

LOKESHVISWA M

21/07/2025

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

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

```
okeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git (master)
mkdar gitdemo
ash: mkdar: command not found

okeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git (master)
mkdir gitdemo

okeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git (master)
cd gitdemo

okeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (master)
git init
initialized empty Git repository in C:/Users/lokeshviswa.m/Desktop/git/gitdemo/.git/
```

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

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

```
lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (master)
$ echo "hello to the world " >welcome.txt

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (master)
$ cat welcome.txt
hello to the world

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (master)
$ ls -al
total 5
drwxr-xr-x 1 lokeshviswa.m 1049089  0 Jul 21 16:45 ./
drwxr-xr-x 1 lokeshviswa.m 1049089  0 Jul 21 16:37 ../
drwxr-xr-x 1 lokeshviswa.m 1049089  0 Jul 21 16:38 .git/
-rw-r--r-- 1 lokeshviswa.m 1049089 21 Jul 21 16:45 welcome.txt
```

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

6. Execute git status

```

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (master)
$ cat welcome.txt
hello to the world

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (master)
$ git status
On branch master

No commits yet

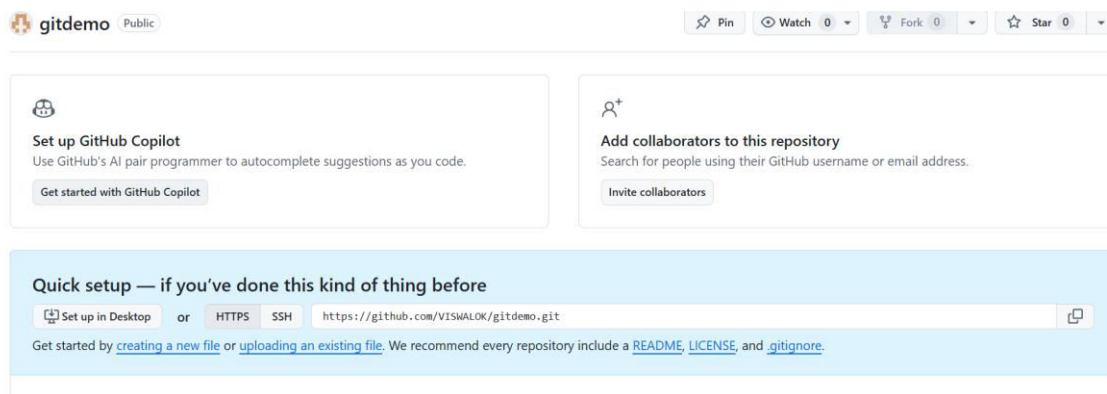
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        welcome.txt

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

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (master)
$

```

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



8. Add remote origin

```

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (master)
$ git remote add origin https://github.com/VISWALOK/gitdemo.git

```

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

```

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ git add .
warning: in the working copy of 'welcome.txt', LF will be replaced by CRLF the next time Git touches it

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ git commit -m "first commit"
[main (root-commit) 2fb372c] first commit
1 file changed, 1 insertion(+)
create mode 100644 welcome.txt

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 238 bytes | 79.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/VISWALOK/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.

```

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/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), 919 bytes | 25.00 KiB/s, done.
From https://github.com/VISWALOK/gitdemo
 * branch            main       -> FETCH_HEAD
 2fb372c..d360880    main       -> origin/main
Updating 2fb372c..d360880
Fast-forward
 welcome.txt | 2 +
1 file changed, 1 insertion(+), 1 deletion(-)

```

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

The screenshot displays the GitHub web interface for a repository named 'gitdemo'. On the left, a sidebar shows the file explorer with 'main' selected and a search bar. Below it, a list of files includes '.gitignore' and 'welcome.txt'. The main area shows the 'welcome.txt' file, which contains the text 'hello to the world git training'. The file is associated with commit 'd360880' from 16 minutes ago. The interface includes tabs for 'Code' and 'Blame', and a status bar at the bottom indicating '1 lines (1 loc) · 35 Bytes'.

```

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ mkdir tempt

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ echo "git">tempt/test.txt

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ git commit -m"test file"
[main 2464aaa] test file
1 file changed, 1 insertion(+)
create mode 100644 .gitignore

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ git push
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), 288 bytes | 96.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/VISWALOK/gitdemo.git
d360880..2464aaa  main -> main

```

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

```

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ touch Docker-compose.yml

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ nano Docker-compose.yml

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ git add .
warning: in the working copy of 'Docker-compose.yml', LF will be replaced by CRLF the next time Git touches it

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

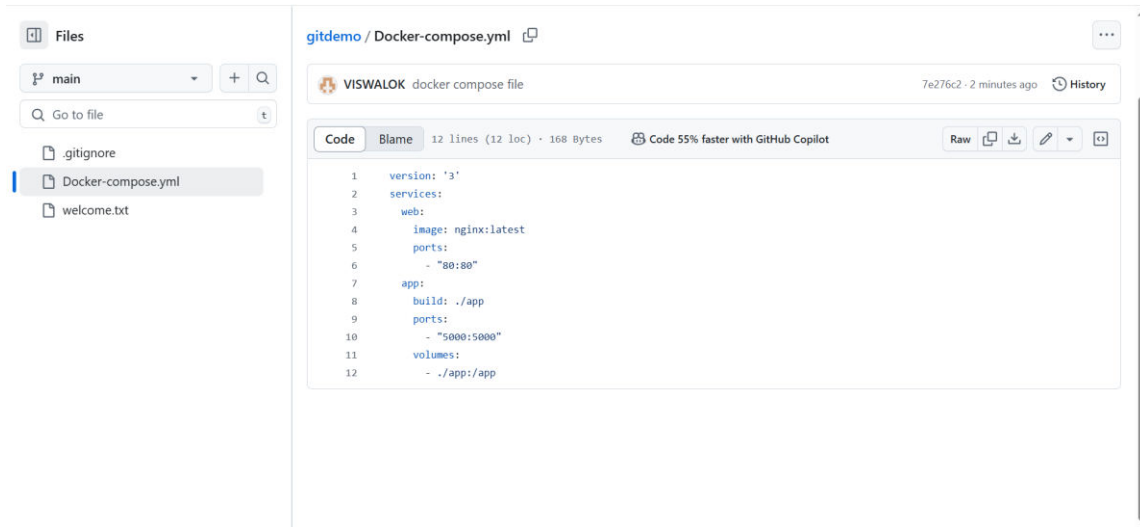
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Docker-compose.yml

it
lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ git commit -m "docker compose file"
[main 7e276c2] docker compose file
1 file changed, 12 insertions(+)
create mode 100644 Docker-compose.yml

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$ git push
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), 438 bytes | 146.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/VISWALOK/gitdemo.git
2464aaa..7e276c2  main -> main

lokeshviswa.m@PTPLL434 MINGW64 ~/Desktop/git/gitdemo (main)
$

```



2. What is Docker hub?

Docker Hub is a cloud-based repository where Docker users can store, share, and manage Docker images. It's like GitHub, but specifically for Docker container images.

Key Features of Docker Hub:

Image Repository:

Public and private repositories to host container images.

Public images can be used freely by anyone (like official Ubuntu, Nginx, MySQL images).

Private images are restricted to your account or team.

Official Images:

Maintained by Docker, these are trusted and secure images for popular software like nginx, mysql, node, python, etc.

Automated Builds:

You can connect Docker Hub to a GitHub or Bitbucket repo. When you push code, Docker Hub automatically builds and updates the image.

Image Search:

Search and pull prebuilt images instead of creating your own from scratch.

Push and Pull Support:

You can push images to your Docker Hub account and pull them to any system using Docker.

User Collaboration:

Add collaborators or teams for shared access to private repositories.

