

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

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

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

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 branch 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: git config --global init.defaultBranch <name>
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
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 169f89aba6294630d19e57b7962afab7c5ae8ed43 (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")
```

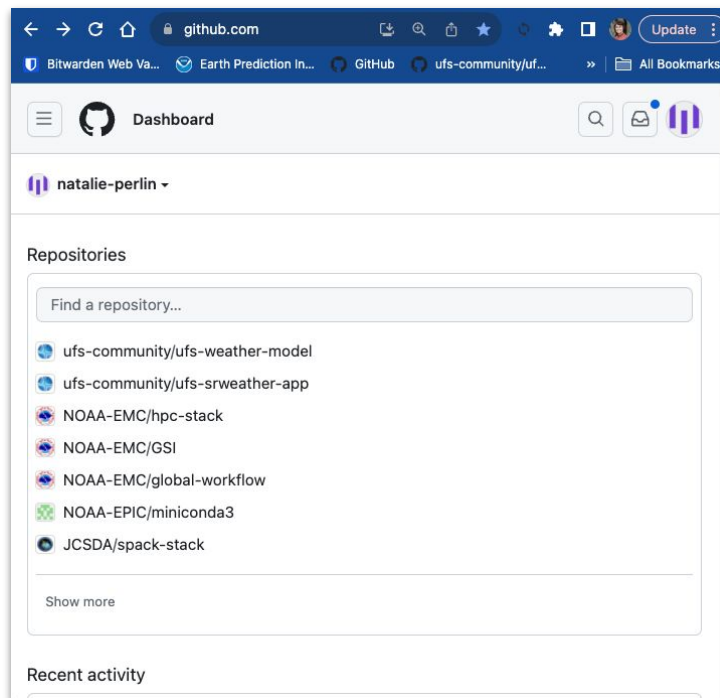
Common Local Git Commands

- **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
@@ -1,2 @@
 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
@@ -1,2 @@
 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
@@ -1,2 @@
 This is my new repository.
+This is an important README
[Natalie@Mac:~/my_new_repo]$
```

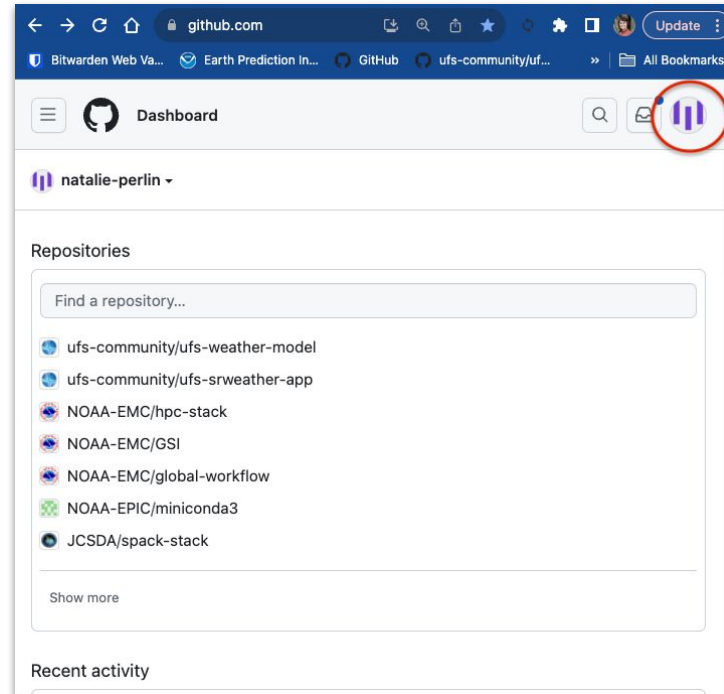
Creating a public GitHub profile

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



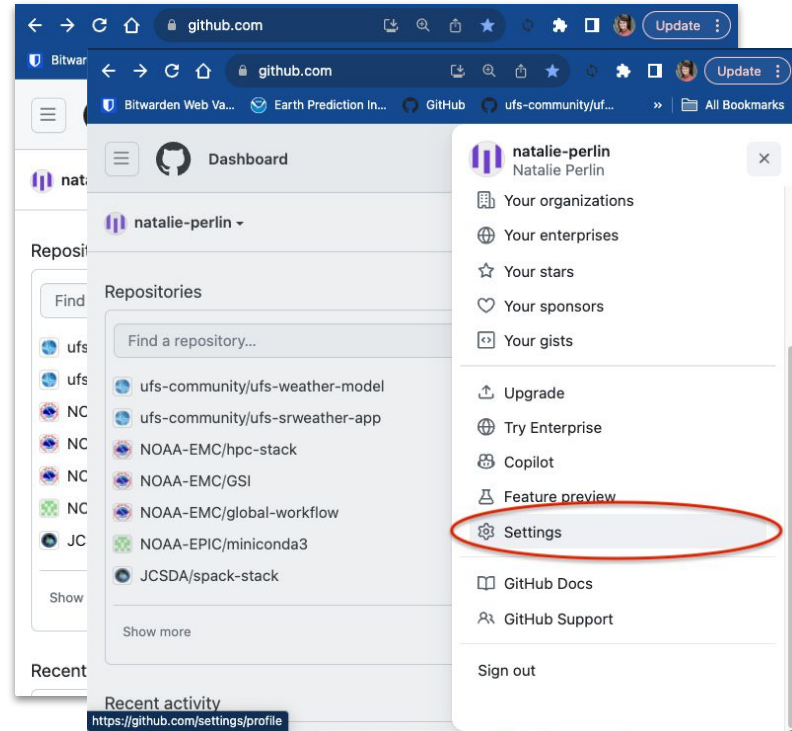
Creating a public GitHub profile

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



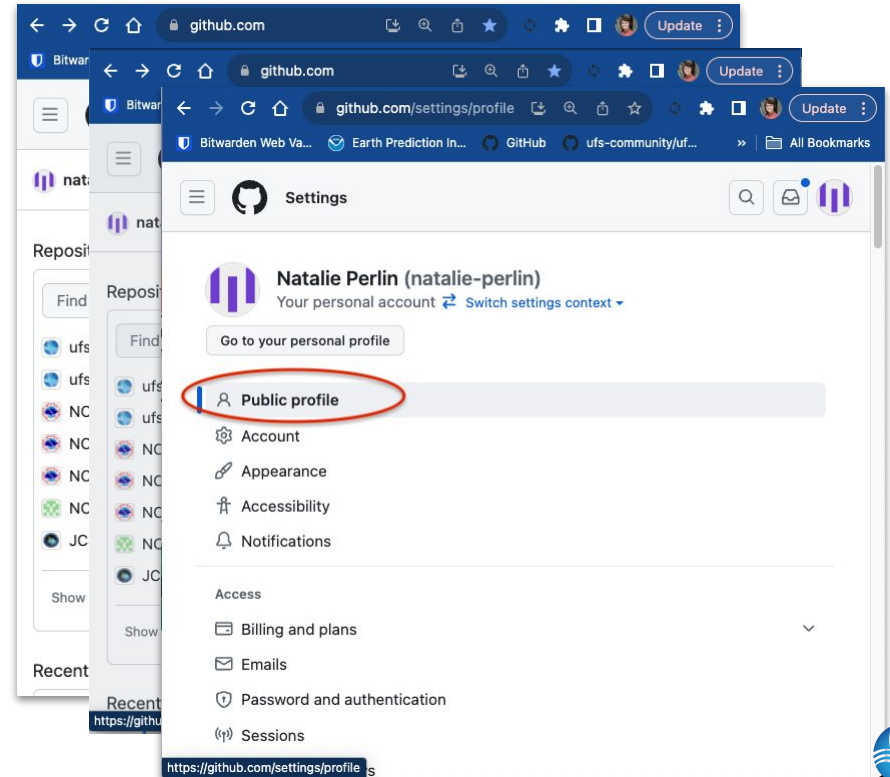
Creating a public GitHub profile

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



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](#)

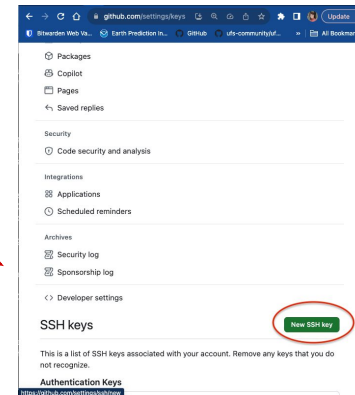
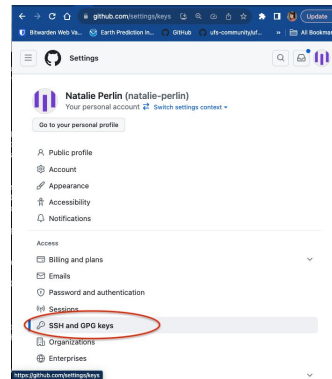


Generate SSH key pair for easy Git/GitHub communication

- 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>

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:AWDLDNF6QNAGGgD1T2d686e1sYnF76UYKeuKx40D9WI  Natalie@MacBook-Pro.local
The key's randomart image is:
+----[RSA 4096]-----+
|0*=oo..|
|.+.+.|. |
|..o=+. |
|.o+. |
|.oS |
|.+. |
|o+ .Bo. |
|.E.= %o |
|.+.+*+ |
+----[SHA256]-----+
[Natalie@Mac:~/ssh]$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQChGSs49zKLPOduAnn206s5nb58Y4CZXVc
53p5jm/zyb56BsavBI/j19Db2VcvaomkVHPF2Zox/W12SsD5oBJabIXgUsvh16Mv6DXHRsq
GD1zdhzDzJxfgdtuFha/KJCs97cjqwVz8Z5/u3aRvUsjn2ILRygnAyfE8xc64q4f1aB+mc8
D7LzKTtdZgUjrrrbQMxpBo16qouBXudiLF9k+eBoAsJaVGCABS660JnDs5BkSt1I405anuQ
XqV4PhV1FfNkzMHwOmRjPFBE1aq5c80XoYGuCgr5ac910iqVmaiY/76GRYrNgFLab19W6kJ
4npPLWbxxPtAGfmpRMIUNKUdeVqkZmxInaPV6xKiTvE8VxKkPFEBkPoF5cCTcXYz1paNRDS
SHFmhaWi1dBdrV/Mk6B8puIk+PErARBPOrdp9TkyV93oRQJ0/XMC02p1FJLxbI7MdXY4Yku
OtPgTEw1k0CsRcxkWTU6HFFhV1Fhbz/1j1RhZwV69cSXB15gp/h5zqEBkZ22QVDFY0TC9TE
7wQ5ZJePrSSScrxEqSmXjpPSxr88dS0RauHURw6eguIbgKgwXboY1xJfbGvu63px6pcdux
tCr4QwoTeS/Dax0AbmkBN91w3AVD+tCGpUfwrXU0CC+VqzWX7cwi4zV1RPEabrUaWv8Vaf
+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>

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:AWDLDNF6QNAGGgD1T2d686e1sYnF76UYKeuKx40D9WI  Natalie@MacBook-Pro.local
The key's randomart image is:
+--[RSA 4096]-----+
|0*=oo..|
|.+.+.|. |
|..o=+. |
|.o+. |
|.oS |
|.+. |
|o+ .Bo. |
|.E.= %o |
|.+.+.+. |
+-----[SHA256]-----+
[Natalie@Mac:~/ssh]$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQChGSs49zKLPOduAnn206s5nb58Y4CZXVc
53p5jm/zyb56BsavBI/j19Db2VcvaomKVHPF2Zox/W12SsD5oBJabIXgUsvh16Mv6DXHRsq
GD1zdhzDzJx fdqtuFha/KJCs97cjqwVz8Z5/u3aRvUsjn2ILRygnAyfE8xc64q4f1aB+mc8
D7LzKTtdZgUjrrrbQMxpBo16qouBXudiLF9k+eBoAsJaVGCABS660JnDs5BkSt1I405anuQ
XqV4PhV1FfNkzMHwOmRjPFBE1aq5c80XoYGuCgr5ac9I0iqVmaiY/76GRYrNgFLab19W6kJ
4npPLWbxxPtAGfmpRMIUNKUdeVqkZmxInaPV6xKiTvE8VxKkPFEBkPoF5cCTcXYZ1paNRDS
SHFmhaWi1dBdrV/Mk6B8puIk+PErARBpOrdp9TkyV93oRQJ0/XMC02p1FJLxbI7MdXY4Yku
OtPgTEw1k0CsRcxkWTU6HFFhV1Fhbz/1j1RhZwV69cSXB15gp/h5zqEBkZ22QVDFY0TC9TE
7wQ5ZJePrSSScrxEqSmXjpPSxr88dS0RauHURw6eguIbgKgwXboY1xJfbGvu63px6pcdux
tCr4QwoTeS/Dax0AbmkBN91w3AVD+tCGpUfwrXU0CC+VqzWX7cwi4zV1RPEabrUaWv8Vaf
+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>

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:AWDLDNF6QNAGGgD1T2d686e1sYnF76UYKeuKx40D9WI  Natalie@MacBook-Pro.local
The key's randomart image is:
+----[RSA 4096]-----+
|0*=oo..|
|.+.+. .|
|..o=.+.|
|.o+. .|
|.oS|
|.+. .|
|o+ .Bo.|
|.E.= %o|
|.+.+.+.|
+----[SHA256]-----+
[Natalie@Mac:~/ssh]$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQChGSs49zKLPOduAnn206s5nb58Y4CZXVc
53p5jm/zyb56BsavBI/j19Db2VcvaomKVHPF2Zox/W12SsD5oBJabIXgUsvh16Mv6DXHRsg
GD1zdhzDzJxfdqtuFha/KJCs97cjqwVz8Z5/u3aRvUsjn2ILRygnAyfE8xc64q4f1aB+mc8
D7LzKTtdZgUjrrrbQMxpBo16qouBXudiLF9k+eBoAsJavGCABS660JnDs5BkSt1I405anuQ
XqV4PhV1FfNkzMhWMrjPFBE1aq5c80XoYGuCgr5ac910iqVmaiY/76GRYrNgF1ab19W6kJ
4npPLWbxxPtAGfmpRMIUNKUdeVqkZmxInaPV6xKiTE8VxKkPFEBkPoF5cCTcXYZ1paNRDS
SHFmhaWi1dBdrV/Mk6B8puIk+PErARBpOrdp9TkyV93oRQJ0/XMC02p1FJLxbI7MdXY4Yku
OtPgTEw1k0CsRcxkWTU6HFFhV1Fhbz/1j1RhZwV69cSXB15gp/h5zqEBkZ22QVDFY0TC9TE
7wQ5ZJePrSSScrxEqSmXjpPSxr88dS0RauHURw6eguIbgKgwXboY1xJfbGvu63px6pcedux
tCr4QwoTeS/Dax0AbmkBN91w3AVD+tCGpUfwrXU0CC+VqzWX7cwi4zV1RPEabrUaWv8Vaf
+bMURKP7k3Nw==  Natalie@MacBook-Pro.local
[Natalie@Mac:~/ssh]$ █
```

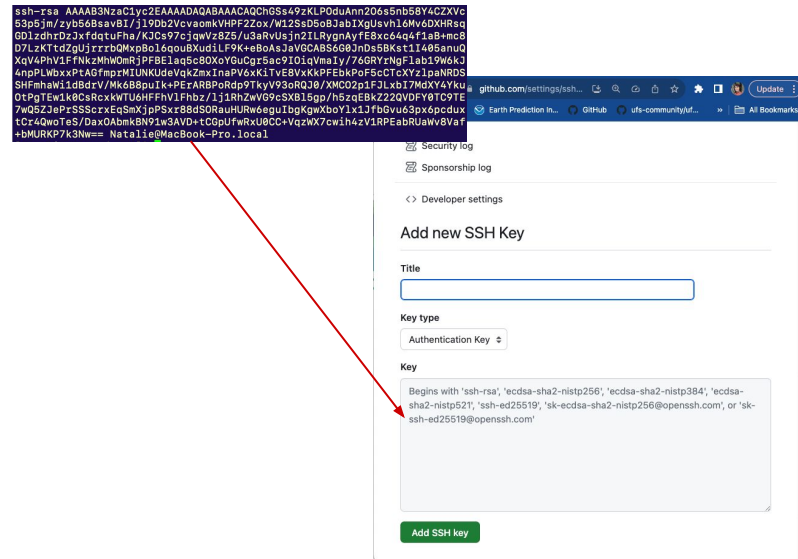
more help: <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

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 `~/.` 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>

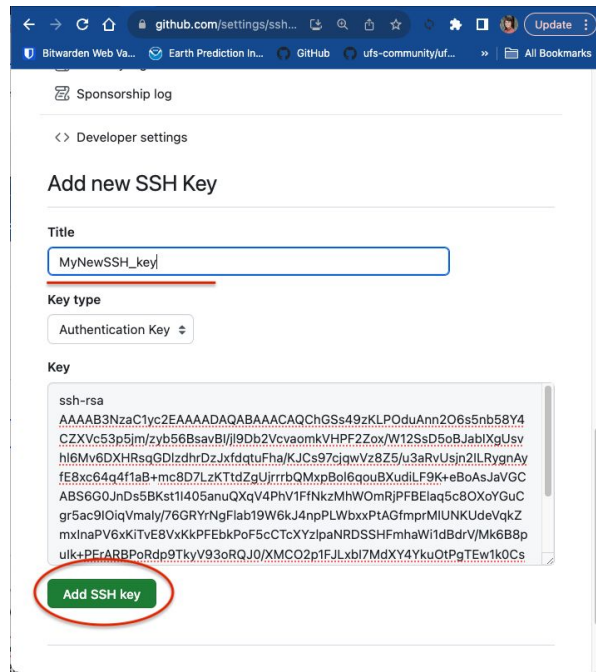


# Generate SSH key pair for easy Git/GitHub communication

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

## 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]$ █
```

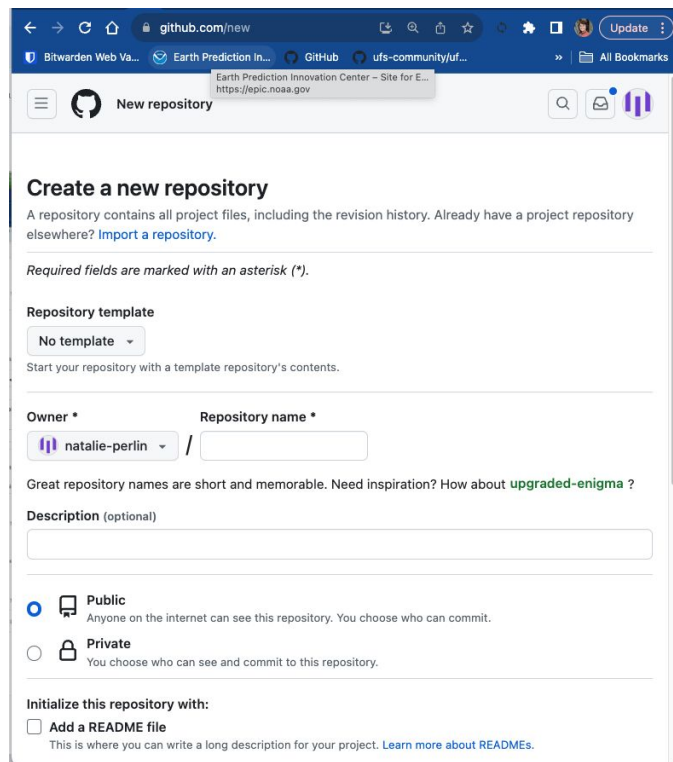
- Now pushes (uploads) to your GitHub repositories will not require a password

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>

# 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”



# 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 'New repository' page. The browser address bar is 'github.com/new'. The page title is 'New repository'. The main heading is 'Create a new repository'. Below this, there is a note: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'

Required fields are marked with an asterisk (\*).

**Repository template**  
No template (dropdown)  
Start your repository with a template repository's contents.

**Owner \*** natalie-perlin / **Repository name \*** my\_new\_repo  
my\_new\_repo is available (green checkmark)

Great repository names are short and memorable. Need inspiration? How about [special-train](#) ?

**Description (optional)**  
This is a new repository

**Initialize this repository with:**  
 **Public** Anyone on the internet can see this repository. You choose who can commit.  
 **Private** You choose who can see and commit to this repository.

**Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

# 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 'New repository' page. The browser address bar is [github.com/new](https://github.com/new). The page title is 'New repository'. The main heading is 'Create a new repository'. Below this, there is a note: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'

Required fields are marked with an asterisk (\*).

**Repository template**  
No template (dropdown)  
Start your repository with a template repository's contents.

**Owner \*** natalie-perlin / **Repository name \*** my\_new\_repo  
my\_new\_repo is available.

Great repository names are short and memorable. Need inspiration? How about [special-train](#) ?

**Description (optional)**  
This is a new repository

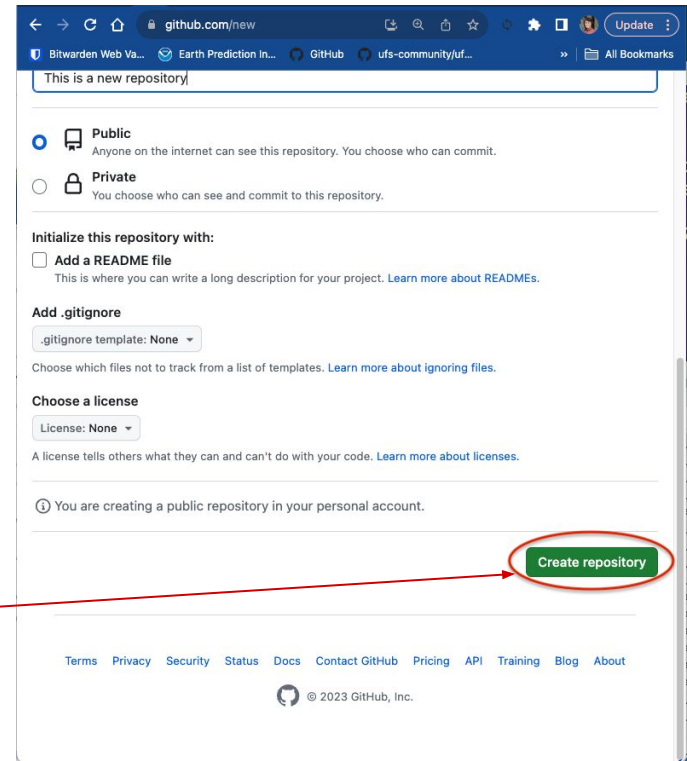
**Public**  
Anyone on the internet can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**  
 **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

# 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”



github.com/new

This is a new repository

**Public**  
Anyone on the internet can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**

**Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**  
.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**  
License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

ⓘ You are creating a public repository in your personal account.

**Create repository**

Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

© 2023 GitHub, Inc.

# 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

# 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)
```



# 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 visit
ing:
remote: https://github.com/natalie-perlin/my_new_repo/pull/n
ew/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 visit
ing:
remote: https://github.com/natalie-perlin/my_new_repo/pull/n
ew/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 visit
ing:
remote: https://github.com/natalie-perlin/my_new_repo/pull/r
ew/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

# 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”

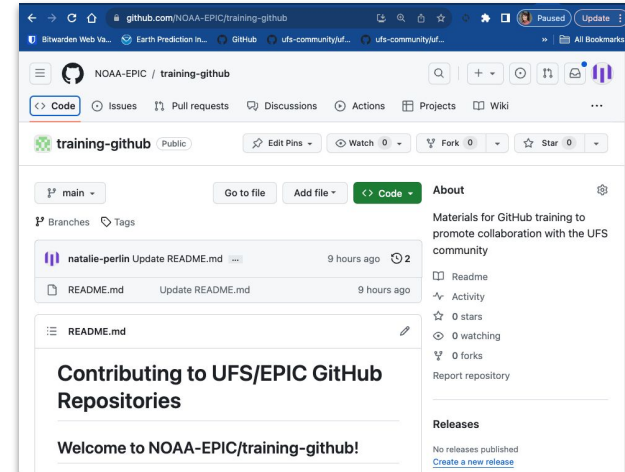
EARTH PREDICTION INNOVATION CENTER (EPIC)



# Navigating GitHub Repositories

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

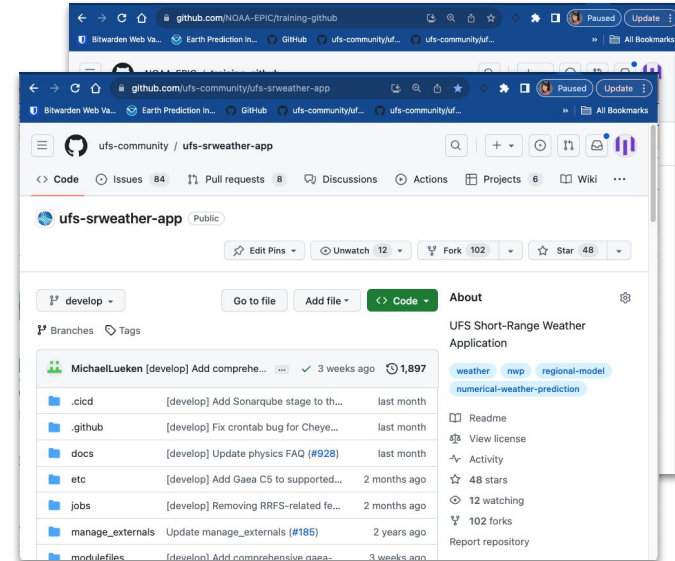
- **fork**: a completely independent copy of Git repository
- **clone**: a linked copy synchronized with the target/remote repo
- **Navigate to <https://github.com/NOAA-EPIC/training-github>**  
(or <https://github.com/ufs-community/ufs-srweather-app> )
- Look through the **Issues** and **Pull requests**
- 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”



# Navigate a GitHub Repository - Create a Fork

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

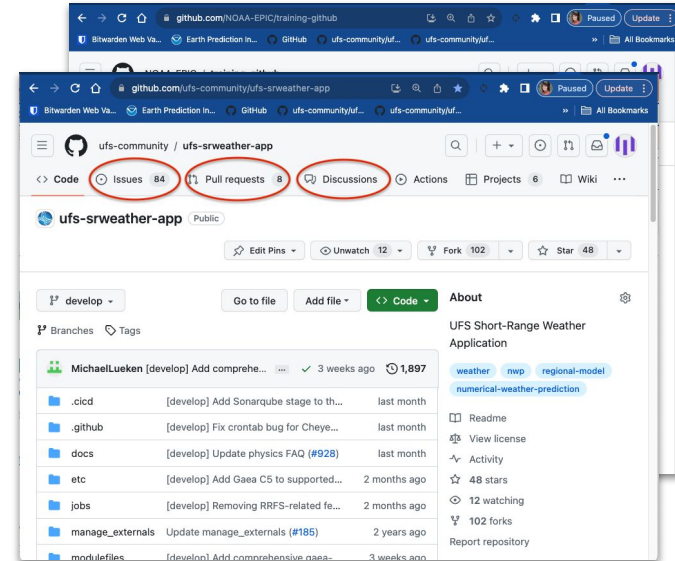
- **fork**: a completely independent copy of Git repository
- **clone**: a linked copy synchronized with the target/remote repo
- Navigate to <https://github.com/NOAA-EPIC/training-github> (or <https://github.com/ufs-community/ufs-srweather-app>)
- Look through the **Issues** and **Pull requests**
- 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”

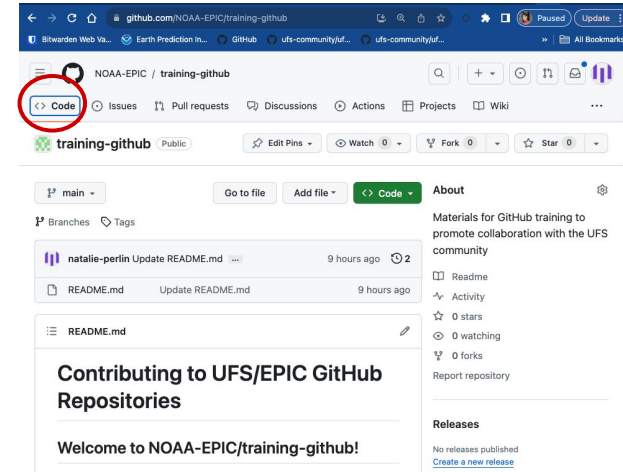




# Navigate a GitHub Repository - Create a Fork

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

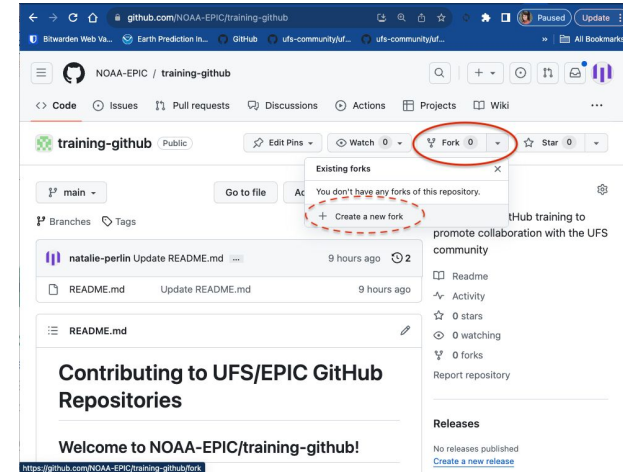
- **fork**: a completely independent copy of Git repository
- **clone**: a linked copy that remains synchronized with the target/remote repo
- Navigate to <https://github.com/NOAA-EPIC/training-github>
- Look through the **Issues** and **Pull requests**
- Hop onto the **Discussions** and reply to a thread
- **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"



# Navigate a GitHub Repository - Create a Fork

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

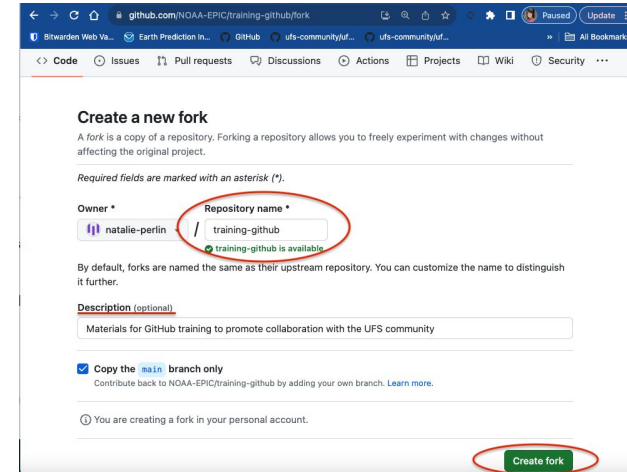
- **fork**: a completely independent copy of Git repository
- **clone**: a linked copy that remains synchronized with the target/remote repo
- Navigate to <https://github.com/NOAA-EPIC/training-github>
- Look through the **Issues** and **Pull requests**
- Hop onto the **Discussions** and reply to a thread
- 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”



# 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”



The screenshot shows the GitHub interface for creating a new fork. The browser address bar displays 'github.com/NOAA-EPIC/training-github/fork'. The page title is 'Create a new fork'. Below the title, there is a brief explanation: 'A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.' A note states 'Required fields are marked with an asterisk (\*)'. The 'Owner' field is set to 'natalie-perlin' and the 'Repository name' field is set to 'training-github'. A green checkmark indicates 'training-github is available'. The 'Description (optional)' field contains the text 'Materials for GitHub training to promote collaboration with the UFS community'. There is a checkbox for 'Copy the main branch only' which is checked. At the bottom right, the 'Create fork' button is highlighted with a red circle.

# 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”

