Essential Git and GitHub Commands: A Step-by-Step Guide

Abdelrahman wael

December 16, 2025

1 Setup and Configuration

These commands are usually run once per computer to set up your identity.

```
# Set your name (appears in your commits)
git config --global user.name "Your Name"

# Set your email (appears in your commits)
git config --global user.email "youremail@example.com"

# Check your configuration
git config --list
```

Listing 1: Initial Configuration

2 Starting a New Repository (Project)

2.1 1. Initialize a Local Repository

Create a new Git repository in your current project directory.

```
# Navigate to your project folder
cd /path/to/my/project

# Initialize the repository
git init -b main
```

Listing 2: Initializing a New Repository

2.2 2. Check Status

See which files are untracked, modified, or staged.

```
git status
```

Listing 3: Checking Repository Status

2.3 3. Add Files to Staging Area

Tell Git you want to track changes in specific files (staging).

```
# Stage a specific file
git add filename.txt

# Stage all changes in the current directory (be cautious with this!)
git add .
```

Listing 4: Staging Changes

2.4 4. Commit Changes

Save the staged changes permanently in the local repository's history with a descriptive message.

```
git commit -m "Brief but descriptive commit message"
```

Listing 5: Committing Staged Changes

3 Connecting to GitHub

These steps link your local repository to a remote repository on GitHub.

3.1 1. Create a Remote Repository

You must first create an empty repository on GitHub.

3.2 2. Link Local to Remote

Tell your local Git repository the URL of the remote GitHub repository.

```
# Replace <URL> with the HTTPS or SSH URL from GitHub
git remote add origin <URL>

# Verify the remote connection
git remote -v
```

Listing 6: Adding a Remote Origin

3.3 3. Push to GitHub

Send your committed local changes up to the remote repository for the first time. The '-u' flag sets the remote branch as the default "upstream."

```
git push -u origin main
```

Listing 7: First Push to GitHub

4 Daily Workflow (Pull, Edit, Commit, Push)

4.1 1. Pull Latest Changes

Before starting work, get any new changes from the remote repository.

```
git pull origin main
```

Listing 8: Pulling Changes

4.2 2. Work and Commit

Make changes to your files, then stage and commit them.

```
# Edit files...
git add .
git commit -m "Implemented feature X"
```

Listing 9: Staging and Committing

4.3 3. Push Changes

Send your new commits to the remote GitHub repository.

```
git push origin main
```

Listing 10: Pushing Changes

5 Branching (Feature Development)

Branching allows you to work on new features or fixes independently of the main code.

5.1 1. Create and Switch Branch

```
# Create a new branch named 'feature/my-new-thing'
git branch feature/my-new-thing

# Switch to that branch
git checkout feature/my-new-thing

# Shortcut to create and switch in one command
git checkout -b feature/my-new-thing
```

Listing 11: Creating and Switching Branches

5.2 2. List Branches

```
git branch
```

Listing 12: Listing Branches

5.3 3. Merge Branches

After development on your branch is complete, switch back to the main branch and merge.

```
# Switch back to the main branch
git checkout main

# Merge the feature branch into main
git merge feature/my-new-thing
```

Listing 13: Merging a Branch

5.4 4. Delete Branch

```
# Delete the local branch
git branch -d feature/my-new-thing

# Delete the remote branch (if you pushed it)
git push origin --delete feature/my-new-thing
```

Listing 14: Deleting a Branch

6 Cloning an Existing Repository

If a project already exists on GitHub, clone it to your local machine.

```
# Replace <URL> with the GitHub repository URL
git clone <URL>
```

Listing 15: Cloning an Existing Repository

7 Viewing History

See a chronological list of all commits.

```
# Standard log
git log

# A cleaner, one-line graphical view
git log --all --decorate --oneline --graph
```

Listing 16: Viewing Commit History