

คู่มือติดตั้ง n8n บนคอมพิวเตอร์ส่วนตัว

ความต้องการระบบ

ขั้นต่ำ:

- **RAM:** 2GB ขึ้นไป (แนะนำ 4GB+)
 - **Storage:** 10GB พื้นที่ว่าง
 - **OS:** Windows 10+, macOS 10.15+, หรือ Linux Ubuntu 18.04+
 - **Internet:** การเชื่อมต่ออินเทอร์เน็ตเสถียร
-

วิธีติดตั้ง (เลือก 1 วิธี)

วิธีที่ 1: ติดตั้งผ่าน Docker (แนะนำ)

 ข้อดี: ง่าย ปลอดภัย ไม่กระทบระบบหลัก

ขั้นตอนที่ 1: ติดตั้ง Docker

Windows:

1. ดาวน์โหลด [Docker Desktop for Windows](#)
2. เรียกใช้ไฟล์ที่ดาวน์โหลด
3. ทำตามขั้นตอนการติดตั้ง
4. รีสตาร์ทคอมพิวเตอร์
5. เปิด Docker Desktop และรอจนกว่าจะพร้อมใช้งาน

macOS:

1. ดาวน์โหลด [Docker Desktop for Mac](#)
2. ลาก Docker.app ไป Applications folder
3. เปิด Docker จาก Applications
4. ทำตามขั้นตอน setup

Linux (Ubuntu/Debian):

```
bash
```

```
# อัปเดตระบบ
```

```
sudo apt update
```

```
# ติดตั้ง dependencies
```

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

```
# เพิ่ม Docker GPG key
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
# เพิ่ม Docker repository
```

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release
```

```
# ติดตั้ง Docker
```

```
sudo apt update
```

```
sudo apt install docker-ce
```

```
# เพิ่มผู้ใช้ปัจจุบันเข้า docker group
```

```
sudo usermod -aG docker $USER
```

```
# ลอกเอาท์และล็อกอินใหม่
```

ขั้นตอนที่ 2: ติดตั้ง n8n

สร้าง Docker Volume (เก็บข้อมูล)

```
bash
```

```
docker volume create n8n_data
```

รัน n8n Container

```
bash
```

```
# รันแบบ basic
```

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

```
# หรือรันแบบ production (แนะนำ)
```

```
docker run -d --restart=unless-stopped --name n8n -p 5678:5678 -v n8n_data:/home/node/.n8n dock
```

เข้าใช้งาน

วิธีที่ 2: ติดตั้งผ่าน NPM

✅ ข้อดี: ควบคุมได้มาก แก้ไขง่าย

ขั้นตอนที่ 1: ติดตั้ง Node.js

Windows & macOS:

1. ดาวน์โหลด [Node.js LTS](#) (เวอร์ชัน 18.10+ หรือ 20.6+)
2. เรียกใช้ installer และทำตามขั้นตอน
3. เปิด Command Prompt (Windows) หรือ Terminal (macOS)
4. ตรวจสอบการติดตั้ง:

```
bash
```

```
node --version
```

```
npm --version
```

Linux:

```
bash
```

```
# ติดตั้ง Node.js 20.x
```

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -  
sudo apt-get install -y nodejs
```

```
# ตรวจสอบ
```

```
node --version
```

```
npm --version
```

ขั้นตอนที่ 2: ติดตั้ง n8n

```
bash
```

```
# ติดตั้ง n8n global  
npm install n8n -g
```

```
# รัน n8n  
n8n start
```

```
# หรือรันแบบกำหนด port  
n8n start --port 5678
```

เข้าใช้งาน

เปิดเบราว์เซอร์ไปที่: <http://localhost:5678>

วิธีที่ 3: ใช้ npx (ไม่ต้องติดตั้ง)

 ข้อดี: ทดลองได้ง่าย ไม่เปลืองพื้นที่

```
bash
```

```
# รันทันทีโดยไม่ต้องติดตั้ง  
npx n8n
```

```
# หรือระบุ port  
npx n8n start --port 5678
```

การตั้งค่าเริ่มต้น

1. สร้าง Admin Account

เมื่อเข้า <http://localhost:5678> ครั้งแรก:

1. กรอก **Email** และ **Password**
2. กรอก **First Name** และ **Last Name**
3. คลิก **Create Account**

2. ตั้งค่าพื้นฐาน

```
bash
```

```
# สร้างไฟล์ environment variables
# Windows: สร้างไฟล์ .env ใน C:\Users\[Username]\.n8n\
# macOS/Linux: สร้างไฟล์ .env ใน ~/.n8n/

# เนื้อหาไฟล์ .env:
N8N_HOST=localhost
N8N_PORT=5678
N8N_PROTOCOL=http
WEBHOOK_URL=http://localhost:5678/
```

3. ตั้งค่า Database (เพื่อประสิทธิภาพ)

SQLite (Default - ง่าย)

```
bash
```

```
# ไม่ต้องตั้งค่าอะไร ไปได้เลย
```

PostgreSQL (แนะนำสำหรับ Production)

```
bash
```

```
# ติดตั้ง PostgreSQL ก่อน
# จากนั้นเพิ่มใน .env:
DB_TYPE=postgresdb
DB_POSTGRESDB_HOST=localhost
DB_POSTGRESDB_PORT=5432
DB_POSTGRESDB_DATABASE=n8n
DB_POSTGRESDB_USER=n8n_user
DB_POSTGRESDB_PASSWORD=your_password
```



การรันแบบ Production

Docker Compose (แนะนำ)

สร้างไฟล์ `docker-compose.yml`:

yaml

version: '3.8'

services:

n8n:

image: docker.n8n.io/n8nio/n8n

container_name: n8n

restart: unless-stopped

ports:

- "5678:5678"

environment:

- N8N_HOST=localhost
- N8N_PORT=5678
- N8N_PROTOCOL=http
- NODE_ENV=production
- WEBHOOK_URL=http://localhost:5678/
- GENERIC_TIMEZONE=Asia/Bangkok

volumes:

- n8n_data:/home/node/.n8n

depends_on:

- postgres

postgres:

image: postgres:13

container_name: n8n_postgres

restart: unless-stopped

environment:

- POSTGRES_USER=n8n
- POSTGRES_PASSWORD=n8n_password
- POSTGRES_DB=n8n

volumes:

- postgres_data:/var/lib/postgresql/data

volumes:

n8n_data:

postgres_data:

รันด้วย:

bash

docker-compose up -d

การแก้ไขปัญหาค้างที่พบบ่อย

ปัญหา: Port 5678 ถูกใช้แล้ว

```
bash

# ตรวจสอบว่าใครใช้ port
# Windows:
netstat -ano | findstr :5678

# macOS/Linux:
lsof -i :5678

# แก้ไข: เปลี่ยน port
n8n start --port 3000
```

ปัญหา: Docker ไม่รัน

```
bash

# ตรวจสอบ Docker status
docker --version
docker ps

# รีสตาร์ท Docker service
# Windows: รีสตาร์ท Docker Desktop
# Linux:
sudo systemctl restart docker
```

ปัญหา: Permission denied (Linux)

```
bash

# เพิ่มสิทธิ์ Docker
sudo usermod -aG docker $USER

# ล็อกเอาท์และล็อกอินใหม่
```

ปัญหา: Node.js เวอร์ชันเก่า

bash

อัปเดต Node.js

Windows/macOS: ดาวน์โหลดใหม่จาก nodejs.org

Linux:

`curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -`

`sudo apt-get install -y nodejs`

การอัปเดต n8n

Docker:

bash

หยุด container เก่า

`docker stop n8n`

`docker rm n8n`

ดาวน์โหลด image ใหม่

`docker pull docker.n8n.io/n8nio/n8n`

รัน container ใหม่

`docker run -d --restart=unless-stopped --name n8n -p 5678:5678 -v n8n_data:/home/node/.n8n dock`



NPM:

bash

อัปเดต n8n

`npm update -g n8n`

หรือถอนแล้วติดตั้งใหม่

`npm uninstall -g n8n`

`npm install -g n8n`

Checklist การติดตั้งสำเร็จ






- ☐ Docker/Node.js ติดตั้งเรียบร้อยแล้ว
- ☐ n8n รันได้โดยไม่มี error
- ☐ เข้า <http://localhost:5678> ได้
- ☐ สร้าง admin account สำเร็จ

- ☐ ทดสอบสร้าง workflow ง่ายๆ
 - ☐ ตั้งค่า environment variables (ถ้าต้องการ)
 - ☐ Backup ข้อมูล (n8n_data volume)
-

ขั้นตอนถัดไป

1. เรียนรู้ **Basic Workflows** - สร้าง workflow แรกด้วย HTTP Request
 2. เชื่อมต่อ **Apps** - ทดลองเชื่อมต่อ Gmail, Google Sheets
 3. ศึกษา **Community** - เข้าร่วม [n8n Community](#)
 4. ใช้ **Templates** - ลองใช้ workflow templates ที่มีอยู่
 5. เรียนรู้ **Advanced Features** - webhooks, custom functions, AI nodes
-

คำแนะนำความปลอดภัย

-  เปลี่ยน default port หากใช้งานจริง
-  ตั้งรหัสผ่านที่แข็งแกร่ง
-  Backup ข้อมูลเป็นประจำ
-  อัปเดต n8n เป็นประจำ
-  อย่าเปิด port ออก internet โดยไม่มี security

 ยินดีด้วย! คุณพร้อมใช้งาน n8n แล้ว