



Proiect Laborator

Cloud Computing



Componenta Echipei

Tofanel Mihaela-Raisa, BDTS 505

Atasie Oana-Andreea, BDTS 505

Catanoiu Gabriela Anamaria, BDTS 505



Introducere

Aplicația construită în cadrul proiectului de laborator reprezintă un serviciu pentru adopția animalelor. Aplicația a fost implementată folosind Java pentru partea de backend și Thymeleaf pentru partea de frontend, cuprinzând patru microservicii.

Scopul aplicației este reprezentat de facilitarea procesului de adopție a animalelor de companie, punând la dispoziția utilizatorului următoarele functionalitati: vizualizarea unei liste a tuturor animalelor de companie existente în cadrul serviciului, vizionarea de detalii legate de un animal de companie, adăugarea unui animal de companie, adăugarea de medicamente, alergii și alte condiții speciale, vaccinuri și vizite la medicul veterinar pentru un animal propriu în vederea menținerii unui istoric complet asupra stării de sanatate a acestuia și adopția unui animal.

Aplicația a fost integrată în Docker, folosind docker-compose pentru configurarea containerelor și Docker Swarm pentru rularea acestora. În plus, acesteia i-a fost integrat Zipkin, pentru monitorizarea apelurilor microserviciilor.

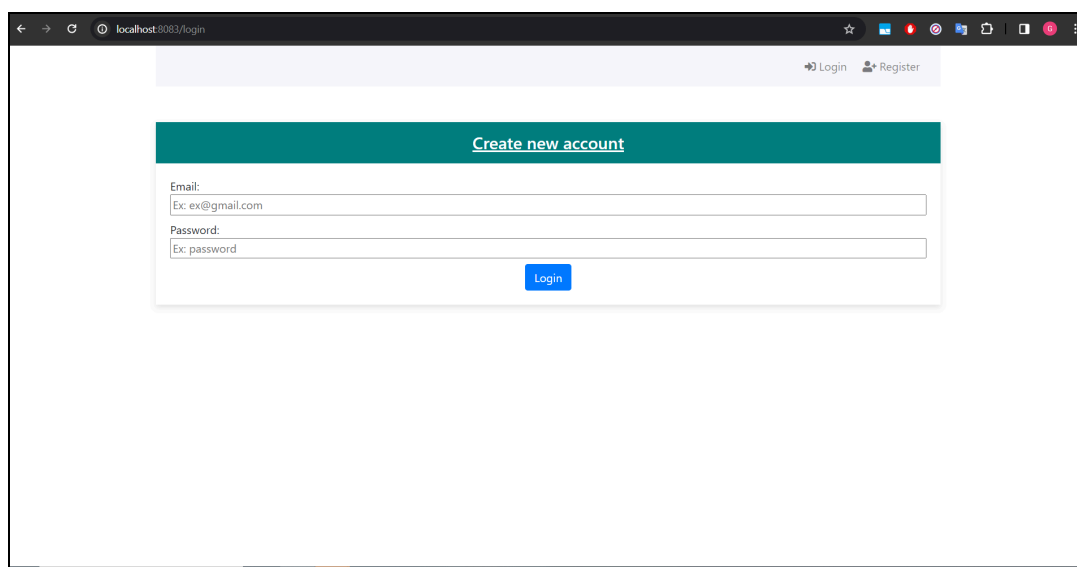
În continuare vor fi prezentate microserviciile aplicației și funcționalitățile oferite de către aceasta, modul de utilizare al platformei Docker pentru rularea aplicației și serviciile adiționale adăugate în cadrul acesteia.

Microservicii

Aplicația este organizată în patru microservicii: *users-service*, ce pune la dispoziție funcționalități privind utilizatorii, *pet-service*, ce pune la dispoziție funcționalități generale privind animalele de companie, *health-service*, ce încapsulează acele funcționalități privind starea de sănătate a animalelor de companie, și *adoption-service*, în care sunt agregate funcționalitățile privind procesul de adopție.

Users service

Acest microserviciu încapsulează toate operațiile care pot fi realizate de un utilizator, punând la dispoziția acestora metodele: login, register și logout. Login-ul este prima operație realizată în cadrul serviciului, acesta fiind necesar pentru accesarea tuturor celorlalte funcționalități. Aceasta operație este realizată pe baza unui email și a unei parole. Funcționalitatea este ilustrată în figura Fig. 1.1. În cazul în care vizitatorul aplicației nu deține un cont, acesta poate să creeze un nou cont prin funcționalitatea de register. Funcționalitatea este ilustrată în figura Fig. 1.2.



The screenshot displays a web browser window with the address bar showing 'localhost:3083/login'. The page features a light blue header with 'Login' and 'Register' links. Below the header is a dark blue bar with the text 'Create new account'. The main form has two input fields: 'Email' with a placeholder 'Ex: ex@gmail.com' and 'Password' with a placeholder 'Ex: password'. A blue 'Login' button is at the bottom right of the form.

Fig.1.1: Login

The screenshot shows a web browser window with the address bar displaying 'localhost:8083/register'. The page has a light blue header with 'Login' and 'Register' links. The main content area features a white box with a teal header that says 'Create new account'. Inside this box is a registration form with the following fields: Firstname (with example 'Ana'), Lastname (with example 'Mirciuc'), Address (with example 'Street Copaceni 20'), Date of birth (with example '2000/12/23'), Email address (with example 'ex@gmail.com'), Password (with example 'password'), and Phone number (with example '0746112345'). Each field has a small text input area with a placeholder and an example. At the bottom right of the form is a blue 'Register' button.

Fig. 1.2: Register

Logarea utilizatorilor este realizată prin verificarea existenței unui cont de utilizator în interiorul bazei de date ale cărui parola și email corespund celor primite de la utilizator. În cazul în care un astfel de utilizator există acesta este redirecționat către pagina ce listează toate animalele disponibile în cadrul serviciului.

De asemenea, după realizarea cu succes a operației de login, id-ul corespunzător utilizatorului este stocat în Redis, pentru a putea fi folosit ulterior și în alte microservicii.

În ceea ce privește configurația microserviciului, acesta folosește portul 8083 și baza de date proprie, numită *cloudproject*.

Pet service

Acest microserviciu conține toate funcționalitățile privind datele generale ale animalelor de companie existente în cadrul aplicației. Funcționalitățile permise sunt: vizualizarea tuturor animalelor de companie, vizualizarea detaliilor privind un animal de companie, adăugarea unui animal de companie, vizualizarea unei liste de specii și a unei liste de rase precum și adăugarea unei specii și a unei rase.

În figurile *Fig.2.1*, *Fig.2.2* și *Fig.2.3* sunt ilustrate metodele de vizualizare a speciei, rasei și a tuturor animalelor de companie existente în acest serviciu. În figura *Fig. 2.4* este inclusă pagina de vizualizare a unui animal de companie.

The screenshot shows a web browser at localhost:8081/species/getAISpecies. The navigation bar includes links for Pets, Species, Breeds, Adoptions as Client, Adoptions as Owner, and Logout. An 'Add species' button is in the top right. The table below lists four species: Dog, Cat, Hamster, and Parrot, each with a description and edit/delete actions.

Id	Species name	Species description	Edit	Delete
2	Dog	a domesticated carnivorous mammal that typically has a long snout, an acute sense of smell, non-retractable claws, and a barking, howling, or whining voice		
3	Cat	a small domesticated carnivorous mammal with soft fur, a short snout, and retractable claws. It is widely kept as a pet or for catching mice, and many breeds have been developed		
4	Hamster	a solitary burrowing rodent with a short tail and large cheek pouches for carrying food, native to Europe and North Asia.		
5	Parrot	a bird, often vividly coloured, with a short downcurved hooked bill, grasping feet, and a raucous voice, found especially in the tropics and feeding on fruits and seeds. Many are popular as cage birds, and some are able to mimic the human voice.		

Fig. 2.1: Vizualizarea speciilor

The screenshot shows a web browser at localhost:8081/breed/getAllBreeds. The navigation bar is the same as in Fig. 2.1. An 'Add breeds' button is in the top right. The table below lists six breeds: Siameze, BritishShorthair, SpanielCocker, Poodle, and Roborovski, each with a description, lifespan, and edit/delete actions.

Id	Breed name	Breed description	Lifespan	Edit	Delete
2	Siameze	The Siamese is built on long athletic lines, standing tall on slender legs with neat oval paws. Ears are large, pricked and give an intelligent expression, following the lines of the triangular shaped head. Eyes are almond shape and slant inwards towards the nose, and the tail is long, tapering, and straight.	10		
3	BritishShorthair	The British Shorthair is a medium to large sized, compact, chunky cat, known for their attractive features and thick, plush fur. Short coupled and robust, the British Shorthair is instantly recognisable by their rounded face and large round eyes, giving this cat an adorable expression and bags of character.	12		
4	SpanielCocker	The Cocker is the AKC's smallest sporting spaniel, standing about 14 to 15 inches. The coat comes in enough colors and patterns to please any taste. The well-balanced body is sturdy and solid, and these quick, durable Gundogs move with a smooth, easy gait.	13		
5	Poodle	The poodle is an extremely smart, energetic, and friendly dog known for their signature curly coat and three size varieties: toy, miniature, and standard. Under the poodle's frilly, low-shedding coat is a powerful athlete and an overall wonderful companion	14		
6	Roborovski	The smallest of the popular hamster breeds, the Roborovski dwarf hamster (Phodopus roborovski) matures to just 2 inches long.	10		

Fig. 2.2: Vizualizarea raselor

The screenshot shows a web browser at localhost:8081/pet/getAllPets. The navigation bar is the same as in Fig. 2.1. An 'Add pet' button is in the top right. The table below lists five pets: Adele, Rex, Roger, and Princess, each with details like species, breed, year of birth, gender, color, price, availability, owner, and edit/delete actions.

Id	Pet name	Species	Breed	Year of birth	Gender	Color	Price	Available	Owner	Info	Edit	Delete
2	Adele	Cat	Siameze	2011	F	Beige	1000.0	true	Raisa Tofanel			
3	Rex	Dog	Poodle	2011	F	Gray	250.0	true	Raisa Tofanel			
4	Roger	Hamster	Roborovski	2020	M	White	100.0	true	Raisa Tofanel			
5	Princess	Dog	SpanielCocker	2020	F	Brown	500.0	true	Raisa Tofanel			

Fig. 2.3: Vizualizarea animalelor de companie

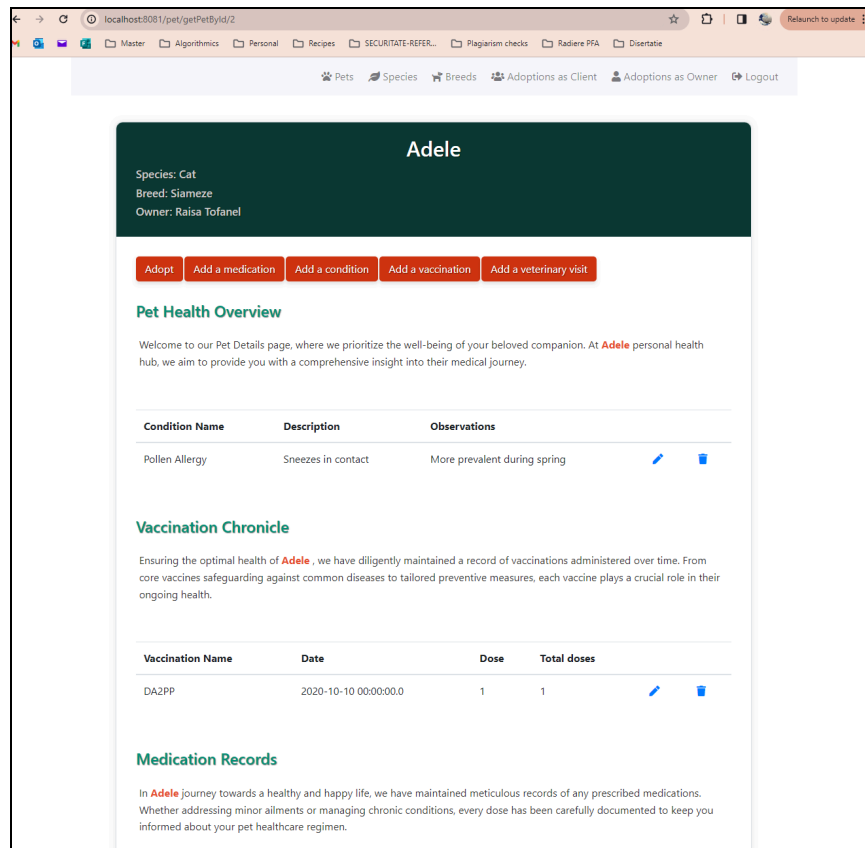


Fig. 2.4: Vizualizarea detaliilor unui animal de companie

În ceea ce privește configurația microserviciului, acesta folosește portul 8081 și baza de date proprie, numită *petproject*.

Health service

Acest microserviciu conține toate funcționalitățile privind istoricul stării de sănătate a animalelor de companie existente în cadrul aplicației. Funcționalitățile permise sunt: vizualizarea tuturor medicamentelor, a condițiilor speciale, a vaccinurilor și vizitelor la medicul veterinar ale unui animal de companie, adăugarea, editarea și ștergerea unor medicamente, condiții speciale, vaccine și vizite la medicul veterinar pentru un animal de companie propriu.

Medicamentele, condițiile speciale, vaccinurile și vizitele la medicul veterinar sunt disponibile pentru vizualizare în cadrul paginii ce expune detaliile generale despre un animal de companie. În figurile *Fig.3.1* și *3.2* sunt ilustrate vizualizarea medicamentelor și a condițiilor

speciale corespunzătoare unui animal de companie. Funcționalitatea de adăugare, editare și ștergere a elementelor istoricului sanatații sunt accesate din aceeași pagina, fiind disponibile doar proprietarilor animalelor de companie.

Figura Fig.3.2 ilustrează meniul prin prisma căruia sunt accesate operațiunile posibile, iar figura Fig.3.3 ilustrează formularul puse la dispoziția utilizatorului pentru adăugarea/editarea medicamentelor în cadrul istoricului stării de sanatate.



Medication Records				
In Adele journey towards a healthy and happy life, we have maintained meticulous records of any prescribed medications. Whether addressing minor ailments or managing chronic conditions, every dose has been carefully documented to keep you informed about your pet healthcare regimen.				
Medication Name	Reason	Dosage (mg)	Frequency (days)	Observations
Apoquel	Allergies	20	2	 

Fig. 3.1: Vizualizarea medicamentelor

Pet Health Overview		
Welcome to our Pet Details page, where we prioritize the well-being of your beloved companion. At Adele personal health hub, we aim to provide you with a comprehensive insight into their medical journey.		
Condition Name	Description	Observations
Pollen Allergy	Sneezes in contact	More prevalent during spring

Fig. 3.2: Vizualizarea condițiilor speciale

Add a medication	Add a condition	Add a vaccination	Add a veterinary visit
------------------	-----------------	-------------------	------------------------

Fig. 3.2: Meniul istoricului stării de sanatate

Fig. 3.3: Adăugarea/Editarea medicamentelor

În ceea ce privește configurația microserviciului, acesta folosește portul 8080 și baza de date proprie, numită *healthproject*.

Adoption service

Acest microserviciu conține toate facilitățile privind adopția animalelor de companie existente în cadrul aplicației. Funcționalitățile permise sunt: adopția unui animal de companie, introducerea detaliilor de plată, vizualizarea animalelor de companie adoptate sau date spre adopție, vizualizarea de comentarii privind adopțiile și adăugarea și editarea de comentarii privind adopția unui animal.

În figura *Fig. 4.1* este ilustrată pagina de adopție a unui animal de companie. În figura *Fig. 4.2* este ilustrată introducerea detaliilor de plată. În figura *Fig. 4.3* este prezentată pagina ce ilustrează o vedere de ansamblu asupra unui proces de adopție, aceasta conținând și un comentariu lăsat de client.

localhost:8082/adoption/addAdoptionForm/2

Pets Species Breeds Adoptions as Client Adoptions as Owner Logout

Adopt Pet Adele

*The animal you want to adopt costs 1000.0 EURO. After completing this form, you will be redirected to the payment page. After completing it, the owner will contact you.

Pet owner: Tofanel Raisa

Write here any comment related to the adoption

Comment...

Adopt pet

Fig. 4.1: Adopția unui animal

localhost:8082/payment/addPaymentForm/2

Pets Species Breeds Adoptions as Client Adoptions as Owner Logout

Payment for adoption

Billing Address:

Email: test@user.com

Address: Vasile Lupu

Payment:

Card Number: 1111-2222-3333-4444

Cardholder Name: Firstname Lastname

Expiration Date: 03/24

CVV: 635

Amount to pay for adoption: 1000.0 EURO

Submit Payment

Fig. 4.2: Introducerea detaliilor de plată

localhost:8082/adoption/2

Pets Species Breeds Adoptions as Client Adoptions as Owner Logout

Adoption Details

Pet Name: Adele

Adoption Date: 2024-01-14

Owner: Raisa Tofanel

Client: Andreea Tofanel

Amount: 1000.0

Feedback for this adoption:

Comment: It went great **Date:** 2024-01-14

Rating: 1

Content: **Rating:** 1

Fig. 4.3: Detalii privind procesul de adopție

În ceea ce privește configurația microserviciului, acesta folosește portul 8082 și baza de date proprie, numită *adoptionproject*.

Comunicarea între microservicii

Comunicarea dintre microserviciile aplicației este ilustrată în tabelul *Table 5.1*.

Microserviciu	Funcționalitate	Microservicii cu care comunică
Pet-service	Vizualizarea unui animal de companie	Health-service (preluarea medicamentelor, vaccinilor, vizitelor la medicul veterinar și a condițiilor speciale)
	Vizualizarea animalelor de companie	Users-service (preluarea datelor despre utilizatorul logat)
	Adăugarea unui animal de companie	Users-service (preluarea datelor despre utilizatorul logat)
Health-service	Adăugarea unui medicament	Pet-service (preluarea deținătorului animalului de companie)
	Adăugarea unui vaccin	Pet-service (preluarea deținătorului animalului de companie)
	Adăugarea unei condiții speciale	Pet-service (preluarea deținătorului animalului de companie)
	Adăugarea unei vizite la veterinar	Pet-service (preluarea deținătorului animalului de companie)
Adoption-service	Adopția unui animal	Pet-service (preluarea detaliilor despre animalul de companie adoptat și actualizarea disponibilității acestuia) Users-service (preluarea datelor despre utilizatorul logat)

Table 5.1: Comunicarea între microservicii



Docker, docker-compose, docker-swarm

În vederea utilizării Docker, au fost create fișiere Dockerfile pentru fiecare dintre microserviciile create, acestea fiind folosite pentru crearea imaginilor pentru microservicii. În următorul fragment de cod este inclus fișierul Dockerfile pentru microserviciul Adoption:

```
# CREATE A JAR FOR THE PROJECT

FROM maven:3.8.4-openjdk-17 AS builder

WORKDIR /app

COPY pom.xml .
COPY src ./src

RUN mvn clean package

# EXECUTE THE PREVIOUSLY CREATED JAR

FROM openjdk:17-slim

WORKDIR /usr/app

COPY --from=builder /app/target/adoption-0.0.1-SNAPSHOT.jar
adoption-service.jar
COPY --from=builder /app/src/main/resources ./resources

EXPOSE 8082

CMD ["java", "-jar", "adoption-service.jar"]
```

Un astfel de fișier a fost creat pentru toate cele patru microservicii. Pentru crearea imaginilor pentru microservicii sunt folosite următoarele comenzi de cod:

```
docker build -t pet-service:latest -f Dockerfile .
docker build -t user-service:latest -f Dockerfile .
```

```
docker build -t health-service:latest -f Dockerfile .
docker build -t adoption-service:latest -f Dockerfile .
```

De asemenea, pentru crearea containerelor corespunzătoare fiecărui microserviciu, au fost adăugate următoarele fragmente de cod în cadrul unui fișier docker-compose. Imaginile utilizate vor fi unele publice menționate ulterior.

```
users:
  image: atasieoana2001/cloud_project:user-service-latest
  container_name: users
  restart: always
  ports:
    - "8083:8083"
  networks:
    - cloudnet
  environment:
    - spring.datasource.url=jdbc:mysql://mysqldb_user:3306/cloudproject?allowPublicKeyRetrieval=true
    - spring.data.redis.host=redis
    - docker.application.ip=192.168.199.84
  depends_on:
    - mysqldb_user
    - redis

pet:
  image: atasieoana2001/cloud_project:pet-service-latest
  container_name: pet
  restart: always
  ports:
    - "8081:8081"
  networks:
    - cloudnet
  environment:
```

```
spring.datasource.url=jdbc:mysql://mysqldb_pet:3306/petproject?allowPublicKeyRetrieval=true
```

- spring.data.redis.host=redis
- docker.application.ip=192.168.199.84

```
depends_on:
```

- mysqldb_pet
- redis
- users
- health

```
health:
```

```
image: atasieoana2001/cloud_project:health-service-latest
```

```
container_name: health
```

```
restart: always
```

```
ports:
```

- "8080:8080"

```
networks:
```

- cloudnet

```
environment:
```

```
spring.datasource.url=jdbc:mysql://mysqldb_health:3306/healthproject?allowPublicKeyRetrieval=true
```

- spring.data.redis.host=redis
- docker.application.ip=192.168.199.84

```
depends_on:
```

- mysqldb_health
- redis

```
adoption:
```

```
image: atasieoana2001/cloud_project:adoption-service-latest
```

```
container_name: adoption
```

```
restart: always
```

```
ports:
```

- "8082:8082"

```
networks:
```

- cloudnet

```

environment:
  spring.datasource.url=jdbc:mysql://mysqladb_adoption:3306/adoptionproject?allowPublicKeyRetrieval=true
  - spring.data.redis.host=redis
  - docker.application.ip=192.168.199.84
depends_on:
  - mysqladb_adoption
  - redis

networks:
  cloudnet:

```

Acest fișier este folosit pentru facilitarea creării containerelor pe baza imaginilor pentru fiecare dintre cele patru microservicii, pentru bazele de date asociate acestora și pentru serviciile adiționale incluse în cadrul proiectului.

În vederea utilizării Docker Swarm, au fost create conturi pentru Docker Hub pentru fiecare dintre membrii echipei. De asemenea, a fost creat un repository pentru stocarea imaginilor, evidențiate în Figura 3.1. Imaginile au fost adăugate în cadrul acestuia folosind următoarele instrucțiuni:

```

docker tag mysqladb_user:latest
atasieoana2001/cloud_project:mysqladb_user-latest
docker tag mysqladb_pet:latest
atasieoana2001/cloud_project:mysqladb_pet-latest
docker tag mysqladb_adoption:latest
atasieoana2001/cloud_project:mysqladb_adoption-latest
docker tag mysqladb_health:latest
atasieoana2001/cloud_project:mysqladb_health-latest
docker tag user-service:latest
atasieoana2001/cloud_project:user-service-latest
docker tag pet-service:latest
atasieoana2001/cloud_project:pet-service-latest
docker tag health-service:latest
atasieoana2001/cloud_project:health-service-latest
docker tag adoption-service:latest
atasieoana2001/cloud_project:adoption-service-latest
docker push atasioana2001/cloud_project:health-service-latest

```

```
docker push atasieoana2001/cloud_project:pet-service-latest
docker push atasieoana2001/cloud_project:adoption-service-latest
docker push atasieoana2001/cloud_project:user-service-latest
docker push atasieoana2001/cloud_project:mysqldb_user-latest
docker push atasieoana2001/cloud_project:mysqldb_pet-latest
docker push atasieoana2001/cloud_project:mysqldb_health-latest
docker push atasieoana2001/cloud_project:mysqldb_adoption-latest
```

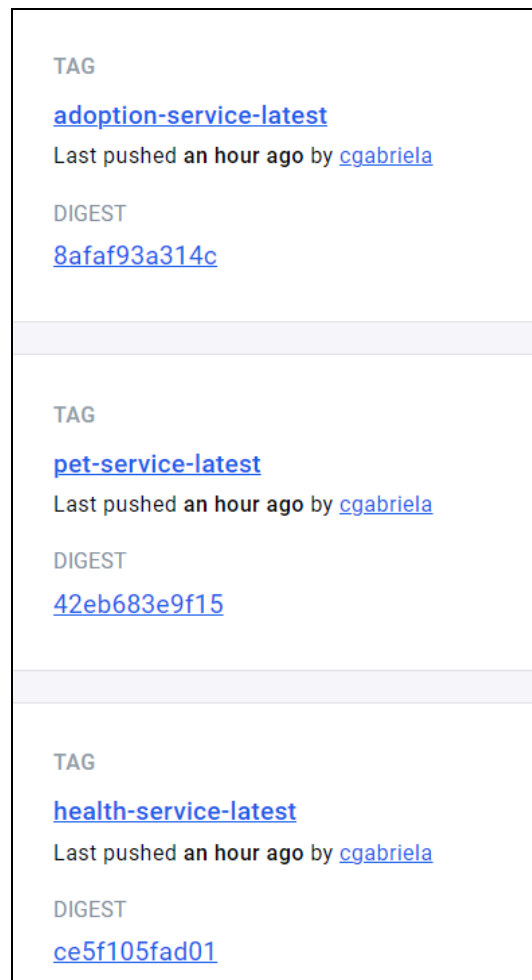


Fig.3.1 Imaginile din Docker Hub

În continuare, pentru rularea proiectului și vizualizarea stării și a logurilor containerelor au fost folosite comenzile:

```
docker-compose build
docker swarm init
docker stack deploy --compose-file docker-compose.yml swarm-project
```

```
docker stack services swarm-project
docker service logs swarm-project_pet
```

Mysql databases

Pentru fiecare dintre cele patru microservicii a fost creată o bază de date separată, pentru fiecare dintre aceste baze de date fiind create câte o imagine și un container. În fragmentul următor de cod este ilustrat conținutul fișierului Dockerfile corespunzător bazei de date a microserviciului health-service:

```
FROM mysql:latest
COPY ./init.sql /docker-entrypoint-initdb.d/
```

În cadrul fragmentului de cod de mai sus, fișierul init.sql se referă la un fișier de inițializare a bazei de date prin crearea tabelor. Pentru construirea containerelor corespunzătoare imaginilor a fost adăugat următorul fragment de cod în cadrul fișierului docker-compose:

```
mysqlpdb_user:
  image: atasieoana2001/cloud_project:mysqlpdb_user-latest
  container_name: mysqlpdb_user
  restart: always
  environment:
    MYSQL_DATABASE: cloudproject
    MYSQL_PASSWORD: admin
    MYSQL_ROOT_PASSWORD: admin
  networks:
    - cloudnet
  volumes:
    - my-datavolume_user:/var/lib/mysql

mysqlpdb_pet:
  image: atasieoana2001/cloud_project:mysqlpdb_pet-latest
  container_name: mysqlpdb_pet
  restart: always
```



```
environment:
  MYSQL_DATABASE: petproject
  MYSQL_PASSWORD: admin
  MYSQL_ROOT_PASSWORD: admin
networks:
  - cloudnet
volumes:
  - my-datavolume_pet:/var/lib/mysql

mysqlpdb_adoption:
  image: atasieoana2001/cloud_project:mysqlpdb_adoption-latest
  container_name: mysqlpdb_adoption
  restart: always
  environment:
    MYSQL_DATABASE: adoptionproject
    MYSQL_PASSWORD: admin
    MYSQL_ROOT_PASSWORD: admin
  networks:
    - cloudnet
  volumes:
    - my-datavolume_adoption:/var/lib/mysql

mysqlpdb_health:
  image: atasieoana2001/cloud_project:mysqlpdb_health-latest
  container_name: mysqlpdb_health
  restart: always
  environment:
    MYSQL_DATABASE: healthproject
    MYSQL_PASSWORD: admin
    MYSQL_ROOT_PASSWORD: admin
  networks:
    - cloudnet
  volumes:
    - my-datavolume_health:/var/lib/mysql

volumes:
  my-datavolume_user:
```

```
my-datavolume_pet:
my-datavolume_adoption:
my-datavolume_health:
```

În vederea asigurării integrității datelor au fost create metode pentru realizarea sincronizării datelor în cadrul fiecărui microserviciu pentru propria sa baza de date, în cadrul acestora fiind realizate apeluri către microserviciile în cadrul cărora sunt adăugate datele de interes. Un exemplu al unei astfel de sincronizari este reprezentat de sincronizarea în cadrul microserviciului health-service. În cadrul acestei sincronizari, se realizează un apel către microserviciul pet-service pentru preluarea tuturor animalelor de companie stocate în baza de date corespunzătoare microserviciului pet-service. Toate acestea sunt stocate apoi în baza de date corespunzătoare microserviciului health-service.

În vederea asigurării persistenței datelor între rulările aplicației au fost folosite volume. Pentru fiecare dintre cele patru baze de date a fost creat câte un volum în care datele să fie stocate între rulări.



Alte servicii utilizate

Redis

În vederea stocării utilizatorului logat este folosit serviciul Redis. Pentru utilizarea acestuia au fost adăugate următoarele linii de cod în cadrul fișierului docker-compose pentru a prelua o imagine pentru acesta:

```
redis:
  image: redis:latest
  container_name: redis
  ports:
    - "6379:6379"
  restart: always
  networks:
    - cloudnet
```

În cadrul microserviciilor a fost creat un serviciu pentru Redis, numit `RedisService`, în cadrul căruia au fost create două metode: *saveData* pentru adăugarea datelor în Redis și *getData* pentru preluarea datelor din Redis.

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.redis.core.RedisTemplate;
import org.springframework.stereotype.Service;

@Service
public class RedisService {

    @Autowired
    private RedisTemplate<String, String> redisTemplate;

    public void saveData(String key, String value) {
        redisTemplate.opsForValue().set(key, value);
    }

    public String getData(String key) {
        return redisTemplate.opsForValue().get(key);
    }

}
```

Zipkin

În vederea monitorizării apelurilor realizate de microservicii în cadrul utilizării aplicației a fost utilizat Zipkin. Pentru utilizarea acestuia au fost adăugate următoarele linii de cod în cadrul fișierului docker-compose pentru a prelua o imagine pentru acesta:

```
zipkin:
  image: ghcr.io/openzipkin/zipkin-slim:${TAG:-latest}
  container_name: zipkin
  networks:
    - cloudnet
  environment:
```

```
- STORAGE_TYPE=mem
ports:
- 9411:9411
```

În cadrul microserviciilor au fost inserate dependențele necesare utilizării Zipkin, împreună cu următoarele proprietăți:

```
spring.zipkin.baseUrl=http://zipkin:9411/
management.zipkin.tracing.endpoint = http://zipkin:9411/api/v2/spans
management.tracing.sampling.probability=1.0
spring.sleuth.sampler.probability=1.0
management.tracing.enabled=true
```

În figura Fig. 4.1 este ilustrată interfața Zipkin, cuprinzând apelurile realizate de microservicii.

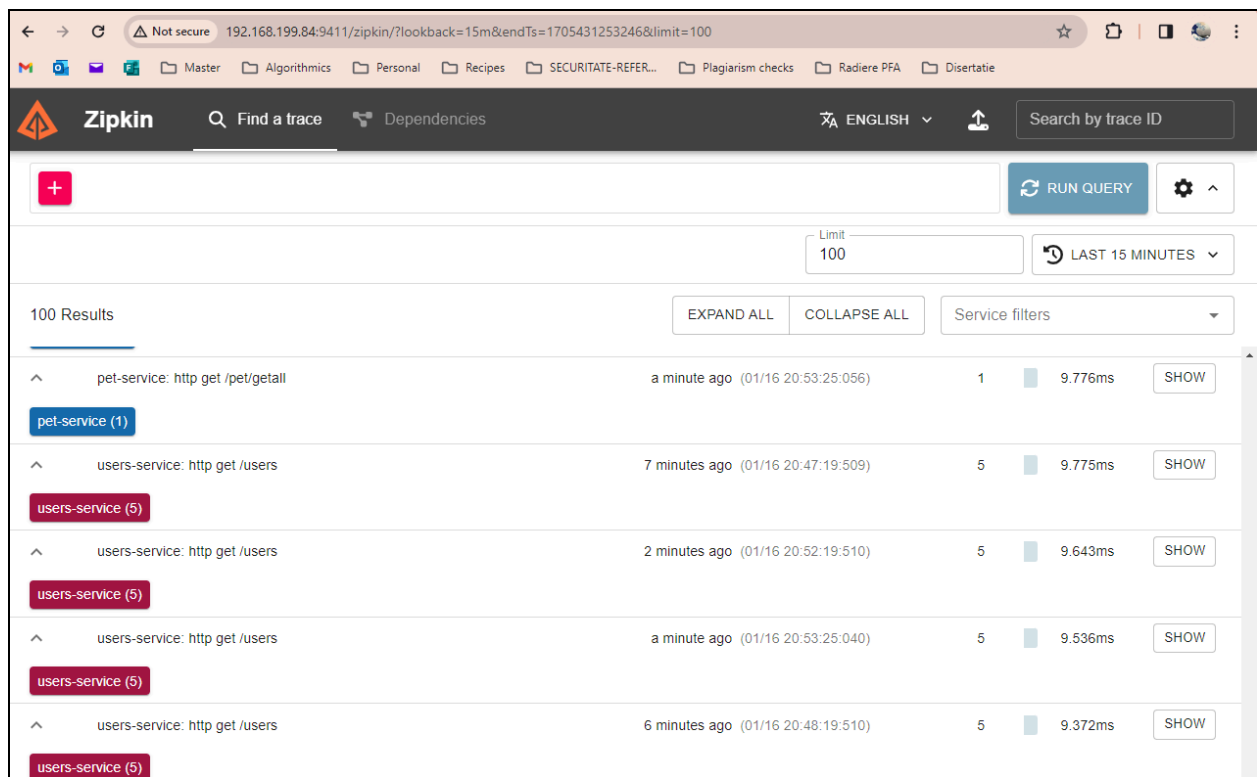


Fig.4.1 Apelurile din Zipkin