Galera Cluster and Docker Swarm

Von erkan am 26.03.16 1:39 | Keine Kommentare | Keine TrackBacks

This blogpost extends last one. In the last blogpost, we had a look into Docker Network and how it makes the communication between the containers (over multiple hosts) easier. Of course we used Galera for that :)

In this blogpost we are going to use Docker Swarm to bootstrap a Galera Cluster.

Why using Docker Swarm?

Docker Swarm is (simplified) a proxy. So we've got one accesspoint to manage multiple hosts. (The swarm manage service will run on 172.17.152.11:2376). We also use Docker Swarm to abstract from the nodes. As we want the cluster to be running but we don't want to define explicitly where to run them. (Think about a 3-node-cluster on Docker Swarm with 100 nodes.)

Let us point the local docker to Docker Swarm:

```
export DOCKER_HOST=tcp://172.17.152.11:2376
```

We still got the cluster from the last blogpost running:

```
$ docker ps -f name=galera3 -f name=galera2 -f name=galera1
CONTAINER ID        IMAGE                       NAMES
751f4f071359        erkules/galera:basic        swarm3/galera3
24d4a2dfe3e2        erkules/galera:basic        swarm2/galera2
d3410d308171        erkules/galera:basic        swarm1/galera1
```

Docker Swarm extends NAMES. So we see the hosts the containers run on also.

Let's get rid from the old cluster:

```
$ docker rm -f swarm3/galera3 swarm2/galera2 swarm1/galera1
swarm3/galera3
swarm2/galera2
swarm1/galera1
```

We are going to deploy a Galera Cluster. For simplicity we are going to reuse the old overlay-network (named galera).

With Docker Swarm we also change to way we run the containers:

- We don't mention where to run the container
- Every container gets the label galera=setup1
- We tell the container not to run on a node with another container running that label (affinity:...). This is to make sure not two Galera instances are on the same host.

Bootstrapcontainer (galera1):

```
$ docker run -d --name galera1 --net galera  -e affinity:galera!=setup1 \
   --label galera=setup1 erkules/galera:basic \
   --wsrep-cluster-address=gcomm://galera1,galera2,galera3 --wsrep-new-cluster
1c3f8576cb124261c35412c7e643b341ec6f69d70c6a601b7dde8c3574774c42
```

galera2

```
$ docker run -d --name galera2 --net galera  -e affinity:galera!=setup1 \
   --label galera=setup1 erkules/galera:basic \
   --wsrep-cluster-address=gcomm://galera1,galera2,galera3
611501de09b64475e9356dfb50be7f5bf179919a9e94e60b9d1e466bb7450437
```

galera3

```
$ docker run -d --name galera3 --net galera  -e affinity:galera!=setup1 \
   --label galera=setup1 erkules/galera:basic \
   --wsrep-cluster-address=gcomm://galera1,galera2,galera3
582aaf272bb449733ca1b95cced1ae7b3ef2e20e105d09261a36b6a0912d9f07
```

So let's check if everything went fine:

```
$ docker ps   -f  label=galera=setup1
CONTAINER ID        IMAGE                   NAMES
582aaf272bb4        erkules/galera:basic    swarm2/galera3
611501de09b6        erkules/galera:basic    swarm1/galera2
1c3f8576cb12        erkules/galera:basic    swarm3/galera1


$ docker exec  swarm1/galera2 \
   mysql -e 'show global status like "wsrep_cluster_size"'
Variable_name Value
wsrep_cluster_size 3
```

Jihaa!!!!

What happens when we start a third Galera container?

```
$ docker run -d --name galera4 --net galera  -e affinity:galera!=setup1 \
  --label galera=setup1 erkules/galera:basic \
  --wsrep-cluster-address=gcomm://galera1,galera2,galera3
docker: Error response from daemon: unable to find a node that satisfies galera!=setup1.
See 'docker run --help'
```

It failed! Very good. Having only three Docker nodes there was no machine left to start a forth Galera container.

For Planetmysql: MySQL *g*

Viel Spaß

Erkan

Kategorien: Docker, Galera, mysql

Keine TrackBacks

TrackBack-URL: http://linsenraum.de/mt/mt-tb.cgi/388

Jetzt kommentieren

Anmelden um zu kommentieren