

N8N Installation process



[Link to Automation Avenue platform](#)

Document was created for [THIS](#) youtube video.

I guess you've heard of the n8n platform already.

It's great workflow automation software that lets you easily create AI Agents and automation workflows , and you don't need any special programming skills to get started.

This solution is called 'no code' or 'low code', so basically anyone can start playing with it without any previous experience.

N8N can be run either free of charge or as a paid service, you can discover more on n8n website:

<https://n8n.io/>

To be able to run it free of charge - you need to self-host it, which means you need a PC or laptop or server or anything really that is able to run either Docker or Proxmox (there is also a way to just install with npm / node.js but we will not cover it as its not recommended way of installing n8n).

Here you will learn how to install n8n 3 different ways:

1. Using 'docker run' command
2. Using 'docker-compose.yml' file
3. On Proxmox using 'helper scripts'

Method 1:

On Windows or on MAC you will need to have Docker Desktop installed or similar program that will let you run docker containers.

On any Linux distribution no pre-requisites are needed.

Below are commands for Debian / Ubuntu distribution, but they will be very similar for any other linux distributions.

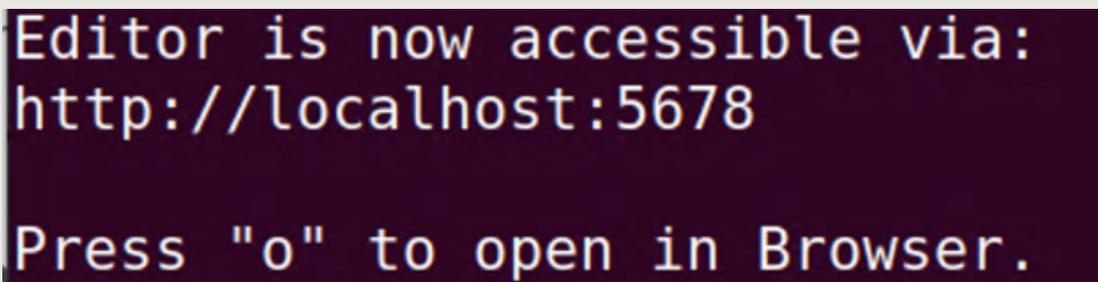
List of commands to run:

```
sudo apt-get update && sudo apt-get upgrade -y
sudo apt install docker.io -y
systemctl status docker
sudo usermod -aG docker $USER
newgrp docker
docker run hello-world
docker volume create n8n_data
docker volume ls
docker run -it --rm --name n8n -p 5678:5678 -v n8n_data:/home/node/.n8n docker.n8n.io/n8nio/n8n
```

What these commands do:

- first one will update and upgrade you current packages on your server
- next one will install Docker (and auto-answer 'yes' to any questions during installation)
- check if docker is up and running
- add your user to docker group so you don't have to use 'sudo' when running docker commands
- reload the docker group to make the changes live
- runs full docker check and displays 'hello world' if docker is able to pull the image
- creates docker volume called 'n8n_data
- checks if the volume indeed exists
- runs n8n service which can be then accessed by going to url: <http://localhost:5678>

Thats basically it, and when the process starts correctly - you should see this in your terminal:



```
Editor is now accessible via:  
http://localhost:5678  
  
Press "o" to open in Browser.
```

So just open any browser, copy-paste that <http://localhost:5678> address and paste it there to start n8n service.

Method 2:

This method is very similar, we will simply use docker-compose.yml file to run the service. Its basically the same command as above, but saved as yaml format in docker-compose.yml and can be started by running 'docker-compose up -d' command.

To run it as docker-compose file, you need to install docker and docker-compose. Commands to run:

```
sudo apt-get update && sudo apt-get upgrade -y  
sudo apt install docker.io docker-compose -y  
systemctl status docker  
sudo usermod -aG docker $USER  
newgrp docker  
docker volume create n8n_data  
docker volume ls
```

Which means up to this point they are exactly the same commands that we would run for docker run, except of the docker-compose component added in second command.

Then we need to create docker-compose.yml file, which should look like that:

```
services:
  n8n:
    image: docker.n8n.io/n8nio/n8n
    container_name: n8n
    ports:
      - "5678:5678"
    volumes:
      - n8n_data:/home/node/.n8n
    stdin_open: true
    tty: true
    restart: "no"

volumes:
  n8n_data:
    external: true
```

So use any text editor, paste the text below, then save as docker-compose.yml file:

```
services:
  n8n:
    image: docker.n8n.io/n8nio/n8n
    container_name: n8n
    ports:
      - "5678:5678"
    volumes:
      - n8n_data:/home/node/.n8n
    stdin_open: true
    tty: true
    restart: "no"

volumes:
  n8n_data:
    external: true
```

Once you've got that file, being in the same folder where this docker compose file is located you need to run:

```
docker-compose up -d
```

to start the service

```
docker ps
```

to see if container is up and running

```
docker-compose stop
```

to stop the service and

```
docker-compose rm
```

to remove the service (it will not remove the docker volume though which is probably what we need to preserve)

Method 3 :

Run one line command to automatically deploy LXC container on Proxmox server.

Visit [THIS](#) website where you will find many more helper scripts, search for ' n8n', then just copy-paste the long command to Proxmox shell.

Currently the command you need to run is:

```
bash -c "$(curl -fsSL https://raw.githubusercontent.com/community-scripts/ProxmoxVE/main/ct/n8n.sh)"
```

however its advisable to check it on the website instead and review entire code every time you want to run this command.

Hope that helps, please remember to visit Automation Avenue platform if you want to learn automation, DevOps, Python programming, Cloud, Linux, Databases and more.

Please just click [THIS](#) link to visit our website!

Thank you - Marek