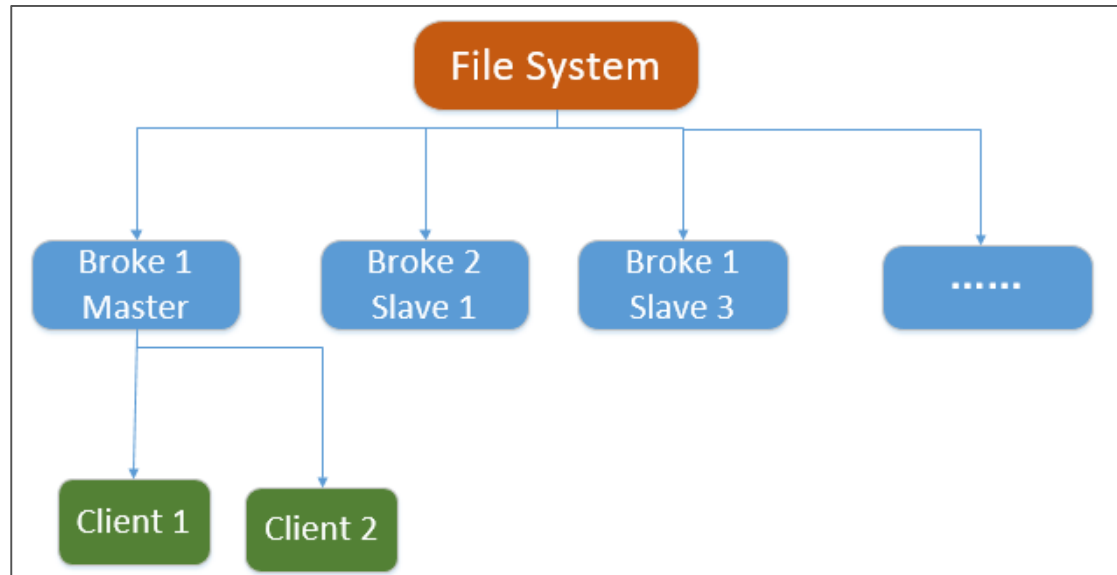


一、基于共享文件

1.1 拓扑图



基于共享文件的集群方式，需要使用分布式文件系统，比如 SAN 等，消息的持久化只保留一份，不用担心数据不全等问题。

使用 NFS 或本地 file system 等也可实现功能，但生产中绝不可以这样做，因为存储本身也是单节点故障的，不过作为测试，可以这样使用。

1.2 测试规划

为了方便测试，这是使用同一台机器启两个程序，为了便于区分管理口和服务口都配置成不相同，其实也可以配置成相同，反正不会同时被监听。

OS: CentOS 7.5 x64

JDK: 1.8.0_171

ActiveMQ: 5.15.4

IP: 10.1.5.201

安装目录:

master: /data/mq-cluster/mq-61616

slave: /data/mq-cluster/mq-61617

data : /data/mq-cluster/data

规划的端口:

master: 0.0.0.0:8161/61616

slave : 0.0.0.0:8162/61617

目录结构:

```

[root@hadoop01 mq-cluster]# tree -L 2
.
├── data
│   ├── db-1.log
│   ├── db.data
│   ├── db.redo
│   └── lock
├── mq-61616
│   ├── activemq-all-5.15.4.jar
│   ├── bin
│   ├── conf
│   ├── data
│   ├── docs
│   ├── examples
│   ├── lib
│   ├── LICENSE
│   ├── NOTICE
│   ├── README.txt
│   ├── tmp
│   ├── webapps
│   └── webapps-demo
└── mq-61617
    ├── activemq-all-5.15.4.jar
    ├── bin
    ├── conf
    ├── data
    ├── docs
    ├── examples
    ├── lib
    ├── LICENSE
    ├── NOTICE
    ├── README.txt
    ├── tmp
    ├── webapps
    └── webapps-demo

21 directories, 12 files
[root@hadoop01 mq-cluster]# ls
data  mq-61616  mq-61617

```

1.3 下载和部署

下载地址: <http://archive.apache.org/dist/activemq/>

选择合适的版本, 这里选择最新版 (20180803)

wget <http://archive.apache.org/dist/activemq/5.15.4/apache-activemq-5.15.4-bin.tar.gz>

```
tar xf apache-activemq-5.15.4-bin.tar.gz
```

```
mkdir -pv /data/mq-cluster/data
```

```
cp -r apache-activemq-5.15.4 /data/mq-cluster/mq-61616
```

```
cp -r apache-activemq-5.15.4 /data/mq-cluster/mq-61617
```

1.4 配置 JDK

下载地址: <http://www.oracle.com/technetwork/cn/java/archive-139210-zhs.html>

找到自己需要的版本, 上传到服务器

```
tar xf jdk-8u171-linux-x64.tar -C /usr/local
```

```
mv /usr/local/ jdk1.8.0_171 /usr/local/jdk #也可以创建软连接, 看个人习惯
保证/etc/profile 中有以下内容:
export JAVA_HOME=/usr/local/jdk
export PATH=$JAVA_HOME/bin:$PATH      #$JAVA_HOME/bin 一定要放到$PATH 前面,
防止识别到系统环境中其他版本的 jdk
source /etc/profile 使配置生效
java -version 查看是否配置成功
```

1.5 修改配置文件

1.5.1 activemq.xml

```
/data/mq-cluster/mq-61616/conf/activemq.xml
/data/mq-cluster/mq-61617/conf/activemq.xml
两个文件都要修改。
修改
<persistenceAdapter>
    <kahaDB directory="${activemq.data}/kahadb"/>
</persistenceAdapter>
为
<persistenceAdapter>
    <kahaDB directory="/data/mq-cluster/data"/>
</persistenceAdapter>
```

1.5.2 activemq.xml

```
修改/data/mq-cluster/mq-61617/conf/activemq.xml
中
<transportConnector name="openwire"
uri="tcp://0.0.0.0:61616?maximumConnections=1000&wireFormat.maxFrameSize=10
4857600"/>
为
<transportConnector name="openwire"
uri="tcp://0.0.0.0:61617?maximumConnections=1000&wireFormat.maxFrameSize=10
4857600"/>
如果是不同机器, 也可以不修改。
```

1.5.3 jetty.xml

另外修改/data/mq-cluster/mq-61617/conf/jetty.xml 中的管理端口为 6162
把

```
<property name="port" value="8161"/>
```

修改为

```
<property name="port" value="8162"/>
```

备注:其实这一步不修改也可以的,反正 master 和 slave 不会同时启动,只是为了便于观察,这里改成不相同的。

1.6 启动集群

启动 mq 集群

```
master: /data/mq-cluster/mq-61616/bin/activemq start
```

```
slave : /data/mq-cluster/mq-61617/bin/activemq start
```

备注:可以使用/data/mq-cluster/mq-61616/bin/activemq console 启动,实时观察输出结果,否则只能到/data/mq-cluster/mq-61616/data/activemq.log 日志中查看

1.7 测试

1.7.1 观察主从切换

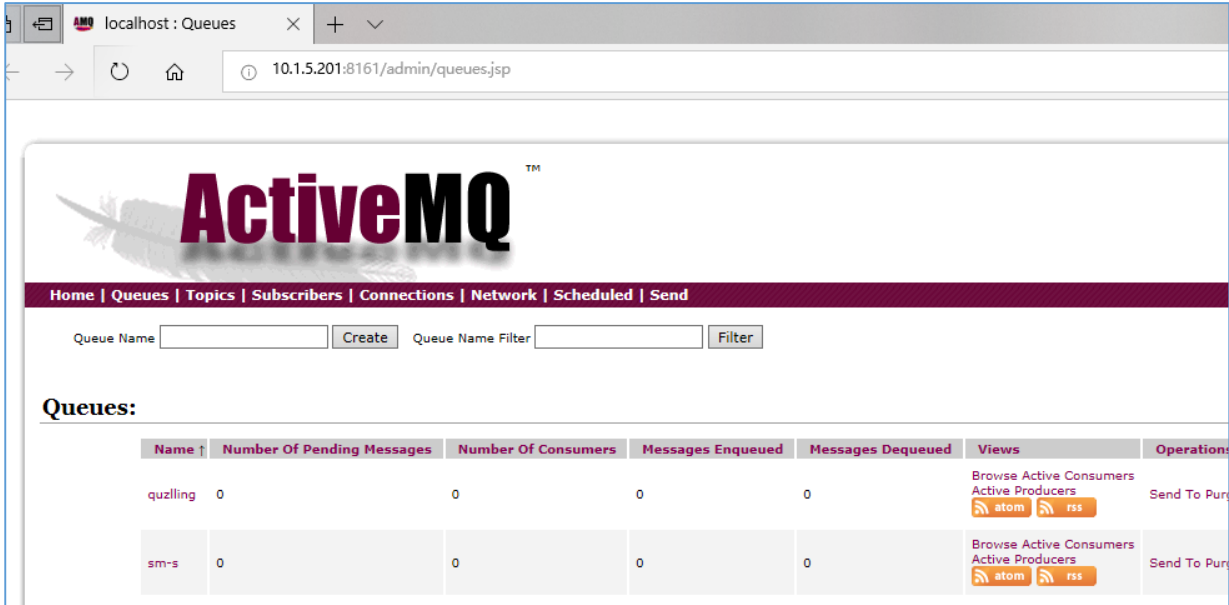
先启动的 61616 获得了文件锁,端口全部监听,而 61617 只启动了一个随机端口 31926,处于等待状态

直接 kill -9 杀掉 master,在极短的时间内,经观察小于 10s 内,slave 的 61617 即启动了起来,相应端口也都起来了,此时 slave 变成了 master。

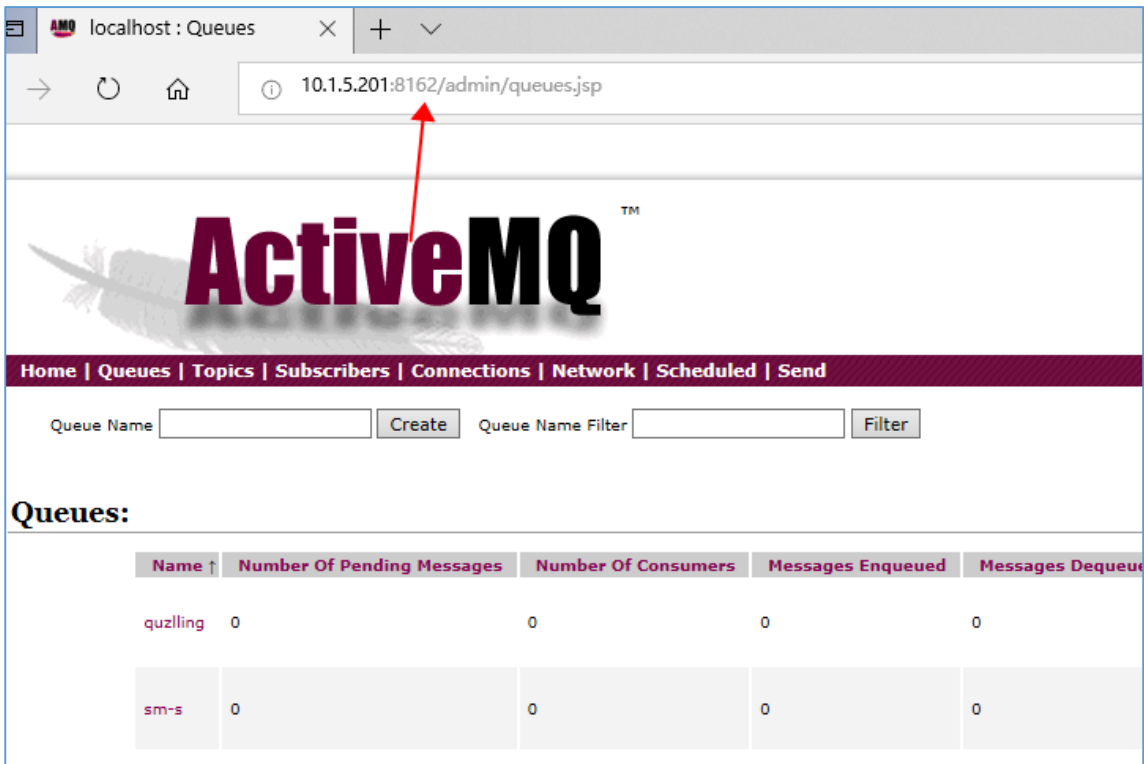
```
[root@hadoop01 mq-cluster]# jps
11728 activemq.jar
11835 jps
11822 activemq.jar
[root@hadoop01 mq-cluster]#
[root@hadoop01 mq-cluster]#
[root@hadoop01 mq-cluster]#
[root@hadoop01 mq-cluster]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1032/sshd
tcp        0      0 :::22                  :::*                    LISTEN      1032/sshd
tcp        0      0 :::58041                :::*                    LISTEN      11728/java
tcp        0      0 :::1883                 :::*                    LISTEN      11728/java
tcp        0      0 :::32926                :::*                    LISTEN      11822/java
tcp        0      0 :::8161                 :::*                    LISTEN      11728/java
tcp        0      0 :::5672                 :::*                    LISTEN      11728/java
tcp        0      0 :::873                  :::*                    LISTEN      1041/xinetd
tcp        0      0 :::61613                :::*                    LISTEN      11728/java
tcp        0      0 :::61614                :::*                    LISTEN      11728/java
tcp        0      0 :::61616                :::*                    LISTEN      11728/java
[root@hadoop01 mq-cluster]#
[root@hadoop01 mq-cluster]# kill -9 11728
[root@hadoop01 mq-cluster]# data
-bash: data: command not found
[root@hadoop01 mq-cluster]# date
2018年 08月 02日 星期四 14:40:36 CST
[root@hadoop01 mq-cluster]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1032/sshd
tcp        0      0 :::61617                :::*                    LISTEN      11822/java
tcp        0      0 :::22                   :::*                    LISTEN      1032/sshd
tcp        0      0 :::1883                 :::*                    LISTEN      11822/java
tcp        0      0 :::32926                :::*                    LISTEN      11822/java
tcp        0      0 :::5672                 :::*                    LISTEN      11822/java
tcp        0      0 :::873                  :::*                    LISTEN      1041/xinetd
tcp        0      0 :::61613                :::*                    LISTEN      11822/java
tcp        0      0 :::61614                :::*                    LISTEN      11822/java
[root@hadoop01 mq-cluster]# netstat -tnlp
```

1.7.2 主从切换数据是否丢失

在 61616 为主时，新增了两个消息



61617 成为 master 后，使用 8162 端口访问查看，发现队列仍然存在。



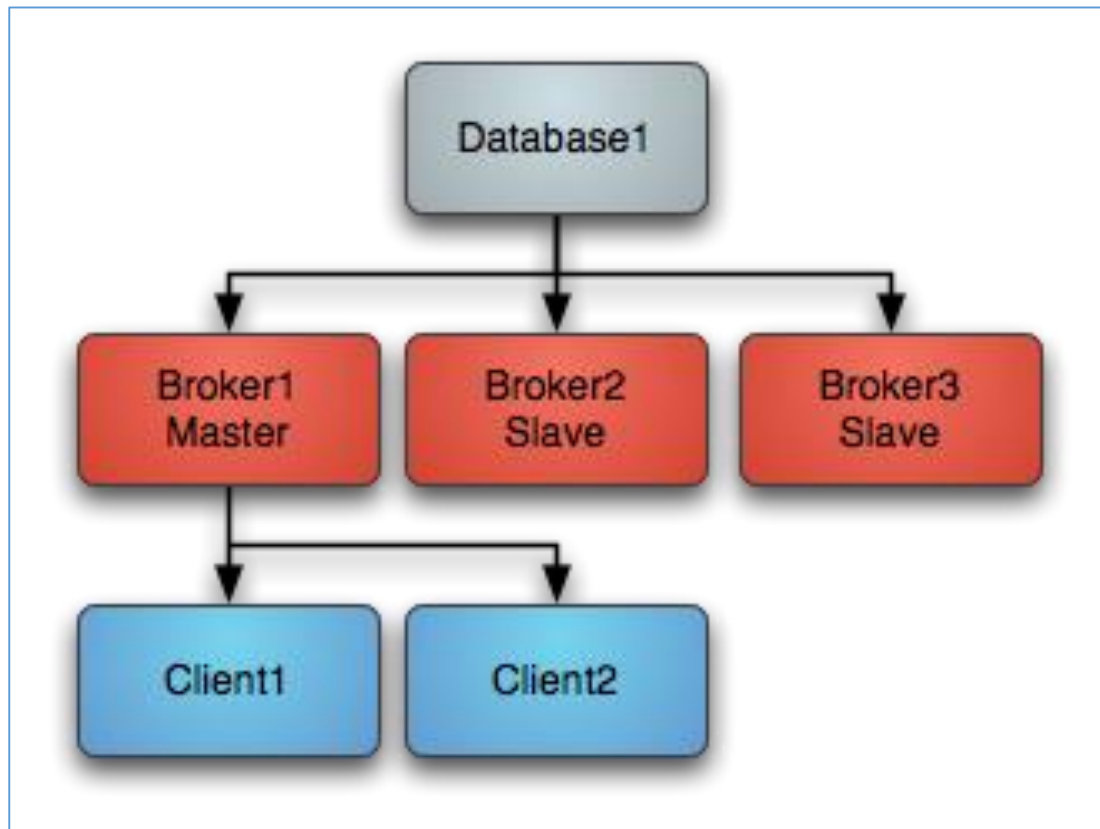
1.7.3 后续

当再次启动 61616 后，61616 处于 slave 状态，只启动了一个随机的端口 36886，61617 仍然是 master，因为文件锁还未释放。

```
root@hadoop01 mq-cluster]# jps
1892 Jps
1822 activemq.jar
root@hadoop01 mq-cluster]#
root@hadoop01 mq-cluster]# ./mq-61616/bin/activemq start
INFO: Loading '/data/mq-cluster/mq-61616/bin/env'
INFO: Using java '/usr/local/jdk/bin/java'
INFO: Starting - inspect logfiles specified in logging.properties and log4j.properties to get details
INFO: pidfile created: '/data/mq-cluster/mq-61616/data/activemq.pid' (pid '11952')
root@hadoop01 mq-cluster]# jps
1952 activemq.jar
1966 Jps
1822 activemq.jar
root@hadoop01 mq-cluster]# !netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1032/sshd
tcp        0      0 :::61617               :::*                    LISTEN      11822/java
tcp        0      0 :::36886               :::*                    LISTEN      11952/java
tcp        0      0 :::22                  :::*                    LISTEN      1032/sshd
tcp        0      0 :::1883                :::*                    LISTEN      11822/java
tcp        0      0 :::32926               :::*                    LISTEN      11822/java
tcp        0      0 :::8162                :::*                    LISTEN      11822/java
tcp        0      0 :::5672                :::*                    LISTEN      11822/java
tcp        0      0 :::873                 :::*                    LISTEN      1041/xinetd
tcp        0      0 :::61613               :::*                    LISTEN      11822/java
tcp        0      0 :::61614               :::*                    LISTEN      11822/java
root@hadoop01 mq-cluster]#
```

二、基于共享数据库

2.1 拓扑图



基于共享数据库的集群方式，与基于文件共享的方式类似，只是使用数据库进行持久化存储，当然也包括共同争抢数据库锁。AMQ 支持常见的多种数据库，比如 mysql/oracle/psql 等。这里使用 mysql 做演示。

优点：

- 实现高可用和数据安全

- 简单灵活，2 台节点就可以实现高可用

缺点：

- 稳定性依赖数据库

- 性能依赖数据库

2.2 测试规划

为了方便测试，这是使用同一台机器启两个程序，为了便于区分管理口和服务口都配置成不相同，其实也可以配置成相同，反正不会同时被监听。

OS: CentOS 7.5 x64

JDK: 1.8.0_171

ActiveMQ: 5.15.4

IP: 10.1.5.201

安装目录:

master: /data/mq-cluster/mq-61616

slave : /data/mq-cluster/mq-61617

规划的端口:

master: 0.0.0.0:8161/61616

slave: 0.0.0.0:8162/61617

目录结构:

```
[root@hadoop01 mq-cluster]# tree -L 2
.
├── mq-61616
│   ├── activemq-all-5.15.4.jar
│   ├── bin
│   ├── conf
│   ├── data
│   ├── docs
│   ├── examples
│   ├── lib
│   ├── LICENSE
│   ├── NOTICE
│   ├── README.txt
│   ├── tmp
│   ├── webapps
│   └── webapps-demo
└── mq-61617
    ├── activemq-all-5.15.4.jar
    ├── bin
    ├── conf
    ├── data
    ├── docs
    ├── examples
    ├── lib
    ├── LICENSE
    ├── NOTICE
    ├── README.txt
    ├── tmp
    ├── webapps
    └── webapps-demo

20 directories, 8 files
[root@hadoop01 mq-cluster]# ls
mq-61616 mq-61617
[root@hadoop01 mq-cluster]#
```

2.3 下载和部署

下载地址: <http://archive.apache.org/dist/activemq/>

选择合适的版本, 这里选择最新版 (20180803)

wget <http://archive.apache.org/dist/activemq/5.15.4/apache-activemq-5.15.4-bin.tar.gz>

```
tar xf apache-activemq-5.15.4-bin.tar.gz
```

```
mkdir -pv /data/mq-cluster/data
```

```
cp -r apache-activemq-5.15.4 /data/mq-cluster/mq-61616
```

```
cp -r apache-activemq-5.15.4 /data/mq-cluster/mq-61617
```


2.4 配置 JDK

下载地址: <http://www.oracle.com/technetwork/cn/java/archive-139210-zhs.html>

找到自己需要的版本, 上传到服务器

```
tar xf jdk-8u171-linux-x64.tar -C /usr/local
```

```
mv /usr/local/jdk1.8.0_171 /usr/local/jdk #也可以创建软连接, 看个人习惯
```

保证/etc/profile 中有以下内容:

```
export JAVA_HOME=/usr/local/jdk
```

```
export PATH=$JAVA_HOME/bin:$PATH # $JAVA_HOME/bin 一定要放到$PATH 前面,
```

防止识别到系统环境中其他版本的 jdk

```
source /etc/profile 使配置生效
```

```
java -version 查看是否配置成功
```

2.5 添加驱动包

到下面链接下载 mysql 的 JDBC 的驱动包:

<https://downloads.mysql.com/archives/c-j/>

比如:

```
wget https://cdn.mysql.com/archives/mysql-connector-java-5.1/mysql-connector-java-5.1.45.tar.gz
```

解压后, 将 mysql-connector-java-5.1.45-bin.jar 拷贝两份放到下面路径下

```
/data/mq-cluster/mq-61616/lib
```

```
/data/mq-cluster/mq-61617/lib
```

2.6 准备 mysql 账号密码等

创建 msyql 的可访问的账号密码

```
mysql>grant all privileges on *.* to 'activemq'@'%' identified by "activemq";
```

```
mysql>flush privileges;
```

```
mysql>create database activemq;
```

注意: 这里的信息, 要和下面的配置信息一致

2.7 修改配置文件

2.7.1 activemq.xml

```
/data/mq-cluster/mq-61616/conf/activemq.xml
```

```
/data/mq-cluster/mq-61617/conf/activemq.xml
```

两个文件都要修改。

修改

```

<persistenceAdapter>
    <kahaDB directory="${activemq.data}/kahadb"/>
</persistenceAdapter>
为
<persistenceAdapter>
    <jdbcPersistenceAdapter dataSource="#my-ds"/>

</persistenceAdapter>

```

2.7.2 activemq.xml

/data/mq-cluster/mq-61616/conf/activemq.xml
/data/mq-cluster/mq-61616/conf/activemq.xml
两个文件都要修改。

</broker>的后面添加如下红色部分内容，注意 id 要与上一步中配置要相同。添加完如下：

```

</broker>
<bean id="my-ds" class="org.apache.commons.dbcp2.BasicDataSource" destroy-
method="close">
<property name="driverClassName" value="com.mysql.jdbc.Driver"/>
<property name="url"
value="jdbc:mysql://127.0.0.1:3306/activemq?relaxAutoCommit=true"/>
<property name="username" value="activemq"/>
<property name="password" value="activemq"/>
<property name="poolPreparedStatements" value="true"/>
</bean>
<import resource="jetty.xml"/>
</beans>

```

2.7.3 active.xml

修改/data/mq-cluster/mq-61617/conf/activemq.xml
中

```

<transportConnector name="openwire"
uri="tcp://0.0.0.0:61616?maximumConnections=1000&wireFormat.maxFrameSize=10
4857600"/>
为
<transportConnector name="openwire"
uri="tcp://0.0.0.0:61617?maximumConnections=1000&wireFormat.maxFrameSize=10
4857600"/>

```

如果是不同机器，也可以不修改。

2.7.4 jetty.xml

修改/data/mq-cluster/mq-61617/conf/jetty.xml 中的管理端口为 6162

即修改

```
<property name="port" value="8161"/>
```

为

```
<property name="port" value="8162"/>
```

备注:其实这一步不修改也可以的,反正 master 和 slave 不会同时启动,只是为了便于观察,这里改成不相同的。

其他注意事项:

如果启动失败，使用 `./bin/activemq console` 启动，使日志在前端输出，报以下错误：

Cannot find class [org.apache.commons.dbcp.BasicDataSource]

```
[root@hadoop01 mq-61616]# ./bin/activemq console
INFO: Loading '/data/mq-cluster/mq-61616/bin/env
INFO: Using java '/usr/local/jdk/bin/java'
INFO: Starting in foreground, this is just for debugging purposes (stop process by pressing CTRL+C)
INFO: Creating pidfile '/data/mq-cluster/mq-61616/data/activemq.pid
Java Runtime: Oracle Corporation 1.8.0_171 /usr/local/jdk/jre
Heap sizes: current=62976k free=58699k max=932352k
JVM args: -Xms64M -Xmx1G -Djava.util.logging.config.file=logging.properties -Djava.security.auth.login.config=/data/
n.management.jmxremote -Djava.awt.headless=true -Djava.io.tmpdir=/data/mq-cluster/mq-61616/tmp -Dactivemq.classpath=/da
q-61616/.../lib/: -Dactivemq.home=/data/mq-cluster/mq-61616/ -Dactivemq.base=/data/mq-cluster/mq-61616/ -Dactivemq.conf=
=/data/mq-cluster/mq-61616/data
Extensions classpath:
[/data/mq-cluster/mq-61616/lib,/data/mq-cluster/mq-61616/lib/camel,/data/mq-cluster/mq-61616/lib/optional,/data/mq-clu
/lib/extra]
ACTIVEMQ_HOME: /data/mq-cluster/mq-61616
ACTIVEMQ_BASE: /data/mq-cluster/mq-61616
ACTIVEMQ_CONF: /data/mq-cluster/mq-61616/conf
ACTIVEMQ_DATA: /data/mq-cluster/mq-61616/data
Loading message broker from: xbean:activemq.xml
INFO | Refreshing org.apache.activemq.xbean.XBeanBrokerFactory$1@56ef9176: startup date [Fri Aug 03 10:51:17 CST 2018];
WARN | Exception encountered during context initialization - cancelling refresh attempt: org.springframework.beans.factory
with name 'org.apache.activemq.xbean.XBeanBrokerService#0' defined in class path resource [activemq.xml]; Cannot create
.apache.activemq.store.jdbc.JDBCPersistenceAdapter] while setting bean property 'persistenceAdapter'; nested exception is:
nException: Error creating bean with name '(inner bean)#17497425' defined in class path resource [activemq.xml]: Cannot
bean property 'datasource'; nested exception is org.springframework.beans.factory.CannotLoadBeanClassException: Cannot
urce] for bean with name 'my-ds' defined in class path resource [activemq.xml]; nested exception is java.lang.ClassNotFo
Source
ERROR Failed to load: class path resource [activemq.xml], reason: Error creating bean with name 'org.apache.activemq.x
h resource [activemq.xml]: Cannot create inner bean '(inner bean)#17497425' of type [org.apache.activemq.store.jdbc.JDBC
'persistenceAdapter'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean
ass path resource [activemq.xml]: Cannot resolve reference to bean 'my-ds' while setting bean property 'datasource'; nes
ory.CannotLoadBeanClassException: Cannot find class [org.apache.commons.dbcp.BasicDataSource] for bean with name 'my-ds'
nested exception is java.lang.ClassNotFoundException: org.apache.commons.dbcp.BasicDataSource
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'org.apache.activemq.xbean.XBeanB
[activemq.xml]: Cannot create inner bean '(inner bean)#17497425' of type [org.apache.activemq.store.jdbc.JDBCPersiste
```

这个错误表面未找到 `org.apache.commons.dbcp.BasicDataSource` 类，因为高版本的 AMQ 提供了不同版本 `org.apache.commons.dbcp.BasicDataSource` 类，见下图，

```
[root@hadoop01 mq-61616]# ll lib/optional/commons-dbc2-2.1.1.jar
-rw-r--r-- 1 root root 174167 2月  2 2016 lib/optional/commons-dbc2-2.1.1.jar
[root@hadoop01 mq-61616]#
[root@hadoop01 mq-61616]#
```

加节方法:

修改

```
<bean id="my-ds" class="org.apache.commons.dbcp.BasicDataSource" destroy-
method="close">
```

为

```
<bean id="my-ds" class="org.apache.commons.dbcp2.BasicDataSource" destroy-
method="close">
```

网上很多教程没说这个，这是一个大坑

2.8 启动集群

启动 mq 集群

master: /data/mq-cluster/mq-61616/bin/activemq start

slave : /data/mq-cluster/mq-61617/bin/activemq start

备注:可以使用/data/mq-cluster/mq-61616/bin/activemq console 启动, 实时观察输出结果, 否则只能到/data/mq-cluster/mq-61616/data/activemq.log 日志中查看

2.9 测试

2.9.1 观察主从切换

```
[root@hadoop01 mq-cluster]# jps
22035 jps
22019 activemq.jar
21798 activemq.jar
[root@hadoop01 mq-cluster]# !net
netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              *:*                     LISTEN      1032/sshd
tcp        0      0 0.0.0.0:22              *:*                     LISTEN      1032/sshd
tcp        0      0 0.0.0.0:51099           *:*                     LISTEN      22019/java
tcp        0      0 0.0.0.0:1883            *:*                     LISTEN      21798/java
tcp        0      0 0.0.0.0:47520           *:*                     LISTEN      21798/java
tcp        0      0 0.0.0.0:8161            *:*                     LISTEN      21798/java
tcp        0      0 0.0.0.0:5672            *:*                     LISTEN      21798/java
tcp        0      0 0.0.0.0:873             *:*                     LISTEN      1041/xinetd
tcp        0      0 0.0.0.0:3306             *:*                     LISTEN      12824/mysqld
tcp        0      0 0.0.0.0:61613            *:*                     LISTEN      21798/java
tcp        0      0 0.0.0.0:61614            *:*                     LISTEN      21798/java
tcp        0      0 0.0.0.0:61616            *:*                     LISTEN      21798/java
[root@hadoop01 mq-cluster]#
[root@hadoop01 mq-cluster]#
[root@hadoop01 mq-cluster]# kill -9 21798
[root@hadoop01 mq-cluster]#
[root@hadoop01 mq-cluster]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              *:*                     LISTEN      1032/sshd
tcp        0      0 0.0.0.0:61617           *:*                     LISTEN      22019/java
tcp        0      0 0.0.0.0:22              *:*                     LISTEN      1032/sshd
tcp        0      0 0.0.0.0:1883            *:*                     LISTEN      22019/java
tcp        0      0 0.0.0.0:51099           *:*                     LISTEN      22019/java
tcp        0      0 0.0.0.0:8162            *:*                     LISTEN      22019/java
tcp        0      0 0.0.0.0:5672            *:*                     LISTEN      22019/java
tcp        0      0 0.0.0.0:873             *:*                     LISTEN      1041/xinetd
tcp        0      0 0.0.0.0:3306             *:*                     LISTEN      12824/mysqld
tcp        0      0 0.0.0.0:61613            *:*                     LISTEN      22019/java
tcp        0      0 0.0.0.0:61614            *:*                     LISTEN      22019/java
[root@hadoop01 mq-cluster]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
```

先启动的 61616 获得了数据库锁, 端口全部监听, 而 61617 只启动了一个随机端口 51099, 处于等待状态

直接 kill -9 杀掉 master, 在极短的时间内, 经观察小于 10s 内, slave 的 61617 即启动了起来, 相应端口也都起来了, 此时 slave 变成了 master。

2.9.2 观察数据库情况

自动创建三张表

```
mysql> use activemq;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_activemq |
+-----+
| activemq_acks       |
| activemq_lock       |
| activemq_msgs       |
+-----+
3 rows in set (0.00 sec)
```

activemq_acks: 用于存储订阅关系。如果是持久化 Topic，订阅者和服务器的订阅关系在这个表保存。

activemq_lock: 在集群环境中才有用，只有一个 Broker 可以获得消息，称为 Master Broker，其他的只能作为备份，等待 Master Broker 不可用，才可能成为下一个 Master Broker。这个表用于记录哪个 Broker 是当前的 Master Broker。

activemq_msgs: 用于存储消息，Queue 和 Topic 都存储在这个表中。
比如下图中持久化到 mysql 的消息。

ID	CONTAINER	MSGID_PROD	MSGID_SEQ	EXPIRATION	MSG	PRIORITY	XID
5	queue://q.open.api.o	ID:DESKTOP-I3RIE'	1	0 (BLOB)	0 (Null)		
13	queue://q.open.api.o	ID:DESKTOP-I3RIE'	1	0 (BLOB)	0 (Null)		
14	queue://q.open.api.o	ID:DESKTOP-I3RIE'	1	0 (BLOB)	0 (Null)		

2.9.3 观察主从切换数据是否丢失

在 61617 为主时，使用 java 程序（找研发添加的）新增了 2 个队列，3 条消息，使用 8162 端口访问查看如下：

ActiveMQ™

Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send

Queue Name Create Queue Name Filter Filter

Queues:

Name ↑	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued
q.open.api.order.invite	2	0	2	0
q.open.api.order.invoice	1	0	2	1

且 mysql 中有对应的消息：

对象 activemq_msgs @activemq ...			
ID	CONTAINER	MSGID_PROD	MSGID_SEQ
5	queue://q.open.api.order.invite	ID:DESKTOP-I3RIE'	
13	queue://q.open.api.order.invoice	ID:DESKTOP-I3RIE'	
14	queue://q.open.api.order.invite	ID:DESKTOP-I3RIE'	

在 61616 为主时，使用 8161 端口访问查看如下：

10.1.5.201:8161/admin/queues.jsp

[Home](#) | [Queues](#) | [Topics](#) | [Subscribers](#) | [Connections](#) | [Network](#) | [Scheduled](#) | [Send](#)

Queue Name Queue Name Filter

Queues:

Name ↑	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Message Dequeued
q.open.api.order.invite	2	0	0	0
q.open.api.order.invoice	1	0	0	0

数据库中消息依然存在。

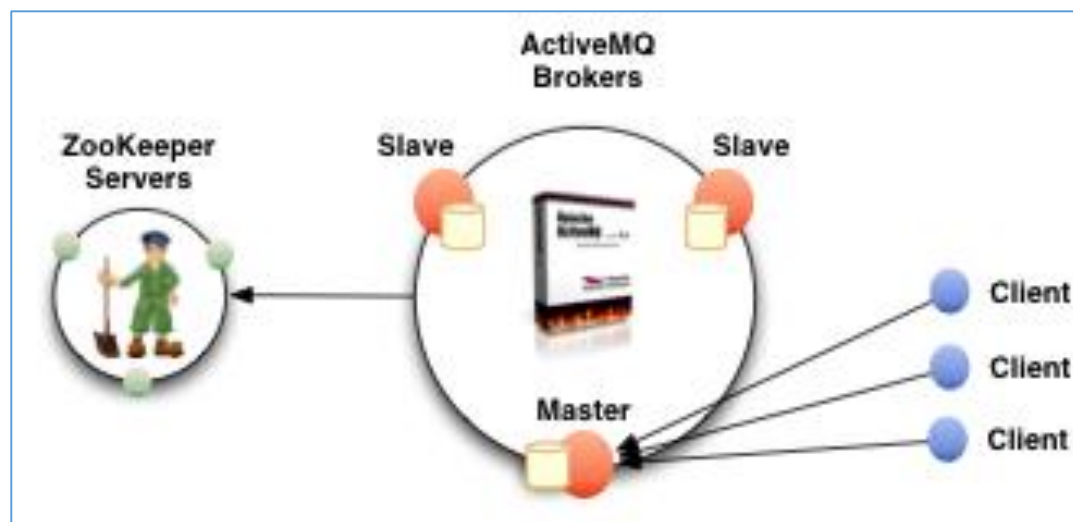
activemq_msgs @activemq ...							
ID	CONTAINER	MSGID_PROD	MSGID_SEQ	EXPIRATION	MSG	PRIORITY	XID
5	queue://q.open.api.order.invite	ID:DESKTOP-I3RIE'	1		0 (BLOB)		0 (Null)
13	queue://q.open.api.order.invoice	ID:DESKTOP-I3RIE'	1		0 (BLOB)		0 (Null)
14	queue://q.open.api.order.invite	ID:DESKTOP-I3RIE'	1		0 (BLOB)		0 (Null)

后续：

当再次启 61617 后，61616 仍处于 master 状态。

三、 基于 zookeeper

3.1 拓扑图



AMQ5.9 新推出的主从实现方式，基于 zookeeper 来选举出一个 master，其他节点自动作为 slave 实时同步消息。

因为有实时同步数据的 slave 的存在，master 不用担心数据丢失，所以 leveldb 会优先采用内存存储消息，异步同步到磁盘。所以该方式的 activeMQ 读写性能都最好，特别是写性能能够媲美非持久化消息。

优点：

- 实现高可用和数据安全
- 性能较好

缺点：

因为选举机制要超过半数，所以最少需要 3 台节点，才能实现高可用。

3.2 测试规划

为了方便测试，这是使用同一台机器启两个程序，为了便于区分管理口和服务口都配置成不相同，其实也可以配置成相同，反正不会同时被监听。

OS: CentOS 7.5 x64

JDK: 1.8.0_171

ActiveMQ: 5.15.4

IP: 10.1.5.201

安装目录：

- master: /data/mq-cluster/mq-61616
- slavel: /data/mq-cluster/mq-61617
- slave2: /data/mq-cluster/mq-61618

规划的端口：

- master: 0.0.0.0:8161/61616
- slavel: 0.0.0.0:8162/61617

slave2: 0.0.0.0:8163/61618

ZooKeeper:

10.1.5.203:2181

10.1.5.204:2181

10.1.5.205:2181

目录结构:

```
[root@hadoop01 mq-cluster]# tree -L 2
.
├── mq-61616
│   ├── activemq-all-5.15.4.jar
│   ├── bin
│   ├── conf
│   ├── data
│   ├── docs
│   ├── examples
│   ├── lib
│   ├── LICENSE
│   ├── NOTICE
│   ├── README.txt
│   ├── tmp
│   ├── webapps
│   └── webapps-demo
├── mq-61617
│   ├── activemq-all-5.15.4.jar
│   ├── bin
│   ├── conf
│   ├── data
│   ├── docs
│   ├── examples
│   ├── lib
│   ├── LICENSE
│   ├── NOTICE
│   ├── README.txt
│   ├── tmp
│   ├── webapps
│   └── webapps-demo
└── mq-61618
    ├── activemq-all-5.15.4.jar
    ├── bin
    ├── conf
    ├── data
    ├── docs
    ├── examples
    ├── lib
    ├── LICENSE
    ├── NOTICE
    ├── README.txt
    ├── tmp
    ├── webapps
    └── webapps-demo
```

```
30 directories, 12 files
[root@hadoop01 mq-cluster]# ls
mq-61616 mq-61617 mq-61618
```

3.3 下载和部署

下载地址: <http://archive.apache.org/dist/activemq/>

选择合适的版本, 这里选择最新版 (20180803)

wget <http://archive.apache.org/dist/activemq/5.15.4/apache-activemq-5.15.4-bin.tar.gz>

```
tar xf apache-activemq-5.15.4-bin.tar.gz
mkdir -pv /data/mq-cluster/data
cp -r apache-activemq-5.15.4 /data/mq-cluster/mq-61616
cp -r apache-activemq-5.15.4 /data/mq-cluster/mq-61617
```

3.4 配置 JDK

下载地址: <http://www.oracle.com/technetwork/cn/java/archive-139210-zhs.html>

找到自己需要的版本, 上传到服务器

```
tar xf jdk-8u171-linux-x64.tar -C /usr/local
```

```
mv /usr/local/jdk1.8.0_171 /usr/local/jdk #也可以创建软连接, 看个人习惯
```

保证/etc/profile 中有以下内容:

```
export JAVA_HOME=/usr/local/jdk
```

```
export PATH=$JAVA_HOME/bin:$PATH # $JAVA_HOME/bin 一定要放到$PATH 前面,
```

防止识别到系统环境中其他版本的 jdk

```
source /etc/profile 使配置生效
```

```
java -version 查看是否配置成功
```

3.5 修改配置文件

3.5.1 activemq.xml

```
/data/mq-cluster/mq-61616/conf/activemq.xml
```

```
/data/mq-cluster/mq-61617/conf/activemq.xml
```

```
/data/mq-cluster/mq-61618/conf/activemq.xml
```

三个文件都要修改。

修改

```
<persistenceAdapter>
```

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

```
</persistenceAdapter>
```

为

```
<persistenceAdapter>
```

```
  <replicatedLevelDB
```

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

```
    replicas="3"
```

```
    bind="tcp://0.0.0.0:0"
```

```
      zkAddress="10.1.5.203:2181, 10.1.5.204:2181, 10.1.5.205:2181"
```

```
      zkPassword=""
```

```
      hostname="10.1.5.201"
```

```

        sync="local_disk"
        zkPath="/activemq/leveldb-stores"
    />
</persistenceAdapter>

```

配置项解释:

directory:

消息持久化保存到服务器上的路径，如果不存在会自动创建。

replicas:

ActiveMQ 集群的节点个数

bind:

当这个节点成为 master 后，绑定的机器的地址与端口。此处 0.0.0.0:0 表示绑定到本机所有可用 IP，而端口是随机的。

zkAddress:

ZooKeeper 的 ip 和 port。如果是 ZooKeeper 集群的话，则用逗号隔开

zkSessionTimeout:

ActiveMQ 与 ZooKeeper 集群连接的会话超时时间

hostname:

本机的 IP 地址。部署到不同机器，这个配置要做对应变化

sync:

在消息被消费完成前，同步信息所存储的策略。多种策略用逗号隔开，优先选择较强的策略。比如 sync="local_mem,local_disk"，则选择 local_disk 策略，存储在本地硬盘。

zkPath:

ActiveMQ 在 ZooKeeper 集群上创建的 znode 节点的路径，也即是 ZooKeeper 选举信息交换的存储路径，注意，不是文件系统的路径。

3.5.2 active.xml

修改/data/mq-cluster/mq-61617/conf/activemq.xml

中

```

<transportConnector name="openwire"
uri="tcp://0.0.0.0:61616?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>

```

为

```

<transportConnector name="openwire"
uri="tcp://0.0.0.0:61617?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>

```

修改/data/mq-cluster/mq-61618/conf/activemq.xml

中

```
<transportConnector name="openwire"
uri="tcp://0.0.0.0:61616?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>
```

为

```
<transportConnector name="openwire"
uri="tcp://0.0.0.0:61618?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>
```

如果是不同机器，也可以不修改。

3.5.3 jetty.xml

修改/data/mq-cluster/mq-61617/conf/jetty.xml 中的管理端口为 6162
把

```
<property name="port" value="8161"/>
```

修改为

```
<property name="port" value="8162"/>
```

修改/data/mq-cluster/mq-61618/conf/jetty.xml 中的管理端口为 6163
把

```
<property name="port" value="8161"/>
```

修改为

```
<property name="port" value="8163"/>
```

备注:其实这一步不修改也可以的,反正 master 和 slave 不会同时启动,只是为了便于观察,这里改成不相同的。

3.6 启动集群

启动 mq 集群

master: /data/mq-cluster/mq-61616/bin/activemq start

slave1: /data/mq-cluster/mq-61617/bin/activemq start

slave2: /data/mq-cluster/mq-61618/bin/activemq start

备注:可以使用/data/mq-cluster/mq-61616/bin/activemq console 启动,实时观察输出结果,否则只能到/data/mq-cluster/mq-61616/data/activemq.log 日志中查看

3.7 测试

3.7.1 观察主从切换

先启动的 61616 成为了主，61616/8161 等端口全部监听，而 61617 只启动了一个随机端口 33924，61618 启动了另一个随机端口 34824。

直接 kill 30910 模拟 master 挂掉，在极短的时间内，经观察小于 10s 内，slave 的 61617 即启动了起来，相应端口 61617/8162 等也都起来了，此时 slave 变成了 master。

```
root@hadoop01 mq-cluster]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1032/sshd
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1032/sshd
tcp        0      0 0.0.0.0:1883           0.0.0.0:*               LISTEN      30910/java
tcp        0      0 0.0.0.0:38909          0.0.0.0:*               LISTEN      30910/java
tcp        0      0 0.0.0.0:8161           0.0.0.0:*               LISTEN      30910/java
tcp        0      0 0.0.0.0:33924          0.0.0.0:*               LISTEN      31345/java
tcp        0      0 0.0.0.0:5672           0.0.0.0:*               LISTEN      30910/java
tcp        0      0 0.0.0.0:34824          0.0.0.0:*               LISTEN      31133/java
tcp        0      0 0.0.0.0:873            0.0.0.0:*               LISTEN      1041/xinetd
tcp        0      0 0.0.0.0:3306           0.0.0.0:*               LISTEN      12824/mysqld
tcp        0      0 0.0.0.0:61613          0.0.0.0:*               LISTEN      30910/java
tcp        0      0 0.0.0.0:61614          0.0.0.0:*               LISTEN      30910/java
tcp        0      0 0.0.0.0:32911          0.0.0.0:*               LISTEN      30910/java
tcp        0      0 0.0.0.0:61616          0.0.0.0:*               LISTEN      30910/java
[root@hadoop01 mq-cluster]#
[root@hadoop01 mq-cluster]#
[root@hadoop01 mq-cluster]# jps
31345 activemq.jar
31368 jps
31133 activemq.jar
30910 activemq.jar
[root@hadoop01 mq-cluster]#
[root@hadoop01 mq-cluster]# kill 30910
[root@hadoop01 mq-cluster]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1032/sshd
tcp        0      0 0.0.0.0:61617          0.0.0.0:*               LISTEN      31133/java
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1032/sshd
tcp        0      0 0.0.0.0:1883           0.0.0.0:*               LISTEN      31133/java
tcp        0      0 0.0.0.0:44863          0.0.0.0:*               LISTEN      31133/java
tcp        0      0 0.0.0.0:8162           0.0.0.0:*               LISTEN      31133/java
tcp        0      0 0.0.0.0:33924          0.0.0.0:*               LISTEN      31345/java
tcp        0      0 0.0.0.0:5672           0.0.0.0:*               LISTEN      31133/java
tcp        0      0 0.0.0.0:34824          0.0.0.0:*               LISTEN      31133/java
tcp        0      0 0.0.0.0:873            0.0.0.0:*               LISTEN      1041/xinetd
tcp        0      0 0.0.0.0:3306           0.0.0.0:*               LISTEN      12824/mysqld
tcp        0      0 0.0.0.0:61613          0.0.0.0:*               LISTEN      31133/java
tcp        0      0 0.0.0.0:61614          0.0.0.0:*               LISTEN      31133/java
[root@hadoop01 mq-cluster]#
```

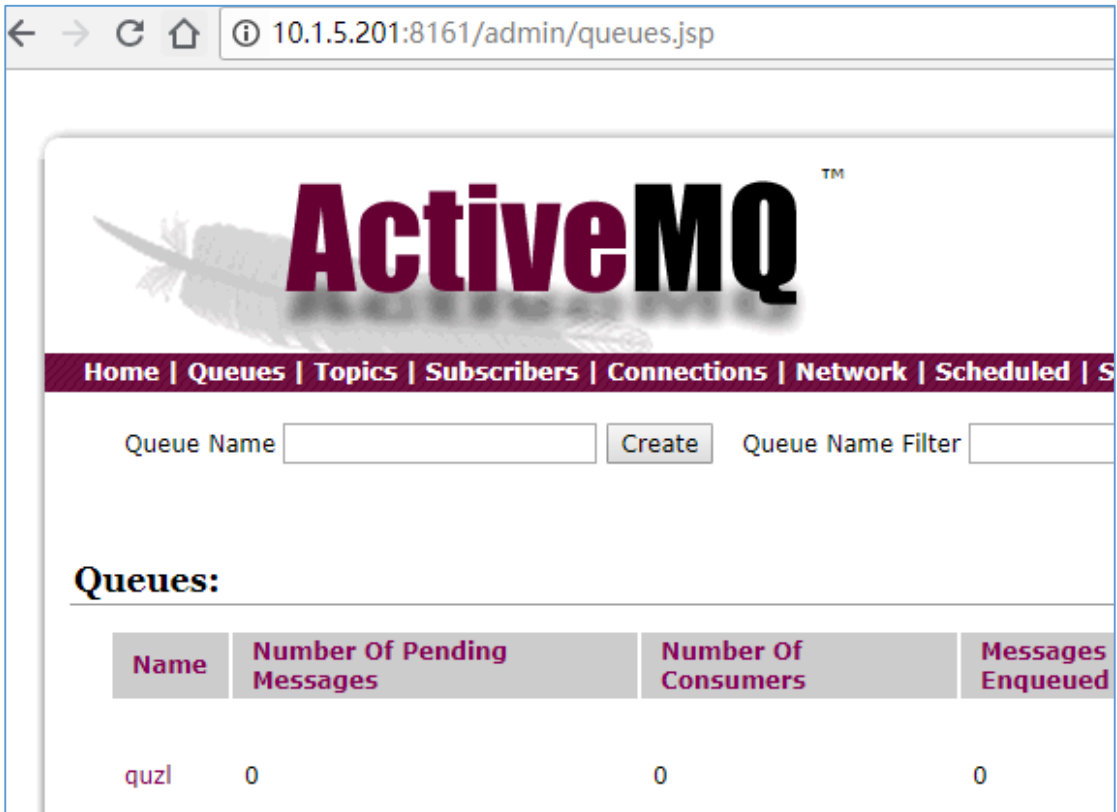
引申：3 个节点的集群，停止其中任意 2 个，集群将停止工作，如下图所示，具体原因，大家可以看官方文档，大致就是需要大于 $n/2$ 个节点在线进行投票才可以选出新的 master。

```
netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1032/sshd
tcp        0      0 0.0.0.0:59281          0.0.0.0:*               LISTEN      31470/java
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1032/sshd
tcp        0      0 0.0.0.0:1883           0.0.0.0:*               LISTEN      31470/java
tcp        0      0 0.0.0.0:8161           0.0.0.0:*               LISTEN      31470/java
tcp        0      0 0.0.0.0:56678          0.0.0.0:*               LISTEN      31927/java
tcp        0      0 0.0.0.0:5672           0.0.0.0:*               LISTEN      31470/java
tcp        0      0 0.0.0.0:873            0.0.0.0:*               LISTEN      1041/xinetd
tcp        0      0 0.0.0.0:57450          0.0.0.0:*               LISTEN      31470/java
tcp        0      0 0.0.0.0:3306           0.0.0.0:*               LISTEN      12824/mysqld
tcp        0      0 0.0.0.0:34307          0.0.0.0:*               LISTEN      31825/java
tcp        0      0 0.0.0.0:61613          0.0.0.0:*               LISTEN      31470/java
tcp        0      0 0.0.0.0:61614          0.0.0.0:*               LISTEN      31470/java
tcp        0      0 0.0.0.0:61616          0.0.0.0:*               LISTEN      31470/java
[root@hadoop01 mq-cluster]# jps
31825 activemq.jar
31927 activemq.jar
31962 jps
31470 activemq.jar
[root@hadoop01 mq-cluster]# kill 31825 31927
[root@hadoop01 mq-cluster]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1032/sshd
tcp        0      0 0.0.0.0:59281          0.0.0.0:*               LISTEN      31470/java
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1032/sshd
tcp        0      0 0.0.0.0:873            0.0.0.0:*               LISTEN      1041/xinetd
tcp        0      0 0.0.0.0:3306           0.0.0.0:*               LISTEN      12824/mysqld
[root@hadoop01 mq-cluster]# jps
31984 jps
31470 activemq.jar
[root@hadoop01 mq-cluster]#
```

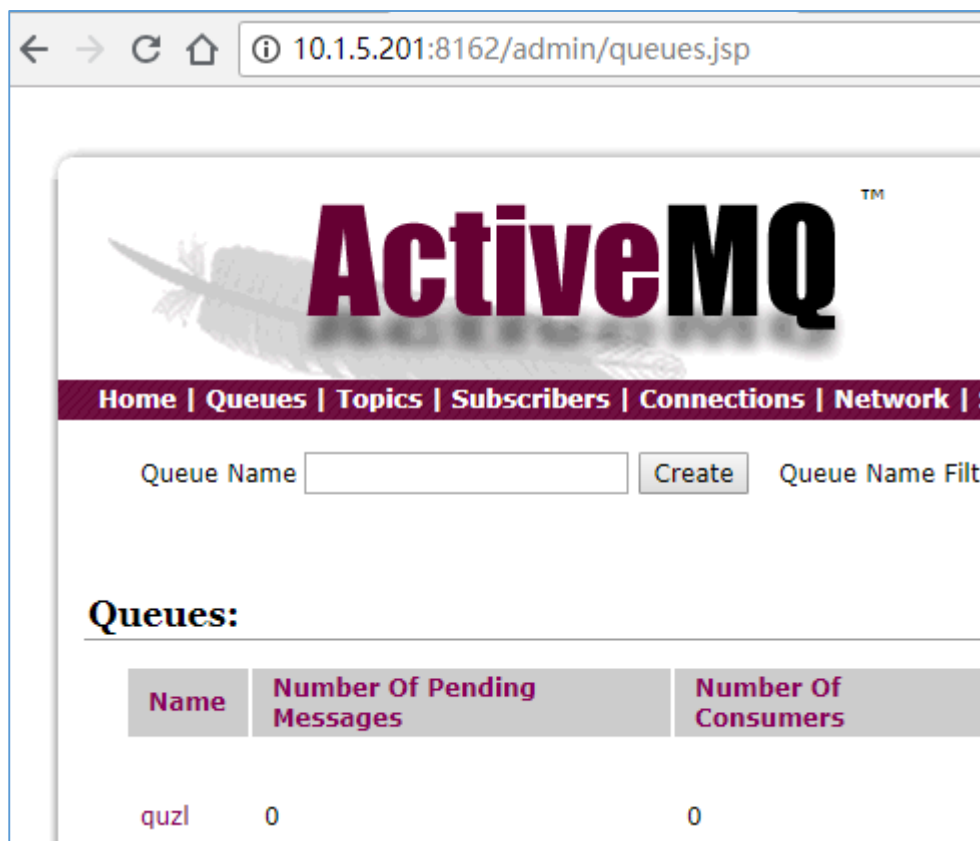
比如 3 节点集群，至少要有 2 个节点在线，最多挂 1 个节点；5 节点集群，至少要有 3 节点在线，最多挂 2 个；7 节点集群，至少要有 4 节点在线，最多挂 3 个；以此类推。所以，需要做好监控，当有节点挂时，应及时恢复服务。
此外不建议偶数个节点。

3.7.2 主从切换数据是否丢失

在 61616 为主时，新增一个队列



61617 成为 master 后，使用 8162 端口访问查看，发现队列仍然存在。



3.7.3 观察 zookeeper

使用 zkCli.sh 连接 zk 集群，默认 IP 127.0.0.1，默认端口 2181

```
[root@hadoop04 bin]# pwd
/data/zookeeper-3.4.12/bin
[root@hadoop04 bin]#
[root@hadoop04 bin]#
[root@hadoop04 bin]# ls
README.txt zkCleanup.sh zkCli.cmd zkCli.sh zkEnv.cmd zkEnv.sh zkServer.cmd zkServer.sh
[root@hadoop04 bin]#
[root@hadoop04 bin]# ./zkCli.sh
Connecting to localhost:2181
```

使用 ls/get 等查看相关信息，更多命令，请参考官方文档。注意：zk 中的路径和 3.5.1 步中的配置有关。

```
[zk: localhost:2181(CONNECTED) 0] ls /activemq
[leveldb-stores]
[zk: localhost:2181(CONNECTED) 1] ls /activemq/leveldb-stores
[000000000013, 000000000014, 000000000015]
[zk: localhost:2181(CONNECTED) 2]
[zk: localhost:2181(CONNECTED) 2] get /activemq/leveldb-stores

czxid = 0x300000005
ctime = Thu Aug 09 20:29:24 CST 2018
mzxid = 0x300000005
mtime = Thu Aug 09 20:29:24 CST 2018
pzxid = 0x30000007c
cversion = 29
dataversion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 0
numChildren = 3
[zk: localhost:2181(CONNECTED) 3] █
```

3.7.4 后续

当再次启 61616 后，61616 处于 salve 状态，61617 仍然是 master。

四、安装 zookeeper 集群

4.1 规划

zookeeper 集群至少需要三个节点，规划如下：

10.1.5.203, clientPort=2181

10.1.5.204, clientPort=2181

10.1.5.205, clientPort=2181

4.2 配置jdk 环境

建议使用 oracle 的 jdk，当然使用系统自带的 openjdk 也是可以的，见 1.4。

4.3 下载 zookeeper

下载地址：<http://www.apache.org/dist/zookeeper/>
`tar xvzf zookeeper-3.4.12.tgz -C /data/`

4.4 修改配置文件

在 10.1.5.203 上：

```
cd /data/zookeeper-3.4.12/conf
```

```
cp zoo_sample.cfg zoo.cfg
```

修改配置文件如下：

```
cat zoo.cfg
```

```
tickTime=2000
```

```
initLimit=10
```

```
syncLimit=5
```

```
dataDir=/data/zookeeper-data
```

```
clientPort=2181
```

```
server.1=10.1.5.203:2888:3888
```

```
server.2=10.1.5.204:2888:3888
```

```
server.3=10.1.5.205:2888:3888
```

4.5 配置其他节点

同 6.1.4 完全相同，配置 10.1.5.204 和 10.1.5.205

4.6 新建好相关目录

三台机器上均操作：

```
mkdir -pv /data/zookeeper-data
```

4.7 新建 myid 文件

10.1.5.203 上：

```
echo 1 > /data/zookeeper-data/myid
```

10.1.5.204 上：

```
echo 2 > /data/zookeeper-data/myid
```

10.1.5.205 上：

```
echo 3 > /data/zookeeper-data/myid
```

4.8 启动

```
cd /data/zookeeper-3.4.12/bin
```

```
./zkServer.sh start
```

以此启动 3 个节点。

bin 目录下会生产 zookeeper.out 文件

4.9 查看状态

分布在 3 个节点上执行

```
/data/zookeeper-3.4.12/bin/zkServer.sh status
```

会有以下两种结果，leader 表示主，follower 表示从：

Mode: leader

Mode: follower

3 点的分布输入 jps，会看下类似如下的进程

```
[root@hadoop04 ~]# jps
```

```
11628 QuorumPeerMain
```

```
11950 Jps
```