n8n Cheat Sheet Dashboard

Docker Self-Hosting

Use **Docker** to install, update, and run n8n quickly. Adjust volumes, environment variables, and ports as needed.

Clone the Starter

```
git clone https://github.com/n8n-io/self-hosted-n8n-starter.git
cd self-hosted-n8n-starter
```

Pull the Latest Version

```
sudo docker pull n8nio/n8n:latest
```

Stop Existing Container

```
sudo docker stop n8n
```

Remove Container

```
sudo docker rm n8n
```

Example Docker Run

```
docker run -it --rm \
--name n8n \
-p 5678:5678 \
-v ~/.n8n:/home/node/.n8n \
-e WEBHOOK_URL="https://n8n.example.com/" \
-e N8N_SMTP_HOST="mail.server.com" \
-e N8N_SMTP_PORT="587" \
-e N8N_SMTP_USER="YOUR_EMAIL" \
-e N8N_SMTP_PASS="YOUR_EMAIL_PW" \
-e N8N_SMTP_SENDER="n8n" \
-e N8N_SMTP_SSL=false \
-e GENERIC_TIMEZONE="Asia/Seoul" \
-e N8N_ENCRYPTION_KEY="RANDOM_STRING" \
-e N8N_BASIC_AUTH_ACTIVE=true \
-e N8N_BASIC_AUTH_USER="n8nAdmin" \
-e N8N_BASIC_AUTH_PASSWORD="n8nPassword" \
-e EXECUTIONS_TIMEOUT=3600 \
-e EXECUTIONS_TIMEOUT_MAX=7200 \
n8nio/n8n:latest
```

n8n Hints & Tips

Keyboard Shortcuts

A selection of default shortcuts you can use while editing workflows in n8n:

Category	Detail	Shortcut / Gesture
Workflow Control	Create New Workflow	`Ctrl + Alt + N`
	Open Workflow	`Ctrl + O`
	Save Current Workflow	`Ctrl + S`
	Undo	`Ctrl + Z`
	Redo	`Ctrl + Shift + Z`
	Run Workflow	`Ctrl + Enter`
	Stop Workflow	`Alt + S`
Canvas Zoom	Zoom In	`+`
	Zoom Out	`-`
	Reset Zoom	`0`
	Fit Workflow to View	`1`
	Zoom In/Out	`Ctrl + Mouse Wheel`
Canvas Navigation	Move Node (View)	`Ctrl + Left Mouse + Drag`
	Move Node (View)	`Middle Mouse + Drag`
	Move Node (View)	`Space + Drag`
	Move Node (View)	`Two Fingers (Touch)`
Canvas Nodes	Select All Nodes	`Ctrl + A`
	Copy Node(s)	`Ctrl + C`
	Paste Node(s)	`Ctrl + V`
	Delete Node(s)	`Delete`

Category	Detail	Shortcut / Gesture
	Disable / Enable Node	`Ctrl + K` / `Ctrl + E`

Getting Started with Triggers

Trigger Node (Start)

A Trigger node is the *entry point* to a workflow, defining **when** and **how** a workflow should start.

- **Manual:** For quick tests or debugging
- **Schedule:** Time-based triggers (e.g. every hour)
- **Webhook:** External events (HTTP POST, etc.)

Common flow: *Trigger Node → transform/validate data → next Node(s) → final action*

Examples

Type	Description
Manual Trigger	Click "Execute Workflow" in n8n UI to run workflow immediately
Schedule Trigger	Define CRON expression or time intervals for auto runs
Webhook Trigger	Waits for an HTTP request, runs with the request's data

</> Expressions

Expressions let you dynamically set or transform values within n8n. Some sample expressions:

Type	Expression	Description
Basic	`{{ $json.field }}`	Access `field` from the incoming JSON data
Specific Node	`{{ $node["NodeName"].json["value"] }}`	Access data from a different node's output

Type	Expression	Description
Ternary	`{{ $json["count"] > 10 ? "High" : "Low" }}`	Conditional logic in-line
Date	`{{ new Date().toISOString() }}`	Generate an ISO timestamp
Built-ins	`$json`, `$binary`, `$itemIndex`, etc.	Special references in n8n (environment, node meta, etc.)

Built-in Nodes

Common nodes that ship with n8n:

Category	Node Name	Description
Trigger Node	Manual, Schedule, Webhook...	Start or schedule workflow execution
Core Node	Function	Run custom JS code to manipulate data
Core Node	HTTP Request	Make REST or API calls
Data Transform	Edit Fields	Add/modify fields in incoming data
Data Transform	Filter	Remove items that don’t meet certain conditions
Data Transform	Merge	Combine data from two different sources
Data Transform	Split In Batches	Chunk incoming items into smaller batches
File Operations	Read Binary File	Load file data as binary
File Operations	Write Binary File	Write binary data to disk
Utility	Stop And Error	Intentionally stop the workflow or throw an error

AI Agent

The **AI Agent** node (or custom node) integrates Large Language Models or other AI services into your workflow. Possible uses:

- LLM prompt or chat interactions
- Context memory stored in external DB or memory node
- Search results or knowledge base used as context for the AI

Flow example: *Webhook → AI Agent → Memory Node → Output*

OpenAI, Cohere, or custom endpoints are feasible.

HTTP Request (API)

HTTP Request Node: Used to call external REST APIs. You can also import a *cURL* command to auto-populate the fields.

Sample cURL

```
curl -X POST https://api.example.com/data \ -H "Content-Type: application/json" \ -d '{"name": "Alice", "email": "alice@example.com"}'
```

After pasting the cURL, click *"Parse to HTTP Request"* in n8n to automatically fill in method, URL, headers, and body.

Nodes Common Settings

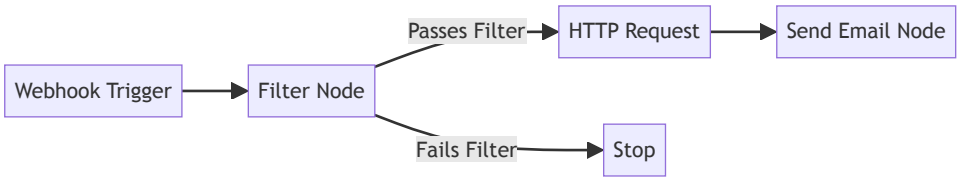
Common settings and error handling options in many nodes:

Setting	Description
Always Output Data	Outputs an empty item if no data is found. Helpful, but can break loops or filters.
Execute Once	Node only processes the *first* incoming item, ignoring others.
Retry on Fail	Node automatically retries on error (API timeouts, etc.).
Error Handling: Stop Workflow	Entire workflow stops immediately if this node errors.

Setting	Description
Error Handling: Continue	Ignores errors and continues the workflow with the last valid data.
Error Handling: Pass Error Data	Sends error details on a separate error output, letting the workflow proceed.
Node Notes	Adds a note or comment visible in the workflow for documentation.

Workflow Diagram Sample (Mermaid)

This is a sample **Mermaid** diagram showing a simple n8n workflow flowchart.



Authentication & Credentials

n8n has a built-in credentials manager that lets you securely store and re-use your authentication details. Supported credential types include:

- API Key (e.g. Bearer token, Query param)
- OAuth2, including refresh tokens
- Basic Auth / Username & Password

Once configured, you can select these credentials in your nodes. Example:

Setting up OAuth2:
1. Go to Credentials > Create New > OAuth2
2. Provide Client ID, Secret, Auth URLs
3. Grant n8n access (manual or PKCE flow)

For more details, see n8n Docs: Credentials.

n8n uses a secure credentials management system to handle authentication for hundreds of external services. This allows you to connect your workflows to various platforms without exposing sensitive information directly in your workflow logic. Credentials are encrypted and stored securely within your n8n instance.

Here's a list of some popular service connections you can manage credentials for:

Icon	Title	Description
	Trello	Manage your projects and collaborate with your team.
	Raindrop	Save and organize your bookmarks and online content.
	Microsoft (e.g., Outlook, OneDrive)	Access various Microsoft services for email, file storage, and more.
	Amazon (e.g., S3, SNS, SES)	Integrate with Amazon Web Services for storage, messaging, and email.
	Dropbox	Access and manage your files stored in Dropbox.
	Google (e.g., Gmail, Sheets, Calendar)	Connect to Google services for email, spreadsheets, and calendar events.
	Slack	Send and receive messages and interact with your Slack workspace.
	Twitter/X	Post tweets, retrieve data, and interact with the Twitter/X API.
	GitHub	Automate tasks related to your GitHub repositories and workflows.
	Email (SMTP/IMAP)	Send and receive emails using standard email protocols.
	Databases (e.g., PostgreSQL, MySQL)	Connect to and interact with various database systems.
	Airtable	Work with data stored in Airtable bases.
	OneDrive	Access and manage files on Microsoft OneDrive.

Icon	Title	Description
	Box	Manage and access content stored in Box.
	Shopify	Automate tasks related to your Shopify store.
stripe	Stripe	Manage payments and customer data through Stripe.
	Discord	Interact with Discord servers and channels.
	LinkedIn	Engage with the LinkedIn platform for professional networking.
	Jira	Manage projects and track issues with Jira.
	Salesforce	Automate workflows involving your Salesforce CRM data.

Community Nodes

The n8n community is a vibrant ecosystem that significantly extends the platform's capabilities through custom-built nodes.

Explore Hundreds of Community Nodes

Discover a vast collection of community-developed nodes on npm. These nodes provide integrations with a wide array of services and tools, enabling you to connect to platforms not natively supported by n8n. Browse Community Nodes on npm

Extend n8n's Functionality

Community nodes allow you to seamlessly incorporate specialized functionalities into your n8n workflows. From interacting with niche APIs to performing unique data transformations, these nodes empower you to automate complex tasks with ease.

Build Your Own Custom Nodes

If you have a specific integration in mind that isn't available, you can leverage n8n's powerful SDK to build your own custom nodes. The

n8n documentation offers comprehensive guides to assist developers in creating and sharing their contributions with the community.

The active community ensures a constantly growing library of nodes, making n8n an incredibly adaptable and versatile automation platform.

⚠ Advanced Error Handling

For complex workflows, n8n provides robust debugging features:

- **Error Trigger Node**: Automatically catches unhandled errors across workflows
- **Execution Logs**: Track each run (success or error), see full input/output data
- **Pinned Data**: Temporarily store data at a node for repeated local testing
- **Retry Logic**: Built into some nodes or can be orchestrated with loops

Example debugging flow might send error details to Slack or email you logs. Combine this with the "Stop on Error" or "Continue on Error" settings, depending on your needs.

Sub-Workflows & Reusability

The **Execute Workflow** node can call another workflow (a "sub-workflow") for reusability and modular design.

- Extract common tasks (e.g. CSV to JSON conversion) into a dedicated workflow
- Reference that "utility" workflow from many other main workflows
- Pass data in/out seamlessly between calling workflow and sub-workflow

Tip: Keep your sub-workflows small and dedicated to one function, so they're easy to maintain.

Security & Hardening

When self-hosting, ensure your instance is secure:

- **HTTPS**: Use reverse proxies (e.g. Nginx, Caddy) or direct SSL for all traffic
- **Basic Auth**: Enable `N8N_BASIC_AUTH_ACTIVE=true` plus username/password variables
- **Encryption Key**: Always set `N8N_ENCRYPTION_KEY` for your environment
- Restrict IP ranges or firewall inbound traffic for Webhook endpoints
- Avoid exposing your n8n container directly to the internet without a proxy

Community & Resources

The n8n community is active and growing. Check out:

- **Official Docs**: docs.n8n.io
- **Community Forum**: community.n8n.io
- **Discord Chat**: discord.gg/n8n
- **GitHub** (Issues, PRs, source code): n8n-io/n8n

Community contributed nodes and examples can often be found on GitHub or in the forum. Keep an eye on the n8n blog for release notes and new features.

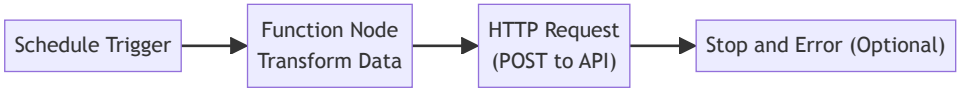
Core Nodes Workflows

Here are three example workflows using key n8n core nodes. These diagrams are generated via **Mermaid.js**. Make sure to *avoid HTML tags* in your labels to prevent syntax errors.

---

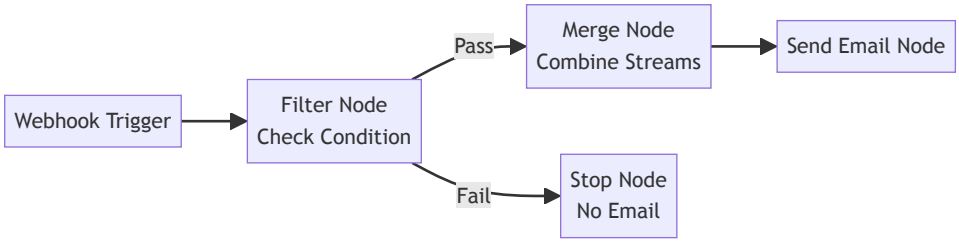
1) Schedule → Function → HTTP Request

Runs on a time-based trigger, modifies data in a Function node, and sends it via HTTP Request.



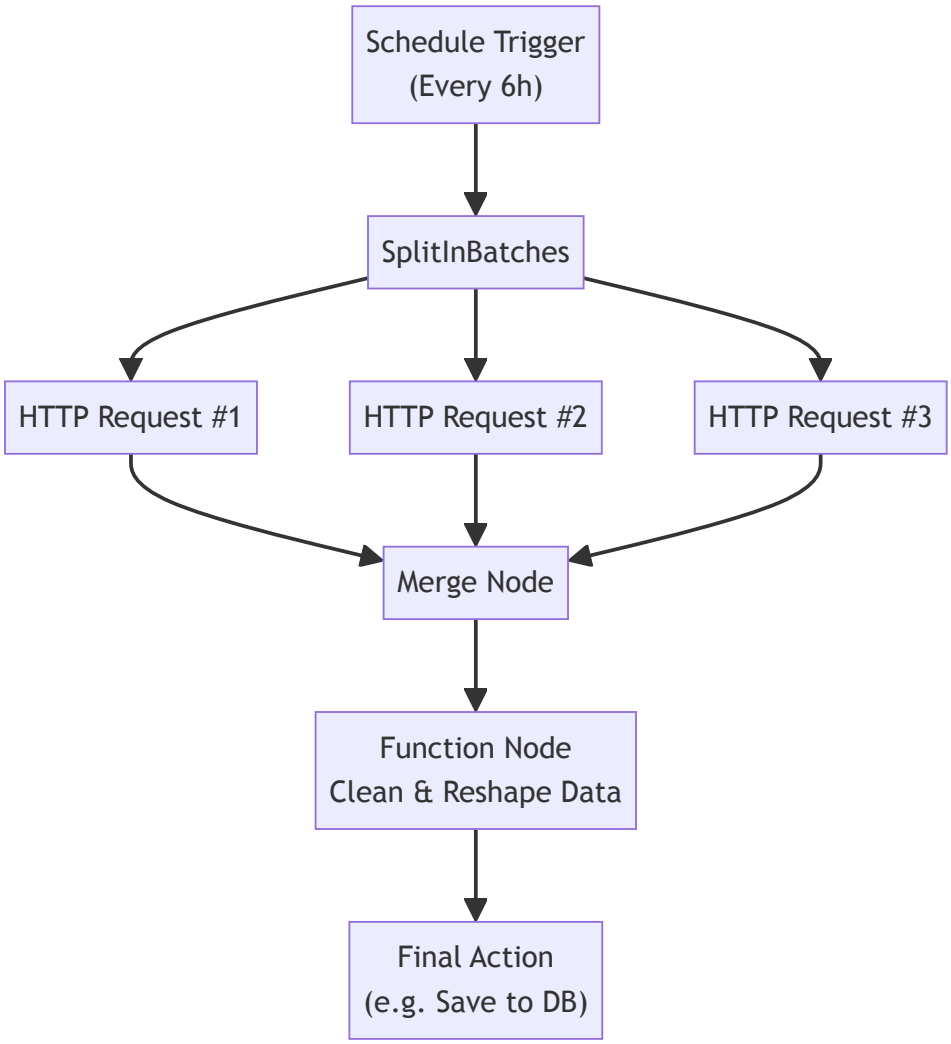
2) Webhook → Filter → Merge → Send Email

A webhook receives incoming data, then a Filter node checks conditions. If it passes, data merges and eventually is sent via Email.



3) Multiple Branches & Merge

Splits data into batches, processes each with different HTTP calls, merges them, and applies a Function node for final transformation.



n8n Hints & Tips

Below are some useful hints and tips to help you master n8n:

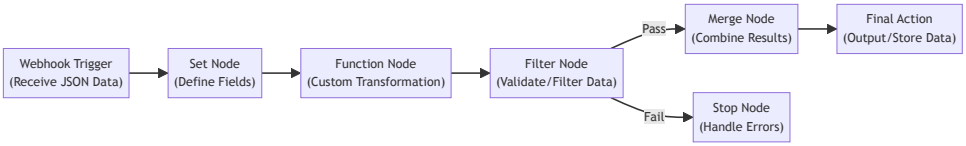
- **Use Expression Mode**
  - Activate by clicking the "=" icon in node parameters.
  - Leverage built-in variables like `$json`, `$node`, `$env`, and others.
- **Modular Workflows**
  - Break complex workflows into reusable sub-workflows using the `Execute Workflow` node.
  - This makes maintenance and scaling easier.
- **Debugging Techniques**
  - Review the execution logs to identify and correct errors.
  - Pin node data to capture transient issues for troubleshooting.
  - Enable detailed logging by configuring your environment variables.
- **Credential Management**
  - Store API keys, OAuth2 tokens, and other secrets in the n8n credentials manager.
  - Use environment variables to secure sensitive information.
- **Performance Optimization**
  - Use the `SplitInBatches` node for processing large data sets.
  - Implement error handling and retries to prevent workflow disruptions.
- **Community Resources**
  - Join the n8n Community Forum and Discord for troubleshooting and best practices.
  - Check out the official n8n documentation for regular updates.

⇄ Data Transformation & Nodes

This section explains how data is transformed within an n8n workflow using core nodes. In many workflows, raw data is received from an external source, transformed, filtered, and merged before being delivered to the final destination.

- **Function Node:** Execute custom JavaScript code to manipulate data.
- **Set Node:** Define or modify data fields directly.
- **Filter Node:** Check conditions and only allow data that meets the criteria.
- **Merge Node:** Combine data from different sources or branches.

The following Mermaid diagram illustrates a workflow that demonstrates a data transformation process:



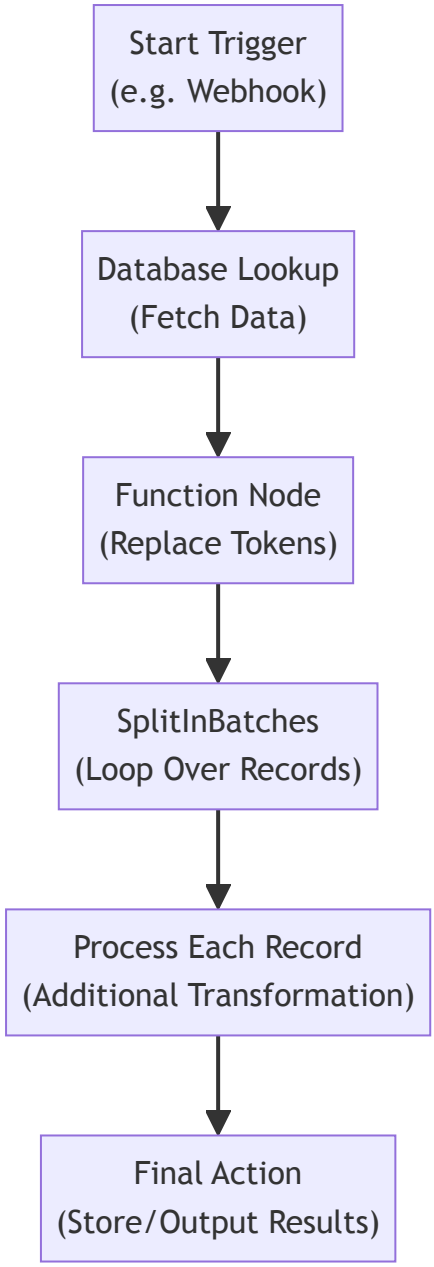
In this workflow:
A receives the data. **B** preps the data structure. **C** applies custom transformations; for example, converting date formats or computing new fields. **D** then filters the data based on your conditions. If the data passes, it is merged and sent to **F**; otherwise, the flow is stopped at **G**.

Database Lookup, Replace Function & Looping

This workflow demonstrates a common pattern in n8n where data is first retrieved from a database, then processed with a replace function, and finally looped for further processing.

- **Database Lookup:** Uses a database node (e.g. MySQL, PostgreSQL) to fetch related data.
- **Replace Function:** A Function node is used to search through the data and replace parts of it—for instance, replacing tokenized strings with real values.
- **Looping:** The SplitInBatches node is applied to process multiple records iteratively.

The following Mermaid diagram illustrates this workflow:



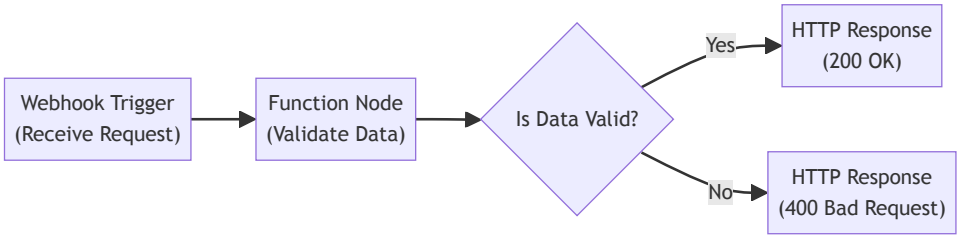
Workflow Explanation:
A initiates the workflow (e.g., via a webhook or schedule). Then, *B* performs a lookup in your database for related data. The retrieved data is passed to a *Function Node (C)* where a replace operation occurs (e.g., using regular expressions or simple string replacements). The result is split into individual records by the *SplitInBatches Node (D)*, allowing you to process each record in a loop. Finally, each record is further transformed or enriched at *E*, and the output is sent to the final destination at *F*.

Triggers & Webhook Responses

This section explains how n8n triggers work and how workflows respond to HTTP requests using webhooks. n8n supports several trigger types including manual, scheduled, and webhook triggers.

- **Webhook Trigger:** Initiates a workflow when an HTTP request is received. It can handle various HTTP methods such as GET, POST, etc.
- **Response Handling:** Once the workflow processes the request, you can send back a custom HTTP response (e.g., 200 OK, 400 Bad Request) with custom headers and a message.
- **Authentication & Security:** Webhook triggers can be secured using secret tokens, IP whitelisting, or Basic Auth to ensure that only authorized requests trigger the workflow.
- **Debugging Tip:** Use the Execution Logs and Pin Node Data to troubleshoot webhook responses and data processing.

The following Mermaid diagram illustrates a typical webhook workflow where a webhook triggers the workflow, a Function node validates the incoming payload, and an HTTP Response is returned based on the validation result.



In this example, when the *Webhook Trigger* receives data, the *Function Node* checks the payload. Depending on whether the data is valid or not, the workflow returns a *200 OK* or *400 Bad Request* response.

Complementary Software Stack

The following software components can help build a robust environment around n8n. They manage server processes, provide reverse proxy services, secure connections, and enhance development, multimedia, and security capabilities.

Icon	Software	Description
	Node.js	JavaScript runtime environment for running server-side code—essential for n8n and other backend processes.
	Python 3	A versatile programming language for scripting, data science, and automation,

Icon	Software	Description
		complementing n8n for custom scripts or AI/ML workflows.
	**Nginx**	High-performance web server and reverse proxy to efficiently manage HTTP traffic and provide load-balancing.
	**Caddy**	A modern web server with automatic HTTPS using Let's Encrypt, offering simpler configuration compared to traditional servers.
	**Certbot**	A tool for obtaining and renewing SSL/TLS certificates from Let's Encrypt, ensuring secure connections.
	**PM2**	Production process manager for Node.js applications, ensuring reliable service and automatic restarts on failure.
	**Docker**	Containerisation platform that simplifies deployment and scaling by packaging your services into portable containers.
	**OLLAMA**	An open-source local LLM runner enabling offline use of AI models, ideal for privacy-focused or air-gapped environments.
	**Git**	A distributed version control system to efficiently manage code changes and collaborate on workflow developments.
	**PostgreSQL**	A robust open-source relational database for managing persistent data, with alternatives like MySQL or MongoDB available.
	**Redis**	An in-memory data store used as a cache or message broker, enhancing performance with fast data retrieval.
	**FFmpeg**	A powerful multimedia framework to decode, encode, transcode, and stream

Icon	Software	Description
		audio/video files—useful for processing media in workflows.
	**ImageMagick**	An extensive image processing tool for creating, editing, and composing bitmap images, ideal for dynamic image transformations.
	**Vue.js**	A progressive JavaScript framework for building interactive user interfaces, perfect for developing custom dashboards or control panels.
	**n8n Community Nodes**	Custom nodes contributed by the community that extend n8n's native functionality, providing integrations and features not available out of the box.
	**Cockpit**	A web-based interface to manage GNU/Linux servers, monitor performance, and handle storage/networking tasks easily via a graphical dashboard.
	**Whazu Security**	A security monitoring solution designed to provide threat detection, integrity monitoring, and incident response, ensuring your infrastructure remains secure.

Performance & Specs Comparison

Use this guide to plan and gauge your deployment environments. The table below outlines recommended server specifications for different stages: Prototype, Test, and Production.

Environment	CPU	RAM	Storage	Additional Features
Prototype	1-2 Cores	2-4 GB	20-50 GB SSD	Basic logging, minimal load