= Felhők hálózati szolgáltatásai laboratórium - BMEVITMMB11-HU

Házi feladat

Remény Olivér - IEY82F - 2024.04.12.

== Áttekintés

Ez a feladat

- Egy webes felület létrehozásából áll, amely lehetővé tesz képek feltöltését leírásokkal együtt,
- Egy olyan rendszer kialakításából ami tárolja a képeket leírásaikkal együtt
- Egy olyan program integrálása a feltöltés folyamatába ami felismeri az autó alakú dolgokat a feltöltött képen, bekeretezi azokat és visszatér a felismert autó alakú objektumok számával
- Egy üzenőrendszerből, ami az összes feliratkozót értesíti az összes feltöltési eseményről, a feldolgozás eredményével együtt.
- Egy CI/CD folyamat összerakásáról szól, ami az előbb felsorolt szolgáltatásokat célszerűen lefordítja, és elindítja egy szerveren.

Ez a dokumentum meghatározza a webszolgáltatás fejlesztése során használt architektúrális komponenseket, technológiákat és folyamatokat.

=== 1. Felhasználói felület (UI)

- **Technológia**: HTML+CSS+JS

  > Megjegyzés: Várhatóan nem lesz szükség reaktivitásra, de ha fejlesztés során kiderül hogy mégis, akkor a Vue3 JS keretrendszer lesz alkalmazva.

- **Funkcionalitás**:
  - Feltöltőoldal biztosítása a felhasználók számára a képek és a hozzájuk tartozó leírásokkal.
  - Egy adott feltöltés megtekintése, ahol látható az eredeti és a feldolgozott kép (ahol az észlelt autók be vannkak keretezve), emellett a keretek száma és a felhasználói leírás is elolvasható.
  - Lehetővé teszi a felhasználók számára az értesítésekre való feliratkozást egy egyszerű űrlapon keresztül. (Várhatóan nem kell megadni semmit, csak egy gomb lesz.)

=== 2. Képek és leírás tárolása (DB)

- **Technológia**: Fájlrendszer

  > Megjegyzés: Több folyamat is hozzá fog férni egy fájlhoz, azonban egy fájlt csak egy folyamat fog írni valaha.

- **Funkcionalitás**: Egyszerű fájl írás és olvasás funkcionalitás a követelmény.
  - Eredeti képek tárolása egy mappában
  - Módosított képek tárolása (egy másik mappában azonos névvel mint az eredeti kép.)
  - Leírás tárolása (egy másik mappában azonos névvel mint az eredeti kép.)
  - (Ha ezt a felsimerő modell igényli, feltöltés során a kép egységes méretre és formátumra alakítása.)

=== 3. Objektumfelismerés képeken (ML)

- **Technológia**: Yolo objektumfelismerő modell
- **Funkcionalitás**:
    - Képek nevének fogadása http apin keresztül
    - Képek beolvasása fájlrendszerből
    - Képeken autó alakú objektumok bekeretezése, opcionálisan címkézése
    - "Keretezett" képek mentése fájlrendszerbe.

> Megjegyzés: Egy potenciális megoldást mutat be ez a blogpost: https://dev.to/andreygermanov/how-to-create-yolov8-based-object-detection-web-service-using-python-julia-nodejs-javascript-go-and-rust-4o8e

=== 4. Üzenőrendszer

- **Technológia**: Web push
- **Funkcionalitás**:
    - Feliratkozók nyilvántartása (fájlrendszerben tárolt lista segítségével)
    - Belső esemény (feltöltés és felismerés) hatására az összes feliratkozó értesítése új feltöltésről

> Megjegyzés: Egy potenciális megoldást mutat be ez a blogposzt: https://web.dev/articles/sending-messages-with-web-push-libraries

=== 5. CI/CD rendszer

- **Technológia**: Github Actions + Docker (Build) + Docker compose remote deploy
- **Funckionalitás**:
    - A feltöltött forráskódból docker imagek előállítása
    - Imagek feltöltése a GitHub(vagy DockerHub) Container Registry-be
        - Ennek a megoldásáról szól ez a bejegyzés: https://docs.github.com/en/packages/managing-github-packages-using-github-actions-workflows/publishing-and-installing-a-package-with-github-actions#upgrading-a-workflow-that-accesses-a-registry-using-a-personal-access-token
    - Egy távoli szerveren futó docker compose példány utasítása, hogy töltse le az összes új image-t, és frissítse a futó konténereit az új verziókkal.
        - Ennek a megoldásáról szól ez a bejegyzés: https://www.docker.com/blog/how-to-deploy-on-remote-docker-hosts-with-docker-compose/
    - Ez pedig Github Actions specifikus: https://github.com/marketplace/actions/docker-compose-remote-deploy

---

# Megvalósítás

=== CI/CD rendszer

Források:

- How to deploy on remote Docker hosts with docker-compose

- Docker Compose Remote Action
- Publishing and installing a package with GitHub Actions

```
sequenceDiagram
    actor u as User
    participant a as GitHub Actions
    participant s as Repository secrets
    participant b as Build Action (Docker)
    participant g as Github Container Registry (ghcr)
    participant d as Docker Compose Remote Action
    participant r as Remote host (Docker Engine + Compose)
    u-->>+a: push happened
    a->>+b: build the images
    b->>-a: OK
    a->>+s: Get ghcr password
    s->>-a: <<ghcr password>>
    a->>+g: login(ghcr password)
    g->>-a: OK
    a->>+b: push images
    b->>+g: store(images)
    g->>-b: OK
    b->>-a: OK
    a->>+s: Get remote host credentials
    s->>-a: <<remote host credentials>>
    a->>+d: configure_remote_host(compose.yaml, remote host credentials)
    d->>+r: login(remote host credentials)
    r->>d: OK
    d->>r: docker compose up -d -f compose.yaml
    par
        r->>-d: OK
        d->>-a: OK
    and
        r->>+g: get_images(image1,image2...)
        g->>-r: <<images>>
        note over r: rollout new versions of changed containers
    end
    note over a: Action finished
    deactivate a
```