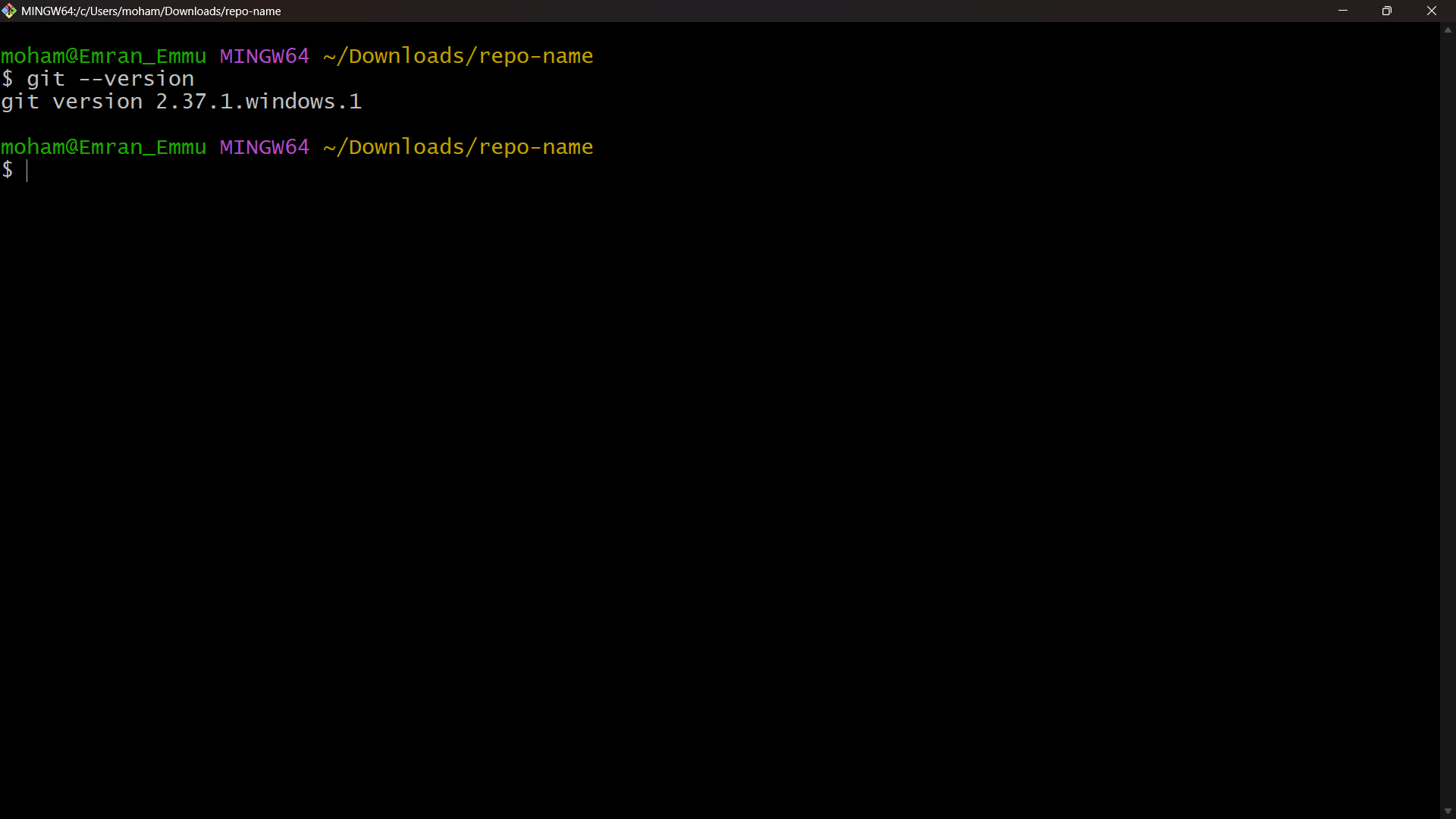
1)Install git.

sudo yum install git -y



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

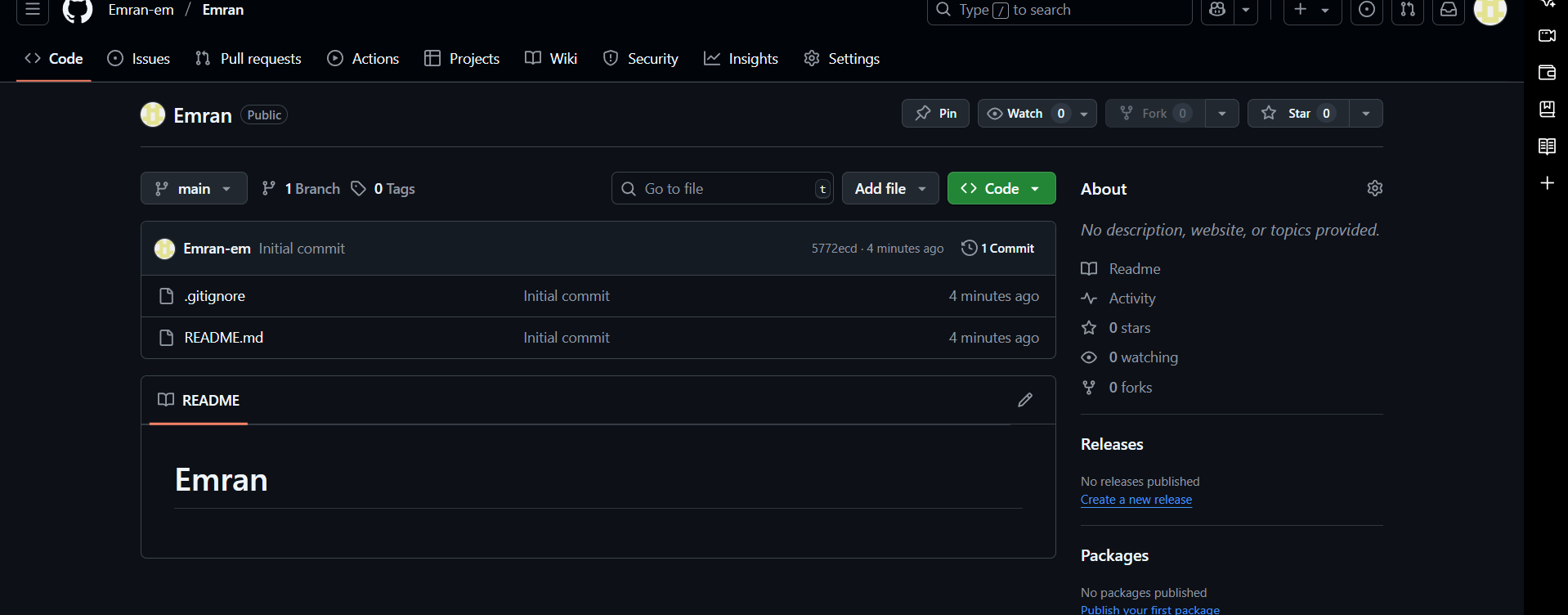
Log in to **GitHub**.

Click **New Repository** → Give **name**.

Select:

* ✅ Initialize with **README.md**
* ✅ Add .gitignore (choose template, e.g., Linux)

Click **Create Repository**.



3)Clone the created repo to local.

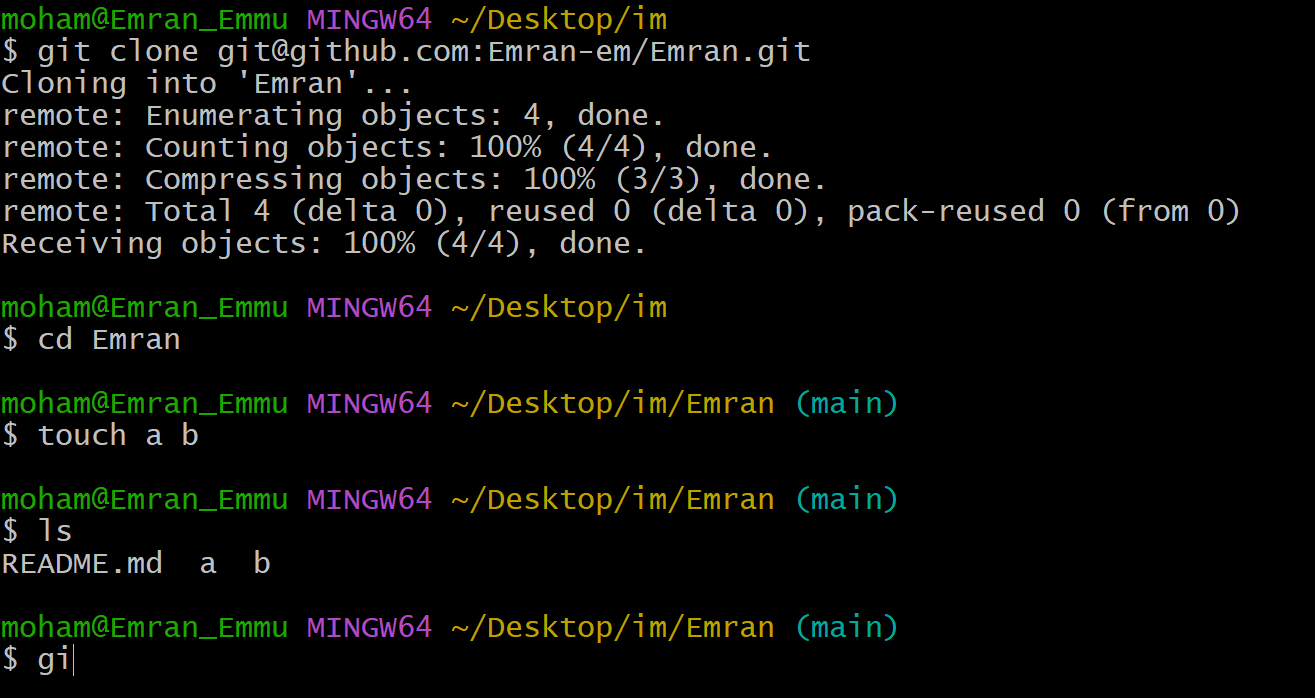
Copy the repo **HTTPS/SSH URL** from GitHub:

git clone [https://github.com/<username>/<repo>.git](https://github.com/%3cusername%3e/%3crepo%3e.git)

OR via SSH:

git clone git@github.com:<username>/<repo>.git

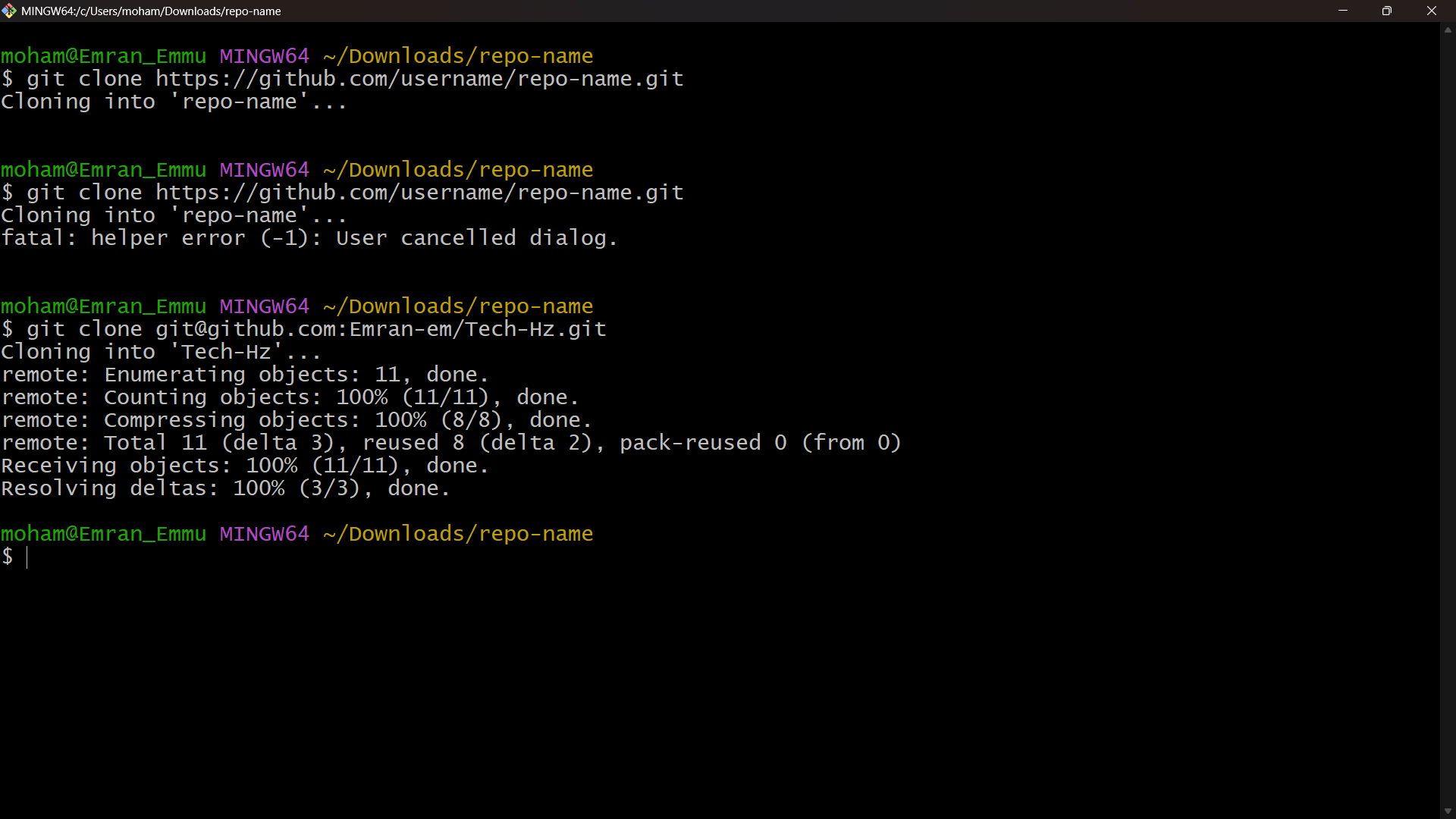
4)Create two files in local repo  
  
cd <repo>

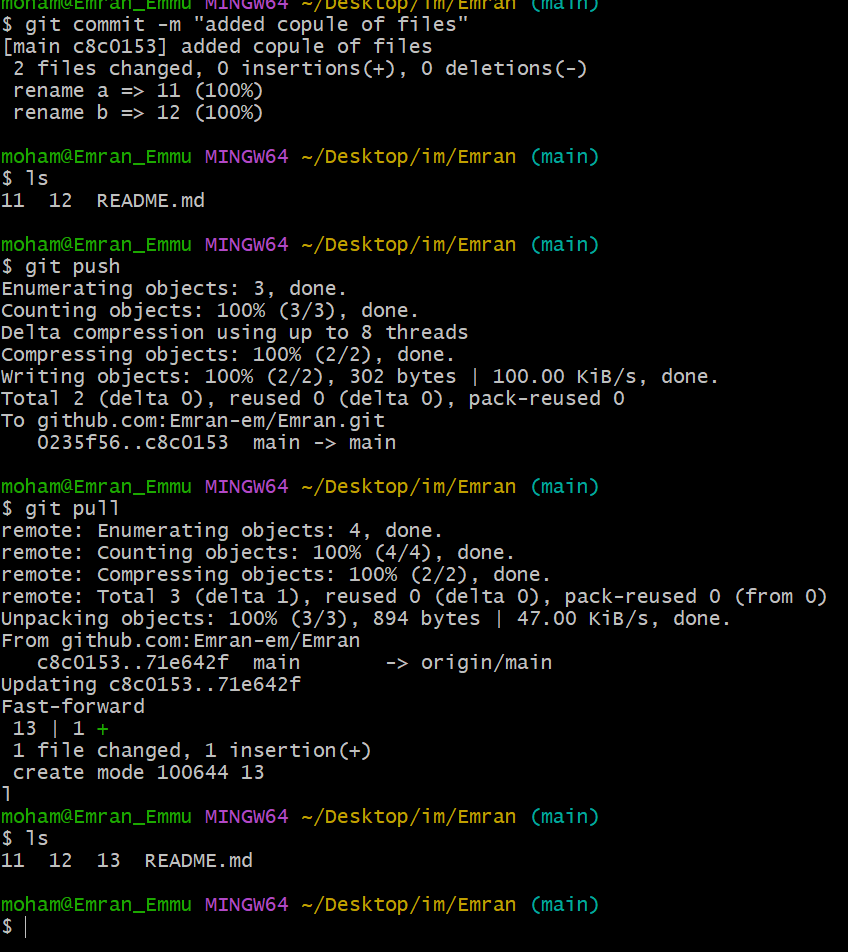
touch file1.txt file2.txt  


5)Commit two files and push to central Repository.

git add file1.txt file2.txt

git commit -m "Added two files"

git push origin main



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

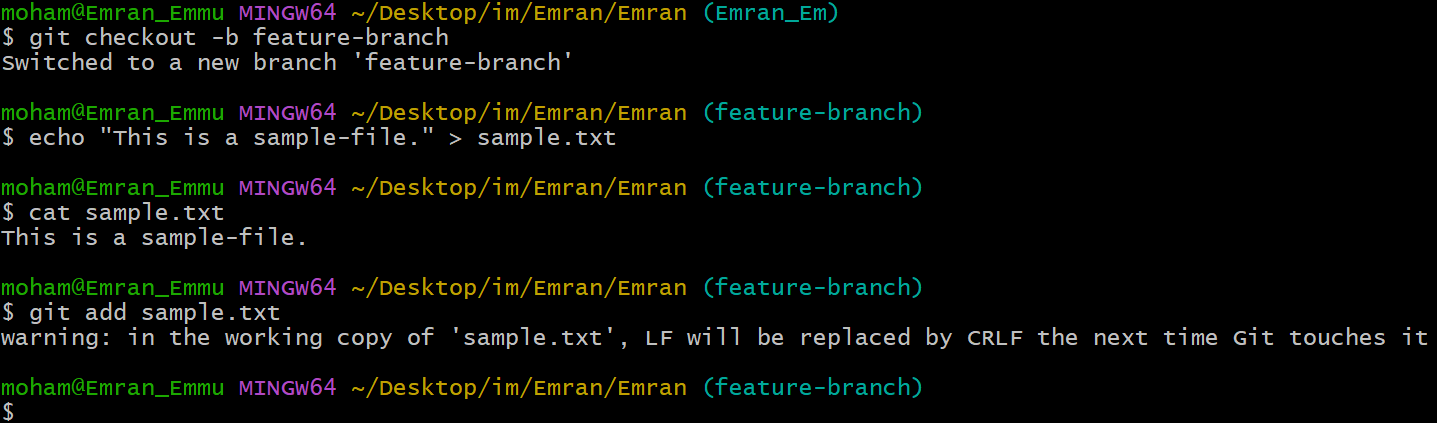
git checkout -b feature-branch

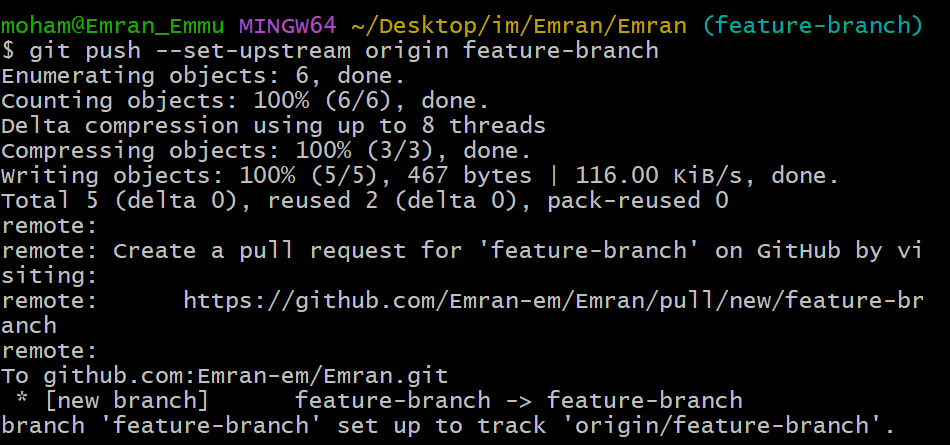
touch sample.txt

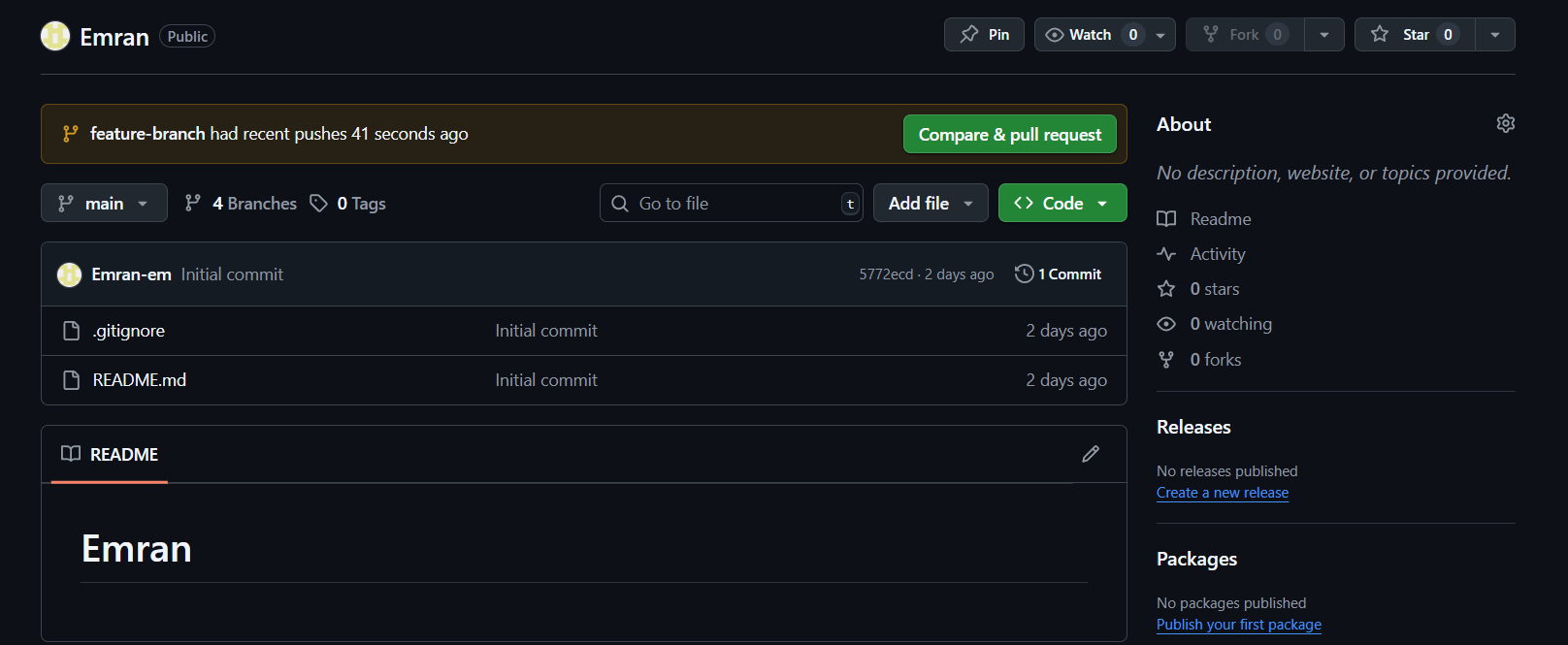
git add sample.txt

git commit -m "Added sample file in feature-branch"

git push origin feature-branch







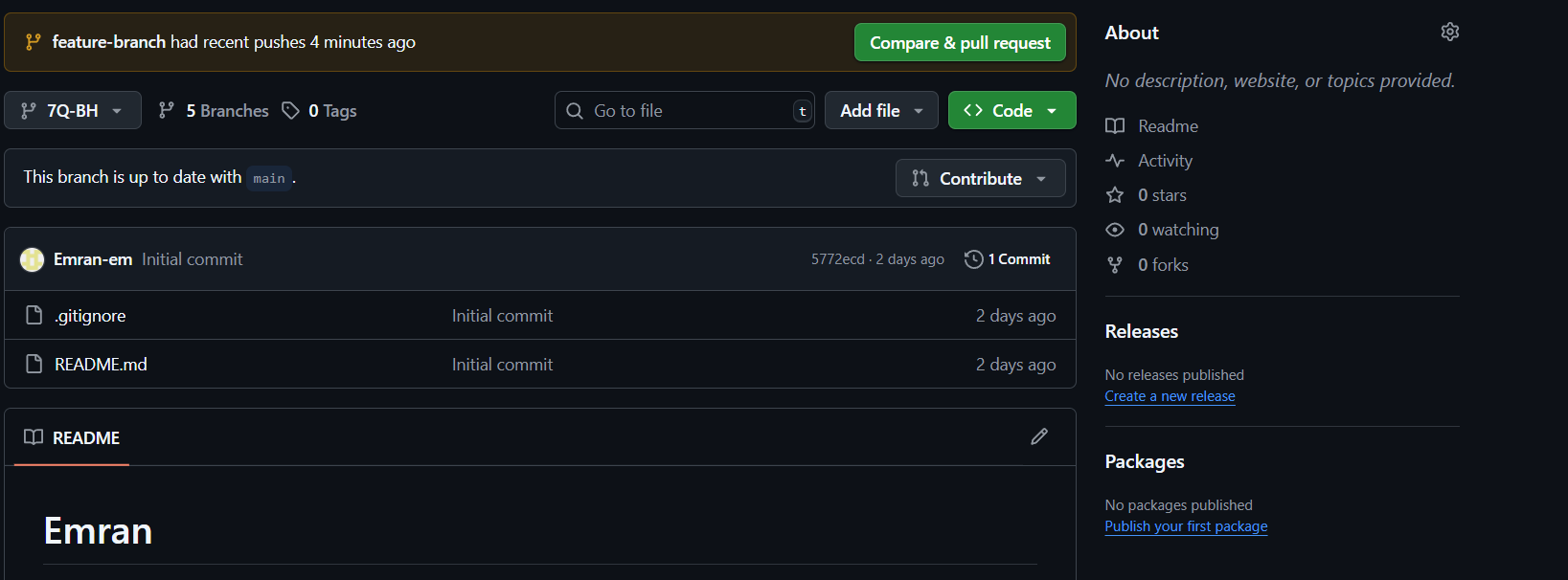
7)Create a branch in github and clone that

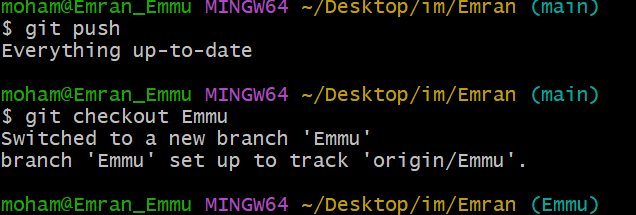
Go to GitHub → Click **Branches** → Create new branch github-branch.

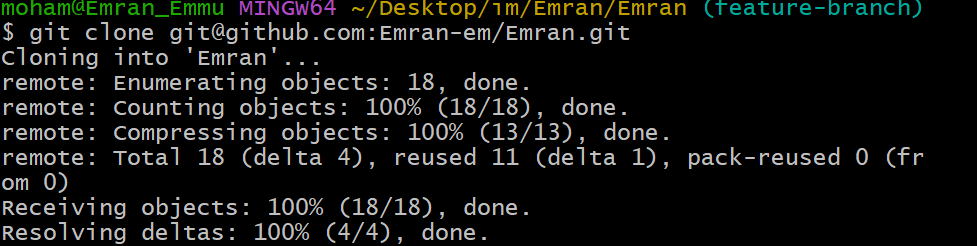
Then pull it locally:

🡪git fetch origin

🡪git checkout github-branch



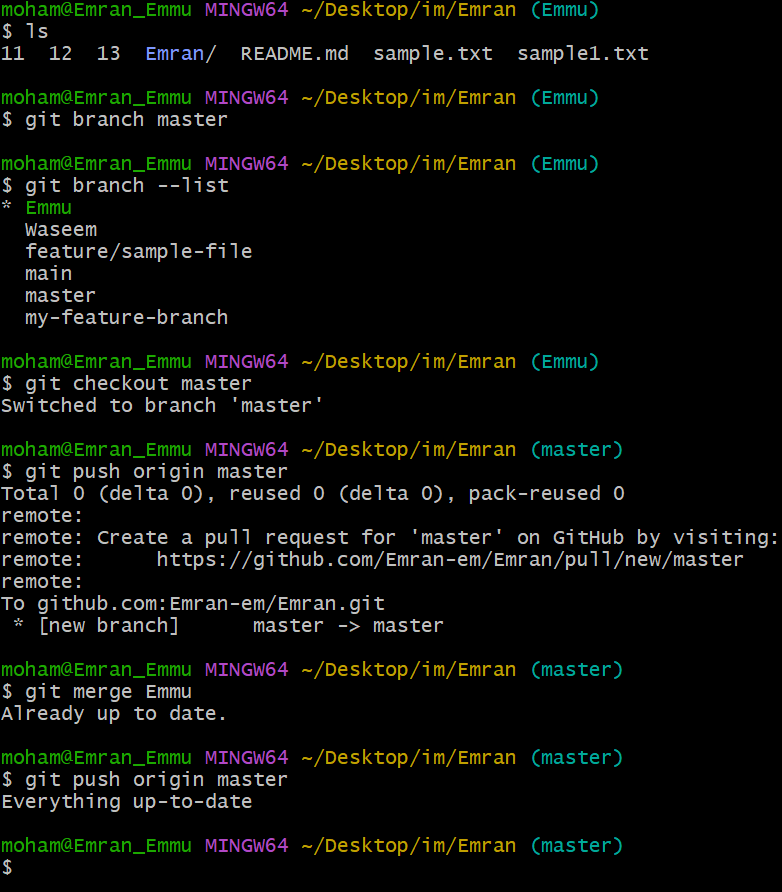


  
  
8)Merge the created branch with master in git local.

git checkout main

git merge feature-branch

git push origin main

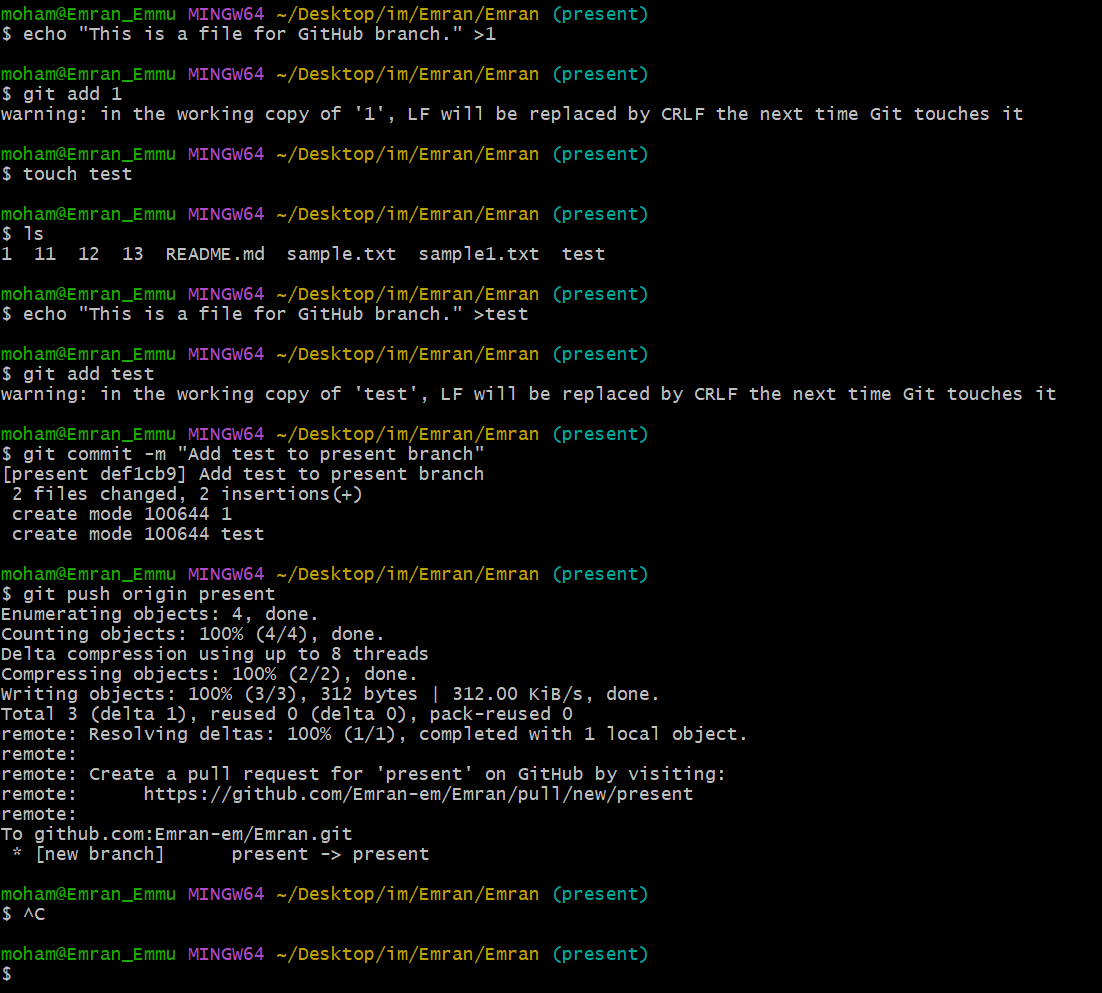


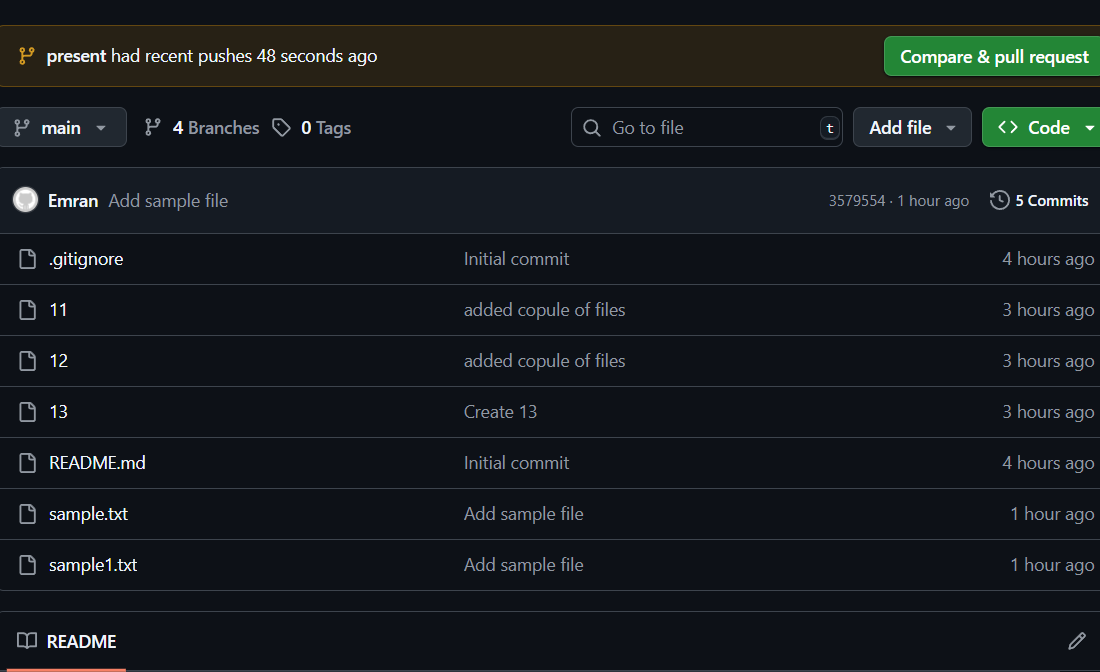
9)Merge the created branch with master in GitHub by sending a pull request.

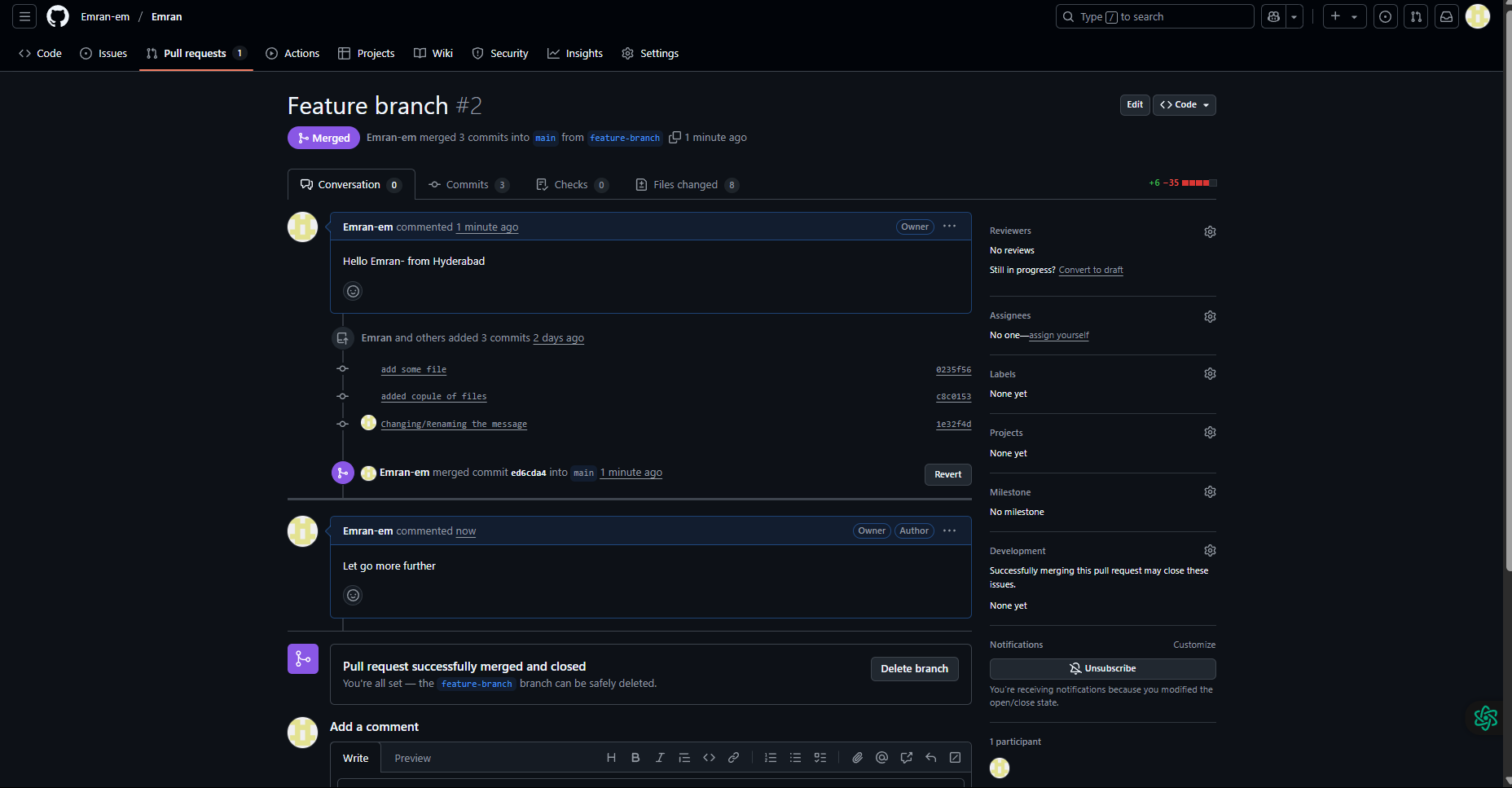
Go to GitHub → **Compare & Pull Request**.

Select **base: main** and **compare: feature-branch**.

Create **Pull Request** → Merge.







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

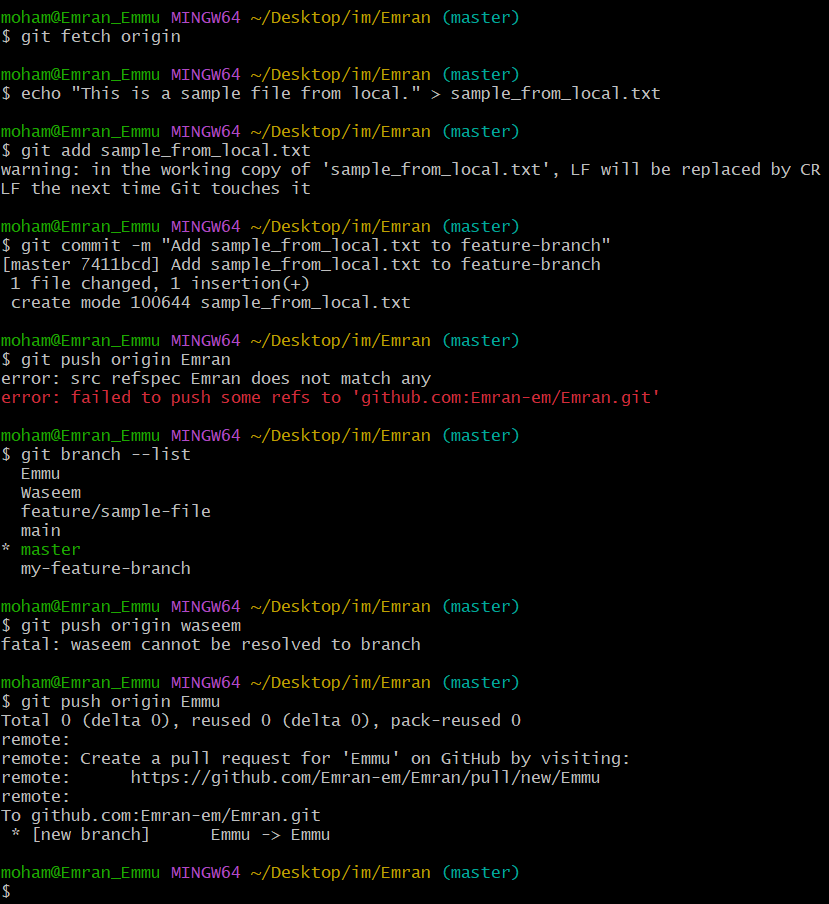
git checkout feature-branch

touch newfile.txt

git add newfile.txt

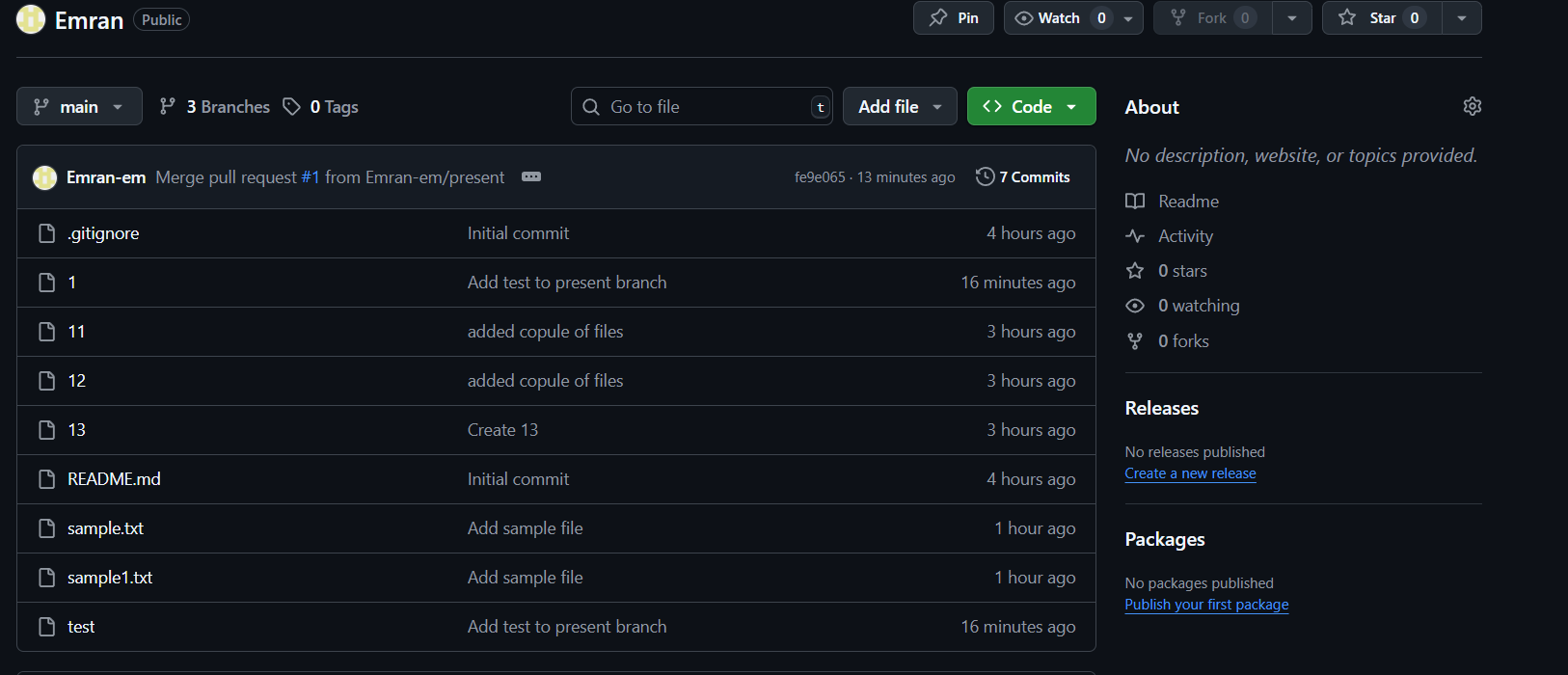
git commit -m "Added new file in feature-branch"

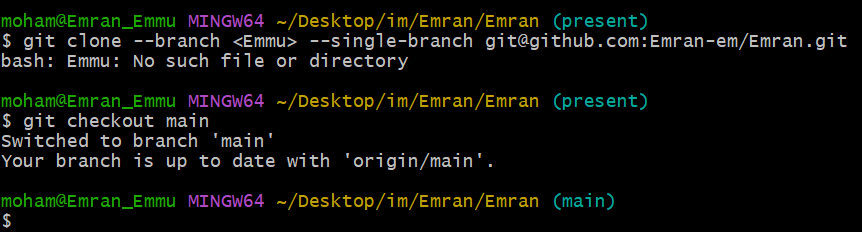
git push origin feature-branch



11)clone only a branch from github to local.

git clone -b branch-name --single-branch https://github.com/<username>/<repo>.git

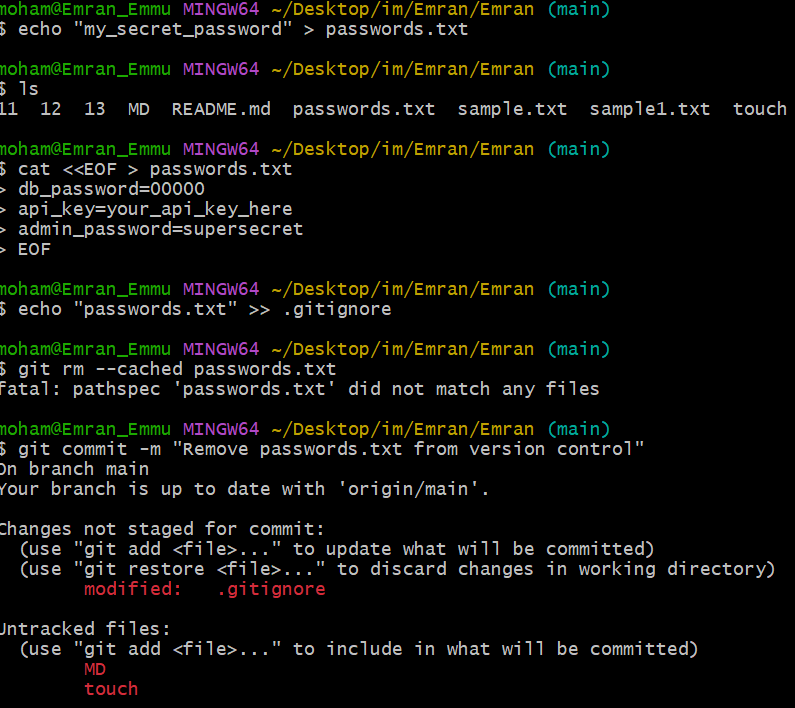




12)create a file with all passwords and make that untrackable with git.

touch passwords.txt

echo "passwords.txt" >> .gitignore



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

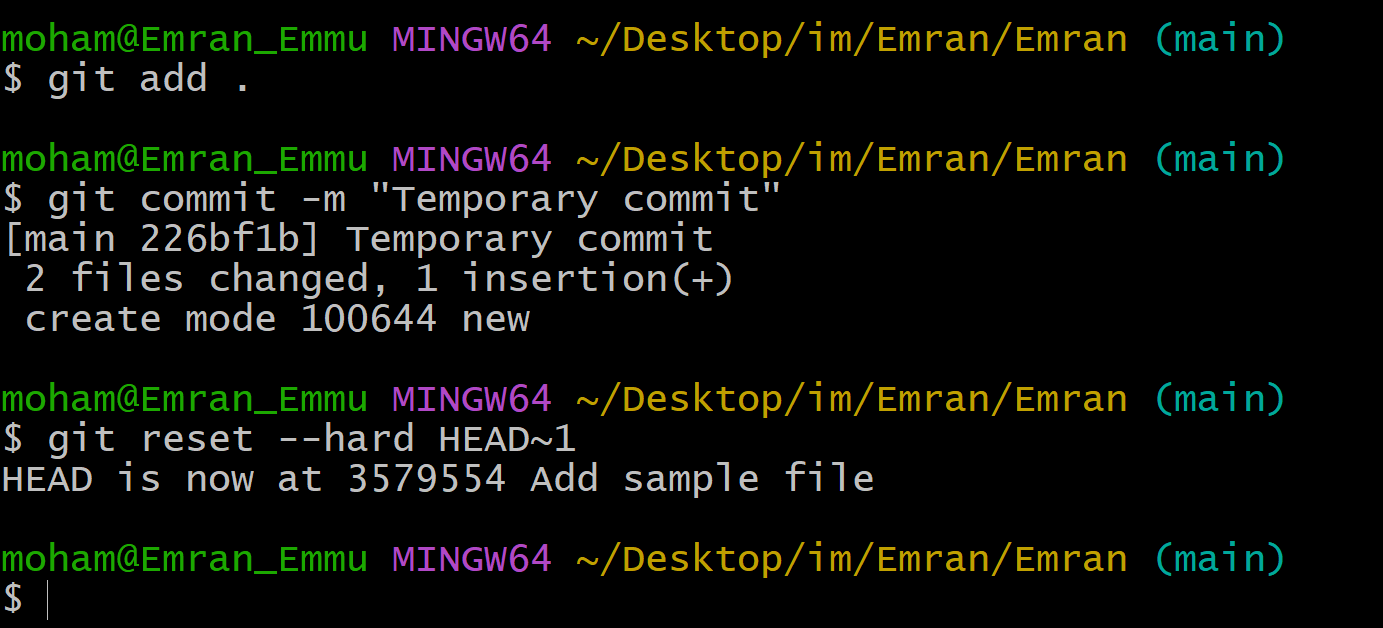
touch temp.txt

git add temp.txt

git commit -m "Temporary file"

* Reset and discard changes:

git reset --hard HEAD~1

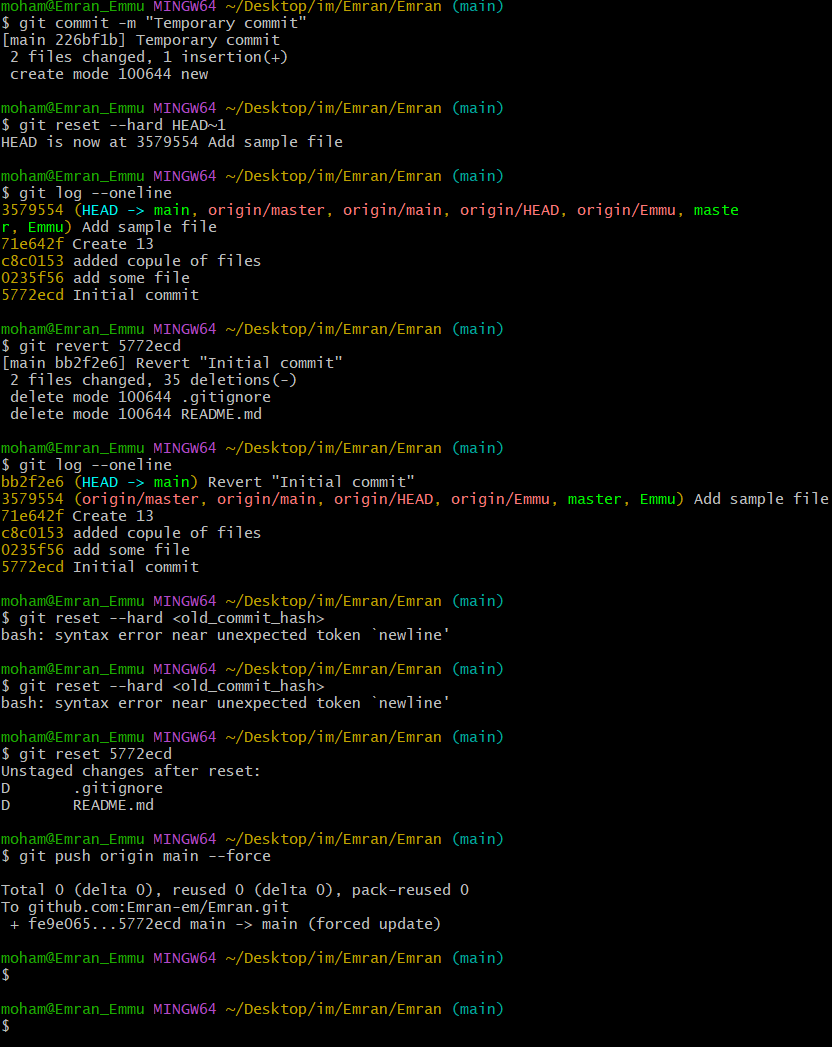


14)Revert a commited commit to the older version.

git log # Copy commit hash

git revert <commit-hash>

git push origin main



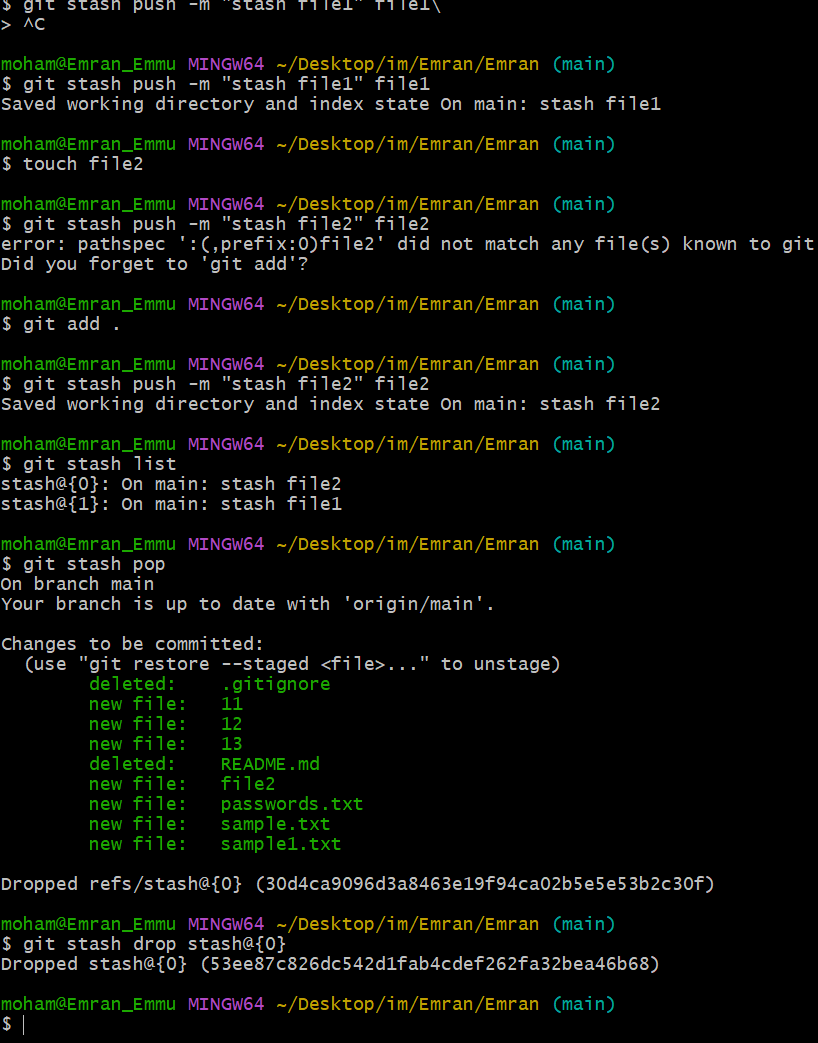
15)push a file to stash without savings the changes and work on another file.

# Edit file1.txt but don’t commit

git stash

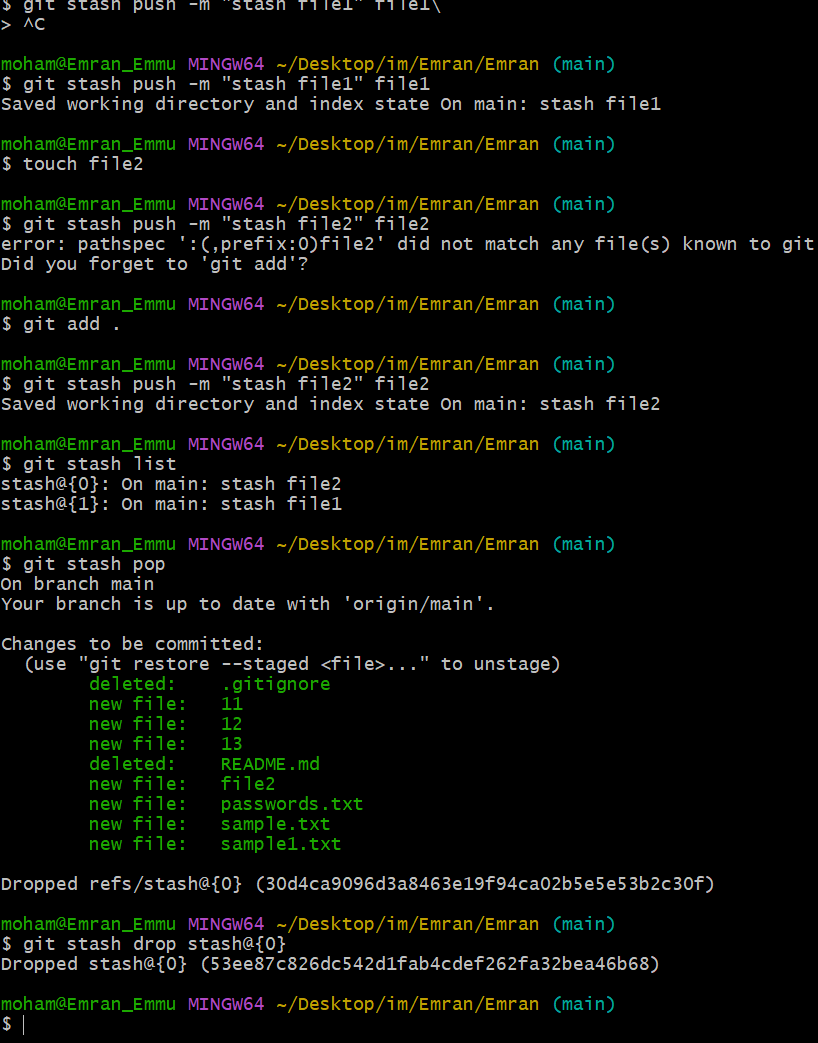
# Work on another file

touch another.txt



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

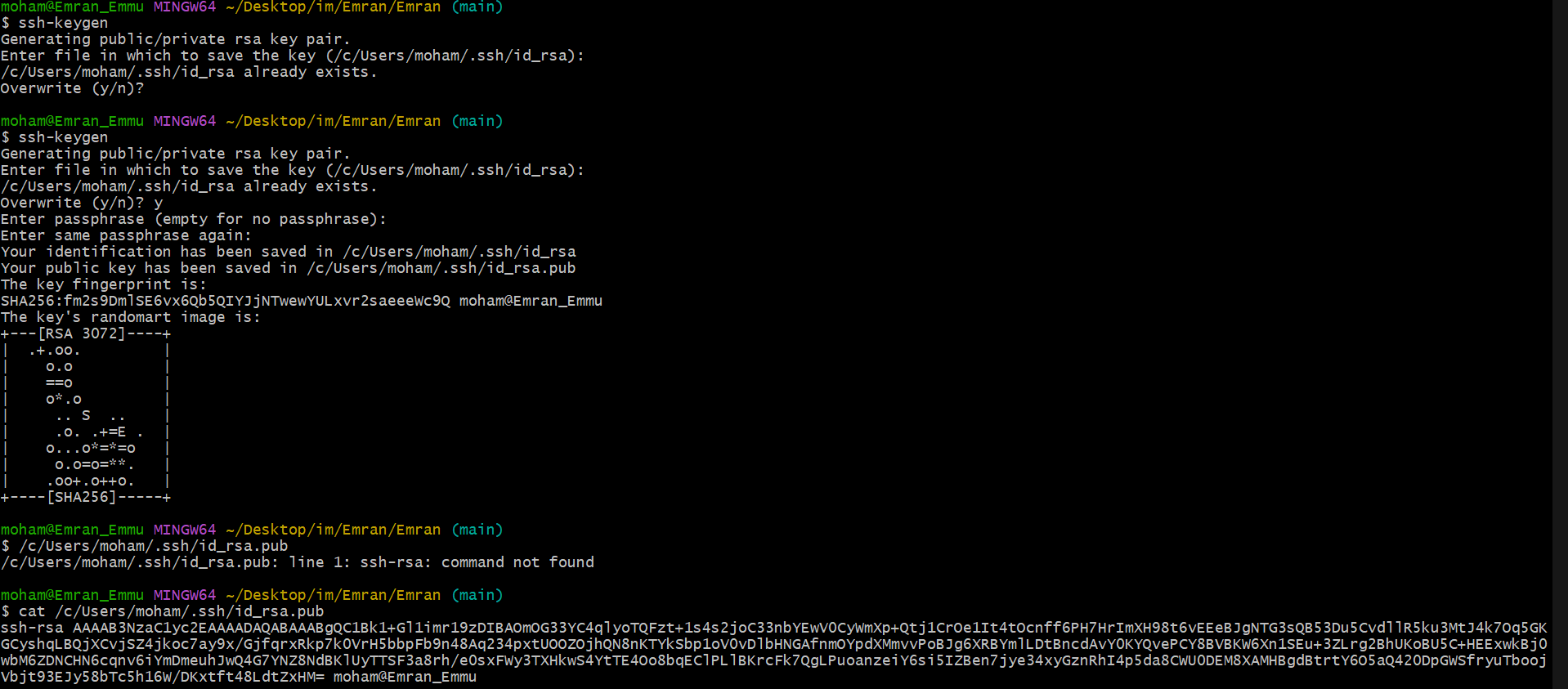
git stash pop

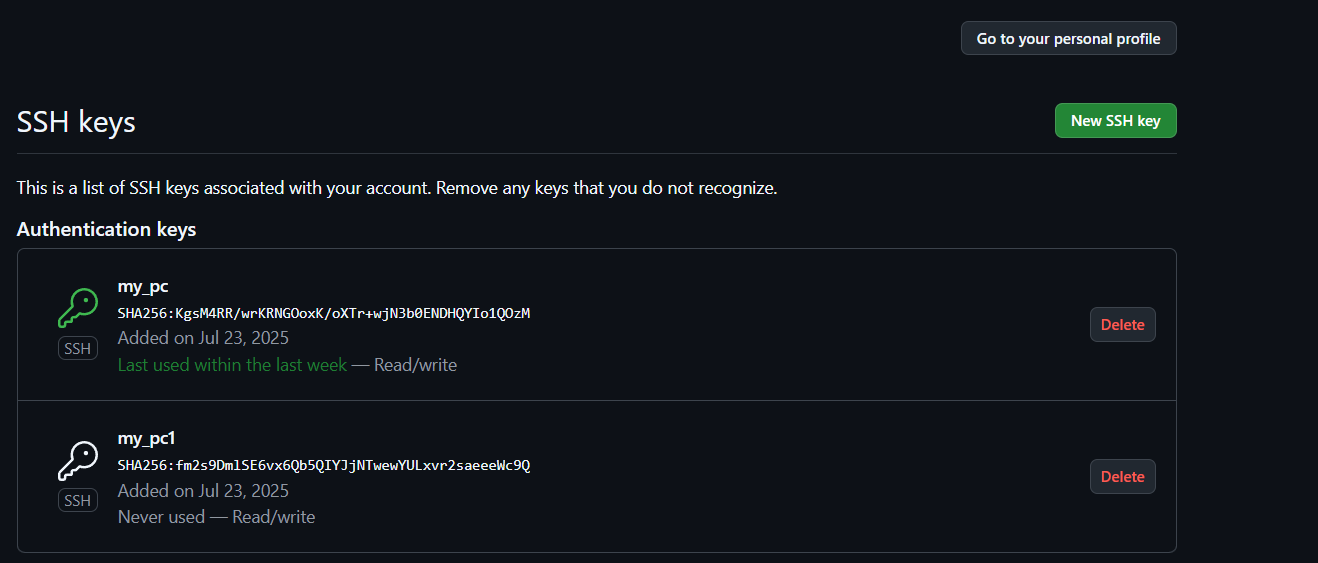


17)generate a ssh-keygen and configure into github.

ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

cat ~/.ssh/id\_rsa.pub





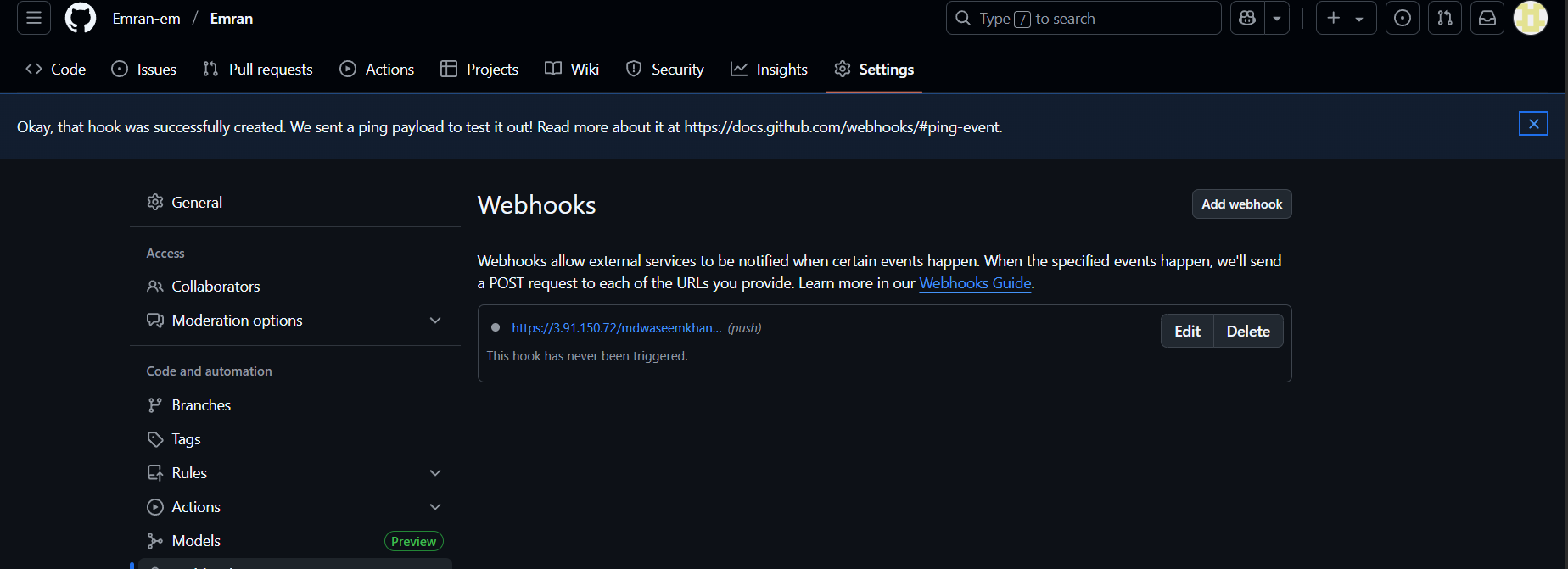
18)configure webhooks to github

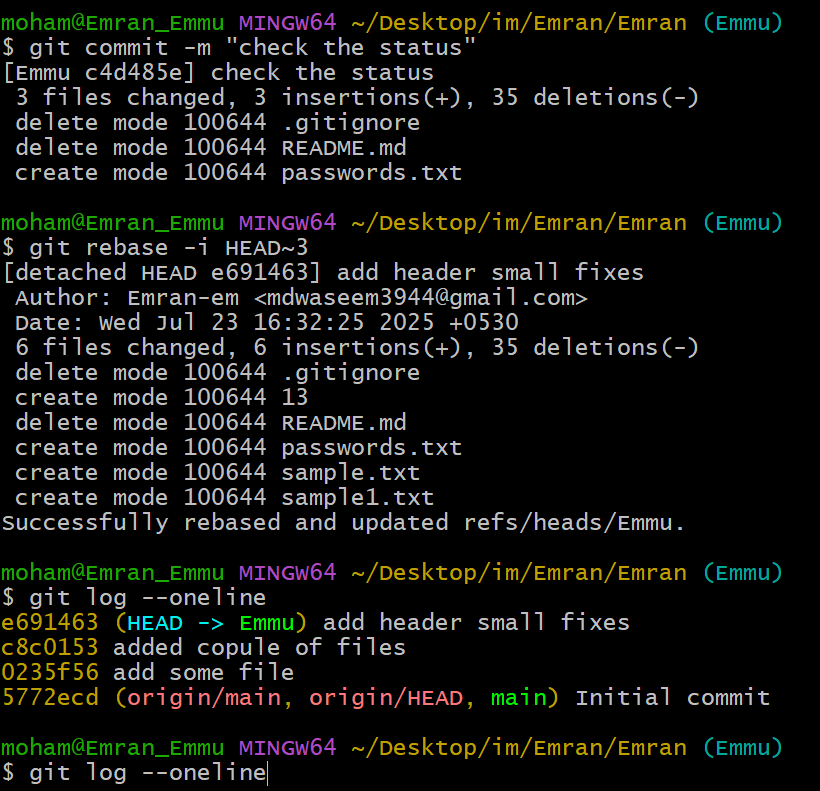
Go to **GitHub repo → Settings → Webhooks → Add Webhook**.

Provide:

* Payload URL: your server endpoint (e.g., Jenkins URL).
* Content type: application/json.

Select events → Add Webhook.

.



19)basic understanding of .git file.

The .git folder in a repo stores:

* **Objects** (commits, blobs, trees)
* **Refs** (branches, tags)
* **Config** (user info, remotes)
* **HEAD** (current branch)

Git is a distribution version control system widely used for tracking changes sources code during software development .it allow multiple developers to collaborate on the projects by providing tools for managing code history,branches,merging and more

.git file is a directory is like a database that store all the information about git repo, including … it’s a crucial part of git that helps your track changes, collaborate with other, and manages your codebasae

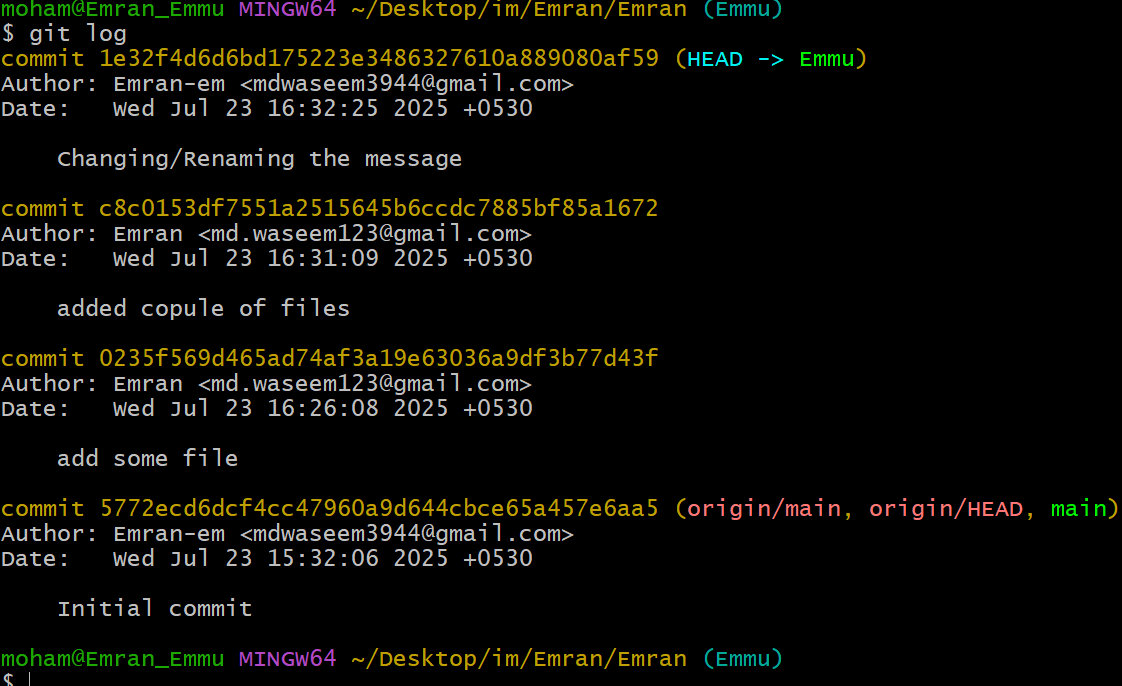
* Your commit history
* Branch information
* Remote repo details
* Configuration settings

ls .git

20)Check all the logs of git.

git log

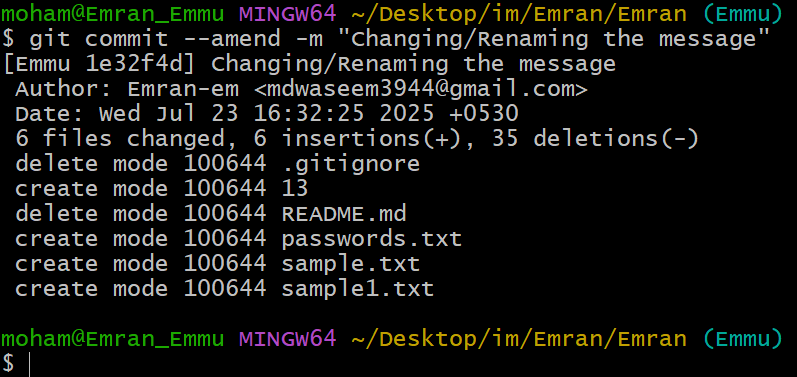
git log --oneline --graph --decorate --all



21)Rename the commit message.

git commit --amend -m "New commit message"

git push origin main --force



22)Merge multiple commits into single commit.

git rebase -i HEAD~3

# In editor, change 'pick' to 'squash' for commits to combine

# Save and exit

git push origin main --force

