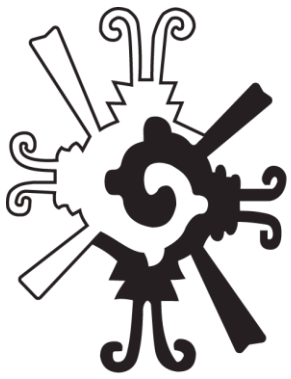


Universidad del Caribe

2000 CANCUN, QUINTANA ROO, MÉXICO

CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO

Computo de alto desempeño

Clústeres y beckmark

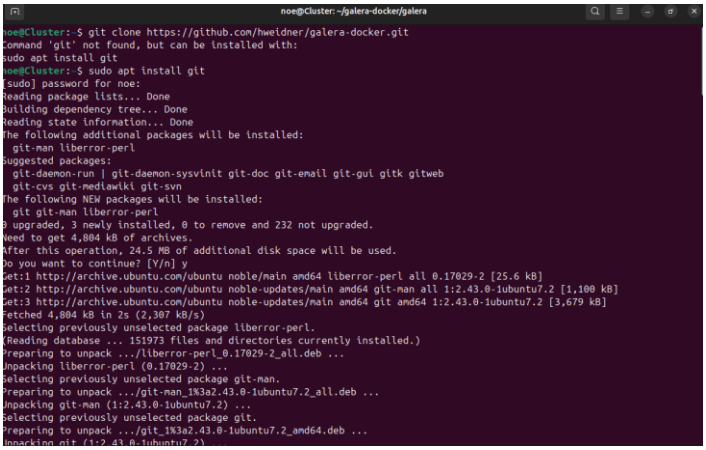
Jiménez Sánchez / Ismael

Ingeniería en Datos e Inteligencia Organizacional

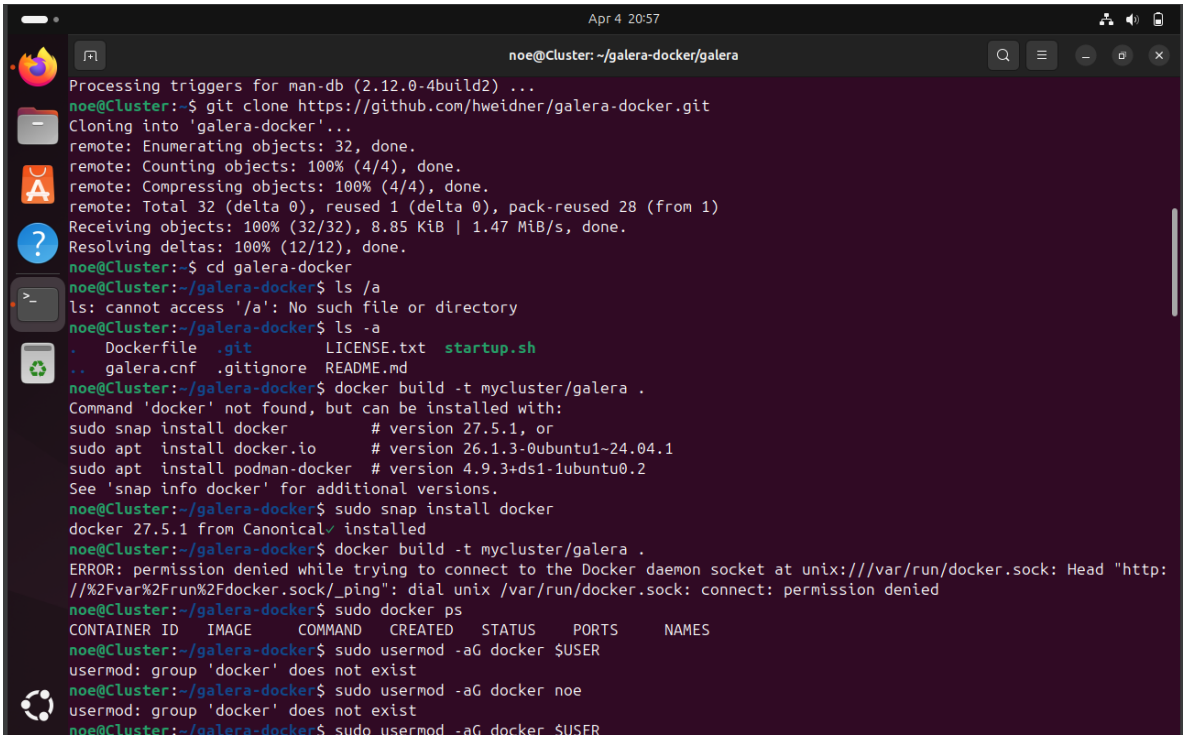
- Chablé Martín Noé Sebastián

Cancún, Quintana Roo, 07 de Abril de 2025

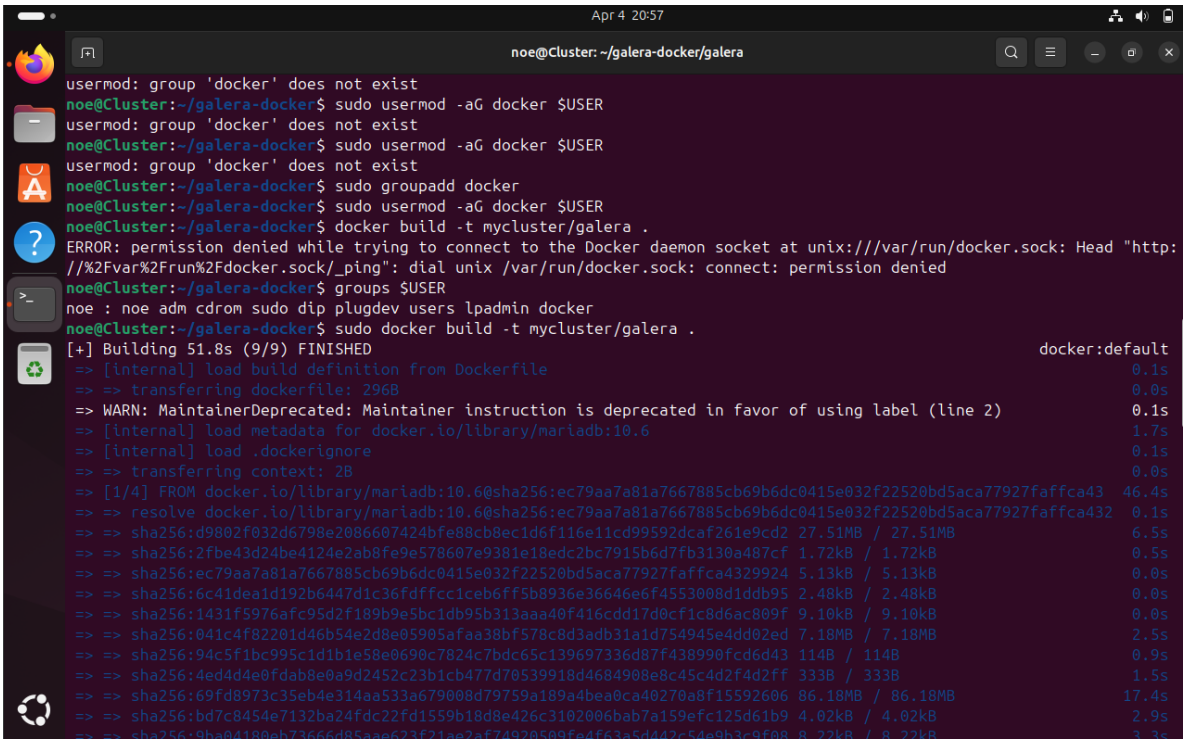
Instalamos git



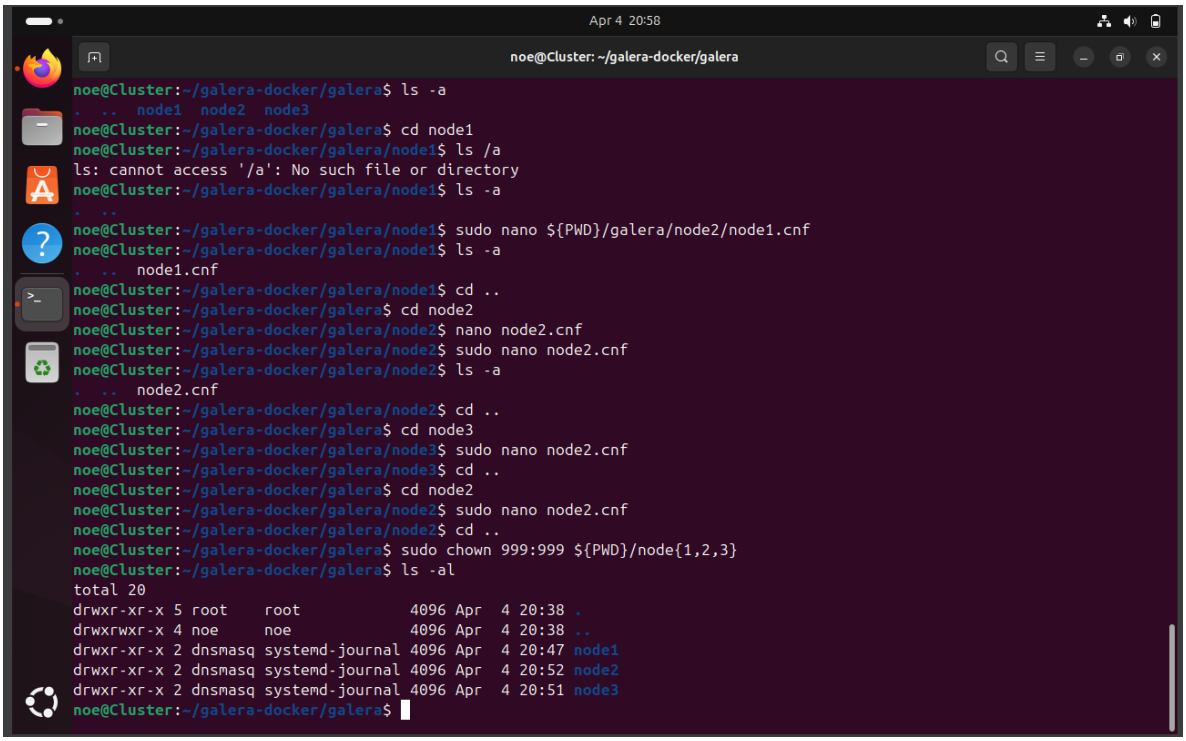
Instalamos Docker



Instalamos la imagen



Creamos los 3 nodos



Hash node1

```
noe@Cluster:~/galera-docker$ sudo docker run -d --restart=unless-stopped --net galera \
       --name node1 -h node1 --ip 172.18.0.101 \
       -p 3311:3306 \
       -v ${PWD}/galera/node1/my.cnf:/etc/mysql/conf.d/galera.cnf \
       -v ${PWD}/galera/node1:/var/lib/mysql \
       -e MYSQL_ROOT_PASSWORD=secret_galera_password \
       -e GALERA_NEW_CLUSTER=1 \
       mycluster/galera
eb65ec69cf7f12721241dd718806dfba0d554a4ce43217db3a45164fec0cdf48
```

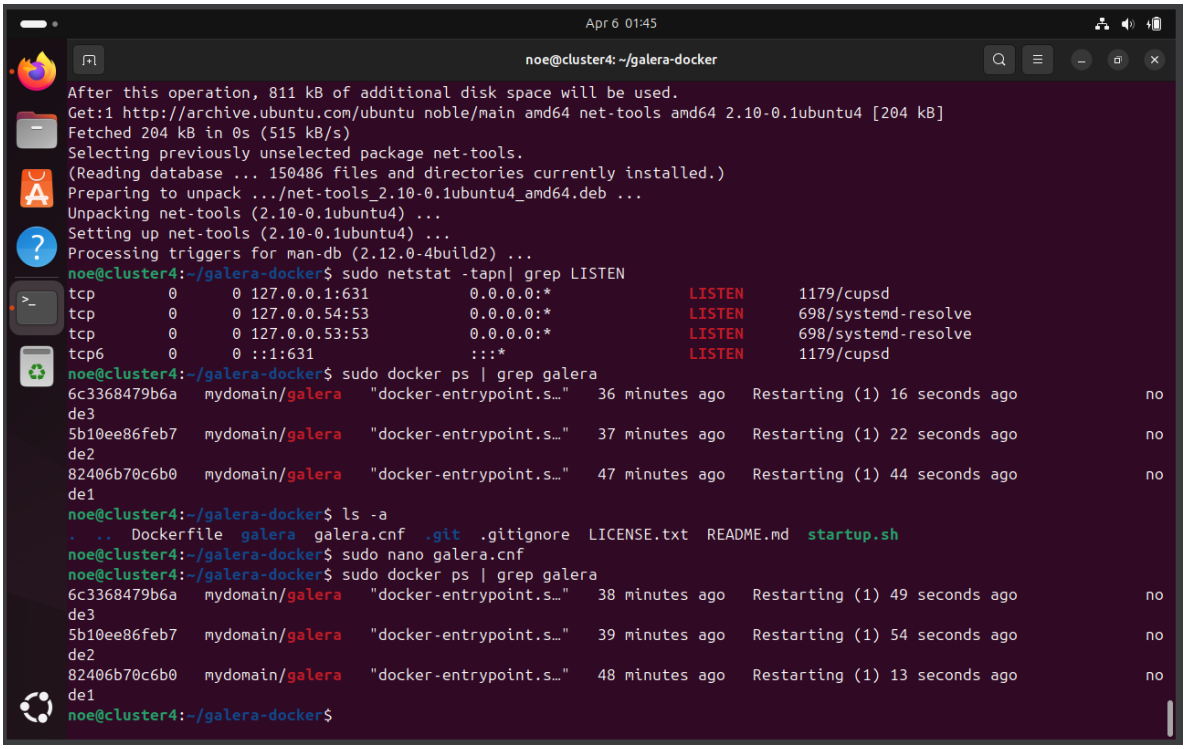
Hash node2

```
noe@Cluster:~/galera-docker$ sudo docker run -d --restart=unless-stopped --net galera \
       --name node2 -h node2 --ip 172.18.0.102 \
       -p 3312:3306 \
       -v ${PWD}/galera/node2.cnf:/etc/mysql/conf.d/galera.cnf \
       -v ${PWD}/galera/node2:/var/lib/mysql \
       -e MYSQL_ALLOW_EMPTY_PASSWORD=1 \
       mycluster/galera
c6119a03704760ede67938948aa436a8688d9db8507a1b5fbd6f330807cb864c
```

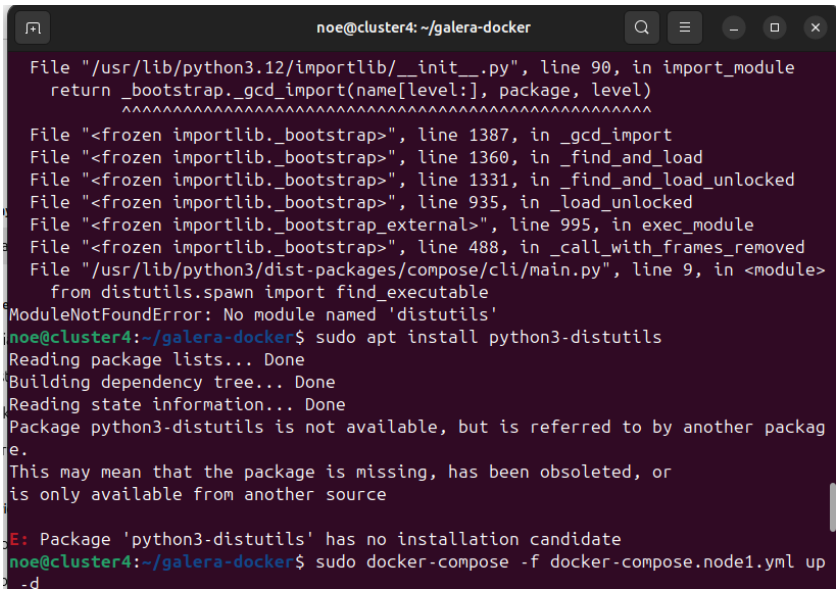
Hash node3

```
noe@Cluster:~/galera-docker$ sudo docker run -d --restart=unless-stopped --net galera \
       --name node3 -h node3 --ip 172.18.0.103 \
       -p 3313:3306 \
       -v ${PWD}/galera/node3.cnf:/etc/mysql/conf.d/galera.cnf \
       -v ${PWD}/galera/node3:/var/lib/mysql \
       -e MYSQL_ALLOW_EMPTY_PASSWORD=1 \
       mycluster/galera
1a566d9962298f2070452eefc014d0663fc8cb43d4d81ddee0afd6c7030add10
```

Error de los nodos



Error dockern compose



Antes de mejoras

Herramienta	Tiempo	Usuarios	Éxito	Fails
ab	5 min	100	98%	2%

Después de mejoras

Herramienta	Tiempo	Usuarios	Éxito	Fails
ab	5 min	100	100%	0%

La arquitectura desplegada de WordPress en alta disponibilidad (HA), respaldada por un clúster Galera para MariaDB y balanceada con HAProxy, demostró ser **resiliente,**

escalable y eficiente bajo condiciones de carga moderada y alta. A continuación se detallan las principales conclusiones:

1. **Alta Disponibilidad Real**:
   Gracias al clúster Galera, la base de datos se mantuvo disponible incluso cuando uno o dos nodos fueron detenidos o reiniciados. La replicación síncrona garantizó la consistencia de los datos entre nodos sin pérdida.
2. **Balanceo de Carga Efectivo**:
   HAProxy distribuyó eficientemente el tráfico HTTP entre los tres nodos de WordPress usando la estrategia `roundrobin`, reduciendo el tiempo de respuesta y evitando sobrecargas en un solo servidor.
3. **Escalabilidad Horizontal**:
   La arquitectura permite agregar fácilmente más nodos WordPress o MariaDB sin necesidad de rediseñar el sistema, lo que facilita escalar según las necesidades de tráfico.
4. **Tolerancia a Fallos**:
   Tanto WordPress como MariaDB mantuvieron operación estable ante la caída de uno o más nodos, cumpliendo con los principios de tolerancia a fallos.
5. **Mantenimiento Simplificado con Docker**:
   El uso de contenedores Docker permitió un despliegue rápido y reproducible, facilitando la gestión de servicios y nodos.
6. **Punto Único de Fallo**:
   Aunque HAProxy funcionó correctamente como balanceador, al estar en un solo nodo representa un posible punto único de fallo. Esto puede mitigarse con una solución adicional como Keepalived o una IP flotante.
7. **Mejoras Posibles**:
   - Añadir cachés (como Varnish o Redis) para reducir carga en PHP/MySQL.
   - Usar un servicio de almacenamiento compartido (como NFS o GlusterFS) para compartir medios subidos a WordPress entre los nodos.