

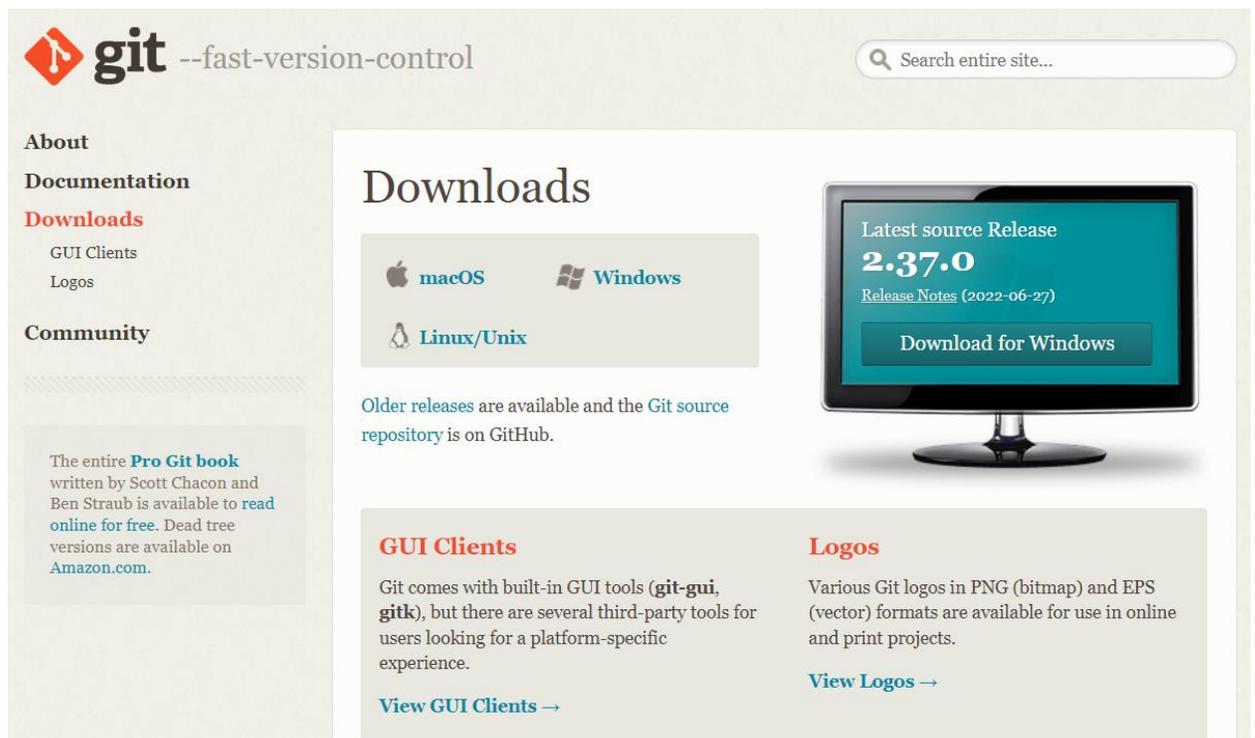
Lab: Các thao tác cơ bản trên Git



I. Version Control

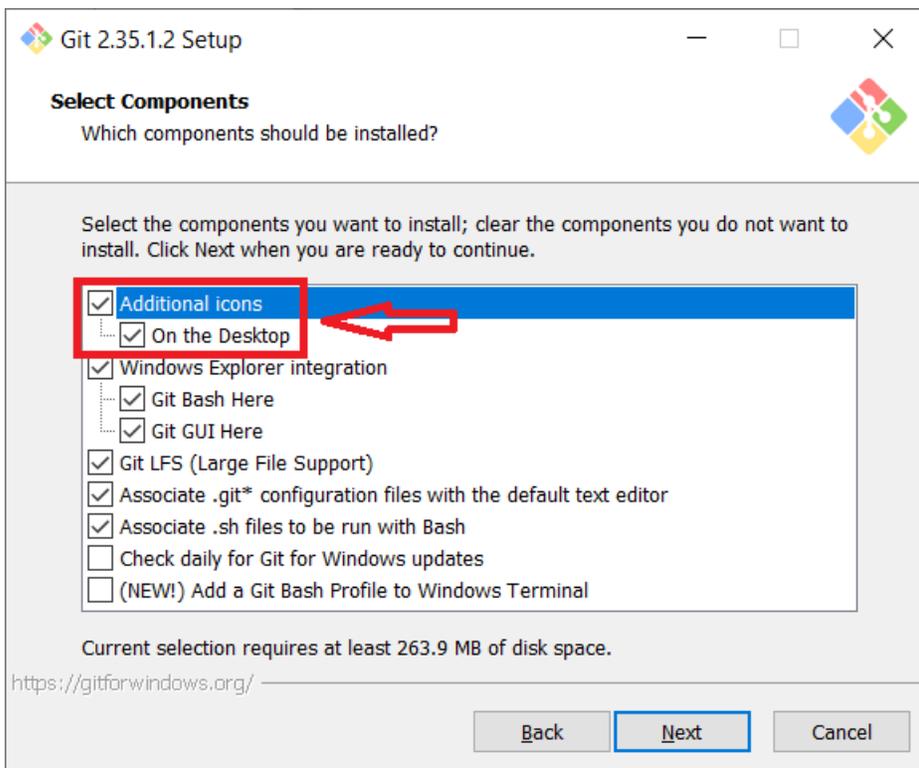
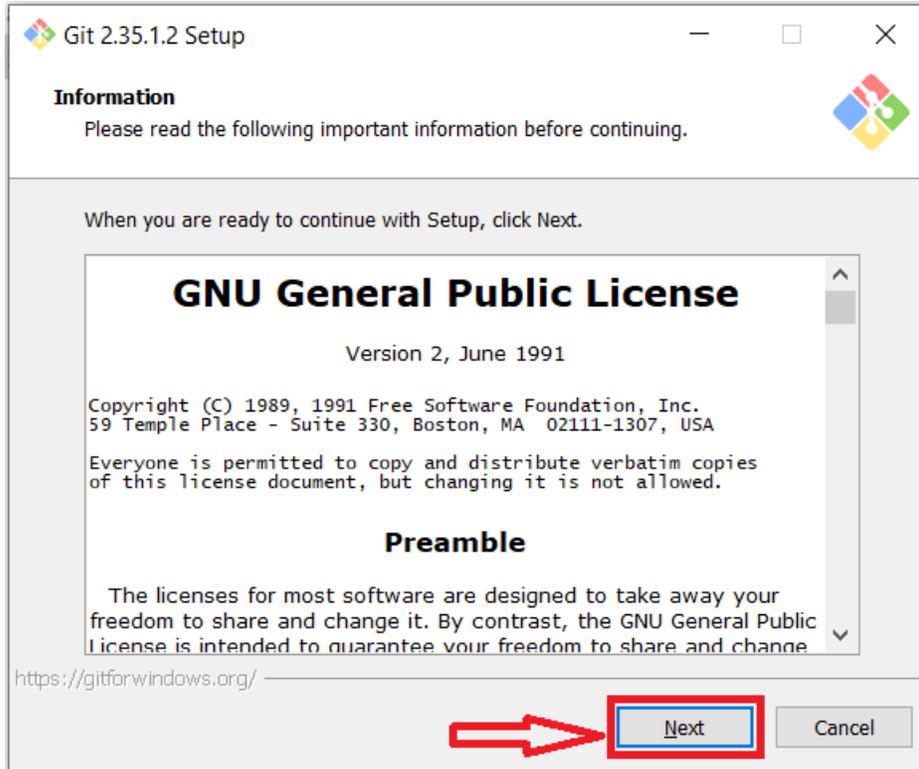
Cài đặt Git cho máy tính, ở đây sử dụng Windows.

Vào đường dẫn: <https://git-scm.com/downloads> để tiến hành download Git:

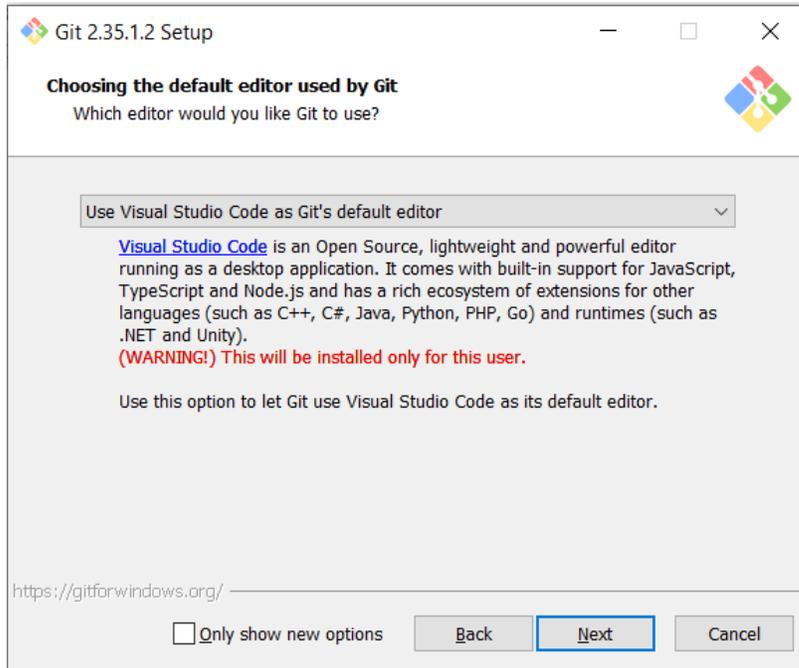


The screenshot shows the Git website's Downloads page. The header includes the Git logo and the tagline "--fast-version-control". A search bar is located in the top right. The left sidebar contains navigation links: About, Documentation, Downloads (highlighted), GUI Clients, Logos, and Community. The main content area features a "Downloads" section with icons for macOS, Windows, and Linux/Unix. A monitor graphic displays the "Latest source Release 2.37.0" with a "Download for Windows" button. Below this, there are sections for "GUI Clients" and "Logos".

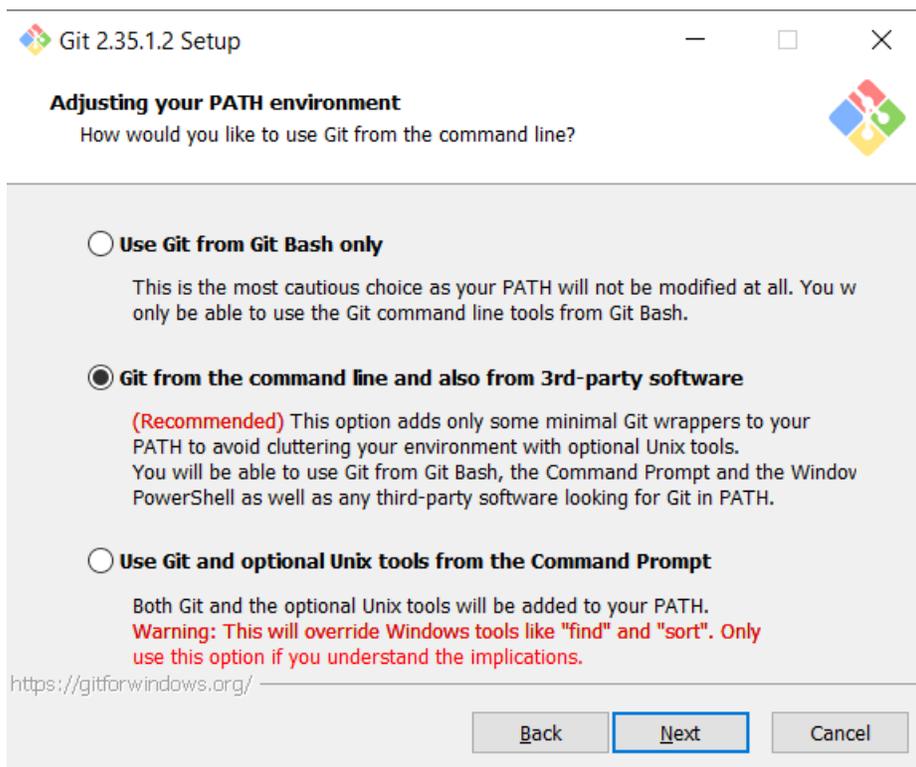
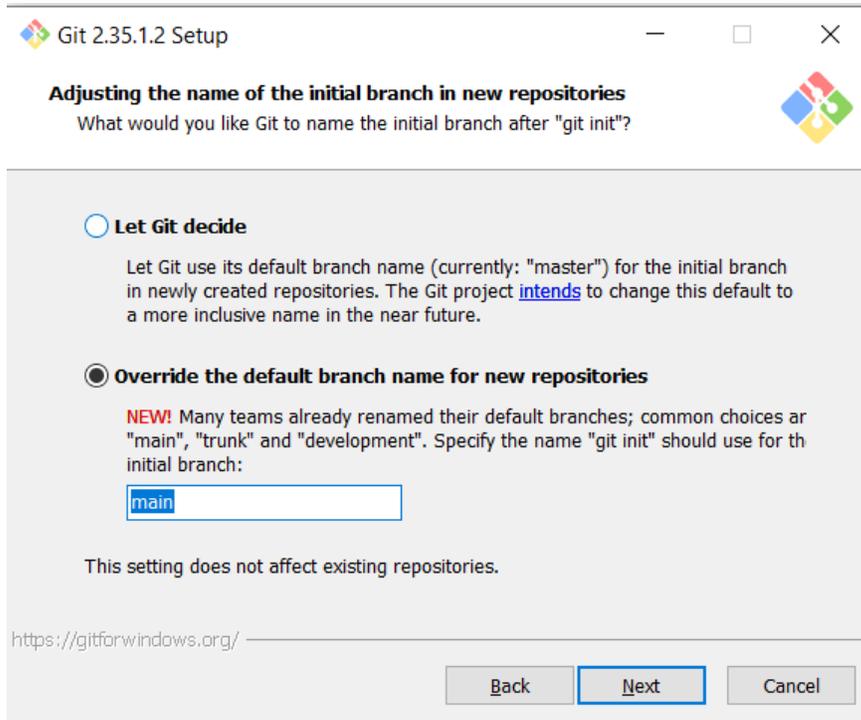
Chạy file vừa tải



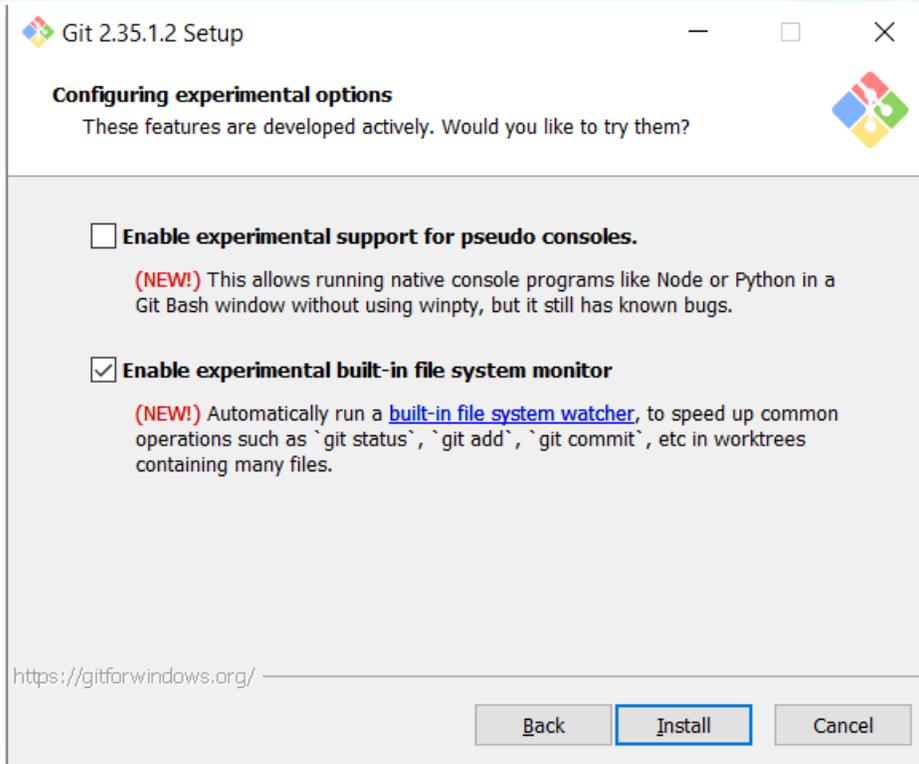
Ở hình bên dưới có thể chọn phần editor mà bạn muốn thì ở đây mình chọn Vim làm editor.



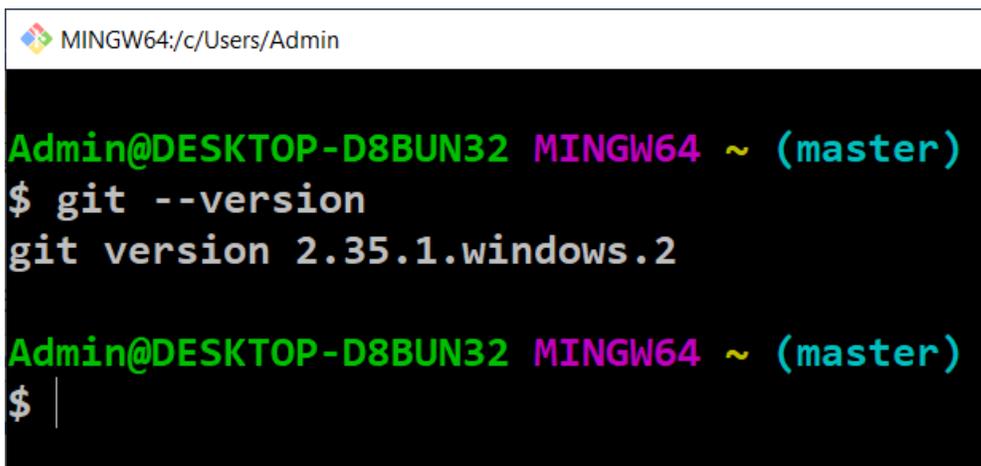
Hình bên dưới có 2 lựa chọn cho tên của branch default là master hoặc main ở đây mọi người có thể lựa chọn branch mà bạn muốn.



Sau đó cứ tiếp tục nhấn Next, đến cửa sổ bên dưới ta chọn vào mục “Enable ex...built-in file sys monitor” sau đó nhấn Install.



Để kiểm tra việc cài đặt git có thành công hay chưa chúng ta vào git bash và nhập `git --version`:



```
MINGW64:/c/Users/Admin  
  
Admin@DESKTOP-D8BUN32 MINGW64 ~ (master)  
$ git --version  
git version 2.35.1.windows.2  
  
Admin@DESKTOP-D8BUN32 MINGW64 ~ (master)  
$ |
```

Sau đó mở trình duyệt web và truy cập vào Ip: “10.215.26.199”, username: “root”, password: “VnPro@149”.



GitLab

A complete DevOps platform

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security.

This is a self-managed instance of GitLab.

Username or email

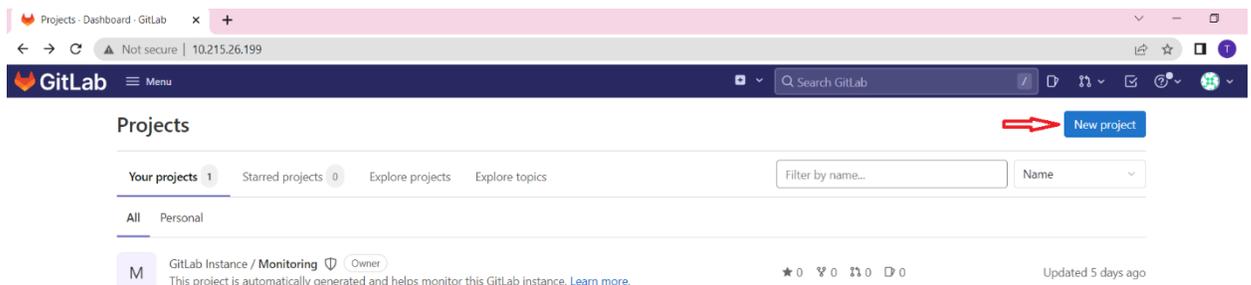
Password

Remember me [Forgot your password?](#)

Sign in

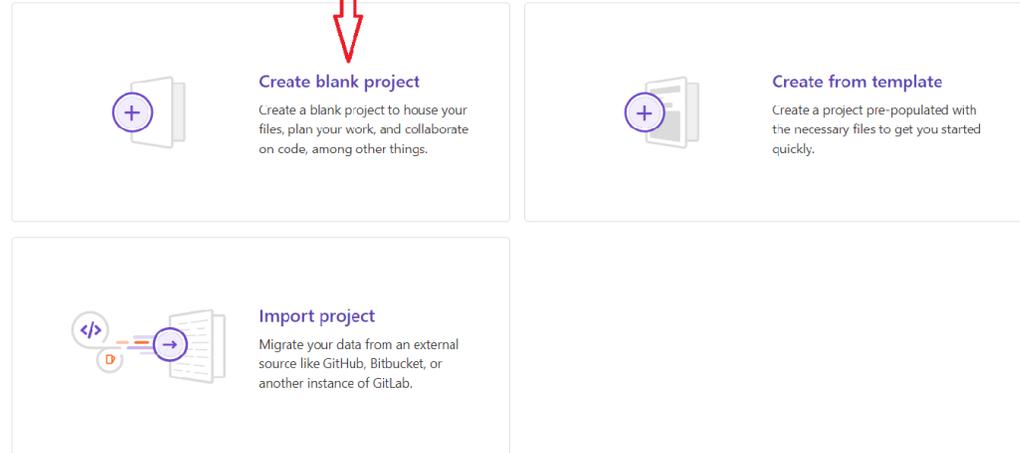
Don't have an account yet? [Register now](#)

Sau đó chọn New Project để tiến hành tạo 1 project mới. Sau khi tạo thành công, project vừa tạo sẽ xuất hiện ở mục Your Project. Tiếp theo, bạn sẽ clone repository này về máy tính của mình.



Sau đó chọn Create blank project

Create new project



The screenshot shows the 'Create new project' interface on GitLab. It features three main options:

- Create blank project**: Create a blank project to house your files, plan your work, and collaborate on code, among other things. A red arrow points to this option.
- Create from template**: Create a project pre-populated with the necessary files to get you started quickly.
- Import project**: Migrate your data from an external source like GitHub, Bitbucket, or another instance of GitLab.

Ta thực hiện tạo tên, viết mô tả cho project, chọn chế độ truy cập đối với project, sau đó là chọn Create Project.

New project > Create blank project

Project name

Project URL

Project slug

Want to house several dependent projects under the same namespace? [Create a group](#).

Project description (optional)

Visibility Level  Private

Project access must be granted explicitly to each user. If this project is part of a group, access will be 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.

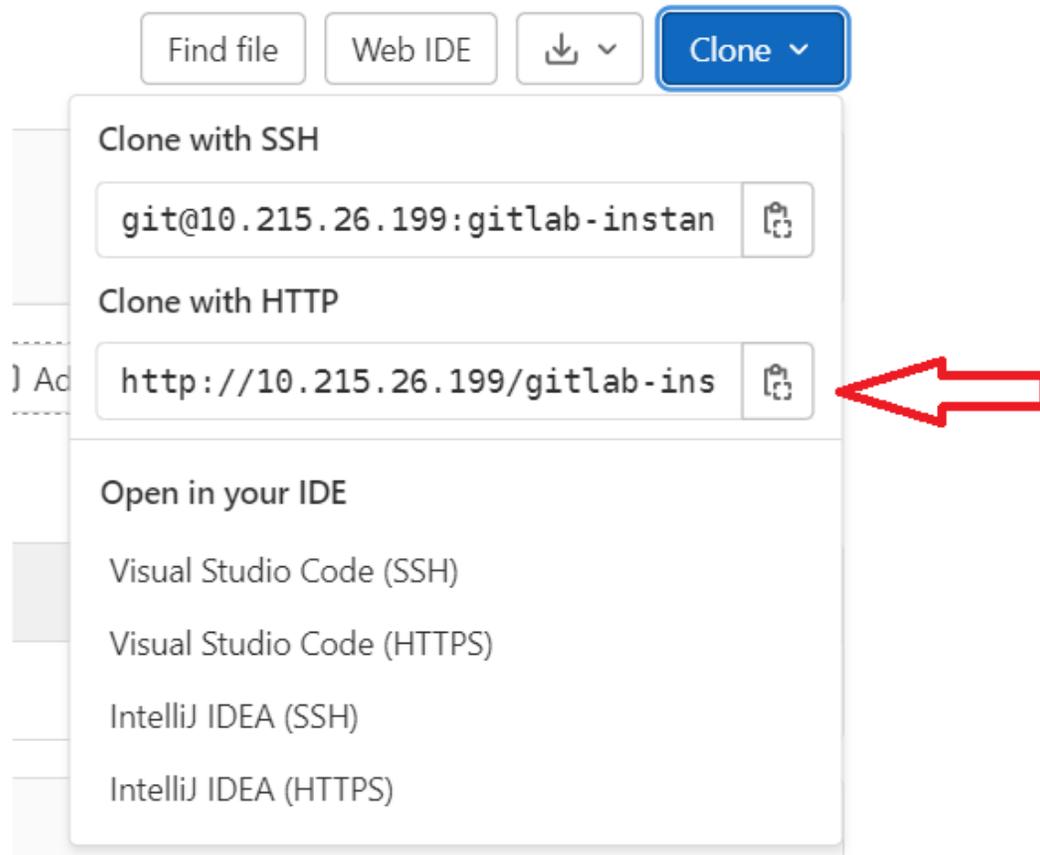
Project Configuration

 Initialize repository with a README

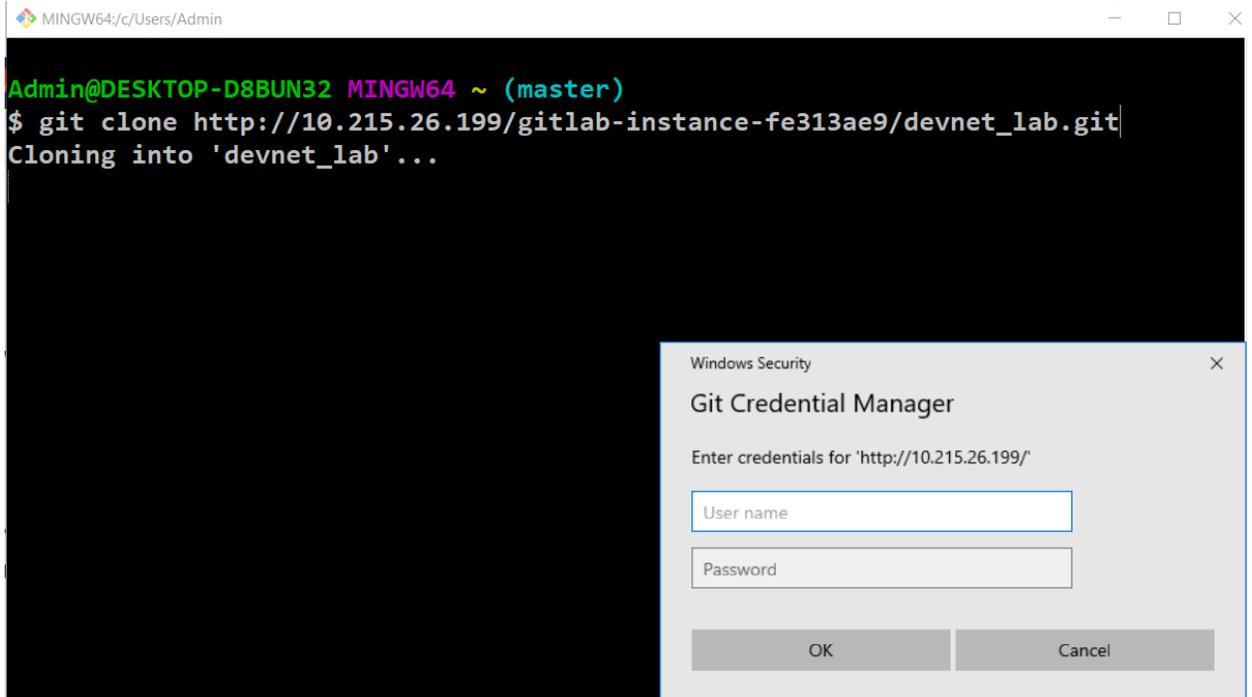
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

 Enable Static Application Security Testing (SAST)Analyze your source code for known security vulnerabilities. [Learn more](#).

Click vào project vừa tạo, chọn clone và click vào copy ở mục “Clone with HTTP”



Sau đó mở git bash và nhập lệnh git clone và đường dẫn mới lấy được sao đó nhập username và password đăng nhập và Gitlab của bạn để xác thực

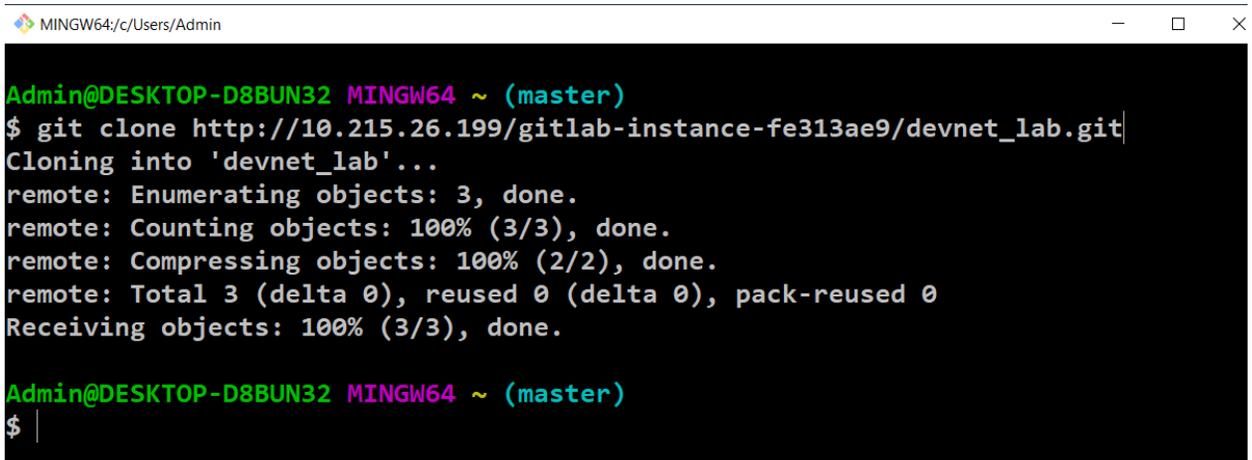


```
MINGW64/c/Users/Admin

Admin@DESKTOP-D8BUN32 MINGW64 ~ (master)
$ git clone http://10.215.26.199/gitlab-instance-fe313ae9/devnet_lab.git
Cloning into 'devnet_lab'...
```

Windows Security
Git Credential Manager
Enter credentials for 'http://10.215.26.199/'
User name
Password
OK Cancel

Project sau khi clone thành công.



```
MINGW64/c/Users/Admin

Admin@DESKTOP-D8BUN32 MINGW64 ~ (master)
$ git clone http://10.215.26.199/gitlab-instance-fe313ae9/devnet_lab.git
Cloning into 'devnet_lab'...
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
Receiving objects: 100% (3/3), done.

Admin@DESKTOP-D8BUN32 MINGW64 ~ (master)
$
```

Sử dụng câu lệnh `cd devnet_lab` (trong đó `devnet_lab` là tên project các bạn đặt lúc này) để vào thư mục vừa clone về. Sử dụng lệnh `ls -la` để show lên danh sách các thư mục hiện hành đang có trong project, ta sẽ thấy thư mục `README.md` ở trên Gitlab đã được download về máy tính của bạn:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO (master)
$ ls -la
total 66
drwxr-xr-x 1 Admin 197609      0 Jun 28 10:46 ./
drwxr-xr-x 1 Admin 197609      0 Jun 25 11:26 ../
drwxr-xr-x 1 Admin 197609      0 Apr 13 11:21 .git/
-rw-r--r-- 1 Admin 197609 17715 Apr 26 09:46 GIT_Commands.docx
drwxr-xr-x 1 Admin 197609      0 Apr 26 09:49 TrainingGit/
drwxr-xr-x 1 Admin 197609      0 Jun 28 09:53 devnet_lab/
-rw-r--r-- 1 Admin 197609      20 Mar 23 20:56 file1.py
-rw-r--r-- 1 Admin 197609    162 Apr 26 08:20 '~$T_Commands.docx'
-rw-r--r-- 1 Admin 197609 17546 Apr 26 08:51 '~WRL2803.tmp'
```

II. Commit and Push Changes

Ở mục này bạn sẽ học cách xác định trạng thái của local repository, cách để add files, commit files và sau đó là push files lên remote repository.

Mở Git Bash và tạo 1 file có tên là lab02.py bằng cách sử dụng câu lệnh touch:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ touch lab02.py

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ ls
README.md  lab02.py
```

Chạy câu lệnh git status để kiểm tra trạng thái hiện tại của repository:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
       lab02.py

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

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$
```

Để add file lab02.py lên repository, trước tiên bạn cần đánh dấu nó bằng câu lệnh git add lab02.py:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git add lab02.py

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$
```

Chạy lại git status, ta sẽ thấy file lab02.py bây giờ đã được đánh dấu và sẽ được đẩy lên repository trong lần commit tiếp theo:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
       new file:   lab02.py

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$
```

Commit file vào local repository bằng câu lệnh git commit -m. Nên thêm chú thích vào mỗi lần commit (ở trong hình đó là câu lệnh “Add lab02.py file”):

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git commit -m "Add lab02.py file"
[main 9953ce0] Add lab02.py file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 lab02.py

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$
```

Chạy lại lệnh `git status` sau khi commit bạn sẽ thấy bản sao của nhánh main sẽ được đẩy vào `origin/main` trên Gitlab Server trong lần commit tiếp theo. Có nghĩa là file `lab02.py` hiện tại vẫn là 1 phần của local repository nhưng chưa có trên Gitlab remote Server:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$
```

Để publish những lần commit ta sử dụng lệnh `git push origin main`. Nhánh local master của bạn sẽ đồng bộ hóa với nhánh main của remote:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git push origin main
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), 298 bytes | 298.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://10.215.26.199/gitlab-instance-fe313ae9/devnet_lab.git
 be9f063..9953ce0 main -> main

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$
```

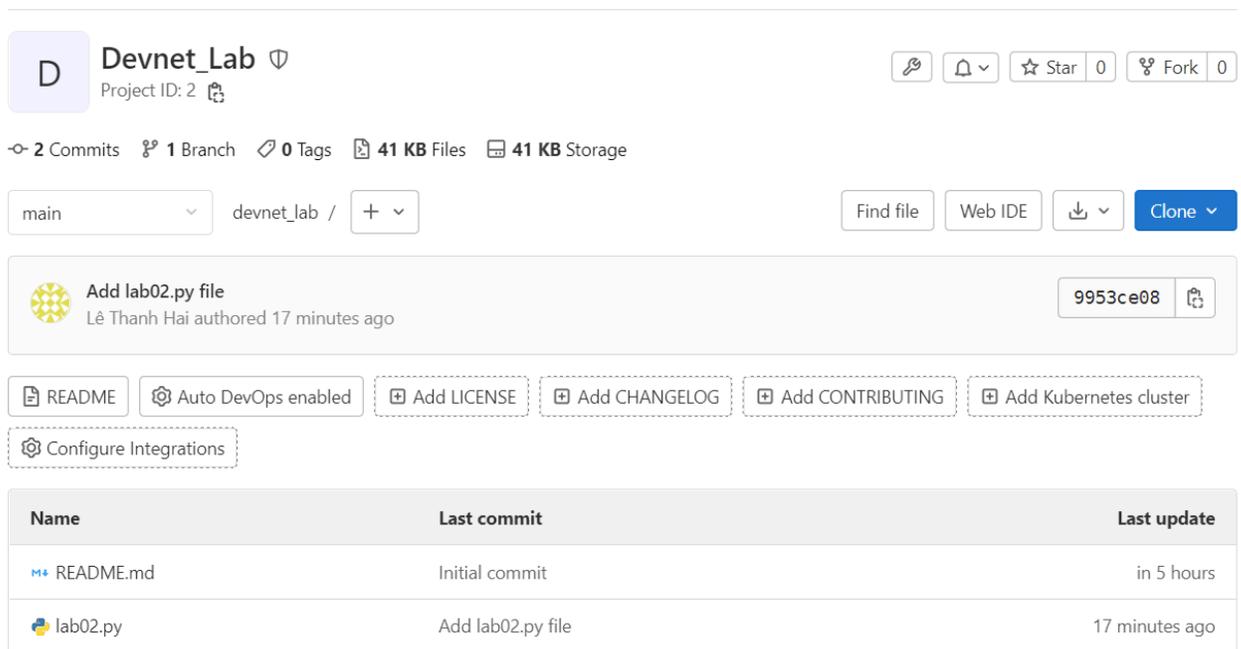
Chạy lệnh `git status` để kiểm tra:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$
```

Lúc này vào Gitlab, vào project mà bạn tạo sẽ thấy file lab02.py đã được đẩy lên remote repository:



The screenshot shows the GitLab interface for a repository named 'Devnet_Lab'. It includes a header with the repository name, project ID, and statistics (2 commits, 1 branch, 0 tags, 41 KB files, 41 KB storage). Below the header, there are buttons for 'Find file', 'Web IDE', and 'Clone'. A recent commit is shown with the message 'Add lab02.py file' by Lê Thanh Hai, 17 minutes ago. Below the commit list, there are buttons for 'README', 'Auto DevOps enabled', 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', and 'Add Kubernetes cluster'. A table at the bottom lists the files in the repository:

Name	Last commit	Last update
README.md	Initial commit	in 5 hours
lab02.py	Add lab02.py file	17 minutes ago

Thêm một đoạn code vào dòng lệnh vào file02.py để tiếp tục kiểm tra, Sửa file bằng câu lệnh vim lab02.py

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ vim lab02.py
```

Ở đây mình thêm vào file lab02.py một đoạn code print (“working with Git”)

```
MINGW64:/d/GIT_DEMO/devnet_lab
print("working with Git")

lab02.py[+] [unix] (10:51 28/06/2022) 1,26 All
-- INSERT --
```

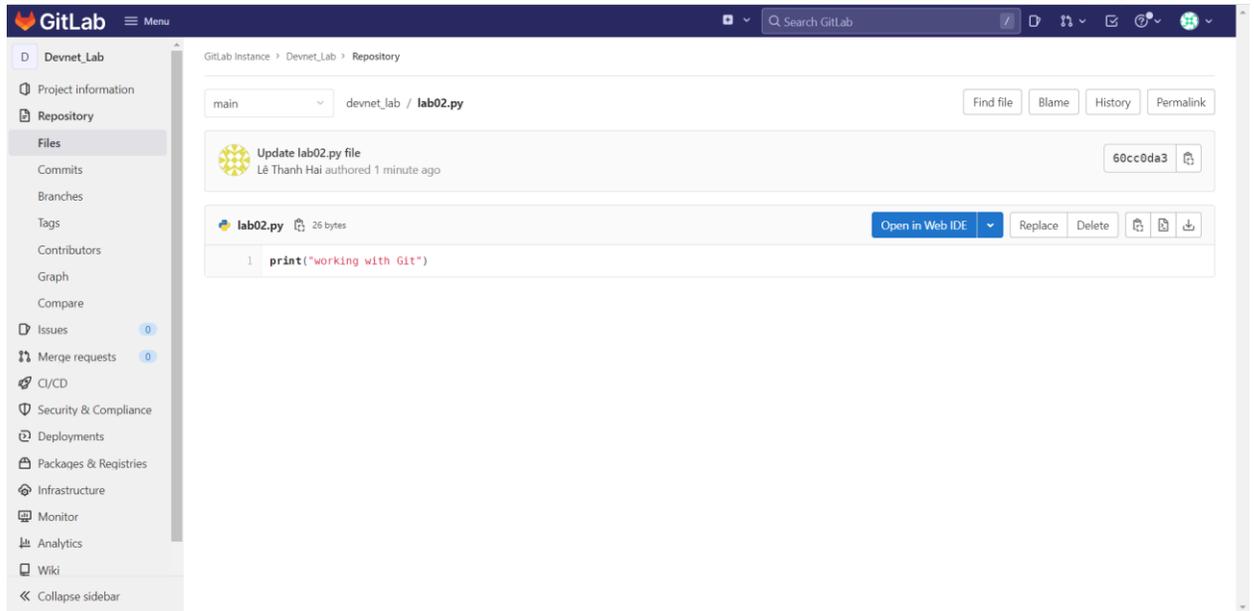
Sau đó tiếp tục dùng lệnh git add, git commit -m và git push origin master để update file lab02.py vừa chỉnh sửa sẽ được kết quả như trong hình:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git add lab02.py
warning: LF will be replaced by CRLF in lab02.py.
The file will have its original line endings in your working directory

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git commit -m "Update lab02.py file"
[main 60cc0da] Update lab02.py file
1 file changed, 1 insertion(+)

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 328 bytes | 328.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://10.215.26.199/gitlab-instance-fe313ae9/devnet_lab.git
9953ce0..60cc0da main -> main

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$
```



III. Compare Branches

Ở phần này bạn sẽ học cách tạo 1 branch, chuyển qua lại giữa các branch và cách đối chiếu các branch để tìm các file khác nhau.

Mở Git bash, tạo một branch mới trên local repository với tên là “staging” và chuyển qua nhánh mới bằng câu lệnh git checkout –b staging:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git checkout -b staging
Switched to a new branch 'staging'

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (staging)
$ git branch
  main
* staging

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (staging)
$
```

Vào file lab02.py sau đó thêm vào một dòng print(“New branch”):


```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (staging)
$ git add lab02.py
warning: LF will be replaced by CRLF in lab02.py.
The file will have its original line endings in your working directory

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (staging)
$ git commit -m "Add print statement"
[staging fa06ac8] Add print statement
1 file changed, 1 insertion(+)

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (staging)
$
```

IV. Merge Branch

Merge Branch là một nhiệm vụ phổ biến khi làm việc với git. Bạn sẽ liệt kê các nhánh hiện có, kiểm tra xem bạn đang làm việc ở đâu và hợp nhất với nhánh chính mà không xảy ra xung đột.

Để kiểm tra Branch hiện có sử dụng lệnh git branch. Dấu * phía trước tên sẽ thể hiện nhánh bạn đang làm việc.

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (staging)
$ git branch
main
* staging

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (staging)
$ |
```

Trước khi merge nhánh staging với nhánh main, chúng ta phải chuyển về nhánh main bằng câu lệnh git checkout main:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (staging)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git branch
* main
  staging

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$
```

Sau đó sử dụng câu lệnh git merge staging để merge nhánh staging vào nhánh main:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git branch
* main
  staging

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git merge staging
Updating 60cc0da..fa06ac8
Fast-forward
 lab02.py | 1 +
 1 file changed, 1 insertion(+)

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ |
```

Kiểm tra lại trạng thái của local repository:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ |
```

Sau đó push những thay đổi tới remote branch:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 337 bytes | 337.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://10.215.26.199/gitlab-instance-fe313ae9/devnet_lab.git
 60cc0da..fa06ac8  main -> main

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$
```

Lúc này ta nhập lệnh git diff sẽ không thấy sự khác nhau giữa nhánh master và nhánh staging nữa:

```
Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$ git diff

Admin@DESKTOP-D8BUN32 MINGW64 /d/GIT_DEMO/devnet_lab (main)
$
```