安装KAFKA

* 1. 部署配置

首先配置好zookeeper，略。

mkdir /usr/local/kafka

cp kafka\_2.11-0.10.1.1.tgz /usr/local/kafka/

tar zxvf kafka\_2.11-0.10.1.1.tgz

cd kafka\_2.11-0.10.1.1

vim config/server.properties

broker.id=1 #当前机器在集群中的唯一标识，和zookeeper的myid性质一样,

每一个broker在集群中的唯一表示，要求是正数。当该服务器的IP地址发生改变时，broker.id没有变化，则不会影响consumers的消息情况

port=9092 #当前kafka对外提供服务的端口默认是9092

host.name=192.168.1.172 #broker的主机地址，若是设置了，那么会绑定到这个地址上，若是没有，会绑定到所有的接口上，并将其中之一发送到ZK，一般不设置

num.network.threads=3 #broker处理消息的最大线程数，一般情况下数量为cpu核数

num.io.threads=8 #broker处理磁盘IO的线程数，数值为cpu核数2倍

socket.send.buffer.bytes=102400 #发送缓冲区buffer大小，数据不是一下子就发送的，先回存储到缓冲区了到达一定的大小后在发送，能提高性能

socket.receive.buffer.bytes=102400 #kafka接收缓冲区大小，当数据到达一定大小后在序列化到磁盘

socket.request.max.bytes=104857600 #这个参数是向kafka请求消息或者向kafka发送消息的请请求的最大数，这个值不能超过java的堆栈大小

log.dirs=/tmp/kafka-logs\_1 #kafka数据的存放地址，多个地址的话用逗号分割,多个目录分布在不同磁盘上可以提高读写性能 /data/kafka-logs-1，/data/kafka-logs-2

num.partitions=3 #每个topic的分区个数，若是在topic创建时候没有指定的话会被topic创建时的指定参数覆盖

num.recovery.threads.per.data.dir=1 #用于在启动时,用于日志恢复的线程个数,默认是1.

log.retention.hours=168 #默认消息的最大持久化时间，168小时，7天

log.segment.bytes=1073741824 #topic的分区是以一堆segment文件存储的，这个控制每个segment的大小，会被topic创建时的指定参数覆盖

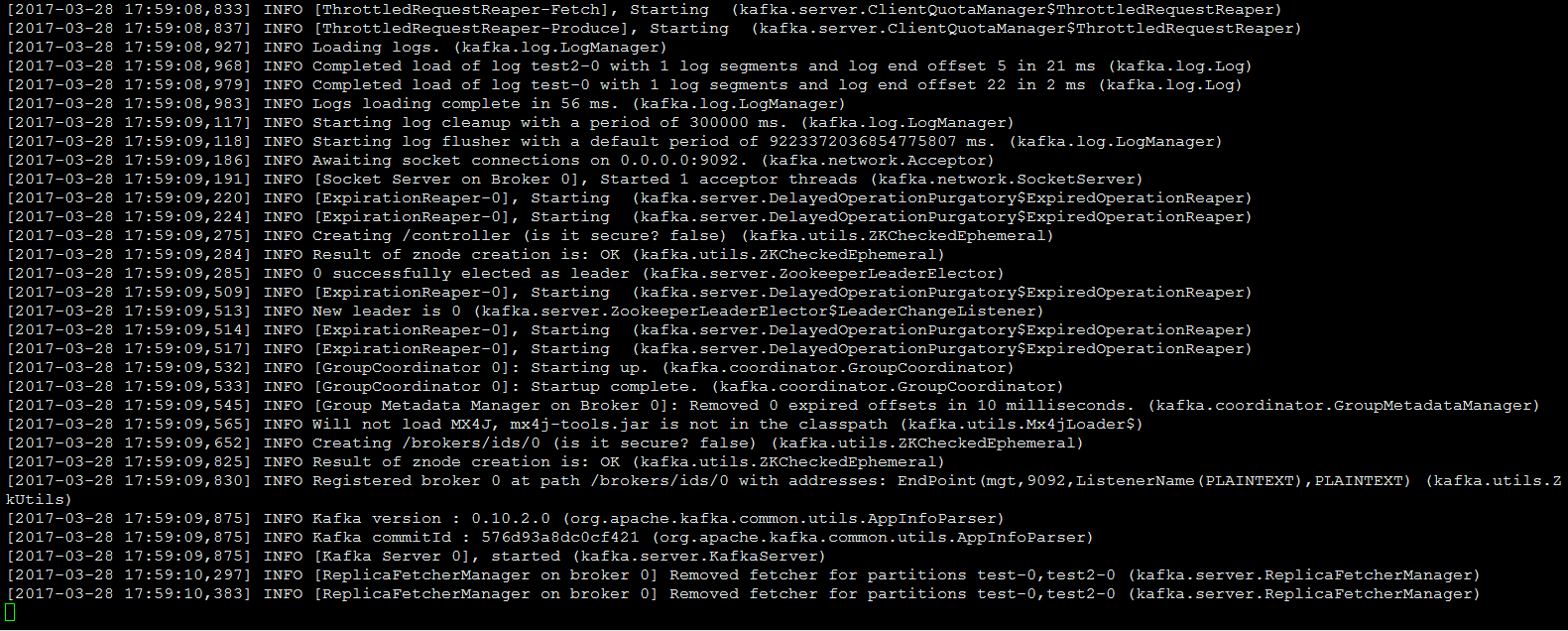
log.retention.check.interval.ms=300000 #每隔300000毫秒去检查上面配置的log失效时间（log.retention.hours=168 ），到目录查看是否有过期的消息如果有，删除

zookeeper.connect=localhost:2181,localhost:2182,localhost:2183

zookeeper.connection.timeout.ms=6000 #ZooKeeper的连接超时时间

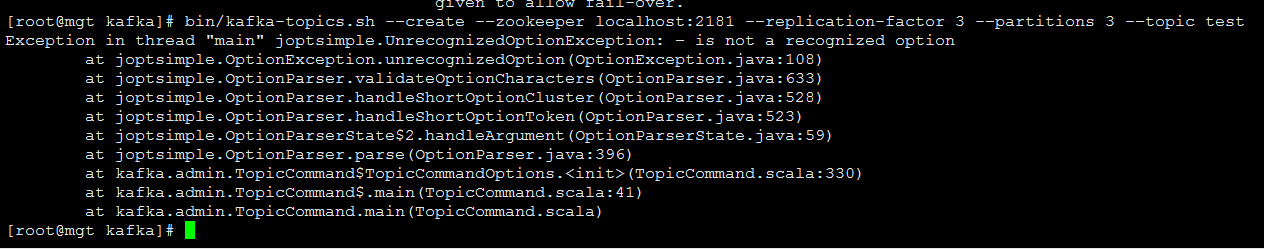
1.2 启动服务器

bin/kafka-server-start.sh config/server.properties &

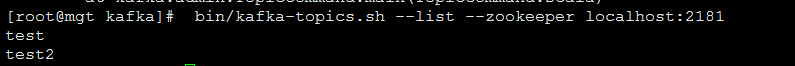


1.3 kafka测试

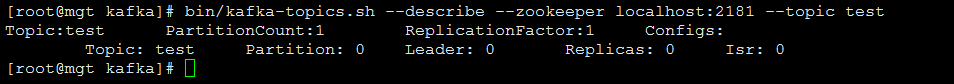
产生topic,3个分片，3个副本   
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 3 --partitions 3 --topic test



[root@paasagento kafka\_1]# bin/kafka-topics.sh --list --zookeeper localhost:2181



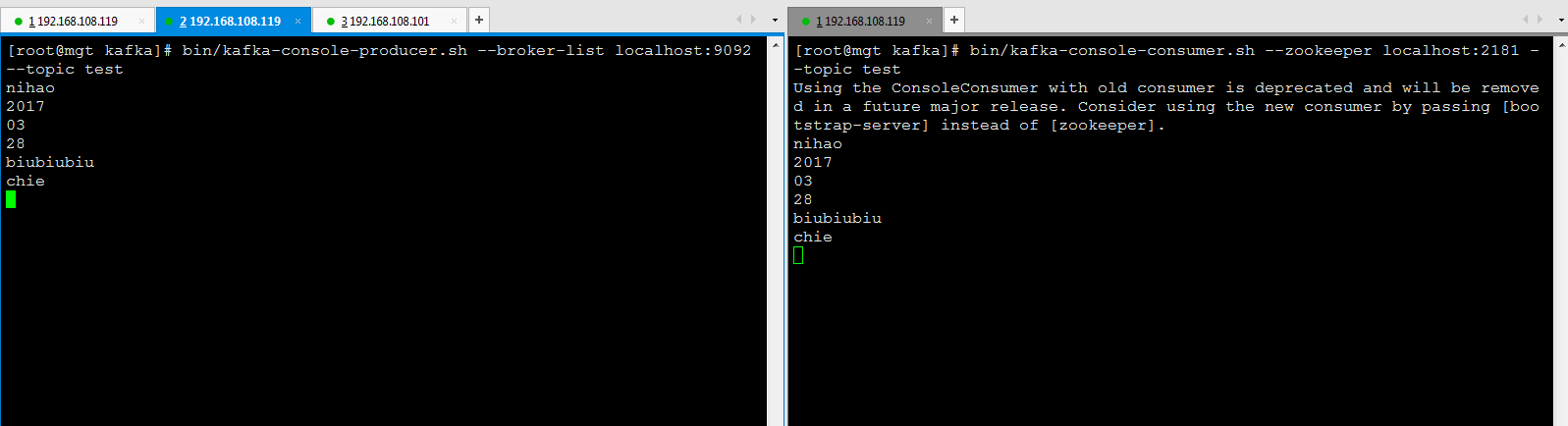
[root@paasagento kafka\_1]# bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic test



发布消息

bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test   
消费消息

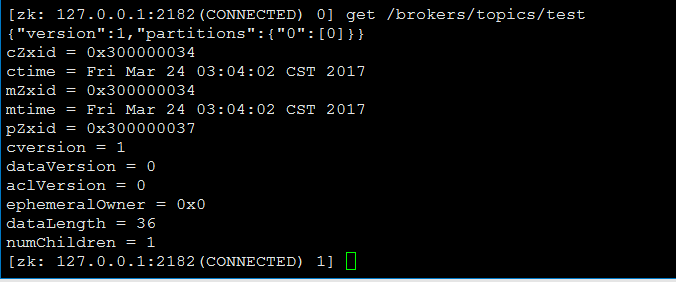
bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic test



zookeeper查看topic   
bin/zkCli.sh -server 127.0.01:2182

[zk: 192.168.1.172:2182(CONNECTED) 2] get /brokers/topics/test\_topic

{"version":1,"partitions":{"2":[3,1,2],"1":[2,3,1],"0":[1,2,3]}}



1.4 日志说明

server.log   
kafka的运行日志   
state-change.log   
kafka他是用zookeeper来保存状态，所以他可能会进行切换，切换的日志就保存在这里   
controller.log   
kafka选择一个节点作为“controller”,当发现有节点down掉的时候它负责在分区的所有节点中选择新的leader,这使得Kafka可以批量的高效的管理所有分区节点的主从关系。如果controller down掉了，活着的节点中的一个会备切换为新的controller.

1.5 小结

1.produce启动的时候参数使用的是kafka的端口而consumer启动的时候使用的是zookeeper的端口；   
2.必须先创建topic才能使用；   
3.topic本质是以文件的形式储存在zookeeper上的。