

IBM MQ multi-instance queue manager setup in Linux

<https://www.ibm.com/support/pages/node/6891131>

Date last updated: 31-Jan-2023

Abhishek Vishwakarma

IBM MQ Support

<https://www.ibm.com/products/mq/support>

Find all the support you need for IBM MQ

Table of Contents

1. Multi-instance setup.....	2
2. Verifying failover using IBM MQ Explorer	7
3. Verifying failover using the sample programs	16

+++ Objective +++

This documentation has the steps for setting up and verifying multi-instance queue manager in Linux. The setup is small to illustrate the concepts involved.

Servers used in the documentation:

NFS Server: sayyid1.fyre.ibm.com (9.46.77.180)

MQ Server1: whelk1.fyre.ibm.com (9.46.78.22)

MQ Server2: sketcher1.fyre.ibm.com (9.46.83.145)

(NOTE: Use your server IPs in the setup)

1. Multi-instance setup

Setting up the NFS server:

1) Login as 'root' user.

2) Install NFS:

2.1) Install the NFS utilities package: **yum install nfs-utils**

2.2) Enable and start the NFS service: **sudo systemctl enable --now nfs-server**

3) Verify if nfs-server service is up and running:

systemctl status nfs-server.service

4) create a directory on NFS Server as 'root' user:

mkdir -p /MQHA

5) Add entries for the mounts in /etc/exports file that are exported for both the MQ Servers:

vi /etc/exports

Paste the below content (Mention your MQ server IPs):

/MQHA 9.46.78.22(rw,sync,no_root_squash)

/MQHA 9.46.83.145(rw,sync,no_root_squash)

```
/MQHA 9.46.78.22(rw,sync,no_root_squash)
/MQHA 9.46.83.145(rw,sync,no_root_squash)
```

```
~
~
~
~
~
~
~
~
~
~
~
~
```

```
[[root@sayyid1 /]# cat /etc/exports
/MQHA 9.46.78.22(rw, sync, no_root_squash)
/MQHA 9.46.83.145(rw, sync, no_root_squash)
[[root@sayyid1 /]# █
```

6) To export the directory as an NFS4 mount, execute the below command every time you update the /etc/exports files:

exportfs -av

```
[[root@sayyid1 /]# exportfs -av
exporting 9.46.78.22:/MQHA
exporting 9.46.83.145:/MQHA
[[root@sayyid1 /]# █
```

(NOTE: Run every time you update the /etc/exports file)

Setting up the MQ server (run in both the MQ Server):

Install MQ in the other two Linux servers. These will be our MQ servers.

1) Login as 'root' user

2) Install IBM MQ

3) Install NFS:

3.1) Install the NFS utilities package: **yum install nfs-utils**

3.2) Enable and start the NFS service: **sudo systemctl enable --now nfs-server**

4) Verify nfs server service is up and running:

systemctl status nfs-server.service

5) Check the mount:

showmount -e 9.46.77.180

```
[root@sketcher1 MQServer]# showmount -e 9.46.77.180
```

```
Export list for 9.46.77.180:
```

```
/MQHA 9.46.83.145,9.46.78.22
```

6)

6.1) Create the directory: **mkdir -p /MQHA**

6.2) execute with root privilege: **mount -t nfs4 9.46.77.180:/MQHA /MQHA**

6.3) Command to verify that the mount is completed or not: **\$ mount or showmount -e 9.46.77.180**

6.4) Verify that files are created successfully by creating a test file in all three servers (NFS server, MQServer1, MQServer2):

a) **cd /MQHA**

b) **touch <filename>**

6.5) Add the following line in the MQ Servers "/etc/fstab" for automatic mount of the shared directory /MQHA after restart:

9.46.77.180:/MQHA /MQHA nfs4 defaults 0 0

Run only on one MQ server:

NOTE: I am using MQ Server1 (wheelk1)

1) Create log and data directories in a common folder, /MQHA, that is to be shared.

mkdir /MQHA/logs

mkdir /MQHA/qmgrs

Set access permissions to rwx for user and group and make mqm the directory owner:

chown -R mqm:mqm /MQHA

chmod -R ug+rwx /MQHA

2) Create the queue manager by entering the following command: **crtmqm -**

ld /MQHA/logs -md /MQHA/qmgrs QM1

```
[mqm@wheelk1 /]$ crtmqm -ld /MQHA/logs -md /MQHA/qmgrs QM1
```

IBM MQ queue manager 'QM1' created.

Directory '/MQHA/qmgrs/QM1' created.

The queue manager is associated with installation 'Installation1'.

Creating or replacing default objects for queue manager 'QM1'.

Default objects statistics : 83 created. 0 replaced. 0 failed.

Completing setup.

Setup completed.

3) Copy the queue manager configuration details from Server1:

dspmqrinf -o command QM1

and copy the result to the clipboard: **addmqinf -s QueueManager -v Name=QM1 -v**

Directory=QM1 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM1

Run only in MQ server2 (sketcher1):

NOTE: I used MQ Server1 earlier so in this step I will be pasting the configurations in Server2. You can do the same steps in opposite manner.

1) Add queue manager configuration data:

addmqinf -s QueueManager -v Name=QM1 -v Directory=QM1 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM1

2) Start the queue manager instances, with the **-x** parameter: **strmqm -x QM1**

Status in current MQ server (sketcher1):

QMNAME(QM2) STATUS(Running)

Status in other MQ server (wheelk1):

QMNAME(QM1) STATUS(Running elsewhere)

Run only in MQ server1(whelk1):

Before the start of QM, its status should be "Running elsewhere"

Start the queue manager instances : **strmqm -x QM1**

```
[mqm@whelk1 root]$ strmqm -x QM1
IBM MQ queue manager 'QM1' starting.
The queue manager is associated with installation 'Installation1'.
Plain text communication is enabled.
A standby instance of queue manager 'QM1' has been started. The active instance
is running elsewhere.
```

After start of QM1, its status should be "Running as standby"

Run only in MQ server2 (sketcher1):

Here, the status of QM is "Running". And in the other MQ server, status is "Running as standby"

1)

Switchover to the other MQ server: **endmqm -is QM1**

QM1 is now running in Server1(whelk1).

Status in this MQ server (sketcher1):

```
QMNAME(QM1)                STATUS(Running elsewhere)
```

Status in the other MQ server (whelk1):

```
QMNAME(QM1)                STATUS(Running)
```

2) Start the queue manager instances: **strmqm -x QM1**

This will start the queue manager as a standby on this server.

Status in this MQ server (sketcher1):

```
QMNAME(QM1)                STATUS(Running as standby)
```

Status in another MQ server (whelk1):

```
QMNAME(QM1)                STATUS(Running)
```

2. Verifying failover using IBM MQ Explorer

Run only on one MQ server:

NOTE: I am using MQ Server1 (whelk1)

1) Create the queue manager by entering the following command:

crtmqm -ld /MQHA/logs -md /MQHA/qmgrs QM3

```
[mqm@whelk1 /]$ dspmq
QMNAME(QM1)                STATUS(Running)
QMNAME(QM2)                STATUS(Running)
QMNAME(QM3)                STATUS(Ended immediately)
[mqm@whelk1 /]$
```

2) Copy the queue manager configuration details from Server1:

dspmqinf -o command QM3

and copy the result to the clipboard: **addmqinf -s QueueManager -v Name=QM3 -v Directory=QM3 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM3**

Run only in MQ server2 (sketcher1):

1) Add queue manager configuration data:

addmqinf -s QueueManager -v Name=QM3 -v Directory=QM3 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM3

```
[mqm@sketcher1 /]$ dspmq
QMNAME(QM1)                STATUS(Running as standby)
QMNAME(QM2)                STATUS(Running as standby)
QMNAME(QM3)                STATUS(Ended immediately)
[mqm@sketcher1 /]$
```

2) Start the queue manager instances, with the **-x** parameter: **strmqm -x QM3**

```
[mqm@sketcher1 /]$ dspmq
QMNAME(QM1)          STATUS(Running as standby)
QMNAME(QM2)          STATUS(Running as standby)
QMNAME(QM3)          STATUS(Running)
[mqm@sketcher1 /]$
```

```
[mqm@whelk1 /]$ dspmq
QMNAME(QM1)          STATUS(Running)
QMNAME(QM2)          STATUS(Running)
QMNAME(QM3)          STATUS(Running elsewhere)
[mqm@whelk1 /]$
```

Run only in MQ server1(whelk1):

Start the queue manager instance: **strmqm -x QM3**

```
[mqm@whelk1 /]$ dspmq
QMNAME(QM1)          STATUS(Running)
QMNAME(QM2)          STATUS(Running)
QMNAME(QM3)          STATUS(Running as standby)
[mqm@whelk1 /]$
```

Run only in MQ server2 (sketcher1):

Create local queues, channels, listener and alter authinfo.

runmqsc QM3

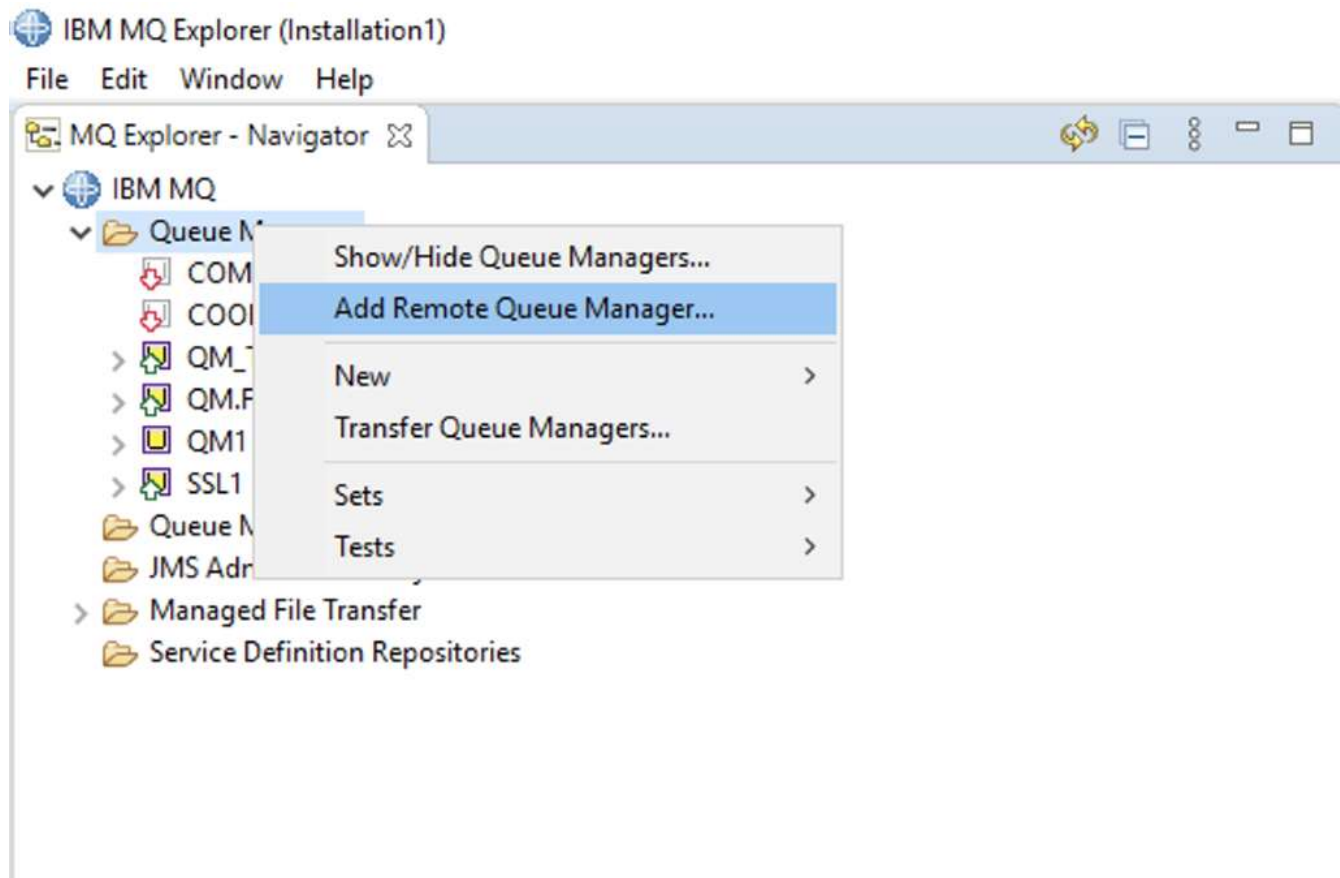
```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) MCAUSER('mqm')
REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN)
TRPTYPE(TCP) CONNAME('9.46.78.22 (1414), 9.46.83.145(1414)') QMNAME(QM3)
REPLACE
START CHANNEL(CHANNEL1)
```



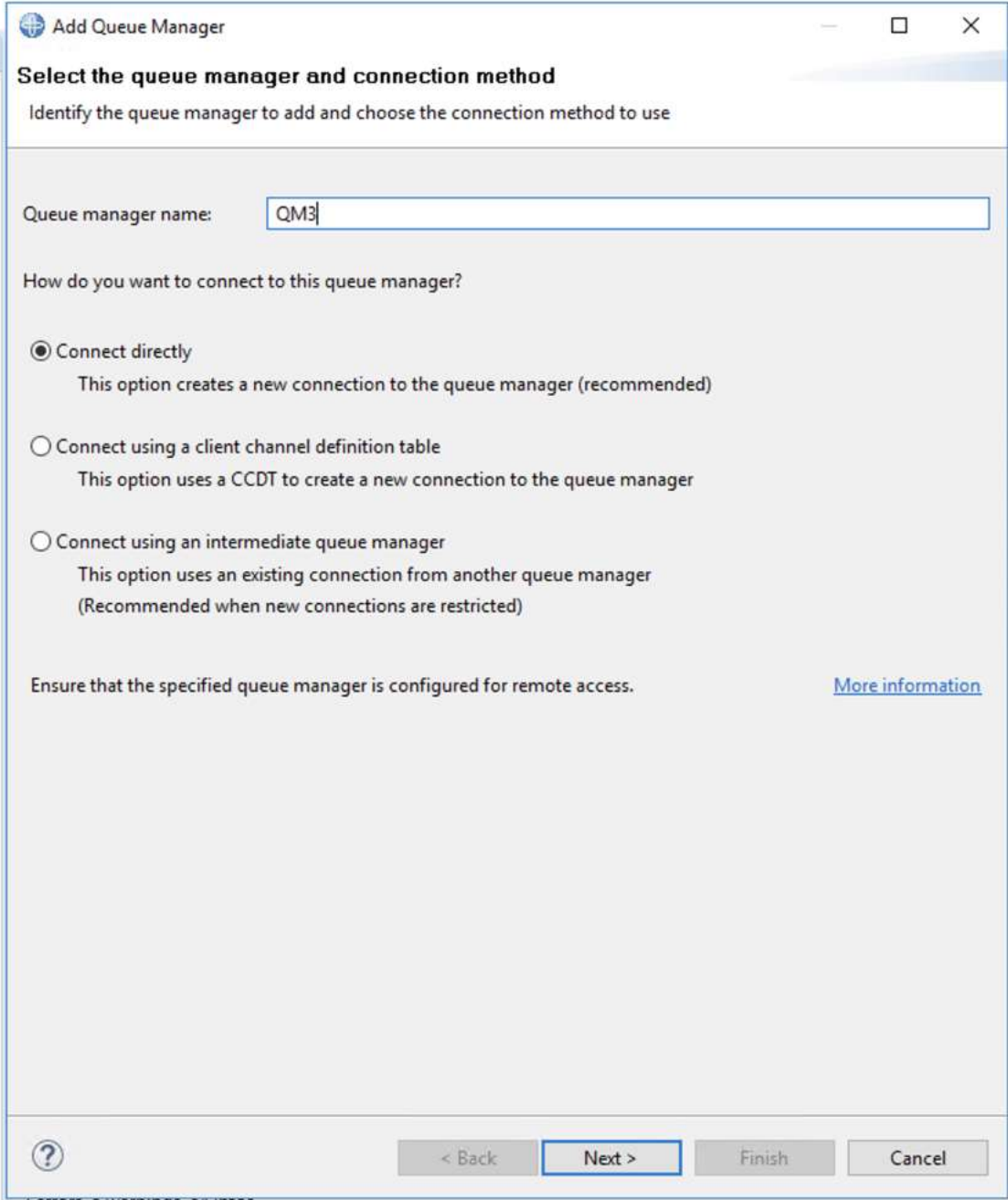
```
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
START LISTENER(LISTENER.TCP)
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
alter authinfo(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS)
CHCKCLNT(OPTIONAL)
SET CHLAUTH('CHANNEL1') TYPE (BLOCKUSER) USERLIST(NOBODY)
refresh security(*)
```

Add QM3 on MQExplorer

1) Select 'Add Remote Queue Manager'



2) Mention queue manager name



The image shows a Windows-style dialog box titled "Add Queue Manager". The main heading is "Select the queue manager and connection method", followed by the instruction "Identify the queue manager to add and choose the connection method to use". There is a text input field for "Queue manager name:" containing the text "QM3". Below this, the question "How do you want to connect to this queue manager?" is followed by three radio button options: "Connect directly" (selected), "Connect using a client channel definition table", and "Connect using an intermediate queue manager". Each option has a descriptive sub-text. A "More information" link is present. At the bottom, there is a status bar with a question mark icon and the text "2 errors, 3 warnings, 34 infos". Navigation buttons include "< Back", "Next >" (highlighted), "Finish", and "Cancel".

Add Queue Manager

Select the queue manager and connection method
Identify the queue manager to add and choose the connection method to use

Queue manager name:

How do you want to connect to this queue manager?

☒ Connect directly
This option creates a new connection to the queue manager (recommended)

☐ Connect using a client channel definition table
This option uses a CCDT to create a new connection to the queue manager

☐ Connect using an intermediate queue manager
This option uses an existing connection from another queue manager
(Recommended when new connections are restricted)

Ensure that the specified queue manager is configured for remote access. [More information](#)

? < Back Next > Finish Cancel

2 errors, 3 warnings, 34 infos

3) Mention the running instance IP first. Select check box "Is this a multi-instance queue manager" then mention IP of standby instance. Fill 'Port number' and 'server-connection channel' field.

Add Queue Manager

Specify new connection details
Provide details of the connection you want to set up

Queue manager name: QM3

Connection details

Host name or IP address: 9.46.83.145

Port number: 1414

Server-connection channel: CHANNEL1

☒ Is this a multi-instance queue manager?

Connection details to second instance

Host name or IP address: 9.46.78.22

Port number: 1414

Server-connection channel: CHANNEL1

☒ Automatically connect to this queue manager at startup or if the connection is lost

☒ Automatically refresh information shown for this queue manager

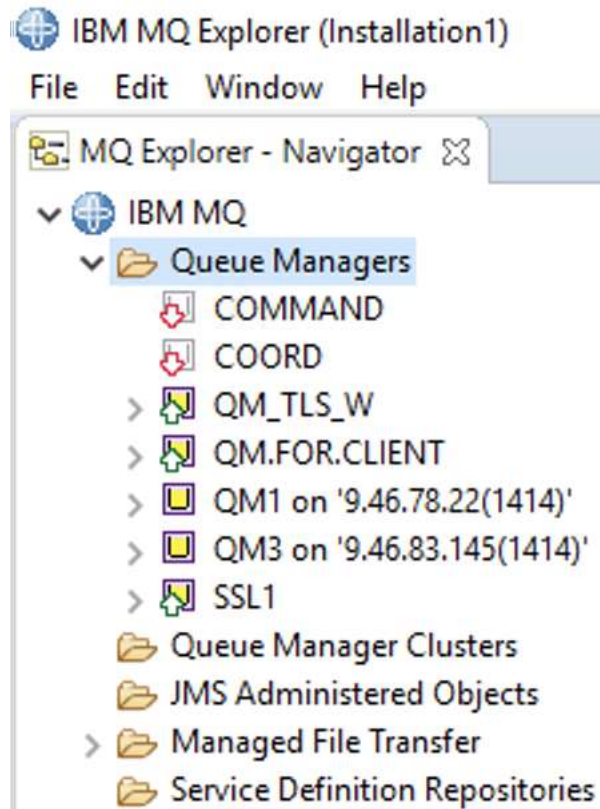
Refresh interval (seconds): 300

? < Back Next > Finish Cancel

2 errors, 3 warnings, 34 infos

4) Click next...next...next...finish

5) QM3 added successfully



Run only in MQ server2 (sketcher1):

Here, the status of QM is "Running". And in the other MQ server, status is "Running as standby"

1)

Switchover to the other MQ server: **endmqm -is QM3**

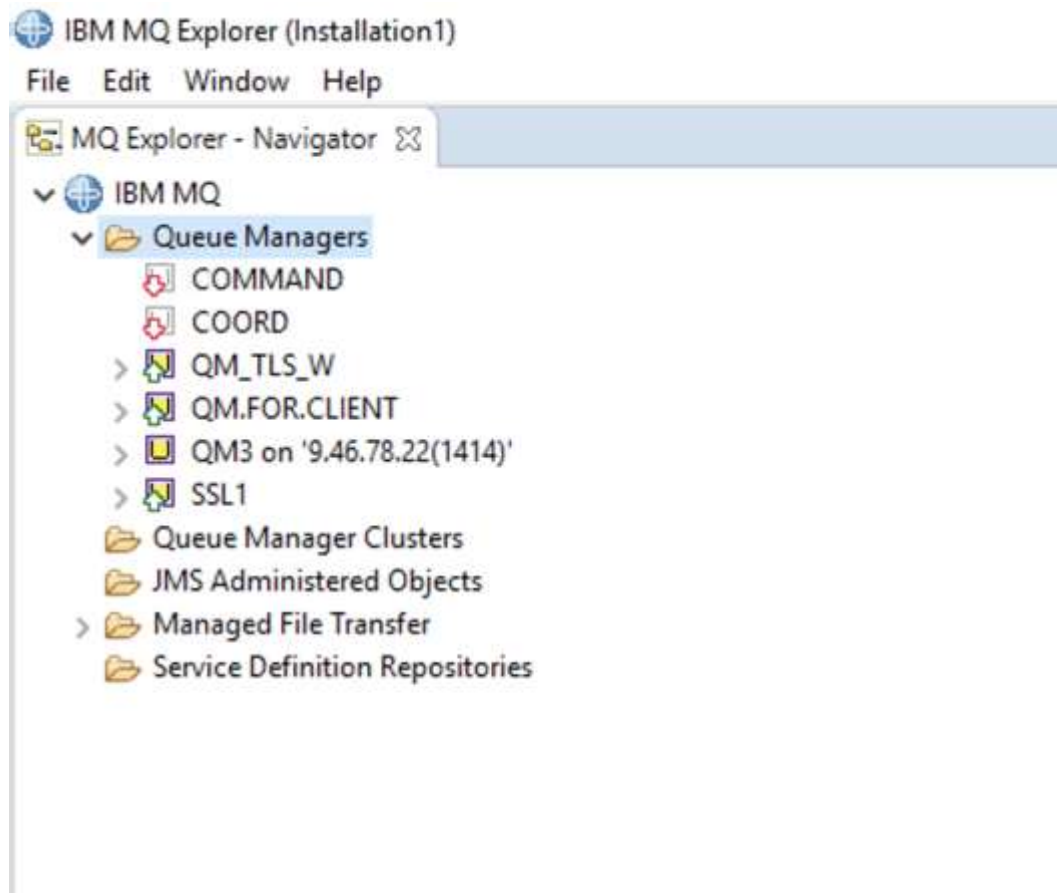
```
[[mqm@sketcher1 /]$ endmqm -s QM3
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete, permitting switchover to a standby instance.
[[mqm@sketcher1 /]$
[[mqm@sketcher1 /]$ dspmq
QMNAME(QM1)                STATUS(Running as standby)
QMNAME(QM2)                STATUS(Running as standby)
QMNAME(QM3)                STATUS(Running elsewhere)
[[mqm@sketcher1 /]$ █
```

QM3 is now running in Server1(whelk1).

```
[[mqm@whelk1 /]$ dspmq
QMNAME(QM1)                STATUS(Running)
QMNAME(QM2)                STATUS(Running)
QMNAME(QM3)                STATUS(Running)
[[mqm@whelk1 /]$ █
```

Status in MQ Explorer:

The IP address is switched to a server having running instance.



2) Start the queue manager instance: **strmqm -x QM3**

This will start the queue manager as a standby on this MQ server.

```
[[mqm@sketcher1 /]$ strmqm -x QM3
IBM MQ queue manager 'QM3' starting.
The queue manager is associated with installation 'Installation1'.
Plain text communication is enabled.
A standby instance of queue manager 'QM3' has been started. The active instance
is running elsewhere.
[[mqm@sketcher1 /]$
[[mqm@sketcher1 /]$ dspmq
QMNAME(QM1)                                STATUS(Running as standby)
QMNAME(QM2)                                STATUS(Running as standby)
QMNAME(QM3)                                STATUS(Running as standby)
[[mqm@sketcher1 /]$
```

```
[mqm@wheelk1 ~]$ dspmq
QMNAME(QM1)                STATUS(Running)
QMNAME(QM2)                STATUS(Running)
QMNAME(QM3)                STATUS(Running)
[mqm@wheelk1 ~]$
```

3. Verifying failover using the sample programs

1) Copy AMQCLCHL.TAB file from MQ HA server to client server. Use FileZilla or other software to transfer file from one server to other.

2) Open three sessions/terminal for the same client server.

3) Set below environmental variables in each:

```
export MQCHLLIB=/var/mqm/mq_server_details_for_client_connection/whelk1_sketcher1_HA_server/QM3
export MQCHLTAB=AMQCLCHL.TAB
unset MQSERVER
cd /opt/mqm/samp/bin
```

4) Check the status of queue manager in both the MQ servers

4.1) In active server:

```
[root@whelk1 ~]# dspmq
QMNAME(QM3)                                STATUS(Running)
```

4.2) In standby server:

```
[root@sketcher1 ~]# dspmq
QMNAME(QM3)                                STATUS(Running as standby)
```

5)

5.1) Run in client session1: **./amqsgnac TARGET QM3**


```
~ — root@whelk1:~ — ssh root@9.46.78.22
[[mqm@gruffest1 bin]$ ./amqsgnac TARGET QM3
Sample AMQSGHAC start
message <Message 1>
message <Message 2>
message <Message 3>
message <Message 4>
message <Message 5>
█
```

5.2) Run in client session2: **./amqsmnac -s SOURCE -t TARGET -m QM3**

```
~ — root@whelk1:~ — ssh root@9.46.78.22
[[mqm@gruffest1 bin]$ ./amqsmnac -s SOURCE -t TARGET -m QM3
Sample AMQSMHAC start
█
```

5.3) Run in client session3: **./amqsphac SOURCE QM3**

```

~ — root@wheel1:~ — ssh root@9.46.78.22

[[mqm@gruffest1 bin]$ ./amqsphac SOURCE QM3
Sample AMQSPHAC start
target queue is SOURCE
message <Message 1>
message <Message 2>
message <Message 3>
message <Message 4>
message <Message 5>
message <Message 6>
message <Message 7>
message <Message 8>
message <Message 9>
message <Message 10>
message <Message 11>
message <Message 12>
message <Message 13>
message <Message 14>
message <Message 15>
message <Message 16>

```

6) In active MQ instance, ran this command for switchover to another MQ server

endmqm-is QM3

```
[[mqm@wheel1 root]$ endmqm-is QM3
```

Quiesce request accepted. The queue manager will stop when all outstanding work is complete, permitting switchover to a standby instance.

The queue manager instance running on another MQ server.

7)

7.1) Connection reconnection in session1

```

message <Message 40>
message <Message 41>
message <Message 42>
message <Message 43>
message <Message 44>
22:22:40 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 2ms)
22:22:40 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1384ms)
22:22:42 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 2312ms)
22:22:44 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 4955ms)
22:22:49 : EVENT : Connection Reconnected
message <Message 45>
message <Message 46>
message <Message 47>
message <Message 48>
message <Message 49>
message <Message 50>
message <Message 51>
message <Message 52>
message <Message 53>
message <Message 54>
message <Message 55>
message <Message 56>

```

7.2) Connection reconnection in session2

```

-- mqm@wheel1:/root -- ssh root@9.46.78.22
-- root@sketcher1:~ -- ssh root

[mqm@gruffest1 bin]$ ./amqsmhac -s SOURCE -t TARGET -m QM3
Sample AMQSMHAC start

22:22:40 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 152ms)
22:22:41 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1043ms)
22:22:42 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 2206ms)
22:22:44 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 4722ms)
22:22:49 : EVENT : Connection Reconnected

```

7.3) Connection reconnection in session3

```

message <Message 37>
message <Message 38>
message <Message 39>
message <Message 40>
message <Message 41>
message <Message 42>
message <Message 43>
message <Message 44>
22:22:40 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 67ms)
22:22:40 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1394ms)
22:22:42 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 2044ms)
22:22:44 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 4948ms)
22:22:49 : EVENT : Connection Reconnected
message <Message 45>
message <Message 46>
message <Message 47>
message <Message 48>
message <Message 49>
message <Message 50>
message <Message 51>

```

Client successfully switched from one MQ sever to another. No messages were lost after switchover. Connection reestablished after message sequence 44 and again consumed from 45, means no loss of messages.

+++ end +++