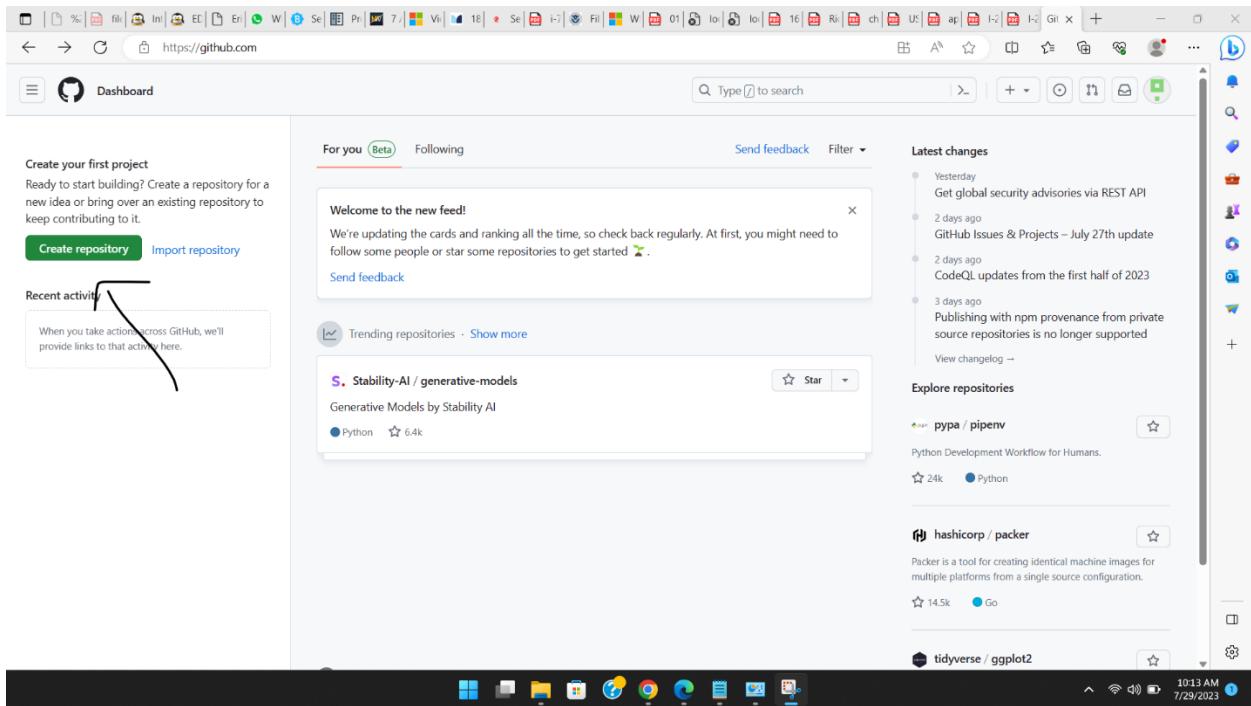


## Git Hub Setup for Windows

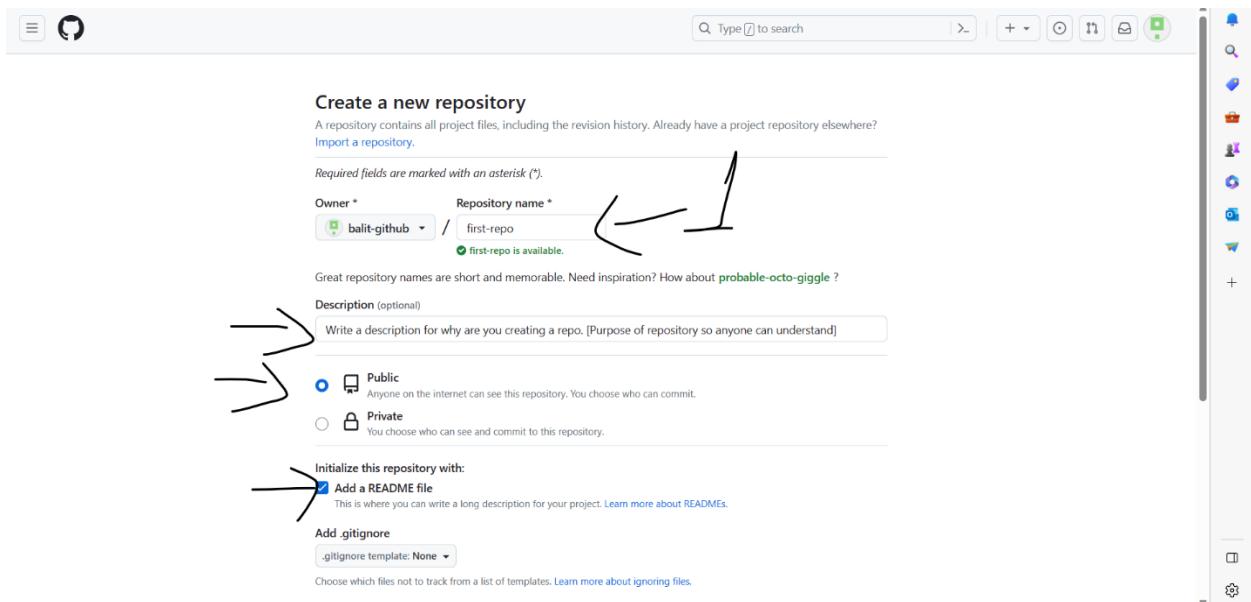
Step 1: Create a github account (<https://github.com/>)

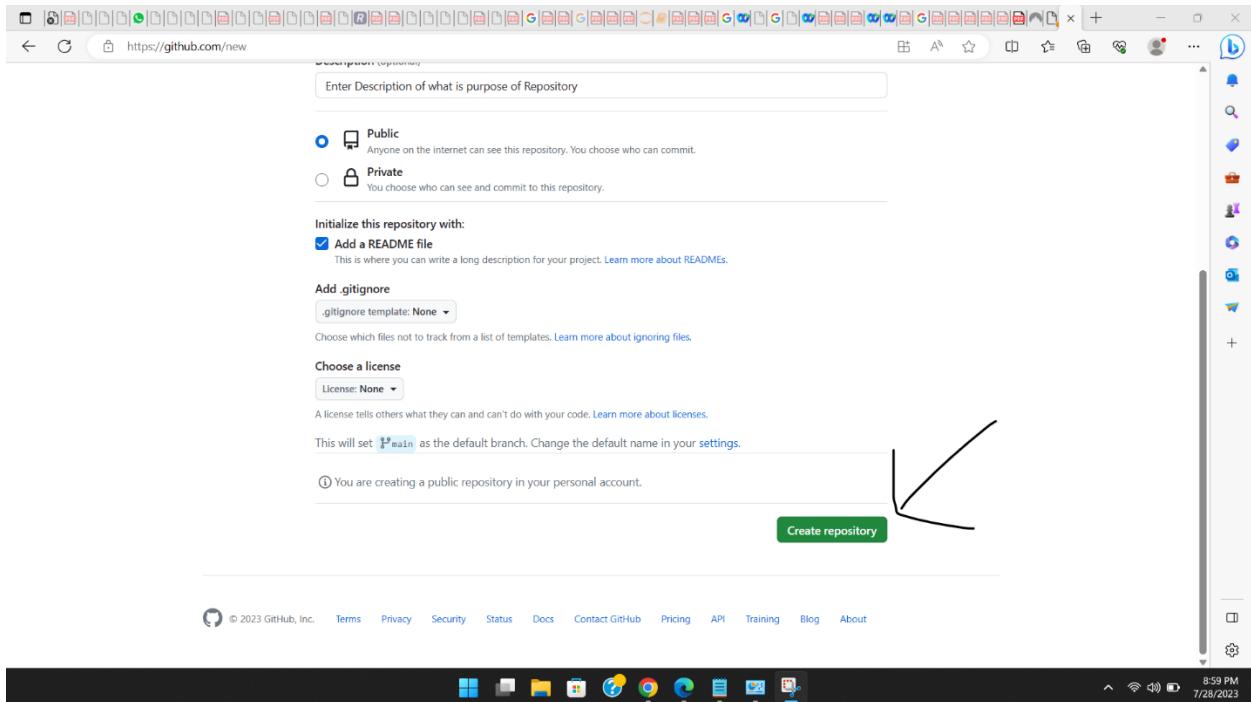
Step 2: Once you created Git hub account it will look like this below screen shot (SS).

- Now let's Create First Repository.

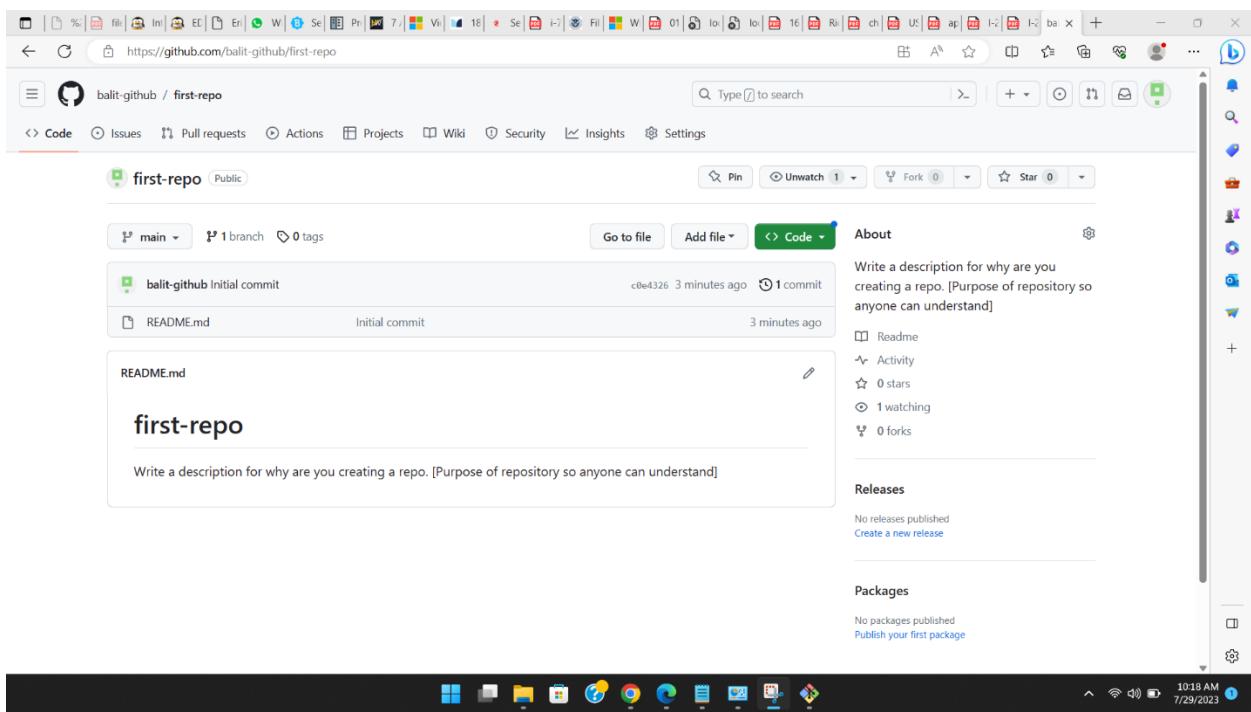


Step 3: Enter the repository name, Write repository Description, make sure it is public or private whatever you prefered and click on Readme file. Then scroll down and click on create repository.

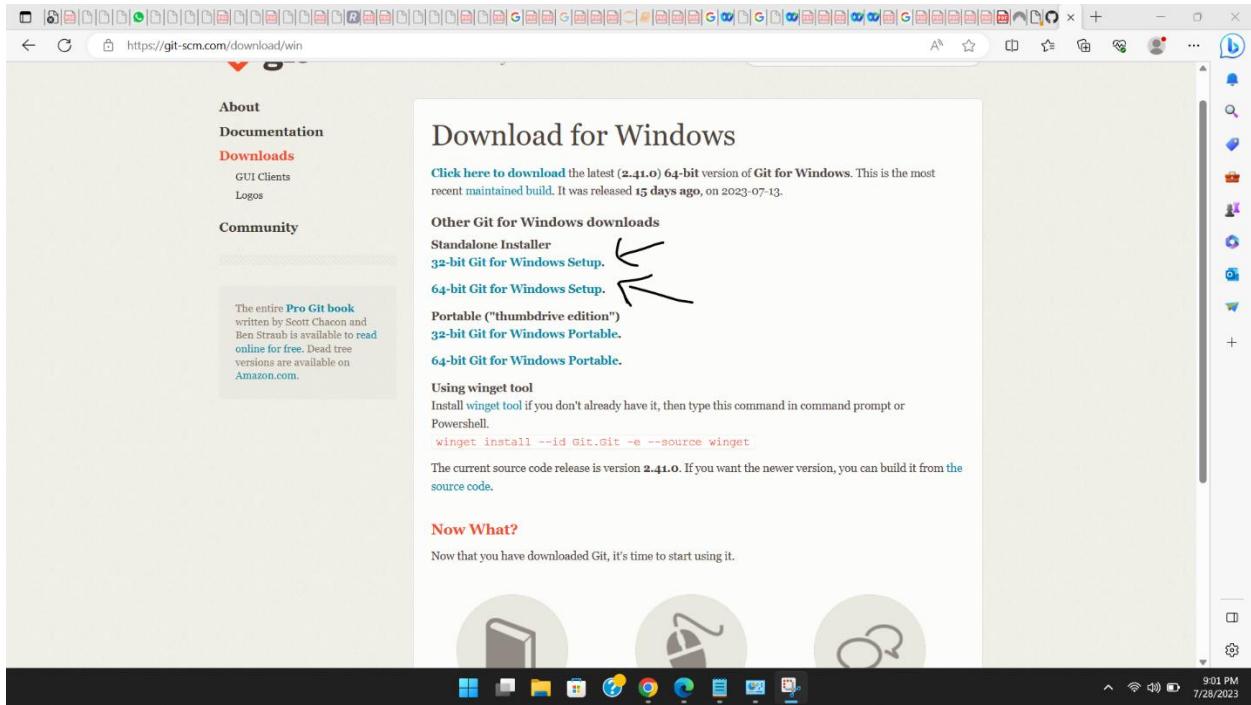




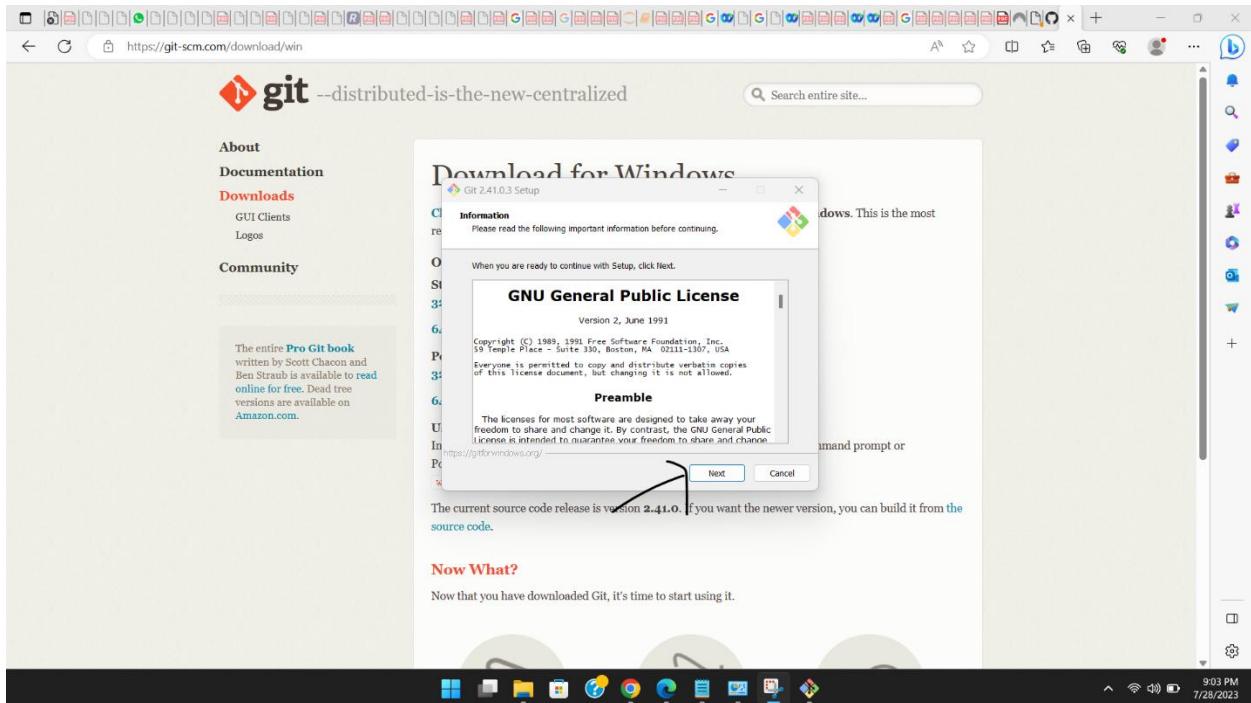
Step 4: After you sucessfully created repository your webpage will look like this.



Step 5: Now let's install git in desktop (<https://git-scm.com/download/win>)



- You can choose based on your OS architecture. (here mine is 64 bit so I downloaded 64 bit).



**Select Components**

Which components should be installed?

Additional icons

On the Desktop

Windows Explorer Integration

Open Git Bash Here

Git GUI Here

Git LFS Large File Support

Associate .git\* configuration file with the default text editor

Associate .sh files to be run with Bash

Check daily for Git for Windows updates

(NEW!) Add a Git Bash Profile to Windows Terminal

Current selection requires at least 319.4 MB of disk space.

The current source code release is version 2.41.0. If you want the newer version, you can build it from the source code.

**Now What?**

Now that you have downloaded Git, it's time to start using it.

9:04 PM  
7/28/2023

**Select Start Menu Folder**

Where should Setup place the program's shortcuts?

Git

Don't create a Start Menu folder

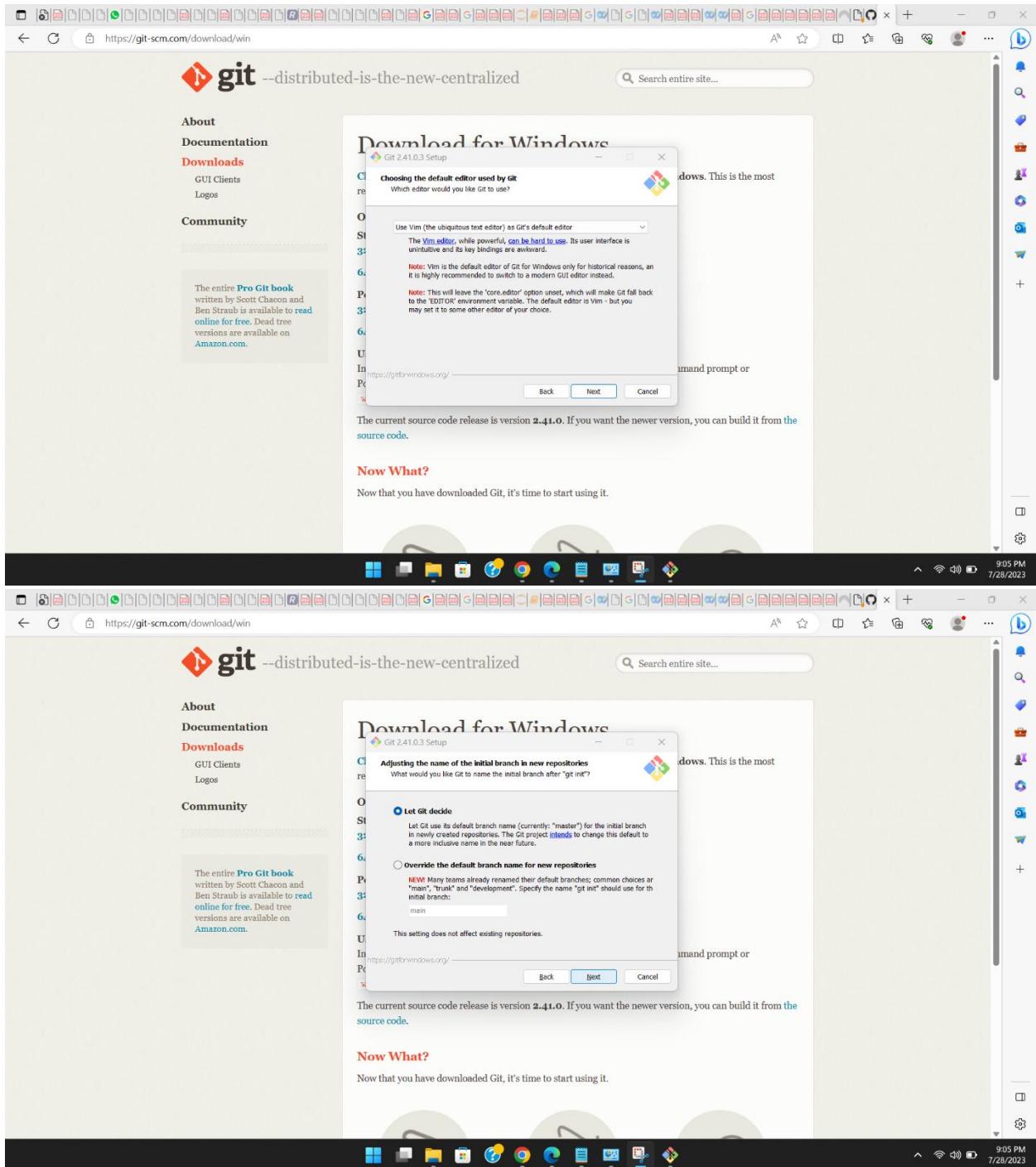
Current selection requires at least 319.4 MB of disk space.

The current source code release is version 2.41.0. If you want the newer version, you can build it from the source code.

**Now What?**

Now that you have downloaded Git, it's time to start using it.

9:04 PM  
7/28/2023



The image shows two screenshots of a web browser displaying the Git download page for Windows. The top screenshot shows the initial setup window for adjusting the PATH environment, while the bottom screenshot shows the configuration for choosing the SSH executable.

**Top Screenshot: Adjusting your PATH environment**

The window title is "Download for Windows" and "Git 2.41.0.3 Setup". It asks, "How would you like to use Git from the command line?". Three options are listed:

- Use Git from Git Bash only
- Use Git and also from 3rd-party software (Recommended)
- Use Git and optional Unix tools from the Command Prompt

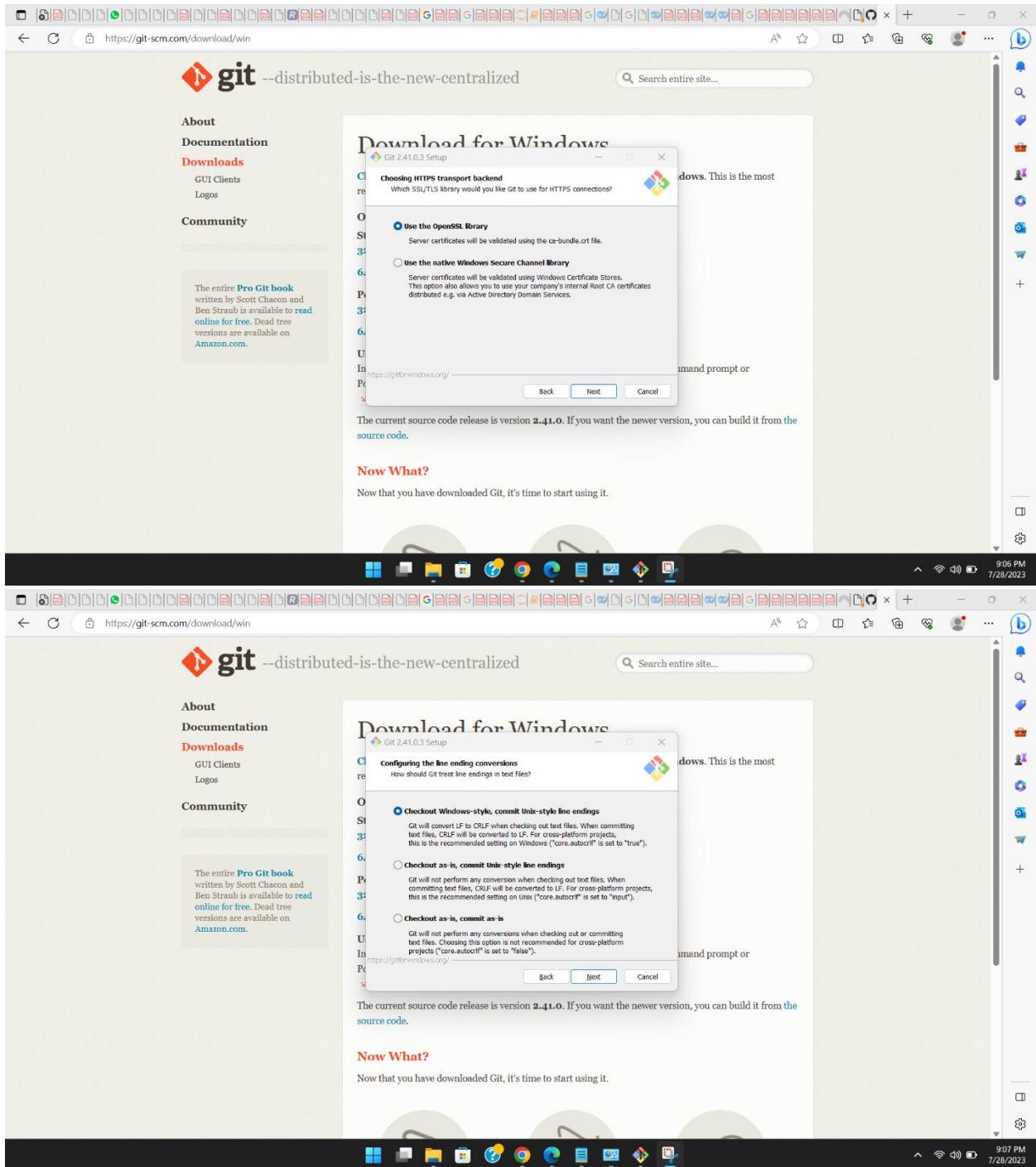
Details for the second option state: "Both Git and the optional Unix tools will be added to your PATH." and "Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications." A note at the bottom says, "The current source code release is version 2.41.0. If you want the newer version, you can build it from the source code."

**Bottom Screenshot: Choosing the SSH executable**

The window title is "Download for Windows" and "Git 2.41.0.3 Setup". It asks, "Which Secure Shell client program would you like Git to use?". Two options are listed:

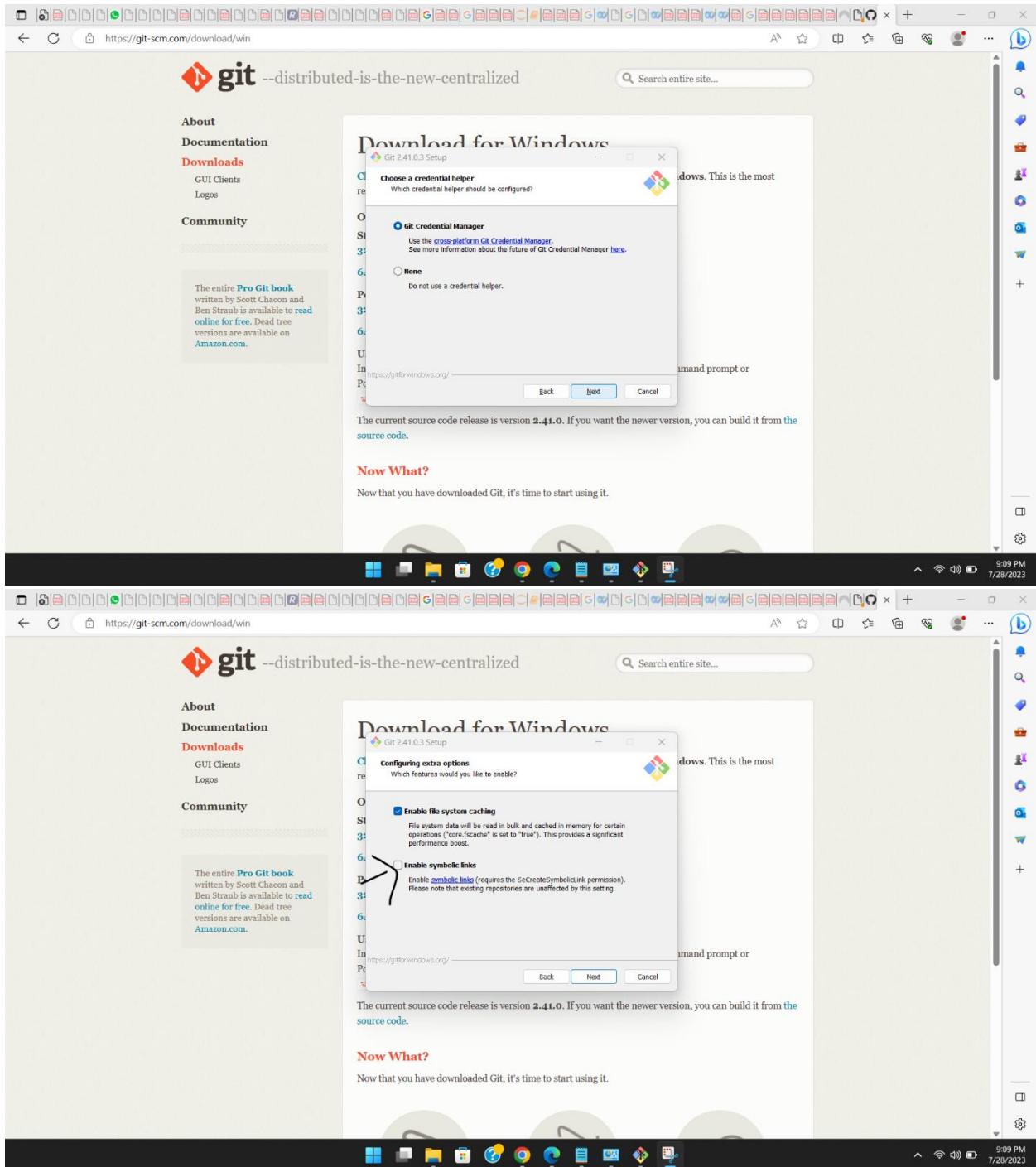
- Use bundled OpenSSH
- Use external OpenSSH

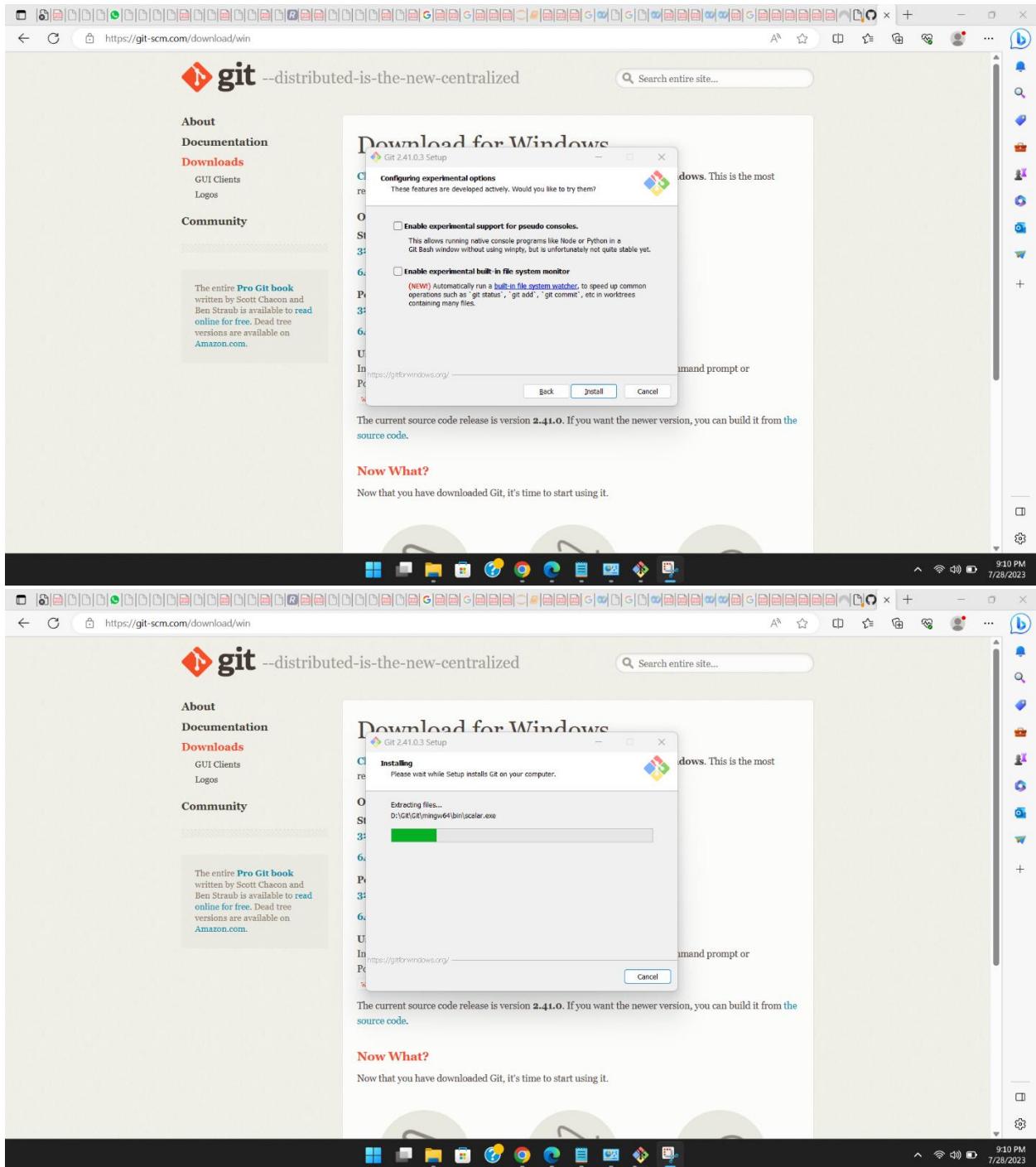
Details for the first option state: "This uses ssh.exe that comes with Git." Details for the second option state: "This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH." A note at the bottom says, "The current source code release is version 2.41.0. If you want the newer version, you can build it from the source code."

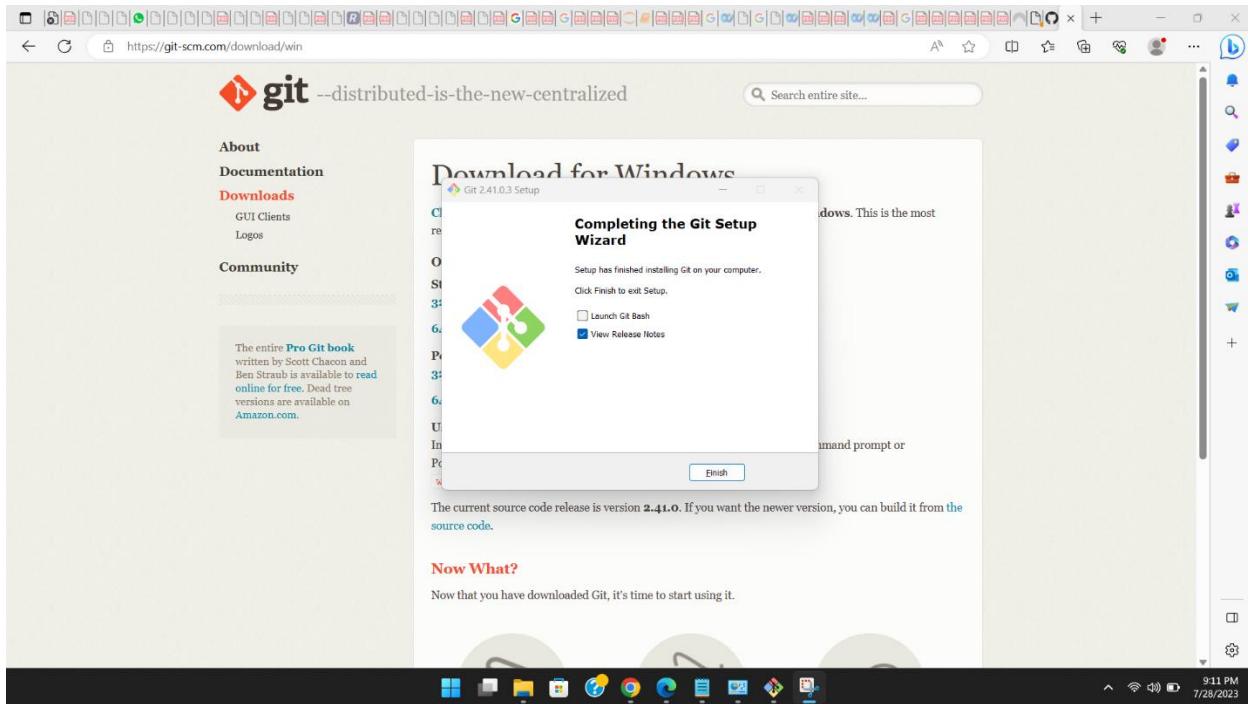


Screenshot of a web browser showing the Git download page for Windows. The URL is https://git-scm.com/download/win. A modal window titled "Download for Windows" is open, asking "Configuring the terminal emulator to use with Git Bash". It shows two options: "Use MinTTY (the default terminal of MSYS2)" (selected) and "Use Windows' default console window". Below the modal, a note says "The current source code release is version 2.41.0. If you want the newer version, you can build it from the source code." At the bottom, a "Now What?" section says "Now that you have downloaded Git, it's time to start using it." The browser toolbar at the top includes icons for back, forward, search, and refresh.

The second screenshot shows the same process, but the "Choose the default behavior of 'git pull'" step is now visible. It offers three options: "Default (fast-forward or merge)" (selected), "Rebase", and "Only ever fast-forward". The rest of the interface and system status bar at the bottom are identical to the first screenshot.



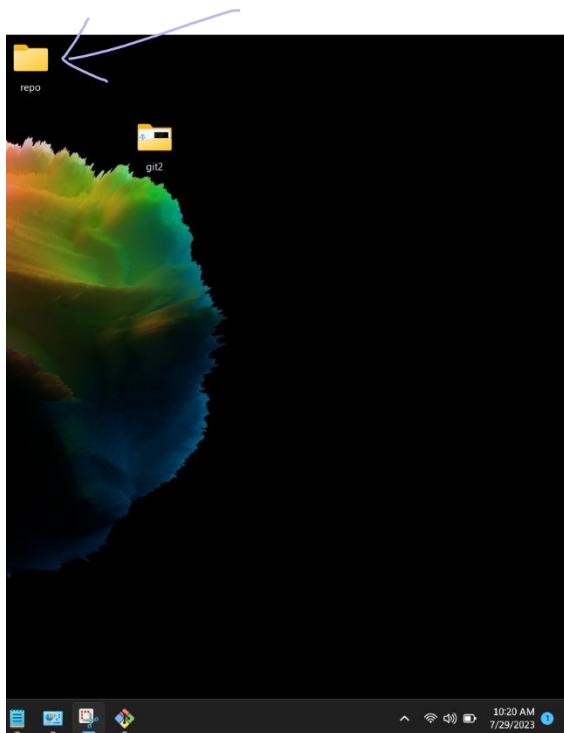




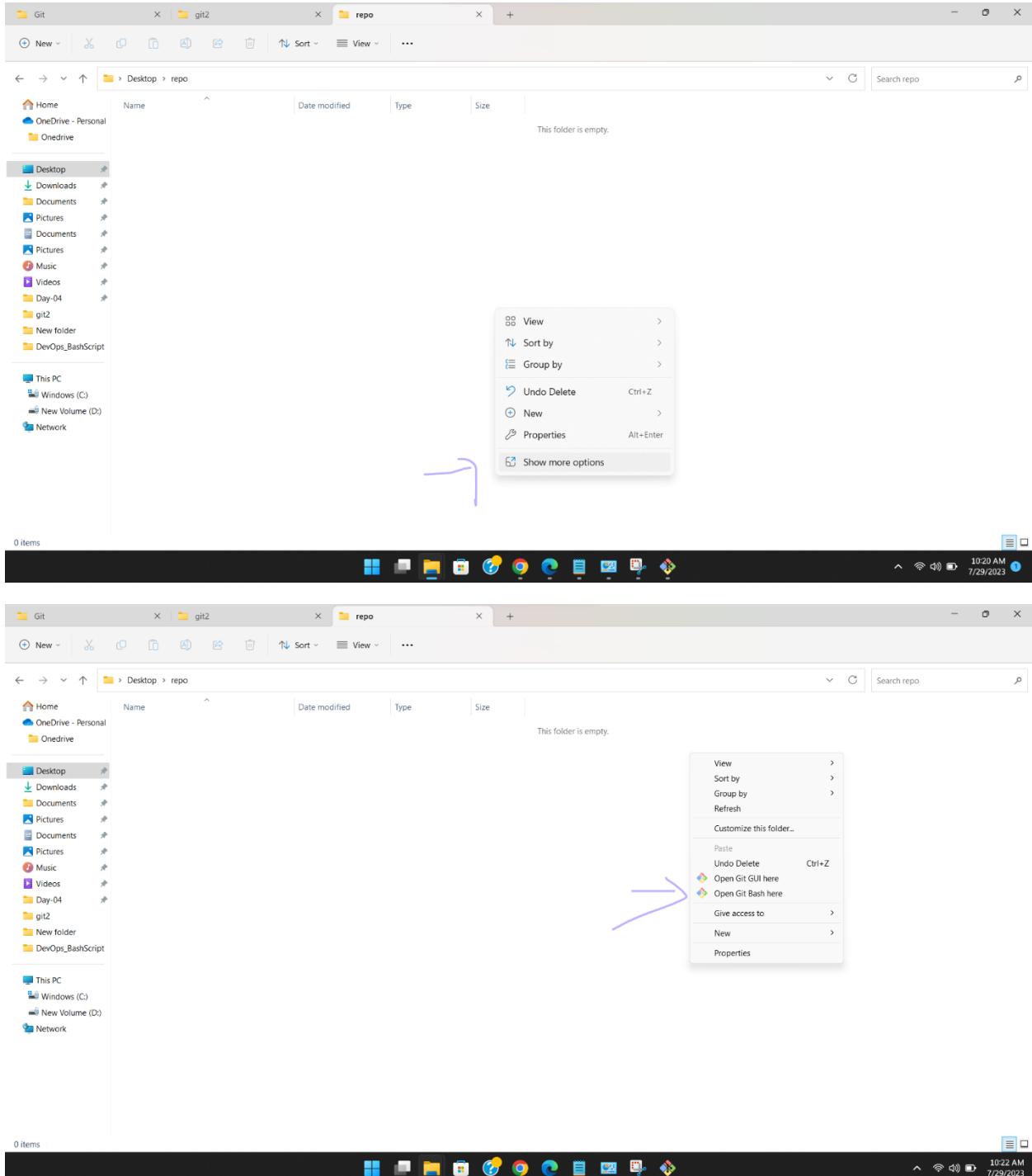
- You can install git configuration based on your requirements by adding additional functionality.

Step 6: Create a new folder for cloning repository (Copying repository from github to your local machine, downloading repository from github to your local machine) all in one place.

- Here I created “repo” folder on my desktop.



Step 7: Open folder which you just created and now we will open git bash. So right click on screen and click on git bash.



- If you can't see this option it means you haven't installed **Git**. So install first or open git bash from where you installed it and go to a folder which you created.

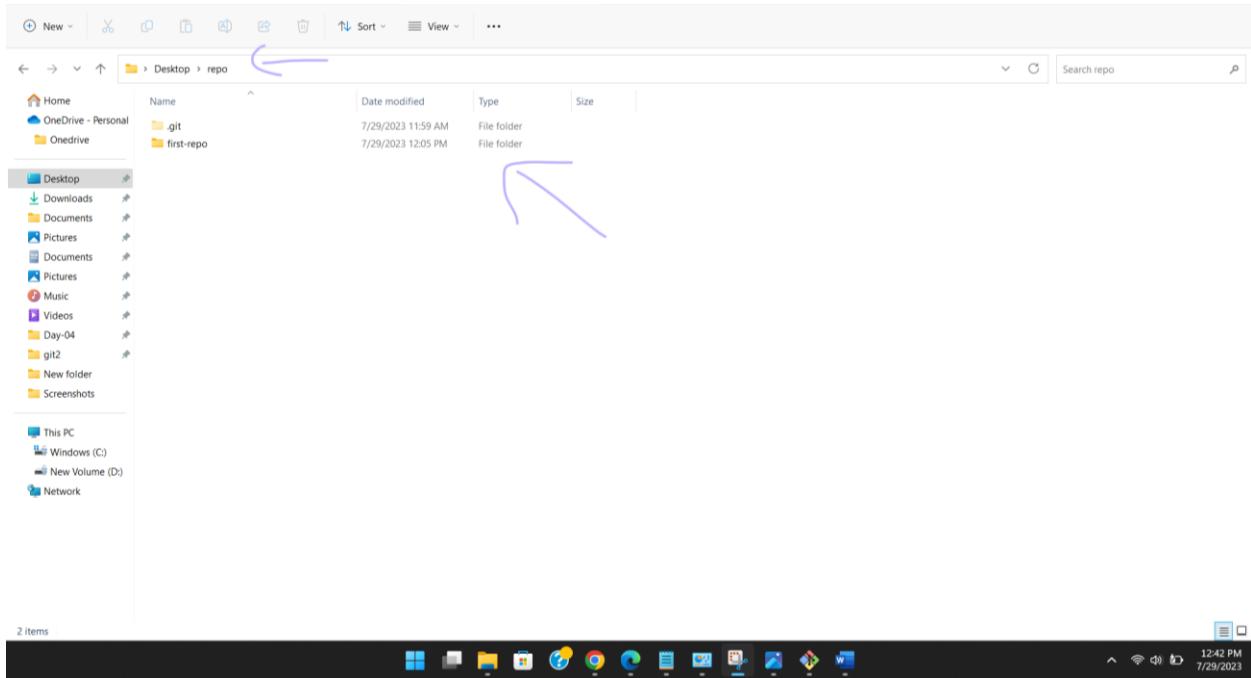
Step 8: let's clone (download, copy) github repository.

- Type command
  - o Git clone https://github.com/balit-github/first-repo.git

The screenshot shows a Windows desktop environment. In the center, a web browser window displays the GitHub repository page for 'first-repo'. The repository is public and has one branch ('main') and no tags. The README file contains the text: 'Write a description for why you are creating a repo. [Purpose of repository so anyone can understand]'. Below the README, there are sections for Activity, Stars (0), Watchers (1), Forks (0), and Releases (No releases published). On the right side of the GitHub page, there are sections for About, Packages (No packages published), and Settings. In the bottom left corner of the screen, a terminal window is open, showing the command 'git clone https://github.com/balit-github/first-repo.git' being typed. The terminal output shows the cloning process: 'Cloning into 'first-repo''. The desktop taskbar at the bottom shows various pinned icons, and the system tray indicates the date and time as 10:23 AM on 7/29/2023.

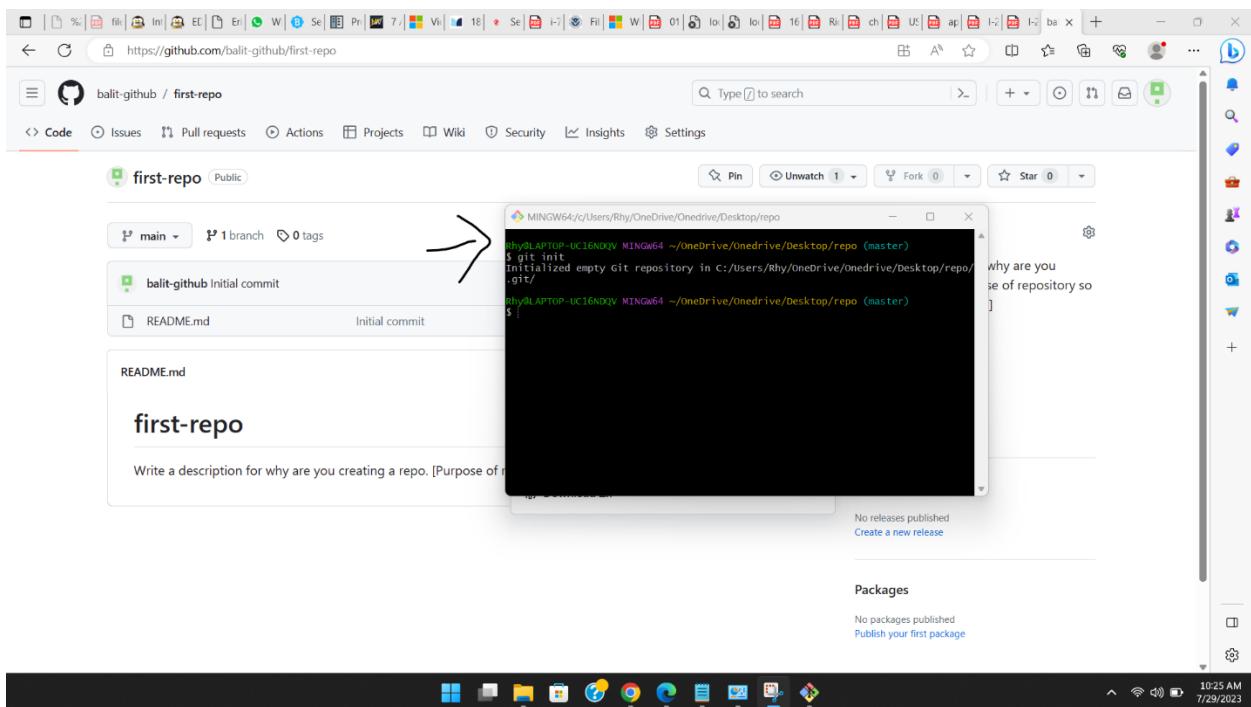
Step 9: type 'ls' command which will shows a all files inside a folder (directory). Or you can check by opening file manager in windows.

This screenshot is similar to the previous one, showing a Windows desktop with a GitHub repository page for 'first-repo'. The terminal window now shows the output of the 'ls' command, which lists the files in the directory: 'README.md' and 'first-repo/'. The GitHub page itself shows the repository details, including the README content and various repository metrics. The desktop taskbar and system tray are visible at the bottom.



## Step 10: Type command (this command used for make a repository)

- Git init



Step 11: Now we will go to cloned directory (in my case cloned directory name is ‘first-repo’). So I will type command.

- cd first-repo

```
Rhy@LAPTOP-UC16NQXV MINGW64 ~/OneDrive/Onedrive/Desktop/repo (master)
$ ls
First-repo/
Rhy@LAPTOP-UC16NQXV MINGW64 ~/OneDrive/Onedrive/Desktop/repo (master)
$ cd First-repo
Rhy@LAPTOP-UC16NQXV MINGW64 ~/OneDrive/Onedrive/Desktop/repo/First-repo (main)
$ ls
README.md
```

- ls

Step 12: now let’s make a file with some content.

- echo “this is my sample file” > sample.txt
- ls

```
Rhy@LAPTOP-UC16NQXV MINGW64 ~/OneDrive/Onedrive/Desktop/repo (master)
$ ls
First-repo/
Rhy@LAPTOP-UC16NQXV MINGW64 ~/OneDrive/Onedrive/Desktop/repo (master)
$ cd First-repo
Rhy@LAPTOP-UC16NQXV MINGW64 ~/OneDrive/Onedrive/Desktop/repo/First-repo (main)
$ ls
README.md
Rhy@LAPTOP-UC16NQXV MINGW64 ~/OneDrive/Onedrive/Desktop/repo/First-repo (main)
$ echo "this is my sample file" > sample.txt
Rhy@LAPTOP-UC16NQXV MINGW64 ~/OneDrive/Onedrive/Desktop/repo/First-repo (main)
$ ls
README.md sample.txt
Rhy@LAPTOP-UC16NQXV MINGW64 ~/OneDrive/Onedrive/Desktop/repo/First-repo (main)
```

Step 13: Now we want to upload our file to github so how can we do it ?

- For that we need to push our file to github but before we push to github how git (git terminal) will know to which repository we are pushing file ?? so we have to configure some setting such as username, email ID and password.
- So follow below commands
  - o git config --global user.name "balit-github"
  - o git config --global user.email "personalapikeys@gmail.com"

```

MINGW64:/c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo (main)
$ ls
README.md sample.txt
MINGW64:/c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo (main)
$ git add .
warning: in the working copy of 'sample.txt', LF will be replaced by CRLF the next time git touches it
MINGW64:/c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo (main)
$ git commit -m 'first time pushing to github'
[balit-github] first time pushing to github
1 file changed, insertion+deletion, create mode 100644 sample.txt
MINGW64:/c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo (main)
$ git config --global user.name "balit-github"
MINGW64:/c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo (main)
$ git config --global user.email "personalapikeys@gmail.com"
MINGW64:/c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo (main)
$ 

```

Step 14: still we need to provide a repository path and we haven't yet authenticated github (login to github)

- let's add remote repository url (remember we copied url at clone time, yes that url we need again)
  - o git remote add origin https://github.com/balit-github/first-repo.git

```

MINGW64:/c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo (main)
$ git remote add origin https://github.com/balit-github/first-repo.git

```

- To check all setting that we just made so type command.
    - o git config --list --show-origin

The screenshot shows a GitHub repository page for 'balit-github / first-repo'. The repository has one branch ('main') and one tag ('balit-github Initial commit'). A file named 'README.md' is present with the content 'Initial commit'. The terminal window in the background displays the output of the command 'git config --list --show-origin', listing various global configuration options like 'http.sslcainfo', 'core.autocrlf', and 'remote.origin.fetch'. Three red arrows point from the right margin of the GitHub interface to specific lines in the terminal output: 'filter.lfs.clean=git-lfs clean --', 'filter.lfs.process=git-lfs filter-process', and 'remote.origin.fetch=refs/heads/\*:refs/remotes/origin/\*'.

Step 15: Now let's add a file (upload to github or push to github)

- type command
    - o git add . [here this command will add all the files which are created or edited in first-repo directory]
    - o git commit -m 'first time pushing to github' [-m is argument which will pass the message, like what changed have been made or why are we updating existing repository on github. So describe about it in ‘.....’].

The screenshot shows a GitHub repository named 'first-repo'. The repository has one branch ('main') and no tags. It contains two commits:

- balit-github Initial commit**: README.md, Initial commit
- balit-github Initial commit**: README.md, Initial commit

A terminal window is open in the repository view, showing the following commit history:

```
MINGW64 /c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo
$ git status
# On branch main
# Initial commit
# Untracked files:
#   (use "git add <file>" to include in what will be committed)
#
#       README.md
nothing added to commit but untracked files present (use "git add" to track)

MINGW64 /c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo
$ echo "this is my sample file" > sample.txt
MINGW64 /c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo
$ ls
README.md  sample.txt

MINGW64 /c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo
$ git add .
warning: LF will be replaced by CRLF in README.md.
warning: LF will be replaced by CRLF in sample.txt
nothing added to commit but untracked files present (use "git add" to track)

MINGW64 /c/Users/Rhy/OneDrive/Onedrive/Desktop/repo/first-repo
$ git commit -m 'First time pushing to github'
[main 57be881] First time pushing to github
 1 file changed, 1 insertion(+)
 create mode 100644 sample.txt
```

The terminal output includes several warning messages about line endings being converted from LF to CRLF.

Step 16: - Finally we reach to command which is responsible for pushing (uploading or updating github existing repository)

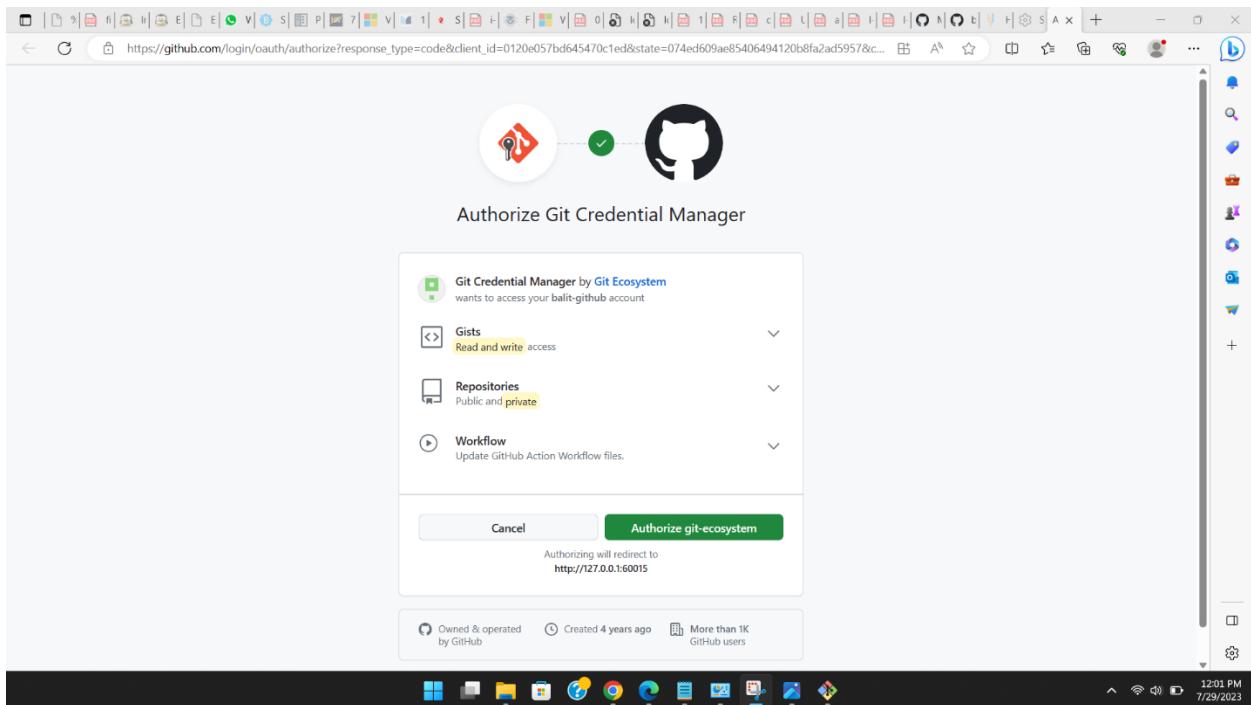
- o git push -u origin main

The screenshot shows a Windows desktop environment. In the center, a web browser window displays the GitHub repository 'balit-github / first-repo'. A modal dialog box titled 'GitHub Sign in' is open, offering 'Sign in with your browser' and 'Sign in with a code' options. Below the modal, a message says 'Don't have an account? Sign up'. To the right of the modal, a terminal window is open, showing the following command sequence:

```
$ echo "this is my sample file" > sample.txt
$ git add .
warning: In the working copy of 'sample.txt', LF will be replaced by CRLF the next time you touch it
$ git commit -m 'first time pushing to github'
[main def9a3] first time pushing to github
 1 file changed, 1 insertion(+)
 create mode 100644 sample.txt
$ git push -u origin main
```

At the bottom of the GitHub page, there are sections for 'Releases' (No releases published) and 'Packages' (No packages published). The taskbar at the bottom shows various pinned icons.

- click on sign in with your browser.



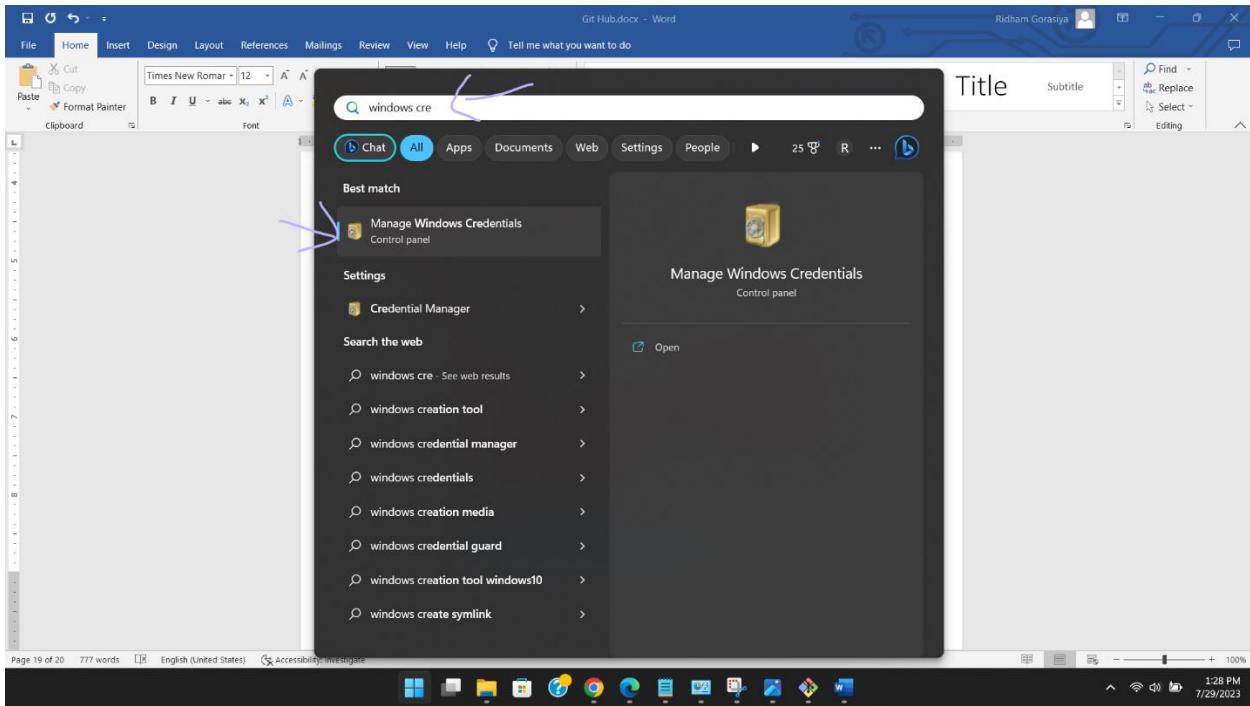
- login to your github userid or email and passwd.

The screenshot shows a GitHub repository named 'first-repo' owned by 'balit-github'. The repository has 1 branch and 0 tags. It contains two files: 'README.md' (Initial commit) and 'sample.txt' (first time pushing to github). A terminal window is open in the browser showing the command \$ git push -u origin main being run. The output indicates the first time pushing to GitHub, listing objects and compression details, and confirming the push to https://github.com/balit-github/first-repo/. The GitHub interface also shows a 'Releases' section with a 'Create a new release' button.

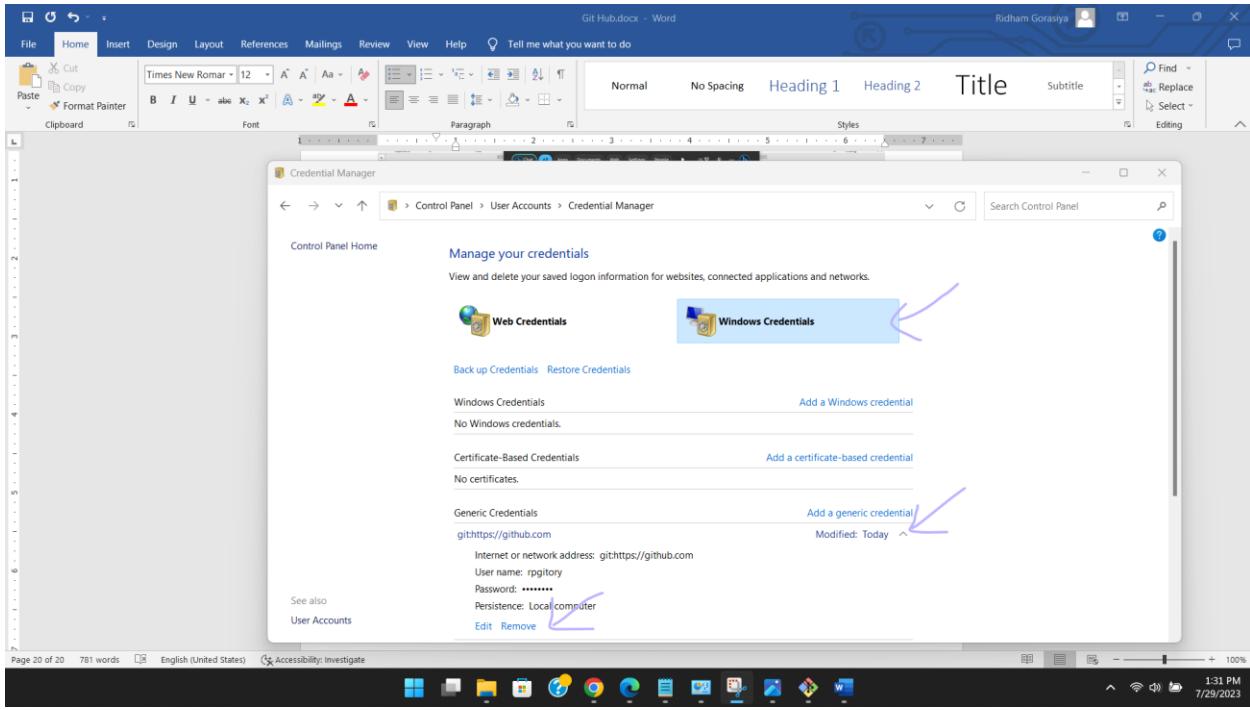
### Note: THIS IS NOT COMPULSORY STEP ONLY FOR WHO HAVE ERROR.

if anyone have error like “`$ git push -u origin main` remote: Permission to balit-github/first-repo.git denied to rpgitory. fatal: unable to access '<https://github.com/balit-github/first-repo/>': The requested URL returned error: 403”

- so there are several cases
- Case 1: If you ever configured git with your local machine and now you created new github account which is different than earlier so in that case you will have error like this.
  - o Let's understand the error.
  - o `$ git push -u origin main` (this was a command which we passed last time)
  - o We got an error like this “Permission to balit-github/first-repo.git denied to rpgitory. fatal: unable to access '<https://github.com/balit-github/first-repo/>': The requested URL returned error: 403”
  - o So here ‘Permission to balit-github/first-repo.git denied to rpgitory’ this line error shows like earlier I had been configured with github account and uswername should be ‘rpgitory’ [<https://github.com/rpgitory>] but right now my username is “**balit-github**” [<https://github.com/balit-github/first-repo.git>] as we seen in above SS. so now we need to remove the previous git configuration from our computer.
  - o This solution only for windows computer.
  - o Open start menu and write.



- If you see a user name which shows on error. (in my case rpgitory) so remove it.

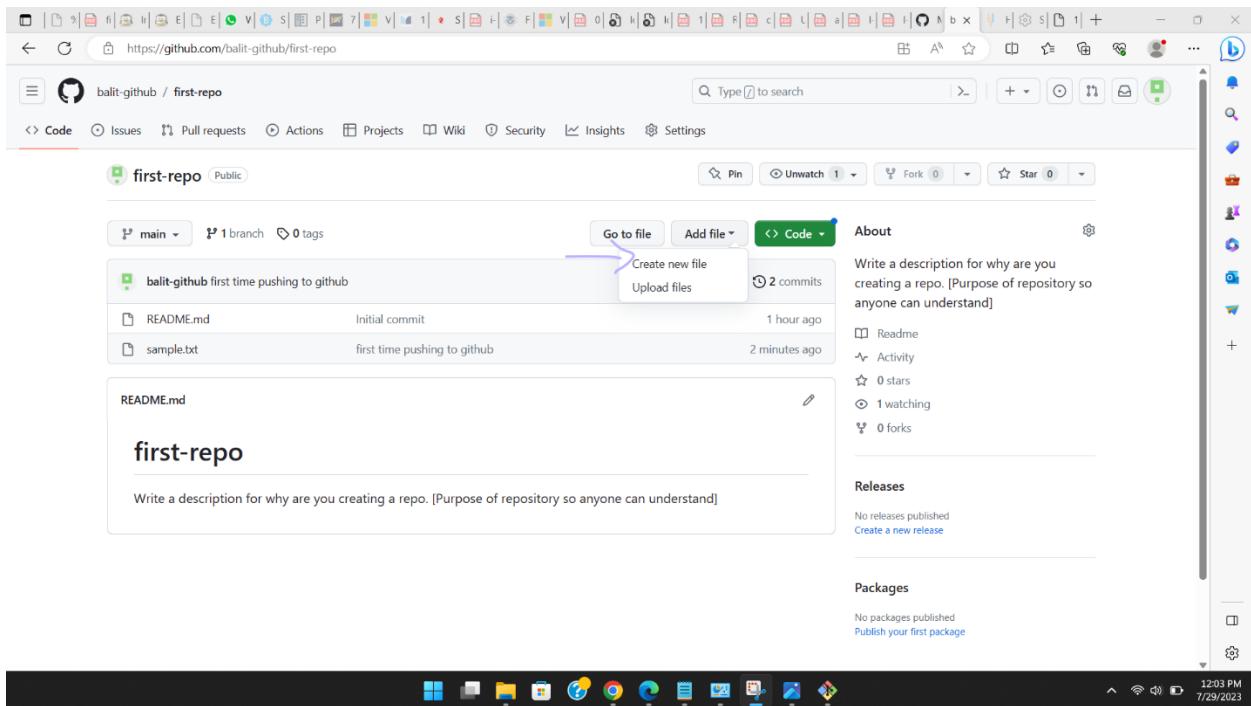


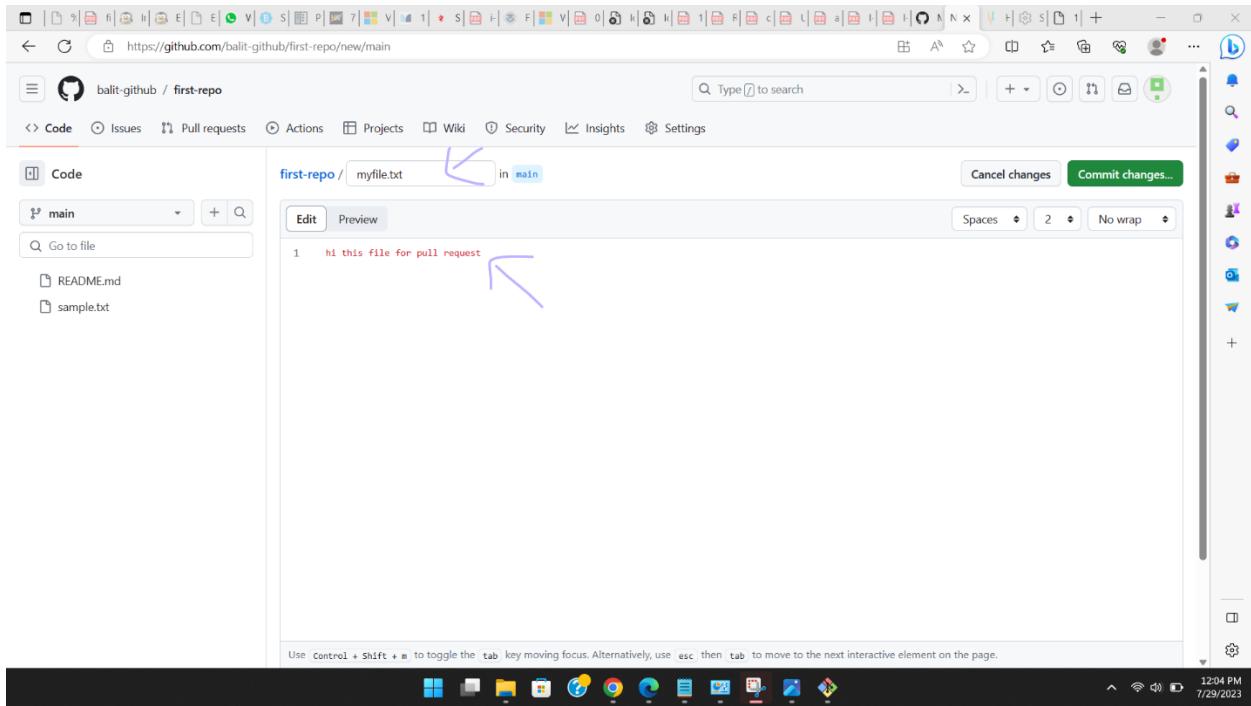
- Run last command again
- **git push -u origin main**
- If you can't understand from documentation, you can prefer video.  
(<https://www.youtube.com/watch?v=CeVwyIVFy-M>)

- Case 2: if that's not a case then please check a remote url again, make sure you copied exact url which you used for cloned. For changing url write below command.
- **git remote set-url origin https://github.com/balit-github/first-repo.git** (if you received **error** like origin is not declared or not defined then add origin first by writing this command “**git remote add origin https://github.com/balit-github/first-repo.git**”)
- now run last command again
- **git push -u origin main**
- If either of case worked you should have sign in pop box appeared on screen and now you can continue you next command which is mention above.

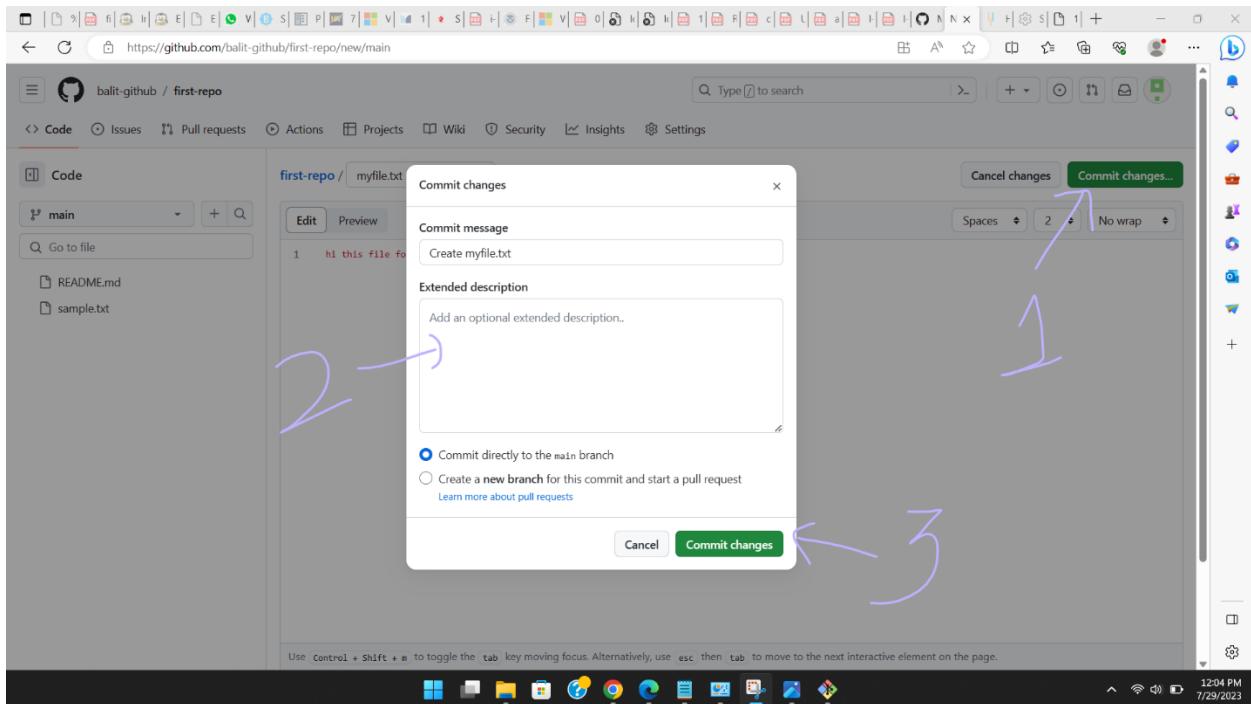
Step 17: Now we successfully push file in github so let's try to pull or download files from github to local machine (here you think like oh we know we can download using git clone command right) but why we download a files which is already we have. So that's why pull command is introduced.

- Create a file in github with some content.





- Commit the changes in repository.



Step 18: let's check in our local machine. Do we have a file which we just created (in my case it is myfile.txt)

- o ls (only two files are there)

The screenshot shows a GitHub repository named 'first-repo'. A terminal window is open in the repository's details page, displaying the following command and its output:

```
git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
```

A blue arrow points from the terminal window to the right sidebar, which shows a history of commits:

- now History
- Last commit date
- 1 hour ago
- now
- 4 minutes ago

The system tray at the bottom right shows the date and time as 7/29/2023 12:05 PM.

## Step 19: let's pull the files

- here we only pull all the files which is not in our local machine.
  - o git pull

The screenshot shows the same GitHub repository 'first-repo'. A terminal window is open, and a blue arrow points to the 'git pull' command in the output:

```
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 721 bytes | 40.00 KiB/s, done.
From https://github.com/balit-github/first-repo
 * [new branch]  main      -> origin/main
Updating d6f93e..a0176cc
 1 file changed, 1 insertion(+)
 create mode 100644 myfile.txt
```

Another blue arrow points to the newly created file 'myfile.txt' in the repository's file list.

The system tray at the bottom right shows the date and time as 7/29/2023 12:05 PM.