Hosting n8n Installation

Docker Installation

Docker offers the following advantages:

- Installs n8n in a clean environment.
- Easier setup for your preferred database.
- Can avoid issues due to different operating systems, as Docker provides a consistent system.
- Can avoid compatibility issues due to differences in operating systems and tools.
- Makes migrating to new hosts or environments more straightforward.

You can also use n8n in Docker with Docker Compose. You can find Docker Compose configurations for various architectures in the n8n-hosting repository.



Self-hosting knowledge prerequisites

Self-hosting n8n requires technical knowledge, including:

- Setting up and configuring servers and containers
- Managing application resources and scaling
- Securing servers and applications
- Configuring n8n

n8n recommends self-hosting for expert users. Mistakes can lead to data loss, security issues, and downtime. If you aren't experienced at managing servers, n8n recommends n8n Cloud.

Prerequisites

Before proceeding, install Docker Desktop.



Linux Users

Docker Desktop is available for Mac and Windows. Linux users must install Docker Engine and Docker Compose individually for your distribution.



Latest and Next versions

n8n releases a new minor version most weeks. The latest version is for production use. next is the most recent release. You should treat next as a beta: it may be unstable. To report issues, use the forum.

Current latest: 1.99.1 Current next: 1.100.1

Starting n8n

From your terminal, run:

```
docker volume create n8n_data
1
    docker run -it --rm --name n8n -p 5678:5678 -v n8n data:/home/node/.n8n
    docker.n8n.io/n8nio/n8n
```

This command creates a volume to store persistent data, downloads the required n8n image, and starts your container, exposed on port 5678. To save your work between container restarts, it also mounts a docker volume, n8n data, to persist your data locally.

Once running, you can access n8n by opening: http://localhost:5678

Using with PostgreSQL

By default, n8n uses SQLite to save <u>credentials</u>, past executions, and workflows. n8n also supports PostgreSQL, configurable using environment variables as detailed below.

When using PostgreSQL, it's still important to persist the data stored in the /home/node/.n8n folder. This includes n8n user data and, even more importantly, the encryption key for credentials. It's also the name of the webhook when using the n8n tunnel.

If n8n can't find the /home/node/.n8n directory on startup, it automatically creates one. In this case, all existing credentials that n8n saved with a different encryption key will no longer work.



While persisting the /home/node/.n8n directory with PostgreSQL is the recommended best practice, it's not explicitly required. You can provide the encryption key by passing the N8N_ENCRYPTION_KEY environment variable when starting your Docker container.

To use n8n with PostgreSQL, execute the following commands, replacing the placeholders (depicted within angled brackets, for example <POSTGRES_USER>) with your actual values:

```
docker volume create n8n_data
docker run -it --rm \
--name n8n \
```

```
-p 5678:5678 \
6 -e DB_TYPE=postgresdb \
7 -e DB_POSTGRESDB_DATABASE=<POSTGRES_DATABASE> \
8 -e DB_POSTGRESDB_HOST=<POSTGRES_HOST> \
9 -e DB_POSTGRESDB_PORT=<POSTGRES_PORT> \
10 -e DB_POSTGRESDB_USER=<POSTGRES_USER> \
11 -e DB_POSTGRESDB_SCHEMA=<POSTGRES_SCHEMA> \
12 -e DB_POSTGRESDB_PASSWORD=<POSTGRES_PASSWORD> \
13 -v n8n_data:/home/node/.n8n \
14 docker.n8n.io/n8nio/n8n
```

You can find a complete docker-compose file for PostgreSQL in the n8n hosting repository.

Setting timezone

To define the timezone n8n should use, you can set the GENERIC_TIMEZONE environment variable. Schedule-oriented nodes, like the Schedule Trigger node use this to determine the correct timezone.

You can set the system timezone, which controls what some scripts and commands like date return, using the TZ environment variable.

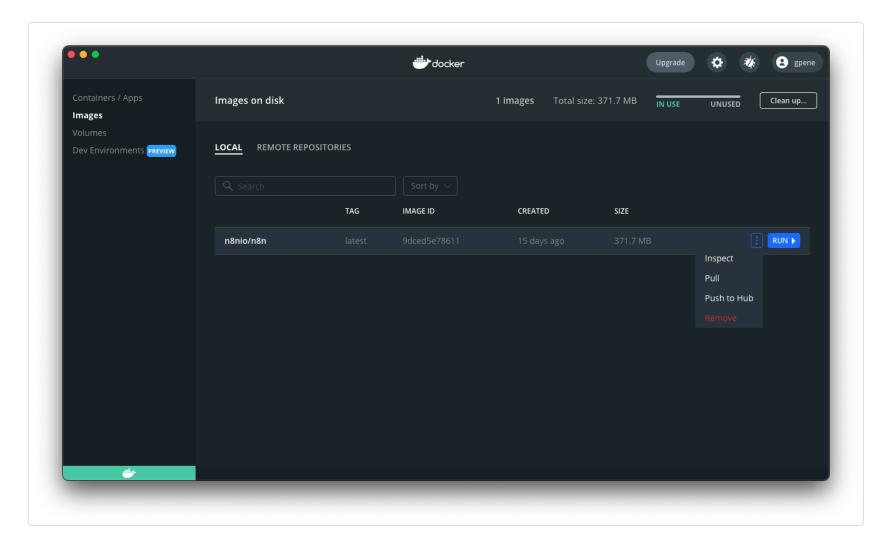
This example sets the same timezone for both variables:

```
1 docker volume create n8n_data
2
```

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

Updating

To update n8n, in Docker Desktop, navigate to the **Images** tab and select **Pull** from the context menu to download the latest n8n image:



You can also use the command line to pull the latest, or a specific version:

```
# Pull latest (stable) version
docker pull docker.n8n.io/n8nio/n8n

# Pull specific version
docker pull docker.n8n.io/n8nio/n8n:1.81.0

# Pull next (unstable) version
docker pull docker.n8n.io/n8nio/n8n:next
```

After pulling the updated image, stop your n8n container and start it again. You can also use the command line. Replace <container_id> in the commands below with the container ID you find in the first command:

```
# Find your container ID
     docker ps -a
2
 3
4
     # Stop the container with the `<container_id>`
 5
     docker stop <container_id>
6
     # Remove the container with the `<container id>`
     docker rm <container id>
8
9
     # Start the container
10
     docker run --name=<container_name> [options] -d docker.n8n.io/n8nio/n8n
11
```

Updating Docker Compose

If you run n8n using a Docker Compose file, follow these steps to update n8n:

```
# Navigate to the directory containing your docker compose file
cd </path/to/your/compose/file/directory>

# Pull latest version
docker compose pull

# Stop and remove older version
docker compose down

# Start the container
docker compose up -d
```

Further reading

You can find more information about Docker setup in the README file for the Docker image.

n8n with tunnel



Use this for local development and testing. It isn't safe to use it in production.

To use webhooks for trigger nodes of external services like GitHub, n8n has to be reachable from the web. n8n runs a tunnel service that can redirect requests from n8n's servers to your local n8n instance.

Start n8n with --tunnel by running:

```
docker volume create n8n_data

docker run -it --rm \
--name n8n \
-p 5678:5678 \
-v n8n_data:/home/node/.n8n \
docker.n8n.io/n8nio/n8n \
start --tunnel
```

Next steps

- Learn more about configuring and scaling n8n.
- Or explore using n8n: try the Quickstarts.