SECOND BRAIN · BUILD EXPLAINER

Local Semantic Search for Your Second Brain

A local, offline, zero-cost layer that lets Claude recall your notes by meaning instead of keywords. Built over several days; finished, fixed, and verified today.

Date: June 15, 2026 **Status:** ● Live, re-indexing nightly **Cost:** $0 (runs on your Mac)

TL;DR

- **What it does:** Claude now finds vault notes by **meaning**, not exact words. Ask "cost to serve target" and it surfaces Ross's OKR that says "$0.22" even if your phrasing never appears in the note.
- **How:** a small AI model (BGE-M3) runs locally on your Mac through Ollama, turns every note into a "meaning fingerprint," and matches your question against all of them in about 0.3 seconds. Nothing leaves the machine.
- **Done today:** I verified the whole build is real, caught and fixed a flaw that would have silently disabled it, and made semantic search Claude's default way to recall from your vault.
- **Will it lag your Mac?** No. Right now it uses **19 MB** of memory and holds **zero** models loaded when idle. Your Mac has 36 GB. You will not feel it.
- **Pending:** one optional "insurance" model (Qwen3-14B) that we deliberately did **not** install. Skip it unless you want an offline reasoning backup.

1. What it is, in one line

A search engine for your own brain. Your second brain holds roughly 840 notes (meetings, people, projects, decisions). Until now, when Claude needed to recall something, it scanned a catalog of note *descriptions* and grep'd for exact keywords. This new layer lets Claude search by **what a note means**, so the right note surfaces even when your question uses completely different words than the note does.

2. Why we built it

It started with a YouTube video on Claude memory systems. The useful frame from it: any memory system has three jobs.

- **Storage** — saving a fact. Claude was already fine here.
- **Injection** — auto-loading context at the start of a session. Also fine.
- **Recall** — finding old info later. **This was the weak link.**

The old recall path matched **words**. If a note about support costs said "cost to serve" but you asked about "support efficiency target," a keyword search could miss it entirely. Semantic search matches **meaning**, so related ideas connect even when the wording differs. That was the gap worth closing, and it is the one piece that makes recall feel like the system actually knows what is in your vault.

3. What we built (in plain terms)

Five small pieces, all running locally on your Mac. No cloud, no subscription, no data leaving the machine.

THE ENGINE

Ollama

A free program that runs AI models locally on your Mac. Always on in the background, nearly weightless when idle.

THE MODEL

BGE-M3

The "embedding" model. It reads text and turns it into a list of numbers that represent its meaning. 1.2 GB on disk.

THE INDEXER

THE SEARCH

vault_embed.py

Walks your whole vault nightly and fingerprints every note. Incremental, so it only re-does notes that actually changed (about 2 seconds).

vault_semantic_search.py

Takes your question, fingerprints it the same way, and returns the closest-matching notes in about 0.3 seconds, each with its file path.

THE INDEX

Vector store

The searchable file of all those fingerprints (about 4,378 chunks), kept in a hidden folder inside your vault so Obsidian ignores it.

THE SCHEDULE

Nightly cron

Folded into your existing 3:45 AM catalog job. No new automation. If the engine is ever down, it skips quietly and search falls back to grep.

WHY LOCAL, NOT CLOUD

Your vault is personal, so privacy matters. A local model means it is free, works offline, and nothing ever leaves your Mac. Claude stays the reasoner; this model only does the meaning-matching for retrieval. Anthropic has no embeddings endpoint, so "use my Claude subscription for this" was never an option anyway.

4. What I changed and fixed today

The build came over in a handoff doc from a Slack session. The Slack bot could not finish the last step (it is blocked from editing your core config), so I took over, verified everything, and closed it out.

- **Verified it is real.** Ran live queries. "Cost to serve target" surfaced Ross's OKR with "$0.22." "Pam self-approving refunds" surfaced the CST policy file. Matches by meaning, about 0.3 seconds each.
- **Caught a flaw the handoff missed.** Claude's command sandbox blocks local connections, so the search would have silently failed and fallen back to grep *every single time*. The feature would have looked installed but never actually run. I confirmed the fix and proved that the obvious workaround (allowlisting it) does not work.

- **Made it the default.** Updated your core config so semantic search is now Claude's first recall step, with the old catalog and grep as an automatic fallback. This was the one step the Slack bot could not do.
- **Cleaned up.** Archived (moved, not deleted) four leftover files from an abandoned May prototype that required sending notes to an outside API.

5. What it accomplishes

Before

Claude scanned a catalog of note descriptions and grep'd for keywords. If your words did not match the note's words, the note was missed. Recall depended on guessing the right search term.

After

Claude searches by meaning across the full text of every note. The right note surfaces even with different wording, ranked by relevance, with the source cited. The catalog and grep remain as a safety net.

BENEFIT	WHAT IT MEANS FOR YOU
Better recall	Finds the right note by meaning, not just matching keywords.
Private	100% local and offline. Nothing leaves your Mac. $0 forever.
Resilient	If the engine is ever down, it silently falls back to the old catalog + grep. It never breaks an answer.
Self-maintaining	Re-indexes itself every night, only re-fingerprinting notes that changed.

6. Will this lag my Mac?

No. Here is the honest, measured answer, taken from your machine right now.

19 MB
Memory used by the engine

0
Models held in memory when idle

36 GB
Total memory on your Mac

~0.3s
Time for one search

1.2 GB
Model size on disk (not in

right now (idle)

memory)

Ollama runs quietly in the background, but it holds **nothing** in memory when you are not searching. When Claude runs a search, the model loads, answers in about a third of a second, and unloads itself after a few idle minutes. Against 36 GB of memory, the footprint is a rounding error. You will not notice it during normal work.

BOTTOM LINE

What is installed and running today (the BGE-M3 search model) is tiny and idle-friendly. It does not lag your Mac. The thing you may be worried about, a heavier model, is the optional item below that we did **not** install.

7. Pending and optional

There is exactly one open item, and it is optional.

OPTIONAL: QWEN3-14B (NOT INSTALLED)

A local *reasoning* model, separate from the search model. It is "insurance" so you could still get AI answers if the cloud ever went down. We deliberately did not install it. If you ever did, it would use roughly 8 to 9 GB of memory **only while actively answering a question**, then unload. On your 36 GB Mac that is comfortable, but it is a break-glass backup, not a replacement for Claude. **Recommendation: skip it for now.** You can add it anytime with one command, and it changes nothing until you ask for it.

To be completely clear on the lag question: nothing that touches the heavier model is installed, so the impact on your computer today is **zero**. Everything described above is the lightweight search layer only.