



n8n selbst hosten: Schritt-für-Schritt Anleitung für deinen eigenen Automation Server

Diese Anleitung führt dich durch die komplette Einrichtung eines n8n-Servers auf Hetzner Cloud mit allen wichtigen Sicherheitsfeatures wie HTTPS, Firewall und automatischen Backups.

Hetzner Cloud Account & Projekt

Registriere dich unter <https://console.hetzner.cloud> und lege ein neues Projekt an, z. B. „n8n-server“.

Server erstellen

Erstelle einen neuen Server mit folgenden Einstellungen:

- Standort: Nürnberg, Falkenstein oder Helsinki (DSGVO-konform)
- Image: Ubuntu 24.04
- Typ: CPX11 (Shared vCPU)
- Netzwerk: Standard (IPv4 & IPv6)
- SSH Key: leer lassen (folgt später)
- Volume direkt hinzufügen:
- Name: n8n-data-volume
- Größe: 10 GB
- Ort: identisch zum Serverstandort
- Firewall: nicht auswählen
- Backup: nicht aktivieren (kommt gleich)
- Alle anderen Felder leer lassen
- Name: z. B. n8n-server
- Klicke auf Erstellen

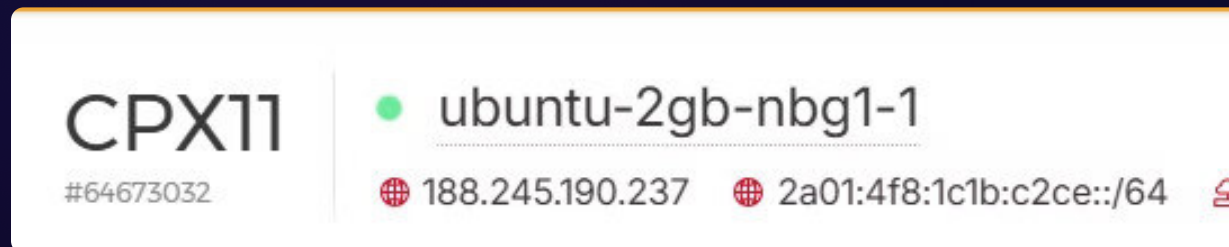
Du erhältst nach dem Start eine E-Mail mit IP-Adresse, Benutzernamen (root) und Passwort.

Mit dem Server verbinden

Öffne auf Windows die Eingabeaufforderung (Win+R → cmd) und gib ein:

```
ssh root@DEINE_SERVER_IP
```

Bestätige beim ersten Mal mit yes und gib das Passwort aus der E-Mail ein. Danach wirst du aufgefordert, ein neues Root-Passwort zu setzen.



Basis konfigurieren

Backups aktivieren

Gehe in der Hetzner Cloud Console auf deinen Server → Tab „Backups“ → Aktiviere Backups. Das kostet ca. 20 % Aufpreis (z. B. 0,80 €/Monat für CPX11) und speichert 7 tägliche Snapshots automatisch.

Firewall einrichten

Gehe in der Hetzner Console → Menü „Firewalls“ → Neue Firewall:

- Name: n8n-firewall
- Eingehende Regeln:
- TCP 22 (SSH)
- TCP 80 (HTTP)
- TCP 443 (HTTPS)

Weisen die Firewall deinem Server zu und speichere.



Docker & Compose installieren

System aktualisieren & Docker installieren

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

```
apt install -y docker.io
```

```
systemctl enable docker
```

```
systemctl start docker
```

Docker Compose installieren

```
curl -L "https://github.com/docker/compose/releases/download/v2.27.0/docker-compose-$(uname -s)-$(uname -m)" -  
o /usr/local/bin/docker-compose
```

```
chmod +x /usr/local/bin/docker-compose
```

Projekt konfigurieren

Verzeichnis & Konfigurationsdateien anlegen

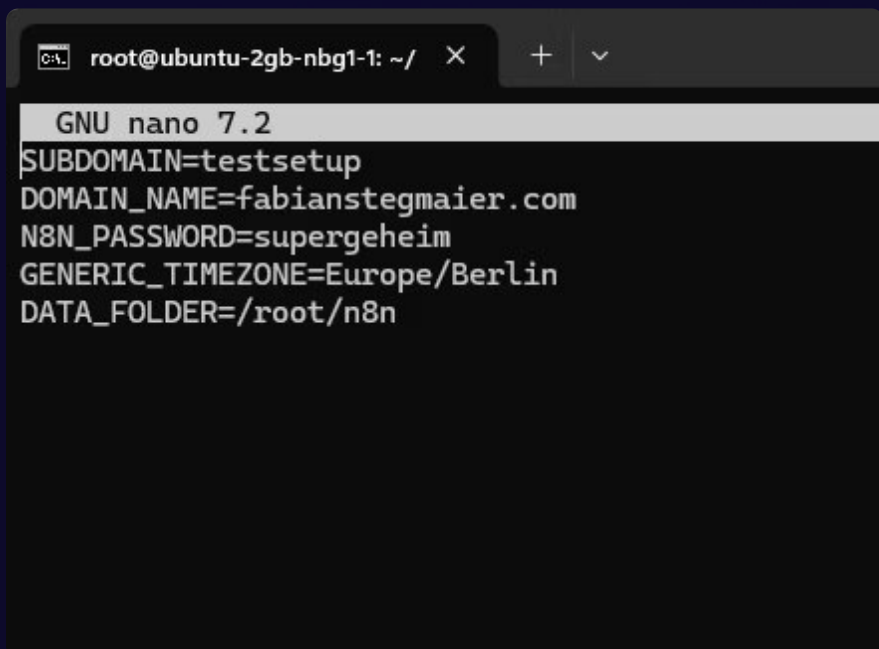
```
mkdir -p /root/n8n
```

```
cd /root/n8n
```

.env Datei erstellen

```
nano .env
```

Inhalt:



```
root@ubuntu-2gb-nbg1-1: ~/  X  +  v  
GNU nano 7.2  
SUBDOMAIN=testsetup  
DOMAIN_NAME=fabianstegmaier.com  
N8N_PASSWORD=supergeheim  
GENERIC_TIMEZONE=Europe/Berlin  
DATA_FOLDER=/root/n8n
```

```
SUBDOMAIN=testsetup  
DOMAIN_NAME=fabianstegmaier.com  
N8N_PASSWORD=supergeheim  
GENERIC_TIMEZONE=Europe/Berlin  
DATA_FOLDER=/root/n8n  
N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=true  
N8N_RUNNERS_ENABLED=true
```

Speichern mit CTRL + O, ENTER → Schließen mit CTRL + X.

docker-compose.yml erstellen

```
nano docker-compose.yml
```

Inhalt:

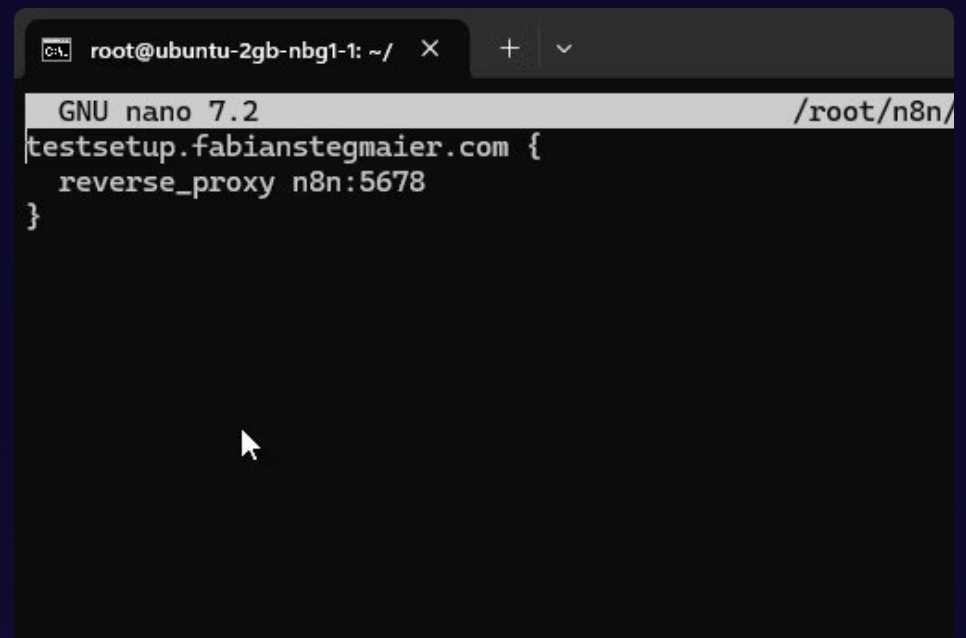
```
services:
  caddy:
    image: caddy:latest
    restart: unless-stopped
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - caddy_data:/data
      - ${DATA_FOLDER}/caddy_config:/config
      - ${DATA_FOLDER}/caddy_config/Caddyfile:/etc/caddy/Caddyfile
  n8n:
    image: n8nio/n8n
    restart: always
    ports:
      - 5678:5678
    environment:
      - N8N_HOST=${SUBDOMAIN}.${DOMAIN_NAME}
      - N8N_PORT=5678
      - N8N_PROTOCOL=https
      - N8N_BASIC_AUTH_ACTIVE=true
      - N8N_BASIC_AUTH_USER=admin
      - N8N_BASIC_AUTH_PASSWORD=${N8N_PASSWORD}
      - NODE_ENV=production
      - WEBHOOK_URL=https://${SUBDOMAIN}.${DOMAIN_NAME}/
      - GENERIC_TIMEZONE=${GENERIC_TIMEZONE}
      - N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=${N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS}
      - N8N_RUNNERS_ENABLED=${N8N_RUNNERS_ENABLED}
    volumes:
      - n8n_data:/home/node/.n8n
      - ${DATA_FOLDER}/local_files:/files
volumes:
  caddy_data:
    external: true
  n8n_data:
    external: true
```

Caddyfile anlegen

```
mkdir -p /root/n8n/caddy_config  
  
nano /root/n8n/caddy_config/Caddyfile
```

Inhalt:

```
testsetup.fabianstegmaier.com {  
  reverse_proxy n8n:5678  
}
```



Docker starten

```
docker volume create caddy_data  
  
docker volume create n8n_data  
  
docker-compose up -d
```

Domain verbinden

DNS-A-Record setzen

Gehe zu deinem Domainanbieter (z. B. IONOS) → DNS-Einstellungen → A-Record anlegen:

- Typ: A
- Name: testsetup
- Ziel: IP deines Hetzner-Servers

Warte ca. 5–30 Minuten, bis die DNS-Änderung greift.

✓ Zugriff & Login

Rufe <https://testsetup.fabianstegmaier.com> im Browser auf.

Login mit:

- Benutzer: admin
- Passwort: aus der .env Datei

Du hast jetzt einen vollständig funktionierenden, abgesicherten n8n-Server mit HTTPS, Authentifizierung, Backup und Firewall – bereit für Automationen! 🎉



Set up owner account

Email *

First Name *

Last Name *

Password *

8+ characters, at least 1 number and 1 capital letter

☐ I want to receive security and product updates

Next

HTTPS-Verschlüsselung

Automatisch durch Caddy
konfiguriert mit Let's Encrypt
Zertifikaten

Firewall-Schutz

Nur die notwendigen Ports 22, 80
und 443 sind geöffnet

Automatische Backups

Tägliche Snapshots für 7 Tage
durch Hetzner Cloud