



Update Cloud-Hosted and Locally-Hosted N8N Instances (Render & Docker)

Overview

This SOP explains how to safely upgrade both a cloud-hosted N8N instance running on Render and a self-hosted N8N instance running in Docker on your local machine. By following these steps you will deploy the latest N8N image while preserving all workflows, credentials, and environment variables. When finished, you will have two fully updated instances—one in the cloud and one on your PC—without losing any data.

ATTENTION! THIS METHOD WORKS ONLY IF YOU HAVE INSTALLED N8N AS I HAVE SHOWN IN THE PREVIOUS VIDEOS WITH SUPABASE FOR RENDER UPDATE AND LOCAL PATH FOR LOCAL UPDATE!

Step-by-Step Guide

1. Prepare for Both Updates

- Ensure you have administrative access to your Render account and Docker Desktop.
- Verify that all critical workflows can tolerate a short outage during the upgrade.
- Keep a copy of your N8N environment variables and the host data directory ready for reference.

2. Update Cloud-Hosted N8N on Render

- Log in to your **Render** dashboard.
- Click your **N8N** service/project.
- Select **Manual Deploy** → **Deploy latest reference**.

- d. Read the warning that active workflows will pause, then click **Deploy** to confirm.
- e. Wait until Render reports **Editor is now accessible**, then refresh your browser to confirm the new N8N version.

3. Update Locally Hosted N8N in Docker

a. Inspect the current container

- i. Open Docker Desktop and select your `n8n` container.
- ii. Click **Inspect** and record:
 - **Host path** under `HostConfig.Binds` (e.g., `C:\Users\<YOU>\Documents\n8n-data:/home/node/.n8n`).
 - **Environment variables** under `Env` (copy each line).

b. Remove the existing container and image

- i. Stop the `n8n` container.
- ii. Right-click → **Delete** the container (no volumes will be lost because data is in the bind mount).
- iii. Go to **Images**, locate the `n8n` image, and delete it.

c. Pull the latest N8N image

- i. In Docker Desktop, click **Pull** and search for `n8nio/n8n:latest`.
- ii. Wait for the download to complete.

d. Run a new container with identical settings

- i. Click **Run** on the freshly pulled image.
- ii. Fill in the dialog exactly as before:
 - **Container name:** `n8n-secure` (or your previous name).
 - **Host port:** `5555` (replace with your original port).
 - **Host path:** browse to your local `n8n-data` folder.
 - **Container path:** `/home/node/.n8n`.
- iii. Re-add each environment variable you saved, for example:

```
WEBHOOK_URL=https://example.com
N8N_PROTOCOL=https
N8N_HOST=example.com
N8N_DEFAULT_BINARY_DATA_MODE=filesystem
```

iv. Click **Run** to start the container.

e. **Verify the update**

- i. Open `https://<host>:5555` (or your configured URL).
- ii. Confirm the version banner shows the newest N8N release and that all workflows load correctly.

f. **(Bonus) Quick One-Line Deployment**

- i. Copy and customise this single command if you prefer a one-shot build-and-run workflow:

```
docker run -d --name n8n-secure -p 5555:5678 \
-e N8N_PROTOCOL=https \
-e N8N_HOST=<your-domain> \
-e N8N_EDITOR_BASE_URL=https://<your-domain> \
-e WEBHOOK_URL=https://<your-domain> \
-e N8N_COMMUNITY_PACKAGES_AVAILABLE=true \
-e N8N_DEFAULT_BINARY_DATA_MODE=filesystem \
-v C:\Users\<YOUR_USERNAME>\Documents\n8n-data:/home/n8n \
n8nio/n8n
```

- ii. Replace `<your-domain>` and `<YOUR_USERNAME>` with your own details, then press **Enter**.
- iii. Browse to `https://<your-domain>` and verify your workflows appear intact.
- iv. Click terminal in DOCKER lower right corner - then write this whole code on one line with your domain and port and host folder and press enter and your n8n will be installed from the new image.(You have to delete and re pull the latest image before this process)

Warnings and Notes

- **Downtime:** Deploying on Render or restarting the Docker container will briefly stop all active workflows. Schedule updates during low-traffic periods.
 - **Persistent Data:** If you skip the bind mount (`/home/node/.n8n`) or forget to re-enter environment variables, your workflows **will** disappear. Always double-check paths and variables before running the new container.
 - **Rollback:** Keep the old image tag (e.g., `n8n:1.46.0`) noted somewhere. If the new version causes issues, you can re-pull and run the earlier image with the same bind mount to restore functionality.
-

Visual Diagram

graph TD

A[Log in to Render] → B[Deploy Latest Reference]

B → C[Wait & Refresh]

C → D[Open Docker Desktop]

D → E[Inspect N8N Container]

E → F[Delete Old Container & Image]

F → G[Pull Latest N8N Image]

G → H[Run New Container with Env & Bind]

H → I[Bonus One-Line Deployment]

I → J[Verify Updated Instances]

End of SOP