

Git Branch Workflow Guide

A repeatable feature-branch workflow with review and recovery reminders.

A safe small-feature workflow

Step	Command	Purpose
1. Check workspace	git status	See changed and untracked files first
2. Create branch	git switch -c feat/short-name	Isolate the change
3. Edit and inspect	git diff	Review what changed
4. Stage deliberately	git add path/to/file	Stage only intended files
5. Review staged diff	git diff --staged	Catch secrets and unrelated edits
6. Commit	git commit -m "Add feature"	Save a useful checkpoint
7. Sync / publish	git push -u origin feat/short-name	Share the branch for review

Before merging

- [] Run tests, type checks, lint, and build appropriate for the project.
- [] Review the final diff and changed-file list.
- [] Resolve conflicts intentionally; do not blindly overwrite remote work.
- [] Confirm CI checks pass.
- [] Merge via your team's normal review process.

Recover carefully

Need	Safer starting point
Unstage file	git restore --staged path/to/file
Discard unstaged edits	Back up first; then git restore path/to/file
Reverse a published commit	git revert COMMIT_SHA
Unsure what is safe	git status; git diff; make a backup before destructive commands

Avoid git reset --hard and git clean -fd unless you understand exactly what they will delete.