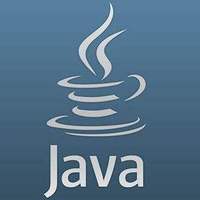
**SpringBoot2 整合Kafka组件，应用案例和流程详解**

播报文章

[](https://author.baidu.com/home?from=bjh_article%26app_id=1555483992904426)

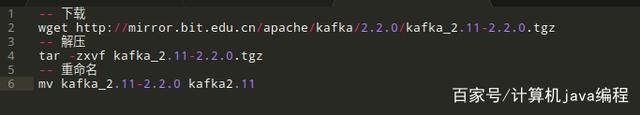
**[计算机java编程](https://author.baidu.com/home?from=bjh_article&app_id=1555483992904426" \t "https://baijiahao.baidu.com/_blank)**

2020-02-17 10:15科技领域创作者

关注

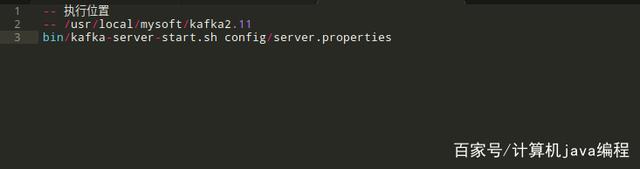
**一、搭建Kafka环境**

**1、下载解压**



**2、启动Kafka服务**

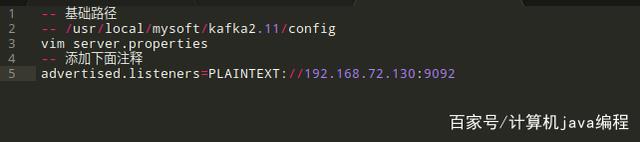
kafka依赖ZooKeeper服务，需要本地安装并启动ZooKeeper。Zookeeper的安装可以参考的另一篇文章：CentOS下正确安装Kafka Broker的方式，这篇文章是kafka安装的前提，主要讲解了kafka安装前需要准备zookeeper环境，下一篇我将接着这篇文章写kafka broker的安装及常用配置的讲解，欢迎大家关注哟。



**3、查看服务**

ps -aux |grep kafka

**4、开放地址端口**



**二、Kafka基础概念**

**1、基础描述**

Kafka是由Apache开源，具有分布式、分区的、多副本的、多订阅者，基于Zookeeper协调的分布式处理平台，由Scala和Java语言编写。通常用来搜集用户在应用服务中产生的动作日志数据，并高速的处理。日志类的数据需要高吞吐量的性能要求，对于像Hadoop一样的日志数据和离线分析系统，但又要求实时处理的限制，这是一个可行的解决方案。Kafka的目的是通过Hadoop的并行加载机制来统一线上和离线的消息处理，也是为了通过集群来提供实时的消息。

**2、功能特点**

(1)、通过磁盘数据结构提供消息的持久化，消息存储也能够保持长时间稳定性;

(2)、高吞吐量，即使是非常普通的硬件Kafka也可以支持每秒超高的并发量;

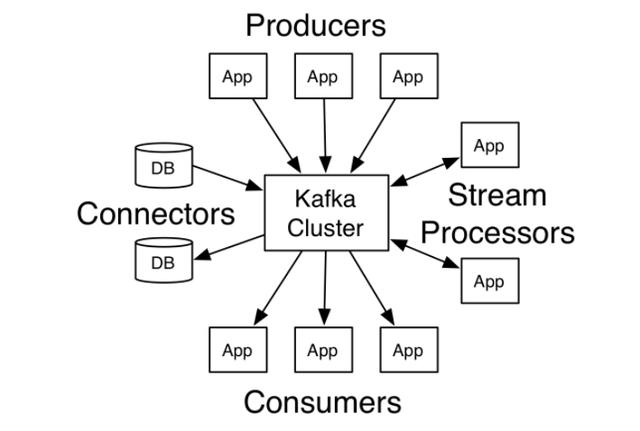
(3)、支持通过Kafka服务器和消费机集群来分区消息;

(4)、支持Hadoop并行数据加载;

(5)、API包封装的非常好，简单易用，上手快 ;

(6)、分布式消息队列。Kafka对消息保存时根据Topic进行归类，发送消息者称为Producer，消息接受者称为Consumer;

**3、消息功能**



点对点模式点对点模型通常是一个基于拉取或者轮询的消息传递模型，消费者主动拉取数据，消息收到后从队列移除消息，这种模型不是将消息推送到客户端，而是从队列中请求消息。特点是发送到队列的消息被一个且只有一个消费者接收处理，即使有多个消费者监听队列也是如此。

发布订阅模式发布订阅模型则是一个基于推送的消息传送模型，消息产生后，推送给所有订阅者。发布订阅模型可以有多种不同的订阅者，临时订阅者只在主动监听主题时才接收消息，而持久订阅者则监听主题的所有消息，即使当前订阅者不可用，处于离线状态。

**4、消息队列作用**

程序解耦，生产者和消费者独立，各自异步执行;消息数据进行持久化存储，直到被全部消费，规避了数据丢失风险;流量削峰，使用消息队列承接访问压力，尽量避免程序雪崩 ;降低进程间的耦合度，系统部分组件崩溃时，不会影响到整个系统;保证消息顺序执行，解决特定场景业务需求 ;**5、专业术语简介**

Broker一台kafka服务器就是一个broker。一个集群由多个broker组成。一个broker可以容纳多个topic。

Producer消息生产者，就是向kafka broker发消息的客户端。

Consumer消息消费者，向kafka broker取消息的客户端。

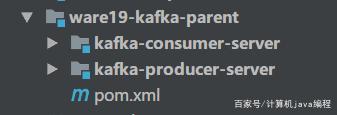
Topic每条发布到Kafka集群的消息都有一个类别，这个类别被称为Topic，可以理解为一个队列。

Consumer Group每个Consumer属于一个特定的Consumer Group，可为每个Consumer指定group name，若不指定group name则属于默认的分组。

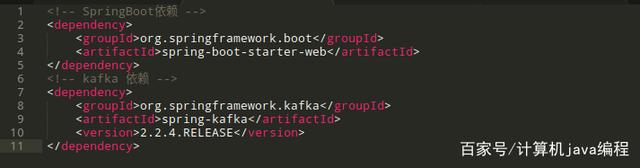
Partition一个庞大大的topic可以分布到多个broker上，一个topic可以分为多个partition，每个partition是一个有序的队列。partition中的每条消息都会被分配一个有序的id。kafka只保证按一个partition中的顺序将消息发给consumer，不保证一个topic的整体的顺序。Partition是物理上的概念，方便在集群中扩展，提高并发

**三、整合SpringBoot2框架**

**1、案例结构**



消息生产者 : kafka-producer-server消息消费方 : kafka-consumer-server**2、基础依赖**



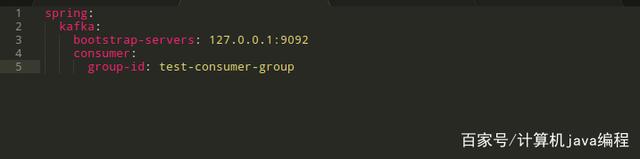
**3、生产者配置**



**4、消息生成**



**5、消费者配置**



**6、消息消费**

