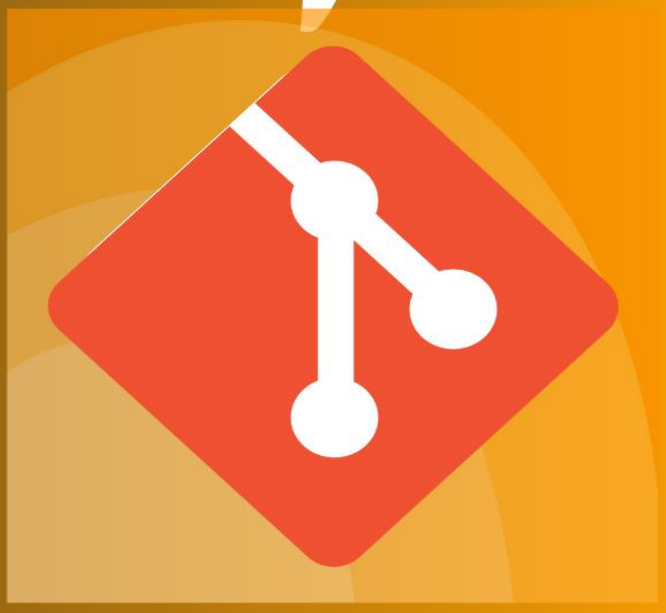
Search www.devopsshack.com

Devops Shack

GIT CHEAT SHEET



215K+ SUBSCRIBERS

46K+ FOLLOWERS

@adityajaiswal7
@devopsshack

CREATED BY

DevOps Shack

For Live & Recorded Trainings on DevOps Visit devopsshack.com

DevOps Shack

GIT CHEAT SHEET

1. Git Setup & Configuration

Set User Info (Local & Global)

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
git config --local user.name "Project-Specific Name"
git config --local user.email "project.email@example.com"
```

Set Default Editor

```
git config --global core.editor "vim"
```

Enable Colored Output

```
git config --global color.ui auto
```

Set Default Merge Tool

```
git config --global merge.tool vimdiff
```

Enable Credential Caching

git config --global credential.helper cache

Configure Git to Cache Credentials for 1 Hour

git config --global credential.helper 'cache --timeout=3600'

2. Git Repository Setup

Initialize a New Repository

git init

Clone a Repository with Submodules

git clone --recursive <repository-url>

Reinitialize an Existing Repository

git init --bare

3. Working with Files & Staging

Check Which Files Are Tracked/Untracked

git ls-files

List Ignored Files

git status --ignored

Remove Files from Git Tracking but Keep in Directory

git rm --cached <file>

Unstage a File

```
git restore --staged <file>
```

Rename a File and Track the Change

```
git mv old-file.txt new-file.txt
```

4. Committing & Amendments

Commit with a Description

```
git commit -m "Initial commit"
```

Amend the Last Commit

```
git commit --amend -m "Updated commit message"
```

Sign Off a Commit (For Compliance)

```
git commit -s -m "Signed commit"
```

5. Branching & Merging

Rename the Current Branch

```
git branch -m new-branch-name
```

List All Remote & Local Branches

```
git branch -a
```

ow Tracking Branch Information

```
git branch -vv
```

Find the Branch That Last Contained a Commit

git branch --contains <commit-ha>

Merge a Branch (Fast-Forward)

git merge --ff-only <branch-name>

Merge with No Fast-Forward

git merge --no-ff <branch-name>

Abort a Merge

git merge --abort

6. Rebasing

Start Interactive Rebase

git rebase -i HEAD~3

Abort a Rebase

git rebase --abort

Continue a Rebase After Resolving Conflicts

git rebase --continue

7. Staing (Temporary Storage)

Sta With a Message

git sta pu -m "WIP: Feature update"

List Staes

git sta list

ow Sta Details

git sta ow -p sta@{0}

Drop a Specific Sta

git sta drop sta@{0}

8. Working with Remote Repositories

Change the Remote URL

git remote set-url origin <new-url>

Fetch Changes Without Merging

git fetch origin

Fetch and Prune Remote-Deleted Branches

git fetch --prune

Pu Changes with Force (Careful!)

git pu --force

Pu Changes Only If No Conflicts

git pu --force-with-lease

9. Git Log & History

View a Compact Log

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

View a Log With Stats

```
git log --stat
```

Filter Log by Author

```
git log --author="John Doe"
```

Find a Commit by Keyword

```
git log --grep="fix bug"
```

ow Last N Commits

```
git log -n 5
```

10. Reverting & Resetting

Undo Last Commit (Keep Changes Staged)

```
git reset --soft HEAD~1
```

Undo Last Commit (Unstage Changes)

```
git reset --mixed HEAD~1
```

Undo Last Commit (Discard Changes Completely)

```
git reset --hard HEAD~1
```

Revert a Specific Commit

git revert <commit-ha>

11. Working with Tags

List Tags in Reverse Order

git tag --sort=-v:refname

Checkout a Tag

git checkout tags/<tag-name>

Create & Pu a Lightweight Tag

git tag <tag-name>

git pu origin <tag-name>

Create & Pu an Annotated Tag

git tag -a <tag-name> -m "Version release"

git pu origin <tag-name>

12. Git Aliases (Custom ortcuts)

Set Up Aliases

git config --global alias.co checkout

git config --global alias.cm "commit -m"

git config --global alias.st status

```
git config --global alias.br branch
```

13. Finding & Fixing Bugs

Find a Bug Using Bisect

```
git bisect start
```

```
git bisect bad
```

```
git bisect good <commit-ha>
```

Find and Fix Whitespace Issues

```
git diff --check
```

14. Handling Large Files

Check Large Files in Repo

```
git rev-list --objects --all | sort -k2 | cut -f2- -d' ' | uniq -c | sort -nr | head -10
```

Remove Large Files from History

```
git filter-branch --tree-filter 'rm -rf largefile.zip' HEAD
```

15. Git Hooks (Automation)

Common Hooks

- pre-commit: Runs before committing.
- pre-pu: Runs before puing.
- commit-msg: Runs to validate commit messages.

Example Pre-Commit Hook

```
#!/bin/

echo "Running pre-commit checks..."

exit 0
```

Save it in .git/hooks/pre-commit and make it executable:

```
chmod +x .git/hooks/pre-commit
```

16. Git Security & Best Practices

Enable GPG Signing for Commits

```
git config --global commit.gpgsign true

git config --global user.signingkey <GPG-key-ID>
```

Avoid Storing Secrets in Git

Use .gitignore:

```
echo ".env" >> .gitignore
```