인하공업전문대학
INHA TECHNICAL COLLEGE

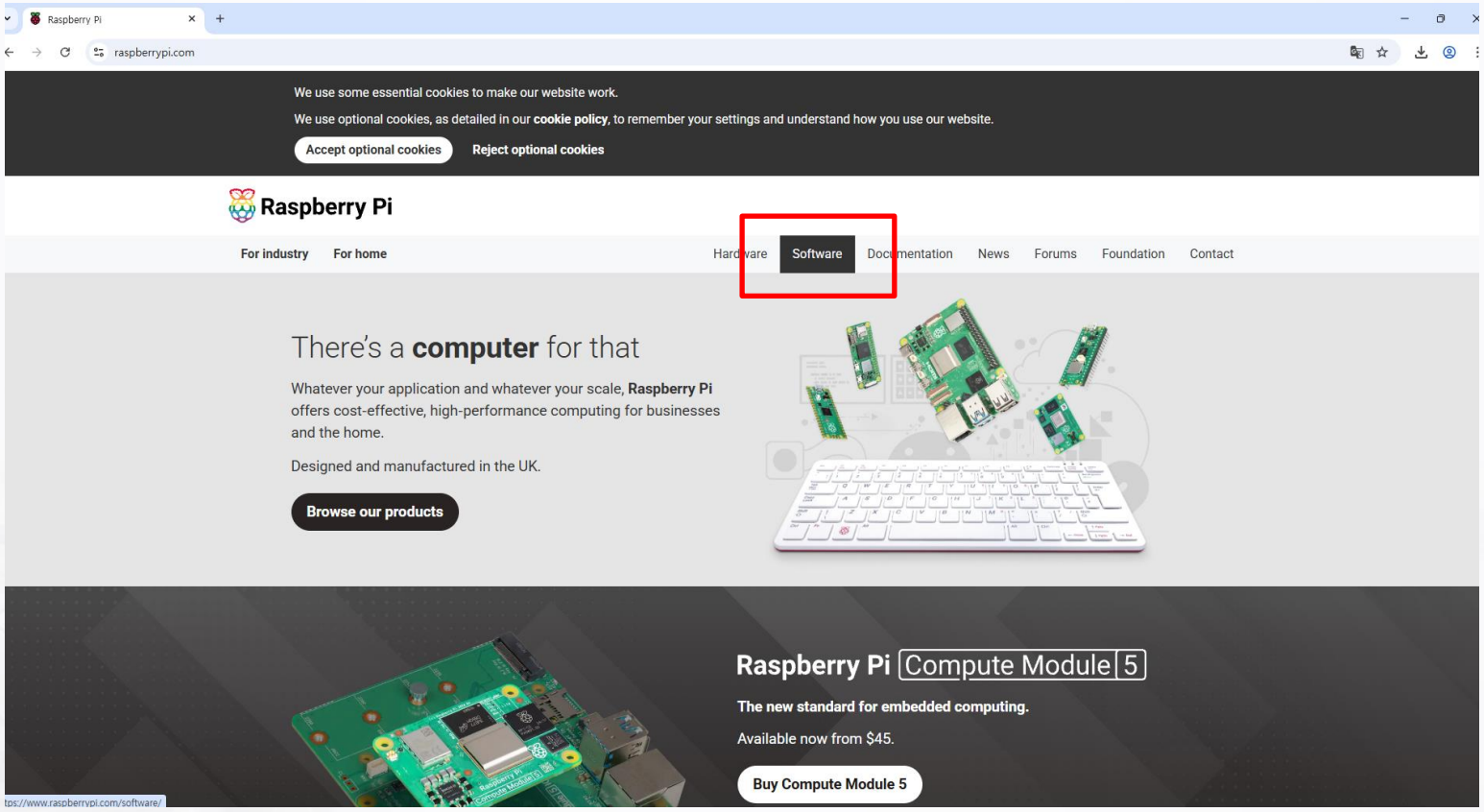
무선 네트워크
2주차

인하공업전문대학 컴퓨터 정보과

김한결

RasberryPi 설치

- 라즈베리파이 공식 사이트
  - ✓ https://www.raspberrypi.com/



- 라즈베리파이 공식 사이트
  - ✓ https://www.raspberrypi.com/

Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install **Raspberry Pi OS** and other operating systems to a microSD card, ready to use with your Raspberry Pi.

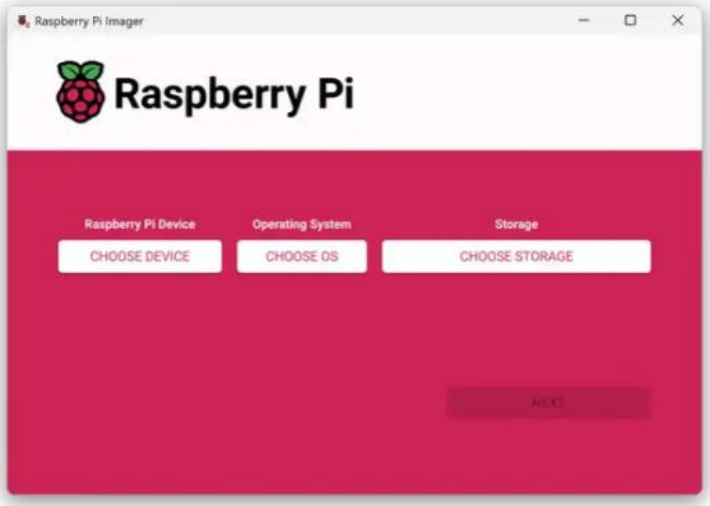
Download and install Raspberry Pi Imager on a computer with an SD card reader. Insert the microSD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

Download for Windows

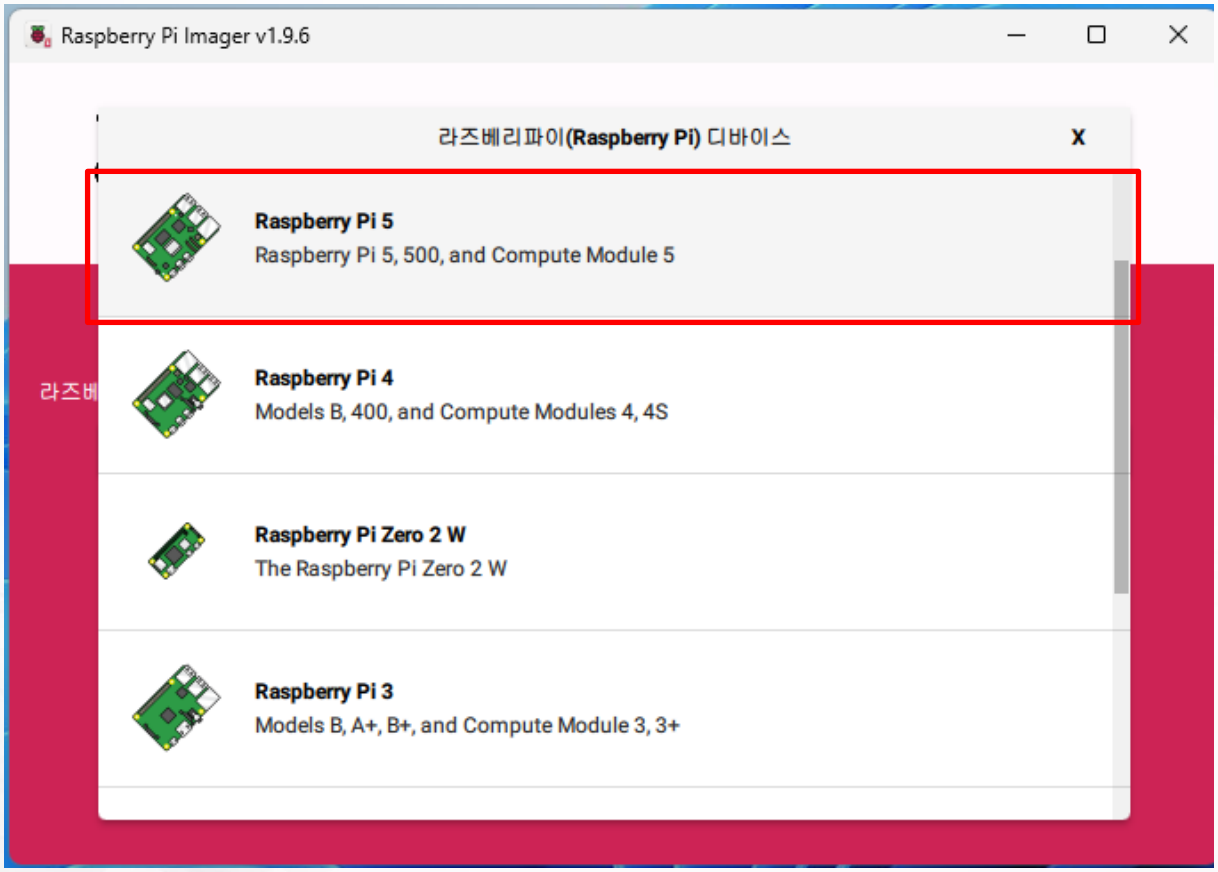
Download for macOS

Download for Debian or Ubuntu (x86_64)

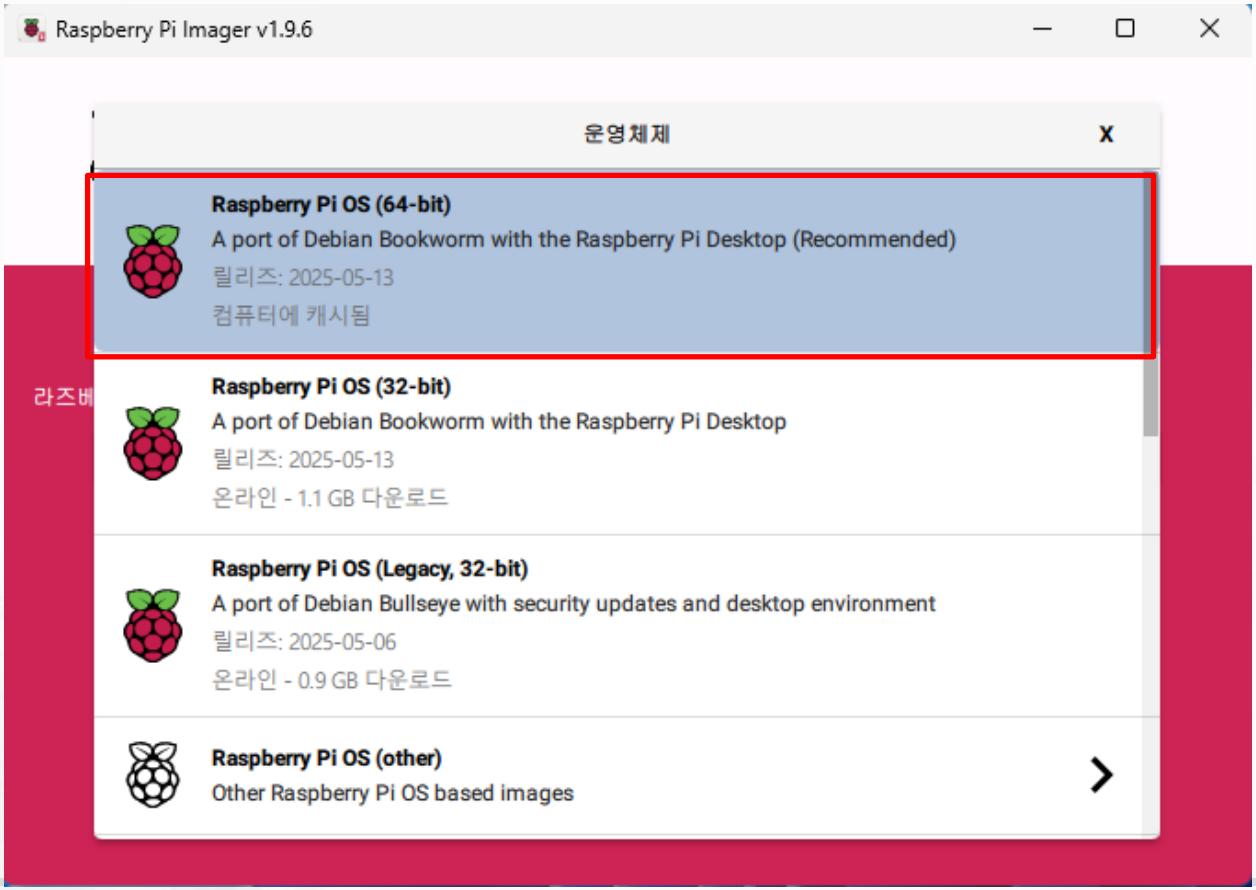
To install on **Raspberry Pi OS**, type `sudo apt install rpi-imager` into a terminal window



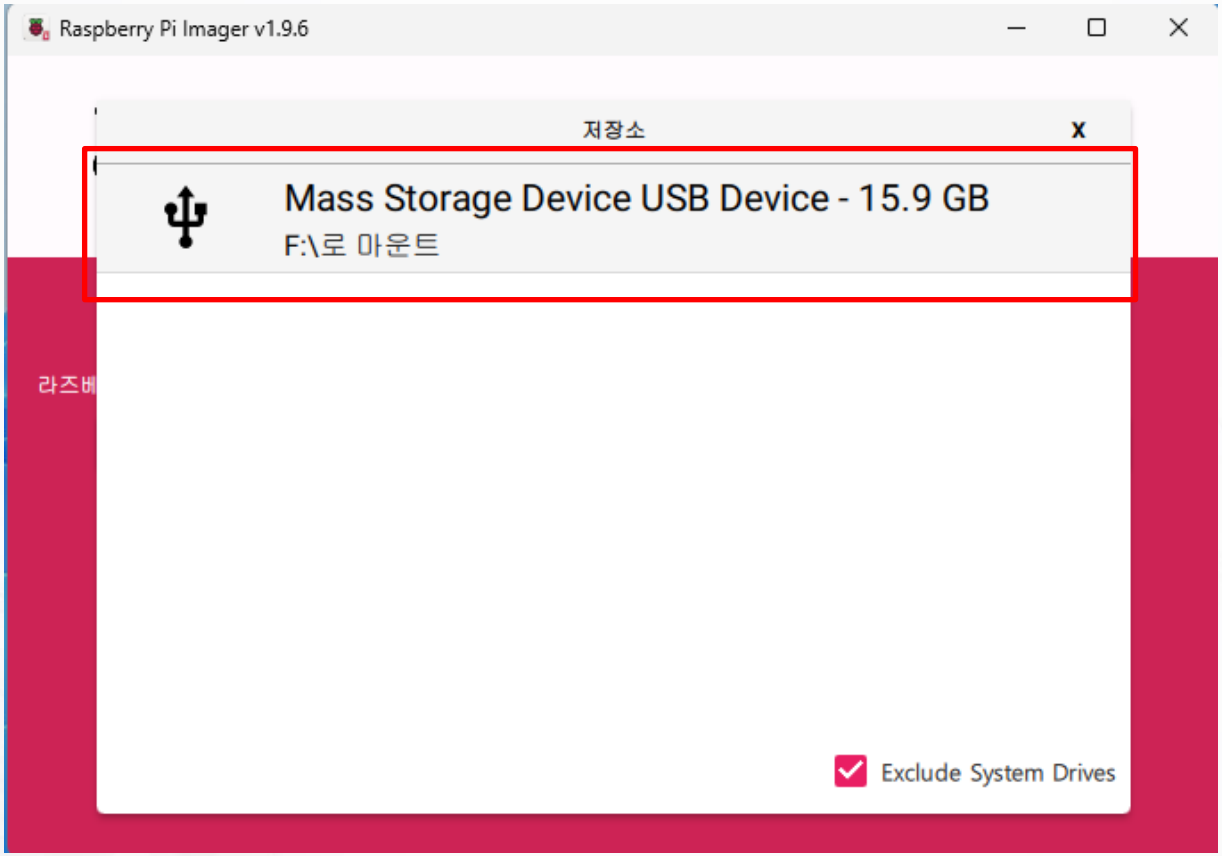
- 라즈베리파이 이미지 설치
  - ✓ 장치선택 (예 Raspberry pi 5)



- 라즈베리파이 이미지 설치
  - ✓ 운영체제 선택 (32bit or 64bit)



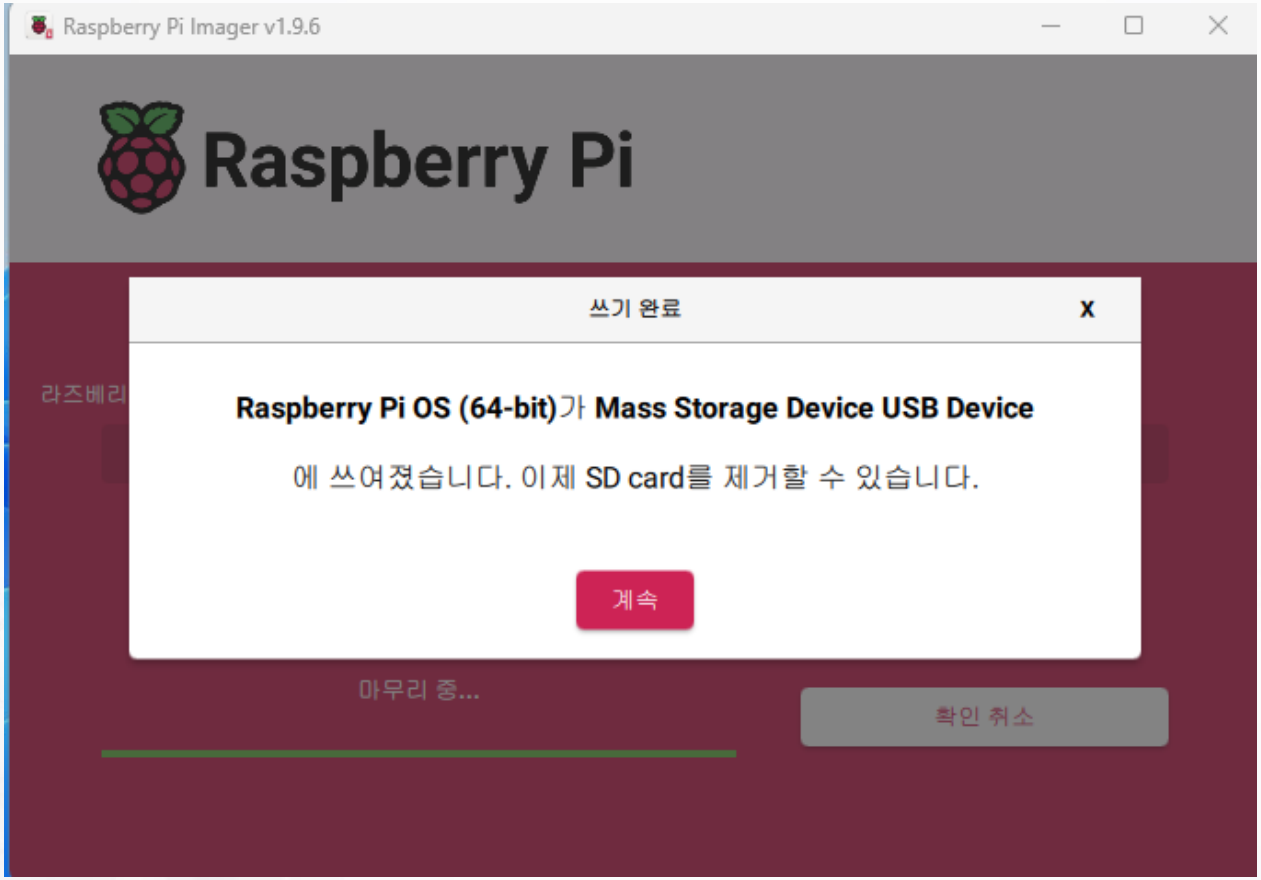
- 라즈베리파이 이미지 설치
  - ✓ 저장소 선택



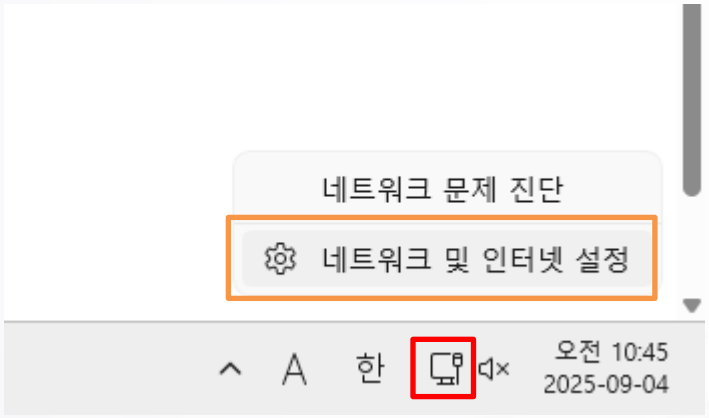
- 라즈베리파이 이미지 설치
  - ✓ 커스터마이징 설정 (Pass)

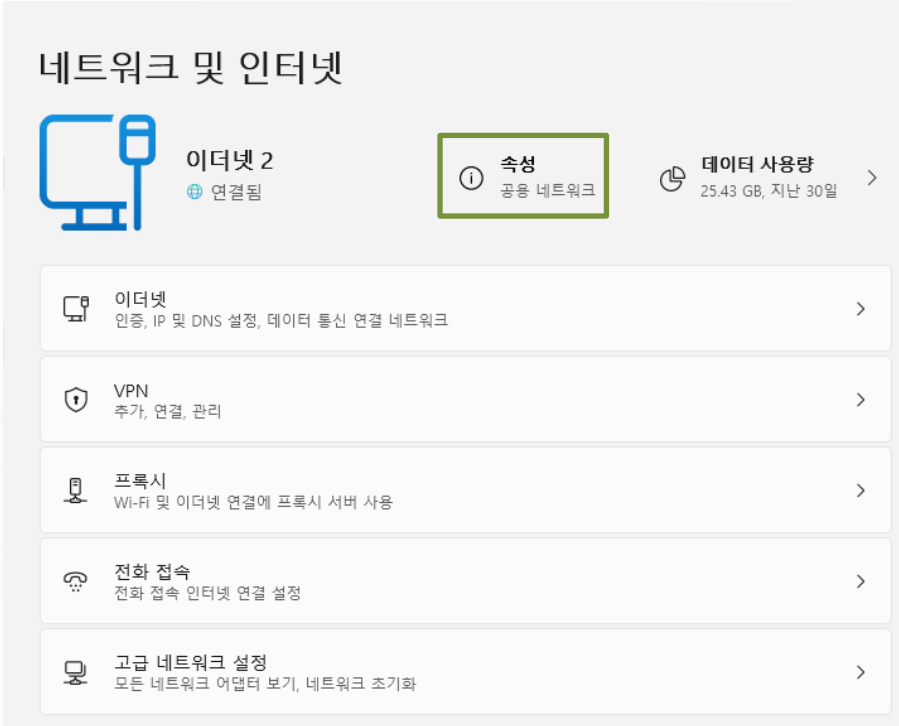
- 라즈베리파이 이미지 설치

- 라즈베리파이 이미지 설치
  - ✓ 완료



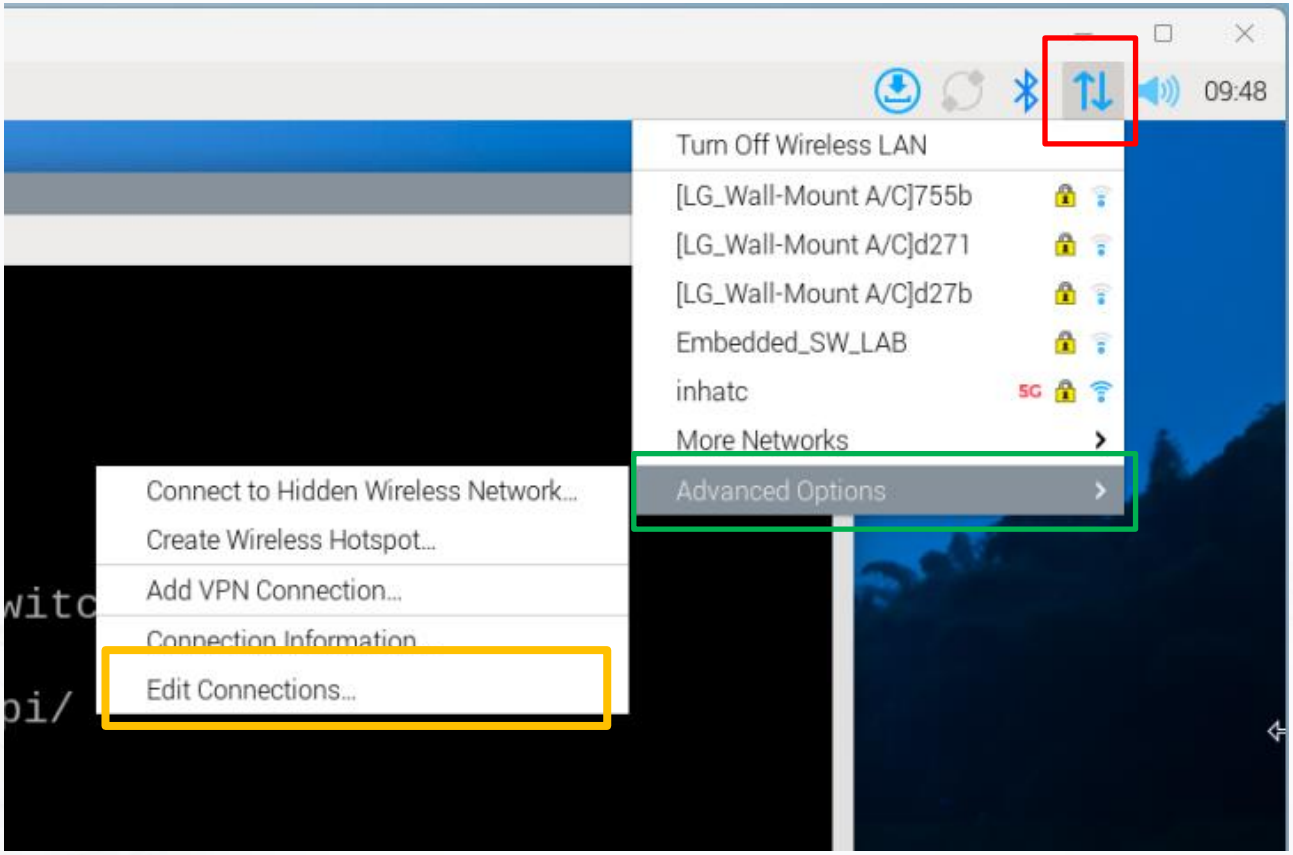
- 라즈베리파이 구동 전 확인
  - ✓ IP주소 확인



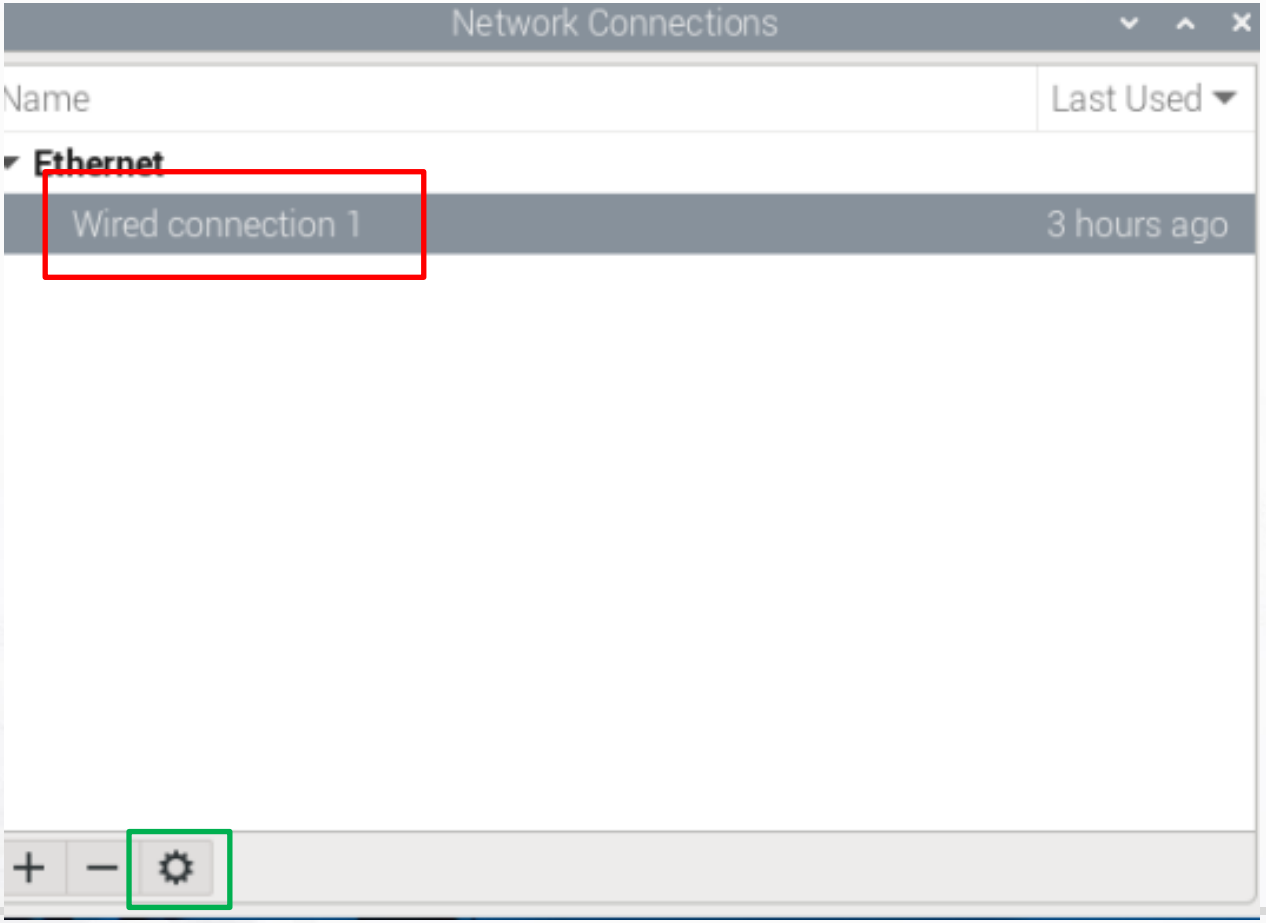


- 라즈베리파이 구동 전 확인
  - ✓ IP주소 확인

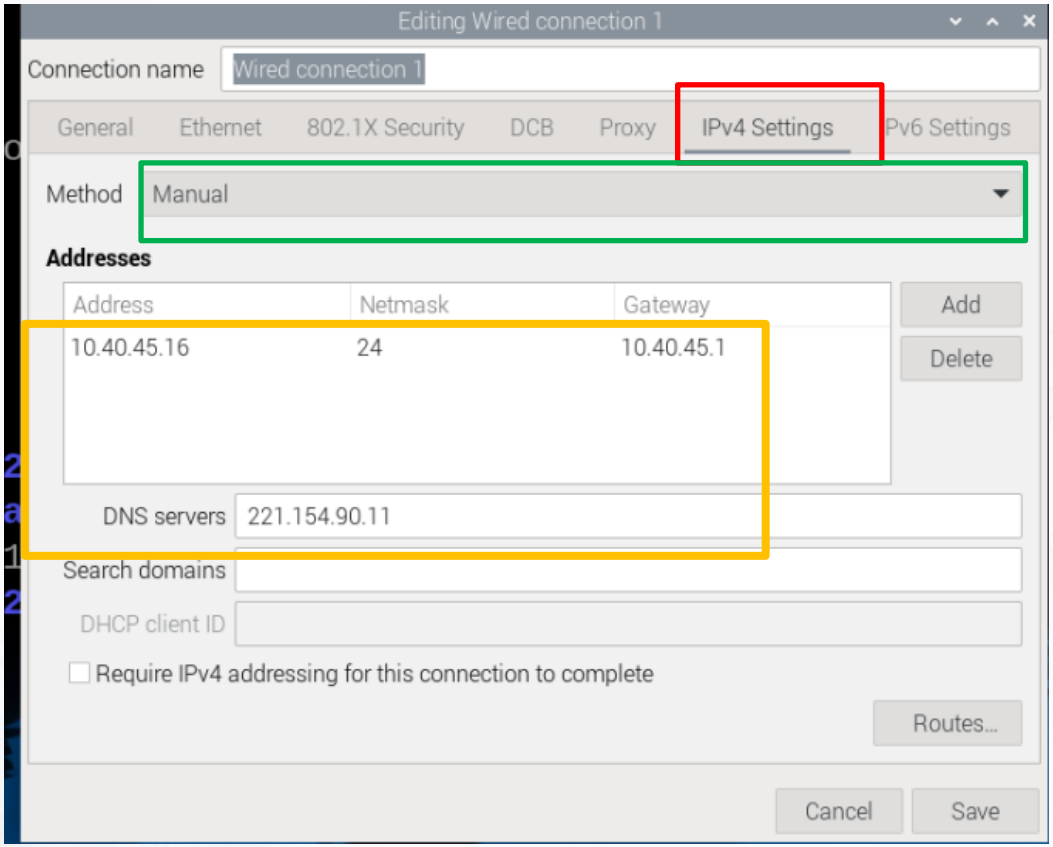
- 라즈베리파이 이더넷 설정
  - ✓ 고정 ＩＰ 설정



- 라즈베리파이 이더넷 설정
  - ✓ 고정 ＩＰ 설정



- 라즈베리파이 이더넷 설정
  - ✓ 고정 ＩＰ 설정



리눅스 사용방법 기초

기본 디렉토리 명령어

ls	현재 경로 파일 리스트
cd	디렉토리 변경
mkdir	디렉토리 생성
pwd	현재 디렉토리

운영 관련 명령어

ps -aux	현재 실행중인 프로세스
netstat -tnl	현재 사용중인 TCP 포트
ufw	방화벽 설정
history	명령어 history

주요 명령어

apt-get	소프트웨어 설치/제거
sudo	슈퍼유저권한 대행
su	User 변경
vim	편집기

리눅스 사용방법 기초

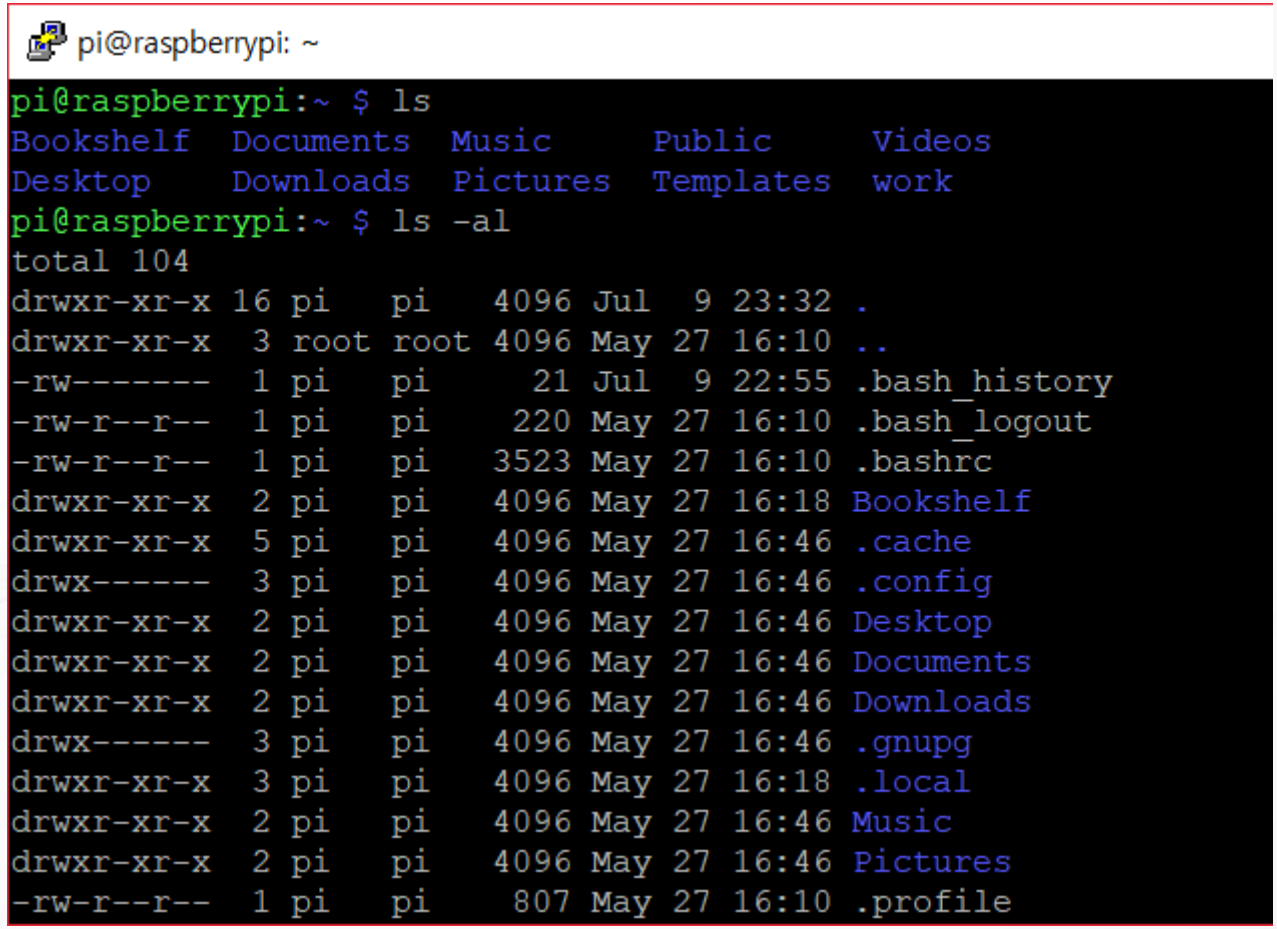
기본 디렉토리 명령어

ls

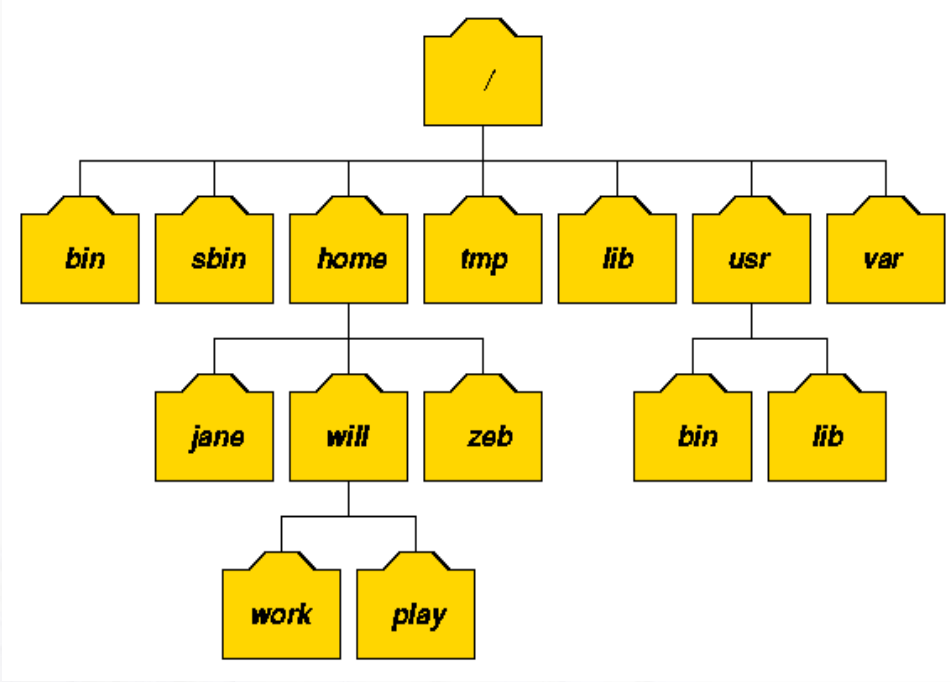
현재 경로 파일 리스트

ls -al

현재 경로 파일 리스트
및
권한, 크기, 날짜



리눅스 디렉토리 구조



/	/
bin 실행 파일 모음	**tmp** 임시파일
home user 폴더	**root** root home 디렉토리
sbin 시스템 관리용 실행 파일	**var** 로그, 시스템 운영중 갱신데이터 저장
usr 각종 프로그램 설치되는 폴더	**lib** 각종 라이브러리 설치되는 폴더

리눅스 사용방법 기초

기본 디렉토리 명령어

cd

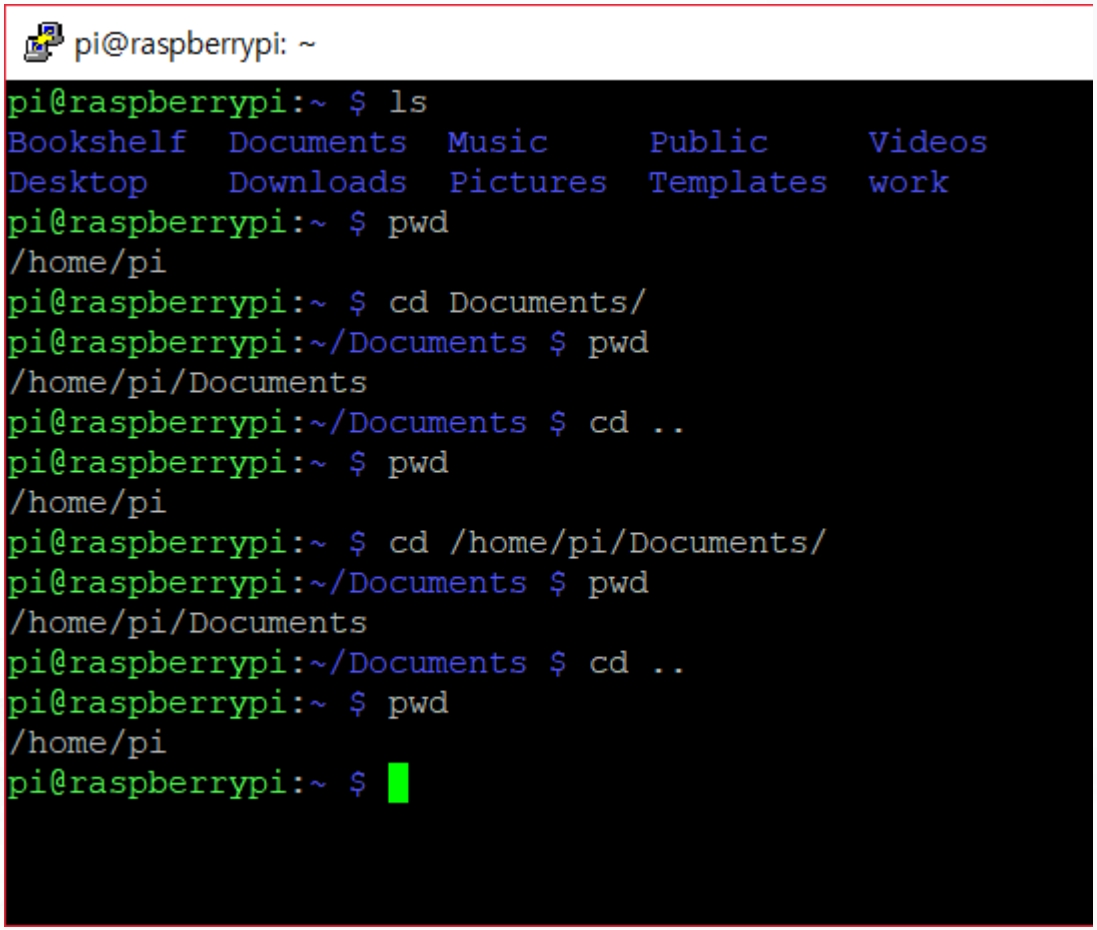
디렉토리 변경

cd ..

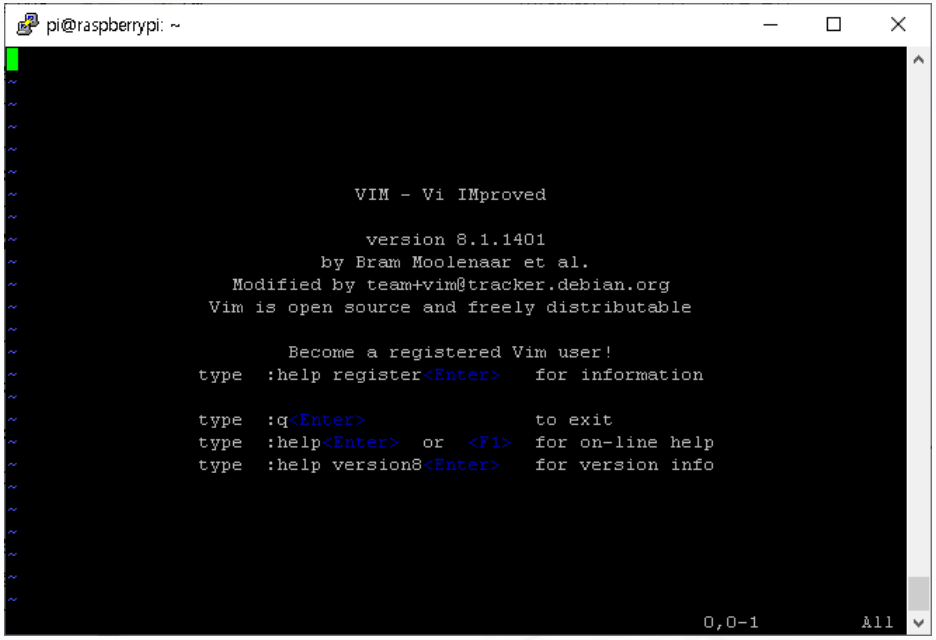
한단계 상위수준
디렉토리로 변경

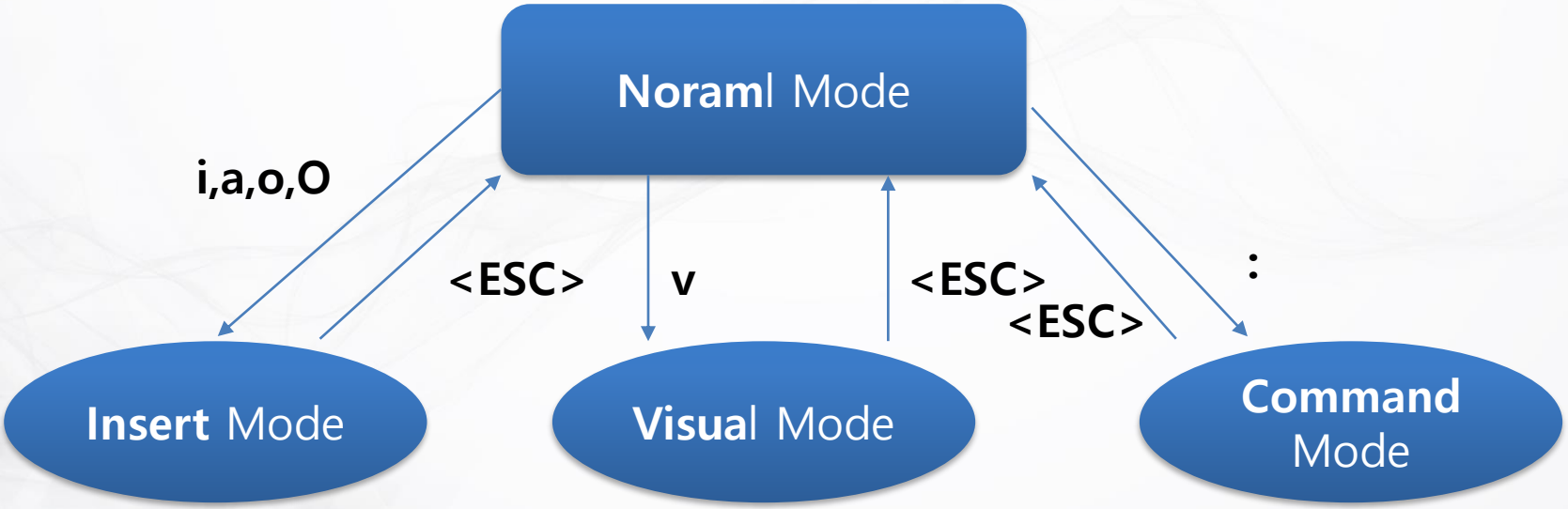
cd /home/pi/Document

절대경로 기준으로
디렉토리로 변경



Vim 편집기


pi@raspberrypi: ~
VIM - Vi IMproved
version 8.1.1401
by Bram Moolenaar et al.
Modified by team+vim@tracker.debian.org
Vim is open source and freely distributable
Become a registered Vim user!
type :help register<Enter> for information
type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version8<Enter> for version info
0,0-1 All


Noraml Mode
i,a,o,O
<ESC>
v
<ESC>
<ESC>
:
Insert Mode
Visual Mode
Command Mode

Vim Editor Setting

파일명: .vimrc

경로: /home/pi/.vimrc

pi@raspberrypi: ~

```
 1 set nu
 2 set cindent
 3 set bg=dark
 4 set autoindent
 5 set expandtab
 6 set softtabstop=4
 7 set tabstop=4
 8 let python_version_2 = 1
 9 let python_highlight_all = 1
10 filetype indent plugin on
11 syntax on
```

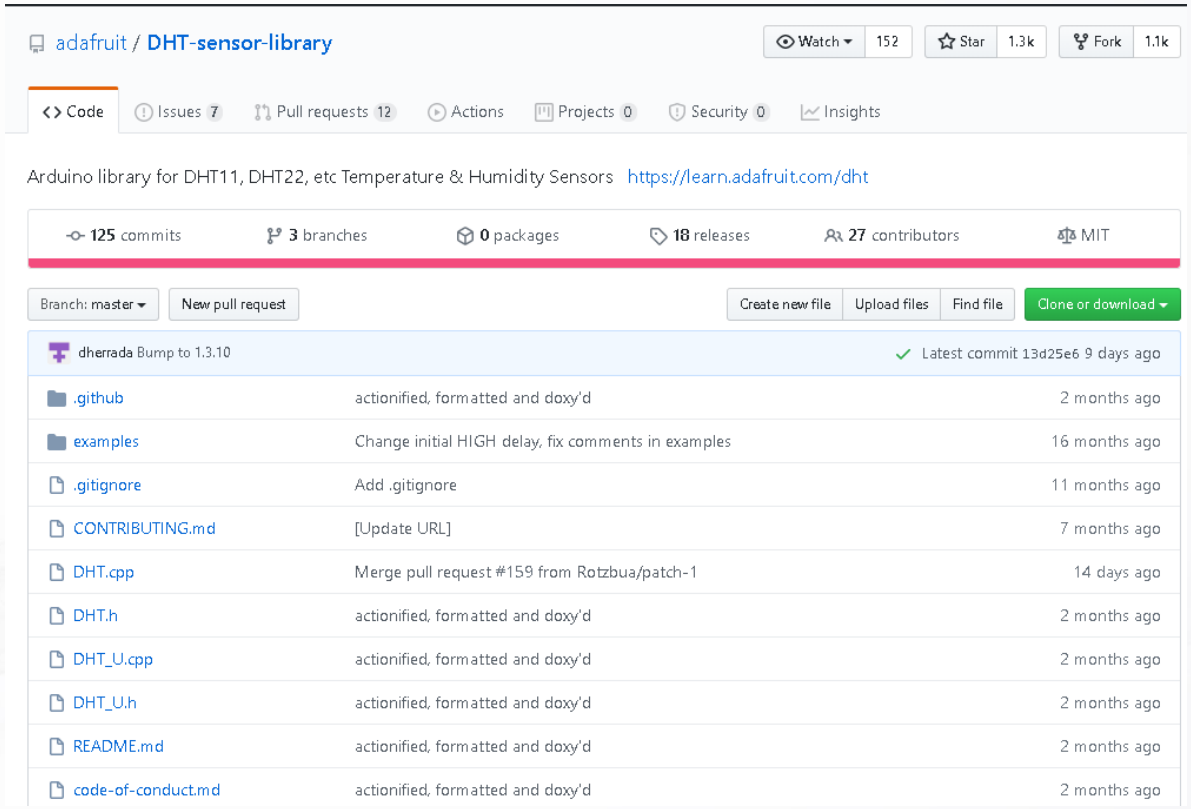
syntax on ----------> 문법 기능 ON
filetype indent plugin on ----------> 확장자로 문서 형식 파악
let python_version_2 = 1 ----------> python 2 문법을 따름(플러그 인)
let python_highlight_all = 1 ----------> 모든 강조(색상) ON(플러그 인)
set tabstop=4 ----------> 탭의 길이 설정
set expandtab
set softtabstop=4 ----------> 탭을 누를 때 실제로 벌어지는 간격
set autoindent ----------> 자동 들여쓰기
set bg=dark ----------> 배경을 어둡게(필수!) 가독성이 좋아짐
set nu ----------> 라인 숫자 ON

GitHub - 분산 버전 관리 툴

Github

https://github.com/

- 깃허브는 버전관리와 협업을 위한 코드 호스팅 플랫폼
- 소프트웨어 개발 프로젝트를 위한 소스 코드 관리 서비스

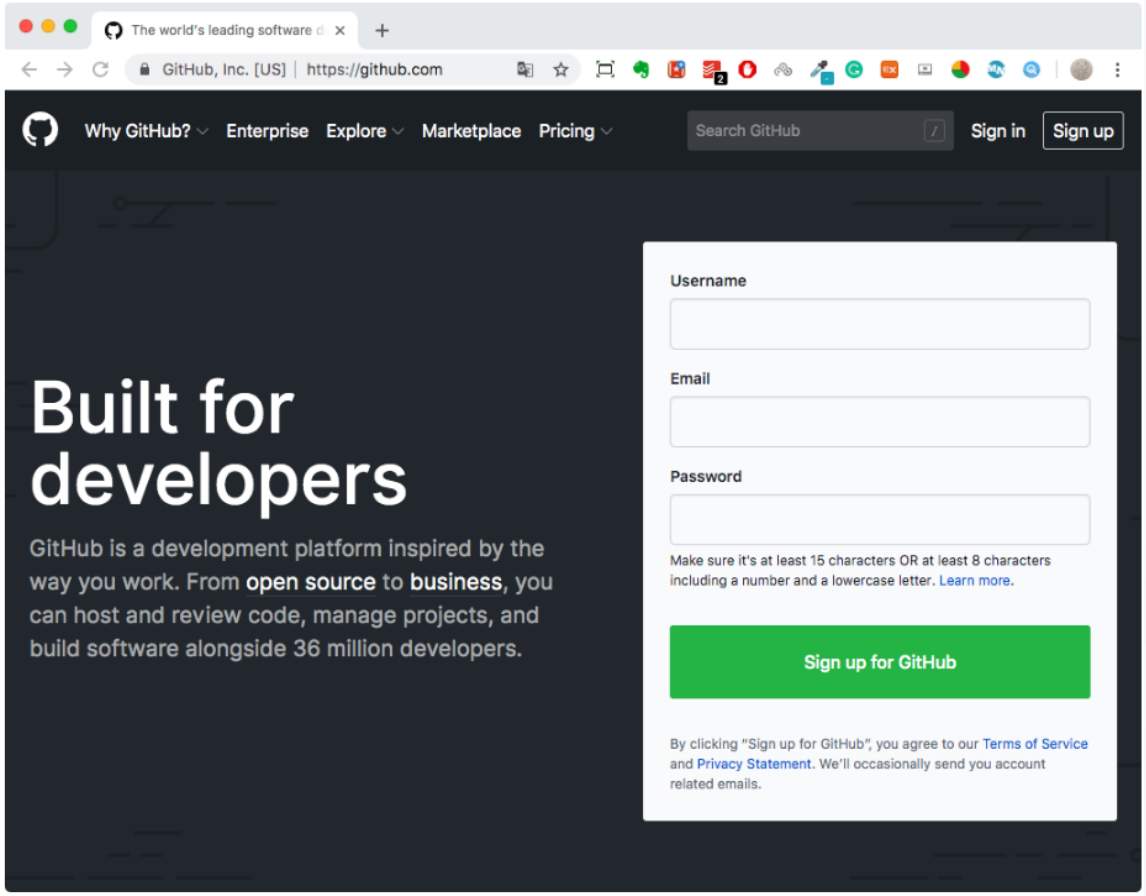


GitHub - 분산 버전 관리 툴

Github

https://github.com/

- 회원 가입(Sign in)



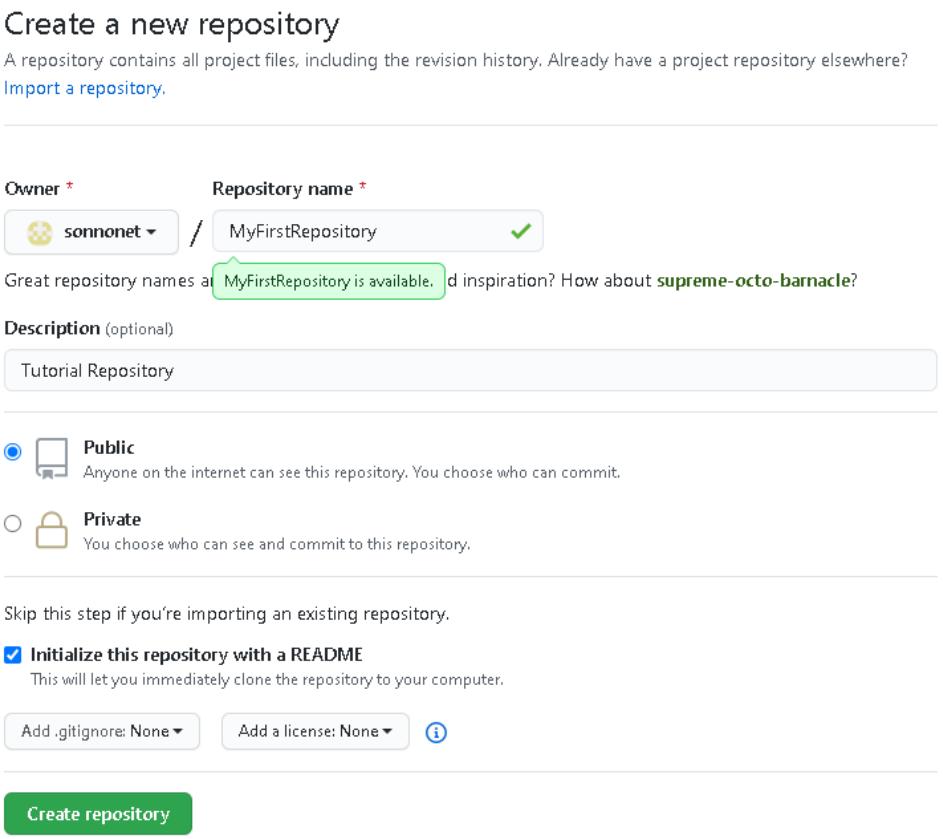
GitHub - 분산 버전 관리 툴

Step 1. 저장소(Repository) 생성하기

- 우측 상단의 '+' 버튼을 클릭 -> New repository

Step 1. 저장소(Repository) 생성하기

- Repository Name : MyFirstRepository
- Desciption : Tutorial Repository
- Check Public
- Check Initalize this repository with a README
- Create Repository 클릭

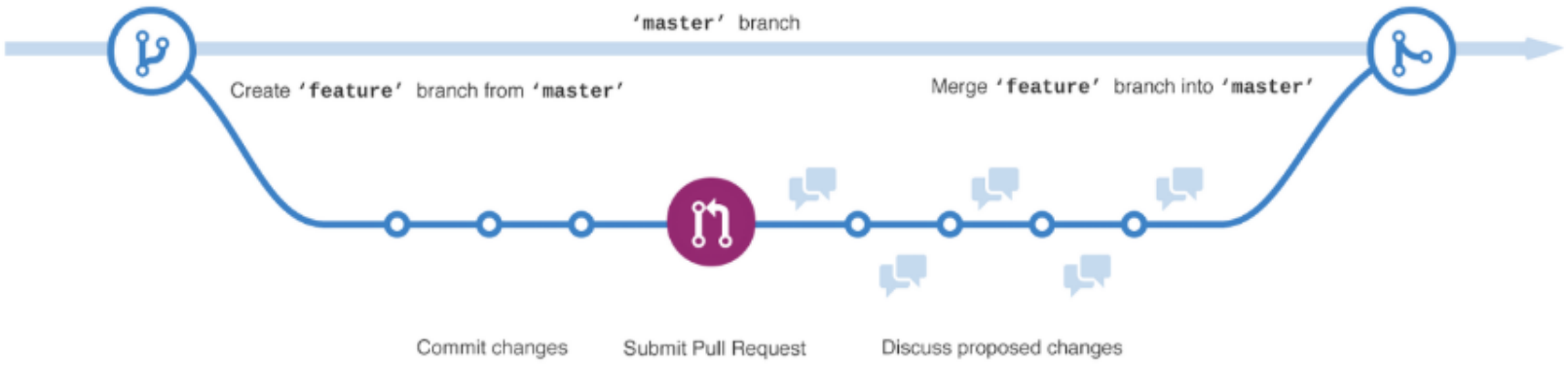


Step 2. 브랜치(Branch) 생성하기

- 브랜칭(Branching)은 하나의 저장소에서 서로 다른 버전을 동시에 작업하는 방법
- 기본적으로 저장소는 "Master"라는 이름의 메인 브랜치를 가짐.
- "Master" 외의 다른 브랜치를 만들어서 "Master"에 커밋하기 전에 수정, 테스트
  - 커밋 (commit) : 파일을 추가하거나 변경 내용을 저장소에 저장하는 작업
- Master 외의 브랜치가 만들어지면 그순간에 Master의 Copy 혹은 스냅샷이 생성
- 새로운 브랜치에서 작업하는 동안 다른 사용자가 Master 브랜치를 수정한다면, 그 업데이트된 저장소를 풀로 받을수 있음.
  - 풀(pull) 다른 사용자의 업데이트 혹은 커밋(commit) 내용을 클라이언트로 내려 받는 명령어.

Step 2. 브랜치(Branch) 생성하기

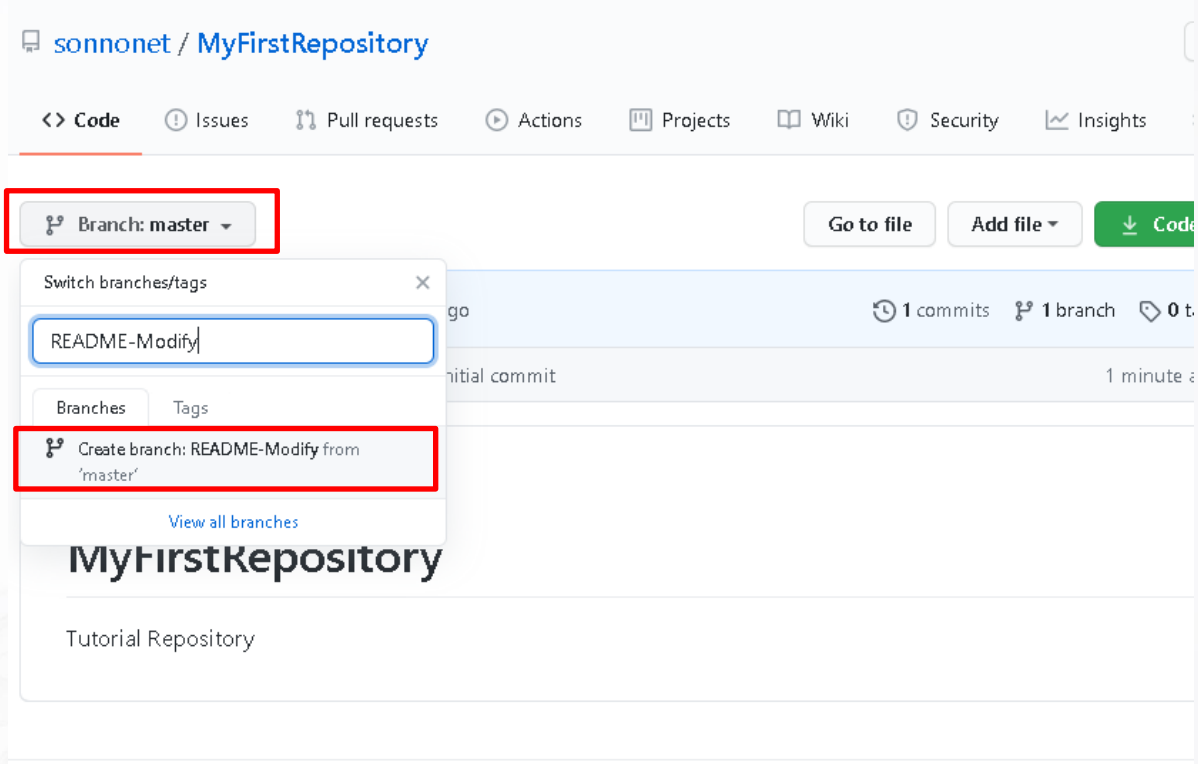
- master 브랜치
- feature라는 새로운 브랜치
- feature 브랜치가 master로 병합(merge)되기 전까지의 과정



- 깃허브에서 개발자,관리자,디자이너들은 브랜치를 사용하여 버그 수정 및 특정작업을 Master 로부터 분리, 수정이 완료되면 작업된 브랜치를 Master로 병합함.

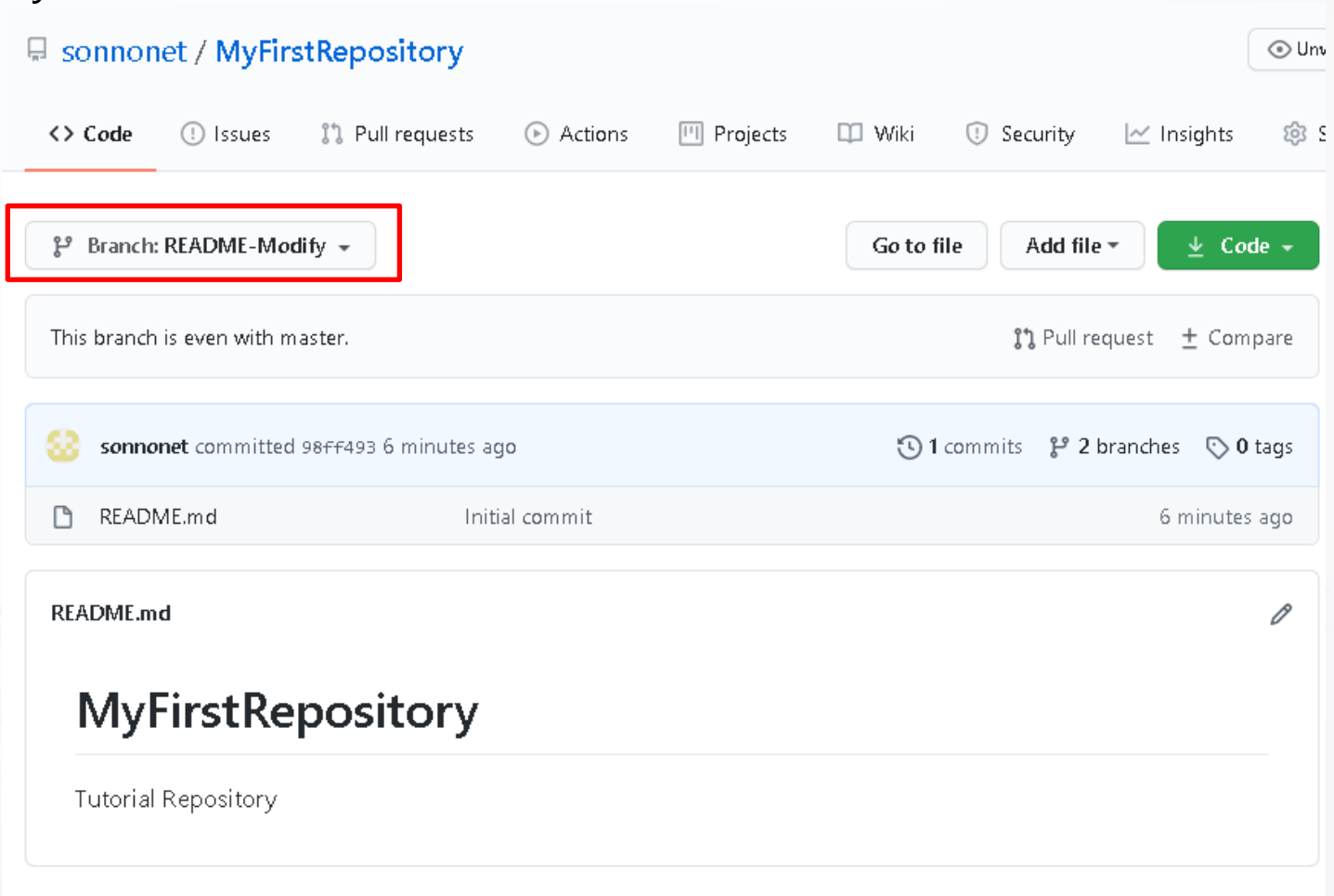
Step 2. 브랜치(Branch) 생성하기

- Branch:master 클릭.
- "README-Modify"라는 master copy 브랜치 생성.
- Create branch:README-Modify from 'master' 클릭.
- 브랜치가 두개 생성
  - master(기본)
  - README-Modify



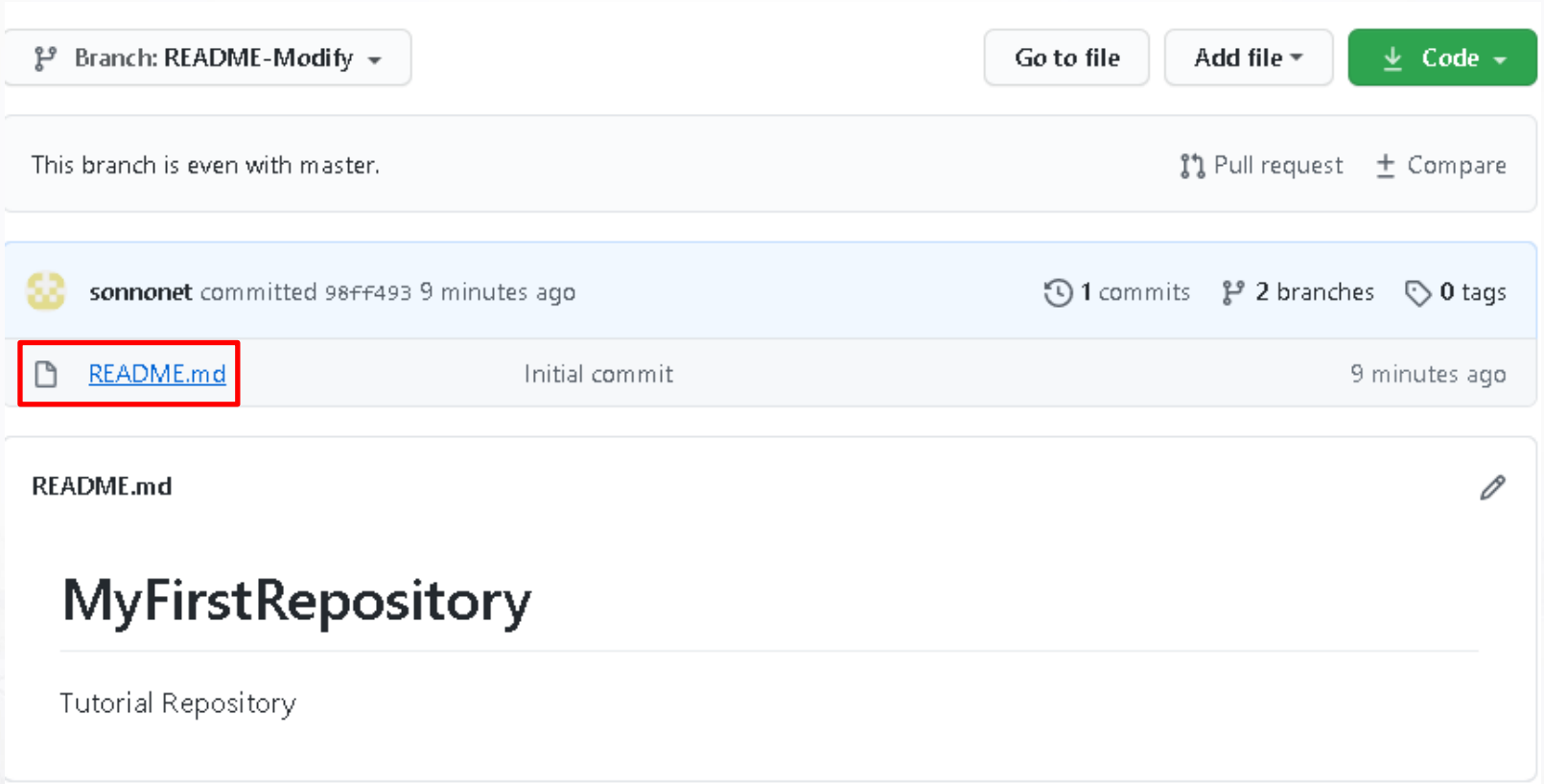
Step 2. 브랜치(Branch) 생성하기

- README-Modify 브랜치 작업



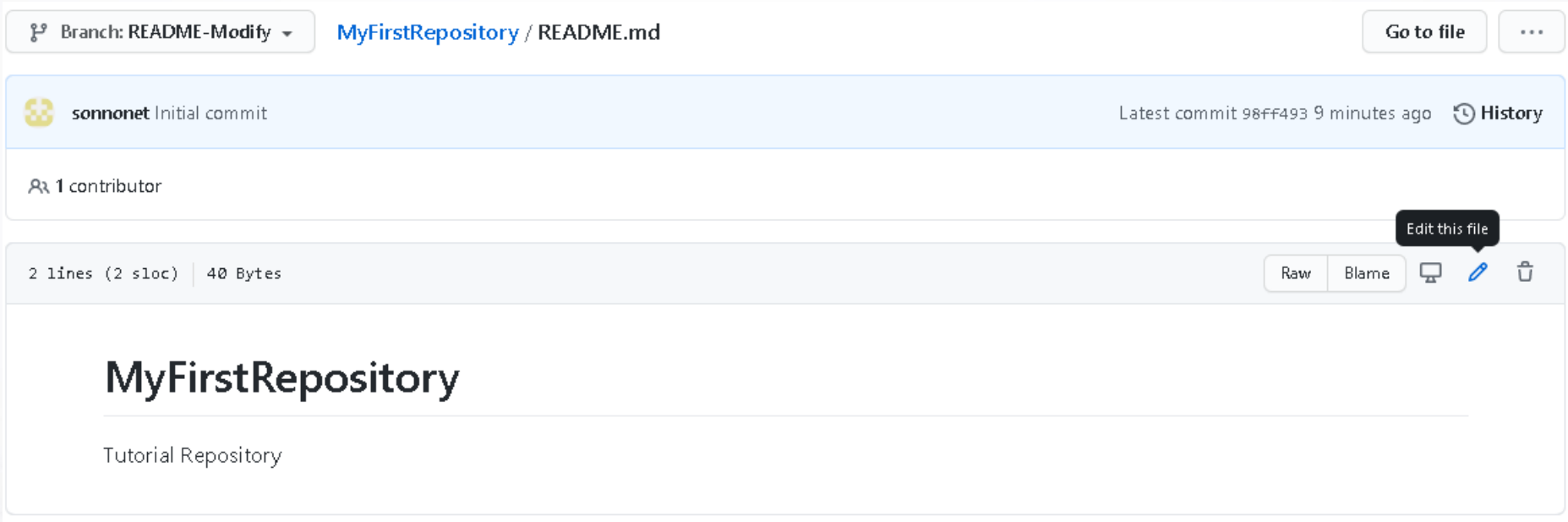
Step 3. 수정하고 커밋(commit)하기

- README.md 클릭



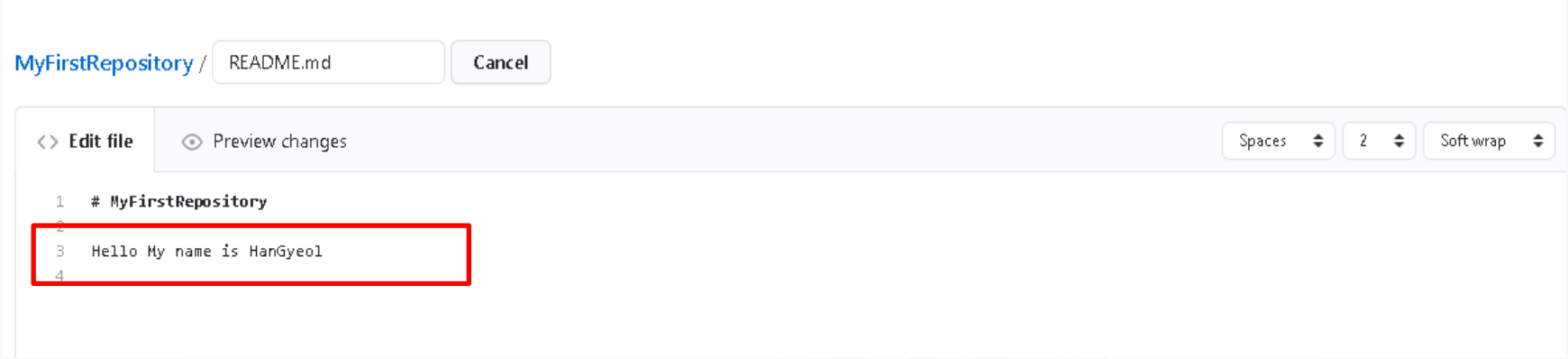
Step 3. 수정하고 커밋(commit)하기

- Edit this file 버튼 클릭

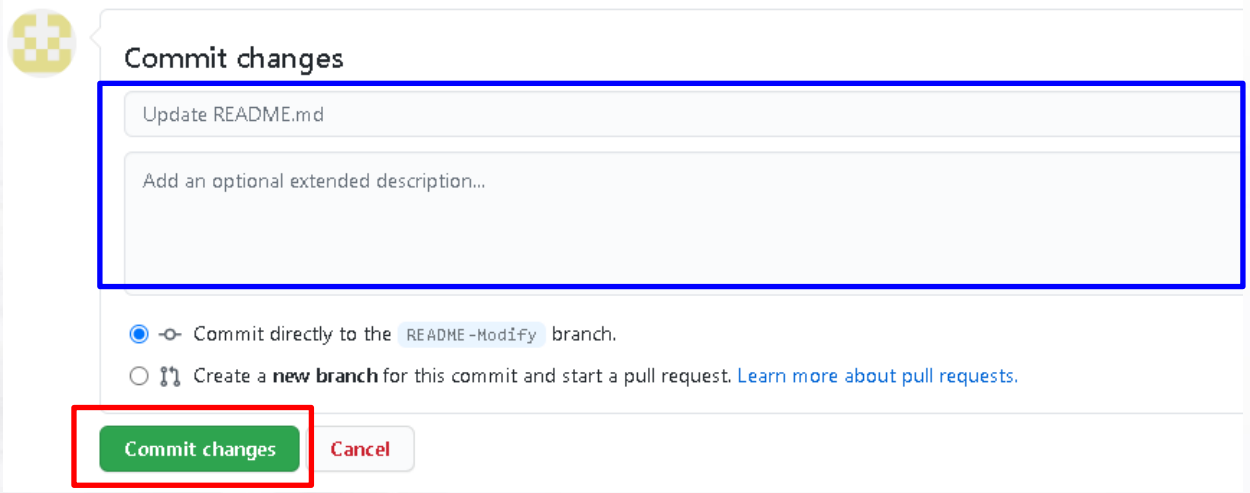


Step 3. 수정하고 커밋(commit)하기

- 원하는 내용으로 수정

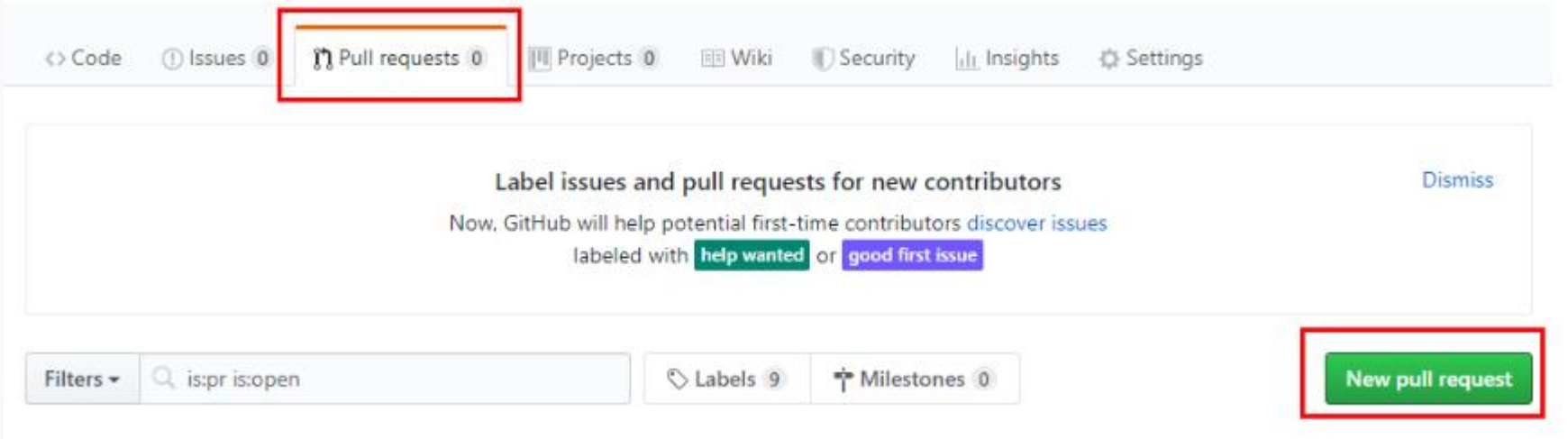


- 커밋 메시지를 작성후, Commit changes 클릭



Step 4. 풀(Pull)요청하기

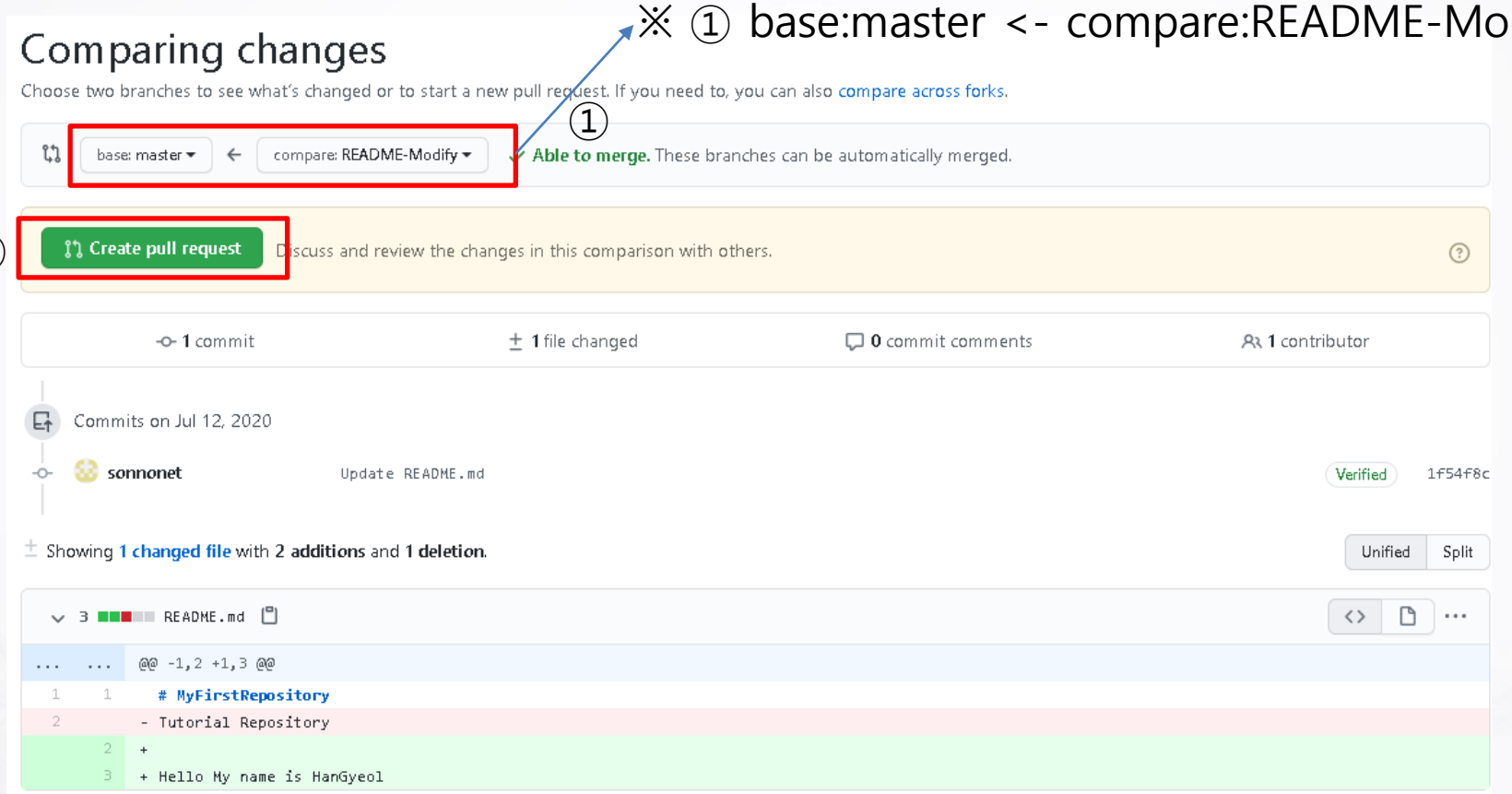
- "Master" 외의 브랜치에서 수정발생, 풀 요청(Pull Request) 가능



- 풀요청(Pull Request)은 깃허브에서 협업의 핵심
- 사용자는 풀 요청을 하여 수정한 내용을 다른 사용자에게 제안하고, 다른사람의 리뷰를 요청하여, 수정된 내용을 그들의 브랜치에 병합
- 커밋을 하면 바로 풀 요청을 열어서 토의를 시작할 수 있음.
- 깃허브의 멘션시스템을 사용하여, 당신의 특정 사람이나 팀에게 피드백을 요청 할 수 있음.

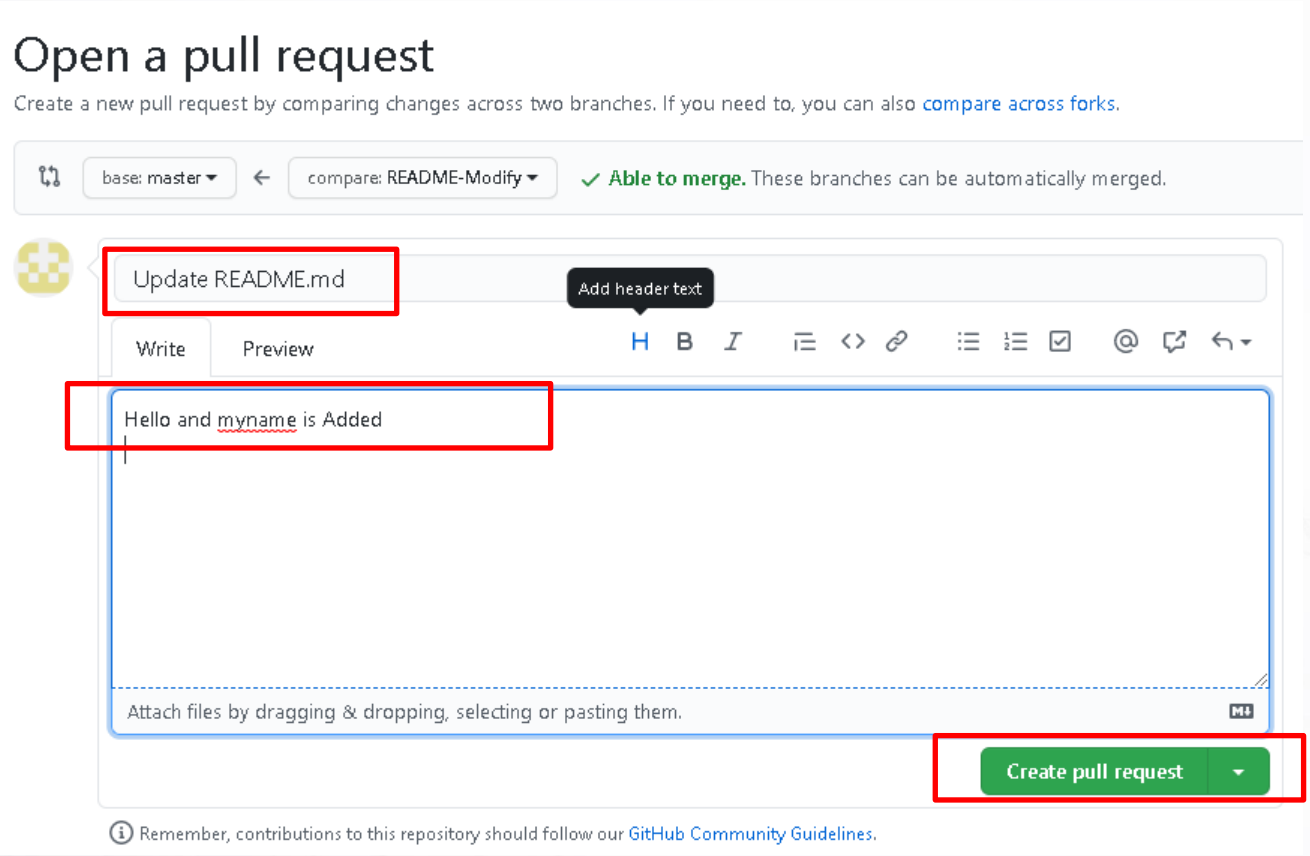
Step 4. 풀(Pull)요청하기

※ ① base:master <- compare:README-Modify 확인



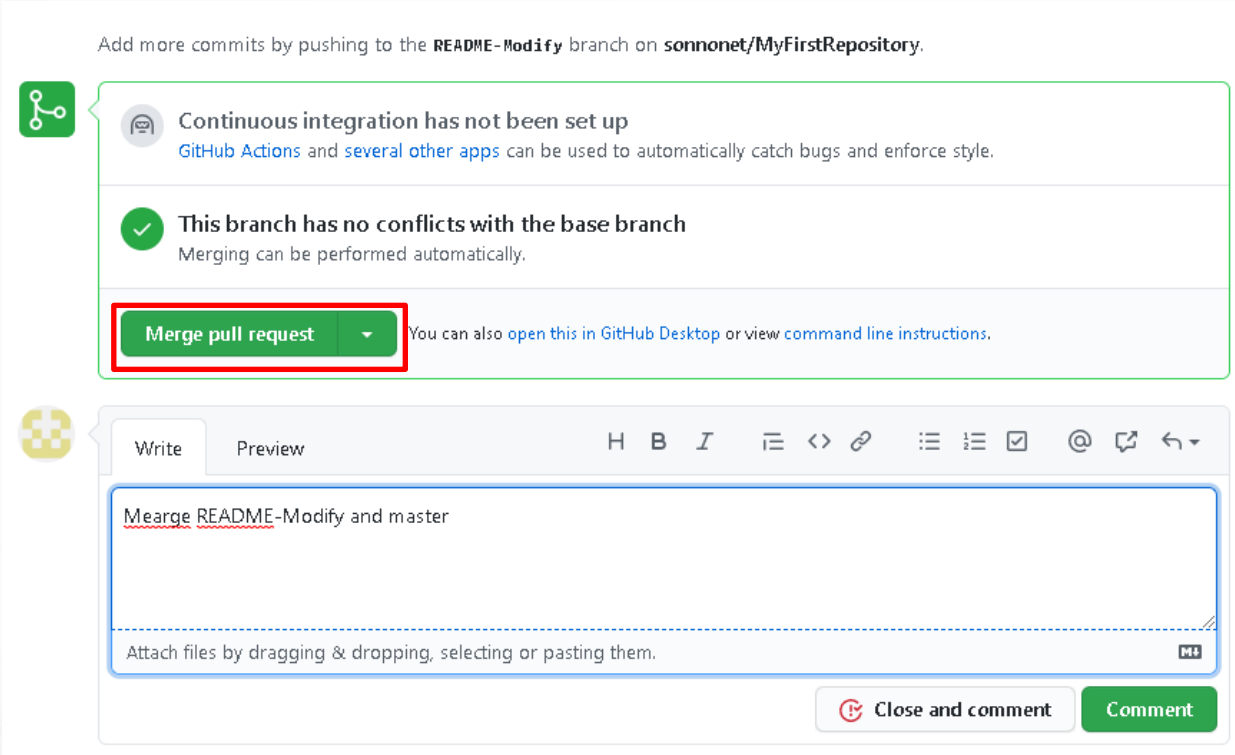
- 풀요청은 두 브랜치로부터 차이점을 보여줌, 수정,추가 (녹색) , 제외(빨간색)
- ② 이상이 없다면 Create pull request 클릭

Step 4. 풀(Pull)요청하기



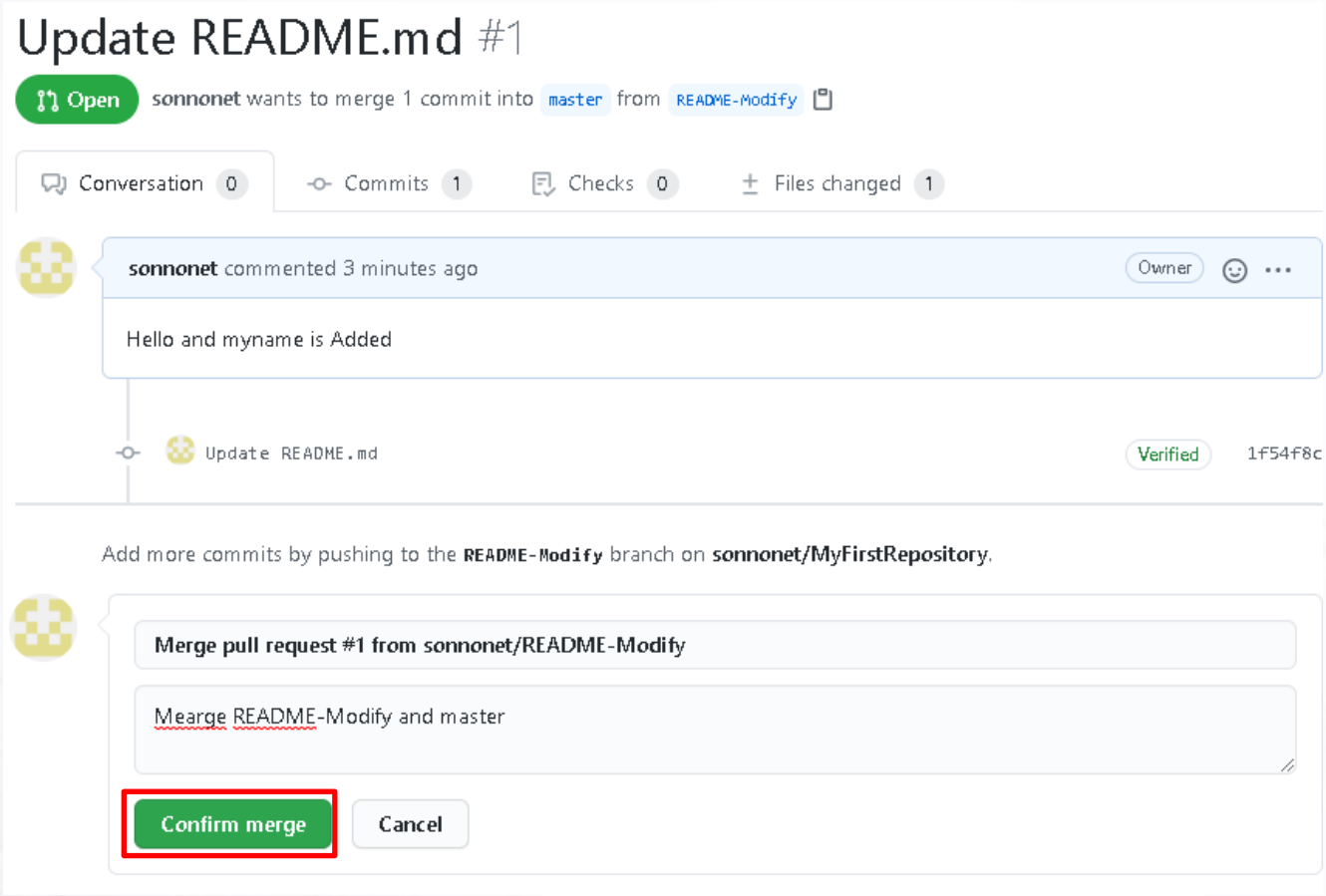
- 간단한 설명을 작성하고, ① Create pull request 클릭

Step 5. 병합(Merge) 하기



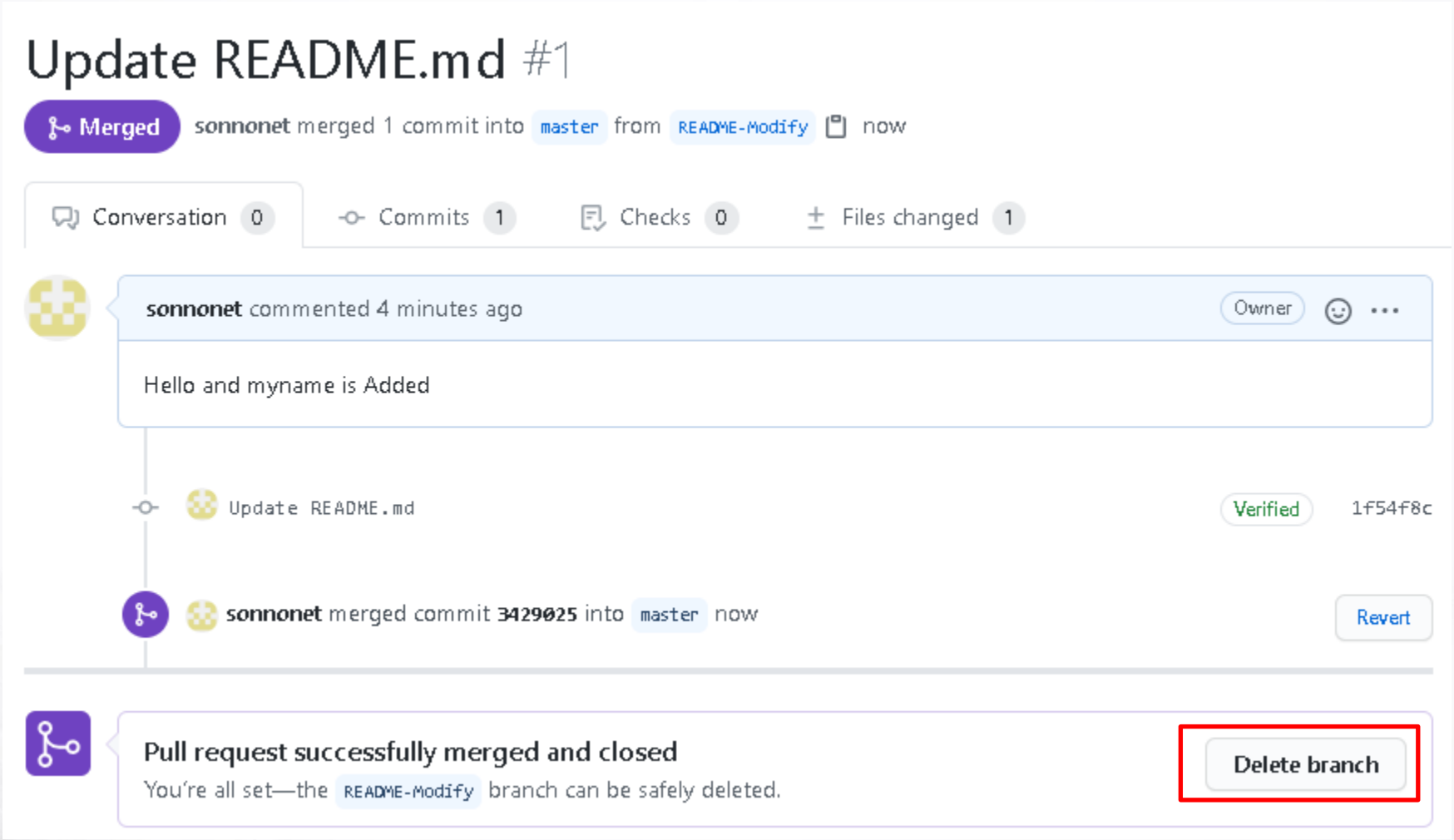
- Merge pull request 클릭

Step 5. 병합(Merge) 하기



- Confirm merge 클릭

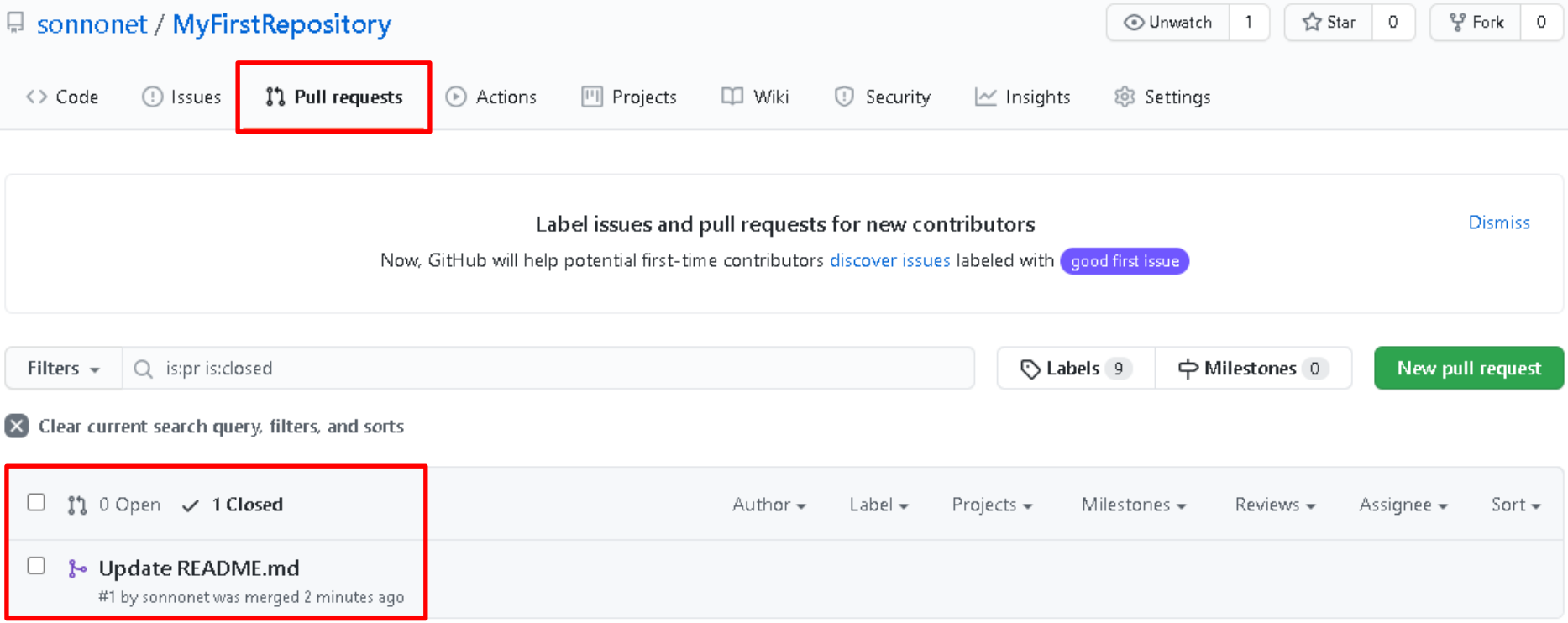
Step 5. 병합(Merge) 하기



- README-modify 는 더 이상 필요 없으므로 Delete branch 클릭

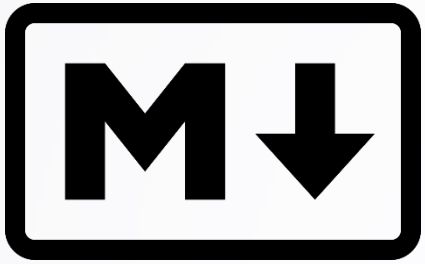
GitHub - 분산 버전 관리 툴

최종 확인



README 파일수정 (md)

마크 다운 문법



https://ko.wikipedia.org/wiki/%EB%A7%88%ED%81%AC%EB%8B%A4%EC%9A%B4

DHT sensor library

Arduino Library CI passing

Description

An Arduino library for the DHT series of low-cost temperature/humidity sensors.

You can find DHT tutorials here.

Dependencies

- Adafruit Unified Sensor Driver

Contributing

Contributions are welcome! Not only you'll encourage the development of the library, but you'll also learn how to best use the library and probably some C++ too

Please read our Code of Conduct before contributing to help this project stay welcoming.

Git 저장소 다운로드

pi@raspberrypi: ~ $	git clone https://github.com/<user_name>/<repository_name>