Clustering AMQ7 brokers

A-MQ 7 has the ability intelligently store & forward messages around a cluster of brokers in order to load-balance and prevent "starvation". In addition to that, however, a clustered setup is required in order to do any type of master/slave (whether shared store, or replicated).

There are a few different options for configuring a clustered setup:

- Multicast
- JGroups
- Static

For the purposes of this lab, we will use the static option.

Prerequisites

Download and install the A-MQ 7 broker package.

Creating the brokers

```
$ $AMQ_HOME/bin/artemis create brokers/node1 --name node1 --user admin --
password admin --allow-anonymous

$ $AMQ_HOME/bin/artemis create brokers/node2 --name node2 --user admin --
password admin --allow-anonymous --port-offset 1
```

Note: We could have passed in some extra arguments to configure our cluster with some defaults, but instead we're going to manually configure the cluster via the `broker.xml` file so we can become familiar with the available options.

Editing the configuration

Node 1

1. Open up the `brokers/node1/etc/broker.xml` file in your favorite text editor.
2. Add the following elements anywhere under the `<core>` element:

```
<connectors>
  <connector name="node1-connector">tcp://localhost:61616</connector>
  <connector name="node2-connector">tcp://localhost:61617</connector>
</connectors>

<cluster-user>admin</cluster-user>
<cluster-password>admin</cluster-password>

<cluster-connections>
  <cluster-connection name="static-cluster">
    <connector-ref>node1-connector</connector-ref>
    <message-load-balancing>ON_DEMAND</message-load-balancing>
    <static-connectors>
```

```
      <connector-ref>node2-connector</connector-ref>
    </static-connectors>
  </cluster-connection>
</cluster-connections>
```

3. Add the following elements anywhere under the `<address-setting>` element whose `match` attribute is equal to "#" (meaning it matches all addresses):

```
<redistribution-delay>0</redistribution-delay>
```

4. Start the broker:

```
$ ./brokers/node1/bin/artemis run
```

Node 2

1. Open up the `brokers/node2/etc/broker.xml` file in your favorite text editor.
2. Add the following elements anywhere under the `<core>` element:

```
<connectors>
  <connector name="node1-connector">tcp://localhost:61616</connector>
  <connector name="node2-connector">tcp://localhost:61617</connector>
</connectors>

<cluster-user>admin</cluster-user>
<cluster-password>admin</cluster-password>

<cluster-connections>
  <cluster-connection name="static-cluster">
    <connector-ref>node2-connector</connector-ref>
    <message-load-balancing>ON_DEMAND</message-load-balancing>
    <static-connectors>
      <connector-ref>node1-connector</connector-ref>
    </static-connectors>
  </cluster-connection>
</cluster-connections>
```

3. Add the following elements anywhere under the `<address-setting>` element whose `match` attribute is equal to "#" (meaning it matches all addresses):

```
<redistribution-delay>0</redistribution-delay>
```

4. Start the broker:

```
$ ./brokers/node2/bin/artemis run
```

Testing

Open up two terminal windows and run the following commands:

Terminal 1

```
$ $AMQ_HOME/bin/artemis producer --verbose --user admin --password admin --
sleep 1000 --message-count 100 --url 'tcp://localhost:61616'
```

Terminal 2

```
$ $AMQ_HOME/bin/artemis consumer --verbose --user admin --password admin --
message-count 100 --url 'tcp://localhost:61617'
```

You should see that the messages are produced to node1, but consumed from node2.