

# Git Configuration - Complete Notes

## 1. Git Configuration Overview

`git config` is used to set or retrieve Git configuration values. These settings control Git's behavior and can be applied at different levels.

Level	Scope	File Location
system	System-wide (all users)	/etc/gitconfig (Linux) C:\ProgramData\Git\config (Windows)
global	Current user (all repos)	~/.gitconfig (Linux) C:\Users\<username>\.gitconfig (Windows)
local	Per repository only	.git/config (inside each project)

## 2. Setting User Information

Examples:

```
$ git config --system user.name "Ladram Verma"
```

```
$ git config user.name "Nubeera Imran"
```

```
$ git config --global user.name "Huma Dil de chuke sanam 2103"
```

Priority Order:

1. --local
2. --global
3. --system

Syntax:

```
$ git config [<level>] <key> <value>
```

- key = what to configure (e.g., user.name)

- value = value to assign (e.g., your name)

## 3. Setting Default Editor

```
$ git config core.editor "nano"
```

- Opens merge commit messages in nano.

```
$ git config core.editor "code --wait"
```

- Opens in VS Code and waits until the editor is closed.

# Git Configuration - Complete Notes

## 4. Aliases for Commands

```
$ git config --global alias.co checkout
```

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

```
$ git config --global alias.st status
```

Usage:

```
- `git co` = `git checkout`
```

```
- `git br` = `git branch`
```

```
- `git st` = `git status`
```

## 5. Credential Management

Git requires authentication (username/password or token) for remote operations.

Temporarily Store in Memory:

```
$ git config --global credential.helper cache
```

- Stores in memory for 15 mins

Store on Disk (Not recommended):

```
$ git config --global credential.helper store
```

- Stored unencrypted in ~/.git-credentials

Example:

```
https://<username>:<token>@github.com
```

Recommended (Windows/Mac only):

```
$ git config --global credential.helper manager-core
```

- Uses Windows Credential Manager or macOS Keychain