# Kafka安装配置实战

1. 下载

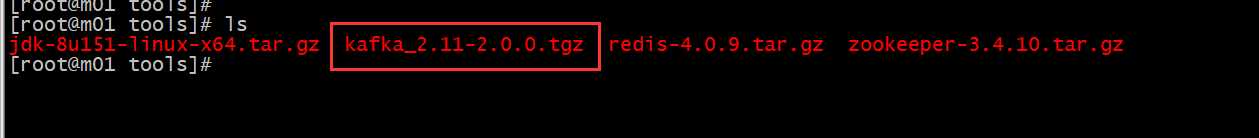
官方下载网址: <http://kafka.apache.org/downloads>

在这里我下载[Scala 2.11 - kafka\_2.11-2.0.0.tgz (asc, sha512)](https://www.apache.org/dyn/closer.cgi?path=/kafka/2.0.0/kafka_2.11-2.0.0.tgz)这个版本

下载后存储linux 主机的 /tools 目录下

下载命令:

wget <http://mirror.bit.edu.cn/apache/kafka/2.0.0/kafka_2.11-2.0.0.tgz>



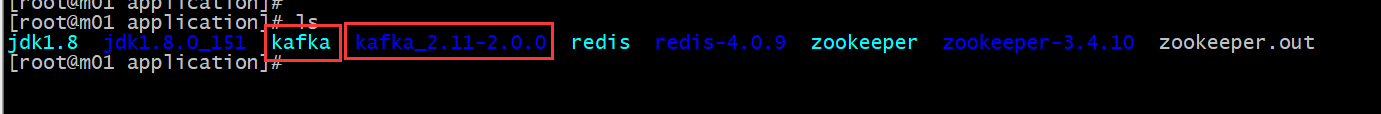
1. 安装

**解压到/application目录下**

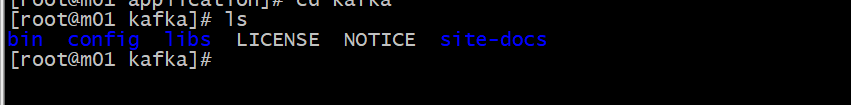
**tar -xvf kafka\_2.11-2.0.0.tgz -C /application/**

**创建软连接**

**ln -s kafka\_2.11-2.0.0/ kafka**



文件结构



bin目录: kafka基本操作的脚本文件

config目录:kafka的相关配置文件

logs：kafka运行过程中的一些日志文件，启动后会创建

1. 配置和启动

完成上面2个操作步骤，实际上kafka已经基本准备完毕，接下来我们要做的就是根据自己的需要去配置kafka基本信息。Kafka启动之前首先需要有zookeeper服务器，因为zookeeper负责维护kafka的broker(代理服务器), topic 以及生产者和消费者的调用关系，在集群模式下，不同的broker没有关联，当broker启动时，会主动到zookeeper下注册自己，利用zk使其联系起来。

在生产环境中我们一般将kafka和zookeeper放在不用的机器上更好的管理。实际上kafka也内置了一个zookeeper，直接使用这个kafka 做练习就行了。

1. **开启zookeeper**

先进入kafka安装目录

./bin/zookeeper-server-start.sh config/zookeeper.properties

## **kafka服务器配置文件**

现在可以启动kafka服务器了，启动之前有几个配置需要注意，kafka的配置文件是server.properties。这里面注意几个重要的属性：

1. **broker.id 申明当前kafka服务器在集群中的唯一ID，需配置为integer,并且集群中的每一个kafka服务器的id都应是唯一的，我们这里采用默认配置即可**

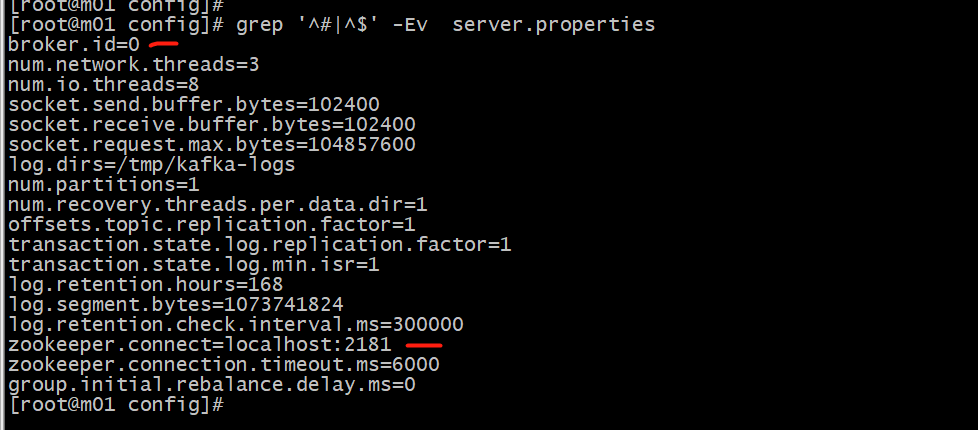
**2、listeners 申明此kafka服务器需要监听的端口号，如果是在本机上跑虚拟机运行可以不用配置本项，默认会使用localhost的地址，如果是在远程服务器上运行则必须配置，**

**例如：**

**listeners=PLAINTEXT:// 192.168.180.128:9092。并确保服务器的9092端口能够访问**

1. **zookeeper.connect 申明kafka所连接的zookeeper的地址 ，需配置为zookeeper的地址，由于本次使用的是kafka高版本中自带zookeeper，使用默认配置即可zookeeper.connect=localhost:2181**

**下面是配置的内容，一般不需要更改什么，取默认设置就行**

****

**启动kafka服务器**

**./bin/kafka-server-start.sh config/server.properties**

**后台运行**

**./bin/kafka-server-start.sh -daemon config/server.properties**

**至此kafka服务已经启动起来了。**

4、快速使用

以上配置只是基于单台服务器，并不是集群模式，集群模式的配置稍后记录。

使用kafka 之前 先明确几个主要的概念

**Broker(代理服务器):就是集群中的服务器，以上操作我们只配置启动了一台。**

**Topic(主题): 相当于一类消息的载体，kafka将不同类别的消息放到到不同的topic可以 提高性能。**

**Partition(分区):topic可以分成多个分区，提高系统吞吐量。**

因此我们在使用kafka的时候，应该先创建一个Topic，Topic的创建可以放到配置文件中，也可以直接通过命令创建Topic

1. 创建topic

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

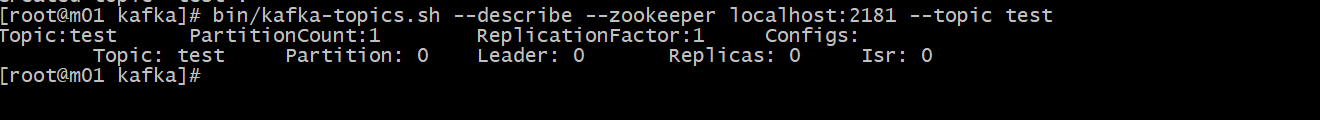
--replication-factor 1 //创建的副本数

--partitions 1 //创建的分区数

--topic test //topic名字

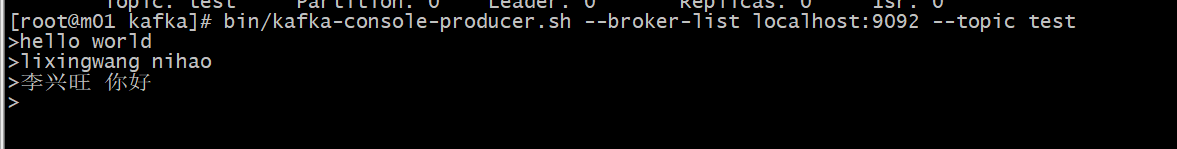
查看创建的topic

bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic test



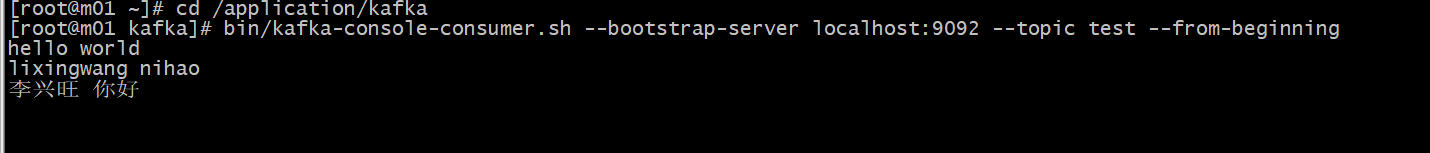
1. **生产者发送消息**

bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test



**③消费者消费消息**

bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning



从上面可以看到kafka可以正常的发送消息，消费消息了。我们可以使用java python等语言去写客户端程序去连接到kafka服务实例进行生产应用了。

5、集群模式

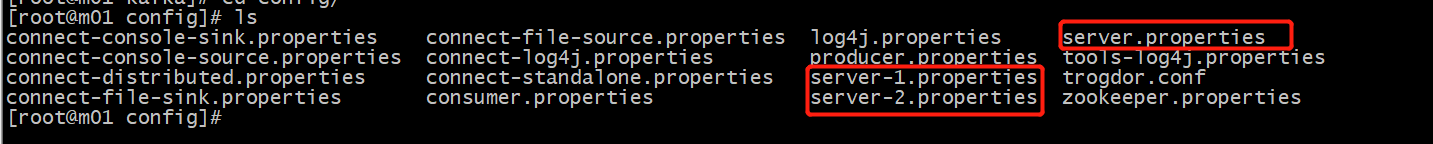
1. 设置多个broker

以上四步搭建的kafka服务器只是单点服务器模式，日常生产上也不会采用此种方式，所以需要多设置几个broker，设置broker的方式就是使用不同的配置文件即可。

Kafka服务配置文件在是config/server.properties，现在需求创建三个broker，复制两份配置文件为 server-1.properties server-2.properties

cp config/server.properties config/server-1.properties

cp config/server.properties config/server-2.properties



现在编辑这些新建的文件，设置以下属性：

config/server-1.properties:

broker.id=1

listeners=PLAINTEXT://:9093

log.dir=/tmp/kafka-logs-1

config/server-2.properties:

broker.id=2

listeners=PLAINTEXT://:9094

log.dir=/tmp/kafka-logs-2

broker.id是集群中每个节点的唯一且永久的名称，我们修改端口和日志目录是因为我们现在在同一台机器上运行，我们要防止broker在同一端口上注册和覆盖对方的数据。

1. 启动broker

各个broker配置文件配置完毕后，可以将3个broker全部启动

./bin/kafka-server-start.sh config/server.properties

./bin/kafka-server-start.sh config/server-1.properties

./bin/kafka-server-start.sh config/server-2.properties

如果开启第三个broker失败，确认配置无误，可以适当调大虚拟机的内存。

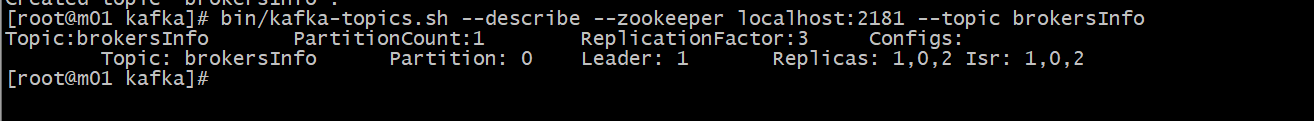
1. 创建topic

创建一个测试用的topic 备份数设置为3 分区数为 1

bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 3 --partitions 1 --topic brokersInfo

现在是一个集群了，查看下topic在集群中的情况

bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic brokersInfo



输出解释：第一行是所有分区的摘要，其次，每一行提供一个分区信息，因为我们只有一个分区，所以只有一行。

"leader"：该节点负责该分区的所有的读和写，每个节点的leader都是随机选择的。

"replicas"：备份的节点列表，无论该节点是否是leader或者目前是否还活着，只是显示。

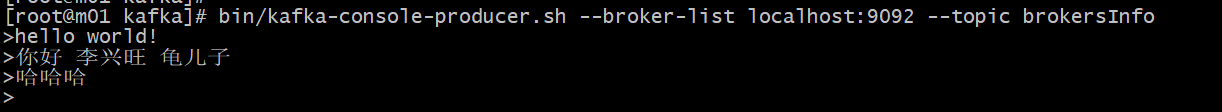
"isr"：“同步备份”的节点列表，也就是活着的节点并且正在同步leader。

1. 生产/消费消息

生产者 生产消息 和消费者消费消息和单点服务一样，生产者 需要指定生产的消息放在哪个broker上。

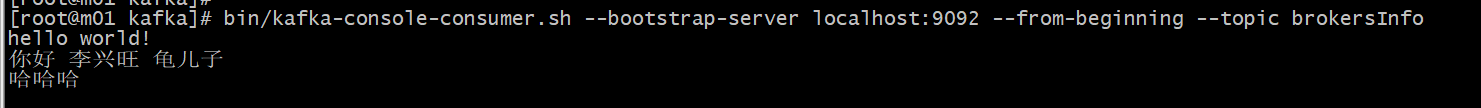
生产者 发送消息

bin/kafka-console-producer.sh --broker-list localhost:9092 --topic brokerInfo



消费者消费消息

bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --from-beginning --topic brokersInfo



1. 模拟broker故障

当前broker的leader是 broker.id=1的服务器，现在我们强制杀掉该服务进程。观察

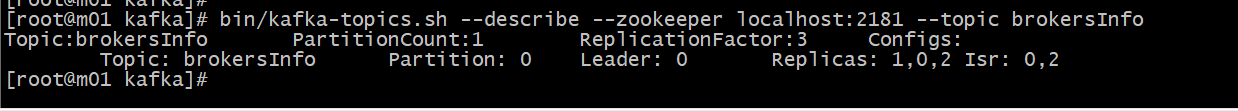
[root@m01 ~]# netstat -lntup|grep 9093

tcp 0 0 :::9093 :::\* LISTEN 4396/java

[root@m01 ~]# kill -9 4396

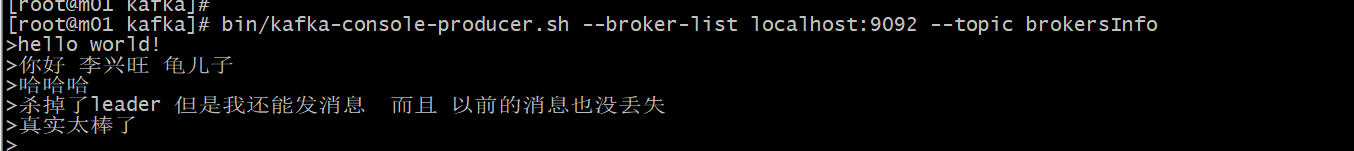
再次查看topic 的情况

bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic brokersInfo

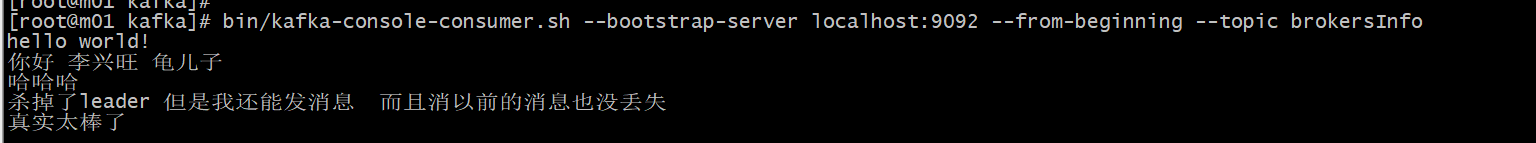


备份节点0成为新的leader，而broker1已经不在同步备份集合里了。

继续发送消息



消费端以前的消息也并没有丢失



至此集群的搭建已经完成了。 具体在生产环境中应用 文件配置 都需要在java里面实现了。