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

Name:-VanshBhatt

SapId:- 500125395

R.No:- R2142231689

Batch:- DevOps B1

To:- Hitesh Kumar Sharma Sir

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.

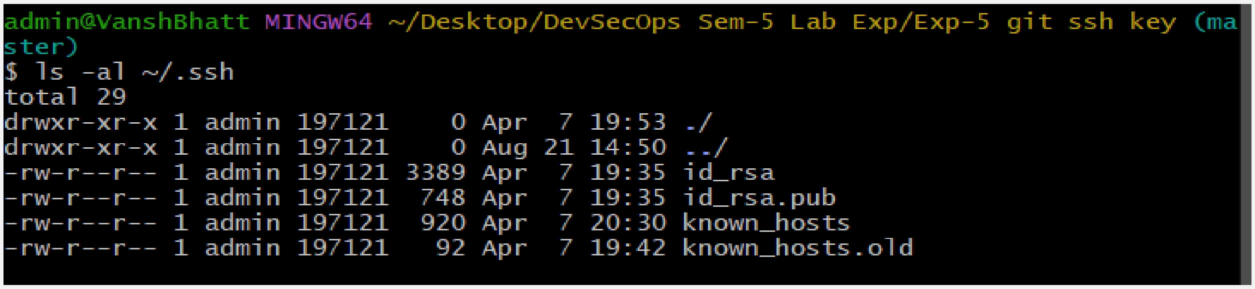
Prerequisites

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

Step 1 – Check for Existing SSH Keys

Run:

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



```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/Exp-5 git ssh key (ma
ster)
$ ls -al ~/.ssh
total 29
drwxr-xr-x 1 admin 197121    0 Apr  7 19:53 ./
drwxr-xr-x 1 admin 197121    0 Aug 21 14:50 ../
-rw-r--r-- 1 admin 197121 3389 Apr  7 19:35 id_rsa
-rw-r--r-- 1 admin 197121  748 Apr  7 19:35 id_rsa.pub
-rw-r--r-- 1 admin 197121  920 Apr  7 20:30 known_hosts
-rw-r--r-- 1 admin 197121   92 Apr  7 19:42 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:

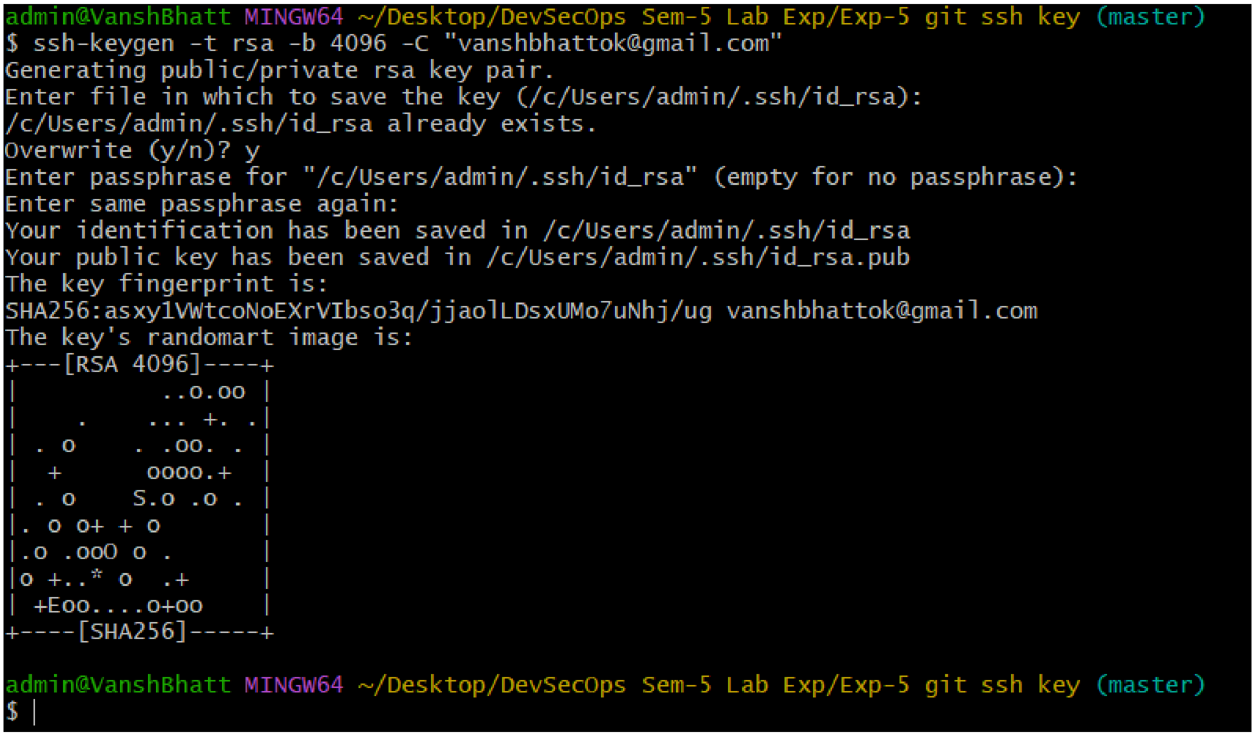
```
ssh-keygen -t rsa -b 4096 -C your_email@example.com
```

- **-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.

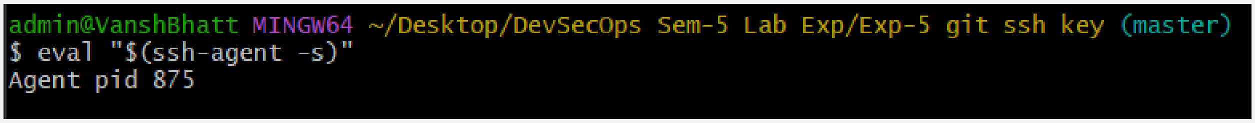


```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/Exp-5 git ssh key (master)
$ ssh-keygen -t rsa -b 4096 -C "vanshbhattok@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/admin/.ssh/id_rsa):
/c/Users/admin/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase for "/c/Users/admin/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/admin/.ssh/id_rsa
Your public key has been saved in /c/Users/admin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:asxy1VWtcoNoEXrVIbso3q/jjaolLDsxUMo7uNhj/ug vanshbhattok@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|          ..o.oo |
|    .    ... +. .|
| . o    . .oo. . |
|  +      oooo.+  |
| . o    S.o .o . |
|. o o+ + o       |
|.o .ooO o .      |
|o +..* o  .+     |
| +Eoo....o+oo    |
+----[SHA256]-----+

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/Exp-5 git ssh key (master)
$ |
```

Step 3 – Start the SSH Agent

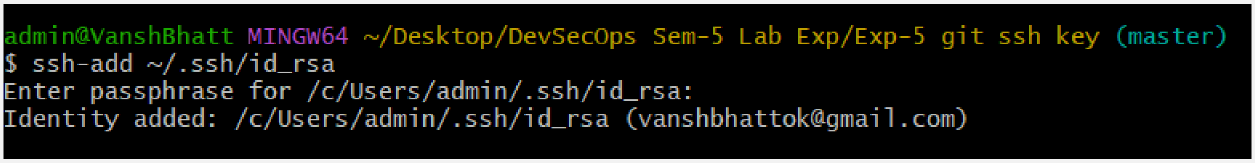
```
eval "$(ssh-agent -s)"
```



```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/Exp-5 git ssh key (master)
$ eval "$(ssh-agent -s)"
Agent pid 875
```

Step 4 – Add SSH Key to the Agent

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

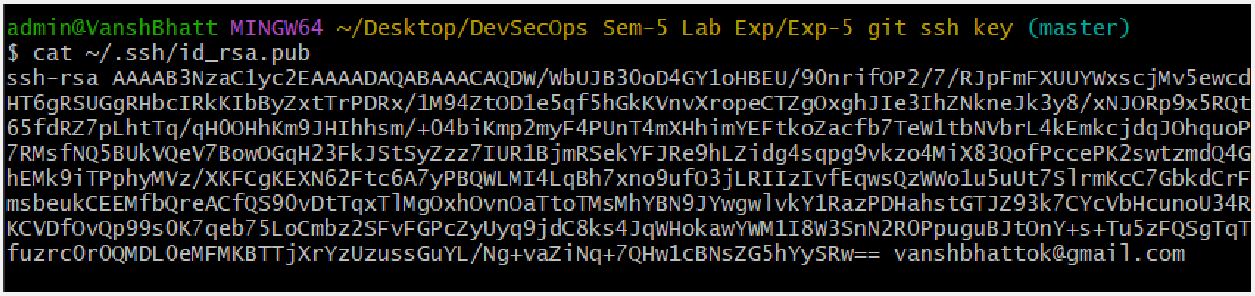


```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/Exp-5 git ssh key (master)
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /c/Users/admin/.ssh/id_rsa:
Identity added: /c/Users/admin/.ssh/id_rsa (vanshbhattok@gmail.com)
```

Step 5 – Add SSH Key to GitHub

1. Copy the public key:

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



```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/Exp-5 git ssh key (master)
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDW/WbUJB30oD4GY1oHBEU/90nrifOP2/7/RJpFmFXUUYWxscjMv5ewcd
HT6gRSUGgRHbcIRkKIbByZxtTrPDRx/1M94ZtOD1e5qf5hGkKVnvXropeCTZgOxghJIe3IhZNkneJk3y8/xNJORp9x5RQt
65fdRZ7pLhtTq/qHOOHhKm9JHIhhsm/+O4biKmp2myF4PUnT4mXHhimYEFtkoZacfb7TeW1tbNVbrL4kEmkcjdqJOhquoP
7RMsfNQ5BUkVQeV7BowOGqH23FkJStSyZzz7IUR1BjmRSekYFJRe9hLZidg4sqpg9vkzo4MiX83QofPccePK2swtzmdQ4G
hEMk9iTPphyMVz/XKFCgKEXN62Ftc6A7yPBQWLMI4LqBh7xno9ufO3jLRIIzIvfEqwsQzWWo1u5uUt7SlrmKcC7GbkdCrF
msbeukCEEMfbQreACfQS90vDtTqxTlMgOxhOvnOaTtoTMsMhYBN9JYwgwlvkY1RazPDHahstGTJZ93k7CYcVbHcunoU34R
KCVDfOvQp99sOK7qeb75LoCmbz2SFvFGPcZyUyq9jdC8ks4JqWHokawYWM1I8W3SnN2ROPpuguBJtOnY+s+Tu5zFQSgTqT
fuzrcOrOQMDLOeMFMKBTTjXrYzUzussGuYL/Ng+vaZiNq+7QHw1cBNsZG5hYySRw== vanshbhattok@gmail.com
```

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

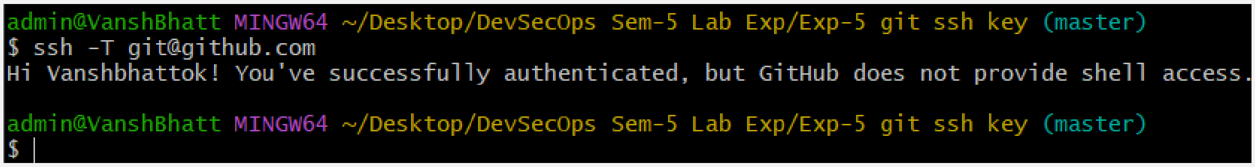
3. Paste the key and save.

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.
```

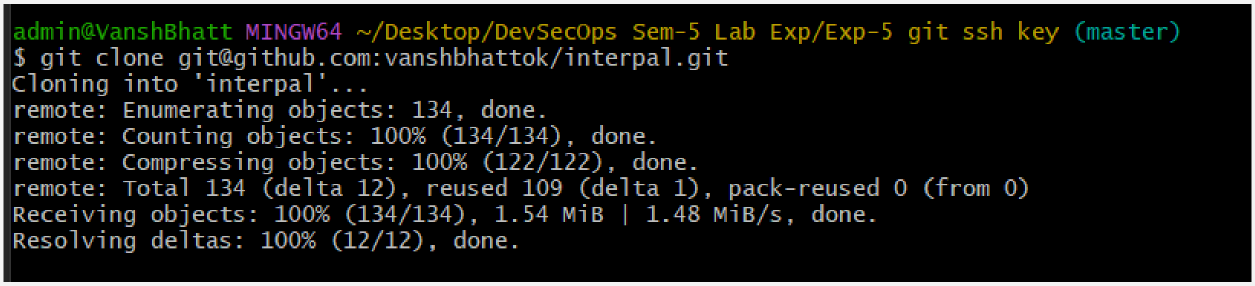


```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/Exp-5 git ssh key (master)
$ ssh -T git@github.com
Hi Vanshbhattok! You've successfully authenticated, but GitHub does not provide shell access.

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/Exp-5 git ssh key (master)
$ |
```

Step 7 – Use SSH to Clone a Repository

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



```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/Exp-5 git ssh key (master)
$ git clone git@github.com:vanshbhattok/interpal.git
Cloning into 'interpal'...
remote: Enumerating objects: 134, done.
remote: Counting objects: 100% (134/134), done.
remote: Compressing objects: 100% (122/122), done.
remote: Total 134 (delta 12), reused 109 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (134/134), 1.54 MiB | 1.48 MiB/s, done.
Resolving deltas: 100% (12/12), 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