ActiveMQ 高可用集群安装、配置(伪集群)

（ZooKeeper + LevelDB）

1、ActiveMQ 集群部署规划：

环境：CentOS 6.6 x64 、 JDK7

版本：ActiveMQ 5.11.1

ZooKeeper 集群环境：192.168.1.81:2181,192.168.1.82:2182,192.168.1.83:2183

（ZooKeeper 集群部署请参考《高可用架构篇--第 01 节--ZooKeeper 集群的安装、配置、高可用测试》）

主机	集群端口	消息端口	管控台端口	节点安装目录
192.168.1.101	63631	53531	8361	/home/wusc/activemq/node-01
192.168.1.101	63632	53532	8362	/home/wusc/activemq/node-02
192.168.1.101	63633	53533	8363	/home/wusc/activemq/node-03

2、防火墙打开对应的端口

```
## mq cluster
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8361 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8362 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8363 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 53531 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 53532 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 53533 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 63631 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 63632 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 63633 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 63631 -j ACCEPT
```

3、分别在三台主机中创建/home/wusc/activemq 目录

```
$ mkdir /home/wusc/activemq
```

上传 apache-activemq-5.11.1-bin.tar.gz 到/home/wusc/activemq 目录

4、解压并按节点命名

```
$ cd /home/wusc/activemq
$ tar -xvf apache-activemq-5.11.1-bin.tar.gz
$ mv apache-activemq-5.11.1 node-0X    #(X 代表节点号 1、2、3，下同)
```

5、修改管理控制台端口（默认为 8161）可在 conf/jetty.xml 中修改，如下：

Node-01 管控台端口：

```
<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="8361"/>
</bean>
```

Node-02 管控台端口：

```
<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="8362"/>
</bean>
```

Node-03 管控台端口：

```
<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="8363"/>
</bean>
```

6、集群配置：

在 3 个 ActiveMQ 节点中配置 conf/activemq.xml 中的持久化适配器。修改其中 bind、zkAddress、hostname 和 zkPath。注意：每个 ActiveMQ 的 BrokerName 必须相同，否则不能加入集群。

Node-01 中的持久化配置:

```
<broker xmlns="http://activemq.apache.org/schema/core" brokerName="DubboEdu2" dataDirectory="${activemq.data}">
  <persistenceAdapter>
    <!-- kahaDB directory="${activemq.data}/kahadb"/ -->
    <replicatedLevelDB
          directory="${activemq.data}/leveldb"
          replicas="3"
          bind="tcp://0.0.0.0:63631"
          zkAddress="192.168.1.81:2181,192.168.1.82:2182,192.168.1.83:2183"
          hostname="edu-mq-01"
          zkPath="/activemq2/leveldb-stores"
    />
  </persistenceAdapter>
</broker>
```

Node-02 中的持久化配置:

```
<broker xmlns="http://activemq.apache.org/schema/core" brokerName="DubboEdu2" dataDirectory="${activemq.data}">
  <persistenceAdapter>
    <!-- kahaDB directory="${activemq.data}/kahadb"/ -->
    <replicatedLevelDB
          directory="${activemq.data}/leveldb"
          replicas="3"
          bind="tcp://0.0.0.0:63632"
          zkAddress="192.168.1.81:2181,192.168.1.82:2182,192.168.1.83:2183"
          hostname="edu-mq-01"
          zkPath="/activemq2/leveldb-stores"
     />
  </persistenceAdapter>
</broker>
```

Node-03 中的持久化配置:

```
<broker xmlns="http://activemq.apache.org/schema/core" brokerName="DubboEdu2" dataDirectory="${activemq.data}">
```

```
  <persistenceAdapter>
      <!-- kahaDB directory="${activemq.data}/kahadb"/ -->
      <replicatedLevelDB
            directory="${activemq.data}/leveldb"
            replicas="3"
            bind="tcp://0.0.0.0:63633"
            zkAddress="192.168.1.81:2181,192.168.1.82:2182,192.168.1.83:2183"
            hostname="edu-mq-01"
            zkPath="/activemq2/leveldb-stores"
      />
  </persistenceAdapter>
</broker>
```

修改各节点的消息端口（注意，避免端口冲突）：

Node-01 中的消息端口配置：

```
<transportConnectors>
   <!-- DOS protection, limit concurrent connections to 1000 and frame size to 100MB -->
   <transportConnector name="openwire" uri="tcp://0.0.0.0:53531?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="amqp" uri="amqp://0.0.0.0:5672?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="stomp" uri="stomp://0.0.0.0:61613?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="mqtt" uri="mqtt://0.0.0.0:1883?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="ws" uri="ws://0.0.0.0:61614?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
</transportConnectors>
```

Node-02 中的消息端口配置：

```
<transportConnectors>
   <!-- DOS protection, limit concurrent connections to 1000 and frame size to 100MB -->
   <transportConnector name="openwire" uri="tcp://0.0.0.0:53532?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="amqp" uri="amqp://0.0.0.0:5672?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="stomp" uri="stomp://0.0.0.0:61613?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="mqtt" uri="mqtt://0.0.0.0:1883?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="ws" uri="ws://0.0.0.0:61614?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
</transportConnectors>
```

Node-03 中的消息端口配置：

```
<transportConnectors>
   <!-- DOS protection, limit concurrent connections to 1000 and frame size to 100MB -->
   <transportConnector name="openwire" uri="tcp://0.0.0.0:53533?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="amqp" uri="amqp://0.0.0.0:5672?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="stomp" uri="stomp://0.0.0.0:61613?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="mqtt" uri="mqtt://0.0.0.0:1883?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
   <transportConnector name="ws" uri="ws://0.0.0.0:61614?maximumConnections=1000&amp;wireFormat.maxFrameSize=104857600"/>
</transportConnectors>
```

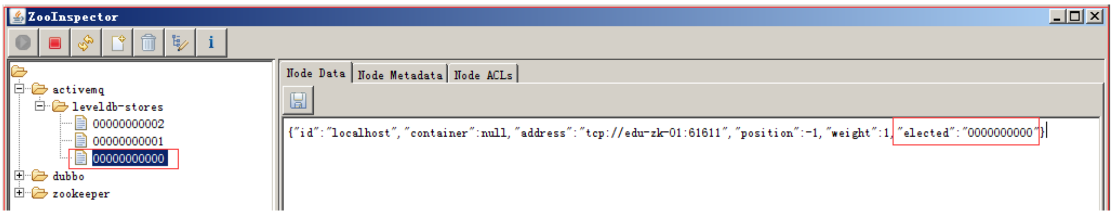
7、按顺序启动 3 个 ActiveMQ 节点：

```
$ /home/wusc/activemq/node-01/bin/activemq start
$ /home/wusc/activemq/node-02/bin/activemq start
$ /home/wusc/activemq/node-03/bin/activemq start
```

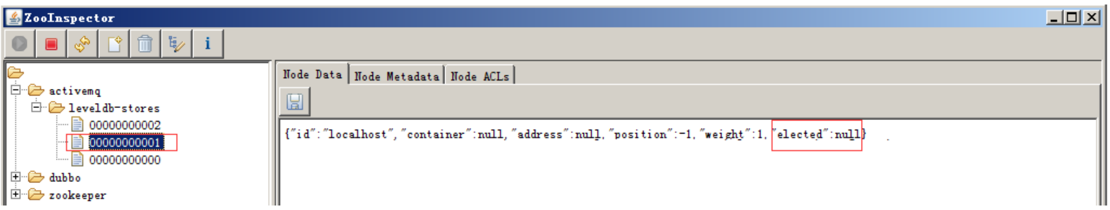
8、集群的节点状态分析：

集群启动后对 ZooKeeper 数据的抓图，可以看到 ActiveMQ 的有 3 个节点，分别是 00000000000，00000000001，00000000002。

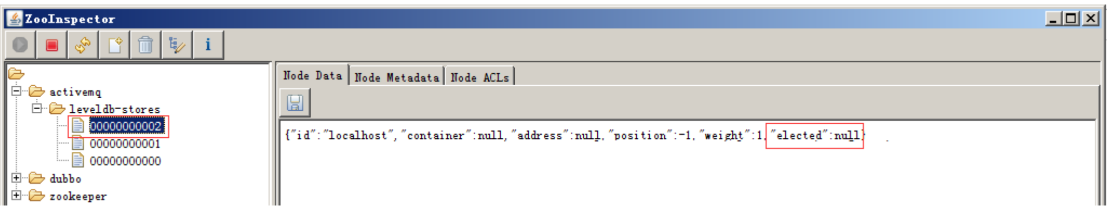
以下第一张图展现了 00000000000 的值，可以看到 elected 的值是不为空，说明这个节点是 Master，其他两个节点是 Slave。



```
{"id":"localhost","container":null,"address":"tcp://edu-zk-01:61611","position":-1,"weight":1,"elected":"0000000000"}
```



```
{"id":"localhost","container":null,"address":null,"position":-1,"weight":1,"elected":null}
```



```
{"id":"localhost","container":null,"address":null,"position":-1,"weight":1,"elected":null}
```

9、集群可用性测试（配置和测试代码，请看视频）：

ActiveMQ 的客户端只能访问 Master 的 Broker，其他处于 Slave 的 Broker 不能访问。所以客户端连接 Broker 应该使用 failover 协议。

```
failover:(tcp://192.168.1.101:53531,tcp://192.168.1.101:53532,tcp://192.168.1.101:53533)?randomize=false
```

10、集群高可用测试：

当一个 ActiveMQ 节点挂掉，或者一个 ZooKeeper 节点挂掉，ActiveMQ 服务依然正常运转。如果仅剩一个 ActiveMQ 节点，因为不能选举 Master，ActiveMQ 不能正常运转；同样的，如果 ZooKeeper 仅剩一个节点活动，不管 ActiveMQ 各节点是否存活，ActiveMQ 也不能正常提供服务。

11、设置开机启动：

```
# vi /etc/rc.local
su - wusc -c '/home/wusc/activemq/node-01/bin/activemq start'
su - wusc -c '/home/wusc/activemq/node-02/bin/activemq start'
su - wusc -c '/home/wusc/activemq/node-03/bin/activemq start'
```