Neovim Configuration - Complete Keybindings & Plugin Guide

Table of Contents

Basic Settings

Your Neovim is configured with modern defaults: - **Leader Key**: `Space` () - **Line Numbers**: Enabled with relative numbers - **Tab Size**: 4 spaces - **Auto-save**: Enabled on file changes - **Clipboard**: Synchronized with system clipboard - **Undo**: Persistent undo history

---

Core Keybindings

Window Navigation

Keybinding	Description	Mode
`Ctrl+h`	Move to left window	Normal
`Ctrl+j`	Move to lower window	Normal
`Ctrl+k`	Move to upper window	Normal
`Ctrl+l`	Move to right window	Normal

Window Resizing

Keybinding	Description	Mode
`Ctrl+Up`	Increase window height	Normal
`Ctrl+Down`	Decrease window height	Normal
`Ctrl+Left`	Decrease window width	Normal
`Ctrl+Right`	Increase window width	Normal

Buffer Management

Keybinding	Description	Mode
`Shift+l`	Next buffer	Normal
`Shift+h`	Previous buffer	Normal
`<leader>bd`	Delete buffer	Normal
`<leader>bD`	Force delete buffer	Normal
`<leader>ba`	Delete all buffers except current	Normal
`<leader>bw`	Wipeout buffer	Normal

Quick Save/Quit

Keybinding	Description	Mode
`<leader>w`	Save file	Normal
`<leader>q`	Quit	Normal
`<leader>x`	Save and quit	Normal

Escape Shortcuts

Keybinding	Description	Mode
`jk`	Escape to normal mode	Insert
`jj`	Escape to normal mode (alternative)	Insert

File Management

Oil.nvim - Modern File Explorer

Plugin: `stevearc/oil.nvim`
Purpose: Edit your filesystem like a normal Neovim buffer

Keybinding	Description	Mode
`-`	Open parent directory	Normal
`<leader>-`	Open parent directory in float	Normal

Inside Oil Buffer

Keybinding	Description
`Enter`	Open file/directory
`Ctrl+s`	Open in vertical split
`Ctrl+h`	Open in horizontal split
`Ctrl+t`	Open in new tab
`Ctrl+p`	Preview file
`Ctrl+c`	Close oil buffer
`Ctrl+l`	Refresh view
`g.`	Toggle hidden files
`gs`	Change sort order
`gx`	Open with system default

Text Navigation & Editing

Movement Enhancements

Keybinding	Description	Mode
`j`	Move down (visual line)	Normal/Visual
`k`	Move up (visual line)	Normal/Visual

Text Manipulation

Keybinding	Description	Mode
`<`	Indent left (stay in visual mode)	Visual
`>`	Indent right (stay in visual mode)	Visual
`p`	Paste without yanking selection	Visual
`J`	Move selected text down	Visual
`K`	Move selected text up	Visual
`Alt+j`	Move selected text down	Visual
`Alt+k`	Move selected text up	Visual

Harpoon - Quick File Navigation

Plugin: `ThePrimeagen/harpoon`
Purpose: Quickly jump between frequently used files

Keybinding	Description	Mode
`<leader>a`	Add file to harpoon	Normal
`<leader>e`	Open harpoon quick menu	Normal
`<leader>1`	Jump to harpoon file 1	Normal
`<leader>2`	Jump to harpoon file 2	Normal
`<leader>3`	Jump to harpoon file 3	Normal
`<leader>4`	Jump to harpoon file 4	Normal
`Ctrl+Shift+P`	Previous harpoon file	Normal
`Ctrl+Shift+N`	Next harpoon file	Normal

Flash.nvim - Enhanced Motion

Plugin: `folke/flash.nvim`
Purpose: Jump to any location on screen with minimal keystrokes

Keybinding	Description	Mode
`s`	Flash jump	Normal/Visual/Operator
`S`	Flash treesitter	Normal/Visual/Operator
`r`	Remote flash	Operator
`R`	Treesitter search	Visual/Operator
`Ctrl+s`	Toggle flash search	Command

Enhanced Marks

Plugin: `chentoast/marks.nvim`
Purpose: Better mark visualization and management

Keybinding	Description	Mode
`m,`	Toggle marks signs	Normal
`m;`	List marks in buffer	Normal
`m:`	List global marks	Normal
`m[`	Previous mark	Normal
`m]`	Next mark	Normal
`m<space>`	Delete all marks in buffer	Normal

Text Objects & Surrounding

Plugin: `echasnovski/mini.surround`
Purpose: Enhanced surrounding operations

Keybinding	Description	Mode
`gsa`	Add surrounding	Normal
`gsd`	Delete surrounding	Normal
`gsr`	Replace surrounding	Normal
`gsf`	Find surrounding	Normal
`gsF`	Find surrounding (left)	Normal
`gsh`	Highlight surrounding	Normal
`gsn`	Update n_lines	Normal

Code Splitting/Joining

Plugin: `Wansmer/treesj`
Purpose: Smart split/join of code blocks

Keybinding	Description	Mode
`<leader>m`	Toggle split/join	Normal
`<leader>j`	Join code block	Normal
`<leader>s`	Split code block	Normal

Multiple Cursors

Plugin: `mg979/vim-visual-multi`
Purpose: Multiple cursor editing

Keybinding	Description	Mode
`Ctrl+d`	Find and select next occurrence	Normal
`Alt+Ctrl+Down`	Add cursor down	Normal
`Alt+Ctrl+Up`	Add cursor up	Normal

Search & Telescope

Plugin: `nvim-telescope/telescope.nvim`
Purpose: Fuzzy finder and search interface

File Finding

Keybinding	Description	Mode
`<leader>pf`	Find files	Normal
`Ctrl+p`	Git files	Normal
`<leader>pr`	Recent files	Normal
`<leader>pb`	Open buffers	Normal

Text Searching

Keybinding	Description	Mode
`<leader>pg`	Live grep	Normal
`<leader>ps`	Grep string (input)	Normal

Documentation & Help

Keybinding	Description	Mode
`<leader>vh`	Help tags	Normal

Diagnostics

Keybinding	Description	Mode
`<leader>pd`	All diagnostics	Normal
`<leader>pe`	Error diagnostics only	Normal
`<leader>pw`	Warning diagnostics only	Normal

LSP Integration

Keybinding	Description	Mode
`<leader>lr`	LSP references	Normal
`<leader>ld`	LSP definitions	Normal
`<leader>li`	LSP implementations	Normal
`<leader>ls`	Document symbols	Normal
`<leader>lS`	Workspace symbols	Normal

Inside Telescope

Keybinding	Description	Mode
`Ctrl+k`	Move selection up	Insert
`Ctrl+j`	Move selection down	Insert
`Ctrl+q`	Send to quickfix list	Insert/Normal

Git Integration

GitSigns - Git Gutter

Plugin: `lewis6991/gitsigns.nvim`
Purpose: Git signs in the gutter and hunk operations

Hunk Navigation

Keybinding	Description	Mode
`]h`	Next hunk	Normal
`[h`	Previous hunk	Normal

Hunk Operations

Keybinding	Description	Mode
`<leader>hs`	Stage hunk	Normal/Visual
`<leader>hr`	Reset hunk	Normal/Visual
`<leader>hS`	Stage buffer	Normal
`<leader>hR`	Reset buffer	Normal
`<leader>hu`	Undo stage hunk	Normal
`<leader>hp`	Preview hunk	Normal

Blame & Diff

Keybinding	Description	Mode
`<leader>hb`	Blame line (full)	Normal
`<leader>hB`	Toggle line blame	Normal
`<leader>hd`	Diff this	Normal
`<leader>hD`	Diff this (~)	Normal

Text Objects

Keybinding	Description	Mode
`ih`	Git hunk text object	Operator/Visual

Neogit - Full Git Interface

Plugin: `NeogitOrg/neogit`
Purpose: Magit-like git interface for Neovim

Keybinding	Description	Mode
`<leader>gg`	Open Neogit	Normal
`<leader>gc`	Git commit	Normal
`<leader>gp`	Git pull	Normal

Keybinding	Description	Mode
`<leader>gP`	Git push	Normal

Diffview - Enhanced Diff Viewing

Plugin: `sindrets/diffview.nvim`
Purpose: Enhanced diff viewing capabilities

Keybinding	Description	Mode
`<leader>gd`	Open Diffview	Normal
`<leader>gh`	File history	Normal
`<leader>gq`	Close Diffview	Normal

LSP & Code Intelligence

Plugin: `neovim/nvim-lspconfig`
Purpose: Language Server Protocol integration

Navigation

Keybinding	Description	Mode
`gd`	Go to definition	Normal
`gD`	Go to declaration	Normal
`K`	Hover documentation	Normal

Diagnostics

Keybinding	Description	Mode
`<leader>vd`	Open diagnostic float	Normal
`[d`	Next diagnostic	Normal
`]d`	Previous diagnostic	Normal

Code Actions

Keybinding	Description	Mode
`<leader>vca`	Code action	Normal
`<leader>vrn`	Rename symbol	Normal
`<leader>vrr`	References	Normal
`<leader>vws`	Workspace symbol	Normal

Signature Help

Keybinding	Description	Mode
`Ctrl+h`	Signature help	Insert

Auto-completion

Plugin: `hrsh7th/nvim-cmp`
Purpose: Intelligent code completion

Keybinding	Description	Mode
`Tab`	Next completion item / Expand snippet	Insert
`Shift+Tab`	Previous completion item	Insert
`Ctrl+Space`	Trigger completion	Insert
`Ctrl+b`	Scroll docs up	Insert
`Ctrl+f`	Scroll docs down	Insert
`Ctrl+e`	Abort completion	Insert
`Enter`	Confirm selection	Insert

Debugging (DAP)

Plugin: `mfussenegger/nvim-dap`
Purpose: Debug Adapter Protocol integration

Debug Controls

Keybinding	Description	Mode
`<leader>db`	Toggle breakpoint	Normal
`<leader>dc`	Continue debugging	Normal
`<leader>ds`	Step over	Normal
`<leader>di`	Step into	Normal
`<leader>do`	Step out	Normal
`<leader>dr`	Open REPL	Normal
`<leader>dl`	Run last	Normal
`<leader>dt`	Toggle DAP UI	Normal

Terminal Integration

Plugin: `akinsho/toggleterm.nvim`
Purpose: Enhanced terminal integration

Basic Terminals

Keybinding	Description	Mode
`Ctrl+\`	Toggle default terminal	Normal/Insert
`<leader>tf`	Float terminal	Normal
`<leader>th`	Horizontal terminal	Normal
`<leader>tv`	Vertical terminal	Normal

Language-Specific Terminals

Keybinding	Description	Mode
`<leader>tR`	Rust terminal (cargo run)	Normal
`<leader>tP`	Python terminal	Normal
`<leader>tN`	Node.js terminal	Normal
`<leader>ip`	IPython terminal	Normal

Advanced Tools

Keybinding	Description	Mode
`<leader>lg`	Lazygit	Normal
`<leader>ht`	Htop	Normal

Terminal Mode Navigation

Keybinding	Description	Mode
`Esc`	Exit terminal mode	Terminal
`jk`	Exit terminal mode	Terminal
`Ctrl+h`	Move to left window	Terminal
`Ctrl+j`	Move to lower window	Terminal
`Ctrl+k`	Move to upper window	Terminal
`Ctrl+l`	Move to right window	Terminal

Workflow & Productivity

Comments

Plugin: `numToStr/Comment.nvim`
Purpose: Smart commenting - Uses default mappings (`gcc`, `gc` in visual mode)

TODO Comments

Plugin: `folke/todo-comments.nvim`
Purpose: Highlight and navigate TODO comments

Navigation

Keybinding	Description	Mode
`]t`	Next todo comment	Normal
`[t`	Previous todo comment	Normal

Search

Keybinding	Description	Mode
`<leader>st`	Search todos (Telescope)	Normal
`<leader>sT`	Search todos/fixes (Telescope)	Normal
`<leader>xt`	Todos in Trouble	Normal

Keybinding	Description	Mode
`<leader>xT`	Todos/fixes in Trouble	Normal

Search & Replace

Keybinding	Description	Mode
`<leader>sr`	Search & replace word under cursor	Normal
`<leader>sr`	Search & replace in selection	Visual

Quick Fix Lists

Keybinding	Description	Mode
`<leader>co`	Open quickfix	Normal
`<leader>cc`	Close quickfix	Normal
`[q`	Previous quickfix	Normal
`]q`	Next quickfix	Normal

Utility Toggles

Keybinding	Description	Mode
`<leader>h`	Clear search highlights	Normal
`<leader>tw`	Toggle word wrap	Normal
`<leader>ts`	Toggle spell check	Normal
`<leader>tn`	Toggle line numbers	Normal
`<leader>tr`	Toggle relative numbers	Normal

Plugin-Specific Features

WhichKey

Plugin: `folke/which-key.nvim`
Purpose: Key binding discovery

Keybinding	Description	Mode
`<leader>?`	Show buffer keymaps	Normal

Key Groups (When pressing leader)

- `<leader>f` - File operations
- `<leader>p` - Project/Find operations
- `<leader>g` - Git operations
- `<leader>d` - Debug operations
- `<leader>t` - Terminal operations
- `<leader>v` - LSP operations
- `<leader>w` - Window operations
- `<leader>b` - Buffer operations

Auto-save

Plugin: `okuuva/auto-save.nvim`
Purpose: Automatic file saving - Saves automatically on buffer leave, focus lost, and text changes - No manual keybindings needed

Theme & UI Enhancements

- Modern color scheme with proper highlighting
- Enhanced statusline and UI elements
- File icons and syntax highlighting
- Smooth scrolling and animations

Supported Languages

Your configuration includes language servers for: - **C/C++** (clangd) - **Rust** (rust-analyzer) - **Python** (pyright) - **JavaScript/TypeScript** (ts_ls) - **Lua** (lua_ls)

Debug adapters are configured for: - **Python** (debugpy) - **Rust/C/C++** (codelldb) - **JavaScript/Node.js** (js-debug-adapter)

Tips & Best Practices

1. **Use Harpoon** for quick file switching between your most important files
2. **Telescope** is your main search interface - use it for files, text, and LSP features
3. **Flash.nvim** makes movement incredibly fast - use `s` followed by characters to jump anywhere
4. **Oil.nvim** lets you edit directories like files - very powerful for file management
5. **GitSigns** provides all git operations you need without leaving Neovim
6. **Which-key** will help you discover keybindings - just wait after pressing `<leader>`
7. **Auto-save** keeps your work safe automatically
8. Use **marks** (both built-in and enhanced) for quick navigation within and between files

---

Customization

This configuration is highly modular. Each plugin is configured in its own file under `~/.config/nvim/lua/plugins/`. You can easily: - Modify keybindings in individual plugin files - Add new plugins by creating new files in the plugins directory - Adjust settings in `init.lua` for global behavior

Remember to restart Neovim or use `:source %` after making changes to see them take effect.