

Universidad
del Caribe

2000

CANCUN, QUINTANA ROO, MÉXICO

CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO

PARCIAL 2

Docente: Ismael Jimenez Sanchez

Asignatura: Cómputo de alto desempeño

Programa educativo: Ingeniería en Datos e
Inteligencia Organizacional

Nombre: Amairani Ramón Perera

Este informe explica la implementación de una arquitectura de alta disponibilidad usando Docker y HAProxy, con el fin de garantizar la operación continua de dos servicios clave: MariaDB y WordPress. Para ello, se crearon clústeres redundantes para cada aplicación. MariaDB se configuró con Galera Cluster en tres nodos para asegurar replicación y tolerancia a fallos, mientras que WordPress se desplegó en tres contenedores balanceados por HAProxy, que distribuye el tráfico entre ellos de forma eficiente. La documentación detalla cómo levantar estos clústeres en contenedores Docker, con HAProxy funcionando como balanceador de carga inteligente, asegurando así estabilidad y disponibilidad ante cualquier interrupción.

Crear carpetas de trabajo

```
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo$ mkdir parcial2
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo$ cd parcial2
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2$ git clone https://github.com/hweidner/galera-docker
Cloning into 'galera-docker'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 32 (delta 0), reused 1 (delta 0), pack-reused 28 (from 1)
Receiving objects: 100% (32/32), 8.85 KiB | 697.00 KiB/s, done.
Resolving deltas: 100% (12/12), done.
```

Clonar repo

```
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker$ docker build -t micluster/galera .
[+] Building 1.1s (10/10) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 296B                                0.0s
=> WARN: MaintainerDeprecated: Maintainer instruction is deprecated in favor of using label (line 2) 0.0s
=> [internal] load metadata for docker.io/library/mariadb:10.6    0.7s
=> [auth] library/mariadb:pull token for registry-1.docker.io     0.0s
=> [internal] load .dockerignore                                   0.0s
=> => transferring context: 2B                                       0.0s
=> [1/4] FROM docker.io/library/mariadb:10.6@sha256:ec79aa7a81a7667885cb69b6dc0415e032f22520bd5aca77927faffca432 0.0s
=> => resolve docker.io/library/mariadb:10.6@sha256:ec79aa7a81a7667885cb69b6dc0415e032f22520bd5aca77927faffca432 0.0s
=> [internal] load build context                                   0.0s
=> => transferring context: 505B                                       0.0s
=> CACHED [2/4] RUN touch /tmp/.wsrep-new-cluster && chown -R mysql:mysql /tmp/.wsrep-new-cluster 0.0s
=> CACHED [3/4] COPY galera.cnf /etc/mysql/conf.d/01-galera.cnf  0.0s
=> CACHED [4/4] COPY startup.sh /startup.sh                       0.0s
=> exporting to image                                              0.1s
=> => exporting layers                                              0.0s
=> => exporting manifest sha256:6cedbc7b2c181f7675b14206dff7101e8ec052857a81e5766657471960bdcfc7 0.0s
=> => exporting config sha256:3af69761a8a8f8ffc754b6dd04f2ce637b103d7428934e61af33a55117c38753 0.0s
=> => exporting attestation manifest sha256:76ff1a160612e5821431c2d87f85f77f95f2a1e4150dbbe6d5c34b7de571c2d2 0.0s
=> => exporting manifest list sha256:717b4040fe86d65f57e98abe4de3d7265ab9784f4444ece8a662b4759099b0f9 0.0s
=> => naming to docker.io/micluster/galera:latest                 0.0s
=> => unpacking to docker.io/micluster/galera:latest              0.0s
```

Imagen

```
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
micluster/galera    latest     717b4040fe86  11 minutes ago 539MB
```

chechar codigo de docker compose y verificar en ubuntu si las direcciones ip son correctas, además de ver si todos los contenedores se han levantado correctamente

```
1 version: '3'
2 services:
3
4   dbnode1:
5     container_name: dbnode1
6     image: micluster/galera
7     environment:
8       MYSQL_ROOT_PASSWORD: jejelolrofl
9       GALERA_NEW_CLUSTER: 1
10    hostname: dbnode1
11    privileged: true
12    networks:
13      network_galera:
14        ipv4_address: 172.18.0.104
15    volumes:
16      - ./galera/node1.cnf:/etc/mysql/conf.d/galera.conf
17      - ./galera/node1:/var/lib/mysql
18
19   dbnode2:
20     container_name: dbnode2
21     image: micluster/galera
22     environment:
23       MYSQL_ALLOW_EMPTY_PASSWORD: 1
24     hostname: dbnode2
25     privileged: true
26     networks:
27       network_galera:
28         ipv4_address: 172.18.0.105
29     volumes:
30       - ./galera/node2.cnf:/etc/mysql/conf.d/galera.conf
31       - ./galera/node2:/var/lib/mysql
32
33   dbnode3:
34     container_name: dbnode3
```

```
ama@LAPTOP-U0PK3NT4: /mnt/c/Computo/parcial2/galera-docker/haproxyb/containerized$ docker compose up -d
WARN[0000] /mnt/c/Computo/parcial2/galera-docker/haproxyb/containerized/docker-compose.yml: the attribute 'version' is
obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 30/30
  ✓ master Pulled 29.5s
  ✓ webnode1 Pulled 45.9s
  ✓ webnode2 Pulled 46.0s
  ✓ webnode3 Pulled 46.0s
[+] Running 9/9
  ✓ Network galera Created 0.1s
  ✓ Volume "containerized_apache_data_volume" Created 0.0s
  ✓ Container dbnode3 Started 21.9s
  ✓ Container dbnode2 Started 21.9s
  ✓ Container dbnode1 Started 21.8s
  ✓ Container webnode1 Started 2.9s
  ✓ Container webnode2 Started 2.9s
  ✓ Container webnode3 Started 3.0s
  ✓ Container master Started 3.2s
```

aquí master no mostraba su ip

Eso normalmente pasa si:

1. El contenedor no arrancó bien.
2. El contenedor no está en la red esperada.
3. Algo salió mal en la definición de red.

```
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy/b/containerized$ for c in dbnode1 dbnode2 dbnode3 webnode1 webnode2 webnode3 master; do
  echo "$c: $(docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' $c)"
done
dbnode1: 172.18.0.104
dbnode2: 172.18.0.105
dbnode3: 172.18.0.106
webnode1: 172.18.0.107
webnode2: 172.18.0.108
webnode3: 172.18.0.109
master:

ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy/b/containerized$ docker inspect dbnode1 | grep IPAddress
ss
      "SecondaryIPAddresses": null,
      "IPAddress": "",
      "IPAddress": "172.18.0.104",
```

El contenedor **master** no levanta porque el archivo **haproxy.cfg** tiene un error en la línea 22, de un salto de línea y corrigiendolo ahora sí el contenedor master está funcionando y tiene su IP asignada correctamente

```
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy/b/containerized$ docker logs master
[NOTICE] (1) : haproxy version is 3.1.6-d929ca2
[NOTICE] (1) : path to executable is /usr/local/sbin/haproxy
[ALERT] (1) : config : parsing [/usr/local/etc/haproxy/haproxy.cfg:22]: Missing LF on last line, file might have been truncated at position 26.
[ALERT] (1) : config : Error(s) found in configuration file : /usr/local/etc/haproxy/haproxy.cfg
[ALERT] (1) : config : Fatal errors found in configuration.
[NOTICE] (1) : haproxy version is 3.1.6-d929ca2
[NOTICE] (1) : path to executable is /usr/local/sbin/haproxy
[ALERT] (1) : config : parsing [/usr/local/etc/haproxy/haproxy.cfg:22]: Missing LF on last line, file might have been truncated at position 26.
[ALERT] (1) : config : Error(s) found in configuration file : /usr/local/etc/haproxy/haproxy.cfg
[ALERT] (1) : config : Fatal errors found in configuration.
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy/b/containerized$ nano haproxy.cfg
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy/b/containerized$ docker restart master
master
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy/b/containerized$ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' master
172.18.0.110
```

Probar que funcione



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 alojamiento.

Nombre de la base de datos

El nombre de la base de datos que quieres usar con WordPress.

Nombre de usuario

El nombre de usuario de tu base de datos.

Contraseña [Mostrar](#)

La contraseña de tu base de datos.

Servidor de la base de datos

Si localhost no funciona, deberías poder obtener esta información de tu proveedor de alojamiento web.

Prefijo de tabla

Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

Resumen de estado actual:

- Contenedores creados y corriendo correctamente (dbnode1, webnode1, master, etc.).
- Red galera funcionando con IPs estáticas, como las definiste en el docker-compose.yml.
- master (HAProxy) ya tiene asignada su IP (172.18.0.110) después de corregir el archivo de configuración.
- Todos los servicios están conectados a la misma red, y no hay conflictos visibles.
- No hay errores activos en los logs de Docker, y HAProxy ya no se queja de la config.

Checar estructura:

```
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy1b/containerized$ ls -al
total 4
drwxrwxrwx 1 ama ama 4096 Apr  7 01:00 .
drwxrwxrwx 1 ama ama 4096 Apr  7 00:51 ..
-rwxrwxrwx 1 ama ama 2425 Apr  7 00:51 docker-compose.yml
drwxrwxrwx 1 ama ama 4096 Apr  7 00:57 galera
-rwxrwxrwx 1 ama ama 566 Apr  7 01:00 haproxy.cfg

ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy1b/containerized$ ls -al galera/
total 0
drwxrwxrwx 1 ama ama 4096 Apr  7 01:18 .
drwxrwxrwx 1 ama ama 4096 Apr  7 01:00 ..
-rwxrwxrwx 1 ama ama 193 Apr  7 00:51 node1.cnf
-rwxrwxrwx 1 ama ama 193 Apr  7 00:51 node2.cnf
-rwxrwxrwx 1 ama ama 193 Apr  7 00:51 node3.cnf
```

Crear carpetas

```
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy1b/containerized$ ls -al galera/
total 0
drwxrwxrwx 1 ama ama 4096 Apr  7 01:19 .
drwxrwxrwx 1 ama ama 4096 Apr  7 01:00 ..
drwxrwxrwx 1 ama ama 4096 Apr  7 01:19 node1
-rwxrwxrwx 1 ama ama 193 Apr  7 00:51 node1.cnf
drwxrwxrwx 1 ama ama 4096 Apr  7 01:19 node2
-rwxrwxrwx 1 ama ama 193 Apr  7 00:51 node2.cnf
drwxrwxrwx 1 ama ama 4096 Apr  7 01:19 node3
-rwxrwxrwx 1 ama ama 193 Apr  7 00:51 node3.cnf
```

cambiar permisos

```
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy1b/containerized$ sudo chown 999:999 galera/node{1,2,3}
[sudo] password for ama:
```

Checar

```
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy1b/containerized$ netstat -tlnp
(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 10.255.255.254:53      0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.54:53         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:40759       0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53         0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                 :::*                    LISTEN      -
tcp6       0      0 :::80                 :::*                    LISTEN      -
ama@LAPTOP-U0PK3NT4:/mnt/c/Computo/parcial2/galera-docker/haproxy1b/containerized$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
fc670e396750	haproxy	"docker-entrypoint.s..."	26 minutes ago	Up 23 minutes	0.0.0.0:80->80/tcp	master
3c6beb61a40e3	wordpress	"docker-entrypoint.s..."	26 minutes ago	Up 26 minutes	80/tcp	webnode
a0e2e63746c72	wordpress	"docker-entrypoint.s..."	26 minutes ago	Up 26 minutes	80/tcp	webnode
837615c06b531	wordpress	"docker-entrypoint.s..."	26 minutes ago	Up 26 minutes	80/tcp	webnode
878f3de79bca	micluster/galera	"docker-entrypoint.s..."	27 minutes ago	Up 26 minutes	3306/tcp	dbnode3
1e5a3c9d2c50	micluster/galera	"docker-entrypoint.s..."	27 minutes ago	Up 26 minutes	3306/tcp	dbnode1
fcc53249960b	micluster/galera	"docker-entrypoint.s..."	27 minutes ago	Up 26 minutes	3306/tcp	dbnode2

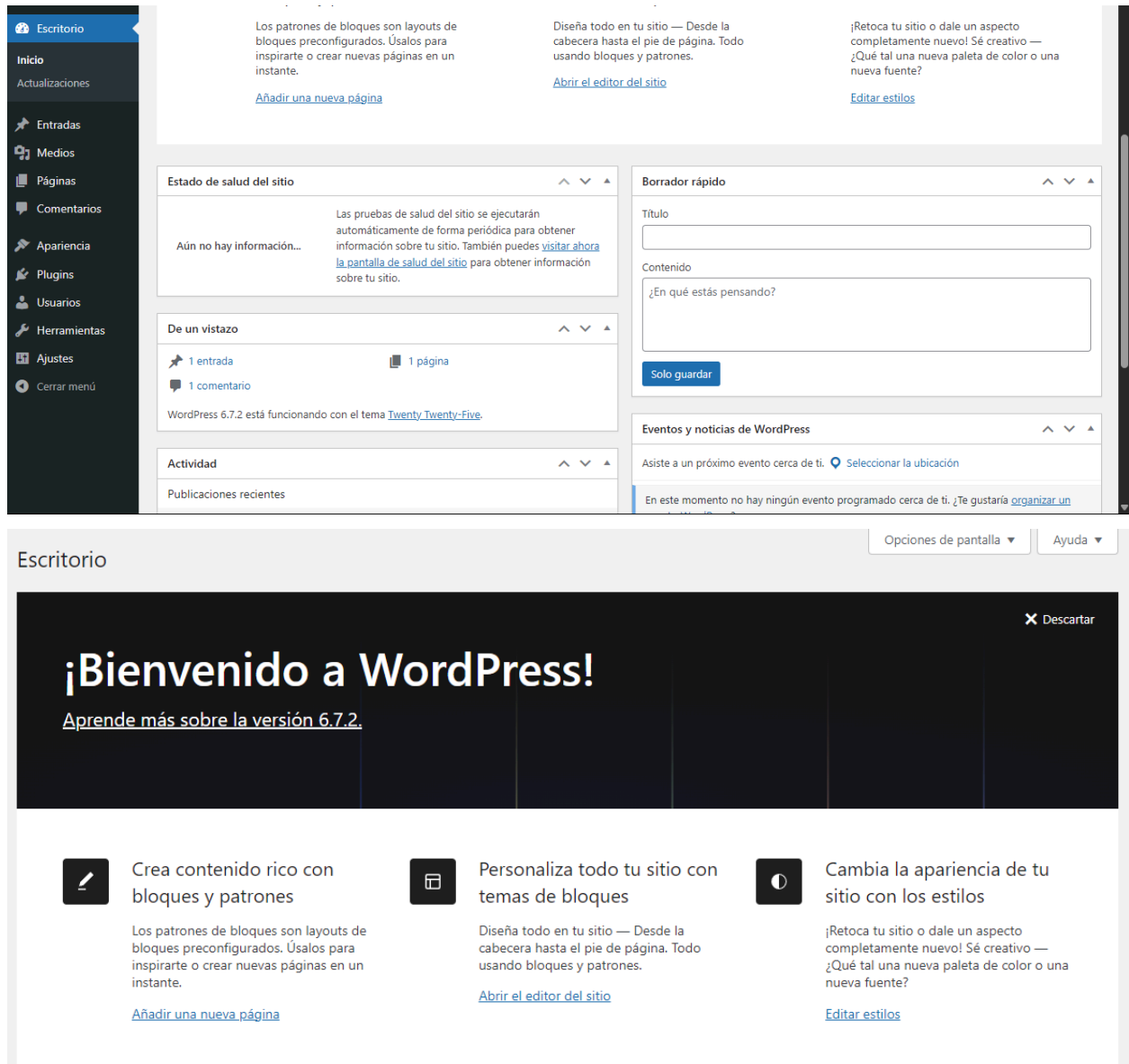
Entrar a un nodo a revisar que no hay database

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.017 sec)
```

Crear una bd

```
MariaDB [(none)]> create database wordpress;
Query OK, 1 row affected (0.005 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| wordpress |
+-----+
5 rows in set (0.010 sec)
```



La implementación de una arquitectura de alta disponibilidad utilizando Docker, HAProxy, MariaDB y WordPress ha logrado asegurar la continuidad y estabilidad de los servicios críticos. Gracias a la configuración de clústeres redundantes y el balanceo de carga inteligente, se garantiza tanto la resiliencia ante fallos como la distribución eficiente del tráfico. La solución permite una gestión más ágil y escalable, aprovechando las ventajas de la contenedorización con Docker y la replicación de datos en MariaDB. Este enfoque proporciona una base sólida para entornos productivos, asegurando un alto rendimiento y disponibilidad constante.