使用 Docker 建立 Mysql 集群

软件环境介绍

操作系统：Ubuntu server 64bit 14.04.1
Docker 版本 1.6.2
数据库：Mariadb 10.10 （Mariadb 是 MySQL 之父在 MySQL 被 Oracle 收购之后创建的分支，性能上优于 MySQL 开源版本）

第一步 安装 Docker

对于 Ubuntu，建议直接联网安装 Docker 最新版本，apt-get 中版本较老。
首先获取安装脚本：
`wget https://get.docker.com`
下下来的虽然名字是 index.html，但其实是脚本文件，所以我们
`chmod +x index.html`
这样我们就可以执行这个文件：
`sudo ./index.html`
安装完成后根据提示，可以将当前用户加到 docker 用户组里，这样就不用每次执行 docker 都需要 sudo 了。
`sudo usermod -aG docker <你的用户名>`

对于 Centos6,首先要把企业常用软件包的软件源安装上
`yum install epel-release`
然后再
`yum install docker-io`

第二步 运行 Mariadb 容器

首先要将数据镜像拉下来
`docker pull mariadb:latest`
注意，如果不加:latest 标签，docker 会把所有的镜像版本都拉下来。

然后我们就可以启动镜像了，参数方面需要注意的有一下几点：
1，-name <给容器取个好记的名称>
2，-e MYSQL_ROOT_PASSWORD = '<给数据库一个 root 用户密码>'
3，-p <映射到本机的端口>:3306
4，-v <本机的数据库存放目录>:/var/lib/mysql

5，设定 MYSQL_USER、MYSQL_PASSWORD、MYSQL_DATABASE 环境变量可以使容器在运行时同时创建你所需要的数据库和带有全部权限的用户及其对应密码
6，设定 TERM 环境变量的值可以解决容器不能进入 mysql 控制台的问题。
对于不是自己建立的镜像，建立出来的容器未必能一次达到要求，建议是将 run 命令写成脚本，创建后使用
`docker inspect <容器名>`
仔细查看容器信息，关注镜像公开的端口和文件目录。如果发现达不到要求，使用
`docker rm -f <容器名>`
删除容器后修改 run 脚本再次运行，直到满意为止。
下面是我的 run 命令：

```
docker run --name mdb1 \
-p 13306:3306 \
-v /home/wonders/docker_mdb1_data:/var/lib/mysql \
-e MYSQL_ROOT_PASSWORD=wondersgroup \
-e MYSQL_USER=medical_waste \
-e MYSQL_PASSWORD=medical_waste \
-e MYSQL_DATABASE=medical_waste \
-e TERM=linux \
-d mariadb
```

第三步 配置一主一从集群

3.1
接下来启动另一个容器作为从数据库，因为镜像不支持在容器内进入 mysql 控制台，所以依然需要把端口暴露出来以供局域网访问,但主数据库容易可以链接进来作为一个可访问的主机 master_db。
`docker run --name <从数据库名> -e MYSQL_ROOT_PASSWORD=<从数据库 root 密码> --link <主数据库容器名>:master_db -d mariadb`
3.2
接下来就需要配置两个数据库了，前提工作是镜像中并没有自带 vi，所以在两个容器内都需要:
`apt-get update`
`apt-get install vim`
这样我们才能在容器内修改配置文件。
还有一种方法，就是我们在主数据库容器中操作，之后，使用
`docker commit <主数据库容器名> mariadb`
这样再创建的容器就包含 vi 了。
3.3
为讲述方便，现在假设：我们有了连个数据库，mdb1 和 mdb2, mdb1 我们作为主数据库，mdb2 作为从数据库。
首先修改主数据库：
`docker exec -it mdb1 /bin/bash`

进入主数据库容器内之后，

```
vi /etc/mysql/my.cnf
```

把“server-id = 1”行的注释去掉即可，保存，退出容器，然后

```
docker restart mdb1
```

同样的，把从数据库的 my.cnf 修改“server-id = 2”，需要是比主 server-id 大的数字，mdb2 同样需要重启。

3.4

使用客户端连接上主数据库，这里我使用的是 mysql workbench，从数据库因为安全考虑并没有公开端口给主机，只能进入容器的 mysql 控制台进行操作。

在主数据库中执行 SQL 脚本：

```
/*设定用于同步的账号、密码*/
grant replication slave on *.* to ‘sync’@’%’ identified by ’sync’;
/*保存权限设定*/
flush privileges;
/*查看主数据日志状态，需要记住查询结果 File 和 Position 值，是从数据库复制的日志起点*/
show master status;
```

在从数据库中执行 SQL 脚本：

```
/*如果已经开启了同步，停止同步*/
stop slave;
/*设定主数据库*/
change master to
master_host=’master_db’,
master_user=’sync’,
master_password=’sync’,
master_port=3306,
master_log_file=’<主数据库查询得到的 File 值>’,
master_log_pos=<主数据库查询得到的 Positions 值>;
```

下面是我的脚本例子：

```
change master to
master_host=’master_db’,
master_user=’sync’,
master_password=’sync’,
master_port=3306,
master_log_file=’mariadb-bin.000004’,
master_log_pos=789;
/*开启从数据库复制*/
start slave;
```

最后可以通过

```
show slave status;
```

查看同步情况。

至此我们就建立了一个基于 Docker 的 Mariadb 数据库集群。