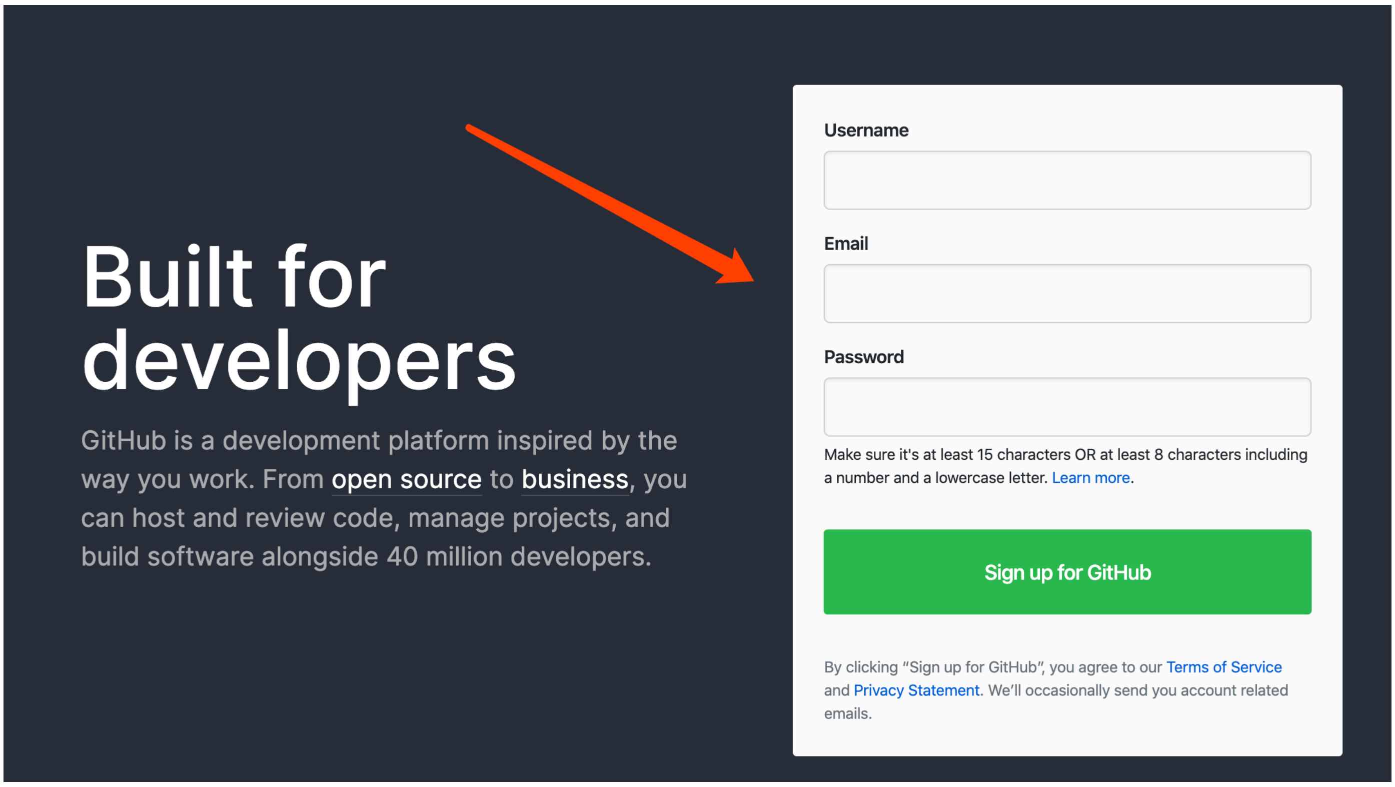
Setup Github

GitHub is an online platform for hosting and managing source code repositories, using Git as the version control software.

Create a GitHub account.

Go to the website of Github and *Sign Up*:



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 40 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account related emails.

Simple Git Configuration

> Note: From this step onward, for Windows users, it is recommended to right-click on the desktop and then select **Git Bash Here** from the menu. This is a command line provided by Git that you just downloaded. Linux and macOS users can use the terminal that comes with the system.

Configure username and password.

Open the command line and enter the following two commands (press Enter after each one, do not copy `$`):

```
$ git config --global user.name "your_username"
$ git config --global user.email "your@email.com"
```

Note: Replace `your_username` with the username of your newly created GitHub account, and replace `your@email.com` with the email you used for registration. (Keep the double quotes outside as they are.)

Configure SSH key and associate it with GitHub.

Generate ssh-key

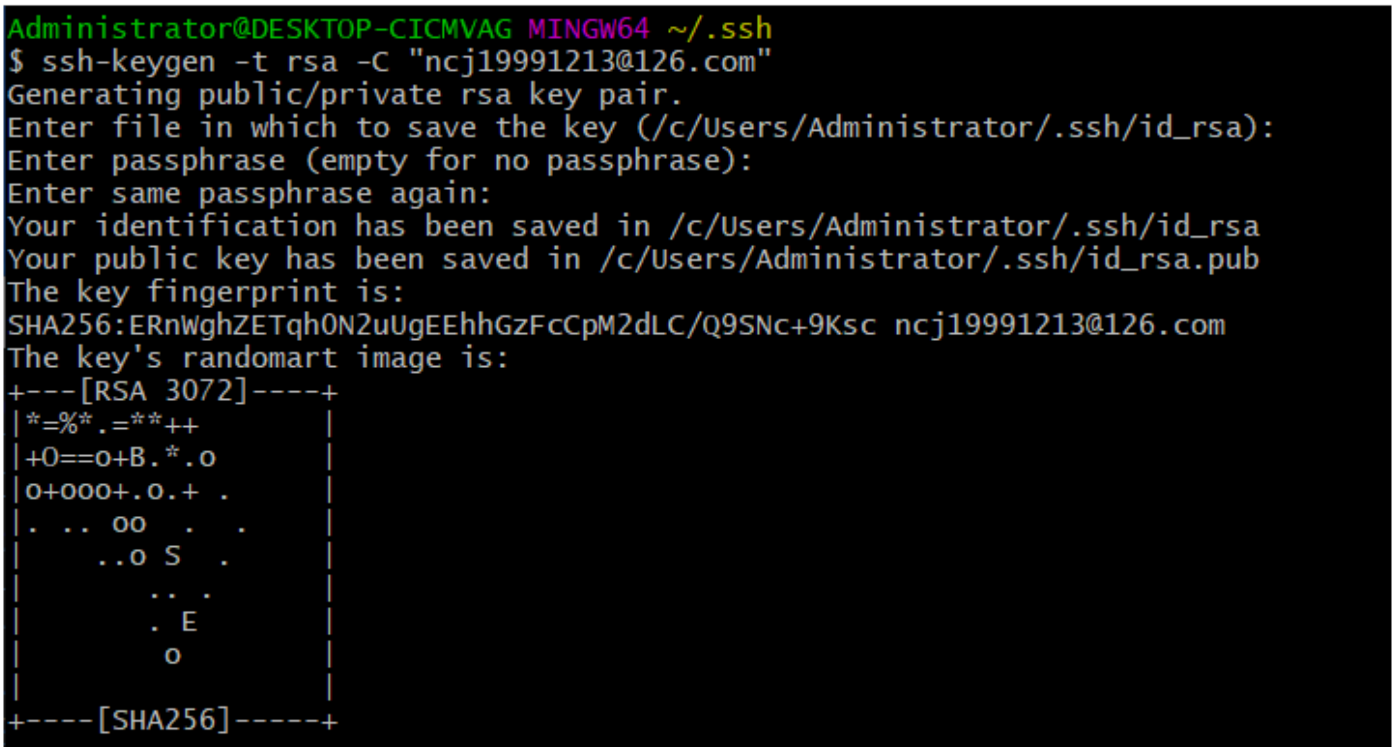
In simple terms, this step is to allow GitHub to authenticate your identity, enabling you to smoothly download and upload code. An SSH key is a type of password, usually stored in two files: a private key and a public key. The public key is shared with others for identity verification, while the private key must be securely kept and not shared with anyone.

Similarly, still within the command line, run the following(do not copy `$`):

```
$ ssh-keygen -t rsa -C "your@email.com"
```

Similarly, here you need to replace it with the email you used for registration. Then, you will be prompted to enter some information one by one. If you don't want to specify, you can simply press `enter` throughout. The meanings of these input parameters are as follows:

- Path to save the SSH key file: It defaults to ~/.ssh, and if you don't remember, it's better not to change it.
- Passphrase: This is a password, generally not widely used. It is only required in certain situations where your key needs to be accessed. Usually, you can leave it blank.
- Confirm Passphrase: If you set a passphrase in the previous step, enter it again for confirmation.



```
Administrator@DESKTOP-CICMVAG MINGW64 ~/.ssh
$ ssh-keygen -t rsa -C "ncj19991213@126.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Administrator/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Administrator/.ssh/id_rsa
Your public key has been saved in /c/Users/Administrator/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ERnWghZETqhON2uUgEEhhGzFcCpM2dLC/Q9SNc+9Ksc ncj19991213@126.com
The key's randomart image is:
+---[RSA 3072]----+
|*=%*.=**++       |
|+O==o+B.*.o      |
|o+ooo+.o.+ .     |
|. .. oo  .  .    |
|    ..o S  .     |
|       .. .      |
|       . E       |
|        o        |
|                 |
+----[SHA256]-----+
```

When you see a series of strange characters appear, it indicates that you have successfully created it.

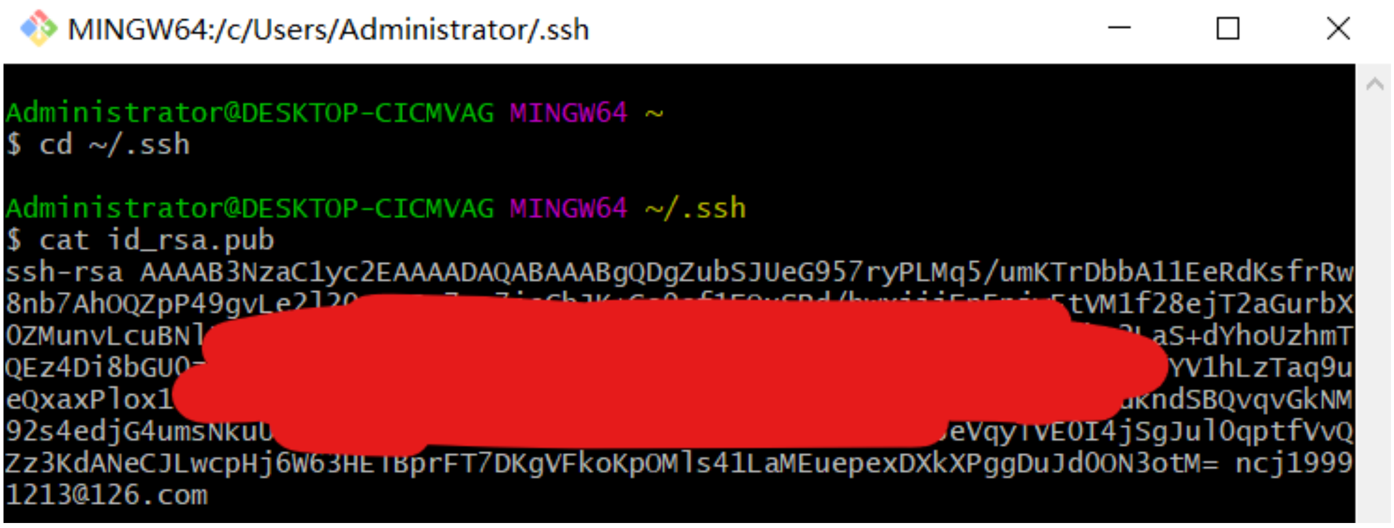
Upload ssh-key

Firstly, you need to locate the key you just created. Run the following two commands in sequence(do not copy `$`):

```
$ cd ~/.ssh
$ cat id_rsa.pub
```

The meanings are as follows:

- `cd ~/.ssh`: Enter a folder named `.ssh`, where `~` represents your user directory. (If you are using Windows and cmd, open a new cmd and type `cd .ssh`.)
- `cat id_rsa.pub`: Display the content of the file `id_rsa.pub`, which is your public key. (If you are using cmd, replace `cat` with `type`.)

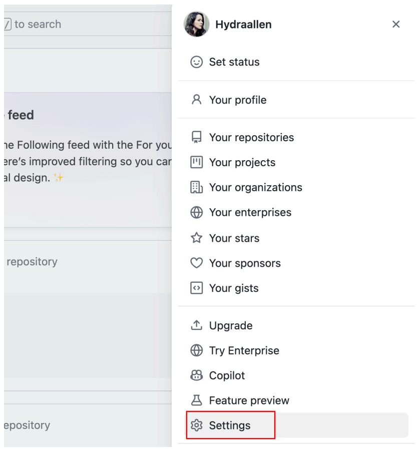

MINGW64:/c/Users/Administrator/.ssh

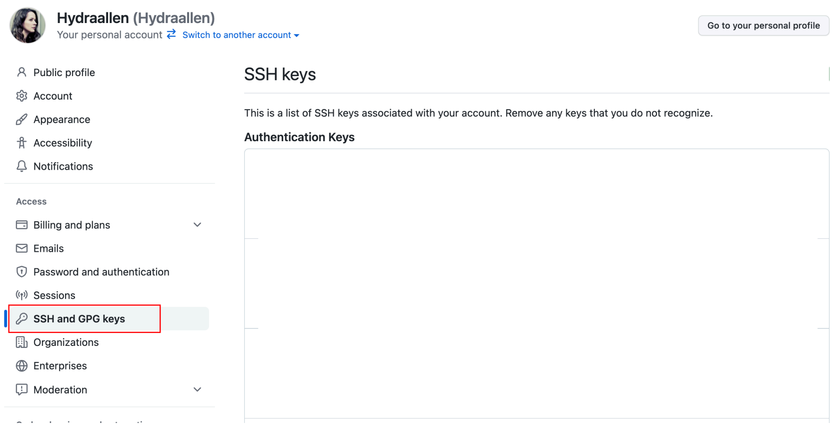
```
Administrator@DESKTOP-CICMVAG MINGW64 ~
$ cd ~/.ssh

Administrator@DESKTOP-CICMVAG MINGW64 ~/.ssh
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDgZubSJUeG957ryPLMq5/umKTrDbbA11EeRdKsfrRw
8nb7AhOQZpP49gvLe2l2[redacted]tVM1f28ejT2aGurbX
0ZMunvLcuBNl                                                    aS+dYhoUzhmT
QEz4Di8bGU0                                                      YV1hLzTaq9u
eQxaxPlox1                                                     KndSBQvqvGkNM
92s4edjG4umsNkuU                                       eVqyTVEOI4jSgJul0qptfVvQ
Zz3KdANeCJLwcpHj6W63HETBprFT7DKgVFkoKpOMls41LaMEuepexDXkXPggDuJd0ON3otM= ncj1999
1213@126.com
```

Then, log in to GitHub, go to your account settings in the upper right corner, and navigate to SSH and GPG keys. Choose to add a new SSH key. You can input any title you like. In the input box below, paste the content of the public key that you just printed in the command line (from "ssh-rsa" to the end of your email). Finally, click the bottom button to complete the addition.

> Note: Do not copy your private key (`id_rsa`). The private key is much longer than the public key.




Hydraallen (Hydraallen)
Your personal account ⇄ Switch to another account ▾
Go to your personal profile

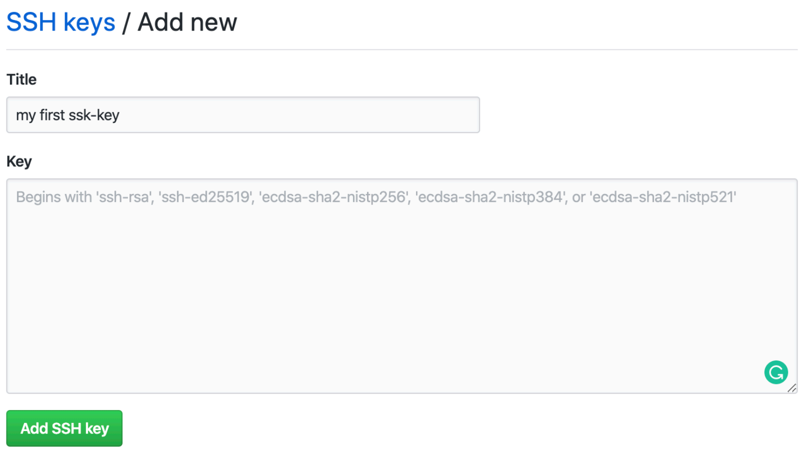
Public profile
Account
Appearance
Accessibility
Notifications

Access
Billing and plans
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations
Enterprises
Moderation

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication Keys


SSH keys / Add new

Title

my first ssk-key

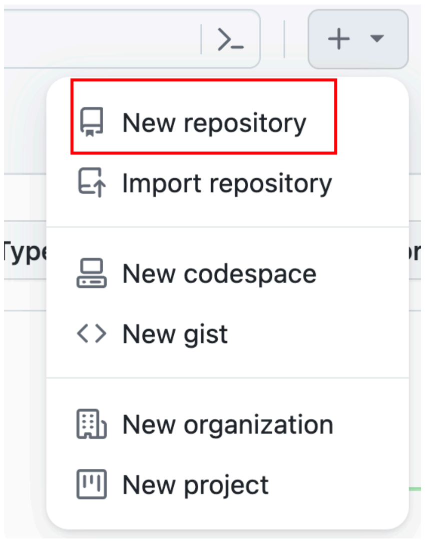
Key

Begins with 'ssh-rsa', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

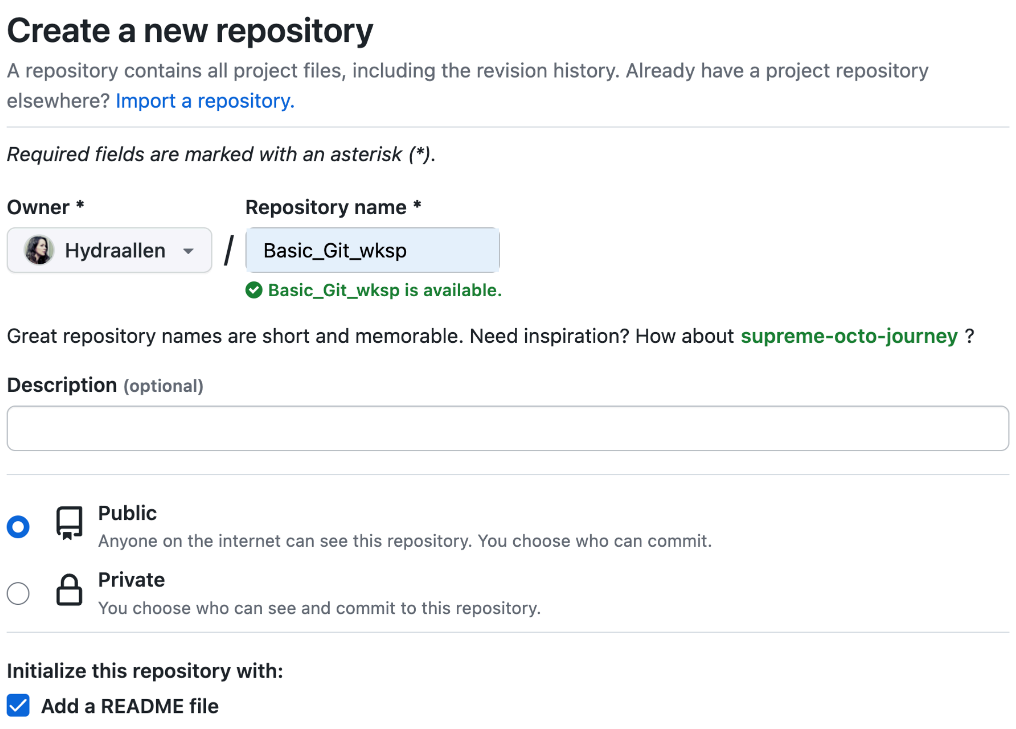
Add SSH key

Verify

Let's quickly validate. Still at the top right corner of the page, click on the plus sign, select "New Repository," and create a new code repository.



`Repository name` is the name of your project and cannot contain spaces. `Description` is a brief description of your project. `Public` means anyone can see this project, while `Private` means only you can see it. Finally, it is recommended to select "Initialize this repository with a README" so that a README file is automatically included when the project is created.


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 (*).

Owner * : Hydraallen / Repository name * : Basic_Git_wksp

Basic_Git_wksp is available.

Great repository names are short and memorable. Need inspiration? How about supreme-octo-journey ?

Description (optional)

[✓ Public] Anyone on the internet can see this repository. You choose who can commit.

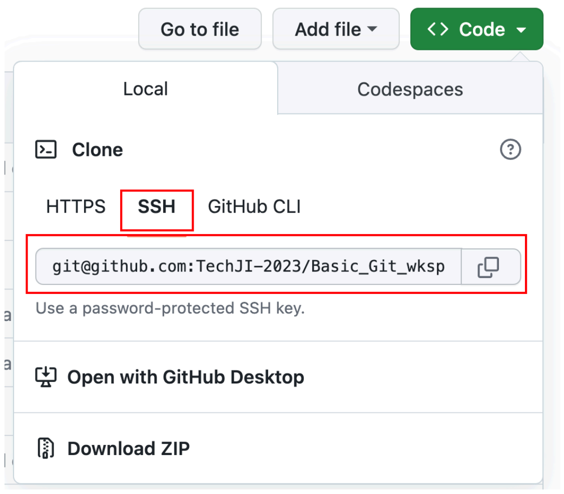
Private You choose who can see and commit to this repository.

Initialize this repository with:

[✓ Add a README file]

Then you will be redirected to the homepage of your new project. Locate the only green button on the page, click on it, and then click on the button in the image to copy your project's address to the clipboard.

> Here, try to copy the address starting with 'git@' if the default one displayed starts with 'https.' If your default address is in the form of 'https,' click on "use SSH" to view the 'git@' address.



git@github.com:TechJI-2023/Basic_Git_wksp

Use a password-protected SSH key.

Finally, return to your command line and execute the following commands one by one (do not copy the part after the # symbol and do not copy `$`):

```
$ cd ~/Desktop                        # 进入桌面
$ git clone <your_repo_address>       # clone你刚刚创建的项目
```

Note: Replace `<your_repo_address>` with the project address you just copied, **without using angle brackets or double quotes**.

You may encounter a prompt asking if you want to continue the connection. Type `yes` and press Enter.

After a while, when it finishes running, you will find a folder on your desktop named after your project, containing the content of the project you just created. At this point, it indicates that you have successfully installed and configured Git.

Now, **you can use the command line to operate Git and perform all version control functions**.

Install Git GUI（Optinal）

In simple terms, Git GUI is a type of desktop software that **saves you from the pain of typing command lines**. Its relationship with Git is just icing on the cake—without GUI software, you can still perform all operations using Git commands in the command line. Of course, GUI software adds some practical tools to the mix.

> GUI software is just a shell; all the operations it performs are still silently invoking the `git` commands in the background.

Suggestions：

- Lazygit (Windows, Mac, Linux) -> You may refer to `Installation_Lazygit.md` in this repo.
- Sourcetree (Windows, Mac)
- GitKraken (Windows, Mac, Linux)

These are generally easy to operate, relatively stable, and, moreover, all of them are free.