

# Authentication-GithUb

Hamid Semiyari

## Authentication

- To integrate GitHub (the website) with git (the command-line program), you have to set up the proper authorization.
- If you can already integrate with GitHub through the command line then you can skip the rest of these notes. If you are not sure, then please continue.
- GitHub now requires you to use either [Personal Access Tokens](#) or [SSH Keys](#) for interfacing. For this class, we'll use SSH Keys since its implementation is slightly easier.
- An alternative setup using personal access tokens can be found [here](#).
- SSH uses what's called a [key pair](#) where GitHub has a “public key” (a very long publicly known password) and you have on your computer a “private key” (a very long privately known password).
- Think about the public key as a padlock, and the private key as the key. You have to use the private key to unlock the public key.
- You need to generate a new private key, and go through the below steps, for each computer for which you intend to use git.

## Generate Key Pair on R Studio

- R Studio makes it easy to generate a key pair.
- Check to see if you already have an SSH key pair by running the following in R

```
file.exists("~/ssh/id_rsa.pub")
```

It will return FALSE if you do not have a key pair.

- Open up Tools > Global Options... > Git/SVN.
- If you do not have an SSH key pair, then click on “Create RSA(SS Key...”

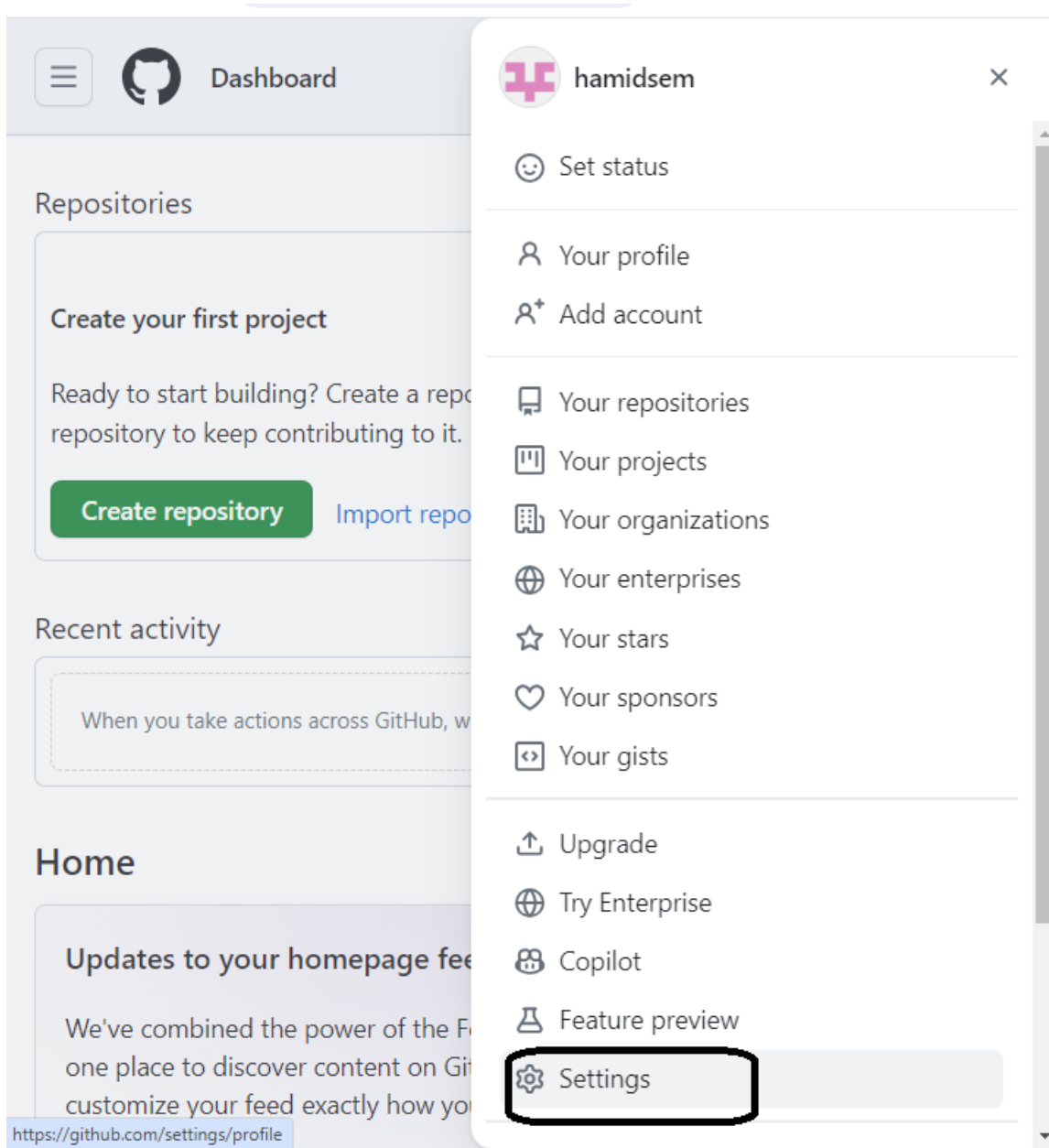
You may receive a note indicating similar to

Generating public/private ed25519 key pair. Your identification has been saved in C:/Users/semiyari/.ssh/id\_ed25519. Your public key has been saved in C:/Users/semiyari/.ssh/id\_ed25519.pub

- Click on “View public key”
- Copy the entire text that shows up. This is your public key.

### **Add Public Key to GitHub**

1. On GitHub, in the upper right corner, click on your profile photo and click on “Settings”



2.

On the left sidebar, click on “SSH and GPG keys”



**hamidsem (hamidsem)**  
Your personal 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

Code, planning, and automation

Repositories

Codespaces

Packages

Copilot

## Public profile

### Name

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

### Public email

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

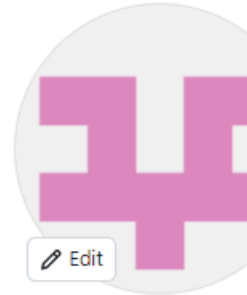
### Bio

You can @mention other users and organizations to link to them.

### Pronouns

### URL

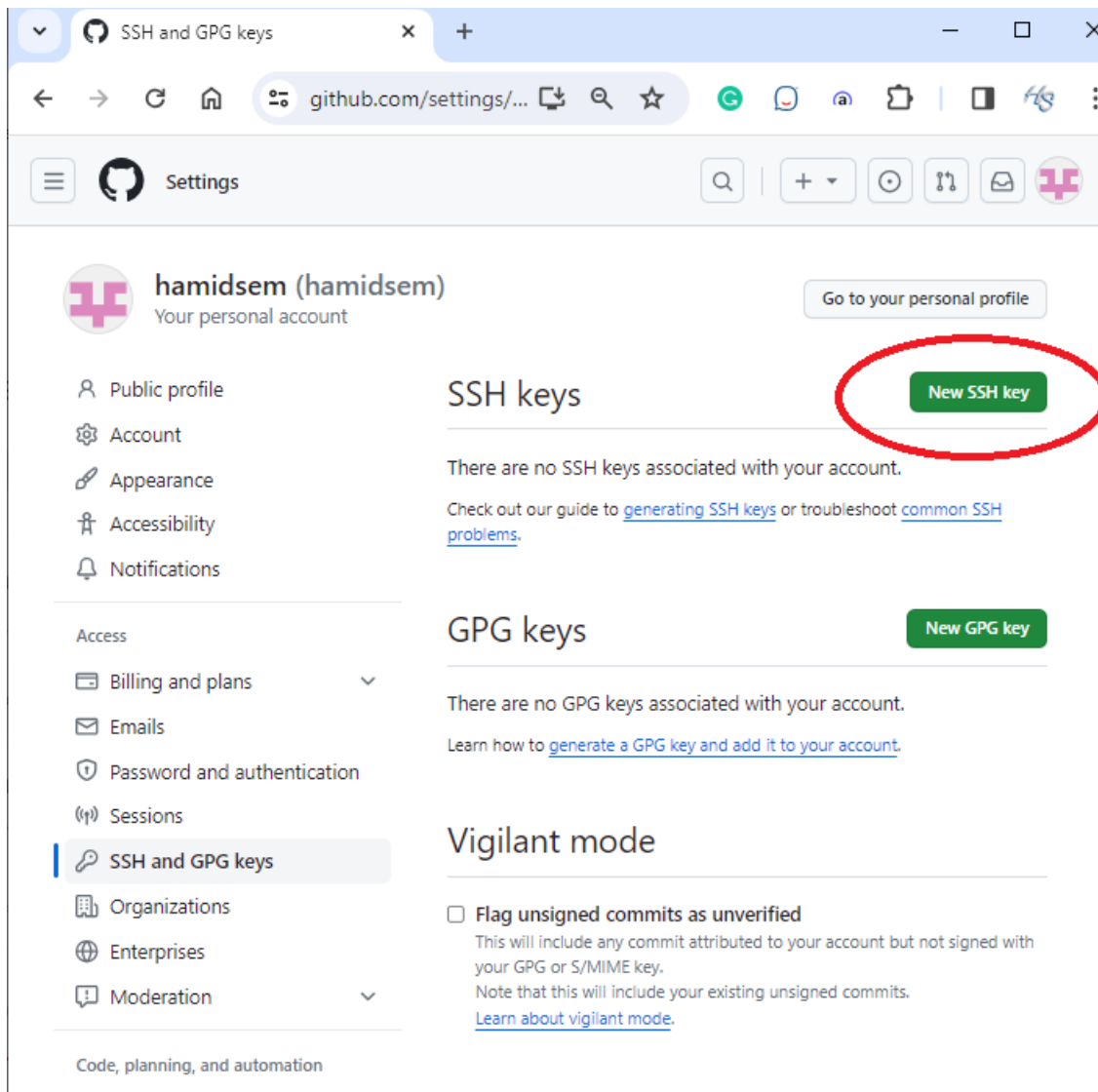
### Profile picture



Edit

3.

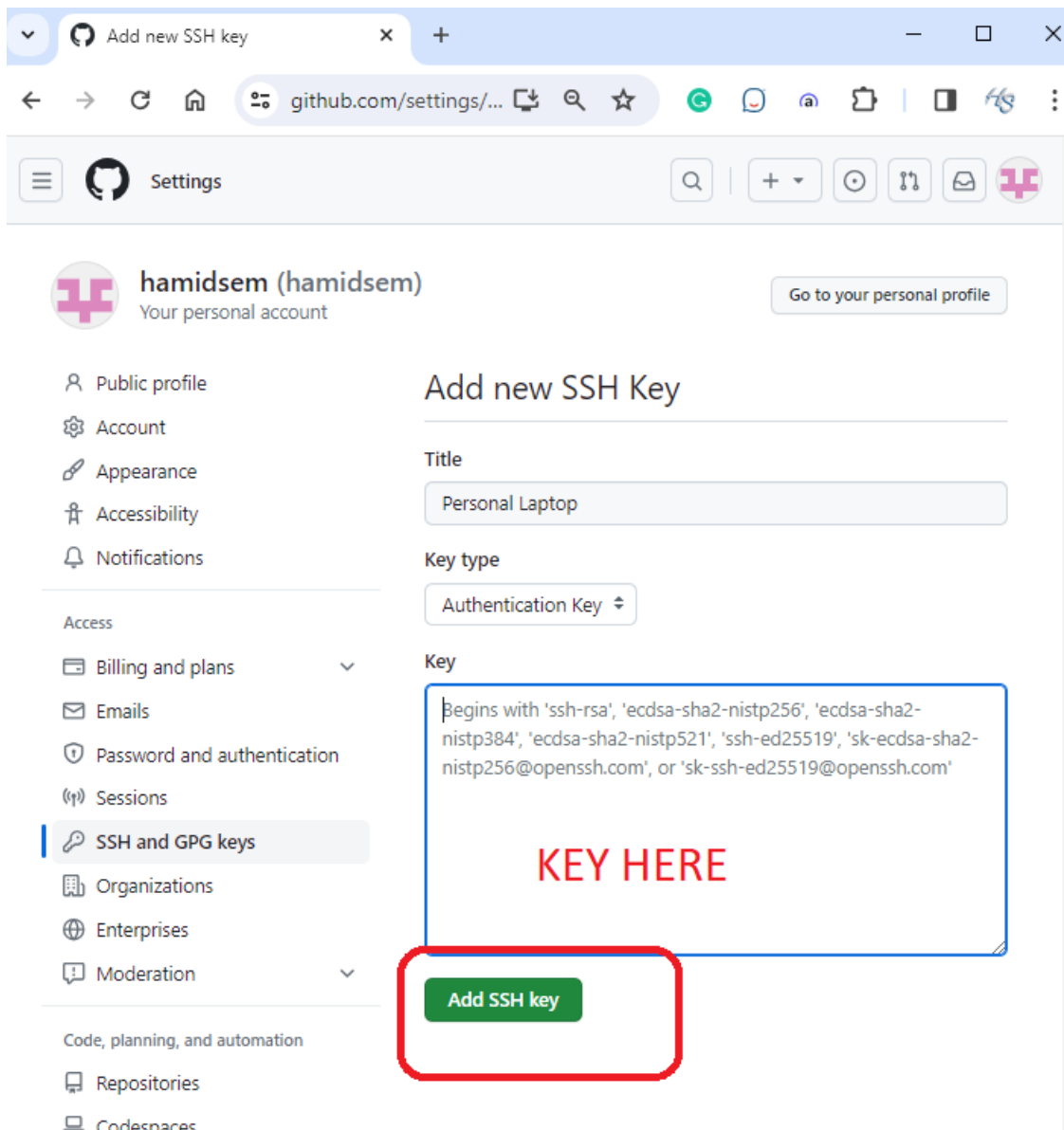
Click on "New SSH key"



4.

In the title field, choose a descriptive title, like “Personal Laptop”.

5. Paste your key into the “Key” field.




6. Click “Add SSH key” and confirm your GitHub password.

## SSH keys

[New SSH key](#)

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

### Authentication keys

  
SSH

**Personal Laptop**  
SHA256:bbNZxTjGypW3D6CM+A/NeakINF6SSVv4zbReyXy713s  
Added on Jan 16, 2024  
Never used — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

### Test to see if it worked

1. Enter the following in the terminal:

```
ssh -T git@github.com
```

2. Type “yes” if prompted to continue connecting.
3. You are successful if you see something like

```
Hi hamidsem! You've successfully authenticated, but GitHub does not provide shell access.
```

### Generate Key Pair on the Terminal

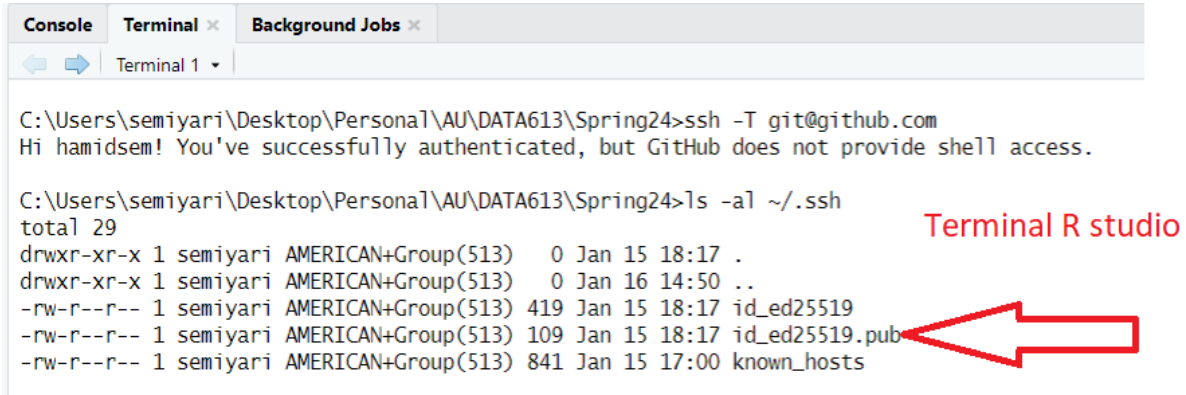
- If the R Studio pipeline does not work, then try generating your key pair on the terminal with the following instructions.

### Check for existing SSH Keys

- In the terminal run

```
ls -al ~/.ssh
```

- This will say something like “ls: cannot access ‘/c/Users/Vlad Dracula/.ssh’: No such file or directory” if you don’t have any public/private key pairs.
- This will list out files names like “id\_rsa.pub”, “id\_ecdsa.pub”, or “id\_ed25519.pub” if you do have a public/private key pair.



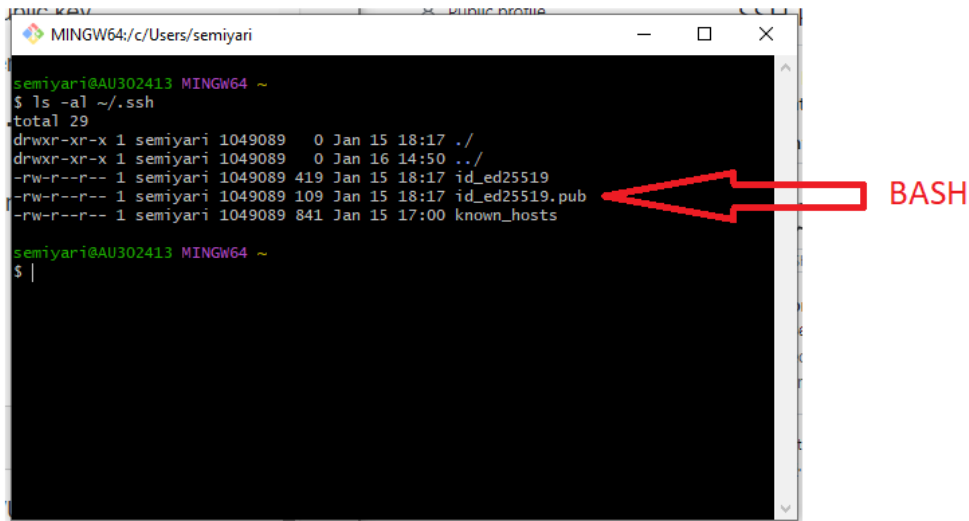
The screenshot shows the RStudio interface with the Terminal pane active. The terminal output shows a successful SSH authentication to github.com, followed by a command to list the contents of the .ssh directory. The output lists three files: id\_ed25519, id\_ed25519.pub, and known\_hosts. A red arrow points from the text 'Terminal R studio' to the terminal window, and another red arrow points from the text 'Terminal R studio' to the file 'id\_ed25519.pub' in the output.

```

C:\Users\semyari\Desktop\Personal\AU\DATA613\Spring24>ssh -T git@github.com
Hi hamidsem! You've successfully authenticated, but GitHub does not provide shell access.

C:\Users\semyari\Desktop\Personal\AU\DATA613\Spring24>ls -al ~/.ssh
total 29
drwxr-xr-x 1 semiyari AMERICAN+Group(513)  0 Jan 15 18:17 .
drwxr-xr-x 1 semiyari AMERICAN+Group(513)  0 Jan 16 14:50 ..
-rw-r--r-- 1 semiyari AMERICAN+Group(513) 419 Jan 15 18:17 id_ed25519
-rw-r--r-- 1 semiyari AMERICAN+Group(513) 109 Jan 15 18:17 id_ed25519.pub
-rw-r--r-- 1 semiyari AMERICAN+Group(513) 841 Jan 15 17:00 known_hosts

```



The screenshot shows a Windows Command Prompt window titled 'MINGW64: c:/Users/semyari'. The prompt shows the user 'semyari@AU302413' running the command 'ls -al ~/.ssh'. The output is identical to the one in the RStudio terminal, listing the files id\_ed25519, id\_ed25519.pub, and known\_hosts. A red arrow points from the text 'BASH' to the file 'id\_ed25519.pub' in the output.

```

semyari@AU302413 MINGW64 ~
$ ls -al ~/.ssh
total 29
drwxr-xr-x 1 semiyari 1049089  0 Jan 15 18:17 ./
drwxr-xr-x 1 semiyari 1049089  0 Jan 16 14:50 ../
-rw-r--r-- 1 semiyari 1049089 419 Jan 15 18:17 id_ed25519
-rw-r--r-- 1 semiyari 1049089 109 Jan 15 18:17 id_ed25519.pub
-rw-r--r-- 1 semiyari 1049089 841 Jan 15 17:00 known_hosts

semyari@AU302413 MINGW64 ~
$ |

```

## Generate a new SSH key

- If you **do** have an SSH key, go to the next section (“Add an SSH key to the ssh-agent”)
- If you **don’t** have an SSH key, follow the below steps.
- In the terminal, run



```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

- Press enter if prompted where to save the key to accept the default location.
- At the prompt, type a password that you can remember.

### Add an SSH key to the ssh-agent

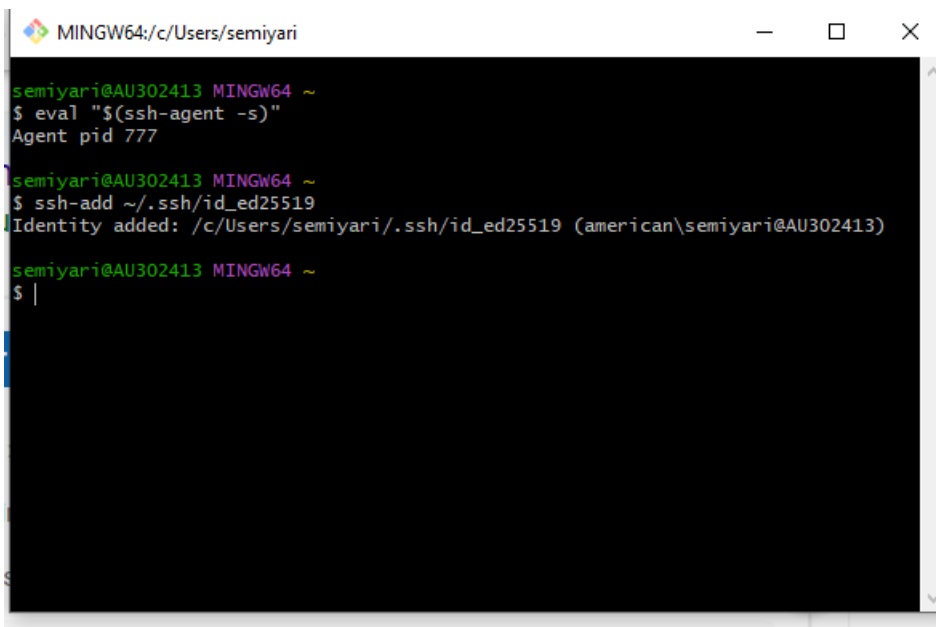
- Why do I need to add an SSH key to the ssh-agent?
  - By adding the key to the ssh-agent, you can securely store the key's passphrase, avoiding the need to enter it every time you connect to the server. If you're connecting to multiple servers or services that require the same SSH key, the ssh-agent can act as a single sign-on solution.

1. In the terminal, run the following to start the ssh-agent in the background.

```
eval "$(ssh-agent -s)"
```

2. In the terminal, run the following to add your SSH private key to the ssh-agent:

```
ssh-add ~/.ssh/id_ed25519
```

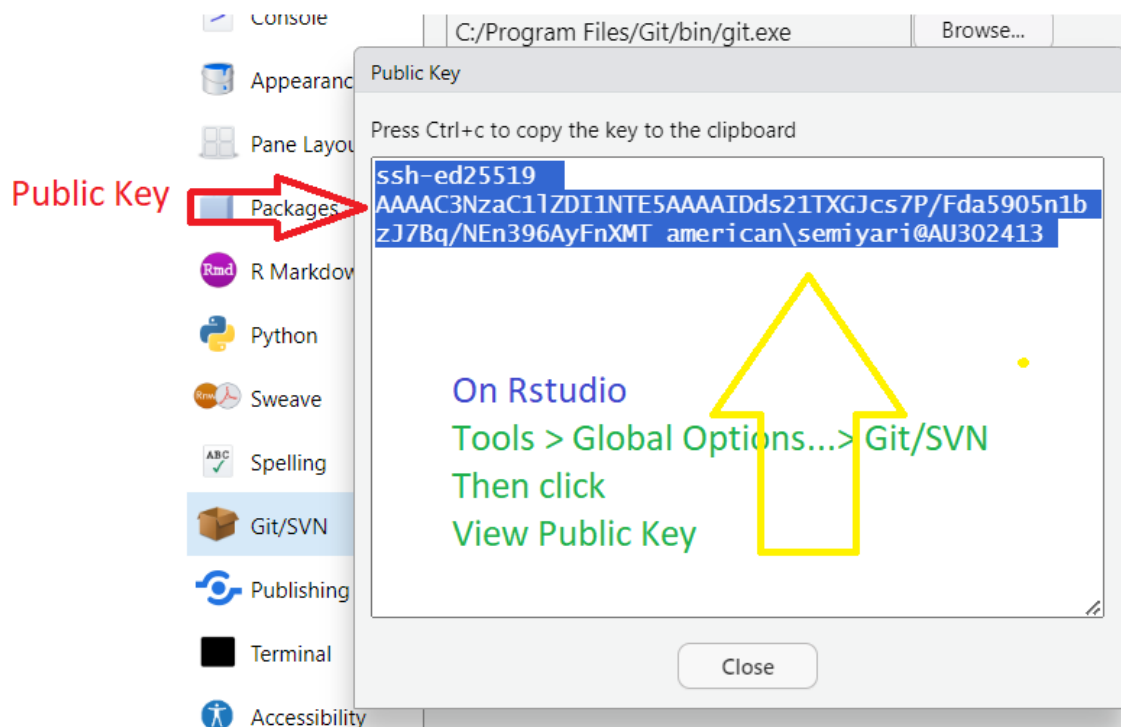
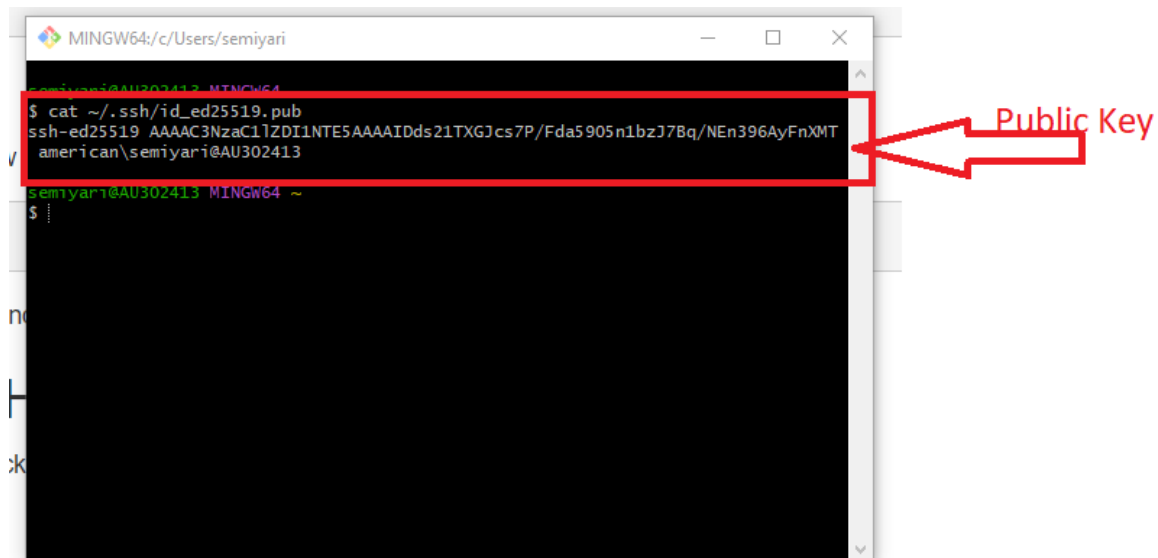


```
MINGW64:/c/Users/semyari
semyari@AU302413 MINGW64 ~
$ eval "$(ssh-agent -s)"
Agent pid 777
semyari@AU302413 MINGW64 ~
$ ssh-add ~/.ssh/id_ed25519
Identity added: /c/Users/semyari/.ssh/id_ed25519 (american\semyari@AU302413)
semyari@AU302413 MINGW64 ~
$ |
```

## Copy Public Key

1. Run the following in the terminal to show the contents of “id\_ed25519.pub”.

```
cat ~/.ssh/id_ed25519.pub
```



2. Highlight the output using your mouse and copy the contents.