😊 Git SSH Login

In this guide, we're configuring SSH authentication for Git repositories, focusing on GitHub. The end goal is to enhance security by replacing password-based authentication with SSH keys. By generating SSH keys, adding the public key to GitHub, and using the SSH endpoint to clone repositories, users can securely interact with remote repositories without the risk of exposing passwords. This approach promotes better security practices and safeguards sensitive data during Git operations.

1. In this lab we are going to see git SSH login with the remote repository.
2. If you do cat to this file **.git/config** file, you will see there is an https URL. The authentication will be based on usernames and passwords, which is not a good idea. Then you have to remember the password and store the password, there are chances of exposing the password.

```
$ cat .git/config
[core]
        repositoryformatversion = 0
        filemode = false
        bare = false
        logallrefupdates = true
        symlinks = false
        ignorecase = true
[remote "origin"]
        url = https://github.com/                /titanwork.git
        fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
        remote = origin
        merge = refs/heads/main
```

3. So, GitHub also provides SSH-based login and almost every remote Git repository provides SSH-based authentication, which will be through SSH keys. So first you need to generate SSH keys.
4. You need to write this command to generate public key and private key at this path.

   ssh-keygen.exe

```
$ ssh-keygen.exe
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/PULKIT/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/PULKIT/.ssh/id_ed25519
Your public key has been saved in /c/Users/PULKIT/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:i7ljvwPhO8FSZ/GlSCbARP7eLe37nhXcHPghe5WbAQE PULKIT@LAPTOP-G2CAKBK8
The key's randomart image is:
+--[ED25519 256]--+
|   ++.     E.o.  |
|   .. . +   . o .|
|    .  + + o o =.|
|     .o + o  .+oB|
|     +.+S    .o=o|
|    ..=+ +    .. |
|     .+++ o   .  |
|      =..o   o   |
|     ..+ooo++    |
+----[SHA256]-----+
```

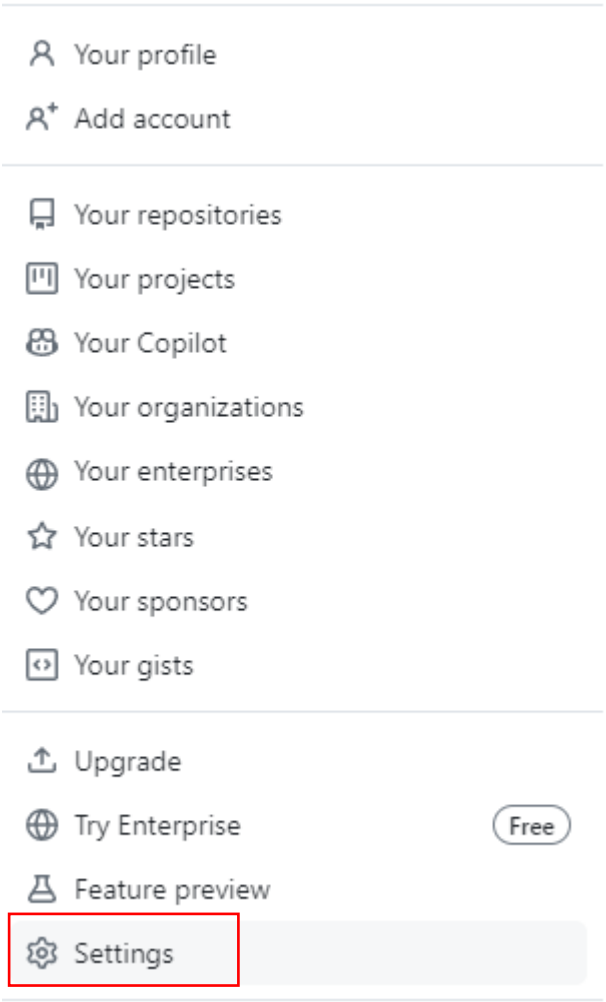
5. Then you have to use these two commands to see your public key. Once you have the public key visible copy the contents and move to GitHub website in your browser.

   ls .ssh/
   cat .ssh/id_ed25519.pub

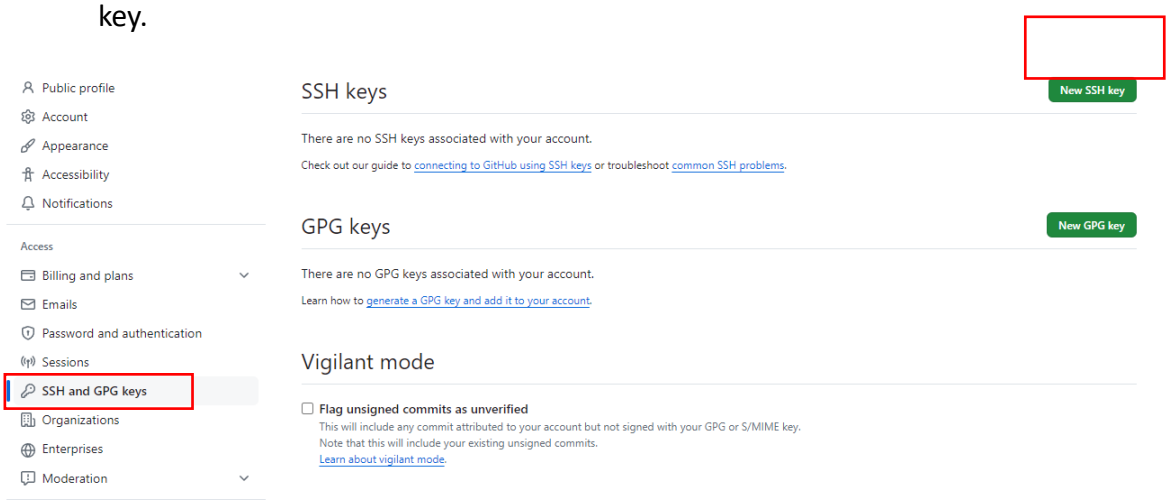
```
PULKIT@LAPTOP-G2CAKBK8 MINGW64 ~
$ ls .ssh/
id_ed25519  id_ed25519.pub

PULKIT@LAPTOP-G2CAKBK8 MINGW64 ~
$ cat .ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOmatCW+7eSh8uu55JJLsrF7itZxSAITJYoKyvMOo5LF PULKIT@LAPTOP-G2CAKBK8
```

6. From the homepage of your GitHub account click on your profile, then a dialog box will appear you have to navigate to settings from here.



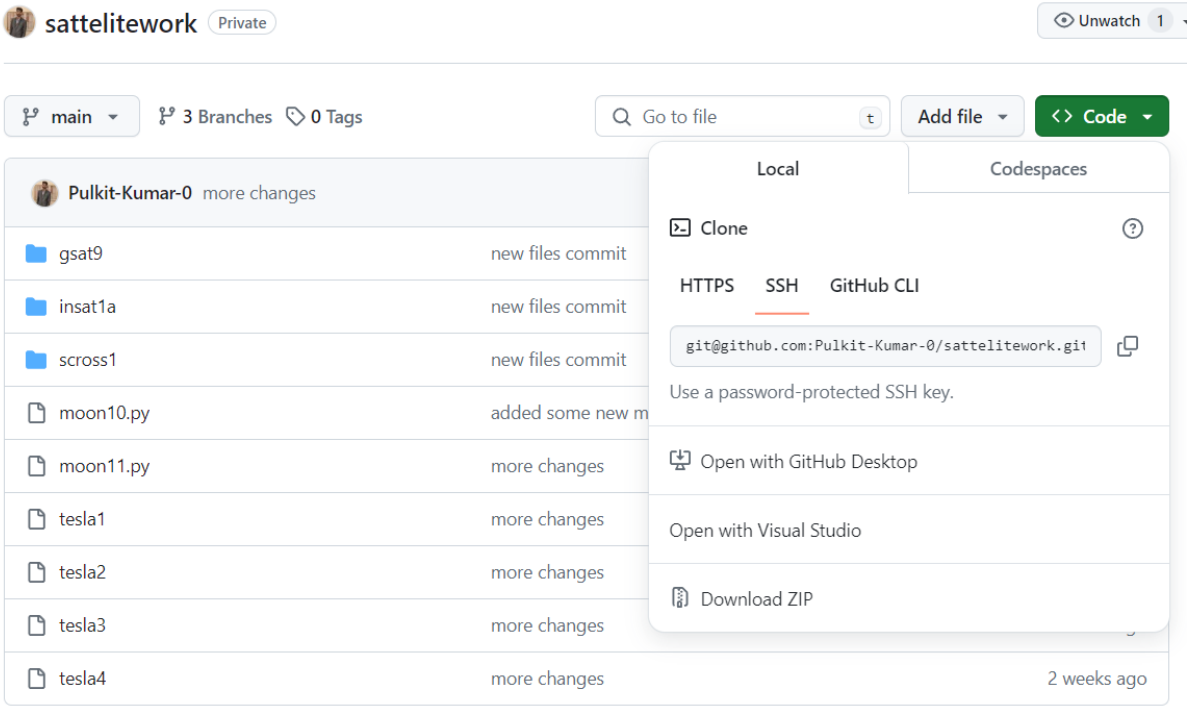
7. Here you have the SSH and GPG keys section, go over there. Then click on new SSH key.



8. Now you need to give it a name to your new SSH key and then paste your key here.

9. Now come back to your git bash and write this command to clone your SSH endpoint.
10. So, you need to copy the SSH endpoint of your private repository and then paste it in your git bash with git clone command.

    git clone git@github.com:Pulkit-Kumar-0/sattelitework.git



11. Then it will ask whether you are sure you want to connect. Say yes and then the cloning will take place.

```
PULKIT@LAPTOP-G2CAKBK8 MINGW64 ~/Desktop/gitrepository
$ git clone git@github.com:Pulkit-Kumar-0/sattelitework.git
Cloning into 'sattelitework'...
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.
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 22 (delta 6), reused 16 (delta 2), pack-reused 0
Receiving objects: 100% (22/22), done.
Resolving deltas: 100% (6/6), done.
```

12. Git SSH login is considered to be safer than HTTPS.