



# **GIT COMMANDS CHEAT SHEET FOR DEVOPS**

**By Joshua**



# Git Commands Cheat Sheet for DevOps

## 1. Git Setup

Command	Use Case
<code>git config --global user.name "Your Name"</code>	Set your Git username globally.
<code>git config --global user.email "you@example.com"</code>	Set your Git email address globally.
<code>git config --list</code>	View all Git configurations.

## 2. Starting a Git Project

Command	Use Case
<code>git init</code>	Initialize a new Git repository.
<code>git clone &lt;repo-url&gt;</code>	Clone an existing repository.

## 3. Basic Snapshotting

Command	Use Case
<code>git status</code>	Check the current status of the repository.
<code>git add &lt;file&gt;</code>	Stage a file for commit.
<code>git add .</code>	Stage all changes in the directory.
<code>git commit -m "message"</code>	Commit staged changes with a message.
<code>git diff</code>	View unstaged changes.
<code>git diff --staged</code>	View changes staged for commit.

## 4. Branching

Command	Use Case
<code>git branch</code>	List all branches.
<code>git branch &lt;branch-name&gt;</code>	Create a new branch.
<code>git checkout &lt;branch-name&gt;</code>	Switch to another branch.
<code>git checkout -b &lt;branch-name&gt;</code>	Create and switch to a new branch.
<code>git branch -d &lt;branch-name&gt;</code>	Delete a branch safely.



# Git Commands Cheat Sheet for DevOps

## 5. Merging and Rebasing

Command	Use Case
<code>git merge &lt;branch-name&gt;</code>	Merge a branch into the current branch.
<code>git rebase &lt;branch-name&gt;</code>	Rebase the current branch onto another.

## 6. Working with Remote Repositories

Command	Use Case
<code>git remote add origin &lt;repo-url&gt;</code>	Link a remote repository.
<code>git remote -v</code>	Show remote repository URLs.
<code>git push origin &lt;branch-name&gt;</code>	Push a branch to remote.
<code>git pull</code>	Pull changes from remote.
<code>git fetch</code>	Fetch changes without merging.

## 7. Undoing Changes

Command	Use Case
<code>git checkout -- &lt;file&gt;</code>	Discard changes in a file.
<code>git reset HEAD &lt;file&gt;</code>	Unstage a file.
<code>git revert &lt;commit&gt;</code>	Create a new commit to undo a previous one.
<code>git reset --hard &lt;commit&gt;</code>	Reset to a previous commit (destructive).

## 8. Stashing Changes

Command	Use Case
<code>git stash</code>	Stash current changes temporarily.
<code>git stash pop</code>	Apply the stashed changes back.

## 9. Tagging

Command	Use Case
<code>git tag &lt;tag-name&gt;</code>	Create a tag.
<code>git tag</code>	List all tags.
<code>git push origin &lt;tag-name&gt;</code>	Push a tag to remote.



# Git Commands Cheat Sheet for DevOps

## 10. Viewing History

Command	Use Case
<code>git log</code>	View commit history.
<code>git log --oneline</code>	View compact commit history.
<code>git show &lt;commit&gt;</code>	View details of a specific commit.

## 11. Collaboration (Advanced)

Command	Use Case
<code>git cherry-pick &lt;commit&gt;</code>	Apply a specific commit onto current branch.
<code>git rebase -i &lt;commit&gt;</code>	Interactive rebase for squashing or editing commits.

## 12. Cleaning Up

Command	Use Case
<code>git clean -fd</code>	Remove untracked files and directories.

## 13. Git Aliases

Command	Use Case
<code>git config --global alias.co checkout</code>	Create alias <code>git co</code> for <code>git checkout</code> .
<code>git config --global alias.br branch</code>	Create alias <code>git br</code> for <code>git branch</code> .

## 14. Interactive Rebase and Commit Rewriting

Command	Use Case
<code>git rebase -i HEAD~n</code>	Interactive rebase last n commits (edit, squash, fixup).
<code>git commit --amend</code>	Modify the last commit (edit message or add files).
<code>git rebase -i &lt;commit&gt;</code>	Reorder, squash, edit old commits.

## 15. Advanced Branch Management



# Git Commands Cheat Sheet for DevOps

Command	Use Case
<code>git checkout -b &lt;new-branch&gt; origin/&lt;remote-branch&gt;</code>	Create and track a remote branch.
<code>git branch -m &lt;old&gt; &lt;new&gt;</code>	Rename a local branch.
<code>git push origin :&lt;branch&gt;</code>	Delete a remote branch.

## 16. Advanced Merging Strategies

Command	Use Case
<code>git merge --no-ff &lt;branch&gt;</code>	Force a merge commit (even if fast-forward is possible).
<code>git merge --squash &lt;branch&gt;</code>	Combine all branch commits into one and merge.
<code>git cherry-pick &lt;commit&gt;</code>	Apply a specific commit from another branch.

## 17. Submodules (Working with Nested Git Repos)

Command	Use Case
<code>git submodule add &lt;repo-url&gt; &lt;path&gt;</code>	Add a submodule inside a repository.
<code>git submodule update --init --recursive</code>	Initialize and update all submodules.
<code>git submodule foreach git pull</code>	Pull latest changes in all submodules.

## 18. Git Hooks (Automation in Git)

Command	Use Case
Create scripts inside <code>.git/hooks/</code>	Automate tasks (e.g., pre-commit, pre-push checks).
Example: <code>pre-commit</code> script	Run linting, tests before every commit automatically.

## 19. Git Reset (Advanced Usage)

Command	Use Case
<code>git reset --soft &lt;commit&gt;</code>	Move HEAD to commit but keep all changes staged.
<code>git reset --mixed &lt;commit&gt;</code>	Move HEAD to commit, unstage changes (default).
<code>git reset --hard &lt;commit&gt;</code>	Move HEAD and erase all changes (⚡ dangerous).

## 20. Git Clean (Advanced Workspace Cleanup)



# Git Commands Cheat Sheet for DevOps

Command	Use Case
<code>git clean -n</code>	Preview which untracked files would be removed.
<code>git clean -fd</code>	Forcefully remove untracked files and directories.
<code>git clean -fx</code>	Remove even ignored files (very destructive).

## 21. Git Reflog (Recover Deleted Commits)

Command	Use Case
<code>git reflog</code>	View the history of HEAD changes (good for recovery).
<code>git checkout &lt;reflog-hash&gt;</code>	Restore lost commits using reflog hash.

## 22. Git Archive (Export as Zip/Tar)

Command	Use Case
<code>git archive --format zip HEAD &gt; latest.zip</code>	Create a zip archive of the latest commit.

## 23. Shallow Clone (Faster Clone for Big Repos)

Command	Use Case
<code>git clone --depth=1 &lt;repo-url&gt;</code>	Clone only the latest snapshot, not full history.

### Quick Note:

- `rebase -i`, `cherry-pick`, `reflog`, `submodules` — these are used in real production systems, especially in **DevOps**, **CI/CD pipelines**, **multi-repo monorepos**, and **collaboration across teams**.