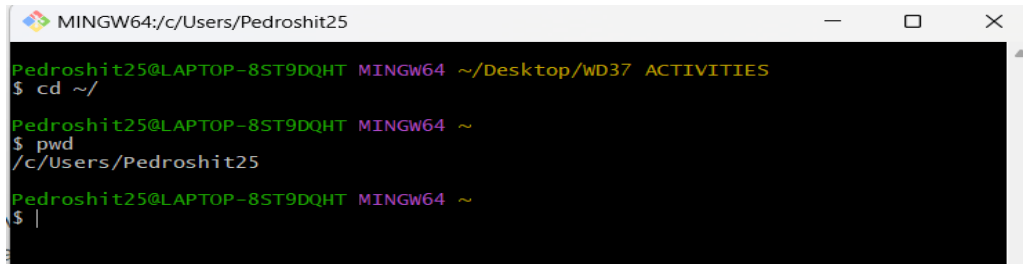


# SSH Key OBSERVATION

## JERALD A. SEGOVIA

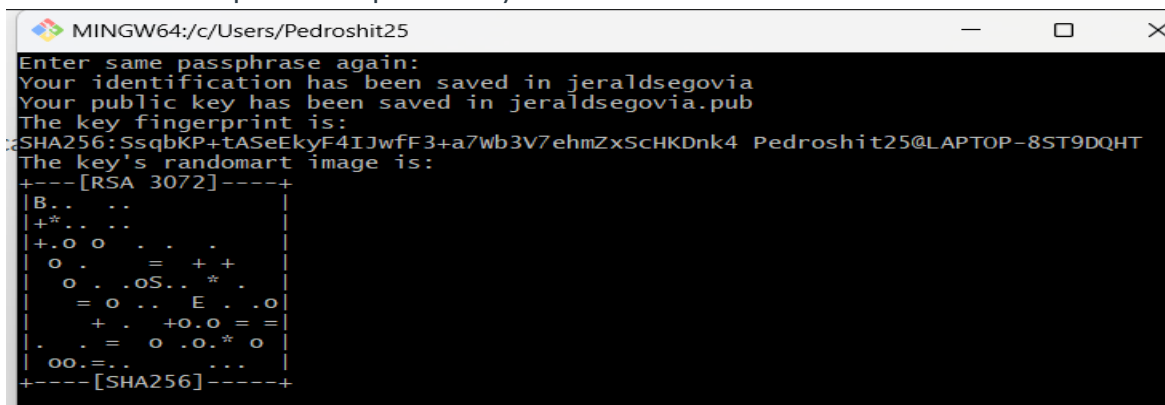
12/07/2022

1. I directly open Gitbash terminal in my target directory



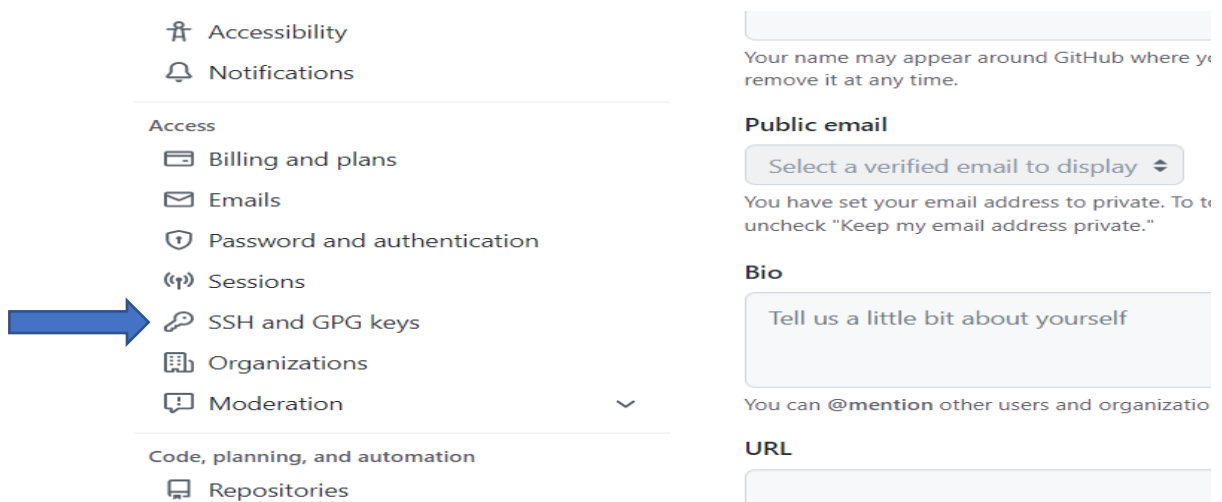
```
MINGW64:/c/Users/Pedroshit25
Pedroshit25@LAPTOP-8ST9DQHT MINGW64 ~/Desktop/WD37 ACTIVITIES
$ cd ~/
Pedroshit25@LAPTOP-8ST9DQHT MINGW64 ~
$ pwd
/c/Users/Pedroshit25
Pedroshit25@LAPTOP-8ST9DQHT MINGW64 ~
$ |
```

2. Use the command "ssh-keygen" to generate the public and private keys. Then it will ask for specific file name you want to use in my case I name the files "jeraldsegovia" which will be the file name of both public and private keys.



```
MINGW64:/c/Users/Pedroshit25
Enter same passphrase again:
Your identification has been saved in jeraldsegovia
Your public key has been saved in jeraldsegovia.pub
The key fingerprint is:
SHA256:SsqbKP+tASeEkyF4IJwFF3+a7Wb3V7ehmZxScHKDnk4 Pedroshit25@LAPTOP-8ST9DQHT
The key's randomart image is:
+---[RSA 3072]-----+
|B.. ..|
|+*.. ..|
|+.o o . . .|
|o . . = + +|
|o . .oS.. * .|
|= o .. E . .o|
| . . +o.o = =|
| . = o .o.* o|
|oo.=.. ...|
+---[SHA256]-----+
```

3. Go to your GitHub account Settings click SSH and GPG keys



#### 4. New SSH key

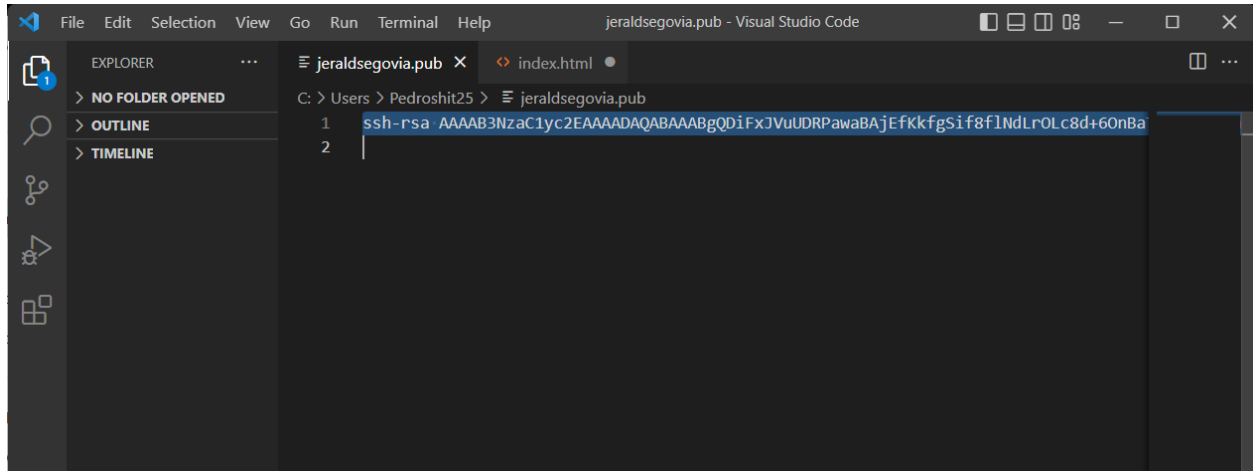
### SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

#### Authentication Keys

#### 5. Open your Public key file to any text editor. Ctrl + A or highlight all Copy



#### 6. Name your SSH Connection. Paste the Public key file in to the key text area. Public key looks like this.

#### SSH Key Add new

### SSH keys / Add new

#### Title

#### Key type

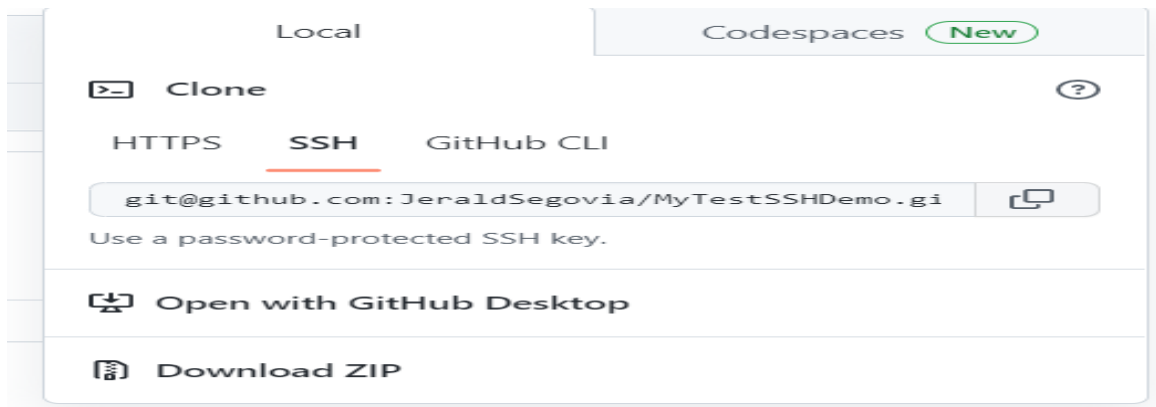
#### Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDDiFvJvUDRPawaBAjEfKkfgSif8f1NdLrOLc8d+60nBaTTJCbyqBnn1AC8SepFrC
ZqONnd20VYSFvez293g00WvZpXJ+/7r5ZzhEgMrFoGYXcgERhKyTXcZ32t2piaRuxbuDYYjqQGBAAiTHSNQ5yvL1HhGO
ZS0vWPA7V8nD/VnnERGHJTscBV0lavz54hdQukYmnWi3yAMJnnojc/RprLCYkPYUOkt0wOBo3M6LXVAoquNbl2/QV8pI7
OFWz06MtCzBllcK2KJ/MCZRR4OioPLyJEmhdSWOknHdIMnldigJTMg1CCO6ODdKPQLUDtPMtN5iHe2HGnGcyADw9ka
Ba+Q0Dga8XLNyfGq0WwUjMnqfzMrWu49vecYIUJ7LdaHP1GAZgDQTPzjZ75uGVPtSr0JBAC5oXT+fvUyL+A95U23tZ
W1mLjwLyrCsFQZ6Xi+bszjk2k9aYm4/xR7+mX2cNjeiDb8thzeAYA5xcNGmyD7zgYgeWM9s5sKTW4XyYqE=
Pedroshitt25@LAPTOP-8ST9DQHT
```

[Add SSH key](#)

7. Add new Repository in Github Click New

8. In your created Repository click Code button then click SSH, copy the link or click the copy button. we need this to check if the SSH connection was successful.



8. Go back to your desktop. Open a New GitBash Terminal.

9. Enable SSH connection by using "eval `ssh-agent`" command.

10. Then we can add now the SSH name using "ssh-add ~/.sshName" in my case "ssh-add ~/.jeraldsegovia"

```
Pedroshit25@LAPTOP-8ST9DQHT MINGW64 ~  
$ eval `ssh-agent`  
Agent pid 1261  
  
Pedroshit25@LAPTOP-8ST9DQHT MINGW64 ~  
$ ssh-add ~/.jeraldsegovia  
Identity added: /c/Users/Pedroshit25/jeraldsegovia (Pedroshit25@LAPTOP-8ST9DQHT)  
  
Pedroshit25@LAPTOP-8ST9DQHT MINGW64 ~  
$
```

11. After that you would see that the SSH identity is added therefore we already established SSH connection.

12. To check the connection is working we can clone the repository by using "git clone [ssh link]" which is the link we copied as per step no. 10.

```
Pedroshit25@LAPTOP-8ST9DQHT MINGW64 ~  
$ eval `ssh-agent`  
Agent pid 1261  
  
Pedroshit25@LAPTOP-8ST9DQHT MINGW64 ~  
$ ssh-add ~/.jeraldsegovia  
Identity added: /c/Users/Pedroshit25/jeraldsegovia (Pedroshit25@LAPTOP-8ST9DQHT)  
  
Pedroshit25@LAPTOP-8ST9DQHT MINGW64 ~  
$ git clone git@github.com:JeraldSegovia/MyTestSSHDemo.git  
Cloning into 'MyTestSSHDemo'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (3/3), done.  
  
Pedroshit25@LAPTOP-8ST9DQHT MINGW64 ~  
$
```

