JFN Hightech University Institute

Google Cloud

(Docker-compose-PROJECT)

APPLICATION PRESENTATION

The petclinic app

Name: petclinic

Deployment strategie: **Docker-compose**

GOAL:

- Deploy The petclinic application using docker-compose

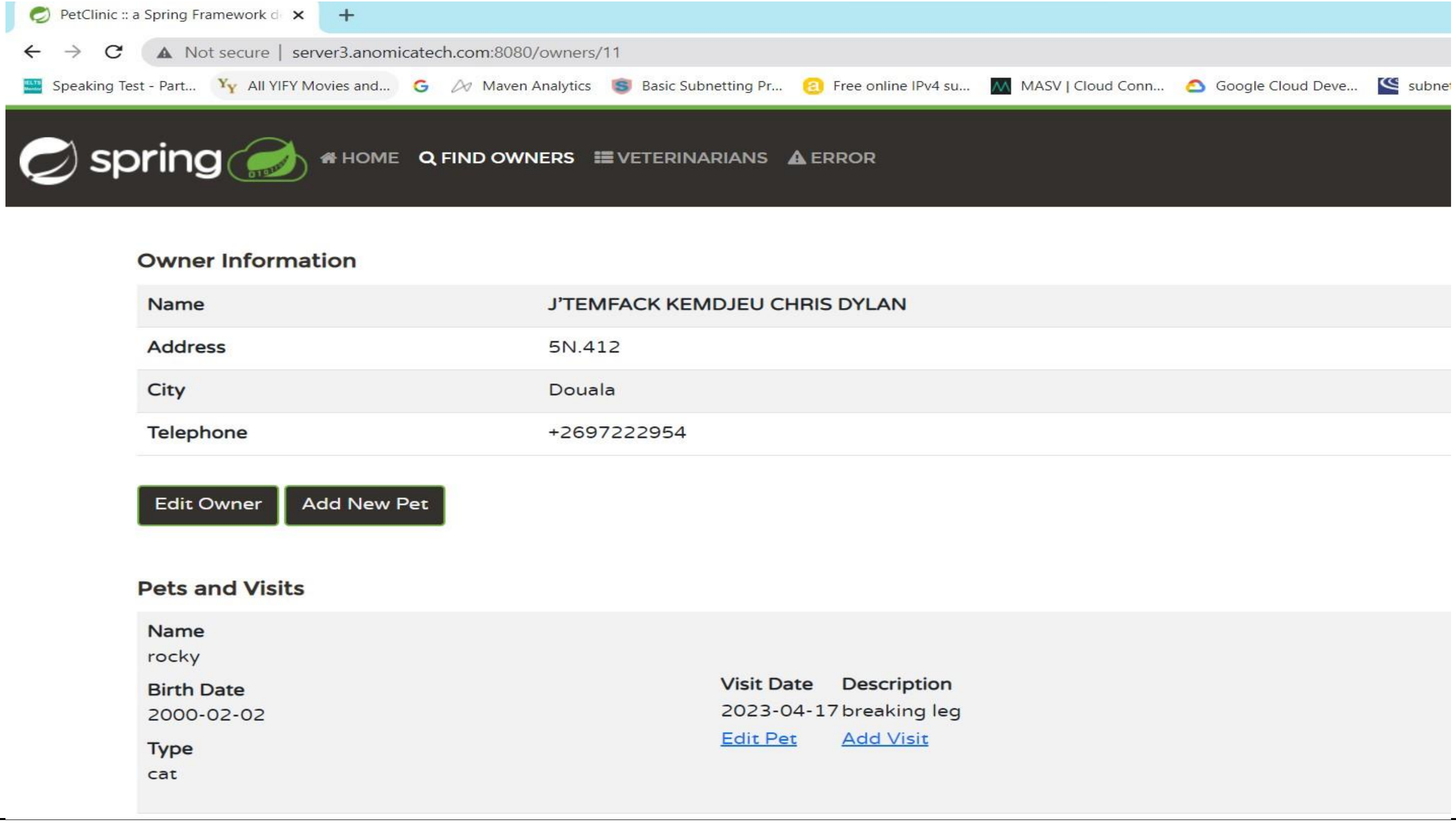
INFO:

- Customer Name: **Naruto**

INFO:

The applications allows you to perform the following set of functions:

- Add Pets
- Add Owners
- Finding Owners
- Finding Veterinarians
- Exceptional handling



Owner Information

Name	J'TEMFACK KEMDJEU CHRIS DYLAN
Address	5N.412
City	Douala
Telephone	+2697222954

Pets and Visits

Name
rocky

Birth Date
2000-02-02

Type
cat

Visit Date	Description
2023-04-17	breaking leg

Services for PetClinic Application

Frond End:

- Frontend(petclinic)

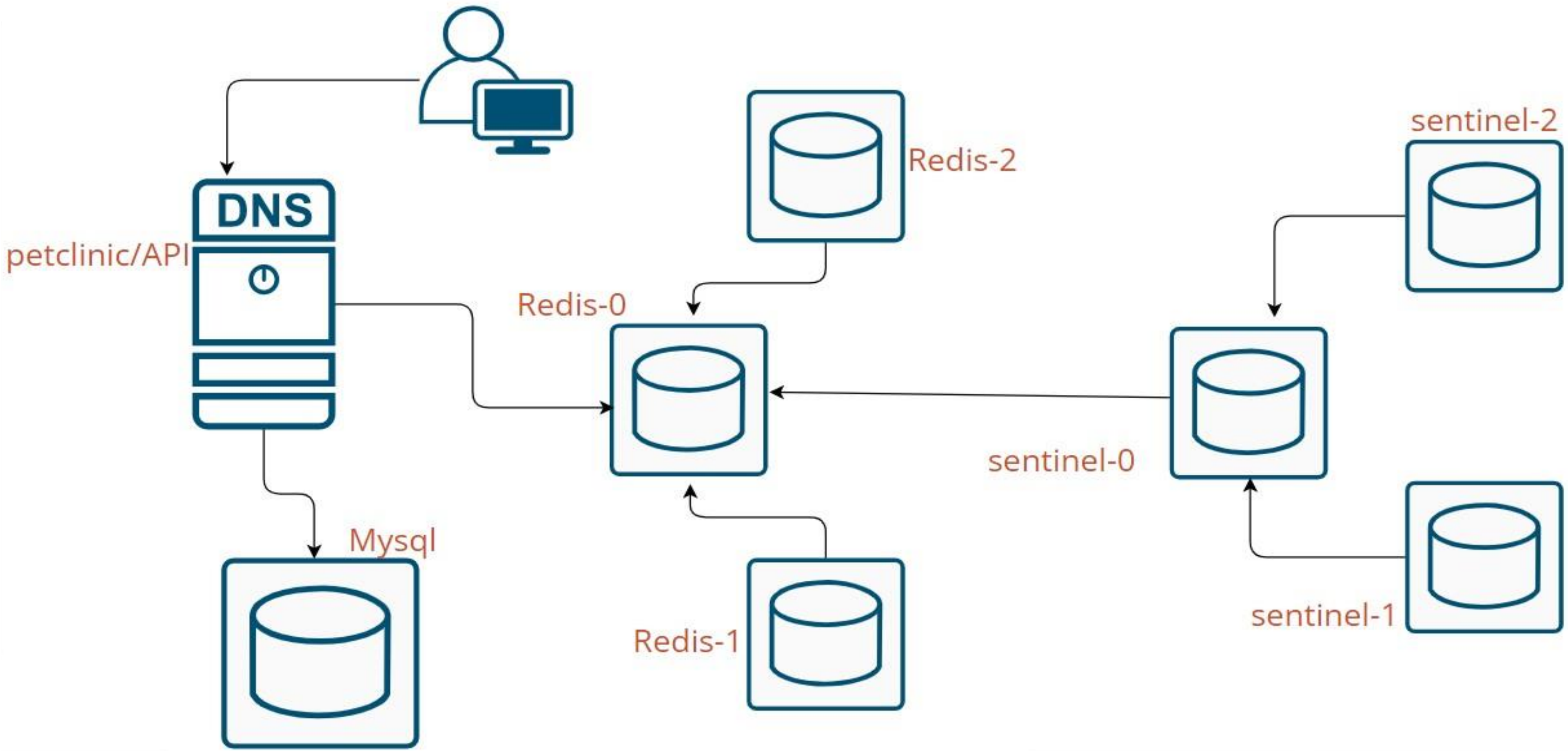
Databases 06:

- Redis catching (03) ==>> For session catching
- Mysql

Redis Failover Management:

- Sentinel (03)

FLOW


DNS
petclinic/API
Redis-2
sentinel-2
Redis-0
sentinel-0
Mysql
Redis-1
sentinel-1

Requirement from DevOps team

1. deploy the current application
2. B- ensure high availability

Instructions from Development team

- The code is available on company s3bucket

- Perform the following command to access it
  - Wget https://group5-braincells.s3.amazonaws.com/petclinic-docker.zip
- Then cd inside petclinic directory
- All work are to be done inside this is directory

- **Do the following task before writing your docker-compose instruction:**
  - Type ls you would see 3 directory called **devcontainer, mvn** and **github**
  - Create directory call **.devcontainer**, **.mvn** and **.github**
  - Copy the the content of the devcontainer and paste it inside the .devcontainer directory
    - ie : cp -r devcontainer/* .devcontainer
  - Do thesame task for mvn and github

Instructions from Development team

The who application is made of the following services:

- Petclinic:

environment:

SERVER_PORT=8080

MYSQL_URL=jdbc:mysql://mysqlserver/petclinic

Volume: /app

This services uses an image that's built from the dockerfile called **"Dockerfile.multi", feel free to check on it.**

Ports:

- 8000
- 8080

Note the application is listening on port 8080

Instructions from Development team

The who application is made of the following services:

- **Mysql**

The base image is: **mysql:8**

environment:

- MYSQL_ROOT_PASSWORD=
- MYSQL_ALLOW_EMPTY_PASSWORD=true
- MYSQL_USER=petclinic
- MYSQL_PASSWORD=petclinic
- MYSQL_DATABASE=petclinic

Create files and mount under /etc/redis/redis.conf inside the container

- file name Redis-0:

```
protected-mode no
port 6379

#authentication
masterauth a-very-complex-password-here
requirepass a-very-complex-password-here
```

- file name Redis-1:

```
protected-mode no
port 6379
slaveof redis-0 6379

#authentication
masterauth a-very-complex-password-here
requirepass a-very-complex-password-here
```

- file name Redis-2:

```
protected-mode no
port 6379
slaveof redis-0 6379

#authentication
masterauth a-very-complex-password-here
requirepass a-very-complex-password-here
```

Create files and mount under /etc/redis/sentinel.conf inside the container

Create files : **sentinel-01, sentinel-02, sentinel-03 ,** With The following same contain

```
port 5000
sentinel monitor mymaster redis-0 6379 2
sentinel down-after-milliseconds mymaster 5000
sentinel failover-timeout mymaster 60000
sentinel parallel-syncs mymaster 1
sentinel auth-pass mymaster a-very-complex-password-here
```

Instructions from Development team

For session caching use redis cluster with the following:
- 03 redis nodes with data replication
- 03 redis-sentinels with failover detection

Propose base image :
redis:4.0.2

Special instructions

Start redis replication with following command

```
redis-server /etc/redis/redis.conf
```

Start sentinel with following command

```
redis-sentinel /etc/redis/sentinel.conf
```

Mount this volume on your msql service:

```
mysql_config:/etc/mysql/conf.d
```

GOOD JOB