# Kafka消息中间件安装手册

## 安装前准备

* 建立目录

mkdir -p /opt/data/kafka

mkdir -p /opt/data/zookeeper

* 解压文件

将安装文件解压到/opt/app/kafka-[version]目录

* 安装JDK
* 切换目录

cd /opt/app/kafka-[version]

cd config

## 配置

### 配置zookeeper

配置文件：zookeeper.properties

如果已经存在zookeeper可以略过。

|  |
| --- |
| #服务节点Id，全局唯一，与server.1中的1保持一致  serverid=1  dataDir=/opt/data/zookeeper  # the port at which the clients will connect  clientPort=2181  # disable the per-ip limit on the number of connections since this is a non-production config  maxClientCnxns=0  # The number of ticks that the initial  # synchronization phase can take,  initLimit=10  # The number of ticks that can pass between  # sending a request and getting an acknowledgement  syncLimit=5  #集群server配置, 2888是follower和leader进行通讯的端口, 3888是选举leader时使用的端口  server.1=172.20.3.42:2888:3888  server.2=172.20.3.43:2888:3888  server.3=172.20.3.44:2888:3888 |

vi /opt/data/zookeeper/myid写入serverid中配置的值

### 修改Kafka配置文件

配置文件：server.properties

|  |
| --- |
| # The id of the broker. This must be set to a unique integer for each broker.  #每台机器不一样  broker.id=42  ############################# Socket Server Settings #############################  # The port the socket server listens on  port=9092  # Hostname the broker will bind to. If not set, the server will bind to all interfaces  #根据服务器IP设置  host.name=172.20.3.42  # Hostname the broker will advertise to producers and consumers. If not set, it uses the  # value for "host.name" if configured. Otherwise, it will use the value returned from  # java.net.InetAddress.getCanonicalHostName().  #根据服务器IP设置  advertised.host.name=172.20.3.42  # The port to publish to ZooKeeper for clients to use. If this is not set,  # it will publish the same port that the broker binds to.  advertised.port=9092  # The number of threads handling network requests  num.network.threads=30  # The number of threads doing disk I/O  num.io.threads=20  # The send buffer (SO\_SNDBUF) used by the socket server  socket.send.buffer.bytes=104857600  # The receive buffer (SO\_RCVBUF) used by the socket server  socket.receive.buffer.bytes=104857600  # The maximum size of a request that the socket server will accept (protection against OOM)  socket.request.max.bytes=104857600  ############################# Log Basics #############################  # A comma seperated list of directories under which to store log files  log.dirs=/opt/data/kafka-logs  # The default number of log partitions per topic. More partitions allow greater  # parallelism for consumption, but this will also result in more files across  # the brokers.  num.partitions=10  ############################# Log Flush Policy #############################  # Messages are immediately written to the filesystem but by default we only fsync() to sync  # the OS cache lazily. The following configurations control the flush of data to disk.  # There are a few important trade-offs here:  # 1. Durability: Unflushed data may be lost if you are not using replication.  # 2. Latency: Very large flush intervals may lead to latency spikes when the flush does occur as there will be a lot of data to flush.  # 3. Throughput: The flush is generally the most expensive operation, and a small flush interval may lead to exceessive seeks.  # The settings below allow one to configure the flush policy to flush data after a period of time or  # every N messages (or both). This can be done globally and overridden on a per-topic basis.  # The number of messages to accept before forcing a flush of data to disk  #log.flush.interval.messages=10000  # The maximum amount of time a message can sit in a log before we force a flush  log.flush.interval.ms=100  ############################# Log Retention Policy #############################  # The following configurations control the disposal of log segments. The policy can  # be set to delete segments after a period of time, or after a given size has accumulated.  # A segment will be deleted whenever \*either\* of these criteria are met. Deletion always happens  # from the end of the log.  # The minimum age of a log file to be eligible for deletion  log.retention.hours=720  #超时后执行的操作  log.cleanup.policy=delete  # A size-based retention policy for logs. Segments are pruned from the log as long as the remaining  # segments don't drop below log.retention.bytes.  #log.retention.bytes=1073741824  # The maximum size of a log segment file. When this size is reached a new log segment will be created.  log.segment.bytes=536870912  # The interval at which log segments are checked to see if they can be deleted according  # to the retention policies  log.retention.check.interval.ms=60000  # By default the log cleaner is disabled and the log retention policy will default to just delete segments after their retention expires.  # If log.cleaner.enable=true is set the cleaner will be enabled and individual logs can then be marked for log compaction.  log.cleaner.enable=false  #是否允许自动创建topic  auto.create.topics.enable=**true**  **#自动创建topic的默认副本数**  **default**.replication.factor =3  #控制器关闭的尝试次数  controlled.shutdown.max.retries =3  #每次关闭尝试的时间间隔  controlled.shutdown.retry.backoff.ms =5000  #leader的不平衡比例，若是超过这个数值，会对分区进行重新的平衡  leader.imbalance.per.broker.percentage =10  #检查leader是否不平衡的时间间隔  leader.imbalance.check.interval.seconds =300  ############################# Zookeeper #############################  # Zookeeper connection string (see zookeeper docs for details).  # This is a comma separated host:port pairs, each corresponding to a zk  # server. e.g. "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002".  # You can also append an optional chroot string to the urls to specify the  # root directory for all kafka znodes.  zookeeper.connect=172.20.3.42:2181,172.20.3.43:2181,172.20.3.44:2181,172.20.3.45:2181  # Timeout in ms for connecting to zookeeper  zookeeper.connection.timeout.ms=60000 |

### 安装kafka管理程序

解压<http://amoadev.aspirecn.com:8803/common/tools/kafka-web-console-2.1.0-SNAPSHOT.zip>

## 启动

### 启动zookeeper

cd /opt/app/kafka-[version]

分别在集群主机上运行：

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

### 启动kafka

cd /opt/app/kafka-[version]

分别在集群主机上运行：

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

### 启动监控

./kafka-web-console -DapplyEvolutions.default=true