Red Hat AMQ 2021.Q3

Release Notes for Red Hat AMQ Broker 7.9

Release Notes for AMQ Broker

Last Updated: 2022-07-07

Red Hat AMQ 2021.Q3 Release Notes for Red Hat AMQ Broker 7.9

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 7.9 release.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE . . . 3

CHAPTER 1. ENHANCEMENTS . . . 4

CHAPTER 2. REMOVED FEATURES . . . 6

CHAPTER 3. DEPRECATED FEATURES . . . 7

CHAPTER 4. TECHNOLOGY PREVIEW . . . 8

CHAPTER 5. FIXED ISSUES . . . 9

CHAPTER 6. FIXED COMMON VULNERABILITIES AND EXPOSURES . . . 10

CHAPTER 7. KNOWN ISSUES . . . 11

CHAPTER 8. IMPORTANT LINKS . . . 15

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message .

CHAPTER 1. ENHANCEMENTS

This section describes a highlighted set of enhancements and new features in AMQ Broker 7.9. For a complete list of enhancements in the release, see AMQ Broker 7.9.0 Enhancements.

NOTE

If you require the latest AMQ Broker Long Term Support (LTS) release version, see AMQ Broker 7.8.

AMQP server connections

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 and create elements on those connections. This feature is implemented using the **<broker-connection>** element as described in Configuring AMQ Broker.

Operator supports watching all or multiple namespaces

In previous releases, you installed the AMQ Broker Operator in every namespace you required a broker deployment. Starting in 7.9, the AMQ Broker Operator supports watching all or multiple namespaces for broker custom resources. For more information, see Deploying AMQ Broker on OpenShift.

NOTE

If you have already installed a previous version of the AMQ Broker Operator in a namespace on your cluster, Red Hat recommends that you do not install the AMQ Broker Operator 7.9 version to watch that namespace to avoid potential conflicts.

Temporary queue namespace

In AMQ Broker 7.9, you can specify a **temporary-queue-namespace** in the **broker.xml** configuration file. You can then specify address settings that match the namespace and the broker applies those settings to all temporary queues. If a temporary queue namespace does not exist, temporary queues use the same address settings configuration as other queues. For more information, see Applying specific address settings to temporary queues in *Configuring AMQ Broker*.

Operator channels

The AMQ Broker Operator **Red Hat Integration - AMQ Broker for RHEL 8 (Multiarch)** is available with the following channels:

- **7.x** - This channel will install **7.9** and update to **7.10** at some point in the future. If you want to install **7.10** now, you can subscribe to the **7.10.x** channel.
- **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.

Hosts verified by default

The default value for **verifyHost** has changed from **false** to **true** when applied to connectors. All inter-broker connections now verify hosts by default. The default value for acceptors continues to be **false**.

Enabling the Prometheus plugin using a CR

You can enable the Prometheus plugin on OpenShift using a CR in addition to enabling the plugin using an environment variable. Both options are described in Deploying AMQ Broker for On-Premise .

CHAPTER 2. REMOVED FEATURES

The following features are no longer available in 7.9:

OpenJ9 images

The IBM Z (s390x) and IBM Power Systems (ppc64le) images based on OpenJ9 are no longer supported. Newer images for those platforms are based on OpenJDK.

Template based installations

The use of application templates for deploying AMQ Broker on OpenShift Container Platform was deprecated in previous releases and is now removed. Use the AMQ Broker Operator as described in Deploying AMQ Broker on OpenShift Container Platform using the AMQ Broker Operator .

OpenShift Container Platform 3.11

Deploying AMQ Broker on OpenShift Container Platform 3.11 is no longer supported. AMQ Broker is supported on OpenShift Container Platform 4.6, 4.7 or 4.8.

RHEL 7 based images

All deployments of AMQ Broker on OpenShift Container Platform now use RHEL 8 based images.

Documentation

The **Using JON with AMQ Broker**guide is no longer published as part of the AMQ Broker documentation. However, you can still access the last published version as part of the AMQ Broker 7.8 documentation.

CHAPTER 3. DEPRECATED FEATURES

This section describes features that are supported, but have been deprecated from AMQ Broker.

OpenWire protocol

Starting in 7.9, the OpenWire protocol is a deprecated feature. If you are creating a new AMQ Broker-based system, use one of the other supported protocols. This feature will be removed in a future release.

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*..

CHAPTER 4. TECHNOLOGY PREVIEW

This section describes Technology Preview features in AMQ Broker 7.9.

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.

Quorum voting improvements

In previous versions of AMQ Broker you needed to configure at least three live-backup pairs to use quorum voting to avoid having two live brokers when using replication high availability (HA) policy. Starting in 7.9, you can configure failover to use Apache Curator and Apache ZooKeeper to provide quorum voting using two brokers. For information about using this feature, see High Availability and Failover in the Apache ActiveMQ Artemis documentation.

Client connection balancing improvements

In previous releases, there was no method to balance client connections server-side. Starting in 7.9, you can specify pools of brokers and policies for balancing client connections. For example, you can specify a **LEAST_CONNECTIONS** policy that ensures that clients are redirected to brokers with the fewest active connections. For information about using this feature, see Broker Balancers 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

For a complete list of issues that have been fixed in the release, see AMQ Broker 7.9.0 Fixed Issues and AMQ Broker - 7.9.x Resolved Issues .

CHAPTER 6. FIXED COMMON VULNERABILITIES AND EXPOSURES

This section details Common Vulnerabilities and Exposures (CVEs) fixed in the AMQ Broker 7.9 release.

- ENTMQBR-4071 - **CVE-2020-13956 httpclient: apache-httpclient: incorrect handling of malformed authority component in request URIs**
- ENTMQBR-4677 - **CVE-2021-21290 netty: Information disclosure via the local system temporary directory**
- ENTMQBR-4775 - **CVE-2020-27223 jetty: request containing multiple Accept headers with a large number of "quality" parameters may lead to DoS**
- ENTMQBR-4779 - **CVE-2021-3425 broker: Red Hat AMQ Broker: discloses JDBC username and password in the application log file**
- ENTMQBR-4795 - **CVE-2021-21295 netty: possible request smuggling in HTTP/2 due missing validation**
- ENTMQBR-4829 - **CVE-2021-21409 netty: Request smuggling via content-length header**
- ENTMQBR-4907 - **CVE-2021-28163 jetty-server: jetty: Symlink directory exposes webapp directory contents**
- ENTMQBR-4911 - **CVE-2021-28165 jetty-server: jetty: Resource exhaustion when receiving an invalid large TLS frame**
- ENTMQBR-4912 - **CVE-2021-28164 jetty-server: jetty: Ambiguous paths can access WEB-INF**
- ENTMQBR-4960 - **CVE-2021-29425 commons-io: apache-commons-io: Limited path traversal in Apache Commons IO 2.2 to 2.6**
- ENTMQBR-5118 - **CVE-2021-28169 jetty-server: jetty: requests to the ConcatServlet and WelcomeFilter are able to access protected resources within the WEB-INF directory**
- ENTMQBR-5165 - **CVE-2021-34428 jetty-server: jetty: SessionListener can prevent a session from being invalidated breaking logout**
- ENTMQBR-5229 - **CVE-2021-20289 resteasy-jaxrs: resteasy: Error message exposes endpoint class information**
- ENTMQBR-5250 - **CVE-2021-34429 jetty-server: jetty: crafted URIs allow bypassing security constraints**
- ENTMQBR-5398 - **CVE-2021-3763 AMQ Broker 7: Incorrect privilege in Management Console**

CHAPTER 7. KNOWN ISSUES

This section describes known issues in AMQ Broker 7.9.

- **ENTMQBR-5749 - Remove unsupported operators that are visible in OperatorHub**
  Only the Operators and Operator channels mentioned in Deploying the Operator from OperatorHub are supported. For technical reasons associated with Operator publication, other Operator and channels are visible in the OperatorHub and should be ignored. For reference, the following list shows which Operators are visble, but not supported:

  - Red Hat Integration - AMQ Broker LTS - all channels
  - Red Hat Integration - AMQ Broker - alpha, current, and current-76

- **ENTMQBR-5615 - Unexpected breaking change in artemis.profile prevents "init container image" approach**
  If you use the JVM option **-Dhawtio.role** to set user roles as part of the $JAVA_ARGS section of the **artemis_profile** file, users might not be able to access the broker console.

  This issue is caused by a new property **HAWTIO_ROLE** which overrides any values set by **-Dhawtio.role**. To workaround this problem, set the appropriate roles using the **HAWTIO_ROLE** property in the **etc/artemis.profile** file.

- **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-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-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-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-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-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-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 7.8 Release Notes
- 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 2022-07-07 11:41:10 UTC