



2. เข้าไปที่ c:\Users\ชื่อเครื่องเรา\.ssh
  - a. เปิดไฟล์ .pub and copy key
  - b. และนำไปวางที่ website gitlab โดยอยู่ที่ setting -> SSH Keys และวาง key และ กด Add Key

**Key**

Paste your public SSH key, which is usually contained in the file '~/.ssh/id\_ed25519.pub' or '~/.ssh/id\_rsa.pub' and begins with 'ssh-ed25519' or 'ssh-rsa'. Don't use your private SSH key.

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQAC07EPRD4S4IDIXO6GVEVW6oJoWp6GgF9IWeAviN9hcx
g6nXUOJc7jkEyqqvp3UAQ72EpSWQCpbO+BITO4mAwTQZyp+bV9f21lj7cZEZT7Pdg+sPMliqlctVJY
OSc72ljgXXwxtPsgNRS1bjJyYin29Kd+v25qWuNHeuqidSiAO6LsrCluzNvDUR6kypbKkkgw7jxLCr/nPr
3WTjjWUYS07VTHBvtBv6B4Nb6+sfzTy/R+VnUy09mDc0n8hyb53jf8+6q/hy0FSWeo9o+JJ5olpoyQ
kzmvNMgIMZTaf2IKf6NkjF1dstL+sYGuvAQZM1sawCursG500EkArgTxVpKVP
macbook@PAKORN_MACBOOK
```

**Title**

Name your individual key via a title

**Add key**

---

**Your SSH keys (0)**

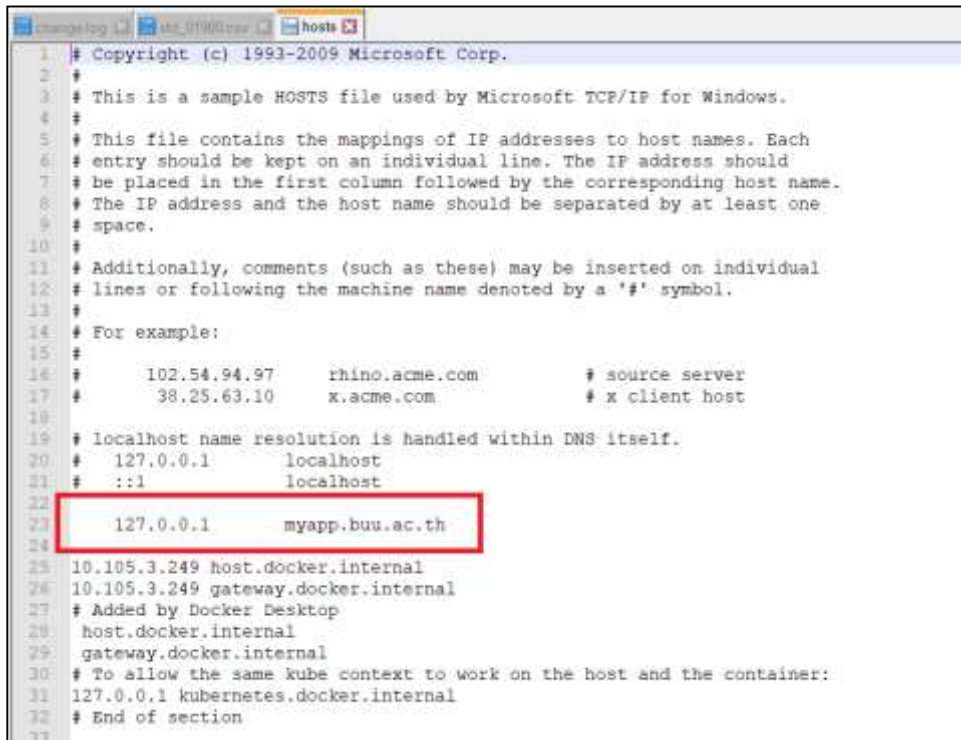
There are no SSH keys with access to your account.

## สำหรับทุกคนในทีม

Set hosts on windows

เปิดไฟล์ C:\Windows\System32\drivers\etc\hosts

1. พิมพ์ 127.0.0.1 myapp.buu.ac.th
2. ทำให้เราสามารถ เรียก app ที่เราสร้างผ่าน browser ได้

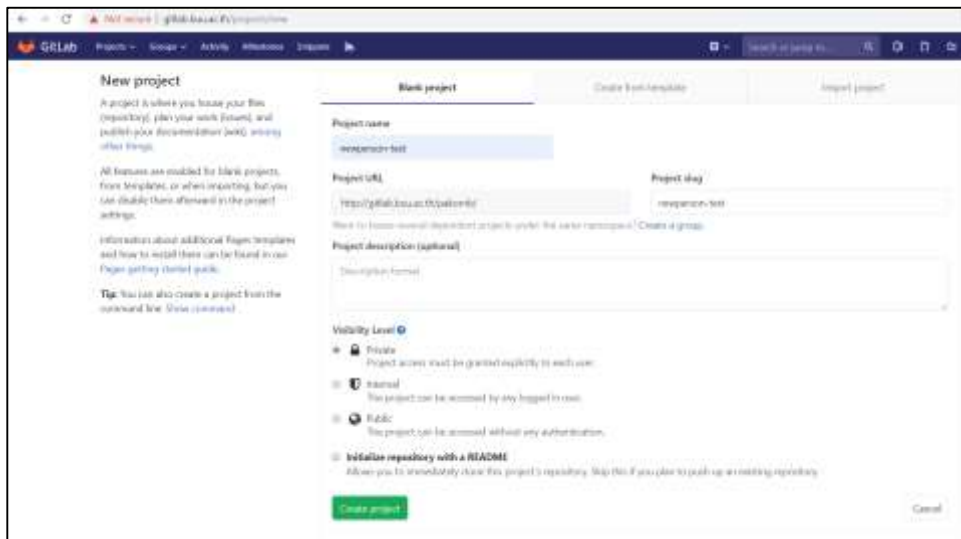


```
1 # Copyright (c) 1993-2009 Microsoft Corp.
2 #
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4 #
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #       102.54.94.97       rhino.acme.com       # source server
17 #       38.25.63.10      x.acme.com           # x client host
18 #
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1       localhost
21 #   ::1             localhost
22
23 127.0.0.1       myapp.buu.ac.th
24
25 10.105.3.249 host.docker.internal
26 10.105.3.249 gateway.docker.internal
27 # Added by Docker Desktop
28 host.docker.internal
29 gateway.docker.internal
30 # To allow the same kube context to work on the host and the container:
31 127.0.0.1 kubernetes.docker.internal
32 # End of section
33
```

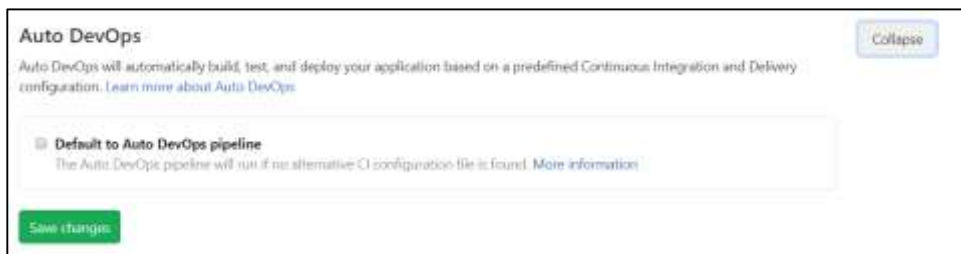
## สำหรับหัวหน้าทีม (Leader)

1. New project with
  - a. กรณี เป็น Project ใหม่
    - i. พิมพ์คำสั่ง นี้ ที่ cmd  

```
docker run --rm -v %cd%:/app composer create-project --prefer-dist laravel/laravel myproject-name
```
    - ii. เป็นคำสั่ง สร้าง source code ของ template laravel
  - b. กรณีที่มี Project อยู่แล้ว ก่อนที่จะมาสร้าง GitLab
2. Create New Repository on <http://gitlab.buu.ac.th>
  - a. คือการสร้าง project ใหม่



- b. เมื่อสร้างเสร็จแล้ว Disable autodevops
  - i. ไปที่ project -> setting -> CI/CD ทำตามรูปเลย (เอาเครื่องหมายถูกออก) และกด save change



c. Run คำสั่ง จะได้

```
Git global setup
git config --global user.name "Pakorn Burkungsan"
git config --global user.email "pakorn@buu.ac.th" 1

Create a new repository
git clone git@gitlab.buu.ac.th:pakorn/newperson-test.git
cd newperson-test
touch README.md
git add README.md
git commit -m "add README"
git push -u origin master 2

Push an existing folder
cd existing_folder
git init
git remote add origin git@gitlab.buu.ac.th:pakorn/newperson-test.git
git add .
git commit -m "initial commit"
git push -u origin master 3

Push an existing Git repository
cd existing_repo
git remote rename origin old-origin
git remote add origin git@gitlab.buu.ac.th:pakorn/newperson-test.git
git push -u origin --all
git push -u origin --tags 4
```

หมายเลข 1 คือ เวลา commit ระบบเราจะใช้ชื่อนี้ในบันทึกลงใน history git จะทำก็ต่อเมื่อเครื่องคอมพิวเตอร์ของท่าน ไม่เคยใช้คำสั่งนี้มาก่อน แต่ถ้าเคยใช้คำสั่งนี้มาแล้ว ก็ไม่ต้องทำ เพราะทำแค่ครั้งแรกเพียงครั้งเดียว

พิมพ์คำสั่ง หมายเลข 1 ที่ cmd

หมายเลข 2 คือ สร้าง repository ใหม่เลย

หมายเลข 3 คือ กรณีที่เครื่องคอมพิวเตอร์ตัวเอง มี sourcecode ในโฟลเดอร์แล้ว

หมายเลข 4 คือ กรณีที่เครื่องคอมพิวเตอร์ตัวเอง มี sourcecode และโฟลเดอร์ .git ในโฟลเดอร์แล้ว

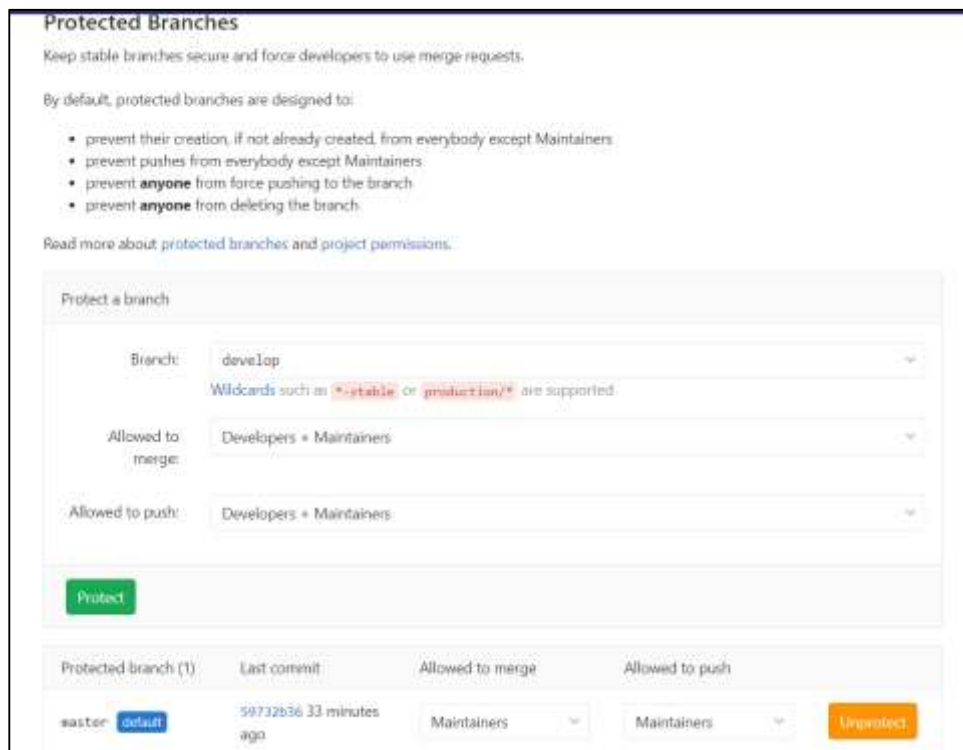
3. Git push project to repository

a. เป็นการเพิ่ม project ไปยัง repository ที่ gitlab.buu.ac.th

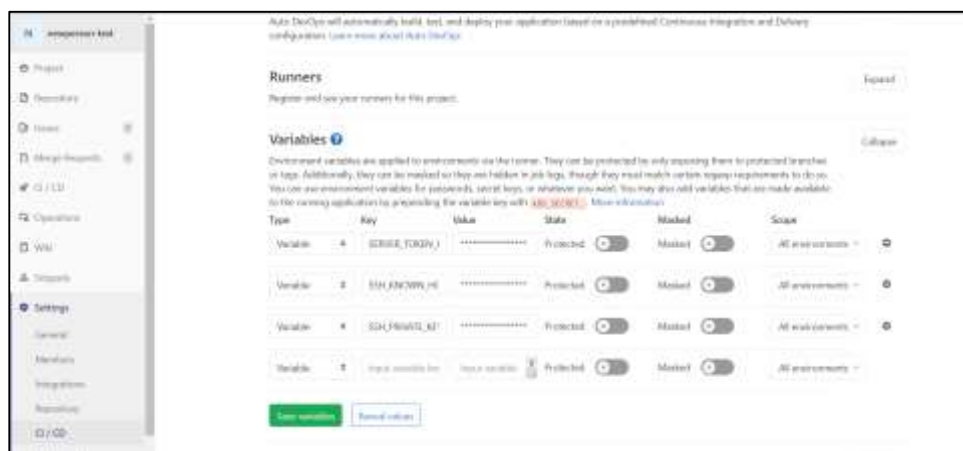
b. ในกรณีนี้ เป็นระบบ newperson จึงใช้ขั้นตอน ของ หมายเลข 3 (ลบโฟลเดอร์ .git ก่อน)



6. Set Protected branch ให้กับ branch "develop"  
ให้ Leader set สิทธิการใช้งานให้กับแต่ละ branch (Settings -> Respository)



7. Set variabel on Repository
  - a. สร้างตัวแปร สำหรับการทำงานของ Pipeline
  - b. Settings -> CI / CD -> Variables



8. Add team Member as developer role
  - a. Leader set สิทธิการใช้งานให้กับ branch "develop" เพื่อให้สมาชิกสามารถใช้งาน branch นี้ได้
  - b. (Settings -> Members)

### สำหรับนักพัฒนาในทีม

1. Clone project จาก gitlab.buu.ac.th
  - a. Git clone -b develop git://path/to-repository
  - b. เช่น git clone git@gitlab.buu.ac.th:pakornb/newperson-test.git
  - c. เป็นการ clone project มาไว้ที่เครื่องเรา โดยมันจะสร้างโฟลเดอร์ newperson-test
2. cd in project
  - a. เปิด cmd และ cd เข้าไปที่ project ที่เราสร้าง
  - b. เช่น cd newperson-test
3. การสร้าง ไฟล์ .env ใน folder project  
docker exec -it mydev php artisan key:generate
4. Install composer ที่เครื่องตัวเอง  
php composer.phar update
5. Run คำสั่ง Install composer ที่เครื่องตัวเอง  
docker run --rm --interactive --tty --volume %CD%:/var/www/html aekapop/buccdev  
composer install
6. Run docker เพื่อที่จะเปิดระบบ ผ่าน browser  
docker run -p 80:80 -p 443:443 -v %cd%:/var/www/html -d --name mydev aekapop/buccdev
  - i. คำสั่ง docker ps -a คือ แสดงว่ามี container อะไรอยู่บ้าง
  - ii. คำสั่ง docker rm ชื่อ container คือการลบ container  
เช่น docker rm mydev
  - iii. คำสั่ง docker stop ชื่อ container  
เช่น docker stop mydev
7. **กรณี developer ใช้งานครั้งแรก**
  - a. ตรวจสอบว่า มีการ config user.name และ user.email แล้วหรือยัง
    - i. ใช้คำสั่ง git config --list
    - ii. ถ้ายังไม่ได้ config ให้ทำ ข้อ b เลย
    - iii. แต่ถ้าทำแล้ว ไม่ต้องทำอะไรเลย
  - b. ให้เปิด cmd เพื่อ run คำสั่งต่อไปนี้ เพื่อตั้งค่าการใช้งานตอน commit code ไปที่ git
    - i. ตั้งชื่อผู้ใช้งาน  
git config --global user.name "ชื่อตัวเอง"
    - ii. ตั้งอีเมลของผู้ใช้งาน



- git config --global user.email "ชื่ออีเมล"
- c. การทำให้ atom สามารถตรวจ standard ของ source code ได้  
เปิดไฟล์ php.ini
  1. ไปที่บรรทัดสุดท้าย เพิ่ม extension=php\_openssl.dll
  2. ค้นหาคำว่า "extension\_dir" ลบ ; ออก (เป็นการเปิดให้ใช้งาน)
  3. เปิด cmd พิมพ์คำสั่ง  
composer global require "squizlabs/php\_codesniffer=\*"
- d. การ set ให้ line-ending เป็น LF
  - i. เปิด atom และเลือก File -> Settings -> Install Packages
  - ii. ค้นหา line-ending-selector และ install
  - iii. เมื่อ install เสร็จ ให้ click setting
  - iv. ที่ Default line ending เป็น LF

### การสร้างและรวม Branch และการตรวจสอบคุณภาพซอร์สโค้ด

#### 1. Git branch feature-name

- a. เป็นการสร้าง branch ใหม่
- b. กรณีเมื่อมีการจะพัฒนาโปรแกรมใหม่ ต้องสร้าง branch ใหม่ทุกครั้ง
- c. เช่น git branch feature-personinfo เป็นต้น
- d. เทคนิคการสร้าง branch
  - i. **feature/xxxx** : เมื่อเราต้องการจะทำงานตาม Feature ที่เราได้รับ เช่น User Login, Edit User, Add Member Button เป็นต้น
  - ii. **chore/xxx** : เมื่อเราต้องการจะเริ่มทำ Task ที่มันหยิบย่อย ต่างๆ ไม่ได้อยู่ใน User Story หรือ Planning
  - iii. **task/xxx** : เมื่อเราต้องการแตก Task แยกย่อยออกมาจาก User Story หรือจริงๆ สามารถใช้แบบเดียวกับ chore/xxx ได้ ซึ่งข้อนี้เป็น (Optional)
  - iv. **bug/xxx** หรือ **issue/xxx**: เมื่อเราได้รับ Bug หรือ Issue จาก Tester
  - v. **hotfix/xxx** : เมื่อเราได้รับ Bug หรือ Issue ที่เกิดตอน Production แล้ว ซึ่ง Branch นี้จะแตกต่างจากชั้นอื่น คือจะ Hotfix จาก Branch master เนื่องจากโค้ดของเราที่ขึ้น Production คือ master และเมื่อทำการ Hotfix เรียบร้อยแล้วก็ต้อง Merge กลับมาทั้ง maste และ dev ผ่านการ Pull Request เหมือนปกติ
  - vi. **release/v1.x.x** : สำหรับ release version ขึ้น Production

#### 2. Git branch -a

- a. การตรวจสอบว่า ณ ตอนนี้ เราอยู่ที่ branch ไหน
- b. เช่น git branch -a

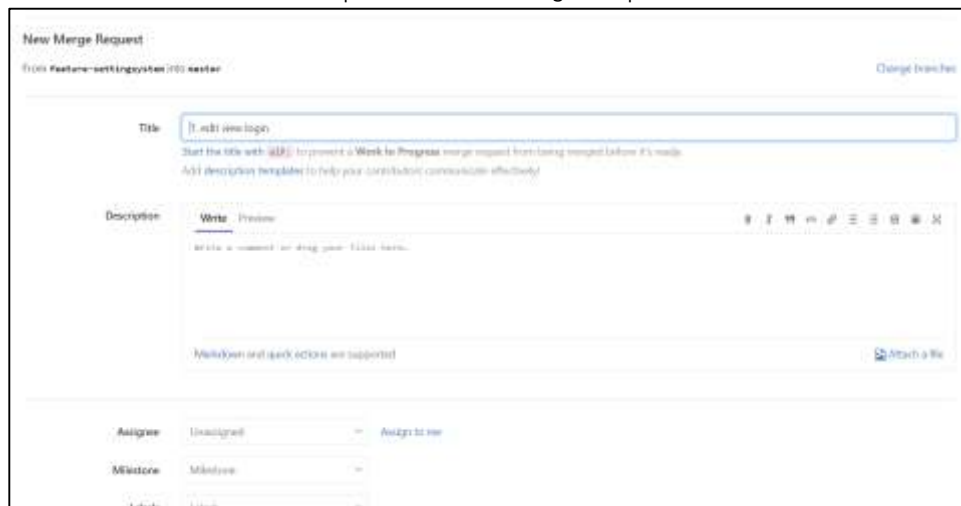
3. Git checkout feature-name
  - a. คือการ เปลี่ยนมาใช้ branch feature-name
  - b. เช่น git checkout feature-personinfo
4. Git branch -d feature-name
 

คือการ ลบ branch feature-name
5. Edit code and test in local sever
 

เป็นขั้นตอนการ พัฒนาโปรแกรมและทดสอบบน local server ที่เครื่องของเรา
6. Git add
 

เป็นการคำสั่งที่ใช้เพิ่มการเปลี่ยนแปลงของไฟล์ source code ที่มีการแก้ไข เข้าไปที่สถานะ Staged Change (เพื่อรอ commit ขึ้น git )
7. Git commit -m "commit name"
 

เป็นการยืนยัน source code ที่อยู่ที่สถานะ Staged Change เข้าไปเก็บไว้ที่ local Repository
8. Git push origin feature-name
9. Create merge request feature-name to develop
  - a. คลิกที่เมนูด้านซ้าย ชื่อ Merge Requests
  - b. คลิกปุ่ม create merge request
  - c. เขียน comment ที่ช่อง description ถึงการ merge request ครั้งนี้



- d. ตรวจสอบ source branch และ target branch ให้ถูกต้อง
  - i. Source branch คือ branch ที่เราสร้างขึ้นใหม่
  - ii. Target branch คือ develop
- e. คลิกปุ่ม compare branches and continue

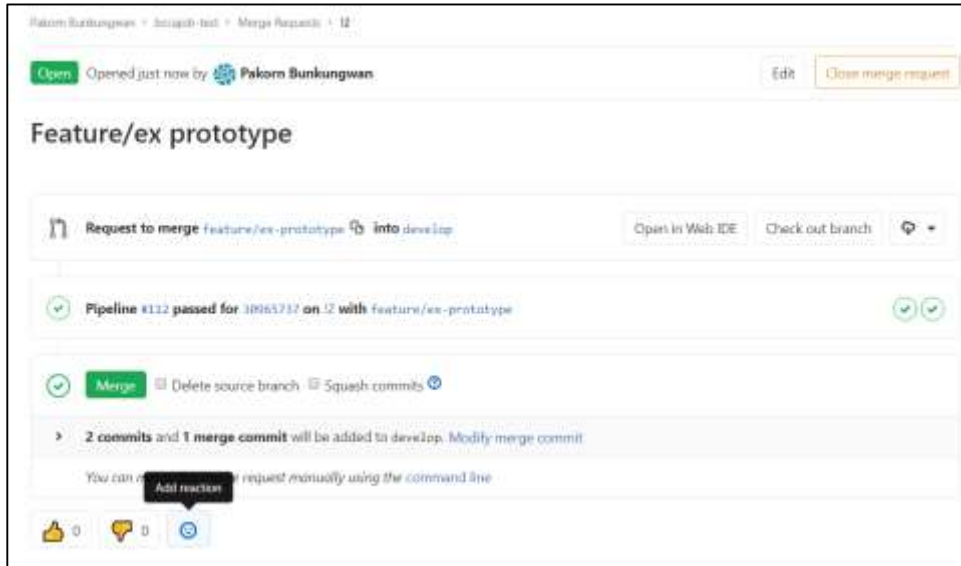


f. คลิกปุ่ม submit merge request

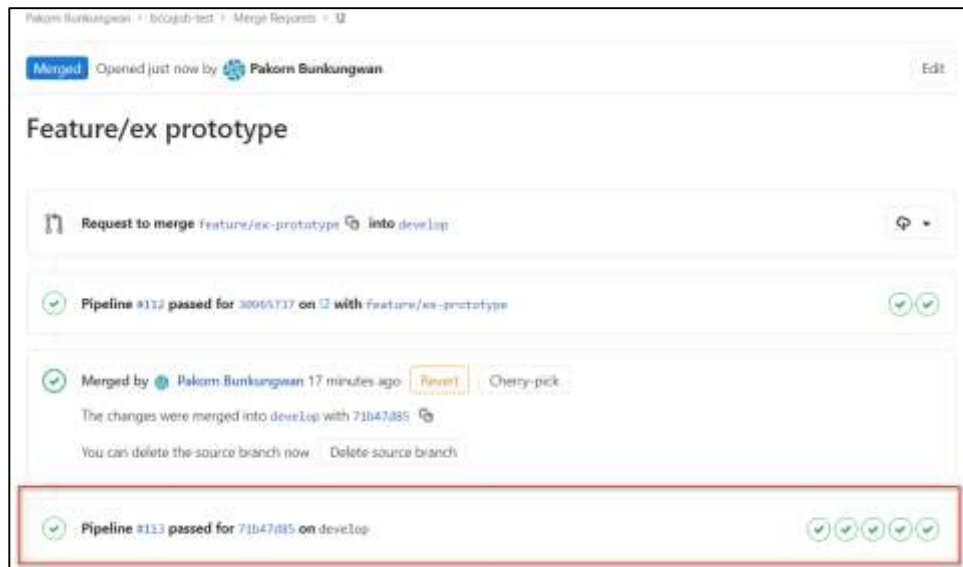


g. คลิกปุ่ม Merge

จะทำการตรวจสอบ Pipeline ถ้าผ่านหมด จะทำการ merge เข้ากับ branch develop



h. เมื่อตรวจสอบ pipeline เสร็จแล้ว จะแสดงเป็น เครื่องหมายลูกสีเขียวทั้งหมด



การตรวจสอบการทำงานของ pipeline

1. กรณีถ้าเกิด error ให้อัปเดต packages ต่าง ด้วย composer เช่น ให้อัปเดต version ของ laravel/framework
2. ให้ใช้คำสั่ง
  - a. เข้าไปที่ path ของ folder ของโปรแกรมนั้น
  - b. พิมพ์ `docker exec -it ชื่อ_CONTAINER_ID /bin/bash`  
เช่น `docker exec -it 03c088d6bad7 /bin/bash`
  - c. หน้าจอจะแสดงว่า `root@ชื่อ_CONTAINER_ID:/var/www/html#`  
พิมพ์คำสั่ง `composer update`
    - เป็นการอัปเดต package ต่าง ๆ ในระบบเราให้เป็นปัจจุบัน
  - d. เมื่อทำเสร็จแล้ว ไฟล์ `composer.lock` จะมีการเปลี่ยนแปลง และให้
    - Commit ไปที่ git
    - Push ขึ้น gitlab.buu.ac.th
  - e. หลังจากนั้นทำการตรวจสอบ pipeline อีกครั้ง