

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.

```
PS C:\WINDOWS\system32> ls ~/.ssh

Directory: C:\Users\ASUS\.ssh

Mode                LastWriteTime         Length Name
----                -
-a----           24-08-2025         00:21       3434 id_rsa
-a----           24-08-2025         00:21        752 id_rsa.pub
-a----           14-07-2025         15:43       1533 known_hosts
```

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:

```
PS C:\WINDOWS\system32> ssh-add C:\Users\ASUS\.ssh\id_rsa
Enter passphrase for C:\Users\ASUS\.ssh\id_rsa:
Identity added: C:\Users\ASUS\.ssh\id_rsa (ayushbhardwaj2212@gmail.com)
PS C:\WINDOWS\system32> ssh-add -l
4096 SHA256:6rU6IJL0CBsB1D2V1mzwbm9wheF1fMdt6yRhgySjnGM ayushbhardwaj2212@gmail.com (RSA)
PS C:\WINDOWS\system32> cat C:\Users\ASUS\.ssh\id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACuWqt74x5uQYcu3QnRb0xxdNeYeF165fQ6Ir2Z2X
Mh4QWqc6ApQ8z108NtnzZKA/YERvBWS4+Apezw6b+IfPMoVfFyOt/PDLh2LNgLjCo+H5ovmew0dD0xQc2P/t
2REog2XqIN0T3x8NH3YQ/qqn6B8NXfENXvrnWbByQVX7LDb1VthLj09bsdaGUnnF+8jv9fnqvtrKizHMMq05
xy9TdnW5rDfayusB/tqDALIjceu1Yv6XfDLdHx1HDeyS7z7qA74vWzvrTmSoIXrwhF5xIzlnfIOz0EfYRYm8
s7W7uRruhYvLpMdSYAt0LhHddnD7uCOkVP/y7fbFOYF80/8U97oh2paNeHFgpstU9nNybAZFa4Z/KcZaNOwA
1J23naDT8szx1NQPV61DbWnMhtmv1FgGekSdreIXnO/hsUngwKhppDi5dGfL0zIDcqrSxIG189UuItw6G0s2
MzXjxmCqivpPcvo5mMdCoqigeC/+EaE894HsBN1ACsywaLW6LJSEcwyS8Nr6Zbake5i0Etpvt13M93J74dpO
z7Aaqbjk/bwr1mk4xsNHvdGhJd/q7NjRNUAQI9IaKIBKw4bYZ0C0eF7/UUMo/1DryTvcTCLRcX1B+WaFkqXz
2IgaCAfGoKv9s01PFs6kDYrUbI/8iya19Ushwds 1la2004@gmail.com
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> ssh -T git@github.com
The authenticity of host 'github.com (20.207.73.82)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi ayush2005! You've successfully authenticated, but GitHub does not provide shell
access.
PS C:\WINDOWS\system32> git clone git@github.com:ayush2005/Signed-commits.git
Cloning into 'Signed-commits'...
Enter passphrase for key '/c/Users/ASUS/.ssh/id_rsa':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
PS C:\WINDOWS\system32>
```

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