spring-cloud-stream-binder-ibmmq

Spring Cloud Stream Binder implementation for IBM MQ

IBM MQ Setup

1. Login to the docker host, if running windows or Mac, the command will be something like

   `- a a` 2. Update maximum open files settings

   `- . - a =2000000`

2. Navigate to `` folder of the submission and build the custom docker image

   `- - - .` 4. Run the following to start server

   `-- -- LICENSE=a -- MQ_QMGR_NAME=QM1 -- / / a1/ - :/ a / -- a - - -- 14`

Build the binders and demos

Prerequisites

1. Java 8
2. Maven 3.3.9

Build

1. Install native MQ library, from `/` directory run

   `a : a - -D = . . .a . a -D F = . . .a . .`

1. From the root submission directory, run

   `a a -P`

Verification

In the verification steps below, we will use the two supplied demos, demo-sender and demo-receiver.

To run a demo with the supplied command you will need to navigate to the relevant demo sub-directory.

Common Configuration

update the `/ a / /a a .` files in demo-sender and demo-receiver apps with the correct values for your environment, specifically, you will need to set the following properties (the values below are the default)

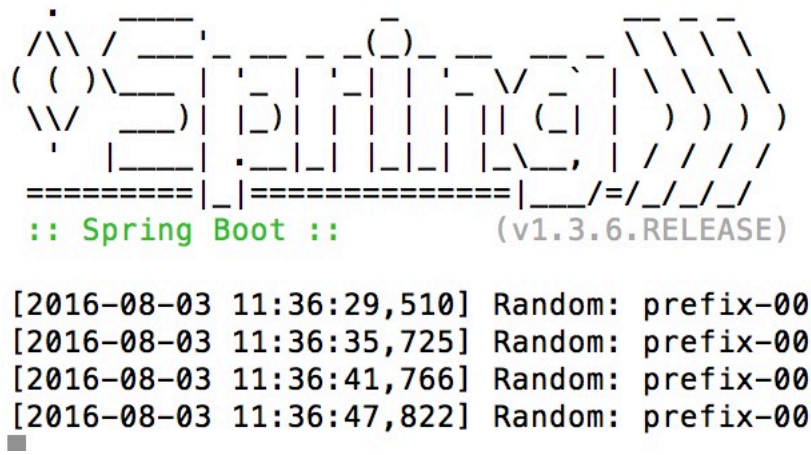
```
. .  =127.0.0.1
. .  =1414
. . =QM1
. .  a =?
. . a  =?
. .  a  =SYSTEM.DEF.SVRCONN
```

Sender/Receiver With Queue Destination and No Consumer Grouping

1. Update the `/ a / /a a .` file of the demo-sender and make sure it contains the following setting

   `. . a . . . a - a - =` 2. Run Sender

   `- : -D .a ="-- . =8881"`



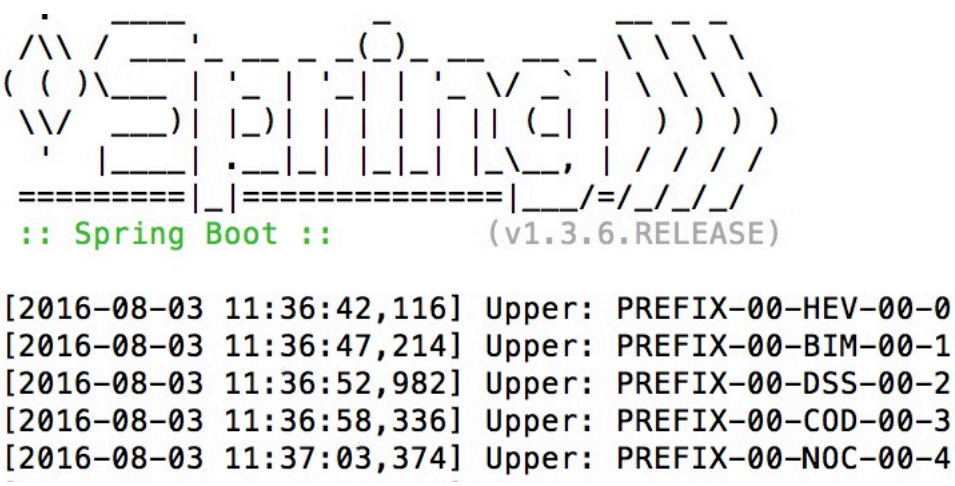
Please note the generated strings with the index suffex that will be used to track the messages when running receivers.

2. Update the `/a/a.` file of the demo-receiver and make sure it contains the following setting

```
.a.a-a-=
```

1. Run Receiver

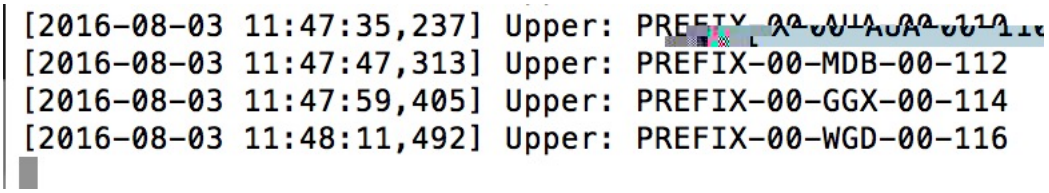
```
-:-D.a="--.=8882"
```



1. Run Multiple Receivers Just make sure to change the server.port for each receiver

```
-:-D.a="--.=8883"
```

First receiver log



Second receiver log

```
[2016-08-03 11:47:29,224] Upper: PREFIX-00-XPL-00-109
[2016-08-03 11:47:41,276] Upper: PREFIX-00-RIY-00-111
[2016-08-03 11:47:53,366] Upper: PREFIX-00-ULR-00-113
[2016-08-03 11:48:05,451] Upper: PREFIX-00-TWC-00-115
[2016-08-03 11:48:17,533] Upper: PREFIX-00-BRM-00-117
[2016-08-03 11:48:29,616] Upper: PREFIX-00-IUR-00-119
[2016-08-03 11:48:41,698] Upper: PREFIX-00-PYZ-00-121
```

Note that the receivers are competing for the messages.

Sender/Receiver With Queue Destination and with Consumer Grouping

1. Keep the sender running
2. Terminate the receivers
3. Re-run the first receiver with the following command

```
          -    :    -D   .a        ="--      .    =8882,--      .     .    a .        .     .     =     1"
```

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v1.3.6.RELEASE)

2016-08-03 11:56:51,045  WARN main z.c.a.c.s.b.i.IbmMqMessageChannelBinder:142 - Defining group with a QUEUE destination type has no effect
[2016-08-03 11:56:51,584] Upper: PREFIX-00-NCZ-00-197
[2016-08-03 11:56:56,671] Upper: PREFIX-00-TUN-00-198
[2016-08-03 11:57:01,705] Upper: PREFIX-00-IOR-00-199
```

The grouping with a Queue Destination type has no sense, so it has no effect on functionality and a warning message is logged in this case.

Sender/Receiver With Topic Destination and No Consumer Grouping

1. Terminate the sender and receivers
2. Update the `/a/a.` file of the demo-sender and demo-receiver and make sure it contains the following setting

```
      .    .    a .    .     .   a   -     a    -    =
```

1. Run sender

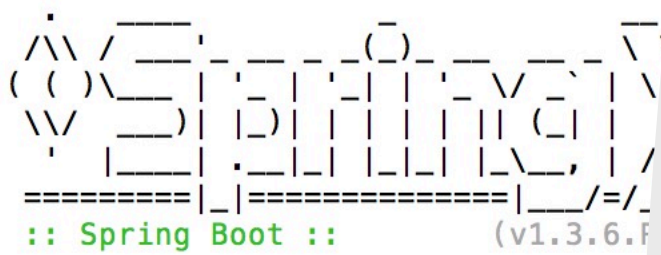
```
          -    :    -D   .a        ="--      .    =8881"
```

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v1.3.6.RELEASE)

[2016-08-03 12:07:40,430] Random: prefix-00-eGY-00-0
[2016-08-03 12:07:46,634] Random: prefix-00-GkJ-00-1
[2016-08-03 12:07:52,678] Random: prefix-00-abG-00-2
```

1. Run Multiple Receivers

```
          -    :    -D   .a        ="--      .    =8882"
```




```
[2016-08-03 12:10:48,120] Upper: P
[2016-08-03 12:10:54,129] Upper: P                    -32
```

And the message "prefix-00-aXu-00-11" will have suffix index equal to 11 and will be routed to partition 11%5 = 1

2. We will have multiple receivers that will subscribe to one of the partitions and so will ONLY receive messages related to this partition.

Steps

1. Terminate the sender and receivers.
2. Update application.properties of the sender to enable partitioning

```
#  a                                                    a
      .     .    a .        .      .        . a       K  E         =T( a a. a  .I      ). a   I  ( a   a .         (17))
      .     .    a .        .      .        . a       C    =5
      .     .    a .        .      .        .        -      [0]=     1
      .     .    a .        .      .        .        -      [1]=     2
```

The required-groups property will trigger the creation of required durable subscriptions before any message is sent.

3. Run the sender

```
          -    :    -D   .a        ="--      .    =8881"
```

4. Update application.properties of the receiver to enable partitioning

```
      .     .    a .        .     .        . a         =
      .     .    a .    a   -     =5
```

5. Run multiple receivers

```
1.          -    :    -D   .a        ="--      .    =8882,--      .     .    a .        .     .    =     1,--      .     .    a .    a
2.          -    :    -D   .a        ="--      .    =8883,--      .     .    a .        .     .    =     1,--      .     .    a .    a
3.          -    :    -D   .a        ="--      .    =8884,--      .     .    a .        .     .    =     2,--      .     .    a .    a
4.          -    :    -D   .a        ="--      .    =8885,--      .     .    a .        .     .    =     1,--      .     .    a .    a
5.          -    :    -D   .a        ="--      .    =8886,--      .     .    a .        .     .    =     1,--      .     .    a .    a
6.          -    :    -D   .a        ="--      .    =8887,--      .     .    a .        .     .    =     1,--      .     .    a .    a
7.          -    :    -D   .a        ="--      .    =8888,--      .     .    a .        .     .    =     1,--      .     .    a .    a
```

In this topology we have

- Receiver #1 and #2 both in "group1" will compete for messages in partition 0
- Receiver #3 in "group2" will receive another copy of messages in partition 0
- Receiver #4 in "group1" will receive messages in partition 1
- Receiver #5 in "group1" will receive messages in partition 2
- Receiver #6 in "group1" will receive messages in partition 3
- Receiver #7 in "group1" will receive messages in partition 4