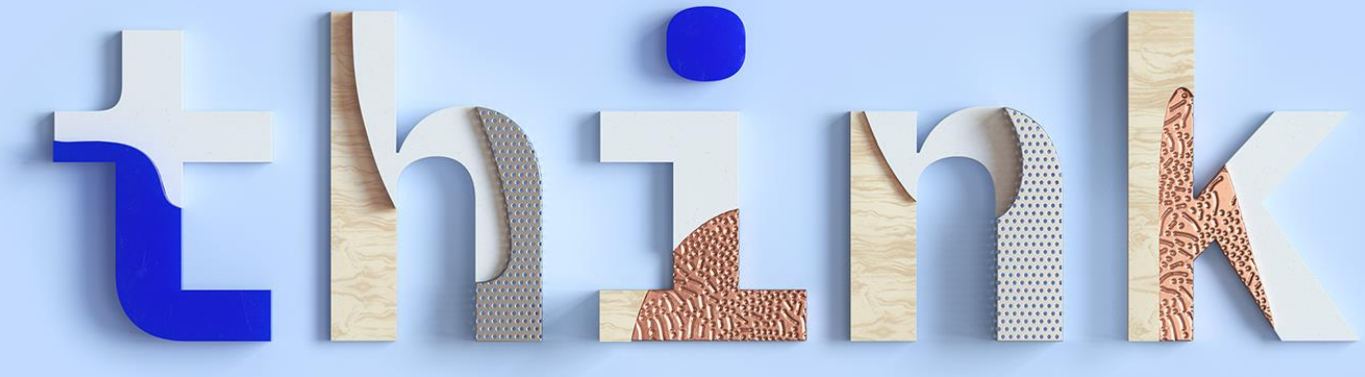
What's New in IBM MQ v9.0.5 CDR

think 2018

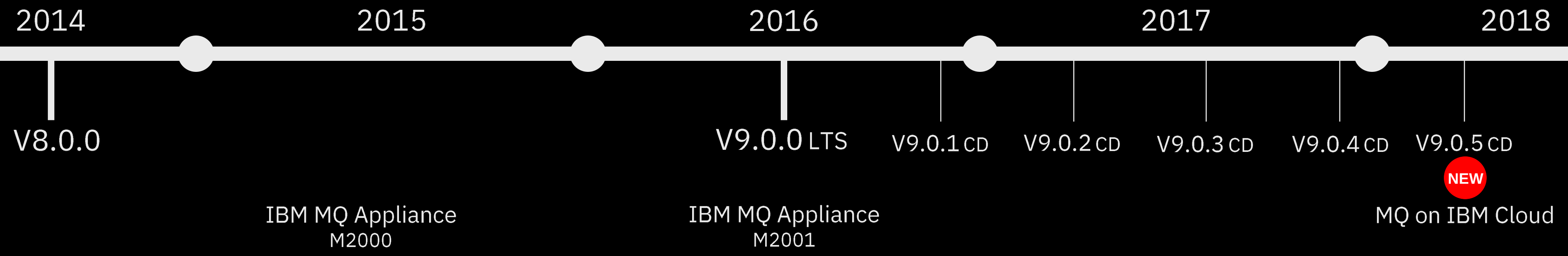
Carl Farkas

IBM Europe z Hybrid Cloud consultant

(many slides “borrowed” from David Ware’s Think presentation… thanks Dave!)



IBM MQ: long term support and continuous delivery



2014 2015 2016 2017 2018

V8.0.0

V9.0.0 LTS

V9.0.1 CD

V9.0.2 CD

V9.0.3 CD

V9.0.4 CD

V9.0.5 CD

NEW

IBM MQ Appliance
M2000

IBM MQ Appliance
M2001

MQ on IBM Cloud

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.

Intended for those that can continually integrate.

Long Term Support

Approximately every two years a new LTS version is released, rolling up many of 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, just like any previous version of MQ.

Statement of Direction: With the next release of MQ, IBM intends to initiate deployment of a new **LTS** release cycle

IBM MQ 9.0.5 CDR

- Announced 13 March and available as of 16 March. See ZP18-0084 letter.

- Several new enhancements....
  - Replicated Data Queue Manager improved
  - IBM MQ Managed File Transfer new features (eg. REST support)
  - Performance improvements

- See Knowledge Center q130600_ for v9.0.5 details

- Statement Of Direction:
- *“With the next release of MQ, IBM intends to initiate deployment of a new Long Term Support (LTS) release cycle. The LTS release will provide many of the features that will be delivered in the Version 9.0.x Continuous Delivery releases”*

- For Distributed platforms, available for download via Passport advantage; search CJ329ML

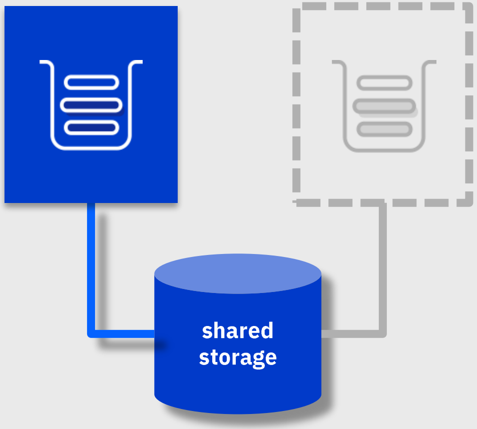
- For MQ z/OS CDR, APAR PI89159 to find the PTFs to update your installation from 9.0.4 to 9.0.5, APAR P192774 for MQ Managed File Transfer for z/OS and APAR P193404 for MQ Advanced for z/OS Value Unit Edition.

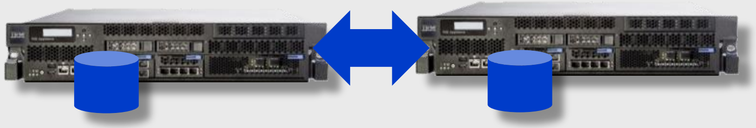
HA, there are no excuses

MQ delivers HA through the ability to build horizontally scaled, active-active systems and typically **active-passive HA** of the data itself*, the messages.

Traditionally active-passive HA has been achieved through **HA clusters** or **multi instance** queue managers. Both rely on highly available infrastructure to be setup and relied on.

The **MQ Appliance** changed this with a fully integrated HA solution, providing built in machine to machine data replication and failover.





* z/OS shared queue provides active-active HA of the message data!

Replicated Data Queue Managers

Linux only, MQ Advanced HA solution with no need for a shared file system or HA cluster

MQ configures the underlying resources to make setup and operations natural to an MQ user

Three-way replication for quorum support

Synchronous data replication for once and once only transactional delivery of messages

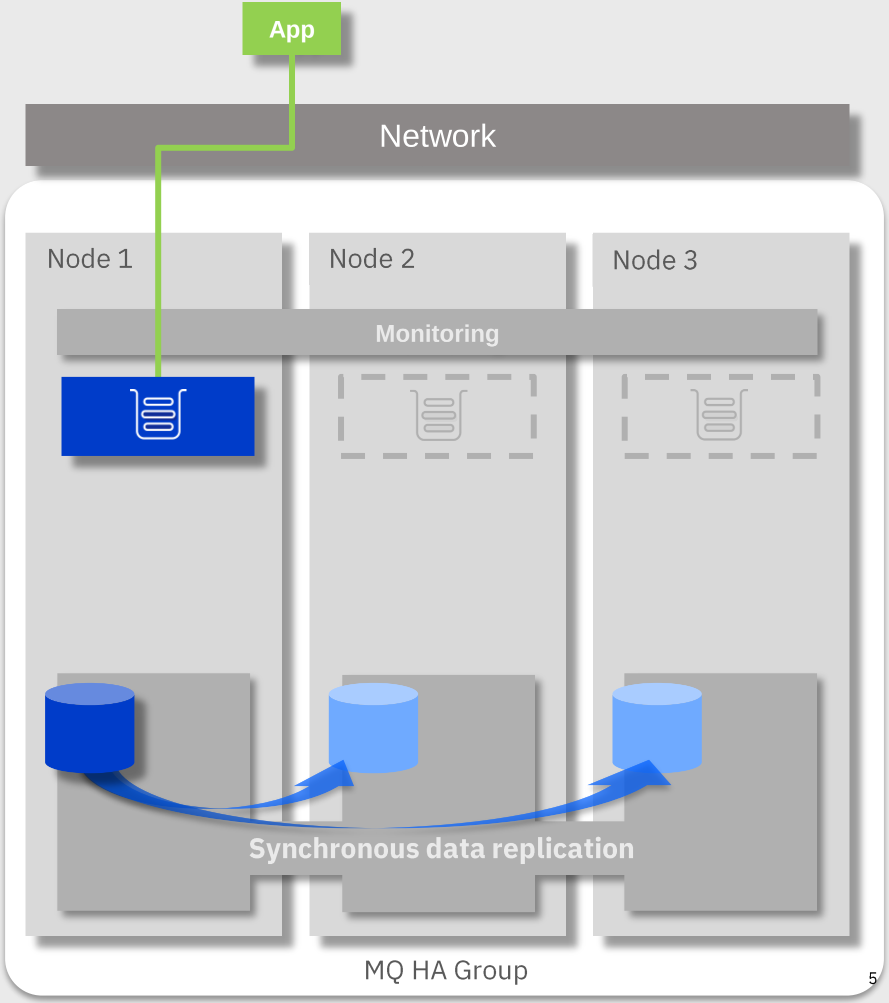
Active/passive queue managers with **automatic takeover**

Per queue manager control to support active/active utilisation of nodes

Per queue manager **IP address** to provide simple application setup

Supported on **RHEL v7 x86-64 only**

New in V9.0.4 CD MQ Advanced for Linux



Replicated Data Queue Managers

Linux only, MQ Advanced HA solution with no need for a shared file system or HA cluster

MQ configures the underlying resources to make setup and operations natural to an MQ user

Three-way replication for quorum support

Synchronous data replication for once and once only transactional delivery of messages

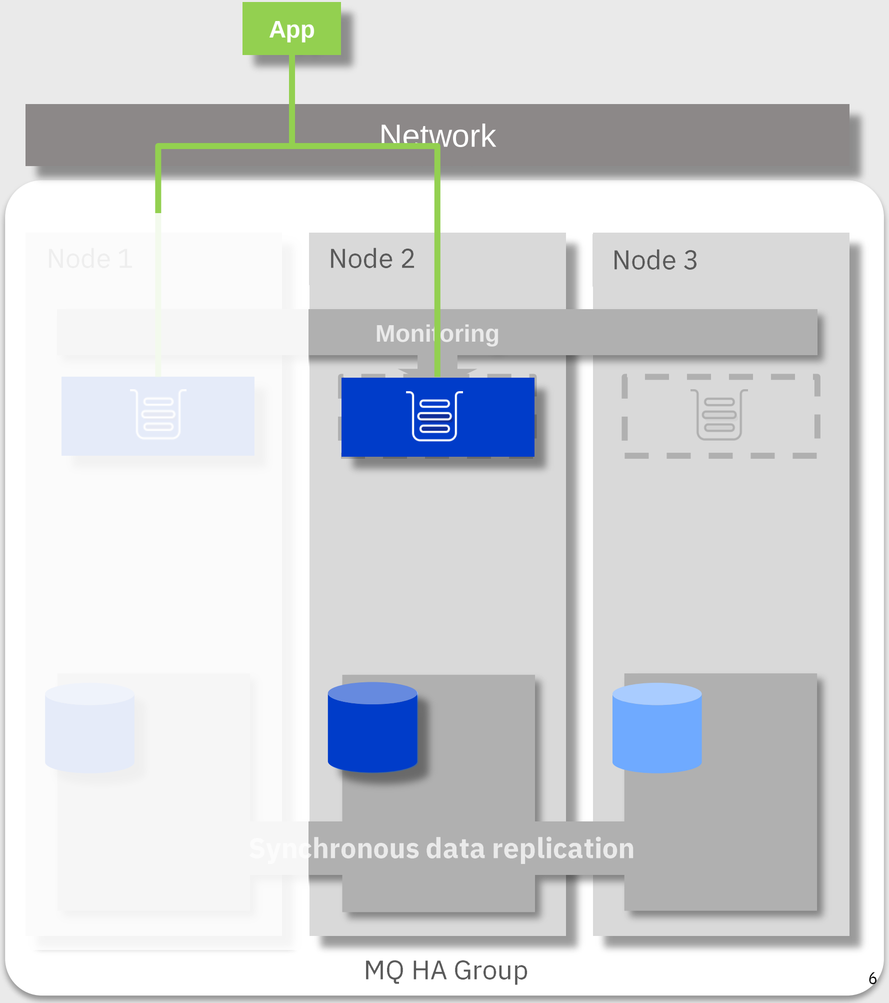
Active/passive queue managers with **automatic takeover**

Per queue manager control to support active/active utilisation of nodes

Per queue manager **IP address** to provide simple application setup

Supported on **RHEL v7 x86-64 only**

New in V9.0.4 CD MQ Advanced for Linux



Replicated Data Queue Managers

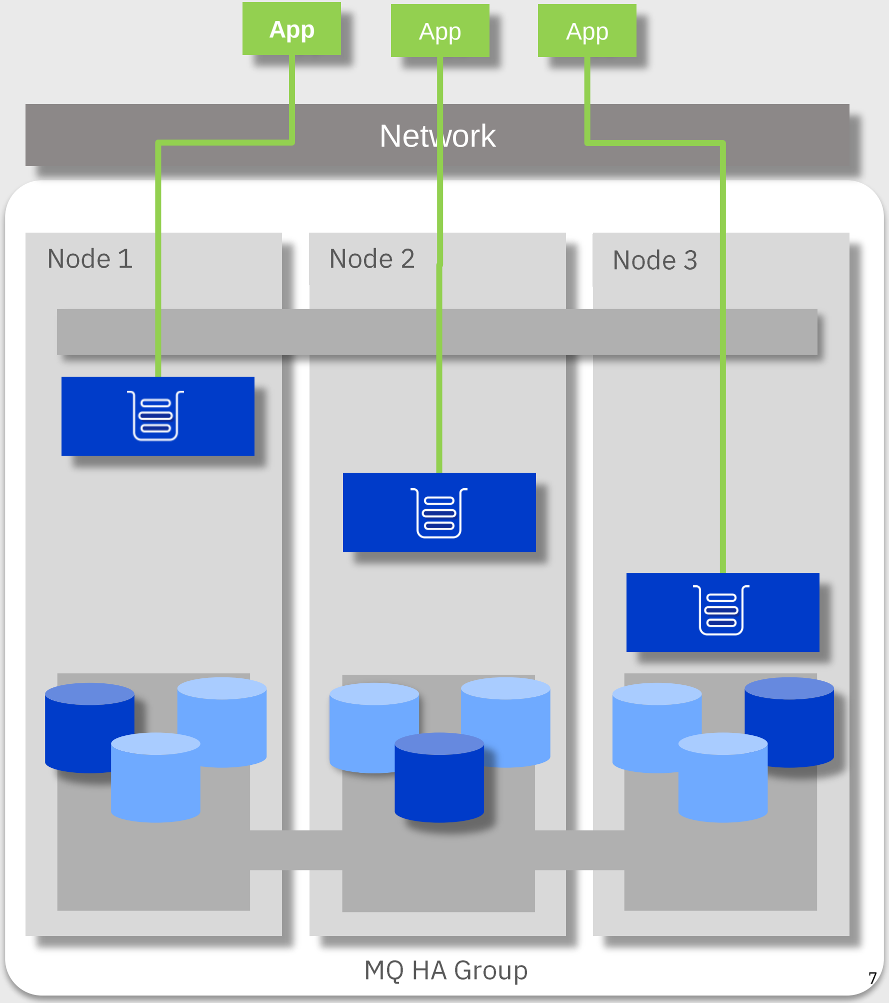
Recommended deployment pattern:

- Spread the workload across multiple queue managers and distribute them across all three nodes
- Even better, more than one queue manager per node for better failover distribution
- Use MQ Clusters for additional routing of messages to work around problems

MQ **licensing** is aligned to maximise benefits

- One full **IBM MQ Advanced** license and two **High Availability Replica** licenses (previously named *Idle Standby*)

New in V9.0.4 CD MQ Advanced for Linux



Connecting Applications at Scale

Binding an application to a specific queue manager restricts future changes, impacting availability, scalability and maintenance of the system

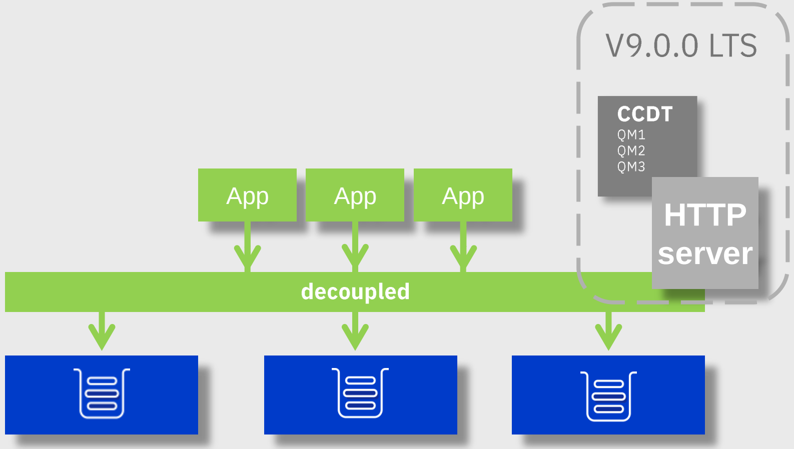
Run applications remote from the queue managers, connecting as **MQ clients**

For horizontal scaling, create **groups of matching queue managers** and connect applications to the group, not each queue manager

Now you can add and remove queue managers without changing the application

Use one of multiple ways to map applications to queue managers and their location. For example

Infrastructure level routing
Client Channel Definition Tables (**CCDTs**)



Replicated Data Queue Managers

Manual failover

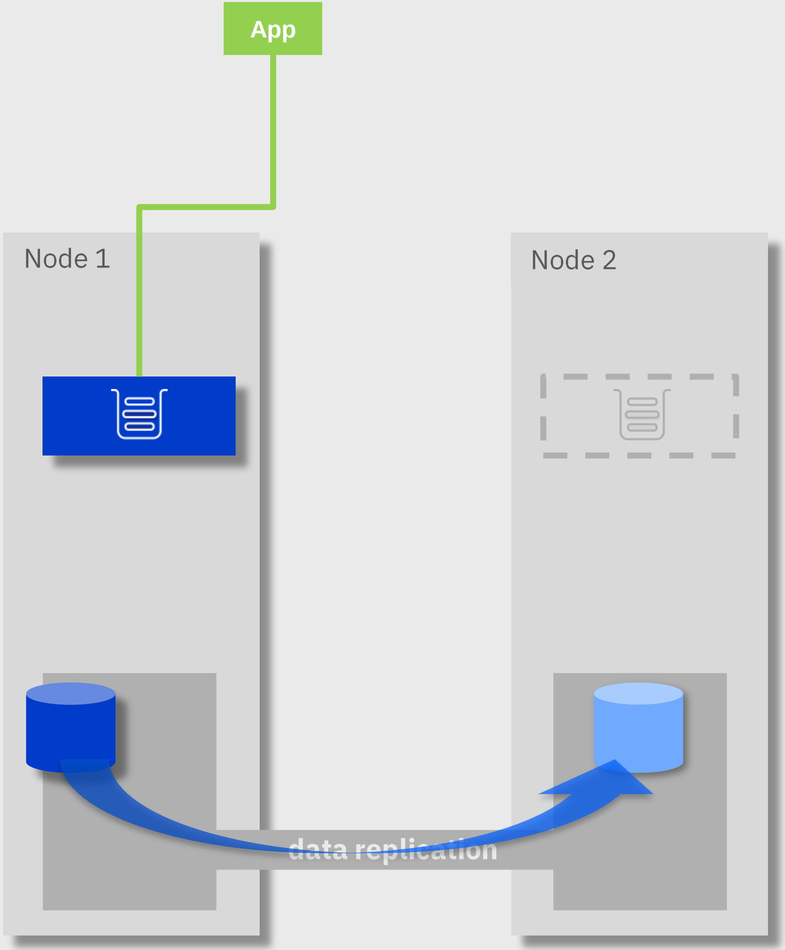
9.0.5 CD MQ Advanced adds the ability to build a looser coupled pair of nodes for data replication but no automatic failover, typically for **Disaster Recovery**

Data replication can be

> **Asynchronous** for systems separated by a high latency network
>
> **Synchronous** for systems on a low latency network

No automatic takeover means no need for a third node to provide a quorum

New in V9.0.5 CD MQ Advanced for Linux



Bridging to MQ

As well as connecting a wide array of applications directly to an MQ system, there are a growing set of bridges and connectors between MQ and external systems

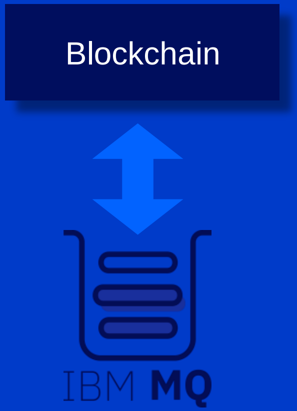
Salesforce

Integrate MQ’s publish/subscribe with Salesforce. Exchange **Salesforce events** and **MQ publications** using the MQ Bridge for Salesforce with no need for your backend applications to connect to Salesforce directly.

Salesforce
IBM MQ

Blockchain

Use MQ messages to query and update a Blockchain ledger. Connects to **Hyperledger** Fabric networks in IBM Cloud and locally. Supported for use with V9.0.x **MQ Advanced** queue managers



Kafka

IBM MQ sink and source connectors are currently being openly developed by IBM and provides **as-is**, allowing you to connect your MQ systems with your Kafka clusters

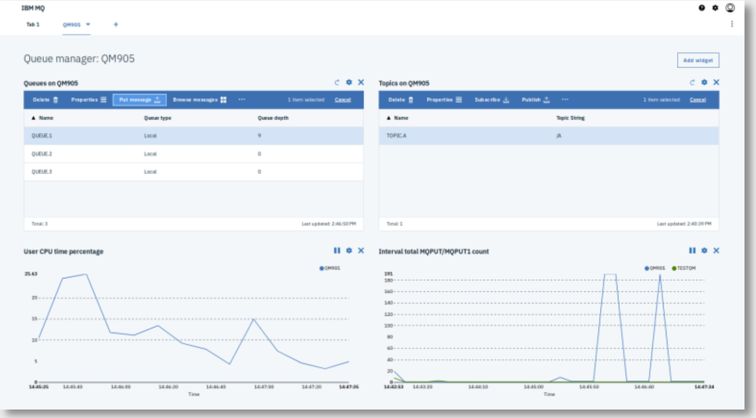
www.confluent.io/product/connectors

Kafka
IBM MQ

Software MQ Web Console

Point a browser at the MQ installation to create and manage queue managers and their resources

Provides a very simple way to access MQ resources



Queue manager: QM905

Queues on QM905

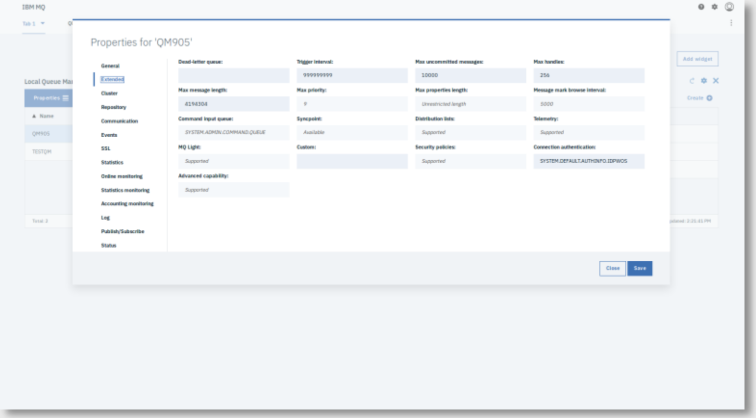
▲ Name	Queue type	Queue depth
QUEUE.1	Local	9
QUEUE.2	Local	0
QUEUE.3	Local	0

Total: 3 Last updated: 2:46:50 PM

Topics on QM905

▲ Name	Topic String
TOPIC.A	/A

Total: 1 Last updated: 2:40:39 PM



Properties for 'QM905'

Dead-letter queue:

Trigger interval: 999999999

Max uncommitted messages: 10000

Max handles: 256

Max message length: 4194304

Max priority: 9

Max properties length: Unrestricted length

Message mark browse interval: 5000

Command input queue: SYSTEM.ADMIN.COMMAND.QUEUE

Syncpoint: Available

Distribution lists: Supported

Telemetry: Supported

MQ Light: Supported

Custom:

Security policies: Supported

Connection authentication: SYSTEM.DEFAULT.AUTHINFO.IDPWOS

Advanced capability: Supported

Restful administration

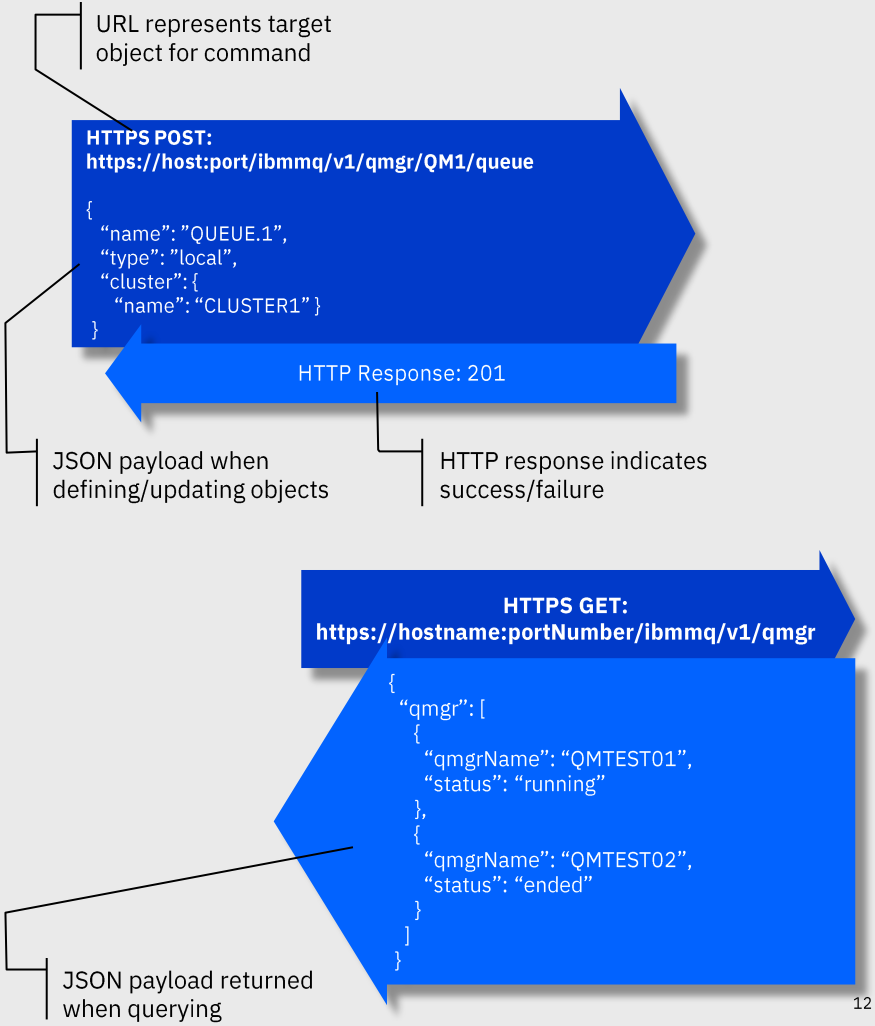
MQ has supported scripting and programmatic administration for many years, but it requires MQ knowledge and tooling.

MQ has been increasing support for a RESTful administrative API to enable much of what's available today with MQSC and PCF. But in a more intuitive way to many, using the URL and structured JSON payloads to define the operations.

Being over **HTTPS** also enables the embedding of MQ administrative operations into many environments and tools that previously would not be possible

Each 9.0.5 CD release has extended the level of REST support: *you can now view channel information and run MQSC commands on remote queue managers. See q130620_ mftrest in the KC for more info.*

NEW



URL represents target object for command

HTTPS POST:
https://host:port/ibmmq/v1/qmgr/QM1/queue

```
{
  "name": "QUEUE.1",
  "type": "local",
  "cluster": {
    "name": "CLUSTER1" }
}
```

HTTP Response: 201

JSON payload when defining/updating objects

HTTP response indicates success/failure

HTTPS GET:
https://hostname:portNumber/ibmmq/v1/qmgr

```
{
  "qmgr": [
   {
     "qmgrName": "QMTEST01",
     "status": "running"
   },
   {
     "qmgrName": "QMTEST02",
     "status": "ended"
   }
  ]
}
```

JSON payload returned when querying

Restful administration

MQ has supported scripting and programmatic administration for many years, but it requires MQ knowledge and tooling.

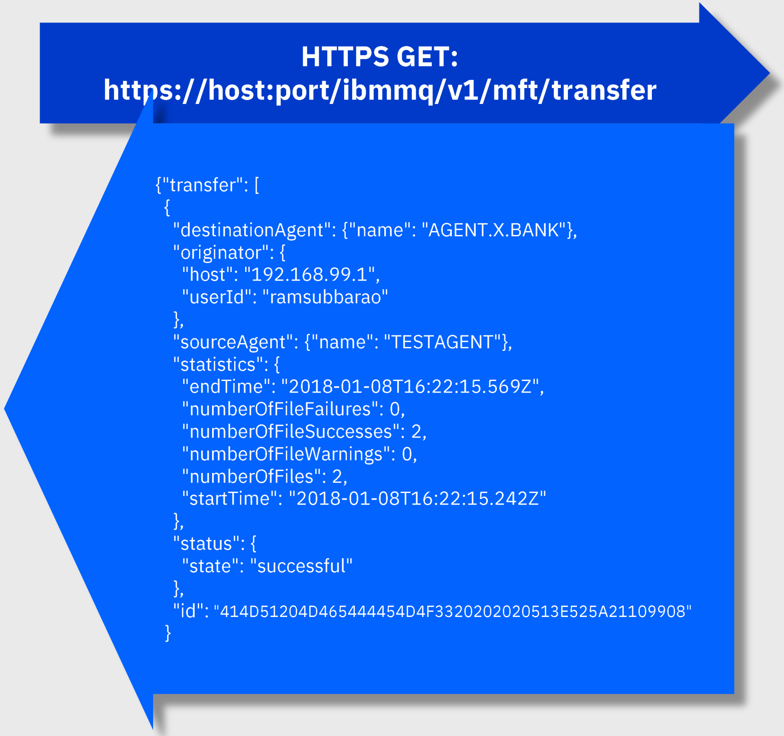
MQ has been increasing support for a RESTful administrative API to enable much of what's available today with MQSC and PCF. But in a more intuitive way to many, using the URL and structured JSON payloads to define the operations.

Being over **HTTPS** also enables the embedding of MQ administrative operations into many environments and tools that previously would not be possible

Each 9.0.x CD release has extended the level of REST support, *and it continues…*

9.0.5 adds support for monitoring your Managed File Transfer system with **agent** and **transfer** status lists over REST

NEW



HTTPS GET:
https://host:port/ibmmq/v1/mft/transfer

```
{"transfer": [
 {
  "destinationAgent": {"name": "AGENT.X.BANK"},
  "originator": {
   "host": "192.168.99.1",
   "userId": "ramsubbarao"
  },
  "sourceAgent": {"name": "TESTAGENT"},
  "statistics": {
   "endTime": "2018-01-08T16:22:15.569Z",
   "numberOfFileFailures": 0,
   "numberOfFileSuccesses": 2,
   "numberOfFileWarnings": 0,
   "numberOfFiles": 2,
   "startTime": "2018-01-08T16:22:15.242Z"
  },
  "status": {
   "state": "successful"
  },
  "id": "414D51204D465444454D4F3320202020513E525A21109908"
 }
```

See DevWorks article on using MFT REST API

Implementing MQ MFT REST

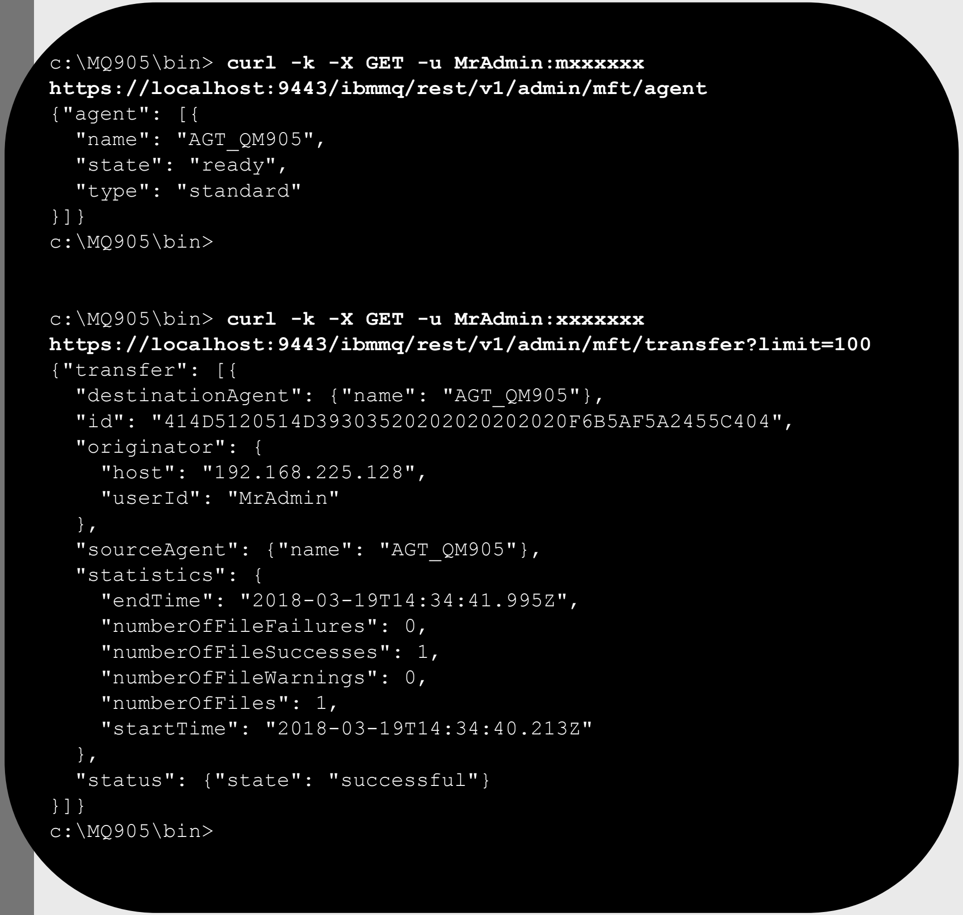
Get the MQ REST interface configured and running. Essentially 4 commands to get it going for test!

1. Select and copy a supplied mqwebuser.xml sample security model
2. Configure a port (setmqweb)
3. Update the mqwebuser.xml with MFT statements:

```
<variable name="mqRestMftEnabled"
value="true" />

 <variable
name="mqRestMftCoordinationQmgr"
value="QM905" />
```

4. Start it up (strmqweb)

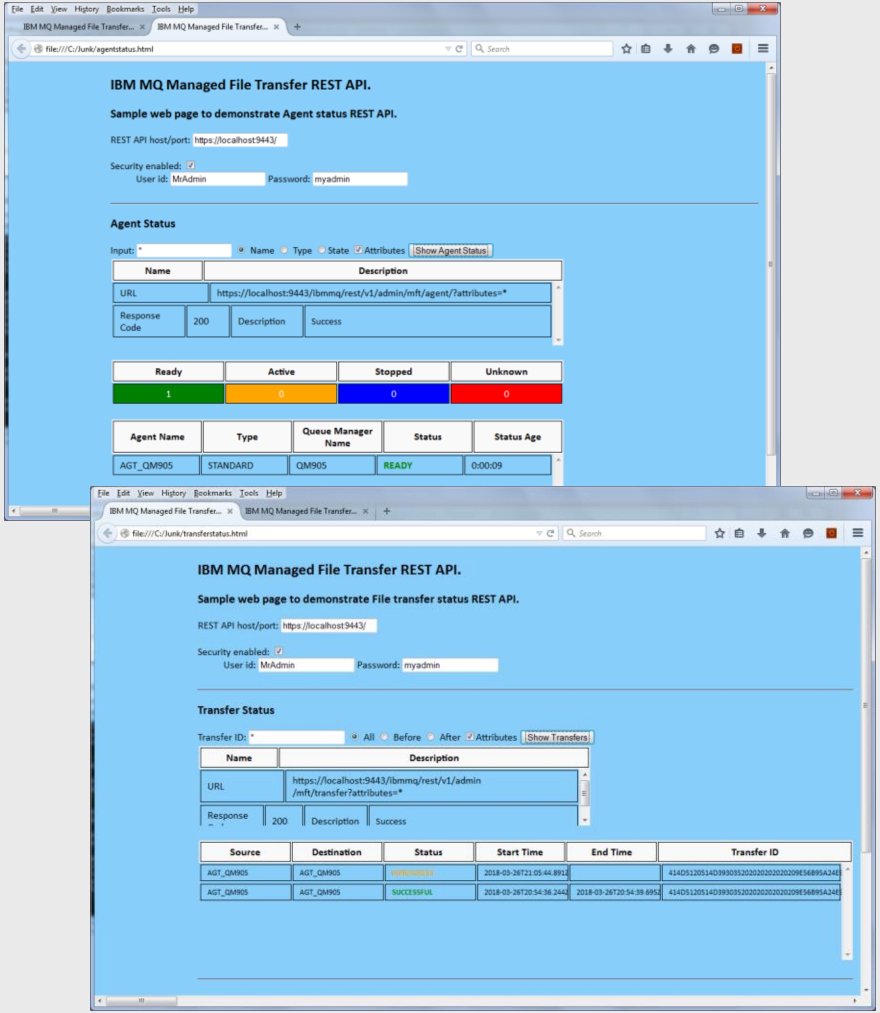


```
c:\MQ905\bin> curl -k -X GET -u MrAdmin:mxxxxxx
https://localhost:9443/ibmmq/rest/v1/admin/mft/agent
{"agent": [{
  "name": "AGT_QM905",
  "state": "ready",
  "type": "standard"
}]}
c:\MQ905\bin>


c:\MQ905\bin> curl -k -X GET -u MrAdmin:xxxxxxx
https://localhost:9443/ibmmq/rest/v1/admin/mft/transfer?limit=100
{"transfer": [{
  "destinationAgent": {"name": "AGT_QM905"},
  "id": "414D5120514D39303520202020202020F6B5AF5A2455C404",
  "originator": {
    "host": "192.168.225.128",
    "userId": "MrAdmin"
  },
  "sourceAgent": {"name": "AGT_QM905"},
  "statistics": {
    "endTime": "2018-03-19T14:34:41.995Z",
    "numberOfFileFailures": 0,
    "numberOfFileSuccesses": 1,
    "numberOfFileWarnings": 0,
    "numberOfFiles": 1,
    "startTime": "2018-03-19T14:34:40.213Z"
  },
  "status": {"state": "successful"}
}]}
c:\MQ905\bin>
```

New MQ MFT sample for MFT REST API

- Available for download at https://github.com/ibm-messaging/mft-sample-web-ui
- Runs either as a WAR deployed in a (different!) Liberty instance (licence restriction) or…
- Runs by directly running HTML in your Browser
- See the README.md delivered with package for details


File Edit View History Bookmarks Tools Help
IBM MQ Managed File Transfer... ×
IBM MQ Managed File Transfer... ×
Search

file:///C:/Junk/agentstatus.html

IBM MQ Managed File Transfer REST API.

Sample web page to demonstrate Agent status REST API.

REST API host/port: https://localhost:9443/

Security enabled: ☑
User id: MrAdmin Password: myadmin

Agent Status

Input: * ● Name ○ Type ○ State ☑ Attributes [Show Agent Status]

Name	Description		
URL	https://localhost:9443/ibmmq/rest/v1/admin/mft/agent/?attributes=*		
Response Code	200	Description	Success

Ready	Active	Stopped	Unknown
1	0	0	0

Agent Name	Type	Queue Manager Name	Status	Status Age
AGT_QM905	STANDARD	QM905	READY	0:00:09

file:///C:/Junk/transferstatus.html

IBM MQ Managed File Transfer REST API.

Sample web page to demonstrate File transfer status REST API.

REST API host/port: https://localhost:9443/

Security enabled: ☑
User id: MrAdmin Password: myadmin

Transfer Status

Transfer ID: * ● All ○ Before ○ After ☑ Attributes [Show Transfers]

Name	Description		
URL	https://localhost:9443/ibmmq/rest/v1/admin /mft/transfer?attributes=*		
Response	200	Description	Success

Source	Destination	Status	Start Time	End Time	Transfer ID
AGT_QM905	AGT_QM905	INPROGRESS	2018-03-26T21:05:44.8912		414D5120514D39303520202020202020209E56B95A24E
AGT_QM905	AGT_QM905	SUCCESSFUL	2018-03-26T20:54:36.2442	2018-03-26T20:54:39.6952	414D5120514D39303520202020202020209E56B95A24E

Managing diagnostic data

The need to centrally collect and analyse diagnostic data is increasing, using tools such as Splunk, Elasticsearch and Grafana

MQ generates a wide range of information and has demonstrated how this can be collected using off the shelf tooling

Subscribing to metrics in MQ V9 makes that even easier

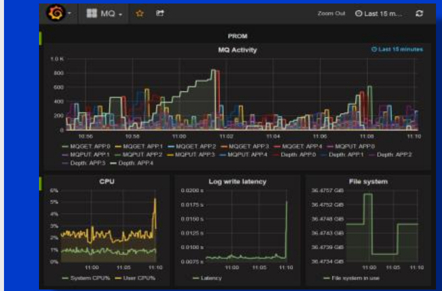
MQ 9.0.5 has seen further enhancements to the error log data it generates to aid such solutions

Universal timestamps and severity levels
Separated inserts
JSON output
Multiple logs
Syslog output

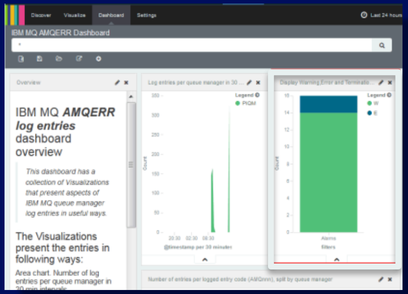
NEW

See KC q018795_

Publish MQ statistics to Prometheus and Grafana



Forward MQ error logs to ElasticSearch or Splunk


Discover
Visualize
Dashboard
Settings
Last 24 hours
IBM MQ AMQERR Dashboard
Overview
Log entries per queue manager in 30 ...
Legend
PIQM
350
300
250
200
150
100
50
0
Count
20:30
02:30
08:30
@timestamp per 30 minutes
Display Warning,Error and Terminato...
Legend
W
E
16
14
12
10
8
6
4
2
0
Count
Alarms
filters
Number of entries per logged entry code (AMQnnn), split by queue manager

IBM MQ ***AMQERR log entries*** dashboard overview

This dashboard has a collection of Visualizations that present aspects of IBM MQ queue manager log entries in useful ways.

The Visualizations present the entries in following ways:

Area chart. Number of log entries per queue manager in 30 min intervals

Error logs output JSON for easy parsing

```
{
   "ibm_messageId":"AMQ6287I",
   "ibm_arithInsert1":0,
   "ibm_arithInsert2":0,
   "ibm_commentInsert1":"Linux 4.13.0-36-generic (MQ Linux (x86-64 platform) 64-bit)",
   "ibm_commentInsert2":"/opt/mqm (Installation1)",
   "ibm_commentInsert3":"9.0.5.0 (p905-L180228.1)",
   "ibm_datetime":"2018-03-04T13:18:27.506Z",
   "ibm_serverName":"QM905",
   "type":"mq_log",
   "host":"david-VirtualBox",
   "loglevel":"INFO",
   "module":"amqxeida.c:6238",
   "ibm_sequence":"1520169507_506462655",
   "ibm_processId":2119,
   "ibm_threadId":1,
   "ibm_version":"9.0.5.0",
   "ibm_processName":"strmqm",
   "ibm_userName":"david",
   ”ibm_installationName":"Installation1",
   "ibm_installationDir":"/opt/mqm",
   "message":"AMQ6287I: IBM MQ V9.0.5.0 (p905-L180228.1).”
}
```