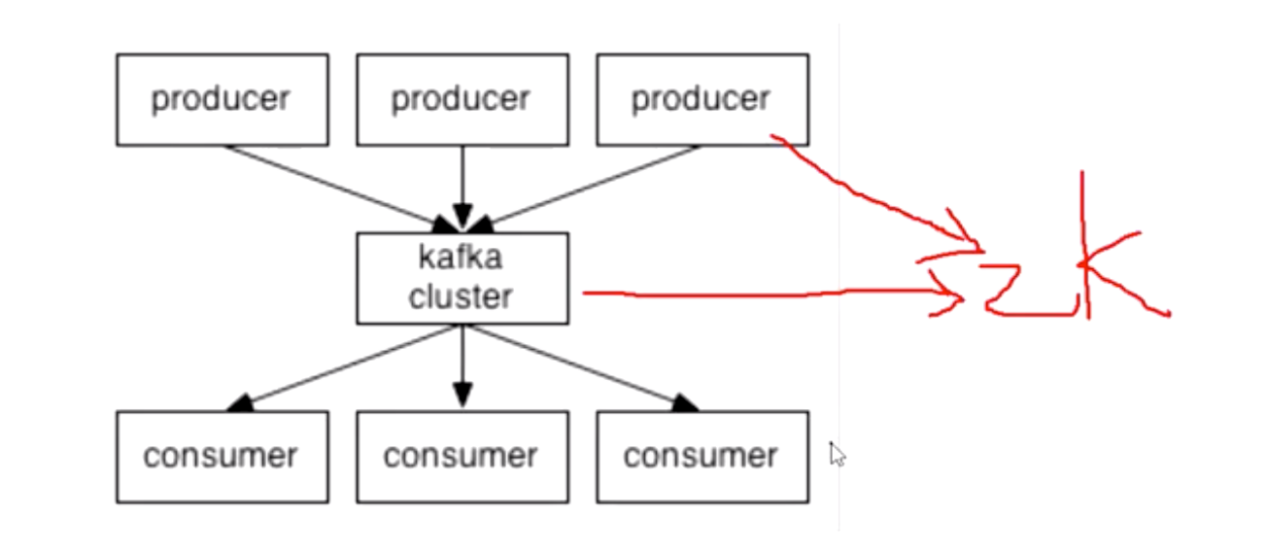
# Kafka

## Base

Replication factor备份因子.



Kafka cluster中的单个节点叫做broker.

kafka很多东西都是依赖zookeeper的.

每一个topic下面多有个partition, partition就是一个message的队列,每一个消息在队列中的序号叫做offset,对于consumer的offset是维护在zookeeper中的.

有的消息队列收到consumer的正确的ack之后就会把msg删了,然后再从队列里就能拿到下一个了.而kafka不是这样的,他是为每个consumer都维护了一个你消费到哪里的一个offset的记录. 这个记录就是写在zookeeper里面的.

默认消息是7天消息才会被清掉.

一个topic下面设置了多个分区, 可以提高消费者的并发量,如果只有一个分区,那么同时只能有一个consumer去消费,如果你想扩大消费的并发量,你去扩几个partination就行了.

一个消费者从头到尾永远只会访问同一个partition,才能offset维护对.

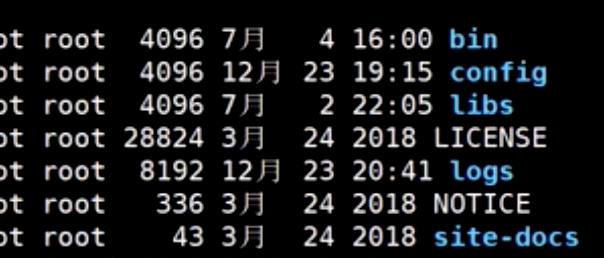
kafka能能够保证消息消费的顺序性,这个是通过个topic多个partination,然后每个consumer固定从一个partination接收消息来保证的.

如果你只有一个queue,abc三个消息,被三个consumer拿走,到底abc哪个先被执行完是没办法保证的,但是如果向kafka这样可以保证abc三个消息在所有的consumer中都是这个顺序执行的.

还有个要求就是有依赖性的消息你得都发到一个partination里面.

kafka原生就带了zookeeper.

kafka是scala语言写的,需要安装jdk.



安装目录这个德行,bin里面有很多的.sh的脚本,用来启动kafka和自带的zookeeper.

config里面有kafka和zookeeper的配置,其中的server.conf文件中有个broker.id的配置,这个配置的是钙kafka在集群中的标志id.

默认端口9092

屏幕快照 2020-03-06 下午6.20.52

这是创建topic的命令,可以看到kafka的操作人家都封好了脚本了,不是那种命令行,不同的操作还封了不同的脚本,不错.

可以看到命令中指定了用的哪个zookeeper, 多个个partination, topic的name.

屏幕快照 2020-03-06 下午6.31.17

列出所有主题,可以看到,这还有一个专门操作topic的脚本.

屏幕快照 2020-03-06 下午6.37.38

发消息的脚本 等等这种脚本.

消费消息的脚本可以指定哪个topic哪个partition的哪个offset的消息.

搭集群的时候就把配置的brokerid改了,注册到同一个zookeeper上,启动,集群自己就组建好了.

创建一个备份因子为3的topic

屏幕快照 2020-03-06 下午8.53.47

kafka的的集群也是有leader和follower的,但是不像mysql的主从,主就永远是主,从就永远是从,对不同的partination来说,leader是不一样的,也就是说,同一个,同一个topic的partination的leader都可能不同. leader是细分到partination上的,为什么这样呢,如果你一个topic下的partination都分配到同一个leader上,那么这个机器挂了,所有的partination都用不了了,都需要等重新选举之后才能用,分开的话只会影响一个.

对于一个topic,producer在发消息的时候只会发到leader,leader再同步消息到follower,类似于消息又消费了一遍.

也就是你这个consumer和这个producer,搞了这个topic,那么你会被分配到一个partination上, 然后而且你从头到位都会跟这个partination的leader producer交互,其他的follower不会有任何作用,只是备份.

## Tips

不要用一个consumer消费多个topic,为啥,因为如果一个topic的消费出问题了,也会影响另一个.