Unit 12. Introduction to queue manager clusters

Estimated time

01:00

Overview

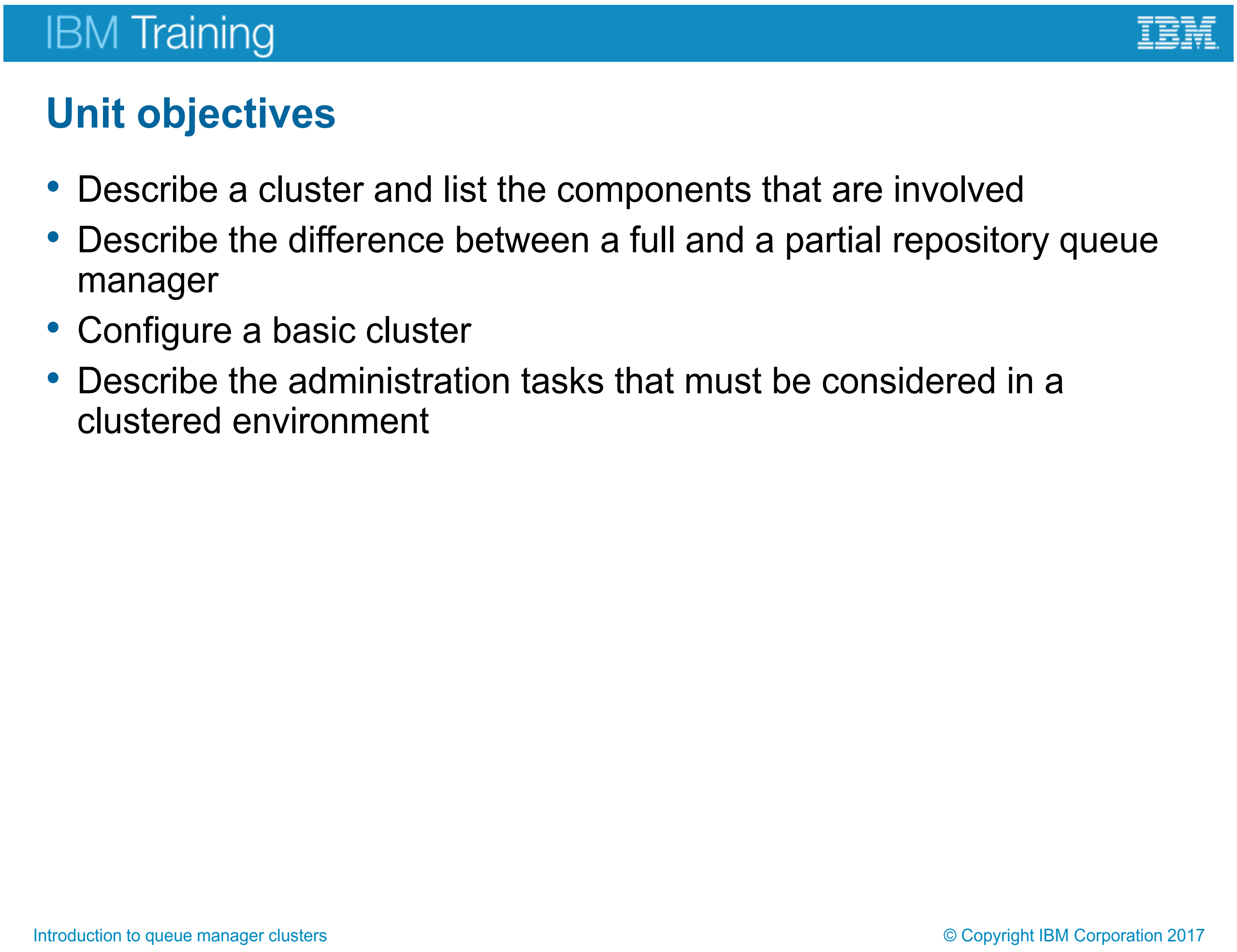
In this unit, you learn about the basic concepts of queue manager clustering. The unit provides an overview of queue manager cluster components and definitions that are required for setting up a simple clustered environment.

How you will check your progress

- Review questions
- Hands-on exercise

References

IBM Knowledge Center for IBM MQ V9



IBM Training

Unit objectives

- Describe a cluster and list the components that are involved
- Describe the difference between a full and a partial repository queue manager
- Configure a basic cluster
- Describe the administration tasks that must be considered in a clustered environment

Introduction to queue manager clusters

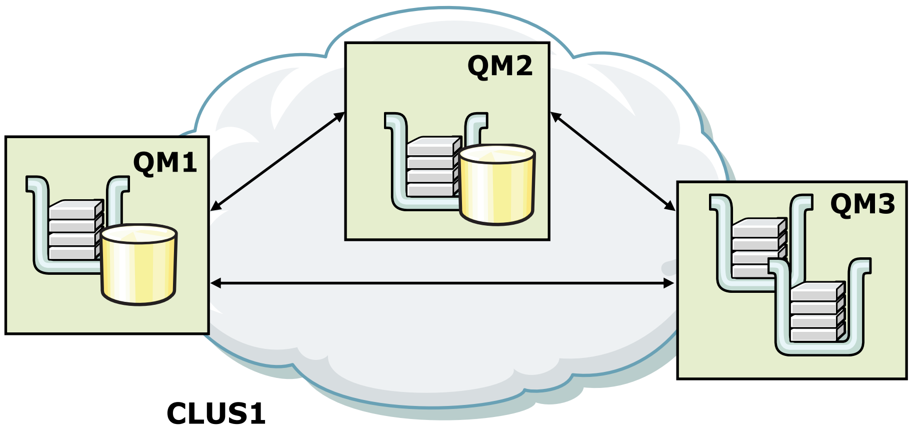
© Copyright IBM Corporation 2017

Figure 12-1. Unit objectives

IBM Training

What is a cluster?

- Named group of queue managers that make the queues that they host available to other queue managers in the cluster



Introduction to queue manager clusters

Figure 12-2. What is a cluster?

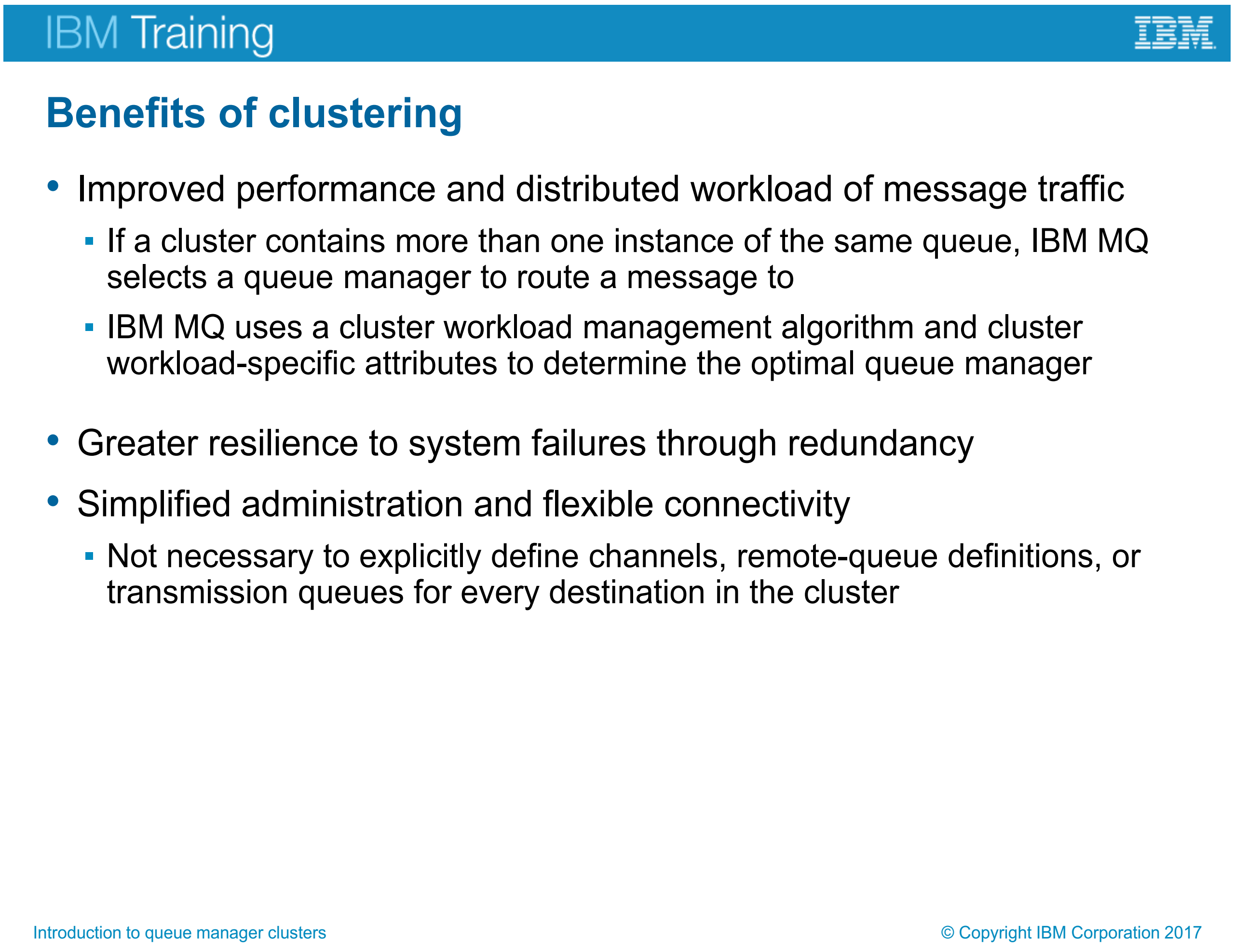
A cluster is a collection of queue managers that can be on different servers and operating systems, and typically serve a common application.

Every queue manager can make the queues that it hosts available to every other queue manager in the cluster, without the need for (remote) queue definitions. Cluster-specific objects also remove the need for explicit channel definitions and transmission queues for each destination queue manager.

The queue managers in a cluster often assume the role of a client or a server. The servers host the queues that are available to the members of the cluster and applications that process these messages and generate responses. The clients PUT messages to the server queues and might receive response messages.

Queue managers in a cluster normally communicate directly with each other, although typically, many of the client systems never need to communicate with other client systems.

Clustering was introduced in the prerequisite course WM103/ZM103, *A Technical Introduction to IBM MQ*.



IBM Training

Benefits of clustering

- Improved performance and distributed workload of message traffic
  - If a cluster contains more than one instance of the same queue, IBM MQ selects a queue manager to route a message to
  - IBM MQ uses a cluster workload management algorithm and cluster workload-specific attributes to determine the optimal queue manager
- Greater resilience to system failures through redundancy
- Simplified administration and flexible connectivity
  - Not necessary to explicitly define channels, remote-queue definitions, or transmission queues for every destination in the cluster

Introduction to queue manager clusters © Copyright IBM Corporation 2017

Figure 12-3. Benefits of clustering

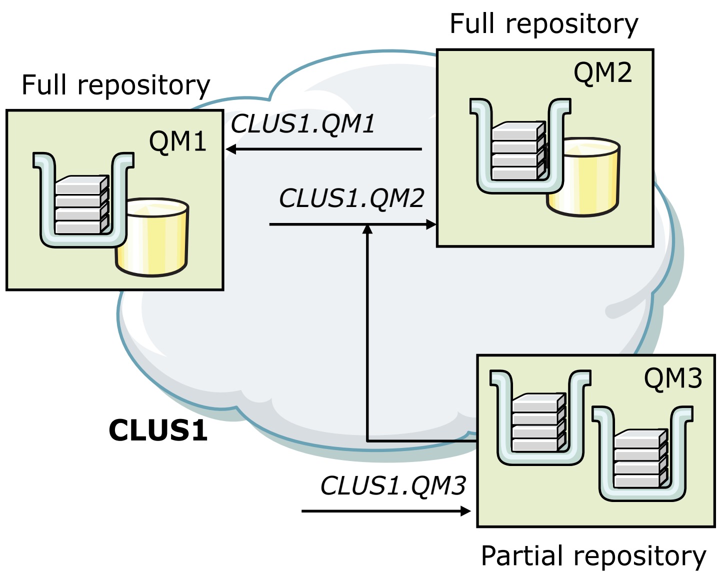
Clusters simplify the administration of MQ networks, which usually require many object definitions for channels, transmission queues, and remote queues.

Clusters can be used to distribute the workload of message traffic across queues and queue managers in the cluster. Such distribution allows the message workload of a single queue to be distributed across equivalent instances of that queue that are on multiple queue managers.

The distribution of the workload can be used to achieve greater resilience with system failures, and to improve the scaling performance of active message flows in a system.

IBM Training

Overview of cluster components



- *Full repository queue manager* hosts a complete set of information about every queue manager in the cluster
- *Partial repository queue manager* contains information about those queue managers with which it needs to exchange messages
- *Cluster queues*
  - Application queues
  - Transmission queues
- *Cluster channels*
  - Cluster-receiver
  - Cluster-sender

Introduction to queue manager clusters

Figure 12-4. Overview of cluster components

A cluster can contain two or more queue managers. The example cluster in the figure contains three queue managers in a cluster that is named CLUS1: QM1, QM2, and QM3.

Two of queue managers, QM1 and QM2, are configured as *full repository queue managers*. Full repository queue managers contain information about all the queue managers in the cluster. The repositories are represented in the figure by the shaded cylinders. The other queue manager in this cluster is a *partial repository queue manager*. A partial repository queue manager holds records for local objects and remote objects that are used locally.

All the queue managers in the cluster can host local queues that are accessible to any other queue manager in the cluster. These queues are called *cluster queues*. As with distributed queuing, an application uses an MQPUT call to put a message on a cluster queue at any queue manager. An application uses the MQGET call to retrieve messages from a cluster queue on the local queue manager.

Each queue manager has a *cluster-receiver channel* for the receiving end of a channel. Include the name of the cluster and the name of the queue manager that is the destination of that channel in each channel name. A cluster-receiver channel is similar to a receiver channel that is used in distributed queuing, but in addition to carrying messages, this channel also carries information about the cluster.

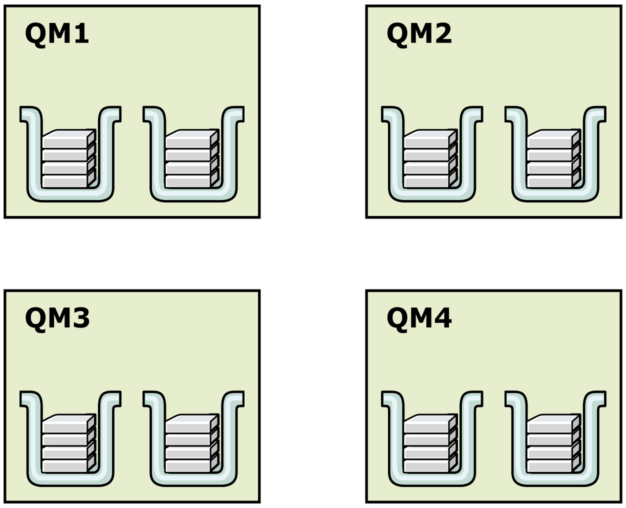
Each queue manager also has a *cluster-sender channel* definition for the sending end of a channel. A cluster-sender channel is similar to a sender channel that is used in distributed queuing, but in addition to carrying messages, this channel also carries information about the cluster.

IBM Training

Distributed queuing versus clustering (1 of 3)

- Example shows the definitions that are required in a network with four queue managers, each with two local queues in the configuration:
  - Using distributed queuing
  - Using a cluster

Four queue managers , each with two local queues



Introduction to queue manager clusters

Figure 12-5. Distributed queuing versus clustering (1 of 3)

IBM MQ clusters allow queue instances to be added. If you do not use clusters, your queue managers are independent and communicate by using distributed queuing. If one queue manager needs to send messages to another, it must define:

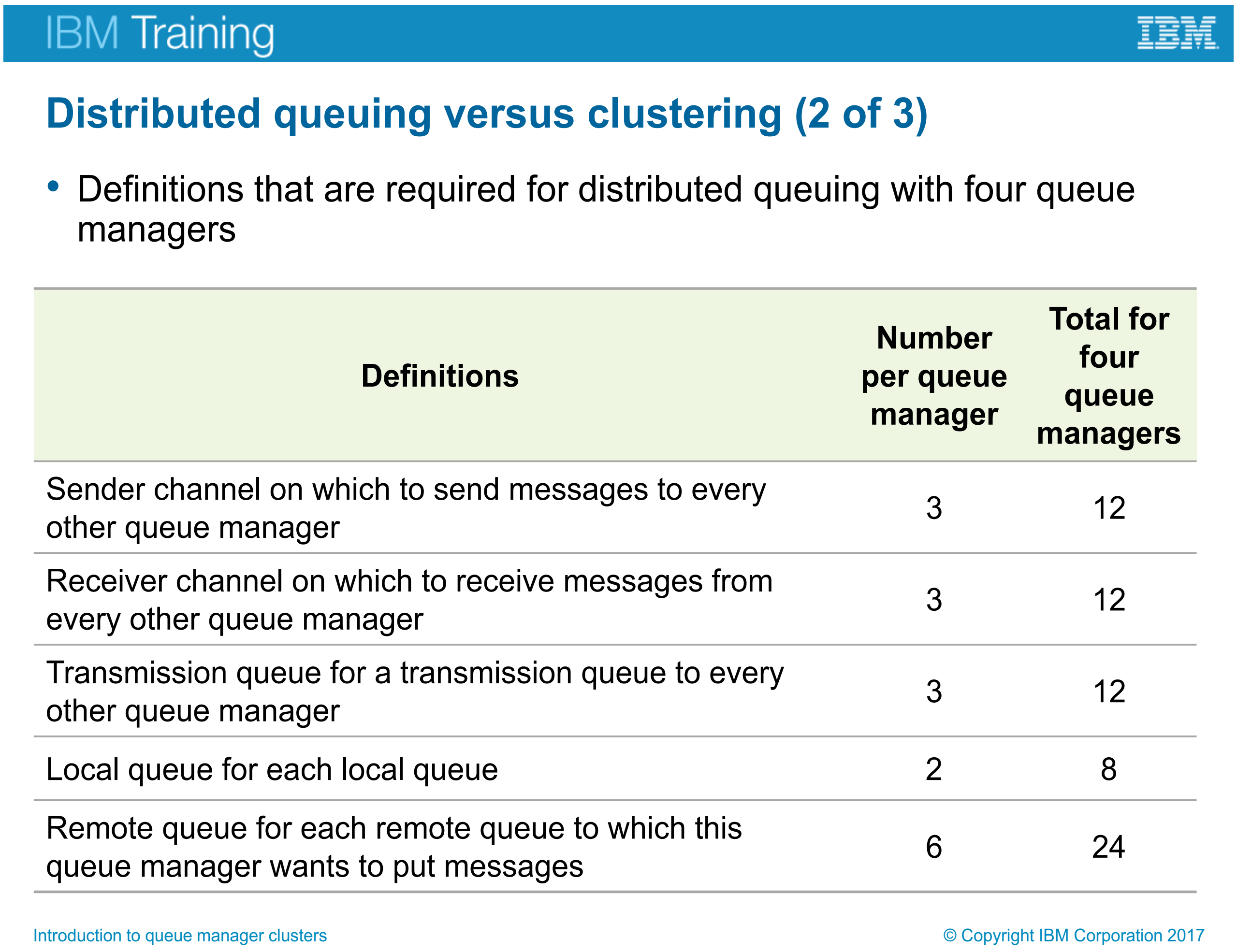
- A transmission queue
- A channel to the remote queue manager
- Remote queues

If queue managers are grouped in a cluster, the queue managers can make the queues that they host available to every other queue manager in the cluster. Any queue manager can send a message to any other queue manager in the same cluster without explicit channel definitions, remote-queue definitions, or transmission queues for each destination. Every queue manager in a cluster has a single transmission queue from which it can transmit messages to any other queue manager in the cluster.

Each queue manager in a cluster needs to define only:

- One cluster-receiver channel on which to receive messages
- One cluster-sender channel that points to one of the full repository queue managers

The MQ network in this comparison scenario uses four queue managers. Each queue manager has two application queues. Consider how many definitions are needed to connect these queue managers with distributed queuing when compared with a cluster.



IBM Training

Distributed queuing versus clustering (2 of 3)

- Definitions that are required for distributed queuing with four queue managers

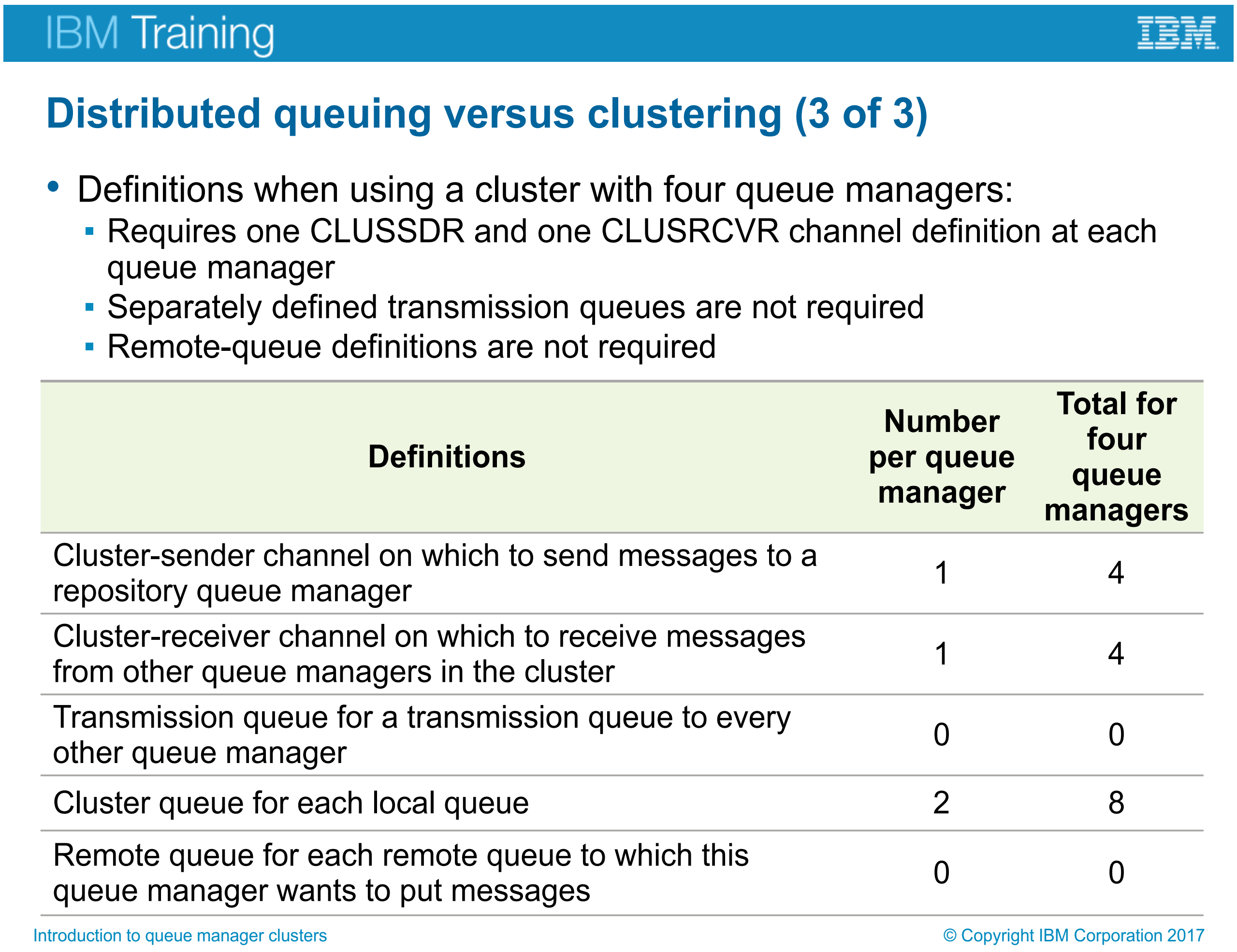
Definitions	Number per queue manager	Total for four queue managers
Sender channel on which to send messages to every other queue manager	3	12
Receiver channel on which to receive messages from every other queue manager	3	12
Transmission queue for a transmission queue to every other queue manager	3	12
Local queue for each local queue	2	8
Remote queue for each remote queue to which this queue manager wants to put messages	6	24

Introduction to queue manager clusters

Figure 12-6. Distributed queuing versus clustering (2 of 3)

The figure shows the number of definitions that are required to use distributed queuing with four queue managers and two application queues on each queue manager.

The maximum number of definitions can be as many as 17 on each queue manager, which is a total of 68 for this network. You might be able to reduce the number of definitions by using generic receiver-channel definitions, for example.



IBM Training

Distributed queuing versus clustering (3 of 3)

- Definitions when using a cluster with four queue managers:
  - Requires one CLUSSDR and one CLUSRCVR channel definition at each queue manager
  - Separately defined transmission queues are not required
  - Remote-queue definitions are not required

Definitions	Number per queue manager	Total for four queue managers
Cluster-sender channel on which to send messages to a repository queue manager	1	4
Cluster-receiver channel on which to receive messages from other queue managers in the cluster	1	4
Transmission queue for a transmission queue to every other queue manager	0	0
Cluster queue for each local queue	2	8
Remote queue for each remote queue to which this queue manager wants to put messages	0	0

Introduction to queue manager clusters © Copyright IBM Corporation 2017

Figure 12-7. Distributed queuing versus clustering (3 of 3)

This figure shows the number of definitions that are required when you use clustering to support the simplified administration example with four queue managers.

To set up this cluster of queue managers (with two full repositories), you would need four definitions on each queue manager; a total of 16 definitions. You would also need to alter the queue-manager definitions for two of the queue managers to make them full repository queue managers for the cluster.

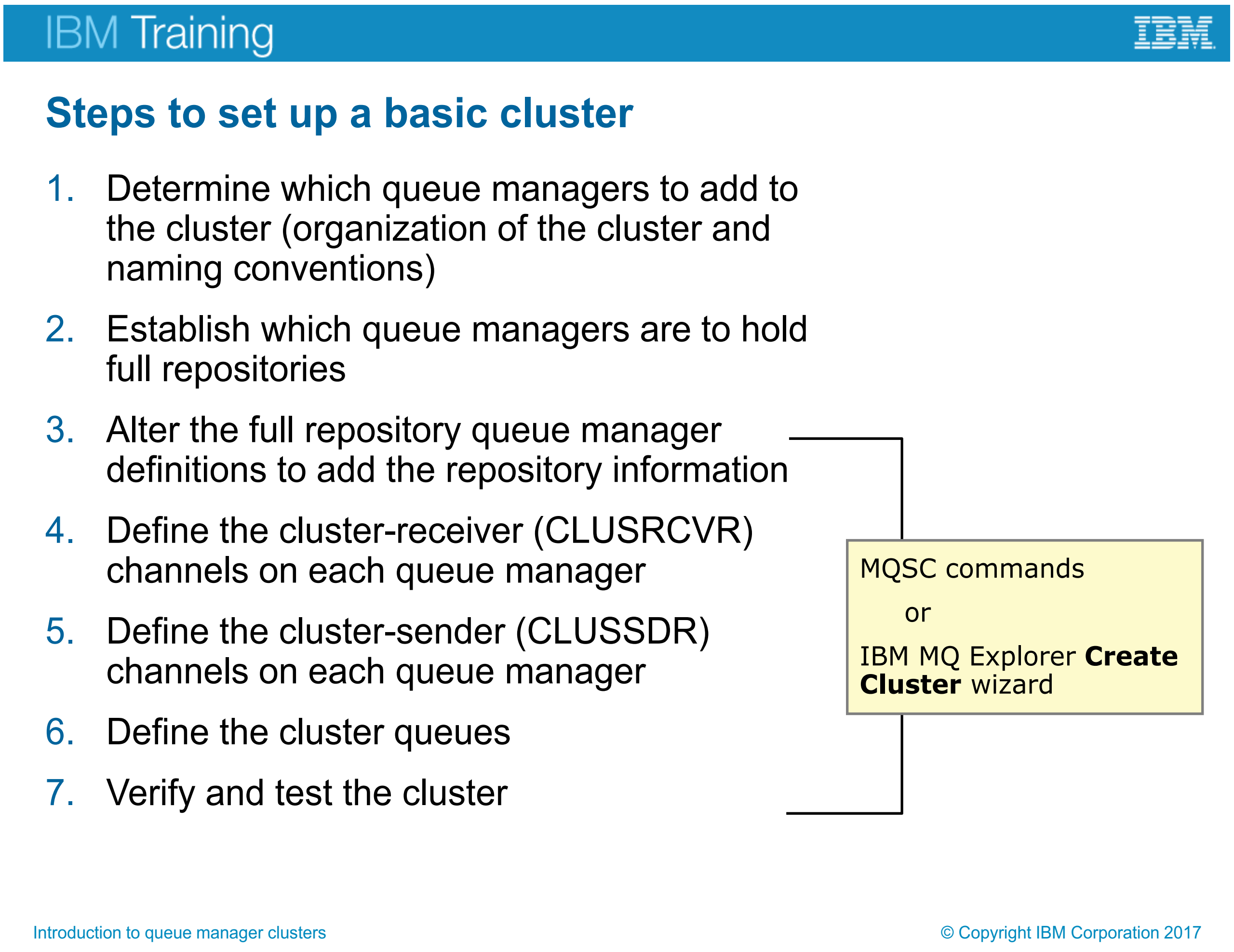
The cluster-sender and cluster-receiver channel definitions are made one time. When the cluster is in place, you can add or remove queue managers (other than the repository queue managers) without any disruption to the other queue managers. This process significantly reduces the number of definitions that are required to set up a network that contains many queue managers.

You have less risk for errors because you define fewer objects.

- Object names always match, for example, the channel name in a sender-receiver pair.
- The transmission queue name that is specified in a channel definition always matches the correct transmission queue definition or the transmission queue name that is specified in a remote queue definition.
- After a cluster is set up, cluster queues can be moved from one queue manager to another within the cluster without any system management on any other queue manager. You have no

chance of forgetting to delete or modify channels, remote queues, or transmission queues. You can add new queue managers to a cluster without any disruption to the existing network.

Compare the tables on both figures so that students see the difference in the configuration and recognize that clustering simplifies administration.


IBM Training

Steps to set up a basic cluster

1. Determine which queue managers to add to the cluster (organization of the cluster and naming conventions)
2. Establish which queue managers are to hold full repositories
3. Alter the full repository queue manager definitions to add the repository information
4. Define the cluster-receiver (CLUSRCVR) channels on each queue manager
5. Define the cluster-sender (CLUSSDR) channels on each queue manager
6. Define the cluster queues
7. Verify and test the cluster

MQSC commands
or
IBM MQ Explorer **Create Cluster** wizard

Introduction to queue manager clusters

© Copyright IBM Corporation 2017

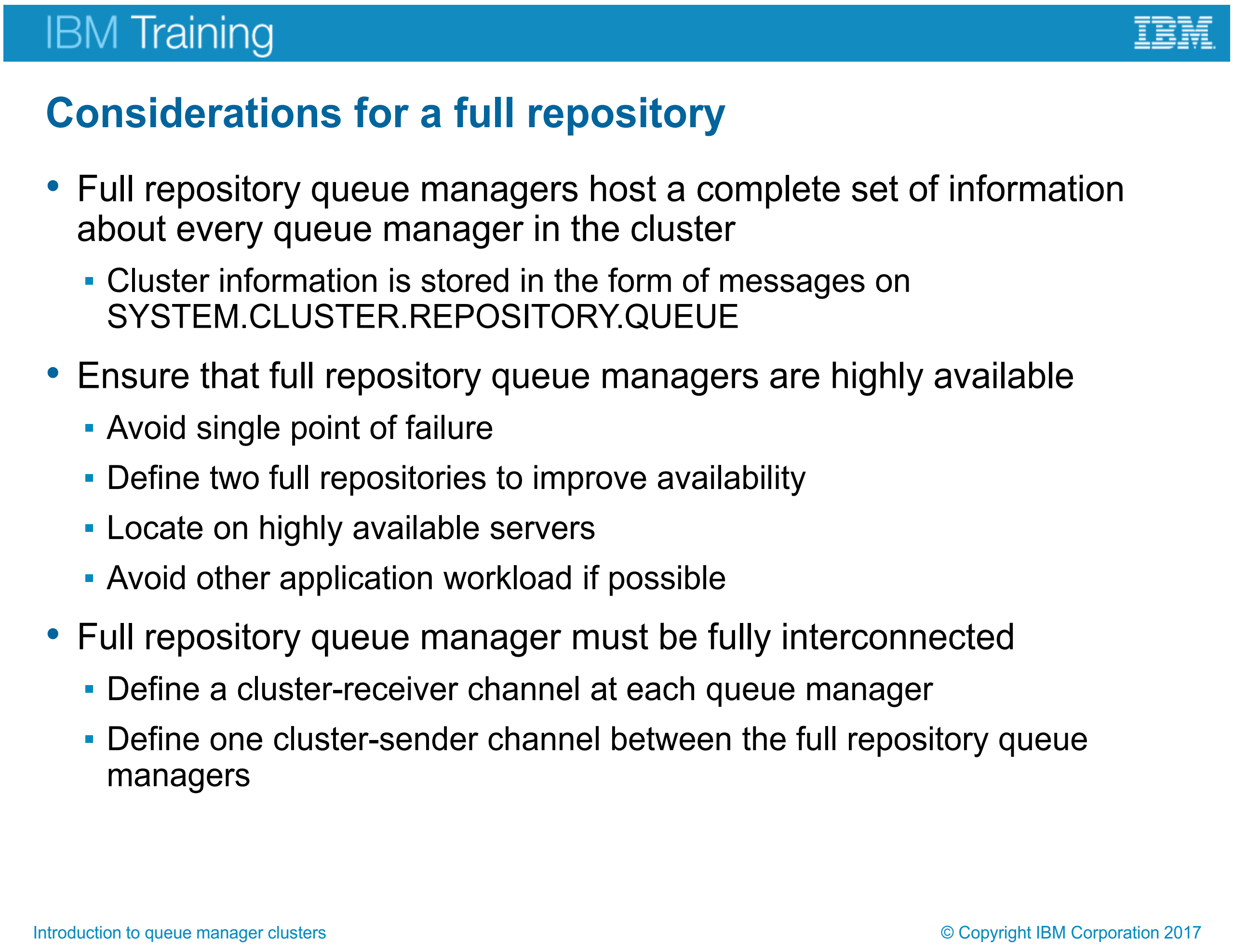
Figure 12-8. Steps to set up a basic cluster

The figure lists the basic steps for setting up a cluster.

You can also use MQSC or one of the wizards that are supplied with IBM MQ Explorer to create a cluster.

To use IBM MQ Explorer, right-click the **Queue Manager Clusters** folder, click **New > Queue Manager Cluster**, and follow the instructions in the wizard.

Each of these steps is described in detail next. You also define a simple cluster in the lab exercise for this unit.



IBM Training

Considerations for a full repository

- Full repository queue managers host a complete set of information about every queue manager in the cluster
  - Cluster information is stored in the form of messages on SYSTEM.CLUSTER.REPOSITORY.QUEUE
- Ensure that full repository queue managers are highly available
  - Avoid single point of failure
  - Define two full repositories to improve availability
  - Locate on highly available servers
  - Avoid other application workload if possible
- Full repository queue manager must be fully interconnected
  - Define a cluster-receiver channel at each queue manager
  - Define one cluster-sender channel between the full repository queue managers

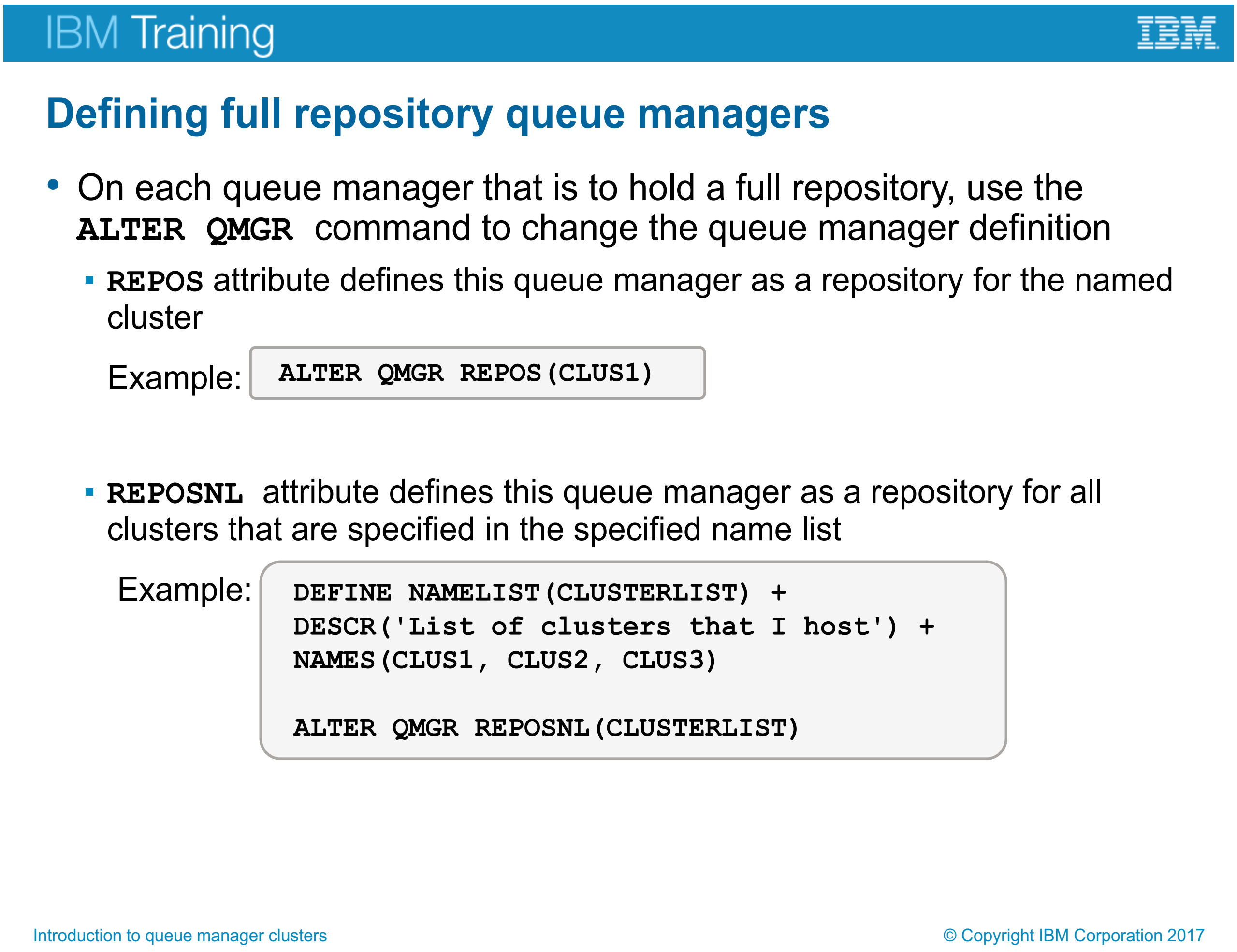
Introduction to queue manager clusters

© Copyright IBM Corporation 2017

Figure 12-9. Considerations for a full repository

It is best to have two full repositories in a cluster so that the cluster has a backup repository. You interconnect the full repository queue managers by defining cluster-sender channels between them.

When a queue manager sends out information about itself or requests information about another queue manager, the information or request is sent to the full repositories. A full repository that is named on a cluster-sender channel handles the request whenever possible, but if the chosen full repository is not available, another full repository is used. When the first full repository becomes available again, it collects the most recent information from the other full repository so that they contain the same information.



IBM Training

Defining full repository queue managers

- On each queue manager that is to hold a full repository, use the **`ALTER QMGR`** command to change the queue manager definition
  - **`REPOS`** attribute defines this queue manager as a repository for the named cluster

    Example:
    ```
    ALTER QMGR REPOS(CLUS1)
    ```
  - **`REPOSNL`** attribute defines this queue manager as a repository for all clusters that are specified in the specified name list

    Example:
    ```
    DEFINE NAMELIST(CLUSTERLIST) +
    DESCR('List of clusters that I host') +
    NAMES(CLUS1, CLUS2, CLUS3)

    ALTER QMGR REPOSNL(CLUSTERLIST)
    ```

Introduction to queue manager clusters © Copyright IBM Corporation 2017

Figure 12-10. Defining full repository queue managers

In each cluster, you must select at least one, preferably two, queue managers to hold full repositories. A cluster can work adequately with only one full repository, but two full repositories improve availability.

The full repository queue managers store information about all the queue managers in the cluster in the SYSTEM.CLUSTER.REPOSITORY.QUEUE. The full repository queue managers store queue manager information for 30 days. To prevent this data from expiring, the queue manager resends information about its configuration after 27 days. When data expires, it is not immediately removed; instead, it has a grace period of 60 days. If, during the grace period, no changes occur, then the data is removed. This grace period provides a queue manager that was temporarily out of service at the expiry date the right to rejoin if it does not stay disconnected from the cluster for 90 days. Then, that queue manager no longer is part of the cluster.

To designate a queue manager as a cluster repository queue manager, use the **`ALTER QMGR`** command.

To specify that a queue manager holds a full repository for one cluster, use the **`ALTER QMGR`** command and specify the **`REPOS(`***clustername***`)`** attribute. The value in the **`REPOS`** attribute is the name of the cluster. In the example in the figure, the cluster name is **`CLUS1`**.

To specify that a queue manager holds a full repository for several clusters, define a namelist that contains the names of the clusters. Then, use the **`REPOSNL(namelist)`** attribute on the **`ALTER QMGR`** command.

IBM Training

Cluster channels

- Cluster-receiver (CLUSRCVR)
  - Define one for each cluster queue manager
  - Enables queue managers to automatically define corresponding cluster-sender channels advertise the channel to the cluster
- Cluster-sender (CLUSDR)
  - Explicitly define one between the full repositories and one from each partial repository to one full repository
  - Other cluster-sender channels are defined automatically

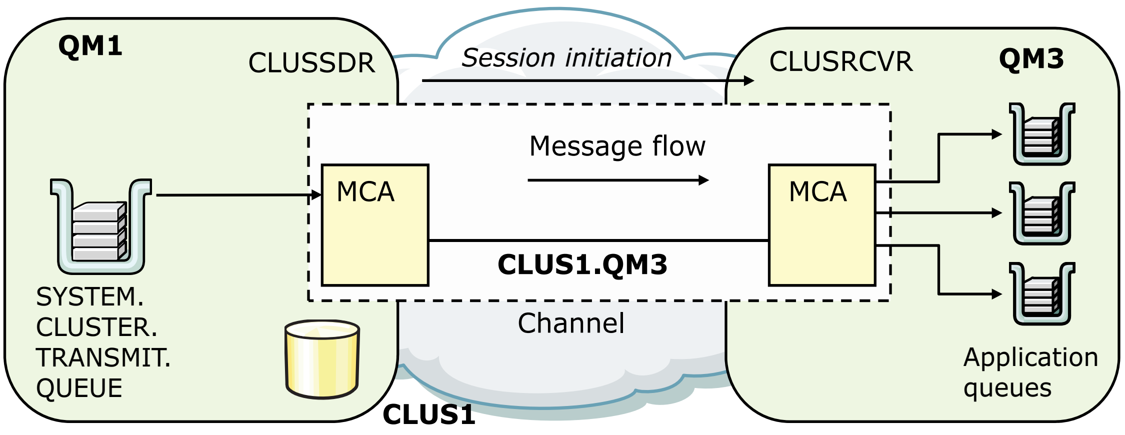
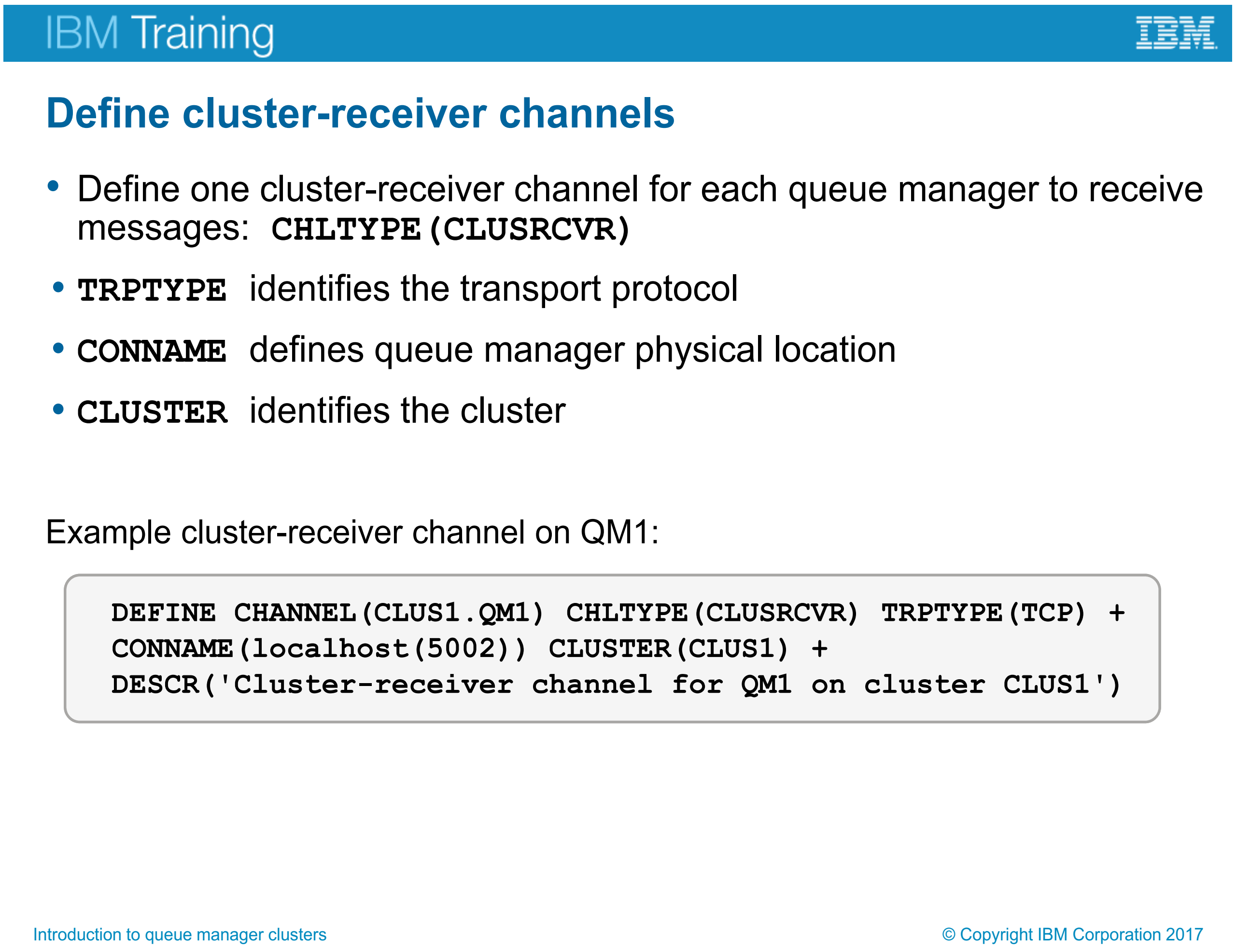


Figure 12-11. Cluster channels

In a cluster, each queue manager has a cluster-receiver channel on which it can receive messages and information about the cluster. You must manually define a cluster-receiver channel on each queue manager in the cluster.

In a cluster, each queue manager has a cluster-sender channel on which it can send cluster information to one of the full repository queue managers. Queue managers can also send messages to other queue managers on cluster-sender channels. Manually define a cluster-sender channel between the full repository queue managers and from each partial repository queue manager to one of the full repository queue managers. The other cluster-sender channels are defined automatically.



IBM Training

Define cluster-receiver channels

- Define one cluster-receiver channel for each queue manager to receive messages: **CHLTYPE(CLUSRCVR)**
- **TRPTYPE** identifies the transport protocol
- **CONNAME** defines queue manager physical location
- **CLUSTER** identifies the cluster

Example cluster-receiver channel on QM1:

```
DEFINE CHANNEL(CLUS1.QM1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
CONNAME(localhost(5002)) CLUSTER(CLUS1) +
DESCR('Cluster-receiver channel for QM1 on cluster CLUS1')
```

Introduction to queue manager clusters

© Copyright IBM Corporation 2017

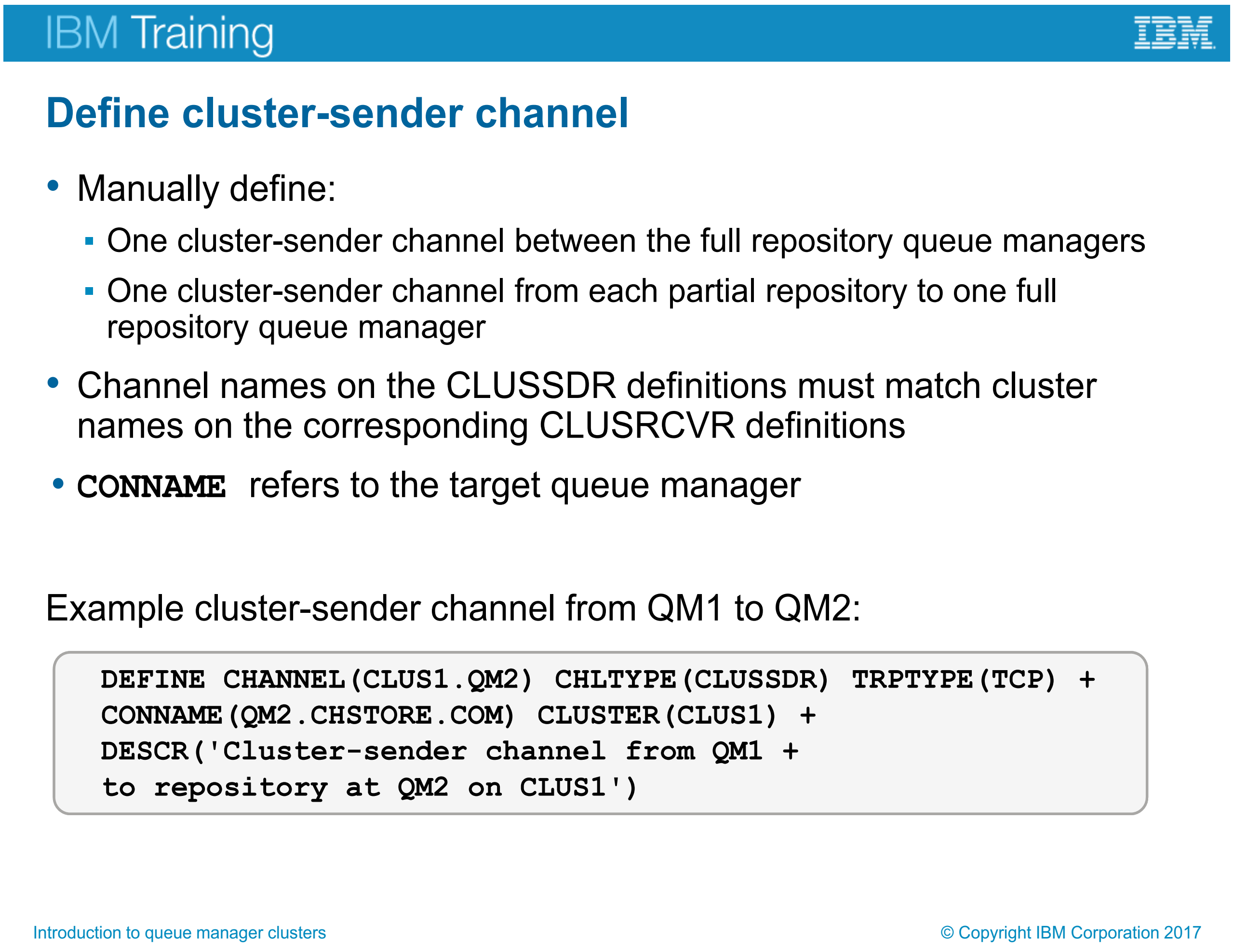
Figure 12-12. Define cluster-receiver channels

A cluster-receiver channel is a channel definition of the **TYPE(CLUSRCVR)** on which a cluster queue manager can receive messages from within the cluster. Through the definition of this object, a queue manager is advertised to the other queue managers in the cluster so that they can automatically define the appropriate cluster-sender (**CLUSSDR**) channels. At least one cluster-receiver channel is required for each cluster queue manager.

Each cluster channel is given a unique name. The **CLUSTER** attribute identifies the cluster, which in this example is CLUS1.

The naming convention that is used in this example includes the cluster name and the queue manager name in the channel definition. This convention makes administration easier in cases where the queue manager is shared with other clusters. In this example, the channel name is CLUS1.QM1.

The connection name (**CONNAME**) is the network address and port of the queue manager server. The network address can be entered as a DNS host name, or an IP address. If the port number is not specified, the default TCP listener port for MQ (1414) is used.



IBM Training

Define cluster-sender channel

- Manually define:
  - One cluster-sender channel between the full repository queue managers
  - One cluster-sender channel from each partial repository to one full repository queue manager
- Channel names on the CLUSSDR definitions must match cluster names on the corresponding CLUSRCVR definitions
- **`CONNAME`** refers to the target queue manager

Example cluster-sender channel from QM1 to QM2:

```
DEFINE CHANNEL(CLUS1.QM2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
CONNAME(QM2.CHSTORE.COM) CLUSTER(CLUS1) +
DESCR('Cluster-sender channel from QM1 +
to repository at QM2 on CLUS1')
```

Introduction to queue manager clusters

© Copyright IBM Corporation 2017

Figure 12-13. Define cluster-sender channel

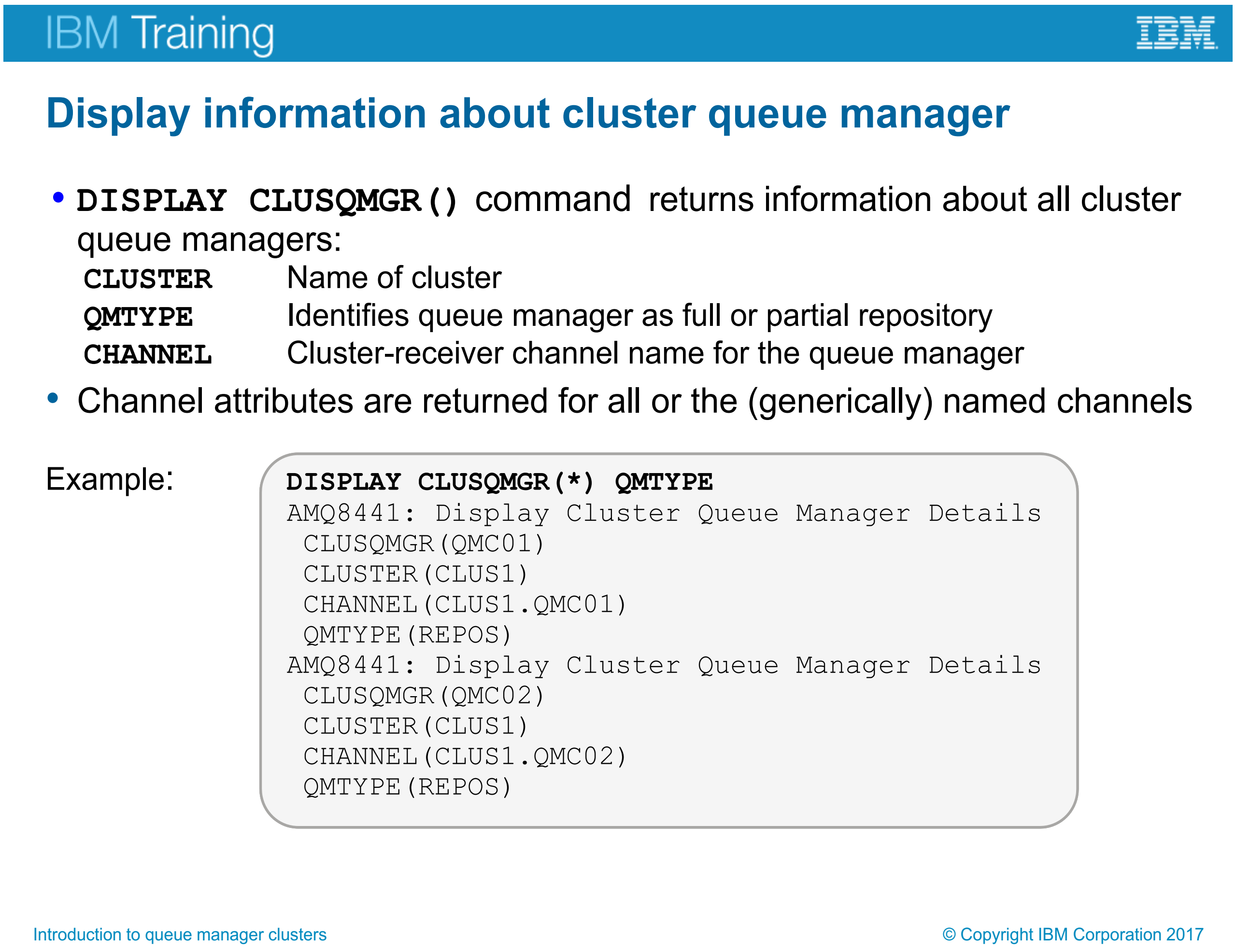
A cluster-sender channel is a channel definition of the **`TYPE(CLUSSDR)`** on which a cluster queue manager can send messages to another queue manager in the cluster. This channel is used to notify the repository of any changes of the status of the queue manager, for example, the addition or removal of a queue. This channel is also used to send messages to the other queue managers in the cluster.

Manually define one cluster-sender channel from each queue manager in the cluster to one of the full repository queue managers. The other cluster-sender channels are created automatically when the queue manager needs the channel. Manually defining the other cluster-sender channels results in duplicate channels.

For a group of full repositories to be fully connected, manually define one cluster-sender channel from each full repository to the other full repositories.

The cluster-sender name must match the cluster-receiver name on the other end of the channel.

The figure includes an example for defining a cluster-sender channel with the **`DEFINE CHANNEL`** command and TCP as the transport protocol.



IBM Training

Display information about cluster queue manager

- **`DISPLAY CLUSQMGR()`** command returns information about all cluster queue managers:

CLUSTER	Name of cluster
QMTYPE	Identifies queue manager as full or partial repository
CHANNEL	Cluster-receiver channel name for the queue manager

- Channel attributes are returned for all or the (generically) named channels

Example:

```
DISPLAY CLUSQMGR(*) QMTYPE
AMQ8441: Display Cluster Queue Manager Details
 CLUSQMGR(QMC01)
 CLUSTER(CLUS1)
 CHANNEL(CLUS1.QMC01)
 QMTYPE(REPOS)
AMQ8441: Display Cluster Queue Manager Details
 CLUSQMGR(QMC02)
 CLUSTER(CLUS1)
 CHANNEL(CLUS1.QMC02)
 QMTYPE(REPOS)
```

Introduction to queue manager clusters

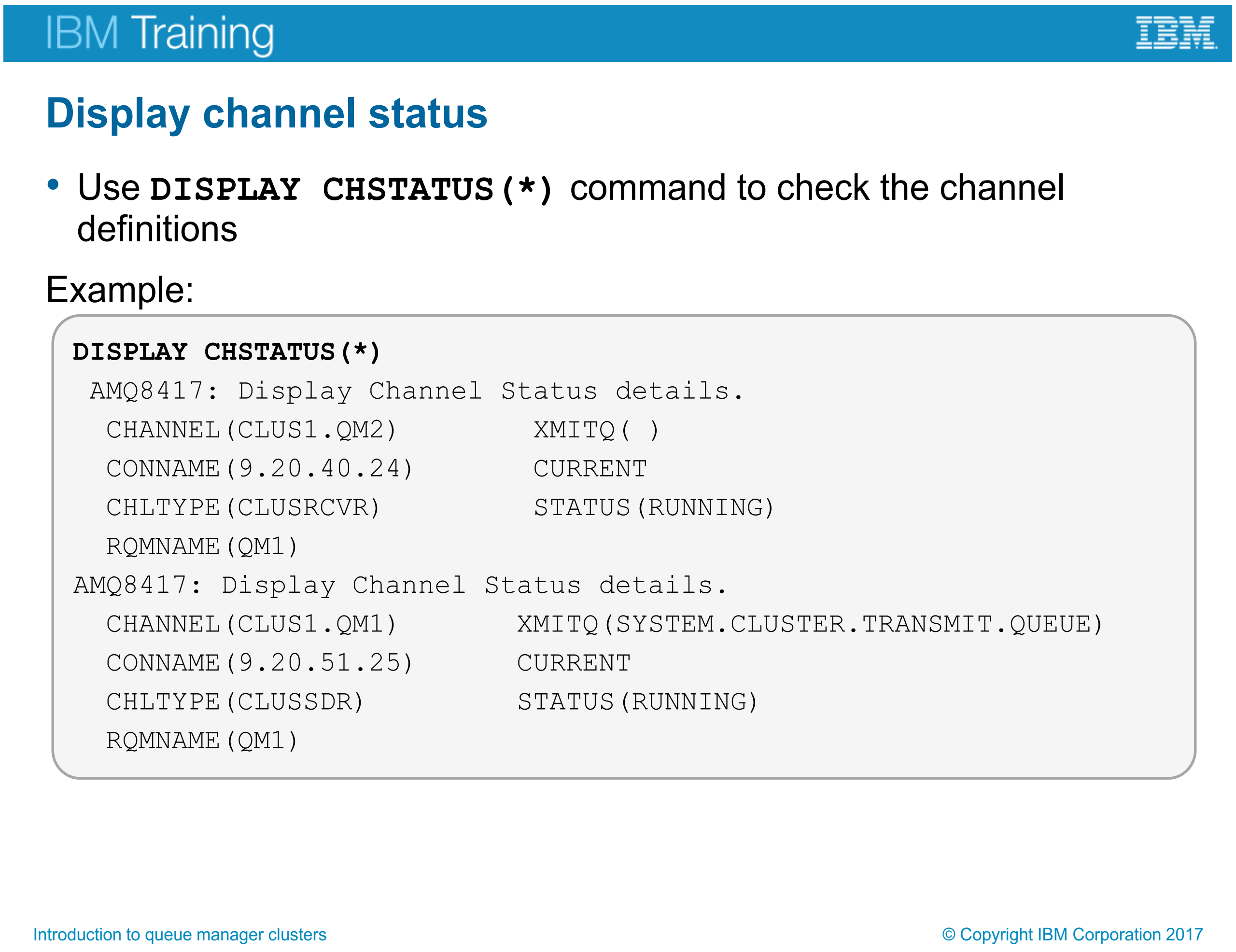
© Copyright IBM Corporation 2017

Figure 12-14. Display information about cluster queue manager

The **DISPLAY CLUSMGR** command returns cluster information about queue managers in a cluster, which is stored in the local SYSTEM.CLUSTER.REPOSITORY.QUEUE.

The **DEFTYPE** (definition type) parameter can be set to one of the following options:

- **CLUSSDR**: Cluster-sender channel from an explicit definition
- **CLUSSDRA**: Cluster-sender by automatic definition
- **CLUSSDRB**: Cluster-sender channel, both from an explicit definition and by automatic definition
- **CLUSRCVR**: Cluster-receiver channel



IBM Training

Display channel status

- Use **`DISPLAY CHSTATUS(*)`** command to check the channel definitions

Example:

```
DISPLAY CHSTATUS(*)
 AMQ8417: Display Channel Status details.
  CHANNEL(CLUS1.QM2)         XMITQ( )
  CONNAME(9.20.40.24)        CURRENT
  CHLTYPE(CLUSRCVR)          STATUS(RUNNING)
  RQMNAME(QM1)
AMQ8417: Display Channel Status details.
  CHANNEL(CLUS1.QM1)        XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
  CONNAME(9.20.51.25)       CURRENT
  CHLTYPE(CLUSSDR)          STATUS(RUNNING)
  RQMNAME(QM1)
```

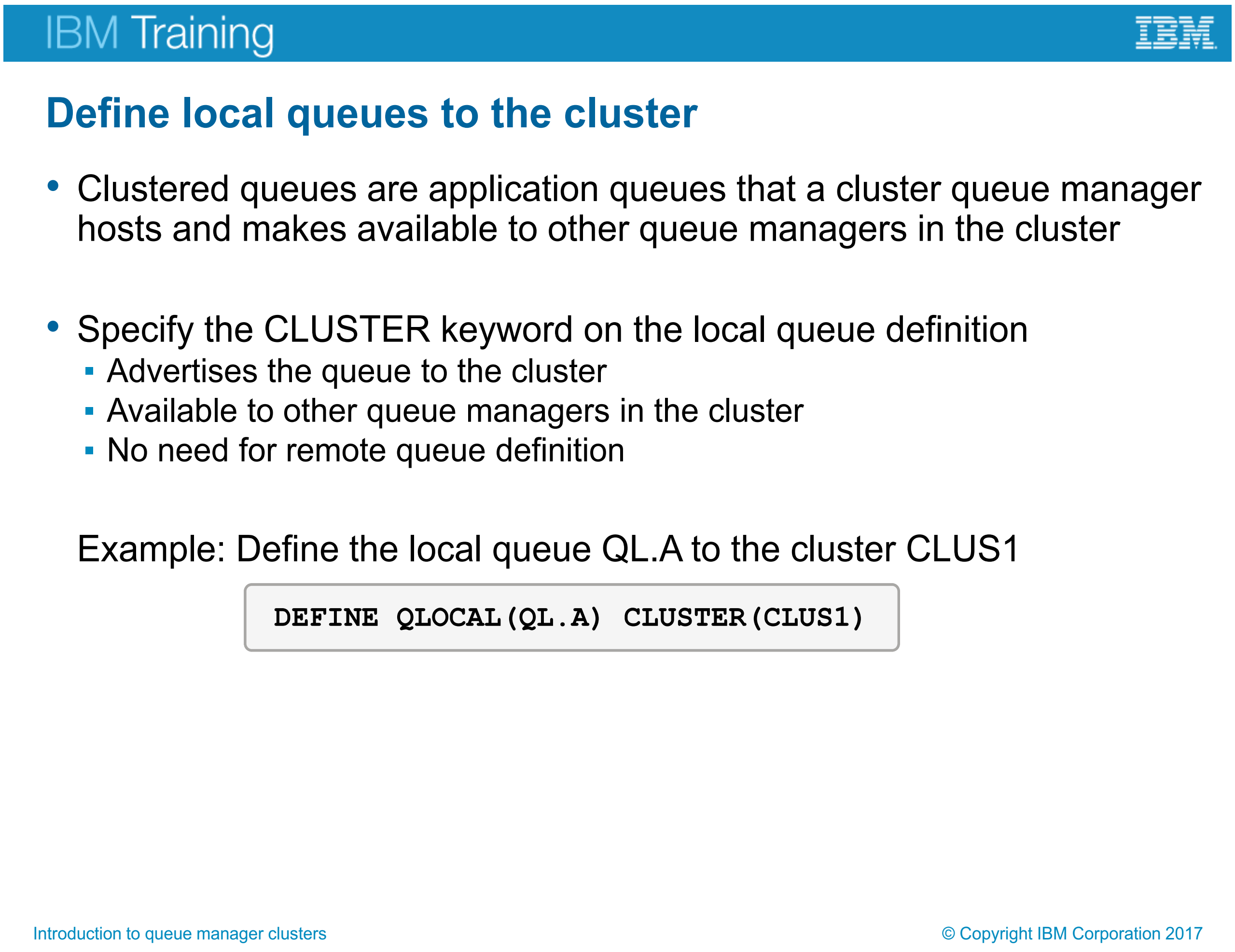
Introduction to queue manager clusters

© Copyright IBM Corporation 2017

Figure 12-15. Display channel status

Check the channel status for the cluster by entering: **`DISPLAY CHSTATUS(*)`**

As shown in the figure, the channel status includes the channel name, transmission queue name, connection name, channel type, status, and remote queue manager name.



IBM Training

Define local queues to the cluster

- Clustered queues are application queues that a cluster queue manager hosts and makes available to other queue managers in the cluster
- Specify the CLUSTER keyword on the local queue definition
  - Advertises the queue to the cluster
  - Available to other queue managers in the cluster
  - No need for remote queue definition

Example: Define the local queue QL.A to the cluster CLUS1

```
DEFINE QLOCAL(QL.A) CLUSTER(CLUS1)
```

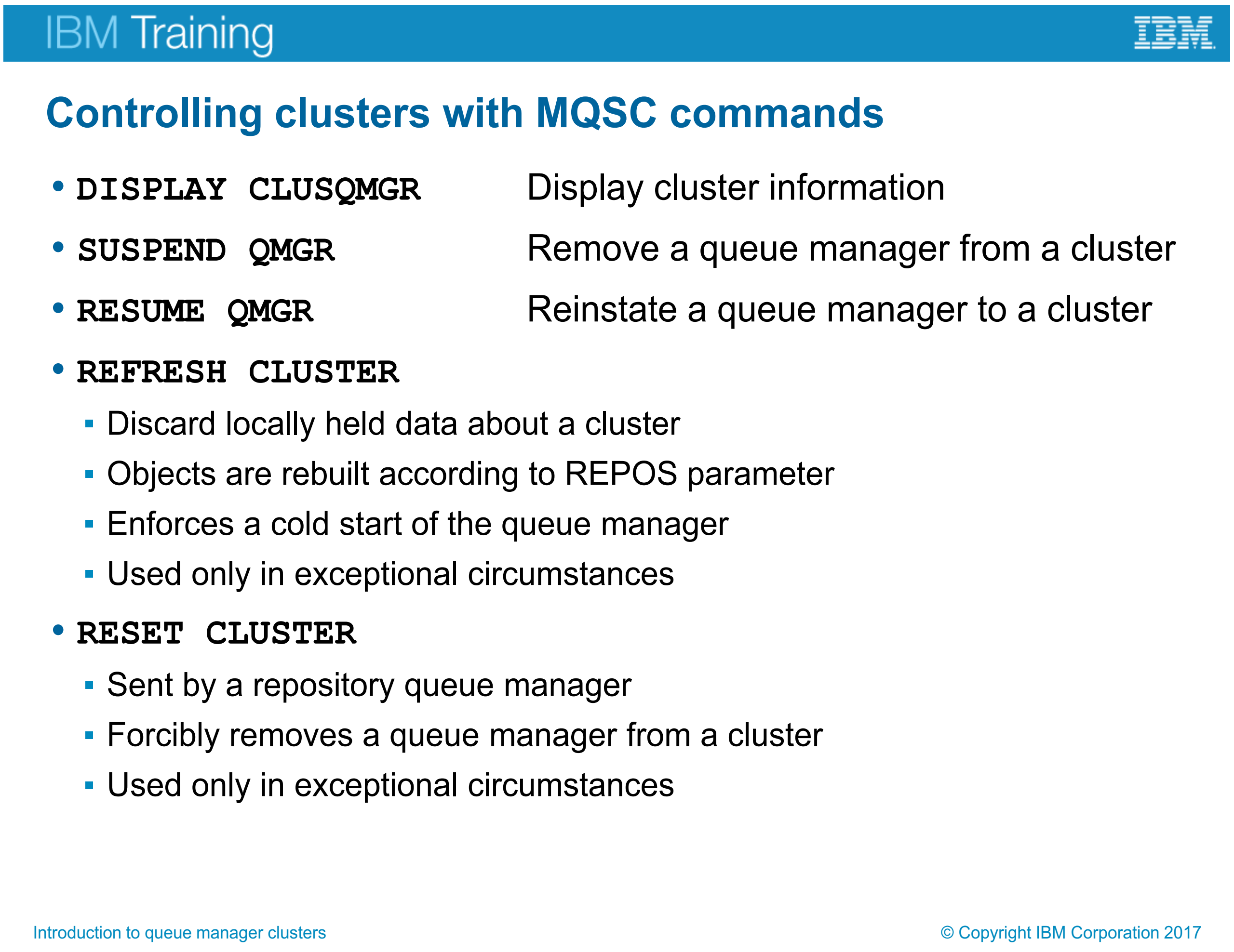
Introduction to queue manager clusters © Copyright IBM Corporation 2017

Figure 12-16. Define local queues to the cluster

A *cluster queue* is a queue that a cluster queue manager hosts and is accessible by queue managers in the cluster. The local queue is either preexisting or created on the local queue manager. The other queue managers can see this queue and use it to put messages to it without the use of remote queue definition. The cluster queue can be advertised in more than one cluster.

It is not necessary to define remote-queue definitions for cluster queues. Applications can put messages to any cluster queue in the cluster. They can receive responses to their messages by providing a reply-to queue and specifying its name when they PUT the message on the queue.

The **`CLUSTER`** option on the **`DEFINE QLOCAL`** command identifies the cluster.



IBM Training

Controlling clusters with MQSC commands

• **`DISPLAY CLUSQMGR`**	Display cluster information
• **`SUSPEND QMGR`**	Remove a queue manager from a cluster
• **`RESUME QMGR`**	Reinstate a queue manager to a cluster

- **`REFRESH CLUSTER`**
  - Discard locally held data about a cluster
  - Objects are rebuilt according to REPOS parameter
  - Enforces a cold start of the queue manager
  - Used only in exceptional circumstances
- **`RESET CLUSTER`**
  - Sent by a repository queue manager
  - Forcibly removes a queue manager from a cluster
  - Used only in exceptional circumstances

Introduction to queue manager clusters © Copyright IBM Corporation 2017

Figure 12-17. Controlling clusters with MQSC commands

The figure lists the MQSC commands that you can use to control clusters:

- **`DISPLAY CLUSQMGR`**

  If you enter this command from a queue manager with a full repository, the information that is returned is for every queue manager in the cluster. If you enter this command from a queue manager that does not have a full repository, the information that is returned is for only the queue managers in which it has an interest.

- **`SUSPEND QMGR`** and **`RESUME QMGR`**

  Use the **`SUSPEND QMGR`** command to remove a queue manager from a cluster temporarily. Then, use the **`RESUME QMGR`** command to reinstate into the cluster. The **`SUSPEND`** command does not completely stop messages from being sent to a queue manager. If the suspended queue manager is the only queue manager with an available copy of a queue, the messages are sent to the queue on the suspended queue manager.

- **`REFRESH CLUSTER`**

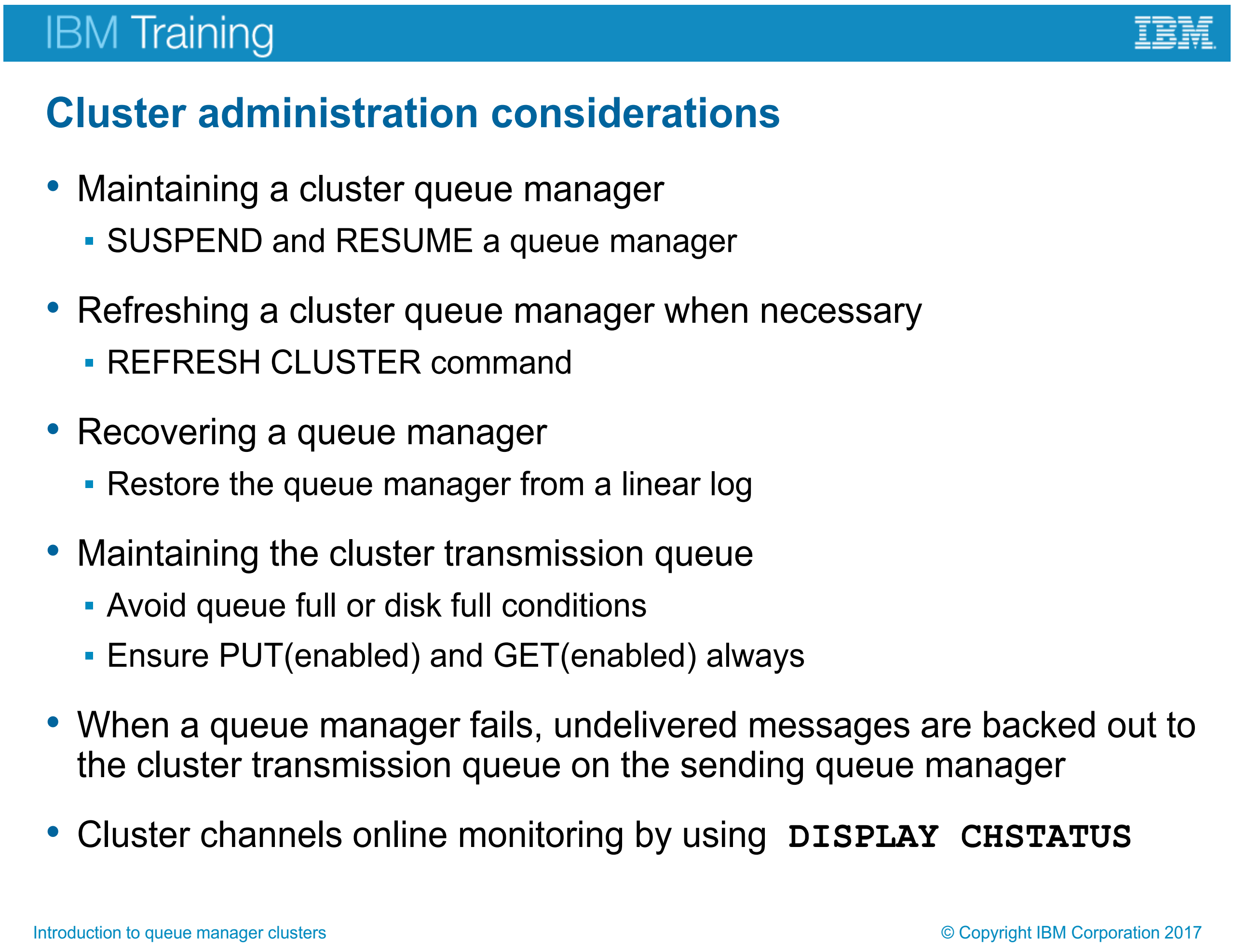
  You can enter **`REFRESH CLUSTER(*)`** to refresh the queue manager in all of the clusters that it is a member of. If used with **`REPOS(YES)`**, this command forces the queue manager to restart its search for full repositories from the information in the local cluster-sender definitions. The search occurs even if the cluster-sender channel connects the queue manager to several clusters.

- **RESET CLUSTER**

  In an emergency where a queue manager is temporarily damaged, you might want to inform the rest of the cluster before the other queue managers try to send it messages. The **RESET CLUSTER** command can be used to remove the damaged queue manager. The **RESET CLUSTER** command can also be used to remove a queue manager that does not belong to a cluster, perhaps because of a security problem. If necessary, you can use the **REFRESH CLUSTER** command to reverse the effect of **RESET CLUSTER** and put the queue manager back into the cluster.

Important

Use the commands to suspend, refresh, and reset the cluster with caution.



IBM Training

Cluster administration considerations

- Maintaining a cluster queue manager
  - SUSPEND and RESUME a queue manager
- Refreshing a cluster queue manager when necessary
  - REFRESH CLUSTER command
- Recovering a queue manager
  - Restore the queue manager from a linear log
- Maintaining the cluster transmission queue
  - Avoid queue full or disk full conditions
  - Ensure PUT(enabled) and GET(enabled) always
- When a queue manager fails, undelivered messages are backed out to the cluster transmission queue on the sending queue manager
- Cluster channels online monitoring by using **`DISPLAY CHSTATUS`**

Introduction to queue manager clusters © Copyright IBM Corporation 2017

Figure 12-18. Cluster administration considerations

At some point, you need to back up the queue manager data or apply software updates. If the queue manager hosts any queues, its activities must be suspended. When the maintenance is complete, the queue manager activities can be resumed.

The **`SUSPEND`** command does not completely stop messages from being sent to a queue manager. If only the suspended queue manager has an available copy of a queue, the messages are sent to the queue on the suspended queue manager.

The **`REFRESH CLUSTER`** command allows a queue manager to be restarted regarding its full repository content. MQ ensures that no data is lost from your queues.

To restore from a point-in-time backup, enter the **`REFRESH CLUSTER`** command on the restored queue manager for all clusters in which the queue manager participates.

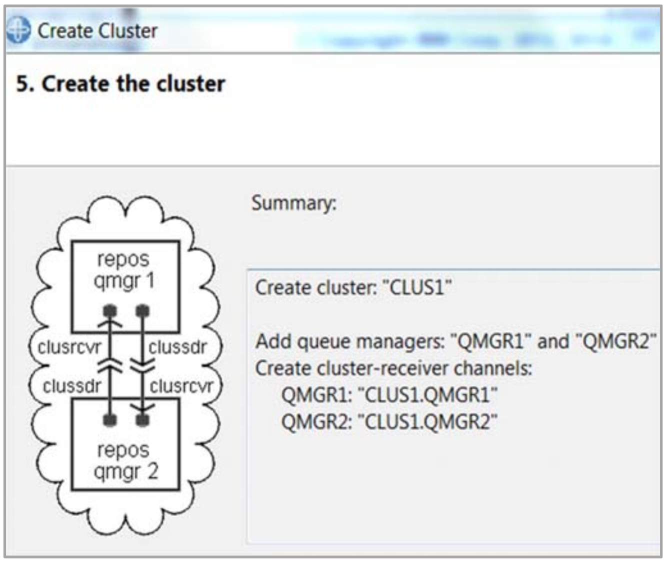
The availability and performance of the cluster transmission queue are essential to the performance of clusters. Make sure that it does not become full, and do not change this queue to get-disabled or put-disabled.

The **`DISPLAY CHSTATUS`** is a useful command when you troubleshoot connection problems in a cluster, such as identifying duplicate channels or mismatches in channel names.

IBM Training

Defining a cluster by using IBM MQ Explorer

- **Create Cluster** wizard
  1. Specify cluster name
  2. Identify two full repository queue managers
  3. Specify first full repository cluster-receiver channel name and connection details
  4. Specify second full repository cluster-receiver channel name and connection details
- Prerequisites:
  - Each full repository queue manager in the cluster must have a running listener
  - You must know the connection details of each full repository queue manager in the cluster
- If the full repository queue managers belong to another cluster, you cannot use the **Create Cluster** wizard



5. Create the cluster

Summary:

Create cluster: "CLUS1"

Add queue managers: "QMGR1" and "QMGR2"
Create cluster-receiver channels:
QMGR1: "CLUS1.QMGR1"
QMGR2: "CLUS1.QMGR2"

Figure 12-19. Defining a cluster by using IBM MQ Explorer

You can define a cluster by using the **Create Cluster** wizard in IBM MQ Explorer.

Before you use the Create Cluster wizard, you must create the queue managers and the queue manager must be running. You must also ensure that remote queue managers are accessible by IBM MQ Explorer.

IBM Training

Controlling clusters with IBM MQ Explorer

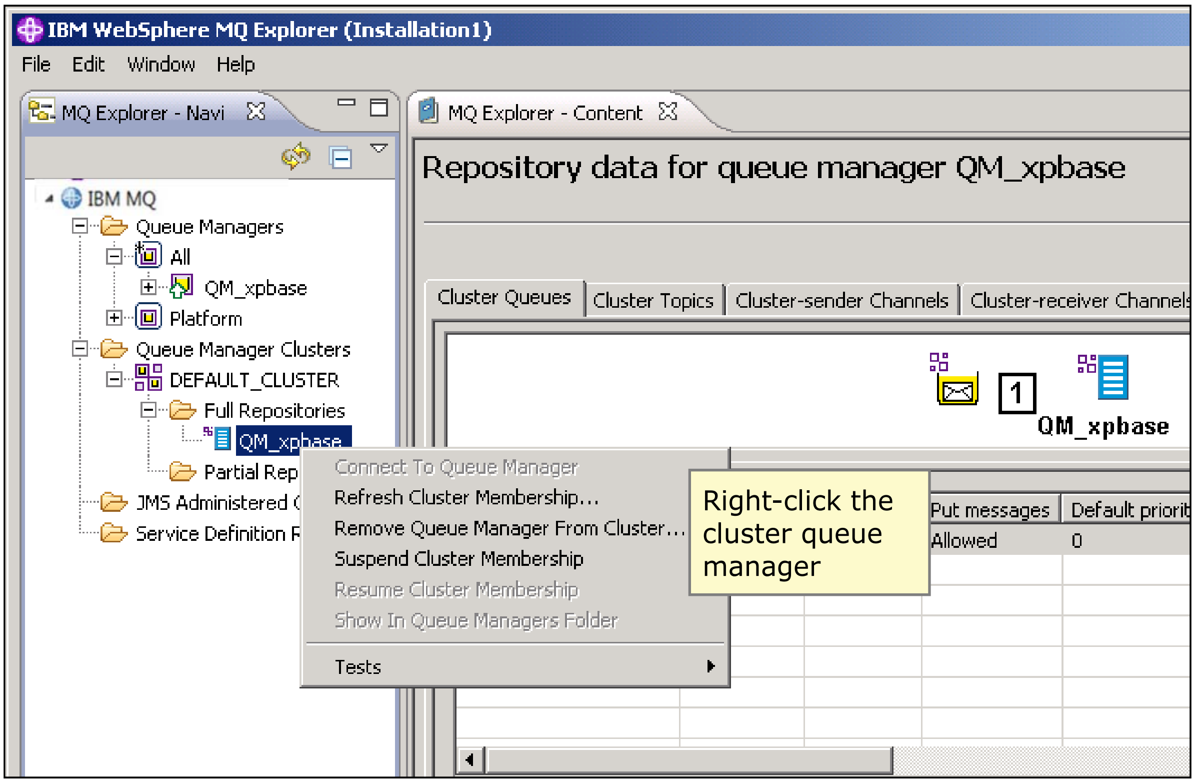


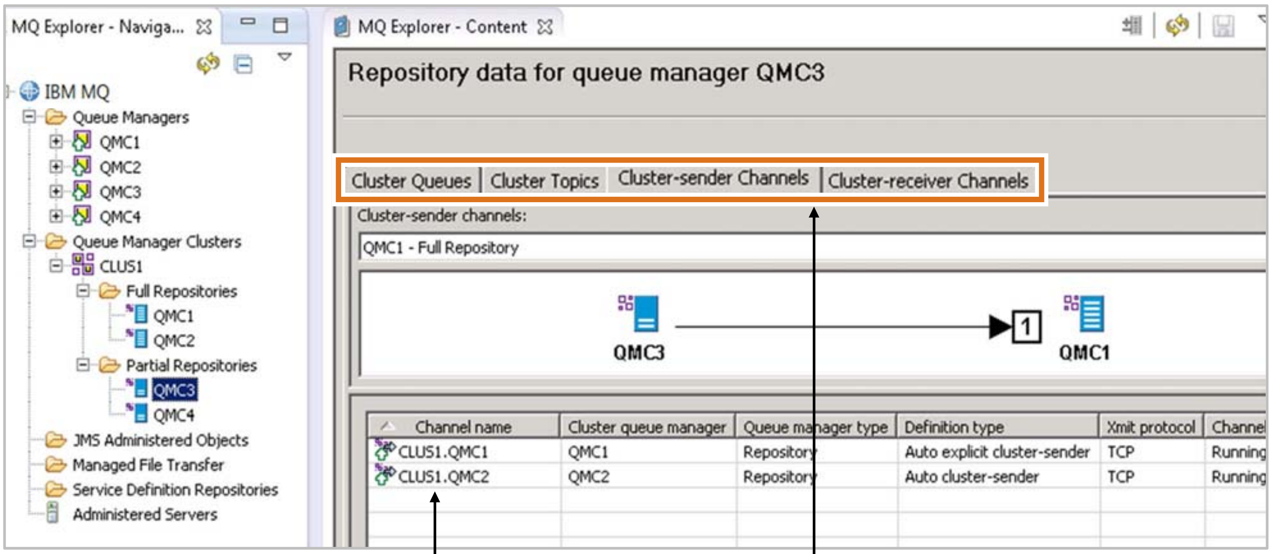
Figure 12-20. Controlling clusters with IBM MQ Explorer

You can use IBM MQ Explorer to refresh and suspend cluster membership, and remove a queue manager from a cluster. The queue managers that are in a cluster are listed in the **Queue Managers Clusters** folder in the **MQ Explorer - Navigator**.

Cluster management by IBM MQ Explorer assumes that any remote servers are accessible from IBM MQ Explorer.

IBM Training

Monitoring clusters with IBM MQ Explorer



Repository data for queue manager QMC3

Cluster-sender channels:

QMC1 - Full Repository

Channel name	Cluster queue manager	Queue manager type	Definition type	Xmit protocol	Channel
CLUS1.QMC1	QMC1	Repository	Auto explicit cluster-sender	TCP	Running
CLUS1.QMC2	QMC2	Repository	Auto cluster-sender	TCP	Running

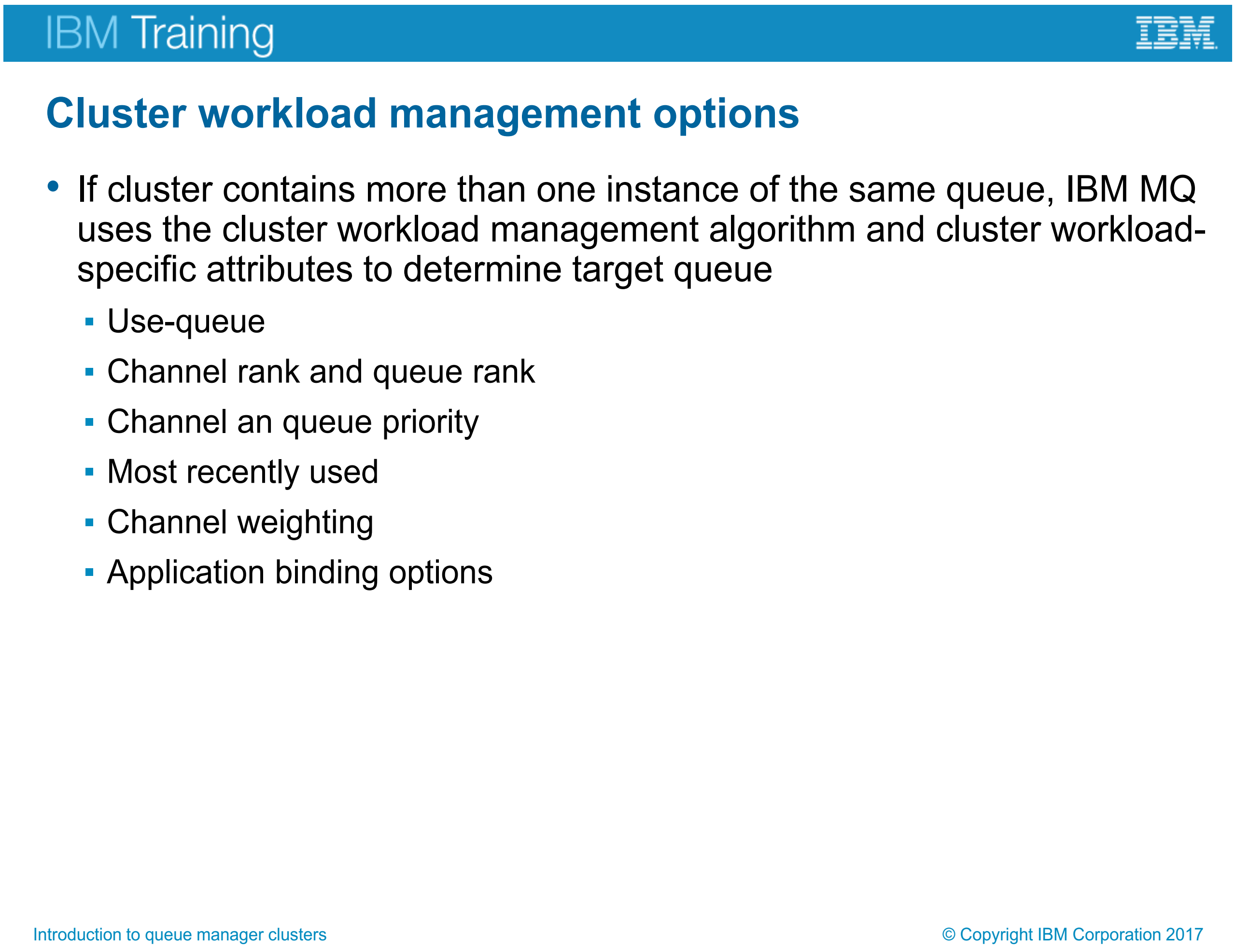
Double-click channel name to view properties

Click tabs to access additional information about cluster queues, cluster topics, cluster-sender channels, and cluster-receiver channels

Introduction to queue manager clusters © Copyright IBM Corporation 2017

Figure 12-21. Monitoring clusters with IBM MQ Explorer

When a cluster queue manager is selected, tabs in the MQ Explorer **Content** view can be selected to access additional information about the cluster. Double-click an object, such as the channel name, to view its properties.



IBM Training

Cluster workload management options

- If cluster contains more than one instance of the same queue, IBM MQ uses the cluster workload management algorithm and cluster workload-specific attributes to determine target queue
  - Use-queue
  - Channel rank and queue rank
  - Channel an queue priority
  - Most recently used
  - Channel weighting
  - Application binding options

Introduction to queue manager clusters © Copyright IBM Corporation 2017

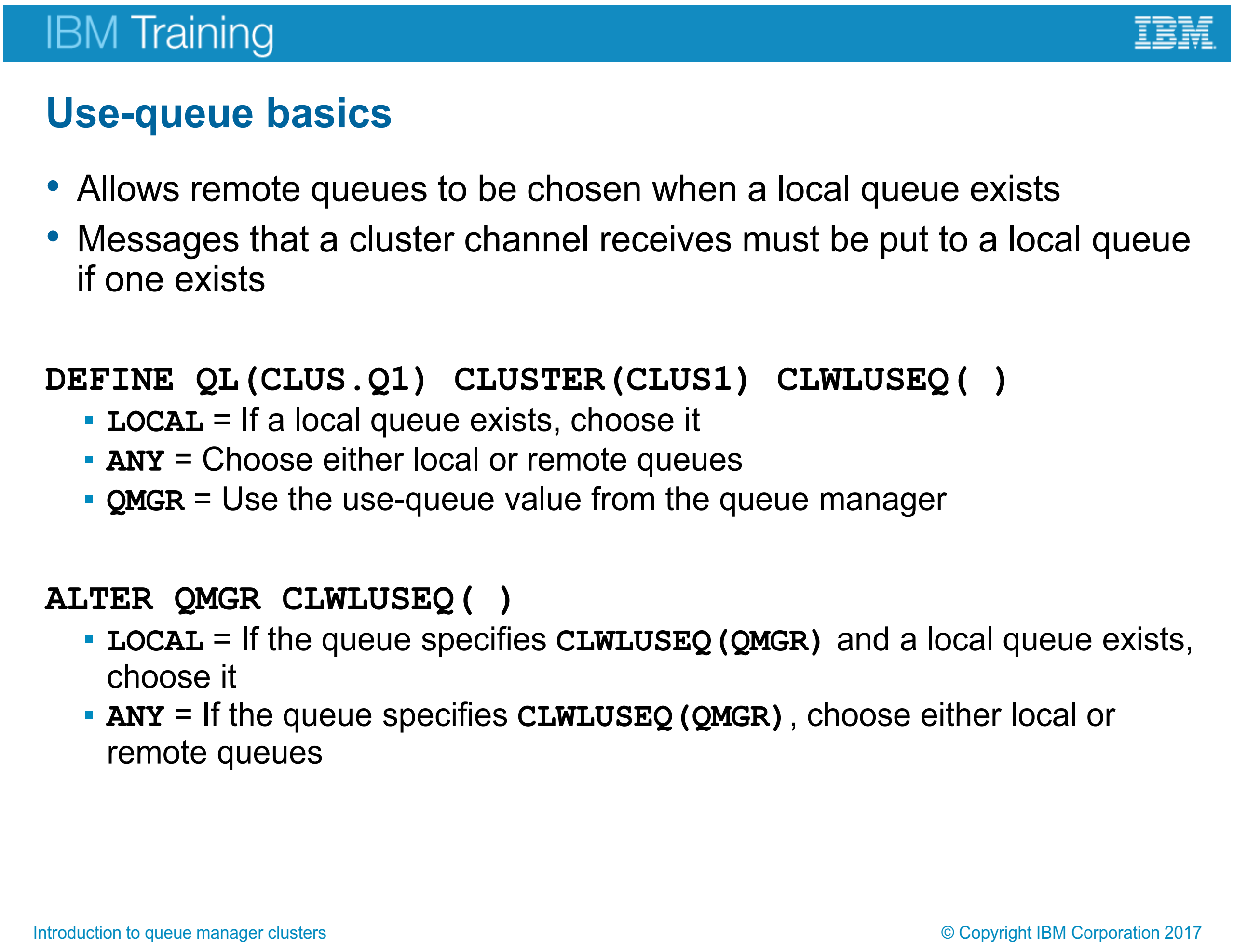
Figure 12-22. Cluster workload management options

Cluster support allows more than one queue manager to host an occurrence of the same queue. So, two or more queue managers can be clones of each other, capable of running the same applications and having local definitions of the same queues. This technique can be used to implement queues for two reasons:

- To increase the capacity available to process messages
- To introduce an ability to fail over work from one server to another and improve availability of the service

This architecture can be used to spread the workload between queue managers, if applications allow it to do so.

With multiple choices, which are based on availability and channel priorities, a built-in workload management algorithm determines the remote queue manager. A local occurrence takes precedence by default. The figure lists the cluster workload options that can be specified on channel and queue definitions. Cluster workload management is taught in detail in course WM213, *IBM MQ V9 Advanced System Administration*.



IBM Training

Use-queue basics

- Allows remote queues to be chosen when a local queue exists
- Messages that a cluster channel receives must be put to a local queue if one exists

```
DEFINE QL(CLUS.Q1) CLUSTER(CLUS1) CLWLUSEQ( )
```

- **`LOCAL`** = If a local queue exists, choose it
- **`ANY`** = Choose either local or remote queues
- **`QMGR`** = Use the use-queue value from the queue manager

```
ALTER QMGR CLWLUSEQ( )
```

- **`LOCAL`** = If the queue specifies **`CLWLUSEQ(QMGR)`** and a local queue exists, choose it
- **`ANY`** = If the queue specifies **`CLWLUSEQ(QMGR)`**, choose either local or remote queues

Introduction to queue manager clusters © Copyright IBM Corporation 2017

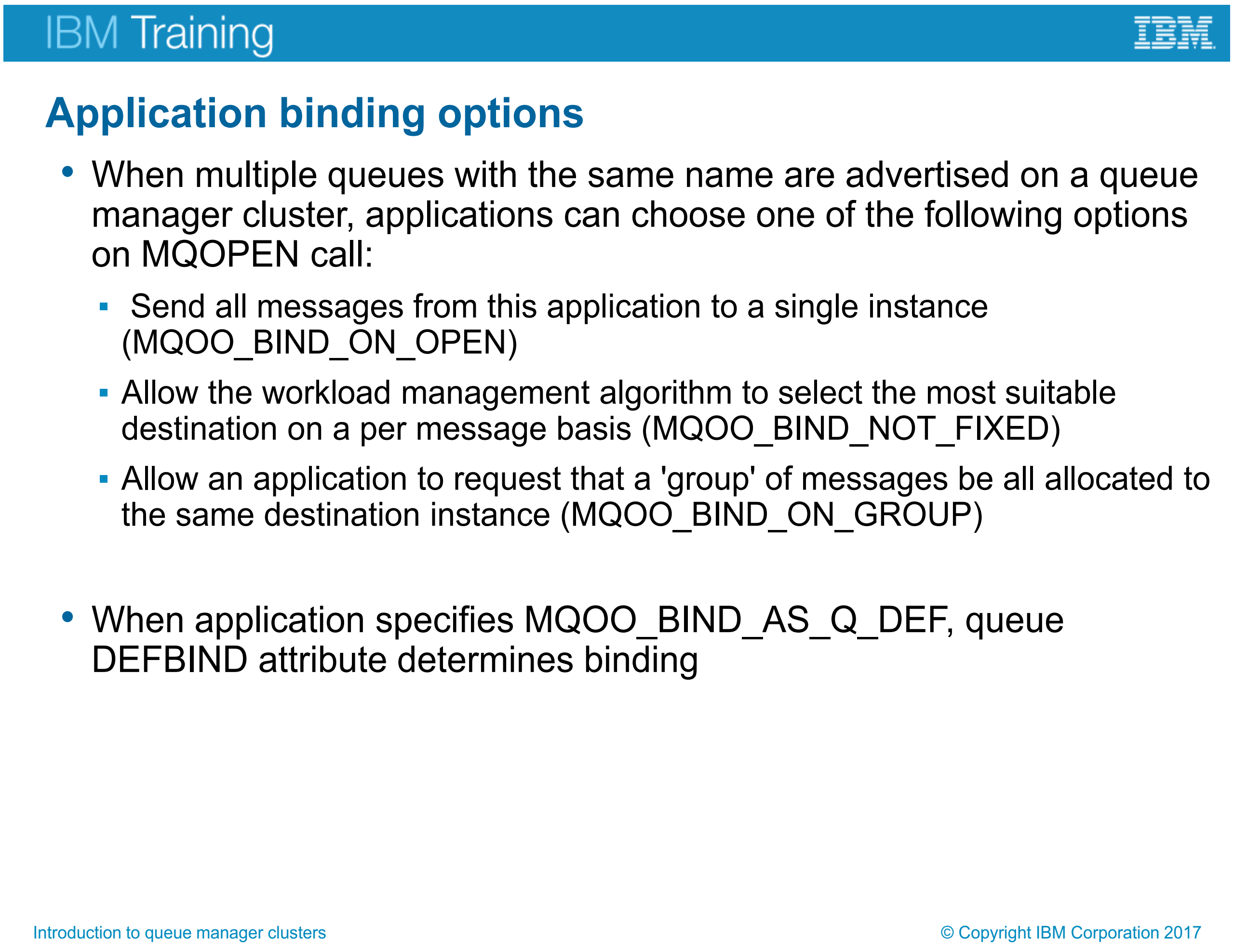
Figure 12-23. Use-queue basics

The use-queue attribute, **`CLWLUSEQ`**, specifies the behavior of an MQPUT operation when the target queue has a local instance and at least one remote cluster instance in most cases. The exception is in cases where the MQPUT originates from a cluster channel.

The valid options for the use-queue attribute on a queue are **`LOCAL`**, **`ANY`**, and **`QMGR`**.

- If you specify **`QMGR`** for the attribute value on the queue, the **`CLWLUSEQ`** parameter of the queue manager definition determines the behavior. This parameter is valid only for local queues.
- If you specify **`ANY`**, the queue manager treats the local queue as another instance of the cluster queue for the purposes of workload distribution.
- If you specify **`LOCAL`**, the local queue is the only target of the MQPUT operation.

The valid options for the use-queue attribute on a queue manager are **`LOCAL`** and **`ANY`**.



IBM Training

Application binding options

- When multiple queues with the same name are advertised on a queue manager cluster, applications can choose one of the following options on MQOPEN call:
  - Send all messages from this application to a single instance (MQOO_BIND_ON_OPEN)
  - Allow the workload management algorithm to select the most suitable destination on a per message basis (MQOO_BIND_NOT_FIXED)
  - Allow an application to request that a 'group' of messages be all allocated to the same destination instance (MQOO_BIND_ON_GROUP)
- When application specifies MQOO_BIND_AS_Q_DEF, queue DEFBIND attribute determines binding

Introduction to queue manager clusters © Copyright IBM Corporation 2017

Figure 12-24. Application binding options

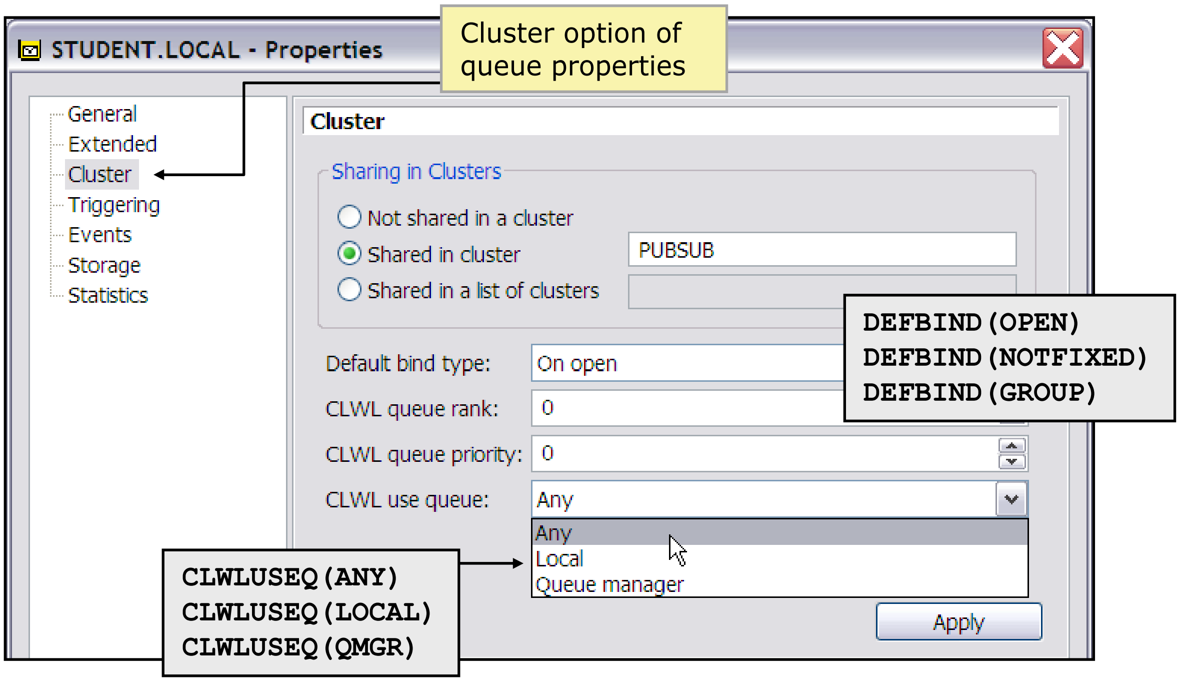
An option on the MQOPEN call allows the application to specify that the target queue manager needs to be fixed when there are multiple instances of the same queue within a cluster. That is, all messages that are put to the queue that specifies the object handle that is returned from the MQOPEN call must be directed to the same queue manager, by using the same route.

If it is not necessary for the application to force all the messages to be written to the same destination, specify BIND_NOT_FIXED on the MQOPEN call. This configuration selects a destination at MQPUT time, that is, on a message-by-message basis.

If the application does not specify BIND_ON_OPEN, BIND_NOT_FIXED, or BIND_ON_GROUP, the default option is BIND_AS_Q_DEF. Using BIND_AS_Q_DEF takes the binding that is used for the queue handle from the DEFBIND queue attribute.

IBM Training

Using IBM MQ Explorer to specify cluster queue properties


STUDENT.LOCAL - Properties
Cluster option of queue properties
General
Extended
Cluster
Triggering
Events
Storage
Statistics
Apply

Cluster

Sharing in Clusters

- Not shared in a cluster
- ● Shared in cluster: PUBSUB
- Shared in a list of clusters

Default bind type: On open

CLWL queue rank: 0

CLWL queue priority: 0

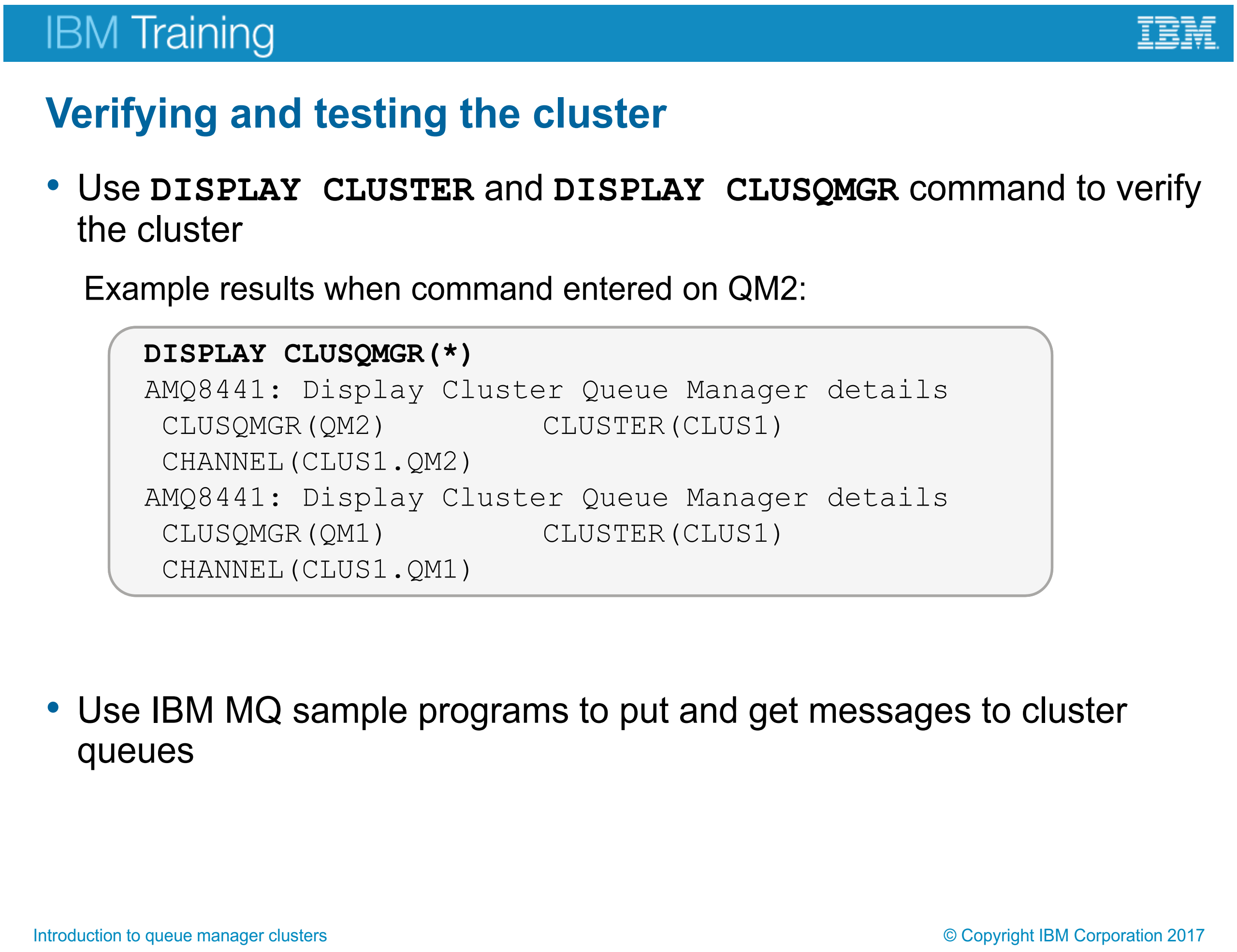
CLWL use queue: Any (Any, Local, Queue manager)

```
DEFBIND(OPEN)
DEFBIND(NOTFIXED)
DEFBIND(GROUP)
```

```
CLWLUSEQ(ANY)
CLWLUSEQ(LOCAL)
CLWLUSEQ(QMGR)
```

Figure 12-25. Using IBM Explorer to specify cluster queue properties

The figure shows how to configure the cluster default bind and workload use-queue attribute for a queue in the queue **Cluster** properties.



IBM Training

Verifying and testing the cluster

- Use **`DISPLAY CLUSTER`** and **`DISPLAY CLUSQMGR`** command to verify the cluster

  Example results when command entered on QM2:

```
DISPLAY CLUSQMGR(*)
AMQ8441: Display Cluster Queue Manager details
 CLUSQMGR(QM2)          CLUSTER(CLUS1)
 CHANNEL(CLUS1.QM2)
AMQ8441: Display Cluster Queue Manager details
 CLUSQMGR(QM1)          CLUSTER(CLUS1)
 CHANNEL(CLUS1.QM1)
```

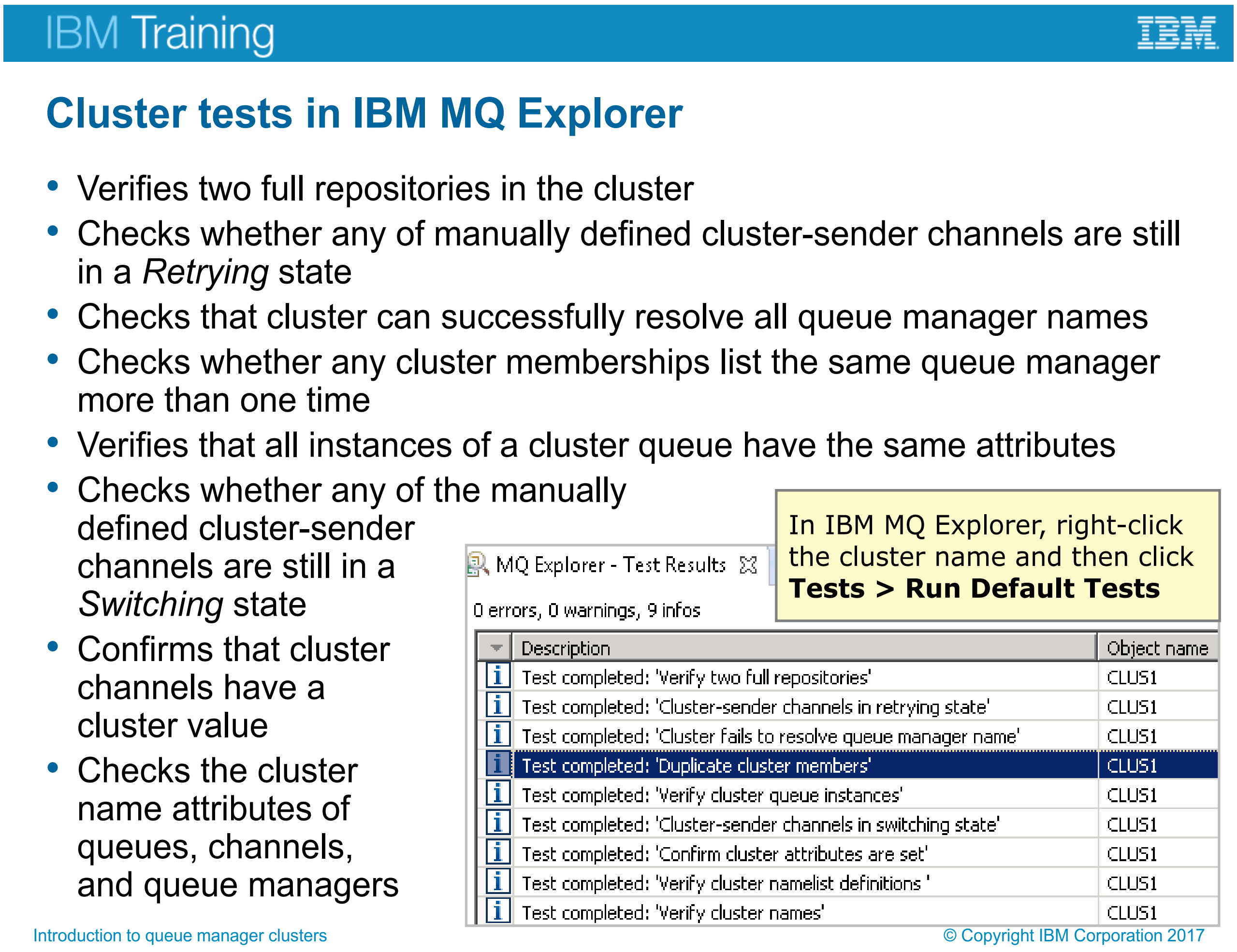
- Use IBM MQ sample programs to put and get messages to cluster queues

Introduction to queue manager clusters © Copyright IBM Corporation 2017

Figure 12-26. Verifying and testing the cluster

You can verify the cluster configuration by entering the **`DISPLAY CLUSQMGR(*)`** command to list the cluster queue manager details.

Use the IBM MQ sample programs, such as **`amqsput`** and **`amqsget`**, to test the cluster.



IBM Training

Cluster tests in IBM MQ Explorer

- Verifies two full repositories in the cluster
- Checks whether any of manually defined cluster-sender channels are still in a *Retrying* state
- Checks that cluster can successfully resolve all queue manager names
- Checks whether any cluster memberships list the same queue manager more than one time
- Verifies that all instances of a cluster queue have the same attributes
- Checks whether any of the manually defined cluster-sender channels are still in a *Switching* state
- Confirms that cluster channels have a cluster value
- Checks the cluster name attributes of queues, channels, and queue managers

In IBM MQ Explorer, right-click the cluster name and then click **Tests > Run Default Tests**

MQ Explorer - Test Results

0 errors, 0 warnings, 9 infos

	Description	Object name
i	Test completed: 'Verify two full repositories'	CLUS1
i	Test completed: 'Cluster-sender channels in retrying state'	CLUS1
i	Test completed: 'Cluster fails to resolve queue manager name'	CLUS1
i	Test completed: 'Duplicate cluster members'	CLUS1
i	Test completed: 'Verify cluster queue instances'	CLUS1
i	Test completed: 'Cluster-sender channels in switching state'	CLUS1
i	Test completed: 'Confirm cluster attributes are set'	CLUS1
i	Test completed: 'Verify cluster namelist definitions '	CLUS1
i	Test completed: 'Verify cluster names'	CLUS1

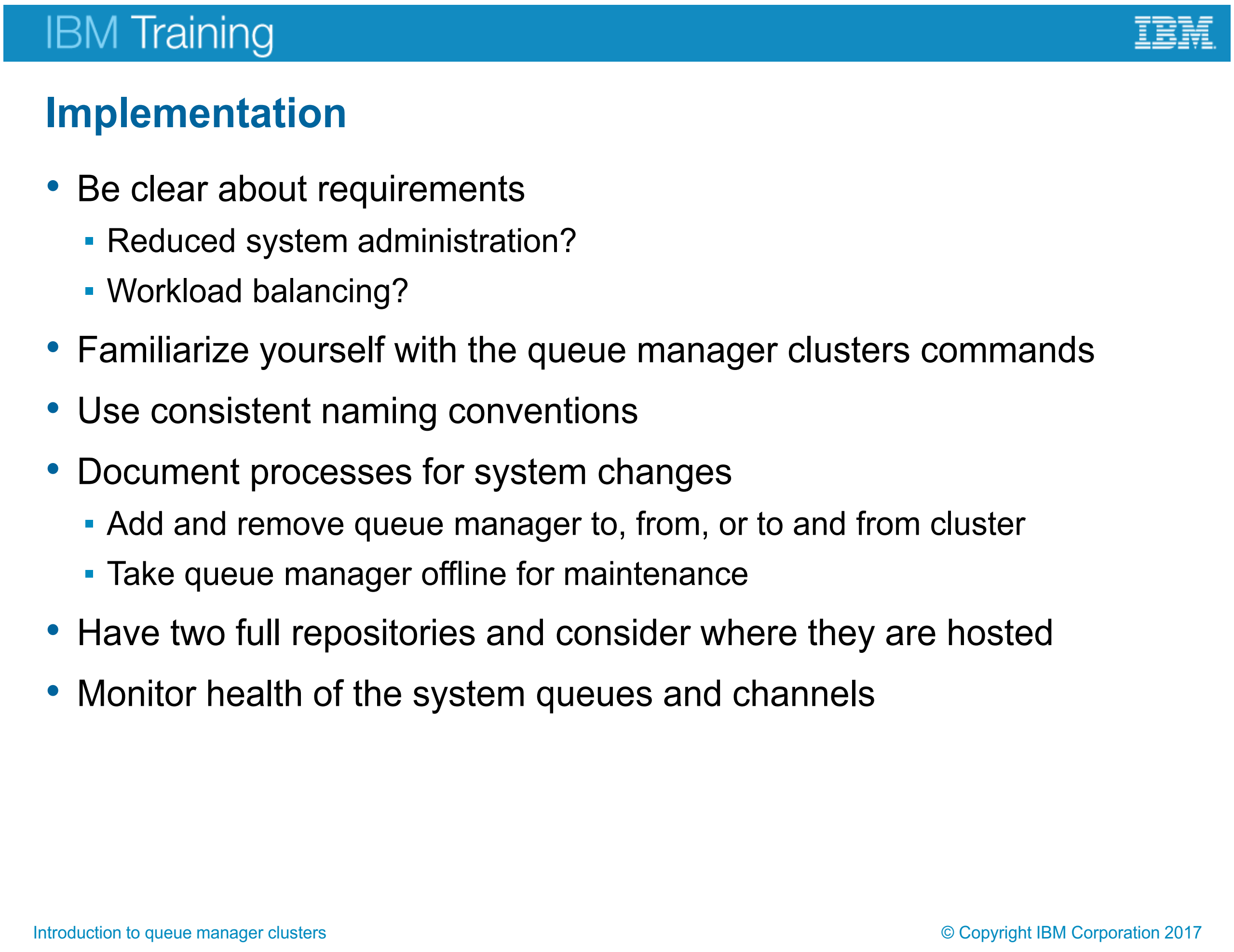
Introduction to queue manager clusters

© Copyright IBM Corporation 2017

Figure 12-27. Cluster tests in IBM MQ Explorer

IBM MQ Explorer includes built-in tests to verify some of the cluster properties and configuration.

- **Verify two full repositories** checks that the cluster contains at least two full repository queue managers.
- **Cluster-sender channels in retrying state** checks whether any of manually defined cluster-sender channels are still in a *Retrying* state.
- **Cluster fails to resolve queue manager name** checks that the cluster can successfully resolve all queue manager names.
- **Duplicate cluster members** checks whether any cluster memberships list the same queue manager more than one time.
- **Verify cluster queue instances** verifies that all instances of a cluster queue have the same attributes.
- **Cluster-sender channels in switching state** checks whether any of the manually defined cluster-sender channels are still in a *Switching* state.
- **Confirm that cluster attributes are set** checks that all cluster channels have a cluster value.
- **Verify cluster names** checks the cluster name attributes of queues, channels, and queue managers, except for capitalization.



IBM Training

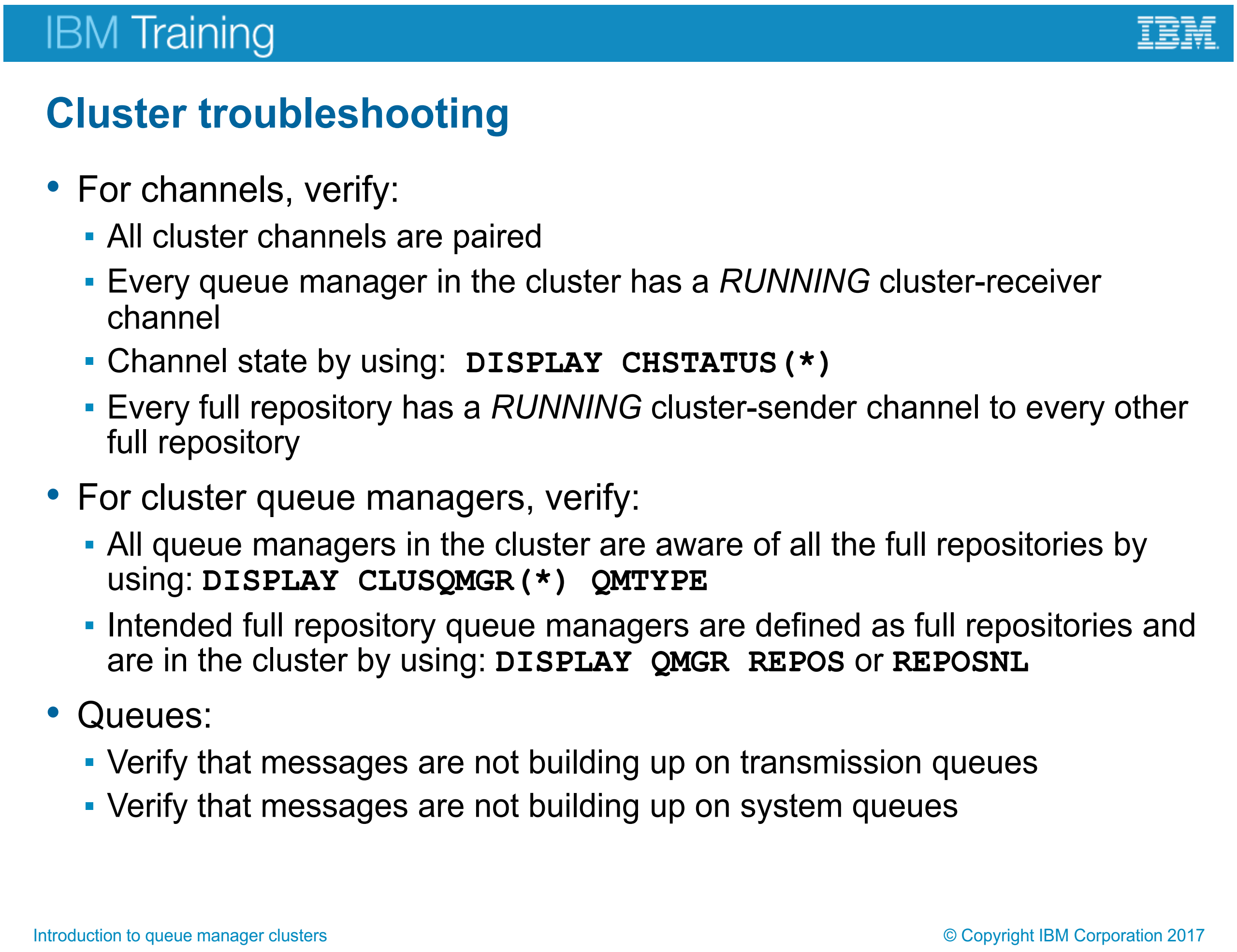
Implementation

- Be clear about requirements
  - Reduced system administration?
  - Workload balancing?
- Familiarize yourself with the queue manager clusters commands
- Use consistent naming conventions
- Document processes for system changes
  - Add and remove queue manager to, from, or to and from cluster
  - Take queue manager offline for maintenance
- Have two full repositories and consider where they are hosted
- Monitor health of the system queues and channels

Introduction to queue manager clusters

Figure 12-28. Implementation

The figure lists some guidelines to use when you implement clusters.



IBM Training

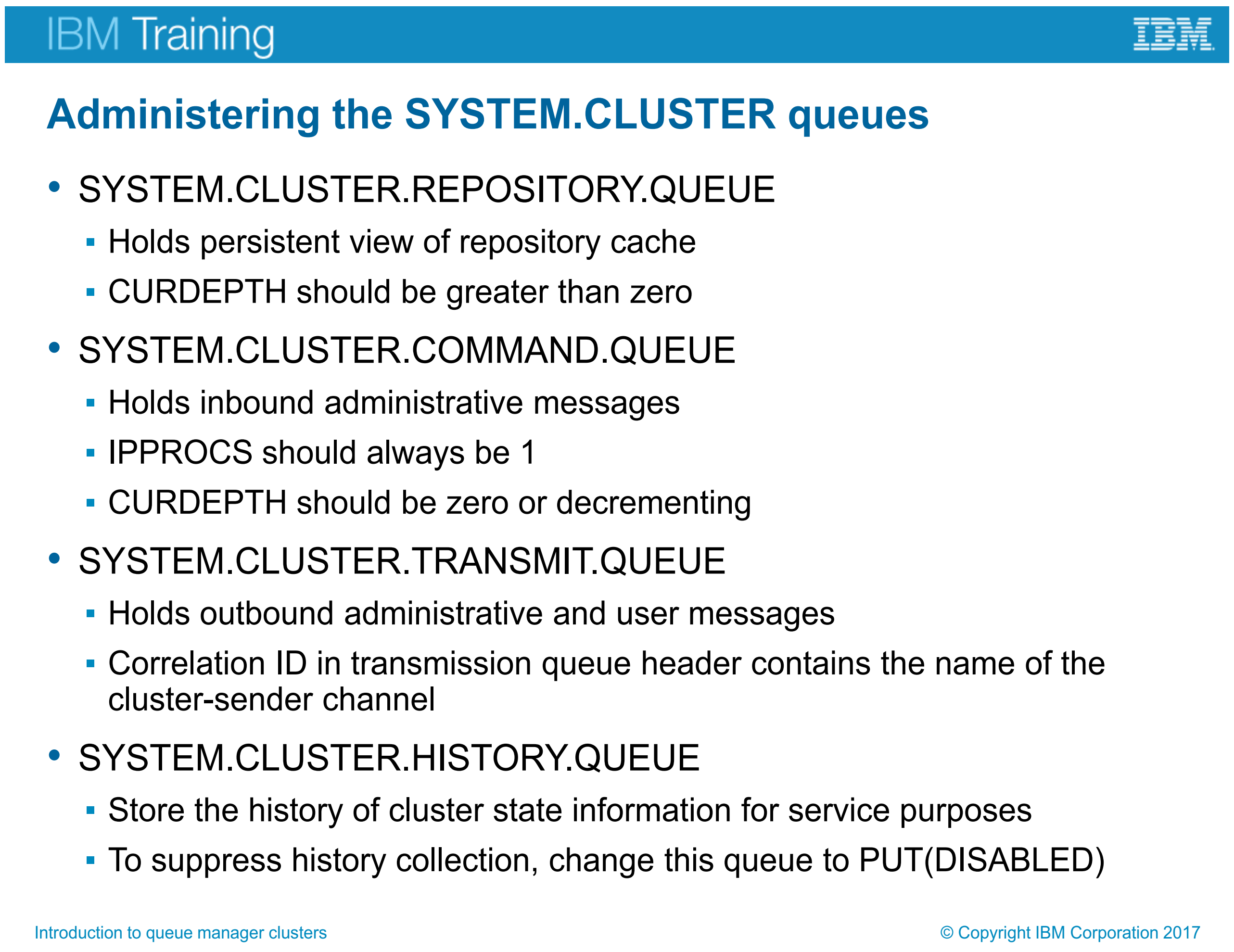
Cluster troubleshooting

- For channels, verify:
  - All cluster channels are paired
  - Every queue manager in the cluster has a *RUNNING* cluster-receiver channel
  - Channel state by using: **`DISPLAY CHSTATUS(*)`**
  - Every full repository has a *RUNNING* cluster-sender channel to every other full repository
- For cluster queue managers, verify:
  - All queue managers in the cluster are aware of all the full repositories by using: **`DISPLAY CLUSQMGR(*) QMTYPE`**
  - Intended full repository queue managers are defined as full repositories and are in the cluster by using: **`DISPLAY QMGR REPOS`** or **`REPOSNL`**
- Queues:
  - Verify that messages are not building up on transmission queues
  - Verify that messages are not building up on system queues

Introduction to queue manager clusters © Copyright IBM Corporation 2017

Figure 12-29. Cluster troubleshooting

Use this list and the troubleshooting advice in the IBM Knowledge Center to help you to detect and resolve problems when you use queue manager clusters.



IBM Training

Administering the SYSTEM.CLUSTER queues

- SYSTEM.CLUSTER.REPOSITORY.QUEUE
  - Holds persistent view of repository cache
  - CURDEPTH should be greater than zero
- SYSTEM.CLUSTER.COMMAND.QUEUE
  - Holds inbound administrative messages
  - IPPROCS should always be 1
  - CURDEPTH should be zero or decrementing
- SYSTEM.CLUSTER.TRANSMIT.QUEUE
  - Holds outbound administrative and user messages
  - Correlation ID in transmission queue header contains the name of the cluster-sender channel
- SYSTEM.CLUSTER.HISTORY.QUEUE
  - Store the history of cluster state information for service purposes
  - To suppress history collection, change this queue to PUT(DISABLED)

Introduction to queue manager clusters © Copyright IBM Corporation 2017

Figure 12-30. Administering the SYSTEM.CLUSTER queues

MQ contains special SYSTEM queues for clustering. These queues are created automatically when a queue manager is created.

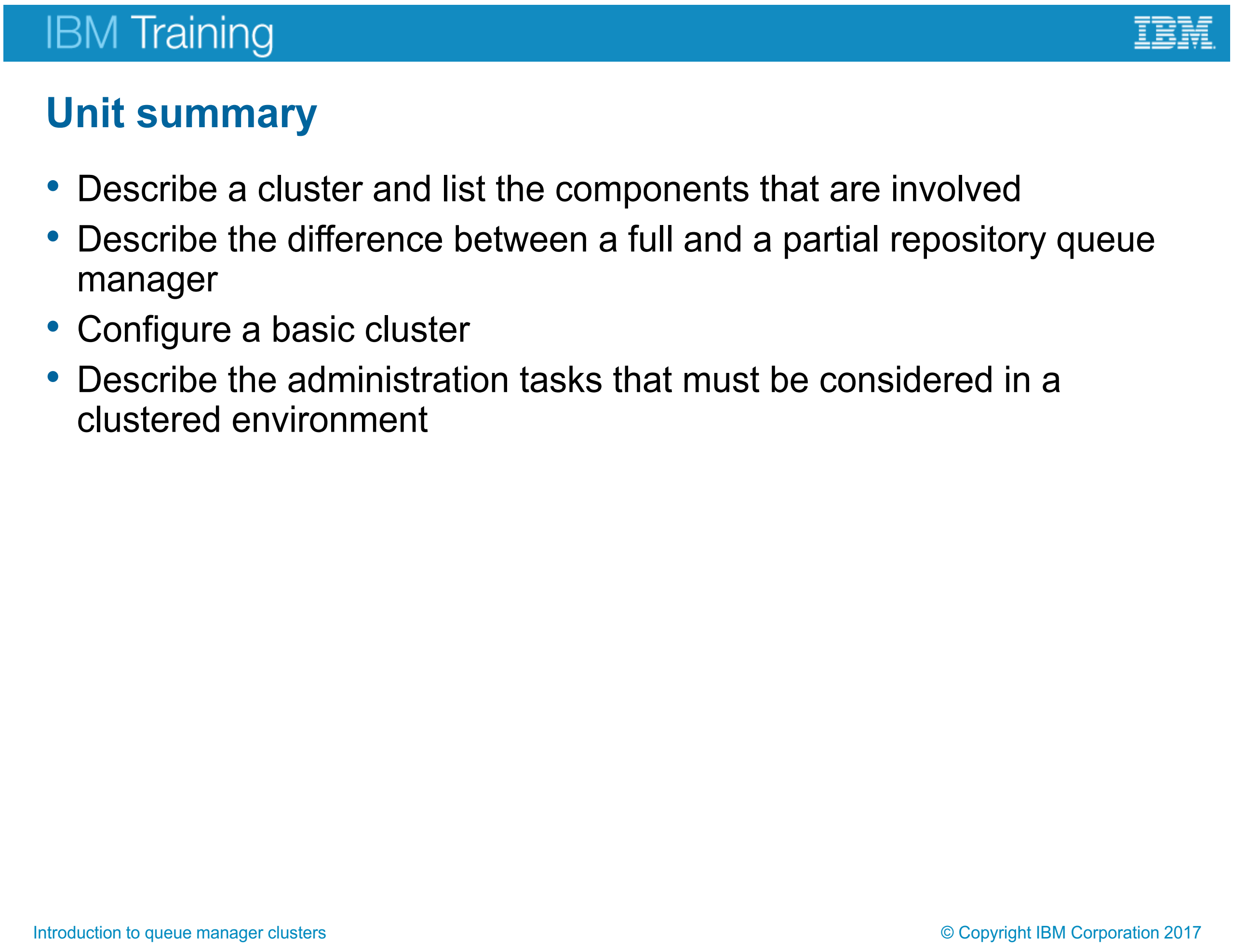
The SYSTEM.CLUSTER.REPOSITORY.QUEUE holds a persistent copy of the cluster repository.

The SYSTEM.CLUSTER.COMMAND.QUEUE is used to communicate repository changes between queue managers. Messages that are written to this queue contain updates to the repository data that applies to the local copy of the repository, or requests for repository data.

The SYSTEM.CLUSTER.TRANSMIT.QUEUE is the transmission queue for all destinations in the cluster.

The SYSTEM.CLUSTER.HISTORY.QUEUE stores the history of cluster state information for service purposes. In the default object settings, SYSTEM.CLUSTER.HISTORY.QUEUE is set to PUT(ENABLED). To suppress history collection, change the setting to PUT(DISABLED).

All of these queues typically contain many messages.



IBM Training

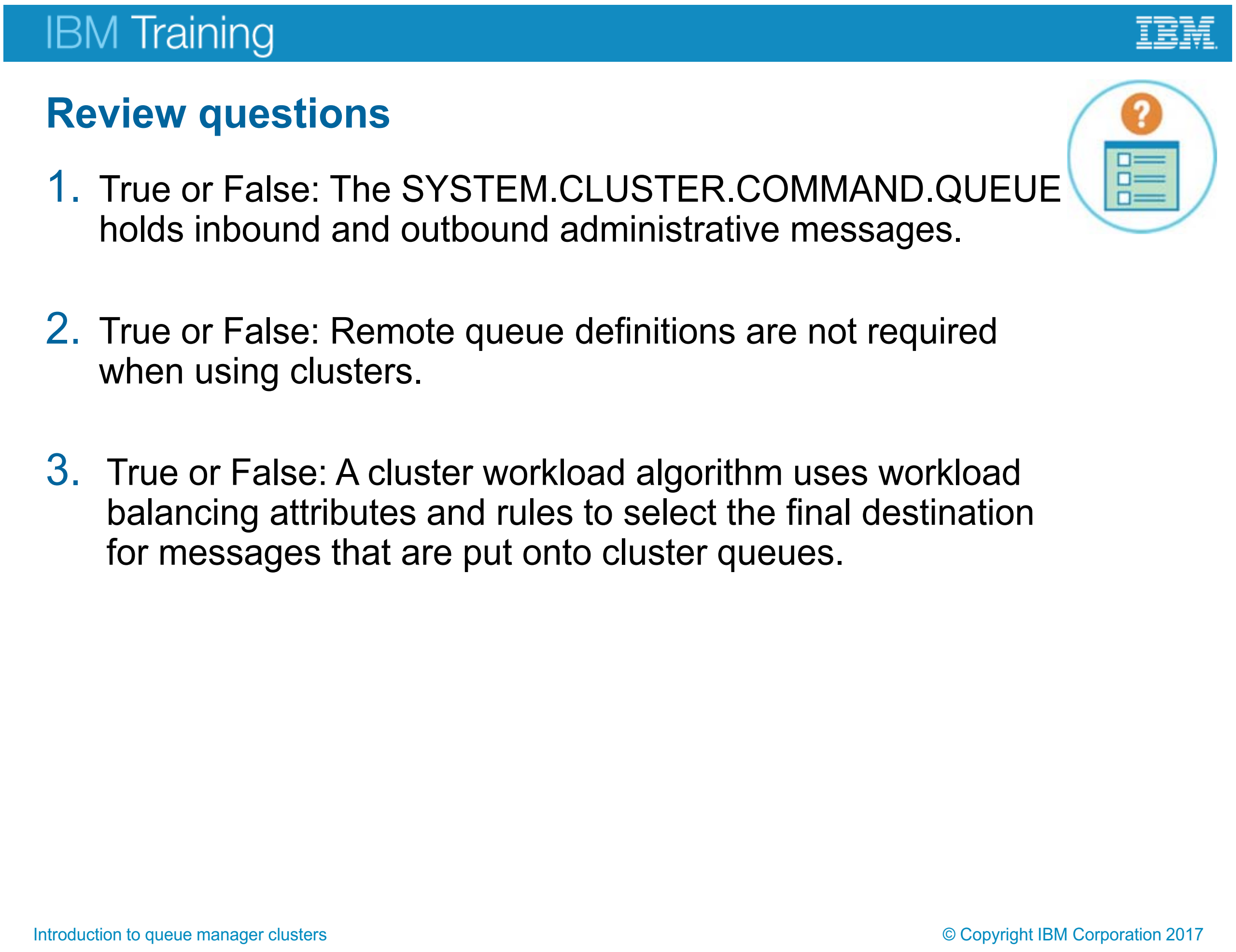
Unit summary

- Describe a cluster and list the components that are involved
- Describe the difference between a full and a partial repository queue manager
- Configure a basic cluster
- Describe the administration tasks that must be considered in a clustered environment

Introduction to queue manager clusters

© Copyright IBM Corporation 2017

Figure 12-31. Unit summary



IBM Training

Review questions

1. True or False: The SYSTEM.CLUSTER.COMMAND.QUEUE holds inbound and outbound administrative messages.

2. True or False: Remote queue definitions are not required when using clusters.

3. True or False: A cluster workload algorithm uses workload balancing attributes and rules to select the final destination for messages that are put onto cluster queues.

Introduction to queue manager clusters

© Copyright IBM Corporation 2017

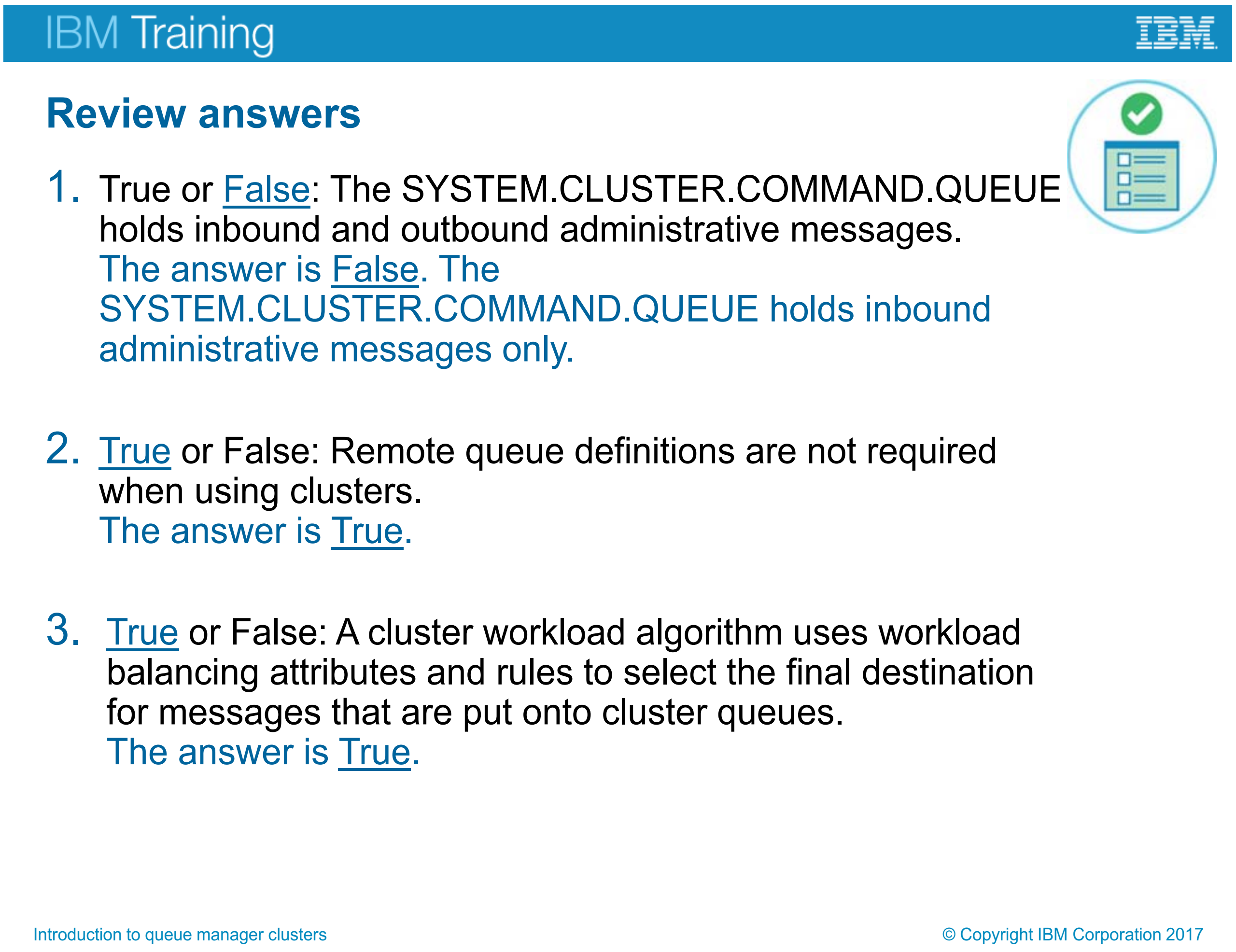
Figure 12-32. Review questions

Write your answers down here:

1.

2.

3.



IBM Training

Review answers

1. True or False: The SYSTEM.CLUSTER.COMMAND.QUEUE holds inbound and outbound administrative messages.
   The answer is False. The SYSTEM.CLUSTER.COMMAND.QUEUE holds inbound administrative messages only.

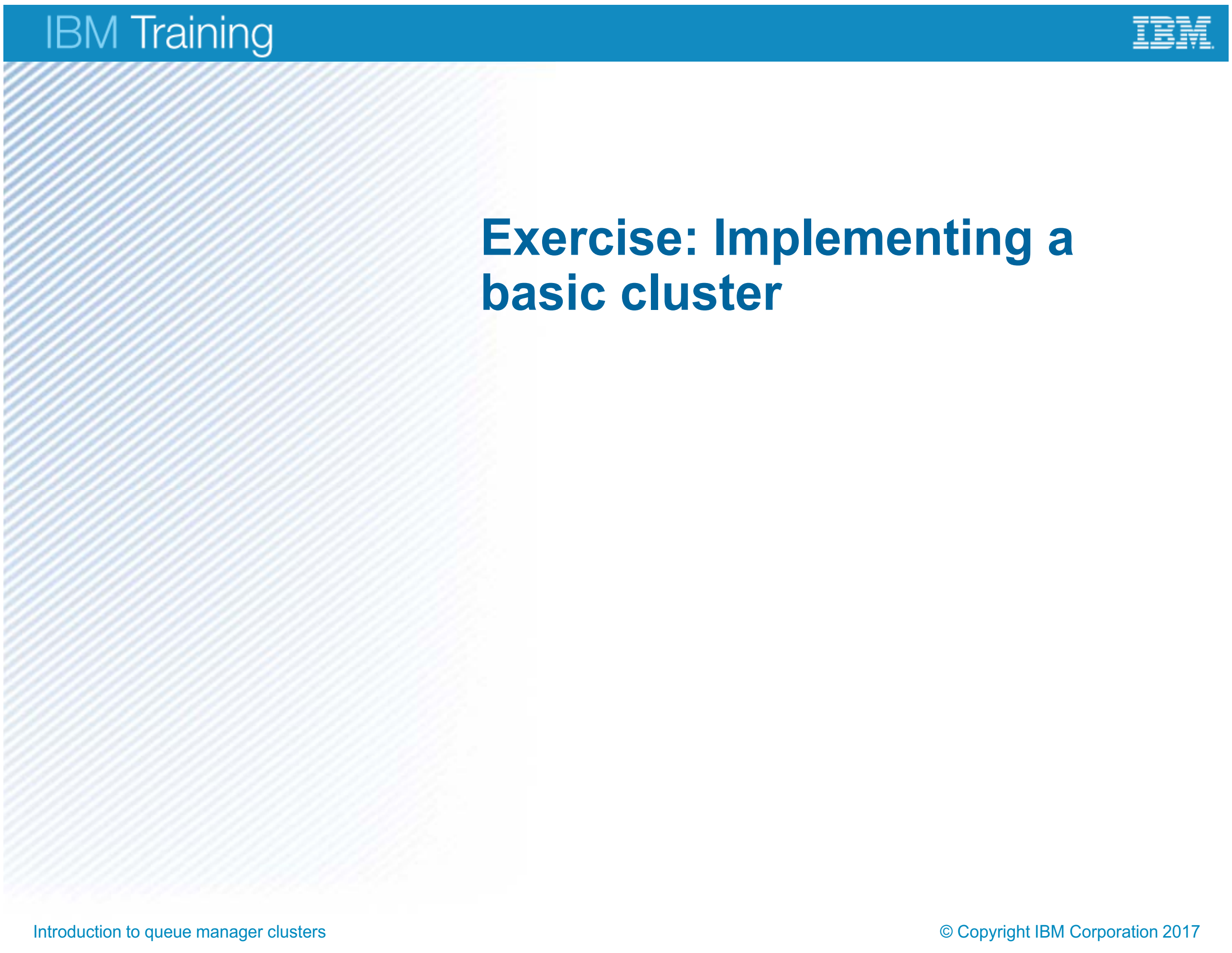
2. True or False: Remote queue definitions are not required when using clusters.
   The answer is True.

3. True or False: A cluster workload algorithm uses workload balancing attributes and rules to select the final destination for messages that are put onto cluster queues.
   The answer is True.

Introduction to queue manager clusters

© Copyright IBM Corporation 2017

Figure 12-33. Review answers


IBM Training

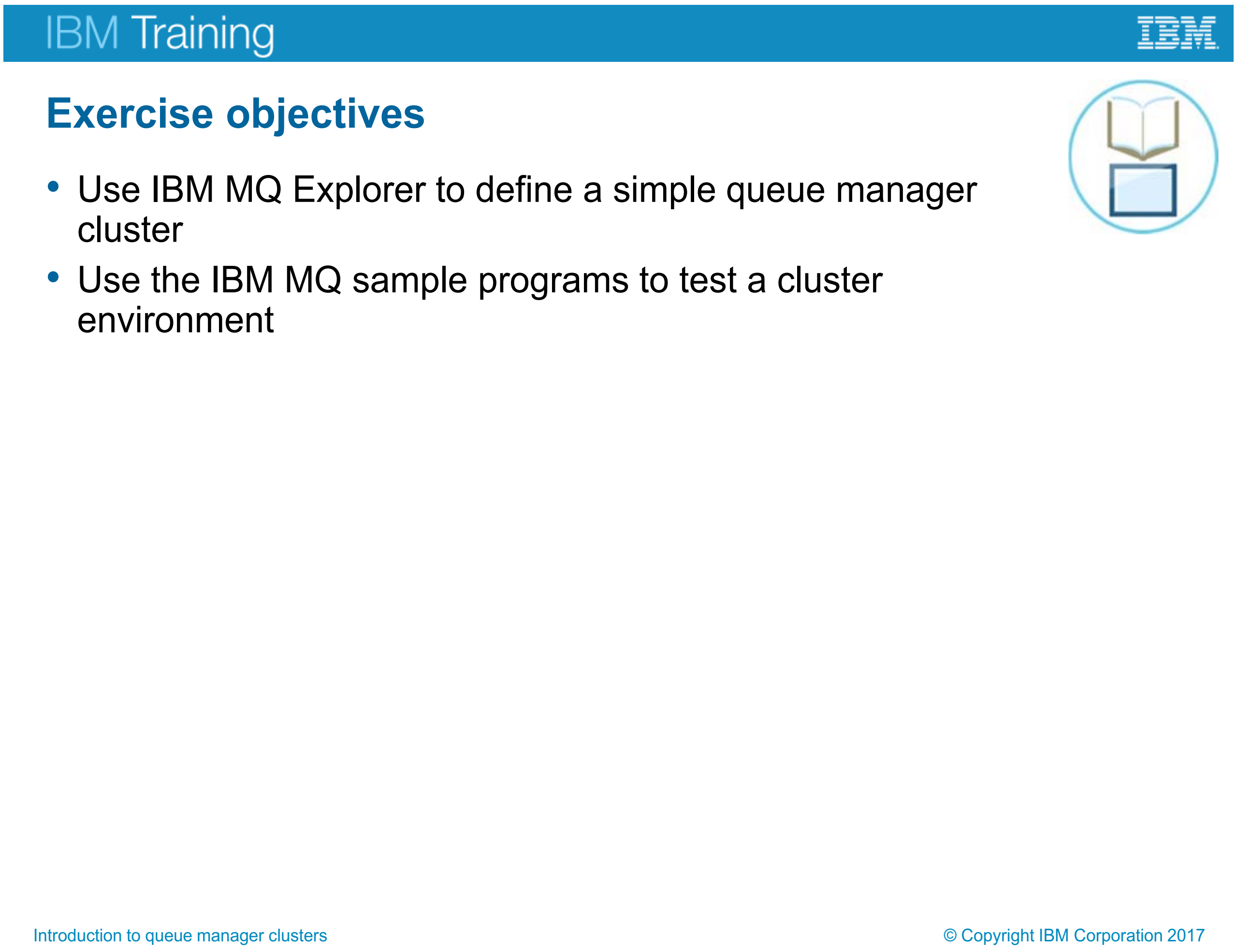
Exercise: Implementing a basic cluster

Introduction to queue manager clusters

© Copyright IBM Corporation 2017

Figure 12-34. Exercise: Implementing a basic cluster

In this exercise, you use IBM MQ Explorer to create a cluster of four queue managers. You then test the cluster by using the cluster mechanism to send messages between queues on all queue managers in the cluster.



IBM Training

Exercise objectives

- Use IBM MQ Explorer to define a simple queue manager cluster
- Use the IBM MQ sample programs to test a cluster environment

Introduction to queue manager clusters

© Copyright IBM Corporation 2017

Figure 12-35. Exercise objectives

See the *Course Exercises Guide* for detailed instructions.