

# 前言

## **1、ActiveMQ的networkConnector是什么**

　　在某些情况下，需要多个ActiveMQ的Broker做集群，那么就涉及到Broker到Broker的通信，**这个就称为ActiveMQ的networkConnector**.

　　ActiveMQ的networkConnector**默认是单向**的，一个Broker在一端发送消息，另一个Broker在另一端接收消息，这就是所谓的"桥接"。**ActiveMQ也支持双向链接**，创建一个双向的通道对于两个Broker不仅发送消息而且也能从相同的通道接收消息，通常作为duplex connector来映射，如下：

# 1、一个broker发，一个broker接（比自己接收的快）

## 1、编辑activemq.xml文件，添加标签networkConnectors，一个端口:61616，另一个端口:61716，然后通过程序往61616端口的Broker发送数据，再从61716端口的Broker接收数据 （前面url61616只能发送，后面的url61617只能接受）

|  |
| --- |
| <!-- 静态的网络连接 -->  <networkConnectors>  <networkConnector  name="local network"  uri="static://(tcp://localhost:61616,tcp://localhost:61617)"/>  </networkConnectors> |

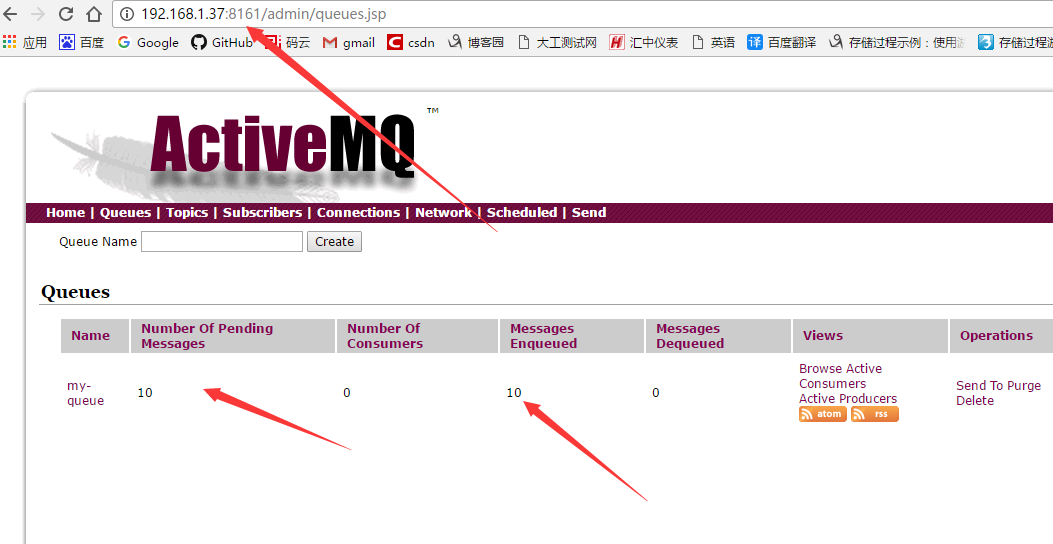
## 2、配置发送者 端口为61616

|  |
| --- |
| **public** **class** JmsSend {  **public** **static** **void** main(String[] args) **throws** Exception {    System.*out*.println("networkConnection success send");  String linuxIp = "myLinuxQj";  ConnectionFactory connectionFactory = **new** ActiveMQConnectionFactory("tcp://"+linuxIp+":61616");  Connection connection = connectionFactory.createConnection();  connection.start();    Session session = connection.createSession(Boolean.*TRUE*, Session.*AUTO\_ACKNOWLEDGE*);  Destination destination = session.createQueue("my-queue");    MessageProducer producer = session.createProducer(destination);  **for**(**int** i = 0;i < 10;i++){  TextMessage message = session.createTextMessage("message,1212 --->" + i);  Thread.*sleep*(1000);  //通过生产者发出消息  producer.send(message);  }  System.*out*.println("networkConnection success send");  session.commit();  session.close();  connection.close();  }  } |

## 3、接受者 端口为61617

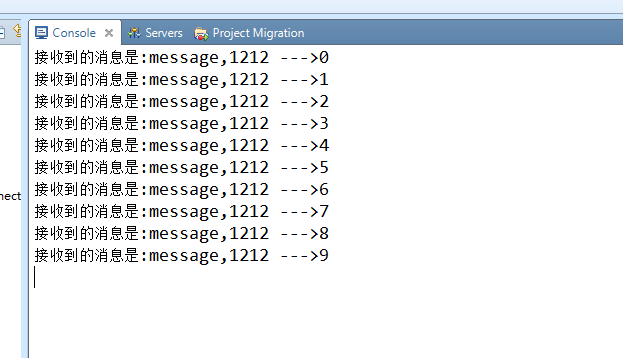
|  |
| --- |
| **public** **class** JmsReceiver {  **public** **static** **void** main(String[] args) **throws** Exception {  String linuxIp = "myLinuxQj";  ConnectionFactory cf = **new** ActiveMQConnectionFactory("tcp://"+linuxIp+":61617");  Connection connection = cf.createConnection();  connection.start();    Session session = connection.createSession(Boolean.*TRUE*, Session.*AUTO\_ACKNOWLEDGE*);  Destination destination = session.createQueue("my-queue");  MessageConsumer consumer = session.createConsumer(destination);  **int** i = 0;  **while**(i < 10){  i++;  TextMessage message = (TextMessage)consumer.receive();  session.commit();  System.*out*.println("接收到的消息是:"+message.getText());  }  session.close();  connection.close();  }  } |

## 4、运行生产者61616，观察浏览器 和network





## 5、运行消费者61617，成功





# 2、设置1中为双向 duplex="true"

|  |
| --- |
| <networkConnectors>  <networkConnector  duplex="true"  name="local network"  uri="static://(tcp://localhost:61616,tcp://localhost:61617)"/>  </networkConnectors> |

# 文件

