

git lab program - 5

collaboration and Remote Repositories

Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.

Remote repository - Repository stored on github
ex: origin (default remote name in git)

Local Repository - The copy on your computer.

Fetch - git fetch - Download the latest changes commits from the github

Rebase - take your changes and place them on top those new updates.

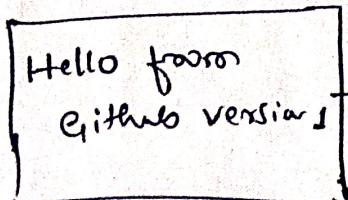
If someone changes something on github, you can fetch to see those changes and rebase your own changes on top of them to keep your version up-to-date.

Step 1: Create a new repository on github

ex: gitlab-rebase-demos

Step 2: Add one file in github

click add file → Create new file → commit



click commit

Step 3: clone it your computer

cd Desktop

mkdir gitlab5

cd gitlab5

git clone https://github.com/gayathrih/gitlab-rebase-demo.git

cd gitlab5-rebase-demo

Step 4: open it and edit the file locally



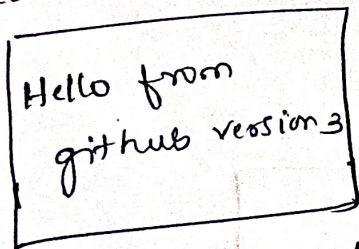
→ git add.

git commit -m "add version"

Step 5: simulate a change on Github

→ open demo.txt

edit the file add one more line.



→ Then commit

Now the remote repo has new changes that your local repo doesn't know about yet.

Step 6: Fetch the latest changes from github

In gitbash run

git fetch origin

This downloads the new Github commit but doesn't apply it yet.

If you want to see what changes

git log origin/main --oneline

Step 7: Rebase your local branch
get rebase origin/main

This tells git → Apply my local commits after
the latest GitHub commit
if no conflicts occur, rebase finishes successfully

→ If there are conflicts occur
Fix the conflicts in the file manually

~~==== Head~~
~~lines(s) from remote (origin/main)~~ | remove the
~~=====~~ | ~~=====, =====,~~
~~lines(s) from your local commits~~ | >>> marker
~~>>> (comm - len)~~ | save file

→ add the file git add demo.txt
→ continue the rebase git rebase --continue

Step 8: Verify

cat demo.txt

```
Hello from your version 1
Hello from GitHub version 3
Hello from GitHub version 2
```

This means both your local and remote changes
are now together

Step 9: Push back to GitHub

Finally push your updated work

git push origin main --force

needed bcz we changed history and rebased