

LB-Projekt Modul 210 – Docker & GlobeNotes

- ▶ Natalia & Anina
- ▶ Abgabedatum: 11.12.2025
- ▶ Lehrer: Patrick Venzin

- ▶ (Bildplatzhalter)

Agenda

- ▶ • Projektverlauf
- ▶ • Erste Projektidee (WordPress)
- ▶ • Erste Infrastruktur & Pipeline
- ▶ • Wechsel der Projektidee
- ▶ • GlobeNotes – Finale Lösung
- ▶ • Infrastruktur & Konfiguration
- ▶ • CI/CD Pipeline
- ▶ • Testplan
- ▶ • Installationsanleitung

Projektverlauf

- ▶ Diese Präsentation zeigt den Verlauf des Projekts sowie den finalen Stand.
- ▶ Frühe Ideen, Entscheidungen und Anpassungen werden erklärt.

Erste Projektidee – WordPress

- ▶ • Ursprünglich: Rezepteblog mittels WordPress + MariaDB
- ▶ • Deployment via Docker Compose
- ▶ • CI/CD Pipeline zur Validierung & Deployment
- ▶ • Idee später verworfen zugunsten besserer Lernzielerfüllung

Erstes Git Repository

- ▶ • Ordnerstruktur definiert
- ▶ • Erste Commits ins GitLab
- ▶ • Aufbau von infra/, app/ und pipelines/

System bauen & automatisieren

- ▶ • docker-compose.yml angelegt
- ▶ • Tests via Docker durchgeführt
- ▶ • WordPress lief erfolgreich unter
<http://localhost:8080>
- ▶ (Bildplatzhalter)

Erste Infrastruktur (WordPress)

- ▶ • WordPress + MariaDB in Containern
- ▶ • Kommunikation über internes Netzwerk
- ▶ • Persistenz via Volumes

- ▶ (Bildplatzhalter – Diagramm)

Wechsel der Projektidee

- ▶ Warum Wechsel?
- ▶ • GlobeNotes war bereits begonnen
- ▶ • Lernziele (CI/CD, Container, Deployment) konnten sinnvoller umgesetzt werden
- ▶ • Praxisnäher durch bestehendes Frontend & Backend

GlobeNotes – Finale Projektidee

- ▶ GlobeNotes ist eine Webapp zum Verwalten von Reisezielen.
- ▶ Benutzer kann:
 - ▶ • Ziele erfassen, anzeigen, löschen
 - ▶ • Kategorien filtern
 - ▶ • Bilder speichern
 - ▶ • Daten bleiben dank Persistenz erhalten

Endgültige Infrastruktur

- ▶ • Frontend in Docker + NGINX (Port 80)
- ▶ • Backend in Docker (Spring Boot, Port 8080)
- ▶ • Persistente H2-Datenbank via Volume
- ▶ • CI/CD Pipeline mit Build, Validate & optional Deploy
- ▶ (Bildplatzhalter – Diagramm)

Docker Compose – Konfiguration

- ▶ Services:
 - ▶ • globenotes-backend
 - ▶ • globenotes-frontend
 - ▶ • Volumes: h2-data, uploads
- ▶ Backend Port: 8080
- ▶ Frontend Port: 80

CI/CD Pipeline

- ▶ • Automatisch bei Push auf main
- ▶ • Stage 1: Build Frontend & Backend
- ▶ • Stage 2: Validate docker-compose.yml
- ▶ • Optional: Deploy Job via docker compose up -d

Testplan – Übersicht

- ▶ Getestet wurde:
 - ▶ • Deployment
 - ▶ • Frontend erreichbar
 - ▶ • Backend erreichbar
 - ▶ • API-Funktionalität
 - ▶ • CRUD-Operationen
 - ▶ • Persistenz
 - ▶ • H2-Konsole
 - ▶ • Pipeline-Build

Installationsanleitung

- ▶ 1. Docker Desktop installieren
- ▶ 2. Repo klonen
- ▶ 3. Deployment:
 - ▶ cd infra
 - ▶ docker compose up --build
- ▶ Erreichbarkeit:
 - ▶ • <http://localhost> (Frontend)
 - ▶ • <http://localhost:8080> (Backend)
 - ▶ • [/h2-console](#)

Danke!

- ▶ Fragen?