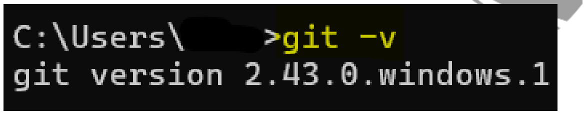
USING GIT TO PUBLISH FILES TO A GITHUB REPOSITORY

• What is Git?

Git is a distributed version control system (DVCS) used for tracking changes in source code during software development. It was created by Linus Torvalds in 2005 to manage the development of the Linux kernel, and since then, it has become one of the most popular version control systems in the world.

• Installing Git into your system

- Install Git at as per requirements: https://git-scm.com/downloads
- Check version at Terminal / CMD: *git -v*



```
C:\Users\[redacted]>git -v
git version 2.43.0.windows.1
```

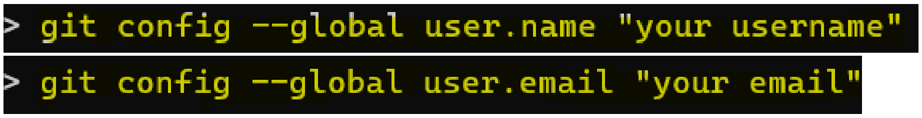
Figure 1:Checking if git is installed

• Configuring Git for GitHub into your system

!** Make sure you have your GitHub username and email with you**!

Use following syntax in your terminal / CMD sequentially:

- *git config --global user.name "your_username"*
- *git config –global user.email "your_email"*



```
> git config --global user.name "your username"
> git config --global user.email "your email"
```

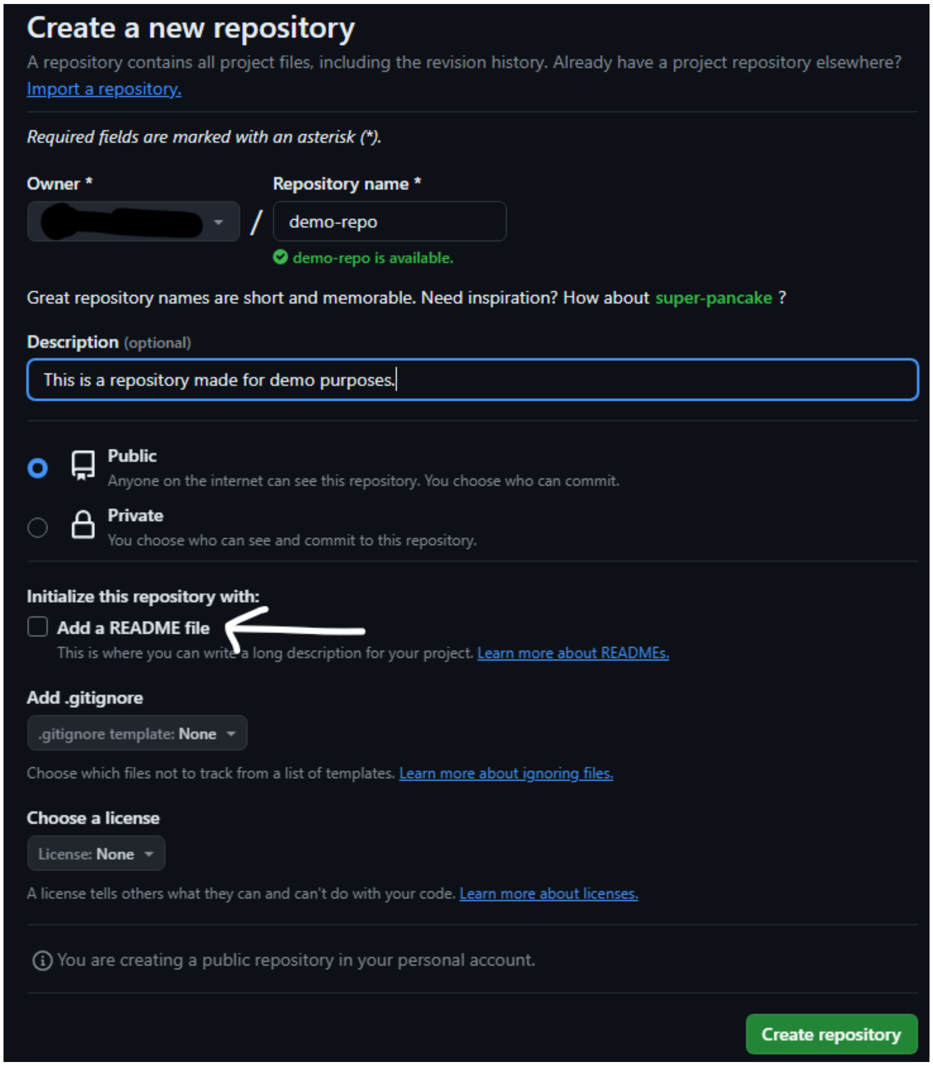
Figure 2: Configuring git to access GitHub

- Check if the username and email is correct using following syntax individually: *"--global user.name" & "--global user.email"*
  ***!** You may be navigated to your browser > GitHub login page to confirm your email and password!*

- ## Adding files into your GitHub Repo

----YOUR GITHUB PART----

- Firstly, make a GitHub repository and make sure that "README" files aren't checked.



Create a new repository

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

Required fields are marked with an asterisk (*).

Owner * [redacted] / Repository name * demo-repo

demo-repo is available.

Great repository names are short and memorable. Need inspiration? How about super-pancake ?

Description (optional)

This is a repository made for demo purposes.

● 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.

Initialize this repository with:

□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

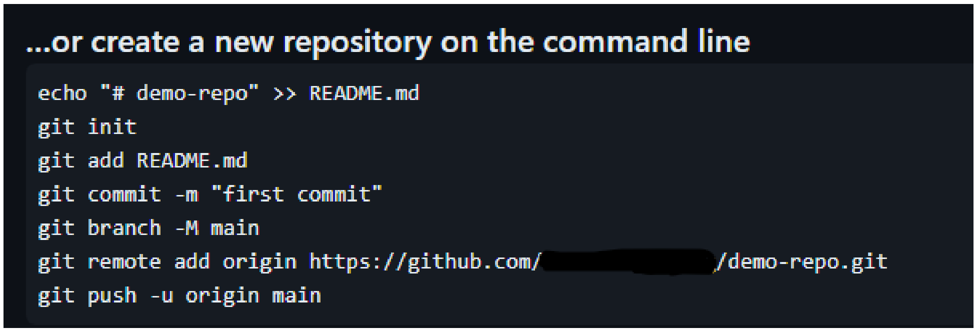
License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a public repository in your personal account.

Figure 3: Creating GitHub Repository

- Then you'll head on to this page (this is visible only when the above isn't checked):



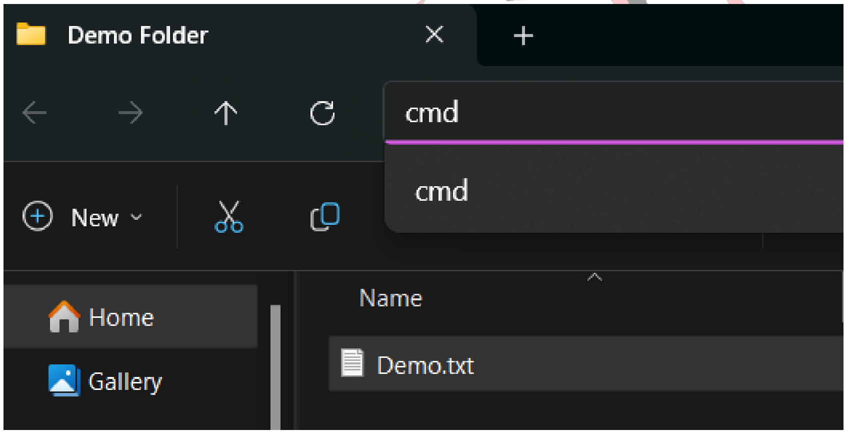
...or create a new repository on the command line

```
echo "# demo-repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/[redacted]/demo-repo.git
git push -u origin main
```

Figure 4: Commands to push files

----YOUR SYSTEM'S PART----

- Now, choose a folder containing files of your choice
- Type CMD at the search bar of your folder for quick access to CMD or navigate to your folder using your system's terminal / CMD syntaxes.



- At the terminal type the commands sequentially.

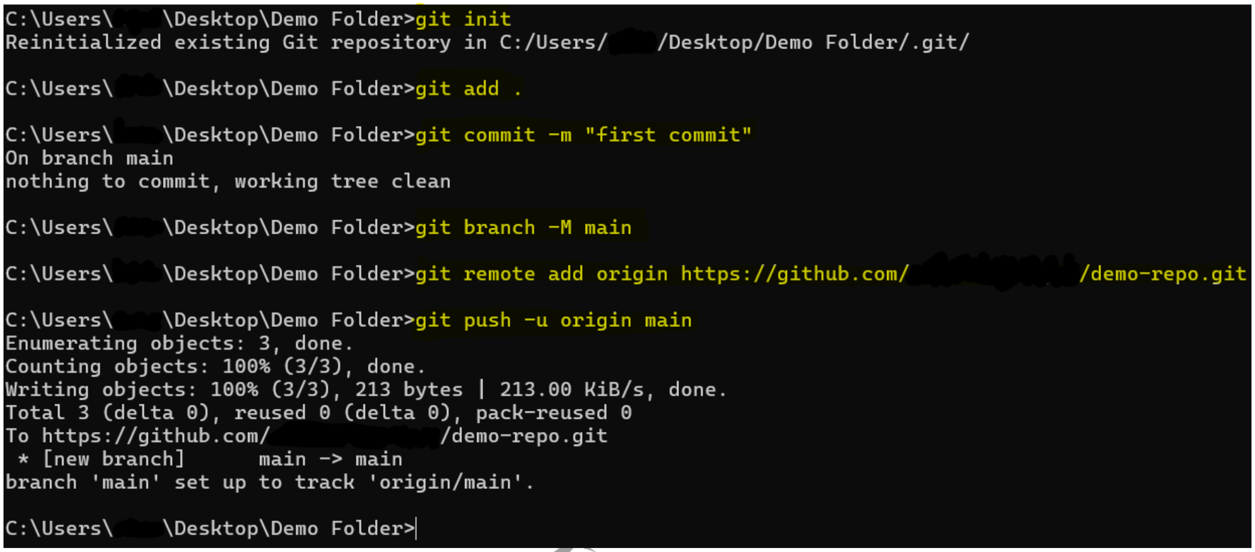
!!

You may or may not use the first command (it is used just to add a readme file)

Replace "*git add README.md*" with "*git add .*", in order to add all files that are in your folder

You may at anything inside the 4th syntax’s (“”) e.g. (“first initial upload ”)

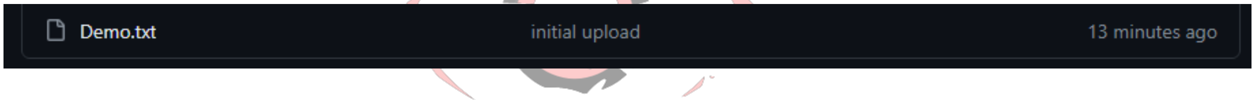
!!



```
C:\Users\[redacted]\Desktop\Demo Folder>git init
Reinitialized existing Git repository in C:/Users/[redacted]/Desktop/Demo Folder/.git/

C:\Users\[redacted]\Desktop\Demo Folder>git add .

C:\Users\[redacted]\Desktop\Demo Folder>git commit -m "first commit"
On branch main
nothing to commit, working tree clean

C:\Users\[redacted]\Desktop\Demo Folder>git branch -M main

C:\Users\[redacted]\Desktop\Demo Folder>git remote add origin https://github.com/[redacted]/demo-repo.git

C:\Users\[redacted]\Desktop\Demo Folder>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 213 bytes | 213.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/[redacted]/demo-repo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

C:\Users\[redacted]\Desktop\Demo Folder>
```

Figure 5: Uploading the File

- Your GitHub view:



Demo.txt initial upload 13 minutes ago

Figure 6: After upload

- **Benefits of doing this instead of uploading files manually**

- It enables you to upload large files e.g. If you have a React JS app and want to upload it to your GitHub, uploading it manually is difficult or sometimes isn’t possible. But, using git makes the process easier and possible.
- You can use git in your text editor’s terminal e.g. VS Code.
- Uploading new files as you make more files becomes much easier.
- You can even make different branches in order to make sure your previous files aren’t affected by the new ones.
- And many more.

- ## How to add new files to the same repo.

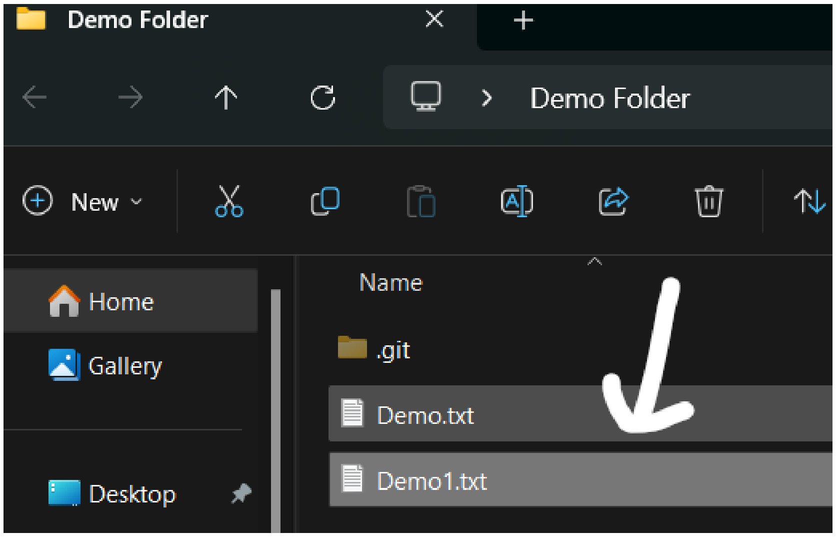
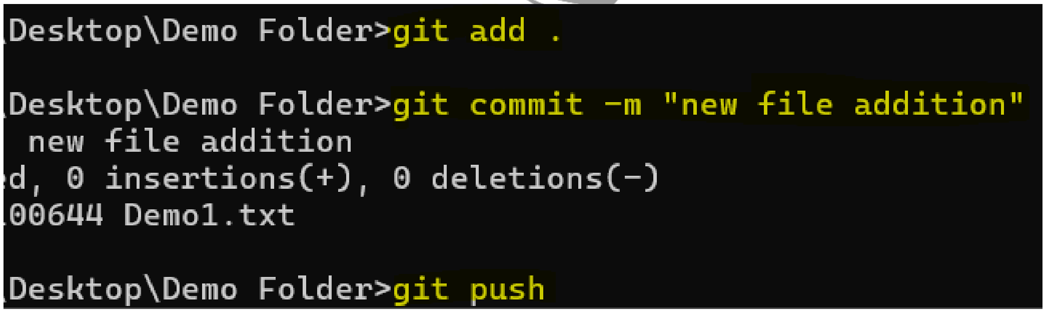


Figure 7: Addition of new file (Demo1.txt)

Doing this is extremely easy. Go to your CMD and access the same folder and use following syntaxes.

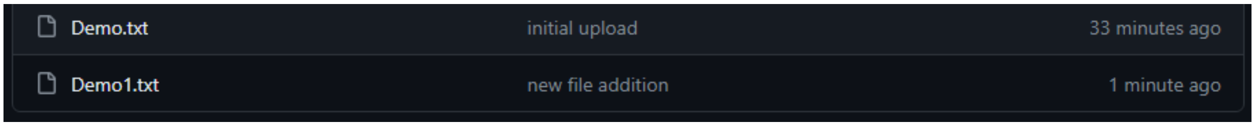
- *git add .*
- *git commit -m "new file addition"*
- *git push*



```
Desktop\Demo Folder>git add .

Desktop\Demo Folder>git commit -m "new file addition"
 new file addition
d, 0 insertions(+), 0 deletions(-)
00644 Demo1.txt

Desktop\Demo Folder>git push
```

Figure 8: Addition of new files into the same repo

- Your latest GitHub view:



Demo.txt	initial upload	33 minutes ago
Demo1.txt	new file addition	1 minute ago

Figure 9: GitHub view after addition of new files

KEEP IN MIND THAT GIT IS NOT JUST FOR GITHUB