

# Broker架构设计

---

Broker

Topic

Partition

Segment

Broker集群

leader位置如何确定？

分区的Replica副本机制

## Broker

消息处理的节点。

可以理解为每个broker是一个单独的kafka进程，一般部署在不同的机器上，多个broker共同组成一个集群。

## Topic

消息主题。

每一条消息都必须指定主题。[kafka](#)集群可以同时进行多个topic的分发。

## Partition

分区。是topic在物理上的分组。每个partition实际上就是一个目录。

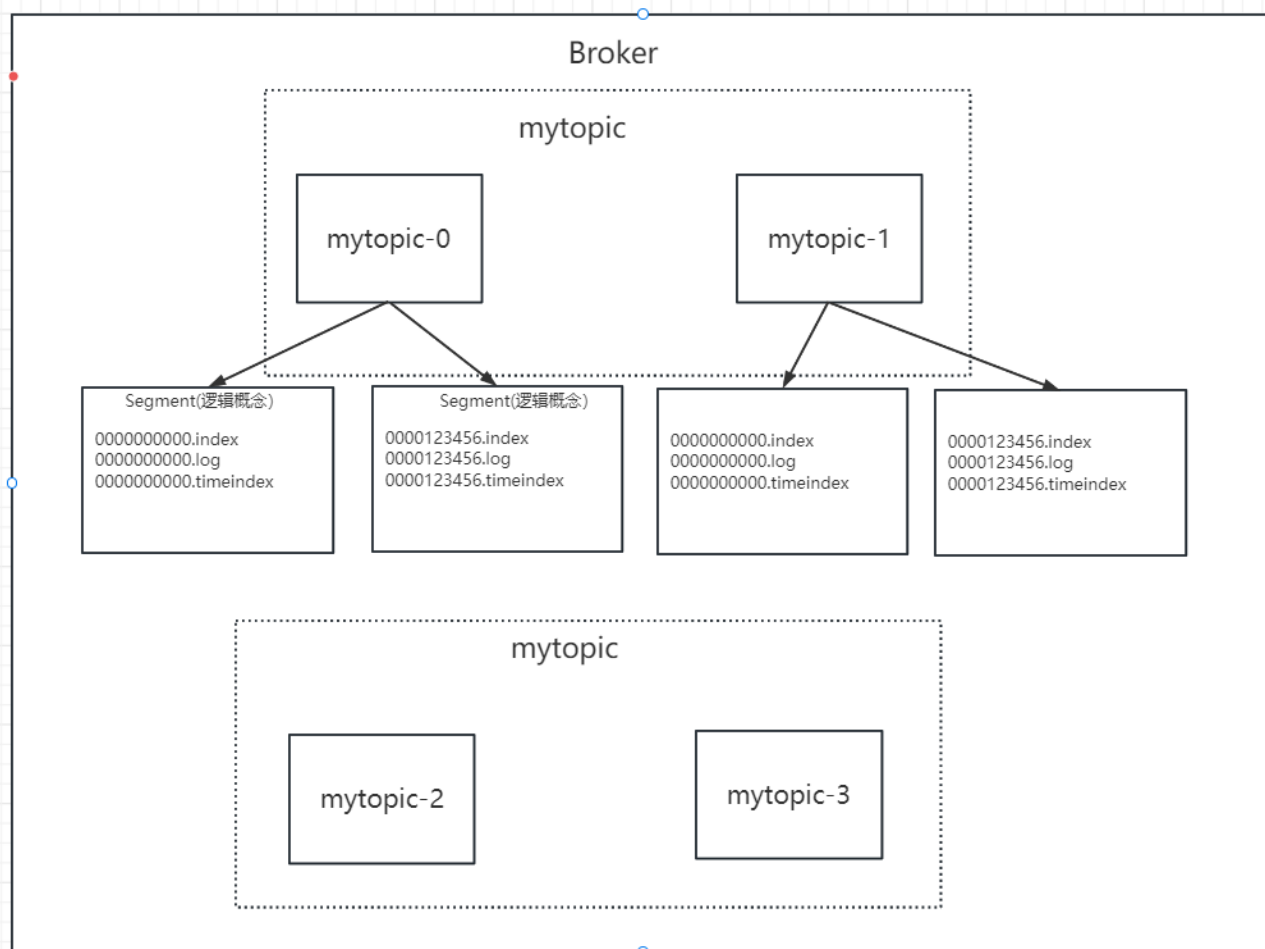
对于同一个topic下的多条消息，按照一定的规则存储在多个partition上。对于每个partition而言，消息的存储是有序的，但是对于整个topic而言，消息的存储是无序的。

## Segment

Kafka的数据是放在后缀.log的文件中的，如果一个Partition只有一个log文件，消息不断地追加，这个log文件也会变

得越来越大，这个时候要检索数据效率就很低了；所以干脆把Partition再做一个切分，切分出来的单位就叫做段

[Segment]，实际上Kafka的存储文件就是划分成段来存储的。



.index 索引文件、.log 数据文件、.timeindex 时间索引文件

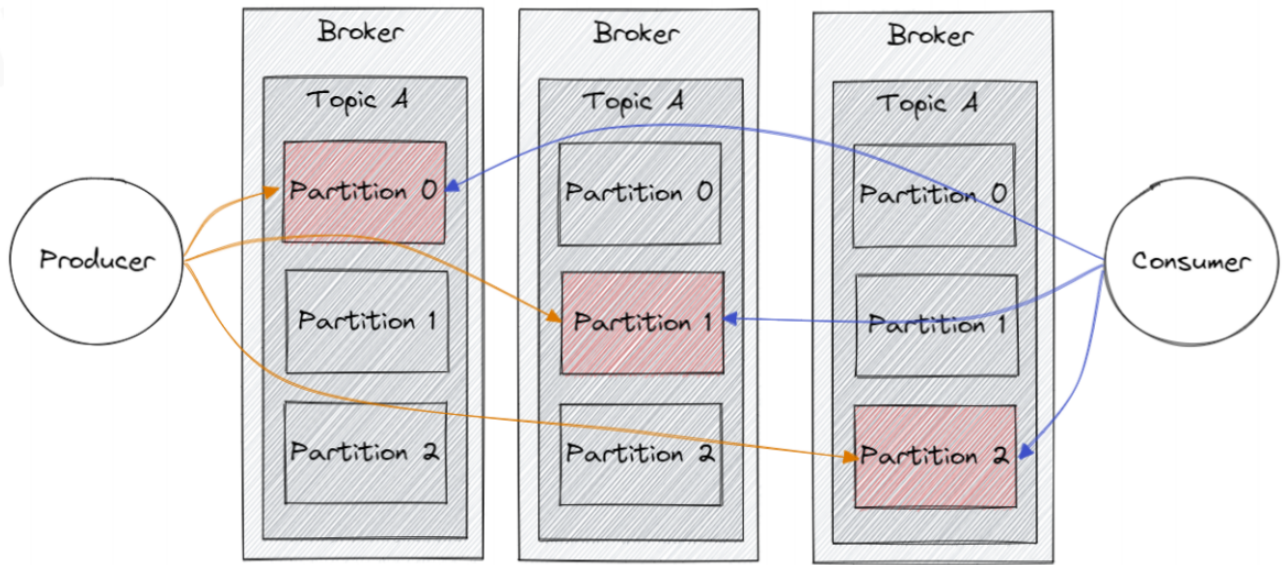
Segment默认大小是1073741824 bytes (1G) ， 由这个参数控制：[log.segment.bytes](#)

▼ 查看.log文件的命令

Java | 复制代码

```
1 sh kafka-run-class.sh kafka.tools.DumpLogSegments --files /tmp/kafka-logs/test3p1r-2/00000000000000000000.log --print-data-log
```

# Broker集群



## leader位置如何确定？

比如Partition0，先确定Partition0 leader在 3个节点中的其中一个，现在是在第一个，然后再确定Partition1 leader，broker节点加一，Partition2 leader，broker节点再加一。

所以他们的leader分别分布在第一 二 三个broker。从节点是没有规律的 一般从节点不会和leader在同一个broker 不然没有意义

副本数不能大于broker数，否则会报错

## 分区的Replica副本机制

如果Partition的数据只存储一份，在发生网络或者硬件故障的时候，该分区的数据就无法访问或者无法恢复了，

Kafka在0.8的版本之后增加了副本机制，每个Partition可以有若干个Replica[副本]，副本必须要在不同的Broker上面，

一般我们说的副本包括其中的主节点，由replication-factor指定一个Topic的副本数：

▼ 3分区3副本命令

Java | [复制代码](#)

```
1 sh kafka-topics.sh --create --topic test3p3r --bootstrap-server 192.168.8.14:9092 --replication-factor 3 --partitions 3
```

API中也有参数可以控制默认副本数：offsets.topic.replication.factor

如上图，有3个Broker节点，一个Topic有3个分区，一个分区有3个副本，红色的是leader。

注意：副本数不能大于broker数，否则会报错