
// pillar 2 · part 2

Build the pipeline



Widen the net with a market-wide keyword search, flag every job where you already know someone, remove the duplicates, and put the whole pipeline on a schedule so it runs itself.

baswe.ai // become a software engineer

This is part two of pillar two, and it completes the pipeline. You already have Tier A pulling jobs from your target companies. Here you add Tier B, a keyword search across the whole market, and Tier C, which flags every job where you already know someone. Then you remove duplicates and put the whole thing on a schedule so it runs on its own. This part ships with a companion code file, `Tier_BC_Market_and_Connections_n8n_code.js`.

Where part 1 left off

Tier A pulls jobs straight from the companies on your target list into a Jobs sheet. That is powerful but narrow: it only sees companies you already knew to add. In this part you widen the net to the whole market, and you layer in the single most valuable signal in the search, whether you have a warm path into each role.

Tier B: search the whole market by keyword

Tier B catches roles anywhere that match your criteria, so you stop missing good companies you had not found yet. Instead of hitting each company one by one, you run a single search across an aggregator, `Hiring.cafe`, through a service called `Apify` that does the heavy lifting and hands back a big list of matching jobs.

▪ Get an Apify token

Sign up at `apify.com` (there is a free tier), then copy your API token from Settings, under Integrations. You paste it into the Search Criteria node in a moment.

▪ Build the Tier B chain

Three nodes, mirroring Tier A but pointed at the market instead of a company list:

1 · Search Criteria (Code node). Run Once for All Items. Paste the first block, then set your token, keyword, and location. It builds the request.

2 · HTTP Request (POST). Fetches the results from Apify. Use the settings in the panel below.

3 · Normalize Hiring.cafe (Code node). Run Once for All Items. Paste the second block. It maps the results into the same clean schema Tier A uses, so both tiers speak the same language.

HTTP REQUEST SETTINGS FOR THE APIFY CALL

Method: POST

URL: {{ \$json.apifyUrl }}

Send Body: ON → JSON (Raw) → {{ \$json.bodyString }}

Options → Timeout: 300000 (the actor runs a real browser, so be patient)

Merge and dedupe

You now have two streams of jobs, Tier A and Tier B, and the same role often shows up in both. Combine them and clean them:

Merge (Append). Add a Merge node in Append mode, and feed both the Tier A Normalize output and the Tier B Normalize output into it. Now everything is one list.

Dedupe (Code node). Run Once for All Items. Paste the Dedupe block. It drops repeats of the same role, matched on company plus title, so each job appears once.

Tier C: flag your warm leads

This is the tier that makes the pipeline yours. A job at a company where you know someone is a warm lead, and warm leads convert far better, so Tier C tags every job Warm or Cold and records who you know there.

▪ Load your connections

Take the LinkedIn connections export you made in pillar one and paste it into a Connections tab in your sheet. Then add a Google Sheets "Get rows" node, rename it exactly Get Connections, and point it at that tab.

▪ Match

Add a Code node named Match Connections, set to Run Once for All Items, and paste the Match Connections block. It normalizes company names and checks each job's company against your connections. If it finds a match, it tags the job Warm and attaches the connection's name and title; otherwise it tags it Cold.

One caution: company names are messy in the export (Acme versus Acme Inc versus Acme AI), and the code normalizes the common variations, but it is not perfect. Skim your Warm tags before you rely on them.

Now every job in your sheet is tagged Warm or Cold, with the name of who you know. Sort by lead type and work your warm leads first. That is where the offers come fastest.

Write it out, tagged

Point your Google Sheets Append node at the Jobs tab, now including the lead_type and connection columns. From there you can sort or filter the sheet by lead type and hand the warm rows to pillar three and the cold rows to pillar four.

Put it on a schedule

Right now the flow runs when you click. To make it run itself, replace the Manual Trigger at the very start with a Schedule Trigger, set to run once a day or a few times a week. Then activate the workflow using the toggle at the top right of the

canvas in n8n Cloud. From then on, your Jobs sheet refreshes on its own, and you just open it and work the warm leads.

The full pipeline, end to end

THE WHOLE MACHINE

Schedule Trigger

```
├ Tier A:  Targets → Build Endpoint → HTTP → Normalize
└ Tier B:  Search Criteria → HTTP (POST) → Normalize Hiring.cafe
→ Merge (Append) → Dedupe → Match Connections → Sheets (Jobs)
```

Troubleshooting and tips

- ▶ **Apify is slow or times out.** That's normal, it runs a real browser. Raise the timeout and lower maxItems while testing.
- ▶ **Blank columns from Tier B.** The actor's field names vary by version. The companion file marks those lines with VERIFY, so do one live pull and adjust.
- ▶ **Warm matches look off.** Company names in the export are inconsistent. The normalizer helps, but eyeball the Warm tags.
- ▶ **Watch your Apify usage.** The free tier has limits, so keep maxItems modest until you're happy with the results.

What you have

You now have a running machine. It sources jobs from your target companies and from the whole market, removes the duplicates, flags every role where you have a warm path, and refreshes on a schedule into your tracker. You never scroll a job board again; you open a sheet that already knows where you have an in.

From here, the search is pure execution. Pillar three works your warm leads into referrals, and pillar four turns your cold leads into direct conversations with decision makers.

→ With the pipeline live, move on to Pillar 3: Warm Paths, referrals and references.