Assignment : 08

Title : Deploy a project from local machine to git and vice versa.

About Git :

Git is an open-source version control system (VCS) called Git. Git is responsible for everything Git-Hub related that happens locally on your computer.

The following steps to be followed :

1 -- Creating a new repository in your github account.

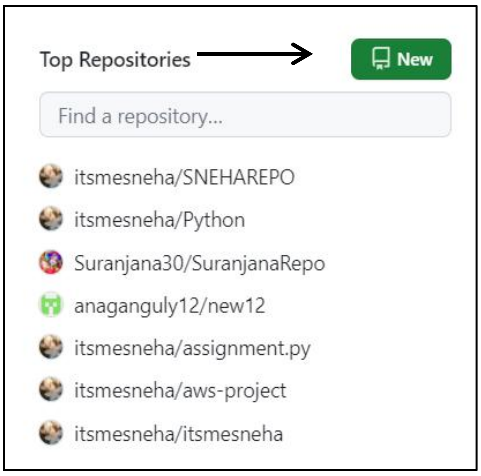
2 -- Generating and storing a token.

3 -- Upload files on github through git bash.

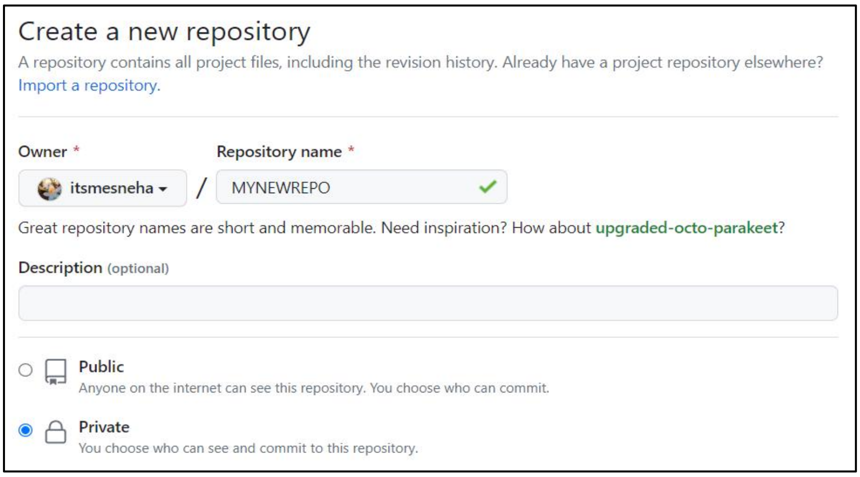
Steps to create a new repository on github :

1. Go to the *Github* and create account .then click on that github logo and in the left side there is a section named *Top Repositories* and click that *New* button to create a repository(***A repository contains all of our project's files and each file's revision history. we can discuss and manage our project's work within the repository***).

2. Enter repository name *MYNEWREPO* and according to user's choice keep repo public or private.(Here we keep it as ***private***)



itsmesneha/SNEHAREPO
itsmesneha/Python
Suranjana30/SuranjanaRepo
anaganguly12/new12
itsmesneha/assignment.py
itsmesneha/aws-project
itsmesneha/itsmesneha



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * itsmesneha / Repository name * MYNEWREPO ✓

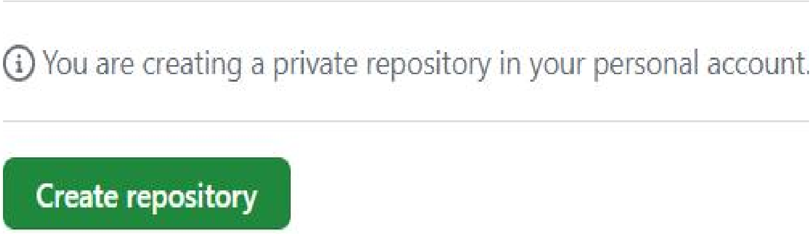
Great repository names are short and memorable. Need inspiration? How about upgraded-octo-parakeet?

Description (optional)

○ Public
Anyone on the internet can see this repository. You choose who can commit.

● Private
You choose who can see and commit to this repository.

3. Now, click create repository and it is created.

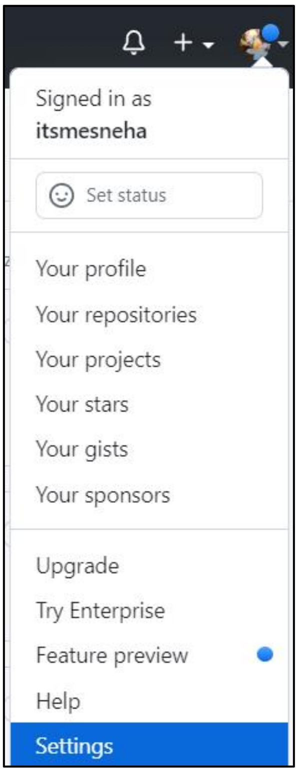


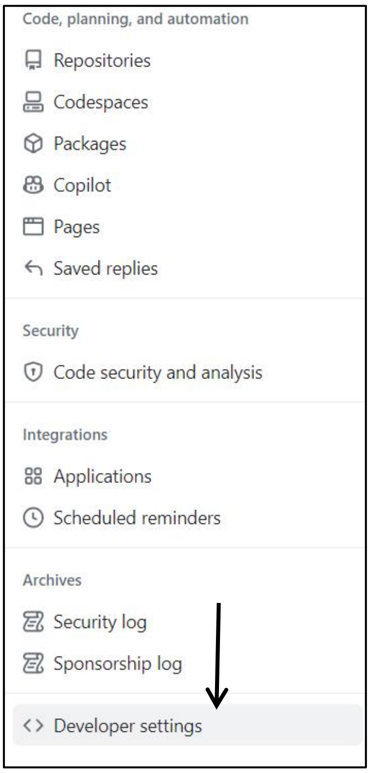
ⓘ You are creating a private repository in your personal account.

After creation of repository we need ***to create a token.*** Tokens are an alternative to using passwords for authentication to GitHub when using the GitHub API or the command line.

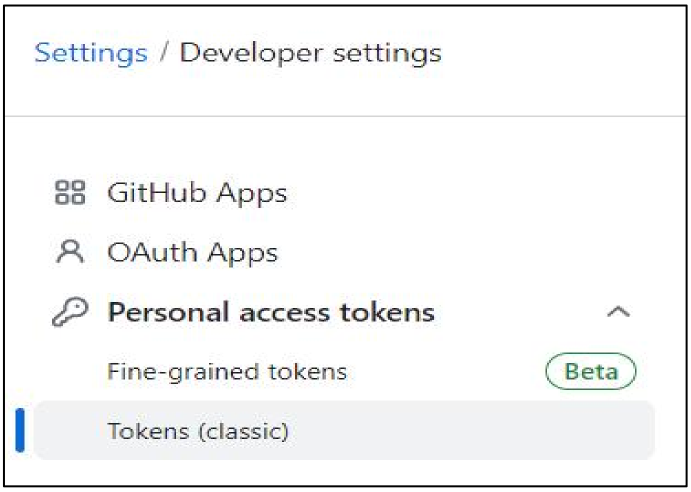
Steps to create token:

1. Goto profile name and click ***Settings*** option.



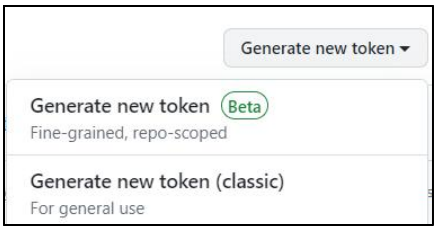


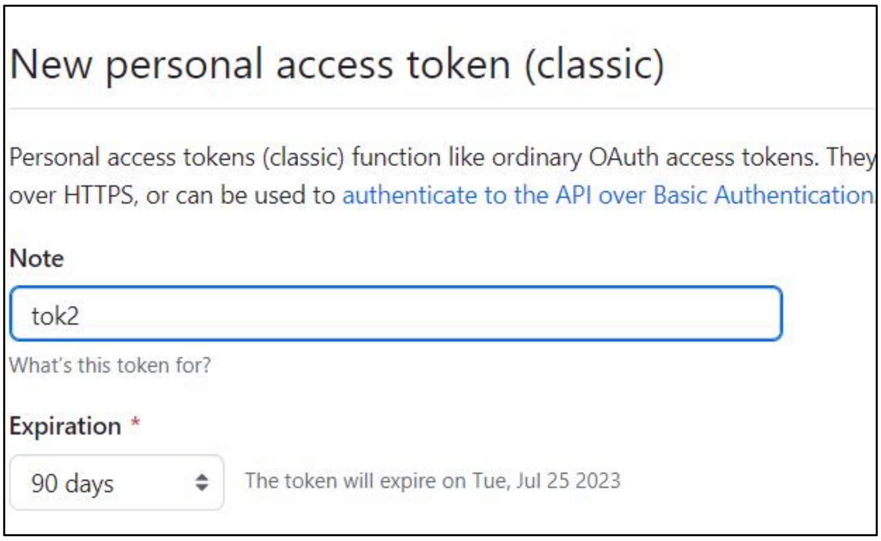
2. Now go to ***Developer Setting*** on the left side. In developer setting click ***Tokens(classic)*** on the left side.



3. Click ***generate new token▼-> generate new token(classic).***

4. Enter token name(like ***tok2***) and set expiration according to user's choice example ***90 days*** and then check all boxes.





New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They over HTTPS, or can be used to authenticate to the API over Basic Authentication

Note

tok2

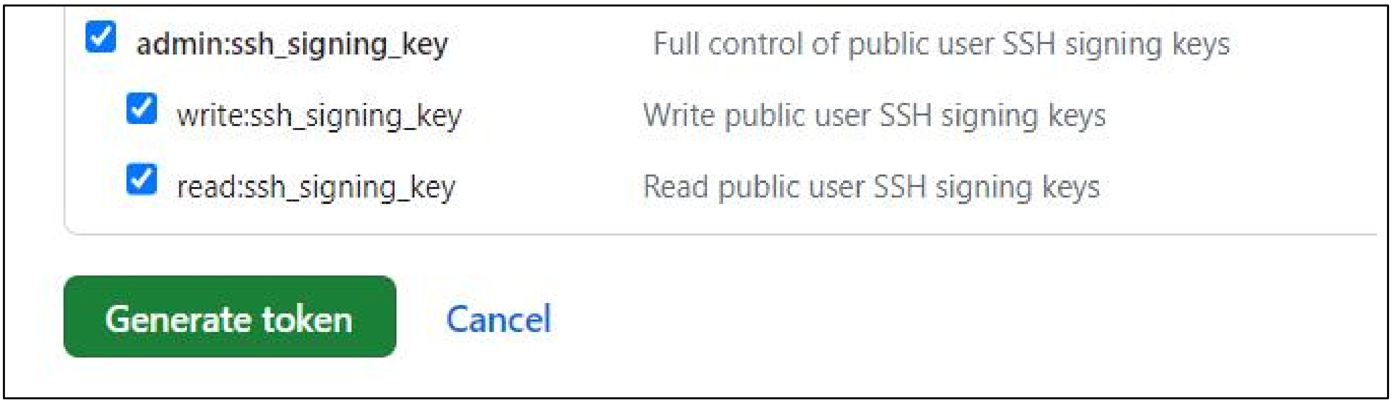
What's this token for?

Expiration *

90 days

The token will expire on Tue, Jul 25 2023

Now click generate token and copy the token for future use.



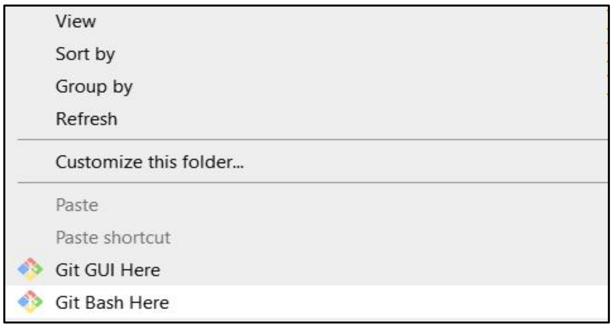
☑ admin:ssh_signing_key — Full control of public user SSH signing keys

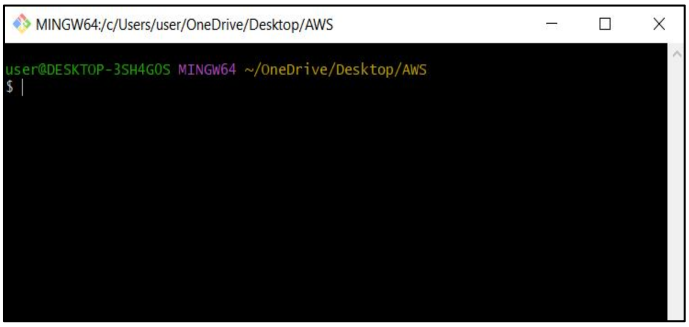
☑ write:ssh_signing_key — Write public user SSH signing keys

☑ read:ssh_signing_key — Read public user SSH signing keys

Steps to Upload files on github through git bash:

1. Before uploading,we need to download ***git for windows.***
2. In desktop click on that folder we want to upload in github.Then right click and select *Git Bash Here.*

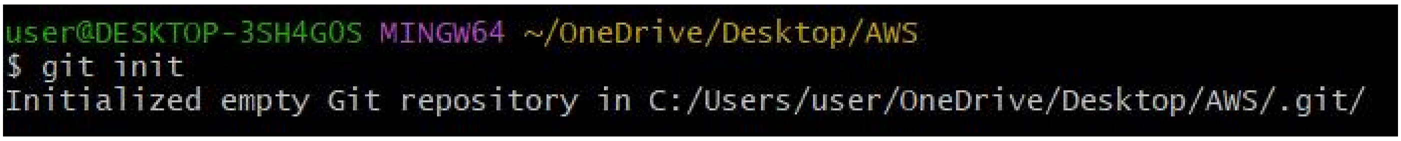




```
user@DESKTOP-3SH4GOS MINGW64 ~/OneDrive/Desktop/AWS
$ |
```

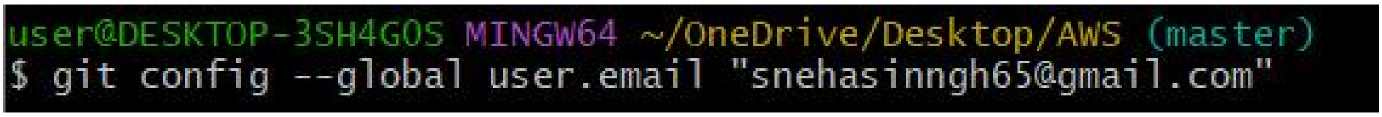
3. In Git Bash give the following command-

✓ *git init* - creates a new Git repository. It can be used to convert an existing, unversioned project to a Git repository or initialize a new, empty repository.



```
user@DESKTOP-3SH4GOS MINGW64 ~/OneDrive/Desktop/AWS
$ git init
Initialized empty Git repository in C:/Users/user/OneDrive/Desktop/AWS/.git/
```

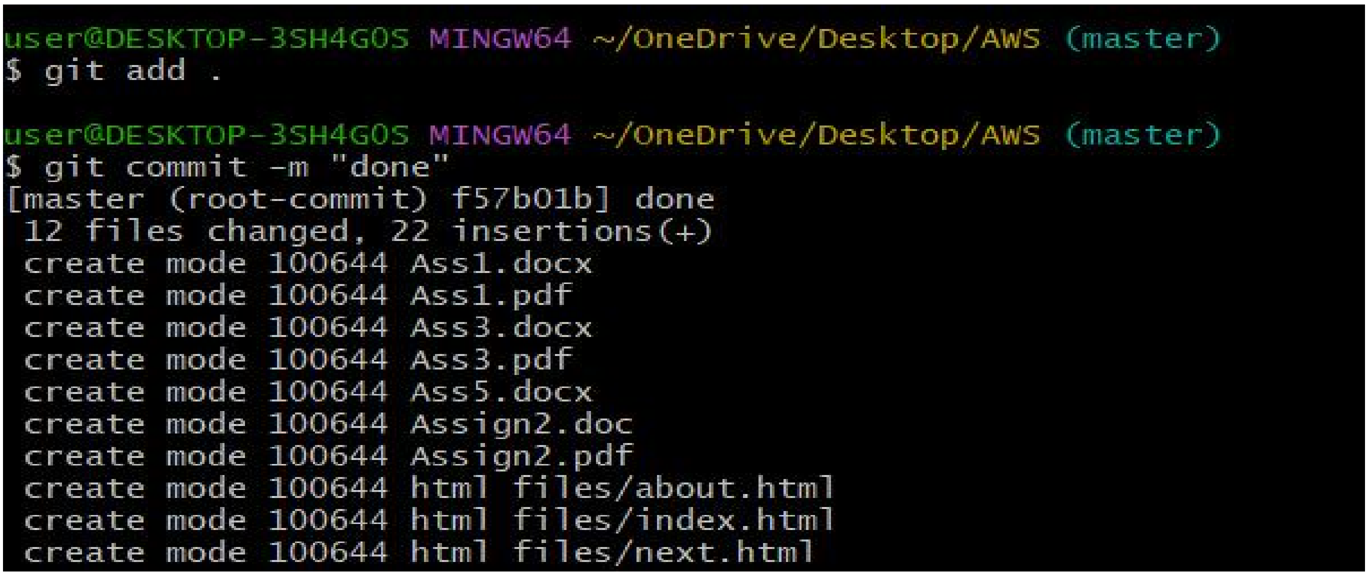
✓ *git config –global user.email "your mail id"* - is used for connecting with our Github account.



```
user@DESKTOP-3SH4GOS MINGW64 ~/OneDrive/Desktop/AWS (master)
$ git config --global user.email "snehasinngh65@gmail.com"
```

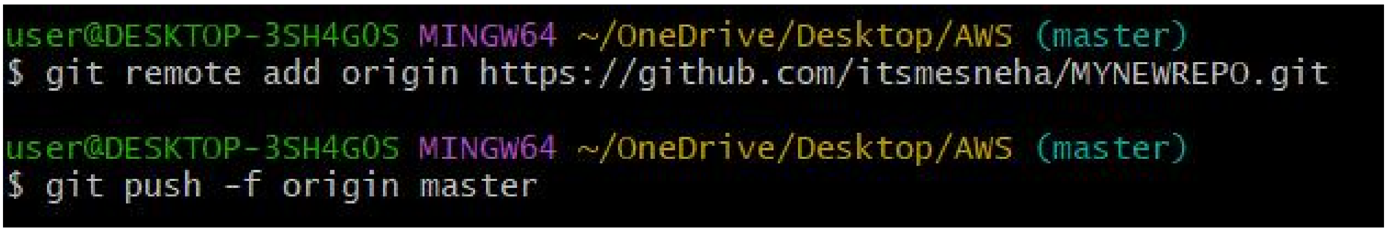
✓ *git add .* - adds a change in the working directory to the staging area. It tells Git that we want to include updates to a particular file in the next commit.

✓ *git commit -m "done"* - is used to commit changes made to a Git repository with a message describing the changes. In this case, the message is "done".



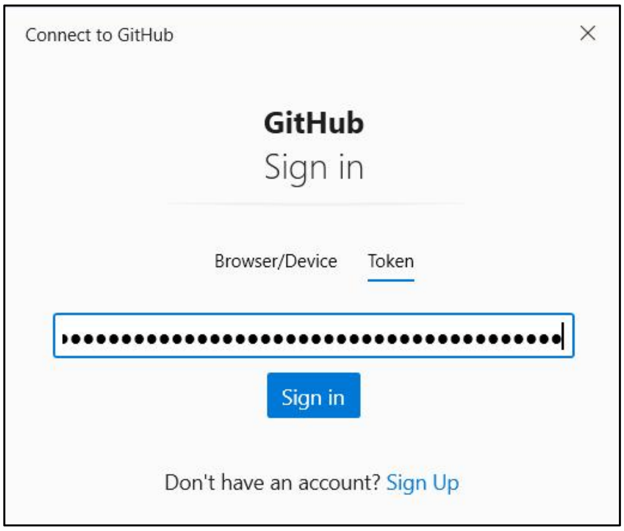
```
user@DESKTOP-3SH4GOS MINGW64 ~/OneDrive/Desktop/AWS (master)
$ git add .

user@DESKTOP-3SH4GOS MINGW64 ~/OneDrive/Desktop/AWS (master)
$ git commit -m "done"
[master (root-commit) f57b01b] done
 12 files changed, 22 insertions(+)
 create mode 100644 Ass1.docx
 create mode 100644 Ass1.pdf
 create mode 100644 Ass3.docx
 create mode 100644 Ass3.pdf
 create mode 100644 Ass5.docx
 create mode 100644 Assign2.doc
 create mode 100644 Assign2.pdf
 create mode 100644 html files/about.html
 create mode 100644 html files/index.html
 create mode 100644 html files/next.html
```

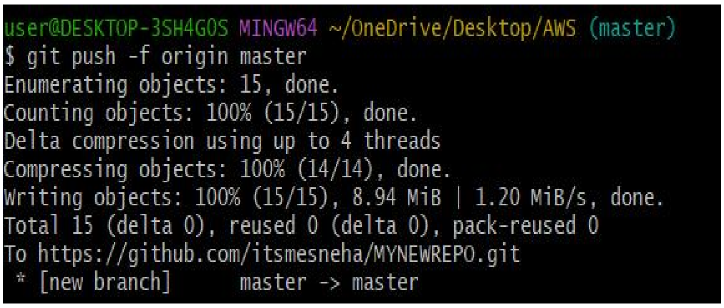
- ✓ *git remote add origin <"remote repository url">* –is used to add a remote repository to a local Git repository. The word "origin" is a conventionally used alias for the remote repository.
- ✓ For example here – *git remote add origin* https://github.com/itsmesneha/MYNEWREPO.git
- ✓ *git push -f origin master* –is used to force push the local branch "master" to the remote repository's branch "master". The "-f" option stands for "force", which means that Git will overwrite the remote branch with your local branch, regardless of whether they have diverged or not.



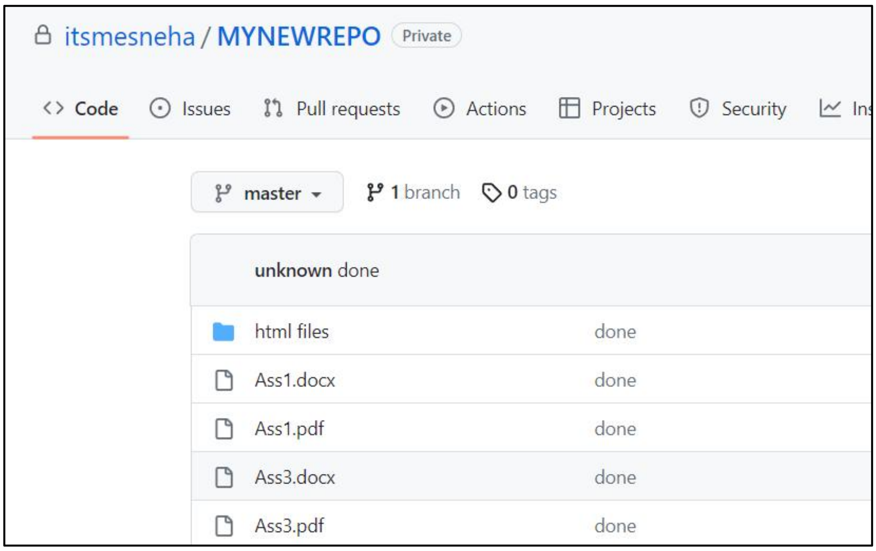
```
user@DESKTOP-3SH4GOS MINGW64 ~/OneDrive/Desktop/AWS (master)
$ git remote add origin https://github.com/itsmesneha/MYNEWREPO.git

user@DESKTOP-3SH4GOS MINGW64 ~/OneDrive/Desktop/AWS (master)
$ git push -f origin master
```

4.After last command a popup window will come where we enter that token(ex-tok2)and it will upload the files from that folder to github repo(ex-MYNEWREPO).





```
user@DESKTOP-3SH4GOS MINGW64 ~/OneDrive/Desktop/AWS (master)
$ git push -f origin master
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (15/15), 8.94 MiB | 1.20 MiB/s, done.
Total 15 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/itsmesneha/MYNEWREPO.git
 * [new branch]      master -> master
```



html files	done
Ass1.docx	done
Ass1.pdf	done
Ass3.docx	done
Ass3.pdf	done

As we can see all the files from that folder of AWS is uploaded to github repo (e.g.- **MYNEWREPO**).

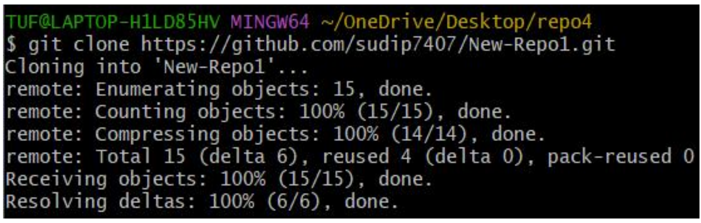
It includes all the files contained in the folder. The token was used to log in to the github account.

Steps to upload from repository to local machine to another repository:

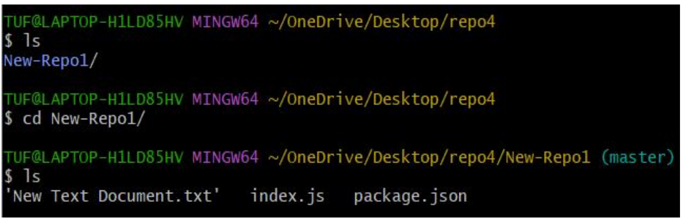
We have a repo named https://github.com/sudip7407/New-Repo1.git. We have to save files to our local machine from this repo.

1. Create a folder in the desktop(here ***folder2***).
2. Now right click and select ***Git Bash Here***.
3. And run following commands- ***git clone <remote repository URL>*** -is used to create a local copy of a remote Git repository. The command creates a complete copy of the remote repository, including all of its branches, tags, and commit history.

- ✓ for checking type ***ls*** and we can see that created repo.
- ✓ Then go to that repo by giving the command ***cd <folder name>***.
- ✓ then in that folder type ***ls*** and we can see that files which is copied.



```
TUF@LAPTOP-H1LD85HV MINGW64 ~/OneDrive/Desktop/repo4
$ git clone https://github.com/sudip7407/New-Repo1.git
Cloning into 'New-Repo1'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 6), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (6/6), done.
```



```
TUF@LAPTOP-H1LD85HV MINGW64 ~/OneDrive/Desktop/repo4
$ ls
New-Repo1/

TUF@LAPTOP-H1LD85HV MINGW64 ~/OneDrive/Desktop/repo4
$ cd New-Repo1/

TUF@LAPTOP-H1LD85HV MINGW64 ~/OneDrive/Desktop/repo4/New-Repo1 (master)
$ ls
'New Text Document.txt'   index.js   package.json
```

Now we can upload this files by running the previous steps (except token creation as we have it)and files are uploaded on the repo.