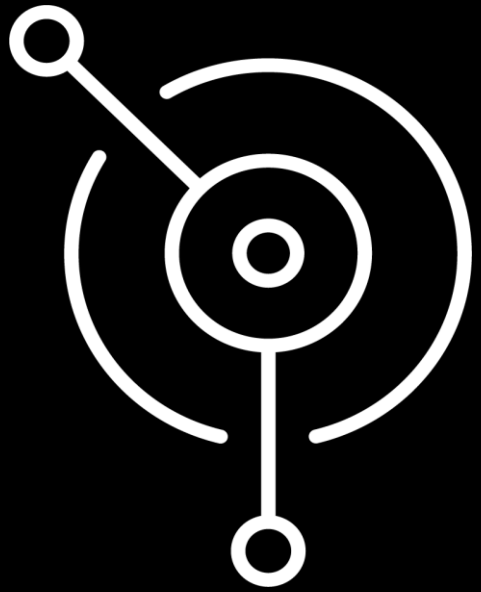
Messaging with IBM MQ

IBM MQ 9.2.3 CD

July 2021

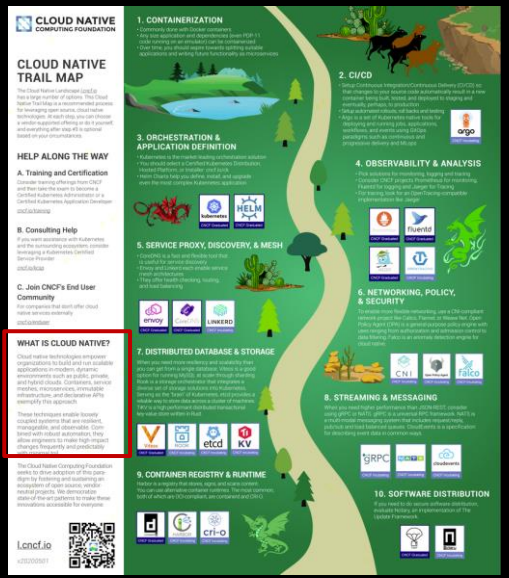
David Ware

IBM MQ Chief Architect



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

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



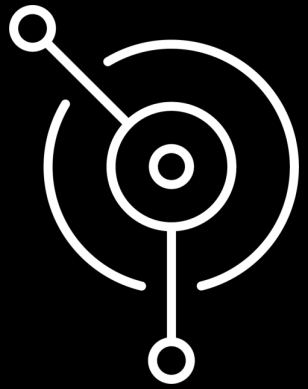
github.com/cncf/landscape#trail-map

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**

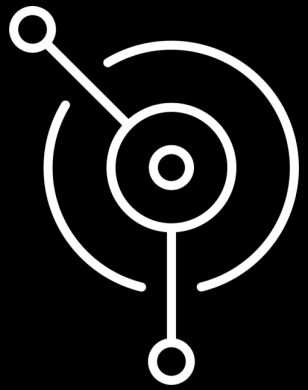
RED HAT® OPENSHIFT

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



Availability and scalability

A messaging and event service

Purpose

Loosely couple applications

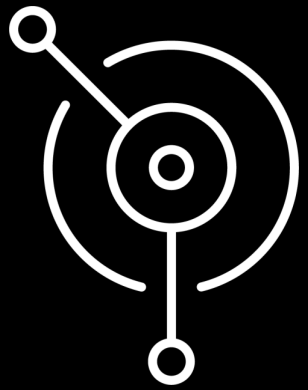
Shield applications from their own availability issues

Requirements

Scale with the application

Don’t lose the messages

Be more available than the applications

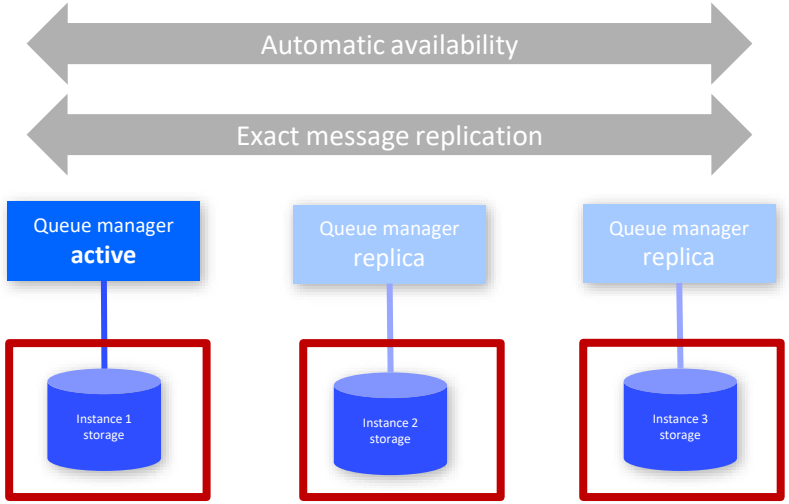


Cloud native availability

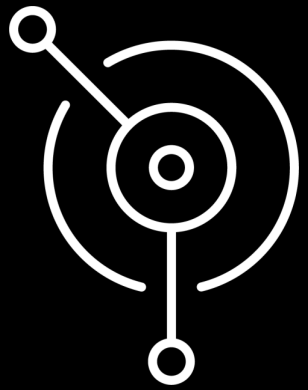
Replication and consensus

MQ Native HA

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

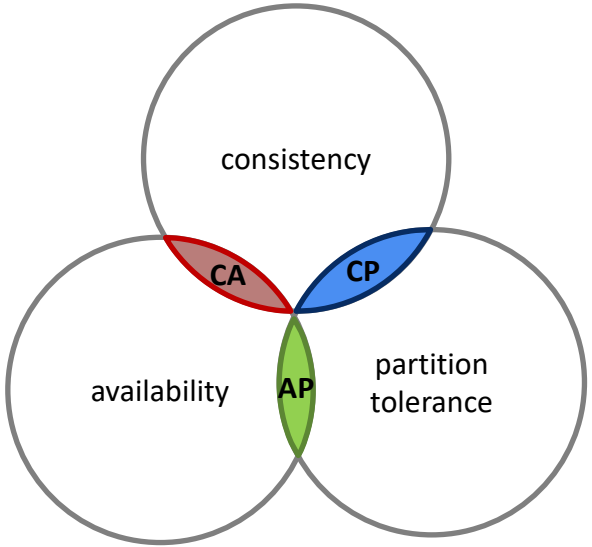


Messages persisted in three locations
Exact replicas, maintaining configuration, message order, transactional state
Quorum ensures consistency and rapid failure detection and recovery

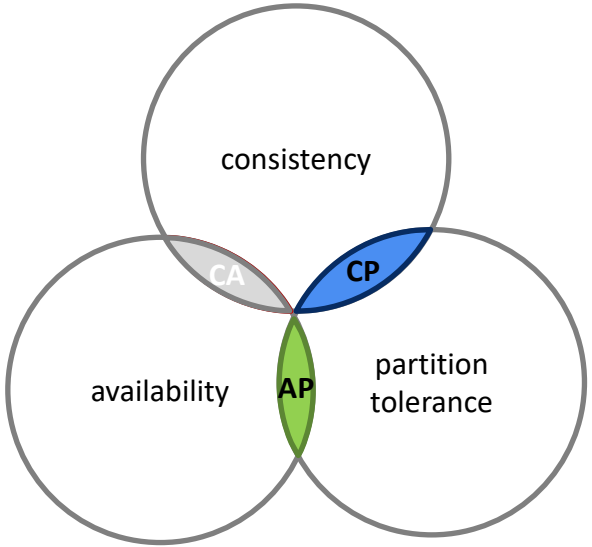


Demo

A little bit of CAP theory

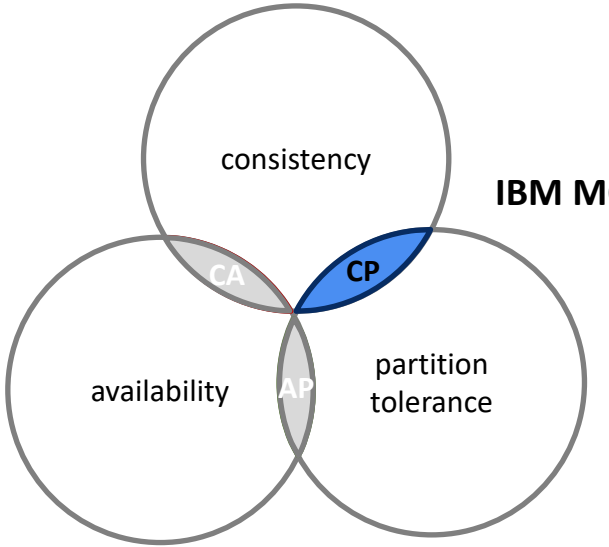


A little bit of CAP theory



In the event of failures, which will this system sacrifice?
consistency or availability?

A little bit of CAP theory



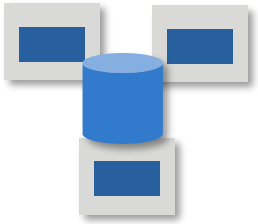
IBM MQ messages

In the event of failures, which will this system sacrifice?
consistency or availability?

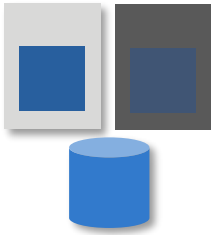
When choosing consistency, it's about ***maximizing*** availability
as much as possible, it's never an RTO of zero

The solution as a **leader/follower** model

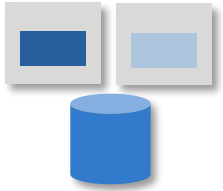
Constantly evolving to meet your availability needs



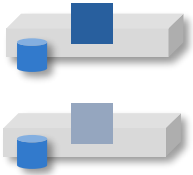
z/OS Queue Sharing Groups



System managed HA

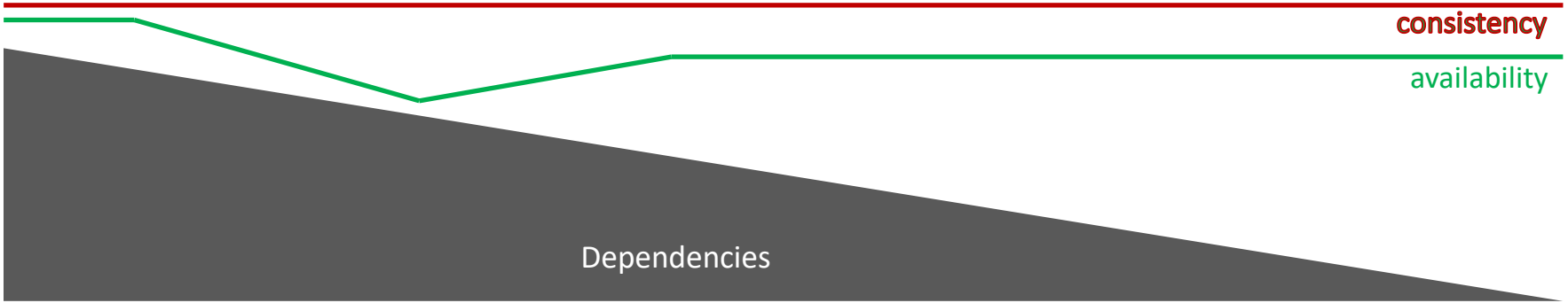


Multi-instance queue managers



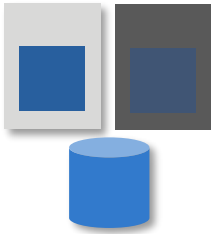
MQ Appliance

Replicated data queue manager

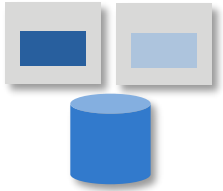


Message availability in the cloud

z/OS Queue Sharing Groups



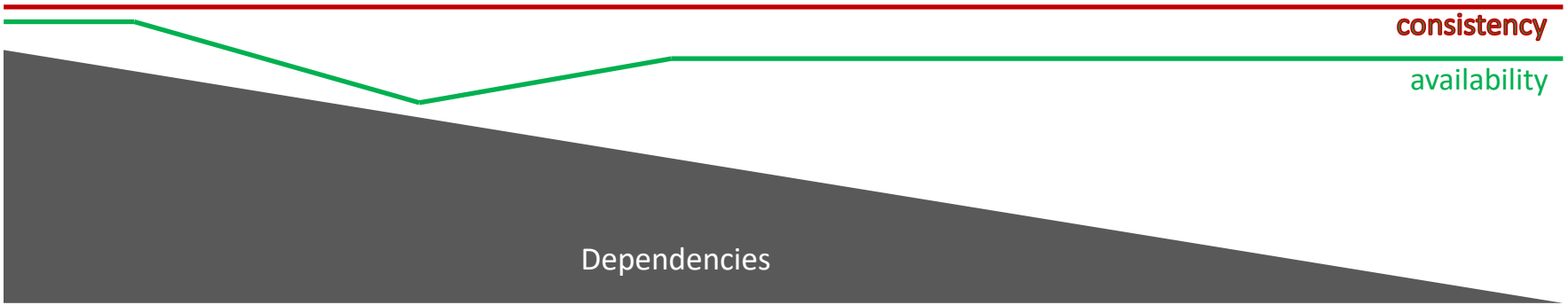
System managed HA



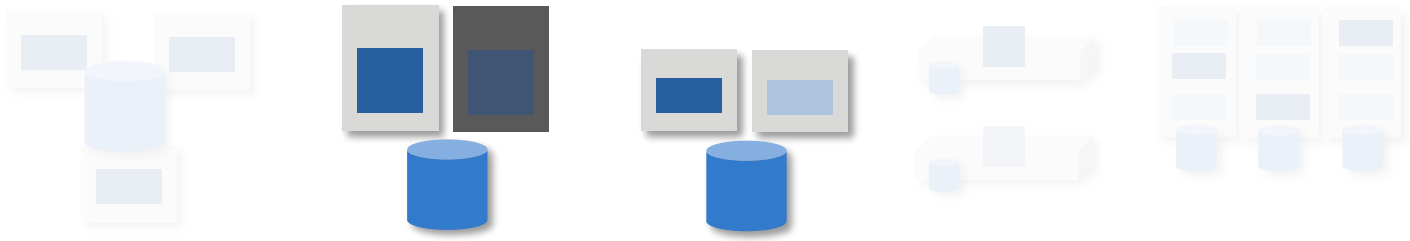
Multi-instance queue managers

MQ Appliance

Replicated data queue manager



Message availability in containers



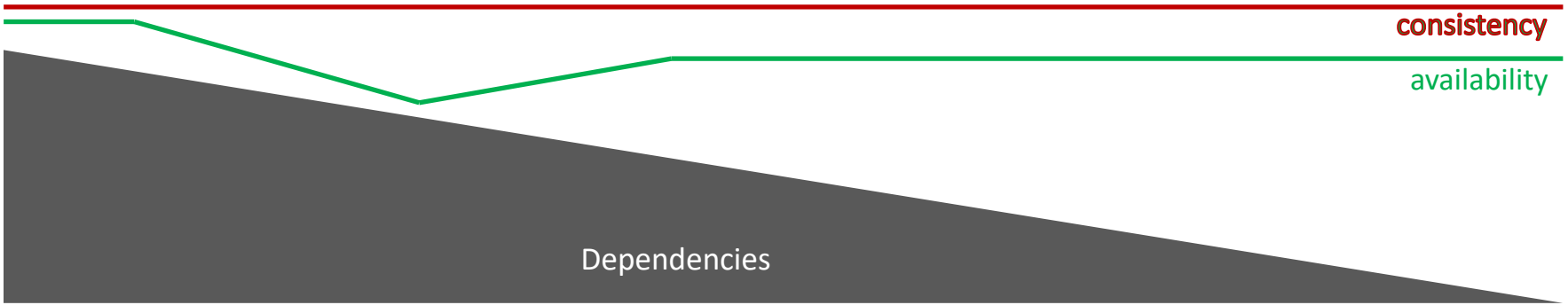
z/OS Queue Sharing Groups

System managed HA

Multi-instance queue managers

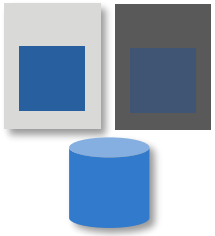
MQ Appliance

Replicated data queue manager

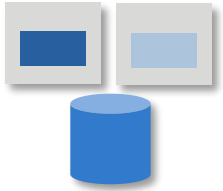


Cloud native message availability

z/OS Queue Sharing Groups



System managed HA



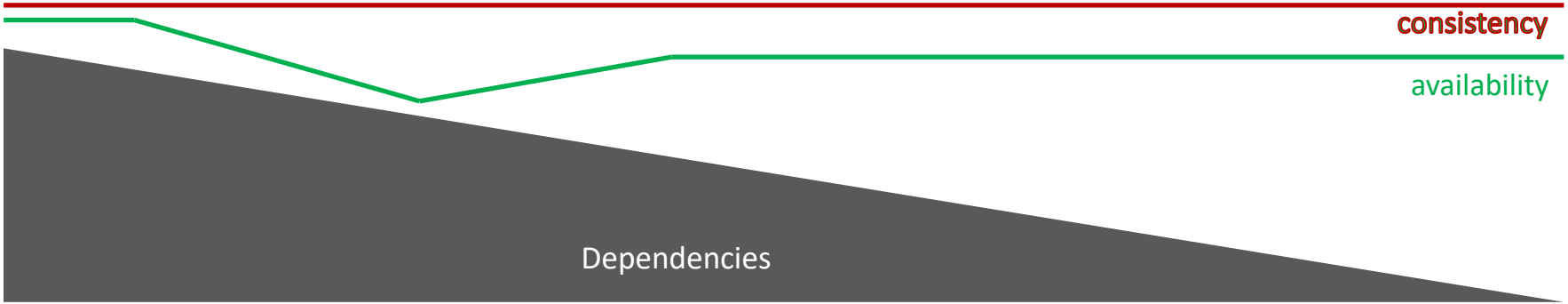
Multi-instance queue managers

MQ Appliance

Replicated data queue manager

MQ 9.2.3 CD
in OpenShift with Cloud Pak for Integration

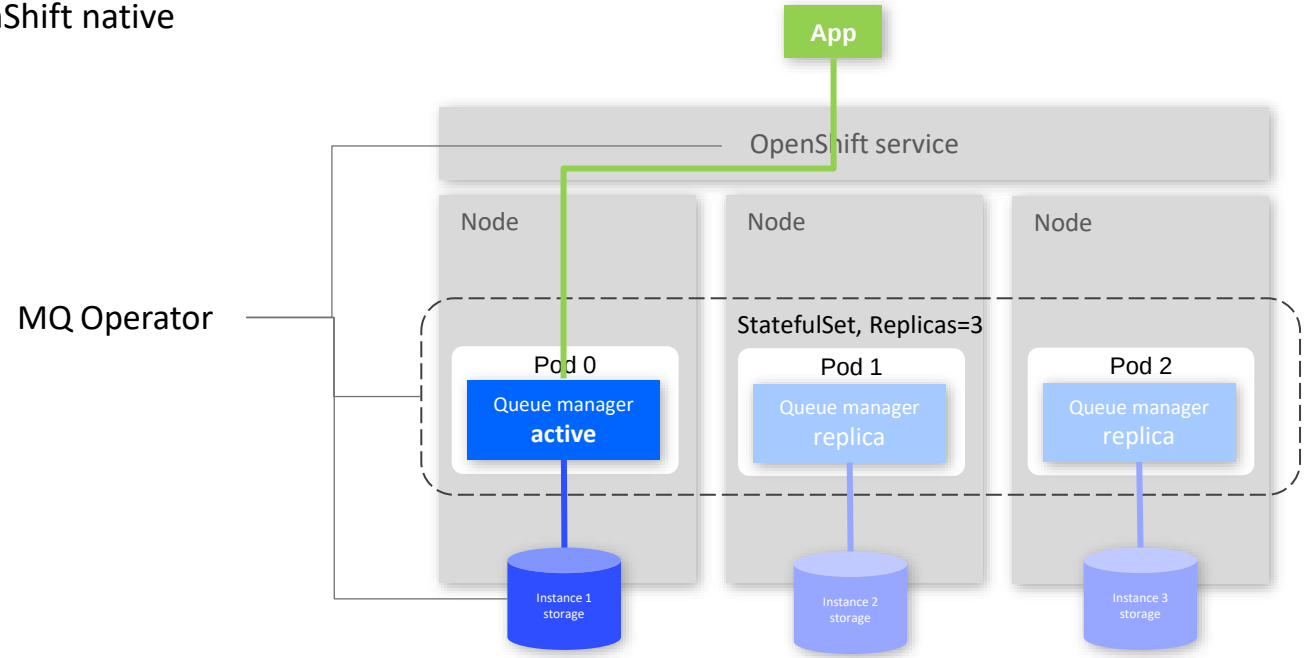
Native HA



MQ Native HA

OpenShift native

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



Availability:	cross AZ **RPO=0**, RTO “a few seconds”
Compatibility:	simple RWO block storage requirement
Cost:	Included in CP4I license (MQ Advanced ratio)
Complexity:	No external services to manage
Performance:	Network + block storage

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

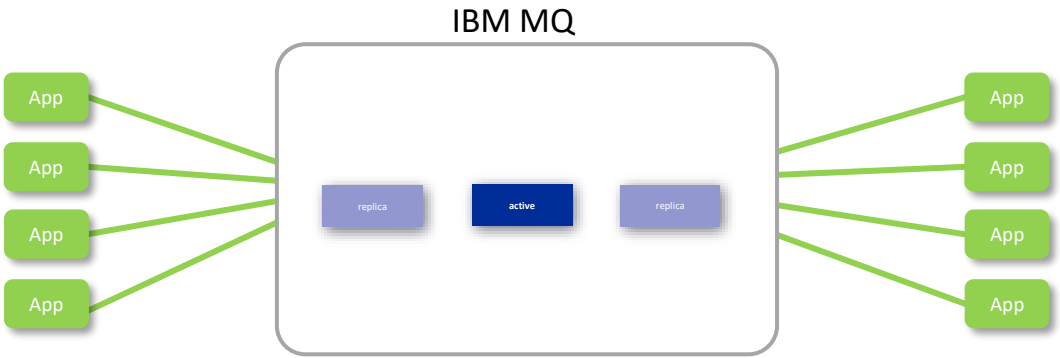
Based on the concept of a **sequential log of state changes**

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

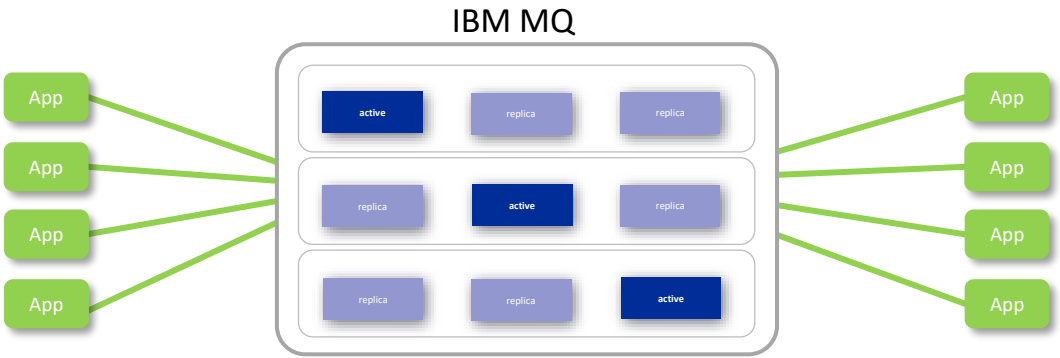
Requirements

Scale with the application

Don’t lose the messages

Be more available than the applications

A messaging and event service



Purpose

Loosely couple applications

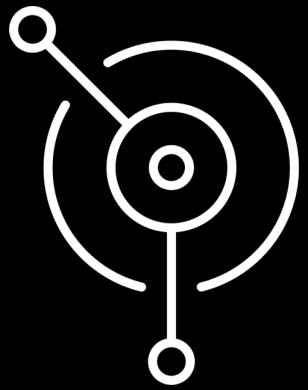
Shield applications from their own availability issues

Requirements

Scale with the application

Don’t lose the messages

Be more available than the applications



Always-on

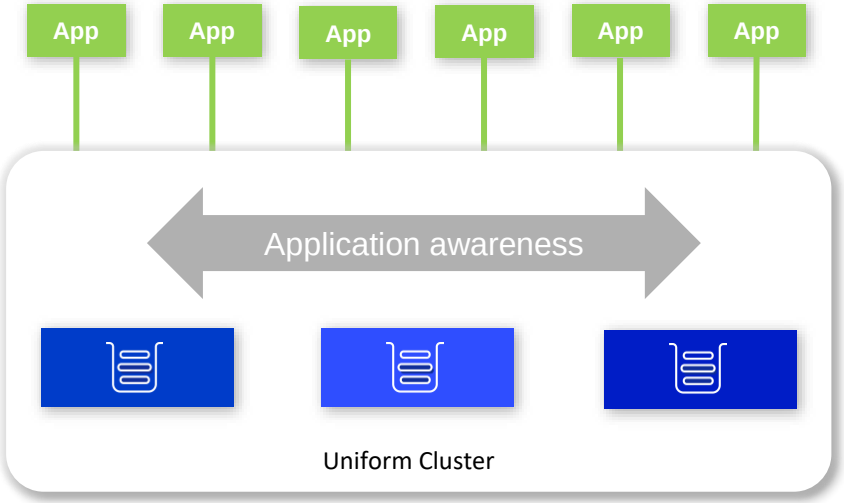
Building scalable, active-active, solutions

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

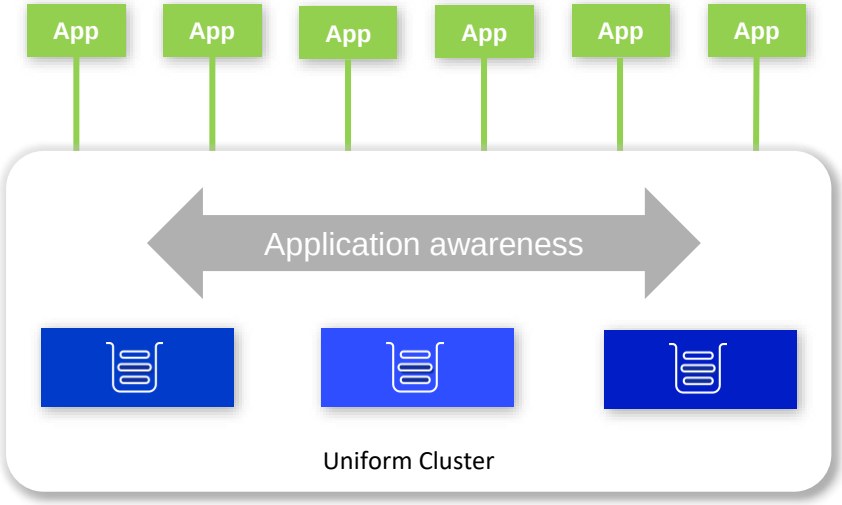


Always-on MQ

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

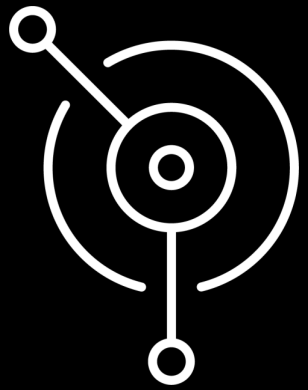
With IBM MQ 9.2 LTS, queue managers can form a **uniform cluster**, each queue manager provides the same messaging capabilities



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



Insight to your data

Stream MQ data to new applications

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.

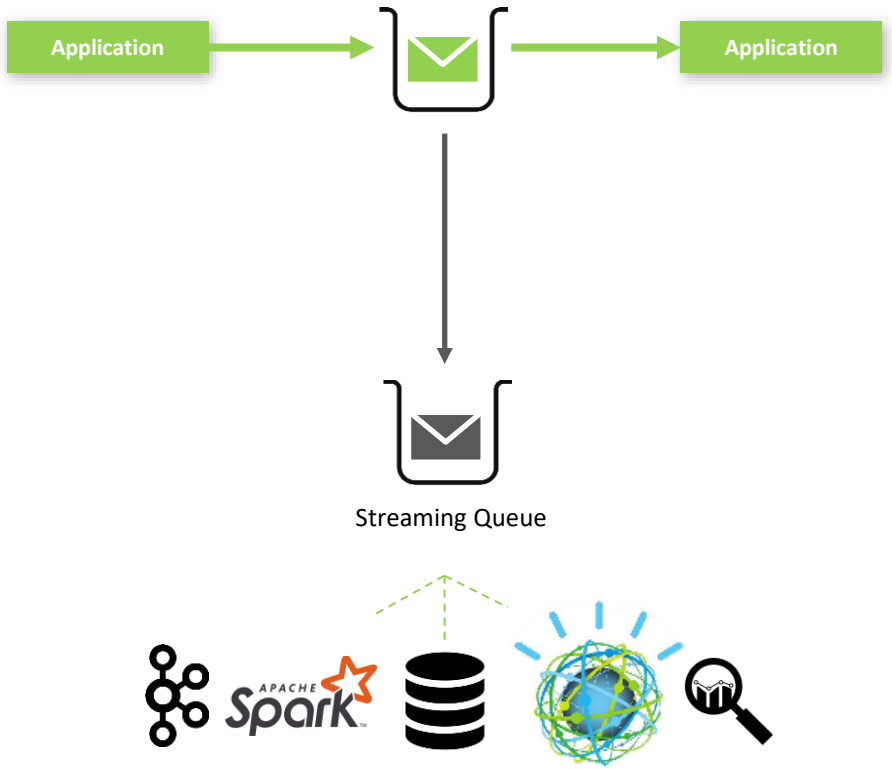
1. **Streaming Processing** to accelerate time to insight from existing data.
2. **Real world data** to accurately simulate production workloads to test the impact of architectural changes on applications.
3. **Auditing and Replay** of data in the event of disasters. Auditing and replay use cases require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

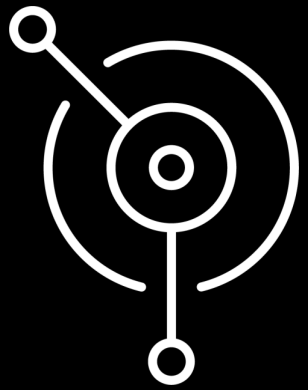
MQ Streaming Queues

New in MQ 9.2.3
Distributed platforms

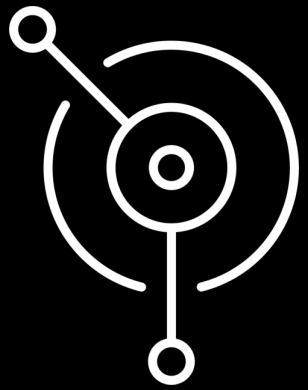
Tap into the value of existing data flowing over MQ by making message data available to Kafka, AI, and analytics applications with **zero impact to the existing applications or their messages**, and without a need for re-architecting your message flows.

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, coral IDs etc.





Demo

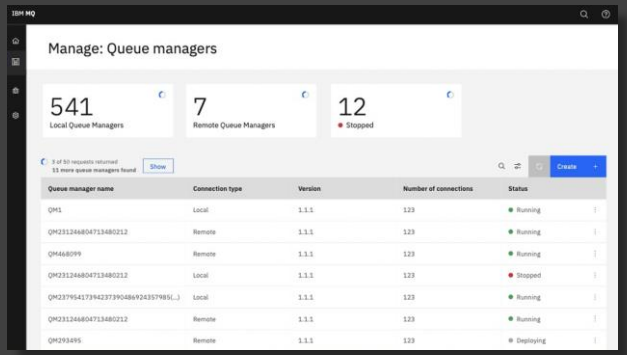


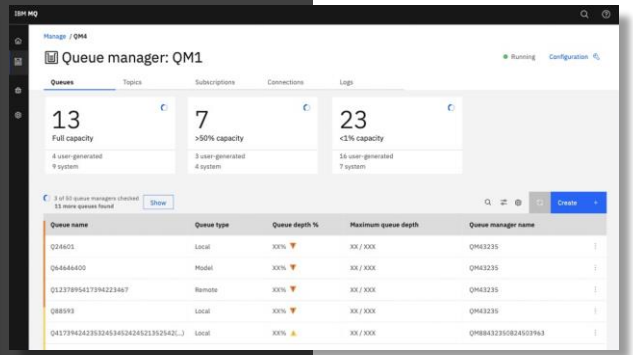
Managing MQ

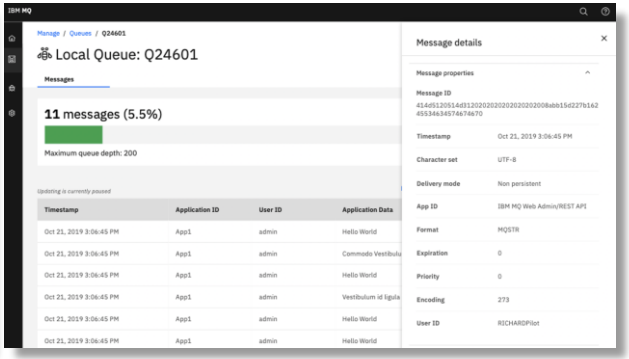
New Web Console

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







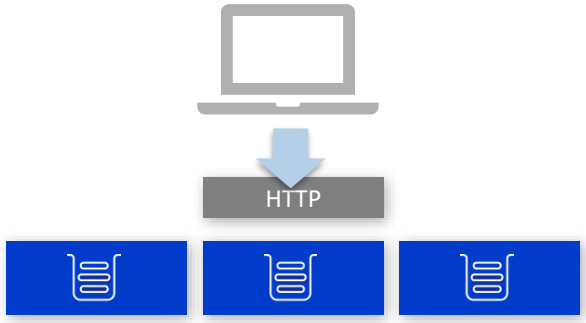
https://community.ibm.com/community/user/imwuc/blogs/callum-jackson1/2020/04/09/enhanced-web-console-in-ibm-mq-915

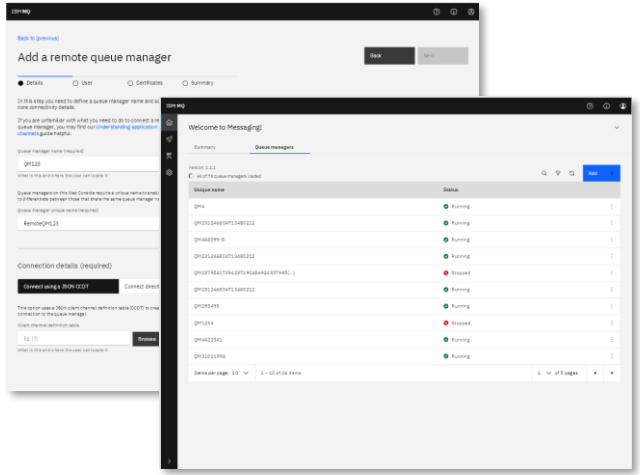
Central Web Console

New in MQ 9.2.3
All installable platforms

Originally, the web server component of MQ that underpins the web console was collocated 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.





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. Com-bined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

…continually evolving

Thank you

David Ware
IBM MQ Chief Architect

dware@uk.ibm.com