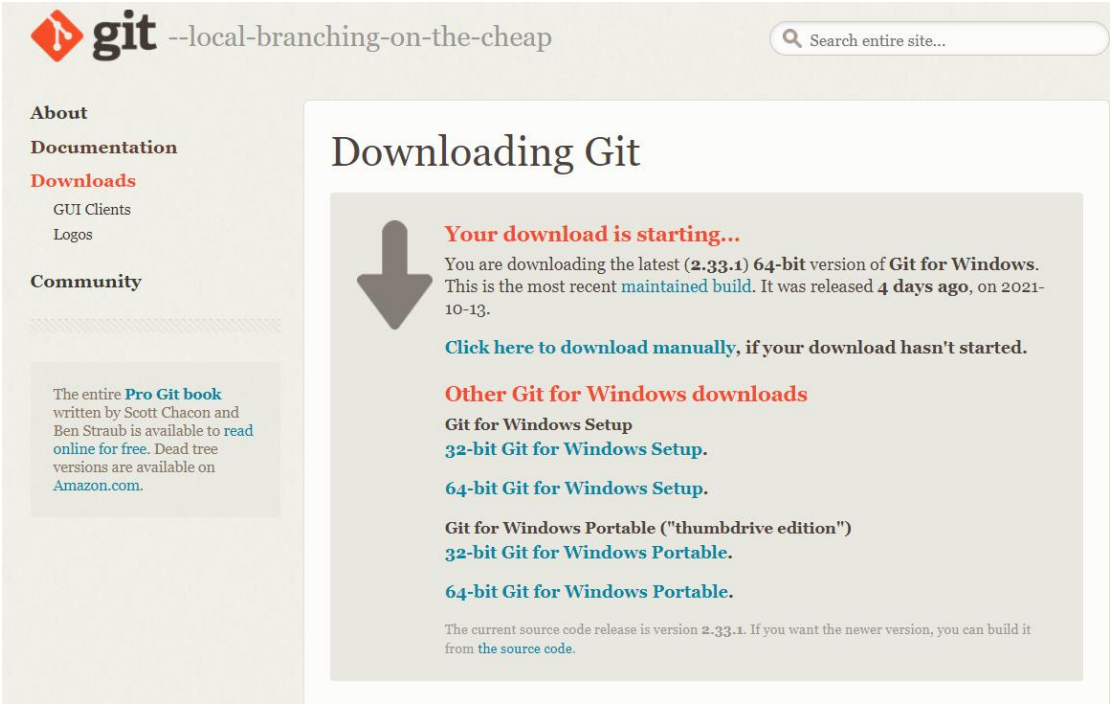
Tutorial of Git Bash for Windows

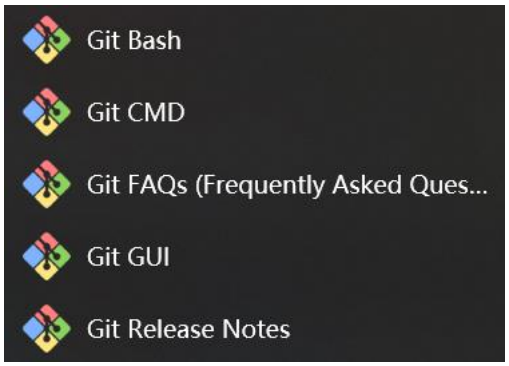
Step 1: Download and Install Git Bash for Windows

Visit the official website https://git-scm.com/download/win



Download and install Git Bash.

Once complete the installing, following will occur in Git folder

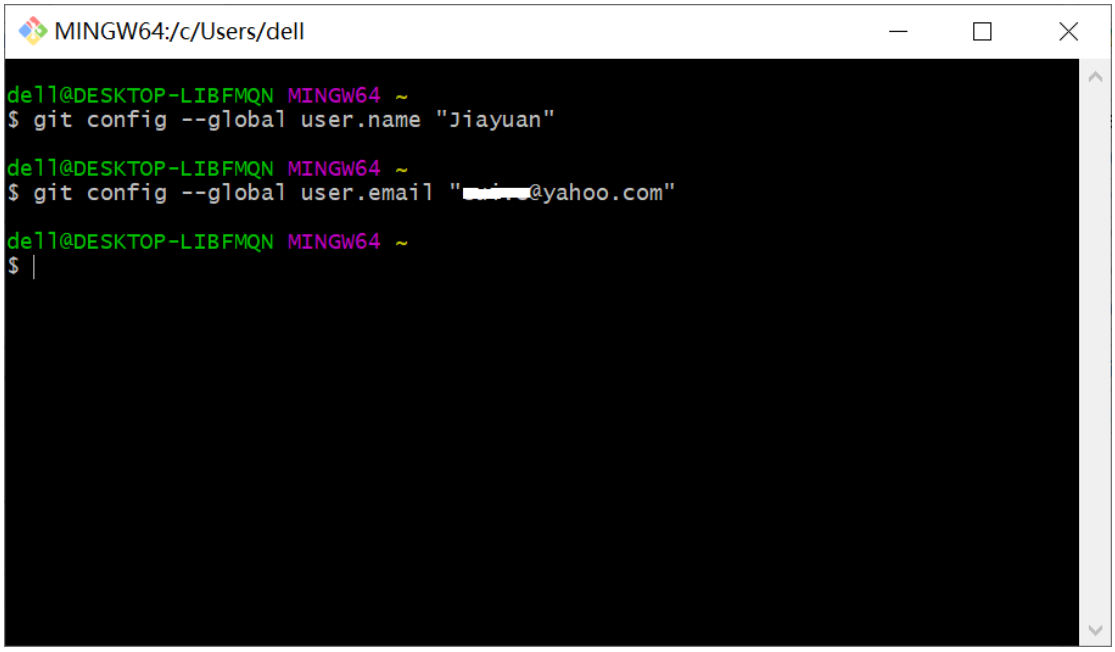


Step 2: User Setup

Open Git Bash and set your username and email address using following command lines:

```
$ git config --global user.name "Your Name"

$ git config --global user.email "Your email address"
```



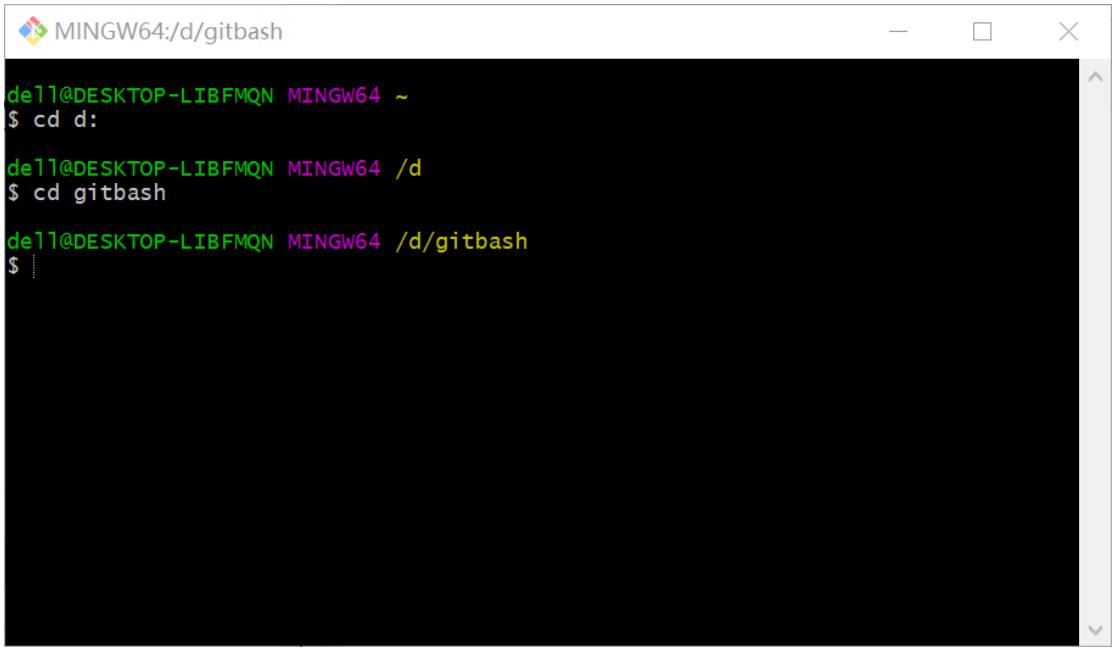
This is used to configure user information used across all local repositories. When you try to push files, this will tell your Git repository who the user is.

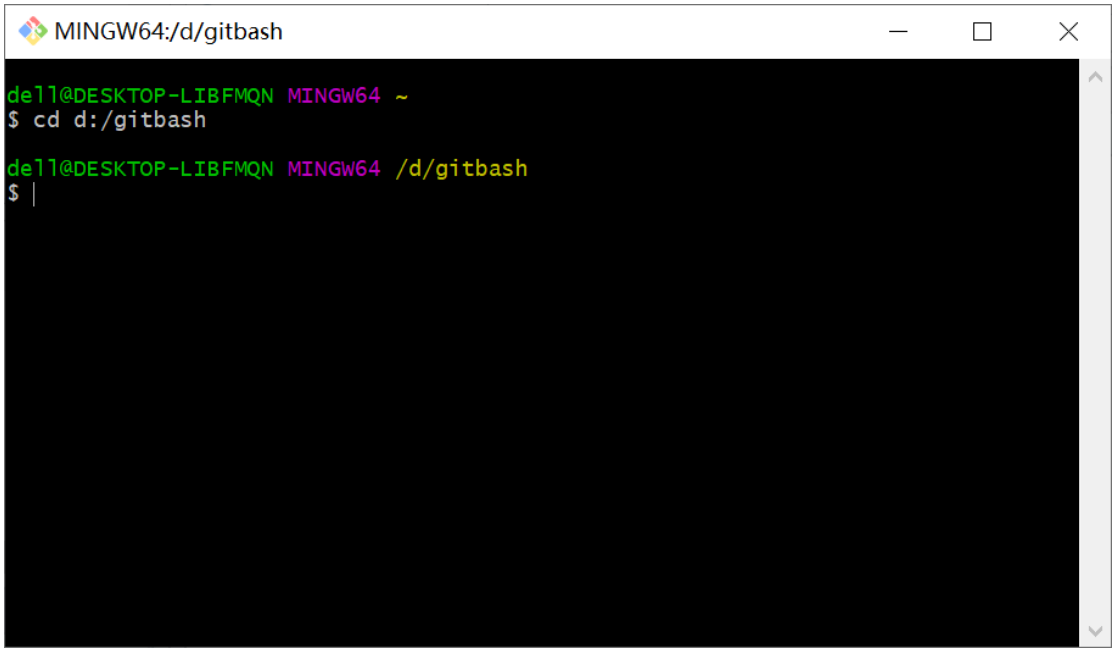
Step 3: Manage Local Repository

1. Navigate to a directory

Open a directory where you keep your files, for example, D:\gitbash. Right click on the folder and click on Git Bash Here.

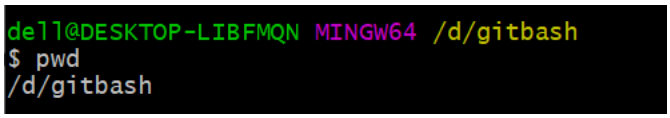
cd command refers to change directory and use command line $ cd folder_path to get into the directory.





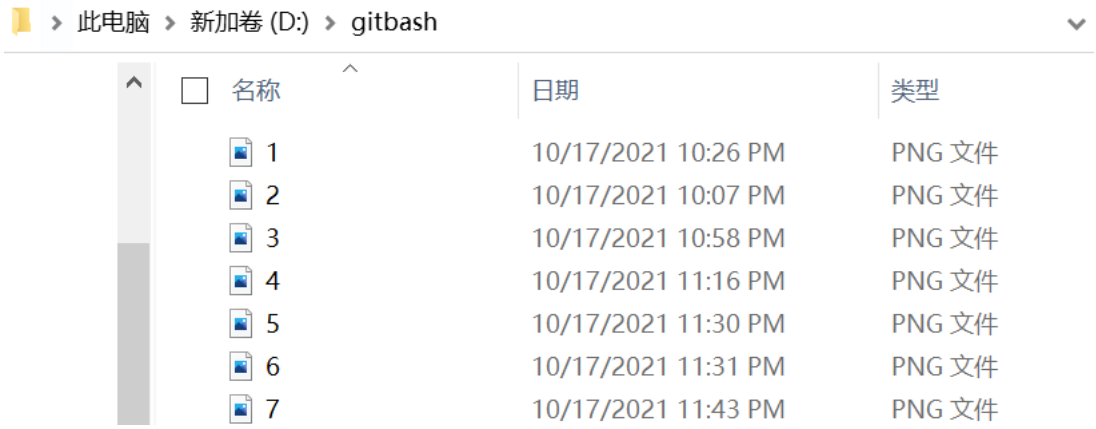
2. Check files

Yellow line indicates your present working directory. Use **pwd command** to print and check the present working directory.



Use **ls command** to list all the files and folders in the present directory.

```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash
$ ls
1.PNG  2.png  3.PNG  4.png  5.PNG  6.PNG  7.PNG
```



3. Exit a folder

When you get into a folder by mistake, use **cd ..**(space between 'cd' and '..') **command** to return to the previous folder.

```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash
$ cd ..

dell@DESKTOP-LIBFMQN MINGW64 /d
$
```

4. Create and delete folders and files

mkdir command is used to create a new directory and use command line $mkdir new_folder_name to create a new directory under the current directory.

```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash
$ mkdir 5702
```

touch command is used to create a new file and use command line $touch new_file_name.file_type to create a new file.


```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash
$ touch 'Community Contribution'.txt
```

rm command is used to delete a file and use command line $rm file_name.file_type to delete a file.

```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash
$ rm 'Community Contribution'.txt
```

Use command line $rm -r file_name to delete a folder. Files within a directory can only be deleted from outside the directory. For example, I want to delete 5702 folder, I have to go back to gitbash folder.

```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash/5702
$ cd ..

dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash
$ rm -r 5702
```

Step 4: Manage Git Repository

1. Initialize the local directory as a Git repository

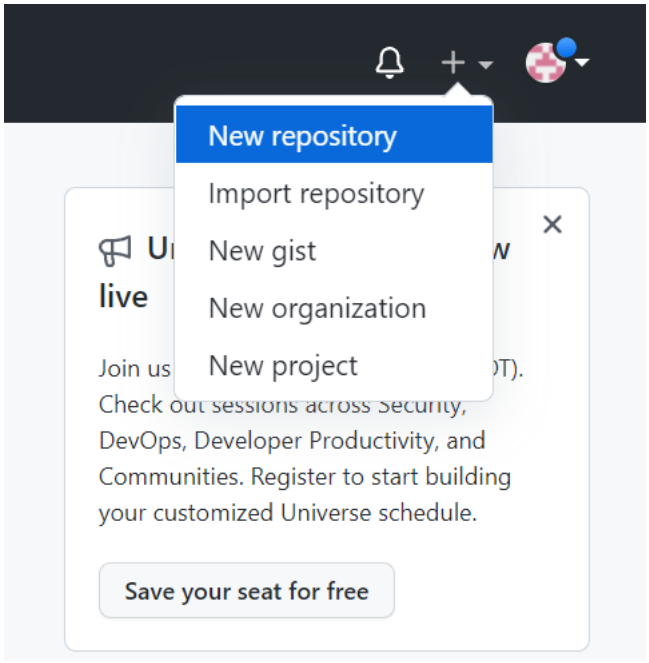
```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash
$ git init
Initialized empty Git repository in D:/gitbash/.git/
```

This creates a .git directory.

2. Create a repository on GitHub

Open GitHub https://github.com/ and create an account.

Click on the "+" button on the upper-right corner, then click on "New repository" and create a repository.



Create a new repository

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

Owner * JiayuanCui / **Repository name** * repository

Great repository names are short and memorable. Need inspiration? How about ideal-guacamole?

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:

Skip this step if you're importing an existing repository.

- **Add a README file** — This is where you can write a long description for your project. Learn more.
- **Add .gitignore** — Choose which files not to track from a list of templates. Learn more.
- **Choose a license** — A license tells others what they can and can't do with your code. Learn more.

Create repository

*Public repositories are accessible to everyone on the internet. Private repositories are

only accessible to you.

3. Connect the local repository with GitHub remote repository

There are several ways to clone repositories available on GitHub.com, the URL could be your repository on GitHub. The first type is HTTPS URL, another is SSH URL.

3.1 Configure Secure Shell (SSH)

Many Git servers authenticate using SSH public keys. Each user must generate a public key if they don't have one to provide a public key to the servers.

1) Check if you have a key using command `~/.ssh` or command `~/.ssh ls`.

If users have a key, it will show "bash: /c/Users/.../.ssh: Is a directory"

If users don't have a key, it will show "bash: /c/Users/.../.ssh: No such file or directory"

2) Generate a SSH Key

If you already have a key, skip this step.

In Git Bash, type command line `$ ssh-keygen -t rsa -C "your_email"`

Following will occur:

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/.../.ssh/id_rsa):

Press Enter, following will occur:

Create directory '/c/Users/.../.ssh/id_rsa'

Enter passphrase (empty for no passphrase):

This is asking you to set a password. If you set one, once you push or clone files using SSH, you must enter the password.

Enter same passphrase again:

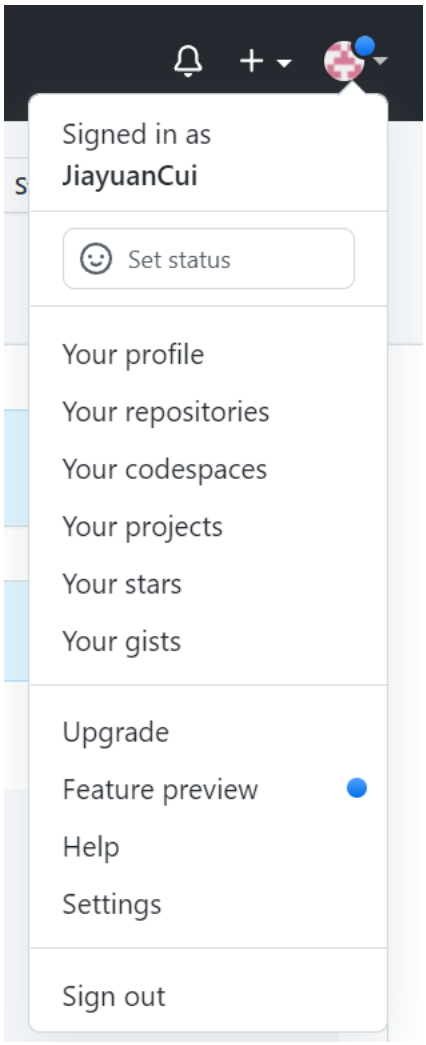
Then the following will occur and the SSH Key is generated successfully:

Your identification has been saved in /c/Users/.../.ssh/id_rsa

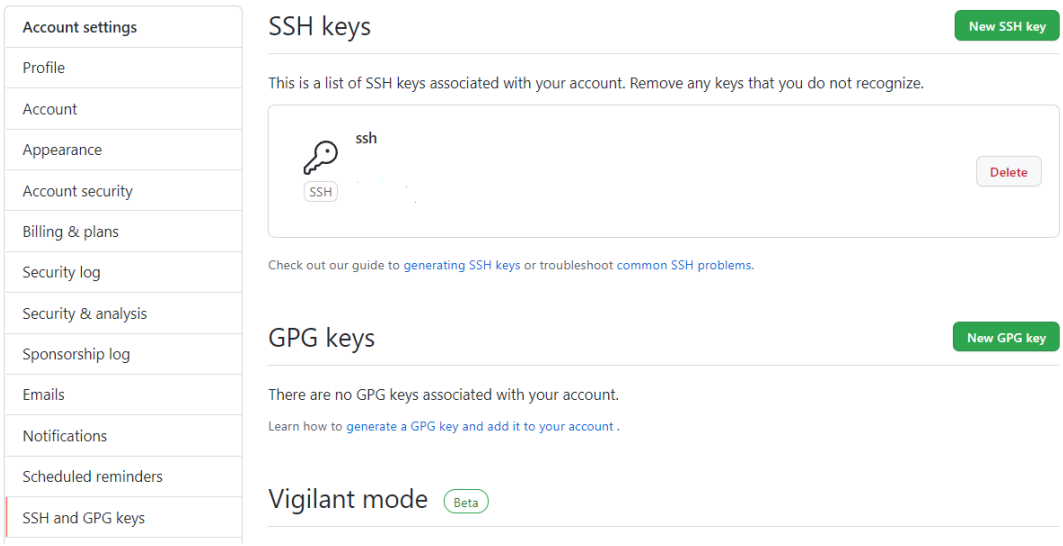
Your public key has been saved in /c/Users/.../.ssh/id_rsa.pub

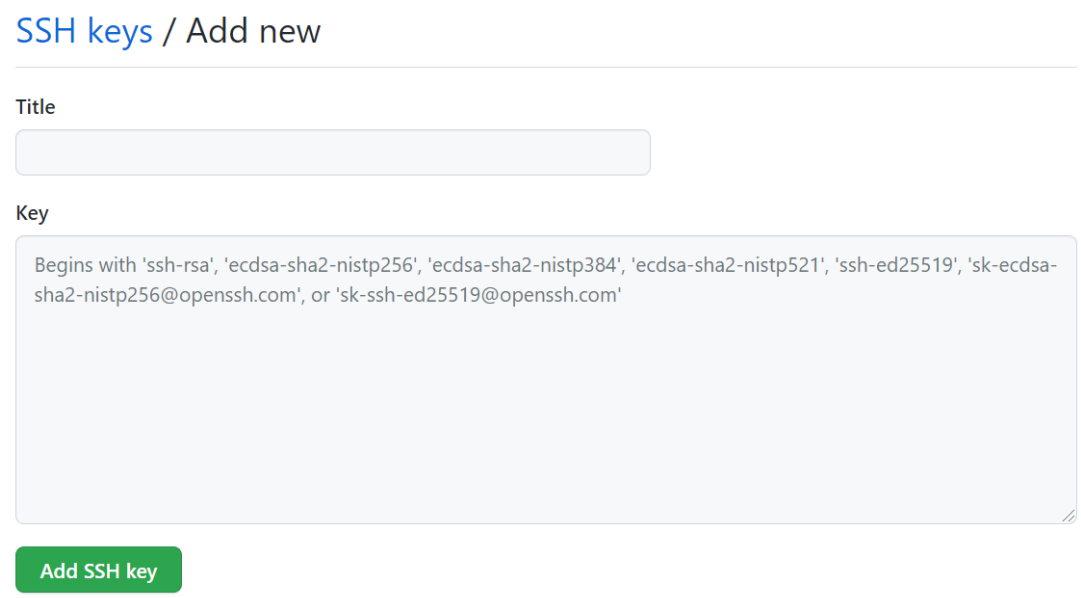
3) Add SSH Key on GibHub

Click on Settings

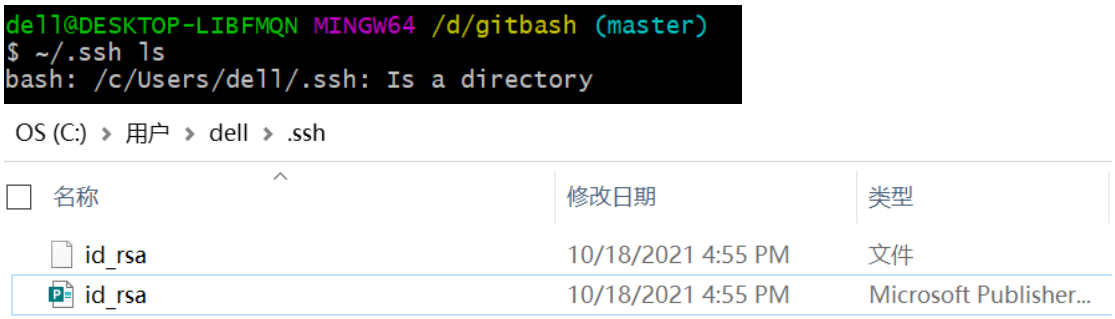


Choose SSH and GPG keys under Personal settings and click on the "New SSH key" button.





Use `~/.ssh ls` to check your key and use this path to find the file id_rsa.pub. Use Notepad to open it. Copy and paste the content in the "Key" text box.



After setting up, you will receive an Email from GitHub saying a public key was added to your account.

4) Test the SSH Key

Type command line `$ ssh -T git@github.com`, the following will occur:

> The authenticity of host 'github.com(13.229.188.59)' can't be established. RSA key fingerprint is SHA256:nThbg6kXUp...
>
> Are you sure you want to continue connecting(yes/no/[fingerprint])?

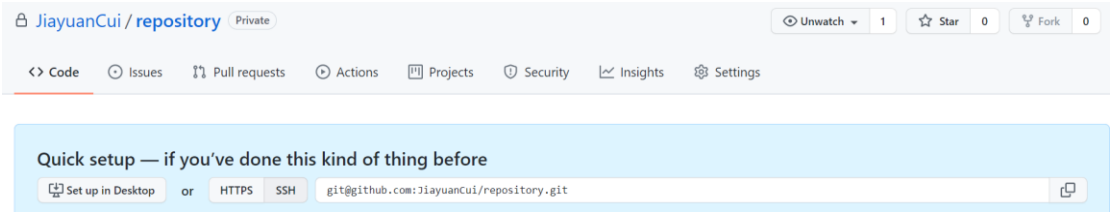
Type “yes”, enter. Then it will ask you to enter the password. If you didn’t set a password, the following will occur:

> Warning: Permanently added ‘github.com, 192.30.255.112’ (RSA) to the list of known hosts.

Finally, the following line will show which means you successfully setup a SSH key:

Hi “your_username”! You’ve successfully authenticated, but CitHub does not provide shell access.

Then you could find your HTTPS and SSH URL on GitHub.



3.2 Add a remote named for the repository at URL

Use command line $ git remote add “name” “URL”.

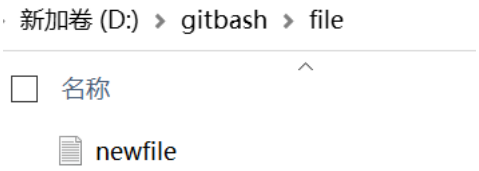
```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash (master)
$ git remote add origin git@github.com:JiayuanCui/repository.git
```

I added a new remote Git repository with the “origin”.

If you want to remove a remote, use the command line $ git remote remove “name”.

4. Upload files

Upload a file to GitHub, for example, “newfile.txt”. Create a new file called “newfile.txt” in the “file” folder.



4.1 add command to add any new or modified files to the index

Before run the commit command, you must run add command to add file contents to the index.

Command line $ git add -A updates all files in the entire working tree.

```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash/file (master)
$ git add -A
```

4.2 commit command to record changes to the repository

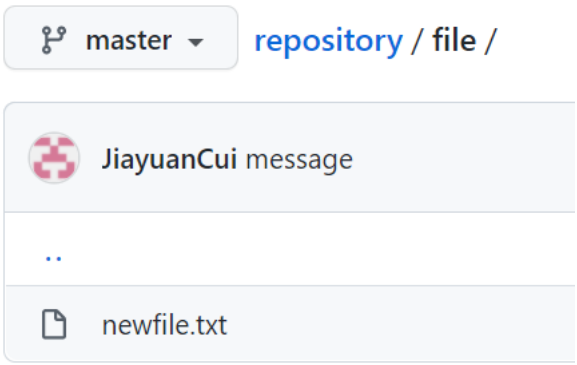
Use the command line $ git commit -m “message”. “message” adds the commit message.

```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash/file (master)
$ git commit -m "msg"
[master 4dd258d] msg
 1 file changed, 0 insertions(+), 0 deletions(-)
```

4.3 push command

Use the command line $ git push -u “set” “upstream” to upload local repository content to a remote repository.

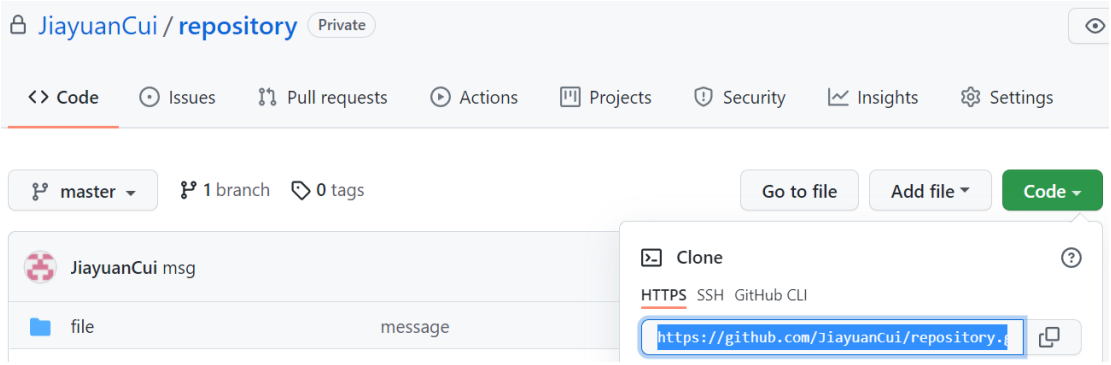
```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash/file (master)
$ git push -u origin master
Enumerating objects: 46, done.
Counting objects: 100% (46/46), done.
Delta compression using up to 4 threads
Compressing objects: 100% (43/43), done.
Writing objects: 100% (46/46), 816.07 KiB | 2.67 MiB/s, done.
Total 46 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), done.
To github.com:JiayuanCui/repository.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```



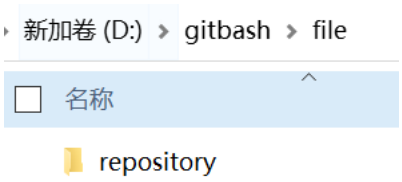
The newfile.txt is uploaded successfully.

5. Clone

Use command line $ git clone “HTTPS” or “SSH” to clone a repository into a new directory.



```
dell@DESKTOP-LIBFMQN MINGW64 /d/gitbash/file (master)
$ git clone https://github.com/JiayuanCui/repository.git
Cloning into 'repository'...
remote: Enumerating objects: 46, done.
remote: Counting objects: 100% (46/46), done.
remote: Compressing objects: 100% (38/38), done.
```



Successfully cloned repository “repository” in the local folder.

$ git pull command line fetch and download content from a remote repository and immediately update the local repository to match that content. If there are new updates in the remote repository, use the git pull command.