Free (Mini) Kurs · n8n auf Raspberry Pi im LAN mit Docker

Ziel: n8n läuft stabil im Heimnetz (LAN-only, HTTP) auf Raspberry Pi OS Lite 64-bit – mit Persistenz, Docker Compose und praxisnaher Fehlerbehebung. **Nach dem Kurs erreichbar unter:** `http://core.local:5678` (oder `http://IP-ADRESSE:5678`)

Hinweis (Meta-Posts): Für das automatische Senden von Meta-Posts (ausgehende API-Requests) ist **kein eingehendes HTTPS** erforderlich. HTTPS/Reverse Proxy ist erst nötig, wenn externe Dienste **Webhooks zu Dir senden** sollen.

Schnellstart

- 1) **Docker & Compose installieren** → 2.1 + 2.2 + 2.3
- 2) **n8n im LAN starten** → 3.1 + 3.2 + 3.3
- 3) **Fehler schnell diagnostizieren** → 4.1 + 4.2 + 4.3
- 4) **Python aktivieren (optional)** → 5.1 + 5.2 + 5.3
- 5) **Betrieb & Updates** → 6.1 + 6.2

Voraussetzungen

- Raspberry Pi (Pi 4 / 2 GB reicht für den Einstieg)
- Raspberry Pi OS Lite **64-bit**
- SSH-Zugriff (Terminal)
- Internet für Docker-Installation & Images
- Browser im gleichen LAN

Kursmaterial (Download / Handout)

Lege die Dateien z. B. unter `/srv/docker/n8n/` ab (bzw. als Wiki-Downloads im Kurs-Namespace).

- **docker-compose.yml** (n8n + n8n-runners)
- **.env.example** (inkl. Pruning + Secure Cookie + Runner-Token)
- **00-runbook.md** (Start/Stop/Update/Backup)
- **troubleshooting.md** (EACCES, Key-Mismatch, Cookie-Warnung)

Kursplan (Kapitel / Lektionen)

Kapitel	Inhalt	Link
Kapitel 1	Einstieg & Zielbild	Öffnen
Kapitel 2	Docker & Compose installieren	Öffnen
Kapitel 3	n8n per Docker Compose (LAN-only)	Öffnen
Kapitel 4	Troubleshooting Praxis (3 Klassiker)	Öffnen
Kapitel 5	Python im Code-Node (External Task Runner)	Öffnen
Kapitel 6	Betrieb, Updates, Next Steps	Öffnen
Kapitel 7	Betrieb, Updates, Next Steps	Öffnen

Direktlinks: Lektionen

Kapitel 1

- Lektion 1.1 – Was wir heute bauen
- Lektion 1.2 – Mini-Architektur & Ports

Kapitel 2

- Lektion 2.1 – System vorbereiten
- Lektion 2.2 – Docker Engine + Compose Plugin installieren
- Lektion 2.3 – Funktionstest

Kapitel 3

- Lektion 3.1 – Projektstruktur anlegen
- Lektion 3.2 – .env erstellen
- Lektion 3.3 – docker-compose.yml erstellen & starten

Kapitel 4

- Lektion 4.1 – Container-Status, Logs, Ports lesen
- Lektion 4.2 – Fehlerbild: EACCES / Permission denied
- Lektion 4.3 – Fehlerbild: Mismatching encryption keys

Kapitel 5 (optional)

- Lektion 5.1 – Runner-Token erzeugen & .env erweitern
- Lektion 5.2 – Compose um Runner erweitern
- Lektion 5.3 – Python-Test im Code-Node

Kapitel 6

- Lektion 6.1 – Runbook: Start/Stop/Update
- Lektion 6.2 – Nächste Ausbaustufe

Next Steps (nach dem Free-Kurs)

- **Pro-Modul:** Reverse Proxy + HTTPS (Apache/Traefik), saubere URLs ohne Port, sichere Sessions
- **Backup-Modul:** Pull-Backup (core.local zieht von mcp.local), versioniert & wartungsfest
- **CoreStack:** n8n + Postgres + DokuWiki + Gitea + code-server + FileBrowser

Merksatz: Der ``N8N_ENCRYPTION_KEY`` ist produktiv kritisch. Einmal setzen, sicher speichern (Passwortmanager).