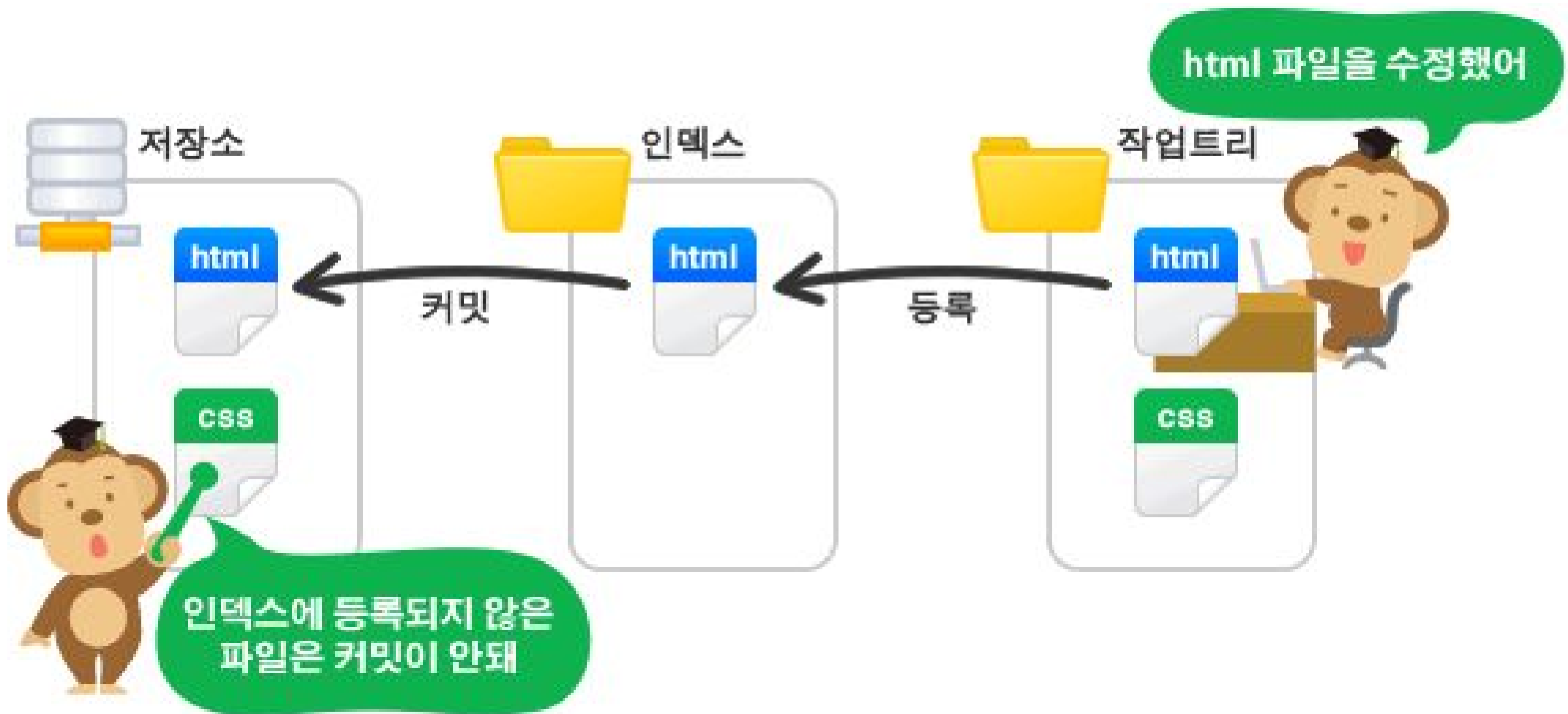



Git

*

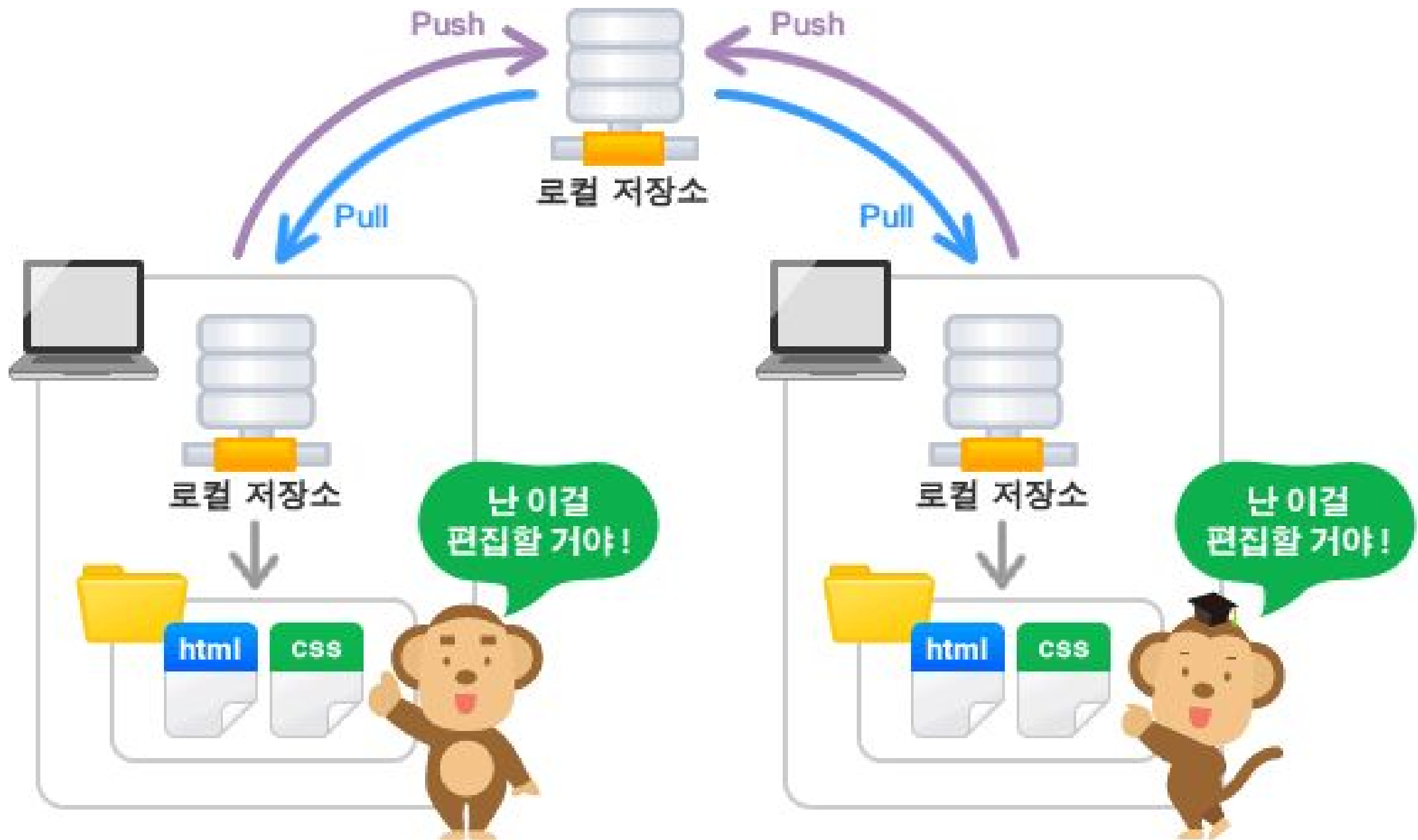
stage



Try - Git(Local Commit & Stash)



```
~$ sudo apt-get install git gitk
~$ git --version
~$ mkdir eclipse-workspace_html
~$ cd eclipse-workspace_html
~$ git init          → git 초기화
~$ git status        → 상태 확인
~$ git config --global user.email "otter.oh@gmail.com" → Remote Account
~$ git config --global user.name "SanghunOh"
~$ git add .          → 변경사항 index 등록
~$ git commit -m "first commit" → Apply Local Worktree
~$ git log            → 등록 내용
~$ touch my_stash.txt
~$ git add .
~$ git stash save "my_stash_temp" → 임시 저장
~$ git stash list
~$ ls
~$ git stash apply stash@{1}
~$ git stash pop
~$ git stash drop stash@{0}
```

Git(Remote)



Try(1) - GitHub(Remote, Create Git Server)

- ❖ <https://github.com/> → Login
- ❖ Click 'New Repository' -> Repository name : gitTest → Save
- ❖ Setting -> Collaborators > Typing ID Name > Add Collaborators



 GitHub, Inc. [US] | <https://github.com/new> 

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner


Repository name

 SanghunOh ▾ / gitRepo 


Great repository names are short and memorable. Need inspiration? How about [bug-free-octo-meme](#).

Description (optional)

Test Git Repository


☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ 

Create repository

Try(2) - Git(Remote, push/pull Source)

❖ Copy URI in TextBox (ex.https://github.com/SanghunOh/gitTest.git)

~\$ git remote add original https://github.com/SanghunOh/gitTest.git

~\$ git push -u original master

Username for 'https://github.com': SanghunOh

Password for 'https://SanghunOh@github.com': *****

~\$ touch sample.txt

~\$ git add sample.txt

~\$ git commit -m "add sample.txt file"

~\$ git push

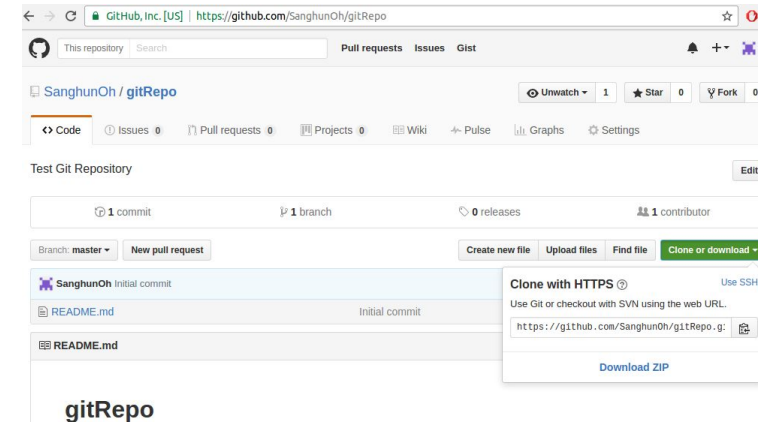
❖ Modify README.md on Github

~\$ git pull original master

Conflit on

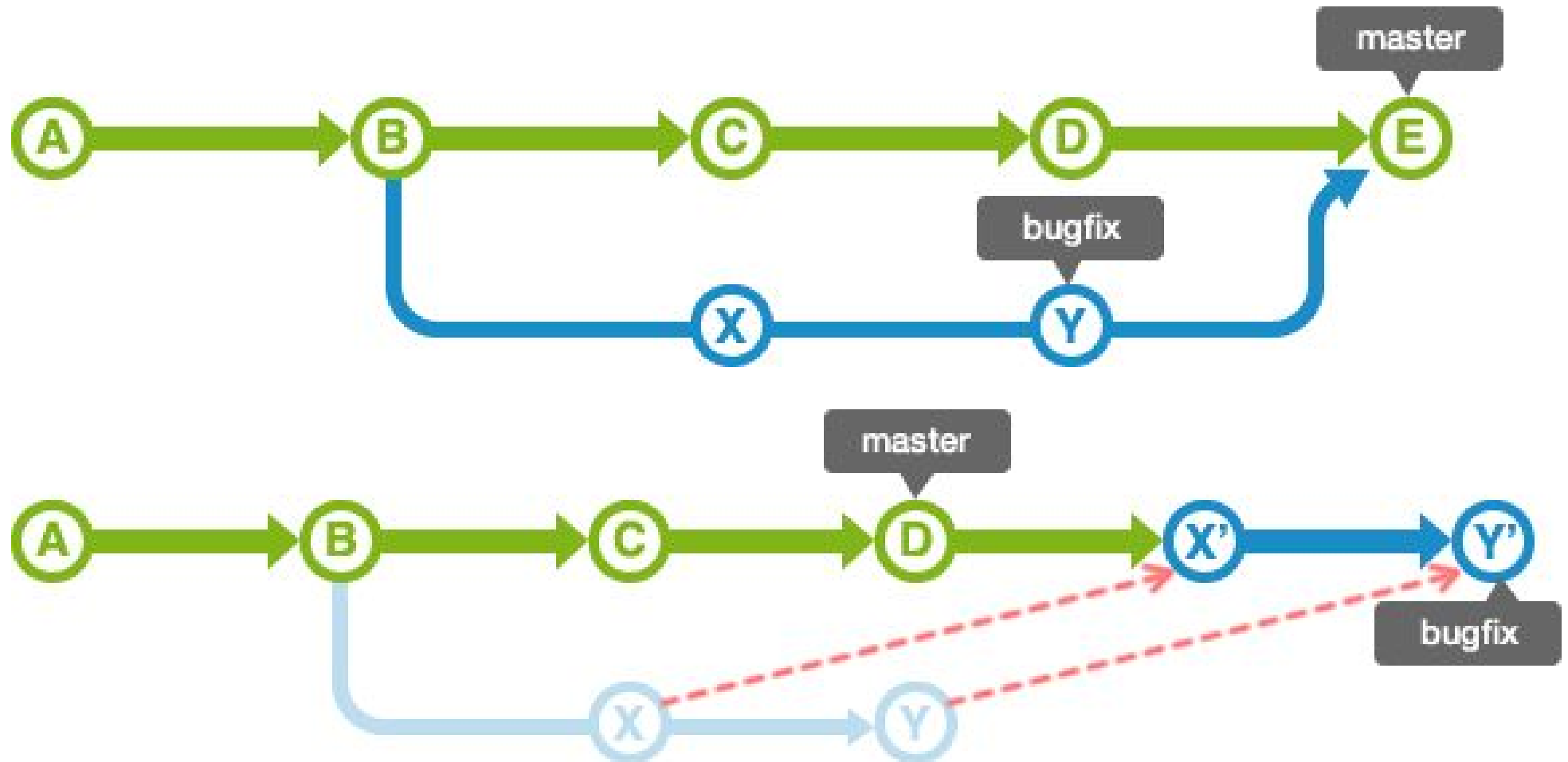
~\$ \$ git log --graph --oneline

~\$ git clone https://github.com/SanghunOh/lectureHtml.git ./Downloads/downGit



Git(Branch)

- ❖ Main branch : master(배포 가능본), develop(개발본)
- ❖ Feature branch : 새로운 기능 개발 및 버그 수정본.
- ❖ Release branch : 정상동작 확인 후 master와 병합, prefix 'release-'
- ❖ Hotfix branch : 배포 후 긴급 수정본, prefix 'hotfix-'



Try - Git(Branch)

~\$ git branch feature

~\$ git branch

~\$ git checkout feature → or git checkout -b <branch>

~\$ touch feature_file.txt

~\$ git add . ~\$ git commit -m 'add files' ~\$ ls

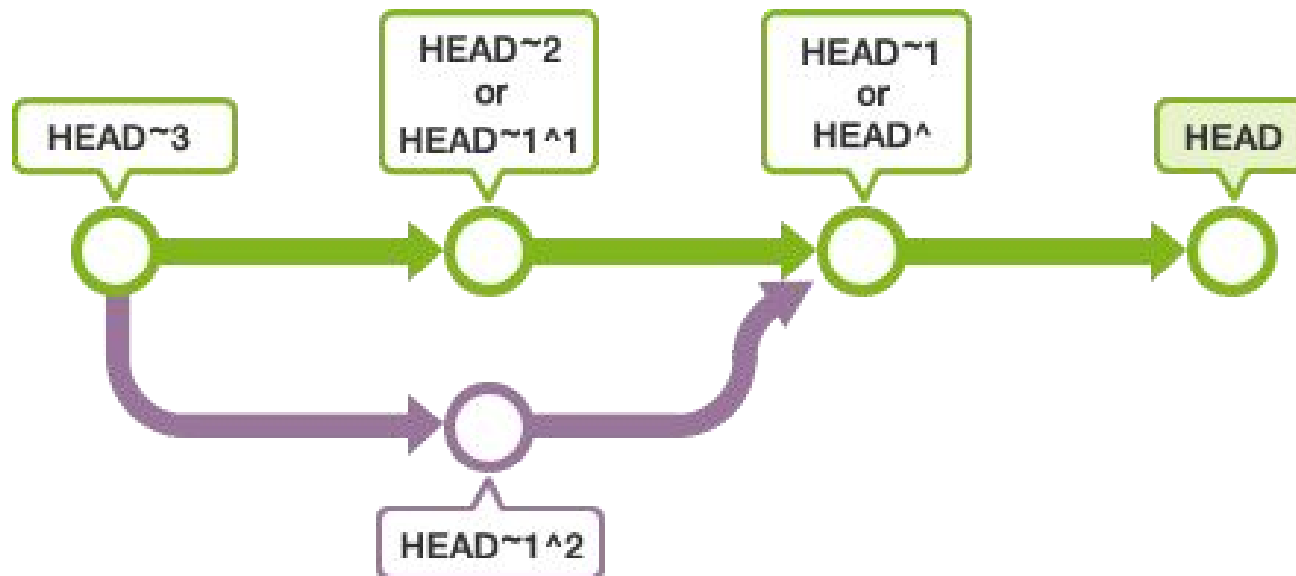
~\$ git checkout master

~\$ git merge feature → delete : ~\$ git branch -d feature

~\$ git reset --hard HEAD~

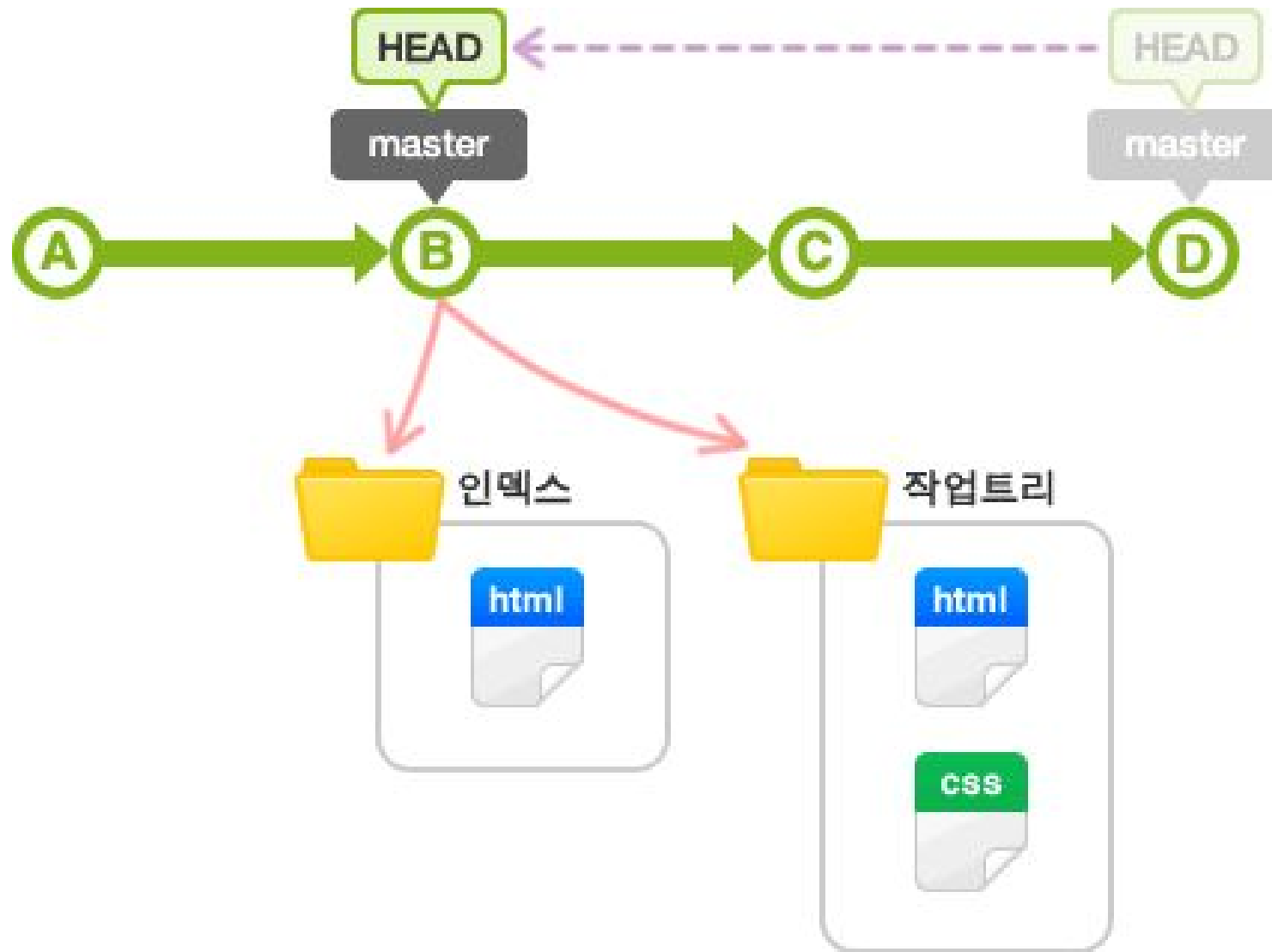
~\$ git tag -a ver0.1 -m 'version 0.1'

~\$ git tag -n → delete : ~\$ git tag -d version01



Git(Revert)

- ❖ **revert** : 특정 commit 삭제
- ❖ **reset** : 이전 commits Rollback(대상 : soft-commit, mixed-index, hard-worktree)
- ❖ **cherry-pick** : 다른 브랜치에서 지정한 커밋을 현재 브랜치로 복사

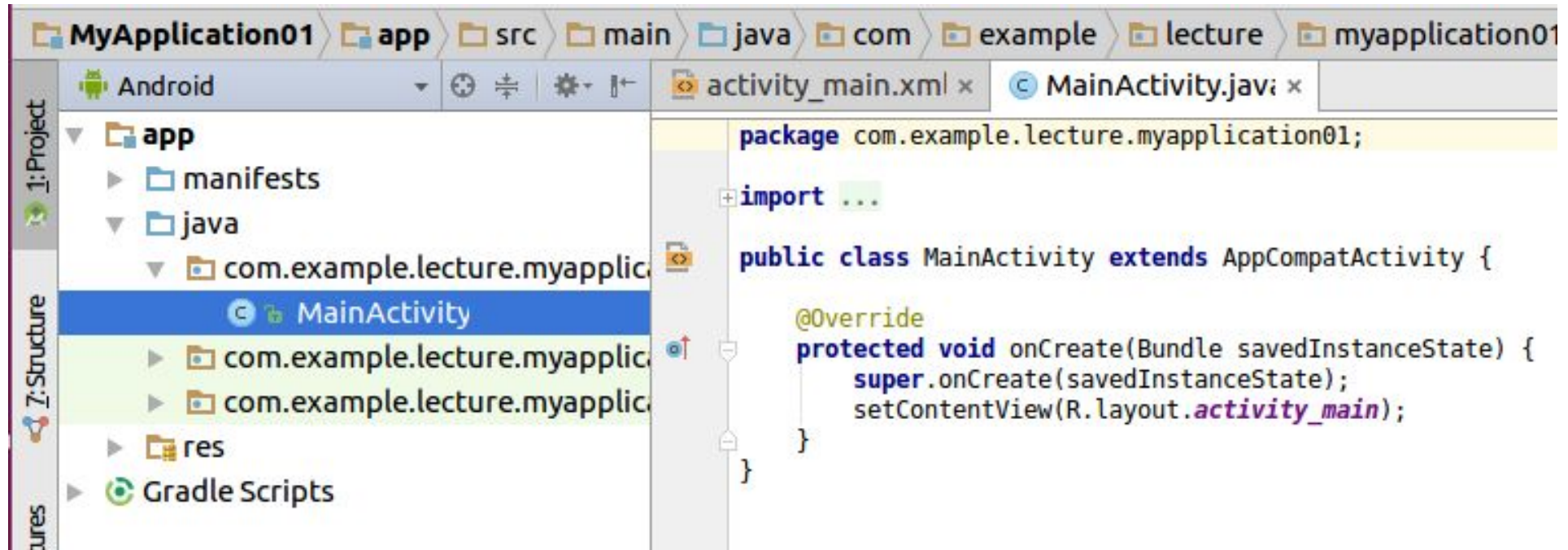


Try - Git(Revert)

```
~$ git init
~$ git add README.md
~$ git commit -m "first commit"
~$ git remote add origin https://github.com/SanghunOh/testGit.git
~$ git push -u origin master
~$ touch first.txt
~$ git add .
~$ git commit -m "first commit"
~$ git revert HEAD
~$ touch second.txt third.txt
~$ git add .
~$ git commit -m "files commit"
~$ git reset --hard HEAD~~
~$ git reset --hard ORIG_HEAD
~$ git log --graph
~$ git rebase -i HEAD~~~
modify pick → squesh
~$ git log --graph
```

Android Studio

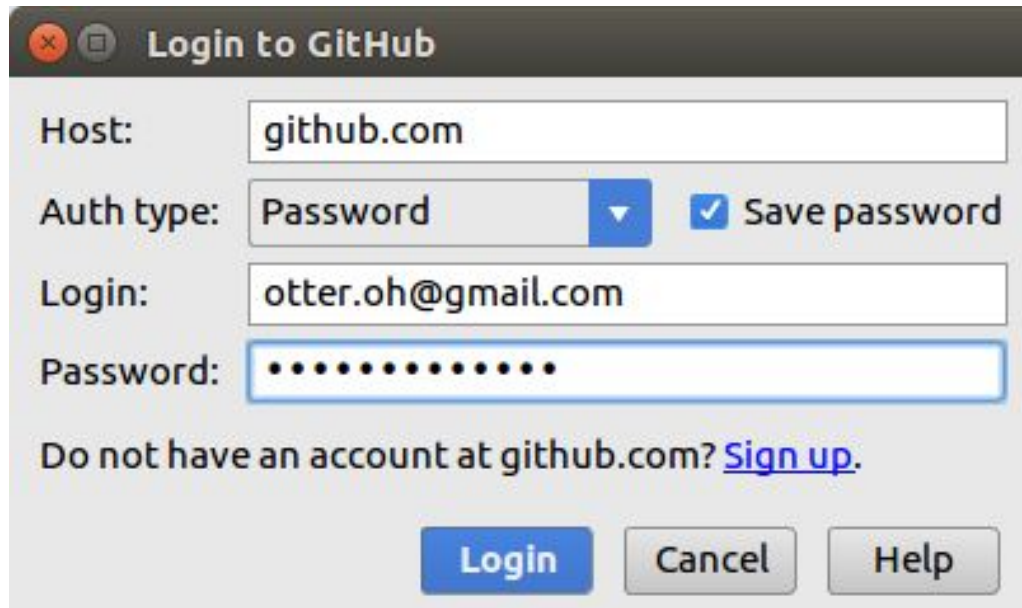
- ❖ Install git
 - `sudo apt -y install git`
- ❖ Create Project



- ❖ Shared Project
 - VCS > Import into Version Control > Share Project on GitHub

Android Studio

- ❖ Shared Project
 - VCS > Import into Version Control > Share Project on GitHub
- ❖ Login to GitHub



The 'Login to GitHub' dialog box contains the following fields and controls:

- Host:** A text field containing 'github.com'.
- Auth type:** A dropdown menu set to 'Password' with a blue checkmark icon to its right.
- Save password:** A checkbox that is checked, with the text 'Save password' to its right.
- Login:** A text field containing 'otter.oh@gmail.com'.
- Password:** A password field with masked characters (dots).
- Footer:** A link that says 'Do not have an account at github.com? [Sign up.](#)' and three buttons: 'Login' (blue), 'Cancel', and 'Help'.

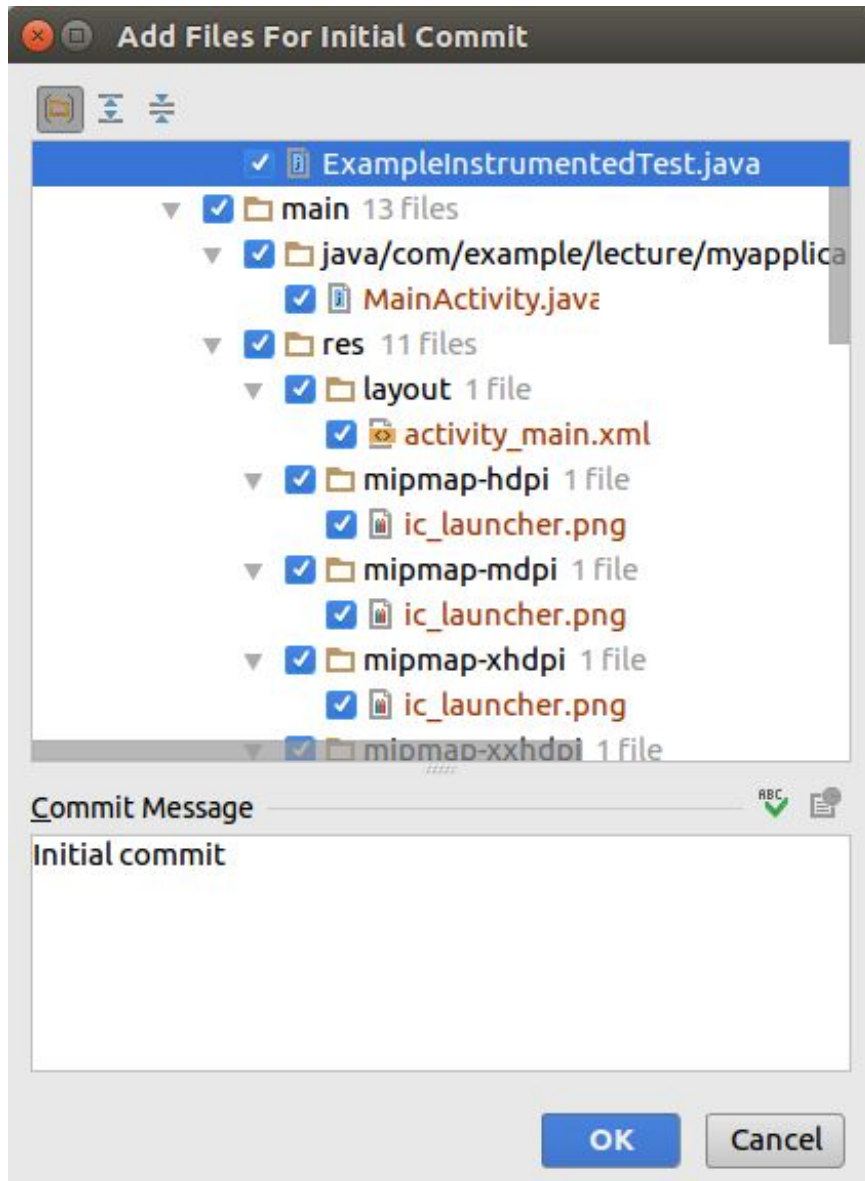


The 'Setup Master Password' dialog box contains the following fields and controls:

- Icon:** A blue padlock icon with a keyhole.
- Password:** A password field with masked characters (dots).
- Confirm password:** A password field with masked characters (dots).
- Instructions:** Text that reads 'Specify the new password for the password database. Leave blank to disable the master password protection.'
- Buttons:** 'Help', 'OK' (blue), and 'Cancel'.

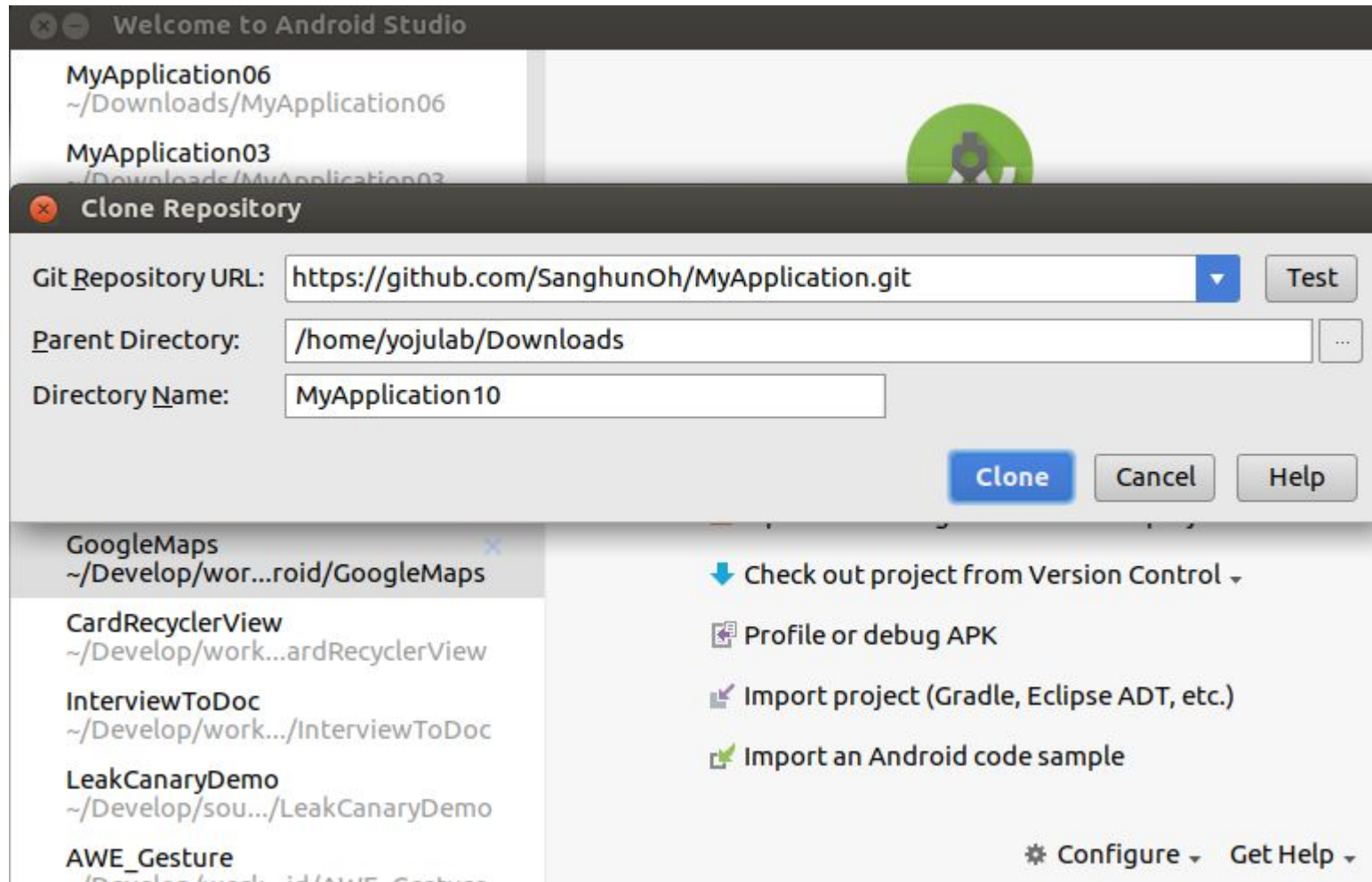
Android Studio

❖ Add Files For Initial Commit



Android Studio

❖ clone Project

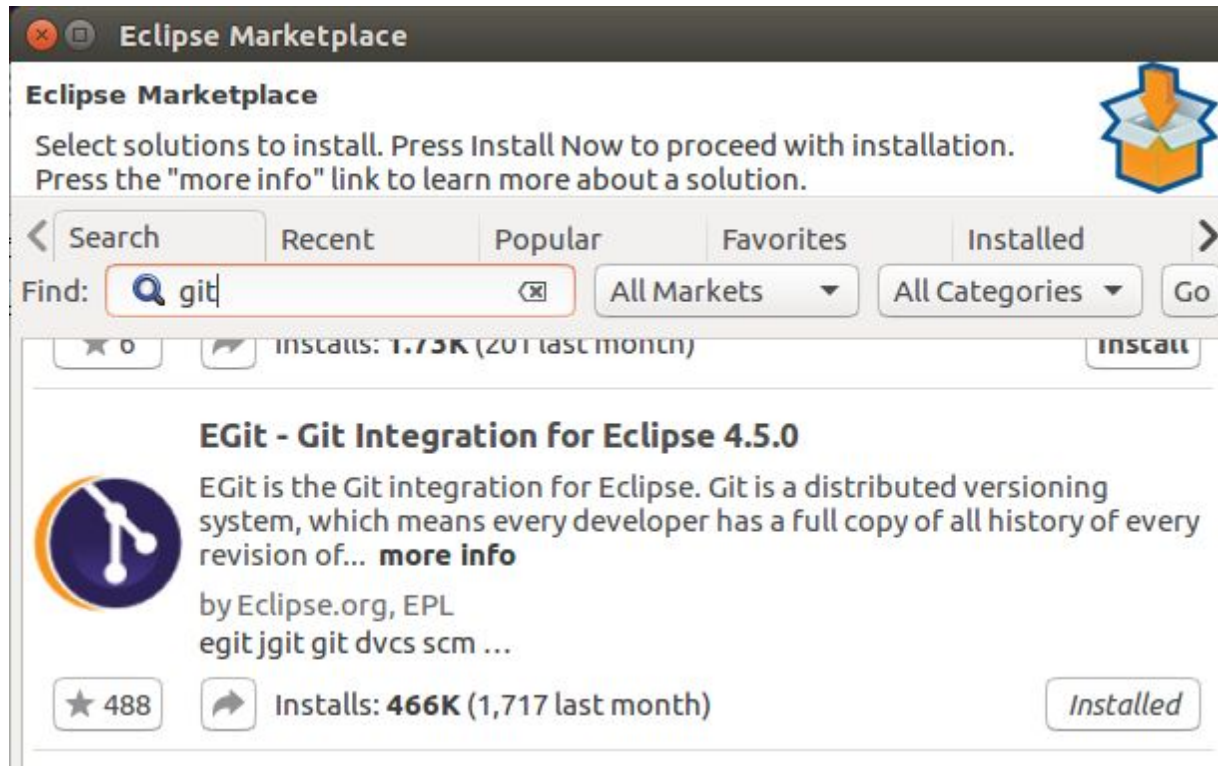


❖ Sync Project with Gradle files

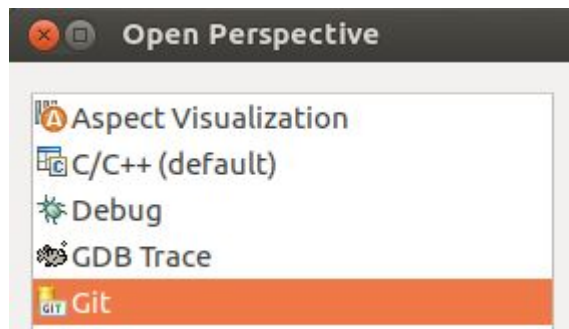


Eclipse - plugin Egit

❖ PlugIn EGit

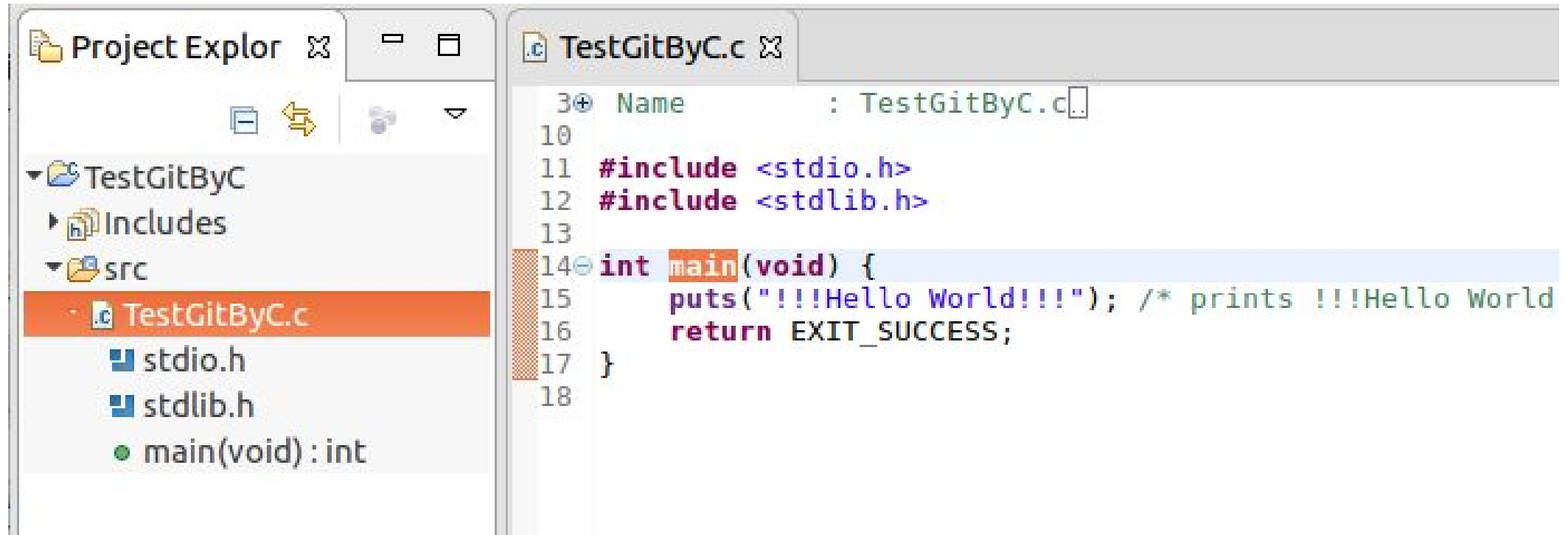


❖ Open Git From Perspective



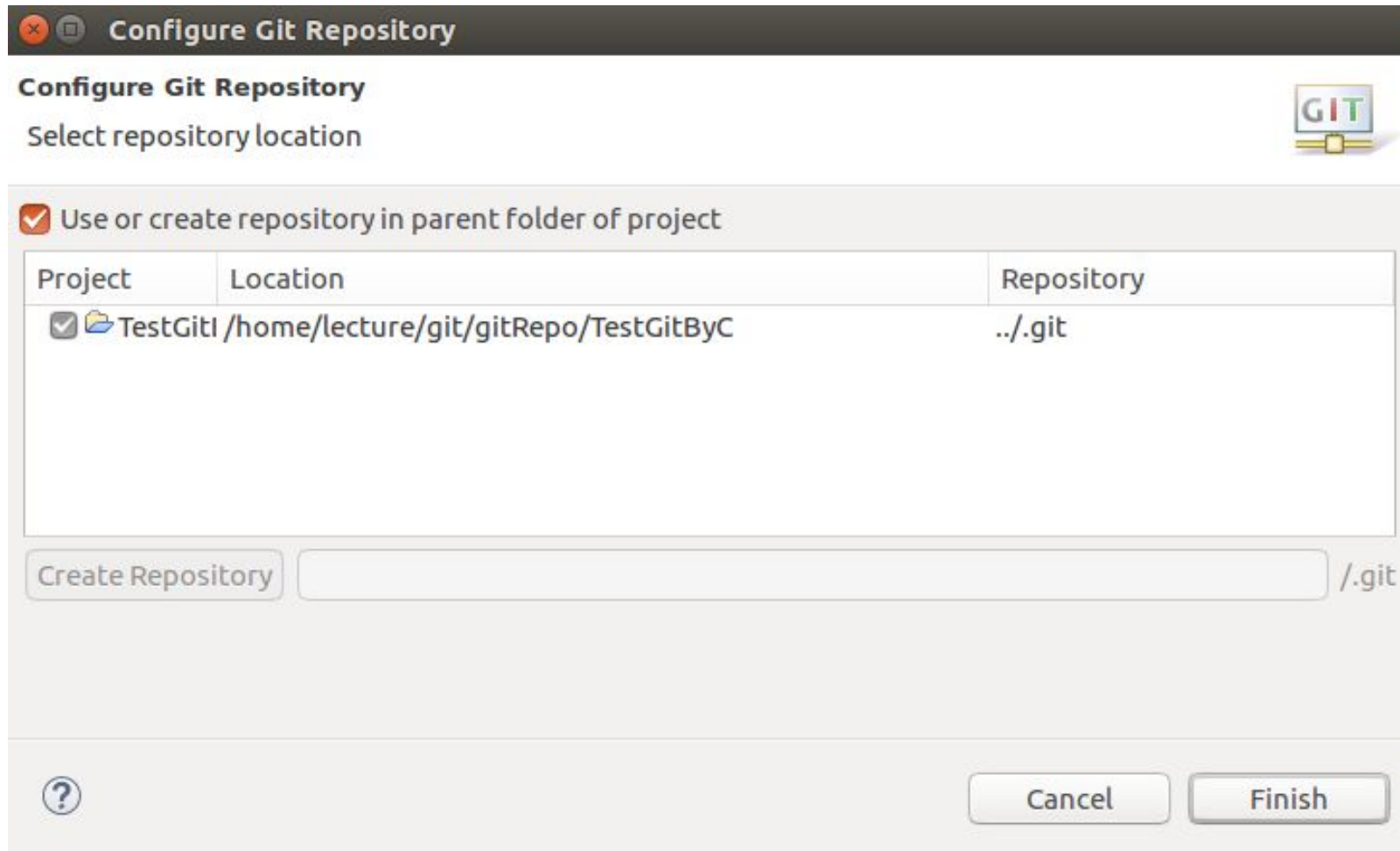
Eclipse - initial Project(1)

❖ Create Project



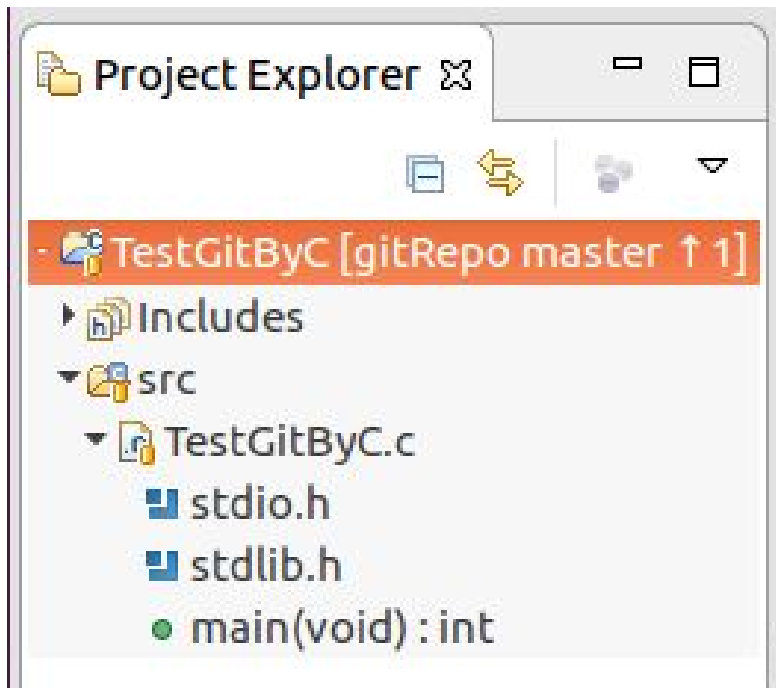
Eclipse - initial Project(2)

❖ Team > Share Project

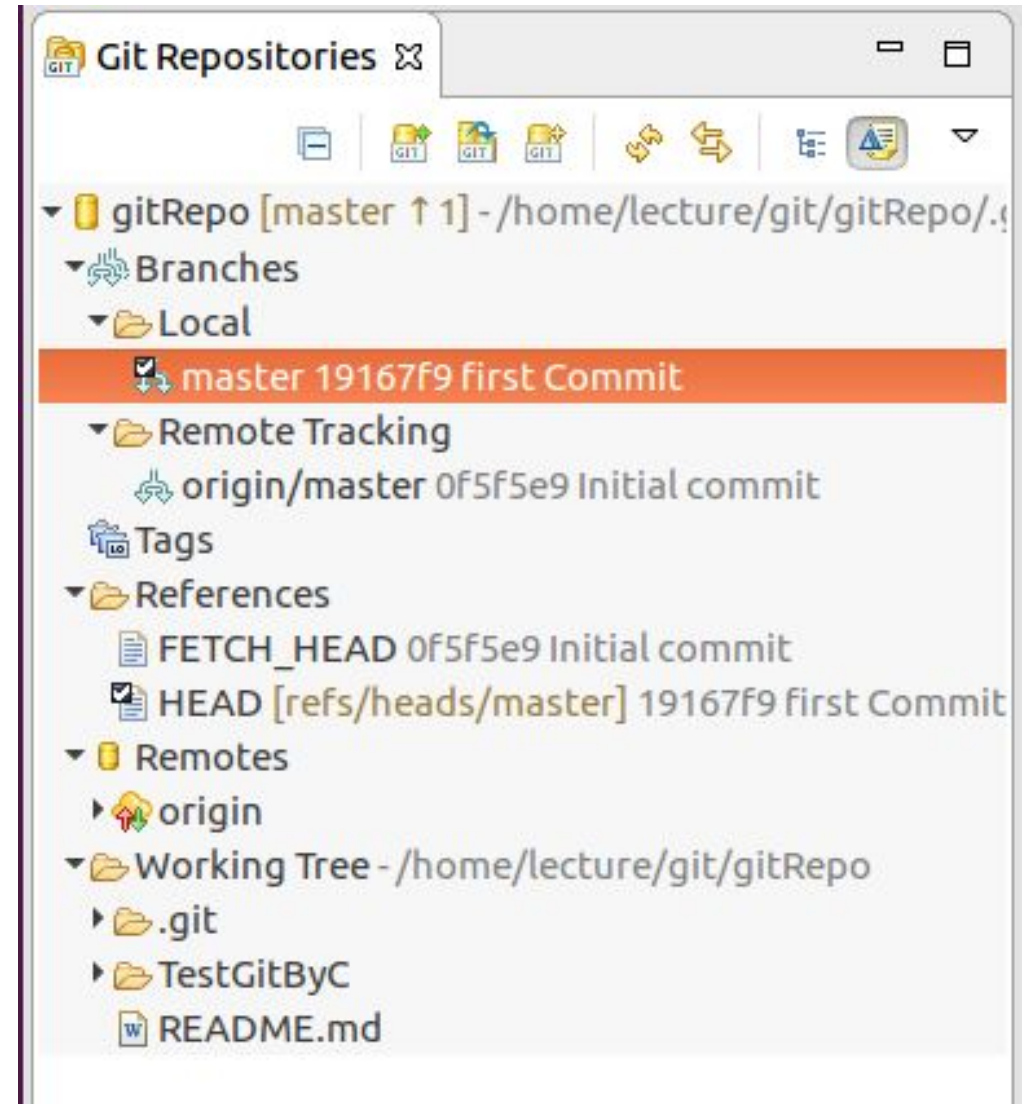


Eclipse - Commit & push(1)

- ❖ <https://github.com/> → Login
- ❖ Create New Repository
- ❖ Commit To Local Git

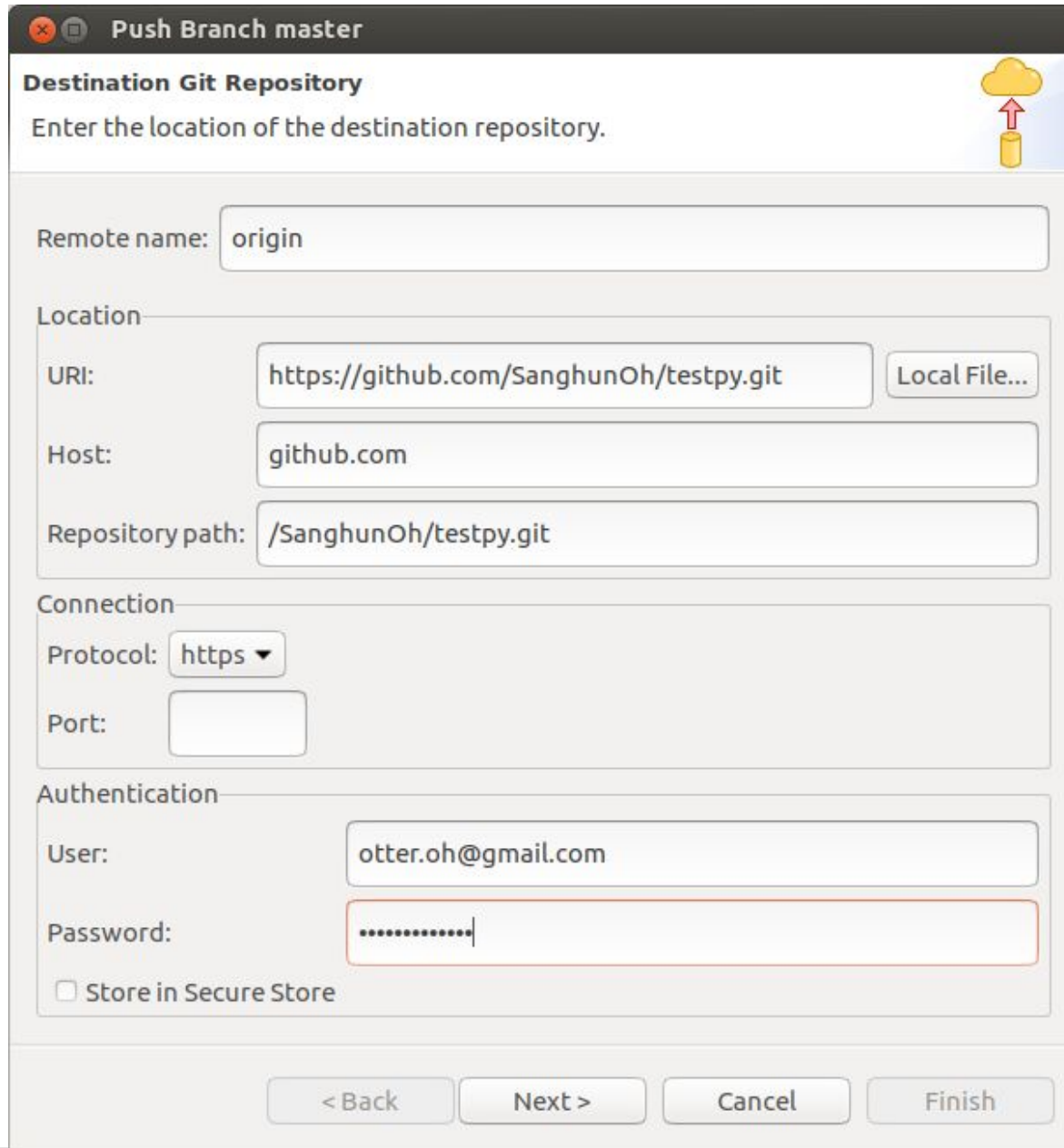


- ❖ Push To Remote Github



Eclipse - Commit & push(2)

❖ Enroll Remote Git URI



Push Branch master

Destination Git Repository
Enter the location of the destination repository.

Remote name:

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

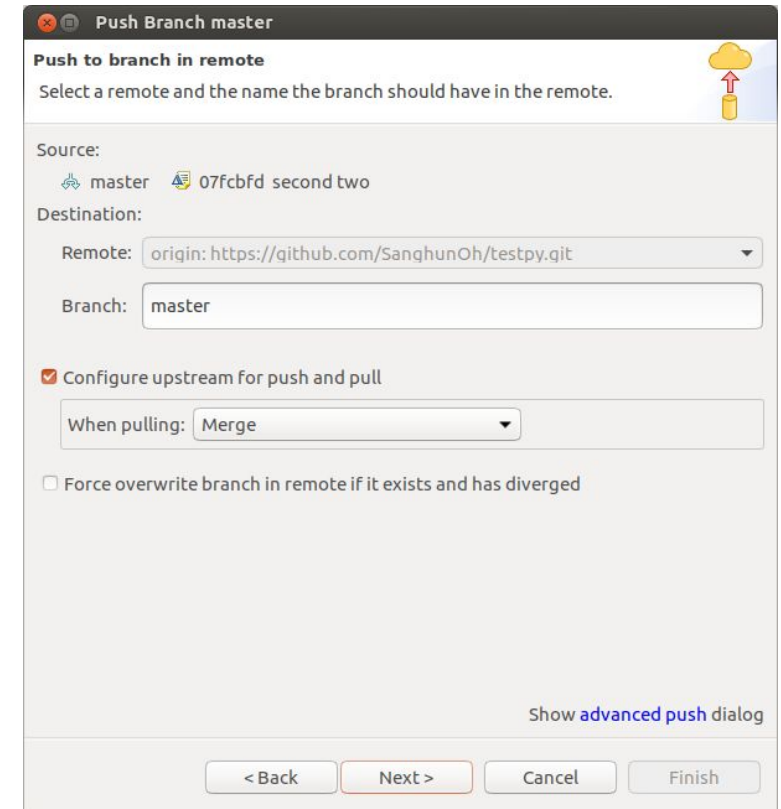
Authentication

User:

Password:

☐ Store in Secure Store

< Back Next > Cancel Finish



Push Branch master

Push to branch in remote
Select a remote and the name the branch should have in the remote.

Source:
master 07fcbfd second two

Destination:

Remote:

Branch:

☒ Configure upstream for push and pull

When pulling:

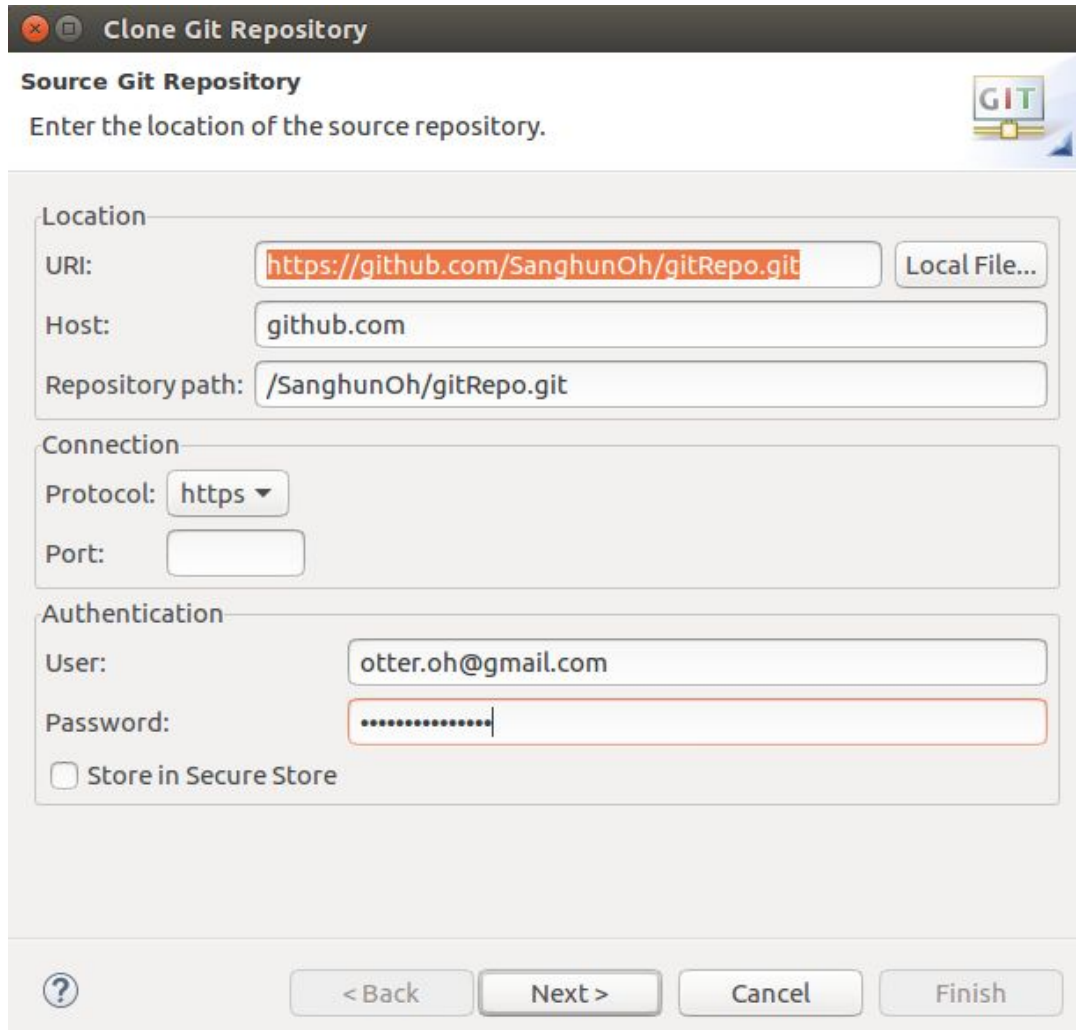
☐ Force overwrite branch in remote if it exists and has diverged

Show [advanced push](#) dialog

< Back Next > Cancel Finish

Eclipse - Clone Project(1)

- ❖ Window > Perspective > Open Perspective > Other
- ❖ Open Perspective -> Click 'Git'



Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:

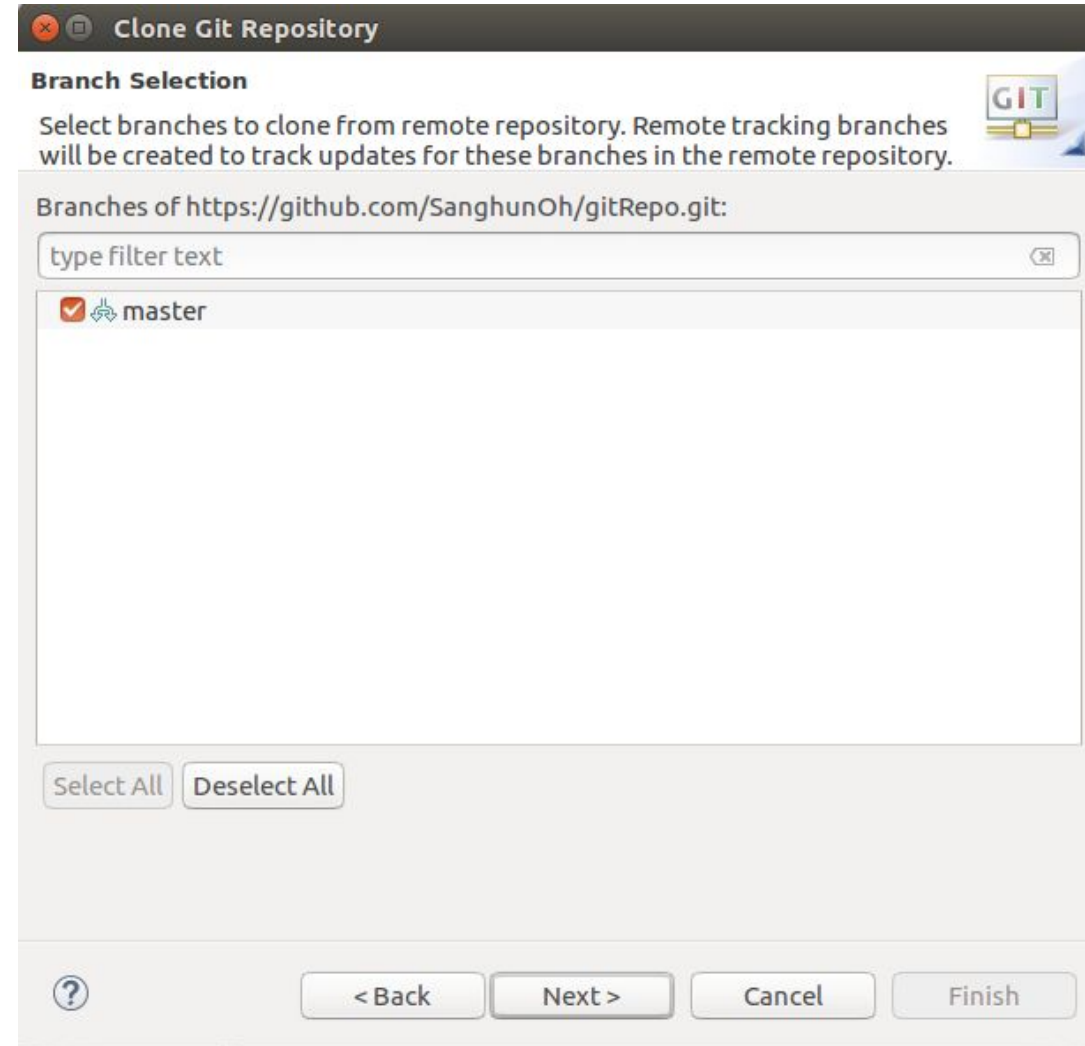
Port:

Authentication

User:

Password:

☐ Store in Secure Store




Clone Git Repository

Branch Selection

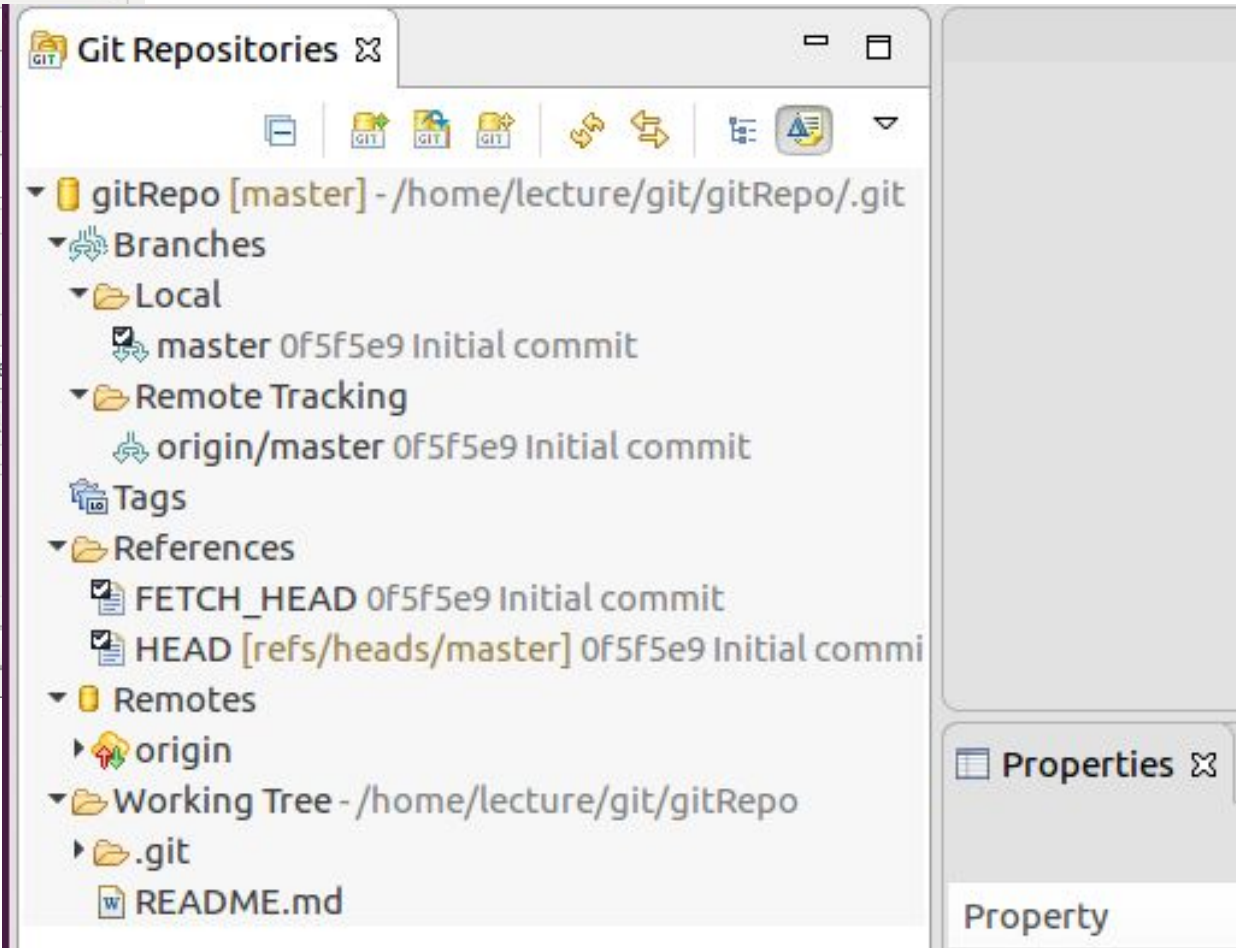
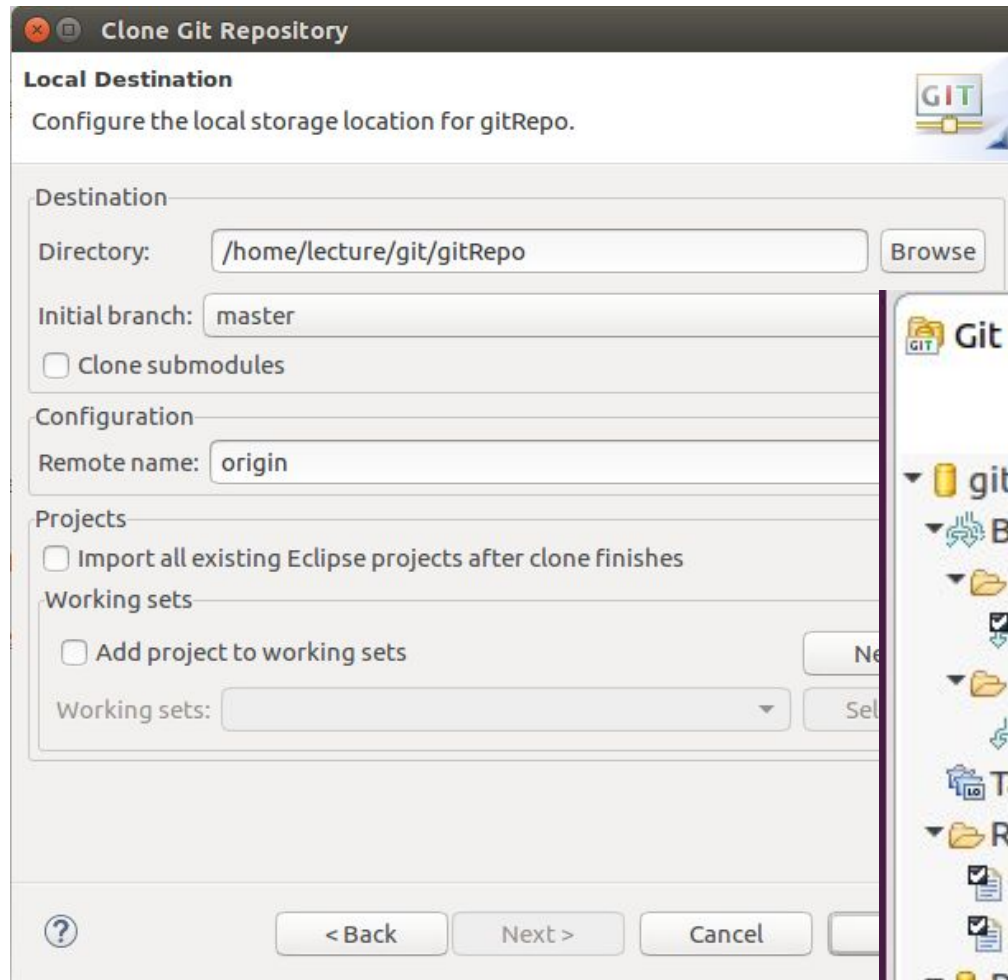
Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository.

Branches of https://github.com/SanghunOh/gitRepo.git:

☒  master

Eclipse - Clone Project(2)

❖ Check 'import all existing ...'





수고하셨습니다.