

EXPERIMENT NO. 2

Aim:

To study and practice GIT commands for version control.

Theory:

Git is one of the most popular version control systems. It is a distributed version control system. Changes do not have to be committed to the same central repository, which would require that every person working on the project to access that central repository and download the latest code in order to save changes.

Some of the basic operations in Git are:

1. Initialize
2. Add
3. Commit
4. Pull
5. Push

Some advanced Git operations are:

1. Branching
2. Merging
3. Rebasing

The following diagram depicts all supported operations in GIT

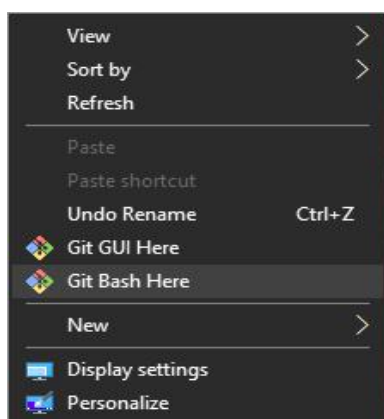


Installation of GIT

1) In windows, download GIT from <https://git-scm.com/> and perform the straightforward installation.

2) In Ubuntu, install GIT using `$sudo apt install git`, Confirm the version after installation `$git --version`

Once installation is done, open the terminal in Ubuntu and perform the following steps or in windows Right click and select Git bash here.



Version Control

To perform version control, let us create a directory dvcs (Distributed version control system)

and change directory to dvcs.

```
$ mkdir git-dvcs
```

```
$ cd git-dvcs/
```

A terminal window with a black background and green text. The title bar shows 'MINGW64:/c/Users/Jay Parmar/git-dvcs'. The prompt is 'Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~ (master)'. The commands entered are '\$ mkdir git-dvcs' and '\$ cd git-dvcs/'. The output shows the directory has been created and the current directory is now ~/git-dvcs.

```
MINGW64:/c/Users/Jay Parmar/git-dvcs

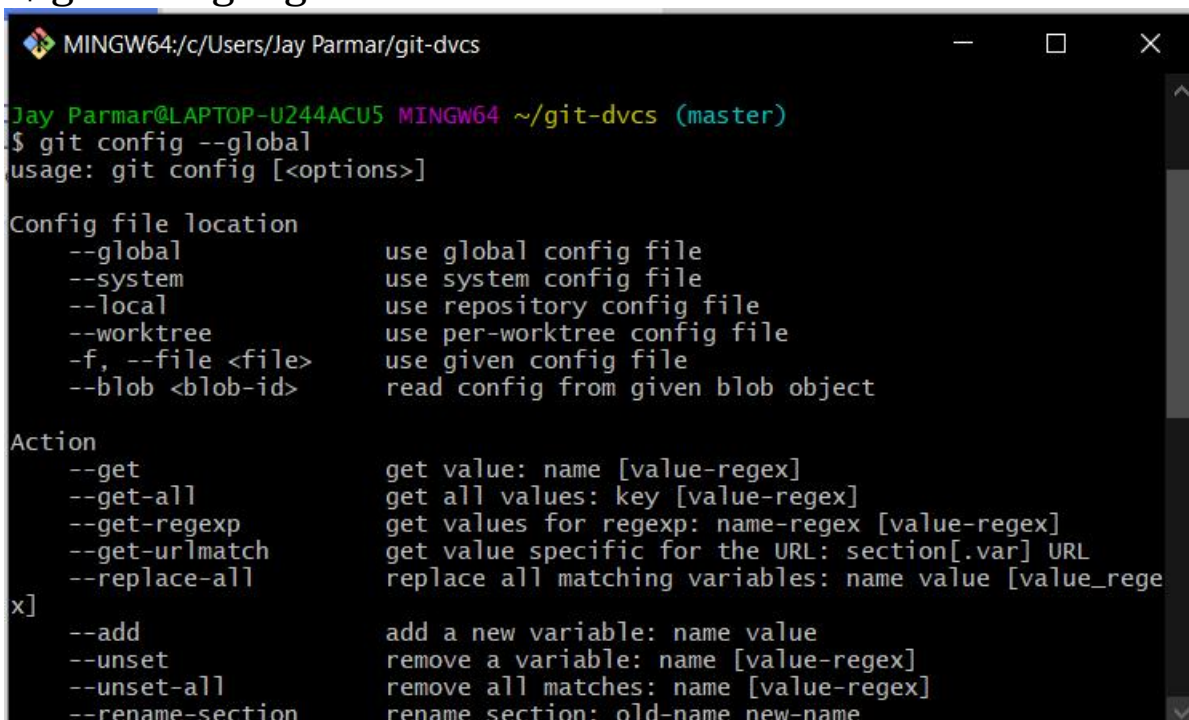
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~ (master)
$ mkdir git-dvcs

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~ (master)
$ cd git-dvcs/

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)
$
```

Now check the user information using

```
$ git config --global
```

A terminal window with a black background and green text. The title bar shows 'MINGW64:/c/Users/Jay Parmar/git-dvcs'. The prompt is 'Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)'. The command entered is '\$ git config --global'. The output shows the usage of the git config command, including config file location and action options.

```
MINGW64:/c/Users/Jay Parmar/git-dvcs

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)
$ git config --global
usage: git config [<options>]

Config file location
  --global      use global config file
  --system      use system config file
  --local       use repository config file
  --worktree    use per-worktree config file
  -f, --file <file> use given config file
  --blob <blob-id> read config from given blob object

Action
  --get         get value: name [value-regex]
  --get-all    get all values: key [value-regex]
  --get-regexp  get values for regexp: name-regex [value-regex]
  --get-urlmatch get value specific for the URL: section[.var] URL
  --replace-all replace all matching variables: name value [value_regex]

x]
  --add         add a new variable: name value
  --unset       remove a variable: name [value-regex]
  --unset-all  remove all matches: name [value-regex]
  --rename-section rename section: old-name new-name
```

As there are no users defined, let us define it using following two commands

```
$ git config --global user.name "jay-2000"
```

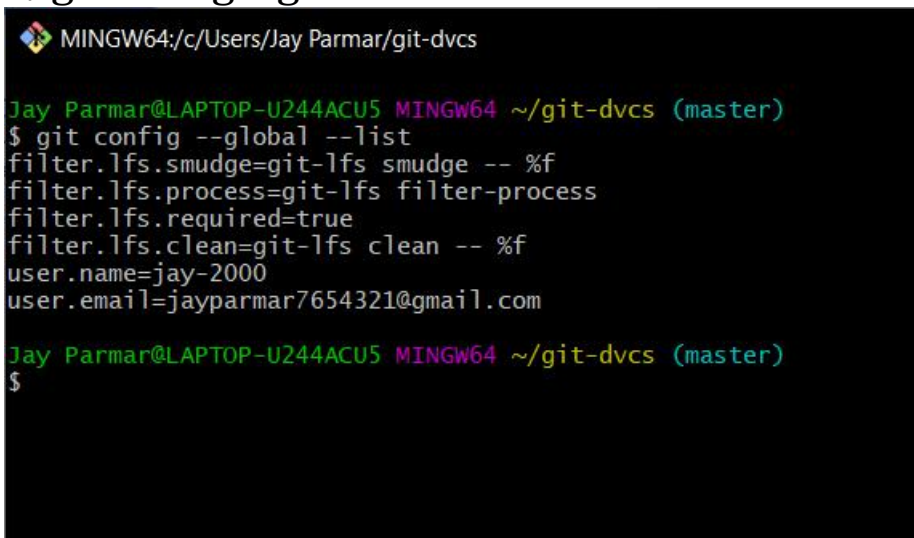
```
$ git config --global user.email  
"jayparmar7654321@gmail.com"
```

A terminal window with a black background and green text. The prompt is 'Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)'. The first command is '\$ git config --global user.name "jay-2000"'. The second command is '\$ git config --global user.email "jayparmar7654321@gmail.com"'. The third command is '\$', which is followed by a blank line.

```
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)  
$ git config --global user.name "jay-2000"  
  
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)  
$ git config --global user.email "jayparmar7654321@gmail.com"  
  
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)  
$
```

Now, check the list of users

```
$ git config --global --list
```

A terminal window with a black background and green text. The prompt is 'Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)'. The command is '\$ git config --global --list'. The output is: 'filter.lfs.smudge=git-lfs smudge -- %f', 'filter.lfs.process=git-lfs filter-process', 'filter.lfs.required=true', 'filter.lfs.clean=git-lfs clean -- %f', 'user.name=jay-2000', and 'user.email=jayparmar7654321@gmail.com'. The prompt is then '\$'.

```
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)  
$ git config --global --list  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
filter.lfs.clean=git-lfs clean -- %f  
user.name=jay-2000  
user.email=jayparmar7654321@gmail.com  
  
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)  
$
```

Let us create a repository for version control named "git-demo-project"

```
$ mkdir git-demo-project
```

```
$ cd git-demo-project/
```

Now, initialize the repository using following command

```
$ git init
```

```
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)
$ mkdir git-demo-project

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs (master)
$ cd git-demo-project/

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git init
Initialized empty Git repository in C:/Users/Jay Parmar/git-dvcs/git-demo-project/.git/

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ |
```

Now, let us add some files inside our repository “git-demo-project”

To add files in index and staging area, add command is used along with dot (. Dot means current directory)

\$ git add .

A terminal window with a black background and green text. The prompt is 'Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)'. The command '\$ git add .' is entered and executed. The prompt returns to '\$ |'.

Index and staging area

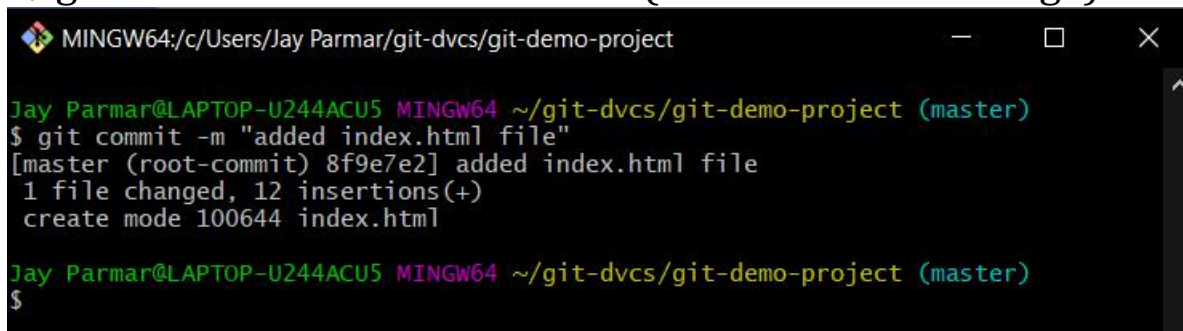
To check the status of repository, use

\$ git status

Which will show you some untrack files, so untracks files can be tracked using commit command.

Now, let us commit the changes

\$ git commit -m "First Commit" (#here -m for message)

A terminal window titled 'MINGW64:/c/Users/Jay Parmar/git-dvcs/git-demo-project'. The prompt is 'Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)'. The command '\$ git commit -m "added index.html file"' is entered and executed. The output is: '[master (root-commit) 8f9e7e2] added index.html file', '1 file changed, 12 insertions(+)', and 'create mode 100644 index.html'. The prompt returns to '\$'.

Added index.html in our directory

\$ touch teststatus

```
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ touch teststatus

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        teststatus

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

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ |
```

History of Commits

\$ git log

```
MINGW64:/c/Users/Jay Parmar/git-dvcs/git-demo-project

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git log
commit 8f9e7e2db2fb604f654875fe9a2dc30d0313c5c3 (HEAD -> master)
Author: jay-2000 <jayparmar7654321@gmail.com>
Date: Tue Aug 17 22:03:37 2021 +0530

    added index.html file

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ |
```

Now Create a Repository on github.com. Open github.com→ create an account→After login

Select New repository from the menu.

← → ↻ 🏠 🔍

https://github.com/jay-2000

🔔 ⚙️ ⭐ 🔄 👤

🐙

Search or jump to...

Pull requests

Issues

Marketplace

Explore


🔔 + 👤

📄 Overview

📁 Repositories 36

📁 Projects

📁 Packages



Jay Parmar

jay-2000

Hello world

Edit profile

38 followers · 101 following · 57 stars

DMCE23


mumbai, India

https://jay-2000.github.io/portfolio/

@_jaystwt

Highlights

jay-2000 / README.md




Hi 🇮🇳, I'm Jay Parmar

A passionate Budding Web Engineer from India


Profile views 1.277

FOLLOW @_JAYSTWT 43

- I'm currently working as an Opensource contributor at LGM-SOC'21 AND DCP'21(DevIncept)
- I'm currently learning Docker as well as DevOps stuff, JS and its libraries and frameworks



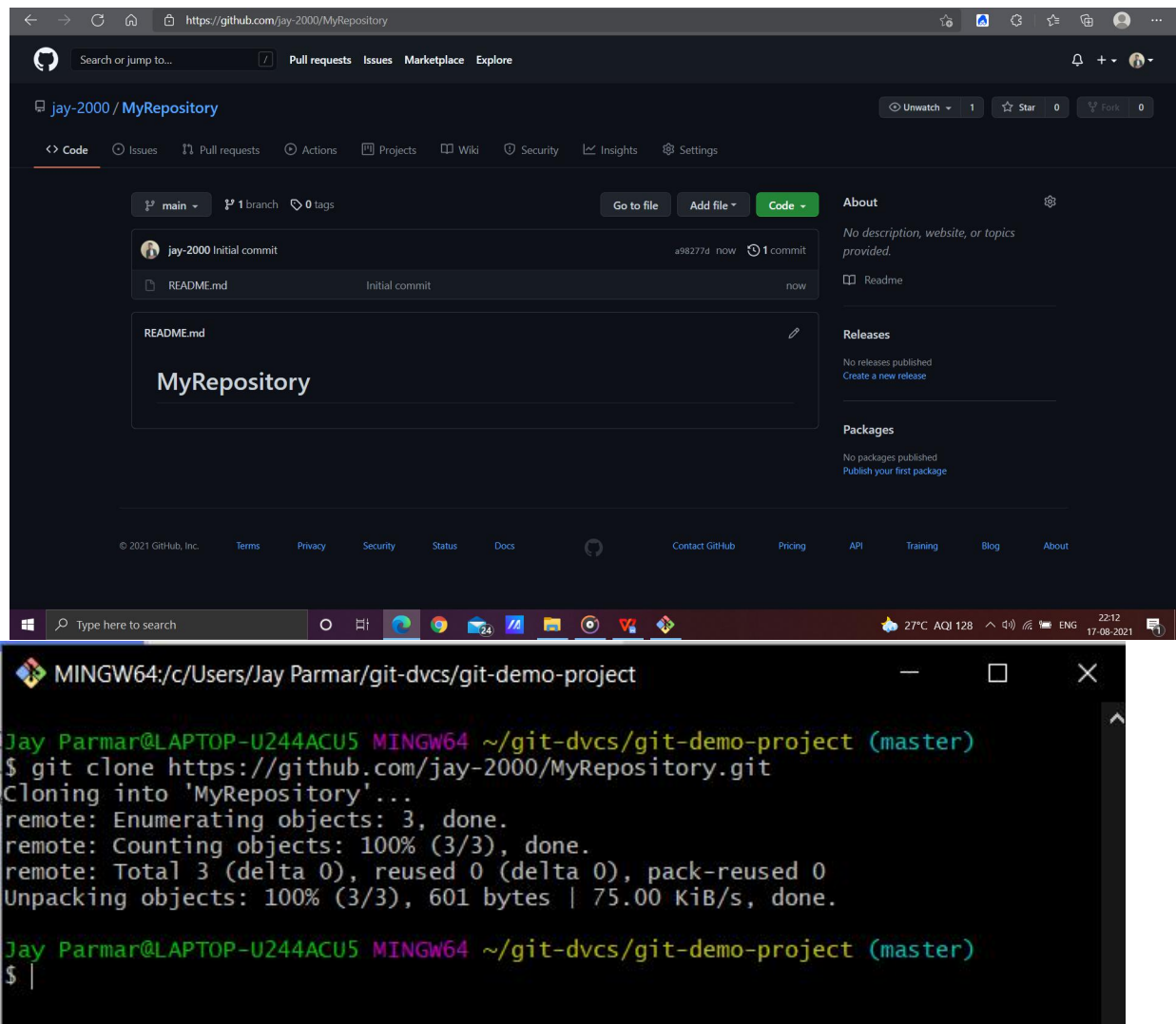
Type here to search



🔥 27°C AQI 128

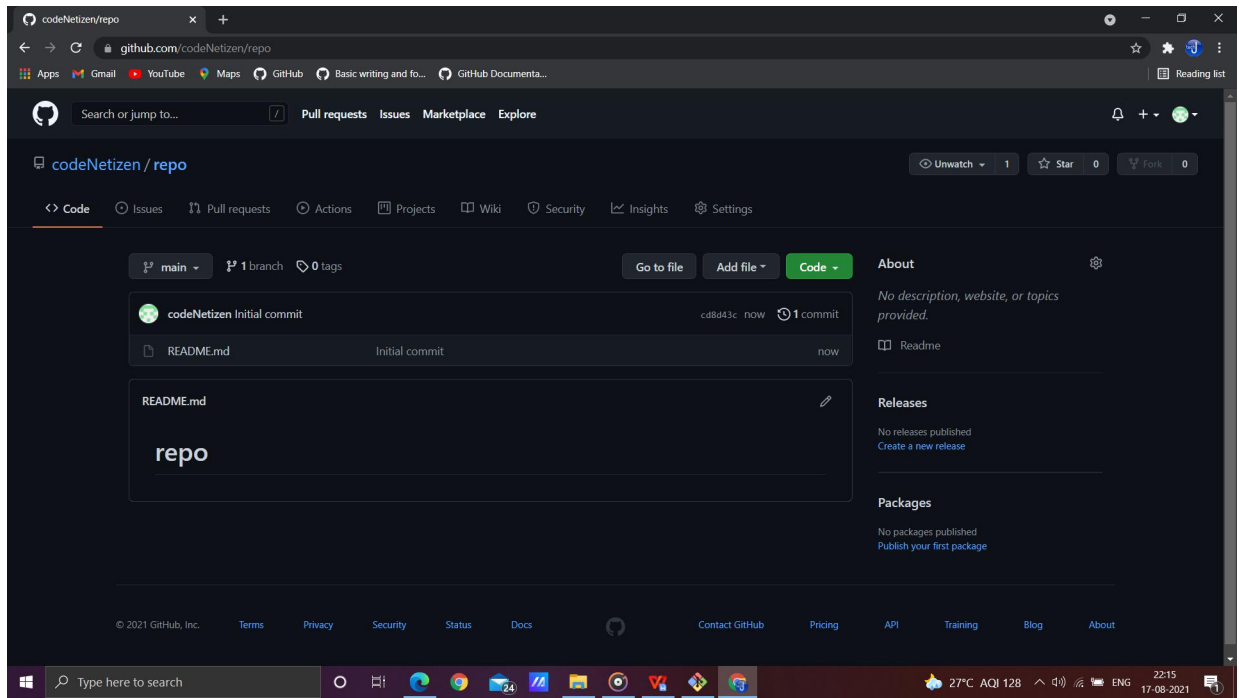
📶 ENG

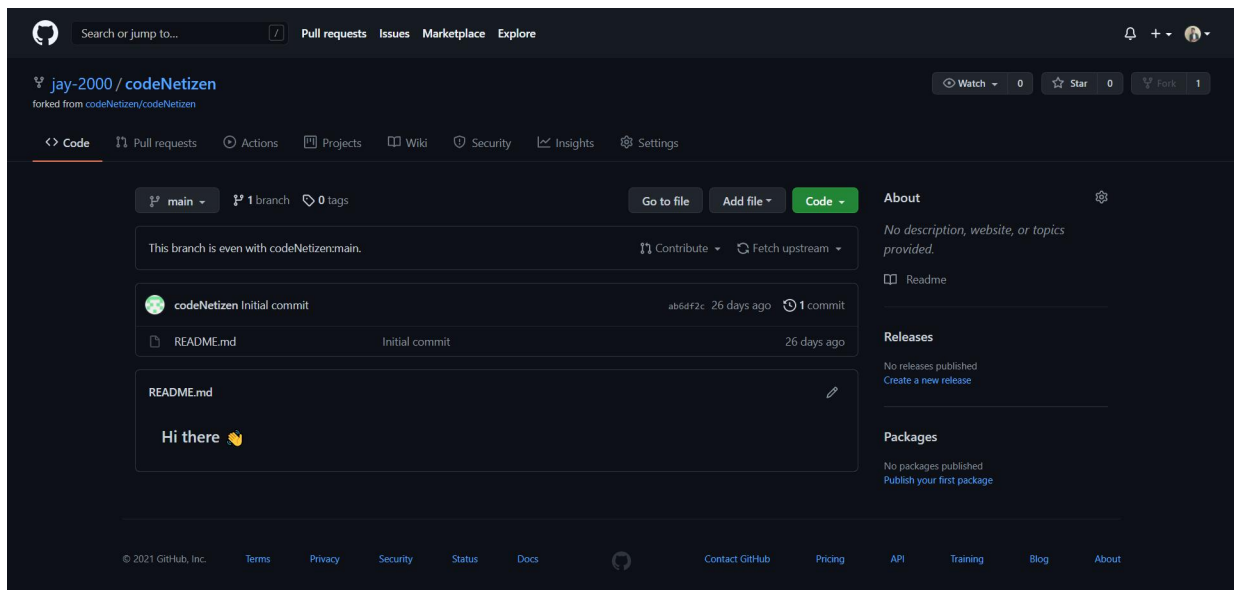
22:09 17-08-2021



Now fork the repository (Sharing with other users who wants to contribute).

Login with another account→Copy and Paste URL of repository→then just click on fork to clone to others account.





```
MINGW64:/c/Users/Jay Parmar/git-dvcs/git-demo-project

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git clone https://github.com/jay-2000/codeNetizen.git
Cloning into 'codeNetizen'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 884 bytes | 126.00 KiB/s, done.

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$
```

Pull and Push Processes

Push → Push changes to Web repository

Pull → Pull changes to Local repository

1) Push command to remote reference origin master

\$ git remote add origin

<https://github.com/bhushanjadhav1/siesworkshop.git>

\$ git remote show origin

```
MINGW64:/c/Users/Jay Parmar/git-dvcs/git-demo-project

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git remote add origin https://github.com/jay-2000/codeNetizen.git

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/jay-2000/codeNetizen.git
Push URL: https://github.com/jay-2000/codeNetizen.git
HEAD branch: main
Remote branch:
main new (next fetch will store in remotes/origin)

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$
```

```
main new (next fetch with store in remotes/origins)
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git pull https://github.com/jay-2000/codeNetizen.git
warning: no common commits
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 884 bytes | 68.00 KiB/s, done.
From https://github.com/jay-2000/codeNetizen
 * branch          HEAD      -> FETCH_HEAD
fatal: refusing to merge unrelated histories

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
```

```
MINGW64:/c/Users/Jay Parmar/git-dvcs/git-demo-project

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    MyRepository/
    codeNetizen/
    teststatus

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

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ |
```

```
MINGW64:/c/Users/Jay Parmar/git-dvcs/git-demo-project

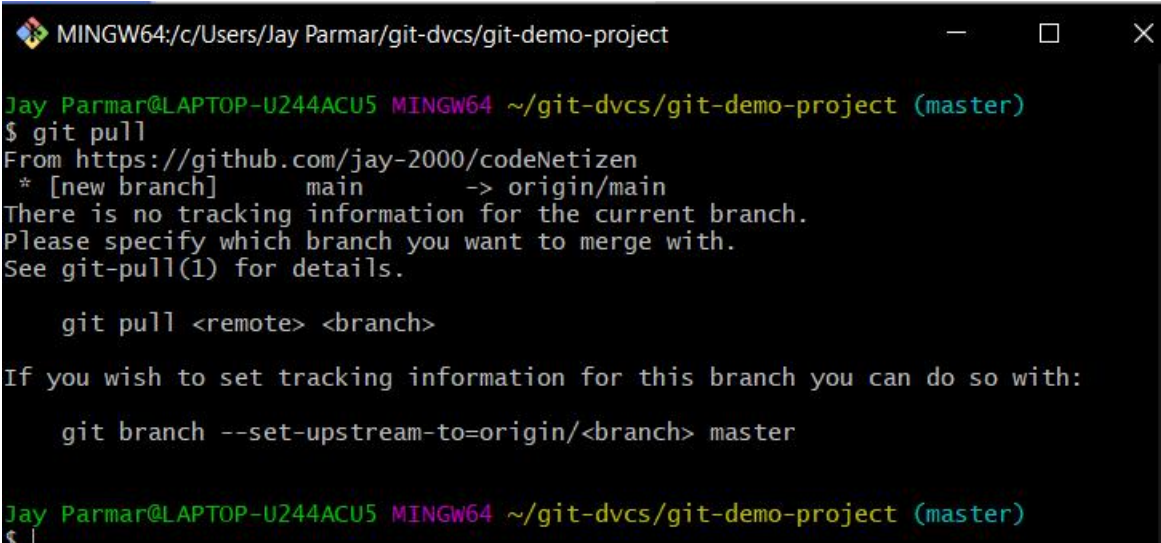
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git remote -v
origin  https://github.com/jay-2000/codeNetizen.git (fetch)
origin  https://github.com/jay-2000/codeNetizen.git (push)

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ |
```

```
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git push -v origin master
Pushing to https://github.com/jay-2000/codeNetizen.git
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 400 bytes | 400.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
POST git-receive-pack (563 bytes)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/jay-2000/codeNetizen/pull/new/master
remote:
To https://github.com/jay-2000/codeNetizen.git
 * [new branch]      master -> master
updating local tracking ref 'refs/remotes/origin/master'

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ |
```


\$ git pull

A screenshot of a Windows terminal window titled "MINGW64:/c/Users/Jay Parmar/git-dvcs/git-demo-project". The prompt is "Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)". The command "\$ git pull" has been entered. The output shows the source as "https://github.com/jay-2000/codeNetizen", a new branch "main" created from "origin/main", and a message stating there is no tracking information for the current branch. It suggests using "git pull <remote> <branch>" or "git branch --set-upstream-to=origin/<branch> master". The prompt returns to "Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)".

```
Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git pull
From https://github.com/jay-2000/codeNetizen
* [new branch]      main      -> origin/main
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> master

Jay Parmar@LAPTOP-U244ACU5 MINGW64 ~/git-dvcs/git-demo-project (master)
$
```

Conclusion:

Hence studied and practiced GIT commands for version control.