



DEVOPS EASY LEARNING

S5GROUP5  
BrainsCells  
(K8s/Docker-compose)

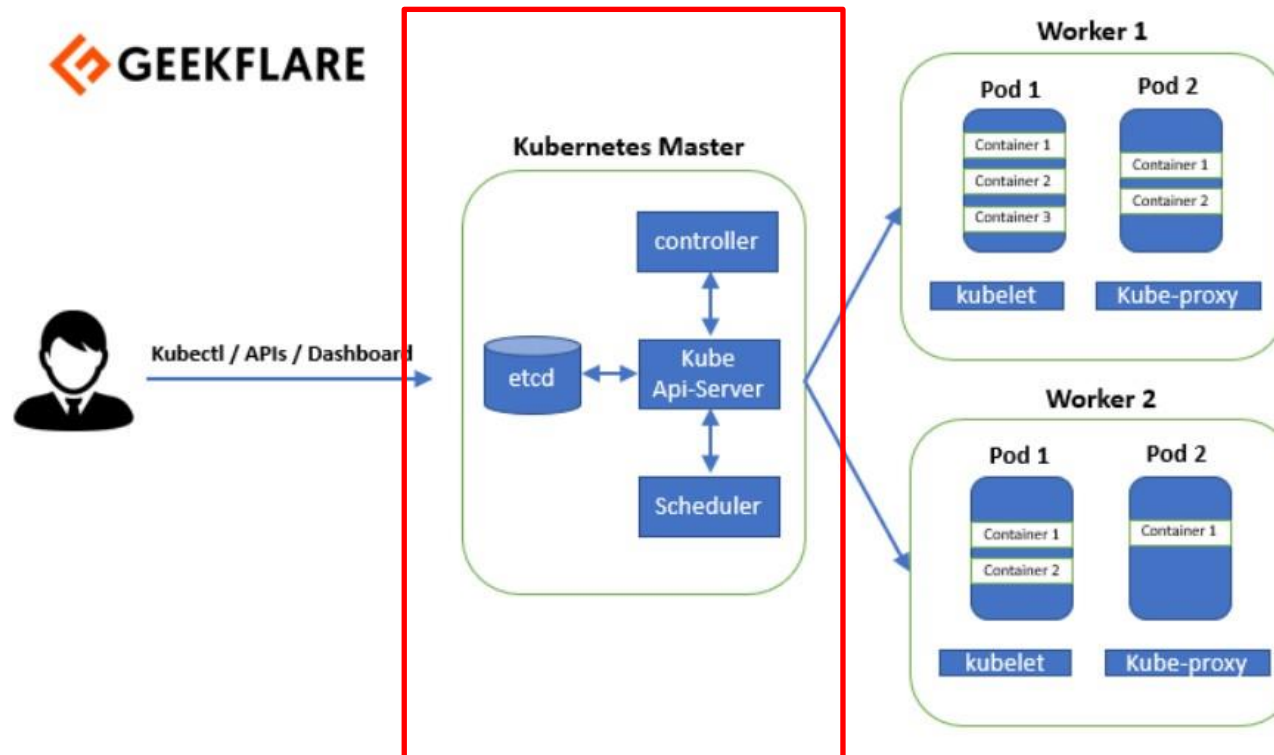
# Questions

1. Give me the difference between the following
  - Docker-compose and K8s
  - On-premise k8s and cloud k8s
2. Can you please opened a 10min discussion about the part of k8s



# Questions

3. Look at the following architecture and discuss about the flow that happened inside the **kubernetes Master**



# The petclinic app

Name: petclinic

Deployment strategie: **Docker-compose**



## GOAL:

- Deploy The petclinic application using docker-compose

## INFO:

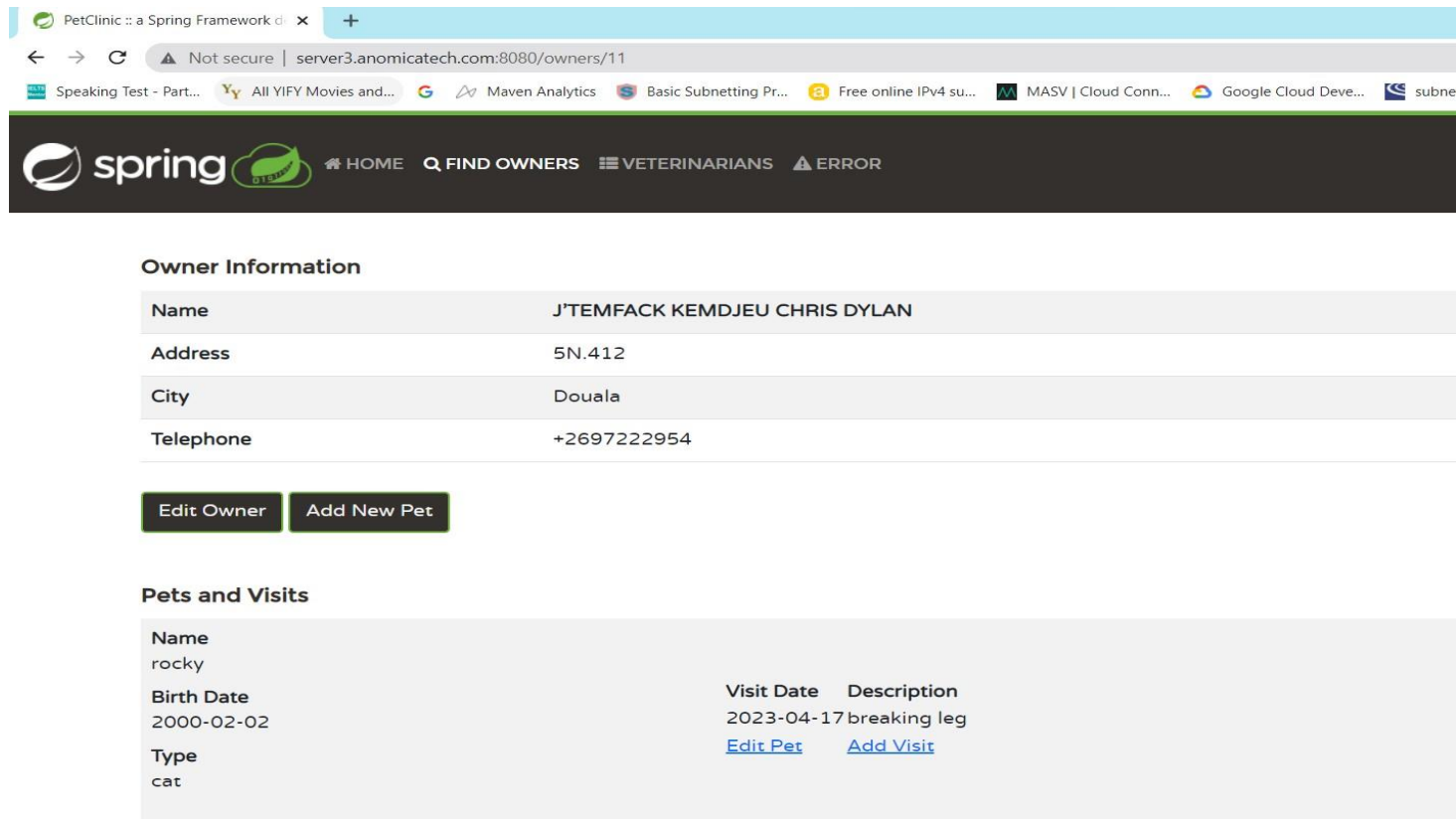
- Customer Name: **Naruto**



# INFO:

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

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



The screenshot shows a web browser window with the address bar displaying "server3.anomicatech.com:8080/owners/11". The page features a dark navigation bar with the Spring logo and menu items: HOME, FIND OWNERS, VETERINARIANS, and an ERROR indicator. Below the navigation bar, the "Owner Information" section displays details for J'TEMFACK KEMDJEU CHRIS DYLAN, including address (5N.412), city (Douala), and telephone (+2697222954). Two buttons, "Edit Owner" and "Add New Pet", are located below the owner information. The "Pets and Visits" section shows details for a pet named "rocky", including birth date (2000-02-02), type (cat), and a visit record from 2023-04-17 with the description "breaking leg". Links for "Edit Pet" and "Add Visit" are provided for the visit record.

**Owner Information**

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

[Edit Owner](#) [Add New Pet](#)

**Pets and Visits**

Name	rocky	Visit Date	2023-04-17	Description	breaking leg
Birth Date	2000-02-02			<a href="#">Edit Pet</a>	<a href="#">Add Visit</a>
Type	cat				



# Services for PetClinic Application

## **Fronnd End:**

- Frontend(petclinic)

## **Databases 06:**

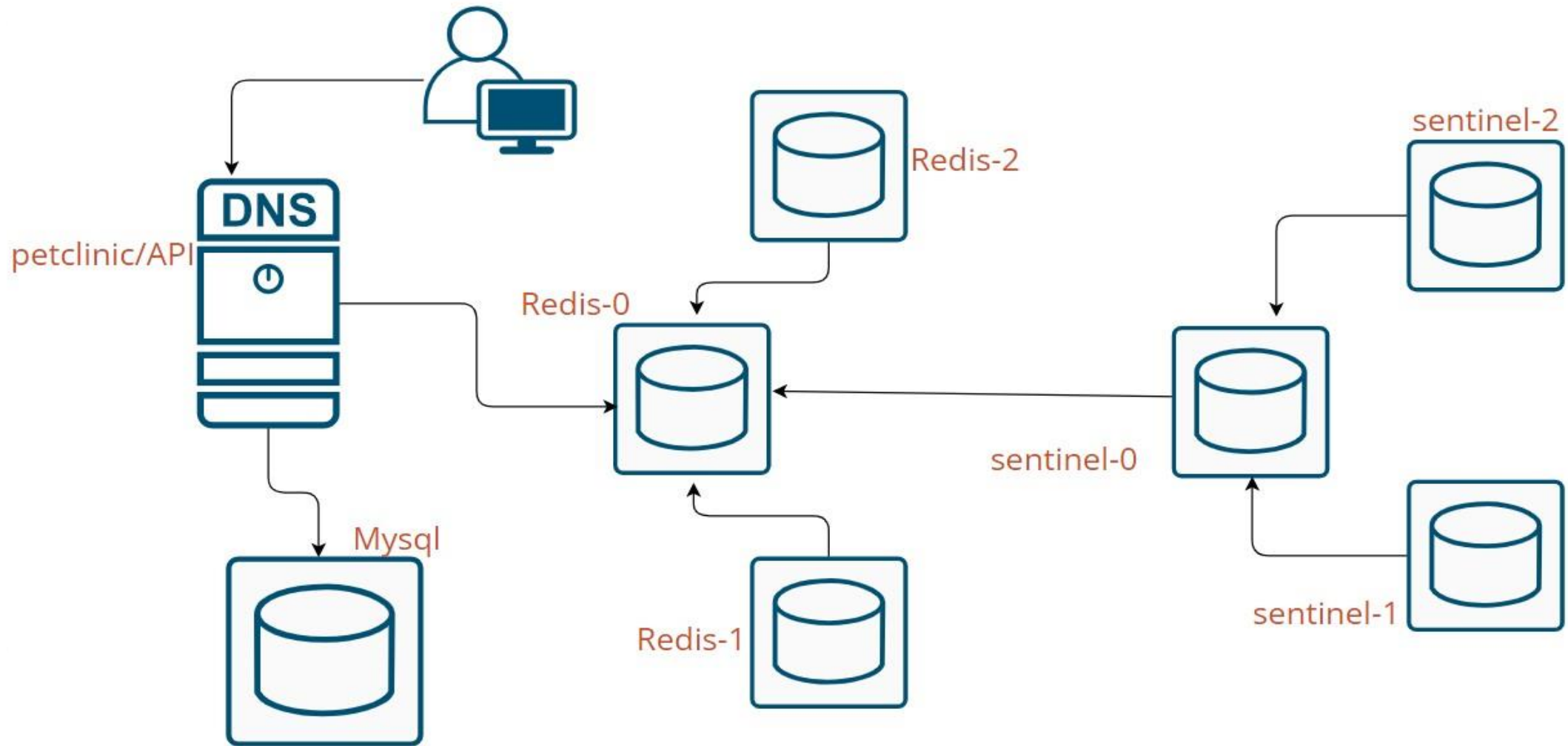
- Redis catching (03) ==>> For session catching
- Mysql

## **Redis Failover Management:**

- Sentinel (03)



# FLOW





# 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 mysql service:**

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





# GOOD JOB

