Red Hat AMQ 2020.Q4

Release Notes for Red Hat AMQ Broker 7.8

Release Notes for AMQ Broker

Last Updated: 2022-03-15

Red Hat AMQ 2020.Q4 Release Notes for Red Hat AMQ Broker 7.8

Release Notes for AMQ Broker

Legal Notice

Abstract

These release notes contain the latest information about new features, enhancements, fixes, and issues contained in the AMQ Broker 7.8 release.

Table of Contents

CHAPTER 1. LONG TERM SUPPORT FOR AMQ BROKER 7.8

AMQ Broker 7.8 has been designated as a Long Term Support (LTS) release version. Bug fixes and security advisories will be made available for AMQ Broker 7.8 in a series of micro releases (7.8.1, 7.8.2, and so on) for a period of at least 12 months.

This means that you will be able to get recent bug fixes and security advisories for AMQ Broker *without* having to upgrade to a new minor release.

Note the following important points about the LTS release stream:

- The LTS release stream provides only bug fixes. No new enhancements will be added to this stream.
- To remain in a supported configuration, you must upgrade to the latest micro release in the LTS release stream.
- The LTS version will be supported for at least 12 months from the time of the AMQ Broker 7.8.0 GA release.

Support for Red Hat Enterprise Linux and OpenShift Container Platform

The AMQ Broker 7.8 LTS version supports:

- Red Hat Enterprise Linux 6, 7, and 8
- OpenShift Container Platform 3.11 and 4.5 and 4.6

Note the following important points about support for Red Hat Enterprise Linux and OpenShift Container Platform:

- AMQ Broker 7.8 is the **last** version that will support Red Hat Enterprise Linux 6 and OpenShift Container Platform 3.11.
- Red Hat **does not** guarantee that AMQ Broker 7.8 will be supported on *future* versions (that is, versions greater than 4.6) of OpenShift Container Platform.

For information about issues resolved in AMQ Broker 7.8 LTS micro releases, see AMQ 7 Broker - 7.8.x Resolved Issues.

CHAPTER 2. ENHANCEMENTS

This section describes a highlighted set of enhancements and new features in AMQ Broker 7.8. For a complete list of enhancements in the release, see AMQ Broker 7.8.0 Enhancements.

New version of AMQ Management Console

AMQ Broker 7.8 includes a new version of AMQ Management Console. For more information on using the console, see Using AMQ Management Console in *Managing AMQ Broker*.

New database certifications

AMQ Broker 7.8 adds support for PostgreSQL 11.5 and MySQL 8. For complete information about the databases supported by different versions of AMQ Broker, see Red Hat AMQ 7 Supported Configurations.

Federating address and queues

In AMQ Broker 7.8 you can configure federation of addresses and queues. Federation enables transmission of messages between brokers, without requiring the brokers to be in a common cluster. For example, federation is suitable for reliably sending messages from one cluster to another. This transmission might be across a Wide Area Network (WAN), *Regions* of a cloud infrastructure, or over the Internet. For more information, see Federating addresses and queues in *Configuring AMQ Broker*.

Disabling queues

In AMQ Broker 7.8, you can disable queues that you have defined in your broker configuration. For example, you might want to define a queue so that clients can subscribe to it, but you are not ready to use the queue for message routing. Alternatively, you might want to stop message flow to a queue, but still keep clients bound to the queue. In these cases, you can disable the queue. For more information, see Disabling queues in *Configuring AMQ Broker*.

Performance improvements for vertical scaling of queues

AMQ Broker 7.8 adds scalability improvements that improve broker performance when a deployment automatically scales to large numbers of queues. This improvement applies to all supported protocols, but is particularly beneficial for MQTT, which is often used in large-scale deployments. This performance enhancement is most noticeable for broker deployments with **very** large numbers of queues, for example, 50,000 or more.

Updating running Operator-based broker deployments with address settings

In AMQ Broker 7.8, you can now add address settings to an Operator-based broker deployment that is already running. Support for including address settings in the Custom Resource (CR) instance for a broker deployment was added in AMQ Broker 7.7. However, in 7.7, you needed to configure the address settings when creating the broker deployment for the first time. For more information on configuring addresses, queues, and address settings, see Configuring addresses and queues for Operator-based broker deployments in *Deploying AMQ Broker on OpenShift*.

Additions to the base audit logger

The base audit logger now logs when you pause and resume addresses. To learn how to configure logging, see Logging in *Configuring AMQ Broker*.

New metric for broker address memory usage percentage

In 7.8, the Prometheus metrics plugin for AMQ Broker exports a new metric named **artemis_address_memory_usage_percentage**. This metric is the total address memory used by all addresses on a broker as a percentage of the value of the **global-max-size** parameter. To learn how to configure the Prometheus metrics plugin, see Monitoring broker runtime metrics in *Managing AMQ Broker*.

Improved configuration of diverts

In 7.8, if you use AMQ Management Console or the management API to configure a runtime divert on a live broker, the divert is automatically propagated to the backup broker. This was not the case in previous releases.

Specifying a custom Init Container image

The latest version of the Operator for 7.8 uses a specialized container called an *Init Container* to generate the broker configuration. By default, the Operator uses a built-in Init Container image. However, you can also specify a custom Init Container image that modifies or adds to the configuration created by the built-in Init Container. For more information, see Specifying a custom Init Container image in *Deploying AMQ Broker on OpenShift*..

Operator support for multiple container platforms

In 7.8, the AMQ Broker Operator supports the following container platforms:

- OpenShift Container Platform
- OpenShift Container Platform on IBM Z
- OpenShift Container Platform on IBM Power Systems

Operator support for OpenShift Container Platform on IBM Power Systems is new in 7.8. A version of the Operator for AMQ Broker 7.5 supports OpenShift Container Platform on IBM Z.

In 7.5, you need to install and deploy a **separate** version of the Operator for each supported platform. In 7.8, you need to install only a **single** version, which supports all three container platforms. Based on the container platform that you are using, the Operator automatically chooses a broker container image to use in your deployment.

To learn how to install the latest version of the Operator, see the following sections in *Deploying AMQ Broker on OpenShift*:

- Installing the Operator using the CLI
- Installing the Operator using the Operator Lifecycle Manager

Automatic selection of broker container image by Operator

In the latest version of the Operator for 7.8, when you use a Custom Resource (CR) instance to create a broker deployment, you no longer need to explicitly specify a broker container image name in the CR. Instead, when you deploy the CR, the Operator automatically determines the appropriate broker container image to use. This also applies to the Init Container that generates the broker configuration. To learn more, see How the Operator chooses container images in *Deploying AMQ Broker on OpenShift*.

RHEL 8 Operator

An Operator named **Red Hat Integration - AMQ Broker for RHEL 8 (Multiarch)** is available on x86_64 platforms, IBM Z, and IBM Power Systems. It supports the following channels:

- **7.x** - This channel will update to **7.9** when available.
- **7.8.x** - This is the Long Term Support (LTS) channel.

To determine which Operator to choose, see the Red Hat Enterprise Linux Container Compatibility Matrix.

Operator channels

In the latest version of the Operator for 7.8, the following new update channels are available for the **Red Hat Integration - AMQ Broker** Operator`:

- **7.x** - This is equivalant to the **current** channel which is now deprecated.
- **7.8.x** - This is the Long Term Support (LTS) channel.

Operator versioning

In the latest version of the Operator for 7.8, Operators now adopt the same versioning scheme as AMQ Broker. For example, the previous Operator release on x86_64 platforms was version **0.19** in OperatorHub, that version is updated to **7.8.2-opr-1**.

Documentation updates

The AMQ Broker documentation is updated to provide instructions regarding the new Operators and channels and support for IBM Z and IBM Power Systems.

Additional resources

- For a complete list of enhancements in the AMQ Broker 7.8 release, see AMQ Broker 7.8 Enhancements.

CHAPTER 3. DEPRECATED FEATURES

This section describes features that have been deprecated from AMQ Broker.

Hawtio dispatch console plugin

Starting in 7.3, AMQ Broker no longer ships with the Hawtio dispatch console plugin, **dispatch-hawtio-console.war**. Previously, the dispatch console was used to manage AMQ Interconnect. However, AMQ Interconnect now uses its own, standalone web console.

Network pinger

Starting in 7.5, network pinging is a deprecated feature. Network pinging cannot protect a broker cluster from network isolation issues that can lead to irrecoverable message loss. This feature will be removed in a future release. Red Hat continues to support existing AMQ Broker deployments that use network pinging. However, Red Hat no longer recommends use of network pinging in new deployments. For guidance on configuring a broker cluster for high availability and to avoid network isolation issues, see Implementing high availability in *Configuring AMQ Broker*..

Application templates for AMQ Broker on OpenShift Container Platform

Starting in 7.8, the use of application templates for deploying AMQ Broker on OpenShift Container Platform is a deprecated feature. This feature will be removed in a future release. Red Hat continues to support existing deployments that are based on application templates. However, Red Hat does not recommend using application templates for new deployments. For new deployments, Red Hat recommends using the AMQ Broker Operator. For information on installing and deploying the Operator, see Deploying AMQ Broker on OpenShift Container Platform using the AMQ Broker Operator in *Deploying AMQ Broker on OpenShift*.

CHAPTER 4. TECHNOLOGY PREVIEW

This section describes Technology Preview features in AMQ Broker 7.8.

IMPORTANT

Technology Preview features are not supported with Red Hat production service-level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them for production. For more information, see Red Hat Technology Preview Features Support Scope.

AMQP server connections

In AMQ Broker 7.8, a broker can initiate connections to other endpoints using the AMQP protocol. This means, for example, that the broker can connect to other AMQP servers (not necessarily instances of AMQ Broker) and create elements on those connections. For more information, see Broker Connections in the Apache ActiveMQ Artemis documentation.

Viewing brokers in Fuse Console

You can configure an Operator-based broker deployment to use Fuse Console for OpenShift instead of AMQ Management Console. When you have configured your broker deployment appropriately, Fuse Console discovers the brokers and displays them on a dedicated **Artemis** tab. For more information, see Viewing brokers in Fuse Console in *Deploying AMQ Broker on OpenShift*.

NOTE

Viewing brokers in Fuse Console is a Technology Preview feature **for Fuse 7.8**.

CHAPTER 5. FIXED ISSUES

This section describes a highlighted set of issues that have been fixed in AMQ Broker 7.8. For a complete list of issues that have been fixed in the release, see AMQ Broker 7.8.0 Fixed Issues and AMQ 7 Broker - 7.8.x Resolved Issues.

- **ENTMQBR-1815 - Hawtio view changes on automated refresh**
  Previously, when automatic refresh was enabled, the Hawtio console updated the screen every 5 seconds. This behavior also switched the view to the **Attributes** tab, causing you to lose focus from any other tab you were viewing. This issue is now resolved.

- **ENTMQBR-2890 - Infrequently, creating a CR instance with size n > 1 causes the the nth broker pod to start and immediately restart once**
  Previously, if you used a Custom Resource (CR) instance to deploy a broker cluster using the AMQ Broker Operator, the final broker Pod in the deployment (as determined by the **size** property of your CR) started and then immediately restarted one time before it was usable. This issue is now resolved. None of the broker Pods in the deployment undergo a restart before they are usable.

- **ENTMQBR-3059 - AMQ Broker Operator: Operator doesn't pick up CR name after restart / update**
  Previously, if you created a clustered broker deployment of two or more brokers that were configured to use persistence and message migration, the AMQ Broker Operator might generate an invalid name when instantiating the scaledown controller for message migration. Specifically, this issue occurred if the Operator had been restarted or its image had been updated prior to scaledown of the broker deployment. The result of this situation was that you needed to delete and recreate your broker deployment. This issue is now resolved.

- **ENTMQBR-3514 - AMQ Broker Operator: Address not created if address CR is submitted before broker instantiated**
  Previously, for an Operator-based broker deployment, if you created a Custom Resource (CR) instance for addresses before the brokers had been instantiated, the Operator failed to create the addresses. This issue is now resolved.

- **ENTMQBR-3578 - AMQ Broker Operator: No operator support for, upon startup, using the existing CR instances as a baseline to move forward from**
  Previously, when the AMQ Broker Operator started, it did not check for existing Custom Resource (CR) instances in your project. This behavior meant that if you needed to restart the Operator (for example, to apply a new Operator image version), the Operator and the broker deployment were no longer in sync. In this situation, you needed to delete and then recreate your broker deployment. This issue is now resolved.

- **ENTMQBR-3587 - Avoid notifications when shutting down on critical IO error**
  Previously, when shutting itself down due to a critical IO error, the broker generated multiple notifications that triggered disk IO. These notifications could delay or event prevent shutdown of the broker. This issue is now resolved.

- **ENTMQBR-3617 - User information on the hawtio console for durable address is null**
  Previously, when a consumer created a shared, durable subscription, the AMQ Management Console might show the associated user as **null** for the subscription queue that was automatically created by the broker. This issue is now resolved.

- **ENTMQBR-3692 - User information on the hawtio console for durable address is null**
  Previously, if a message-driven bean (MDB) used the ActiveMQ Java Connector Architecture (JCA) resource adapter (for example, in Red Hat JBoss Enterprise Application Platform) to create a durable topic subscription, the MDB would fail to deploy. This issue is now resolved.

- ENTMQBR-3705 - **LVQ + non-destructive not delivering message to existing consumer**
  Previously, when a consumer created a shared, durable address, AMQ Management Console might show the associated user as **null**. This issue is now resolved.

- ENTMQBR-3710 - **Incorrect audit message on queue resume**
  Previously, if you paused and then resumed a queue, the audit logger incorrectly included text that described the resume event as another pause event. For example:

  ```
  2020-07-09 11:18:00,352 [AUDIT](qtp795748540-39) AMQ601213: User
  amq(amq)@192.168.100.1:40858 is resuming on target resource:
  QueueImpl[name=helloworld, postOffice=PostOfficeImpl
  2020-07-09 11:18:00,352 [AUDIT](qtp795748540-39) AMQ601721: User
  amq(amq)@192.168.100.1:40858 has paused queue helloworld
  ```

  This issue is now resolved.

- ENTMQBR-3719 - **LegacyLDAPSecuritySettingPlugin allows new user to access any newly created destinations**
  Previously, when a new permission was added to LDAP, the **LegacyLDAPSecuritySettingPlugin** plugin used the new permission to modify the default security match. This could break authorization for existing users. This issue is now resolved.

- ENTMQBR-3726 - **JVM property hawtio.role doesn't parse a role with space and hyphen**
  Previously, if the **artemis.profile** file defined a **hawtio.role** property containing whitespace or a hyphen, the property didn't work properly. This issue is now resolved.

- ENTMQBR-3744 - **Enabling group rebalancing with default / non-zero consumer-window-size can lead to out-of-order message consumption**
  Previously, if some connected consumers were using a value of **consumerWindowSize** greater than zero (meaning that messages were pre-fetched into buffers on those clients) **and** you configured the broker to use group rebalancing (by setting **group-rebalance** or **default-group-rebalance** to **true**), this might result in out-of-order message consumption. This issue is now resolved.

- ENTMQBR-3752 - **Backup broker cannot reestablish connection with its master**
  In the event of a network outage, it is possible for both brokers in a live-backup group to become live at the same time (a situation known as *network isolation* or *split brain*). Previously, if this situation occurred, any connected AMQ Core Protocol JMS clients received incorrect broker topology information. As a result, when the network and split brain issues were solved, the client could not reconnect to the right brokers. To work around this issue, you needed to restart the clients. This issue is now resolved.

- ENTMQBR-3782 - **page-max-concurrent-io cannot be disabled**
  Previously, setting the value of **page-max-concurrent-io** to **-1** to remove the upper limit on the number of concurrent reads allowed on paging did not work. Instead, the broker continued to use the default value, or the previously configured value, if you had changed it from the default. This issue is now resolved.

- ENTMQBR-3797 - **Activation failure can result in zombie broker**
  In prior releases, if a live-backup broker group was configured to use shared store high availability, the live broker might fail to properly activate after a restart, but would continue to hold the journal lock. For example, this issue might occur if the live-backup group was using a network file system (NFS), and the NFS was unexpectedly stopped or removed, causing the live broker to restart. This situation resulted in a non-functional broker that couldn't service clients but which prevented the backup broker from activating. This issue is now resolved.

- ENTMQBR-3798 - **JDBC XML config can't use custom password codec**
  Previously, if you specified a masked password for the **jdbc-password** parameter and a custom codec for the **password-codec** parameter in your broker configuration, the broker always used the default **org.apache.activemq.artemis.utils.DefaultSensitiveStringCodec** codec to decode the password. This issue is now resolved.

- ENTMQBR-3812 - **Potential deadlock when destroying a queue and depaging concurrently**
  Previously, if a queue was destroyed (for example when a non-durable consumer closed its connection) and depaging was concurrently happening, this might result in a deadlock condition. This issue is now resolved.

- ENTMQBR-3813 - **Null pointer exception on queue update**
  Previously, if you tried to update a queue that was originally created without a filter, the broker might display a null pointer exception (NPE). This issue is now resolved.

- ENTMQBR-3841 - **Concurrent Jolokia operations can incorrectly update artemis-roles.properties or artemis-users.properties**
  Previously, if multiple, concurrent Jolokia operations that manipulated users and roles or permissions took place on the broker, the broker might incorrectly update or remove some data in the **artemis-roles.properties** or **artemis-users.properties** configuration files. This issue is now resolved.

- ENTMQBR-3880 - **Destination header replaced for wildcard address during paging**
  Previously, before a message was stored, the broker set the address field on the message to reflect the page store name. If the message was first paged for a consumer that subscribed using a wildcard address expression, this resulted in an incorrect destination name header that other consumers could not process. This issue is resolved. The broker no longer changes the message content when writing the message to the page store. This leaves the original target address intact.

- ENTMQBR-4034 - **LVQ broken after restart**
  Previously, after a broker restart, an existing message in a last-value queue was not replaced with a new message that was sent to the queue with the same last-value key. This issue is now resolved.

- ENTMQBR-4143 - **AMQ Broker Operator: Type mismatch forpageSizeBytes property between CRD and Operator**
  In the previous release, if you added an address settings configuration to the Custom Resource (CR) instance for a broker deployment (that is, by adding an **addressSettings.addressSetting** section), you could not include the **pageSizeBytes** property. If you included this property and specified a value, the Operator either failed to process the CR, or it processed the CR but could start any brokers. This issue is now resolved.

- ENTMQBR-4144 - **AMQ Broker Operator: Address setting redeliveryCollisionAvoidanceFactor cannot be specified**
  In the previous release, if you added an address settings configuration to the Custom Resource (CR) instance for a broker deployment (that is, by adding an **addressSettings.addressSetting** section), you could not use the **redeliveryCollisionAvoidanceFactor** property. If you included this property and specified a value, the Operator failed to process the CR. This issue is now resolved.

- ENTMQBR-4145 - **AMQ Broker Operator: Address settingautoCreateJmsTopics cannot be specified**
  In the previous release, if you added an address settings configuration to the Custom Resource (CR) instance for a broker deployment (that is, by adding an **addressSettings.addressSetting** section), you could not use the **autoCreateJmsTopics** property. If you included this property

and specified a value, the Operator processed the CR but failed to include the property in the resulting broker configuration. This issue is now resolved.

- ENTMQBR-4146 - **Broker fails to start whendefault-group-rebalance-pause-dispatch property is specified in address settings**
  Previously, if you configured the **address-setting** element of the **broker.xml** configuration file to set the value of the **default-group rebalance-pause-dispatch** property to **true**, the broker could not start.

  This issue also occurred for broker deployments on OpenShift Container Platform. Specifically, if you added an address settings configuration to the Custom Resource (CR) instance for the broker deployment (that is, by adding an **addressSettings.addressSetting** section) and set the value of the **defaultGroupRebalancePauseDispatch** property to **true**, the brokers in the deployment could not start.

  This issue is now resolved.

- ENTMQBR-4159 - **AMQ Broker Operator: Route not created for STOMP protocol**
  In prior releases, if you defined an acceptor to use the STOMP protocol, but did not specify a port for the acceptor to use, the Operator failed to create a Service and Route for the acceptor. This issue is now resolved.

- ENTMQBR-4195 - **Deleted scheduled message reappears after AMQ broker restart**
  In prior releases, if you used the management API to delete a scheduled message, the message was removed from memory but not from storage. This caused the message to reappear when you restarted the broker. This issue is now resolved.

- ENTMQBR-4263 - **A message becoming large through DLQ shuts down the broker**
  Previously, if a message for given protocol was close to the configured large message size for that protocol and the broker attempted to deliver the message to a dead letter queue, the broker might unexpectedly shut down. This issue is now resolved.

Additional resources

- For a complete list of issues that have been fixed in the AMQ Broker 7.8 release, see AMQ Broker 7.8.0 Fixed Issues.

CHAPTER 6. FIXED COMMON VULNERABILITIES AND EXPOSURES

This section details Common Vulnerabilities and Exposures (CVEs) fixed in the AMQ Broker 7.8 release.

- ENTMQBR-3755 - CVE-2020-13932 - mqtt-client: activemq: remote XSS in web console diagram plugin [amq-7]
- ENTMQBR-3382 - CVE-2015-5183 Hawtio: HTTPOnly and Secure attributes not set on cookies [amq-7]
- ENTMQBR-4037 - CVE-2019-12749 - DBusServer DBUS_COOKIE_SHA1 authentication bypass
- ENTMQBR-4068 - CVE-2019-9827 - hawtio: server side request forgery via initial /proxy/ substring of a URI [amq-7.7.0]
- ENTMQBR-4158 - CVE-2020-27216 - jetty: local temporary directory hijacking vulnerability [amq-7]

CHAPTER 7. KNOWN ISSUES

This section describes known issues in AMQ Broker 7.8.

- ENTMQBR-17 - **AMQ222117: Unable to start cluster connection**
  A broker cluster may fail to initialize properly in environments that support IPv6. The failure is due to a **SocketException** that is indicated by the log message **Can't assign requested address**. To work around this issue, set the **java.net.preferIPv4Stack** system property to **true**.

- ENTMQBR-463 - **Attributes in clustering settings have order restrictions. Would be nice to either have better error message or simply ignore the order**
  Currently the sequence of the elements in the cluster connection configuration has to be in a specific order. The workaround is to adhere to the order in the configuration schema.

- ENTMQBR-520 - **Receiving from address named the same as a queue bound to another address should not be allowed**
  A queue with the same name as an address must only be assigned to address. Creating a queue with the same name as an existing address, but bound to an address with a different name, is an invalid configuration. Doing so can result in incorrect messages being routed to the queue.

- ENTMQBR-522 - **Broker running on windows write problems with remove temp files when shutting down**
  On Windows, the broker does not successfully clean up temporary files when it shuts down. This issue causes the shutdown process to be slow. In addition, temporary files not deleted by the broker accumulate over time.

- ENTMQBR-569 - **Conversion of IDs from OpenWire to AMQP results in sending IDs as binary**
  When communicating cross-protocol from an A-MQ 6 OpenWire client to an AMQP client, additional information is encoded in the application message properties. This is benign information used internally by the broker and can be ignored.

- ENTMQBR-599 - **Define truststore and keystore by Artemis cli**
  Creating a broker instance by using the **--ssl-key**, **--ssl-key-password**, **--ssl-trust**, and **--ssl-trust-password** parameters does not work. To work around this issue, set the corresponding properties manually in **bootstrap.xml** after creating the broker.

- ENTMQBR-636 - **Journal breaks, causingJavaNullPointerException, under perf load (mpt)**
  To prevent IO-related issues from occurring when the broker is managing heavy loads, verify that the JVM is allocated with enough memory and heap space. See the section titled "Tuning the VM" in the Performance Tuning chapter of the ActiveMQ Artemis documentation.

- ENTMQBR-648 - **JMS Openwire client is unable to send messages to queue with defined purgeOnNoConsumer or queue filter**
  Using an A-MQ 6 JMS client to send messages to an address that has a queue with **purgeOnNoConsumer** set to **true** fails if the queue has no consumers. It is recommended that you do not set the **purgeOnNoConsumer** option when using A-MQ 6 JMS clients.

- ENTMQBR-652 - **List of knownamq-jon-plugin bugs**
  This version of **amq-jon-plugin** has known issues with the MBeans for broker and queue.

  Issues with the broker MBean:

  - Closing a connection throws **java.net.SocketTimeoutException** exception
  - **listSessions()** throws **java.lang.ClassCastException**

  - Adding address settings throws **java.lang.IllegalArgumentException**
  - **getConnectorServices()** operation cannot be found
  - **listConsumersAsJSON()** operation cannot be found
  - **getDivertNames()** operation cannot be found
  - Listing network topology throws **IllegalArgumentException**
  - Remove address settings has wrong parameter name

  Issues with the queue MBean:

  - **expireMessage()** throws argument type mismatch exception
  - **listDeliveringMessages()** throws **IllegalArgumentException**
  - **listMessages()** throws **java.lang.Exception**
  - **moveMessages()** throws **IllegalArgumentException** with error message argument type mismatch
  - **removeMessage()** throws **IllegalArgumentException** with error message argument type mismatch
  - **removeMessages()** throws exception with error Can't find operation removeMessage with 2 arguments
  - **retryMessage()** throws argument type mismatch **IllegalArgumentException**

- ENTMQBR-655 - **[AMQP] Unable to send message whenpopulate-validated-user is enabled**
  The configuration option **populate-validated-user** is not supported for messages produced using the AMQP protocol.

- ENTMQBR-738 - **Unable to build AMQ 7 examples offline with provided offline repo**
  You cannot build the examples included with AMQ Broker in an offline environment. This issue is caused by missing dependencies in the provided offline Maven repository.

- ENTMQBR-897 - **Openwire client/protocol issues with special characters in destination name**
  Currently AMQ OpenWire JMS clients cannot access queues and addresses that include the following characters in their name: comma (','), hash ('#'), greater than ('>'), and whitespace.

- ENTMQBR-944 - **[A-MQ7, Hawtio, RBAC] User gets no feedback if operation access was denied by RBAC**
  The console can indicate that an operation attempted by an unauthorized user was successful when it was not.

- ENTMQBR-1498 - **Diagram in management console for HA (replication, sharedstore) does not reflect real topology**
  If you configure a broker cluster with some extra, passive slaves, the cluster diagram in the web console does not show these passive slaves.

- **ENTMQBR-1848 - "javax.jms.JMSException: Incorrect Routing Type for queue, expecting: ANYCAST" occurs when qpid-jms client consumes a message from a multicast queue as javax.jms.Queue object with FQQN**
  Currently, sending a message by using the Qpid JMS client to a multicast queue by using FQQN (fully qualified queue name) to an address that has multiple queues configured generates an error message on the client, and the message cannot be sent. To work around this issue, modify the broker configuration to resolve the error and unblock the client.

- **ENTMQBR-1875 - [AMQ 7, ha, replicated store] backup broker appear not to go "live" or shutdown after - ActiveMQIllegalStateException errorType=ILLEGAL_STATE message=AMQ119026: Backup Server was not yet in sync with live**
  Removing the paging disk of a master broker while a backup broker is trying to sync with the master broker causes the master to fail. In addition, the backup broker cannot become live because it continues trying to sync with the master.

- **ENTMQBR-2068 - some messages received but not delivered during HA fail-over, fail-back scenario**
  Currently, if a broker fails over to its slave while an OpenWire client is sending messages, messages being delivered to the broker when failover occurs could be lost. To work around this issue, ensure that the broker persists the messages before acknowledging them.

- **ENTMQBR-2452 - Upgraded broker AMQ 7.3.0 from AMQ 7.2.4 on Windows cannot log**
  If you intend to upgrade a broker instance from 7.2.4 to 7.3.0 on Windows, logging will not work unless you specify the correct log manager version during your upgrade process. For more information, see Upgrading from 7.2.x to 7.3.0 on Windows .

- **ENTMQBR-2470 - [AMQ7, openwire,redelivery] redelivery counter for message increasing, if consumer is closed without consuming any messages**
  If a broker sends a message to an Openwire consumer, but the consumer is closed before consuming the message, the broker wrongly increments the redelivery count for the pending message. If the number of occurrences of this behavior exceeds the value of the **max-delivery-attempts** configuration parameter, the broker sends the message to the dead letter queue (DLQ) or drops the message, based on your configuration. This issue does not affect other protocols, such as the Core protocol.

- **ENTMQBR-2593 - broker does not set message ID header on cross protocol consumption**
  A Qpid JMS client successfully retrieves a message ID only if the message was produced by another Qpid JMS client. If the message was produced by a Core JMS or OpenWire client, the Qpid JMS client cannot read the message ID.

- **ENTMQBR-2678 - After isolated master is live again it is unable to connect to the cluster**
  In a cluster of three or more live-backup groups that is using the replication high availability (HA) policy, the live broker shuts down when its replication connection fails. However, when the replication connection is restored and the original live broker is restarted, the broker is sometimes unable to rejoin the broker cluster. To enable the original live broker to rejoin the cluster, first stop the new live (original backup) broker, restart the original live broker, and then restart the original backup broker.

- **ENTMQBR-2928 - Broker Operator unable to recover from CR changes causing erroneous state**
  If the AMQ Broker Operator encounters an error when applying a Custom Resource (CR) update, the Operator does not recover. Specifically, the Operator stops responding as expected to further updates to your CRs.

  For example, say that a misspelling in the value of the **image** attribute in your main broker CR causes broker Pods to fail to deploy, with an associated error message of **ImagePullBackOff**. If you then fix the misspelling and apply the CR changes, the Operator does not deploy the specified number of broker Pods. In addition, the Operator does not respond to any further CR changes.

To work around this issue, you must delete the CRs that you originally deployed, before redeploying them. To delete an existing CR, use a command such as **oc delete -f <CR name>**.

- ENTMQBR-2942 - **Pod #0 tries to contact non-existent Pods**
  If you change the **size** attribute of your Custom Resource (CR) instance to scale down a broker deployment, the first broker Pod in the cluster can make repeated attempts to connect to the drainer Pods that started up to migrate messages from the brokers that shut down, before they shut down themselves. To work around this issue, follow these steps:

  1) Scale your deployment to a single broker Pod.

  2) Wait for all drainer Pods to start, complete message migration, and then shut down.

  3) If the single remaining broker Pod has log entries for an “unknown host exception”, scale the deployment down to zero broker Pods, and then back to one.

  4) When you have verified that the single remaining broker Pod is not recording exception-based log entries, scale your deployment back to its original size.

- ENTMQBR-3131 - **Topology Fails to Update correctly for Backup Brokers when Master is Killed**
  When a live broker fails in a cluster with more than four live-backup pairs, the live brokers, including the newly-elected live broker, all correctly report the updated topology. However, the remaining backup brokers might show the wrong topology in the following ways:

  - If a backup broker has failed over in place of the failed live broker, the remaining backup brokers show this backup broker twice in the topology.

  - If a backup broker has not yet failed over in place of the failed live broker, the remaining backup brokers still show the failed live broker in the topology.

  To work around this issue, ensure that the first **connector-ref** element in the **cluster-connection** > **static-connectors** configuration of each backup broker specifies the expected live broker.

- ENTMQBR-3604 - **Enabling Pooling for the LDAP Login Module Causes Shutdown to Hang**
  If you enable connection pooling for an LDAP provider (that is, by setting **connectionPool** to **true** in the **LDAPLoginModule** section of the **login.config** configuration file), this can cause connections to the LDAP provider to remain open indefinitely, even when you stop the broker clients. As a result, if you try to shut down the broker in the normal way, the broker does not shut down. Instead, you need to use a Linux command such as **SIGKILL** to terminate the broker process. This situation occurs even if you specify a pool timeout in the JVM arguments for the broker (for example, **-Dcom.sun.jndi.ldap.connect.pool.timeout=30000**) and there are no active clients when you try to shut down the broker.

  To work around this issue, set a value for the **connectionTimeout** property in the **LDAPLoginModule** section of the **login.config** configuration file. When connection pooling has been requested for a connection, the **connectionTimeout** property specifies the maximum time that the broker waits for a connection when the maximum pool size has already been reached and all connections in the pool are in use. For more information, see Using LDAP for Authentication in *Configuring AMQ Broker*.

- ENTMQBR-3653 - **NPE thrown if metrics plugin is not configured and the metrics web context is invoked**
  If the /**metrics** web context on a broker is invoked, but the metrics plugin has not yet been configured, the broker displays a null pointer exception. For more information about configuring the *Prometheus* metrics plugin for AMQ Broker, see Enabling the Prometheus plugin for AMQ Broker (on-premise broker deployments) or Enabling the Prometheus plugin for a running broker deployment (OpenShift broker deployments).

- ENTMQBR-3724 - **OperatorHub displays inappropriate variant of AMQ Broker Operator**
  If you use OperatorHub to deploy the AMQ Broker Operator on OpenShift Container Platform 4.5 or earlier, OperatorHub displays a variant of the Operator that is not appropriate for your host platform. This makes it possible to select the incorrect Operator variant. In particular, regardless of your host platform, OperatorHub displays both the **Red Hat Integration - AMQ Broker** Operator (the Operator for OpenShift Container Platform) and the **AMQ Broker** Operator (the Operator for OpenShift Container Platform on IBM Z).

  To work around this issue, select the Operator variant that is appropriate to your platform, as described above. Alternatively, install the Operator using the OpenShift command-line interface (CLI).

  In OpenShift Container Platform 4.6, this issue is resolved. OperatorHub displays **only** the Operator variant that corresponds to your host platform.

- ENTMQBR-3846 - **MQTT client does not reconnect on broker restart**
  When you restart a broker, or a broker fails over, the active broker does not restore connections for previously-connected MQTT clients. To work around this issue, to reconnect an MQTT client, you need to manually call the **subscribe()** method on the client.

- ENTMQBR-4023 - **AMQ Broker Operator: Pod Status pod names do not reflect the reality**
  For an Operator-based broker deployment in a given OpenShift project, if you use the **oc get pod** command to list the broker Pods, the ordinal values for the Pods start at **0**, for example, **amq-operator-test-broker-ss-0**. However, if you use the **oc describe** command to get the status of broker Pods created from the **activemqartmises** Custom Resource (that is, **oc describe activemqartemises**), the Pod ordinal values incorrectly start at **1**, for example, **amq-operator-test-broker-ss-1**. There is no way to work around this issue.

- ENTMQBR-4127 - **AMQ Broker Operator: Route name generated by Operator might be too long for OpenShift**
  For each broker Pod in an Operator-based deployment, the default name of the Route that the Operator creates for access to the AMQ Broker management console includes the name of the Custom Resource (CR) instance, the name of the OpenShift project, and the name of the OpenShift cluster. For example, **my-broker-deployment-wconsj-0-svc-rte-my-openshift-project.my-openshift-domain**. If some of these names are long, the default Route name might exceed the limit of 63 characters that OpenShift enforces. In this case, in the OpenShift Container Platform web console, the Route shows a status of **Rejected**.

  To work around this issue, use the OpenShift Container Platform web console to manually edit the name of the Route. In the console, click the Route. On the **Actions** drop-down menu in the top-right corner, select **Edit Route**. In the YAML editor, find the **spec.host** property and edit the value.

- ENTMQBR-4140 - **AMQ Broker Operator: Installation becomes unusable if storage.size is improperly specified**
  If you configure the **storage.size** property of a Custom Resource (CR) instance to specify the size of the Persistent Volume Claim (PVC) required by brokers in a deployment for persistent storage, the Operator installation becomes unusable if you do not specify this value properly.

For example, suppose that you set the value of **storage.size** to **1** (that is, without specifying a unit). In this case, the Operator cannot use the CR to create a broker deployment. In addition, even if you remove the CR and deploy a new version with **storage.size** specified correctly, the Operator still cannot use this CR to create a deployment as expected.

To work around this issue, first stop the Operator. In the OpenShift Container Platform web console, click **Deployments**. For the Pod that corresponds to the AMQ Broker Operator, click the **More options** menu (three vertical dots). Click **Edit Pod Count** and set the value to **0**. When the Operator Pod has stopped, create a new version of the CR with **storage.size** correctly specified. Then, to restart the Operator, click **Edit Pod Count** again and set the value back to **1**.

- ENTMQBR-4141 **- AMQ Broker Operator: Increasing Persistent Volume size requires manual involvement even after recreating Stateful Set**
  If you try to increase the size of the Persistent Volume Claim (PVC) required by brokers in a deployment for persistent storage, the change does not take effect without further manual steps. For example, suppose that you configure the **storage.size** property of a Custom Resource (CR) instance to specify an initial size for the PVC. If you modify the CR to specify a *different* value of **storage.size**, the existing brokers continue to use the original PVC size. This is the case even if you scale the deployment down to zero brokers and then back up to the original number. However, if you scale the size of the deployment up to add additional brokers, the new brokers use the new PVC size.

  To work around this issue, and ensure that all brokers in the deployment use the same PVC size, use the OpenShift Container Platform web console to expand the PVC size used by the deployment. In the console, click **Storage → Persistent Volume Claims**. Click your deployment. On the **Actions** drop-down menu in the top-right corner, select **Expand PVC** and enter a new value.

CHAPTER 8. IMPORTANT LINKS

- Red Hat AMQ Broker 7.7 Release Notes
- Red Hat AMQ Broker 7.6 Release Notes
- Red Hat AMQ Broker 7.1 to 7.5 Release Notes (aggregated)
- Red Hat AMQ 7 Supported Configurations
- Red Hat AMQ 7 Component Details

Revised on 2022-03-15 13:56:47 UTC