

# Replicated Data Queue Managers (RDQM) - DR for HA Group

Messaging Administrator

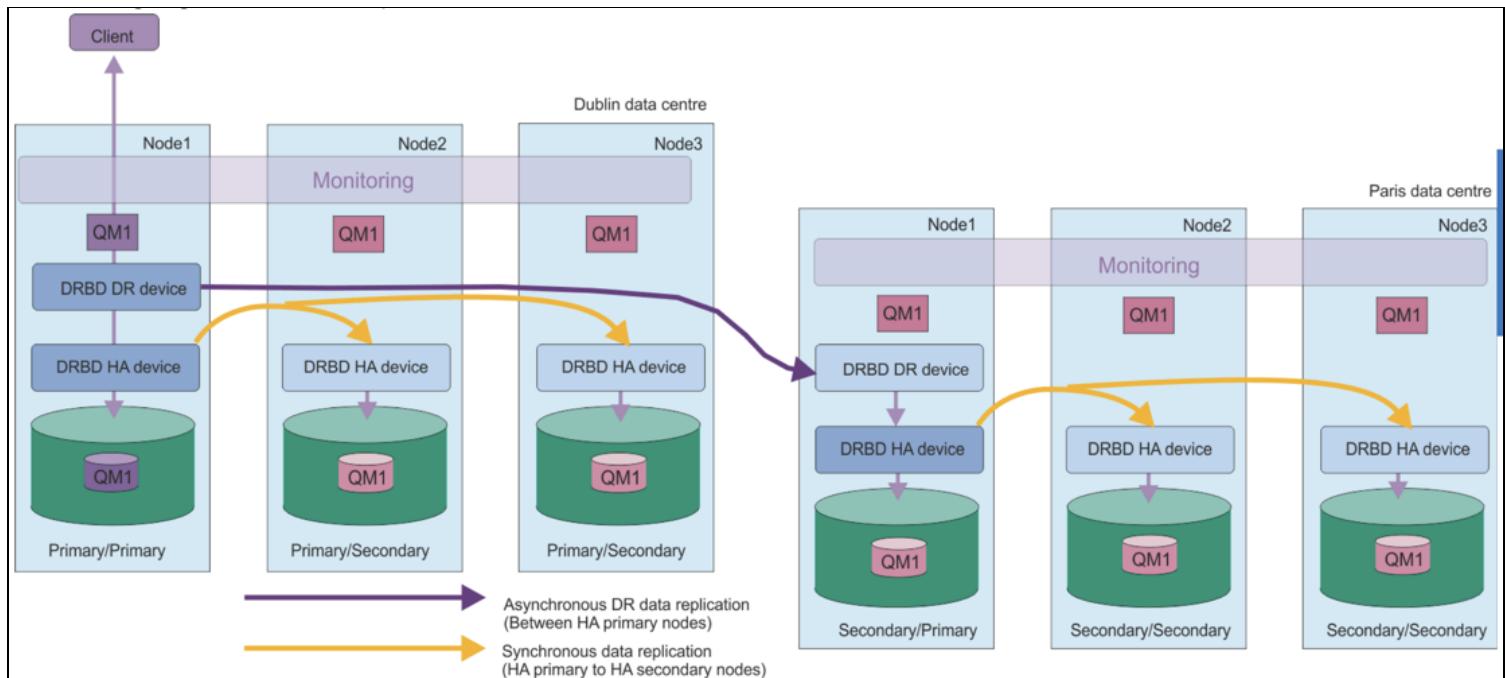
**Summary:** Replicated Data Queue Managers (RDQM) for DR of HA Group

## Introducing IBM MQ Replicated Data Queue Manager - DR for HA Group

### DR for HA Overview

You can configure a replicated data queue manager (RDQM) that runs on a high availability group on one site, but can fail over to another high availability group at another site if some disaster occurs that makes the first group unavailable. This is known as a DR/HA RDQM.

A DR/HA RDQM combines the features of a high availability RDQM (Lab 2) and a disaster recovery RDQM (Lab 3). The following diagram shows an example DR/HA RDQM.



The replication between the DR/HA RDQMs on the main site and the disaster recovery site is always asynchronous. With asynchronous replication, operations such as MQ PUT or GET complete and return to the application before the event is replicated to the secondary queue manager.

You can have two active sites rather than ‘main’ and ‘recovery’ sites, if required, so some of your DR/HA RDQMs run on one site and some on the other during normal operation. If a disaster occurs and one site becomes unavailable, then all DR/HA RDQMs run on the same HA group at the same site. You will demonstrate this in the lab.

Each HA group is configured in the same way as an ordinary HA group. You can define floating IP addresses for a DR/HA RDQM in each HA group. The floating IP address can be the same or different for each HA group.

You cannot upgrade an existing RDQM to be a DR/HA RDQM, you must create a DR/HA RDQM. (If required, you could back up the data of an existing RDQM, delete it, recreate it as a DR/HA RDQM, and then restore the data.)

To configure DR/HA RDQMs, you must complete the following major steps:

1. Configure an HA group on the ‘main’ site.
2. Configure an HA group on the ‘recovery’ site.
3. Create a primary/primary DR/HA RDQM on one node of the HA group in the ‘main’ site.
4. Create primary/secondary DR/HA RDQMs on the other two nodes in the ‘main’ site.
5. Define a floating IP address for an application to access the DR/HA RDQM when it is running on any of the nodes of the HA group on the ‘main’ site.
6. Create a secondary/primary DR/HA RDQM on one node of the HA group on the ‘recovery’ site.

7. Create secondary/secondary DR/HA RDQMs on the other two nodes in the ‘recovery’ site.
8. Define a floating IP address for an application to access the DR/HA RDQM when it is running on any of the nodes of the HA group on the ‘recovery’ site.

This lab is concerned with demonstrating the failover of an HA group to an HA group running in a DR site. This requires that the DR site has three equivalent nodes running in the DR site.

## Lab Introduction

- Six RHEL 7.7 x86\_64 systems running in Skytap:
  - rdqm1 - This will be our primary node.
  - rdqm2 - This will be a secondary node.
  - rdqm3 - This will be another secondary node.
  - dr1 - DR Fail over primary node.
  - dr2 - DR Fail over secondary node.
  - dr3 - DR Fail over other secondary node.
- VMWare Workstation virtual networks primary HA group:

Name	Type	SkyTap Network	Subnet	DHCP
ens34	Administration	ens34	10.0.0.0	no
ens35	HA Replication	ens35	10.0.1.0	no
ens36	DR Replication	ens36	10.0.2.0	no
ens37	HA primary	ens37	10.0.3.0	no
ens38	HA secondary	ens38	10.0.4.0	no

- Network interfaces for HA group:

Interface Purpose	Interface Name	rdqm1 (Primary node)	rdqm2 (Secondary node)	rdqm3 (Secondary node)
MQ Fixed IP	ens35	10.0.1.1	10.0.1.2	10.0.1.3
MQ Floating IP		10.0.1.20	10.0.1.20	10.0.1.20

HA interfaces are used as follows:

- HA Primary - to monitor the nodes in the cluster
- HA Alternate - backup for monitoring the cluster if the HA Primary network fails
- HA Replication - for synchronous data replication (the higher the bandwidth the better and the lower the latency the better)

- VMWare Workstation virtual networks secondary DR HA group:

Name	Type	SkyTap Network	Subnet	DHCP
ens34	Administration	ens34	10.0.0.0	no
ens35	HA Replication	ens35	10.0.1.0	no
ens36	DR Replication	ens36	10.0.2.0	no
ens37	HA primary	ens37	10.0.3.0	no
ens38	HA secondary	ens38	10.0.4.0	no

- Network interfaces for DR HA group:

Interface Purpose	Interface Name	dr1 (Primary node)	dr2 (Secondary node)	dr3 (Secondary node)
MQ Fixed IP	ens35	10.0.1.14	10.0.1.15	10.0.1.16
MQ Floating IP		10.0.1.20	10.0.1.20	10.0.1.20

HA interfaces are used as follows:

- HA Primary - to monitor the nodes in the cluster
- HA Alternate - backup for monitoring the cluster if the HA Primary network fails
- HA Replication - for synchronous data replication (the higher the bandwidth the better and the lower the latency the better)
- Dedicated volume group “drbdpool” containing a single physical volume on each node for RDQM, but please note, you will not see any further reference to this in this document.
- The following groups configured:
  - **mqm** to allow user to run specific MQ commands
  - **haclient** to allow user to run HA-specific commands

- A normal user “ibmuser” has been defined for running applications and MQ commands.

Name	Password	Purpose	Group
root	IBMDem0s!	superuser	
ibmuser	engageibm	host vm user - mq user	mqm

- Firewall (firewalld) enabled, and ports 1500 & 1501 will be defined during the lab.
- The following Pacemaker dependencies have already been installed. This list should be sufficient for a standard installation of RHEL 7.7 Server or Workstation. For your own environment setup, or if you are using some other installation, additional packages may be needed:
  - cifs-utils
  - gnutls
  - libcgroup
  - libtool-ltdl
  - lvm2
  - net-snmp-libs
  - nfs-utils
  - perl-TimeDate
  - psmisc
  - PyYAML

Depending on your security configuration, there are three different ways to configure the RDQM feature:

1. The simplest way is if the mqm user can ssh between the three nodes of the cluster without a password and can sudo to run the necessary commands.
2. The intermediate option is if the mqm user can sudo but not ssh. It is preferable if the actual users are also in the haclient group.
3. The default is that the mqm user cannot ssh or sudo.

In this lab, instructions are provided to setup and test using the intermediate method.

## Setup the RHEL image (pre-configured on SkyTap):

In the Skytap environment there are 6 virtual machines: rdqm1, rdqm2, rdqm3, dr1, dr2, and dr3 which should be in a powered off or paused state.

**RDQM - HADR - 20200618 - MQ 9.1.5 - Installed - DO-NOT-DELETE**

Last run: 21 minutes ago (created: 28 minutes ago)  
 RDQM - MC9.1.4 MQ / RDQM Installed. RDQM not configured. SSH. Added new disk for drbdpool. Reset hostnames - network adapters. drbdpool configured. Opsys limits set. firewall configured. Added 2 network adapters for internet access.  
 Downloaded MQ 9.1.5. 20200402 - Removed MQ 9.1.4. Downloaded MQ 9.1.5. Ready for MQ / RDQM installation and configuration. MQ / RDQM 9.1.5 installed on all VMs except mqmp where only MQ installed (students will install RDQM support before starting). 20200611 - add default rdqm.ini to /home/ibmuser.

**Settings:**  **Tags:**  DTE\_Base  RDQM  RHEL

<input type="checkbox"/> Region/Owner US-Central Jack Carnes	<input type="checkbox"/> VMs (6): <a href="#">Settings</a> Metered RAM: 48 GB	<input type="checkbox"/> Networking: <a href="#">Settings</a> Networks: 5 <input type="checkbox"/> Pub. services: 0 <input type="checkbox"/> WANs: 0 <input type="checkbox"/> Public IPs: 0	<input type="checkbox"/> Automation VM sequencing: On <input type="checkbox"/> Power options: On	<input type="checkbox"/> Collaboration Projects: 0 <input type="checkbox"/> Schedules: 0 <input type="checkbox"/> Metadata
<a href="#">VMs (6)</a> <a href="#">Containers (0)</a> <a href="#">Sharing Portals</a> <a href="#">Network Topology</a> <a href="#">Labels (4)</a> <a href="#">Activity</a>				

**VMs (6)** [Delete \(6\)](#) [Add VMs](#)

Sort by name

| Powered off  |
|--|--|--|--|--|--|
| <input checked="" type="checkbox"/> <a href="#">dr3 - Red Hat Enterprise Lin ux 7.7 64-bit</a><br>Endpoints: 5   | <input checked="" type="checkbox"/> <a href="#">drp - Red Hat Enterprise Lin ux 7.7 64-bit</a><br>Endpoints: 5   | <input checked="" type="checkbox"/> <a href="#">drs - Red Hat Enterprise Lin ux 7.7 64-bit</a><br>Endpoints: 5   | <input checked="" type="checkbox"/> <a href="#">rdqm1 - Red Hat Enterprise Linux 7.7 64-bit</a><br>Endpoints: 5  | <input checked="" type="checkbox"/> <a href="#">rdqm2 - Red Hat Enterprise Linux 7.7 64-bit</a><br>Endpoints: 5  | <input checked="" type="checkbox"/> <a href="#">rdqm3 - Red Hat Enterprise Linux 7.7 64-bit</a><br>Endpoints: 5  |
| METERED RAM<br>8 GB <input type="button" value="▼"/><br><a href="#">A1</a> <a href="#">Q0</a> <a href="#">D0</a> <a href="#">S0</a> <a href="#">G0</a> <a href="#">X</a> | METERED RAM<br>8 GB <input type="button" value="▼"/><br><a href="#">A1</a> <a href="#">Q0</a> <a href="#">D0</a> <a href="#">S0</a> <a href="#">G0</a> <a href="#">X</a> | METERED RAM<br>8 GB <input type="button" value="▼"/><br><a href="#">A1</a> <a href="#">Q0</a> <a href="#">D0</a> <a href="#">S0</a> <a href="#">G0</a> <a href="#">X</a> | METERED RAM<br>8 GB <input type="button" value="▼"/><br><a href="#">A1</a> <a href="#">Q0</a> <a href="#">D0</a> <a href="#">S0</a> <a href="#">G0</a> <a href="#">X</a> | METERED RAM<br>8 GB <input type="button" value="▼"/><br><a href="#">A1</a> <a href="#">Q0</a> <a href="#">D0</a> <a href="#">S0</a> <a href="#">G0</a> <a href="#">X</a> | METERED RAM<br>8 GB <input type="button" value="▼"/><br><a href="#">A1</a> <a href="#">Q0</a> <a href="#">D0</a> <a href="#">S0</a> <a href="#">G0</a> <a href="#">X</a> |

1. Click the run button to start or resume the VMs. Wait for the VM icons to become active, when they turn green.
2. Click the monitor icon for *rdqm1* which will launch the desktop in another browser tab.

Running	Running	Running
<input checked="" type="checkbox"/> <a href="#">rdqm1 - Red Hat Enterprise Linux 7.7 64-bit</a> Endpoints: 5	<input checked="" type="checkbox"/> <a href="#">rdqm2 - Red Hat Enterprise Linux 7.7 64-bit</a> Endpoints: 5	<input checked="" type="checkbox"/> <a href="#">rdqm3 - Red Hat Enterprise Linux 7.7 64-bit</a> Endpoints: 5
METERED RAM 8 GB <input type="button" value="▼"/> <a href="#">A1</a> <a href="#">Q0</a> <a href="#">D0</a> <a href="#">S0</a> <a href="#">G0</a> <a href="#">X</a>	METERED RAM 8 GB <input type="button" value="▼"/> <a href="#">A1</a> <a href="#">Q0</a> <a href="#">D0</a> <a href="#">S0</a> <a href="#">G0</a> <a href="#">X</a>	METERED RAM 8 GB <input type="button" value="▼"/> <a href="#">A1</a> <a href="#">Q0</a> <a href="#">D0</a> <a href="#">S0</a> <a href="#">G0</a> <a href="#">X</a>

3. Log on to VM *rdqm1* as user **ibmuser**, using password **engageibm**.



## Pre-configuration steps

The following steps are necessary for configuring RDQM. They have already been completed on the VMs.

- Extract and Install MQ 9.1.5

The code is provided as a compressed tar file in the directory /home/ibmuser/.

- Install the MQ and RDQM code

RDQM is a single feature which now supports HA and/or DR. The RDQM support requires the Server and Runtime packages.

Run the installation script.

- Configure the RedHat firewall

If there is a firewall between the nodes in the HA group, then the firewall must allow traffic between the nodes on a range of ports. Open another terminal, switch to user root, and run the sample file.

- Configure the OS storage settings

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

```
...
semanage permissive -a drbd_t
...
```

- Configure groups

To create, delete, or configure replicated data queue managers (RDQMs), you must use a user ID that belongs to both the mqm and haclient groups.

If you want to allow a normal user in the mqm group to create RDQM instances etc., you need to grant the user access to certain commands via sudo. You will add the mqm user to the root and haclient group. Then add root and ibmuser to the mqm and haclient groups.

- Create the Logical Group for the QM data

Each node requires a volume group named drbdpool. The storage for each replicated data queue manager is allocated as a separate logical volume per queue manager from this volume group. For the best performance, this volume group should be made up of one or more physical volumes that correspond to internal disk drives (preferably SSDs).

The above steps must be completed on each node before RDQM can be configured. At this point, you are ready to begin RDQM configuration.

## Configure RDQM

This lab assumes that HA and DR have not been configured. If you have attempted or completed either of the previous labs (12 - 13) using this Skytap environment, you must ensure that all RDQM queue managers have been deleted and the Pacemaker cluster removed using the Cleanup instructions at the end of Lab 12 (HA) and Lab 13 (DR). If you did not complete the Cleanup instructions in the previous labs, do so now before continuing with this lab.

If you have not attempted or completed the previous labs using this Skytap environment, you should skip ahead to “Install RDQM support”.

### Install RDQM support

As previously stated, MQ 9.1.5 has already been installed on all VMs. The advanced feature RDQM support has also been installed on all the VMs *except* on **rdqm1**. You will need to install RDQM support on rdqm1 so you can see how easy it is to install. You will review the requirements for RDQM and scripts for installation and configuration.

1. On **rdqm1** open a new terminal window and change to the directory for RDQM support. The installation tar file was extracted to */home/ibmuser/mq915*.

Change to the */home/ibmuser/mq915/MQServer/Advanced/RDQM/* with the following command:

```
cd ~/mq915/MQServer/Advanced/RDQM
```

2. List the members of directory *PreReqs* to see the important prerequisites of RDQM.

```
ls PreReqs
```

```
ibmuser@rdqm1:~/mq915/MQServer/Advanced/RDQM
File Edit View Search Terminal Help
[ibmuser@rdqm1 ~]$ cd ~/mq915/MQServer/Advanced/RDQM/
[ibmuser@rdqm1 RDQM]$ ls PreReqs/
drbd-9.0 pacemaker-1.1.20
```

Here you see Pacemaker and DRBD. If you drill into those subdirectories, you will see the RPM packages for installing these prerequisites.

```
ibmuser@rdqm1:~/mq915/MQServer/Advanced/RDQM
File Edit View Search Terminal Help
[ibmuser@rdqm1 RDQM]$ ls PreReqs/drbd-9.0/
drbd-bash-completion-9.12.0-1.el7.x86_64.rpm
drbd-pacemaker-9.12.0-1.el7.x86_64.rpm
drbd-udev-9.12.0-1.el7.x86_64.rpm
drbd-utils-9.12.0-1.el7.x86_64.rpm
kmod-drbd-9.0.22_3.10.0_1062-1.x86_64.rpm
kmod-drbd-9.0.22_3.10.0_514.36.5-1.x86_64.rpm
[ibmuser@rdqm1 RDQM]$ ls PreReqs/pacemaker-1.1.20/
cluster-glue-libs-1.0.12.linbit-2.0+20180815+be86a9f22546.el7.x86_64.rpm
corosync-2.4.4.linbit-1.0+20181123+918bc4781279.el7.x86_64.rpm
corosynclib-2.4.4.linbit-1.0+20181123+918bc4781279.el7.x86_64.rpm
crmsh-3.0.3.linbit-1.0+20181114+2309c21a33a9.el7.noarch.rpm
crmsh-scripts-3.0.3.linbit-1.0+20181114+2309c21a33a9.el7.noarch.rpm
libqb-1.0.3.linbit-3.0+20190115+c880fcfd8d67.el7.x86_64.rpm
pacemaker-1.1.20.linbit-1+20190404+eab6a2092b71.el7.2.x86_64.rpm
pacemaker-cli-1.1.20.linbit-1+20190404+eab6a2092b71.el7.2.x86_64.rpm
pacemaker-cluster-libs-1.1.20.linbit-1+20190404+eab6a2092b71.el7.2.x86_64.rpm
pacemaker-libs-1.1.20.linbit-1+20190404+eab6a2092b71.el7.2.x86_64.rpm
python-parallax-1.0.1.linbit-1.1.noarch.rpm
resource-agents-4.2.0.linbit-1+20190116+6a6f48180a59.el7.7.x86_64.rpm
[ibmuser@rdqm1 RDQM]$
```

3. Enter the command to edit the *installRDQMsupport* shell script.

gedit *installRDQMsupport*

Review the script noting:

- current directory RDQM
- defining packages for RPMs
  - DRBD-KMOD, DRBD, PACEMAKER
  - MQ\_DEPENDENCIES, RDQM\_PACKAGES, ADDITIONAL\_MQ\_PACKAGES
- yum install for packages

4. No changes are required, so close the editor by clicking the “X”.

5. Run the script to install RDQM with the following command:

```
sudo ./installRDOMsupport
```

```
ibmuser@rdqm1:~/mq915/MQServer/Advanced/RDQM
File Edit View Search Terminal Help
[ibmuser@rdqm1 RDQM]$ .edit installRDQMsupport
[ibmuser@rdqm1 RDQM]$ sudo ./installRDQMsupport
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Examining ./PreReqs/drbd-9.0/kmod-drbd-9.0.22_3.10.0_1062-1.x86_64.rpm: kmod-drbd-9.0.22_3.10.0_1062-1.x86_64
Marking ./PreReqs/drbd-9.0/kmod-drbd-9.0.22_3.10.0_1062-1.x86_64.rpm to be installed
Examining ./PreReqs/drbd-9.0/drbd-bash-completion-9.12.0-1.el7.x86_64.rpm: drbd-bash-completion-9.12.0-1.el7.x86_64
Marking ./PreReqs/drbd-9.0/drbd-bash-completion-9.12.0-1.el7.x86_64.rpm to be installed
Examining ./PreReqs/drbd-9.0/drbd-pacemaker-9.12.0-1.el7.x86_64.rpm: drbd-pacemaker-9.12.0-1.el7.x86_64
Marking ./PreReqs/drbd-9.0/drbd-pacemaker-9.12.0-1.el7.x86_64.rpm to be installed
Examining ./PreReqs/drbd-9.0/drbd-udev-9.12.0-1.el7.x86_64.rpm: drbd-udev-9.12.0-1.el7.x86_64
Marking ./PreReqs/drbd-9.0/drbd-udev-9.12.0-1.el7.x86_64.rpm to be installed
Examining ./PreReqs/drbd-9.0/drbd-utils-9.12.0-1.el7.x86_64.rpm: drbd-utils-9.12.0-1.el7.x86_64
Marking ./PreReqs/drbd-9.0/drbd-utils-9.12.0-1.el7.x86_64.rpm to be installed
Examining ./PreReqs/pacemaker-1.1.20/cluster-glue-libs-1.0.12.linbit-2.0+20180815+be86a9f22546.el7.x86_64.rpm: cluster-glue-libs-1.0.12.linbit-2.0+20180815+be86a9f22546.el7.x86_64
Marking ./PreReqs/pacemaker-1.1.20/cluster-glue-libs-1.0.12.linbit-2.0+20180815+be86a9f22546.el7.x86_64.rpm to be installed
Examining ./PreReqs/pacemaker-1.1.20/corosync-2.4.4.linbit-1.0+20181123+918bc4781279.el7.x86_64.rpm: corosync-2.4.4.linbit-1.0+20181123+918bc4781279.el7.x86_64
Marking ./PreReqs/pacemaker-1.1.20/corosync-2.4.4.linbit-1.0+20181123+918bc4781279.el7.x86_64.rpm to be installed
Examining ./PreReqs/pacemaker-1.1.20/corosynclib-2.4.4.linbit-1.0+20181123+918bc4781279.el7.x86_64.rpm: corosynclib-2.4.4.linbit-1.0+20181123+918bc4781279.el7.x86_64
Marking ./PreReqs/pacemaker-1.1.20/corosynclib-2.4.4.linbit-1.0+20181123+918bc4781279.el7.x86_64.rpm to be installed
Examining ./PreReqs/pacemaker-1.1.20/crmsh-3.0.3.linbit-1.0+20181114+2309c21a33a9.el7.noarch.rpm: crmsh-3.0.3.linbit-1.0+20181114+2309c21a33a9.el7.noarch
Marking ./PreReqs/pacemaker-1.1.20/crmsh-3.0.3.linbit-1.0+20181114+2309c21a33a9.el7.noarch.rpm to be installed
Examining ./PreReqs/pacemaker-1.1.20/crmsh-scripts-3.0.3.linbit-1.0+20181114+2309c21a33a9.el7.noarch.rpm: crmsh-scripts-3.0.3.linbit-1.0+20181114+2309c21a33a9.el7.noarch
Marking ./PreReqs/pacemaker-1.1.20/crmsh-scripts-3.0.3.linbit-1.0+20181114+2309c21a33a9.el7.noarch.rpm to be installed
Examining ./PreReqs/pacemaker-1.1.20/libob-1.0.2.linbit-2.0+20180115+e890fcfd0d67.el7.x86_64.rpm: libob-1.0.2.linbit-2.0+20180115+e890fcfd0d67.el7.x86_64
```

Observe the script as it runs. It will take approximately three minutes. You will be notified of the results when complete.

```
ibmuser@rdqm1:~/mq915/MQServer/Advanced/RDQM
File Edit View Search Terminal Help
Examining ../../MQSeriesSamples-9.1.5-0.x86_64.rpm: MQSeriesSamples-9.1.5-0.x86_64
../../MQSeriesSamples-9.1.5-0.x86_64.rpm: does not update installed package.
Examining ../../MQSeriesClient-9.1.5-0.x86_64.rpm: MQSeriesClient-9.1.5-0.x86_64
../../MQSeriesClient-9.1.5-0.x86_64.rpm: does not update installed package.
Resolving Dependencies
--> Running transaction check
--> Package MQSeriesRDQM.x86_64 0:9.1.5-0 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository          Size
=====
Installing:
MQSeriesRDQM     x86_64   9.1.5-0           /MQSeriesRDQM-9.1.5-0.x86_64  142 k

Transaction Summary
=====
Install 1 Package

Total size: 142 k
Installed size: 142 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : MQSeriesRDQM-9.1.5-0.x86_64          1/1
    Verifying  : MQSeriesRDQM-9.1.5-0.x86_64          1/1

Installed:
MQSeriesRDQM.x86_64 0:9.1.5-0

Complete!
[ibmuser@rdqm1 RDQM]$
```

% include note.html content="The script will fail if you have done Lab 12 or 13 because the RDQM

support is already installed.” %}

6. RDQM is now ready as it has been installed with prereqs on all of the VMs.

**Note:**

RHEL default time before the screen locks is very short. If you need it to be longer, you can turn off the screen lock in settings. Applications > System Tools > Settings > Power > Power Saving > Blank screen > Never.

## Configure the firewall

Normally, the firewall would have been configured as a pre-req. However during preparation of this environment, the default RHEL firewall was not configured. You need to do that now for the RDQM cluster.

1. Start the firewall with following command:

```
sudo systemctl start firewalld
```

2. Run the following command to allow MQ, DRDB, and Pacemaker ports to be opened in the firewall:

```
sudo /opt/mqm/samp/rdqm/firewalld/configure.sh
```

```
ibmuser@rdqm1:~/mq915/MQServer/Advanced/RDQM
File Edit View Search Terminal Help
[ibmuser@rdqm1 RDQM]$ sudo systemctl start firewalld
[ibmuser@rdqm1 RDQM]$ sudo /opt/mqm/samp/rdqm/firewalld/configure.sh
+ MQ_INSTALLATION_PATH=/opt/mqm
+ cp /opt/mqm/samp/rdqm/firewalld/services/rdqm-drbd.xml /usr/lib/firewalld/services
+ chmod 644 /usr/lib/firewalld/services/rdqm-drbd.xml
+ cp /opt/mqm/samp/rdqm/firewalld/services/rdqm-pacemaker.xml /usr/lib/firewalld/services
+ chmod 644 /usr/lib/firewalld/services/rdqm-pacemaker.xml
+ cp /opt/mqm/samp/rdqm/firewalld/services/rdqm-mq.xml /usr/lib/firewalld/services
+ chmod 644 /usr/lib/firewalld/services/rdqm-mq.xml
+ firewall-cmd --reload
success
+ firewall-cmd --permanent --add-service=rdqm-drbd
Warning: ALREADY_ENABLED: rdqm-drbd
success
+ firewall-cmd --permanent --add-service=rdqm-pacemaker
Warning: ALREADY_ENABLED: rdqm-pacemaker
success
+ firewall-cmd --permanent --add-service=rdqm-mq
Warning: ALREADY_ENABLED: rdqm-mq
success
+ firewall-cmd --reload
success
[ibmuser@rdqm1 RDQM]$
```

3. Normally you would repeat the above steps on the other VMs, but it is not necessary for this lab as it was preconfigured.

## Configure the HA and DR - HA clusters

### Configure the HA cluster

The cluster must first be created, and then an RDQM instance defined containing one or more queue managers. The RDQM code expects the rdqm.ini file to be in the /var/mqm directory.

1. On node rdqm1, in a terminal window (as **ibmuser**), navigate to /var/mqm.
2. Copy the provided rdqm.ini file to the /var/mqm directory with the following command:

```
cd /var/mqm
sudo cp ~/mq915/rdqm-hadr.ini rdqm.ini
```

```
ibmuser@rdqm1:/var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm1 RDQM]$ cd /var/mqm
[ibmuser@rdqm1 mqm]$ cp ~/mq915/rdqm-hadr.ini rdqm.ini
[ibmuser@rdqm1 mqm]$
```

3. **IMPORTANT:** Repeat the above commands to copy rdqm-hadr.ini on **rdqm2** and **rdqm3** before continuing.

4. Display the rdqm.ini file.

```
cat rdqm.ini
```

```
ibmuser@rdqm1:~var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm1 mqm]$ cat rdqm.ini
# The configuration in this file is not dynamic.
# The HA configuration is read when an HA group is created.
# The DR configuration is read when when a DR/HA queue manager is created.

Node:
  HA_Replication=10.0.1.1
  DR_Replication=10.0.2.1
#  HA_Primary=10.0.3.1
#  HA_Alternate=10.0.4.1
Node:
  HA_Replication=10.0.1.2
  DR_Replication=10.0.2.2
#  HA_Primary=10.0.3.2
#  HA_Alternate=10.0.4.2
Node:
  HA_Replication=10.0.1.3
  DR_Replication=10.0.2.3
#  HA_Primary=10.0.3.3
#  HA_Alternate=10.0.4.3

DRGroup:
  Name=DRG1
    DR_Replication=10.0.2.14
    DR_Replication=10.0.2.15
    DR_Replication=10.0.2.16
[ibmuser@rdqm1 mqm]$
```

If you have done the previous labs this file should look familiar. Let's review the configuration before we continue.

This file must be the same on all the primary HA VMs. The Node stanzas define the addresses to be used for HA\_Replication and DR\_Replication. For each parameter, the subnet used must be the same on all three nodes.

DRGroup stanza is new in MQ 9.1.5 which defines the IP addresses to be used for DR\_Replication to the secondary DR site. The name is DRG1 and uses the same subnet (10.0.2.x) used by the local DR\_Replication parameters.

This allows the entire HA cluster to fail over to another like HA cluster at the DR site. This would be used if the primary HA site was not available.

5. Return to **rdqm1** and enter the command to configure RDQM on the primary machine:

```
sudo rdqmadm -c
```

The screenshot shows a terminal window with the title bar "ibmuser@dr1:/var/mqm". The menu bar includes File, Edit, View, Search, Terminal, and Help. The command `sudo rdqmadm -c` is entered in the terminal. The output shows the configuration process: "Configuring the replicated data subsystem.", "The replicated data subsystem has been configured on this node.", "The replicated data subsystem has been configured on 'dr2'.", "The replicated data subsystem has been configured on 'dr3'.", and finally "The replicated data subsystem configuration is complete." The entire command and its output are highlighted with a red box.

```
ibmuser@dr1:/var/mqm
File Edit View Search Terminal Help
[ibmuser@dr1 mqm]$ sudo rdqmadm -c
Configuring the replicated data subsystem.
The replicated data subsystem has been configured on this node.
The replicated data subsystem has been configured on 'dr2'.
The replicated data subsystem has been configured on 'dr3'.
The replicated data subsystem configuration is complete.
[ibmuser@dr1 mqm]$
```

Wait for the configuration to complete, approximately two minutes.

6. You have received the messages that the replicated data system has been configured on this node and also on remote nodes **rdqm2** and **rdqm3**. You also receive a message that says configuration is complete for your HA group at the primary site.

The screenshot shows a terminal window with the title bar "ibmuser@rdqm1:/var/mqm". The menu bar includes File, Edit, View, Search, Terminal, and Help. The command `sudo rdqmadm -c` is entered in the terminal. The output shows the configuration process: "Configuring the replicated data subsystem.", "The replicated data subsystem has been configured on this node.", "The replicated data subsystem has been configured on 'rdqm2'.", "The replicated data subsystem has been configured on 'rdqm3'.", and finally "The replicated data subsystem configuration is complete." The entire command and its output are highlighted with a red box.

```
ibmuser@rdqm1:/var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm1 mqm]$ sudo rdqmadm -c
Configuring the replicated data subsystem.
The replicated data subsystem has been configured on this node.
The replicated data subsystem has been configured on 'rdqm2'.
The replicated data subsystem has been configured on 'rdqm3'.
The replicated data subsystem configuration is complete.
[ibmuser@rdqm1 mqm]$
```

The RDQM HA pacemaker cluster is now ready.

## Configure the DR - HA cluster

The simulated disaster recovery site has been preconfigured. Let's review the configuration in relation to the simulated high availability site. We will refer to the main HA site as primary and the DR site as secondary.

1. Sign in to **dr1** just as you have done for the rdqmx VMs using the same credentials.
2. Change the directory to `/var/mqm`. Then copy `rdqm-hadr.ini` to `rdqm.ini` as you did previously. But since `dr1`, `dr2`, and `dr3` are at the recovery or secondary site, the values for this HA group are reversed.

```
cd /var/mqm  
cp ~/mq915/rdqm-hadr.ini rdqm.ini
```

3. Display the *rdqm.ini* file. This file is the same on all of the VMs for the DR site; dr1, dr2 and dr3. This file will be used to create another HA group at the secondary site.

```
cat rdqm.ini
```

```
ibmuser@dr1:/var/mqm
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ cd /var/mqm
[ibmuser@dr1 mqm]$ cp ~/mq915/rdqm-hadr.ini rdqm.ini
[ibmuser@dr1 mqm]$ cat rdqm.ini
# The configuration in this file is not dynamic.
# The HA configuration is read when an HA group is created.
# The DR configuration is read when when a DR/HA queue manager is created.

Node:
  HA_Replication=10.0.1.14
  DR_Replication=10.0.2.14
#  HA_Primary=10.0.3.14
#  HA_Alternate=10.0.4.14
Node:
  HA_Replication=10.0.1.15
  DR_Replication=10.0.2.15
#  HA_Primary=10.0.3.15
#  HA_Alternate=10.0.4.15
Node:
  HA_Replication=10.0.1.16
  DR_Replication=10.0.2.16
#  HA_Primary=10.0.3.16
#  HA_Alternate=10.0.4.16

DRGroup:
  Name=DRG2
    DR_Replication=10.0.2.1
    DR_Replication=10.0.2.2
    DR_Replication=10.0.2.3
[ibmuser@dr1 mqm]$
```

The secondary HA cluster must be configured using the same subnets as the primary HA cluster. You will notice that the node definitions for the secondary site have the local IP addresses for dr1, dr2, and dr3. The HA-Replication subnet (10.0.1.x) is the same subnet for HA\_Replication on the primary site. Likewise the DR-Replication subnet (10.0.2.x) is the same subnet for DR\_Replication on the primary site.

#### 4. **IMPORTANT:** Repeat the above commands to copy rdqm-hadr.ini on **dr2** and **dr3** before continuing.

This file must be the same on all the secondary DR VMs. The Node stanzas define the addresses to be used for HA\_Replication and DR\_Replication. For each parameter, the subnet used must be the same on all three nodes.

DRGroup stanza is new in MQ 9.1.5 which defines the IP addresses to be used for DR-Replication to the secondary DR site. The name is DRG2 to distinguish it from the DRGroup on the primary site. It uses the same subnet (10.0.2.x) used by the local DR\_Replication parameters.

This configuration allows the primary site to act as the DR site for the secondary site. In this manner you can have active RDQMs running in an HA group at each site and the other site acting as the failover (recovery site) for the other. This provides resiliency where you can have an active / active architecture. When planning for RDQM (not covered in this lab) you must allow enough headroom on each server so that all the queue managers in your RDQM environment can run on one single server.

5. Return to **dr1** and enter the command to configure the RDQM pacemaker cluster on the DR site:

```
sudo rdqmadm -c
```

6. Wait for the configuration to complete, approximately two minutes. You have received a message that the replicated data system has been configured on this node and remote nodes **dr2** and **dr3**. You are also notified that the configuration is complete.

The screenshot shows a terminal window titled "ibmuser@dr1:~". The window contains a command-line interface with the following text:  
File Edit View Search Terminal Help  
[ibmuser@dr1 ~]\$ sudo rdqmadm -c  
Configuring the replicated data subsystem.  
The replicated data subsystem has been configured on this node.  
The replicated data subsystem has been configured on 'dr2'.  
The replicated data subsystem has been configured on 'dr3'.  
The replicated data subsystem configuration is complete.  
[ibmuser@dr1 ~]\$

The RDQM DR pacemaker cluster is now ready.

## Configure the HA RDQM

### Overview of the steps to create an RDQM

The high availability replicated data queue manager (RDQM) now needs to be created. You use the **crtmqm** command to create a replicated data queue manager (RDQM) in a DR/HA configuration.

You can create a DR/HA RDQM as a user in the **mqm** group if the user can use **sudo**. Otherwise you must create the RDQM as root. User **ibmuser** is a member of the **mqm** group and is a sudoer.

We will refer to the local (HA) site as primary and the remote (DR) site as secondary. We will also refer to the preferred node as primary and the standby nodes as secondary.

Therefore primary/primary means that the RDQM is running on the local site on the preferred node and

primary/secondary means the RDQM is running on the local site on a failed over node. Conversely secondary/primary means the HA group has failed over to the remote DR site and the RDQM is running on the preferred node of the remote HA group. And finally secondary/secondary means that the RDQM HA group has failed over to the remote DR site and the RDQM is not running on the preferred node of the remote HA group.

You must create a number of DR/HA RDQMs:

- On the HA group on the “main” site:
  - On the node where you want the queue manager to run under normal conditions, create the primary/primary DR/HA RDQM.
  - On each of the other two nodes in the HA group, create a primary/secondary DR/HA RDQM.
- On the HA group on the “recovery” site:
  - On the node where the queue manager will run if it fails over to the recovery site, create the secondary/primary DR/HA RDQM. You can use the command output when you created the primary/primary queue manager on the ‘main’ site.
- On each of the other two nodes in the HA group, create a secondary/secondary DR/HA RDQM.
- All of the queue manager instances must have the same name and be allocated the same amount of storage.

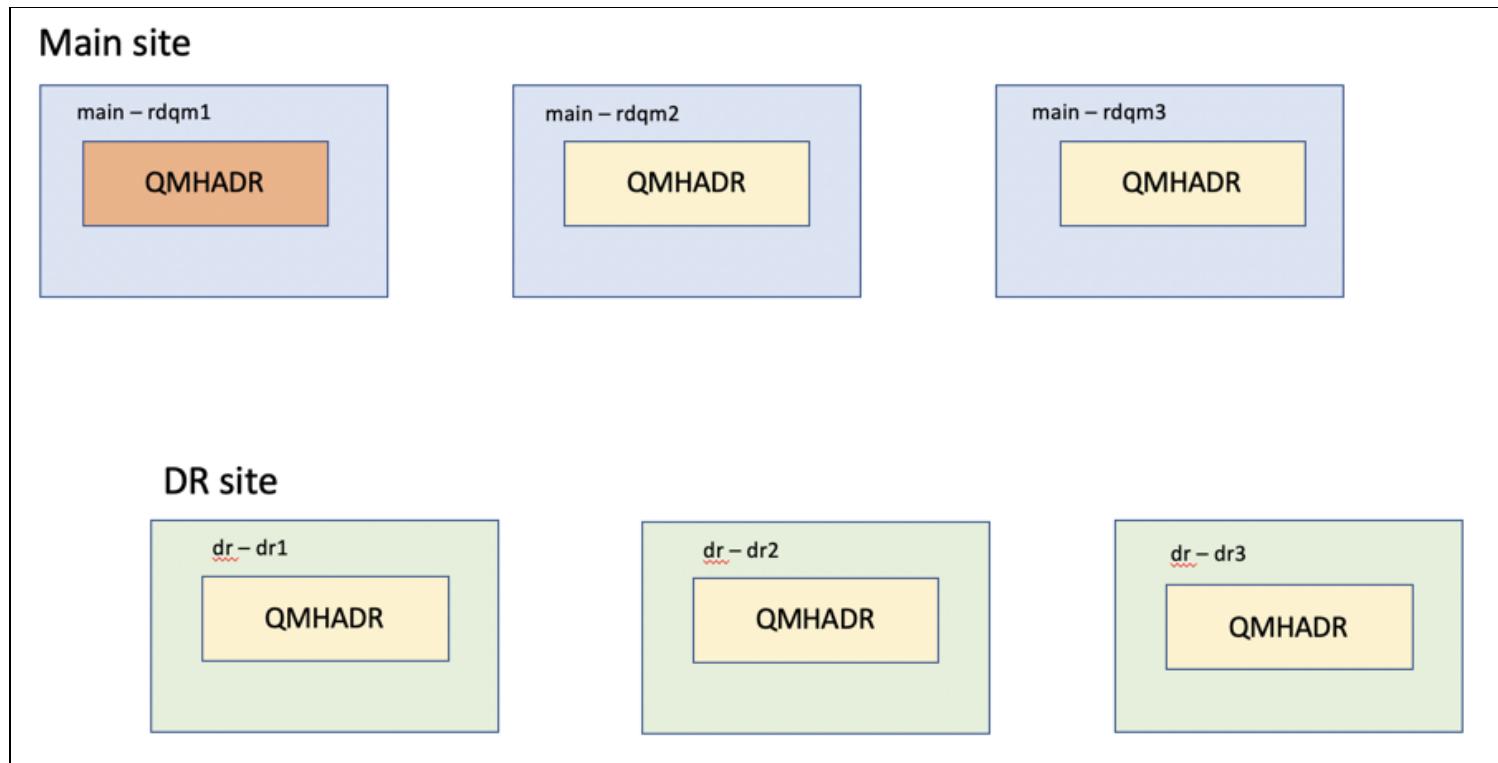
## Create the RDQM

Let's do it.

The example configuration has two sites, named *main* and *dr*. Each site has three nodes. The main site nodes are named **rdqm1**, **rdqm2**, and **rdqm3**. And at the dr site there are matching nodes named **dr1**, **dr2**, and **dr3**.

The nodes have full name consisting of their site name and name, so **main-rdqm1**, **dr-dr1**, and so on.

The following steps create a DR/HA RDQM named QMHADR that runs on **main-rdqm1**. The **main-rdqm1** node is the HA and DR primary.



1. If the local and remote DR IP addresses are specified in the rdqm.ini file, then there is no need to specify any IP addresses on the command line and a DR/HA RDQM named QMHADR can be created by running the command on main-rdqm1. All the IP addresses are in the rdqm.ini, so use this command to create a primary/primary DR/HA RDQM queue manager called **QMHADR** with the following command:

```
sudo crtmqm -sx -rr p -rn DRG1 -rp 7001 -fs 3 QMHADR
```

The screenshot shows a terminal window titled "ibmuser@rdqm1:/var/mqm". The command entered is "sudo crtqm -sx -rr p -rn DRG1 -rp 7001 -fs 3 QMHADR". The output indicates the creation of a replicated data queue manager configuration, mentioning secondary queue managers created on 'rdqm2' and 'rdqm3'. It also notes the creation of an IBM MQ queue manager and the setup of default objects for the queue manager 'QMHADR'. A callout bubble points to the text "Secondary queue managers created" with the annotation "Secondary queue managers created". Another callout bubble points to the command "crtqm -sx -rr s -rl 10.0.2.14,10.0.2.15,10.0.2.16 -ri 10.0.2.1,10.0.2.2,10.0.2.3 -rp 7001 -fs 3072M QMHADR" with the annotation "Copy this command to run on dr1".

```
[ibmuser@rdqm1 mqm]$ sudo crtqm -sx -rr p -rn DRG1 -rp 7001 -fs 3 QMHADR
Creating replicated data queue manager configuration.
Secondary queue manager created on 'rdqm2'.
Secondary queue manager created on 'rdqm3'.
IBM MQ queue manager created.
Directory '/var/mqm/vols/qmhadr/qmgr/qmhadr' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QMHADR'.
Default objects statistics : 83 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
Enabling replicated data queue manager.
Replicated data queue manager enabled.
Issue the following command on the remote HA group to create the DR/HA
secondary queue manager:
crtqm -sx -rr s -rl 10.0.2.14,10.0.2.15,10.0.2.16 -ri 10.0.2.1,10.0.2.2,10.0.2.3 -rp 7001 -fs 3072M QMHADR
[ibmuser@rdqm1 mqm]$
```

Examples of equivalent command line commands:

- If the local DR IP addresses are not specified in the rdqm.ini file, then the remote DR IP addresses can be specified on the command line:

```
sudo crtqm -sx -rr p -ri 10.0.2.14,10.0.2.15,10.0.2.16 -rp 7001 -fs 3 QMHADR
```

- If no DR IP addresses are specified in the rdqm.ini file, then both remote and local DR IP addresses can be specified on the command line:

```
crtqm -sx -rr p -rl 10.0.2.1,10.0.2.2,10.0.2.3 -ri 10.0.2.14,10.0.2.15,10.0.2.16
-rp 7001 QM1
```

2. The primary/primary queue manager has been created. And the command to run on the secondary/primary nodes has been provided.

```
sudo crtqm -sx -rr p -rl 10.0.2.14,10.0.2.15,10.0.2.16 -ri 10.0.2.1,10.0.2.2,10.0.2.3 -rp 7001 -fs 3 QMHADR
```

3. Copy the provided command provided in the output.
4. Return to node **dr1**. In the terminal window, paste the command and run it.

```
ibmuser@dr1:~var/mqm
File Edit View Search Terminal Help
[ibmuser@dr1 mqm]$ crtqm -sx -rr s -rl 10.0.2.14,10.0.2.15,10.0.2.16 -ri 10.0.2.1,10.0.2.2,10.0.2.3 -rp 7001 -fs 3072M QMHADR
Creating replicated data queue manager configuration.
Secondary queue manager created on 'dr2'.
Secondary queue manager created on 'dr3'.
IBM MQ secondary queue manager created.
Enabling replicated data queue manager.
Replicated data queue manager enabled.
Command '/opt/mqm/bin/crtqm' run with sudo.
[ibmuser@dr1 mqm]$
```

**⚠ Important:**

Be careful when copying / pasting the command from **rdqm1**. On **rdqm1**, copy the command to the Skytap clipboard, then copy from rdqm1's Skytap clipboard and paste it into dr1's Skytap clipboard. Then you can paste the command on the command line in **dr1**. Ask your instructor for assistance if you don't get the correct command pasted.

## Create a floating IP address for the HA / DR queue manager

You can create floating IP addresses for each of your HA groups in a DR/HA RDQM configuration.

A floating IP address enables a client to use the same IP address for a DR/HA RDQM regardless of which node in an HA group it is running on. If your two HA groups have private/isolated networks for application connectivity, then the same floating IP address can be defined for both groups. You must still define that floating IP address twice, however, once on each of your HA groups.

Since our Skytap template is using one network, we can create one floating IP that can be used on either HA group. The floating IP address must be defined in the same subnet as the HA\_Replication parameter in the rdqm.ini file.

1. Return to **rdqm1** and in the terminal window run the following command to create the floating IP.

```
sudo rdqmint -m QMHADR -a -f 10.0.1.20 -l ens35
```

The screenshot shows a terminal window with a red border. The title bar says "ibmuser@rdqm1:~". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command [ibmuser@rdqm1 ~]\$ sudo rdqmint -m QMHADR -a -f 10.0.2.20 -l ens36 is highlighted with a red box. The output shows: "The floating IP address '10.0.2.20' has been added to queue manager 'QMHADR'." followed by three blank lines.

2. Enter the same command on **dr1**.

## Create MQ resources

To run sample programs you will define two queues: for *SOURCE* and *TARGET*. You will also create a new channel using the 'MQ' IP address for each of the three nodes in our cluster (as the queue manager could run on any one of them) and the listener port for the queue manager.

It is recommended to add another listener specifically for the floating point IP address. By default, the standard listener will listen on the same port on every IP address, so a different port needs to be chosen for the additional listener.

You will define the RDQM with default port 1414 which is already allowed thru the firewall. Since you will add another listener, with a separate port, we need to add that port to the firewall rules. Assign port 1420 to second listener.

You will also turn off CHLAUTH and CONNAUTH completely to keep things simple.

1. On rdqm1 where the QMHADR queue manager is running, in ibmuser's terminal window, run the command:

```
runmqsc QMHADR
```

Run the following MQSC commands (remember that MQ objects are case sensitive):

```
ALTER QMGR CHLAUTH(DISABLED) CONNAUTH(' ')
```

```
REFRESH SECURITY TYPE(CONNAUTH)
```

```
DEFINE QLOCAL(SOURCE) DEFPSIST(YES)
```

```
DEFINE QLOCAL(TARGET) DEFPSIST(YES)
```

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP)
```

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) CONNAME('10.0.1.1(1414),1  
0.0.1.2(1414),10.0.1.3(1414)') QMNAME(QMHADR)
```

```
START LISTENER
```

```
DEFINE LISTENER(FLIP) TRPTYPE(TCP) CONTROL(QMGR) IPADDR(10.0.1.20) PORT(1420)
```

```
START LISTENER(FLIP)
```

```
END
```

```
ibmuser@rdqm1:~var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm1 mqm]$ runmqsc QMHADR
5724-H72 (C) Copyright IBM Corp. 1994, 2020.
Starting MQSC for queue manager QMHADR.

ALTER QMGR CHLAUTH(DISABLED) CONNAUTH(' ')
 1 : ALTER QMGR CHLAUTH(DISABLED) CONNAUTH(' ')
AMQ8005I: IBM MQ queue manager changed.
REFRESH SECURITY TYPE(CONNAUTH)
 2 : REFRESH SECURITY TYPE(CONNAUTH)
AMQ8560I: IBM MQ security cache refreshed.
DEFINE QLOCAL(SOURCE) DEFPSIST(YES)
 3 : DEFINE QLOCAL(SOURCE) DEFPSIST(YES)
AMQ8006I: IBM MQ queue created.
DEFINE QLOCAL(TARGET) DEFPSIST(YES)
 4 : DEFINE QLOCAL(TARGET) DEFPSIST(YES)
AMQ8006I: IBM MQ queue created.
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) CONNAME('10.0.1.1(1414),10.0.1.2(1414),10.0.1.3(1414)') QMNAME(QMHADR)
 5 : DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) CONNAME('10.0.1.1(1414),10.0.1.2(1414),10.0.1.3(1414)') QMNAME(QMHADR)
AMQ8014I: IBM MQ channel created.
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP)
 6 : DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP)
AMQ8014I: IBM MQ channel created.
START LISTENER
 7 : START LISTENER
AMQ8021I: Request to start IBM MQ listener accepted.
DEFINE LISTENER(FLIP) TRPTYPE(TCP) CONTROL(QMGR) IPADDR(10.0.1.20) PORT(1420)
 8 : DEFINE LISTENER(FLIP) TRPTYPE(TCP) CONTROL(QMGR) IPADDR(10.0.1.20) PORT(1420)
AMQ8626I: IBM MQ listener created.
START LISTENER(FLIP)
 9 : START LISTENER(FLIP)
AMQ8021I: Request to start IBM MQ listener accepted.
end
 10 : end
9 MQSC commands read.
No commands have a syntax error.
All valid MQSC commands were processed.
[ibmuser@rdqm1 mqm]$
```

2. The new listener will not be started until the queue manager is restarted. To stop the RDQM, enter the following command on the DR/HA RDQM's primary node (rdqm1):

```
endmqm QMHADR
```

Pacemaker ceases to manage the RDQM, and then the RDQM is ended. All other *endmqm* parameters can be used when stopping an RDQM.

3. Restart the RDQM with the following command on the RDQM's primary node (rdqm1):

```
strmqm QMHADR
```

```
ibmuser@rdqm1:~
File Edit View Search Terminal Help
[ibmuser@rdqm1 ~]$ endmqm QMHADR
Replicated data queue manager disabled.
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
[ibmuser@rdqm1 ~]$ strmqm QMHADR
IBM MQ queue manager 'QMHADR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMHADR' during the log replay phase.
Log replay for queue manager 'QMHADR' complete.
Transaction manager state recovered for queue manager 'QMHADR'.
Replicated data queue manager enabled.
IBM MQ queue manager 'QMHADR' started using V9.1.5.0.
[ibmuser@rdqm1 ~]$
```

The RDQM is started, and Pacemaker starts managing the RDQM. You must specify the `-ns` option with `strmqm` if you want to specify any other `strmqm` options.

4. You can check that both listeners are running by running the following command:

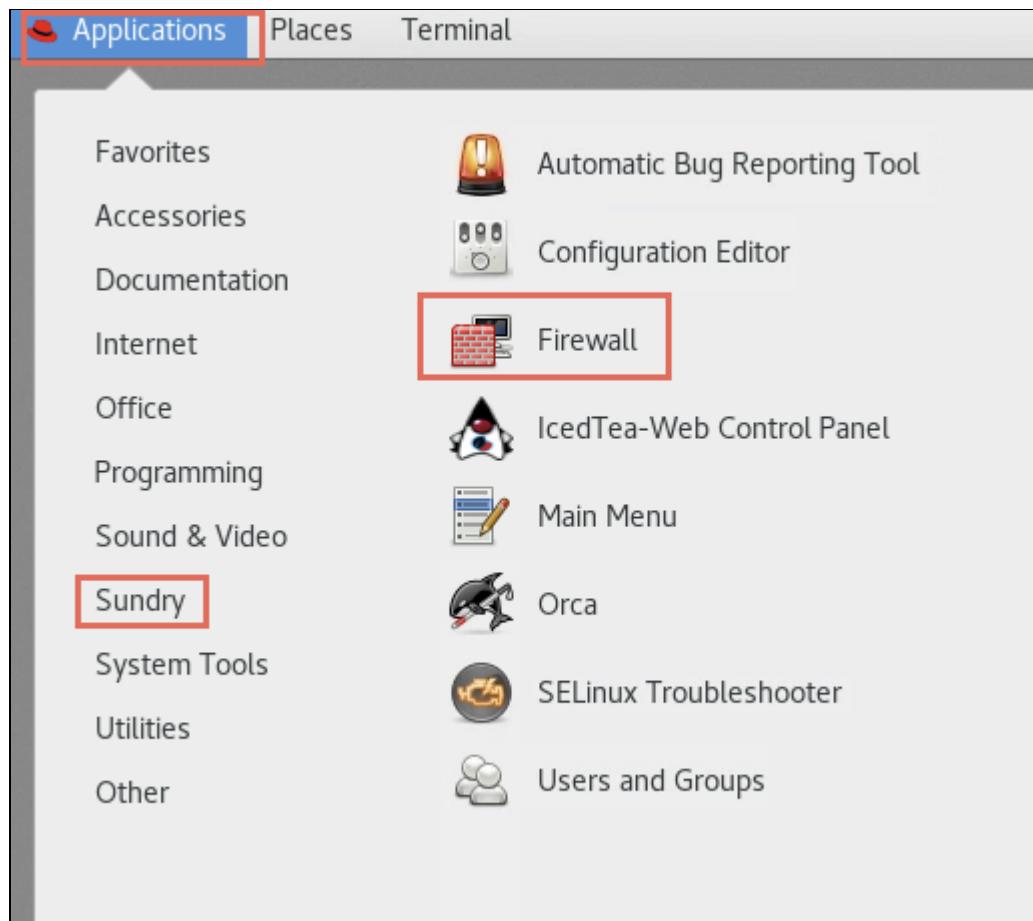
```
netstat -ant | grep 14
```

```
ibmuser@rdqm1:/var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm1 mqm]$ netstat -ant | grep 14
tcp        0      0 10.0.2.1:7001          10.0.2.14:57443      ESTABLISHED
tcp        0      0 10.0.2.1:54513         10.0.2.14:7001      ESTABLISHED
tcp6       0      0 10.0.2.20:1420        ::::*                  LISTEN
[ibmuser@rdqm1 mqm]$
```

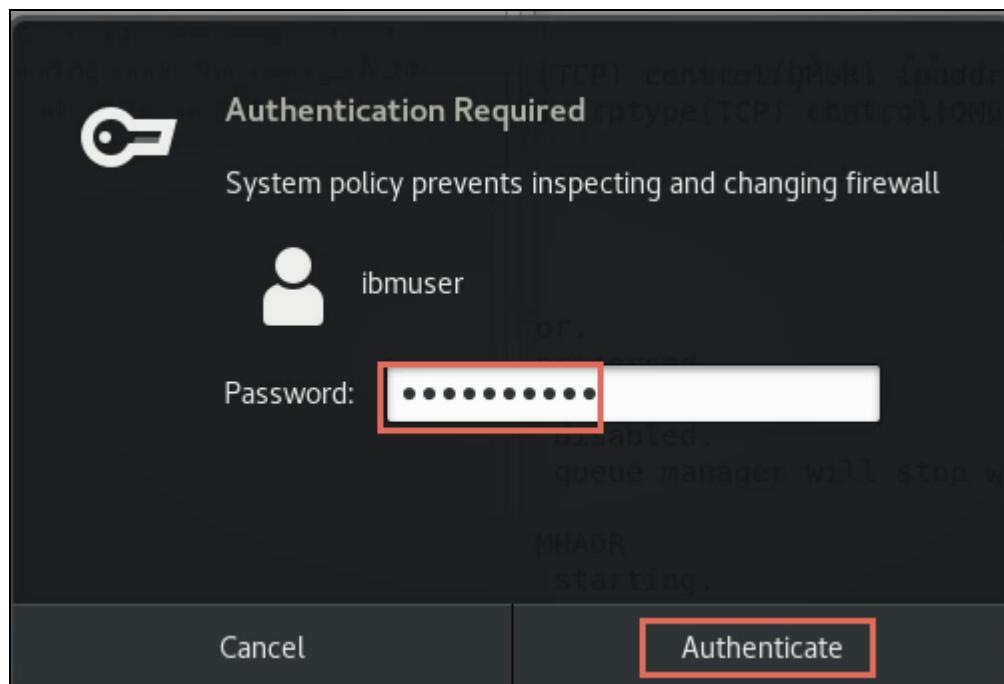
The first listener is the one created because `-p 1414` was specified on the `crtmqm` command. This listener is listening on port 1414 on every IP address. The second listener, however, is listening on port **1420** on the floating IP address only.

## Update the firewall

1. Update the firewall to allow port 1420. Click *Applications > Sundry > Firewall*.

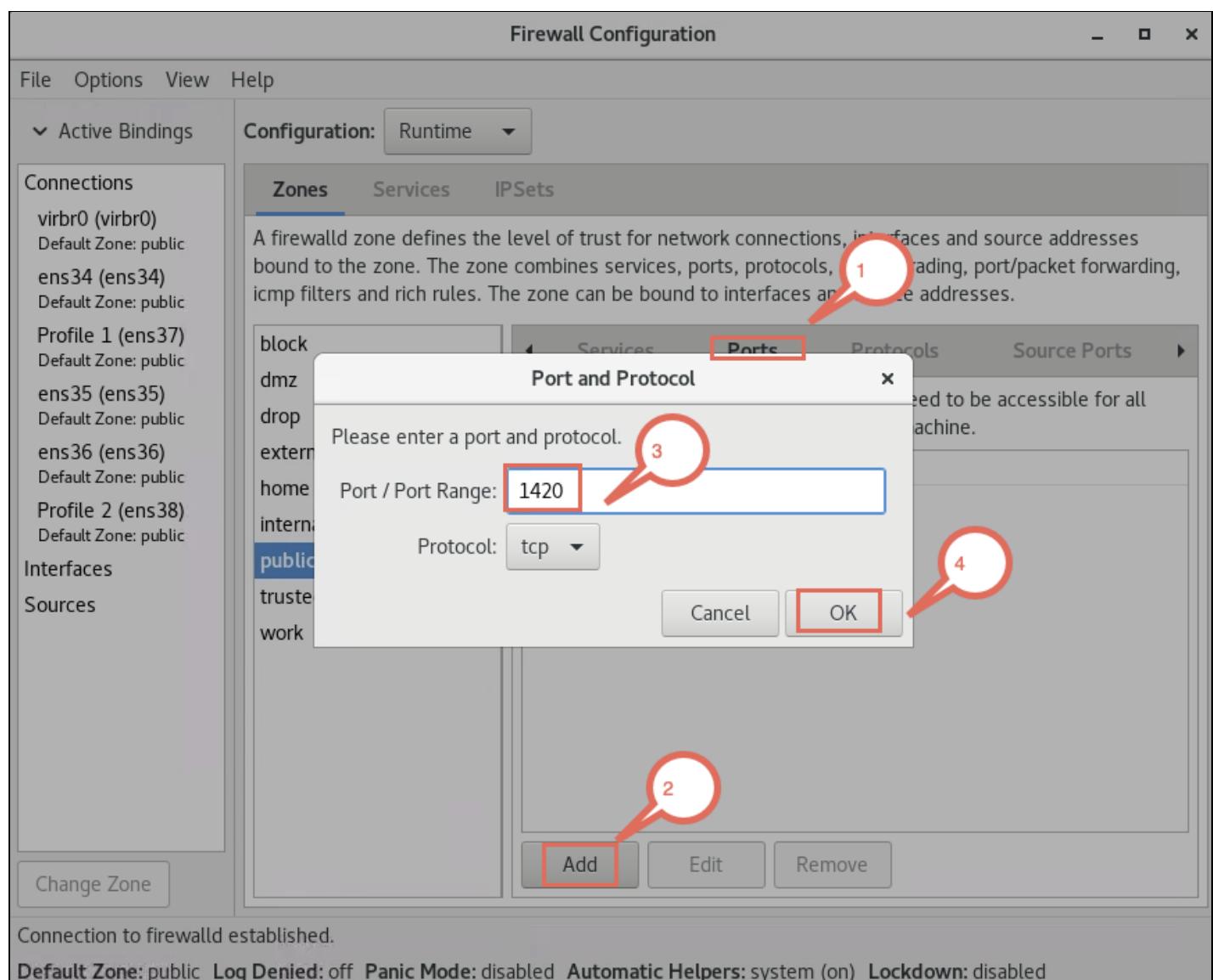


2. Enter the password for ibmuser - **enageibm** and click *Authenticate*.



3. Click *Ports, Add*, then enter **1420** and click *OK*. Repeat to add port **1415** for an additional queue manager

to be defined later.



Close the *Firewall Configuration* window.

4. Repeat the firewall update on all other servers - **rdqm2**, **rdqm3**, **dr1**, **dr2**, and **dr3**.
5. Now that you have a floating IP address associated with QMHADR, you can change the MQSERVER environment variable to CHANNEL1/TCP/10.0.2.0(1420) when you run applications.

```
export MQSERVER='CHANNEL1/TCP/10.0.2.0(1420)'
```

## Simple testing of RDQM

Now that you have two active HA groups, one primary with the primary queue manager and a secondary with the secondary queue manager created, you can proceed to test the failover of the HA group.

When operating in a DR/HA environment there are separate considerations for high availability and disaster recovery.

If the node on which a DR/HA RDQM is running fails, the RDQM automatically fails over to another node in that HA group. If the entire site fails, then you must manually start the RDQM on the preferred node in the HA group on the recovery site. The considerations here are the same as for an ordinary DR RDQM covered in Lab 3.

### Understanding RDQM status

#### Using `dspmq`

1. To view the state of an RDQM, enter the following command on **rdqm1**.

```
dspmq -m QMHADR
```

The screenshot shows a terminal window with the title bar "ibmuser@rdqm1:~". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command prompt is "[ibmuser@rdqm1 ~]\$". The user has run the command "dspmq -m QMHADR". The output shows "QMNAME(QMHADR)" followed by a red box containing "STATUS(Running)". The command prompt "[ibmuser@rdqm1 ~]\$" appears again at the bottom.

```
[ibmuser@rdqm1 ~]$ dspmq -m QMHADR
QMNAME(QMHADR)
[ibmuser@rdqm1 ~]$ STATUS(Running)
[ibmuser@rdqm1 ~]$
```

The state information that is output depends on whether you run the command on the RDQM's primary or secondary node. If run on the primary node then one of the normal status messages returned by `dspmq` is displayed - *Running*.

2. Run the same command on **dr1**.

```
ibmuser@dr1:~  
File Edit View Search Terminal Help  
[ibmuser@dr1 ~]$ dspmq -m QMHADR  
QMNAME(QMHADR)  
[ibmuser@dr1 ~]$ STATUS(Ended immediately)
```

If you run the command on a secondary node then the status *Ended immediately* is displayed.

3. You can use arguments with `dspmq` to establish whether an RDQM is configured for disaster recovery, and whether it is currently the primary or the secondary instance. Return to `rdmq1` and run the command again with these parameters:

```
dspmq -m QMHADR -o dr
```

```
ibmuser@rdqm1:~  
File Edit View Search Terminal Help  
[ibmuser@rdqm1 ~]$ dspmq -m QMHADR -o dr  
QMNAME(QMHADR)  
[ibmuser@rdqm1 ~]$ DRROLE(Primary)
```

One of the following responses is displayed:

- DRROLE() Indicates that the queue manager is not configured for disaster recovery.
- DRROLE(Primary) Indicates that the queue manager is configured as the DR primary.
- DRROLE(Secondary) Indicates that the queue manager is configured as the DR secondary.

QMHADR is running on `rdmq1`, so the DRROLE shows *Primary*.

4. Run the same command on `dr1`.

```
ibmuser@dr1:~  
File Edit View Search Terminal Help  
[ibmuser@dr1 ~]$ dspmq -m QMHADR -o dr  
QMNAME(QMHADR)  
[ibmuser@dr1 ~]$ DRROLE(Secondary)
```

DRROLE shows *Secondary*.

5. Use the `dspmq -o all` command to view the disaster recovery and high availability information for DR/HA RDQMs. For example, if you run `dspmq -o all` on the node where the DR/HA RDQM is running, **rdqm1**, you see the following state information:

```
dspmq -o all
```

QMNAME (QMADR)	STATUS (Running) DE
FAULT(no) STANDBY(Not permitted) TNSTNAME(Installation1) INSTPATH(/opt/mqm) INSTVER(9.1.5.0)	HA(Replicated) DRROLE(Primary)
QMNAME (QMHA)	STATUS (Running) DE
FAULT(no) STANDBY(Not permitted) TNSTNAME(Installation1) INSTPATH(/opt/mqm) INSTVER(9.1.5.0)	HA(Replicated) DRROLE()

You see that QMHADR is Running and has DRROLE Primary and the HA status is Replicated. Since you specified all, any other queue managers such as QMHA will also appear as Running with HA status as Replicated, but no DRROLE. This is expected if you ran the first Lab 1 where you created the RDQM QMHA which is running in an HA cluster but was not defined for DR.

**Note:**

When using the `dspmq` commands, your environment may not look the same as the screen shots. This depends on whether you ran Labs 2 and 3 before this lab and whether you ran the cleanup in those labs. You may also have experimented with other non-RDQM queue managers such as QMJ in the screen shots.

6. If you run the same command on **rdqm2**, you see that the display is the same except QMHADR is Running elsewhere. The DRROLE is still Primary because in a DR relationship the *HA cluster* is primary or secondary, not the node.

```
ibmuser@rdqm2:~/mq915
File Edit View Search Terminal Help
[ibmuser@rdqm2 mq915]$ dspmq -o all
QMNAME(QMHADR) STATUS(Running elsewhere) DEFAULT(no)
STVER(9.1.5.0) HA(Replicated) DRROLE(Primary)
QMNAME(QMHA) STATUS(Running elsewhere) DEFAULT(no)
STVER(9.1.5.0) HA(Replicated) DRROLE()
[ibmuser@rdqm2 mq915]$
```

7. Run the command on **dr1**.

```
ibmuser@dr1:~
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ dspmq -o all
QMNAME(QMJ) STATUS(Running) DEFAULT(no)
/mqm INSTVER(9.1.5.0) HA(Replicated) DRROLE()
QMNAME(QMHADR) STATUS(Ended immediately) DE
STPATH(/opt/mqm) INSTVER(9.1.5.0) HA(Replicated) DRROLE(Secondary)
[ibmuser@dr1 ~]$
```

QMHADR is not running so *Ended immediately* is displayed as status. The DRROLE is Secondary as expected and the HA status is replicated. There is another RDQM which is running with no DRROLE and has HA status Replicated. This RDQM is running in the DR site's HA cluster and is not known in the Main site HA cluster.

### Using rdqmstatus

You use the rdqmstatus command to view the status of individual RDQMs or get an overview of the status of all the RDQMs known to the HA group.

You must be a user in the mqm and haclient groups to run the rdqmstatus command so the environment has been preconfigured in that way. You can run the command on any of the nodes in either of the HA groups.

1. On **rdqm1** issue the following *rdqmstatus* command.

```
sudo rdqmstatus
```

```
ibmuser@rdqm1:~$ sudo rdqmstatus
Node: rdqm1
Queue manager name: QMHADR
Queue manager status: Running
HA current location: This node
DR role: Primary

Queue manager name: QMHA
Queue manager status: Running
HA current location: This node
[ibmuser@rdqm1 ~]$
```

The result is similar to the `dspmq` display.

## 2. Issue the command on `dr1`.

```
ibmuser@dr1:~$ sudo rdqmstatus
Node: dr1
Queue manager name: QMJ
Queue manager status: Running
HA current location: This node

Queue manager name: QMHADR
Queue manager status: Ended immediately
HA current location: This node
DR role: Secondary
[ibmuser@dr1 ~]$
```

Again the displays are very similar to that of `dspmq`.

## 3. When you specify the queue manager on the `rdqmstatus` command is where things get interesting. Switch to `rdqm1` and issue the following command.

```
sudo rdqmstatus -m QMHADR
```

```
ibmuser@rdqm1:~ - □
File Edit View Search Terminal Help
[ibmuser@rdqm1 ~]$ sudo rdqmstatus -m QMHADR
Node: rdqm1
Queue manager status: Running
CPU: 0.00%
Memory: 181MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: ens36
HA floating IP address: 10.0.2.20
DR role: Primary
DR status: Normal
DR port: 1414
DR local IP address: 10.0.2.1
DR remote IP address list: 10.0.2.14,10.0.2.15,10.0.2.16
DR current remote IP address: 10.0.2.14

Node: rdqm2
HA status: Normal

Node: rdqm3
HA status: Normal
[ibmuser@rdqm1 ~]$
```

Look closely at the HA and DR fields. The HA role is currently *Primary* (that's why it is running here), HA status is *Normal* and HA control is *enabled*. Both the HA preferred and current locations are *This node*. The HA floating IP interface and address are displayed.

DR role is *Primary* and status is *Normal*. The DR local replication IP address and port are shown and the list of remote DR IP addresses with the current remote IP address.

### *Failing over an RDQM instance to another node*

The easiest way to force an RDQM instance to fail over to another node in the local Main site HA cluster is to

change its preferred location.

The default location for RDQM is **rdqm1**. You will fail the RDQM instance to Main site node 2, **rdqm2**.

1. Switch to **rdqm2**.
2. Open a new terminal window, and as the user `ibmuser`, issue the command to set the MQ environment.  
Make this node the primary instance with the following command:

```
sudo rdqmadm -m QMHADR -p
```

3. Confirm that **rdqm2** is now the primary node:

```
sudo rdqmstatus -m QMHADR
```

```
ibmuser@rdqm2:~$ sudo rdqmadm -m QMHADR -p
The preferred replicated data node has been set to 'rdqm2' for queue manager
'QMHADR'.
[ibmuser@rdqm2 ~]$ sudo rdqmstatus -m QMHADR
Node: rdqm2
Queue manager status: Running
CPU: 2.60%
Memory: 182MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: ens36
HA floating IP address: 10.0.2.20
DR role: Primary
DR status: Normal
DR port: 1414
DR local IP address: 10.0.2.2
DR remote IP address list: 10.0.2.14,10.0.2.15,10.0.2.16
DR current remote IP address: 10.0.2.14

Node: rdqm1
HA status: Normal

Node: rdqm3
HA status: Normal
[ibmuser@rdqm2 ~]$
```

4. Now you can move the queue manager back to rdqm1. Return to **rdqm1** and run the **sudo rdqmadm** command again.

```
sudo rdqmadm -m QMHADR -p
```

Check the status again to see it is now running on **rdqm1**.

```
sudo sudo rdqmstatus -m QMHADR
```

The screenshot shows a terminal window titled "ibmuser@rdqm1:~". The window contains the following command and its output:

```
[ibmuser@rdqm1 ~]$ sudo rdqmadm -m QMHADR -p
The preferred replicated data node has been set to 'rdqm1' for queue manager 'QMHADR'.
[ibmuser@rdqm1 ~]$ sudo rdqmstatus -m QMHADR
Node: rdqm1
Queue manager status: Running
CPU: 0.74%
Memory: 184MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: ens36
HA floating IP address: 10.0.2.20
DR role: Primary
DR status: Normal
DR port: 1414
DR local IP address: 10.0.2.1
DR remote IP address list: 10.0.2.14,10.0.2.15,10.0.2.16
DR current remote IP address: 10.0.2.14

Node: rdqm2
HA status: Normal

Node: rdqm3
HA status: Normal
[ibmuser@rdqm1 ~]$
```

### Move the RDQM by suspending a node

Another test is to move a RDQM by suspending the node on which it is running, as you may want to do when applying a Fix Pack.

1. On the node where QMHADR is running (**rdqm1**), return to ibmuser's terminal window. As the user **ibmuser**, issue the command to suspend the queue manager:

```
sudo rdqmadm -s
```

2. Check the status of the RDQM with the following command:

```
sudo rdqmstatus -m QMHADR
```

As shown in the display, the replicated data node is suspended and goes into standby.

The screenshot shows a terminal window titled "ibmuser@rdqm1:~". The window contains the following text:

```
File Edit View Search Terminal Help
[ibmuser@rdqm1 ~]$ sudo rdqmadm -s
The replicated data node 'rdqm1' has been suspended.
[ibmuser@rdqm1 ~]$ sudo rdqmstatus -m QMHADR
Node: rdqm1
Queue manager status: Running elsewhere
HA role: Unknown
HA status: This node in standby
HA control: Enabled
HA current location: rdqm2
HA preferred location: This node
HA floating IP interface: ens36
HA floating IP address: 10.0.2.20
DR role: Primary
DR status: See rdqm2
DR port: 1414
DR local IP address: 10.0.2.1
DR remote IP address list: 10.0.2.14,10.0.2.15,10.0.2.16
DR current remote IP address: See rdqm2

Node: rdqm2
HA status: Unknown

Node: rdqm3
HA status: Unknown
[ibmuser@rdqm1 ~]$
```

3. Switch to **rdqm2** and issue the command to display the status of QMHADR and you see that it is now

running on rdqm2.

```
sudo rdqmstatus -m QMHADR
```

```
ibmuser@rdqm2:~$ sudo rdqmstatus -m QMHADR
Node: rdqm2
Queue manager status: Running
CPU: 0.00%
Memory: 181MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
HA role: Primary
HA status: Mixed
HA control: Enabled
HA current location: This node
HA preferred location: rdqm1
HA floating IP interface: ens36
HA floating IP address: 10.0.2.20
DR role: Primary
DR status: Normal
DR port: 1414
DR local IP address: 10.0.2.2
DR remote IP address list: 10.0.2.14,10.0.2.15,10.0.2.16
DR current remote IP address: 10.0.2.14

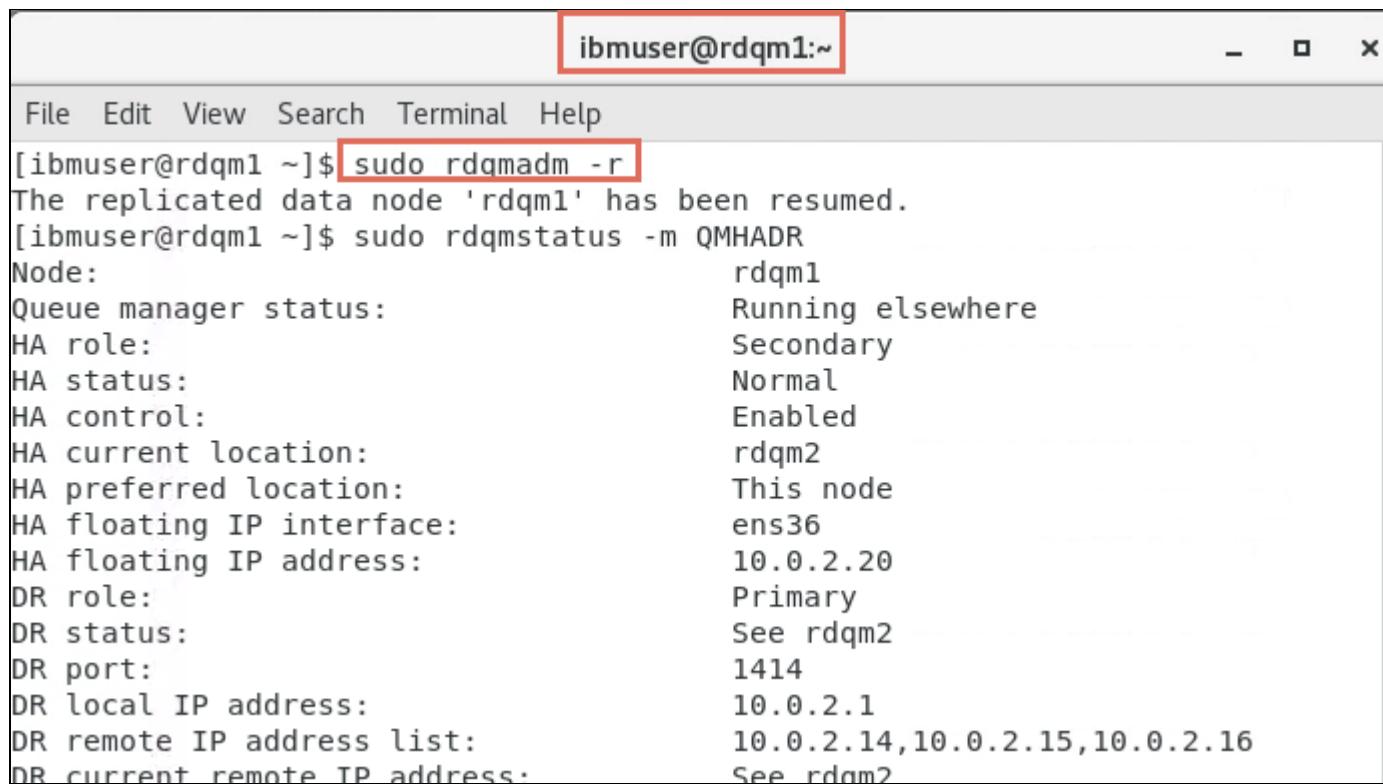
Node: rdqm1
HA status: Remote node in standby
HA out of sync data: 656KB

Node: rdqm3
HA status: Normal
[ibmuser@rdqm2 ~]$
```

4. Return to **rdqm1**, issue the command to resume the replicated data node in the cluster.

```
sudo rdqmadm -r
```

Quickly run the status command again. QMHADR will initially run in a secondary role on this node. If you aren't quick enough, you may not catch this transitory state.



```
ibmuser@rdqm1:~$ sudo rdqmadm -r
The replicated data node 'rdqm1' has been resumed.
[ibmuser@rdqm1 ~]$ sudo rdqmstatus -m QMHADR
Node: rdqm1
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: rdqm2
HA preferred location: This node
HA floating IP interface: ens36
HA floating IP address: 10.0.2.20
DR role: Primary
DR status: See rdqm2
DR port: 1414
DR local IP address: 10.0.2.1
DR remote IP address list: 10.0.2.14,10.0.2.15,10.0.2.16
DR current remote IP address: See rdqm2
```

5. After the node has fully resumed, QMHADR will run in a primary role on this node, as it was prior to being suspending. Issue the status command again to confirm that this has indeed happened.

```
sudo rdqmstatus -m QMHAHR
```

```
ibmuser@rdqm1:~$ sudo rdqmstatus -m QMHADR
Node: rdqm1
Queue manager status: Running
CPU: 5.16%
Memory: 184MB
Queue manager file system: 59MB used, 2.9GB allocated [2%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: ens36
HA floating IP address: 10.0.2.20
DR role: Primary
DR status: Normal
DR port: 1414
DR local IP address: 10.0.2.1
DR remote IP address list: 10.0.2.14,10.0.2.15,10.0.2.16
DR current remote IP address: 10.0.2.14

Node: rdqm2
HA status: Normal

Node: rdqm3
HA status: Normal
[ibmuser@rdqm1 ~]$
```

### *Failing over the RDQM to another HA remote cluster*

1. To test the disaster recovery features of QMHADR, stop the queue manager on Main site node **rdqm1**.

endmqm QMHADR

2. Run the following command on Main **rdqm1** to make QMHADR the DR secondary instance:

```
sudo rdqmldr -m QMHADR -s
```

Queue manager **QMHADR** has been made the DR secondary on this node.

3. Display the status of QMHADR on **rdqm1**.

```
sudo rdqmstatus -m QMHADR
```

```
ibmuser@rdqm1:/var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm1 mqm]$ endmqm QMHADR
Replicated data queue manager disabled.
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
[ibmuser@rdqm1 mqm]$ sudo rdqmldr -m QMHADR -s
Queue manager 'QMHADR' has been made the DR secondary on this node.
[ibmuser@rdqm1 mqm]$ sudo rdqmstatus -m QMHADR
Node: rdqm1
Queue manager status: Ended immediately
HA role: Primary
HA status: Normal
HA control: Disabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: ens36
HA floating IP address: 10.0.2.20
DR role: Secondary
DR status: Normal
DR port: 7001
DR local IP address: 10.0.2.1
DR remote IP address list: 10.0.2.14,10.0.2.15,10.0.2.16
DR current remote IP address: 10.0.2.14

Node: rdqm2
HA status: Normal

Node: rdqm3
HA status: Normal
[ibmuser@rdqm1 mqm]$
```

Notice that the queue manager has ended and its *DRRole* is *Secondary*. Although its *HARole* is still *Primary*.

4. Run the following command on DR site node **dr1** to make QMHADR the DR primary instance on that node:

```
sudo rdqmldr -m QMHADR -p
```

Queue manager **QMHADR** has been made the DR primary on this node.

```
ibmuser@dr1:/var/mqm
File Edit View Search Terminal Help
[ibmuser@dr1 mqm]$ sudo rdqmldr -m QMHADR -p
Queue manager 'QMHADR' has been made the DR primary on this node.
[ibmuser@dr1 mqm]$
```

5. Start RDQM QMHADR.

```
strmqm QMHADR
```

6. Display the status of RDQM QMHADR on DR site node **dr1**. Note that **dr1** is now the HA and DR Primary.

```
sudo rdqmstatus -m QMHADR
```

```
ibmuser@dr1:~$ sudo rdqmstatus -m QMHADR
[ibmuser@dr1 mqm]$ sudo rdqmstatus -m QMHADR
Node: dr1
Queue manager status: Running
CPU: 0.00%
Memory: 181MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: ens36
HA floating IP address: 10.0.2.20
DR role: Primary
DR status: Normal
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address list: 10.0.2.1,10.0.2.2,10.0.2.3
DR current remote IP address: 10.0.2.1

Node: dr2
HA status: Normal

Node: dr3
HA status: Normal
[ibmuser@dr1 mqm]$
```

### *Move the RDQM by suspending a node on the DR HA cluster*

Now you can test the HA functionality on the DR site HA cluster. Move a RDQM by suspending the node on which it is running.

1. On the node where QMHADR is running (**dr1**), return to ibmuser's terminal window. As the user **ibmuser**, issue the command to suspend the queue manager:

```
sudo rdqadm -s
```

2. Check the status of the RDQM with the following command:

```
sudo rdqmstatus -m QMHADR
```

As shown in the display, the replicated data node is suspended and goes into standby.

The screenshot shows a terminal window titled "ibmuser@dr1:~". The window contains the following text:

```
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ sudo rdqmadm -s
The replicated data node 'dr1' has been suspended.
[ibmuser@dr1 ~]$ dspmq
QMNAME(QMJ)
QMNAME(QMHADR)
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMHADR
Node: dr1
Queue manager status: Running elsewhere
HA role: Unknown
HA status: This node in standby
HA control: Enabled
HA current location: dr2
HA preferred location: This node
HA floating IP interface: ens36
HA floating IP address: 10.0.2.20
DR role: Primary
DR status: See dr2
DR port: 1414
DR local IP address: 10.0.2.14
DR remote IP address list: 10.0.2.1,10.0.2.2,10.0.2.3
DR current remote IP address: See dr2

Node: dr2
HA status: Unknown

Node: dr3
HA status: Unknown
[ibmuser@dr1 ~]$
```

3. Switch to DR site node **dr2** and issue the command to display the status of QMHADR and you see that it is now running on dr2.

```
sudo rdqmstatus -m QMHADR
```

```
ibmuser@dr2:~/ssh
File Edit View Search Terminal Help
[ibmuser@dr2 .ssh]$ sudo rdqmstatus -m QMHADR
Node: dr2
Queue manager status: Running
CPU: 0.00%
Memory: 181MB
Queue manager file system: 59MB used, 2.9GB allocated [2%]
HA role: Primary
HA status: Mixed
HA control: Enabled
HA current location: This node
HA preferred location: dr1
HA floating IP interface: ens36
HA floating IP address: 10.0.2.20
DR role: Primary
DR status: Normal
DR port: 1414
DR local IP address: 10.0.2.15
DR remote IP address list: 10.0.2.1,10.0.2.2,10.0.2.3
DR current remote IP address: 10.0.2.1

Node: dr1
HA status: Remote node in standby
HA out of sync data: 676KB

Node: dr3
HA status: Normal
[ibmuser@dr2 .ssh]$
```

4. Return to **dr1**, issue the command to resume the replicated data node in the cluster.

```
sudo rdqmadm -r
```

Quickly run the status command again. QMHADR will initially run in a secondary role on this node. If you aren't quick enough, you may not catch this transitory state.

```
ibmuser@dr1:~ -  
File Edit View Search Terminal Help  
[ibmuser@dr1 ~]$ sudo rdqmadm -r  
The replicated data node 'dr1' has been resumed.  
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMHADR  
Node: dr1  
Queue manager status: Running elsewhere  
HA role: Secondary  
HA status: Normal  
HA control: Enabled  
HA current location: dr2  
HA preferred location: This node  
HA floating IP interface: ens36  
HA floating IP address: 10.0.2.20  
DR role: Primary  
DR status: See dr2  
DR port: 1414  
DR local IP address: 10.0.2.14  
DR remote IP address list: 10.0.2.1,10.0.2.2,10.0.2.3  
DR current remote IP address: See dr2  
  
Node: dr2  
HA status: Normal  
  
Node: dr3  
HA status: Normal  
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMHADR  
Node: dr1  
Queue manager status: Status not available  
HA role: Secondary  
HA status: Normal  
HA control: Enabled  
HA current location: Unknown  
HA preferred location: This node  
HA floating IP interface: ens36  
HA floating IP address: 10.0.2.20  
DR role: Primary  
DR status: Unknown  
DR port: 1414  
DR local IP address: 10.0.2.14  
DR remote IP address list: 10.0.2.1,10.0.2.2,10.0.2.3  
DR current remote IP address: Unknown  
  
Node: dr2  
HA status: Normal
```

5. After the node has fully resumed, QMHADR will run in a primary role on this node, as it was prior to being suspending. Issue the status command again to confirm that this has indeed happened.

```
sudo rdqmstatus -m QMHADR
```

```
ibmuser@dr1:~ -  
File Edit View Search Terminal Help  
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMHADR  
Node: dr1  
Queue manager status: Running  
CPU: 3.98%  
Memory: 184MB  
Queue manager file system: 59MB used, 2.9GB allocated [2%]  
HA role: Primary  
HA status: Normal  
HA control: Enabled  
HA current location: This node  
HA preferred location: This node  
HA floating IP interface: ens36  
HA floating IP address: 10.0.2.20  
DR role: Primary  
DR status: Normal  
DR port: 1414  
DR local IP address: 10.0.2.14  
DR remote IP address list: 10.0.2.1,10.0.2.2,10.0.2.3  
DR current remote IP address: 10.0.2.1  
  
Node: dr2  
HA status: Normal  
  
Node: dr3  
HA status: Normal  
[ibmuser@dr1 ~]$
```

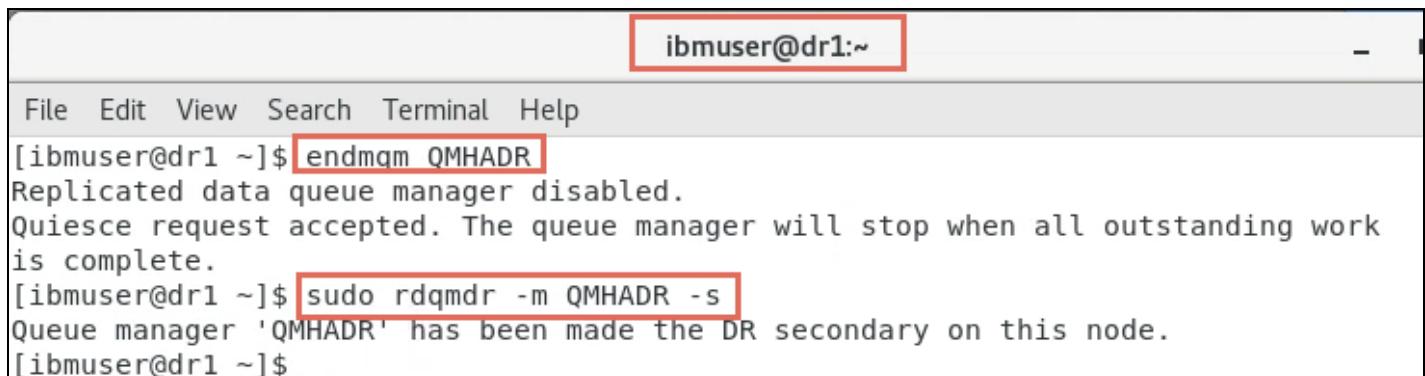
### Move RDQM QMHADR to Main site HA cluster

1. Stop the queue manager on DR site node **dr1**.

```
endmqm QMHADR
```

## 2. Make DR site DR secondary.

```
sudo rdqmldr -m QMHADR -s
```



A terminal window titled "ibmuser@dr1:~". The window contains the following text:

```
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ endmqm QMHADR
Replicated data queue manager disabled.
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
[ibmuser@dr1 ~]$ sudo rdqmldr -m QMHADR -s
Queue manager 'QMHADR' has been made the DR secondary on this node.
[ibmuser@dr1 ~]$
```

## 3. Switch to Main site node **rdqm1**. Make Main site DR Primary.

```
sudo rdqmldr -m QMHADR -p
```

## 4. Start the queue manager on Main site node **rdqm1**.

```
strmqm QMHADR
```

The screenshot shows a terminal window titled "ibmuser@rdqm1:~". The window contains the following text:

```
[ibmuser@rdqm1 ~]$ sudo rdqmldr -m QMHADR -p
Queue manager 'QMHADR' has been made the DR primary on this node.
[ibmuser@rdqm1 ~]$ strmqm QMHADR
IBM MQ queue manager 'QMHADR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMHADR' during the log replay phase
.
Log replay for queue manager 'QMHADR' complete.
Transaction manager state recovered for queue manager 'QMHADR'.
Replicated data queue manager enabled.
IBM MQ queue manager 'QMHADR' started using V9.1.5.0.
[ibmuser@rdqm1 ~]$
```

The commands `sudo rdqmldr -m QMHADR -p` and `strmqm QMHADR` are highlighted with red boxes.

## Testing DR / HA with sample programs

Some High Availability sample programs are provided with MQ, which are a good visible demonstration for testing failovers. You will use these for testing:

- **amqsphac** - puts a sequence of messages to a queue with a two second delay between each message and displays events sent to its event handler. This will run on **rdqm3**.
- **amqsmhac** - copies messages from one queue to another with a default wait interval of 15 minutes after the last message that is received before the program finishes. This will run on **rdqm2**.
- **amqsghac** - gets messages from a queue and displays events sent to its event handler. This will run on **rdqm1**.

## Start the HA sample programs

The easiest way to configure access to the queue manager from the sample programs is to use the MQSERVER environment variable. Again, as there are 3 possible nodes where our queue manager could run, each needs to be specified, along with the listener port for the queue manager. Now that you have a floating IP address associated with QMHADR, you can change the MQSERVER environment variable to CHANNEL1/TCP /10.0.1.0(1420).

1. On **rdqm1**, open a new terminal window and enter:

```
export MQSERVER='CHANNEL1/TCP/10.0.1.20(1420)'
```

2. Change to the **/opt/mqm/samp/bin** directory, and run the command: **amqsghac TARGET QMHADR**

```
cd /opt/mqm/samp/bin  
./amqsghac TARGET QMHADR
```

The screenshot shows a terminal window with the following content:

```
ibmuser@rdqm1:/opt/mqm/samp/bin  
File Edit View Search Terminal Help  
[ibmuser@rdqm1 ~]$ export MQSERVER='CHANNEL1/TCP/10.0.1.20(1420)'  
[ibmuser@rdqm1 ~]$ cd /opt/mqm/samp/bin  
[ibmuser@rdqm1 bin]$ ./amqsghac TARGET QMHADR  
Sample AMQSGHAC start
```

The command `export MQSERVER='CHANNEL1/TCP/10.0.1.20(1420)'` and the entire line `./amqsghac TARGET QMHADR` are highlighted with a red box.

Later, this will display the messages generated by amqsphac on rdqm3.

**Leave this command to run!**

3. Now switch to **rdqm2**. Open a new terminal window and enter:

```
export MQSERVER='CHANNEL1/TCP/10.0.1.20(1420)'
```

4. Change directory to **/opt/mqm/samp/bin** and run the command: **amqsmhac -s SOURCE -t TARGET -m QMHADR**

```
cd /opt/mqm/samp/bin  
.amqsphac -s SOURCE -t TARGET -m QMHADR
```

The screenshot shows a terminal window with the following session:

```
ibmuser@rdqm2:/opt/mqm/samp/bin  
File Edit View Search Terminal Help  
[ibmuser@rdqm2 ~]$ export MQSERVER='CHANNEL1/TCP/10.0.1.20(1420)'  
[ibmuser@rdqm2 ~]$ cd /opt/mqm/samp/bin  
[ibmuser@rdqm2 bin]$ ./amqsphac -s SOURCE -t TARGET -m QMHADR  
Sample AMQSPHAC start
```

The last three commands are highlighted with a red box.

**Leave this command to run!**

**Note:**

While creating this lab it was noticed that copying commands from the PDF lab guide or the online lab guide sometimes added an extra dash to the hyphen before the parameters in the command. This causes the command to fail. If this occurs, delete the hyphen and retype it then rerun the command.

5. Now switch to **rdqm3**. As before open a new terminal window. As the user ibmuser enter:

```
export MQSERVER='CHANNEL1/TCP/10.0.1.20(1420)'
```

6. Change directory to **/opt/mqm/samp/bin** and run the command: **amqsphac SOURCE QMHADR**

```
cd /opt/mqm/samp/bin  
.amqsphac SOURCE QMHADR
```

```
ibmuser@rdqm3:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@rdqm3 bin]$ export MQSERVER='CHANNEL1/TCP/10.0.1.20(1420)'
[ibmuser@rdqm3 bin]$ cd /opt/mqm/samp/bin/
[ibmuser@rdqm3 bin]$ ./amqsphac SOURCE QMHADR
Sample AMQSPHAC start
target queue is SOURCE
message <Message 1>
message <Message 2>
message <Message 3>
message <Message 4>
message <Message 5>
```

**Leave this command to run!**

7. Confirm that these messages are also being displayed on **rdqm1**.

```
ibmuser@rdqm1:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@rdqm1 ~]$ export MQSERVER='CHANNEL1/TCP/10.0.1.20(1420)'
[ibmuser@rdqm1 ~]$ cd /opt/mqm/samp/bin
[ibmuser@rdqm1 bin]$ ./amqsghac TARGET QMHADR
Sample AMQSGHAC start
message <Message 1>
message <Message 2>
message <Message 3>
message <Message 4>
message <Message 5>
```

**Note:** At this stage, the queue manager is running on the primary HA node (**rdqm1**) and has DR Role Primary and each sample program is able to communicate with it, using the floating IP address in the MQSERVER environment variable:

**CHANNEL1/TCP/10.0.1.20(1420)**

## Move the RDQM within local HA cluster

You will now use the approach of controlling where the RDQM runs by suspending HA on node **rdqm1**.

1. Still on **rdqm1**. In a new terminal window, run the following command as ibmuser:

```
sudo rdqmadm -s
```

ibmuser@rdqm1:~

File Edit View Search Terminal Help
[ibmuser@rdqm1 ~]\$ sudo rdqmadm -s
The replicated data node 'rdqm1' has been suspended.

**Note:**

You suspended HA on rdqm1. Since you did not specify the preferred node Pacemaker can move the queue manager to either standby node. Check both rdqm2 and rdqm3 to see which node QMHADR is running on. In either case the applications will reconnect.

- Check if the queue manager is running on **rdqm2**, by issuing command:

```
sudo rdqmstatus -m QMHADR
```

The output from the amqsmhac command, running in another window, should now be like this:

ibmuser@rdqm2:/opt/mqm/samp/bin	ibmuser@rdqm2:~
File Edit View Search Terminal Help	File Edit View Search Terminal Help
[ibmuser@rdqm2 ~]\$ export MQSERVER='CHANNEL1/TCP/10.0.1.20(1420)' [ibmuser@rdqm2 ~]\$ cd /opt/mqm/samp/bin [ibmuser@rdqm2 bin]\$ ./amqsmhac -s SOURCE -t TARGET -m QMHADR Sample AMQSMHAC start	[ibmuser@rdqm2 ~]\$ sudo rdqmstatus -m QMHADR Node: rdqm2 Queue manager status: Running elsewhere HA role: Secondary HA status: Mixed HA control: Enabled HA current location: rdqm3 HA preferred location: rdqm1
07:54:53 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 222ms) 07:54:53 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1041ms) 07:54:57 : EVENT : Connection Reconnected	

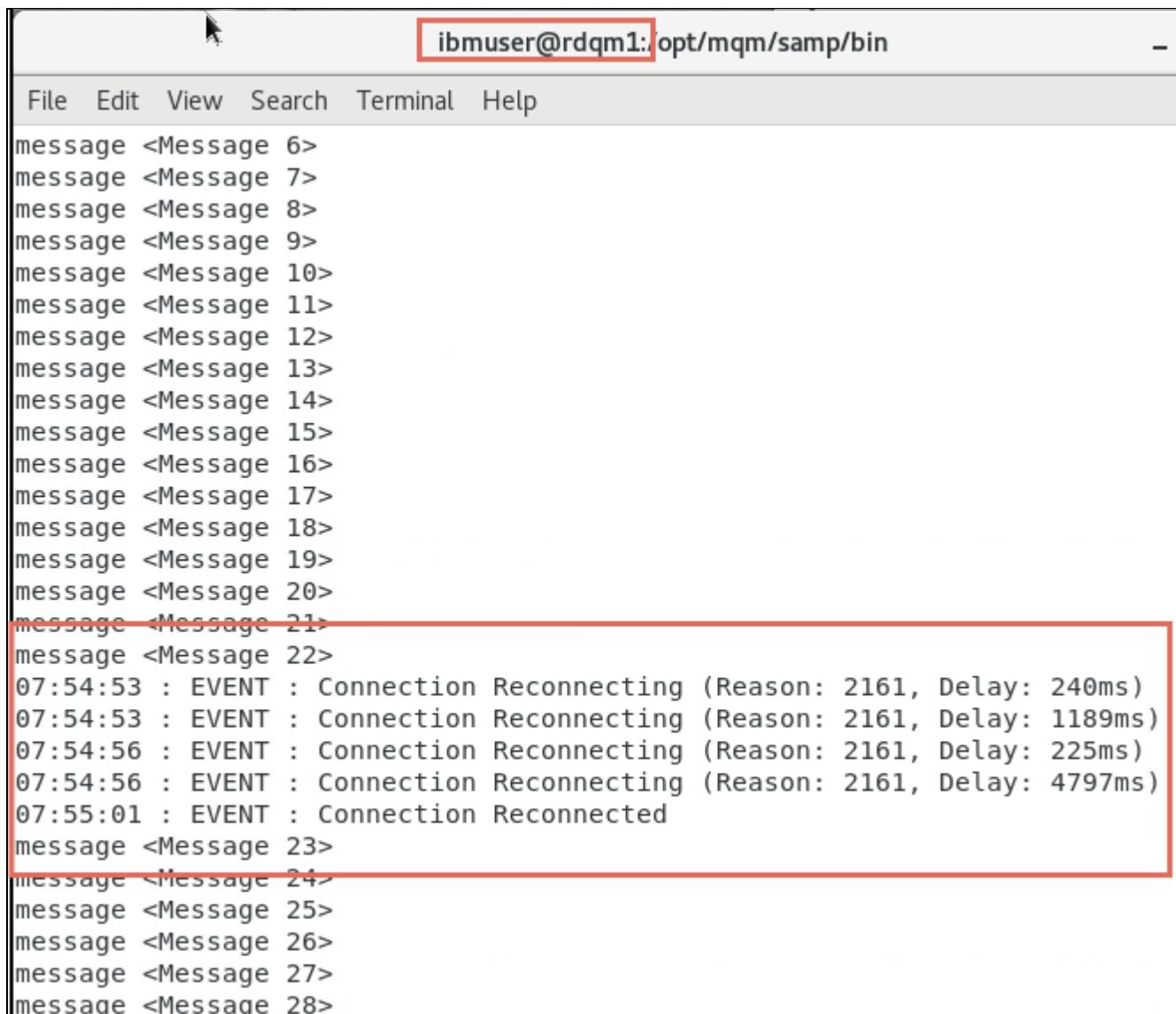
- Now switch to **rdqm3**. In a terminal window, check to see if QMHADR is running here.

The output from the amqsphac command, running on rdqm3, should similarly show messages continuing to be sent without loss after reconnecting:

ibmuser@rdqm3:/opt/mqm/samp/bin	ibmuser@rdqm3:-
File Edit View Search Terminal Help 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> message <Message 17> message <Message 18> message <Message 19> message <Message 20> message <Message 21> message <Message 22> 07:54:53 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 245ms) 07:54:53 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1059ms) 07:54:57 : EVENT : Connection Reconnected  message <Message 23> message <Message 24> message <Message 25> message <Message 26> message <Message 27> message <Message 28>	File Edit View Search Terminal Help [ibmuser@rdqm3 ~]\$ sudo rdqmstatus -m QMHADR Node: rdqm3 Queue manager status: Running CPU: 0.00% Memory: 181MB Queue manager file system: 58MB used, 2.9GB allocated [2%] HA role: Primary HA status: Mixed HA control: Enabled HA current location: This node HA preferred location: rdqm1 HA floating IP interface: ens35 HA floating IP address: 10.0.1.20 DR role: Primary DR status: Normal DR port: 7001 DR local IP address: 10.0.2.3 DR remote IP address list: 10.0.2.14,10.0.2.15,10.0.2.16 DR current remote IP address: 10.0.2.14  Node: rdqm1 HA status: Remote node in standby HA out of sync data: 624KB  Node: rdqm2 HA status: Normal

#### 4. Now switch to **rdqm1**.

Messages should continue to be received without loss, by amqsghac, after it connects with the queue manager at the new location:



The screenshot shows a terminal window with the title bar "ibmuser@rdqm1: opt/mqm/samp/bin". The menu bar includes File, Edit, View, Search, Terminal, and Help. The main area displays a series of messages and event logs:

```
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>
message <Message 17>
message <Message 18>
message <Message 19>
message <Message 20>
message <Message 21>
message <Message 22>
07:54:53 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 240ms)
07:54:53 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1189ms)
07:54:56 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 225ms)
07:54:56 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 4797ms)
07:55:01 : EVENT : Connection Reconnected
message <Message 23>
message <Message 24>
message <Message 25>
message <Message 26>
message <Message 27>
message <Message 28>
```

**Note:** Now the queue manager is running on rdqm2 or rdqm3, but each sample program is still able to communicate with it by using the floating IP address specified in the MQSERVER environment variable:

**CHANNEL1/TCP/10.0.1.20(1420)**

5. Still on **rdqm1** resume HA on this node with command:

```
sudo rdqmadm -r
```

```
ibmuser@rdqm1:~  
File Edit View Search Terminal Help  
[ibmuser@rdqm1 ~]$ sudo rdqmadm -r  
The replicated data node 'rdqm1' has been resumed.
```

6. Ensure that QMHADR is now running on **rdqm1**.

```
sudo rdqmstatus -m QMHADR
```

```
ibmuser@rdqm1:~  
File Edit View Search Terminal Help  
[ibmuser@rdqm1 ~]$ sudo rdqmstatus -m QMHADR  
Node: rdqm1  
Queue manager status: Running  
CPU: 0.00%  
Memory: 181MB  
Queue manager file system: 59MB used, 2.9GB allocated [2%]  
HA role: Primary  
HA status: Normal  
HA control: Enabled  
HA current location: This node  
HA preferred location: This node  
HA floating IP interface: ens35  
HA floating IP address: 10.0.1.20  
DR role: Primary  
DR status: Partitioned  
DR port: 7001  
DR local IP address: 10.0.2.1  
DR remote IP address list: 10.0.2.14,10.0.2.15,10.0.2.16  
DR current remote IP address: Unknown  
DR out of sync data: 45072KB  
  
Node: rdqm2  
HA status: Normal  
  
Node: rdqm3  
HA status: Normal  
[ibmuser@rdqm1 ~]$ █
```

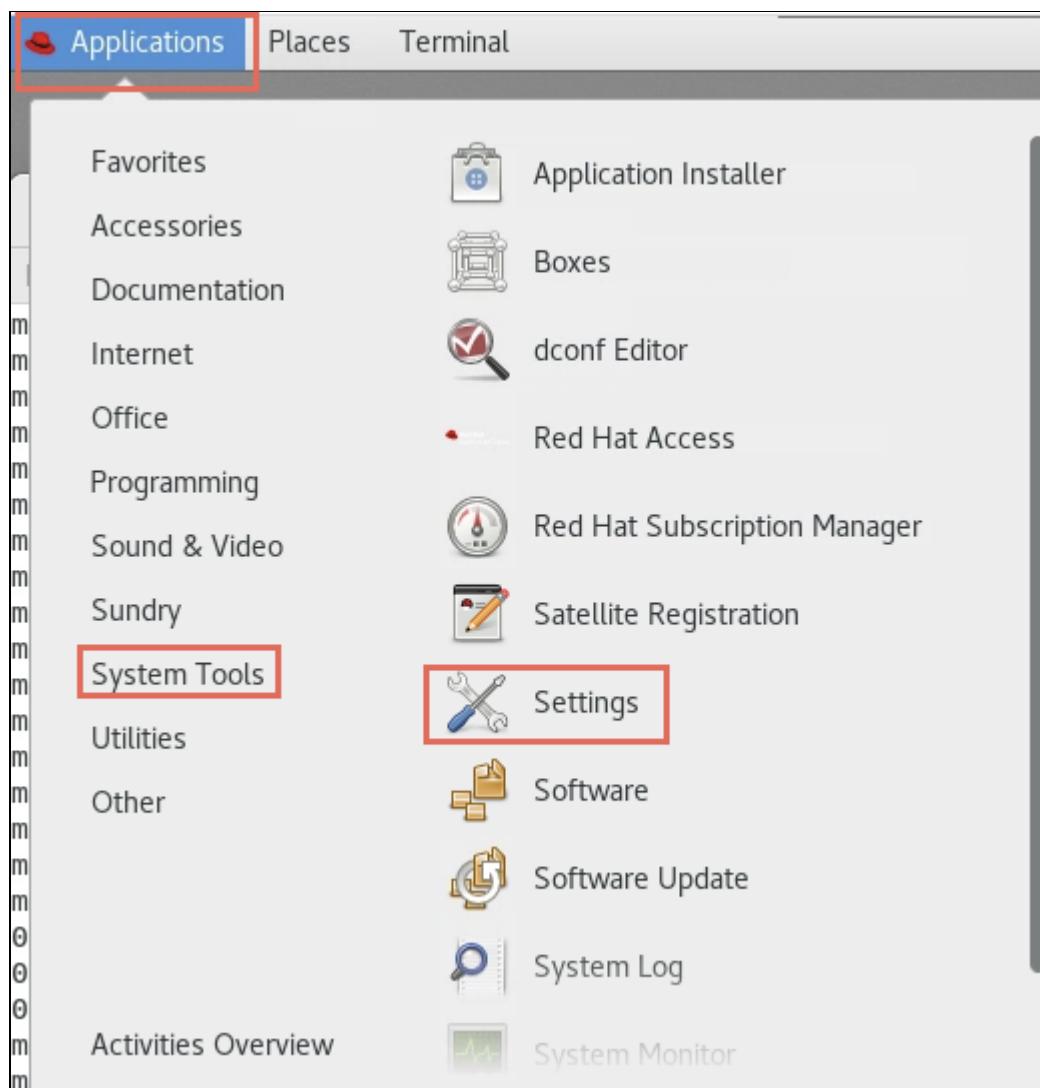
7. Leave the sample programs running.

## Move the RDQM to DR - HA cluster

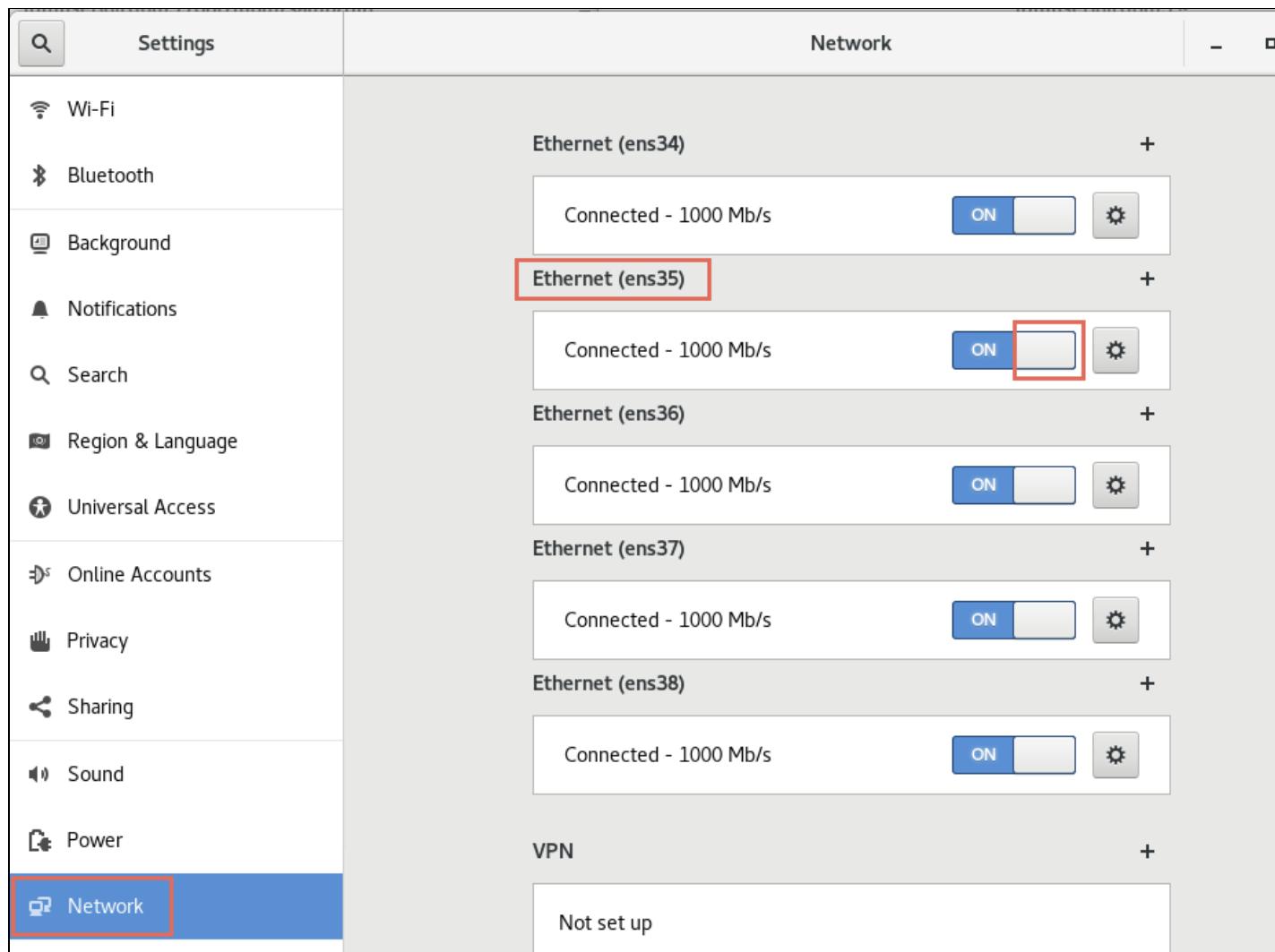
The floating address 10.0.1.20 is using virtual adapter ens35, where currently you also have a fixed IP address configured on each virtual machine. You will disable ens35 on each node in the local main HA cluster to simulate a failover to the DR HA cluster.

Since the RDQM QMHADR is currently running on **rdqm1**, you will start by disabling the adapter on the standby nodes, so the RDQM cannot failover to them and must failover to the remote HA cluster.

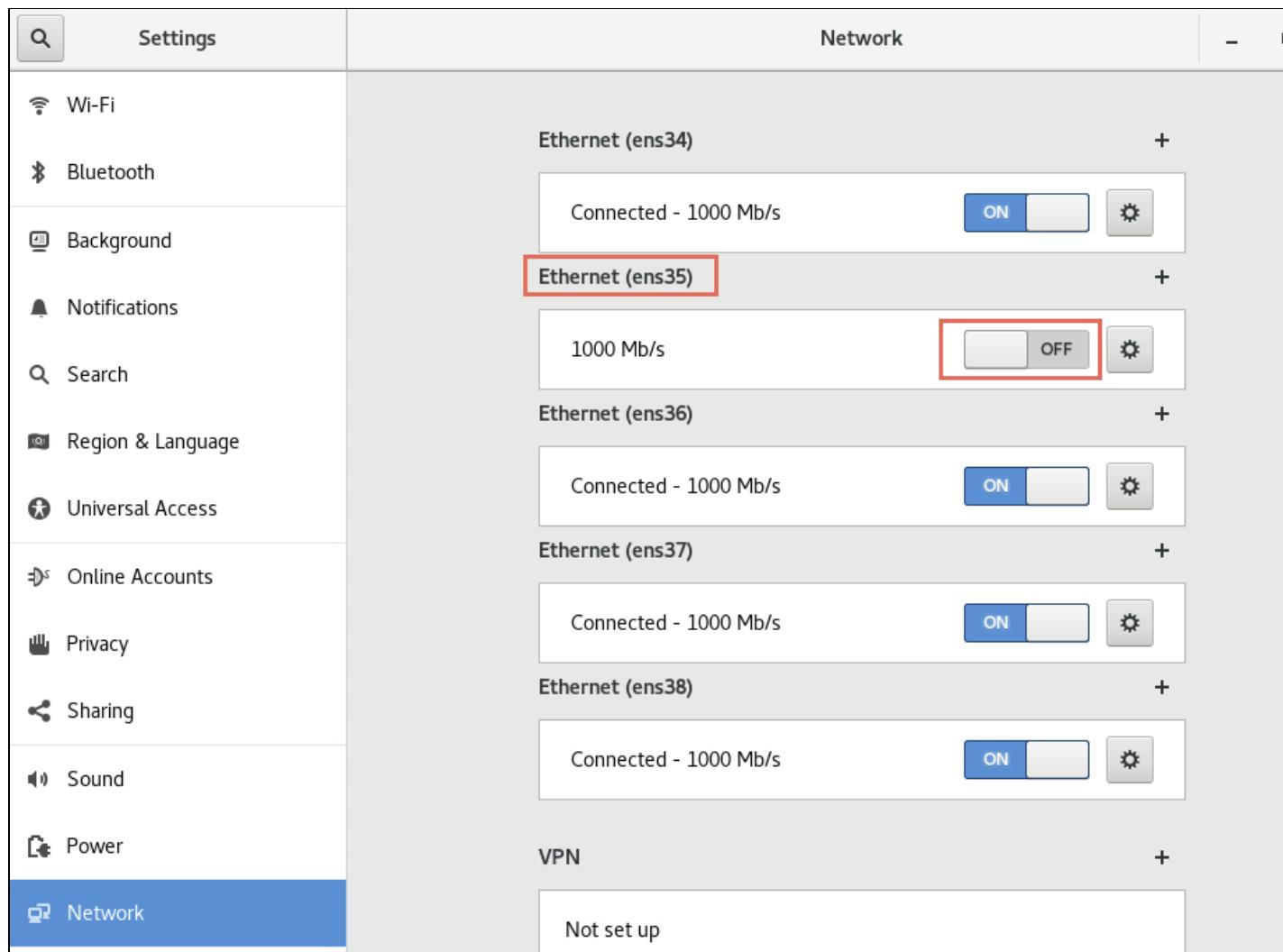
1. On **rdqm3**, open the network settings: *Applications > System Tools > Settings*.



2. Select Network. Click the blank switch next to ON for *Ethernet (ens35)*.



3. The adapter now shows that it is *OFF*.



Leave the Network settings open as you will use it again later.

4. Turn off the adapter **ens35** on **rdqm2** just as you did on **rdqm3**.

Leave the Network settings open as you will use it again later.

5. Using the same procedure, finally turn off the adapter **ens35** **rdqm1** to cause a DR situation.

Leave the Network settings open as you will use it again later.

6. Switch to **dr1**. Make this node the primary node with the following command.

```
sudo rdqmldr -m QMHADR -p
```

7. Start the RDQM:

```
strmqm QMHADR
```

8. Display the status of QMHADR.

```
sudo rdqmstatus -m QMHADR
```

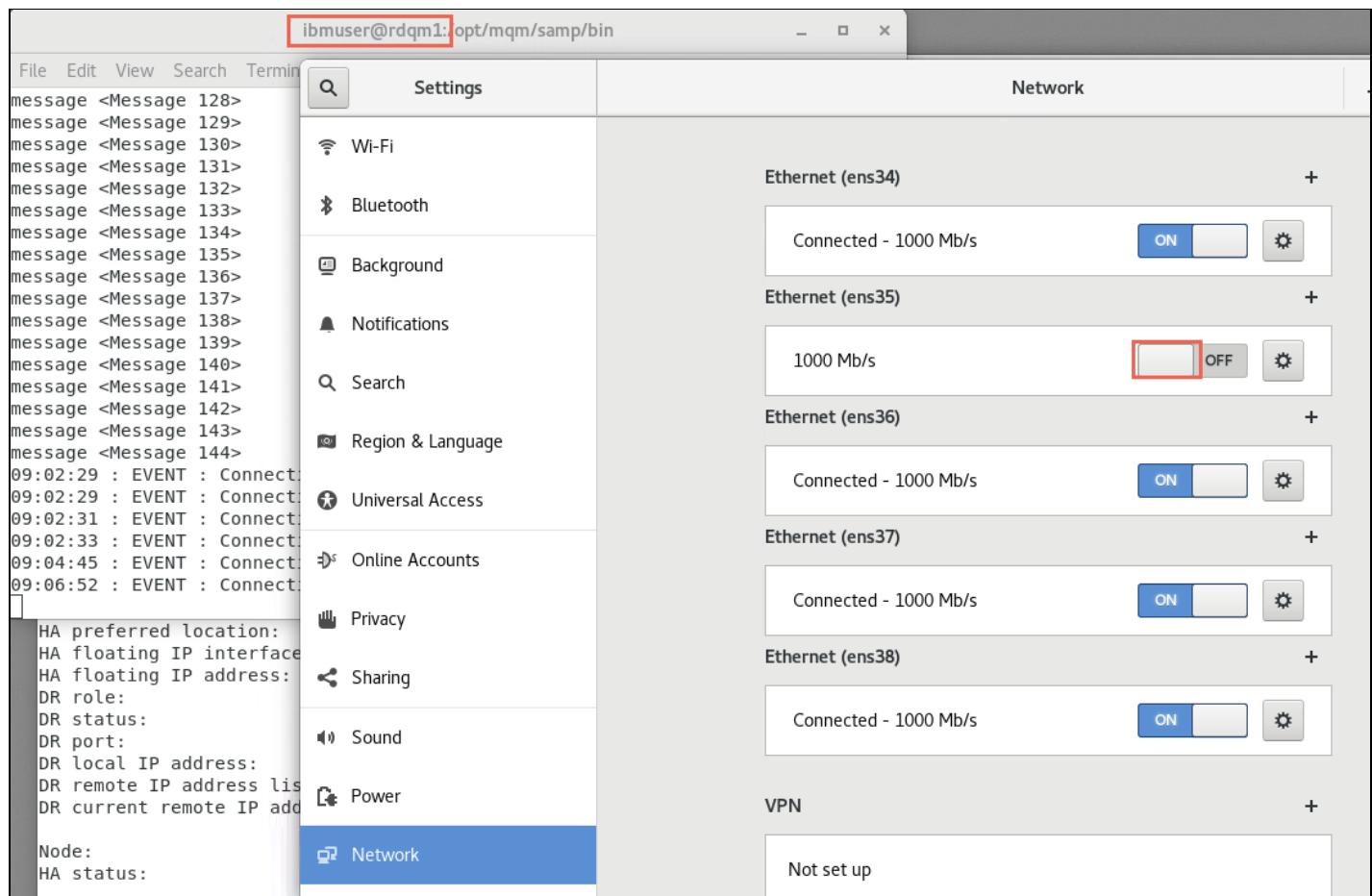
The screenshot shows a terminal window titled "ibmuser@dr1:~". The user has run several commands to manage a queue manager named 'QMADR'. The first command, `sudo rdqmldr -m QMADR -p`, makes the queue manager primary. The second command, `strmqm QMADR`, starts the queue manager. The third command, `sudo rdqmstatus -m QMADR`, displays the status of the queue manager across three nodes: dr1, dr2, and dr3. The status output is as follows:

Category	Node	Status
HA role	dr1	Primary
HA status	dr1	Normal
HA control	dr1	Enabled
HA current location	dr1	This node
HA preferred location	dr1	This node
HA floating IP interface	dr1	ens35
HA floating IP address	dr1	10.0.1.20
DR role	dr1	Primary
DR status	dr1	Normal
DR port	dr1	7001
DR local IP address	dr1	10.0.2.14
DR remote IP address list	dr1	10.0.2.1,10.0.2.2,10.0.2.3
DR current remote IP address	dr1	10.0.2.1
Node	dr2	
HA status	dr2	Normal
Node	dr3	
HA status	dr3	Normal

At the bottom, the user runs `sudo rdqmadm -s` to save the changes.

You see the queue manager is now running on dr1 in the DR HA cluster. This node is primary for both HA and DR. Notice the floating IP is also listed.

9. In our simulation, you will need to re-enable the network adapters. Return to the nodes in the main HA cluster. Start with **rdqm1**. You should have left the network settings window open. Turn the network adapter for ens35 back on by clicking the ON switch.



Repeat this on **rdqm2** and **rdqm3**, re-enabling the `ens35` network adapter on those nodes.

10. You know that QMHADR has failed over to the DR HA cluster. So check that the `amqsphac` client to make sure it has reconnected and continues to send messages.

```
ibmuser@rdqm3:/opt/mqm/samp/bin
File Edit View Search Terminal Help
message <Message 36>
message <Message 37>
message <Message 38>
message <Message 39>
08:01:32 : EVENT : Connection Reconnecting (Reason: 2009, Delay: 96ms)
08:01:33 : EVENT : Connection Reconnected
message <Message 40>
```

11. Similarly check **rdqm2** to see that `amqsmhac` client has reconnected and is forwarding messages.

```
ibmuser@rdqm2:/opt/mqm/samp/bin
File Edit View Search Terminal Help
08:01:28 : EVENT : Connection Reconnecting (Reason: 2009, Delay: 113ms)
08:01:28 : EVENT : Connection Reconnecting (Reason: 2009, Delay: 1345ms)
08:01:33 : EVENT : Connection Reconnected
```

12. And lastly check on **rdqm1** that *amqsghac* client is getting the messages.

```
ibmuser@rdqm1:/opt/mqm/samp/bin
File Edit View Search Terminal Help
message <Message 38>
message <Message 39>
08:01:32 : EVENT : Connection Reconnecting (Reason: 2009, Delay: 18ms)
08:01:32 : EVENT : Connection Reconnected
message <Message 42>
message <Message 43>
message <Message 44>
```

Main site has recovered and the RDQM can be moved back

1. On **dr1**, stop the queue manager.

```
endmqm QMHADR
```

2. Now you need to ensure this site's HA cluster becomes the DR secondary. To ensure this happens, you need to make each node secondary/secondary. Switch to **dr3** and issue the rdqmdr command to make this node DR secondary.

```
sudo rdqmdr -m QMHADR -s
```

```
ibmuser@dr3:~  
File Edit View Search Terminal Help  
[ibmuser@dr3 ~]$ sudo rdqmldr -m QMHADR -s  
You should ensure this command is also run on nodes 'dr1' and 'dr2'.  
Queue manager 'QMHADR' has been made the DR secondary on this node.  
[ibmuser@dr3 ~]$ 
```

The command response tells you to run the command in the other nodes within the HA cluster.

3. Switch to **dr2** and issue the rdqmldr command again.

```
sudo rdqmldr -m QMHADR -s
```

```
ibmuser@dr2:~  
File Edit View Search Terminal Help  
[ibmuser@dr2 ~]$ sudo rdqmldr -m QMHADR -s  
You should ensure this command is also run on nodes 'dr1' and 'dr3'.  
Queue manager 'QMHADR' has been made the DR secondary on this node.  
[ibmuser@dr2 ~]$ 
```

4. And now make **dr1** secondary DR.

```
sudo rdqmldr -m QMHADR -s
```

```
ibmuser@dr1:/var/mqm  
File Edit View Search Terminal Help  
[ibmuser@dr1 mqm]$ endmqm QMHADR  
Replicated data queue manager disabled.  
Quiesce request accepted. The queue manager will stop when all outstanding work  
is complete.  
[ibmuser@dr1 mqm]$ sudo rdqmldr -m QMHADR -s  
Queue manager 'QMHADR' has been made the DR secondary on this node.  
[ibmuser@dr1 mqm]$ 
```

5. Once all of the adapters have been turned back on and when **dr1** became secondary, **rdqm1** automatically becomes primary again and the queue manager starts back up on **rdqm1**. Run the status command on **rdqm1**.

```
sudo rdqmstatus -m QMHADR
```

6. Switch to **rdqm3**. Make sure the test application has reconnected and is sending messages.
7. Switch to **rdmq2** and check for reconnection of the test application. Then switch to **rdqm3** and make sure the test application is receiving messages.
8. When completed testing, stop (with ctrl-C) the HA sample programs that are currently running on each of the nodes.

## CONGRATULATIONS!

You have completed this hands-on lab.

You have created replicated data queue managers to provide high availability for IBM MQ, and tested failing over.

## Continue to experiment with RDQM HA and DR

If you have additional time continue to experiment. You now know the commands for HA and DR. You know how to move the RDQM and cause it to failover to another node within the HA cluster and how to cause a DR scenario.

Try using the secondary HA cluster as the main and failover to the original main HA cluster. Use your imagination.

