FREE GUIDE · HUSNAINBUKHARI.COM

n8n Lead-Generation Workflow Pack

12 production-ready n8n lead-generation and enrichment workflow blueprints — node-by-node, with the APIs used, a self-hosting cost table, and copy-paste-ready logic.

By Syed Husnain Haider Bukhari — AI engineer & founder, Revolutionary Technologies · www.husnainbukhari.com

n8n Lead-Generation Workflow Pack

This is the companion field guide to my long-form post on n8n lead generation and enrichment. The blog gives you the narrative and the twelve workflows; this pack gives you the reference material you will actually keep open while you build — the node maps, the decision tables, the cost math, and the checklists. It is written to be printed and marked up. Everything here comes from pipelines I run in production for client teams in the United States, United Kingdom, and United Arab Emirates.

The three-stage model every workflow follows

Before you open the n8n canvas, anchor on the model. Every lead-gen pipeline — no matter how many nodes it grows to — is three stages. Get the stages clear and the node choices become obvious.

1. **Acquisition** — get one thin identifier into the system: an email, a domain, a LinkedIn URL, a phone number, or a raw form submission.
2. **Enrichment & validation** — turn that identifier into a full, verified profile, and discard anything that fails verification. This is where most of the value and most of the API spend live.
3. **Routing & activation** — score, deduplicate, upsert to the CRM, and trigger a good first touch.

> A verified, enriched list of 200 leads beats 20,000 raw rows every time. Over-invest in stages two and three, not in scraping everything that moves.

The 12 workflows at a glance

#	Workflow	Trigger	Core nodes
1	Directory scrape → enriched CRM	Schedule	HTTP / HTML Extract / Loop / Verify / Upsert
2	Scrape Google searches	Schedule	SERP API / Split In Batches / Sheet
3	Form-to-CRM instant enrich	Webhook	Set / Enrich / Score / Upsert / Slack
4	LinkedIn / Apollo enrichment	Queue / Schedule	HTTP / Filter / Cache / Upsert
5	Email verification	Inline	HTTP / Switch
6	Lead scoring (rules + LLM)	Inline	Function / LLM / Set
7	Dedupe & identity resolution	Inline	Normalize / Remove Duplicates / DB query
8	Webhook capture (ads / chat)	Webhook	Respond / Switch / Normalize
9	Company → decision-maker	Queue	People API / Filter / Rank / Verify
10	WhatsApp / SMS follow-up	Schedule	Function / WhatsApp API / Webhook
11	CRM hygiene sweep	Schedule	DB query / Verify / Re-enrich / Archive
12	Daily digest & reporting	Schedule	Aggregate / Format / Slack / Email

Scraping Google searches without getting blocked

The single most common mistake I see: pointing a plain HTTP node at google.com and looping it. You will be CAPTCHA-walled within the hour and you will poison the server's IP. Use a SERP API instead. Build a query matrix, throttle it, and log every billed call so a crash never repeats a charge.

```
// Set node — build the SERP request body
{
  "q": "{{ $json.service }} in {{ $json.city }}",
  "gl": "us",
  "num": 20
}

// Split In Batches → 2–4s delay between batches
// Map results[].link into the enrichment loop
// Append every query to a "searches_run" sheet BEFORE the call
```

When a SERP API cannot see what you need — local pack, map data, tight geo-targeting — fall back to a headless browser (Browserless or self-hosted Playwright). Reserve that for the cases that genuinely

require it; it is slower and pricier per call.

Email verification routing table

Put verification in front of every send. Email decays at roughly 2–3% per month, so re-verify before any large campaign too. Branch on the status with a Switch node.

Status	Meaning	Action
Valid	Mailbox exists, accepts mail	Send with confidence
Invalid	Hard bounce guaranteed	Discard, never retry
Catch-all	Domain accepts everything	Warmed domain only
Risky / role	info@, sales@, etc.	Avoid for cold outreach
Unknown	Verifier timed out	Re-queue once, then treat as risky

Lead scoring pattern

Blend a deterministic rule pass with an optional LLM judgment call. Run the cheap deterministic layer first; only send to the model the leads that clear a threshold, to control cost.

```
// Function node — deterministic pass
let score = 0;
const d = $json;
if (d.employees >= 50) score += 25;
if (['us','uk','ae'].includes(d.country_code)) score += 20;
if (d.pricing_page_visits > 0) score += 30;
if (/gmail|yahoo|hotmail/.test(d.email)) score -= 15;
if (d.is_competitor) score -= 100;
return [{ json: { ...d, rule_score: score } }];
```

For the LLM pass, constrain the output to a JSON object — a numeric intent score and a one-sentence reason — and keep the temperature low. You want a consistent judge, not a creative writer.

Dedupe checklist

A CRM your sales team trusts is one with no duplicates and no clobbered data. Normalize, then match, then merge — never overwrite.

- Lowercase and trim every email; strip domains to their root.
- Normalize phone numbers to E.164 before comparison.

- Match on verified email first; fall back to domain + fuzzy name.
- Merge enrichment data on conflict — never blindly overwrite paid-for fields.
- Write an audit row on every merge so you can trace what happened.

Self-hosting cost worksheet

The n8n community edition is free to license. Your real spend is infrastructure and the enrichment APIs that do the actual work. Fill in your own volumes.

Line item	Typical monthly (USD)
VPS (4 vCPU / 8GB) + managed Postgres	40 – 90
Redis for queue mode (scale)	10 – 30
SERP / Google search API	30 – 80
People-data / enrichment API	50 – 200
Email verification	20 – 60
Backups, monitoring, alerting	10 – 25
All-in (small-to-mid pipeline)	**160 – 500**

The number this table hides is labor: upgrades, patches, backups, queue tuning, and the occasional 2 a.m. page. Below a few thousand executions a day, n8n Cloud is usually cheaper than your time. Above it, self-host and put a technical owner on it.

Pre-launch hardening checklist

Run through this before any workflow touches a client's data. Skipping these is the difference between a demo and a system.

- Idempotent upserts keyed on a stable identifier — re-runs never duplicate.
- Retry-with-backoff on every HTTP node, with a dead-letter sheet for permanent failures.
- Rate limiting and jittered delays on all scraping and SERP calls.
- Email verification in front of every send; a re-verification sweep before big campaigns.
- Enrichment responses cached by input key to eliminate repeat spend.
- Structured logging of every billed call plus a daily cost digest.
- Secrets in n8n credentials; PII handling and retention documented and signed off.

Build vs buy — a quick decision rule

Build in n8n where the logic is a genuine competitive edge and you have engineering capacity to maintain it. Buy a managed pipeline for the commodity middle that every business needs and nobody should maintain from scratch. If you want reliable pipeline without standing up and babysitting a scraping-and-enrichment stack, ProLeads is the managed layer I reach for — build your custom edges in n8n, buy the commodity middle.

Work with me

If you want these workflows built, audited, or rescued, that is the core of what I do. Work with Syed Husnain Haider Bukhari and tell me what your pipeline looks like today — I will tell you honestly what to build, what to buy with something like ProLeads, and what to leave alone.
