Git Tutorial

Introduction to Intelligent Systems

Robot Learning Lab

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
SEOUL NATIONAL UNIVERSITY

RLLAB
http://rllab.snu.ac.kr

Overview

- Introduction : What is Git?
- How does git work?
- Git Tutorial

Introduction

- **What is Git?**

- **Git - Version Control System**

- **Github - Code hosting platform for version control and collaboration.**

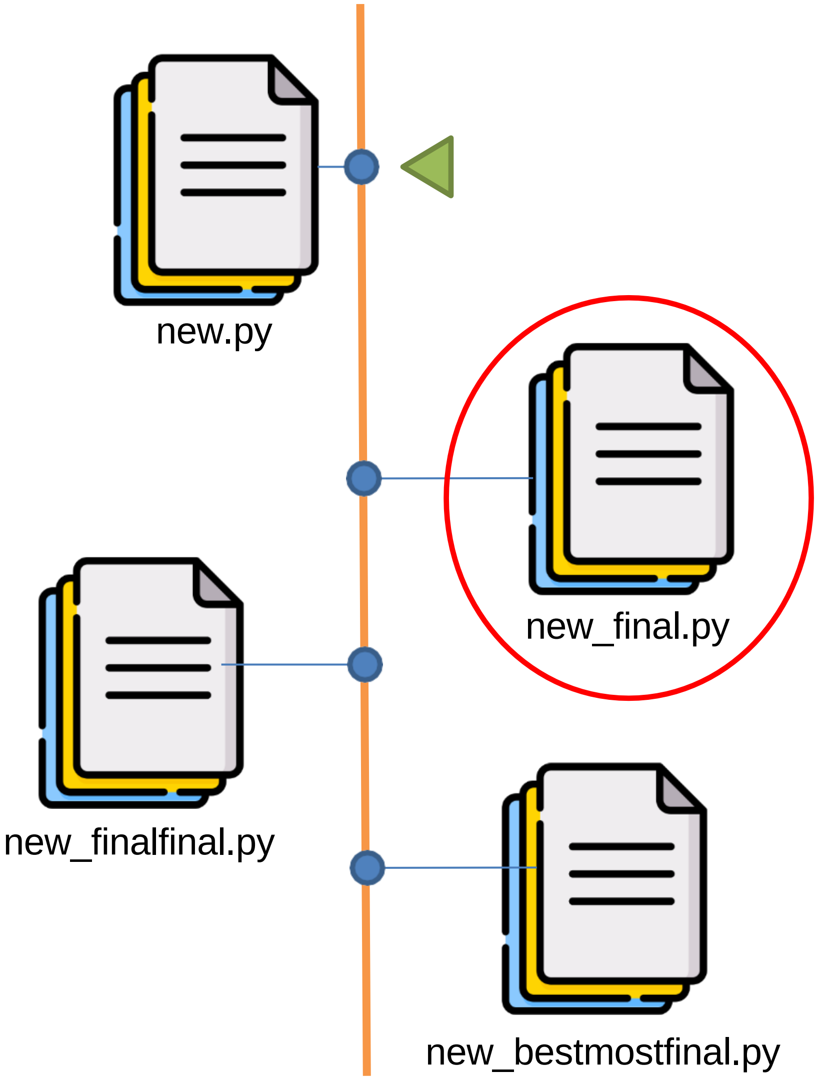
Introduction

- **Version Control**

- **Version control keeps record of your changes in codes**

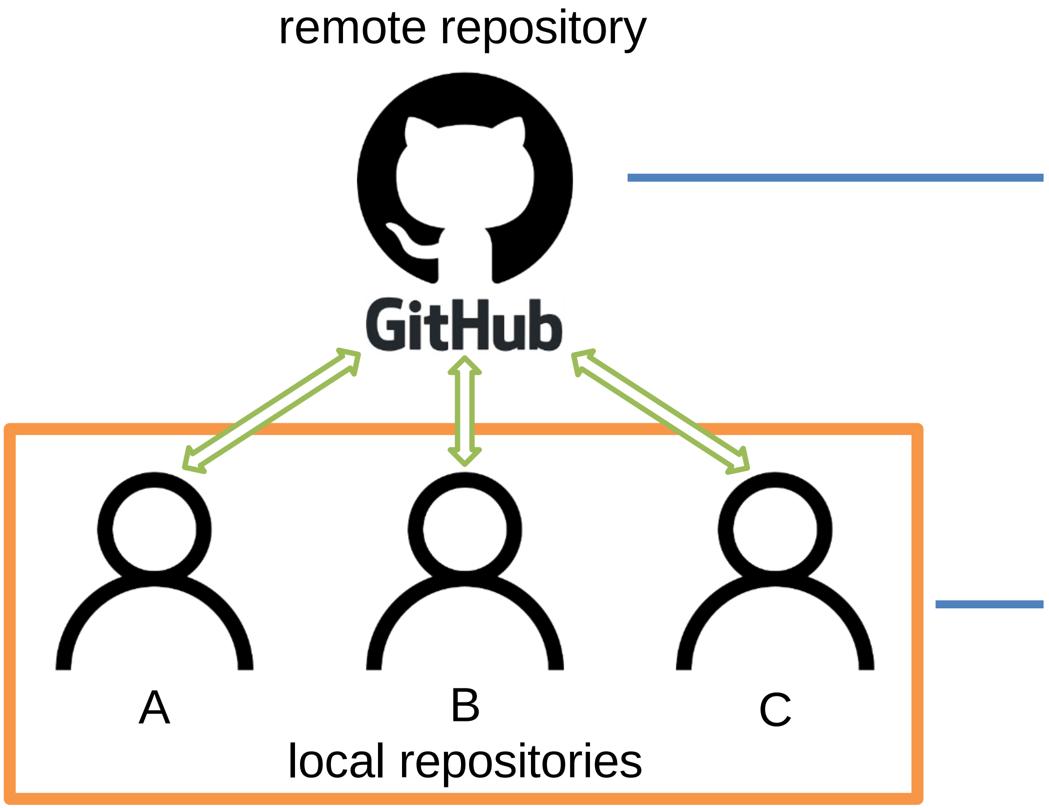
- **Allows you to revert any changes and go back to a previous state**

- **We can restore our previous codes!**



How does Git work?

- **Repository**

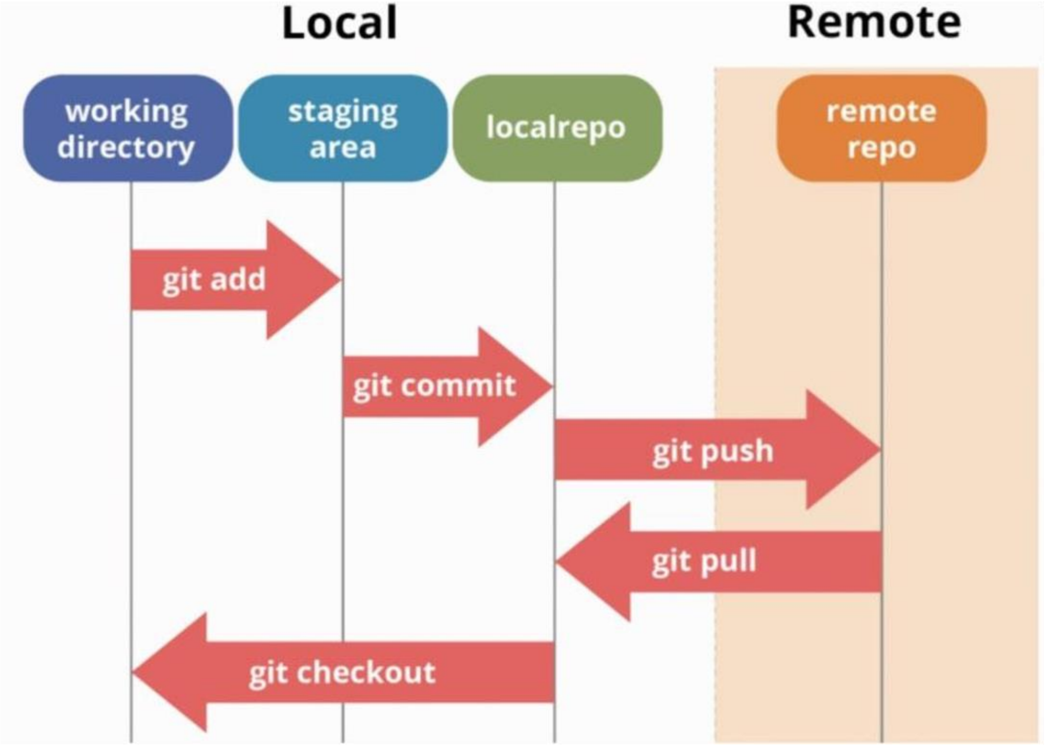


Files are managed in remote server. Users can push(upload)/pull(download) files to this repository.

Local repository stores files in my local PC.

How does Git work?

- **Basic Git commands**



git init
create a new local repository

git remote add origin <server>
connect to a remote repository

git add [file]
adds a file to the staging area

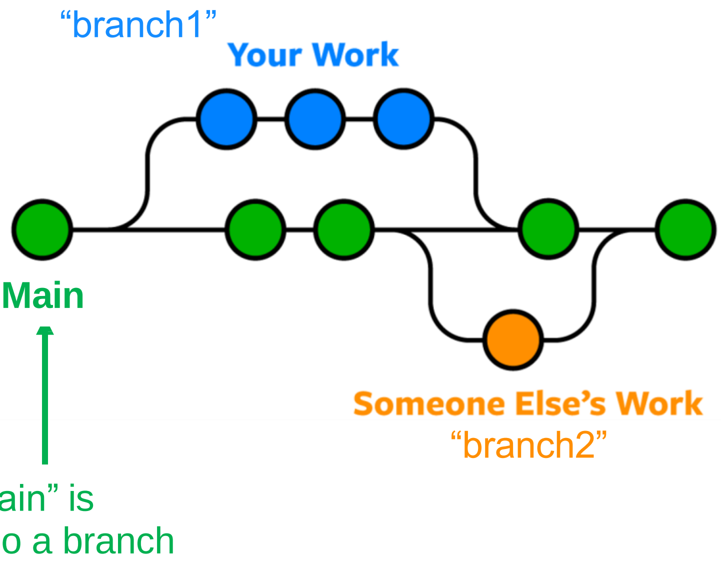
git commit –m “commit message”
records staged files in version history

git push origin <branchname>
sends commited changes of your branch to your remote repository

git pull origin <branchname>
update from the remote repository

How does Git work?

- **What is Branch?**



git branch
lists all the branches in your repo and show your current branch

git checkout –b <branchname>
create a new branch and switch to it

git checkout <branchname>
switch from one branch to another

git merge <branchname1> <branchname2>
adds a file to the staging area

Git Tutorial

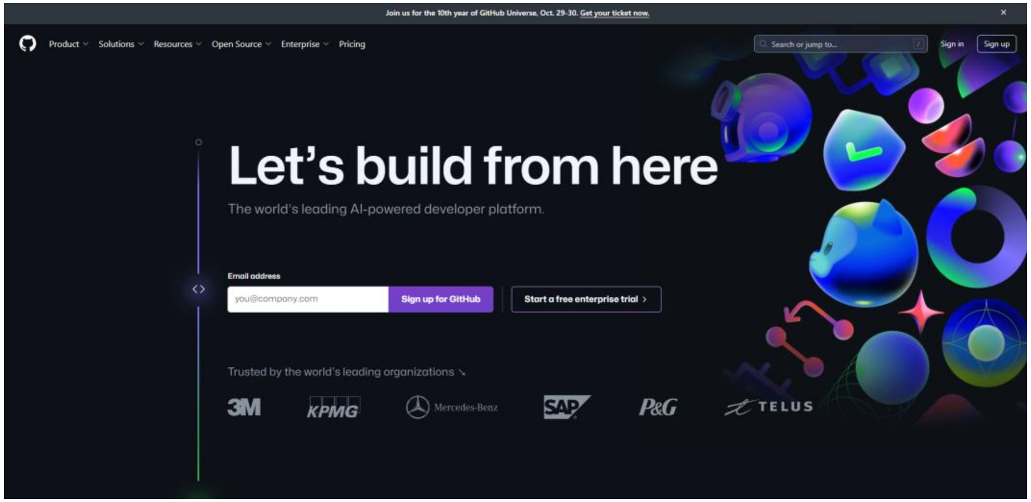
1) Install Git and create Github account

- Linux

```
sudo apt-get install git
```

- Create Github account

  https://github.com/



Join us for the 10th year of GitHub Universe, Oct. 29-30. Get your ticket now.

Let's build from here

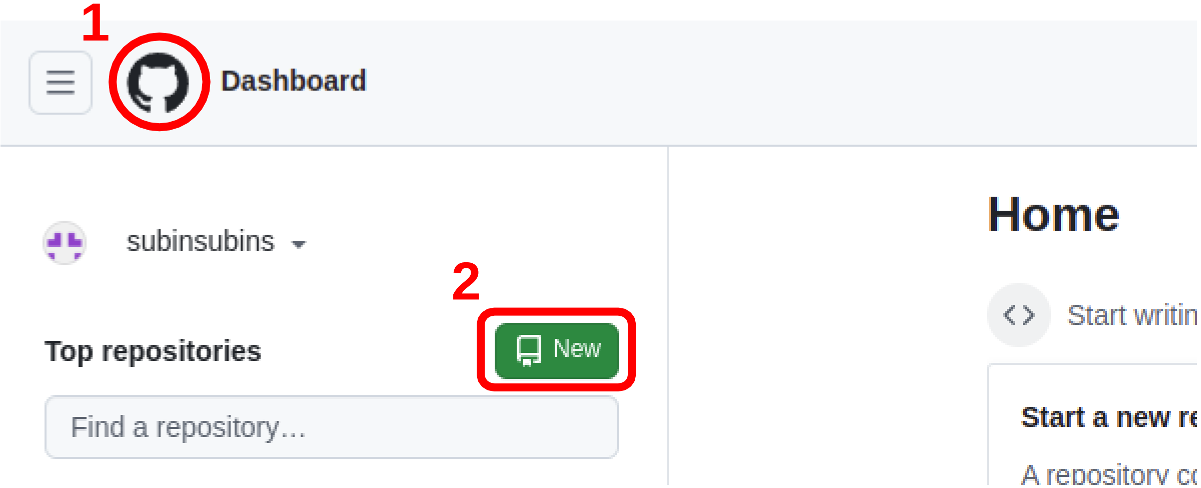
The world's leading AI-powered developer platform.

Email address

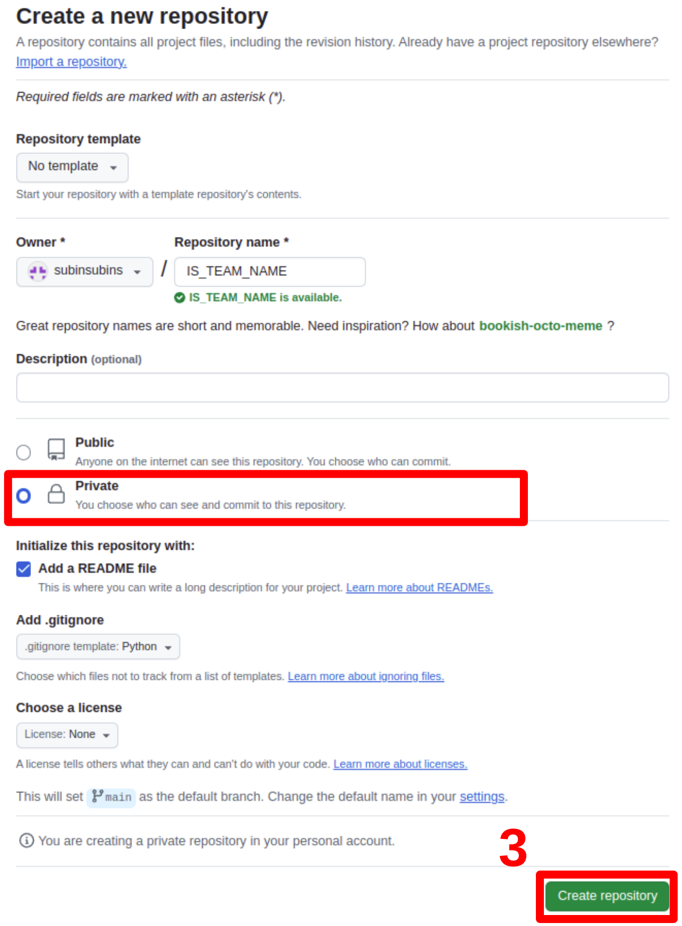
Trusted by the world's leading organizations ↘

2) Make your own private repository

Github web page -> new repository



2) Make your own private repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner * subinsubins ▾ / **Repository name *** IS_TEAM_NAME

✓ IS_TEAM_NAME is available.

Great repository names are short and memorable. Need inspiration? How about **bookish-octo-meme** ?

Description (optional)

○ **Public** Anyone on the internet can see this repository. You choose who commit.

● **Private** You choose who can see and commit to this repository.

Initialize this repository with:

☑ **Add a README file** This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: Python ▾

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None ▾

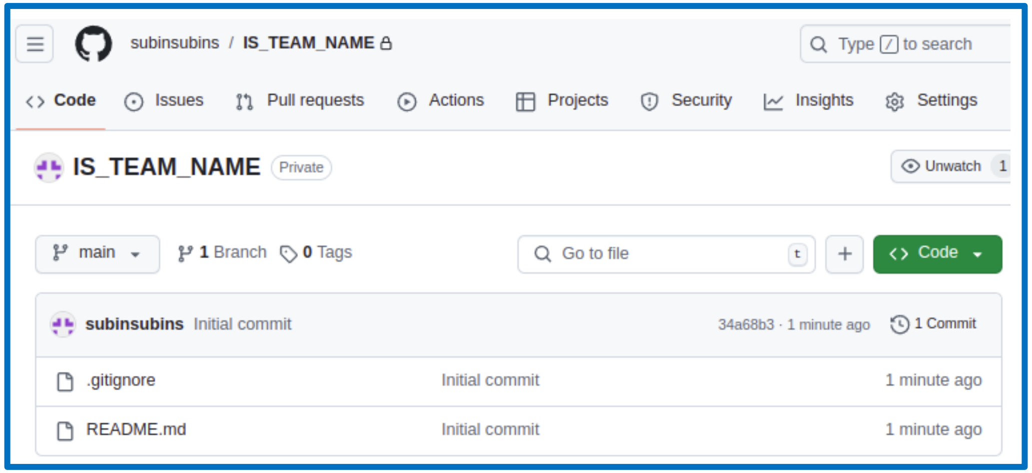
A license tells others what they can and can't do with your code. Learn more about licenses.

This will set `main` as the default branch. Change the default name in your settings.

ⓘ You are creating a private repository in your personal account.

Make sure to make your repository "private" so that other teams cannot view your team's code. Repository name should follow IS_{TEAM_NAME} format. (e.g. IS_RLLAB)

Main page of your repository



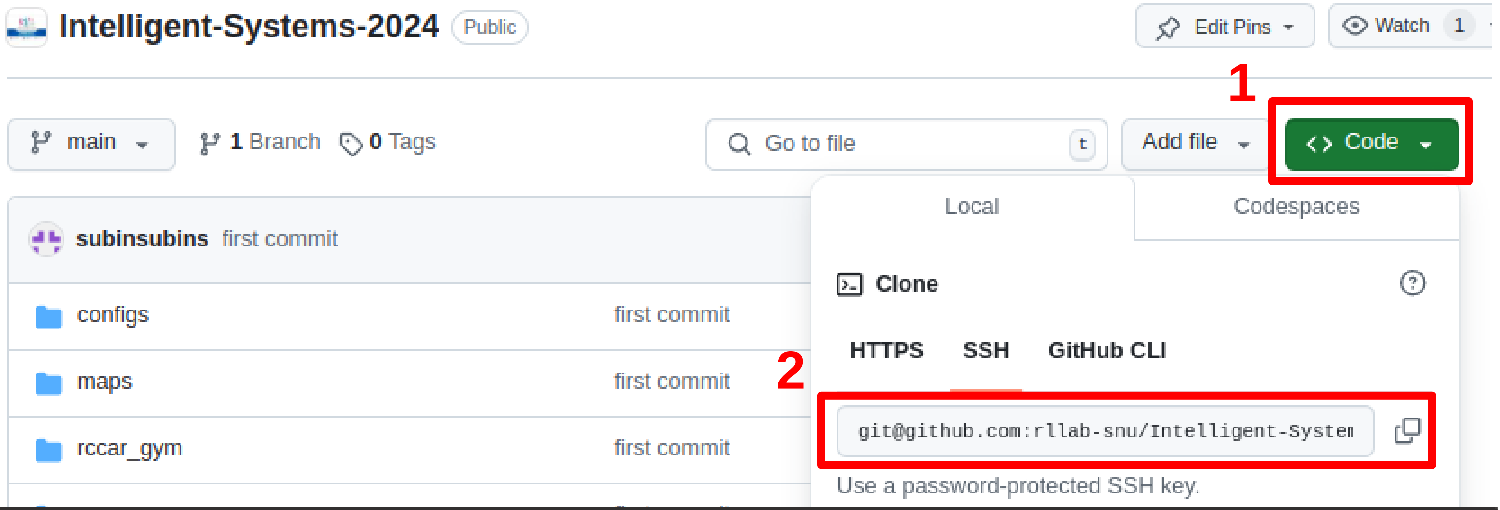
subinsubins Initial commit 34a68b3 · 1 minute ago 1 Commit

.gitignore	Initial commit	1 minute ago
README.md	Initial commit	1 minute ago

Git Tutorial

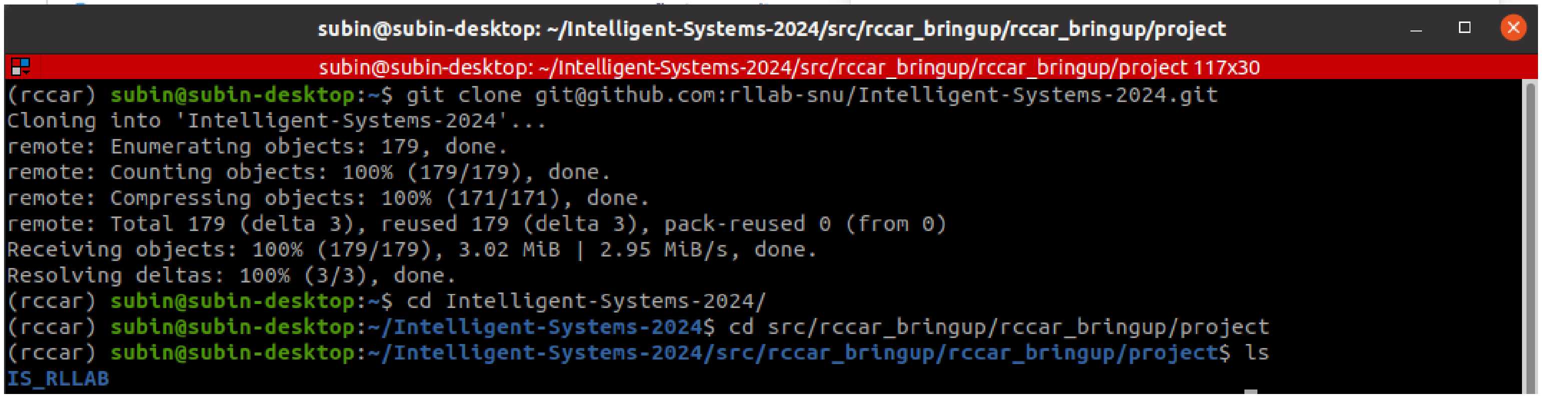
3) Clone Intelligent-Systems-2024 repository on your local PC

git clone git@github.com:rllab-snu/Intelligent-Systems-2024.git



git@github.com:rllab-snu/Intelligent-Systen

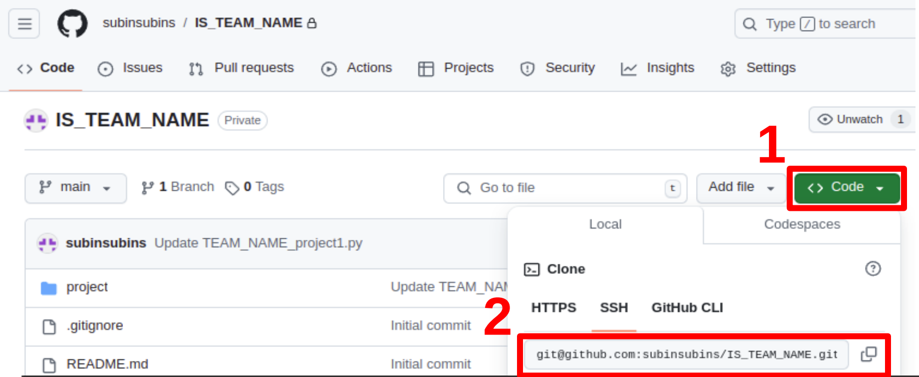
Use a password-protected SSH key.


subin@subin-desktop: ~/Intelligent-Systems-2024/src/rccar_bringup/rccar_bringup/project
subin@subin-desktop: ~/Intelligent-Systems-2024/src/rccar_bringup/rccar_bringup/project 117x30

```
(rccar) subin@subin-desktop:~$ git clone git@github.com:rllab-snu/Intelligent-Systems-2024.git
Cloning into 'Intelligent-Systems-2024'...
remote: Enumerating objects: 179, done.
remote: Counting objects: 100% (179/179), done.
remote: Compressing objects: 100% (171/171), done.
remote: Total 179 (delta 3), reused 179 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (179/179), 3.02 MiB | 2.95 MiB/s, done.
Resolving deltas: 100% (3/3), done.
(rccar) subin@subin-desktop:~$ cd Intelligent-Systems-2024/
(rccar) subin@subin-desktop:~/Intelligent-Systems-2024$ cd src/rccar_bringup/rccar_bringup/project
(rccar) subin@subin-desktop:~/Intelligent-Systems-2024/src/rccar_bringup/rccar_bringup/project$ ls
IS_RLLAB
```

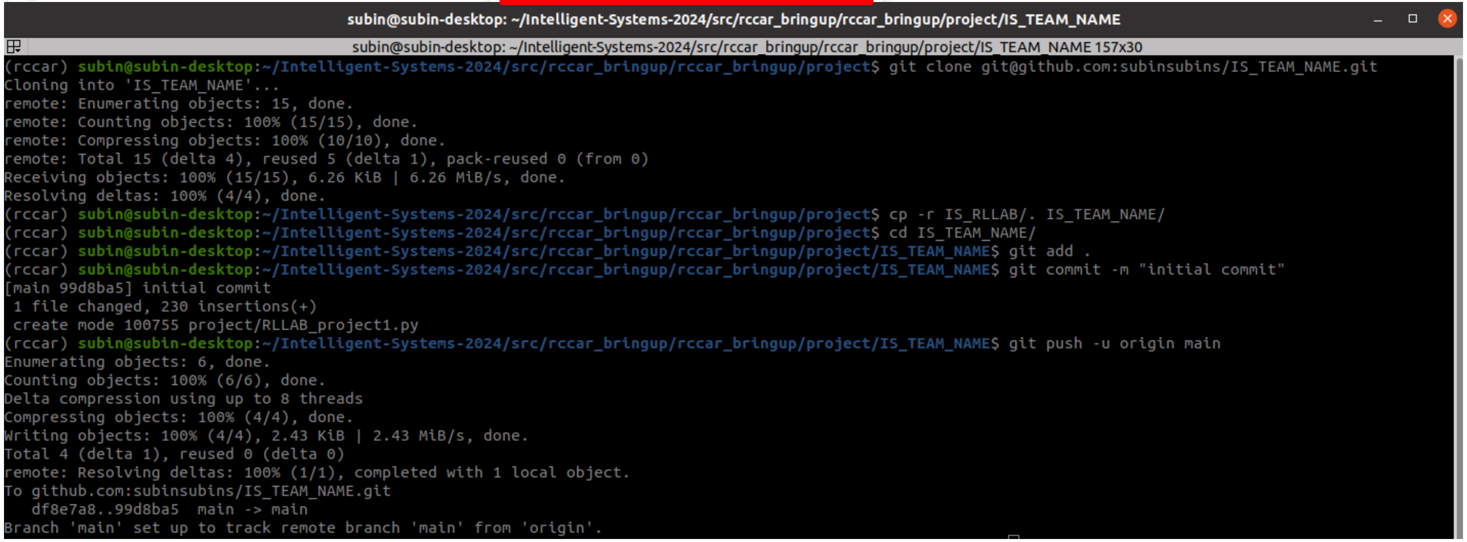
4) Clone your private IS_{TEAM_NAME} repo

git clone git@github.com:{username}/IS_{TEAM_NAME}.git



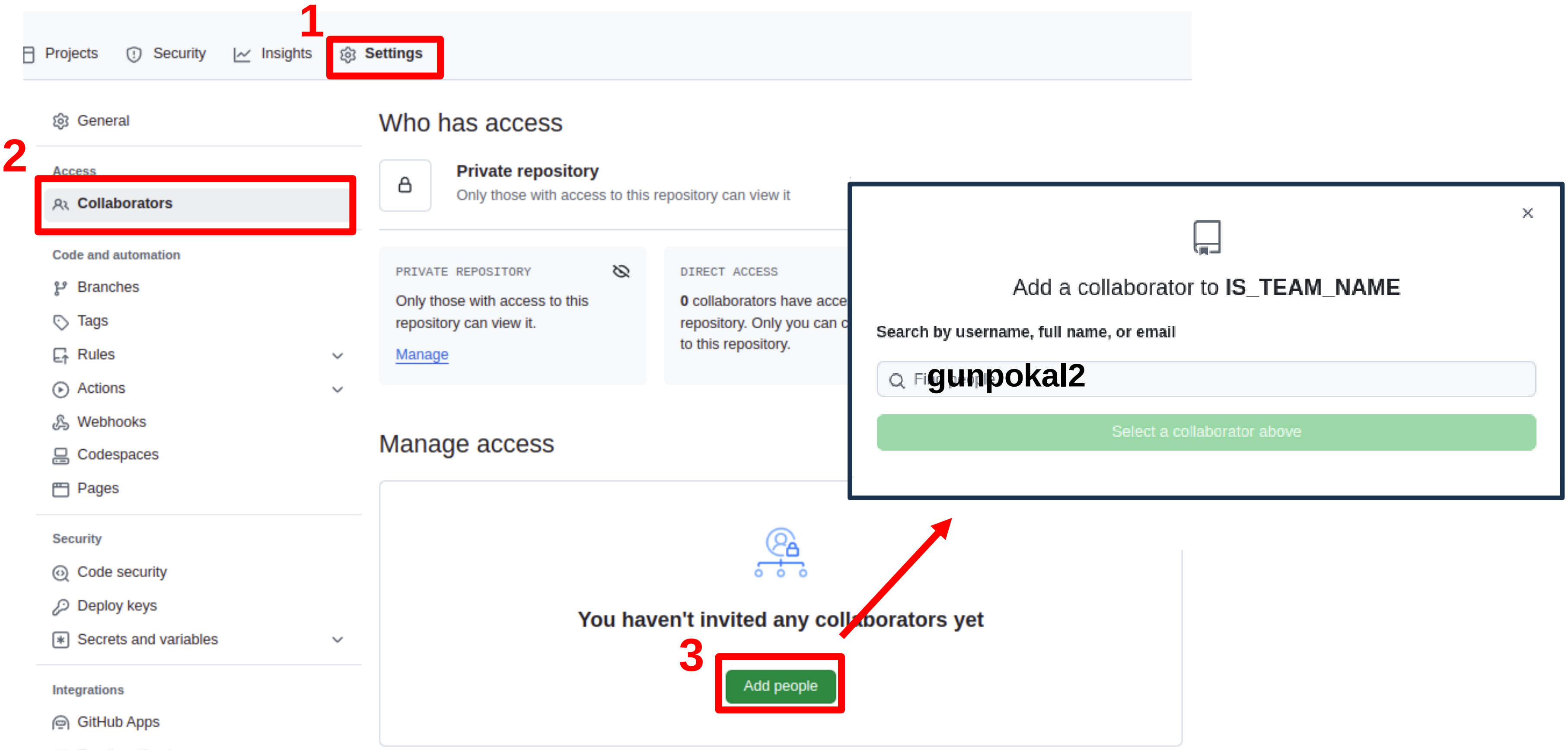
```
git@github.com:subinsubins/IS_TEAM_NAME.git
```

cp –r IS_RLLAB/. IS_TEAM_NAME/
cd IS_TEAM_NAME/
git add .
git commit –m “Initial Commit”
git push –u origin main


subin@subin-desktop: ~/Intelligent-Systems-2024/src/rccar_bringup/rccar_bringup/project/IS_TEAM_NAME
subin@subin-desktop: ~/Intelligent-Systems-2024/src/rccar_bringup/rccar_bringup/project/IS_TEAM_NAME 157x30

```
(rccar) subin@subin-desktop:~/Intelligent-Systems-2024/src/rccar_bringup/rccar_bringup/project$ git clone git@github.com:subinsubins/IS_TEAM_NAME.git
Cloning into 'IS_TEAM_NAME'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 15 (delta 4), reused 5 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 6.26 KiB | 6.26 MiB/s, done.
Resolving deltas: 100% (4/4), done.
(rccar) subin@subin-desktop:~/Intelligent-Systems-2024/src/rccar_bringup/rccar_bringup/project$ cp -r IS_RLLAB/. IS_TEAM_NAME/
(rccar) subin@subin-desktop:~/Intelligent-Systems-2024/src/rccar_bringup/rccar_bringup/project$ cd IS_TEAM_NAME/
(rccar) subin@subin-desktop:~/Intelligent-Systems-2024/src/rccar_bringup/rccar_bringup/project/IS_TEAM_NAME$ git add .
(rccar) subin@subin-desktop:~/Intelligent-Systems-2024/src/rccar_bringup/rccar_bringup/project/IS_TEAM_NAME$ git commit -m "initial commit"
[main 99d8ba5] initial commit
 1 file changed, 230 insertions(+)
 create mode 100755 project/RLLAB_project1.py
(rccar) subin@subin-desktop:~/Intelligent-Systems-2024/src/rccar_bringup/rccar_bringup/project/IS_TEAM_NAME$ git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 2.43 KiB | 2.43 MiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:subinsubins/IS_TEAM_NAME.git
   df8e7a8..99d8ba5  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

5) Add team members and TA(gunpokal2) as collaborators



1

Projects | Security | Insights | Settings

2

General

Access

Collaborators

Code and automation

- Branches
- Tags
- Rules
- Actions
- Webhooks
- Codespaces
- Pages

Security

- Code security
- Deploy keys
- Secrets and variables

Integrations

- GitHub Apps
- Email notifications

Who has access

Private repository
Only those with access to this repository can view it

PRIVATE REPOSITORY
Only those with access to this repository can view it.
Manage

DIRECT ACCESS
0 collaborators have acce
repository. Only you can c
to this repository.

Manage access

You haven't invited any collaborators yet

3

Add people

Add a collaborator to **IS_TEAM_NAME**

Search by username, full name, or email

Fi **gunpokal2**

Select a collaborator above

Git Tutorial - Summary

6) Modify your codes!

```
# Upload your local codes to remote repository
git add .
git commit –m “commit message”
git push –u origin main

# Download remote repository’s codes to local PC
git pull origin main

# make branch
git checkout –b <branchname>
git merge <branchname1> <branchname2>

# undo local changes
git fetch origin
git reset –-hard origin/main
```

For more git commands, go to this link
https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html

RLLAB
http://rllab.snu.ac.kr

Thank you