Assignment 1: Branch Operations

Description

What is a Squash Commit in Git?

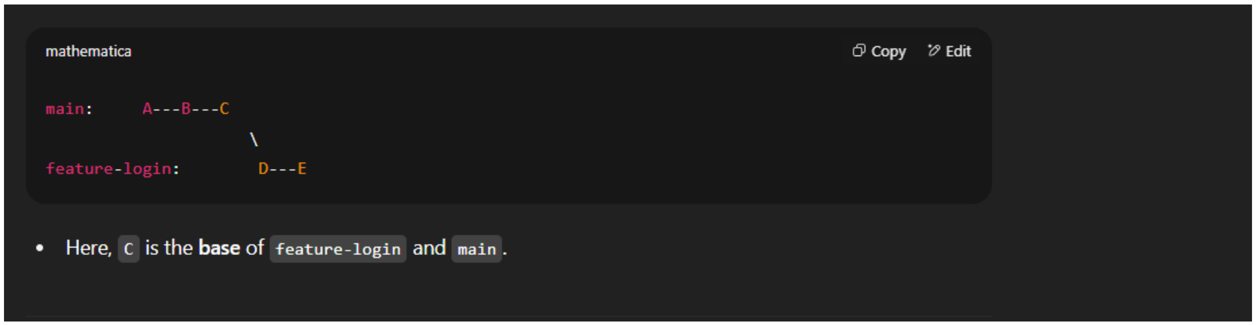
Squashing commits means **combining multiple commits into a single one**.

What ia base?

In Git, the **base** is the common ancestor commit from which a feature branch and another branch (usually main or develop) diverge.

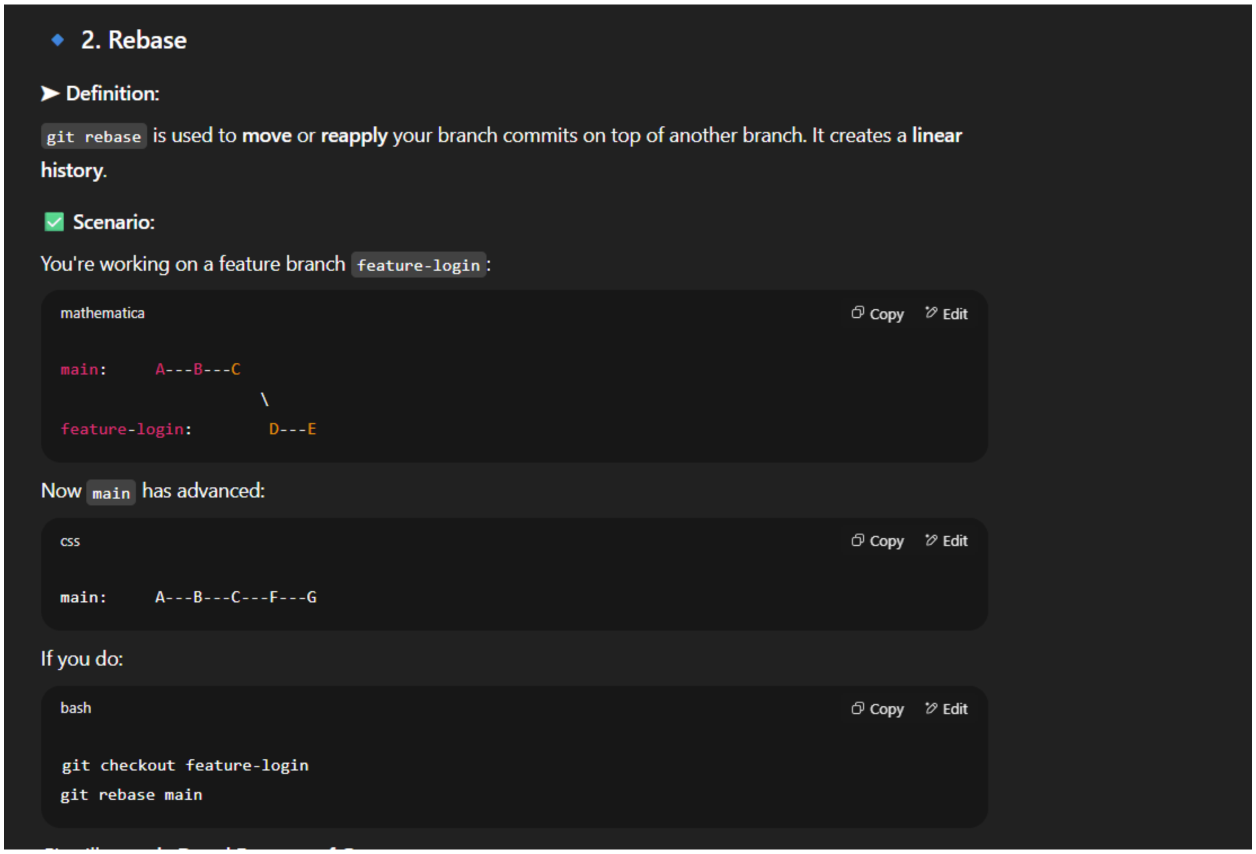
✓ **Real-world Scenario:**

- You have a main branch main.
- You create a new feature branch feature-login from main.
- Later, you compare feature-login with main to merge — Git finds the **base** (the commit where they split) to determine the changes.



```
main:     A---B---C
                   \
feature-login:      D---E
```

- Here, `C` is the **base** of `feature-login` and `main`.



🔹 2. Rebase

➤ Definition:

`git rebase` is used to **move** or **reapply** your branch commits on top of another branch. It creates a **linear history**.

✅ **Scenario:**

You're working on a feature branch `feature-login`:

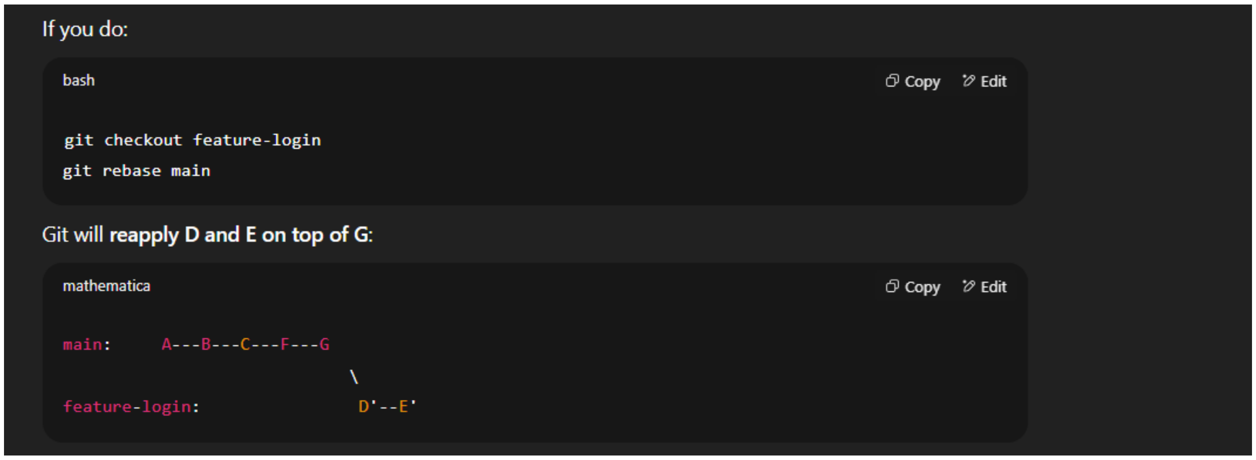
```mathematica
main:      A---B---C
                    \
feature-login:        D---E
```

Now `main` has advanced:

```css
main:      A---B---C---F---G
```

If you do:

```bash
git checkout feature-login
git rebase main
```

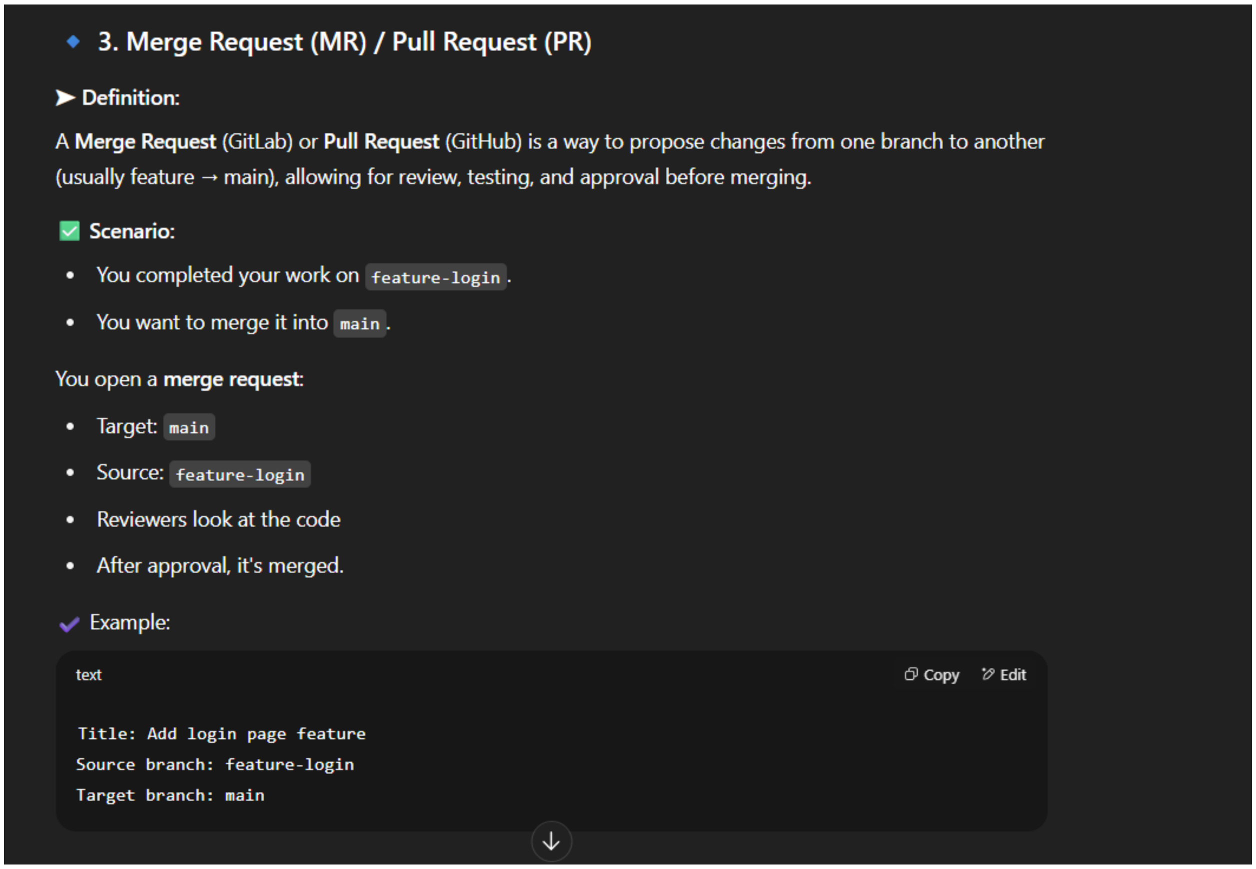


If you do:

```bash
git checkout feature-login
git rebase main
```

Git will **reapply D and E on top of G**:

```mathematica
main:      A---B---C---F---G
                            \
feature-login:                D'--E'
```


3. Merge Request (MR) / Pull Request (PR)

Definition:

A **Merge Request** (GitLab) or **Pull Request** (GitHub) is a way to propose changes from one branch to another (usually feature → main), allowing for review, testing, and approval before merging.

✅ **Scenario:**

- You completed your work on `feature-login`.
- You want to merge it into `main`.

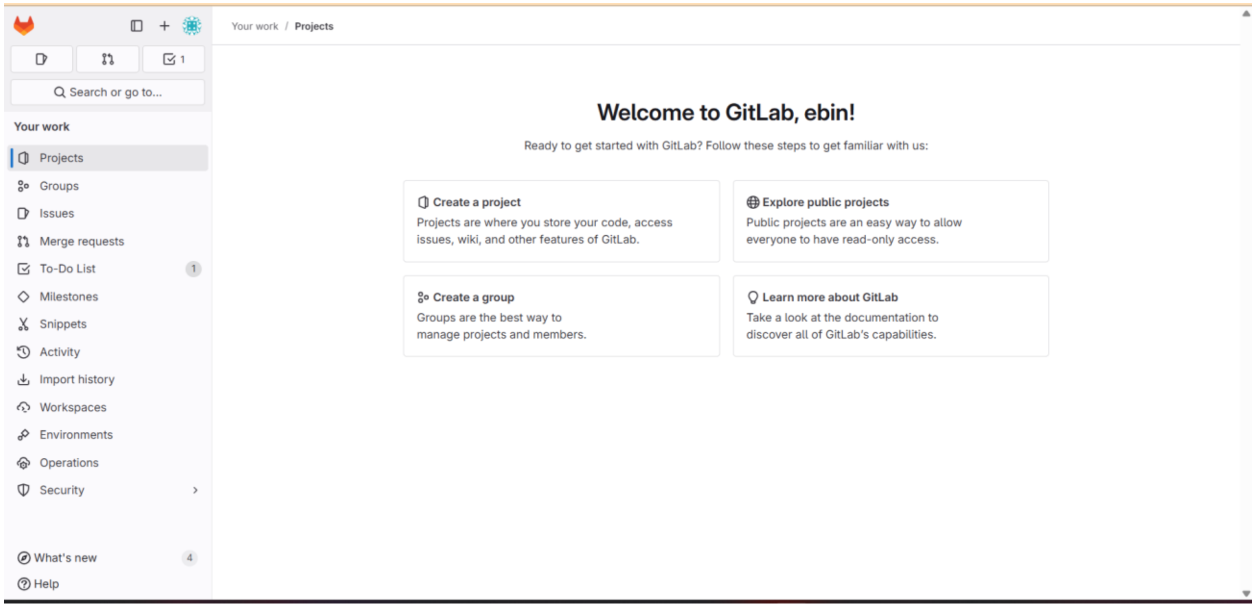
You open a **merge request**:

- Target: `main`
- Source: `feature-login`
- Reviewers look at the code
- After approval, it's merged.

✔ Example:

```text
Title: Add login page feature
Source branch: feature-login
Target branch: main
```

Step1 : I am going to create a new project



Welcome to GitLab, ebin!

Ready to get started with GitLab? Follow these steps to get familiar with us:

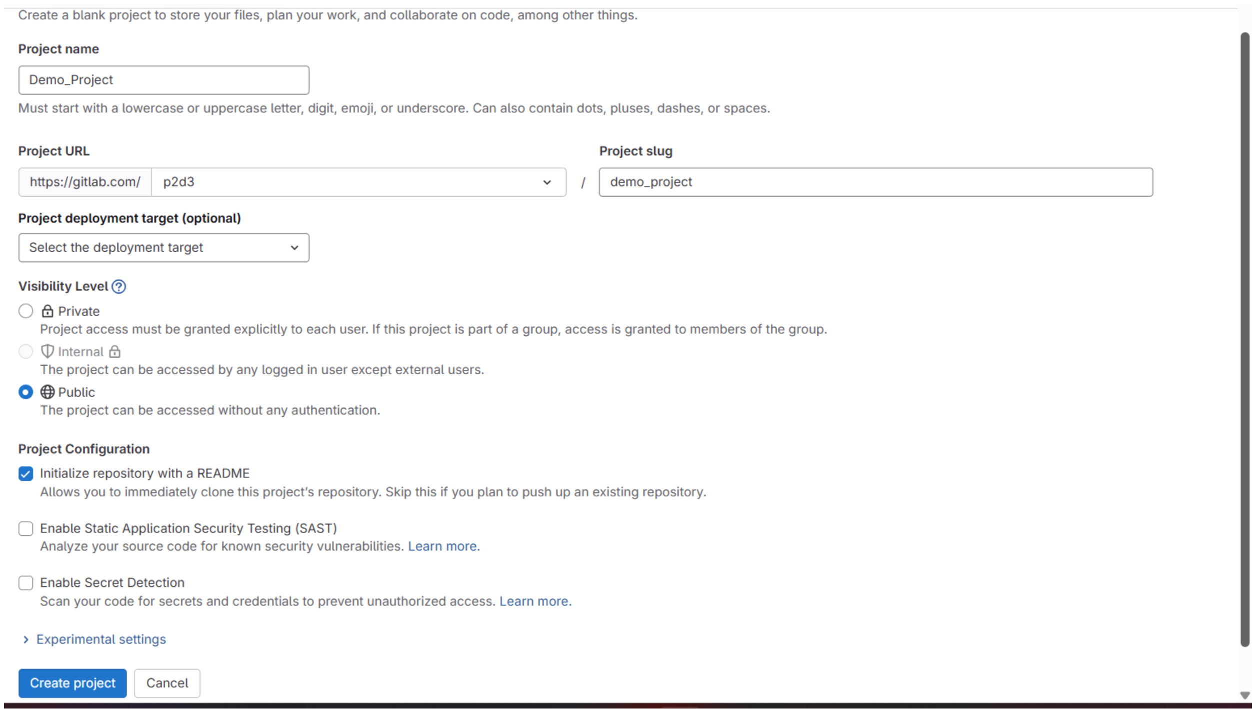
Create a project
Projects are where you store your code, access issues, wiki, and other features of GitLab.

Explore public projects
Public projects are an easy way to allow everyone to have read-only access.

Create a group
Groups are the best way to manage projects and members.

Learn more about GitLab
Take a look at the documentation to discover all of GitLab's capabilities.

Step2: Enter projectname and make it as public



Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name

Demo_Project

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL: https://gitlab.com/ p2d3 / **Project slug**: demo_project

Project deployment target (optional): Select the deployment target

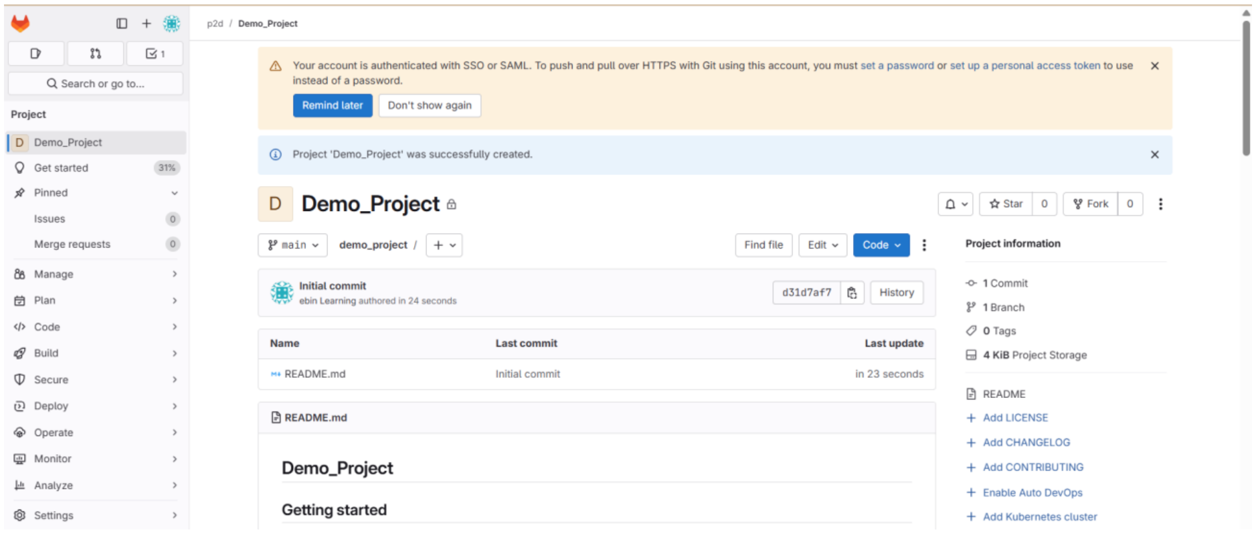
Visibility Level

- [ ] Private — Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.
- [ ] Internal — The project can be accessed by any logged in user except external users.
- [✓] Public — The project can be accessed without any authentication.

Project Configuration

- [✓] Initialize repository with a README — Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.
- [ ] Enable Static Application Security Testing (SAST) — Analyze your source code for known security vulnerabilities. Learn more.
- [ ] Enable Secret Detection — Scan your code for secrets and credentials to prevent unauthorized access. Learn more.

Step3 : After clicking 'Create Project', the project was successfully created in GitLab.



Your account is authenticated with SSO or SAML. To push and pull over HTTPS with Git using this account, you must set a password or set up a personal access token to use instead of a password.

Project 'Demo_Project' was successfully created.

Demo_Project

Initial commit — ebin Learning authored in 24 seconds — d31d7af7

Name	Last commit	Last update
README.md	Initial commit	in 23 seconds

README.md

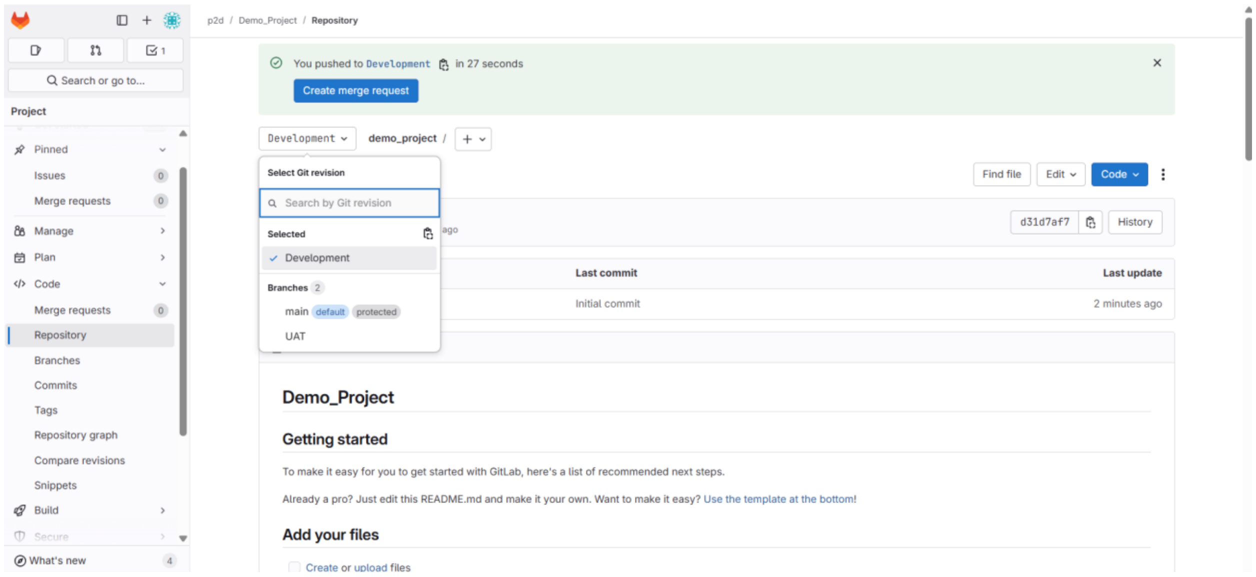
Demo_Project

Getting started

Project information: 1 Commit, 1 Branch, 0 Tags, 4 KiB Project Storage

README, Add LICENSE, Add CHANGELOG, Add CONTRIBUTING, Enable Auto DevOps, Add Kubernetes cluster

Step4: Since only the main branch exists, I'll create the additional branches development and UAT.



You pushed to `Development` in 27 seconds

Demo_Project

Getting started

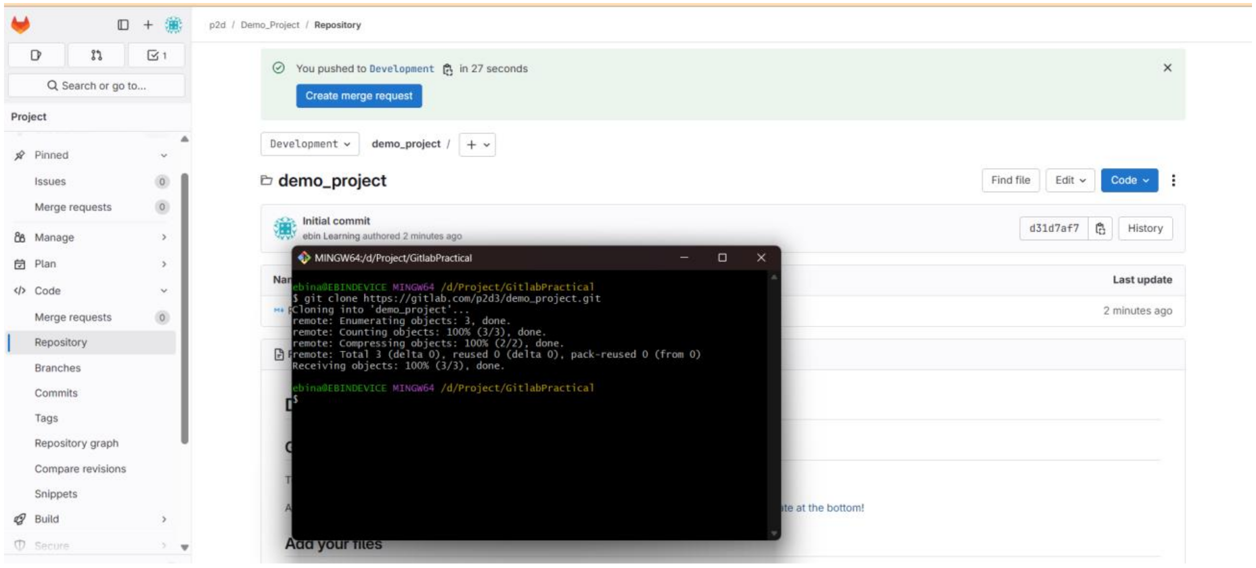
To make it easy for you to get started with GitLab, here's a list of recommended next steps.

Already a pro? Just edit this README.md and make it your own. Want to make it easy? Use the template at the bottom!

Add your files

Create or upload files

Step 5: I am going to clone the code from the main branch.

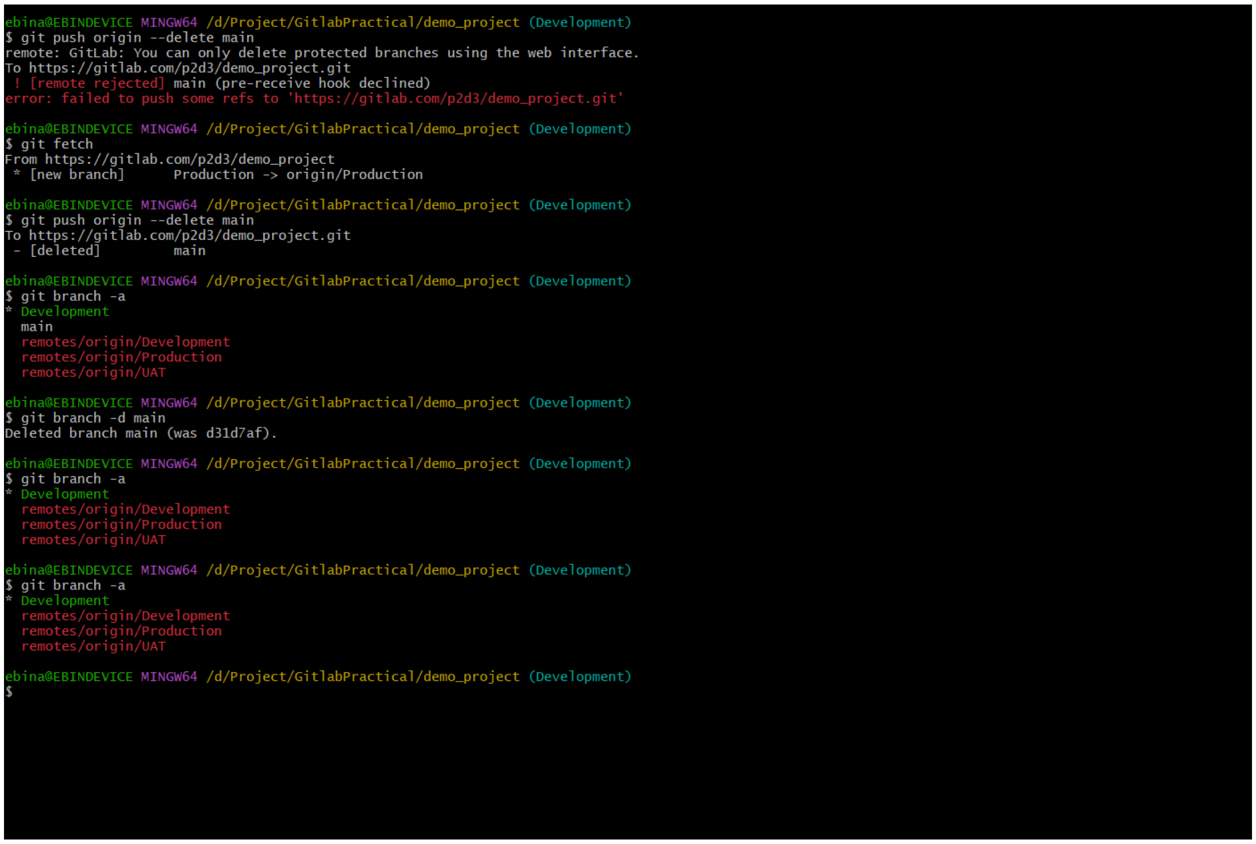


You pushed to `Development` in 27 seconds

Initial commit
ebin Learning authored 2 minutes ago

```
ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical
$ git clone https://gitlab.com/p2d3/demo_project.git
Cloning into 'demo_project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical
$
```

Step6: I'm going to list all branches, switch from main to development, and remove the main branch

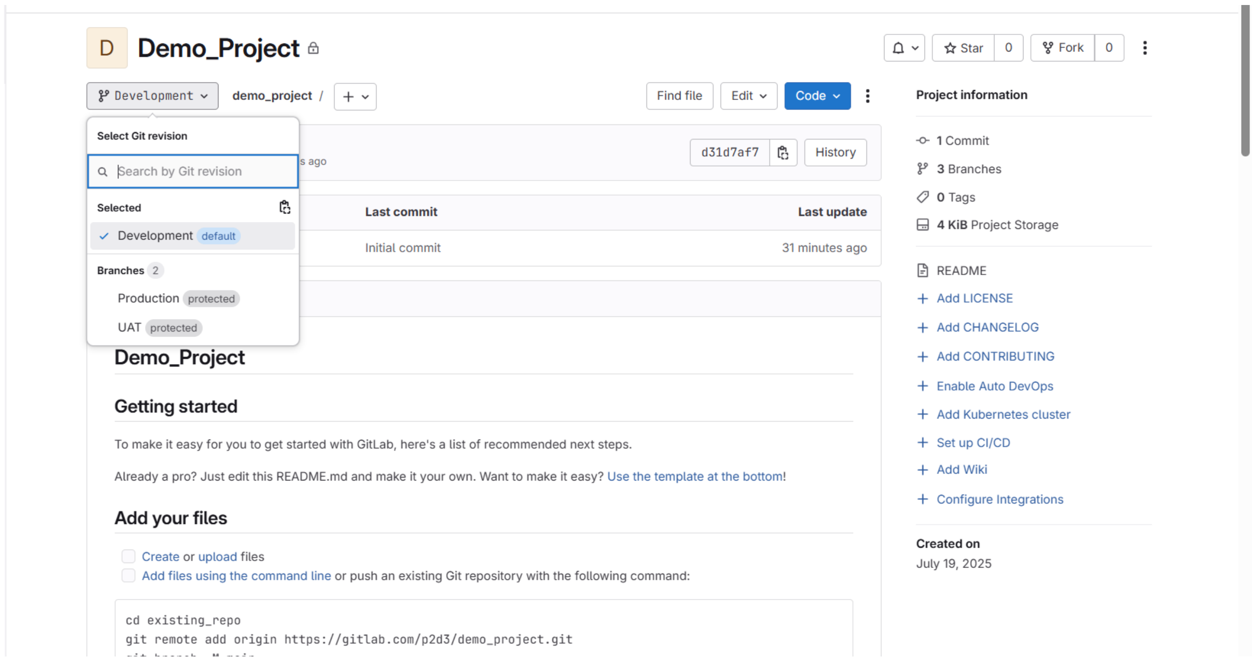


```
ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git push origin --delete main
remote: GitLab: You can only delete protected branches using the web interface.
To https://gitlab.com/p2d3/demo_project.git
 ! [remote rejected] main (pre-receive hook declined)
error: failed to push some refs to 'https://gitlab.com/p2d3/demo_project.git'

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git fetch
From https://gitlab.com/p2d3/demo_project
 * [new branch]      Production -> origin/Production

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git push origin --delete main
To https://gitlab.com/p2d3/demo_project.git
 - [deleted]         main

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git branch -a
* Development
  main
  remotes/origin/Development
  remotes/origin/Production
  remotes/origin/UAT

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git branch -d main
Deleted branch main (was d31d7af).

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git branch -a
* Development
  remotes/origin/Development
  remotes/origin/Production
  remotes/origin/UAT

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git branch -a
* Development
  remotes/origin/Development
  remotes/origin/Production
  remotes/origin/UAT

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$
```

Command

Git push origin –delete main

Git -d main



Demo_Project

Getting started

To make it easy for you to get started with GitLab, here's a list of recommended next steps.

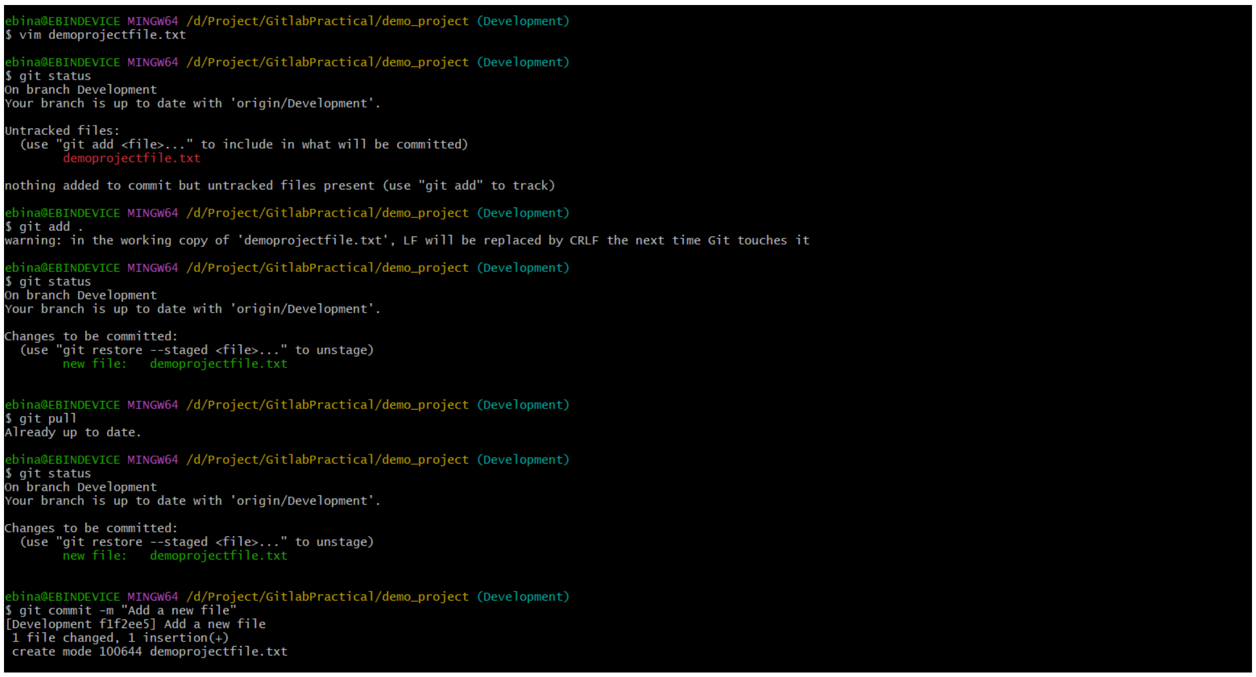
Already a pro? Just edit this README.md and make it your own. Want to make it easy? Use the template at the bottom!

Add your files

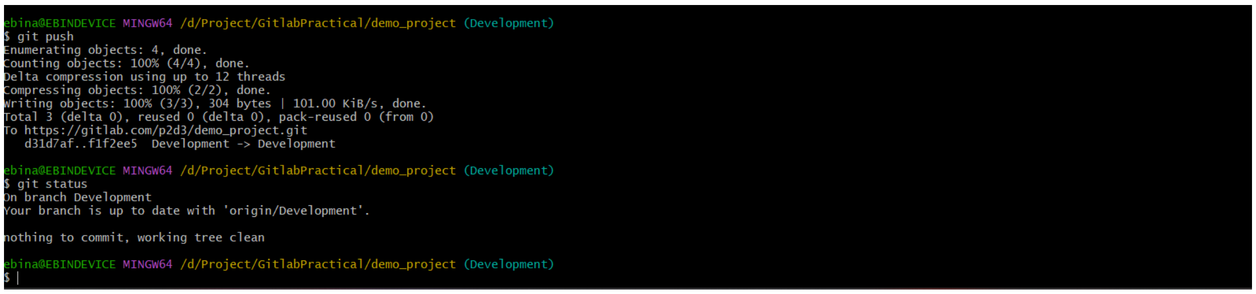
- [ ] Create or upload files
- [ ] Add files using the command line or push an existing Git repository with the following command:

```
cd existing_repo
git remote add origin https://gitlab.com/p2d3/demo_project.git
```

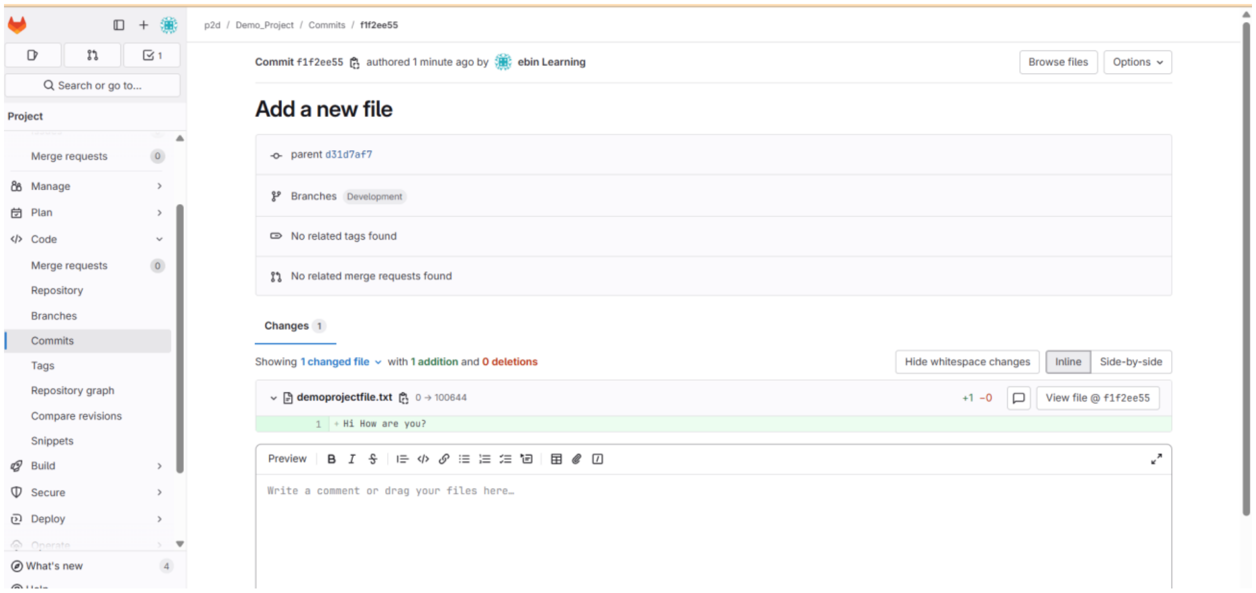
Step 7: I will create a new file and perform fetch, commit, pull, and push operations.



```
ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ vim demoprojectfile.txt

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git status
On branch Development
Your branch is up to date with 'origin/Development'.

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

nothing added to commit but untracked files present (use "git add" to track)

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git add .
warning: in the working copy of 'demoprojectfile.txt', LF will be replaced by CRLF the next time Git touches it

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git status
On branch Development
Your branch is up to date with 'origin/Development'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   demoprojectfile.txt


ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git pull
Already up to date.

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git status
On branch Development
Your branch is up to date with 'origin/Development'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   demoprojectfile.txt


ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git commit -m "Add a new file"
[Development f1f2ee5] Add a new file
 1 file changed, 1 insertion(+)
 create mode 100644 demoprojectfile.txt
```



```
ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 304 bytes | 101.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://gitlab.com/p2d3/demo_project.git
   d31d7af..f1f2ee5  Development -> Development

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git status
On branch Development
Your branch is up to date with 'origin/Development'.

nothing to commit, working tree clean

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ |
```



Commit f1f2ee55 authored 1 minute ago by ebin Learning

Add a new file

parent d31d7af7

Branches Development

No related tags found

No related merge requests found

Changes 1

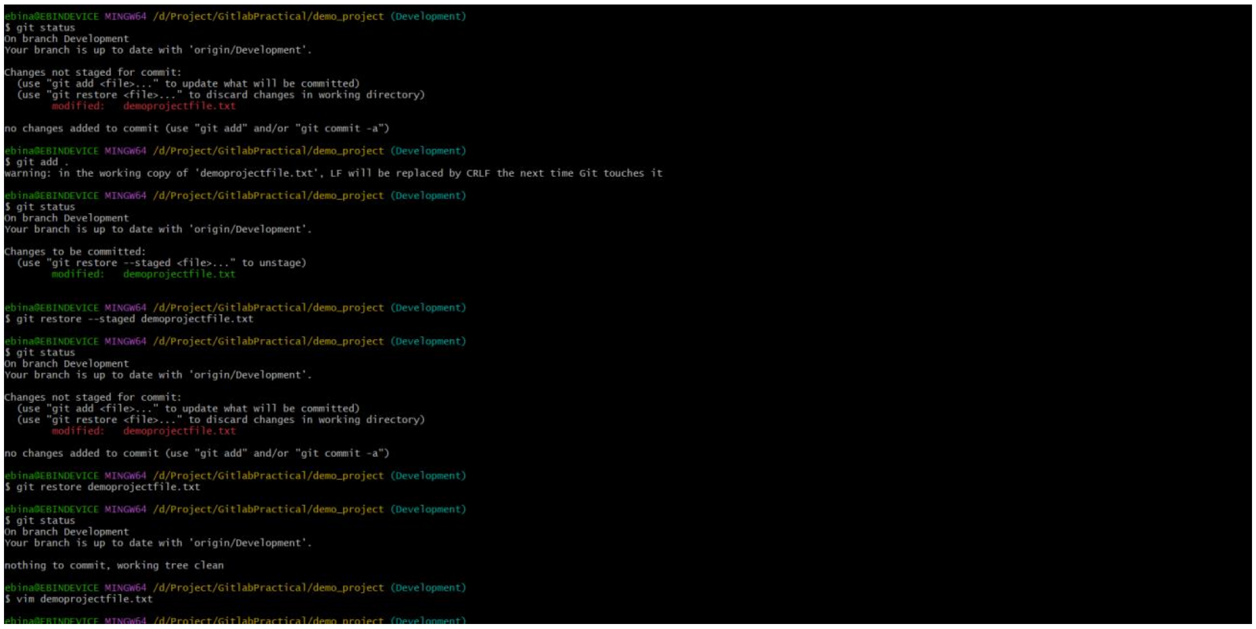
Showing 1 changed file with 1 addition and 0 deletions

demoprojectfile.txt 0 → 100644 +1 −0

```
1 + Hi How are you?
```

Write a comment or drag your files here…

Step 8: Revert commit



```
ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git status
On branch Development
Your branch is up to date with 'origin/Development'.

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:   demoprojectfile.txt

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

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git add .
warning: in the working copy of 'demoprojectfile.txt', LF will be replaced by CRLF the next time Git touches it

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git status
On branch Development
Your branch is up to date with 'origin/Development'.

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


ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git restore --staged demoprojectfile.txt

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git status
On branch Development
Your branch is up to date with 'origin/Development'.

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:   demoprojectfile.txt

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

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git restore demoprojectfile.txt

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git status
On branch Development
Your branch is up to date with 'origin/Development'.

nothing to commit, working tree clean

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ vim demoprojectfile.txt

ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
```