



**Red Hat AMQ 2020.Q4**

**Red Hat AMQ Broker 7.8 发行注记**

AMQ Broker 发行注记



# Red Hat AMQ 2020.Q4 Red Hat AMQ Broker 7.8 发行注记

---

## AMQ Broker 发行注记

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律通告

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Release\_Notes\_for\_Red\_Hat\_AMQ\_Broker\_7.8.ent file |.

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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.

## 摘要

这些发行注记包含有关 AMQ Broker 7.8 发行版本中包含的新功能、增强功能、修复和问题的最新信息。

---

## 目录

第 1 章 AMQ BROKER 7.8 的长期支持 .....	3
支持 Red Hat Enterprise Linux 和 OpenShift Container Platform .....	3
第 2 章 功能增强 .....	4
第 3 章 已弃用的功能 .....	6
第 4 章 技术预览 .....	7
第 5 章 修复的问题 .....	8
第 6 章 修复了常见漏洞和暴露的问题 .....	12
第 7 章 已知问题 .....	13
第 8 章 重要链接 .....	18



## 第 1 章 AMQ BROKER 7.8 的长期支持

AMQ Broker 7.8 被指定为 Long Term Support(LTS)发行版本。程序错误修复及安全公告会在一系列微版本（7.8.1、7.8.1.2 等）内为 AMQ Broker 7.8 提供，至少 12 个月。

这意味着，您可以在 *无需* 升级到新的次版本的情况下，为 AMQ Broker 获取最新的错误修复和安全公告。

注意以下有关 LTS 发行流的重要点：

- LTS 发行流仅提供程序错误修正。这个流中不会添加新改进。
- 要保留在受支持的配置中，您必须升级到 LTS 发行流中的最新微版本。
- LTS 版本自 AMQ Broker 7.8.0 GA 发行版本中至少支持 12 个月。

### 支持 Red Hat Enterprise Linux 和 OpenShift Container Platform

AMQ Broker 7.8 LTS 版本支持：

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

请注意与支持 Red Hat Enterprise Linux 和 OpenShift Container Platform 的以下重要点：

- AMQ Broker 7.8 是支持 Red Hat Enterprise Linux 6 和 OpenShift Container Platform 3.11 的**最后一个**版本。
- 红帽 **不保证** 在以后的 OpenShift Container Platform 版本（大于 4.6）上支持 AMQ Broker 7.8。

有关 AMQ Broker 7.8 LTS 微版本中的问题的信息，请参阅 [AMQ 7 Broker - 7.8.x 解决问题](#)。

## 第 2 章 功能增强

这部分论述了 AMQ Broker 7.8 中已突出显示的功能增强和新功能集合。如需版本改进的完整列表，请参阅 [AMQ Broker 7.8.0 增强](#)。

### AMQ Management Console 的新版本

AMQ Broker 7.8 包括新版本的 AMQ 管理控制台。有关使用控制台的更多信息，请参阅 [管理 AMQ Broker 中的 AMQ 管理控制台](#)。

### 新数据库认证

AMQ Broker 7.8 添加了对 PostgreSQL 11.5 和 MySQL 8 的支持。有关 AMQ Broker 不同版本的数据库的完整信息，请参阅 [Red Hat AMQ 7 支持的配置](#)。

### 迭代地址和队列

在 AMQ Broker 7.8 中，您可以配置地址和队列的联合。联合启用代理之间的信息传输，无需代理位于共同的集群中。例如，联合适合可靠地将消息从一个集群发送到另一个集群。此传输可能是跨广域网 (WAN)、云基础架构的区域或互联网。如需更多信息，请参阅 [配置 AMQ Broker 中的迭代地址和队列](#)。

### 禁用队列

在 AMQ Broker 7.8 中，您可以禁用代理配置中定义的队列。例如，您可能要定义一个队列，以便客户端可以订阅它，但您尚未准备好将队列用于消息路由。或者，您可能要停止消息流到队列，但仍然使客户端与队列绑定。在这些情况下，您可以禁用队列。如需更多信息，请参阅 [配置 AMQ Broker 时禁用队列](#)。

### 垂直扩展队列的性能改进

AMQ Broker 7.8 添加了可扩展性改进，可在部署自动扩展到大量队列时提高代理性能。这个改进适用于所有支持的协议，但对 MQTT 来说尤其有用，这通常用于大型部署。此性能增强对于带有大量队列的代理部署（例如 50,000 或更多）。

### 使用地址设置更新正在运行的基于 Operator 的代理部署

在 AMQ Broker 7.8 中，现在可以在已在运行的基于 Operator 的代理部署中添加地址设置。在 AMQ Broker 7.7 中添加了代理部署的地址设置，支持在自定义资源 (CR) 实例中包含地址设置。但是在 7.7 中，您需要在第一次创建代理部署时配置地址设置。有关配置地址、队列和地址设置的更多信息，请参阅在 OpenShift 上部署 AMQ Broker [中为基于 Operator 的代理部署配置地址](#) 和队列。

### 除了基本审计日志记录器外

现在，在暂停和恢复地址时，基本审计日志记录。要了解如何配置日志记录，请参阅 [配置 AMQ Broker 中的日志记录](#)。{`BrokeConfigureBookUrl`}/logging

### 用于代理地址内存用量百分比的新指标

在 7.8 中，AMQ Broker 的 Prometheus 指标插件会导出一个名为 **artemis\_address\_memory\_usage\_percentage** 的新指标。此指标是代理上的所有地址使用的总地址内存，作为 **global-max-size** 参数的值百分比。要了解如何配置 Prometheus 指标插件，请参阅 [管理 AMQ Broker 中的监控代理 运行时指标](#)。

### 改进了 moverts 配置

在 7.8 中，如果您使用 AMQ 管理控制台或管理 API 在 live 代理上配置运行时传播，则 movert 会自动传播到备份代理。这不是以前的版本。

### 指定自定义初始容器镜像

用于 7.8 的 Operator 的最新版本使用称为 Init Container 的专用容器来生成代理配置。默认情况下，Operator 使用内置的初始容器镜像。但是，您还可以指定自定义 Init 容器镜像，用于修改或添加到由内置 Init 容器创建的配置中。如需更多信息，请参阅在 OpenShift 上部署 AMQ Broker 中 [指定自定义 Init 容器镜像](#)。

### Operator 支持多个容器平台

在 7.8 中，AMQ Broker Operator 支持以下容器平台：



- [OpenShift Container Platform](#)
- [IBM Z 上的 OpenShift Container Platform](#)
- [IBM Power 系统上的 OpenShift Container Platform](#)

IBM Power Systems 上的 OpenShift Container Platform 支持在 7.8 中是新的。用于 AMQ Broker 7.5 的 Operator 版本，支持 IBM Z 上的 OpenShift Container Platform。

在 7.5 中，您需要为每个支持的平台安装和部署 Operator 的单独版本。在 7.8 中，您只需要安装一个支持所有三个容器平台的版本。基于您要使用的容器平台，Operator 会自动选择要在部署中使用的代理容器镜像。

要了解如何安装 Operator 的最新版本，请参阅 OpenShift 中部署 AMQ Broker 中的以下部分：

- [使用 CLI 安装 Operator](#)
- [使用 Operator Lifecycle Manager 安装 Operator](#)

### Operator 自动选择代理容器镜像

在最新版本的 7.8 中，当您使用自定义资源(CR)实例创建代理部署时，您不再需要在 CR 中明确指定代理容器镜像名称。相反，当部署 CR 时，Operator 会自动决定要使用的相应代理容器镜像。这也适用于生成代理配置的 Init 容器。如需了解更多相关信息，请参阅 [Operator 如何在 OpenShift 上部署 AMQ Broker 中选择容器镜像](#)。

### RHEL 8 Operator

x86\_64 平台、IBM Z 和 IBM Power Systems 上提供了名为 **Red Hat Integration - AMQ Broker for RHEL 8(Multiarch)** 的 Operator。它支持以下频道：

- **7.x** - 当有可用时，此频道将更新至 **7.9**。
- **7.8.x** - 这是 Long Term Support(LTS) 频道。

要确定要选择哪个 Operator，请查看 [Red Hat Enterprise Linux Container Compatibility Matrix](#)。

### Operator 频道

在 7.8 的 Operator 的最新版本中，**Red Hat Integration - AMQ Broker** Operator 提供了以下新的更新频道：

- **7.x** - 等同于 **当前** 现已弃用的频道。
- **7.8.x** - 这是 Long Term Support(LTS) 频道。

### Operator 版本

在 7.8 的 Operator 的最新版本中，Operator 现在使用与 AMQ Broker 相同的版本。例如，x86\_64 平台上的以前的 Operator 版本是 OperatorHub 中的 **0.19** 版本，该版本更新至 **7.8.2-opr-1**。

### 文档更新

AMQ Broker 文档已更新，提供有关新 Operator 和频道以及 IBM Z 和 IBM Power Systems 支持的说明。

### 其他资源

- 有关 AMQ Broker 7.8 发行版本中的改进的完整列表，请参阅 [AMQ Broker 7.8](#)。

## 第 3 章 已弃用的功能

本节介绍了从 AMQ Broker 中弃用的功能。

### Hawtio 分配控制台插件

从 7.3 开始，AMQMQ Broker 不再随 Hawtio 分配控制台插件一同提供，即 `assign -hawtio-console.war`。在以前的版本中，分配控制台用于管理 AMQ Interconnect。但是，AMQ 互联现在使用自己的独立 Web 控制台。

### 网络 pinger

从 7.5 开始，网络 ping 是一个已弃用的功能。网络 ping 无法防止代理集群不受网络隔离问题导致无法恢复的消息丢失。这个功能将在以后的发行版本中被删除。红帽继续支持使用网络 ping 的现有 AMQ Broker 部署。但是，红帽不推荐在新部署中使用网络 ping 网络。有关配置代理集群以实现高可用性并避免网络隔离问题的指导，请参阅在 [配置 AMQ Broker 中 实施高可用性](#)。

### OpenShift Container Platform 上 AMQ Broker 的应用程序模板

从 7.8 开始，在 OpenShift Container Platform 中使用应用程序模板在 OpenShift Container Platform 中部署 AMQ Broker 是一个已弃用的功能。这个功能将在以后的发行版本中被删除。红帽继续支持基于应用程序模板的现有部署。但是，红帽不推荐在新部署中使用应用程序模板。对于新部署，红帽建议使用 AMQ Broker Operator。如需有关安装和部署 Operator 的信息，请参阅使用 [AMQ Broker Operator 在 OpenShift Container Platform 上部署 AMQ Broker](#)。

## 第 4 章 技术预览

本节论述了 AMQ Broker 7.8 中的技术预览功能。



### 重要

技术预览功能不被红帽产品服务等级协议(SLA)支持，且功能可能并不完善。红帽不推荐在生产环境中使用它们。如需更多信息，请参阅[红帽技术预览功能支持范围](#)。

### AMQP 服务器连接

在 AMQ Broker 7.8 中，代理可以使用 AMQP 协议发起到其他端点的连接。例如，这意味着代理可以连接到其他 AMQP 服务器（不仅仅是 AMQ Broker 的实例），并在这些连接中创建元素。如需更多信息，请参阅 [Apache ActiveMQ Artemis 文档中的代理连接](#)。

### 在 Fuse 控制台中查看代理

您可以将基于 Operator 的代理部署配置为使用 Fuse Console for OpenShift，而不是 AMQ 管理控制台。适当配置了代理部署后，Fuse Console 会发现代理并将其显示在专用 **Artemis** 选项卡上。如需更多信息，请参阅 [在 OpenShift 上部署 AMQ Broker 中的 Fuse Console 中的查看代理](#)。



### 注意

在 Fuse Console 中查看代理是 Fuse 7.8 的技术预览功能。

## 第5章 修复的问题

这部分论述了在 AMQ Broker 7.8 中修复的已突出显示的问题集合。有关发行版本里修复的问题的完整列表，请参阅 [AMQ Broker 7.8.0 修复问题](#) 和 [AMQ 7 Broker - 7.8.x 解决了的问题](#)。

- ENTMQBR-1815 - Hawtio 查看自动刷新的变化**  
在以前的版本中，当启用自动刷新时，C Hawtio 控制台每 5 秒更新屏幕。此行为还会将视图切换到 **Attributes** 选项卡，从而导致您丢失查看的任何其他选项卡中的关注。这个问题现已解决。
- ENTMQBR-2890 - 经常，创建大小为  $n > 1$  的 CR 实例会导致第  $n$  个代理 pod 启动并立即重启一次**  
在以前的版本中，如果您使用自定义资源(CR)实例使用 AMQ Broker Operator 部署代理集群，则部署中最终的代理 Pod (根据 CR 的大小 属性决定)，然后在可用前立即重启一次代理集群。这个问题现已解决。部署中的任何代理 Pod 在被使用前被重启。
- ENTMQBR-3059 - AMQ Broker Operator:在重启 / update 后，Operator 不会获取 CR 名称**  
在以前的版本中，如果您创建了一个集群的代理部署，它被配置为使用持久性和消息迁移，AMQ Broker Operator 会在实例化扩展控制器进行消息迁移时生成无效的名称。具体来说，如果 Operator 已在代理部署扩展前更新或其镜像，会出现这种情况。这种情形的结果是需要删除并重新创建代理部署。这个问题现已解决。
- ENTMQBR-3514 - AMQ Broker Operator:如果在代理实例化前提交地址 CR，则不会创建地址**  
在以前的版本中，对于基于 Operator 的代理部署，如果您在代理被实例化前为地址创建了自定义资源(CR)实例，Operator 无法创建地址。这个问题现已解决。
- ENTMQBR-3578 - AMQ Broker Operator:在启动时不支持将现有的 CR 实例用作进出的基准，**  
在以前的版本中，当 AMQ Broker Operator 启动时，它不会检查项目中现有的自定义资源(CR)实例。这意味着，如果需要重启 Operator (例如，要应用新的 Operator 镜像版本)，Operator 和代理部署将不再同步。在这种情况下，需要删除并重新创建代理部署。这个问题现已解决。
- ENTMQBR-3587 - 关闭关键 IO 错误时的 Avoid 通知**  
在以前的版本中，当因为关键 IO 错误而关闭自身时，代理会生成多个触发磁盘 IO 的通知。这些通知可能会延迟或者事件阻止关闭代理。这个问题现已解决。
- ENTMQBR-3617 - 对于持久地址的 hawtio 控制台的用户信息是 null**  
在以前的版本中，当使用者创建共享的、持久订阅时，AMQ 管理控制台可能会将关联的用户显示为由代理自动创建的订阅队列的 **null**。这个问题现已解决。
- ENTMQBR-3692 - 关于 hawtio 控制台的持久地址的用户信息为空**  
在以前的版本中，如果消息驱动的 Bean(MDB)使用 ActiveMQ Java 连接器架构 (例如，在 Red Hat JBoss Enterprise Application Platform 中) 创建持久的主题订阅，MDB 将无法部署。这个问题现已解决。
- ENTMQBR-3705 - LVQ + 非破坏性不会向现有消费者传送消息**  
在以前的版本中，当使用者创建共享的、持久地址时，AMQ 管理控制台可能会将关联的用户显示为 **null**。这个问题现已解决。
- ENTMQBR-3710 - 增加审核消息**  
在以前的版本中，如果您暂停并恢复某个队列，审计日志记录器会错误地包含将恢复事件描述为另一个暂停事件的文本。例如：

```
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

这个问题现已解决。

- **ENTMQBR-3719 - LegacyLDAPSecuritySettingPlugin 允许新用户访问任何新创建的目的地**  
在以前的版本中，当为LDAP 添加新权限时，**LegacyLDAPSecuritySettingPlugin** 插件使用新权限修改默认安全匹配。这可能会破坏现有用户的授权。这个问题现已解决。
- **ENTMQBR-3726 - JVM 属性 hawtio.role 不通过空格和连字符解析角色**  
在以前的版本中，如果 **artemis.profile** 文件定义了包含空格或连字符的 **hawtio.role** 属性，则该属性无法正常工作。这个问题现已解决。
- **ENTMQBR-3744 - 启用带有默认 / non-zero consumer-window-size 的组重新平衡可能会导致后续消息消耗**  
在以前的版本中，如果某些连接使用者使用 **consumerWindowSize** 的值大于零（将消息预填充到那些客户端的缓冲区中），并将代理配置为使用组重新平衡（通过设置 **group-rebalance** 或 **default-group-rebalance** to **true**），这可能会导致对订购的消息消耗不足。这个问题现已解决。
- **ENTMQBR-3752 - 备份代理无法重新建立与 master 的连接**  
如果网络中断，在 **live-backup** 组中两个代理都可能同时出现（称为网络隔离或分割brain 的情况）。在以前的版本中，如果发生这种情况，任何连接的AMQ Core Protocol JMS 客户端都会收到不正确的代理拓扑信息。因此，当解决网络和脑裂问题时，客户端无法重新连接到正确的代理。要临时解决这个问题，您需要重启客户端。这个问题现已解决。
- **ENTMQBR-3782 - page-max-concurrent-io 无法禁用**  
在以前的版本中，将 **page-max-concurrent-io** 的值设置为 **-1** 以删除分页上允许的并发读取数量的上限。如果您从默认值更改了默认值，则代理继续使用默认值，或者以前配置的值。这个问题现已解决。
- **ENTMQBR-3797 - 激活失败可能会导致 Zombie 代理**  
在以前的版本中，如果将 **live-backup** 代理组配置为使用共享存储高可用性，则实时代理可能无法在重启后正确激活，但仍会保留日志锁定。例如，如果 **live-backup** 组使用网络文件系统(NFS)，且 NFS 意外停止或删除，则可能会出现此问题，从而导致 **live** 代理重启。这种情况会导致无法服务客户端但无法激活备份代理。这个问题现已解决。
- **ENTMQBR-3798 - JDBC XML 配置无法使用自定义密码代码**  
在以前的版本中，如果您为 **jdbc-password** 参数和代理配置中的 **password-codec** 参数指定了已屏蔽的密码，则代理总是使用默认的 **org.apache.activemq.artemis.utils.DefaultSensitiveStringCodec** codec 来解码密码。这个问题现已解决。
- **ENTMQBR-3812 - 在同时销毁队列和存储库时，可能死锁**  
在以前的版本中，如果队列被销毁（例如，当一个不可中断的使用者关闭了连接时），且依赖用户会同时发生，这可能会导致死锁状况。这个问题现已解决。
- **ENTMQBR-3813 - 队列更新上的 Null 指针异常**  
在以前的版本中，如果您尝试更新最初创建没有过滤器的队列，代理可能会显示 null pointer exception(NPE)。这个问题现已解决。
- **ENTMQBR-3841 - Concurrent Jolokia 操作可能会错误地更新 artemis-roles.properties 或 artemis-users.properties**  
在以前的版本中，如果对代理进行了多个并发 Jolokia 操作，则处理程序可能会错误地更新或删除 **artemis-roles.properties** 或 **artemis-users.properties** 配置文件中的数据。这个问题现已解决。

- ENTMQBR-3880 - 在分页期间为通配符地址替换了目标标头**  
 在以前的版本中，在存储消息前，代理会在消息上设置 `address` 字段来反映页面存储名称。如果消息是首次使用通配符地址表达式订阅的消费者页面，这将导致其他消费者无法处理的目的地名标头不正确。这个问题已解决。在将消息写入页面存储时，代理不再更改消息内容。这将使原始目标地址保持不变。
- ENTMQBR-4034 - 重启后 LVQ 中断**  
 在以前的版本中，在代理重启后，最后一个值队列中的现有消息不会被替换为发送到具有相同 `last-value` 键队列的新消息。这个问题现已解决。
- ENTMQBR-4143 - AMQ Broker Operator:CRD 和 Operator 间的 `pageSizeBytes` 属性类型不匹配**  
 在以前的版本中，如果您为代理部署（即添加 `addressSettings.addressSetting` 部分）添加地址设置配置到自定义资源(CR)实例，则您无法包含 `pageSizeBytes` 属性。如果您包含此属性并指定了一个值，Operator 将无法处理 CR，或者处理 CR，但可以启动任何代理。这个问题现已解决。
- ENTMQBR-4144 - AMQ Broker Operator:无法指定 `address` 设置 `redeliveryCollisionAvoidanceFactor`**  
 在以前的版本中，如果您为代理部署（即添加 `addressSettings.addressSetting` 部分）添加地址设置配置到自定义资源(CR)实例，则无法使用 `redeliveryCollisionAvoidanceFactor` 属性。如果包含此属性并指定值，Operator 将无法处理 CR。这个问题现已解决。
- ENTMQBR-4145 - AMQ Broker Operator:无法指定 `address` 设置 `autoCreateJmsTopics`**  
 在以前的版本中，如果您为代理部署（即添加 `addressSettings.addressSetting` 部分）添加地址设置配置到自定义资源(CR)实例，则无法使用 `autoCreateJmsTopics` 属性。如果您包含此属性并指定了一个值，Operator 会处理 CR，但无法将属性包括在生成的代理配置中。这个问题现已解决。
- 在地址设置中指定 `default-group-rebalance-pause-dispatch` 属性时，ENTMQBR-4146 - Broker 无法启动**  
 在以前的版本中，如果您将 `broker.xml` 配置文件的 `address-setting` 元素配置为将 `default-group-rebalance-pause-dispatch` 属性的值设置为 `true`，则代理无法启动。  
 这个问题还会在 OpenShift Container Platform 上部署代理。特别是，如果您在自定义资源(CR)实例中为代理部署添加了地址设置配置（即，通过添加 `addressSettings.addressSetting` 部分），并将 `defaultGroupRebalancePauseDispatch` 属性的值设置为 `true`，部署中的代理无法启动。  
 这个问题现已解决。
- ENTMQBR-4159 - AMQ Broker Operator:未为 STOMP 协议创建路由**  
 在之前的发行版本中，如果您定义了接受者使用 STOMP 协议，但没有为接受者指定一个端口，Operator 无法为接受者创建 Service 和 Route。这个问题现已解决。
- ENTMQBR-4195 - 在 AMQ 代理重启后删除调度的消息重新应用**  
 在之前的发行版本中，如果您使用 management API 删除调度的信息，则消息会从内存中删除，而不是从存储中删除。这会导致在重启代理时该消息重新应用。这个问题现已解决。
- ENTMQBR-4263 - 通过 DLQ 将消息变为较大的消息，关闭代理**  
 在以前的版本中，如果给定协议的消息接近了为该协议配置的大量消息大小，代理会尝试将消息传送到死信队列，则代理可能会意外关闭。这个问题现已解决。

## 其他资源

- 有关 AMQ Broker 7.8 发行版本中修复的问题的完整列表，请参阅 [AMQ Broker 7.8.0 修复的问题](#)。

## 第 6 章 修复了常见漏洞和暴露的问题

本节详细介绍了 AMQ Broker 7.8 发行版本中的通用漏洞和风险(CVE)修复。

- [ENTMQBR-3755](#) - CVE-2020-13932 - mqtt-client: activemq: remote XSS in web 控制台图插件 [amq-7]
- [ENTMQBR-3382](#) - CVE-2015-5183 Hawtio : 在 Cookie [amq-7] 上未设置 HTTPOnly 和 Secure 属性
- [ENTMQBR-4037](#) - CVE-2019-12749 - DBusServer DBUS\_COOKIE\_SHA1 身份验证绕过
- [ENTMQBR-4068](#) - CVE-2019-9827 - hawtio:通过 URI [amq-7.7.0] 的初始 /proxy/ 子字符串进行服务器端请求
- [ENTMQBR-4158](#) - CVE-2020-27216 - jetty:本地临时目录劫持漏洞 [amq-7]



## 第7章 已知问题

本节论述了 AMQ Broker 7.8 中的已知问题。

- **ENTMQBR-17 - AMQ222117:无法启动集群连接**  
代理集群可能无法在支持 IPv6 的环境中正确初始化。故障是由于日志消息 **无法** 分配请求的地址表示的 **SocketException** 表示。要临时解决这个问题，请将 **java.net.preferIPv4Stack** 系统属性设置为 **true**。
- **ENTMQBR-463 - 集群设置中的属性具有顺序限制。最好有一条更好的错误消息，或者只是忽略顺序**  
目前，集群连接配置中的元素序列需要按特定顺序排列。解决办法是遵循配置方案中的顺序。
- **ENTMQBR-520 - 不允许从名称与另一个地址绑定的地址接收。**  
名称与地址相同的队列只能分配给地址。创建与现有地址相同的队列，但绑定到具有不同名称的地址是无效的配置。这样做可能会导致将不正确的消息路由到队列中。
- **ENTMQBR-522 - 在窗口上运行的代理编写问题，并在关闭时删除临时文件**  
在 Windows 上，代理不会在关闭时成功清理临时文件。此问题会导致关闭过程很慢。另外，代理会不断积累了不被代理删除的临时文件。
- **ENTMQBR-569 - 将 OpenWire 中的 ID 转换为 AMQP 会导致发送 ID 为二进制**  
当从 A-MQ 6 OpenWire 客户端与 AMQP 客户端通信跨协议时，将在应用程序消息属性中编码其他信息。这是代理内部使用的 benign 信息，可以忽略。
- **ENTMQBR-599 - 由 Artemis cli 定义信任存储和密钥存储**  
使用 **--ssl-key**、**--ssl-key-password**、**--ssl-trust** 和 **--ssl-trust-password** 参数创建代理实例将无法正常工作。要临时解决这个问题，在创建代理后手动设置 **bootstrap.xml** 中的对应属性。
- **ENTMQBR-636 - Journal break，导致 perf 负载下 JavaNullPointerException。**  
为防止代理管理重重负载时发生 IO 相关问题，请验证 JVM 是否已分配足够内存和堆空间。请参阅 ActiveMQ Artemis 文档中的 **性能调优** 章节中标题为“虚拟机”的部分。
- **ENTMQBR-648 - JMS Openwire 客户端无法将消息发送到带有定义的purgeOnNoConsumer 或 queue 过滤器的队列**  
使用 A-MQ 6 JMS 客户端将消息发送到地址，如果队列没有使用者，则会将 **purgeOnNoConsumer** 设置为 **true** 的队列设为 true。在使用 A-MQ 6 JMS 客户端时，建议您不要设置 **purgeOnNoConsumer** 选项。
- **ENTMQBR-652 - 已知 amq-jon-plugin 错误列表**  
此版本的 **amq-jon-plugin** 已知的与代理和队列的 MBeans 相关的问题。

broker MBean 的问题：

- 关闭连接会引发 **java.net.SocketTimeoutException** 异常
- **listSessions()** throws **java.lang.ClassCastException**
- 添加地址设置会引发 **java.lang.IllegalArgumentException**
- **getConnectorServices ()** 操作无法找到
- 无法找到 **listConsumersAsJSON ()** 操作
- 无法找到 **getDivertNames ()** 操作

- 列出网络拓扑会抛出 **IllegalArgumentException**
- 删除地址设置有错误的参数名称

队列 MBean 的问题：

- **expireMessage ()** 丢弃参数类型不匹配异常
  - **listDeliveringMessages()** throws **IllegalArgumentException**
  - **listMessages ()** throws **java.lang.Exception**
  - **moveMessages ()** throws **IllegalArgumentException** 带有错误消息参数类型不匹配
  - **removeMessage ()** throws **IllegalArgumentException** 带有错误消息参数类型不匹配
  - **removeMessages ()** 抛出异常，错误 Can't find removeMessage with 2 参数
  - **retryMessage ()** throws 参数类型不匹配 **IllegalArgumentException**
- **ENTMQBR-655 - [AMQP] 在启用 fill-validated-user 时无法发送消息**  
对于使用 AMQP 协议生成的消息，配置选项不支持 填充-validated-user。
  - **ENTMQBR-738 - 通过提供离线存储库离线构建 AMQ 7 示例**  
您不能在离线环境中构建 AMQ Broker 中包含的示例。造成这个问题的原因是，提供的离线 Maven 存储库中缺少依赖项。
  - **ENTMQBR-897 - 在目标名称中使用特殊字符的 Openwire 客户端/协议问题**  
目前 AMQ OpenWire JMS 客户端无法访问在其名称中包含以下字符的队列和地址：逗号(',')、hash('#')、大于('>')和空格。
  - **ENTMQBR-944 - [A-MQ7, Hwtio, RBAC] 用户在 RBAC 拒绝操作访问时不会获得反馈**  
控制台可以指示未授权用户当没有成功时尝试的操作被成功。
  - **ENTMQBR-1498 - HA (复制、共享存储) 的管理控制台图不反映真正的拓扑**  
如果您使用一些额外的被动从设备配置代理集群，Web 控制台中的集群图不会显示这些被动的从设备。
  - **ENTMQBR-1848 - "javax.jms.JMSEException:队列的路由类型不正确，预期为：当 qpid-jms 客户端使用来自多播队列的消息为 javax.jms.Queue 对象时，会发生 ANYCAST"。**  
目前，通过将 Qpid JMS 客户端使用 FQQN（完全限定队列名称）发送到配置了多个队列的地址，这会在客户端上生成错误消息，并且该消息无法发送。要临时解决这个问题，修改代理配置以解决错误并取消阻塞客户端。
  - **ENTMQBR-1875 - [AMQ 7, ha, Replica store] 备份代理似乎在 - ActiveMQIllegalStateException 错误Type=ILLEGAL\_STATE message=AMQ119026 后出现。备份服务器尚未与 live 同步**  
当备份代理试图与 master 代理同步时，删除 master 代理的分页磁盘会导致 master 代理失败。另外，备份代理将无法变为现实，因为它继续尝试与 master 同步。
  - **ENTMQBR-2068 - 在 HA 故障切换过程中收到的一些消息，故障恢复场景**  
目前，如果代理在 OpenWire 客户端发送消息时失败，则在故障切换发生时发送到代理的消息可能会丢失。要临时解决这个问题，请确保代理在确认信息前保留它们。
  - **ENTMQBR-2452 - 从 Windows 上的 AMQ 7.2.4 升级代理 AMQ 7.3.0**

如果要代理实例从 Windows 上的 7.2.4 升级到 7.3.0，则日志记录将无法正常工作，除非您在升级过程中指定正确的日志管理器版本。如需更多信息，请参阅在 [Windows 上从 7.2.x 升级到 7.3.0](#)。

- **ENTMQBR-2470 - [AMQ7, openwire, redelivery] 重新传送计数器以增加消息，如果消费者在不使用任何消息的情况下关闭**

如果代理发送消息到 Openwire consumer，但消费者在消耗消息前关闭，代理会错误地递增待处理消息的重新发送计数。如果出现此行为的数量超过 **max-delivery-attempts** 配置参数的值，代理会将消息发送到死信队列(DLQ)，或者根据您的配置丢弃消息。这个问题不会影响其他协议，如核心协议。

- **ENTMQBR-2593 - 代理不会在跨协议消耗中设置消息 ID 标头**

只有在被另一 Qpid JMS 客户端生成消息时，Qpid JMS 客户端才会成功检索消息 ID。如果消息由核心 JMS 或 OpenWire 客户端生成，则 Qpid JMS 客户端无法读取消息 ID。

- **ENTMQBR-2678 - 在隔离的 master 处于活动状态后，它无法连接到集群**

在三个或更多使用复制高可用性(HA)策略的实时备份组中，实时代理会在复制连接失败时关闭。但是，当恢复复制连接时，原始 live 代理被重启时，代理有时无法重新加入代理集群。要启用原始实时代理重新加入集群，首先停止新的实时（原始备份）代理，重启原始实时代理，然后重启原始备份代理。

- **ENTMQBR-2928 - Broker Operator 无法从 CR 更改中恢复，从而导致错误的状态**

如果在应用自定义资源(CR)更新时，如果 AMQ Broker Operator 遇到错误，Operator 不会恢复。特别是，Operator 会停止响应，以便进一步更新 CR。

例如，如果主代理 CR 中的 **image** 属性的值出现错误拼写，则代理 Pod 将无法部署，且相关的错误消息为 **ImagePullBackOff**。如果修复拼写错误并应用 CR 更改，Operator 不会部署指定的代理 Pod 的数量。另外，Operator 不会响应任何进一步的 CR 更改。

要临时解决这个问题，您必须在重新部署前删除最初部署的 CR。要删除现有的 CR，请使用 **oc delete -f <CR name>** 等命令。

- **ENTMQBR-2942 - Pod #0 会尝试联系不存在的 Pod**

如果您将自定义资源(CR)实例的 **size** 属性更改为缩减代理部署，集群中的第一个代理 Pod 可以进行重复尝试连接到启动排空器 Pod，以便在关闭前从关闭的代理迁移消息。要临时解决这个问题，请按照以下步骤执行：

- 1) 将部署扩展到单个代理 Pod。
- 2) 等待所有排空程序 Pod 启动、完成消息迁移，然后关闭。
- 3) 如果单个剩余代理 Pod 的日志条目为“未知主机异常”，将部署扩展为零代理 Pod，然后返回到一个代理 Pod。
- 4) 当您验证单个剩余代理 Pod 没有记录基于异常的日志条目时，请将您的部署扩展至其原始大小。

- **ENTMQBR-3131 - 当 Master 为 Killed 时，Topology Fails 无法正确更新备份代理**

当一个集群中有超过四个 live-backup 对的 live 代理失败时，实时代理（包括新选择的实时代理）会正确报告更新的拓扑。但是，剩余的备份代理可能会以以下方法显示错误拓扑：

- 如果备份代理故障转移了失败的 live 代理，则剩余的备份代理会在拓扑中显示这个备份代理两次。
- 如果备份代理还没有失败，则剩余的备份代理仍然会在拓扑中显示失败的实时代理。

要临时解决这个问题，请确保 `cluster-connection > static-connectors` 配置每个备份代理的第一个 `connector-ref` 元素指定了预期的 live 代理。

- **ENTMQBR-3604 - 为LDAP 登录模块使用关闭启用池(Hang)**

如果您为LDAP 供应商启用连接池（即，在 `login.config` 配置文件的 `LDAPLoginModule` 部分中将 `connectionPool` 设置为 `true`），这将导致到LDAP 供应商的连接保持开放，即使您停止代理客户端也是如此。因此，如果您尝试以正常方式关闭代理，代理不会关闭。反之，您需要使用一个Linux 命令（如 `SIGKILL`）终止代理进程。即使您在JVM 参数中指定池超时（例如，`-Dcom.sun.jndi.ldap.connect.pool.timeout=30000`），并且当尝试关闭代理时没有活跃的客户

要临时解决这个问题，请在 `login.config` 配置文件的 `LDAPLoginModule` 部分中为 `connectionTimeout` 属性设置一个值。当为连接池请求连接池时，`connectionTimeout` 属性指定当最大池大小已达到并且池中所有连接都处于连接的最长时间。如需更多信息，请参阅配置 AMQ Broker 中的 [使用LDAP 进行身份验证](#)。

- **ENTMQBR-3653 - 如果尚未配置指标插件并且调用指标 Web 上下文，则丢弃NPE**

如果代理上的 `/metrics` web 上下文被调用，但 `metrics` 插件尚未配置，则代理会显示 null pointer 异常。如需有关为 AMQ Broker 配置 Prometheus 指标插件的更多信息，请参阅 [为 AMQ Broker（内部代理部署）启用 Prometheus 插件](#)，或为正在运行的代理部署（OpenShift 代理部署）启用 Prometheus 插件。

- **ENTMQBR-3724 - OperatorHub 显示在 AMQ Broker Operator 的变体**

如果使用 OperatorHub 在 OpenShift Container Platform 4.5 或更早版本部署 AMQ Broker Operator，则 OperatorHub 会显示一个与主机平台相关的 Operator 变体。这样便可选择不正确的 Operator 变体。特别是，无论您的主机平台是什么，Operator 会显示 **Red Hat Integration - AMQ Broker Operator**（用于 OpenShift Container Platform 的 Operator）和 **AMQ Broker Operator**（IBM Z 上的 OpenShift Container Platform 的 Operator）。

要临时解决这个问题，请选择适合您的平台的 Operator 变体，如上文所述。另外，也可使用 OpenShift 命令行界面(CLI)安装 Operator。

在 OpenShift Container Platform 4.6 中，这个问题已解决。OperatorHub 仅显示与主机平台相对应的 Operator 变体。

- **ENTMQBR-3846 - MQTT 客户端在代理重启时不会重新连接**

当您重启代理或代理失败时，活跃代理不会恢复之前连接的 MQTT 客户端的连接。要临时解决这个问题，若要重新连接 MQTT 客户端，您需要在客户端上手动调用 `subscribe()` 方法。

- **ENTMQBR-4023 - AMQ Broker Operator:Pod 状态 pod 名称没有反映现实**

对于给定 OpenShift 项目中的基于 Operator 的代理部署，如果您使用 `oc get pod` 命令列出代理 Pod，则 Pod 的 ordinal 值从 0 开始，如 `amq-operator-test-broker-s-0`。但是，如果您使用 `oc describe` 命令获取从 `activemqartmises` 自定义资源（即 `oc describe activemqartemises`）创建的代理 Pod 的状态，则 Pod ordinal 值从 1 不正确启动，例如 `amq-operator-test-broker-ss-1`。无法解决这个问题。

- **ENTMQBR-4127 - AMQ Broker Operator:Operator 生成的路由名称可能太长于 OpenShift**

对于基于 Operator 的部署中的每个代理 Pod，Operator 创建用于访问 AMQ Broker 管理控制台的路由的默认名称包括自定义资源(CR)实例的名称、OpenShift 项目的名称，以及 OpenShift 集群的名称。例如：`my-broker-deployment-wconsj-0-svc-rte-my-openshift-project.my-openshift-domain`。如果其中有些名称很长，则默认 Route 名称可能会超过 OpenShift 强制执行的 63 个字符的限制。在本例中，在 OpenShift Container Platform Web 控制台中，Route 会显示 **Rejected** 状态。

要临时解决这个问题，使用 OpenShift Container Platform Web 控制台手动编辑 Route 的名称。在控制台中，点 Route。在右上角的 **Actions** 下拉菜单中，选择 **Edit Route**。在 YAML 编辑器中，找到 **spec.host** 属性并编辑值。

- **ENTMQBR-4140 - AMQ Broker Operator: 如果 storage.size 没有被指定，则安装将不可用**  
如果您将自定义资源(CR)实例的 **storage.size** 属性配置为在部署持久性存储中指定代理所需的持久性卷声明(PVC)的大小，则 Operator 安装将无法使用。例如，假设您将 **storage.size** 的值设置为 **1**（即，不指定单位）。在这种情况下，Operator 无法使用 CR 创建代理部署。另外，即使删除了 CR，并部署一个带有正确指定的 **storage.size** 的新版本，Operator 仍然无法使用此 CR 按预期创建部署。

要临时解决这个问题，首先停止 Operator。在 OpenShift Container Platform web 控制台中点 **Deployments**。对于与 AMQ Broker Operator 对应的 Pod，点 **More options** 菜单（三个垂直点）。点 **Edit Pod Count**，将值设为 **0**。当 Operator Pod 停止后，创建正确指定 **storage.size** 的 CR 的新版本。然后，要重启 Operator，再次点击 **Edit Pod Count** 并将值设为 **1**。

- **ENTMQBR-4141 - AMQ Broker Operator: 增加持久性卷大小需要即使在重新创建 Stateful 设置后手动参与**  
如果您尝试增加部署持久性存储中的代理所需的持久性卷声明(PVC)的大小，则更改不会在进一步手动步骤的情况下生效。例如，假设您配置自定义资源(CR)实例的 **storage.size** 属性来指定 PVC 的初始大小。如果您将 CR 修改为指定不同的 **storage.size** 值，现有的代理将继续使用原始 PVC 大小。即使将部署扩展到零个代理，然后再备份到原始数字，也是如此。但是，如果您扩展部署的大小以添加额外的代理，则新代理使用新的 PVC 大小。

要临时解决这个问题，并确保部署中的所有代理都使用相同的 PVC 大小，使用 OpenShift Container Platform Web 控制台扩展部署使用的 PVC 大小。在控制台中，点击 **Storage** → **Persistent Volume Claims**。点您的部署。在右上角的 **Actions** 下拉菜单中，选择 **Expand PVC** 并输入一个新值。

## 第 8 章 重要链接

- [Red Hat AMQ Broker 7.7 发行注记](#)
- [Red Hat AMQ Broker 7.6 发行注记](#)
- [Red Hat AMQ Broker 7.1 到 7.5 发行注记\(aggregated\)](#)
- [Red Hat AMQ 7 支持的配置](#)
- [Red Hat AMQ 7 组件详情](#)

修订到 2022-03-19 14:20:09 +1000