ASUS IoT

Tinker board

IoT를 위한 최선의 엣지 단말기

MAKER SPACE G·CAMP

Beomseok Park – baam.park.95@gmail.com

Contents

- What is Git and Github
- Git installation
- Git workflow in Local Environment
- Collaboration on Github
- VCS on Android Stdio

*eMMC by SKU

What is Git and Github

Git vs Github

Git

- Version Control을 위한 Software
- Local machine에서 Git을 이용하여 repository 작업

Github

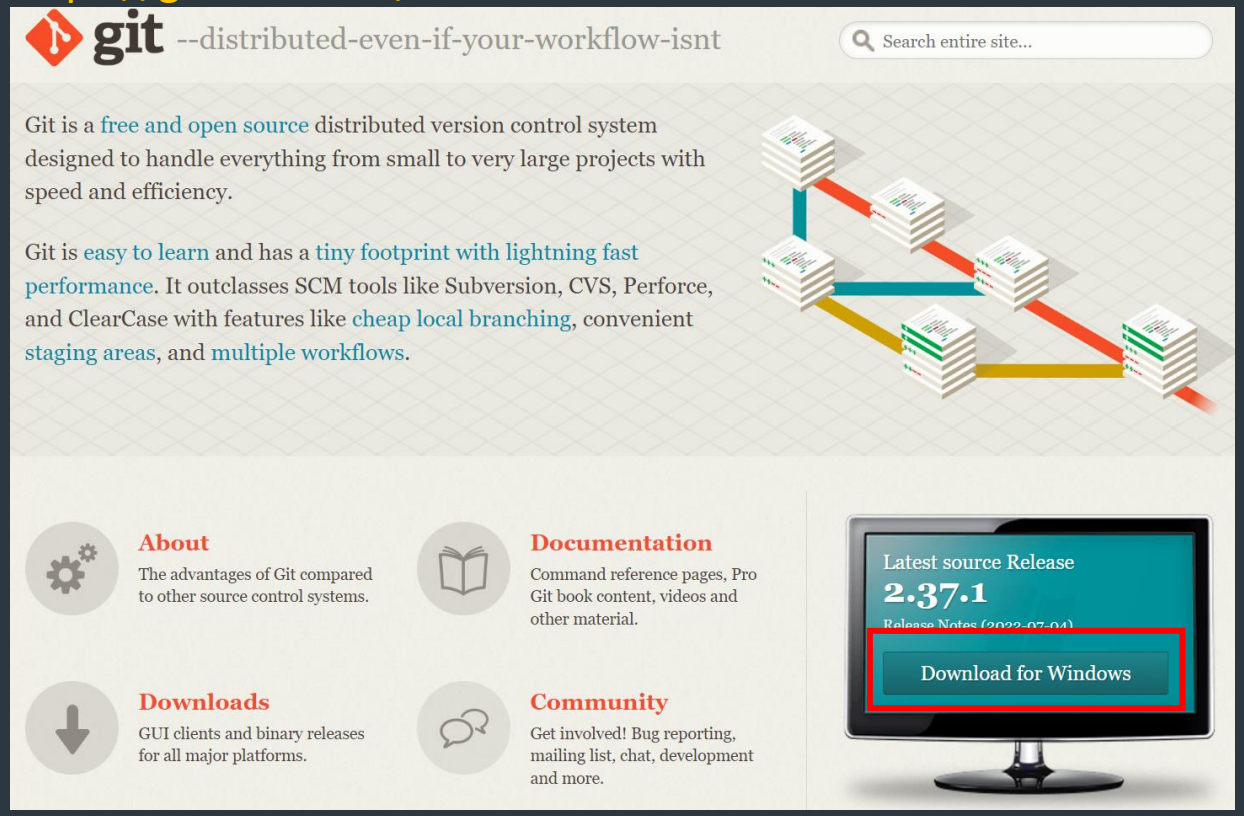
- 협업을 위한 Cloud Service
- local에서 작업한 repository를 Github에 업데이트
- Github에서 생성된 repository를 local에서 작업

Git Installation

How to install Git

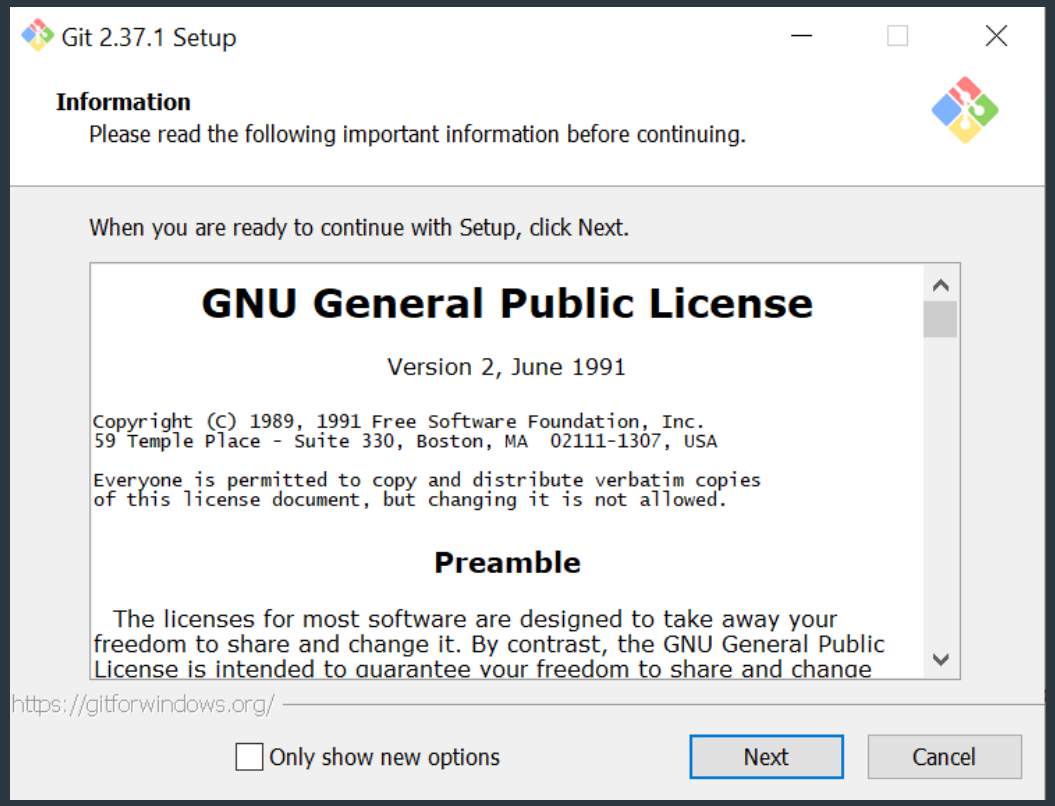
링크를 통해 다운로드

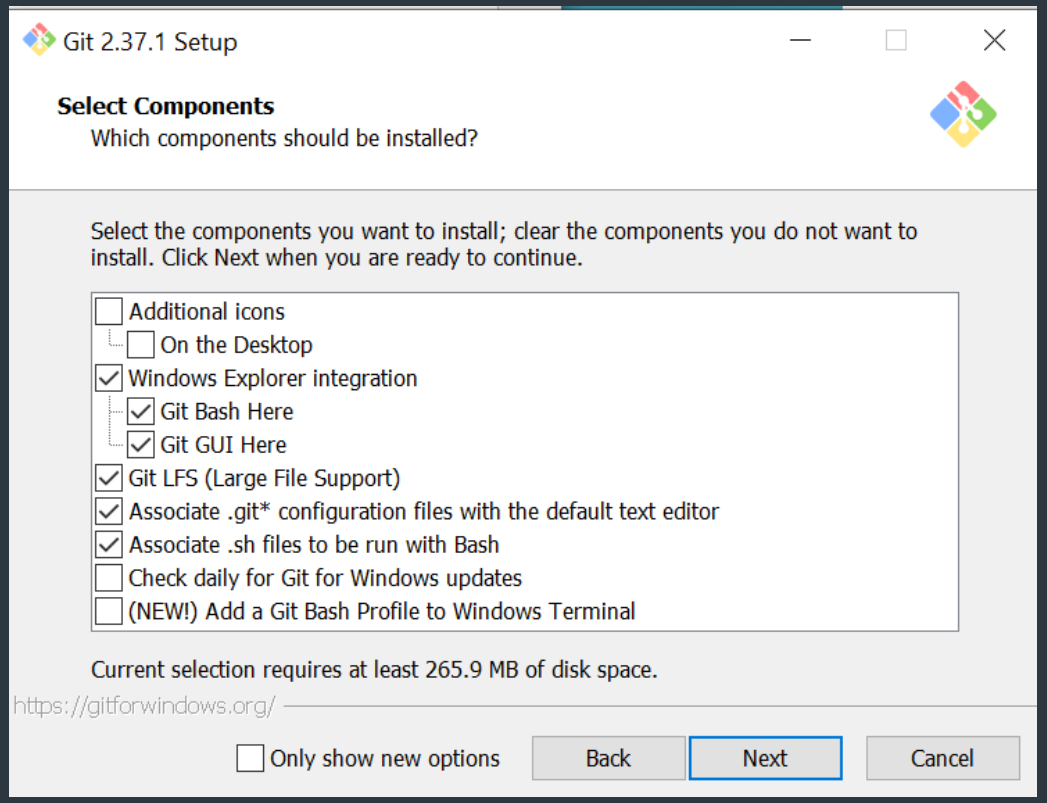
https://git-scm.com/



사양에 맞는 다운로드 링크를 클릭

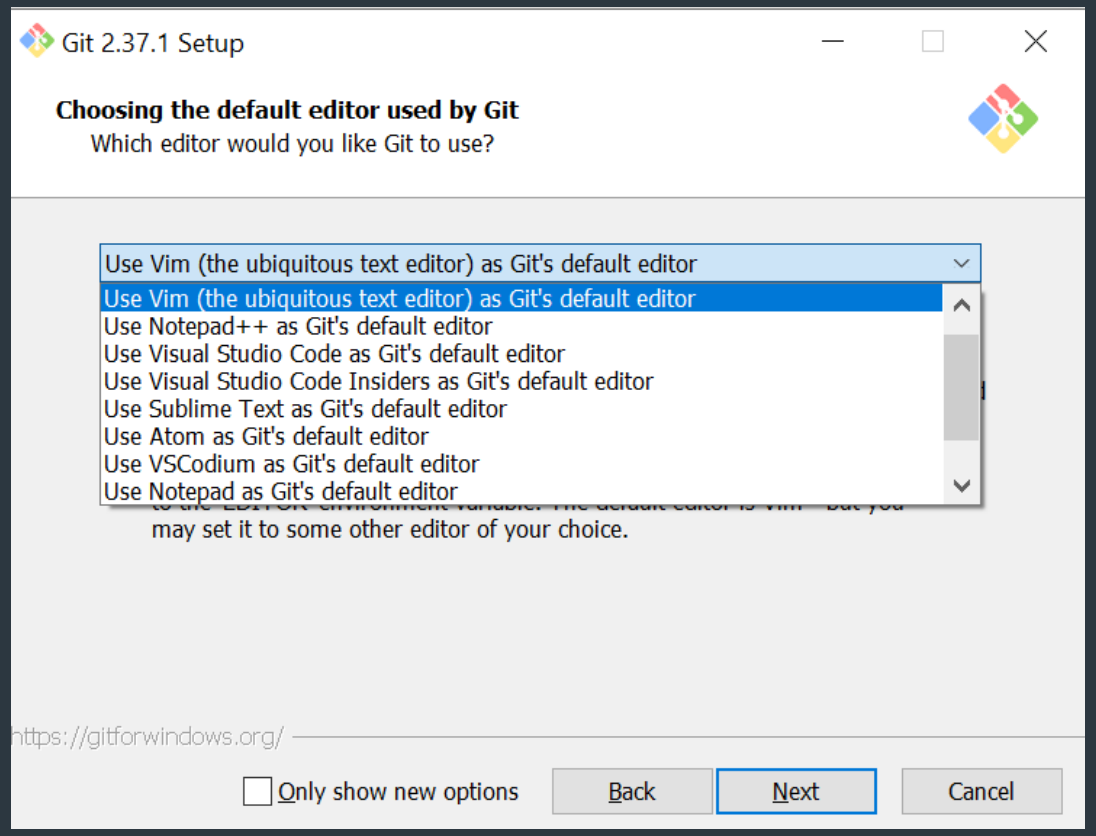
How to install Git

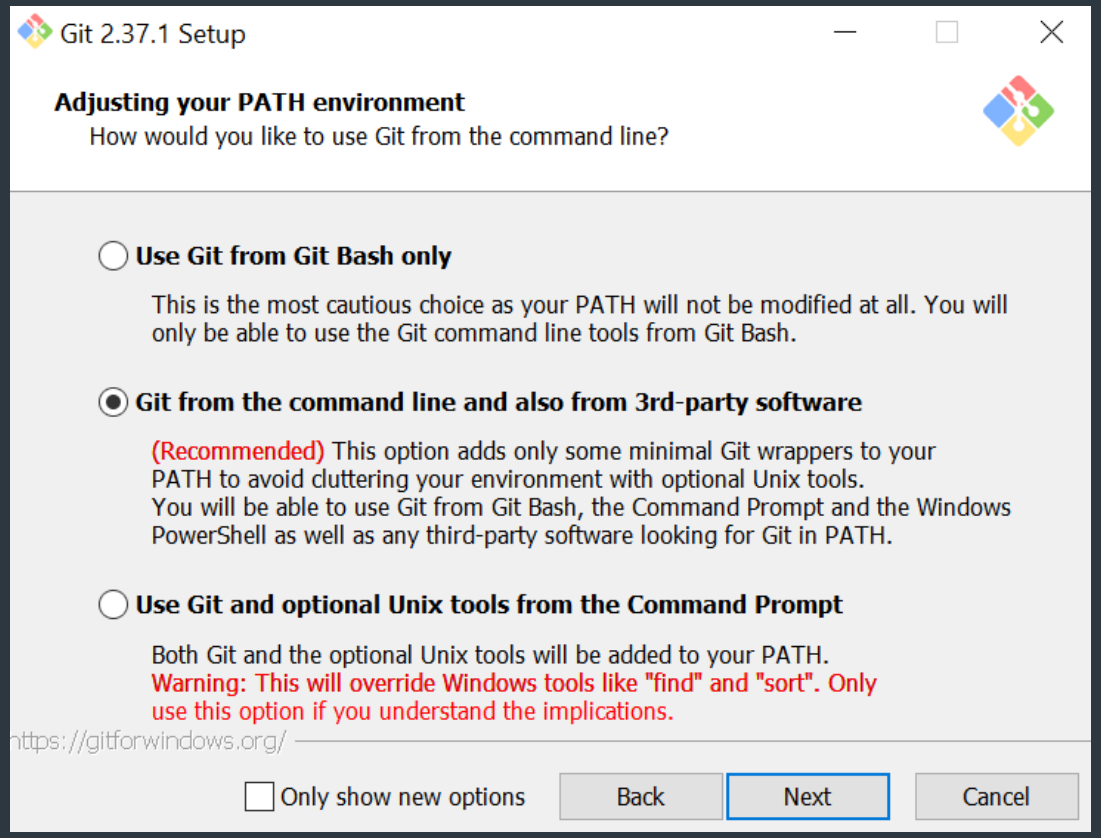




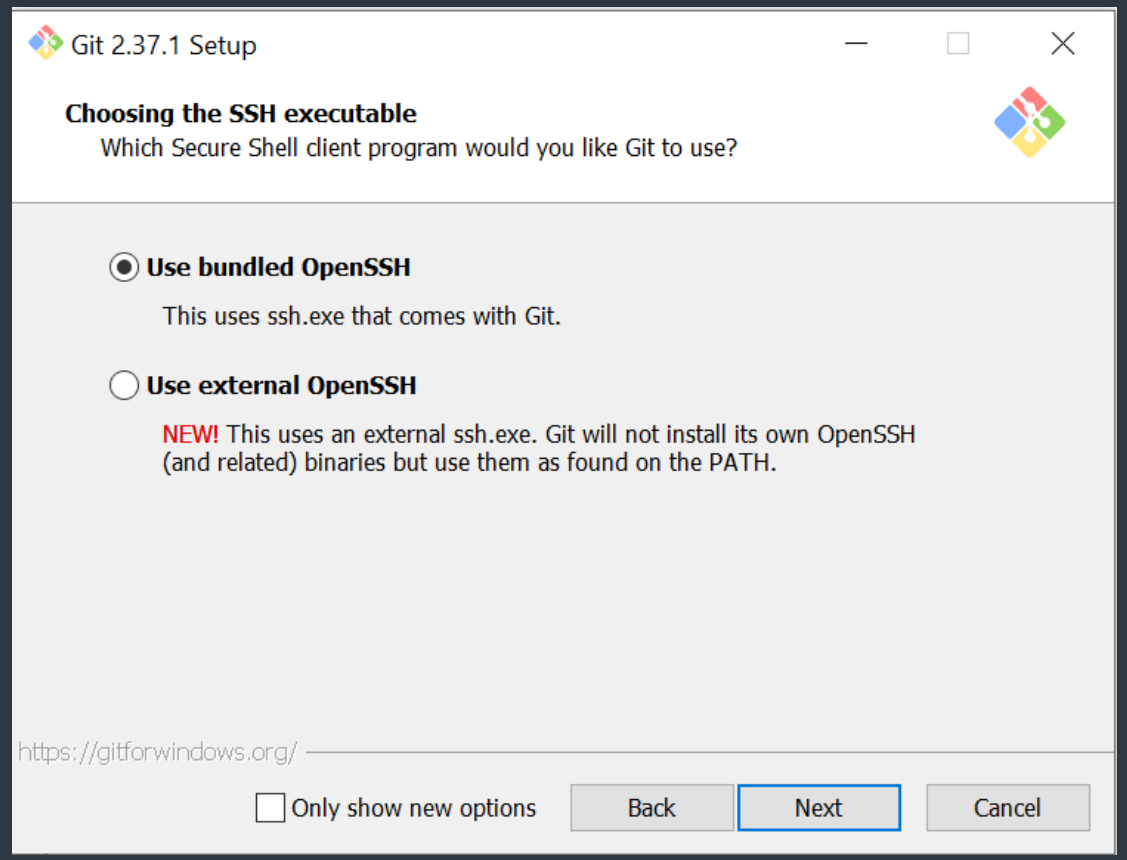
How to install Git

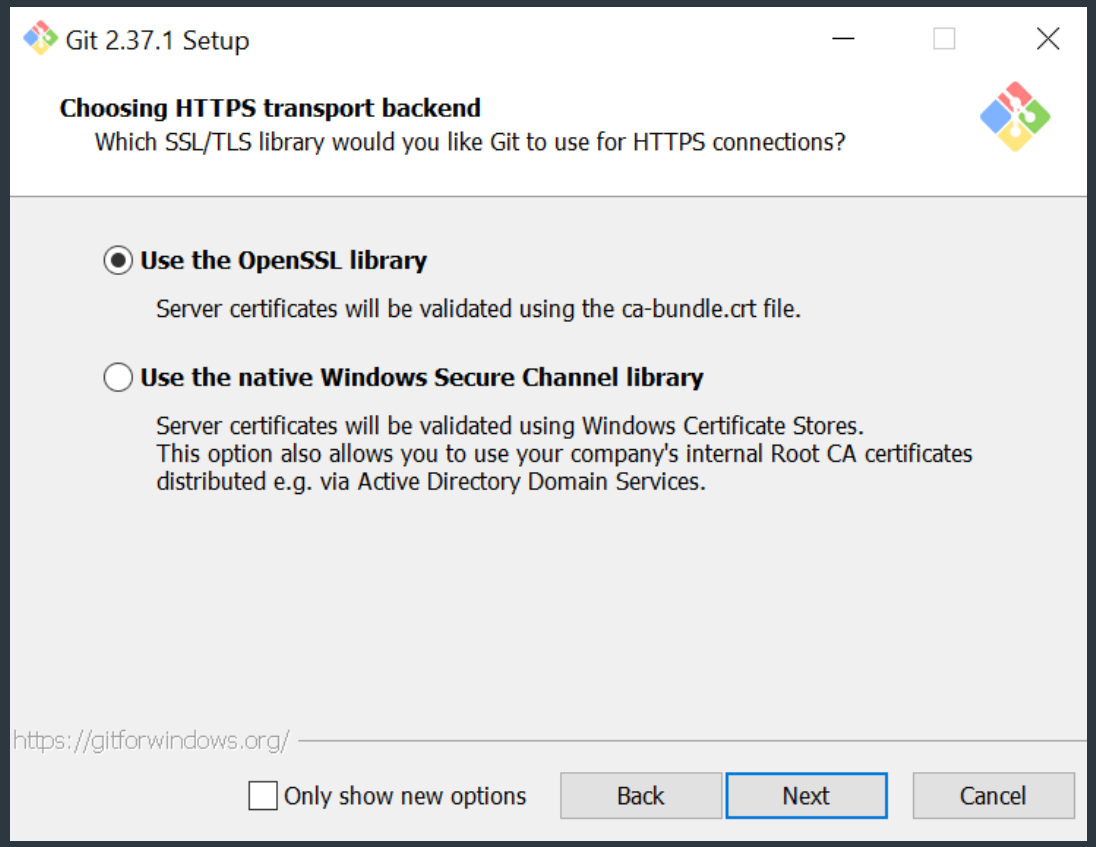
Default Editor를 선택



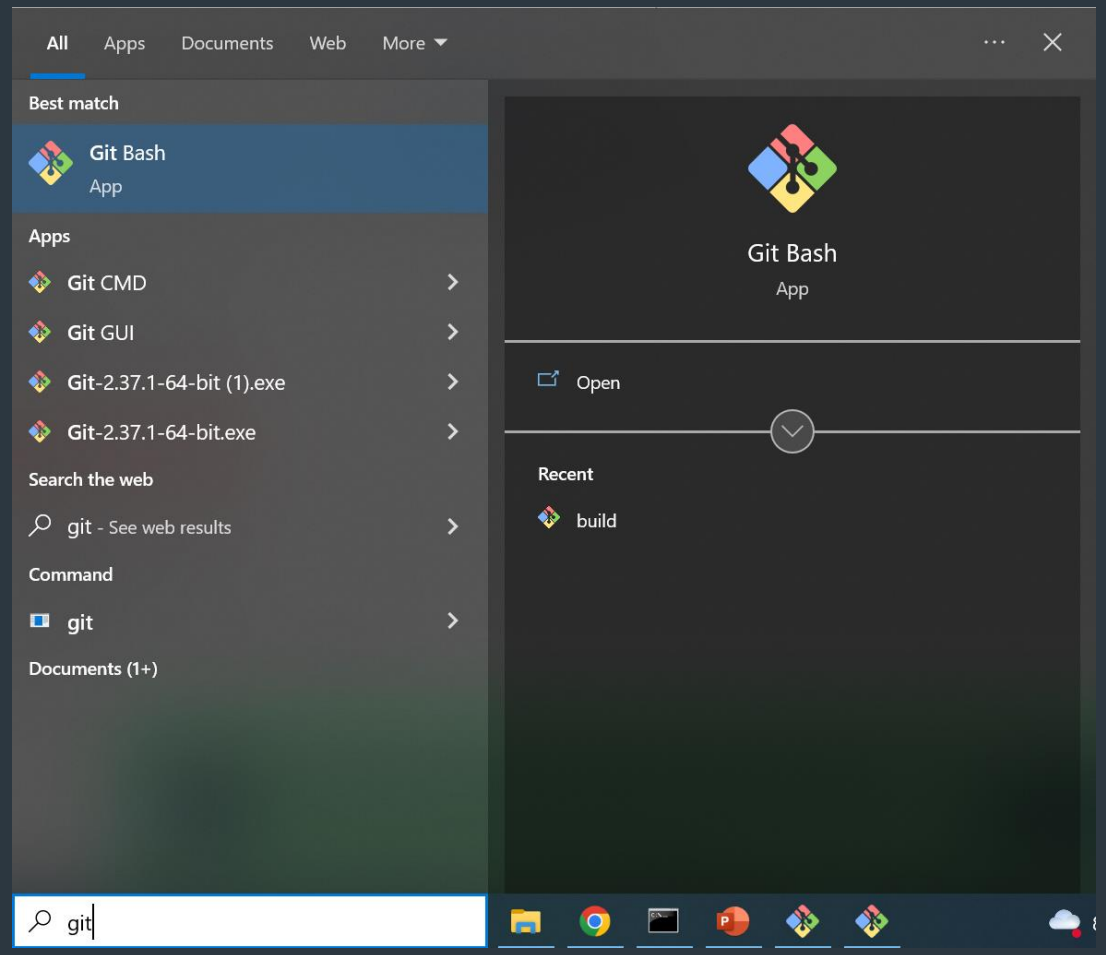


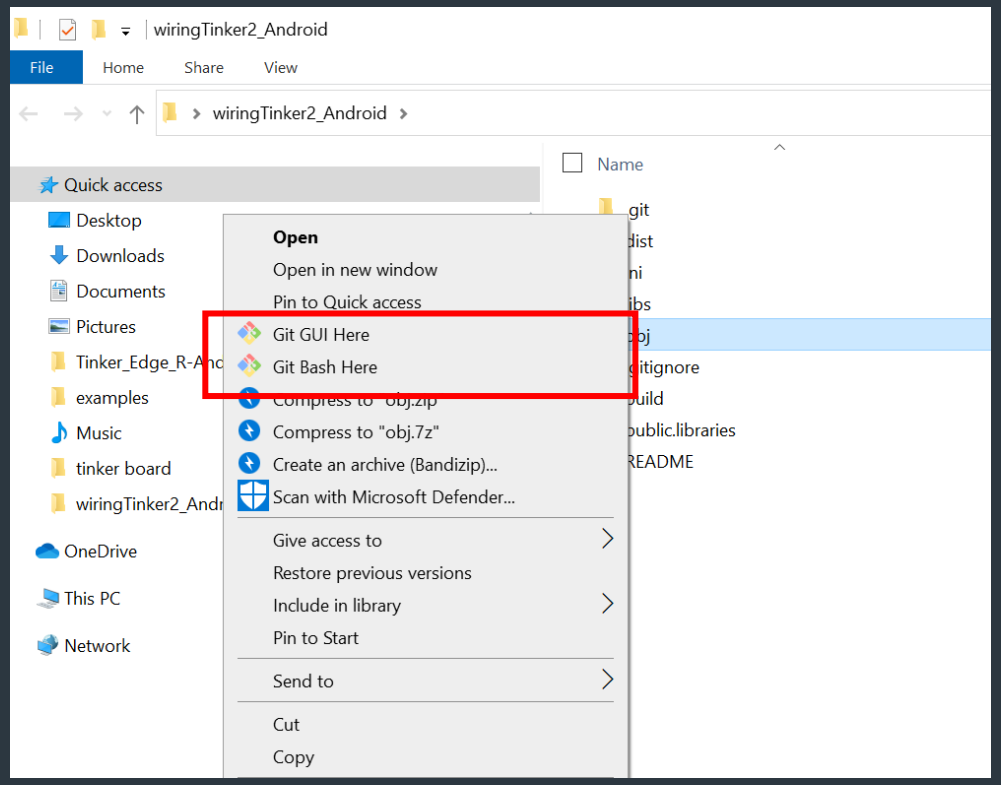
How to install Git





설치 확인


All
Apps
Documents
Web
More
Best match
Git Bash
App
Apps
Git CMD
Git GUI
Git-2.37.1-64-bit (1).exe
Git-2.37.1-64-bit.exe
Search the web
git - See web results
Command
git
Documents (1+)
Git Bash
App
Open
Recent
build
git


wiringTinker2_Android
File
Home
Share
View
wiringTinker2_Android
Quick access
Desktop
Downloads
Documents
Pictures
Tinker_Edge_R-And
examples
Music
tinker board
wiringTinker2_Andr
OneDrive
This PC
Network
Name
Open
Open in new window
Pin to Quick access
Git GUI Here
Git Bash Here
Compress to "obj.zip"
Compress to "obj.7z"
Create an archive (Bandizip)...
Scan with Microsoft Defender...
Give access to
Restore previous versions
Include in library
Pin to Start
Send to
Cut
Copy

Git CMD vs Git Bash

Git CMD

Git CMD

```
   commit     Record changes to the repository
   merge      Join two or more development histories together
   rebase     Reapply commits on top of another base tip
   reset      Reset current HEAD to the specified state
   switch     Switch branches
   tag        Create, list, delete or verify a tag object signed with
 GPG

collaborate (see also: git help workflows)
   fetch      Download objects and refs from another repository
   pull       Fetch from and integrate with another repository or a l
ocal branch
   push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

C:\Users\Baam>
```

- 일반적인 Windows cmd와 동일
- Git 명령어 사용 가능

Git Bash

MINGW64:/c/Users/Baam/projects

```
'Saved Games'/
 Searches/
 SendTo@
'Start Menu'@
 Templates@
 Videos/
 ntuser.dat.LOG1
 ntuser.dat.LOG2
 ntuser.ini

Baam@DESKTOP-4A292VV MINGW64 ~
$ mkdir projects

Baam@DESKTOP-4A292VV MINGW64 ~
$ cd projects

Baam@DESKTOP-4A292VV MINGW64 ~/projects
$ ls

Baam@DESKTOP-4A292VV MINGW64 ~/projects
$
```

- Unix 커맨드를 사용할 수 있는 Shell 프로그램
- Git 명령어 사용 가능

Git workflow and commands

User Name과 User Email 설정

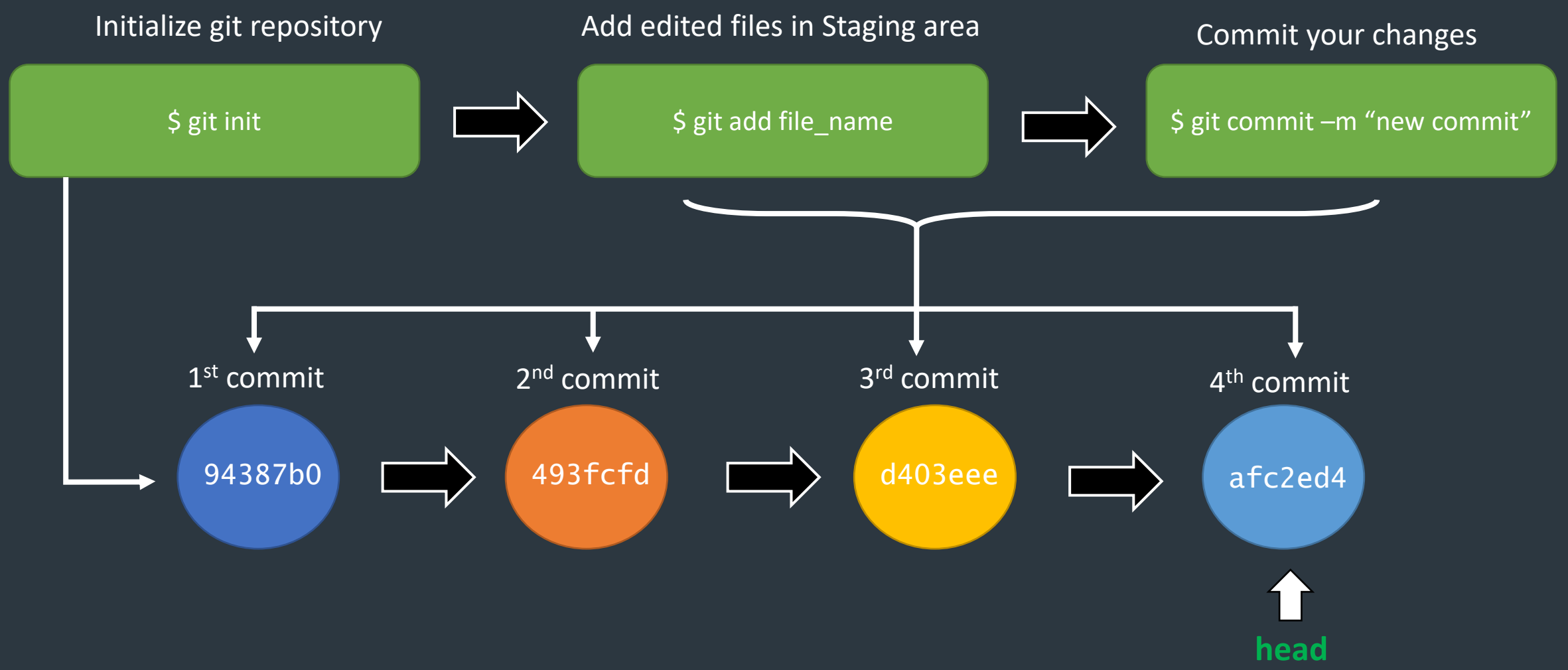
Git Bash에서 아래와 같이 명령어를 입력

```
$ git config --global user.name “Beom”
$ git config --global user.email “myemail@gmail.com”
```

MINGW64:/c/Users/Baam/projects

```
Baam@DESKTOP-4A292VV MINGW64 ~
$ mkdir projects

Baam@DESKTOP-4A292VV MINGW64 ~
$ cd projects

Baam@DESKTOP-4A292VV MINGW64 ~/projects
$ ls

Baam@DESKTOP-4A292VV MINGW64 ~/projects
$ git config -global user.name "Beomseok"
error: did you mean `--global` (with two dashes)?

Baam@DESKTOP-4A292VV MINGW64 ~/projects
$ git config --global user.name "Baam"

Baam@DESKTOP-4A292VV MINGW64 ~/projects
$ git config --global user.email "baam.park.95@gmail.com"

Baam@DESKTOP-4A292VV MINGW64 ~/projects
$
```

Workflow


Initialize git repository
$ git init
Add edited files in Staging area
$ git add file_name
Commit your changes
$ git commit –m “new commit”
1st commit
94387b0
2nd commit
493fcfd
3rd commit
d403eee
4th commit
afc2ed4
head

Exercise 1

1. 폴더를 생성하고 git repository를 initialize
2. 텍스트 파일 한 개를 생성
3. git status 확인
4. 생성한 파일을 Staging Area에 추가
5. git commit
6. 텍스트 파일안에 내용을 추가하여 다시 commit
7. commit log를 확인
8. 첫번째 commit으로 추적

MINGW64:/c/Users/Baam/Desktop/Git Exercise/ex1

```
Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1
$ git init
Initialized empty Git repository in C:/Users/Baam/Desktop/Git Exercise/ex1/.git/

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ ls

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ touch text.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ vi text.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git add text.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git commit -m "add text file"
[master (root-commit) aa6da3a] add text file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 text.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ ls
text.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ ls
text.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git log --oneline
aa6da3a (HEAD -> master) add text file

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git log
commit aa6da3af15deeb2fe7edebfcc52273ebcd80d92e (HEAD -> master)
Author: Baam <baam.park.95@gmail.com>
Date:   Fri Jul 22 07:30:37 2022 +0900

    add text file
```

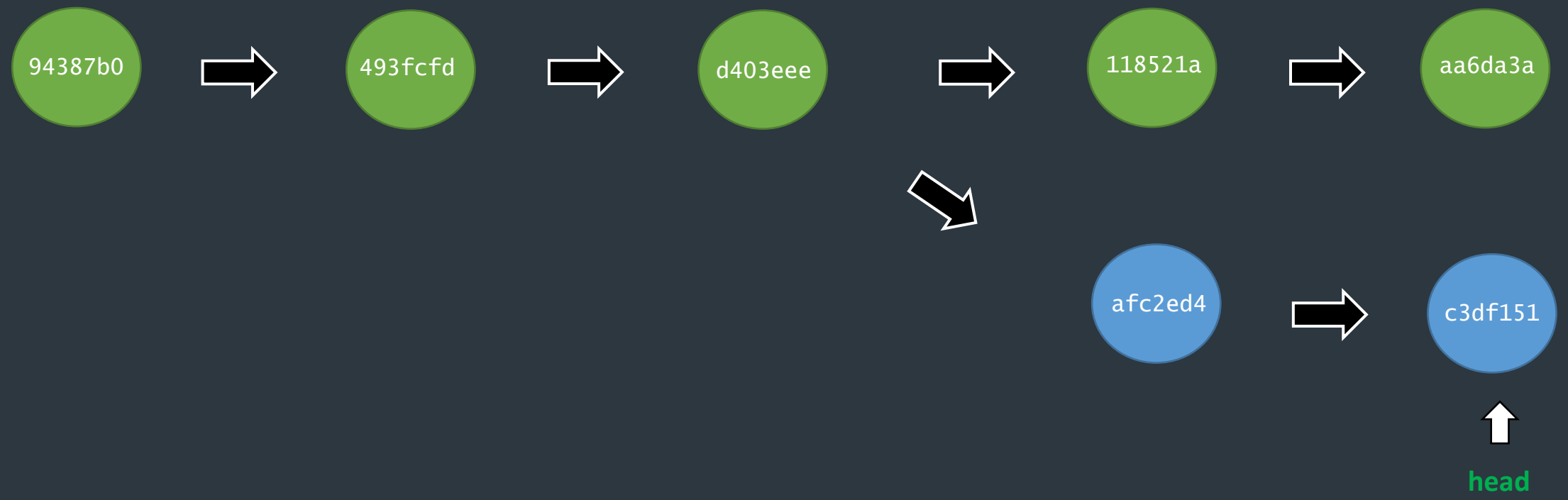
git 명령어 모음 1

git init	initialize git repo
git status	check git status in git repo
git add <file name>	add the file to Staging Area
git add .	add all changed files to Staging Area
git commit –m “commit log”	commit your change
git log	commit log 확인
git log --oneline	simple commit logs 확인
git checkout <commit-hash>	이전 commit으로 가기
git checkout master	현재 commit으로 돌아가기

What is Branch

When to use branch

- project에서 new feature를 만들 때
- project에서 bug fix를 할 때
- master branch를 최종 version으로 사용할 때



Exercise 2

Exercise 1의 repository에서 작업

1. new branch 생성
2. new branch로 이동
3. text.txt 파일 수정하고 commit
4. master branch로 이동
5. text.txt 파일 수정하고 commit

```
Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git branch newBranch

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git branch
* master
  newBranch

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git checkout newBranch
Switched to branch 'newBranch'

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (newBranch)
$ vi text.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (newBranch)
$ git add .

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (newBranch)
$ git commit -m "add change"
[newBranch 34d606a] add change
 1 file changed, 3 insertions(+)

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (newBranch)
$ git switch master
Switched to branch 'master'

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ vi text.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git add .

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git commit -m "add change"
[master 01b7262] add change
 1 file changed, 3 insertions(+)
```

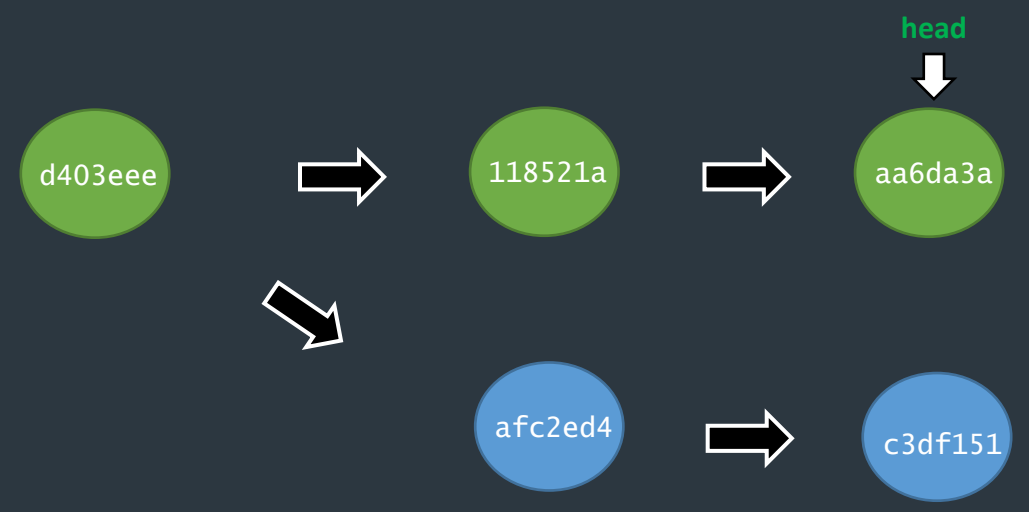
git 명령어 모음 2

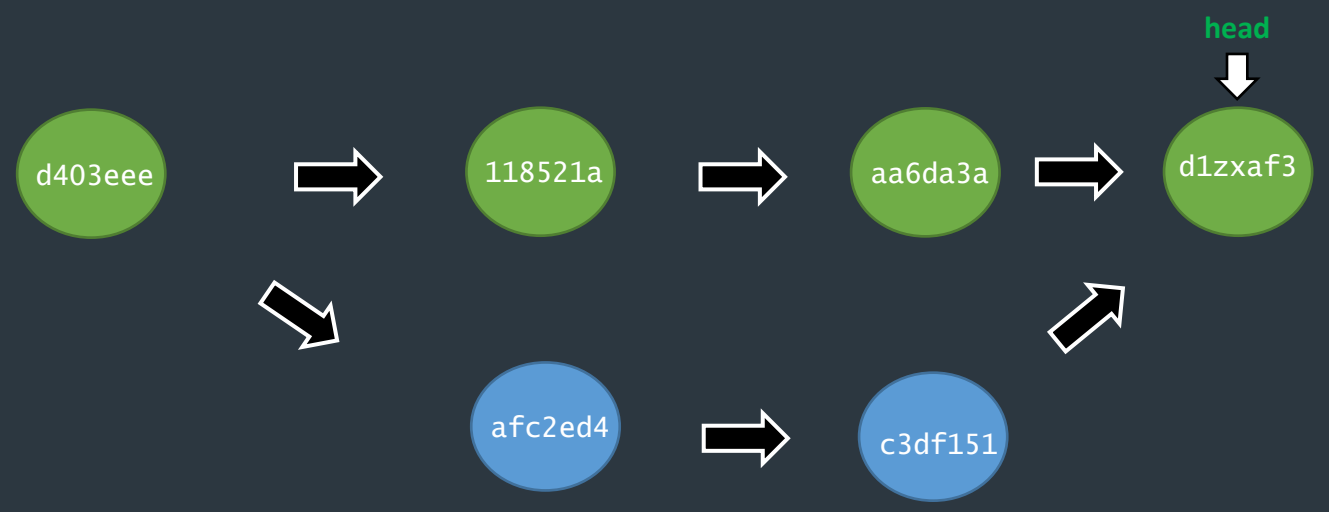
git branch	생성된 branch 확인
git checkout <branch name>	해당 branch로 이동
git switch <branch name>	git checkout과 동일
git switch –c <branch name>	branch를 생성하고 이동

Merge

When to merge two branches

- 양쪽의 branch에서 작업한 독립적인 feature를 합칠 때





Merge Conflict

Merge Conflict는 두 개의 branch에서 같은 파일의 내용이 다를 때 발생한다.

conflict가 발생하는 코드를 수정하고 git add와 commit을 통해 conflict를 해결해야한다.

```
<<<<<<< HEAD
I have 2 cats
I also have chickens
=======
I used to have a dog :(
>>>>>>> bug-fix
```

Exercise 3

Exercise 2의 repository에서 작업

1. master branch에서 git merge newBranch
2. resolve merge conflict
3. git add & git commit
4. newBranch 삭제

MINGW64:/c/Users/Baam/Desktop/Git Exercise/ex1

```
Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git merge newBranch
Auto-merging text.txt
CONFLICT (content): Merge conflict in text.txt
Automatic merge failed; fix conflicts and then commit the result.

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master|MERGING)
$ cat text.txt
Hello

<<<<<<< HEAD
My name is Baam.
I like Java and JS
=======
My name is Beomseok.
I like python and C++
>>>>>>> newBranch

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master|MERGING)
$ vi text.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master|MERGING)
$ git status
On branch master
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:   text.txt

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

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master|MERGING)
$ git add .

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master|MERGING)
$ git commit -m "merge text file"
[master 3d50d91] merge text file

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git status
On branch master
nothing to commit, working tree clean
```

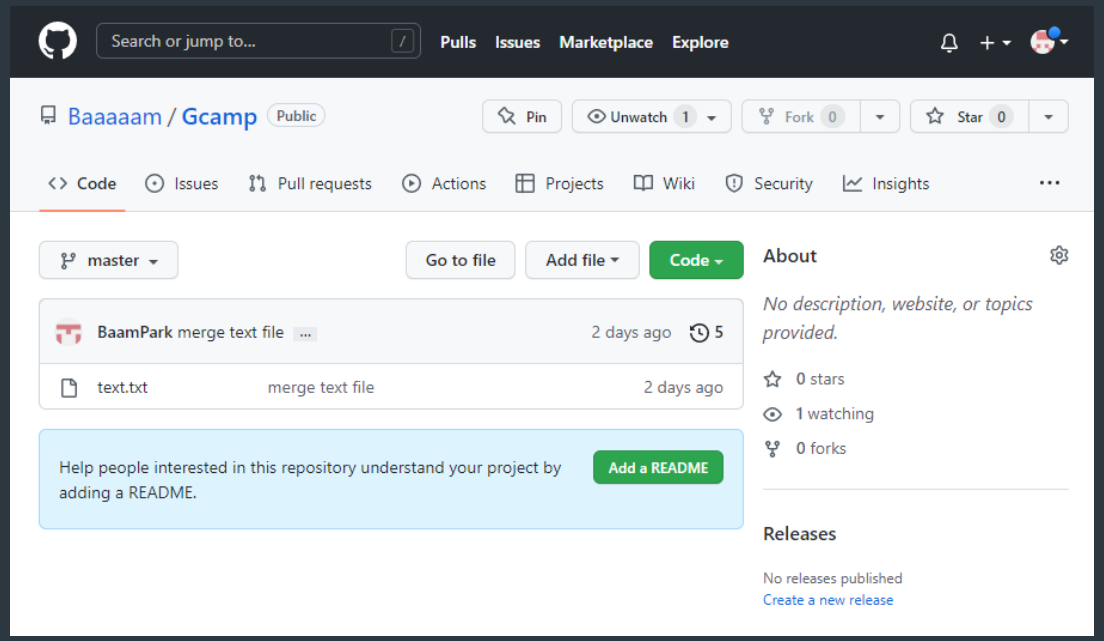
git 명령어 모음 3

git merge <branch name>	현재 branch에서 해당 branch와 merge
git branch –d <branch-name>	branch를 삭제

Collaboration on Github

Github

- 협업을 위한 Cloud Service
- local에서 작업한 repository를 Github에 업데이트
- Github에서 생성된 repository를 local에서 작업



Exercise 4

1. Github 계정 생성 (git local email과 같은 이메일로 생성)
2. New repository 생성
3. git remote add 명령어로 github repo와 local repo를 연결
4. git push 명령어로 local repo를 업데이트

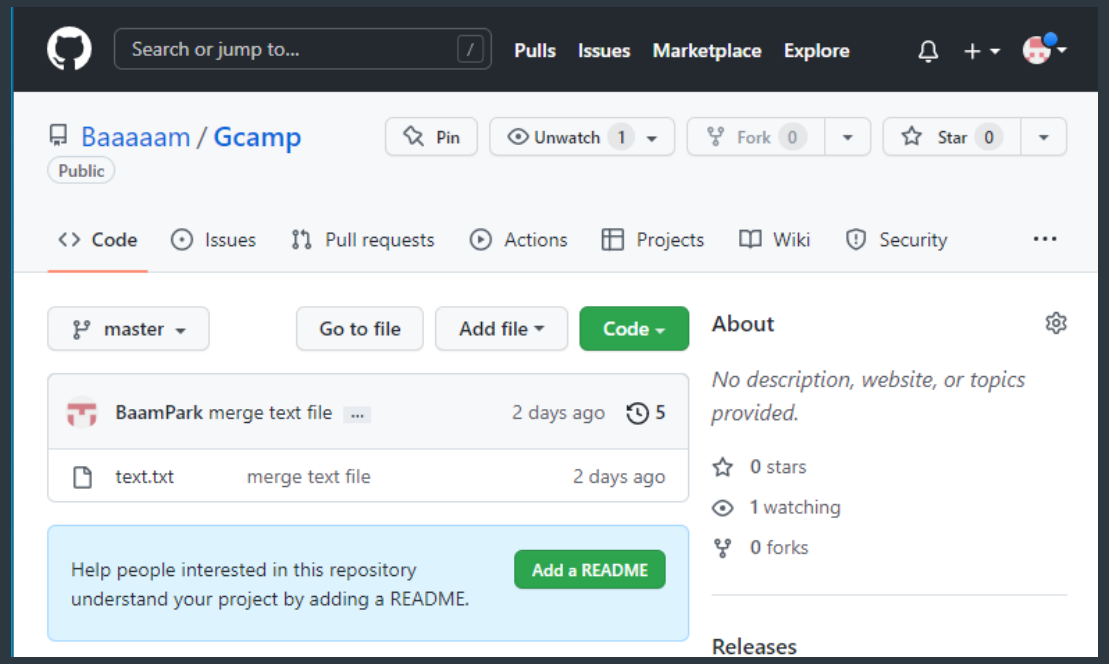
git 명령어 모음 4

git remote add <remote name> <http>	local repo와 github repo를 연결 (remote name은 보통 origin으로 명명)
git remote -v	local repo와 연결된 remote 보기
git remote remove <remote name>	해당 remote 삭제
git push <remote name> <branch name>	해당 branch를 remote에 업데이트

Exercise 4

MINGW64:/c/Users/Baam/Desktop/Git Exercise/ex1

```
Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git remote add origin https://github.com/Baaaaam/Gcamp.git
error: remote origin already exists.

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git remote -v
origin  https://github.com/Baaaaam/Gcamp.git (fetch)
origin  https://github.com/Baaaaam/Gcamp.git (push)

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git push origin master
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (15/15), 1.14 KiB | 233.00 KiB/s, done.
Total 15 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Baaaaam/Gcamp.git
 * [new branch]      master -> master
```



git fetch & git pull

git pull = git fetch + git merge

update the remote tracking branch with the latest changes from the remote repository

update my current branch with whatever changes are on the remote tracking branch

git fetch

- Github repository에 새로운 commit이 있는 지 확인할 때
- Repository Owner가 Github상에서 commit을 할 때
- Collaborator가 new commit을 추가하고 git push를 할 때

Git pull

- Github repository을 local repository에 업데이트
- origin/master branch와 master branch를 merge
- merge conflict가 발생할 수 도 있음

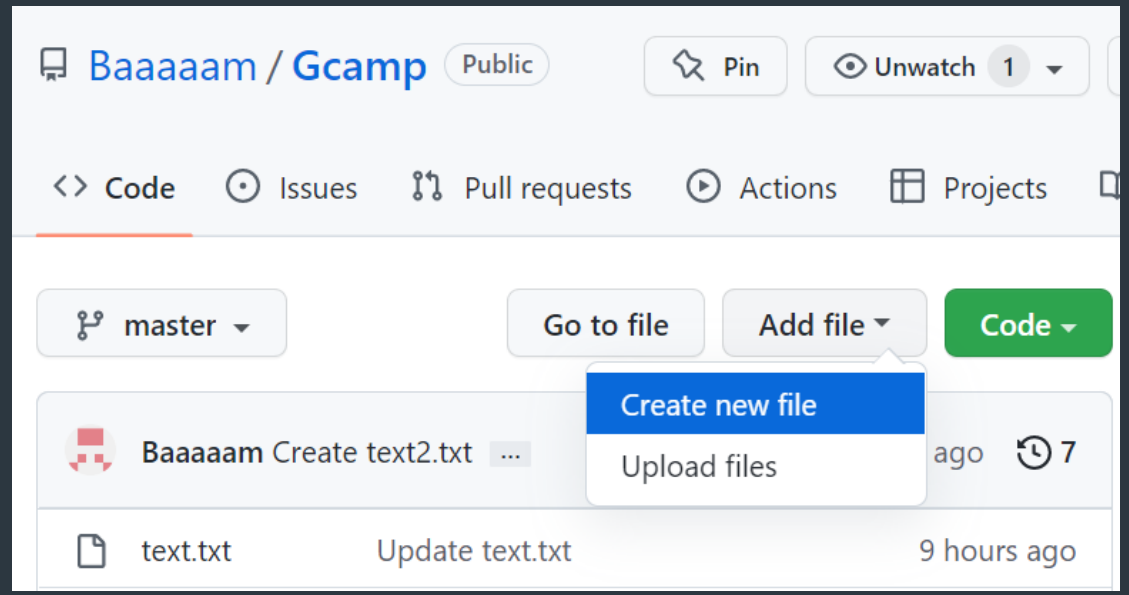
Exercise 5

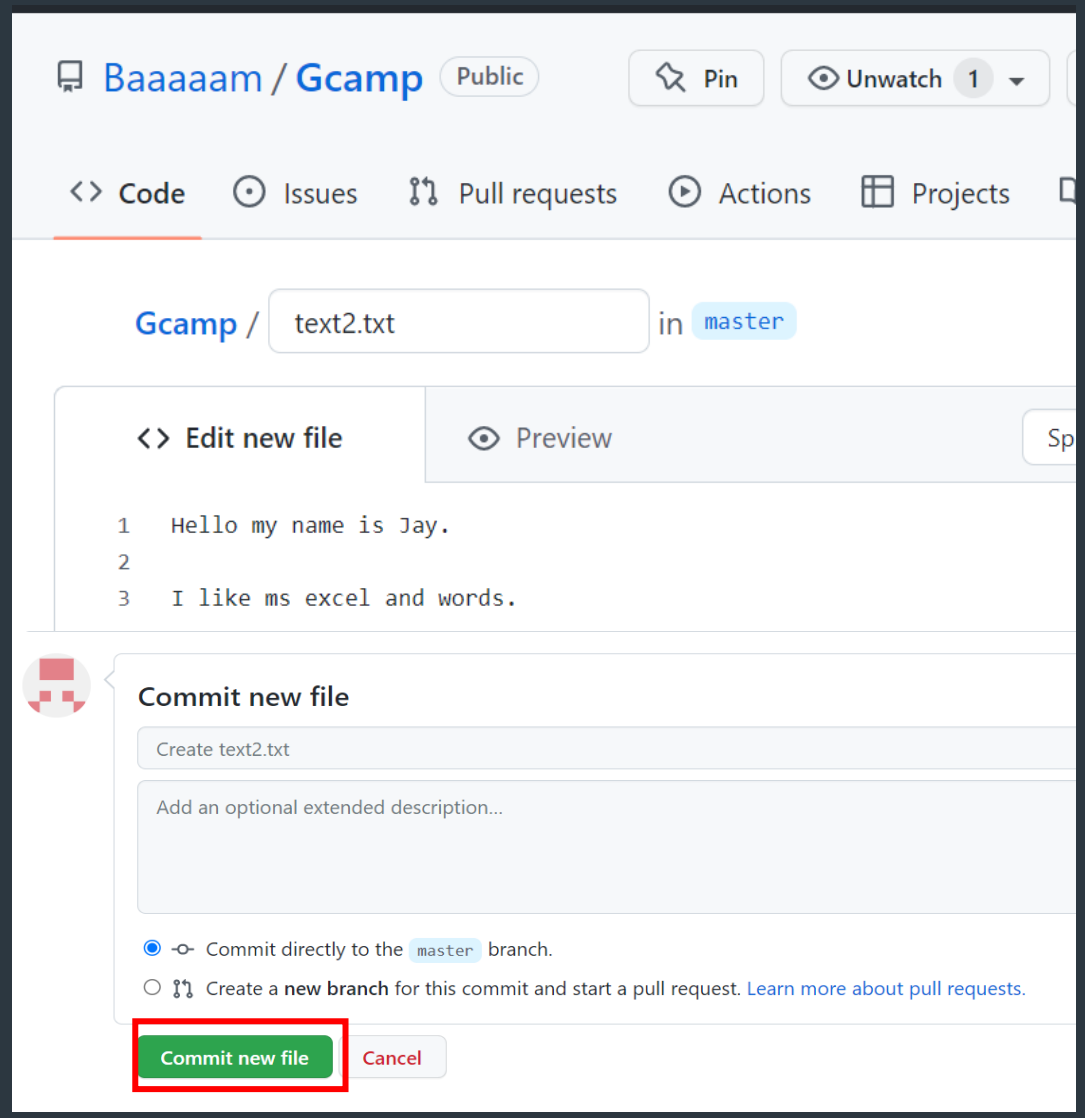
1. Github repository에서 텍스트 파일을 추가하고 commit
2. local repository에서 git fetch
3. local repository에서 origin/master branch로 이동
4. Github에서 추가되었던 파일을 확인
5. git pull

git 명령어 모음 5

git fetch <remote name>	Github의 repository를 remote branch로 local repo에 가져오기
git branch -r	가져온 remote branch들을 조회
git checkout <remote branch name>	remote branch로 이동
git pull <remote name> branch	local branch와 remote branch를 merge

Exercise 5


Baaaaam / Gcamp
Public
Pin
Unwatch 1
Code
Issues
Pull requests
Actions
Projects
master
Go to file
Add file
Code
Create new file
Upload files
Baaaaam Create text2.txt
ago
7
text.txt
Update text.txt
9 hours ago


Baaaaam / Gcamp
Public
Pin
Unwatch 1
Code
Issues
Pull requests
Actions
Projects
Gcamp /
text2.txt
in
master
Edit new file
Preview
Hello my name is Jay.
I like ms excel and words.
Commit new file
Create text2.txt
Add an optional extended description...
Commit directly to the master branch.
Create a new branch for this commit and start a pull request. Learn more about pull requests.
Commit new file
Cancel

Exercise 5

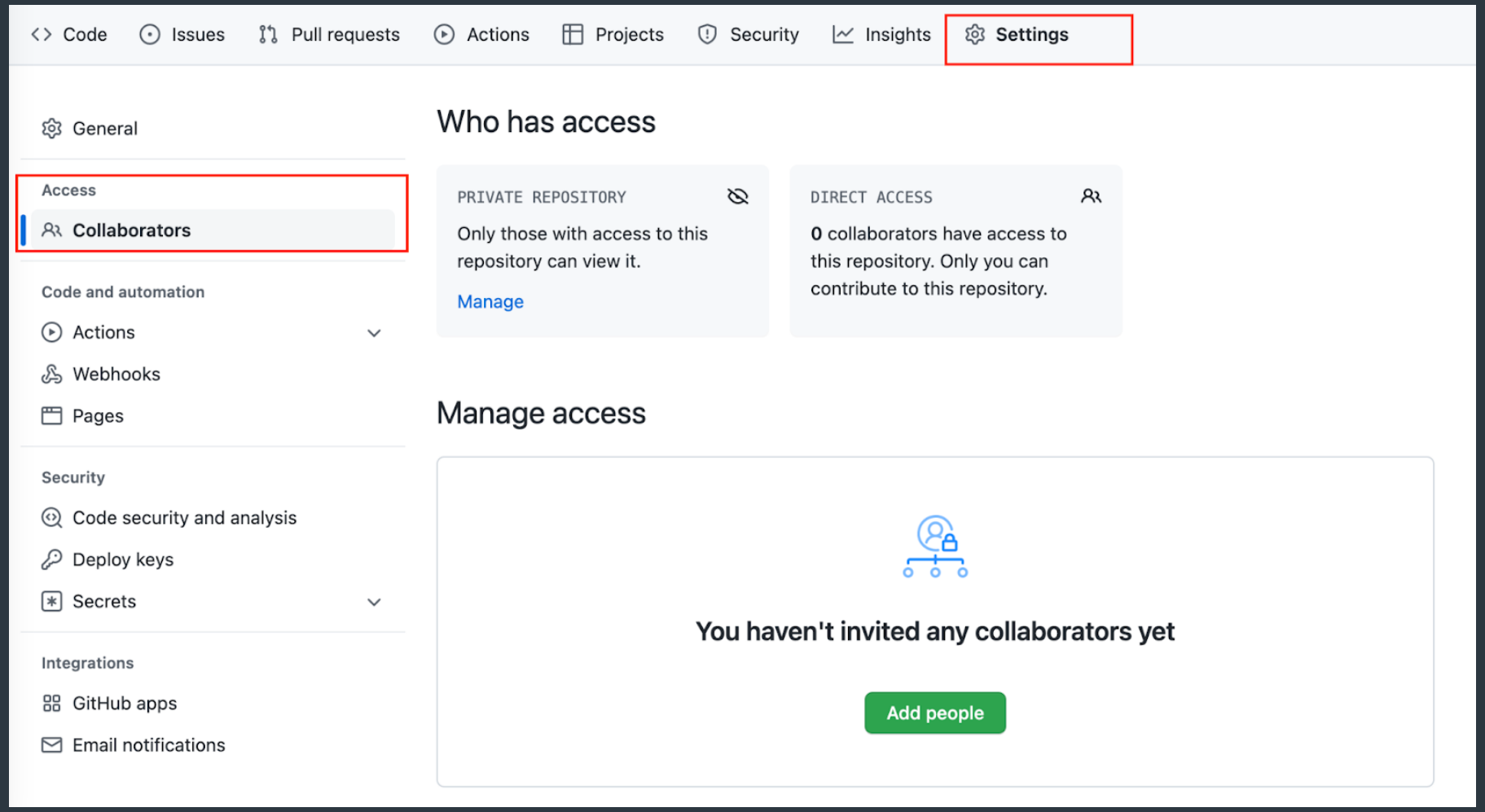
MINGW64:/c/Users/Baam/Desktop/Git Exercise/ex1

```
Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git fetch origin
From https://github.com/Baaaaam/Gcamp
 * [new branch]      master     -> origin/master

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git branch -r
  origin/master

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git checkout origin/master
Note: switching to 'origin/master'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -
```

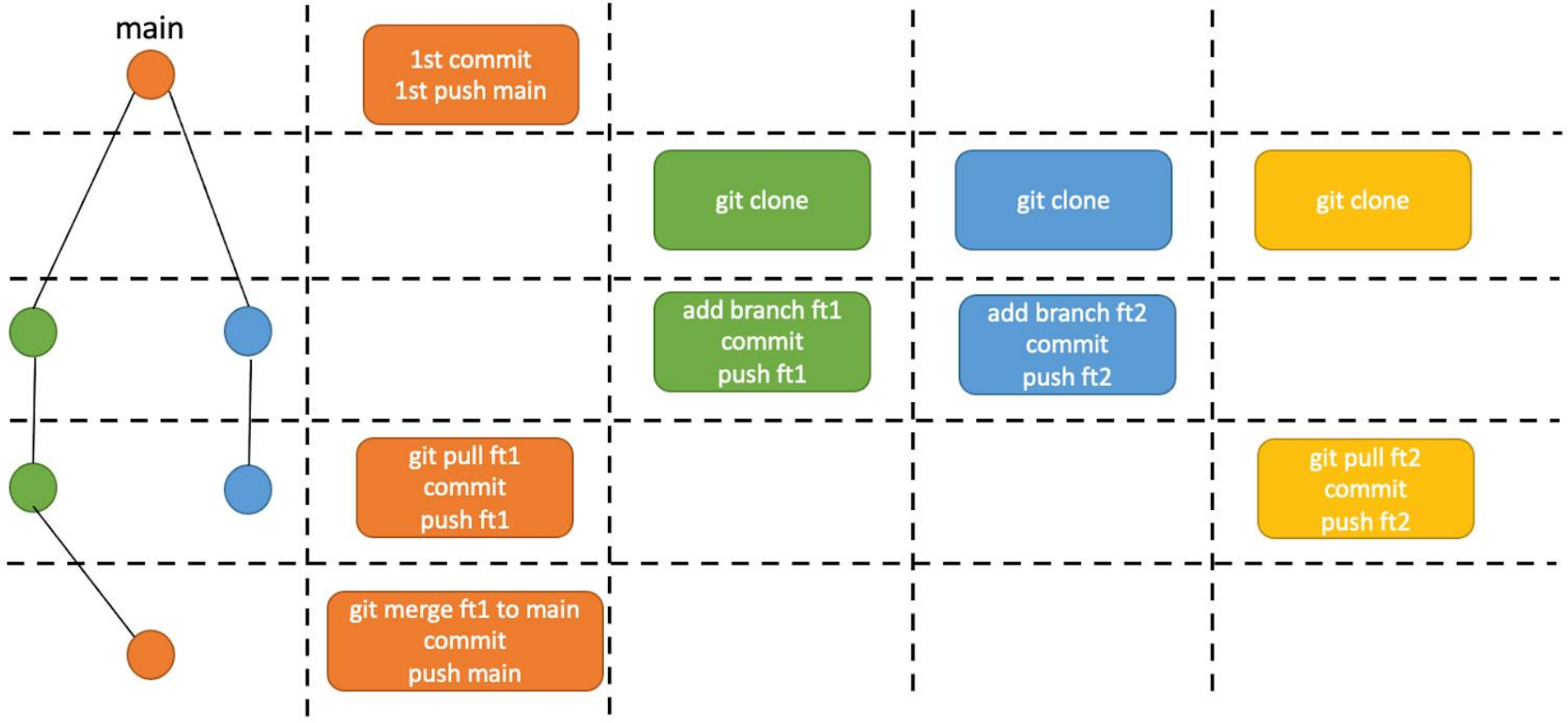
MINGW64:/c/Users/Baam/Desktop/Git Exercise/ex1

```
Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 02273cd Create text2.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 ((02273cd...))
$ ls
text.txt  text2.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 ((02273cd...))
$ git switch master
Previous HEAD position was 02273cd Create text2.txt
Switched to branch 'master'

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ git pull origin master
From https://github.com/Baaaaam/Gcamp
 * branch            master     -> FETCH_HEAD
Updating 3d50d91..02273cd
Fast-forward
 text.txt  | 2 ++
 text2.txt | 4 ++++
 2 files changed, 6 insertions(+)
 create mode 100644 text2.txt

Baam@DESKTOP-4A292VV MINGW64 ~/Desktop/Git Exercise/ex1 (master)
$ |
```

Github repository에서 collaborators 추가하기



Github Collaboration Workflow

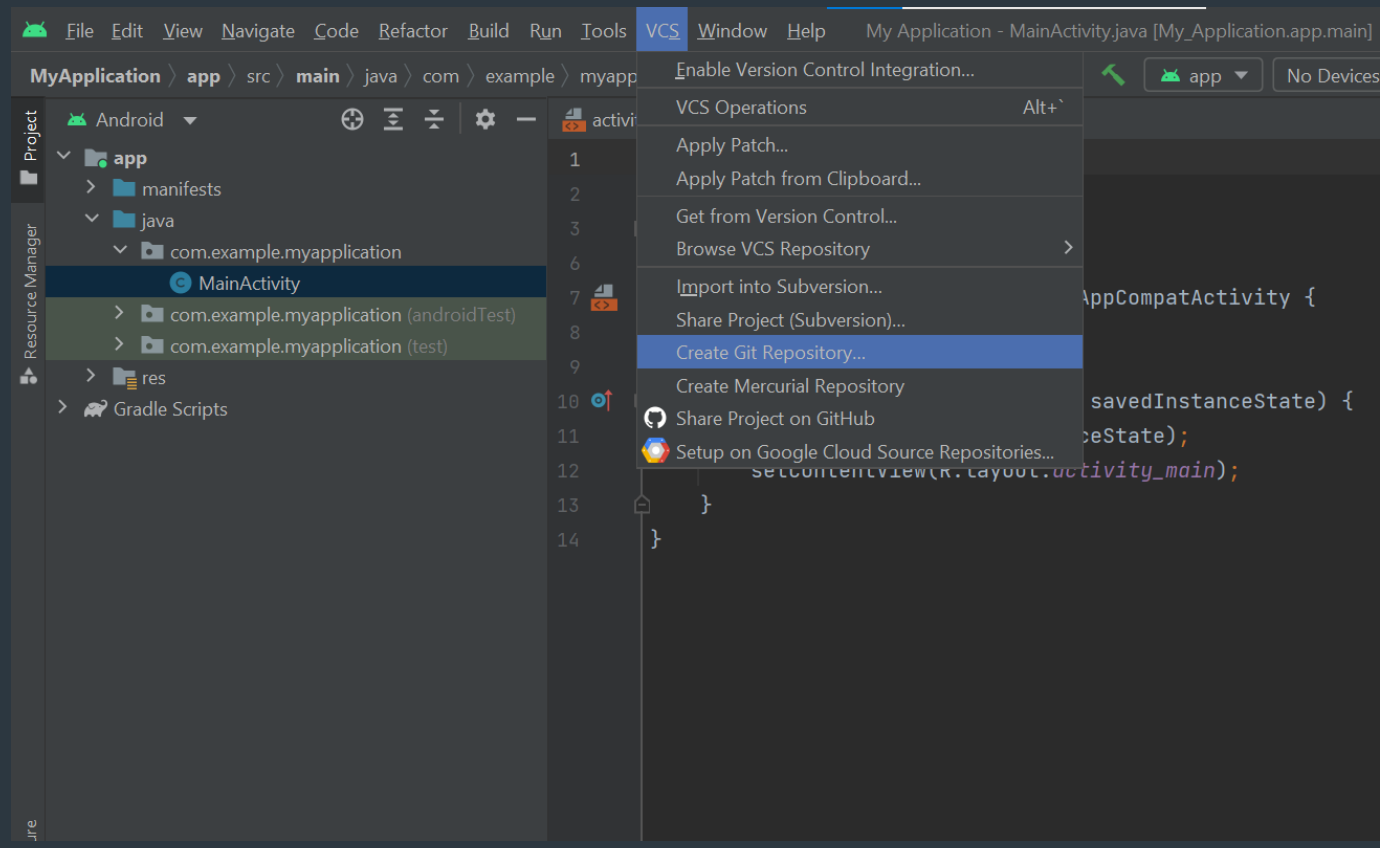


Beomseok Park – baam.park.95@gmail.com

VCS on Android Studio

VCS를 통해 Git Repository 생성

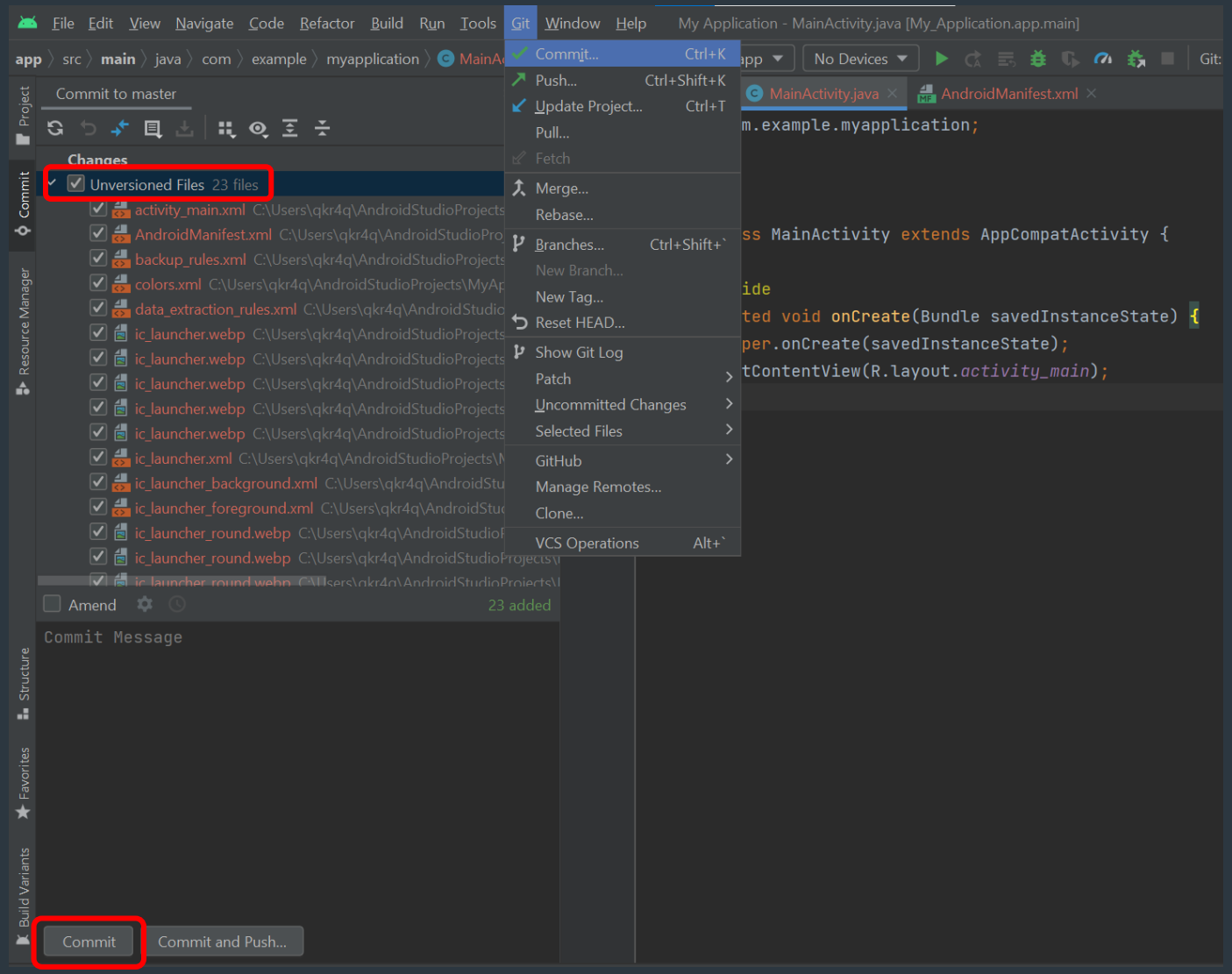
1. 상단 바의 VCS 클릭
2. Create Git Repository 클릭
3. git repository 위치를 Project/app/src/main으로 설정



ASUS IoT

git add & commit

1. 상단 바의 git 클릭
2. Commit 클릭
3. 왼쪽에 열린 Changes의 Unversioned Files 확인
4. 모든 파일 체크
5. commit 메세지 작성
6. 왼쪽 하단의 commit 버튼 클릭
7. checks failed 메세지가 뜨면 한번더 commit 클릭
8. 빨간 색으로 표시된 파일들이 흰색으로 표시됨

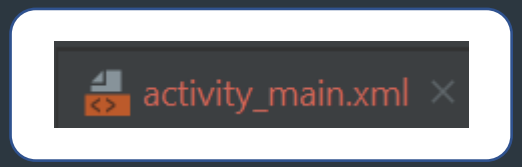


Beomseok Park – baam.park.95@gmail.com

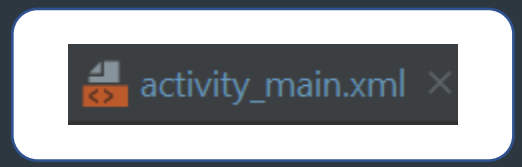
ASUS IoT

git status에 따른 색상 변화

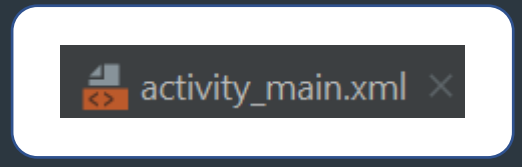
not added



edited

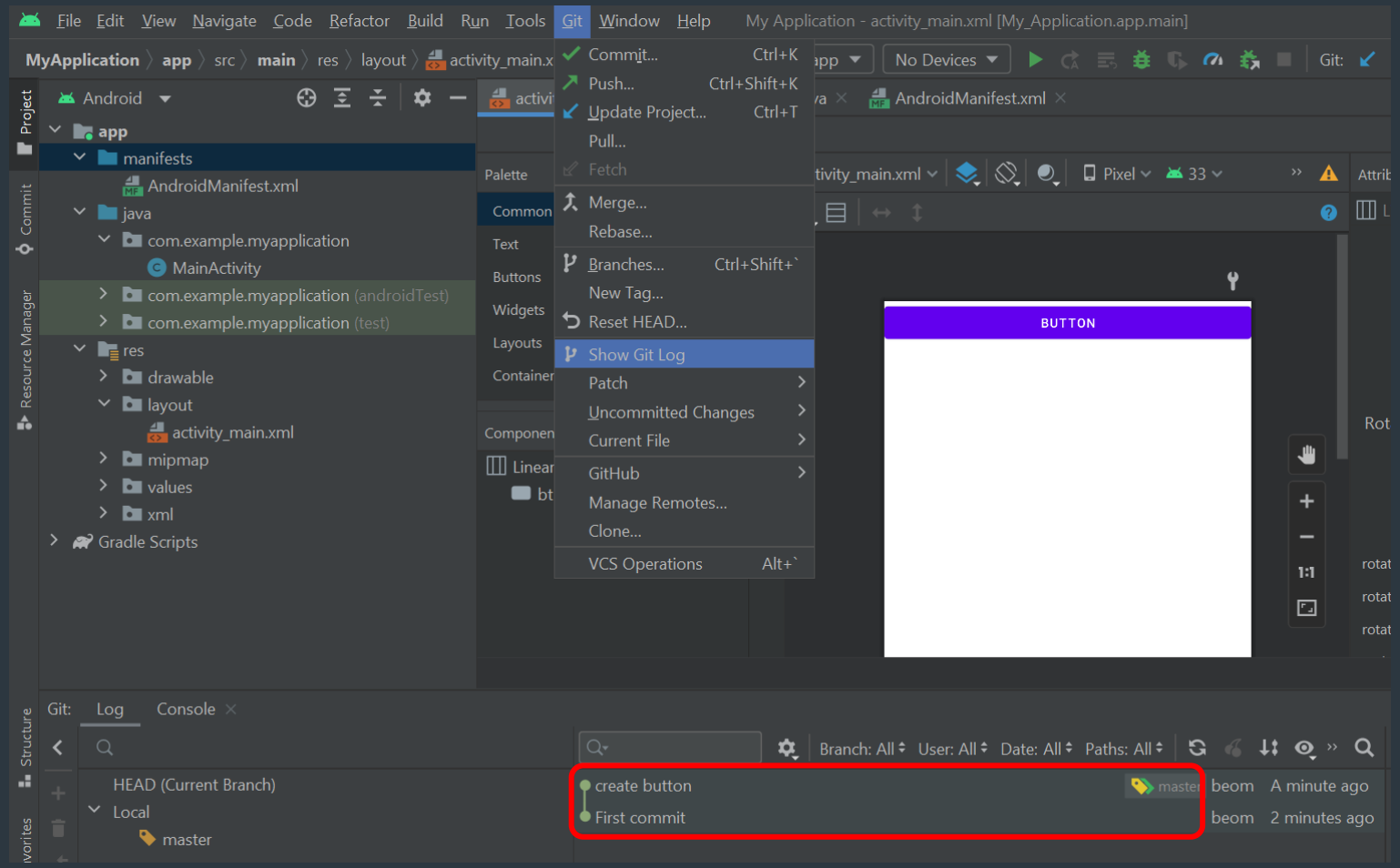


committed



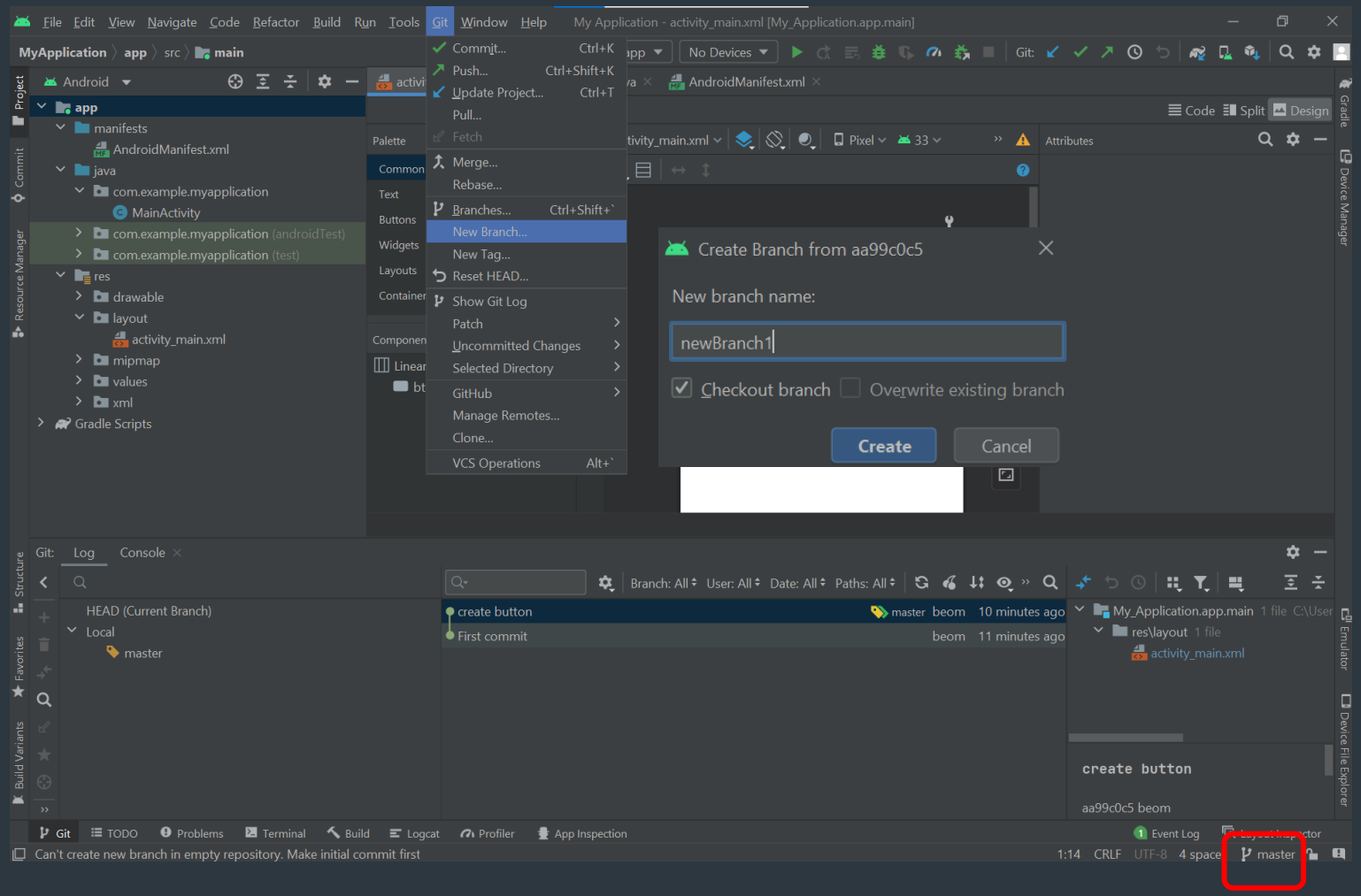
Git Log 확인 및 체크아웃

1. 상단 바의 git 클릭
2. Show Git Log 클릭
3. 하단 부에서 나타나는 git log 확인
4. checkout 하고자 하는 commit 우클릭
5. Checkout Revision ‘commit hash’ 클릭



git branch 생성

1. 상단 바의 Git 클릭
2. New Branch 클릭
3. branch name 설정 후 create
4. 오른쪽 하단의 git branch icon 클릭
5. 원하는 branch 선택 후 checkout 클릭



Beomseok Park – baam.park.95@gmail.com

ASUS IoT