



**UNIVERSITY OF PETROLEUM & ENERGY STUDIES**

**College of Engineering Studies**

**Dehradun**

# **DevOps Overview and Source** **Code Management Lab**

**CSDV 2104**

**NAME- YADRISHI DIXIT**

**BRANCH- COMPUTER SCIENCE ENGINEERING**

**BATCH- B-4 DEVOPS**

**SAP ID- 500097959**

**SUBMITTED TO- Mr Arjun Arora sir**  
**(Cybernetics department)**

## Index

S. No	TOPIC	DATE	REMARKS
1	Introduction to git	6 <sup>th</sup> August'22	
2	Installing gitbash	6 <sup>th</sup> August'22	
3	Create a git repository	6 <sup>th</sup> August'22	
4	Commands- init, add, version	6 <sup>th</sup> August'22	
	Add, commit, status, clone, remote, log, branch, checkout, push	27th Aug'22	
5	GITHUB: Introduction to working	3 <sup>rd</sup> Sep'22	

# **DevOps Lab Experiments**

## **EXPERIMENT-1**

### **Introduction to GIT**

Version control, also known as source control, is the practice of tracking and managing changes to software code. Version control systems are software tools that help software teams manage changes to source code over time. As development environments have accelerated, version control systems help software teams work faster and smarter. They are especially useful for DevOps teams since they help them to reduce development time and increase successful deployments.

Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

Software developers working in teams are continually writing new source code and changing existing source code. The code for a project, app or software component is typically organized in a folder structure or "file tree". One developer on the team may be working on a new feature while another developer fixes an unrelated bug by changing code, each developer may make their changes in several parts of the file tree

## **GIT**

The most widely used modern version control system in the world today is Git. Git is a mature, actively maintained open-source project originally

developed in 2005 by Linus Torvalds. Git is an example of a DVCS (hence Distributed Version Control System).

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows

## EXPERIMENT-2

### Installing git CLI and git GUI

Download Git from <https://git-scm.com/download/win>

### Downloading Git



You are downloading the latest (2.34.1) 64-bit version of **Git for Windows**. This is the most recent **maintained build**. It was released **17 days ago**, on 2021-11-25.

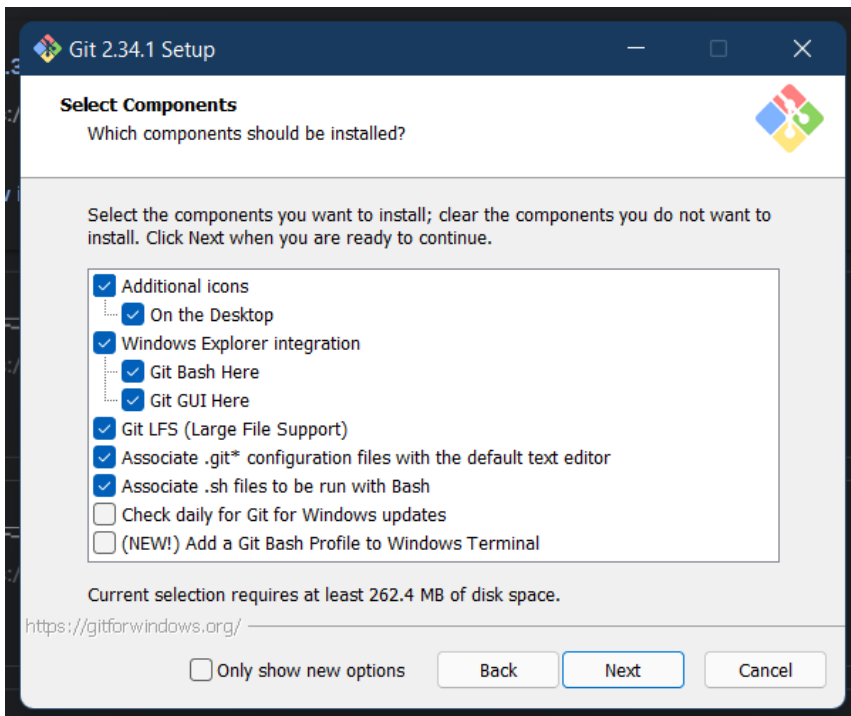
[Click here to download manually](#)

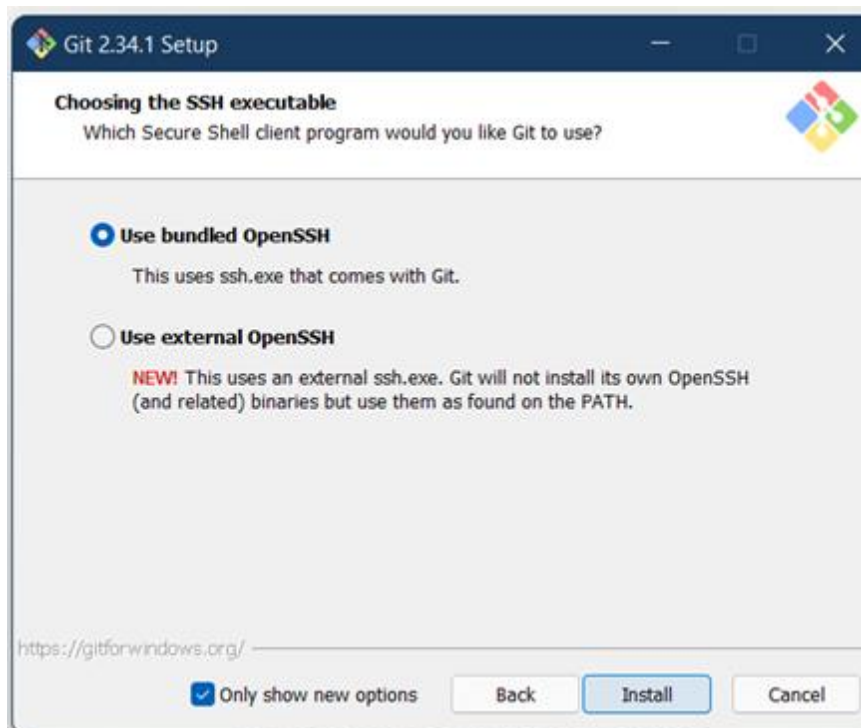
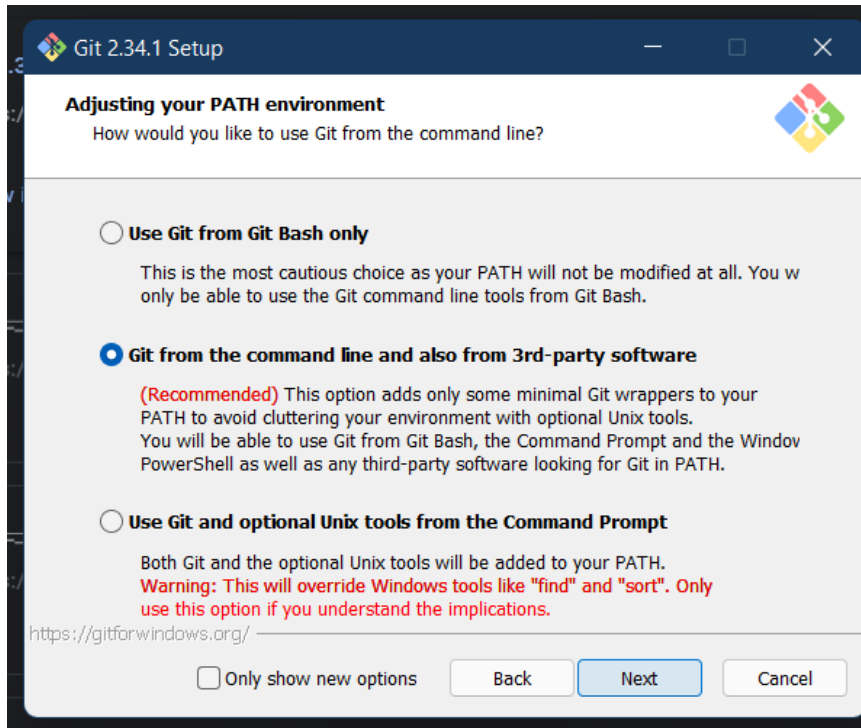
#### Other Git for Windows downloads

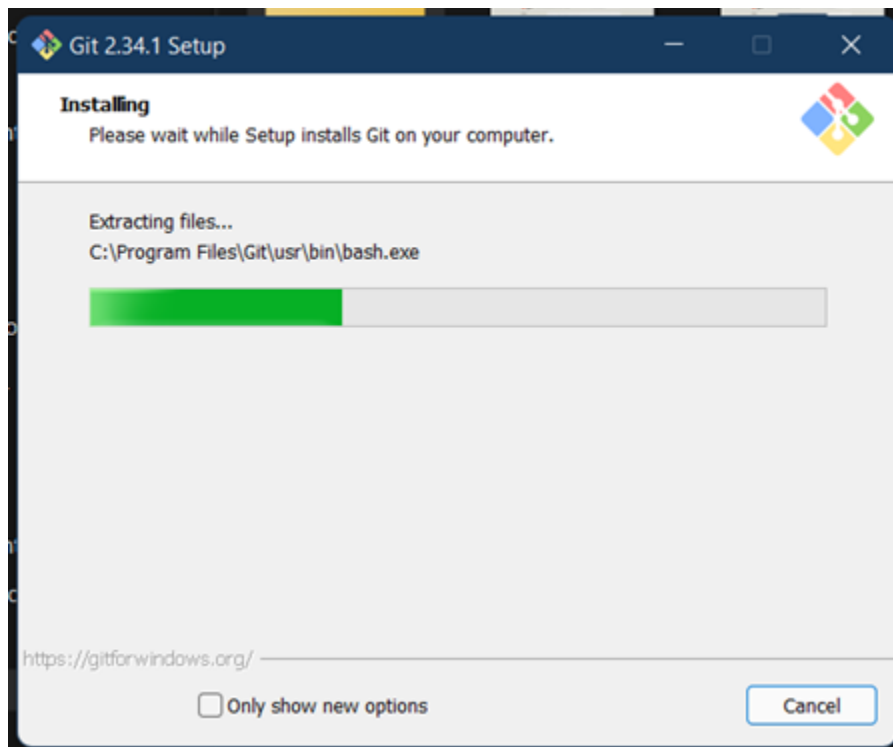
Git for Windows Setup  
[32-bit Git for Windows Setup.](#)  
[64-bit Git for Windows Setup.](#)

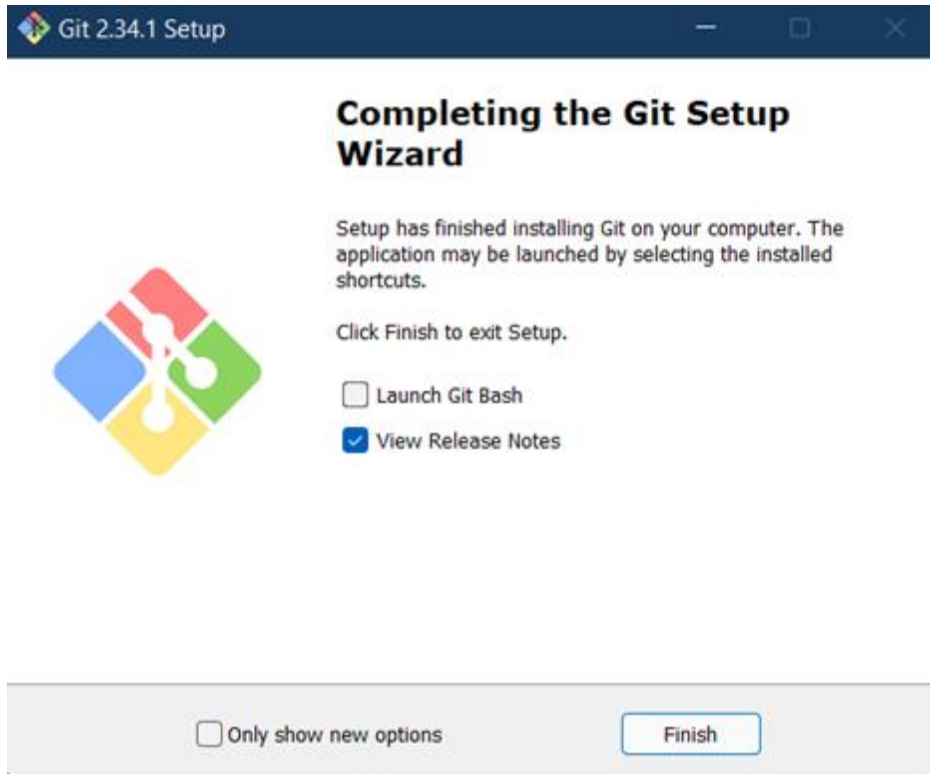
Git for Windows Portable ("thumbdrive edition")  
[32-bit Git for Windows Portable.](#)  
[64-bit Git for Windows Portable.](#)

The current source code release is version 2.34.1. If you want the newer version, you can build it from [the source code](#).



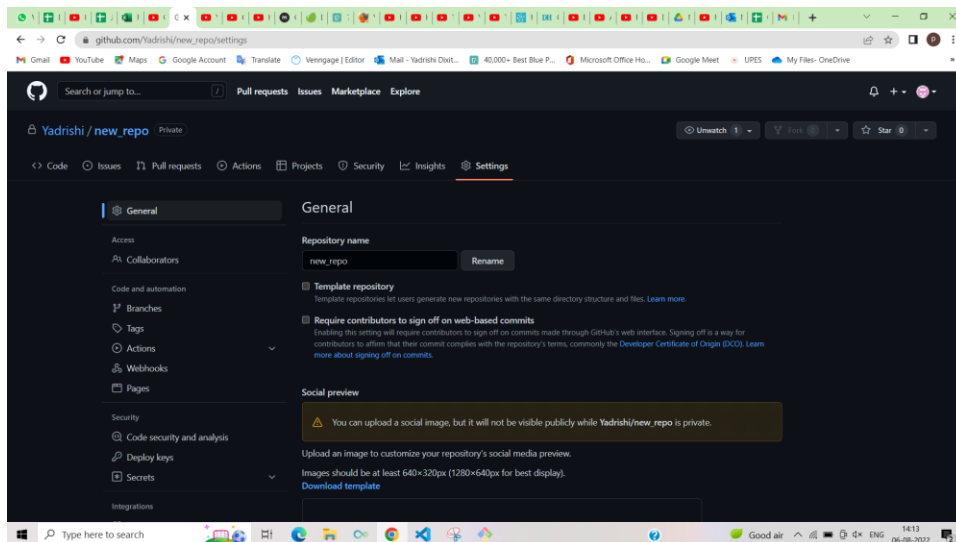




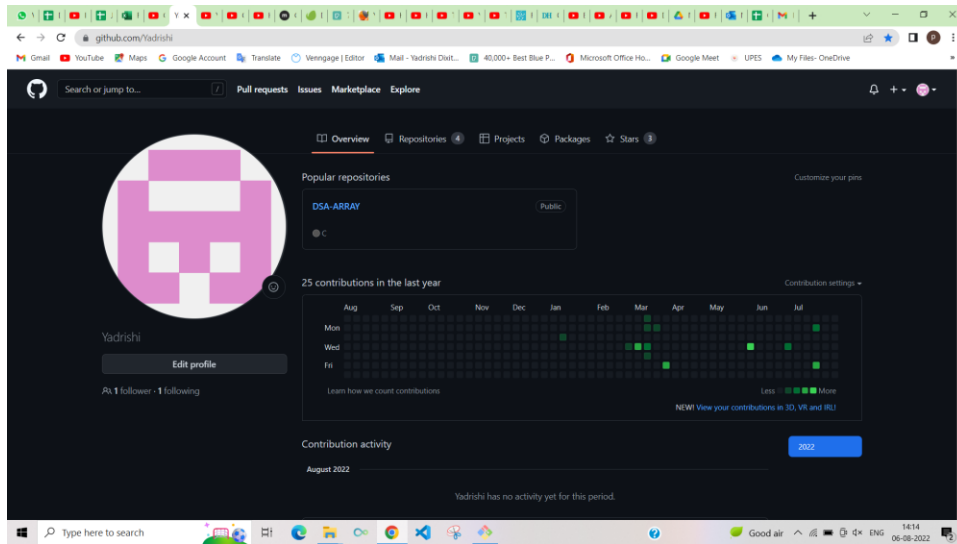


## EXPERIMENT-3

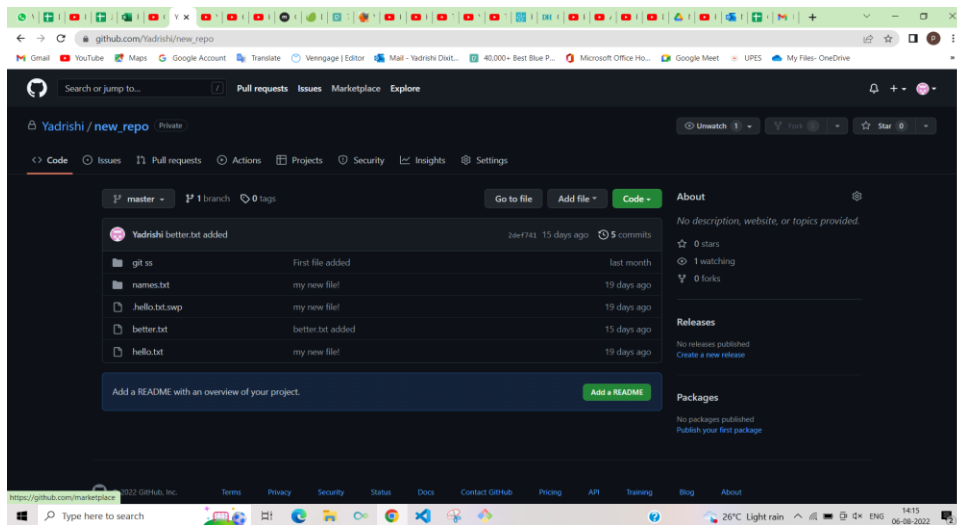
### Creating a profile on GITHUB







## Making Git Repositories



## EXPERIMENT-4

### Working With Git – Commands

- Git version: -

MINGW64:/c/Users/dixit

```
YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~ (master)
$ git --version
git version 2.35.1.windows.2
```

```
YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~ (master)
$
```

- **Git configuration:–**

The git config command is a convenience function that is used to set Git configuration values on a global or local project level.

MINGW64:/c/Users/dixit

```
YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~ (master)
$ git --version
git version 2.35.1.windows.2

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~ (master)
$ git config
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-pattern]
--get-all        get all values: key [value-pattern]
--get-regexp      get values for regexp: name-regex [value-pattern]
--get-urlmatch    get value specific for the URL: section[.var] URL
--replace-all    replace all matching variables: name value [value-pattern]
--add            add a new variable: name value
--unset          remove a variable: name [value-pattern]
--unset-all      remove all matches: name [value-pattern]
--rename-section  rename section: old-name new-name
--remove-section  remove a section: name
-l, --list        list all
--fixed-value     use string equality when comparing values to 'value-pattern'
-e, --edit        open an editor
--get-color       find the color configured: slot [default]
--get-colorbool   find the color setting: slot [stdout-is-tty]

Type
-t, --type <v>  value is given this type
--bool           value is "true" or "false"
--int            value is decimal number
--bool-or-int    value is --bool or --int
--bool-or-str    value is --bool or string
--path           value is a path (file or directory name)
--expiry-date    value is an expiry date

Other
-z, --null        terminate values with NUL byte
--name-only       show variable names only
--includes        respect include directives on lookup
--show-origin     show origin of config (file, standard input, blob, command line)
--show-scope      show scope of config (worktree, local, global, system, command)
--default <value> with --get, use default value when missing entry
```

```
YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~ (master)
$ |
```

- **Git init:-**

```
--show-scope      show scope of config (worktree, local, global, system, command)
--default <value> with --get, use default value when missing entry
```

```
YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~ (master)
$ git init
Reinitialized existing Git repository in C:/Users/dixit/.git/

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~ (master)
$ git init --
Reinitialized existing Git repository in C:/Users/dixit/.git/

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~ (master)
$ |
```

- Add

```
YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (master)
$ git add .

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (master)
$ git restore --staged names.txt

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (master)
$ git add names.txt
warning: LF will be replaced by CRLF in names.txt.
The file will have its original line endings in your working directory

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .names.txt.swp
    modified:   names.txt

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (master)
$ git commit -m "hello"
[master d452bb9] hello
2 files changed, 13 insertions(+)
create mode 100644 .names.txt.swp
```

- Commit

```

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (devprac)
$ git commit -m "This is a new practice branch!"
[devprac 91dc33e] This is a new practice branch!
3 files changed, 5 insertions(+)
create mode 100644 .NetrwMessage.swp
create mode 100644 example.txt

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (devprac)
$ git log
commit 91dc33ead255ed38160409f96bf6fc80f3e864d3 (HEAD -> devprac)
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Sat Aug 27 14:19:13 2022 +0530

    This is a new practice branch!

commit 85f00149d21d0be1bb29bf1a89396842100e686a (origin/master, master, kunal, feature)
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Sat Aug 27 11:14:03 2022 +0530

    this ia a new file fro practicing git and github

commit 2de20976c3dfe18edf8928c6b7c8ef92119ca200
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Fri Jul 22 21:37:03 2022 +0530

    bottle.txt added

commit 40dbbd1dafd80a61ed17964206f1aab0e187fda2
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Fri Jul 22 21:35:02 2022 +0530

    hotel.txt added

commit 2def741c4948ffad81f14d59f864a8714a95325f
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Fri Jul 22 18:15:59 2022 +0530

    better.txt added

commit 39049871c13b0592fe75fb31524f0fe6aa77b050
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Mon Jul 18 23:56:40 2022 +0530

    my new file!

commit 35ec36a946efa2a5fca5253e97d3e89612394164
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Wed Jun 29 17:40:04 2022 +0530

    First file added

commit d452bb9533221c6f12ea0c6bd4a5618a9ec6910c

```

- Status

```

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (feature)
$ vi new.txt

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (feature)
$ git status
On branch feature
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   example.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   new.txt

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

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (feature)
$ |

```

- Remote

```

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (feature)
$ git remote -v
origin https://github.com/Yadrishi/new_repo.git (fetch)
origin https://github.com/Yadrishi/new_repo.git (push)

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (feature)
$ git push master
fatal: 'master' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

```

- Push

```

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (master)
$ git push origin master
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (11/11), 1.10 KiB | 161.00 KiB/s, done.
Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Yadrishi/new_repo.git
    2def741..85f0014 master -> master

```

- Clone

```

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (devprac)
$ git clone https://github.com/Revenant1913/Eclipse
Cloning into 'Eclipse'...
remote: Enumerating objects: 885, done.
remote: Counting objects: 100% (885/885), done.
remote: Compressing objects: 100% (532/532), done.
remote: Total 885 (delta 210), reused 833 (delta 158), pack-reused 0
Receiving objects: 100% (885/885), 14.40 MiB | 1.38 MiB/s, done.
Resolving deltas: 100% (210/210), done.
Updating files: 100% (432/432), done.

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (devprac)
$ |

```

- Branch

```

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (master)
$ git push origin master
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (11/11), 1.10 KiB | 161.00 KiB/s, done.
Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Yadrishi/new_repo.git
    2def741..85f0014 master -> master

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (master)
$ git branch feature

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (master)
$ git checkout feature
Switched to branch 'feature'

```

- Checkout

```
YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (feature)
$ git branch devprac

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (feature)
$ git checkout devprac
Switched to branch 'devprac'
A       example.txt
M       new.txt

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (devprac)
$ git add .
warning: LF will be replaced by CRLF in new.txt.
The file will have its original line endings in your working directory

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (devprac)
$ |
```

- Log

```
YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (feature)
$ git log
commit 85f00149d21d0be1bb29bf1a89396842100e686a (HEAD -> feature, origin/master, master)
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Sat Aug 27 11:14:03 2022 +0530

    this ia a new file fro practicing git and github

commit 2de20976c3dfe18edf8928c6b7c8ef92119ca200
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Fri Jul 22 21:37:03 2022 +0530

    bottle.txt added

commit 40dbbd1dafd80a61ed17964206f1aab0e187fda2
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Fri Jul 22 21:35:02 2022 +0530

    hotel.txt added

commit 2def741c4948ffad81f14d59f864a8714a95325f
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Fri Jul 22 18:15:59 2022 +0530

    better.txt added

commit 39049871c13b0592fe75fb31524f0fe6aa77b050
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Mon Jul 18 23:56:40 2022 +0530

    my new file!

commit 35ec36a946efa2a5fca5253e97d3e89612394164
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Wed Jun 29 17:40:04 2022 +0530

    First file added

commit d452bb9533221c6f12ea0c6bd4a5618a9ec6910c
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Wed Jun 1 23:44:18 2022 +0530

    hello

commit 8aff84765a04c0ead0fab8d7a3ccb6df59c573d0
Author: Yadrishi <dixityadrishi@gmail.com>
Date: Wed Jun 1 20:27:18 2022 +0530

    Hello! names.txt added!

YADRISHI DIXIT@LAPTOP-ABL437AQ MINGW64 ~/desktop/gitpractice (feature)
$ git remote -v
```