Docker+kafka+springboot

前言:我们暂时先搭建单机的zookeeper和kafka，后面会持续更新伪分布式集群构建；网上很多资料都残缺不全，对于新手来说是一件非常痛苦事，我们直接步入正题吧

# 1拉取zookeeper和kafka镜像

我拉取的是wurstmeister/kafka（维护较多）和zookeeper这两个镜像；当然也可以自己下载客户端打包成docker 镜像，但是我个人觉得自己打包意义不大。大家跟紧我的脚步，别

docker pull wurstmeister/kafka

docker pull zookeeper

拉取成功后查看镜像

docker images



# 2.运行镜像

## 2.1先运行zookeeper

docker run --name zk -p 2181:2181 -d your-docker-id

your-docker-id是docker images 下获取到的本地docker 镜像id,zk通常使用的是2181端口

zookeeper启动起来后

## 2.2运行kafka

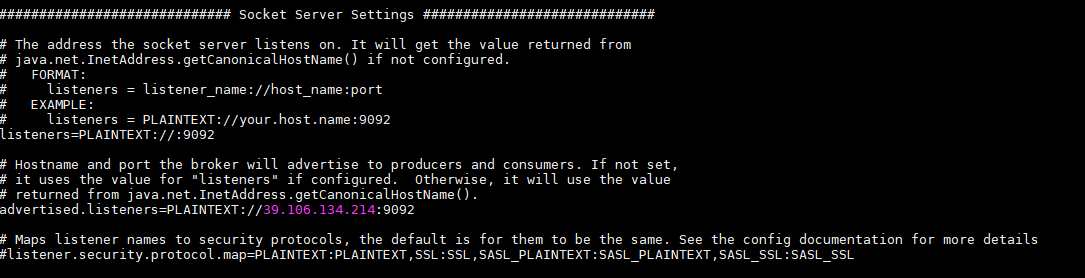
由于我的是阿里云主机，需要在kafka实现注册时将公网ip暴露给zookeeper，我这里直接更该镜像kafka的配置

find / -name server.properties

得到如下



修改如下



主要添加两行

listeners=PLAINTEXT://:9092

advertised.listeners=PLAINTEXT://公网ip:9092

再运行kafka

docker run

--name kafka -p 9092:9092

-e KAFKA\_ADVERTISED\_HOST\_NAME=172.17.120.202

-e KAFKA\_CREATE\_TOPICS="test:1:1"

-e KAFKA\_ZOOKEEPER\_CONNECT=172.17.120.202:2181/kafka

-d your-docker-id

-e是制定运行配置环境

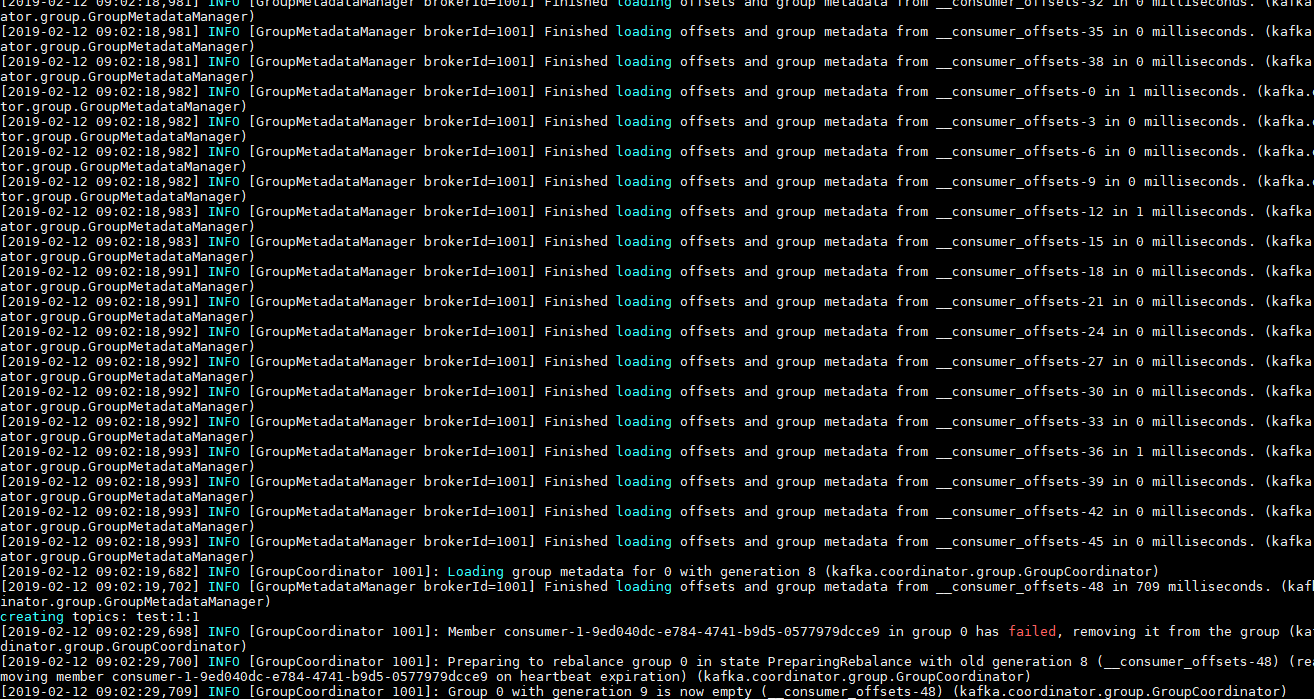
KAFKA\_ADVERTISED\_HOST\_NAME是内网ip

KAFKA\_CREATE\_TOPICS 是创建一个topic为test的队列test将会有1个分区和1个副本

KAFKA\_ZOOKEEPER\_CONNECT是我们的zk地址，后面/Kafka是我们制定的根路径，这个必须要指定，命名随意

通过日志我们可以看到kafka已经正常启动并且创建了topic-test的队列

docker logs -f kafka



# 3 springboot整合

## 3.1添加maven

这里注意maven-kafka版本要与你的kafka版本对应下，我的kafka版本比较新，使用的最新版

可以进入kafka容器，在opt/kafka/lib目录下，看到



这种文件名，前面的2.12是语言版本，后面的2.1.0是kafka版本

*<!--kafka-->*<**dependency**>  
 <**groupId**>org.springframework.kafka</**groupId**>  
 <**artifactId**>spring-kafka</**artifactId**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>com.fasterxml.jackson.core</**groupId**>  
 <**artifactId**>jackson-databind</**artifactId**>  
 <**version**>2.9.8</**version**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>com.alibaba</**groupId**>  
 <**artifactId**>fastjson</**artifactId**>  
 <**version**>1.2.47</**version**>  
</**dependency**>

## 3.2添加配置文件

**server.port**=**8080  
logging.level.root**=**info***#============== kafka ===================  
# 指定kafka 代理地址，可以多个***spring.kafka.bootstrap-servers**=**39.106.134.214:9092***#=============== provider =======================***spring.kafka.producer.retries**=**1***# 每次批量发送消息的数量***spring.kafka.producer.batch-size**=**16384  
spring.kafka.producer.buffer-memory**=**33554432***# 指定消息key和消息体的编解码方式***spring.kafka.producer.key-serializer**=**org.apache.kafka.common.serialization.StringSerializer  
spring.kafka.producer.value-serializer**=**org.apache.kafka.common.serialization.StringSerializer***#=============== consumer =======================  
# 指定默认消费者group id***spring.kafka.consumer.group-id**=**0  
  
spring.kafka.consumer.auto-offset-reset**=**earliest  
spring.kafka.consumer.enable-auto-commit**=**true  
spring.kafka.consumer.auto-commit-interval**=**100***# 指定消息key和消息体的编解码方式***spring.kafka.consumer.key-deserializer**=**org.apache.kafka.common.serialization.StringDeserializer  
spring.kafka.consumer.value-deserializer**=**org.apache.kafka.common.serialization.StringDeserializer**

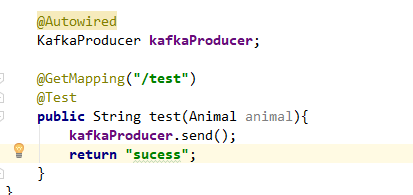
## 3.3创建测试生产者

@Component  
**public class** KafkaProducer {  
 @Autowired  
 **private** KafkaTemplate **kafkaTemplate**;  
  
 **public void** send(){  
 **for** (**int** i = 0 ;i < 10 ; i++){  
 Message message = **new** Message(i);  
 System.***out***.println(**"发送第"** + i +**"消息"**);  
 **kafkaTemplate**.send(**"test"**,JSONObject.*toJSONString*(message));  
 }  
 }  
}

## 3.4创建测试消费者

@Component  
**public class** KafkaTestComsumer {  
  
 @Autowired  
 **private** KafkaTemplate **kafkaTemplate**;  
  
 @KafkaListener(topics = {**"test"**})  
 **public void** listen(ConsumerRecord<?,?> consumerRecord){  
 Optional<?> kafkaMessage = Optional.*ofNullable*(consumerRecord.value());  
 **if**(kafkaMessage.isPresent()){  
 *//得到Optional实例中的值* Object message = kafkaMessage.get();  
 System.***err***.println(**"消费消息:"**+message);  
 }  
 }  
}

## 3.5测试代码及测试结果







完美结束。

后言：我这个2.1.0是目前非常新的kafka版本，自带了zookeeper，不过我们通常有自己的zk，所以我这里顺便弄了个zk，如果还有什么问题我们可以一起讨论，附一个apache-kafka地址-http://kafka.apache.org/quickstart