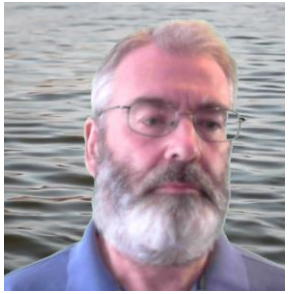
What's New in IBM MQ

December 2021 – includes IBM MQ 9.2.4



Mark Taylor

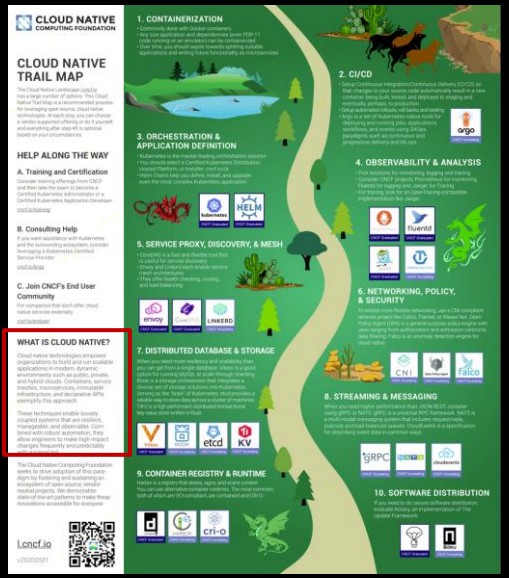
marke_taylor@uk.ibm.com

IBM Hursley

Vision: IBM MQ is the **cloud native** choice for enterprise messaging

How can IBM MQ be cloud native? What is *cloud native?*

CLOUD NATIVE COMPUTING FOUNDATION



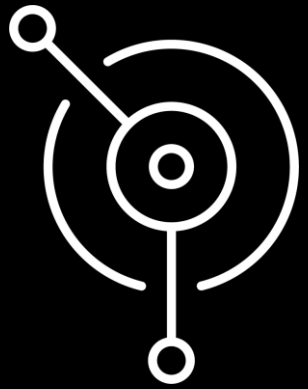
github.com/cncf/landscape#trail-map

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

A focus on where you need MQ today and tomorrow



On-premise, software and the MQ Appliance, exactly as you need it

IBM Z

Linux

AIX

Windows

IBMi

HPE NonStop

zLinux

Appliance

Run MQ yourself in public or private clouds, virtual machines or containers

AWS

Azure

IBM Cloud

RED HAT OPENSHIFT

Kubernetes

Let IBM host MQ for you with its managed SaaS MQ service in public clouds, IBM Cloud and AWS

IBM Cloud

powered by aws

IBM MQ: long term support and continuous delivery

In 2016 MQ introduced a dual Long Term Support and a Continuous Delivery model.

Continuous Delivery

New CD versions of MQ are released approximately every four months, incrementally introducing new product capabilities.

Requires users to migrate forward within 12 months.

Long Term Support

Approximately every two years a new LTS version is released, rolling up the CD capabilities into a release with 5+3 support attached.

Required by those looking for fixed function.

Mix and Match

Both are available under the same license.

Both can interoperate either between servers or clients, just like any previous version of MQ.

All the function delivered in the 9.1.x CD releases is available in the long term support release **V9.2 LTS**

MQ 9.2 LTS content...

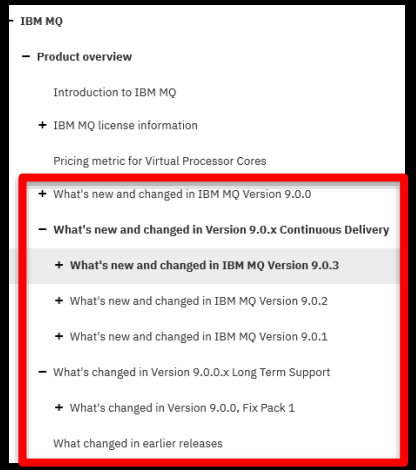
Uniform Cluster automatic application rebalancing	Microsoft .NET Core support	Client connectivity with zCEE	Developer toolkit for MacOS	Automatic TLS CipherSpec negotiation	Enhanced Salesforce Bridge	Build toolkit for zCEE	Idempotent MQSC commands	Browse messages using REST	MQ Appliance certificate expiry notifications
Channel enabled AMS policies for z/OS	JSON format CCDT	Permitted TLS CipherSpec control	REST messaging performance enhancements	Full JSON-syntax REST administration	MQ Appliance HA event notifications	Improved distributed queue manager restart times	Stream MQ Appliance error logs	Rapid Uniform Cluster rebalancing	Improved MQIPT management
New application status checking	ini file and MQSC injection at startup	Escalating end queue manager	MQFT REST list resource monitors	Enhanced Blockchain Bridge	WebSphere Liberty MDB pause	New consistent MQ samples	MFT REST create file transfer	FTP server support on IBM I for MFT	AMS HSM with Oracle JRE
MQ Appliance admin activity audit logging	XA support in Liberty for decoupled JMS connections	Automatic Uniform Cluster configuration	Packaged MQ Internet Passthru (IPT)	Highly available MFT Agent deployments	z/OS data set encryption support	User controlled application naming	TLS 1.3 support ...	High speed transfer over long distances with Fasp.io	Qpid JMS shared subscriptions
Publish messages over REST	.NET project templates	Increased queue size support for Distributed	New improved Web Console	Full HA-DR-HA replicated data queue manager deployments	Uniform Cluster application monitoring	Java 11 application support	Distributed queue size control		

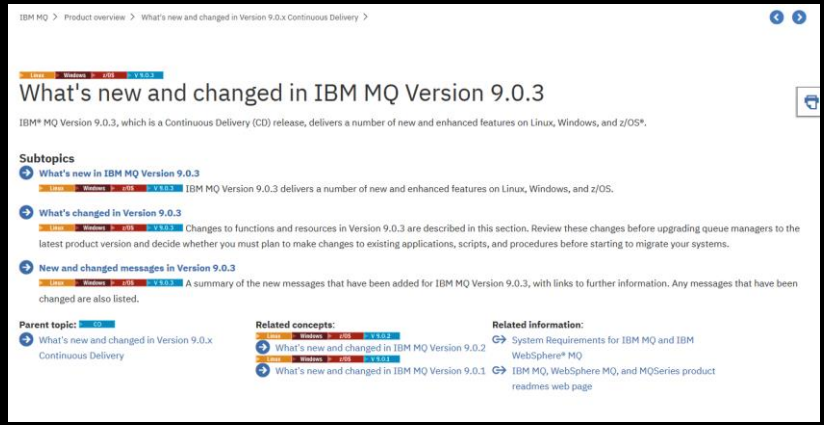
And since then with CD ...

Idempotent delete operations	Linux upgrade in place	New Web Console accelerated experiences	Hostname SNI routing	Default long password support for Java apps	AMQP point-to-point support	Keda scaler for autoscaling container apps	Syncronous replication for Appliance DR	Recreate Appliance DR secondary operation	Native HA for CP4I
Reduced cost non-prod license	Start/stop resource monitors independently	Containerised MFT agents on DockerHub	Last in sync reporting for RDQM	Failed resource action resolution control	TLS enabled by default for MQ on Cloud	Streaming Queues	Ansible improvements for z/OS	Remote Admin for Web Console	Uniform Cluster Patterns
AT-TLS	Java 17	SMF timer granularity	Web Console Message Handling	Appliance REST API extensions					

MQ release-to-release changes

Always read the **What's new and changed** sections of the Documentation to see what each release adds





MQ in Containers, continually evolving

MQ first supported Docker containers in 2015, showing how a stateful solution can run in an often stateless world.

MQ was one of the first certified containers available on IBM's Kubernetes platform, IBM Cloud Private. Showing how to run MQ in a managed container environment.

MQ added support for running on Red Hat OpenShift

MQ is a core component of IBM's Cloud Pak for Integration, providing enterprise messaging for the Integration Platform solution

2015 **hub.docker.com/r/ibmcom/mq** **github.com/ibm-messaging/mq-container** 2021

MQ within the Cloud Pak for Integration

Strategic focus

IBM is committed to building ever increasing value into its IBM and Red Hat OpenShift platform

Red Hat OpenShift

Certified Container

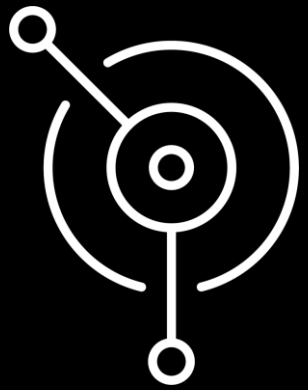
Production ready container images with a Kubernetes Operator that simplifies the operational activities

Deep Insight

Built-in emission of logging and tracing data, empowering developers and administrator to observe and troubleshoot

Flexible Adoption

A flexible deployment model allowing traditional software and container technology to be adopted at your own speed

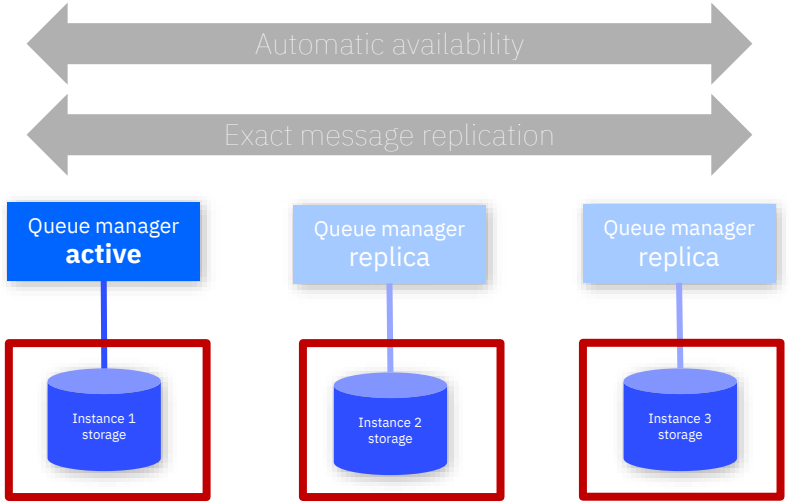


Cloud native availability

Replication and consensus

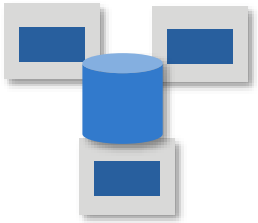
MQ Native HA

New in MQ 9.2.3
Available for OpenShift with
Cloud Pak for Integration

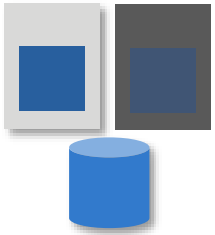


Messages persisted in three locations
Exact replicas, maintaining configuration, message order, transactional state
Quorum ensures consistency and rapid failure detection and recovery

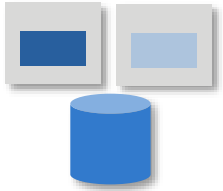
Constantly evolving to meet your availability needs



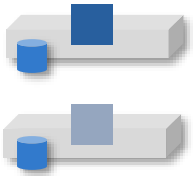
z/OS Queue Sharing Groups



System managed HA

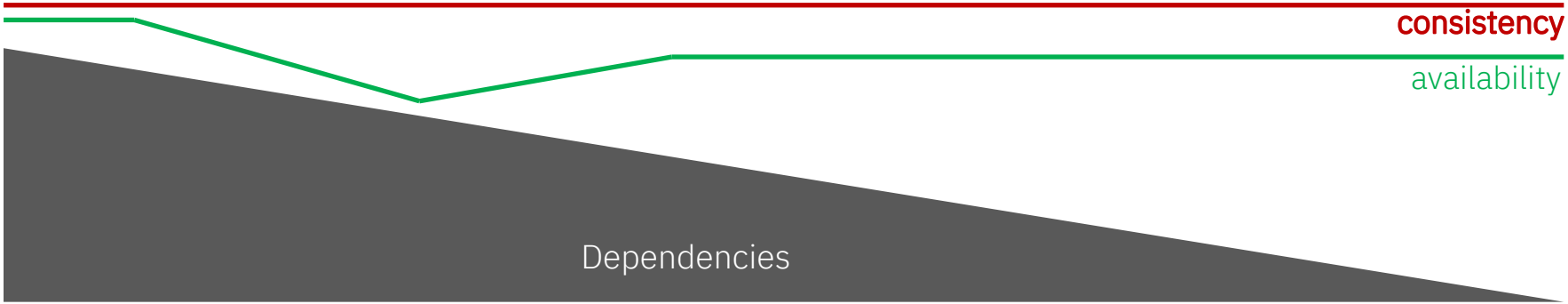


Multi-instance queue managers

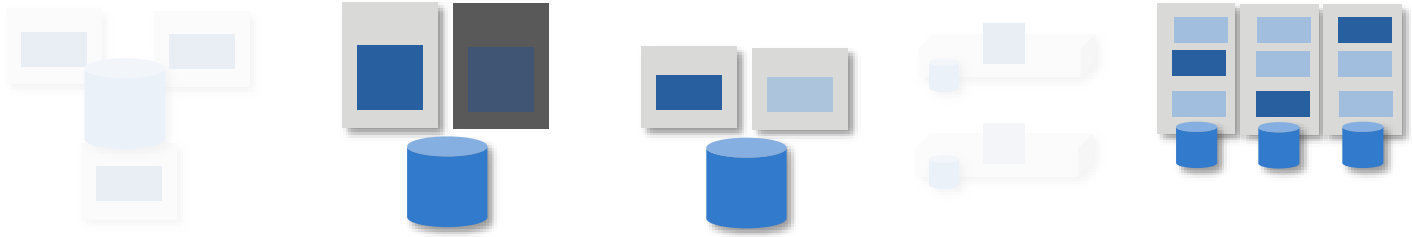


MQ Appliance

Replicated data queue manager



Message availability in the cloud



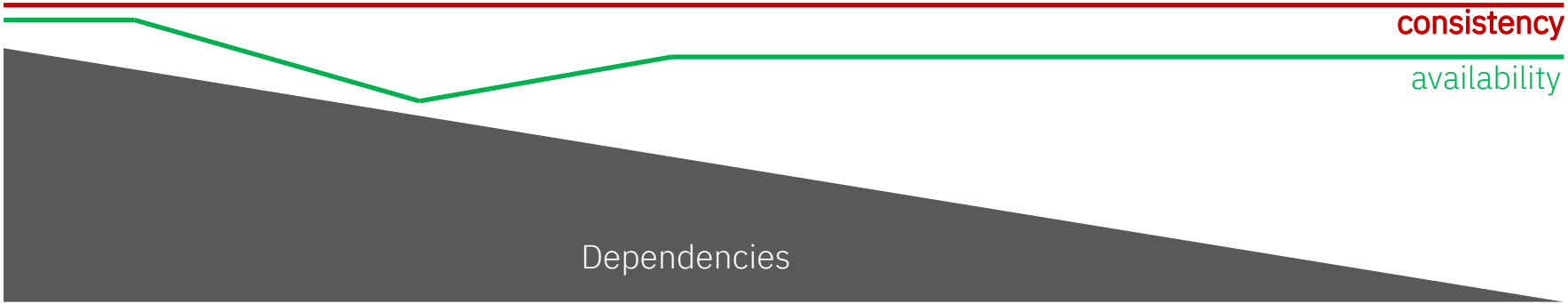
z/OS Queue Sharing Groups

System managed HA

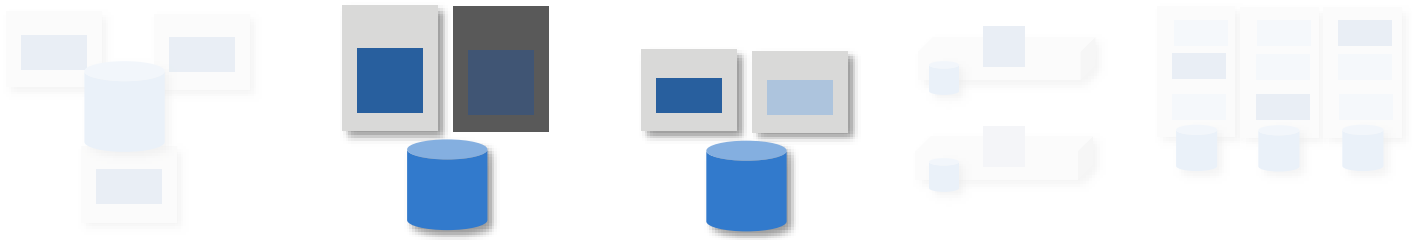
Multi-instance queue managers

MQ Appliance

Replicated data queue manager



Message availability in containers



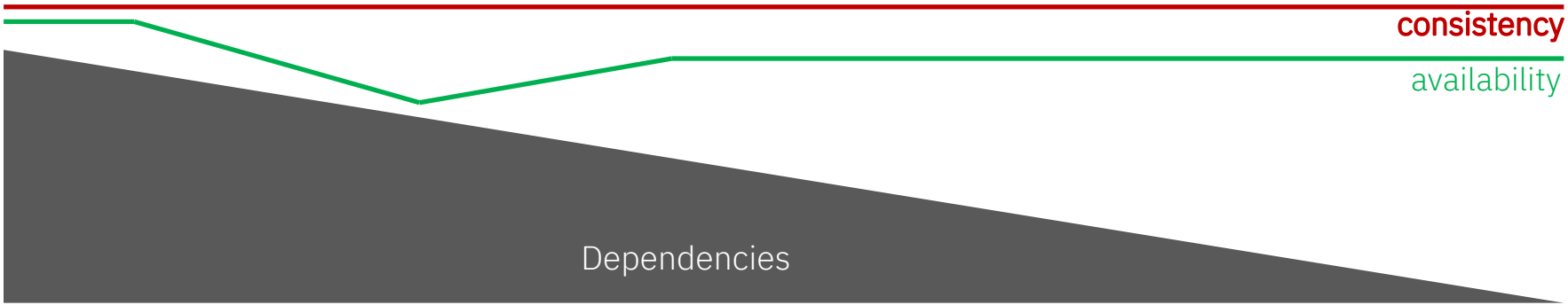
z/OS Queue Sharing Groups

System managed HA

Multi-instance queue managers

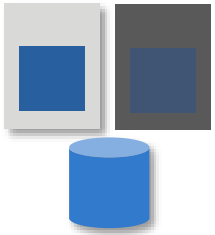
MQ Appliance

Replicated data queue manager

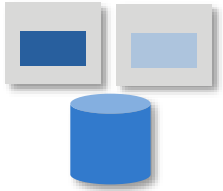


Cloud native message availability

z/OS Queue Sharing Groups



System managed HA



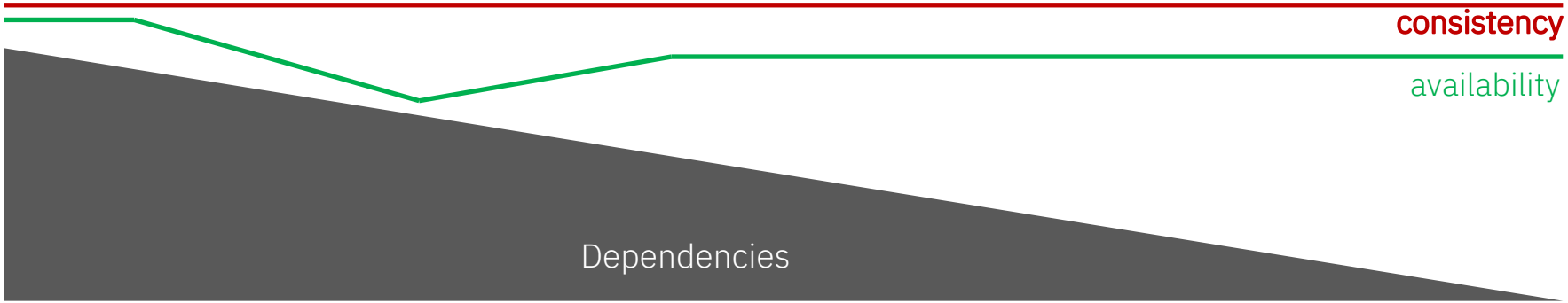
Multi-instance queue managers

MQ Appliance

Replicated data queue manager

MQ 9.2.3 CD
in OpenShift with Cloud Pak for Integration

Native HA



MQ Native HA

Solution: Convert MQ's persistence layer to be cloud native

New in MQ 9.2.3
Available for OpenShift with
Cloud Pak for Integration

Problems to solve:

MQ persistent data replicated across AZs
Consistency across replicas guaranteed
Fast and reliable failure detection and fail over

Raft

A proven, yet *understandable*, consensus algorithm

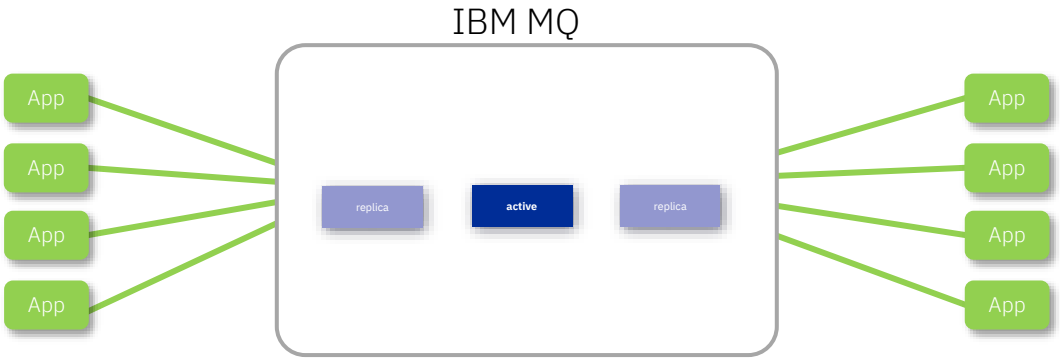
Based on the concept of a **sequential log of state changes**

IBM MQ

A proven, high performing and reliable, messaging solution

Built from day one around a **sequential log of state changes**

A messaging and event service



Purpose

Loosely couple applications

Shield applications from their own availability issues

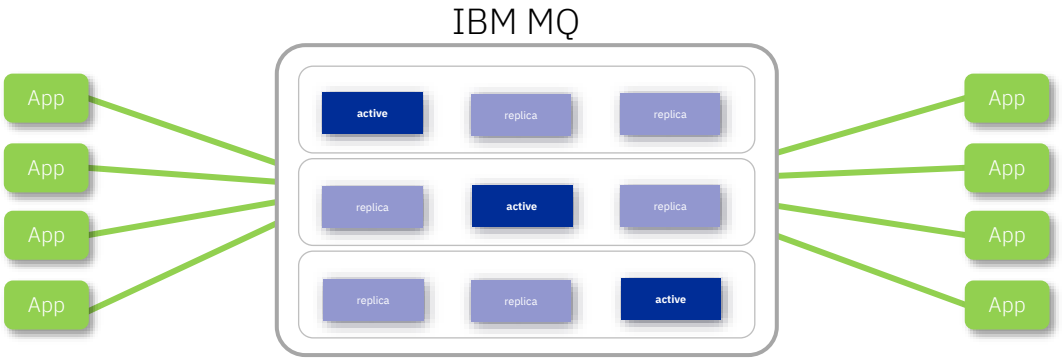
Requirements

Scale with the application

Don’t lose the messages

Be more available than the applications

A messaging and event service



Purpose

Loosely couple applications

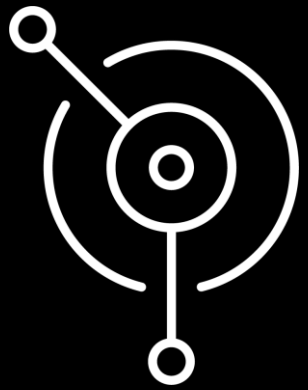
Shield applications from their own availability issues

Requirements

Scale with the application

Don’t lose the messages

Be more available than the applications



Always-on

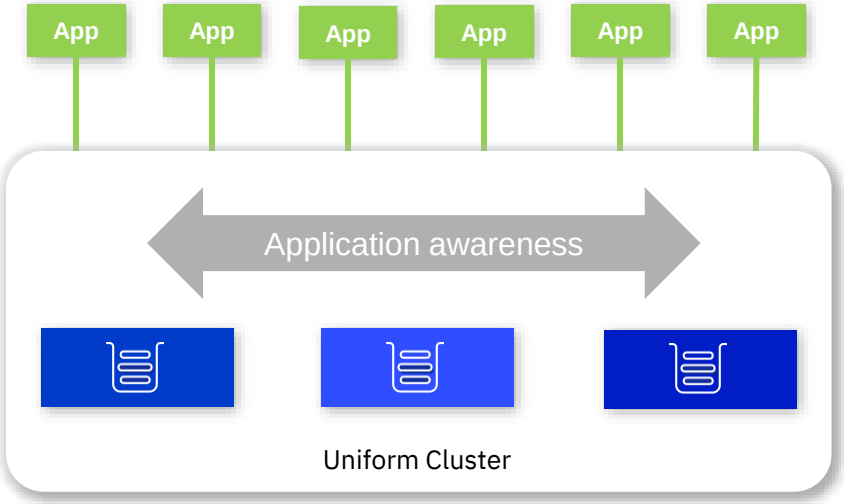
Building scalable, active-active, solutions

Always-on MQ

To provide an active/active, solution you need to consider multiple active queue managers acting as a *single messaging service*

Applications should treat the queue managers as interchangeable and want to connect to the group in the most efficient and available distribution

With IBM MQ 9.2 LTS, queue managers can form a **uniform cluster**, each queue manager provides the same messaging capabilities

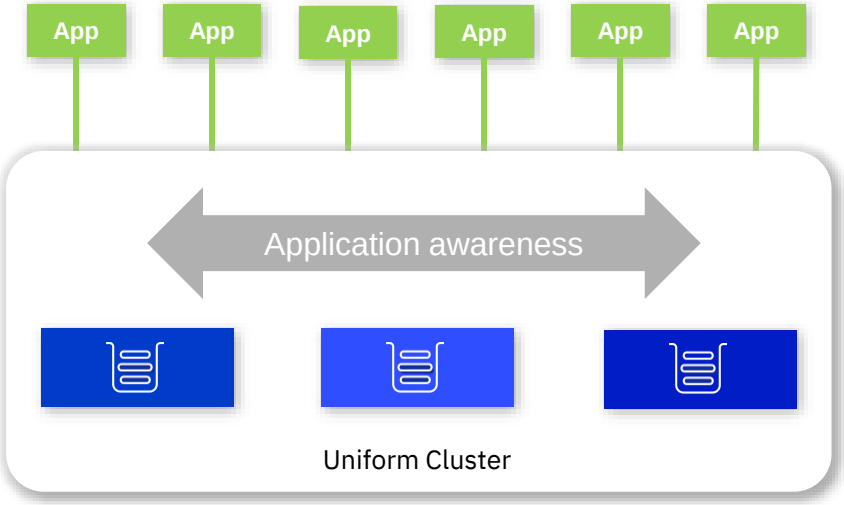


Always-on MQ

To provide an active/active, solution you need to consider multiple active queue managers acting as a *single messaging service*

Applications should treat the queue managers as interchangeable and want to connect to the group in the most efficient and available distribution

With IBM MQ 9.2 LTS, queue managers can form a **uniform cluster**, each queue manager provides the same messaging capabilities



Application language and environment support has been growing ever since MQ first delivered uniform clusters.

IBM MQ 9.2.3 Resource Adapter adds JEE Message Driven Bean support to automatically balance your clustered MDB applications.

New in MQ 9.2.3
Resource Adapter

Increased range of application styles supported with uniform clusters

Uniform clusters work best with decoupled applications, ones that have little affinity or have been designed appropriately for active/active deployments

Good use cases

Applications that can handle being moved from one queue manager to another without even realising and can run with multiple instances

- Datagram producers, e.g. events
- Services that respond to request messages
- No message ordering requirements
- MDBs

Poor use cases

Applications that create persistent state across multiple messaging operations, or require a single instance to be running

- Requestors waiting for specific replies
- Dependant on message ordering
- Transactional applications (‘works’, but far from optimal)
- ‘Managed’ environments (e.g. JEE)

From MQ 9.2.4

Increased range of application styles supported with uniform clusters

Uniform clusters work best with decoupled applications, ones that have little affinity or have been designed appropriately for active/active deployments

Good use cases

Applications that can handle being moved from one queue manager to another without even realising and can run with multiple instances

- **Requestors waiting for specific replies**
- **Transactional applications**
- Datagram producers, e.g. events
- Services that respond to request messages
- No message ordering requirements
- MDBs

Poor use cases

Applications that create persistent state across multiple messaging operations, or require a single instance to be running

- Dependant on message ordering
- ‘Managed’ environments (e.g. JEE)

From MQ 9.2.4

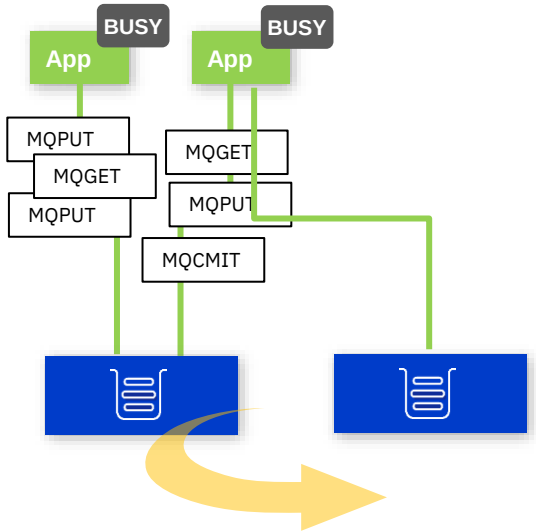
Transactional applications

To avoid frequent rollbacks, the **default** behaviour will change for applications which are in a transaction

This can be overridden if you **want** the interruptions!

Applications currently processing a unit of work will wait until commit/rollback to reconnect

If no application eligible to move before configurable time limit reached, one will be interrupted anyway



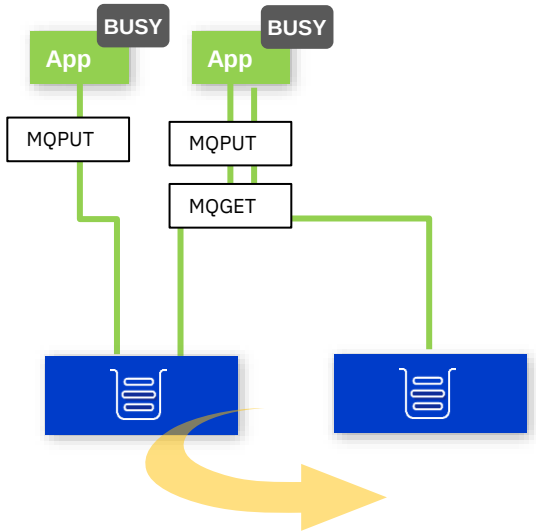
Request/reply applications

From MQ 9.2.4

If you tell MQ that an application is performing request/reply messaging, it will wait until any outstanding response arrives before moving a connection

’Outstanding’ will take into account request expiry if any

As with transactional applications, there will be a configurable backstop ‘timeout’ mechanism to prevent applications refusing to move forever



From MQ 9.2.4

How to use the new options:

In code:

```
MQCNO cno = {MQCNO_DEFAULT};
MQBNO bno = {MQBNO_DEFAULT};

cno.Version = MQCNO_VERSION_8;
cno.BalanceParmsPtr = &bno;

bno.Timeout = 50;
bno.ApplType = MQBNO_BALTYPE_SIMPLE;
```

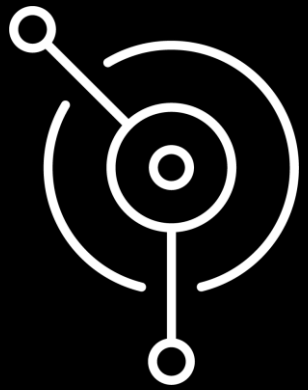
(and/or)

In config(client.ini):

```
[...]

Application:
  Name=MyApp
  Type=Simple
  BalanceTimeout=default
```

Changes are a collaboration between queue manager and client so both should be at 9.2.4
If you just update the queue manager then the defaults will change for transactions, but request reply will work as before



Insight to your data

Stream MQ data to new applications

MQ Streaming Queues

Tap into the value of existing data flowing over MQ by making message data available to Kafka, AI, and analytics applications with **zero impact to the existing applications or their messages**, and without a need for re-architecting your message flows.

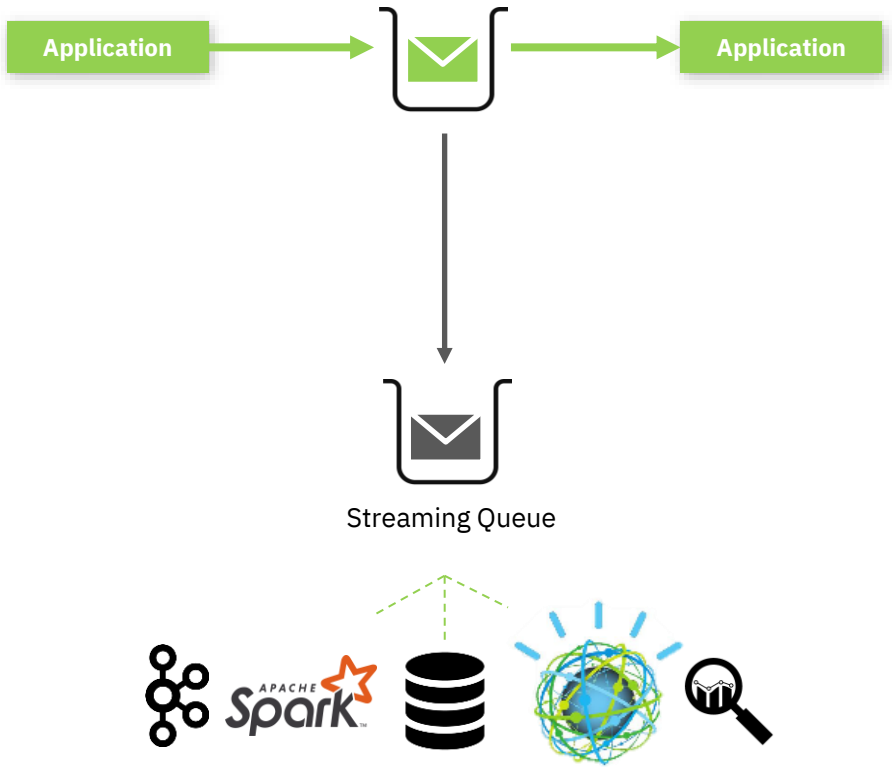
1. **Streaming Processing** to accelerate time to insight from existing data.
2. **Real world data** to accurately simulate production workloads to test the impact of architectural changes on applications.
3. **Auditing and Replay** of data in the event of disasters. Auditing and replay use cases require exact duplicates of message content as well as message attributes including Message IDs, Correl IDs etc.

MQ Streaming Queues

New in MQ 9.2.3
Distributed platforms

Tap into the value of existing data flowing over MQ by making message data available to Kafka, AI, and analytics applications with **zero impact to the existing applications or their messages**, and without a need for re-architecting your message flows.

1. **Streaming Processing** to accelerate time to insight from existing data.
2. **Real world data** to accurately simulate production workloads to test the impact of architectural changes on applications.
3. **Auditing and Replay** of data in the event of disasters. Auditing and replay use cases require exact duplicates of message content as well as message attributes including Message IDs, Correl IDs etc.

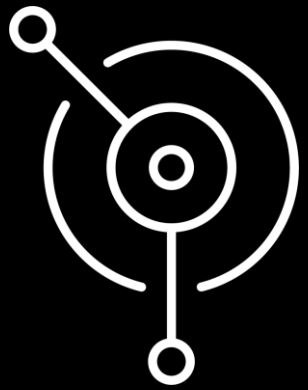


Streaming queues configuration

New attributes for LOCAL and MODEL queues:

- STREAMQ - The name of the streaming queue to put duplicate messages
- STRMQOS - The quality of service to use when delivering messages to the streaming queue. Either:
  - MUSTDUP - Put of message to both original and streaming queues must succeed, otherwise overall put operation fails
  - BESTEF - A failure to put message to streaming queue will not affect the outcome of the put of the message to the original queue (this is the default)

```
DEFINE QLOCAL(Q1)
  STREAMQ(QDUP)
  STRMQOS(MUSTDUP)
```

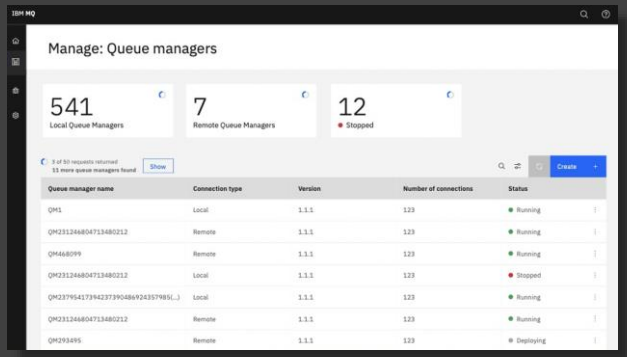


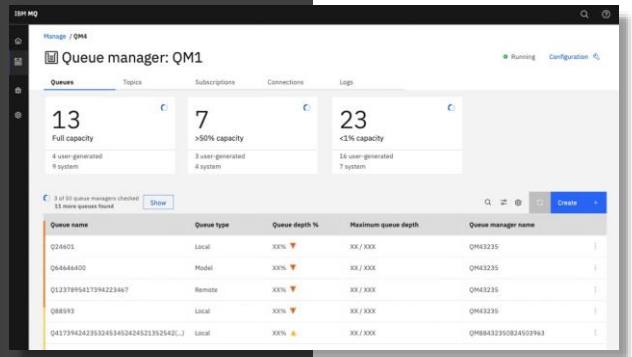
Managing MQ

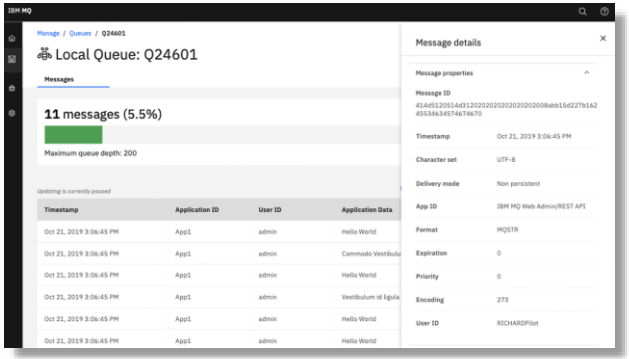
New Web Console

MQ 9.2 replaces the existing web console with a new web console across all platforms

Focus is on user experience and consistency across IBM products







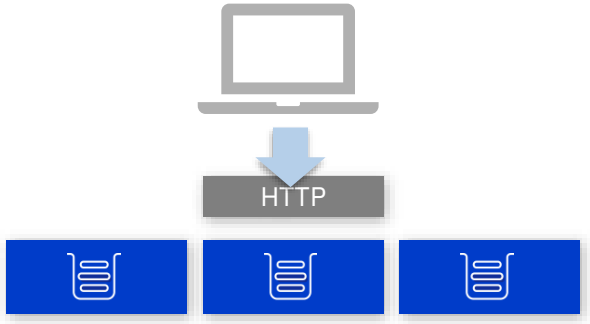
https://community.ibm.com/community/user/imwuc/blogs/callum-jackson1/2020/04/09/enhanced-web-console-in-ibm-mq-915

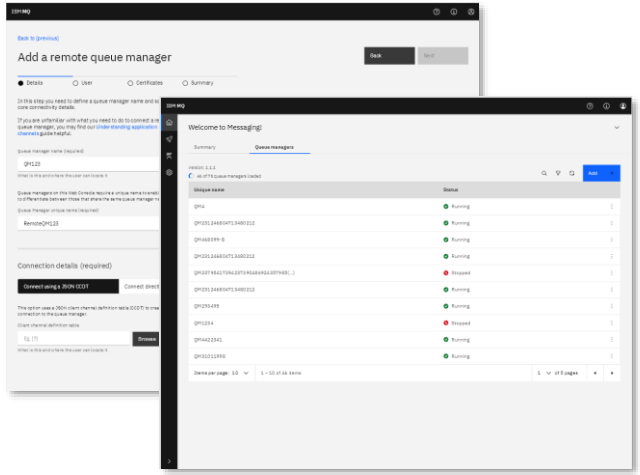
Central Web Console

New in MQ 9.2.3
All installable platforms

Originally, the web server component of MQ that underpins the web console was co-located with the queue managers. A simple way to point at each MQ installation and see the queue managers there.

With IBM MQ 9.2.3 CD you can point a browser at a single system, one that just hosts the MQ web server, and now manage multiple queue managers across multiple systems, of any type.





Further web console improvements

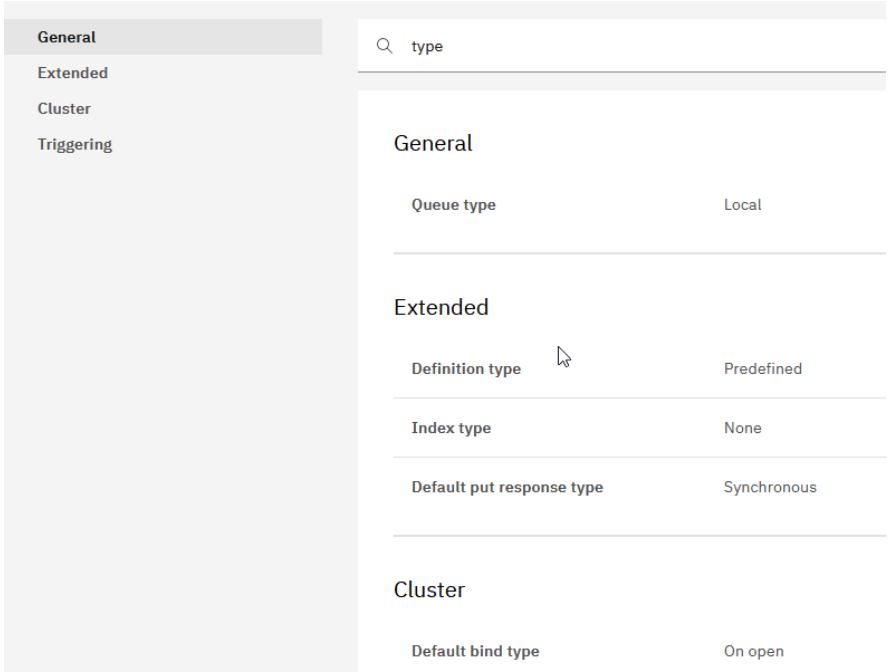
New in MQ 9.2.4
All installable platforms

Filtering for attributes on property pages (shown)

Messages can now be downloaded from queues, both text and binary

The ability to control how much message text is shown / downloaded (previously limited to 1000 characters)

Can toggle dark mode from settings



MQ Appliance enhancements

The MQ administrative REST API has been enhanced so that HA and DR state information can be queried

Similar information is provided to the output of the dspmq –o ha | dr commands

The MQ Console has also been enhanced to support the failed resource action capability added in 9.2.2 for HA enabled queue managers

I.e. the ability to see any resource that has a failure associated with it, and to clear it if required

HTTP GET:
https://host:port/ibmmq/rest/v2/admin/qmgr?ha=*


```
{"qmgr": [{
 "name": "HAQM1",
 "state": "running",
 "ha": {
  "type": "replicated",
  "floatingIPAddress": “9.20.10.4”,
  "floatingIPInterface": “eth10”
 }
}]}
```

More flexible monitoring

Customers increasingly want to understand what their messaging system is doing and SMF is a natural way to do this on z/OS

With MQ 9.2.4 it is possible to generate statistics records (SMF 115) every second allowing for high fidelity monitoring

However customers typically don't require collection of accounting data (SMF 116) at the same frequency so 9.2.4 also allows statistics and accounting data to be collected at different intervals

SET SYSTEM STATIME(0.05) ACCTIME(30)

Collect stats data every 5 seconds and accounting data every 30 minutes

CSQUDSPM

In the 9.0.* CD releases a new utility was added to z/OS: CSQUDSPM

This is the equivalent to dspmq on distributed and gives details about queue managers available on an LPAR

Not many people know about it, so here is a bit of advertising… In 9.2.4 we made a minor tweak so that it will accept upper case parameters which makes it easier to call from JCL

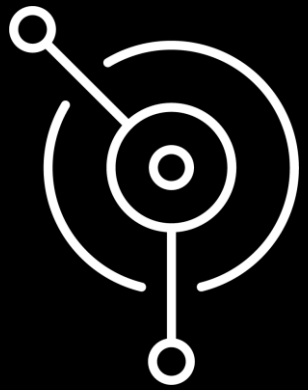
DSPMQ -O ALL

QMNAME(MQ21) STATUS(Running) INSTVER(9.2.4) ERLYVER(9.2.4) CMDPFX(!MQ21) QSGNAME(SQ21) RELTYPE(CDR)

QMNAME(MQ22) STATUS(Running) INSTVER(9.2.4) ERLYVER(9.2.4) CMDPFX(!MQ22) QSGNAME(SQ21) RELTYPE(CDR)

QMNAME(MQ23) STATUS(Running) INSTVER(9.2.0) ERLYVER(9.2.4) CMDPFX(!MQ23) QSGNAME(SQ21) RELTYPE(LTS)

QMNAME(MQ24) STATUS(Running) INSTVER(9.2.0) ERLYVER(9.2.4) CMDPFX(!MQ24) QSGNAME(SQ21) RELTYPE(LTS)



Securing MQ

Security on Distributed platforms

It is now possible to define permissions for users without users having to be defined in a LDAP repository. This is driven by the OpenShift security best practices.

A custom hostname instead of a channel name can now be set in the TLS SNI (Server Name Indicator) header. This makes it easier to route through third-party network layers. This change improves the experience of configuring MQ with OpenShift network routing.

MQ Java clients will now support long passwords by default. Previously the default was limited to 12 characters.

Many components that store passwords in files have been upgraded for improved security

MQ client enhancements

Java 17 (Oracle or Adoptium) now supported with MQ classes for Java and JMS ensuring application currency

TLS 1.3 support is now provided when using the JRE that comes with MQ

The .NET client now provides parity with other clients by allowing users to control whether the name of the channel or the hostname is sent in the TLS SNI extension. This simplifies configuration when connecting to Red Hat OpenShift as you don't need to define an OpenShift route

AT-TLS

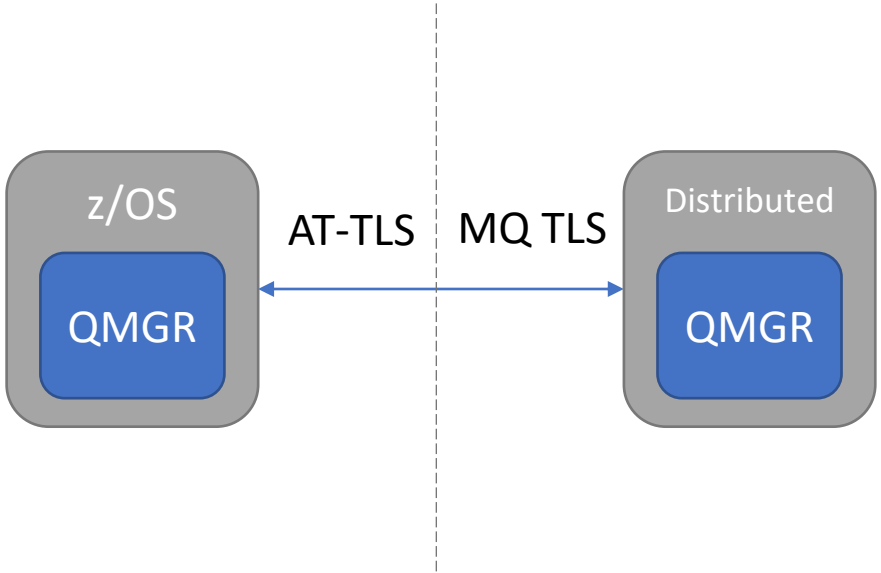
More and more z/OS customers are centralizing their TLS configuration by defining AT-TLS policies for all their middleware

This is trivial when connecting a pair of z/OS queue managers, but more tricky when going between distributed and z/OS

IBM documentation now provides guidance on how to use AT-TLS with MQ for z/OS for the following scenarios:

- z/OS to z/OS
- z/OS to distributed and vice-versa
- Distributed client to z/OS

For both single, and alias CipherSpecs



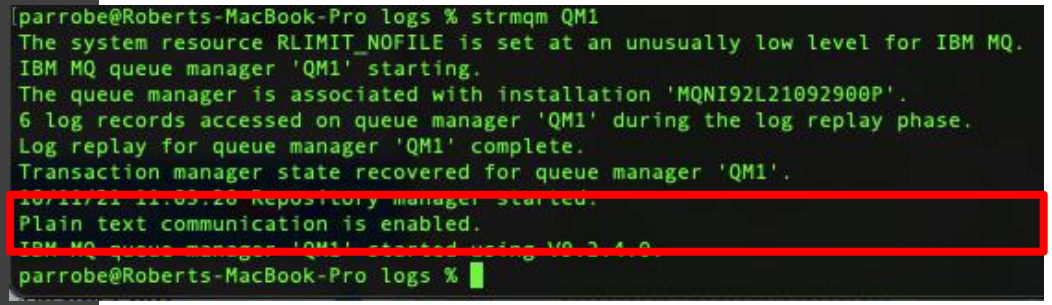
SecureCommsOnly

Distributed queue managers can now be configured so that they will only allow channels to be started up if they are TLS enabled

This ensures that administrative errors where a channel is defined with a blank SSLCIPH can't lead to a security breach

Enabled via the SecureCommsOnly = YES | NO parameter in the TCP stanza of the qm.ini file

A message indicating whether the function is enabled or not is output at queue manager start up and also in the error logs



Vision: IBM MQ is the **cloud native** choice for enterprise messaging

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

...continually evolving

Thank you