Kafka安装指导手册

# 一、环境准备：

**Kafka:** kafka\_2.11-2.0.0.tgz

**Zookeeper：**zookeeper-3.4.13.tar.gz

**Centos 7.X三台:**

172.18.31.201,172.18.31.202,172.18.31.203

**Jdk:**请自行安装1.8以上的jdk环境

*备注：三台机器均使用kafka用户，用户目录/home/kafka*

*useradd -d /home/kafka -g users kafka*

*passwd kafka*

# 二、Zookeeper安装

下载地址：

<https://archive.apache.org/dist/zookeeper/>

## 2.1创建使用路径

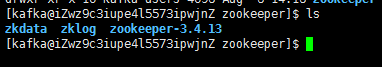
mkdir zookeeper

cd zookeeper

mkdir zkdata

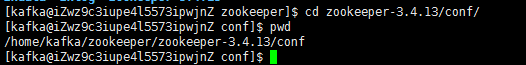
mkdir zklog

解压zookeeper-3-4.13.tar.gz到当前目录，目录结果如下图所示：



## 2.2 zookeeper配置

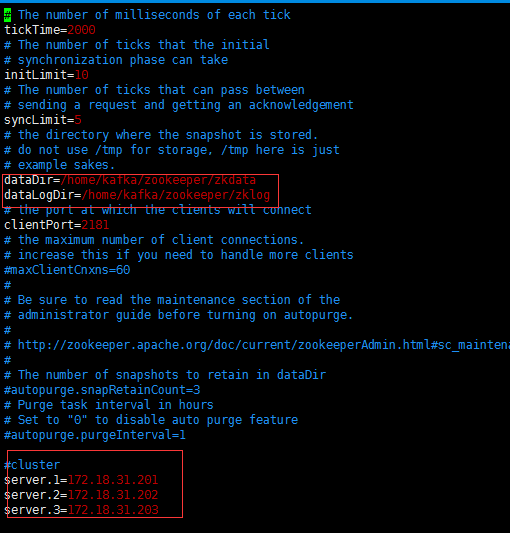
cd zookeeper-3.4.13/conf/



cp zoo\_sample.cfg zoo.cfg



vi zoo.cfg



a.修改数据路径

b.添加日志路径

c.集群配置

d.各项配置说明：

|  |
| --- |
| #tickTime：  这个时间是作为 Zookeeper 服务器之间或客户端与服务器之间维持心跳的时间间隔，也就是每个 tickTime 时间就会发送一个心跳。  #initLimit：  这个配置项是用来配置 Zookeeper 接受客户端（这里所说的客户端不是用户连接 Zookeeper 服务器的客户端，而是 Zookeeper 服务器集群中连接到 Leader 的 Follower 服务器）初始化连接时最长能忍受多少个心跳时间间隔数。当已经超过 5个心跳的时间（也就是 tickTime）长度后 Zookeeper 服务器还没有收到客户端的返回信息，那么表明这个客户端连接失败。总的时间长度就是 5\*2000=10 秒  #syncLimit：  这个配置项标识 Leader 与Follower 之间发送消息，请求和应答时间长度，最长不能超过多少个 tickTime 的时间长度，总的时间长度就是5\*2000=10秒  #dataDir：  快照日志的存储路径  #dataLogDir：  事物日志的存储路径，如果不配置这个那么事物日志会默认存储到dataDir制定的目录，这样会严重影响zk的性能，当zk吞吐量较大的时候，产生的事物日志、快照日志太多  #clientPort：  这个端口就是客户端连接 Zookeeper 服务器的端口，Zookeeper 会监听这个端口，接受客户端的访问请求。修改他的端口改大点  Eg：server.1=192.168.7.100:12888:13888  server.1=172.18.31.201:2888:3888  #server.1：  这个1是服务器的标识也可以是其他的数字， 表示这个是第几号服务器，用来标识服务器，这个标识要写到快照目录下面myid文件里  #192.168.7.107为集群里的IP地址，第一个端口是master和slave之间的通信端口，默认是2888，第二个端口是leader选举的端口，集群刚启动的时候选举或者leader挂掉之后进行新的选举的端口默认是3888 |

*注：zoo.cfg 文件是zookeeper配置文件 在conf目录里，初始提供的是zoo\_sample.cfg文件。*

## 2.3 创建myid文件

切换到zkdata目录



三台服务分别执行：

echo "1" > myid

echo "2" > myid

echo "3" > myid

1对应172.18.31.201服务器，2对应172.18.31.202服务器，3对应172.18.31.203服务器



注：myid文件和server.myid  在快照目录下存放的标识本台服务器的文件，他是整个zk集群用来发现彼此的一个重要标识

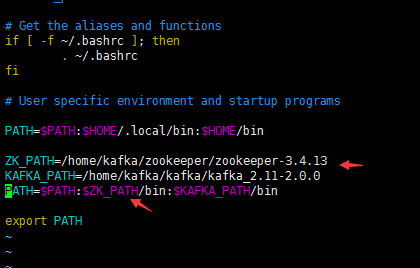
## 2.4 添加zk到环境变量中方便直接执行命令(可跳过本章节)

回到用户根目录更改.base\_profile文件



vi .bash\_profile

添加：ZK\_PATH=/home/kafka/zookeeper/zookeeper-3.4.13



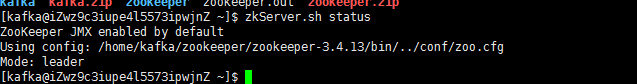
source .bash\_profile

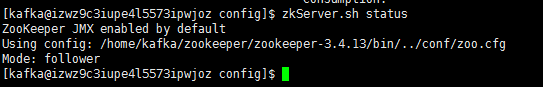
## 2.5 分别启动三台服务器上的zk

zkServer.sh start

查看启动后状态

zkServer.sh status





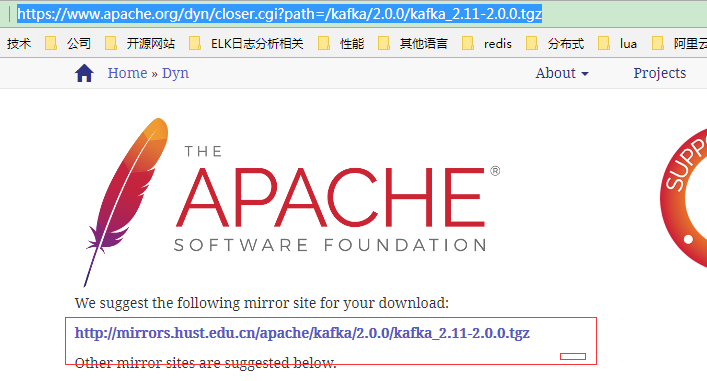
*注：*

*Mode:leader & follower 两种值*

# 三、kafka安装

3.1下载地址：

<https://www.apache.org/dyn/closer.cgi?path=/kafka/2.0.0/kafka_2.11-2.0.0.tgz>



可直接在linxu执行

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

3.2 用户根目录下创建kafka目录

mkdir kafka

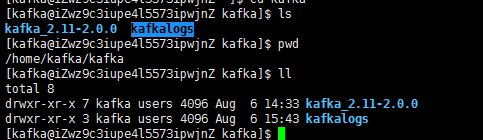


cd kafka

创建kafka日志目录

mkdir kafkalogs

下载kafka2.11解压到kafka目录，如下图结构



3.3环境变量配置，方便直接使用命令(可跳过)

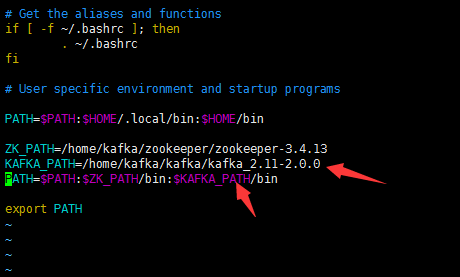
回到用户根目录编辑.bash\_profile文件



vi .bash\_profile

添加kafka目录

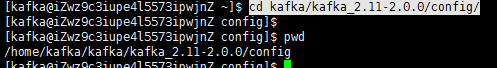
KAFKA\_PATH=/home/kafka/kafka/kafka\_2.11-2.0.0



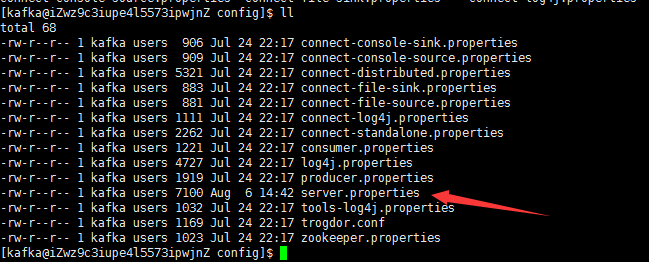
source .bash\_profile

3.4 kafka配置

cd kafka/kafka\_2.11-2.0.0/config/

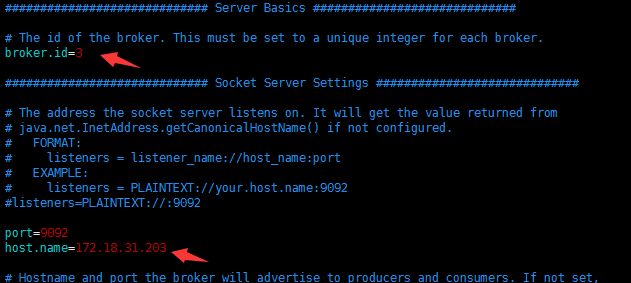


重点关注server.properties文件



vi server.properties

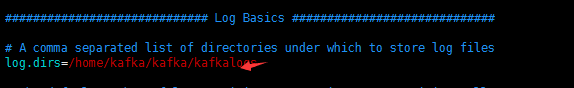
主要修改地方如下几点：



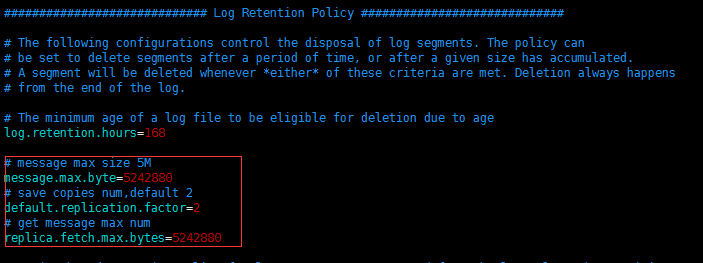
broker.id在集群内全局唯一

port 默认9092

host.name 绑定主机IP



log.dirs前步骤所创建的logs目录

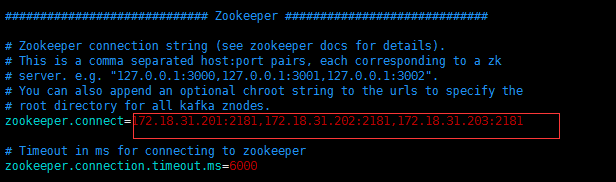


添加如下三项：

message.max.byte：消息保存最大值

default.repliaction.factor：消息副本数，如果其中一个失效，另一个还可以继续提供服务

replica.fetch.max.bytes：取消息的最大数



Zookeeper的配置

各参数参考说明：

|  |
| --- |
| broker.id=0 #当前机器在集群中的唯一标识，和zookeeper的myid性质一样  port=19092 #当前kafka对外提供服务的端口默认是9092  host.name=192.168.7.100 #这个参数默认是关闭的，在0.8.1有个bug，DNS解析问题，失败率的问题。  num.network.threads=3 #这个是borker进行网络处理的线程数  num.io.threads=8 #这个是borker进行I/O处理的线程数  log.dirs=/opt/kafka/kafkalogs/ #消息存放的目录，这个目录可以配置为“，”逗号分割的表达式，上面的num.io.threads要大于这个目录的个数这个目录，如果配置多个目录，新创建的topic他把消息持久化的地方是，当前以逗号分割的目录中，那个分区数最少就放那一个  socket.send.buffer.bytes=102400 #发送缓冲区buffer大小，数据不是一下子就发送的，先回存储到缓冲区了到达一定的大小后在发送，能提高性能  socket.receive.buffer.bytes=102400 #kafka接收缓冲区大小，当数据到达一定大小后在序列化到磁盘  socket.request.max.bytes=104857600 #这个参数是向kafka请求消息或者向kafka发送消息的请请求的最大数，这个值不能超过java的堆栈大小  num.partitions=1 #默认的分区数，一个topic默认1个分区数  log.retention.hours=168 #默认消息的最大持久化时间，168小时，7天  message.max.byte=5242880 #消息保存的最大值5M  default.replication.factor=2 #kafka保存消息的副本数，如果一个副本失效了，另一个还可以继续提供服务  replica.fetch.max.bytes=5242880 #取消息的最大直接数  log.segment.bytes=1073741824 #这个参数是：因为kafka的消息是以追加的形式落地到文件，当超过这个值的时候，kafka会新起一个文件  log.retention.check.interval.ms=300000 #每隔300000毫秒去检查上面配置的log失效时间（log.retention.hours=168 ），到目录查看是否有过期的消息如果有，删除  log.cleaner.enable=false #是否启用log压缩，一般不用启用，启用的话可以提高性能  zookeeper.connect=192.168.7.100:12181,192.168.7.101:12181,192.168.7.107:1218 #设置zookeeper的连接端口 |

3.5 启动集群(三台服务器同样的方式启动)

如进行环境变量的配置3.3章节，进入zookeeper/conf目录下启动



kafka-server-start.sh -daemon server.properties

3.6 测试

203服务器创建一个 Topic

kafka-topics.sh --create --zookeeper 172.18.31.203:2181 --replication-factor 2 --partitions 1 --topic omni-test



|  |
| --- |
| #解释  --replication-factor 2 #复制两份  --partitions 1 #创建1个分区  --topic #主题为shuaige |

202服务器创一个 生产者

kafka-console-producer.sh --broker-list 172.18.31.203:9092 --topic omni

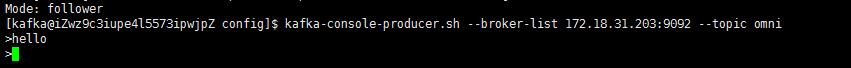


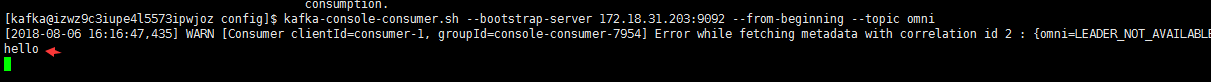
201服务器创建一个 消费者

kafka-console-producer.sh --broker-list 172.18.31.203:9092 --topic omni



在生产端发送消息 hello，消费端即能收到消息：





至此，kafka集群搭建完成