

13/01/2025

Git: It is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git Features:

Distributed: Distributed systems are those which allow the users to perform work on a project from all over the world.

Compatible:

Non-Linear: Git allows users from all over the world to perform operations on a project remotely. A user can pick up any part of the project and do the required operation and then further update the project.

Branching: Git allows its users to work on a line that runs parallel to the main project files. These lines are called branches.

Light Weight: Git stores all the data from the central repository on to the local repository while cloning is done.

Speed: Since Git stores all the data related to a project in the local repository by the process of cloning, it is very much efficient to fetch data from the local repository instead of doing the same from the remote repository.

Open Source:

Reliable: Providing a central repository that is being cloned each time a User performs the Pull operation, the data of the central repository is always being backed up in every collaborator's local repository

Secure: Git keeps a record of all the commits done by each of the collaborators on the local copy of the developer. A log file is maintained and is pushed to the central repository each time the push operation is performed.

Economical: Git is released under the General Public's License(GPL) and hence it is available for free.

VCS:

SCM:

SCM refers to tools and practices used to manage changes in source code, track versions, and handle code revisions.

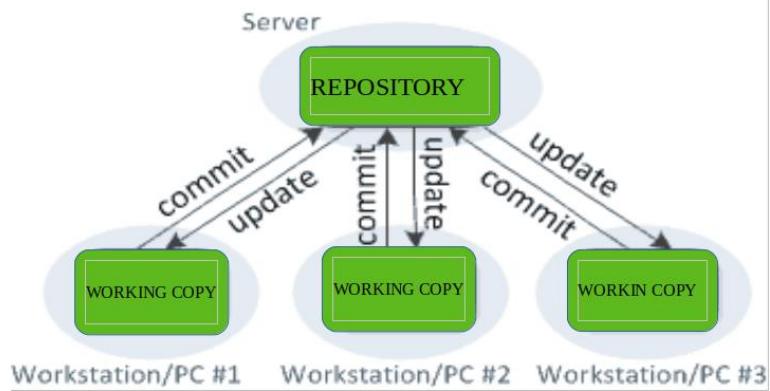
SCM systems help manage the history of code changes and collaborate among multiple developers working on a project.

Types of Repositories

Local Repository

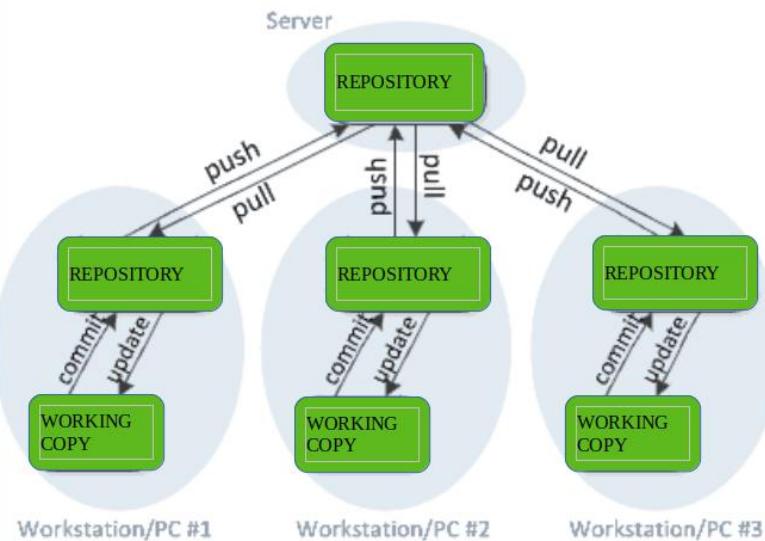
Central Repository

Centralized version control



Distributed Repository:

Distributed version control



14/01/2025

Commands	Description
<code>git add <file></code>	Adds a specific file to the staging area.
<code>git add . or git add –all</code>	Adds all modified and new files to the staging area.
<code>git status</code>	Shows the current state of your repository, including tracked and untracked files, modified files, and branch information.
<code>git status –ignored</code>	Displays ignored files in addition to the regular status output.
<code>git diff</code>	Shows the changes between the working directory and the staging area (index).
<code>git diff <commit1> <commit2></code>	Displays the differences between two commits.
<code>git diff –staged or git diff –cached</code>	Displays the changes between the staging area (index) and the last commit.
<code>git diff HEAD</code>	Display the difference between the current directory and the last commit
<code>git commit</code>	Creates a new commit with the changes in the staging area and opens the default text editor for adding a commit message.
<code>git commit -m “<message>” or git commit –message “<message>”</code>	Creates a new commit with the changes in the staging area and specifies the commit message inline.

Commands	Description
<code>git commit -a or git commit –all</code>	Commits all modified and deleted files in the repository without explicitly using git add to stage the changes.

Getting & Creating Projects

Command	Description
<code>git init</code>	Initialize a local Git repository
<code>git clone ssh://git@github.com/ [username] / [repository-name].git</code>	Create a local copy of a remote repository

Basic Snapshotting

Command	Description
<code>git status</code>	Check status
<code>git add [file-name.txt]</code>	Add a file to the staging area
<code>git add -A</code>	Add all new and changed files to the staging area
<code>git commit -m "[commit message]"</code>	Commit changes
<code>git rm -r [file-name.txt]</code>	Remove a file (or folder)
<code>git remote -v</code>	View the remote repository of the currently working file or directory

Branching & Merging

Command	Description
<code>git branch</code>	List branches (the asterisk denotes the current branch)
<code>git branch -a</code>	List all branches (local and remote)
<code>git branch [branch name]</code>	Create a new branch
<code>git branch -d [branch name]</code>	Delete a branch
<code>git push origin --delete [branch name]</code>	Delete a remote branch
<code>git checkout -b [branch name]</code>	Create a new branch and switch to it

Command	Description
git checkout -b [branch name] origin/[branch name]	Clone a remote branch and switch to it
git branch -m [old branch name] [new branch name]	Rename a local branch
git checkout [branch name]	Switch to a branch
git checkout -	Switch to the branch last checked out
git checkout -- [file-name.txt]	Discard changes to a file
git merge [branch name]	Merge a branch into the active branch
git merge [source branch] [target branch]	Merge a branch into a target branch
git stash	Stash changes in a dirty working directory
git stash clear	Remove all stashed entries
git stash pop	Apply latest stash to working directory

Sharing & Updating Projects

Command	Description
git push origin [branch name]	Push a branch to your remote repository
git push -u origin [branch name]	Push changes to remote repository (and remember the branch)
git push	Push changes to remote repository (remembered branch)
git push origin --delete [branch name]	Delete a remote branch
git pull	Update local repository to the newest commit
git pull origin [branch name]	Pull changes from remote repository
git remote add origin ssh://git@github.com/[username]/[repository-name].git	Add a remote repository
git remote set-url origin ssh://git@github.com/[username]/[repository-name].git	Set a repository's origin branch to SSH

Inspection & Comparison

Command	Description
git log	View changes
git log --summary	View changes (detailed)
git log --oneline	View changes (briefly)
git diff [source branch] [target branch]	Preview changes before merging

Command	Description
15/01/2024	
Restoring	
<code>git restore . for restoring from staging to working directory</code>	
<code>git restore -workspace for restoring from local repository to working directory</code>	
<code>git restore -staged for restoring from local repository to staging</code>	

Key Concepts:

Repository: It is a place to store and track the changes of files in software development

Commit: A commit is an action that saves changes made to the project's files to the repository. Each commit has a unique identifier and a message describing the changes

Branching: Creating separate line of development

Merging: merging allow multiple developers to work on different features or bug fixes simultaneously

Version History:

Git keeps a detailed **history** of all changes made to the codebase. This helps you track progress, go back to previous states, and find bugs introduced in past commits.

Benefits of SCM:

Collaboration: Repositories enable multiple developers to work on the same project, making it easier to contribute and collaborate

Version Control: Tracks all code changes, allowing developers to go back to previous versions if needed.

Code Review: Changes can be reviewed before merging into the main codebase.

Risk Mitigation: The ability to track changes and revert to previous versions plays a crucial role in risk minimization. By maintaining a comprehensive log of modifications, SCM tools help prevent the loss of valuable work and ensure that code changes are well-documented and traceable.

Improved Productivity: Developers can work in parallel, switch between versions, and merge changes easily.

Core Concepts:

Object based

Key Object Types:

Commit: A commit is like a **snapshot** of your project at a specific point in time. Each commit in Git records changes to the files and is assigned a unique ID (commit hash).

Tag: Git tags are markers used to highlight specific commits in the history of your Git repository.

Hooks:

It is a way to trigger other functions. Git comes with a set of script. Script can automatically run at every meaningful phase.

Index: File is where git stores staging area information

16/01/2025

git commit -- amend

This command helps to edit the commit message of the last message. The below screenshot displays the commit messaged modified.

```
MINGW64:/c/Users/Administrator/devops/git/gitpush
Date: Wed Jan 15 15:58:17 2025 +0530
    local

commit 8fac9c1d70357a9abaadf64e1e0256e45a32eb05
Author: Gowrisankar5877 <mgowrisankar5877@gmail.com>
Date: Wed Jan 15 15:54:29 2025 +0530

    file deletion

commit 085d16d8159d6e16669aa6e6502e32ea3b787db8 (feature1)
Author: Gowrisankar5877 <mgowrisankar5877@gmail.com>
Date: Wed Jan 15 15:48:28 2025 +0530

    Sample file

commit fa7a239bb72cb653f5b15c7fc5898b3cefbb5200 (origin/feature1)
Author: Gowrisankar5877 <mgowrisankar5877@gmail.com>
Date: Wed Jan 15 14:57:14 2025 +0530

    learningPR

commit a00033a91f99676bd9461ef5c4580d84d804e096 (origin/main)
Author: Gowrisankar5877 <mgowrisankar5877@gmail.com>
Date: Wed Jan 15 14:34:21 2025 +0530

    V1
~
~
~

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gitpush (main)
$ git log --oneline
9dc8098 (HEAD -> main) Commit Modified
d1da994 Adding to Local Repo
2bf48f8 local
8fac9c1 file deletion
085d16d (feature1) Sample file
fa7a239 (origin/feature1) learningPR
a00033a (origin/main) V1
```

Tagging

```
MINGW64:/c/Users/Administrator/devops/git/gitpush
cbfa4cf8 local
bfacc1 file deletion
085d16d (feature1) Sample file
fa7a239 (origin/feature1) learningPR
a00033a (origin/main) V1

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gitpush (main)
$ git tag

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gitpush (main)
$ git tag -a @author 9dc8098 -m gowrisankar

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gitpush (main)
$ git log
commit 9dc8098587920d9bdb2e9132b11eb99924e72b (HEAD -> main, tag: @author)
Author: Gowrisankar5877 mgowrisankar5877@gmail.com
Date:   Wed Jan 15 16:52:27 2025 +0530

    Commit Modified

commit d1da994acba6253ea9346acd8a5120c97a508569
Author: Gowrisankar5877 mgowrisankar5877@gmail.com
Date:   Wed Jan 15 16:01:20 2025 +0530

    Adding to Local Repo

commit 2bf48f8a0ba65bf6bdcc2d664981cc91c974aafb
Merge: 8fac9c1 085d16d
Author: Gowrisankar5877 mgowrisankar5877@gmail.com
Date:   Wed Jan 15 15:58:17 2025 +0530

    local

commit 8fac9c1d70357a9abaadf64e1e0256e45a32eb05
Author: Gowrisankar5877 mgowrisankar5877@gmail.com
Date:   Wed Jan 15 15:54:29 2025 +0530

    file deletion

commit 085d16d8159d6e16669aa6e6502e32ea1b787db8 (feature1)
Author: Gowrisankar5877 mgowrisankar5877@gmail.com
Date:   Wed Jan 15 15:48:28 2025 +0530

    Sample file

commit fa7a239bb72cb653f5b15c7fc5898b3cefb5200 (origin/feature1)

Windows Search Bar Type here to search Taskbar Icons System Tray
```

```
MINGW64:/c/Users/Administrator/devops/git/gitpush
commit a00033a91f99676b9461ef5c4580d84d904e096 (origin/main)
Author: Gowrisankar5877 mgowrisankar5877@gmail.com
Date:   Wed Jan 15 14:34:21 2025 +0530

    V1

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gitpush (main)
$ git show @author
tag @author
Tagger: Gowrisankar5877 mgowrisankar5877@gmail.com
Date:   Thu Jan 16 10:23:20 2025 +0530

gowrisankar

commit 9dc8098587920d9bdb2e9132b11eb99924e72b (HEAD -> main, tag: learning-tagging, tag: @author)
Author: Gowrisankar5877 mgowrisankar5877@gmail.com
Date:   Wed Jan 15 16:52:27 2025 +0530

    Commit Modified

diff --git a/.gitignore b/.gitignore
new file mode 100644
index 0000000..3810sec
--- /dev/null
+++ b/.gitignore
@@ -0,0 +1,24 @@
+# Compiled class file
+*.<class>
+
+# Log file
+*.log
+
+# BlueJ files
+*.ctxt
+
+## Mobile Tools for Java (J2ME)
+.mtj1.tmp/
+
+## Package Files #
+*.jar
+*.war
+*.ear
+*.tar
+*.tar.gz
+*.rar

Windows Search Bar Type here to search Taskbar Icons System Tray
```

```
MINGW64:/c/Users/Administrator/devops/git/gtpush
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gtpush (main)
$ git checkout feature
error: pathspec 'feature' did not match any file(s) known to git

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gtpush (main)
$ git checkout feature1
Switched to branch 'feature1'
Your branch is ahead of 'origin/feature1' by 1 commit.
  (use "git push" to publish your local commits)

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gtpush (feature1)
$ ls
abc.class logfile.log sample.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gtpush (feature1)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gtpush (main)
$ git branch -d feature1
warning: not deleting branch 'feature1' that is not yet merged to
  'refs/remotes/origin/feature1', even though it is merged to HEAD
error: the branch 'feature1' is not fully merged
hint: If you are sure you want to delete it, run 'git branch -D feature1'
hint: Disable this message with 'git config advice.forceDeleteBranch false'

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gtpush (main)
$ git merge feature1
Already up to date.

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gtpush (main)
$ git branch -d feature1
Deleted branch feature1 (was 085d16d).

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/gtpush (main)
$
```

Stashing

Reverting

```
MINGW64:/c/Users/Administrator/devops/git/revert
Administrator@a817d07c78135a7 MINGW64 ~/devops/git
$ cd revert

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert
$ git init
Initialized empty Git repository in C:/Users/Administrator/devops/git/revert/.git/
t/

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ echo "Hello, I am Gowri Sankar" >> file1.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ git commit -m "m1"
[master (root-commit) 0f7ed67] m1
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ echo "I am second message" >> file1.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ git commit -m "m2"
[master d10e037] m2
 1 file changed, 1 insertion(+)

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ echo "I am third message" >> file1.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ git commit -m "m3"
[master 68f42bd] m3
 1 file changed, 1 insertion(+)

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
```

```
MINGW64:/c/Users/Administrator/devops/git/revert
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ git commit -m "m3"
[master 68f42bd] m3
 1 file changed, 1 insertion(+)

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ cat file1.txt
Hello, I am Gowri Sankar
I am second message
I am third message

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ git log --oneline
68f42bd (HEAD -> master) m3
d10e037 m2
0f7ed67 m1

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master)
$ git revert HEAD-2
CONFLICT (modify/delete): file1.txt deleted in (empty tree) and modified in HEAD. Version HEAD of file1.txt left in tree.
error: could not revert 0f7ed67... m1
hint: after resolving the conflicts, mark them with
hint: 'git add/rm <pathspec>', then run
hint: 'git revert --continue'.
hint: you can instead skip this commit with "git revert --skip".
hint: to abort and get back to the state before "git revert",
hint: run "git revert --abort".
hint: Disable this message with "git config advice.mergeConflict false"

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ git status
On branch master
You are currently reverting commit 0f7ed67.
  (fix conflicts and run "git revert --continue")
  (use "git revert --skip" to skip this patch)
  (use "git revert --abort" to cancel the revert operation)

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add/mv <file>..." as appropriate to mark resolution)
    deleted by them: file1.txt

no changes added to commit (use "git add" and/or "git commit -a")

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)

Windows Search Type here to search 任务栏图标 14:38 ENG US 16-01-2025
```

```
MINGW64:/c/Users/Administrator/devops/git/revert
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ use "git revert --abort" to cancel the revert operation

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add/mv <file>..." as appropriate to mark resolution)
    deleted by them: file1.txt

no changes added to commit (use "git add" and/or "git commit -a")

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ cat file1.txt
Hello, I am Gowri Sankar
I am second message
I am third message

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ git log --oneline
68f42bd (HEAD -> master) m3
d10e037 m2
0f7ed67 m1

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ git restore --staged
fatal: you must specify path(s) to restore

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ git restore --staged file1.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ git status
On branch master
You are currently reverting commit 0f7ed67.
  (all conflicts fixed; run "git revert --continue")
  (use "git revert --skip" to skip this patch)
  (use "git revert --abort" to cancel the revert operation)

nothing to commit, working tree clean

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ cat file1.txt
Hello, I am Gowri Sankar
I am second message
I am third message

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ git add .

Windows Search Type here to search 任务栏图标 14:38 ENG US 16-01-2025
```

```
MINGW64:/c/Users/Administrator/devops/git/revert
I am third message

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ git add .

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ git commit -m "M4"
0 files changed, 0 insertions(+), 0 deletions(-)
branch master
You are currently reverting commit 0f7ad67.
(all conflicts fixed; run "git revert --continue")
Reverting "M3"

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ git revert 0f7ad67
Reverting "M2"

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/revert (master|REVERTING)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean
```

Reset:

Soft : moves the branch pointer (effects only in local repository)

You will remove the last commit from the current branch, but the file changes will stay in working tree and in index

Hard: it removes in all stages

Mixed: moves the branch pointer and

Default mode. You will still keep the changes in your working tree but not on index

```
MINGW64/c/Users/Administrator/devops/git/reset
Administrator@817d07c78135a7 MINGW64 ~/devops/git
$ mkdir reset

Administrator@817d07c78135a7 MINGW64 ~/devops/git
$ cd reset

Administrator@817d07c78135a7 MINGW64 ~/devops/git/reset
$ git init
initialized empty Git repository in C:/Users/Administrator/devops/git/reset/.git/

Administrator@817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ echo "M1" > file1.txt

Administrator@817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it

Administrator@817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git commit -m "M1"
[master (root-commit) deceac6] M1
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt

Administrator@817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git log --oneline
deceac6 (HEAD --> master) M1

Administrator@817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git diff

Administrator@817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git diff --staged

Administrator@817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset HEAD~1
fatal: ambiguous argument 'HEAD~1': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git [command] [<revision...>] -- [<file>...]''

Administrator@817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset --soft HEAD

Administrator@817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git diff --staged

Administrator@817d07c78135a7 MINGW64 ~/devops/git/reset (master)
```

```
MINGW64:/c/Users/Administrator/devops/git/reset
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset --soft HEAD
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git diff --staged
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git status
On branch master
nothing to commit, working tree clean
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset --mixed HEAD
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git diff --staged
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git status
On branch master
nothing to commit, working tree clean
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ echo "M2" >> file2.txt
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git add .
warning: in the working copy of 'file2.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git commit -m "M2"
[master cf862cc] M2
 1 file changed, 1 insertion(+)
 create mode 100644 file2.txt
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ echo "M3" >> file3.txt
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git add .
warning: in the working copy of 'file3.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git commit -m "M3"
[master a23aa52] M3
 1 file changed, 1 insertion(+)

MINGW64:/c/Users/Administrator/devops/git/reset
create mode 100644 file2.txt
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ echo "M3" >> file3.txt
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git add .
warning: in the working copy of 'file3.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git commit -m "M3"
[master a23aa52] M3
 1 file changed, 1 insertion(+)
 create mode 100644 file3.txt
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ echo "M4" >> file4.txt
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git add .
warning: in the working copy of 'file4.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git commit -m "M4"
[master af23ef] M4
 1 file changed, 1 insertion(+)
 create mode 100644 file4.txt
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git log --oneline
af23ef (HEAD -> master) M4
a23aa52 M3
cf862cc M2
deceac M1
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset --soft HEAD
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git status
On branch master
nothing to commit, working tree clean
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset --soft cf862cc
```

```
MINGW64:/c/Users/Administrator/devops/git/reset
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git log --oneline
af23ef (HEAD -> master) M4
a23aa52 M3
cf862cc M2
deceac M1
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset --soft HEAD
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git status
On branch master
nothing to commit, working tree clean
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset --soft cf862cc
```

```
MINGW64:/c/Users/Administrator/devops/git/reset
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset --soft cf862cc
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git log --oneline
cf862cc (HEAD -> master) M2
deceac4 MI

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  file3.txt
    new file:  file4.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git commit -m "both files added"
[master c00d3e7] both files added
 2 files changed, 2 insertions(+)
 create mode 100644 file3.txt
 create mode 100644 file4.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git log --oneline
c00d3e7 (HEAD -> master) both files added
cf862cc M2
deceac4 MI

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset head-1

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3.txt
    file4.txt

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

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git add .
warning: in the working copy of 'file3.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'file4.txt', LF will be replaced by CRLF the next time Git touches it

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git commit -m "M8"
[master 7b12ee7] M8
 2 files changed, 2 insertions(+)
 create mode 100644 file3.txt
 create mode 100644 file4.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset --hard HEAD-1
HEAD is now at cf862cc M2

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git status
On branch master
nothing to commit, working tree clean

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git log --oneline
cf862cc (HEAD -> master) M2
deceac4 MI

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ ls
file1.txt  file2.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ |
```

```
MINGW64:/c/Users/Administrator/devops/git/reset
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset head-1

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3.txt
    file4.txt

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

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git add .
warning: in the working copy of 'file3.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'file4.txt', LF will be replaced by CRLF the next time Git touches it

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git commit -m "M8"
[master 7b12ee7] M8
 2 files changed, 2 insertions(+)
 create mode 100644 file3.txt
 create mode 100644 file4.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git reset --hard HEAD-1
HEAD is now at cf862cc M2

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git status
On branch master
nothing to commit, working tree clean

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ git log --oneline
cf862cc (HEAD -> master) M2
deceac4 MI

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ ls
file1.txt  file2.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/reset (master)
$ |
```

Rebase

Revert

Reset

Restore

Tracking remote repository

Tracking: git checkout –tracking

```
MINGW64/c/Users/Administrator/devops/git/publicgit
$ git pull
From github.com:Gowrisankar5877/publicgit
 * [new branch] B1      -> origin/main
Merge made by the 'ort' strategy.
 hello_world | 1 +-
 1 file changed, 1 insertion(+)
 create mode 100644 hello_world

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/publicgit (main)
$ git branch
 * feature1
   main

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/publicgit (main)
$ git pull
Already up to date.

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/publicgit (main)
$ git branch
 * feature1
   main

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/publicgit (main)
$ git checkout --track remote/B1
fatal: 'remote/B1' is not a commit and a branch 'B1' cannot be created from it

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/publicgit (main)
$ git checkout --track origin/B1
branch 'B1' set up to track 'origin/B1'.
Switched to a new branch 'B1'

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/publicgit (B1)
$ git branch
 * B1
   feature1
   main

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/publicgit (B1)
$ ls
README.md  SampleFile.txt  hello_world  publicGit/  sampleclass.class.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/publicgit (B1)
$ |
```

```
MINGW64/c/Users/Administrator/devops/git/dev/GithubIntegrationWithJenkins

Administrator@a817d07c78135a7 MINGW64 ~
$ cd devops

Administrator@a817d07c78135a7 MINGW64 ~/devops
$ cd git

Administrator@a817d07c78135a7 MINGW64 ~/devops/git
$ mkdir gitJenkins

Administrator@a817d07c78135a7 MINGW64 ~/devops/git
$ git clone git@github.com:Gowrisankar5877/GithubIntegrationWithJenkins.git
Cloning into 'GithubIntegrationWithJenkins'...
warning: You appear to have cloned an empty repository.

Administrator@a817d07c78135a7 MINGW64 ~/devops/git
$ echo "file1" >> file1.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git
$ git add .
fatal: not a git repository (or any of the parent directories): .git

Administrator@a817d07c78135a7 MINGW64 ~/devops/git
$ git init
Initialized empty Git repository in C:/Users/Administrator/devops/git/.git/

Administrator@a817d07c78135a7 MINGW64 ~/devops/git (master)
$ cd GithubIntegrationWithJenkins

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationWithJenkins (main)
$ cd ..
```

```
MINGW64:/c/Users/Administrator/devops/git/dev/GithubIntegrationWithJenkins
$ rm -rf .git

Administrator@a817d07c78135a7 MINGW64 ~/devops/git
$ cd GithubIntegrationwithJenkins/

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ echo "file1" >> file1.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ git commit -m "File 1 added"
[main (root-commit) 13a5ad8] File 1 added
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 221 bytes | 221.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Gowrisankar5877/GithubIntegrationwithJenkins.git
 * [new branch]      main -> main

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ echo "file2" >> file2.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
```

```
MINGW64:/c/Users/Administrator/devops/git/dev/GithubIntegrationWithJenkins
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ echo "file2" >> file2.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ git add .
warning: in the working copy of 'file2.txt', LF will be replaced by CRLF the next time Git touches it

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ git commit -m "file2 added"
[main e772733] file2 added
 1 file changed, 1 insertion(+)
 create mode 100644 file2.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (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), 276 bytes | 276.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Gowrisankar5877/GithubIntegrationwithJenkins.git
 13a5ad8..e772733 main -> main

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ echo "file3" >> file3.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ git add .
warning: in the working copy of 'file3.txt', LF will be replaced by CRLF the next time Git touches it
```

```
MINGW64:/c/Users/Administrator/devops/git/dev/GithubIntegrationWithJenkins
To github.com:Gowrisankar5877/GithubIntegrationwithJenkins.git
  13a5ad8..e772733  main -> main

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ echo "file3" >> file3.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ git add .
warning: in the working copy of 'file3.txt', LF will be replaced by CRLF the next time Git touches it

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ git commit -m "file3 added"
[main ef37cea] file3 added
 1 file changed, 1 insertion(+)
 create mode 100644 file3.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (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), 302 bytes | 302.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Gowrisankar5877/GithubIntegrationwithJenkins.git
 e772733..ef37cea main -> main

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/GithubIntegrationwithJenkins (main)
$ git log --oneline
ef37cea (HEAD -> main, origin/main) file3 added
e772733 file2 added
```

```
MINGW64:/c/Users/Administrator/devops/git/dev/GithubIntegrationWithJenkins
Administrator@a817d07c78135a7 MINGW64 ~/devops/git
$ mkdir dev

Administrator@a817d07c78135a7 MINGW64 ~/devops/git
$ git clone git@github.com:Gowrisankar5877/GithubIntegrationwithJenkins.git
fatal: destination path 'GithubIntegrationwithJenkins' already exists and is not an empty directory.

Administrator@a817d07c78135a7 MINGW64 ~/devops/git
$ cd dev

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev
$ git clone git@github.com:Gowrisankar5877/GithubIntegrationwithJenkins.git\
>
Cloning into 'GithubIntegrationwithJenkins'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 12 (delta 2), reused 7 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (12/12), done.
Resolving deltas: 100% (2/2), done.

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev
$ 

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev
$ git log --oneline main origin/feature1
fatal: not a git repository (or any of the parent directories): .git

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev
$ git diff origin/feature1
warning: Not a git repository. Use --no-index to compare two paths outside a working tree
```

```
MINGW64:/c/Users/Administrator/devops/git/dev/GithubIntegrationWithJenkins
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev
$ cd GithubIntegrationwithJenkins/

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ git diff origin/feature1
diff --git a/file4.txt b/file4.txt
deleted file mode 100644
index d09c6c0..0000000
--- a/file4.txt
+++ /dev/null
@@ -1 +0,0 @@
-we are creating a new file

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ git log --oneline main..origin/feature1
fe8f1bb (origin/feature1) Create file4.txt

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ git fetch origin

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ git branch
* main

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ git fetch --all

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ git branch
* main

Windows PowerShell Type here to search ENG 17:45 US 20-01-2025
```

```
MINGW64:/c/Users/Administrator/devops/git/dev/GithubIntegrationWithJenkins
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ git fetch --all

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ git branch
* main

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ git diff origin/feature1-p
fatal: ambiguous argument 'origin/feature1-p': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ git diff origin/feature1 -p
diff --git a/file4.txt b/file4.txt
deleted file mode 100644
index d09c6c0..0000000
--- a/file4.txt
+++ /dev/null
@@ -1 +0,0 @@
-we are creating a new file

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ git fetch origin/feature1
fatal: 'origin/feature1' 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.

Windows PowerShell Type here to search ENG 17:46 US 20-01-2025
```

```
MINGW64:/c/Users/Administrator/devops/git/dev/GithubIntegrationWithJenkins
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationWithJenkins (main)
$ git fetch origin feature1
From github.com:Gowrisankar5877/GithubIntegrationWithJenkins
 * branch      feature1    -> FETCH_HEAD

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationWithJenkins (main)
$ git branch
* main

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationWithJenkins (main)
$ git log --oneline
ef37cea (HEAD -> main, origin/main, origin/HEAD) file3 added
e772733 file2 added
13a5ad8 File 1 added

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationWithJenkins (main)
$ git diff origin feature1
fatal: ambiguous argument 'feature1': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationWithJenkins (main)
$ git diff origin/feature1
diff --git a/file4.txt b/file4.txt
deleted file mode 100644
index d09c6c0..0000000
--- a/file4.txt
+++ /dev/null
@@ -1 +0,0 @@
-we are creating a new file
Windows Search Type here to search 任务栏图标 ENG 17:46
US 20-01-2025
```

```
MINGW64:/c/Users/Administrator/devops/git/dev/hooks/hooksautomation
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/GithubIntegrationwithJenkins (main)
$ cd ..
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev
$ mkdir hooks
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev
$ cd hooks

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks
$ git clone git@github.com:Gowrisankar5877/hooksautomation.git
Cloning into 'hooksautomation'...
warning: You appear to have cloned an empty repository.

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks
$ cd hooksautomation/
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation (main)
$ cd .git/
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation/.git (GIT_DIR!)
$ ls
HEAD config description hooks/ info/ objects/ refs/
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation/.git (GIT_DIR!)
$ cd hooks

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation/.git/hooks (GIT_DIR!)
$ code post-commit

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation/.git/hooks (GIT_DIR!)
```

```
MINGW64:/c/Users/Administrator/devops/git/dev/hooks/hooksautomation
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation/.git/hooks (GIT_DIR!)
$ code post-commit

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation/.git/hooks (GIT_DIR!)
$ ls
applypatch-msg.sample*    post-update.sample*    pre-push.sample*    push-to-checkout.sample*
commit-msg.sample*        pre-applypatch.sample*  pre-rebase.sample*  sendemail-validate.sample*
fsmonitor-watchman.sample*  pre-commit.sample*   pre-receive.sample*  update.sample*
post-commit               pre-merge-commit.sample*  prepare-commit-msg.sample*

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation/.git/hooks (GIT_DIR!)
$ code commit-msg.sample

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation/.git/hooks (GIT_DIR!)
$ cd ../..
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation (main)
$ echo "file1 added with automation using hooks" >> file1.txt
Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation (main)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it

Administrator@a817d07c78135a7 MINGW64 ~/devops/git/dev/hooks/hooksautomation (main)
$ git commit -m "Automatically the hookpushes the file"
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 271 bytes | 271.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Gowrisankar5877/hooksautomation.git
 * [new branch]      main -> main
```

