

# Kafka

C-2 创建: 张林伟, 最后修改: 张林伟 2018-07-19 16:07

## 目录

- 简介
- QuickStart
- 原理

## 简介

Kafka 是一个基于分布式的消息发布-订阅系统，原本开发自 LinkedIn，用作 LinkedIn 的活动流（Activity Stream）和运营数据处理管道（Pipeline）的基础。现在它已被多家公司作为多种类型的数据管道和消息系统使用。

消息队列技术是分布式应用间交换信息的一种技术。消息队列可驻留在内存或磁盘上, 队列存储消息直到它们被应用程序读走。

发布/订阅 (Publish/Subscribe) 模式：发布/订阅功能使消息的分发可以突破目的队列地理指向的限制，使消息按照特定的主题甚至内容进行分发，用户或应用程序可以根据主题或内容接收到所需要的消息。

发布/订阅功能使得发送者和接收者之间的耦合关系变得更为松散，发送者不必关心接收者的目的地址，而接收者也不必关心消息的发送地址，而只是根据消息的主题进行消息的收发。

## QuickStart

1.下载kafka\_2.11-1.1.0并解压

^ 代码块Shell

```
1 > tar -xzf kafka_2.11-1.1.0.tgz
2 > cd kafka_2.11-1.1.0
```

2.打开服务器

Kafka基于ZooKeeper，所以需要先开启ZooKeeper服务器

^ 代码块Shell

```
1 > bin/zookeeper-server-start.sh config/zookeeper.properties
2 [2013-04-22 15:01:37,495] INFO Reading configuration from: config/zookeeper.properties
  (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
3 ...
```

然后开启Kafka服务器

^ 代码块Shell

```
1 > bin/kafka-server-start.sh config/server.properties
2 [2013-04-22 15:01:47,028] INFO Verifying properties (kafka.utils.VerifiableProperties)
3 [2013-04-22 15:01:47,051] INFO Property socket.send.buffer.bytes is overridden to 1048576 (kafka.utils.VerifiableProperties)
4 ...
```

3.创建Topic

创建名为“test”的Topic

^ 代码块Shell

```
1 > bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test
```

输入如下命令可以查看创建的Topic

^ 代码块Shell

```
1 > bin/kafka-topics.sh --list --zookeeper localhost:2181
2 test
```

4.发送消息

kafka接受来自命令行的标准输入或者文件，作为消息发送给kafka集群，默认每行作为一条消息。

运行生产者，并输入消息

^ 代码块Shell

```
1 > bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test
2 This is a message
3 This is another message
```

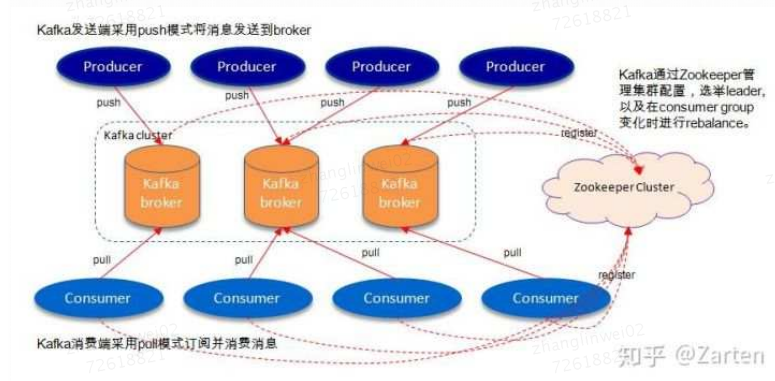
## 5.消费者

### 代码块

Shell

```
1 > bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning
2 This is a message
3 This is another message
```

## 原理



## 概念

- Broker: Kafka 集群包含一个或多个服务器, 这种服务器被称为 broker。
- 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。(物理上不同 Topic 的消息分开存储, 逻辑上一个 Topic 的消息虽然保存于一个或多个 broker 上, 但用户只需指定消息的 Topic 即可生产或消费数据而不必关心数据存于何处)。
- Partition: Partition 是物理上的概念, 每个 Topic 包含一个或多个 Partition。
- Producer: 负责发布消息到 Kafka broker。
- Consumer: 消息消费者, 向 Kafka broker 读取消息的客户端。
- 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 group)。

Kafka 通过给每一个消息绑定一个键值的方式来保证生产者可以把所有的消息发送到指定位置。属于某一个消费者群组的消费者订阅了一个主题, 通过该订阅消费者可以跨节点地接收所有与该主题相关的消息, 每一个消息只会发送给群组中的一个消费者, 所有拥有相同键值的消息都会被确保发给这一个消费者。

© 仅供内部使用, 未经授权, 切勿外传