

Hướng Dẫn Sử Dụng Neovim Config

Leader Key

- **Leader key:** `<Space>` (phím cách)
 - Tất cả các shortcut chính đều bắt đầu bằng `<Space>`
-

Tổng Quan Plugins

UI & Appearance

- **tokyonight.nvim** - Colorscheme (theme màu đẹp)
- **alpha-nvim** - Dashboard chào mừng khi mở Neovim
- **nvim-web-devicons** - Icons cho files và folders
- **indent-blankline.nvim** - Hiển thị đường kẻ indent
- **which-key.nvim** - Hiển thị gợi ý keybindings
- **bufferline.nvim** - Tab line đẹp cho buffers/tabs
- **lualine.nvim** - Status line đẹp với thông tin hữu ích
- **dressing.nvim** - UI cải thiện cho input và select

File Management

- **nvim-tree.lua** - File explorer (cây thư mục)
- **telescope.nvim** - Fuzzy finder (tìm files, text, symbols)
- **auto-session** - Tự động lưu và restore workspace

Development Tools

- **nvim-lspconfig** - Cấu hình Language Server Protocol
- **mason.nvim** - Quản lý LSP servers, formatters, linters
- **nvim-cmp** - Autocompletion (gợi ý code)
- **nvim-treesitter** - Syntax highlighting và parsing

Code Editing

- **nvim-autopairs** - Tự động đóng ngoặc, quotes
- **Comment.nvim** - Comment/uncomment code
- **nvim-surround** - Thao tác với surrounding characters
- **substitute.nvim** - Thay thế text nâng cao
- **vim-maximizer** - Maximize/minimize splits

Code Quality

- **conform.nvim** - Code formatting
- **nvim-lint** - Code linting (kiểm tra lỗi)
- **trouble.nvim** - UI đẹp cho diagnostics
- **todo-comments.nvim** - Highlight và navigate TODO comments

Git Integration

- **gitsigns.nvim** - Git decorations và actions trong buffer
- **lazygit.nvim** - Integration với LazyGit TUI

Utilities

- **plenary.nvim** - Lua functions library (dependency cho nhiều plugins)
- **vim-tmux-navigator** - Navigate giữa Neovim và tmux panes
- **LuaSnip** - Snippet engine
- **friendly-snippets** - Collection of useful snippets

File Explorer (nvim-tree)

Mục đích: File explorer như VSCode sidebar, giúp navigate và quản lý files/folders

Keybindings:

- **<Space>ee** - Toggle file explorer
- **<Space>ef** - Toggle file explorer và focus vào file hiện tại
- **<Space>ec** - Collapse tất cả folders
- **<Space>er** - Refresh file explorer

Trong nvim-tree:

- **Enter** hoặc **o** - Mở file/folder
- **a** - Tạo file/folder mới
- **d** - Xóa file/folder
- **r** - Rename file/folder
- **x** - Cut file/folder
- **c** - Copy file/folder
- **p** - Paste file/folder
- **y** - Copy filename
- **Y** - Copy relative path
- **gy** - Copy absolute path
- **R** - Refresh
- **H** - Toggle hidden files
- **q** - Đóng nvim-tree

Telescope (Fuzzy Finder)

Mục đích: Tìm kiếm siêu nhanh files, text, symbols, git commits, v.v. Giống Ctrl+P trong VSCode nhưng mạnh hơn

Keybindings:

- **<Space>ff** - Find files (tìm file)
- **<Space>fr** - Find recent files (file gần đây)

- `<Space>fs` - Live grep (tìm text trong files)
- `<Space>fc` - Find string under cursor (tìm từ dưới con trỏ)
- `<Space>ft` - Find todos (tìm TODO comments)

Trong Telescope:

- `<C-j>` / `<C-k>` - Di chuyển lên/xuống
- `<C-q>` - Gửi kết quả vào quickfix list
- `<C-u>` / `<C-d>` - Scroll preview
- `<Tab>` - Toggle selection
- `<Enter>` - Mở file
- `<C-x>` - Mở file trong split horizontal
- `<C-v>` - Mở file trong split vertical
- `<C-t>` - Mở file trong tab mới
- `<Esc>` - Đóng Telescope

Alpha Dashboard

Mục đích: Màn hình chào đẹp khi mở Neovim, có shortcuts nhanh đến các tác vụ thường dùng

Menu shortcuts:

- `e` - Tạo file mới
- `<Space>ee` - Toggle file explorer
- `<Space>ff` - Find file
- `<Space>fs` - Find word
- `<Space>wr` - Restore session
- `q` - Quit Neovim

Session Management (auto-session)

Mục đích: Tự động lưu và khôi phục workspace (tabs, splits, file positions) giống VSCode workspaces

Keybindings:

- `<Space>wr` - Restore session cho thư mục hiện tại
- `<Space>ws` - Save session cho thư mục hiện tại

Cách hoạt động:

- Session tự động save khi thoát Neovim
- Restore session khi mở Neovim trong cùng thư mục
- Session chỉ save cho project directories (không save cho ~/Downloads, ~/Desktop, etc.)

LSP (Language Server Protocol)

Mục đích: Cung cấp IntelliSense, go-to-definition, error checking, auto-complete như VSCode

Keybindings:

- **gd** - Go to definition
- **gD** - Go to declaration
- **gi** - Go to implementation
- **gt** - Go to type definition
- **gR** - Show references
- **K** - Hover documentation
- **<Space>ca** - Code actions
- **<Space>rn** - Rename symbol
- **<Space>d** - Show line diagnostics
- **<Space>D** - Show buffer diagnostics
- **[d** - Previous diagnostic
- **]d** - Next diagnostic
- **<Space>rs** - Restart LSP

Supported Languages:

- TypeScript/JavaScript (ts_ls)
- HTML (html)
- CSS (cssls)
- Tailwind CSS (tailwindcss)
- Svelte (svelte)
- Lua (lua_ls)
- GraphQL (graphql)
- Emmet (emmet_ls)
- Prisma (prismals)
- Python (pyright)



Autocompletion (nvim-cmp)

Mục đích: Autocomplete thông minh kết hợp LSP, snippets, buffer text. Giống IntelliSense

Trong Insert Mode:

- **<C-k>** - Previous suggestion
- **<C-j>** - Next suggestion
- **<C-Space>** - Trigger completion
- **<C-e>** - Close completion menu
- **<Enter>** - Confirm selection
- **<C-b> / <C-f>** - Scroll docs
- **<Tab>** - Navigate snippets

Sources:

- LSP suggestions

- Buffer text
 - File paths
 - Snippets (LuaSnip + friendly-snippets)
-

Code Actions & Editing

Auto Pairs (nvim-autopairs):

Mục đích: Tự động đóng ngoặc, quotes khi gõ. VD: gõ (sẽ tự thêm)

- Tự động đóng (), [], {}, "", ''
- Hỗ trợ treesitter context

Comments (Comment.nvim):

Mục đích: Comment/uncomment code nhanh, hỗ trợ nhiều ngôn ngữ

- gcc - Toggle comment line
- gc + motion - Comment với motion (vd: gcap comment paragraph)
- gc trong Visual mode - Comment selection
- Hỗ trợ context-aware comments cho JSX/TSX

Surround (nvim-surround):

Mục đích: Thao tác với ký tự bao quanh (quotes, brackets, tags). VD: đổi "hello" thành 'hello'

- ys + motion + char - Add surrounding (vd: ysiw" bao từ bằng quotes)
- ds + char - Delete surrounding (vd: ds" xóa quotes)
- cs + old + new - Change surrounding (vd: cs"' đổi quotes)
- Trong Visual mode: S + char - Surround selection

Substitute (substitute.nvim):

Mục đích: Thay thế text nâng cao với motion. VD: copy text rồi thay thế nhanh

- s + motion - Substitute với motion
 - ss - Substitute line
 - S - Substitute to end of line
 - Trong Visual mode: s - Substitute selection
-

Navigation & TODO

TODO Comments (todo-comments.nvim):

Mục đích: Highlight và navigate các TODO, FIXME, NOTE trong code. Giúp track công việc cần làm

-]t - Next TODO comment
- [t - Previous TODO comment
- <Space>ft - Find all TODOs với Telescope

Supported TODO types:

- **TODO:** - Cần làm
 - **HACK:** - Code hack
 - **WARN:** - Cảnh báo
 - **PERF:** - Performance
 - **NOTE:** - Ghi chú
 - **TEST:** - Test
 - **FIX:** - Bug fix
-

Diagnostics & Trouble

Mục đích: UI đẹp để xem errors, warnings, TODOs. Giống Problems panel trong VSCode

Keybindings:

- **<Space>xw** - Workspace diagnostics
 - **<Space>xd** - Document diagnostics
 - **<Space>xq** - Quickfix list
 - **<Space>xl** - Location list
 - **<Space>xt** - TODOs trong trouble
-

Formatting & Linting

Formatting (conform.nvim):

Mục đích: Tự động format code đẹp (giống Prettier trong VSCode). Auto format khi save

- **<Space>mp** - Format file hoặc selection
- Auto format on save (có thể tắt/bật)

Supported Formatters:

- **JavaScript/TypeScript:** Prettier
- **HTML/CSS/JSON:** Prettier
- **Lua:** Stylua
- **Python:** isort + black

Linting (nvim-lint):

Mục đích: Kiểm tra lỗi code, code style (giống ESLint trong VSCode)

- **<Space>l** - Trigger linting manually
- Auto lint on: BufEnter, BufWritePost, InsertLeave

Supported Linters:

- **JavaScript/TypeScript:** ESLint
- **Python:** Pylint

Window & Tab Management

Window Splits:

- `<Space>sv` - Split vertical
- `<Space>sh` - Split horizontal
- `<Space>se` - Make splits equal size
- `<Space>sx` - Close current split
- `<Space>sm` - Maximize/minimize current split (vim-maximizer)

Tabs:

- `<Space>to` - Open new tab
- `<Space>tx` - Close current tab
- `<Space>tn` - Next tab
- `<Space>tp` - Previous tab
- `<Space>tf` - Open current buffer in new tab

General:

- `<C-h/j/k/l>` - Navigate between splits (vim-tmux-navigator)
- `j k` trong Insert mode - Exit to Normal mode

Git Integration

Gitsigns (gitsigns.nvim):

Mục đích: Hiển thị git changes trong sign column, staging/unstaging hunks trực tiếp trong buffer

Navigation:

- `]h` - Next git hunk
- `[h` - Previous git hunk

Actions:

- `<Space>hs` - Stage hunk
- `<Space>hr` - Reset hunk
- `<Space>hS` - Stage entire buffer
- `<Space>hR` - Reset entire buffer
- `<Space>hu` - Undo stage hunk
- `<Space>hp` - Preview hunk
- `<Space>hb` - Blame line (full)
- `<Space>hB` - Toggle line blame
- `<Space>hd` - Diff this
- `<Space>hD` - Diff this ~

Visual Mode:

- **<Space>hs** - Stage selected hunk
- **<Space>hr** - Reset selected hunk

LazyGit (lazygit.nvim):

Mục đích: Integration với LazyGit - TUI git client mạnh mẽ cho quản lý git

Keybindings:

- **<Space>lg** - Open LazyGit

Trong LazyGit:

- **q** - Quit LazyGit
- **Space** - Stage/unstage
- **c** - Commit
- **P** - Push
- **p** - Pull
- **Tab** - Switch panels
- **Enter** - Select/open
- **?** - Help để xem tất cả shortcuts

UI Enhancements

Bufferline (bufferline.nvim):

Mục đích: Tab line đẹp hiển thị buffers/tabs với icons và styling

Features:

- Hiển thị tabs với slant separators
- Icons cho từng file type
- Tab mode thay vì buffer mode
- Click để switch tabs

Lualine (lualine.nvim):

Mục đích: Status line đẹp với thông tin file, git, LSP, mode, etc.

Thông tin hiển thị:

- Current mode (Normal, Insert, Visual, etc.)
- File name và path
- Git branch và changes
- LSP diagnostics (errors, warnings)
- File encoding và format
- Cursor position

- Lazy.nvim pending updates

Dressing (dressing.nvim):

Mục đích: Cải thiện UI cho vim.ui.select và vim.ui.input với Telescope

Features:

- Input prompts đẹp hơn
- Select menus với Telescope
- Consistent UI experience

Utility Commands

Which-key (which-key.nvim):

Mục đích: Hiển thị gợi ý keybindings khi bạn nhấn leader key. Giúp nhớ shortcuts

- Nhấn **<Space>** và đợi 500ms để xem available keybindings

Mason (mason.nvim):

Mục đích: Package manager cho LSP servers, formatters, linters. Giống Extension Marketplace

- **:Mason** - Mở Mason UI để quản lý LSP servers, formatters, linters
- **:MasonUpdate** - Update tất cả tools
- **:MasonInstall <tool>** - Cài đặt tool cụ thể

Lazy (Plugin Manager):

Mục đích: Quản lý tất cả plugins của Neovim. Lazy loading để khởi động nhanh

- **:Lazy** - Mở Lazy UI
- **:Lazy update** - Update plugins
- **:Lazy sync** - Sync plugins
- **:Lazy clean** - Clean unused plugins

Treesitter (nvim-treesitter):

Mục đích: Parser code thông minh cho syntax highlighting, text objects, indentation

- **:TSUpdate** - Update parsers
- **:TSInstall <language>** - Cài parser cho ngôn ngữ
- **<C-Space>** - Incremental selection (expand)
- **<Backspace>** - Decremental selection (shrink)

Appearance

Colorscheme:

- **Tokyo Night** với custom colors - Theme màu đẹp, dễ nhìn
- Dark theme với blue accents

UI Enhancements:

- **File icons** (nvim-web-devicons) - Icons đẹp cho từng loại file
- **Indent guides** (indent-blankline) - Đường kẻ giúp thấy rõ indentation
- **Relative line numbers** - Số dòng relative giúp navigate nhanh
- **Sign column** - Cột hiển thị git changes và diagnostics
- **Cursor line highlighting** - Highlight dòng hiện tại
- **Bufferline** - Tab line đẹp với slant separators
- **Lualine** - Status line với theme tùy chỉnh
- **Git decorations** - Gitsigns hiển thị changes trong sign column

Tips & Tricks

1. **Tìm file nhanh:** `<Space>ff` rồi gõ tên file
2. **Tìm text:** `<Space>fs` rồi gõ text cần tìm
3. **Navigate code:** Dùng LSP keybindings (`gd`, `gr`, etc.)
4. **Project session:** `<Space>ws` để save, `<Space>wr` để restore
5. **Code actions:** `<Space>ca` để xem available actions
6. **Quick format:** `<Space>mp` để format code
7. **See TODOs:** `<Space>ft` để xem tất cả TODOs trong project
8. **File tree:** `<Space>ee` để toggle, navigate bằng `h j k l`
9. **Git workflow:** `<Space>lg` mở LazyGit để quản lý git
10. **Stage hunks:** `<Space>hs` để stage git changes từng hunk
11. **Maximize split:** `<Space>sm` để maximize split hiện tại
12. **Navigate git changes:** `]h` và `[h` để jump giữa các hunks
13. **Quick blame:** `<Space>hb` để xem ai sửa dòng này
14. **Preview changes:** `<Space>hp` để preview git hunk

Getting Started

1. Mở Neovim trong project directory
2. `:Lazy` để check plugins
3. `:Mason` để check LSP servers
4. `<Space>ee` để mở file explorer
5. `<Space>ff` để tìm files
6. `<Space>lg` để mở LazyGit cho git management
7. Bắt đầu coding! 🚀

First Time Setup:

1. **Cài LazyGit:** `brew install lazygit` (macOS)
2. **Check health:** `:checkhealth` để verify dependencies
3. **Install parsers:** `:TSUpdate` để cài Treesitter parsers

4. Install LSP servers: `:Mason` rồi install servers cần thiết

Troubleshooting

Plugin không hoạt động:

1. `:Lazy reload <plugin-name>`
2. Restart Neovim
3. `:checkhealth` để check dependencies

LSP không hoạt động:

1. `:LspInfo` để check LSP status
2. `:Mason` để check server installation
3. `:LspRestart` để restart LSP

Performance issues:

1. `:Lazy profile` để check startup time
2. Disable unused plugins
3. `:checkhealth` để check health