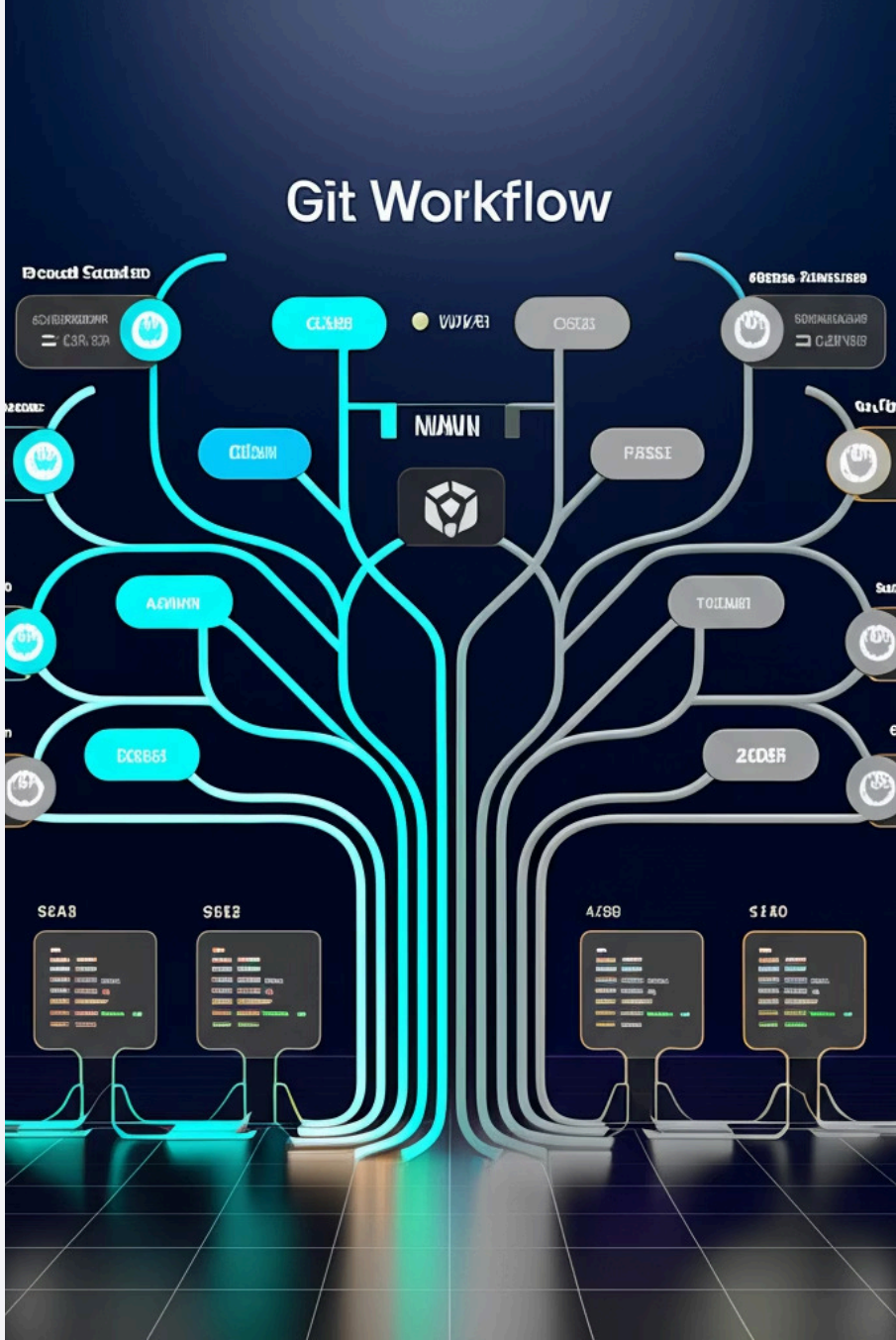
GIT: Version Control Made Simple

Ravi Kiran Maroju

RK by Ravi Kiran



What is Version Control?

Ravi Kiran Maroju

Time Machine for Files

Version control is like a time machine for your files. It remembers everything you change in your work.

Mistake Recovery

If you make a mistake, you can go back to how it was before.

Change Tracking

It helps you see what was changed, when, and by who.

Collaboration Tool

It's super helpful when working with other people too!

What is Git?

Ravi Kiran Maroju

Change Tracking Tool

Git is a special tool that helps you save changes to your work. You can write code, save it, and Git will remember what you did.

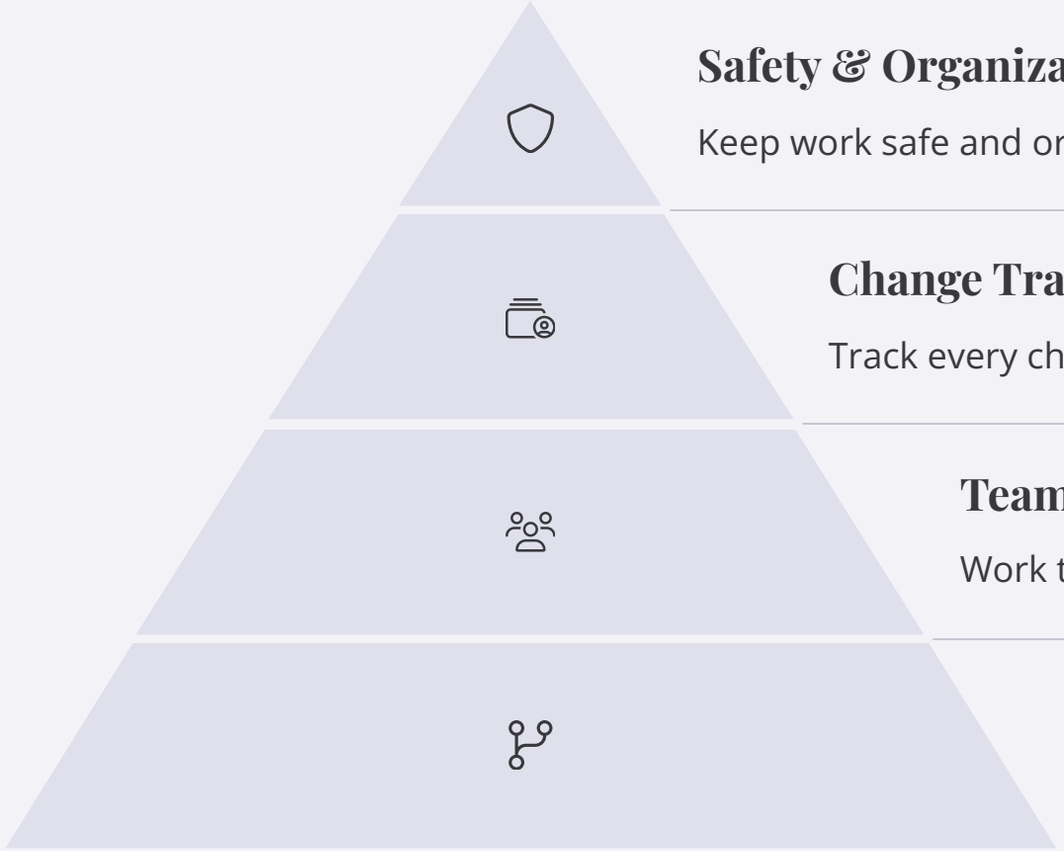
Time Travel

If something breaks later, Git can take you back to when it was working.

Collaboration System

Many people use Git to work together on big projects without confusion. Git keeps everything neat and easy to fix.

Why Do People Use Git?



Safety & Organization

Keep work safe and organized

Change Tracking

Track every change made

Team Collaboration

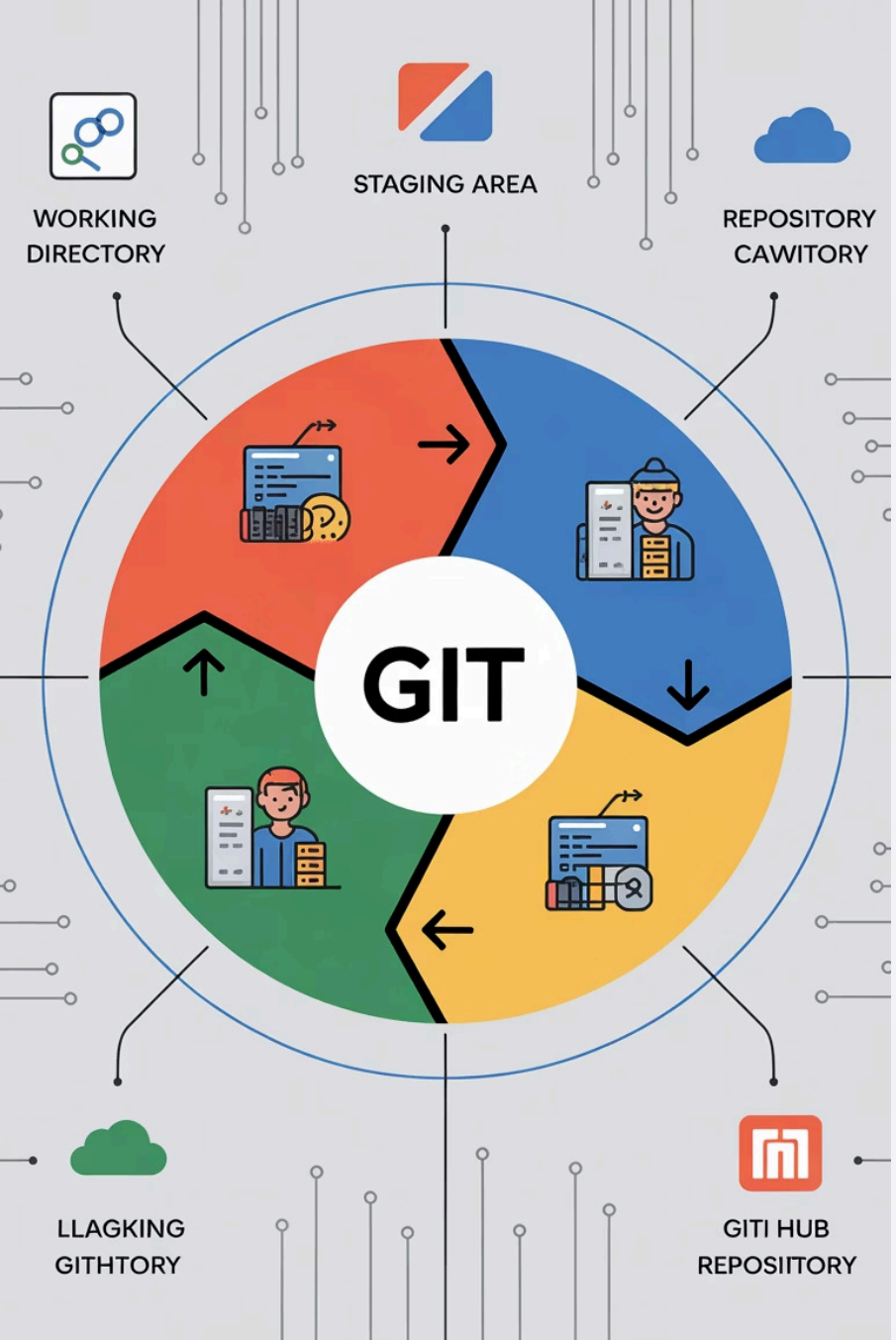
Work together seamlessly

Experimentation

Try new ideas safely

People use Git to keep their work safe and organized. It helps them track every change they make. If someone makes a mistake, Git can help fix it quickly. It also lets teams of people work on the same files at the same time. Git makes it easy to try new ideas without messing up the main project.

Ravi Kiran Maroju



Git Project's Three Main Parts

Working Directory

This is where you do your work. You can write, delete, or change files here. It's like your personal workspace on the computer.

Staging Area

This is where you get your work ready to be saved. You say, "I want to save these changes," and put them here. It's like putting toys in a box before putting them on the shelf.

Git Directory (Repository)

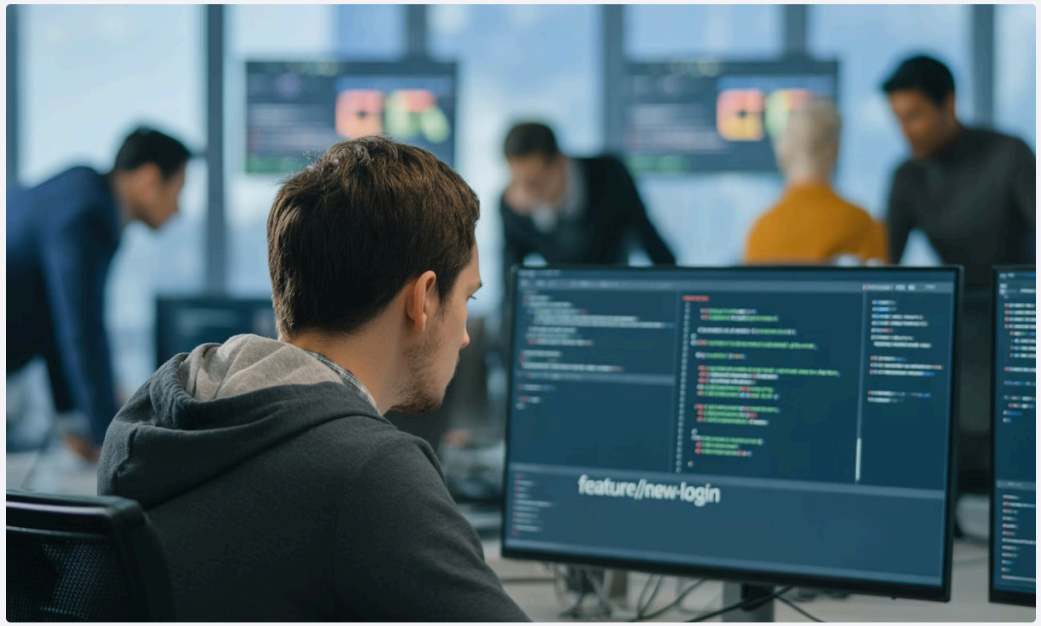
This is where Git keeps your saved changes. Every time you say "save" (commit), it takes a snapshot and stores it here. It remembers everything for you.

Ravi Kiran Maroju

Git Terminology: Branch

What is a Branch?

A branch is like a copy of your project where you can try new things. It doesn't change the main project until you're ready to add it. Think of it like a new path you take to explore ideas.



Branches allow developers to work on features or fixes independently without affecting the main codebase until they're ready to merge their changes.

Ravi Kiran Maroju

Git Terminology: Commit

Ravi Kiran Maroju

Code Snapshot

A commit is when you save your work in Git. It's like taking a picture of your code at that moment.

Historical Record

You can come back to it anytime.

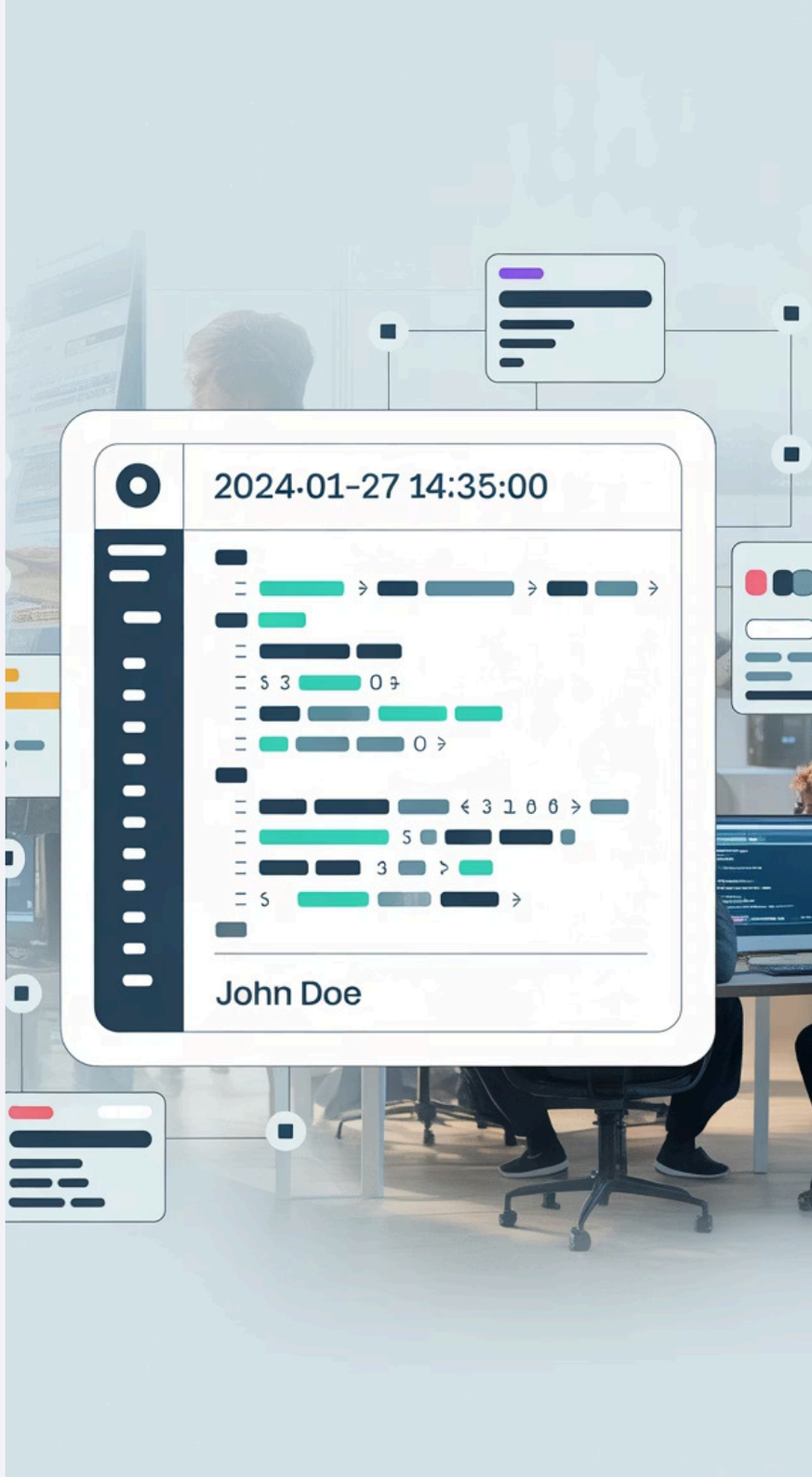
Unique Identifier

Each commit has a unique hash ID that identifies it.

Message Included

Commits include messages explaining what changed.

Ravi Kiran Maroju

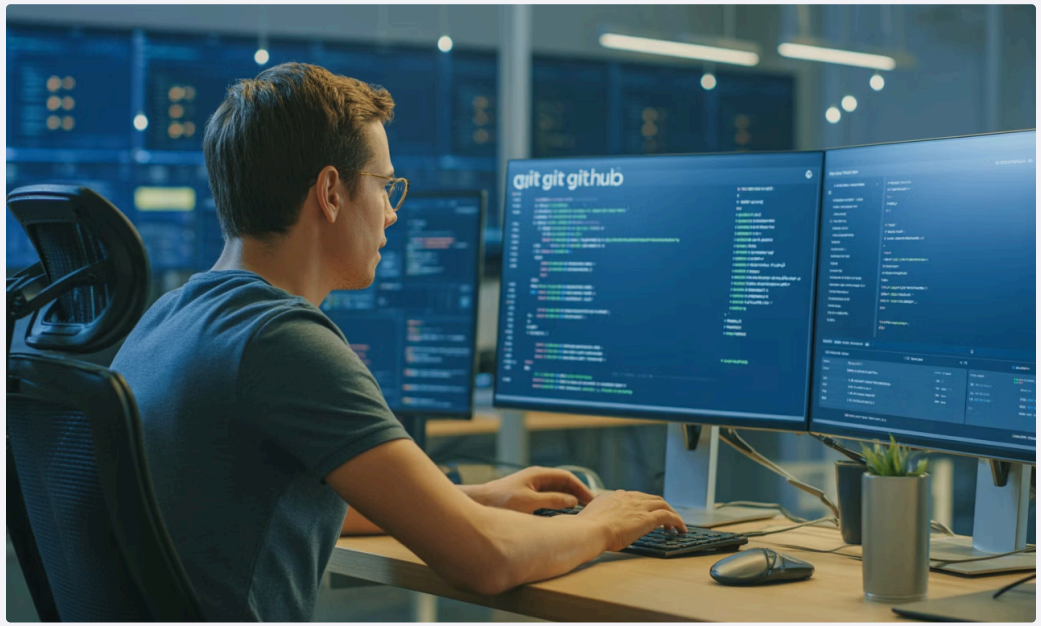


Git Terminology: Diff

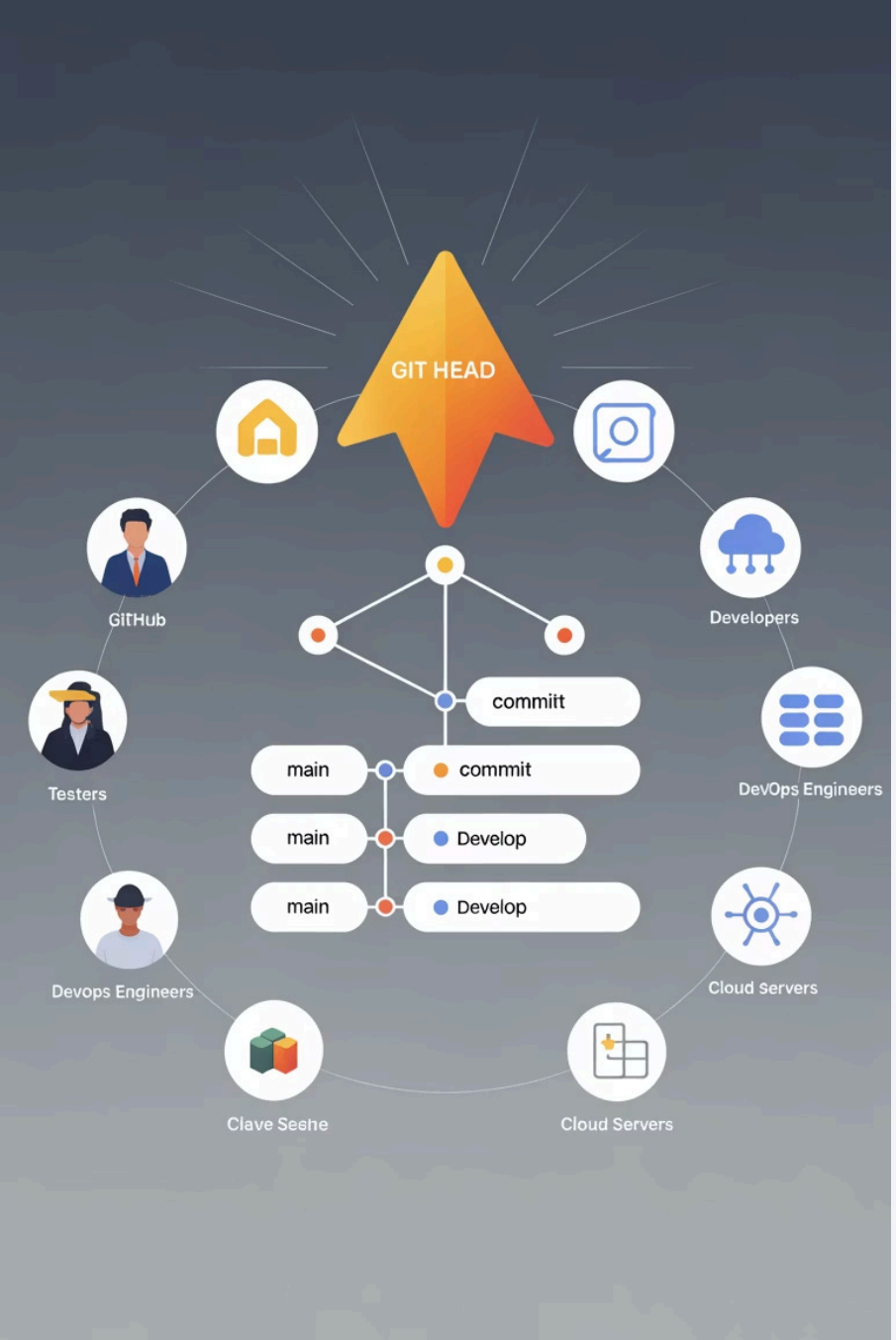
Ravi Kiran Maroju

What is a Diff?

Diff shows what has changed between two versions. It highlights the lines you added or removed. It's like comparing two drawings to spot the differences.



Diffs are essential for code reviews and understanding what changes were made between versions of your project.



Git Terminology: HEAD

By **Ravi Kiran Maroju**

Current Position

HEAD means the most recent version you are working on. It tells Git where you are right now in your project.

Moving Pointer

When you switch branches or make commits, HEAD moves to point to your current location.

Reference Point

Commands often use HEAD as a reference point, like "git reset HEAD~1" to go back one commit.

Git Terminology: Index (Staging Area)

Make Changes

Edit files in working directory

Stage Changes

Add files to the index

Review Changes

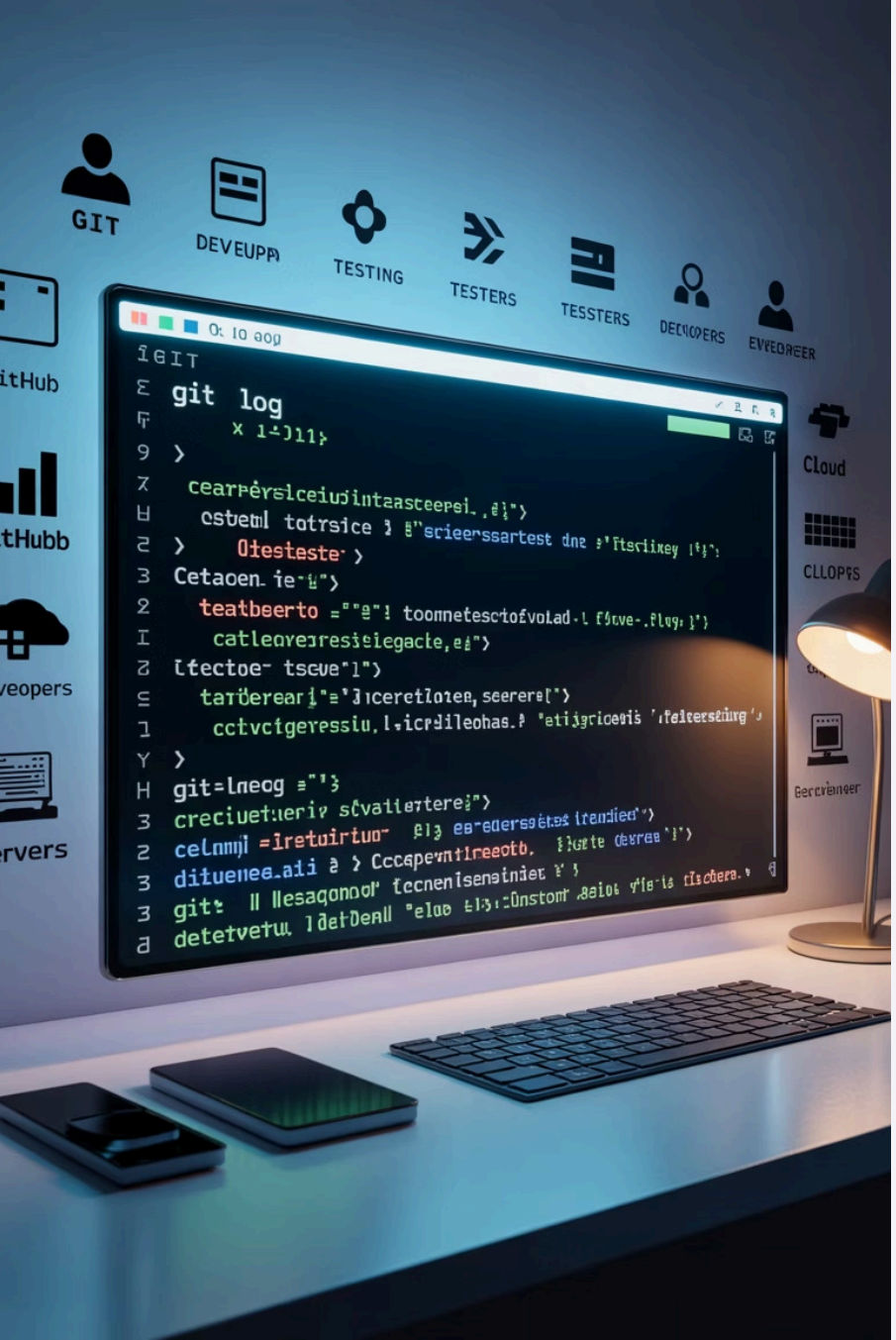
Verify what will be committed

Commit Changes

Save staged changes to repository

The index is a place where changes wait before being saved. It's like lining up your homework pages before putting them in a folder.

By **Ravi Kiran Maroju**



Git Terminology: Log

Commit History

The log is a list of all the commits you made.

Change Tracking

It helps you see what changes were made and when.

Author Information

Shows who made each commit in the project history.

Timestamps

Records exactly when each change was committed.

By **Ravi Kiran Maroju**

Git Terminology: Merge

Create Branch

Start a new line of development

Make Changes

Develop new features or fixes

Resolve Conflicts

Fix any competing changes

Merge Changes

Combine work back to main branch

Merging is when you take work from one branch and add it to another. It combines two sets of changes into one. Like mixing colors in art to make a new one.

- **Ravi Kiran Maroju**

Git Terminology: Remote

By **Ravi Kiran Maroju**

What is a Remote?

A remote is a Git project stored on another computer (usually online). You can share and get updates from this shared place. Think of it like a cloud folder.

Remotes like GitHub, GitLab, or Bitbucket allow teams to collaborate on projects from anywhere in the world.

- **Ravi Kiran Maroju**

Git Terminology: Rebase

By **Ravi Kiran Maroju**

1

Branch from main

Create feature branch from main

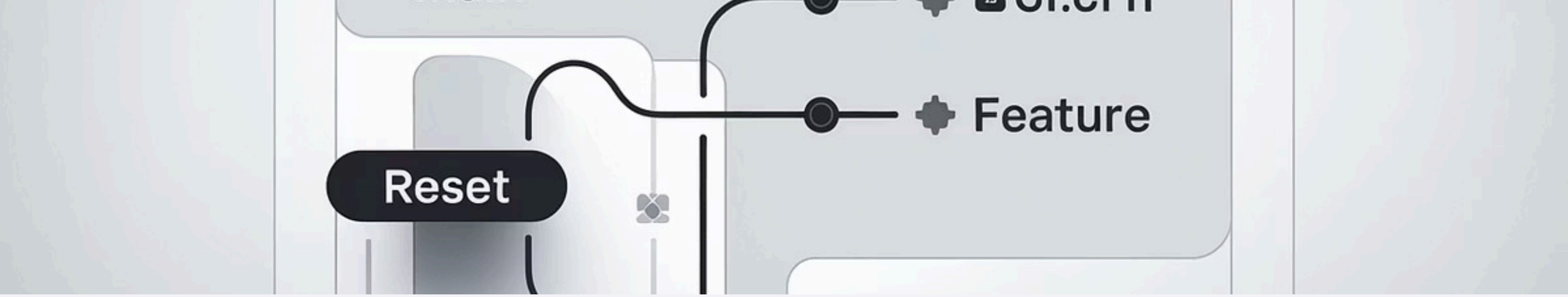
Work on feature

Make commits on your branch

Rebase on main

Move your changes on top of updated main

Rebasing moves your changes on top of the newest version. It helps keep your work neat and clean. Like moving your homework to the latest template.



Git Terminology: Reset

By **Ravi Kiran Maroju**

Time Travel

Reset means going back to an earlier version. It erases the changes you made after that point.

Different Modes

Soft reset keeps your changes staged, mixed reset unstages them, and hard reset discards them completely.

Careful Usage

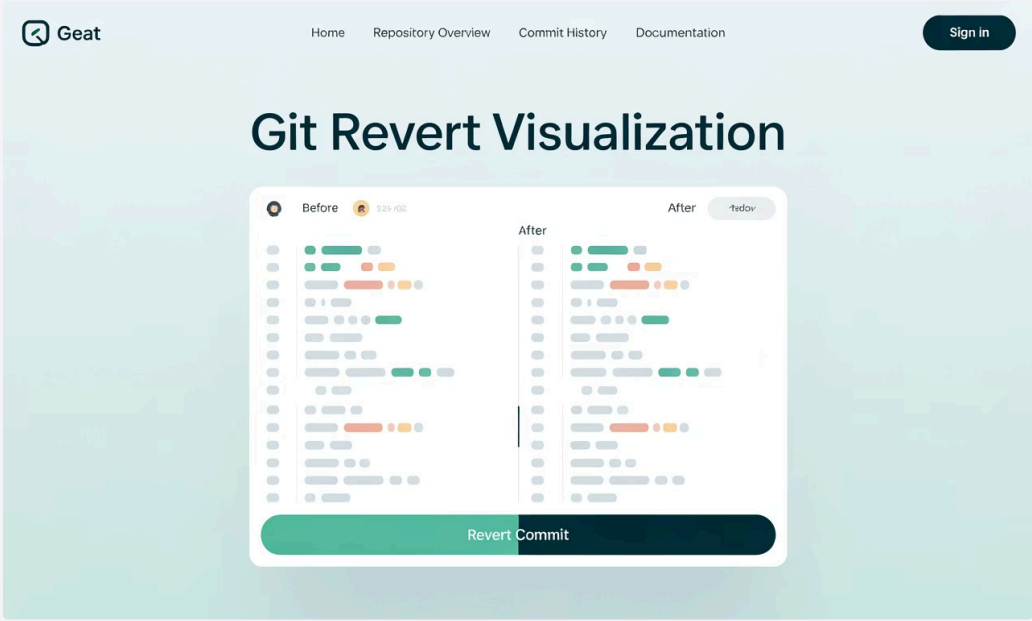
Hard reset can permanently lose work, so it should be used with caution.

Git Terminology: Revert

By **Ravi Kiran Maroju**

What is Revert?

Revert means undoing one specific commit (save). It's like saying, "Oops! Let's take that step back."



Unlike reset, revert creates a new commit that undoes changes, preserving history and making it safe for shared branches.

Git Terminology: Staging Area

By ***Ravi Kiran Maroju***

Working Directory

Files you're actively editing

git add

Move changes to staging area

Staging Area

Changes ready to be committed

git commit

Save staged changes to repository

This is where you prepare files before saving. It's like getting ready to take a photo of your work.

Git Terminology: Tag

By **Ravi Kiran Maroju**

Version Marker

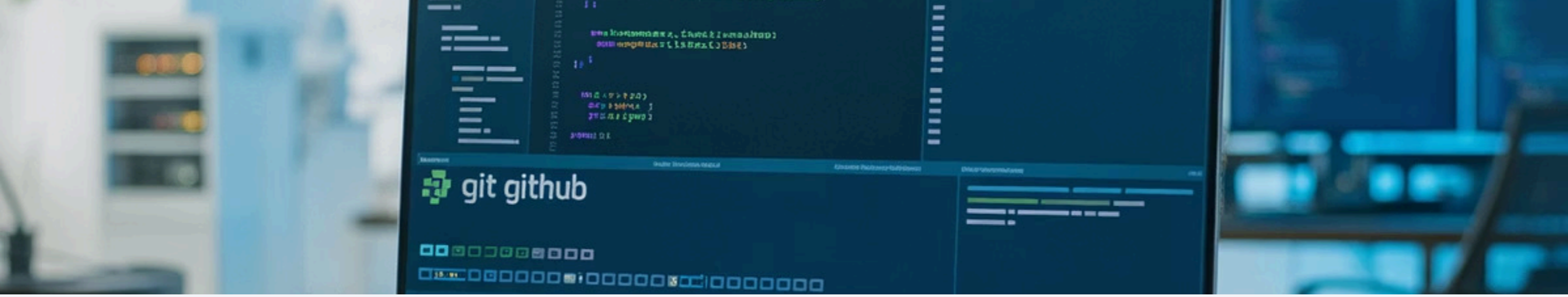
A tag is like a special name you give to a big moment. For example, "Version 1.0" or "Finished project."

Permanent Reference

Unlike branches, tags don't move when new commits are added.

Release Indicator

Tags are commonly used to mark release versions in software projects.



Git Terminology: Working Directory

Ravi Kiran Maroju - **LinkedIn**

Your Workspace

This is your real folder where you create and edit things. It's where you type, click, and work before saving with Git.

Tracked & Untracked Files

Contains both files Git knows about (tracked) and new files it doesn't yet track.

Starting Point

All Git workflows begin with changes in the working directory before being staged and committed.

Git Command: git add

By **Ravi Kiran Maroju**

Stage Changes

Adds changes to be saved.

Specific Files

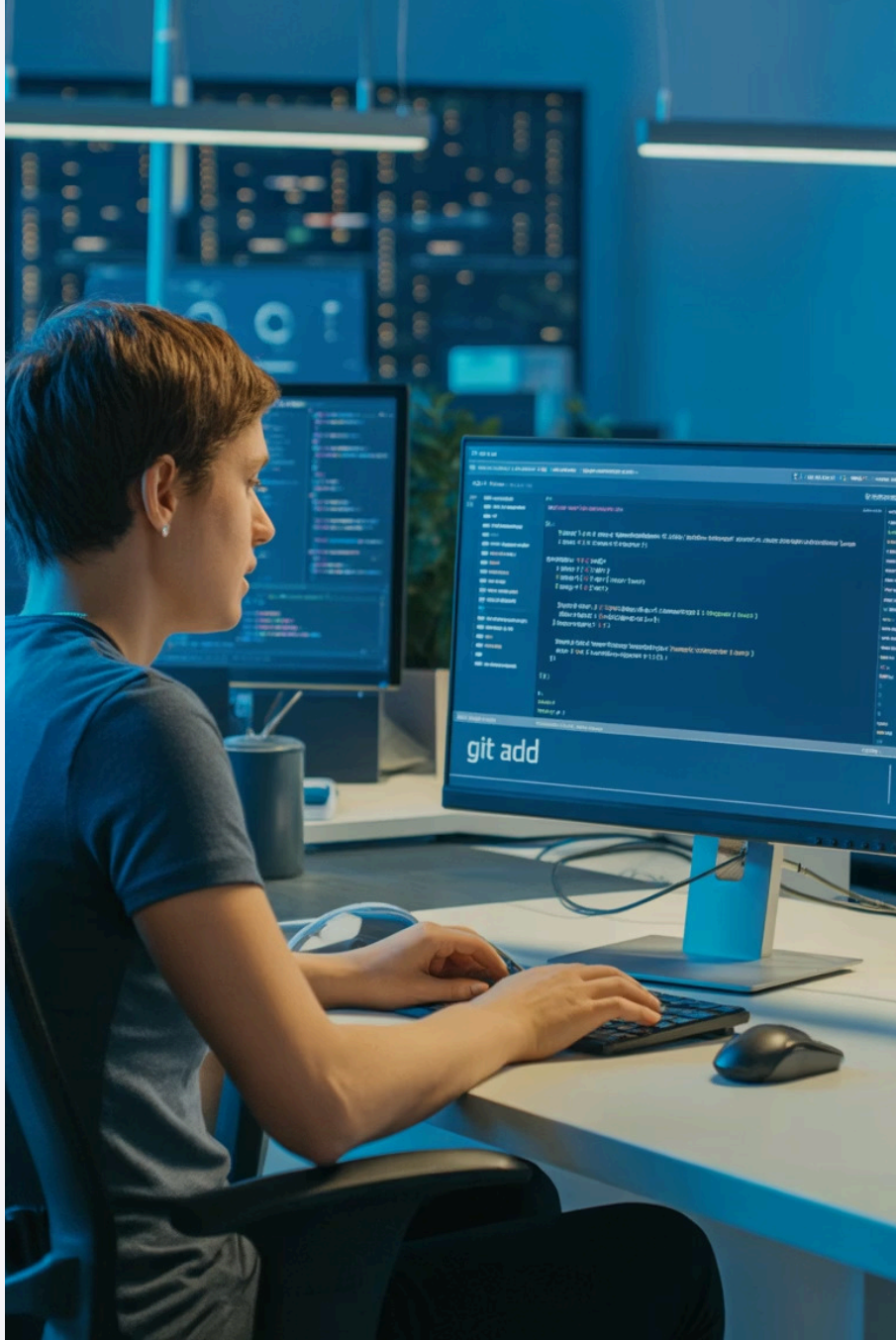
Can add individual files: git add file.txt

Directories

Can add entire directories: git add src/

All Changes

Can add all changes: git add .

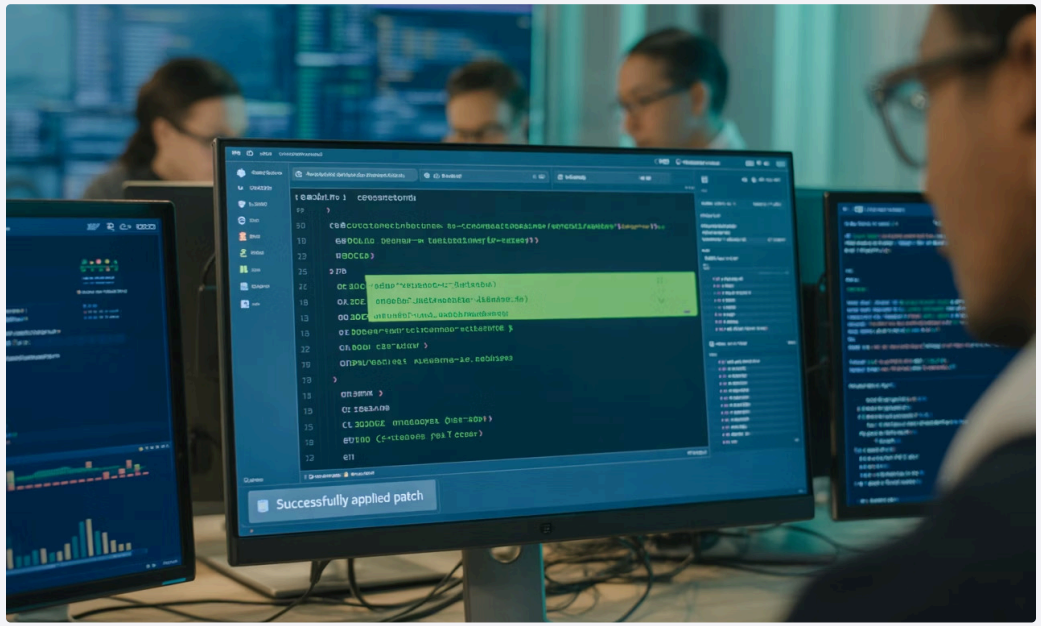


Git Command: git am

Ravi Kiran Maroju

Apply Mail Patches

Adds patches (fixes) from emails. **Example**: git am fixes.patch applies fixes from an email.



The git am command is particularly useful in open-source projects where contributors might send patches via email rather than using pull requests.

Git Command: git archive

By **Ravi Kiran Maroju**

Create Archives

Creates a zip/tar file of your project.

Export Code

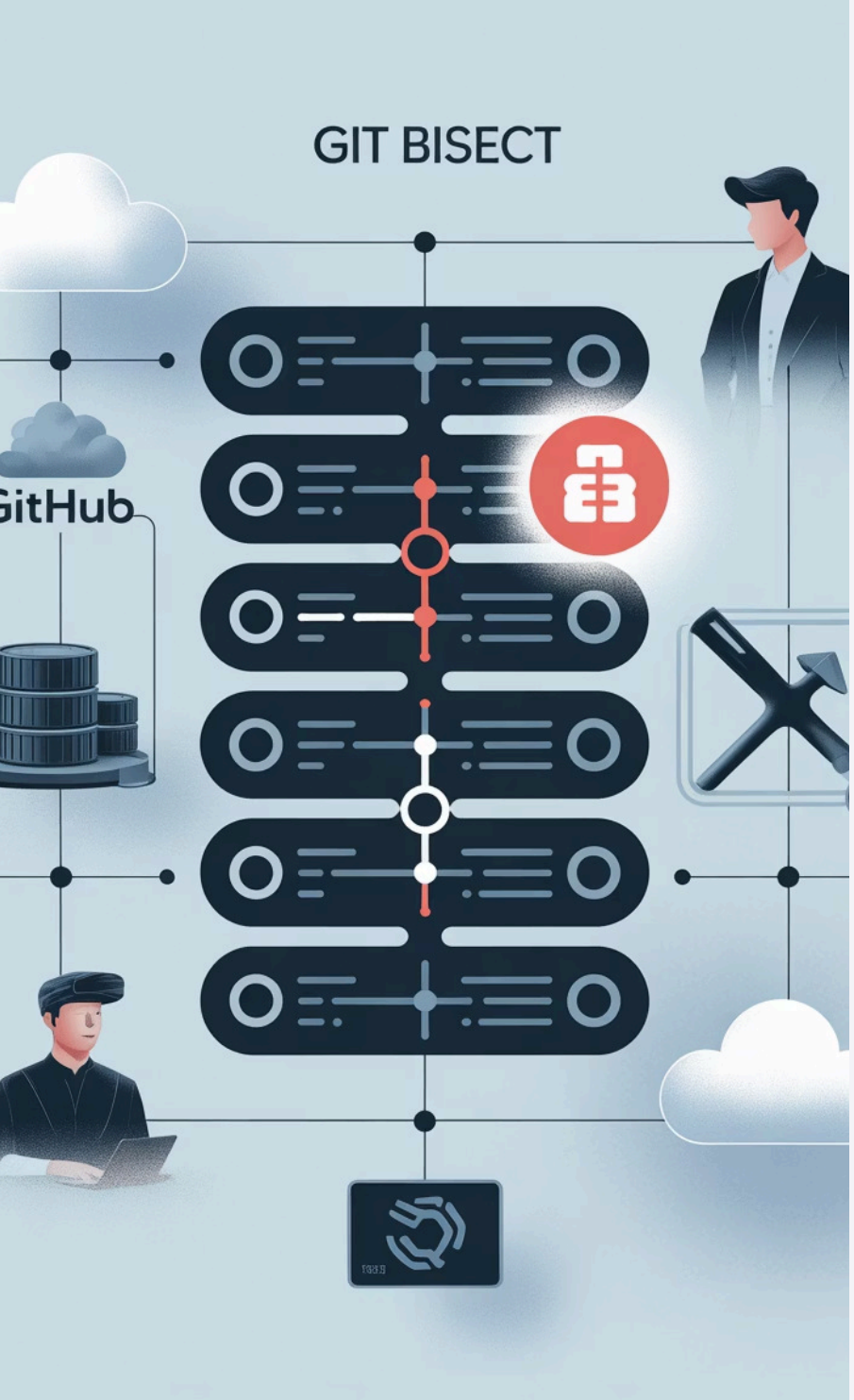
Example: git archive --output=project.zip HEAD saves the current project as a zip.

Share Code

Useful for distributing code without Git history.

Archive Releases

Often used to create release archives from specific tags.



Git Command: git bisect

Start Bisect

git bisect start

2

Mark Good Commit

git bisect good [commit]

Mark Bad Commit

git bisect bad [commit]

Find Bug Commit

Git narrows down the culprit

Helps find which commit caused a bug. **Example**: git bisect start begins the bug search process.

Ravi Kiran Maroju

Git Command: git branch

List Branches

git branch shows all local branches

Create Branch

git branch new-feature creates a new branch

Delete Branch

git branch -d branch-name deletes a branch

Show Remote Branches

git branch -r shows remote branches

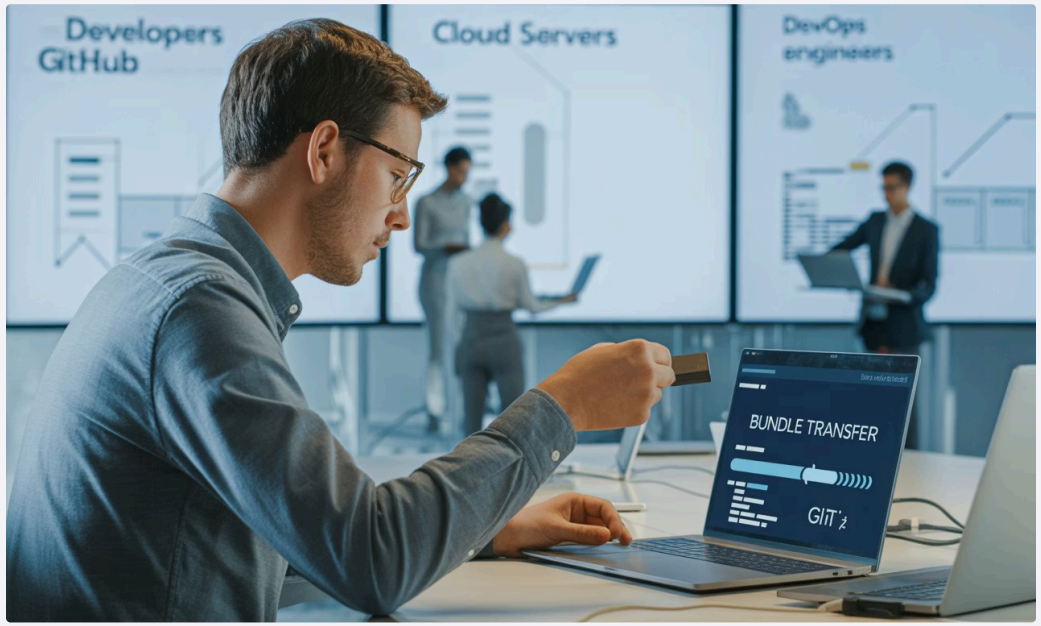
Ravi Kiran Maroju

Git Command: git bundle

By Ravi Kiran Maroju

What is git bundle?

Saves Git history in one file to share. **Example**: git bundle create repo.bundle --all makes a bundle of the repo.



Bundles are useful when you need to transfer Git repositories without using a network, such as via USB drives or when network access is limited.

Maroju Ravi Kiran

Git Command: git checkout

Switch Branches

git checkout main goes to the main branch

Restore Files

git checkout -- file.txt restores a file

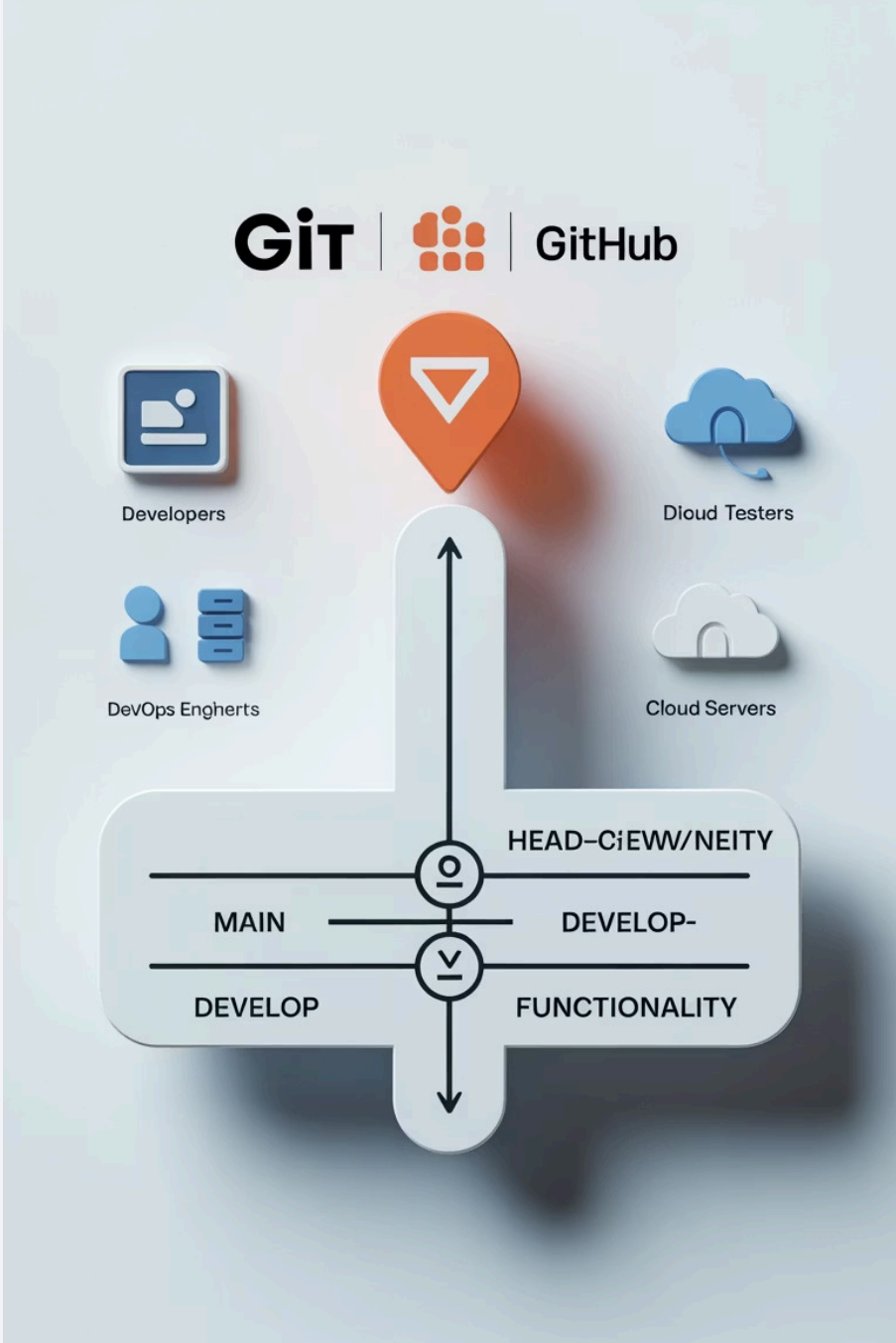
Checkout Tags

git checkout v1.0 goes to tagged version

Create & Checkout

git checkout -b new-branch creates and switches

*By **Ravi Kiran Maroju***



Git Command: git cherry-pick

1

Identify Commit

Find commit hash to apply

Select Commit

Choose specific commit to apply

3

Apply to Branch

Apply commit to current branch

4

Resolve Conflicts

Fix any issues that arise

Applies one commit from another branch. **Example**: git cherry-pick abc123 copies changes from commit abc123.

By ***Ravi Kiran Maroju***

Git Command: git citool

Graphical Interface

Opens a GUI to commit changes. **Example**: Just run git citool to use a graphical commit tool.

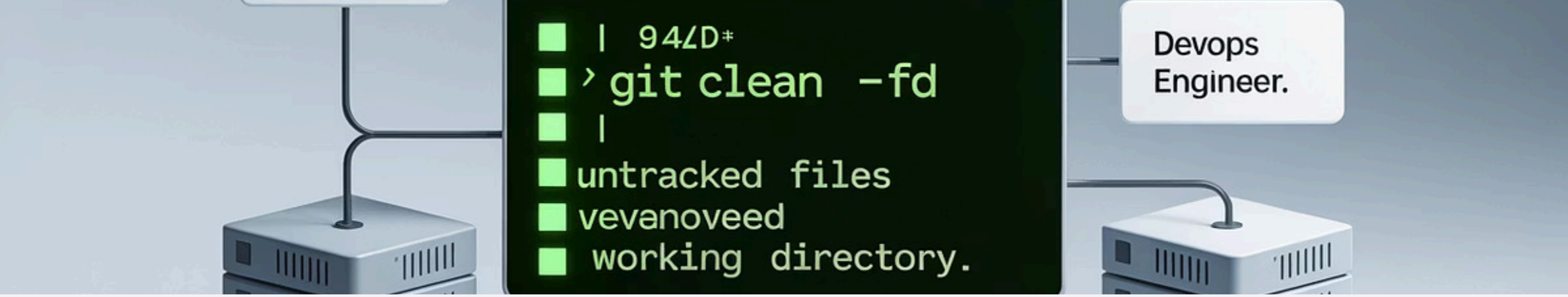
Visual Staging

Allows you to visually select which changes to include in your commit.

Commit Creation

Provides a form to enter commit messages and finalize your changes.

Contact: **Ravi Kiran Maroju**



Git Command: git clean

Remove Untracked Files

Deletes files not tracked by Git.

Clean Working Directory

Example: git clean -f removes all untracked files.

Preview Cleaning

git clean -n shows what would be deleted without actually removing files.

Include Directories

git clean -fd removes untracked directories too.

Contact: **Ravi Kiran Maroju**

Git Command: git clone

*By **Ravi Kiran Maroju***

Find Repository URL

Locate the URL of the Git repository you want to copy.

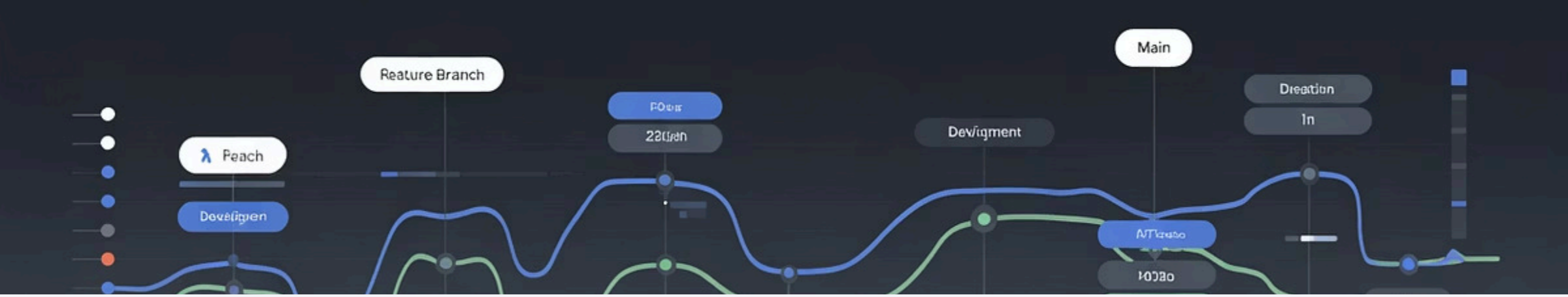
Run Clone Command

Copies a project from another place. **Example**: git clone https://github.com/user/repo.git

Start Working

The repository is now on your local machine with full history and branches.

Contact: **Ravi Kiran Maroju**



Git Command: git commit

Ravi Kiran Maroju

Save Changes

Saves your staged (added) changes.

Add Message

Example: git commit -m "Fixed bug"

Stage & Commit

git commit -a stages and commits all changes

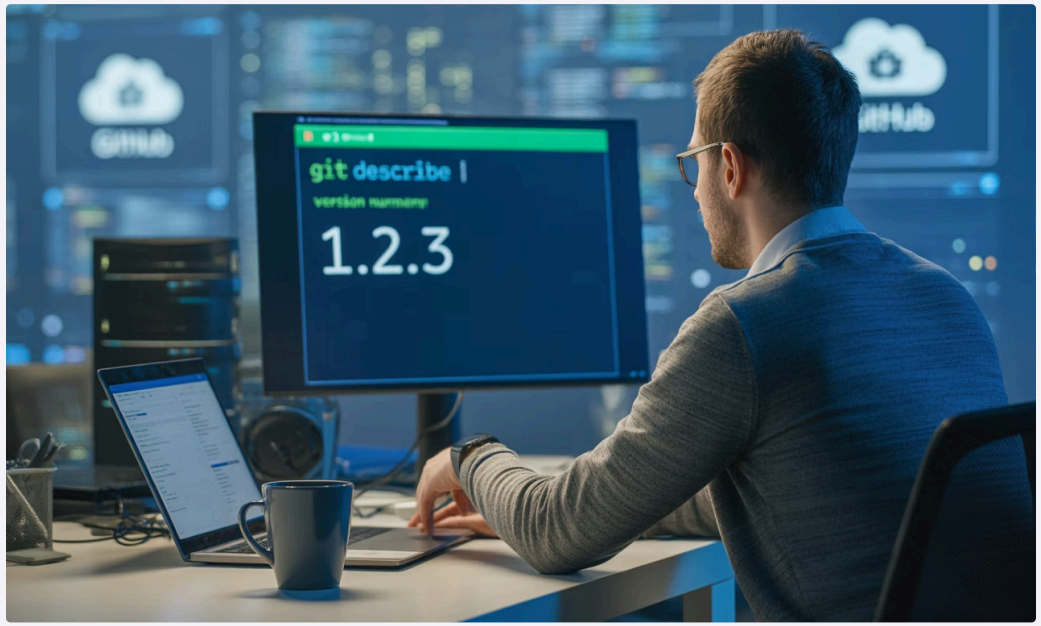
Amend Commit

git commit --amend updates previous commit

Git Command: git describe

What is git describe?

Shows a name for a commit using tags. **Example**: git describe might return something like v1.0-2-g9c1a1a.



The output format is typically [tag]-[number-of-commits]-g[commit-hash], making it easy to understand how far a commit is from the nearest tag.

- ***Maroju Ravi Kiran***

Git Command: git diff

View Changes

Shows the changes in files.
Example: git diff shows what was edited.

Compare Staged Changes

git diff --staged shows changes that are staged for commit.

Compare Branches

git diff branch1 branch2 shows differences between branches.

- **Ravi Kiran Maroju**

Git Command: git fetch

1

Download Updates

Get remote changes

2

Update References

Update remote branch info

3

Review Changes

Inspect before merging

4

Optional Merge

Merge only when ready

Gets updates from another repo, but doesn't apply them yet. **Example**: git fetch origin

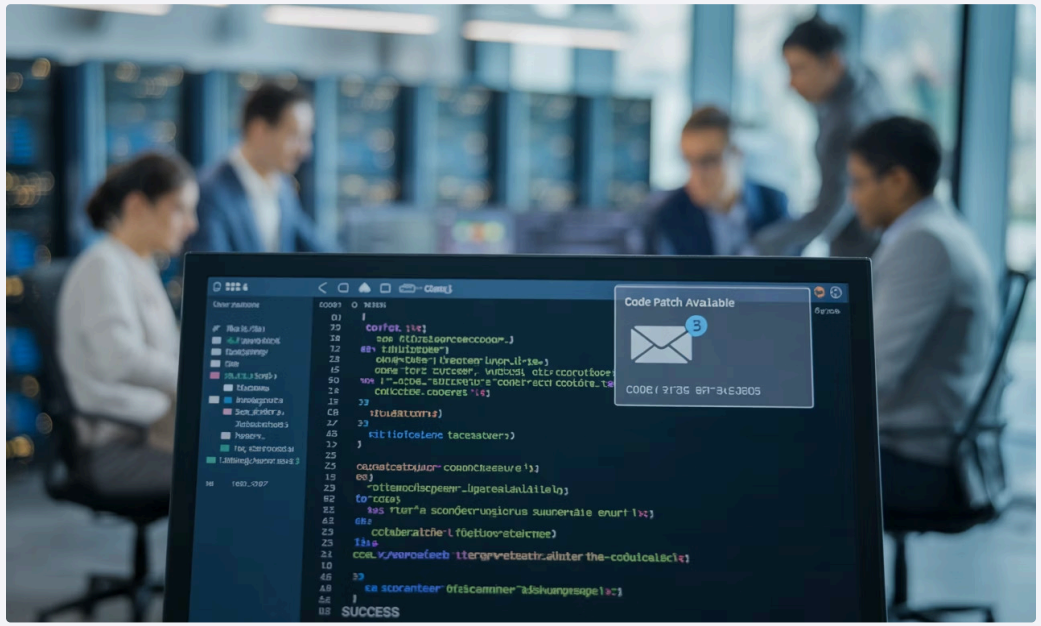
- **Ravi Kiran Maroju**

Git Command: git format-patch

Ravi Kiran Maroju

Create Email Patches

Prepares commits to be sent as patches. **Example**: git format-patch -1 makes a patch of the last commit.



Format-patch creates files that can be emailed to others and applied to their repositories, useful for projects that accept contributions via email.

- **Ravi Kiran Maroju**

Git Command: git gc

Ravi Kiran Maroju

Garbage Collection

Cleans and optimizes your repo.

Compress Objects

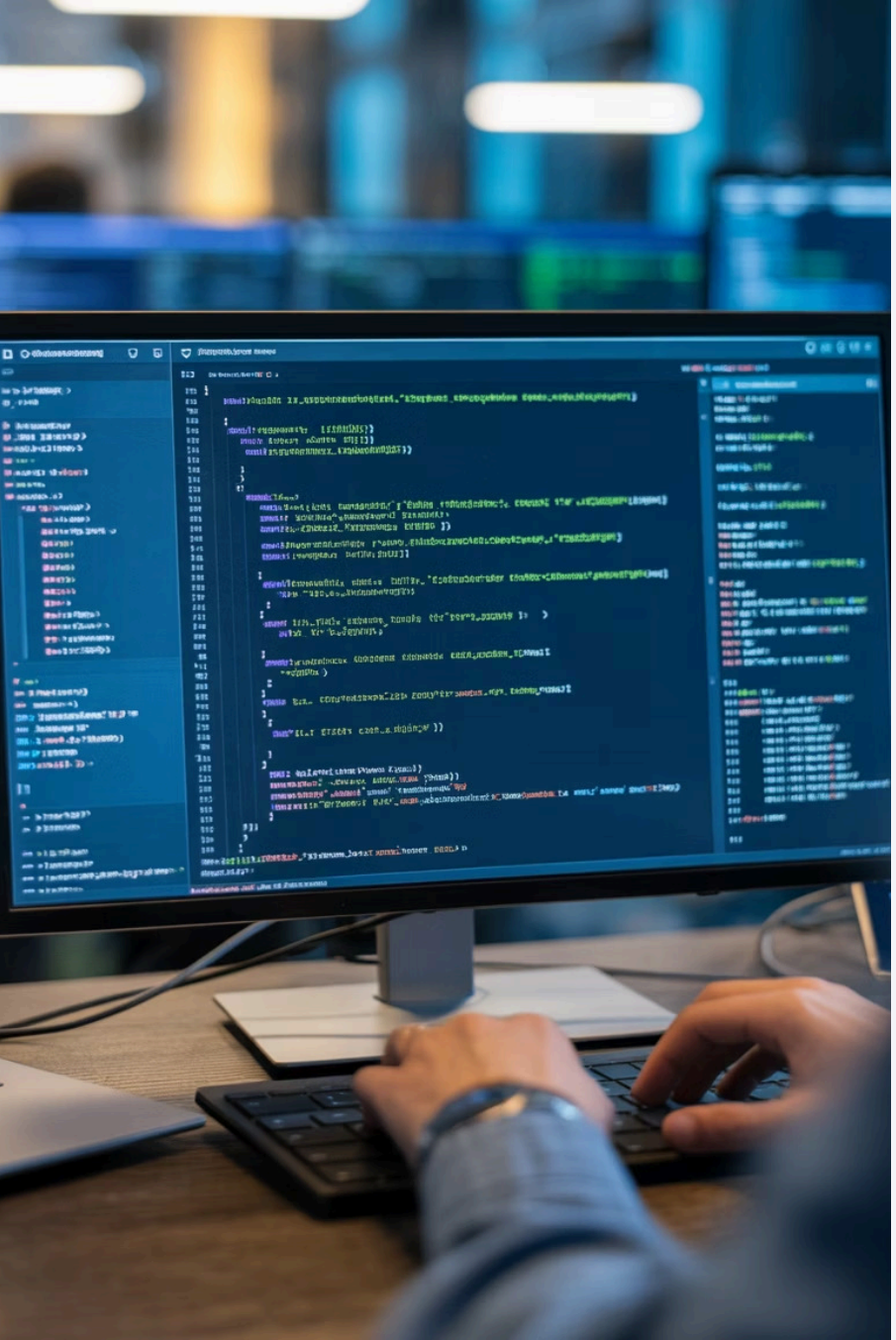
Packs loose objects to save space.

Improve Performance

Makes Git operations faster.

Automatic Process

Git runs this automatically when needed, but you can run it manually too.



Git Command: git grep

By **Ravi Kiran Maroju**

Search Repository

Searches for text in files.
Example: git grep "password" looks for "password" in code.

Search History

git grep "text" SHA searches in a specific commit.

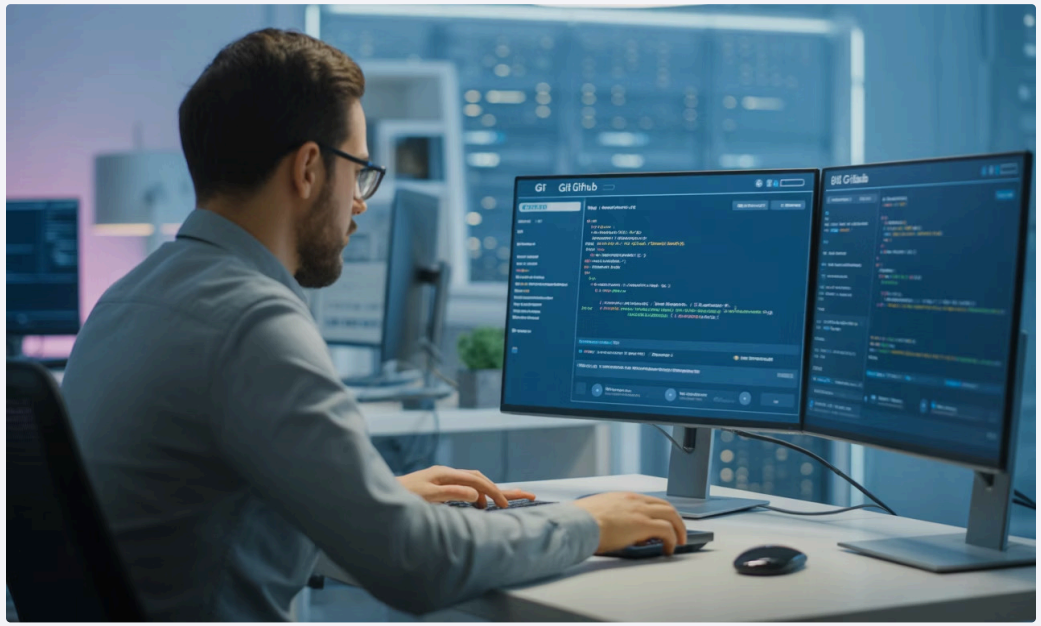
Show Line Numbers

git grep -n "text" shows line numbers with matches.

Git Command: git gui

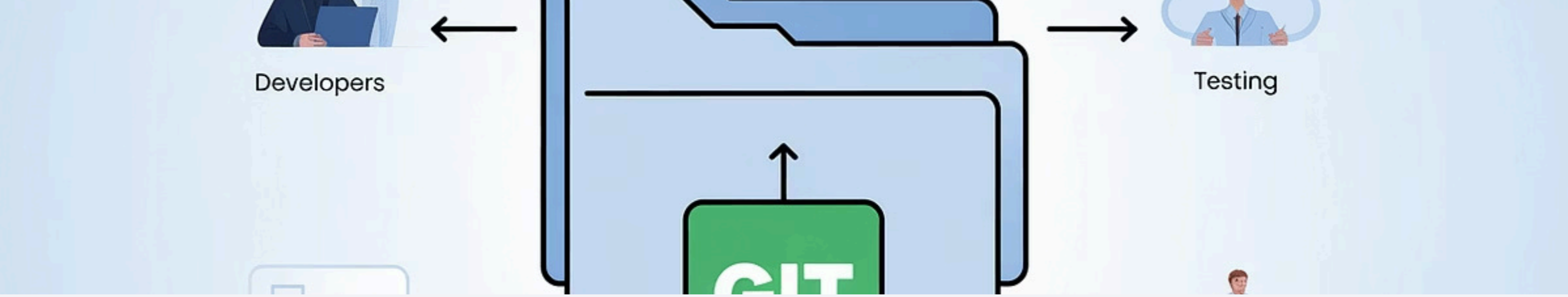
Graphical Interface

Opens a visual Git tool. **Example**: Run git gui to see a graphical interface.



Git GUI provides a visual alternative to command-line operations, making it easier for visual learners to understand Git workflows.

*By **Ravi Kiran Maroju***



Git Command: git init

Create Directory

Make a new folder or navigate to an existing one for your project.

Ravi Kiran Maroju

Initialize Repository

Starts a new Git project in a folder.
Example: git init makes the current folder a Git repo.

Start Working

Begin adding files and making commits to your new repository.

Git Command: git log

View History

Shows history of commits.

Basic Usage

Example: git log lists all saved changes.

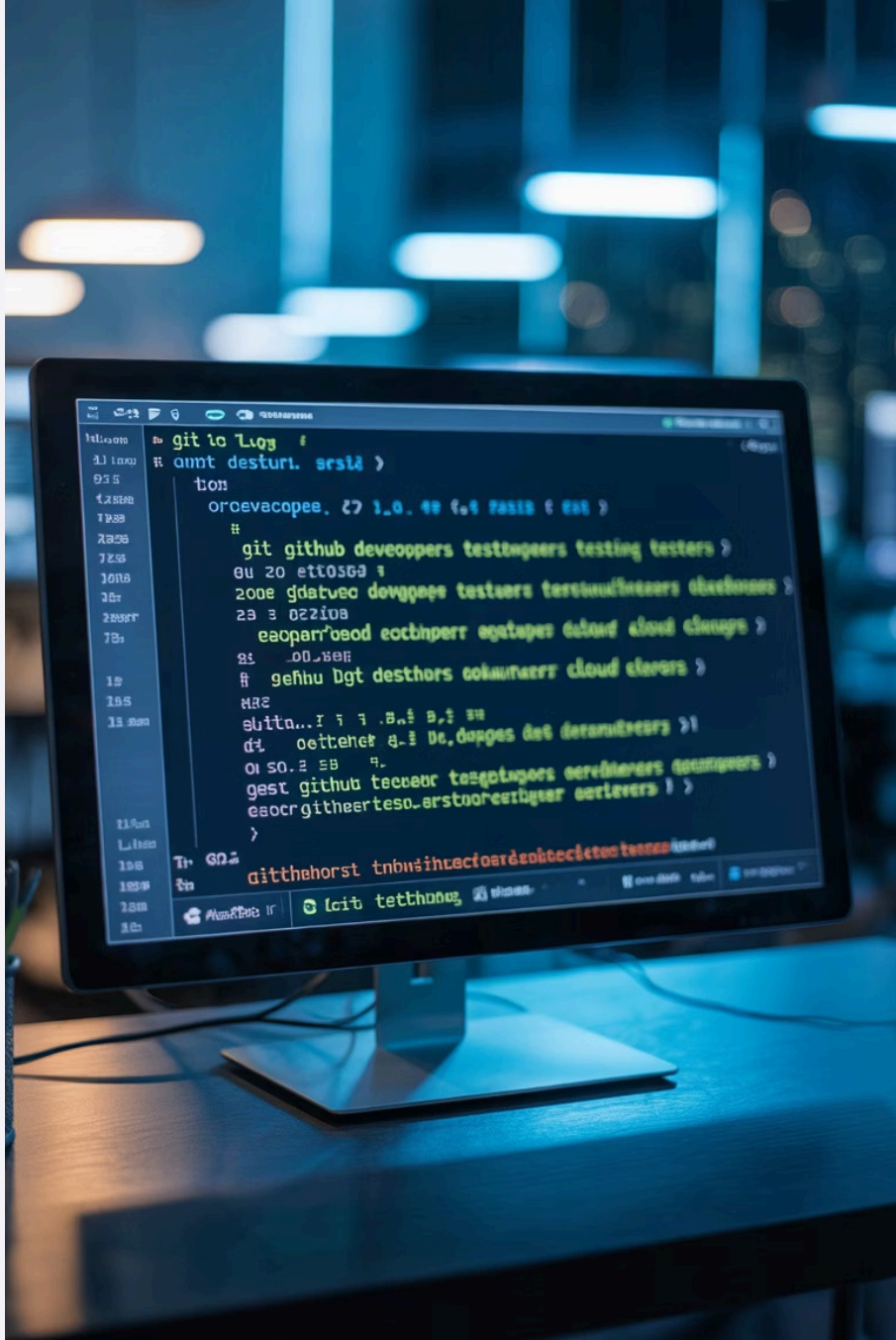
Graph View

git log --graph shows branch structure.

Compact View

git log --oneline shows condensed history.

Ravi Kiran Maroju



Git Command: git maintenance

Repository Maintenance

Runs cleanup and optimization tasks.

Scheduled Tasks

Can be configured to run automatically.

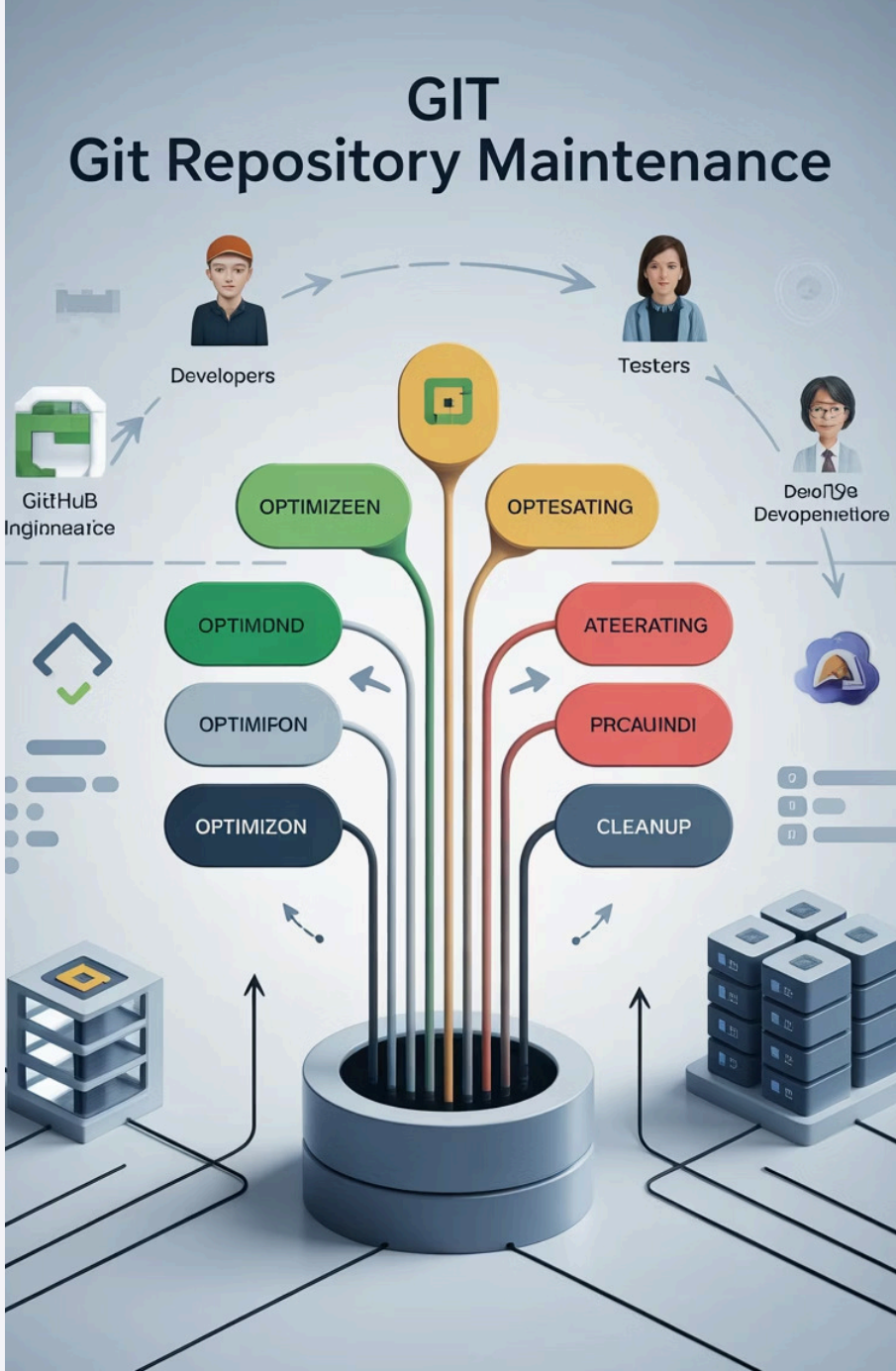
Performance Improvement

Keeps repository running efficiently.

Manual Execution

Example: git maintenance run

Presented by **Ravi Kiran Maroju**



Git Command: git merge

Select Target Branch

Checkout the branch to merge into

Merge Source Branch

Combine changes from source branch

Complete Merge

Commit the merged changes

Resolve Conflicts

Fix any competing changes

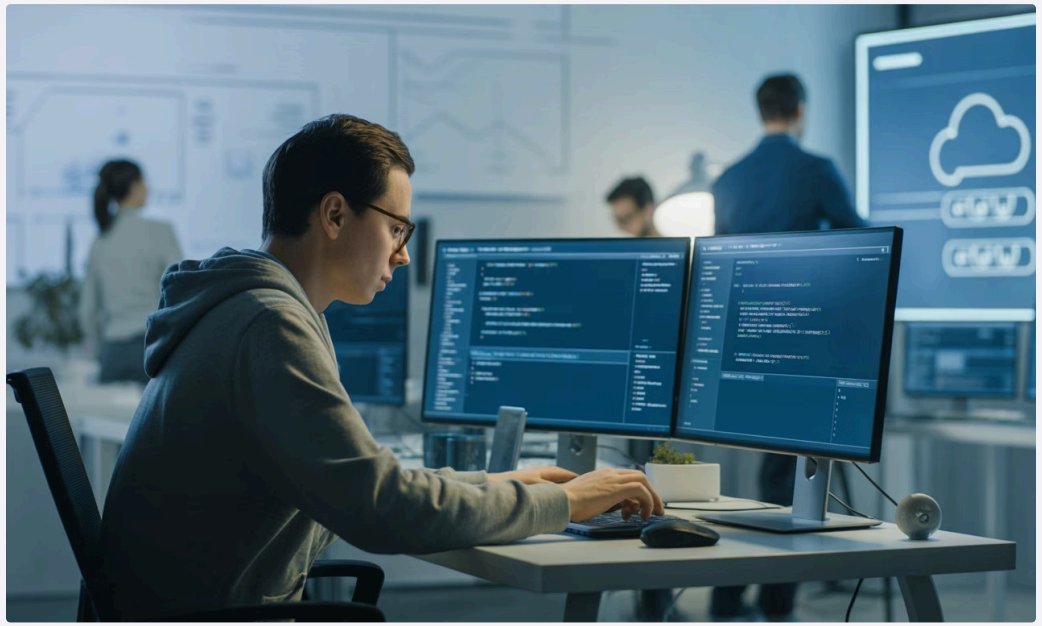
Combines two branches. **Example**: git merge dev adds dev changes into current branch.

*Presented by **Ravi Kiran Maroju***

Git Command: git mv

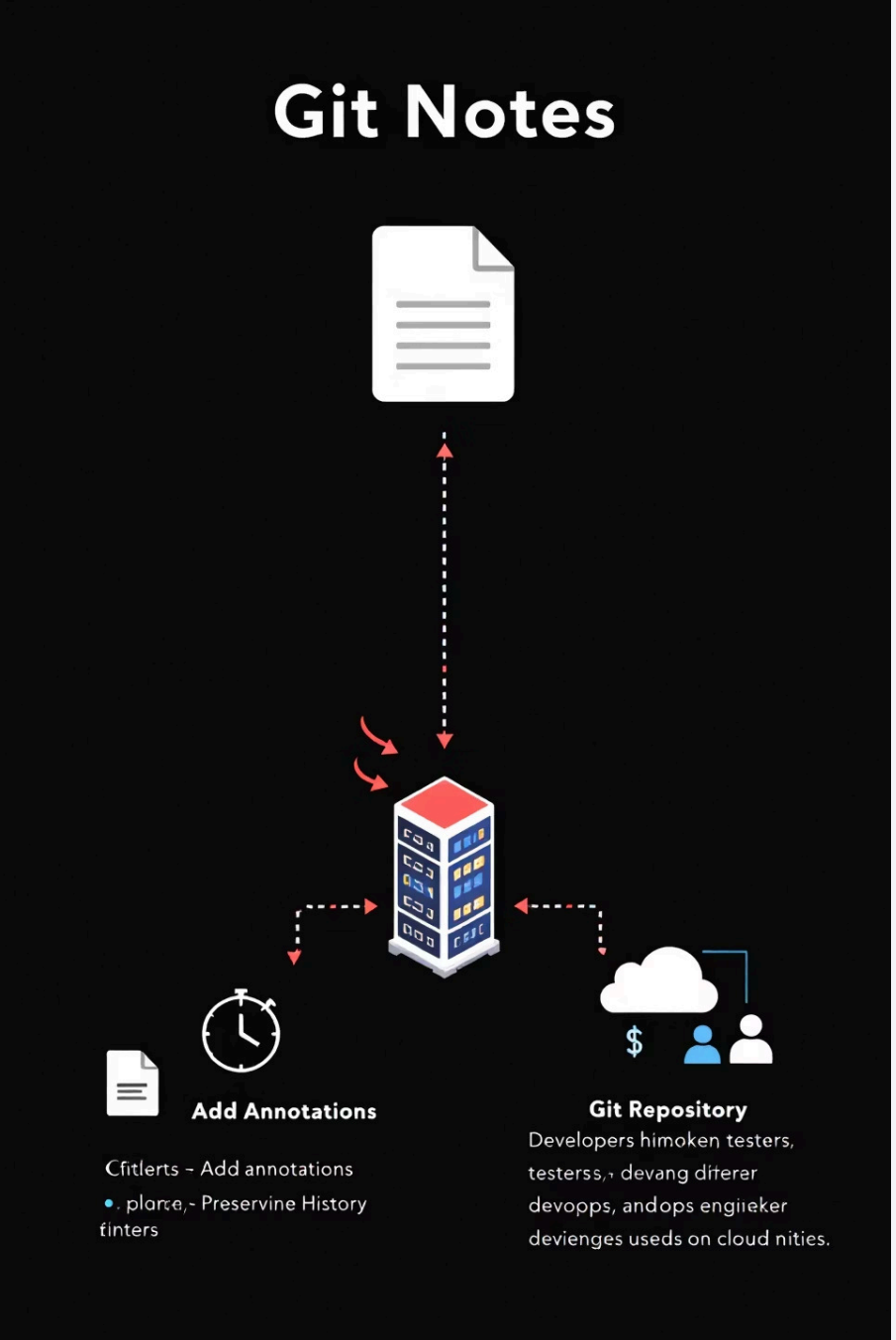
Move or Rename Files

Moves or renames a file. **Example**: git mv old.txt new.txt renames the file.



Using git mv instead of regular file system operations ensures Git tracks the file's history across the rename or move operation.

Presented by **Ravi Kiran Maroju**



Git Command: git notes

Annotate Commits

Adds notes to a commit.
Example: git notes add -m "Reviewed by team" adds a note.

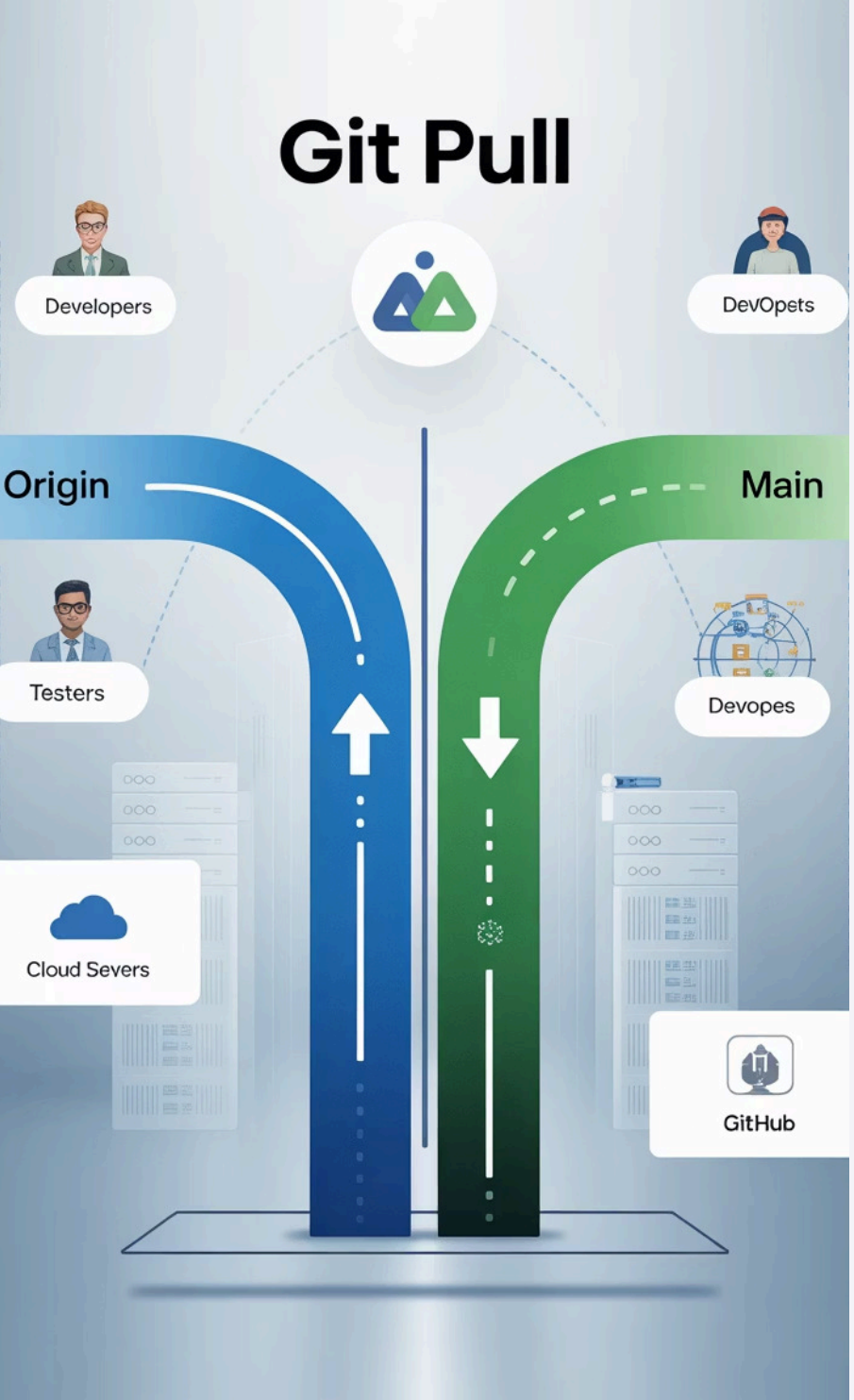
View Notes

git notes show [commit] displays notes for a specific commit.

Edit Notes

git notes edit [commit] modifies existing notes on a commit.

Ravi Kiran Maroju



Git Command: git pull

1

Fetch Remote Changes

Download updates from remote

2

Merge Changes

Integrate remote changes into local branch

3

Resolve Conflicts

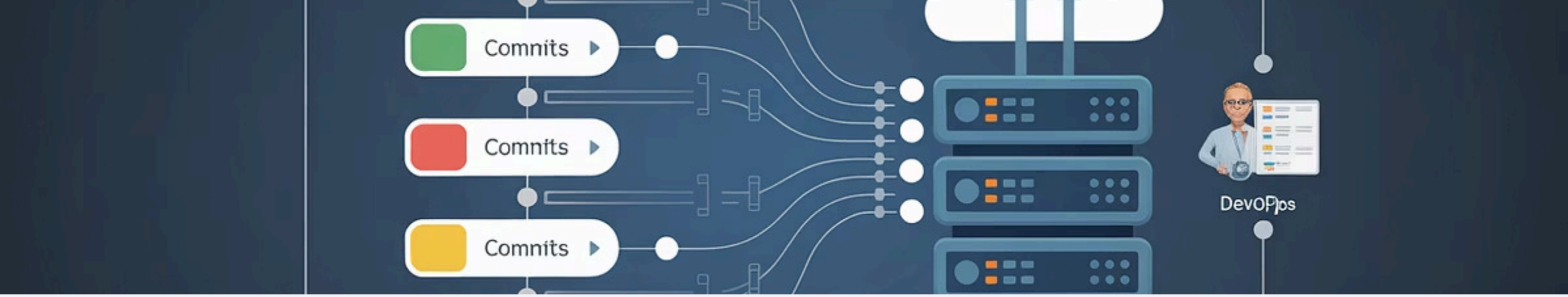
Fix any competing changes if needed

Update Working Directory

Working files reflect merged state

Fetches and applies new changes from a remote repo. **Example**: git pull origin main

- ***Ravi Kiran Maroju***



Git Command: git push

Make Local Commits

Create and save your changes locally first.

Created by **Ravi Kiran Maroju**

Push to Remote

Sends your saved changes to a remote repo. **Example**: git push origin main

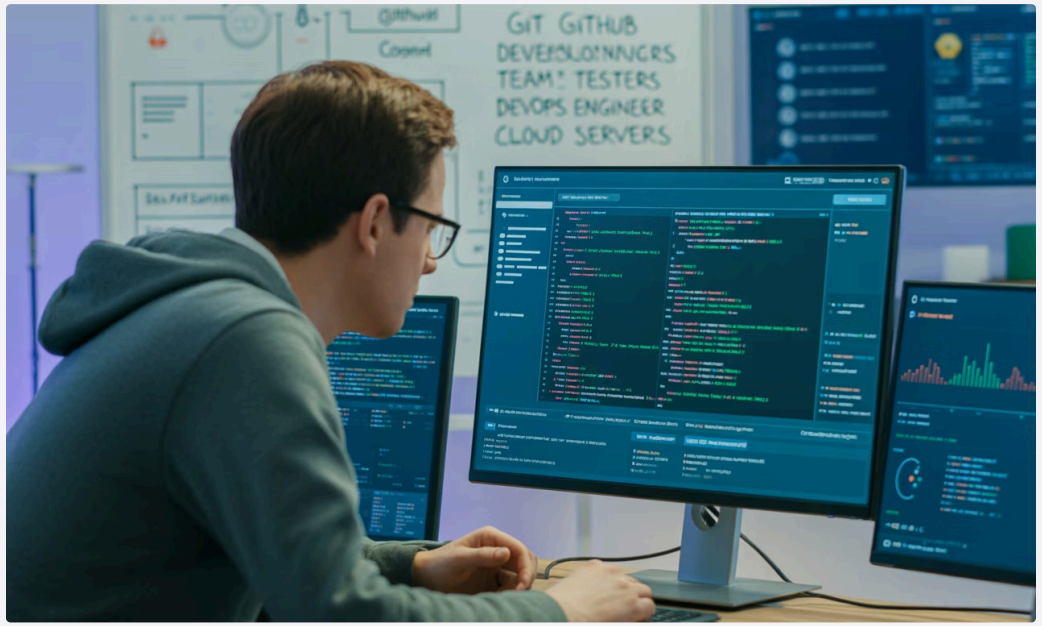
Verify Changes

Check the remote repository to confirm your changes were uploaded successfully.

Git Command: git range-diff

Compare Commit Ranges

Compares two sets of commits. **Example**: git range-diff origin/main main



Range-diff is particularly useful for seeing how a branch has changed after being rebased, showing which commits were modified and how.

Created by **Ravi Kiran Maroju**

Git Command: git rebase

1

Start with feature branch

Work on your feature branch

Update main branch

Fetch latest changes from main

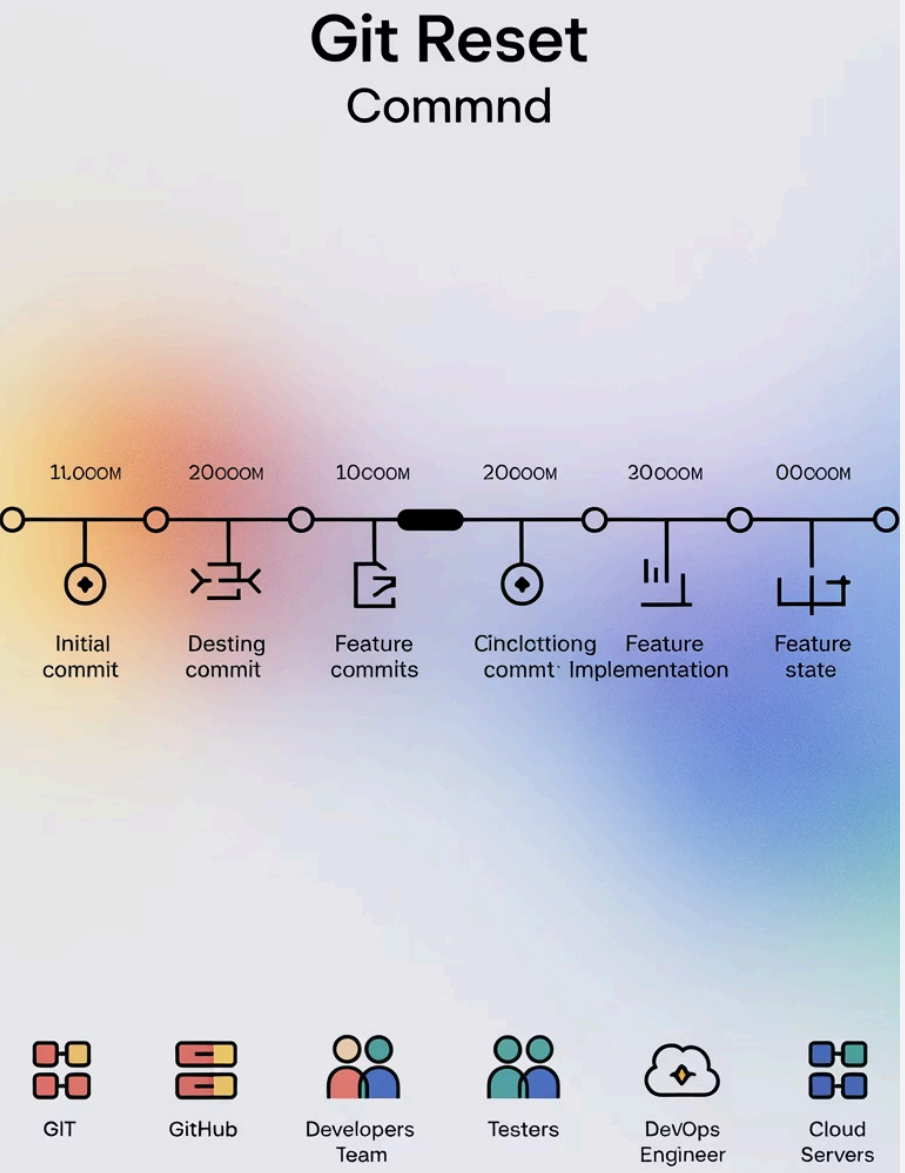
3

Rebase onto main

Move your changes on top of latest main

Moves your changes to a new base commit. **Example**: git rebase main

Created by **Ravi Kiran Maroju**



Git Command: git reset

Time Travel

Goes back to an earlier version.

Undo Changes

Example: git reset --hard HEAD~1 removes the last commit.

Unstage Files

git reset file.txt removes file from staging area.

Use With Caution

Hard reset can permanently lose work.

Ravi Kiran Maroju

Git Command: git restore

Undo Working Changes

Undoes changes to files.
Example: git restore file.txt undoes edits to file.txt.

Unstage Files

git restore --staged file.txt removes file from staging area without changing it.

Restore From Commit

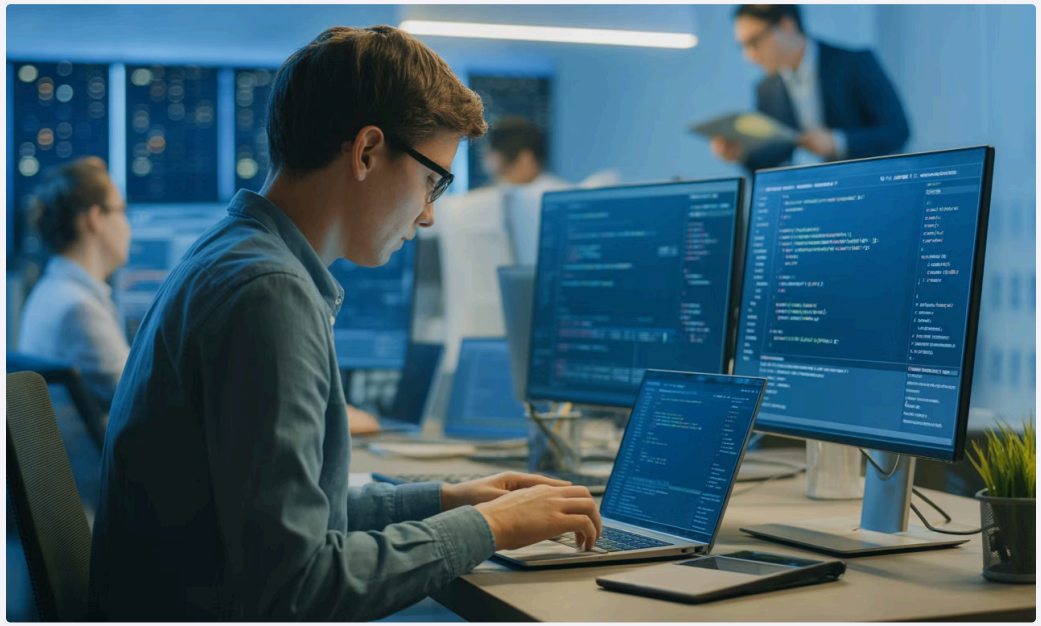
git restore --source=HEAD~1 file.txt restores file from a specific commit.

Ravi Kiran Maroju

Git Command: git revert

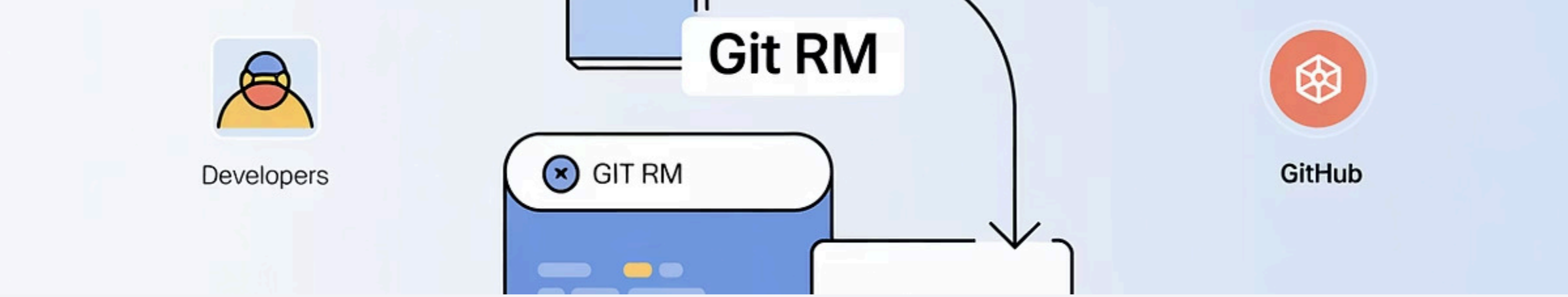
Safe Undo

Makes a new commit that undoes a past one. **Example**: git revert abc123 cancels changes from that commit.



Unlike reset, revert is safe for shared branches because it adds a new commit rather than rewriting history.

Ravi Kiran Maroju



Git Command: git rm

Ravi Kiran Maroju

Remove Files

Removes a file and tracks its deletion.

Delete & Stage

Example: git rm file.txt

Remove Directories

git rm -r directory/ removes a directory and its contents.

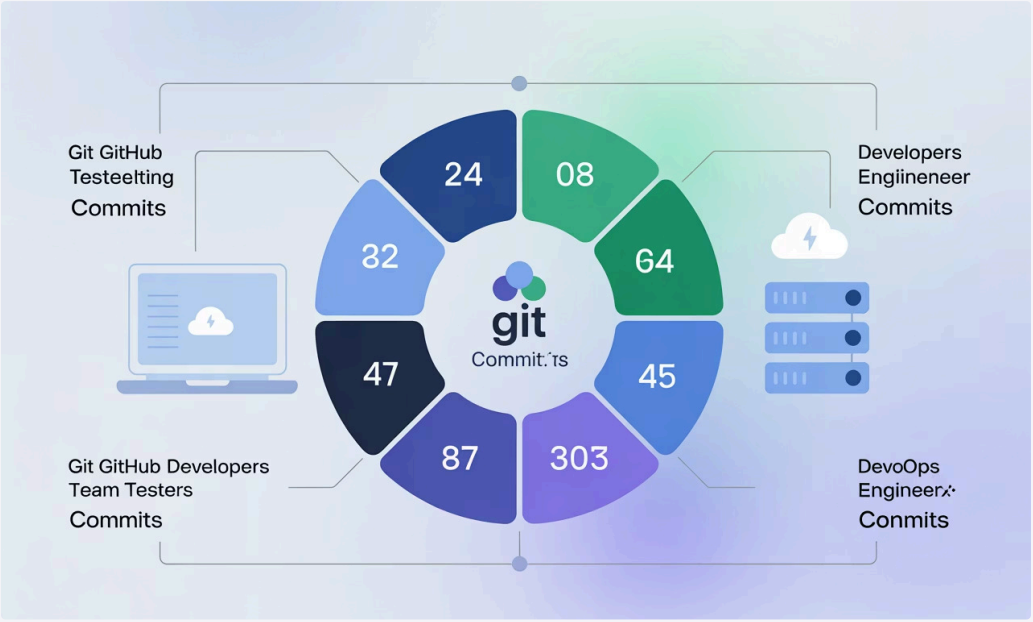
Keep Local Copy

git rm --cached file.txt removes from Git but keeps the file.

Git Command: git shortlog

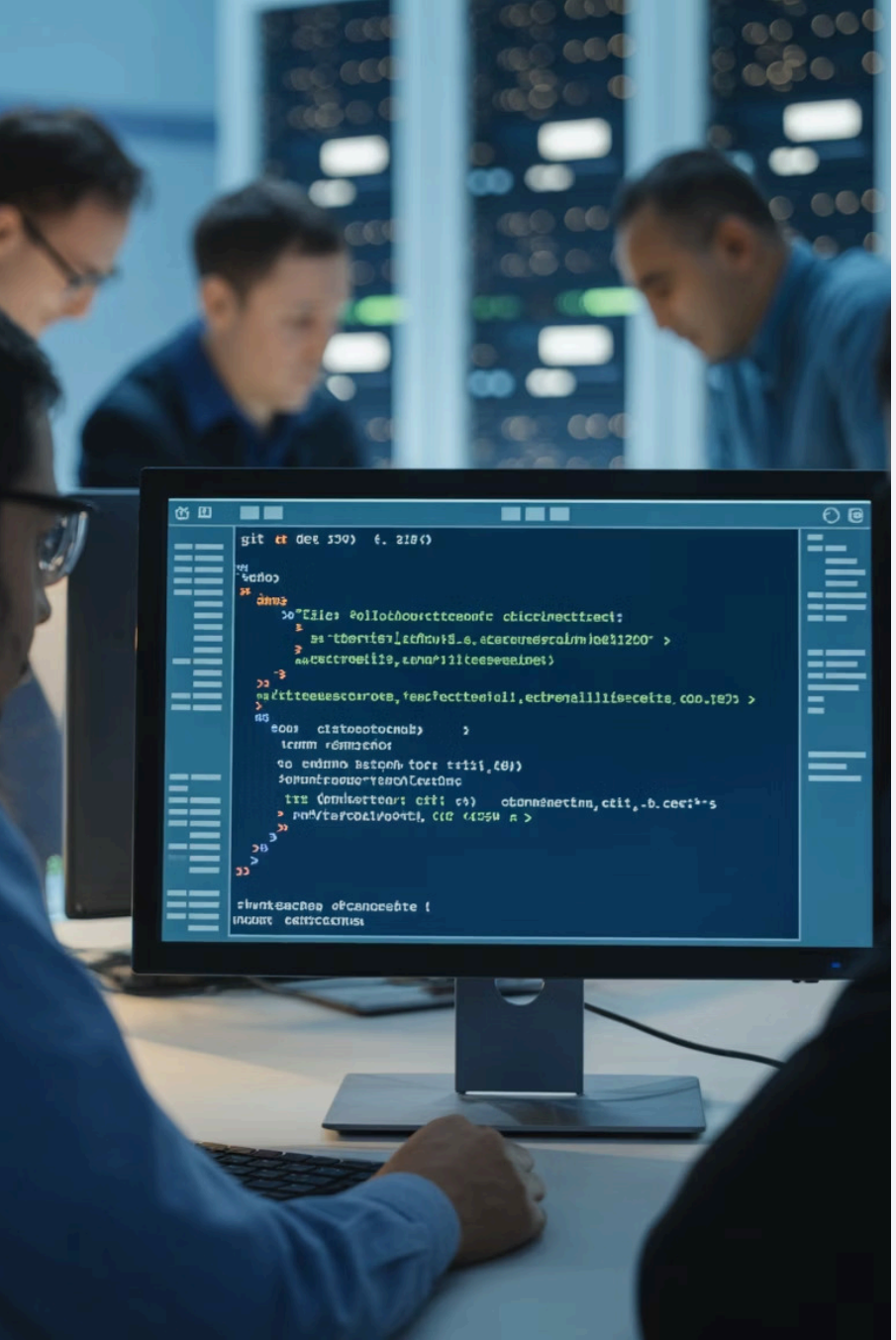
Author Summary

Shows commit summary by each author. **Example**: git shortlog



Shortlog is particularly useful for project managers to see who has contributed what to a project, or for generating release notes.

Ravi Kiran Maroju



Git Command: git show

Ravi Kiran Maroju

View Commit Details

Displays information about a commit. **Example**: git show abc123

Show Tags

git show v1.0 displays information about a tagged commit.

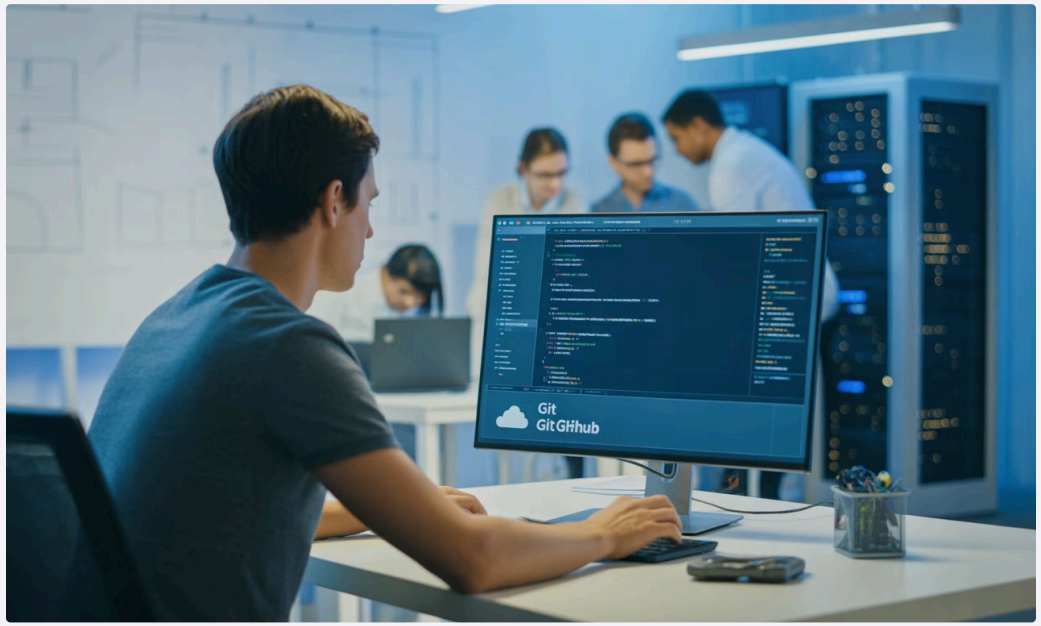
View File at Commit

git show abc123:file.txt shows a file's content at a specific commit.

Git Command: git sparse-checkout

Partial Checkout

Lets you only use certain folders. **Example**: git sparse-checkout set foldername



Sparse checkout is useful for large repositories when you only need to work with a small subset of files, saving disk space and improving performance.

LinkedIn: Ravi Kiran Maroju

Git Command: git stash

Save Changes

git stash save "message"

List Stashes

git stash list

View Stash

git stash show stash@{0}

Apply Stash

git stash apply stash@{0}

Temporarily hides your changes. **Example**: git stash saves changes for later.

Presented by **Ravi Kiran Maroju**

Git Command: git status

Repository State

Shows current changes and file states.

Branch Information

Displays which branch you're on.

File Status

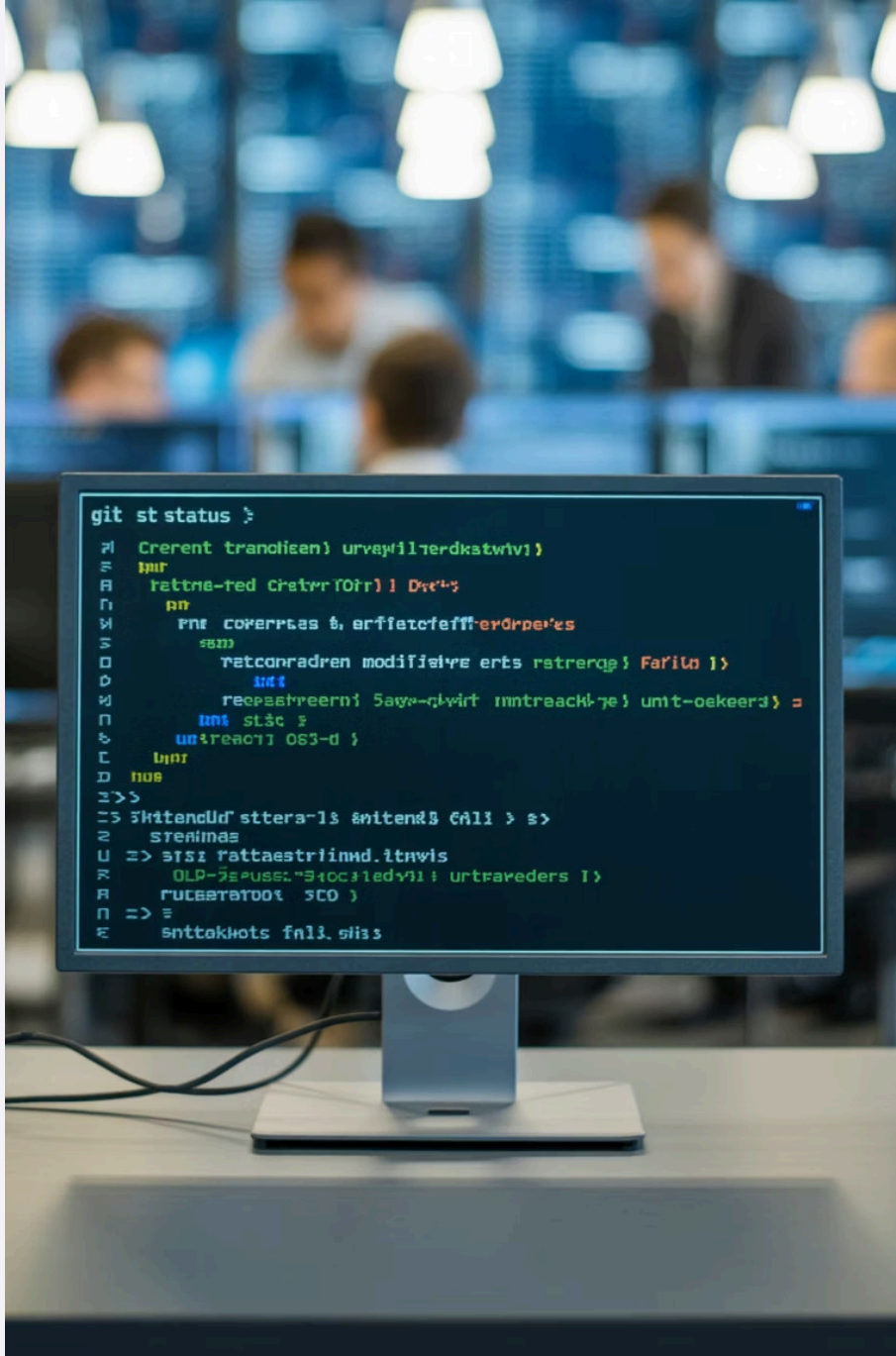
Shows modified, staged, and untracked files.

Simple Usage

Example: git status

Presented by **Ravi Kiran Maroju**

LinkedIn: Ravi Kiran Maroju



Git Command: git submodule

Add Submodule

Adds or updates another repo inside your repo. **Example**: git submodule add https://github.com/lib/project.git

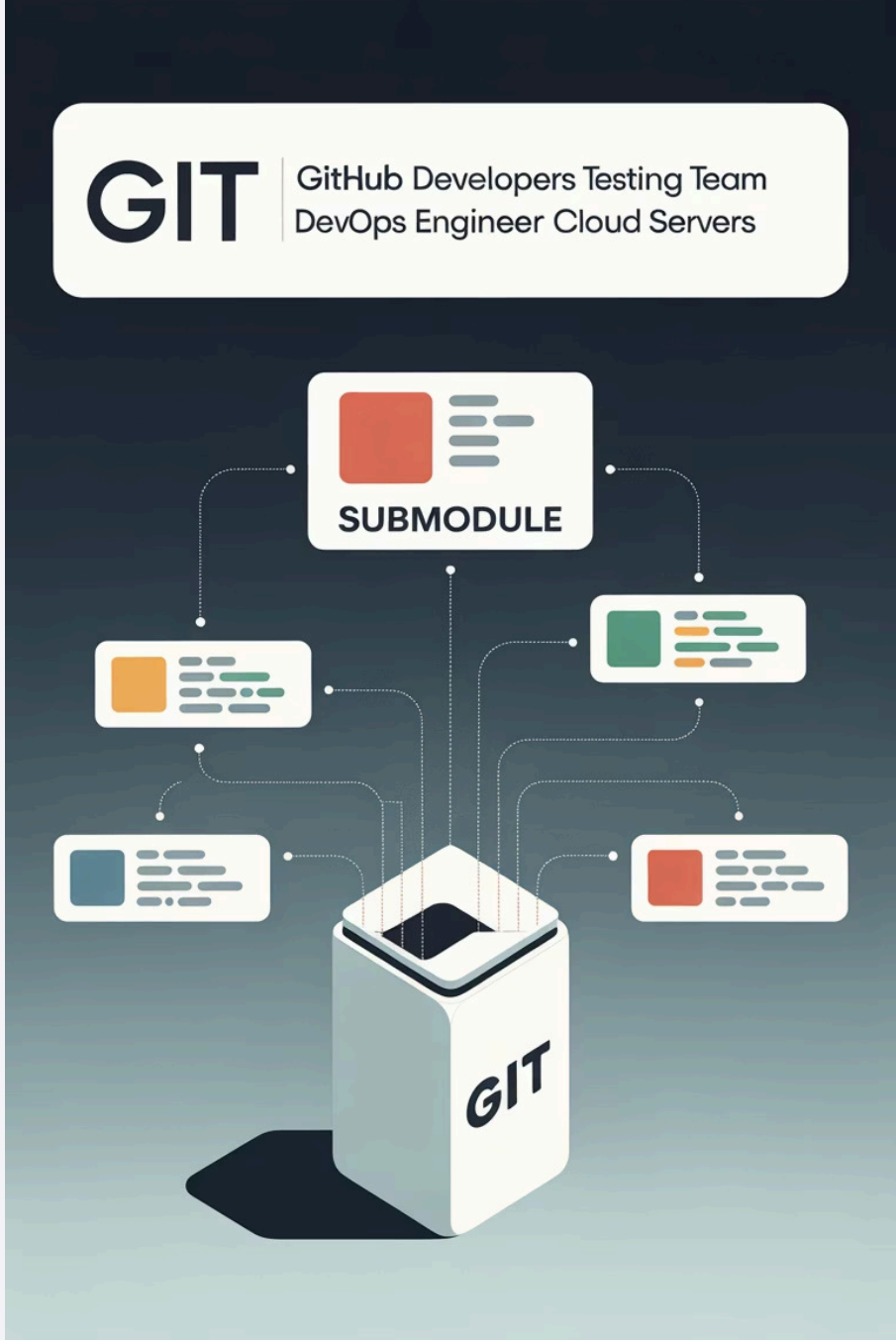
Initialize Submodules

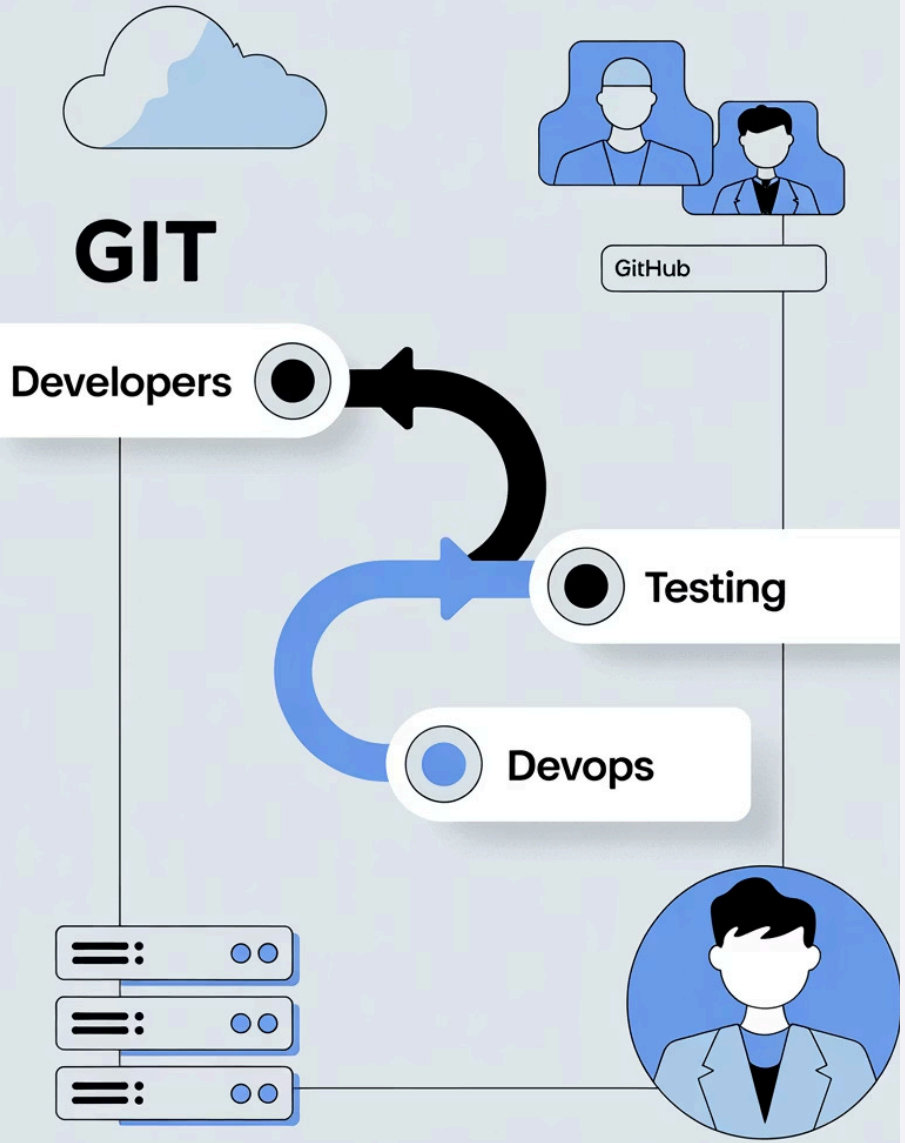
After cloning a repo with submodules: git submodule init followed by git submodule update

Update All Submodules

git submodule update --remote updates all submodules to their latest versions

Presented by **Ravi Kiran Maroju**





Git Command: git switch

Change Branches

Switches to another branch.

Basic Usage

Example: git switch feature-1

Create & Switch

git switch -c new-branch creates and switches

Return to Previous

git switch - returns to previous branch

Ravi Kiran Maroju

Git Command: (git tag)

Create Version Markers

Marks a commit with a label. **Example**: git tag v1.0 adds a tag named v1.0

List Tags

git tag shows all tags in the repository.

Annotated Tags

git tag -a v1.0 -m "Release version 1.0" creates a tag with a message.

Share Tags

git push origin --tags pushes all tags to the remote repository.

Ravi Kiran Maroju

git annotate Shows who made each line change in a file. Example: git annotate file.txt

git blame Tells who last changed each line of a file and when. Example: git blame index.html

Ravi Kiran Maroju

by Ravi Kiran

Finding Commits

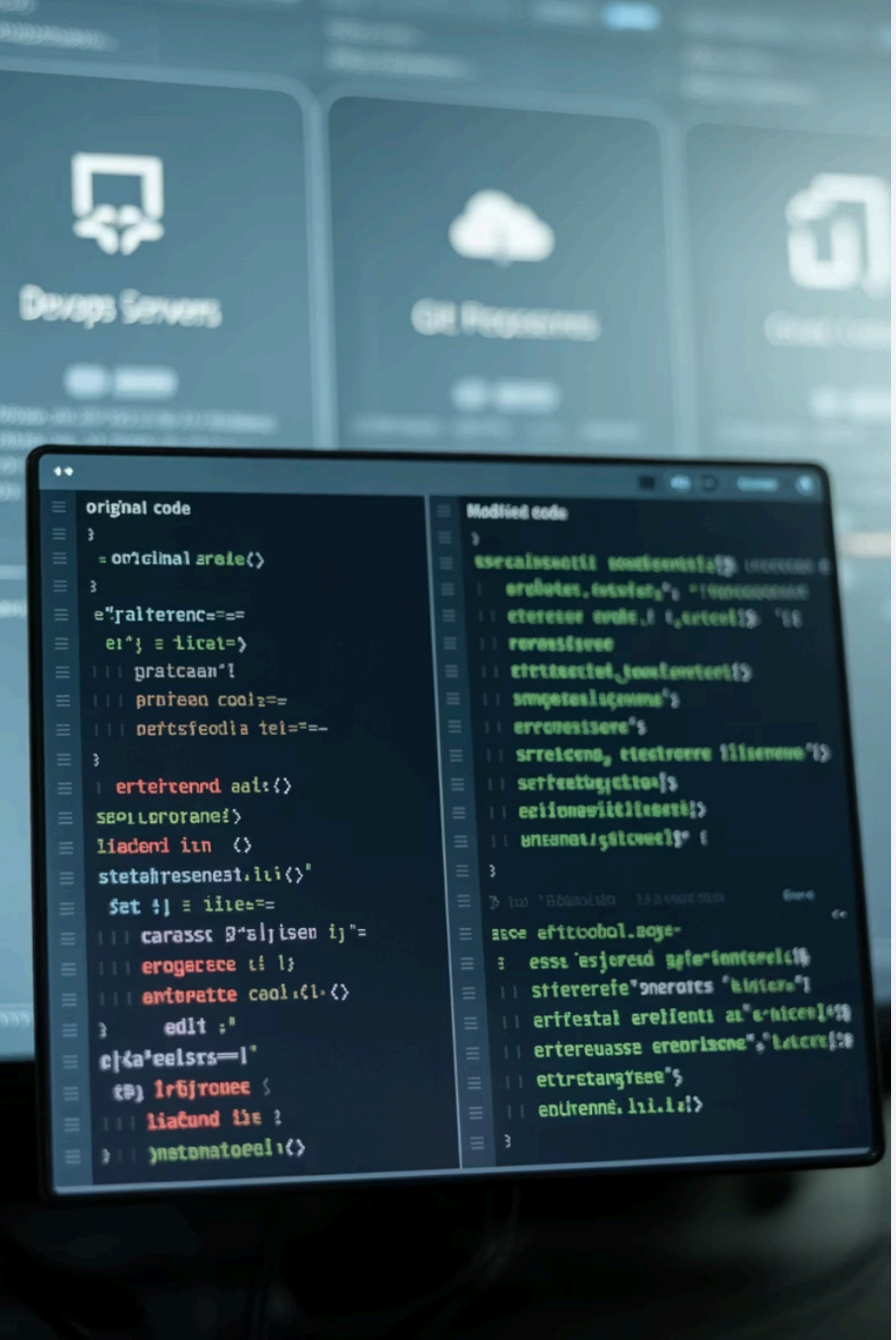
git cherry

Finds commits in your branch that are not in the main one.
Example: git cherry main

git count-objects

Shows how many Git objects are not packed and their size.
Example: git count-objects -v

https://www.linkedin.com/in/ravi-kiran-maroju-175225134/



Comparing Changes

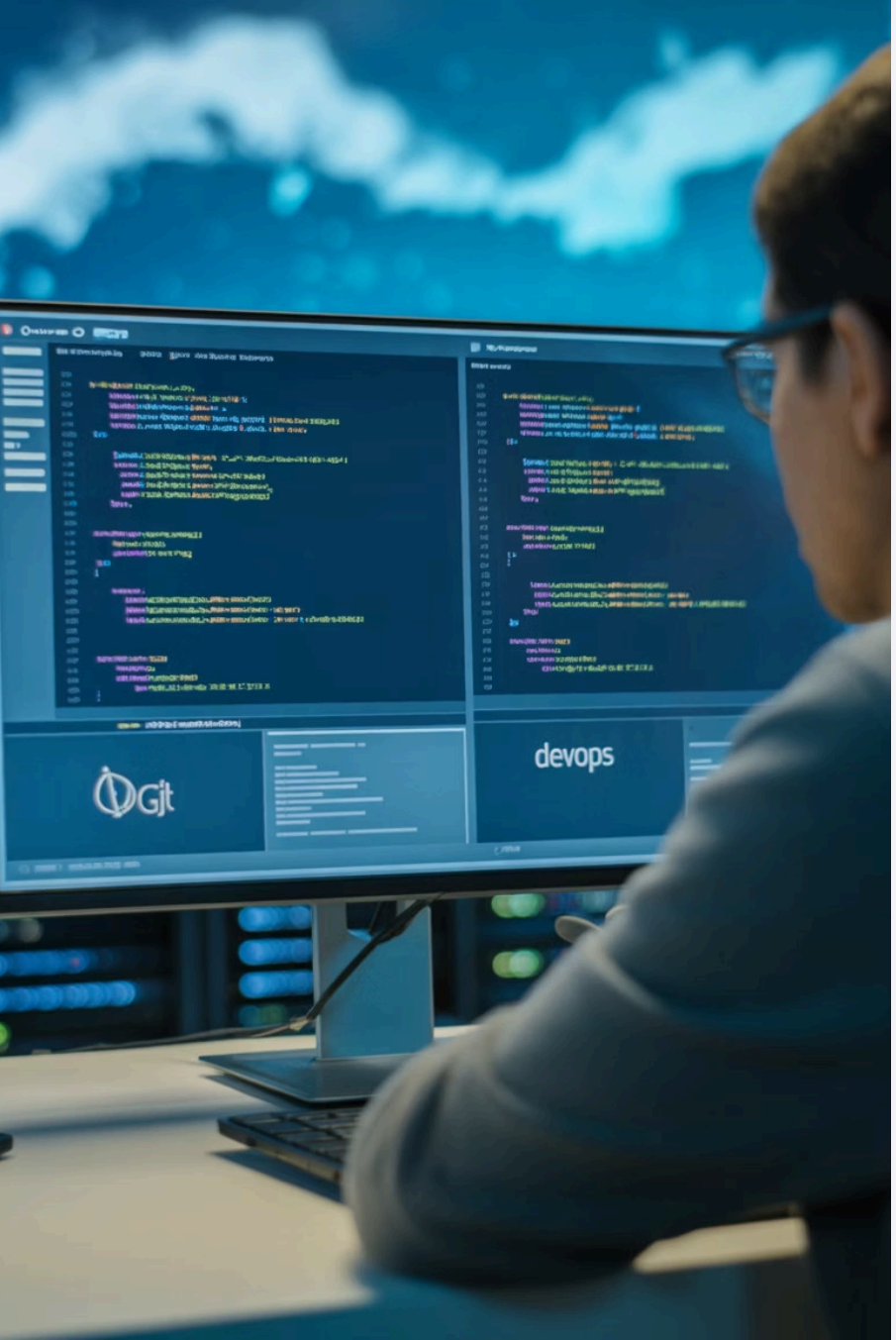
git diff-files

Shows changes between your current files and staged files. Example: git diff-files

git diff-index

Compares the current working files with a saved tree. Example: git diff-index HEAD

Connect with us: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**



Visualization Tools

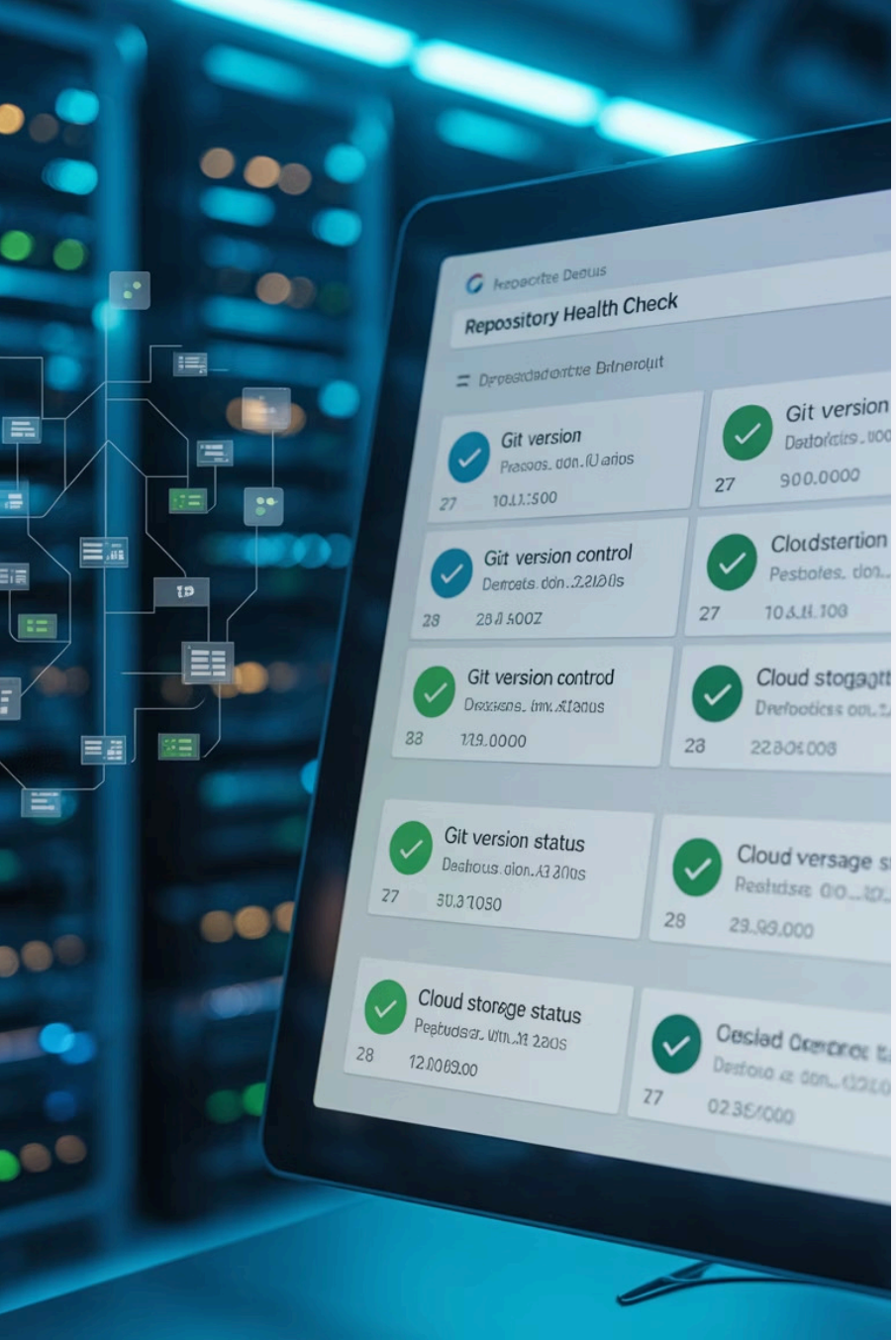
git difftool

Shows file changes using a visual tool. Example: git difftool

git for-each-ref

Lists all refs (branches, tags) with details. Example: git for-each-ref

Contact: **Ravi Kiran Maroju**



Repository Health

git fsck

Checks the health of your Git repo and its objects. Example: git fsck

git get-tar-commit-id

Gets the commit ID from a .tar file made by Git. Example: git get-tar-commit-id < file.tar

https://www.linkedin.com/in/ravi-kiran-maroju-175225134/

Getting Help

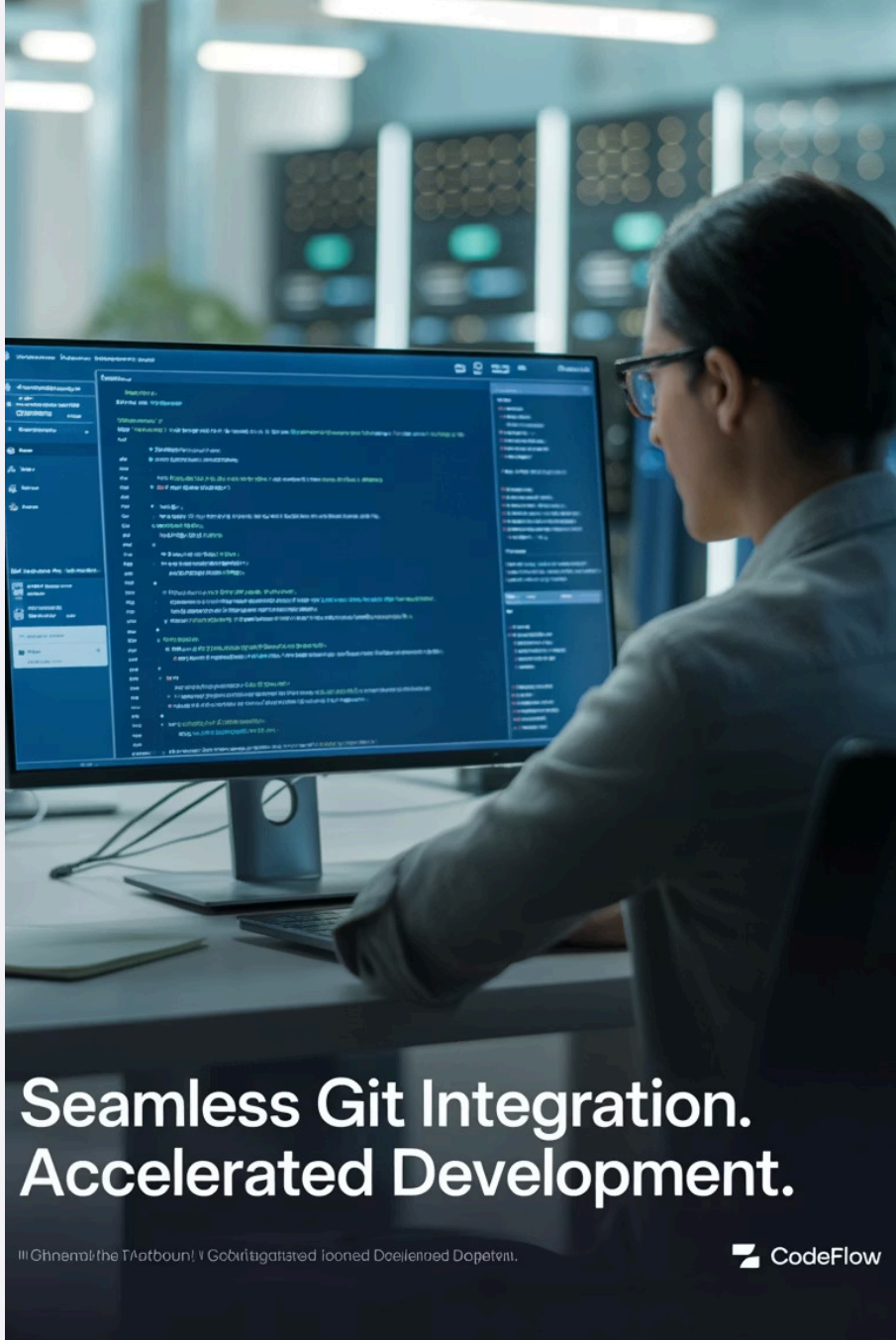
git help

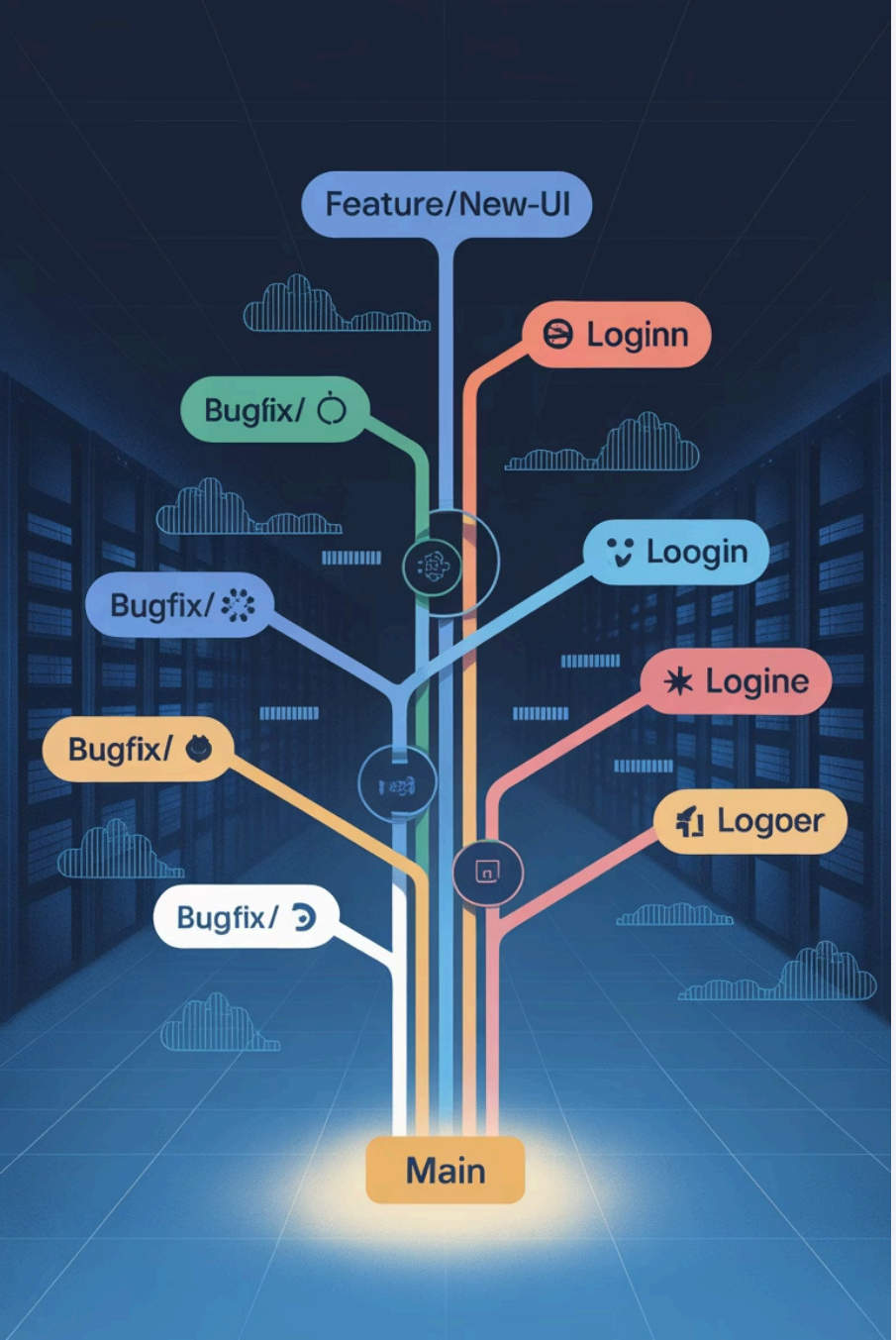
Opens Git's help for commands. Example: git help commit

git instaweb

Starts a temporary web browser to explore your Git repo. Example: git instaweb --httpd=webrick

For more information: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**





Branch Management

git merge-base

Finds the shared parent of two branches. Example: git merge-base branch1 branch2

2

git merge-file

Merges changes from three versions of a file. Example: git merge-file file1 base file2

Connect with me: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**

Merge Helpers

git merge-index

Helps merge multiple files during a merge

git merge-one-file

A helper used during file merges

git merge-index Helps merge multiple files during a merge. Example: git merge-index git-merge-one-file

git merge-one-file A helper used during file merges. Usually not run manually.

For more information: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**

Conflict Resolution

git mergetool

Opens a tool to fix merge conflicts

git mergetool–lib

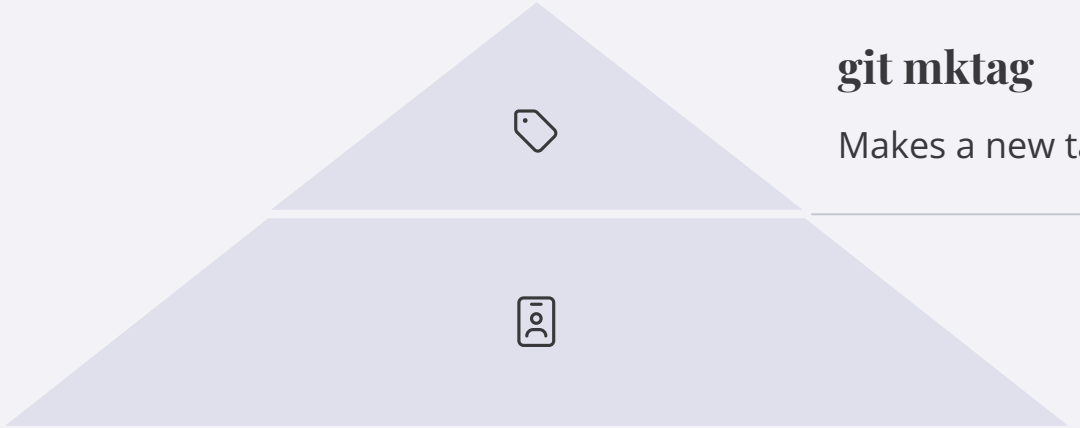
Gives shared functions to help resolve file conflicts

git mergetool Opens a tool to fix merge conflicts. Example: git mergetool

git mergetool–lib Gives shared functions to help resolve file conflicts. Used internally by Git.

For more information, visit **Ravi Kiran Maroju's LinkedIn profile**.

Tag Management



git mktag

Makes a new tag object in Git

git name-rev

Shows the name of a commit instead of its hash

git mktag Makes a new tag object in Git. Example: git mktag < tag-info.txt

git name-rev Shows the name of a commit instead of its hash. Example: git name-rev HEAD

For more information, visit **LinkedIn**

Object Management

Command	Description	Example
git pack-objects	Packs Git objects into archive files	git pack-objects packfile
git prune	Deletes old and unreachable Git objects	git prune

For more information, visit **Ravi Kiran Maroju**

Repository Maintenance

1

Prune Packed

git prune-packed Removes loose objects already in pack files. Example: git prune-packed

2

Quilt Import

git quiltimport Imports patches from a quilt patch directory. Example: git quiltimport patches/

For more information: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**

Tree Operations

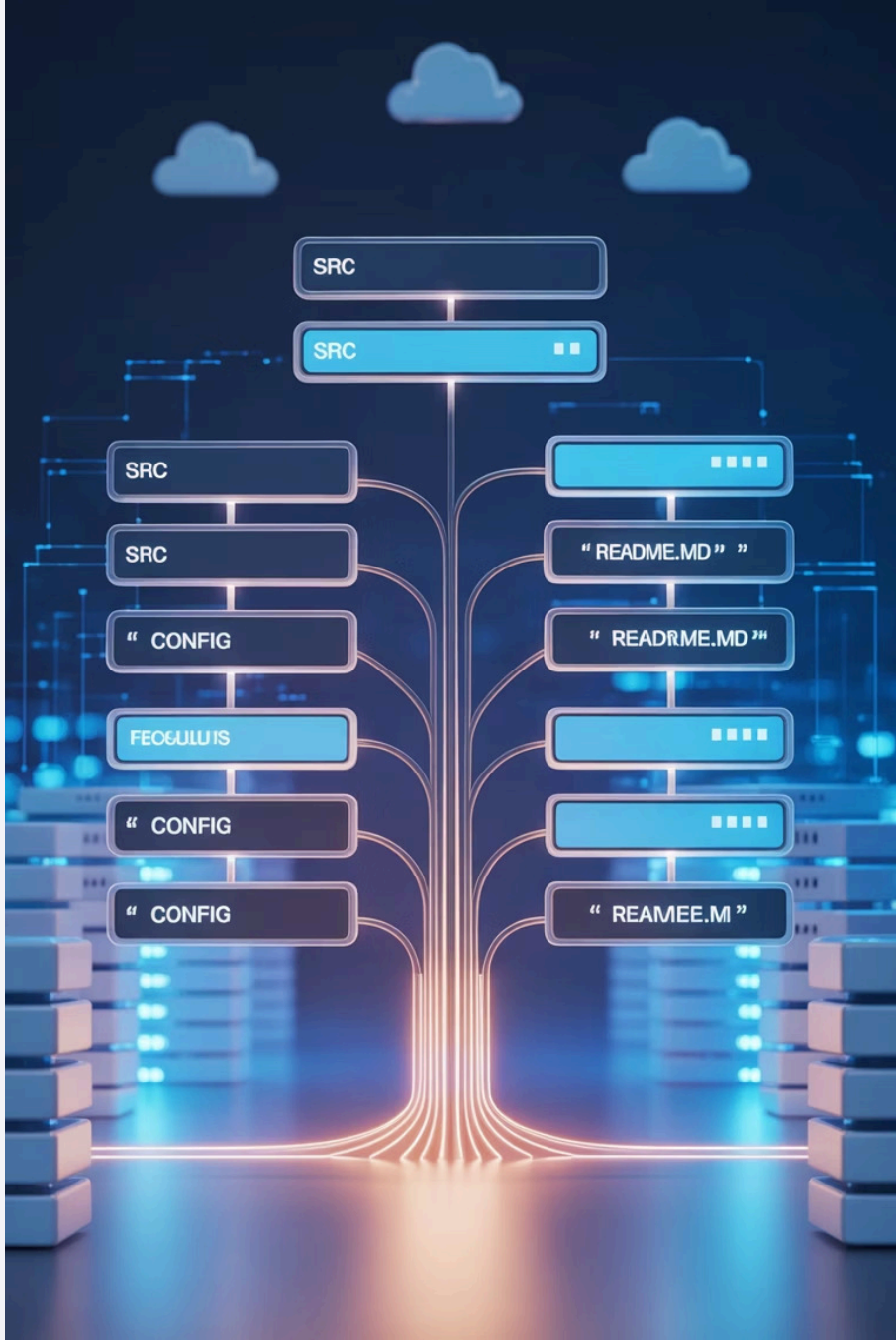
git read-tree

Loads tree data into the index for next operations. Example: git read-tree HEAD

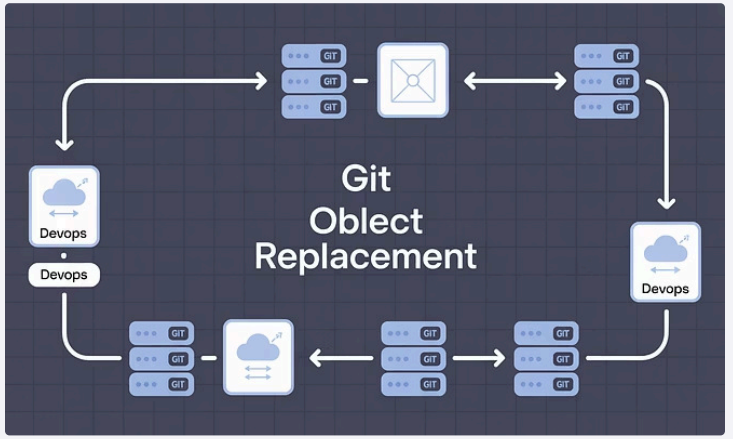
git receive-pack

Receives pushed changes in a server. Usually used by Git servers.

For more information, visit **my LinkedIn profile**.



Repository Optimization



git repack Packs all loose Git objects to save space. Example: git repack -a -d

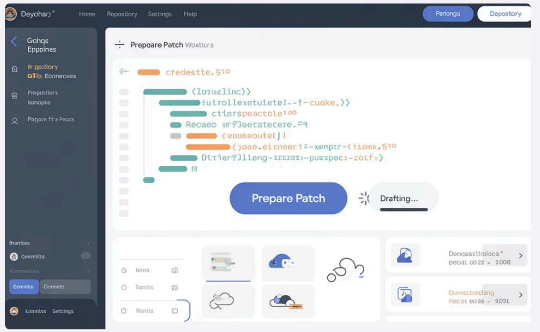
git replace Temporarily replace a commit or object with another. Example: git replace abc123 def456

Visit my profile: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**

Conflict Management

git rerere

Remembers how you fixed a conflict so it can be reused.
Example: git rerere



git send-email

Sends patches by email.
Example: git send-email *.patch

For more information: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**

Internationalization

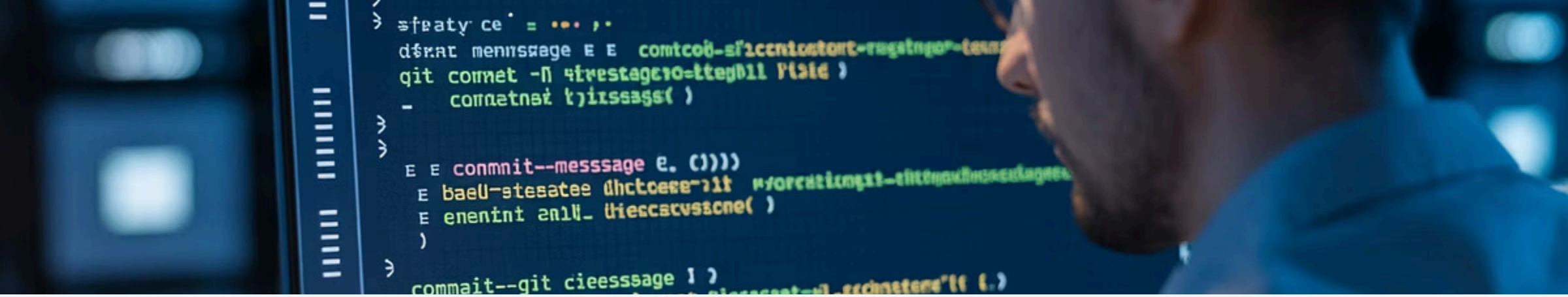
git sh-i18n

Handles international (language) support in Git scripts. Used in scripts, not manually.

git shell

Limited shell used for SSH access to Git only. Example: Used by Git server admins.

For more information, visit **Ravi Kiran Maroju's LinkedIn**



Shell Scripting

git sh-setup

Gives basic setup for Git shell scripts. Used internally.

git sh-i18n--envsubst

Adds language support for environment-based Git scripts. Used in advanced scripting.

For more information, visit **Ravi Kiran Maroju's LinkedIn**

Index Operations

git show-index

Lists contents of a packed archive index. Example: git show-index < packfile.idx

git show-ref

Lists all refs (branches, tags) in the repo. Example: git show-ref

For more information, visit **Ravi Kiran Maroju's LinkedIn profile**

File Management

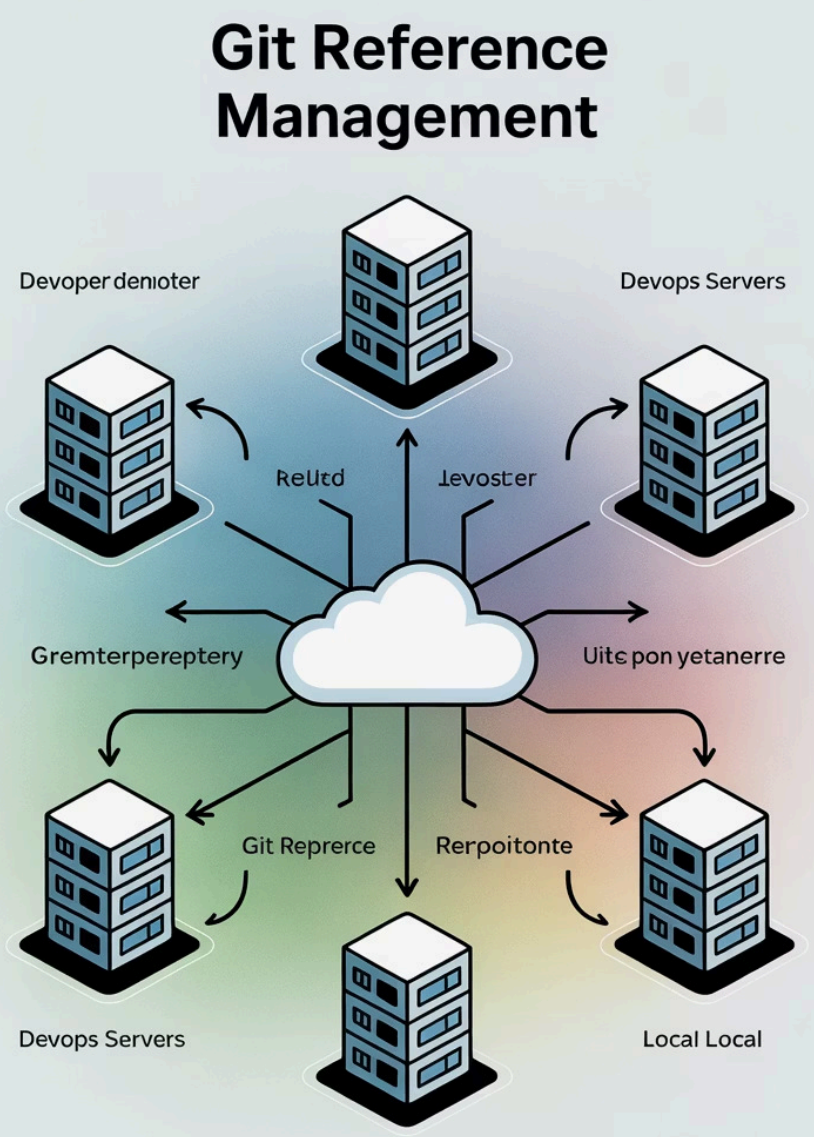
git stage

Adds file changes to staging area (like git add). Example: git stage file.txt

git stripspace

Removes extra blank lines or spaces. Example: git stripspace < input.txt

Connect with me: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**



Reference Management

git symbolic-ref

Manages symbolic references. Example: git symbolic-ref HEAD refs/heads/main

git tar-tree

Creates a .tar file from a tree in Git. Example: git tar-tree HEAD > code.tar

For more information: **Ravi Kiran Maroju**

File Extraction

git unpack-file

Writes a file's blob to a temporary file

git unpack-objects

Extracts files from a packed object archive

git unpack-file Writes a file's blob to a temporary file. Example: git unpack-file hash

git unpack-objects Extracts files from a packed object archive. Example: git unpack-objects < packfile.pack

Contact: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**

Index Updates

git update-index

Manually updates the staging index

2

git update-ref

Changes what a branch or ref points to

git update-index Manually updates the staging index. Example: git update-index --add file.txt

git update-ref Changes what a branch or ref points to. Example: git update-ref refs/heads/main

Learn more: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**

Network Operations

git upload-archive

Sends a Git archive over the network

git upload-pack

Sends Git objects to git fetch

git upload-archive Sends a Git archive over the network. Used on Git server side.

git upload-pack Sends Git objects to git fetch. Used during cloning/fetching.

For more information, contact: **Ravi Kiran Maroju**

Configuration Inspection

git var

Shows Git's internal config variables. Example: git var GIT_AUTHOR_IDENT

git verify-pack

Checks if a pack file is valid. Example: git verify-pack packfile.idx

https://www.linkedin.com/in/ravi-kiran-maroju-175225134/

Tree Operations

1

Write Tree

git write-tree Saves the current index as a tree object. Example: git write-tree

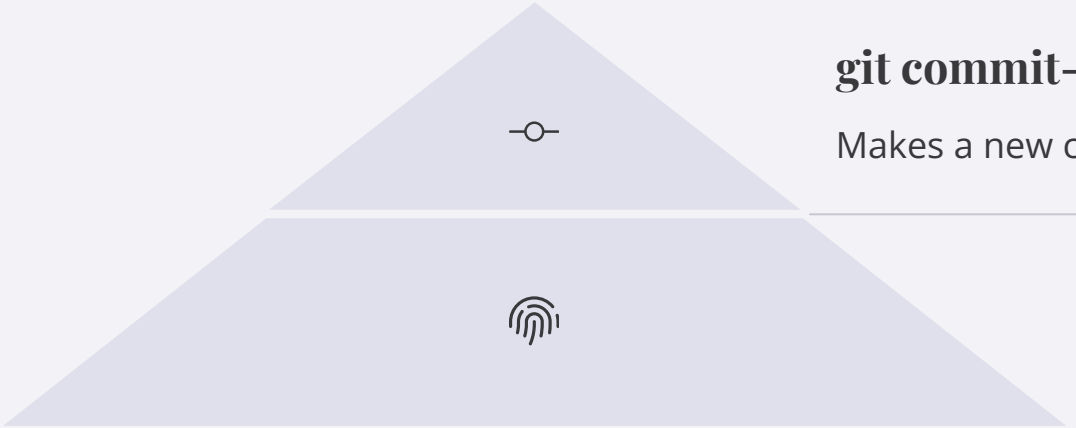
2

Checkout Index

git checkout-index Copies files from staging area to your working folder. Example: git checkout-index -a

Visit **Ravi Kiran Maroju** for more information.

Object Creation



git commit-tree

Makes a new commit object from a tree

git hash-object

Gets the SHA1 hash of a file and saves it in Git

git commit-tree Makes a new commit object from a tree. Example: git commit-tree -p HEAD -m "message"

git hash-object Gets the SHA1 hash of a file and saves it in Git. Example: git hash-object -w file.txt

For more information: **Ravi Kiran Maroju**

Tree Management

git mktree

Makes a tree object from directory listing text. Example: git mktree < file.txt

git pack-refs

Stores all refs into a single file for better performance. Example: git pack-refs --all

https://www.linkedin.com/in/ravi-kiran-maroju-175225134/

History Management

git revert

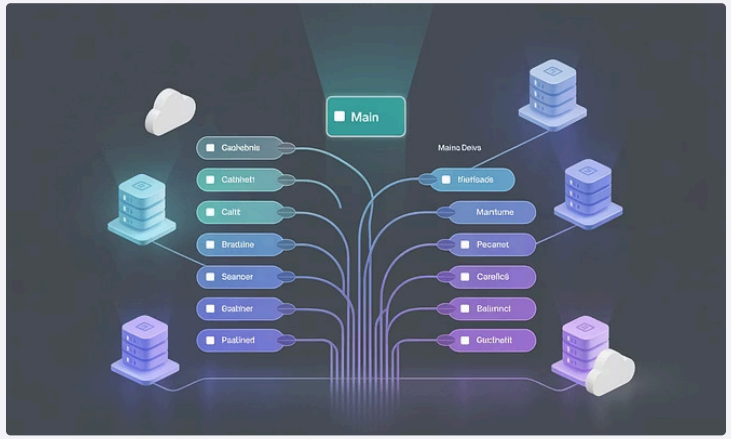
Undoes a previous commit by making a new one. Example: git revert abc123

git show

Displays info about commits, tags, or files. Example: git show HEAD

For more information: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**

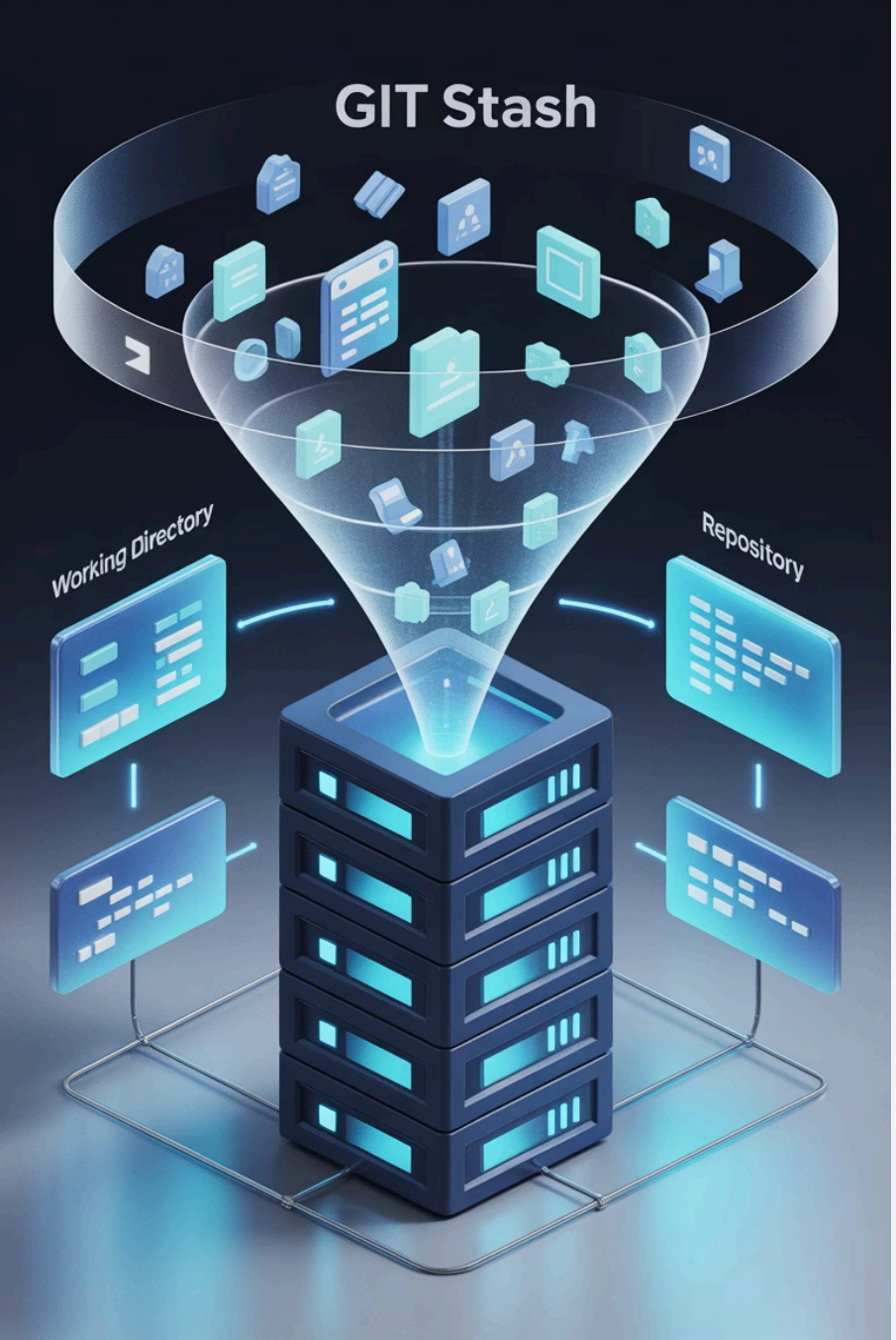
Branch Visualization



git show-branch Shows multiple branches and their commits. Example: git show-branch

git show-index Shows the index of a packed Git archive. Example: git show-index < pack.idx

Visit my profile: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**



Reference Management

git show-ref

Lists all reference names and commit hashes. Example: git show-ref

git stash

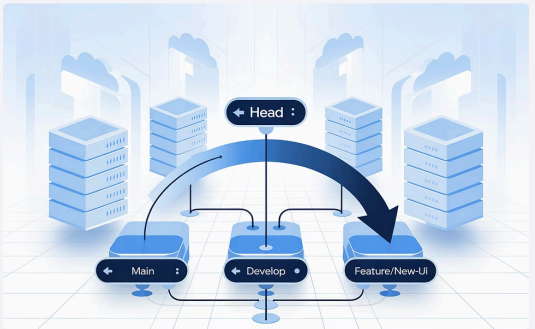
Temporarily saves your changes without committing. Example: git stash

Visit: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**

Repository Status

git status

Shows what has changed and what's staged. Example: git status



git switch

Switches to another branch. Example: git switch main

For more information: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**

Tagging and Unpacking

git tag

Adds or lists tags (like bookmarks) on commits. Example: git tag v1.0

git unpack-objects

Extracts files from a pack. Example: git unpack-objects < pack.pack

Contact: **LinkedIn Profile**

Index Management

I

git update-index

Manually adds or updates a file in the index

2

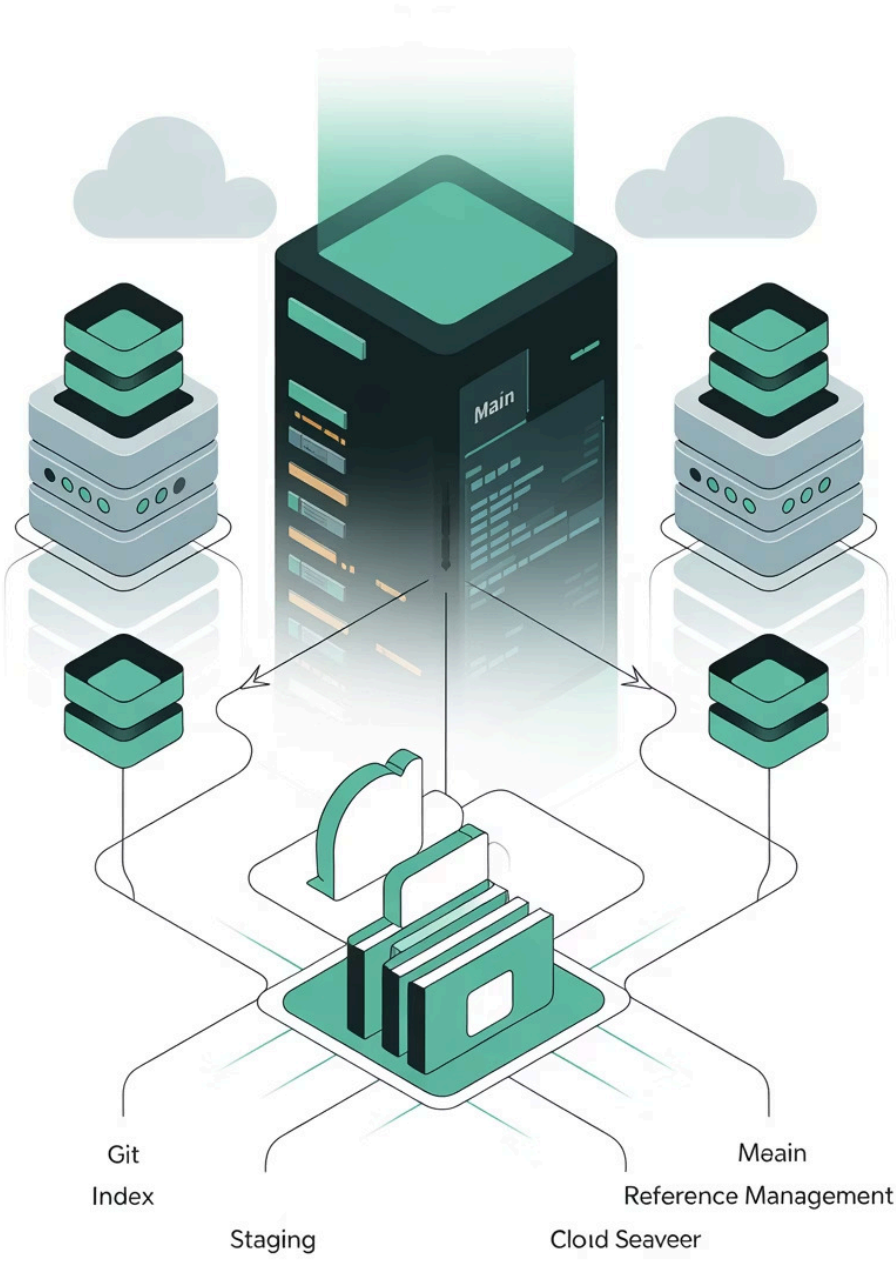
git update-ref

Manages where branch names point

git update-index Manually adds or updates a file in the index. Example: git update-index --add file.txt

git update-ref Manages where branch names point. Example: git update-ref refs/heads/dev abc123

Connect with me: **https://www.linkedin.com/in/ravi-kiran-maroju-175225134/**



Working with Multiple Branches

git worktree

Work with multiple branches in different folders. Example: git worktree add ../new-feature feature

git write-tree

Writes current staging area as a tree object. Example: git write-tree

https://www.linkedin.com/in/ravi-kiran-maroju-175225134/