

Great call. Handling hidden folders like .git via the terminal is much faster and ensures you don't accidentally miss it.

Here is how to safely remove the old history and start your own.

1. Delete the existing .git folder

First, you need to navigate to your Neovim configuration directory. For most users (Linux/macOS), this is located in ~/.config/nvim.

Run these commands in your terminal:

Bash

```
# Navigate to your Neovim folder  
cd ~/.config/nvim
```

```
# Remove the .git folder and all its contents  
rm -rf .git
```

What do those flags mean?

- **rm**: The "remove" command.
 - **-r** (Recursive): Tells the computer to go inside the folder and delete everything inside it first.
 - **-f** (Force): Tells the computer "Don't ask me for permission for every single file, just do it."
-

2. Initialize your own history

Now that the "blank slate" is ready, you can tell Git to start tracking **your** specific changes from this moment forward.

Run these commands in the same folder:

Bash

```
# Start a brand new Git history  
git init
```

```
# Add all current NvChad files to your new history  
git add .
```

```
# Save this as your "starting point"
git commit -m "Initial commit: My custom NvChad setup"
```

Why is this helpful?

By doing this, you've created a **safety net**. If you spend three hours tweaking your `chadrc.lua` file and accidentally break your editor, you can now use Git to "roll back" to this exact moment.

Quick Tip for Windows Users

If you are on Windows using PowerShell, the location and command are slightly different:

PowerShell

```
cd $HOME\AppData\Local\nvim
Remove-Item -Recurse -Force .git
```

Would you like me to explain how to connect this new local history to a GitHub account so your settings are backed up in the cloud?