

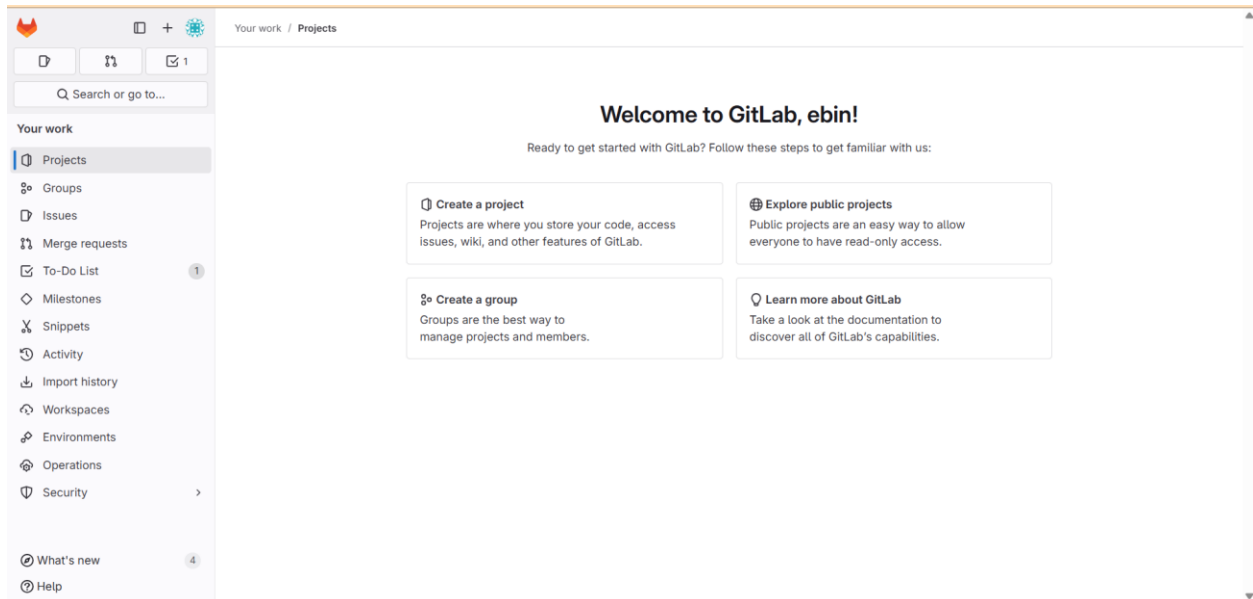
Assignment 1: Branch Operations

Description

What is a Squash Commit in Git?

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

Step1 : I am going to create a new project



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.](#)

> Experimental settings

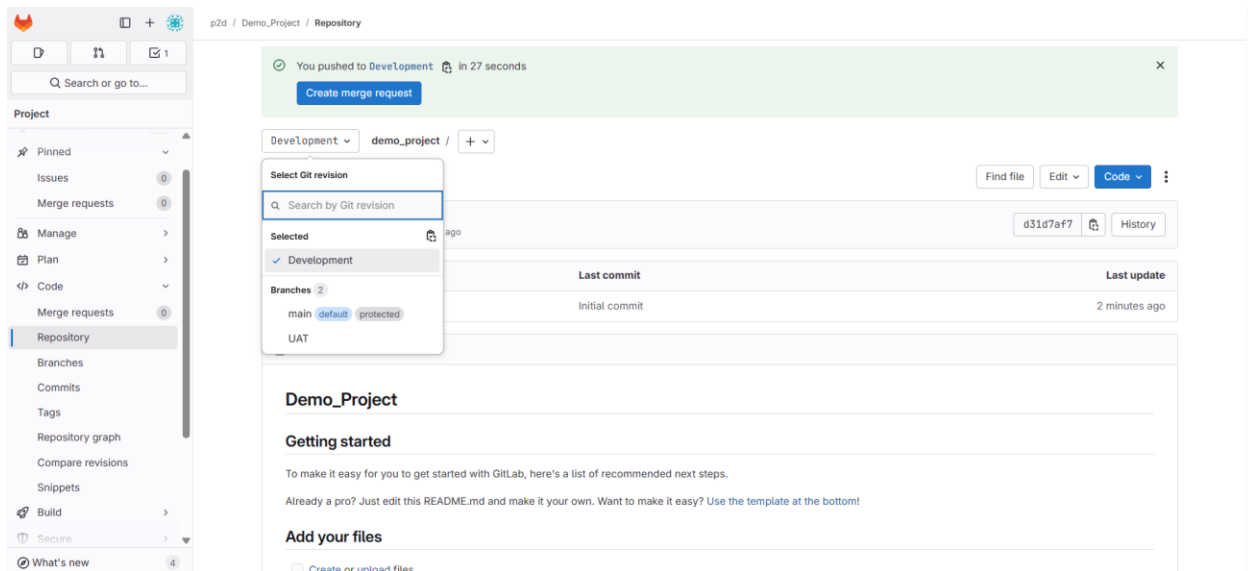
Create project

Cancel

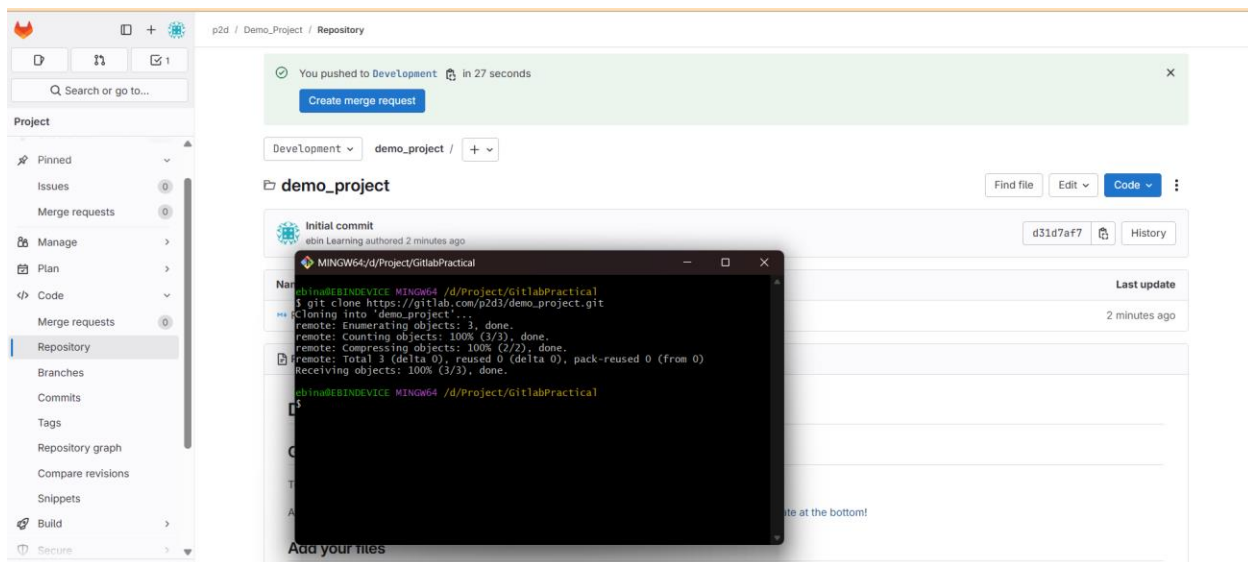
Step3 : After clicking 'Create Project', the project was successfully created in GitLab.

The screenshot shows the GitLab interface for a newly created project named 'Demo_Project'. The left sidebar contains navigation links: Project, Get started (31%), Pinned, Issues, Merge requests, Manage, Plan, Code, Build, Secure, Deploy, Operate, Monitor, Analyze, and Settings. The main content area displays a success message: 'Project "Demo_Project" was successfully created.' Below this, the project details are shown, including the initial commit 'Initial commit' with a commit hash 'd31d7af7' and a timestamp 'in 23 seconds'. A table lists the files in the repository, showing 'README.md' as the only file. The right sidebar provides project information, including 1 commit, 1 branch, 0 tags, and 4 KIB Project Storage. It also offers links to add a LICENSE, CHANGELOG, CONTRIBUTING file, and enable Auto DevOps and Kubernetes cluster.

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



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



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

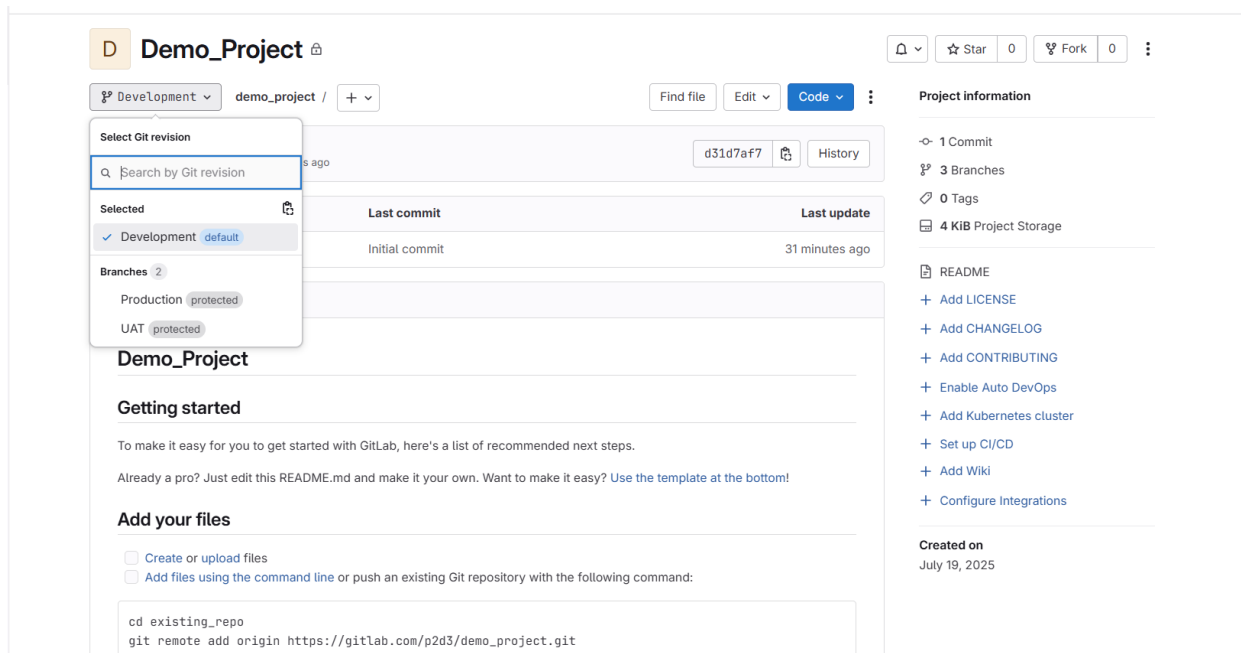
ebina@EBINDEVICE MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git branch -a
* Development
  remotes/origin/Development
  remotes/origin/Production
  remotes/origin/UAT

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

Command

Git push origin --delete main

Git -d main



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

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

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

ebina@EBINDEVIC 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

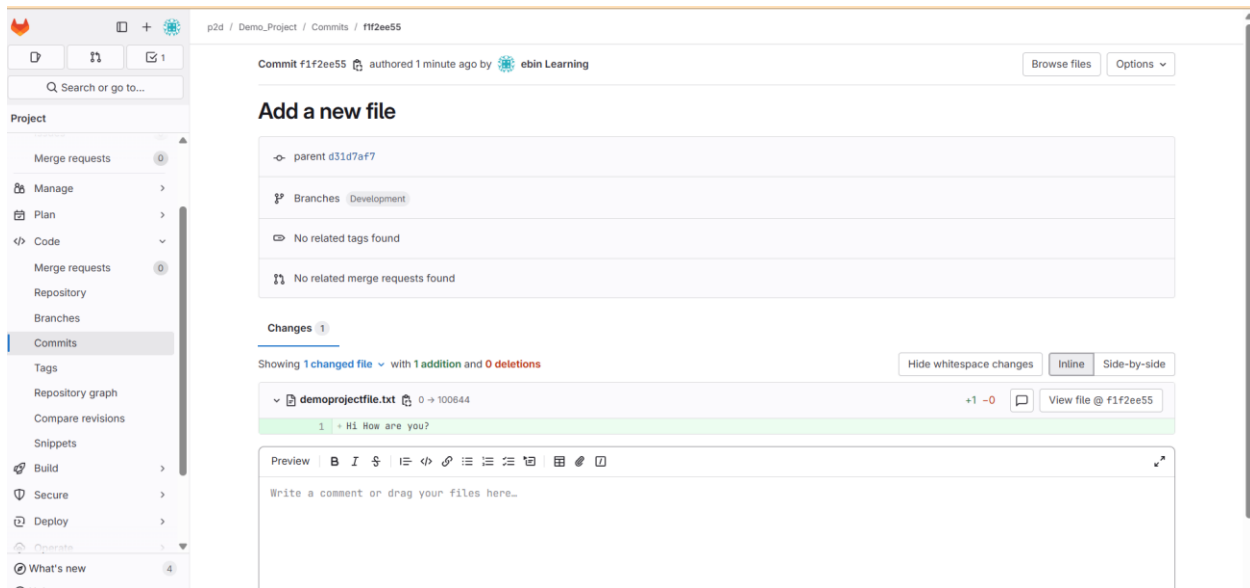
ebina@EBINDEVIC MINGW64 /d/Project/GitlabPractical/demo_project (Development)
$ git commit -m "Add a new file"
[Development f1f2ee5] Add a new file
1 file changed, 1 insertion(+)
create mode 100644 demoprojectfile.txt
```

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



Step 8: Revert commit

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

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

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

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