

## Section 9 : GitHub and remote repositories

What is Git hosting service?



GIT COMPLETE GUIDE



Git

vs

GitHub



Distributed version-control system

Repository hosting service

- You are able to create git repositories on your local computer and after that, upload them to GitHub hosting service. If you host your repository at GitHub and the repository is public, it means that any person in the world is able to clone or download your repository and use it locally on computer.
- Of course, you can create private repositories and in such case they will be visible only to you and probably to your collaborators.
- There are also other repository hosting services, for example, BitBucket, GitLab
- And when you host your git repositories using one of the git hosting services like GitLab, Bitbucket or GitHub, you actually create backup copy of your local git repository. If you occasionally delete your local git repository, you will be able to clone it again from remote repository and you won't lose any information.

→ Services like GitHub allow you to collaborate with other people and work on the same repositories and that is actually git feature and that's why git is called distributed version control system. Git allows you to perform simultaneous development in different branches in a single repository. You can even create your own hosting service and host your git repositories there, git itself fully supports that.

## Creating GitHub account

Now default branch for new repositories at GitHub is main

## Exploring first repository at GitHub

This is a method for pushing your project in GitHub by making some changes in our account setting and adding SSH key.

```
Last login: Sat Aug 21 20:56:42 on ttys001
→ ~ cd Desktop
→ Desktop mkdir firstcode
→ Desktop cd firstcode
→ firstcode echo "Hello, Git" > file.txt
→ firstcode ls
file.txt
→ firstcode cat file.txt
Hello, Git
→ firstcode git config --global user.name hema8Codes
→ firstcode git config --global user.email "developer8hema@gmail.com"
→ firstcode 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:
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 command:
hint:
hint: git branch -m <name>
Initialized empty Git repository in /Users/hemakshipandey/Desktop/firstcode/.git/
→ firstcode git:(master) x git add .
→ firstcode git:(master) x git commit -m "Initial Commit"
[master (root-commit) 7b242bf] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 file.txt
→ firstcode git:(master) git status
On branch master
nothing to commit, working tree clean
→ firstcode git:(master) git remote add origin git@github.com:hema8Codes/My-first-new-repository-.git
→ firstcode git:(master) ssh-keygen -t ed25519 -C "developer8hema@gmail.com"
Generating public/private ed25519 key pair.
```

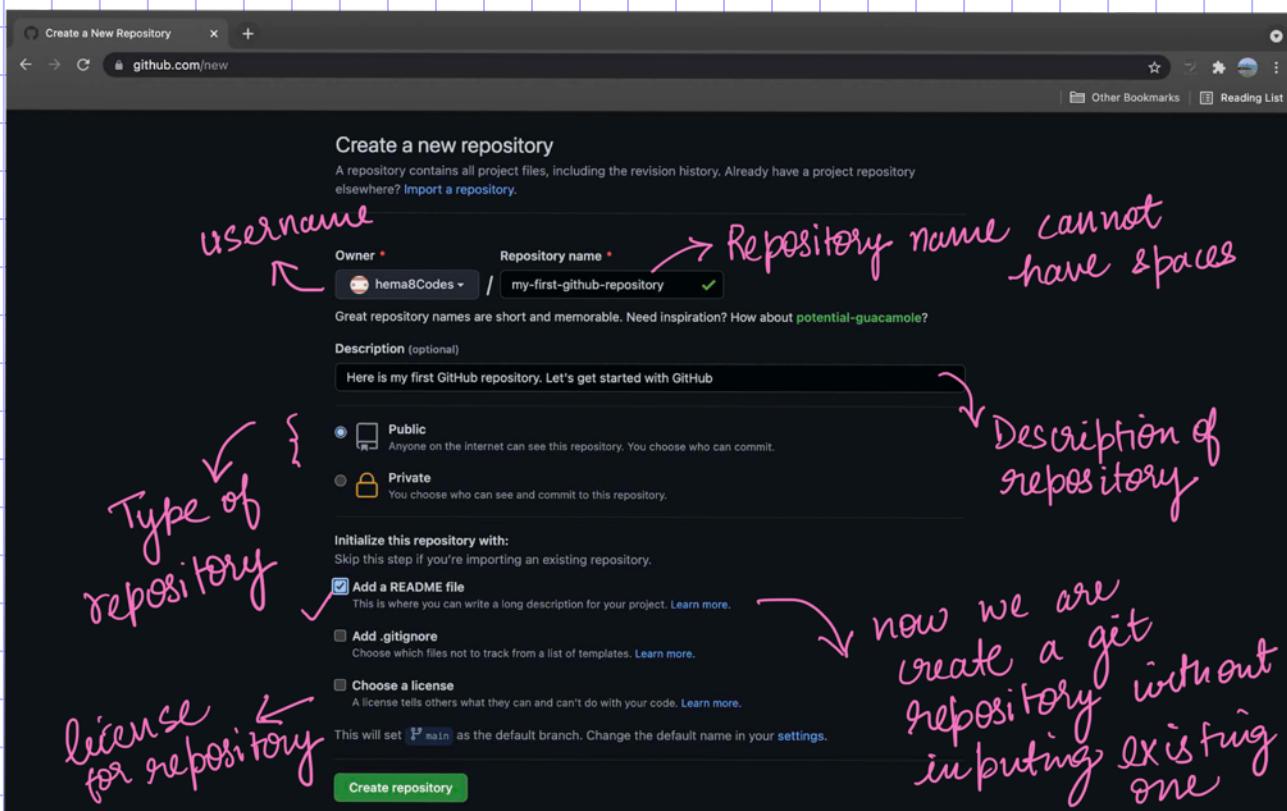
→ from CodeWithHarry

```

+ firstcode git:(master) git remote add origin git@github.com:hema8Codes/My-first-new-repository-.git
+ firstcode git:(master) ssh-keygen -t ed25519 -C "developer8hema@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/Users/hemakshipandey/.ssh/id_ed25519): /Users/hemakshipandey/.ssh/id_ed25519
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/hemakshipandey/.ssh/id_ed25519.
Your public key has been saved in /Users/hemakshipandey/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:Ls2G7yY6DvFFVuWpd4FdGnLcEl1ZAgb08TfzHBe5oKo developer8hema@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|       .o+...o|
|       o + o+|
|       o o==l|
|       . . =B|
|   S . o . +|
| * o o . .=.|
| . . o 0 o + B=|
| o...E. + + = |
| .oo.+o . . . |
+---[SHA256]---+
+ firstcode git:(master) cat /Users/hemakshipandey/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIA0tNDa0QS8HX0f96Tybm5UGIKKjagJel0EG5+pABVD developer8hema@gmail.com
+ firstcode git:(master) git push origin master
Warning: Permanently added the RSA host key for IP address '13.234.210.38' to the list of known hosts.
Enter passphrase for key '/Users/hemakshipandey/.ssh/id_ed25519':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 226 bytes | 226.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:hema8Codes/My-first-new-repository-.git
 * [new branch]      master -> master
+ firstcode

```

→ SSH Key is generated add to your github settings account.



permanent link to your repository

name of repository

first repository was successfully created

single branch

contributor

message of the commit

SHA1 has of this commit

Single commit

name of repository my-first-github-repository

Here is my first GitHub repository. Let's get started with GitHub

Our working directory contains single file

contents of (description) readme.md file

GitHub has automatically created contents of the README.md file.

After clicking on single commit

Commits - hema8Codes/my-first-github-repository

Initial commit

hema8Codes committed 1 minute ago

Aug 24, 2021

Verified

398b48e

Newer Older

github has added commit signature

name of the commit

time

author of the commit

SHA1 hash of this commit object

you are able to browse at specific point in history

Commits - hema8Codes/my-first-github-repository

Initial commit

hema8Codes committed 1 minute ago

Aug 24, 2021

Verified

398b48e

Newer Older

After clicking on single branch

A screenshot of the GitHub 'Branches' page for the repository 'hema8Codes/my-first-github-repository'. The main heading is 'Branches'. The top navigation bar includes 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation is a search bar and a list of branches. The first branch listed is 'main', which is highlighted with a pink circle and a handwritten note 'main Updated 2 minutes ago by hema8Codes'. A pink arrow points from this note to the branch name. To the right of the branch list is a button labeled 'Default'. A pink arrow points from this button to the word 'default' in a handwritten note. The page footer contains links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', 'Settings', and various GitHub services like 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About'.

Right now our working directory contains single file README.md

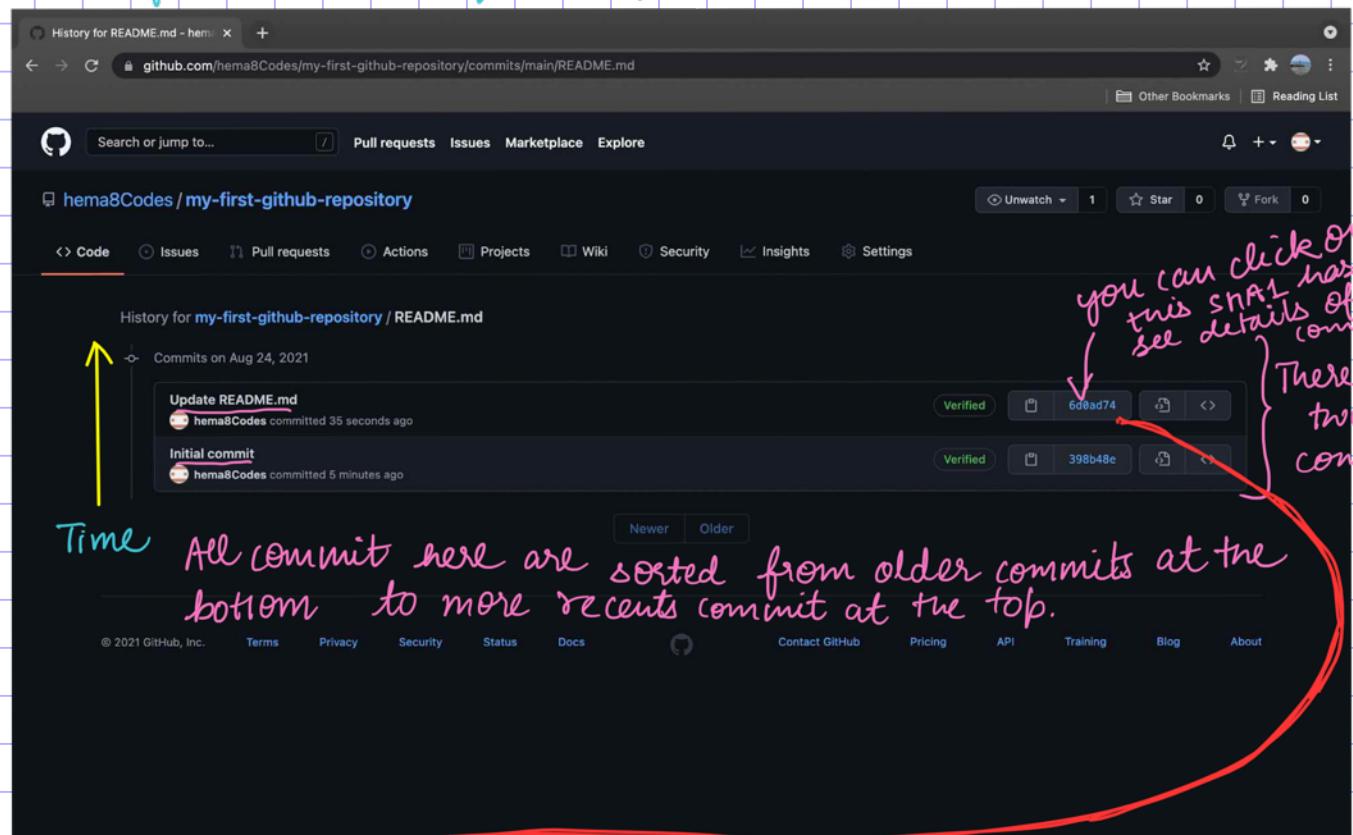
editing readme.md file .

A screenshot of the GitHub 'Edit file' interface for the 'README.md' file in the 'main' branch of the repository 'hema8Codes/my-first-github-repository'. The title bar shows 'Editing my-first-github-repository / README.md in main'. The editor displays the following content:

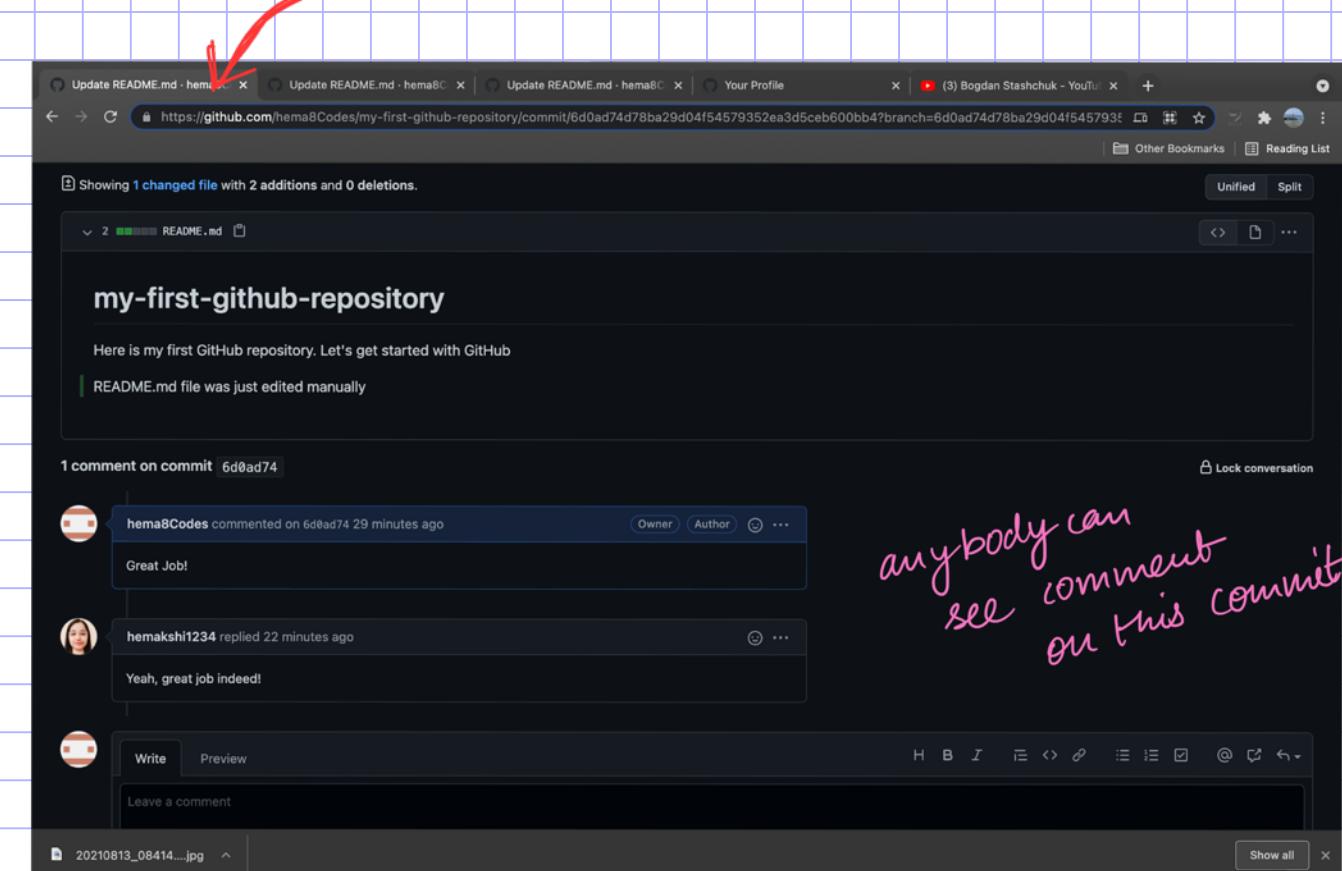
```
1 # my-first-github-repository
2 Here is my first GitHub repository. Let's get started with GitHub
3
4 README.md file was just edited manually
```

A handwritten note 'adding new line' points to the fourth line of the file. Below the editor is a 'Commit changes' dialog box. It contains a 'Commit changes' button at the bottom left and a 'Cancel' button at the bottom right. The dialog also includes fields for 'Update README.md' and 'Add an optional extended description...'. At the bottom, there are two radio button options: one selected for 'Commit directly to the main branch.' with a checkmark, and another for 'Create a new branch for this commit and start a pull request.' A handwritten note 'default commit message' points to the first option. Another handwritten note 'committing changes' points to the 'Commit changes' button.

After committing changes for README.md file



A screenshot of a GitHub commit history page for the file README.md. The commits are listed in reverse chronological order, with the most recent at the top. The first commit is titled "Initial commit" and was made 5 minutes ago by "hema8Codes". The second commit is titled "Update README.md" and was made 35 seconds ago by "hema8Codes". The page includes navigation buttons for "Newer" and "Older" commits. A yellow arrow on the left side points upwards, indicating the direction of time from bottom to top. Handwritten notes in red ink are overlaid on the page: "you can click on this SHA1 hash to see details of this commit object" pointing to the commit hash "6d0ad74", and "There are two commit" enclosed in a red bracket next to the commit list.



A screenshot of a GitHub commit details page for the commit "6d0ad74". The commit message is "Update README.md". The commit has 2 additions and 0 deletions, affecting the file README.md. The commit was made 29 minutes ago by "hema8Codes" and has 1 comment. The first comment is from "hema8Codes" saying "Great Job!". The second comment is from "hemakshi1234" replying "Yeah, great job indeed!". A red arrow points from the top of the commit message area down to the comment section. Handwritten notes in red ink say "anybody can see comment on this commit" pointing to the comment section.

# Creating another commit at GitHub

A screenshot of a GitHub repository page for 'hema8Codes/my-first-github-repository'. The repository has 1 branch and 0 tags. The main branch contains a file named 'README.md' which has been updated. A new file, '20210813\_08414....jpg', is being uploaded. The 'Code' dropdown menu is open, with 'Create new file' highlighted. Handwritten notes include 'our current branch' pointing to the branch count, and 'creating new file' pointing to the 'Create new file' button.

A screenshot of the same GitHub repository page after a new commit. The main branch now contains two files: 'README.md' and 'hello-github.txt'. The 'hello-github.txt' file was created by 'hema8Codes' at '3a3e202 now'. Handwritten notes include 'head points to master branch' pointing to the branch dropdown, 'main branch points to very last commit' pointing to the commit history, 'contents of working directory' pointing to the file list, and 'now there are two file' pointing to the file list.

Commits - hema8Codes/my-first-github-repository

github.com/hema8Codes/my-first-github-repository/commits/main

Search or jump to... Pull requests Issues Marketplace Explore

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main branch points to very last commit in this list

SHA 1 hash of this commit

last commit

Create hello-github.txt  
hema8Codes committed 1 minute ago

Update README.md  
hema8Codes committed 3 hours ago

Initial commit  
hema8Codes committed 3 hours ago

Newer Older

20210813\_08414....jpg

This screenshot shows the GitHub repository 'my-first-github-repository' with the 'Code' tab selected. The 'main' branch is highlighted. A handwritten note above the commits says 'main branch points to very last commit in this list'. A blue arrow points from the branch name to the commit list. Another handwritten note to the right says 'SHA 1 hash of this commit' with an arrow pointing to the commit's SHA-1 value '3a3e202a8ffc7361245fd8416e496d423e697a37'. Below the commits, a file named '20210813\_08414....jpg' is listed.

Create hello-github.txt - hema8Codes

github.com/hema8Codes/my-first-github-repository/commit/3a3e202a8ffc7361245fd8416e496d423e697a37

Search or jump to... Pull requests Issues Marketplace Explore

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

content of commit object in get repository

main branch

author

Showing 1 changed file with 1 addition and 0 deletions.

hello-github.txt

parent commit for this commit

Unified Split

1 parent 6d0ad74 commit 3a3e202a8ffc7361245fd8416e496d423e697a37

0 comments on commit 3a3e202

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

20210813\_08414....jpg

This screenshot shows the GitHub commit page for the commit '3a3e202a8ffc7361245fd8416e496d423e697a37'. A handwritten note at the top says 'content of commit object in get repository'. Another note says 'main branch' with an arrow pointing to the branch name. A note labeled 'author' points to the author information. A note labeled 'parent commit for this commit' points to the parent commit '6d0ad74'. The commit details show it was created by 'hema8Codes' 1 minute ago and has 1 parent commit. The commit message is 'Create hello-github.txt'. The commit content shows a single addition to 'hello-github.txt'. A comment section is shown below, with a note 'Unified Split' pointing to the unified diff link.

Checking out first initial commit

SHA1 hash

"tree" in the GitHub URL doesn't represent Git tree object. Instead it is just pointer to specific branch or commit

Code

398b48e4ec 1 branch 0 tags

hema8Codes Initial commit 398b48e 3 hours ago 1 commit

README.md Initial commit 3 hours ago

README.md

my-first-github-repository

Here is my first GitHub repository. Let's get started with GitHub

About

Here is my first GitHub repository. Let's get started with GitHub

Readme

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

20210813\_08414....jpg Show all

Creating new branch at GitHub

Single branch

main 1 branch 0 tags

hema8Codes Create hello-github.txt 3a3e202 13 minutes ago 3 commits

README.md Update README.md 3 hours ago

hello-github.txt Create hello-github.txt 13 minutes ago

README.md

my-first-github-repository

Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually

About

Here is my first GitHub repository. Let's get started with GitHub

Readme

Releases

No releases published Create a new release

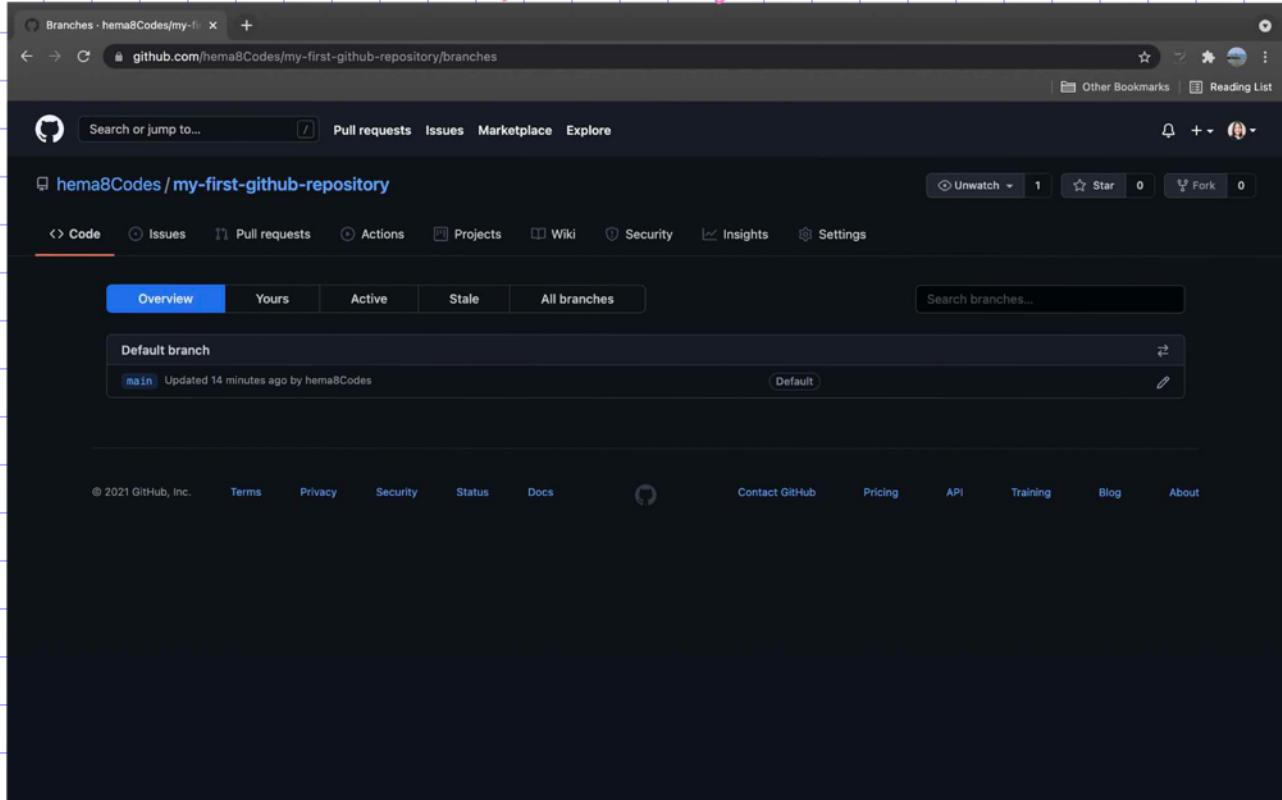
Packages

No packages published Publish your first package

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

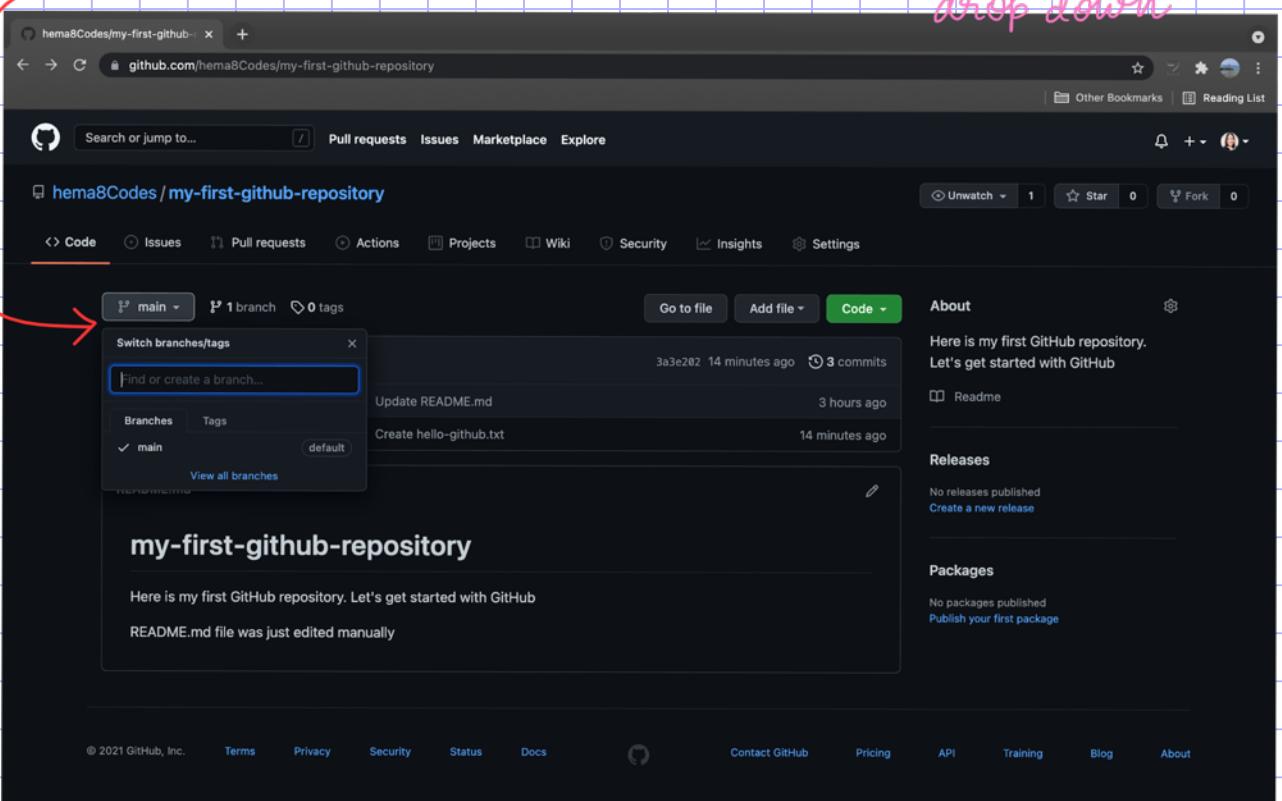
https://github.com/hema8Codes/my-first-github-repository/branches

After clicking on single branch



A screenshot of a GitHub repository's branches page. The URL in the address bar is `github.com/hema8Codes/my-first-github-repository/branches`. The main navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. Below that is a secondary navigation bar with Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is selected and highlighted in blue. A sub-navigation bar below it shows Overview (selected), Yours, Active, Stale, and All branches. A search bar for branches is present. The main content area displays a section titled 'Default branch' with a single entry: 'main Updated 14 minutes ago by hema8Codes'. There is also a small edit icon next to the branch name. The footer contains standard GitHub links like Contact GitHub, Pricing, API, Training, Blog, and About.

In order to create new branch you need to click on this drop down



A screenshot of the same GitHub repository page, but with a red arrow pointing to the 'main' dropdown menu in the top left corner of the main content area. This dropdown menu is open, showing the current branch ('main'), the number of branches ('1 branch'), and the number of tags ('0 tags'). It also contains a search bar labeled 'Find or create a branch...' and buttons for 'Switch branches/tags' and 'View all branches'. The main content area shows the repository's README and recent commits. To the right, there are sections for About, Releases, and Packages, each with a brief description and a 'Create a new [something]' link.

name of branch

click here to create new branch.

And this branch will be created from master branch

my-first-github-repository

Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

This screenshot shows the GitHub repository page for "my-first-github-repository". A red arrow points to the "feature-1" branch in the dropdown menu. Handwritten notes explain that this is the "name of branch" and "click here to create new branch." Another note states "And this branch will be created from master branch". The repository has 1 branch and 0 tags.

new feature-1 branch was created

Branch created.

And immediately after creation, this new branch was automatically checked out

That means master branch and feature branch points to the same commit object

master feature-1

hema8Codes / my-first-github-repository

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Switch branches or tags

feature-1 2 branches 0 tags

hema8Codes Create hello-github.txt

README.md Update README.md 3 hours ago

hello-github.txt Create hello-github.txt 15 minutes ago

README.md

my-first-github-repository

Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually

About

Here is my first GitHub repository. Let's get started with GitHub

Readme

Releases

Packages

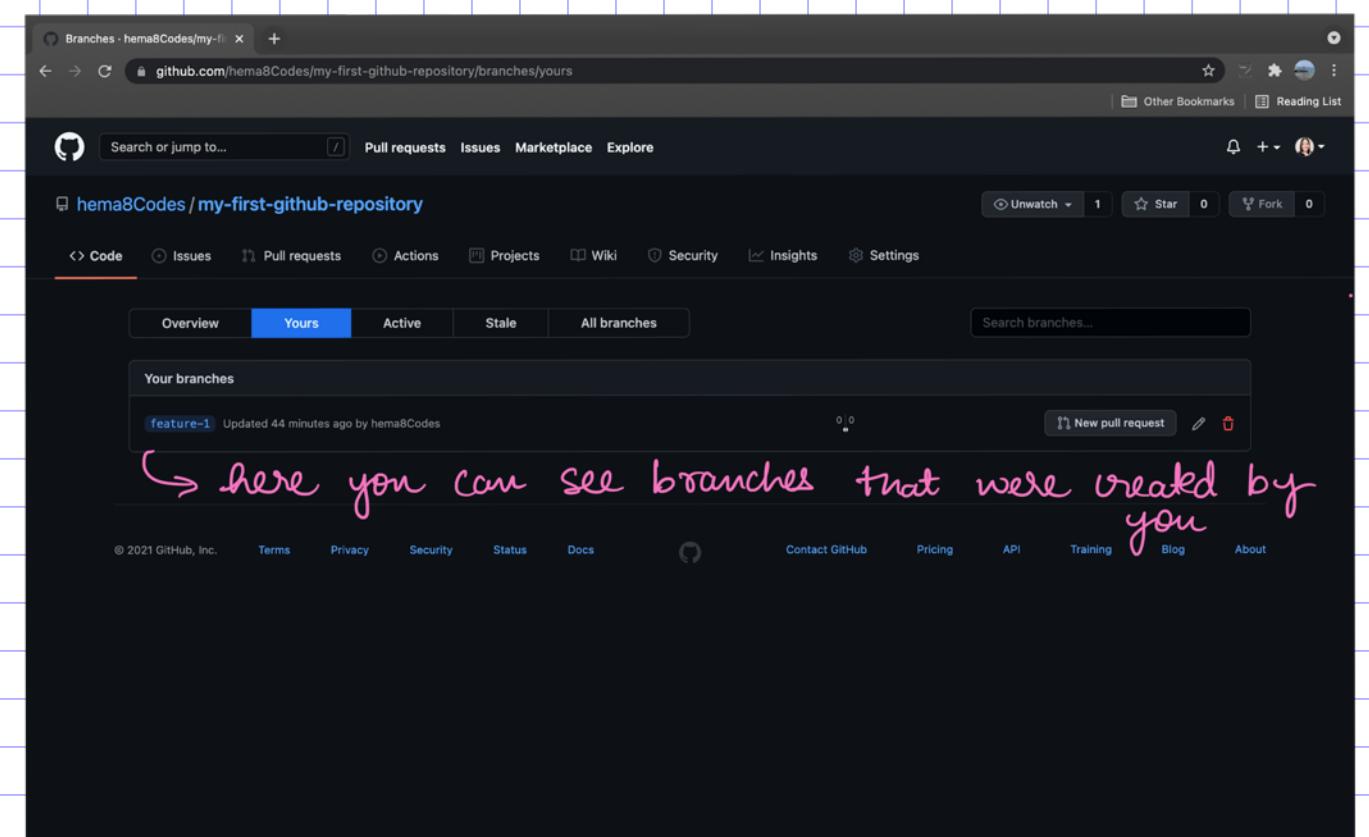
No packages published Publish your first package

This screenshot shows the repository tree for the "feature-1" branch. It lists files: "Create hello-github.txt", "README.md", and "hello-github.txt". Handwritten notes state "new feature-1 branch was created" and "And immediately after creation, this new branch was automatically checked out". A diagram shows two boxes labeled "master" and "feature-1" with arrows pointing down to a single commit object. The "feature-1" branch is highlighted in green.

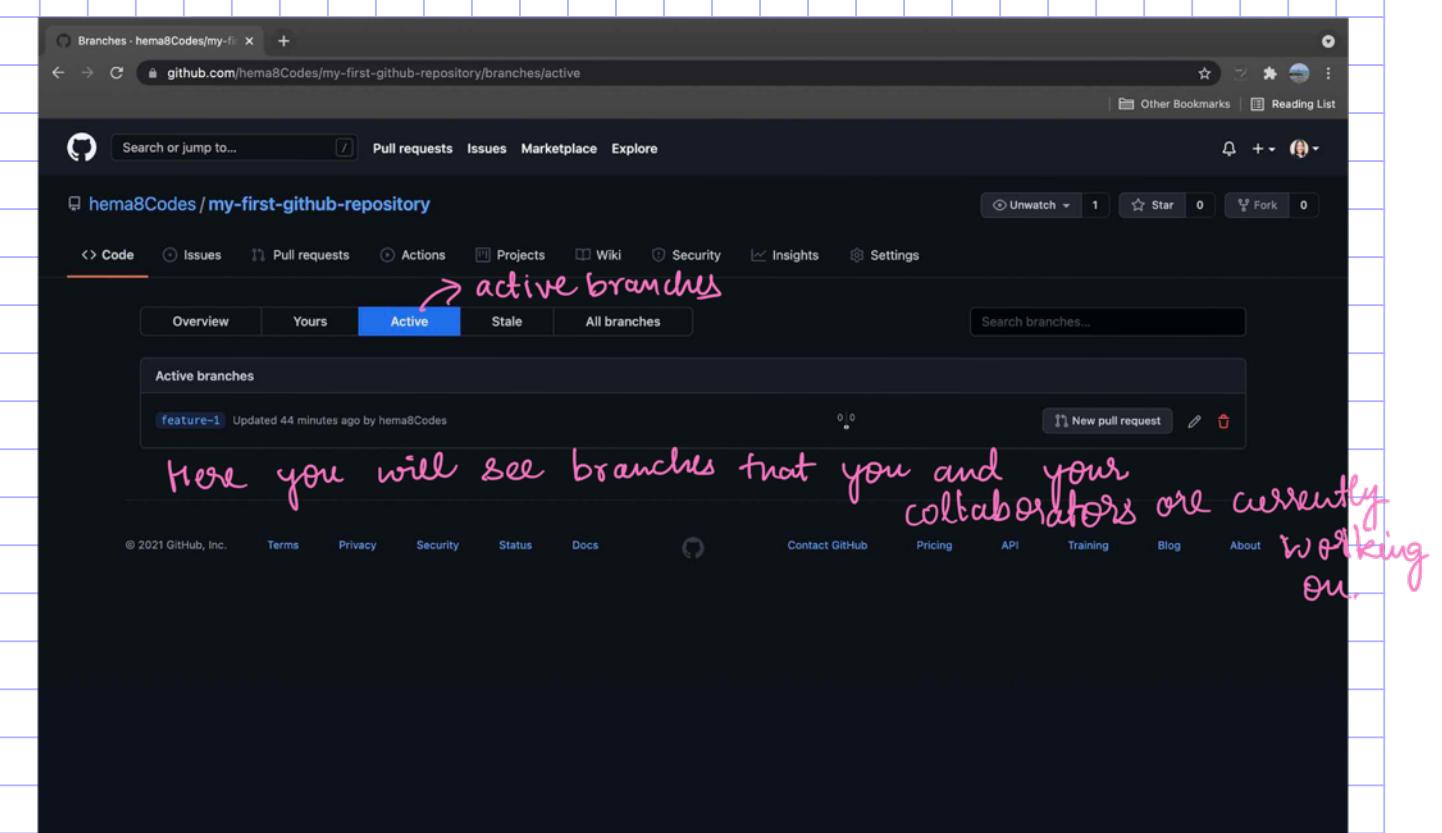
# Making changes in the new branch

A screenshot of a GitHub repository page. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the navigation is a search bar and a sidebar with options for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area shows a list of files: README.md, Update README.md, and hello-github.txt. A message says "This branch is even with main." Handwritten notes on the screen say "we have created two branches" pointing to the branch dropdown, and "feature-1 branch" and "master branch" pointing to the branches listed in the dropdown. A red arrow on the left points to the "feature-1" link in the dropdown. A blue URL bar at the bottom contains the URL <https://github.com/hema8Codes/my-first-github-repository/branches>.

A screenshot of the GitHub branches page for the same repository. It shows a list of branches under sections: Default branch, Your branches, and Active branches. The "feature-1" branch is selected, indicated by a checkmark and a pink border. Handwritten notes on the screen show a checkmark next to the "feature-1" entry in the "Your branches" section. The footer of the page includes links for Contact GitHub, Pricing, API, Training, Blog, and About.



A screenshot of the GitHub 'Branches' page for the repository 'hema8Codes/my-first-github-repository'. The URL is [github.com/hema8Codes/my-first-github-repository/branches/yours](https://github.com/hema8Codes/my-first-github-repository/branches/yours). The page shows a single branch named 'feature-1' with a status of 'Updated 44 minutes ago by hema8Codes'. Navigation tabs at the top include 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. A hand-drawn annotation in blue ink points to the 'feature-1' branch with the text: 'here you can see branches that were created by you'.



A screenshot of the GitHub 'Active branches' page for the same repository. The URL is [github.com/hema8Codes/my-first-github-repository/branches/active](https://github.com/hema8Codes/my-first-github-repository/branches/active). It shows the same 'feature-1' branch. A hand-drawn annotation in blue ink points to the 'Active' tab with the text: 'active branches'. Another annotation in pink ink points to the 'feature-1' branch with the text: 'Here you will see branches that you and your collaborators are currently working on'.

A screenshot of a web browser displaying the GitHub interface for a repository named "hema8Codes/my-first-github-repository". The user is on the "Code" tab, specifically the "Stale" sub-tab. A search bar at the top right shows "Search branches...". Below the tabs, a section titled "Stale branches" displays the message "There aren't any stale branches." Handwritten notes in pink ink overlay the page, stating: "If some branches were not updated for a long time, they will be moved to stale branches".

A screenshot of a web browser displaying the GitHub interface for the same repository. The user is on the "Code" tab, specifically the "All branches" sub-tab. A search bar at the top right shows "Search branches...". Below the tabs, a section titled "All branches" lists two branches: "main" and "feature-1", both updated 44 minutes ago by "hema8Codes". A red arrow points from the handwritten note "you can list all branches by clicking here on this all branch list" to the "All branches" tab. Handwritten notes in pink ink also appear on the right side of the screen.

→ Updating feature-1 branch

Creating new file

Code

Create new file

Upload files

This branch is even with main.

hema8Codes Create hello-github.txt

README.md Update README.md

hello-github.txt Create hello-github.txt

README.md

my-first-github-repository

Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually

New File

github.com/hema8Codes/my-first-github-repository/new/feature-1

Search or jump to...

Code

my-first-github-repository / feature in feature-1

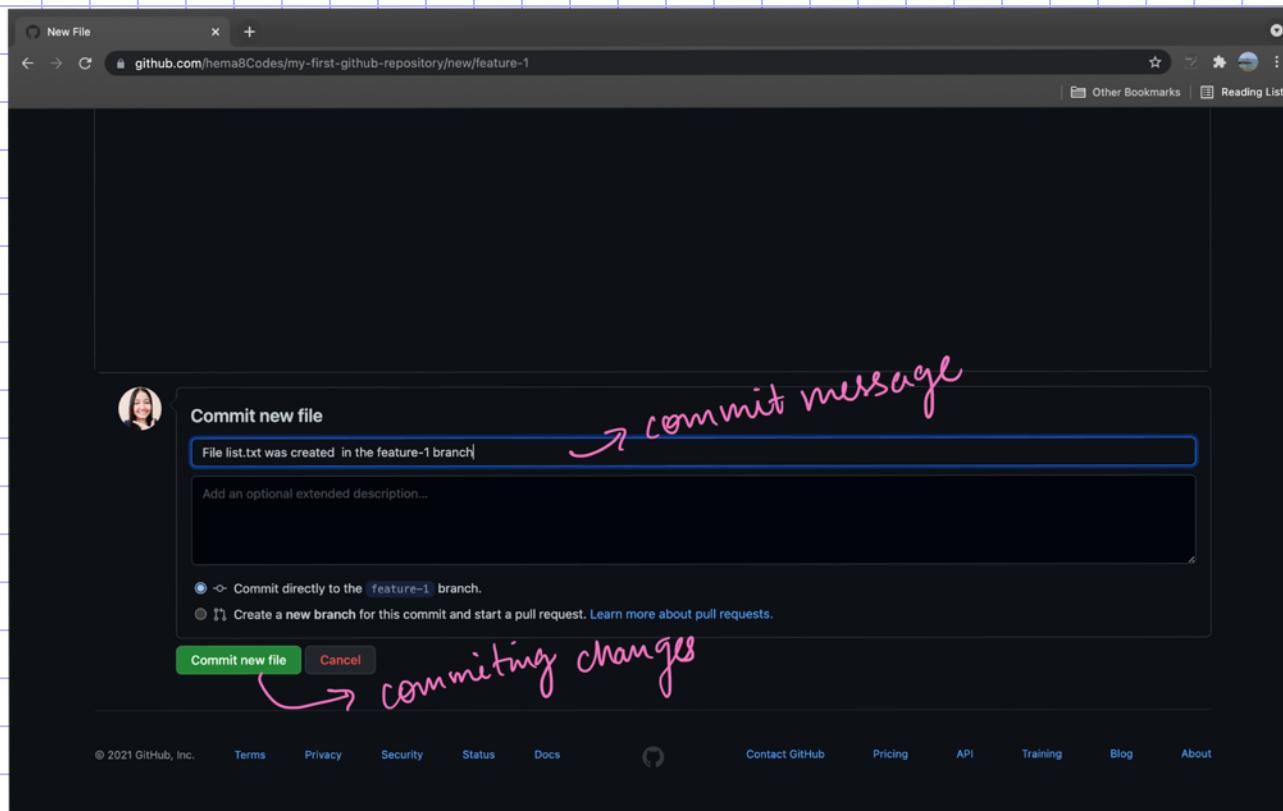
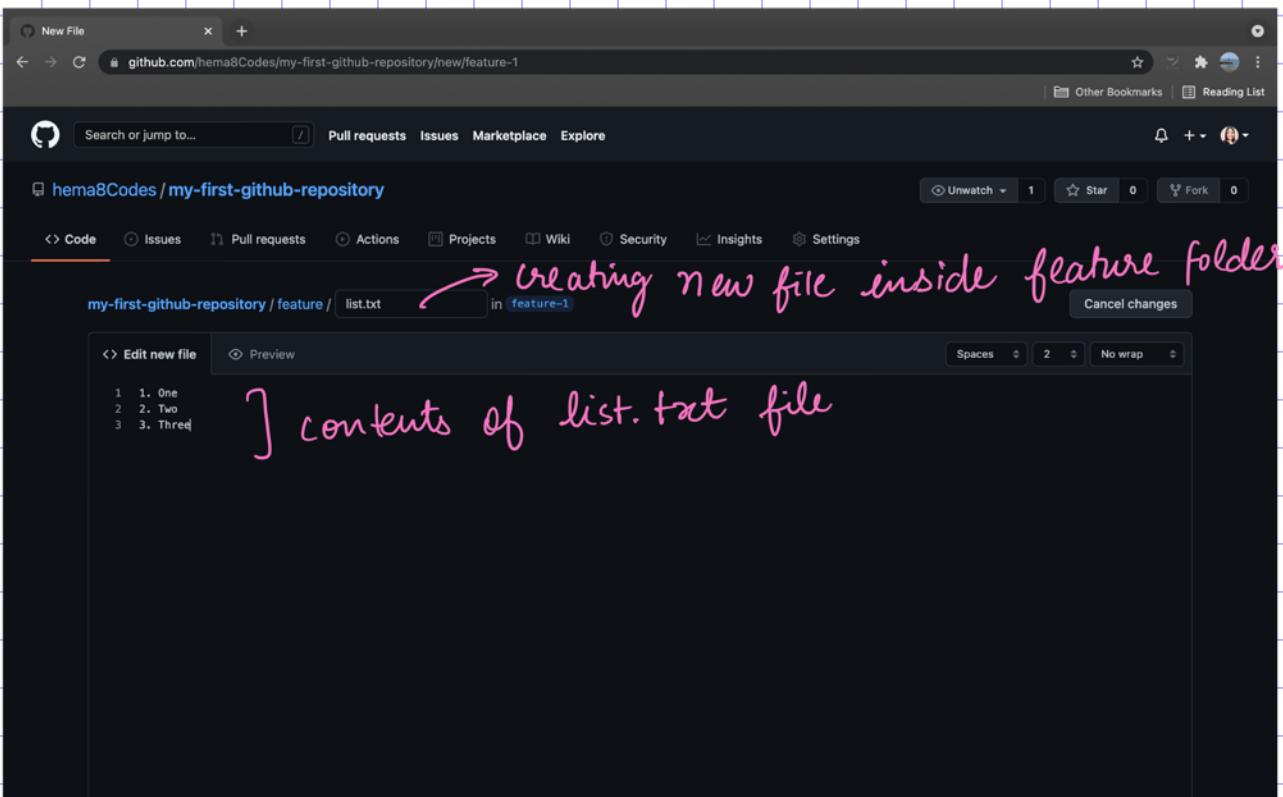
Cancel changes

Edit new file Preview

Spaces 2 No wrap

creating folder

→ put "/" after feature



This screenshot shows a GitHub repository page for 'my-first-github-repository'. The 'feature' branch is selected. A handwritten note in pink ink states: 'This means that changes in feature branch were yet not incorporated in master branch'. Another note says: 'contents of feature folder'.

Code

feature-1

This branch is 1 commit ahead of main.

hema8Codes File list.txt was created in the feature-1 branch

list.txt

File list.txt was created in the feature-1 branch

Now

This screenshot shows a GitHub repository page for 'my-first-github-repository'. The 'feature-1' branch has recent pushes. A callout points to the 'Compare & pull request' button with the text: 'you are able to compare branches and create new pull request'.

feature-1 had recent pushes less than a minute ago

Compare & pull request

feature-1

2 branches

0 tags

This branch is 1 commit ahead of main.

hema8Codes File list.txt was created in the feature-1 branch

feature

README.md

hello-github.txt

File list.txt was created in the feature-1 branch

Update README.md

Create hello-github.txt

26 seconds ago

4 commits

26 seconds ago

4 hours ago

1 hour ago

About

Here is my first GitHub repository. Let's get started with GitHub

Readme

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

my-first-github-repository

Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually

# Cloning remote repository

hema8Codes/my-first-github-repository

This branch is 1 commit ahead of main.

File list.txt was created in the feature-1 branch

feature feature-1 2 branches 0 tags

Go to file Add file + Code

Clone

HTTPS SSH GitHub CLI (New)

<https://github.com/hema8Codes/my-first>

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

About

Here is my first GitHub repository.  
Let's get started with GitHub

Readme

Releases

No releases published  
Create a new release

Packages

No packages published  
Publish your first package

my-first-github-repository

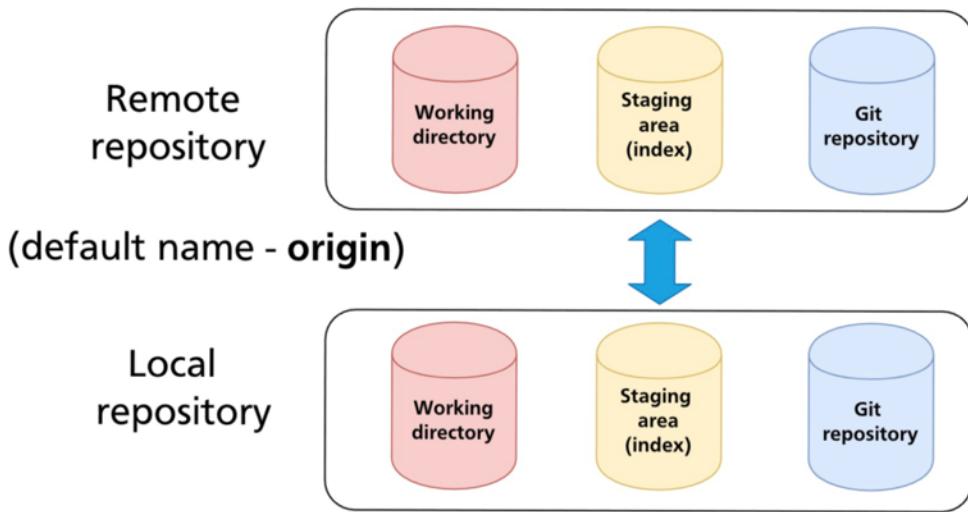
Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually

```
→ Desktop ls -l
total 0
lrwxr-xr-x@ 1 hemakshipandey staff    29 Aug 10 19:26 Relocated Items.nosync -> /Users/Shared/Relocated Items
drwxr-xr-x@ 8 hemakshipandey staff   256 Aug 20 23:00 first-project
drwxr-xr-x@ 38 hemakshipandey staff  1216 Aug 19 21:38 hello-world
→ Desktop git clone https://github.com/hema8Codes/my-first-github-repository.git
Cloning into 'my-first-github-repository'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 13 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (1/1), done.
→ Desktop ls -l
total 3128
lrwxr-xr-x@ 1 hemakshipandey staff    29 Aug 10 19:26 Relocated Items.nosync -> /Users/Shared/Relocated Items
-rw-r--r--@ 1 hemakshipandey staff 1600145 Aug 24 17:31 Screenshot 2021-08-24 at 5.31.21 PM.png
drwxr-xr-x@ 8 hemakshipandey staff   256 Aug 20 23:00 first-project
drwxr-xr-x@ 38 hemakshipandey staff  1216 Aug 19 21:38 hello-world
drwxr-xr-x@ 5 hemakshipandey staff    160 Aug 24 18:37 my-first-github-repository
→ Desktop cd my-first-github-repository
→ my-first-github-repository git:(main) ls -la
total 16
drwxr-xr-x@ 5 hemakshipandey staff  160 Aug 24 18:37 .
drwxr-xr-x@ 9 hemakshipandey staff  288 Aug 24 18:37 ..
drwxr-xr-x@ 12 hemakshipandey staff  384 Aug 24 18:37 .git
-rw-r--r-- 1 hemakshipandey staff  136 Aug 24 18:37 README.md
-rw-r--r-- 1 hemakshipandey staff   13 Aug 24 18:37 hello-github.txt
→ my-first-github-repository git:(main) cat README.md
# my-first-github-repository
Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually
→ my-first-github-repository git:(main) cat hello-github.txt
Hello GitHub
→ my-first-github-repository git:(main) 
```

# What is remote repository



For the moment our remote repository is connected to local repository and now we can interact with remote repository. Notice that remote repository is actually the same as a local repository in terms of structure. There is working directory, staging area and the git repository, same as in your local computer.

But now when you have two repositories, a remote repository and local repository, there are multiple questions that should be answered.

- Q 1. What happens when you update your local repository, does remote repository update itself automatically?  
No, you need to push your changes from local repository to remote repository.
- Q 2. What happens when your remote repository updates? Does your local repository update itself automatically?  
No, you need to pull changes. You need to download all updates manually from a remote repository. If you use Git GUI applications like GitHub Desktop, SourceTree or VSCode may fetch updates from the remote repository in background but you need to pull them manually. You are able to semi-automate this process. What does it mean? You may see which changes were made

in remote repository and after that you can decide whether you want to download those changes or not.

Q. 3. Who has rights to update your remote repository?  
(make changes locally on computer and after that push changes to your remote repository)

Nobody is allowed to update your remote repository. You need to grant access for writing to your remote repository.

→ Remote repository owner must grant write access.

Q. 4. Who has rights to read (download or clone) remote repository?

→ Everyone if remote repository is public.

→ Selected people if remote repository is private.

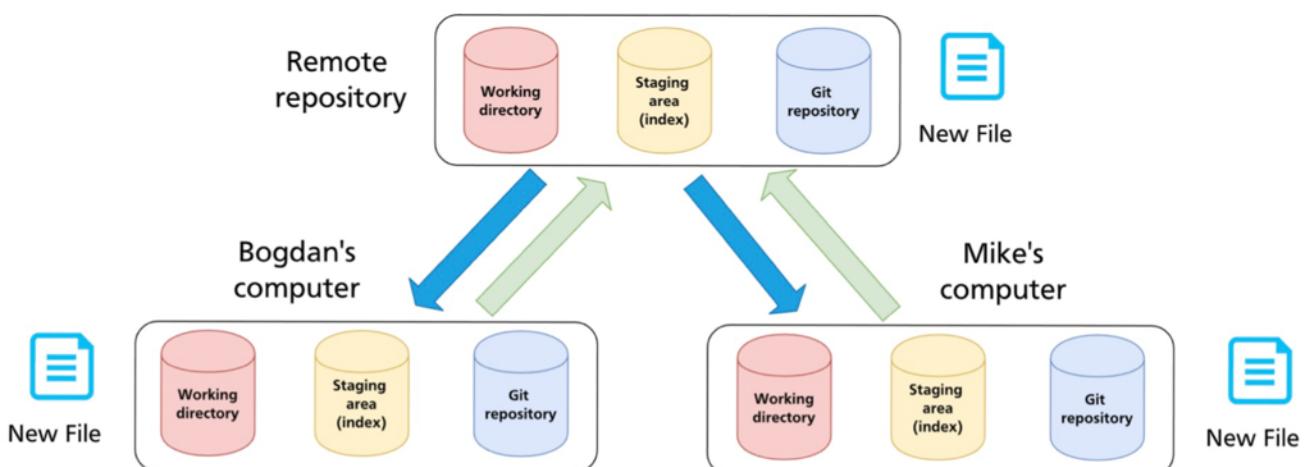
## WHY DO WE ACTUALLY NEED REMOTE REPOSITORY

### How remote repository empowers collaboration

BOGDAN  
STASHCHUK.COM

GIT COMPLETE GUIDE

git pull, git push, git fetch

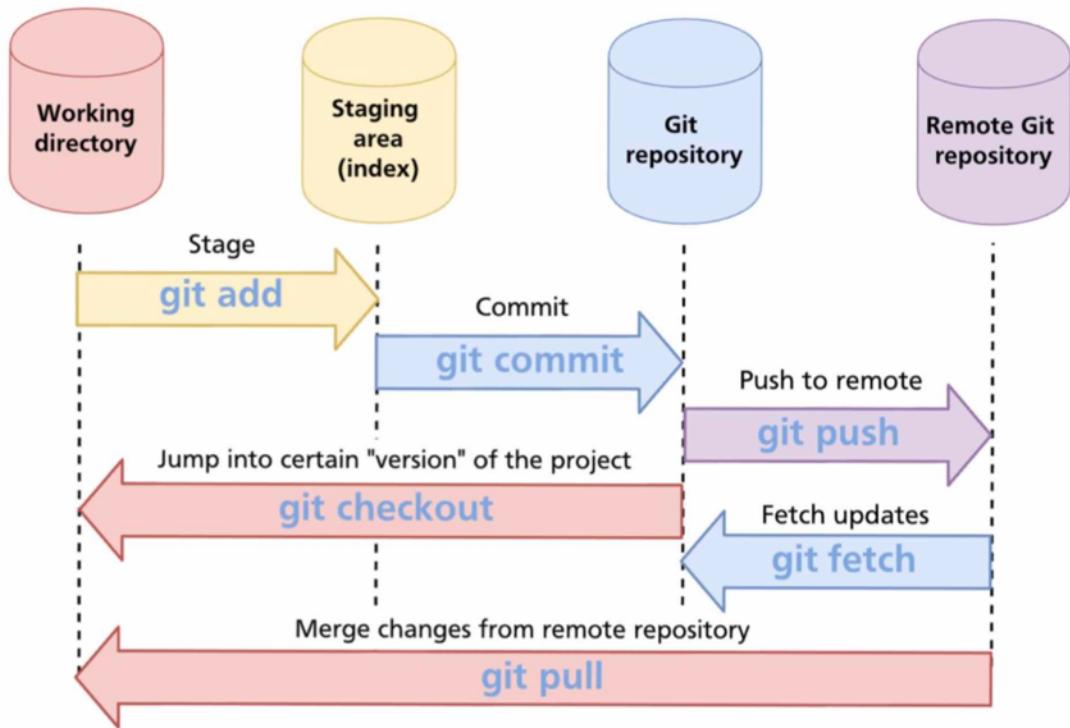


→ Remote repository is a single point of interaction between all collaborators.

## Section 10: Git push, fetch and pull

### Overview of the push, fetch and pull Git commands

#### Interaction with remote Git repository



→ Commands that are used for interaction with remote repositories : git push, git fetch, git pull.

→ Commands like git add, git commit and git checkout are used in local git repository.

git push - Usually you push to remote git repository changes that were already committed in local git repository. It means that you take some of the features in the project, you work on this feature, add your files, modify files, delete files and so on. After that, when you are happy with those changes, you add them into a staging area and commit changes.

Stage and commit changes:

git commit -a -m "Commit message"

You are able to commit changes and stage them in one command

When changes are committed and are located in your local git repository, you are able to push those changes into remote git repository. Of course, for that you need to have write access to the remote git repository and not anybody in the world is able to push to your own remote repository. You, as owner of your own remote repository, are able to grant access for writing to your repository.

**Note :-** push to remote repository only fully tested and verified features.

**git fetch** - git fetch command doesn't update your local working directory. It also does update staging area. It means files and folders you are currently seeing in your project will be affected by git fetch command, that's because git fetch command only updates your local git repository. It doesn't touch staging area or working directory.

**Example of fetch :-** Let's suppose that new branch was created in the Remote repository. After "git fetch" same branch will be created in your local Git repository because branch is just text file in the .git/refs/heads.

**git pull** :- This command will write changes into your working directory. Actually, under the hood it is a two step process; first git performs git fetch and downloads update from remote repository to your local git repository and after that it merges changes if any, into your local working directory and of course, changes are applied to your staging area.

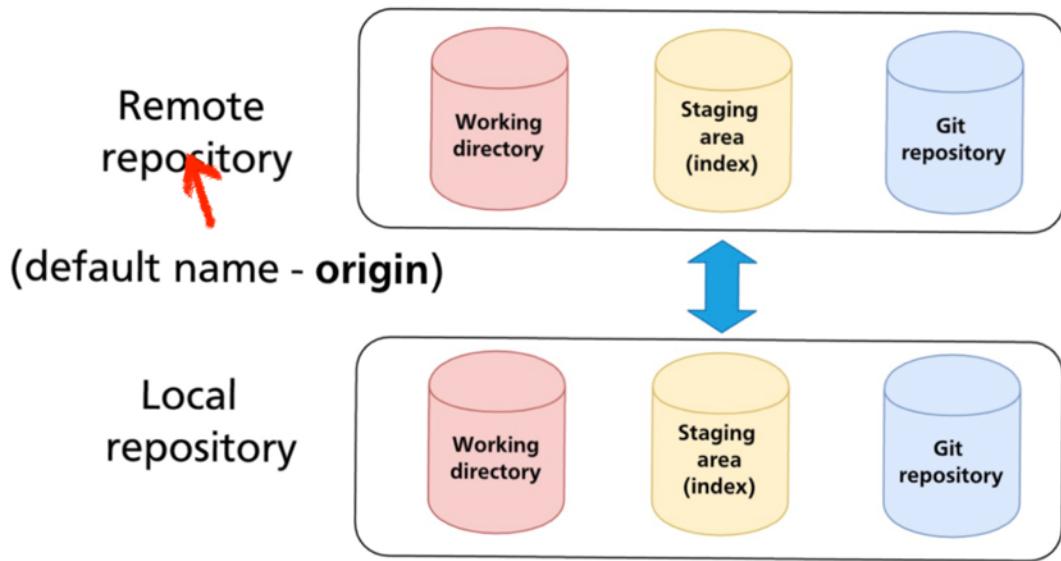
**Example of pull**

1. Clone remote repository.
2. Checkout "main" or "master" branch locally.
3. Make changes and commit them in "master" branch in remote repository.
4. After "git pull" locally Git will fetch remote changes.
5. After fetching Git will merge remote "main" or "master" branch into local "main" or "master" branch (fast forward or 3-way merge).

6. Staging area (index) and working directory will be automatically updated after merge.

Q How your local Git repository is "connected" to the remote repository?

What is origin



By default when you clone remote repository to your local repository, git automatically creates binding between remote repository and local repository and it creates default remote repository for your local repository and default name of remote repository is origin. You can actually connect your local repository to multiple remote repositories and in such case, every remote repository will have different name.

When you perform, for example, push, pull or fetch operations you actually choose which remote repository you want to interact with. And by default, if you, for example just write command `git pull`, you will pull from origin, from default remote repository.

with this command you are able to list all remote servers for your local repositories.

\* main

(END)

```
→ my-first-github-repository git:(main) git remote  
origin  
→ my-first-github-repository git:(main) git remote -v  
origin https://github.com/hema8Codes/my-first-github-repository.git (fetch)  
origin https://github.com/hema8Codes/my-first-github-repository.git (push)  
→ my-first-github-repository git:(main) git branch  
[1] + 1394 suspended git branch  
→ my-first-github-repository git:(main)
```

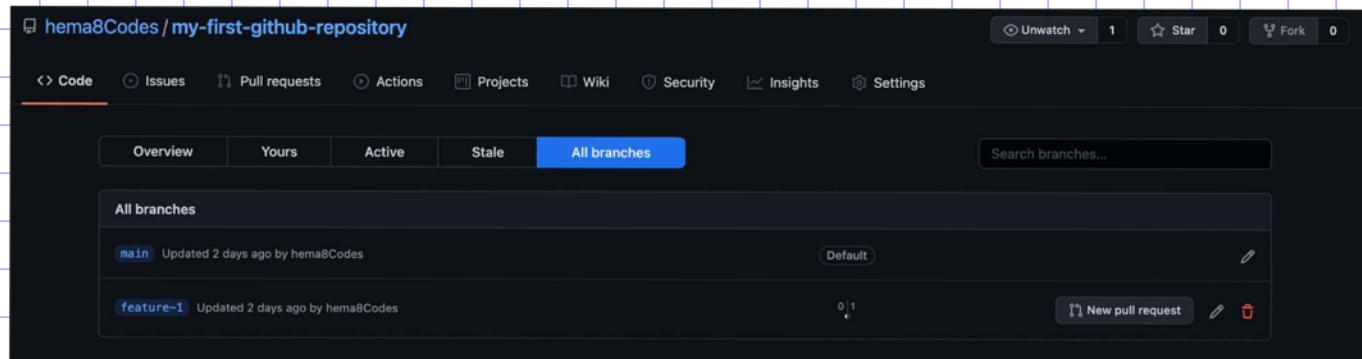
VRL of remote repository

listing branches

Here we see only one branch main. But in our remote repository at GitHub, we have created another branch called feature-1 branch. But here, we don't see that branch on the list and that's because by default git branch command list only your local branches.

Note: By default after cloning Git will not create corresponding local branches for all remote branches except default remote branch.

## List remote and local branches



At the moment in our remote git repository called my-first-github-repository there are two branches master branch and feature-1 branch. And we have already committed changes in feature-1 branch. After that we have cloned repository to our local computer.

If we list branches here using branch command, we see only default master branch

- my-first-github-repository git:(main) git branch  
that means that by default, GIT will not create local branches  
for every corresponding remote branch
- [1] + 1394 suspended git branch
- my-first-github-repository git:(main) git branch -a  
now you see all local branches and remote
- [2] + 1826 suspended git branch -a
- my-first-github-repository git:(main) git branch -r
- [3] + 1844 suspended git branch -r

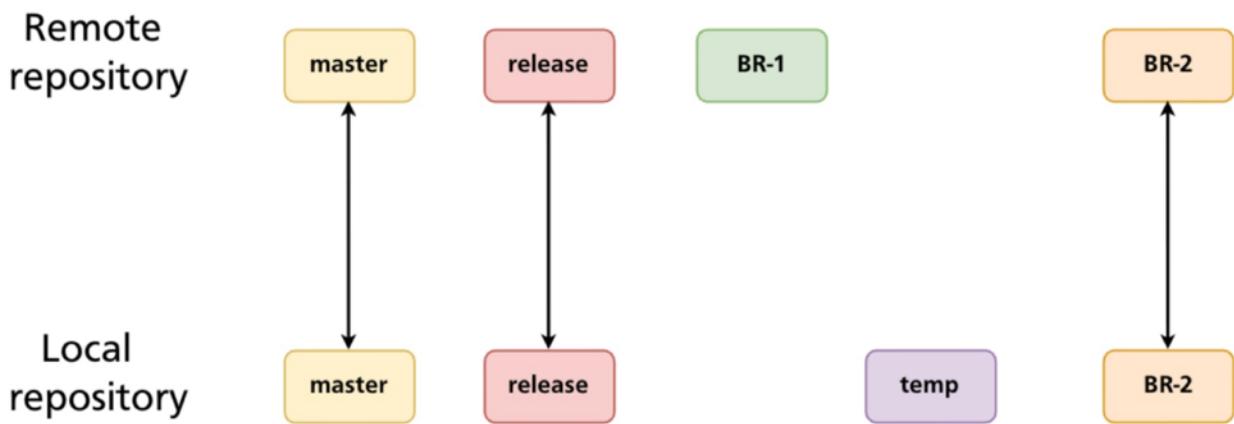
We will see list only of remote branch located at remote git repository

How you can actually list all branches available locally and on remote server?

→ git branch -a

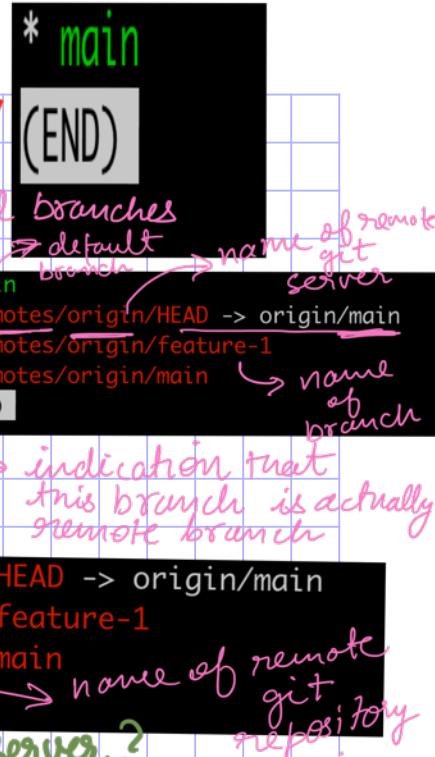
## What is tracking branch

### Tracking branches



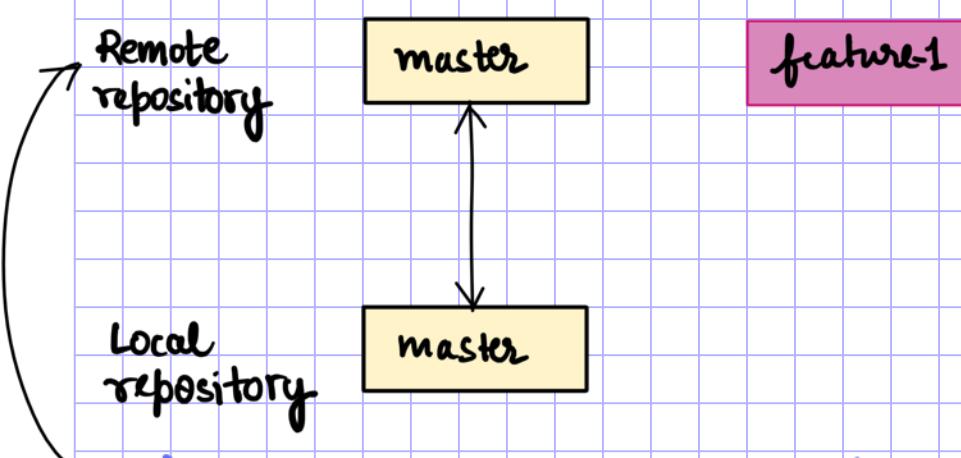
→ Tracking branch is your local branch that is connected to specific remote branch. And by default when you clone any remote repository, git creates only one tracking branch locally in your local git repository and name of this branch matches the full name of the branch in remote git repository.

→ In the diagram above you see three tracking branches: master, release and BR-2. All those



branches are present in remote repository and in local repository. Also notice, that in this diagram there is BR-1 branch that is present in remote repository but is absent in local repository and that means that there is no local tracking branch for BR-1 branch. Also, you may notice another branch called temp that is present locally in local repository but there is no corresponding branch in remote repository and that means that temp branch is not tracking branch

- Of course, you can create remote tracking branch for the temp branch by creating remote branch in remote repository. Same for BR-1 branch, simply by checking out remote branch, you will create corresponding local BR-1 branch in your local repository and this local branch will be automatically connected to remote branch and this branch will become tracking branch as well.
- Note that after cloning git will create only single tracking branch. In our case



in our case, there are two branches in remote git repository master and feature-1.

But after cloning, git has created only single local branch called master or main and this branch is tracking branch.

How you can verify which branches are tracking branches and which are not.

```
→ my-first-github-repository git:(main) git branch -vv
```

```
[4] + 1973 suspended git branch -vv
```

```
→ my-first-github-repository git:(main)
```

Here you will see all local branches, including corresponding remote branches.

```
* main 3a3e202 [origin/main] Create hello-github.txt
```

```
(END)
```

that means your local master branch is tracking branch

SHA1 hash of last commit object in git repository

message of the last commit made in this branch

→ Using git branch command :-

git branch -vv → you will see information

about tracking branches.

now there is only single tracking branch in our local repository → master / main branch

and remote corresponding branch is origin / master

### Checkout remote branch

→ We will create another local tracking branch for remote feature - 1 branch.

→ There is already single tracking branch in our cloned repository and that branch is main / master.

main branch has remote tracking branch

```
* main 3a3e202 [origin/main] Create hello-github.txt  
(END)
```

origin/  
main

```
→ my-first-github-repository git:(main) git branch -vv
```

\* main

remotes/origin/HEAD -> origin/main  
remotes/origin/feature-1  
remotes/origin/main

(END)

There  
is  
another  
feature-1

[4] + 1973 suspended git branch -vv

```
→ my-first-github-repository git:(main) git branch -a
```

[5] + 2098 suspended git branch -a

→ my-first-github-repository git:(main) git checkout feature-1

Branch 'feature-1' set up to track remote branch 'feature-1' from 'origin'.

Switched to a new branch 'feature-1'

```
→ my-first-github-repository git:(feature-1) git branch -vv
```

[6] + 2136 suspended git branch -vv

Now here we see two branches

```
* feature-1 e0cffcb [origin/feature-1] File list.txt was created in the feature-1 branch  
main 3a3e202 [origin/main] Create hello-github.txt  
(END)
```

main branch still  
tracks remote origin/main branch

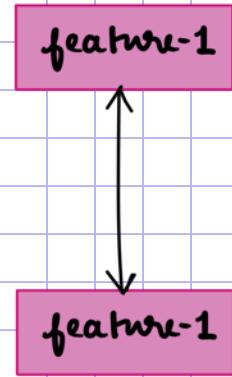
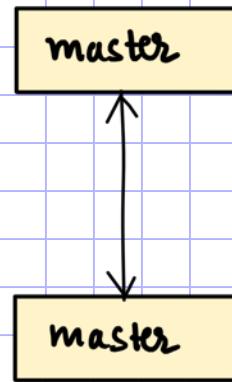
## Tracking branches

feature-1 branch  
now tracks remote  
origin feature-1  
branch.

Now that means we  
have two local  
tracking branches.

our local branches  
are tracking branch  
for remote's branches  
master and feature-1

Remote  
repository



```
→ my-first-github-repository git:(feature-1) git checkout main
```

Switched to branch 'main'

Your branch is up to date with 'origin/main'.

```
→ my-first-github-repository git:(main) git branch -d feature-1
```

warning: deleting branch 'feature-1' that has been merged to  
'refs/remotes/origin/feature-1', but not yet merged to HEAD.

Deleted branch feature-1 (was e0cffcb).

```
→ my-first-github-repository git:(main) git branch
```

[7] + 2260 suspended git branch

listing local branch

```
→ my-first-github-repository git:(main) git branch -vv
```

```
* main  
(END)
```

```
* main 3a3e202 [origin/main] Create hello-github.txt  
(END)
```

This main branch is still tracking  
remote origin/main branch.

git has  
allowed us to  
delete local  
branch because  
there was corresponding  
feature-1 branch  
in remote repository.

# Git remote show origin

→ This command "git remote show origin" shows us information about connections between your local repository and remote repositories.

- feature-1 branch tracks remote origin/feature-1
- main branch tracks origin/master branch

```
* feature-1 e0cffcb [origin/feature-1] File list.txt was created in the feature-1 branch  
main 3a3e202 [origin/main] Create hello-github.txt  
(END)
```

→ origin is simply default name for remote repository.

Now we have two local tracking branch  
feature-1 branch and master branch

```
→ my-first-github-repository git:(main) git checkout feature-1
```

Branch 'feature-1' set up to track remote branch 'feature-1' from 'origin'.  
Switched to a new branch 'feature-1'

```
→ my-first-github-repository git:(feature-1) git branch
```

```
[9] + 2500 suspended git branch
```

```
→ my-first-github-repository git:(feature-1) git branch -vv
```

```
[10] + 2518 suspended git branch -vv
```

```
→ my-first-github-repository git:(feature-1) git remote  
origin
```

```
→ my-first-github-repository git:(feature-1) git remote -v
```

origin https://github.com/hema8Codes/my-first-github-repository.git (fetch)  
origin https://github.com/hema8Codes/my-first-github-repository.git (push)

```
→ my-first-github-repository git:(feature-1) git remote show origin
```

\* remote origin with this command you will see additional information.

Fetch URL: https://github.com/hema8Codes/my-first-github-repository.git

Push URL: https://github.com/hema8Codes/my-first-github-repository.git

HEAD branch: main → name of default remote branch in remote repository.

Remote branches:

- feature-1 tracked
- main tracked

Local branches configured for 'git pull':

- feature-1 merges with remote feature-1

- main merges with remote main

Local refs configured for 'git push':

- feature-1 pushes to feature-1 (up to date)

- main pushes to main (up to date)

```
→ my-first-github-repository git:(feature-1)
```

This command will show us list of remote servers.

URL

And that means our local branches feature-1 branch and main branch can be used for git pull and git push operations. That's because those branches are tracking branches and they have corresponding remote branches.

and when you will enter git pull

command feature-1 branch will merge with remote feature-1 branch. (same will be for main branch). When you perform push, your local changes from feature-1 branch will be pushed to corresponding remote feature-1 branch. (same for main branch)

## Git fetch in action

→ Creating new branch in remote repository and we will observe whether we will get information about this new branch in our local repository

A screenshot of a GitHub repository page for 'hema8Codes/my-first-github-repository'. The page shows a message 'Branch created.' and a list of branches. A blue arrow points from the text 'new branch is created' to the 'temp' branch in the list. Handwritten notes explain that 'temp' and 'main' branches point to the same commit. The repository has 3 branches and 0 tags. It contains files like 'README.md' and 'hello-github.txt'.

now we still see only two branches feature-1 branch and master branch. we don't see temp branch in this list and that's where git fetch command comes in.

origin/HEAD -> origin/main  
origin/feature-1  
} origin/main  
(END)

→ my-first-github-repository git:(feature-1) git branch -r listing remote repositories

[1] + 1038 suspended git branch -r

→ my-first-github-repository git:(feature-1) git fetch with this command you will be able to fetch From https://github.com/hema8Codes/my-first-github-repository remote changes and

\* [new branch] temp -> origin/temp download them into your local repository

→ my-first-github-repository git:(feature-1) git remote -v

origin https://github.com/hema8Codes/my-first-github-repository.git (fetch)

origin https://github.com/hema8Codes/my-first-github-repository.git (push)

→ URL configured with this list

*We still see only two local branches.*

```
* feature-1 e0cffcb [origin/feature-1] File list.txt was created in the feature-1 branch  
main 3a3e202 [origin/main] Create hello-github.txt  
(END)
```

{ origin/HEAD -> origin/main      *remote branches*

  origin/feature-1

  origin/main

  origin/temp

(END)

*remote branch*

*Here we will see temp branch on the list*

*that means after git fetch command, git has not created temp branch locally*

→ my-first-github-repository git:(feature-1) git branch -vv  
          *listing local branches again*

[2] + 1107 suspended git branch -vv

→ my-first-github-repository git:(feature-1) git branch -r

[3] + 1125 suspended git branch -r

→ my-first-github-repository git:(feature-1) git checkout temp

Branch 'temp' set up to track remote branch 'temp' from 'origin'.

Switched to a new branch 'temp' *We need to checkout remote branch if we want to create corresponding tracking branch*

→ my-first-github-repository git:(temp) git branch -vv

[4] + 1157 suspended git branch -vv

*Now we have three local branches*

```
feature-1 e0cffcb [origin/feature-1] File list.txt was created in the feature-1 branch  
main 3a3e202 [origin/main] Create hello-github.txt  
* temp 3a3e202 [origin/temp] Create hello-github.txt  
(END)
```

→ currently checkout branch

→ my-first-github-repository git:(temp) git remote show origin

\* remote origin

  Fetch URL: https://github.com/hema8Codes/my-first-github-repository.git

  Push URL: https://github.com/hema8Codes/my-first-github-repository.git

  HEAD branch: main

  Remote branches:

    feature-1 tracked

    main tracked

    temp tracked

  Local branches configured for 'git pull':

    feature-1 merges with remote feature-1

    main merges with remote main

    temp merges with remote temp

  Local refs configured for 'git push':

    feature-1 pushes to feature-1 (up to date)

    main pushes to main (up to date)

    temp pushes to temp (up to date)

→ deleting temp branch in remote repository.

GitHub branches page for hema8Codes/my-first-github-repository. The 'All branches' tab is selected. The branches listed are 'main' (Default), 'feature-1', and 'temp'. The 'temp' branch has a delete icon next to it, which is circled with a blue arrow pointing to it.

→ my-first-github-repository git:(temp) git remote show origin

\* remote origin

Fetch URL: https://github.com/hema8Codes/my-first-github-repository.git  
Push URL: https://github.com/hema8Codes/my-first-github-repository.git  
HEAD branch: main

Remote branches:

feature-1	tracked
main	tracked
refs/remotes/origin/temp stale (use 'git remote prune' to remove)	

Local branches configured for 'git pull':

feature-1	merges with remote feature-1
main	merges with remote main
temp	merges with remote temp

Local refs configured for 'git push':

feature-1	pushes to feature-1 (up to date)
main	pushes to main (up to date)

→ my-first-github-repository git:(temp) git remote prune origin

Pruning origin

URL: https://github.com/hema8Codes/my-first-github-repository.git

\* [pruned] origin/temp → and this branch origin/temp was successfully pruned.

now we will see here in this list of remote branches that remote branch origin/temp is now stale and we can use command git remote prune in order to move temp branch from this list

```

→ my-first-github-repository git:(temp) git remote show origin
* remote origin
  Fetch URL: https://github.com/hema8Codes/my-first-github-repository.git
  Push URL: https://github.com/hema8Codes/my-first-github-repository.git
  HEAD branch: main
  Remote branches:
    feature-1 tracked
    main tracked ] now we will see only two remote branch
  Local branches configured for 'git pull':
    feature-1 merges with remote feature-1
    main merges with remote main
    temp merges with remote temp } But we still see here local
  Local refs configured for 'git push':
    feature-1 pushes to feature-1 (up to date)
    main pushes to main (up to date)
→ my-first-github-repository git:(temp) git branch -vv

```

[5] + 1265 suspended git branch -vv

```

feature-1 e0cffcb [origin/feature-1] File list.txt was created in the feature-1 branch
main 3a3e202 [origin/main] Create hello-github.txt
* temp 3a3e202 [origin/temp: gone] Create hello-github.txt
(END)

```

We will see now here that there is local temp branch that is configured for tracking remote origin/temp branch but this remote tracking branch is gone or removed. You can create remote branch again by using, for example, git push c main. But we will now remove this local temp branch.

```

→ my-first-github-repository git:(temp) git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
→ my-first-github-repository git:(main) git branch -d temp → deleting local branch
Deleted branch temp (was 3a3e202).
→ my-first-github-repository git:(main) git branch -vv → now temp branch is absent here

```

[6] + 1337 suspended git branch -vv

```

[6] + 1337 suspended git branch -vv } feature-1 e0cffcb [origin/feature-1] File list.txt was created in the feature-1 branch
* main 3a3e202 [origin/main] Create hello-github.txt
(END)

```

```

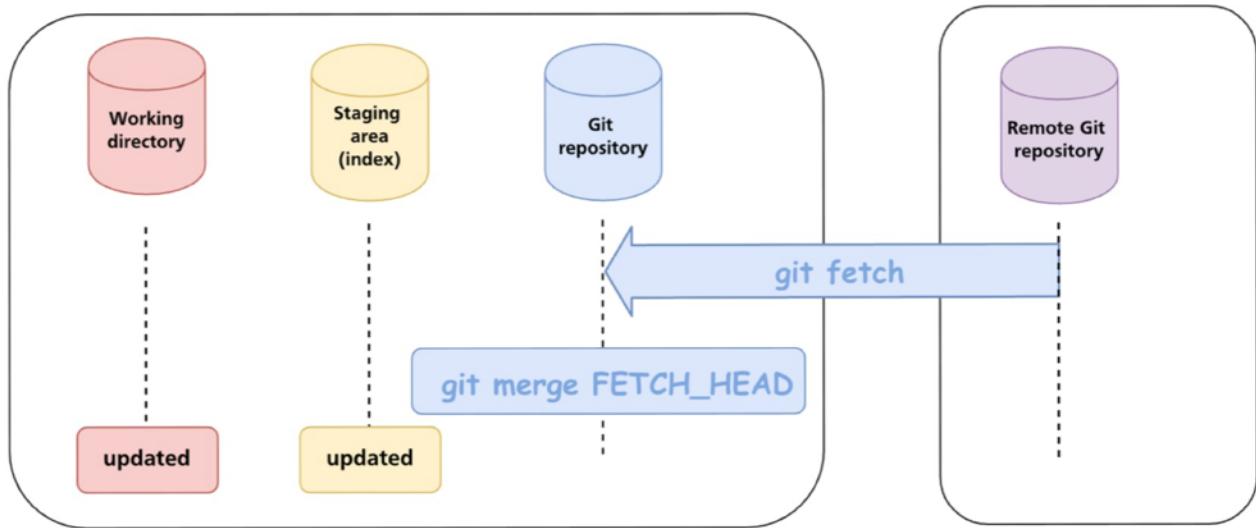
→ my-first-github-repository git:(main) git remote show origin
* remote origin
  Fetch URL: https://github.com/hema8Codes/my-first-github-repository.git
  Push URL: https://github.com/hema8Codes/my-first-github-repository.git
  HEAD branch: main
  Remote branches:
    feature-1 tracked
    main tracked
  Local branches configured for 'git pull':
    feature-1 merges with remote feature-1
    main merges with remote main
  Local refs configured for 'git push':
    feature-1 pushes to feature-1 (up to date)
    main pushes to main (up to date)
→ my-first-github-repository git:(main) | } we will not see temp branch here on this list as well.

```

## Git pull is 2-step process

→ Git Pull consists of Git Fetch and Git Merge.

Git pull - merges remote branch into current branch



→ Git pull merges remote branch into current local branch and of course, current local branch should track remote branch and under the hood git pull is actually two step process:-

- ① Git performs git fetch and fetches all updates from remote repository and write them into local repository. For example :- all new branches, all new commits that were created in remote repository will be also created in your local git repository.
- ② After git has performed git fetch , it performs git merge . But git merge is performed locally in your local git repository, for that we don't invoke remote repository. All is performed locally. In merge process , there are actually two branches and the receiving in the process will be local branch and let's say feature branch will be corresponding remote branch and git will merge remote branch into local branch . But interestingly git uses such name as `FETCH_HEAD` instead of name of remote branch. After git merge remote branch is merge into local branch.

→ After fetching local Git merges remote branch into local tracking branch using either fast forward merge or 3-way merge.

③ After merge, if it was successful, git updates staging area and working directory and that happens automatically because currently checked out branch is actually the branch in which we have merged remote branch. That's how git pull works!

→ Git pull updates only single local currently checked out branch.

## How to perform git pull

### Git pull

1. Checkout local branch and make sure it is tracking branch and has corresponding remote branch. Use **git branch -vv** for check

2. Enter **git pull**

3. Git will fetch all changes from remote repository - **git fetch** is executed in background

4. After fetching Git updates **FETCH\_HEAD** file that contains SHA1 hashes of last commits in remote repository for all tracking branches

5. Git merges remote branch into current branch **git merge FETCH\_HEAD**

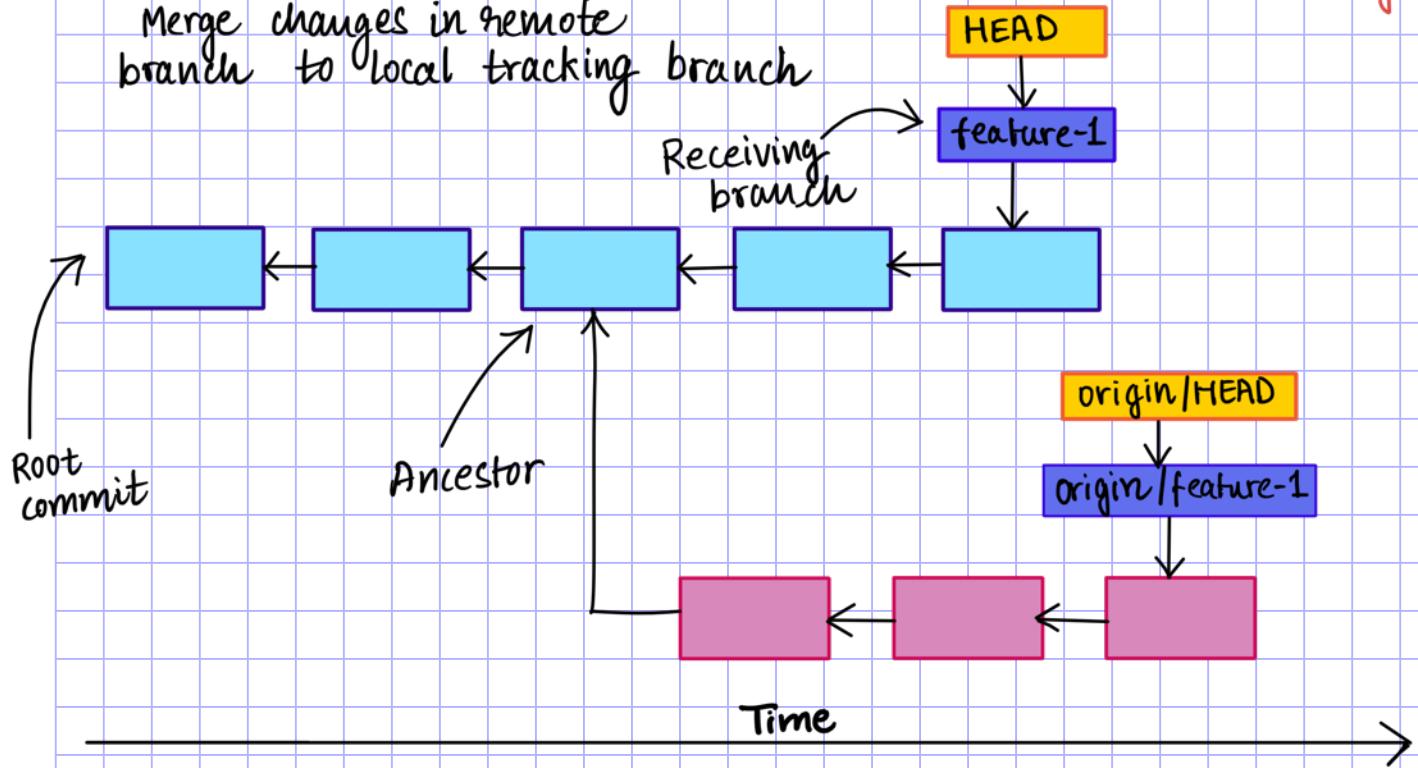
If only remote  
branch was changed  
git applies fast  
forward merge.

if both remote  
and local branches  
were changed git  
applies 3-way merge.

# Git pull

Merge changes in remote branch to local tracking branch

your local branch  
is receiving branch



From the diagram above you may see differences and there are actually two same branches , but one branch is local branch ; feature-1 branch . There is corresponding remote branch ; origin/ feature-1 , but this branch is located in remote repository .

In the diagram you see two HEADS ; local HEAD that points to feature-1 branch and origin/HEAD that points to remote branch → origin / feature-1 branch , when you perform git pull , you need to first checkout local branch and this branch will be receiving branch . In the above diagram, you can see that there are actually changes both in local feature-1 branch and in remote corresponding branch . In such case , when you would want to merge changes from remote feature-1 branch into local feature-1 branch , git will perform three way merge .

# What is FETCH\_HEAD

They are tracking branches

Corresponding remote branch

```
* feature-1
  main
  remotes/origin/HEAD -> origin/main
  remotes/origin/feature-1
  remotes/origin/main
(END)
```

```
* feature-1 e0cffcb [origin/feature-1] File list.txt was created in the feature-1 branch
  main 3a3e202 [origin/main] Create hello-github.txt
(END)
```

```
→ my-first-github-repository git:(feature-1) git branch -a
```

list of remote and local branch in our repository

```
[1] + 1939 suspended git branch -a
```

```
→ my-first-github-repository git:(feature-1) git branch -vv
```

list of local branches with corresponding remote branches.

```
[2] + 1960 suspended git branch -vv
```

```
→ my-first-github-repository git:(feature-1) git remote show origin
```

```
* remote origin
```

```
  Fetch URL: https://github.com/hema8Codes/my-first-github-repository.git
```

```
  Push URL: https://github.com/hema8Codes/my-first-github-repository.git
```

```
  HEAD branch: main
```

```
  Remote branches:
```

```
    feature-1 tracked
```

```
    main tracked
```

Local branches configured for 'git pull':

```
  feature-1 merges with remote feature-1
```

```
  main merges with remote main
```

Local refs configured for 'git push':

```
  feature-1 pushes to feature-1 (up to date)
```

```
  main pushes to main (up to date)
```

```
→ my-first-github-repository git:(feature-1)
```

```
→ my-first-github-repository git:(feature-1) git fetch -v
```

```
POST git-upload-pack (144 bytes)
```

```
From https://github.com/hema8Codes/my-first-github-repository
```

```
= [up to date] feature-1 -> origin/feature-1 ]
```

```
= [up to date] main -> origin/main ]
```

```
→ my-first-github-repository git:(feature-1) git pull
```

```
Already up to date.
```

```
→ my-first-github-repository git:(feature-1) git pull -v
```

```
POST git-upload-pack (144 bytes)
```

```
From https://github.com/hema8Codes/my-first-github-repository
```

```
= [up to date] feature-1 -> origin/feature-1 ]
```

```
= [up to date] main -> origin/main ]
```

```
Already up to date.
```

```
→ my-first-github-repository git:(feature-1)
```

git has updates from this URL

but there is nothing to update locally as both branches are up to date

git pull actually performs git fetch first.

```

→ my-first-github-repository git:(feature-1) git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
→ my-first-github-repository git:(main) git pull -v
POST git-upload-pack (144 bytes)
From https://github.com/hema8Codes/my-first-github-repository
 = [up to date]      main → origin/main
 = [up to date]      feature-1 → origin/feature-1
    ] order of branches
    is opposite
Already up to date.
→ my-first-github-repository git:(main) 

```

```

→ my-first-github-repository git:(main) cd .git
→ .git git:(main) ls -la
total 56
drwxr-xr-x@ 14 hemakshipandey staff 448 Aug 28 13:18 .
drwxr-xr-x@ 5 hemakshipandey staff 160 Aug 28 13:18 ..
-rw-r--r-- 1 hemakshipandey staff 250 Aug 28 13:18 FETCH_HEAD ✓
-rw-r--r-- 1 hemakshipandey staff 21 Aug 28 13:13 HEAD ✓
-rw-r--r-- 1 hemakshipandey staff 41 Aug 28 13:18 ORIG_HEAD ✓
-rw-r--r-- 1 hemakshipandey staff 394 Aug 27 14:51 config
-rw-r--r-- 1 hemakshipandey staff 73 Aug 24 18:37 description
drwxr-xr-x@ 15 hemakshipandey staff 480 Aug 24 18:37 hooks
-rw-r--r-- 1 hemakshipandey staff 217 Aug 28 13:13 index
drwxr-xr-x@ 3 hemakshipandey staff 96 Aug 24 18:37 info
drwxr-xr-x@ 4 hemakshipandey staff 128 Aug 28 13:13 logs
drwxr-xr-x@ 4 hemakshipandey staff 128 Aug 28 13:18 objects
-rw-r--r-- 1 hemakshipandey staff 183 Aug 27 14:51 packed-refs
drwxr-xr-x@ 5 hemakshipandey staff 160 Aug 27 14:51 refs
→ .git git:(main) cat FETCH_HEAD
3a3e202a8ffc7361245fd8416e496d423e697a37
e0cffcb157f5dccc3b2b98599b2b908932ff0c0
branch 'main' of https://github.com/hema8Codes/my-first-github-repository
not-for-merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository
b-repository
→ .git git:(main) cd ..
→ my-first-github-repository git:(main) git checkout feature-1
Switched to branch 'feature-1'
Your branch is up to date with 'origin/feature-1'.
→ my-first-github-repository git:(feature-1) git fetch
→ my-first-github-repository git:(feature-1) cat .git/FETCH_HEAD
e0cffcb157f5dccc3b2b98599b2b908932ff0c0
branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository
not-for-merge branch 'main' of https://github.com/hema8Codes/my-first-github-repository
}
branch 'main' of https://github.com/hema8Codes/my-first-github-repository

```

that means when you will perform git merge operation under the hood git will execute git merge FETCH\_HEAD. After that command git will find in this list single branch (first branch) that doesn't have "not-for-merge" tag and it will find this SHA-1 hash (of last commit) of feature-1 branch and then merge this commit and all previous commits, if any, into local branch.

During "git pull" Git will first execute "git fetch". After fetch it will update .git /FETCH\_HEAD list and first branch in this list will be currently checked out branch. Finally Git executes "git merge FETCH\_HEAD" command that finds first branch in .git /FETCH\_HEAD list without "not-for-merge" tag and merges it into local tracking correctly checked out branch.

```
→ my-first-github-repository git:(feature-1) git cat-file -p e0cffcb157f5dcccc3b2b98599b2b908932ff0c0
tree a6fd87d1d5c8de53bfc99a93b0b9c215be423f2f
parent 3a3e202a8ffc7361245fd8416e496d423e697a37
author Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com> 1629804595 +0530
committer GitHub <noreply@github.com> 1629804595 +0530
gpgsig -----BEGIN PGP SIGNATURE-----
```

wsBcBAABCQAQBQJhJNgzCRBK7hj40v3rIwAADyIIAJPt0aftiWktyngUU9an/Sgl  
irX7pA3nPw7ykxr4+ePlezHxPW2Bt4tn9Yp/8j4KACYP65S9S2ictbLJ+VkJwXIH  
81Xct92URENatMo+5LP1AeKY4xV4T6rBgBa1xKx+KH/Zb1bngpI7d++VPIKwP7G  
p8+r5ECAftHauPAAt7P65WG/T0tUhI9YJFyQGuXYXKh2yCe3fU++PL4edEFgYgphc  
5Ch6Yw0E1s7wJtgyJEVK4p/PNBqt+6QYNjmeG5Nr2f0H2sZ+kmpqurwmocFFzHAU  
guyCZA73noHWQBwSMm48AAfKJd9PcrLkk9XX+FgmB2j8nsVFLKUD5fYSyTgnMVM=

=665h  
-----END PGP SIGNATURE-----

→ reading SHA1 hash  
that means now  
in FETCH\_HEAD file  
this SHA1 hash  
points to very last  
commit in remote  
repository.

```
File list.txt was created in the feature-1 branch
→ my-first-github-repository git:(feature-1) git log
```

```
commit e0cffcb157f5dcccc3b2b98599b2b908932ff0c0 (HEAD -> feature-1, origin/feature-1)
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Tue Aug 24 16:59:55 2021 +0530

  File list.txt was created in the feature-1 branch
  local feature-1 branch is
  in sync with remote feature-1
commit 3a3e202a8ffc7361245fd8416e496d423e697a37 (origin/main, origin/HEAD, main)
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Tue Aug 24 16:12:03 2021 +0530

  Create hello-github.txt

commit 6d0ad74d78ba29d04f54579352ea3d5ceb600bb4
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Tue Aug 24 13:07:24 2021 +0530

  Update README.md

commit 398b48e4ecae92a3ff076853d3ec39b764a98a07
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Tue Aug 24 13:03:00 2021 +0530

  Initial commit
(END)
```

# Git pull with fast forward merge

Making changes in our remote repository in feature-1 branch

A screenshot of a GitHub repository page for 'my-first-github-repository'. The 'Code' tab is selected. A file named 'another-file.txt' is being edited in the 'feature-1' branch. The code contains a single line: '1 Just another file created in the feature-1 branch'. There are options to 'Edit new file' or 'Preview'.

After committing changes

A screenshot of the GitHub commit history for the 'feature-1' branch. It shows several commits:

- Commits on Aug 28, 2021:
  - Create another-file.txt (verified, hash e0cffcb)
- Commits on Aug 24, 2021:
  - File list.txt was created in the feature-1 branch (verified, hash 3a3e282)
  - Create hello-github.txt (verified, hash 6d8ad74)
  - Update README.md (verified, hash 398b48e)
  - Initial commit (verified, hash 398b48e)

A handwritten note points to the commit hash e0cffcb with the text 'SHA1 hash of the commit'.

```
→ my-first-github-repository git:(feature-1) git pull -v
POST git-upload-pack (144 bytes)
POST git-upload-pack (258 bytes)
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 840 bytes | 210.00 KiB/s, done.
From https://github.com/hema8Codes/my-first-github-repository
  e0cffcb..065266f  feature-1  -> origin/feature-1
 = [up to date]      main    -> origin/main
Updating e0cffcb..065266f
Fast-forward
  feature/another-file.txt | 1 +
   1 file changed, 1 insertion(+)
  create mode 100644 feature/another-file.txt
```

git performs some other actions and git uploads local changes in this feature-1 branch

this slightly matches with the output after git fetch command

main branch is upto date  
But there were changes in feature-1 branch

as only changes were made in remote repository, fast forward merge is possible

```
→ my-first-github-repository git:(feature-1) ls -la
total 16
drwxr-xr-x@ 6 hemakshipandey staff 192 Aug 28 18:43 .
drwx-----@ 10 hemakshipandey staff 320 Aug 28 18:43 ..
drwxr-xr-x@ 14 hemakshipandey staff 448 Aug 28 18:43 .git
-rw-r--r-- 1 hemakshipandey staff 136 Aug 24 18:37 README.md
drwxr-xr-x@ 4 hemakshipandey staff 128 Aug 28 18:43 feature
-rw-r--r-- 1 hemakshipandey staff 13 Aug 24 18:37 hello-github.txt
→ my-first-github-repository git:(feature-1) ls -la feature
total 16
drwxr-xr-x@ 4 hemakshipandey staff 128 Aug 28 18:43 .
drwxr-xr-x@ 6 hemakshipandey staff 192 Aug 28 18:43 ..
-rw-r--r-- 1 hemakshipandey staff 50 Aug 28 18:43 another-file.txt
-rw-r--r-- 1 hemakshipandey staff 23 Aug 28 13:43 list.txt
→ my-first-github-repository git:(feature-1)
→ my-first-github-repository git:(feature-1) git ls-files -s → listing files in
100644 b589ea26cb5d599a8eb468bf93ff5aa210159dc7 0 README.md
100644 6a2d7e4af29fd9c891b71c2e4ba0b7c398188a16 0 feature/another-file.txt
100644 3776b8c0940c336c34a03f2542abb1f58d8cb9c0 0 feature/list.txt
100644 bca999f8e67cf145e5c2eb7b72fd6fc0acd9bdc 0 hello-github.txt
```

last commit in our current local feature-1 branch

```
commit 065266f33a06b17bc8bb5232452d33467867a659 (HEAD -> feature-1, origin/feature-1)
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Sat Aug 28 18:42:03 2021 +0530
```

Create another-file.txt

```
commit e0cffcb157f5dcccc3b2b98599b2b908932ff0c0
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Tue Aug 24 16:59:55 2021 +0530
```

File list.txt was created in the feature-1 branch

```
commit 3a3e202a8ffc7361245fd8416e496d423e697a37 (origin/main, origin/HEAD, main)
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Tue Aug 24 16:12:03 2021 +0530
```

Create hello-github.txt

```
commit 6d0ad74d78ba29d04f54579352ea3d5ceb600bb4
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Tue Aug 24 13:07:24 2021 +0530
```

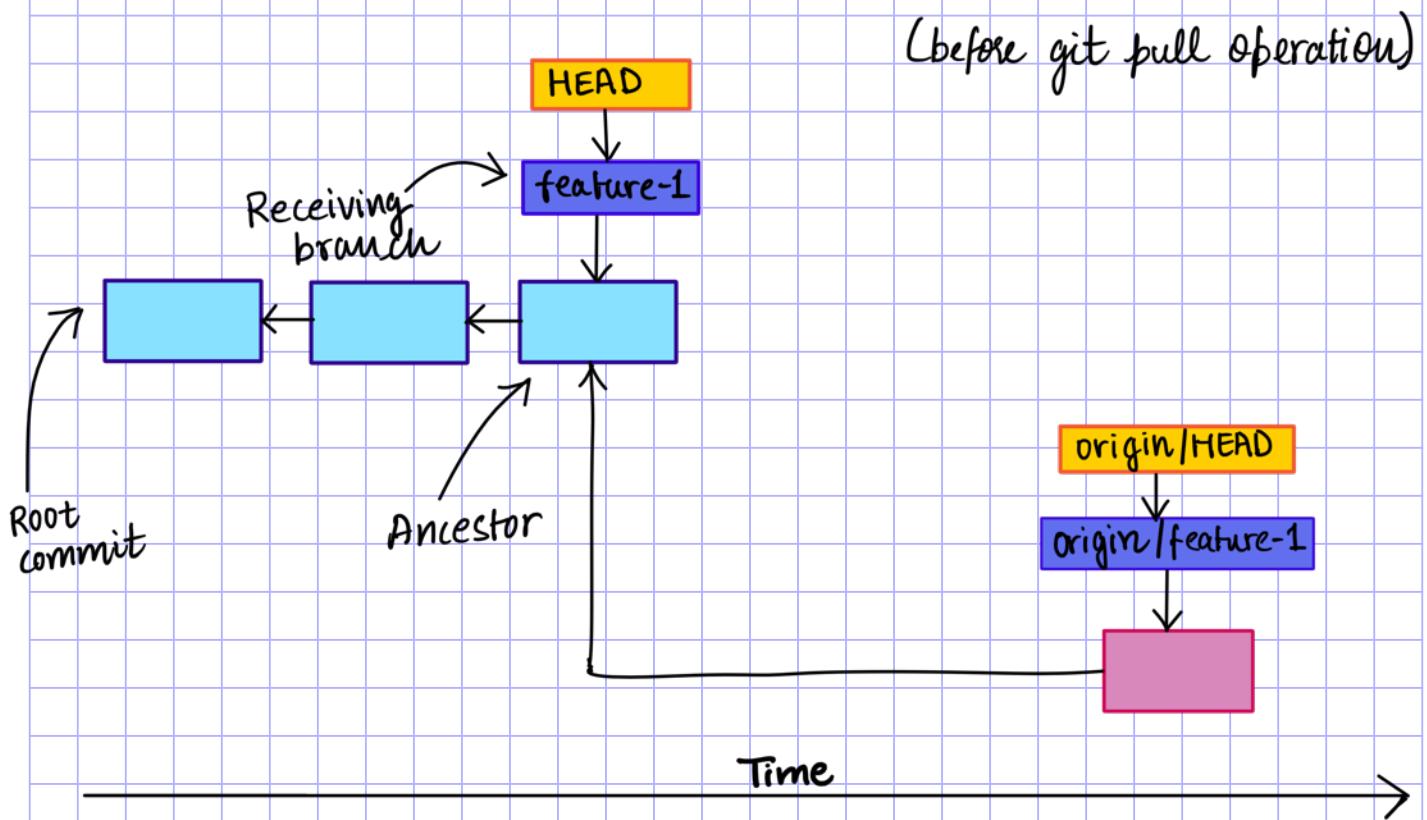
Update README.md

```
commit 398b48e4ecae92a3ff076853d3ec39b764a98a07
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Tue Aug 24 13:03:00 2021 +0530
```

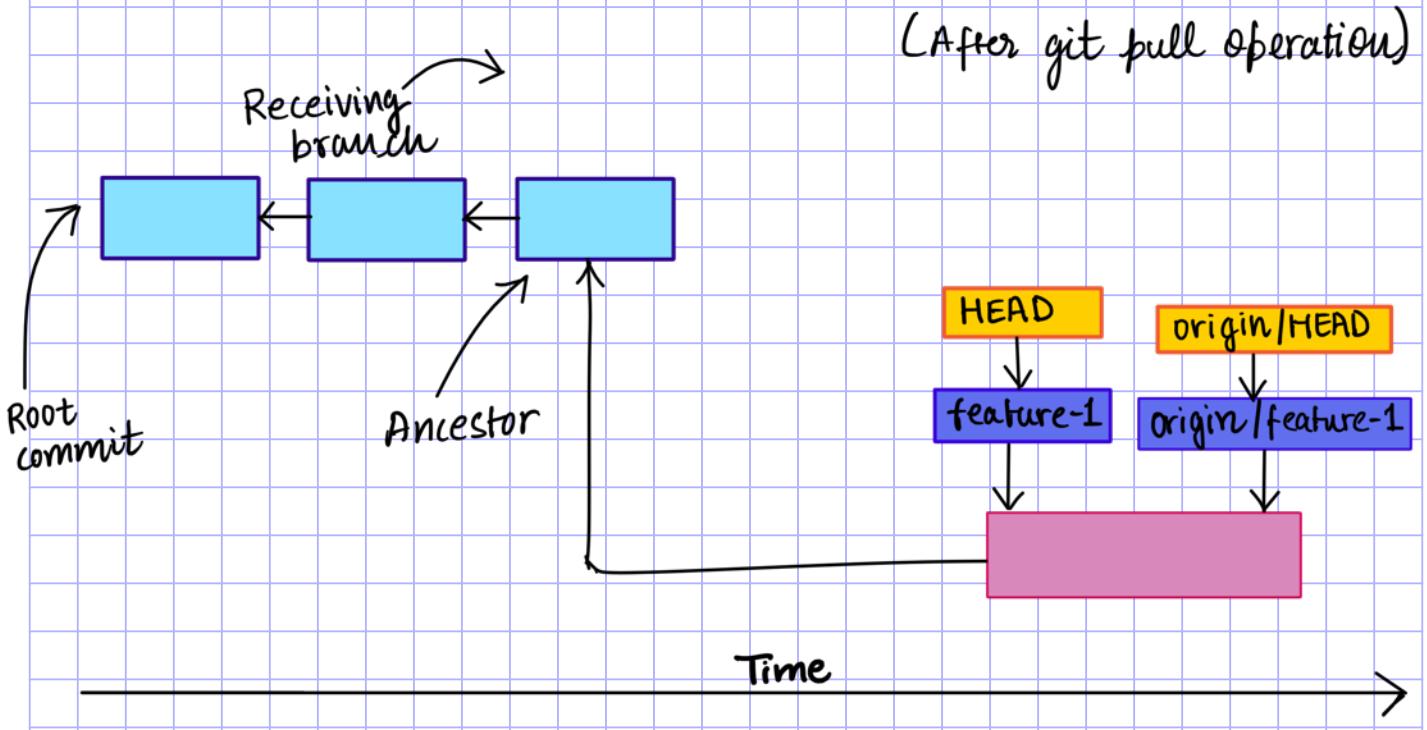
Initial commit

(END)

## Git pull with fast forward



After git pull operation, in our case, git has seen that there are no other commits in local feature-1 branch and that why in this case it is possible to perform fast forward merge and in fast forward merge, git simply moves pointer of the branch to last commit in the branch we are merging into receiving branch.



# Fetch remote changes manually

let's make some changes both in remote repository and in local repository and after that we would try to perform git pull operation.

The screenshot shows a dark-themed code editor interface. In the top bar, there are tabs for 'SOURCE CONTR...' and '[Preview] CHANGELOG\_7.8.md'. Below the tabs, a sidebar displays a list of changes: 'Modified another-file.txt locally' under 'Changes', with a note '(1)'. The main editor area shows a file named 'another-file.txt' with the following content:

```
feature > another-file.txt
1 Just another file created in the feature-1 branch
2
3 This file was modified locally.
```

A handwritten note in blue ink is overlaid on the right side of the editor, reading: "making changes in feature-1 branch and committing those changes in local repository."

The screenshot shows a GitHub repository page for 'hema8Codes/my-first-github-repository'. The URL in the address bar is 'github.com/hema8Codes/my-first-github-repository/new/feature-1'. The page title is 'Making changes in remote repository and after that we will commit changes.' The repository navigation bar includes links for 'Code', 'Issues', 'Pull requests', 'Marketplace', and 'Explore'. The main content area shows a file named 'one-more-file.txt' with the following content:

```
my-first-github-repository / feature / one-more-file.txt in feature-1
Cancel changes
Edit new file Preview
1 This file was created in the remote repository
```

Search or jump to... Pull requests Issues Marketplace Explore

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

feature-1

Commits on Aug 28, 2021

- Create one-more-file.txt
- Create another-file.txt

Commits on Aug 24, 2021

- File list.txt was created in the feature-1 branch
- Create hello-github.txt
- Update README.md
- Initial commit

SHA-1 hash: 27e4949

last commit

There are 6 commits in feature-1 branch in remote repository.

→ git log

```
commit d9f543913f018bf0d851dd8584a9372a13c44f43 (HEAD -> feature-1)
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:   Sat Aug 28 20:51:01 2021 +0530

    Modified another-file.txt locally

commit 065266f33a06b17bc8bb5232452d33467867a659 (origin/feature-1)
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:   Sat Aug 28 18:42:03 2021 +0530

    Create another-file.txt

commit e0cffcb157f5dccccc3b2b98599b2b908932ff0c0
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:   Tue Aug 24 16:59:55 2021 +0530

    File list.txt was created in the feature-1 branch

commit 3a3e202a8ffc7361245fd8416e496d423e697a37 (origin/main, origin/HEAD, main)
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:   Tue Aug 24 16:12:03 2021 +0530

    Create hello-github.txt

commit 6d0ad74d78ba29d04f54579352ea3d5ceb600bb4
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:   Tue Aug 24 13:07:24 2021 +0530

    Update README.md

commit 398b48e4ecae92a3ff076853d3ec39b764a98a07
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:   Tue Aug 24 13:03:00 2021 +0530
:
```

→ outdated information

we yet have not fetched changes from remote repository.

## Plan :

1. Fetch remote updates manually using "git fetch".
2. Merge updates manually using "git merge FETCH\_HEAD"

```

→ my-first-github-repository git:(feature-1) git fetch -v
POST git-upload-pack (144 bytes)
POST git-upload-pack (358 bytes)
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 874 bytes | 218.00 KiB/s, done.
From https://github.com/hema8Codes/my-first-github-repository
  065266f..27e4949  feature-1    -> origin/feature-1
= [up to date]      main        -> origin/main

```

But remote feature-1 branch was changed

master branch is up-to-date

→ contents of staging area

```

100644 b589ea26cb5d599a8eb468bf93ff5aa210159dc7 0
100644 eca15b869f12fd2a751b5b970dcbb3b0441b0ae37 0
100644 3776b8c0940c336c34a03f2542abb1f58d8cb9c0 0
100644 bca999f8e67cf145e5c2eb7b72fd6fc0acd9bdc 0

```

→ here we do not see one-more-file.txt

Locally staging area wasn't updated after fetch.

```

→ my-first-github-repository git:(feature-1) ls -la .git/objects
total 0
drwxr-xr-x@ 16 hemakshipandey  staff  512 Aug 28 21:04 .
drwxr-xr-x@ 15 hemakshipandey  staff  480 Aug 28 21:04 ..
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 18:43 06
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 20:51 22
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 21:04 27
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 18:43 47
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 21:04 59
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 20:51 5d
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 18:43 6a
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 21:04 6c
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 21:04 a1
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 18:43 a4
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 20:51 d9
drwxr-xr-x@  3 hemakshipandey  staff   96 Aug 28 20:51 ec
drwxr-xr-x@  2 hemakshipandey  staff   64 Aug 24 18:37 info
drwxr-xr-x@  4 hemakshipandey  staff  128 Aug 24 18:37 pack

```

→ list files in 27 folder

→ there are four folder that were created

→ gt contains new root tree object

→ This one contains tree object for "feature" folder

Note: After "git fetch" local Git has created remote commit object in the local Git repository, but we have not yet merged this commit into the local branch.

```
→ my-first-github-repository git:(feature-1) git cat-file -p 27e4949 → reading the  
tree 5922ce87c4e51b72d28868d7187a116fb5d2ef63  
parent 065266f33a06b17bc8bb5232452d33467867a659  
author Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com> 1630164224 +0530  
committer GitHub <noreply@github.com> 1630164224 +0530  
gpgsig -----BEGIN PGP SIGNATURE-----
```

```
wsBcBAABCQAQBQJhKLUBCRBK7hj40v3rIwAADJ4IAAhNply66KK/7cqYNBcUecwW  
VkgVoPiyaPxhzC+n7gPH7dXQkLBmYW0N/pYb2WHH686JDlqE8vZrQalXruVhFX7o  
Mi7AYS+SnRwf108F4j4YDC3/S65xSUEbr755PY8hygpXZmaN0wkSng70mMJIhtzj  
E4kzyAQn0IImgdEuDeRl4PzvfE9NPu2XX+lkgocQK1RqlKz4Ty+Q8qxRPxDC7+P1D  
NPr+aVZIVieq//C13mzy1hs6LNni/h1+RN4x5yndZTBmReyFmkaiYNrTcQi6o16E  
IXSsb92k01dsGWq7a0s54pBlzq0T4pmGhz4W192v0GF/gJhgUmrlpxfeYWNyB1E= ← Note:  
=kKUe  
-----END PGP SIGNATURE-----
```

Create one-more-file.txt

```
→ my-first-github-repository git:(feature-1) ls -la .git/objects/a1  
total 8  
drwxr-xr-x@ 3 hemakshipandey staff 96 Aug 28 21:04 . ← were created in tracking  
drwxr-xr-x@ 16 hemakshipandey staff 512 Aug 28 21:04 .. ← remote branches in the remote  
-r--r--r-- 1 hemakshipandey staff 60 Aug 28 21:04 67ae6354b88ed9c29ce4eeb0abb0b88230e03a ← repository and recreates them in the  
→ my-first-github-repository git:(feature-1) git cat-file -p a167ae  
This file was created in the remote repository ✓ ← local git repository  
→ my-first-github-repository git:(feature-1) git cat-file -t a167ae  
blob ✓  
→ my-first-github-repository git:(feature-1) |
```

During "git fetch"  
local git downloads all new  
Git objects such as commits,  
blobs and trees that  
were created in tracking  
remote branches in the remote  
repository and recreates them  
in the local git repository.

```
→ my-first-github-repository git:(feature-1) ls -la .git/objects/6c  
total 8  
drwxr-xr-x@ 3 hemakshipandey staff 96 Aug 28 21:04 .  
drwxr-xr-x@ 16 hemakshipandey staff 512 Aug 28 21:04 ..  
-r--r--r-- 1 hemakshipandey staff 125 Aug 28 21:04 104fb0fe292ed6c21048fe857a0f1012970747  
→ my-first-github-repository git:(feature-1) git cat-file -p 6c104f → contents of tree object  
100644 blob 6a2d7e4af29fd9c891b71c2e4ba0b7c398188a16 another-file.txt  
100644 blob 3776b8c0940c336c34a03f2542abb1f58d8cb9c0 list.txt  
100644 blob a167ae6354b88ed9c29ce4eeb0abb0b88230e03a one-more-file.txt } that points to  
→ my-first-github-repository git:(feature-1) git cat-file -t 6c104f } three files  
tree  
→ my-first-github-repository git:(feature-1) ls -la .git/objects/59  
total 8  
drwxr-xr-x@ 3 hemakshipandey staff 96 Aug 28 21:04 .  
drwxr-xr-x@ 16 hemakshipandey staff 512 Aug 28 21:04 ..  
-r--r--r-- 1 hemakshipandey staff 129 Aug 28 21:04 22ce87c4e51b72d28868d7187a116fb5d2ef63  
→ my-first-github-repository git:(feature-1) git cat-file -p 5922ce  
100644 blob b589ea26cb5d599a8eb468bf93ff5aa210159dc7 README.md  
040000 tree 6c104fb0fe292ed6c21048fe857a0f1012970747 feature  
100644 blob bca999f8e67cf145e5c2eb7b72fd6fc0acd9bdc hello-github.txt  
→ my-first-github-repository git:(feature-1) git cat-file -t 5922ce  
tree  
→ my-first-github-repository git:(feature-1) |
```

## Merge FETCH\_HEAD manually

We have just verified that after git fetch operation, git has created several objects in local git repository and now we can actually perform git merge of the FETCH\_HEAD. But before that let's have a look at the contents of the FETCH\_HEAD file.

```
→ my-first-github-repository git:(feature-1) cat .git/FETCH_HEAD
branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository
not-for-merge branch 'main' of https://github.com/hema8Codes/my-first-github-repository
→ my-first-github-repository git:(feature-1) git log
```

feature-1 branch locally and remotely are in sync.

```
commit d9f543913f018bf0d851dd8584a9372a13c44f43 (HEAD -> feature-1)
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:   Sat Aug 28 20:51:01 2021 +0530

    Modified another-file.txt locally

commit 065266f33a06b17bc8bb5232452d33467867a659
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:   Sat Aug 28 18:42:03 2021 +0530

    Create another-file.txt

commit e0cffcb157f5dcc3b2b98599b2b908932ff0c0
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:   Tue Aug 24 16:59:55 2021 +0530

    File list.txt was created in the feature-1 branch

commit 3a3e202a8ffc7361245fd8416e496d423e697a37 (origin/main, origin/HEAD, main)
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:   Tue Aug 24 16:12:03 2021 +0530

    Create hello-github.txt

commit 6d0ad74d78ba29d04f54579352ea3d5ceb600bb4
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:   Tue Aug 24 13:07:24 2021 +0530

    Update README.md

commit 398b48e4ecae92a3ff076853d3ec39b764a98a07
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:   Tue Aug 24 13:03:00 2021 +0530
:|
```

Local HEAD points to feature-1 branch but feature-1 branch points to this commit

But this commit is absent in remote repository and we don't see actually pointer to last commit in remote feature-1 branch that because we haven't yet merged remote changes into local feature-1 branch

```
→ my-first-github-repository git:(feature-1) git merge FETCH_HEAD
Merge made by the 'recursive' strategy. → three-way merge
feature/one-more-file.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 feature/one-more-file.txt
→ my-first-github-repository git:(feature-1) ls -la feature
```

```
→ my-first-github-repository git:(feature-1) ls -la feature listing contents  
total 24  
drwxr-xr-x@ 5 hemakshipandey staff 160 Aug 28 22:49 .  
drwxr-xr-x@ 6 hemakshipandey staff 192 Aug 28 22:49 ..  
-rw-r--r-- 1 hemakshipandey staff 82 Aug 28 20:49 another-file.txt  
-rw-r--r-- 1 hemakshipandey staff 23 Aug 28 13:43 list.txt  
-rw-r--r-- 1 hemakshipandey staff 47 Aug 28 22:49 one-more-file.txt  
→ my-first-github-repository git:(feature-1) git ls-files -s → file just created  
100644 b589ea26cb5d599a8eb468bf93ff5aa210159dc7 0 README.md  
100644 eca15b869f12fd2a751b5b970dcbb3b0441b0ae37 0 feature/another-file.txt  
100644 3776b8c0940c336c34a03f2542abb1f58d8cb9c0 0 feature/list.txt  
100644 a167ae6354b88ed9c29ce4eeb0abb0b88230e03a 0 feature/one-more-file.txt  
100644 bca999f8e67cf145e5c2eb7b72fd6fc0acd9bdc 0 hello-github.txt  
→ Same file can be seen in git staging area
```

```
→ my-first-github-repository git:(feature-1) git pull
hint: Pulling without specifying how to reconcile divergent branches is
hint: discouraged. You can squelch this message by running one of the following
hint: commands sometime before your next pull:
hint:
hint:   git config pull.rebase false  # merge (the default strategy)
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only      # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
```

## SHA1 hash of new last merge commit

```
commit 1d5eb0460ed4fd88a60dece20bc08078ffb67a3a (HEAD -> feature-1)
Merge: d9f5439 27e4949
Author: hemakshi1234 <mahe173fas@gmail.com>
Date: Sat Aug 28 22:49:08 2021 +0530

Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1

commit 27e4949c66bbe5e7db35af6cff4f2cdcf5573186 (origin/feature-1)
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Sat Aug 28 20:53:44 2021 +0530

Create one-more-file.txt

commit d9f543913f018bf0d851dd8584a9372a13c44f43
Author: hemakshi1234 <mahe173fas@gmail.com>
Date: Sat Aug 28 20:51:01 2021 +0530

Modified another-file.txt locally

commit 065266f33a06b17bc8bb5232452d33467867a659
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Sat Aug 28 18:42:03 2021 +0530

Create another-file.txt

commit e0cffcb157f5dcccc3b2b98599b2b908932ff0c0
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Tue Aug 24 16:59:55 2021 +0530

File list.txt was created in the feature-1 branch

commit 3a3e202a8ffc7361245fd8416e496d423e697a37 (origin/main, origin/HEAD, main)
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
:|
```

Now local HEAD points to feature-1 branch  
and here at the top you see  
new merge  
commit

and this new  
commit now point  
to two parents.

But please notice  
that remote feature-1 branch  
still points to this commit, it doesn't  
point to merge commit. That's  
because we have n't yet pushed  
our updates.

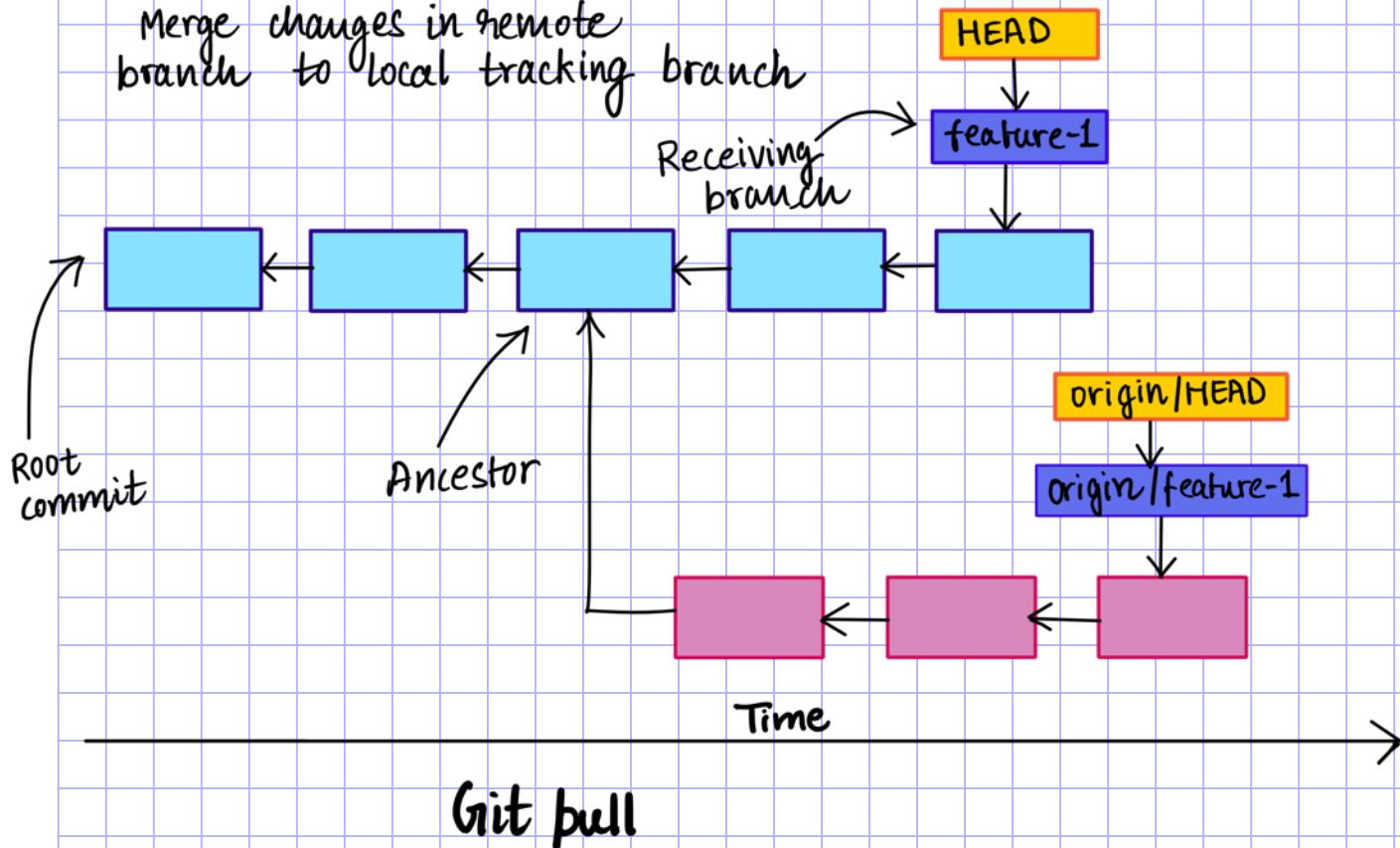
```
→ my-first-github-repository git:(feature-1) git cat-file -p 1d5eb046
tree b749005c051c785f56db25eee001fe6413420b0c
parent d9f543913f018bf0d851dd8584a9372a13c44f43
parent 27e4949c66bbe5e7db35af6cff4f2cdcf5573186
author hemakshi1234 <mahe173fas@gmail.com> 1630171148 +0530
committer hemakshi1234 <mahe173fas@gmail.com> 1630171148 +0530
```

```
Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1
→ my-first-github-repository git:(feature-1) |
```

## Git pull with 3-way merge

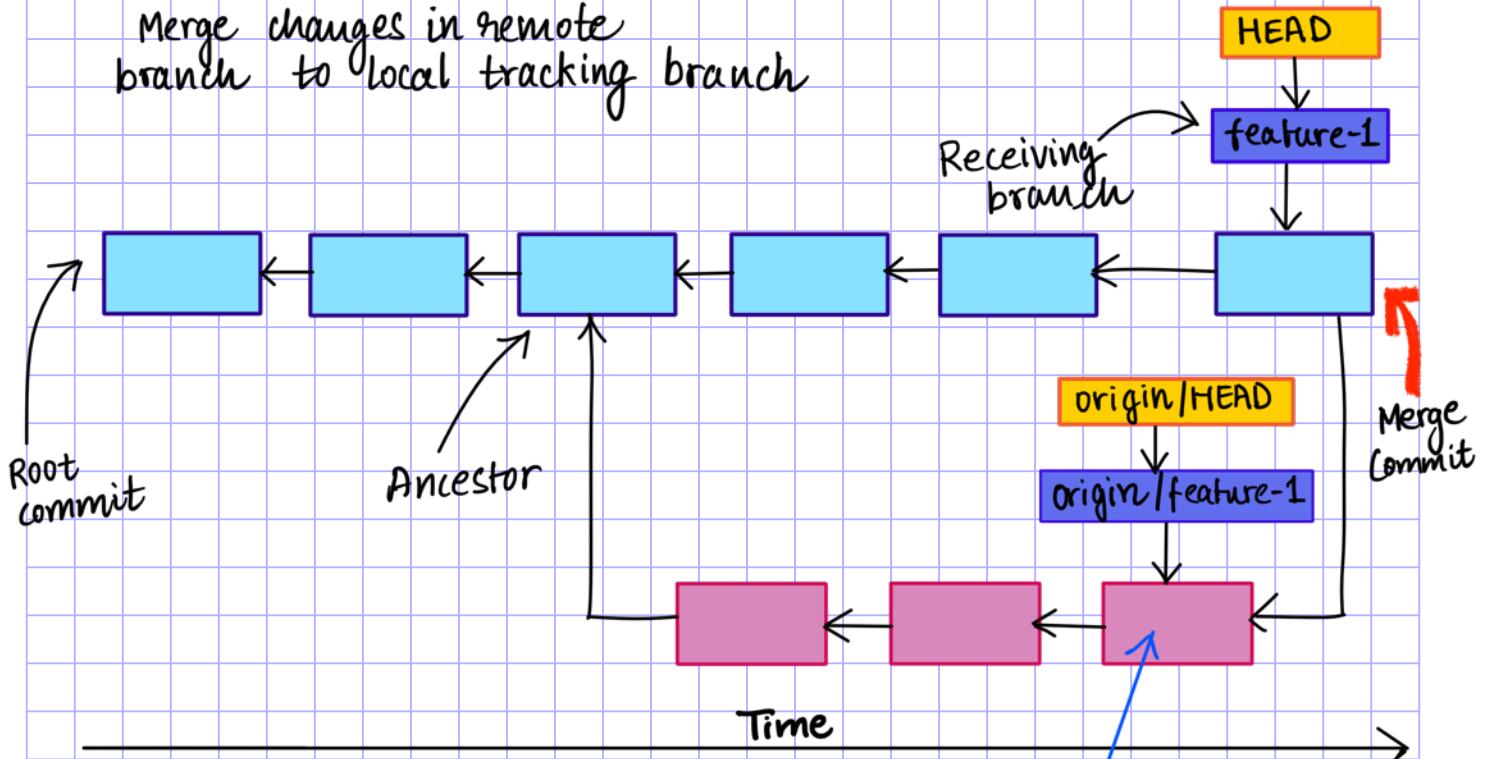
### Git pull

Merge changes in remote branch to local tracking branch



### Git pull

Merge changes in remote branch to local tracking branch



But please notice that remote feature-1 branch still points to this last commit here but local feature-1 branch points to last merge commit

# Resolving conflicts during Git pull

- Resolve merge conflict during "git pull" if files in the local and corresponding remote branch were changed in the same places.
- In this, we will try to create merge conflict during git pull operation and for that we will make changes in remote feature-1 branch and in the local feature-1 branch in the same file, in the same place.

hema8Codes / my-first-github-repository going to github and making changes in README.md file

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

feature-1 my-first-github-repository / README.md

hema8Codes Update README.md

Latest commit 6d8ad74 5 days ago History

1 contributor

4 lines (3 sloc) | 136 Bytes

Edit this file Raw Blame

my-first-github-repository

Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually

hema8Codes / my-first-github-repository

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

my-first-github-repository / README.md in feature-1

Cancel changes

Edit file Preview Spaces 2 Soft wrap

1 # my-first-github-repository  
2 Here is my first Github repository. Let's get started with GitHub  
3  
4 README.md file was just edited manually. One more edit at GitHub.  
5

Attach files by dragging & dropping, selecting or pasting them.

Commit changes

Update README.md

Add an optional extended description...

Commit directly to the feature-1 branch.

Create a new branch for this commit and start a pull request. Learn more about pull requests.

Commit changes Cancel

Committing changes in feature-1 branch in github

# Editing README.md file using nano editor

```
GNU nano 2.0.6          File: README.md
# my-first-github-repository
Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually. This file was modified locally.
```

[ Wrote 4 lines ]

AG Get Help ^O WriteOut ^R Read File ^Y Prev Page ^C Cut Text ^G Cur Pos
^X Exit AJ Justify ^W Where Is ^V Next Page AU Uncut Text ^T To Spell

```
→ my-first-github-repository git:(feature-1) nano README.md
→ my-first-github-repository git:(feature-1) x git status
On branch feature-1
Your branch is ahead of 'origin/feature-1' by 2 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md
```

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

```
→ my-first-github-repository git:(feature-1) x git commit -a -m "Modified README.md file locally"
[feature-1 798e243] Modified README.md file locally } committing changes in local repository
  1 file changed, 1 insertion(+), 1 deletion(-)
```

```
→ my-first-github-repository git:(feature-1) git log
```

```
commit 798e2432c8718d651169ec1db29414ed50f0893b (HEAD → feature-1)
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:  Sun Aug 29 16:41:35 2021 +0530

  Modified README.md file locally → But this commit is absent in remote repository

commit 1d5eb0460ed4fd88a60dece20bc08078ffb67a3a
Merge: d9f5439 27e4949
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:  Sat Aug 28 22:49:08 2021 +0530

  Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1

commit 27e4949c66bbe5e7db35af6cff4f2cdcf5573186 (origin/feature-1)
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:  Sat Aug 28 20:53:44 2021 +0530

  Create one-more-file.txt

commit d9f543913f018bf0d851dd8584a9372a13c44f43
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:  Sat Aug 28 20:51:01 2021 +0530

  Modified another-file.txt locally
```

Commits - hema8Codes/my-first-github-repository

github.com/hema8Codes/my-first-github-repository/commits/feature-1

hema8Codes / my-first-github-repository

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

feature-1

Commits on Aug 29, 2021

Update README.md  
hema8Codes committed 5 minutes ago

Commits on Aug 28, 2021

Create one-more-file.txt  
hema8Codes committed 20 hours ago

Create another-file.txt  
hema8Codes committed 22 hours ago

Commits on Aug 24, 2021

File list.txt was created in the feature-1 branch  
hema8Codes committed 5 days ago

Create hello-github.txt  
hema8Codes committed 5 days ago

Update README.md  
hema8Codes committed 5 days ago

Initial commit  
hema8Codes committed 5 days ago

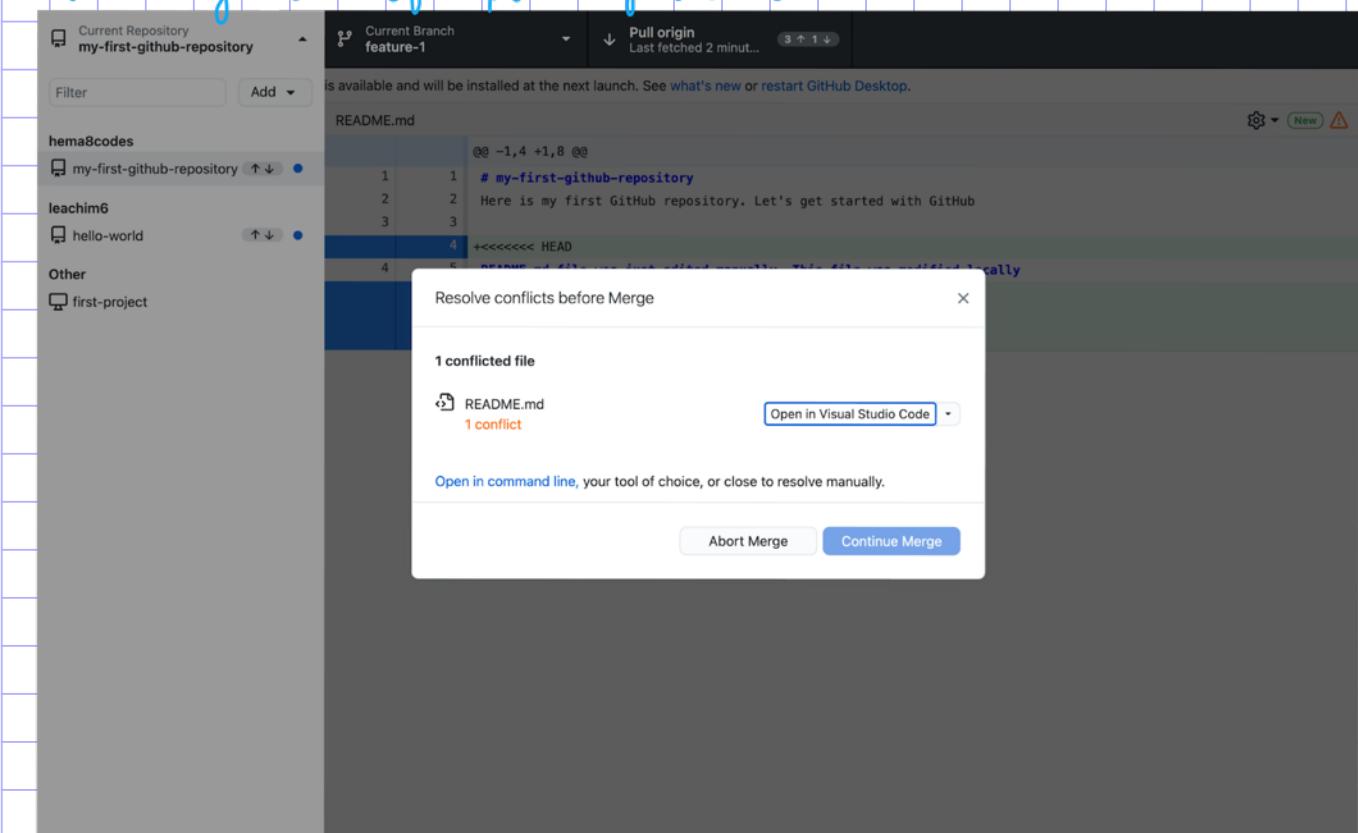
→ this commit is absent in local repository

```
→ my-first-github-repository git:(feature-1) git pull -v
POST git-upload-pack (144 bytes)
POST git-upload-pack (508 bytes)
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 760 bytes | 253.00 KiB/s, done.
From https://github.com/hema8Codes/my-first-github-repository
  27e4949..9b2f4cf  feature-1  -> origin/feature-1
 = [up to date]      main      -> origin/main
hint: Pulling without specifying how to reconcile divergent branches is
hint: discouraged. You can squelch this message by running one of the following
hint: commands sometime before your next pull:
hint:
hint:   git config pull.rebase false # merge (the default strategy)
hint:   git config pull.rebase true # rebase
hint:   git config pull.ff only    # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
→ my-first-github-repository git:(feature-1) x }
```

} git fetch has shown us that feature-1 branch in local repository was updated

} But during automatic merge process git has discovered that there are conflicts

# Observing status of repository at Github Desktop



## Conflicts in Visual Studio Code

A screenshot of Visual Studio Code showing a merge conflict in the 'README.md' file. The code editor has three tabs: [Preview] CHANGELOG\_7.8.md, another-file.txt, and README.md. The README.md tab shows a conflict between the local 'HEAD' (Current Change) and the remote 'feature-1' branch (Incoming Change). Handwritten notes explain: 'current change is change that was made in receiving branch.', 'feature-1 branch in Local Repository.', and 'This is from remote repository. → feature-1 branch from remote repository.' A callout box highlights the SHA1 hash of the remote commit: 'SHA1 hash of remote commit where this change was introduced in remote repository.'

Under the hood "git pull" performs "git merge FETCH\_HEAD". That's why here you see SHA1 hash of the last commit in corresponding branch.

observing status of staging area

```
→ my-first-github-repository git:(feature-1) ✘ git ls-files -s
100644 b589ea26cb5d599a8eb468bf93ff5aa210159dc7 1 README.md } 3 versions of
100644 797e3ecd6f35e8ebc4549071cbf37532a707be4f 2 README.md } README.md file
100644 eb78d170d91aa6845bdd7057fa08feeb75f180ed 3 README.md } due to conflicts
100644 eca15b869f12fd2a751b5b970dcbb3b0441b0ae37 0 feature/another-file.txt in file
100644 3776b8c0940c336c34a03f2542abb1f58d8cb9c0 0 feature/list.txt
100644 a167ae6354b88ed9c29ce4eeb0abb0b88230e03a 0 feature/one-more-file.txt
100644 bca999f8e67cf145e5c2eb7b72fd6fc0acd9bdc 0 hello-github.txt
```

```
→ my-first-github-repository git:(feature-1) ✘ git cat-file -p eb78d17
```

```
# my-first-github-repository
```

Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually. One more edit at GitHub.

```
→ my-first-github-repository git:(feature-1) ✘ git cat-file -p 797e3e
```

```
# my-first-github-repository
```

Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually. This file was modified locally

```
→ my-first-github-repository git:(feature-1) ✘ git cat-file -p b589e
```

```
# my-first-github-repository
```

Here is my first GitHub repository. Let's get started with GitHub

README.md file was just edited manually

```
→ my-first-github-repository git:(feature-1) ✘ cat README.md
```

```
# my-first-github-repository
```

Here is my first GitHub repository. Let's get started with GitHub

<<<<< HEAD

README.md file was just edited manually. This file was modified locally

=====

README.md file was just edited manually. One more edit at GitHub.

```
>>>>> 9b2f4cf9aef37f9530d3189ac460c2fb90f78c
```

```
→ my-first-github-repository git:(feature-1) ✘
```

```
→ my-first-github-repository git:(feature-1) ✘ git status
```

On branch feature-1

Your branch and 'origin/feature-1' have diverged,  
and have 3 and 1 different commits each, respectively.  
(use "git pull" to merge the remote branch into yours)

You have unmerged paths.

(fix conflicts and run "git commit")

(use "git merge --abort" to abort the merge)

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: README.md

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

```
→ my-first-github-repository git:(feature-1) ✘ git ls-files -s
```

```
100644 b589ea26cb5d599a8eb468bf93ff5aa210159dc7 1 README.md } still 3 different versions
100644 797e3ecd6f35e8ebc4549071cbf37532a707be4f 2 README.md } of the file README.md.
100644 eb78d170d91aa6845bdd7057fa08feeb75f180ed 3 README.md
100644 eca15b869f12fd2a751b5b970dcbb3b0441b0ae37 0 feature/another-file.txt
100644 3776b8c0940c336c34a03f2542abb1f58d8cb9c0 0 feature/list.txt
100644 a167ae6354b88ed9c29ce4eeb0abb0b88230e03a 0 feature/one-more-file.txt
100644 bca999f8e67cf145e5c2eb7b72fd6fc0acd9bdc 0 hello-github.txt
```

```
→ my-first-github-repository git:(feature-1) ✘ git commit -a
```

After editing in VSCode  
and saving file

Git still shows that there are conflicts  
because we have only saved modified  
file with resolved conflict in the  
working directory.

In staging area there are  
still 3 different versions  
of the file README.md.

committing changes and  
adding to staging area

```
commit a44f055f50c0d16737673049069fc83a70829363 (HEAD -> feature-1)
Merge: 798e243 9b2f4cf
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:  Sun Aug 29 16:56:49 2021 +0530

Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1

commit 798e2432c8718d651169ec1db29414ed50f0893b
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:  Sun Aug 29 16:41:35 2021 +0530

    Modified README.md file locally

commit 9b2f4cf9aefa37f9530d3189ac460c2fb90f78c (origin/feature-1) → remote feature-1 branch still points to this commit as we haven't pushed our changes to local repository to remote repository.
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:  Sun Aug 29 16:37:53 2021 +0530

    Update README.md

commit 1d5eb0460ed4fd88a60dece20bc08078ffb67a3a
Merge: d9f5439 27e4949
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:  Sat Aug 28 22:49:08 2021 +0530

Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1

commit 27e4949c66bbe5e7db35af6cff4f2cdcf5573186
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:  Sat Aug 28 20:53:44 2021 +0530

    Create one-more-file.txt

commit d9f543913f018bf0d851dd8584a9372a13c44f43
:
```

# Pushing to remote repository

notice that during git

pull operation it has

hint: Pulling without specifying how to reconcile divergent branches is not asked us

hint: discouraged. You can squelch this message by running one of the following for

hint: commands sometime before your next pull:

hint:

hint: git config pull.rebase false # merge (the default strategy) any credentials.

hint: git config pull.rebase true # rebase that because Anybody

hint: git config pull.ff only # fast-forward only in the world can fetch

hint: from the public repositories

hint: You can replace "git config" with "git config --global" to set a default

hint: preference for all repositories. You can also pass --rebase, --no-rebase,

hint: or --ff-only on the command line to override the configured default per

hint: invocation. local feature-1 branch is already up to date

Already up to date.

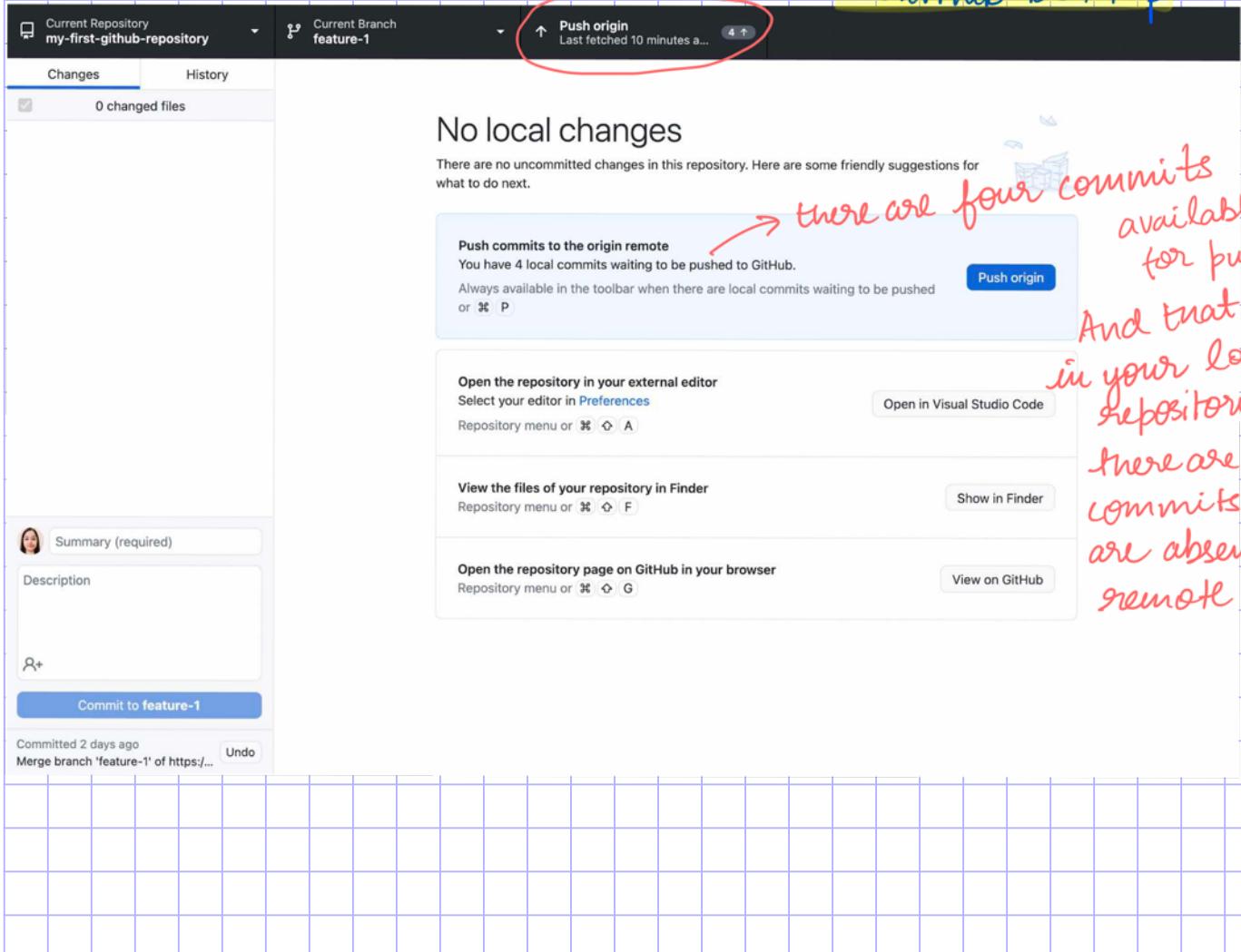
→ my-first-github-repository git:(feature-1) git remote -v } URLs

origin https://github.com/hema8Codes/my-first-github-repository.git (fetch)

origin https://github.com/hema8Codes/my-first-github-repository.git (push)

→ Now there are changes in the local repository that are absent in the remote repository.

GitHub Desktop



→ git log

```
commit a44f055f50c0d16737673049069fc83a70829363 (HEAD -> feature-1)
Merge: 798e243 9b2f4cf
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:  Sun Aug 29 16:56:49 2021 +0530

  Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1

commit 798e2432c8718d651169ec1db29414ed50f0893b
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:  Sun Aug 29 16:41:35 2021 +0530

  Modified README.md file locally

commit 9b2f4cf9aefa37f9530d3189ac460c2fbb90f78c (origin/feature-1)
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:  Sun Aug 29 16:37:53 2021 +0530

  Update README.md

commit 1d5eb0460ed4fd88a60dece20bc08078ffb67a3a
Merge: d9f5439 27e4949
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:  Sat Aug 28 22:49:08 2021 +0530

  Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1

commit 27e4949c66bbe5e7db35af6cff4f2cdcf5573186
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:  Sat Aug 28 20:53:44 2021 +0530

  Create one-more-file.txt

commit d9f543913f018bf0d851dd8584a9372a13c44f43
:|
```

There are different authors in different commits

If you have write access to the remote repository you can push commit with any author.

→ my-first-github-repository git:(feature-1) git push -v

Pushing to https://github.com/hema8Codes/my-first-github-repository.git  
Enumerating objects: 25, done.

Counting objects: 100% (21/21), done.

Delta compression using up to 8 threads

Compressing objects: 100% (13/13), done.

Writing objects: 100% (13/13), 1.73 KiB | 1.73 MiB/s, done.

Total 13 (delta 2), reused 0 (delta 0), pack-reused 0

POST git-receive-pack (1945 bytes)

remote: Resolving deltas: 100% (2/2), completed with 1 local object.

To https://github.com/hema8Codes/my-first-github-repository.git

9b2f4cf..a44f055 feature-1 -> feature-1 *a remote feature-1 branch now points to this commit*

updating local tracking ref 'refs/remotes/origin/feature-1'

→ my-first-github-repository git:(feature-1)

```
commit a44f055f50c0d16737673049069fc83a70829363 (HEAD -> feature-1, origin/feature-1)
Merge: 798e243 9b2f4cf
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:  Sun Aug 29 16:56:49 2021 +0530

  Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1

commit 798e2432c8718d651169ec1db29414ed50f0893b
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:  Sun Aug 29 16:41:35 2021 +0530

  Modified README.md file locally

commit 9b2f4cf9aefa37f9530d3189ac460c2fbb90f78c
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date:  Sun Aug 29 16:37:53 2021 +0530

  Update README.md
```

A screenshot of a GitHub repository page for 'hema8Codes / my-first-github-repository'. The 'Code' tab is selected. A dropdown menu shows 'feature-1'. The commit history is displayed in two sections: 'Commits on Aug 29, 2021' and 'Commits on Aug 28, 2021'. In the first section, there are three commits: 'Merge branch 'feature-1'' (hash a44f055), 'Modified README.md file locally', and 'Update README.md'. In the second section, there are four commits: 'Merge branch 'feature-1'' (hash 1d5eb04), 'Create one-more-file.txt', 'Modified another-file.txt locally', and 'Create another-file.txt'. A red circle highlights the commit hash 'a44f055' in the first commit of the first section. The URL 'https://github.com/hema8Codes/my-first-github-repository/commit/a44f055f50c0d16737673049069fc83a70829363' is shown at the bottom.

that means that all changes from local repository were incorporated in remote repository.

```
→ my-first-github-repository git:(feature-1) cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
    ignorecase = true
    precomposeunicode = true
[remote "origin"]
    url = https://github.com/hema8Codes/my-first-github-repository.git
    fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
    remote = origin
    merge = refs/heads/main
[branch "feature-1"]
    remote = origin
    merge = refs/heads/feature-1
→ my-first-github-repository git:(feature-1)
```

# Commit under another author

git commit -m "Another local change" README.md

Changes to be committed:

- new file: CHANGELOG\_7.8.md
- new file: another-file.txt
- modified: README.md

diff --git a/README.md b/README.md

1 # my-first-github-repository

2 Here is my first GitHub repository. Let's get started with GitHub

3

4 This line was modified manually during conflicts resolution

5

6 New line was added locally

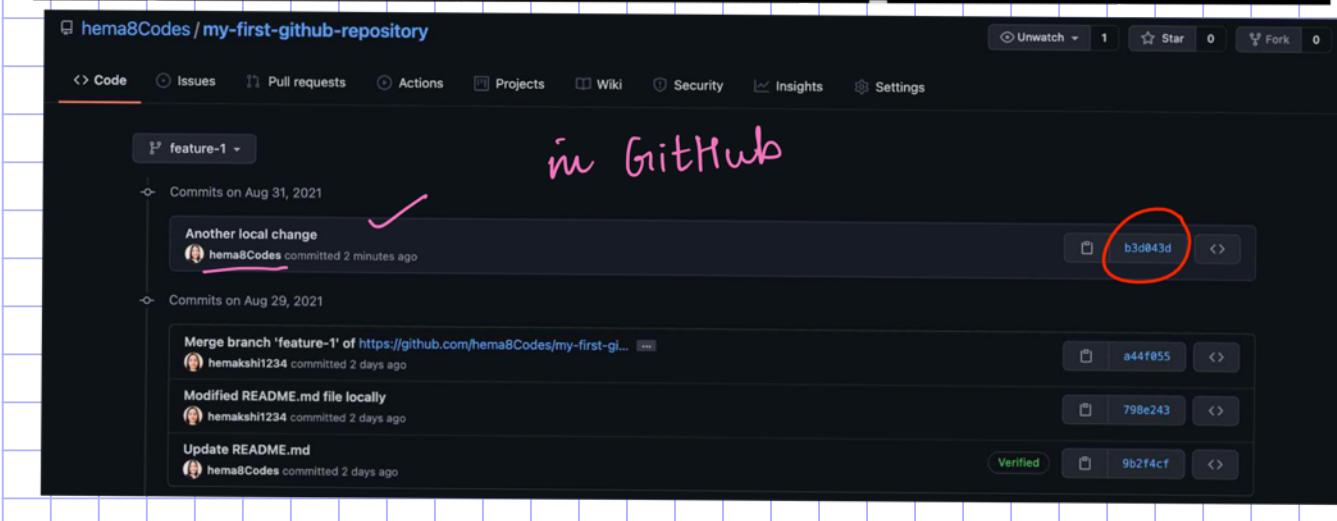
7

→ editing README.md file locally  
→ committing the changes

There is single change ready to be pushed.

```
→ my-first-github-repository git:(feature-1) git push -v
Pushing to https://github.com/hema8Codes/my-first-github-repository.git
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 380 bytes | 380.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
POST git-receive-pack (558 bytes)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/hema8Codes/my-first-github-repository.git
 a44f055..b3d043d  feature-1 -> feature-1
updating local tracking ref 'refs/remotes/origin/feature-1'
```

→ now changes were successfully pushed



→ git log

```
commit b3d043d54eb9decc6774e042f317849cd196de7 (HEAD -> feature-1, origin/feature-1)
Author: hema8Codes <developer8hema@gmail.com>
Date: Tue Aug 31 18:37:36 2021 +0530

    Another local change

commit a44f055f50c0d16737673049069fc83a70829363
Merge: 798e243 9b2f4cf
Author: hemakshi1234 <mahe173fas@gmail.com>
Date: Sun Aug 29 16:56:49 2021 +0530

    Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1

commit 798e2432c8718d651169ec1db29414ed50f0893b
Author: hemakshi1234 <mahe173fas@gmail.com>
Date: Sun Aug 29 16:41:35 2021 +0530

    Modified README.md file locally

commit 9b2f4cf9aef937f9530d3189ac460c2fb90f78c
Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>
Date: Sun Aug 29 16:37:53 2021 +0530

    Update README.md

commit 1d5eb0460ed4fd88a60dece20bc08078ffb67a3a
Merge: d9f5439 27e4949
Author: hemakshi1234 <mahe173fas@gmail.com>
Date: Sat Aug 28 22:49:08 2021 +0530

    Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1

commit 27e4949c66bbe5e7db35af6cff4f2cdcf5573186
:|
```

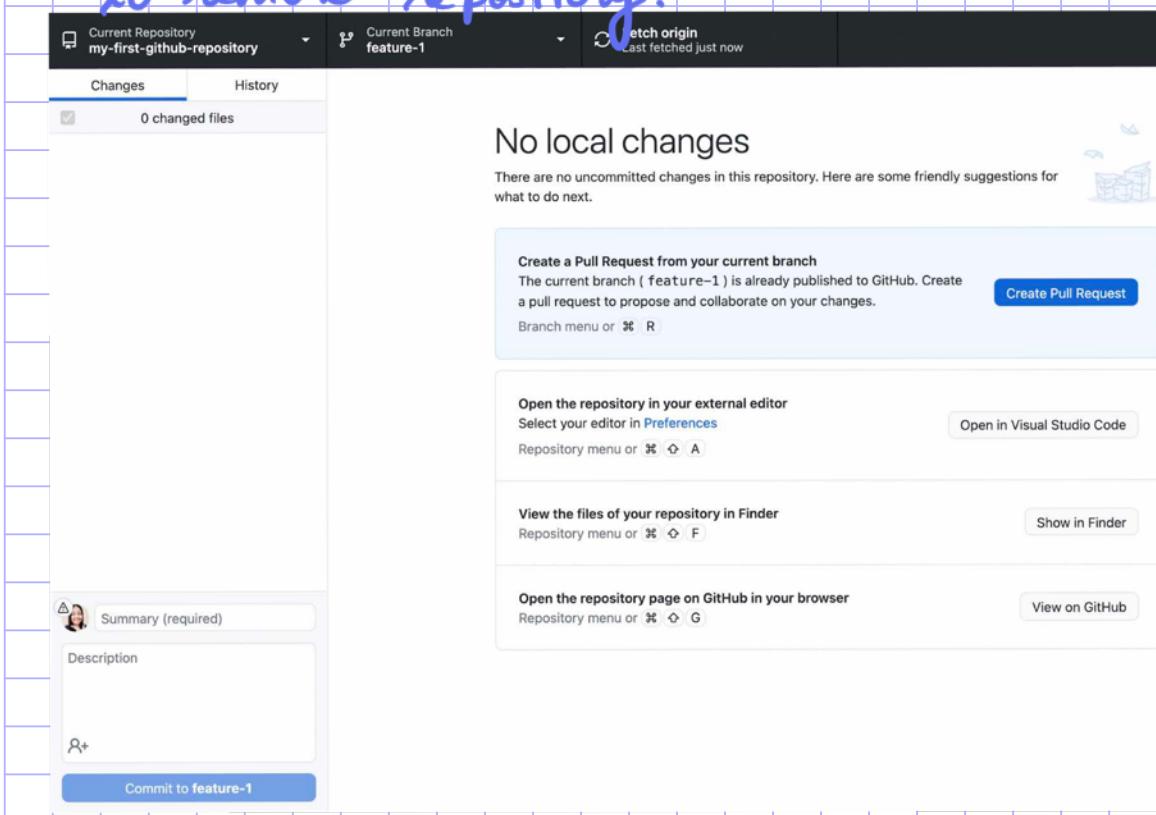
→ now our local feature-1 branch is in sync with remote feature-1 branch

→ same information can be seen in last commit.  
→ Local repository user.name and user.email settings override global Git settings.

→ Remote and Local branches are in sync now

→ Remote and local repositories are now consistent.

→ All changes from remote repository were pulled to local repository and all local changes were pushed to remote repository.



## GitHub Desktop

If we click on Fetch origin button, you will see that after fetch, you will not see any updates. And that means that you are in sync with remote repository.

```

→ my-first-github-repository git:(feature-1) git pull -v
POST git-upload-pack (144 bytes)
From https://github.com/hema8Codes/my-first-github-repository
 = [up to date]      feature-1  -> origin/feature-1
 = [up to date]      main       -> origin/main
Already up to date.
→ my-first-github-repository git:(feature-1) git push -v
Pushing to https://github.com/hema8Codes/my-first-github-repository.git
To https://github.com/hema8Codes/my-first-github-repository.git
 = [up to date]      feature-1 -> feature-1
updating local tracking ref 'refs/remotes/origin/feature-1'
Everything up-to-date
→ my-first-github-repository git:(feature-1) cat .git/refs/remotes/origin/feature-1
b3d043d54eb9decc6774e042f317849cd196de7 ← SHA1 hash of the last commit in remote
→ my-first-github-repository git:(feature-1) cat .git/refs/heads/feature-1
b3d043d54eb9decc6774e042f317849cd196de7 ← same SHA1 hash in local
→ my-first-github-repository git:(feature-1) feature-1 branch.

```

} we are upto date in both branches, feature branch and master branch and there is nothing to pull

} There is nothing to push and both branches are upto date.

Remote feature-1 branch and local feature-1 branch point to the same commit.

## Create remote branch based on local branch

→ Create new remote branch based on the existing local branch.



```

→ my-first-github-repository git:(feature-1) git branch
[3] + 3595 suspended git branch
→ my-first-github-repository git:(feature-1) git branch -vv
[4] + 3615 suspended git branch -vv
→ my-first-github-repository git:(feature-1)

```

list of branches

list of corresponding remote branch for every local branch.

```

* feature-1 b3d043d [origin/feature-1] Another local change
  main      3a3e202 [origin/main] Create hello-github.txt
(END)

```

## Our plan

1. Create new local "feature-2" branch.
2. Make some changes and commit them locally in the new branch.
3. Create new remote "feature-2" branch from the local "feature-2" branch by using "git push -u" command.

```
→ my-first-github-repository git:(feature-1) git checkout -b feature-2
Switched to a new branch 'feature-2' ↗ creating new branch
→ my-first-github-repository git:(feature-2) git branch
[5] + 3720 suspended git branch ↗ we will see three branches
on the list
→ my-first-github-repository git:(feature-2) git branch -r ↗ This will list remote
branches
[6] + 3744 suspended git branch -r
→ my-first-github-repository git:(feature-2) git branch -vv
```

```
[7] + 3761 suspended git branch -vv
```

```
feature-1
* feature-2
main
(END)
```

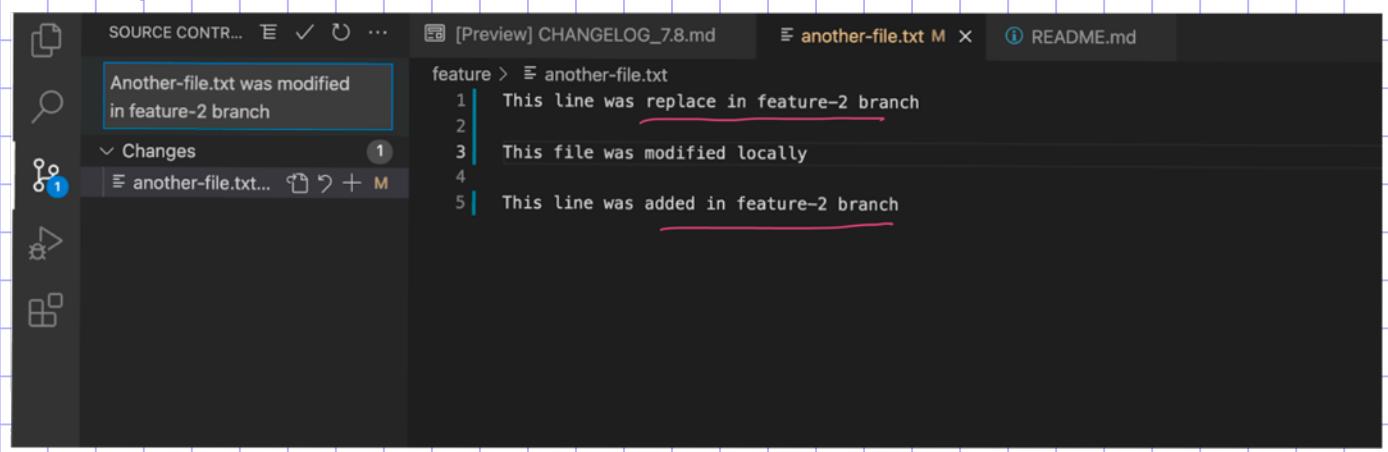
```
origin/HEAD -> origin/main
origin/feature-1
origin/main → we do not see
(END)           newly created feature-2 branch
on the list
```

```
feature-1 b3d043d [origin/feature-1] Another local change
* feature-2 b3d043d Another local change
  main      3a3e202 [origin/main] Create hello-github.txt
(END)
```

And that means when you create any local branch, new remote branch will not get created. We need to do that manually.

→ we will see feature-2 here on the list but there is no corresponding remote feature-2 branch here.

Making some changes in our project in feature-2 branch



```
→ my-first-github-repository git:(feature-2) git log
```

```
[8] + 3981 suspended git log
```

```
→ my-first-github-repository git:(feature-2) git branch -a
```

```
[9] + 3999 suspended git branch -a
```

```
commit ae432c9c6e2b3ed57793839e70aaa9313717ab29 (HEAD -> feature-2)
```

```
Author: hema8Codes <developer8hema@gmail.com>
```

```
Date: Tue Aug 31 22:05:16 2021 +0530
```

```
Another-file.txt was modified in feature-2 branch
```

→ last commit

→ local HEAD points to feature-2  
branch and feature-2  
branch points to last  
commit. But remotely in  
remote repository feature-2  
branch is still absent.

```
commit b3d043d54eb9decccd6774e042f317849cd196de7 (origin/feature-1, feature-1)
```

```
Author: hema8Codes <developer8hema@gmail.com>
```

```
Date: Tue Aug 31 18:37:36 2021 +0530
```

```
Another local change
```

```
commit a44f055f50c0d16737673049069fc83a70829363
```

```
Merge: 798e243 9b2f4cf
```

```
Author: hemakshi1234 <mahe173fas@gmail.com>
```

```
Date: Sun Aug 29 16:56:49 2021 +0530
```

```
Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1
```

```
commit 798e2432c8718d651169ec1db29414ed50f0893b
```

```
Author: hemakshi1234 <mahe173fas@gmail.com>
```

```
Date: Sun Aug 29 16:41:35 2021 +0530
```

```
Modified README.md file locally
```

```
commit 9b2f4cf9aefa37f9530d3189ac460c2fbb90f78c
```

```
Author: Hemakshi Pandey <89300996-hema8Codes@users.noreply.github.com>
```

```
Date: Sun Aug 29 16:37:53 2021 +0530
```

```
Update README.md
```

```
commit 1d5eb0460ed4fd88a60dece20bc08078ffb67a3a
```

```
Merge: d9f5439 27e4949
```

```
:|
```

```
feature-1
```

```
* feature-2
```

```
main
```

```
remotes/origin/HEAD -> origin/main
```

```
remotes/origin/feature-1
```

```
remotes/origin/main
```

```
(END)
```

we will still see feature-2 branch locally and there is no corresponding remote feature-2 branch.

Let now create new remote feature-2 branch based on newly created local feature-2 branch.

git push --set-upstream origin feature-2

option      name of  
remote server

instead of this  
option you can use -u

you need to  
specify name  
of new Remote  
branch (matching  
the name of  
local branch)

```
→ my-first-github-repository git:(feature-2) git push -v -u origin feature-2
```

Pushing to https://github.com/hema8Codes/my-first-github-repository.git

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 545 bytes | 545.00 KiB/s, done.

Total 4 (delta 0), reused 0 (delta 0), pack-reused 0

POST git-receive-pack (723 bytes)

remote:

remote: Create a pull request for 'feature-2' on GitHub by visiting:

remote: https://github.com/hema8Codes/my-first-github-repository/pull/new/feature-2

remote:

To https://github.com/hema8Codes/my-first-github-repository.git  
\* [new branch] feature-2 -> feature-2 remote feature-2 branch was successfully created.

Branch 'feature-2' set up to track remote branch 'feature-2' from 'origin'.

updating local tracking ref 'refs/remotes/origin/feature-  
branch'

→ my-first-github-repository git:(feature-2)

branch is branching  
branch,

```
→ my-first-github-repository git:(feature-2) git branch -vv
```

[10] + 4080 suspended git branch -vv

```
feature-1 b3d043d [origin/feature-1] Another local change
* feature-2 ae432c9 [origin/feature-2] Another file.txt was modified in feature-2 branch
  main      3a3e202 [origin/main] Create hello-github.txt
(END)
```

→ Local feature-2 branch has corresponding remote feature-2 branch

→ When local branch has already corresponding remote branch you need simply enter "git push" to push changes to remote.

A screenshot of the VS Code interface. The left sidebar shows a tree view with 'Changes' expanded, showing a file named 'another-file.txt'. The main editor tab is titled '[Preview] CHANGELOG\_7.8.md'. A handwritten note in pink ink is overlaid on the screen, pointing to the commit message in the editor:

→ modifying another-file.txt, saving file  
and committing changes in VS Code

```
→ my-first-github-repository git:(feature-2) git push -v
Pushing to https://github.com/hema8Codes/my-first-github-repository.git
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 526 bytes | 526.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
POST git-receive-pack (704 bytes)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/hema8Codes/my-first-github-repository.git
  ae432c9..e07bf88  feature-2 -> feature-2
updating local tracking ref 'refs/remotes/origin/feature-2'
→ my-first-github-repository git:(feature-2) git log
```

No need to specify name of the server and name of the remote branch because Local "feature-2" branch is already tracking branch and knows its remote branch.

→ git log

```
commit e07bf887b287eb9ce880b1df20b3b1b80bb136a1 (HEAD -> feature-2, origin/feature-2)
```

Author: hema8Codes <developer8hema@gmail.com>  
Date: Tue Aug 31 22:18:18 2021 +0530

Another change in feature-2 branch

```
commit ae432c9c6e2b3ed57793839e70aaa9313717ab29
```

Author: hema8Codes <developer8hema@gmail.com>  
Date: Tue Aug 31 22:05:16 2021 +0530

Another-file.txt was modified in feature-2 branch

```
commit b3d043d54eb9decccd6774e042f317849cd196de7 (origin/feature-1, feature-1)
```

Author: hema8Codes <developer8hema@gmail.com>  
Date: Tue Aug 31 18:37:36 2021 +0530

Another local change

```
commit a44f055f50c0d16737673049069fc83a70829363
```

Merge: 798e243 9b2f4cf  
Author: hemakshi1234 <mahe173fas@gmail.com>  
Date: Sun Aug 29 16:56:49 2021 +0530

Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1

```
commit 798e2432c8718d651169ec1db29414ed50f0893b
```

Author: hemakshi1234 <mahe173fas@gmail.com>  
Date: Sun Aug 29 16:41:35 2021 +0530

Modified README.md file locally

```
commit 9b2f4cf9aefaa37f9530d3189ac460c2fb90f78c
```

Author: Hemakshi Pandey <89300996+hema8Codes@users.noreply.github.com>

:|

local HEAD points to feature-2 branch  
and remote feature-2 branch located  
at origin server points to the same  
commit

hema8Codes / my-first-github-repository

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

feature-2

Commits on Aug 31, 2021

- Another change in feature-2 branch  
hema8Codes committed 3 minutes ago → author of commit
- Another-file.txt was modified in feature-2 branch  
hema8Codes committed 16 minutes ago
- Another local change  
hema8Codes committed 4 hours ago

Commits on Aug 29, 2021

- Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-2  
hemakshi1234 committed 2 days ago
- Modified README.md file locally  
hemakshi1234 committed 2 days ago
- Update README.md  
hema8Codes committed 2 days ago

Commits on Aug 28, 2021

Observing changes in  
remote repository  
GitHub

hema8Codes / my-first-github-repository

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

feature-2 my-first-github-repository / feature / another-file.txt

hema8Codes Another change in feature-2 branch

2 contributors

7 lines (4 sloc) | 160 Bytes

```
1 This line was replace in feature-2 branch
2
3 This file was modified locally
4
5 This line was added in feature-2 branch
6
7 Another modification in the feature-2 branch
```

Raw Blame Go to file ...

Latest commit e07bf88 3 minutes ago History

fill push to  
remote repository

```
→ my-first-github-repository git:(feature-2) cat .git/config
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
  ignorecase = true
  precomposeunicode = true
[remote "origin"]
  url = https://github.com/hema8Codes/my-first-github-repository.git
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
  remote = origin
  merge = refs/heads/main
[branch "feature-1"]
  remote = origin
  merge = refs/heads/feature-1
[branch "feature-2"]
  remote = origin
  merge = refs/heads/feature-2
→ my-first-github-repository git:(feature-2)
```

You can create remote branch based on local branch.  
and for that you need to use git push command  
with option -u.

git push -u origin <REMOTE BRANCH NAME>

# Update tracking statuses of the branches

→ How to refresh local tracking statuses if remote branch was deleted.

## Update tracking statuses

1. Create new branch **temp** in the remote repository
2. Checkout this new remote branch locally using **git checkout temp**. Local tracking branch **temp** will be created
3. Delete **temp** branch in the remote repository
4. Use **git fetch**. After fetching observe tracking statuses **git branch -vv** You will see that local temp branch still tracks remote temp branch
5. Update tracking statuses using **git remote update origin --prune**
6. Observe tracking statuses again using **git branch -vv**

Creating new branch - **temp** in remote repository.

The screenshot shows a GitHub repository page for 'hema8Codes / my-first-github-repository'. The 'Code' tab is selected. A dropdown menu is open over the 'feature-2' branch, showing the option 'Create branch: temp from 'feature-2''. A modal window titled 'Switch branches/tags' also displays the 'temp' branch name. Below the dropdown, the commit history shows three recent commits: 'Merge branch 'feature-1'' (commit e07bf88), 'Modified README.md file locally' (commit ae432c9), and 'Update README.md' (commit b3d043d). The commits were made by 'hemakshi1234' and 'hema8Codes' on August 29, 2021.

→ new temp branch was created

A screenshot of a GitHub repository page for 'hema8Codes/my-first-github-repository'. The 'temp' branch is currently selected. Handwritten notes on the page say: 'now temp is currently selected as temp branch'. Below the branch selection, there is a list of commits from the 'feature-2' branch. On the right side of the page, there are sections for 'About', 'Releases', and 'Packages'.

listing all branches →  
in terminal. Now we  
don't see newly created  
remote branch on the list.

We need to perform fetch operation

```
feature-1
* feature-2
main
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/feature-2
remotes/origin/main
(END)
```

```
→ my-first-github-repository git:(feature-2) git branch -a
```

```
[12] + 4356 suspended git branch -a
```

```
→ my-first-github-repository git:(feature-2) git fetch
```

```
From https://github.com/hema8Codes/my-first-github-repository
```

```
* [new branch] temp -> origin/temp
```

```
→ my-first-github-repository git:(feature-2) git branch -a
```

After performing git fetch.

If we list branches again  
we will see the temp branch  
at origin and we will not  
see this branch locally.

This is because fetch operation  
only writes remote changes to  
local git repository.

```
feature-1
* feature-2
main
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/feature-2
remotes/origin/main
remotes/origin/temp
(END)
```

```
→ my-first-github-repository git:(feature-2) git checkout temp  
Branch 'temp' set up to track remote branch 'temp' from 'origin'.  
Switched to a new branch 'temp'  
→ my-first-github-repository git:(temp) git branch -a
```

[14] + 4442 suspended git branch -a

we need to checkout newly created  
remote branch using :-  
git checkout temp

Now the local temp branch  
tracks remote temp branch

```
feature-1  
feature-2  
main  
* temp → local temp branch  
remotes/origin/HEAD -> origin/main  
remotes/origin/feature-1  
remotes/origin/feature-2  
remotes/origin/main  
remotes/origin/temp  
(END)
```

```
→ my-first-github-repository git:(temp) git branch -vv
```

[15] + 4467 suspended git branch -vv

```
feature-1 b3d043d [origin/feature-1] Another local change  
feature-2 e07bf88 [origin/feature-2] Another change in feature-2 branch  
main 3a3e202 [origin/main] Create hello-github.txt  
* temp e07bf88 [origin/temp] Another change in feature-2 branch  
(END)
```

↳ local temp branch is tracking branch.

→ Deleting temp branch in remote repository

The screenshot shows a GitHub repository page for 'hema8Codes / my-first-github-repository'. The 'Code' tab is selected. The page displays three sections: 'Default branch', 'Your branches', and 'Active branches'. In the 'Default branch' section, there is one branch named 'main'. In the 'Your branches' section, there are three branches: 'temp', 'feature-2', and 'feature-1'. The 'temp' branch was updated 22 minutes ago by hema8Codes. The 'feature-2' branch was updated 22 minutes ago by hema8Codes. The 'feature-1' branch was updated 4 hours ago by hema8Codes. Each branch entry includes a 'New pull request' button and a 'Delete this branch' button. In the 'Active branches' section, there are three branches: 'temp', 'feature-2', and 'feature-1', which correspond to the ones in the 'Your branches' section. Each branch entry also includes a 'New pull request' button and a 'Delete this branch' button. The GitHub interface includes standard navigation links like 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings' at the top, and a search bar for branches at the top right.

But after deleting  
temp branch in  
remote repository  
we still see temp  
branch on the list

```
feature-1
feature-2
main
* temp
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/feature-2
remotes/origin/main
remotes/origin/temp
(END)
```

→ my-first-github-repository git:(temp) git branch -a

[16] + 4508 suspended git branch -a

→ my-first-github-repository git:(temp) git branch -vv

[17] + 4525 suspended git branch -vv

→ my-first-github-repository git:(temp) git fetch → fetching changes

→ my-first-github-repository git:(temp) git fetch -v

POST git-upload-pack (144 bytes)

From https://github.com/hema8Codes/my-first-github-repository

= [up to date]	feature-1	-> origin/feature-1	}
= [up to date]	feature-2	-> origin/feature-2	}
= [up to date]	main	-> origin/main	

→ my-first-github-repository git:(temp) git branch -vv

[18] + 4581 suspended git branch -vv

```
feature-1 b3d043d [origin/feature-1] Another local change
feature-2 e07bf88 [origin/feature-2] Another change in feature-2 branch
main      3a3e202 [origin/main] Create hello-github.txt
* temp     e07bf88 [origin/temp] Another change in feature-2 branch
(END)
```

→ we still see that local temp branch tracks remote  
origin / temp branch and that's because we haven't yet  
entered git fetch command and we haven't yet fetched  
changes from remote.

```
feature-1 b3d043d [origin/feature-1] Another local change
feature-2 e07bf88 [origin/feature-2] Another change in feature-2 branch
main      3a3e202 [origin/main] Create hello-github.txt
* temp     e07bf88 [origin/temp] Another change in feature-2 branch
(END)
```

→ we still see here that local temp branch  
tracks remote temp branch

But remote "temp" branch was  
deleted!

If we go back to remote repository and click on all branches, we see only three branches :-

A screenshot of a GitHub repository page for 'hema8Codes / my-first-github-repository'. The 'All branches' tab is selected. Three branches are listed: 'main' (Default), 'feature-2', and 'feature-1'. Each branch has a small icon, the last updated time, and the author's name. There are also edit and delete icons next to each branch entry.

There is no temp branch. But local temp branch still tracks remote temp branch.  
And That's where this command comes in :-

git remote update origin --prune

→ my-first-github-repository git:(temp) git remote update origin →  
Fetching origin  
→ my-first-github-repository git:(temp) git branch -vv

[19] + 4649 suspended git branch -vv  
→ my-first-github-repository git:(temp) git remote update origin --prune  
Fetching origin  
From https://github.com/hema8Codes/my-first-github-repository  
→ [deleted] (none) -> origin/temp  
→ my-first-github-repository git:(temp) git branch -vv

[20] + 4690 suspended git branch -vv

feature-1 b3d043d [origin/feature-1]	Another local change
feature-2 e07bf88 [origin/feature-2]	Another change in feature-2 branch
main 3a3e202 [origin/main]	Create hello-github.txt
* temp e07bf88 [origin/temp]	Another change in feature-2 branch

(END) → local temp branch still tracks remote temp branch

feature-1 b3d043d [origin/feature-1]	Another local change
feature-2 e07bf88 [origin/feature-2]	Another change in feature-2 branch
main 3a3e202 [origin/main]	Create hello-github.txt
* temp e07bf88 [origin/temp: gone]	Another change in feature-2 branch

(END) how local git repository → we see gone here in this knows that remote temp branch is gone. list

If we go on GitHub Desktop, you will see now we are able to publish our branch

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

**Publish your branch**  
The current branch (temp) hasn't been published to the remote yet. By publishing it to GitHub you can share it, open a pull request, and collaborate with others.  
Always available in the toolbar or ⌘ P

**Open the repository in your external editor**  
Select your editor in Preferences  
Repository menu or ⌘ ⌘ A

**View the files of your repository in Finder**  
Repository menu or ⌘ ⌘ F

**Open the repository page on GitHub in your browser**  
Repository menu or ⌘ ⌘ G

Summary (required)

Description

R+

Commit to temp

→ Deleting local temp branch

```
→ my-first-github-repository git:(temp) git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
→ my-first-github-repository git:(main) git branch -d temp
error: The branch 'temp' is not fully merged. ↴
If you are sure you want to delete it, run 'git branch -D temp'.
→ my-first-github-repository git:(main) git branch -D temp
Deleted branch temp (was e07bf88).           force delete
→ my-first-github-repository git:(main) git branch -vv
```

[21] + 4859 suspended git branch -vv

```
→ my-first-github-repository git:(main) █
```

Now "temp" branch is absent both in the local and remote repository.

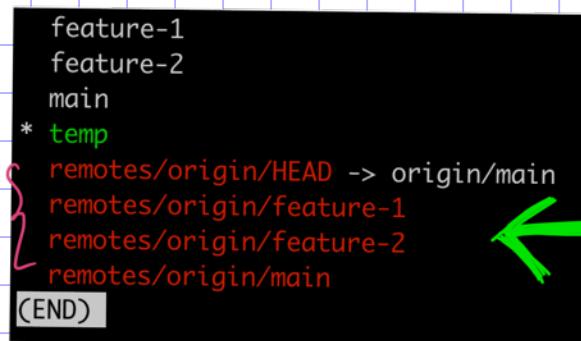
```
feature-1 b3d043d [origin/feature-1] Another local change
feature-2 e07bf88 [origin/feature-2] Another change in feature-2 branch
* main      3a3e202 [origin/main] Create hello-github.txt
(END)
```

# Remove remote branch using local terminal

→ Deleting remote branches from the local terminal.

Plan:

1. Create new local branch "temp"
2. Push this local branch to remote repository using "git push -u Origin temp"
3. Delete remote branch using "git push origin -d temp"



```
→ my-first-github-repository git:(main) git checkout -b temp
Switched to a new branch 'temp' new branch was created and we are automatically switched to temp branch.
→ my-first-github-repository git:(temp) git branch listing all branches
[1] + 926 suspended git branch
→ my-first-github-repository git:(temp) git branch -a listing all branches

[2] + 957 suspended git branch -a
→ my-first-github-repository git:(temp) git push -u origin temp
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 Creating corresponding remote temp branch
remote:
remote: Create a pull request for 'temp' on GitHub by visiting: in remote repository.
remote:     https://github.com/hema8Codes/my-first-github-repository/pull/new/temp
remote:
To https://github.com/hema8Codes/my-first-github-repository.git
 * [new branch]      temp -> temp
Branch 'temp' set up to track remote branch 'temp' from 'origin'.
```

```

feature-1 b3d043d [origin/feature-1] Another local change
feature-2 e07bf88 [origin/feature-2] Another change in feature-2 branch
main      3a3e202 [origin/main] Create hello-github.txt
* temp     3a3e202 [origin/temp] Create hello-github.txt
(END) there is corresponding remote temp branch for local temp branch.

→ my-first-github-repository git:(temp) git branch -vv
[3] + 1017 suspended git branch -vv

```

hema8Codes / my-first-github-repository

Unwatch 1 Star 0 Fork

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

All branches

All branches

Branch	Last Commit	Status	Actions
main	Updated 8 days ago by hema8Codes	(Default)	<i>Pencil</i>
feature-2	Updated 13 hours ago by hema8Codes	0   11	<i>New pull request</i> <i>Pencil</i> <i>Delete</i>
feature-1	Updated 17 hours ago by hema8Codes	0   9	<i>New pull request</i> <i>Pencil</i> <i>Delete</i>
temp	Updated 8 days ago by hema8Codes	0   0	<i>New pull request</i> <i>Pencil</i> <i>Delete</i>

verify in GitHub

We see local temp branch but remote temp branch was removed.

```

feature-1
feature-2
main
* temp ✓
remotes/origin/HEAD -> origin/main
remotes/origin/feature-1
remotes/origin/feature-2
remotes/origin/main
(END)

```

```

→ my-first-github-repository git:(temp) git push origin -d temp
To https://github.com/hema8Codes/my-first-github-repository.git
 - [deleted] temp
→ my-first-github-repository git:(temp) git branch -a
[4] + 1070 suspended git branch -a

```

→ We will not see temp branch on GitHub.

A screenshot of a GitHub repository page for 'hema8Codes / my-first-github-repository'. The page shows the 'All branches' tab selected. There are three branches listed: 'main' (updated 8 days ago), 'feature-2' (updated 13 hours ago), and 'feature-1' (updated 17 hours ago). A red curly brace on the left side groups the 'main' branch under the heading 'All branches'. On the right side, there are buttons for 'New pull request' and edit/delete icons for each branch.

```
→ my-first-github-repository git:(temp) git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
→ my-first-github-repository git:(main) git branch -D temp
Deleted branch temp (was 3a3e202).
→ my-first-github-repository git:(main) git branch -a
```

```
feature-1
feature-2
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/feature-1
  remotes/origin/feature-2
  remotes/origin/main
(END)
```

## Git show-ref

→ Quick way to examine branch references using "git show-ref"

local refs  
remote refs  
branch  
remote ref

```
→ my-first-github-repository git:(main) git show-ref → used for checking
b3d043d54eb9decccd6774e042f317849cd196de7 refs/heads/feature-1 remote refs
e07bf887b287eb9ce880b1df20b3b1b80bb136a1 refs/heads/feature-2 and local refs
3a3e202a8ffc7361245fd8416e496d423e697a37 refs/heads/main
3a3e202a8ffc7361245fd8416e496d423e697a37 refs/remotes/origin/HEAD
b3d043d54eb9decccd6774e042f317849cd196de7 refs/remotes/origin/feature-1
e07bf887b287eb9ce880b1df20b3b1b80bb136a1 refs/remotes/origin/feature-2
3a3e202a8ffc7361245fd8416e496d423e697a37 refs/remotes/origin/main
→ my-first-github-repository git:(main) ls .git/refs/heads → listing files for each local branch
feature-1 feature-2 main
→ my-first-github-repository git:(main) ls .git/refs/remotes/origin HEAD      feature-1 feature-2 → remote references
```

→ with `git show-ref` command you can also compare references for a specific branch.

```
→ my-first-github-repository git:(main) git show-ref main  
3a3e202a8ffc7361245fd8416e496d423e697a37 refs/heads/main → reference for local main branch  
3a3e202a8ffc7361245fd8416e496d423e697a37 refs/remotes/origin/main → reference for remote master branch  
→ my-first-github-repository git:(main) git show-ref feature-1  
b3d043d54eb9decccd6774e042f317849cd196de7 refs/heads/feature-1  
b3d043d54eb9decccd6774e042f317849cd196de7 refs/remotes/origin/feature-1  
→ my-first-github-repository git:(main) git show-ref feature-2  
e07bf887b287eb9ce880b1df20b3b1b80bb136a1 refs/heads/feature-2  
e07bf887b287eb9ce880b1df20b3b1b80bb136a1 refs/remotes/origin/feature-2
```

and for example if we try to change something in for eg → feature-2 branch locally.

```
→ my-first-github-repository git:(main) git checkout feature-2  
Switched to branch 'feature-2'  
Your branch is up to date with 'origin/feature-2'.  
→ my-first-github-repository git:(feature-2) git show-ref feature-2  
b484a3d27ca1c507f35e1f12606d05204b8ca761 refs/heads/feature-2  
e07bf887b287eb9ce880b1df20b3b1b80bb136a1 refs/remotes/origin/feature-2  
→ my-first-github-repository git:(feature-2) git log
```

```
SOURCE CONTR...  ✓ ...  
Changes in feature-2 branch  
Changes  
another-file.txt... ↗ + M  
[Preview] CHANGELOG_7.8.md  another-file.txt M  README.md  
feature > another-file.txt  
1 This line was replace in feature-2 branch  
2  
3 This file was modified locally  
4  
5 This line was added in feature-2 branch  
6  
7 Another modification in the feature-2 branch  
8  
9 Edit in the feature-2 branch
```

editing another-file.txt in feature-2 branch.

we see here two different SHA1 hashes

→ git log

```
commit b484a3d27ca1c507f35e1f12606d05204b8ca761 (HEAD -> feature-2)
Author: hema8Codes <developer8hema@gmail.com>
Date:   Wed Sep 1 12:47:57 2021 +0530
```

→ pointer for local feature-2 branch

Changes in feature-2 branch

```
commit e07bf887b287eb9ce880b1df20b3b1b80bb136a1 (origin/feature-2)
Author: hema8Codes <developer8hema@gmail.com>
Date:   Tue Aug 31 22:18:18 2021 +0530
```

→ pointer for remote feature-2 branch.

Another change in feature-2 branch

```
commit ae432c9c6e2b3ed57793839e70aaa9313717ab29
Author: hema8Codes <developer8hema@gmail.com>
Date:   Tue Aug 31 22:05:16 2021 +0530
```

Another file was modified in feature-2 branch

```
commit b3d043d54eb9decccd6774e042f317849cd196de7 (origin/feature-1, feature-1)
Author: hema8Codes <developer8hema@gmail.com>
Date:   Tue Aug 31 18:37:36 2021 +0530
```

Another local change

```
commit a44f055f50c0d16737673049069fc83a70829363
Merge: 798e243 9b2f4cf
Author: hemakshi1234 <mahe173fas@gmail.com>
Date:   Sun Aug 29 16:56:49 2021 +0530
```

Merge branch 'feature-1' of https://github.com/hema8Codes/my-first-github-repository into feature-1

```
commit 798e2432c8718d651169ec1db29414ed50f0893b
Author: hemakshi1234 <mahe173fas@gmail.com>
:|
```

→ my-first-github-repository git:(feature-2) git push -v pushing changes to remote

Pushing to https://github.com/hema8Codes/my-first-github-repository.git

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 506 bytes | 506.00 KiB/s, done.

Total 4 (delta 1), reused 0 (delta 0), pack-reused 0

POST git-receive-pack (684 bytes)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To https://github.com/hema8Codes/my-first-github-repository.git

e07bf88..b484a3d feature-2 -> feature-2

updating local tracking ref 'refs/remotes/origin/feature-2'

→ my-first-github-repository git:(feature-2) git show-ref feature-2

b484a3d27ca1c507f35e1f12606d05204b8ca761 refs/heads/feature-2

b484a3d27ca1c507f35e1f12606d05204b8ca761 refs/remotes/origin/feature-2

→ my-first-github-repository git:(feature-2) Remote feature-2 branch

Same SHA1

is in sync with local feature-2 branch.