Introduction to Git & GitHub

Distributed Version Control System

Introduction to Git

- Git is a Distributed Version Control tool that supports distributed non-linear workflows by providing data assurance for developing quality software.
- Primarily used to manage project, comprising a set of code/text files that may change.
- There are two types of VCS:
  - Centralized Version Control System (CVCS)
  - Distributed Version Control System (DVCS)

Features of Git

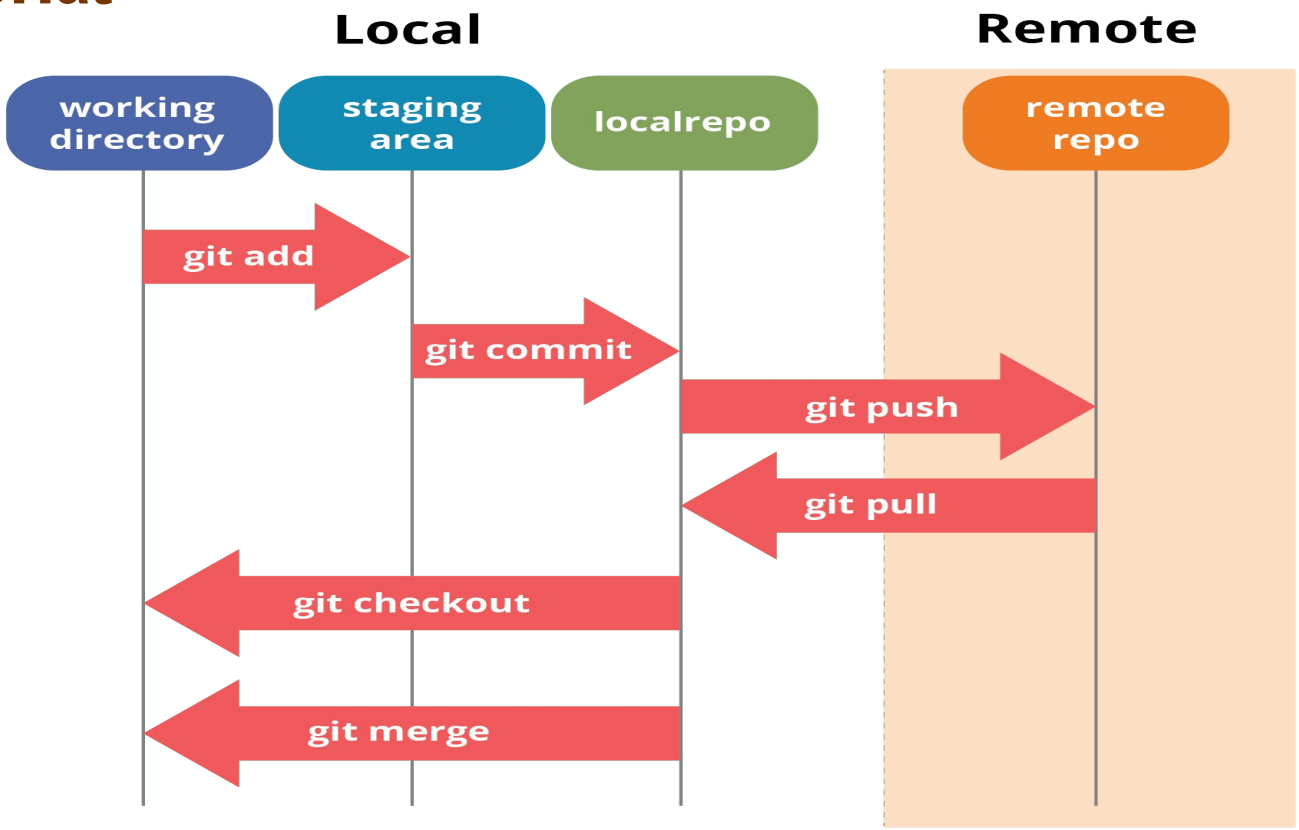
- **Free and open source**: No need to purchase Git. You can modify the source code as per your requirement.
- **Scalable**: As and when the number of collaborators increases, the Git can easily handle this change.
- **Reliable**: Since every contributor has its own local repository, on the events of a system crash, the lost data can be recovered from any of the local repositories.
- **Secure**: Git uses the SHA1 (Secure Hash Function) to name and identify objects within its repository.
- **Supports non-linear development**: Git supports rapid branching and merging.
- **Easy Branching**: Easy to create, delete, and merge branches. Feature branches provide an isolated environment for every change to your codebase.

Git Tutorial

- Some of the basic operations in Git are:
  - Initialize
  - Add
  - Commit
  - Pull
  - Push

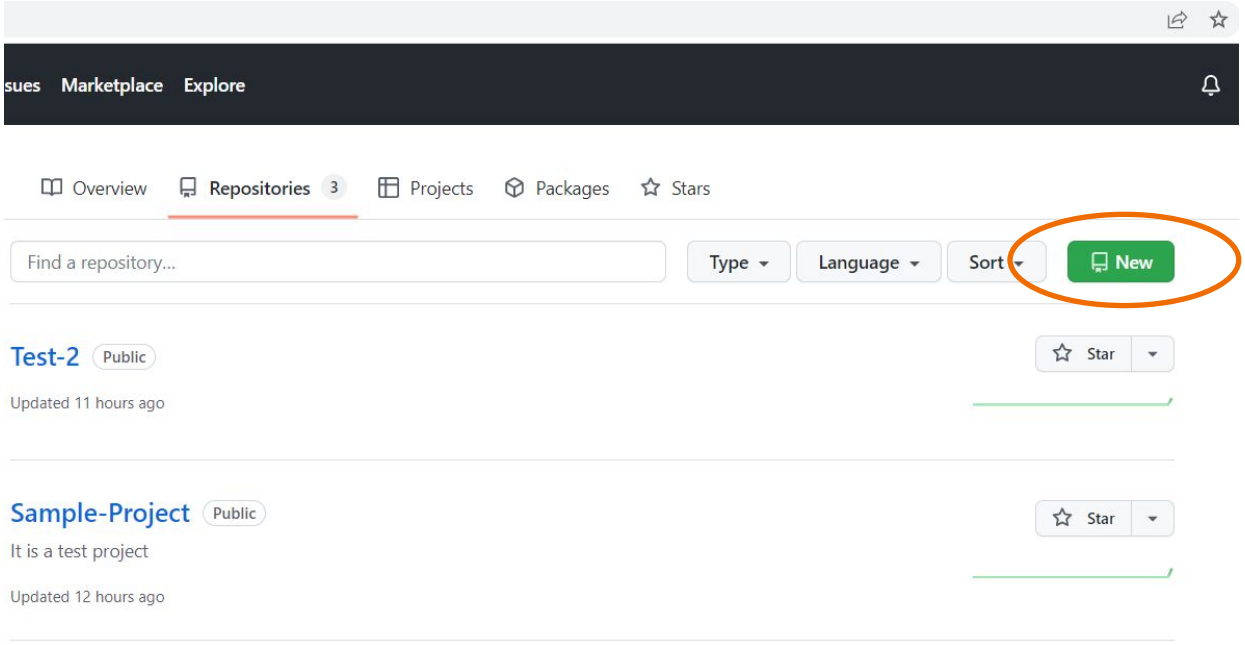
- Some advanced Git operations are:
  - Branching
  - Merging
  - Rebasing

Git Tutorial



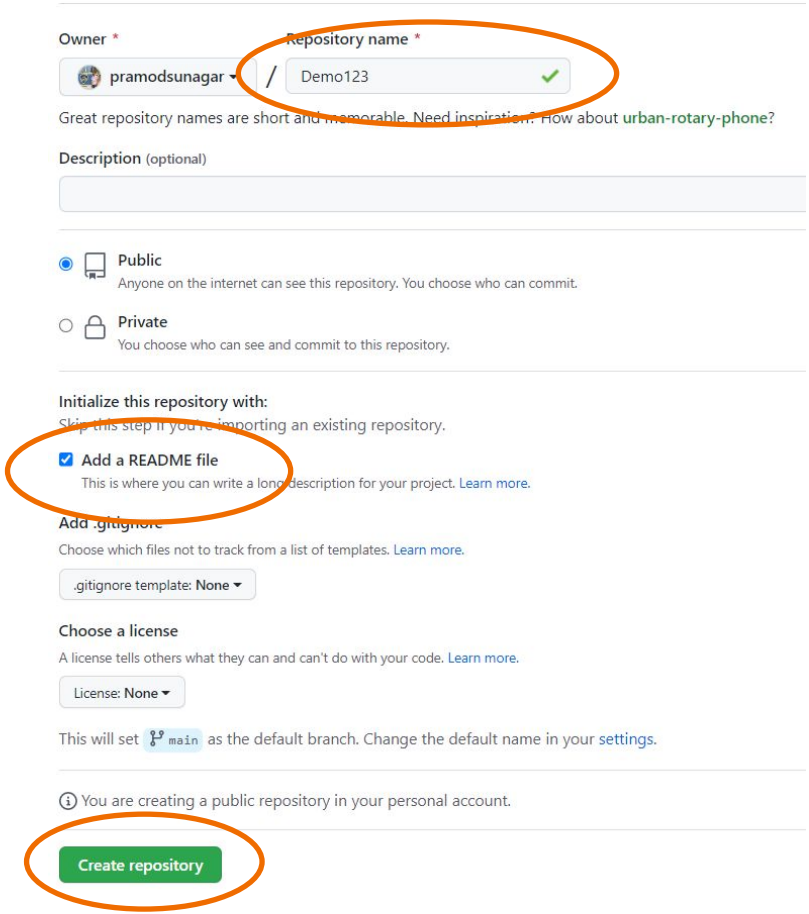
Remote Repository in GitHub

- Create a new repository in GitHub.



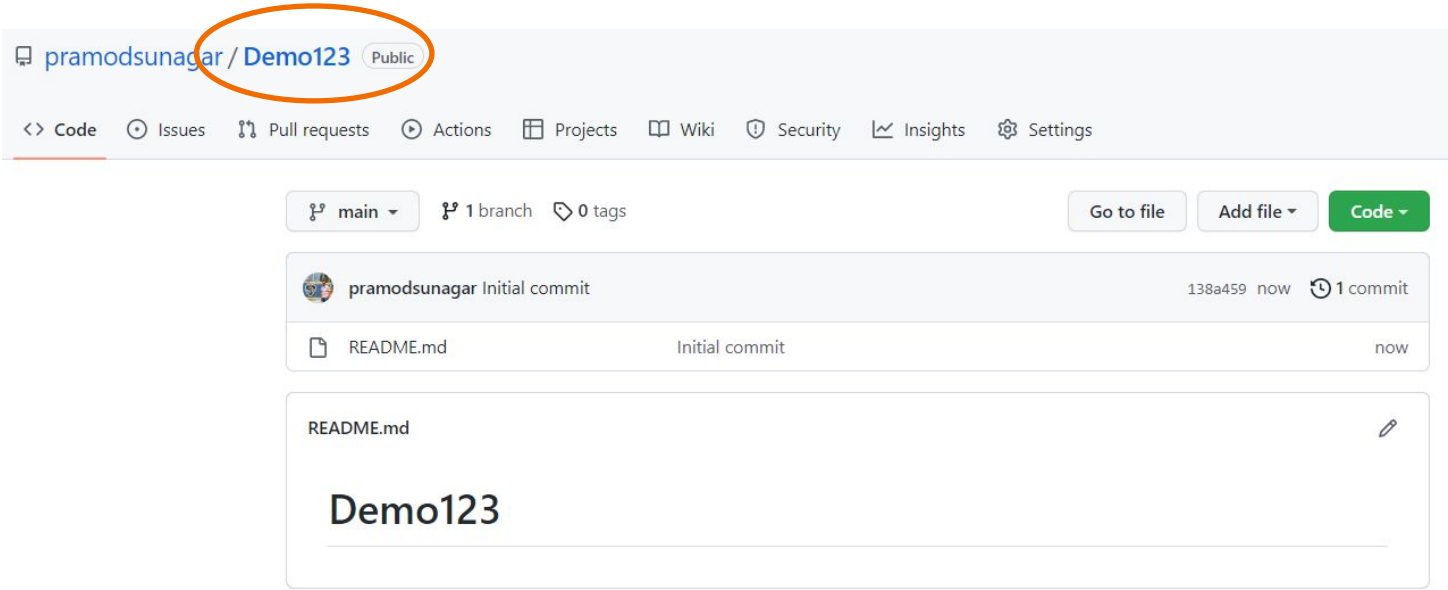
Remote Repository in GitHub

- Add an unique Repository name.
- Select README for any info on the project.
- Choose other settings as per the requirements.
- Click on Create Repository



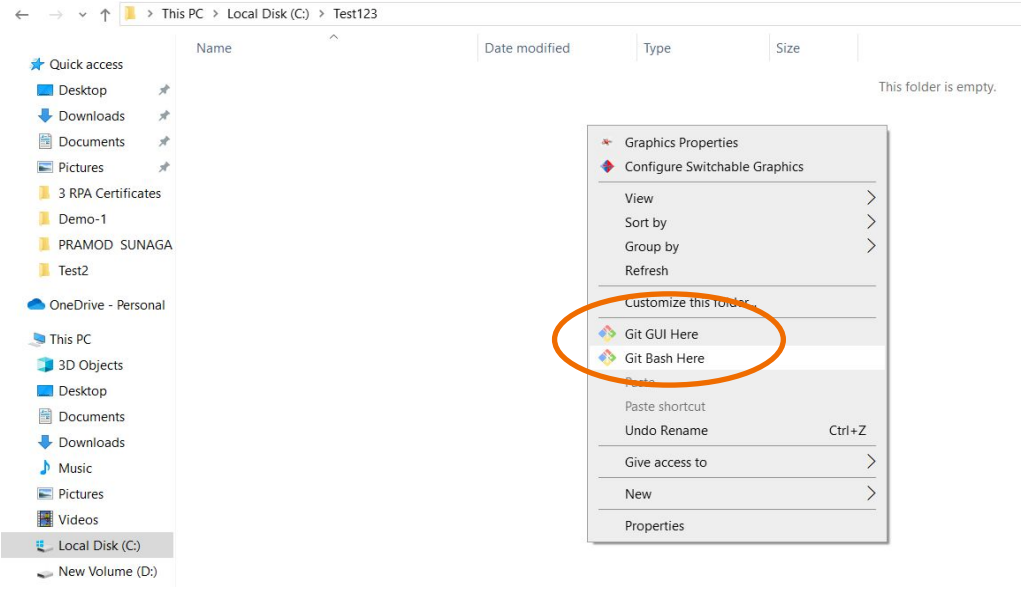
Remote Repository in GitHub

- Remote repository will be created in GitHub.



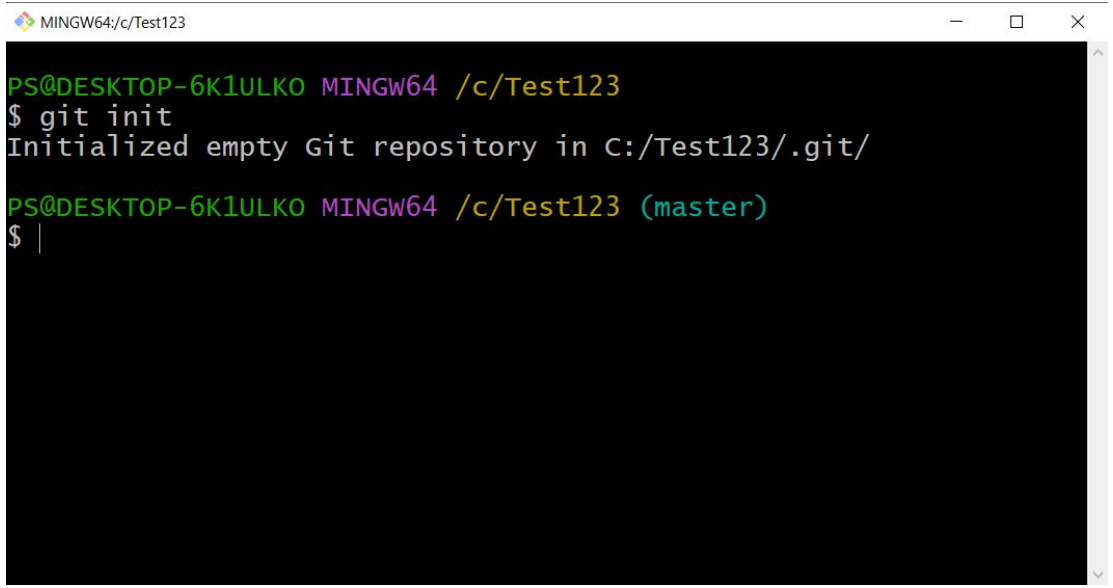
Git Commands

- Download and install the Git from https://gitforwindows.org/
- Create a folder in one drive and right click and select **Git Bash Here**.



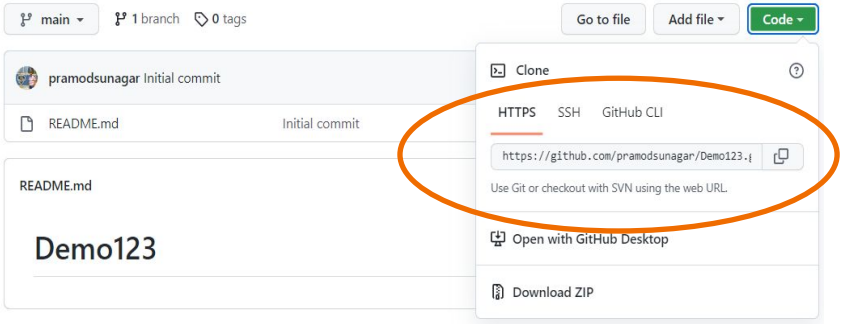
Git Commands

- Initialize: The git init creates an empty Git repository or re-initializes an existing one. It basically creates a .git directory with sub directories and template files.



Git Commands

- Link to remote repo:
- Copy the web URL of the remote repo to link to local repo.
- **git remote add origin "https://github.com/pramodsunagar/Demo123.git"**



Git Commands

- git pull: The **git pull origin main** command syncs all the files from remote repo to local repo.

MINGW64:/c/Test123

```
PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$ git pull origin main
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 596 bytes | 3.00 KiB/s, done.
From https://github.com/pramodsunagar/Demo123
 * branch            main       -> FETCH_HEAD
 * [new branch]      main       -> origin/main

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$ |
```

Git Commands

- git status: The git status command lists all the modified files which are ready to be added to the local repository.

```
MINGW64:/c/Test123

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        test1.txt

nothing added to commit but untracked files present (use "git add" to track)

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$
```

Git Commands

- git add: This command updates the index using the current content found in the working tree and then prepares the content in the staging area for the next commit.

MINGW64:/c/Test123

```
PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$ git add test1.txt

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$ git add -A

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   test1.txt
        new file:   test2.txt
        new file:   test3.txt
```

Git Commands

- git commit: This will commit the staged snapshot.
- git commit –a –m “Message”

MINGW64:/c/Test123

```
PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$ git commit -a -m "First Commit"
[master 234143b] First Commit
 3 files changed, 3 insertions(+)
 create mode 100644 test1.txt
 create mode 100644 test2.txt
 create mode 100644 test3.txt

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$ |
```

Git Commands

- git log

MINGW64:/c/Test123

```
PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$ git log
commit 234143bcbe7fb62f9f5339abf465515f781aef59 (HEAD -> master)
Author: pramodsunagar <pramod.sunagar@gmail.com>
Date:   Mon Aug 22 11:46:05 2022 +0530

    First Commit

commit 138a459da3e1c3d0ae38cf80d1243b787309babe (origin/main)
Author: Pramod Sunagar <71660046+pramodsunagar@users.noreply.github.com>
Date:   Mon Aug 22 11:05:22 2022 +0530

    Initial commit
```

Git Commands

- To create a new branch
- git branch firstbranch
- To move to another branch
- git checkout firstbranch

MINGW64:/c/Test123

```
PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$ git branch firstbranch

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (master)
$ git checkout firstbranch
Switched to branch 'firstbranch'

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (firstbranch)
$ |
```

Git Commands

- To show merging
- Create a file in new branch and commit

MINGW64:/c/Test123

```
nothing added to commit but untracked files present (use "git add" to track)

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (firstbranch)
$ git add new1.txt

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (firstbranch)
$ git status
On branch firstbranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   new1.txt


PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (firstbranch)
$ git commit -m "Adding the new file to the first branch"
[firstbranch f376282] Adding the new file to the first branch
 1 file changed, 1 insertion(+)
 create mode 100644 new1.txt
```

Git Commands

- To show merging
- Go to main branch
- git merge firstbranch

MINGW64:/c/Test123

```
PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (firstbranch)
$ git checkout main
Switched to a new branch 'main'
branch 'main' set up to track 'origin/main'.

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (main)
$ git merge firstbranch
Updating 138a459..f376282
Fast-forward
 new1.txt  | 1 +
 test1.txt | 1 +
 test2.txt | 1 +
 test3.txt | 1 +
 4 files changed, 4 insertions(+)
 create mode 100644 new1.txt
 create mode 100644 test1.txt
 create mode 100644 test2.txt
 create mode 100644 test3.txt

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (main)
$ |
```

Pushing the files to Github repository

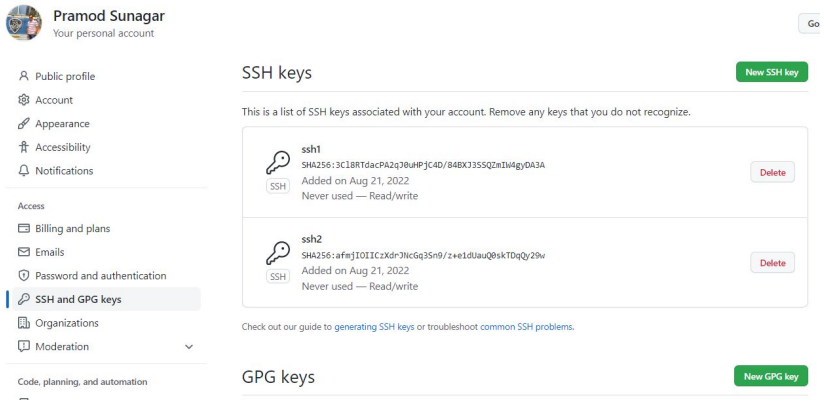
- To sync the files of local repo to remote repo, create a ssh key.
- ssh-keygen

MINGW64:/c/Test123

```
PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (main)
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/PS/.ssh/id_rsa):
/c/Users/PS/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/PS/.ssh/id_rsa
Your public key has been saved in /c/Users/PS/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:3THGnoY2luXvRRObyIRKHa+lExpktUEQHu6+KxKOUJo PS@DESKTOP-6K1ULKO
The key's randomart image is:
+---[RSA 3072]----+
|         **=o.   |
|        =..+* .  |
|        .+.o*= o |
|   .    .ooO=+  =|
|  +     SoB+*  .o|
| E   .  .o o.. . |
|  . o .  .    . .|
|   . o .  .  . . |
|      . .o.   .  |
+----[SHA256]-----+
```

Pushing the files to Github repository

- To add the ssh key go to github account.
- go to setting
- go to SSH and GPG Keys
- Click on new ssh key



- Add title and copy the ssh keys from the command prompt to the ssh key and save.

```
PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (main)
$ cat /c/Users/PS/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDN3EpF40vdmivhk5ZvUT7y21g4FjV4WMF8a1z/Wpx
k5czqqARicCPQV3Tp4IF7NomwKcPG1gkKDhde+H6TbvYW4KgQ1ZwbLDQBBqyuYkMdUcMOfHCId9ko/D
79VsiF9ZeygQOA71ONJ+OXK+/LkbxqH80ejuBZYSg8Pok9myPTbwjDa0uw/aTTRUClMEqhHdi1wdTqm
hB0sPTDB+FTU8oWaktY7GAgGCJzcIftBbZQh+PiPuk9OjUZOvKTgme5WDpwrFm7Oa0zb+EfMcmLmH2N
Hanv5cenm4JBM1gRYIwjPSmjF/jU4FStKnxrzsaHk2s2LegpWzGbhJ+x85Zn0kgYsprWqvxqoCzdB3s
jeXSt0NajbHJco8ozOyZ3QussBzdR64iY71OqwND9r/rz6SoskLeYvABy+1XT+JE5Hu3ZPWUf62N6JX
64vm2zVO83ts6Ef4yHyJArl2BP6VzXtlnjlpXFkUe1M3w2xwhk0sic0rkE7SuMsbvmh5F3Mxe1Cr0=
PS@DESKTOP-6K1ULKO
```

Pushing the files to Github repository

- To authenticate the SSH Key type
- ssh -T git@github.com

MINGW64:/c/Test123

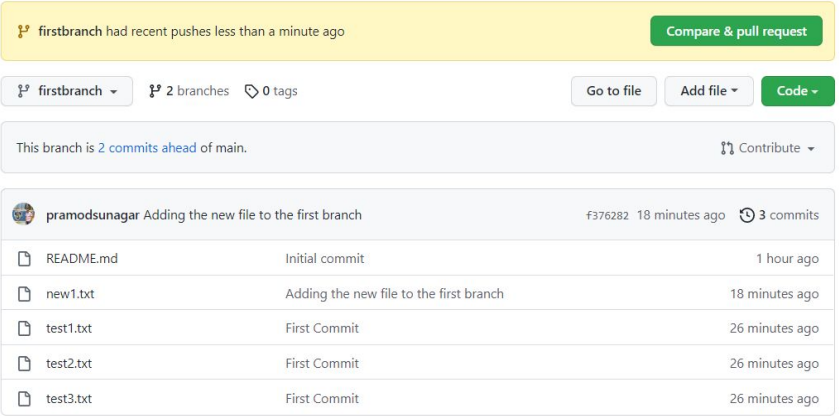
```
PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (main)
$ ssh -T git@github.com
Hi pramodsunagar! You've successfully authenticated, but GitHub does not provid
e shell access.

PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (main)
$
```

Pushing the files to Github repository

- Push all the files
- git push origin firstbranch
- git push origin main

MINGW64:/c/Test123

```
PS@DESKTOP-6K1ULKO MINGW64 /c/Test123 (main)
$ git push origin firstbranch
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 660 bytes | 82.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'firstbranch' on GitHub by visiting:
remote:      https://github.com/pramodsunagar/Demo123/pull/new/firstbranch
remote:
To https://github.com/pramodsunagar/Demo123.git
 * [new branch]      firstbranch -> firstbranch
```



Thank You