1. apache kafka 于2017-11-1 推出全新版本1.0.0。

0.10.0 版本里开始引入的 Streams API 在 1.0.0 版本里继续演进，改进了 builder API（KIP-120），新增了用于查看运行时活跃任务的 API（KIP-130）和用于聚合分区的 cogroup API（KIP-150）。增强的 print() 和 writeAsText() 方法让调试变得更容易（KIP-160）。其他更多信息可以参考 Streams 文档。

改进了 Connect 的度量指标（KIP-196），新增了大量用于健康监测的度量指标（KIP-188），并提供了集群的 GloabalTopicCount 和 GlobalPartitionCount 度量指标（KIP-168）。

支持 Java 9，实现更快的 TLS 和 CRC32C，加快了加密速度，降低了计算开销。

调整了 SASL 认证模块的错误处理逻辑（KIP-152），原先的认证错误信息现在被清晰地记录到日志当中。

更好地支持磁盘容错（KIP-112），更优雅地处理磁盘错误，单个 JBOD 上的磁盘错误不会导致整个集群崩溃。

0.11.0 版本中引入的幂等性生产者需要将max.in.flight.requests.per.connection 参数设置为 1，这对吞吐量造成了一定的限制。而在 1.0.0 版本里，这个参数最大可以被设置为 5（KAFKA-5949），极大提升了吞吐量范围。

可以在$KAFKA\_HOME/config/server.properties中通过配置项num.partitions来指定新建Topic的默认Partition数量，也可在创建Topic时通过参数指定，同时也可以在Topic创建之后通过Kafka提供的工具修改

落后太多”指Follower复制的消息落后于Leader后的条数超过预定值（该值可在$KAFKA\_HOME/config/server.properties中通过replica.lag.max.messages配置，其默认值是4000）或者Follower超过一定时间（该值可在$KAFKA\_HOME/config/server.properties中通过replica.lag.time.max.ms来配置，其默认值是10000）未向Leader发送fetch请求。

而对于Producer而言，它可以选择是否等待消息commit，这可以通过request.required.acks来设置。这种机制确保了只要ISR有一个或以上的Follower，一条被commit的消息就不会丢失。

request.required.acks 的设置，

* acks = 0，发就发了，不需要 ack，无论成功与否 ；
* acks = 1，当写 leader replica 成功后就返回，其他的 replica 都是通过fetcher去异步更新的，当然这样会有数据丢失的风险，如果leader的数据没有来得及同步，leader挂了，那么会丢失数据；
* acks = –1, 要等待所有的replicas都成功后，才能返回；这种纯同步写的延迟会比较高。

无论消息是否被消费，kafka 都会保留所有消息。有两种策略可以删除旧数据：

1、 基于时间：log.retention.hours=168

2、 基于大小：log.retention.bytes=1073741824