

# n8n Production-Readiness Checklist

By Syed Noor / [noorflows.com](https://noorflows.com)

---

Use this checklist before promoting any n8n workflow to production. Walk through each dimension, tick the boxes, note the red flags. If you score below 18/24, your workflow is not production-ready — it will break at 2 AM on a Friday and you will not know until Monday.

---

## 1. Idempotency

---

*Can the workflow safely run twice on the same input without creating duplicates or corrupting state?*

- ☐ Webhook trigger has a dedup check using a request hash or external ID stored in a persistent store (Postgres/Redis), not just n8n's built-in "last execution" memory
- ☐ Upsert operations use a natural key ( `ON CONFLICT DO UPDATE` ), not blind `INSERT` — a retry will update the existing row, not create a duplicate
- ☐ Side effects (Slack messages, emails, invoice creation) are guarded by a "was this already sent?" lookup keyed on the triggering event ID
- ☐ If the workflow processes batches, each item is handled independently — one item's failure does not roll back or skip the rest

**Red flag:** You trigger the workflow manually twice and get two Slack messages / two CRM contacts / two invoices for the same event.

**Quick fix:** Add a Function node at the top that checks `SELECT 1 FROM processed_events WHERE event_id = $json.id`. If found, stop execution with a "No Operation" node.

---

## 2. Retry & Backoff

---

*What happens when an external API returns 429 (rate limit) or 503 (service unavailable)?*

- ☐ Every HTTP Request node has retry enabled with exponential backoff (n8n Settings > Retry On Fail > Wait Between Retries = `{{ $retryCount * 2000 }}` ms or equivalent)
- ☐ Rate-limited APIs are called through a "throttle" pattern — SplitInBatches with a Wait node set to the provider's documented rate limit (e.g., 1 req/sec for Notion API)
- ☐ OAuth2 token refresh is handled by n8n credential type, not a manual "get new token" HTTP call that can itself fail and cascade
- ☐ The workflow has been tested with a simulated 429 response (use a Respond to Webhook mock) to confirm retry actually fires

**Red flag:** You see chains of `ERROR: Request failed with status code 429` in execution logs, each 0ms apart — the workflow is hammering the API with no pause.

**Quick fix:** On every HTTP Request node: Settings > Retry On Fail = On, Max Tries = 3, Wait Between Retries = 2000ms. Then wrap multi-item loops in SplitInBatches with batch size 1 and a 1-second Wait

node.

---

### 3. Audit Trails

---

*Can you reconstruct what happened, to which record, and when — without reading n8n execution logs line by line?*

- ☐ Every workflow writes a structured log entry (timestamp, workflow name, execution ID, action, target record ID, outcome) to a dedicated `audit_log` table or external log sink (Loki, Datadog, a Google Sheet — anything queryable)
- ☐ Error paths write to the same audit log with `status: 'error'` and the error message, not just the n8n error handler's default "send me an email"
- ☐ The audit log is append-only — no workflow has UPDATE or DELETE permissions on the log table
- ☐ Execution data retention is set explicitly ( `EXECUTIONS_DATA_MAX_AGE` / `EXECUTIONS_DATA_PRUNE_MAX_COUNT` ) and matches your compliance window (SOC 2 = 1 year, GDPR audit = variable)

**Red flag:** Someone asks "did we send that invoice to Acme Corp on Tuesday?" and the only way to answer is scrolling through n8n's execution list hoping the data hasn't been pruned.

**Quick fix:** Add a Postgres/MySQL node at the end of every workflow (including error branches) that inserts one row: `{ ts: $now, workflow: $workflow.name, exec_id: $execution.id, action: 'completed', detail: $json.summary }`.

---

### 4. Secrets Management

---

*Can you rotate an API key without opening the n8n editor and touching every workflow that uses it?*

- ☐ All credentials use n8n's built-in credential store (not hardcoded strings in Function nodes or HTTP header fields)
- ☐ Environment variables ( `$env.STRIPE_KEY` ) are used for secrets that multiple workflows share, so rotation is a single env-var update + restart
- ☐ No credential values appear in workflow JSON exports — verified by exporting a workflow and searching for key prefixes ( `sk_live_` , `Bearer` , `xoxb-` )
- ☐ Credential access is scoped: production workflows use a dedicated n8n credential per service, not "Syed's personal API key" shared across dev and prod

**Red flag:** You exported all workflows as JSON, ran `grep -i "sk_live\|Bearer\|api_key\|password"` and got hits.

**Quick fix:** For every hardcoded secret found: create an n8n credential of the matching type, paste the value there, delete the plaintext from the node, and reference the credential instead. Takes 2 minutes per node.

---

## 5. Dead-Letter Queues

---

*Where do failed executions go? Can you replay them?*

- ☐ Every workflow has an Error Trigger workflow attached (Settings > Error Workflow) that captures the failed execution ID, input data, and error message
- ☐ Failed execution payloads are persisted outside n8n (a `dead_letter` table, an S3 bucket, a dedicated Slack channel with structured JSON) so they survive n8n restarts and data pruning
- ☐ There is a documented replay procedure: someone can take a dead-letter entry, feed it back into the workflow's webhook/trigger, and reprocess it without manual data surgery
- ☐ The dead-letter store is monitored — if it grows beyond N entries in 24 hours, an alert fires (see Dimension 6)

**Red flag:** `EXECUTIONS_DATA_PRUNE` is on, max age is 7 days, and nobody checks failed executions. Failures silently disappear.

**Quick fix:** Create one global Error Trigger workflow that writes `{ execution_id, workflow_name, error_message, input_data, timestamp }` to a Postgres `dead_letters` table. Set it as the error workflow for every production workflow. Total setup: 15 minutes.

---

## 6. Monitoring & Alerting

---

*If n8n crashes at 3 AM, who finds out — and how fast?*

- ☐ An external uptime monitor (UptimeRobot free tier, Healthchecks.io, or a cron curl) pings n8n's `/healthz` endpoint every 60 seconds and alerts on two consecutive failures
- ☐ Workflow failure rate is tracked — if more than 3 executions fail within 10 minutes on any single workflow, an alert fires (Slack/email/PagerDuty)
- ☐ Resource usage (CPU, memory, disk) on the n8n host is monitored with threshold alerts (disk > 85%, memory > 90%) — n8n silently stops processing when the DB disk fills up
- ☐ There is a weekly "n8n health" review cadence: someone looks at total executions, failure rate, queue depth, and longest-running workflows

**Red flag:** n8n's Postgres disk filled up on Saturday. Nobody noticed until Monday when a customer complained their webhook wasn't processed. 48 hours of lost data.

**Quick fix:** Sign up for Healthchecks.io (free, 20 checks). Add a Schedule Trigger workflow that runs every 5 minutes and pings your Healthchecks.io URL. If n8n dies, Healthchecks.io emails you within 10 minutes.

---

## Scoring Guide

---

Count your checked boxes across all 6 dimensions (24 total).

| Score | Rating           | Interpretation             |
|-------|------------------|----------------------------|
| 22-24 | Production-ready | Ship it. Review quarterly. |

| Score        | Rating       | Interpretation                                                                   |
|--------------|--------------|----------------------------------------------------------------------------------|
| <b>18-21</b> | Almost there | Fix the gaps this sprint. No critical risk but you have blind spots.             |
| <b>13-17</b> | Risky        | Do not go live until you address every unchecked item in Dimensions 1, 4, and 6. |
| <b>0-12</b>  | Not ready    | This is a prototype, not a production workflow. Budget a full hardening sprint.  |

---

## Want a Professional Review?

---

This checklist catches the obvious issues. A trained eye catches the subtle ones — the race condition in your parallel branches, the OAuth token that expires mid-batch, the webhook that silently drops payloads when n8n restarts during a deploy.

**Pre-flight Review — \$247** I audit your workflow exports against 40+ production criteria and deliver a prioritized fix list with exact node-level instructions. Fully async, delivered in 24–72 hours — no calls, no time-zone friction.

Book yours: [noorflows.com/products/a-preflight-review/](https://noorflows.com/products/a-preflight-review/)

---

*noorflows.com — n8n automation consulting for teams that ship.*