The Harvey Outreach Setup

How to install the open-source Harvey sales agent, train it on your own offer, and run a safe first cold email batch on your Claude Code subscription.

Why I built this

Most cold outreach tools charge you twice. You pay for the software, then you pay again per API call every time it reads a website or drafts an email. Harvey skips the second bill entirely.

Harvey is an open-source sales agent that runs its thinking through the Claude Code or Codex CLI, the same subscription you already pay for. It finds local businesses, reads their actual websites, writes cold email that references something real it found there, and holds every message for your approval before anything sends.

The one-line promise: a full outbound pipeline with zero per-message cost, and nothing goes out without you looking at it first.

01

How it actually works

1. Discover finds local businesses matching the industry and location you set, using free providers by default or paid ones like DataForSEO and Serper if you add them.
2. Profile visits each business's real website and pulls out specific, honest details: services listed, recent updates, the tone of the site, anything an email can reference without faking familiarity.
3. Write and Gate hand off to each other: the Writer agent drafts a three-email sequence using frameworks stored in skills/, then gate.py rejects anything with unrendered template tags, banned phrases, length violations, or a non-deliverable address before a human ever sees it.

02

The one thing almost everyone gets wrong

People install Harvey, run harvey train, and flip require_approval to false before they've read a single draft.

> **WATCH OUT**
>
> The default in harvey.yaml is channels.email.require_approval: true, and it should stay true for your first several batches. Read every email in harvey outbox before approving it. The Writer agent is good, not psychic, and the only way to catch a wrong detail about a prospect's business is to actually read what it wrote before it goes to their inbox.

---

03 The files

SOURCE CODE

The full repo:

https://github.com/ethanplusai/harvey

HARVEY.YAML (KEY SETTINGS)

```
channels:
  email:
    require_approval: true

usage:
  max_daily_claude_percent: <your limit>

heartbeat_interval_minutes: <your interval>
quiet_hours:
  start: "<e.g. 20:00>"
  end: "<e.g. 08:00>"
max_daily_sends: <your limit>
```

SKILLS/ (EDITABLE MARKDOWN)

```
email_frameworks.md     # AIDA, PAS, BAB, QVC, 3Ps, with compliance rules
objection_handling.md   # LAARC loop responses for replies
signal_playbook.md      # turns website observations into email openers
product_knowledge.md    # generated by `harvey train <your-url>`
```

Set it up in Claude Code or Codex

1. Confirm you have Python 3.11 or newer installed, and an active Claude Pro/Max or ChatGPT subscription with the matching CLI already logged in on your machine.
2. Clone the repo: git clone https://github.com/ethanplusai/harvey.git, then cd harvey.
3. If you're running this as a Claude Code skill rather than standalone, the skill folder lives at ~/.claude/skills/harvey/ and the folder name must match the name: field in that skill's SKILL.md exactly, or Claude Code won't find it. For a standalone CLI install, just stay inside the cloned harvey/ project folder.
4. Create the virtual environment and install: python3.13 -m venv .venv && source .venv/bin/activate && pip install -e .
5. Copy the environment template and fill it in: cp .env.example .env && $EDITOR .env. Every variable in .env.example is documented inline, including your mail provider credentials (Gmail, SMTP, or Instantly) and any optional discovery provider keys.
6. If Claude is managing this for you, ask it to add any required key to your shell profile (~/.zshrc or ~/.bashrc) rather than pasting it into a command, then open a new terminal so the variable loads.
7. Train Harvey on your own offer: harvey train https://your-company.com. This writes product_knowledge.md, which the Writer agent references so emails describe your actual product instead of something generic.
8. Confirm the signal catalog on the free tier: harvey signals --confirm free.
9. Run your first discovery pass on a small, narrow list: harvey discover.
10. A successful first run looks like a short list of prospects in harvey export or the dashboard, each with a profiled website and a drafted email sequence sitting in harvey outbox, waiting for you, not already sent.

> 💡 **QUICK TIPS**
>
> - Run harvey dashboard to review prospects and drafts visually at localhost:5555 instead of reading raw CLI output.
> - Keep max_daily_sends low for your first week. It's easier to raise a limit than to recover a reputation after bulk-sending a batch of drafts you never read.

04

The commands I actually run

TRAIN HARVEY ON A NEW OFFER

```
harvey train [YOUR-WEBSITE-URL]
```

CHECK WHAT THE SIGNAL CATALOG FOUND BEFORE TRUSTING IT

```
harvey signals --confirm free
```

RUN DISCOVERY WITHOUT COMMITTING TO A SEND

```
harvey discover --estimate
```

READ EVERY DRAFTED EMAIL BEFORE IT GOES ANYWHERE

```
harvey outbox
```

STOP EVERYTHING IMMEDIATELY IF SOMETHING LOOKS OFF

```
harvey sending pause
```

When it breaks

ERROR	FIX
CLI not found / not logged in	Confirm the Claude Code or Codex CLI is installed and you're logged into an active Pro/Max or ChatGPT subscription before running any harvey command.
Emails stuck with unrendered template tags	This is gate.py doing its job. Open skills/email_frameworks.md and fix the template field the Writer agent left blank, then re-run write for that prospect.
Harvey keeps emailing a business you already contacted	Check data/harvey.db wasn't deleted or reset between runs. The SQLite ledger in state.py is what prevents duplicate contact, and a fresh database has no memory of past sends.
Mail won't send at all	Re-check the mail provider credentials in .env. Gmail, SMTP, and Instantly each need their own specific fields filled in exactly as .env.example documents them.
Campaign stopped itself mid-run	That's the bounce-rate threshold or quiet_hours in harvey.yaml working as designed, not a bug. Check both settings before assuming something crashed.
Too many emails approved at once by accident	Avoid harvey outbox --approve-all until you've manually reviewed at least a few full batches. It approves everything currently queued, including drafts you haven't read.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

https://hub.digicuratoragency.com/join