

**Name :-** Laxmi Swami

**Topic :-** Git Clone Uses

**Task :-** Clone others repository to our repository also take there files and folders into you're existing repository.

- Create one new repository in you're github.
- Now create an instance and connect it using gitbash.
- Now create one file and folder in you're instance using command line.
- Now push the files and folder to you're github.
- Now now go to others repository by searching username.
- Copy the URL from fork.
- Now fetch the command git clone URL....
- Now all the data will be clone to you're local but to move it in you're existing Repo using

CP filename1 filename2 filename3 path of you're folder

- Now use below commands to push the data on you're repository  
**git add \*** **git commit -m "updated"** **git push add origin master/main**
- Now go to repository and check the files and folder's you copy also check the data inside the files copied or not.

github.com/new

No template

Start your repository with a template repository's contents.

Owner \*  
Laxmismwami

Repository name \*  
newcopy-repo

Your new repository will be created as newcopy-repo.  
The repository name can only contain ASCII letters, digits, and the characters -, ., and \_.

Great repository names are short and memorable. Need inspiration? How about [automatic-rotary-phone](#)?

Description (optional)

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

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

Initialize this repository with:

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

Add .gitignore

.gitignore template: None

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

Choose a license

License: None

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

26°C  
Partly sunny

Search

ENG IN

10:56  
21-07-2025

```
root@ip-172-31-17-185: ~
ls419@LAXMI MINGW64 ~
$ cd Downloads/

ls419@LAXMI MINGW64 ~/Downloads
$ ssh -i "my-key.pem" ubuntu@ec2-34-229-138-80.compute-1.amazonaws.com
The authenticity of host 'ec2-34-229-138-80.compute-1.amazonaws.com (34.229.138.80)' can't be established.
ED25519 key fingerprint is SHA256:WomliMSDyhTteIf40G5GZZXsclgE1Im/vdRDKhKRPw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-229-138-80.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Mon Jul 21 05:02:43 UTC 2025

System load:  0.04          Temperature:   -273.1 C
Usage of /:   25.3% of 6.71GB Processes:      112
Memory usage: 24%          Users logged in: 0
Swap usage:   0%           IPv4 address for ens5: 172.31.17.185

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

Light rain  
in the afternoon

Search

ENG IN

10:33  
21-07-2025

```
root@ip-172-31-25-240: ~/Aws/dev
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 /root/.git/
root@ip-172-31-25-240:~/Aws# git add .
root@ip-172-31-25-240:~/Aws# git commit -m "added"
[master (root-commit) 8121db4] added
  committer: root <root@ip-172-31-25-240.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

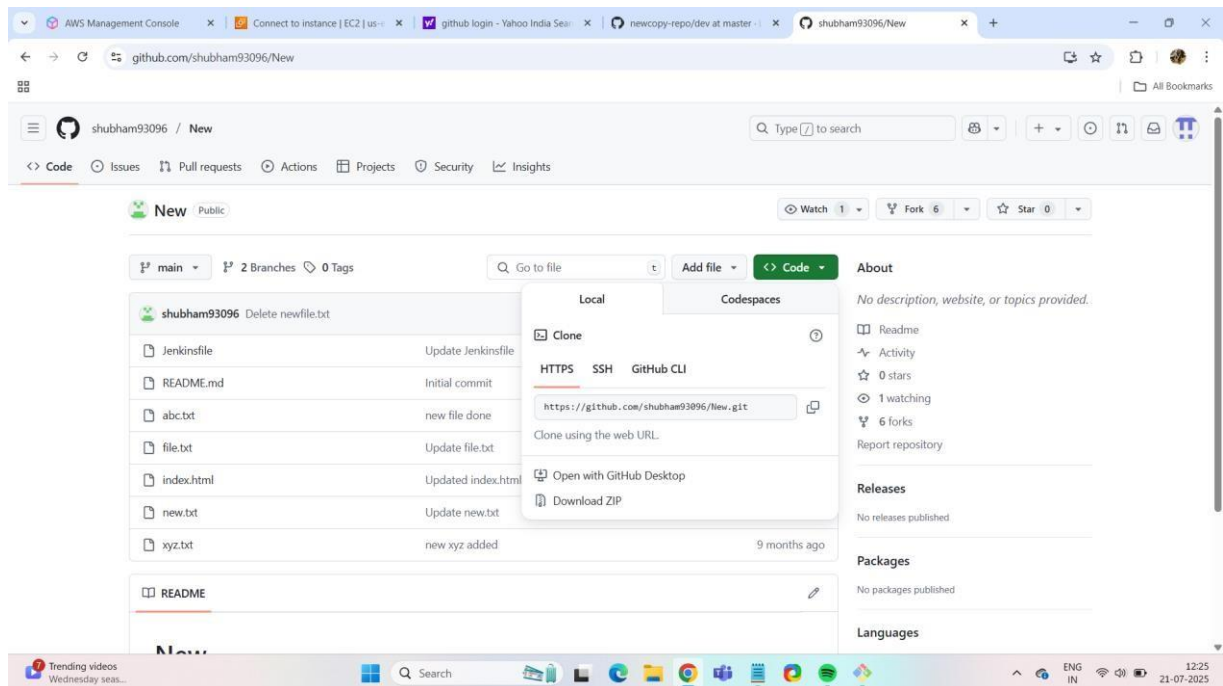
    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

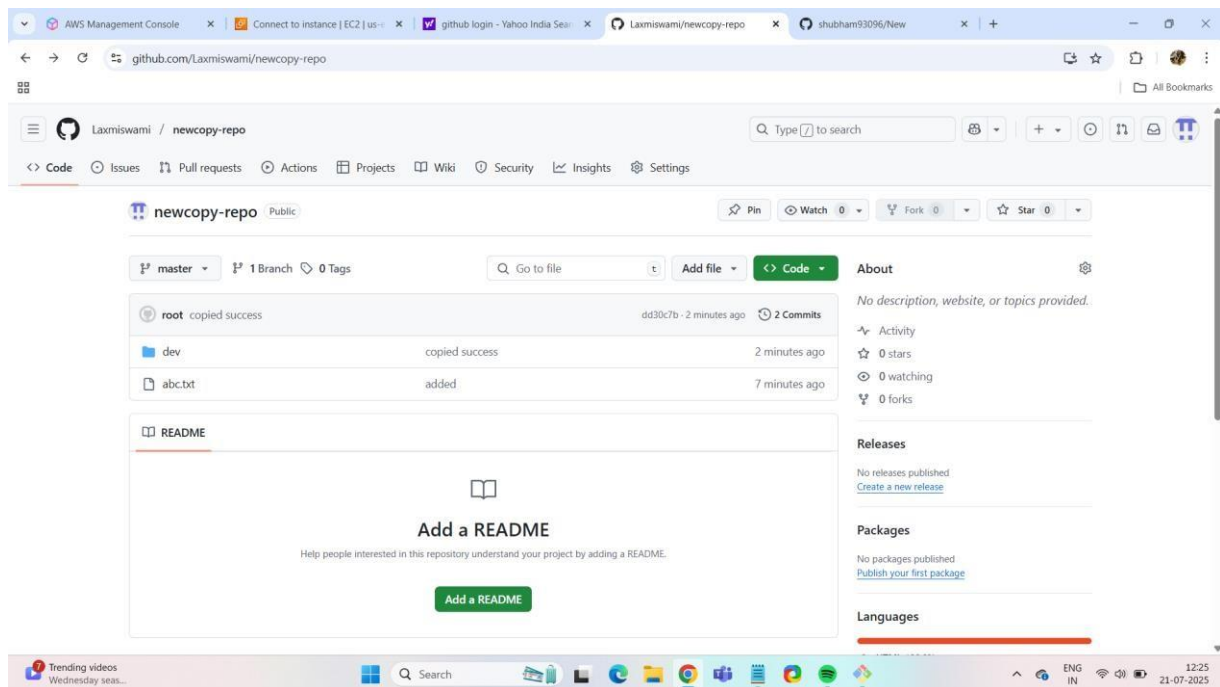
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 abc.txt
root@ip-172-31-25-240:~/Aws# git remote add origin https://github.com/Laxmismwami/newcopy-repo.git
root@ip-172-31-25-240:~/Aws# git push origin master
Username for 'https://github.com': Laxmismwami
Password for 'https://Laxmismwami@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 209 bytes | 209.0 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Laxmismwami/newcopy-repo.git
 * [new branch]      master -> master
root@ip-172-31-25-240:~/Aws# ls
abc.txt
root@ip-172-31-25-240:~/Aws# git clone https://github.com/shubham93096/New.git
```

## Other repository clone



```
root@ip-172-31-25-240: ~/Aws/dev
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 209 bytes | 209.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Laxmishwami/newcopy-repo.git
* [new branch] master -> master
root@ip-172-31-25-240:~/Aws# ls
abc.txt
root@ip-172-31-25-240:~/Aws# git clone https://github.com/shubham93096/New.git
Cloning into 'New'...
remote: Enumerating objects: 79, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (59/59), done.
remote: Total 79 (delta 24), reused 16 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (79/79), 22.05 KiB | 7.35 MiB/s, done.
Resolving deltas: 100% (24/24), done.
root@ip-172-31-25-240:~/Aws# ls
New abc.txt
root@ip-172-31-25-240:~/Aws# cd New
root@ip-172-31-25-240:~/Aws/New# ls
Jenkinsfile README.md abc.txt file.txt index.html new.txt xyz.txt
root@ip-172-31-25-240:~/Aws/New# cd ..
root@ip-172-31-25-240:~/Aws# mkdir dev
root@ip-172-31-25-240:~/Aws# cd New
root@ip-172-31-25-240:~/Aws/New# cd ..
root@ip-172-31-25-240:~/Aws# pwd
/root/Aws
root@ip-172-31-25-240:~/Aws# cd New
root@ip-172-31-25-240:~/Aws/New# cp -r file.txt index.html /root/Aws
root@ip-172-31-25-240:~/Aws/New# ls
Jenkinsfile README.md abc.txt file.txt index.html new.txt xyz.txt
root@ip-172-31-25-240:~/Aws/New# cd ..
root@ip-172-31-25-240:~/Aws# ls
New abc.txt dev file.txt index.html
root@ip-172-31-25-240:~/Aws# cd dev
root@ip-172-31-25-240:~/Aws/dev# ls
root@ip-172-31-25-240:~/Aws/dev# pwd
/root/Aws/dev
root@ip-172-31-25-240:~/Aws/dev# cp -r file.txt index.html /root/Aws/dev
```

After copied the dev folder added here



In that dev the other repository existing file will be here which we are copied:

The screenshot displays the GitHub interface for a repository named 'newcopy-repo'. The 'dev' branch is selected, and the file explorer on the left shows a directory structure with 'file.txt', 'index.html', and 'abc.txt'. The commit history table in the center shows three commits, all with the message 'copied success' and a timestamp of '2 minutes ago'.

Name	Last commit message	Last commit date
..	root	2 minutes ago
file.txt	copied success	2 minutes ago
index.html	copied success	2 minutes ago