Git Commands Guide

A Step-by-Step Illustrated Guide with Examples

1. Cloning a Repository

- To clone a repository from GitHub, use:
- git clone <repository_url>
- Example:
- git clone https://github.com/abinrajmk8/myproject.git
- This creates a local copy of the repository on your system.

2. Checking Status

- To check the current status of your repository, use:
- git status
- This shows which files have been modified, added, or are untracked.

3. Ignoring Files (.gitignore)

- To ignore files like node_modules/ and .env, create a `.gitignore` file and add:

- *.env
- node_modules/

- This prevents these files from being tracked in the repository.

4. Adding and Committing Changes

- To stage and commit changes at once:

- git add .
- git commit -m "Your commit message"

- The `git add .` stages all changes, and `git commit -m` commits them with a message.

5. Pushing Changes to GitHub

- To push your committed changes to GitHub:
- git push origin main
- This uploads your local changes to the remote repository.

6. Pulling Updates from Remote Repository

- If changes are made in the remote repository, pull them to your local system:

- git pull origin main

- This ensures your local repository stays updated with the latest changes.

7. Stashing Uncommitted Changes

- If you need to switch branches but have uncommitted changes, use:
- git stash
- This temporarily saves changes. Retrieve them later with:
- git stash pop

8. Summary

- • Clone the repository: `git clone`
- • Check status: `git status`
- • Ignore files: `.gitignore`
- • Add & commit: `git add .` & `git commit`
- • Push changes: `git push`
- • Pull updates: `git pull`
- • Stash changes: `git stash`

- Mastering these commands ensures smooth