

# Install n8n in WSL2



guide me in installing n8n in wsl2

Here is a **step-by-step guide to installing n8n in WSL2** (Windows Subsystem for Linux 2) on Ubuntu:

## ✓ Step 1: Ensure Prerequisites Are Installed

Make sure your WSL2 is set up with Ubuntu (20.04 or 22.04 recommended) and updated:

```
bash

sudo apt update && sudo apt upgrade -y
```

You'll need:

- **Node.js (LTS version)**
- **npm**
- **n8n**
- **Optional: PM2 or Docker** (for running n8n persistently)

## ✓ Step 2: Install Node.js & npm

Install Node.js LTS version using NodeSource:

```
bash

curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs
```

Verify installation:

```
bash

node -v
npm -v
```

## ✓ Step 3: Install n8n

Install n8n globally via npm:

```
bash

npm install n8n -g
```

Check if n8n installed:

```
bash

n8n --version
```

## ✅ Step 4: Run n8n

Just run:

```
bash

n8n
```

It will start a local instance accessible via:

```
arduino

http://localhost:5678
```

📌 If accessing from Windows browser, use: **http://127.0.0.1:5678**

## ✅ Step 5 (Optional): Run n8n in Background with PM2

Install PM2:

```
bash

npm install pm2 -g
```

Start n8n with PM2:

```
bash

pm2 start n8n
pm2 startup
pm2 save
```

## ✅ Step 6 (Optional): Use .env for Custom Configuration

Create a **.env** file for settings like credentials encryption key or webhook URLs:

```
bash

nano ~/.n8n/.env
```

Example content:

```
env

N8N_BASIC_AUTH_ACTIVE=true
N8N_BASIC_AUTH_USER=admin
N8N_BASIC_AUTH_PASSWORD=yourpassword
```

Then run with:

```
bash

n8n
```

## ✅ Step 7 (Optional): Expose n8n to Public Internet with Tunneling (e.g., for Webhooks)

Use tools like **ngrok**:

```
bash

ngrok http 5678
```

Would you like to use Docker instead? I can guide you through that too.



will this n8n saves our work

Yes, **by default, n8n running locally in WSL2 saves your workflows in memory only** — so if you close the terminal or reboot, **your work is lost** unless you configure **persistent storage**.

## ✅ To Persist Your Workflows, You Have 3 Options:

### ◆ Option 1: Use n8n with Local Filesystem Database (recommended for WSL2)

By default, n8n uses SQLite, but you must specify a **folder for the database and config**:

```
bash

mkdir -p ~/.n8n
```

Then run n8n like this:

```
bash

n8n --data-folder ~/.n8n
```

This makes sure it saves all workflows and credentials under `~/n8n`.

### ◆ Option 2: Use PM2 to Keep it Running (Auto-Restart)

```
bash

pm2 start n8n --name n8n -- --data-folder ~/n8n
pm2 save
```

Your workflows and credentials will persist in the `~/n8n` folder.

### ◆ Option 3: Use External Database (PostgreSQL, MySQL)

For larger-scale or team setups:

- Setup a database (e.g., PostgreSQL)
- Set these environment variables in `.env` or as inline before running `n8n`:

```
bash

DB_TYPE=postgresdb
DB_POSTGRESDB_HOST=localhost
DB_POSTGRESDB_PORT=5432
DB_POSTGRESDB_DATABASE=n8n
DB_POSTGRESDB_USER=youruser
DB_POSTGRESDB_PASSWORD=yourpass
```

### Important Notes:

- Workflows, credentials, execution history, etc. are stored in the data folder (SQLite or external DB).
- Always use `--data-folder` to make data persistent across sessions.
- Don't rely on just `n8n` without any config — it's for testing only.

Let me know if you want a sample `.env` file or full Docker setup for persistent storage.



now guide the n8n installation through docker

Here's a **complete guide to install and run n8n using Docker inside WSL2**, with **persistent storage** and optional **basic authentication**:

## Step-by-Step: Install n8n with Docker in WSL2

### Prerequisites

## 1. WSL2 installed with Ubuntu

## 2. Docker installed inside WSL2

*(If not done, ask me for Docker installation in WSL2)*

### Step 1: Create a Project Folder

```
bash

mkdir -p ~/n8n-docker
cd ~/n8n-docker
```

### Step 2: Create Docker Compose File

Create `docker-compose.yml`:

```
bash

nano docker-compose.yml
```

Paste this content:

```
yml

version: "3.1"

services:
  n8n:
    image: n8nio/n8n
    restart: always
    ports:
      - "5678:5678"
    environment:
      - N8N_BASIC_AUTH_ACTIVE=true
      - N8N_BASIC_AUTH_USER=admin
      - N8N_BASIC_AUTH_PASSWORD=yourpassword
      - N8N_HOST=localhost
      - N8N_PORT=5678
      - N8N_PROTOCOL=http
      - N8N_EDITOR_BASE_URL=http://localhost:5678
    volumes:
      - ./n8n_data:/home/node/.n8n
```

 Replace **yourpassword** with a secure password.

### Step 3: Start the Container

Run:

```
bash

docker-compose up -d
```

### Step 4: Access n8n in Browser

Open:

```
arduino

http://localhost:5678
```

Login with:

- **Username:** admin
- **Password:** yourpassword (from above)

## Step 5: Persistence Confirmed

All workflows, credentials, and settings are saved in:

```
bash

~/n8n-docker/n8n_data/
```

You can back up this folder or move it across systems.

## Common Commands

- Restart service:

```
bash

docker-compose restart
```

- Stop service:

```
bash

docker-compose down
```

- View logs:

```
bash

docker-compose logs -f
```

## Optional: Auto-start on WSL Boot

Add this to your `.bashrc` or `.zshrc`:

```
bash

cd ~/n8n-docker && docker-compose up -d
```

Let me know if you'd like to use PostgreSQL or enable webhook tunneling with **ngrok**.



how to start n8n service

To **start the n8n service**, the method depends on **how you installed it**. Here's how to start it based on your setup:



## If You Installed n8n via Docker Compose:

Go to the project directory where **docker-compose.yml** is:

```
bash

cd ~/n8n-docker
```

Then start the service:

```
bash

docker-compose up -d
```

✅ **-d** means "detached" (runs in the background)

To check if it's running:

```
bash

docker ps
```



## To Restart or Stop Docker-Based n8n:

```
bash

docker-compose restart    # Restart
docker-compose down       # Stop
```



## If You Installed n8n via npm (native):

**Start manually:**

```
bash

n8n --data-folder ~/.n8n
```

This runs n8n in the terminal. Keep it open or use background tools like **tmux** or **pm2**.

---

## Start with **pm2** (recommended for always-on):

If using **pm2**:

```
bash

pm2 start n8n --name n8n -- --data-folder ~/.n8n
pm2 save
```

Check status:

```
bash

pm2 list
```

---

## Access n8n in Your Browser:

```
text

http://localhost:5678
```

If you're using basic auth, enter your credentials when prompted.

---

Let me know your install method (Docker or npm) and I'll tailor next steps like auto-start on reboot or database config.