKiTvE8VxKKPFEbkPoF5cCtCXYz1paNRDS
SHfmhaw1idBdrv/Mk6B8puIk+PEtARBPoRdp9TkV93oRQJ0/XMC02p1fJLxbI7mdXY4Yku
OtPgTEw1k0CsRcxkWTU6HFhV1Fhbz/lj1RhZwV69csXB15gp/h5zqEBkZ22QVDFY0TC9TE
7wQ5ZJePrSSScrxEqSmPxsr88dS0RaUHRw6eguIbgKgwXboYlx1JfbGvu63px6pcdux
tCr4QwoTeS/Dax0AbmkBN91w3AVD+tC6pUfRxU0CC+vqzWx7cwh4zV1RPeabRuWv8Vaf
+bMURKp7k3Nw== Natalie@MacBook-Pro.local
[Natalie@Mac:~/.ssh]$
```

more help: <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>



EARTH PREDICTION INNOVATION CENTER (EPIC)



Generate SSH key pair for easy Git/GitHub communication

- From Settings, on the left, click on “SSH and GPG keys”, then “New SSH key”
- In a local terminal, go to `~/.ssh` and generate a key pair:

```
ssh-keygen -t rsa -b 4096
```

- Name your key (full path) or use a default `id_rsa` name
- Copy the public key from the output:

```
cat id_rsa.pub
```

- Paste the output into the “Key” portion of the GitHub page
- Give it a title and click “Add SSH key”

gitHub.com/settings/ssh/

Add new SSH Key

Title:

Key type: Authentication Key

Key:

```
Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'
```

Add SSH key

more help: <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

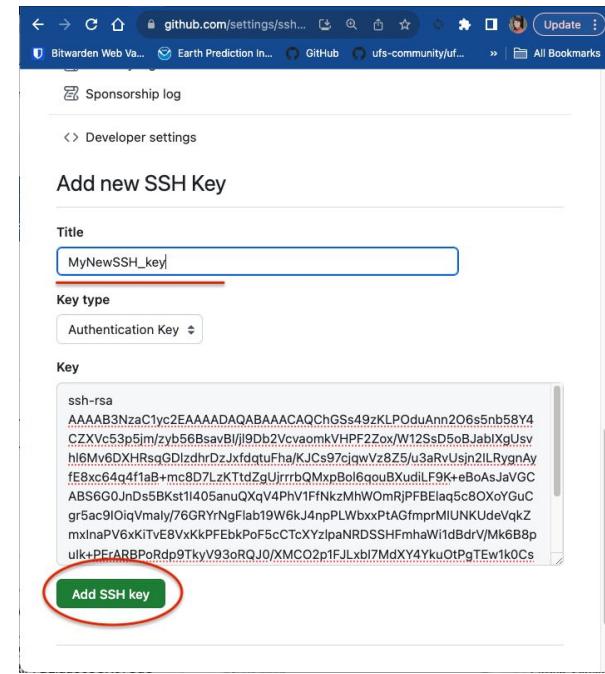


EARTH PREDICTION INNOVATION CENTER (EPIC)



Generate SSH key pair for easy Git/GitHub communication

- From Settings, on the left, click on “SSH and GPG keys”, then “New SSH key”
- In a local terminal, go to `~/.ssh` and generate a key pair:
`ssh-keygen -t rsa -b 4096`
- Name your key (full path) or use a default `id_rsa` name
- Copy the public key from the output:
`cat id_rsa.pub`
- Paste the output into the “Key” portion of the GitHub page
- Give it a title and click “Add SSH key”**



The screenshot shows the GitHub settings interface for managing SSH keys. The user has entered a title 'MyNewSSH_key' and selected 'Authentication Key' as the key type. The 'Key' field contains a long string of characters representing an RSA public key. At the bottom of the form, there is a green 'Add SSH key' button, which is also circled in red to indicate it as the next step.

more help: <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

EARTH PREDICTION INNOVATION CENTER (EPIC)

Add a private SSH key to your local ssh agent

- Start a new SSH agent:

```
eval "$(ssh-agent -s)"
```

- Add a new private key to the ssh agent (Mac OS):

```
ssh-add ~/.ssh/id_rsa
```

```
[Natalie@Mac:~/ssh]$ eval "$(ssh-agent -s)"  
Agent pid 35216  
[Natalie@Mac:~/ssh]$ ssh-add ~/.ssh/id_rsa  
Identity added: /Users/Natalie/.ssh/id_rsa (Natalie@MacBook-Pro.local)  
[Natalie@Mac:~/ssh]$ █
```

- Now pushes (uploads) to your GitHub repositories will not require a password

more help on SSH Keys: <https://www.ssh.com/academy/ssh/keygen>

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>



EARTH PREDICTION INNOVATION CENTER (EPIC)



Creating a repository on GitHub

- Head to <https://github.com/new>
- Give your new repository a unique name (my_new_repo) and optionally a description
- Since we will be pushing (uploading) our repository from the terminal, **do not** select “Add a README file”
- Scroll down and click “Create repository”

The screenshot shows the GitHub interface for creating a new repository. At the top, the URL is https://github.com/new. The main title is "Create a new repository". Below it, a sub-instruction says "A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)". A note states "Required fields are marked with an asterisk (*)." The "Repository template" dropdown is set to "No template". A placeholder text "Start your repository with a template repository's contents." is visible. The "Owner" field is populated with "natalie-perlin". The "Repository name" field is empty and highlighted with a red border. A note below says "Great repository names are short and memorable. Need inspiration? How about [upgraded-enigma](#) ?". The "Description (optional)" field is empty. Below these, there are two radio button options for visibility: "Public" (selected) and "Private". The "Public" option is described as "Anyone on the internet can see this repository. You choose who can commit.". The "Private" option is described as "You choose who can see and commit to this repository.". At the bottom, there is a section titled "Initialize this repository with:" containing a checkbox for "Add a README file". A note next to it says "This is where you can write a long description for your project. [Learn more about READMEs.](#)".



EARTH PREDICTION INNOVATION CENTER (EPIC)

Creating a repository on GitHub

- Head to <https://github.com/new>
- Give your new repository a unique name (`my_new_repo`) and optionally a description
- Since we will be pushing (uploading) our repository from the terminal, **do not** select “Add a README file”
- Scroll down and click “Create repository”

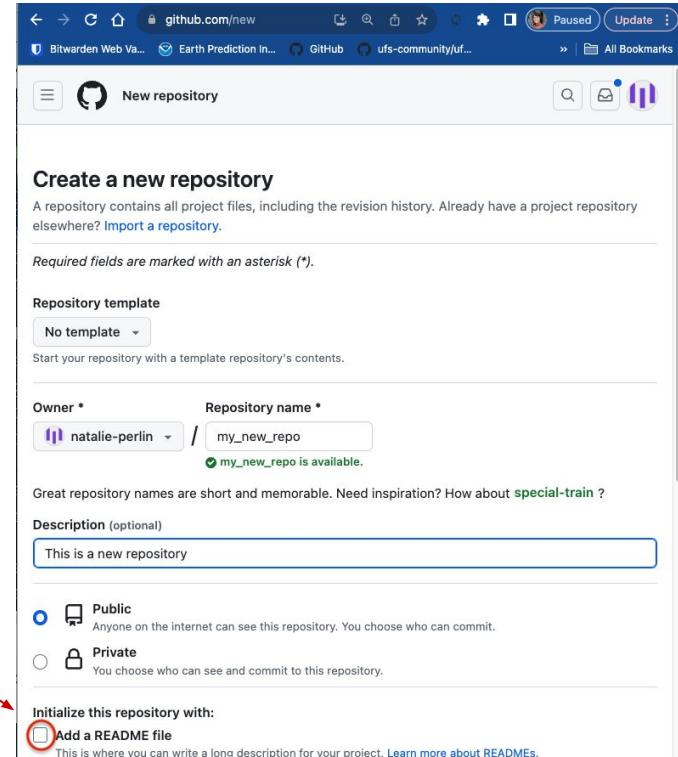
The screenshot shows the GitHub interface for creating a new repository. At the top, the URL is `github.com/new`. Below the header, there's a "New repository" section with a "Create a new repository" heading. A note says "A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository." It indicates that required fields are marked with an asterisk (*). The "Repository template" dropdown is set to "No template". The "Owner" field shows "natalie-perlin". The "Repository name" field is highlighted with a red oval and contains "`my_new_repo`". A green message below it says "`my_new_repo` is available.". The "Description (optional)" field contains "This is a new repository". Under "Visibility", the "Public" option is selected, with a description: "Anyone on the internet can see this repository. You choose who can commit." The "Private" option is also shown. At the bottom, there's a section for initializing the repository with "Add a README file" checked, and a note: "This is where you can write a long description for your project. Learn more about READMEs."



EARTH PREDICTION INNOVATION CENTER (EPIC)

Creating a repository on GitHub

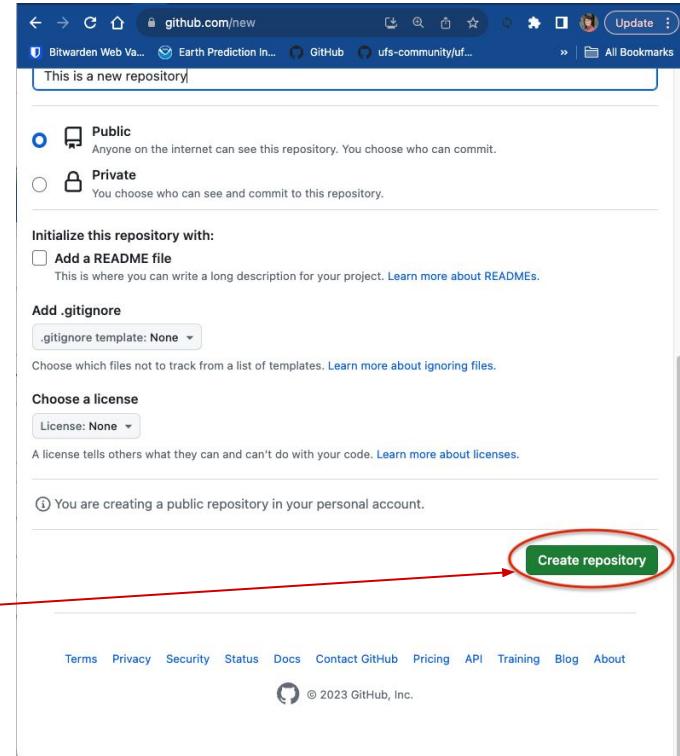
- Head to <https://github.com/new>
- Give your new repository a unique name (my_new_repo) and optionally a description
- Since we will be pushing (uploading) our repository from the terminal, **do not** select “Add a README file”
- Scroll down and click “Create repository”



EARTH PREDICTION INNOVATION CENTER (EPIC)

Creating a repository on GitHub

- Head to <https://github.com/new>
- Give your new repository a unique name (my_new_repo) and optionally a description
- Since we will be pushing (uploading) our repository from the terminal, **do not** select “Add a README file”
- Scroll down and click “Create repository”



EARTH PREDICTION INNOVATION CENTER (EPIC)

Pushing a Repository to GitHub

New gits commands:

- **push**: upload local repository content with committed changes to a remote repository
- **remote**: manage a list of remote entries tracked by a local repository
 - **remote -v** : query a remote repository or repositories tracked
 - **remote add origin**: "origin" remote refers to the default remote repository name



EARTH PREDICTION INNOVATION CENTER (EPIC)



Pushing a Repository to GitHub

- Return to your terminal and navigate back to your new repository (`my_new_repo`)
- Set the remote URL for the repository using `git remote add origin`
- Push both branches to GitHub
- Congrats on your new repo!! Go check it out on GitHub!

```
[Natalie@Mac:~/my_new_repo]$ git remote
[Natalie@Mac:~/my_new_repo]$ git remote add origin git@github.com:natalie-perlin/my_new_repo.git
[Natalie@Mac:~/my_new_repo]$ git remote -v
origin  git@github.com:natalie-perlin/my_new_repo.git (fetch)
origin  git@github.com:natalie-perlin/my_new_repo.git (push)
```



EARTH PREDICTION INNOVATION CENTER (EPIC)



Pushing a Repository to GitHub

- Return to your terminal and navigate back to your new repository (`my_new_repo`)
- Set the remote URL for the repository using `git remote add origin`
- Push both branches to GitHub
- Congrats on your new repo!! Go check it out on GitHub!

```
[Natalie@Mac:~/my_new_repo]$ git remote
[Natalie@Mac:~/my_new_repo]$ git remote add origin git@github.com:natalie-perlin/my_new_repo.git
[Natalie@Mac:~/my_new_repo]$ git remote -v
origin git@github.com:natalie-perlin/my_new_repo.git (fetch)
origin git@github.com:natalie-perlin/my_new_repo.git (push)
[Natalie@Mac:~/my_new_repo]$ git branch
  master
* new_branch
[Natalie@Mac:~/my_new_repo]$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 254 bytes | 254.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:natalie-perlin/my_new_repo.git
 * [new branch]      master -> master
[Natalie@Mac:~/my_new_repo]$ git push origin new_branch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 303.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'new_branch' on GitHub by visiting:
remote:     https://github.com/natalie-perlin/my_new_repo/pull/new/new_branch
remote:
To github.com:natalie-perlin/my_new_repo.git
 * [new branch]      new_branch -> new_branch
[Natalie@Mac:~/my_new_repo]$
```



Pushing a Repository to GitHub

- Return to your terminal and navigate back to your new repository (`my_new_repo`)
- Set the remote URL for the repository using `git remote add origin`
- Push both branches to GitHub
- Congrats on your new repo!! Go check it out on GitHub!

```
[Natalie@Mac:~/my_new_repo]$ git remote
[Natalie@Mac:~/my_new_repo]$ git remote add origin git@github.com:natalie-perlin/my_new_repo.git
[Natalie@Mac:~/my_new_repo]$ git remote -v
origin git@github.com:natalie-perlin/my_new_repo.git (fetch)
origin git@github.com:natalie-perlin/my_new_repo.git (push)
[Natalie@Mac:~/my_new_repo]$ git branch
  master
* new_branch
[Natalie@Mac:~/my_new_repo]$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 254 bytes | 254.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:natalie-perlin/my_new_repo.git
 * [new branch]      master -> master
[Natalie@Mac:~/my_new_repo]$ git push origin new_branch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 303.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'new_branch' on GitHub by visiting:
remote:     https://github.com/natalie-perlin/my_new_repo/pull/new/new_branch
remote:
To github.com:natalie-perlin/my_new_repo.git
 * [new branch]      new_branch -> new_branch
[Natalie@Mac:~/my_new_repo]$
```



Pushing a Repository to GitHub

- Return to your terminal and navigate back to your new repository (`my_new_repo`)
- Set the remote URL for the repository using `git remote add origin`
- Push both branches to GitHub
- Congrats on your new repo!! Go check it out on GitHub!

```
[Natalie@Mac:~/my_new_repo]$ git remote
[Natalie@Mac:~/my_new_repo]$ git remote add origin git@github.com:natalie-perlin/my_new_repo.git
[Natalie@Mac:~/my_new_repo]$ git remote -v
origin git@github.com:natalie-perlin/my_new_repo.git (fetch)
origin git@github.com:natalie-perlin/my_new_repo.git (push)
[Natalie@Mac:~/my_new_repo]$ git branch
  master
* new_branch
[Natalie@Mac:~/my_new_repo]$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 254 bytes | 254.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:natalie-perlin/my_new_repo.git
 * [new branch]      master -> master
[Natalie@Mac:~/my_new_repo]$ git push origin new_branch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 303.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'new_branch' on GitHub by visiting:
remote:     https://github.com/natalie-perlin/my_new_repo/pull/new_branch
remote:
To github.com:natalie-perlin/my_new_repo.git
 * [new branch]      new_branch -> new_branch
[Natalie@Mac:~/my_new_repo]$
```



Part 2. Working with Remote Repositories

- Checkout remote repositories: forks and clones
- Creating and managing GitHub forks, branches
- Submitting pull requests
- Fetching and merging remote branches
- Resolve merge conflicts
- Keeping local repositories in sync with remote ones



EARTH PREDICTION INNOVATION CENTER (EPIC)

Navigating GitHub Repositories

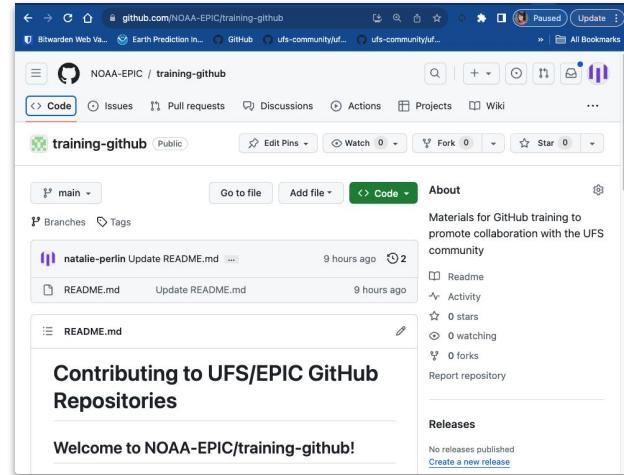
Any public Git repository can be *forked* or *cloned*.

- **fork:** a completely independent copy of Git repository
- **clone:** a linked copy synchronized with the target/remote repo
- Navigate to <https://github.com/NOAA-EPIC/training-github>
(or <https://github.com/ufs-community/ufs-srweather-app>)
- Look through the **Issues** and **Pull requests**
- Hop onto the **Discussions** and reply to a thread
- Create your own fork of the repository
 - Navigate to the main repository page by clicking on “Code”
 - In the top-right corner, click on “Fork”
 - Keep the “Repository name” and “Description” fields unchanged
 - Click “Create fork”

Navigating GitHub Repositories

Any public Git repository can be *forked* or *cloned*.

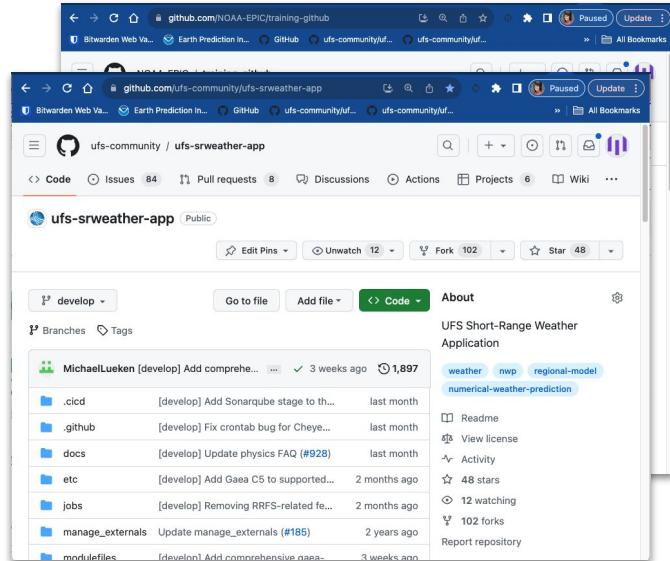
- **fork:** a completely independent copy of Git repository
- **clone:** a linked copy synchronized with the target/remote repo
- Navigate to <https://github.com/NOAA-EPIC/training-github>
(or <https://github.com/ufs-community/ufs-srweather-app>)
- Look through the **Issues** and **Pull requests**
- Hop onto the **Discussions** and reply to a thread
- Create your own fork of the repository
 - Navigate to the main repository page by clicking on “Code”
 - In the top-right corner, click on “Fork”
 - Keep the “Repository name” and “Description” fields unchanged
 - Click “Create fork”



Navigate a GitHub Repository - Create a Fork

Any public Git repository can be *forked* or *cloned*.

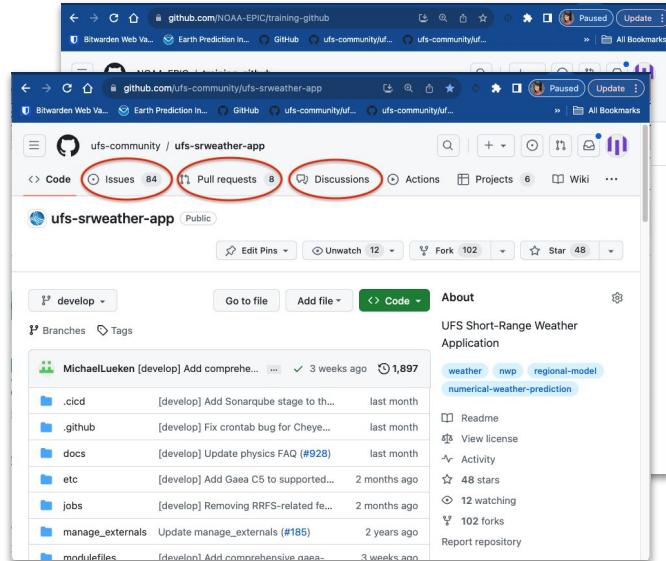
- **fork:** a completely independent copy of Git repository
- **clone:** a linked copy synchronized with the target/remote repo
- Navigate to <https://github.com/NOAA-EPIC/training-github>
(or <https://github.com/ufs-community/ufs-srweather-app>)
- Look through the **Issues** and **Pull requests**
- Hop onto the **Discussions** and reply to a thread
- Create your own fork of the repository
 - Navigate to the main repository page by clicking on “Code”
 - In the top-right corner, click on “Fork”
 - Keep the “Repository name” and “Description” fields unchanged
 - Click “Create fork”



Navigate a GitHub Repository - Create a Fork

Any public Git repository can be *forked* or *cloned*.

- **fork:** a completely independent copy of Git repository
- **clone:** a linked copy synchronized with the target/remote repo
- Navigate to <https://github.com/NOAA-EPIC/training-github>
(or <https://github.com/ufs-community/ufs-srweather-app>)
- Look through the **Issues** and **Pull requests**
- Hop onto the **Discussions** and reply to a thread
- Create your own fork of the repository
 - Navigate to the main repository page by clicking on “Code”
 - In the top-right corner, click on “Fork”
 - Keep the “Repository name” and “Description” fields unchanged
 - Click “Create fork”



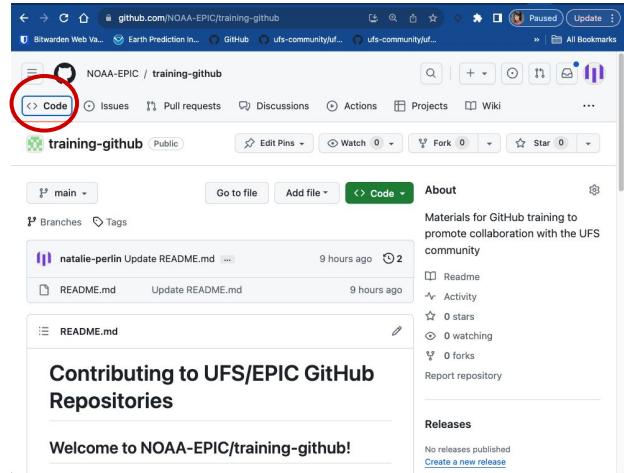
EARTH PREDICTION INNOVATION CENTER (EPIC)

 **40**
UNIFIED FORECAST SYSTEM

Navigate a GitHub Repository - Create a Fork

Any public Git repository can be *forked* or *cloned*.

- **fork:** a completely independent copy of Git repository
- **clone:** a linked copy that remains synchronized with the target/remote repo
- Navigate to <https://github.com/NOAA-EPIC/training-github>
- Look through the **Issues** and **Pull requests**
- Hop onto the **Discussions** and reply to a thread
- **Create your own fork of the repository**
 - Navigate to the main repository page by clicking on “Code”
 - In the top-right corner, click on “Fork” --> “Create a new fork”
 - Keep the “Repository name” and “Description” fields unchanged
 - Click “Create fork”



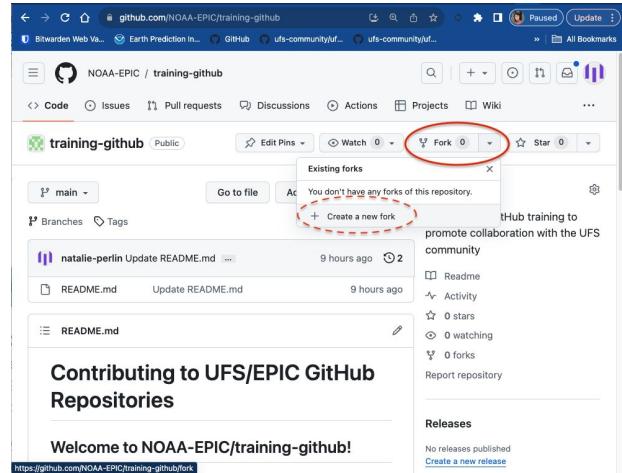
EARTH PREDICTION INNOVATION CENTER (EPIC)



Navigate a GitHub Repository - Create a Fork

Any public Git repository can be *forked* or *cloned*.

- **fork:** a completely independent copy of Git repository
- **clone:** a linked copy that remains synchronized with the target/remote repo
- Navigate to <https://github.com/NOAA-EPIC/training-github>
- Look through the **Issues** and **Pull requests**
- Hop onto the **Discussions** and reply to a thread
- Create your own fork of the repository
 - Navigate to the main repository page by clicking on “**Code**”
 - In the top-right corner, click on “**Fork**” --> “Create a new fork”
 - Keep the “Repository name” and “Description” fields unchanged
 - Click “Create fork”



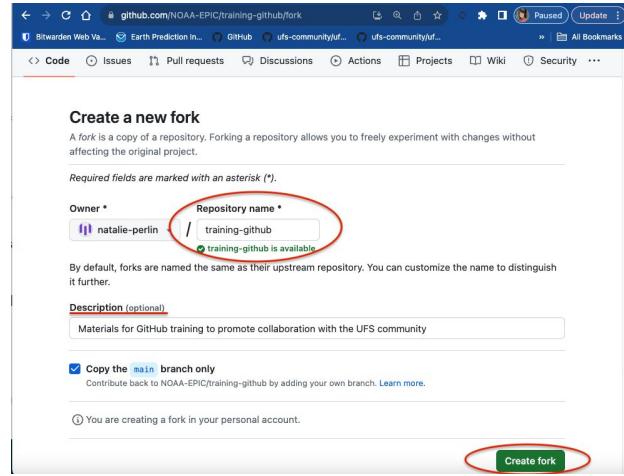
EARTH PREDICTION INNOVATION CENTER (EPIC)



Navigate a GitHub Repository - Create a Fork

Any public Git repository can be *forked* or *cloned*.

- **fork:** a completely independent copy of Git repository
- **clone:** a linked copy that remains synchronized with the target/remote repo
- Navigate to <https://github.com/NOAA-EPIC/training-github>
- Look through the **Issues** and **Pull requests**
- Hop onto the **Discussions** and reply to a thread
- Create your own fork of the repository
 - Navigate to the main repository page by clicking on “**Code**”
 - In the top-right corner, click on “**Fork**” --> “Create a new fork”
 - **Keep the “Repository name” and “Description” fields unchanged**
 - **Click “Create fork”**



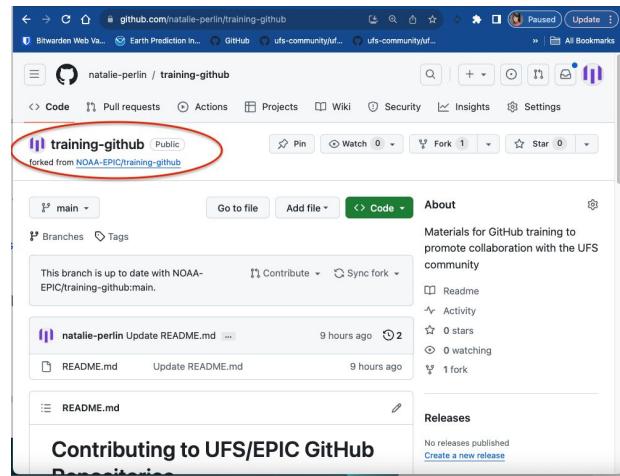
EARTH PREDICTION INNOVATION CENTER (EPIC)



Navigate a GitHub Repository - Create a Fork

Any public Git repository can be *forked* or *cloned*.

- **fork:** a completely independent copy of Git repository
- **clone:** a linked copy that remains synchronized with the target/remote repo
- Navigate to <https://github.com/NOAA-EPIC/training-github>
- Look through the **Issues** and **Pull requests**
- Hop onto the **Discussions** and reply to a thread
- Create your own fork of the repository
 - Navigate to the main repository page by clicking on “Code”
 - In the top-right corner, click on “Fork” --> “Create a new fork”
 - Keep the “Repository name” and “Description” fields unchanged
 - Click “Create fork”



EARTH PREDICTION INNOVATION CENTER (EPIC)