

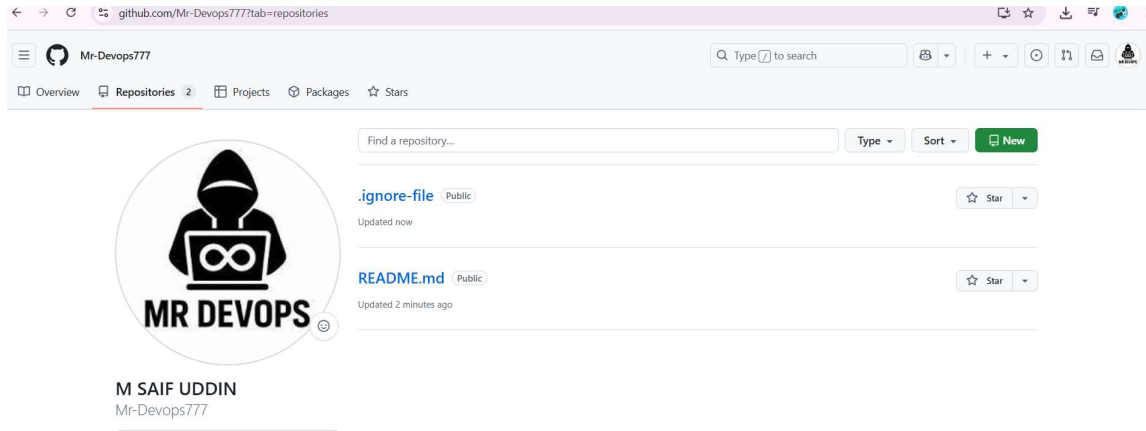
1)Install git.

```
root@ip-172-31-35-195:/home/ec2-user
[root@ip-172-31-35-195 ec2-user]# yum list git
Last metadata expiration check: 0:00:29 ago on wed Aug 13 12:30:14 2025.
Installed Packages
git.x86_64
2.50.1-1.amzn2023.0.1
[root@ip-172-31-35-195 ec2-user]#
```

yum install git -y

yum list git

2)Create a repo in github with README.md and .ignore file.



go to github account and click on "+" button and
add repository--name of repo---public or private---create

3)Clone the created repo to local.

```
root@ip-172-31-35-195:/home/ec2-user/.ignore-file
[root@ip-172-31-35-195 .ignore-file]#
```

git clone "url"

```
root@ip-172-31-35-195:/home/ec2-user/README.md
[root@ip-172-31-35-195 README.md]#
```

4)Create two files in local repo.

```

root@ip-1/2-31-42-84:/home/ec2-user/.git/README.md
[root@ip-172-31-42-84 README.md]# ls
file1 file2 file3
[root@ip-172-31-42-84 README.md]# git stauts
git: 'stauts' is not a git command. See 'git --help'.

The most similar command is
  status
[root@ip-172-31-42-84 README.md]# git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1
        new file:   file2

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file3

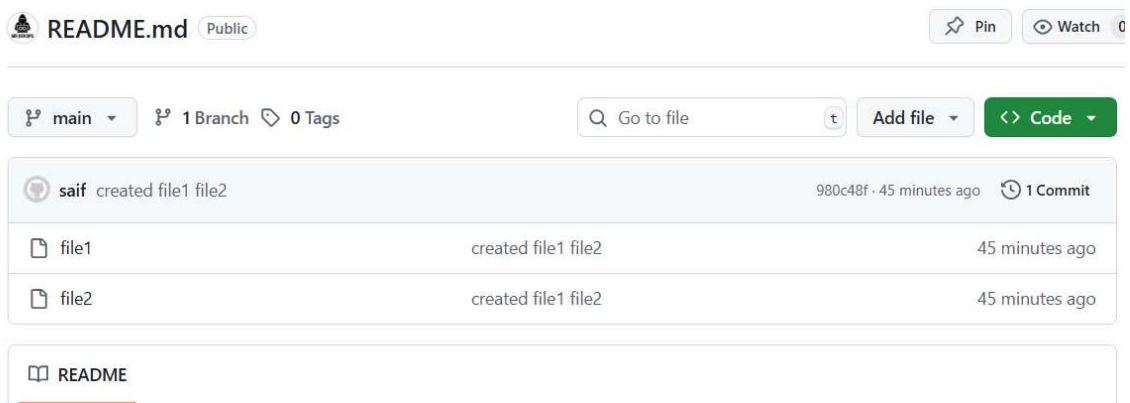
[root@ip-172-31-42-84 README.md]# |

```

mkdir newdir
git init
touch file1 file2
git add .

git status

5)Commit two files and push to central Repository.



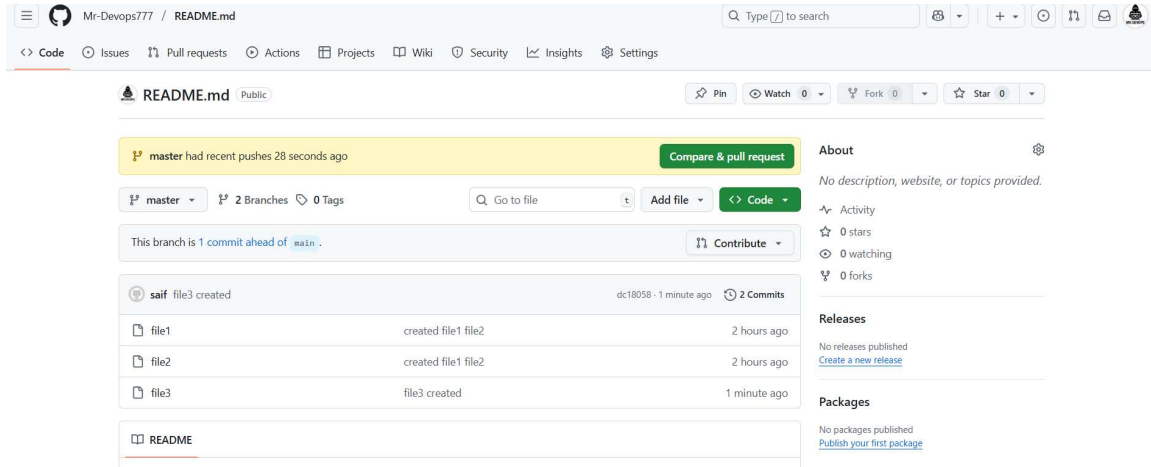
ssh-keygen
cat ~/.ssh/id_rsa.pub

copy public key and paste on github-setting-ssh&GPG-add ssh

git remote set-url origin git@github.com:Mr-Devops777/README.md.git

git push -u origin master

6) Create a branch in local and create a sample file and push to central.



git branch master

git checkout master (automaticall copied all data of previous branch)

create new file using touch

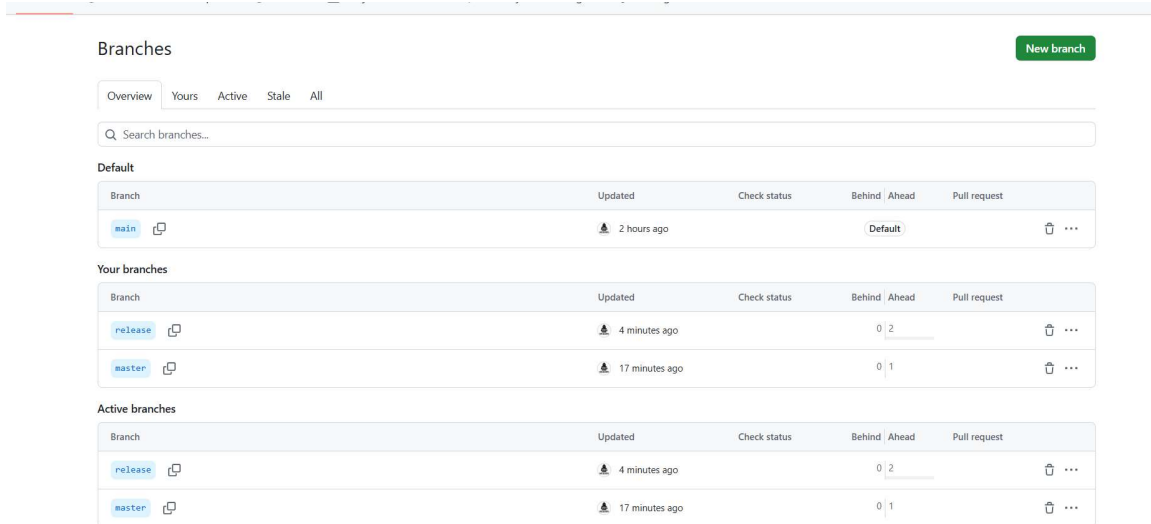
commit add

git status

git push -u origin master

7) Create a branch in github and clone that to local.

```
[root@ip-172-31-42-84 README.md]# ls
file1 file2
[root@ip-172-31-42-84 README.md]# git branch --list
* main
[root@ip-172-31-42-84 README.md]# git checkout release
branch 'release' set up to track 'origin/release'.
Switched to a new branch 'release'
[root@ip-172-31-42-84 README.md]# ls
file1 file2 file3 file4
[root@ip-172-31-42-84 README.md]#
```



created new branch "release" on github
created a file

`git clone --branch <branch name> --single-branch <url>`

`git clone "repo url"`

`git checkout release`

ls

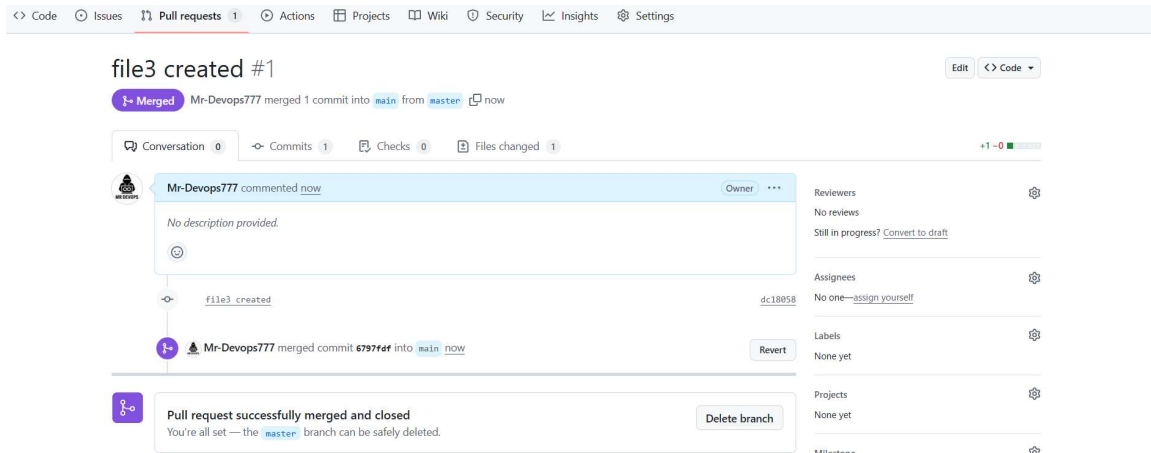
8) Merge the created branch with master in git local.

```
root@ip-172-31-42-84:/home/ec2-user/saif/.git/README.md/README.md
[root@ip-172-31-42-84 README.md]# git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
[root@ip-172-31-42-84 README.md]# ls
file1 file2 file3
[root@ip-172-31-42-84 README.md]# git merge release
Updating dc18058..39e0460
Fast-forward
 file4 | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 file4
[root@ip-172-31-42-84 README.md]# ls
file1 file2 file3 file4
[root@ip-172-31-42-84 README.md]# |
```

`git checkout master`

`git merger release`

9) Merge the created branch with master in github by sending a pull request.

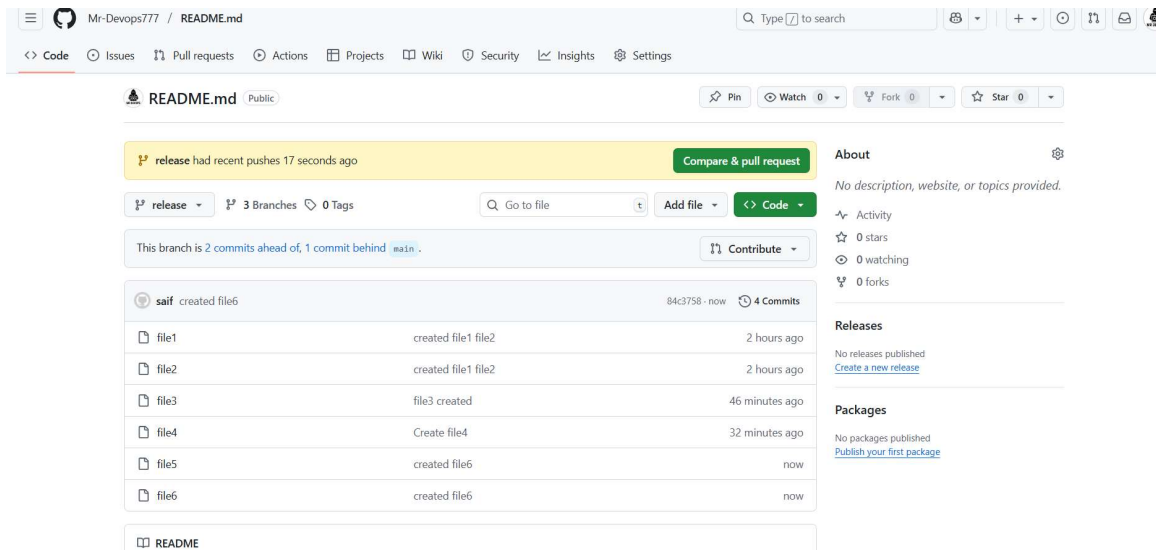


go to pull request under any repo

README.md--pull request--new branch--select branch to merge

commit the message and pull

10)create a file in local and send that to branch in github.



git push

11)clone only a branch from github to local.

MINGW64; c:/Users/DELL/Desktop/gitdir/README.md/README.md

```
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md (devops)
$ git clone --branch master --single-branch https://github.com/Mr-Devops777/README.md.git
Cloning into 'README.md'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 30 (delta 3), reused 12 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (30/30), 7.91 KiB | 3.96 MiB/s, done.
Resolving deltas: 100% (3/3), done.

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md (devops)
$ ls
README.md/  file1  masterfile1

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md (devops)
$ cd README.md/

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md/README.md (master)
$ ls
masterfile  rel-file1

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md/README.md (master)
$ git branch --list
* master

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md/README.md (master)
$
```

git clone --branch <branch name> --single-branch <url>

(to copy only a single branch we use the above command)

12) create a file with all passwords and make that untrackable with git.

```

MINGW64:/c/Users/DELL/Desktop/gitdir/README.md
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md (devops)
$ echo "9063722380" >>secret.txt

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md (devops)
$ cat secret.txt
9063722380
9063722380

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md (devops)
$ echo "secret.txt" >>.gitignore

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md (devops)
$ git status
On branch devops
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore
    new file:   README.md/masterfile
    new file:   README.md/rel-file1
    new file:   README.md/secret-file
    new file:   secretfile

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   .gitignore
    deleted:    secretfile

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md (devops)
$ git commit -m "secret.txt file created"
[devops 56dc904] secret.txt file created
5 files changed, 5 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md/masterfile
create mode 100644 README.md/rel-file1
create mode 100644 README.md/secret-file
create mode 100644 secretfile

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md (devops)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md (devops)
$ git status
On branch devops
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   .gitignore

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    secretfile

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/gitdir/README.md (devops)
$

```

echo "9063722380">>secret.txt

echo"secret.txt">>.gitignore

git commit -m "secret.txt created"

git add .gitignore

git status

13)make a commit and make that commit reset without savings changes.

```

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git status
On branch release
Your branch is ahead of 'origin/release' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ echo "first reset">reset-file

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ ls
file1-release reset-file revert.txt sample-release

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git status
On branch release
Your branch is ahead of 'origin/release' by 1 commit.
(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:   reset-file

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

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git add reset-file
warning: in the working copy of 'reset-file', LF will be replaced by CRLF the next time Git touches it

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git status
On branch release
Your branch is ahead of 'origin/release' by 1 commit.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   reset-file

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git commit -m "first reset in reset-file"
[release 5c22525] first reset in reset-file
1 file changed, 1 insertion(+), 1 deletion(-)

```



```

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git commit -m "first reset in reset-file"
[release 5c22525] first reset in reset-file
1 file changed, 1 insertion(+), 1 deletion(-)

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git log --oneline
5c22525 (HEAD -> release) first reset in reset-file
e56881c Revert
2c29f9d (origin/release) second added revert.txt
d3a1f7c first revert.txt
c3b1346 Revert "added reset.txt"
b92c8a6 added reset.txt
4ffbbf8 added reset file
f9c439e (origin/main, origin/HEAD, main) added sample-release file
19211b0 Create file1-release
49a2c20 Delete file2.txt
54bf593 Delete file1.txt
3480bc2 Delete README.md
cf7902f third commit updated file1.txt
64084e0 second commit updated file1.txt
54cd96b first commit added file1.txt file2.txt
603e667 Initial commit

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git reset e56881c
Unstaged changes after reset:
M   reset-file

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git status
On branch release
Your branch is ahead of 'origin/release' by 1 commit.
  (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:   reset-file

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

```

first create a file

echo "first reset">reset-file

git add reset-file

git commit -m "added first reset in reset-file"

git status (it will show the changes in tracking area/staging area)

git log --oneline (to check the commits it will show your commit)

to move the file from tracking area to untracking area we use

git reset <one previous commit id>

git log --oneline (it will delete your latest commit)

git status (your file will be in staging area to make it track again we need to add through <git add filename>

then again we need to commit

and hence we can push to central repo

git reset previous commit id to latest (it will take by default git reset mixed and keep the file in staging area)

git reset --soft previous commit id (delete the commit and keeps the changes in staging area)

git reset --mixed previous commit id (delete the commit and keeps the changes in working area)

git reset --hard previous commit id (delete the changes and also deletes the changes in repo)

git reset command is used in local repo only. it enables to make changes in files and its commits, we cannot modify the commits and changes if pushed to central repo (for these we need to use git revert if pushed to central)

14) Revert a committed commit to the older version.

```
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git log --oneline
6ded67f (HEAD -> release) Revert "second commit"
654172e third commitwq!
beb4246 second commit
6f44f8e first commit
a021dbb (origin/release) Revert "third commit deleted in revert-file3"
b4e9e0e third commit in revert-file3
79c69ec second commit revert-file3
290e103 first commit in revert-file3
311c138 Revert "first revert on revert-file2"
2c42149 first revert on revert-file2
7bcce14 first commit in revert-file1
42bc227 added reset.txt
c18c76c added reset file
3d6016f added sample-release file
3824c36 Create file1-release
54bf593 Delete file1.txt
3480bc2 Delete README.md
cf7902f third commit updated file1.txt
64084e0 second commit updated file1.txt
54cd96b first commit added file1.txt file2.txt
603e667 Initial commit

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git reset --hard 654172e
HEAD is now at 654172e third commitwq!

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ cat file4
first commit
second commit
third commit

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git status
On branch release
Your branch is ahead of 'origin/release' by 3 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git revert 654172e
[release ec3c987] Revert "third commitwq!"
1 file changed, 1 deletion(-)

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ cat file4
first commit
second commit
```

work flow

```
echo "first commit">file4
```

```
git add file4
```

```
git commit -m "first commit"
```

```
vi file4 (add second commit in it)
```

```
git add file4
```

```
git commit -m "second commit"
```

```
vi file4 (add third commit in it)
```

```
git add file4
```

```
git commit -m "third commit"
```

```
git log --oneline(to check the commits)
```

git revert commit_id (it will open the vi editor where you need to save the file simply without overwriting in it. It will delete the previous changes in your file and creates a new commit for these by default)

15) push a file to stash without saving the changes and work on another file.

```

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git commit -m "second commit on stash"
[release 79c5d06] second commit on stash
3 files changed, 3 insertions(+), 1 deletion(-)
create mode 100644 login.js
delete mode 100644 stash

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git status
On branch release
Your branch is ahead of 'origin/release' by 5 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ vi stash.txt

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git status
On branch release
Your branch is ahead of 'origin/release' by 5 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:   stash.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        1

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

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git checkout master
error: Your local changes to the following files would be overwritten by checkout:
        stash.txt
Please commit your changes or stash them before you switch branches.
Aborting

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git stash
warning: in the working copy of 'stash.txt', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state WIP on release: 79c5d06 second commit on stash

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git stash
No local changes to save

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git stash list
stash@{0}: WIP on release: 79c5d06 second commit on stash

```

```

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git status
On branch release
Your branch is ahead of 'origin/release' by 5 commits.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
1
nothing added to commit but untracked files present (use "git add" to track)
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ ls
1 file1-release file2.txt file4 login.js reset-file reset.txt revert-file1 revert-file3 sample-release stash-file stash.txt task-file
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ vi login.js
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git add login.js
warning: in the working copy of 'login.js', LF will be replaced by CRLF the next time Git touches it
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git add login.js
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git commit -m "added login.js"
[release 02de7e2] added login.js
1 file changed, 1 insertion(+)
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git stash pop
On branch release
Your branch is ahead of 'origin/release' by 6 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:   stash.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
1
no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (29b7c21ced0b74eb046d8ea89c3860dd94f55777)
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ vi stash.txt

```

```

$ vi stash.txt
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git status
On branch release
Your branch is ahead of 'origin/release' by 6 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:   stash.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
1
no changes added to commit (use "git add" and/or "git commit -a")
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git add stash.txt
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git commit -m "third commit added to stash.txt"
[release dfefae7] third commit added to stash.txt
1 file changed, 2 insertions(+)

```

echo "first stash">stash.txt

git add stash.txt

git commit -m "first commit on stash.txt"

echo "second stash">>stash.txt

git add stash.txt

git commit -m "second commit on stash.txt"

echo "third commit">>stash.txt

git status (shows stash.txt in working area)

git stash (to push all files in working area to stashing memory)

git stash push -m (here you can write a message)stash.txt {it will add only stash.txt with a message}

git stash list (it will show the list of files in stash memory)

git status (it will not show stash.txt in working area)

now you can work on any other file and add it and commit it

then switch back to stash using

git stash pop

echo "fourth stash">>stash.txt

git add stash.txt

git commit -m "third commit on stash.txt"

16)undo the stash file and start working on that again.

```

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git stash pop
On branch release
Your branch is ahead of 'origin/release' by 6 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:   stash.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        1

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (29b7c21ced0b74eb046d89c3860dd94f55777)

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ vi stash.txt

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git status
On branch release
Your branch is ahead of 'origin/release' by 6 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:   stash.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        1

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

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git add stash.txt

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md (release)
$ git commit -m "third commit added to stash.txt"
[release dfefae7] third commit added to stash.txt
1 file changed, 2 insertions(+)

```

git stash pop stash_id (to move a selected file from stashing to working)

git stash pop (to move all files from stashing to working)

again you can continue your work on file file

then check status

git status (it show the file in working area)

git add stash.txt (to add the file in staging area)

git commit -m "last commit" (finally your file is ready for push to central repo)

17)generate a ssh-keygen and configure into github.

Settings

🔍 Type 7 to search

Okay, you have successfully deleted that key.

ssh-keygen

define the name again after ?keyfile.pub

define again

cat keyfile.pub or cat id_rsa.pub(copy the file code and paste it on)

git-hub/account/setting/ssh and keygen/add ssh key (define name of the ssh key and paste the code here like asjbdshndald@#\$^&hgsvdhasbjhgs13135154hsdjhgdu)

18)configure webhooks to github.

19)basic understanding of .git file.

a.(HEAD):

A file that points to the current branch or commit you're working on.

Example: ref: refs/heads/main

b. (config)

Contains repository-specific configuration (like remote URL, username, etc.).

c. (description)

A text file used by GitWeb (not usually important in daily use).

d. (hooks/)

Scripts that run on certain Git events (like before a commit or after a push).

Example: pre-commit hook can check code formatting before allowing a commit.

e. (.info/)

Contains exclude file – similar to .gitignore, but applies only to this local repo.

f. (objects/)

Stores all the actual data (commits, trees, blobs, tags).

Each object is saved in a compressed format, indexed by its SHA-1 hash.

Example:

Blob → stores file contents

Tree → stores directories and file structure

Commit → stores metadata + reference to tree & parent commits

g. (refs/)

Contains references to commits.

Subdirectories:

heads/ → local branches (e.g., main, dev)

remotes/ → remote-tracking branches (e.g., origin/main)

tags/ → tags pointing to specific commits

20) Check all the logs of git.

git log → commit history (normal project history)

git log --oneline

git log --oneline --graph --all → pretty tree of all branches

```
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/main-branch/README.md/.git/hooks (GIT_DIR!)
$ git log --oneline --graph --all
* dfefae7 (HEAD -> release) third commit added to stash.txt
* 02de7e2 added login.js
* 79c5d06 second commit on stash
* c2009fc first commit added stash.txt
* 8ce0e66 second commit on stash-file
* 17ec1aa added task file
* afdcaf8 added stash file
* ec3c987 (origin/release) Revert "third commitwq!"
* 654172e third commitwq!
* beb4246 second commit
* 6f44f8e first commit
* a021dbb Revert "third commit deleted in revert-file3"
* b4e9e0e third commit in revert-file3
* 79c69ec second commit revert-file3
* 290e103 first commit in revert-file3
* 311c138 Revert "first revert on revert-file2"
* 2c42149 first revert on revert-file2
* 7bcece14 first commit in revert-file1
* 42bc227 added reset.txt
* c18c76c added reset file
* 3d6016f added sample-release file
* 3824c36 Create file1-release
* a1990d5 (master) second commit on stashing
* fad9268 (origin/master) sample file added to master
* 8630992 Merge pull request #1 from Mr-Devops777/release
|
| * 18bdec3 first commit in master added file1-master.txt file2-master.txt
| * f9c439e (origin/main, origin/HEAD, main) added sample-release file
|
| * 19211b0 Create file1-release
| * 49a2c20 Delete file2.txt
|
|/
* 54bf593 Delete file1.txt
* 3480bc2 Delete README.md
|
|/
* cf7902f third commit updated file1.txt
* 64084e0 second commit updated file1.txt
* 54cd96b first commit added file1.txt file2.txt
* 603e667 Initial commit
```

git reflog → everything HEAD has done (recover lost commits)

```

MINGW64/c/Users/UELL/Desktop/project-1/main-branch/README.md/.git/hooks
dfe7ae7 (HEAD -> release) HEAD@{0}: commit: third commit added to stash.txt
02de7e2 HEAD@{1}: commit: added login.js
79c5d06 HEAD@{2}: reset: moving to HEAD
79c5d06 HEAD@{3}: commit: second commit on stash
c2009fc HEAD@{4}: commit: first commit added stash.txt
8ce0e66 HEAD@{5}: commit: second commit on stash-file
17ec1aa HEAD@{6}: commit: added task file
afdca8 HEAD@{7}: reset: moving to HEAD
afdca8 HEAD@{8}: checkout: moving from master to release
a1990d5 (master) HEAD@{9}: commit: second commit on stashing
fad9268 (origin/master) HEAD@{10}: checkout: moving from release to master
afdca8 HEAD@{11}: commit: added stash file
ec3c987 (origin/release) HEAD@{12}: checkout: moving from master to release
fad9268 (origin/master) HEAD@{13}: checkout: moving from release to master
ec3c987 (origin/release) HEAD@{14}: checkout: moving from master to release
fad9268 (origin/master) HEAD@{15}: checkout: moving from release to master
ec3c987 (origin/release) HEAD@{16}: revert: Revert "third commitwq!"
654172e HEAD@{17}: reset: moving to 654172e
6ded67f HEAD@{18}: commit: Revert "second commit"
654172e HEAD@{19}: commit: third commitwq!
9eb4246 HEAD@{20}: commit: second commit
6f44f8e HEAD@{21}: commit: first commit
a021dbb HEAD@{22}: reset: moving to a021dbb
34c7aea HEAD@{23}: reset: moving to 34c7ae80c8b5df1a31b842d51e8917820db0cae
34c7aea HEAD@{24}: reset: moving to 34c7aea
18c8176 HEAD@{25}: commit: Revert "second commit deleted on file 4"
fd2a471 HEAD@{26}: commit: Revert "second commit on file 4"
34c7aea HEAD@{27}: commit: third commit on file 4
9b179a0 HEAD@{28}: commit: second commit on file 4
dcc1a5c HEAD@{29}: commit: first commit on file 4
a021dbb HEAD@{30}: revert: Revert "third commit deleted in revert-file3"
b4e9e0e HEAD@{31}: reset: moving to b4e9e0e
8343385 HEAD@{32}: commit: Revert "second commit revert-file3"
631466c HEAD@{33}: commit: Revert "second commit revert-file3"
b4e9e0e HEAD@{34}: reset: moving to b4e9e0e89e006a1d6269cce6e39b637663b67cb
b4e9e0e HEAD@{35}: reset: moving to b4e9e0e89e006a1d6269cce6e39b637663b67cb
b4e9e0e HEAD@{36}: reset: moving to b4e9e0e
0d9453f HEAD@{37}: commit: Revert "second commit revert-file3"
2c988fb HEAD@{38}: reset: moving to 2c988fb653e84cb6fd592b2eaf47fb421dbf0174
2c988fb HEAD@{39}: commit: "second commit revert-file3"
b4e9e0e HEAD@{40}: commit: third commit in revert-file3
79c69ec HEAD@{41}: commit: second commit revert-file3
290e103 HEAD@{42}: commit: first commit in revert-file3
311c138 HEAD@{43}: revert: Revert "first revert on revert-file2"
2c42149 HEAD@{44}: reset: moving to 2c42149
8249c01 HEAD@{45}: commit: revert file
2c42149 HEAD@{46}: commit: first revert on revert-file2
7bcce14 HEAD@{47}: rebase (continue) (finish): returning to refs/heads/release
7bcce14 HEAD@{48}: rebase (start): checkout 7bcce14
eaa42cb HEAD@{49}: revert: Revert "second commit in revert-file1"
64341e3 HEAD@{50}: rebase (finish): returning to refs/heads/release
64341e3 HEAD@{51}: rebase (start): checkout 7bcce14
64341e3 HEAD@{52}: rebase (finish): returning to refs/heads/release
64341e3 HEAD@{53}: rebase (start): checkout 64341e3
2d6b547 HEAD@{54}: rebase (finish): returning to refs/heads/release
2d6b547 HEAD@{55}: rebase (start): checkout 64341e3
2d6b547 HEAD@{56}: revert: Revert "good commit in revert-file1"
;

```

.git/logs/refs/ → raw logs per branch

git stash list → stash logs

21)Rename the commit message.

MINGW64; c:/Users/DELL/Desktop/project-1/README.md

```
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/README.md (master)
$ git log --oneline
98af08e (HEAD -> master, origin/master) sample file created
8630992 Merge pull request #1 from Mr-Devops777/release
19211b0 Create file1-release
49a2c20 Delete file2.txt
54bf593 Delete file1.txt
3480bc2 Delete README.md
18bdec3 first commit in master added file1-master.txt file2-master.txt
cf7902f third commit updated file1.txt
64084e0 second commit updated file1.txt
54cd96b first commit added file1.txt file2.txt
603e667 Initial commit

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/README.md (master)
$ git commit --amend -m "sample file added"
[master 04e6053] sample file added
Date: Mon Sep 8 02:52:49 2025 +0530
1 file changed, 2 insertions(+)
create mode 100644 sample-file

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/README.md (master)
$ git push --force
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 347 bytes | 347.00 KiB/s, done.
Total 3 (delta 1), reused 2 (delta 1), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Mr-Devops777/README.md.git
+ 98af08e...04e6053 master -> master (forced update)

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/README.md (master)
$ git log --oneline
04e6053 (HEAD -> master, origin/master) sample file added
8630992 Merge pull request #1 from Mr-Devops777/release
19211b0 Create file1-release
49a2c20 Delete file2.txt
54bf593 Delete file1.txt
3480bc2 Delete README.md
18bdec3 first commit in master added file1-master.txt file2-master.txt
cf7902f third commit updated file1.txt
64084e0 second commit updated file1.txt
54cd96b first commit added file1.txt file2.txt
603e667 Initial commit
```

git commit --amend -m "add new commit message in it" (it will replace a new commit message)

(it will modify the latest commit only)

git push --force (to push all changes to central repo)

git log --oneline

22) Merge multiple commits into single commit..

```

$ git log --oneline
8dba242 (HEAD -> main) third commit
3319851 second commit
d8a8e8d first commit
a8ef705 (origin/main, origin/HEAD) Initial commit

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/testing (main)
$ git rebase -i HEAD~4
fatal: invalid upstream 'HEAD~4'

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/testing (main)
$ git rebase -i HEAD~3
error: cannot rebase: Your index contains uncommitted changes.
error: Please commit or stash them.

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/testing (main)
$
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/testing (main)
$ git rebase -i HEAD~3
error: cannot rebase: Your index contains uncommitted changes.
error: Please commit or stash them.

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/testing (main)
$ git add .
DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/testing (main)
$ git commit -m "WIP: save changes before rebase"
[main 34c89aa] WIP: save changes before rebase
1 file changed, 2 insertions(+)

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/testing (main)
$ git commit -m "WIP: save changes before rebase"
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/testing (main)
$ git rebase -i HEAD~N
fatal: invalid upstream 'HEAD~N'

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/testing (main)
$ git rebase -i HEAD~4
[detached HEAD 4a2ddac] first commit
Date: Tue Sep 9 18:56:14 2025 +0530
1 file changed, 3 insertions(+)
create mode 100644 file1.txt
Successfully rebased and updated refs/heads/main.

DELL@DESKTOP-RCNKJC4 MINGW64 ~/Desktop/project-1/testing (main)
$ git log --oneline
4a2ddac (HEAD -> main) first commit
a8ef705 (origin/main, origin/HEAD) Initial commit

```

git log --oneline

git rebase -i HEAD~4 (vi editor mode opens under change all picks with squash except the one where you want to add all commit)

git log --oneline