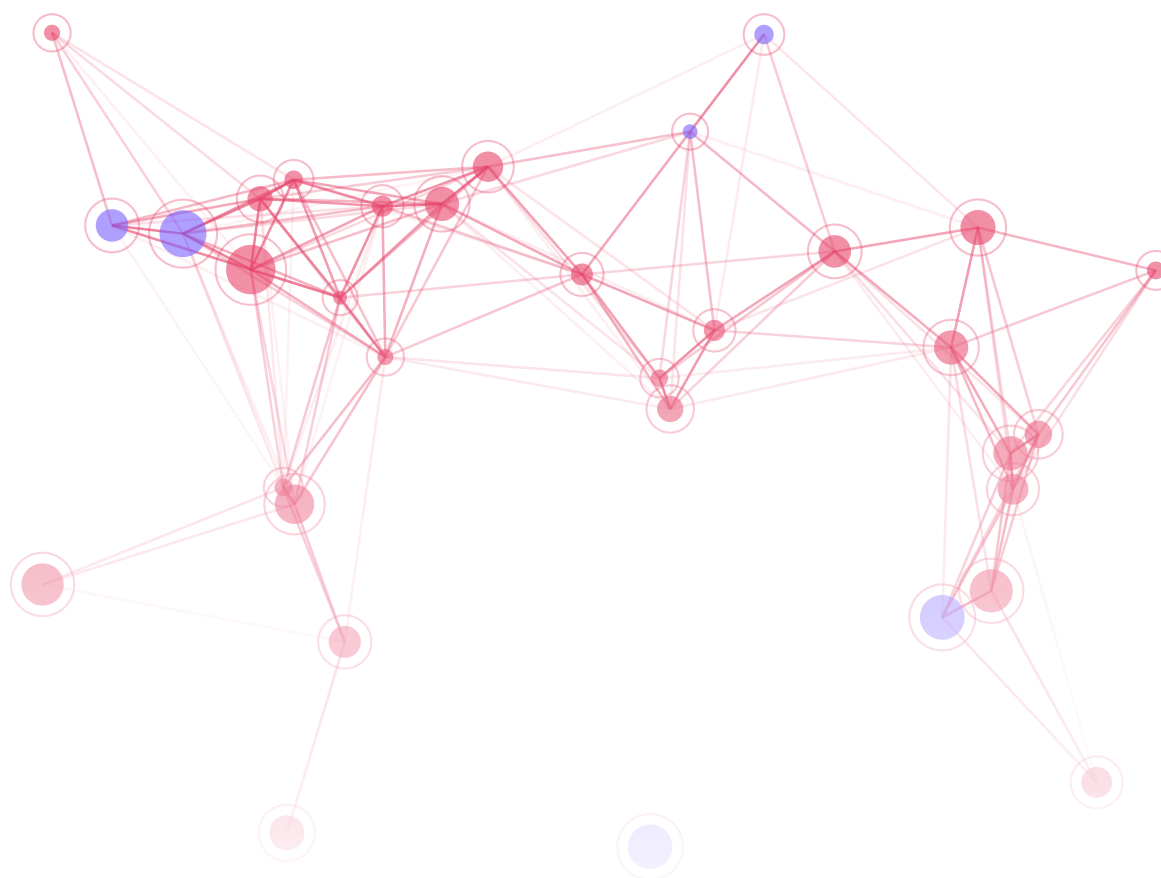




A BEGINNER'S MANUAL

GROK EDITION

n8n

Automate your work and build AI agents, visually, one node at a time

n8n

Automate your work and build AI agents, visually, one node at a time

GROK EDITION

Concise edition, revised and fact-checked by Grok Bot in GrokAllManualsV1.

n8n (say "n-eight-n," short for "nodemation") is a visual automation tool. You connect building blocks, called **nodes**, on a canvas to make **workflows** that run on their own: "every morning, check the weather and email me," "when someone fills in my form, add them to a spreadsheet and send a welcome message," or "when a support email arrives, have an AI draft a reply."

n8n connects to more than a thousand apps, has powerful AI features built in, and, unlike most automation tools, you can run it on your own computer for free. This manual takes you from your first click to AI agents, MCP, and reliable production workflows.

HOW TO READ THIS MANUAL

Tip — a shortcut or good habit.

Warning — easy to get wrong.

\$ **command** — text you type, exactly as shown.

Note — useful background.

Danger — can lose data, money, or access.

Settings > **Privacy** — a menu path to click.

New words appear in **bold** the first time and are collected in the Glossary at the back.

Contents

01 Meet n8n	5
1.1 What is workflow automation?	5
1.2 How n8n compares	6
1.3 Cloud or self-hosted?	6
02 Key Ideas in Five Minutes	7
2.1 Items: the most important idea	7
2.2 Draft versus published	7
03 Get n8n Running	8
3.1 Option A: n8n Cloud (recommended)	8
3.2 Option B: self-host on your own computer	8
04 A Tour of the Editor	10
4.1 The main areas	10
4.2 Adding nodes	10
4.3 The node details view	10
05 Your First Workflow: A Morning Weather Email	12
5.1 Add a decision	13
06 Working with Data	14
6.1 Mapping by dragging	14
6.2 Expressions cheat sheet	14
6.3 Nodes for reshaping data	15
6.4 Pinning data	15
07 Triggers: How Workflows Start	16
7.1 Test URLs versus production URLs	16
08 Connecting Your Apps	17
8.1 Credentials	17
8.2 When there's no node: HTTP Request	17
8.3 Security habits	17
09 AI in n8n	18
9.1 AI steps in a workflow	18
9.2 The AI Agent node	18
9.3 Agents (first-class, in preview)	19
9.4 MCP and n8n	19
9.5 Let AI build workflows for you	20

10	Reliable Workflows	21
10.1	Debugging with executions	21
10.2	Handling errors	21
10.3	Good habits	22
11	Templates and Community	23
12	Self-Hosting Essentials	24
13	Five Complete Projects	25
13.1	Project 1: Daily RSS digest	25
13.2	Project 2: Form to spreadsheet	26
13.3	Project 3: AI email summariser	26
13.4	Project 4: A tiny API	27
13.5	Project 5: Error alerts	27
14	Node and Expression Reference	29
14.1	Core nodes	29
14.2	Expression variables	30
14.3	Code node snippets	31
14.4	Cron cheat sheet	31
15	Troubleshooting and FAQ	32
16	Self-Hosting Environment Variable Reference	33
16.1	Setting variables	33
16.2	Deployment	34
16.3	Endpoints	37
16.4	Executions	38
16.5	Security	40
16.6	Database	41
16.7	Nodes	42
16.8	Logs	43
16.9	Timezone and Localization	44
16.10	Workflows	44
16.11	Credentials	45
16.12	Binary Data	45
16.13	Queue Mode	45
16.14	Task Runners	46
16.15	User Management and 2FA	49
17	Built-In Node Directory	50
	Quick Reference	57
	Glossary	58

CHAPTER 1

Meet n8n

IN THIS CHAPTER

- What n8n does and how it compares to Zapier and Make
- Cloud versus self-hosted
- What you'll build in this manual

1.1 What is workflow automation?

A **workflow** is a recipe that a computer follows automatically. It always starts with a **trigger** (*something happens*), then runs **steps** (*do these things*):

TRIGGER	STEPS
Every weekday at 8am	Get today's calendar → summarize with AI → send to Slack
A new row in Google Sheets	Create an invoice in Stripe → email the customer
A form is submitted	Add the person to your mailing list → notify you on Telegram
An email arrives at support@	AI sorts it → drafts a reply → creates a task

In n8n, every trigger and every step is a **node**: a box on a canvas. You connect the boxes with lines, and data flows along those lines from left to right.

1.2 How n8n compares

	N8N	ZAPIER	MAKE
Style	Visual canvas with branches and loops	Simple step-by-step lists	Visual canvas
Self-hosting	Yes, free	No	No
Pricing basis	Per workflow <i>execution</i> , however many steps	Per task (each step counts)	Per operation
Code when you need it	JavaScript or Python nodes	Limited	Limited
AI agents and MCP	Built in	Yes	Yes
Best for	Power, flexibility, and low cost at scale	The quickest simple automations	Visual mid-complexity flows

NOTE

n8n is “fair-code”: its source code is public and the self-hosted Community Edition is free to use, including for your own business, under n8n’s Sustainable Use License. You only pay for n8n Cloud or for enterprise features.

1.3 Cloud or self-hosted?

- **n8n Cloud:** n8n runs it for you. Sign up and start in the browser. Easiest, with a free trial. **Recommended for beginners.**
- **Self-hosted:** you run n8n on your own computer or a server. Free and unlimited, but you’re responsible for keeping it running, updated, and secure.

You can start on Cloud and move later. Workflows export and import as files.

CHAPTER 2

Key Ideas in Five Minutes

TERM	MEANING
Workflow	A set of connected nodes that does one job
Node	One block: a trigger, an action in an app, or a logic step
Trigger node	The first node, which starts the workflow (a schedule, a webhook, a new email...)
Item	One piece of data moving through the workflow, like one email or one spreadsheet row. Nodes run once per item
JSON	The text format n8n uses to show data: <code>{ "name": "Sam", "city": "Austin" }</code>
Expression	A small formula inside <code>{{ }}</code> that pulls in data from earlier nodes
Credential	Saved sign-in details that let a node use an app on your behalf
Execution	One run of a workflow; n8n keeps a log of each
Publish	Make a workflow live, so its triggers fire automatically

2.1 Items: the most important idea

When a node outputs **five rows** from a spreadsheet, that's **five items**, and the next node normally runs **five times**, once for each. That's how a “send an email” node can send one personalized email per row without any loops.

2.2 Draft versus published

n8n saves your edits automatically as a **draft**. Your triggers only run automatically after you click **Publish** (or press `Shift + P`). Later edits stay in draft until you publish again, so you can safely tinker with a live workflow.

Get n8n Running

3.1 Option A: n8n Cloud (recommended)

START ON N8N CLOUD

- 1 **Go to n8n.io** and click **Get started**. Create an account; a free trial starts automatically.
- 2 **Choose a name** for your workspace. It becomes your address, like `yourname.app.n8n.cloud`.
- 3 **Open the editor.**
You'll see the Overview page with your workflows list.

CLOUD PLAN	PRICE (BILLED YEARLY)	EXECUTIONS PER MONTH	HIGHLIGHTS
Starter	€20/month	2,500	Unlimited workflows, AI agents and MCP, 5 concurrent runs
Pro	€50/month	10,000	More concurrency, admin roles, workflow history, insights
Business	€667/month	40,000	SSO, environments, Git version control (self-hosted)
Enterprise	Contact sales	Custom	Advanced security, support with an SLA

Prices from n8n.io/pricing, October 2026. Monthly billing costs a little more. An **execution** is one full run of a workflow, however many nodes it has.

3.2 Option B: self-host on your own computer

The one-line setup (macOS, Linux, or WSL):

TERMINAL

```
curl -fsSL https://get.n8n.io | sh
```

With Docker (any system with Docker Desktop installed):

TERMINAL

```
docker volume create n8n_data

docker run -it --rm --name n8n -p 5678:5678 \
  -e GENERIC_TIMEZONE="America/New_York" -e TZ="America/New_York" \
  -v n8n_data:/home/node/.n8n \
  n8nio/n8n
```

Just trying it? If you have Node.js (version 20.19–24) installed, run `npx n8n`.

Then open <http://localhost:5678> in your browser and create the owner account.

⚠ WARNING

A self-hosted n8n on your laptop only runs while your laptop is on and awake, and outside services can't reach its webhooks. For always-on automations, use n8n Cloud or a small cloud server (see Chapter 12).

💡 TIP

Replace `America/New_York` with your own time zone (for example `Europe/London` or `America/Los_Angeles`) so schedules fire at the right local time.

A Tour of the Editor

4.1 The main areas

- **Left sidebar:** Overview (all workflows), Templates, Credentials, and Settings.
- **Canvas:** the big grid where you place and connect nodes. Scroll to zoom; drag the background to pan.
- **Top bar:** the workflow's name, the **Publish** button, and the **Editor / Executions** tabs.
- **Execute workflow** button (bottom of canvas): runs the whole workflow once, right now, for testing.

4.2 Adding nodes

Click the + on the canvas (or press `Tab`) to open the **nodes panel**. Search by app ("Gmail") or by action ("send email"). Nodes are grouped as:

- **Triggers:** how a workflow starts.
- **App nodes:** actions in other services (Google Sheets, Slack, Notion, Airtable...).
- **Core nodes:** logic and data: If, Switch, Edit Fields (Set), Merge, Loop Over Items, Wait, Code, HTTP Request.
- **AI nodes:** AI Agent, chat models, memory, tools, vector stores.

4.3 The node details view

Double-click any node to open it. You'll see three panels:

PANEL	WHAT IT SHOWS
Input (left)	The data arriving from the previous node
Parameters (middle)	The node's settings: which account, which operation, which fields
Output (right)	What the node produced after you click Execute step

Switch the input and output views between **Table**, **JSON**, and **Schema**. Table is easiest to read; JSON shows the exact structure.

 TIP

Test every node as you add it. Click **Execute step**, check the output looks right, then move on. Building in small tested steps is the secret to workflows that work first time.

Your First Workflow: A Morning Weather Email

Let's build a workflow that checks the weather every morning and emails you a summary. It uses a free weather service that needs no account.

BUILD THE WEATHER WORKFLOW

1

Create a workflow.

On the Overview page click **Create Workflow**. Name it "Morning weather".

2

Add a Schedule Trigger.

Click **Add first step**, search **Schedule**, and choose **Schedule Trigger**. Set **Trigger Interval** to **Days** and **Trigger at Hour** to **7am**. Close the node.

3

Add an HTTP Request node.

Click the + after the trigger, search **HTTP Request**, and paste this into **URL** (it's New York; change the latitude and longitude to your city):

TEXT

```
https://api.open-meteo.com/v1/forecast?
latitude=40.71&longitude=-74.01&daily=temperature_2m_max,temperature_2m_min,precipitation_proba
bility_max&temperature_unit=fahrenheit&timezone=auto
```

Click **Execute step**. The output shows today's forecast in a `daily` section.

- 4 **Add an Edit Fields (Set) node** to make a friendly message. Add a field named `message` and switch its value to **Expression** mode, then enter:

TEXT

```
Today: high {{ $json.daily.temperature_2m_max[0] }}°F, low {{ $json.daily.temperature_2m_min[0] }}°F, {{ $json.daily.precipitation_probability_max[0] }}% chance of rain.
```

Click **Execute step** and check the message reads correctly.

- 5 **Add a Gmail node** (or Outlook, or Send Email). Choose the **Send a message** operation. Under **Credential**, choose **Create new credential** and sign in with Google. Set **To** to your address, **Subject** to `Your morning weather`, and drag the `message` field from the input panel into **Message**.

- 6 **Test the whole thing.**

Click **Execute workflow** and check your inbox.

- 7 **Publish.**

Click **Publish** (top right). From tomorrow, it runs every morning at 7.

Congratulations: you've built a real automation. Notice the pattern you'll use forever: **trigger** → **get data** → **reshape data** → **send it somewhere**.

NOTE

The `[0]` in the expression means “the first entry in the list,” which is today. `[1]` would be tomorrow.

5.1 Add a decision

Make it smarter: only send an umbrella warning if rain is likely.

ADD AN IF NODE

- 1 **Insert an If node** between Edit Fields and Gmail. Hover over the connecting line and click **+**.
- 2 **Set the condition.**
Value 1: drag in `precipitation_probability_max` (or use the expression `{{ $('HTTP Request').item.json.daily.precipitation_probability_max[0] }}`). Operation: **Number > is greater than**. Value 2: `50`.
- 3 **Connect the outputs.**
Connect the **true** output to your Gmail node with the subject “☔ Take an umbrella!”, and add a second Gmail node on **false** for normal days, or leave false empty to send nothing.

Working with Data

6.1 Mapping by dragging

The easiest way to use data from an earlier node: open a node, find the field in the **Input** panel, and **drag it** into a parameter. n8n writes the expression for you.

6.2 Expressions cheat sheet

Expressions live inside double curly braces and can contain small bits of JavaScript.

EXPRESSION	GIVES YOU
<code>{{ \$json.email }}</code>	The <code>email</code> field of the current item
<code>{{ \$json.name.toUpperCase() }}</code>	The name in capital letters
<code>{{ \$('Google Sheets').item.json.total }}</code>	A field from a specific earlier node, by name
<code>{{ \$now.toFormat('yyyy-MM-dd') }}</code>	Today's date, like <code>2026-10-15</code>
<code>{{ \$today.minus(7, 'days') }}</code>	The date one week ago
<code>{{ \$json.price * 1.08 }}</code>	Arithmetic: price plus 8% tax
<code>{{ \$json.status === 'paid' ? 'Thanks!' : 'Reminder' }}</code>	Choose text based on a condition

TIP

Not sure how to write an expression? Click **Ask AI** or the AI assistant in the expression editor and describe what you want: “the first name from the `full_name` field.”

6.3 Nodes for reshaping data

NODE	USE IT TO
Edit Fields (Set)	Add, rename, or remove fields
If / Switch	Send items down different paths
Filter	Keep only items that match a rule
Merge	Combine data from two branches
Aggregate / Split Out	Turn many items into one list, or one list into many items
Loop Over Items	Process items in batches (useful with rate limits)
Code	Write JavaScript or Python when nothing else fits
Data tables	Store data inside n8n itself, like a simple spreadsheet, to reuse across workflows

6.4 Pinning data

While building, click **Pin** on a node's output to freeze it. Later nodes then use that saved sample instead of calling the real service again, which is perfect for working on a webhook or an expensive AI step without re-triggering it.

Triggers: How Workflows Start

TRIGGER	STARTS WHEN...	EXAMPLE
Schedule Trigger	A time arrives: every hour, every Monday at 9...	Weekly report
Webhook	Another app sends data to n8n's unique URL	Payment received in Stripe
n8n Form Trigger	Someone submits a form n8n hosts for you	Event sign-up
App triggers	Something happens in an app (new email, new row, new file)	New Gmail message with a label
Chat Trigger	Someone sends a message in a chat window	A chatbot for your website
Execute Sub-workflow Trigger	Another workflow calls this one	Reusable "send notification" sub-workflow
Error Trigger	Another workflow fails	Alert yourself on failure
Manual	You click Execute workflow	Testing

7.1 Test URLs versus production URLs

Webhook and Form triggers each have **two** addresses:

- **Test URL:** works while you click **Listen for test event** in the editor, so you can see data arrive.
- **Production URL:** works once the workflow is **published**.

The most common beginner puzzle, “my webhook works in testing but not for real,” is almost always because the workflow isn’t published, or the other app is still using the test URL.

Connecting Your Apps

8.1 Credentials

A **credential** stores the sign-in for an app so nodes can use it. Create them from a node (**Credential** > **Create new credential**) or from **Credentials** in the sidebar. There are two main kinds:

- **OAuth** (“Sign in with Google,” “Connect to Slack”): click a button and approve in a pop-up. On n8n Cloud, most of these work instantly.
- **API key or token**: copy a secret key from the app’s developer settings and paste it in.

NOTE

On **self-hosted** n8n, some OAuth apps (notably Google) require you to create your own “OAuth app” in that company’s developer console first. Each credential page in n8n links to step-by-step instructions.

8.2 When there’s no node: HTTP Request

Almost every online service has an **API**. If n8n has no dedicated node for it, use the **HTTP Request** node with the service’s documentation. Many API docs show a `curl` command; n8n’s HTTP Request node has **Import cURL**, which fills in everything for you.

8.3 Security habits

- Name credentials clearly, for example “Gmail: personal” and “Gmail: business.”
- Give each credential the minimum access it needs.
- Never paste keys into Code nodes or text fields; always use credentials.
- Remove credentials you no longer use.

AI in n8n

n8n has three layers of AI: AI *steps* inside workflows, AI *agents* that decide what to do, and AI that *builds workflows for you*.

9.1 AI steps in a workflow

Add an AI model node anywhere to summarize, classify, translate, or extract information. The **Basic LLM Chain**, **Information Extractor**, **Text Classifier**, and **Sentiment Analysis** nodes cover the common jobs. Connect a **chat model** (OpenAI, Anthropic Claude, Google Gemini, Mistral, Ollama for local models, and more) using your API key.

TIP

On n8n Cloud Starter and Pro plans you can choose **Use Gateway credits** in a model's credential field. n8n handles billing, so you don't need your own OpenAI or Anthropic account. Great for getting started.

9.2 The AI Agent node

The **AI Agent** node is a mini-agent inside a workflow. You attach:

PART	WHAT IT'S FOR
Chat model	The AI brain (e.g. Claude or GPT)
Memory	Remembers the conversation, e.g. Simple Memory , or a database for persistence
Tools	Things the agent can do: Gmail, Google Calendar, HTTP Request, a sub-workflow, or an MCP Client Tool
System message	Instructions: who it is and how to behave

BUILD A SIMPLE CHATBOT AGENT

- 1 **Add a Chat Trigger** as the first node.
- 2 **Add an AI Agent node** after it.
- 3 **Attach a chat model** (click **Chat Model +**) and choose, for example, **Anthropic Chat Model** with your credential or Gateway credits.
- 4 **Attach Simple Memory** so it remembers the conversation.
- 5 **Attach a tool**, for example **Google Calendar** with the operation *Get many events*, so it can answer “what’s on my schedule?”
- 6 **Write the system message:**
“You are my friendly scheduling assistant. Use the calendar tool to answer questions about my day. Be brief.”
- 7 **Click Open chat** at the bottom of the canvas and talk to it.

9.3 Agents (first-class, in preview)

Newer versions of n8n also offer **Agents** as their own kind of project item, alongside workflows. You configure a model, instructions, tools, skills, and a knowledge base in the **Agent Builder**, then let people reach the agent through chat, channels, or schedules. Agents can call your workflows as tools. The feature is in preview, so expect changes.

9.4 MCP and n8n

n8n speaks MCP in three ways (see the *MCP* manual in this series):

FEATURE	DIRECTION	USE IT TO
MCP Client Tool node	n8n uses other MCP servers	Give your AI Agent tools from GitHub, Notion, Linear, and more. The node panel also has a one-click MCP Servers registry
MCP Server Trigger node	Other AI apps use one n8n workflow	Turn a workflow into a custom MCP server that Claude or Cursor can call
Instance-level MCP server	Other AI apps use all of n8n	Let Claude, ChatGPT, or a coding agent search, run, <i>and build</i> your workflows from a chat

9.5 Let AI build workflows for you

- **n8n Assistant / Ask n8n AI:** describe what you want inside n8n and it drafts or fixes workflows and expressions.
- **From Claude or another MCP client:** connect to n8n's instance-level MCP server and say "build a workflow that saves Gmail attachments to Google Drive."

WARNING

Review AI-built workflows before publishing, especially nodes that send messages, delete data, or spend money. Test with pinned or sample data first.

Reliable Workflows

10.1 Debugging with executions

Click the **Executions** tab to see every run, its status (succeeded, failed, running), and the exact data each node received and produced. Click a failed execution, find the red node, and read its error. You can **Debug in editor** to load that run's data onto the canvas and fix it.

10.2 Handling errors

TOOL	WHAT IT DOES
Retry On Fail (node settings)	Automatically retry a flaky step, e.g. 3 times, waiting 5 seconds
On Error > Continue (node settings)	Keep the workflow going and send the error down a separate output
Error workflow	In workflow Settings , choose a workflow that starts with an Error Trigger ; it runs whenever this one fails, so you can email or Slack yourself the details
Stop and Error node	Deliberately fail with a clear message when something's wrong

TIP

Make one “Alert me” error workflow, with Error Trigger → Gmail or Slack, and set it as the error workflow for everything you publish. You'll know about problems before anyone else does.

10.3 Good habits

- ☐ Name every node clearly (“Get today’s orders,” not “HTTP Request3”)
- ☐ Add **sticky notes** on the canvas explaining tricky parts
- ☐ Test with pinned data before publishing
- ☐ Set an error workflow
- ☐ Watch your execution count on Cloud; use filters early to skip unnecessary runs
- ☐ Use **sub-workflows** (Execute Sub-workflow) for steps you repeat in many workflows

TIP

Review these habits whenever a workflow fails in a way that surprised you. Most production problems in n8n trace back to an unnamed node nobody understands, a missing error workflow, or a trigger that runs far more often than it needs to. Fixing those three things prevents most late-night surprises.

Templates and Community

You rarely need to start from scratch. **n8n.io/workflows** hosts thousands of free templates: lead capture, social media posting, invoice processing, AI support agents, and more. Click **Use workflow** to copy one into your n8n, then add your own credentials.

The **community forum** (community.n8n.io) is friendly and active. Search it first; someone has usually solved your problem. Self-hosted users can also install **community nodes**, extra nodes built by other users, from **Settings > Community nodes**. Install only well-maintained ones from authors you trust.

Self-Hosting Essentials

If you self-host for real (not just to experiment), cover these basics:

TOPIC	WHAT TO DO
Always-on hosting	Use a small cloud server (DigitalOcean, Hetzner, AWS...) or a service with one-click n8n, rather than your laptop
Docker Compose	n8n's recommended production setup, with a proper database (PostgreSQL)
HTTPS and a domain	Put n8n behind a reverse proxy with a certificate, and set <code>WEBHOOK_URL</code> to your public address so webhooks work
Encryption key	n8n encrypts credentials with a key in its data folder. Back it up; without it, saved credentials can't be decrypted
Backups	Back up the data volume and database regularly; export important workflows too
Updates	Update regularly for features and security fixes; read the release notes for breaking changes first
Access	Use a strong owner password and two-factor authentication; don't expose the editor to the internet unprotected

NOTE

The npm installation method is deprecated from n8n 3.0. Use Docker or Docker Compose for anything long-lived.

Five Complete Projects

Each project is a full build, from an empty canvas to a published workflow. Work through them in order, because each one adds a new skill.

IN THIS CHAPTER

- Project 1: RSS news digest by email (schedules, loops, aggregation)
- Project 2: Web form to a spreadsheet (form trigger, mapping)
- Project 3: AI email summariser (AI nodes, filters)
- Project 4: A tiny API with a webhook (webhooks, responses)
- Project 5: Error alerts for all your workflows (Error Trigger)

13.1 Project 1: Daily RSS digest

1

Add a Schedule Trigger.

Set it to every day at 07:00. Check the workflow's time zone in **Workflow settings** > **Timezone**.

2

Add an RSS Read node.

Paste a feed URL, such as a news site's RSS feed. Click **Execute step**. You'll get one item per article.

3

Add a Limit node.

Keep the first 10 items.

4

Add an Aggregate node.

Choose *Individual fields*, field `title`, and also `link`. Now you have one item holding lists.

5 Add a Code node to build the email body:

JAVASCRIPT

```
const t = $json.title, l = $json.link;
const html = t.map((x, i) => `<li><a href="${l[i]}">${x}</a></li>`).join('');
return [{ json: { html: `<ul>${html}</ul>` } }];
```

6 Add a Send Email (or Gmail) node.

Set the subject to `Morning digest {{ $now.toFormat('yyyy-MM-dd') }}`, choose HTML, and set the body to `{{ $json.html }}`.

7 Test the whole workflow, then Publish it.

13.2 Project 2: Form to spreadsheet

1 Add an n8n Form Trigger.

Add fields: Name (text, required), Email (email), Message (textarea).

2 Open the test URL and submit a test entry.

3 Add a Google Sheets node (Append row). Connect your Google credential, then choose the spreadsheet and sheet.

4 Map columns by dragging Name, Email, and Message from the input panel.

5 Add a "thank you" step.

Use the form's completion settings to show a message.

6 Publish and share the production form URL.

13.3 Project 3: AI email summariser

1 Trigger:

a Gmail Trigger (or Email Trigger (IMAP)) that polls for new mail.

2 Filter node:

keep only messages where the sender matches the people you care about.

3

AI step:

add a Basic LLM Chain (or an OpenAI / Anthropic node) with the prompt:

PROMPT

Summarise this email in 2 sentences, then list any action items with due dates. Email: {{ \$json.text }}

4

Send the summary to yourself with Telegram, Slack, or email.

5

Publish.

Watch the first few executions in the **Executions** tab.

⚠ WARNING

Emails can contain prompt injection, meaning text written to manipulate the AI. In this workflow the AI only produces a summary for *you*. Don't give an email-reading AI agent tools that send, delete, or pay without a human approval step.

13.4 Project 4: A tiny API

1

Add a Webhook node.

Method `POST`, path `greet`, and set *Respond* to **Using 'Respond to Webhook' node**.

2

Add an Edit Fields (Set) node.

Create `message` = `Hello, {{ $json.body.name }}`!

3

Add a Respond to Webhook node.

Respond with the first incoming item as JSON.

4

Test it.

Click *Listen for test event*, then from a terminal:

TERMINAL

```
curl -X POST <test-url> -H "Content-Type: application/json" -d '{"name":"Dylan"}'
```

5

Publish and use the production URL. Add header authentication on the Webhook node before you share it.

13.5 Project 5: Error alerts

1

Create a new workflow called “Error handler.”

- 2 **Add an Error Trigger node.**
- 3 **Add a notification node** with: `Workflow {{ $json.workflow.name }} failed: {{ $json.execution.error.message }}` and the link `{{ $json.execution.url }}`.
- 4 **Save it.**
It doesn't need publishing to act as an error workflow.
- 5 **In each important workflow**, open **Workflow settings** > **Error workflow** and choose "Error handler."

Node and Expression Reference

A desk reference for the nodes and expressions you'll use most. Node names change a little between versions. Search the nodes panel if one isn't exactly where this book says.

IN THIS CHAPTER

- Core nodes by job
- Expression variables
- Common data-transformation snippets
- Cron cheat sheet

14.1 Core nodes

NODE	JOB
Manual Trigger	Run by clicking, for testing
Schedule Trigger	Run on an interval or cron schedule
Webhook	Run when a URL receives a request
n8n Form Trigger	Hosted form that starts a workflow
Chat Trigger	Chat window that starts a workflow (often with an AI Agent)
Error Trigger	Runs when another workflow fails
Execute Sub-workflow / Execute Sub-workflow Trigger	Call one workflow from another
HTTP Request	Call any web API
Code	Run JavaScript (or Python) on items
Edit Fields (Set)	Add, rename, or set fields
If	Split into true and false branches

Switch	Route to many branches by rules
Filter	Drop items that don't match
Merge	Combine two inputs (append, by key, by position...)
Split Out	Turn a list inside one item into many items
Aggregate	Turn many items into one item holding lists
Summarize	Count, sum, or average, grouped by field
Loop Over Items	Process items in batches
Wait	Pause for a time or until a webhook call
Sort / Limit / Remove Duplicates	Tidy lists
Date & Time	Format and calculate dates
Respond to Webhook	Send a custom reply to a webhook caller
Send Email	Send mail through SMTP
AI Agent	An LLM that can use tools you attach
Basic LLM Chain	A single prompt → response step

14.2 Expression variables

Expressions go inside `{{ }}`.

EXPRESSION	MEANING
<code>\$json.field</code>	A field on the current item
<code>\$json["field with spaces"]</code>	A field with spaces in its name
<code>\$('Node Name').item.json.field</code>	The matching item from an earlier node
<code>\$('Node Name').first().json</code>	The first item from an earlier node
<code>\$input.all()</code>	All input items (Code node)
<code>\$input.first()</code>	The first input item
<code>\$now / \$today</code>	Current date-time / today at midnight (Luxon)
<code>\$now.toFormat('yyyy-MM-dd')</code>	Format a date
<code>\$now.minus({ days: 7 })</code>	Date arithmetic
<code>\$execution.id</code>	This run's ID
<code>\$workflow.name</code>	This workflow's name
<code>\$vars.NAME</code>	A workspace variable (on plans that have variables)
<code>\$env.NAME</code>	An environment variable (self-hosted, if allowed)
<code>{{ \$json.price * 1.2 }}</code>	Any JavaScript expression
<code>{{ \$json.name ?? 'unknown' }}</code>	Fallback when a value is missing

14.3 Code node snippets

ADD A FIELD TO EVERY ITEM

```
return $input.all().map(i => ({ json: { ...i.json, total: i.json.qty * i.json.price } }));
```

KEEP ONLY ITEMS WITH AN EMAIL

```
return $input.all().filter(i => i.json.email);
```

GROUP BY A FIELD

```
const groups = {};  
for (const i of $input.all()) (groups[i.json.team] ??= []).push(i.json.name);  
return Object.entries(groups).map([team, names] => ({ json: { team, names } }));
```

14.4 Cron cheat sheet

Fields are: minute, hour, day of month, month, day of week.

EXPRESSION	RUNS
0 7 * * *	Every day at 07:00
*/15 * * * *	Every 15 minutes
0 9 * * 1-5	Weekdays at 09:00
0 0 1 * *	First day of every month at midnight
30 18 * * 0	Sundays at 18:30

Troubleshooting and FAQ

PROBLEM	CAUSE	FIX
Workflow never runs on schedule	It isn't published	Publish it. Check the time zone setting
Webhook returns 404	Using the test URL while not listening, or the production URL while not published	Use the right URL for the state
"Referenced node is unexecuted"	An expression points to a node that didn't run in this branch	Run the earlier node, or guard with <code>?.</code>
Only the first item is processed	A node is set to "Execute once"	Turn off Execute once in the node settings
Credentials error	Expired OAuth or wrong scope	Reconnect the credential. Check its scopes
Rate-limit errors (429)	Calling an API too fast	Use Loop Over Items with a Wait. Enable <i>Retry on fail</i>
Code node "cannot read properties of undefined"	A field is missing on some items	Use <code>?.</code> and <code>??</code> defaults
Self-hosted data lost after restart	No persistent volume	Mount a volume for <code>/home/node/.n8n</code>
Webhooks unreachable when self-hosted	No public URL	Set <code>WEBHOOK_URL</code> and use a reverse proxy or tunnel

Do I need to know how to code? No. Code nodes help, and AI can write them for you.

Cloud or self-hosted? Cloud is easiest and handles updates and uptime. Self-hosting is cheaper at scale and keeps data on your server, but you handle backups, security, and updates.

How do I back up workflows? Download them as JSON (from the workflow menu), or use the n8n CLI `export:workflow` command on self-hosted instances. Keep the files in a private Git repository.

Can n8n run AI agents unattended? Yes. Give agents only the tools they need, add human approval for risky actions, and set error workflows. See Vol. 8 for autonomy settings.

Self-Hosting Environment Variable Reference

If you self-host n8n, you configure it with **environment variables**. This chapter lists the documented variables for the areas beginners touch most. It was generated from the official n8n docs (the *Environment variables* section of docs.n8n.io, source at github.com/n8n-io/n8n-docs) in October 2026. Defaults and names can change between versions, so check the docs for your version before relying on a value.

IN THIS CHAPTER

- How to set variables (Docker and Docker Compose)
- Variables by category, with type, default, and meaning

16.1 Setting variables

DOCKER-COMPOSE.YML (EXCERPT)

```
services:
  n8n:
    image: docker.n8n.io/n8nio/n8n
    environment:
      - GENERIC_TIMEZONE=America/Indiana/Indianapolis
      - TZ=America/Indiana/Indianapolis
      - WEBHOOK_URL=https://n8n.example.com/
    volumes:
      - n8n_data:/home/node/.n8n
volumes:
  n8n_data:
```

16.2 Deployment

VARIABLE	TYPE	DEFAULT	MEANING
<code>HTTP_PROXY</code>	String	-	A URL to proxy unencrypted HTTP requests through.
<code>HTTPS_PROXY</code>	String	-	A URL to proxy TLS/SSL encrypted HTTP requests through.
<code>ALL_PROXY</code>	String	-	A URL to proxy both unencrypted and encrypted HTTP requests through.
<code>NO_PROXY</code>	String	-	A comma-separated list of hostnames or URLs that should bypass the proxy.
<code>N8N_ENFORCE_GLOBAL_USER_AGENT</code>	Boolean	<code>false</code>	When set to <code>true</code> , n8n replaces the default bare <code>n8n</code> User-Agent string with an RFC-compliant value (<code>'Mozilla/5.0...</code>
<code>N8N_GLOBAL_USER_AGENT_VALUE</code>	String	-	A custom User-Agent string to use for all outbound HTTP requests.
<code>N8N_EDITOR_BASE_URL</code>	String	-	Public URL where users can access the editor.
<code>N8N_DISABLE_UI</code>	Boolean	<code>false</code>	Set to <code>true</code> to disable the UI.
<code>N8N_PREVIEW_MODE</code>	Boolean	<code>false</code>	Set to <code>true</code> to run in preview mode.
<code>N8N_CANVAS_ONLY</code>	Boolean	<code>false</code>	Set to <code>true</code> to run n8n in canvas-only mode, hiding the header, sidebar, and overlay elements when viewing an individual...
<code>N8N_TEMPLATES_ENABLED</code>	Boolean	<code>true</code>	Enables workflow templates[^1] (true) or disable (false).
<code>N8N_TEMPLATES_HOST</code>	String	<code>https://api.n8n.io</code>	Change this if creating your own workflow template library.
<code>N8N_ENCRYPTION_KEY</code>	String	Random key generated by n8n	Provide a custom key used to encrypt credentials in the n8n database.
<code>N8N_ENV_FEAT_ENCRYPTION_KEY_ROTATION</code>	Boolean	<code>false</code>	Set to <code>true</code> on all instances (main and workers) to enable encryption key rotation.
<code>N8N_ENV_FEAT_OAUTH2_JWE</code>	Boolean	<code>false</code>	Set to <code>true</code> on all instances (main and workers) to enable JWE token decryption for OAuth 2.0 credentials.
<code>N8N_ENV_FEAT_TOKEN_EXCHANGE</code>	Boolean	<code>false</code>	Set to <code>true</code> to enable token exchange for embedding partners.
<code>N8N_OAUTH_JWE_JWKS_PER_MINUTE</code>	Number	<code>60</code>	Per-IP rate limit on the public JWKS endpoint (<code>/rest/.well-known/jwks.json</code>) used by JWE token decryption for OAuth 2.0...
<code>N8N_MCP_BASE_URL</code>	String	-	Public base URL where MCP clients reach the instance-level MCP server, when it differs from the instance base URL. Useful for...

<code>N8N_MCP_SERVER_RATE_LIMIT</code>	Number	<code>100</code>	Maximum number of requests per IP, per 5 minutes, to the MCP server endpoint (<code>/mcp-server/http</code>).
<code>N8N_OAUTH_SERVER_REGISTER_RATE_LIMIT</code>	Number	<code>10</code>	Maximum number of requests per IP, per 5 minutes, to the OAuth server client registration endpoint (<code>/oauth/register</code> and...
<code>N8N_OAUTH_SERVER_AUTHORIZE_RATE_LIMIT</code>	Number	<code>50</code>	Maximum number of requests per IP, per 5 minutes, to the OAuth server authorization endpoint (<code>/oauth/authorize</code> and...
<code>N8N_OAUTH_SERVER_TOKEN_RATE_LIMIT</code>	Number	<code>20</code>	Maximum number of requests per IP, per 5 minutes, to the OAuth server token endpoint (<code>/oauth/token</code> and <code>/mcp-oauth/token</code>).
<code>N8N_OAUTH_SERVER_REVOKE_RATE_LIMIT</code>	Number	<code>30</code>	Maximum number of requests per IP, per 5 minutes, to the OAuth server token revocation endpoint (<code>/oauth/revoke</code> and...
<code>N8N_USER_FOLDER</code>	String	<code>user-folder</code>	Provide the path where n8n will create the <code>.n8n</code> folder.
<code>N8N_PATH</code>	String	<code>/</code>	The path n8n deploys to.
<code>N8N_HOST</code>	String	<code>localhost</code>	Host name n8n runs on.
<code>N8N_PORT</code>	Number	<code>5678</code>	The HTTP port n8n runs on.
<code>N8N_LISTEN_ADDRESS</code>	String	<code>::</code>	The IP address n8n should listen on.
<code>N8N_PROTOCOL</code>	Enum string: <code>http</code> , <code>https</code>	<code>http</code>	The protocol used to reach n8n.
<code>N8N_SSL_KEY</code>	String	-	The SSL key for HTTPS protocol.
<code>N8N_SSL_CERT</code>	String	-	The SSL certificate for HTTPS protocol.
<code>N8N_PERSONALIZATION_ENABLED</code>	Boolean	<code>true</code>	Whether to ask users personalisation questions and then customise n8n accordingly.
<code>N8N_VERSION_NOTIFICATIONS_ENABLED</code>	Boolean	<code>true</code>	When enabled, n8n sends notifications of new versions and security updates.
<code>N8N_VERSION_NOTIFICATIONS_ENDPOINT</code>	String	<code>'https://api.n8n.io/versi...</code>	The endpoint to retrieve where version information.

<code>N8N_VERSION_NOTIFICATIONS_INFO_URL</code>	String	<code>'https://docs.n8n.io/gett...</code>	The URL displayed in the New Versions panel for more information.
<code>N8N_DIAGNOSTICS_ENABLED</code>	Boolean	<code>true</code>	Whether to share selected, anonymous telemetry with n8n.
<code>N8N_DIAGNOSTICS_CONFIG_FRONTEND</code>	String	<code>'1zPn9bgWPzIQc0p8Gj1uiK6D...</code>	Telemetry configuration for the frontend.
<code>N8N_DIAGNOSTICS_CONFIG_BACKEND</code>	String	<code>'1zPn7YoGC3ZXE9zLeTKLuQCB...</code>	Telemetry configuration for the backend.
<code>N8N_PUSH_BACKEND</code>	String	<code>websocket</code>	Choose whether the n8n backend uses server-sent events (<code>sse</code>) or WebSockets (<code>websocket</code>) to send changes to the UI.
<code>VUE_APP_URL_BASE_API</code>	String	<code>http://localhost:5678/</code>	Used when building the <code>n8n-editor-ui</code> package manually to set how the frontend can reach the backend API. Refer to Configure...
<code>N8N_HIRING_BANNER_ENABLED</code>	Boolean	<code>true</code>	Whether to show the n8n hiring banner in the console (true) or not (false).
<code>N8N_PUBLIC_API_SWAGGERUI_DISABLED</code>	Boolean	<code>false</code>	Whether the Swagger UI (API playground) is disabled (true) or not (false).
<code>N8N_PUBLIC_API_DISABLED</code>	Boolean	<code>false</code>	Whether to disable the public API (true) or not (false).
<code>N8N_PUBLIC_API_ENDPOINT</code>	String	<code>api</code>	Path for the public API endpoints.
<code>N8N_GRACEFUL_SHUTDOWN_TIMEOUT</code>	Number	<code>30</code>	How long should the n8n process wait (in seconds) for components to shut down before exiting the process.
<code>N8N_DEV_RELOAD</code>	Boolean	<code>false</code>	When working on the n8n source code, set this to <code>true</code> to automatically reload or restart the application when changes occur...
<code>N8N_REINSTALL_MISSING_PACKAGES</code>	Boolean	<code>false</code>	If set to <code>true</code> , n8n will automatically attempt to reinstall any missing packages.
<code>N8N_TUNNEL_SUBDOMAIN</code>	String	<code>-</code>	Specifies the subdomain for the n8n tunnel.
<code>N8N_PROXY_HOPS</code>	Number	<code>0</code>	Number of reverse-proxies n8n is running behind.

16.3 Endpoints

VARIABLE	TYPE	DEFAULT	MEANING
<code>N8N_PAYLOAD_SIZE_MAX</code>	Number	16	The maximum payload size in MiB. The default is 16 MiB, but this can be increased by adjusting the variable to a higher value.
<code>N8N_FORMDATA_FILE_SIZE_MAX</code>	Number	200	Max payload size for files in form-data webhook payloads in MiB.
<code>N8N_METRICS</code>	Boolean	false	Whether to enable the <code>/metrics</code> endpoint.
<code>N8N_METRICS_PREFIX</code>	String	<code>n8n_</code>	Optional prefix for n8n specific metrics names.
<code>N8N_METRICS_INCLUDE_DEFAULT_METRICS</code>	Boolean	true	Whether to expose default system and node.js metrics.
<code>N8N_METRICS_INCLUDE_CACHE_METRICS</code>	Boolean	false	Whether to include metrics (true) for cache hits and misses, or not include them (false).
<code>N8N_METRICS_INCLUDE_MESSAGE_EVENT_BUS_METRICS</code>	Boolean	false	Whether to include metrics (true) for events, or not include them (false).
<code>N8N_METRICS_INCLUDE_WORKFLOW_ID_LABEL</code>	Boolean	false	Whether to include a label for the workflow ID on workflow metrics.
<code>N8N_METRICS_INCLUDE_NODE_TYPE_LABEL</code>	Boolean	false	Whether to include a label for the node type on node metrics.
<code>N8N_METRICS_INCLUDE_CREDENTIAL_TYPE_LABEL</code>	Boolean	false	Whether to include a label for the credential type on credential metrics.
<code>N8N_METRICS_INCLUDE_API_ENDPOINTS</code>	Boolean	false	Whether to expose metrics for API endpoints.
<code>N8N_METRICS_INCLUDE_API_PATH_LABEL</code>	Boolean	false	Whether to include a label for the path of API invocations.
<code>N8N_METRICS_INCLUDE_API_METHOD_LABEL</code>	Boolean	false	Whether to include a label for the HTTP method (GET, POST, ...) of API invocations.
<code>N8N_METRICS_INCLUDE_API_STATUS_CODE_LABEL</code>	Boolean	false	Whether to include a label for the HTTP status code (200, 404, ...) of API invocations.
<code>N8N_METRICS_INCLUDE_QUEUE_METRICS</code>	Boolean	false	Whether to include metrics for jobs in scaling mode.
<code>N8N_METRICS_QUEUE_METRICS_INTERVAL</code>	Integer	20	How often (in seconds) to update queue metrics.
<code>N8N_METRICS_INCLUDE_SCHEDULER_METRICS</code>	Boolean	false	Whether to include metrics for the durable scheduler, such as queue depth, scheduling lag, dispatch throughput, retries, and...
<code>N8N_METRICS_SCHEDULER_INTERVAL</code>	Integer	20	How often (in seconds) to refresh the queue-health gauges for the durable scheduler.
<code>N8N_METRICS_INCLUDE_POLL_TRIGGER_METRICS</code>	Boolean	false	Whether to include metrics for trigger nodes that poll, such as poll duration, poll errors, overlapping polls, and cursor commits.
<code>N8N_METRICS_INCLUDE_SYSTEM_TASK_METRICS</code>	Boolean	false	Whether to include metrics for n8n's internal maintenance tasks, such as run duration, last success, skipped occurrences, and...
<code>N8N_METRICS_INCLUDE_SSRF_METRICS</code>	Boolean	false	Whether to include metrics for SSRF protection checks.

<code>N8N_METRICS_INCLUDE_DNS_CACHE_METRICS</code>	Boolean	<code>false</code>	Whether to include metrics for the DNS cache (currently only used by SSRF protection).
<code>N8N_ENDPOINT_REST</code>	String	<code>rest</code>	The path used for REST endpoint.
<code>N8N_ENDPOINT_WEBHOOK</code>	String	<code>webhook</code>	The path used for webhook endpoint.
<code>N8N_ENDPOINT_WEBHOOK_TEST</code>	String	<code>webhook-test</code>	The path used for test-webhook endpoint.
<code>N8N_ENDPOINT_WEBHOOK_WAIT</code>	String	<code>webhook-waiting</code>	The path used for waiting-webhook endpoint.
<code>N8N_ENDPOINT_HEALTH</code>	String	<code>healthz</code>	The path used for health check endpoint.
<code>N8N_WEBHOOK_URL</code>	String	-	Sets the base URL for both test and production webhooks when running n8n behind a reverse proxy.
<code>WEBHOOK_URL</code>	String	-	Deprecated from n8n 2.35.0; alias of <code>N8N_WEBHOOK_URL</code> .
<code>N8N_DISABLE_PRODUCTION_MAIN_PROCESS</code>	Boolean	<code>false</code>	Disable production webhooks from main process.

16.4 Executions

VARIABLE	TYPE	DEFAULT	MEANING
<code>EXECUTIONS_MODE</code>	Enum string: <code>regular</code> , <code>queue</code>	<code>regular</code>	Whether executions should run directly or using queue. Refer to Queue mode for more details.
<code>EXECUTIONS_TIMEOUT</code>	Number	<code>-1</code>	Sets a default timeout (in seconds) to all workflows after which n8n stops their execution.
<code>EXECUTIONS_TIMEOUT_MAX</code>	Number	<code>3600</code>	The maximum execution time (in seconds) that users can set for an individual workflow.
<code>N8N_AI_TIMEOUT_MAX</code>	Number	<code>3600000</code>	Sets the HTTP request timeout in milliseconds for AI and LLM nodes (such as OpenAI, Anthropic, Mistral, and Ollama).
<code>EXECUTIONS_DATA_SAVE_ON_ERROR</code>	Enum string: <code>all</code> , <code>none</code>	<code>all</code>	Whether n8n saves execution data on error.
<code>EXECUTIONS_DATA_SAVE_ON_SUCCESS</code>	Enum string: <code>all</code> , <code>none</code>	<code>all</code>	Whether n8n saves execution data on success.
<code>EXECUTIONS_DATA_SAVE_ON_PROGRESS</code>	Boolean	<code>false</code>	Whether to save progress for each node executed (true) or not (false).
<code>EXECUTIONS_DATA_SAVE_MANUAL_EXECUTIONS</code>	Boolean	<code>true</code>	Whether to save data of executions when started manually.

<code>N8N_EXECUTION_DATA_STORAGE_MODE</code>	Enum string: database, filesystem, s3, azure	database	Where n8n stores execution data.
<code>N8N_STORAGE_PATH</code>	String	<code>N8N_USER_FOLDER/storage</code>	Base path for filesystem storage.
<code>EXECUTIONS_DATA_PRUNE</code>	Boolean	true	Whether to delete data of past executions on a rolling basis.
<code>EXECUTIONS_DATA_MAX_AGE</code>	Number	336	The execution age (in hours) before it's deleted.
<code>EXECUTIONS_DATA_PRUNE_MAX_COUNT</code>	Number	10000	Maximum number of executions to keep in the database. 0 = no limit.
<code>EXECUTIONS_DATA_HARD_DELETE_BUFFER</code>	Number	1	How old (hours) the finished execution data has to be to get hard-deleted.
<code>EXECUTIONS_DATA_PRUNE_HARD_DELETE_INTERVAL</code>	Number	15	How often (minutes) execution data should be hard-deleted.
<code>EXECUTIONS_DATA_PRUNE_SOFT_DELETE_INTERVAL</code>	Number	60	How often (minutes) execution data should be soft-deleted.
<code>EXECUTIONS_DATA_MAX_DISPLAY_SIZE</code>	Number	104857600	Maximum size (in bytes) of execution data that n8n loads when displaying an execution (editor, execution detail, and public API).
<code>N8N_CONCURRENCY_PRODUCTION_LIMIT</code>	Number	-1	Max production executions allowed to run concurrently, in both regular and scaling modes. -1 to disable in regular mode.
<code>N8N_CONCURRENCY_EVALUATION_LIMIT</code>	Number	License-tier default	Max test cases that can run in parallel within a single evaluation test run.

<code>N8N_WORKFLOW_AUTODEACTIVATION_ENABLED</code>	Boolean	<code>false</code>	Whether workflows are automatically unpublished after repeated crashed executions.
<code>N8N_WORKFLOW_AUTODEACTIVATION_MAX_LAST_EXECUTIONS</code>	Number	<code>3</code>	Number of crashed executions before unpublishing a workflow.

16.5 Security

VARIABLE	TYPE	DEFAULT	MEANING
<code>N8N_BLOCK_ENV_ACCESS_IN_NODE</code>	Boolean	<code>false</code>	Whether to allow users to access environment variables in expressions and the Code node (false) or not (true).
<code>N8N_BLOCK_FILE_ACCESS_TO_N8N_FILES</code>	Boolean	<code>true</code>	Set to <code>true</code> to block access to all files in the <code>.n8n</code> directory and user defined configuration files.
<code>N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS</code>	Boolean	<code>false</code>	Set to <code>true</code> to try to set 0600 permissions for the settings file, giving only the owner read and write access.
<code>N8N_RESTRICT_FILE_ACCESS_TO</code>	String		Limits access to files in these directories.
<code>N8N_SECURITY_AUDIT_DAYS_ABANDONED_WORKFLOW</code>	Number	<code>90</code>	Number of days to consider a workflow abandoned if it's not executed.
<code>N8N_CONTENT_SECURITY_POLICY</code>	String	<code>'script-src 'stri...</code>	The Content-Security-Policy n8n serves on its own pages.
<code>N8N_CONTENT_SECURITY_POLICY_REPORT_ONLY</code>	String		The policy n8n serves as <code>Content-Security-Policy-Report-Only</code> , in the same formats as <code>N8N_CONTENT_SECURITY_POLICY</code> .
<code>N8N_SECURE_COOKIE</code>	Boolean	<code>true</code>	Ensures that cookies are only sent over HTTPS, enhancing security.
<code>N8N_SAMESITE_COOKIE</code>	Enum string: <code>strict</code> , <code>lax</code> , <code>none</code>	<code>lax</code>	Controls cross-site cookie behavior (learn more): <code>strict</code> : Sent only for first-party requests. <code>lax</code> (default): Sent with...
<code>N8N_GIT_NODE_DISABLE_BARE_REPOS</code>	Boolean	<code>false</code>	Set to <code>true</code> to prevent the Git node from working with bare repositories, enhancing security.
<code>N8N_GIT_NODE_ENABLE_HOOKS</code>	Boolean	<code>false</code>	Set to <code>true</code> to allow the Git node to execute Git hooks.
<code>N8N_POSTMESSAGE_ALLOWED_ORIGINS</code>	String		Origins allowed to exchange <code>postMessage</code> commands with the editor when it's embedded in another page.

16.6 Database

VARIABLE	TYPE	DEFAULT	MEANING
<code>DB_TYPE</code> <code>/_FILE</code>	Enum string: <code>sqlite</code> , <code>postgresdb</code>	<code>sqlite</code>	The database to use.
<code>DB_TABLE_PREFIX</code>	*	-	Prefix to use for table names.
<code>DB_PING_INTERVAL_SECONDS</code>	Number	<code>2</code>	How often, in seconds, n8n pings the database to check that the connection is still alive.
<code>DB_PING_TIMEOUT_MS</code>		Number <code>5000</code>	How long, in milliseconds, n8n waits for a single ping to respond before counting it as a failure.
<code>DB_PING_MAX_FAILURES_BEFORE_RECOVERY</code>		Number <code>3</code>	How many pings in a row must fail before n8n treats the connection as lost and starts recovery.
<code>DB_RECOVERY_BACKOFF_MIN_MS</code>		Number <code>1000</code>	How long, in milliseconds, n8n waits before its first recovery attempt.
<code>DB_RECOVERY_BACKOFF_MAX_MS</code>		Number <code>30000</code>	The longest, in milliseconds, n8n waits between recovery attempts.
<code>DB_CONNECTION_ACQUISITION_TIMEOUT_MS</code>		Number <code>30000</code>	How long, in milliseconds, a query waits while recovery is in progress before failing fast with an error.
<code>DB_POSTGRESDB_DATABASE</code> <code>/_FILE</code>		String <code>n8n</code>	The name of the PostgreSQL database.
<code>DB_POSTGRESDB_HOST</code> <code>/_FILE</code>		String <code>localhost</code>	The PostgreSQL host.
<code>DB_POSTGRESDB_PORT</code> <code>/_FILE</code>		Number <code>5432</code>	The PostgreSQL port.
<code>DB_POSTGRESDB_USER</code> <code>/_FILE</code>		String <code>postgres</code>	The PostgreSQL user.
<code>DB_POSTGRESDB_PASSWORD</code> <code>/_FILE</code>		String -	The PostgreSQL password.
<code>DB_POSTGRESDB_POOL_SIZE</code> <code>/_FILE</code>		Number <code>2</code>	Control how many parallel open Postgres connections n8n should have.
<code>DB_POSTGRESDB_MAX_CONNECTION_LIFETIME_MS</code> <code>/_FILE</code>		Number <code>3600000</code>	How long, in milliseconds, n8n keeps a pooled connection before recycling it.
<code>DB_POSTGRESDB_KEEP_ALIVE</code> <code>/_FILE</code>		Boolean <code>true</code>	Whether to enable TCP keep-alive on connections.
<code>DB_POSTGRESDB_KEEP_ALIVE_INITIAL_DELAY_MS</code> <code>/_FILE</code>		Number <code>10000</code>	How long, in milliseconds, n8n waits before sending the first TCP keep-alive probe on a connection.
<code>DB_POSTGRESDB_SCHEMA</code> <code>/_FILE</code>		String <code>public</code>	The PostgreSQL schema.

<code>DB_POSTGRESDB_SSL_ENABLED</code> <code>/_FILE</code>	Boolean	<code>false</code>	Whether to enable SSL. Automatically enabled if <code>DB_POSTGRESDB_SSL_CA</code> , <code>DB_POSTGRESDB_SSL_CERT</code> or <code>DB_POSTGRESDB_SSL_KEY</code> is...
<code>DB_POSTGRESDB_SSL_CA</code> <code>/_FILE</code>	String	-	The PostgreSQL SSL certificate authority.
<code>DB_POSTGRESDB_SSL_CERT</code> <code>/_FILE</code>	String	-	The PostgreSQL SSL certificate.
<code>DB_POSTGRESDB_SSL_KEY</code> <code>/_FILE</code>	String	-	The PostgreSQL SSL key.
<code>DB_POSTGRESDB_SSL_REJECT_UNAUTHORIZED</code> <code>/_FILE</code>	Boolean	<code>true</code>	If n8n should reject unauthorized SSL connections (true) or not (false).
<code>DB_SQLITE_POOL_SIZE</code>	Number	<code>0</code>	Controls whether to open the SQLite file in WAL mode or rollback journal mode.
<code>DB_SQLITE_VACUUM_ON_STARTUP</code>	Boolean	<code>false</code>	Runs VACUUM operation on startup to rebuild the database.

16.7 Nodes

VARIABLE	TYPE	DEFAULT	MEANING
<code>N8N_COMMUNITY_PACKAGES_AUTH_TOKEN</code>	String	-	Authentication token for a private npm registry.
<code>N8N_COMMUNITY_PACKAGES_ENABLED</code>	Boolean	<code>true</code>	Enables (true) or disables (false) the functionality to install and load community nodes.
<code>N8N_COMMUNITY_PACKAGES_PREVENT_LOADING</code>	Boolean	<code>false</code>	Prevents (true) or allows (false) loading installed community nodes on instance startup.
<code>N8N_COMMUNITY_PACKAGES_REGISTRY</code>	String	<code>'https://registry.npmjs.o...</code>	NPM registry URL to pull community packages from (license required).
<code>N8N_CUSTOM_EXTENSIONS</code>	String	-	Specify the path to directories containing your custom nodes.
<code>N8N_PYTHON_ENABLED</code>	Boolean	<code>true</code>	Whether to enable Python execution on the Code node.
<code>N8N_UNVERIFIED_PACKAGES_ENABLED</code>	Boolean	<code>true</code>	When <code>N8N_COMMUNITY_PACKAGES_ENABLED</code> is true, this variable controls whether to enable the installation and use of unverified...
<code>N8N_VERIFIED_PACKAGES_ENABLED</code>	Boolean	<code>true</code>	When <code>N8N_COMMUNITY_PACKAGES_ENABLED</code> is true, this variable controls whether to show verified community nodes in the nodes...
<code>NODE_FUNCTION_ALLOW_BUILTIN</code>	String	-	Permit users to import specific built-in modules in the Code node.
<code>NODE_FUNCTION_ALLOW_EXTERNAL</code>	String	-	Permit users to import specific external modules (from <code>n8n/node_modules</code>) in the Code node. n8n disables importing modules by...
<code>NODES_ERROR_TRIGGER_TYPE</code>	String	<code>'n8n-nodes-base.errorTrig...</code>	Specify which node type to use as Error Trigger.

<code>NODES_EXCLUDE</code>	Array of strings	<code>["n8n-nodes-base.execut...</code>	Specify which nodes not to load.
<code>NODES_INCLUDE</code>	Array of strings	-	Specify which nodes to load.
<code>NODES_MERGE_SQL_SANDBOX_MEMORY_LIMIT_MB</code>	Number	<code>64</code>	Memory limit in MB for the Merge node's SQL sandbox.
<code>N8N_COMPRESSION_NODE_MAX_DECOMPRESSED_SIZE_BYTES</code>	Number	<code>268435456</code>	Maximum total decompressed output size in bytes.
<code>N8N_COMPRESSION_NODE_MAX_ZIP_ENTRIES</code>	Number	<code>1000</code>	Maximum number of entries allowed in a ZIP archive.

16.8 Logs

VARIABLE	TYPE	DEFAULT	MEANING
<code>N8N_LOG_LEVEL</code>	Enum string: <code>info</code> , <code>warn</code> , <code>error</code> , <code>debug</code>	<code>info</code>	Log output level.
<code>N8N_LOG_OUTPUT</code>	Enum string: <code>console</code> , <code>file</code>	<code>console</code>	Where to output logs.
<code>N8N_LOG_FORMAT</code>	Enum string: <code>text</code> , <code>json</code>	<code>text</code>	The log format to use. <code>text</code> prints human readable messages. <code>json</code> prints one JSON object per line containing the message,...
<code>N8N_LOG_CRON_ACTIVE_INTERVAL</code>	Number	<code>0</code>	Interval in minutes to log currently active cron jobs.
<code>N8N_LOG_FILE_COUNT_MAX</code>	Number	<code>100</code>	Max number of log files to keep.
<code>N8N_LOG_FILE_SIZE_MAX</code>	Number	<code>16</code>	Max size of each log file in MB.
<code>N8N_LOG_FILE_LOCATION</code>	String	<code>"/log...</code>	Log file location.
<code>DB_LOGGING_ENABLED</code>	Boolean	<code>false</code>	Whether to enable database-specific logging.
<code>DB_LOGGING_OPTIONS</code>	Enum string: <code>query</code> , <code>error</code> , <code>schema</code> , <code>warn</code> , <code>info</code> , <code>log</code>	<code>error</code>	Database log output level.
<code>DB_LOGGING_MAX_EXECUTION_TIME</code>	Number	<code>1000</code>	Maximum execution time (in milliseconds) before n8n logs a warning.
<code>CODE_ENABLE_STDOUT</code>	Boolean	<code>false</code>	Set to <code>true</code> to send Code node logs from <code>console.log</code> or <code>print</code> to the process's stdout, only for production executions.
<code>NO_COLOR</code>	any	<code>undefined</code>	Set to any value to output logs without ANSI colors.
<code>N8N_EVENTBUS_CHECKUNSENTINTERVAL</code>	Number	<code>0</code>	How often (in milliseconds) to check for unsent event messages.

<code>N8N_EVENTBUS_LOGWRITER_SYNCFILEACCESS</code>	Boolean	<code>false</code>	Whether all file access happens synchronously within the thread (true) or not (false).
<code>N8N_EVENTBUS_LOGWRITER_KEEPCOUNT</code>	Number	<code>3</code>	Number of event log files to keep.
<code>N8N_EVENTBUS_LOGWRITER_MAXFILESIZEINKB</code>	Number	<code>10240</code>	Maximum size (in kilo-bytes) of an event log file before a new one starts.
<code>N8N_EVENTBUS_LOGWRITER_LOGBASENAME</code>	String	<code>n8nEventLog</code>	Basename of the event log file.
<code>N8N_EVENTBUS_LOGWRITER_LOGFULLPATH</code>	String	<code>''</code>	Absolute path to the event log file.
<code>N8N_EVENTBUS_LOGWRITER_MAXTOTALMESSAGESPERFILE</code>	Number	<code>500000</code>	Maximum number of lines parsed from a single event log file during recovery.

16.9 Timezone and Localization

VARIABLE	TYPE	DEFAULT	MEANING
<code>GENERIC_TIMEZONE</code>	*	<code>America/New_York</code>	The n8n instance timezone.
<code>N8N_DEFAULT_LOCALE</code>	String	<code>en</code>	A locale identifier, compatible with the Accept-Language header. n8n doesn't support regional identifiers, such as <code>de-AT</code> .

16.10 Workflows

VARIABLE	TYPE	DEFAULT	MEANING
<code>N8N_ONBOARDING_FLOW_DISABLED</code>	Boolean	<code>false</code>	Whether to disable onboarding tips when creating a new workflow (true) or not (false).
<code>N8N_WORKFLOW_CALLER_POLICY_DEFAULT_OPTION</code>	String	<code>workflowsFromSameOwner</code>	Which workflows can call a workflow.
<code>N8N_WORKFLOW_TAGS_DISABLED</code>	Boolean	<code>false</code>	Whether to disable workflow tags (true) or enable tags (false).
<code>WORKFLOWS_DEFAULT_NAME</code>	String	<code>My workflow</code>	The default name used for new workflows.

16.11 Credentials

VARIABLE	TYPE	DEFAULT	MEANING
<code>CREDENTIALS_OVERWRITE_DATA</code> <code>/_FILE</code>	*	-	Overwrites for credentials.
<code>CREDENTIALS_OVERWRITE_ENDPOINT</code>	String	-	The API endpoint to fetch credentials.
<code>CREDENTIALS_OVERWRITE_PERSISTENCE</code>	Boolean	<code>false</code>	Enable database persistence for credential overwrites.
<code>N8N_MANAGED_OAUTH_SHOW_SCOPES</code>	String	-	Comma-separated list of managed OAuth credential types for which users can configure scope fields.
<code>CREDENTIALS_DEFAULT_NAME</code>	String	<code>My credentials</code>	The default name for credentials.

16.12 Binary Data

VARIABLE	TYPE	DEFAULT	MEANING
<code>N8N_AVAILABLE_BINARY_DATA_MODES</code>	String	<code>filesystem</code>	A comma separated list of available binary data modes.
<code>N8N_BINARY_DATA_DATABASE_MAX_FILE_SIZE</code>	Number	<code>512</code>	Maximum size (in MiB) of a single file n8n stores when <code>N8N_DEFAULT_BINARY_DATA_MODE</code> is <code>database</code> .
<code>N8N_BINARY_DATA_STORAGE_PATH</code>	String	<code>`N8N_USER_FOLDER/binaryDa...</code>	The path where n8n stores binary data.
<code>N8N_DEFAULT_BINARY_DATA_MODE</code>	String	<code>default</code>	The default binary data mode. <code>default</code> keeps binary data in memory.

16.13 Queue Mode

VARIABLE	TYPE	DEFAULT	MEANING
<code>OFFLOAD_MANUAL_EXECUTIONS_TO_WORKERS</code>	Boolean	<code>false</code>	Set to <code>true</code> if you want manual executions to run on the worker rather than on main.
<code>QUEUE_BULL_PREFIX</code>	String	-	Prefix to use for all queue keys.
<code>QUEUE_BULL_REDIS_DB</code>	Number	<code>0</code>	The Redis database used.
<code>QUEUE_BULL_REDIS_HOST</code>	String	<code>localhost</code>	The Redis host.
<code>QUEUE_BULL_REDIS_PORT</code>	Number	<code>6379</code>	The Redis port used.
<code>QUEUE_BULL_REDIS_USERNAME</code>	String	-	The Redis username (needs Redis version 6 or above).
<code>QUEUE_BULL_REDIS_PASSWORD</code>	String	-	The Redis password.
<code>QUEUE_BULL_REDIS_TIMEOUT_THRESHOLD</code>	Number	<code>10000</code>	The Redis timeout threshold (in ms).

<code>QUEUE_BULL_REDIS_CLUSTER_NODES</code>	String	-	Expects a comma-separated list of Redis Cluster nodes in the format <code>host:port</code> , for the Redis client to initially connect to.
<code>QUEUE_BULL_REDIS_TLS</code>	Boolean	<code>false</code>	Enable TLS on Redis connections.
<code>QUEUE_BULL_REDIS_DUALSTACK</code>	Boolean	<code>false</code>	Enable dual-stack support (IPv4 and IPv6) on Redis connections.
<code>QUEUE_WORKER_TIMEOUT</code> (deprecated)	Number	<code>30</code>	Deprecated from n8n 1.22.0. Use <code>N8N_GRACEFUL_SHUTDOWN_TIMEOUT</code> instead.
<code>QUEUE_HEALTH_CHECK_ACTIVE</code>	Boolean	<code>false</code>	Whether to enable health checks (true) or disable (false).
<code>QUEUE_HEALTH_CHECK_PORT</code>	Number	<code>5678</code>	The port to serve health checks on.
<code>QUEUE_WORKER_LOCK_DURATION</code>	Number	<code>60000</code>	How long (in ms) is the lease period for a worker to work on a message.
<code>QUEUE_WORKER_LOCK_RENEW_TIME</code>	Number	<code>10000</code>	How frequently (in ms) should a worker renew the lease time.
<code>QUEUE_WORKER_STALLED_INTERVAL</code>	Number	<code>30000</code>	How often should a worker check for stalled jobs (use 0 for never).
<code>QUEUE_WORKER_MAX_STALLED_COUNT</code> (deprecated)	Number	<code>1</code>	Deprecated Removed in n8n 2.0. Setting this has no effect.
<code>N8N_WEBHOOK_RESPONSE_RELAY_SIZE_MAX</code>	Number	<code>64</code>	Maximum size (in MiB) of a response a worker sends back to the main instance inside a queue message.
<code>N8N_WEBHOOK_RESPONSE_RELAY_OFFLOAD_ENABLED</code>	Boolean	<code>false</code>	Whether a worker stores a response body above <code>N8N_WEBHOOK_RESPONSE_RELAY_SIZE_MAX</code> in binary data storage, so the main...
<code>N8N_MULTI_MAIN_SETUP_ENABLED</code>	Boolean	<code>false</code>	Whether to enable multi-main setup for queue mode (license required).
<code>N8N_MULTI_MAIN_SETUP_KEY_TTL</code>	Number	<code>10</code>	Time to live (in seconds) for leader key in multi-main setup.
<code>N8N_MULTI_MAIN_SETUP_CHECK_INTERVAL</code>	Number	<code>3</code>	Interval (in seconds) for leader check in multi-main setup.

16.14 Task Runners

VARIABLE	TYPE	DEFAULT	MEANING
<code>N8N_RUNNERS_ENABLED</code> (deprecated)	Boolean	<code>false</code>	Are task runners enabled. Deprecated from n8n 2.0.
<code>N8N_RUNNERS_MODE</code>	Enum string: <code>internal</code> (deprecated) , <code>external</code>	<code>internal</code>	How to launch and run the task runner. <code>internal</code> means n8n will launch a task runner as child process. Deprecated ...
<code>N8N_RUNNERS_AUTH_TOKEN</code>	String	Random string	Shared secret used by a task runner to authenticate to n8n.
<code>N8N_RUNNERS_BROKER_PORT</code>	Number	<code>5679</code>	Port the task broker listens on for task runner connections.
<code>N8N_RUNNERS_BROKER_LISTEN_ADDRESS</code>	String	<code>127.0.0.1</code>	Address the task broker listens on.

<code>N8N_RUNNERS_MAX_PAYLOAD</code>	Number	<code>1 073 741 824</code>	Maximum payload size in bytes for communication between a task broker and a task runner.
<code>N8N_RUNNERS_MAX_OLD_SPACE_SIZE</code>	String		The <code>--max-old-space-size</code> option to use for a task runner (in MB).
<code>N8N_RUNNERS_MAX_CONCURRENCY</code>	Number	<code>5</code>	The number of concurrent tasks a task runner can execute at a time.
<code>N8N_RUNNERS_TASK_TIMEOUT</code>	Number	<code>300</code>	The maximum time, in seconds, a task can run before the runner stops it and restarts.
<code>N8N_RUNNERS_HEARTBEAT_INTERVAL</code>	Number	<code>30</code>	The interval, in seconds, at which the runner must send a heartbeat to the broker.
<code>N8N_RUNNERS_INSECURE_MODE</code>	Boolean	<code>false</code>	Whether to disable all security measures in the task runner, for compatibility with modules that rely on insecure JS features....
<code>N8N_RUNNERS_TASK_REQUEST_TIMEOUT</code>	Number	<code>60</code>	How long (in seconds) a task request can wait for a runner to become available before timing out.
<code>N8N_RUNNERS_LAUNCHER_LOG_LEVEL</code>	Enum string: <code>debug</code> , <code>info</code> , <code>warn</code> , <code>error</code>	<code>info</code>	Which log messages to show.
<code>N8N_RUNNERS_AUTH_TOKEN</code>	String	-	Shared secret used to authenticate to n8n.
<code>N8N_RUNNERS_AUTO_SHUTDOWN_TIMEOUT</code>	Number	<code>15</code>	The number of seconds to wait before shutting down an idle runner.
<code>N8N_RUNNERS_TASK_BROKER_URI</code>	String	<code>http://127.0.0.1:5679</code>	The URI of the task broker server (n8n instance).
<code>N8N_RUNNERS_LAUNCHER_HEALTH_CHECK_PORT</code>	Number	<code>5680</code>	Port for the launcher's health check server.
<code>N8N_RUNNERS_MAX_PAYLOAD</code>	Number	<code>1 073 741 824</code>	Maximum payload size in bytes for communication between a task broker and a task runner.
<code>N8N_RUNNERS_MAX_CONCURRENCY</code>	Number	<code>5</code>	The number of concurrent tasks a task runner can execute at a time.
<code>N8N_RUNNERS_GRANT_TOKEN</code>	String	Random string	Token the runner uses to authenticate with the task broker.

<code>N8N_RUNNERS_ID</code>	String	Random string	ID the runner identifies itself with to the task broker.
<code>N8N_RUNNERS_AUTO_SHUTDOWN_TIMEOUT</code>	Number	15	The number of seconds to wait before shutting down an idle runner.
<code>N8N_RUNNERS_TASK_BROKER_URI</code>	String	<code>http://127.0.0.1:5679</code>	The URI of the task broker server (n8n instance).
<code>N8N_RUNNERS_LAUNCHER_HEALTH_CHECK_PORT</code>	Number	5680	Port for the launcher's health check server.
<code>N8N_RUNNERS_MAX_PAYLOAD</code>	Number	1 073 741 824	Maximum payload size in bytes for communication between a task broker and a task runner.
<code>N8N_RUNNERS_MAX_CONCURRENCY</code>	Number	5	The number of concurrent tasks a task runner can execute at a time.
<code>NODE_FUNCTION_ALLOW_BUILTIN</code>	String	-	Permit users to import specific built-in modules in the Code node.
<code>NODE_FUNCTION_ALLOW_EXTERNAL</code>	String	-	Permit users to import specific external modules (from <code>n8n/node_modules</code>) in the Code node. n8n disables importing modules by...
<code>N8N_RUNNERS_ALLOW_PROTOTYPE_MUTATION</code>	Boolean	false	Whether to allow prototype mutation for external libraries.
<code>GENERIC_TIMEZONE</code>	*	America/New_York	The same default timezone as configured for the n8n instance.
<code>NODE_OPTIONS</code>	String	-	Options for Node.js.
<code>N8N_RUNNERS_MAX_OLD_SPACE_SIZE</code>	String		The <code>--max-old-space-size</code> option to use for a task runner (in MB).
<code>N8N_RUNNERS_STDLIB_ALLOW</code>	String	-	Python standard library modules that you can use in the Code node, including their submodules.
<code>N8N_RUNNERS_EXTERNAL_ALLOW</code>	String	-	Third-party Python modules that are allowed to be used in the Code node, including their submodules.
<code>N8N_RUNNERS_ALLOW_TRANSITIVE_IMPORTS</code>	Boolean	false	Whether imports performed by an allowlisted package's own code skip the import allowlist (both <code>N8N_RUNNERS_EXTERNAL_ALLOW</code> ...
<code>N8N_RUNNERS_BUILTINS_DENY</code>	String	<code>eval,exec,compile,open,i...</code>	Python built-ins that you can't use in the Code node.
<code>N8N_BLOCK_RUNNER_ENV_ACCESS</code>	Boolean	true	Whether to block access to the runner's environment from within Python code tasks.

16.15 User Management and 2FA

VARIABLE	TYPE	DEFAULT	MEANING
<code>N8N_EMAIL_MODE</code>	String	<code>smtp</code>	Enable emails.
<code>N8N_SMTP_HOST</code>	String	-	<i>your_SMTP_server_name.</i>
<code>N8N_SMTP_PORT</code>	Number	-	<i>your_SMTP_server_port.</i>
<code>N8N_SMTP_USER</code>	String	-	<i>your_SMTP_username.</i>
<code>N8N_SMTP_PASS</code>	String	-	<i>your_SMTP_password.</i>
<code>N8N_SMTP_OAUTH_SERVICE_CLIENT</code>	String	-	If using 2LO with a service account this is your client ID.
<code>N8N_SMTP_OAUTH_PRIVATE_KEY</code>	String	-	If using 2LO with a service account this is your private key.
<code>N8N_SMTP_SENDER</code>	String	-	Sender email address.
<code>N8N_SMTP_SSL</code>	Boolean	<code>true</code>	Whether to use SSL for SMTP (true) or not (false).
<code>N8N_SMTP_STARTTLS</code>	Boolean	<code>true</code>	Whether to use STARTTLS for SMTP (true) or not (false).
<code>N8N_UM_EMAIL_TEMPLATES_INVITE</code>	String	-	Full path to your HTML email template.
<code>N8N_UM_EMAIL_TEMPLATES_PWRESET</code>	String	-	Full path to your HTML email template.
<code>N8N_UM_EMAIL_TEMPLATES_WORKFLOW_SHARED</code>	String	-	Overrides the default HTML template for notifying users that a workflow was shared.
<code>N8N_UM_EMAIL_TEMPLATES_CREDENTIALS_SHARED</code>	String	-	Overrides the default HTML template for notifying users that a credential was shared.
<code>N8N_UM_EMAIL_TEMPLATES_PROJECT_SHARED</code>	String	-	Overrides the default HTML template for notifying users that a project was shared.
<code>N8N_USER_MANAGEMENT_JWT_SECRET</code>	String	-	Set a specific JWT secret.
<code>N8N_USER_MANAGEMENT_JWT_DURATION_HOURS</code>	Number	168	Set an expiration date for the JWTs in hours.
<code>N8N_USER_MANAGEMENT_JWT_REFRESH_TIMEOUT_HOURS</code>	Number	0	How many hours before the JWT expires to automatically refresh it. 0 means 25% of <code>N8N_USER_MANAGEMENT_JWT_DURATION_HOURS</code> . -1...
<code>N8N_MFA_ENABLED</code>	Boolean	<code>true</code>	Whether to enable two-factor authentication (true) or disable (false). n8n ignores this if existing users have 2FA enabled.
<code>N8N_INVITE_LINKS_EMAIL_ONLY</code>	Boolean	<code>false</code>	When set to true, n8n will only deliver invite links via email and will not expose them through the API. This option enhances...

Built-In Node Directory

Every core node, trigger node, and AI root node documented in the official n8n docs, A–Z. Generated from the n8n documentation source (github.com/n8n-io/n8n-docs) in October 2026. n8n also has hundreds of **app nodes** (Google Sheets, Slack, Notion...) for actions inside specific services. Search the nodes panel for those.

IN THIS CHAPTER

- Core nodes: the building blocks
- Trigger nodes: every documented way a workflow can start from an outside app
- AI root nodes: agents, chains, and vector stores

TIP

In the editor, press **Tab** or click + and type part of a node name. The names below are the ones you'll see in that panel.

NODE	KIND	WHAT IT DOES
Activation Trigger	Core	The Activation Trigger node gets triggered when an event gets fired by n8n or a workflow.
ActiveCampaign Trigger	Trigger	N8n provides an app node for ActiveCampaign.
Acuity Scheduling Trigger	Trigger	Starts a workflow when an event happens in Acuity Scheduling.
Affinity Trigger	Trigger	N8n provides an app node for Affinity.
Aggregate	Core	Take separate items, or portions of them, and group them together into individual items.
AI Agent	AI (cluster root)	The AI Agent node lets you build an AI agent in n8n.
AI Transform	Core	Generate code snippets based on your prompt.
Airtable Trigger	Trigger	N8n provides an app node for Airtable.
AMQP Trigger	Trigger	Starts a workflow when an event happens in AMQP.
Asana Trigger	Trigger	N8n provides an app node for Asana.

Autopilot Trigger	Trigger	Starts a workflow when an event happens in Autopilot.
AWS SNS Trigger	Trigger	Starts a workflow when a message is published to an AWS SNS topic.
Azure AI Search Vector Store	AI (cluster root)	Azure AI Search (formerly Azure Cognitive Search) is a cloud search service with vector search capabilities for RAG and semantic search applications.
Basic LLM Chain	AI (cluster root)	Set the prompt that the model will use along with setting an optional parser for the response.
Bitbucket Trigger	Trigger	Starts a workflow when an event happens in Bitbucket.
Box Trigger	Trigger	To get your Target ID in Box:.
Brevo Trigger	Trigger	N8n provides an app node for Brevo.
Cal Trigger	Trigger	Starts a workflow when an event happens in Cal.
Calendly Trigger	Trigger	Here are some common errors and issues with the Calendly Trigger node and steps to resolve or troubleshoot them.
Chargebee Trigger	Trigger	To add a Webhook URL in Chargebee:.
Chat	Core	Send messages into the chat and optionally wait for responses from users.
Chat Trigger	Core	Use the Chat Trigger node when building AI workflows for chatbots and other chat interfaces.
Chroma Vector Store	AI (cluster root)	Interact with your Chroma database as vector store.
ClickUp Trigger	Trigger	N8n provides an app node for ClickUp.
Clockify Trigger	Trigger	This node uses the workflow timezone setting to specify the range of time entries starting time.
Code	Core	The Code node lets you run your own JavaScript or Python inside a workflow, so you can transform data or add logic the built-in nodes don't cover.
Compare Datasets	Core	The Compare Datasets node helps you compare data from two input streams.
Compression	Core	Compress and decompress files.
Convert to File	Core	Take input data and output it as a file.
ConvertKit Trigger	Trigger	N8n provides an app node for ConvertKit.
Copper Trigger	Trigger	N8n provides an app node for Copper.
Crypto	Core	Perform cryptographic operations in workflows.
Customer.io Trigger	Trigger	N8n provides an app node for Customer.io.
Data table	Core	Create and manage internal data tables.
Databricks Trigger	Trigger	Respond to events in Databricks and integrate Databricks with other applications.
Date & Time	Core	The Date & Time node manipulates date and time data and convert it to different formats.
Debug Helper	Core	Trigger different error types or generate random datasets to help test n8n workflows.
Edit Fields (Set)	Core	Set workflow data.
Edit Image	Core	Manipulate and edit images.
Email Trigger (IMAP)	Core	Receive emails using an IMAP email server.
Emelia Trigger	Trigger	Starts a workflow when an event happens in Emelia.
Error Trigger	Core	You can use the Error Trigger node to create error workflows.

Evaluation	Core	The Evaluation node performs various operations related to evaluations to validate your AI workflow reliability.
Evaluation Trigger	Core	Use the Evaluation Trigger node when setting up evaluations to validate your AI workflow reliability.
Eventbrite Trigger	Trigger	Starts a workflow when an event happens in Eventbrite.
Execute Command	Core	The Execute Command node runs shell commands on the host machine that runs n8n.
Execute Sub-workflow	Core	Run a different workflow on the host machine that runs n8n.
Execute Sub-workflow Trigger	Core	Use this node to start a workflow in response to another workflow.
Execution Data	Core	Use this node to save metadata for workflow executions.
Extract From File	Core	A common pattern in n8n workflows is to receive a file, either from an [HTTP Request node][] (for files you are fetching from a website), a...
Facebook Lead Ads Trigger	Trigger	Respond to events in Facebook Lead Ads and integrate Facebook Lead Ads with other applications. n8n has built-in support for responding to new leads.
Facebook Trigger	Trigger	Trigger a workflow when events occur in Facebook.
Figma Trigger (Beta)	Trigger	Starts a workflow when an event happens in Figma.
Filter	Core	Filter items based on a condition.
Flow Trigger	Trigger	Flow is modern task and project management software for teams.
Form.io Trigger	Trigger	Starts a workflow when an event happens in Form.io.
Formstack Trigger	Trigger	Starts a workflow when an event happens in Formstack.
FTP	Core	The FTP node is useful to access and upload files to an FTP or SFTP server.
GetResponse Trigger	Trigger	Starts a workflow when an event happens in GetResponse.
Git	Core	Configure this operation with these parameters:.
GitHub Trigger	Trigger	N8n provides an app node for GitHub.
GitLab Trigger	Trigger	If your repository is located within nested GitLab groups or subgroups (for example, <code>https://gitlab.com/org/subgroup1/subgroup2/my-repo</code>),...
Gmail Trigger	Trigger	Configure the node with these parameters:.
Google Business Profile Trigger	Trigger	Respond to events in Google Business Profile and integrate Google Business Profile with other applications. n8n has built-in support for...
Google Calendar Trigger	Trigger	N8n provides an app node for Google Calendar.
Google Drive Trigger	Trigger	For common questions or issues and suggested solutions, refer to Common issues.
Google Sheets Trigger	Trigger	N8n provides an app node for Google Sheets.
GraphQL	Core	Select the type of authentication to use. n8n recommends using Predefined Credential Type when it's available for the service you want to...
Guardrails	Core	Enforce safety, security, and content policies on text.
Gumroad Trigger	Trigger	Starts a workflow when an event happens in Gumroad.
Help Scout Trigger	Trigger	Starts a workflow when an event happens in Help Scout.

HTML	Core	The HTML node provides operations to help you work with HTML in n8n.
HTTP Request	Core	The HTTP Request node is one of the most versatile nodes in n8n.
HubSpot Trigger	Trigger	N8n provides an app node for HubSpot.
If	Core	Split a workflow conditionally based on comparison operations.
Information Extractor	AI (cluster root)	Extract structured information from incoming data.
Invoice Ninja Trigger	Trigger	Starts a workflow when an event happens in Invoice Ninja.
Jira Trigger	Trigger	Starts a workflow when an event happens in Jira.
Jotform Trigger	Trigger	Starts a workflow when an event happens in Jotform.
JWT	Core	Work with JSON web tokens in your n8n workflows.
Kafka Trigger	Trigger	Version 2 runs on a new, actively maintained Kafka engine.
Keap Trigger	Trigger	Starts a workflow when an event happens in Keap.
KoboToolbox Trigger	Trigger	This node starts a workflow upon new submissions of a specified form.
LangChain Code	AI (cluster root)	Import LangChain.
LDAP	Core	This node allows you to interact with your LDAP servers to create, find, and update objects.
Lemlist Trigger	Trigger	Starts a workflow when an event happens in Lemlist.
Limit	Core	Remove items beyond a defined maximum number.
Linear Trigger	Trigger	Starts a workflow when an event happens in Linear.
Local File Trigger	Core	The Local File Trigger node starts a workflow when it detects changes on the file system.
LoneScale Trigger	Trigger	Respond to workflow events in LoneScale and integrate LoneScale with other applications.
Loop Over Items	Core	The Loop Over Items node helps you loop through data when needed.
Mailchimp Trigger	Trigger	Starts a workflow when an event happens in Mailchimp.
MailerLite Trigger	Trigger	Starts a workflow when an event happens in MailerLite.
Mailjet Trigger	Trigger	Starts a workflow when an event happens in Mailjet.
Manual Trigger	Core	Use this node if you want to start a workflow by selecting Execute Workflow and don't want any option for the workflow to run automatically.
Markdown	Core	The Markdown node converts between Markdown and HTML formats.
Mautic Trigger	Trigger	N8n provides an app node for Mautic.
MCP Server Trigger	Core	Allow n8n to act as a Model Context Protocol (MCP) server, making n8n tools and workflows available to MCP clients.
Merge	Core	Combine data from multiple streams, once data of all streams is available.
Message an Agent	Core	Send a message to an agent from a workflow, then use the agent's reply in the nodes that follow.
Microsoft Agent 365 Trigger	AI (cluster root)	Receive messages from Microsoft Agent 365 and respond with AI-powered agent capabilities.

Microsoft OneDrive Trigger	Trigger	Respond to events in Microsoft OneDrive and integrate Microsoft OneDrive with other applications. n8n has built-in support for file and folder...
Microsoft Outlook Trigger	Trigger	Respond to events in Microsoft Outlook and integrate Microsoft Outlook with other applications.
Microsoft SharePoint Trigger	Trigger	Start a workflow when a file or a list item changes in Microsoft SharePoint.
Microsoft Teams Trigger	Trigger	Respond to events in Microsoft Teams and integrate Microsoft Teams with other applications.
Milvus Vector Store	AI (cluster root)	Interact with your Milvus database as vector store.
MongoDB Atlas Vector Store	AI (cluster root)	MongoDB Atlas Vector Search is a feature of MongoDB Atlas that enables users to store and query vector embeddings.
MQTT Trigger	Trigger	Starts a workflow when an event happens in MQTT.
n8n	Core	A node to integrate with n8n itself.
n8n Form	Core	Create user-facing forms with multiple steps.
n8n Form Trigger	Core	Use the n8n Form trigger to start a workflow when a user submits a form, taking the input data from the form.
n8n Trigger	Core	The n8n Trigger node triggers when the workflow containing this node updates or gets published, or when the n8n instance starts or restarts.
Netlify Trigger	Trigger	N8n provides an app node for Netlify.
No Operation, do nothing	Core	Use the No Operation, do nothing node when you don't want to perform any operations.
Notion Trigger	Trigger	N8n provides an app node for Notion.
Onfleet Trigger	Trigger	Trigger a workflow on:.
Oracle Database Vector Store	AI (cluster root)	Interact with Oracle Database as a vector store.
PayPal Trigger	Trigger	Starts a workflow when an event happens in PayPal.
PGVector Vector Store	AI (cluster root)	PGVector is an extension of Postgresql.
Pinecone Vector Store	AI (cluster root)	Interact with your Pinecone database as vector store.
Pipedrive Trigger	Trigger	Starts a workflow when an event happens in Pipedrive.
Postgres Trigger	Trigger	Respond to events in Postgres and integrate Postgres with other applications. n8n has built-in support responding to insert, update, and delete...
Postmark Trigger	Trigger	Starts a workflow when an event happens in Postmark.
Pushcut Trigger	Trigger	Follow these steps to configure your Pushcut Trigger node with your Pushcut app.
Qdrant Vector Store	AI (cluster root)	Interact with your Qdrant collection as a vector store.
Question and Answer Chain	AI (cluster root)	Use a vector store as a retriever.
RabbitMQ Trigger	Trigger	Starts a workflow when a message arrives in a RabbitMQ queue.

Read/Write Files from Disk	Core	Read and write files from/to the machine where n8n is running.
Redis Trigger	Trigger	Subscribe to a Redis channel.
Redis Vector Store	AI (cluster root)	Interact with your Redis database as a vector store.
Remove Duplicates	Core	Identify and delete items that are:.
Rename Keys	Core	Rename the keys of a key-value pair in n8n.
Respond to Webhook	Core	Control the response to incoming webhooks.
RSS Feed Trigger	Core	The RSS Feed Trigger node allows you to start an n8n workflow when a new RSS feed item has been published.
RSS Read	Core	Read data from RSS feeds published on the internet.
Salesforce Trigger	Trigger	Respond to events in Salesforce and integrate Salesforce with other applications.
Schedule Trigger	Core	Run workflows at fixed intervals and times.
SeaTable Trigger	Trigger	Starts a workflow when an event happens in SeaTable.
Send Email	Core	The Send Email node sends emails using an SMTP email server.
Sentiment Analysis	AI (cluster root)	Analyze the sentiment of incoming text data.
Shopify Trigger	Trigger	Starts a workflow when an event happens in Shopify.
Simple Vector Store	AI (cluster root)	Store and retrieve embeddings in n8n's in-app memory.
Slack Trigger	Trigger	Respond to events in Slack and integrate Slack with other applications.
Sort	Core	Organize lists of items in a desired ordering, or generate a random selection.
Split Out	Core	Separate a single data item containing a list into multiple items.
SSE Trigger	Core	Server-Sent Events (SSE) is a server push technology enabling a client to receive automatic updates from a server using HTTP connection.
SSH	Core	The SSH node is useful for executing commands using the Secure Shell Protocol.
Stop And Error	Core	Display custom error messages, cause executions to fail under certain conditions, and send custom error information to error workflows.
Strava Trigger	Trigger	Starts a workflow when an event happens in Strava.
Stripe Trigger	Trigger	The Stripe Trigger node can verify that incoming webhook requests genuinely come from Stripe.
Summarization Chain	AI (cluster root)	Summarize multiple documents.
Summarize	Core	Aggregate items together, in a manner similar to Excel pivot tables.
Supabase Vector Store	AI (cluster root)	Use the Supabase Vector Store to interact with your Supabase database as vector store.
SurveyMonkey Trigger	Trigger	Starts a workflow when an event happens in SurveyMonkey.
Switch	Core	Route a workflow conditionally based on comparison operations.
Taiga Trigger	Trigger	Starts a workflow when an event happens in Taiga.

Telegram Trigger	Trigger	Some events may require additional permissions , see Telegram's API documentation for more information.
Text Classifier	AI (cluster root)	Classify (categorize) incoming data.
TheHive 5 Trigger	Trigger	Respond to events in TheHive and integrate TheHive with other applications.
TheHive Trigger	Trigger	N8n provides an app node for TheHive.
Toggl Trigger	Trigger	Starts a workflow when an event happens in Toggl.
TOTP	Core	The TOTP node provides a way to generate a TOTP (time-based one-time password).
Track Time Saved	Core	Record how many minutes a workflow saves, based on the path an execution actually takes. n8n adds up every Track Time Saved node that runs during...
Trello Trigger	Trigger	The model ID is the ID of any model in Trello.
Twilio Trigger	Trigger	Respond to events in Twilio and integrate Twilio with other applications.
Typeform Trigger	Trigger	Starts a workflow when an event happens in Typeform.
Venafi TLS Protect Cloud Trigger	Trigger	Start a workflow in n8n in response to events in the cloud-based Venafi TLS Protect service.
Wait	Core	Use the Wait node pause your workflow's execution.
Weaviate Vector Store	AI (cluster root)	Interact with your Weaviate collection as a vector store.
Webex by Cisco Trigger	Trigger	Starts a workflow when an event happens in Webex by Cisco.
Webflow Trigger	Trigger	Starts a workflow when an event happens in Webflow.
Webhook	Core	Create webhooks, which can receive data from apps and services when an event occurs.
WhatsApp Trigger	Trigger	Respond to events in WhatsApp and integrate WhatsApp with other applications.
Wise Trigger	Trigger	Starts a workflow when an event happens in Wise.
WooCommerce Trigger	Trigger	Starts a workflow when an event happens in WooCommerce.
Workable Trigger	Trigger	Respond to events in the Workable recruiting platform and integrate Workable with other applications.
Workflow Trigger	Core	The Workflow Trigger node gets triggered when a workflow is updated or activated.
Wufoo Trigger	Trigger	Starts a workflow when an event happens in Wufoo.
XML	Core	Convert data from and to XML.
Zendesk Trigger	Trigger	Starts a workflow when an event happens in Zendesk.
Zep Vector Store	AI (cluster root)	Use the Zep Vector Store to interact with Zep vector databases.

Quick Reference

Pattern: Trigger → get data → reshape → decide → act

Start: n8n Cloud trial · or `curl -fsSL https://get.n8n.io | sh` · or Docker `n8nio/n8n` → <http://localhost:5678>

Editor: `Tab` add node · **Execute step** test one node · **Execute workflow** test all · **Publish** (`Shift` + `P`) go live

Expressions: `{{ $json.field }}` · `{{ $('Node name').item.json.field }}` · `{{ $now.toFormat('yyyy-MM-dd') }}`

Webhooks: test URL while listening · production URL only after publishing

Reliability: Executions tab · Retry On Fail · error workflow with Error Trigger · pin data while building

AI: AI Agent = chat model + memory + tools + system message · MCP Client Tool to use servers · MCP Server Trigger to become one

Glossary

AI Agent node	A workflow node that uses an AI model to decide which attached tools to call to complete a request.
API	A way for programs to talk to each other; n8n's HTTP Request node can call almost any API.
Credential	Saved sign-in details, an OAuth connection or an API key, that let nodes use an app.
Data table	A simple table stored inside n8n for keeping data between workflow runs.
Execution	One run of a workflow, logged with all its data.
Expression	A formula inside <code>{{ }}</code> that inserts or calculates values from earlier nodes.
Gateway credits	Prepaid n8n Cloud credits for running supported AI models without your own provider account.
Item	A single piece of data, such as one row or one email, flowing through a workflow.
JSON	The structured text format n8n uses to represent data.
MCP	Model Context Protocol, the standard for connecting AI apps to tools; see the <i>MCP</i> manual.
Node	One building block in a workflow: a trigger, an app action, or a logic step.
OAuth	A secure “sign in with...” method for connecting apps without sharing your password.
Pin data	Freeze a node's output so you can keep building without re-running it.
Publish	Make the current version of a workflow live so its triggers run automatically.
Self-hosting	Running n8n on your own computer or server instead of n8n Cloud.

Sub-workflow	A workflow called by another workflow, used to reuse common steps.
Trigger	The node that starts a workflow, such as a schedule, webhook, form, or app event.
Webhook	A unique web address that other apps send data to, which starts a workflow.
Workflow	A set of connected nodes that automates a task.
