Red Hat AMQ Broker 7.10

Red Hat AMQ Broker 7.10 发行注记

AMQ Broker 发行注记

Last Updated: 2024-06-11

Red Hat AMQ Broker 7.10 Red Hat AMQ Broker 7.10 发行注记

AMQ Broker 发行注记

法律通告

摘要

本发行注记包含有关 AMQ Broker 7.10 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

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 7.10 的长期支持

AMQ Broker 7.10 已被指定为 Long Term Support(LTS)发行版本。有关 LTS 发行版本术语的详细信息，请参阅 AMQ LTS 发行版本的支持时间？

支持 Red Hat Enterprise Linux 和 OpenShift Container Platform

AMQ Broker 7.10 LTS 版本支持：

- Red Hat Enterprise Linux 7 和 8
- 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 版本兼容，但无法保证这种兼容性。每个新的 OpenShift Container Platform 版本都会进行互操作性测试。如果没有找到兼容性问题，新的 OpenShift Container Platform 版本将添加到 Red Hat AMQ Broker 7 支持的配置 。

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 。

注意

至少，AMQ Broker 7.10 需要 Java 版本 11 运行。

第 3 章 新的和更改的功能

这部分论述了 AMQ Broker 7.10 中突出显示的改进和新功能。

高可用性复制改进

在以前的 AMQ Broker 版本中，若要使用复制高可用性(HA)策略，至少需要三个实时备份代理对。需要三个对，因此对对一个对都必须获得多数仲裁投票，并避免两个代理同时存在的情况。从 7.10 开始，您可以将代理配置为使用 Apache Zookeeper 协调服务来协调每个实时备份代理对，从而消除了至少具有三个实时备份对的需要。如需更多信息，请参阅配置代理集群以使用配置 *AMQ Broker* 中的 ZooKeeper 协调服务复制高可用性。

客户端连接分区

在以前的版本中，无法对服务器端分区客户端连接进行分区。从 7.10 开始，您可以对客户端连接进行分区，这需要在客户端在每次发起连接时将各个客户端的路由连接到同一代理。分区客户端连接的两个用例是：

- 对持久订阅进行分区，以确保订阅者始终连接到位于 durable subscriber 队列所在代理。
- 最大程度减少在代理间移动数据的需要，方法是吸引客户端进入其源自的数据，也称为数据获取性。
  如需了解更多有关分区客户端连接的信息，请参阅 *配置 AMQ Broker 中的 分区 客户端连接*。

AMQ 管理控制台身份验证

要使用 AMQ 管理控制台验证用户，您可以配置基于证书的验证。要了解如何配置基于证书的验证，请参阅配置代理控制台以使用 配置 *AMQ Broker* 中的基于证书的身份验证。

控制节点上的 pod 放置

在基于 operator 的代理部署中，您可以使用节点选择器、节点关联性规则和污点(tolerations)配置自定义资源(CR)来控制 OpenShift Container Platform 节点上的 AMQ Broker pod 的放置。如需更多信息，请参阅在 OpenShift *上部署 AMQ Broker* 中的 OpenShift Container Platform 节点上控制代理 pod 放置。

代理健康检查

在基于 operator 的代理部署中，您可以使用存活度和就绪度探测在正在运行的代理容器上配置定期健康检查。存活度探测通过 ping 代理的 HTTP 端口来检查代理是否在运行。就绪度探测(Readiness probe)通过打开与为代理配置的每个接受器端口的连接来检查代理是否可以接受网络流量。要了解如何配置健康检查，*请参阅在 OpenShift 的 Deploying AMQ Broker* 中配置 代理健康检查。

覆盖默认内存限值

在基于 operator 的代理部署中，您可以覆盖为代理设置的默认内存限值。默认情况下，代理被分配一个一半内存，内存可用于代理的 Java 虚拟机。要了解如何覆盖默认内存限制，请参阅在 *OpenShift 上部署 AMQ Broker* 中的 代理的默认内存限值。

在持久性卷声明(PVC)中请求存储类

默认情况下，OpenShift Container Platform 上的 AMQ Broker 的任何持久性卷声明(PVC)都使用为集群配置的默认存储类。在这个版本中，您可以配置 CR 为 AMQ Broker 指定存储类。要了解如何在 PVC 中指定存储类，*请参阅在 OpenShift 中部署 AMQ Broker* 中的配置代理 存储大小和存储类。

为 pod 配置安全上下文

在基于 operator 的代理部署中，您可以为 pod 配置安全上下文。安全上下文为 pod 定义特权和访问控制设置，包括自由裁量访问控制、Security Enhanced Linux(SELinux)、安全计算模式(seccomp)、sysctl 接口和 Window 在 Windows 上运行的容器的特定属性。如需更多信息，*请参阅在 OpenShift 中部署 AMQ Broker* 中的 自定义资源配置参考。

支持 OpenShift Container Platform 4.15

除了支持 OpenShift Container Platform 4.6、4.7、4.8、4.9 和 4.10 外，AMQ Broker 还支持 OpenShift Container Platform 4.15。

更改代理 pod 的默认服务帐户名称

您可以使用 **serviceAccountName** name 属性来更改代理 pod 的默认服务帐户名称。如需更多信息，*请参阅在 OpenShift 中部署 AMQ Broker* 中的 自定义资源配置参考。

标记代理 pod

您可以使用 **labels** 属性为代理 pod 分配标签。如需更多信息，*请参阅在 OpenShift 中部署 AMQ Broker* 中的 自定义资源配置参考。

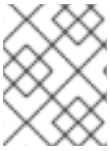
使用 *StoreType 和 *StoreProvider 更新接收器和连接器配置

在接收器和连接器的 CR 配置中，您可以指定代理使用的密钥存储和信任存储的详细信息。

Operator 频道

AMQ Broker Operator、**Red Hat Integration - AMQ Broker for RHEL 8(Multiarch)** 包括以下频道：

- **7.10.x** - 此频道仅为版本 7.10 提供更新，并且是当前的 Long Term Support (LTS)频道。
- **7.x** - 当前，此频道只为版本 7.9 提供更新。
- **7.8.x** - 此频道仅为版本 7.8 提供更新，并且是之前的 Long Term Support (LTS)频道。



注意

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

要确定要选择哪个 Operator，请参阅 Red Hat Enterprise Linux Container Compatibility Matrix 。

使用通配符值，通过 Management API 授予所有域的访问权限

从 **management.xml** 文件中的 7.10.1 开始，您可以在 **条目 domain** 字段中指定一个通配符值。当您访问管理 API 时，**entry domain** 字段中的通配符值授予所有域的访问权限。

```
<authorisation>
  <allowlist>
    <entry domain="*"/>
  </allowlist>
```

JGroups 5.x

以前的 AMQ Broker 版本使用 JGroups 3.x。AMQ Broker 7.10 使用 JGroups 5.x，它不与 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 7.10 时更改 JGroups 堆栈配置。

第 4 章 已弃用的功能

本节介绍了支持的功能，但已从 AMQ Broker 中弃用。

队列 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，get**Addresses**Settings 方法（包含在 org.apache.activemq.artemis.core.config.Configuration 接口）已被弃用。使用 get**Address**Settings 方法为代理程序配置地址和队列。

OpenWire 协议

从 7.9 开始，OpenWire 协议是一个已弃用的功能。如果要创建新的 AMQ Broker 的系统，请使用其他受支持的协议之一。这个功能将在以后的发行版本中被删除。

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，会从 CLI 接口将用户添加到代理。

网络 pinger

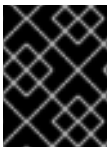
从 7.5 开始，网络 ping 是一项已弃用的功能。网络 ping 无法保护代理集群不受网络隔离问题而导致无法恢复的消息丢失。这个功能将在以后的发行版本中被删除。红帽继续支持使用网络 ping 的现有 AMQ Broker 部署。但是，红帽不会在新部署中使用网络 ping。有关为高可用性配置代理集群并避免网络隔离问题的指导，请参阅 *配置 AMQ Broker* 中的 高可用性。

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，valued **-hawtio-console.war**。在以前的版本中，发送控制台用于管理 AMQ Interconnect。但是，AMQ 互联现在使用自己的独立 Web 控制台。

第 5 章 技术预览

本节论述了 AMQ Broker 7.10 中的技术预览功能。



重要

红帽产品服务级别协议(SLA)不支持技术预览功能，且可能无法完成。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。

为分页配置地址限制的新属性

您可以配置以下新属性，以根据消息数量设置个人和全局地址限制。

max-size-messages 是代理执行为 address-full-policy 指定的策略前允许的最大信息数。默认值为 -1，这意味着没有消息限制。

global-max-messages 是代理可用于所有地址的消息总数。当达到这个限制时，对于与传入消息关联的地址，代理执行指定为 address-full-policy 的值的策略。默认值为 -1，这意味着没有消息限制。

注意

如果为 **max-size-bytes** 或 **global-max-size** 属性设置的限制，则在为 **max-size-message** 或 **global-max-messages** 属性设置限值前，代理执行 address-full 策略。

第 6 章 修复的问题

有关发行版本中已解决的完整问题列表，请参阅链接： AMQ Broker 7.10.0 Fixed 问题 ， 并参阅 AMQ Broker - 7.10.x Resolved 问题 列表， 了解补丁版本中已修复的问题列表。

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 7.10 发行版本中解决的安全漏洞漏洞和暴露(CVE)。

- ENTMQBR-5140 - CVE-2019-10744 nodejs-lodash: pollution in defaultDeep 功能，从而导致修改属性
- ENTMQBR-5893 - CVE-2021-4040 broker: AMQ Broker: Malformed message 可能会导致部分 DoS(OOM)
- ENTMQBR-5933 - CVE-2021-43797 netty: control chars in header name may cause HTTP request smuggling
- ENTMQBR-6401 - CVE-2022-23913 artemis-commons: Apache ActiveMQ Artemis DoS
- ENTMQBR-6477 - CVE-2020-36518 jackson-databind: 通过大量嵌套对象深入拒绝服务

第 8 章 已知问题

这部分论述了 AMQ Broker 7.10 中已知的问题。

- ENTMQBR-7359 - **使用 7.10.0 Operator 开始当前处理凭证 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 名称采用 < ***custom-resource-name>*** **-credentials-secret** 的形式。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，您必须：

  - 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，如果它们在 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-7363 - **从 7.9 CR 的 AddressSettingsType 中的 redeliveryDelayMultiplier 无法被协调**
  如果 **redeliveryDelayMultiplier** 和 **redeliveryCollisionAvoidanceFactor** 属性在 7.8.x 或 7.9.x 部署中的主代理 CR 中配置，新的 Operator 在升级到 7.10.x 后无法协调任何 CR。协调失败，因为两个属性的数据类型从 float 改为 7.10.x 中的字符串。

  您可以通过删除 **spec.deploymentPlan.addressSettings.addressSetting** 元素的 **redeliveryDelayMultiplier** 和 **redeliveryCollisionAvoidanceFactor** 属性来解决这个问题。然后，在 **brokerProperties** 元素中配置属性。例如：

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

  注意

  在 **brokerProperties** 元素中，使用 **redeliveryMultiplier** 属性名称而不是您删除的 **redeliveryDelayMultiplier** 属性名称。

- ENTMQBR-7396 - **[Operator, upgrade] 升级到 7.10.1 无法创建新的 Acceptor/Connector *v1.ServiceAdmission**
  在从 AMQ Broker 7.10.0 升级到 7.10.1 后，如果在升级过程中没有删除错误的服务标签和在 7.10.0 中的 Pod 选择器，则消息传递将停止工作。如果升级后消息传递没有工作，请完成以下步骤来解决这个问题。

  1. 以集群管理员身份登录 OpenShift Container Platform Web 控制台。
  2. 在页面顶部的 Project 下拉菜单中选择安装 Operator 的项目。
  3. 在左侧导航菜单中，单击 **Networking → Services**。
  4. 在 **Labels** 和 **Pod Selectors** 列中，检查是否配置了 **ActiveMQArtemis** 和 **application** 以外的任何服务。

5. 对于配置了 **ActiveMQArtemis** 和 **application** 以外的标签的每个服务，请完成以下步骤以删除该标签。

   a. 单击服务以打开 **Details** 选项卡。

   b. 在 **Labels** 字段中，单击 **Edit**，并删除 **ActiveMQArtemis** 和 **application** 标签外的所有标签。

   c. 点击 **Save**。

   d. 点 **YAML** 标签。

   e. 在 **selector** 元素中，删除 **ActiveMQArtemis**、应用程序和 **statefulset.kubernetes.io/podname** 标签之外的所有标签。

   f. 点击 **Save**。

- ENTMQBR-7111 - **7.10 版本会在升级过程中删除 StatefulSet**
  如果您要升级到 AMQ Broker Operator 7.10.0，新 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 并孤立运行的 pod，从而删除 StatefulSet: oc delete statefulset < *statefulset-name*& gt; --cascade=orphan

  在升级过程中手动删除 StatefulSet 可让新 Operator 协调 StatefulSet，而不删除正在运行的 pod。如需更多信息，请参阅在 *OpenShift 上部署 AMQ Broker* 中使用 OperatorHub 升级 Operator。

- ENTMQBR-6991 - **7.10-opr-3 不会为 7.10-opr-2 用户修复 PV ownerRef**
  如果您部署或升级到 7.10.0-opr-2 并扩展部署，则会使用 **ownerReference** 属性创建新的 PV，如果稍后删除部署 CR，则会导致数据丢失。例如，如果您部署 7.10.0-opr-1，升级到 7.10.0-opr-2，然后从 3 扩展到 4 代理实例，如果您删除 **ActiveMQArtemis** CR，则可能会丢失数据。

  要解决这个问题，您可以：

  - 跳过 7.10.0-opr-2 升级（如果可能）。
  - 在集群中已激活 7.10.0-opr-2 版本时，避免扩展部署。您可以在部署 7.10.0-opr-3 后扩展。
  - 避免删除部署 CR，直到后续发行版本解决了这个问题。
  - 手动删除受影响 PV 的 **ownerReference** 值。

- ENTMQBR-6712 - **Taints 和 Tolerations - "tolerationSeconds" 会破坏部署**
  如果您在 CR 的 **tolerations** 部分添加 **tolerationSeconds** 属性，Operator 协调过程无法正常工作，且代理 pod 不会被正确调度。要临时解决这个问题，在 CR 的 **tolerations** 部分不要在 **tolerationSeconds** 属性中添加 tolerationSeconds 属性。

- ENTMQBR-6473 - **因为 schema URL 的变化而兼容配置**
  当您尝试使用带有版本 7.9 或 7.10 实例的之前的版本中的代理实例配置时，schema URL 改变会导致代理崩溃。要临时解决这个问题，在从 7.9.0 升级到 7.10.0 时，将相关配置文件中的 schema URL 更新至 7.10.0。

- ENTMQBR-4813 **AsynchronousCloseException，带大型消息和多个 C++ 订阅者**
  如果多个使用 AMQP 协议的 C++ publisher 客户端与订阅者和代理一样运行，且发布者会发送一个大型消息，订阅者崩溃。

- ENTMQBR-6655 - **命令 artemis 检查队列失败，并显示 "Could not start Jolokia agent"**
  在运行之前，**artemis check queue** 命令显示以下错误消息： **Could not start Jolokia agent: java.lang.IllegalStateException: Cannot open keystore for https 通信： java.net.BindException: Address already in use**.

- ENTMQBR-6654 - **requireLogin:true 仅适用于应用新的代理 CR，而不适用于现有代理 CR。**
  如果在 CR 中将 **requireLogin** 属性设置为 **true**，则 AMQ_REQUIRE_LOGIN 环境变量不会在现有代理实例集合中更新，并且控制台凭证不会被验证。要临时解决这个问题，请手动更新现有实例的有状态集的环境变量值。

- ENTMQBR-5936 - **如果 URL 以非集群端口为目标，客户端不会故障切换到备份服务器。**
  如果客户端用于连接 HA 集群的连接 URL 具有没有在代理的 **static-connectors** 中配置的端口，在故障切换发生后，客户端会重试到之前 live 代理的连接，且不会尝试连接到新的实时代理。

- ENTMQBR-6728 - **升级路径无法正常工作**
  这个问题可防止升级到 **7.x** 频道的 AMQ Broker 7.9 用户自动升级到 AMQ Broker 7.10。要解决这个问题，请订阅 **7.10.x** 频道。

- ENTMQBR-5749 - **删除 OperatorHub 中可见的、不支持的 operator**
  仅支持 通过 OperatorHub 部署 Operator 中的 Operator 和 Operator 频道。对于与 Operator 发布相关的技术原因，其他 Operator 和频道在 OperatorHub 中可见，并应忽略。为了参考，以下列表显示了哪些 Operator 可见，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-17 - **AMQ222117: Unable to start cluster connection**
  在支持 IPv6 的环境中，代理集群可能无法正确初始化。故障是由于 **SocketException** 所指示**的，它通过日志消息不分配请求的地址**。要临时解决这个问题，请将 **java.net.preferIPv4Stack** 系统属性设置为 **true**。

- ENTMQBR-520 - **不允许与绑定到另一个地址的队列相同的地址接收**
  名称与地址相同的队列必须仅被分配到地址。创建名称与现有地址相同的队列，但绑定到具有不同名称的地址是一个无效的配置。这样做可能会导致错误消息路由到队列。

- ENTMQBR-569 - **将 OpenWire 的 ID 转换为 AMQP 会导致将 ID 发送为二进制**
  当将跨协议从 A-MQ 6 OpenWire 客户端与 AMQP 客户端通信时，会在应用程序消息属性中对额外的信息进行编码。这是代理在内部使用的信息，并可以忽略。

- ENTMQBR-636 - Journal breaks, causeing**JavaNullPointerException**, under perf load(mpt)
  为了防止在代理管理大量负载时发生 IO 相关问题，请验证 JVM 已分配有足够内存和堆空间。请参阅 ActiveMQSOURCE 文档的 Performance Tuning 一章中标题为 "Tuning the VM" 的部分。

- ENTMQBR-648 - **JMS Openwire 客户端无法通过定义的purgeOnNoConsumer 或 queue过滤器将消息发送到队列**
  如果队列没有消费者，使用 A-MQ 6 JMS 客户端将消息发送到带有 **purgeOnNoConsumer** 设置为 **true** 的队列的地址，会失败。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 MBeans 用于代理和队列相关的问题。

  代理 MBean 的问题：

- 关闭连接会引发 **java.net.SocketTimeoutException** 异常
- **listSessions()** throws **java.lang.ClassCastException**
- 添加地址设置会抛出 **java.lang.IllegalArgumentException**
- **getConnectorServices（）** 操作无法找到
- 无法找到 **列表ConsumersAsJSON（）** 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（）** 抛出参数类型不匹配异常
- **listDeliveringMessages（）** throws **IllegalArgumentException**
- **listMessages（）** throws **java.lang.Exception**
- **moveMessages（）** 抛出 **IllegalArgumentException**，显示错误消息参数类型不匹配
- **removeMessage（）** 抛出 **IllegalArgumentException**，并显示错误消息参数类型不匹配
- **removeMessages（）** 引发异常，error Can't find operation removeMessage with 2 参数
- **retryMessage（）** 抛出参数类型不匹配 **IllegalArgumentException**

- ENTMQBR-655 - [AMQP] Unable to send message when**populate-validated-user** is enabled
  对于使用 AMQP 协议生成的消息，配置选项 **populate-validated-user** 不被支持。

- ENTMQBR-897 - **在目标名称中带有特殊字符的 Openwire 客户端/协议问题**
  目前，AMQ OpenWire JMS 客户端无法访问队列和地址，它们的名称中包含以下字符： comma comma(',')、hash('#')、大于('>')和空格。

- ENTMQBR-944 - **[A-MQ7, Hawtio, RBAC] 用户在 RBAC 拒绝操作时不会获得任何反馈**
  控制台可能会表示未授权用户尝试的操作在未成功时获得成功。

- ENTMQBR-1875 - **[AMQ 7, ha, replicated store] backup 代理看不到"live"或关机 - ActiveMQIllegalStateException errorType=ILLEGAL_STATE message=AMQ119026：备份服务器尚未与实时同步**
  在备份代理尝试与 master 代理同步时，删除 master 代理的分页磁盘会导致 master 失败。另外，备份代理无法变为 live，因为它继续尝试与 master 同步。

- ENTMQBR-2068 - **在 HA 故障切换过程中收到一些消息但不提供。**
  目前，如果在 OpenWire 客户端发送信息时代理故障转移到它的从属服务器，则当故障转移丢失时向代理传输的消息。要临时解决这个问题，请确保代理会在确认信息前保留它们。

- ENTMQBR-3331 - **Stateful set controller 不能从 CreateContainerError 中恢复，阻止 Operator**
  如果 AMQ Broker Operator 从具有配置错误的自定义资源(CR)创建有状态集，则有状态集控制器无法在错误解决时回滚更新的有状态集。

例如，在主代理 CR 中 **image** 属性的值中拼写错误，会导致有状态集控制器创建的第一个 Pod 的状态保持 **Pending**。如果修复错误拼写并应用 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部署更新的有状态集。控制器会无限期地等待具有 **Pending** 状态变为 **Ready** 的 Pod，因此不会部署新的 Pod。

要临时解决这个问题，您必须删除状态为 **Pending** 的 Pod，以便有状态集控制器部署新 Pod。要检查哪些 Pod 是否有 **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 - **PlacementBinding 客户端在代理重启时不会重新连接**
  当您重启代理或者代理故障切换时，活跃代理不会恢复之前连接的 awx 客户端的连接。要临时解决这个问题，要重新连接 expertise 客户端，您需要在客户端上手动调用 **subscribe（）** 方法。

- ENTMQBR-4023 - **AMQ Broker Operator: Pod Status pod 名称不会反映实际情况**
  对于给定的 OpenShift 项目中的基于 Operator 的代理部署，如果使用 **oc get pod** 命令列出代理 Pod，则 Pod 的 ordinal 值从 **0** 开始，例如 **amq-operator-test-broker-s-0**。但是，如果您使用 **oc describe** 命令获取从 **activemqartmises** 自定义资源创建的代理 Pod 的状态（即 **oc describe activemqartemises**），则 Pod 或dinal 值在 **1** 中错误启动，例如 **amq-operator-test-broker-s-1**。无法解决这个问题。

- ENTMQBR-4127 - **AMQ Broker Operator: Operator 生成的 Route 名称对于 OpenShift 可能太长**
  对于基于 Operator 的部署中的每个代理 Pod，Operator 为访问 AMQ Broker 管理控制台创建的 Route 的默认名称包括自定义资源(CR)实例的名称、OpenShift 项目名称和 OpenShift 集群的名称。例如，**my-broker-deployment-wconsj-0-svc-rte-my-openshift-project.my-openshift-domain**。如果其中有些名称很长，则默认路由名称可能会超过 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，Router 会显示 **拒绝** 的状态。

  要临时解决这个问题，使用 OpenShift Container Platform Web 控制台手动编辑路由的名称。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 不当指定，安装将变得不可用**
  如果您配置自定义资源(CR)实例的 **storage.size** 属性，以指定部署的持久性卷声明(PVC)大小，如果没有为持久性存储指定这个值，Operator 安装将无法使用。例如，假设您将 **storage.size** 的值设置为 1（即不指定单元）。**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 并使用正确的 storage.size 部署新版本，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）。点 Edit Pod Count，将值设为 0。当 Operator Pod 停止后，创建带有正确指定 storage.size 的 CR 的新版本。然后，要重启 Operator，请再次点击 Edit Pod Count，并将值设为 1。

- ENTMQBR-4141 - **AMQ Broker Operator: Increasing Persistent Volume size 需要手动参与，即使恢复 Stateful 设置**

  如果您尝试增加部署用于持久性存储的代理所需的持久性卷声明(PVC)，则更改不会起作用，而无需进一步手动步骤。例如，假设您配置自定义资源(CR)实例的 storage.size 属性，以指定

PVC 的初始大小。如果您修改 CR 来指定 storage.size *的不同值*，则现有代理将继续使用原始 PVC 大小。即使将部署缩减到零代理，然后再备份到原始数量，也是如此。但是，如果您扩展部署的大小来添加额外的代理，新代理会使用新的 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。点击您的部署。在右上角 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 7.9 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(aggregated)
- Red Hat AMQ 7 支持的配置
- Red Hat AMQ 7 组件详情

更新于 2024-06-11