Universidad del Caribe

2000 CANCUN, QUINTANA ROO, MÉXICO

CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO

Load Balanced Wordpress

with Galera DB and Haproxy

Materia: Computo de alto desempeño

Docente: Ismael Jimenez

Programa Educativo: IDeIO

Nombre: Ricardo Alecio Garcia

Introducción

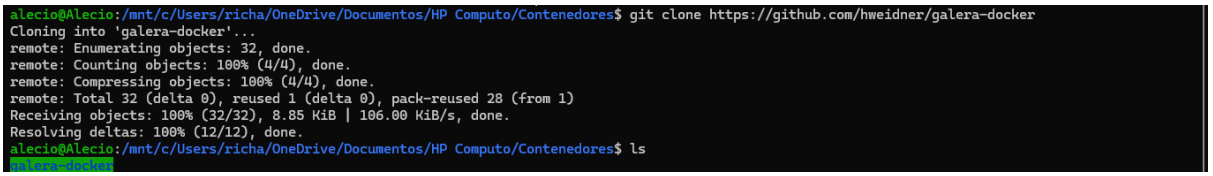
Esta guía explica cómo usar Docker y HAproxy para crear dos grupos de servidores que sigan funcionando aunque haya fallos. Un grupo es para MariaDB (con 3 servidores que se copian información entre sí para evitar problemas) y el otro para WordPress (con 3 "mini-servidores" que comparten el trabajo para que la página web vaya más rápido). HAproxy se encarga de dirigir el tráfico a estos grupos de forma inteligente.

Software

- Ubuntu
- Docker
- Wordpress
- Vs code

Procedimiento

Clonamos el repositorio de github “galera-docker”



Corremos el docker build

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores$ cd galera-docker/
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/galera-docker$ docker build -t mycluster/ga
lera .
[+] Building 81.2s (9/9) FINISHED                                                                          docker:default
 => [internal] load build definition from Dockerfile                                                                0.1s
 => => transferring dockerfile: 296B                                                                                0.0s
 => [internal] load metadata for docker.io/library/mariadb:10.6                                                     2.4s
 => [internal] load .dockerignore                                                                                   0.1s
 => => transferring context: 2B                                                                                     0.0s
 => [1/4] FROM docker.io/library/mariadb:10.6@sha256:ec79aa7a81a7667885cb69b6dc0415e032f22520bd5aca77927faffca43   74.8s
 => => resolve docker.io/library/mariadb:10.6@sha256:ec79aa7a81a7667885cb69b6dc0415e032f22520bd5aca77927faffca432   0.0s
 => => sha256:9ba04180eb73666d85aae623f21ae2af74920509fe4f63a5d442c54e9b3c9f08 8.22kB / 8.22kB                      0.2s
 => => sha256:bd7c8454e7132ba24fdc22fd1559b18d8e426c3102006bab7a159efc125d61b9 4.02kB / 4.02kB                      0.3s
 => => sha256:69fd8973c35eb4e314aa533a679008d79759a189a4bea0ca40270a8f15592606 86.18MB / 86.18MB                   68.8s
 => => sha256:4ed4d4e0fdab8e0a9d2452c23b1cb477d70539918d4684908e8c45c4d2f4d2ff 333B / 333B                          0.5s
 => => sha256:94c5f1bc995c1d1b1e58e0690c7824c7bdc65c139697336d87f438990fcd6d43 114B / 114B                          0.9s
 => => sha256:041c4f82201d46b54e2d8e05905afaa38bf578c8d3adb31a1d754945e4dd02ed 7.18MB / 7.18MB                     40.1s
 => => sha256:2fbe43d24be4124e2ab8fe9e578607e9381e18edc2bc7915b6d7fb3130a487cf 0B / 1.72kB                          77.8s
 => => sha256:d9802f032d6798e2086607424bfe88cb8ec1d6f116e11cd99592dcaf261e9cd2 27.51MB / 27.51MB                    33.6s
 => => extracting sha256:d9802f032d6798e2086607424bfe88cb8ec1d6f116e11cd99592dcaf261e9cd2                            3.7s
 => => extracting sha256:2fbe43d24be4124e2ab8fe9e578607e9381e18edc2bc7915b6d7fb3130a487cf                            0.1s
 => => extracting sha256:041c4f82201d46b54e2d8e05905afaa38bf578c8d3adb31a1d754945e4dd02ed                            2.5s
 => => extracting sha256:94c5f1bc995c1d1b1e58e0690c7824c7bdc65c139697336d87f438990fcd6d43                            0.1s
 => => extracting sha256:4ed4d4e0fdab8e0a9d2452c23b1cb477d70539918d4684908e8c45c4d2f4d2ff                            0.1s
 => => extracting sha256:69fd8973c35eb4e314aa533a679008d79759a189a4bea0ca40270a8f15592606                            5.6s
 => => extracting sha256:bd7c8454e7132ba24fdc22fd1559b18d8e426c3102006bab7a159efc125d61b9                            0.0s
 => => extracting sha256:9ba04180eb73666d85aae623f21ae2af74920509fe4f63a5d442c54e9b3c9f08                            0.0s
 => [internal] load build context                                                                                   0.1s
 => => transferring context: 505B                                                                                   0.1s
 => [2/4] RUN touch /tmp/.wsrep-new-cluster && chown -R mysql:mysql /tmp/.wsrep-new-cluster                         0.9s
```

Revisamos las imagenes del docker

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/galera-docker$ docker images
REPOSITORY          TAG       IMAGE ID       CREATED             SIZE
mycluster/galera    latest    5ae4df856710   About a minute ago  539MB
custom-nginx        latest    94ed8bc9ba7c   4 months ago        279MB
python              latest    bc78d3c007f8   5 months ago        1.47GB
ubuntu              latest    278628f08d49   5 months ago        117MB
my-nginx            v1        bc5eac5eafc5   6 months ago        279MB
nginx               latest    bc5eac5eafc5   6 months ago        279MB
alpine              latest    1e42bbe25081   7 months ago        12.1MB
hello-world         latest    305243c734571  23 months ago       24.4kB
appropriate/curl    latest    c8bf5bbec639   7 years ago         9.86MB
```

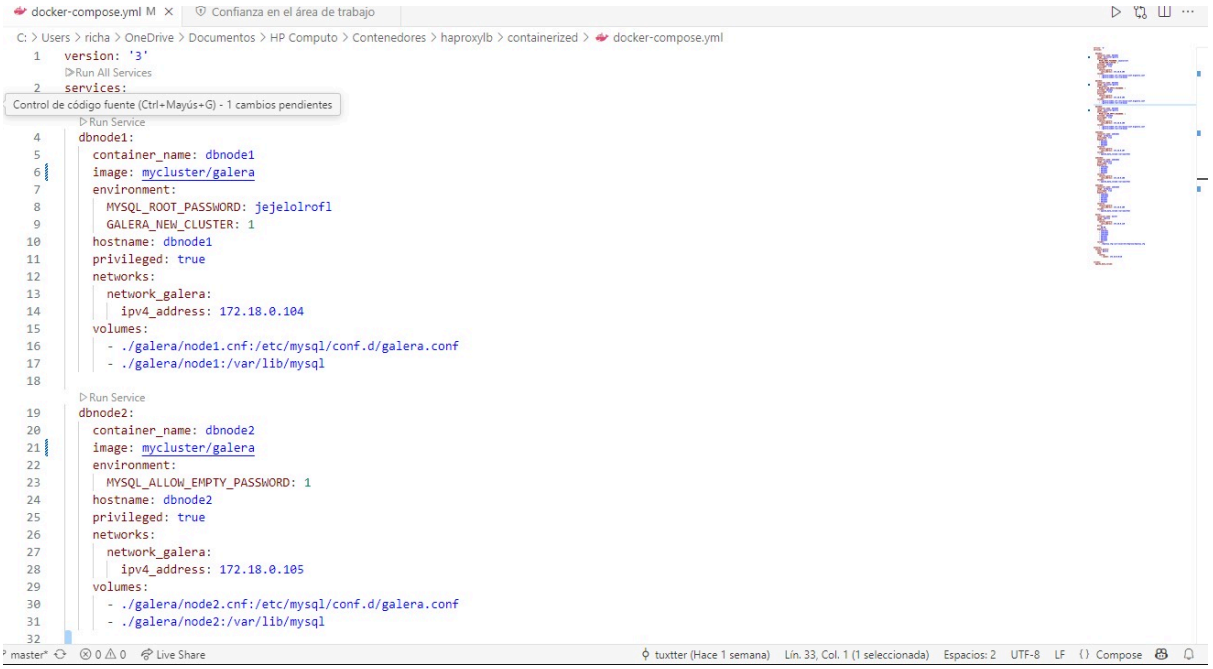
validamos la imagen

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/galera-docker$ docker images | grep galera
mycluster/galera    latest    5ae4df856710   3 minutes ago   539MB
```

Clonamos el nuevo repo con el docker-compose

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores$ git clone https://github.com/tuxtter/haproxylb
Cloning into 'haproxylb'...
remote: Enumerating objects: 84, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 84 (delta 2), reused 8 (delta 2), pack-reused 75 (from 1)
Receiving objects: 100% (84/84), 22.92 KiB | 419.00 KiB/s, done.
Resolving deltas: 100% (43/43), done.
```

en el archivo .yml cambiamos el nombre imagen a la que creamos anteriormente y en este caso tambien las direcciones IP




```yaml
version: '3'
services:
  dbnode1:
    container_name: dbnode1
    image: mycluster/galera
    environment:
      MYSQL_ROOT_PASSWORD: jejelolrofl
      GALERA_NEW_CLUSTER: 1
    hostname: dbnode1
    privileged: true
    networks:
      network_galera:
        ipv4_address: 172.18.0.104
    volumes:
      - ./galera/node1.cnf:/etc/mysql/conf.d/galera.conf
      - ./galera/node1:/var/lib/mysql

  dbnode2:
    container_name: dbnode2
    image: mycluster/galera
    environment:
      MYSQL_ALLOW_EMPTY_PASSWORD: 1
    hostname: dbnode2
    privileged: true
    networks:
      network_galera:
        ipv4_address: 172.18.0.105
    volumes:
      - ./galera/node2.cnf:/etc/mysql/conf.d/galera.conf
      - ./galera/node2:/var/lib/mysql
```

repetimos el paso en los 3 nodos

```
▷ Run Service
dbnode1:
  container_name: dbnode1
  image: mycluster/galera
  environment:
    MYSQL_ROOT_PASSWORD: jejelolrofl
    GALERA_NEW_CLUSTER: 1
  hostname: dbnode1
  privileged: true
  networks:
    network_galera:
      ipv4_address: 172.10.0.101
  volumes:
    - ./galera/node1.cnf:/etc/mysql/conf.d/galera.conf
    - ./galera/node1:/var/lib/mysql

▷ Run Service
dbnode2:
  container_name: dbnode2
  image: mycluster/galera
  environment:
    MYSQL_ALLOW_EMPTY_PASSWORD: 1
  hostname: dbnode2
  privileged: true
  networks:
    network_galera:
      ipv4_address: 172.10.0.102
  volumes:
    - ./galera/node2.cnf:/etc/mysql/conf.d/galera.conf
    - ./galera/node2:/var/lib/mysql
```

```
Run Service
dbnode3:
  container_name: dbnode3
  image: mycluster/galera
  environment:
    MYSQL_ALLOW_EMPTY_PASSWORD: 1
  hostname: dbnode3
  privileged: true
  networks:
    network_galera:
      ipv4_address: 172.10.0.103
  volumes:
    - ./galera/node3.cnf:/etc/mysql/conf.d/galera.conf
    - ./galera/node3:/var/lib/mysql

Run Service
webnode1:
  container_name: webnode1
  image: wordpress
  privileged: true
  depends_on:
    - dbnode1
    - dbnode2
    - dbnode3
  networks:
    network_galera:
      ipv4_address: 172.10.0.104
  volumes:
    - apache_data_volume:/var/www/html

Run Service
webnode2:
  container_name: webnode2
  image: wordpress
  privileged: true
  depends_on:
    - webnode1
    - dbnode1
    - dbnode2
    - dbnode3
  networks:
    network_galera:
      ipv4_address: 172.10.0.105
  volumes:
    - apache_data_volume:/var/www/html

Run Service
webnode3:
  container_name: webnode3
  image: wordpress
  privileged: true
  depends_on:
    - webnode1
    - webnode2
    - dbnode1
    - dbnode2
    - dbnode3
  networks:
    network_galera:
      ipv4_address: 172.10.0.106
  volumes:
    - apache_data_volume:/var/www/html
```

```
▷ Run Service
master:
  container_name: master
  image: haproxy
  networks:
    network_galera:
      ipv4_address: 172.10.0.107
  ports:
    - 80:80
  depends_on:
    - webnode1
    - webnode2
    - webnode3
    - dbnode1
    - dbnode2
    - dbnode3
  volumes:
    - ./haproxy.cfg:/usr/local/etc/haproxy/haproxy.cfg

networks:
  network_galera:
    name: galera
    ipam:
      config:
        - subnet: 172.10.0.0/16
```

Revisamos la estructura de las carpetas

```
  apache_data_volume:alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$ ls -al
total 4
drwxrwxrwx 1 alecio alecio 4096 Apr  6 20:30 .
drwxrwxrwx 1 alecio alecio 4096 Apr  6 20:30 ..
-rwxrwxrwx 1 alecio alecio 2425 Apr  6 20:30 docker-compose.yml
drwxrwxrwx 1 alecio alecio 4096 Apr  6 20:30 galera
-rwxrwxrwx 1 alecio alecio  564 Apr  6 20:30 haproxy.cfg
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$ ls -al galera/
total 0
drwxrwxrwx 1 alecio alecio 4096 Apr  6 20:30 .
drwxrwxrwx 1 alecio alecio 4096 Apr  6 20:30 ..
-rwxrwxrwx 1 alecio alecio  193 Apr  6 20:30 node1.cnf
-rwxrwxrwx 1 alecio alecio  193 Apr  6 20:30 node2.cnf
-rwxrwxrwx 1 alecio alecio  193 Apr  6 20:30 node3.cnf
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$
```

creamos una carpeta para cada nodo

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$ mkdir -p galera/node{1,2,3}
```

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$ ls -al galera
total 0
drwxrwxrwx 1 alecio alecio 4096 Apr  6 21:04 .
drwxrwxrwx 1 alecio alecio 4096 Apr  6 20:30 ..
drwxrwxrwx 1 alecio alecio 4096 Apr  6 21:04 node1
-rwxrwxrwx 1 alecio alecio  193 Apr  6 20:30 node1.cnf
drwxrwxrwx 1 alecio alecio 4096 Apr  6 21:04 node2
-rwxrwxrwx 1 alecio alecio  193 Apr  6 20:30 node2.cnf
drwxrwxrwx 1 alecio alecio 4096 Apr  6 21:04 node3
-rwxrwxrwx 1 alecio alecio  193 Apr  6 20:30 node3.cnf
```

cambiamos los permisos de las carpetas

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$ sudo chown 999:999 galera/node{1,2,3}
[sudo] password for alecio:
```

validamos con netstat

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$ netstat -tlpn
(No info could be read for "-p": geteuid()=1000 but you should be root.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
```

revisamos la lista de contenedores

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS                      PORTS                    NAMES
e9aa325a52c3   ubuntu:latest  "sleep infinity"         4 months ago    Exited (255) 2 hours ago                             my_container
c5bfeb405751   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    80/tcp                   service2
7d0152fb0252   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    80/tcp                   service1
18d3380300f8   alpine         "sleep infinity"         4 months ago    Exited (255) 2 hours ago                             isolated-container
eaa91c1f7039   nginx          "/docker-entrypoint.…"   4 months ago    Exited (1) 4 months ago                              host-container
c5e21f0ff9cd   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    80/tcp                   container3
7e0cdf4175c3   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago                             host-networked
f1a675b9531b   nginx          "/docker-entrypoint.…"   4 months ago    Created                                              exposed-container
0a992336e52b   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    80/tcp                   container2
a651a80f5b29   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    80/tcp                   container1
63f92cc7c6a8   nginx          "/docker-entrypoint.…"   4 months ago    Up 2 hours                  80/tcp                   nginx-restart
faeb8d9917b4   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    80/tcp                   nginx-networked
7ceae465ce7b   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    80/tcp                   nginx-limited
2c238aaa0531   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    80/tcp                   nginx-env
fa16f60cefe5   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    0.0.0.0:8081->80/tcp     nginx-volume
5a9803761761   nginx          "/docker-entrypoint.…"   4 months ago    Created                                              nginx-mapped
9e21ba63e589   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    80/tcp                   my-nginx
117ce870dbce   nginx          "/docker-entrypoint.…"   4 months ago    Exited (0) 4 months ago                              great_kalam
95a6bd809761   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    80/tcp                   limited-nginx
7acdda3b5835   ubuntu         "sleep infinity"         4 months ago    Exited (255) 2 hours ago                             env-file-test
0aa1aa4bd1bc   ubuntu         "sleep infinity"         4 months ago    Exited (255) 2 hours ago                             env-test
ad234f19edd8   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    0.0.0.0:8080->80/tcp     nginx-with-port
7a73d2d40833   nginx          "/docker-entrypoint.…"   4 months ago    Exited (255) 2 hours ago    80/tcp                   nginx-detached
```

levantamos el docker-compose

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$ docker-compose up
[+] Running 8/8
 ✔Network galera      Created                                                                                                0.2s
 ✔Container dbnode3   Created                                                                                                1.1s
 ✔Container dbnode1   Created                                                                                                1.1s
 ✔Container dbnode2   Created                                                                                                1.1s
 ✔Container webnode1  Created                                                                                                0.9s
 ✔Container webnode2  Created                                                                                                0.6s
 ✔Container webnode3  Created                                                                                                1.0s
 ✔Container master    Created                                                                                                1.0s
Attaching to dbnode1, dbnode2, dbnode3, master, webnode1, webnode2, webnode3
dbnode2    | 2025-04-07 04:02:43+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.6.21+maria~ubu2004 started.
dbnode3    | 2025-04-07 04:02:43+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.6.21+maria~ubu2004 started.
dbnode1    | 2025-04-07 04:02:43+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.6.21+maria~ubu2004 started.
dbnode3    | 2025-04-07 04:02:45+00:00 [Note] [Entrypoint]: Initializing database files
dbnode3    | Warning: World-writable config file '/etc/mysql/conf.d/01-galera.cnf' is ignored
dbnode2    | 2025-04-07 04:02:45+00:00 [Note] [Entrypoint]: Initializing database files
dbnode1    | 2025-04-07 04:02:45+00:00 [Note] [Entrypoint]: Initializing database files
dbnode2    | Warning: World-writable config file '/etc/mysql/conf.d/01-galera.cnf' is ignored
dbnode1    | Warning: World-writable config file '/etc/mysql/conf.d/01-galera.cnf' is ignored
dbnode3    | Warning: World-writable config file '/etc/mysql/conf.d/01-galera.cnf' is ignored
dbnode2    | Warning: World-writable config file '/etc/mysql/conf.d/01-galera.cnf' is ignored
dbnode1    | Warning: World-writable config file '/etc/mysql/conf.d/01-galera.cnf' is ignored
webnode1   | WordPress not found in /var/www/html - copying now...
webnode2   | WordPress not found in /var/www/html - copying now...
webnode2   | WARNING: /var/www/html is not empty! (copying anyhow)
```

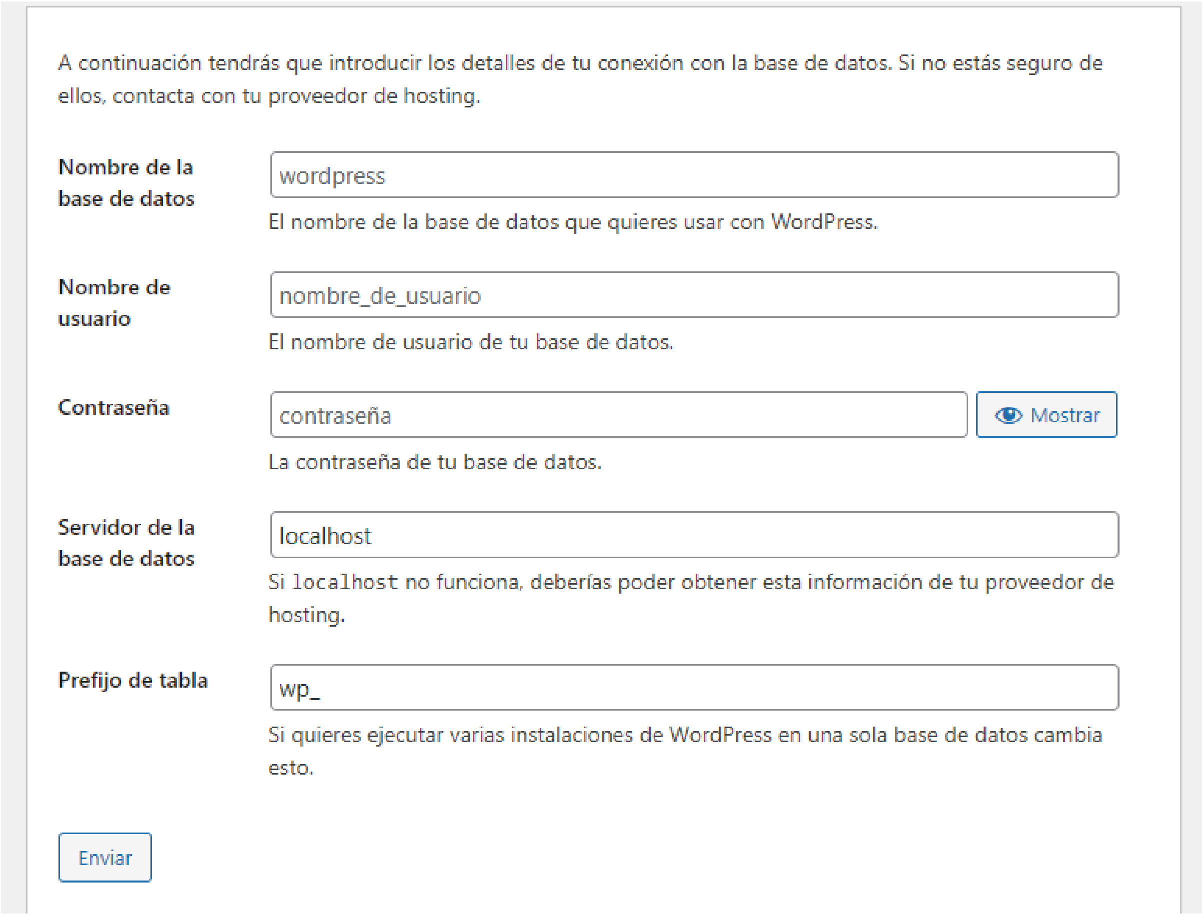
Revisamos con el ps

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$ docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                  NAMES
ad07b1395bf6   haproxy            "docker-entrypoint.s…"   18 seconds ago   Up 9 seconds    0.0.0.0:80->80/tcp     master
06feababd655   wordpress          "docker-entrypoint.s…"   18 seconds ago   Up 10 seconds   80/tcp                 webnode3
6f99481ab488   wordpress          "docker-entrypoint.s…"   19 seconds ago   Up 12 seconds   80/tcp                 webnode2
8ac5f7f4e7a4   wordpress          "docker-entrypoint.s…"   20 seconds ago   Up 13 seconds   80/tcp                 webnode1
ff5352b21b14   mycluster/galera   "docker-entrypoint.s…"   21 seconds ago   Up 15 seconds   3306/tcp               dbnode3
16a8de2016c2   mycluster/galera   "docker-entrypoint.s…"   21 seconds ago   Up 15 seconds   3306/tcp               dbnode1
e5164116fafc   mycluster/galera   "docker-entrypoint.s…"   21 seconds ago   Up 15 seconds   3306/tcp               dbnode2
63f92cc7c6a8   nginx              "/docker-entrypoint.…"   4 months ago     Up 4 hours      80/tcp                 nginx-restart
```

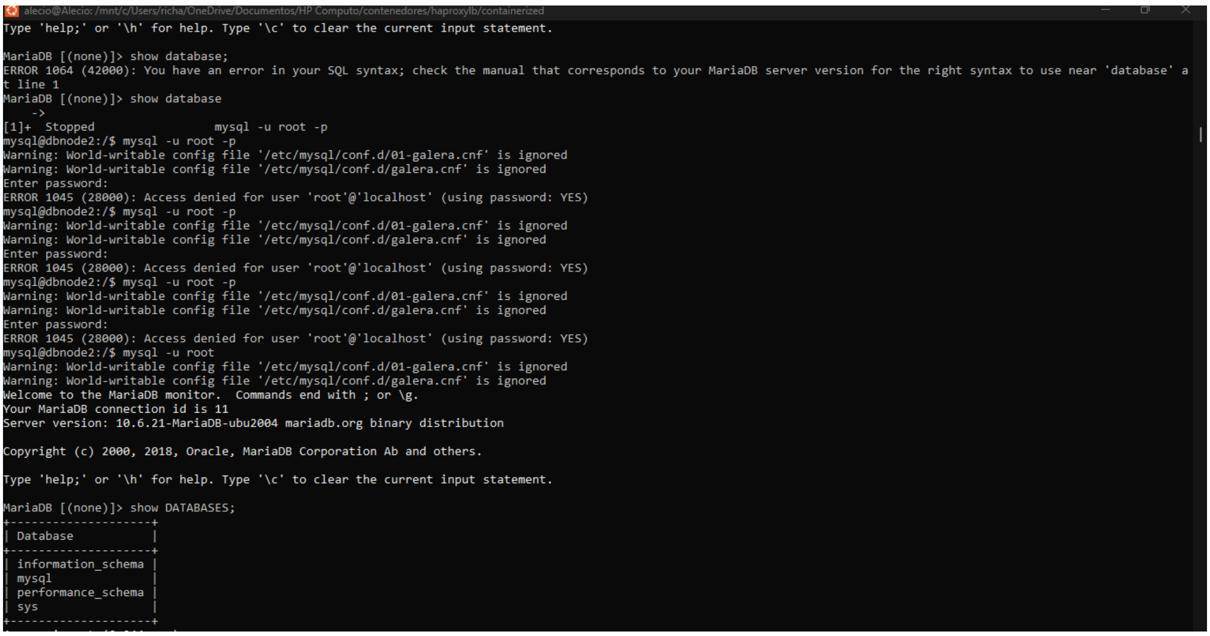
Revisamos que el LISTEN este en el puerto 80

```
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$ netstat -tlpn
(No info could be read for "-p": geteuid()=1000 but you should be root.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
alecio@Alecio:/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores/haproxylb/containerized$
```

Nos vamos al navegador ponemos “Localhost” y configuramos el wordpress



revisamos las bases de datos

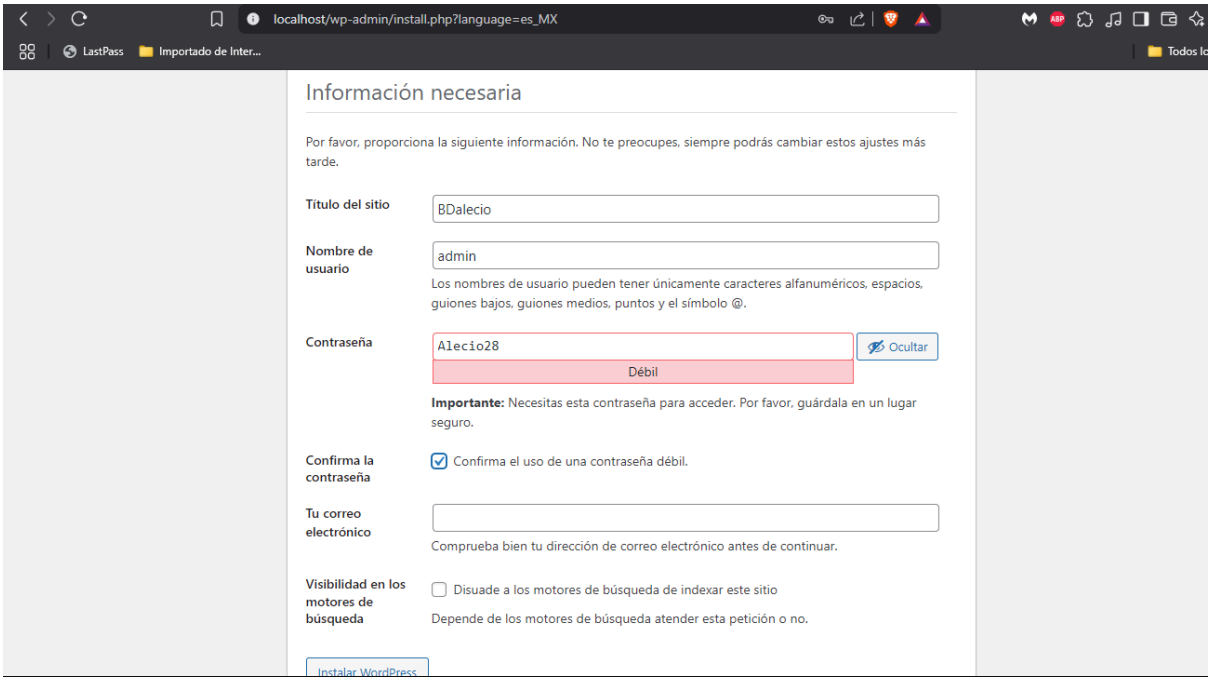


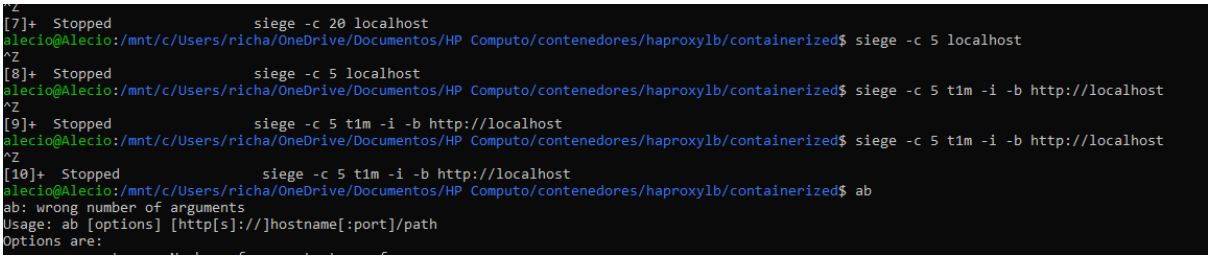
creamos la base de datos wordpress

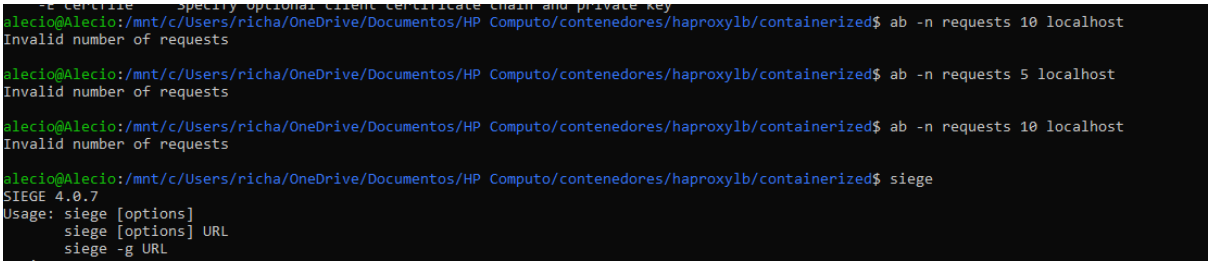
```
MariaDB [(none)]> create database wordpress;
Query OK, 1 row affected (0.019 sec)

MariaDB [(none)]>
```

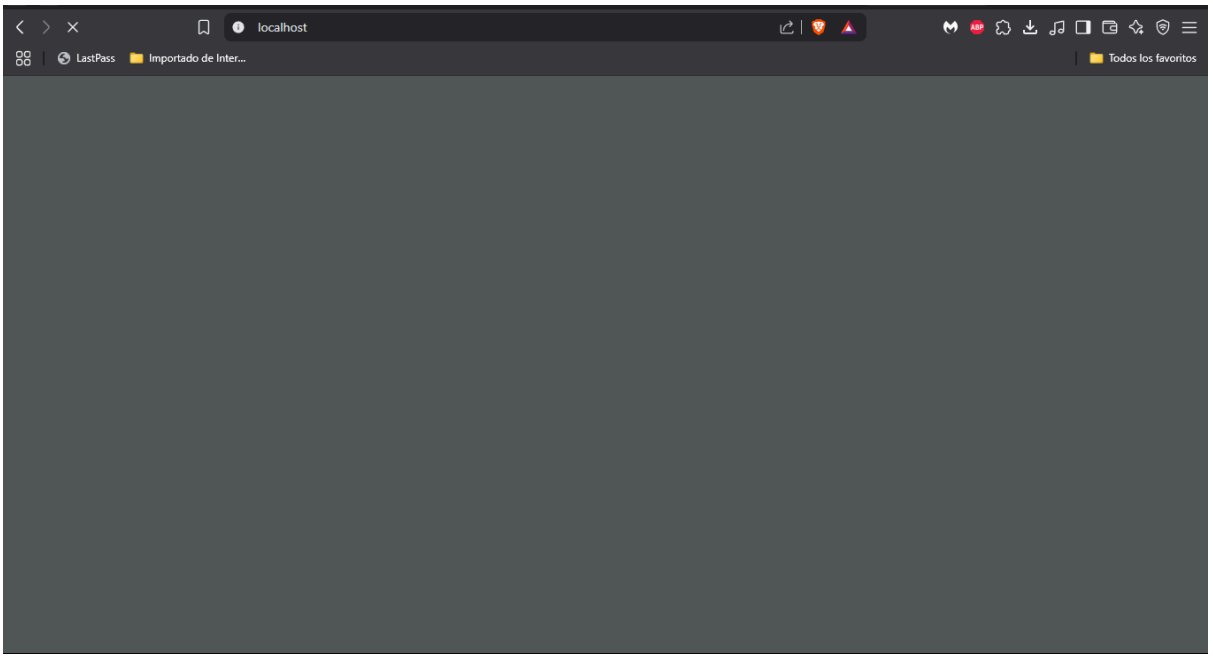
```
MariaDB [(none)]> show DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| wordpress          |
+--------------------+
```

configuramos el wordpress









Conclusión

Esta práctica tuvo como objetivo implementar una infraestructura basada en Docker y HAProxy, utilizando clústers de bases de datos MariaDB (Galera) compuesto por tres nodos replicados, y un conjunto distribuido de tres instancias de WordPress balanceadas mediante HAProxy. Este último fue configurado como el balanceador de carga principal, encargado de redirigir el tráfico entrante de manera eficiente y tolerante a fallos.

Durante el proceso se lograron avances significativos: los contenedores fueron desplegados exitosamente, se configuraron los archivos requeridos y los servicios se levantaron sin errores críticos al inicio. Sin embargo, las pruebas de carga realizadas con herramientas como *siege* y *ab* no arrojaron resultados concluyentes, lo cual impidió validar adecuadamente el comportamiento del entorno bajo condiciones de estrés.

Los principales desafíos surgieron en la formación del clúster Galera, particularmente en lo referente a problemas de permisos y sincronización que impidieron la replicación efectiva entre los nodos. Adicionalmente, a pesar de que los contenedores de WordPress y HAProxy se mantenían activos, el tráfico HTTP no fue redirigido correctamente. Esto indica posibles fallos tanto en la configuración de HAProxy como en la disponibilidad real de los servicios backend.

Si bien la guía que nos proporcionó el profesor fue sólida, la experiencia evidenció que la resiliencia real de un sistema distribuido no radica únicamente en el número de nodos desplegados, sino en la precisión con la que estos se integran, configuran y monitorean. Corregir detalladamente los errores de replicación en MariaDB, asegurar el enrutamiento efectivo de HAProxy y confirmar la disponibilidad operativa de cada instancia son tareas fundamentales que deben anteceder cualquier evaluación de rendimiento o escalabilidad.