

GitHub @ Kyung Hee University

KhuHub 가이드라인

**Department of Computer Engineering,
Kyung Hee University.**

KHU-HUB

GitHub @ Kyung Hee University



경희대학교 컴퓨터공학과 원격 Git 서버입니다.

Existing user? Sign in

학번(Username) 또는 email

Password

Remember me

[Forgot your password?](#)

Sign in

로그인

New user? Create an account

이름

학번(Username)

email

Password - minimum length 8 characters

Sign up

회원가입

Didn't receive a confirmation email? [Request a new one.](#)

프로젝트 탐색

[Explore](#) [Professor Guideline](#) [Student Guideline](#) [Home page](#)

가이드라인

컴퓨터공학과 홈페이지

Sign Up Convention

KHU-HUB

GitHub @ Kyung Hee University



경희대학교 컴퓨터공학과 원격 Git 서버입니다.

Existing user? Sign in

학번(Username) 또는 email

Password

Remember me

[Forgot your password?](#)

Sign in

New user? Create an account

이름

학번(Username)

email

Password - minimum length 8 characters

Sign up

Didn't receive a confirmation email? [Request a new one.](#)

- ❖ [학생]
- ❖ 이름 : 홍길동
- ❖ 학번(Username) : 2017000000
- ❖ email : khu@khu.ac.kr
- ❖ Password : 8자 이상의 글자

Sign In

- ❖ 이메일 또는 학번(Username) 중에서 자유롭게 선택하여 로그인 가능
- ❖ 회원가입 시 입력한 이름은 단순 참고용이며, 학번(Username)이 본인을 구별할 수 있는 식별자입니다.
- ❖ 학번(Username)은 본인의 URL이 되므로 도메인 네이밍 규칙을 따릅니다.

Existing user? Sign in

2017000000

.....

Remember me [Forgot your password?](#)

Existing user? Sign in

khu@khu.ac.kr

.....

Remember me [Forgot your password?](#)

참고 사항

❖ 도메인 네임 (인터넷 주소)

❖ 규칙

- 영문자(A~Z), 숫자(0~9)또는 하이픈(-)의 조합으로만 가능하며 영문자의 대, 소문자 구별이 없다.
- 영어나 숫자로 시작해야 하며 하이픈(-)으로 끝내서는 안된다.
- 도메인 네임의 길이는 최소 2자에서 최대 63자까지 가능하다.(개인 도메인은 3자부터)
- 콤마(,)언더바(_)등의 기호와 특수문자(& %)는 사용할 수 없다.

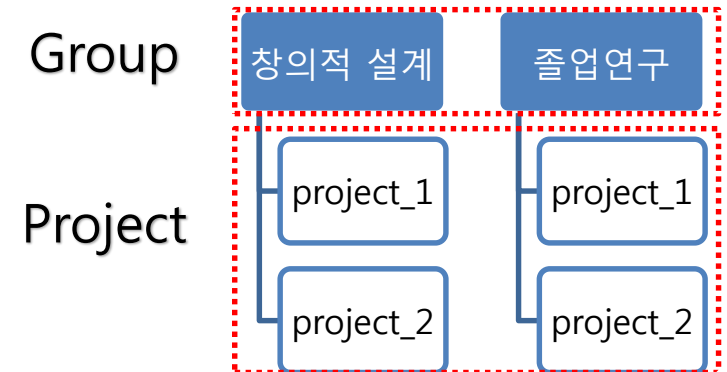
After Sign In

❖ Group

- 개발자들을 멤버로 추가하여 여러 프로젝트들을 모아놓고 함께 협업할 수 있는 집단
- KhuHub에서 그룹이란 것은 한 학기 수업으로 사용됩니다.
- 각 멤버는 그룹에 속한 여러 프로젝트에 대해서 접근 권한이 각기 다릅니다. 그룹 생성자는 모든 프로젝트에 접근하여 읽기/쓰기 모두 가능합니다.
- KhuHub에서는 group은 하나의 수업이 될 수 있으며, 그룹 생성자는 교수님 또는 수업조교가 됩니다.
- 그룹 이름은 해당 그룹의 URL이 되기 때문에 띄어쓰기 없이 영문과 특정 기호만 가능합니다. (도메인 네임 규칙)
 ex> 2016-1-operating-system

❖ Project

- 한 명 이상의 개발자가 참여하여 진행할 수 있는 개발 프로젝트의 최소 단위
- 프로젝트 생성자는 팀원을 초대하여 협업할 수 있습니다.
- 프로젝트 이름은 해당 프로젝트의 URL이 되기 때문에 띄어쓰기 없이 영문과 특정 기호만 가능합니다. (도메인 네임 규칙)



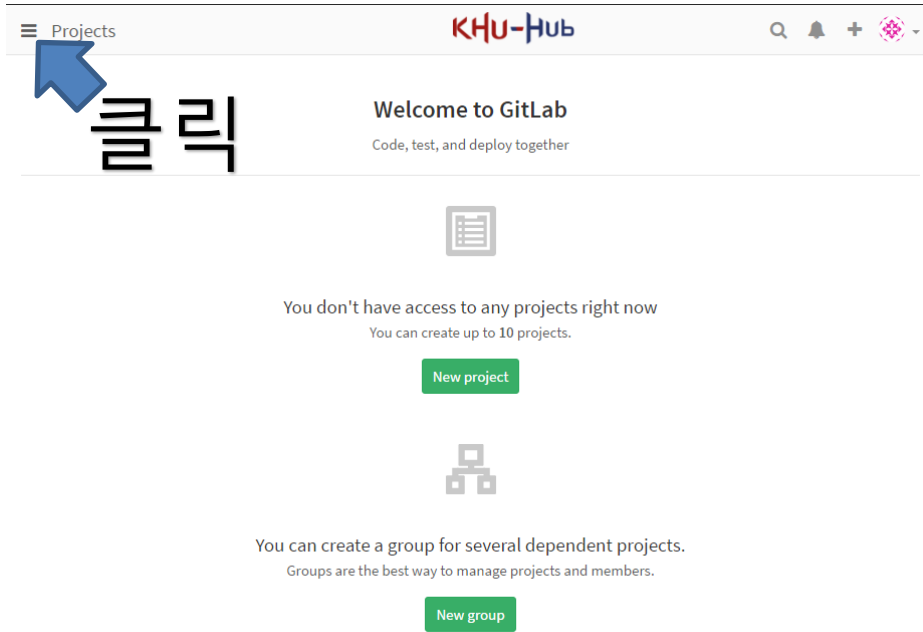
group 생성-1

- ❖ 각 group의 이름은 관리 편의, 검색 등을 위해 명명 규칙을 준수해주세요.
- ❖ 각 group은 동아리 / 프로젝트 팀 등을 기준으로 여러 유저들을 묶어주는 단위입니다.

❖ 명명 규칙

- 도메인 네임 규칙을 기반
- 띄어쓰기는 -(하이픈)으로 구분
- 년도-학기-수업의 영문명
ex> 2016-1-operating-system 또는 2016-2-creative-design
- 동시에 여러 수업이 개설될 경우 -(하이픈)숫자 로 구분
ex> 2016-1-operating-system-1, 2016-1-operating-system-2

group 생성-2



Projects

KHU-HUB

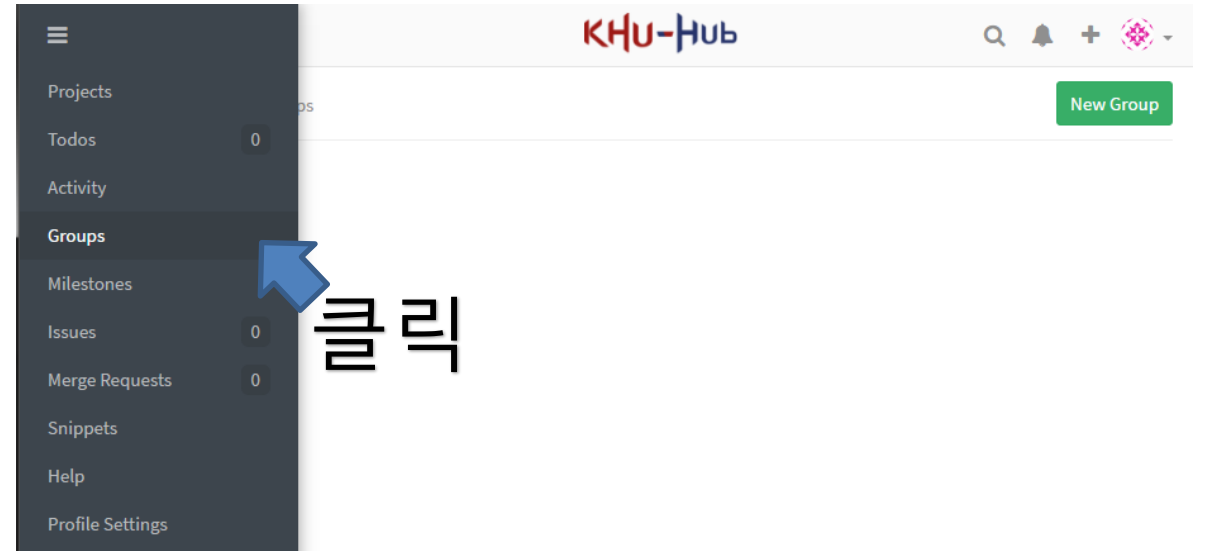
Welcome to GitLab
Code, test, and deploy together

You don't have access to any projects right now
You can create up to 10 projects.

New project

You can create a group for several dependent projects.
Groups are the best way to manage projects and members.

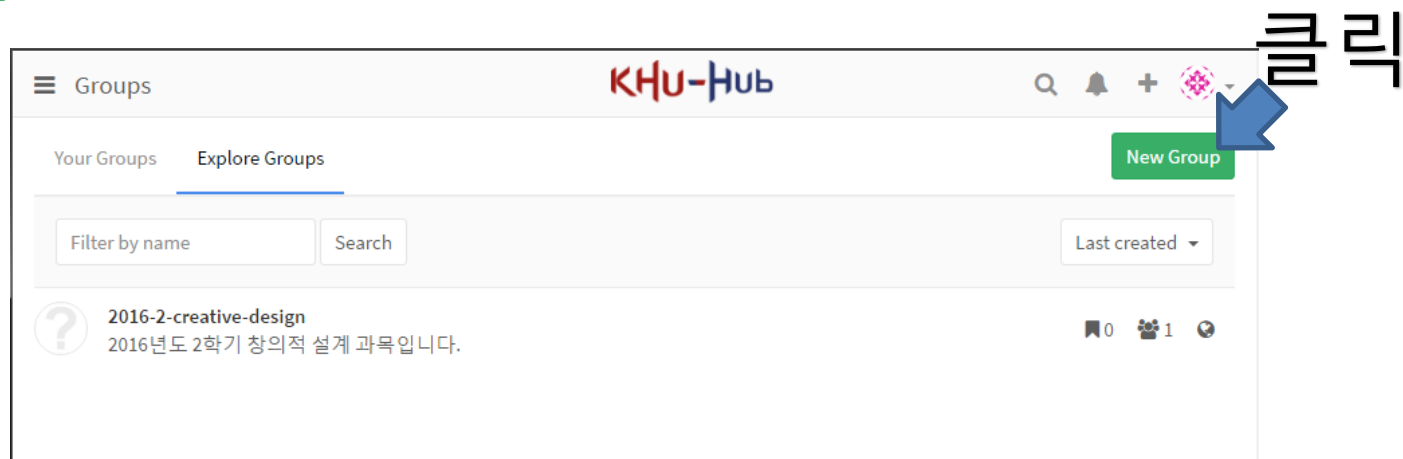
New group



KHU-HUB

- Projects
- Todos 0
- Activity
- Groups**
- Milestones
- Issues 0
- Merge Requests 0
- Snippets
- Help
- Profile Settings

New Group



Groups

KHU-HUB

Your Groups Explore Groups

New Group

Filter by name Search Last created ▾

? 2016-2-creative-design
2016년도 2학기 창의적 설계 과목입니다. 0 1

group 생성-3

Groups
KHU-Hub
🔍 🔔 + 🌐

New Group

Group path:

Description:

Group avatar: File name... **그룹을 상징하는 프로필 사진(옵션)**
The maximum file size allowed is 200KB.

Visibility Level ?

- Private** 해당 그룹을 볼 수 있는 범위
The group and its projects can only be viewed by members.
- Internal**
The group and any internal projects can be viewed by any logged in user.
- Public**
The group and any public projects can be viewed without any authentication.

- A group is a collection of several projects
- Members of a group may only view projects they have permission to access
- Group project URLs are prefixed with the group namespace
- Existing projects may be moved into a group

❖ Group path

- 그룹 명명 규칙에 따라 정해진 그룹명을 입력

❖ Description

- 그룹에 대한 간략한 설명. 문자 제약 없음

❖ Group avatar

- 그룹을 상징하는 프로필 사진(옵션)

❖ Visibility Level

- 해당 그룹을 볼 수 있는 범위
- KhuHub에서는 public을 권장
- Public으로 해야 학생들이 접근 및 검색이 가능
- 개인적인 개발을 위한 경우, Private으로 설정 가능

그룹 생성 예시

Groups **KHU-Hub** Search [] [] [] []

New Group

Group path:

Description:

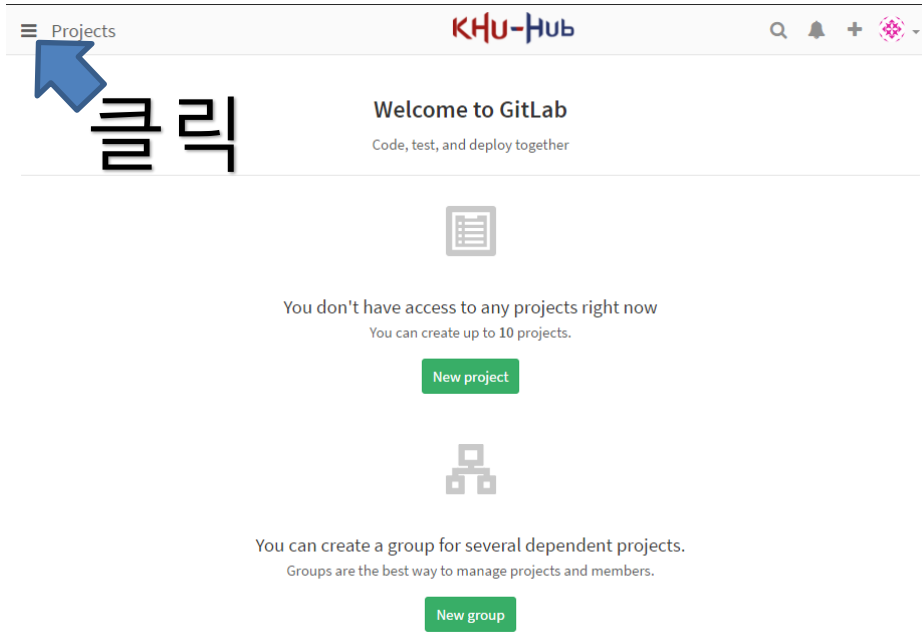
Group avatar:
The maximum file size allowed is 200KB.

Visibility Level ?

- Private
The group and its projects can only be viewed by members.
- Internal
The group and any internal projects can be viewed by any logged in user.
- Public
The group and any public projects can be viewed without any authentication.

- A group is a collection of several projects
- Members of a group may only view projects they have permission to access
- Group project URLs are prefixed with the group namespace
- Existing projects may be moved into a group

그룹 검색



Projects **KHU-HUB** 🔍 🔔 + 🌐

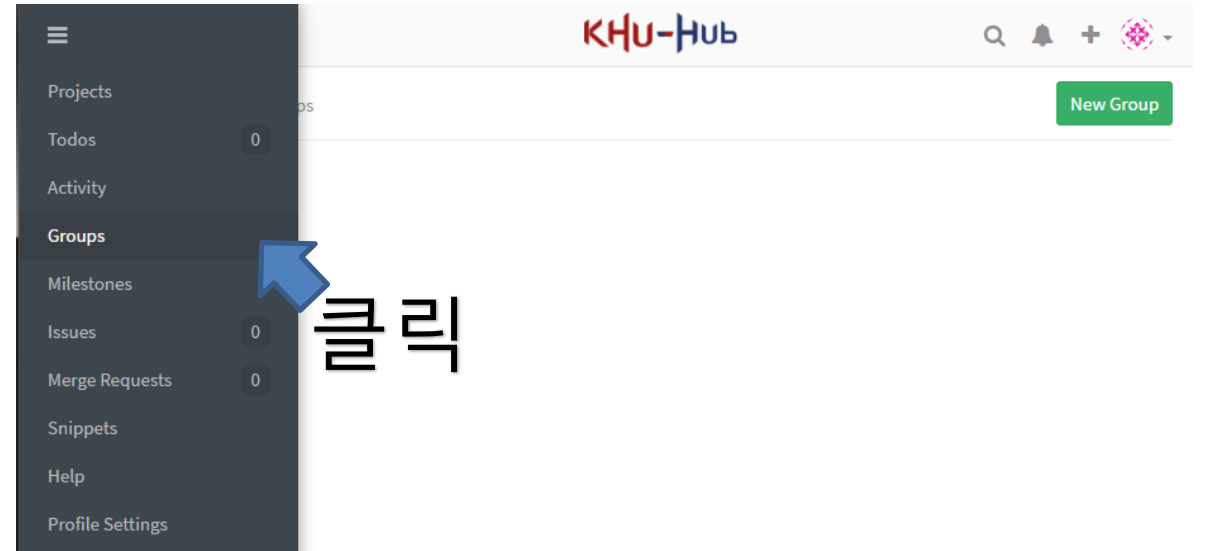
Welcome to GitLab
Code, test, and deploy together

You don't have access to any projects right now
You can create up to 10 projects.

[New project](#)

You can create a group for several dependent projects.
Groups are the best way to manage projects and members.

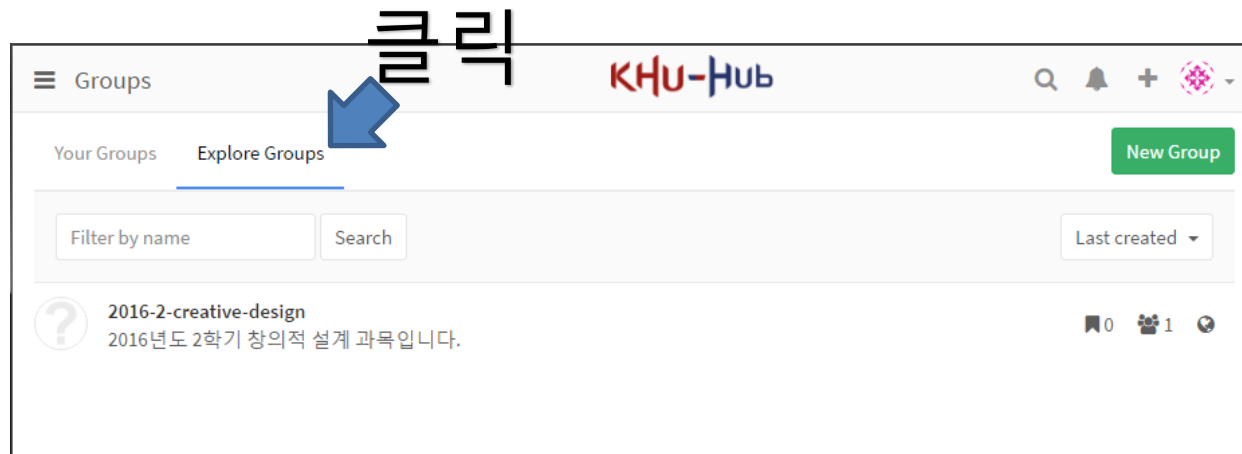
[New group](#)



KHU-HUB 🔍 🔔 + 🌐

- Projects
- Todos 0
- Activity
- Groups**
- Milestones
- Issues 0
- Merge Requests 0
- Snippets
- Help
- Profile Settings


[New Group](#)



Groups **KHU-HUB** 🔍 🔔 + 🌐

Your Groups Explore Groups [New Group](#)

Filter by name Search Last created ▾

 **2016-2-creative-design**
2016년도 2학기 창의적 설계 과목입니다. 0 1

유저 초대 및 운영-1

❖ 그룹 관리자의 초대 or 유저의 액세스 요청

Members of group have access to all group projects.

People

Search for users by name, username, or email, or invite new ones using their email address.

Group Access


Read more about role permissions [here](#)

Access expiration date

On this date, the user(s) will automatically lose access to this group and all of its projects.

Add users to group

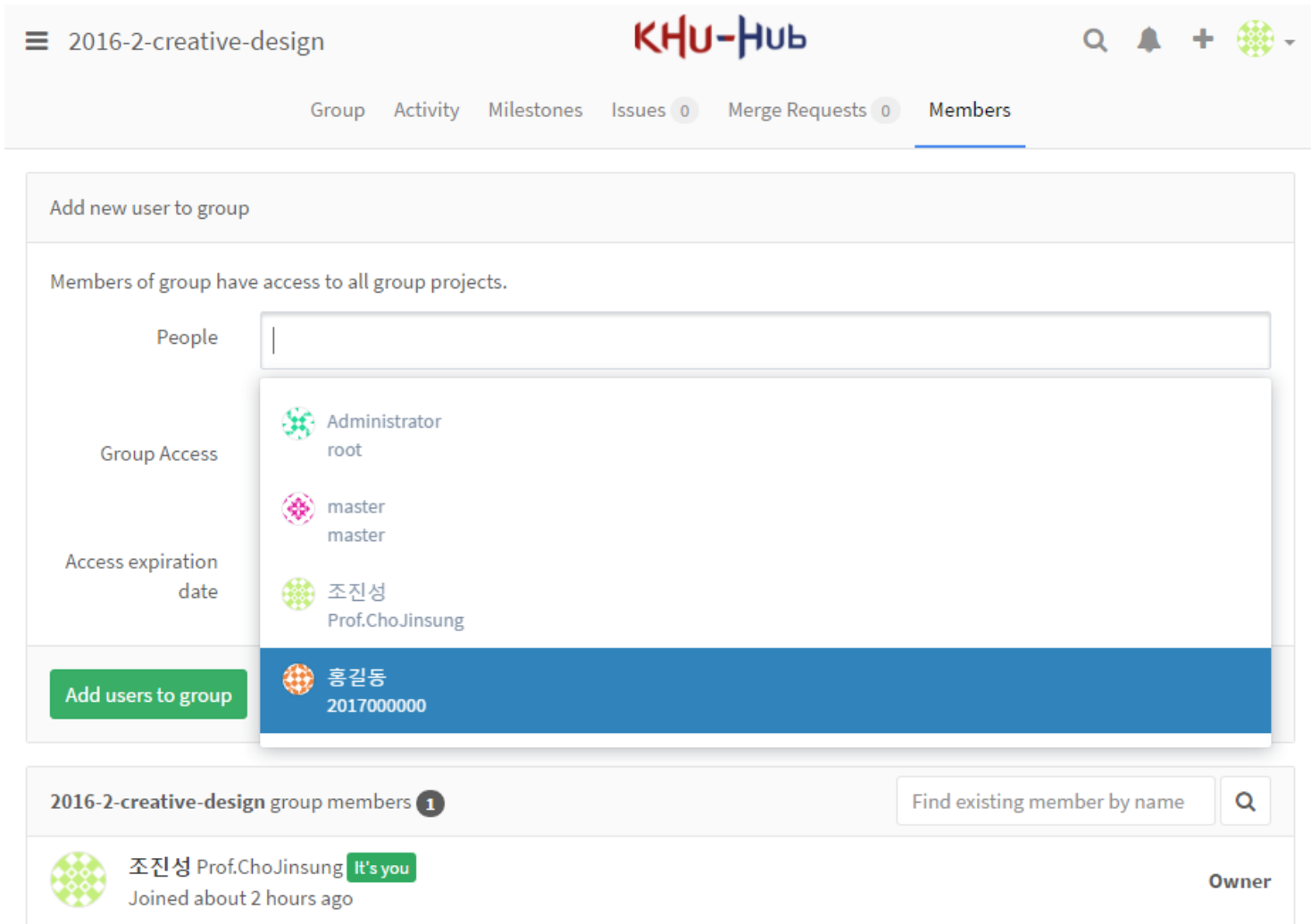
2016-2-creative-design group members **1**

 조진성 Prof.ChoJinsung It's you Joined about 2 hours ago	Owner
-----------------------------------------------------------------------------------------------------------------------------------------------------	--------------

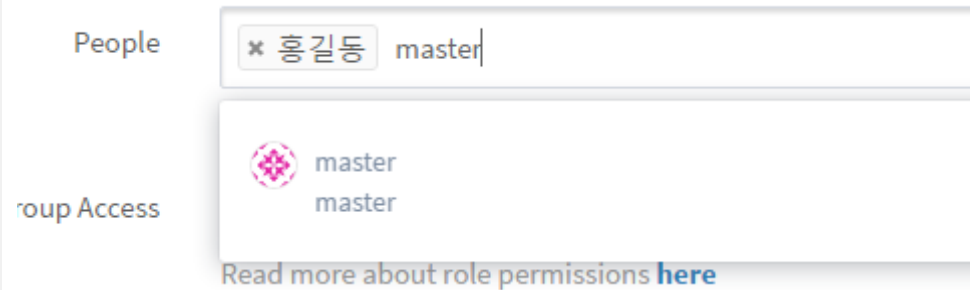
관리자일 경우에만 초대가능

유저 초대 및 운영-2 (그룹 관리자의 초대)

❖ 리스트에서 유저를 고르거나, 이름, 학번, 이메일 등을 입력하여 초대 가능



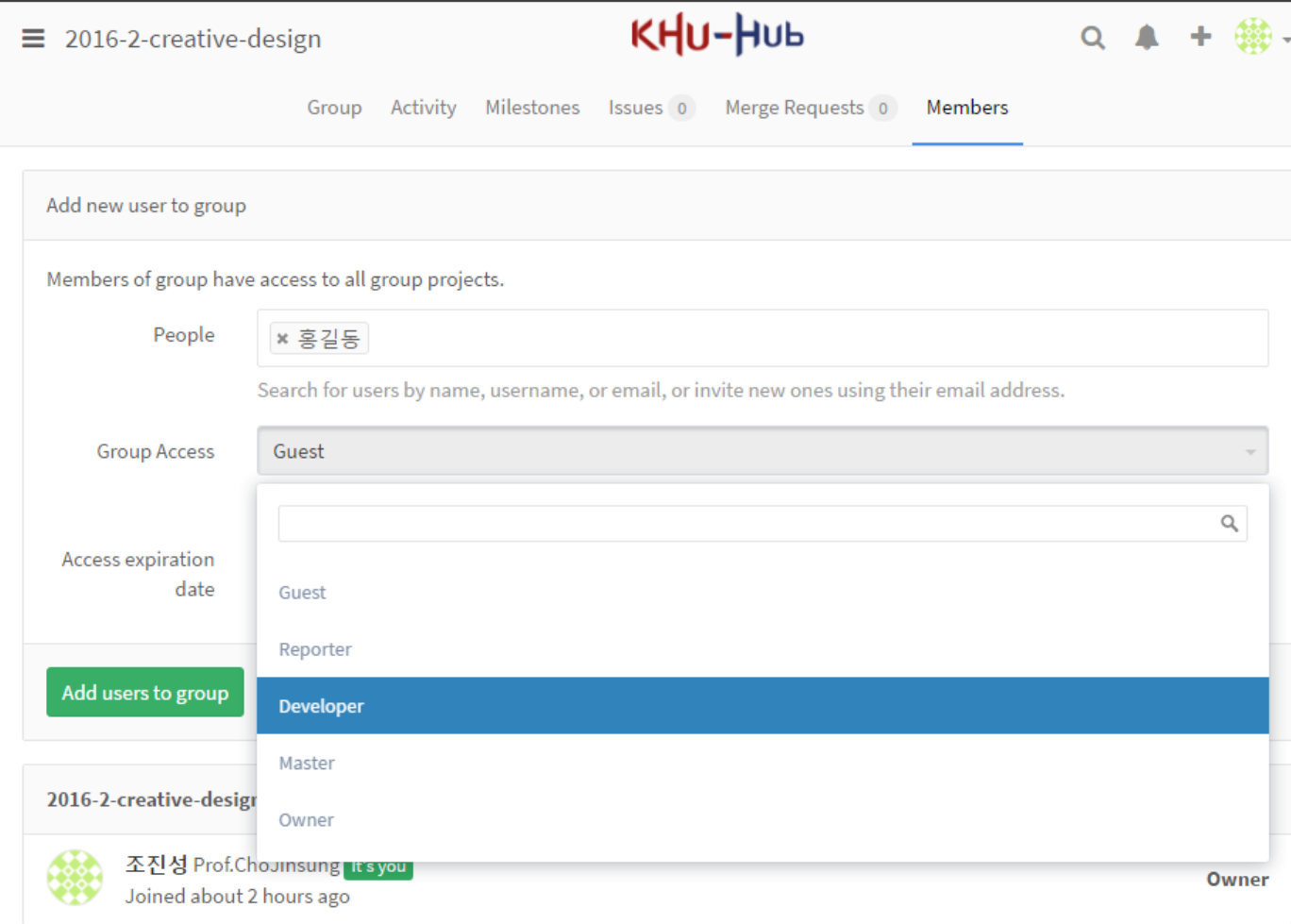
The screenshot shows the '2016-2-creative-design' group page in KJU-Hub. The 'Members' tab is active. A search box for adding new users is open, showing a list of suggestions including 'Administrator root', 'master master', '조진성 Prof.ChoJinsung', and '홍길동 2017000000'. The '홍길동' user is highlighted. Below the search box, a list of existing group members is shown, including '조진성 Prof.ChoJinsung' who is the 'Owner' and joined about 2 hours ago.



This diagram illustrates the user selection and group access options. On the left, the 'People' section shows a search input with '홍길동 master' entered. Below it, the 'Group Access' section shows two options: 'master master' and 'master master'. A link 'Read more about role permissions here' is provided below the access options.

그룹 초대 및 운영-3 (그룹 관리자의 초대)

❖ 권한 추가



The screenshot shows the '2016-2-creative-design' group page on KIU-Hub. The 'Members' tab is active. A search bar contains '홍길동'. The 'Group Access' dropdown menu is open, showing options: Guest, Reporter, **Developer** (highlighted), Master, and Owner. A green button 'Add users to group' is visible. At the bottom, a user profile for '조진성 Prof. Cho Jinsung' is shown as the 'Owner'.



해당 그룹에 프로젝트
생성은 Master 권한부터
가능

유저 초대 및 운영-4 (그룹 관리자의 초대)

❖ 해당 그룹에 초대 완료

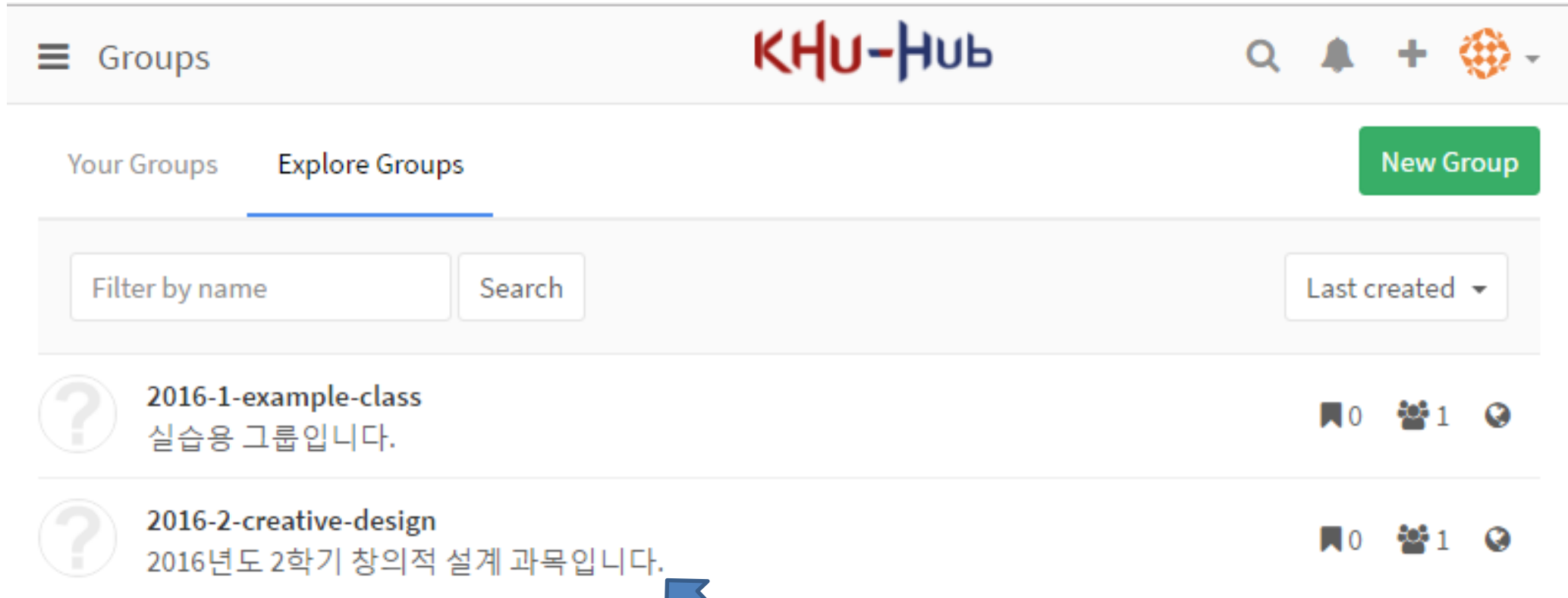
2016-2-creative-design group members **2**

Find existing member by name

	홍길동 2017000000 Joined less than a minute ago	Developer	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>
	조진성 Prof.ChoJinsung It's you Joined about 2 hours ago	Owner		

유저 초대 및 운영-5 (유저의 액세스 요청)

❖ 참여할 그룹 검색

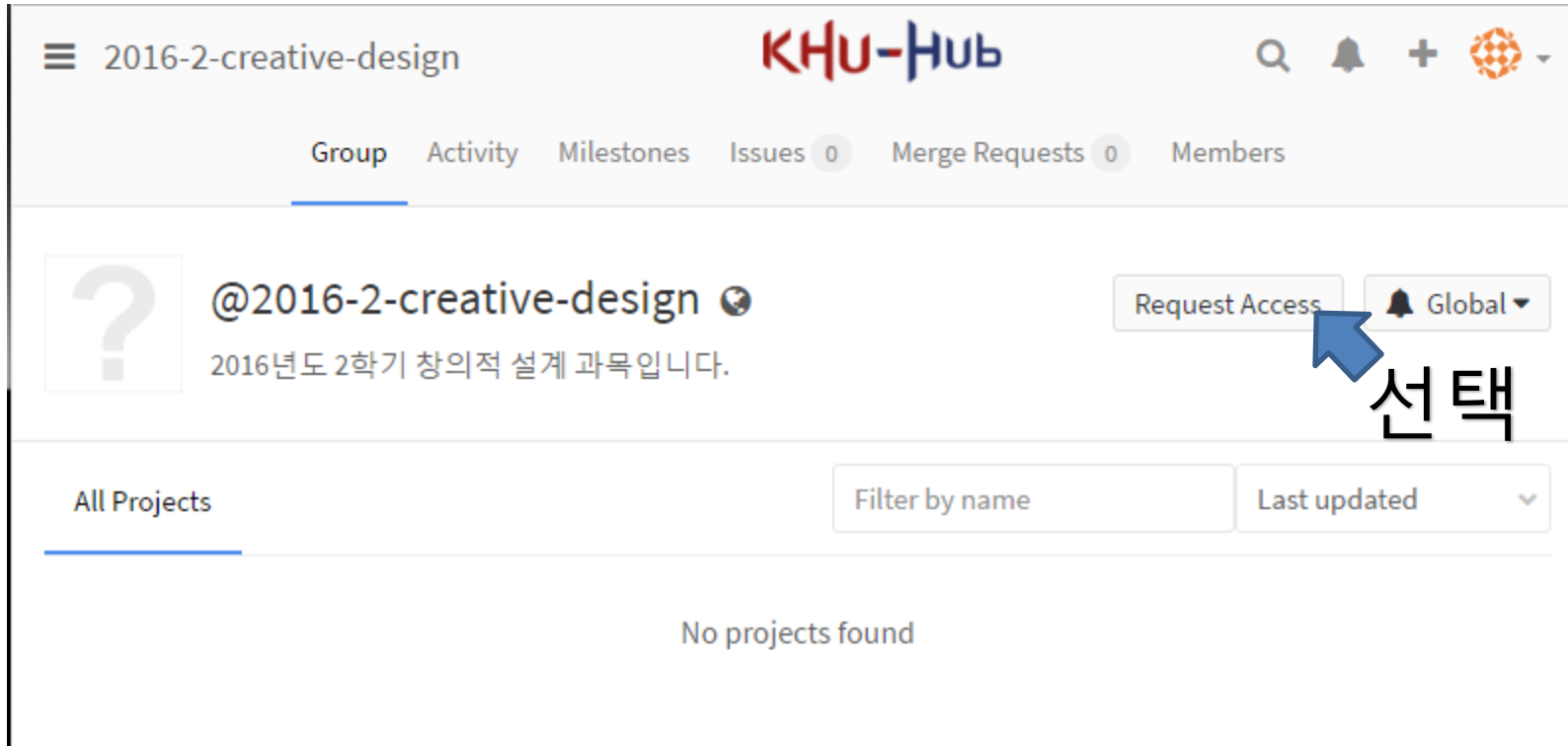


The screenshot shows the KHU-HUB interface for group management. At the top, there is a navigation bar with a menu icon, the text "Groups", the "KHU-HUB" logo, and icons for search, notifications, a plus sign, and a profile icon. Below the navigation bar, there are two tabs: "Your Groups" and "Explore Groups", with "Explore Groups" being the active tab. A green "New Group" button is located in the top right corner. Below the tabs, there is a search section with a "Filter by name" input field, a "Search" button, and a "Last created" dropdown menu. The main content area displays a list of groups. The first group is "2016-1-example-class" with the description "실습용 그룹입니다." and icons for 0 bookmarks, 1 member, and a lock icon. The second group is "2016-2-creative-design" with the description "2016년도 2학기 창의적 설계 과목입니다." and the same icons. A blue arrow points to the second group's description, and the word "선택" (Select) is written next to it.

선택

유저 초대 및 운영-6 (유저의 액세스 요청)

❖ Request Access



2016-2-creative-design KHU-Hub

Group Activity Milestones Issues 0 Merge Requests 0 Members

? @2016-2-creative-design 2016년도 2학기 창의적 설계 과목입니다.

Request Access Global

All Projects Filter by name Last updated

No projects found

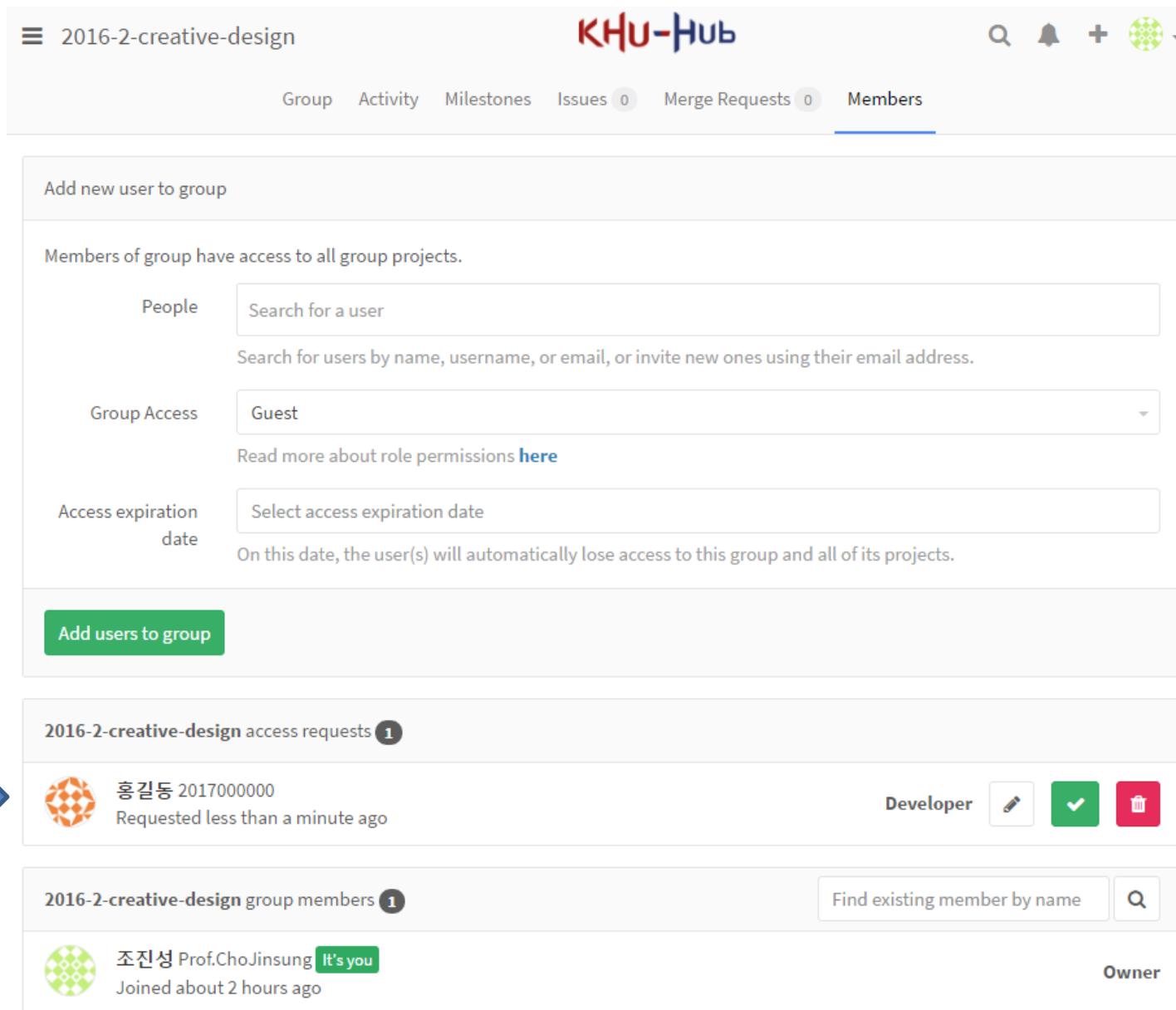
선택

유저 초대 및 운영-7 (유저의 액세스 요청)

❖ 그룹 관리자의 허가

❖ Accept/Delete

❖ 권한 수정 가능

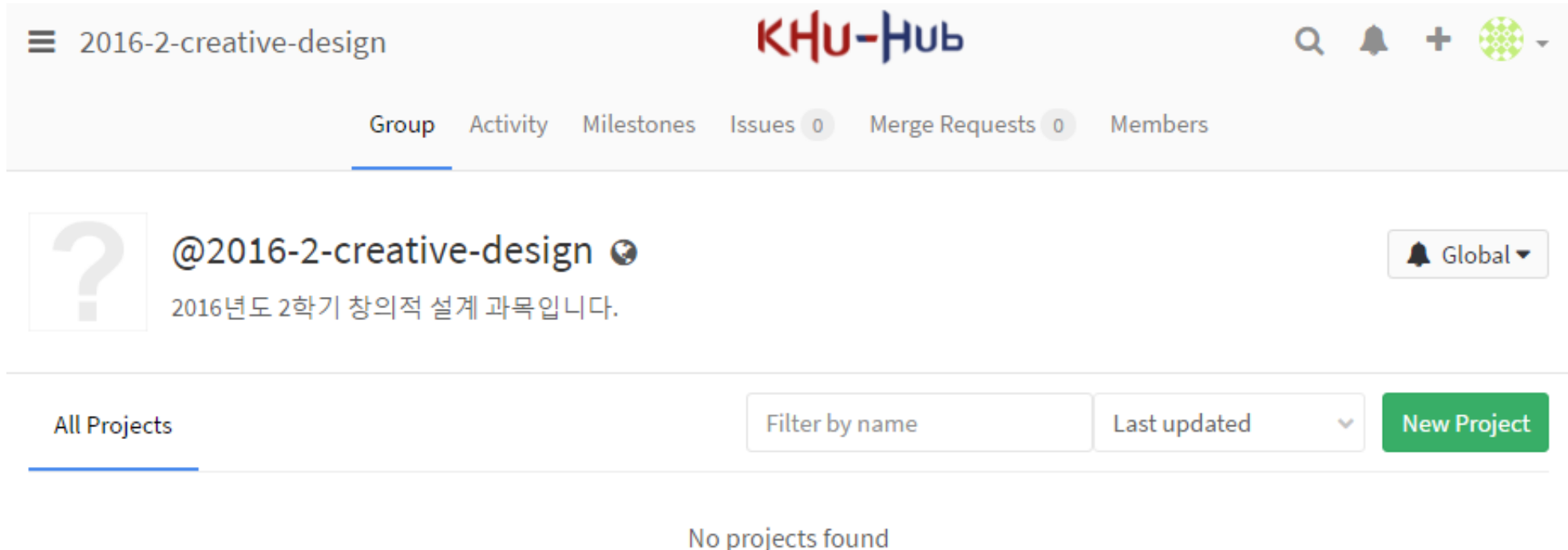


The screenshot shows the 'Members' tab of a group named '2016-2-creative-design' on the KHU-HUB platform. The interface includes a search bar for adding new users, a dropdown for 'Group Access' (currently set to 'Guest'), and a date picker for 'Access expiration date'. Below these options is a green 'Add users to group' button. A section titled '2016-2-creative-design access requests' shows one request from user '홍길동 2017000000' (Developer) who requested access less than a minute ago. The request is currently pending, with icons for edit, accept, and delete. A second section titled '2016-2-creative-design group members' shows one member, '조진성 Prof.ChoJinsung' (Owner), who joined about 2 hours ago and is marked as 'It's you'.

해당 유저의 요청 ➡

프로젝트 생성-1

- ❖ 프로젝트 생성은 해당 그룹의 Master/Owner 권한 이상만 가능
- ❖ 팀장이 프로젝트를 생성해준 뒤, 다른 유저에게 이관하는 방식
- ❖ 유저들에게 권한을 주고 자유롭게 프로젝트 생성 및 삭제하도록 하는 방식



The screenshot shows the KHU-Hub interface for a group named '2016-2-creative-design'. The top navigation bar includes the group name, the KHU-Hub logo, and search, notification, and user icons. Below the navigation bar are tabs for 'Group', 'Activity', 'Milestones', 'Issues' (0), 'Merge Requests' (0), and 'Members'. The group profile section shows a placeholder for a profile picture, the group name '@2016-2-creative-design', and a description: '2016년도 2학기 창의적 설계 과목입니다.' A 'Global' notification setting is visible. At the bottom, there is a filter section with 'All Projects' selected, a 'Filter by name' input field, a 'Last updated' dropdown menu, and a green 'New Project' button. Below the filter section, the text 'No projects found' is displayed.

프로젝트 생성-2

- ❖ 프로젝트 이름도 도메인 네임 규칙에 따라 생성
- ❖ Visibility Level은 반드시 public

New project

Create or Import your project from popular Git services

Project path

http://163.180.118.147/

2016-2-creative-design


Project name


my-awesome-project


Want to house several dependent projects under the same namespace? [Create a group](#)


Import project from

 GitHub

 Bitbucket

 GitLab.com

 Google Code




 Fogbugz

 git Repo by URL

Project description (optional)

Description format

Visibility Level (?)

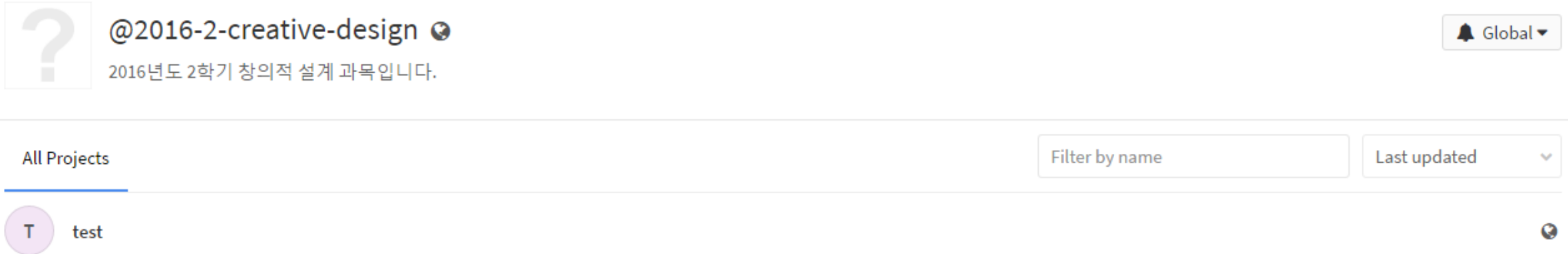
-  Private
Project access must be granted explicitly to each user.
-  Internal
The project can be cloned by any logged in user.
-  Public
The project can be cloned without any authentication.

Create project

Cancel

프로젝트 생성-3

- ❖ 프로젝트 생성 완료
- ❖ 이후 프로젝트 트리 생성, 소스코드 업로드, 버전 컨트롤 및 협업 기능은 Git을 이용



The screenshot shows a user profile for '@2016-2-creative-design' with a bio '2016년도 2학기 창의적 설계 과목입니다.' and a 'Global' notification dropdown. Below the profile is a navigation bar with 'All Projects' selected, a search box 'Filter by name', and a dropdown menu 'Last updated'. A single project entry is visible with a purple circular icon containing the letter 'T' and the name 'test'.

Git / GitHub 관련 참고자료

- ❖ <http://www.slideshare.net/kairosfox/git-github-seminar2-47909062>
- ❖ <http://www.slideshare.net/kairosfox/git-github-seminar2>

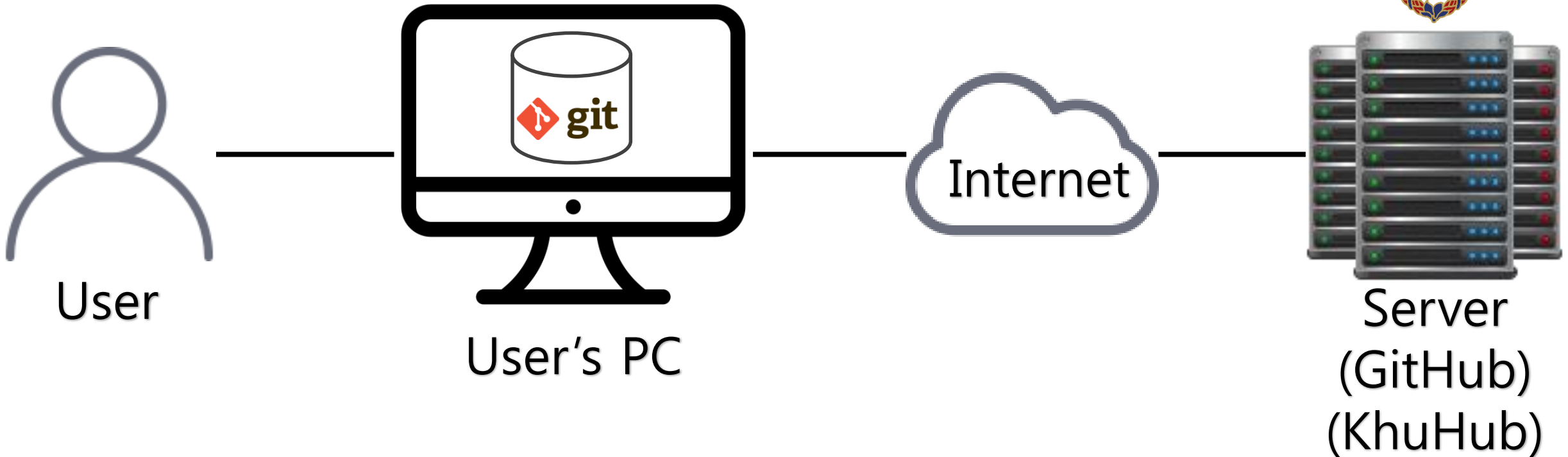
GIT이란?

Git (분산 버전 관리 시스템)

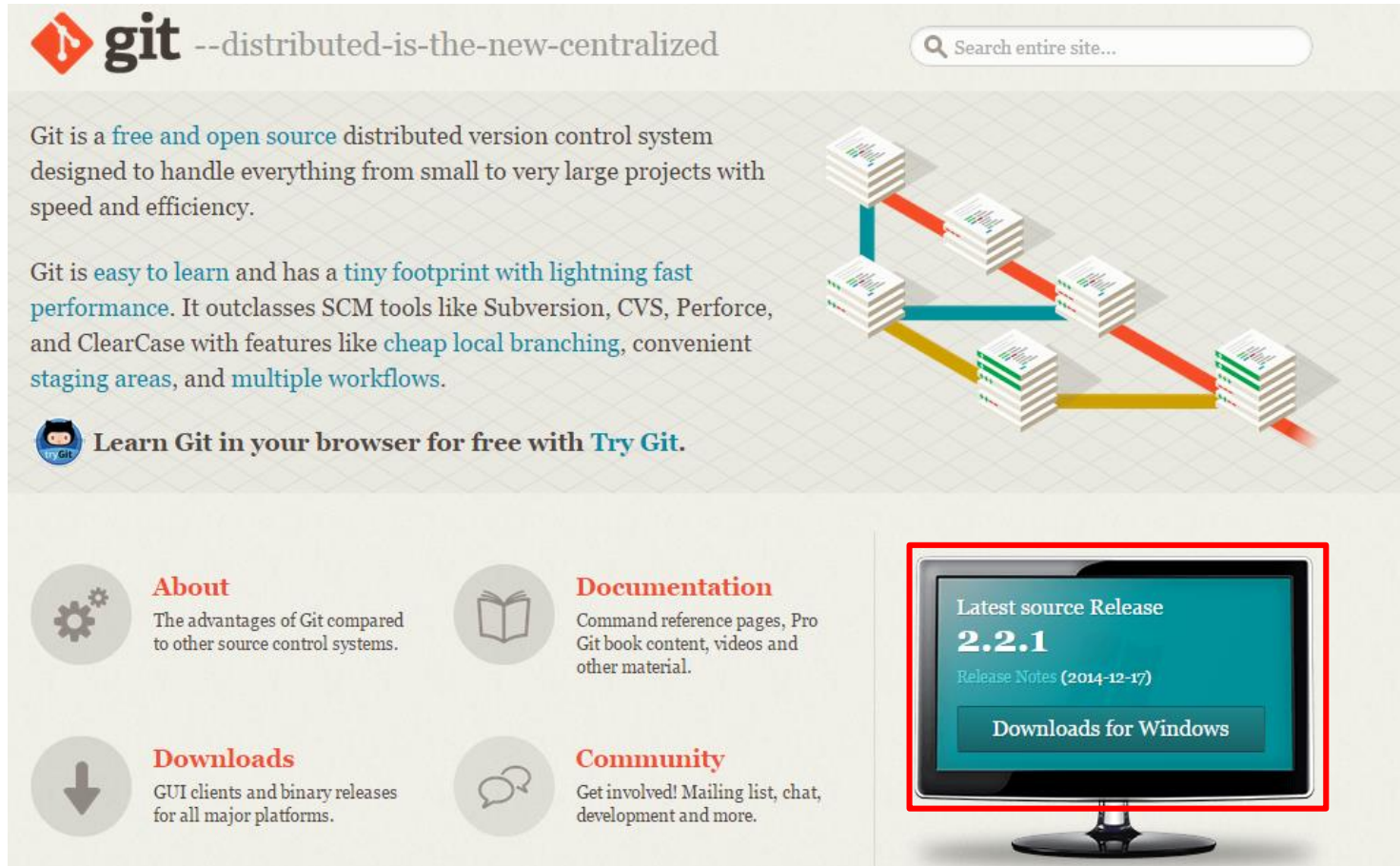
- ❖ 프로젝트(소스코드, 리소스 등)를 관리하기 위한 분산 버전 관리 시스템(DVCS)
- ❖ 네트워크에 접근하거나 중앙 서버에 의존하지 않는 완전한 형태의 저장소
- ❖ 리누즈 토발즈가 리눅스 커널 개발에 이용하려고 최초 개발 및 사용
- ❖ 깃의 작업 폴더는 모두, 전체 기록과 각 기록을 추적할 수 있는 정보를 포함
- ❖ 다운로드 : <https://git-scm.com/downloads>



Git과 GitHub/KhuHub 의 관계



GIT 설치 방법




The screenshot shows the Git website homepage. At the top left is the Git logo and the slogan "--distributed-is-the-new-centralized". To the right is a search bar. The main content area features a descriptive paragraph about Git, a diagram of a distributed version control system, and a "Try Git" button. Below this are four navigation cards: "About", "Documentation", "Downloads", and "Community". On the right side, a computer monitor displays the "Latest source Release 2.2.1" with a "Downloads for Windows" button highlighted by a red border.

git --distributed-is-the-new-centralized

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint** with **lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

 Learn Git in your browser for free with **Try Git**.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

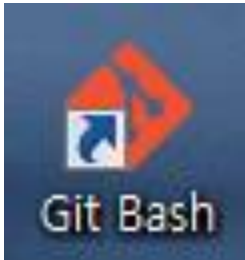
Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Mailing list, chat, development and more.

Latest source Release
2.2.1
[Release Notes \(2014-12-17\)](#)

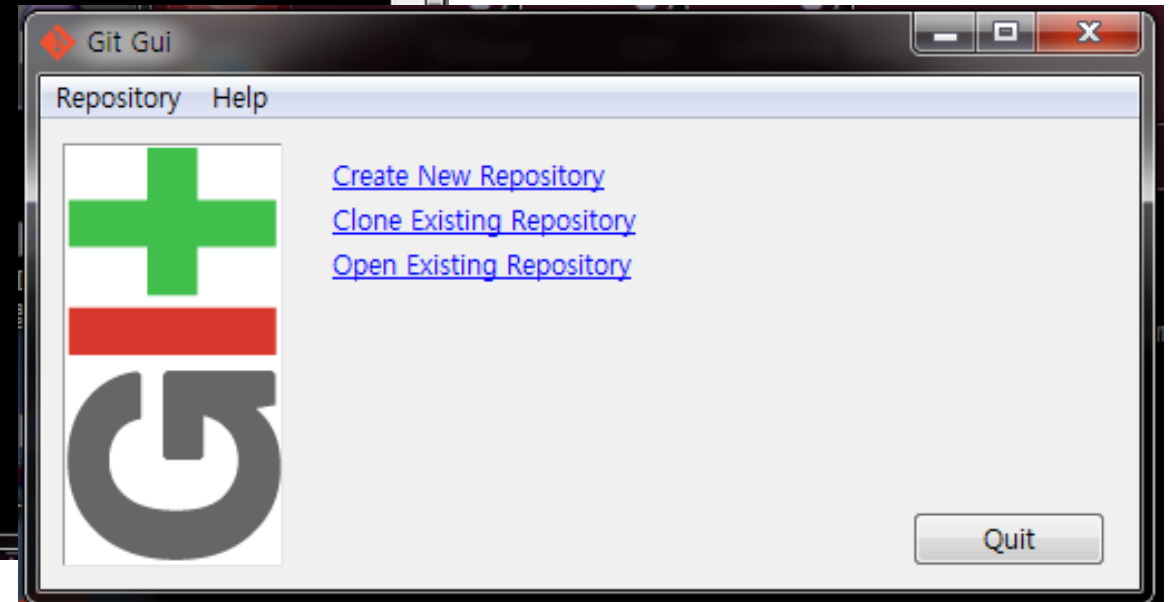
Downloads for Windows

Windows 터미널 프로그램

A screenshot of a Git Bash terminal window. The title bar reads "Git Bash". The terminal text is as follows:

```
Welcome to Git (version 1.8.1.2-preview20130201)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.
HanSangyun@HANSANGYUN-LAB ~
$
```



Linux 설치

1. `sudo apt-get install git-core git-doc`

```
hansangyun@hansangyun:~$ sudo apt-get install git-core git-doc
```

2. `git --version`

```
hansangyun@hansangyun:~$ git --version  
3. git version 1.9.1
```

SSH Key 등록

Profile Settings


KHU-HUB

Search

Profile Account Applications Access Tokens Emails Password Notifications **SSH Keys** Preferences Audit Log

Public Avatar

You can upload an avatar here or change it at gravatar.com



Upload new avatar

Browse file... No file chosen

The maximum file size allowed is 200KB.

Main settings

This information will appear on your profile.

Name

홍길동

Enter your name, so people you know can recognize you.

Email

khu@khu.ac.kr

We also use email for avatar detection if no avatar is uploaded.

SSH Key 발급

- ❖ 리눅스에서는 일반 터미널 / 윈도우에서는 앞서 설치한 git 터미널에서 실행
- ❖ `cat ~/.ssh/id_rsa.pub`
 - 해당 명령어를 통해 ssh 키 확인 후, 복사하여 KhuHub에 붙여넣기

```
khu-github@khu-github:~$ ssh-keygen -t rsa -C khu@khu.ac.kr
Generating public/private rsa key pair.
Enter file in which to save the key (/home/khu-github/.ssh/id_rsa):
Created directory '/home/khu-github/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/khu-github/.ssh/id_rsa.
Your public key has been saved in /home/khu-github/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:9YK9dLcg7CbG9pWrajSfujhrxF/rPvKwFFf0WxpQWgY khu@khu.ac.kr
The key's randomart image is:
+---[RSA 2048]---+
|                 Eo+                 |
|                 =                   |
|                o o o .               |
|               = = . =                |
|              . S B = +               |
|             o .oo.* + .              |
|            . o*+=oo .                |
|           o++B=. .                   |
|          .o+=0*o.                     |
+-----[SHA256]-----+
```

SSH key 발급

- ❖ Key가 저장된 경로에서 키값 복사

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC84A0jtxsfn9zD6W0/1JdGmbpviGh0mp3NuGzE/wz7  
eDoBR1wUd94kvXQayhv6SW6lVe0Co0kjsW7DiXJiWcxrWvDQJcqJuPptq2vLGAW8ES3S5tgSSLKQUJic  
fyuEGI9/L88IdQ8Yaqhqjfr4v5PhYyDffEKpr0cd0aSU7UU5K6GqCNT0V1AlXE0oMKBJ+NhykAaP2DE  
wXxY9zHzvQk+8GhUG9baqp+lVIADUxaC6hEUlZtVC8fhy6Ep1XMXuDBZ0hTpDVPeUTOWHfkD4FykDIzr  
/VgvDp51v53yadV4LFsXUizJyvXi7dYGY2vbdmGdB54pcDQAzaMQ5Nn7ixVf sangyun0628@naver.c  
om
```


SSH key copy & paste

☰ Profile Settings

KHU-HUB

Search



Profile Account Applications Access Tokens Emails Password Notifications **SSH Keys** Preferences Audit Log

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

Before you can add an SSH key you need to [generate it](#).

Key

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQCj+JBo1lInIYIET0mFQOmGxXbw0sRUJeSxtx5mjIjTnfwc9vrZFryozBylQni/fNd  
/tq74SvijV3l3MOpCYuqvfx2Tsm0q0JHY32FEDkUZ5fshTWeNQRGNq1AH/RCBlk04kFjrps1Z0R3aT2o65h9UWzS4JSqci6RmVjv6pKQ3VQy1XQ46FMexe  
/dWwMy8iEqXeLi9FucqL/J1T8eTdnPJ1jUh5XiK7oNOOfSLDMvgohN45M3ikEBvszL+eBV7Daci  
/zWVNJFdmOdBPo4DFvHB5YKNUi8g+2eglu6a13X3NydiB4EK+HjEPdo7MVXgTJToCE0uAs3dFuBiPhi7tP675 khu@khu.ac.kr
```

Title

MyDesktop

Add key

SSH key 등록

☰ Profile Settings

KHU-HUB

Search



Profile Account Applications Access Tokens Emails Password Notifications **SSH Keys** Preferences Audit Log

SSH Key

Title: **MyDesktop**

Created on: **Sep 28, 2016 11:37am**

Fingerprint: 9a:da:b2:04:40:fe:52:2c:61:a2:cf:db:e4:34:b4:fa

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACj+JBo1lIniYlET0mFQ0mGxXbw0sRUJeSxtx5mjIjTnfwc9vrZFryozBylQni/
```

Remove

1. 저장소 생성
2. 사용자 정보 입력
3. 변경 적용
4. 프로젝트 상태 확인

GIT 사용

테스트용 프로젝트 생성

Projects

KHU-Hub

Search



New project

Create or Import your project from popular Git services

Project path

http://localhost/2017000000/

Project name

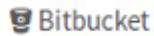
test

Want to house several dependent projects under the same namespace? [Create a group](#)

Import project from



GitHub



Bitbucket



GitLab.com



Google Code



Fogbugz






git Repo by URL

Project description (optional)

This is git test

Visibility Level (?)

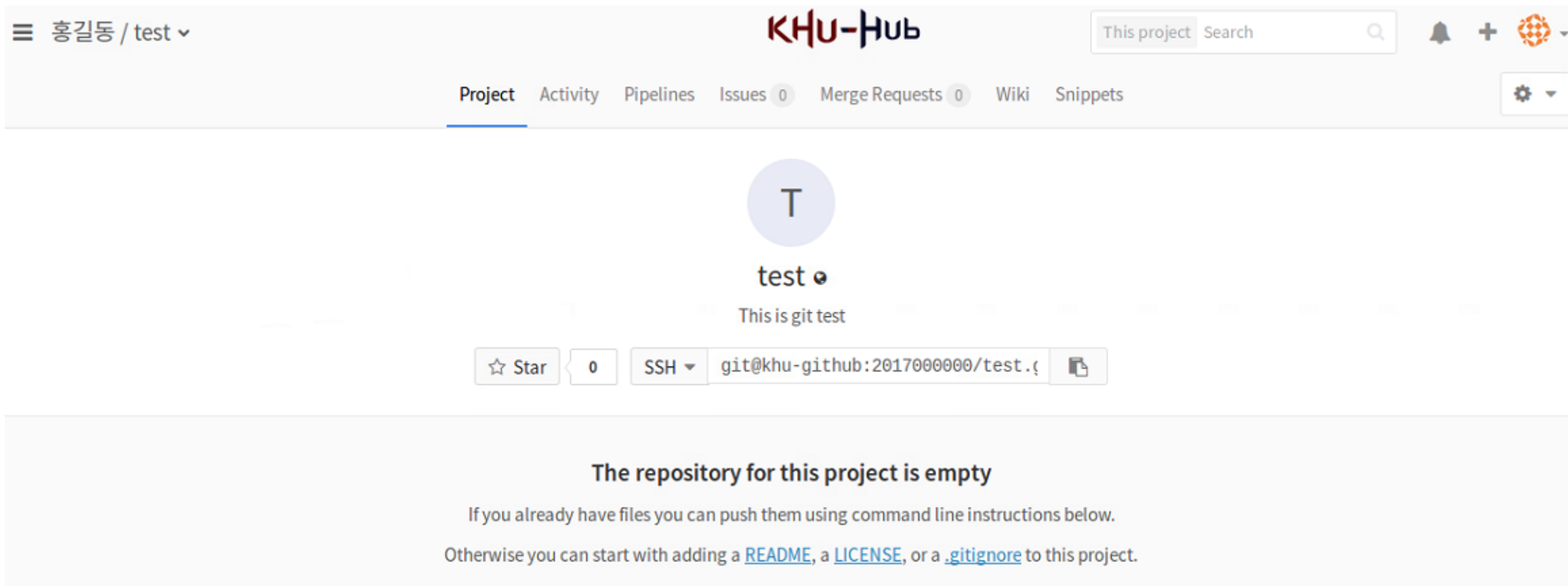
-  Private
Project access must be granted explicitly to each user.
-  Internal
The project can be cloned by any logged in user.
-  Public
The project can be cloned without any authentication.

Create project

Cancel

프로젝트 생성 시 초기 화면

- ❖ 해당 프로젝트의 URL만 안내
- ❖ Git을 통해 소스코드, 리소스 등을 업로드



☰ 홍길동 / test ▾

KKHU-Hub

This project Search 🔍

🔔 + 🌐 ▾

Project Activity Pipelines Issues 0 Merge Requests 0 Wiki Snippets ⚙️ ▾

T

test 🌐

This is git test

☆ Star 0 SSH ▾ git@khu-github:2017000000/test.ζ 📄

The repository for this project is empty

If you already have files you can push them using command line instructions below.

Otherwise you can start with adding a [README](#), a [LICENSE](#), or a [.gitignore](#) to this project.

원격 저장소의 test 프로젝트 가져오기

- ❖ git clone 명령어를 통해 원격 저장소의 프로젝트 가져오기
- ❖ GitHub 또는 KhuHub 페이지에 표기되어 있는 주소를 통해 클론

```
khu-github@khu-github:~$ git clone http://khu-github/2017000000/test.git
Cloning into 'test'...
warning: You appear to have cloned an empty repository.
Checking connectivity... done.
khu-github@khu-github:~$ ls
Desktop      Downloads      Music          Public         test
Documents    examples.desktop  Pictures       Templates      Videos
khu-github@khu-github:~$ cd test/
```

테스트 파일 추가

- ❖ 클론해 온 프로젝트에 임의의 테스트 파일 추가

```
khu-github@khu-github: ~/test
GNU nano 2.5.3      File: test.c      Modified
This is test file.

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^ Go To Line
```

add & commit

- ❖ 새로 추가된 test 파일을 add
- ❖ 변경 사항을 로컬 저장소에 commit

```
khu-github@khu-github:~/test$ nano test.c
khu-github@khu-github:~/test$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   test.c
```

commit 메시지 작성

```
khu-github@khu-github:~/test$ git add test.c
khu-github@khu-github:~/test$ git commit -m "Add test file."
[master (root-commit) 9c06f5e] Add test file.
1 file changed, 1 insertion(+)
create mode 100644 test.c
```


push

- ❖ 로컬 저장소의 변경 내역을 원격 저장소(GitHub/KhuHub)에 푸시
- ❖ `git push <브랜치이름>`

```
khu-github@khu-github:~/test$ git push origin
warning: push.default is unset; its implicit value has changed in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behavior, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

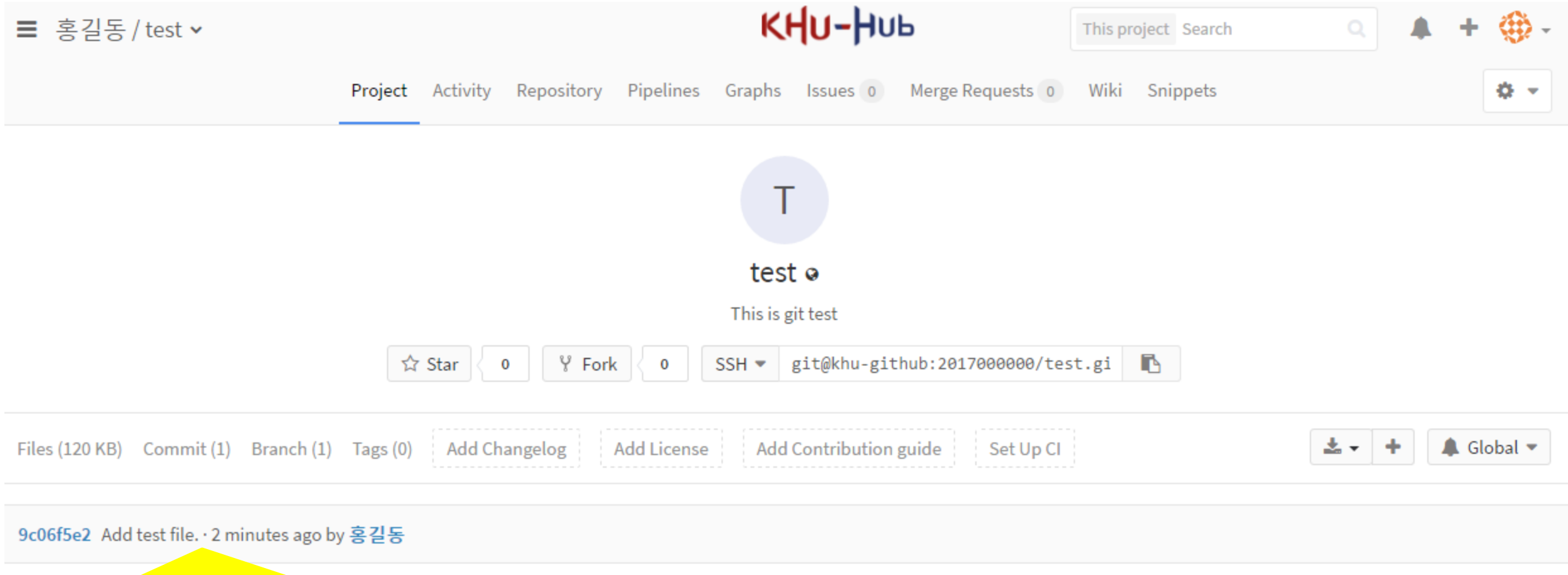
When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'http://khu-github': khu@khu.ac.kr
Password for 'http://khu@khu.ac.kr@khu-github':
Counting objects: 3, done.
Writing objects: 100% (3/3), 221 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To http://khu-github/2017000000/test.git
 * [new branch]      master -> master
```

원격 저장소에서 변경 내역 확인



The screenshot shows a GitHub repository page for a user named 'T' (profile picture with 'T'). The repository is named 'test' and has the description 'This is git test'. The page includes navigation tabs for Project, Activity, Repository, Pipelines, Graphs, Issues (0), Merge Requests (0), Wiki, and Snippets. Below the repository name, there are buttons for Star (0) and Fork (0), an SSH link 'git@khu-github:2017000000/test.gi', and a download icon. A bar at the bottom of the repository section contains links for Files (120 KB), Commit (1), Branch (1), Tags (0), Add Changelog, Add License, Add Contribution guide, Set Up CI, a download icon, a plus icon, and a Global notification bell. A commit history entry is visible: '9c06f5e2 Add test file. · 2 minutes ago by 홍길동'.

commit ID / commit 메시지 / 변경자

원격 저장소에서 변경 내역 확인

Project Activity **Repository** Pipelines Graphs Issues 0 Merge Requests 0 Wiki Snippets

Files Commits Network Compare Branches Tags

master test / + Find File

Name	Last Commit > 9c06f5e2 - 2 minutes ago Add test file. History	Last Update
test.c	Add test file.	2 minutes ago

예제에서 추가했던 테스트 파일

저장소 생성하기(명령어 : **git init**)

1. `mkdir git-test`
2. `cd git-test`
3. `git init`

```
hansangyun@hansangyun:~/바탕화면$ mkdir git-test
hansangyun@hansangyun:~/바탕화면$ cd git-test
hansangyun@hansangyun:~/바탕화면/git-test$ git init
Initialized empty Git repository in /home/hansangyun/바탕화면/git-test/.git/
```

저장소 받아오기(명령어 : **git clone**)

1. 로컬 저장소에서 받아오기

`git clone /로컬/저장소/경로`

2. 원격 저장소에서 받아오기

`git clone 사용자명@호스트:/원격/저장소/경로`

`git clone 저장소 주소(ex: git clone git://github.com/sangyunHan/kairos-xxx)`

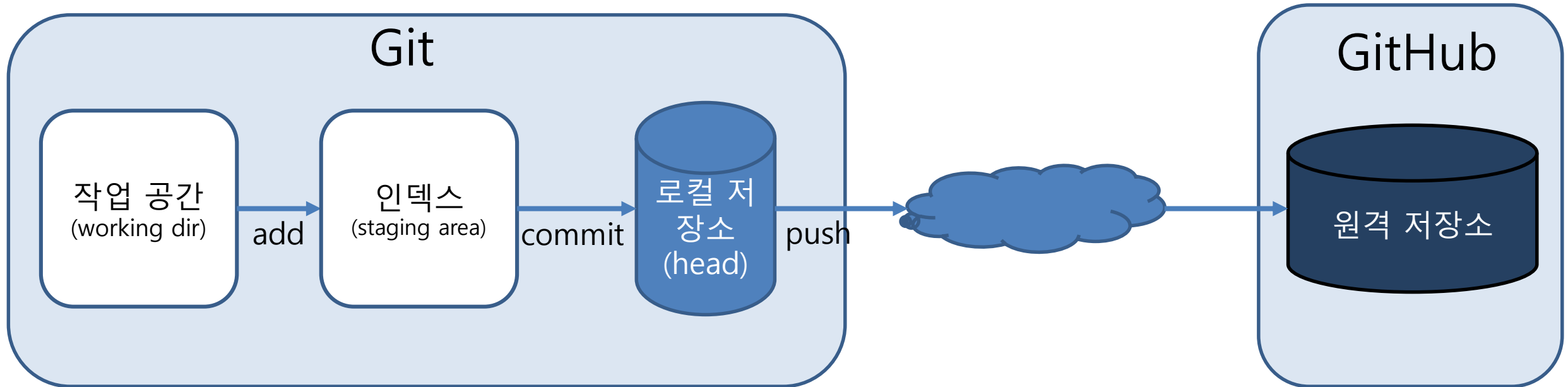
```
hansangyun@hansangyun:~/바탕화면$ git clone git://github.com/sangyunHan/kairos-xxx
Cloning into 'kairos-xxx'...
remote: Counting objects: 300, done.
remote: Total 300 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (300/300), 249.51 KiB | 57.00 KiB/s, done.
Resolving deltas: 100% (47/47), done.
Checking connectivity... done.
```

커미터 등록

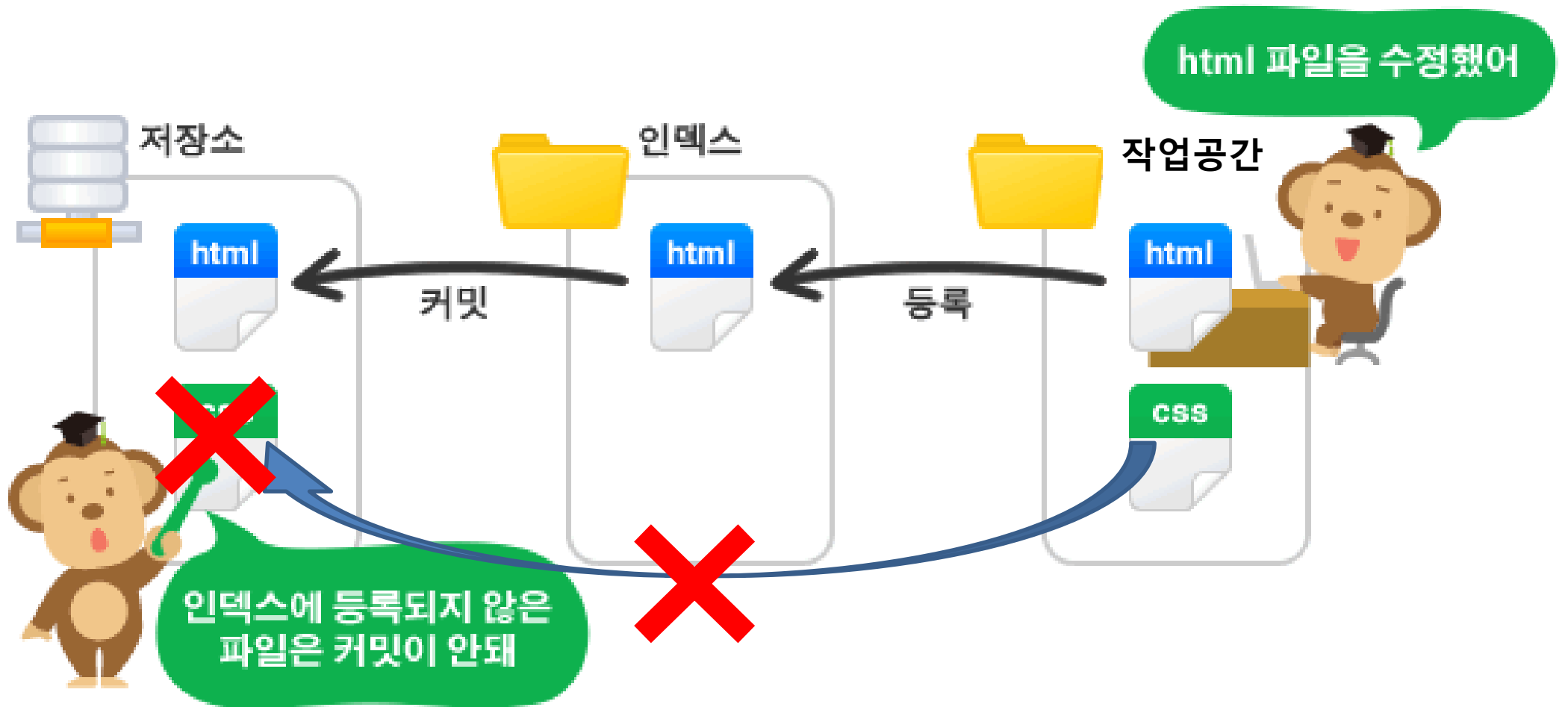
- ❖ Git을 사용하는 사용자에게 대한 정보를 git에 등록한다.
- ❖ 공동 작업을 할 때 각각의 버전이 누구의 작업인지 구별해주는 역할
- ❖ `git config --global user.name "사용자명"`
- ❖ `git config --global user.email "사용자 이메일"`

```
hansangyun@hansangyun:~/바탕화면/git-test$ git config --global user.name "sangyunHan"  
hansangyun@hansangyun:~/바탕화면/git-test$ git config --global user.email "sangyun0628@naver.com"
```

작업의 흐름



인덱스의 중요성!



git status

git add

git commit

ADD와 COMMIT

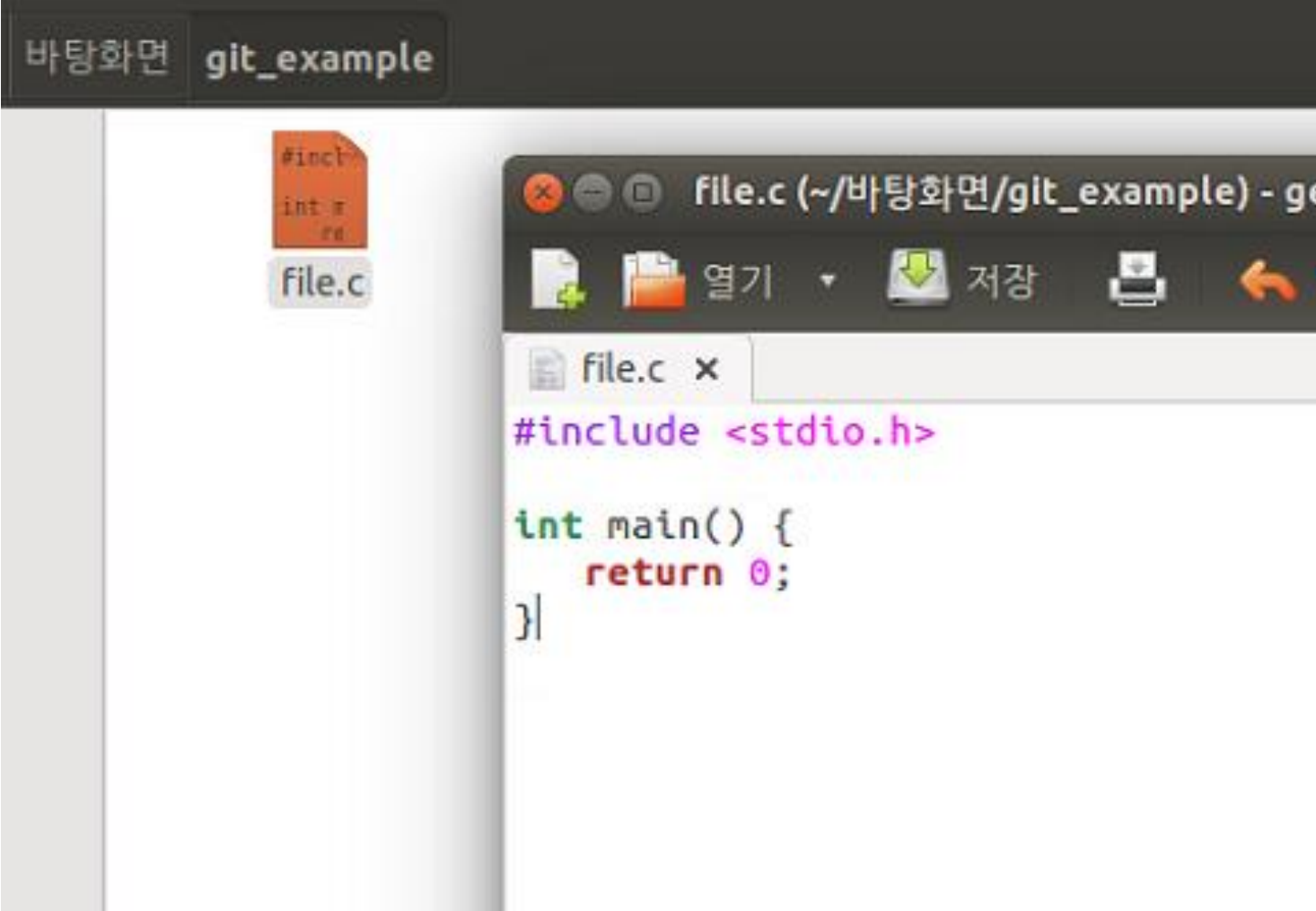
상태보기(명령어 : git status)

```
hansangyun@hansangyun:~/바탕화면/git_example$ git init
Initialized empty Git repository in /home/hansangyun/바탕화면/git_example/.git/
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
```

파일 추가



The screenshot shows a code editor interface. At the top, there are two tabs: '바탕화면' (Home) and 'git_example'. Below the tabs, a file named 'file.c' is shown with a preview of its content: '#incl', 'int m', and 're'. To the right, a window titled 'file.c (~/바탕화면/git_example) - ge' is open, displaying the following code:

```
#include <stdio.h>

int main() {
    return 0;
}
```

파일 추가 후 상태

```
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file.c
        file.c~

nothing added to commit but untracked files present (use "git add" to track)
```

파일 추가 후 상태

```
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    file.c
    file.c~

nothing added to commit but untracked files present (use "git add" to track)
```

인덱스(staging area)에 등록되지 않아서 버전관리가 되고 있지 않은 파일들

add : 인덱스에(staging area) 등록

- ❖ git add 파일명(ex: git add file.c)

```
hansangyun@hansangyun:~/바탕화면/git_examples$ git add file.c
hansangyun@hansangyun:~/바탕화면/git_examples$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

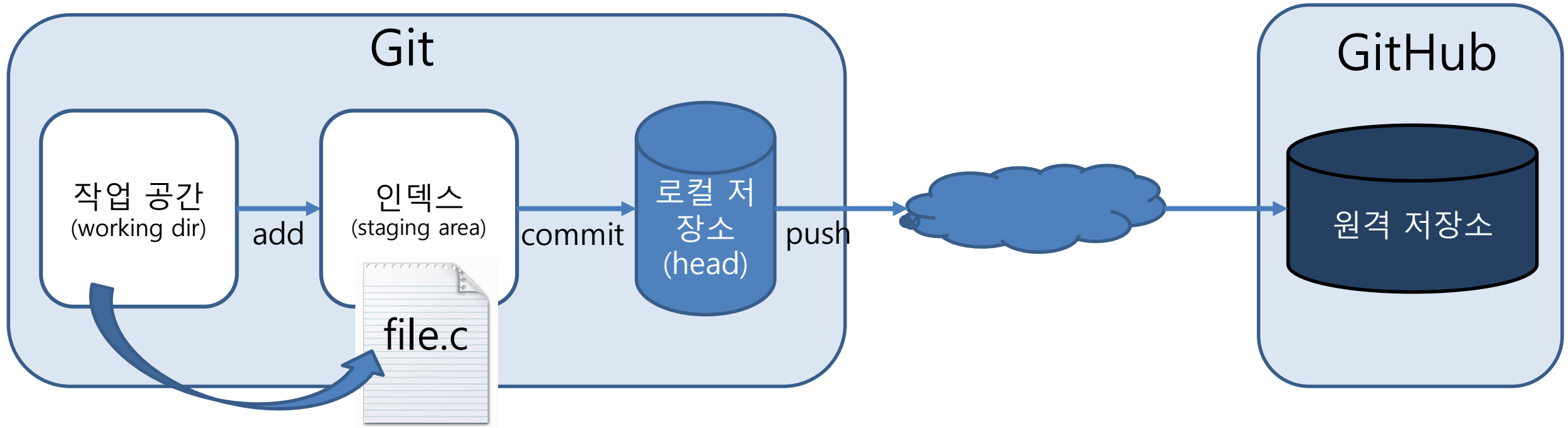
        new file:   file.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file.c~

hansangyun@hansangyun:~/바탕화면/git_examples$
```

작업의 흐름



commit : 저장소에 등록

- ❖ `git commit -m "이번 commit에 대한 설명"`

```
hansangyun@hansangyun:~/바탕화면/git_example$ git commit -m "this is first commit"
[master (root-commit) 614bffd] this is first commit
1 file changed, 5 insertions(+)
create mode 100644 file.c
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    file.c~

nothing added to commit but untracked files present (use "git add" to track)
```


commit : 저장소에 등록

- ❖ `git commit -m "이번 commit에 대한 설명"`

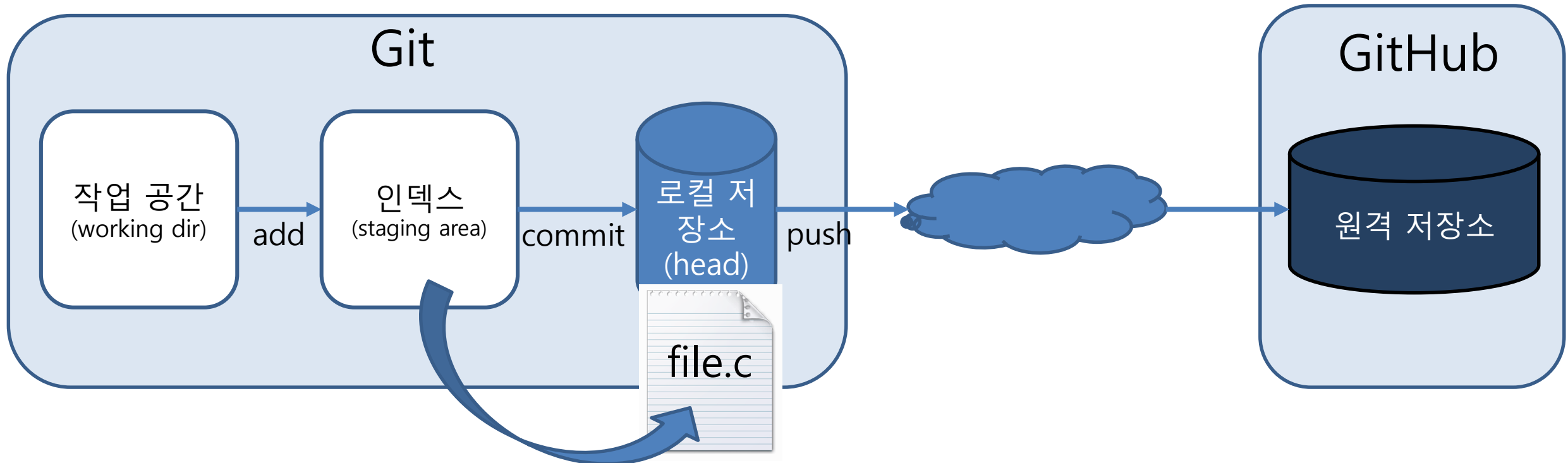
```
hansangyun@hansangyun:~/바탕화면/git_example$ git commit -m "this is first commit"
[master (root-commit) 614bffd] this is first commit
1 file changed, 5 insertions(+)
create mode 100644 file.c
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

file.c~

nothing added to commit but untracked files present (use "git add" to track)
```


커밋에 대한 메시지

작업의 흐름



파일 수정

바탕화면 git_example



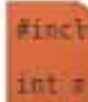
file.c

```
file.c (~/바탕화면/git_example) - ge
열기 저장
file.c x
#include <stdio.h>

int main() {
    return 0;
}
```

코드 추가

바탕화면 git_example



file.c

```
file.c (~/바탕화면/git_example)
열기 저장
file.c x
#include <stdio.h>

int main() {
    printf("change 1");
    return 0;
}
```

add 후, status 확인

```
hansangyun@hansangyun:~/바탕화면/git_example$ git add file.c
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   file.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)

file.c~
```

작업 이력 확인(명령어 : git log)

```
hansangyun@hansangyun:~/바탕화면/git_example$ git log
commit b3e687450a18671e1e3e64d454f790374800ab4b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:40:50 2014 +0900

    wow this is second commit

commit 614bffd7a3530dd3cbe1ce9292bfad6abf901f7b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:23:45 2014 +0900

    this is first commit
```

작업 이력 확인(명령어 : git log)

```
hansangyun@hansangyun:~/바탕화면/gtc_examples$ git log
commit b3e687450a18671e1e3e64d454f790374800ab4b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:40:50 2014 +0900

    wow this is second commit

commit 614bffd7a3530dd3cbe1ce9292bfad6abf901f7b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:23:45 2014 +0900

    this is first commit
```

커밋 리비전

각각의 커밋에 대한 고유 ID
SHA-1 해시값으로 구성

작업 이력 확인(명령어 : git log)

```
hansangyun@hansangyun:~/바탕화면/git_example$ git log
commit b3e687450a18671e1e2e64d454f790374800ab4b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:40:50 2014 +0900

    wow this is second commit

commit 614bffd7a2520dd3cbe1ce9292bfad6abf90217b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:23:45 2014 +0900

    this is first commit
```

커밋 작성자 정보

작업 이력 확인(명령어 : git log)

```
hansangyun@hansangyun:~/바탕화면/git_example$ git log
commit b3e687450a18671e1e3e64d454f790374800ab4b
Author: sangyunHan <sangyun0628@naver.com>
Date: Fri Dec 26 18:40:50 2014 +0900

    wow this is second commit

commit 614bffd7a3530dd3cbe1ce9292bfad6abf90
Author: sangyunHan <sangyun0628@naver.com>
Date: Fri Dec 26 18:23:45 2014 +0900

    this is first commit
```

커밋 날짜와 시간

작업 이력 확인(명령어 : git log)

```
hansangyun@hansangyun:~/바탕화면/git_example$ git log
commit b3e687450a18671e1e3e64d454f790374800ab4b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:40:50 2014 +0900

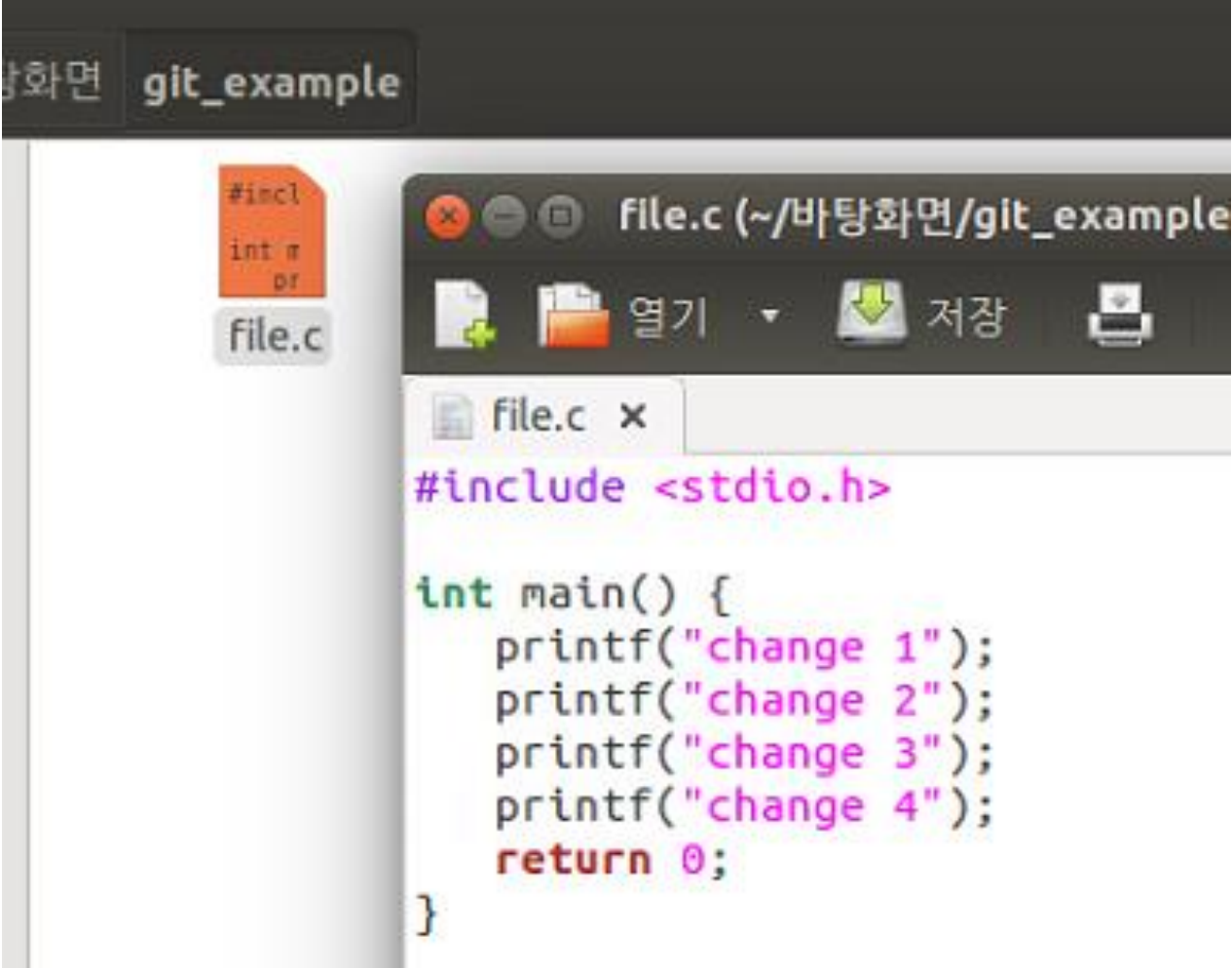
    wow this is second commit

commit 614bffd7a3530dd3cbe1ce9292bfad6abf901f7b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:23:45 2014 +0900

    this is first commit
```

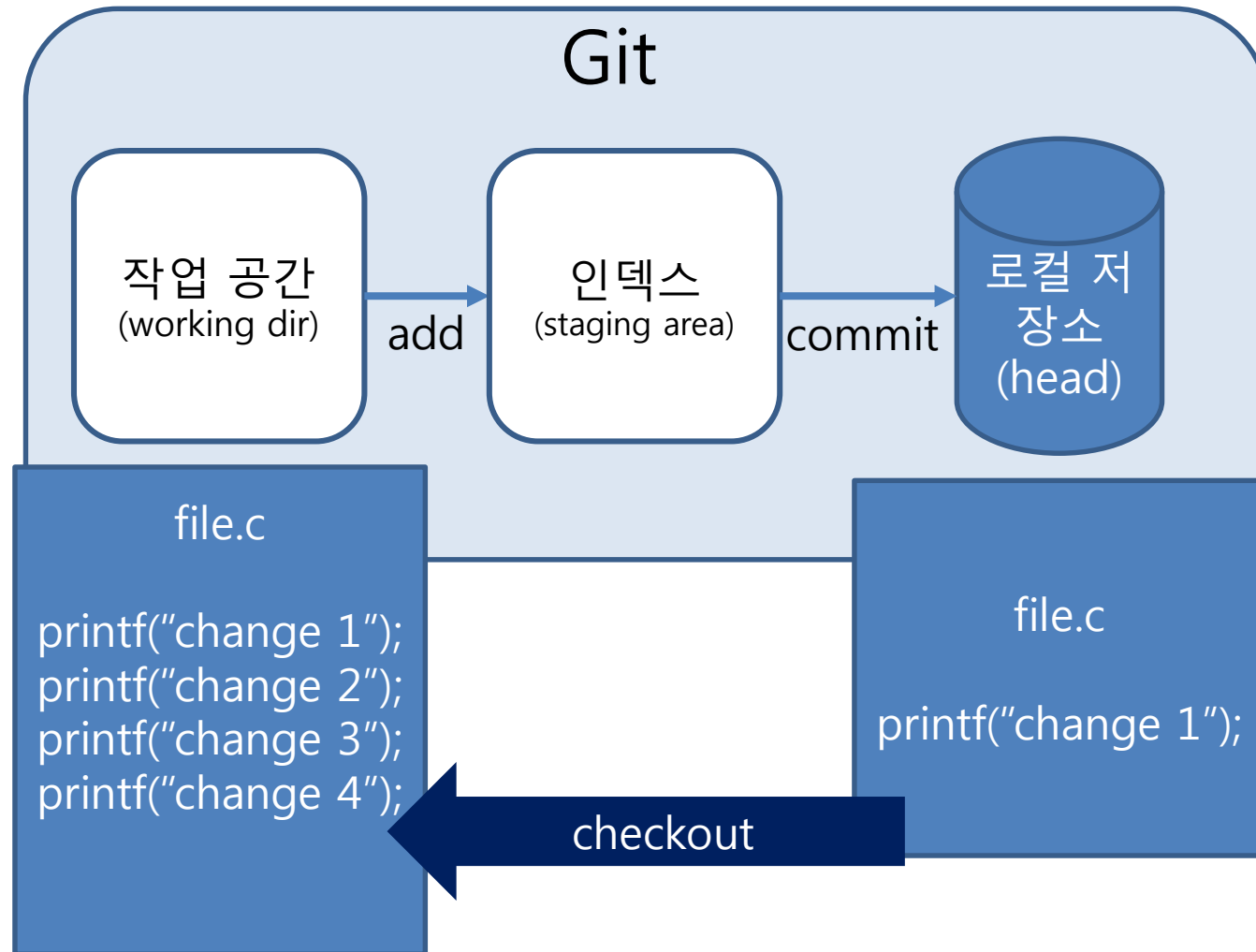
커밋에 대한 메시지들

변경 취소(이전 commit 내용 가져오기)



```
#include <stdio.h>

int main() {
    printf("change 1");
    printf("change 2");
    printf("change 3");
    printf("change 4");
    return 0;
}
```



변경 취소(명령어 : git checkout 파일명)

```
hansangyun@hansangyun: ~/바탕화면/git_example
hansangyun@hansangyun:~/바탕화면/git_example$ git checkout file.c
hansangyun@hansangyun:~/바탕화면/git_example$
```

```
file.c x
#include <stdio.h>

int main() {
    printf("change 1");
    return 0;
}
```

← checkout 결과

```
file.c (~/바탕화면/git_example)
file.c x
#include <stdio.h>

int main() {
    printf("change 1");
    printf("change 2");
    printf("change 3");
    printf("change 4");
    return 0;
}
```