

# CS311 GitLab

---

- <https://cs311.kaist.ac.kr>
- Please sign in with your pre-assigned account
  - Username: st[student id]
  - Ex) st20201234
  - Password: **It will be released in grade section of KLMS.**
- If you have any questions, please contact us by [cs311\\_ta@casys.kaist.ac.kr](mailto:cs311_ta@casys.kaist.ac.kr)

# Fork the deployed project to your repository (1)

- Do not click “Star”
  - If you do, other students can access your private code

The screenshot shows a GitHub project page for 'Project1-MIPS-Assembler'. At the top, there's a navigation bar with 'cs311 > Project1-MIPS-Assembler'. Below it is the project header with a 'P' icon, the project name 'Project1-MIPS-Assembler' (locked), 'Project ID: 37', and a 'fork' icon with the number '1'. A red arrow points to this 'Fork' button. Below the header, there are statistics: '3 Commits', '1 Branch', '0 Tags', and '215 KB Project Storage'. A progress bar is shown below these stats. The main content area displays a commit titled 'untar sample' by 'Kwanghoon-Choi' from 19 hours ago, with a commit hash '6bfa3bb'. Below the commit, there are dropdown menus for 'master' and 'project1-mips-assembler /', and buttons for 'Find file', 'Web IDE', 'Clone', and 'Auto DevOps enabled'. There's also a dashed box labeled 'Add Wiki'. At the bottom, a table lists files with their last commit details:

Name	Last commit	Last update
sample_input	untar sample	19 hours ago
sample_output	untar sample	19 hours ago
Makefile	init	20 hours ago
main.c	init	20 hours ago

# Fork the deployed project to your repository (2)

- Set “Project URL” to your namespaces (username)
  - Not to change other forms

cs311 > Project1-MIPS-Assembler > Fork project

 **Fork project**

A fork is a copy of a project.  
Forking a repository allows you to make changes without affecting the original project.

**Project name**  
Project1-MIPS-Assembler  
Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

**Project URL**  
https://cs311.kaist.ac.kr/ **Select a namespace** ▾  
Want to organize several de  
space? [Create a group](#)

**Project slug**  
project1-mips-assembler

**Namespaces**  
 cs311 leesh6796

**Project description (optional)**

**Visibility level** ⓘ  
 **Private**  
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.  
 **Internal**  
The project can be accessed by any logged in user.  
 **Public**  
The project can be accessed without any authentication.

**Fork project** **Cancel**



# Fork the deployed project to your repository (3)

- If you can see below screen, you are success to fork

The screenshot shows a GitHub repository page for 'Project1-MIPS-Assembler'. At the top, a blue banner displays the message 'Profile was successfully updated'. Below the banner, the repository name 'Project1-MIPS-Assembler' is shown with a lock icon, and the text 'Project ID: 39'. To the right are buttons for notifications (Bell), stars (Star 0), and forks (Fork 0). Below this, it says '3 Commits', '1 Branch', '0 Tags', and '102 KB Project Storage'. A note indicates it was 'Forked from cs311 / Project1-MIPS-Assembler'. A horizontal progress bar is mostly blue, with a green segment indicating the current commit. A commit card for 'untar sample' by Kwanghoon-Choi, authored 19 hours ago, is displayed, showing the commit hash '6bfa3bba'. Below the commit card is a navigation bar with 'master' and a dropdown, followed by 'project1-mips-assembler / +', and buttons for 'Find file', 'Web IDE', 'Clone', and 'Download'. A series of dashed buttons offer options like 'Auto DevOps enabled', 'Add README', 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', 'Add Kubernetes cluster', 'Add Wiki', and 'Configure Integrations'. At the bottom, a table lists the repository's files: 'sample\_input' (last commit 'untar sample', last update '19 hours ago'), 'sample\_output' (last commit 'untar sample', last update '19 hours ago'), 'Makefile' (last commit 'init', last update '20 hours ago'), and 'main.c' (last commit 'init', last update '20 hours ago').

Name	Last commit	Last update
sample_input	untar sample	19 hours ago
sample_output	untar sample	19 hours ago
Makefile	init	20 hours ago
main.c	init	20 hours ago

# Generate and register public SSH key (1)

---

- Please generate the SSH key for gitlab server access.

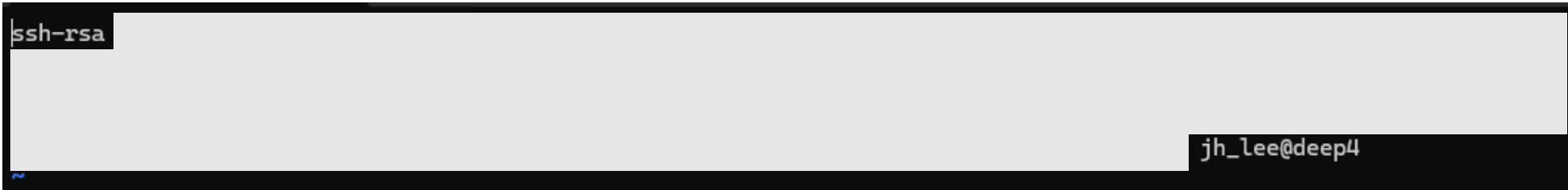
```
jh_lee@deep4:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jh_lee/.ssh/id_rsa):
Created directory '/home/jh_lee/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jh_lee/.ssh/id_rsa
Your public key has been saved in /home/jh_lee/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:eijoqnQPpOzXyLe01+9I1xZAh6oSJgjXo2zFp/39gyQ jh_lee@deep4
The key's randomart image is:
+---[RSA 3072]---+
|   o     ...
| . . = . ...
| oo o = ...
| ..+o. ...
| .o... .S. .
| . oo . oE +
| +oo= +..+ =
| o..++=.oo o o
| +oo.++.oo .
+---[SHA256]---+
```

# Generate and register public SSH key (2)

---

- Move to .ssh directory and check the content of id\_xxx.pub file.
- Please **copy the entire content** including the *user@server\_name* part.
- Press Esc + “:q” to quit vim.

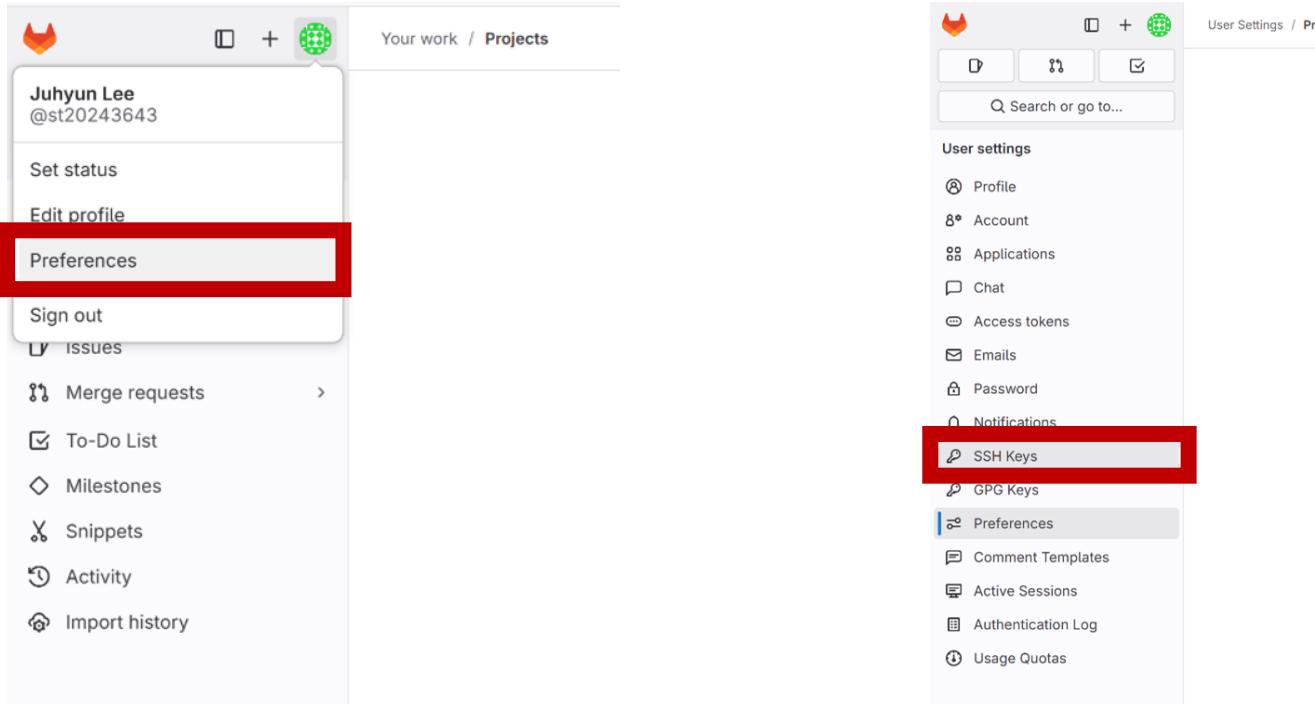
```
jh_lee@deep4:~$ cd .ssh
jh_lee@deep4:~/ssh$ ls
id_rsa  id_rsa.pub
jh_lee@deep4:~/ssh$ vim id_rsa.pub|
```



A screenshot of a terminal window. The title bar says "ssh-rsa". The bottom right corner shows the user information "jh\_lee@deep4". The main area of the terminal is empty, showing only the prompt character (~).

# Generate and register public SSH key (3)

- Click “Preferences” and then “SSH Keys”



# Generate and register public SSH key (4)

- Click “Add new key”

The screenshot shows the GitLab user settings interface. The left sidebar has a navigation menu with various options like Profile, Account, Applications, Chat, Access tokens, Emails, Password, Notifications, SSH Keys (which is selected and highlighted with a blue bar), GPG Keys, Preferences, Comment Templates, Active Sessions, Authentication Log, and Usage Quotas. The main content area is titled "User Settings / SSH Keys". It features a search bar labeled "Search settings". Below it is a section titled "SSH Keys" with a sub-section header "Your SSH keys 0". To the right of this section is a large purple circular icon containing a white padlock. Below the icon, the text "There are no SSH keys with access to your account" is displayed. In the top right corner of the "Your SSH keys" section, there is a button labeled "Add new key". A red rectangular box surrounds this "Add new key" button, and a red arrow points from the top right towards it, indicating where the user should click.

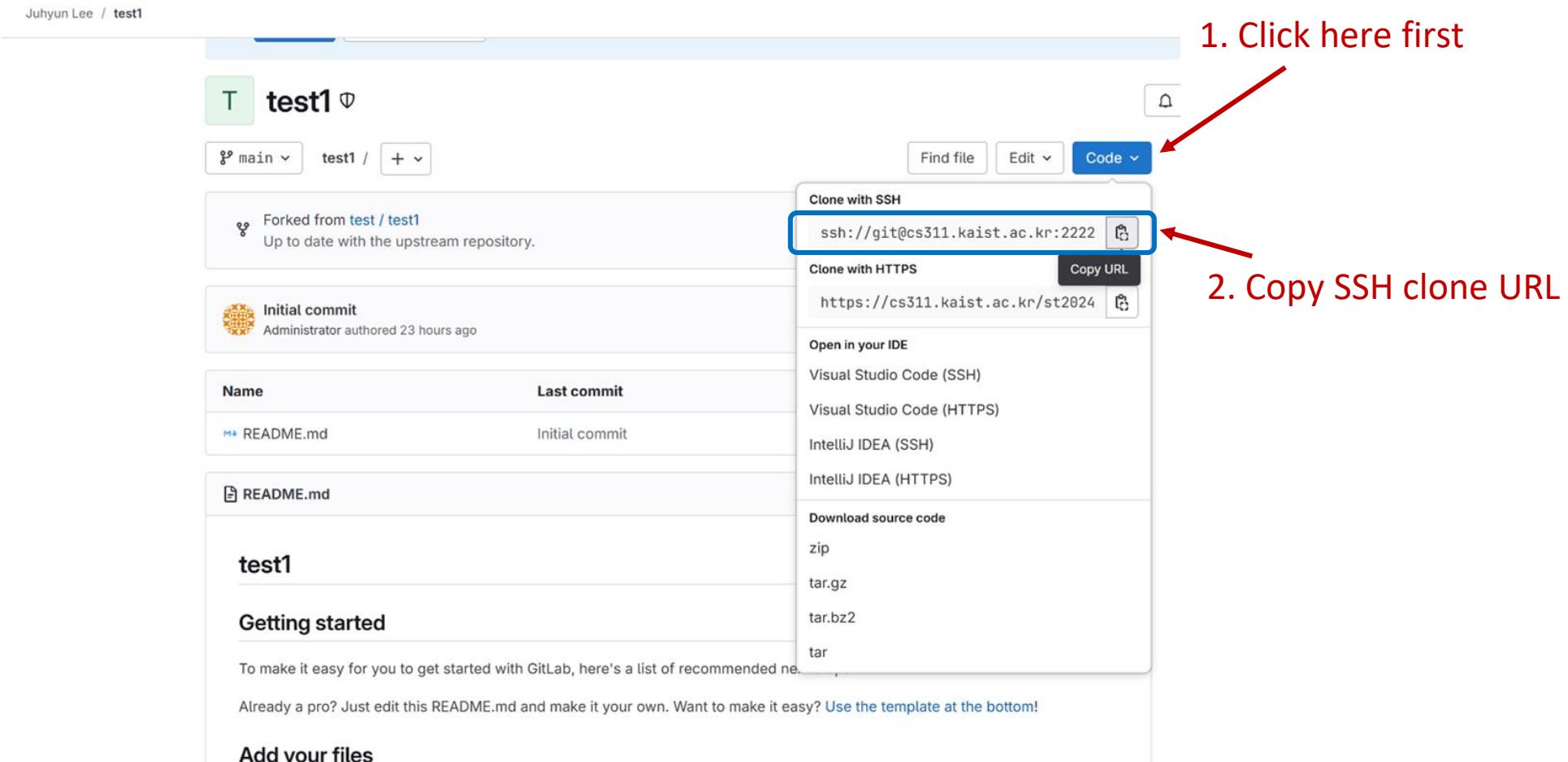
# Generate and register public SSH key (4)

- Paste the public ssh key and click “Add key”.

The screenshot shows the 'User Settings / SSH Keys' page on GitLab. At the top, there is a search bar labeled 'Search settings'. Below it, the 'SSH Keys' section is titled 'SSH Keys' and includes a note: 'SSH keys allow you to establish a secure connection between your computer and GitLab. SSH fingerprints verify that the client is connecting to the correct host. Check the [current instance configuration](#)'. A table titled 'Your SSH keys' shows one entry: 'ssh-rsa' with a fingerprint 'jh\_lee@deep4'. Below this table, the 'Add an SSH key' form is displayed. It contains fields for 'Key' (containing 'ssh-rsa'), 'Title' (containing 'jh\_lee@deep4'), 'Usage type' (set to 'Authentication & Signing'), and an 'Expiration date' set to '2026-03-07'. At the bottom of the form, there are 'Add key' and 'Cancel' buttons, with the 'Add key' button being highlighted with a red box.

# Clone the deployed project (1)

- Please clone the project that you forked with **SSH** (do not use HTTPS)



# Clone the deployed project (2)

---

- Please clone the project that you forked with **SSH** (do not use HTTPS)

```
jhlee@deep9:~$ git clone ssh://git@cs311.kaist.ac.kr:2222/st20243643/test1.git
Cloning into 'test1'...
The authenticity of host '[cs311.kaist.ac.kr]:2222 ([143.248.188.41]:2222)' can't be established.
ED25519 key fingerprint is SHA256:pWaG1211+rL2mW/9N4r1XQyqq2y+f+dfRTtaxjonimo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[cs311.kaist.ac.kr]:2222' (ED25519) to the list of known hosts.
remote: Enumerating objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 3 (from 1)
Receiving objects: 100% (3/3), done.
```

# Submit (1)

---

- Submit your work to your private GitLab repository by adding a “submit” tag.
- Please follow the steps below when submitting.
  1. Commit and push your **code** and **Makefile** to your remote repository.
  2. Type the following command in your working directory.
    - `git tag -a submit -m "whatever message you want"`
    - `git push origin submit`

# Submit (2)

- If you success submitting with tags, you can see below screen

P Project1-MIPS-Assembler 

Project ID: 40 

-o 4 Commits  1 Branch  1 Tag  133 KB Project Storage

Forked from [cs311 / Project1-MIPS-Assembler](#)

A red arrow points to the "1 Tag" button, with the text "A new tag appears" written above it.

이상현 > Project1-MIPS-Assembler > Tags

Tags give the ability to mark specific points in history as being important

 submit

-o [757a5ef2](#) · Complete · 21 minutes ago

Submit

# Delete the submit tag and resubmit

---

- If you have modified the code after submission or submitted it incorrectly, you need to delete the tag and create it again.
- Type the following command in your working directory to delete tag.
  - `git tag -d submit`
  - `git push origin main :submit`
- Submit your code and Makefile with tag again.