Integration Technical Conference 2019

IBM MQ RDQM: Easy HA - Easy once you know how

Morag Hughson

(morag@mqgem.com)

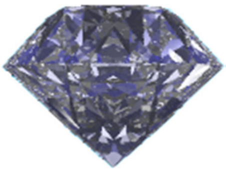
@MoragHughson

IBM Cloud

Your Speaker

- Morag Hughson
- 18 years at IBM
  - I joined IBM as a graduate into the MQSeries development organisation on the MVS/ESA platform. I worked initially on MQ Channels, but over the 18 years I spent working on MQ, I worked on many areas, including SSL/TLS, Publish/Subscribe, all different areas of security, MQSC and PCF interface, and the MQ API. My platform coverage also expanded out from z/OS to take in all the distributed platforms too. After all, when you work on channels, you connect to everything! For the last 10 years in IBM I was the product architect for the IBM MQ product.

- 4 years at MQGem Software
  - I left IBM in 2015 and joined MQGem Software. We make tools for IBM MQ, such as MO71 which some of you may be familiar with. We also do consultancy and education on MQ. During this time I have been honored with the accolade of IBM Champion.



Agenda

- My experience with RDQM
  - I used it and learned about it in order to create an MQ training course for IBM MQ V9 and all its features.
  - This presentation is my view on RDQM, and the hints and tips I noted as I learned about it
  - I hope they will help you have a smooth experience when you make use of RDQM.
- Replicated Data Queue Manager (RDQM) for High Availability (HA)
  - Introduced in IBM MQ V9.0.4
  - Called “Easy HA” by IBM
- Replicated Data Queue Manager (RDQM) for Disaster Recovery (DR)
  - Introduced in IBM MQ V9.0.5
  - Used some of the same technology as RDQM HA

IBM MQ Replicated Data Queue Manager (RDQM)

N

O

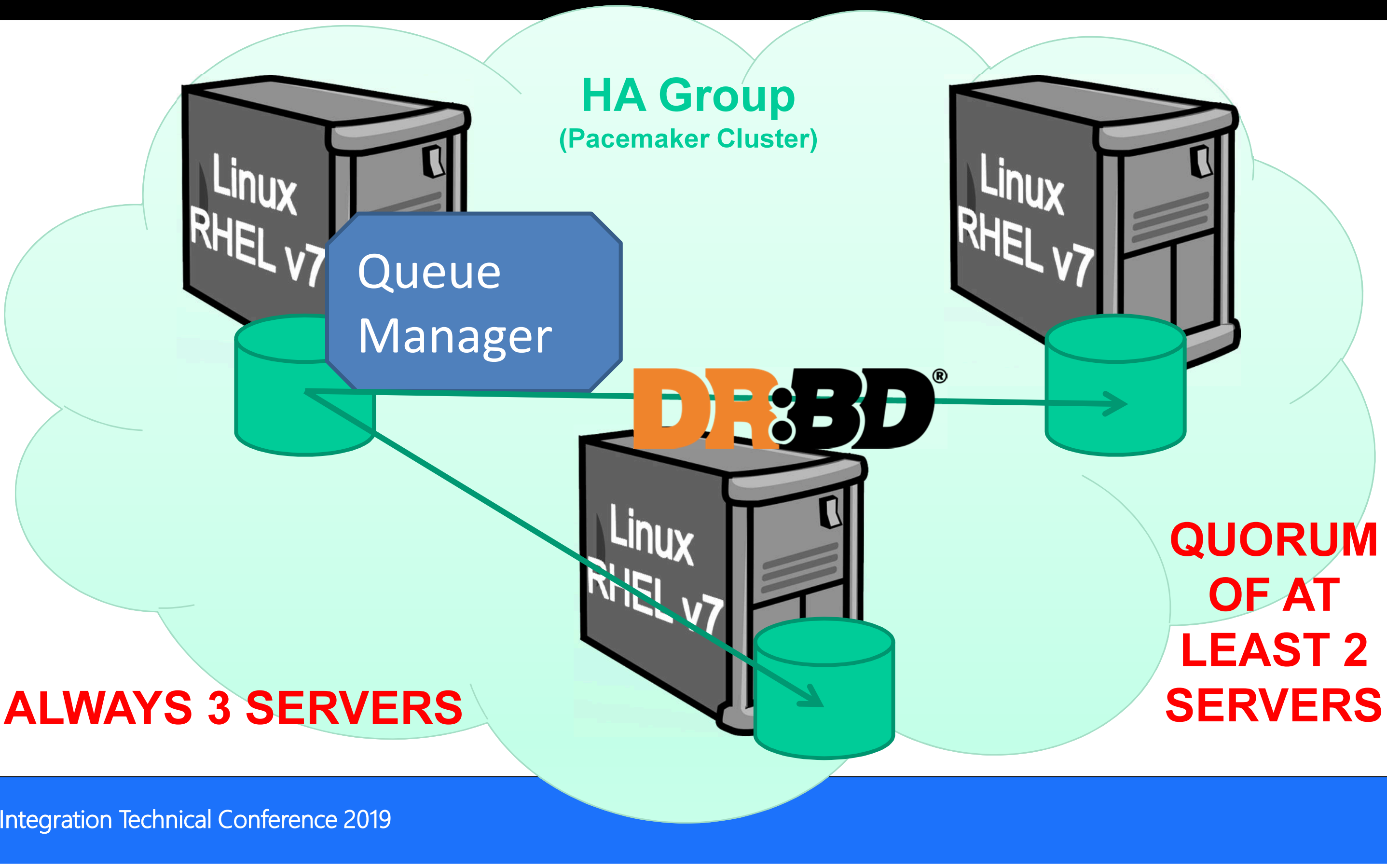
T

E

S

- IBM MQ V9.0.4 introduced the new feature RDQM (Replicated Data Queue Managers) as an HA technology. V9.0.5 used much of the same technology to also offer a DR solution. The RDQM feature is branded "Easy HA" by IBM. This user experience shows that RDQM is easy to set up when you know how. Learn the pitfalls to avoid and the preparation to do to make setting up an RDQM queue manager go smoothly.

RDQM High Availability



ALWAYS 3 SERVERS

QUORUM OF AT LEAST 2 SERVERS

RDQM High Availability - Notes

N O T E S

- RDQM (replicated data queue manager) HA is a high availability solution that is available on RHEL v7.
- An RDQM HA configuration consists of three servers each with an instance of the queue manager. These three servers are configured in a high availability (HA) group (which is using a Pacemaker cluster under the covers).
- The queue manager can only run on one server at any time, and the running instance of the queue manager replicates its data to the other two instances using Distributed Replicated Block Device (DRBD). This means that if the server running this queue manager fails, another instance of the queue manager starts and has current data to operate with.
- RDQM uses three nodes to greatly reduce the possibility of a split-brain situation arising. If it only used two nodes, and network connectivity was lost between the nodes, the queue manager instance could be running on both at the same time and be creating two different sets of data. With three nodes, the primary instance of the queue manager ensures that it can communicate with at least one of the other nodes, thus ensuring that no other node can be running the queue manager because it couldn't form a quorum of at least two.

RDQM Pre-requisite Installations

- The DRBD and Pacemaker RPM packages are supplied on the IBM MQ media. Installation order is important.

DO NOT DOWNLOAD YOUR OWN VERSIONS!

Default Installation Location

- Running as root
- In installation media directory:
  `MQServer/Advanced/RDQM`
- Run script
  `./installRDQMsupport`

Non-Default Installation Location

- Edit script to remove MQ install steps
  `./installRDQMsupport`
- …and run it (as root)
- Finally, install RDQM RPM
  `MQServer/Advanced/RDQM/MQSeriesRDQM*.rpm`

RDQM Pre-requisite Installations - Notes

N
O
T
E
S

- You will need three Linux machines for this setup, and you will have some additional installation steps to do on each machine over and above a normal MQ installation.
- The DRBD and Pacemaker RPM packages are supplied on the IBM MQ media. You should install the versions supplied with IBM MQ. Do not download your own versions. You also need to install them in a particular order.
  - Install DRBD and Pacemaker pre-requisites on each node
  - Install RDQM RPM on each node
- You can install all the other MQ RPMs before or after the pre-requisites are installed, meaning you can take a non-RDQM MQ installation and add RDQM to it later, or install it all at once.
- IBM MQ ships with a script to do all this for you. The supplied DRBD and Pacemaker packages are signed with the LINBIT GPG key. Use the following command to import the public LINBIT GPG key before you run the script.
- `rpm --import https://packages.linbit.com/package-signing-pubkey.asc`
- The script can be found in the installation media in the directory `MQServer/Advanced/RDQM`, and is called `installRDQMsupport`. If you have (or want to have) IBM MQ installed in the default location (`/opt/mqm`) then you can use this script. If the IBM MQ installation is already installed, or needs to be installed, in a non-default location, then you have to do the above steps, using the script for the pre-requisite step.

installRDQMsupport script

```
RDQM_DIR=$(dirname "$0")
RDQM_PACKAGES="$RDQM_DIR/MQSeriesRDQM-9.1.1-0.x86_64.rpm"
MQ_DEPENDENCIES="$RDQM_DIR/../../MQSeriesServer-9.1.1-0.x86_64.rpm ....
ADDITIONAL_MQ_PACKAGES="$RDQM_DIR/../../MQSeriesSamples-9.1.1-0.x86_64.rpm ....
DRBD_PACKAGES="$RDQM_DIR/PreReqs/drbd-9.0/drbd-bash-completion-9.5.0-1.el7.x86_64.rpm ....
PACEMAKER_PACKAGES="$RDQM_DIR/PreReqs/pacemaker-1.1.15/cluster-glue-1.0.12.linbit-2.0 ....
:
    DRBD_KMOD="$RDQM_DIR/PreReqs/drbd-9.0/kmod-drbd-9.0.1 ....
:
yum -y install $DRBD_KMOD $DRBD_PACKAGES $PACEMAKER_PACKAGES
yum -y install $MQ_DEPENDENCIES $RDQM_PACKAGES $ADDITIONAL_MQ_PACKAGES
```

installRDQMsupport script - Notes

NOTES

- This is the script mentioned on the previous page, that is supplied by IBM MQ to help you install the pre-requisite packages for using RDQM.
- The text in blue is the lines to comment out if you are installing RDQM in a non-default location. You must the MQ install steps manually, and use this script only for installing the pre-req packages.

Preparation is Everything!

More Preparation
Less Perspiration!

Preparation is Everything! - Notes

N

O

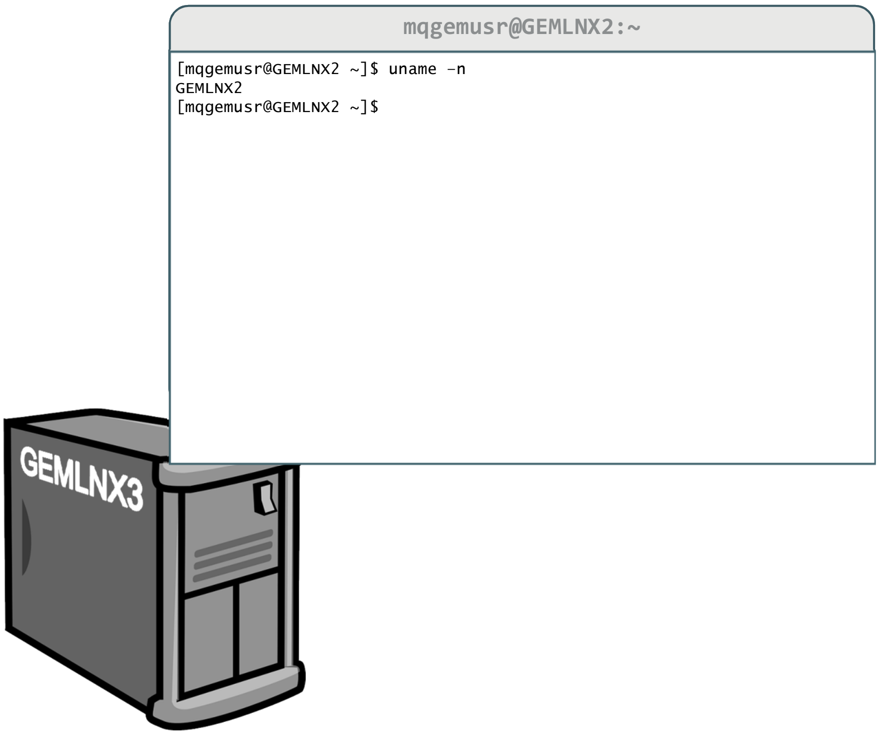
T

E

S

- Just like painting and decorating, or cooking and baking, preparation is everything when ensuring that your RDQM experience is an “Easy HA” one.
- If you get the preparation correct, it truly can be an easy experience.

Preparation 1: host name



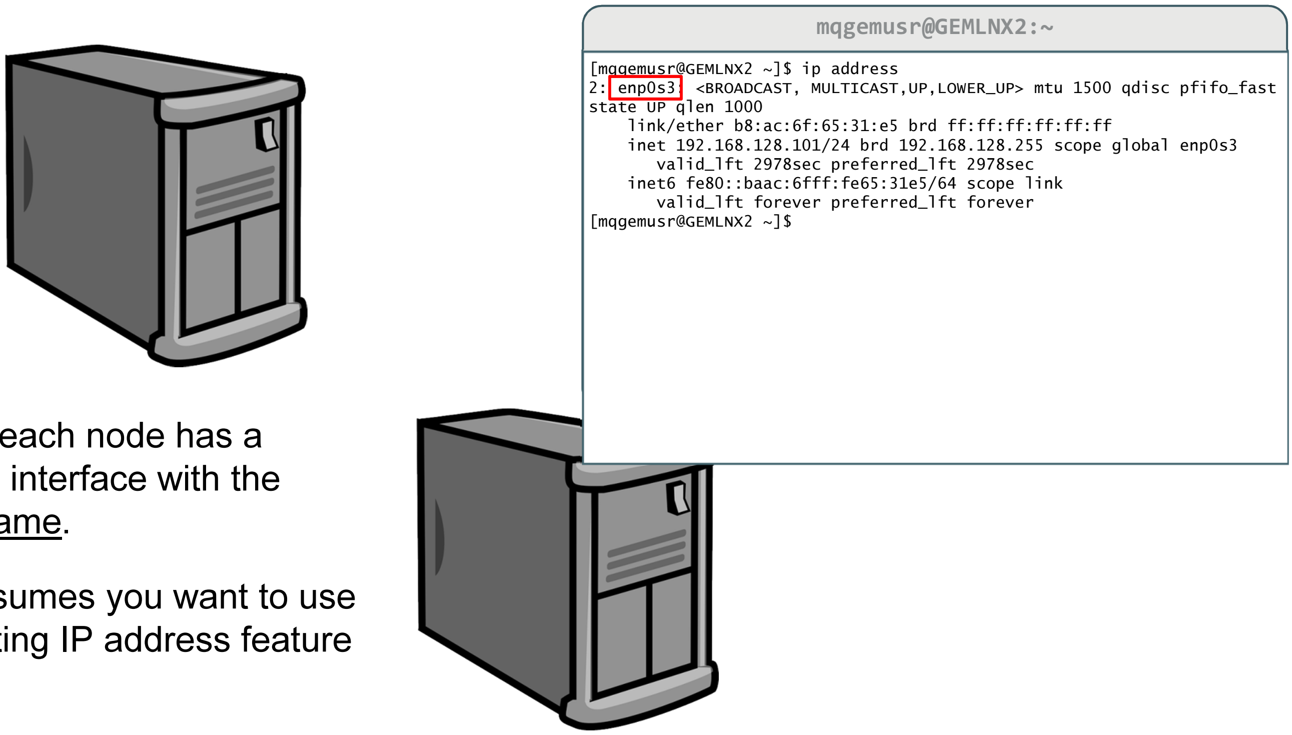
```
[mqgemusr@GEMLNX2 ~]$ uname -n
GEMLNX2
[mqgemusr@GEMLNX2 ~]$
```

- Ensure each node has a valid, and different, internet host name.

Preparation 2: Network Connectivity

- Ensure each node has network connectivity at startup
  - I suspect this is only likely to be a problem if you are using a virtualised environment

Preparation 3: Network Interface



mqgemusr@GEMLNX2:~

```
[mqgemusr@GEMLNX2 ~]$ ip address
2: enp0s3: <BROADCAST, MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP qlen 1000
    link/ether b8:ac:6f:65:31:e5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.128.101/24 brd 192.168.128.255 scope global enp0s3
       valid_lft 2978sec preferred_lft 2978sec
    inet6 fe80::baac:6fff:fe65:31e5/64 scope link
       valid_lft forever preferred_lft forever
[mqgemusr@GEMLNX2 ~]$
```

- Ensure each node has a network interface with the same name.
- This assumes you want to use the floating IP address feature

Preparation 4: Group membership

- Your `mqm` group user needs to also be a member of the `haclient` group.
  - `haclient` is created during the installation of Pacemaker
- If you don’t do this, you’ll see errors like the following when you try to `strmqm` or `endmqm`, even running as an mqm member, or running as root!

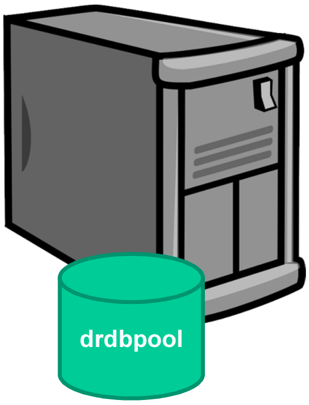
```
AMQ7077E: You are not authorized to perform the requested operation.
```

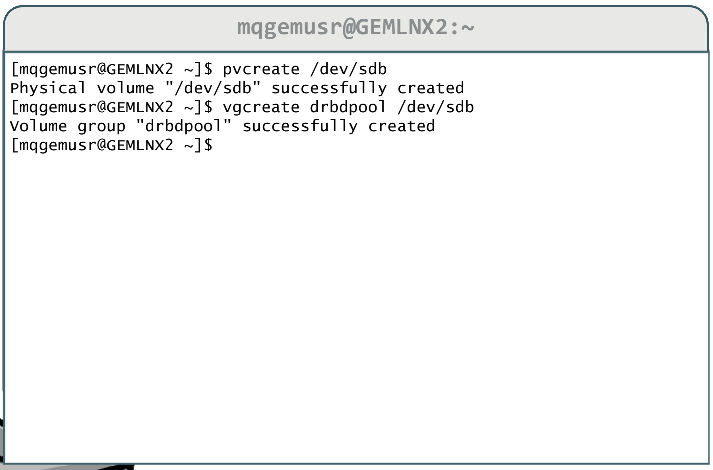
- Use the following command:-

```
usermod -a -G haclient $USER
```

- Log off and on again to ensure this change is picked up – you’ll need it soon enough.

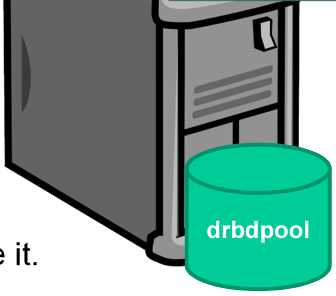
Preparation 5: Physical Volume





```
[mqgemusr@GEMLNX2 ~]$ pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created
[mqgemusr@GEMLNX2 ~]$ vgcreate drbdpool /dev/sdb
Volume group "drbdpool" successfully created
[mqgemusr@GEMLNX2 ~]$
```

- Each node requires a volume group named drbdpool.
- Depending on the disks you have available, will depend on how you create this.
  - Here is one example way to create it.



Preparation 6: Firewall

- You need to configure the firewall for RDQM operation. IBM MQ supplies a script to help you with this. The script utilises three XML files that you can edit to your own specific needs.
  `$MQ_INSTALLATION_PATH/samp/rdqm/firewalld/services/*.xml`
- Edit `rdqm-mq.xml` to ensure it contains the port number your listener is going to use.

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>RDQM MQ firewalld service</short>
  <description>This allows you to use MQ.</description>
  <port protocol="tcp" port="1414"/>
</service>
```

- There's also `rdqm-pacemaker.xml`
  - `No changes needed`

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>RDQM Pacemaker firewalld service</short>
  <description>This allows you to use Pacemaker.</description>
  <port protocol="udp" port="5404-5407"/>
</service>
```

- The other file that you might be interested in later is `rdqm-drbd.xml`
  - No changes needed
  - Just take note of the port range

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>RDQM DRBD firewalld service</short>
  <description>This allows you to use DRBD.</description>
  <port protocol="tcp" port="7000-7100"/>
</service>
```

- Run the script (likely need to be root) to configure the firewall
  `$MQ_INSTALLATION_PATH/samp/rdqm/firewalld/configure.sh`

Preparation 7: SELinux

- You need to know the current mode of SELinux, i.e. is it enforcing, permissive, or disabled.
- If you don't know, issue the following command.

```
getenforce
```

- If the system uses SELinux in a mode other than permissive, you must run the following command.

```
semanage permissive -a drbd_t
```

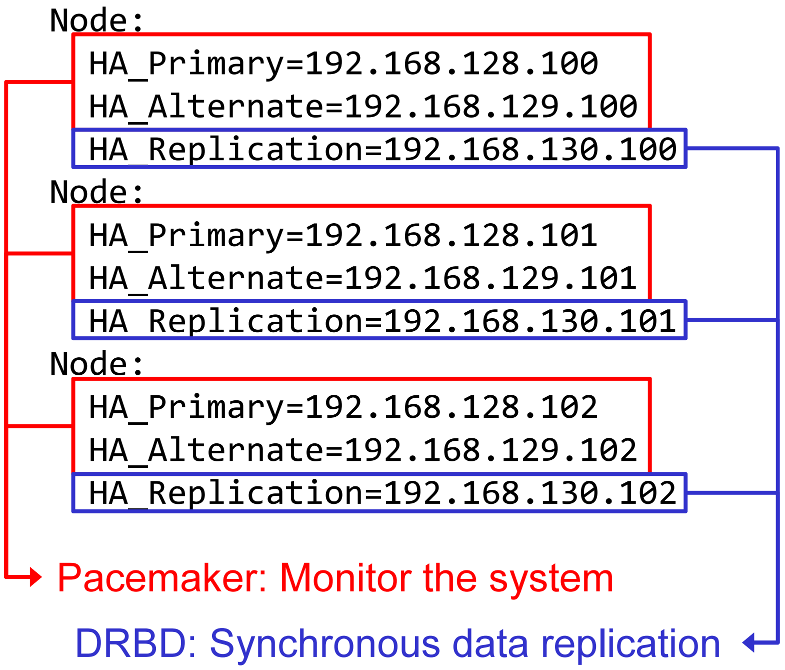
Preparation - Notes

N
O
T
E
S

- These 7 steps must be done on each of your three nodes.
- Make sure that all these steps are done ahead of time and you will have a much smoother RDQM experience.

Define the HA Group

- /var/mqm/rdqm.ini (maximum)
- /var/mqm/rdqm.ini (minimum)



```
Node:
  HA_Primary=192.168.128.100
  HA_Alternate=192.168.129.100
  HA_Replication=192.168.130.100
Node:
  HA_Primary=192.168.128.101
  HA_Alternate=192.168.129.101
  HA_Replication=192.168.130.101
Node:
  HA_Primary=192.168.128.102
  HA_Alternate=192.168.129.102
  HA_Replication=192.168.130.102
```

Pacemaker: Monitor the system

DRBD: Synchronous data replication

```
Node:


  HA_Replication=192.168.130.100
Node:


  HA_Replication=192.168.130.101
Node:


  HA_Replication=192.168.130.102
```

Define the HA Group - Notes

N O T E S

- To define the HA Group, you need to know the IP addresses for each server. A file called /var/mqm/rdqm.ini is where you supply the IP addresses for all of the nodes in the group.
- An RDQM Queue Manager can use up to three IP addresses on each node.
- The HA_Primary and HA_Alternate interfaces are used for the Pacemaker to monitor the system, and the HA_Replication interface is used by DRBD for synchronous data replication. This interface must have sufficient bandwidth to support replication requirements given the expected workload of all the RDQMs running in the HA Group. For maximum fault tolerance, these interfaces should be independent Network Interface Cards (NICs).
- It is also possible to use a Pacemaker cluster without the HA_Alternate interface, so just running two interfaces, or in fact, with a single IP address in which case the HA_Replication interface is all that is needed to be defined.

Password-less SSH

- Save yourself running the following commands on each of three machines
  - `rdqmadm`
  - `crtmqm`
  - `dltmqm`

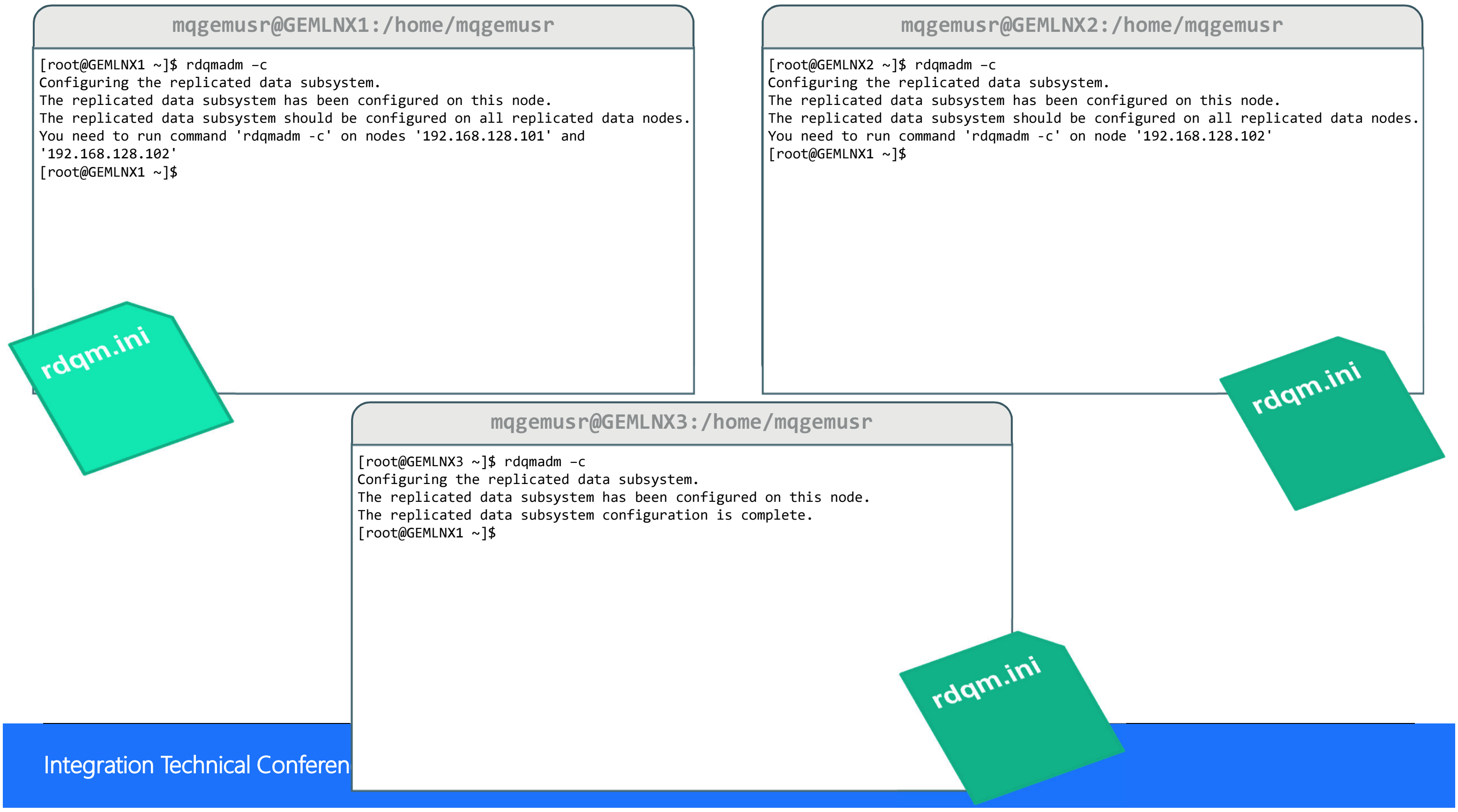
- I did not use this

Password-less SSH - Notes

N O T E S

- It is worth mentioning at this point that instead of running the rdqmadm command individually on each system, you can configure password-less SSH for your user id for each machine. Then you only need to run the rdqmadm command once to set up all three machines.
- As this is a user experience, I only mention this because I did not make use of this. I felt I would learn more about RDQM by knowing exactly what was going on, on each machine, and so chose to do each part on each machine myself.

Creating an RDQM HA subsystem


mqgemusr@GEMLNX1:/home/mqgemusr
rdqm.ini
mqgemusr@GEMLNX2:/home/mqgemusr
rdqm.ini
mqgemusr@GEMLNX3:/home/mqgemusr
rdqm.ini

```
[root@GEMLNX1 ~]$ rdqmadm -c
Configuring the replicated data subsystem.
The replicated data subsystem has been configured on this node.
The replicated data subsystem should be configured on all replicated data nodes.
You need to run command 'rdqmadm -c' on nodes '192.168.128.101' and
'192.168.128.102'
[root@GEMLNX1 ~]$
```

```
[root@GEMLNX2 ~]$ rdqmadm -c
Configuring the replicated data subsystem.
The replicated data subsystem has been configured on this node.
The replicated data subsystem should be configured on all replicated data nodes.
You need to run command 'rdqmadm -c' on node '192.168.128.102'
[root@GEMLNX1 ~]$
```

```
[root@GEMLNX3 ~]$ rdqmadm -c
Configuring the replicated data subsystem.
The replicated data subsystem has been configured on this node.
The replicated data subsystem configuration is complete.
[root@GEMLNX1 ~]$
```

Creating an RDQM HA subsystem

N

O

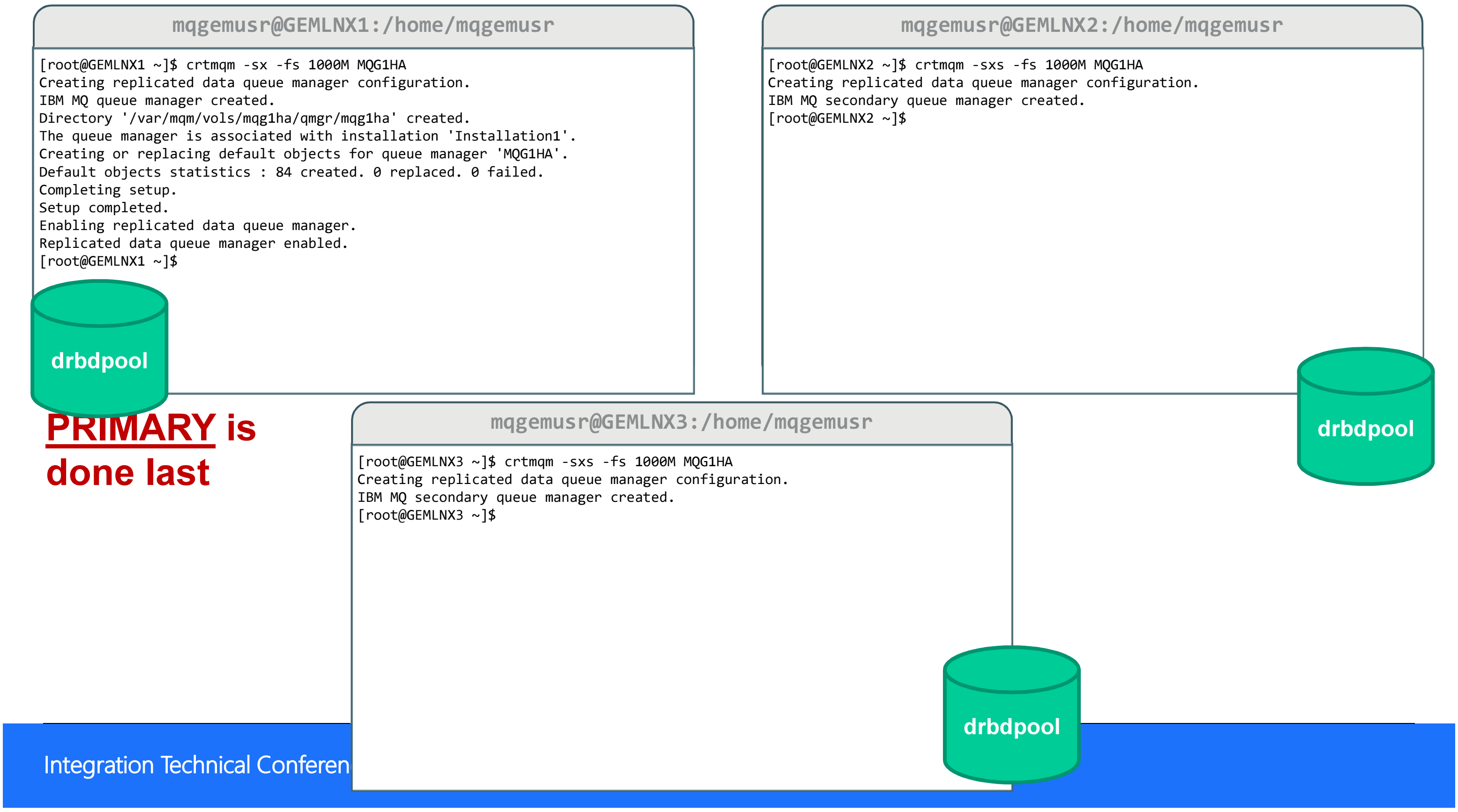
T

E

S

- The /var/mqm/rdqm.ini file that you edited now needs to be copied to each machine.
- Once you have copied it, you run the rdqmadm command with the –c flag to create the RDQM subsystem. This command needs to be run as root. In my case I did this individually on each machine rather than using password-less SSH and in that case you get to the see the messages as each machine is aware of what is currently set up and tells you very clearly what is left to do.
- You will see output like that shown, with the remaining nodes' IP addresses mentioned in each case.

Creating an RDQM Queue Manager - crtmqm



```
[root@GEMLNX1 ~]$ crtmqm -sx -fs 1000M MQG1HA
Creating replicated data queue manager configuration.
IBM MQ queue manager created.
Directory '/var/mqm/vols/mqg1ha/qmgr/mqg1ha' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'MQG1HA'.
Default objects statistics : 84 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
Enabling replicated data queue manager.
Replicated data queue manager enabled.
[root@GEMLNX1 ~]$
```

```
[root@GEMLNX2 ~]$ crtmqm -sxs -fs 1000M MQG1HA
Creating replicated data queue manager configuration.
IBM MQ secondary queue manager created.
[root@GEMLNX2 ~]$
```

```
[root@GEMLNX3 ~]$ crtmqm -sxs -fs 1000M MQG1HA
Creating replicated data queue manager configuration.
IBM MQ secondary queue manager created.
[root@GEMLNX3 ~]$
```

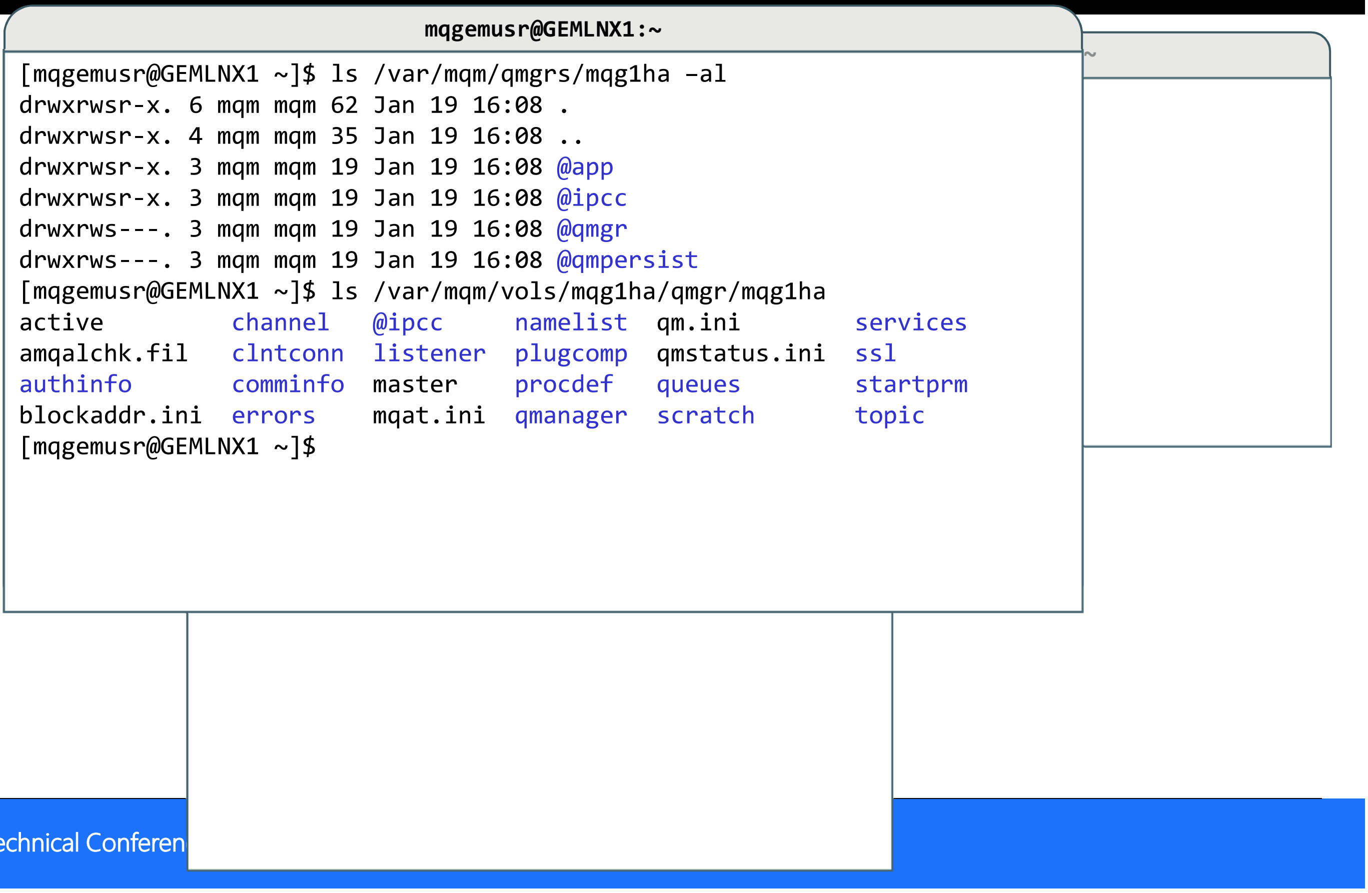
PRIMARY is done last

Creating an RDQM Queue Manager - crtmqm - Notes

N O T E S

- You need to run the crtmqm command on each machine in your cluster. Before you start, decide which of your three machines is going to be the primary node. Do that machine last.
- You will have to issue these commands as root (or be granted sudo access to the specific commands).
- If your machine is a secondary node in the Pacemaker cluster, issue the crtmqm command ensuring the value supplied in the -fs parameter is appropriate for the usable size of your drbdpool volume. It specifies the size of the logical volume which is created. The -fs parameter can be specified in gigabytes by just supplying a number, and in megabytes by adding an 'M' after the number. The default value if you do not use the -fs parameter is 3 gigabytes. The file system size must be specified the same on all three nodes.
- Normally I am sure you are used to looking /var/mqm/qmgrs/MQG1HA to find the queue manager data, the errors directory for your AMQERR01.LOG files and so on. For an RDQM queue manager, things are a little different. If you navigate to the normal data directory (although note that the queue manager name is folded to lower case) and take a look, you will see very little there.

RDQM Queue Manager data directory



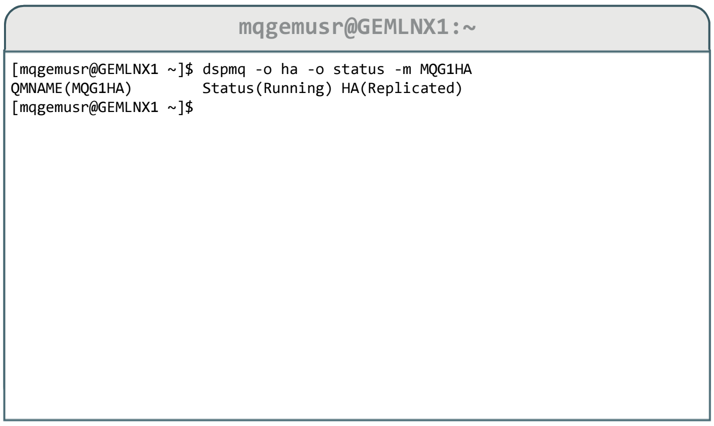
```
[mqgemusr@GEMLNX1 ~]$ ls /var/mqm/qmgrs/mqg1ha -al
drwxrwsr-x. 6 mqm mqm 62 Jan 19 16:08 .
drwxrwsr-x. 4 mqm mqm 35 Jan 19 16:08 ..
drwxrwsr-x. 3 mqm mqm 19 Jan 19 16:08 @app
drwxrwsr-x. 3 mqm mqm 19 Jan 19 16:08 @ipcc
drwxrws---. 3 mqm mqm 19 Jan 19 16:08 @qmgr
drwxrws---. 3 mqm mqm 19 Jan 19 16:08 @qmpersist
[mqgemusr@GEMLNX1 ~]$ ls /var/mqm/vols/mqg1ha/qmgr/mqg1ha
active         channel   @ipcc     namelist  qm.ini        services
amqalchk.fil   clntconn  listener  plugcomp  qmstatus.ini  ssl
authinfo       comminfo  master    procdef   queues        startprm
blockaddr.ini  errors    mqat.ini  qmanager  scratch       topic
[mqgemusr@GEMLNX1 ~]$
```

RDQM Queue Manager data directory - Notes

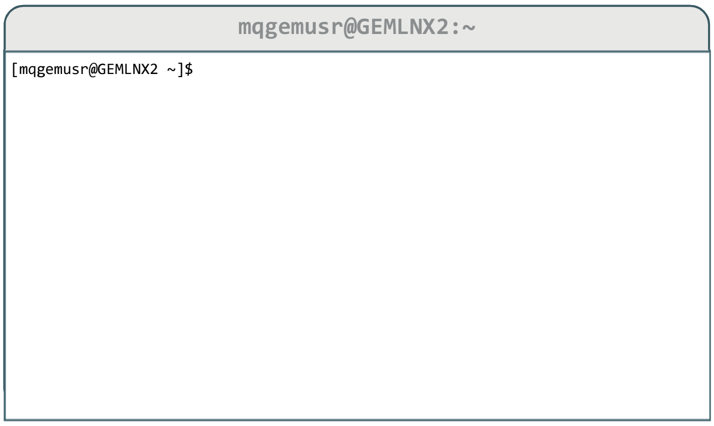
N O T E S

- If you look in the “normal” queue manage data directory after creating an RDQM queue manager, you won’t see much.
- The data is actually somewhere else.
- The output from the `crtmqm` command told you where to look. If you look in `/var/mqm/vols/mqg1ha/qmgr/mqg1ha` (taking note of the queue manager name folded to lower case) and you'll see the familiar set of queue manager data.

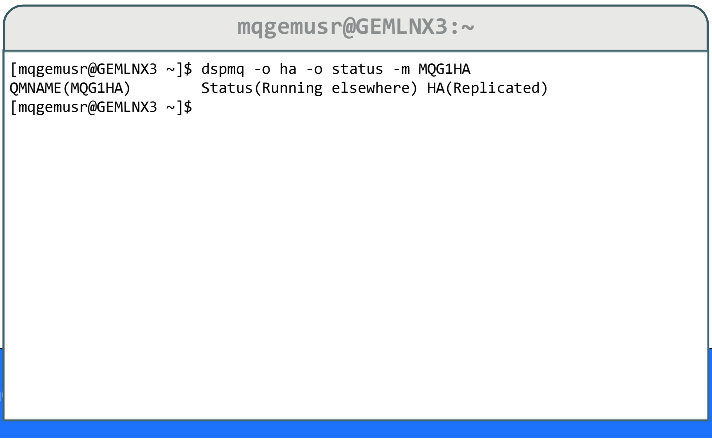
RDQM HA Queue Manager - Running



```
[mqgemusr@GEMLNX1 ~]$ dspmq -o ha -o status -m MQG1HA
QMNAME(MQG1HA)         Status(Running) HA(Replicated)
[mqgemusr@GEMLNX1 ~]$
```



```
[mqgemusr@GEMLNX2 ~]$
```



```
[mqgemusr@GEMLNX3 ~]$ dspmq -o ha -o status -m MQG1HA
QMNAME(MQG1HA)         Status(Running elsewhere) HA(Replicated)
[mqgemusr@GEMLNX3 ~]$
```

RDQM HA Queue Manager - Running - Notes

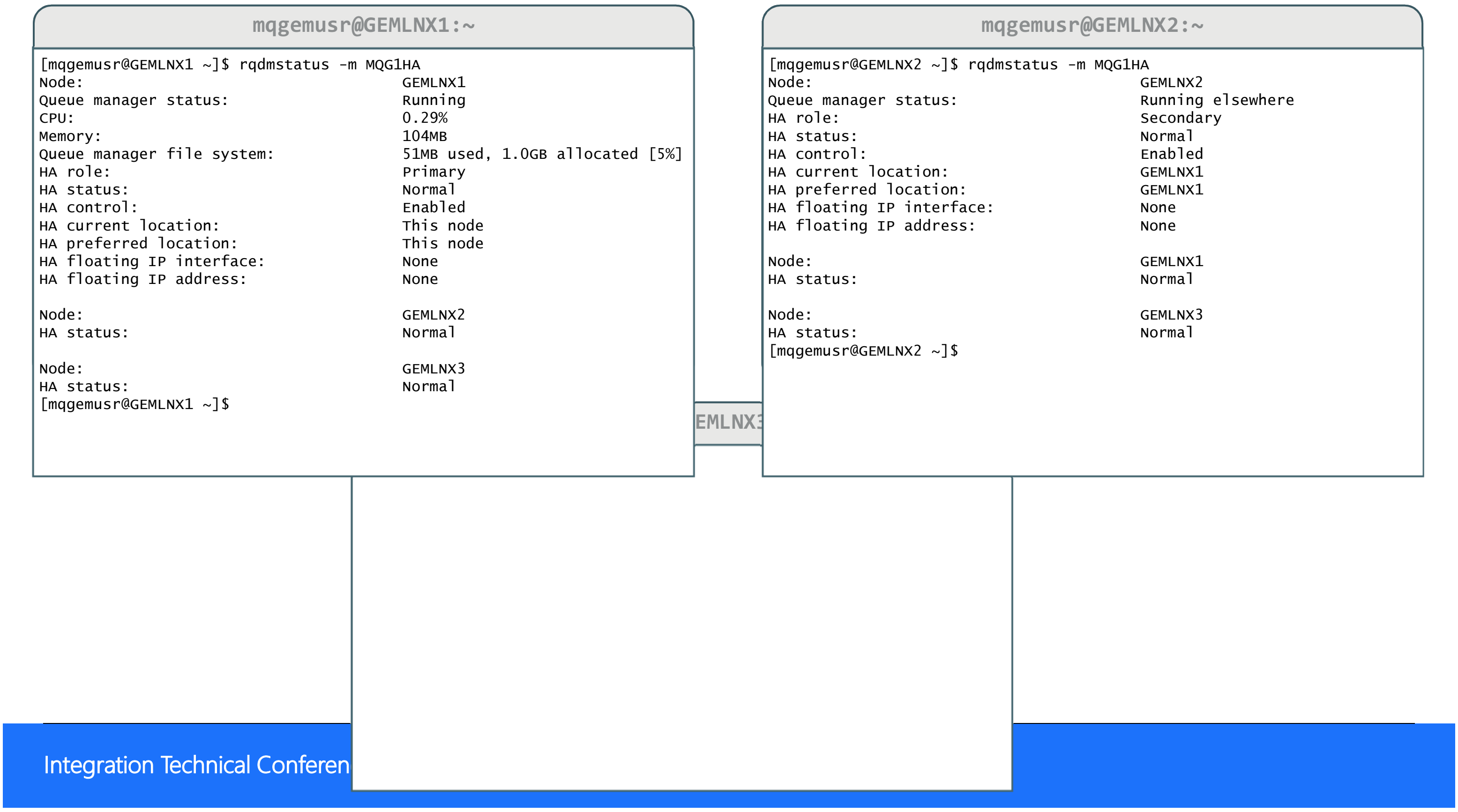
N
O
T
E
S

- One thing that you'll notice about an RDQM queue manager is that as soon as you have created it, it is running!
- Normally a queue manager is temporarily started in order to create the default objects and then shut down again, and it is up to you to issue the strmqm command to start it up. This is not the case for an RDQM queue manager. It does startup briefly to create the default objects, and then ends, but it is then started by Pacemaker on its preferred node – the primary system.
- When the last two messages from the crtmqm command said:-

```
Enabling replicated data queue manager.
Replicated data queue manager enabled.
```

- It was telling you that the queue manager was now running.

RDQM HA Status



```
[mqgemusr@GEMLNX1 ~]$ rqdmstatus -m MQG1HA
Node:                                    GEMLNX1
Queue manager status:                    Running
CPU:                                     0.29%
Memory:                                  104MB
Queue manager file system:               51MB used, 1.0GB allocated [5%]
HA role:                                 Primary
HA status:                               Normal
HA control:                              Enabled
HA current location:                     This node
HA preferred location:                   This node
HA floating IP interface:                None
HA floating IP address:                  None

Node:                                    GEMLNX2
HA status:                               Normal

Node:                                    GEMLNX3
HA status:                               Normal
[mqgemusr@GEMLNX1 ~]$
```

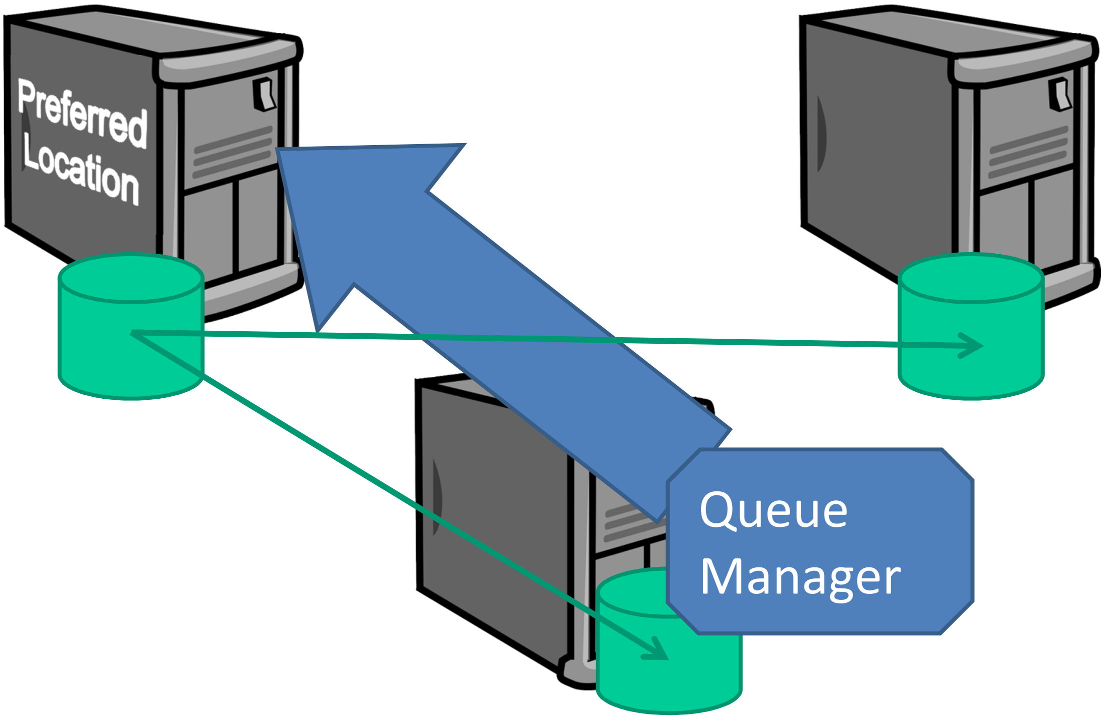
```
[mqgemusr@GEMLNX2 ~]$ rqdmstatus -m MQG1HA
Node:                                    GEMLNX2
Queue manager status:                    Running elsewhere
HA role:                                 Secondary
HA status:                               Normal
HA control:                              Enabled
HA current location:                     GEMLNX1
HA preferred location:                   GEMLNX1
HA floating IP interface:                None
HA floating IP address:                  None

Node:                                    GEMLNX1
HA status:                               Normal

Node:                                    GEMLNX3
HA status:                               Normal
[mqgemusr@GEMLNX2 ~]$
```

RDQM HA Status - Notes

NOTES

- In addition to the new flag on the dspmq command (-o ha), there is also another command that gives you lots of information about the status of the RDQM HA Group – rdqmstatus.
- When issued on the node where the queue manager is currently running, it will show details about the queue manager’s CPU, memory and file system usage.
- It shows you which node is the primary node for this queue manager, and the status of the queue manager, either running here or elsewhere.
- From any of the RDQM nodes, you can learn where the queue manager is currently running.

RDQM HA preferred location

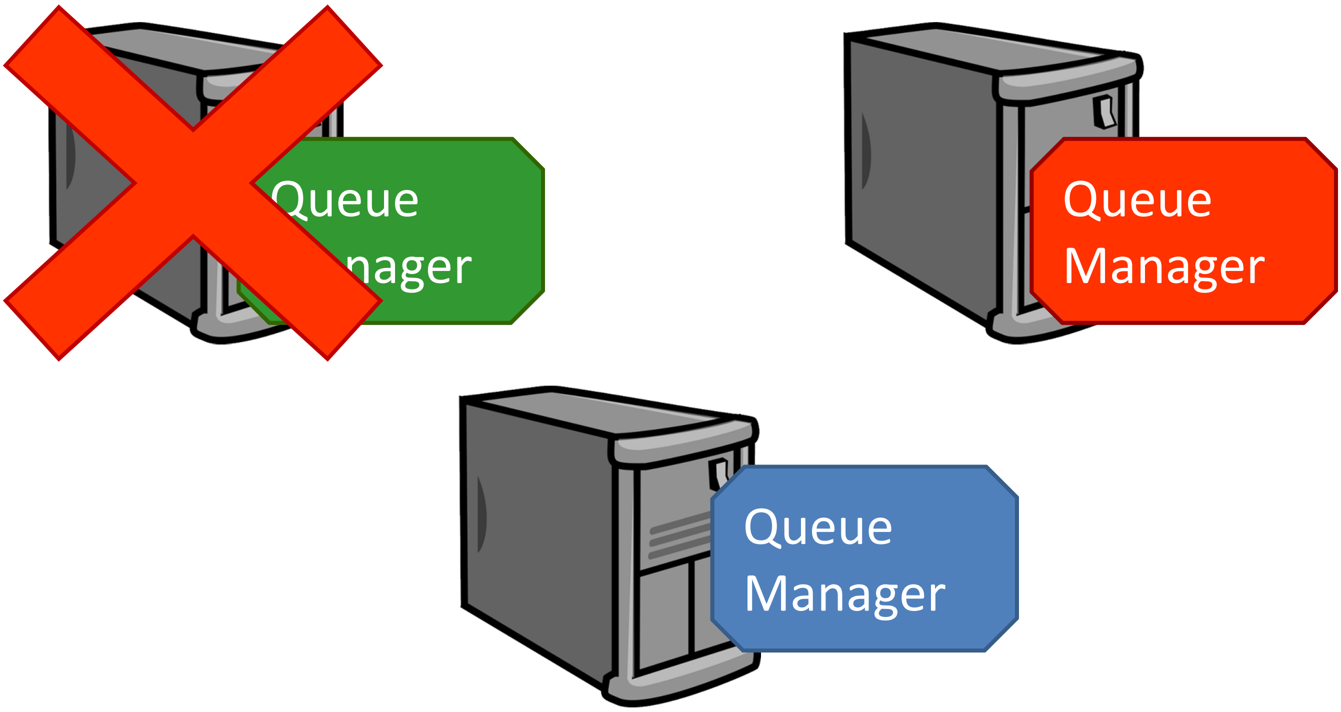


RDQM HA preferred location - Notes

N O T E S

- An RDQM queue manager starts life with a preferred location of the primary node. That was the node where you ran the `crtmqm` command last.
- If you lose the node that the queue manager is currently running on (or suspend it), the queue manager will be moved to another node.
- Once the preferred node comes back online (or is resumed if you suspended it), the queue manager will be moved back to the preferred node.

Why have a preferred location

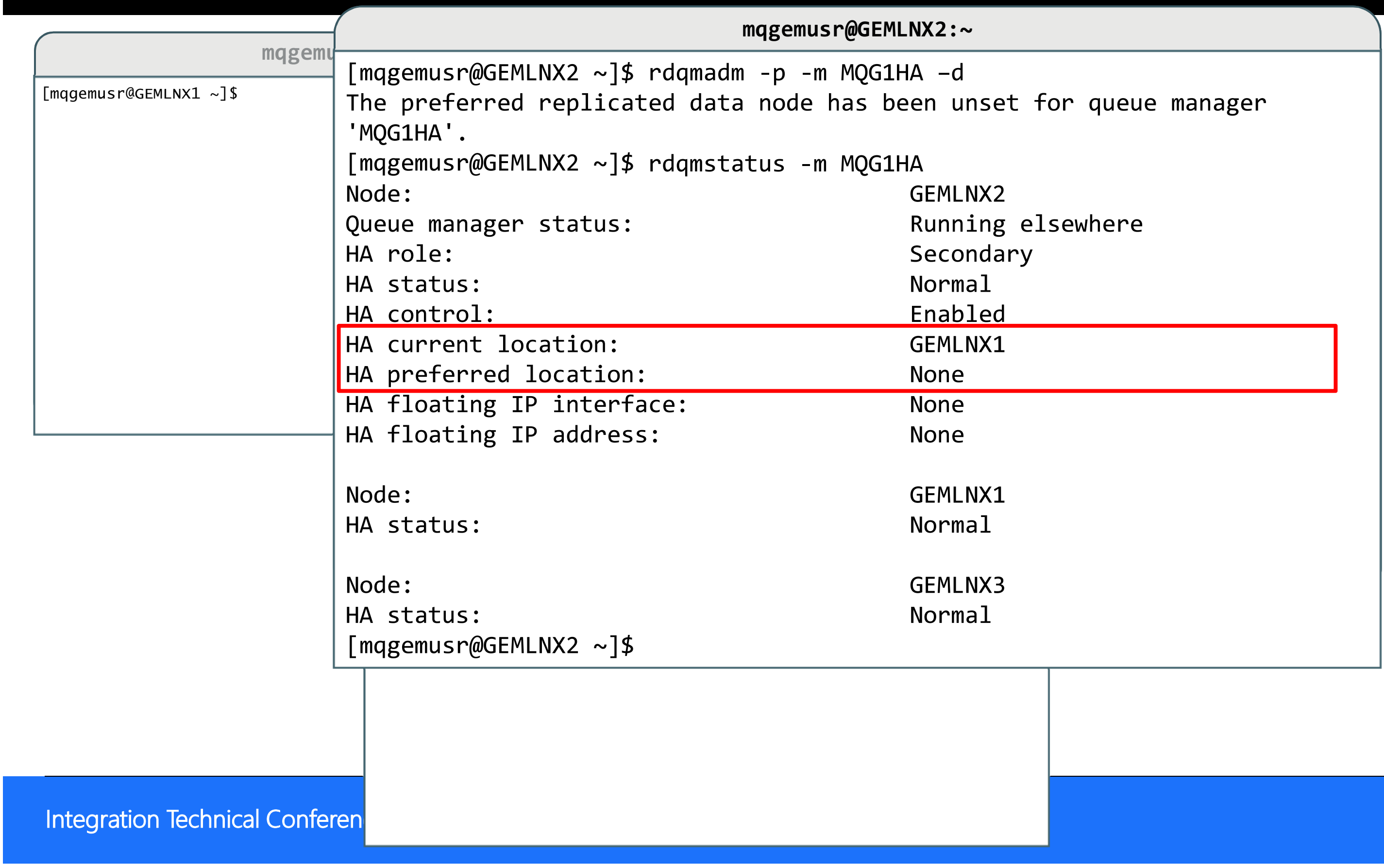


Why have a preferred location - Notes

N
O
T
E
S

- With three servers needed in order to run an RDQM, it is likely that you will have a set up with three queue managers, where the normal state of affairs is for one queue manager to run on each machine.
- If one of the machines goes offline (planned or unplanned) the queue manager who prefers that location moves to one of the remaining machines. Once the offline machine comes back, it would be better to be spreading the work evenly again, and so having that queue manager return to its preferred location is appropriate.
- However, moving a queue manager back to its preferred location does take an outage while it is stopped on one machine and started on another. While such an outage is acceptable when it means the queue manager is being moved to an online machine, it may be less acceptable if it was already running perfectly happily somewhere else and the outage is simply to move it to a different machine.

Controlling preferred location



```
[mqgemusr@GEMLNX1 ~]$
```

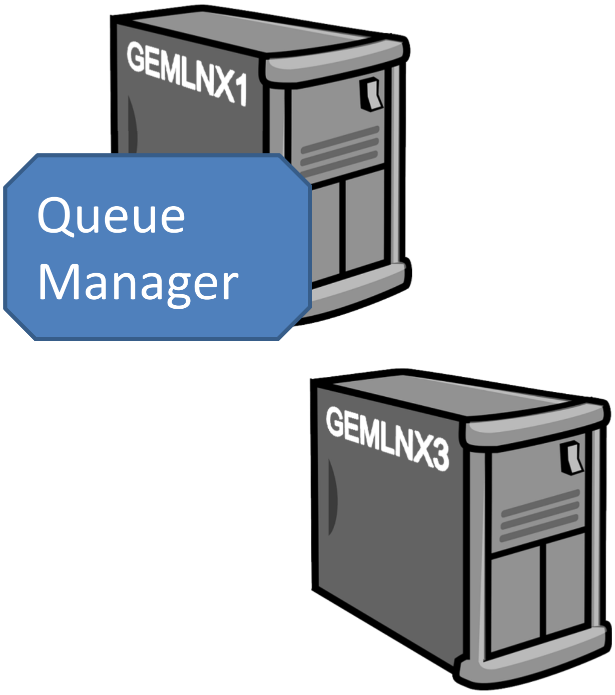
```
[mqgemusr@GEMLNX2 ~]$ rdqmadm -p -m MQG1HA -d
The preferred replicated data node has been unset for queue manager
'MQG1HA'.
[mqgemusr@GEMLNX2 ~]$ rdqmstatus -m MQG1HA
Node:                                     GEMLNX2
Queue manager status:                     Running elsewhere
HA role:                                  Secondary
HA status:                                Normal
HA control:                               Enabled
HA current location:                      GEMLNX1
HA preferred location:                    None
HA floating IP interface:                 None
HA floating IP address:                   None

Node:                                     GEMLNX1
HA status:                                Normal

Node:                                     GEMLNX3
HA status:                                Normal
[mqgemusr@GEMLNX2 ~]$
```

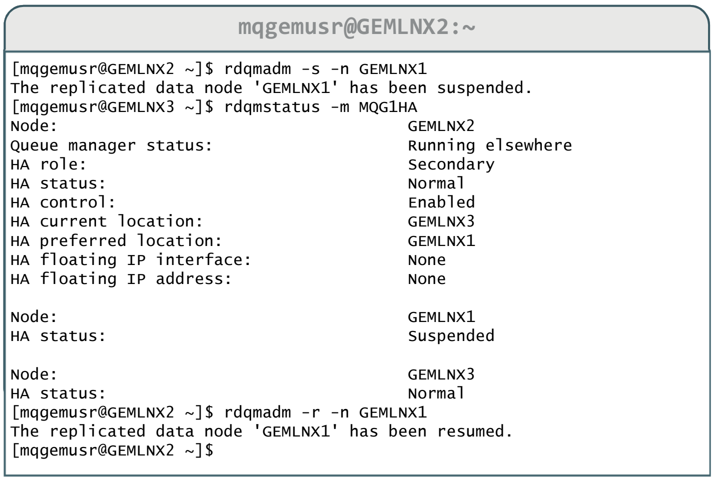
Controlling preferred location - Notes

N O T E S

- As you saw earlier, an RDQM HA queue manager begins life with a preference to run on the primary node.
- You may not have a preference for where the queue manager runs, or you may not want this preference to be automatically switched to in the middle of the day! In which case, you can remove this preference by using the rdqmadm command as shown.

Suspend/Resume a node





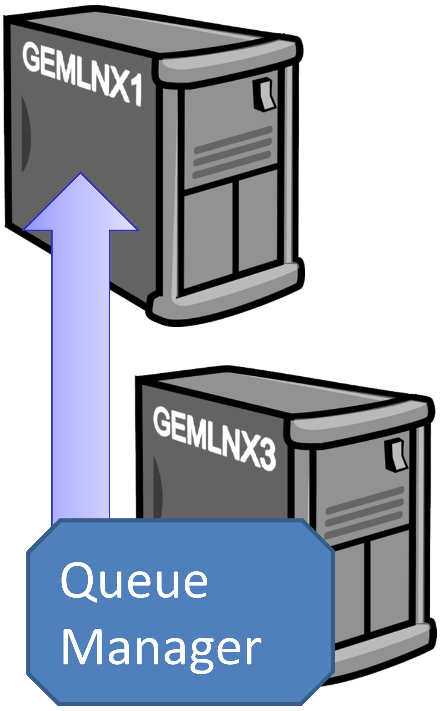
```
[mqgemusr@GEMLNX2 ~]$ rdqmadm -s -n GEMLNX1
The replicated data node 'GEMLNX1' has been suspended.
[mqgemusr@GEMLNX3 ~]$ rdqmstatus -m MQG1HA
Node:                                    GEMLNX2
Queue manager status:                    Running elsewhere
HA role:                                 Secondary
HA status:                               Normal
HA control:                              Enabled
HA current location:                     GEMLNX3
HA preferred location:                   GEMLNX1
HA floating IP interface:                None
HA floating IP address:                  None

Node:                                    GEMLNX1
HA status:                               Suspended

Node:                                    GEMLNX3
HA status:                               Normal
[mqgemusr@GEMLNX2 ~]$ rdqmadm -r -n GEMLNX1
The replicated data node 'GEMLNX1' has been resumed.
[mqgemusr@GEMLNX2 ~]$
```

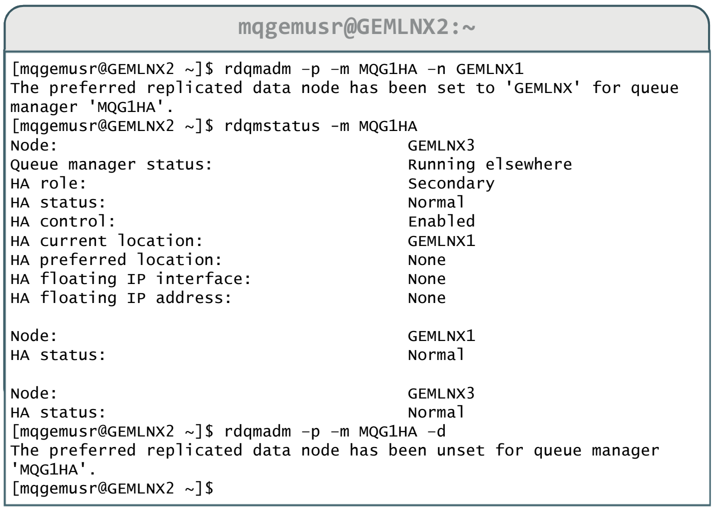
Suspend/Resume a node - Notes

NOTES

- If you have need of a planned outage for one of the nodes in your HA Group, you can use the rdqmadm command to suspend and resume a node. This command can be issued from any of the nodes to suspend/resume any of the other nodes.

Manually move the queue manager





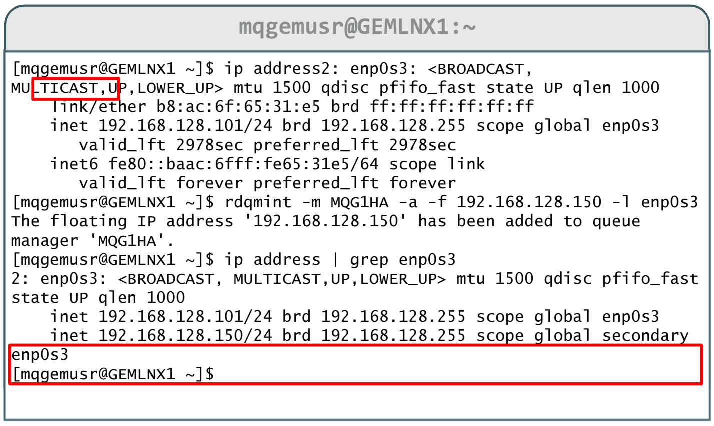
```
[mqgemusr@GEMLNX2 ~]$ rdqmadm -p -m MQG1HA -n GEMLNX1
The preferred replicated data node has been set to 'GEMLNX' for queue
manager 'MQG1HA'.
[mqgemusr@GEMLNX2 ~]$ rdqmstatus -m MQG1HA
Node:                                   GEMLNX3
Queue manager status:                   Running elsewhere
HA role:                                Secondary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    GEMLNX1
HA preferred location:                  None
HA floating IP interface:               None
HA floating IP address:                 None

Node:                                   GEMLNX1
HA status:                              Normal

Node:                                   GEMLNX3
HA status:                              Normal
[mqgemusr@GEMLNX2 ~]$ rdqmadm -p -m MQG1HA -d
The preferred replicated data node has been unset for queue manager
'MQG1HA'.
[mqgemusr@GEMLNX2 ~]$
```

Manually move the queue manager - Notes

N O T E S

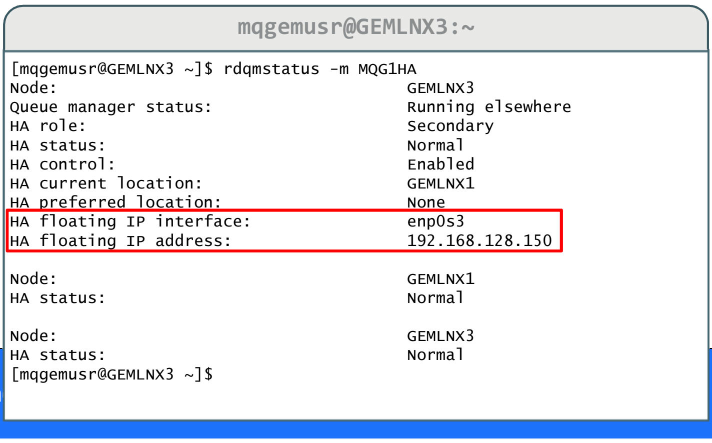
- If you have a preferred location for the queue manager, but only want to end it and start it on the preferred location on your own terms, you would spend most of the time with no preferred location set.
- There is no specific command to tell a queue manager to move when you decide it is a good time to move it, however, you can achieve it by briefly setting the preferred location.
- Once it has moved, you can unset the preference again, putting you back in control of when it moves.

Floating IP address



```
[mqgemusr@GEMLNX1 ~]$ ip address2: enp0s3: <BROADCAST,
MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether b8:ac:6f:65:31:e5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.128.101/24 brd 192.168.128.255 scope global enp0s3
       valid_lft 2978sec preferred_lft 2978sec
    inet6 fe80::baac:6fff:fe65:31e5/64 scope link
       valid_lft forever preferred_lft forever
[mqgemusr@GEMLNX1 ~]$ rdqmint -m MQG1HA -a -f 192.168.128.150 -l enp0s3
The floating IP address '192.168.128.150' has been added to queue
manager 'MQG1HA'.
[mqgemusr@GEMLNX1 ~]$ ip address | grep enp0s3
2: enp0s3: <BROADCAST, MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP qlen 1000
    inet 192.168.128.101/24 brd 192.168.128.255 scope global enp0s3
    inet 192.168.128.150/24 brd 192.168.128.255 scope global secondary
enp0s3
[mqgemusr@GEMLNX1 ~]$
```

```
[mqgemusr@GEMLNX2 ~]$
```



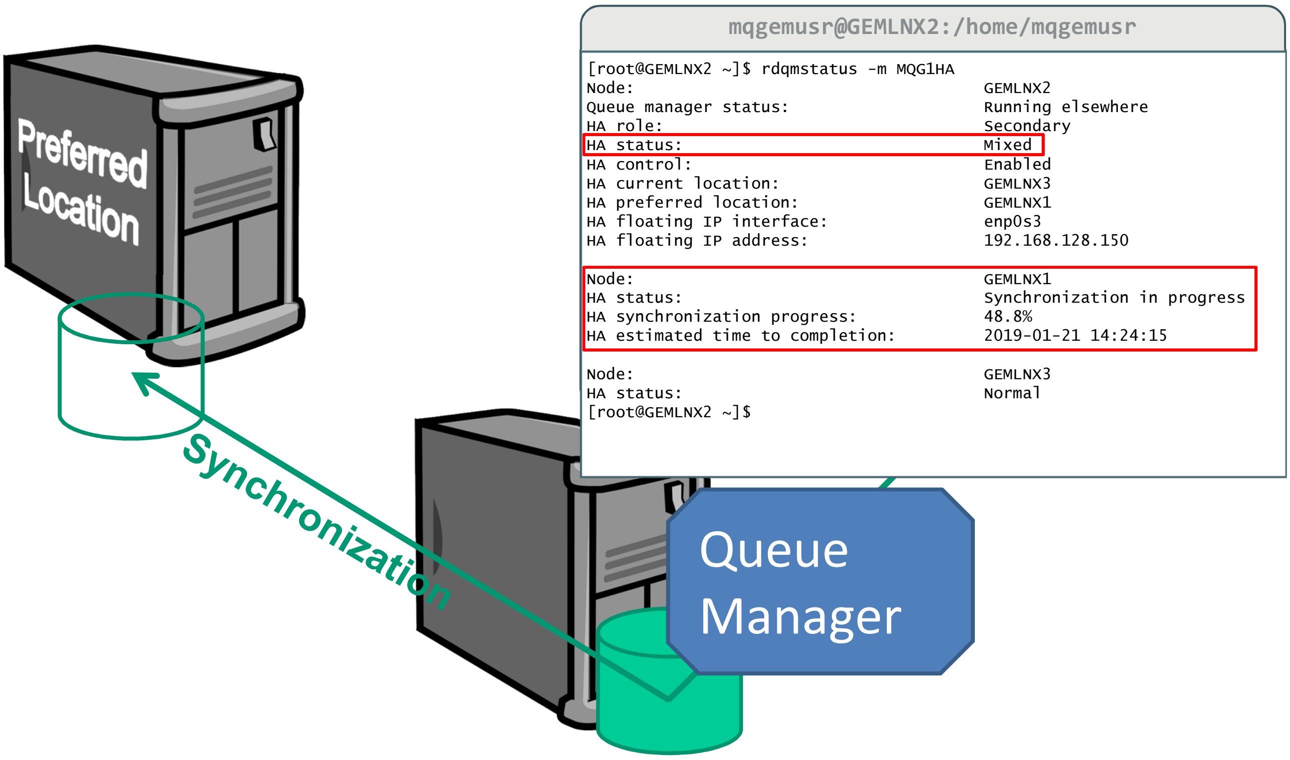
```
[mqgemusr@GEMLNX3 ~]$ rdqmstatus -m MQG1HA
Node:                                   GEMLNX3
Queue manager status:                   Running elsewhere
HA role:                                Secondary
HA status:                              Normal
HA control:                             Enabled
HA current location:                    GEMLNX1
HA preferred location:                  None
HA floating IP interface:               enp0s3
HA floating IP address:                 192.168.128.150

Node:                                   GEMLNX1
HA status:                              Normal

Node:                                   GEMLNX3
HA status:                              Normal
[mqgemusr@GEMLNX3 ~]$
```

Floating IP address

N O T E S

- While it is possible to use a comma-separated CONNAME for your client applications to connect to the one of three machines where the queue manager might be, you can make things much easier by creating a Floating IP address which moves with the queue manager. So the client applications always use the same IP address, and that IP address always takes you straight to the queue manager.
- To create a floating IP address you will need to know the name of your network interface. If you don't know what it is, use the `ip address` command.
- You can issue command shown to create the floating IP address on any one of the nodes. All the nodes have the same interface name (remember the preparation steps?). The new IP address must belong to the same subnet as the physical interfaces on the three nodes.
- The `rdqmstatus` command will show details of the floating IP address once it is set up.

Synchronization of data after an outage



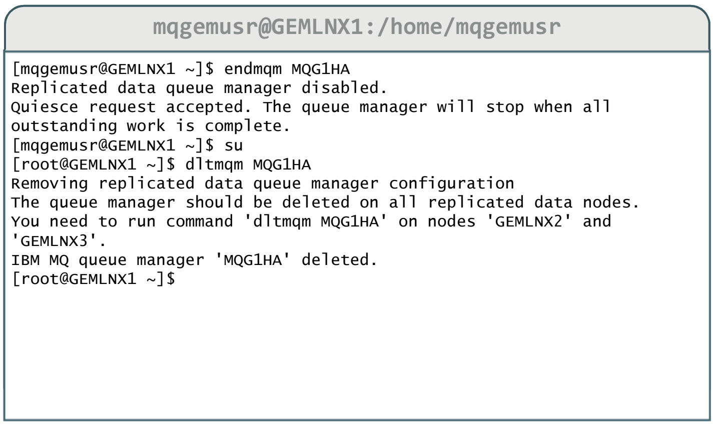
```
[root@GEMLNX2 ~]$ rdqmstatus -m MQG1HA
Node:                                    GEMLNX2
Queue manager status:                    Running elsewhere
HA role:                                 Secondary
HA status:                               Mixed
HA control:                              Enabled
HA current location:                     GEMLNX3
HA preferred location:                   GEMLNX1
HA floating IP interface:                enp0s3
HA floating IP address:                  192.168.128.150

Node:                                    GEMLNX1
HA status:                               Synchronization in progress
HA synchronization progress:             48.8%
HA estimated time to completion:         2019-01-21 14:24:15

Node:                                    GEMLNX3
HA status:                               Normal
[root@GEMLNX2 ~]$
```

Synchronization of data after an outage - Notes

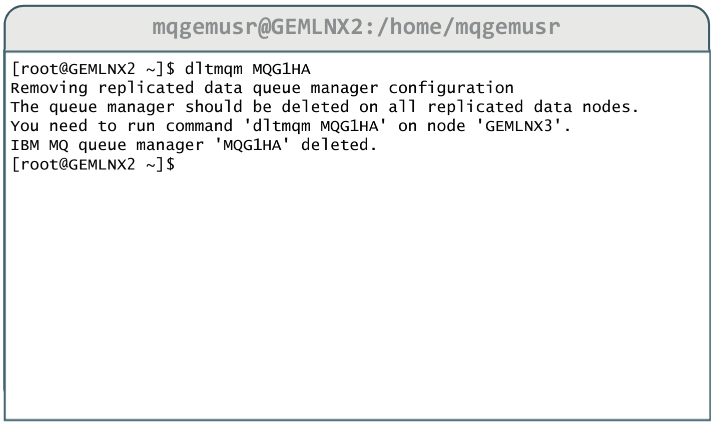
N O T E S

- If one of your nodes has been offline it cannot take over running the queue manager until its data has been brought up to date. It therefore needs to be synchronized with the data held on the node running the queue manager, your `rdqmstatus` command will show you that this is happening.
- If you issue this command as root, you will see further details showing the progress of the synchronization and the estimated time to completion.

Removing an RDQM HA Queue Manager

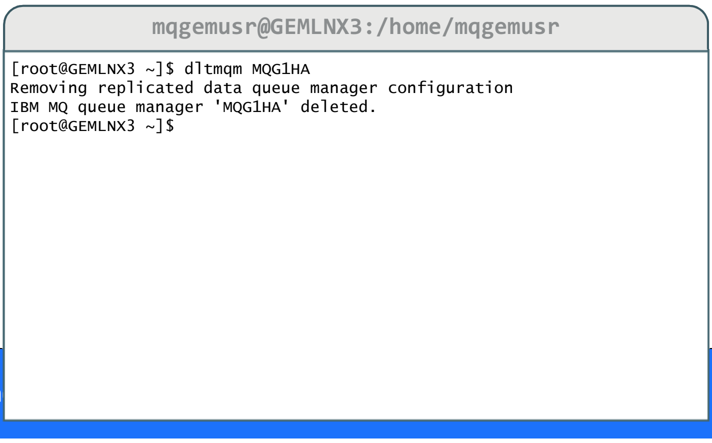


```
[mqgemusr@GEMLNX1 ~]$ endmqm MQG1HA
Replicated data queue manager disabled.
Quiesce request accepted. The queue manager will stop when all
outstanding work is complete.
[mqgemusr@GEMLNX1 ~]$ su
[root@GEMLNX1 ~]$ dltmqm MQG1HA
Removing replicated data queue manager configuration
The queue manager should be deleted on all replicated data nodes.
You need to run command 'dltmqm MQG1HA' on nodes 'GEMLNX2' and
'GEMLNX3'.
IBM MQ queue manager 'MQG1HA' deleted.
[root@GEMLNX1 ~]$
```



```
[root@GEMLNX2 ~]$ dltmqm MQG1HA
Removing replicated data queue manager configuration
The queue manager should be deleted on all replicated data nodes.
You need to run command 'dltmqm MQG1HA' on node 'GEMLNX3'.
IBM MQ queue manager 'MQG1HA' deleted.
[root@GEMLNX2 ~]$
```

PRIMARY is done first



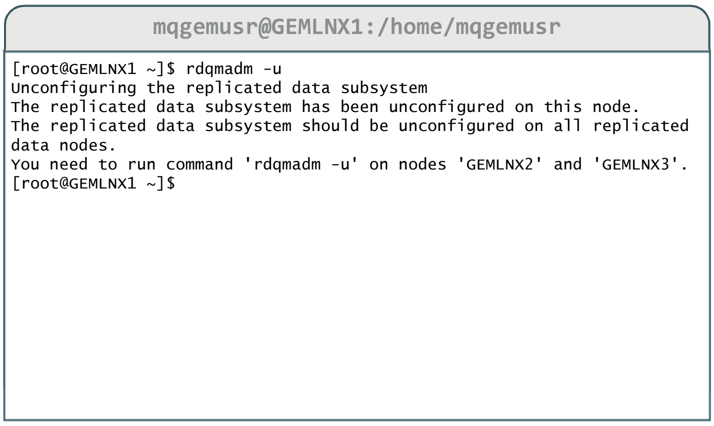
```
[root@GEMLNX3 ~]$ dltmqm MQG1HA
Removing replicated data queue manager configuration
IBM MQ queue manager 'MQG1HA' deleted.
[root@GEMLNX3 ~]$
```

Removing an RDQM HA Queue Manager - Notes

N
O
T
E
S

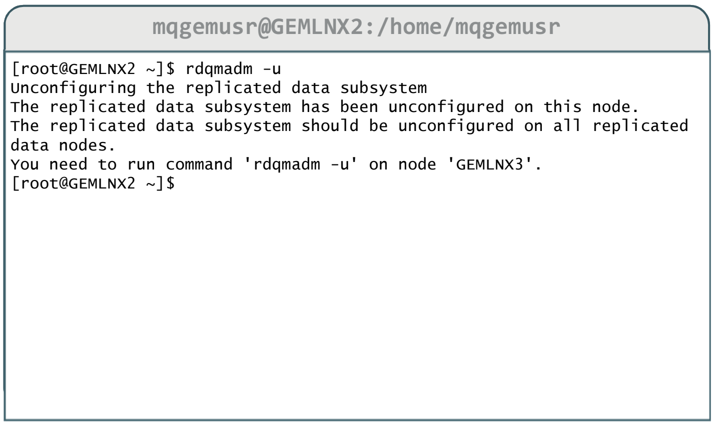
- Now we're going to run through the steps to remove an RDQM HA queue manager and replication data subsystem.
- When you end the queue manager with `endmqm` you will see a message which indicates that Pacemaker is no longer managing this queue manager – after all you wouldn't want it being automatically started up again when you deliberately stopped it!
- Once the queue manager has ended, you will delete it, first on the primary node, and then on both the secondary nodes. You will have to run the `dltmqm` command as root.

Removing replicated data subsystem



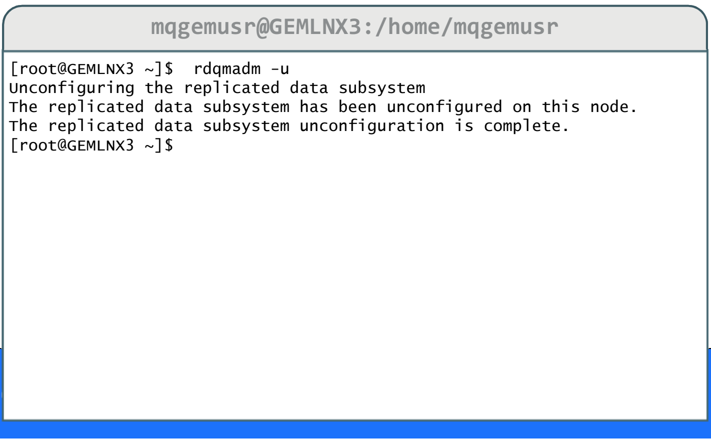
mqgemusr@GEMLNX1:/home/mqgemusr

```
[root@GEMLNX1 ~]$ rdqmadm -u
Unconfiguring the replicated data subsystem
The replicated data subsystem has been unconfigured on this node.
The replicated data subsystem should be unconfigured on all replicated
data nodes.
You need to run command 'rdqmadm -u' on nodes 'GEMLNX2' and 'GEMLNX3'.
[root@GEMLNX1 ~]$
```



mqgemusr@GEMLNX2:/home/mqgemusr

```
[root@GEMLNX2 ~]$ rdqmadm -u
Unconfiguring the replicated data subsystem
The replicated data subsystem has been unconfigured on this node.
The replicated data subsystem should be unconfigured on all replicated
data nodes.
You need to run command 'rdqmadm -u' on node 'GEMLNX3'.
[root@GEMLNX2 ~]$
```



mqgemusr@GEMLNX3:/home/mqgemusr

```
[root@GEMLNX3 ~]$  rdqmadm -u
Unconfiguring the replicated data subsystem
The replicated data subsystem has been unconfigured on this node.
The replicated data subsystem unconfiguration is complete.
[root@GEMLNX3 ~]$
```

Removing replicated data subsystem - Notes

N O T E S

- Finally, once you have deleted the queue manager on all nodes, you can remove the replicated data subsystem by using the `rdqmadm` command on each node. Of course, you'll only do this if you have removed the last of your RDQM HA queue managers.

IBM MQ RDQM HA - Summary

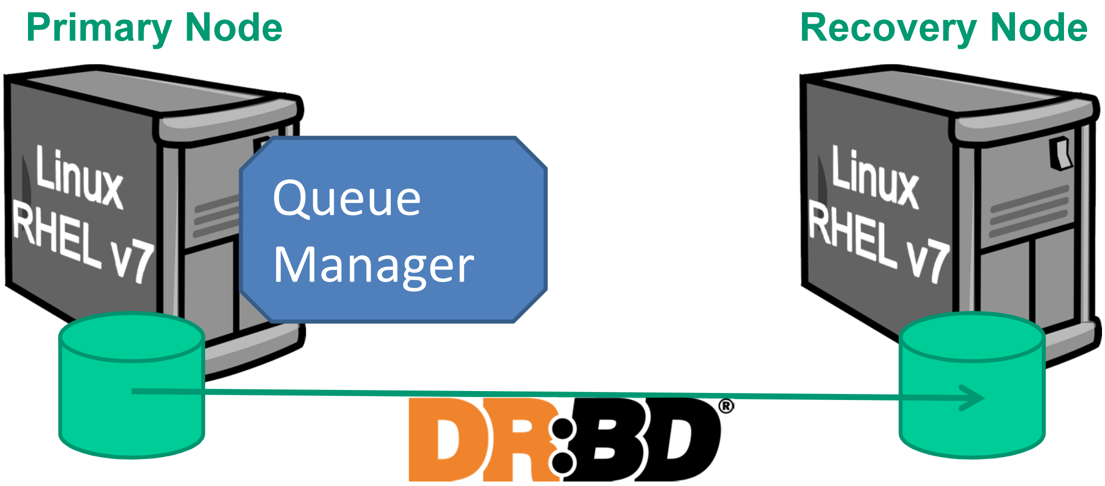
- Introduced in IBM MQ V9.0.4
- Linux only – in fact RHEL V7 required
- Queue manager data stored on local storage
  - No requirement for shared discs
- Data replicated automatically and synchronously to other machines in HA Group
- High Availability without shared disks or “external” HA management solutions

IBM MQ RDQM HA – Summary - Notes

N O T E S

- The Replicated Data Queue Manager (RDQM) High Availability feature was introduced in IBM MQ V9.0.4 CD release, for Linux platforms only. Queue managers that use RDQM have messages and logs written to the local storage, and this storage is automatically, synchronously replicated by the IBM MQ system to two other machines. There is no requirement for shared discs. However, it is possible to use RDQM HA with a dedicated remote volume per node, and such an environment is very likely in cloud deployments.
- This provides High Availability without the need for shared disks or external high-availability management solutions.

RDQM Disaster Recovery



RDQM Disaster Recovery - Notes

N O T E S

- RDQM (replicated data queue manager) DR is a disaster recovery solution that is available on Linux RHEL v7 platforms.
- An RDQM DR configuration consists of two servers, one running the primary instance of the server, and the second instance acting as a recovery node. Data is replicated between the queue manager instances using Distributed Replicated Block Device (DRBD). This means that if you lose your primary queue manager, you can manually make the secondary instance into the primary instance and start the queue manager, then resume work from the same place. You cannot start a queue manager while it is in the secondary role.
- An RDQM DR Queue Manager uses some of the same infrastructure as an RDQM HA Queue Manager, specifically DRBD for replication. However, an RDQM DR Queue Manager cannot also be an RDQM HA Queue Manager.

Preparation for RDQM DR

1. Installed the pre-requiste packages (DRBD) and the MQSeriesRDQM package.

2. Ensure this node has a valid internet host name
   - The value that is returned by `uname -n`
   - Make a note of the name for each of your two nodes now.

3. Created a volume group named `drbdpool`.
   - For RDQM DR you must allow twice as much disc space for your queue manager as for RDQM HA due to the 100% overhead of the snapshot
   - Make sure you know the usuable size of your volume now. Issue the `pvdisplay` command if you don't know.

4. Configured the firewall for RDQM using the provided `$MQ_INSTALLATION_PATH/samp/rdqm/firewalld/configure.sh` script.
   - This opened up a range of ports, 7000-7100 for data replication.

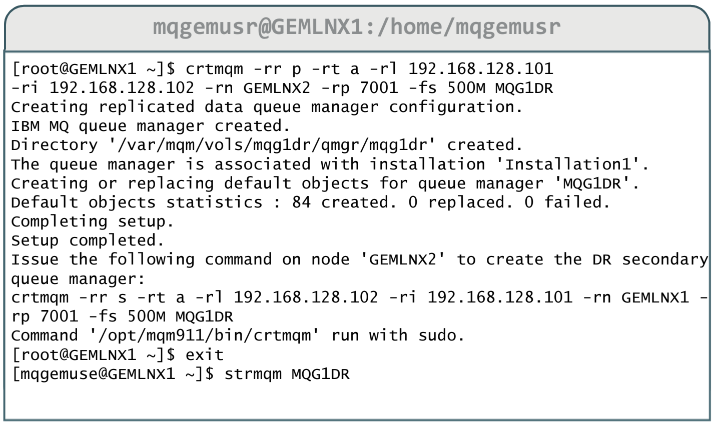
5. Configured SELinux for RDQM requirements.

Preparation for RDQM DR - Notes

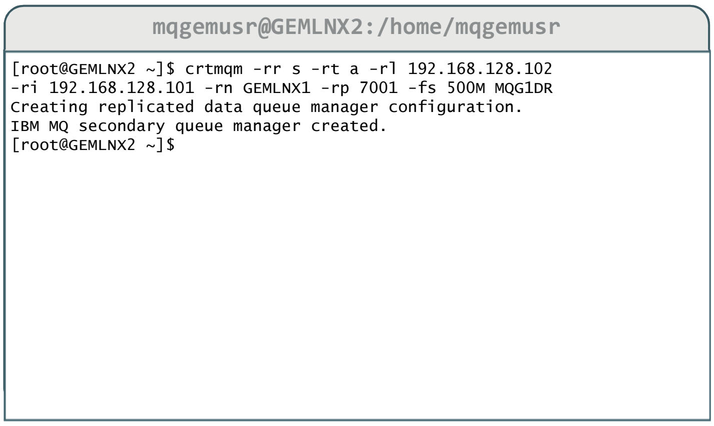
NOTES

- The preparation for creating an RDQM DR queue manager is the same as (or perhaps actually a subset of) what you have already done for the RDQM HA queue manager. As reminder, this is a list of the preparation you did earlier on each node that is relevant to RDQM DR.
- You are only going to need two nodes for an RQDM DR queue manager.

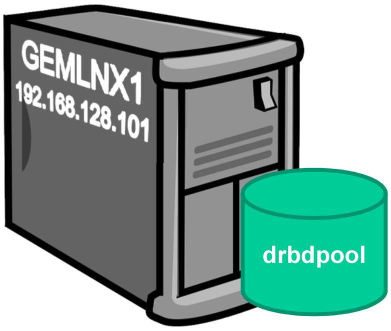
Creating an RDQM DR Queue Manager - crtmqm

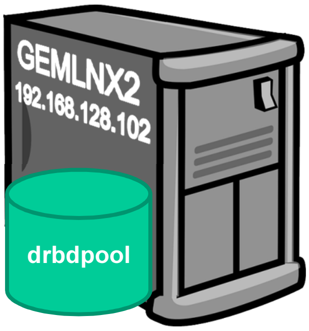


```
[root@GEMLNX1 ~]$ crtmqm -rr p -rt a -rl 192.168.128.101
-ri 192.168.128.102 -rn GEMLNX2 -rp 7001 -fs 500M MQG1DR
Creating replicated data queue manager configuration.
IBM MQ queue manager created.
Directory '/var/mqm/vols/mqg1dr/qmgr/mqg1dr' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'MQG1DR'.
Default objects statistics : 84 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
Issue the following command on node 'GEMLNX2' to create the DR secondary
queue manager:
crtmqm -rr s -rt a -rl 192.168.128.102 -ri 192.168.128.101 -rn GEMLNX1 -
rp 7001 -fs 500M MQG1DR
Command '/opt/mqm911/bin/crtmqm' run with sudo.
[root@GEMLNX1 ~]$ exit
[mqgemuse@GEMLNX1 ~]$ strmqm MQG1DR
```



```
[root@GEMLNX2 ~]$ crtmqm -rr s -rt a -rl 192.168.128.102
-ri 192.168.128.101 -rn GEMLNX1 -rp 7001 -fs 500M MQG1DR
Creating replicated data queue manager configuration.
IBM MQ secondary queue manager created.
[root@GEMLNX2 ~]$
```





Creating an RDQM DR Queue Manager - crtmqm - Notes

N

O

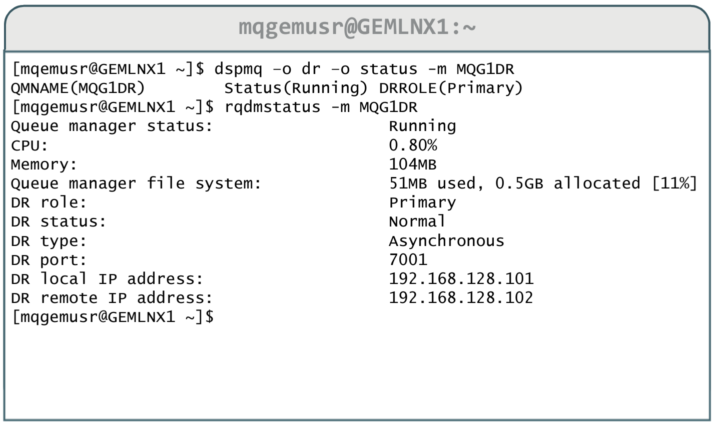
T

E

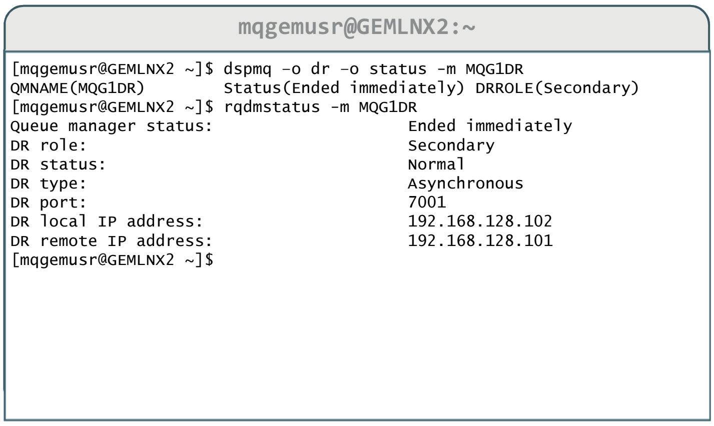
S

- On the machine that is going to be the primary node for this RDQM DR queue manager, issue the command shown to create an RDQM DR Queue Manager using Asynchronous Replication (`-rt a`).
- The file system size you provide (`-fs`) is half the size of your usuable space. This is because IBM MQ creates two logical volumes, the second of which is to support the "reverting to snapshot" operation.
- You will have to issue these commands as root.
- As you can see, the output from this command gives you the exact command to run on the secondary machine. Handy!

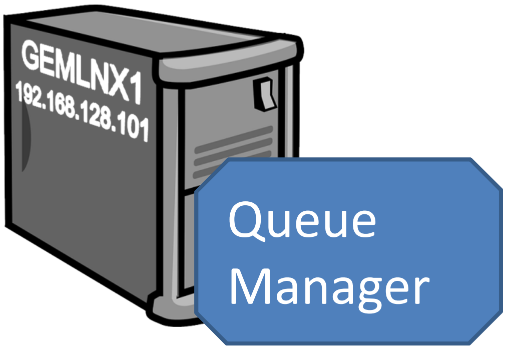
RDQM DR Status


mqgemusr@GEMLNX1:~

```
[mqemusr@GEMLNX1 ~]$ dspmq -o dr -o status -m MQG1DR
QMNAME(MQG1DR)         Status(Running) DRROLE(Primary)
[mqgemusr@GEMLNX1 ~]$ rqdmstatus -m MQG1DR
Queue manager status:                   Running
CPU:                                    0.80%
Memory:                                 104MB
Queue manager file system:              51MB used, 0.5GB allocated [11%]
DR role:                                Primary
DR status:                              Normal
DR type:                                Asynchronous
DR port:                                7001
DR local IP address:                    192.168.128.101
DR remote IP address:                   192.168.128.102
[mqgemusr@GEMLNX1 ~]$
```


mqgemusr@GEMLNX2:~

```
[mqgemusr@GEMLNX2 ~]$ dspmq -o dr -o status -m MQG1DR
QMNAME(MQG1DR)         Status(Ended immediately) DRROLE(Secondary)
[mqgemusr@GEMLNX2 ~]$ rqdmstatus -m MQG1DR
Queue manager status:                      Ended immediately
DR role:                                   Secondary
DR status:                                 Normal
DR type:                                   Asynchronous
DR port:                                   7001
DR local IP address:                       192.168.128.102
DR remote IP address:                      192.168.128.101
[mqgemusr@GEMLNX2 ~]$
```



RDQM DR Status - Notes

N O T E S

- As we saw with RDQM HA, the `dspmq` command has been updated to show the DR role of each RDQM DR instance.
- Note that it is normal for the secondary node of an RDQM DR queue manager to show `Ended immediately`, rather than `Running elsewhere` as you may have seen with High Availability features.
- And also, as we saw with RDQM HA, the new `rdqmstatus` command shows some information about the RDQM DR queue manager. On the running instance you get to see some information about the CPU, memory and file system usage.

Connectivity to an RDQM DR QMgr

- No floating IP address
- Use comma-separated connection name on clients if you need to

```
export MQSERVER='MQGEM.SVRCONN/TCP/192.168.128.101(1901),192.168.128.102(1901)'
```

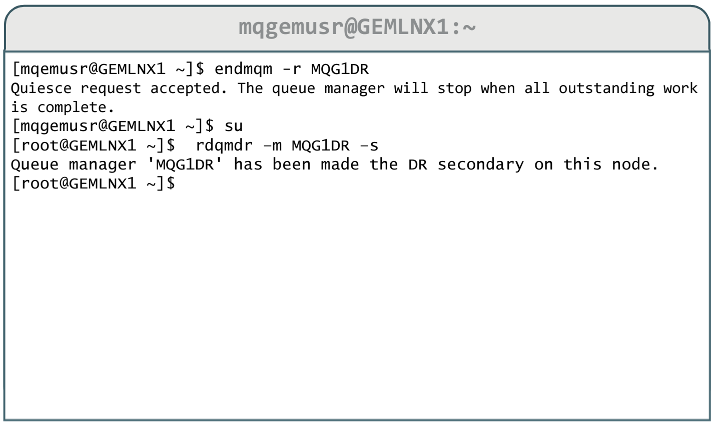
- Your Disaster Recovery planning may already have DNS updates to switch hostnames to use DR site IP addresses.

Connectivity to an RDQM DR QMgr - Notes

N
O
T
E
S

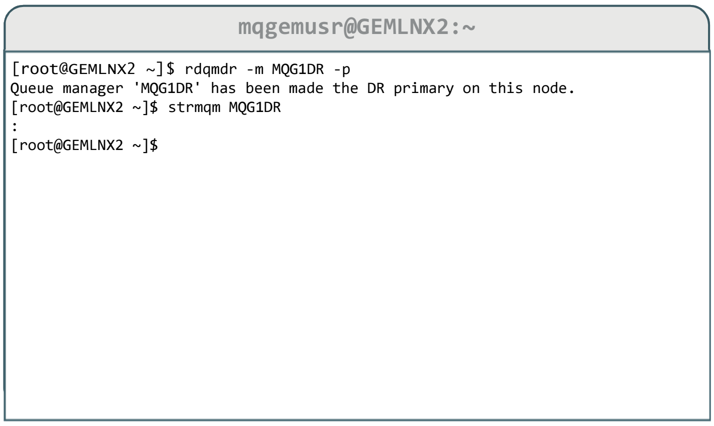
- There’s no floating IP address facility on an RDQM DR Queue manager, so to connect to the correct instance, you could configure your client applications to use a comma-separated connection name.
- Clearly in the case of a disaster where you are moving across to the DR site, you may well have other facilities in place to associate new IP addresses with hostnames and so forth.

Testing your DR setup



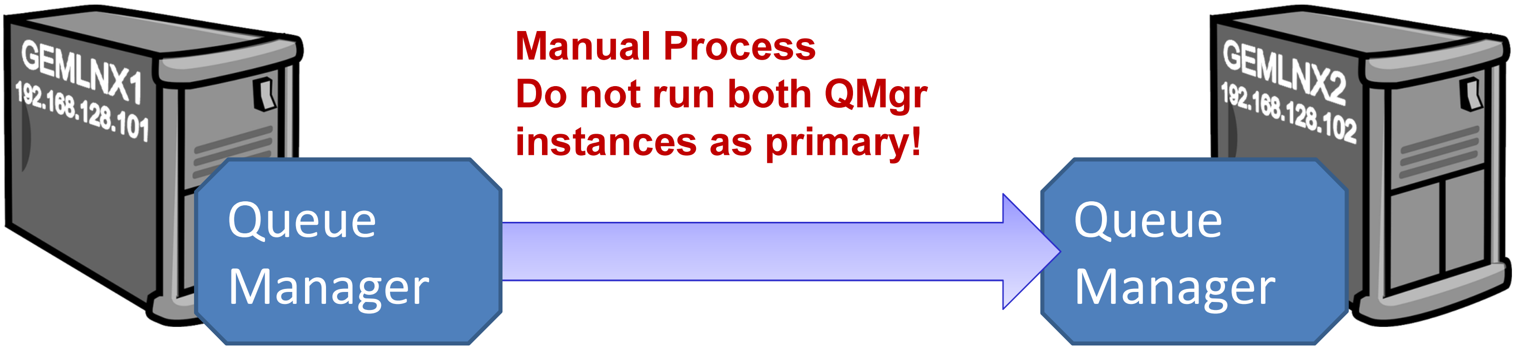
mqgemusr@GEMLNX1:~

```
[mqgemusr@GEMLNX1 ~]$ endmqm -r MQG1DR
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
[mqgemusr@GEMLNX1 ~]$ su
[root@GEMLNX1 ~]$  rdqmdr -m MQG1DR -s
Queue manager 'MQG1DR' has been made the DR secondary on this node.
[root@GEMLNX1 ~]$
```



mqgemusr@GEMLNX2:~

```
[root@GEMLNX2 ~]$ rdqmdr -m MQG1DR -p
Queue manager 'MQG1DR' has been made the DR primary on this node.
[root@GEMLNX2 ~]$ strmqm MQG1DR
:
[root@GEMLNX2 ~]$
```



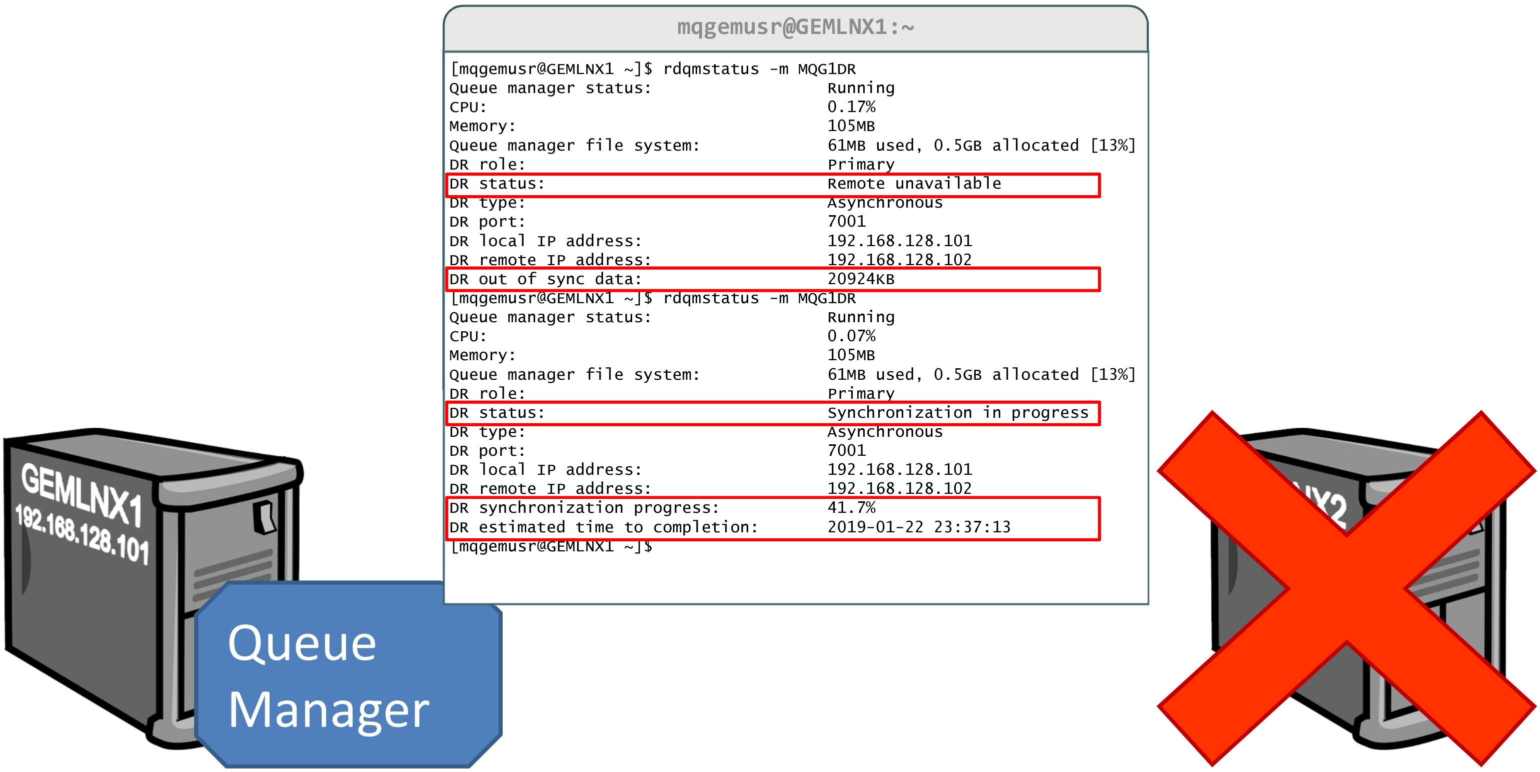
Manual Process
Do not run both QMgr instances as primary!

Testing your DR setup - Notes

N O T E S

- If you wanted to test out your Disaster Recovery setup, to make sure your applications are correctly configured to use the DR site for example, you would follow this procedure.
- End the queue manager on the primary node using the (`-r`) flag to tell any connected clients to reconnect (since otherwise they will get a quiescing return code and end gracefully).
- Now change the role of the queue manager on the primary node to be secondary (running the command as root).
- Now on the secondary node, change its role to primary (again as root).
- And finally start the queue manager up.
- Note: RDQM DR is a manual process. It is up to you to ensure that the two instances of the queue manager do not both believe that they are primary.

Synchronization



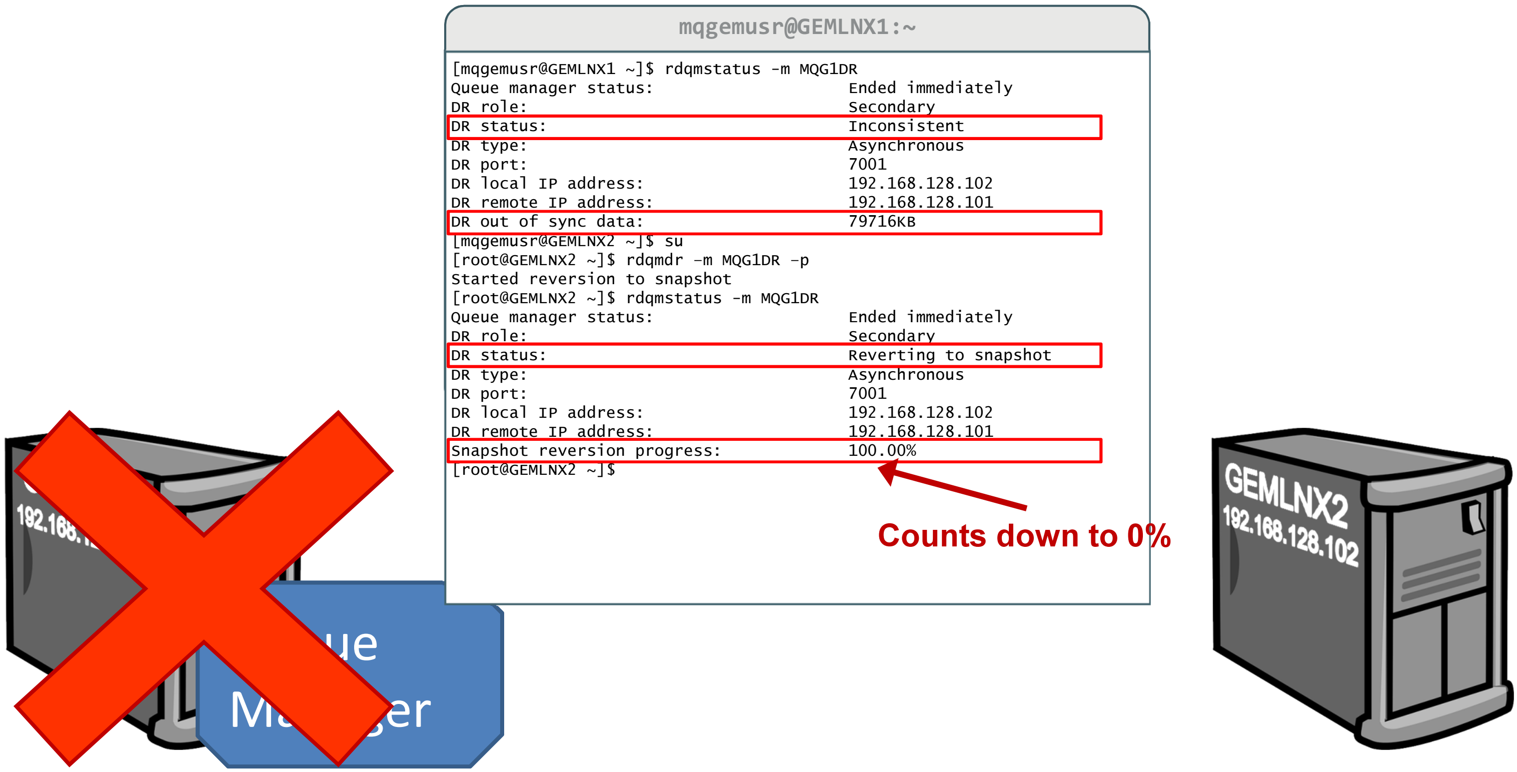
```
[mqgemusr@GEMLNX1 ~]$ rdqmstatus -m MQG1DR
Queue manager status:                 Running
CPU:                                  0.17%
Memory:                               105MB
Queue manager file system:            61MB used, 0.5GB allocated [13%]
DR role:                              Primary
DR status:                            Remote unavailable
DR type:                              Asynchronous
DR port:                              7001
DR local IP address:                  192.168.128.101
DR remote IP address:                 192.168.128.102
DR out of sync data:                  20924KB
[mqgemusr@GEMLNX1 ~]$ rdqmstatus -m MQG1DR
Queue manager status:                 Running
CPU:                                  0.07%
Memory:                               105MB
Queue manager file system:            61MB used, 0.5GB allocated [13%]
DR role:                              Primary
DR status:                            Synchronization in progress
DR type:                              Asynchronous
DR port:                              7001
DR local IP address:                  192.168.128.101
DR remote IP address:                 192.168.128.102
DR synchronization progress:          41.7%
DR estimated time to completion:      2019-01-22 23:37:13
[mqgemusr@GEMLNX1 ~]$
```

Synchronization - Notes

N O T E S

- Most of the time, both nodes of the queue manager, the primary node where it is running, and the recovery node, are available, connected to each other, and the data from the primary is being replicated to the recovery node.
- If however, connectivity between the two nodes is lost, either due to a network issue or because the recovery node is offline, the primary node keeps track of the changes to the persistent data, and once the recovery node comes back online, it is brought back to the current position by a synchronization process.

Reverting to snapshot



```
[mqgemusr@GEMLNX1 ~]$ rdqmstatus -m MQG1DR
Queue manager status:                   Ended immediately
DR role:                                Secondary
DR status:                              Inconsistent
DR type:                                Asynchronous
DR port:                                7001
DR local IP address:                    192.168.128.102
DR remote IP address:                   192.168.128.101
DR out of sync data:                    79716KB
[mqgemusr@GEMLNX2 ~]$ su
[root@GEMLNX2 ~]$ rdqmdr -m MQG1DR -p
Started reversion to snapshot
[root@GEMLNX2 ~]$ rdqmstatus -m MQG1DR
Queue manager status:                   Ended immediately
DR role:                                Secondary
DR status:                              Reverting to snapshot
DR type:                                Asynchronous
DR port:                                7001
DR local IP address:                    192.168.128.102
DR remote IP address:                   192.168.128.101
Snapshot reversion progress:            100.00%
[root@GEMLNX2 ~]$
```

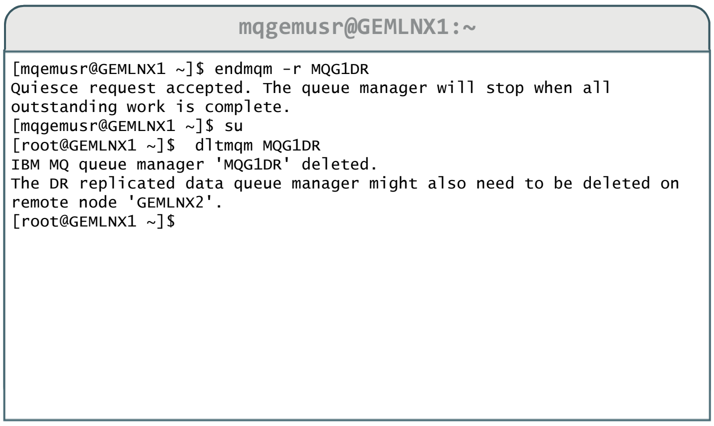
Counts down to 0%

Reverting to snapshot - Notes

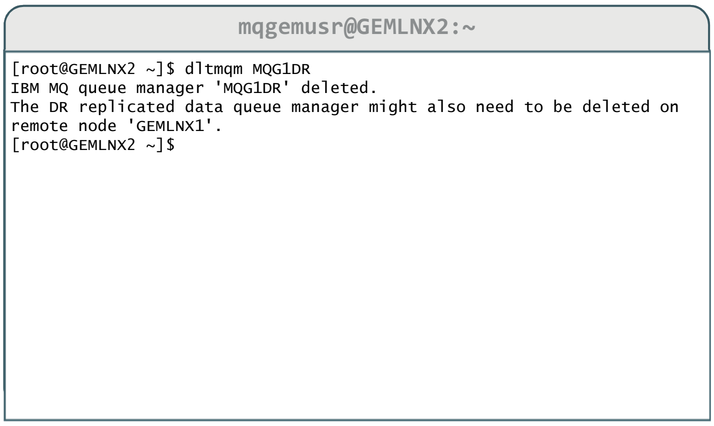
N O T E S

- You've just learned about the synchronization process. Just before that process begins, the recovery node takes a snapshot of its current view of the data - that's why when you created the queue manager you needed twice the storage space. This snapshot means that there is a consistent version of the queue manager's persistent data on the recovery node while synchronization takes place. This is needed because during synchronization, the data on the recovery node is in an inconsistent state.
- If the primary node should fail during that process, there has to be a consistent set of data that you can fall back to on the recovery node.
- If it does fail, you need to get the recovery node up and running as the queue manager. It will be a snapshot in time that is somewhat earlier than the last point the primary was at because we had two outages, first of all the recovery node had an outage, and then the primary node had a disaster before the two nodes were able to be re-synchronized again. But better back-level than no queue manager at all, in times of disaster.
- On the secondary node, issue the `rdqmdr` command to change the secondary node to be the primary, and as a result, cause it to revert to the snapshot.
- Issue the `rdqmstatus` command until it no longer shows a `DR Status` of `Reverting to snapshot` and the `Snapshot reversion progress percentage` has counted all the way down to 0%. That's when you know it is safe to start the queue manager, because it now has a consistent set of persistent data.
- **Note:** I have never seen the percentage reported as anything other than 100% and then done.

Removing an RDQM DR Queue Manager



```
[mqemusr@GEMLNX1 ~]$ endmqm -r MQG1DR
Quiesce request accepted. The queue manager will stop when all
outstanding work is complete.
[mqgemusr@GEMLNX1 ~]$ su
[root@GEMLNX1 ~]$  dltmqm MQG1DR
IBM MQ queue manager 'MQG1DR' deleted.
The DR replicated data queue manager might also need to be deleted on
remote node 'GEMLNX2'.
[root@GEMLNX1 ~]$
```



```
[root@GEMLNX2 ~]$ dltmqm MQG1DR
IBM MQ queue manager 'MQG1DR' deleted.
The DR replicated data queue manager might also need to be deleted on
remote node 'GEMLNX1'.
[root@GEMLNX2 ~]$
```

GEMLNX1
192.168.128.101
Queue Manager
GEMLNX2
192.168.128.102

Removing an RDQM DR Queue Manager - Notes

N O T E S

- Removing an RDQM DR queue manager is a much simpler process than removing an RDQM HA queue manager – mainly because there is no Pacemaker cluster to tear down with the `rdqmadm` command.
- You simply have to remove the replicated data process from both nodes, and this is part of the MQ `dltmqm` command.

IBM MQ RDQM DR - Summary

- Introduced in IBM MQ V9.0.5
- Linux only – in fact RHEL V7 required
- Queue manager data stored on local storage
  - No requirement for shared discs
- Data replicated to recovery node using DRBD
- Queue manager can be switched to run on the recovery node against the replicated data should a disaster befall it

IBM MQ RDQM DR – Summary - Notes

N O T E S

- The Replicated Data Queue Manager (RDQM) Disaster Recovery feature was introduced in IBM MQ V9.0.5 CD release, for Linux RHEL v7 platforms only. This feature allows you to create a primary instance of a queue manager, which replicates its persistent data to a recovery node using DRBD. If a disaster were to befall the primary node, the queue manager could be restarted using the copy of the persistent data that is on the recovery node.