

YUVARAJ KUMAR S

21-07-2025

Git Commands

1. Open Git bash shell and create a new folder “**GitDemo**” and initialize as git repository
2. Git bash initializes the “**GitDemo**” repository.

```
yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo (master)
$ mkdir Gitdemo

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo (master)
$ cd Gitdemo

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (master)
$ git init
Initialized empty Git repository in C:/Users/yuvaraj.k/Desktop/Day11_Daily_Task/GitDemo/Gitdemo/.git/

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (master)
$
```

3. Create a file “**welcome.txt**” and add content to the file (use echo or cat command)

```
yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (master)
$ echo "This is my text content" > Welcome.txt

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (master)
$ cat welcome.txt
This is my text content
```

4. To verify if the file “welcome.txt” is created, execute ls-al

```
yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (master)
$ ls -al
total 5
drwxr-xr-x 1 yuvaraj.k 1049089  0 Jul 21 16:46 ./
drwxr-xr-x 1 yuvaraj.k 1049089  0 Jul 21 16:41 ../
drwxr-xr-x 1 yuvaraj.k 1049089  0 Jul 21 16:41 .git/
-rw-r--r-- 1 yuvaraj.k 1049089 24 Jul 21 16:46 Welcome.txt
```

5. To verify the content, execute the command `cat filename`

```
yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (master)
$ cat welcome.txt
This is my text content
```

6. Execute `git status`

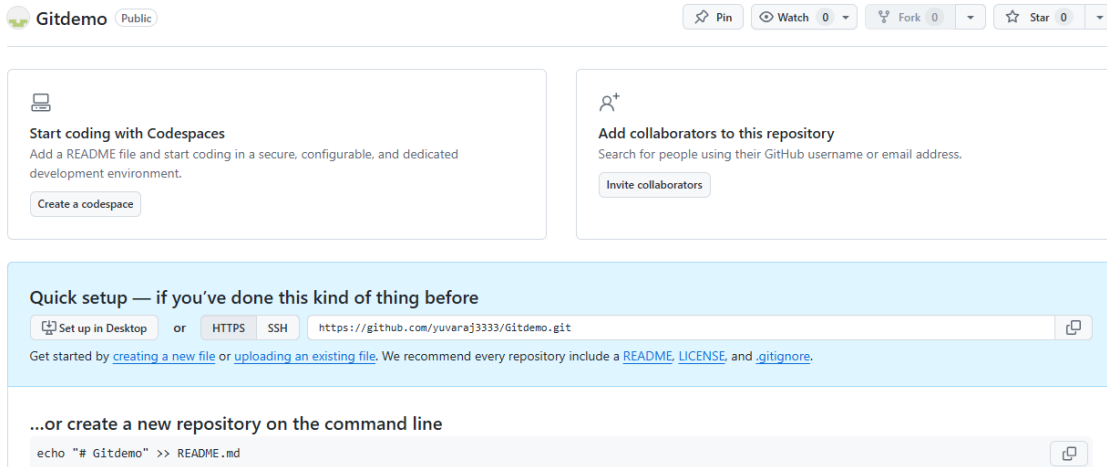
```
yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Welcome.txt

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

7. Signup with GitHub and create a remote repository **"GitDemo"**



8. Add remote origin

```
yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (master)
$ git remote add origin https://github.com/yuvaraj3333/Gitdemo.git
```

9. Create files and push the local to remote repository.

```

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (master)
$ git branch -M main

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Welcome.txt

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/yuvaraj3333/Gitdemo.git'

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ git commit -m "first commit"
[main (root-commit) 45db0fd] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 Welcome.txt

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 236 bytes | 118.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/yuvaraj3333/Gitdemo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

```

10. Make changes in the remote and pull the changes to the local.

```

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 905 bytes | 27.00 KiB/s, done.
From https://github.com/yuvaraj3333/Gitdemo
 * branch            main      -> FETCH_HEAD
   45db0fd..ca0a10d  main      -> origin/main
Updating 45db0fd..ca0a10d
Fast-forward
 Welcome.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

```

11. Create a .gitignore file to ignore contents of temp folder

```
yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ echo "temp/" >.gitignore

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ cat .gitignore
temp/

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

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

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ git commit ".gitignore file created"
error: pathspec '.gitignore file created' did not match any file(s) known to git

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ git commit -m ".gitignore file created"
[main 6107ec6] .gitignore file created
1 file changed, 1 insertion(+)
create mode 100644 .gitignore

yuvaraj.k@PTPLL403 MINGW64 ~/Desktop/Day11_Daily_Task/GitDemo/Gitdemo (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 293 bytes | 97.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/yuvaraj3333/Gitdemo.git
ca0a10d..6107ec6  main -> main
```

Docker

1. What is Docker-compose.yml file? Give sample docker-compose file.

Gitdemo / docker-compose.yml

yuvaraj3333 Docker compose file created d93d17b · now History

Code Blame 18 lines (18 loc) · 291 Bytes

```
1 version: '3'
2 services:
3   web:
4     image: nginx:latest
5     ports:
6       - "8080:80"
7   app:
8     build: ./app
9     ports:
10      - "5000:5000"
11     depends_on:
12       - db
13   db:
14     image: postgres:13
15     environment:
16       POSTGRES_USER: user
17       POSTGRES_PASSWORD: pass
18       POSTGRES_DB: mydb
```

We are using nginx, ./app and then for database Postgres.

2. What is Docker hub?

Docker Hub is a web service that provides a **centralized library for container images**, allowing users to **discover, manage, and deploy pre-built Docker images**. It simplifies the process of sharing containerized applications and helps automate the workflow from development to production by offering both public and private repositories.