

Tarea #996 Realizar el curso de AWS Security, el curso cierra el 2 de Abril

Welcome, DENILSON ALDAHIR

Congratulations on your most recent badge 🎉



AWS Academy Graduate - AWS Academy Cloud Security Foundations

[Amazon Web Services Training and Certification](#)

Share

...

Tarea #995 Realizar el despliegue de Wordpress en HA con un LB con HAProxy y una BD sincronizada en 3 nodos. Entregar 2 de Abril

git clone https://github.com/hweidner/galera-docker.git

A screenshot of a Linux desktop environment showing a terminal window. The terminal is executing the command 'git clone https://github.com/hweidner/galera-docker.git'. The output shows the progress of cloning a repository from GitHub, including object enumeration and receiving data. The background of the desktop features a vibrant, colorful parrot graphic.

```
cd galera-docker
```

```
[user@parrot] ~$ git clone https://github.com/hweidner/galera-docker.git
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 | 8.85 MiB/s, done.
Resolving deltas: 100% (12/12), done.
[user@parrot] ~$ cd galera-docker
[user@parrot] ~/galera-docker$
```

```
docker build -t mycluster/galera .
```

```
[user@parrot] ~$ git clone https://github.com/hweidner/galera-docker.git
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 | 8.85 MiB/s, done.
Resolving deltas: 100% (12/12), done.
[user@parrot] ~$ cd galera-docker
[user@parrot] ~/galera-docker$ docker build -t mycluster/galera .
[*] Building 0.9s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring dockerfile: 296B
=> [internal] MaintainerDeprecation: Maintainer instruction is deprecated in favor of using label (line 2)
=> [internal] load metadata for docker.io/library/mariadb:10.6
=> [internal] load .dockerrcignore
=> transferring context: 2B
=> [internal] FROM docker.io/library/mariadb:10.6@sha256:ec79aa7a81a7667885cb69b6dc0415e032f22520bd5aca77927faffca4329924
=> [internal] load build context
=> transferring context: 568B
=> CACHED [2/4] RUN touch /tmp/.wsrep-new-cluster && chown -R mysql:mysql /tmp/.wsrep-new-cluster
=> CACHED [3/4] COPY galera.cnf /etc/mysql/conf.d/01-galera.cnf
=> CACHED [4/4] COPY startup.sh /startup.sh
=> exporting to image
=> => exporting layers
=> => writing image sha256:b0ff2d501f3bd91327e16e1fb173d50b37c0cf824719864daa21e7783be948
=> => naming to docker.io/mycluster/galera

2 warnings found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 9)
- MaintainerDeprecation: Maintainer instruction is deprecated in favor of using label (line 2)
[user@parrot] ~/galera-docker$
```

```
docker network create --subnet=172.18.0.0/16 galera
```

```
[user@parrot] ~$ docker network create --subnet=172.18.0.0/16 galera
267df4117f3da946ddc49c0de5397cec6a2b3e139108778554abf14b84be5e7
[user@parrot] ~$
```

```
mkdir /srv/galera/node{1,2,3}
```

```
chown 999:999 /srv/galera/node{1,2,3}
```

```
[x] [user@parrot] ~$ sudo chown 999:999 /srv/galera/node{1,2,3}
[user@parrot] ~$
```

```
docker run -d --restart=unless-stopped --net galera \
--name node1 -h node1 --ip 172.18.0.101 \
-p 3311:3306 \
-v /srv/galera/node1.cnf:/etc/mysql/conf.d/galera.cnf \
-v /srv/galera/node1:/var/lib/mysql \
```

```

-e MYSQL_ROOT_PASSWORD=secret_galera_password \
-e GALERA_NEW_CLUSTER=1 \
mycluster/galera
docker run -d --restart=unless-stopped --net galera \
--name node2 -h node2 --ip 172.18.0.102 \
-p 3312:3306 \
-v /srv/galera/node2.cnf:/etc/mysql/conf.d/galera.cnf \
-v /srv/galera/node2:/var/lib/mysql \
-e MYSQL_ALLOW_EMPTY_PASSWORD=1 \
mycluster/galera

docker run -d --restart=unless-stopped --net galera \
--name node3 -h node3 --ip 172.18.0.103 \
-p 3313:3306 \
-v /srv/galera/node3.cnf:/etc/mysql/conf.d/galera.cnf \
-v /srv/galera/node3:/var/lib/mysql \
-e MYSQL_ALLOW_EMPTY_PASSWORD=1 \
mycluster/galera

```

```

mysql> -e MYSQL_ROOT_PASSWORD=secret_galera_password -e GALERA_NEW_CLUSTER=1
[...]-[user@parrot]-[~/galera-docker]
└── $ docker run -d --restart=unless-stopped --net galera \
    --name node1 -h node1 --ip 172.18.0.101 \
    -p 3311:3306 \
    -v /srv/galera/node1.cnf:/etc/mysql/conf.d/galera.cnf \
    -v /srv/galera/node1:/var/lib/mysql \
    -e MYSQL_ROOT_PASSWORD=secret_galera_password \
    -e GALERA_NEW_CLUSTER=1 \
    mycluster/galera
6dce9d6b6eb28b1af1ae8a214144f0402b52a56c890f8144b36e6767b46e79f5
[...]-[user@parrot]-[~/galera-docker]
└── $ docker run -d --restart=unless-stopped --net galera \
    --name node2 -h node2 --ip 172.18.0.102 \
    -p 3312:3306 \
    -v /srv/galera/node2.cnf:/etc/mysql/conf.d/galera.cnf \
    -v /srv/galera/node2:/var/lib/mysql \
    -e MYSQL_ALLOW_EMPTY_PASSWORD=1 \
    mycluster/galera
4793fffc0f3d8d215c9c11bee82a8ce8679fd54c6c01b01f00062ddbae8ef9bc8
[...]-[user@parrot]-[~/galera-docker]
└── $ docker run -d --restart=unless-stopped --net galera \
    --name node3 -h node3 --ip 172.18.0.103 \
    -p 3313:3306 \
    -v /srv/galera/node3.cnf:/etc/mysql/conf.d/galera.cnf \
    -v /srv/galera/node3:/var/lib/mysql \
    -e MYSQL_ALLOW_EMPTY_PASSWORD=1 \
    mycluster/galera
5e21414e8dd31f5ee02d5b2fe8a116573c81638b5641941d0df7dd9062b992c9
[...]-[user@parrot]-[~/galera-docker]
└── $

```

Applications Places System docker ps - Parrot Terminal

```
node3
[ user@parrot:~/galera-docker ]$ docker rm node3
node3
[ user@parrot:~/galera-docker ]$ docker run -d --restart=unless-stopped --net galera \ 
--name node3 -h node3 --ip 172.18.0.103 \ 
-e MYSQL_ALLOW_EMPTY_PASSWORD=1 \
-v /srv/galera/node3.cnf:/etc/mysql/conf.d/galera.cnf \
-v /srv/galera/node3:/var/lib/mysql \
-e MYSQL_ALLOW_EMPTY_PASSWORD=1 \
mycluster/galera fromMySQL clients, e.g.
439d5023eab6f78d7823dd1ac9ecc6107b3098b0425bf8d0fe6b5b482eff3f0
[ user@parrot:~/galera-docker ]$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
439d5023eab6 mycluster/galera "docker-entrypoint.s..." 2 minutes ago Restarting (1) 27 seconds ago node3
4bbbe5e509e8 mycluster/galera "docker-entrypoint.s..." 2 minutes ago Restarting (1) 57 seconds ago node2
b9b5bd283ca3 mycluster/galera "docker-entrypoint.s..." 5 minutes ago Restarting (1) 45 seconds ago node1
c7cdff96c634 mycluster/galera "docker-entrypoint.s..." 2 hours ago Up 2 hours nervous_gagarin
[ user@parrot:~/galera-docker ]$
```

The Dockerfile deals with the fact that, in Galera Cluster, the first node has to be started with a special parameter `--wsrep-new-cluster` (or a script `galera_new_cluster`, which does exactly that).

To achieve this, the Dockerfile adds a temporary file `/tmp/wsrep-new-cluster` to an otherwise unchanged official MariaDB image. This file is deleted during the first invocation of a newly instanciated container. Only if this file exists, and if a non-empty environment variable `GALERA_NEW_CLUSTER` was supplied to the container, the MariaDB server is started with `--wsrep-new-cluster` and creates a new cluster.

Whenever a node is restarted, no new cluster will be restarted, because the file `/tmp/wsrep-new-cluster` is no

Wed Apr 2, 07:21

Menu docker ps - Parrot Terminal GitHub - hweidner/ga... (docker run --rm -it m...