# **使用docker快速搭建kafka**

1、kafka需要zookeeper管理，所以需要先安装zookeeper。

下载zookeeper镜像

$ docker pull wurstmeister/zookeeper

2、启动镜像生成容器

## docker run -d --name zookeeper -p 2181:2181 -v /etc/localtime:/etc/localtime wurstmeister/zookeeper

$ docker run -d --restart=always --log-driver json-file --log-opt max-size=100m --log-opt max-file=2  --name zookeeper -p 2181:2181 -v /etc/localtime:/etc/localtime wurstmeister/zookeeper

3、下载kafka镜像

$ docker pull wurstmeister/kafka

4、启动kafka镜像生成容器

## docker run -d --name kafka -p 9092:9092 -e KAFKA\_BROKER\_ID=0 -e KAFKA\_ZOOKEEPER\_CONNECT=172.16.0.13:2181/kafka -e KAFKA\_ADVERTISED\_LISTENERS=PLAINTEXT://172.16.0.13:9092 -e KAFKA\_LISTENERS=PLAINTEXT://0.0.0.0:9092 -v /etc/localtime:/etc/localtime wurstmeister/kafka

$ docker run -d --restart=always --log-driver json-file --log-opt max-size=100m --log-opt max-file=2 --name kafka -p 9092:9092 -e KAFKA\_BROKER\_ID=0 -e KAFKA\_ZOOKEEPER\_CONNECT=172.16.0.13:2181/kafka -e KAFKA\_ADVERTISED\_LISTENERS=PLAINTEXT://172.16.0.13:9092 -e KAFKA\_LISTENERS=PLAINTEXT://0.0.0.0:9092 -v /etc/localtime:/etc/localtime wurstmeister/kafka

参数说明：

-e KAFKA\_BROKER\_ID=0  在kafka集群中，每个kafka都有一个BROKER\_ID来区分自己

-e KAFKA\_ZOOKEEPER\_CONNECT=172.16.0.13:2181/kafka 配置zookeeper管理kafka的路径172.16.0.13:2181/kafka

****-e KAFKA\_ADVERTISED\_LISTENERS=PLAINTEXT://172.16.0.13:9092  把kafka的地址端口注册给zookeeper，如果是远程访问要改成外网IP,类如Java程序访问出现无法连接。****

-e KAFKA\_LISTENERS=PLAINTEXT://0.0.0.0:9092 配置kafka的监听端口

-v /etc/localtime:/etc/localtime 容器时间同步虚拟机的时间

5、验证kafka是否可以使用

5.1、进入容器

$ docker exec -it kafka bash

5.2、进入 /opt/kafka\_2.12-2.3.0/bin/ 目录下

$ cd /opt/kafka\_2.12-2.3.0/bin/

5.3、运行kafka生产者发送消息

$ ./kafka-console-producer.sh --broker-list localhost:9092 --topic sun

发送消息

> {"datas":[{"channel":"","metric":"temperature","producer":"ijinus","sn":"IJA0101-00002245","time":"1543207156000","value":"80"}],"ver":"1.0"}

5.4、运行kafka消费者接收消息

$ ./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic sun --from-beginning

原文地址https://blog.csdn.net/cyxinda/article/details/105774384?spm=1001.2101.3001.6650.2&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2-105774384-blog-126405864.pc\_relevant\_3mothn\_strategy\_recovery&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2-105774384-blog-126405864.pc\_relevant\_3mothn\_strategy\_recovery&utm\_relevant\_index=3