



Red Hat AMQ Broker 7.11

Red Hat AMQ Broker 7.11 发行注记

AMQ Broker 发行注记

法律通告

Copyright © 2024 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.

摘要

本发行注记包含有关 AMQ Broker 7.11 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

使开源包含更多	3
第 1 章 AMQ BROKER 7.11 的长期支持	4
第 2 章 支持的配置	5
第 3 章 新的和更改的功能	6
第 4 章 已弃用的功能	14
第 5 章 删除的功能	15
第 6 章 修复的问题	16
第 7 章 修复了常见漏洞和暴露的问题	17
第 8 章 已知问题	18
第 9 章 重要链接	21

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。有关更多详情，请参阅[我们的首席技术官 Chris Wright 提供的消息](#)。

第 1 章 AMQ BROKER 7.11 的长期支持

AMQ Broker 7.11 被指定为 Long Term Support (LTS) 发行版本。有关 LTS 发行版本条款的详情，请参阅 [AMQ LTS 版本的支持时间？](#)

支持 Red Hat Enterprise Linux 和 OpenShift Container Platform

AMQ Broker 7.11 LTS 版本支持：

- Red Hat Enterprise Linux 7、8 和 9
- OpenShift Container Platform 4.12, 4.13, 4.14 或 4.15

红帽努力确保 AMQ Broker 与将来的 OpenShift Container Platform 版本保持兼容，但无法保证这个兼容性。为每个新的 OpenShift Container Platform 版本执行互操作性测试。如果没有找到兼容性问题，新的 OpenShift Container Platform 版本会添加到 [Red Hat AMQ Broker 7 支持的配置](#) 中。

第 2 章 支持的配置

有关支持的配置的详情，请参考 [Red Hat AMQ Broker 7 支持的配置](#)。

最低 Java 版本

至少，AMQ Broker 7.11 需要 Java 版本 11 运行。

OpenWire 支持

AMQ 7 Broker 从 2017 年发布以来提供对 Openwire 协议的支持，作为将客户端应用程序迁移到 AMQ 7 的方法。在 2021 年发布 AMQ Broker 7.9.0 后，Openwire 协议已被弃用，并鼓励客户将现有 Openwire 客户端应用程序迁移到 AMQ 7 的完全支持协议之一(CORE、AMQP、MQTT 或 STOMP)。从 AMQ Broker 8.0 发行版本开始，Openwire 协议将从 AMQ Broker 中删除。

第 3 章 新的和更改的功能

本节介绍了 AMQ Broker 7.11 中突出显示的一组增强功能和新功能。

CR 中的 image 和 version 属性的验证更改

在 7.11.5 中，Operator 验证 CR 中的 image 和 version 属性的配置与之前的版本不同。

在 7.11.5 之前，Operator 会验证 CR 没有：

- 没有 `spec.deploymentPlan.initImage` 属性的 `spec.deploymentPlan.image` 属性，反之亦然。
- 带有 `spec.deploymentPlan.image` 和 `spec.deploymentPlan.initImage` 属性的 `spec.version` 属性，或两者。

在 7.11.5 中，Operator 继续验证 CR 没有 `spec.deploymentPlan.image` 属性，但没有 `spec.deploymentPlan.initImage` 属性，反之亦然。在 7.11.5 中，Operator 验证被修改为同时验证：

- `spec.version` 属性中指定的版本号（如果存在）与部署的代理容器镜像版本匹配。如果版本不同，Operator 会将 CR 中的 `BrokerVersionAligned` 条件的状态设置为 `Unknown` 以突出显示不匹配的信息，但不匹配不会影响部署中运行的代理。
- CR 没有 `spec.deploymentPlan.image` 和 `spec.deploymentPlan.initImage` 属性，没有 `spec.version` 属性。如果 CR 的 `spec.deploymentPlan.image` 和 `spec.deploymentPlan.initImage` 属性没有 `spec.version` 属性，Operator 会将 CR 中的 `Valid` 条件的状态设置为 `Unknown` 以警告配置不完整。



注意

缺少 `spec.version` 属性会导致 Operator 每次升级 Operator 时在部署中重启代理 Pod。Pod 重启是必需的，因为 Operator 使用最新支持的代理版本更新 StatefulSet 中的标签，除非在 `spec.version` 属性中明确设置了版本号。

要防止每个将来的 Operator 升级都重启代理，您必须设置 CR 中的 `spec.version` 属性中部署的代理的版本号。

- 在 7.11.5 中，您可以在启动代理后在 CR 的 `status` 部分找到部署的代理的版本号。如需更多信息，[请参阅在 OpenShift 上部署 AMQ Broker 中的查看代理部署的状态信息](#)。
- 在 7.11.5 之前，您可以使用 OpenShift Container Platform Web 控制台在代理 Pod 的日志文件中找到版本号：
 - i. 单击 Workloads → Pods。
 - ii. 点代理 Pod 名称。
 - iii. 点 Logs 选项卡。
版本号显示在日志输出顶部的 Artemis 横幅后。



注意

具有 `Unknown` 状态值为 `Unknown` 的条件不会阻止 Operator 完成代理部署。

可以自定义 Operator 的领导选举设置

从 7.11.5 开始，您可以自定义 Operator 用于领导选举机制的设置。如果使用 Operator Hub 安装 Operator，您可以使用 OpenShift Container Platform Web 控制台在安装 Operator 后在 Operator 订阅中配置领导选举设置。如果使用 OpenShift Container Platform 命令行界面安装 Operator，您可以在 Operator 配置文件 `operator.yaml` 中配置领导选举设置，可以是在安装 Operator 之前或之后。

以下是在 `operator.yaml` 文件中配置的领导选举设置示例：

```
apiVersion: apps/v1
kind: Deployment
...
template
...
spec:
containers:
- args:
- --leader-elect
- --lease-duration=60
- --renew-deadline=40
- --retry-period=5
...
```

以下是 Operator 订阅中配置的领导选举设置示例：

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
...
spec:
...
config:
env:
- name: ARGS
value: "--leader-elect --lease-duration=60 --renew-deadline=60 --retry-period=5"
```

leader-elect：让 Operator 能够与其他实例一起竞争领导选举机制，以确保一次没有多个实例运行。

lease-duration：持续时间（以秒为单位），非领导 Operator 会在尝试获取之前领导未续订的租期前等待。默认值为 15。

renew-deadline：领导 Operator 在尝试续订领导角色前等待的持续时间（以秒为单位）。默认值为 10。

retry-period：Operator 在尝试获取和更新领导角色之间等待的时间（以秒为单位）。默认值为 2。

可以更新 `address-settings` 元素中的单个属性

在 7.11.5 中，您可以使用 Jolokia REST 接口以 JSON 格式更新 `address-settings` 元素中的独立属性。在以前的版本中，如果要更新 `address-settings` 元素中的属性或属性子集，则需要在 `update` 操作中包含所有未更改的 `address-settings` 属性。

警告级别消息在 Critical Analyzer 中变为错误级别

在 7.11.5 中，Critical Analyzer 为之前分配了 **WARN** 级别的消息分配一个 **ERROR** 级别。

新的参数，以更好地控制页面消息流到内存中

在 7.11.3 之前，您可以通过设置 `max-read-page-bytes` 和 `max-read-page-messages` 参数的限值来控制 paged 消息流。在应用这些限制时，代理会计算内存中的消息，这些消息已准备好发送到当前提供的消费者和消息。如果消费者需要确认消息，则当前交付的消息可以使用内存或消息限制，这会阻止

代理将新消息读取在内存中。因此，代理可能会耗尽消息。

从 7.11.3 开始，您可以为两个新参数设置限值，以控制分页消息流到内存中。在应用这些限制时，代理不会考虑发送消息。

- 预抓取页字节 内存（以字节为单位），可用于将页面消息读取到每个队列的内存。默认值为 20MB。
- **Prefetch-page-messages** Number of paged 消息，代理可以从磁盘读取到每个队列的内存。默认值为 -1，这意味着没有应用任何限制。

如果消费者确认消息较慢，您可以增加 `max-read-page-bytes` 和 `max-read-page-message` 参数的默认限制，以提供提供消息的容量。然后，`prefetch-page-bytes` 和 `prefetch-page-messages` 参数的默认限制允许代理将新消息读取到内存中。



注意

如果在 `prefetch-page-bytes` 参数的值前达到 `max-read-page-bytes` 参数的值，代理将停止将进一步页面的消息读入内存中。

AMQ 核心协议 JMS 客户端可以切换到集群中的其他实时代理

在 7.11.2 之前，如果 AMQ Core Protocol JMS 客户端丢失与 live 代理的连接，则只能在两个代理都配置为实时/备份对以进行高可用性(HA)时，它才能切换到备份代理。从 7.11.2 开始，AMQ Core Protocol JMS 客户端可以配置为切换到 HA 对中的备份代理或集群中的任何其他实时代理。

要使客户端能够切换到集群中的任何实时代理，请在客户端的连接 URI 中指定 `failoverAttempts` 配置选项。例如：

```
ConnectionFactory connectionFactory = new
ActiveMQConnectionFactory("(tcp://host1:port,tcp://host2:port,tcp://host3:port,tcp://host4:port,tcp://host5:port)?ha=true&failoverAttempts=2&reconnectAttempts=2");
```

将 `failoverAttempts` 选项设置为 2 时，客户端会尝试连接到集群中的 2 个其他实时代理。如果客户端在非 HA 配置以及 HA 配置中的 live 和 backup 代理都失败，则会发生故障转移尝试。



注意

示例中所示的 `reconnectAttempts` 配置选项仅在 HA 对配置了两个代理时从 live 代理切换到备份代理。

当 AMQ Broker 使用 JDBC 持久性时，分页性能会改进

从 7.11.2 开始，当 AMQ Broker 配置为使用 JDBC 持久性时，会提高分页性能。作为分页改进的一部分，一个新的参数 `jdbc-max-page-size-bytes`，在使用 JDBC 时，将页大小限制为 100KB。您可以自定义默认限制。如需更多信息，请参阅 [配置 AMQ Broker](#) 中的 [配置 JDBC 持久性](#)。

联邦消息的批处理

如果您的队列上的积压超过本地使用者的可用容量，则任何低优先级联邦消费者都会成为接收消息的候选者。最终，太多消息可能会移到联邦消费者，并导致在其他集群中发生相同的场景，因此消息在代理间移动的结果。

从 7.11.2 开始，您可以将联邦消费者配置为仅在本地队列超过容量时拉取消息的批处理。因此，联邦不会移动超过联邦消费者可以处理的消息，这样可避免消息在代理间移动的情况。

要将联邦消费者配置为拉取消息的批处理，请在联邦消费者的连接 URI 上将 `consumerWindowSize` 值设置为 0。

```
tcp://<host>:<port>?consumerWindowSize=0
```

将 `consumerWindowSize` 值设置为 0 时，AMQ Broker 使用匹配地址的地址设置中的 `defaultConsumerWindowSize` 属性的值来确定在代理间移动消息的批处理大小。`defaultConsumerWindowSize` 属性的默认值为 1048576 字节。

使用此批量操作模式用于活跃代理之间的双向联邦。

有关联邦的更多信息，请参阅[配置 AMQ Broker](#)中的联邦[地址和队列](#)。

代理启动健康检查

从 7.11.2 开始，您可以配置启动探测来检查 OpenShift Container Platform 容器中 AMQ Broker 应用程序是否已成功启动。要了解如何配置健康检查，请参阅在 [OpenShift 上部署 AMQ Broker](#) 中的[配置代理健康检查](#)。

限制用于分页的磁盘空间

如果将 AMQ Broker 配置为页面信息，您可以限制用于页面传入消息的磁盘空间，以防止分页操作使用过量磁盘空间。如需更多信息，请参阅[配置 AMQ Broker](#)中的[特定地址在分页过程中限制磁盘用量](#)。

限制用于从分页读取的消息的内存

如果将 AMQ Broker 配置为页面信息，您可以在客户端准备好使用消息时限制代理从磁盘传输到内存的消息的内存。如需更多信息，请参阅[配置 AMQ Broker](#)中的[将页面消息流控制到内存中](#)。

重要

如果客户端应用程序缺少太多消息待处理确认，代理不会在确认待处理消息前读取页面消息，这会导致代理上出现消息不足。

例如，如果将页面消息传输到内存（默认为 20MB）的限值被访问，则代理会在读取任何更多消息前等待来自客户端的确认。如果同时，客户端在向代理发送确认前等待接收足够消息（由客户端使用的批处理大小决定），代理会耗尽消息。

为避免不足，可以将控制页面消息传输的代理限制增加到内存中，或减少传递消息的数量。您可以通过确保客户端很快提交消息确认，或使用超时和提交确认（不从代理接收更多消息）来减少发送消息的数量。

您可以在 AMQ 管理控制台中看到在队列的 `Delivering Count` 和 `Delivering Bytes` 指标中传递消息的编号和大小。

Log4j 2 日志记录支持

从 7.11 开始，AMQ Broker 使用 Log4j 2 日志记录工具，而不是 JBoss Logging 框架来提供消息日志记录。您可以在 OpenShift Container Platform 和 RHEL 平台上自定义 Log4j 2 日志记录配置。

更改 Operator 日志记录级别

在 OpenShift Container Platform 上的 AMQ Broker 7.11 中，您可以更改默认日志级别，以增加或减少写入 Operator 日志的详细信息。如需更多信息，请参阅在 [OpenShift 上部署 AMQ Broker](#) 中的[更改 Operator 的日志级别](#)。

支持 Java 身份验证和授权服务(JAAS)登录模块

在 OpenShift Container Platform 上的 AMQ Broker 7.11 中，您可以在 `secret` 中配置 JAAS 登录模块，而不使用 `ActiveMQArtemisSecurity CR` 为 AMQ Broker 配置用户身份验证和授权。通过在 `secret` 中配置 JAAS 登录模块，您可以在属性文件中更新用户和角色信息，而无需重启代理。另外，

您可以配置在 CR 中不可配置的登录模块，如 LDAP。如需更多信息，请参阅在 *Openshift 上部署 AMQ Broker* 中的在 [secret 中配置 JAAS 登录模块](#)。

限制升级

在 OpenShift Container Platform 上的 AMQ Broker 7.11 中，Operator 会自动将代理容器镜像升级到最新的可用版本。您可以为部署配置自定义资源(CR)，以防止自动升级，或者只允许自动升级到特定版本或特定代理和 init 容器镜像。



注意

如果要限制自动升级，则必须在 CR 中指定组合的 `spec.deploymentPlan.image` 和 `spec.deploymentPlan.initImage` 属性，但不能同时指定 CR 中的 `spec.deploymentPlan.image` 和 `spec.deploymentPlan.initImage` 属性。

如需更多信息，请参阅在 *Openshift 上部署 AMQ Broker* 中的 [限制自动升级](#)。

扩展状态报告

在 OpenShift Container Platform 上的 AMQ Broker 7.11 中，Operator 在主代理 CR 中报告的状态信息被扩展为包括：

- CR 内容的有效性。
- 在 `brokerProperties` 属性中配置的属性应用。
- 将 secret 中的 Java 身份验证和授权服务(JAAS)登录模块文件传播到代理 Pod。
- 部署的代理版本，以及该版本的代理和 init 容器镜像的 URL。
- 将主要、次版本、补丁和安全升级应用到部署的能力。

同步镜像支持

从 7.11 开始，您可以在代理之间配置同步镜像，以确保消息同时写入镜像中的所有代理卷。通过使用同步镜像，您可以确保已镜像的代理是灾难恢复的最新代理。如需更多信息，请参阅 [配置 AMQ Broker 中的配置代理连接](#)。

Pod 中断预算

在 OpenShift Container Platform 上的 AMQ Broker 7.11 中，您可以配置 Pod 中断预算，以指定集群中必须同时可用的最少的 Pod 数量，如维护窗口。如需更多信息，请参阅在 *OpenShift 上部署 AMQ Broker* 中的 [配置 Pod 中断预算](#)。

brokerProperties 配置存储在 mutable secret 中

在 OpenShift Container Platform 上的 AMQ Broker 7.11 中，使用 CR 中的 `brokerProperties` 属性创建的配置存储在 mutable secret 中。可以在不需要代理重启的情况下更新可变 secret。因此，代理定期重新载入配置时应用配置更新，除了特别需要代理重启的任何更新外。

控制嵌入式 Web 服务器的操作

从 7.11 开始，您可以通过在 ActiveMQServerControl JMX MBean 中使用 `stopEmbeddedWebServer`、`startEmbeddedWebServer` 和 `restartEmbeddedWebServer` 操作作为 AMQ Broker 停止和重启嵌入式 Web 服务器。通过使用这些操作，您可以避免在更新 AMQ Broker 的 SSL 证书时重启 AMQ Broker。

用于登录到 AMQ 管理控制台的凭证用于发送消息

在之前的 AMQ Broker 版本中，需要在 AMQ Management Console Preferences 页面中指定用户名和密码，以便在 AMQ 管理控制台中发送消息。从 7.11 开始，消息会使用您用来登录到 AMQ 管理控制台的凭据发送。

您可以覆盖默认行为，并指定不同的凭证来发送单个消息。如需更多信息，请参阅 [管理 AMQ Broker 中的向地址](#) 发送信息。

OpenShift Container Platform 上的 AMQ Broker 已预先配置，以便为 Prometheus 收集指标数据

在 OpenShift Container Platform 上的 AMQ Broker 7.11 中，AMQ Broker 容器 Pod 已预先配置，以便 Prometheus Operator Service Monitor 收集指标数据。在以前的版本中，您需要公开 Service Monitor 需要的端口来收集指标数据。

为代理容器设置环境变量

在 OpenShift Container Platform 上的 AMQ Broker 7.11 中，您可以在传递给每个代理容器的自定义资源(CR)中设置环境变量。例如，您可以添加 TZ 环境变量来设置代理容器的时区。如需更多信息，请参阅 [在 OpenShift 上部署 AMQ Broker 中的为代理容器设置环境变量](#)。

OpenShift Container Platform 上的代理转发支持

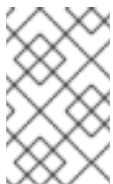
在 OpenShift Container Platform 上的 AMQ Broker 7.11 中，嵌入的 Web 服务器（托管 AMQ 管理控制台）已预先配置，以处理 X-Forwarded 标头。通过处理 X-Forwarded 标头，AMQ 管理控制台可以接收当涉及请求路径时更改或丢失的标头信息。例如，AMQ 管理控制台使用 HTTP，但 OpenShift Container Platform 路由器（一个代理）使用路由器终止的 HTTPS 路由公开 AMQ 管理控制台。从 X-Forwarded 标头中，AMQ 管理控制台可以识别浏览器和路由器之间的连接使用 HTTPS 并切换到 HTTPS 来提供浏览器请求。

只有在 brokerProperties CR 属性中才支持一些重新发送属性

如果 7.8.x 或 7.9.x 部署有在 spec.deploymentPlan.addressSettings.addressSetting CR 元素中配置的 redeliveryDelayMultiplier 或 redeliveryCollisionAvoidanceFactor 属性，则在升级到 7.11.x 后，您必须在 brokerProperties CR 属性中配置这些属性。这是必要的，因为这两个属性的数据类型从浮点改为 7.10.0 中的字符串。因此，spec.deploymentPlan.addressSettings.addressSetting 属性中不再支持这些属性。

以下示例演示了如何在 brokerProperties 元素中配置这两个属性：

```
spec:
  ...
  brokerProperties:
  - "addressSettings.#.redeliveryMultiplier=2.1"
  - "addressSettings.#.redeliveryCollisionAvoidanceFactor=1.2"
  ...
```



注意

在 brokerProperties 属性中，使用 redeliveryMultiplier 属性名称而不是 spec.deploymentPlan.addressSetting 元素中使用的 redeliveryDelayMultiplier 属性名称。

禁用 XML 外部实体(XXE)处理

如果您不需要将代理配置嵌套到 broker.xml 文件包含的独立文件中，您现在可以禁用 XXE 处理来防止 XXE 安全漏洞。红帽建议您尽可能禁用 XXE 处理。如需更多信息，请参阅 [配置 AMQ Broker 中的禁用外部 XML 实体\(XXE\)处理](#)。

JGroups 5.x

7.10.0 使用 JGroups 3.x 前的 AMQ Broker 版本。AMQ Broker 7.11 使用 JGroups 5.x，它与 JGroups 3.x 向后兼容。在两个 JGroup 版本之间更改了一些协议和协议属性，因此您必须在升级到 AMQ Broker 7.11 时更改 JGroups 堆栈配置。

提取到特定目录名称的 AMQ Broker 归档内容

当您在 Red Hat Enterprise Linux 上提取 AMQ Broker 归档时，归档的内容将提取到当前目录中名为 apache-artemis-2.28.0.redhat-00022 的目录。

Operator 频道

AMQ Broker Operator ([Red Hat Integration - AMQ Broker for RHEL 8 \(Multiarch\)](#))可用于以下频道：

- 7.11.x - 此频道只为版本 7.11 提供更新，它是一个长期支持(LTS)频道。
- 7.10.x - 此频道只为版本 7.10 提供更新，它是一个长期支持(LTS)频道。



注意

无法通过切换频道来升级 Operator。您必须卸载现有的 Operator，并从适当的频道安装 Operator 的新版本。

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

进入 Prometheus 指标插件的类名称

在 AMQ Broker 7.11 中，AMQ Broker 中包含的 Prometheus 指标插件的类名称从 `org.apache.activemq.artemis.core.server.metrics.plugins.ArtemisPrometheusMetricsPlugin` 改为 `com.redhat.amq.broker.core.server.metrics.plugins.ArtemisPrometheusMetricsPlugin`。如果在 AMQ Broker 的早期版本中启用了 Prometheus metrics 插件，您必须在升级到 AMQ Broker 7.11 时更新 `broker.xml` 配置文件中的类名称。如需更多信息，请参阅 [管理 AMQ Broker 中的将代理实例从 7.10.x 升级到 7.11.x](#)。

更改了列表 Producers API 方法返回的数据，以及列表 ProducersInfoAsJSON JMX 方法

在 AMQ Broker 7.11 中，以下改进是如何列出 Producers 方法返回的数据，该方法由 AMQ 管理控制台使用：

- 在之前的版本返回的数据中，每个会话会显示一个制作者。因此，如果您使用 Core 协议创建了一个制作者，它会为每个处理程序创建两个会话，则会为每个会话显示单独的制作者。另外，如果您在不从制作者发送消息的情况下创建了制作者，则为制作者返回的地址为空。在 AMQ Broker 7.11 中，列表 Producers 方法返回由核心协议创建的两个会话的单一制作者。此外，`address` 列会显示正确的地址，即使在发送消息之前也是如此。
- 在以前的版本中，当使用匿名制作者使用 Core 或 AMQP 协议向多个地址发送消息时，每个地址会显示生成者。此外，显示的地址是制作者向其发送消息的第一个地址，它类似于发送到单个队列的常规制作者。在 AMQ Broker 7.11 中，当您使用匿名制作者向多个地址发送消息时，会为每个匿名制作者显示一个制作者。此外，该地址没有连接到特定地址，其值为 `ANONYMOUS`。

在以前的版本中，`listProducersInfoAsJSON` 方法提供了特定会话发送到每个队列的消息计数。但是，此方法会错误地返回消息发送到的每个队列的制作者。例如，如果匿名制作者向 1000 个队列发送一条消息，此方法会返回 1000 个制作者。在 AMQ Broker 7.11 中，`listProducersInfoAsJSON` 方法现在准确返回与 `listProducers` 方法相同的数据，但采用不同的格式。

在 AMQ Broker 7.11 中，返回以下新指标数据：

消费者

`messagesInTransit` - 尚未确认的消息数

`messagesInTransitSize` - 尚未确认的消息的总大小

`messagesDelivered` - 发送的消息数

`messagesDeliveredSize` - 发送的消息总量

messagesAcknowledged - 确认的信息总数

messagesAcknowledgedAwaitingCommit - 已确认的事务中消息总数，但等待提交

lastDeliveredTime - 发送的最后一个消息的时间（毫秒）

lastAcknowledgedTime - 确认最后一次消息的时间（毫秒）

producers

msgSent - 生成者发送的消息数量

msgSizeSent - 生成者发送的消息总量

lastProducedMessageID - 发送的最后一个消息的 ID

第 4 章 已弃用的功能

本节论述了支持但已从 AMQ Broker 中弃用的功能。

自定义资源中的升级 属性

从 7.11 开始，升级 属性和关联的 `enabled` 和 `minor` 属性已弃用，因为它们无法象最初设计一样工作。使用 `image` 或 `version` 属性来部署特定的代理容器镜像。

queues 配置元素

从 7.10 开始，`<queues>` 配置元素已弃用。您可以使用 `<addresses>` 配置元素来创建地址和相关队列。`<queues>` 配置元素将在以后的发行版本中删除。

getAddressesSettings method

从 7.10 开始，`getAddressSettings` 方法（包含在 `org.apache.activemq.artemis.core.config.Configuration` 接口）已被弃用。使用 `getAddressSettings` 方法，以编程方式为代理配置地址和队列。

OpenWire 协议

从 7.9 开始，OpenWire 协议是一个已弃用的功能。如果您要创建新的基于 AMQ Broker 的系统，请使用其它支持的协议之一。在 8.0 发行版本中，Openwire 协议将从 AMQ Broker 中删除。

在代理实例没有运行时添加用户

从 7.8 开始，当 AMQ Broker 实例没有运行时，将从 CLI 接口将用户添加到代理的功能。

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的影响，这可能会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，红帽不再建议在新部署中使用网络 ping。有关为高可用性配置代理集群的指导，并避免网络隔离问题，请参阅 [配置 AMQ Broker](#) 中的 [实现高可用性](#)。

Hawtio 分配控制台插件

从 7.3 开始，AMQ Broker 不再附带 Hawtio 分配控制台插件，`dispatch-hawtio-console.war`。在以前的版本中，分配控制台用于管理 AMQ Interconnect。但是，AMQ Interconnect 现在使用自己的独立 Web 控制台。

第 5 章 删除的功能

本节论述了从 AMQ Broker 中删除的功能和功能。

访问 AMQ Broker web 服务器的根目录

在之前的 AMQ Broker 版本中，在浏览器中打开 AMQ Broker Web 服务器的根 URL，例如 <http://localhost:8161/>，浏览器窗口中会显示一个登录页面。登录页面有 AMQ 管理控制台和 AMQ Broker 文档的链接。在 7.11 中，所有静态 HTML 内容都会从 AMQ Broker 中删除。因此，如果您打开 AMQ Broker Web 服务器的根 URL，则不会显示登录页面。相反，浏览器会话会自动重定向到 AMQ 管理控制台。

第 6 章 修复的问题

有关版本中修复的问题的完整列表，请参阅 [AMQ Broker 7.11.0 修复的问题](#)，并参阅 [AMQ Broker - 7.11.x 解决的问题](#) 列表。

第 7 章 修复了常见漏洞和暴露的问题

本节详细介绍了 AMQ Broker 7.11 发行版本中修复的通用漏洞和暴露(CVE)。

- [ENTMQBR-6630](#) - CVE-2022-1278 WildFly: 可能的信息泄漏
- [ENTMQBR-7397](#) - CVE-2022-22970 springframework: 通过数据绑定到 multipartFile 或 servlet 部分的 DoS
- [ENTMQBR-7398](#) - CVE-2022-22971 springframework: STOMP over WebSocket
- [ENTMQBR-7005](#) - CVE-2022-2047 jetty-http:改进器主机名输入处理
- [ENTMQBR-7640](#) - CVE-2022-3782 keycloak: 路径遍历双 URL 编码

第 8 章 已知问题

本节论述了 AMQ Broker 7.11 中已知的问题。

- [ENTMQBR-8106](#) - 在更改 CR 中的 MessageMigration 后 AMQ Broker Drainer pod 无法正常工作
您无法更改正在运行的代理部署中 messageMigration 属性的值。要临时解决这个问题，您必须新的 ActiveMQ Artemis CR 中为 messageMigration 属性设置所需的值，并创建新的代理部署。
- [ENTMQBR-8166](#) - 带有 UseClientAuth=true 的自签名证书可防止与 Jolokia 的 Operator 通信
如果在 ActiveMQ Artemis CR 的 console 部分中将 useClientAuth 属性设置为 true，Operator 无法在代理上配置某些功能，如创建地址。在 Operator 日志中，您会看到以 remote error: tls: bad certificate 结尾的错误消息。
- [ENTMQBR-7385](#) - 缓慢消费者上的联邦队列的消息 flops
如果本地应用程序消费者非常慢或无法消耗消息，可以在应用程序消费者最终消耗前多次发送一条消息，并在联邦连接中发送大量信息。
- [ENTMQBR-7820](#) - [Operator] supported version in 7.11.0 OPR1 operator 日志不正确
Operator 日志列出对以下 AMQ Broker 镜像版本的支持：7.10.0 7.10.1 7.10.2 7.11.0 7.8.1 7.8.2 7.8.3 7.9.0 7.9.1 7.9.2 7.9.3 7.9.4。从 7.10.0 开始，Operator 实际上支持 AMQ Broker 镜像版本。
- [ENTMQBR-7359](#) - 使用 7.10.0 Operator 更改为当前处理凭证 secret
Operator 在 secret 中存储用于连接代理的管理员用户名和密码。默认 secret 名称的格式是 `<custom-resource-name>-credentials-secret`。您可以手动创建 secret，或允许 Operator 创建 secret。

如果在 7.10.0 之前在自定义资源中配置 adminUser 和 adminPassword 属性，Operator 会使用这些属性的值更新手动创建的 secret。从 7.10.0 开始，Operator 不再更新手动创建的 secret。因此，如果更改了 CR 中的 adminUser 和 adminPassword 属性的值，则必须：

- 使用新用户名和密码更新 secret
- 删除 secret，并允许 Operator 创建 secret。当 Operator 创建 secret 时，如果在 CR 中指定，它会添加 adminUser 和 adminPassword 属性的值。如果这些属性不在 CR 中，Operator 会为 secret 生成随机凭证。
- [ENTMQBR-7111](#) - 7.10 版本的 operator 会在升级过程中删除 StatefulSet
如果您要升级到 AMQ Broker Operator 7.10.0，新的 Operator 会在协调过程中自动删除每个部署的现有 StatefulSet。当 Operator 删除 StatefulSet 时，现有代理 pod 会被删除，这会导致临时代理中断。

您可以运行以下命令来手动删除 StatefulSet 并孤立正在运行的 pod，然后删除 StatefulSet: oc delete statefulset < statefulset-name > ; --cascade=orphan

在升级过程中手动删除 StatefulSet 可让新 Operator 在不删除正在运行的 pod 的情况下协调 StatefulSet。如需更多信息，请参阅在 *OpenShift 上部署 AMQ Broker* 中的 [使用 OperatorHub 升级 Operator](#)。

- [ENTMQBR-6473](#) - 由于模式 URL 更改而不兼容的配置
当您尝试使用之前版本 7.9 或 7.10 实例的代理实例时，不兼容的配置会导致代理崩溃。要临时解决这个问题，请更新相关配置文件中的模式 URL，如在 [Linux 上从 7.9.0 升级到 7.10.0 所述](#)。
- [ENTMQBR-4813](#) AsynchronousCloseException 带有大型消息和多个 C++ 订阅者

如果多个使用 AMQP 协议的客户端与订阅者和代理在同一台主机上运行，并且发布者发送大型消息，其中一个订阅者会崩溃。

- [ENTMQBR-5749](#) - 删除 OperatorHub 中不支持的 operator
仅支持从 OperatorHub 部署 Operator 中的 Operator 和 Operator 频道。出于与 Operator 出版物相关的技术原因，其他 Operator 和频道在 OperatorHub 中可见，并应该被忽略。对于参考，以下列表显示了哪些 Operator 是可见的，但不被支持：
 - Red Hat Integration - AMQ Broker LTS - 所有频道
 - Red Hat Integration - AMQ Broker - alpha、current 和 current-76
- [ENTMQBR-569](#) - 将 ID 从 OpenWire 转换为 AMQP 会导致将 ID 作为二进制发送
当从 A-MQ 6 OpenWire 客户端通信到 AMQP 客户端时，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信息，可以被忽略。
- [ENTMQBR-655](#) - [AMQP] Unable to send message wherpopulate-validated-user
配置选项 populate-validated-user 不支持使用 AMQP 协议生成的消息。
- [ENTMQBR-1875](#) - [AMQ 7, ha, replicated store] 备份代理在 - ActiveMQIllegalStateException errorType=ILLEGAL_STATE message=AMQ119026: Backup Server 尚未与 live 同步后无法进入"实时"或关闭
当备份代理试图与主代理同步时，删除主代理的分页磁盘会导致主代理失败。另外，备份代理无法变为 live，因为它继续尝试与主代理同步。
- [ENTMQBR-2068](#) - 有些在 HA 故障切换过程中收到但未发送的消息，故障切换场景
目前，如果在 OpenWire 客户端发送消息时代理切换到其备份，则当故障转移丢失时发送到代理的消息。要临时解决这个问题，请确保代理在确认信息前保留信息。
- [ENTMQBR-3331](#) - Stateful set controller 无法从 CreateContainerError 恢复，阻止 Operator
如果 AMQ Broker Operator 从具有配置错误的自定义资源(CR)创建一个有状态的集合，有状态集控制器在错误解决时无法推出更新的有状态集。

例如，主代理 CR 中的 image 属性的值出现错误，从而导致有状态集控制器创建的第一个 Pod 的状态保持 Pending。如果修复错误，并应用 CR 更改，AMQ Broker Operator 会更新有状态的集合。但是，Kubernetes 已知问题可防止有状态的集合控制器推出更新的有状态集。控制器无限期等待具有 Pending 状态的 Pod 变为 Ready，因此不会部署新的 Pod。

要临时解决这个问题，您必须删除具有 Pending 状态的 Pod，以允许有状态集控制器部署新 Pod。要检查哪些 Pod 具有 Pending 状态，请使用以下命令：`oc get pods --field-selector=status.phase=Pending`。要删除 Pod，请使用 `oc delete pod <pod name>` 命令。

- [ENTMQBR-3846](#) - mq 客户端不会在代理重启时重新连接
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，要重新连接 mq 客户端，您需要在客户端上手动调用 `subscribe ()` 方法。
- [ENTMQBR-4127](#) - AMQ Broker Operator: Operator 生成的路由名称可能太长用于 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 管理控制台的路由名称包括自定义资源(CR)实例的名称、OpenShift 项目的名称和 OpenShift 集群的名称。例如，`my-broker-deployment-wconsj-0-svc-rte-my-openshift-project.my-openshift-domain`。如果其中一些名称长，则默认路由名称可能会超过 OpenShift 强制的 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，请使用 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` 未指定，则安装将不可用
如果您将自定义资源(CR)实例的 `storage.size` 属性配置为指定持久性存储部署中代理所需的持久性卷声明(PVC)的大小，如果没有正确指定这个值，Operator 安装将不可用。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，即使您删除了 CR 并部署正确指定的 `storage.size` 的新版本，Operator 仍然无法使用此 CR 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）。单击 Edit Pod Count，并将值设为 0。当 Operator Pod 停止后，创建正确指定了 `storage.size` 的 CR 的新版本。然后，要重启 Operator，请再次点 Edit Pod Count，并将值设为 1。

- **ENTMQBR-4141** - AMQ Broker Operator: Increasing Persistent Volume size 需要手动参与，即使重新创建 Stateful Set
如果您尝试在为持久性存储的部署中增加代理所需的持久性卷声明(PVC)的大小，则更改不会在进行进一步的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，以便为 PVC 指定初始大小。如果您修改 CR 以指定 `storage.size` 的不同值，现有代理将继续使用原始 PVC 大小。即使将部署缩减为零代理，然后备份到原始数字，也是如此。但是，如果您扩展部署的大小以添加额外的代理，新代理将使用新的 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，请使用 OpenShift Container Platform Web 控制台扩展部署使用的 PVC 大小。在控制台中，点 Storage → Persistent Volume Claims，点您的部署。在右上角的 Actions 下拉菜单中，选择 **Expand PVC** 并输入新值。

第 9 章 重要链接

- [Red Hat AMQ Broker 7.10 发行注记](#)
- [Red Hat AMQ Broker 7.9 发行注记](#)
- [Red Hat AMQ Broker 7.8 发行注记](#)
- [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 \(聚合\)](#)
- [Red Hat AMQ 7 支持的配置](#)
- [Red Hat AMQ 7 组件详情](#)

更新于 2024-10-18