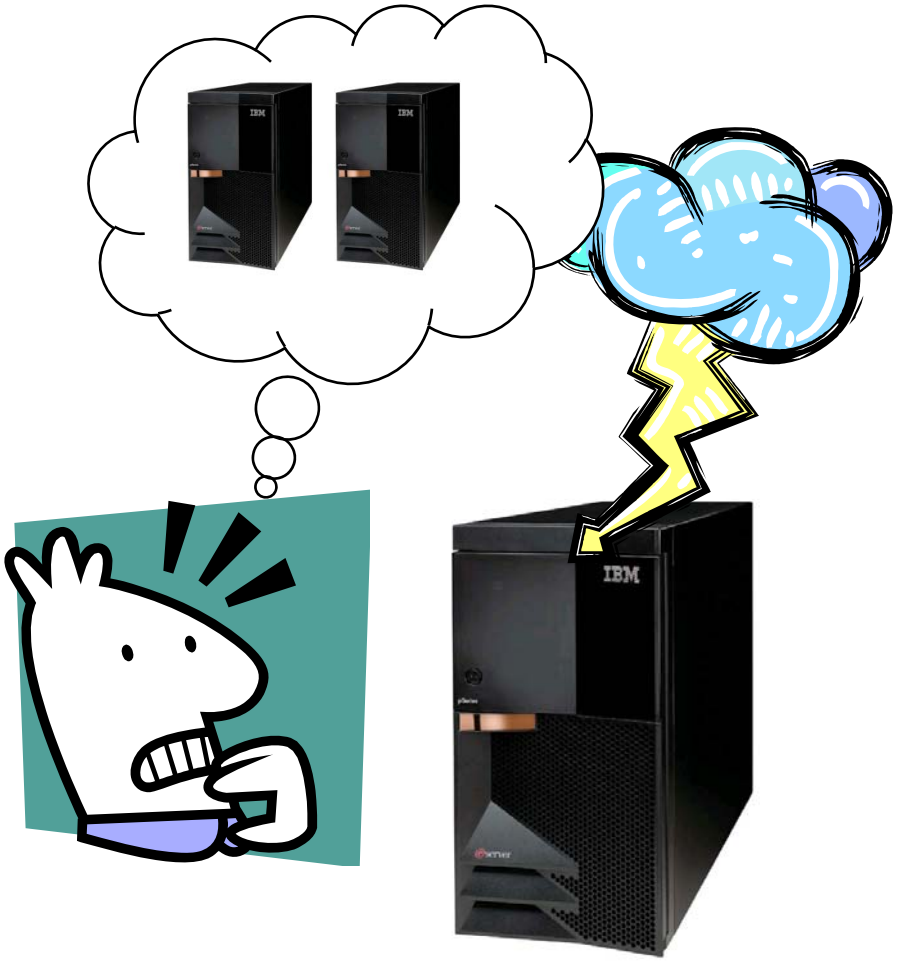
WMQ High Availability

T.Rob Wyatt

t.rob@ioptconsulting.com

Overview

- **Techniques and technologies to ensure availability of messaging**
- **WebSphere MQ technologies**
  - Queue Manager Clusters
  - Multi-instance Queue Managers
  - Shared Queues
- **Platform technologies**
  - Failover with HA clusters



Introduction

- **Availability is a very large subject**
  - We won't be covering everything
- **Not just HA technology - anything that can cause an outage is significant**
  - This might be an overloaded system, etc
  - We will only be covering HA technology
- **You can have the best HA technology in the world, but you have to manage it correctly**
- **HA technology is not a substitute for good planning and testing!**

What are you trying to achieve?

- **The objective is to achieve 24x7 availability of messaging**
- **Not always achievable, but we can get close**
  - 99.9% availability = 8.76 hours downtime/year
  - 99.999% = 5 minutes
  - 99.9999% = 30 seconds
- **Potential outage types:**
  - 80% scheduled downtime (new software release, upgrades, maintenance)
  - 20% unscheduled downtime (source: Gartner Group)
    - 40% operator error
    - 40% application error
    - 20% other (network failures, disk crashes, power outage etc.)
- **Avoid application awareness of availability solutions**

Single Points of Failure

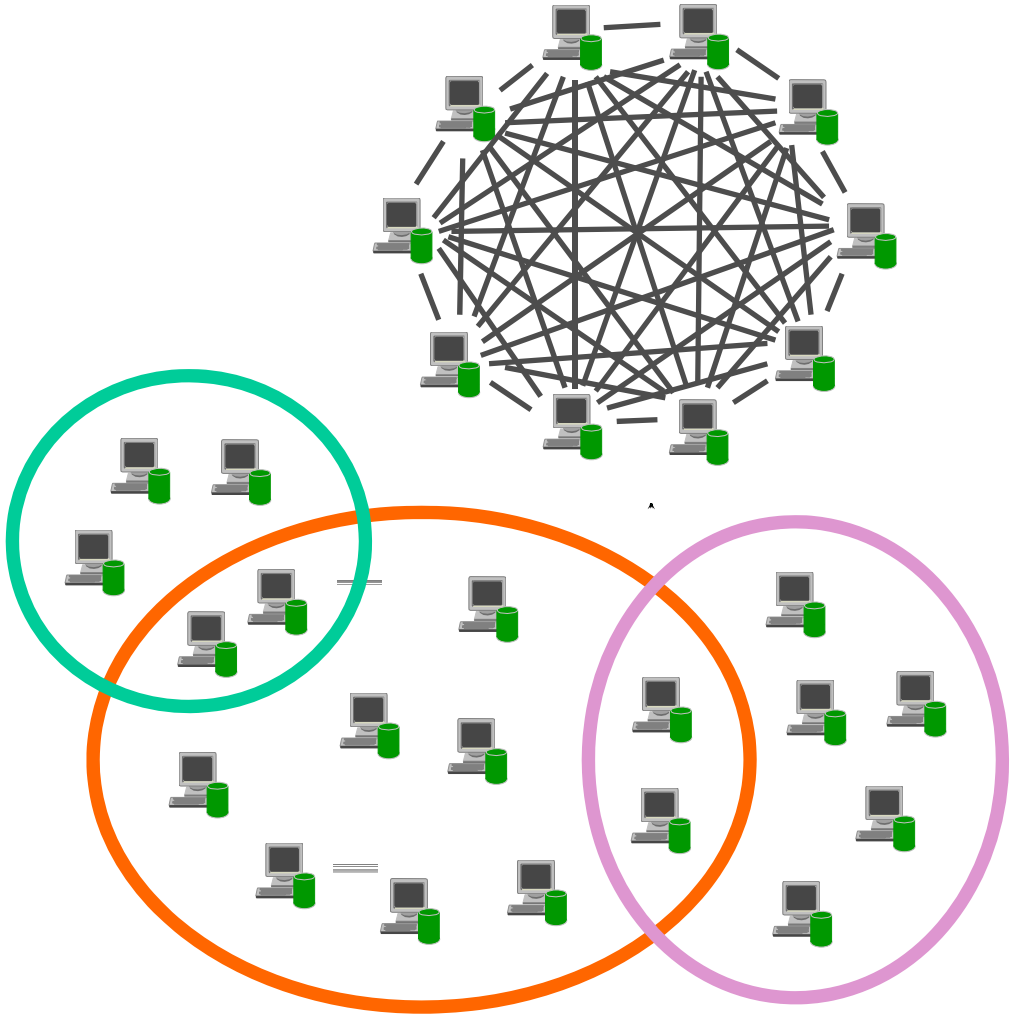
- **With no redundancy or fault tolerance, a failure of any component can lead to a loss of availability**
- **Every component is critical. The system relies on the:**
  - Power supply, system unit, CPU, memory
  - Disk controller, disks, network adapter, network cable
  - ...and so on
- **Various techniques have been developed to tolerate failures:**
  - UPS or dual supplies for power loss
  - RAID for disk failure
  - Fault-tolerant architectures for CPU/memory failure
  - ...etc
- **Elimination of SPOFs is important to achieve HA**

WebSphere MQ HA technologies

- Queue manager clusters
- Queue-sharing groups
- Support for networked storage
- Multi-instance queue managers
- HA clusters
- Client reconnection

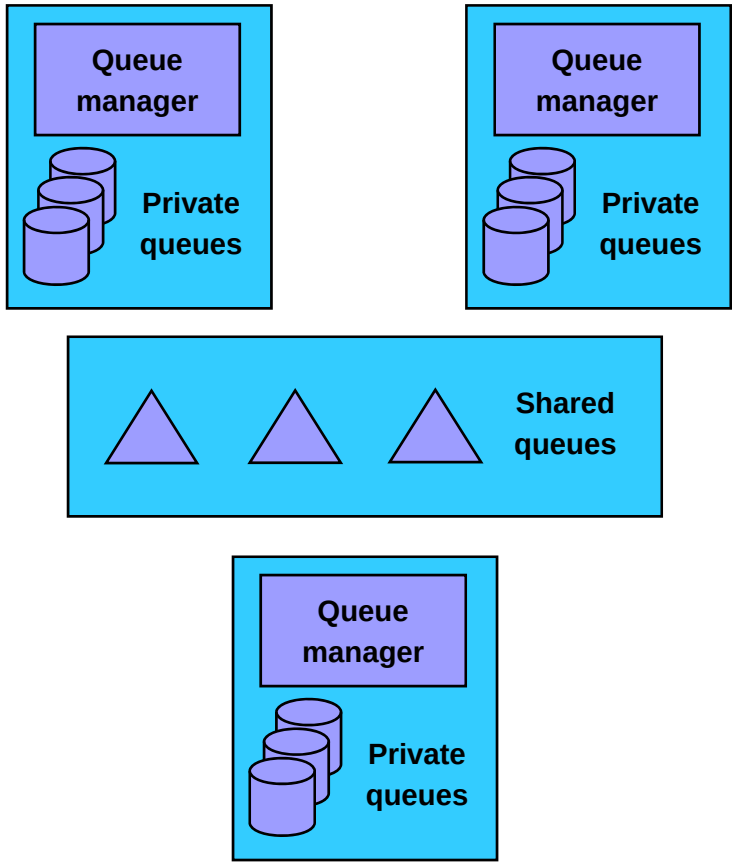
Queue Manager Clusters

- Sharing cluster queues on multiple queue managers prevents a queue from being a SPOF
- Cluster workload algorithm automatically routes traffic away from failed queue managers



Queue-Sharing Groups

- **On z/OS, queue managers can be members of a queue-sharing group**
- **Shared queues are held in a coupling facility**
  - All queue managers in the QSG can access the messages
- **Benefits:**
  - Messages remain available even if a queue manager fails
  - Pull workload balancing
  - Apps can connect to the group



Support for networked storage

- **Support has been added for queue manager data in networked storage**
  - NAS so that data is available to multiple machines concurrently
    - Already have SAN support
  - Added protection against concurrent starting two instances of a queue manager using the same queue manager data
  - On Windows, support for Windows network drives (SMB)
  - On Unix variants, support for POSIX-compliant filesystems with leased file locking
    - NFS v4 has been tested by IBM
- **Some customers have a “no local disk” policy for queue manager data**
  - This is an enabler for some virtualized deployments
  - Allows simple switching of queue manager to another server following a hardware failure

Introduction to Failover and MQ

- **Failover is the automatic switching of availability of a service**
  - For MQ, the “service” is a queue manager
- **Traditionally the preserve of an HA cluster, such as HACMP**
- **Requires:**
  - Data accessible on all servers
  - Equivalent or at least compatible servers
    - Common software levels and environment
  - Sufficient capacity to handle workload after failure
    - Workload may be rebalanced after failover requiring spare capacity
  - Startup processing of queue manager following the failure
- **MQ offers two ways of configuring for failover:**
  - Multi-instance queue managers
  - HA clusters

Failover considerations

- **Failover times are made up of three parts:**
  - Time taken to notice the failure
    - Heartbeat missed
    - Bad result from status query
  - Time taken to establish the environment before activating the service
    - Switching IP addresses and disks, and so on
  - Time taken to activate the service
    - This is queue manager restart
- **Failover involves a queue manager restart**
  - Non-persistent messages, nondurable subscriptions discarded
- **For fastest times, ensure that queue manager restart is fast**
  - No long running transactions, for example
  - Shallow queues

MULTI-INSTANCE QUEUE MANAGERS

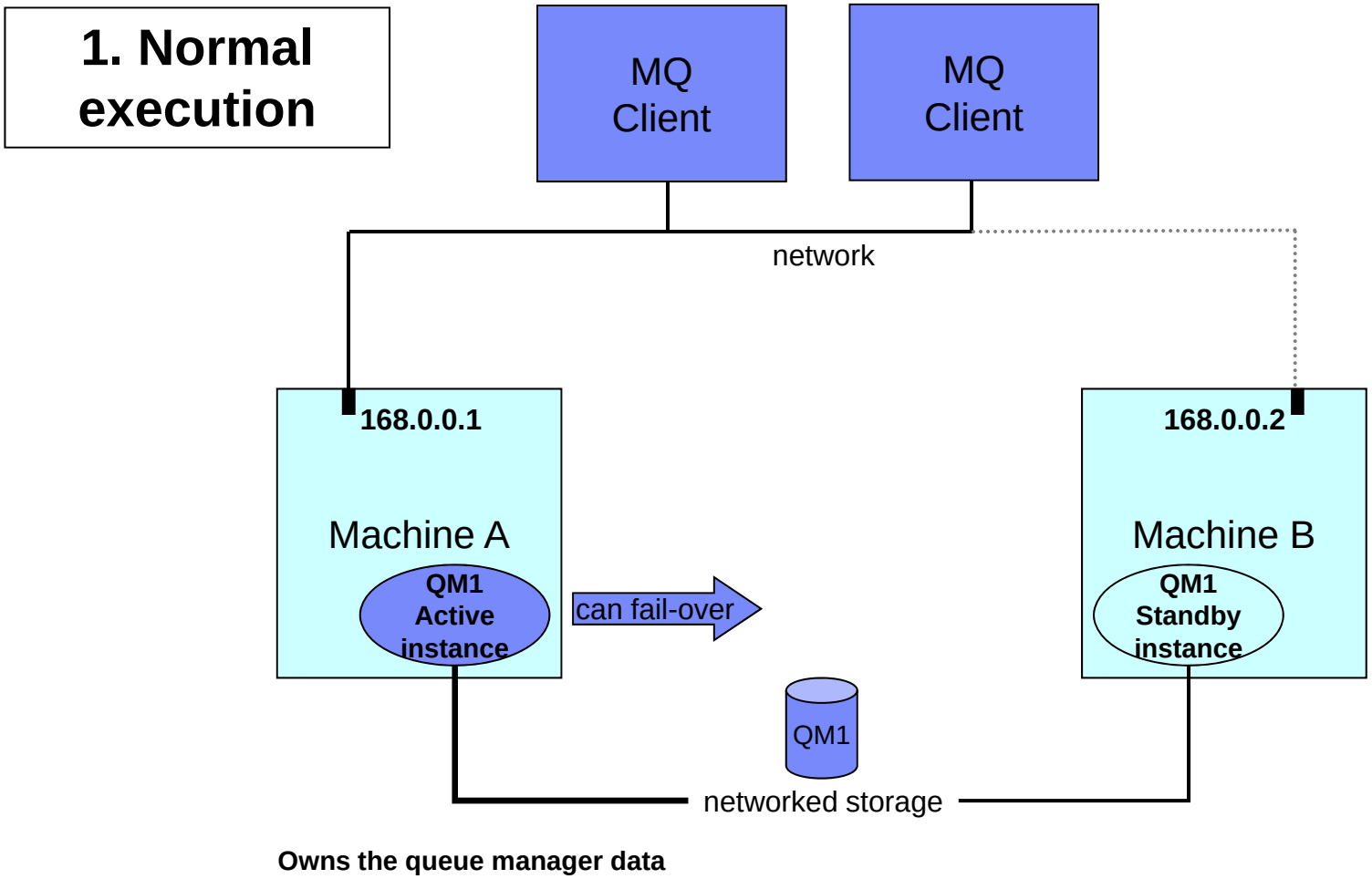
Multi-instance Queue Managers

- **Basic failover support without HA cluster**
- **Two instances of queue manager on different machines**
  - One is the “active” instance, other is the “standby” instance
  - Active instance “owns” the queue manager’s files
    - Accepts connections from applications
  - Standby instance monitors the active instance
    - Applications cannot connect to the standby instance
    - If active instance fails, standby restarts queue manager and becomes active
- **Instances are the SAME queue manager - only one set of data files**
  - Queue manager data is held in networked storage

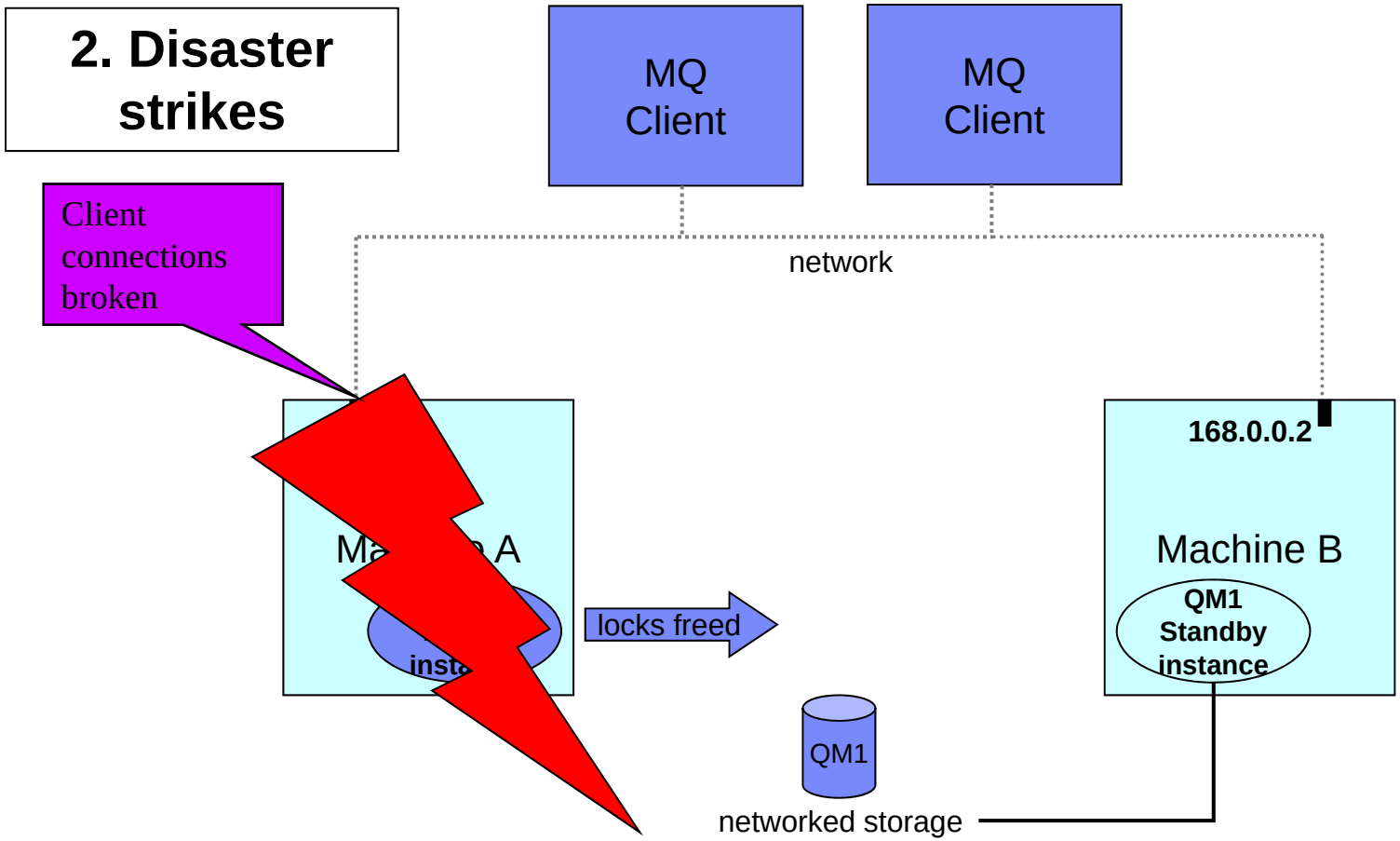
Setting up Multi-instance Queue Manager

- **Set up shared filesystems for QM data and logs**
- **Create the queue manager on machine1**
  - crtmqm -md /shared/qmdata -ld /shared/qmlog QM1
- **Define the queue manager on machine2 (or edit mqs.ini)**
  - addmqinf -v Name=QM1 -v Directory=QM1 -v Prefix=/var/mqm \
    -v DataPath=/shared/qmdata/QM1
- **Start an instance on machine1 - it becomes active**
  - strmqm -x QM1
- **Start another instance on machine2 - it becomes standby**
  - strmqm -x QM1
- **That's it. If the queue manager instance on machine1 fails, the standby instance on machine2 takes over and becomes active**

Multi-instance Queue Managers



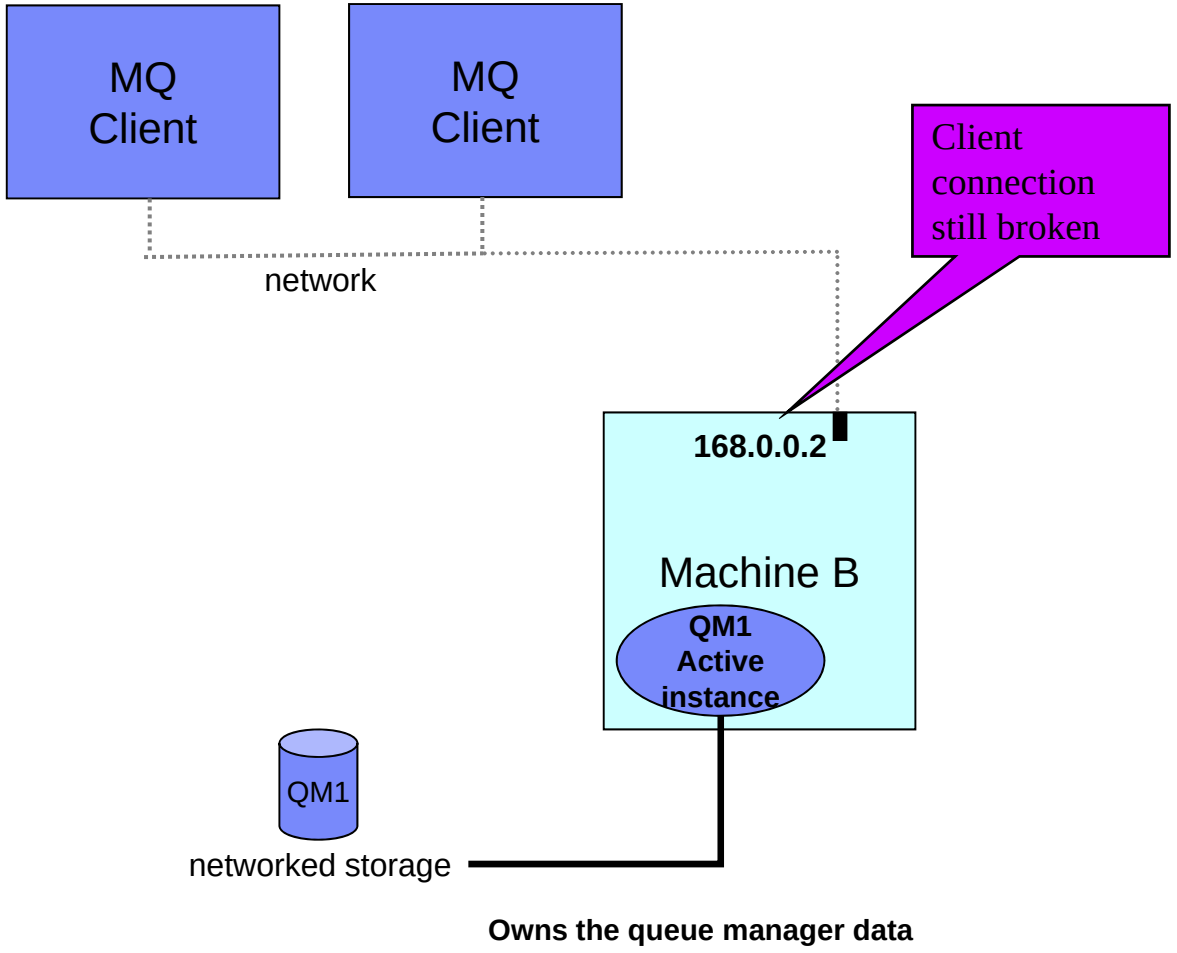
Multi-instance Queue Managers



Multi-instance Queue Managers

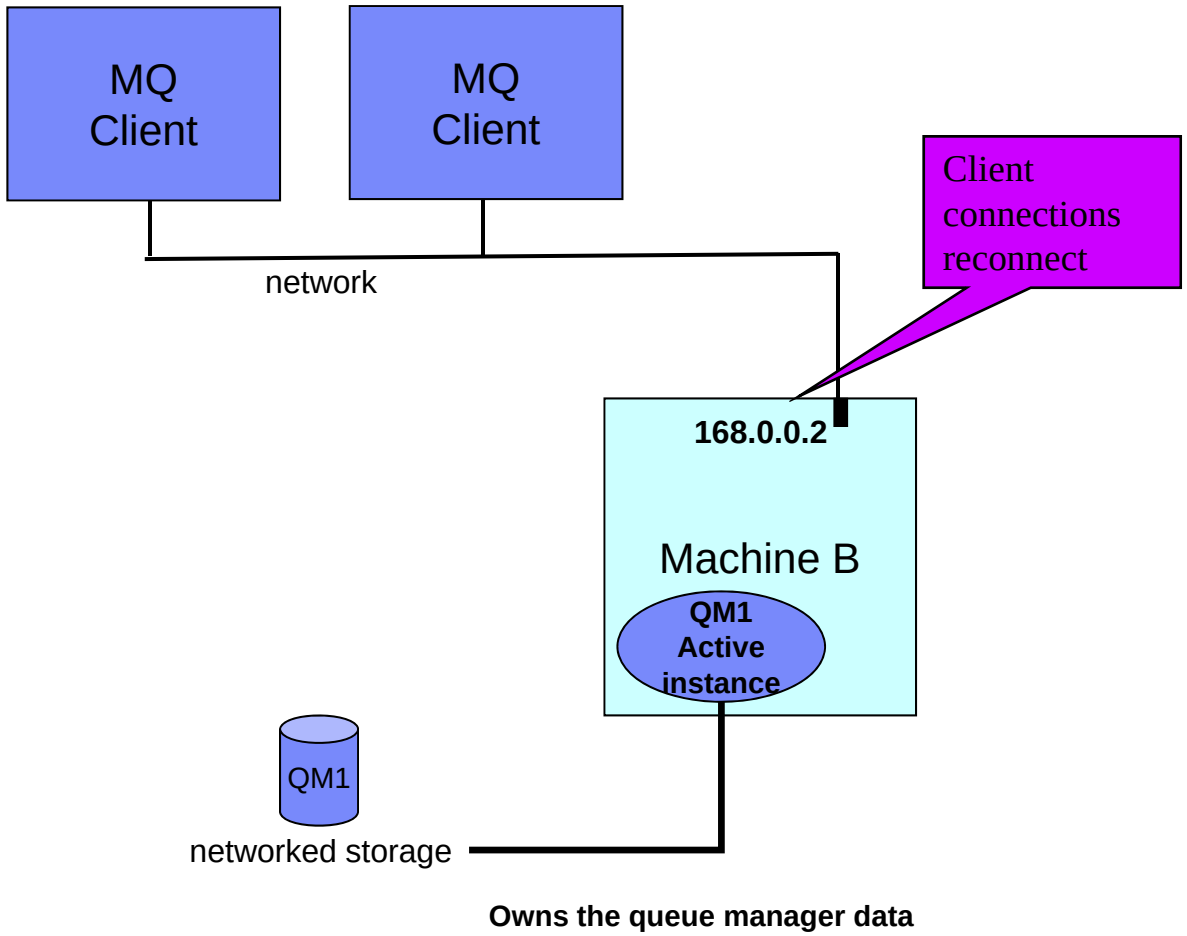
3. FAILOVER

Standby becomes active



Multi-instance Queue Managers

4. Recovery complete



Multi-instance Queue Managers

- **MQ is NOT becoming an HA cluster**
  - If other resources need to be coordinated, you need an HA cluster
  - WebSphere Message Broker will integrate with multi-instance QM
  - Queue manager services can be automatically started, but with limited control
- **The IP address of the queue manager changes when it moves**
  - MQ channel configuration needs list of addresses unless you use external IPAT or an intelligent router
  - Connection name syntax extended to a comma-separated list
    - CONNAME('168.0.0.1,168.0.0.2')
- **System administrator is responsible for restarting another standby instance when failover has occurred**

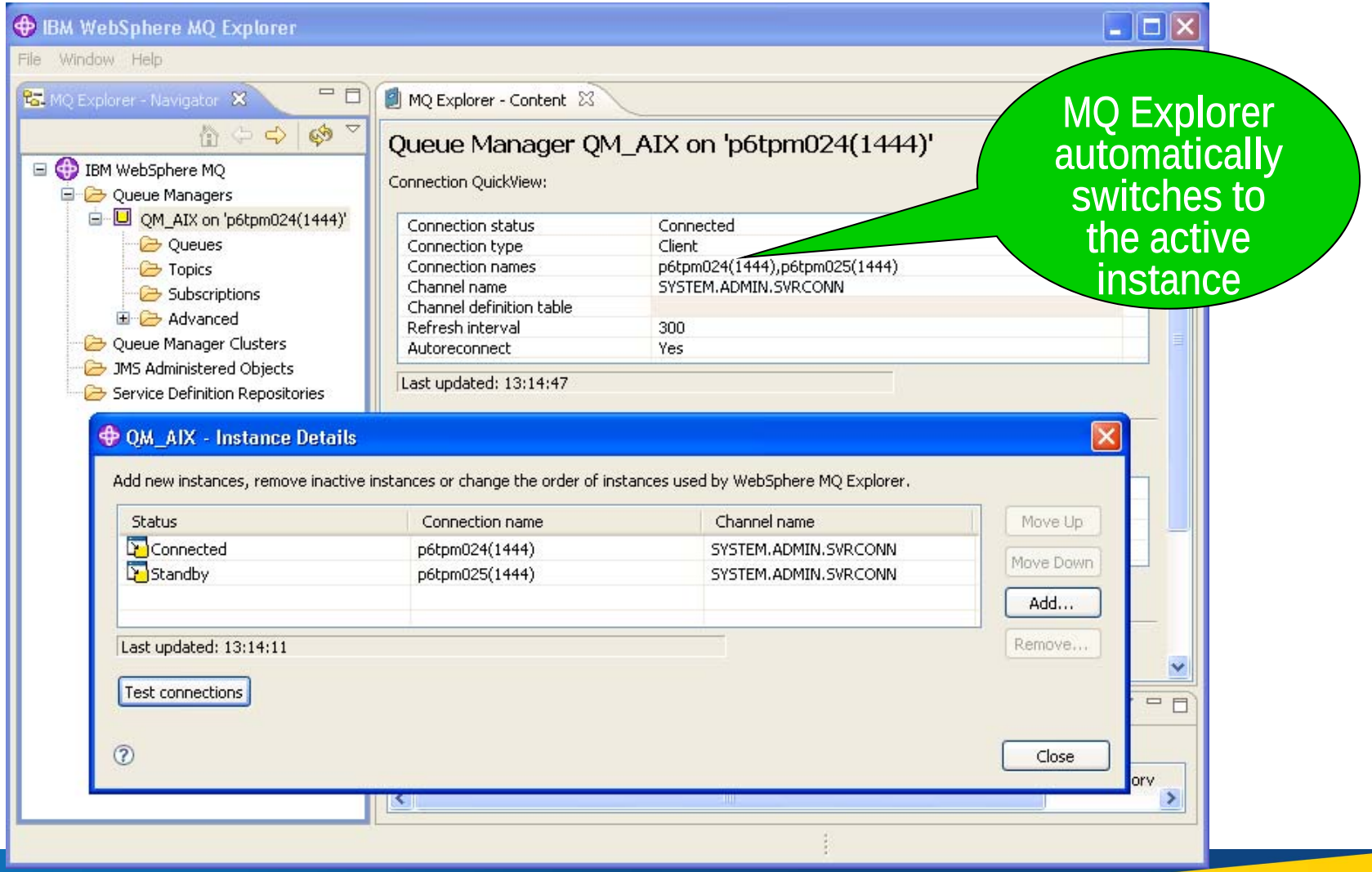
Administering Multi-instance QMgrs

- All queue manager administration must be performed on active instance
- dspmq enhanced to display instance information

```
$ hostname
staravia
$ dspmq -x
QMNAME(MIQM)      STATUS(Running as standby)
    INSTANCE(starly)     MODE(Active)
    INSTANCE(staravia)   MODE(Standby)
```

  - dspmq issued on “staravia”
  - On “staravia”, there’s a standby instance
  - The active instance is on “starly”

Multi-instance QMgr in MQ Explorer



HA CLUSTERS

HA clusters

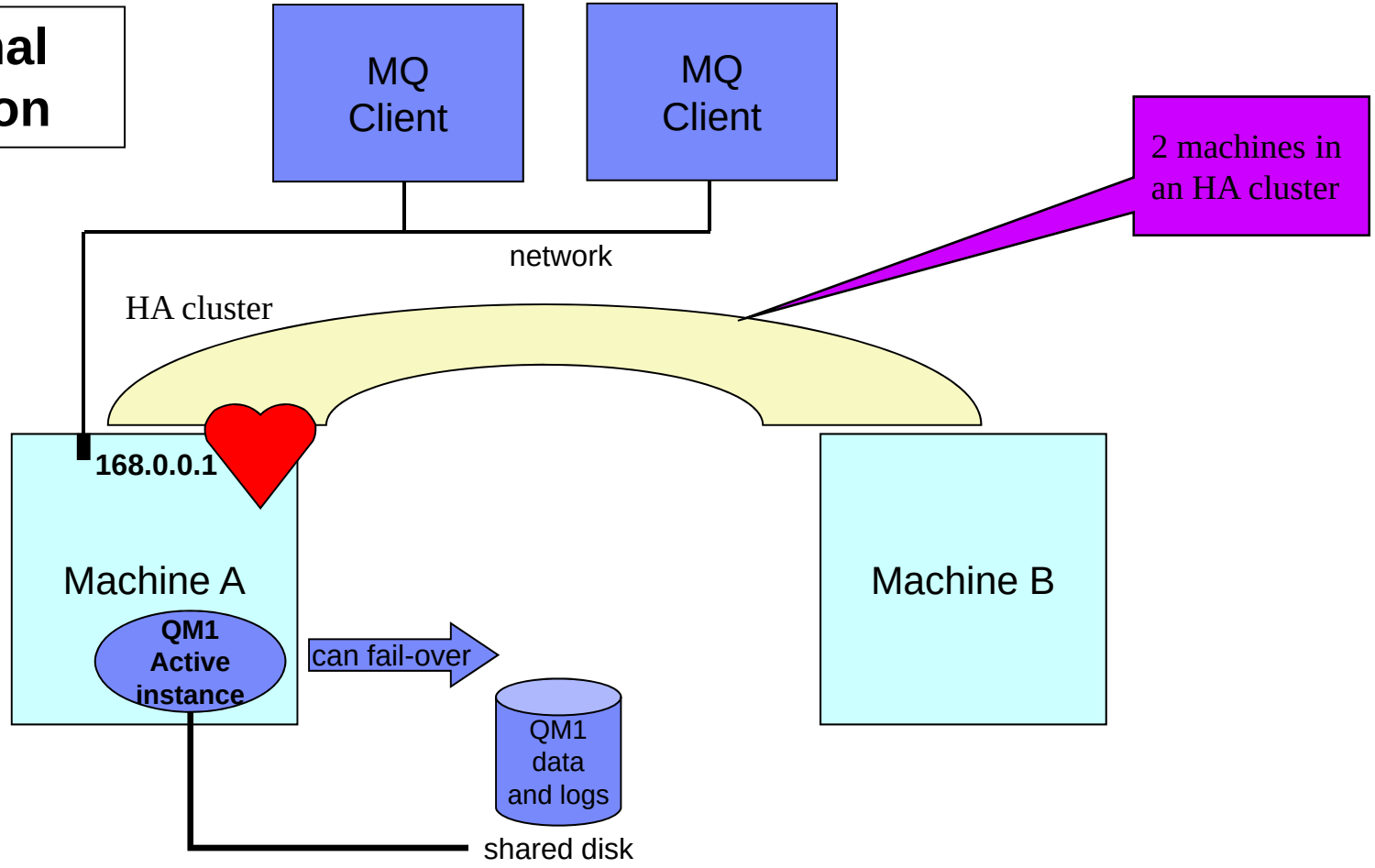
- **MQ traditionally made highly available using an HA cluster**
  - IBM PowerHA for AIX (formerly HACMP), Veritas Cluster Server, Microsoft Cluster Server, HP Serviceguard, …
- **HA clusters can:**
  - Coordinate multiple resources such as application server, database
  - Consist of more than two machines
  - Failover more than once without operator intervention
  - Takeover IP address as part of failover
  - Likely to be more resilient in cases of MQ and OS defects

HA clusters

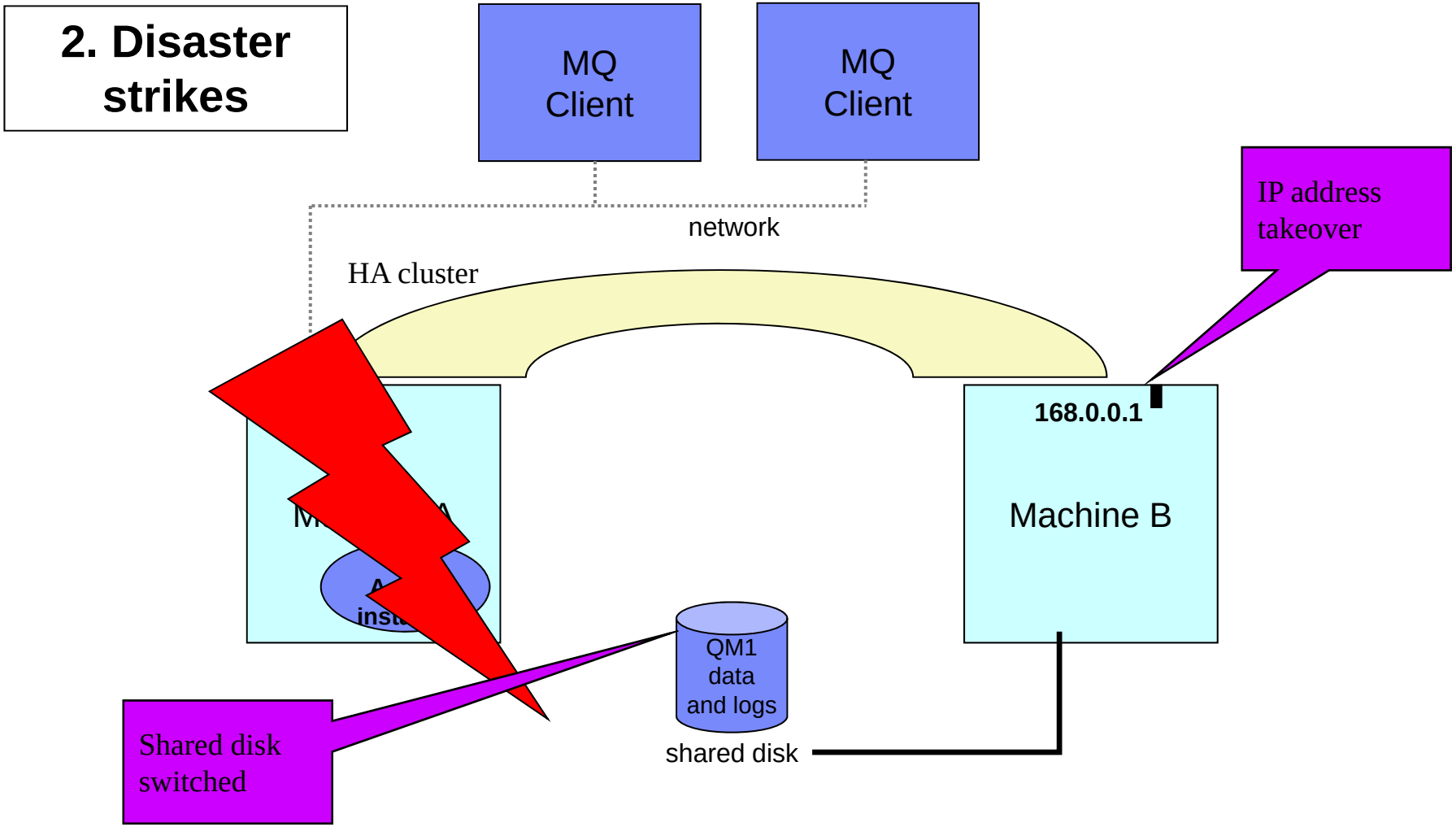
- **In HA clusters, queue manager data and logs are placed on a shared disk**
  - Disk is switched between machines during failover
- **The queue manager has its own “service” IP address**
  - IP address is switched between machines during failover
  - Queue manager’s IP address remains the same after failover
- **The queue manager is defined to the HA cluster as a resource dependent on the shared disk and the IP address**
  - During failover, the HA cluster will switch the disk, take over the IP address and then start the queue manager

MQ in an HA cluster - Cold standby

1. Normal execution

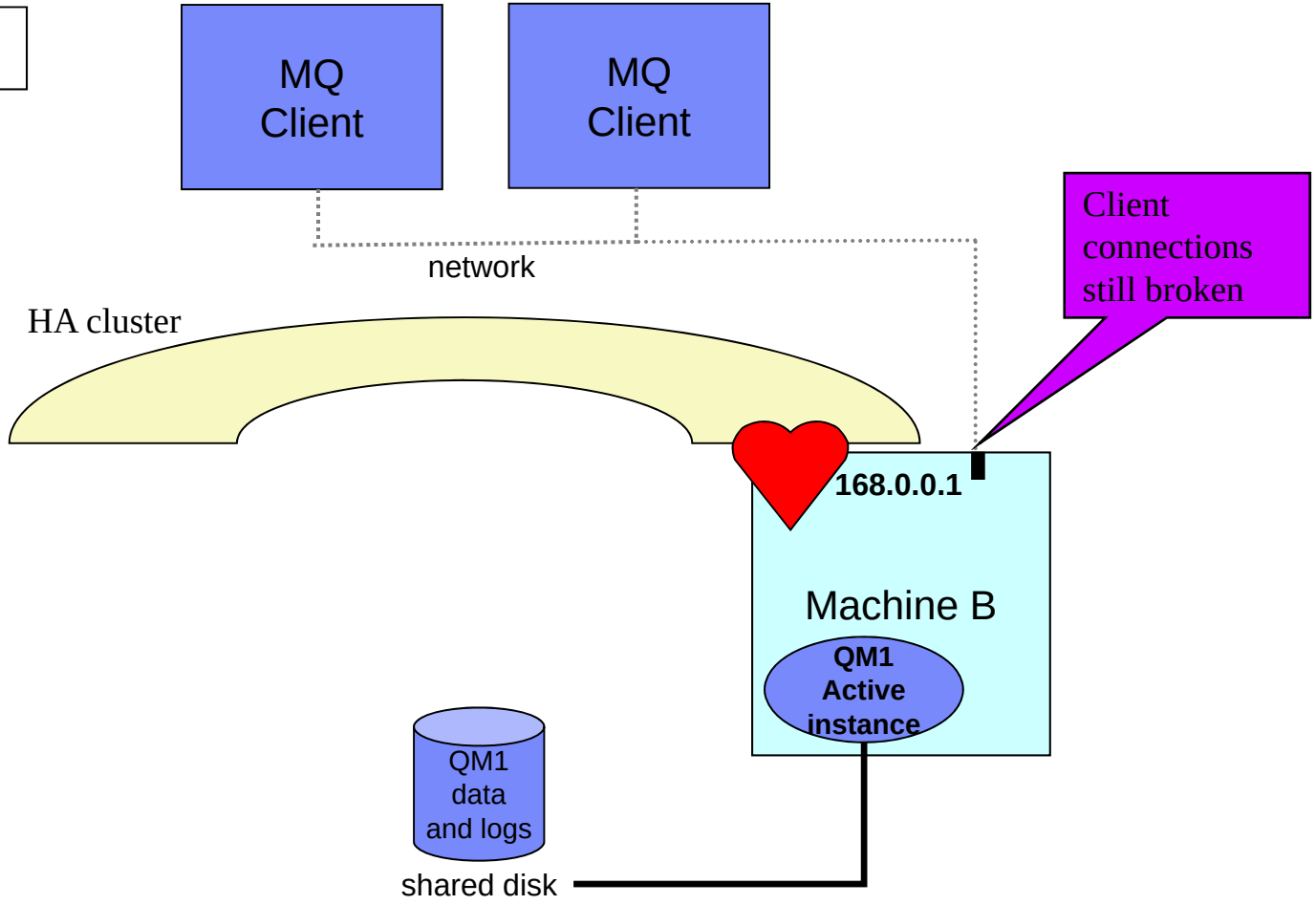


MQ in an HA cluster - Cold standby



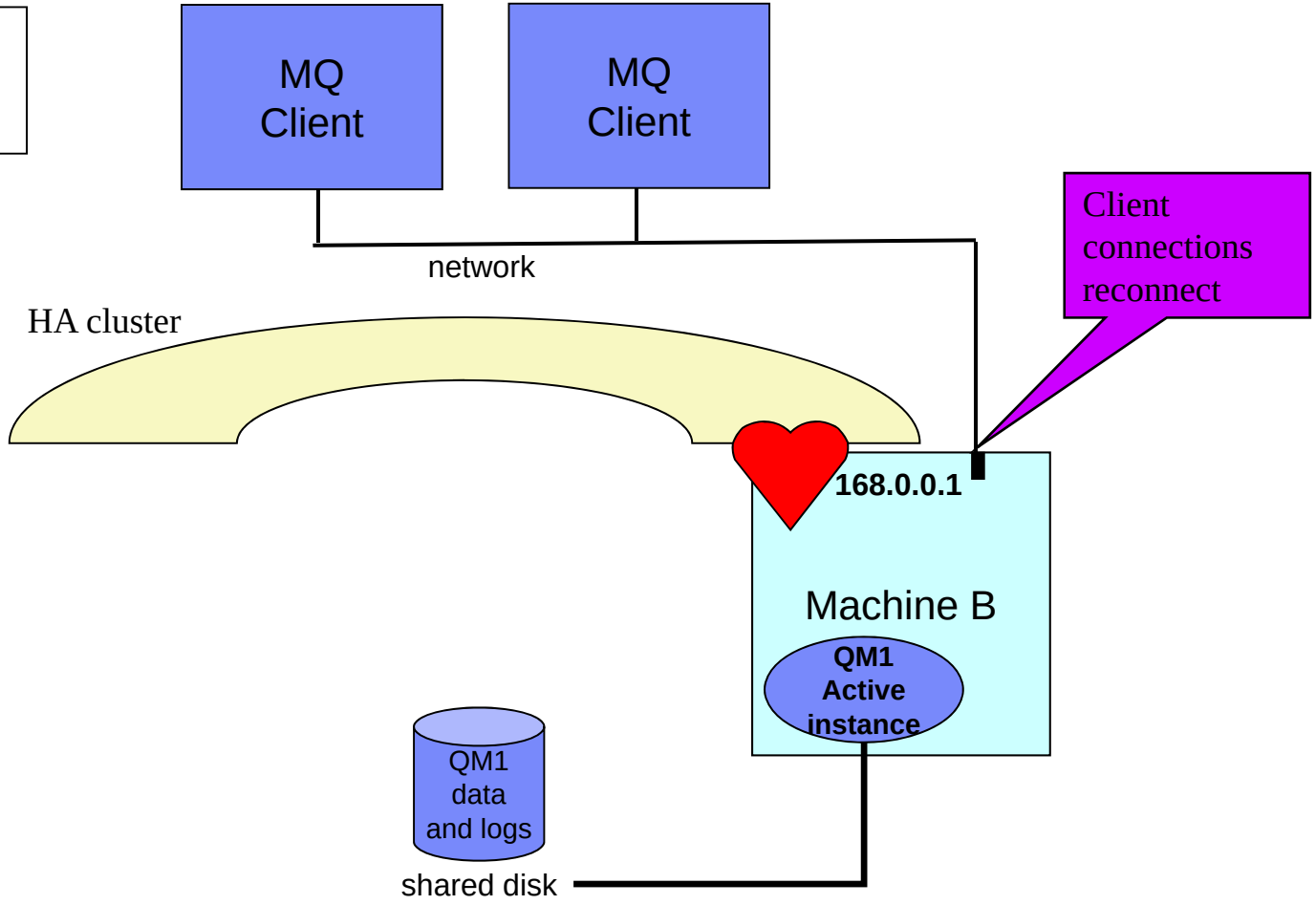
MQ in an HA cluster - Cold standby

3. FAILOVER

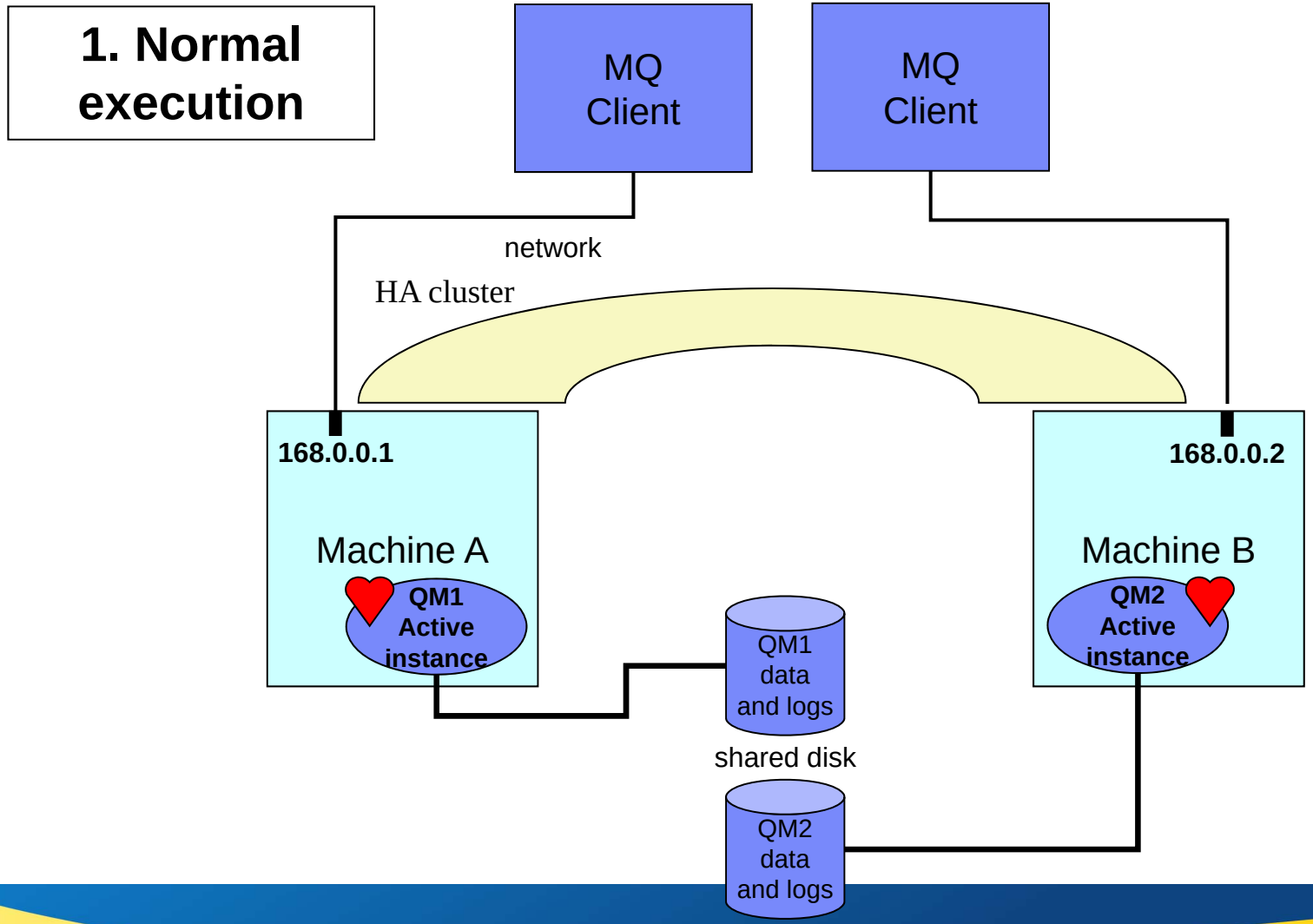


MQ in an HA cluster - Cold standby

4. Recovery complete

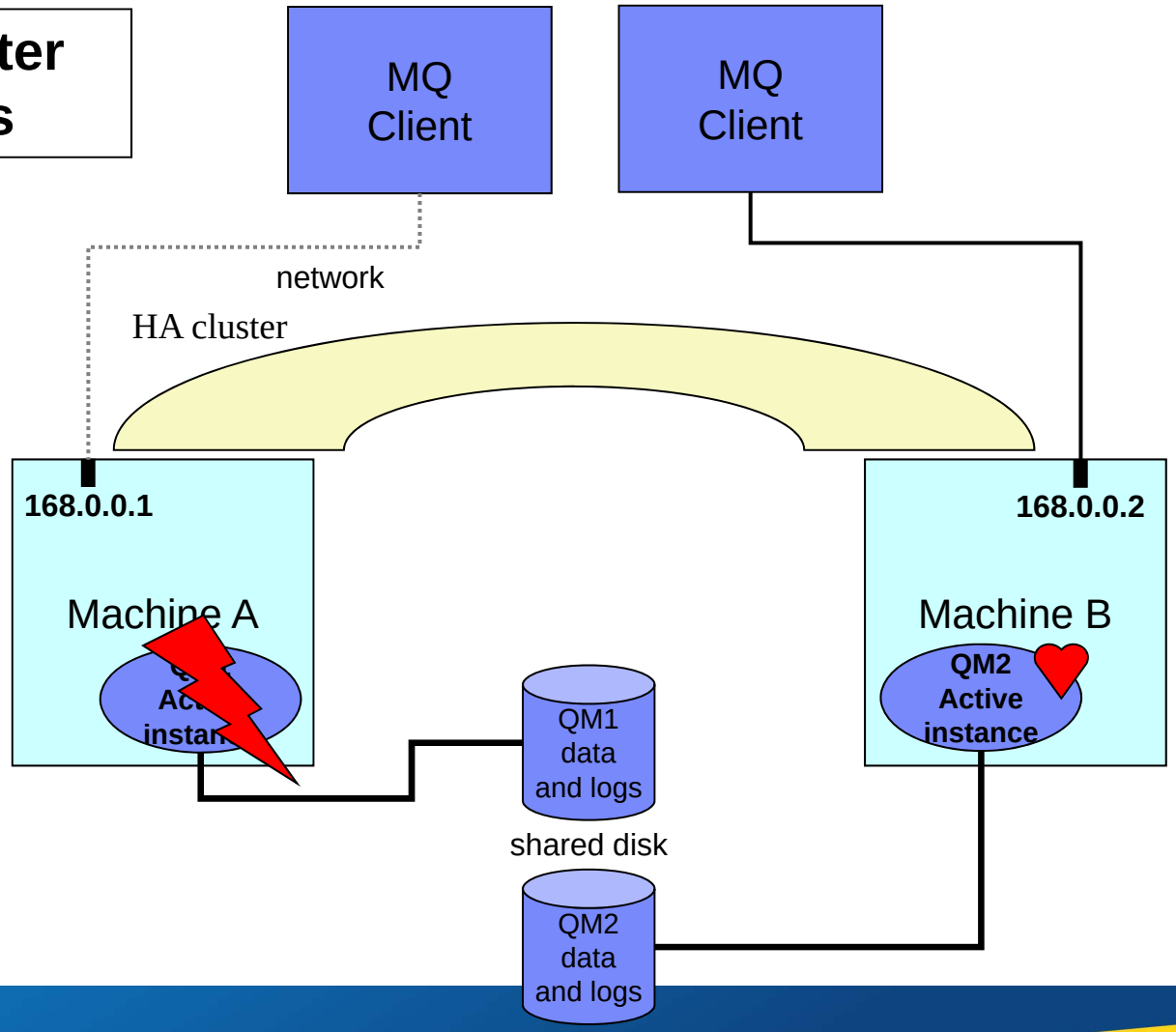


MQ in an HA cluster - Active/active

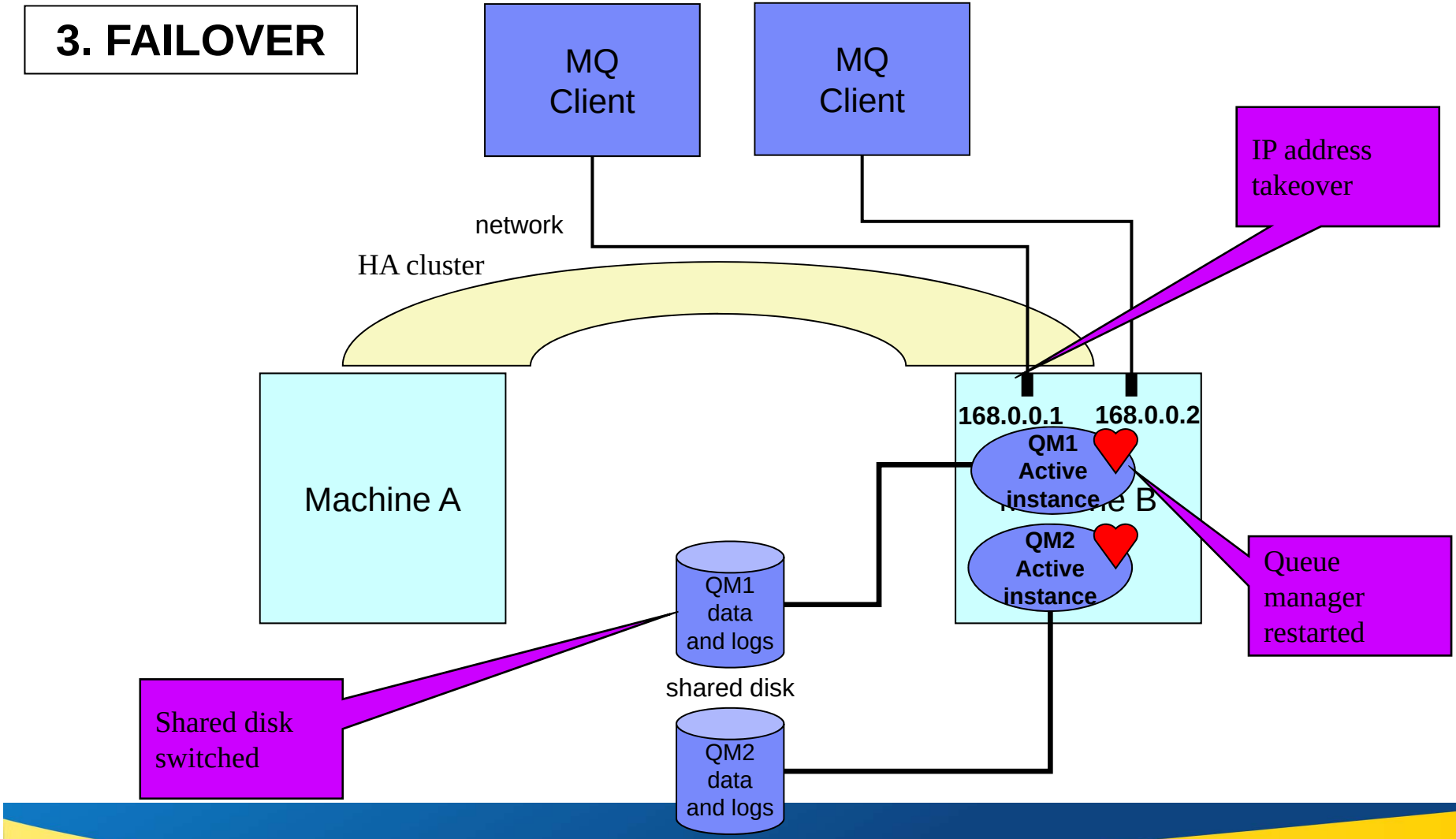


MQ in an HA cluster - Active/active

2. Disaster strikes



MQ in an HA cluster - Active/active



Multi-instance QM or HA cluster?

- **Multi-instance queue manager**
  - ✓ Integrated into the WebSphere MQ product
  - ✓ Faster failover than HA cluster and MC91
    - Delay before queue manager restart is much shorter
  - ✗ Runtime performance of networked storage
  - ✗ More susceptible to MQ and OS defects
- **HA cluster**
  - ✓ Capable of handling a wider range of failures
  - ✓ Failover historically rather slow, but some HA clusters are improving
  - ✗ Some customers frustrated by unnecessary failovers
  - ✗ Require MC91 SupportPac or equivalent configuration
  - ✗ Extra product purchase and skills required
- **Storage distinction**
  - Multi-instance queue manager typically uses NAS
  - HA clustered queue manager typically uses SAN

MC91 SupportPac

- **Scripts for IBM PowerHA for AIX, Veritas Cluster Server and HP Serviceguard**
  - The scripts are easily adaptable for other HA cluster products
- **Scripts provided include:**
  - hacrtmqm - Create queue manager
  - hadltmqm - Delete queue manager
  - halinkmqm - Link queue manager to additional nodes
  - hamqm_start - Start queue manager
  - hamqm_stop - Stop queue manager
  - hamigmqm - Used when migrating from V5.3 to V6

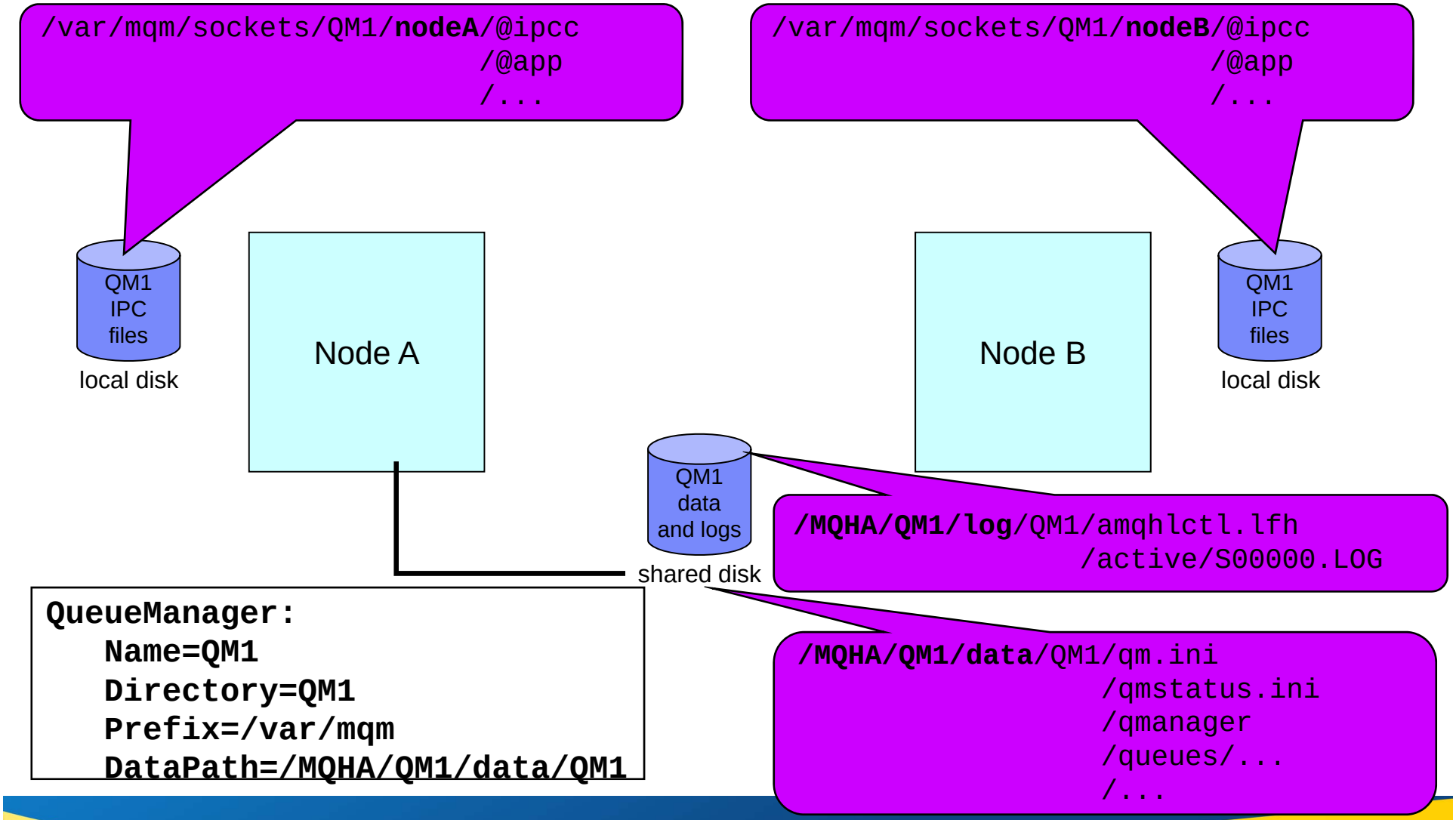
Why withdraw MC91?

- **Dislike of “unsupported” code to use MQ with HA clusters**
  - MC91 was provided “as-is” Category 2 SupportPac
- **MQ 7.0.1 and higher can separate node-specific and shared data without needing environment variables and shell scripts**
  - New DataPath attribute controlled by crtmqm -md
  - Much of what MC91 does is now redundant
- **Each version of MQ means a new version of MC91**
  - Gives customers an extra job when upgrading MQ
- **Support integrated into the product would be preferable**
- **So MC91 has been marked as “withdrawn”**
  - Existing MC91 will still work, but is not really appropriate any more
  - Can still be downloaded but requires extra step

Creating QM in Unix HA cluster

- **Create filesystems on the shared disk, for example**
  - /MQHA/QM1/data for the queue manager data
  - /MQHA/QM1/log for the queue manager logs
- **On one of the nodes:**
- **Mount the filesystems**
- **Create the queue manager**
  - crtmqm -md /MQHA/QM1/data -ld /MQHA/QM1/log QM1
- **Print out the configuration information for use on the other nodes**
  - dspmqinf -o command QM1
- **On the other nodes:**
- **Mount the filesystems**
- **Add the queue manager's configuration information**
  - addmqinf -s QueueManager -v Name=QM1 -v Prefix=/var/mqm -v DataPath=/MQHA/QM1/data/QM1 -v Directory=QM1

Filesystem organisation



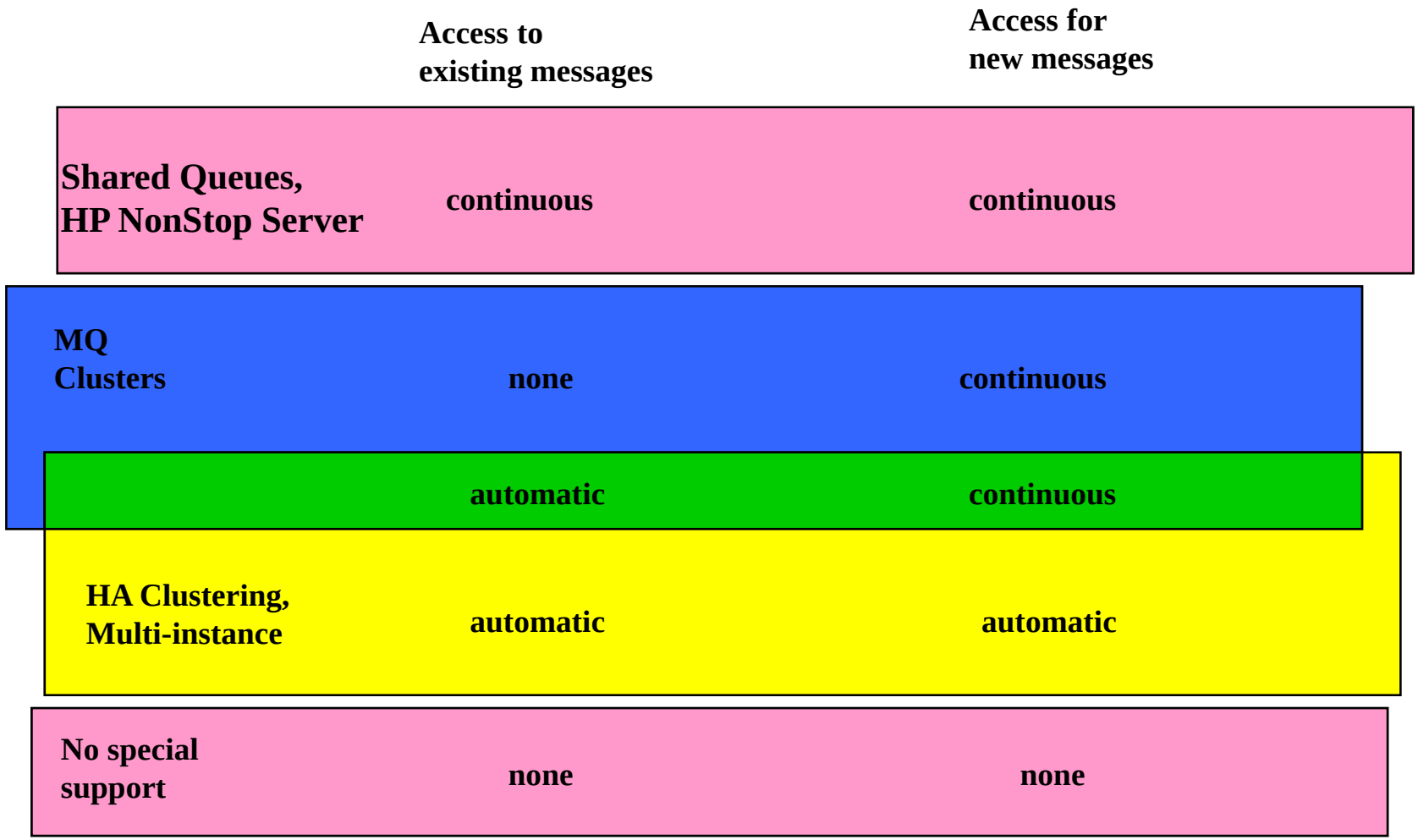
Equivalents to MC91 facilities

MC91	Using MQ 7.0.1
hacrtmqm to create queue manager on shared disk and point symbolic links back to node's /var/mqm	New crtmqm -md option
halinkmqm	New addmqinf command
hadltmqm	New rmvmqinf command to remove queue manager from a node, dltmqm to delete the queue manager
hamqm_start	strmqm
hamqm_stop	
hamqm_applmon	

Summary of Platform Technologies for HA

- **z/OS**
  - Automatic Restart Manager (ARM)
  - Built into product
- **Windows**
  - Microsoft Cluster Service
  - Built into product
- **Unix**
  - IBM PowerHA for AIX (formerly HACMP)
  - Veritas Cluster Server (VCS)
  - HP Serviceguard
  - Previously used MC91
- **Others**
  - HP NonStop Server
  - ...other platforms/HA technologies possible

Comparison of Technologies



APPLICATIONS AND AUTO-RECONNECTION

HA applications - MQ connectivity

- **If an application loses connection to a queue manager, what does it do?**
  - End abnormally
  - Handle the failure and retry the connection
  - Reconnect automatically thanks to application container
    - WebSphere Application Server contains logic to reconnect
  - Use MQ automatic client reconnection

Automatic client reconnection

- **MQ client automatically reconnects when connection broken**
  - MQI C clients and JMS clients
- **Reconnection includes reopening queues, remaking subscriptions**
  - All MQI handles keep their original values
- **Can connect back to the same queue manager or another, equivalent queue manager**
- **MQI or JMS calls block until connection is remade**
  - By default, will wait for up to 30 minutes
  - Long enough for a queue manager failover (even a really *slow* one)

Automatic client reconnection

- **Can register event handler to observe reconnection**
- **Not all MQI is seamless, but majority repaired transparently**
  - Browse cursors revert to the top of the queue
  - Non-persistent messages are discarded during restart
  - Nondurable subscriptions are remade and may miss some messages
  - In-flight transactions backed out
- **Tries to keep dynamic queues with same name**
  - If queue manager doesn't restart, reconnecting client's TDQs are kept for a while in case it reconnects
  - If queue manager does restart, TDQs are recreated when it reconnects

Automatic client reconnection

- **Enabled in application code or ini file**
  - MQI: MQCNO_RECONNECT, MQCNO_RECONNECT_Q_MGR
  - JMS: Connection factories/activation specification properties
- **Plenty of opportunity for configuration**
  - Reconnection timeout
  - Frequency of reconnection attempts
- **Requires:**
  - Threaded client
  - v7.0.1 or higher server - including z/OS
  - Full-duplex client communications (SHARECNV >= 1)

Client Configurations for Availability

- **Use wildcarded queue manager names in CCDT**
  - Gets weighted distribution of connections
  - Selects a “random” queue manager from an equivalent set
- **Use multiple addresses in a CONNAME**
  - Could potentially point at different queue managers
  - More likely pointing at the same queue manager in a multi-instance setup
- **Use automatic reconnection**
- **Can use all of these in combination!**

Summary

- **MQ and operating system products provide lots of options to assist with availability**
  - Many interact and can work well in conjunction with one another
- **But it's the whole stack which is important ...**
  - Think of your application designs
  - Ensure your application works in these environments
- **Decide which failures you need to protect against**
  - And the potential effects of those failures
- **The least available component of your application determines the overall availability of your application**
- **Also look for other publications**
  - RedBook SG24-7839 “High Availability in WebSphere Messaging Solutions”

Questions & Answers