

GIT-HOL 1

Step 2: Integrate notepad++.exe to Git and make it a default editor

1. To verify if notepad++ is the default editor, execute the command

```
anish@DESKTOP-0F6C75J MINGW64 ~ (main)
$ git config --global -e
```

```
anish@DESKTOP-0F6C75J MINGW64 ~ (main)
$ mkdir GitDemo
```

```
anish@DESKTOP-0F6C75J MINGW64 ~ (main)
$ cd GitDemo
```

Step 3: Add a file to source code repository

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (main)
$ git init
Initialized empty Git repository in C:/Users/anish/GitDemo/.git/
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ ls -a
./ ../ .git/
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ echo "Welcome to Git Lab" > welcome.txt
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ ls
welcome.txt
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ cat welcome.txt
Welcome to Git Lab
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git add welcome.txt
warning: in the working copy of 'welcome.txt', LF will be replaced by CRLF the next time
Git touches it
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git commit
[master (root-commit) a4576a1] Initial commit - Created welcome.txt - Added welcome
message
1 file changed, 1 insertion(+)
create mode 100644 welcome.txt
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
```

```
$ git status
On branch master
nothing to commit, working tree clean
```

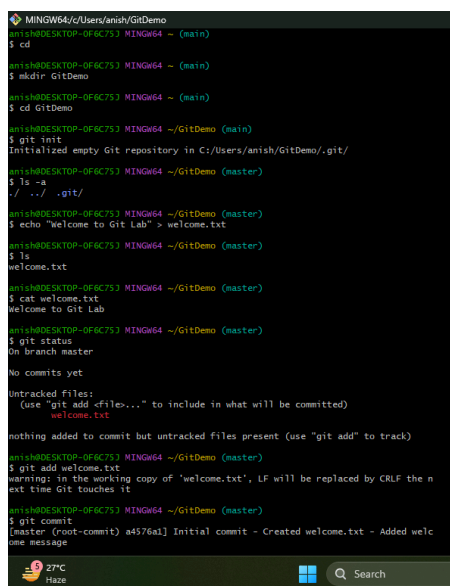
```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git remote add origin https://gitlab.com/anishagarwal670/gitdemo.git
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git pull origin master --allow-unrelated-histories
fatal: couldn't find remote ref master
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 264 bytes | 264.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: To create a merge request for master, visit:
remote:
https://gitlab.com/anishagarwal670/gitdemo/-/merge_requests/new?merge_request%5Bsource_branch%5D=master
remote:
To https://gitlab.com/anishagarwal670/gitdemo.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

```
nothing to commit, working tree clean
```



```
MINGW64/C:/Users/anish/GitDemo
anish@DESKTOP-0F6C75J MINGW64 ~ (main)
$ cd
anish@DESKTOP-0F6C75J MINGW64 ~ (main)
$ mkdir GitDemo
anish@DESKTOP-0F6C75J MINGW64 ~ (main)
$ cd GitDemo
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (main)
$ git init
Initialized empty Git repository in C:/Users/anish/GitDemo/.git/
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ ls -la
./  ../ .git/
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ echo "Welcome to Git Lab" > welcome.txt
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ ls
welcome.txt
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ cat welcome.txt
Welcome to Git Lab
anish@DESKTOP-0F6C75J MINGW64 ~/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)
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git add welcome.txt
warning: in the working copy of 'welcome.txt', LF will be replaced by CRLF the next time Git touches it
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git commit
[master (root-commit) a4576a1] Initial commit - Created welcome.txt - Added welcome message
```

GIT-HOL 2

Create a **“.log”** file and a **log folder** in the working directory of Git. Update the **.gitignore** file in such a way that on committing, these files (.log extensions and log folders) are ignored.

Verify if the git status reflects the same about working directory, local repository and git repository.

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
```

```
$ mkdir -p log
```

```
echo "Server started at $(date)" > log/server.log
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
```

```
$ ls -la
```

```
ls -la log
```

```
cat example.log
```

```
cat log/server.log
```

```
total 22
```

```
drwxr-xr-x 1 anish 197609  0 Aug  9 02:45 ./
```

```
drwxr-xr-x 1 anish 197609  0 Aug  9 00:23 ../
```

```
drwxr-xr-x 1 anish 197609  0 Aug  9 00:29 .git/
```

```
-rw-r--r-- 1 anish 197609 51 Aug  9 02:45 example.log
```

```
drwxr-xr-x 1 anish 197609  0 Aug  9 02:45 log/
```

```
-rw-r--r-- 1 anish 197609 19 Aug  9 00:24 welcome.txt
```

```
total 1
```

```
drwxr-xr-x 1 anish 197609  0 Aug  9 02:45 ./
```

```
drwxr-xr-x 1 anish 197609  0 Aug  9 02:45 ../
```

```
-rw-r--r-- 1 anish 197609 47 Aug  9 02:45 server.log
```

```
Sample log created on Sat Aug  9 02:45:42 IST 2025
```

```
Server started at Sat Aug  9 02:45:54 IST 2025
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
```

```
$ git status --short
```

```
?? example.log
```

```
?? log/
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
```

```
$ # ensure .gitignore exists or create it
```

```
touch .gitignore
```

```
# add rules only if they don't already exist
```

```
grep -qxF '*.log' .gitignore || echo '*.log' >> .gitignore
```

```
grep -qxF 'log/' .gitignore || echo 'log/' >> .gitignore
```

```
# show the file so you can confirm
```

```
cat .gitignore
```

```
*.log
```

```
log/
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git ls-files --others --ignored --exclude-standard
example.log
log/server.log
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git check-ignore -v example.log
git check-ignore -v log/server.log
.gitignore:1:*.log      example.log
.gitignore:2:log/      log/server.log
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git status --ignored
On branch master
Your branch is up to date with 'origin/master'.
```

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

```
Ignored files:
(use "git add -f <file>..." to include in what will be committed)
example.log
log/
```

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

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ # untrack the files from the index (keeps local files)
git rm --cached example.log
git rm -r --cached log
```

```
# confirm they are removed from index (will show as deleted in working tree)
git status --short
fatal: pathspec 'example.log' did not match any files
fatal: pathspec 'log' did not match any files
?? .gitignore
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git add .gitignore
git commit -m "Ignore .log files and log/ directory; stop tracking logs"
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git
touches it
[master 9b9389a] Ignore .log files and log/ directory; stop tracking logs
1 file changed, 2 insertions(+)
create mode 100644 .gitignore
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git branch --show-current
```

```
git remote -v
```

```
master
```

```
origin https://gitlab.com/anishagarwal670/gitdemo.git (fetch)
```

```
origin https://gitlab.com/anishagarwal670/gitdemo.git (push)
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
```

```
$ git push -u origin $(git branch --show-current)
```

```
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), 325 bytes | 325.00 KiB/s, done.
```

```
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
```

```
remote:
```

```
remote: To create a merge request for master, visit:
```

```
remote:
```

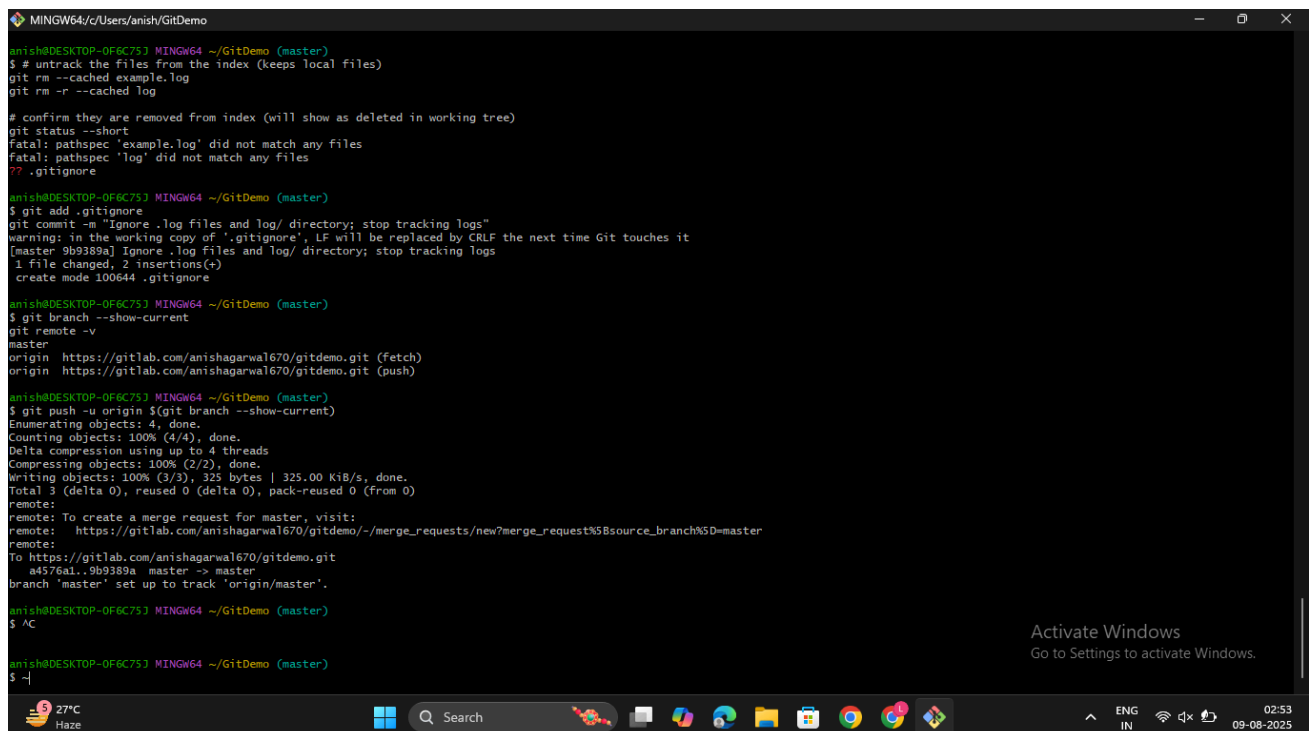
```
https://gitlab.com/anishagarwal670/gitdemo/-/merge_requests/new?merge_request%5Bsource_branch%5D=master
```

```
remote:
```

```
To https://gitlab.com/anishagarwal670/gitdemo.git
```

```
a4576a1..9b9389a master -> master
```

```
branch 'master' set up to track 'origin/master'.
```



```
MINGW64/c/Users/anish/GitDemo
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ # untrack the files from the index (keeps local files)
git rm --cached example.log
git rm -r --cached log
$ confirm they are removed from index (will show as deleted in working tree)
git status --short
fatal: pathspec 'example.log' did not match any files
fatal: pathspec 'log' did not match any files
?? .gitignore

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git add .gitignore
git commit -m "Ignore .log files and log/ directory; stop tracking logs"
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
[master 9b9389a] Ignore .log files and log/ directory; stop tracking logs
1 file changed, 2 insertions(+)
create mode 100644 .gitignore

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git branch --show-current
master
$ git remote -v
master
origin https://gitlab.com/anishagarwal670/gitdemo.git (fetch)
origin https://gitlab.com/anishagarwal670/gitdemo.git (push)

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git push -u origin $(git branch --show-current)
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), 325 bytes | 325.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: To create a merge request for master, visit:
remote: https://gitlab.com/anishagarwal670/gitdemo/-/merge_requests/new?merge_request%5Bsource_branch%5D=master
remote:
To https://gitlab.com/anishagarwal670/gitdemo.git
a4576a1..9b9389a master -> master
branch 'master' set up to track 'origin/master'.

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ ^C

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ ^
```

Activate Windows
Go to Settings to activate Windows.

27°C
Haze

Search

ENG
IN

02:53
09-08-2025

GIT - HOL 3

Please follow the instruction to complete the hands-on. Each instruction expects a command for the Git Bash.

Branching:

1. Create a new branch "**GitNewBranch**".
2. List all the local and remote branches available in the current trunk. Observe the "*" mark which denote the current pointing branch.
3. Switch to the newly created branch. Add some files to it with some contents.
4. Commit the changes to the branch.
5. Check the status with "**git status**" command.

Merging:

1. Switch to the master
2. List out all the differences between trunk and branch. These provide the differences in command line interface.
3. List out all the visual differences between master and branch using **P4Merge tool**.
4. Merge the source branch to the trunk.
5. Observe the logging after merging using "**git log --oneline --graph --decorate**"
6. Delete the branch after merging with the trunk and observe the git status.

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git branch GitNewBranch
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git branch -a
  GitNewBranch
* master
remotes/origin/master
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git checkout GitNewBranch
Switched to branch 'GitNewBranch'
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (GitNewBranch)
$ echo "This is a file in GitNewBranch" > branchfile1.txt
echo "Another file in GitNewBranch" > branchfile2.txt
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (GitNewBranch)
$ git add branchfile1.txt branchfile2.txt
git commit -m "Add two files in GitNewBranch"
```

warning: in the working copy of 'branchfile1.txt', LF will be replaced by CRLF the next time Git touches it

warning: in the working copy of 'branchfile2.txt', LF will be replaced by CRLF the next time Git touches it

[GitNewBranch dec1ed5] Add two files in GitNewBranch

2 files changed, 2 insertions(+)

create mode 100644 branchfile1.txt

create mode 100644 branchfile2.txt

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (GitNewBranch)

\$ git status

On branch GitNewBranch

nothing to commit, working tree clean

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (GitNewBranch)

\$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$ git branch GitNewBranch

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$ git branch -a

GitNewBranch

* master

remotes/origin/master

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$ git checkout GitNewBranch

Switched to branch 'GitNewBranch'

bash: syntax error near unexpected token `('

bash: \$: command not found

bash: syntax error near unexpected token `('

bash: \$: command not found

bash: GitNewBranch: command not found

bash: example.log: command not found

bash: remotes/origin/master: No such file or directory

bash: syntax error near unexpected token `('

bash: \$: command not found

bash: Switched: command not found

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$ git diff master GitNewBranch

diff --git a/branchfile1.txt b/branchfile1.txt

new file mode 100644

index 00000000..70ab14d

--- /dev/null

+++ b/branchfile1.txt

@@ -0,0 +1 @@

+This is a file in GitNewBranch

diff --git a/branchfile2.txt b/branchfile2.txt

new file mode 100644

index 00000000..4285005

--- /dev/null

+++ b/branchfile2.txt

@@ -0,0 +1 @@

+Another file in GitNewBranch

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$ git difftool master GitNewBranch

This message is displayed because 'diff.tool' is not configured.

See 'git difftool --tool-help' or 'git help config' for more details.

'git difftool' will now attempt to use one of the following tools:

opendiff kdiff3 tkdiff xxdiff meld kompare gvimdiff diffuse diffmerge ecmerge p4merge araxis
bc codecompare smerge emerge vimdiff nvimdiff

Viewing (1/2): 'branchfile1.txt'

Launch 'vimdiff' [Y/n]? y

2 files to edit

This message is displayed because 'diff.tool' is not configured.

See 'git difftool --tool-help' or 'git help config' for more details.

'git difftool' will now attempt to use one of the following tools:

opendiff kdiff3 tkdiff xxdiff meld kompare gvimdiff diffuse diffmerge ecmerge p4merge araxis
bc codecompare smerge emerge vimdiff nvimdiff

Viewing (2/2): 'branchfile2.txt'

Launch 'vimdiff' [Y/n]? y

2 files to edit

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$ git merge GitNewBranch

Updating 9b9389a..dec1ed5

Fast-forward

branchfile1.txt | 1 +

branchfile2.txt | 1 +

2 files changed, 2 insertions(+)

create mode 100644 branchfile1.txt

create mode 100644 branchfile2.txt

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$ git log --oneline --graph --decorate --all

* dec1ed5 (HEAD -> master, GitNewBranch) Add two files in GitNewBranch

* 9b9389a (origin/master) Ignore .log files and log/ directory; stop tracking logs

* a4576a1 Initial commit - Created welcome.txt - Added welcome message

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

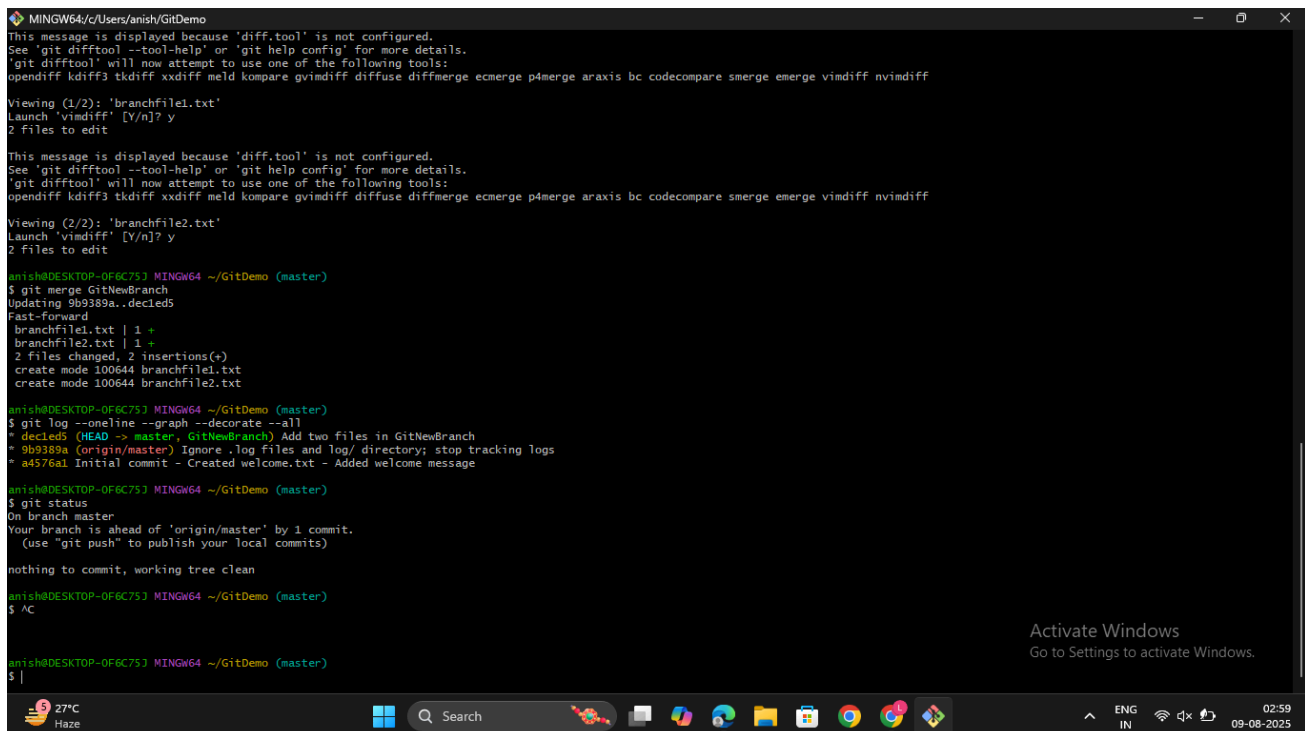
\$ git status

On branch master

Your branch is ahead of 'origin/master' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working tree clean



```
MINGW64/C:/Users/anish/GitDemo
This message is displayed because 'diff.tool' is not configured.
See 'git difftool --tool-help' or 'git help config' for more details.
'git difftool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld kompare gvimdiff diffuse diffmerge ecmerge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff

Viewing (1/2): 'branchfile1.txt'
Launch 'vimdiff' [Y/n]? y
2 files to edit

This message is displayed because 'diff.tool' is not configured.
See 'git difftool --tool-help' or 'git help config' for more details.
'git difftool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld kompare gvimdiff diffuse diffmerge ecmerge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff

Viewing (2/2): 'branchfile2.txt'
Launch 'vimdiff' [Y/n]? y
2 files to edit

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git merge GitNewBranch
Updating 9b9389a..dec1ed5
Fast-forward
 branchfile1.txt | 1 +
 branchfile2.txt | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 branchfile1.txt
 create mode 100644 branchfile2.txt

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git log --oneline --graph --decorate --all
* dec1ed5 (HEAD -> master, GitNewBranch) Add two files in GitNewBranch
* 9b9389a (origin/master) Ignore .log files and log/ directory; stop tracking logs
* a4576a1 Initial commit - Created welcome.txt - Added welcome message

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ ^C

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ |
```

Activate Windows
Go to Settings to activate Windows.

27°C Haze 02:59 09-08-2025

GIT - HOL 4

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
```

```
$ git checkout master      # switch to master
```

```
git status                # should say "working tree clean"
```

```
Already on 'master'
```

```
Your branch is ahead of 'origin/master' by 1 commit.
```

```
(use "git push" to publish your local commits)
```

```
On branch master
```

```
Your branch is ahead of 'origin/master' by 1 commit.
```

```
(use "git push" to publish your local commits)
```

```
nothing to commit, working tree clean
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
```

```
$ git branch GitWork      # create branch
```

```
git checkout GitWork      # switch to it
```

```
echo "<message>Hello from GitWork branch</message>" > hello.xml
```

```
Switched to branch 'GitWork'
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (GitWork)
```

```
$ echo "<update>Branch content update</update>" >> hello.xml
```

```
git status                # should show hello.xml as modified
```

```
On branch GitWork
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
hello.xml
```

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

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (GitWork)
```

```
$ git add hello.xml
```

```
git commit -m "Add and update hello.xml in GitWork branch"
```

```
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it
```

```
[GitWork 4d61d8c] Add and update hello.xml in GitWork branch
```

```
1 file changed, 2 insertions(+)
```

```
create mode 100644 hello.xml
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (GitWork)
```

```
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ echo "<message>Hello from master branch</message>" > hello.xml
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git add hello.xml
git commit -m "Add hello.xml in master with different content"
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git
touches it
[master cb33cfe] Add hello.xml in master with different content
1 file changed, 1 insertion(+)
create mode 100644 hello.xml
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git log --oneline --graph --decorate --all
* cb33cfe (HEAD -> master) Add hello.xml in master with different content
| * 4d61d8c (GitWork) Add and update hello.xml in GitWork branch
|/
* dec1ed5 (GitNewBranch) Add two files in GitNewBranch
* 9b9389a (origin/master) Ignore .log files and log/ directory; stop tracking logs
* a4576a1 Initial commit - Created welcome.txt - Added welcome message
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git diff master GitWork
diff --git a/hello.xml b/hello.xml
index 82868a7..236917a 100644
--- a/hello.xml
+++ b/hello.xml
@@ -1,2 @@
-<message>Hello from master branch</message>
+<message>Hello from GitWork branch</message>
+<update>Branch content update</update>
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git config --global diff.tool p4merge
git config --global difftool.p4merge.path "C:/Program Files/Perforce/p4merge.exe"
git difftool master GitWork
```

Viewing (1/1): 'hello.xml'

Launch 'p4merge' [Y/n]? y

The diff tool p4merge is not available as 'C:/Program Files/Perforce/p4merge.exe'

fatal: external diff died, stopping at hello.xml

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$ git merge GitWork

Auto-merging hello.xml

CONFLICT (add/add): Merge conflict in hello.xml

Automatic merge failed; fix conflicts and then commit the result.

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master|MERGING)

\$ git mergetool

This message is displayed because 'merge.tool' is not configured.

See 'git mergetool --tool-help' or 'git help config' for more details.

'git mergetool' will now attempt to use one of the following tools:

opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecmerge p4merge

araxis bc codecompare smerge emerge vimdiff nvimdiff

Merging:

hello.xml

Normal merge conflict for 'hello.xml':

{local}: created file

{remote}: created file

Hit return to start merge resolution tool (vimdiff):

3 files to edit

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master|MERGING)

\$ git commit -m "Merge GitWork into master and resolve conflict in hello.xml"

[master 2d0ffa0] Merge GitWork into master and resolve conflict in hello.xml

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$ git status

On branch master

Your branch is ahead of 'origin/master' by 4 commits.

(use "git push" to publish your local commits)

Untracked files:

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

hello.xml.orig

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

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

```
$ touch .gitignore
```

```
echo "*.~" >> .gitignore      # example for P4Merge backup files
```

```
git add .gitignore
```

```
git commit -m "Ignore merge backup files"
```

warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

```
[master 11cf603] Ignore merge backup files
```

```
1 file changed, 1 insertion(+)
```

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

```
$ git branch -a
```

```
GitNewBranch
```

```
GitWork
```

```
* master
```

```
remotes/origin/master
```

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

```
$ git branch -d GitWork
```

Deleted branch GitWork (was 4d61d8c).

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

```
$ git log --oneline --graph --decorate
```

```
* 11cf603 (HEAD -> master) Ignore merge backup files
```

```
* 2d0ffa0 Merge GitWork into master and resolve conflict in hello.xml
```

```
|\
```

```
| * 4d61d8c Add and update hello.xml in GitWork branch
```

```
* | cb33cfe Add hello.xml in master with different content
```

```
|/
```

```
* dec1ed5 (GitNewBranch) Add two files in GitNewBranch
```

```
* 9b9389a (origin/master) Ignore .log files and log/ directory; stop tracking logs
```

```
* a4576a1 Initial commit - Created welcome.txt - Added welcome message
```

GIT - HOL 5

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ pwd
git rev-parse --is-inside-work-tree # prints "true" if inside a git repo
git remote -v                      # shows remote names & URLs (ensure origin exists)
/c/Users/anish/GitDemo
true
origin https://gitlab.com/anishagarwal670/gitdemo.git (fetch)
origin https://gitlab.com/anishagarwal670/gitdemo.git (push)
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git checkout master              # switch to master (or: git switch master)
git status --porcelain            # blank output = clean working tree
Already on 'master'
Your branch is ahead of 'origin/master' by 5 commits.
(use "git push" to publish your local commits)
?? hello.xml.orig
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git branch -a
  GitNewBranch
* master
  remotes/origin/master
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git fetch origin
git pull origin master

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 (from 0)
Unpacking objects: 100% (3/3), 2.74 KiB | 133.00 KiB/s, done.
From https://gitlab.com/anishagarwal670/gitdemo
 * [new branch]    main    -> origin/main
From https://gitlab.com/anishagarwal670/gitdemo
 * branch          master  -> FETCH_HEAD
Already up to date.
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git fetch origin
```

```
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ # If branch exists locally:
git branch --list Git-T03-HOL_002
```

```
# If listed, switch to it:
git checkout Git-T03-HOL_002
```

If NOT listed but exists on remote, create & switch:

```
git checkout -b Git-T03-HOL_002 origin/Git-T03-HOL_002
```

error: pathspec 'Git-T03-HOL_002' did not match any file(s) known to git

fatal: 'origin/Git-T03-HOL_002' is not a commit and a branch 'Git-T03-HOL_002' cannot be created from it

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$ # compare remote branch (origin/Git-T03-HOL_002) with local Git-T03-HOL_002

```
git log --oneline origin/Git-T03-HOL_002..Git-T03-HOL_002
```

fatal: ambiguous argument 'origin/Git-T03-HOL_002..Git-T03-HOL_002': unknown revision or path not in the working tree.

Use '--' to separate paths from revisions, like this:

```
'git <command> [<revision>...] -- [<file>...']
```

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

```
$ git push -u origin Git-T03-HOL_002
```

error: src refs spec Git-T03-HOL_002 does not match any

error: failed to push some refs to 'https://gitlab.com/anishagarwal670/gitdemo.git'

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)

\$ # update remote refs locally

```
git fetch origin
```

check that remote branch exists and shows the new commit hash

```
git ls-remote --heads origin Git-T03-HOL_002
```

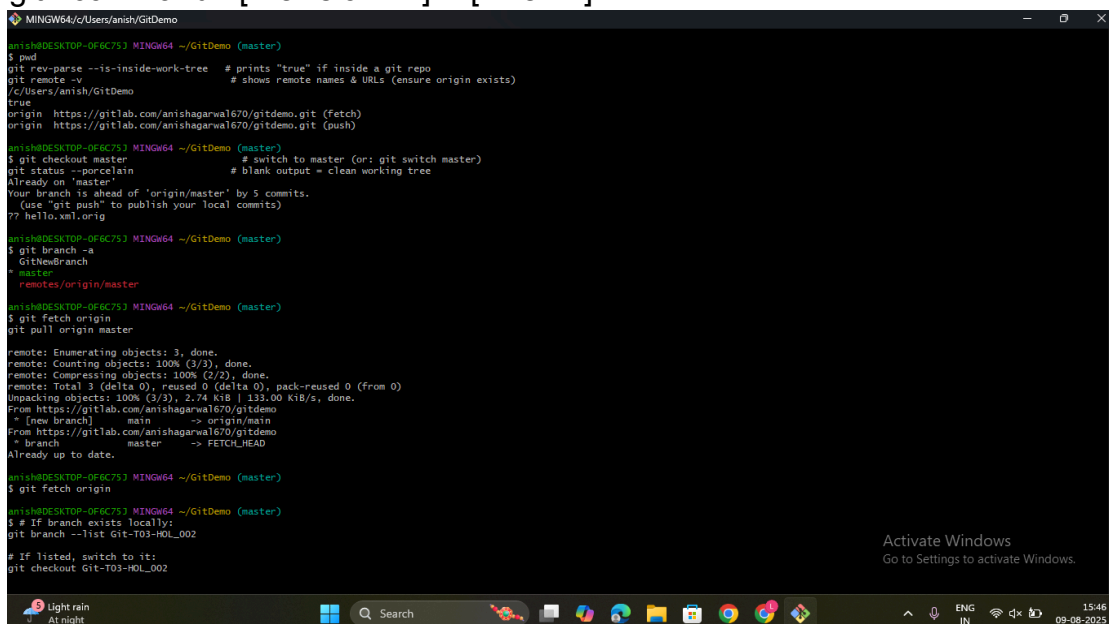
double-check there are no local-only commits remaining:

```
git log --oneline origin/Git-T03-HOL_002..Git-T03-HOL_002
```

fatal: ambiguous argument 'origin/Git-T03-HOL_002..Git-T03-HOL_002': unknown revision or path not in the working tree.

Use '--' to separate paths from revisions, like this:

```
'git <command> [<revision>...] -- [<file>...']
```



```
MINGW64/c/Users/anish/GitDemo
anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ pwd
~/GitDemo
$ git rev-parse --is-inside-work-tree
true
$ git remote -v
origin https://gitlab.com/anishagarwal670/gitdemo.git (fetch)
origin https://gitlab.com/anishagarwal670/gitdemo.git (push)

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git checkout master
git status --porcelain
Already on 'master'
Your branch is ahead of 'origin/master' by 5 commits.
  (use "git push" to publish your local commits)
?? hello.txt.orig

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git branch -a
* GitNewBranch
* master
  remotes/origin/master

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git fetch origin
git pull origin master
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 (from 0)
Unpacking objects: 100% (3/3), 2.74 KiB | 133.00 KiB/s, done.
From https://gitlab.com/anishagarwal670/gitdemo
+ [new branch]      main       -> origin/main
From https://gitlab.com/anishagarwal670/gitdemo
+ branch            master     -> FETCH_HEAD
Already up to date.

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ git fetch origin

anish@DESKTOP-0F6C75J MINGW64 ~/GitDemo (master)
$ # If branch exists locally:
git branch --list Git-T03-HOL_002

# If listed, switch to it:
git checkout Git-T03-HOL_002
```

Activate Windows
Go to Settings to activate Windows.

Light rain
At night

Search

ENG
IN

15:46
09-08-2025