Namit Rampal

SAP ID – 500123236

Batch – B2 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:

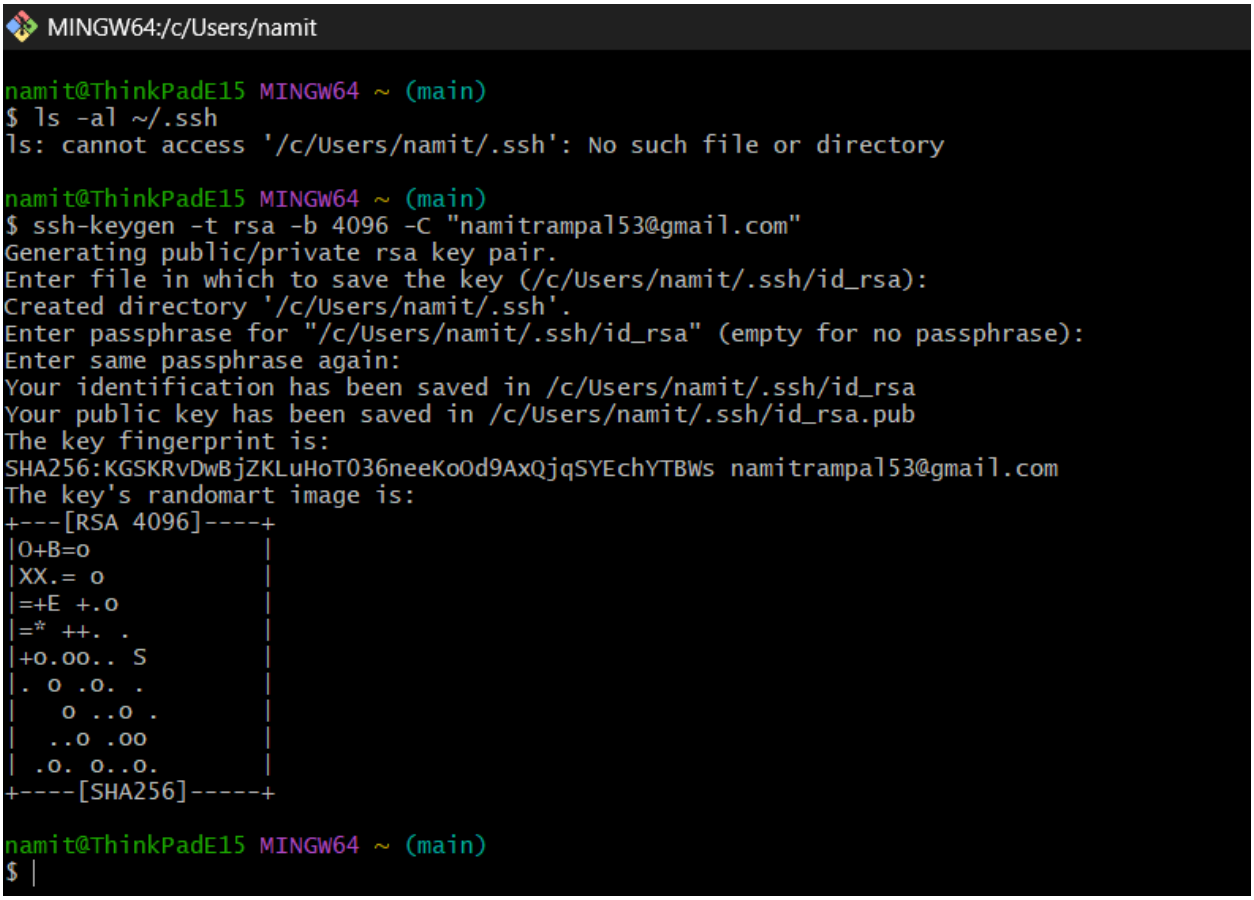
```
ls -al ~/.ssh
```

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

- **-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
```

```
namit@ThinkPadE15 MINGW64 ~ (main)
$ eval "$(ssh-agent -s)"
Agent pid 1113

namit@ThinkPadE15 MINGW64 ~ (main)
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/namit/.ssh/id_rsa (namitrampal53@gmail.com)

namit@ThinkPadE15 MINGW64 ~ (main)
$
```

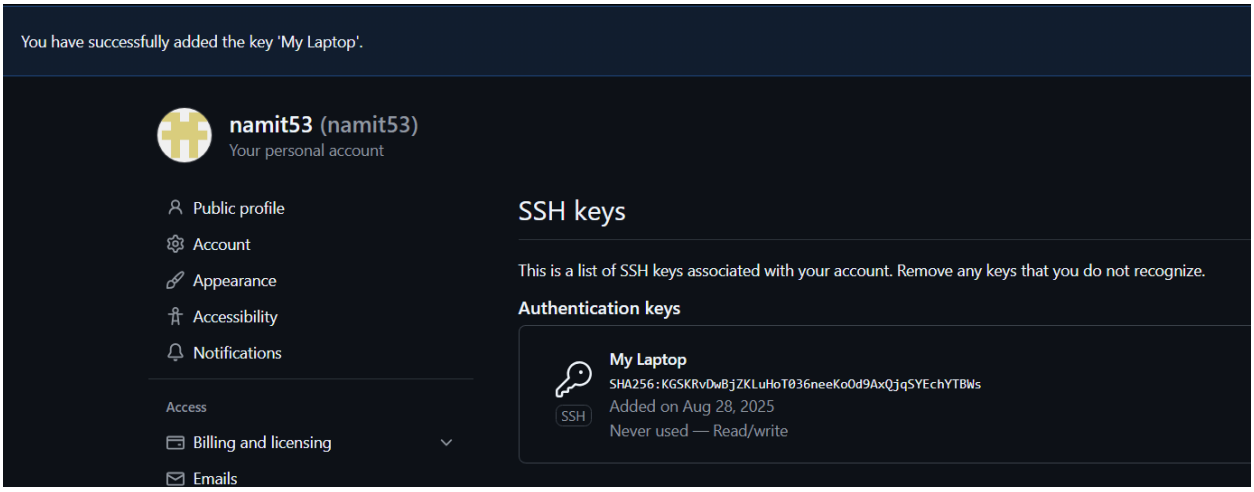
Step 5 – Add SSH Key to GitHub

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
```

```
namit@ThinkPadE15 MINGW64 ~ (main)
$ 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 namit53! You've successfully authenticated, but GitHub does not provide shell access.

namit@ThinkPadE15 MINGW64 ~ (main)
$
```

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.

```
namit@ThinkPadE15 MINGW64 ~ (main)
$ git clone git@github.com:namit53/exp4_DevSecOps.git exp4_DevSecOps_new
Cloning into 'exp4_DevSecOps_new'...
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.

namit@ThinkPadE15 MINGW64 ~ (main)
$
```

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