

Replicated Data Queue Managers (RDQM) for High Availability (9.1.5)

Messaging Administrator

Summary: Replicated Data Queue Managers (RDQM) for High Availability

Introducing IBM MQ Replicated Data Queue Manager

IBM MQ High Availability Options

For many years there has been two well established options for IBM MQ single site high availability:

- **Multi-Instance Queue Managers:** An out of the box technology that requires a shared disk (such as NFS) for storing the Queue Manager data, and two servers in an active / standby configuration. If the active instance was to fail, the lock on the shared disk would be released, and the passive instance promoted to the active instance.
- **Operating System High Availability technologies:** IBM MQ explicitly supports a number of well-established operating system high availability technologies, such as Microsoft Cluster Service, PowerHA for AIX, Veritas Cluster Server, HP Serviceguard, or a Red Hat Enterprise Linux cluster with Red Hat Cluster Suite.

Each of the above have technical considerations and dependencies; for instance, the multi-instance queue manager is dependent on a shared disk, which must be redundant to provide effective high availability; while the operating system approach requires deep knowledge of the operating system, and in certain cases additional license entitlement for the operating system.

IBM MQ V9.0.4 Continuous Delivery Release provided a new option to complement the existing two, called MQ Replicated Data Queue Manager (RDQM). This technology is based on the MQ Appliance high availability capabilities, and is available on the Linux operating system. In the simplest of situations, it will include three standard RedHat Linux servers, one being active handling requests, with the other two replicating the data and waiting to become active in a similar logical manner to the standby instance within a Multi-Instance Queue Manager. When comparing to multi-instance, there are three key differences:

- **No shared disk:** Each individual server includes a complete replica of the Queue Manager data, removing the need for a shared disk, which simplifies the configuration, and potentially improves the performance of the overall solution.
- **Quorum based promotion:** Due to each server including a complete replica, in the case of a network failure between the servers, all could be promoted to be active, and a condition called “split brain” could occur. To avoid this from happening, three servers are part of the RDQM HA group, and a server will only continue to be active or promoted to the active instance if a quorum of the servers within the RDQM HA group are contactable.
- **Floating IP:** With Multi-Instance Queue Managers, two servers, with separate IP addresses could be hosting and running the active instance of the Queue Manager. This means that the client needs to be explicitly aware of the two IP addresses. With the new RDQM, a floating IP address is used, and associated with the server that is running the active queue manager. This means that clients only need to be aware of the floating IP address, simplifying the logic on the client.

Although the above are all drivers of why to consider the new capability, there are some points to consider when evaluating if this new HA model is suitable for a client:

- **Restricted to a single Data Center:** The replication between instances within a RDQM HA group is synchronous, and therefore if the servers are not co-located within the same data center, this can affect the performance of the solution. Also, since a floating IP address is used, this normally is limited to a single data centre.
- **Distributed Writes:** As mentioned above, the writing is completed using a synchronous write, and therefore involves writing to multiple disks across multiple servers. The performance characteristics of this approach, compared to the other options, need to be evaluated.
- **Dedicated Logical Volume Group:** Each RedHat Linux server requires a dedicated logical volume for the RDQM data. This normally means a separate disk associated with the VM or bare metal machine dedicated to the RDQM data. If the server is being specially provisioned for IBM MQ, this is unlikely to be a major issue.
- **Split Brain:** The quorum based approach to electing primaries greatly reduces the chances of a split brain situation occurring, but does not completely remove all edge cases.
- **RedHat Enterprise Linux 7.3, 7.4, 7.5, 7.6, or 7.7:** The RDQM capability is currently supported on RedHat Enterprise Linux 7.3, 7.4, 7.5, 7.6, and 7.7. Depending on the environment, this may or may not be a significant restriction.

To assist with the setup of RDQM, we've created a number of scripts and helper files. The following is based on the process required for a base RedHat Enterprise Linux 7.7 install.

Lab Introduction

This lab provides a demonstration of a new approach to High Availability in MQ on Linux, with the following key

features:

- Use of Distributed Replicated Block Device (DRBD) storage rather than network shared storage
- Use of a cluster resource manager (Pacemaker) to manage a cluster of three nodes
- A new kind of queue manager called a Replicated Data Queue Manager (RDQM):
 - an RDQM is active on only one node at any one time
 - each node can run different active RDQMs
 - each RDQM has a preferred location (node) in normal operation
 - a quorum prevents an RDQM from running on more than one node at the same time
 - a RDQM can have a floating IP address associated with it to simplify configuration of clients and other queue managers that need to communicate with the RDQM

Lab environment

- Three 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:

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

- Network interfaces:

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		10.0.1.10	10.0.1.10	10.0.1.10

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)

Note: Hosts rdqm1, rdqm2, rdqm3 are tied to the Administration IP addresses above.

- 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 MQ 9.1.5 on 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 six virtual machines: rdqm1, rdqm2, rdqm3, dr1, dr2, and dr3 which currently should be in a powered off or paused state.

VM Label	Image Status	Endpoints	Metered RAM	Storage	Licenses
dr1 - Red Hat Enterprise Linu x 7.7 64-bit	Powered off	5	8 GB	200 GB	--
dr2 - Red Hat Enterprise Linu x 7.7 64-bit	Powered off	5	8 GB	200 GB	--
dr3 - Red Hat Enterprise Linu x 7.7 64-bit	Powered off	5	8 GB	200 GB	--
rdqm1 - Red Hat Enterprise Linu x 7.7 64-bit	Powered off	5	8 GB	200 GB	--
rdqm2 - Red Hat Enterprise Linu x 7.7 64-bit	Powered off	5	8 GB	200 GB	--
rdqm3 - Red Hat Enterprise Linu x 7.7 64-bit	Powered off	5	8 GB	200 GB	--

1. dr1, dr2, and dr3 will not be used in this lab, so you can leave them powered off by unchecking their labels.

Region: US-Central

VMs: 6

VM Name	Status	METERED RAM	STORAGE	LICENSE
dr1 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--
dr2 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--
dr3 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--
rdqm1 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--
rdqm2 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--
rdqm3 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--

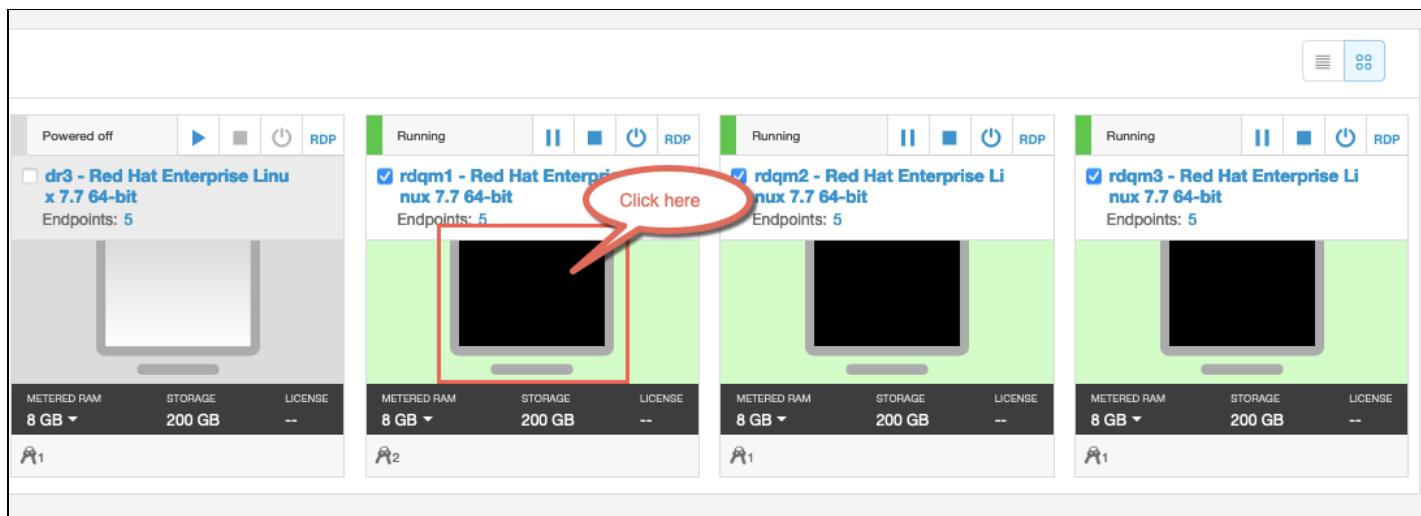
- Click the **run** button to start or resume the VMs.

Region: US-Central

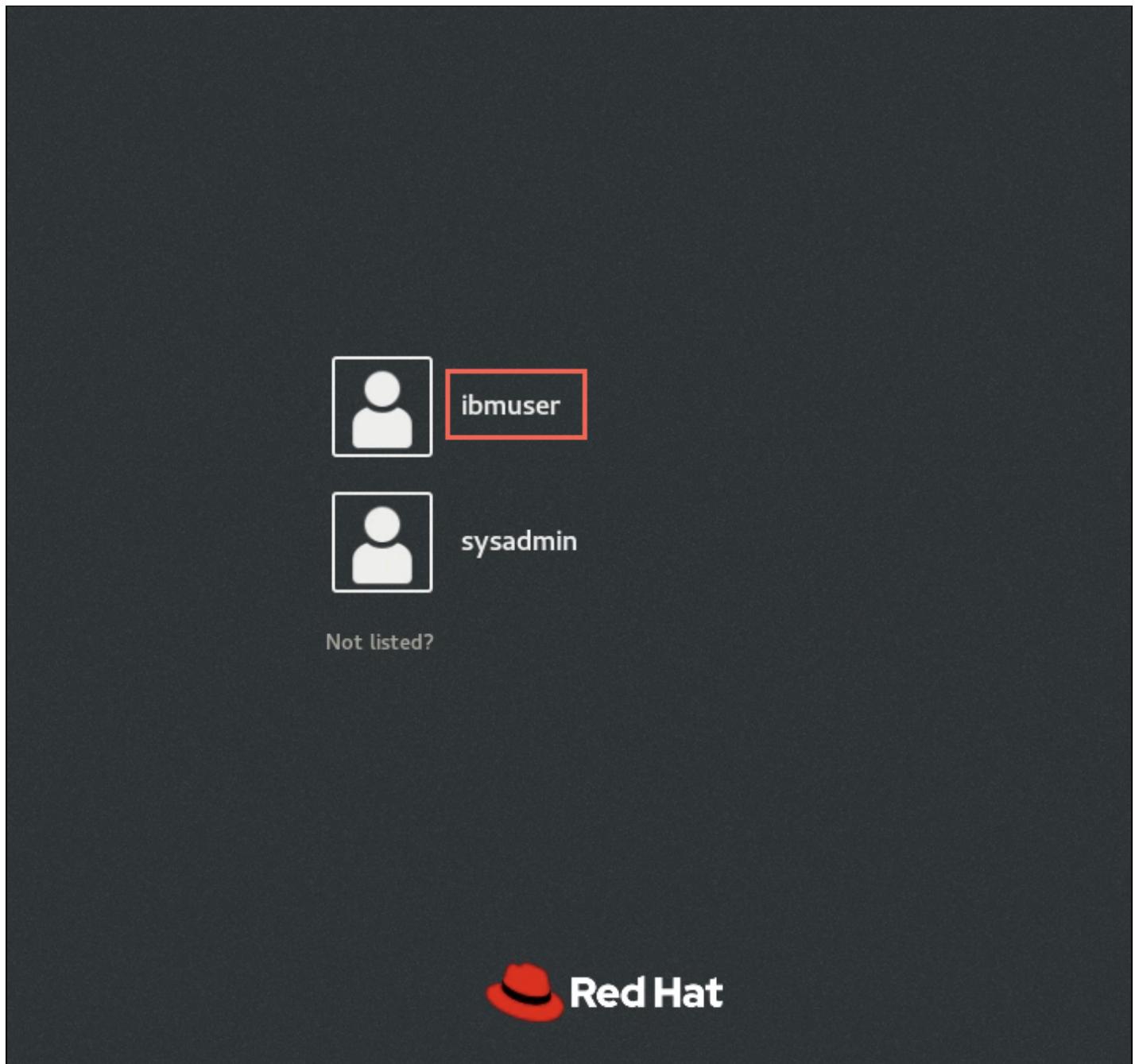
VMs: 6

VM Name	Status	METERED RAM	STORAGE	LICENSE
dr1 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--
dr2 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--
dr3 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--
rdqm1 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--
rdqm2 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--
rdqm3 - Red Hat Enterprise Linux x 7.7 64-bit	Powered off	8 GB	200 GB	--

- Wait for the monitor icons to turn green, approximately three minutes. Click the monitor icon for **rdqm1** which will launch the desktop in another browser tab.



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



Pre-configuration steps

The following steps are necessary for configuring RDQM, and are shown for your reference. 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 DR and DR for the HA group. 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).

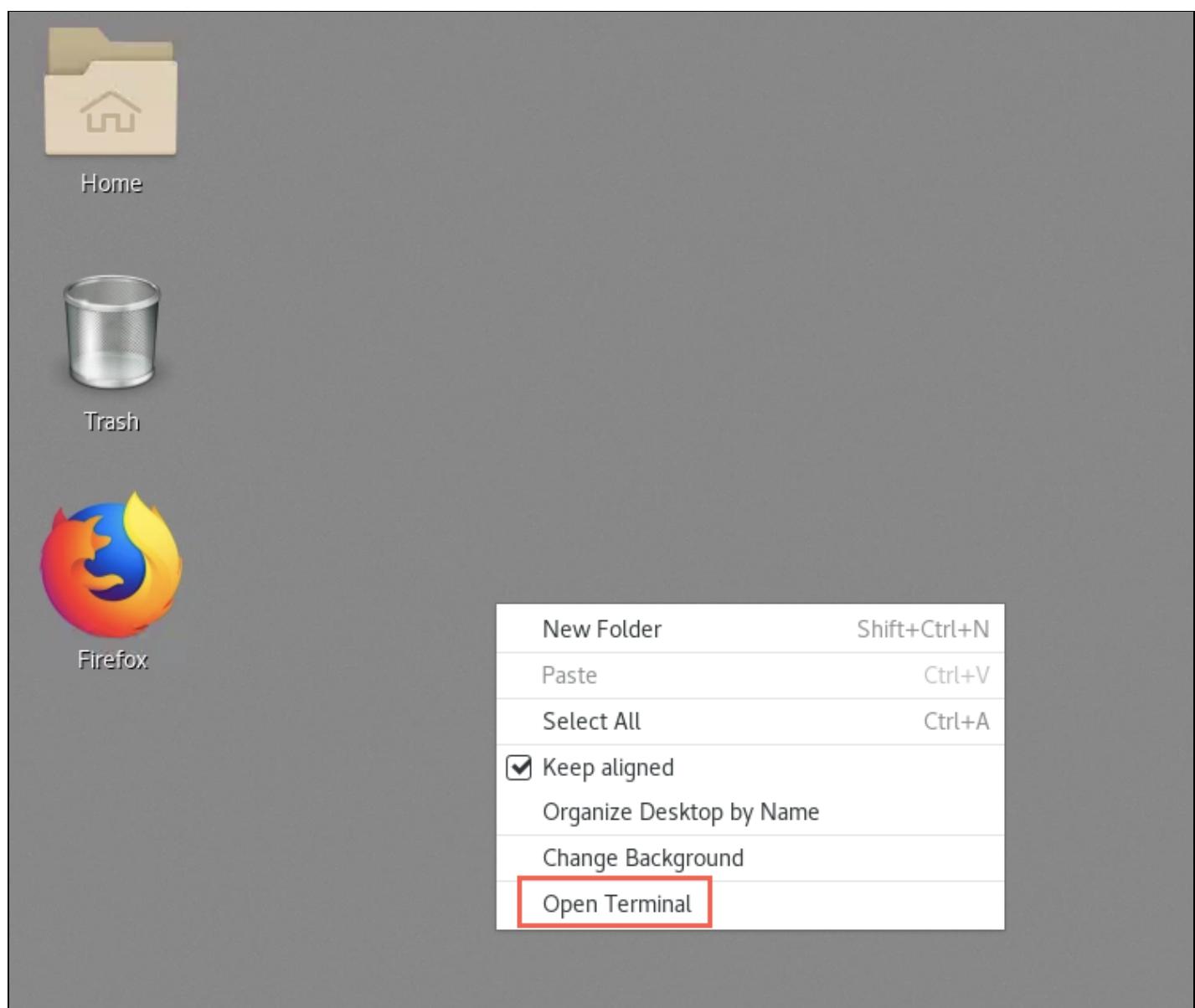
The above steps were completed on each node so at this point you are ready to begin RDQM configuration.

Configure RDQM

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 by right-clicking on the desktop and select Open Terminal.



2. Switch user to root using the following command:

```
su -
```

When prompted, enter root's password *IBMDem0s!*.

3. Change to the /home/ibmuser/mq915/MQServer/Advanced/RDQM/.

```
cd /home/ibmuser/mq915/MQServer/Advanced/RDQM
```

```
root@rdqm1:/home/ibmuser/mq915/MQServer/Advanced/RDQM
File Edit View Search Terminal Help
[ibmuser@rdqm1 ~]$ su -
Password:
Last login: Fri May 29 16:46:25 EDT 2020 on pts/0
[root@rdqm1 ~]# cd /home/ibmuser/mq915/MQServer/Advanced/RDQM
```

4. List the members of the directory. You will see the RPM file for installing RDQM along with the scripts to install and uninstall the RDQM support. Also included is a subdirectory for the RDQM prerequisites.
5. List the members of directory *PreReqs* to see the important prerequisites of RDQM.

```
ls PreReqs
```

```
[root@rdqm1 RDQM]# ls
installRDQMsupport          PreReqs      uninstallRDQMsupport
MQSeriesRDQM-9.1.5-0.x86_64.rpm  repackage
[root@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.

```
[root@rdqm1 RDQM]# ls PreReqs/drbd-9.0/
drbd-bash-completion-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_693-1.x86_64.rpm
drbd-pacemaker-9.12.0-1.el7.x86_64.rpm        kmod-drbd-9.0.22_3.10.0_514.36.5-1.x86_64.rpm    kmod-drbd-9.0.22_3.10.0_693.21.1-1.x86_64.rpm
drbd-udev-9.12.0-1.el7.x86_64.rpm            kmod-drbd-9.0.22_3.10.0_514.6.2-1.x86_64.rpm    kmod-drbd-9.0.22_3.10.0_862-1.x86_64.rpm
drbd-utils-9.12.0-1.el7.x86_64.rpm          kmod-drbd-9.0.22_3.10.0_693.17.1-1.x86_64.rpm  kmod-drbd-9.0.22_3.10.0_957-1.x86_64.rpm
[root@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
[root@rdqm1 RDQM]#
```

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

```
gedit installRDQMsupport
```

```
#!/bin/bash
# <copyright
# notice="lm-source-program"
# pids="5724-H72"
# years="2017,2020"
# crc=""
# Licensed Materials - Property of IBM
#
# 5724-H72
#
# (C) Copyright IBM Corp. 2017, 2020 All Rights Reserved.
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with
# IBM Corp.
# </copyright>

RDQM_DIR=$(dirname "$0")

RDQM_PACKAGES="$RDQM_DIR/MQSeriesRDQM-9.1.5-0.x86_64.rpm"

MQ_DEPENDENCIES="$RDQM_DIR/../../MQSeriesGSKit-9.1.5-0.x86_64.rpm \
$RDQM_DIR/../../MQSeriesServer-9.1.5-0.x86_64.rpm \
$RDQM_DIR/../../MQSeriesRuntime-9.1.5-0.x86_64.rpm"

ADDITIONAL_MQ_PACKAGES="$RDQM_DIR/../../MQSeriesSamples-9.1.5-0.x86_64.rpm \
$RDQM_DIR/../../MQSeriesClient-9.1.5-0.x86_64.rpm"

DRBD_PACKAGES="$RDQM_DIR/PreReqs/drbd-9.0/drbd-bash-completion-9.12.0-1.el7.x86_64.rpm \
$RDQM_DIR/PreReqs/drbd-9.0/drbd-pacemaker-9.12.0-1.el7.x86_64.rpm \
$RDQM_DIR/PreReqs/drbd-9.0/drbd-udev-9.12.0-1.el7.x86_64.rpm \
$RDQM_DIR/PreReqs/drbd-9.0/drbd-utils-9.12.0-1.el7.x86_64.rpm"

PACEMAKER_PACKAGES="$RDQM_DIR/PreReqs/pacemaker-1.1.20/cluster-glue-libs-1.0.12.linbit-2.0+20180815+be86a9f22546.el7.x86_64.rpm \
$RDQM_DIR/PreReqs/pacemaker-1.1.20/corosync-2.4.4.linbit-1.0+20181123+918bc4781279.el7.x86_64.rpm \
$RDQM_DIR/PreReqs/pacemaker-1.1.20/corosynclib-2.4.4.linbit-1.0+20181123+918bc4781279.el7.x86_64.rpm \
$RDQM_DIR/PreReqs/pacemaker-1.1.20/crmsh-3.0.3.linbit-1.0+20181114+2309c21a33a9.el7.noarch.rpm \
$RDQM_DIR/PreReqs/pacemaker-1.1.20/crmsh-scripts-3.0.3.linbit-1.0+20181114+2309c21a33a9.el7.noarch.rpm \
$RDQM_DIR/PreReqs/pacemaker-1.1.20/libqb-1.0.3.linbit-3.0+20190115+c880fcfd8d67.el7.x86_64.rpm \
$RDQM_DIR/PreReqs/pacemaker-1.1.20/pacemaker-1.1.20.linbit-1+20190404+eab6a2092b71.el7.2.x86_64.rpm \
$RDQM_DIR/PreReqs/pacemaker-1.1.20/pacemaker-cli-1.1.20.linbit-1+20190404+eab6a2092b71.el7.2.x86_64.rpm \
$RDQM_DIR/PreReqs/pacemaker-1.1.20/pacemaker-cluster-libs-1.1.20.linbit-1+20190404+eab6a2092b71.el7.2.x86_64.rpm \
$RDQM_DIR/PreReqs/pacemaker-1.1.20/pacemaker-libs-1.1.20.linbit-1+20190404+eab6a2092b71.el7.2.x86_64.rpm \
$RDQM_DIR/PreReqs/pacemaker-1.1.20/python-parallax-1.0.1.linbit-1.1.noarch.rpm \
$RDQM_DIR/PreReqs/pacemaker-1.1.20/resource-agents-4.2.0.linbit-1+20190116+6a6f48180a59.el7.7.x86_64.rpm"
```

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

7. No changes are required, so close the editor by click the “X”.

```

#!/bin/bash
# <copyright
# notice="/lm-source-program"
# pids="5724-H72"
# years="2017,2020"
# crc=""
# Licensed Materials - Property of IBM
#
# 5724-H72
#
# (C) Copyright IBM Corp. 2017, 2020 All Rights Reserved.
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with
# IBM Corp.
# </copyright>
RDQM_DIR=$(dirname "$0")

```

8. Run the script to install RDQM with the following command.

```
./installRDQMsupport
```

```

root@rdqm1:/home/ibmuser/mq915/MQServer/Advanced/RDQM
File Edit View Terminal Help
[root@rdqm1 RDQM]# ./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

```

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

```
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 |
|   Verifying  : MQSeriesRDQM-9.1.5-0.x86_64 |
|=====|
| Installed: |
|   MQSeriesRDQM.x86_64 0:9.1.5-0 |
|=====|
| Complete! |
| [root@rdqm1 RDQM]# |
```

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

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. While still logged in as root on **rdqm1** start the firewall with following command:

```
systemctl start firewalld
```

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

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

```
root@rdqm1:/home/ibmuser/mq915/MQServer/Advanced/RDQM
File Edit View Search Terminal Help
[root@rdqm1 RDQM]# systemctl start firewalld
[root@rdqm1 RDQM]# /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
[root@rdqm1 RDQM]#
```

3. Exit from the root user.

```
exit
```

Be sure to exit out of root before continuing with the next section.

Configure the 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.

The cluster is defined using the rdqm.ini file. The /home/ibmuser/ directory contains the rdqm.ini file we will use.

1. Return to rdqm1 and make sure you have exited your su session. Review this file with command:

```
cd ~/mq915  
cat rdqm-ha.ini
```

```
ibmuser@rdqm1:~/mq915  
File Edit View Search Terminal Help  
[ibmuser@rdqm1 RDQM]$ cd ~/mq915  
[ibmuser@rdqm1 mq915]$ cat rdqm-ha.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 mq915]$ █
```

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

```
cd /var/mqm  
cp ~/mq915/rdqm-ha.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-ha.ini rdqm.ini
[ibmuser@rdqm1 mqm]$
```

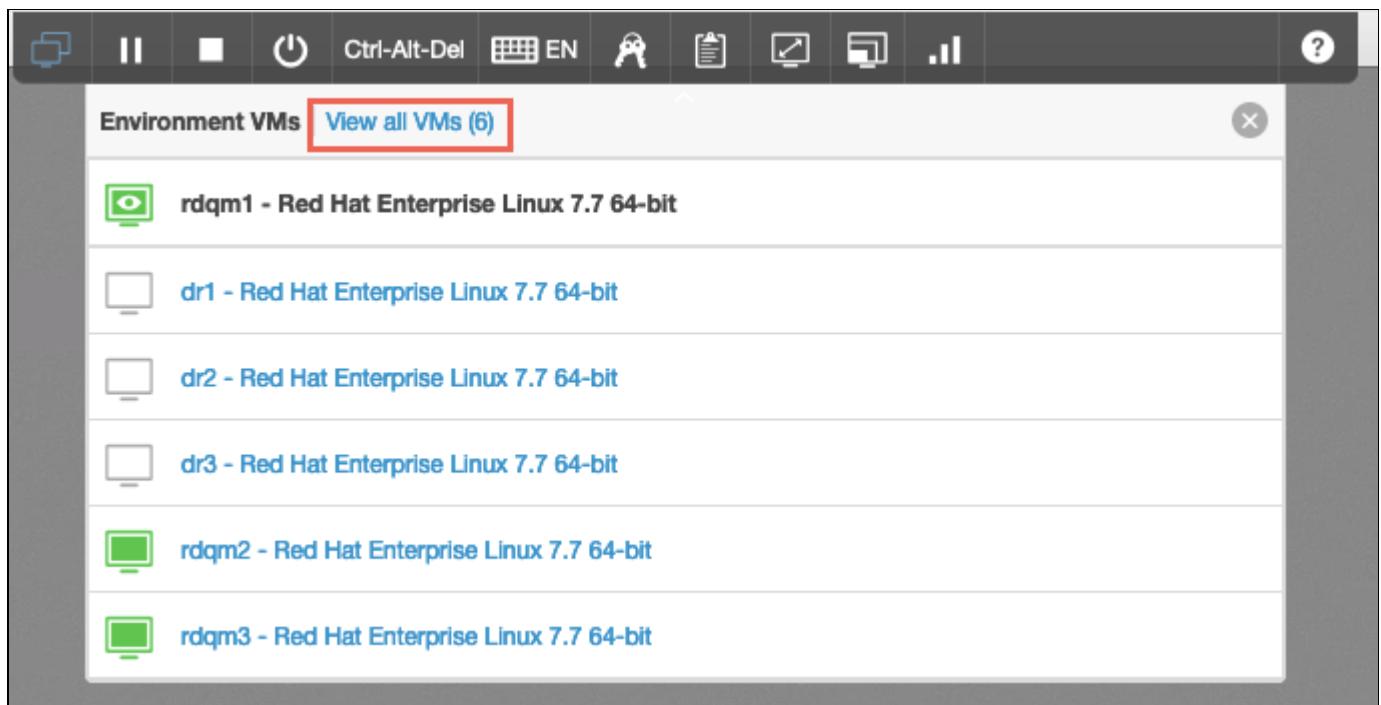
3. **IMPORTANT:** Repeat these commands on **rdqm2** and **rdqm3** before continuing.

Hint: You can click the monitor icon for rdqm2 and rdqm3 which will launch the desktop for each in a new browser tab.

Click the arrow in the black bar at top of screen to open the Skytap menu. Click the monitors icon on left end.



Click **View all VMs (6)** to show the VMs. Then you can click the monitor icons. **Hint**



4. Return to the primary node **rdqm1**. Enter the following command to see the options for the *rdqmadm* command:

```
sudo rdqmadm --help
```

```
ibmuser@rdqm1:/var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm1 mqm]$ sudo rdqmadm --help
Usage: rdqmadm ( -p -m QMgrName [ -n NodeName | -d ] |
                  -s [ -n NodeName ] |
                  -r [ -n NodeName ] |
                  -c |
                  -u )
-c Initialize the Pacemaker cluster, using the settings specified in
      /var/mqm/rdqm.ini.
-d Clear the preferred location.
-m Queue manager name.
-n Node name.
-p Assign the local or specified node as the preferred location for
      the named queue manager.
-r Resume the local or specified node.
-s Suspend the local or specified node.
-u Delete the Pacemaker cluster configuration.
[ibmuser@rdqm1 mqm]$
```

The “-c” parameter is used to configure the Pacemaker cluster

5. Enter the command to configure RDQM Pacemaker HA cluster:

```
sudo rdqmadm -c
```

⚠ Warning:

If you are copying the command snippets from this lab guide and pasting them in the terminal windows: Beware that if the command does not work it may be that the copy assumes a long or double hyphen instead of a single hyphen. Just try overtyping the hyphen with the regular hyphen.

6. You have received the message that the replicated data system has been completed on this node

(10.0.3.1 - rdqm1). You are also told that you need to run the same command on **10.0.3.2** (rdqm2) and **10.0.3.3** (rdqm3).

```
[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 should be configured on all replicated data
nodes.
You need to run the following command on nodes '10.0.3.2' and '10.0.3.3':
rdqmadm -c
[ibmuser@rdqm1 mqm]$ █
```

7. Do as instructed and run the command on rdqm2 and rdqm3. Make sure to use root access (sudo). After running the command on **rdqm3**, you will see that the RDQM subsystem configuration has been completed.

```
sudo rdqmadm -c
```

```
ibmuser@rdqm2:/var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm2 mqm]$ sudo 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 the following command on node '10.0.3.3':
rdqmadm -c
[ibmuser@rdqm2 mqm]$ █
```

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

Configure the HA RDQM

The high availability replicated data queue manager (RDQM) now needs to be created. The `crtmqm` command includes a `-sx` parameter for a replicated data queue manager.

The secondary RDQMs need to be created first on two of the nodes with the `-sxs` parameter. This parameter indicates that the primary node's data will be replicated to this node and required disk space gets allocated.

1. Run the following command on **rdqm2** and **rdqm3**

```
sudo crtmqm -sxs -fs 3 QMHA
```

ibmuser@rdqm3:/var/mqm

File Edit View Search Terminal Help

```
[ibmuser@rdqm3 mqm]$ sudo rdqmadm -c
Configuring the replicated data subsystem.
The replicated data subsystem has been configured on this node.
The replicated data subsystem configuration is complete.
[ibmuser@rdqm3 mqm]$ sudo crtmqm -sxs -fs 3 QMHA
Creating replicated data queue manager configuration.
IBM MQ secondary queue manager created.
[ibmuser@rdqm3 mqm]$
```

Note:

When running RDQM commands, you should run with root access. So precede the commands with `sudo`. You may notice that as a member of the `mqm` group `ibmuser`'s commands complete even though not using root access. But when the command is complete you are reminded to run with `sudo`. You may notice this in the screen shots.

2. Now that the secondary queue managers have been created, the primary RDQM is created on the node where the command is run. This node becomes the RDQM's preferred location by default.

You will now create the primary RDQM on **rdqm1**. Return to **rdqm1**. Create the primary RDQM, which will listen on port 1500 with the following command:

```
sudo crtmqm -p 1500 -sx -fs 3 QMHA
```

```
ibmuser@rdqm1:/var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm1 mqm]$ crtmqm -p 1500 -sx -fs 3 QMHA
Creating replicated data queue manager configuration.
The secondary queue manager must first be created on the other replicated data
nodes.
You need to run the following command on nodes 'rdqm2' and 'rdqm3':
crtmqm -sxs -fs 3072M QMHA
AMQ3812E: Failed to create replicated data queue manager configuration.
Command '/opt/mqm/bin/crtmqm' run with sudo.
[ibmuser@rdqm1 mqm]$ crtmqm -p 1500 -sx -fs 3 QMHA
Creating replicated data queue manager configuration.
IBM MQ queue manager created.
Directory '/var/mqm/vols/qmha/qmgr/qmha' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QMHA'.
Default objects statistics : 84 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
Enabling replicated data queue manager.
Replicated data queue manager enabled.
```

The primary queue manager is now created and running. **Note:**

In order to create RDQMs and issue commands, *ibmuser* must be in the *mqm* and *haclient* groups. *haclient* group is created as part of the RDQM installation. This was done as setup for this lab.

1. Check the status of the queue manager:

```
sudo rdqmstatus -m QMHA
```

Initially, the output should look similar to the following. Synchronization is in progress.

```
ibmuser@rdqm1:~
```

File Edit View Search Terminal Help

```
[ibmuser@rdqm1 ~]$ rdqmstatus -m QMHA
```

Node:	rdqm1
Queue manager status:	Starting
Queue manager file system:	58MB used, 2.9GB allocated [2%]
HA role:	Primary
HA status:	Synchronization in progress
HA control:	Enabled
HA current location:	This node
HA preferred location:	This node
HA floating IP interface:	None
HA floating IP address:	None

Node:	rdqm2
HA status:	Synchronization in progress
HA synchronization progress:	15.40%
HA estimated time of completion:	2020-06-11 14:39:59

Node:	rdqm3
HA status:	Synchronization in progress
HA synchronization progress:	53.20%
HA estimated time of completion:	2020-06-11 14:39:07

```
[ibmuser@rdqm1 ~]$
```

You can run the same command on the other nodes and get similar output.

```
ibmuser@rdqm2:/var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm2 mqm]$ rdqmstatus -m QMHA
Node: rdqm2
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Mixed
HA control: Enabled
HA current location: rdqm1
HA preferred location: rdqm1
HA floating IP interface: None
HA floating IP address: None

Node: rdqm1
HA status: Synchronization in progress
HA synchronization progress: 34.97%
HA estimated time of completion: 2020-06-11 14:39:48

Node: rdqm3
HA status: Inconsistent
HA out of sync data: 122872KB
Command '/opt/mqm/bin/rdqmstatus' run with sudo.
[ibmuser@rdqm2 mqm]$
```

- When the nodes have completed synchronising, the HA Status field on all nodes should change to 'Normal'. This may take a few minutes.

Repeat the **sudo rdqmstatus** command again until you see the normal HA status (as shown on **rdqm1**).

```
ibmuser@rdqm1:~ - □ ×
File Edit View Search Terminal Help
HA synchronization progress: 53.20%
HA estimated time of completion: 2020-06-11 14:39:07
[ibmuser@rdqm1 ~]$ rdqmstatus -m QMHA
Node:
Queue manager status: rdqm1
CPU: Running
Memory: 0.01%
Queue manager file system: 183MB
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: None
HA floating IP address: None

Node: rdqm2
HA status: Normal

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

Note:

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

Simple testing of RDQM

Once all nodes have an HA status of Normal, you can commence testing. You will perform some tests, which will show different use cases.

Failing over an RDQM instance to another node

The easiest way to force an RDQM instance to fail over to another node is to change its preferred location.

The default location for RDQM is rdqm1. You will fail the RDQM instance to 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 rdqadm -m QMHA -p
```

```
ibmuser@rdqm2:~var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm2 mqm]$ rdqadm -m QMHA -p
The preferred replicated data node has been set to 'rdqm2' for queue manager
'QMHA'.
Command '/opt/mqm/bin/rdqadm' run with sudo.
```

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

```
sudo rdqmstatus -m QMHA
```

```
[ibmuser@rdqm2 mqm]$ rdqmstatus -m QMHA
Node: rdqm2
Queue manager status: Running
CPU: 0.45%
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: None
HA floating IP address: None

Node: rdqm1
HA status: Normal

Node: rdqm3
HA status: Normal
Command '/opt/mqm/bin/rdqmstatus' run with sudo.
[ibmuser@rdqm2 mqm]$
```

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

```
sudo rdqmadm -m QMHA -p
```

```
ibmuser@rdqm1:~$ sudo rdqmadm -m QMHA -p
The preferred replicated data node has been set to 'rdqm1' for queue manager
'QMHA'.
```

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

```
sudo sudo rdqmstatus -m QMHA
```

```
[ibmuser@rdqm1 ~]$ sudo rdqmstatus -m QMHA
Node: rdqm1
Queue manager status: Running
CPU: 0.41%
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: None
HA floating IP address: None

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 QMHA is running (**rdqm1**), return to ibmuser's terminal window. As the user **ibmuser** (not as root), issue the command to suspend the queue manager:

```
rdqmadm -s
```

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

```
ibmuser@rdqm1:~$ rdqmadm -s
The replicated data node 'rdqm1' has been suspended.
[ibmuser@rdqm1 ~]$ rdqmstatus -m QMHA
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: None
HA floating IP address: None

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 QMHA and you see that it is now running on rdqm2.

```
sudo rdqmstatus -m QMHA
```

```
ibmuser@rdqm2:var/mqm
File Edit View Search Terminal Help
[ibmuser@rdqm2 mqm]$ sudo rdqmstatus -m QMHA
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: None
HA floating IP address: None

Node: rdqm1
HA status: Remote node in standby
HA out of sync data: 592KB

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

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. QMHA 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:~
```

```
File Edit View Search Terminal Help
[ibmuser@rdqm1 ~]$ rdqmstatus -m QMHA
Node: rdqm1
Queue manager status: Status not available
Queue manager file system: 0MB used, 0.0GB allocated [100%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: Unknown
HA preferred location: This node
HA floating IP interface: None
HA floating IP address: None

Node: rdqm2
HA status: Normal

Node: rdqm3
HA status: Normal
```

5. After the node has fully resumed, QMHA 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 QMHA
```

```
ibmuser@rdqm1:~
```

File	Edit	View	Search	Terminal	Help
[ibmuser@rdqm1 ~]\$ rdqmstatus -m QMHA					
Node:	rdqm1				
Queue manager status:	Running				
CPU:	3.03%				
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:	None				
HA floating IP address:	None				
Node:	rdqm2				
HA status:	Normal				
Node:	rdqm3				
HA status:	Normal				
[ibmuser@rdqm1 ~]\$ █					

Testing RDQM using HA 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**.

Create MQ resources

To run these samples, 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. You will turn off CHLAUTH and CONNAUTH completely to keep things simple.

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

```
runmqsc QMHA
```

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(1500),1  
0.0.1.2(1500),10.0.1.3(1500)') QMNAME(QMHA)
```

```
END
```

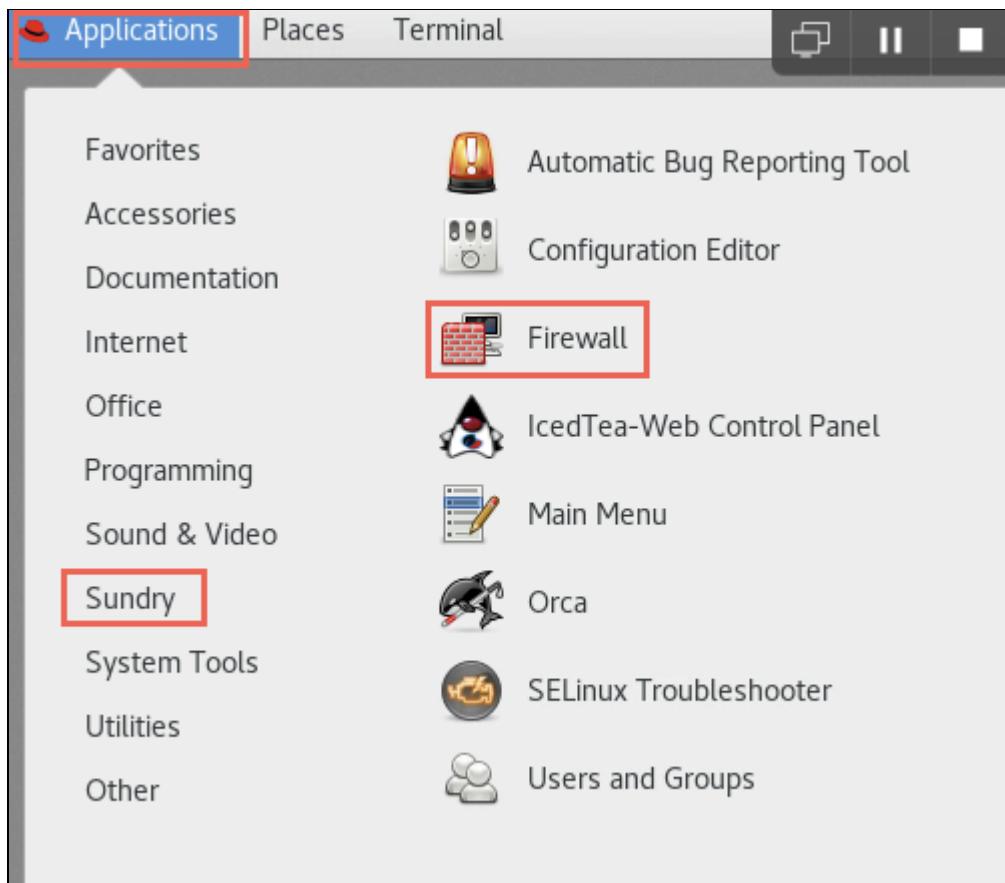
```
ibmdemo@miqmp:/var/mqm
File Edit View Search Terminal Help
[ibmdemo@miqmp mqm]$ runmqsc myRDQM
5724-H72 (C) Copyright IBM Corp. 1994, 2019.
Starting MQSC for queue manager myRDQM.

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(SOURCE) DEFPSIST(YES)
    4 : DEFINE QLOCAL(SOURCE) DEFPSIST(YES)
AMQ8150E: IBM MQ object already exists.
DEFINE QLOCAL(TARGET) DEFPSIST(YES)
    5 : DEFINE QLOCAL(TARGET) DEFPSIST(YES)
AMQ8006I: IBM MQ queue created.
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP)
    6 : DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP)
AMQ8014I: IBM MQ channel created.
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) CONNAME('10.0.0.1(1500),
10.0.0.2(1500),10.0.0.3(1500)') QMNAME(myRDQM)

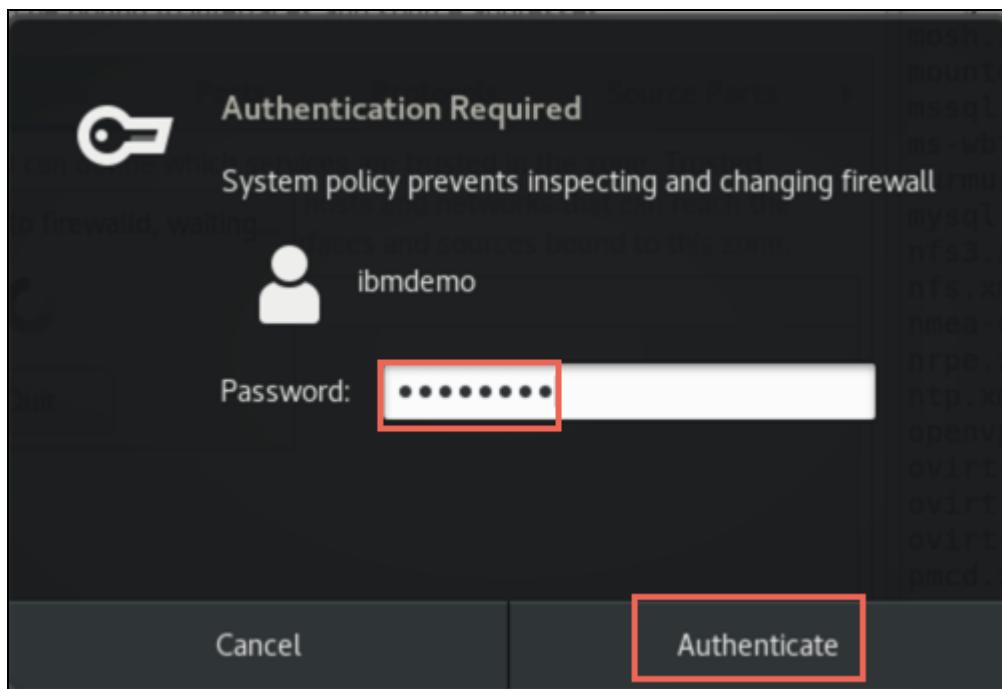
    7 : DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) CONNAME('10.0.0.
1(1500),10.0.0.2(1500),10.0.0.3(1500)') QMNAME(myRDQM)
AMQ8014I: IBM MQ channel created.
end
    8 : end
7 MQSC commands read.
No commands have a syntax error.
One valid MQSC command could not be processed.
[ibmdemo@miqmp mqm]$
```

Update firewall rules

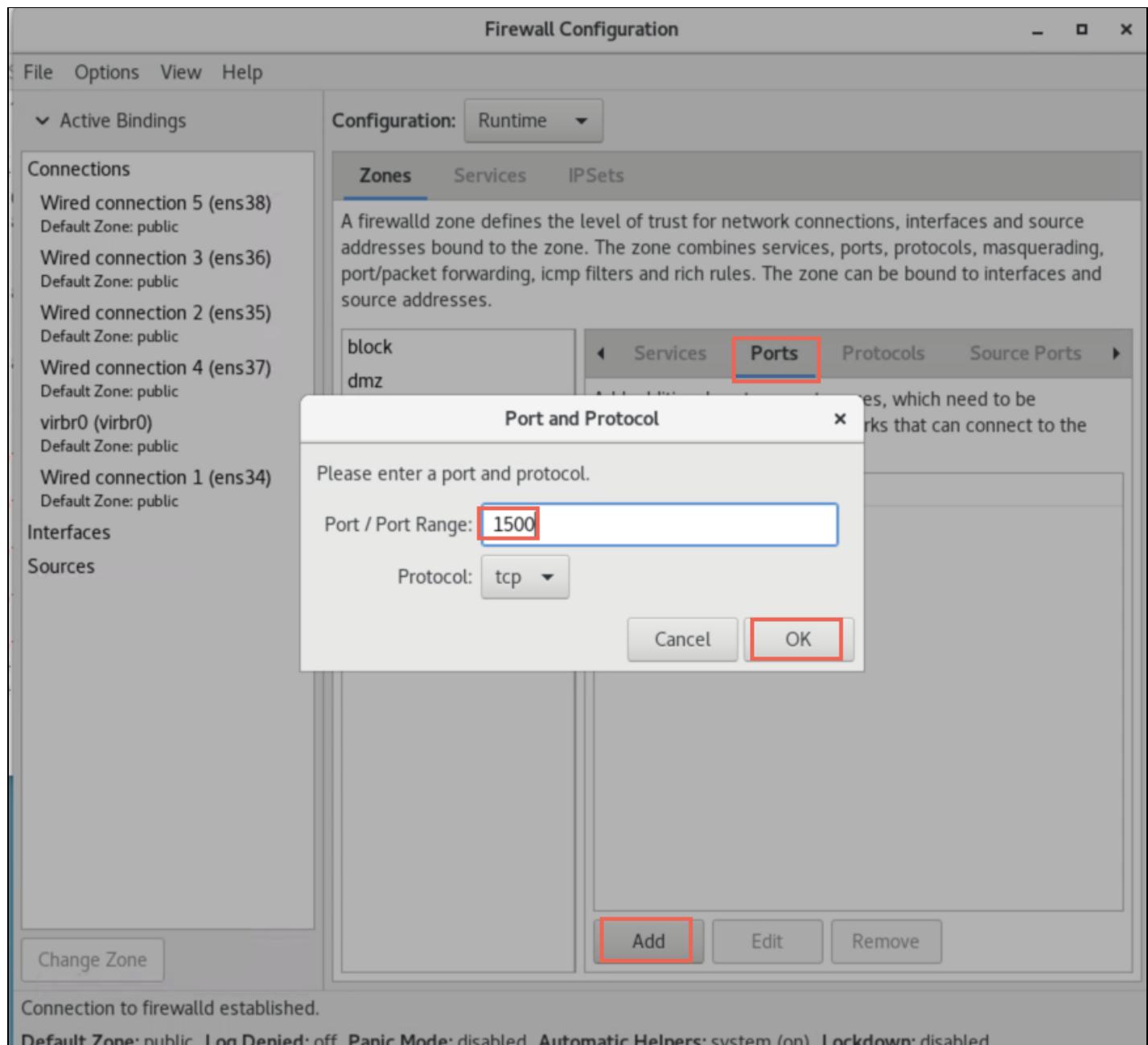
1. On each of the nodes, open the firewall port defined. Open the firewall from the top left of the screen, under *Applications -> Sundry -> Firewall*.



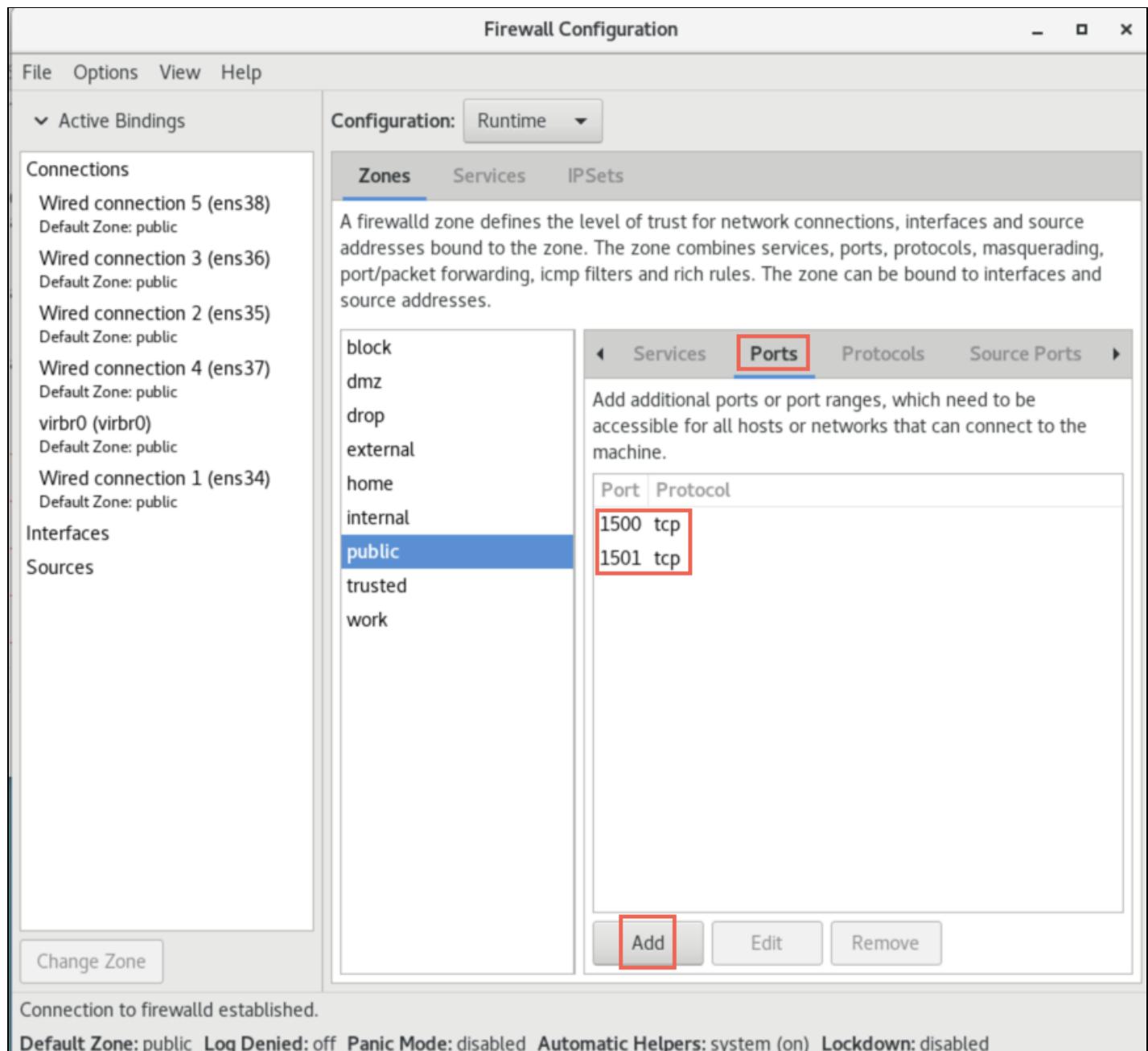
2. Enter the password for ibmuser, **engageibm**, then click *Authenticate*.



3. In the Ports pane, add TCP ports 1500 and 1501 (the latter will be used later).



Results should look like this:



Don't forget, each node must have these ports opened in the firewall.

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.

1. On **rdqm1**, in the user ibmuser terminal window, enter:

```
export MQSERVER='CHANNEL1/TCP/10.0.1.1(1500),10.0.1.2(1500),10.0.1.3(1500)'
```

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

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

The screenshot shows a terminal window with a red box highlighting the user prompt "user@rdqm1:/opt/mqm/samp/bin". Below the prompt, there are three command lines: "File Edit View Search Terminal Help", "[ibmuser@rdqm1 ~]\$ export MQSERVER='CHANNEL1/TCP/10.0.1.1(1500),10.0.1.2(1500),10.0.1.3(1500)'", "[ibmuser@rdqm1 ~]\$ cd /opt/mqm/samp/bin", and "[ibmuser@rdqm1 bin]\$./amqsghac TARGET QMHA". The last command is preceded by "Sample AMQSGHAC start".

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

Leave this command to run!

3. Now switch to **rdqm2**. In the user ibmuser terminal window, enter:

```
export MQSERVER='CHANNEL1/TCP/10.0.1.1(1500),10.0.1.2(1500),10.0.1.3(1500)'
```

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

```
cd /opt/mqm/samp/bin  
./amqsmhac -s SOURCE -t TARGET -m QMHA
```

```
ibmuser@rdqm2:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@rdqm2 ~]$ export MQSERVER='CHANNEL1/TCP/10.0.1.1(1500),10.0.1.2(1500),10.0.1.3(1500)'
[ibmuser@rdqm2 ~]$ cd /opt/mqm/samp/bin
[ibmuser@rdqm2 bin]$ ./amqshac -s SOURCE -t TARGET -m QMHA
Sample AMQSHAC start
```

Leave this command to run!

⚠ Warning:

If you are copying the command snippets from this lab guide and pasting them in the terminal windows: Beware that if the command does not work it may be that the copy assumes a long or double hyphen instead of a single hyphen. Just try overtyping the hyphen with the regular hyphen.

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

```
export MQSERVER='CHANNEL1/TCP/10.0.1.1(1500),10.0.1.2(1500),10.0.1.3(1500)'
```

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

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

```
ibmuser@rdqm3:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@rdqm3 ~]$ export MQSERVER='CHANNEL1/TCP/10.0.1.1(1500),10.0.1.2(1500),10.0.1.3(1500)'
[ibmuser@rdqm3 ~]$ cd /opt/mqm/samp/bin
[ibmuser@rdqm3 bin]$ ./amqsphac SOURCE QMHA
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 bin]$ ./amqshgac TARGET QMHA
bash: ./amqshgac: No such file or directory
[ibmuser@rdqm1 bin]$ export MQSERVER='CHANNEL1/TCP/10.0.1.1(1500),10.0.1.2(1500),10.0.1.3(1500)'
[ibmuser@rdqm1 bin]$ pwd
/opt/mqm/samp/bin
[ibmuser@rdqm1 bin]$ ./amqsg hac TARGET QMHA
Sample AMQSGHAC start
message <Message 1>
message <Message 2>
message <Message 3>
message <Message 4>
message <Message 5>
message <Message 6>
message <Message 7>
message <Message 8>
message <Message 9>
message <Message 10>
message <Message 11>
message <Message 12>
message <Message 13>
message <Message 14>
message <Message 15>
message <Message 16>
message <Message 17>
message <Message 18>
```

Note: At this stage, the queue manager is running on the primary node (rdqm1) and each sample program is able to communicate with it, using the first location specified in the MQSERVER environment variable:

CHANNEL1/TCP/**10.0.1.1(1500)**,10.0.1.2(1500),10.0.1.3(1500)

Move the RDQM

You will now use the approach of controlling where the RDQM runs by changing its preferred location, in this case to rdqm2.

1. Switch to **rdqm2**. In a new terminal window, run the following command as ibmuser:

```
sudo rdqmadm -m QMHA -p
```

2. Check that the queue manager is indeed running on **rdqm2**, by running:

```
sudo rdqmstatus -m QMHA
```

```
ibmuser@rdqm2:/var/mqm/errors - □ ×
File Edit View Search Terminal Help
[ibmuser@rdqm2 errors]$ sudo rdqmadm -m QMHA -p
The preferred replicated data node has been set to 'rdqm2' for queue manager
'QMHA'.
[ibmuser@rdqm2 errors]$ sudo rdqmstatus -m QMHA
Node: rdqm2
Queue manager status: Running
CPU: 0.00%
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: None
HA floating IP address: None

Node: rdqm1
HA status: Normal

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

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

```
ibmuser@rdqm2:/opt/mqm/samp/bin - □ ×
File Edit View Search Terminal Help
[ibmuser@rdqm2 bin]$ ./amqsmhac -s SOURCE -t TARGET -m QMHA
Sample AMQSMHAC start

10:18:53 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 27ms)
10:18:53 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1177ms)
10:18:54 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 2531ms)
10:18:56 : EVENT : Connection Reconnected
```

3. 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:

```
ibmuser@rdqm1:/opt/mqm/samp/bin  
File Edit View Search Terminal Help  
message <Message 21>  
message <Message 22>  
message <Message 23>  
message <Message 24>  
message <Message 25>  
message <Message 26>  
message <Message 27>  
message <Message 28>  
message <Message 29>  
message <Message 30>  
message <Message 31>  
message <Message 32>  
message <Message 33>  
message <Message 34>  
message <Message 35>  
message <Message 36>  
12:02:52 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 86ms)  
12:02:52 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1094ms)  
12:02:53 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 2371ms)  
12:02:56 : EVENT : Connection Reconnected  
message <Message 37>  
message <Message 38>  
message <Message 39>  
message <Message 40>  
message <Message 41>  
message <Message 42>
```

4. Now switch to rdqm3.

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

```
ibmuser@rdqm3:/opt/mqm/samp/bin
File Edit View Search Terminal Help
message <Message 25>
message <Message 26>
message <Message 27>
message <Message 28>
message <Message 29>
message <Message 30>
message <Message 31>
message <Message 32>
message <Message 33>
message <Message 34>
message <Message 35>
message <Message 36>
12:02:52 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 129ms)
12:02:52 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1369ms)
12:02:54 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 2449ms)
12:02:56 : EVENT : Connection Reconnected
message <Message 37>
message <Message 38>
message <Message 39>
message <Message 40>
message <Message 41>
```

Note: Now the queue manager is running on rdqm2, but each sample program is still able to communicate with it, this time using the second location specified in the MQSERVER environment variable:

CHANNEL1/TCP/10.0.1.1(1500),**10.0.1.2(1500)**,10.0.1.3(1500)

HA Sample programs with RDQM & Floating IP address

It is possible to associate a floating IP address with an RDQM so that it is not necessary to reconfigure clients, etc. with three IP addresses for the same queue manager. In this case, you will assign the floating address 10.0.1.10 to virtual adapter ens35, where currently you already have a fixed IP address configured on each virtual machine.

1. Stop (with **ctrl-C**) the HA sample programs that are currently running on each of the nodes.

Do not close the terminal windows just yet as you will be running these programs again!

2. Switch to **rdqm2**. As this is currently the primary node for the QMHA queue manager, add the floating IP address, by running the command (as **ibmuser**):

```
sudo rdqmint -m QMHA -a -f 10.0.1.10 -l ens35
```

ibmuser@rdqm2:~var/mqm/errors

File Edit View Search Terminal Help

[ibmuser@rdqm2 errors]\$ sudo rdqmint -m QMHA -a -f 10.0.1.10 -l ens35

The floating IP address '10.0.1.10' has been added to queue manager 'QMHA'.

3. It is recommended to add another listener specifically for this floating 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. In a terminal window, create a listener by entering the runmqsc commands as follows:

```
runmqsc QMHA
```

```
DEFINE LISTENER(FLOATING_LISTENER) TRPTYPE(TCP) CONTROL(QMGR) IPADDR(10.0.1.10) PORT(1501)
```

```
end
```

```
ibmuser@rdqm2:/var/mqm/errors - □ ×
File Edit View Search Terminal Help
[ibmuser@rdqm2 errors]$ runmqsc QMHA
5724-H72 (C) Copyright IBM Corp. 1994, 2020.
Starting MQSC for queue manager QMHA.

DEFINE LISTENER(FLOATING.LISTENER) TRPTYPE(TCP) CONTROL(QMGR) IPADDR(10.0.1.10)
PORT(1501)
  1 : DEFINE LISTENER(FLOATING.LISTENER) TRPTYPE(TCP) CONTROL(QMGR) IPADDR(10
.0.1.10) PORT(1501)
AMQ8626I: IBM MQ listener created.
end
  2 : end
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
[ibmuser@rdqm2 errors]$ █
```

4. Switch to **rdqm1**. The new listener will not be started until the queue manager is restarted, so move the queue manager back to its original node by running the following command on node **rdqm1** in a terminal window, as user ibmuser:

```
sudo rdqmadm -m QMHA -p
```

```
ibmuser@rdqm1:/var/mqm/errors - □ ×
File Edit View Search Terminal Help
[ibmuser@rdqm1 errors]$ sudo rdqmadm -m QMHA -p
The preferred replicated data node has been set to 'rdqm1' for queue manager
'QMHA'.
```

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

```
netstat -ant | grep 150
```

```
ibmuser@rdqm1:~/var/mqm/errors
File Edit View Search Terminal Help
[ibmuser@rdqm1 errors]$ netstat -ant | grep 150
tcp6      0      0 ::::1500          ::::*                  LISTEN
tcp6      0      0 10.0.1.10:1501    ::::*                  LISTEN
[ibmuser@rdqm1 errors]$
```

The first listener is the one created because **-p 1500** was specified on the **crtmqm** command. This listener is listening on port 1500 on every IP address. The second listener, however, is listening on port **1501** on the floating IP address only.

- Now that you have a floating IP address associated with QMHA, you can change the MQSERVER environment variable to CHANNEL1/TCP/10.0.1.10(1501).

Locate the window where the amqsghac program was running (on rdqm1) and enter:

```
export MQSERVER='CHANNEL1/TCP/10.0.1.10(1501)'
```

- Now re-run the command: **amqsghac TARGET QMHA**.

```
./amqsghac TARGET QMHA
```

```
ibmuser@rdqm1:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@rdqm1 bin]$ export MQSERVER='CHANNEL1/TCP/10.0.1.10(1501)'
[ibmuser@rdqm1 bin]$ ./amqsghac TARGET QMHA
Sample AMQSGHAC start
```

Leave this command to run!

- Switch to **rdqm2**. Locate the window where the amqsmhac program was running and enter:

```
export MQSERVER='CHANNEL1/TCP/10.0.1.10(1501)'
```

8. Now re-run the command: **amqsmhac -s SOURCE -t TARGET -m QMHA**

```
./amqsmhac -s SOURCE -t TARGET -m QMHA
```

The screenshot shows a terminal window with a light gray background and a dark gray header bar. The header bar contains the text "ibmuser@rdqm2:/opt/mqm/samp/bin". Below the header, there is a menu bar with options: File, Edit, View, Search, Terminal, Help. The main area of the terminal shows a command-line session:

```
File Edit View Search Terminal Help  
[ibmuser@rdqm2 bin]$ export MQSERVER='CHANNEL1/TCP/10.0.1.10(1501)'  
[ibmuser@rdqm2 bin]$ ./amqsmhac -s SOURCE -t TARGET -m QMHA  
Sample AMQSMHAC start
```

The last two lines of the command history are highlighted with a red rectangle.

Leave this command to run!

9. Switch to **rdqm3**. Locate the window where the amqsphac program was running and enter:

```
export MQSERVER='CHANNEL1/TCP/10.0.1.10(1501)'
```

10. Now re-run the command: **amqsphac SOURCE QMHA**

```
./amqsphac SOURCE QMHA
```

```
ibmuser@rdqm3:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@rdqm3 bin]$ export MQSERVER='CHANNEL1/TCP/10.0.1.1(1500),10.0.1.2(1500),10.0.1.3(1500)'
[ibmuser@rdqm3 bin]$ ./amqsphac SOURCE QMHA
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!

11. Now repeat the test. This time move it to rdqm3.

- On **rdqm3**, in the another terminal window as user ibmuser.
- Issue the command to move the RDQM to **rdqm3**.

```
...
sudo rdqmadm -p -m QMHA
...
```

```
ibmuser@rdqm3:/var/mqm/errors
File Edit View Search Terminal Help
[ibmuser@rdqm3 errors]$ sudo rdqmadm -m QMHA -p
The preferred replicated data node has been set to 'rdqm3' for queue manager
'QMHA'.
```

- Check the status of the queue manager to make sure it is running on **rdqm3**.

```
ibmuser@rdqm3:/var/mqm/errors
File Edit View Search Terminal Help
[ibmuser@rdqm3 errors]$ sudo rdqmstatus -m QMHA
Node: rdqm3
Queue manager status: Running
CPU: 0.71%
Memory: 186MB
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: 10.0.1.10

Node: rdqm1
HA status: Normal

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

d. Confirm that messages continue to be sent and received without loss after the queue manager has been moved.

```
ibmuser@rdqm3:/opt/mqm/samp/bin
File Edit View Search Terminal Help
message <Message 45>
message <Message 46>
message <Message 47>
message <Message 48>
message <Message 49>
message <Message 50>
message <Message 51>
14:27:22 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 245ms)
14:27:22 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1305ms)
14:27:23 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 2226ms)
14:27:26 : EVENT : Connection Reconnected
message <Message 52>
message <Message 53>
message <Message 54>
message <Message 55>
message <Message 56>
message <Message 57>
```

```
ibmuser@rdqm1:/opt/mqm/samp/bin

File Edit View Search Terminal Help
message <Message 47>
message <Message 48>
message <Message 49>
message <Message 50>
message <Message 51>
14:27:22 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 203ms)
14:27:22 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1301ms)
14:27:24 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1685ms)
14:27:26 : EVENT : Connection Reconnected
message <Message 52>
message <Message 53>
message <Message 54>
message <Message 55>
message <Message 56>
```

```
ibmuser@rdqm2:/opt/mqm/samp/bin

File Edit View Search Terminal Help
[ibmuser@rdqm2 bin]$ export MQSERVER='CHANNEL1/TCP/10.0.1.10(1501)'
[ibmuser@rdqm2 bin]$ ./amqsmhac -s SOURCE -t TARGET -m QMHA
Sample AMQSMHAC start

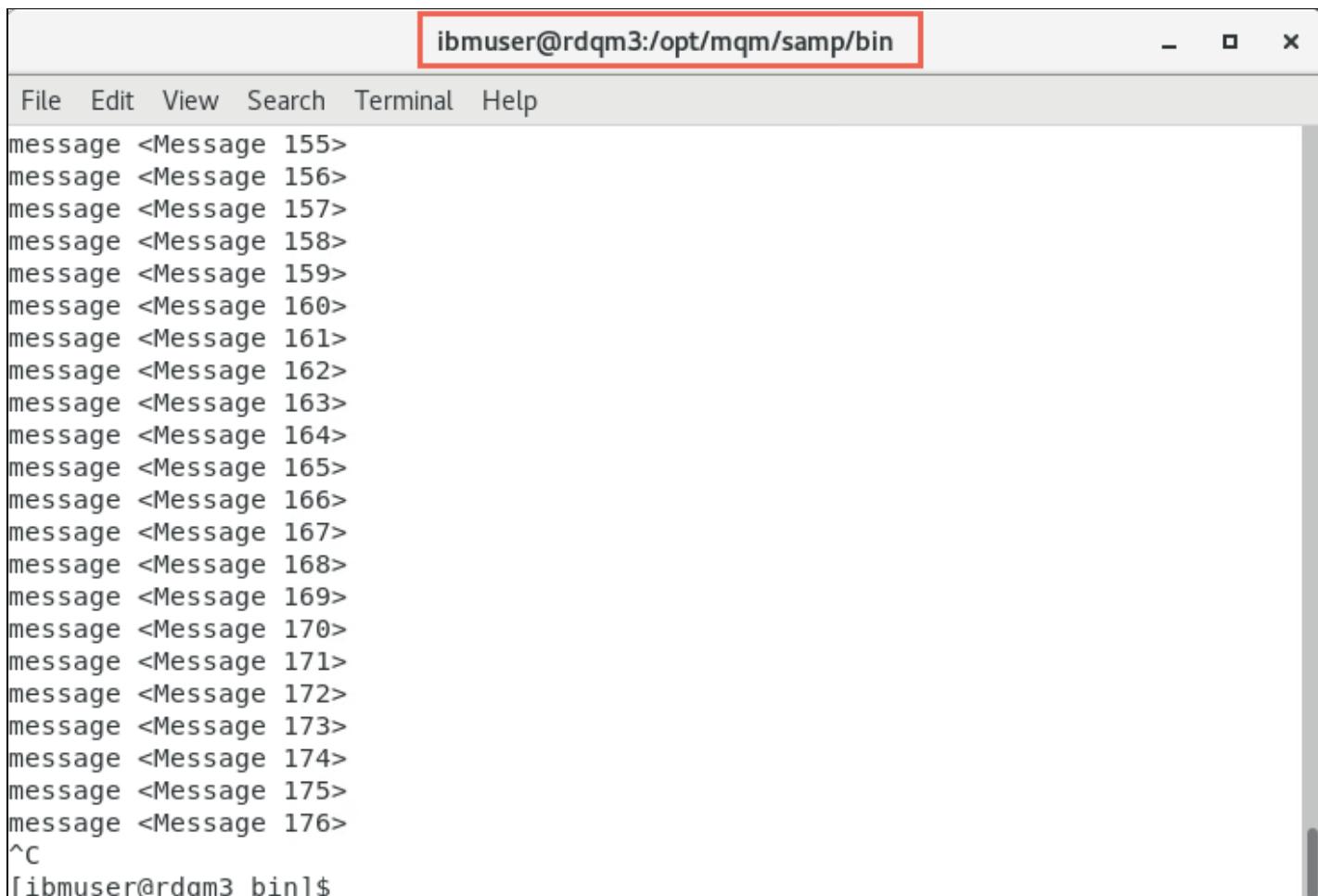
14:27:22 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 86ms)
14:27:22 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1215ms)
14:27:24 : EVENT : Connection Reconnecting (Reason: 2161, Delay: 1478ms)
14:27:26 : EVENT : Connection Reconnected
^C
[ibmuser@rdqm2 bin]$
```

12. When completed testing, stop (with **ctrl-C**) the HA sample programs that are currently running on each of the nodes.

Cleanup environment

Stop and remove RDQMs

1. Stop the sample program on each node by entering **ctrl-c** in the terminal window.



A screenshot of a terminal window titled "ibmuser@rdqm3:/opt/mqm/samp/bin". The window contains a list of 22 messages, each starting with "message <Message 155>" followed by a number from 155 to 176. The window has a red border around its title bar.

```
File Edit View Search Terminal Help
message <Message 155>
message <Message 156>
message <Message 157>
message <Message 158>
message <Message 159>
message <Message 160>
message <Message 161>
message <Message 162>
message <Message 163>
message <Message 164>
message <Message 165>
message <Message 166>
message <Message 167>
message <Message 168>
message <Message 169>
message <Message 170>
message <Message 171>
message <Message 172>
message <Message 173>
message <Message 174>
message <Message 175>
message <Message 176>
^C
[ibmuser@rdqm3 bin]$
```

2. On **rdqm3**, open a terminal window and stop running queue managers. Issue the following commands. Your displays and queue managers may not match the screenshots. Substitute your queue managers.

```
dsmpmq -o all
```

```
endmqm QMHA
```

```
ibmuser@rdqm3:~
```

File Edit View Search Terminal Help

```
[ibmuser@rdqm3 ~]$ dspmq -o all
QMNAME(QMHA) STATUS(Running) DEFAULT(no) STANDBY(Not perm
itted) INSTNAME(Installation1) INSTPATH(/opt/mqm) INSTVER(9.1.5.0) HA(Replicated) DRROLE()
[ibmuser@rdqm3 ~]$ endmqm QMHA
Replicated data queue manager disabled.
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
[ibmuser@rdqm3 ~]$
```

3. Delete the queue manager **QMHA** on the primary node **rdqm3**.

```
sudo dltmqm QMHA
```

```
ibmuser@rdqm3:~
```

File Edit View Search Terminal Help

```
[ibmuser@rdqm3 ~]$ dspmq -o all
QMNAME(QMHA) STATUS(Running) DEFAULT(no) STANDBY(Not perm
itted) INSTNAME(Installation1) INSTPATH(/opt/mqm) INSTVER(9.1.5.0) HA(Replicated) DRROLE()
[ibmuser@rdqm3 ~]$ endmqm QMHA
Replicated data queue manager disabled.
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
[ibmuser@rdqm3 ~]$ sudo dltmqm QMHA
Removing replicated data queue manager configuration.
The queue manager should be deleted on all replicated data nodes.
You need to run the following command on nodes 'rdqm1' and 'rdqm2':
dltmqm QMHA
IBM MQ queue manager 'QMHA' deleted.
[ibmuser@rdqm3 ~]$
```

4. Delete the secondary queue managers on **rdqm2** and **rdqm1** per instructions in the output messages from previous commands.

On **rdqm2** and **rdqm1**, delete the secondary queue manager **QMHA** with the following command:

```
sudo dltmqm QMHA
```

```
ibmuser@rdqm2:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@rdqm2 bin]$ sudo dltmqm QMHA
Removing replicated data queue manager configuration.
The queue manager should be deleted on all replicated data nodes.
You need to run the following command on node 'rdqm1':
dltmqm QMHA
IBM MQ queue manager 'QMHA' deleted.
[ibmuser@rdqm2 bin]$
```

```
ibmuser@rdqm1:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@rdqm1 bin]$ sudo dltmqm QMHA
Removing replicated data queue manager configuration.
IBM MQ queue manager 'QMHA' deleted.
[ibmuser@rdqm1 bin]$
```

Delete HA Cluster

1. Still on **rdqm1** unconfigure (delete) the Pacemaker HA cluster.

```
sudo rdqadm -u
```

```
ibmuser@rdqm1:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@rdqm1 bin]$ sudo rdqadm -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 the following command on nodes 'rdqm2' and 'rdqm3':
rdqadm -u
[ibmuser@rdqm1 bin]$
```

2. Delete the pacemaker HA cluster on the secondary nodes **rdqm2** and **rdqm3** per instructions in the output messages from previous commands.

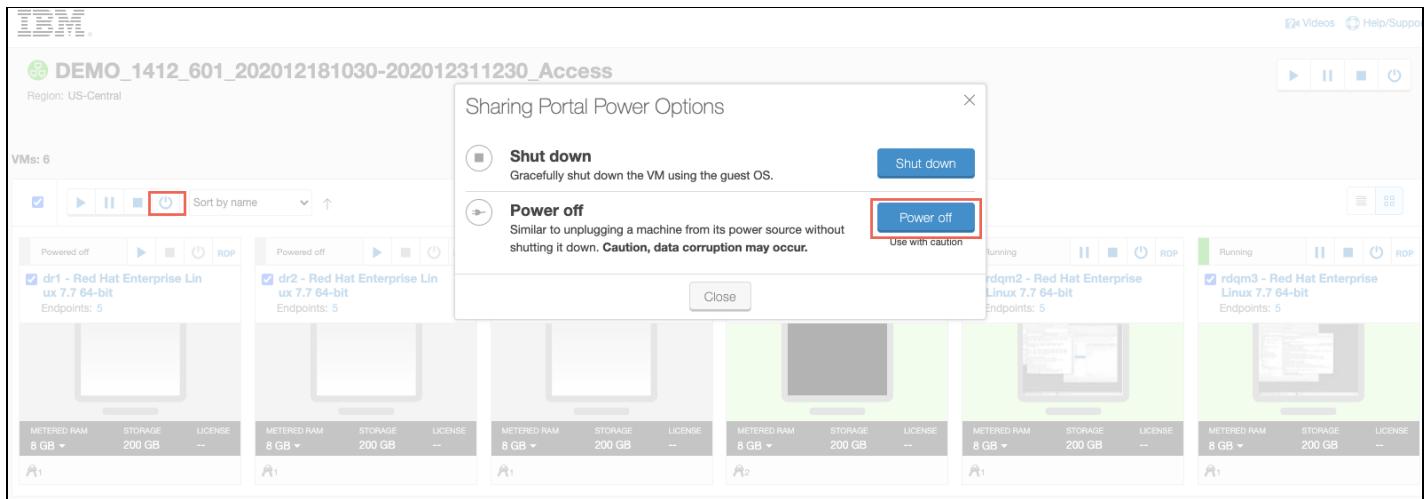
On **rdqm2** and **rdqm3**, unconfigure the HA cluster.

```
sudo rdqmadm -u
```

```
ibmuser@rdqm2:/opt/mqm/samp/bin - □ ×
File Edit View Search Terminal Help
[ibmuser@rdqm2 bin]$ sudo dltmqm QMHA
Removing replicated data queue manager configuration.
The queue manager should be deleted on all replicated data nodes.
You need to run the following command on node 'rdqm1':
dltmqm QMHA
IBM MQ queue manager 'QMHA' deleted.
[ibmuser@rdqm2 bin]$ sudo 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 the following command on node 'rdqm3':
rdqmadm -u
[ibmuser@rdqm2 bin]$ █
```

```
ibmuser@rdqm3:~ - █
File Edit View Search Terminal Help
[ibmuser@rdqm3 ~]$ sudo dltmqm QMHA
Removing replicated data queue manager configuration.
The queue manager should be deleted on all replicated data nodes.
You need to run the following command on nodes 'rdqm1' and 'rdqm2':
dltmqm QMHA
IBM MQ queue manager 'QMHA' deleted.
[ibmuser@rdqm3 ~]$ sudo rdqmadm -u
Unconfiguring the replicated data subsystem.
The replicated data subsystem has been unconfigured on this node.
The replicated data subsystem unconfiguration is complete.
[ibmuser@rdqm3 ~]$ █
```

- Finally power off the VMs. Click the pull down menu and go to the *View VMs* display. The three VMs for this lab should still be running and their labels checked. Click the power icon and select *Power off*.



CONGRATULATIONS!

You have completed this hands-on lab.

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

Continue to Lab 3 ([mq_ha_pot_lab3.html](#))

Return MQ HA Menu ([mq_ha_pot_overview.html](#))

©2020 IBM. All rights reserved.

Site last generated: Dec 28, 2020

