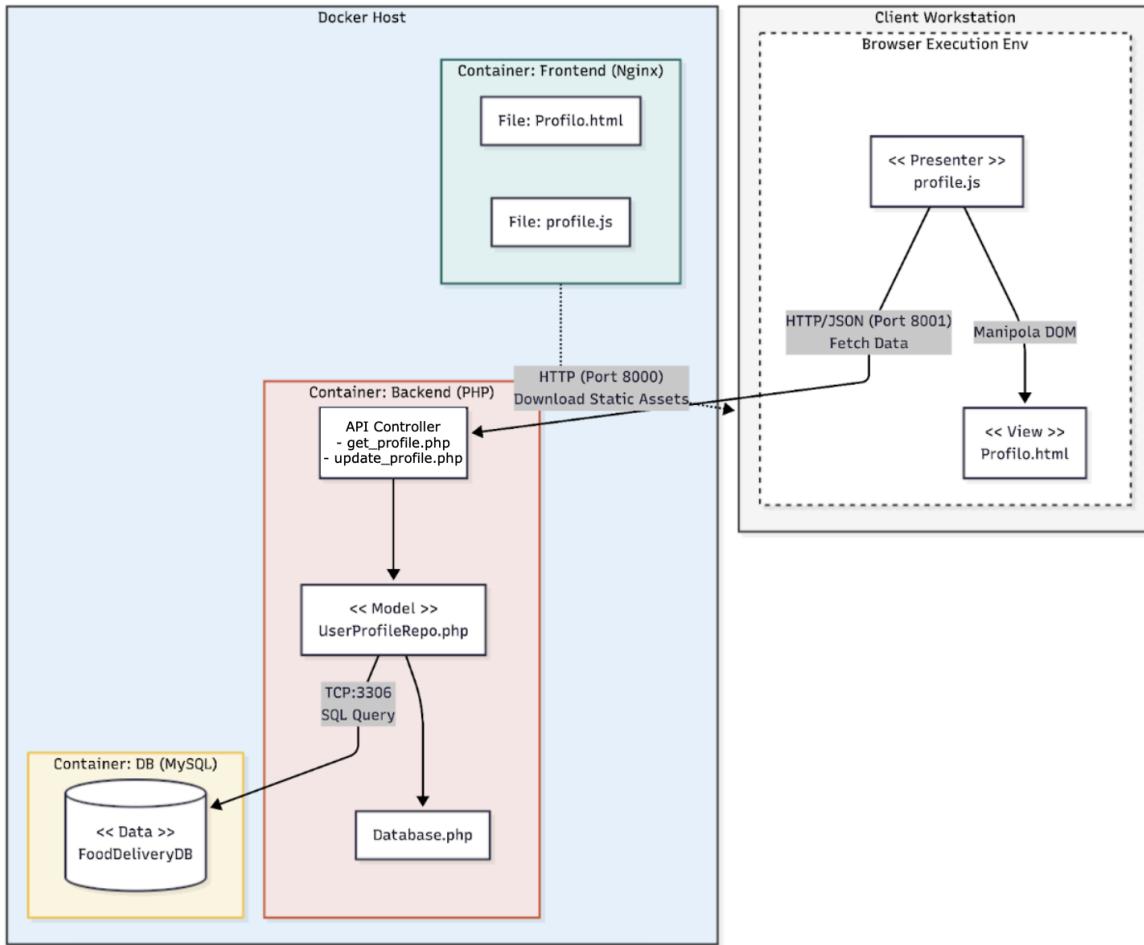


SVOLTO DA COLUCCI PASQUALE, MATR: 358141

Diagramma di Deployment: Modulo Gestione Profilo



Descrizione del Diagramma di Deployment: Modulo Gestione Profilo

Il diagramma di distribuzione illustra la topologia fisica del sistema, evidenziando la netta separazione tra l'ambiente di esecuzione del client e l'infrastruttura server basata su container Docker.

1. Nodi di Esecuzione e Container Il sistema è distribuito su tre servizi principali interconnessi tramite una rete virtuale `food_net`:

- **Frontend (Nginx - Porta 8000):** Serve gli asset statici (`Profilo.html`, `profile.js`) al browser.
- **Backend (PHP/Apache - Porta 8001):** Ospita i controller API (`get_profile.php`, `update_profile.php`) e il Model (`UserProfileRepo.php`).
- **DB (MySQL - Porta 3306):** Mantiene la persistenza dei dati relazionali.

2. Componenti Software

- **Presenter (profile.js)**: Eseguito nel browser dell'utente (Client Workstation), gestisce la logica di presentazione e la sicurezza del client (Token Storage).
- **Model (UserProfileRepo.php)**: Risiede nel container backend. Gestisce l'accesso ai dati in modo isolato (No Singleton globale).

3. Flussi di Comunicazione: La separazione dei container impone una comunicazione strutturata:

- **Download**: Il browser scarica l'interfaccia dal Frontend (Porta 8000).
- **API Runtime (Stateless)**: Il Presenter comunica con il Backend (Porta 8001) tramite chiamate AJAX/Fetch.
 - *Nota di Sicurezza*: Ogni richiesta HTTP include un **Header Authorization: Bearer <Token>**. Questo rende la comunicazione perfettamente **Stateless**, poiché il container Backend non mantiene sessioni in memoria ma verifica crittograficamente il token a ogni chiamata.
- **Accesso ai Dati**: Il backend interroga il database MySQL (nome host: db) solo dopo aver validato il token e i dati in ingresso.

4. Giustificazione Architetturale Questa configurazione garantisce:

- **Isolamento**: Frontend e Backend scalano separatamente.
- **Sicurezza**: L'assenza di sessioni condivise tra i container obbliga l'uso di token, rendendo l'architettura robusta e pronta per il cloud.