Name – Utkarsh Rawat

Sap Id – 500124586

Batch 3 – DevOps

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

Objective:

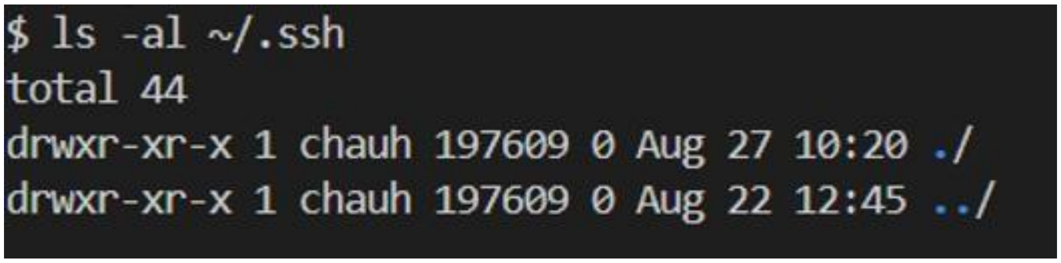
To learn how to generate an SSH key, add it to GitHub, and use it to securely connect and push code without repeatedly entering a password.

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
```



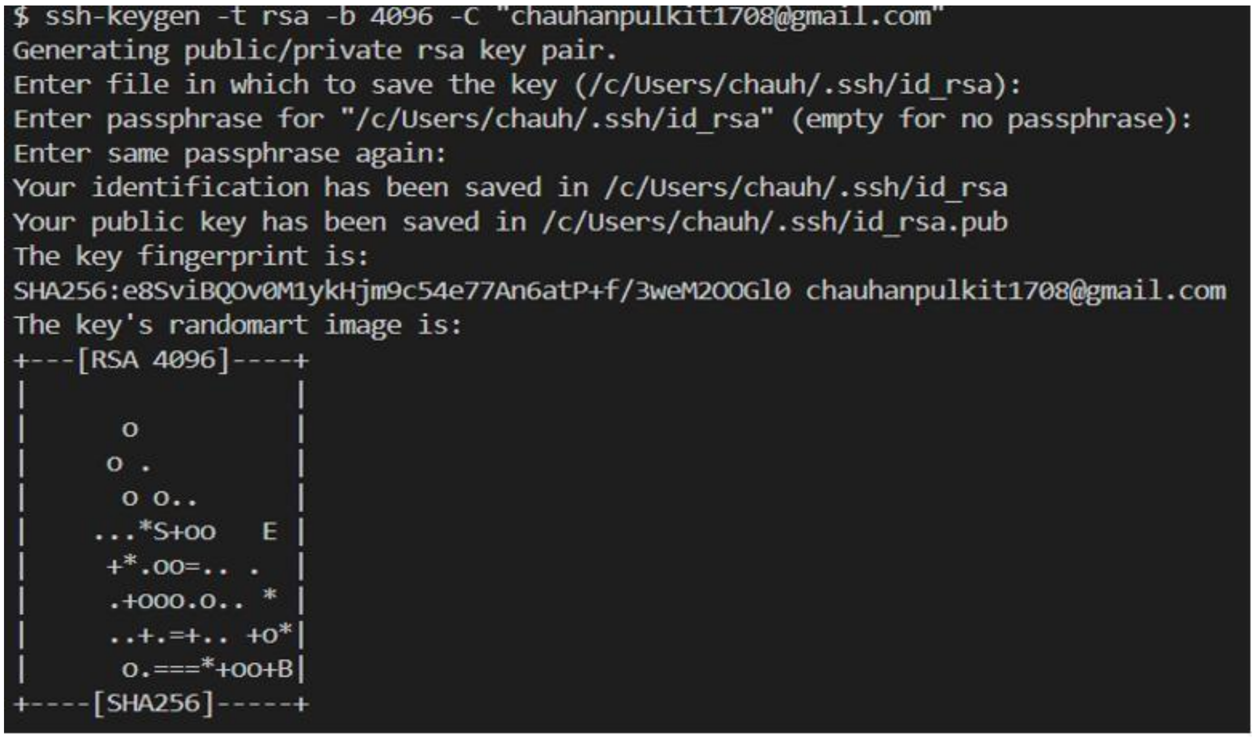
```
$ ls -al ~/.ssh
total 44
drwxr-xr-x 1 chauh 197609 0 Aug 27 10:20 ./
drwxr-xr-x 1 chauh 197609 0 Aug 22 12:45 ../
```

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"
```

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



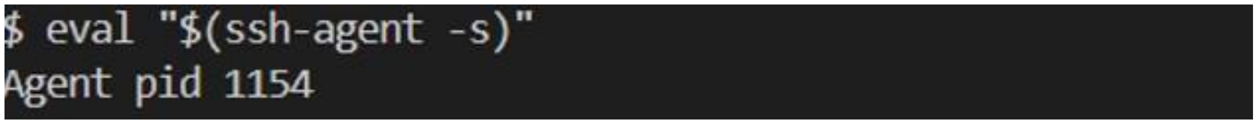
```
$ ssh-keygen -t rsa -b 4096 -C "chauhanpulkit1708@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/chauh/.ssh/id_rsa):
Enter passphrase for "/c/Users/chauh/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/chauh/.ssh/id_rsa
Your public key has been saved in /c/Users/chauh/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:e8SviBQOv0M1ykHjm9c54e77An6atP+f/3weM2OOGl0 chauhanpulkit1708@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|                 |
|       o         |
|      o .        |
|       o o..     |
|     ...*S+oo   E |
|      +*.oo=.. . |
|      .+ooo.o.. * |
|      ..+.=+.. +o*|
|       o.===*+oo+B|
+----[SHA256]-----+
```

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

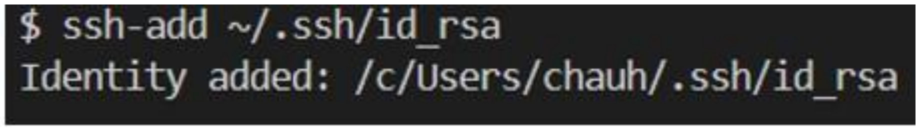
```
eval "$(ssh-agent -s)"
```



```
$ eval "$(ssh-agent -s)"
Agent pid 1154
```

Step 4 – Add SSH Key to the Agent

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

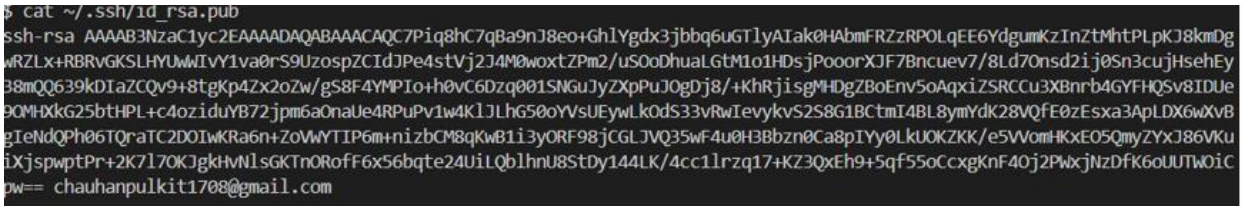


```
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/chauh/.ssh/id_rsa
```

Step 5 – Add SSH Key to GitHub

1. Copy the public key:

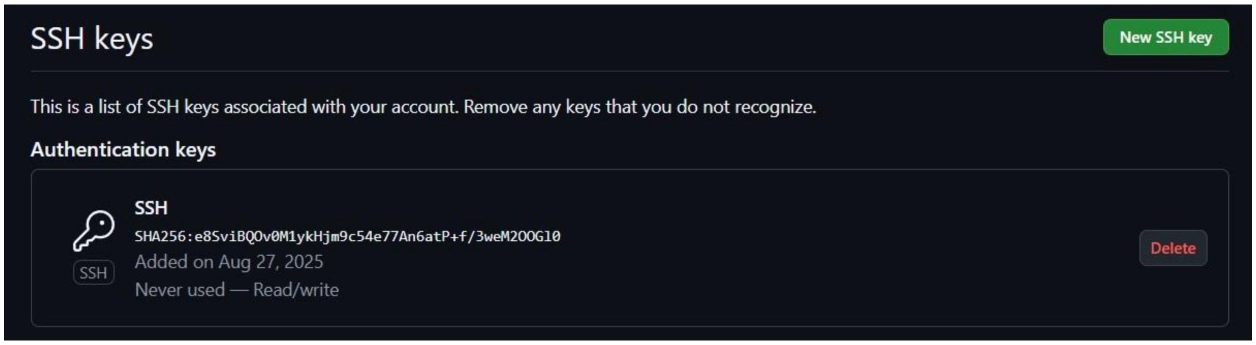
```
cat ~/.ssh/id_rsa.pub
```



```
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQC7Piq8hC7qBa9nJ8eo+GhlYgdx3jbbq6uGTlyAIak0HAbmFRZzRPOLqEE6YdgumKzInZtMhtPLpKJ8kmDg
wRZLx+RBRvGKSLHYUwWIvY1va0rS9UzospZCIdJPe4stVj2J4M0woxtZPm2/uSOoDhuaLGtM1o1HDsjPooorXJF7Bncuev7/8Ld7Onsd2ij0Sn3cujHsehEy
38mQQ639kDIaZCQv9+8tgKp4Zx2oZw/gS8F4YMPIo+h0vC6Dzq001SNGuJyZXpPuJOgDj8/+KhRjisgMHDgZBoEnv5oAqxiZSRCCu3XBnrb4GYFHQSv8IDUe
9OMHXkG25btHPL+c4oziduYB72jpm6aOnaUe4RPuPv1w4KlJLhG50oYVsUEywLkOdS33vRwIevykvS2S8G1BCtmI4BL8ymYdK28VQfE0zEsxa3ApLDX6wXvB
gIeNdQPh06TQraTC2DOIwKRa6n+ZoVWYTIP6m+nizbCM8qKwB1i3yORF98jCGLJVQ35wF4u0H3Bbzn0Ca8pIYy0LkUOKZKK/e5VVomHKxEO5QmyZYxJ86VKu
iXjspwptPr+2K7l7OKJgkHvNlsGKTnORofF6x56bqte24UiLQblhnU8StDy144LK/4cc1lrzq17+KZ3QxEh9+5qf55oCcxgKnF4Oj2PWxjNzDfK6oUUTWOiC
pw== chauhanpulkit1708@gmail.com
```

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

3. Paste the key and save.


SSH keys

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

Authentication keys

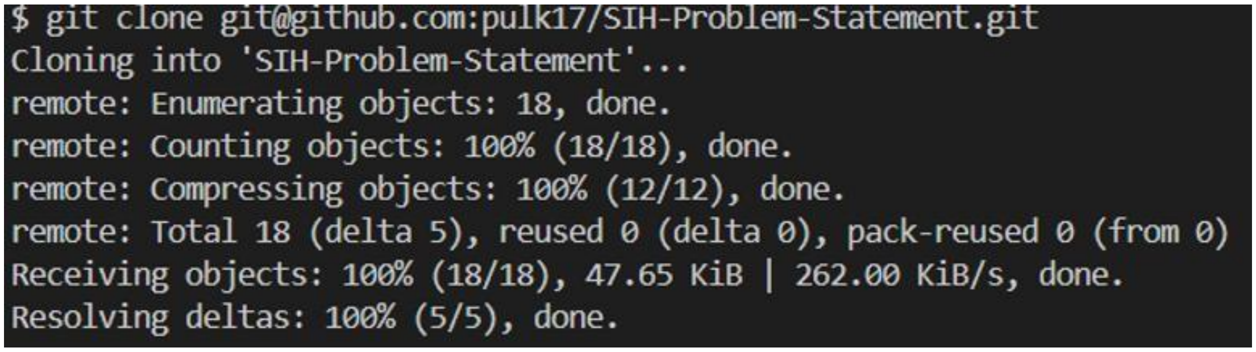
SSH
SHA256:e8SviBQOv0M1ykHjm9c54e77An6atP+f/3weM2OOG10
Added on Aug 27, 2025
Never used — Read/write

Step 6 – Test SSH Connection

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

Step 7 – Use SSH to Clone a Repository

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


```
$ git clone git@github.com:pulk17/SIH-Problem-Statement.git
Cloning into 'SIH-Problem-Statement'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 18 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (18/18), 47.65 KiB | 262.00 KiB/s, done.
Resolving deltas: 100% (5/5), 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