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

Name: Ayush Bhardwaj

Sap id: 500124917

Enrolment no.: R2142231775

Batch 2 DevOps

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

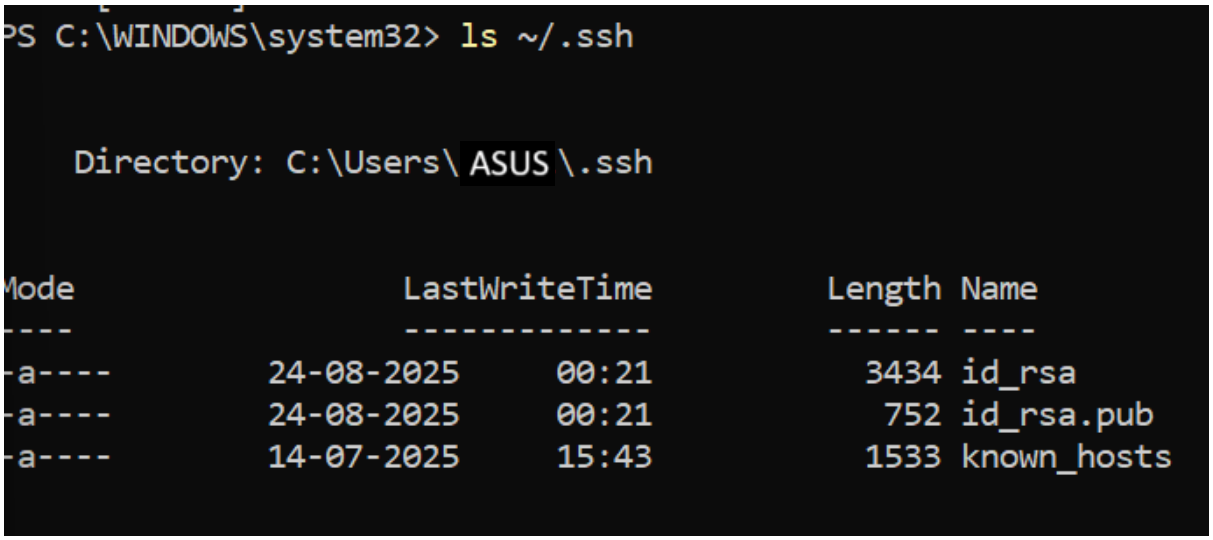
- Git installed on your local machine
- GitHub account
- Basic understanding of Git commands

Step 1 – Check for Existing SSH Keys

Run:

ls -al ~/.ssh

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)

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)"

Step 4 – Add SSH Key to the Agent

ssh-add ~/.ssh/id_rsa

Step 5 – Add SSH Key to GitHub

1. Copy the public key:

cat ~/.ssh/id_rsa.pub

2. Log in to GitHub → **Settings** → **SSH and GPG Keys** → **New SSH key**.
3. Paste the key and save.

Step 6 – Test SSH Connection

ssh -T git@github.com

Expected output:

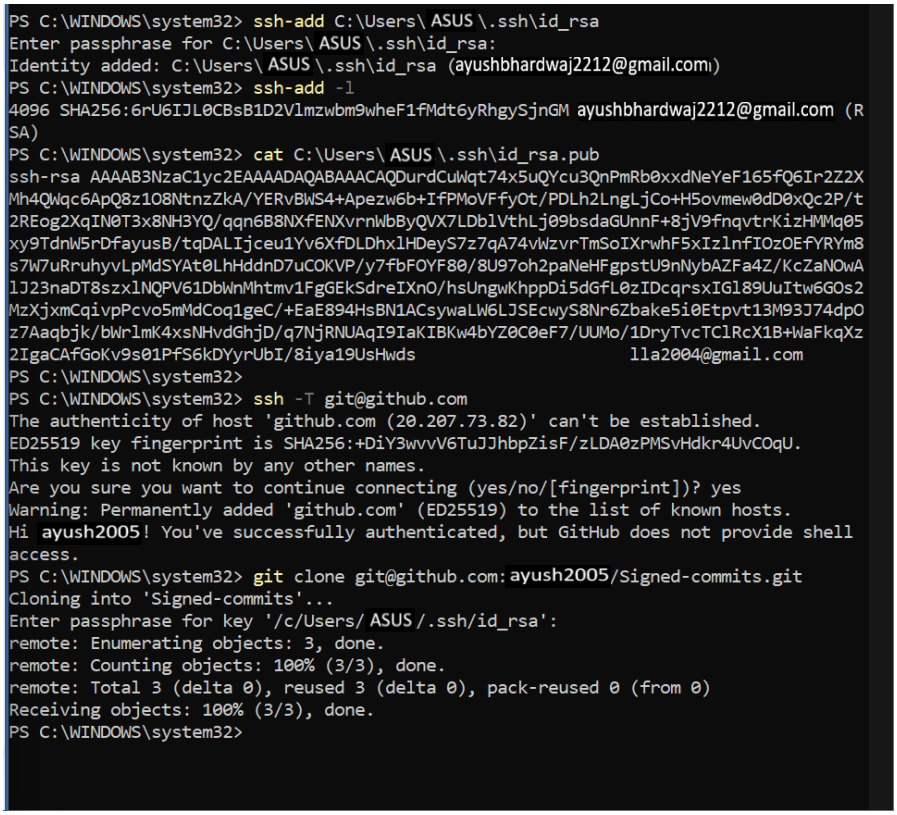
Hi <username>! You've successfully authenticated, but GitHub does not provide shell access.

Step 7 – Use SSH to Clone a Repository

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

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

OUTPUT:



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

Feature	HTTPS	SSH
Best For	Occasional access	Frequent development work