What’s New in IBM MQ?

Lyn Elkins
elkinsc@us.ibm.com

IBM Washington Systems Center

IBM MQ 9.3 LTS, enhancements since 9.2 LTS

Simplified Linux install	Dspmqinst for IBM i	Stream MQ Appliance error logs	MQ Console application quick start	Key repository passwords	Idempotent MQSC DELETE commands	Hardware accelerated compression for AIX	Non-OS user authorisations	TLS 1.3 across all protocols	TLS 1.3 support for MQIPT
Encrypted MQTT channel passphrases	TLS-only communication switch	Streaming queues	Multiple queue manager certificates for MQIPT	TLS enabled .NET XA monitor	Cryptographic hardware support for client passwords	PKCS#12 key repository support	SNI hostname support for channel routing	Uniform Cluster support for request/reply flows	Transaction boundary aware Uniform Cluster
Uniform Cluster aware MDBs	IBM MQ scaler for KEDA	Apache Qpid JMS support over AMQP	Point-to-point support for AMQP	Java 17 support for applications	MQ Console remote queue manager support	Jakarta Messaging 3.0 support	64-bit RBA default for z/OS	MQ Appliance synchronous DR replication	Disk encryption for the MQ Appliance
OpenShift Operator managed rolling upgrade	OpenShift support for zLinux and Power	OpenShift Prometheus integration with ServiceMonitor	Helm chart sample for Kubernetes deployments	Client attached dead-letter handler	MQ Appliance failed resource action control	IBM MQ on Cloud LogDNA integration	Raft based Native HA for OpenShift	Transfer logging for Managed File Transfer	AT-TLS support for z/OS
Separate statistics and accounting intervals for z/OS	Browse support for AMQP applications	Start/stop of MFT resource monitors	Redistributable MFT Logger	MFT managed call control over REST	IBM MQ AsyncAPI binding	AsyncAPI code generator for IBM MQ JMS applications	Queue depth SMF data for z/OS	.NET 6 application support	Extended REST API message properties

www.ibm.com/docs/en/ibm-mq/9.3?topic=930-whats-new-in-mq

What’s new in 9.3.1?

IBM MQ 9.3.1 CD enhancements

Linux
Windows

.NET 6 libraries

Multi ★

CAPEXPRY

Linux

RDQM on RHEL 9

Multi

MQIPT MQCSP Password Protection

z/OS ★

64-bit Channel Initiator

z/OS ★

SMF queue statistics enhancements

z/OS

Streaming queues on shared queues (backported to LTS)

Multi ★

Console observability

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=931-whats-new-in-mq

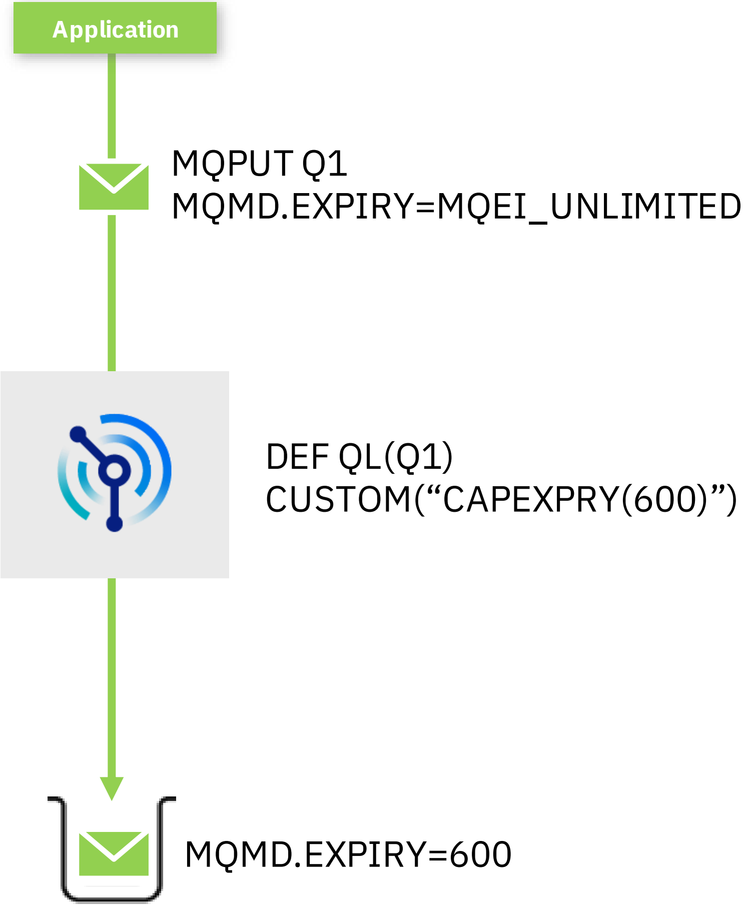
CAPEXPRY

CAPEXPRY provides an administrative way to set the maximum expiry value that a message can have when sent to a queue / published to a topic

Useful if you have a “badly behaved” application which sends messages without setting an expiry value and you can’t easily change it

Originally provided in the MQ 8 time frame (via an APAR) as a custom property

Custom property approach was always intended to be a temporary solution



MQPUT Q1
MQMD.EXPIRY=MQEI_UNLIMITED

DEF QL(Q1)
CUSTOM(“CAPEXPRY(600)”)

MQMD.EXPIRY=600

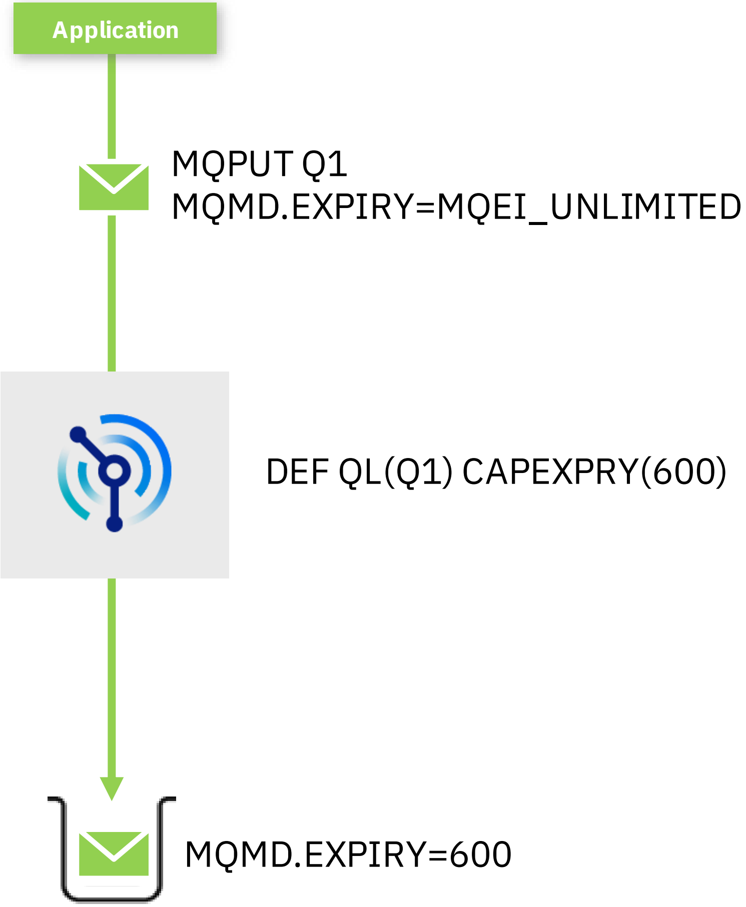
CAPEXPRY

In 9.3.1, on distributed platforms, CAPEXPRY now becomes a first class attribute

z/OS will add support later

For any given object you can either use the custom CAPEXPRY or the first class CAPEXPRY attribute, but you can’t use both at once

Works the same as before, but is now cluster aware



Application

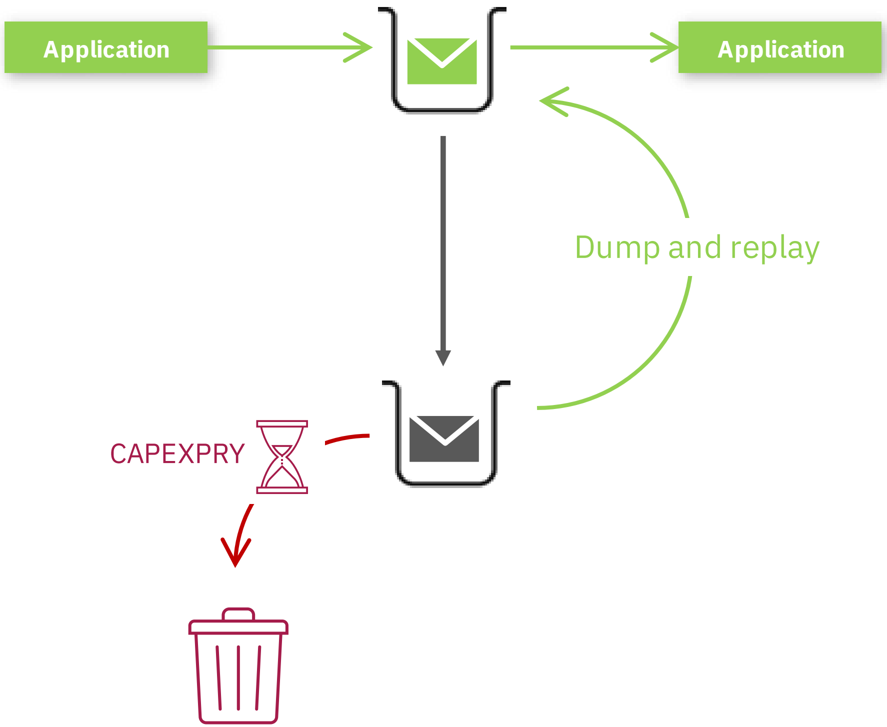
MQPUT Q1
MQMD.EXPIRY=MQEI_UNLIMITED

DEF QL(Q1) CAPEXPRY(600)

MQMD.EXPIRY=600

And why are we doing this?

Streaming queues!



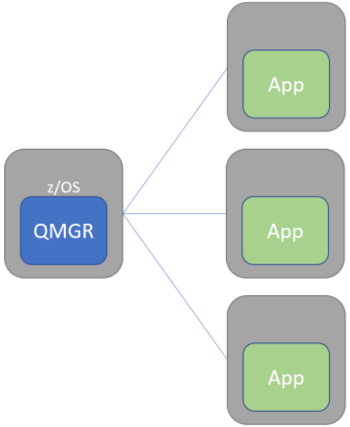
64-bit channel initiator

Previously server-connection channel messages were staged in the channel initiator in a set of buffers in 31 bit storage

Large messages could limit the maximum number of active channels

Now with 64 bit storage in the channel initiator, a larger number of applications can connect at the same time

Samples will be changed to have MEMLIMIT=2G (2GB 64 storage as a starting point)



104 concurrent clients sending 10MB messages.
Memory footprint per client

	9.3.0	9.3.x
31 bit	10.4MB	112KB
64 bit	0	11.6MB

Assuming 1.3GB spare space below the bar this implies max 128 clients at 9.3.0.
At 9.3.1 we could get to 9999 clients (the max) assuming suitable MEMLIMIT

More flexible monitoring

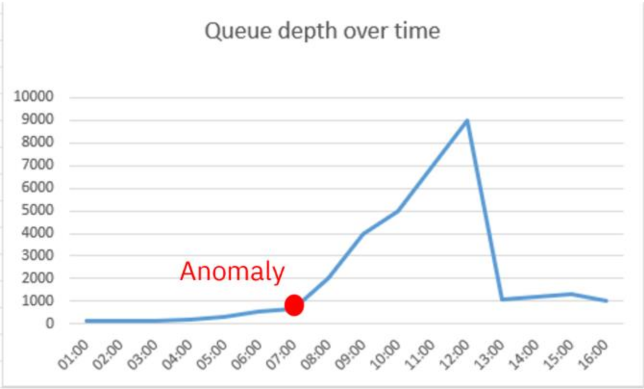
9.3.0 added a new set of per queue statistics

Enabled by setting STATQ(ON) on individual queues or by setting STATQ(ON) on the queue manager and STATQ(QMGR) on the queues

No detectable performance impact when enabled, even when writing records out every second

9.3.1 added
DISPLAY QSTATUS information to these

NB: data is output regardless of the MONQ attribute on the queue



Sample MP1B output

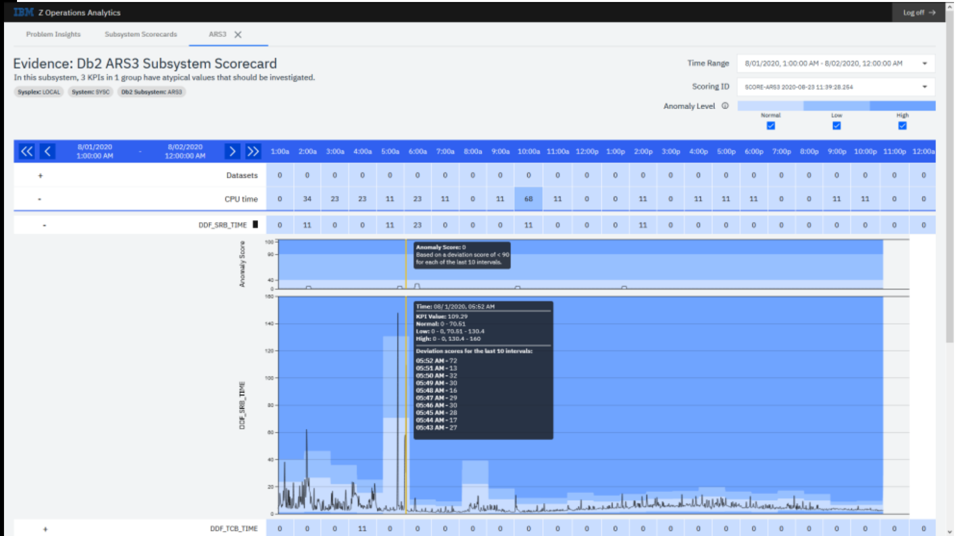
```
MV41,MQ27,2022/06/24,11:27:29,VRM:931
Queue Name..............Q1
Disposition.................Private
Pageset ID..................4
Bufferpool ID..............2
Current Depth.............10
Open Output Count.....2
Open Input Count........1
QTIME Short.................5029503
QTIME Long..................5029503
Last Put Time................2022/06/24,11:17:33.902190
Last Get Time................2022/06/24,11:18:33.902190
Uncommitted Changes.No
Oldest Message Age.....64
```

IBM z Anomaly Analytics now supports MQ
Proactively detect problems

Machine learning and AI to identify anomalous behavior

- Leverage historical data to build a model representative of normal operations
- Real time scoring of subsystem KPIs against the model helps detect operational anomalies through analytics and trend analysis
- Anomalies may point to broader issues impacting the environment
- Generate events to alert operations of when anomalous behavior has been detected
- Events can be correlated with data from your enterprise providing a hybrid cloud view of your entire enterprise

Supports MQ SMF 115 (statistics data)



Evidence: Db2 ARS3 Subsystem Scorecard

In this subsystem, 3 KPIs in 1 group have atypical values that should be investigated.

Sysplex: LOCAL
System: SYSC
Db2 Subsystem: ARS3

Time Range: 8/01/2020, 1:00:00 AM - 8/02/2020, 12:00:00 AM

Scoring ID: SCORE-ARS3 2020-08-23 11:39:28.264

Anomaly Level: Normal ☑ Low ☑ High ☑

8/01/2020 1:00:00 AM - 8/02/2020 12:00:00 AM	1:00a	2:00a	3:00a	4:00a	5:00a	6:00a	7:00a	8:00a	9:00a	10:00a	11:00a	12:00p	1:00p	2:00p	3:00p	4:00p	5:00p	6:00p	7:00p	8:00p	9:00p	10:00p	11:00p	12:00a
+ Datasets	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
- CPU time	0	34	23	23	11	23	11	0	11	68	11	0	0	11	0	11	11	11	0	0	11	11	0	0
- DDF_SRB_TIME	0	11	0	0	11	23	0	0	0	11	0	0	0	11	0	0	0	0	0	0	0	0	0	0
+ DDF_TCB_TIME	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Anomaly Score: 0
Based on a deviation score of < 90 for each of the last 10 intervals.

Time: 08/ 1/2020, 05:52 AM
KPI Value: 109.29
Normal: 0 - 70.51
Low: 0 - 0, 70.51 - 130.4
High: 0 - 0, 130.4 - 160
Deviation scores for the last 10 intervals:
05:52 AM - 72
05:51 AM - 13
05:50 AM - 32
05:49 AM - 30
05:48 AM - 16
05:47 AM - 29
05:46 AM - 30
05:45 AM - 28
05:44 AM - 17
05:43 AM - 27

IBM Z Operational Log and Data Analytics

Advanced Data Streaming

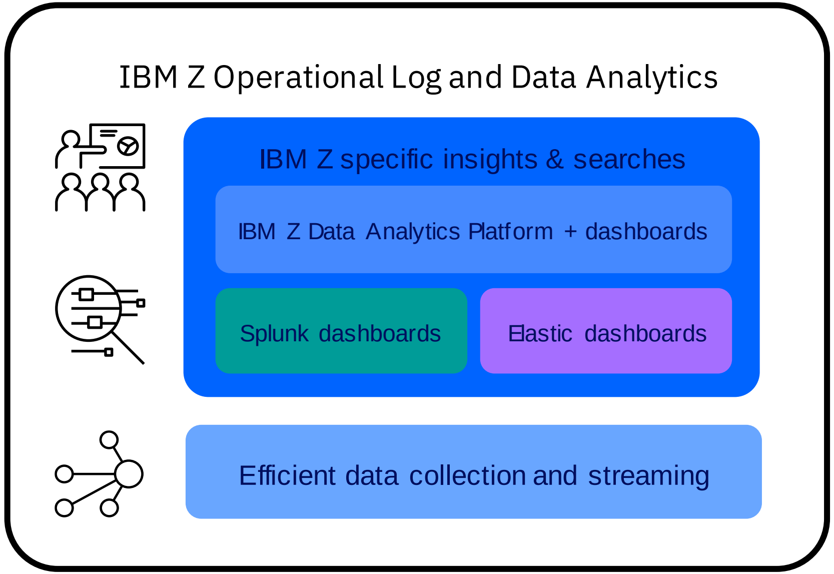
Gain near real-time access to IBM Z operational data where you want to run analytics

Dashboards and searches

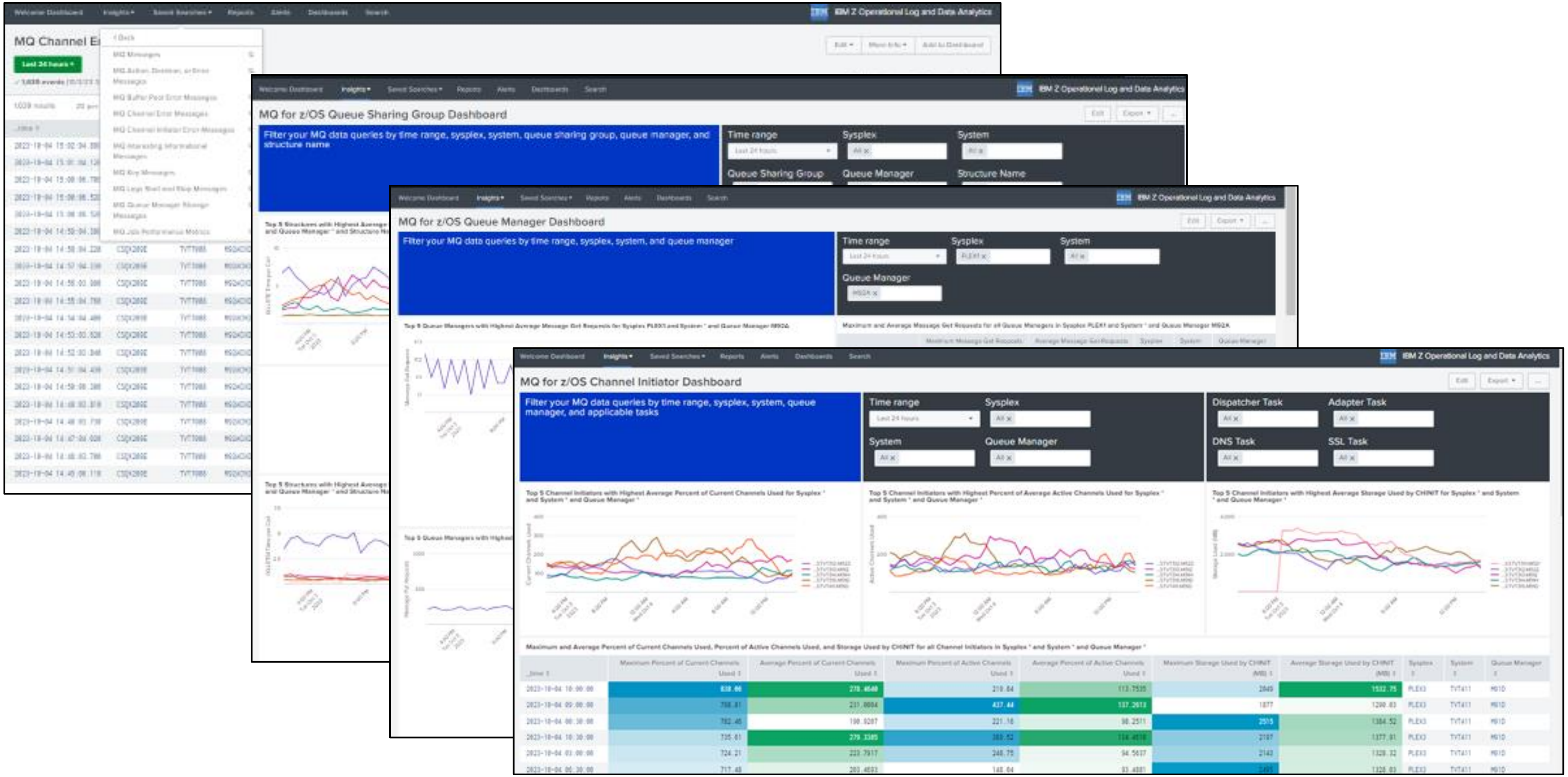
Leverage powerful dashboards to extract contextualized insights from your IBM Z operational data. Prebuilt searches save valuable time and resources when doing deeper analysis

IBM Z Data Analytics Platform

Determine the cause of incidents on platform with the included log analysis platform based on an industry leading open-source solution


IBM Z Operational Log and Data Analytics
IBM Z specific insights & searches
IBM Z Data Analytics Platform + dashboards
Splunk dashboards
Elastic dashboards
Efficient data collection and streaming

MQ Dashboards and Log Searches


Welcome Dashboard
Insights
Saved Searches
Reports
Alerts
Dashboards
Search
IBM Z Operational Log and Data Analytics
MQ Channel Er
Last 24 hours
< Back
MQ Messages
MQ Action, Decision, or Error Messages
MQ Buffer Pool Error Messages
MQ Channel Error Messages
MQ Channel Initiator Error Messages
MQ Interesting Informational Messages
MQ Key Messages
MQ Logs Start and Stop Messages
MQ Queue Manager Storage Messages
MQ Job Performance Metrics
Edit
Export
Time range
Sysplex
System
Queue Sharing Group
Queue Manager
Structure Name
Dispatcher Task
Adapter Task
DNS Task
SSL Task
All

MQ for z/OS Queue Sharing Group Dashboard

Filter your MQ data queries by time range, sysplex, system, queue sharing group, queue manager, and structure name

MQ for z/OS Queue Manager Dashboard

Filter your MQ data queries by time range, sysplex, system, and queue manager

MQ for z/OS Channel Initiator Dashboard

Filter your MQ data queries by time range, sysplex, system, queue manager, and applicable tasks

Top 5 Channel Initiators with Highest Average Percent of Current Channels Used for Sysplex * and System * and Queue Manager *

Top 5 Channel Initiators with Highest Percent of Average Active Channels Used for Sysplex * and System * and Queue Manager *

Top 5 Channel Initiators with Highest Average Storage Used by CHINIT for Sysplex * and System * and Queue Manager *

Maximum and Average Percent of Current Channels Used, Percent of Active Channels Used, and Storage Used by CHINIT for all Channel Initiators in Sysplex * and System * and Queue Manager *

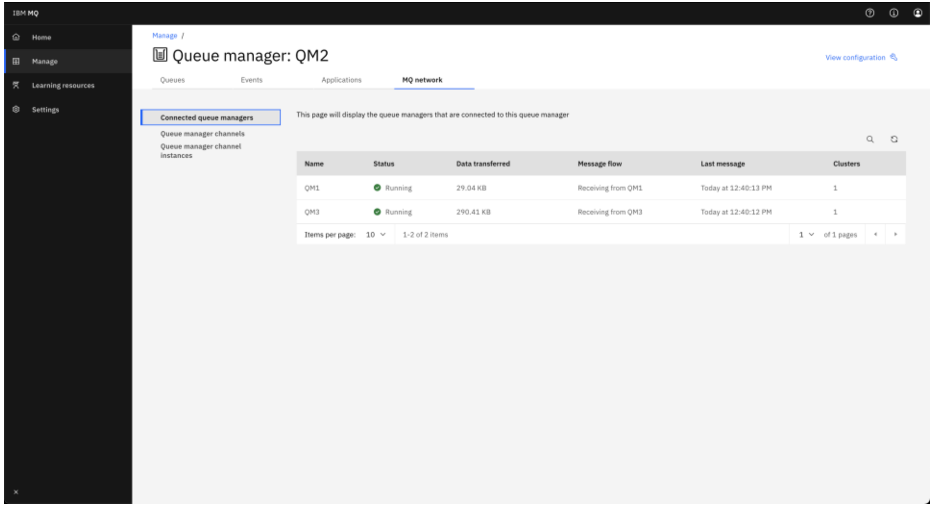
_time	Maximum Percent of Current Channels Used	Average Percent of Current Channels Used	Maximum Percent of Active Channels Used	Average Percent of Active Channels Used	Maximum Storage Used by CHINIT (MB)	Average Storage Used by CHINIT (MB)	Sysplex	System	Queue Manager
2023-10-04 10:00:00	838.06	278.4640	219.84	113.7535	2049	1532.75	PLEX3	TVT411	MQ1D
2023-10-04 09:00:00	798.81	231.0004	437.44	137.2613	1877	1200.83	PLEX3	TVT411	MQ1D
2023-10-04 08:30:00	782.46	198.9287	221.16	98.2511	2515	1384.52	PLEX3	TVT411	MQ1D
2023-10-04 10:30:00	735.61	279.3385	388.52	114.4618	2187	1377.81	PLEX3	TVT411	MQ1D
2023-10-04 03:00:00	724.21	223.7917	246.75	94.5637	2143	1328.32	PLEX3	TVT411	MQ1D
2023-10-04 06:30:00	717.48	203.4693	148.64	93.4881	2495	1328.83	PLEX3	TVT411	MQ1D

MQ Console observability

The MQ Console now makes it easier to see what’s happening on a queue manager.

See which applications are active, over which channel they’re connected, and what they’ve been doing.

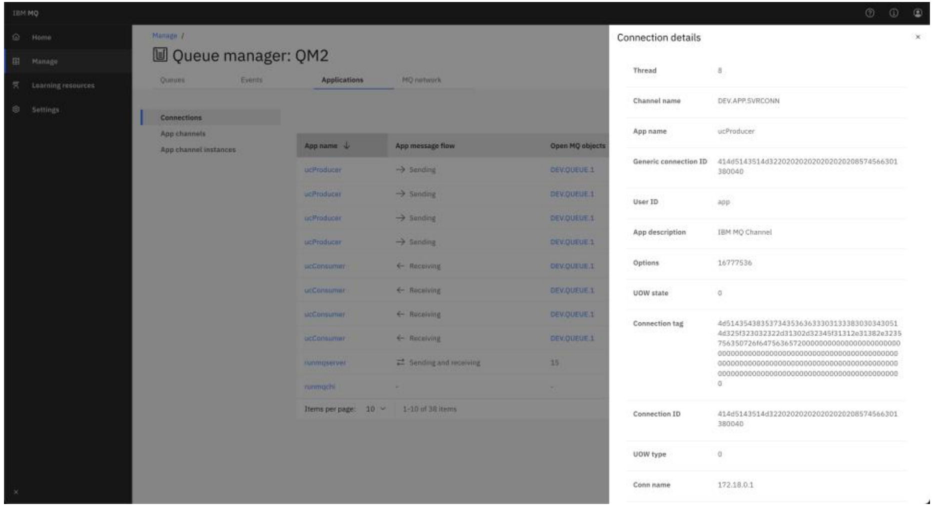
Easily understand how a queue manager is interacting with other queue managers in its network.



This page will display the queue managers that are connected to this queue manager

Name	Status	Data transferred	Message flow	Last message	Clusters
QM1	● Running	29.04 KB	Receiving from QM1	Today at 12:40:13 PM	1
QM3	● Running	290.41 KB	Receiving from QM3	Today at 12:40:12 PM	1

Items per page: 10 1-2 of 2 items 1 of 1 pages



App name	App message flow	Open MQ objects
ucProducer	→ Sending	DEV.QUEUE.1
ucProducer	→ Sending	DEV.QUEUE.1
ucProducer	→ Sending	DEV.QUEUE.1
ucProducer	→ Sending	DEV.QUEUE.1
ucConsumer	← Receiving	DEV.QUEUE.1
ucConsumer	← Receiving	DEV.QUEUE.1
ucConsumer	← Receiving	DEV.QUEUE.1
ucConsumer	← Receiving	DEV.QUEUE.1
runmqserver	⇄ Sending and receiving	15
runmqchi	-	-

Items per page: 10 1-10 of 38 items

Connection details

Thread	8
Channel name	DEV.APP.SVRCONN
App name	ucProducer
Generic connection ID	414d5143514d32202020202020202020208574566301 380040
User ID	app
App description	IBM MQ Channel
Options	16777536
UOW state	0
Connection tag	4d514354383537343536363330313338303034305l 4d325f323032322d31302d32345f31312e31382e3235 756350726f647563657200000000000000000000000 0000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000 0
Connection ID	414d5143514d32202020202020202020208574566301 380040
UOW type	0
Conn name	172.18.0.1

What’s new in 9.3.2?

IBM MQ 9.3.2 CD enhancements

- MQ Console application view
- Multi — More information in DIS QMSTATUS
- MQ Adv. — More information in DIS QMSTATUS for Native HA ★
- Windows Linux — TLS 1.3 support for managed .NET clients
- Java modular application support ★
- Simplified messaging REST API security
- MQ Adv. — Native HA entitlement with advanced in addition to CP4I
- z/OS — SMF116 STREAMEDN for streaming queues
- Appliance — DR between 2 HA pairs ★

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=932-whats-new-in-mq

More Native HA info in DIS QMSTATUS

For Native HA, DIS QMSTATUS has been enhanced to make it easy to discover where the active instance of the queue manager is running, and the status of the replicas (similar to dspmq –o nativeha)

```
DISPLAY QMSTATUS TYPE(NATIVEHA)
     3 : DISPLAY QMSTATUS TYPE(NATIVEHA)
AMQ8705I: Display Queue Manager Status Details.

   INSTANCE(nha_qm1.0)                TYPE(NATIVEHA)
   ROLE(ACTIVE)
   HAINITDA(2022-06-08)                HAINITL(0:0:13:54372)
   HAINITTI(16.06.54)
   REPLADDR(172.30.195.172(8844))

   INSTANCE(nha_qm1.1)                  TYPE(NATIVEHA)
   ROLE(REPLICA)
   BACKLOG(0)                           CONNACTV(YES)
   INSYNC(YES)
   REPLADDR(172.30.195.172(8845))

   INSTANCE(nha_qm1.2)                TYPE(NATIVEHA)
   ROLE(REPLICA)
   BACKLOG(661)                       CONNACTV(YES)
   INSYNC(NO)
   REPLADDR(172.30.195.172(8846))
```

MQ Ansible samples

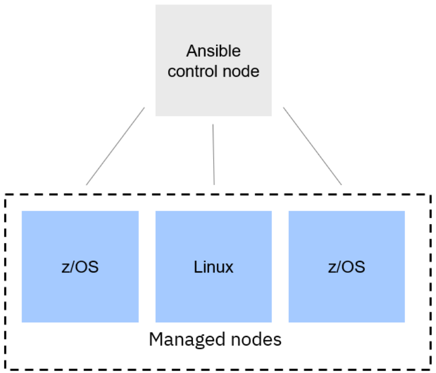
Ansible provides a simple way to automate configuration and management of applications and infrastructure

Ansible playbooks can be run against z/OS providing a common approach across all your IT infrastructure

The IBM z/OS core collection makes it easy for Ansible to interact with data sets, submit jobs, run commands, etc

Sample MQ for z/OS playbooks are now available for creating, altering, displaying and deleting queues. These playbooks can be easily adjusted to work with the other MQ object

An alternative set of playbooks are also available for configuring MQ using simple YAML templates. These can be used on all platforms, via the MQ REST API



https://github.com/IBM/z_ansible_collections_samples/tree/master/zos_subsystems/mq

```
zos_operator:
      cmd: "{{ mq_cpf }} DEF QL ( {{ mq_queue_name }})  DESCR(‘’A QUEUE")"
```

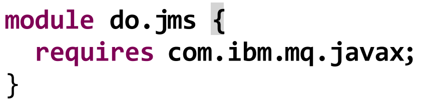
https://github.com/ibm-messaging/mq-ansible-yaml-commands

```
# Start a TCP listener.
 - command: start
   parameters:
     port: 1407
   qualifier: listener
# Start a channel.
 - command: start
   name: MQ07.TO.MQ08
   qualifier: channel
```

Support for Java Modules

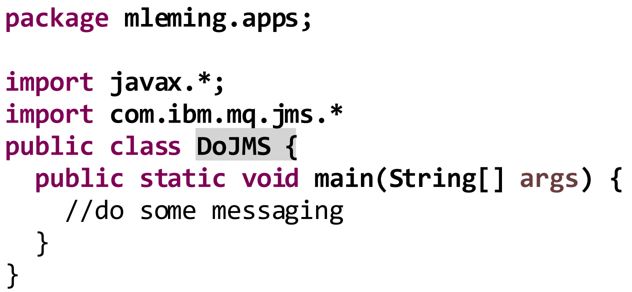
MQ classes for JMS / Jakarta enhanced to allow them to be used properly with modular applications (introduced in Java 9)

1) Import the MQ modules into your modular app



```
module do.jms {
  requires com.ibm.mq.javax;
}
```

2) Use the MQ modules in your app



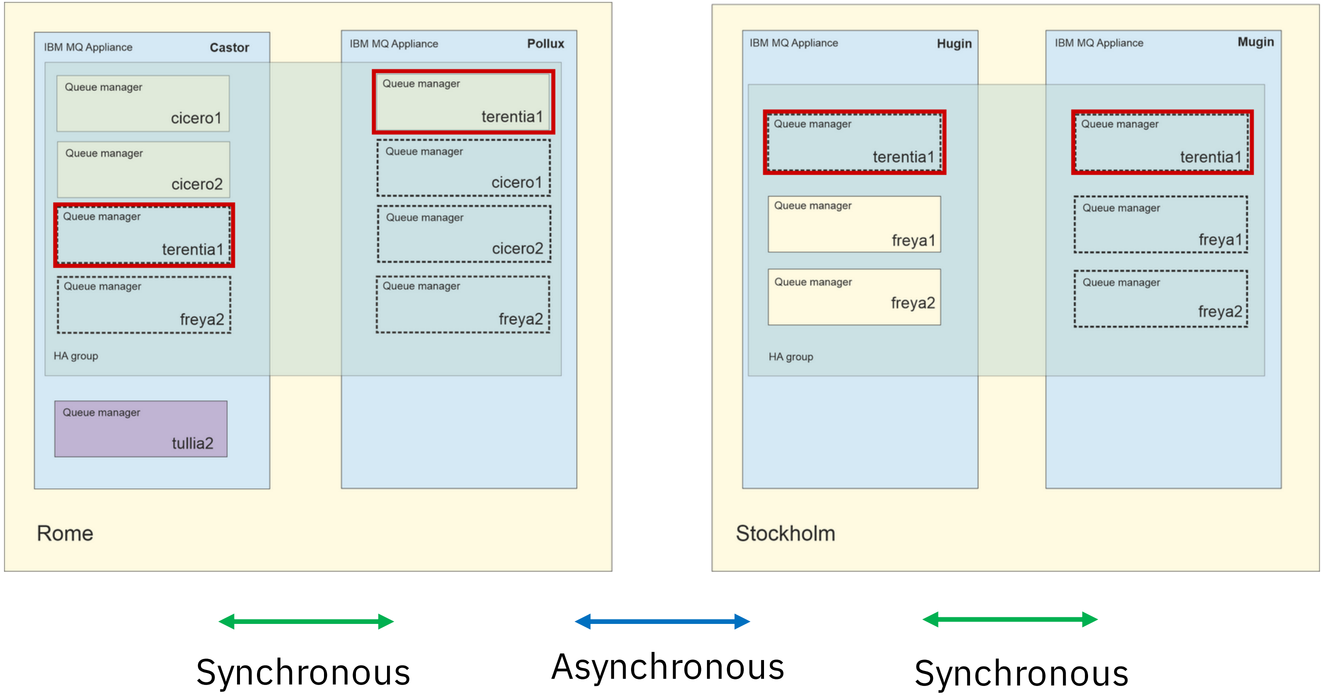
```
package mleming.apps;

import javax.*;
import com.ibm.mq.jms.*
public class DoJMS {
  public static void main(String[] args) {
    //do some messaging
  }
}
```

3) Run app with MQ modules

```
java --module-path MQ_HOME/java/lib/modules/javax
    -m do.jms/mleming.apps.DoJMS
```

DR between two HA pairs on MQ Appliance

The MQ appliance now supports DR between two HA pairs, meaning that HA is automatically available after a DR failover, removing the manual steps that are currently required to setup an HA group after a DR failover



What’s new in 9.3.3?

IBM MQ 9.3.3 CD enhancements

MQ Console better linking, and objects associated with queues

Multi

AMQP performance improvements

Remote REST messaging

MQ Adv.

MQ Adv. VUE

Supported MQ/Kafka Connectors

MQ Adv.

Native HA damaged object auto recovery

z/OS

More per queue SMF data

Appliance

CLI SSH authentication using certificate

Appliance

Disk space monitoring

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=933-whats-new-in-mq

Remote REST messaging

We are seeing more and more customers using the MQ web server

We are gradually removing the requirement that the server is collocated with the queue managers that it interacts with:

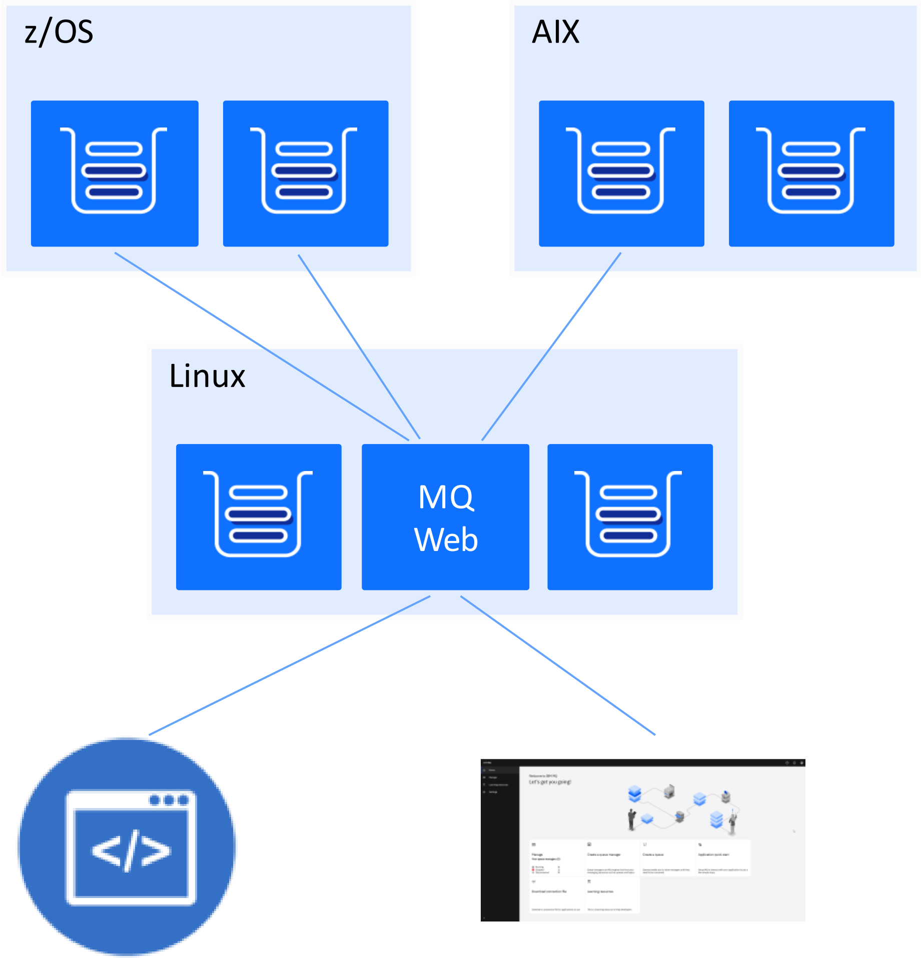
Admin REST API: REST admin gateway in 9.1.0

MQ Console: remote support in 9.3.0

Messaging REST API: remote support in 9.3.3

The last two use JSON CCDTs to connect to remote queue managers via client connections

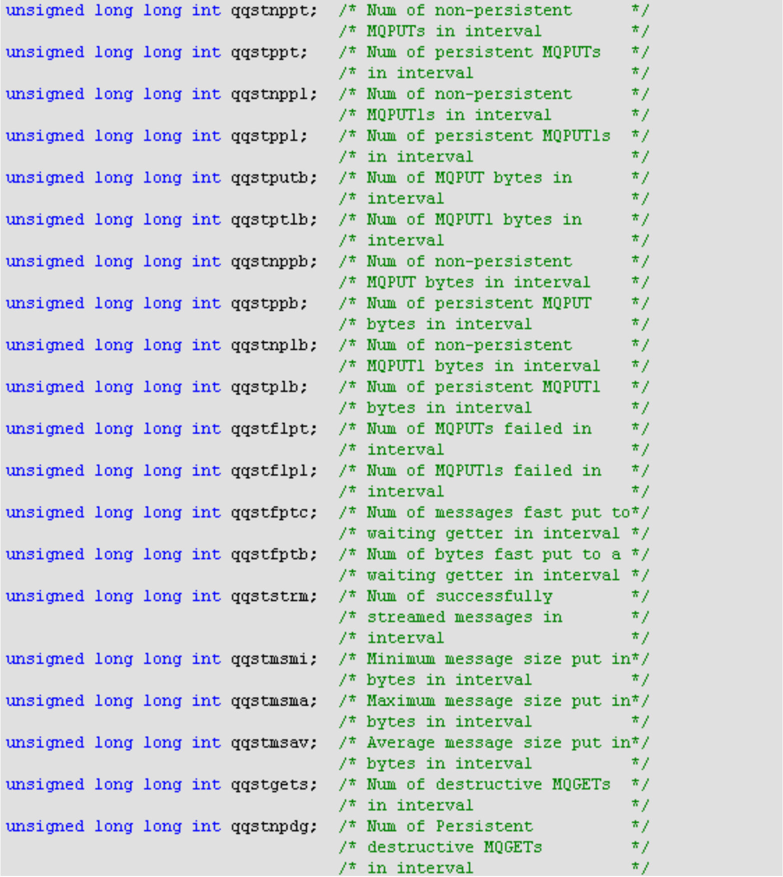
Very useful if you want to interact with MQ on z/OS but not install the web console there



More flexible monitoring

Full set of per queue statistics completed

65 data points per queue



```
unsigned long long int qqstnppt;  /* Num of non-persistent      */
                                  /* MQPUTs in interval         */
unsigned long long int qqstppt;   /* Num of persistent MQPUTs   */
                                  /* in interval                */
unsigned long long int qqstnppl;  /* Num of non-persistent      */
                                  /* MQPUT1s in interval        */
unsigned long long int qqstppl;   /* Num of persistent MQPUT1s  */
                                  /* in interval                */
unsigned long long int qqstputb;  /* Num of MQPUT bytes in      */
                                  /* interval                   */
unsigned long long int qqstptlb;  /* Num of MQPUT1 bytes in     */
                                  /* interval                   */
unsigned long long int qqstnppb;  /* Num of non-persistent      */
                                  /* MQPUT bytes in interval    */
unsigned long long int qqstppb;   /* Num of persistent MQPUT    */
                                  /* bytes in interval          */
unsigned long long int qqstnplb;  /* Num of non-persistent      */
                                  /* MQPUT1 bytes in interval   */
unsigned long long int qqstplb;   /* Num of persistent MQPUT1   */
                                  /* bytes in interval          */
unsigned long long int qqstflpt;  /* Num of MQPUTs failed in    */
                                  /* interval                   */
unsigned long long int qqstflpl;  /* Num of MQPUT1s failed in   */
                                  /* interval                   */
unsigned long long int qqstfptc;  /* Num of messages fast put to*/
                                  /* waiting getter in interval */
unsigned long long int qqstfptb;  /* Num of bytes fast put to a */
                                  /* waiting getter in interval */
unsigned long long int qqststrm;  /* Num of successfully        */
                                  /* streamed messages in       */
                                  /* interval                   */
unsigned long long int qqstmsmi;  /* Minimum message size put in*/
                                  /* bytes in interval          */
unsigned long long int qqstmsma;  /* Maximum message size put in*/
                                  /* bytes in interval          */
unsigned long long int qqstmsav;  /* Average message size put in*/
                                  /* bytes in interval          */
unsigned long long int qqstgets;  /* Num of destructive MQGETs  */
                                  /* in interval                */
unsigned long long int qqstnpdg;  /* Num of Persistent          */
                                  /* destructive MQGETs         */
                                  /* in interval                */
```

AMQP performance improvements

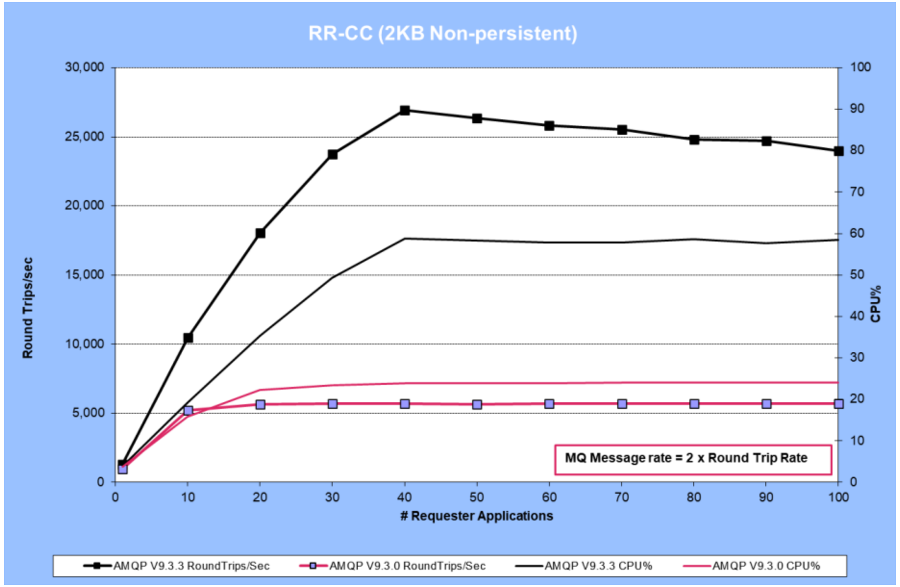
AMQP apps using QOS_AT_LEAST_ONCE have to acknowledge a message once they have processed it

When the queue manager receives the acknowledgement, it deletes the message

Prior to 9.3.3 this delete was done on a per-message basis

From 9.3.3 messages are acknowledged in batches or after a period of time has expired

This new approach provides a big improvement in throughput



Messaging technologies

Different approaches, satisfy different requirements

Apache Kafka

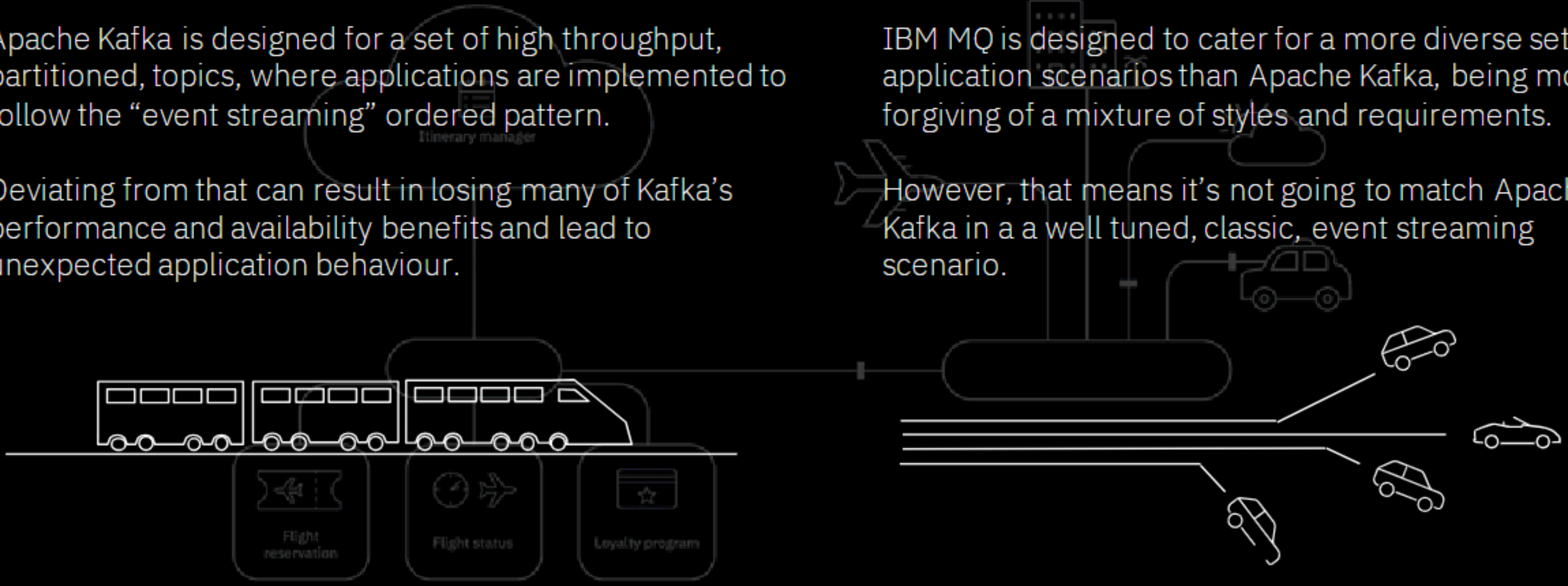
Apache Kafka is designed for a set of high throughput, partitioned, topics, where applications are implemented to follow the “event streaming” ordered pattern.

Deviating from that can result in losing many of Kafka’s performance and availability benefits and lead to unexpected application behaviour.

IBM MQ

IBM MQ is designed to cater for a more diverse set of application scenarios than Apache Kafka, being more forgiving of a mixture of styles and requirements.

However, that means it’s not going to match Apache Kafka in a a well tuned, classic, event streaming scenario.



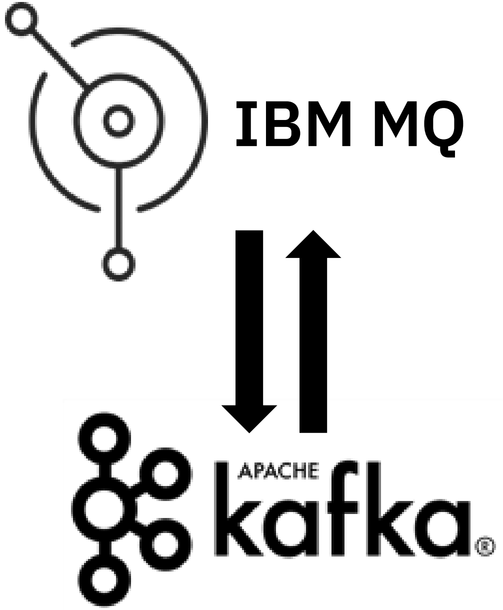
Understand the application’s style and behaviour and pick the technology based on that

Connecting MQ and Kafka: why?

With IBM MQ and Apache Kafka specialising in different aspects of the messaging spectrum, one on connectivity and the other on data, solutions often require messages to flow between the two.

Common scenarios:

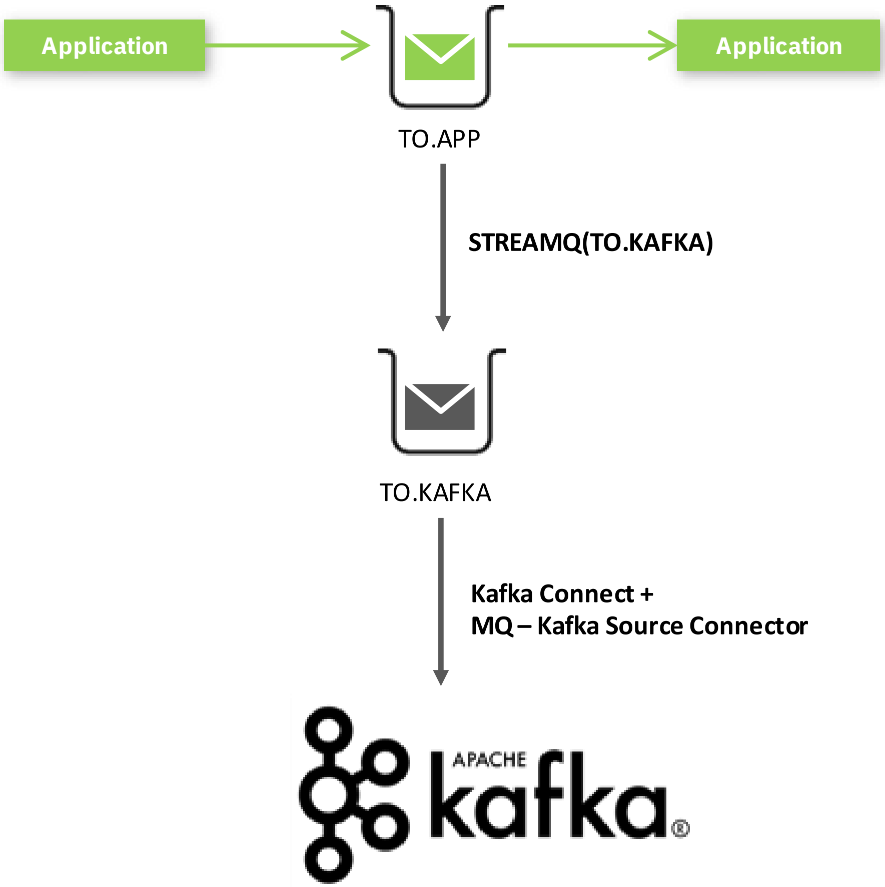
- Core banking system with MQ used as connectivity backbone. Business needs to take a copy of messages from MQ and push them into Kafka for analytics.
- Business needs to extend core banking system to emit data into Kafka, but doesn’t want to add network latency that might affect SLAs, so uses local queue manager as a bridge.
- Business needs to get data into Kafka from z/OS and wants to accelerate delivery so uses in-house MQ experience.
- Customer needs to get data into z/OS from distributed. Distributed development team has experience with Kafka, z/OS team want to exploit MQ integration with CICS / IMS.



MQ and Kafka

We see many customers wanting to use MQ and Kafka together

Either because they want to stream a copy of existing data moving through MQ into Kafka



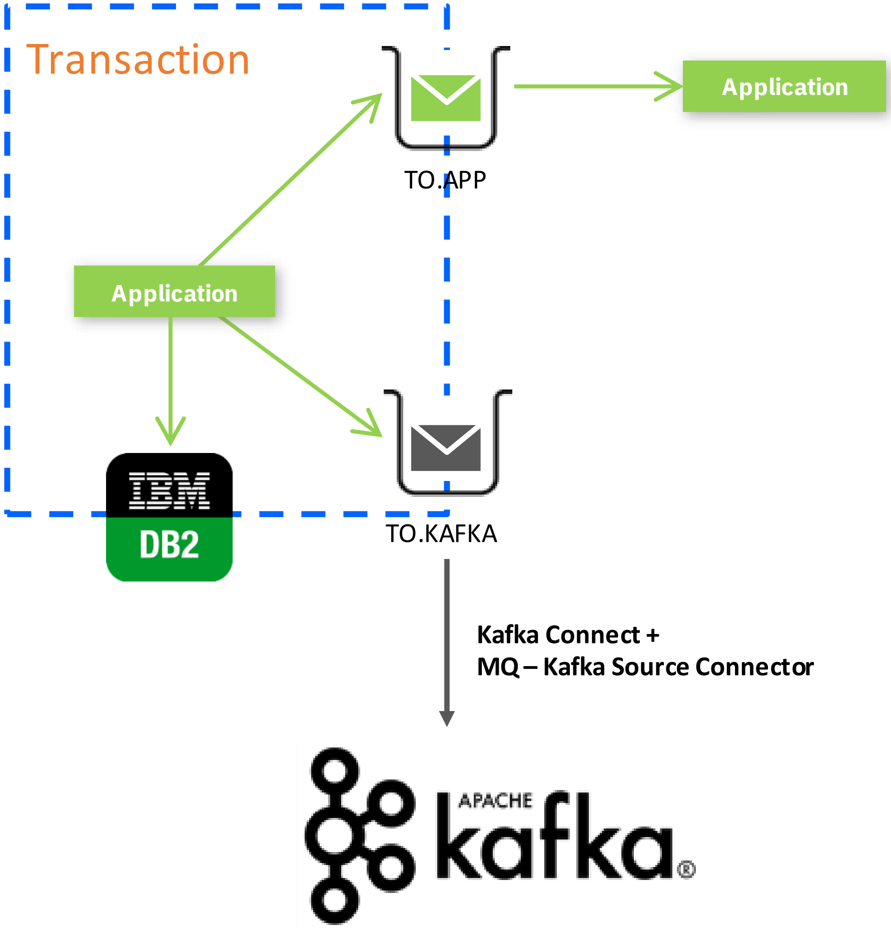
MQ and Kafka

We see many customers wanting to use MQ and Kafka together

Either because they want to stream a copy of existing data moving through MQ into Kafka

Or because they want a way to get data into Kafka that ensures data is only sent if their transaction commits

This is a very common z/OS use case



The IBM MQ / Kafka connector

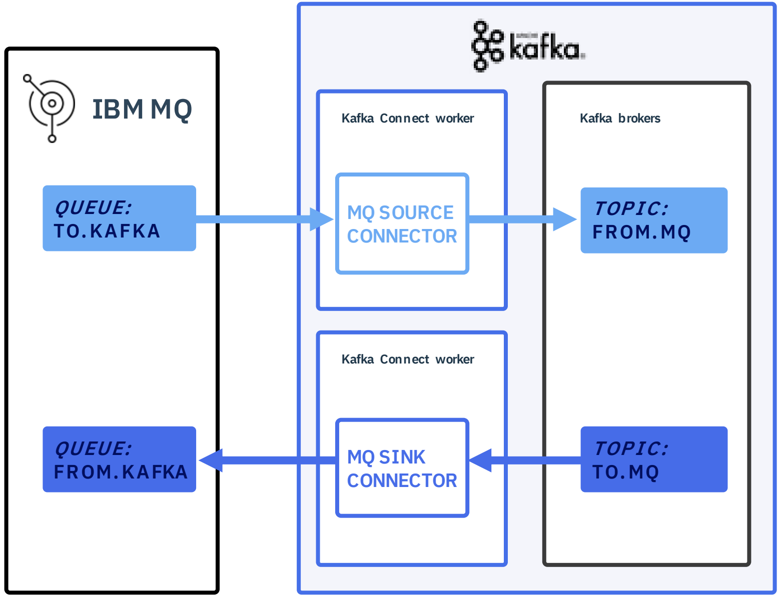
IBM provides MQ-Kafka connectors which can be used to move data between MQ and Kafka

The connectors can be used with any queue manager including those running on z/OS

Supported with either IBM Event Streams or IBM MQ Advanced entitlement, also available unsupported as open source

Connectors will be available from:

- Connector pack on z/OS (9.3.3 onwards)
- Fix Central – distributed (any version)
- GitHub releases page (any version)



https://github.com/ibm-messaging/kafka-connect-mq-sink/releases

https://github.com/ibm-messaging/kafka-connect-mq-source/releases

What’s new in 9.3.4?

IBM MQ 9.3.4 CD enhancements

★ MQ Console qmgr overview tab

Multi
Improvements in media image scheduling

Multi
Tuning options for log I/O warnings

Linux AIX ★
JSON web tokens

Multi
JMS support for influencing rebalancing

MQ Adv.
Native HA, smarter media image scheduling

MQ Adv. MQ Adv. z/OS MQ Adv. VUE ★
Exactly-once MQ/Kafka Connectors

z/OS ★
Simplified Java / JMS Topologies

MQ Adv. VUE MQ Adv. z/OS ★
Functional parity between Advanced and Advanced VUE

z/OS ★
CSQ1LOGP can deal with message properties

Appliance
SSH support for rsa-sha2-256 & rsa-sha2-512 signatures

©

https://www.ibm.com/docs/en/ibm-mq/9.3?topic=934-whats-new-in-mq

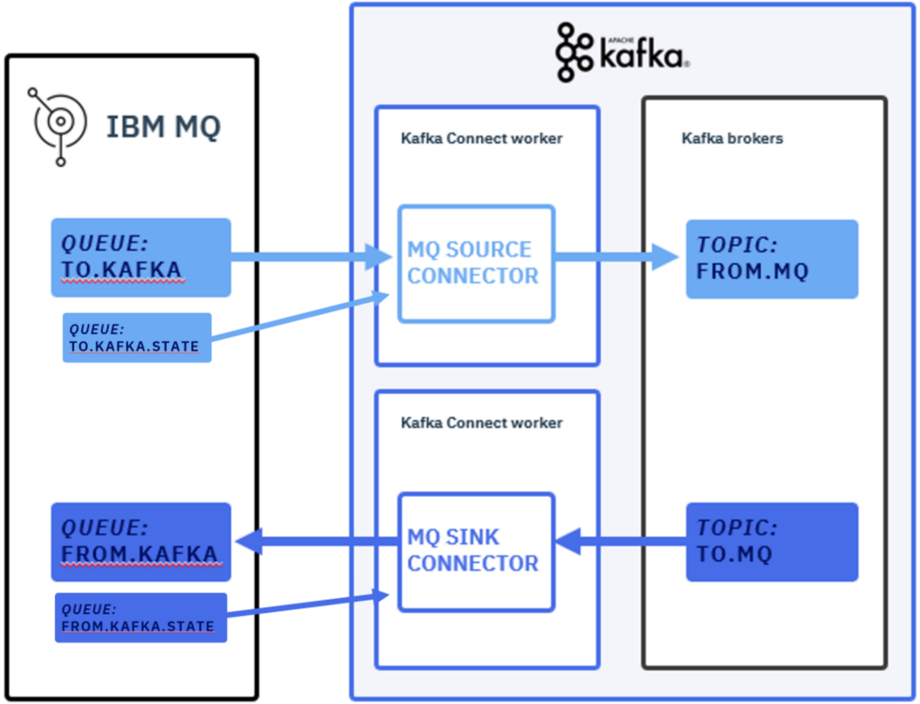
Exactly-once support

In 9.3.4, the MQ / Kafka connectors have been updated so that they support at-least-once **and exactly-once** message delivery semantics

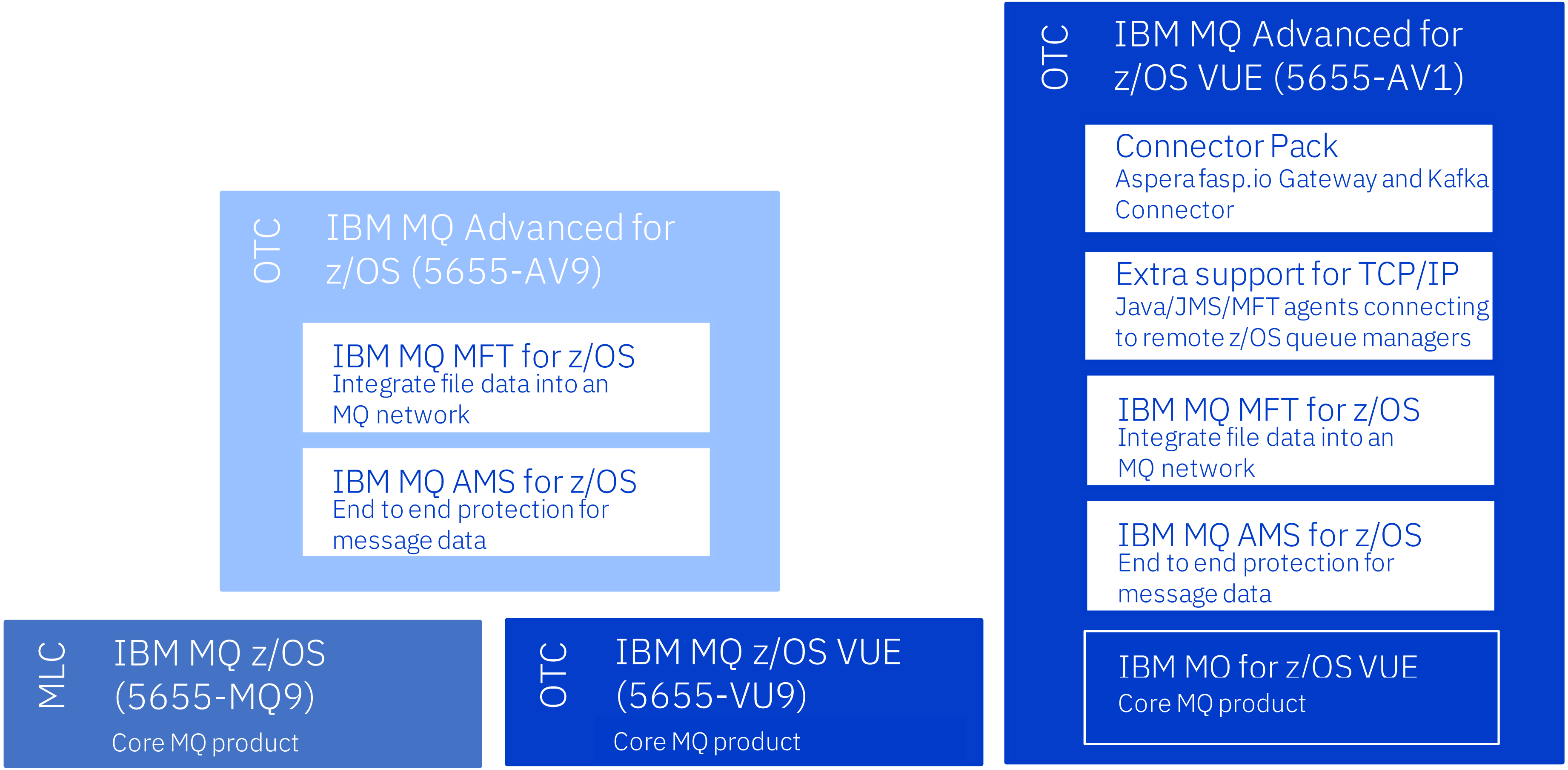
Previous connectors only supported at-least-once. No other vendor has exactly-once MQ connectors

Updated connectors aren’t open source. Can only be obtained with MQ Advanced or IBM ES entitlement

Exactly-once requires extra configuration to enable, both in MQ (a state queue), and Kafka. A relatively recent version of Kafka is also required



MQ for z/OS offerings: before



OTC

IBM MQ Advanced for z/OS (5655-AV9)

IBM MQ MFT for z/OS
Integrate file data into an MQ network

IBM MQ AMS for z/OS
End to end protection for message data

MLC

IBM MQ z/OS (5655-MQ9)

Core MQ product

OTC

IBM MQ z/OS VUE (5655-VU9)

Core MQ product

OTC

IBM MQ Advanced for z/OS VUE (5655-AV1)

Connector Pack
Aspera fasp.io Gateway and Kafka Connector

Extra support for TCP/IP
Java/JMS/MFT agents connecting to remote z/OS queue managers

IBM MQ MFT for z/OS
Integrate file data into an MQ network

IBM MQ AMS for z/OS
End to end protection for message data

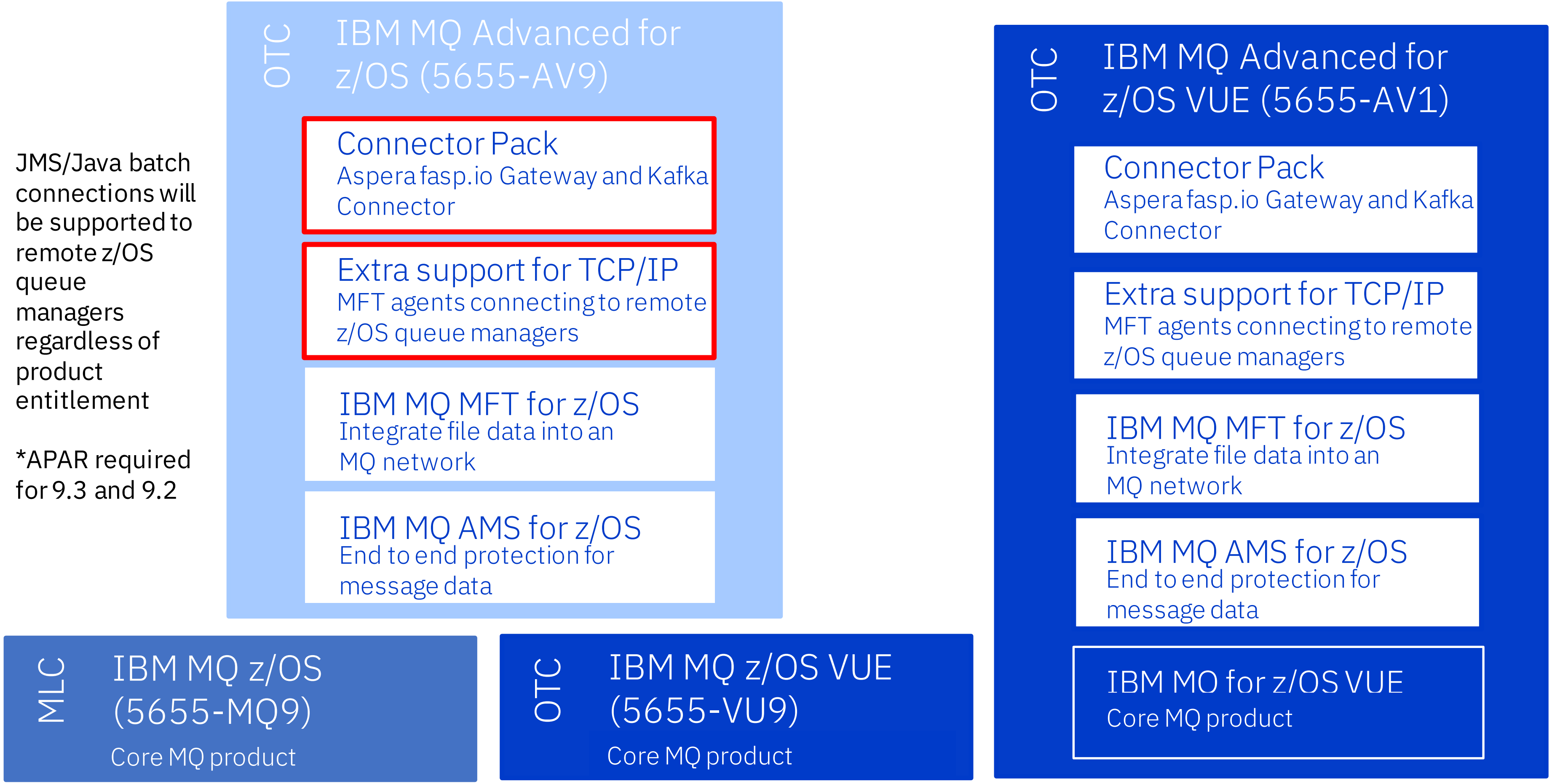
IBM MQ for z/OS VUE
Core MQ product

9.3.0 & 9.2.0

MQ for z/OS offerings: now

JMS/Java batch connections will be supported to remote z/OS queue managers regardless of product entitlement

*APAR required for 9.3 and 9.2



OTC IBM MQ Advanced for z/OS (5655-AV9)

Connector Pack
Aspera fasp.io Gateway and Kafka Connector

Extra support for TCP/IP
MFT agents connecting to remote z/OS queue managers

IBM MQ MFT for z/OS
Integrate file data into an MQ network

IBM MQ AMS for z/OS
End to end protection for message data

MLC IBM MQ z/OS (5655-MQ9)
Core MQ product

OTC IBM MQ z/OS VUE (5655-VU9)
Core MQ product

OTC IBM MQ Advanced for z/OS VUE (5655-AV1)

Connector Pack
Aspera fasp.io Gateway and Kafka Connector

Extra support for TCP/IP
MFT agents connecting to remote z/OS queue managers

IBM MQ MFT for z/OS
Integrate file data into an MQ network

IBM MQ AMS for z/OS
End to end protection for message data

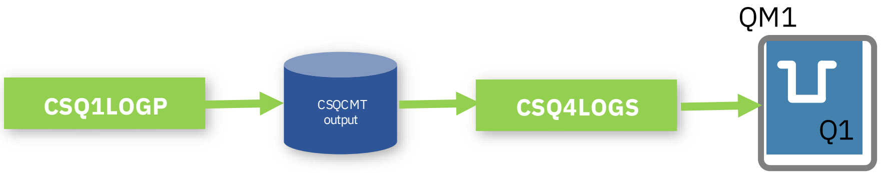
IBM MQ for z/OS VUE
Core MQ product

CSQ1LOGP EXTRACT supports message properties

CSQ1LOGP EXTRACT enhanced to fully extract messages with message properties, with the message properties going into an RFH2 header

This means that, if needed, those messages can be replayed, including the properties, using CSQ4LOGS, just like any other message

```
//STEP1 EXEC PGM=CSQ1LOGP,REGION=0M
//STEPLIB DD DISP=SHR,DSN=thlqual.SCSQANLE
//         DD DISP=SHR,DSN=thlqual.SCSQAUTH
//         DD DISP=SHR,DSN=thlqual.SCSQLOAD
//ARCHIVE DD DISP=SHR,DSN=xxx.yyy.A0030620
//         DD DISP=SHR,DSN=xxx.yyy.A0030621
//SYSPRINT DD SYSOUT=*
//SYSSUMRY DD SYSOUT=*
//CSQCMT DD DSN=xxx.MSGS.COMMIT,
// DISP=(NEW,CATLG),SPACE=(CYL,(1,10),RLSE),UNIT=SYSDA
//SYSIN DD *
EXTRACT(YES) SUMMARY(NO)
URID(xxxxxxxxxxxx)
/*
//STEP2 EXEC PGM=CSQ4LOGS,PARM=(‘QM1 REPLAY'),REGION=0M
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//         DD DSN=thlqual.SCSQAUTH,DISP=SHR
//         DD DSN=thlqual.SCSQLOAD,DISP=SHR
//FILEIN DD DSN=xxx.MSGS.COMMIT,DISP=SHR
//SYSDBOUT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
/*
```

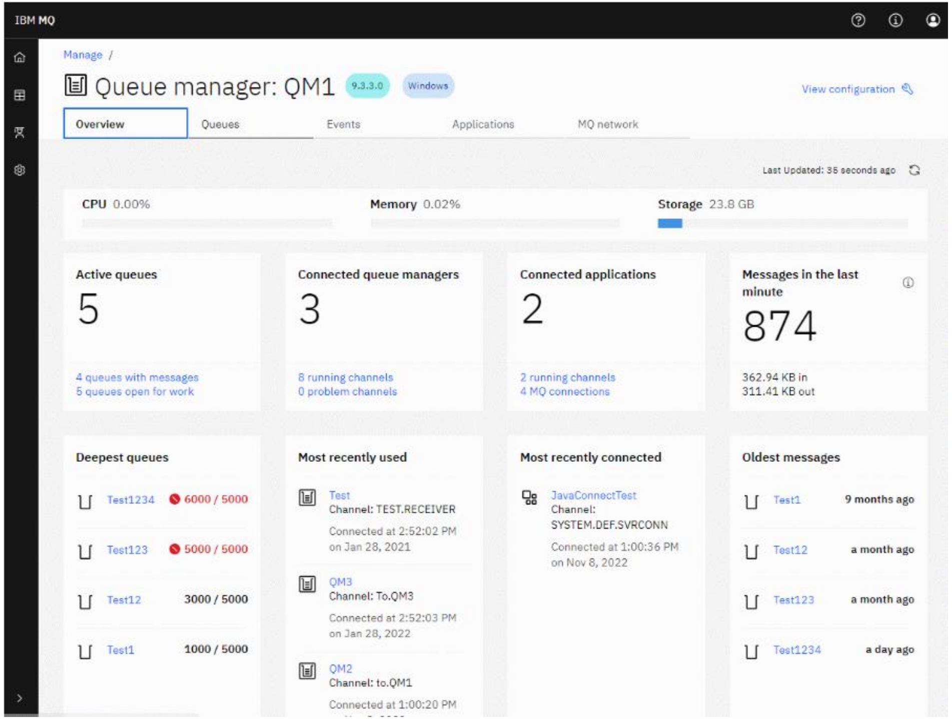


New overview tab for queue managers in IBM MQ Console

The IBM MQ Console **overview** tab of the queue manager page, introduced in v9.3.4, displays information about a queue manager and the resources it is consuming

- ✓ Eliminate manual look-up
- ✓ Immediately understand the overall state of a queue manager
- ✓ Act on any emerging problems before they become issues

Some of the information is derived from monitoring system topics so not everything is available on z/OS



Queue manager: QM1 9.3.3.0 Windows

Last Updated: 35 seconds ago

CPU 0.00%
Memory 0.02%
Storage 23.8 GB

Active queues
5
4 queues with messages
5 queues open for work

Connected queue managers
3
8 running channels
0 problem channels

Connected applications
2
2 running channels
4 MQ connections

Messages in the last minute
874
362.94 KB in
311.41 KB out

Deepest queues

Test1234	6000 / 5000
Test123	5000 / 5000
Test12	3000 / 5000
Test1	1000 / 5000

Most recently used

Test
Channel: TEST.RECEIVER
Connected at 2:52:02 PM on Jan 28, 2021

QM3
Channel: To.QM3
Connected at 2:52:03 PM on Jan 28, 2022

QM2
Channel: to.QM1
Connected at 1:00:20 PM

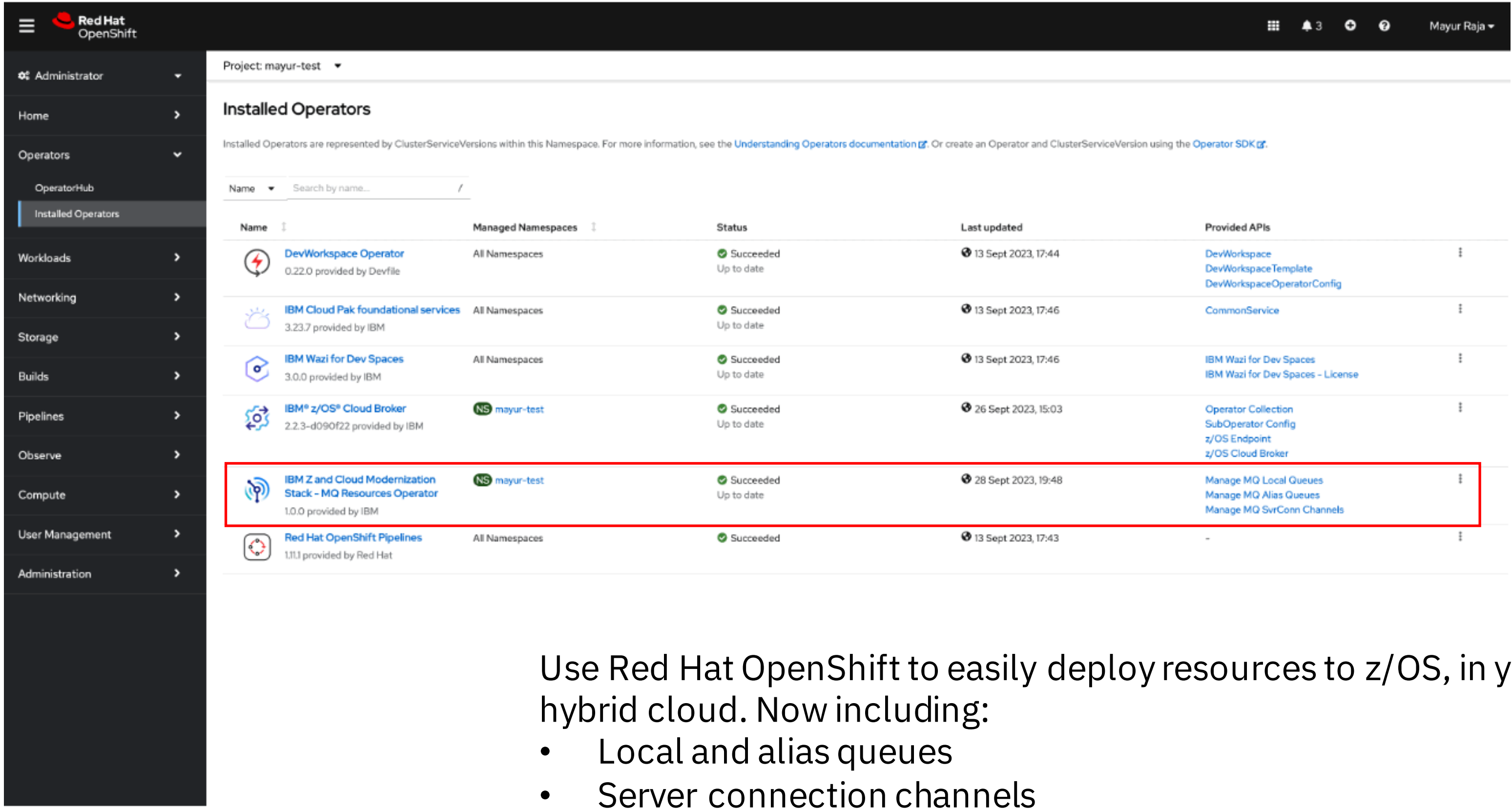
Most recently connected

JavaConnectTest
Channel: SYSTEM.DEF.SVRCONN
Connected at 1:00:36 PM on Nov 8, 2022

Oldest messages

Test1	9 months ago
Test12	a month ago
Test123	a month ago
Test1234	a day ago

MQ available in IBM z/OS Cloud Broker



Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the Understanding Operators documentation. Or create an Operator and ClusterServiceVersion using the Operator SDK.

Name	Managed Namespaces	Status	Last updated	Provided APIs
DevWorkspace Operator 0.22.0 provided by Devfile	All Namespaces	Succeeded Up to date	13 Sept 2023, 17:44	DevWorkspace DevWorkspaceTemplate DevWorkspaceOperatorConfig
IBM Cloud Pak foundational services 3.23.7 provided by IBM	All Namespaces	Succeeded Up to date	13 Sept 2023, 17:46	CommonService
IBM Wazi for Dev Spaces 3.0.0 provided by IBM	All Namespaces	Succeeded Up to date	13 Sept 2023, 17:46	IBM Wazi for Dev Spaces IBM Wazi for Dev Spaces - License
IBM® z/OS® Cloud Broker 2.2.3-d090f22 provided by IBM	NS mayur-test	Succeeded Up to date	26 Sept 2023, 15:03	Operator Collection SubOperator Config z/OS Endpoint z/OS Cloud Broker
IBM Z and Cloud Modernization Stack - MQ Resources Operator 1.0.0 provided by IBM	NS mayur-test	Succeeded Up to date	28 Sept 2023, 19:48	Manage MQ Local Queues Manage MQ Alias Queues Manage MQ SvrConn Channels
Red Hat OpenShift Pipelines 1.11.1 provided by Red Hat	All Namespaces	Succeeded	13 Sept 2023, 17:43	-

Use Red Hat OpenShift to easily deploy resources to z/OS, in your hybrid cloud. Now including:

- Local and alias queues
- Server connection channels

https://github.com/IBM/zos_mq_operator

Token based authentication

IBM MQ is moving towards a password-less approach to authentication so applications can operate in a more secure way with Multi-Factor Authentication (MFA) environment.

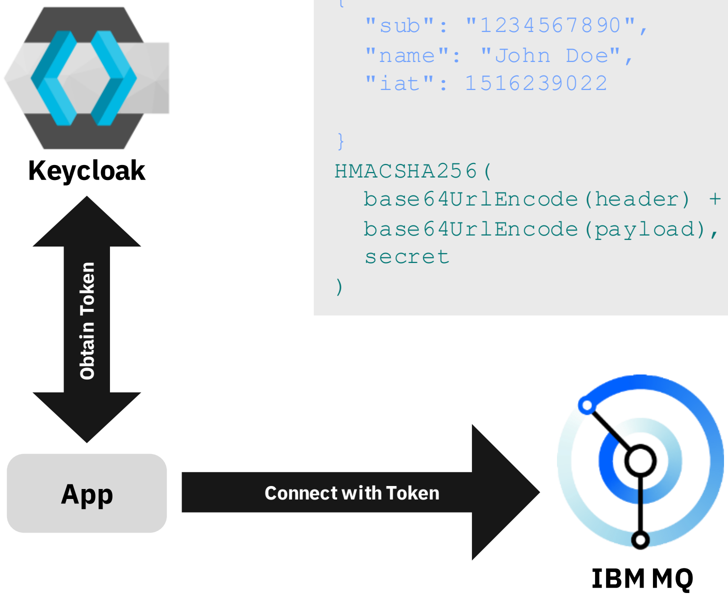
JSON Web Token (JWT) will provide a standard way to secure communication between two parties, with optional signature and/or encryption.

Eliminate the need for storing and transmitting passwords, which can be a security vulnerability. Enable single sign-on (SSO) and simplify the authentication process for users.

You can use one trusted issuer so your applications can authenticate with many services without separately registering with all of them.

Tokens can be used across multiple, diverse applications or other endpoints, enabling easier secure collaboration between enterprises and platforms.

```
{
  "alg": "HS256",
  "typ": "JWT"
}
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022

}
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
)
```



Token based authentication

From IBM MQ 9.3.4 client applications can provide tokens to authenticate with a queue manager. The queue manager must be configured to accept authentication tokens.

Queue managers in IBM MQ 9.3.4 and above that run on AIX®, Linux®, and OpenShift® Container Platforms are configured to accept tokens, IBM MQ MQI clients present tokens on connection and can be authenticated.

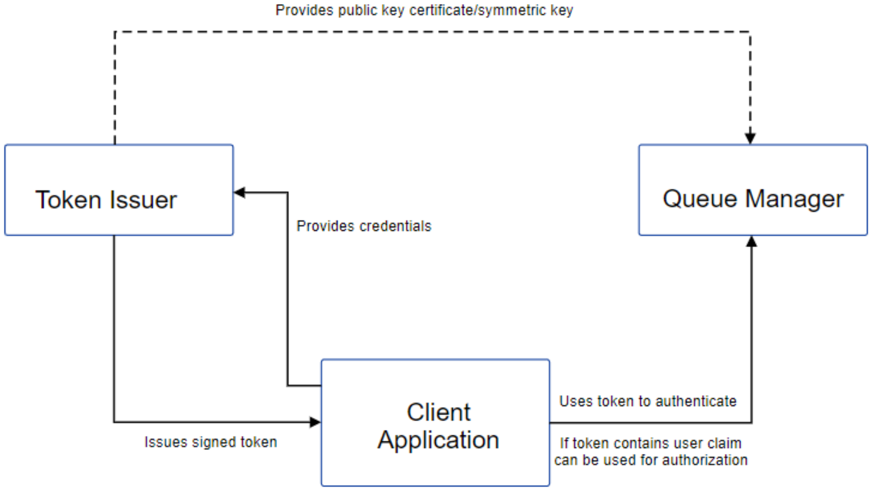
Both IBM MQ MQI clients and Java clients can remain unchanged and use security exits to send authentication tokens to the queue manager.

What is a token?

A token is a simple structure that contains information about a user and can easily be transferred between parties over the internet.

A JWT can be cryptographically signed to form a JWS. This allows for the cryptographic verification of the information inside the JWT.

Tokens that are used with IBM MQ must be valid JWT tokens that have been signed as defined by the internet standard RFC7515 for JSON Web Signature JWS and using an algorithm that IBM MQ supports.


Provides public key certificate/symmetric key
Token Issuer
Queue Manager
Provides credentials
Client Application
Issues signed token
Uses token to authenticate
If token contains user claim can be used for authorization

Thank you.