**Boker配置参数**

下表列出了Boker的重要的配置参数， 更多的配置请参考 [kafka.server.KafkaConfig](https://github.com/apache/kafka/blob/0.8.2/core/src/main/scala/kafka/server/KafkaConfig.scala" \t "_blank)

| **name** | **默认值** | **描述** |
| --- | --- | --- |
| brokerid | none | 每一个boker都有一个唯一的id作为它们的名字。 这就允许boker切换到别的主机/端口上， consumer依然知道 |
| enable.zookeeper | true | 允许注册到zookeeper |
| log.flush.interval.messages | Long.MaxValue | 在数据被写入到硬盘和消费者可用前最大累积的消息的数量 |
| log.flush.interval.ms | Long.MaxValue | 在数据被写入到硬盘前的最大时间 |
| log.flush.scheduler.interval.ms | Long.MaxValue | 检查数据是否要写入到硬盘的时间间隔。 |
| log.retention.hours | 168 | 控制一个log保留多长个小时 |
| log.retention.bytes | -1 | 控制log文件最大尺寸 |
| log.cleaner.enable | false | 是否log cleaning |
| log.cleanup.policy | delete | delete还是compat. 其它控制参数还包括log.cleaner.threads，log.cleaner.io.max.bytes.per.second，log.cleaner.dedupe.buffer.size，log.cleaner.io.buffer.size，log.cleaner.io.buffer.load.factor，log.cleaner.backoff.ms，log.cleaner.min.cleanable.ratio，log.cleaner.delete.retention.ms |
| log.dir | /tmp/kafka-logs | 指定log文件的根目录 |
| log.segment.bytes | 1*1024*1024\*1024 | 单一的log segment文件大小 |
| log.roll.hours | 24 \* 7 | 开始一个新的log文件片段的最大时间 |
| message.max.bytes | 1000000 + MessageSet.LogOverhead | 一个socket 请求的最大字节数 |
| num.network.threads | 3 | 处理网络请求的线程数 |
| num.io.threads | 8 | 处理IO的线程数 |
| background.threads | 10 | 后台线程序 |
| num.partitions | 1 | 默认分区数 |
| socket.send.buffer.bytes | 102400 | socket SO\_SNDBUFF参数 |
| socket.receive.buffer.bytes | 102400 | socket SO\_RCVBUFF参数 |
| zookeeper.connect | localhost:2182/kafka | 指定zookeeper连接字符串， 格式如hostname:port/chroot。chroot是一个namespace |
| zookeeper.connection.timeout.ms | 6000 | 指定客户端连接zookeeper的最大超时时间 |
| zookeeper.session.timeout.ms | 6000 | 连接zk的session超时时间 |
| zookeeper.sync.time.ms | 2000 | zk follower落后于zk leader的最长时间 |

**High-level Consumer配置参数**

下表列出了high-level consumer的重要的配置参数。  
更多的配置请参考 [kafka.consumer.ConsumerConfig](https://github.com/apache/kafka/blob/0.8.2/core/src/main/scala/kafka/consumer/ConsumerConfig.scala" \t "_blank)

| **name** | **默认值** | **描述** |
| --- | --- | --- |
| groupid | groupid | 一个字符串用来指示一组consumer所在的组 |
| socket.timeout.ms | 30000 | socket超时时间 |
| socket.buffersize | 64\*1024 | socket receive buffer |
| fetch.size | 300 \* 1024 | 控制在一个请求中获取的消息的字节数。 这个参数在0.8.x中由fetch.message.max.bytes,fetch.min.bytes取代 |
| backoff.increment.ms | 1000 | 这个参数避免在没有新数据的情况下重复频繁的拉数据。 如果拉到空数据，则多推后这个时间 |
| queued.max.message.chunks | 2 | high level consumer内部缓存拉回来的消息到一个队列中。 这个值控制这个队列的大小 |
| auto.commit.enable | true | 如果true,consumer定期地往zookeeper写入每个分区的offset |
| auto.commit.interval.ms | 10000 | 往zookeeper上写offset的频率 |
| auto.offset.reset | largest | 如果offset出了返回，则 smallest: 自动设置reset到最小的offset. largest : 自动设置offset到最大的offset. 其它值不允许，会抛出异常. |
| consumer.timeout.ms | -1 | 默认-1,consumer在没有新消息时无限期的block。如果设置一个正值， 一个超时异常会抛出 |
| rebalance.retries.max | 4 | rebalance时的最大尝试次数 |

**Producer配置参数**

下表列出了producer的重要的参数。  
更多的配置请参考 [kafka.producer.ProducerConfig](https://github.com/apache/kafka/blob/0.8.2/core/src/main/scala/kafka/producer/ProducerConfig.scala" \t "_blank)

| **name** | **默认值** | **描述** |
| --- | --- | --- |
| serializer.class | kafka.serializer.DefaultEncoder | 必须实现kafka.serializer.Encoder接口，将T类型的对象encode成kafka message |
| key.serializer.class | serializer.class | key对象的serializer类 |
| partitioner.class | kafka.producer.DefaultPartitioner | 必须实现kafka.producer.Partitioner，根据Key提供一个分区策略 |
| producer.type | sync | 指定消息发送是同步还是异步。异步asyc成批发送用kafka.producer.AyncProducer， 同步sync用kafka.producer.SyncProducer |
| metadata.broker.list | boker list | 使用这个参数传入boker和分区的静态信息，如host1:port1,host2:port2, 这个可以是全部boker的一部分 |
| compression.codec | NoCompressionCodec | 消息压缩，默认不压缩 |
| compressed.topics | null | 在设置了压缩的情况下，可以指定特定的topic压缩，为指定则全部压缩 |
| message.send.max.retries | 3 | 消息发送最大尝试次数 |
| retry.backoff.ms | 300 | 每次尝试增加的额外的间隔时间 |
| topic.metadata.refresh.interval.ms | 600000 | 定期的获取元数据的时间。当分区丢失，leader不可用时producer也会主动获取元数据，如果为0，则每次发送完消息就获取元数据，不推荐。如果为负值，则只有在失败的情况下获取元数据。 |
| queue.buffering.max.ms | 5000 | 在producer queue的缓存的数据最大时间，仅仅for asyc |
| queue.buffering.max.message | 10000 | producer 缓存的消息的最大数量，仅仅for asyc |
| queue.enqueue.timeout.ms | -1 | 0当queue满时丢掉，负值是queue满时block,正值是queue满时block相应的时间，仅仅for asyc |
| batch.num.messages | 200 | 一批消息的数量，仅仅for asyc |
| request.required.acks | 0 | 0表示producer毋须等待leader的确认，1代表需要leader确认写入它的本地log并立即确认，-1代表所有的备份都完成后确认。 仅仅for sync |
| request.timeout.ms | 10000 | 确认超时时间 |