**Kafka 部署安装**

**1、kafka有什么？**

producer 消息的生成者，即发布消息

consumer 消息的消费者，即订阅消息

broker Kafka以集群的方式运行，可以由一个或多个服务组成，服务即broker

zookeeper 协调转发

|  |
| --- |
| 1、kafka是一个分布式的消息缓存系统  2、kafka集群中的服务器都叫做broker  3、kafka有两类客户端，一类叫producer（消息生产者），一类叫做consumer（消息消费者），客户端和broker服务器之间采用tcp协议连接  4、kafka中不同业务系统的消息可以通过topic进行区分，而且每一个消息topic都会被分区，以分担消息读写的负载  5、每一个分区都可以有多个副本，以防止数据的丢失  6、某一个分区中的数据如果需要更新，都必须通过该分区所有副本中的leader来更新  7、消费者可以分组，比如有两个消费者组A和B，共同消费一个topic：order\_info,A和B所 |

# **Zookeeper 在 Kafka 中的作用**

<https://www.jianshu.com/p/a036405f989c>

1. **下载 <https://kafka.apache.org/downloads.html>**

**<https://www.kafkatool.com/download.html> 连接工具**

|  |
| --- |
| wget https://mirrors.tuna.tsinghua.edu.cn/apache/kafka/3.0.0/kafka\_2.13-3.0.0.tgz --no-check-certificate  wget <https://dlcdn.apache.org/kafka/3.0.0/kafka_2.13-3.0.0.tgz> --no-check-certificate  http://archive.apache.org/dist/kafka/3.0.0/kafka\_2.12-3.0.0.tgz |



Tips1

找不到或无法加载主类 org.apache.zookeeper.server.quorum.QuorumPeerMain

Tips2 版本

|  |
| --- |
| Scala版本：2.13 Kafka版本：3.0 |

Tips3

|  |
| --- |
| kafka-manager监控 |

#解压：

|  |
| --- |
| tar xf kafka\_2.13-3.0.0.tgz |

#移动到安装目录

|  |
| --- |
| mv kafka\_2.13-3.0.0 /usr/local/  cd /usr/local/ && ln -sfv kafka\_2.13-3.0.0 kafka |

|  |
| --- |
| mkdir /data/logs  mkdir /data/zookeeper |

1. **配置kafka**

**/usr/local/kafka/config/server.properties**

**阿里云服务器9092端口要对自身IP开放**

listeners=PLAINTEXT://172.16.8.19:9092 #阿里内网IP

advertised.listeners=PLAINTEXT://47.254.34.67:9092 #阿里外网IP

|  |
| --- |
| #broker的全局唯一编号，不能重复  broker.id=0  # 指定监听的地址及端口号，该配置项是指定内网ip  listeners=PLAINTEXT://192.168.91.13:9092  # 如果需要开放外网访问，则在该配置项指定外网ip  advertised.listeners=PLAINTEXT://192.168.91.13:9092  # 处理网络请求的线程数量  num.network.threads=3  # 用来处理磁盘IO的现成数量  num.io.threads=8  # 发送套接字的缓冲区大小  socket.send.buffer.bytes=102400  #接受套接字的缓冲区大小  socket.receive.buffer.bytes=102400  # 请求套接字的缓冲区的大小  socket.request.max.bytes=104857600  #消息存放目录，并不是kafka系统日志  log.dirs=/data/logs/kafka-logs  # topic在当前broker上的分片个数  num.partitions=1  # 用来恢复和清理data下数据的线程数量  num.recovery.threads.per.data.dir=1  #\_\_consumer\_offsets副本设置 默认3  offsets.topic.replication.factor=1  transaction.state.log.replication.factor=1  transaction.state.log.min.isr=1  # segment文件保留的最长时间，超时将被删除，保留时间为7天  log.retention.hours=168  # 日志文件中每个segment的大小，默认为1G  log.segment.bytes=1073741824  # 周期性检查文件的时间,这里是300秒，即5分钟  log.retention.check.interval.ms=300000  #broker需要使用zookeeper保存meta数据  zookeeper.connect=localhost:2181  # zookeeper链接超时时间  zookeeper.connection.timeout.ms=18000  group.initial.rebalance.delay.ms=0 |

1. **配置zookeeper**

**/usr/local/kafka/config/zookeeper.properties**

|  |
| --- |
| mkdir /data/zookeeper   #创建zookeeper目录 |

|  |
| --- |
| dataDir=/data/zookeeper  clientPort=2181  maxClientCnxns=0  admin.enableServer=false  dataLogDir=/data/logs/zk-logs |

1. **导出环境变量**

**vim /etc/profile**

|  |
| --- |
| #set kafka environment  export KAFKA\_HOME=/usr/local/kafka  export PATH=$PATH:$KAFKA\_HOME/bin |

**五、启动及停止kafka**

#先启动zookeeper

|  |
| --- |
| /usr/local/kafka/bin/zookeeper-server-start.sh -daemon /usr/local/kafka/config/zookeeper.properties |

#之后才启动kafka

|  |
| --- |
| /usr/local/kafka/bin/kafka-server-start.sh -daemon /usr/local/kafka/config/server.properties |

#关闭kafka

|  |
| --- |
| /usr/local/kafka/bin/kafka-server-stop.sh -daemon /usr/local/kafka/config/server.properties |

#关闭zookeeper

|  |
| --- |
| /usr/local/kafka/bin/zookeeper-server-stop.sh -daemon /usr/local/kafka/config/zookeeper.properties |

可以用jps命令，看有没有kafka的进程



**六、命令**

**<https://www.orchome.com/454>**

**<https://blog.csdn.net/lzb348110175/article/details/119450902>**

# **查看当前服务器中的所有topic**

|  |
| --- |
| kafka-topics.sh --list --bootstrap-server 192.168.91.13:9092 |

# **创建topic**

|  |
| --- |
| kafka-topics.sh --create --bootstrap-server 192.168.91.13:9092 --replication-factor 1 --partitions 1 --topic test  ****备注：****   ****replication-factor**** 1   备份数(1个备份)   ****partitions****1   分区数(1个分区)   ****topic**** test   test为topic的名称 |

# **显示topic详细信息**

|  |
| --- |
| kafka-topics.sh --describe --topic test --bootstrap-server 192.168.91.13:9092 |

# **命令行发送消息**

|  |
| --- |
| kafka-console-producer.sh --topic test --bootstrap-server 192.168.91.13:9092 |

# **命令行消费消息**

|  |
| --- |
| kafka-console-consumer.sh --topic test --from-beginning --bootstrap-server 192.168.91.13:9092 |

# **删除topic**

|  |
| --- |
| kafka-topics.sh --delete --bootstrap-server 192.168.91.13:9092 --topic test |

**七、测试**

 1、创建topic

|  |
| --- |
| cd /usr/local/kafka/bin && ./kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test |

参数说明：

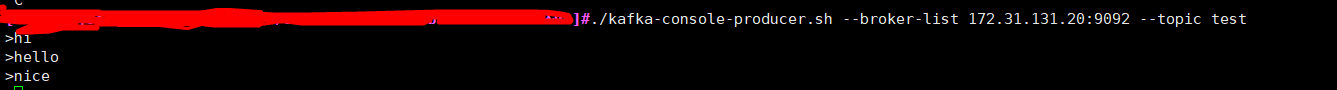
* --create：创建一个topic
* --zookeeper: 指定zookeeper集群的主机列表，多个server可使用逗号分隔，这里因为kafka和zk是在同一个server，所以直接连接了本机的2181端口
* --replication-factor：指定创建这个topic的副本数
* --partitions：指定该topic的分区数
* --topic：指定topic的名称

 2、查看主题

|  |
| --- |
| /kafka-topics.sh --list --zookeeper localhost:2181 |

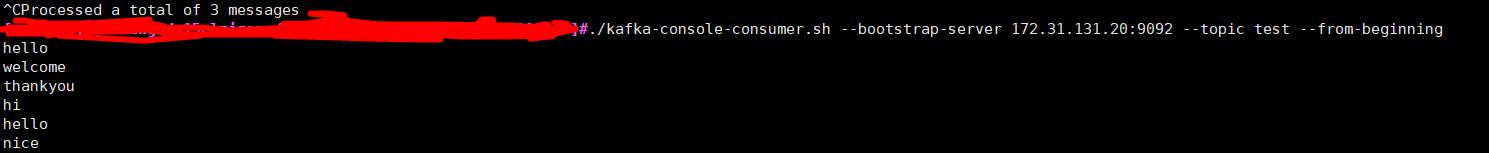
 3、发送消息

|  |
| --- |
| ./kafka-console-producer.sh --broker-list 172.31.131.20:9092 --topic test |



4、接收消息

|  |
| --- |
| ./kafka-console-consumer.sh --bootstrap-server 172.31.131.20:9092 --topic test --from-beginning |



5 、查看特定主题的详细信息

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



从中可以看到，test主题分了1个区，复制因子是1。

参数说明：

* --describe：查看topic的详细信息

输出说明：

* leader：当前负责读写的leader broker
* replicas：当前分区所有的副本对应的broker列表
* isr：处于活动状态的broker

6、 删除主题

|  |
| --- |
| ./kafka-topics.sh --zookeeper localhost:2181 --delete --topic test |

**八、其他**

|  |
| --- |
| 配置说明：   * broker.id：每个broker在集群中的唯一标识，正整数。当该服务器的ip地址发生变更，但broker.id未变，则不会影响consumers的消费情况 * listeners：kafka的监听地址与端口，在实际测试中如果写0.0.0.0会报错。 * num.network.threads：kafka用于处理网络请求的线程数 * num.io.threads：kafka用于处理磁盘io的线程数 * socket.send.buffer.bytes：发送数据的缓冲区 * socket.receive.buffer.bytes：接收数据的缓冲区 * socket.request.max.bytes：允许接收的最大数据包的大小（防止数据包过大导致OOM） * log.dirs：kakfa用于保存数据的目录，所有的消息都会存储在该目录当中。可以通过逗号来指定多个路径，kafka会根据最少被使用的原则选择目录分配新的partition。需要说明的是，kafka在分配partition的时候选择的原则不是按照磁盘空间大小来定的，而是根据分配的partition的个数多少而定 * num.partitions：设置新创建的topic的默认分区数 * number.recovery.threads.per.data.dir：用于恢复每个数据目录时启动的线程数 * log.retention.hours：配置kafka中消息保存的时间，还支持log.retention.minutes和log.retention.ms。如果多个同时设置会选择时间最短的配置，默认为7天。 * log.retention.check.interval.ms：用于检测数据过期的周期 * log.segment.bytes：配置partition中每个segment数据文件的大小。默认为1GB。超出该大小后，会自动创建一个新的segment文件。 * zookeeper.connect：指定连接的zk的地址，zk中存储了broker的元数据信息。可以通过逗号来设置多个值。格式为：hostname:port/path。hostname为zk的主机名或ip，port为zk监听的端口。/path表示kafka的元数据存储到zk上的目录，如果不设置，默认为根目录 * zookeeper.connection.timeout：kafka连接zk的超时时间 * group.initial.rebalance.delay.ms：在实际环境当中，当将多个consumer加入到一个空的consumer group中时，每加入一个consumer就会触发一次对partition消费的重平衡，如果加入100个，就得重平衡100次，这个过程就会变得非常耗时。通过设置该参数，可以延迟重平衡的时间，比如有100个consumer会在10s内全部加入到一个consumer group中，就可以将该值设置为10s，10s之后，只需要做一次重平衡即可。默认为0则代表不开启该特性。 * auto.create.topics.enable：当有producer向一个不存在的topic中写入消息时，是否自动创建该topic * delete.topics.enable：kafka提供了删除topic的功能，但默认并不会直接将topic数据物理删除。如果要从物理上删除（删除topic后，数据文件也一并删除），则需要将此项设置为true |

# **阿里云上部署kafka，远程无法访问问题**

阿里云上部署kafka后，远程总是无法访问，本地机器可以。

确定是网络配置的问题，所以在server.properties 做如下修改：

host.name=阿里云内网地址 *#kafka绑定的interface*

advertised.listeners=PLAINTEXT://阿里云外网映射地址:9092 *# 注册到zookeeper的地址和端口*

**参考文档**

<https://www.osyunwei.com/archives/9345.html>

<https://www.cnblogs.com/zhaoshizi/p/12154518.html>

<https://www.cnblogs.com/breezey/p/10850855.html>