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

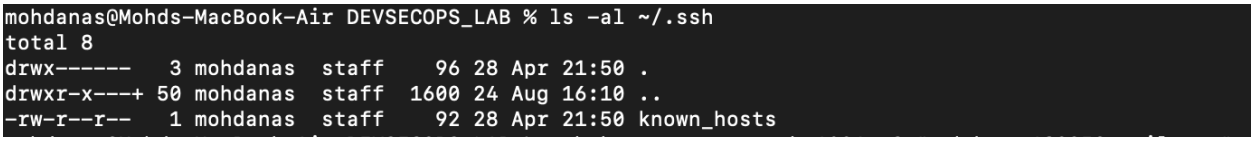
- Git installed on your local machine
- GitHub account
- Basic understanding of Git commands

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.



Step 2 – Generate a New SSH Key

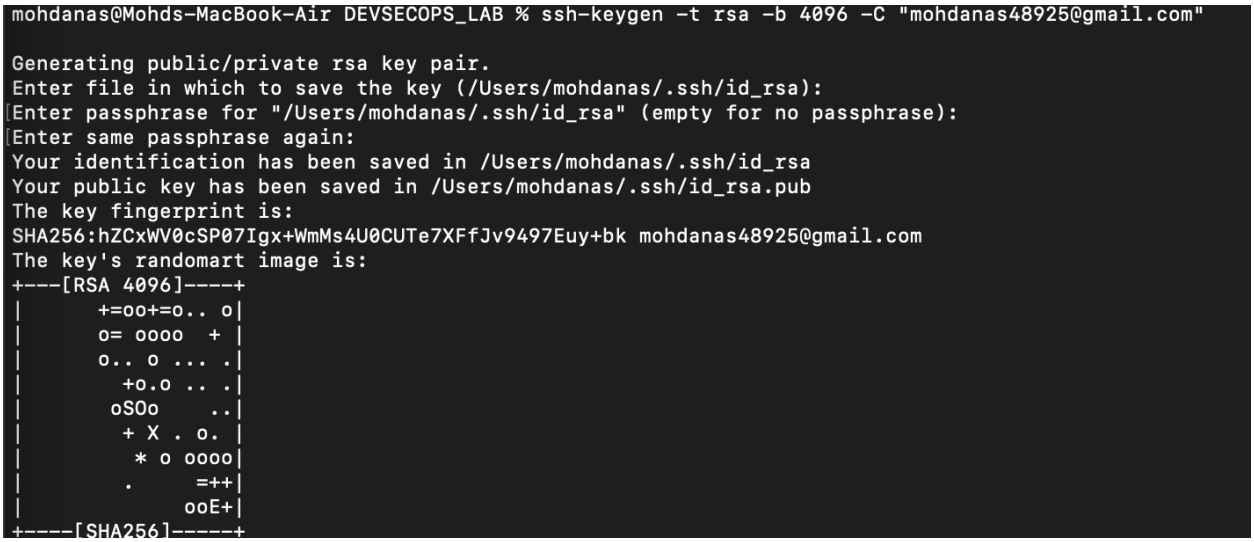
Run:

```
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)"
```

```
mohdanas@Mohds-MacBook-Air DEVSECOPS_LAB % eval "$(ssh-agent -s)"

Agent pid 21950
```

Step 4 – Add SSH Key to the Agent

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

```
mohdanas@Mohds-MacBook-Air DEVSECOPS_LAB % ssh-add ~/.ssh/id_rsa
Identity added: /Users/mohdanas/.ssh/id_rsa (mohdanas48925@gmail.com)
mohdanas@Mohds-MacBook-Air DEVSECOPS_LAB % cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDUFKz3P1l3in5nHDUbhkdjTsdY23csrD+elB5Sz7Cj+zlx2jJZSNj5AsG1NjKJ4O
lqldwE/9on0Be1oyIG1qhp2S2gYoyqL47kxIKZ4CoIjqHwiyE+bIq9j2cI59SE5SSZGn7emsrLCItW3WdDF4inCRsmg/3yFrepDptg
6TDHc0G2AhojyIJo5HhxHMyBOSZKnO15IhBh9qzz1B6Ln05R+q7ZP/8e7BgPVxTfDuWDCfw2sl8IPDFU+EJBoKzKr9zzemtdtZebCg
VwujtVgcLV8DhI+H6a8h7K59vRKfQaUzgtB1VCdsuIgg1hAdZHfvhYO5RWVgF/Koh5Na9KWabIuw10cSbcKqHlVh4EArm6+IBrN36t
1w4+U2Q2/G0EJBu6J7bBtNxCnYurUPn0b+/21MiFsjLp/tyH8KGJ1HYV1mn+7U1i78NGxoxdqY7hAu1rN/6BHscIRIOrGIf3Zw5o+B
TsNLA/FGnRhr605/JPQ78oKHomneIv+6wnqFtHC7mU1zWnpaJlwRPkDSLFsqheSl+4o1BlvFyhJjFQMoWzPgJO2mQkHpJ21N2ULMzh
/yA96RC3pXPOzNHSMXkxY8iSprgooFDzzKom3MYhUJEYAM7fBemN07e94RbmS7/haiYGAK/M/Ki9lvObDeO1BAQ8DPlYXpAA0YUDQh
9Bvq0iww== mohdanas48925@gmail.com
```

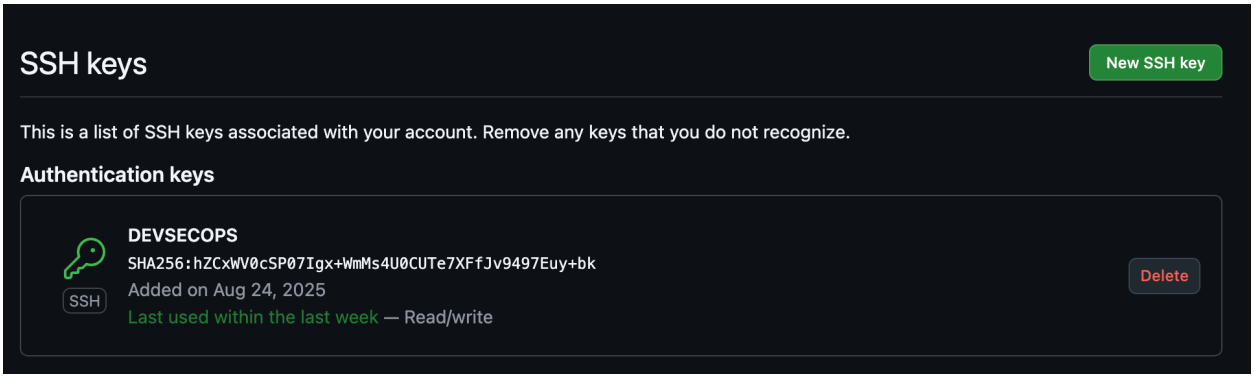
Step 5 – Add SSH Key to GitHub

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.
```

```
Hi mohdd-anas! 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.

```
mohdanas@Mohds-MacBook-Air DEVSECOPS_LAB % git clone git@github.com:mohdd-anas/exp3.git
Cloning into 'exp3'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 27 (delta 6), reused 27 (delta 6), pack-reused 0 (from 0)
Receiving objects: 100% (27/27), 1.80 MiB | 1.63 MiB/s, done.
Resolving deltas: 100% (6/6), done.
```

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