

Replicated Data Queue Managers (RDQM) for Disaster Recovery

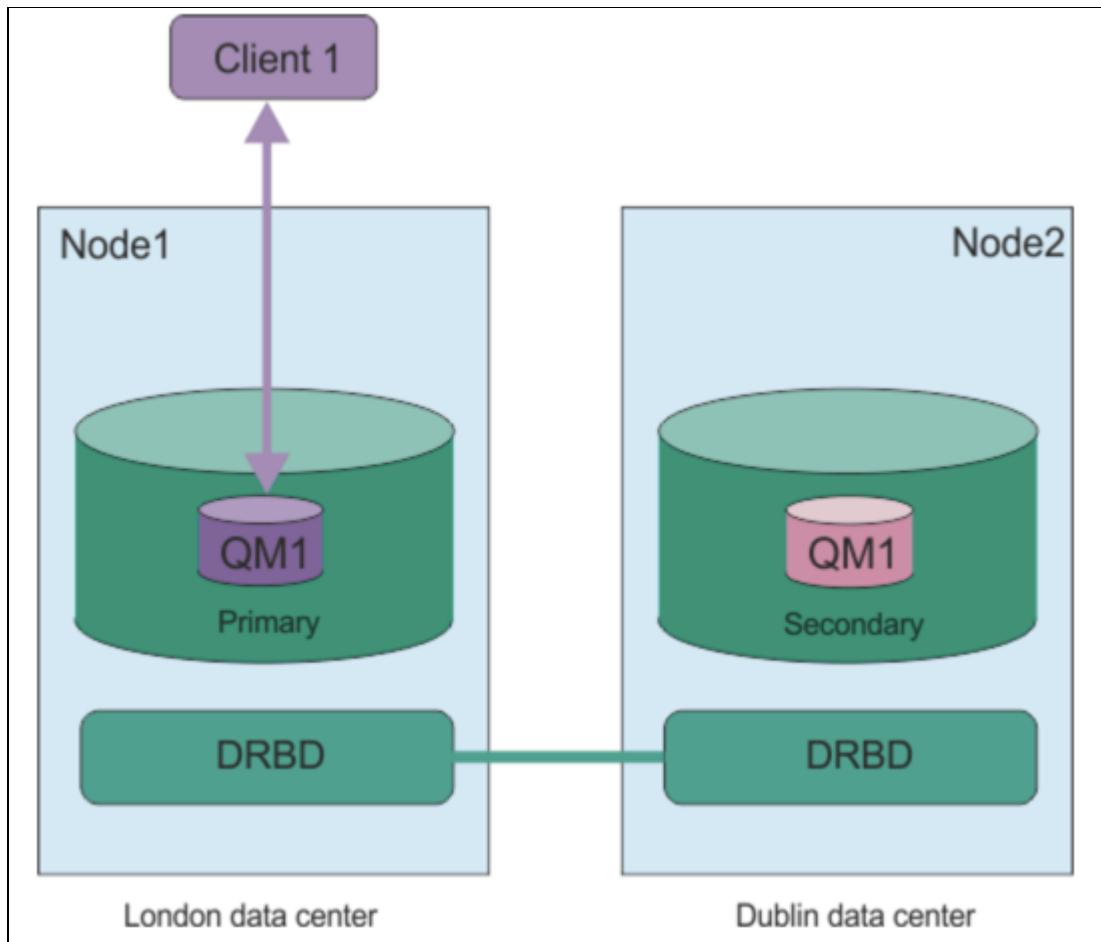
Messaging Administrator

Summary: Replicated Data Queue Managers (RDQM) for Disaster Recovery

Introducing Disaster Recovery for IBM MQ Replicated Data Queue Manager

RDQM DR Background

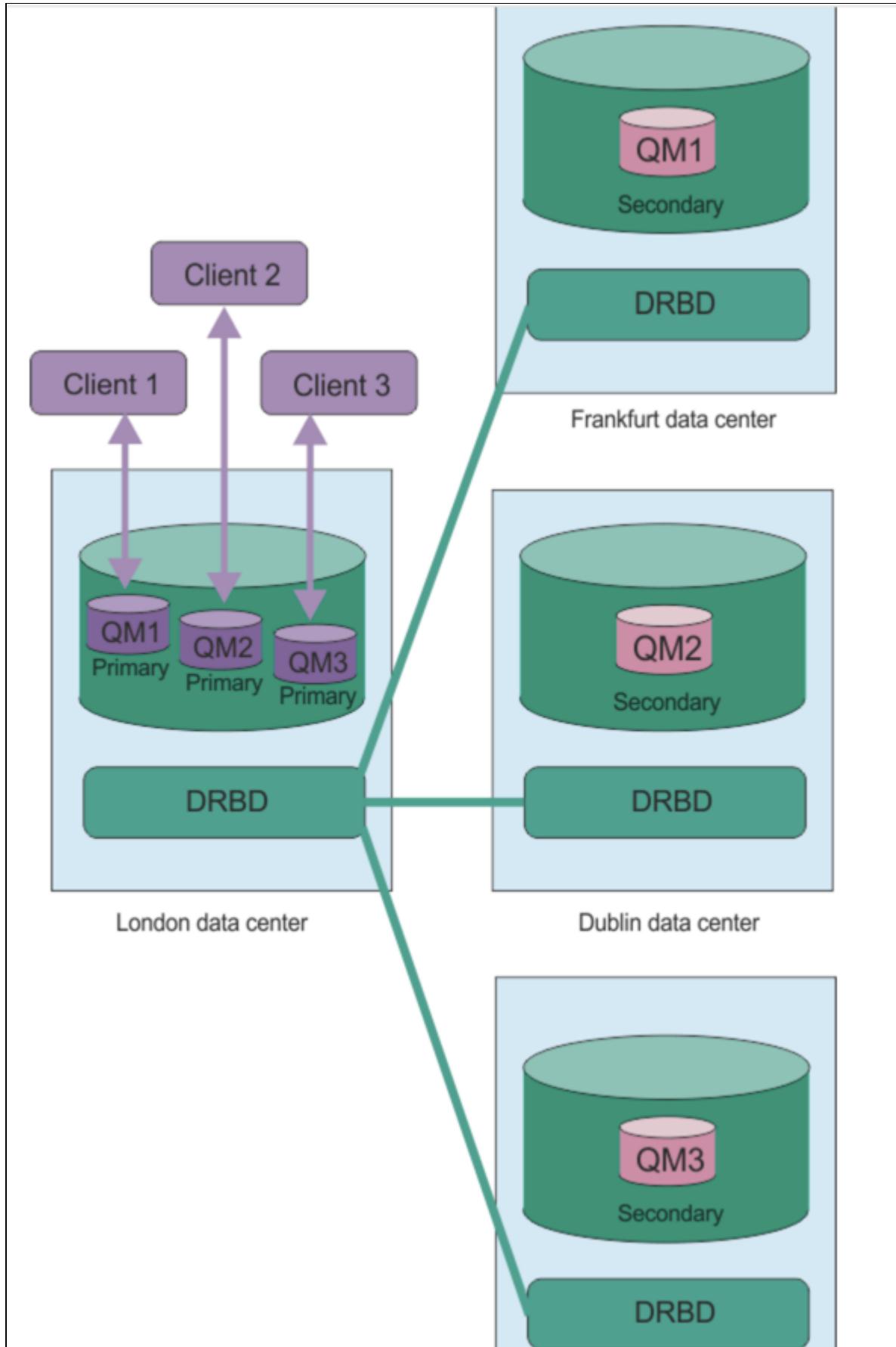
You can create a primary instance of a disaster recovery queue manager running on one server, and a secondary instance of the queue manager on another server that acts as the recovery node. Data is replicated between the queue manager instances. If you lose your primary queue manager, you can manually make the secondary instance into the primary instance and start the queue manager, then resume work from the same place. You cannot start a queue manager while it is in the secondary role. The replication of the data between the two nodes is handled by DRBD.



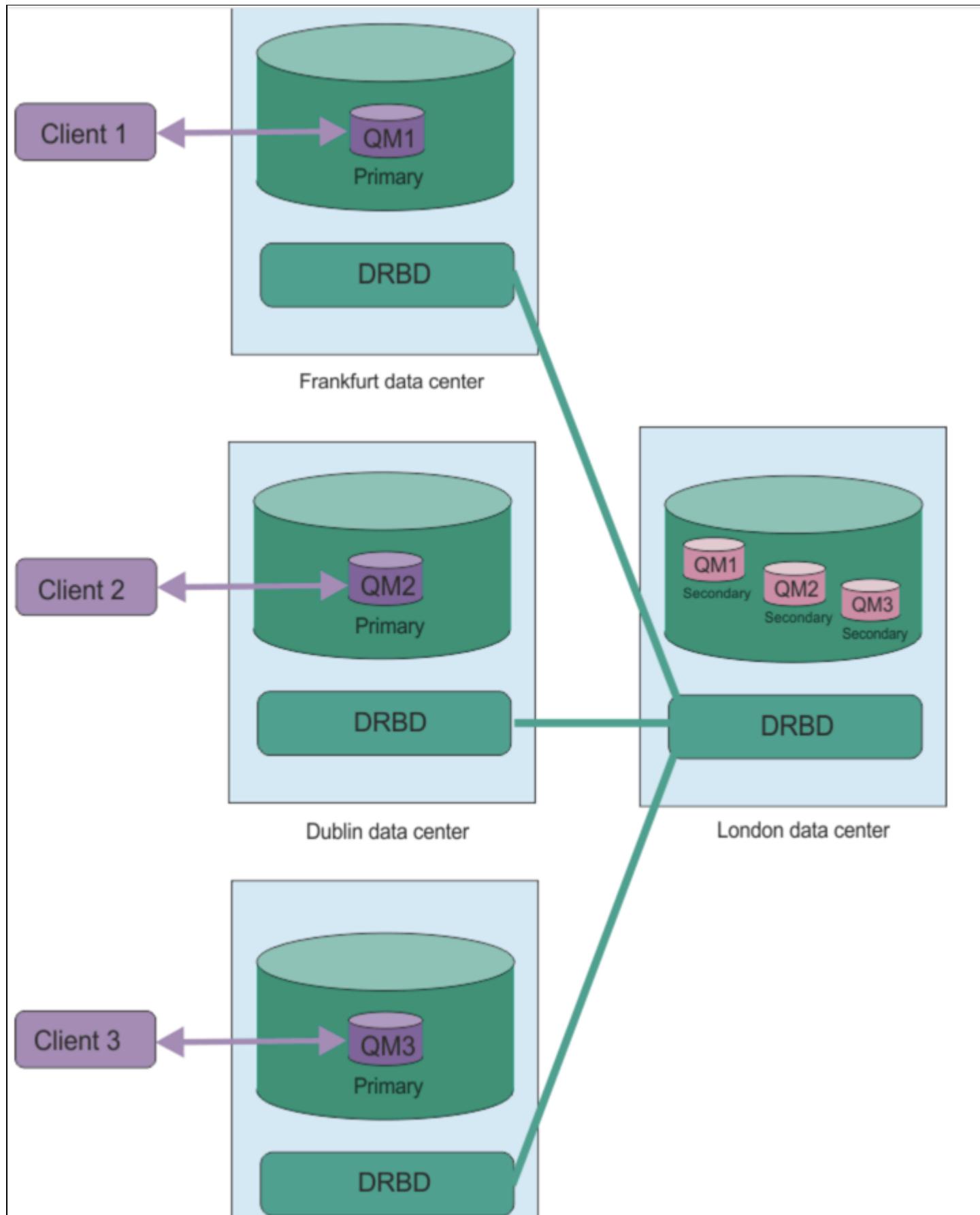
You can choose between synchronous and asynchronous replication of data between primary and secondary queue managers. If you select the asynchronous option, operations such as IBM® MQ PUT or GET complete and return to the application before the event is replicated to the secondary queue manager. Asynchronous replication means that, following a recovery situation, some messaging data might be lost. But the secondary queue manager will be in a consistent state, and able to start running immediately, even if it is started at a slightly earlier part of the message stream.

You cannot add disaster recovery to an existing queue manager, and a queue manager cannot be configured with both RDQM disaster recovery and RDQM high availability.

You can have several pairs of RDQM queue managers running on a number of different servers. For example, you could have six primary DR queue managers running on the same node, while their secondaries are configured on six different nodes in six different data centers.



Equally you could have primary disaster recovery queue managers running on different nodes, while all their secondary disaster recovery queue managers run on the same node. Some example configurations are illustrated in the following diagrams.



Replication, synchronization, and snapshots

While the two nodes in a disaster recovery configuration are connected, any updates to the persistent data for a disaster recovery queue manager are transferred from the primary instance of the queue manager to the secondary instance. This is known as *replication*.

If the network connection between the two nodes is lost, the changes to the persistent data for the primary instance of a queue manager are tracked. When the network connection is restored, a different process is used to get the secondary instance up to speed as quickly as possible. This is known as *synchronization*.

While synchronization is in progress, the data on the secondary instance is in an inconsistent state. A *snapshot* of the state of the secondary queue manager data is taken. If a failure of the main node or the network connection occurs during synchronization, the secondary instance reverts to this snapshot and the queue manager can be started. Any of the updates that happened since the original network failure are lost, however.

RDQM DR Lab Introduction

This lab provides a demonstration of a new approach to Disaster Recovery in MQ on Linux, with the following key features:

- Use of Distributed Replicated Block Device (DRBD) storage rather than network shared storage
- This is still using a Replicated Data Queue Manager (RDQM):
 - Takeover will be manual, not automatic
 - Both asynchronous and synchronous replication is supported
 - An RDQM is active on only one node at any one time
 - Each node can run different active RDQMs
 - An individual DR RDQM is created to use one style of replication and it cannot be changed without recreating the RDQM
 - In 9.0.5 (through 9.1.2) an RDQM can be either HA or DR but not both
 - As only two nodes are involved, it will be possible to get into a split-brain situation; but only if a user has chosen to promote a DR Secondary and start a DR RDQM when the DR network is disconnected and the DR RDQM is still running, or is also started where it was Primary.

The goals for RDQM-DR are:

- Allow an RDQM to be created which is configured to replicate its data to a single Secondary instance at a given IP address
 - Asynchronous replication is supported provided the latency is no more than 50ms for a round trip

time

- Synchronous replication is subject to the same 5ms limits on latency as it is for HA
- Allow manual control of when a DR Secondary becomes a DR Primary and can then run the RDQM

In this lab, instructions are provided to show the setup for both.

Lab environment

1. 2 RHEL 7.7 x86_64 systems running in Skytap:

- dr1 - This will be our primary node
- dr2 - This will be a secondary node

Note: There are four additional VMs in the Skytap template which are not used; dr3, rdqm1, rdqm2, and rdqm3 should be suspended or powered off.

2. Network interfaces:

| Interface Purpose | Interface Name | dr1 (Primary node) | dr2 (Secondary node) |
|-------------------|----------------|--------------------|----------------------|
| Administration | ens34 | 10.0.0.1 | 10.0.0.2 |
| DR Replication | ens36 | 10.0.2.1 | 10.0.2.2 |
| MQ Fixed IP | ens35 | 10.0.1.1 | 10.0.1.2 |
| MQ Floating IP | | 10.0.1.10 | 10.0.1.10 |

DR Replication - for synchronous / asynchronous data replication (the higher the bandwidth the better and the lower the latency the better)

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.

- Although not required for this Lab, the following Pacemaker dependencies required for RDQM HA 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, if you are using some other installation, then additional packages may be needed:
 - cifs-utils
 - gnutls

- libcggroup
- libtool-ltdl
- lvm2
- net-snmp-libs
- nfs-utils
- perl-TimeDate
- psmisc
- PyYAML

- Extract and Install MQ 9.1.5

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

- Install the MQ and RDQM code

RDQM is a single feature which now supports HA and/or DR (but not at the same time for a single queue manager). 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.

Firewall (firewalld) enabled, and ports 1500 & 1501 will be defined during the lab.

- 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 want to allow a normal user in the mqm group to create RDQM instances etc., you need to grant the user access to the certain commands via sudo. The user will also need to be part of the mqm group.

You will add the mqm user to the root and haclient group. Then add root, and ibmuser to the mqm and haclient groups.

The following groups set up:

- **mqm** to allow user to run specific MQ 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 |

- 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 have already been completed on each node so at this point you are ready to begin RDQM configuration.

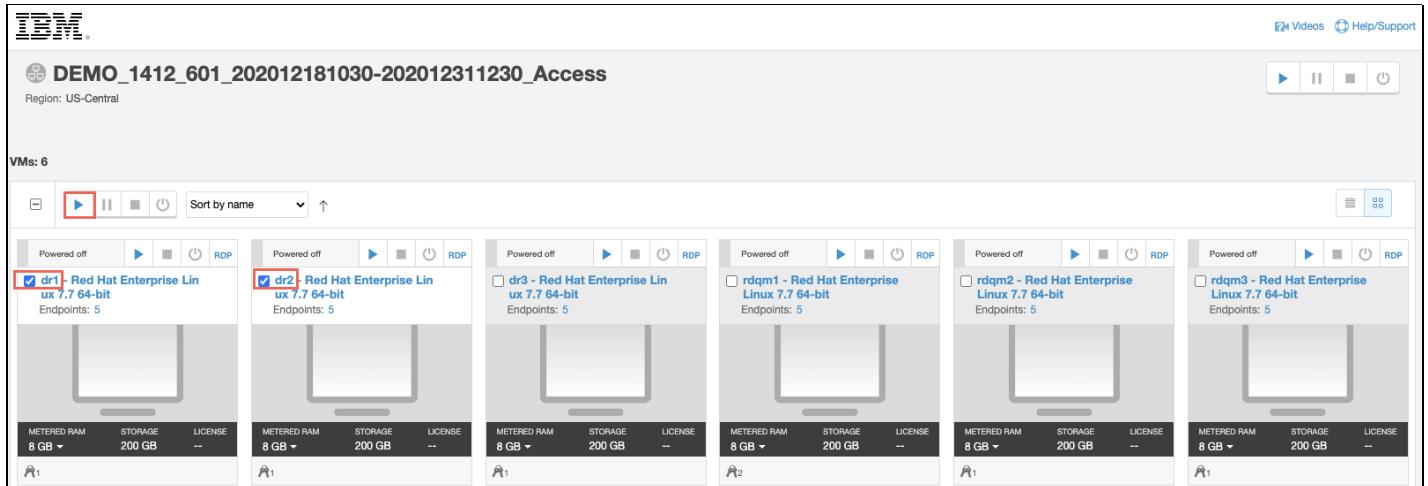
Setup the RHEL image (pre-configured on SkyTap):

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

This template is used for multiple labs and has been configured with the maximum number of VMs that are

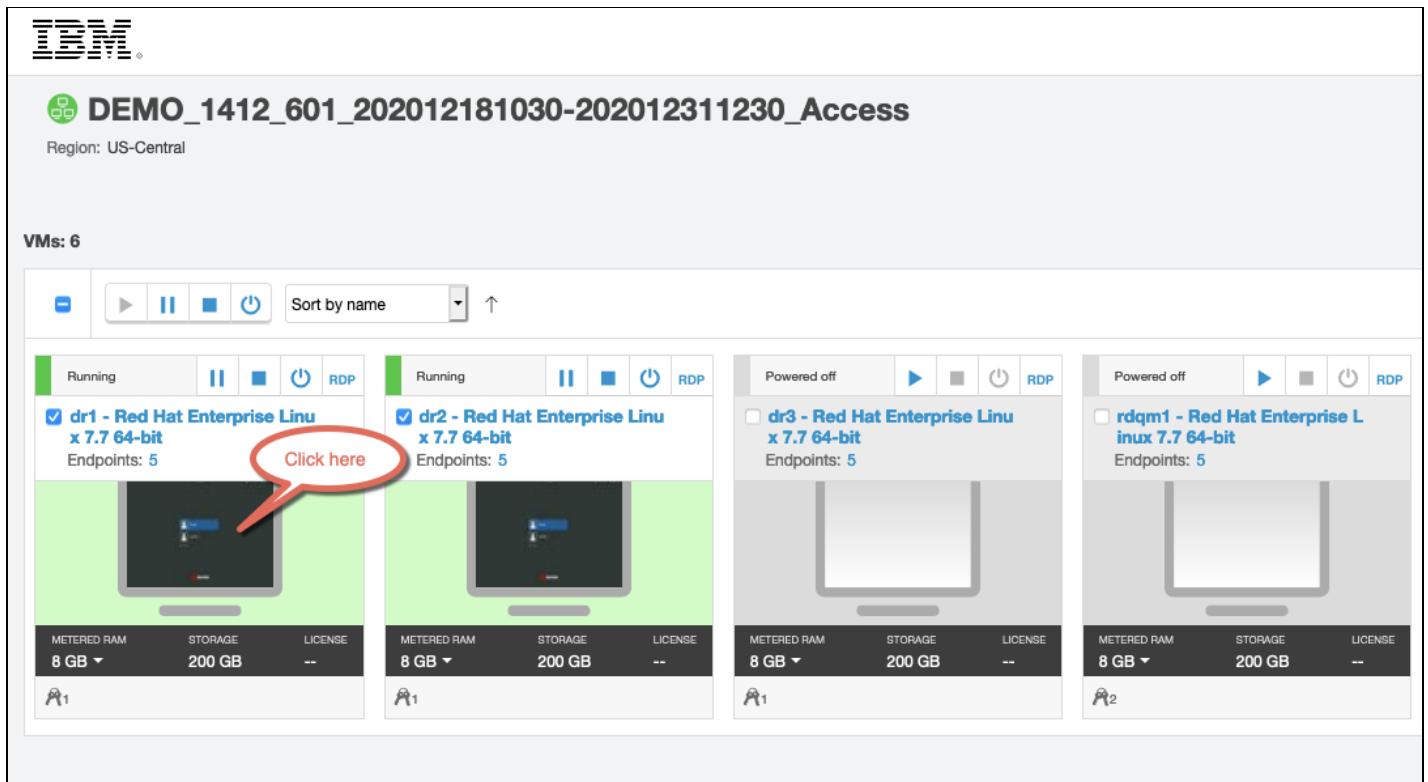
required for all labs. In this lab you will only need dr1 and dr2. The rest of the VMs can remain powered off or suspended.

- Leave the labels checked for *dr1* and *dr2*. Uncheck the labels for all other VMs. Then click the **run** button only for *dr1* and *dr2* to start or resume the VMs.

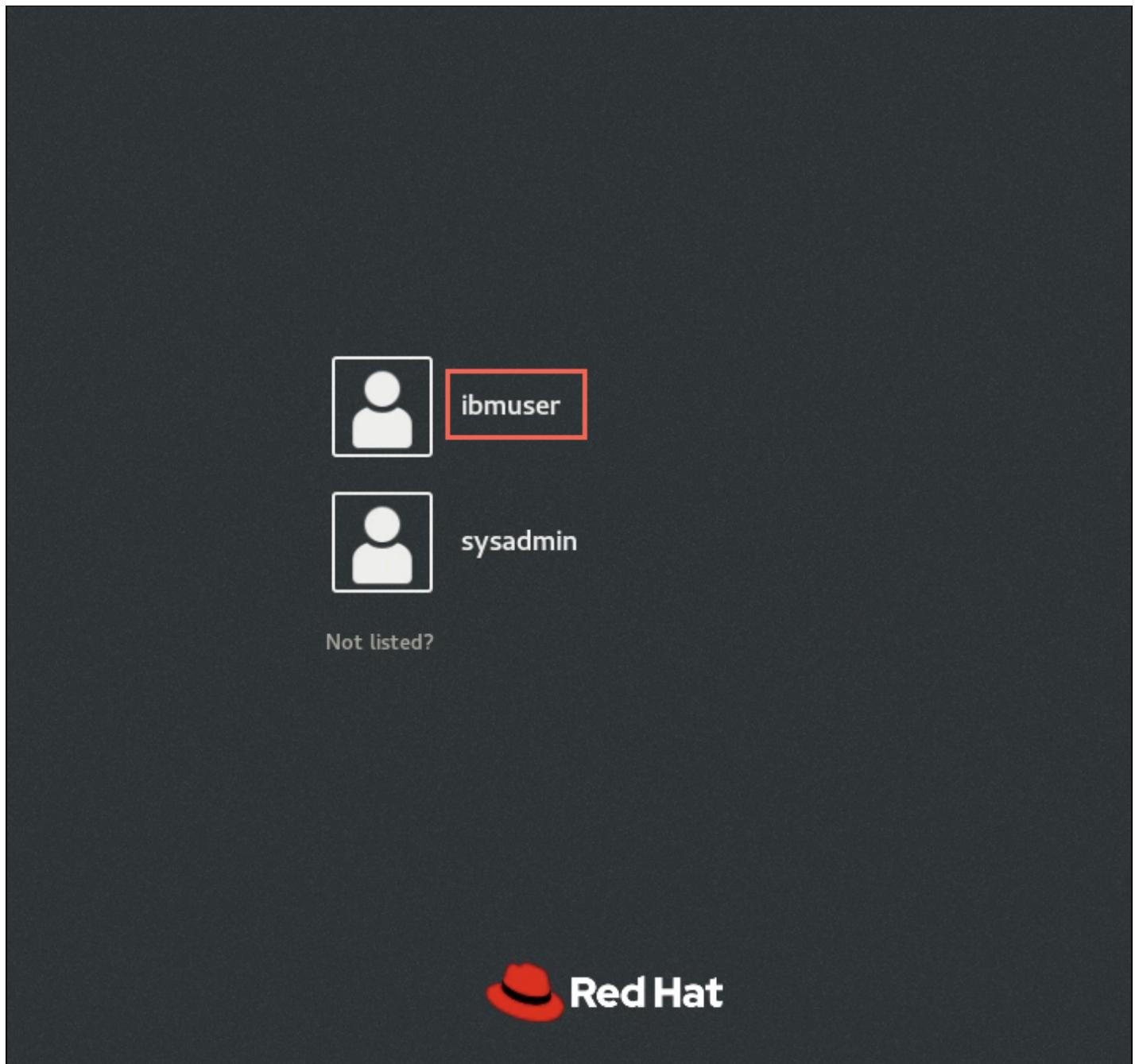


Wait for the monitor icons to turn green, approximately three minutes. Once both are green and running you can proceed to the next step.

- Click the monitor icon for *dr1* which will launch the desktop in another browser tab.



- Log on to VM *dr1* as user **ibmuser**, using password **engageibm**.



Configure RDQM-DR

A primary instance of a disaster recovery queue manager is created on one server. A secondary instance of the same queue manager must be created on another server, which acts as the recovery node. Data is replicated between the queue manager instances. The replication of the data between the two nodes is handled by DRBD.

Unlike the High Availability solution, there is no heartbeat detection between the two nodes. If the primary queue manager node is lost, the secondary instance can be manually made into the primary instance, the queue manager started, and work resumed.

Data replication between primary and secondary queue managers can be done synchronously or asynchronously. If the asynchronous option is selected, operations such as PUT or GET complete and return to the application before the data is replicated to the secondary queue manager. Asynchronous replication means that, following a recovery situation, some messaging data might be lost. But the secondary queue manager will be in a consistent state, and able to start running immediately, even if it is started at a slightly earlier part of the message stream.

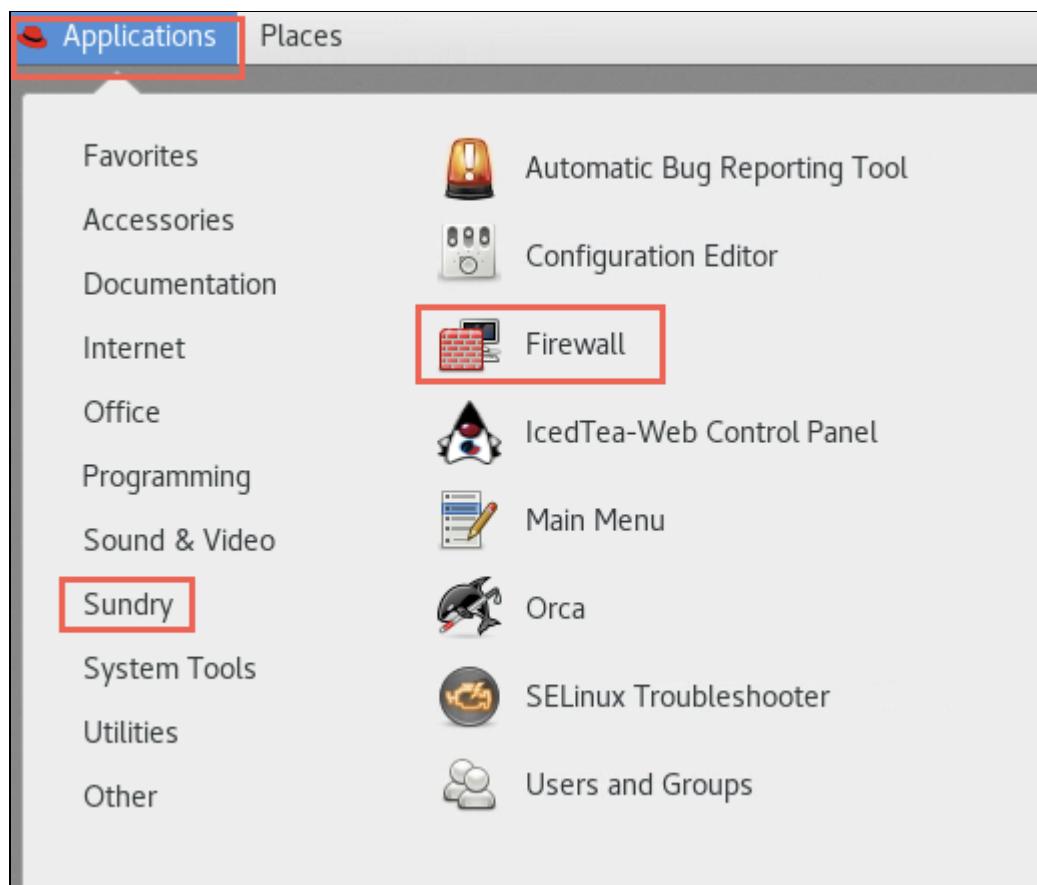
You will configure a DR RDQM that uses asynchronous replication.

Create the DR RDQM

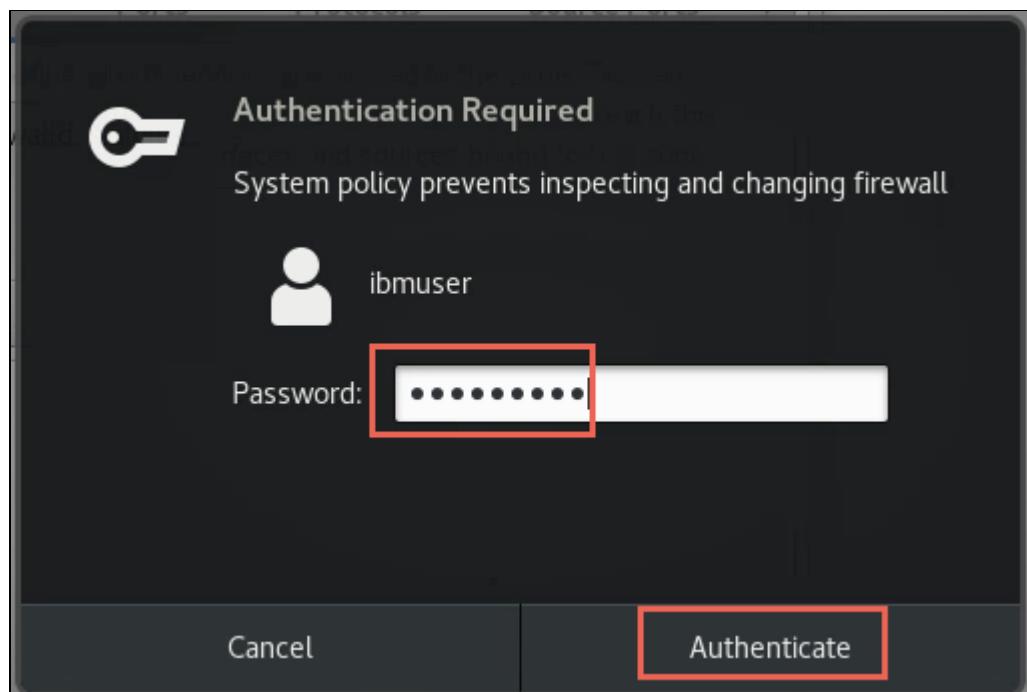
You will create a DR RDQM with asynchronous replication. You must first create a primary RDQM DR queue manager. Then you will create a secondary instance of the same queue manager on another node. The primary and secondary instances must have the same name and be allocated the same amount of storage.

Update firewall rules

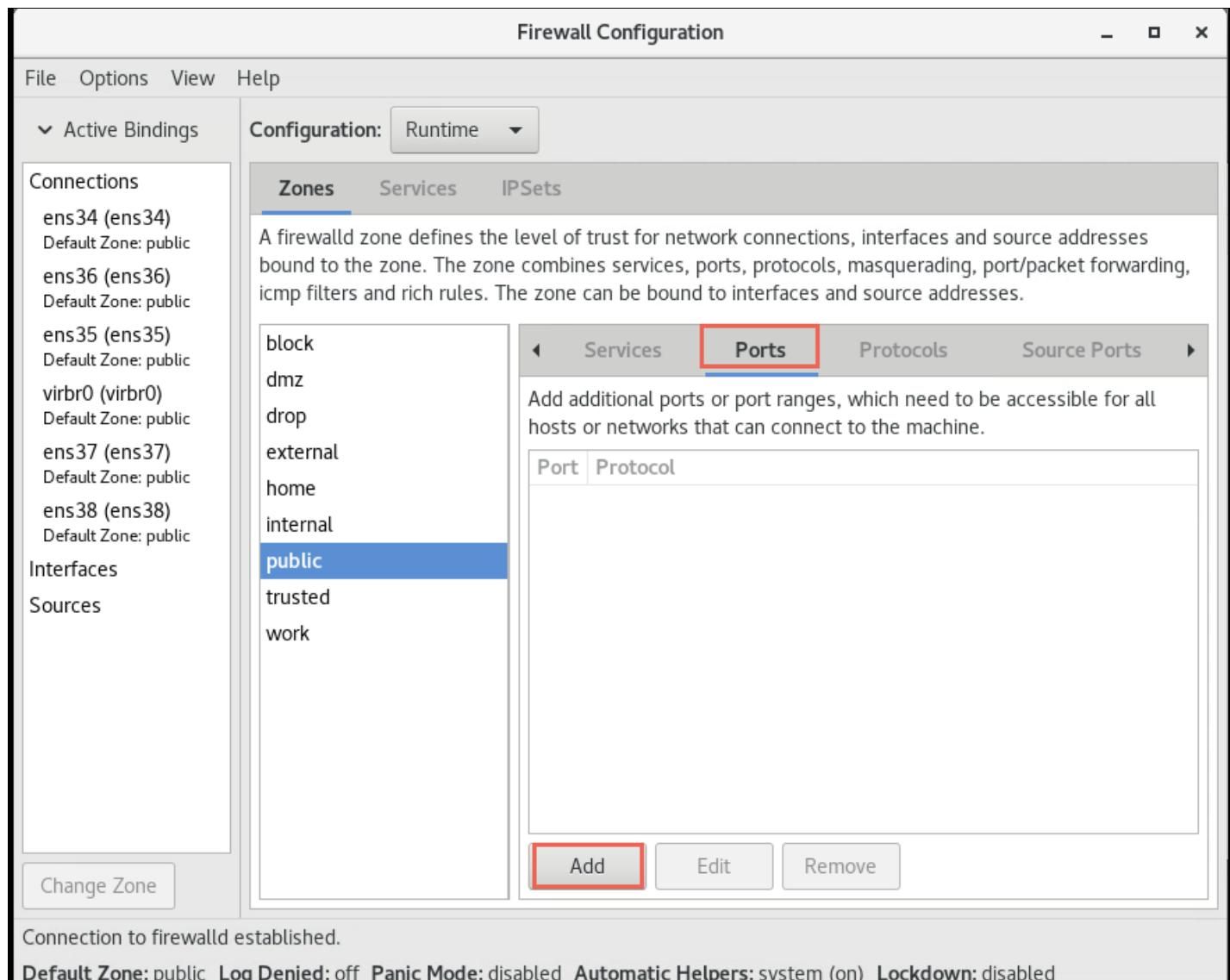
1. You should be logged on as *ibmuser* on **dr1**.
2. On each of the nodes, open the firewall port defined. Open the firewall from the top left of the screen, under *Applications* -> *Sundry* -> *Firewall*.



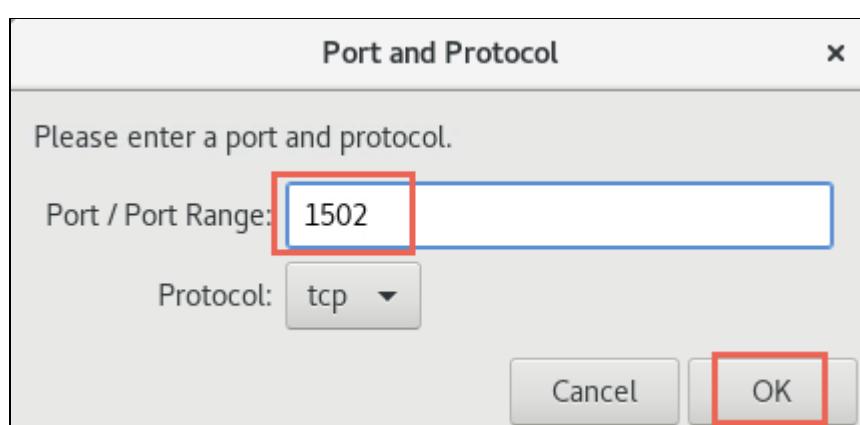
3. Enter the password for ibmuser, **engageibm**, then click *Authenticate*.



4. In the Ports pane, add TCP port 1502.



Results should look like this:



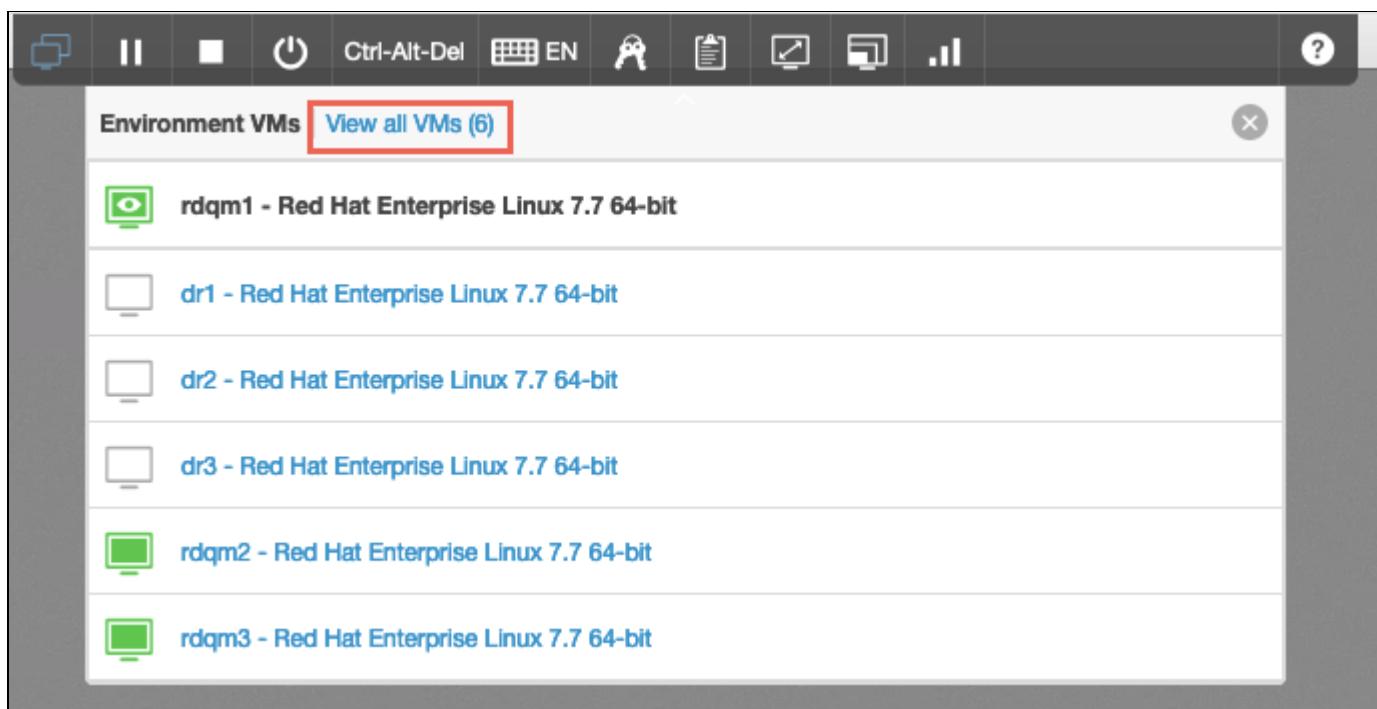
5. You must repeat this on **dr2**.

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



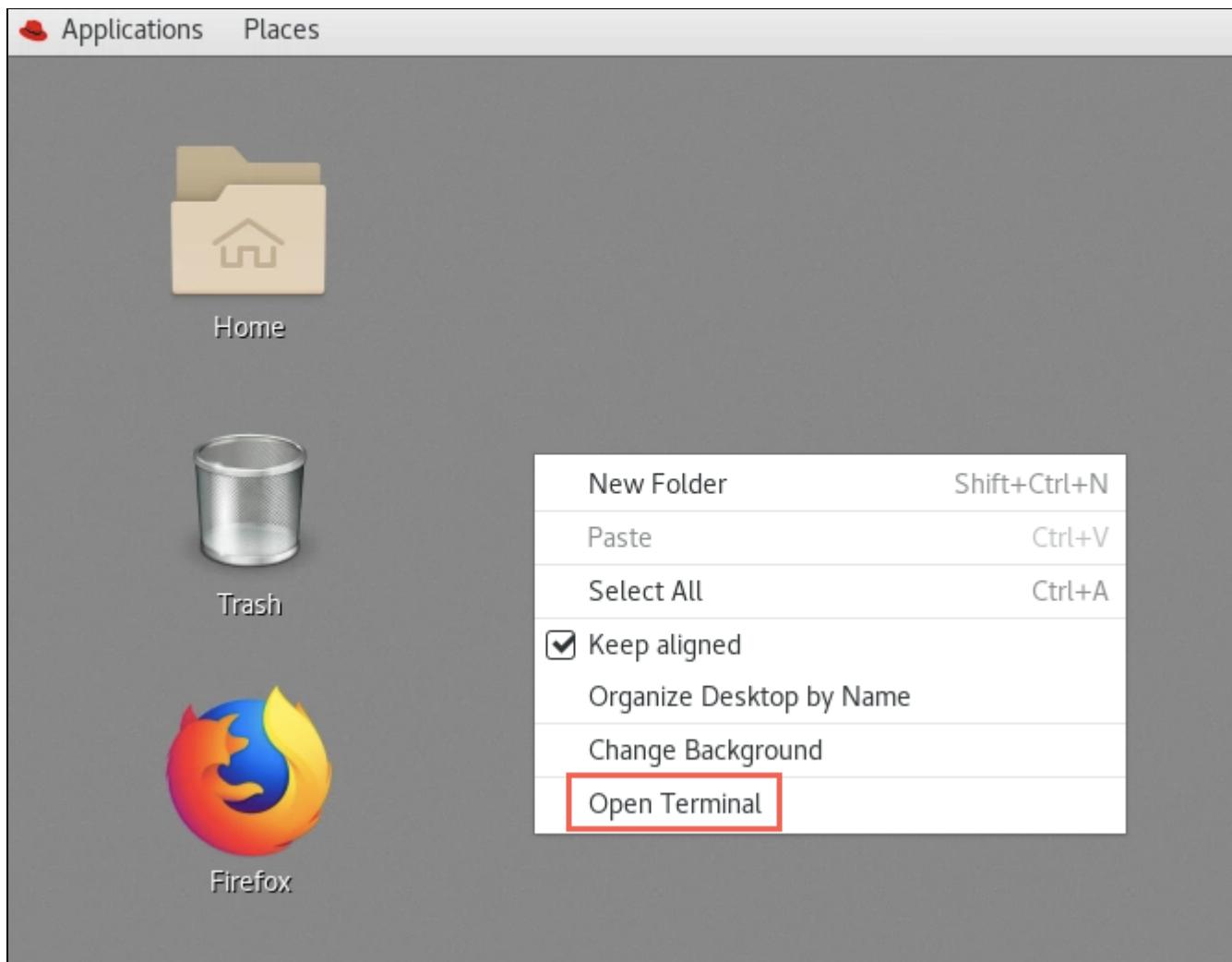
Click **View all VMs (6)** to show the VMs.

Then you can click the monitor icon for **dr2** which will launch the desktop in a new browser tab.



Now open the firewall menu as you did on **dr1** and add port 1502.

6. Right-click on the desktop and select *Open Terminal*



7. Switch to root user with the command:

```
su -
```

Enter the *IBMDem0s!* for root's password.

```
root@dr1:~  
File Edit View Search Terminal Help  
[ibmuser@dr1 ~]$ su -  
Password:  
Last login: Mon Apr  6 12:56:15 EDT 2020 on pts/0  
[root@dr1 ~]# █
```

8. Start firewall by entering the following command.

```
systemctl start firewalld
```

```
root@dr1:~  
File Edit View Search Terminal Help  
[root@dr1 ~]# systemctl start firewalld  
[root@dr1 ~]# █
```

9. Return to **dr1** and repeat the previous step to start the firewall there also.

10. Create a primary queue manager on node **dr1**. It will use asynchronous replication. The local IP for DR replication is 10.0.2.14. The recovery IP used for replication on the secondary instance is 10.0.2.15. Replication will take place using port 7001. The queue manager listener port will be 1502. The queue manager will be QMDR.

In the terminal window, create the primary node:

```
crtmqm -rr p -rt a -rl 10.0.2.14 -ri 10.0.2.15 -rn dr2 -rp 7001 -p 1502 QMDR
```

```
root@dr1:~  
File Edit View Search Terminal Help  
[ibmuser@dr1 ~]$ su -  
Password:  
Last login: Fri Dec 18 16:56:32 EST 2020 on pts/0  
[root@dr1 ~]# systemctl start firewalld  
[root@dr1 ~]# crtmqm -rr p -rt a -rl 10.0.2.14 -ri 10.0.2.15 -rn dr2 -rp 7001 -p 1502 QMDR  
Creating replicated data queue manager configuration.  
IBM MQ queue manager created.  
Directory '/var/mqm/vols/qmdr/qmgr/qmdr' created.  
The queue manager is associated with installation 'Installation1'.  
Creating or replacing default objects for queue manager 'QMDR'.  
Default objects statistics : 84 created. 0 replaced. 0 failed.  
Completing setup.  
Setup completed.  
Issue the following command on node 'dr2' to create the DR secondary queue manager:  
crtmqm -rr s -rt a -rl 10.0.2.15 -ri 10.0.2.14 -rn dr1 -rp 7001 QMDR  
[root@dr1 ~]# rdomstatus -m QMDR
```



Notice at the end, the command needed to create the secondary instance is provided for you.

11. Switch to the **dr2** VM. In the terminal window as *root* create a secondary instance of the queue manager on node **dr2** with the following command:

```
crtmqm -rr s -rt a -rl 10.0.2.15 -ri 10.0.2.14 -rn dr1 -rp 7001 QMDR
```

```
ibmuser@dr2:~  
File Edit View Search Terminal Help  
[ibmuser@dr2 ~]$ sudo crtmqm -rr s -rt a -rl 10.0.2.15 -ri 10.0.2.14 -rn dr1 -rp 7001 QMDR  
Creating replicated data queue manager configuration.  
IBM MQ secondary queue manager created.  
[ibmuser@dr2 ~]$
```

This is the command which was provided for you when you ran the crt mqm command on dr1. Normally you would copy the command from *dr1* and paste it into the terminal window on *dr2*.

12. Check the status on both nodes with the command to ensure they are correct. Use the following command on both nodes. You should be using the terminal where you are logged in as root so you can drop the sudo.

```
rdqmstatus -m QMDR
```

On dr2:

The screenshot shows a terminal window with the title bar "root@dr2:~". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command [root@dr2 ~]# rdqmstatus -m QMDR is highlighted with a red box. The output shows the following information:

| Parameter | Value |
|-----------------------|-------------------|
| Queue manager status: | Ended immediately |
| DR role: | Secondary |
| DR status: | Normal |
| DR type: | Asynchronous |
| DR port: | 7001 |
| DR local IP address: | 10.0.2.15 |
| DR remote IP address: | 10.0.2.14 |

On dr1:

The screenshot shows a terminal window with the title bar "root@dr1:~". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command [root@dr1 ~]# rdqmstatus -m QMDR is highlighted with a red box. The output shows the following information:

| Parameter | Value |
|----------------------------------|---------------------------------|
| Queue manager status: | Ended immediately |
| Queue manager file system: | 58MB used, 2.9GB allocated [2%] |
| DR role: | Primary |
| DR status: | Synchronization in progress |
| DR type: | Asynchronous |
| DR port: | 7001 |
| DR local IP address: | 10.0.2.14 |
| DR remote IP address: | 10.0.2.15 |
| DR synchronization progress: | 80.54% |
| DR estimated time of completion: | 2020-12-19 17:32:46 |

13. Issue the command again until it shows synchronization is complete. When initial synchronization has completed, it should look similar to the following:

```
root@dr1:~  
File Edit View Search Terminal Help  
[root@dr1 ~]# rdqmstatus -m QMDR  
Queue manager status: Ended immediately  
Queue manager file system: 58MB used, 2.9GB allocated [2%]  
DR role: Primary  
DR status: Normal  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.14  
DR remote IP address: 10.0.2.15  
[root@dr1 ~]#
```

14. On the node with the secondary instance, the output should initially look similar to the following:

```
root@dr2:~  
File Edit View Search Terminal Help  
[root@dr2 ~]# rdqmstatus -m QMDR  
Queue manager status: Ended immediately  
DR role: Secondary  
DR status: Normal  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.15  
DR remote IP address: 10.0.2.14  
[root@dr2 ~]#
```

15. On node **dr2** open a new terminal window as the **ibmuser** user to be used to enter the RDQM commands.
16. On node **dr1** open a new terminal window as the **ibmuser** user and then start the queue manager with the following command:

```
strmqm QMDR
```

```
ibmuser@dr1:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ strmqm QMDR
IBM MQ queue manager 'QMDR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMDR' during the log replay phase.
Log replay for queue manager 'QMDR' complete.
Transaction manager state recovered for queue manager 'QMDR'.
IBM MQ queue manager 'QMDR' started using V9.1.5.0.
[ibmuser@dr1 ~]$
```

17. Now check the status on both nodes to ensure they are correct, using the command:

```
sudo rdqmstatus -m QMDR
```

On **dr1**, the output will initially look similar to the following:

```
ibmuser@dr1:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.00%
Memory: 183MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
DR role: Primary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
[ibmuser@dr1 ~]$
```

As the node with the secondary instance only runs the queue manager when DR is needed, the output will be unchanged and look as it did previously.

```
ibmuser@dr2:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr2 ~]$ rdqmstatus -m QMDR
Queue manager status:
DR role: Secondary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.15
DR remote IP address: 10.0.2.14
[ibmuser@dr2 ~]$
```

Test the DR Secondary

Now that the DR nodes have been set up, you will test the secondary DR queue manager.

Make the Primary instance the Secondary node

Only one node can be the Primary. Therefore, before another node can be designated the Primary, the original Primary needs to be designated the Secondary.

1. On node **dr1**, in ibmuser's terminal, stop the queue manager:

```
endmqm QMDR
```

2. On node **dr1** designate node **dr1**, as the secondary using the rdqmldr command:

```
sudo rdqmldr -m QMDR -s
```

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

Make the Secondary node the Primary instance

Designate the Secondary node as the Primary instance.

1. On the recovery node **dr2**, designate it as the primary instance using the rdqmldr command:

```
sudo rdqmldr -m QMDR -p
```

2. Still on **dr2** start the queue manager:

```
strmqm QMDR
```

```
ibmuser@dr2:~  
File Edit View Search Terminal Help  
[ibmuser@dr2 ~]$ strmqm QMDR  
IBM MQ queue manager 'QMDR' starting.  
The queue manager is associated with installation 'Installation1'.  
5 log records accessed on queue manager 'QMDR' during the log replay phase.  
Log replay for queue manager 'QMDR' complete.  
Transaction manager state recovered for queue manager 'QMDR'.  
IBM MQ queue manager 'QMDR' started using V9.1.5.0.  
ibmuser@dr2:~
```

3. Confirm the status of both nodes:

```
sudo rdqmstatus -m QMDR
```

On node **dr2**:

```
ibmuser@dr2:~$ sudo rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.61%
Memory: 181MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
DR role: Primary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.15
DR remote IP address: 10.0.2.14
[ibmuser@dr2 ~]$
```

On node **dr1**:

```
ibmuser@dr1:~$ sudo rdqmstatus -m QMDR
Queue manager status: Ended immediately
DR role: Secondary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
[ibmuser@dr1 ~]$
```

Provided that channels were defined with a list of alternative connection names specifying the primary and secondary queue managers, then applications will automatically connect to the new primary queue manager.

Make the Primary instance the Primary again

If the loss of the primary node was only temporary, you would want to designate it as the *Primary* instance

again. This would be achieved as described below.

1. On node **dr2** stop the queue manager:

```
endmqm QMDR
```

2. With root access designate node **dr2** as the *Secondary* instance:

```
sudo rdqmldr -m QMDR -s
```

The screenshot shows a terminal window with a light gray background and a dark gray header bar. The header bar contains the text "ibmuser@dr2:~". Below the header is a menu bar with options: File, Edit, View, Search, Terminal, Help. The main area of the terminal shows the following command sequence:

```
File Edit View Search Terminal Help
[ibmuser@dr2 ~]$ endmqm QMDR
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
[ibmuser@dr2 ~]$ sudo rdqmldr -m QMDR -s
Queue manager 'QMDR' has been made the DR secondary on this node.
[ibmuser@dr2 ~]$
```

The command `endmqm QMDR` is highlighted with a red rectangle. The command `sudo rdqmldr -m QMDR -s` is also highlighted with a red rectangle.

3. On the primary node **dr1**, with root access, designate it as the *Primary* instance again:

```
sudo rdqmldr -m QMDR -p
```

4. On the primary node **dr1** restart the queue manager:

```
strmqm QMDR
```

5. Confirm the status of both nodes:

```
sudo rdqmstatus -m QMDR
```

On dr1:

```
ibmuser@dr1:~  
File Edit View Search Terminal Help  
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMDR  
Queue manager status: Running  
CPU: 0.00%  
Memory: 181MB  
Queue manager file system: 58MB used, 2.9GB allocated [2%]  
DR role: Primary  
DR status: Normal  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.14  
DR remote IP address: 10.0.2.15  
[ibmuser@dr1 ~]$ █
```

On dr2:

```
ibmuser@dr2:~  
File Edit View Search Terminal Help  
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR  
Queue manager status: Ended immediately  
DR role: Secondary  
DR status: Normal  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.15  
DR remote IP address: 10.0.2.14  
[ibmuser@dr2 ~]$ █
```

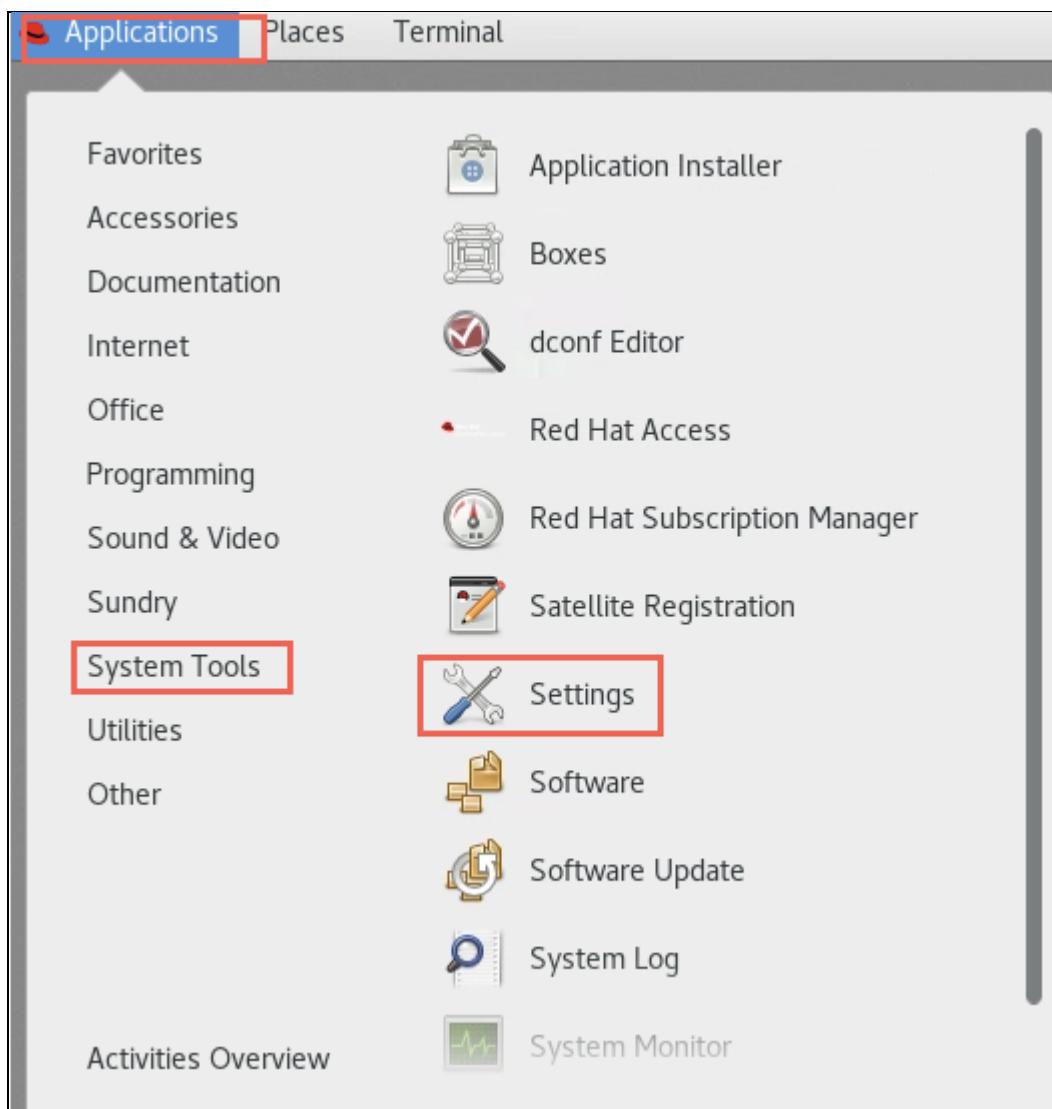
Replace node that was running as DR Primary

Suppose the loss of the primary node was due to a failure, which resulted in the node having to be replaced. You would want to replace the primary node while the queue manager runs on the secondary node. Then restore the original disaster recovery configuration.

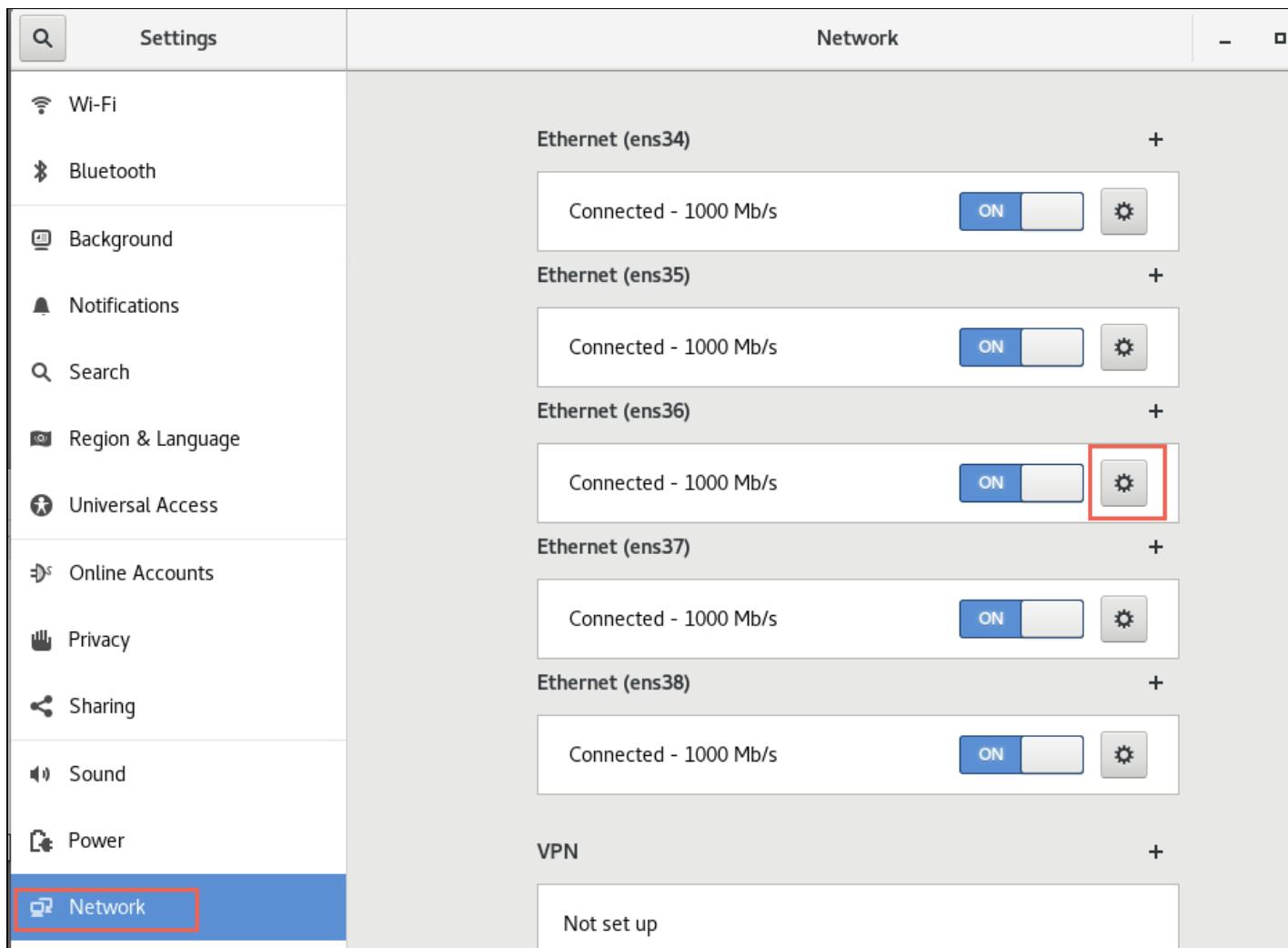
Simulate the loss of the Primary node

Although the node has not been lost, you will simulate it by disabling the DR Replication Network adapter and deleting the queue manager.

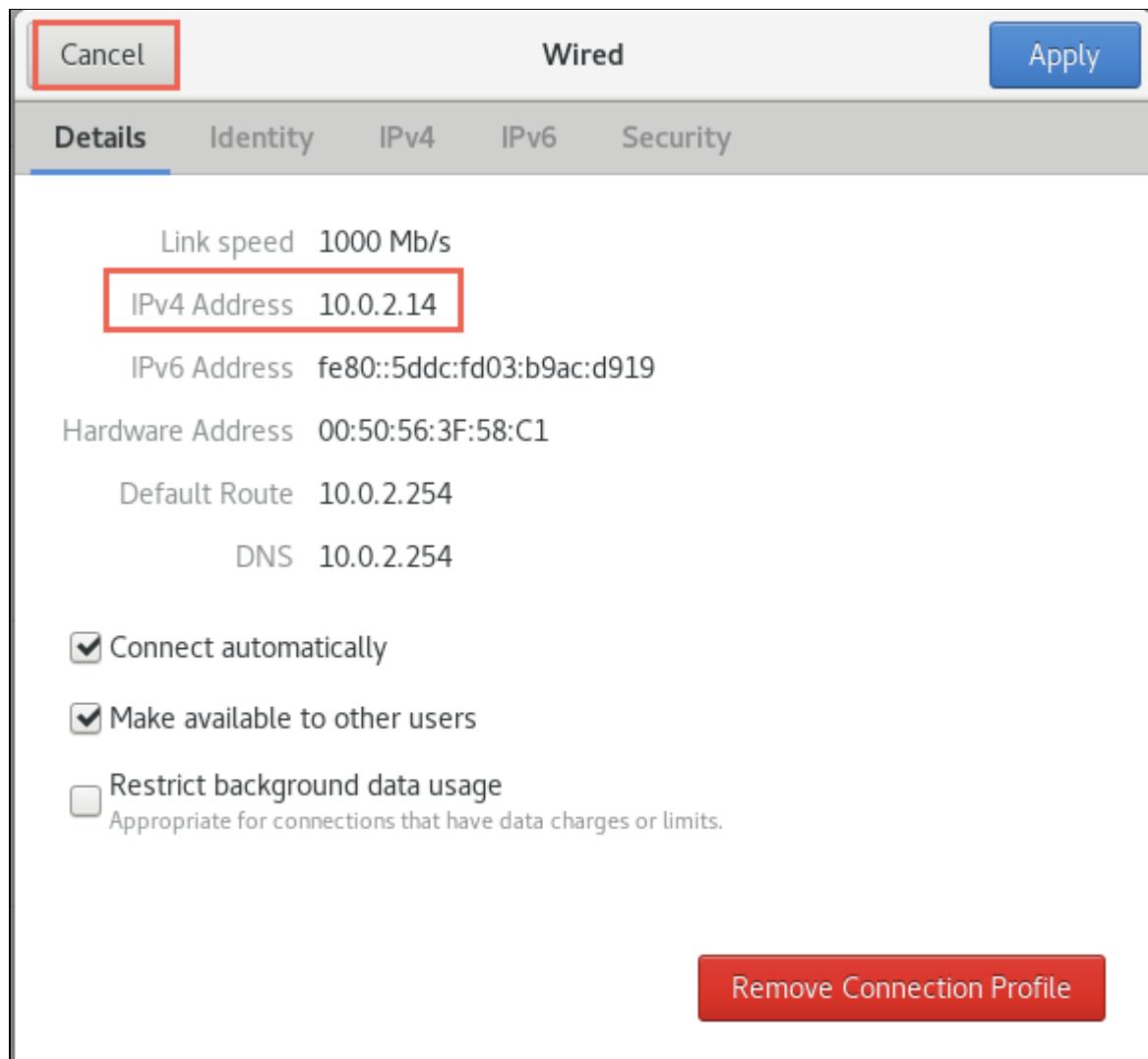
1. On node **dr1**, navigate to **Applications -> System Tools -> Settings**:



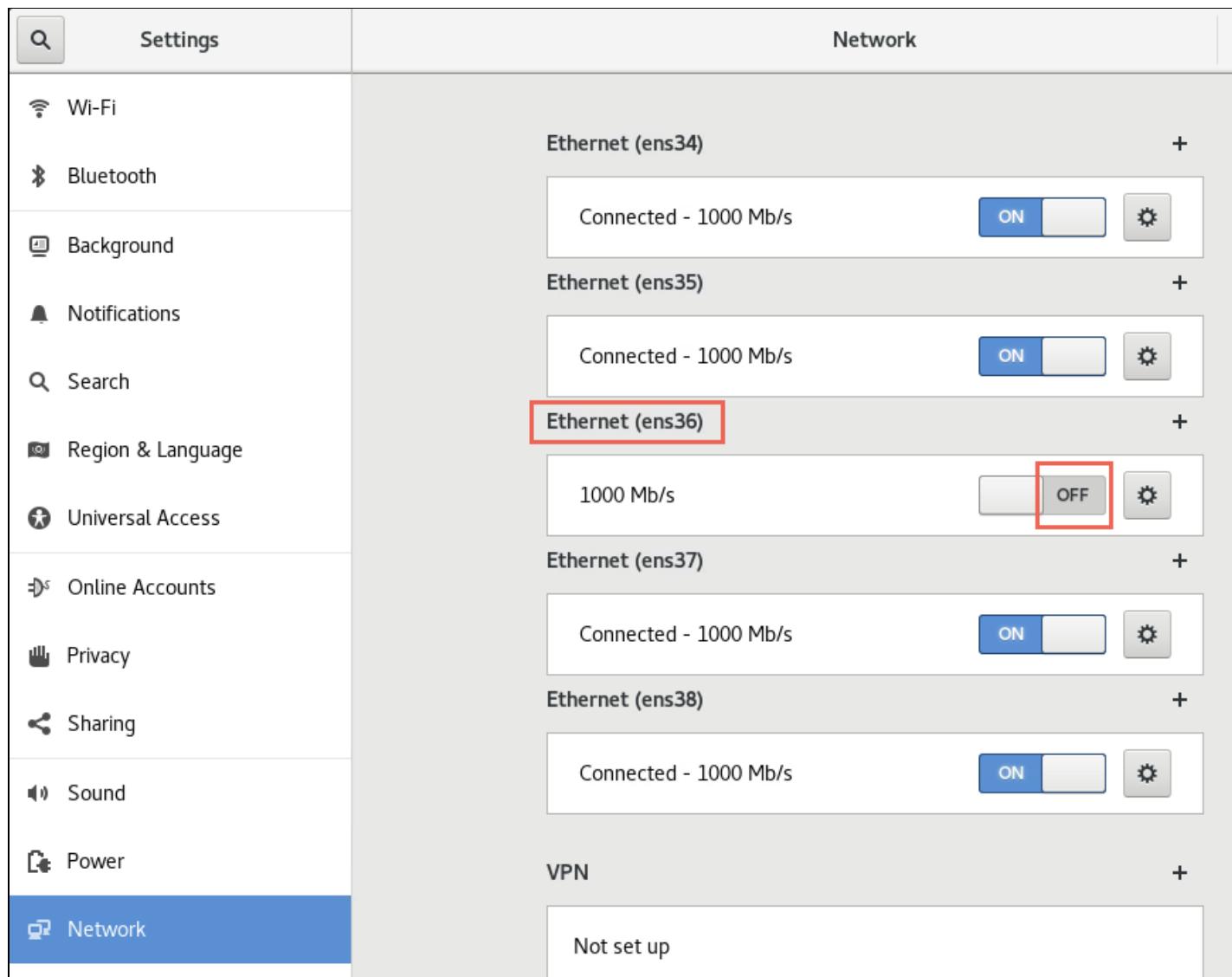
2. On the left list of entries, scroll down and select **Network**.
3. Click the **Settings** gear symbol on the **ens36** network adapter, to verify that it is the DR Replication adapter (IP address 10.0.2.14):



4. In the window that opens, validate the IP address of **10.0.2.14**, then click **Cancel**:



5. For the **ens36** adapter, click the button to switch it *off*.



Note: You can leave the Network window open as you will need it in a later step.

6. On node **dr1**, stop the queue manager:

```
endmqm QMDR
```

7. With root access remove the queue manager:

```
sudo dltmqm QMDR
```

```
ibmuser@dr1:~$ File Edit View Search Terminal Help  
[ibmuser@dr1 ~]$ endmqm QMDR  
Quiesce request accepted. The queue manager will stop when all outstanding work  
is complete.  
[ibmuser@dr1 ~]$ sudo dltmqm QMDR
```

Make the Secondary node the Primary instance

Designate the Secondary node as the Primary instance.

1. On the recovery node **dr2** with root access designate it as the primary instance:

```
sudo rdqmldr -m QMDR -p
```

2. Start the queue manager:

```
strmqm QMDR
```

```
ibmuser@dr2:~  
File Edit View Search Terminal Help  
[ibmuser@dr2 ~]$ sudo rdqmldr -m QMDR -p  
Queue manager 'QMDR' has been made the DR primary on this node.  
[ibmuser@dr2 ~]$ strmqm QMDR  
IBM MQ queue manager 'QMDR' starting.  
The queue manager is associated with installation 'Installation1'.  
57 log records accessed on queue manager 'QMDR' during the log replay phase.  
Log replay for queue manager 'QMDR' complete.  
Transaction manager state recovered for queue manager 'QMDR'.  
IBM MQ queue manager 'QMDR' started using V9.1.5.0.  
[ibmuser@dr2 ~]$
```

3. Confirm the status of both nodes:

```
sudo rdqmstatus -m QMDR
```

On **dr2**:

```
ibmuser@dr2:~  
File Edit View Search Terminal Help  
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR  
Queue manager status: Running  
CPU: 0.00%  
Memory: 181MB  
Queue manager file system: 58MB used, 2.9GB allocated [2%]  
DR role: Primary  
DR status: Remote unavailable  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.15  
DR remote IP address: 10.0.2.14  
DR out of sync data: 612KB  
[ibmuser@dr2 ~]$
```

As there is no longer a queue manager defined on node **dr1**, the output should look similar to the following:

```
ibmuser@dr1:~$  
File Edit View Search Terminal Help  
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMDR  
The queue manager does not exist.  
[ibmuser@dr1 ~]$
```

Add the new Primary node into the DR configuration

For the replacement node to be brought back into the DR configuration, it must assume the identity of the failed node – the name and IP address must therefore be the same.

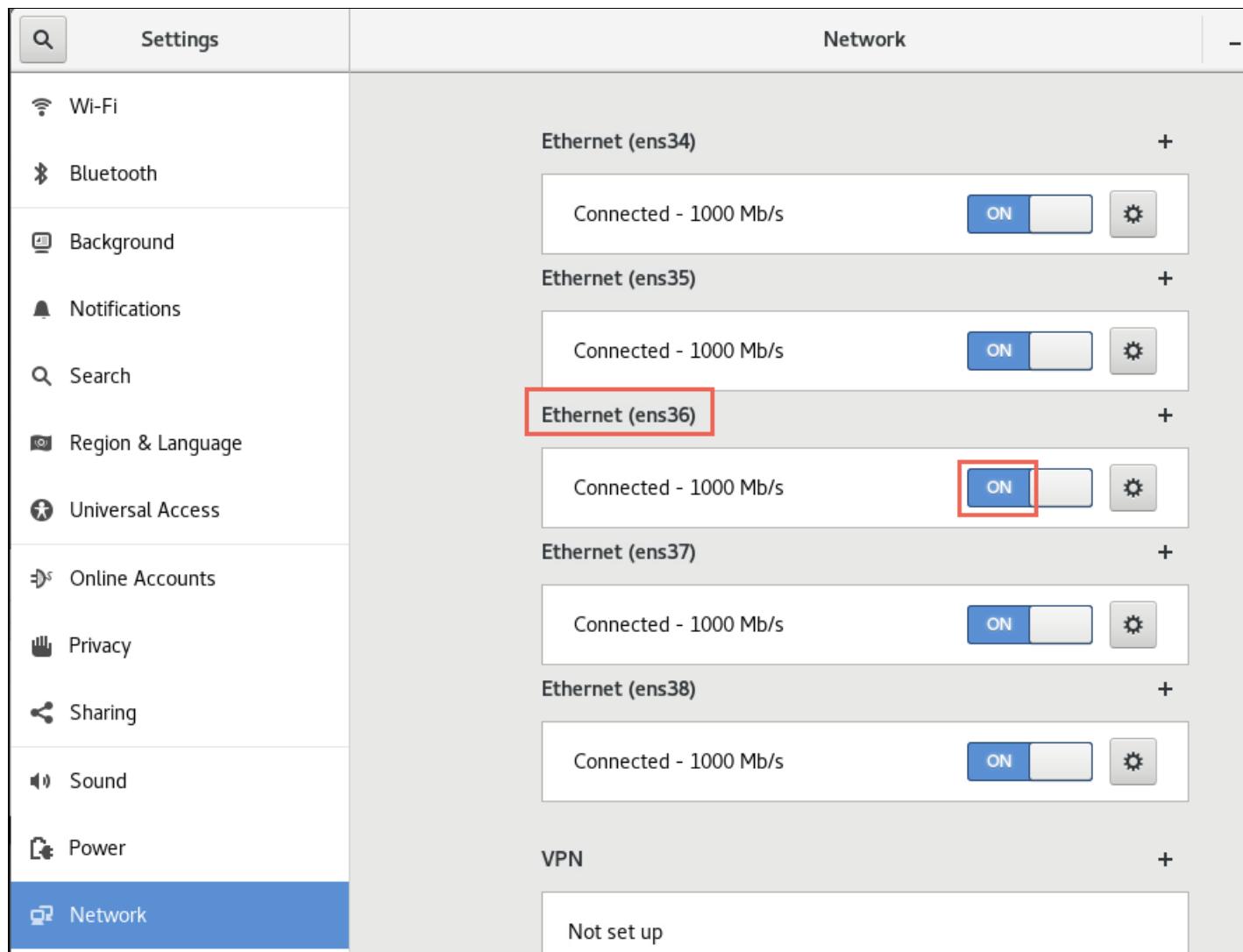
1. Remember, *rdqmldr* commands require root authority. You will determine the command that needs to be run on the new Primary node. On node **dr2**, run the command:

```
sudo rdqmldr -m QMDR -d
```

The output should look similar to the following:

```
ibmuser@dr2:~$  
File Edit View Search Terminal Help  
[ibmuser@dr2 ~]$ sudo rdqmldr -m QMDR -d  
Issue the following command on node 'dr1' to create the DR secondary queue  
manager:  
crtmqm -rr s -rl 10.0.2.14 -ri 10.0.2.15 -rn dr2 -rp 7001 QMDR  
[ibmuser@dr2 ~]$
```

2. Restart the DR Replication network interface. You should still have the network settings window open on node **dr1**. If not, go to **Applications -> System Tools -> Settings -> Network**. Click the button to turn on **ens36**.



3. Copy the command (as highlighted above) into the command line of the new Primary node, **dr1**, to run it:

```
crtmqm -rr s -rl 10.0.2.14 -ri 10.0.2.15 -rn dr2 -rp 7001 QMDR
```

The output should look similar to the following:

```
ibmuser@dr1:~
```

File Edit View Search Terminal Help

```
[ibmuser@dr1 ~]$ crtqm -rr s -rl 10.0.2.14 -ri 10.0.2.15 -rn dr2 -rp 7001 QMDR
Creating replicated data queue manager configuration.
IBM MQ secondary queue manager created.
Command '/opt/mqm/bin/crtqm' run with sudo.
[ibmuser@dr1 ~]$
```

4. Check the status of the synchronization on both nodes:

```
sudo rdqmstatus -m QMDR
```

On dr1:

```
ibmuser@dr1:~
```

File Edit View Search Terminal Help

```
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Ended immediately
DR role: Secondary
DR status: Synchronization in progress
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
DR synchronization progress: 47.92%
DR estimated time of completion: 2020-12-25 10:40:47
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Ended immediately
DR role: Secondary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
[ibmuser@dr1 ~]$
```

On dr2:

```
ibmuser@dr2:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.00%
Memory: 181MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
DR role: Primary
DR status: Synchronization in progress
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.15
DR remote IP address: 10.0.2.14
DR synchronization progress: 77.57%
DR estimated time of completion: 2020-12-25 10:40:49
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.00%
Memory: 181MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
DR role: Primary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.15
DR remote IP address: 10.0.2.14
[ibmuser@dr2 ~]$
```

Wait for synchronization to complete - *DR status = Normal* before continuing.

Restore the original DR configuration

To restore the original DR configuration, you would want to designate the Primary node as the Primary instance again.

1. When the initial synchronization is complete on the primary node **dr1**, you can designate the secondary node as the secondary instance again. End the queue manager on node **dr2**:

```
endmqm QMDR
```

2. Designate node **dr2** as the secondary instance again:

```
sudo rdqmldr -m QMDR -s
```

ibmuser@dr2:~

```
File Edit View Search Terminal Help
[ibmuser@dr2 ~]$ endmqm QMDR
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
[ibmuser@dr2 ~]$ sudo rdqmldr -m QMDR -s
Queue manager 'QMDR' has been made the DR secondary on this node.
[ibmuser@dr2 ~]$
```

3. On node **dr1**, designate it as the Primary instance again:

```
sudo rdqmldr -m QMDR -p
```

4. Start the queue manager on the primary node:

```
strmqm QMDR
```

ibmuser@dr1:~

```
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ sudo rdqmldr -m QMDR -p
Queue manager 'QMDR' has been made the DR primary on this node.
[ibmuser@dr1 ~]$ strmqm QMDR
IBM MQ queue manager 'QMDR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMDR' during the log replay phase.
Log replay for queue manager 'QMDR' complete.
Transaction manager state recovered for queue manager 'QMDR'.
IBM MQ queue manager 'QMDR' started using V9.1.5.0.
```

5. Confirm the status of both nodes:

```
sudo rdqmstatus -m QMDR
```

On dr1:

```
ibmuser@dr1:~  
File Edit View Search Terminal Help  
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMDR  
Queue manager status: Running  
CPU: 0.19%  
Memory: 181MB  
Queue manager file system: 58MB used, 2.9GB allocated [2%]  
DR role: Primary  
DR status: Normal  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.14  
DR remote IP address: 10.0.2.15  
[ibmuser@dr1 ~]$
```

On dr2:

```
ibmuser@dr2:~  
File Edit View Search Terminal Help  
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR  
Queue manager status: Ended immediately  
DR role: Secondary  
DR status: Normal  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.15  
DR remote IP address: 10.0.2.14  
[ibmuser@dr2 ~]$
```

Replace node that was running a DR Secondary instance

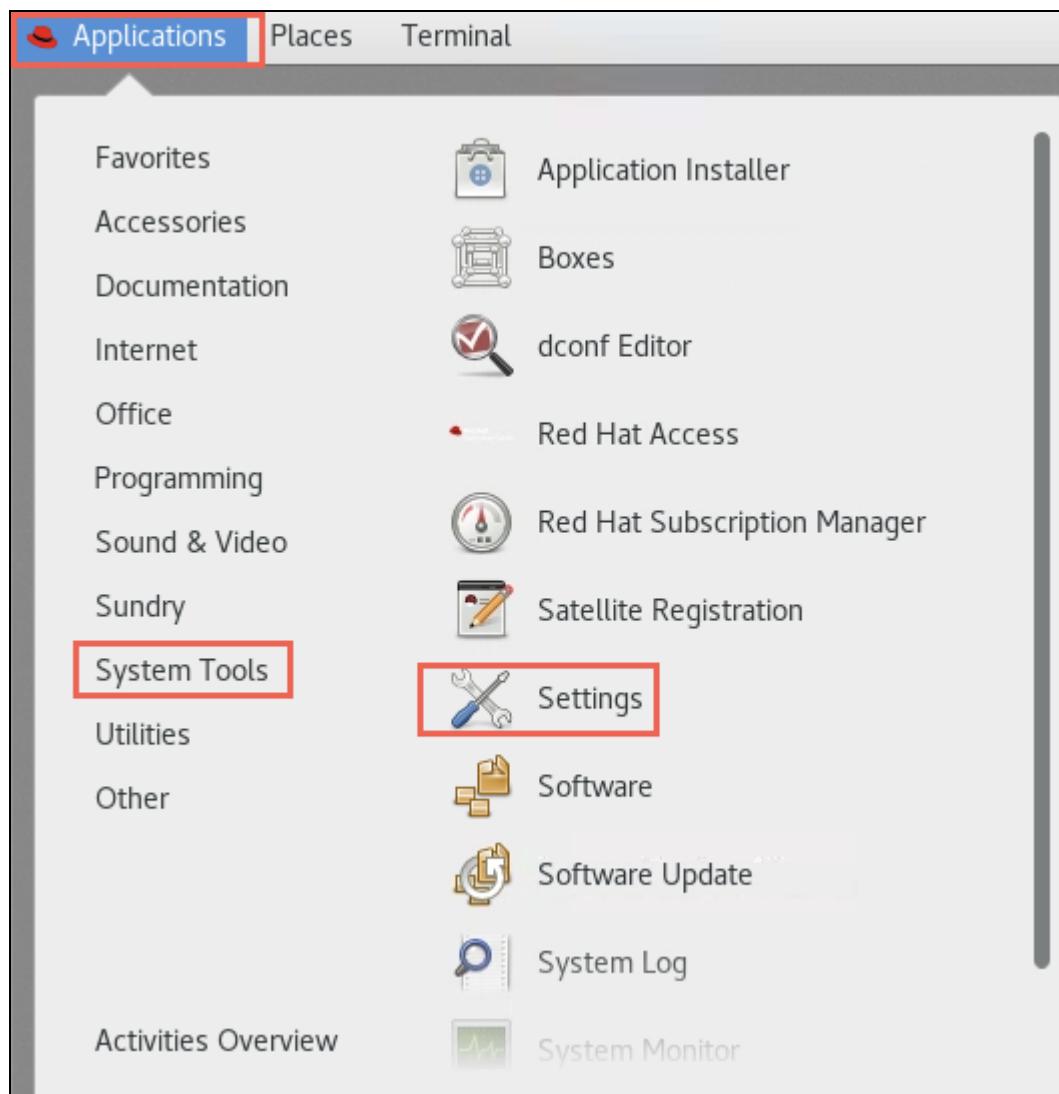
If it is a secondary node that needs to be replaced, you would just replace it and restore it to the original disaster recovery configuration.

Add the new Secondary node into the DR configuration

To simulate this, there is no need to change the DR designations prior to the Secondary node being replaced. You will simply disable the DR Replication Network adapter.

For the replacement node to be brought back into the DR configuration, again it must assume the identity of the failed node – the name and IP address must be the same.

1. On node **dr2**, go to **Applications -> System Tools -> Settings**.



Select **Network**.

2. The DR Replication adapter (IP address 10.0.2.15) is the **ens36** adapter. Click the button to switch it off.

The screenshot shows the 'Network' settings window. On the left sidebar, the 'Network' option is highlighted with a blue bar. The main area displays network interfaces: Ethernet (ens34), Ethernet (ens35), Ethernet (ens36), Ethernet (ens37), and Ethernet (ens38). Each interface has a status box ('Connected - 1000 Mb/s'), an 'ON/OFF' switch (set to ON for ens34, ens35, ens37, and ens38; set to OFF for ens36), and a gear icon for settings. Below these is a 'VPN' section with a 'Not set up' message.

Note: You can leave the Network window open as you will need it in a later step.

3. Still on **dr2** with root access delete the queue manager.

```
sudo dltmqm QMDR
```

```
ibmuser@dr2:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr2 ~]$ sudo dltsqm QMDR
IBM MQ queue manager 'QMDR' deleted.
The DR replicated data queue manager might also need to be deleted on remote
node 'dr1'.
[ibmuser@dr2 ~]$
```

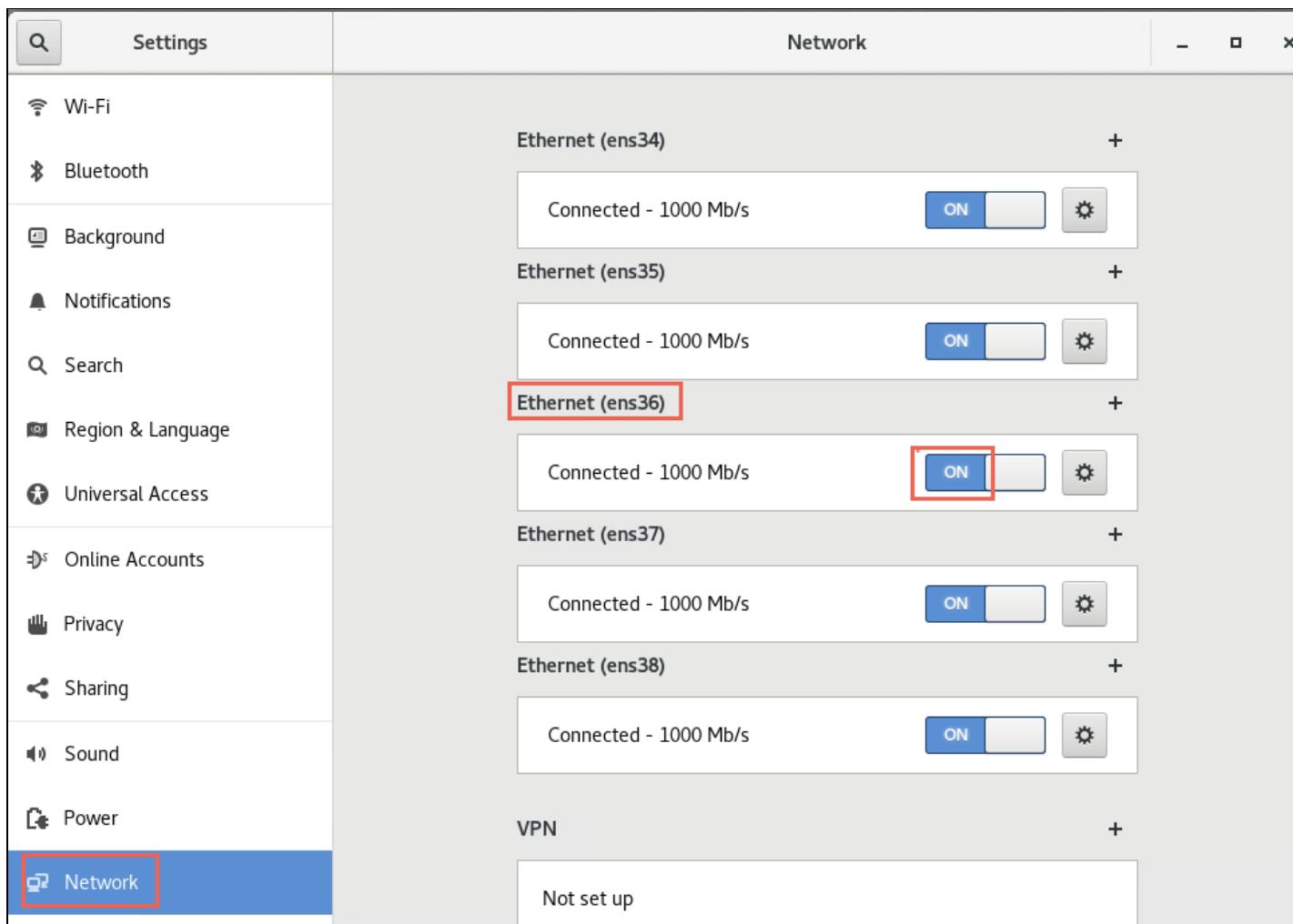
4. Switch to **dr1**. You will assume the secondary node has been replaced. Determine the command that needs to be run on the new Secondary node. On node **dr1**, run the command:

```
sudo rdqmldr -m QMDR -d
```

```
ibmuser@dr1:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ sudo rdqmldr -m QMDR -d
Issue the following command on node 'dr2' to create the DR secondary queue
manager:
crtmqm -rr s -rl 10.0.2.15 -ri 10.0.2.14 -rn dr1 -rp 7001 QMDR
[ibmuser@dr1 ~]$
```

5. On node **dr2**, go to **Applications -> System Tools -> Settings -> Network**. Click the button for the DR Replication adapter **ens36** (IP address 10.0.2.15), to switch it back on.



6. Copy this command, that was displayed on **dr1** into the command line of the new Secondary node, **dr2**, then run it:

```
sudo crtqm -rr s -rl 10.0.2.15 -ri 10.0.2.14 -rn dr1 -rp 7001 QMDR
```

```
ibmuser@dr2:~
```

File Edit View Search Terminal Help

```
[ibmuser@dr2 ~]$ sudo crtqm -rr s -rl 10.0.2.15 -ri 10.0.2.14 -rn dr1 -rp 7001 QMDR
Creating replicated data queue manager configuration.
IBM MQ secondary queue manager created.
Command '/opt/mqm/bin/crtqm' run with sudo.
[ibmuser@dr2 ~]$
```

7. Confirm the status of the DR configuration on both nodes:

```
sudo rdqmstatus -m QMDR
```

On dr2:

```
ibmuser@dr2:~ -  
File Edit View Search Terminal Help  
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR  
Queue manager status: Ended immediately  
DR role: Secondary  
DR status: Synchronization in progress  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.15  
DR remote IP address: 10.0.2.14  
DR synchronization progress: 47.50%  
DR estimated time of completion: 2020-12-25 11:02:33  
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR  
Queue manager status: Ended immediately  
DR role: Secondary  
DR status: Normal  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.15  
DR remote IP address: 10.0.2.14  
[ibmuser@dr2 ~]$
```

On dr1:

```
ibmuser@dr1:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.00%
Memory: 181MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
DR role: Primary
DR status: Synchronization in progress
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
DR synchronization progress: 85.13%
DR estimated time of completion: 2020-12-25 11:02:35
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.00%
Memory: 181MB
Queue manager file system: 58MB used, 2.9GB allocated [2%]
DR role: Primary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
[ibmuser@dr1 ~]$
```

Wait for synchronization to complete - *DR status = Normal* before continuing.

Reverting to a snapshot

Suppose a network connection between the nodes is lost; the changes to the persistent data for the primary instance of a queue manager are tracked. When the network connection is restored, a synchronization process is used to get the secondary instance up to speed as quickly as possible.

While synchronization is in progress, the data on the secondary instance is in an inconsistent state. A snapshot of the state of the secondary queue manager data is taken.

If a failure of the main node or the network connection occurs during synchronization, it would be necessary to revert the secondary instance back to this snapshot.

Create a snapshot

You will first create an inconsistent state on your DR nodes.

1. On **dr1** in the terminal (as ibmuser) run the runmqsc command.

2. Create a local queue for placing messages to provide some data for later synchronization. Use the runmqsc command to create a local, persistent queue, called Q1DR.

```
runmqsc QMDR
```

```
DEFINE QLOCAL(Q1DR) DEFPSIST(YES) MAXDEPTH(100000) REPLACE
```

```
end
```

The screenshot shows a terminal window with the following details:

- Title Bar:** ibmuser@dr1:/opt/mqm/samp/bin
- Menu Bar:** File Edit View Search Terminal Help
- Command Line:** [ibmuser@dr1 bin]\$ runmqsc QMDR
- Output:**
 - 5724-H72 (C) Copyright IBM Corp. 1994, 2020.
 - Starting MQSC for queue manager QMDR.
 - DEFINE QLOCAL(Q1DR) DEFPSIST(YES) MAXDEPTH(50000) REPLACE
 - 1 : DEFINE QLOCAL(Q1DR) DEFPSIST(YES) MAXDEPTH(50000) REPLACE
 - AMQ8006I: IBM MQ queue created.
 - END
 - 2 : END
 - One MQSC command read.
 - No commands have a syntax error.
 - All valid MQSC commands were processed.
- Prompt:** [ibmuser@dr1 bin]\$

3. Check that the status on both nodes is normal.

```
sudo rdqmstatus -m QMDR
```

4. In this section you will simulate a failure of the DR replication network adapter. You should still have the network settings open. You will simulate the outage as you did before by turning off the `ens36` adapter on **dr1** when instructed to do so.
5. In the *ibmuser* window on **dr1**, start putting some messages onto the queue. Use the `amqsblst` sample to do this with the following commands:

```
cd /opt/mqm/samp/bin
```

```
./amqsblst QMDR Q1DR -W -s 1000 -c 100000
```

The output should look similar to the following:

```
ibmuser@dr1:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@dr1 bin]$ ./amqsblst QMDR Q1DR -W -s 1000 -c 100000
welcome to blast
Blast> successfully opened queue <Q1DR>
Blast> 100 messages sent
Blast> 200 messages sent
Blast> 300 messages sent
Blast> 400 messages sent
Blast> 500 messages sent
Blast> 600 messages sent
Blast> 700 messages sent
Blast> 800 messages sent
Blast> 900 messages sent
Blast> 1000 messages sent
Blast> 1100 messages sent
```

6. Before all the messages have been put onto the queue (approximately 50,000 messages), turn off the `ens36` network adapter in the network settings. This will simulate a network outage on node `dr1`.

There will be a short pause, then the placing of messages will resume.

```
ibmuser@dr1:/opt/mqm/samp/bin
File Edit View Search Terminal Help
Blast> 99000 messages sent
Blast> 99100 messages sent
Blast> 99200 messages sent
Blast> 99300 messages sent
Blast> 99400 messages sent
Blast> 99500 messages sent
Blast> 99600 messages sent
Blast> 99700 messages sent
Blast> 99800 messages sent
Blast> 99900 messages sent
Blast> 100000 messages sent
Blast> elapsed time = 167.252499 S
Blast> ended
Blast> 100000 messages have been put
Blast> 0 messages have been got
[ibmuser@dr1 bin]$
```

7. As user `ibmuser`, check the status on both nodes.

```
sudo rdqmstatus -m QMDR
```

Notice on node **dr1**, the DR status is showing as ‘Remote unavailable’.

```
ibmuser@dr1:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@dr1 bin]$ sudo rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.13%
Memory: 183MB
Queue manager file system: 65MB used, 2.9GB allocated [2%]
DR role: Primary
DR status: Remote unavailable
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
DR out of sync data: 9148KB
[ibmuser@dr1 bin]$
```

Similarly, on node **dr2**:

```
ibmuser@dr2:~
File Edit View Search Terminal Help
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Ended immediately
DR role: Secondary
DR status: Remote unavailable
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.15
DR remote IP address: 10.0.2.14
DR out of sync data: 0KB
[ibmuser@dr2 ~]$
```

8. Issuing the command again on node **dr1**, when all the messages have been placed on the queue, you will notice the ‘DR out of sync data’ has changed. Your number will be different than the screenshot.

```
ibmuser@dr1:/opt/mqm/samp/bin
File Edit View Search Terminal Help
[ibmuser@dr1 bin]$ sudo rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.29%
Memory: 183MB
Queue manager file system: 66MB used, 2.9GB allocated [2%]
DR role: Primary
DR status: Remote unavailable
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
DR out of sync data: 14552KB
[ibmuser@dr1 bin]$
```

⚠ Warning:

When you restore the network, synchronization will complete very quickly. If you don't display the status immediately you may not see the 'synchronization in progress' message. Be ready to run the status command as soon as you flip the network switch to **ON**. Once you run the status commands, immediately turn the switch **OFF** again.

9. Simulate the restoration of the network outage on node **dr1** by turning on the **ens36** adapter in network settings.
10. The nodes will start synchronizing as soon as this happens. Check the status on both nodes immediately, before switching off the network. They will look similar to the following:

```
sudo rdqmstatus -m QMDR
```

```
ibmuser@dr1:~$ sudo rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.00%
Memory: 188MB
Queue manager file system: 131MB used, 2.9GB allocated [4%]
DR role: Primary
DR status: Synchronization in progress
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
DR synchronization progress: 20.47%
DR estimated time of completion: 2020-12-26 10:05:34
```

```
ibmuser@dr2:~$ sudo rdqmstatus -m QMDR
Queue manager status: Ended immediately
DR role: Secondary
DR status: Synchronization in progress
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.15
DR remote IP address: 10.0.2.14
DR synchronization progress: 77.57%
DR estimated time of completion: 2020-12-26 10:05:34
```

11. Immediately (before synchronization is *complete*), simulate a network outage on node **dr1** again by turning off the `ens36` adapter in network settings.
12. When the network is detected to have failed again, the status on the primary node, **dr1** goes back to 'Remote unavailable' and the amount of data that is out of synchronization.

```
ibmuser@dr1:~
```

| | | | | | |
|---|----------------------------------|------|--------|----------|------|
| File | Edit | View | Search | Terminal | Help |
| [ibmuser@dr1 ~]\$ sudo rdqmstatus -m QMDR | | | | | |
| Queue manager status: | Running | | | | |
| CPU: | 0.84% | | | | |
| Memory: | 188MB | | | | |
| Queue manager file system: | 131MB used, 2.9GB allocated [4%] | | | | |
| DR role: | Primary | | | | |
| DR status: | Remote unavailable | | | | |
| DR type: | Asynchronous | | | | |
| DR port: | 7001 | | | | |
| DR local IP address: | 10.0.2.14 | | | | |
| DR remote IP address: | 10.0.2.15 | | | | |
| DR out of sync data: | 5720KB | | | | |

The status on the secondary node, **dr2** is ‘Inconsistent’.

```
ibmuser@dr2:~
```

| | | | | | |
|---|-------------------|------|--------|----------|------|
| File | Edit | View | Search | Terminal | Help |
| [ibmuser@dr2 ~]\$ sudo rdqmstatus -m QMDR | | | | | |
| Queue manager status: | Ended immediately | | | | |
| DR role: | Secondary | | | | |
| DR status: | Inconsistent | | | | |
| DR type: | Asynchronous | | | | |
| DR port: | 7001 | | | | |
| DR local IP address: | 10.0.2.15 | | | | |
| DR remote IP address: | 10.0.2.14 | | | | |
| DR out of sync data: | 0KB | | | | |

Revert to a snapshot

You will now see how the secondary instance reverts to its snapshot and the queue manager data. Note any updates that have happened since the original network failure, however, will be lost.

1. The assumption now is that the Primary node is no longer usable, so the replication node must be made the new Primary instance. On the secondary node **dr2**, in the **root** user terminal, designate **dr2** as the primary instance:

```
sudo rdqmdr -m QMDR -p
```

2. Due to its former ‘Inconsistent’ state, **dr2** will revert to a snapshot. Check the status to confirm this.

```
rdqmstatus -m QMDR
```

The output should look like the following:

A screenshot of a terminal window titled "root@dr2:~". The window shows the command "rdqmstatus -m QMDR" being run, followed by the output of the queue manager status. A red box highlights the command "rdqmstatus -m QMDR". A red callout bubble points from this box to the text "Reverting to snapshot" in the status output. The status output includes fields like DR role (Secondary), DR status (Reverting to snapshot), and Snapshot reversion progress (95.99%).

```
root@dr2:~  
File Edit View Search Terminal Help  
[root@dr2 ~]# rdqmstatus -m QMDR  
Started reversion to snapshot.  
[root@dr2 ~]# rdqmstatus -m QMDR  
Queue manager status:  
DR role: Secondary  
DR status: Reverting to snapshot  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.15  
DR remote IP address: 10.0.2.14  
Snapshot reversion progress: 95.99%
```

3. When node **dr2** has completed reverting to the snapshot check the status again.

```
rdqmstatus -m QMDR
```

It should look similar to the following:

The screenshot shows a terminal window with the title bar "root@dr2:~". The menu bar includes File, Edit, View, Search, Terminal, and Help. The command [root@dr2 ~]# rdqmstatus -m QMDR is run, displaying the following information:

| Parameter | Value |
|----------------------------|----------------------------------|
| Queue manager status: | Ended unexpectedly |
| Queue manager file system: | 118MB used, 2.9GB allocated [4%] |
| DR role: | Primary |
| DR status: | Remote unavailable |
| DR type: | Asynchronous |
| DR port: | 7001 |
| DR local IP address: | 10.0.2.15 |
| DR remote IP address: | 10.0.2.14 |
| DR out of sync data: | 3145596KB |

Notice the status indicates the queue manager '*Ended unexpectedly*', and there is data that is out of synchronization.

4. As this situation would have occurred as a result of a possible failure of the Primary node, in reality you would go through the process described earlier to *Add the new Primary node into the DR configuration*.

Once the new Primary node was part of the DR configuration again, you would follow the steps to 'Restore the original DR configuration' of the Primary node, being the primary instance of the queue manager.

Here you will simulate something similar. You will start the queue manager on node **dr2**. You will delete the queue manager on node **dr1**, and go through the latter of the steps mentioned above again, to show some additional screens not seen previously.

Wait for the reversion process to complete before continuing.

```
ibmuser@dr2:~  
File Edit View Search Terminal Help  
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR  
Queue manager status: Ended immediately  
DR role: Secondary  
DR status: Reverting to snapshot  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.15  
DR remote IP address: 10.0.2.14  
Snapshot reversion progress: 99.66%  
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR  
Queue manager status: Ended immediately  
DR role: Secondary  
DR status: Reverting to snapshot  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.15  
DR remote IP address: 10.0.2.14  
Snapshot reversion progress: 100.00%  
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR  
Queue manager status: Status not available  
DR role: Unknown  
DR status: Unknown  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.15  
DR remote IP address: 10.0.2.14  
[ibmuser@dr2 ~]$
```

5. Start the queue manager on node **dr2**, which is now the primary instance.

```
strmqm QMDR
```

6. Confirm the queue manager on node **dr2** is running as the primary instance by checking the status.

```
rdqmstatus -m QMDR
```

```
ibmuser@dr2:~  
File Edit View Search Terminal Help  
[ibmuser@dr2 ~]$ strmqm QMDR  
IBM MQ queue manager 'QMDR' starting.  
The queue manager is associated with installation 'Installation1'.  
2805 log records accessed on queue manager 'QMDR' during the log replay phase.  
4892 log records accessed on queue manager 'QMDR' during the log replay phase.  
7457 log records accessed on queue manager 'QMDR' during the log replay phase.  
7621 log records accessed on queue manager 'QMDR' during the log replay phase.  
Log replay for queue manager 'QMDR' complete.  
Transaction manager state recovered for queue manager 'QMDR'.  
IBM MQ queue manager 'QMDR' started using V9.1.5.0.  
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR  
Queue manager status: Running  
CPU: 0.00%  
Memory:  
Queue manager file system: 181MB 205MB used, 2.9GB allocated [7%]  
DR role: Primary  
DR status: Remote unavailable  
DR type: Asynchronous  
DR port: 7001  
DR local IP address: 10.0.2.15  
DR remote IP address: 10.0.2.14  
DR out of sync data: 3145596KB  
[ibmuser@dr2 ~]$
```

Notice that as a result of reverting to a snapshot there is data out of synchronization.

7. ‘Simulate’ the replacement of node **dr1**. On node **dr1**, in the *ibmuser* window, stop the queue manager:

```
endmqm QMDR
```

8. Confirm the queue manager on node **dr1** ended normally by checking the status

```
rdqmstatus -m QMDR
```

9. In the **root** terminal, delete the queue manager on node **dr1**.

```
dltmqm QMDR
```

```
root@dr1:~ - □ ×
File Edit View Search Terminal Help
[root@dr1 ~]# dltmqm QMDR
IBM MQ queue manager 'QMDR' deleted.
The DR replicated data queue manager might also need to be deleted on remote
node 'dr2'.
[root@dr1 ~]#
```

10. On node **dr1**, restart the network interface by turning on the `ens36` adapter in network settings.
11. On node **dr2**, enter the command to determine the command needed to recreate the queue manager on node dr1.

```
sudo rdqmldr -m QMDR -d
```

```
ibmuser@dr2:~ - □ ×
File Edit View Search Terminal Help
[ibmuser@dr2 ~]$ sudo rdqmldr -m QMDR -d
Issue the following command on node 'dr1' to create the DR secondary queue
manager.
crtmqm -rr s -rl 10.0.2.14 -ri 10.0.2.15 -rn dr2 -rp 7001 QMDR
[ibmuser@dr2 ~]$
```

Copy command and run on dr1

12. On node **dr1**, as user **root**, issue the command to recreate the queue manager:

```
sudo crtmqm -rr s -rl 10.0.2.1 -ri 10.0.2.2 -rn dr2 -rp 7001 QMDR
```

The screenshot shows a terminal window titled "ibmuser@dr1:~". The command run is "sudo crtmdm -rr s -rl 10.0.2.14 -ri 10.0.2.15 -rn dr2 -rp 7001 OMDR". The output indicates that a replicated data queue manager configuration is being created, and an IBM MQ secondary queue manager has been created. A red box highlights the command line, and a green oval highlights the text "Run command copied from dr2".

```
[ibmuser@dr1 ~]$ sudo crtmdm -rr s -rl 10.0.2.14 -ri 10.0.2.15 -rn dr2 -rp 7001 OMDR
Creating replicated data queue manager configuration.
IBM MQ secondary queue manager created.
[ibmuser@dr1 ~]$
```

13. Check the status on both nodes:

```
sudo rdqmstatus -m QMDR
```

On **dr1**:

The screenshot shows a terminal window titled "ibmuser@dr1:~". The command run is "rdqmstatus -m OMDR". The output displays the queue manager status as "Ended immediately" and the DR role as "Secondary". The DR status is shown as "Synchronization in progress". Other details include DR type as "Asynchronous", DR port as "7001", DR local IP address as "10.0.2.14", DR remote IP address as "10.0.2.15", DR synchronization progress as "7.75%", and DR estimated time of completion as "2020-12-26 13:26:20". A red box highlights the command line, and a red box highlights the DR status line.

```
[ibmuser@dr1 ~]$ rdqmstatus -m OMDR
Queue manager status: Ended immediately
DR role: Secondary
DR status: Synchronization in progress
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
DR synchronization progress: 7.75%
DR estimated time of completion: 2020-12-26 13:26:20
```

On **dr2**:

```
ibmuser@dr2:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.00%
Memory: 181MB
Queue manager file system: 133MB used, 2.9GB allocated [5%]
DR role: Primary
DR status: Synchronization in progress
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.15
DR remote IP address: 10.0.2.14
DR synchronization progress: 73.15%
DR estimated time of completion: 2020-12-26 13:26:24
```

Notice that synchronization of data is taking place between the nodes. Indications are given on its progress and estimated completion time.

14. When data synchronization has completed the status of the nodes will look similar to the following:

On dr1:

```
ibmuser@dr1:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Ended immediately
DR role: Secondary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
[ibmuser@dr1 ~]$
```

On dr2:

```
ibmuser@dr2:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.00%
Memory: 181MB
Queue manager file system: 133MB used, 2.9GB allocated [5%]
DR role: Primary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.15
DR remote IP address: 10.0.2.14
[ibmuser@dr2 ~]$
```

15. Restore the DR configuration. On node **dr2**, as *ibmuser*, stop the queue manager:

```
endmqm QMDR
```

16. Make node **dr2** the secondary instance:

```
sudo rdqmldr -m QMDR -s
```

```
ibmuser@dr2:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr2 ~]$ endmqm QMDR
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
[ibmuser@dr2 ~]$ sudo rdqmldr -m QMDR -s
Queue manager 'QMDR' has been made the DR secondary on this node.
```

17. On node **dr1**, as user **root**, make it the primary instance of the queue manager:

```
sudo rdqmldr -m QMDR -p
```

18. Start the queue manager on node **dr1**:

```
strmqm QMDR
```

A terminal window titled "ibmuser@dr1:~". The window shows the command "sudo rdqmldr -m QMDR -p" being run, followed by a message indicating the queue manager has been made the DR primary. Then, the command "strmqm QMDR" is run, and the output shows the queue manager starting, associating with "Installation1", replaying log records, and recovering the transaction manager state.

```
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ sudo rdqmldr -m QMDR -p
Queue manager 'QMDR' has been made the DR primary on this node.
[ibmuser@dr1 ~]$ strmqm QMDR
IBM MQ queue manager 'QMDR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMDR' during the log replay phase.
Log replay for queue manager 'QMDR' complete.
Transaction manager state recovered for queue manager 'QMDR'.
```

19. Confirm the DR configuration by checking the status on both nodes:

```
sudo rdqmstatus -m QMDR
```

```
ibmuser@dr1:~
```

File Edit View Search Terminal Help

```
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Running
CPU: 0.02%
Memory: 181MB
Queue manager file system: 133MB used, 2.9GB allocated [5%]
DR role: Primary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
[ibmuser@dr1 ~]$
```

```
ibmuser@dr2:~
```

File Edit View Search Terminal Help

```
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Ended immediately
DR role: Secondary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.15
DR remote IP address: 10.0.2.14
[ibmuser@dr2 ~]$
```

Delete a DR RDQM

If Disaster Recovery is no longer required for a queue manager, the queue manager needs to be deleted to be removed from the DR configuration. This is achieved as follows:

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

```
endmqm QMDR
```

2. View the status to confirm that has happened :

```
sudo rdqmstatus -m QMDR
```

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

```
File Edit View Search Terminal Help
[ibmuser@dr1 ~]$ endmqm QMDR
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMDR
Queue manager status: Ended normally
Queue manager file system: 133MB used, 2.9GB allocated [5%]
DR role: Primary
DR status: Normal
DR type: Asynchronous
DR port: 7001
DR local IP address: 10.0.2.14
DR remote IP address: 10.0.2.15
[ibmuser@dr1 ~]$
```

3. On **dr1** remove the queue manager:

```
sudo dltmqm QMDR
```

4. Viewing the status:

```
sudo rdqmstatus -m QMDR
```

will confirm the queue manager no longer exists on the primary node.

```
ibmuser@dr1:~
```

File Edit View Search Terminal Help

```
[ibmuser@dr1 ~]$ sudo dltnqm QMDR
IBM MQ queue manager 'QMDR' deleted.
The DR replicated data queue manager might also need to be deleted on remote
node 'dr2'.
[ibmuser@dr1 ~]$ sudo rdqmstatus -m QMDR
The queue manager does not exist.
[ibmuser@dr1 ~]$
```

5. Also remove the queue manager on **dr2**:

```
sudo dltnqm QMDR
```

6. Again, viewing the status:

```
sudo rdqmstatus -m QMDR
```

will confirm the queue manager no longer exists on the secondary node either.

```
ibmuser@dr2:~
```

File Edit View Search Terminal Help

```
[ibmuser@dr2 ~]$ sudo dltnqm QMDR
IBM MQ queue manager 'QMDR' deleted.
The DR replicated data queue manager might also need to be deleted on remote
node 'dr1'.
[ibmuser@dr2 ~]$ sudo rdqmstatus -m QMDR
The queue manager does not exist.
[ibmuser@dr2 ~]$
```

Cleanup environment

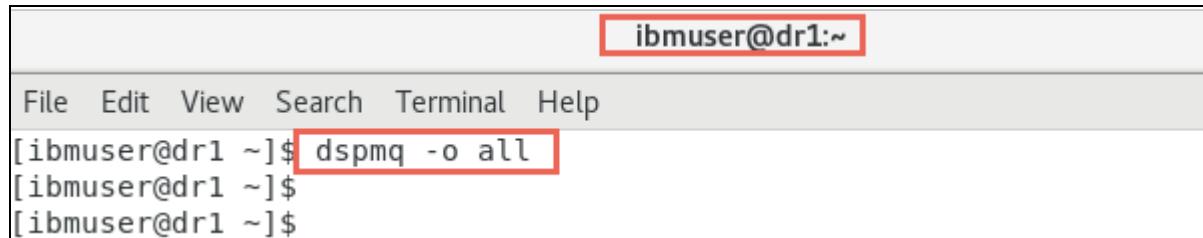
Before continuing to Lab 4, you must clean the environment by removing any RDQM definitions.

1. On **dr1**, 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
```

There should not be any remaining queue managers, but if you have defined any then stop and delete the queue managers.

```
endmqm xxxx  
sudo dltmqm xxxx
```



A screenshot of a terminal window titled "ibmuser@dr1:~". The window has a standard Mac OS X style menu bar with File, Edit, View, Search, Terminal, and Help. Below the menu bar, the prompt "[ibmuser@dr1 ~]\$" appears three times, followed by the command "dspmq -o all" which is highlighted with a red rectangle. The terminal window is set against a light gray background.

```
ibmuser@dr1:~  
File Edit View Search Terminal Help  
[ibmuser@dr1 ~]$ dspmq -o all  
[ibmuser@dr1 ~]$  
[ibmuser@dr1 ~]$
```

2. On **dr2**, 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
```

There should not be any remaining queue managers, but if you have defined any then stop and delete the queue managers.

```
endmqm xxxx  
sudo dltmqm xxxx
```

```
ibmuser@dr2:~
```

```
File Edit View Search Terminal Help
[ibmuser@dr2 ~]$ dspmq -o all
[ibmuser@dr2 ~]$
[ibmuser@dr2 ~]$
```

CONGRATULATIONS!

You have completed this hands-on lab.

You have created replicated data queue managers to provide disaster recovery for IBM MQ.

[Continue to Lab 4 \(mq_ha_pot_lab4.html\)](#)

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

©2020 IBM. All rights reserved.

Site last generated: Dec 28, 2020

