

N8N Self-Hosted AI Starter Kit Installation and Usage Guide

N8N AI Self hosted kit gives you a low AI code development environment on your local machine and access to locally hosted AI models at no cost. It can be used as a perfect local development environment to build N8N workflows across different platforms. In our course we will make use of REST APIs, SQL APIs to connect to various platforms like Snowflake, Atlassian JIRA, Microsoft Outlook, Gmail etc. to build out data operations workflows.

Prerequisites:

- Docker
- Gitbash

Installation Steps

1. Clone the Repository:

Open your terminal and run the commands below to clone the GitHub repository into your local system:

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

2. Set Up Environment Variables:

- Copy the example environment file and edit secrets/passwords:
- `cp. env.example. env`
Edit .env to set your secrets and passwords but it might not be needed at this moment so just run as it is.

3. Choose Your Hardware Profile and Start Services

- For Nvidia GPU Users:
`docker compose --profile gpu-nvidia up`
- For AMD GPU Users (Linux):
`docker compose --profile gpu-amd up`
- For Mac / Apple Silicon Users:

Option 1: Run everything in Docker (CPU only)

`docker compose --profile cpu up`

Option 2: Run Ollama natively on Mac and connect to it from Docker

1. Install Ollama from <https://ollama.com/download>
2. Run Ollama locally on your Mac
3. Start Docker services:
docker compose up
4. Update Ollama credentials in n8n:
 - Go to <http://localhost:5678/home/credentials>
 - Click on "Local Ollama service"
 - Change base URL to <http://host.docker.internal:11434/>

• For CPU-Only Systems: THIS WILL APPLY TO WINDOWS 11

docker compose --profile cpu up

```
jaspr@PRECISION MINGW64 ~/self-hosted-ai-starter-kit (main)
$ docker compose --profile cpu up
[+] Running 0/6
- qdrant Pulling
- ollama-pull-llama-cpu Pulling
- postgres Pulling
- n8n Pulling
- ollama-cpu Pulling
- n8n-import Pulling
```

```
jaspr@PRECISION MINGW64 ~/self-hosted-ai-starter-kit (main)
$ docker compose --profile cpu up
[+] Running 19/39
- qdrant [#####] 10.47MB / 66.28MB Pulling
  - 8c7716127147 Downloading [====>] 2.097MB/29.78MB
  ✓ 7d870fd66ece Download complete
  - 0641a714b018 Downloading [=====>] 5.243MB/33.37MB
  ✓ ab5cf6b93de6 Download complete
  - 6fc04e638788 Downloading [=====>] 2.179MB/2.179MB
  ✓ 736b7f8a757c Download complete
  ✓ 7c6ccda58d7a Download complete
- ollama-pull-llama-cpu [#####] Pulling
  - 09bd5f2738f8 Downloading [>] 5.243MB/1.906GB
  - 98bf6a5ec929 Pulling fs layer
  - ebc0a999d8e Downloading [=====>] 5.243MB/12.38MB
  - 4b3ffd8ccb52 Downloading [=====>] 3.146MB/29.72MB
- postgres [#####] 8.276MB / 105.7MB Pulling
  ✓ cb12d8d02c19 Download complete
  ✓ c096dc07967e Download complete
  ✓ d3654297bda3 Download complete
  ✓ 8532eda074db Download complete
  ✓ d29e13a2aece Download complete
  ✓ a0e3da949ee5 Download complete
  ✓ 8f1d922bde53 Download complete
  ✓ 5be20dc47298 Download complete
  ✓ 90693fb5a6be Download complete
  - bf5d155c40a7 Downloading [====>] 7.34MB/104.8MB
- n8n [#####] 50.95MB / 270.3MB Pulling
  ✓ 67220c627675 Download complete
  ✓ da9f1bd4a075 Download complete
  - 4b7cc13c92b9 Downloading [=====>] 12.58MB/25.41MB
  ✓ 44a43d49c511 Download complete
  - f1777faed328 Downloading [=====>] 6.376MB/6.376MB
  - 2f385331c129 Downloading [=====>] 14.68MB/51.49MB
  ✓ 4f4fb700ef54 Download complete
  - 586f3a742593 Downloading [=====>] 5.243MB/26.85MB
  ✓ acde3a17a713 Download complete
  ✓ b6cd7e47c1da Download complete
  - 2d35ebdb57d9 Downloading [=====>] 3.802MB/3.802MB
  - 056ebdca45ee Downloading [=>] 4.194MB/152.3MB
- ollama-cpu Pulling
- n8n-import Pulling
```

4. Verify Services Are Running

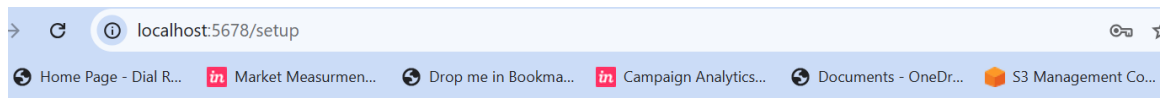
- Check running containers:
- docker-compose ps
- You should see n8n, Ollama, Qdrant, and PostgreSQL running.

```
jasprr@PRECISION MINGW64 ~/self-hosted-ai-starter-kit (main)
$ docker-compose ps
NAME                IMAGE              COMMAND              SERVICE    CREATED      STATUS      PORTS
n8n                 n8nio/n8n:latest  "tini -- /docker-ent..."  n8n       19 minutes ago Up 19 minutes 0.0.0.0:5678->5678/tcp, [
ollama             ollama/ollama:latest "/bin/ollama serve"      ollama-cpu 19 minutes ago Up 19 minutes 0.0.0.0:11434->11434/tcp, [
qdrant             qdrant/qdrant     "./entrypoint.sh"      qdrant    19 minutes ago Up 19 minutes 0.0.0.0:6333->6333/tcp, [
self-hosted-ai-starter-kit-postgres-1 postgres:16-alpine "docker-entrypoint.s..." postgres 19 minutes ago Up 19 minutes (healthy) 5432/tcp
```

Using the n8n Self-Hosted AI Starter Kit

1. Access the n8n Interface

- Open your browser and go to: <http://localhost:5678/>
- Create an account if prompted.
- You'll see the n8n dashboard and workflow editor.



Set up owner account

Email *

First Name *

Last Name *

Password *

8+ characters, at least 1 number and 1 capital letter

2. Core Components

- Ollama (LLM Service): Powers AI capabilities. Models are downloaded on first use.
- Qdrant (Vector DB): For embeddings and similarity search. Web UI at <http://localhost:6333/dashboard>
- PostgreSQL: Persistent storage for n8n workflows.

3. Running Demo Workflows

- The kit includes a demo workflow to showcase AI features.
- You can create, edit, and run workflows using the drag-and-drop editor.
- Integrate AI nodes (Ollama), vector search (Qdrant), and database operations (PostgreSQL) as needed.

4. Pulling AI Models with Ollama

- To add models for text generation or embeddings:
- `docker exec -it n8n-ollama ollama pull mistral`
- `docker exec -it n8n-ollama ollama pull nomic-embed-text`
- Replace "mistral" with other model names as needed (e.g., "llama2", "phi").

Tips

- Harden and secure your setup before using in production.
- Regularly update containers and restrict access for security.
- Explore n8n documentation for advanced workflow examples.