

SCM (Source Code Management) and Git

1. SCM (Source Code Management):

- **What is 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.

Key Benefits of Using SCM:

- **Collaboration:** Multiple developers can work on the same project simultaneously without overwriting each other's work.
 - **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:** Prevents loss of code and ensures that any mistakes or bugs can be traced back and corrected.
 - **Improved Productivity:** Developers can work in parallel, switch between versions, and merge changes easily.
-

Git: A Distributed SCM Tool

1. Git Overview:

- **What is Git?**
 - Git is a **distributed version control system**. It helps track changes to code, manage versions, and collaborate on development.
 - Git is **incremental**, meaning it stores only the changes made (not the whole file each time).

2. Core Concepts in Git:

- **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).
- **Incremental Versioning:**
 - Git tracks only changes (additions, deletions, modifications) rather than copying the entire file, making it more efficient.
 - **Example:**
If you're traveling from **Tvm** (Thiruvananthapuram) to **Kazhakootam**, you only note

the changes (Tvm to Kazhakootam). On another trip, you record all stops (Tvm → Chakka → Infosys → Tvm) but avoid repeating the same information.

- **Branching:**
 - Git allows developers to **branch** off the main codebase, make changes, and then merge them back. This lets developers work on different features simultaneously without interfering with each other's code.
 - **Feature Branching:** Each new feature is developed in a separate branch, keeping the main codebase stable.
- **Repository:**
 - A Git repository is where your project's **version history** is stored. It holds all the commits, branches, and information about your project.
- **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 Using Git:

- **Collaboration:** Multiple developers can contribute to the same project without conflict.
- **Version Control:** Git keeps track of changes and allows reverting to previous versions of the code.
- **Code Review:** Changes can be reviewed before being merged into the main branch.
- **Risk Mitigation:** Mistakes are easier to identify and fix since all changes are recorded.
- **Improved Productivity:** Git allows for parallel development, easy switching between versions, and faster project management.

Installing Git:

For Ubuntu (Debian-based systems):

- To install Git:
- `sudo apt install git -y`

For Fedora (Red Hat-based systems):

- To install Git:
- `sudo dnf install git -y`

Basic Git Commands:

1. **git init:** Initializes a new Git repository in your project directory.

2. **git add .**: Stages all changes in your project for the next commit.
3. **git commit -m "message"**: Commits your staged changes with a message describing the changes.
4. **git push**: Pushes the local commits to the remote repository (e.g., GitHub).
5. **git pull**: Pulls the latest changes from the remote repository.
6. **git status**: Shows the current status of your files (modified, staged, untracked, etc.).
7. **git log --oneline**: Shows a simplified commit history with each commit's hash and message.
8. **git branch**: Lists the branches in your repository.
9. **git checkout branch-name**: Switches to the specified branch.
10. **git merge branch-name**: Merges changes from the specified branch into the current branch.

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

Sharing & Updating Projects

Command	Description
<code>git push origin [branch name]</code>	Push a branch to your remote repository
<code>git push -u origin [branch name]</code>	Push changes to remote repository (and remember the branch)
<code>git push</code>	Push changes to remote repository (remembered)

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

Inspection & Comparison

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

Restoring

`git restore .` for restoring from staging to working directory

`git restore --workspace` for restoring from local repository to working directory

`git restore --staged` for restoring from local repository to staging

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

Stashing

```
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git stash save "first stash"
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state on master: first stash

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ cat file1.txt
hi

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git pull
Updating 3fe82c1..be9e917
Fast-forward
 file1.txt | 1 +
 1 file changed, 1 insertion(+)

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ cat file1.txt
hi
guys

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git stash list
stash@{0}: On master: first stash

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git stash apply
Auto-merging file1.txt
CONFLICT (content): Merge conflict in file1.txt
On branch master
Your branch is up to date with 'origin/master'.

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
    both modified:   file1.txt

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

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ cat file1.txt
<<<<<< Updated upstream
```

```
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git commit -m "hi added"
[master (root-commit) 3fe82c1] hi added
1 file changed, 1 insertion(+)
create mode 100644 file1.txt

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git remote add origin git@github.com:devanarayanantm/stash.git

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 214 bytes | 214.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:devanarayanantm/stash.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ echo welcome >> file1.txt

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ echo welcome > file1.txt

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 904 bytes | 75.00 KiB/s, done.
From github.com:devanarayanantm/stash
  3fe82c1..be9e917 master -> origin/master
Updating 3fe82c1..be9e917
error: Your local changes to the following files would be overwritten by merge:
  file1.txt
Please commit your changes or stash them before you merge.
Aborting

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$
```

```
MINGW64/c/Users/Administrator/stash
2047419 M1
Administrator@0b2660fad0e35d8 MINGW64 ~/rebase (master)
$ cd
Administrator@0b2660fad0e35d8 MINGW64 ~
$ mkdir stash
Administrator@0b2660fad0e35d8 MINGW64 ~
$ cd
display all 52 possibilities? (y or n)AC
Administrator@0b2660fad0e35d8 MINGW64 ~
$ cd stash/
Administrator@0b2660fad0e35d8 MINGW64 ~/stash
$ git init
Initialized empty Git repository in C:/Users/Administrator/stash/.git/
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ echo "hi " >> file1.txt
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git commit -m "hi added"
[master (root-commit) 3fe82c1] hi added
1 file changed, 1 insertion(+)
create mode 100644 file1.txt
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git remote add origin git@github.com:devanarayanantm/stash.git
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 214 bytes | 214.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
to github.com:devanarayanantm/stash.git
* [new branch]      master -> master

Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ cat file1.txt
<<<<<< updated upstream
hi
guys
=====
welcome
>>>>>> Stashed changes
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git add .
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file1.txt
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ cat file1.txt
hi
guys
welcome
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
$ git commit -m "changed conglit"
[master 0ba7185] changed conglit
1 file changed, 1 insertion(+)
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (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
Administrator@0b2660fad0e35d8 MINGW64 ~/stash (master)
```

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: k mode You will still keep the changes in your working tree but not on index

Revert

```
MINGW64/c/Users/Administrator/testGit
Administrator@Ob2660fad0e35d8 MINGW64 ~/testGit (master|REVERTING)
$ git status
On branch master
You are currently reverting commit 3c27bb9.
(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/rm <file>..." as appropriate to mark resolution)
    deleted by them: file.txt

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

Administrator@Ob2660fad0e35d8 MINGW64 ~/testGit (master|REVERTING)
$ cat file.txt
hello world
hii
bye

Administrator@Ob2660fad0e35d8 MINGW64 ~/testGit (master|REVERTING)
$ git diff
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
* Unmerged path file.txt

Administrator@Ob2660fad0e35d8 MINGW64 ~/testGit (master|REVERTING)
$ git diff --staged
* Unmerged path file.txt

Administrator@Ob2660fad0e35d8 MINGW64 ~/testGit (master|REVERTING)
Revert "T3"
```

```
MINGW64/c/Users/Administrator/testGit
Administrator@Ob2660fad0e35d8 MINGW64 ~/testGit (master)
$ git add .
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

Administrator@Ob2660fad0e35d8 MINGW64 ~/testGit (master)
$ git commit -m "T3"
[master 09b23c9] T3
1 file changed, 1 insertion(+)

Administrator@Ob2660fad0e35d8 MINGW64 ~/testGit (master)
$ cat file.txt
hello world
hii
bye

Administrator@Ob2660fad0e35d8 MINGW64 ~/testGit (master)
$ git log --oneline
09b23c9 (HEAD -> master) T3
1ef6d15 T2
3c27bb9 T1

Administrator@Ob2660fad0e35d8 MINGW64 ~/testGit (master)
$ git revert HEAD~2
CONFLICT (modify/delete): file.txt deleted in (empty tree) and modified in HEAD. Version HEAD of file.txt left in tree.
error: could not revert 3c27bb9... T1
hint: After resolving the conflicts, mark them with
hint: "git add/rm <paths>...", 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@Ob2660fad0e35d8 MINGW64 ~/testGit (master|REVERTING)
$ git status
On branch master
You are currently reverting commit 3c27bb9.
(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/rm <file>..." as appropriate to mark resolution)
    deleted by them: file.txt
```

```
MINGW64/c/Users/Administrator/testGit
Administrator@0b2660fad0e35d8 MINGW64 ~
$ mkdir testGit
Administrator@0b2660fad0e35d8 MINGW64 ~
$ cd testGit/
Administrator@0b2660fad0e35d8 MINGW64 ~/testGit
$ ls
Administrator@0b2660fad0e35d8 MINGW64 ~/testGit
$ ls -a
./ ../
Administrator@0b2660fad0e35d8 MINGW64 ~/testGit
$ git init
Initialized empty Git repository in C:/Users/Administrator/testGit/.git/
Administrator@0b2660fad0e35d8 MINGW64 ~/testGit (master)
$ echo hello world >> file.txt
Administrator@0b2660fad0e35d8 MINGW64 ~/testGit (master)
$ git add .
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@0b2660fad0e35d8 MINGW64 ~/testGit (master)
$ git commit -m "T1"
[master (root-commit) 3c27bb9] T1
1 file changed, 1 insertion(+)
create mode 100644 file.txt
Administrator@0b2660fad0e35d8 MINGW64 ~/testGit (master)
$ echo hi >> file.txt
Administrator@0b2660fad0e35d8 MINGW64 ~/testGit (master)
$ git add .
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@0b2660fad0e35d8 MINGW64 ~/testGit (master)
$ git commit -m "T2"
[master 1ef6d15] T2
1 file changed, 1 insertion(+)
Administrator@0b2660fad0e35d8 MINGW64 ~/testGit (master)
$ echo bye >> file.txt
```

LFS

```
MINGW64/c/Users/Administrator/devops/git/lfs
Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ cat .gitattributes
*.jpeg filter=lfs diff=lfs merge=lfs -text

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ git lfs track "*.jpeg"
Tracking "*.jpeg"

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ ls -a
./ ../ .git/ .gitattributes

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ cat .gitattributes
*.jpeg filter=lfs diff=lfs merge=lfs -text
*.jpeg filter=lfs diff=lfs merge=lfs -text

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ echo "jpeg" >> file1.jpeg

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ cat file1.jpeg
jpeg

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ git add .

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ git commit -m "T1"
[master (root-commit) ad5f5b3] T1
2 files changed, 5 insertions(+)
create mode 100644 .gitattributes
create mode 100644 file1.jpeg

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ git remote add origin git@github.com:devanarayanantm/lfs.git

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ git push -u origin master
Uploading LFS objects: 100% (1/1), 6 B | 0 B/s, done.
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 409 bytes | 409.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:devanarayanantm/lfs.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ |

MINGW64/c/Users/Administrator/devops/git/lfs
Administrator@QB2660Fad0e35d8 MINGW64 ~/Desktop
$ cd --

Administrator@QB2660Fad0e35d8 MINGW64 ~
$ cd devops/git/

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git
$ git lfs install
Git LFS initialized.

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git
$ mkdir lfs

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git
$ cd lfs/

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs
$ git init
Initialized empty Git repository in C:/Users/Administrator/devops/git/lfs/.git/

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ git lfs track "*.jpeg"
Tracking "*.jpeg"

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ ls -a
./ ../ .git/ .gitattributes

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ cat .gitattributes
*.jpeg filter=lfs diff=lfs merge=lfs -text

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ git lfs track "*.jpeg"
Tracking "*.jpeg"

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ ls -a
./ ../ .git/ .gitattributes

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ cat .gitattributes
*.jpeg filter=lfs diff=lfs merge=lfs -text
*.jpeg filter=lfs diff=lfs merge=lfs -text

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ echo "jpeg" >> file1.jpeg

Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ cat file1.jpeg
jpeg
```

SIGNED COMMIT

```
MINGW64/c/Users/Administrator/devops/git/ifs
$ git log --show-signature
commit 32bbe02241d5214aee835ceaedf0bd5fd98f3f (HEAD -> master)
Author: devanarayanantm <tmdevanarayan@gmail.com>
Date: Fri Jan 17 10:35:15 2025 +0530

    add new more file

Signed-off-by: devanarayanantm <tmdevanarayan@gmail.com>

commit 01c2e5b1f1260bc7eea32080006f86ab8d7faa2 (origin/master)
Author: devanarayanantm <tmdevanarayan@gmail.com>
Date: Fri Jan 17 10:33:58 2025 +0530

    add new file

commit a8eb3cbada6c7d3b01ceb58493b03c68c334fa13
Author: narendramodi <narendramodi@gmail.com>
Date: Fri Jan 17 10:18:06 2025 +0530

    Learning signed commit

commit a45f5b3983a8f4143a2fd2966c266a80e28d05bd
Author: devanarayanantm <tmdevanarayan@gmail.com>
Date: Fri Jan 17 10:01:10 2025 +0530

    T1

Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git config --global F1FE4C358D4AD487
error: key does not contain a section: F1FE4C358D4AD487

Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git config --global user.signkey F1FE4C358D4AD487

Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git config --global commit.gpgsign true

Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git commit -S -m "encryption add new more file"
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

Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ echo "New test " >> testfile.txt

Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git add .
warning: in the working copy of 'testfile.txt', LF will be replaced by CRLF the next time Git touches it
```

```
Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ echo "new more" >> file4.txt

Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git add .
warning: in the working copy of 'file4.txt', LF will be replaced by CRLF the next time Git touches it

Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git commit -S -m "add new more file"
(master 32bbe0f) add new more file
1 file changed, 1 insertion(+)
create mode 100644 file4.txt

Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git commit -S -m "add new more file"
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

Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git log
commit 32bbe02241d5214aee835ceaedf0bd5fd98f3f (HEAD -> master)
Author: devanarayanantm <tmdevanarayan@gmail.com>
Date: Fri Jan 17 10:35:15 2025 +0530

    add new more file

Signed-off-by: devanarayanantm <tmdevanarayan@gmail.com>

commit 01c2e5b1f1260bc7eea32080006f86ab8d7faa2 (origin/master)
Author: devanarayanantm <tmdevanarayan@gmail.com>
Date: Fri Jan 17 10:33:58 2025 +0530

    add new file

commit a8eb3cbada6c7d3b01ceb58493b03c68c334fa13
Author: narendramodi <narendramodi@gmail.com>
Date: Fri Jan 17 10:18:06 2025 +0530

    Learning signed commit

commit a45f5b3983a8f4143a2fd2966c266a80e28d05bd
Author: devanarayanantm <tmdevanarayan@gmail.com>
Date: Fri Jan 17 10:01:10 2025 +0530

    T1

Administrator@QB2660fa0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git log --show-signature
```

```
MINGW64/c/Users/Administrator/devops/git/lfs
pub  rsa4096 2025-01-17 [SC]
    34700DBA9535A428E156BC53F1FE4C358D4AD487
uid          devanarayanantm (pgp key generation) <tmdevanarayan@gmail.com>
sub  rsa4096 2025-01-17 [E]

Administrator@MINGW64:~/devops/git/lfs (master)
$ gpg --list
gpg: option "--list" is ambiguous

Administrator@MINGW64:~/devops/git/lfs (master)
$ gpg --list-secret-keys --keyid-format long
gpg: invalid option "--list-secret-keys"

Administrator@MINGW64:~/devops/git/lfs (master)
$ gpg --list-secret-keys --keyid-format long
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid: 1  signed: 0  trust: 0-, 0q, 0n, 0m, 0f, 1u
[keyboard]
-----
sec  rsa4096/34700DBA9535A428E156BC53F1FE4C358D4AD487 2025-01-17 [SC]
uid          [ultimate] devanarayanantm (pgp key generation) <tmdevanarayan@gmail.com>
ssb  rsa4096/4A8B866C2F2AF7B8 2025-01-17 [E]

Administrator@MINGW64:~/devops/git/lfs (master)
$ echo "new" >> file3.txt
Administrator@MINGW64:~/devops/git/lfs (master)
$ git add .
warning: in the working copy of 'file3.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@MINGW64:~/devops/git/lfs (master)
$ git commit -m "add new file"
(master 01c2e5b) add new file
1 file changed, 1 insertion(+)
create mode 100644 file3.txt
Administrator@MINGW64:~/devops/git/lfs (master)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 549 bytes | 274.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Remote: Resolving deltas: 100% (3/3), done.
To github.com:devanarayanantm/lfs.git
  a45f5b3..01c2e5b master -> master

MINGW64/c/Users/Administrator/devops/git/lfs
Administrator@MINGW64:~/devops/git/lfs (master)
$ gpg --full-generate-key
gpg (GnuPG) 2.4.4-unknown; Copyright (C) 2024 g10 Code GmbH
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/c/Users/Administrator/.gnupg' created
Please select what kind of key you want:
  (1) RSA and RSA
  (2) DSA and ElGamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (5) ECC (sign and encrypt) "default"
  (10) ECC (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <nw> = key expires in n weeks
  <nm> = key expires in n months
  <ny> = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: devanarayanantm
Email address: tmdevanarayan@gmail.com
Comment: gpg key generation
You selected this USER-ID:
  "devanarayanantm (pgp key generation) <tmdevanarayan@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /c/Users/Administrator/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/c/Users/Administrator/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/c/Users/Administrator/.gnupg/openpgp-revocs.d/34700DBA9535A428E156BC53F1FE4C358D4AD487.rev'
public and secret key created and signed.

pub  rsa4096 2025-01-17 [SC]
```

```
MINGW64/c/Users/Administrator/devops/git/ifs
Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/ifs (master)
$ echo "hi" >> file2.txt
Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git add .
warning: in the working copy of 'file2.txt', LF will be replaced by CRLF the next time Git touches it
Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git commit -m "Learning signed commit" --author="narendra modi <naredramodi@gmail.com>"
fatal: malformed --author parameter
Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git commit -m "Learning signed commit" --author="narendra modi <naredramodi@gmail.com>"
fatal: --author 'narendra modi <naredramodi@gmail.com>' is not 'Name <email>' and matches no existing author
Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git commit -m "Learning signed commit" --author="narendra modi<naredramodi@gmail.com>"
fatal: --author 'narendra modi<naredramodi@gmail.com>' is not 'Name <email>' and matches no existing author
Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git commit -m "Learning signed commit" --author="narendra modi" <naredramodi@gmail.com>
bash: naredramodi@gmail.com: No such file or directory
Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git commit -m "Learning signed commit" --author="narendra modi" <naredramodi@gmail.com>
bash: naredramodi@gmail.com: No such file or directory
Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git commit -m "Learning signed commit" --author="narendra modi<naredramodi@gmail.com>"
[master a8eb3cb] Learning signed commit
Author: narendra modi <naredramodi@gmail.com>
1 file changed, 1 insertion(+)
create mode 100644 file2.txt
Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git log --oneline
a8eb3cb (HEAD -> master) Learning signed commit
a45f5b3 (origin/master) T1
Administrator@QB2660Fad0e35d8 MINGW64 ~/devops/git/ifs (master)
$ git log
commit a8eb3cbada6c7d7b01ceb58493b03c68c334fa13 (HEAD -> master)
Author: narendra modi <naredramodi@gmail.com>
Date: Fri Jan 17 10:18:06 2025 +0530

    Learning signed commit
commit a45f5b393abf4143a2fd2866c266a80e28d05bd (origin/master)
Author: devanarayanantm <tmddevanarayan@gmail.com>
Date: Fri Jan 17 10:01:10 2025 +0530
```

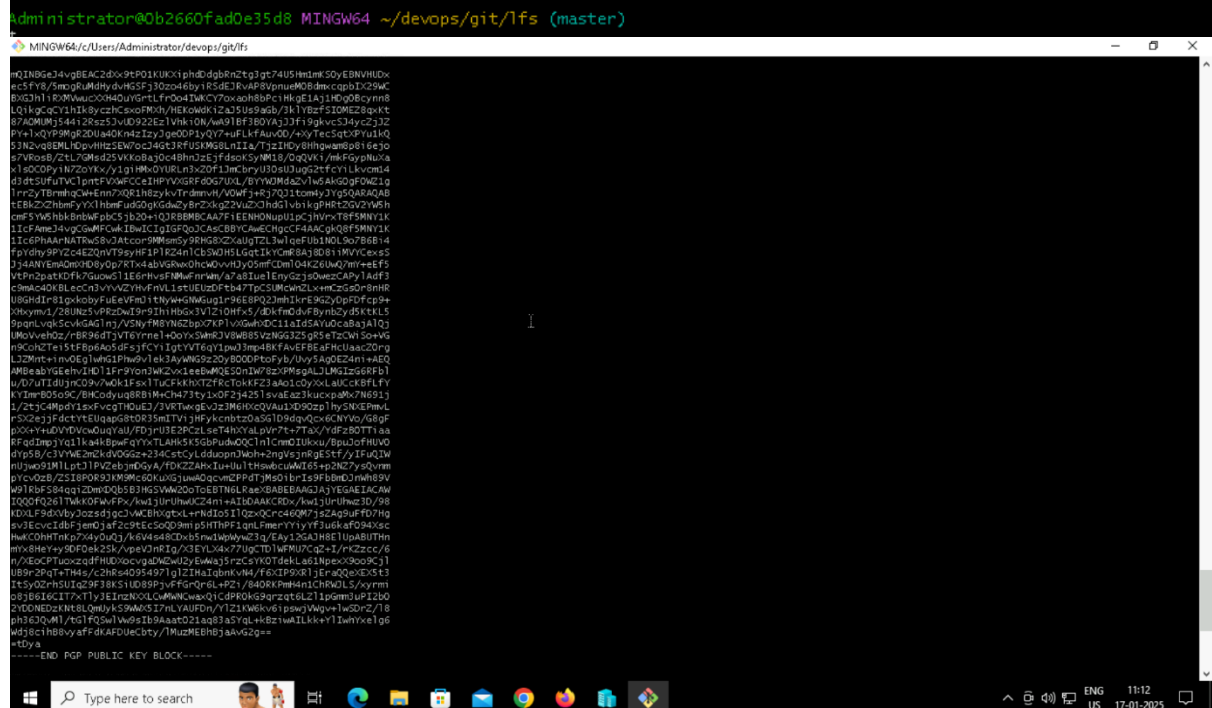
```
Administrator@0b2660fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ gpg --armor --export tmdevanarayanan@gmail.com >> a.txt

Administrator@0b2660fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ AC

Administrator@0b2660fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ git add .
warning: in the working copy of 'a.txt', LF will be replaced by CRLF the next time Git touches it

Administrator@0b2660fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ git commit -S -m "adding a.txt"
[master ce15f8e] adding a.txt
1 file changed, 52 insertions(+)
create mode 100644 a.txt

Administrator@0b2660fad0e35d8 MINGW64 ~/devops/git/lfs (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 3.26 KiB | 3.26 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:devanarayanantm/lfs.git
eb9ed54..ce15f8e master -> master
```




```
MINGW64/C:/Users/Administrator/devops/git/ifs
commit 32bbe02d4d5214ee835ceae6f0bd5fd98f3f
Author: devanarayanantm <tmdevanarayan@gmail.com>
Date: Fri Jan 17 10:35:15 2025 +0530

    add new more file

Signed-off-by: devanarayanantm <tmdevanarayan@gmail.com>
commit 01c2e5b1f1260bc7eea32080006f86ab8d7faa2 (origin/master)
Author: devanarayanantm <tmdevanarayan@gmail.com>
Date: Fri Jan 17 10:33:58 2025 +0530

    add new file

commit a8eb3cbada6c7d3b01ceb58493b03c68c334fa13
Author: narendramodi <narendramodi@gmail.com>
Date: Fri Jan 17 10:18:06 2025 +0530

    Learning signed commit

commit s45f5b3983abf4143a2fd2966c266a80e28d05bd
Author: devanarayanantm <tmdevanarayan@gmail.com>
Date: Fri Jan 17 10:01:10 2025 +0530

    T1

Administrator@WS2660f8de35d8 MINGW64 ~/devops/git/ifs (master)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 1.17 KiB | 1.17 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
to github.com:devanarayanantm/ifs.git
01c2e5b..eb9ed54 master -> master

Administrator@WS2660f8de35d8 MINGW64 ~/devops/git/ifs (master)
$ gpg --armor --export 'tmdevanarayan@gmail.com'
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQINBGe34vgBEAC2d0x9P01KUKX1phdDgbrn2tg3gt74U5Hedmk50yEBWVH0x
c3Fy8/mcghJHdydHMS5F3Gz0r6y8F5dE3VAP8ymuM08dmccpB3Z2wC
BXG3h1iZMwWuc02H40UvG=LfF0d41WKY7oxa0h8BPC1H8E1a3110q08cyrm8
LQ1kGcQy1h1K8yczhC5x0F0dh/HEK0dWk12A35Us9asb/3k1YBzfSIOME28qxt
87ADUMJ544126sz53VUD922Ez1Vnk1ON/wa91BF380VAg3Jf19kvc534yc23J2
Fv1xQY8m9P2Dua40Knd2zy2ed0P1yQV7=afLkFAuX00/xy7ec5epPru1KQ
53N2vq8EMLHDpVHtzSEW7oc34gt3RFU5KMS8Ln1Ia/Tj2IHcy8Hhpwan861e3jo
s7VROS8/ZtL7QMsd25VKko8ag0c48hn3zEjfdsoKSyNM18/QgZvki/mkFGypNuXa
x1S0COPy1N72oYKx/y1giHMK0YURLn3X20F1McLbryU30sUug2TfcY1Lkvcml4
-----

Administrator@WS2660f8de35d8 MINGW64 ~/devops/git/ifs (master)
$ git commit -s -m "for testing encryption add newtestfile"
error: gpg failed to sign the data
gpg: skipped "devanarayanantm <tmdevanarayan@gmail.com>": No secret key
[GNUPG:] INV_SGNR 9 devanarayanantm <tmdevanarayan@gmail.com>
[GNUPG:] FAILURE sign 17
gpg: signing failed: No secret key
fatal: failed to write commit object

Administrator@WS2660f8de35d8 MINGW64 ~/devops/git/ifs (master)
$ gpg --list-secret-keys --devanarayanantm--format LONG
gpg: invalid option "--devanarayanantm--format"

Administrator@WS2660f8de35d8 MINGW64 ~/devops/git/ifs (master)
$ gpg --list-secret-keys --devanarayanantm LONG
gpg: invalid option "--devanarayanantm"

Administrator@WS2660f8de35d8 MINGW64 ~/devops/git/ifs (master)
$ gpg --list-secret-keys --devanarayanantm
gpg: invalid option "--devanarayanantm"

Administrator@WS2660f8de35d8 MINGW64 ~/devops/git/ifs (master)
$ git config --global user.signingkey F1FE4C358D4AD487

Administrator@WS2660f8de35d8 MINGW64 ~/devops/git/ifs (master)
$ git commit -s -m "for testing encryption add newtestfile"
[master eb9ed54] for testing encryption add newtestfile
1 file changed, 1 insertion(+)
create mode 100644 testfile.txt

Administrator@WS2660f8de35d8 MINGW64 ~/devops/git/ifs (master)
$
Administrator@WS2660f8de35d8 MINGW64 ~/devops/git/ifs (master)
$ git log --show-signature
commit eb9ed5463088635386a38de231211ab54500a8c (HEAD -> master)
gpg: Signature made Fri Jan 17 10:52:42 2025 IST
gpg: using s2s key 3470CDBA9335A428E156BC5F1FE4C358D4AD487
gpg: Good signature from "devanarayanantm (gpg key generation) <tmdevanarayan@gmail.com>" [ultimate]
Author: devanarayanantm <tmdevanarayan@gmail.com>
Date: Fri Jan 17 10:52:42 2025 +0530

    for testing encryption add newtestfile

commit 32bbe02d4d5214ee835ceae6f0bd5fd98f3f
Author: devanarayanantm <tmdevanarayan@gmail.com>
Date: Fri Jan 17 10:35:15 2025 +0530
```