

Universidad del Caribe

2000

CANCUN, QUINTANA ROO, MÉXICO

CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO

Load Balanced Wordpress with Galera DB and Haproxy

Asignatura:

Computo de Alto desempeño

Alumno:

Jesus Alberto Perera Santiago

Matricula:

200300592

Profesor:

Ismael Jimenez

Fecha de entrega:

07/04/2025

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.

Desarrollo

Clonamos el repositorio de github "galera-docker"

```
/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores$ git clone https://github.com/hweidner/galera-docker
galera-docker'...
ating objects: 32, done.
ng objects: 100% (4/4), done.
ssing objects: 100% (4/4), done.
32 (delta 0), reused 1 (delta 0), pack-reused 28 (from 1)
cts: 100% (32/32), 8.85 KiB | 106.00 KiB/s, done.
as: 100% (12/12), done.
/mnt/c/Users/richa/OneDrive/Documentos/HP Computo/Contenedores$ ls
```

```

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
=> => sha256:4ed4d4e0fdab8e9a9d2452c23b1cb477d70539918d4684908e8c45c4d2f4d2ff 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:d9802f032d6798e2086607424bfe88cb8ecd6f116e11cd99592dcafc261e9cd2 27.51MB / 27.51MB 33.6s
=> => extracting sha256:d9802f032d6798e2086607424bfe88cb8ecd6f116e11cd99592dcafc261e9cd2 3.7s
=> => extracting sha256:2fbe43d24be4124e2ab8fe9e578607e9381e18edc2bc7915b6d7fb3130a487cf 0.1s
=> => extracting sha256:041c4f82201d46b54e2d8e05905afaa38bf578c8d3adb31a1d754945e4dd02ed 2.5s
=> => extracting sha256:94c5f1bc995c1d1b1e58e0690c7824c7bdc65c139697336d87f438990fcd6d43 0.1s
=> => extracting sha256:4ed4d4e0fdab8e9a9d2452c23b1cb477d70539918d4684908e8c45c4d2f4d2ff 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

The screenshot shows a Docker Compose file for a Galera MySQL cluster. The file is titled 'docker-compose.yml' and is located in the directory 'C:\Users\richa\OneDrive\Documentos\HP Computo\Contenedores\haproxy\containerized'. The file defines two services, 'dbnode1' and 'dbnode2', both using the 'mycluster/galera' image. 'dbnode1' has environment variables 'MYSQL_ROOT_PASSWORD: jejelolrof1' and 'GALERA_NEW_CLUSTER: 1', and is connected to a network named 'network_galera' with IP address '172.18.0.104'. 'dbnode2' has environment variables 'MYSQL_ALLOW_EMPTY_PASSWORD: 1' and is connected to the same network with IP address '172.18.0.105'. Both nodes have volumes for '/etc/mysql/conf.d/galera.conf' and '/var/lib/mysql'.

Repetimos en los 3 nodo

▷ 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
```

```

32  ▸ Run Service
33  dbnode3:
34      container_name: dbnode3
35      image: mycluster/galera
36      environment:
37          MYSQL_ALLOW_EMPTY_PASSWORD: 1
38      hostname: dbnode3
39      privileged: true
40      networks:
41          network_galera:
42              ipv4_address: 172.10.0.103
43      volumes:
44          - ./galera/node3.cnf:/etc/mysql/conf.d/galera.conf
45          - ./galera/node3:/var/lib/mysql
46
47  ▸ Run Service
48  webnode1:
49      container_name: webnode1
50      image: wordpress
51      privileged: true
52      depends_on:
53          - dbnode1
54          - dbnode2
55          - dbnode3
56      networks:
57          network_galera:
58              ipv4_address: 172.10.0.104
59      volumes:
60          - apache_data_volume:/var/www/html
61
62  ▸ Run Service
63  webnode2:
64      container_name: webnode2
65      image: wordpress
66      privileged: true
67      depends_on:
68          - webnode1
69          - dbnode1
70          - dbnode2
71          - dbnode3
72      networks:
73          network_galera:
74              ipv4_address: 172.10.0.105
75      volumes:
76          - apache_data_volume:/var/www/html
77
78  ▸ Run Service
79  webnode3:
80      container_name: webnode3
81      image: wordpress
82      privileged: true
83      depends_on:
84          - webnode1
85          - webnode2
86          - dbnode1
87          - dbnode2
88          - dbnode3
89      networks:
90          network_galera:
91              ipv4_address: 172.10.0.106
92      volumes:
93          - 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 lista de contenedores

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e9aa325a52c3	ubuntu:latest	"sleep infinity"	4 months ago	Exited (255) 2 hours ago		my_container
c5bfeb408751	nginx	"/docker-entrypoint..."	4 months ago	Exited (255) 2 hours ago	80/tcp	service2
7d0152fb8252	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
ea91c1f7039	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
7e0cd447f5c3	nginx	"/docker-entrypoint..."	4 months ago	Exited (255) 2 hours ago		host-networked
1a675b9531b	nginx	"/docker-entrypoint..."	4 months ago	Created		exposed-container
0a023316e52b	nginx	"/docker-entrypoint..."	4 months ago	Exited (255) 2 hours ago	80/tcp	container2
a651180f5b29	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
7cae465c01784	nginx	"/docker-entrypoint..."	4 months ago	Exited (255) 2 hours ago	80/tcp	nginx-networked
7cae465ce7b7	nginx	"/docker-entrypoint..."	4 months ago	Exited (255) 2 hours ago	80/tcp	nginx-limited
2c238aa0531	nginx	"/docker-entrypoint..."	4 months ago	Exited (255) 2 hours ago	80/tcp	nginx-env
1a16f68cefa5	nginx	"/docker-entrypoint..."	4 months ago	Exited (255) 2 hours ago	0.0.0.0:8081->80/tcp	nginx-volume
5a9803716171	nginx	"/docker-entrypoint..."	4 months ago	Created		nginx-mapped
0e21ba63e589	nginx	"/docker-entrypoint..."	4 months ago	Exited (255) 2 hours ago		my/nginx
117c8070dbce	nginx	"/docker-entrypoint..."	4 months ago	Exited (0) 4 months ago	80/tcp	great_kalam
95a6b4d89761	nginx	"/docker-entrypoint..."	4 months ago	Exited (255) 2 hours ago		limited-nginx
7acdda3b5835	ubuntu	"sleep infinity"	4 months ago	Exited (255) 2 hours ago	80/tcp	env-file-test
0aa1aa4b4db1bc	ubuntu	"sleep infinity"	4 months ago	Exited (255) 2 hours ago		env-test
ad234f140e4d8	nginx	"/docker-entrypoint..."	4 months ago	Exited (255) 2 hours ago	0.0.0.0:8080->80/tcp	nginx-with-port
7a734140a833	nginx	"/docker-entrypoint..."	4 months ago	Exited (255) 2 hours ago	80/tcp	nginx-datapod

levantamos el docker-compose

```

✓ Network galera Created 0.2s
✓ Container dbnode3 Created 1.1s
✓ Container dbnode1 Created 1.3s
✓ 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
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...
```

Revisamos con el 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
06feabab6d55	wordpress	"docker-entrypoint.s..."	18 seconds ago	Up 10 seconds	80/tcp	webnode1
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	mysqlcluster/galera	"docker-entrypoint.s..."	21 seconds ago	Up 15 seconds	3306/tcp	dbnode3
16a8de2016c2	mysqlcluster/galera	"docker-entrypoint.s..."	21 seconds ago	Up 15 seconds	3306/tcp	dbnode1
5e16411616cf	mysqlcluster/galera	"docker-entrypoint.s..."	21 seconds ago	Up 15 seconds	3306/tcp	dbnode2
63f92cc7c6a8	nginx	"docker-entrypoint.s..."	4 months ago	Up 4 hours	80/tcp	nginx-restart

Revisamos que el LISTEN este en el puerto 80

```
(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      -
```


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

A continuación tendrás que introducir los detalles de tu conexión con la base de datos. Si no estás seguro de ellos, contacta con tu proveedor de hosting.

Nombre de la base de datos	<input type="text" value="wordpress"/>	El nombre de la base de datos que quieres usar con WordPress.
Nombre de usuario	<input type="text" value="nombre_de_usuario"/>	El nombre de usuario de tu base de datos.
Contraseña	<input type="password" value="contraseña"/> Mostrar	La contraseña de tu base de datos.
Servidor de la base de datos	<input type="text" value="localhost"/>	Si localhost no funciona, deberías poder obtener esta información de tu proveedor de hosting.
Prefijo de tabla	<input type="text" value="wp_"/>	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

[Enviar](#)

revisamos las bases de datos

```
alecio@Aleco: ~/mnt/2/Users/ricna/OneDrive/Documentos/HP Computo/contenedores/haproxyib/containerized
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'database' a
t line 1
MariaDB [(none)]> show database
->
[1]+  Stopped                  mysql -u root -p
mysql@dbnode2:/$ mysql -u root -p
Warning: World-writable config file '/etc/mysql/conf.d/01-galera.cnf' is ignored
Warning: World-writable config file '/etc/mysql/conf.d/galera.cnf' is ignored
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
mysql@dbnode2:/$ mysql -u root -p
Warning: World-writable config file '/etc/mysql/conf.d/01-galera.cnf' is ignored
Warning: World-writable config file '/etc/mysql/conf.d/galera.cnf' is ignored
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
mysql@dbnode2:/$ mysql -u root -p
Warning: World-writable config file '/etc/mysql/conf.d/01-galera.cnf' is ignored
Warning: World-writable config file '/etc/mysql/conf.d/galera.cnf' is ignored
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
mysql@dbnode2:/$ mysql -u root
Warning: World-writable config file '/etc/mysql/conf.d/01-galera.cnf' is ignored
Warning: World-writable config file '/etc/mysql/conf.d/galera.cnf' is ignored
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.6.21-MariaDB-ubu2004 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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


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

Información necesaria

Por favor, proporciona la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Titulo del sitio: BDalecio

Nombre de usuario: admin

Contraseña: Débil

Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

Confirma la contraseña: ☒ Confirma el uso de una contraseña débil.

Tu correo electrónico:

Visibilidad en los motores de búsqueda: ☐ Disuade a los motores de búsqueda de indexar este sitio.

Depende de los motores de búsqueda atender esta petición o no.

Instalar WordPress

BDalecio

Hola, admin

Acerca de WordPress

Únete

WordPress.org

Documentación

Aprende WordPress

Soporte

Sugerencias

Apariencia

Plugins 1

Usuarios

Herramientas

Ajustes

Cerrar menú

Presentamos Twenty Twenty-Five

Posibilidades infinitas sin complejidad

Twenty Twenty-Five ofrece un tema flexible y centrado en el diseño que te permite crear sitios asombrosos con facilidad. Adapta tu estética con una gran variedad de opciones de estilo, patrones de bloques y paletas de colores. Reducido a lo esencial, es un tema que realmente puede crecer contigo.

Stories

Tell your story

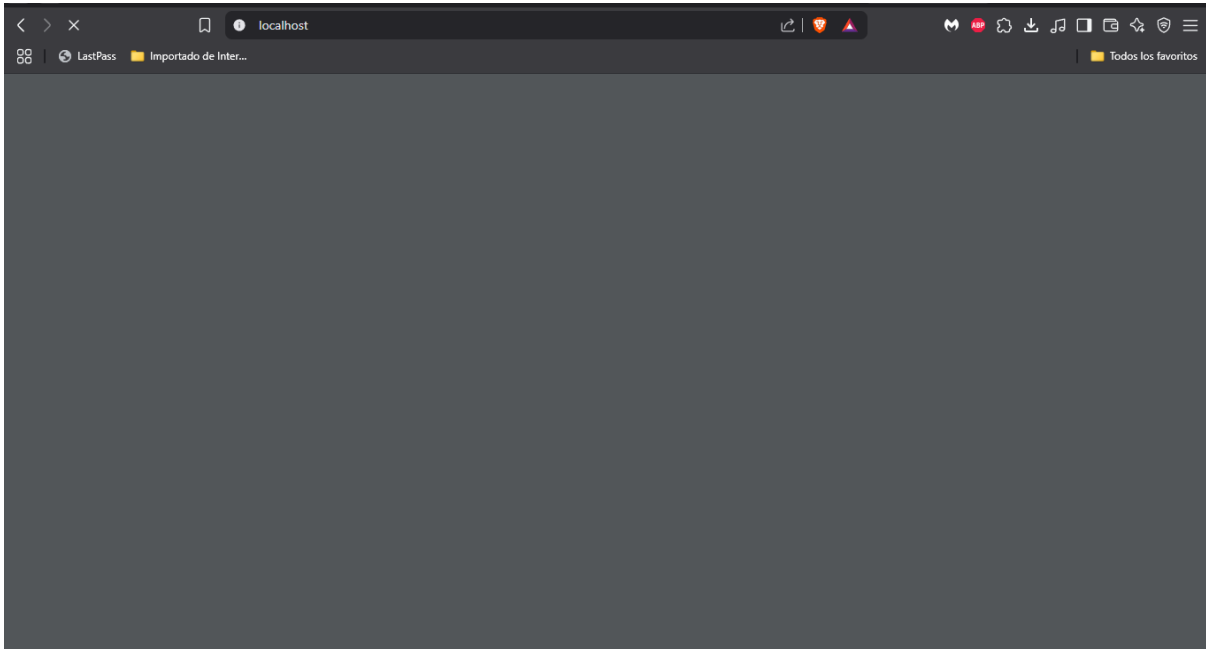
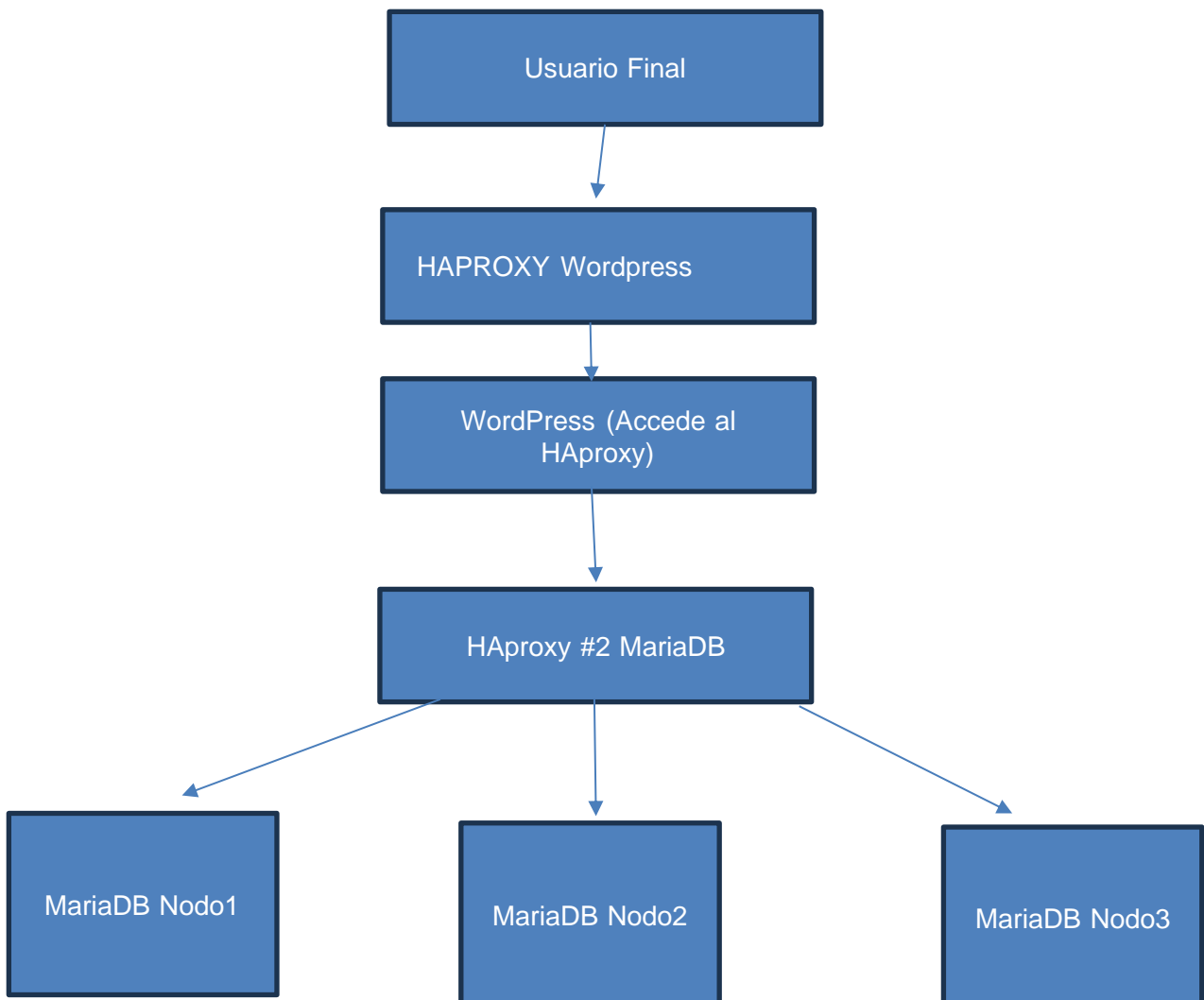


Diagrama de Arquitectura:



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.

