Lab Exercise 5- Generate and Use SSH Key with Git and GitHub

Objective:

To learn how to generate an SSH key, add it to GitHub, and use it to securely connect and push code without repeatedly entering a password.

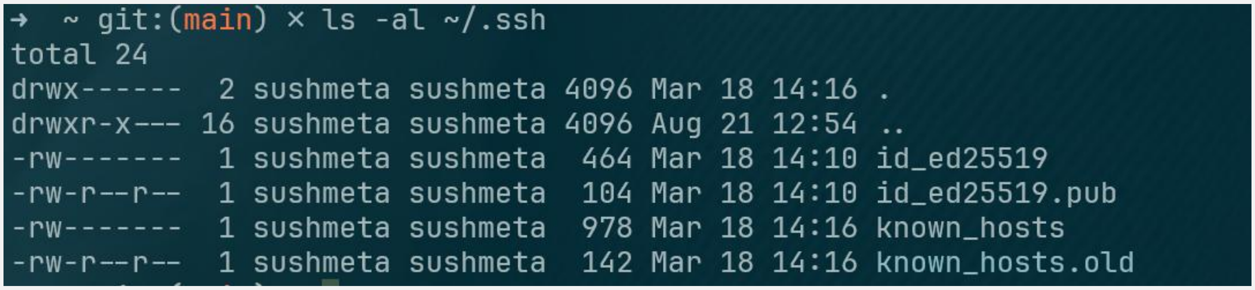
Prerequisites

- Git installed on your local machine
- GitHub account
- Basic understanding of Git commands

Step 1 – Check for Existing SSH Keys

Run:

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



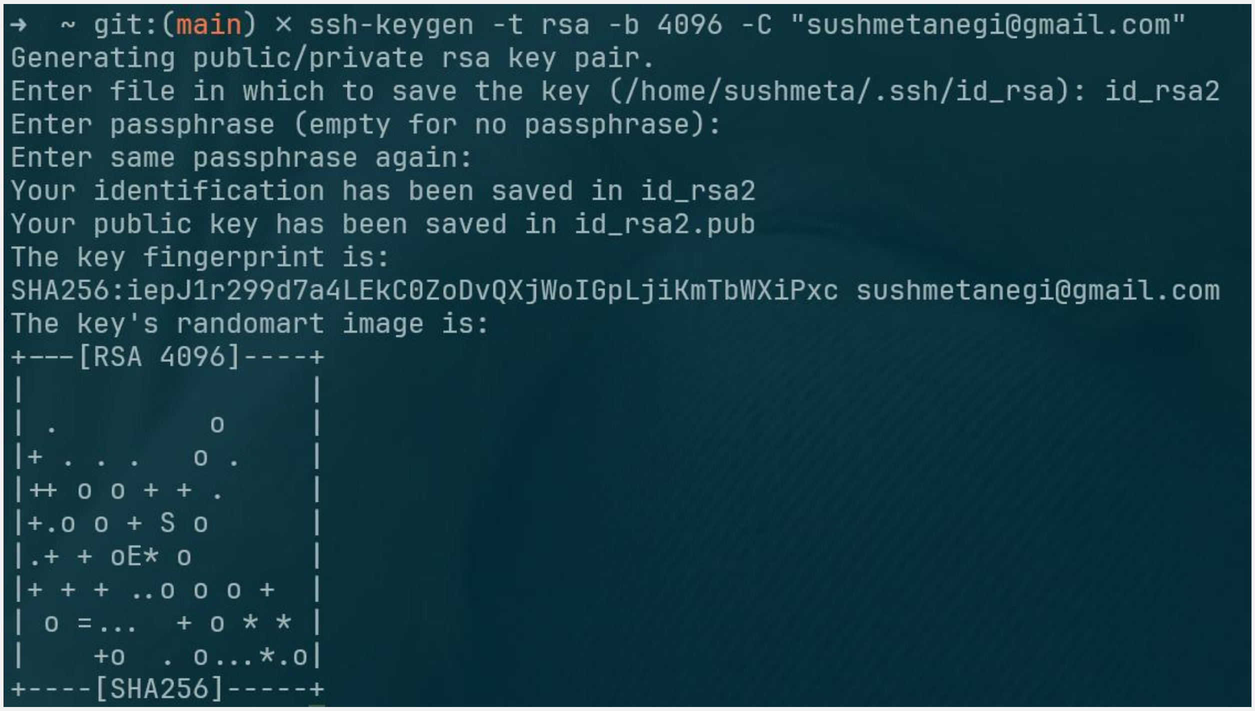
```
➜  ~ git:(main) ✗ ls -al ~/.ssh
total 24
drwx------  2 sushmeta sushmeta 4096 Mar 18 14:16 .
drwxr-x--- 16 sushmeta sushmeta 4096 Aug 21 12:54 ..
-rw-------  1 sushmeta sushmeta  464 Mar 18 14:10 id_ed25519
-rw-r--r--  1 sushmeta sushmeta  104 Mar 18 14:10 id_ed25519.pub
-rw-------  1 sushmeta sushmeta  978 Mar 18 14:16 known_hosts
-rw-r--r--  1 sushmeta sushmeta  142 Mar 18 14:16 known_hosts.old
```

Look for files like id_rsa and id_rsa.pub. If they exist, you may already have an SSH key.

Step 2 – Generate a New SSH Key

Run:

```
ssh-keygen -t rsa -b 4096 -C your_email@example.com
```



```
➜  ~ git:(main) ✗ ssh-keygen -t rsa -b 4096 -C "sushmetanegi@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sushmeta/.ssh/id_rsa): id_rsa2
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_rsa2
Your public key has been saved in id_rsa2.pub
The key fingerprint is:
SHA256:iepJ1r299d7a4LEkC0ZoDvQXjWoIGpLjiKmTbWXiPxc sushmetanegi@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|                 |
| .         o     |
|+ . . .   o .    |
|++ o o + + .     |
|+.o o + S o      |
|.+ + oE* o       |
|+ + + ..o o o +  |
| o =...  + o * * |
|    +o  . o...*.o|
+----[SHA256]-----+
```

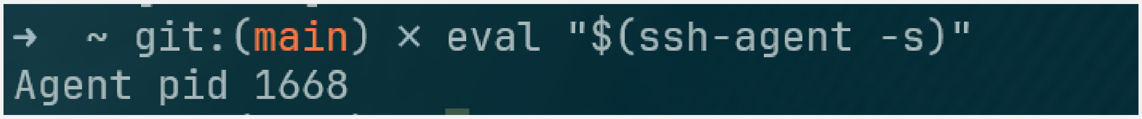
- **-t rsa** → key type
- **-b 4096** → key length
- **-C** → comment (your GitHub email)

When prompted:

- Press **Enter** to save in the default location: /home/user/.ssh/id_rsa (Linux/Mac) or C:\Users\<username>\.ssh\id_rsa (Windows)
- Optionally, set a passphrase for extra security.

Step 3 – Start the SSH Agent

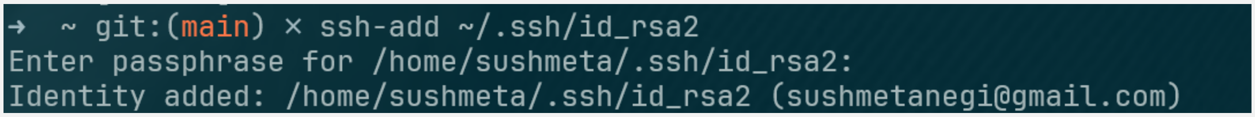
```
eval "$(ssh-agent -s)"
```



```
→  ~ git:(main) × eval "$(ssh-agent -s)"
Agent pid 1668
```

Step 4 – Add SSH Key to the Agent

```
ssh-add ~/.ssh/id_rsa
```

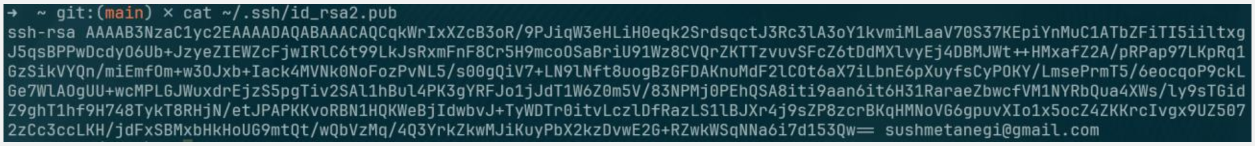


```
→  ~ git:(main) × ssh-add ~/.ssh/id_rsa2
Enter passphrase for /home/sushmeta/.ssh/id_rsa2:
Identity added: /home/sushmeta/.ssh/id_rsa2 (sushmetanegi@gmail.com)
```

Step 5 – Add SSH Key to GitHub

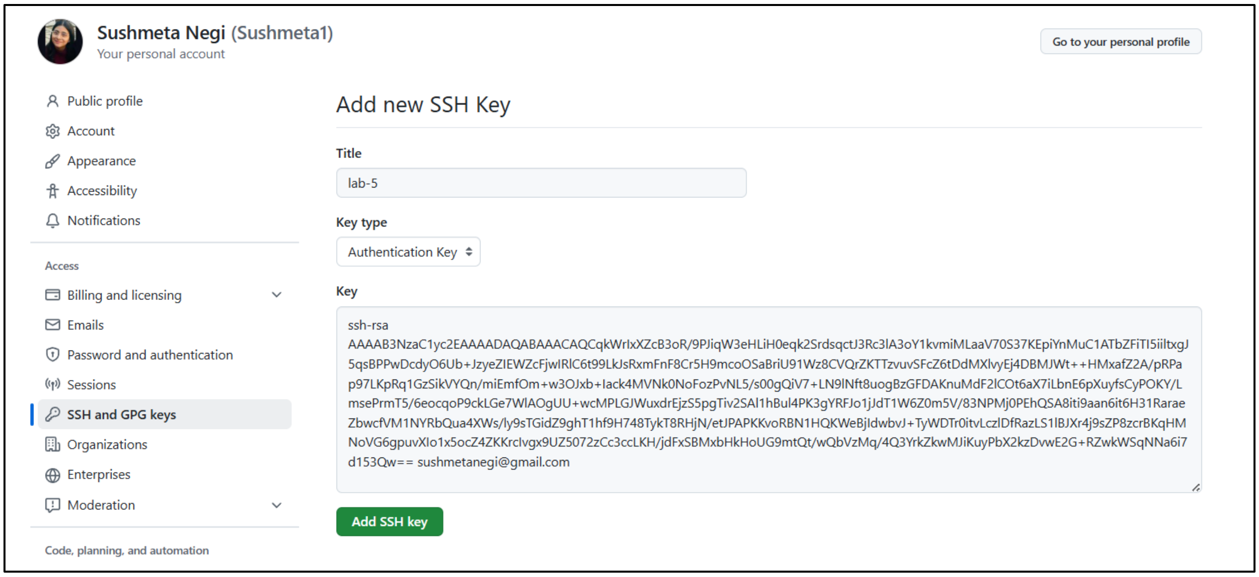
1. Copy the public key:

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



```
➜  ~ git:(main) ✗ cat ~/.ssh/id_rsa2.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCqkWrIxXZcB3oR/9PJiqW3eHLiH0eqk2SrdsqctJ3Rc3lA3oY1kvmiMLaaV70S37KEpiYnMuC1ATbZFiTI5iiltxg
J5qsBPPwDcdyO6Ub+JzyeZIEWZcFjwIRlC6t99LkJsRxmFnF8Cr5H9mcoOSaBriU91Wz8CVQrZKTTzvuvSFcZ6tDdMXlvyEj4DBMJWt++HMxafZ2A/pRPap97LKpRq1
GzSikVYQn/miEmfOm+w3OJxb+Iack4MVNk0NoFozPvNL5/s00gQiV7+LN9lNft8uogBzGFDAKnuMdF2lCOt6aX7iLbnE6pXuyfsCyPOKY/LmsePrmT5/6eocqoP9ckL
Ge7WlAOgUU+wcMPLGJWuxdrEjzS5pgTiv2SAl1hBul4PK3gYRFJo1jJdT1W6Z0m5V/83NPMj0PEhQSA8iti9aan6it6H31RaraeZbwcfVM1NYRbQua4XWs/ly9sTGid
Z9ghT1hf9H748TykT8RHjN/etJPAPKKvoRBN1HQKWeBjIdwbvJ+TyWDTr0itvLczlDfRazLS1lBJXr4j9sZP8zcrBKqHMNoVG6gpuvXIo1x5ocZ4ZKKrcIvgx9UZ507
2zCc3ccLKH/jdFxSBMxbHkHoUG9mtQt/wQbVzMq/4Q3YrkZkwMJiKuyPbX2kzDvwE2G+RZwkWSqNNa6i7d153Qw== sushmetanegi@gmail.com
```

2. Log in to GitHub → **Settings** → **SSH and GPG Keys** → **New SSH key**.



Add new SSH Key

Title

lab-5

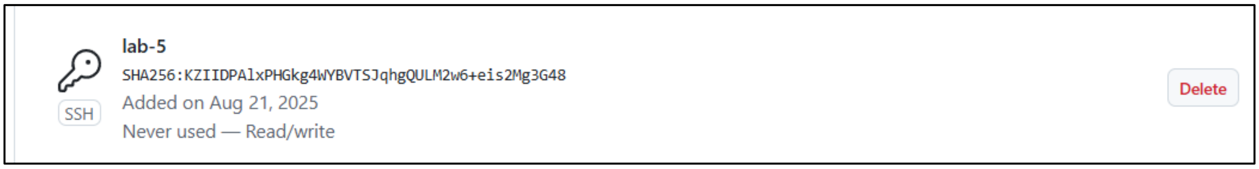
Key type

Authentication Key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQCqkWrIxXZcB3oR/9PJiqW3eHLiH0eqk2SrdsqctJ3Rc3lA3oY1kvmiMLaaV70S37KEpiYnMuC1ATbZFiTI5iiltxgJ
5qsBPPwDcdyO6Ub+JzyeZIEWZcFjwIRlC6t99LkJsRxmFnF8Cr5H9mcoOSaBriU91Wz8CVQrZKTTzvuvSFcZ6tDdMXlvyEj4DBMJWt++HMxafZ2A/pRPa
p97LKpRq1GzSikVYQn/miEmfOm+w3OJxb+Iack4MVNk0NoFozPvNL5/s00gQiV7+LN9lNft8uogBzGFDAKnuMdF2lCOt6aX7iLbnE6pXuyfsCyPOKY/L
msePrmT5/6eocqoP9ckLGe7WlAOgUU+wcMPLGJWuxdrEjzS5pgTiv2SAl1hBul4PK3gYRFJo1jJdT1W6Z0m5V/83NPMj0PEhQSA8iti9aan6it6H31Rarae
ZbwcfVM1NYRbQua4XWs/ly9sTGidZ9ghT1hf9H748TykT8RHjN/etJPAPKKvoRBN1HQKWeBjIdwbvJ+TyWDTr0itvLczlDfRazLS1lBJXr4j9sZP8zcrBKqHM
NoVG6gpuvXIo1x5ocZ4ZKKrcIvgx9UZ5072zCc3ccLKH/jdFxSBMxbHkHoUG9mtQt/wQbVzMq/4Q3YrkZkwMJiKuyPbX2kzDvwE2G+RZwkWSqNNa6i7
d153Qw== sushmetanegi@gmail.com
```

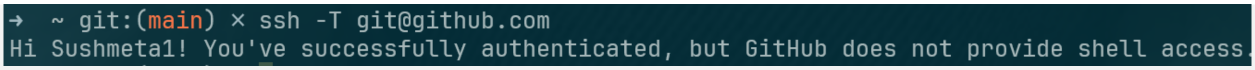
3. Paste the key and save.



lab-5
SHA256:KZIIDPAlxPHGkg4WYBVTSJqhgQULM2w6+eis2Mg3G48
Added on Aug 21, 2025
Never used — Read/write

Step 6 – Test SSH Connection

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



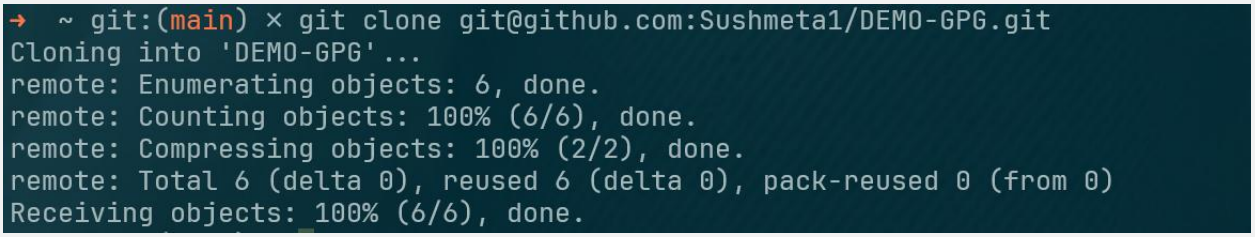
```
➜  ~ git:(main) × ssh -T git@github.com
Hi Sushmeta1! You've successfully authenticated, but GitHub does not provide shell access.
```

Expected output:

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

Step 7 – Use SSH to Clone a Repository

```
git clone git@github.com:<username>/<repository>.git
```



```
➜  ~ git:(main) × git clone git@github.com:Sushmeta1/DEMO-GPG.git
Cloning into 'DEMO-GPG'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.
```

Now you can pull and push without entering your username/password.

Use Case

Scenario:

An organization’s developers often need to push code to GitHub multiple times a day. Using SSH keys eliminates the need to repeatedly enter credentials, while maintaining secure, encrypted communication between the developer’s machine and GitHub.

Table – HTTPS vs SSH for GitHub

Feature	**HTTPS**	**SSH**
Authentication	Username & password / token	SSH key pair
Convenience	Requires login each session	No password once key is added
Security	Encrypted, but password-based auth	Encrypted, key-based authentication
Best For	Occasional access	Frequent development work