**Kafka安装与配置**

# 环境准备

1、（3台服务器）

192.168.7.100 server1

192.168.7.101 server2

192.168.7.107 server3

2、已经搭建好的zookeeper集群，参考《zookeeper安装.docx》文档

3、软件版本kafka\_2.11-2.1.0.tgz

# 安装

本章节操作所有节点一样，最好在一个节点完成，然后远拷贝到其它节点

创建目录

cd /opt/

mkdir kafka #项目目录

cd kafka

mkdir kafkalogs #消息目录，主要存放kafka消息

下载并解压软件

cd /opt/kafka

wget http://apache.opencas.org/kafka/2.1.0/kafka\_2.11-2.1.0.tgz

tar -zxvf kafka\_2.11-2.1.0.tgz

# 修改配置

每个节点都要修改

进入到config目录

cd /opt/kafka/kafka\_2.11-2.1.0/config/

主要关注：server.properties 这个文件即可，我们可以发现在目录下：

有很多文件，这里可以发现有Zookeeper文件，我们可以根据Kafka内带的zk集群来启动，但是建议使用独立的zk集群

修改配置文件中的以下几项：

# broker.id每台服务器都不能相同，例如：

broker.id=0

# 当前节点ip，例如：

host.name=192.168.7.100

# 日志目录

log.dirs=/opt/kafka/kafkalogs/

# 在log.retention.hours=168 下面新增下面三项

message.max.byte=5242880

default.replication.factor=2

replica.fetch.max.bytes=5242880

# 设置zookeeper的连接端口格式为ip:port，多个zk节点用英文逗号相隔

zookeeper.connect=ip:port,ip:port,ip:port,ip:port

# 启动Kafka集群并测试

## 启动服务

#从后台启动Kafka集群（3台都需要启动）

cd /opt/kafka/kafka\_2.11-2.1.0/bin

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

## 检查服务是否启动

#执行命令

jps

# 结果

20348 Jps4233 QuorumPeerMain18991 Kafka

## 验证是否创建成功

创建Topic来验证是否创建成功

更多请看官方文档：<http://kafka.apache.org/documentation.html>

创建Topic

./kafka-topics.sh --create --zookeeper 192.168.7.100:12181 --replication-factor 2 --partitions 1 --topic mytopic

参数说明：

--replication-factor 2 #复制两份

--partitions 1 #创建1个分区

--topic #主题为mytopic

在一台服务器上创建一个发布者

./kafka-console-producer.sh --broker-list 192.168.7.100:19092 --topic mytopic

在一台服务器上创建一个订阅者

./kafka-console-consumer.sh --zookeeper localhost:12181 --topic mytopic --from-beginning

测试（在发布者那里发布消息看看订阅者那里是否能正常收到~）：

## 其他命令

大部分命令可以去官方文档查看

### 查看topic

./kafka-topics.sh --list --zookeeper localhost:12181

### 查看topic状态

/kafka-topics.sh --describe --zookeeper localhost:12181 --topic shuaige

结果如下：

Topic:mytopic Partition: 0 Leader: 1 Replicas: 0,1 Isr: 1

# 其它配置

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，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

[参考](https://www.cnblogs.com/luotianshuai/p/5206662.html)

# 安装kafka-manager

kafka-manager是雅虎开发的一套kafka可视化管理套件。

环境准备：安装sbt

kafka-manager下载地址：<https://github.com/yahoo/kafka-manager/releases>

编译kafka-manager

修改配置文件：application.properties里zookeeper host部分

启动:

进入kafka-manager的bin目录下，执行 ./kafka-manager

（如果报错说已经启动了，删除上级目录下的RUNNING\_PID文件即可）

默认端口9000