Commonly Used Git Commands for DevOps Engineers (With Examples)

As a **DevOps Engineer**, you interact with Git daily for source code management, CI/CD automation, and infrastructure as code (IaC). Below are the most commonly used Git commands categorized by their use cases.

1. Git Setup & Configuration

a) Set up your Git identity (for commits)

git config --global user.name "Your Name" git config --global user.email "your.email@example.com"

b) Check your Git configuration

git config --list

2. Working with Repositories

a) Clone a remote repository

git clone https://github.com/example/repo.git

b) Initialize a new Git repository

git init

c) Add a remote repository

git remote add origin https://github.com/example/repo.git

3. Branching & Merging

a) Create a new branch and switch to it

git checkout -b feature-branch

b) List all branches

git branch

c) Switch between branches

git checkout main

Or, in newer versions:

git switch main

d) Merge changes from another branch

git checkout main git merge feature-branch

e) Delete a branch (local and remote)

git branch -d feature-branch #

Local git push origin --delete feature-branch # Remote

4. Staging & Committing Changes

a) Check the current status of your repo

git status

b) Add files to staging area

git add file1.txt # Add a specific file git add . # Add all changes

c) Commit changes with a message

git commit -m "Updated configuration file"

d) Commit changes and amend the last commit

git commit --amend -m "Updated config with correct settings"

5. Working with Remote Repositories

a) Push changes to the remote repository

git push origin main

b) Fetch latest changes from remote

git fetch

c) Pull the latest changes (fetch + merge)

git pull origin main

6. Resetting & Reverting Changes

a) Undo changes in a specific file

git checkout -- file1.txt

Or in newer Git versions:

git restore file1.txt

b) Remove a file from staging area (keep

changes locally)

git reset HEAD file1.txt

c) Undo the last commit (keep changes)

git reset --soft HEAD~1

d) Revert a commit (create a new commit undoing it)

git revert <commit-hash>

7. Comparing & Reviewing Changes

a) Show the commit history

git log --oneline --graph --decorate --all

b) Compare changes between commits or branches

git diff main feature-branch

c) Show changes of a specific file

git diff HEAD -- file1.txt

d) Show which commit last modified a specific line in a file

git blame file1.txt

8. Working with Git Tags

a) Create a new tag

git tag v1.0.0

b) Push tags to the remote repository

git push origin --tags

c) Delete a tag (local and remote)

git tag -d v1.0.0 # Local git push origin --delete v1.0.0 # Remote

9. Stashing Changes (Temporary Save)

a) Stash uncommitted changes

git stash

b) List all stashes

git stash list

c) Apply the last stashed changes

git stash pop

d) Apply a specific stash

git stash apply stash@{1}

e) Drop a stash (delete it)

git stash drop stash@{1}

10. Rebasing &

Squashing Commits

a) Rebase a feature branch onto main

git checkout feature-branch git rebase main

b) Squash multiple commits into one

git rebase -i HEAD~3
(Replace pick with squash (s) for unnecessary commits)

11. Git Hooks (Automation in DevOps)

a) Create a pre-commit

hook to prevent commits with TODO comments

echo '#!/bin/sh if grep -r "TODO" .; then echo "Commit rejected: Found TODO comments" exit 1 fi' > .git/hooks/pre-commit chmod +x .git/hooks/pre-commit

12. Git in CI/CD Pipelines

a) Pull the latest changes before deployment

git pull origin main

b) Clone a repo inside a CI/CD pipeline

git clone https://github.com/example/repo.git cd repo

c) Use GitHub Actions to trigger a build on push

name: CI/CD Pipeline on: [push] jobs: build: runs-on: ubuntu-latest steps: - uses: actions/checkout@v2 - run: echo "Build process starts"

13. Git Security Best Practices

a) Remove sensitive data

from commit history

git filter-branch --force --index-filter \ 'git rm --cached --ignore-unmatch secrets.txt' \ --prune-empty --tag-name-filter cat -- --all

b) Use SSH instead of HTTPS for authentication

git remote set-url origin git@github.com:user/repo.git

Final Thoughts

These commands are **essential for DevOps Engineers** in day-to-day work, helping with:

✅ Code versioning

✅ Automation & CI/CD

✅ Infrastructure as Code (IaC)
✅ Debugging and rollback
✅ Security and compliance

Would you like a deeper dive into any of these commands?