To set up an LSP (Language Server Protocol) in NvChad, you use a tool called **Mason**. Think of Mason as an "App Store" for your editor—it handles the downloading and installing of the actual language engines (like Pyright for Python) so you don't have to do it manually on your system.
Follow these steps to get Python fully powered up:

---

1. Install the Language Server (The "Engine")

1. Open Neovim.
2. Type :Mason and press Enter. This opens the Mason UI.
3. Search for **pyright** (you can type /pyright to find it).
4. Press i to install it.
5. (Optional) While you're there, search for **ruff** (a very fast linter) and install that too.

---

2. Tell NvChad to Use the Engine

Installing it via Mason puts the files on your computer, but now you have to "wire it up" so Neovim knows to turn it on when you open a .py file.

1. Create (or open) the file lua/configs/lspconfig.lua.
2. Add the following code to "enable" Pyright:

Lua

```
local configs = require("nvchad.configs.lspconfig")

local on_attach = configs.on_attach
local on_init = configs.on_init
local capabilities = configs.capabilities

local lspconfig = require "lspconfig"
local servers = { "html", "cssls", "pyright" } -- Add pyright here

for _, lsp in ipairs(servers) do
  lspconfig[lsp].setup {
    on_attach = on_attach,
    on_init = on_init,
    capabilities = capabilities,
  }
end
```

3. Now, you must tell NvChad to load this file. Open lua/plugins/init.lua and find the nvim-lspconfig block. Change it to look like this:

Lua

```
{
 "neovim/nvim-lspconfig",
 config = function()
  require("nvchad.configs.lspconfig").defaults()
  require "configs.lspconfig" -- This loads your custom file above
 end,
},
```

---

3. The "Final Boss": MasonInstallAll

NvChad has a shortcut to make sure everything in your config is actually installed.

- Run the command: :MasonInstallAll

---

4. How to Test It

1. Open any Python file (e.g., test.py).
2. Type some code, like import os and then os. on the next line.
3. **Auto-completion:** A menu should pop up with suggestions.
4. **Error Checking:** Try writing something wrong (like print(x) without defining x). You should see a red sign or underline indicating an error.
5. **LSP Info:** Type :LspInfo to see a window confirming that pyright is "Active" in the current buffer.

What's happening behind the scenes?

- **Mason:** Downloads the pyright binary.
- **lspconfig:** Tells Neovim *how* to talk to that binary.
- **nvim-cmp:** (Built into NvChad) Takes the information from the LSP and shows it to you as that nice completion menu.

Would you like me to show you how to set up **auto-formatting** so your Python code automatically follows the "Black" or "Ruff" style every time you save?