**ActiveMq**

#### ActiveMq安装与配置

1. 下载 <http://activemq.apache.org/download.html>
2. 将下载的tar包上传到linux
3. 解压缩
4. 高可用集群搭建

修改根目录下conf下的activemq.xml配置文件，修改持久化方式

把kahadb改为leveldb

这样就实现了activemq的高可用，通过zookeeper管理选举

<persistenceAdapter>

<!--<kahaDB directory="${activemq.data}/kahadb"/>-->

<replicatedLevelDB

directory="${activemq.data}/leveldb"

replicas="3"

bind="tcp://0.0.0.0:0"

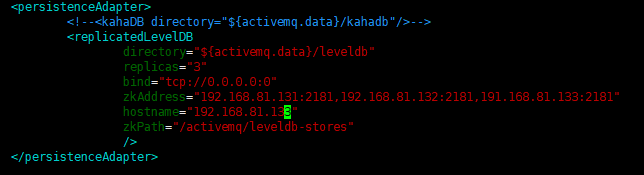
zkAddress="192.168.81.131:2181,192.168.81.132:2181,191.168.81.133:2181"

hostname="192.168.81.133"

zkPath="/activemq/leveldb-stores"

/>

</persistenceAdapter>



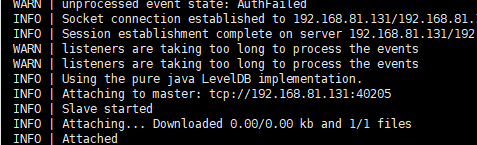
1. 启动集群

首先启动zookeeper服务

然后启动activemq

bin/activemq console & 可以查看启动信息

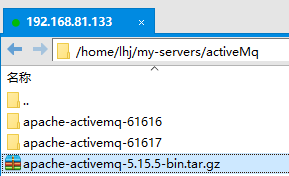
bin/activemq start 相当于后台启动



启动成功

1. 高可用负载均衡的集群搭建

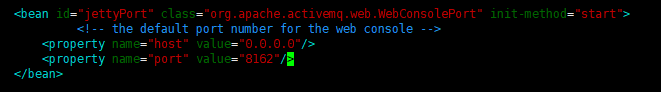
每个应用在三个服务器上各复制一份，这样就有6个activemq



分别修改端口号

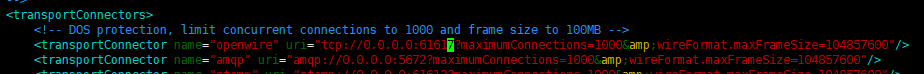
首先修改jetty的端口号

vim conf/jetty.xml

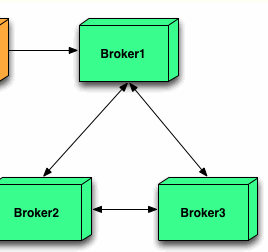


接下来修改activemq的端口号

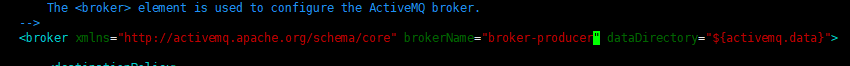
vim conf/activemq.xml

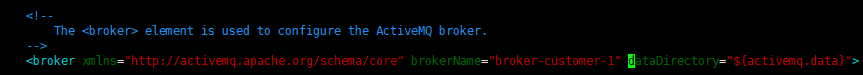


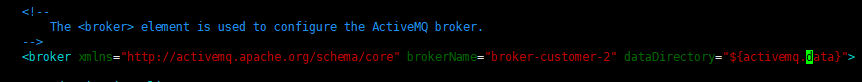
然后修改brokername



参考上面的搭建方式，所以第一台机器的brokername相同作为集群1，第二台机器作为集群2，第三台机器作为集群3

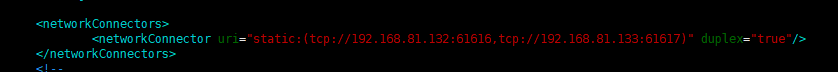




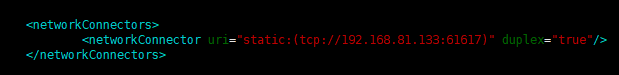


配置networkConnectors（高可用集群只有master才可以配置networkConnectors）

集群1的配置使得1<->2,1<->3



集群2的配置使得2<->3，这样不用配置集群3



改成duplex="true"就可以了，表示双向连接，不用在另一方配置

至此配置成功

1. 可以在activemq管控台查看加管理

localhost:port

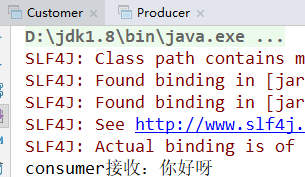
192.168.81.131:9092

#### ActiveMq之java api

1. Jms方式

*//jms方式的生产者***public static void** jmsProducer() **throws** Exception{  
 *//创建连接工厂* ConnectionFactory factory = **new** ActiveMQConnectionFactory(  
 ActiveMQConnectionFactory.***DEFAULT\_USER***,  
 ActiveMQConnectionFactory.***DEFAULT\_PASSWORD***,  
 **"tcp://192.168.81.135:61616"**);  
 *//通过工厂创建一个连接* Connection connection = factory.createConnection();  
 *//启动连接* connection.start();  
 *//创建一个session会话* Session session = connection.createSession(**true**,Session.***AUTO\_ACKNOWLEDGE***);  
 *//创建一个destination消息队列* Destination destination = session.createQueue(**"myActiveMq"**);  
 *//创建消息创建者* MessageProducer producer = session.createProducer(destination);  
 producer.setDeliveryMode(DeliveryMode.***NON\_PERSISTENT***);  
 *//创建一条消息* TextMessage textMessage = session.createTextMessage(**"你好呀"**);  
 producer.send(textMessage);  
 *//提交会话* session.commit();  
 session.close();  
 connection.close();  
}

*//jms方式创建消费者***public static void** jmsCustomer() **throws** Exception{  
 *//创建连接工厂* ConnectionFactory factory = **new** ActiveMQConnectionFactory(  
 ActiveMQConnectionFactory.***DEFAULT\_USER***,  
 ActiveMQConnectionFactory.***DEFAULT\_PASSWORD***,  
 **"tcp://192.168.81.135:61616"**);  
 *//通过工厂创建一个连接* Connection connection = factory.createConnection();  
 *//开启连接* connection.start();  
 *//创建一个session会话* Session session = connection.createSession(**true**,Session.***AUTO\_ACKNOWLEDGE***);  
 *//创建一个destination消息队列* Destination destination = session.createQueue(**"myActiveMq"**);  
 *//创建一个消息消费者* MessageConsumer consumer = session.createConsumer(destination);  
 **while** (**true**){  
 *// 接收数据的时间（等待） 100 ms* Message message = consumer.receive(100\*1000);  
 TextMessage textMessage = (TextMessage) message;  
 **if**(textMessage!=**null**){  
 System.***out***.println(**"consumer接收："**+textMessage.getText());  
 }  
 }  
}



1. Queue方式
2. Topic方式