ACTIVE\_MQ\_MASTER\_SLAVE\_SETUP

<http://chamilad.github.io/blog/2015/11/17/creating-a-simple-activemq-master-slash-slave-setup/>

**Creating Two Broker Instances**

Unzip the ActiveMQ distribution to two places and offset port values in the second one to use different ports so that there will be no conflicts for ports used by the different protocol connectors. The places to change are in <ACTIVEMQ\_HOME>/conf/activemq.xml and <ACTIVEMQ\_HOME>/conf/jetty.xml.

activemq.xml

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <transportConnectors>  <!-- DOS protection, limit concurrent connections to 1000 and frame size to 100MB -->  <transportConnector name="openwire" uri="tcp://0.0.0.0:61626?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>  <transportConnector name="amqp" uri="amqp://0.0.0.0:5682?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>  <transportConnector name="stomp" uri="stomp://0.0.0.0:61623?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>  <transportConnector name="mqtt" uri="mqtt://0.0.0.0:1893?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>  <transportConnector name="ws" uri="ws://0.0.0.0:61624?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>  </transportConnectors> |

jetty.xml

|  |  |
| --- | --- |
| 1  2  3  4  5 | <bean id="jettyPort" class="org.apache.activemq.web.WebConsolePort" init-method="start">  <!-- the default port number for the web console -->  <property name="host" value="0.0.0.0"/>  <property name="port" value="8171"/>  </bean> |

Now let’s point the KahaDB persistence to the same location. This will result in only one instance at a time being able to acquire the lock to the DB and when the lock is released the other instance will be able get it from the same location.

Modify the persistenceAdapter tag inside <ACTIVEMQ\_HOME>/conf/activemq.xml as follows.

|  |  |
| --- | --- |
| 1  2  3 | <persistenceAdapter>  <kahaDB directory="/tmp/mq/kahadb"/>  </persistenceAdapter> |

Do this change for both of the instances.

Now, let’s introduce the two instances

Now, let’s introduce the two instances to each other by adding a networkConnector pointing to each other. Add the following block to the <ACTIVEMQ\_HOME>/conf/activemq.xml after the persistenceAdapter block in the master.

|  |  |
| --- | --- |
| 1  2  3 | <networkConnectors>  <networkConnector uri="static:(tcp://localhost:61626)" />  </networkConnectors> |

Port 61626 is the OpenWire port in the Slave instance. Similarly add the same block in the Slave activemq.xmlfile, pointing to the Master’s OpenWire port. Static discovery is used here to statically point to the existing broker instances.

**Starting the instances**

Now let’s start the Master broker instance.

|  |  |
| --- | --- |
| 1  2  3  4 | cd <ACTIVEMQ\_HOME>/bin  ./activemq start  # tail the logs just for the fun of it  tail -100f ../data/activemq.log |

When observing the logs you will see some log entries similar to the following repeatedly appearing.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | 2015-11-17 19:10:19,731 | INFO | Establishing network connection from vm://localhost?async=false&network=true to tcp://localhost:61626 | org.apache.activemq.network.DiscoveryNetworkConnector | ActiveMQ Task-61  2015-11-17 19:10:19,733 | INFO | Connector vm://localhost started | org.apache.activemq.broker.TransportConnector | ActiveMQ Task-61  2015-11-17 19:10:19,736 | INFO | localhost Shutting down | org.apache.activemq.network.DemandForwardingBridgeSupport | ActiveMQ BrokerService[localhost] Task-134  2015-11-17 19:10:19,738 | INFO | localhost bridge to Unknown stopped | org.apache.activemq.network.DemandForwardingBridgeSupport | ActiveMQ BrokerService[localhost] Task-134  2015-11-17 19:10:19,739 | INFO | Connector vm://localhost stopped | org.apache.activemq.broker.TransportConnector | ActiveMQ Task-61  2015-11-17 19:10:19,741 | WARN | Could not start network bridge between: vm://localhost?async=false&network=true and: tcp://localhost:61626 due to: Connection refused | org.apache.activemq.network.DiscoveryNetworkConnector | ActiveMQ Task-61 |

This is because we “introduced” the Slave broker to the Master broker and now it’s looking for it.

Now start the Slave broker and tail the logs. You will see a different set of logs appearing.

|  |  |
| --- | --- |
| 1 | 2015-11-17 18:34:37,359 | INFO | Database /tmp/mq/kahadb/lock is locked... waiting 10 seconds for the database to be unlocked. Reason: java.io.IOException: File '/tmp/mq/kahadb/lock' could not be locked. | org.apache.activemq.store.SharedFileLocker | main |

This is because the lock for the shared DB is already acquired by the Master broker. The Slave broker will not start until it is able to acquire the lock for the DB. If you try to see which ports are open using the netstat command you will see that only the Master broker is up and running and ready to accept requests.

Now if you connect to the broker setup using the failover: transport you will see that the client connected the Master broker. Create a queue and publish an event to the queue without consuming it. Now stop the Master broker. You will see the Slave broker acquiring the lock to the DB and become ready to accept requests. Start a consumer with the failover transport and observe it connecting to and retrieving the event (which was published to the Master broker) from the Slave broker. There was no data loss and the service didn’t stop responding for more than a few moments which the Slave took to start up after acquiring the DB lock.