**Lab GitLab Command**

**Lab1: Basic git Command**

1. Install git on Local Computer URL for download git: <https://git-scm.com/downloads>
2. ตรวจสอบ Version Git บน Local Computer โดยใช้คำสั่ง git --version
3. ให้ Config Username, Email
4. ใช้คำสั่ง git clone <http://ta-git-app1.bdms.co.th/training/TrainingGitLab.git>
5. ใช้คำสั่ง cd .. ไปที่ Folder “..\TrainingGitLab”
6. ใช้คำสั่ง git checkout -b **branch\_training\_amp** เพื่อสร้าง branch
7. สร้างไฟล์ Folder ชื่อของต้นเอง(..\TrainingGitLab\nickname) ภายใต้ project
8. สร้างไฟล์ชื่อ “README.md” ภายใต้ Folder ข้อ 7.
9. แก้ไขไฟล์ “README.md” โดยเพิ่มข้อความด้านล่าง จากนั้นบันทึกไฟล์

**#### ADD YOUR INFORMATION**

**#### Press: control + X**

**#### Type: Y**

**#### Press: enter**

1. ใช้คำสั่ง git add ., git commit -m “Test Add File Readme”เพื่อเก็บประวัติไฟล์ “README.md”
2. ใช้คำสั่ง git status เพื่อตรวจสอบการเปลี่ยนแปลง
3. ใช้คำสั่ง git push -u origin **branch\_training\_nickname** เพื่อส่งการเปลี่ยนแปลงไปยัง gitlab
4. ใส่ Username/Password: crmft01/crmft01

**Lab2: untrack file**

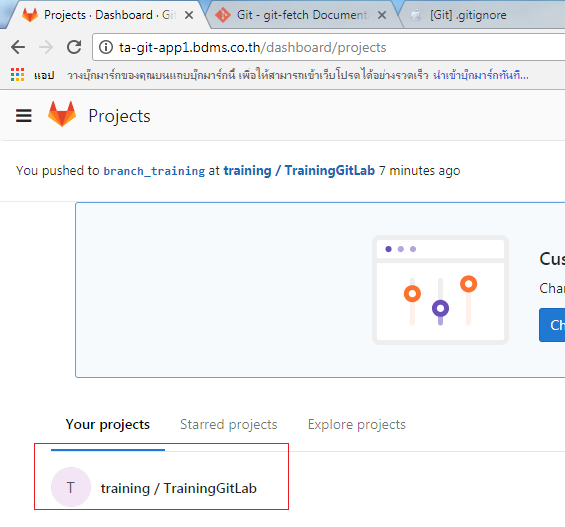
1. สร้างไฟล์ชื่อ “untrack\_file.txt” ภายใต้ Folder “..\TrainingGitLab\nickname”
2. แก้ไขไฟล์ “untrack\_file.txt” โดยเพิ่มข้อความด้านล่าง จากนั้นบันทึกไฟล์

**#### Test untrack file**

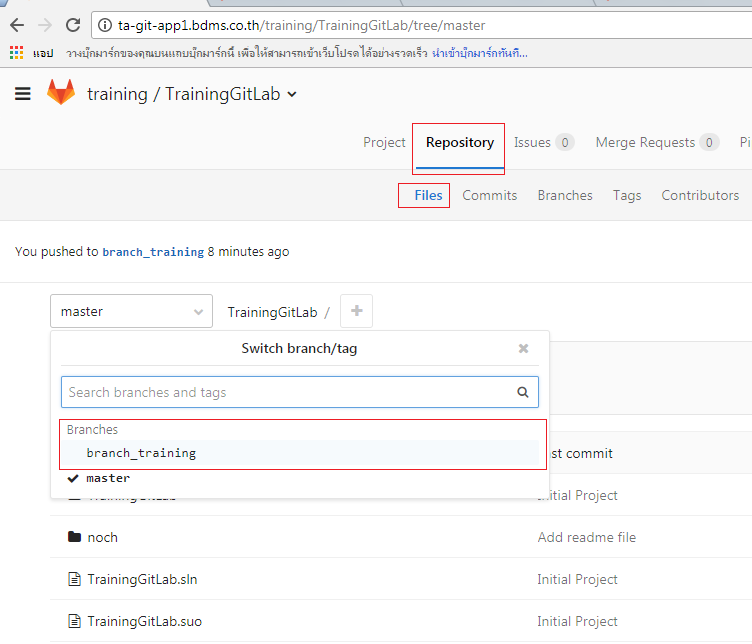
1. สร้าง Folder ชื่อ “untrack\_folder” ภายใต้ Folder ..\TrainingGitLab\nickname
2. สร้างไฟล์ชื่อ “untrack\_file\_in\_folder.txt” ภายใต้ Folder “..\TrainingGitLab\nickname\untrack\_folder”
3. แก้ไขไฟล์ “untrack\_file\_in\_folder.txt” โดยเพิ่มข้อความด้านล่าง จากนั้นบันทึกไฟล์

**#### Test untrack file in folder**

1. ใช้คำสั่ง cd .. เพื่อกลับมาที่ Folder “..\TrainingGitLab\nickname”
2. ใช้คำสั่ง git add ., git commit -m เพื่อเก็บประวัติการเปลี่ยนแปลงใน project
3. ใช้คำสั่ง git pull origin **branch\_training\_nickname**
4. ใช้คำสั่ง git push -u origin **branch\_training\_nickname** เพื่อส่งการเปลี่ยนแปลงไปยัง gitlab
5. ตรวจสอบการเปลี่ยนแปลงใน Repository -> File -> **branch\_training\_nickname** -> nickname
   1. Go to Web Git: <http://ta-git-app1.bdms.co.th/users/sign_in>
   2. Login Username/Password: crmft01/crmft01
   3. Click Project “training/TrainingGitLab”



* 1. Click Repository -> Files -> **branch\_training\_nickname**



1. Copy ไฟล์ชื่อ .gitignore ภายใต้ Folder “..\TrainingGitLab” ไปไว้ที่ “..\TrainingGitLab\nickname”
2. แก้ไขไฟล์ .gitignore โดยเพิ่มข้อความด้านล่าง จากนั้นบันทึกไฟล์

**# Exclude the build directory**

**untrack\_folder/\***

**# Exclude filename**

**untrack\_file\***

**# Exclude lib file**

**\*.dll**

1. ใช้คำสั่ง CD .. เพื่อกลับมาที่ Folder “..\TrainingGitLab\nickname”
2. ใช้คำสั่ง git rm -r --cached . เพื่อลบการเปลี่ยนแปลงที่เคย Track ใน Folder นั้นๆ
3. แก้ไขไฟล์ “untrack\_file.txt”, “untrack\_file\_in\_folder.txt” โดยเพิ่มข้อความด้านล่าง จากนั้นบันทึกไฟล์

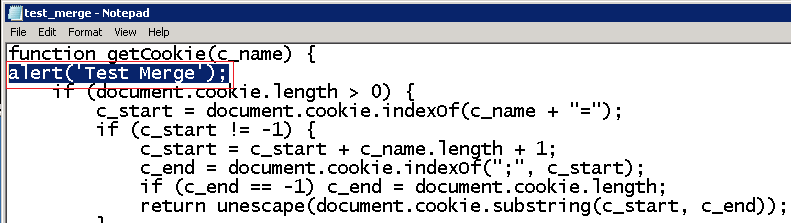
**#### ADD YOUR INFORMATION**

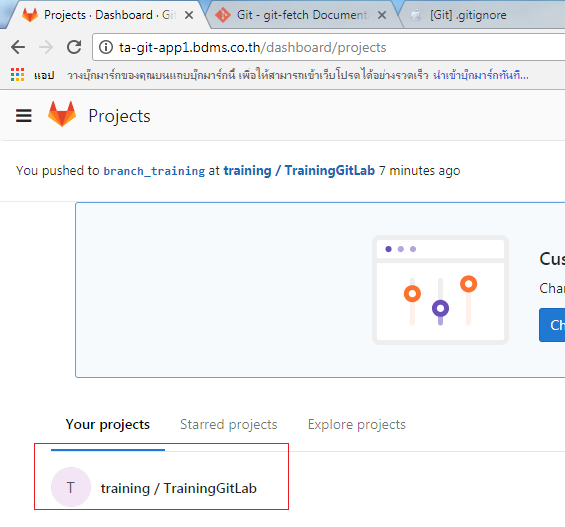
**#### Test** gitignore

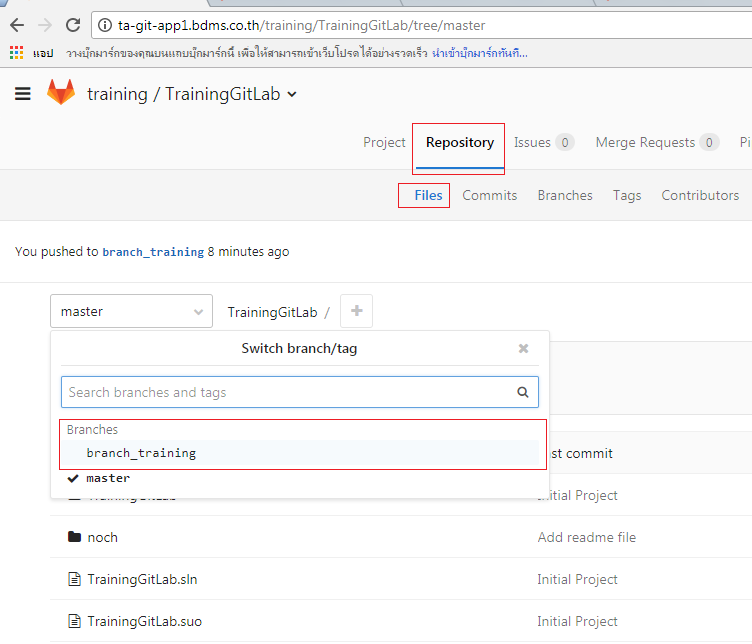
1. ใช้คำสั่ง CD .. เพื่อกลับมาที่ Folder “..\TrainingGitLab\nickname”
2. ใช้คำสั่ง git add ., git commit -m เพื่อเก็บประวัติการเปลี่ยนแปลงใน project
3. ใช้คำสั่ง git push -u origin **branch\_training\_nickname** เพื่อส่งการเปลี่ยนแปลงไปยัง gitlab
4. ใส่ Username/Password: crmft01/crmft01
5. ตรวจสอบการเปลี่ยนแปลงใน Repository -> File -> **branch\_training\_nickname** -> nickname

**Lab3: Edit Conflict File**

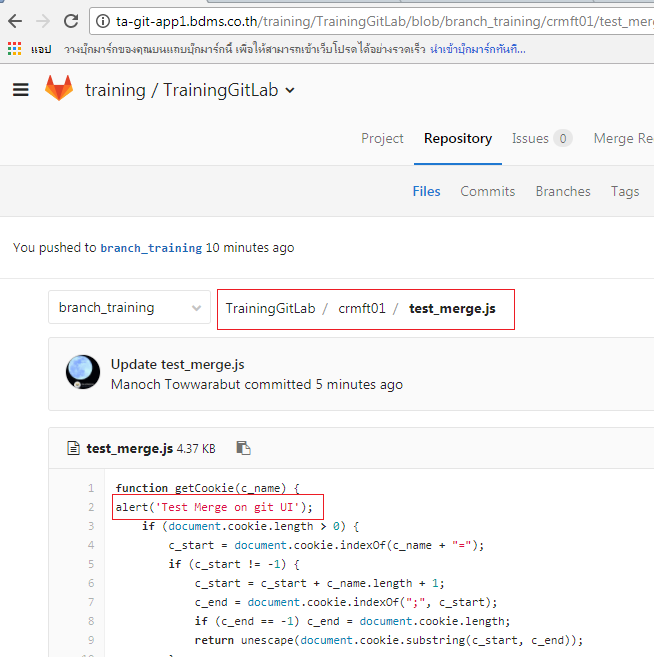
1. Copy ไฟล์จาก “..\TrainingGitLab\TrainingGitLab\Scripts\test\_merge.js” ไปที่ “..\TrainingGitLab\nickname\”
2. ใช้คำสั่ง CD เพื่อกลับมาที่ Folder “..\TrainingGitLab\nickname”
3. ใช้คำสั่ง git add ., git commit -m เพื่อเก็บประวัติการเปลี่ยนแปลงใน project
4. ใช้คำสั่ง git push -u origin **branch\_training\_nickname** เพื่อส่งการเปลี่ยนแปลงไปยัง gitlab
5. ใส่ Username/Password: crmft01/crmft01
6. แก้ไขไฟล์ “..\TrainingGitLab\nickname\test\_merge.js” ตามรูปด้านล่าง จากนั้นบันทึกไฟล์



1. Go to Web Git: <http://ta-git-app1.bdms.co.th/users/sign_in>
2. Login Username/Password: crmft01/crmft01
3. Click Project “training/TrainingGitLab”
4. Click Repository -> Files -> **branch\_training\_nickname**



1. Click TrainingGitLab -> “nickname” -> test\_merge.js และแก้ไขไฟล์ตามรูป



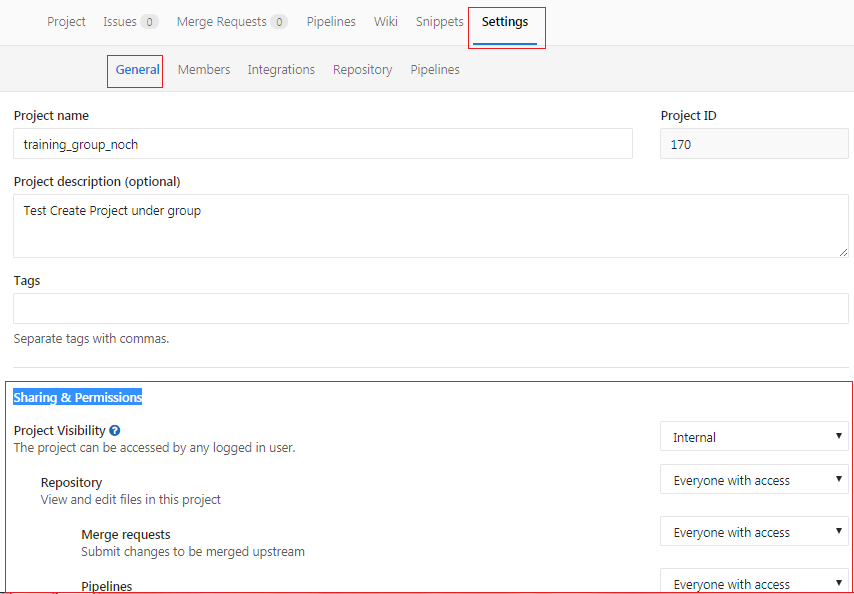
1. Click Commit Change
2. กับมาที่ Git Command Line
3. ใช้คำสั่ง git add ., git commit -m เพื่อเก็บประวัติการเปลี่ยนแปลงใน project
4. ใช้คำสั่ง git push -u origin **branch\_training\_nickname** เพื่อส่งการเปลี่ยนแปลงไปยัง gitlab
5. ตรวจสอบผล Reject
6. ใช้คำสั่ง git pull origin **branch\_training\_nickname**
7. ตรวจสอบผล Conflict
8. แก้ไข Conflict ที่ไฟล์ “..\TrainingGitLab\nickname\test\_merge.js”
9. ใช้คำสั่ง git add ., git commit -m เพื่อเก็บประวัติการเปลี่ยนแปลงใน project
10. ใช้คำสั่ง git push -u origin **branch\_training\_nickname** เพื่อส่งการเปลี่ยนแปลงไปยัง gitlab

**Lab4:**

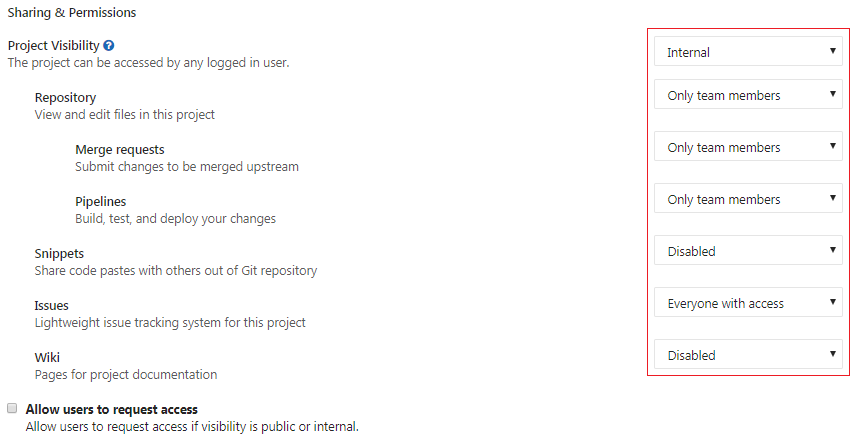
1. Go to Web Git: <http://ta-git-app1.bdms.co.th/users/sign_in>
2. Login Username/Password: your\_user\_ad/ your\_pass\_ad
3. ให้สร้าง Group ชื่อ “training\_group\_nickname”
4. Visibility Level -> Internal
5. Click Create group

**Lab5:**

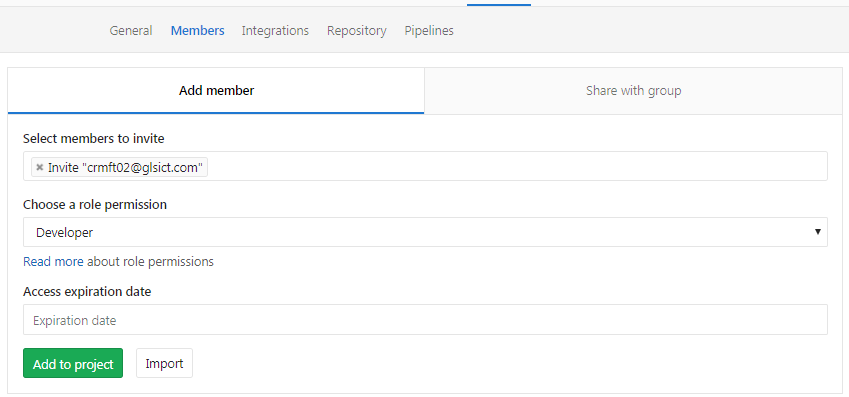
1. จาก Lab4 ให้สร้าง Project ชื่อ “training\_project\_nickname” ภายใต้ Group
2. Visibility Level -> Internal
3. Click Create project
4. Go to project Setting Tab -> General



1. Setting Sharing & Permissions



1. Click Save Change
2. Go to project Setting Tab -> Member
3. Select member to invite, Role: Developer and Click Add to project



1. Go to command line for Create a new repository
2. ให้ clone project จาก URL: [http://ta-git-app1.bdms.co.th/...(HTTP Your git project)](http://ta-git-app1.bdms.co.th/...(HTTP%20Your%20git%20project))
3. ใช้คำสั่ง CD ไปที่ Folder (Folder root clone project)
4. สร้างไฟล์ชื่อ “README.md”
5. แก้ไขไฟล์ “README.md” โดยเพิ่มข้อความด้านล่าง จากนั้นบันทึกไฟล์

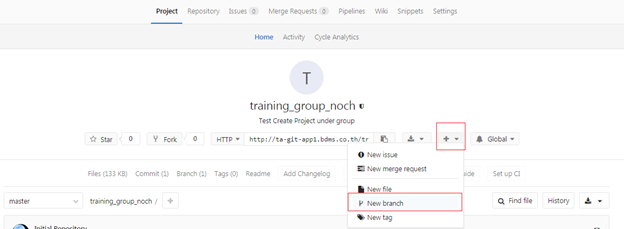
**#### ADD YOUR INFORMATION**

**#### Hello Git**

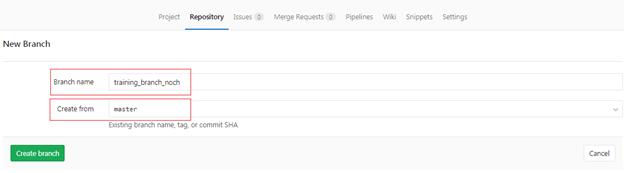
1. ใช้คำสั่ง git add, git commit -m เพื่อเก็บประวัติไฟล์ “README.md”
2. ใช้คำสั่ง git status เพื่อตรวจสอบการเปลี่ยนแปลง
3. ใช้คำสั่ง git push -u origin master เพื่อส่งการเปลี่ยนแปลงไปยัง gitlab

**Lab6:**

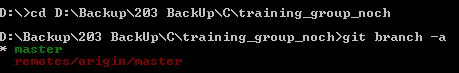
1. จาก Lab5 ให้กับมาที่ Project -> เลือกตามรูปด้านล่างเพื่อสร้าง New Branch



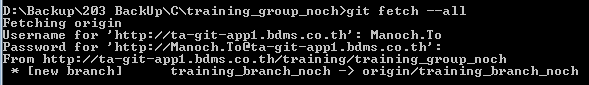
1. กรอกข้อมูลชื่อ Branch “training\_branch\_nickname” และเลือก Create from: master



1. Click ปุ่ม Create branch
2. Go to command line
3. ใช้คำสั่ง git branch –a เพื่อดูรายการ branch ใน Local Computer



1. เนื่องจากเป็น branch ที่พึงถูกสร้างใน remote repository เราจึงไม่เห็น branch ที่สร้าง ให้ใช้คำสั่ง git fetch --all เพื่อ Download



1. ใช้คำสั่ง git checkout **training\_branch\_nickname** เพื่อ switch ไปที่ branch “training\_branch\_nickname”
2. แก้ไขไฟล์ “README.md” โดยเพิ่มข้อความด้านล่าง จากนั้นบันทึกไฟล์

**#### ADD YOUR INFORMATION**

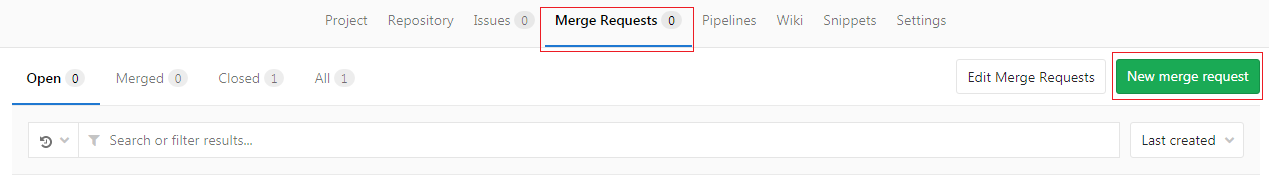
**#### Hello Git**

**#### Test Create New Branch**

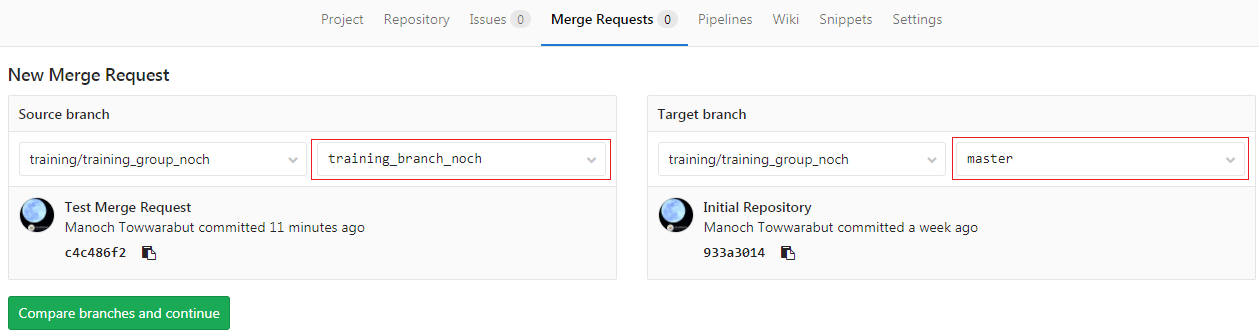
1. ใช้คำสั่ง git add, git commit -m เพื่อเก็บประวัติไฟล์ “README.md”
2. ใช้คำสั่ง git push -u origin **training\_branch\_nickname** เพื่อส่งการเปลี่ยนแปลงไปยัง gitlab

**Lab7:**

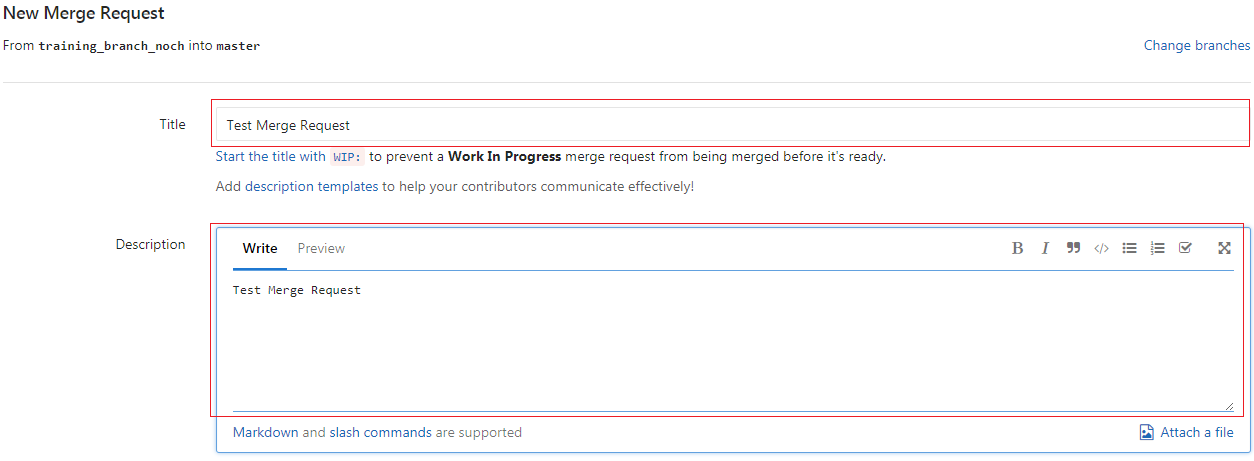
1. จาก Lab6 ให้กับมาที่ Project -> Merge Requests -> Click New merge request เพื่อสร้าง New Merge Request

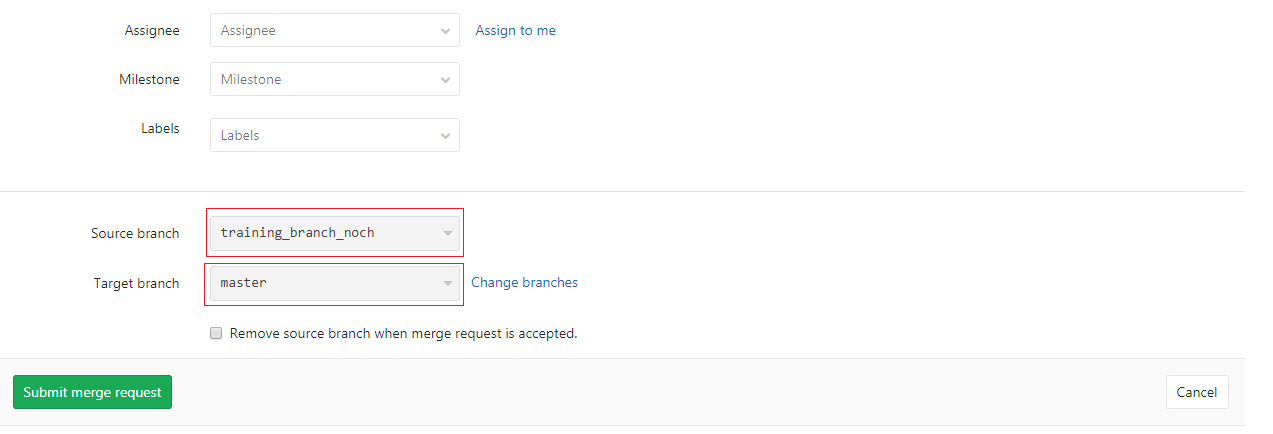


1. เลือก Source branch: branch ต้นทางที่จะขอนำ Source ไป Merge , Target: branch ปลายทางที่จะขอนำ Source มา Merge



1. Click compare branch and continue
2. ใส่ Title , Description ที่เกี่ยวข้องกับการขอ Merge Request ในครั้งนั้น

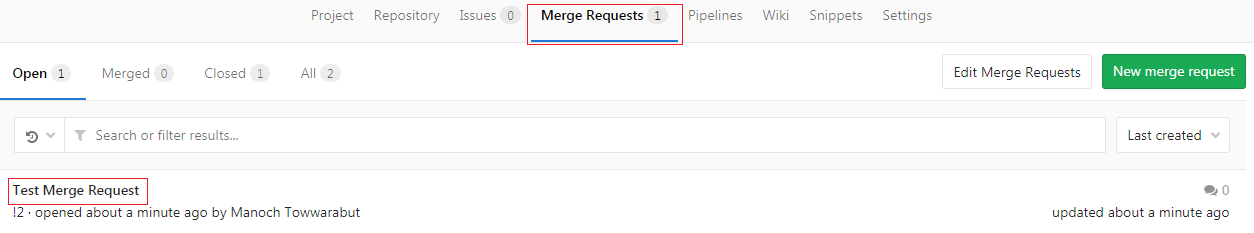




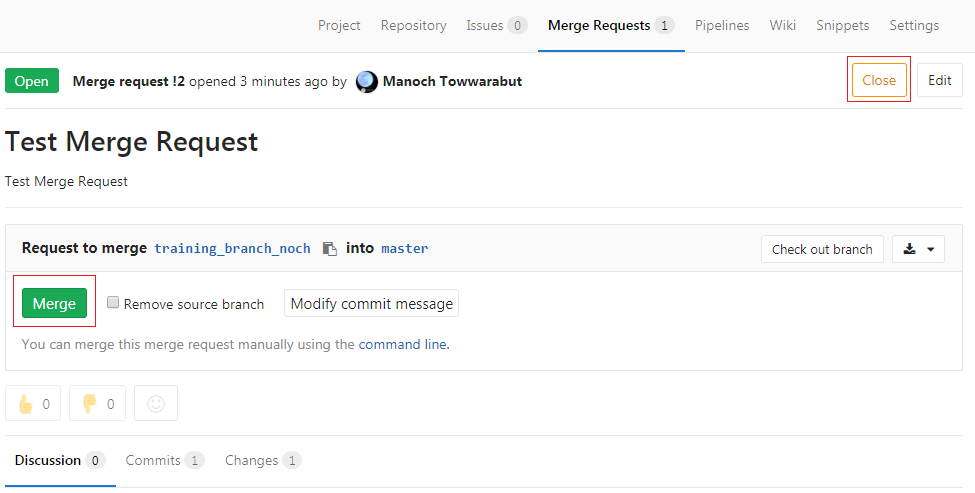
1. Click Submit merge request

**Lab8:**

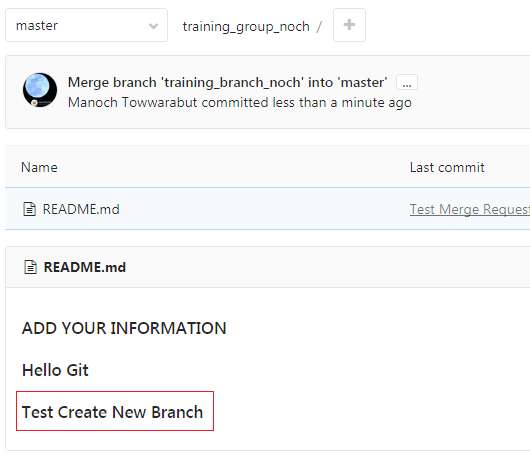
1. จาก Lab7 ให้กับมาที่ Project -> Merge Request -> Click หัวข้อรายการ Merge Request ตามรูปด้านล่างเพื่อ Merge, Close Merge Request



1. Click Merge เพื่อยอมรับการ Merge Request นั้นๆ หรือ Click Close เพื่อปิด Merge Request นั้นๆ โดยไม่ยอมรับ

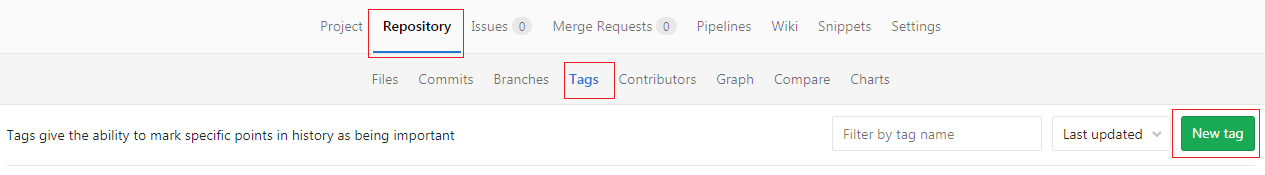


1. กรณี Click Merge ให้กับมาที่ Branch Master เพื่อตรวจสอบการเปลี่ยนแปลง

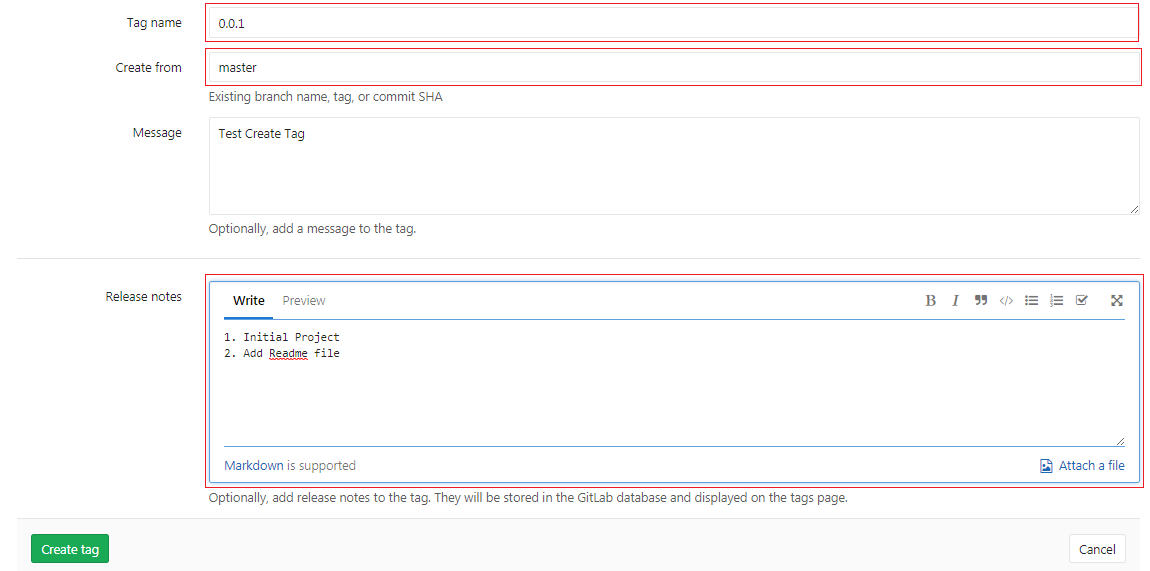


**Lab9:**

1. จาก Lab8 ให้กับมาที่ Project -> Repository -> Tags -> Click New Tag ตามรูปด้านล่างเพื่อสร้าง Tag



1. ใส่ Tag name (Running ตามข้อกำหนดมาตฐาน), Create from: branch ที่ต้องการ Tag, Release notes: รวมรายละเอียดการแก้ไขโปรแกรมของการ Tag ครั้งนั้น



1. Click Create tag