

The Second-Brain Setup

Turn a folder of docs into a connected knowledge graph your AI can actually navigate — Graphify + Obsidian + Claude Code, in 3 steps.

You commented SECOND BRAIN, so here's the full thing. The exact 3-step method from the carousel, with the real commands, so you can build it today.

■ The thing that's actually broken

You dump a folder of docs on Claude Code and expect it to just know your stuff. It doesn't. It reads the files one at a time, greps around, and misses how the ideas connect. Page 12 and page 90 are about the same concept — Claude has no idea, because to it they're just two separate files.

That's the whole problem. Your AI isn't dumb. It just has no map. It has a pile of pages and no sense of what links to what.

A second brain fixes that. Instead of a folder of files, you give your AI a map of *ideas* — every concept pulled out, every connection drawn, the whole thing browsable. Then when it needs "everything about context windows," it follows the links instead of re-reading 145 files.

Here's how to build one.

■ The taste — what you end up with

I pointed this at the Claude Code docs as a test. One folder, 145 documents. Here's what came out the other side:

- **145 docs** → **591 ideas** (nodes)
- **591 ideas** → **685 connections** (links between them)
- **685 connections** → **67 topic groups** (clusters of related ideas)

(Those are the numbers from that one run — an example of what the tools produce, not a client result. Your folder gives you your own numbers.)

So 145 flat files became 591 connected notes, grouped into 67 topics, each note linked to every related note. That's a searchable second brain your AI walks through — not a folder it greps.

Three tools do it:

- **Graphify** — reads your docs and turns them into a map of connected ideas.
- **Obsidian** — a local notebook (a "vault") where every idea becomes a note your AI can browse.
- **Claude Code** — the AI that runs the whole flow and then uses the result.

■ Step 1 — point Graphify at your docs and map it

Graphify is a free, open-source knowledge-graph tool for AI coding assistants. It reads a folder, pulls out the actual ideas inside every file (one page can hold five), and draws the connections between them.

Install it (it's on PyPI as `graphifyy`; the command stays `graphify`):

```
pip install graphifyy
```

Repo, if you want to look first: `github.com/safishamsi/graphify`
(canonical org: `Graphify-Labs/graphify`). It's a real project — around

58k GitHub stars and 1.2M PyPI downloads — built on Tree-sitter, NetworkX, and Leiden clustering. It runs inside Claude Code, Codex, Cursor, Gemini CLI, OpenCode, Aider, and more.

Map your folder. Either just tell Claude Code in plain English:

```
Map the docs in ./my-docs with Graphify.
```

or run the CLI yourself:

```
graphify ./my-docs
```

(Inside an assistant you can also run the `/graphify` command.) It reads every file, extracts the ideas, and finds how they connect. When it's done you get a `graphify-out/` folder with:

- `graph.html` — the interactive map you can open in a browser
- `GRAPH_REPORT.md` — the plain-English summary (your node/connection/topic counts)
- `graph.json` — the raw graph data
- `cache/` — so re-runs are fast

Open `graph.html` and you'll see it: every dot is one idea, every line is a connection. Bigger dot = more important (more things link to it). This is already useful on its own — but it's stuck inside Graphify's own output. Step 2 sets it free.

■ Step 2 — turn every idea into a linked note, and open it as a vault

One flag does the whole thing. It takes all those ideas and writes each one out as its own note, auto-linked to every related note — in the exact format Obsidian reads:

```
graphify ./my-docs --obsidian
```

That emits an Obsidian vault: 591 markdown notes (in my run), each one already wired to the notes it connects to.

Now open it as a vault:

1. Install Obsidian (free) from `obsidian.md`. 2. Open Obsidian → **Open folder as vault** → pick the vault Graphify just wrote. 3. Hit the **graph view** (the little graph icon in the left rail).

There it is — your docs, as a living map you can click through. Every note links to its neighbors. Your AI can now follow those links instead of re-reading files.

■ Step 3 — wire it back to the source, and move it into your real vault

Here's the catch nobody mentions: right now that map is a beautiful bubble. It's a fresh, separate vault full of *summaries* of your ideas — but it's cut off from your actual source docs and from the rest of your notes. A second brain that's an island isn't a second brain. It's a demo.

Two small moves fix it:

1. Link every note back to its source doc. Each generated note is a distilled idea — you want a line at the top pointing to the real file it came from, so you (and Claude) can jump from the idea to the full context. Just ask Claude Code:

```
For every note in this vault, add a link at the top back to the
source
document in ./my-docs that the idea came from.
```

2. Move it into your main vault. Don't leave it as a standalone island. Drop the generated notes into the Obsidian vault you already live in — your real notes, projects, everything else. The moment they share a vault, Obsidian's graph connects them: your new idea-map fuses with your existing knowledge instead of floating next to it.

Now it's connected, not a bubble. Point Claude Code at that vault and it has a real second brain — every idea, every connection, every source, one search away.

■ Where this fits in the bigger picture

This isn't a one-off trick. It's the front door to how you feed *any* AI a body of knowledge.

- **Upstream:** whatever you're pointing it at — product docs, a codebase, your own research, a client's whole knowledge base. Graphify doesn't care what the folder is.
- **This step:** turning that raw pile into a connected map, so the AI reasons over *ideas and links*, not files.

- **Downstream:** every task you then run — Claude answering from your docs, writing with your context, keeping decisions consistent — gets better because it's walking a map instead of guessing across files.

Do this once and every AI task after it inherits a brain that's actually organized. That's the point: you're not organizing notes for yourself, you're building the thing your AI thinks *with*.

■ Your next step

Don't map your whole life today. Pick one folder — the docs for one project, or one tool you keep re-explaining to Claude — and run the 3 steps on just that:

```
pip install graphifyy
graphify ./that-one-folder --obsidian
```

Open it as a vault, click the graph view, and watch a flat folder turn into a map. Ten minutes. Once you've seen it work on one folder, you'll want it on everything — and now you know exactly how.

The cost of not doing it: your AI keeps re-reading the same files, missing the same connections, on every single task. This fixes it at the source, once.

Built by Axionox. I build real AI systems for a living — this second-brain setup is one small piece of how I keep an AI's knowledge connected instead of dumping raw files at it. If you want to see what this looks like wired into a real workflow, reply to the DM and I'll walk you through it. Follow for more like this.