

Karnish N
21/07/2025

1. OpenGit bash shell and create a new folder “**GitDemo**” and initialize as git repository

2. Git bash initializes the “**GitDemo**” repository.

```
karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo (master)
$ mkdir GitDemo

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo (master)
$ cd GitDemo

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (master)
$ git init
Initialized empty Git repository in C:/Users/karnish.n/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo/.git/
C
```

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

```
karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (master)
$ echo "This is my text content" > Welcome.txt

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_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

```
karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (master)
$ ls -al
total 5
drwxr-xr-x 1 karnish.n 1049089  0 Jul 21 17:41 ./
drwxr-xr-x 1 karnish.n 1049089  0 Jul 21 17:40 ../
drwxr-xr-x 1 karnish.n 1049089  0 Jul 21 17:41 .git/
-rw-r--r-- 1 karnish.n 1049089 24 Jul 21 17:41 Welcome.txt
```

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

```
karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (master)
$ cat welcome.txt
This is my text content
```

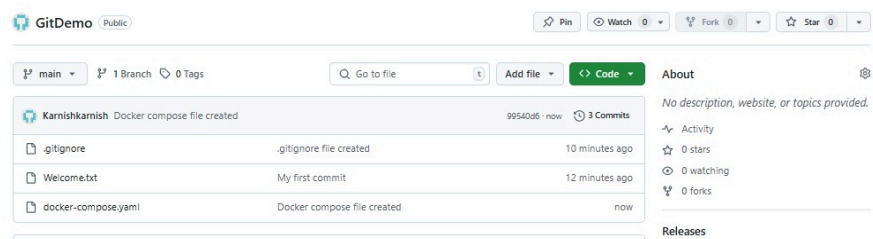
6. Execute `git status`

```
karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_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
```

7. Signup with GitHub and create a remote repository “GitDemo”



8. Add remote origin

```
karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (master)
$ git remote add origin https://github.com/Karnishkarnish/GitDemo.git
```

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

```
karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (master)
$ git branch main
fatal: not a valid object name: 'master'

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (master)
$ git branch -m
fatal: branch name required

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (master)
$ git branch -M main

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_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

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (main)
$ git commit -m "My first commit"
[main (root-commit) e016cf4] My first commit
 1 file changed, 1 insertion(+)
 create mode 100644 Welcome.txt

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_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), 240 bytes | 80.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Karnishkarnish/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.

```
karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (main)
$ git pull origin main
From https://github.com/Karnishkarnish/GitDemo
 * branch            main      -> FETCH_HEAD
Already up to date.
```

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

```
karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (main)
$ cat .gitignore
temp/

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_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)

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_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

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (main)
$ git commit -m ".gitignore file created"
[main e8cae38] .gitignore file created
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_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), 292 bytes | 97.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Karnishkarnish/GitDemo.git
 e016cf4..e8cae38  main -> main
```

Docker :

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

```
karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (main)
$ touch docker-compose.yaml

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (main)
$ nano docker-compose.yaml

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (main)
$ git add .
warning: in the working copy of 'docker-compose.yaml', LF will be replaced by CRLF the next time Git touches it

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_Daily_Task/Gitdemo/GitDemo (main)
$ git commit -m "Docker compose file created"
[main 99540d6] Docker compose file created
 1 file changed, 33 insertions(+)
 create mode 100644 docker-compose.yaml

karnish.n@PTPLL419 MINGW64 ~/Desktop/Day_11_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% (3/3), done.
Writing objects: 100% (3/3), 553 bytes | 184.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Karnishkarnish/GitDemo.git
 e8cae38..99540d6  main -> main
```

```
1  version: '3.8'
2
3  services:
4    nginx:
5      image: nginx:latest
6      ports:
7        - "80:80"
8      volumes:
9        - ./nginx.conf:/etc/nginx/nginx.conf
10     depends_on:
11       - web
12
13   web:
14     build: ./web
15     ports:
16       - "3000:3000"
17     volumes:
18       - ./web:/usr/src/app
19     environment:
20       - NODE_ENV=development
21     depends_on:
22       - mongo
23
24   mongo:
25     image: mongo:latest
26     ports:
27       - "27017:27017"
28     volumes:
29       - mongo-data:/data/db
30
31   volumes:
32     mongo-data:
```

2.What is Docker hub?

Docker Hub is a cloud-based repository where Docker users can store, share, and manage container images. It allows users to access thousands of pre-built images and automate the deployment of containerized applications.

