



Red Hat AMQ 7.7

Release Notes for Red Hat AMQ Broker 7.7

Release Notes for AMQ Broker

Red Hat AMQ 7.7 Release Notes for Red Hat AMQ Broker 7.7

Release Notes for AMQ Broker

Legal Notice

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

These release notes contain the latest information about new features, enhancements, fixes, and issues contained in the AMQ Broker 7.7 release.

Table of Contents

CHAPTER 1. ENHANCEMENTS	3
CHAPTER 2. DEPRECATED FEATURES	5
CHAPTER 3. TECHNOLOGY PREVIEW	6
CHAPTER 4. FIXED ISSUES	7
CHAPTER 5. KNOWN ISSUES	10
CHAPTER 6. IMPORTANT LINKS	17

CHAPTER 1. ENHANCEMENTS

- **ENTMQBR-1187 - Automatic creation of expiry and dead letter resources**

In AMQ Broker 7.7, you can configure the broker to automatically create resources (that is, addresses and queues) to handle expired or undelivered messages. For more information, see [Creating Expiry Resources Automatically](#) and [Creating Dead Letter Resources Automatically](#) in *Configuring AMQ Broker*.
- **ENTMQBR-2411 - Filtering based on AMQP message properties and annotations**

Before the broker moves an expired or undelivered AMQP message to an expiry or dead letter queue that you have configured, the broker now applies annotations and properties to the message. A client can use a filter based on these properties or annotations to select which messages to consume. For more information, see [Filtering AMQP Messages Based on Annotations](#) in *Configuring AMQ Broker*.
- **ENTMQBR-2736 - New resource audit logger**

AMQ Broker 7.7 adds a new resource audit logger. The resource audit logger logs events related to authentication, creation or deletion of broker resources from JMX or the AMQ management console, and browsing of messages in the management console. For more information, see [Logging](#) in *Configuring AMQ Broker*.
- **ENTMQBR-2883 - Configuring handling of large AMQP messages**

In AMQ Broker 7.7, you can configure an AMQP acceptor to handle an AMQP message larger than a specified size as a large message. For more information, see [Configuring AMQP acceptors for large message handling](#) in *Configuring AMQ Broker*.
- **ENTMQBR-2919 - Setting ranges of message expiry values for addresses**

In AMQ Broker 7.7, you can specify minimum and maximum message expiry values for an address or set of addresses. For more information, see [Configuring Message Expiry](#) in *Configuring AMQ Broker*.
- **ENTMQBR-3046 - Performing broker health checks**

In AMQ Broker 7.7, you can use a command called **artemis check** to perform various health checks on a broker and its queues. For more information about the checks that you can run, use the **help** function. For example:

```
$ <broker-instance-dir>/bin/artemis help check
```
- **ENTMQBR-3134 - AMQ Broker OpenShift Operator: Configuring PVC size for broker deployments**

In AMQ Broker 7.7, you can use the Custom Resource (CR) instance for an Operator-based deployment to configure the size of the Persistent Volume Claim (PVC) required by brokers in the deployment for persistent storage. For more information, see [Configuring broker storage requirements](#) in *Deploying AMQ Broker on OpenShift*.
- **ENTMQBR-3162 - Disabling metrics collection on individual addresses**

In AMQ Broker 7.7, if you have a metrics plugin configured, you can disable metrics collection for specific addresses. For more information, see [Disabling metrics collection for specific addresses](#) in *Managing AMQ Broker*.
- **ENTMQBR-3338 - AMQ Broker OpenShift Operator: Configuring resource limits for broker deployments**

In AMQ Broker 7.7, you can use the Custom Resource (CR) instance for an Operator-based deployment to configure CPU and memory limits and requests for the brokers in the deployment. For more information, see [Configuring resource limits and requests for Operator-](#)

[based broker deployments](#) in *Deploying AMQ Broker on OpenShift*.

- **ENTMQBR-3573 - Configuring JVM metrics collection**

In AMQ Broker 7.7, you can configure the broker to capture standard sets of metrics relating to the host Java Virtual Machine (JVM) for the broker. Specifically, you can specify whether the broker captures JVM metrics for Garbage Collection (GC), memory, and threads. For more information, see [Configuring the broker to collect JVM metrics](#) in *Managing AMQ Broker*.

- **ENTMQBR-3608 - AMQ Broker OpenShift Operator: Configuring address settings for broker deployments**

In AMQ Broker 7.7, you can use the Custom Resource (CR) instance for an Operator-based deployment to configure address settings. For more information, see [Configuring addresses and queues for Operator-based broker deployments](#) in *Deploying AMQ Broker on OpenShift*.

- **ENTMQBR-3746 - AMQ Broker OpenShift Operator: Configuring large message handling for AMQP messages**

In AMQ Broker 7.7, you can use the Custom Resource (CR) instance for an Operator-based deployment to configure large message handling for AMQP messages. For more information, see [Configuring large message handling for AMQP messages](#) in *Deploying AMQ Broker on OpenShift*.

- **ENTMQBR-4366 - Added AMQ Broker support for running within OpenShift Container Platform on IBM Z and IBM Power Systems**

AMQ Broker 7.7 adds broker container images that support RHEL 8 on IBM Z and IBM Power Systems. This added support enables you to create broker deployments within OpenShift Container Platform on IBM Z and IBM Power Systems. For more information, see [Deploying AMQ Broker on OpenShift Container Platform Using Application Templates](#).

CHAPTER 2. DEPRECATED FEATURES

- 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.
- 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](#).

CHAPTER 3. TECHNOLOGY PREVIEW

This section details Technology Preview features in AMQ Broker 7.7.



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](#).

- **Federated queues and addresses**

Federated queues and addresses enable you to transmit messages between brokers without the brokers being part of a cluster. For more information, see [Federation](#) in the Apache ActiveMQ Artemis documentation.

CHAPTER 4. FIXED ISSUES

- **ENTMQBR-1942** - [AMQ 7.2, shared store, scale down] **NullPointerException** exception when slave activates and tries to scale down
Previously, if you configured a slave broker to use a shared store and scaledown in a cluster, you could encounter an exception on the slave if the master broker went offline. This issue is resolved.
- **ENTMQBR-3103** - [AMQ7, AMQP, Openwire] **Issue consuming AMQP message using OpenWire consumer**
Previously, when a message was converted from OpenWire to AMQP and then back to OpenWire, the message could no longer be consumed by an OpenWire client. For example, this situation might occur when:
 - 1) An OpenWire client sent a message to the broker
 - 2) An AMQP client consumed the message
 - 3) The AMQP client returned the message to the broker
 - 4) An OpenWire consumer tried to consume the messageThis issue is now resolved.
- **ENTMQBR-3130** - **Broker Operator doesn't recreate Address after broker redeploy**
Previously, if you used a Custom Resource (CR) instance to create addresses in an Operator-based broker deployment, the Operator did not recreate those addresses if you removed and then redeployed the brokers. This issue is now resolved.
- **ENTMQBR-3137** - **Broker Operator does not automatically deploy CR-created addresses on scale-up**
In prior releases, if you used a Custom Resource (CR) instance to create addresses in an Operator-based broker deployment, and then scaled up the size of that deployment, the Operator did not automatically deploy existing addresses to the new broker instances. Instead, you needed to delete the address CR, before redeploying the CR for the scaled-up broker deployment. This issue is now resolved. When you scale up the size of a broker deployment, the Operator automatically deploys existing addresses to the new broker instances.
- **ENTMQBR-3228** - **Operator Pod not authenticating correctly when processing address CR**
In prior releases, if you used the AMQ Broker Operator to create a broker Pod that required login (that is, by setting **requireLogin** to **true** in the Custom Resource (CR) used for the deployment), any addresses you tried to create on the broker via a CR were not created. This issue is now resolved.
- **ENTMQBR-3328** - "FQQN support is limited to Anycast queues where the queue name equals the address" error occurs
In prior releases, if a client specified a fully qualified queue name (FQQN) when it used the **artemis producer** command to send messages to a queue with the **anycast** routing type, the broker connection returned an error if the queue and its address had different names, for example, **fqqn://exampleaddress::examplequeue**. This issue is now resolved.
- **ENTMQBR-3356** - **AMQP consumption stalls under during high message throughput**
Previously, if a broker in a live-backup group configured to use the replication high-availability (HA) policy experienced a very high throughput of AMQP messages (that is, a rate of 10,000 or more messages per second) message consumption might stall. In addition, the broker and

clients might display errors such as "Could not complete operations" (broker) and "Remote did not respond to a drain request in time" (client). This issue is now resolved.

- **ENTMQBR-3365 - Paging is Broken for AMQP**

By default, when you first get a message after the page cache has been evicted, the broker reads the entire page back into the cache. This behavior is specified by the **read-whole-page** configuration parameter, which has a default value of **true**. If you set the value of **read-whole-page** to **false**, the broker reads messages back into the page cache on a message-by-message basis instead.

Previously, reading messages back into the cache on a message-by-message basis **did not work** if you were using the AMQP protocol. If you were using AMQP, you needed to ensure that the value of **read-whole-page** was set to **true**. This issue is now resolved.

- **ENTMQBR-3370 - LegacyLDAPSecuritySettingPlugin ignores group changes**

Previously, if you made an update on an LDAP server, the **LegacyLDAPSecuritySettingPlugin** security settings plugin configured on the broker might not immediately detect the change. For example, if you added a new user to an existing group on the LDAP server, the plugin did not immediately authorize the new user with the same broker permissions as existing users in the same group. Instead, for a change such as this to take effect, you needed to restart the broker. This issue is now resolved.

- **ENTMQBR-3372 - Broker created using Operator fails to start if Base64-encoded keyStorePassword and trustStorePassword values contain "/"**

Previously, a broker instance created using the Operator failed to start if Base64-encoded **keyStorePassword** and **trustStorePassword** values that you specified when configuring Transport Layer Security (TLS) on the broker contained a "/" (that is, a forward slash) character. This issue is now resolved.

- **ENTMQBR-3380 - AMQP shared non-durable queues are not being deleted same as Core**

Previously, the broker did not automatically delete a shared, non-durable queue created by an AMQP producer if the queue had at least one message in it, even if the queue had no connected consumers. This issue is resolved. The broker now marks a shared, non-durable AMQP queue as *temporary*, in the same way that it does for the Core protocol. This means that the broker automatically deletes such a queue when there are no connected consumers.

- **ENTMQBR-3383 - Deletion of address custom resource instance removes address and messages**

Previously, if you deleted the Custom Resource (CR) instance used to create addresses in a broker deployment, this also removed existing addresses and any associated messages for all brokers in the deployment. This issue is resolved. A new configuration item called **removeFromBrokerOnDelete** in the address CR enables you to specify whether the Operator removes existing addresses in your broker deployment when you delete the underlying address CR. By default, **removeFromBrokerOnDelete** is set to **false**, which means that existing addresses are **not** removed.

- **ENTMQBR-3386 - [AMQ7, message expiry, auto-delete] auto-created queue may not be auto-deleted when messages expire**

Previously, the broker did not delete an automatically-created queue in the case where no consumers ever connected to the queue and all messages on the queue had expired. This issue is resolved. The broker deletes the automatically-created queue in this situation.

- **ENTMQBR-3409 - JMX/Jolokia addSecuritySettings - permissions are not processed until broker restart**

Previously, if you used Jolokia to dynamically set security settings for an address, the broker did not detect the update. Instead, you needed to restart the broker for the changes to take effect. This issue is now resolved.

- **ENTMQBR-3432 - OpenWire consumption stalls under during high message throughput**
Previously, if a broker in a live-backup group configured to use the replication high-availability (HA) policy experienced a very high throughput of OpenWire messages (that is, a rate of 10,000 or more messages per second) message consumption might stall. In addition, the broker and clients might display errors such as "Could not complete operations" (broker) and "Remote did not respond to a drain request in time" (client). This issue is now resolved.
- **ENTMQBR-3490 - The createSession() method throws java.lang.NullPointerException**
Previously, if you used `javax.jms.Connection.createSession()` to create a session on a connection, you could see a null pointer exception. Specifically, this exception could occur if multiple threads were using the connection, and one thread closed the connection while the `createSessionInternal` method concurrently tried to create the session on a separate thread. This issue is resolved. Now, there are concurrency protections that prevent this type of exception from occurring.
- **ENTMQBR-3597 - Statefulset recreated on startup**
Previously, if you deployed a Custom Resource (CR) instance to create an Operator-based broker deployment, the Operator created a StatefulSet for the deployment, immediately removed the StatefulSet, and then created it one more time. As a result of this issue, the broker Pods created for the deployment started, immediately terminated, and then started again. This issue is now resolved.

CHAPTER 5. KNOWN ISSUES

This section details known issues in AMQ Broker 7.7.

- **ENTMQBR-17 - AMQ22217: 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 queuefilter**
 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 wherpopulate-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-1815 - Hawtio view changes on automated refresh**
When automatic refresh is enabled, the Hawtio console updates the screen every 5 seconds.

Automatic refreshes also change the view to the **Attributes** screen, causing you to lose focus from any other screen you are viewing. To work around this issue, select **Preferences** in the top-right corner of the console. In the **Update rate** drop-down list, select **No refreshes**.

- **ENTMQBR-1848** - "javax.jms.JMSEException: 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-2890** - Infrequently, creating a CR instance with size $n > 1$ causes the the n th broker pod to start and immediately restart once

If you use a Custom Resource (CR) instance to deploy a broker cluster via the AMQ Broker Operator, the final broker Pod in the deployment (as determined by the **size** attribute of your

CR) starts and then immediately restarts one time before it is usable. To work around this issue, incrementally increase the **size** attribute of your CR to reach the desired cluster size. First, set **size** to **1** and wait for the broker Pod to start. Then, set **size** to **2**, wait for the Pod to start, and so on. In this case, none of the broker Pods undergo a restart before they are usable.

- **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-3617 - User information on the hawtio console for durable address is null**

When a consumer creates a shared, durable address, the AMQ Management Console might show the associated user as **null**. There is no way to work around this issue.

- **ENTMQBR-3653 - NPE thrown if metrics plugin is not configured and the metrics web context is invoked**

If you invoke the metrics web context (for example, **/metrics**) on a broker, but a metrics plugin has not yet been configured, the broker displays a null pointer exception (NPE).

For more information about configuring the *Prometheus* metrics plugin for AMQ Broker, see [Enabling the Prometheus plugin for AMQ Broker](#) (on-premise deployments) or [Enabling the Prometheus plugin for a running broker deployment](#) (OpenShift deployments).

- **ENTMQBR-3695 - Prometheus metrics plugin is loaded but causes NPE**

If you upgrade an on-premise broker instance to AMQ Broker 7.7, and a metrics plugin such as the Prometheus plugin was already enabled on that broker, the broker loads the plugin, but the plugin does not collect broker runtime metrics. In addition, you see a null pointer exception (NPE) when you invoke the metrics web context (for example, **/metrics**) on the upgraded broker.

To work around this issue, modify the configuration of the upgraded broker to resemble that shown in [Enabling the Prometheus plugin for AMQ Broker](#) in *Managing AMQ Broker*.

- **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.

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

- **ENTMQBR-4127 - 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.

- **ENTMQBR-4143 - AMQ Broker Operator: Type mismatch for `pageSizeBytes` property between CRD and Operator**

If you add an address settings configuration to the Custom Resource (CR) instance for a broker deployment (that is, by adding an **addressSettings.addressSetting** section), you cannot include the **pageSizeBytes** property in the address settings. If you include this property and specify a value, the Operator either fails to process the CR, or it processes the CR but cannot start any brokers. There is no way to work around this issue.

- **ENTMQBR-4144 - AMQ Broker Operator: The `redeliveryCollisionAvoidanceFactor` property cannot be specified in an Operator configuration**

If you add an address settings configuration to the Custom Resource (CR) instance for a broker deployment (that is, by adding an **addressSettings.addressSetting** section), you cannot include the **redeliveryCollisionAvoidanceFactor** property. If you include this property and specify a value, the Operator fails to process the CR. There is no way to work around this issue.

- **ENTMQBR-4146** - Broker fails to start when **default-group-rebalance-pause-dispatch** property is specified in address settings

If you configure 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 cannot start.

This issue also occurs for broker deployments on OpenShift Container Platform. Specifically, if you add 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 cannot start.

There is no way to work around this issue. The default values of both **default-group-rebalance-pause-dispatch** and **defaultGroupRebalancePauseDispatch** are **false**.

- **ENTMQBR-4163** - AMQ Broker Operator: Incorrect specification for memory limit and memory request in sample Custom Resource

In the **broker_activemqartemis_cr.yaml** sample CR instance that is included with version 0.17 of the AMQ Broker Operator, the values for the **limits.memory** and **requests.memory** properties are incorrectly specified. For these properties, the values included in the sample CR are **1024m** and **512m**. A lowercase **m** suffix is not valid for these properties. Valid suffixes are **E**, **P**, **T**, **G**, **M**, and **K** or the binary equivalents, **Ei**, **Pi**, **Ti**, **Gi**, **Mi**, **Ki**. The sample CR is included in the **deploy/crs** directory of the installation archive that you download and extract when installing the Operator using the OpenShift command-line interface.

CHAPTER 6. IMPORTANT LINKS

- [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 2020-12-17 16:08:23 UTC