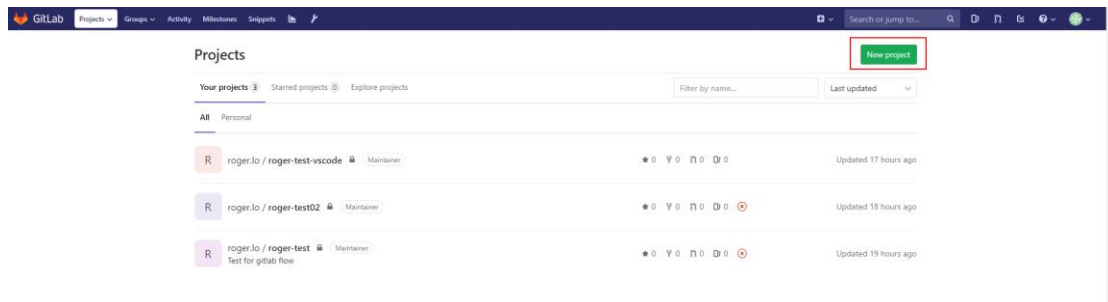


1. Create a remote repository and clone it.

Login gitlab web page and then click the “New project” button



Type the name & description, choose visibility level, check the last item to auto make a readme file.

Blank project

Create from template

Import project

Project name

git-operate-sample

Project URL

http://10.4.3.157:8088/ roger

Project slug

git-operate-sample

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

Project description (optional)

A sample for try git

Visibility Level

☒ Private
Project access must be granted explicitly to each user.

☐ Internal
The project can be accessed by any logged in user.

☐ Public
The project can be accessed without any authentication.

☒ 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.

Create project

Cancel

Get the clone URL

The screenshot shows the GitHub interface for a repository named 'git-operate-sample'. The repository is owned by 'roger.io' and has 0 stars, 0 forks, and 0 tags. A 'Clone' button is highlighted with a red box. Below it, two options for cloning are shown: 'Clone with SSH' (git@localhost:roger/git-operate) and 'Clone with HTTP' (http://localhost:8088/roger/git). The 'Clone with HTTP' option is also highlighted with a red box. Below the cloning options, there is a section for the 'Initial commit' by 'roger.io' 35 seconds ago. At the bottom, there is a table with columns 'Name', 'Last commit', and 'Last update'. The table contains one row for 'README.md' with the last commit being 'Initial commit' 36 seconds ago. Below the table, there is a section for the 'README.md' file, which contains the text 'git-operate-sample' and 'A sample for try git'.

Clone this project to your PC by this command

```
git clone http://10.4.3.157:8088/roger/git-operate-sample.git git-operate-sample
```

It should be:

```
C:\vscode_workspace>git clone http://10.4.3.157:8088/roger/git-operate-sample.git git-operate-sample
Cloning into 'git-operate-sample'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
```

WIN10 (C:) > vscode_workspace

名稱	修改日期	類型
[redacted]	2019/3/27 下午 01:59	檔案資料夾
git-operate-sample	2019/4/2 上午 10:26	檔案資料夾
[redacted]	2019/3/27 下午 02:35	檔案資料夾
[redacted]	2019/3/28 上午 09:57	檔案資料夾
roger-test	2019/4/1 下午 02:30	檔案資料夾
roger-test02	2019/4/1 下午 03:55	檔案資料夾
roger-test-vscode2	2019/4/1 下午 04:49	檔案資料夾
temp	2019/3/26 下午 05:53	檔案資料夾

2. Basic operation

Get this folder status of git

```
git status
```

Normal case:

```
C:\vscode_workspace\git-operate-sample>git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

If something change:

```
C:\vscode_workspace\git-operate-sample>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

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

After modify something, we have to add the modified file to temp stage first

```
git add
```

It should be:

```
C:\vscode_workspace\git-operate-sample>git add *.md
C:\vscode_workspace\git-operate-sample>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   README.md
```

The next step, we have to commit it to local repository, use “-m” to type commit message.

```
git commit -m
```

It should be:

```
C:\vscode_workspace\git-operate-sample>git commit -m "First time to commit"
[master cb82bd3] First time to commit
1 file changed, 3 insertions(+), 1 deletion(-)
```

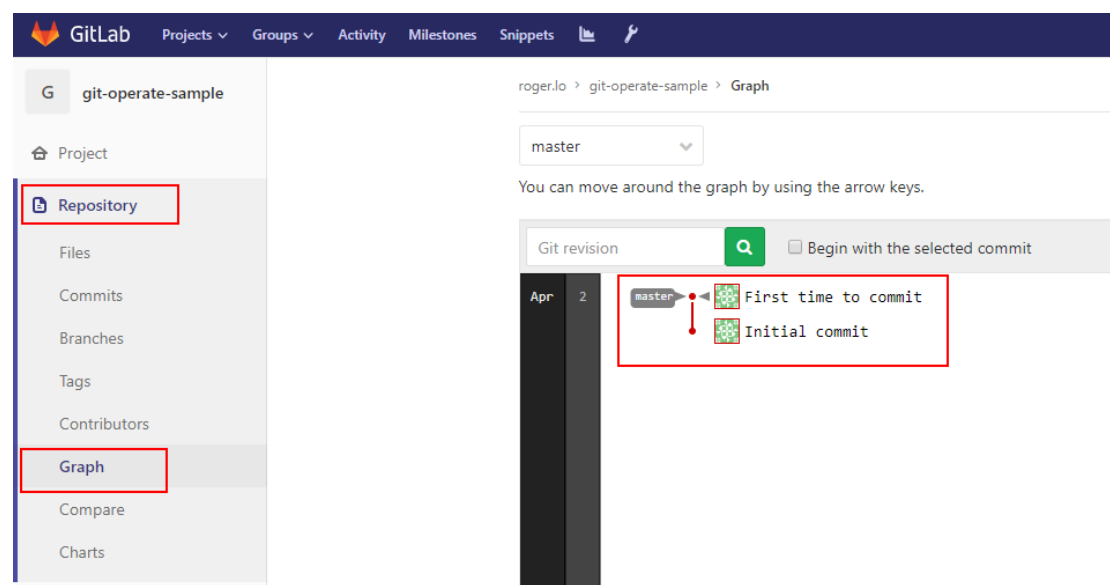
Now, we can push this commit to remote repository

```
git push
```

It should be:

```
C:\vscode_workspace\git-operate-sample>git push
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), 297 bytes | 148.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To http://10.4.3.157:8088/roger/git-operate-sample.git
    1da293e..cb82bd3  master -> master
```

And you can see the graph at gitlab web page



If we want to get the last version from remote repository

```
git pull --no-ff
```

Use “—no-ff” to make a merge commit node.

It should be:

```
C:\vscode_workspace\git-operate-sample>git pull --no-ff
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From http://10.4.3.157:8088/roger/git-operate-sample
   cb82bd3..a4b38c3  master      -> origin/master
Merge made by the 'recursive' strategy.
 README.md | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

If we want to get log for local repository

```
git log --oneline
```

```
C:\vscode_workspace\git-operate-sample>git log --oneline
64f1234 (HEAD -> master) Merge branch 'master' of http://10.4.3.157:8088/roger/git-operate-sample
a4b38c3 (origin/roger_remote, origin/roger, origin/master, origin/HEAD) Update README.md
cb82bd3 First time to commit
1da293e Initial commit
```

3. Branch

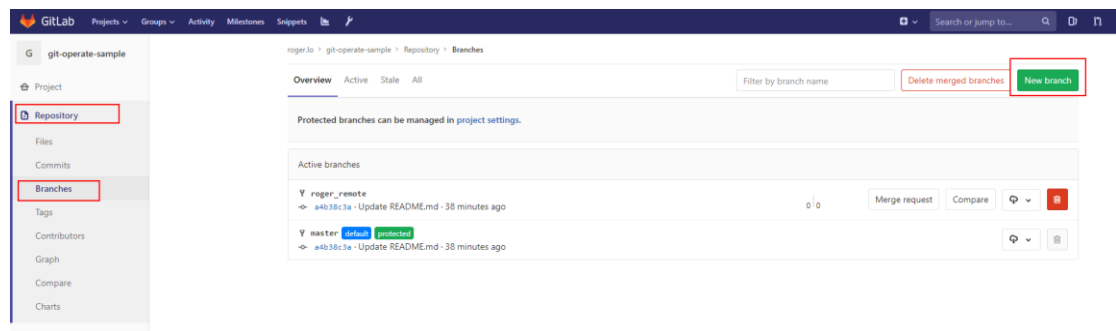
Get the branch status

```
git branch
```

It should be:

```
C:\vscode_workspace\git-operate-sample>git branch
* master
```

Create branch on remote at gitlab web page



Update the remote branch for local by “git pull” and checkout the branch

```
git checkout
```

It should be:

```
C:\vscode_workspace\git-operate-sample>git checkout roger_remote
Switched to a new branch 'roger_remote'
Branch 'roger_remote' set up to track remote branch 'roger_remote' from 'origin'.
```

Now, we should have two branch:

```
C:\vscode_workspace\git-operate-sample>git branch
master
* roger_remote
```

Create local branch by "git branch"

If only "git branch", this command can get branch status.

If append a branch name, this command will create a new branch.

```
C:\vscode_workspace\git-operate-sample>git branch roger_local  
C:\vscode_workspace\git-operate-sample>git branch  
master  
roger_local  
* roger_remote
```

Checkout the new branch, it should be:

```
C:\vscode_workspace\git-operate-sample>git checkout roger_local  
Switched to branch 'roger_local'  
C:\vscode_workspace\git-operate-sample>git branch  
master  
* roger_local  
roger_remote  
C:\vscode_workspace\git-operate-sample>git status  
On branch roger_local  
nothing to commit, working tree clean
```

4. Merge

Use command to merge other branch,

```
git merge --no-ff
```

Use “--no-ff” to make a merge commit node.

It should be:

```
C:\vscode_workspace\git-operate-sample>git merge roger_local --no-ff
Merge made by the 'recursive' strategy.
 README.md | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

The git log:

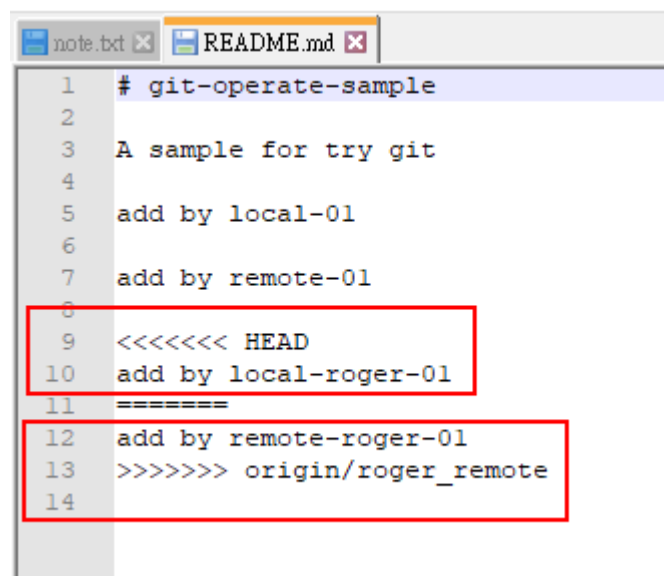
```
C:\vscode_workspace\git-operate-sample>git log --oneline
d533e6b (HEAD -> roger_remote) Merge branch 'roger_local' into roger_remote
eee781c (roger_local) commit by local roger
a4b38c3 (origin/roger_remote, origin/roger, origin/master, origin/HEAD) Update README.md
cb82bd3 First time to commit
1da293e Initial commit
```

5. Solve conflict

If the source code has conflict when merge, it should be:

```
C:\vscode_workspace\git-operate-sample>git merge origin/roger_remote
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```

And the file content should be:



The screenshot shows a code editor with two tabs: 'note.txt' and 'README.md'. The 'README.md' tab is active, displaying the following content:

```
1 # git-operate-sample
2
3 A sample for try git
4
5 add by local-01
6
7 add by remote-01
8
9 <<<<<<< HEAD
10 add by local-roger-01
11 =====
12 add by remote-roger-01
13 >>>>>> origin/roger_remote
14
```

Red boxes highlight the conflict markers: lines 9-10 ('<<<<<< HEAD' and 'add by local-roger-01'), line 11 ('====='), and lines 12-13 ('add by remote-roger-01' and '>>>>>> origin/roger_remote').

The git status should be:

```
C:\vscode_workspace\git-operate-sample>git status
On branch roger_remote
Your branch and 'origin/roger_remote' have diverged,
and have 2 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)

        both modified:   README.md

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

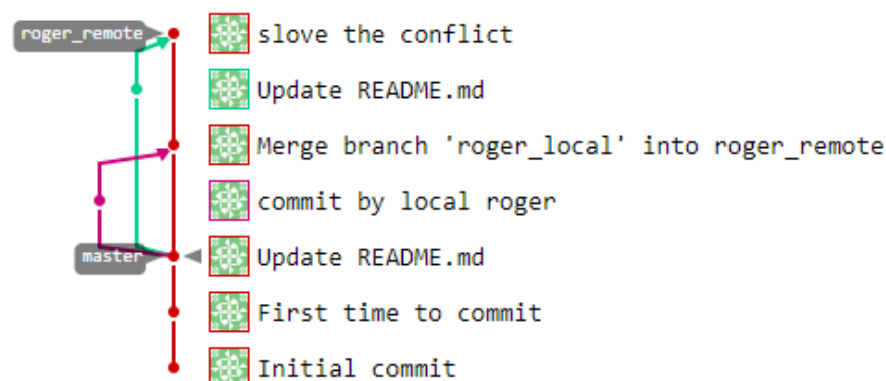
After we solve the conflict, just add the file to temp stage, and then the git status will change to:

```
C:\vscode_workspace\git-operate-sample>git add *.md
C:\vscode_workspace\git-operate-sample>git status
On branch roger_remote
Your branch and 'origin/roger_remote' have diverged,
and have 2 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
        modified:   README.md
```

Finally, we can just commit & push, and you'll see the graph at gitlab web page.



6. Rebase

We can also use “rebase” to merge two branch

```
git rebase
```

It should be:

```
C:\vscode_workspace\git-operate-sample>git rebase roger_local
First, rewinding head to replay your work on top of it...
```

And the git status & log will be:

```
C:\vscode_workspace\git-operate-sample>git status
On branch master
Your branch is ahead of 'origin/master' by 6 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\vscode_workspace\git-operate-sample>git log --oneline
72b2b3d (HEAD -> master, roger_local) commit change from local roger
a/aa54f Merge branch 'roger_remote' into roger_local
ef34dal (origin/roger_remote) solve the conflict
24df3ea Update README.md
d533e6b Merge branch 'roger_local' into roger_remote
eee781c commit by local roger
a4b38c3 (origin/roger, origin/master, origin/HEAD) Update README.md
cb82bd3 First time to commit
1da293e Initial commit
```

If we have conflict when rebase, it should be:

```
C:\vscode_workspace\git-operate-sample>git rebase master
First, rewinding head to replay your work on top of it...
Applying: commit change for prepare rebase conflict
Using index info to reconstruct a base tree...
M   README.md
Falling back to patching base and 3-way merge...
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
error: Failed to merge in the changes.
hint: Use 'git am --show-current-patch' to see the failed patch
Patch failed at 0001 commit change for prepare rebase conflict
Resolve all conflicts manually, mark them as resolved with
"git add/rm <conflicted_files>", then run "git rebase --continue".
You can instead skip this commit: run "git rebase --skip".
To abort and get back to the state before "git rebase", run "git rebase --abort".
```

Now, the git status should be:

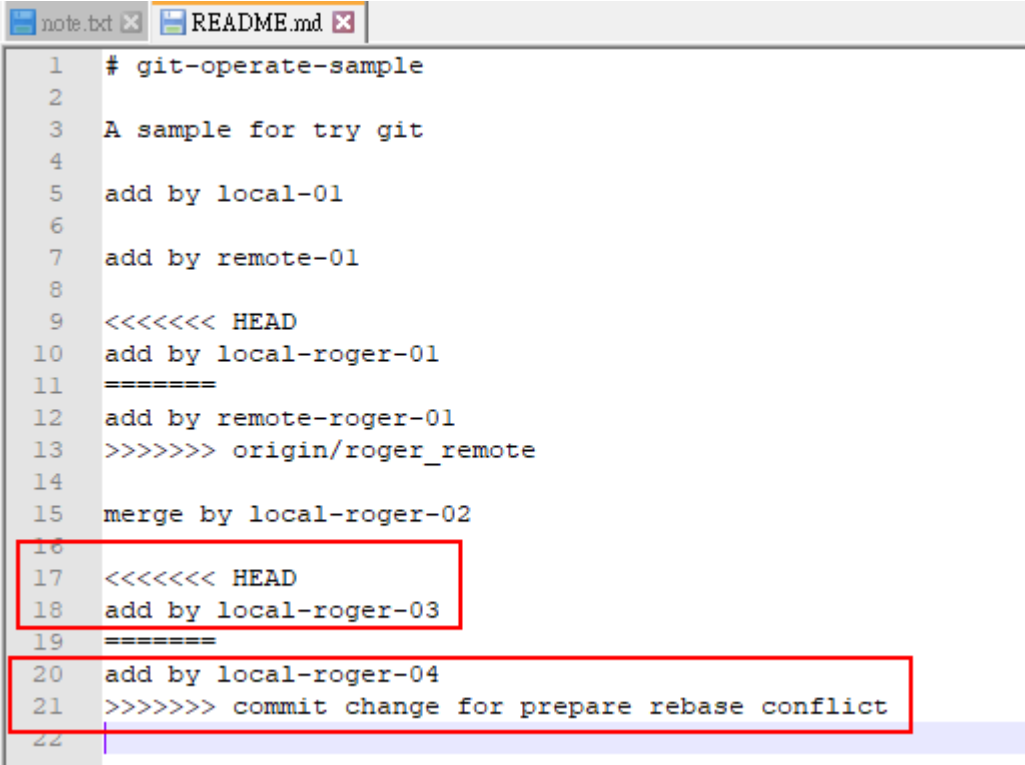
```
C:\vscode workspace\git-operate-sample>git status
rebase in progress; onto 72b2b3d
You are currently rebasing branch 'roger_remote' on '72b2b3d'.
  (fix conflicts and then run "git rebase --continue")
  (use "git rebase --skip" to skip this patch)
  (use "git rebase --abort" to check out the original branch)

Unmerged paths:
  (use "git reset HEAD <file>..." to unstage)
  (use "git add <file>..." to mark resolution)

        both modified:   README.md

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

And the file content will be:



```
note.txt x README.md x
1 # git-operate-sample
2
3 A sample for try git
4
5 add by local-01
6
7 add by remote-01
8
9 <<<<<<< HEAD
10 add by local-roger-01
11 =====
12 add by remote-roger-01
13 >>>>>>> origin/roger_remote
14
15 merge by local-roger-02
16
17 <<<<<<< HEAD
18 add by local-roger-03
19 =====
20 add by local-roger-04
21 >>>>>>> commit change for prepare rebase conflict
22
```

```
C:\vscode_workspace\git-operate-sample>git add *.md
C:\vscode_workspace\git-operate-sample>git status
rebase in progress; onto 72b2b3d
You are currently rebasing branch 'roger_remote' on '72b2b3d'.
(all conflicts fixed: run "git rebase --continue")

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   README.md

C:\vscode_workspace\git-operate-sample>git rebase --continue
Applying: commit change for prepare rebase conflict
```

The diagram illustrates the sequence of operations for a rebase conflict:

- Initial commit
- First time to commit
- Update README.md
- commit by local roger
- Merge branch 'roger_local' into roger_remote
- Update README.md
- Merge branch 'roger_remote' into roger_local
- slove the conflict
- commit change from local roger
- commit change for prepare rebase conflict

7. Chrry pick

If you like some commit of other branch, like this sample:

```
C:\vscode_workspace\git-operate-sample>git branch
master
* roger_local
roger_remote

C:\vscode_workspace\git-operate-sample>git log --oneline
763fb32 (HEAD -> roger_local) add txt03
f57a2f6 add txt02
d5e0603 add txt01
1e50bfb solve the conflict
3caa66a Merge branch 'master' into roger_remote
ee42elf (origin/master, origin/HEAD, master) Merge branch 'roger_remote' into 'master'
a8219d4 (origin/roger_remote) solve conflict
4e23482 add new line03
c4170df add new line02
d3a8c02 add new line01
86b4324 Merge branch 'roger_local' into roger_remote
```

And you want to get the commit “add txt02” & “add txt03” to “roger_remote” branch

Checkout to the target branch, and then use this command

```
git cherry-pick
```

You need append the SHA-1 string of the commit that you want to this command like this sample:

```
C:\vscode_workspace\git-operate-sample>git branch
master
roger_local
* roger_remote

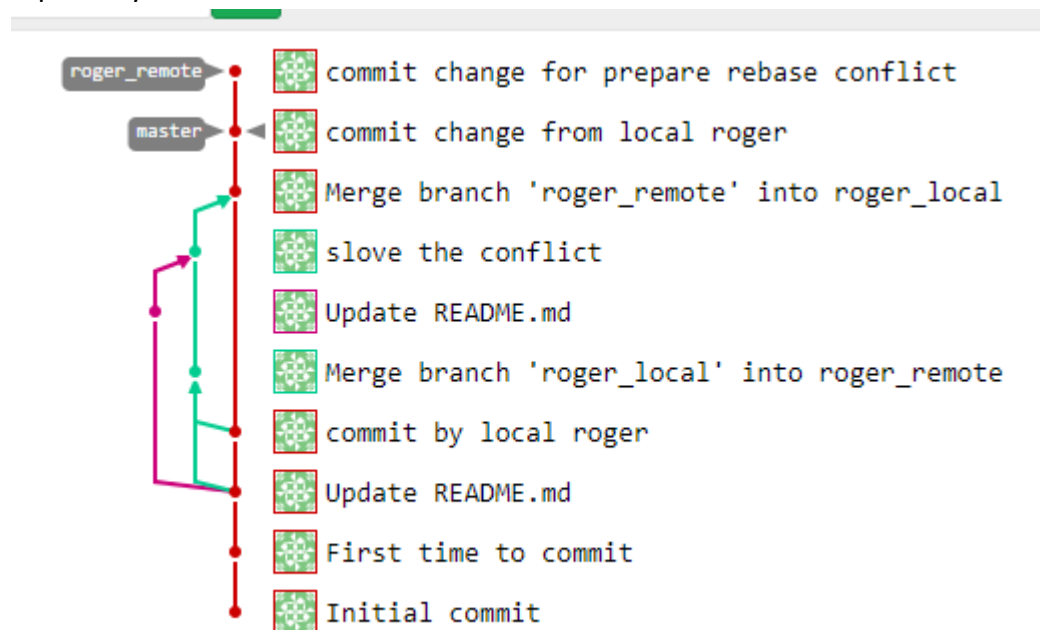
C:\vscode_workspace\git-operate-sample>git cherry-pick f57a2f6 763fb32
[roger_remote e54e5ac] add txt02
Date: Tue Apr 2 15:37:29 2019 +0800
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 add02.txt
[roger_remote 1dd1e7b] add txt03
Date: Tue Apr 2 15:37:41 2019 +0800
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 add03.txt
```

And the log will be:

```
C:\vscode_workspace\git-operate-sample>git log --oneline
1dd1e7b (HEAD -> roger_remote) add txt03
e54e5ac add txt02
148f8a0 Merge branch 'roger_local' into roger_remote
1e50bfb solve the conflict
3caa66a Merge branch 'master' into roger_remote
```

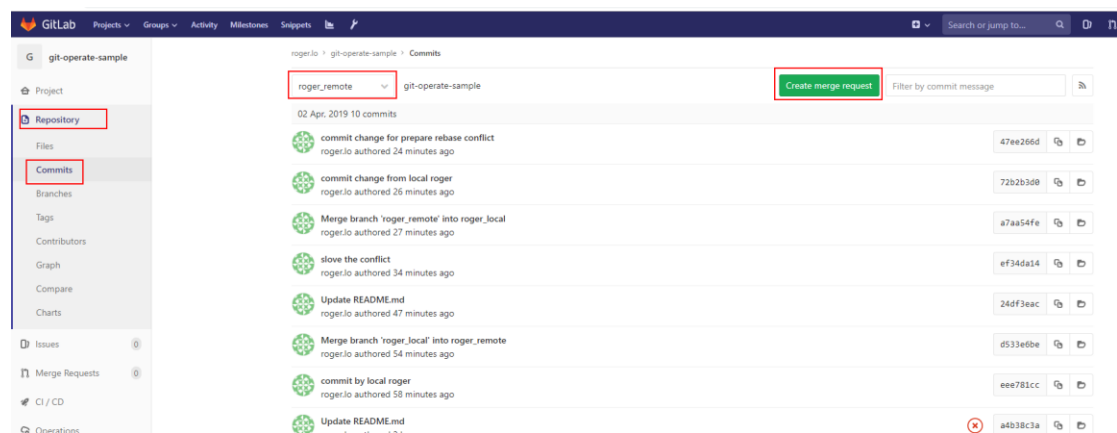
8. Merge request

When some branch will have more commit than branch “master” at remote repository.



For protected, we can't merge into “master” branch directly.

We have to create a “merge request” to administrator of “master” branch first.



Fill out all information, and then submit

From roger_remote into master [Change branches](#)

Title

Start the title with **WIP:** to prevent a **Work In Progress** merge request from being merged before it's ready.
Add [description templates](#) to help your contributors communicate effectively!

Description Write Preview

Try to create a merge request

Markdown and quick actions are supported [Attach a file](#)

Assignee [Assign to me](#)

Milestone

Labels

Source branch

Target branch [Change branches](#)

☐ Delete source branch when merge request is accepted.

☐ Squash commits when merge request is accepted. [?](#)

[Submit merge request](#) [Cancel](#)

Administrator will see it:

GitLab Projects Groups Activity Milestones Snippets

git-operate-sample

Project

Repository

Issues 0

Merge Requests 1

CI / CD

Operations

Wiki

Snippets

Settings

roger.io > git-operate-sample > Merge Requests

Open 1 Merged 0 Closed 0 All 1

Search or filter results...

commit change for prepare rebase conflict

!1 · opened 30 seconds ago by roger.io

Administrator can write some comment and choose that close this request or merge it.

roger.io > git-operate-sample > Merge Requests > !!

[Open](#) Opened 1 minute ago by roger.io [Edit](#) [Close merge request](#)

commit change for prepare rebase conflict

Try to create a merge request

Request to merge `roger_remote` into `master` [Open in Web IDE](#) [Check out branch](#)

[Merge](#) ☐ Delete source branch

> 1 commit and 1 merge commit will be added to master. [Modify merge commit](#)

You can merge this merge request manually using the [command line](#)

0 0

Discussion 0 Commits 1 Changes 1 [Show all activity](#)

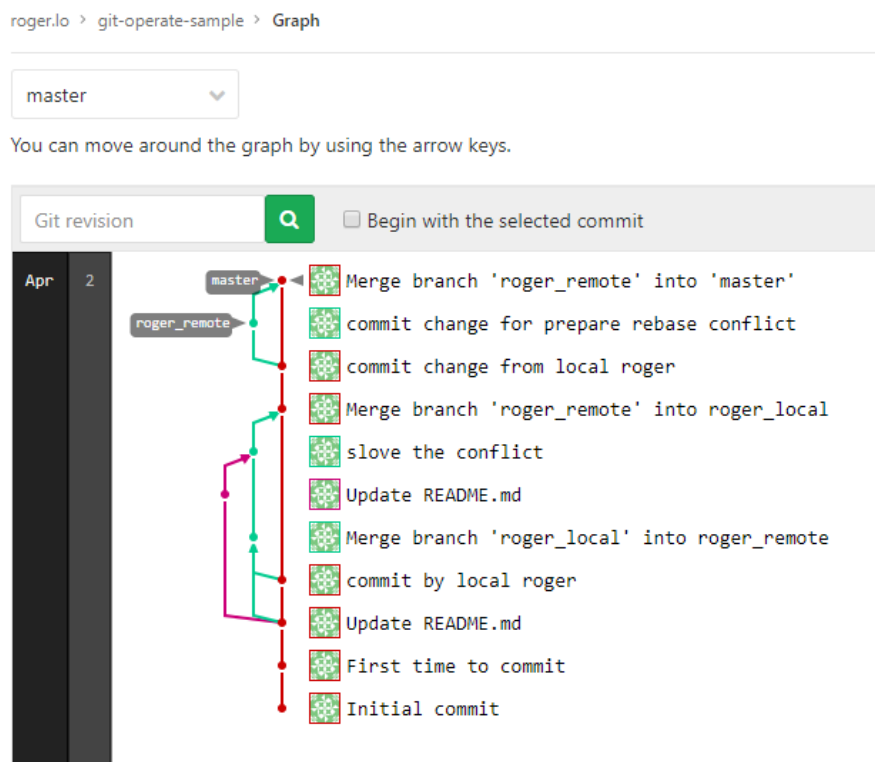
Write [Preview](#) [B](#) [I](#) [Q](#) [Code icon](#) [Link icon](#) [List icon](#) [Image icon](#) [Video icon](#) [Giphy icon](#)

Write a comment or drag your files here...

Markdown and quick actions are supported [Attach a file](#)

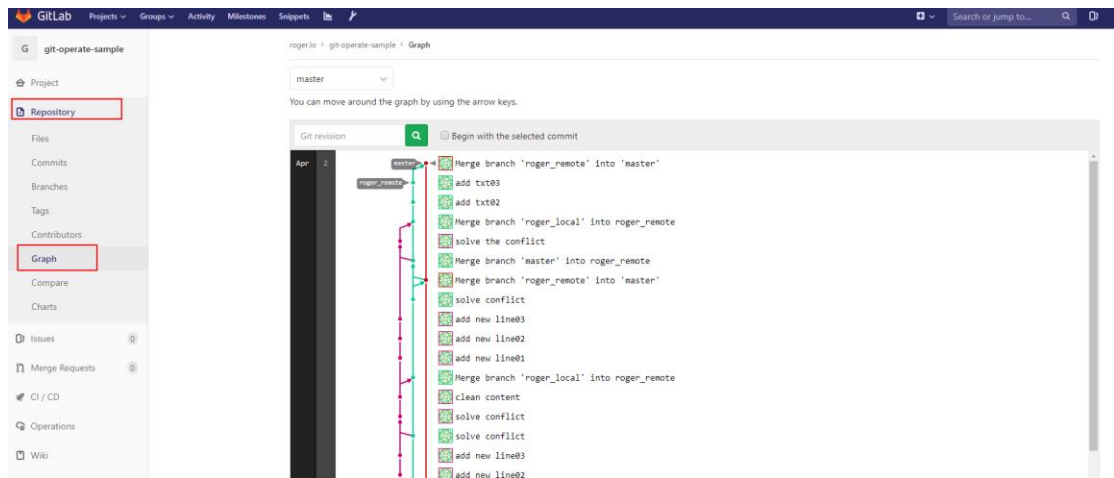
[Comment](#) [Close merge request](#)

After administrator merge it, you'll see the graph at gitlab web page:

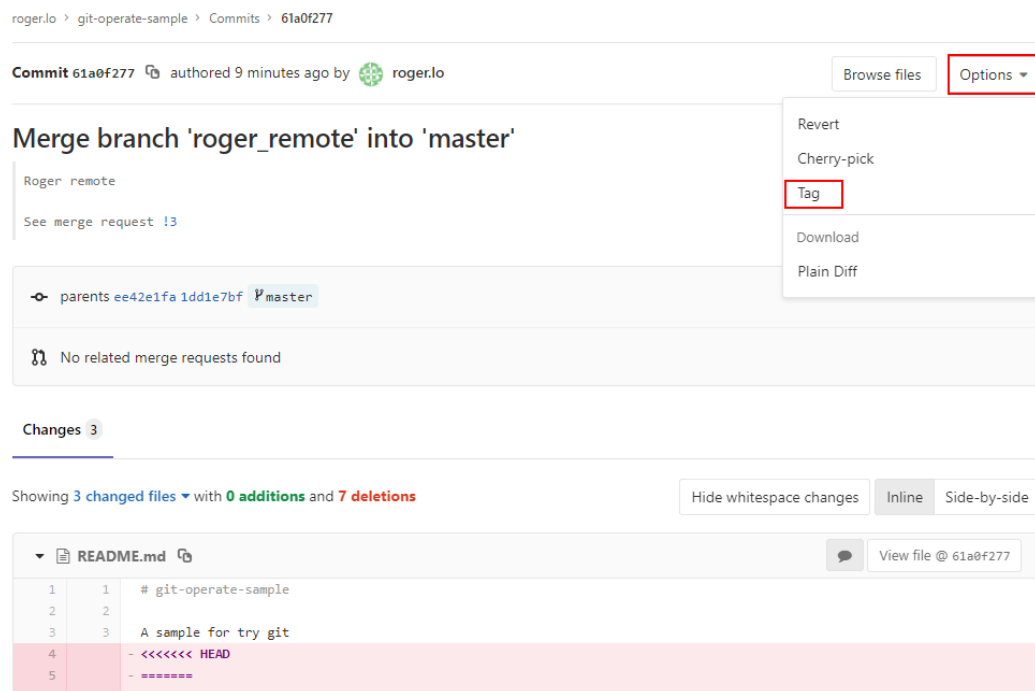


9. Create tag

Choose which commit you want to tag from gitlab web page.



Just click the commit node and then click the “tag” button at commit page.



Fill out the tag information and create it

roger.io > git-operate-sample > New Tag

New Tag

Tag name

Create from

Existing branch name, tag, or commit SHA

Message

Optionally, add a message to the tag.

Release notes Write Preview

Write your release notes or drag files here...

Markdown is supported [Attach a file](#)

Optionally, add release notes to the tag. They will be stored in the GitLab database and displayed on the tags page.

[Create tag](#) [Cancel](#)

You'll see the tag on the graph at gitlab web page.

GitLab Projects Groups Activity Milestones Snippets Search or jump to...

git-operate-sample

Repository

Files

Commits

Branches

Tags

Contributors

Graph

Compare

Charts

Issues 0

Merge Requests 0

CI / CD

Operations

Wiki

roger.io > git-operate-sample > Graph

master

You can move around the graph by using the arrow keys.

Git revision ☐ Begin with the selected commit

Apr 2

merge branch 'roger_remote' into 'master'

add txt03

add txt02

Merge branch 'roger_local' into roger_remote

solve the conflict

Merge branch 'master' into roger_remote

Merge branch 'roger_remote' into 'master'

solve conflict

add new line03

add new line02

add new line01

Merge branch 'roger_local' into roger_remote

clean content

solve conflict

solve conflict

add new line03

add new line02