Red Hat AMQ 2020.Q4

RHEL 上的 AMQ Streams 1.6 发行注记

用于 Red Hat Enterprise Linux 上的 AMQ Streams

Last Updated: 2023-02-23

Red Hat AMQ 2020.Q4 RHEL 上的 AMQ Streams 1.6 发行注记

用于 Red Hat Enterprise Linux 上的 AMQ Streams

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

摘要

这些发行注记包含 AMQ Streams 1.6 发行版中包含的新功能、增强、修复和问题的最新信息。

目录

第 1 章 功能

本发行版本中添加的功能，以及之前 AMQ Streams 版本中没有的功能，如下所述。

注意

要查看此版本中已解决的所有增强和错误，请查看 AMQ Streams Jira 项目 。

1.1. AMQ STREAMS 1.6.X 中的 KAFKA 支持（长期终止支持）

本节论述了 AMQ Streams 1.6 以及后续补丁版本中支持的 Kafka 和 ZooKeeper 版本。

AMQ Streams 1.6.x 是 Long Term Support 发行版本，可用于 RHEL 7 和 8。

有关 AMQ LTS 版本支持日期的信息，请参阅红帽知识库解决方案 AMQ LTS 版本的支持时间？

仅支持由红帽构建的 Kafka 发行版本。对于升级目的，AMQ Streams 1.6.x 支持早期版本的 Kafka。

有关支持的 Kafka 版本的更多信息，请参阅 客户门户网站中的 Red Hat AMQ 7 组件详情页 。

1.1.1. Kafka 支持 AMQ Streams 1.6.6 和 1.6.7

AMQ Streams 1.6.6 和 1.6.7 发行版本支持 Apache Kafka 版本 2.6.3。

有关升级说明，请参阅 AMQ Streams 和 Kafka 升级 。

如需了解更多信息，请参阅 Kafka 2.6.3 发行注记。

Kafka 2.6.3 需要 ZooKeeper 版本 3.5.9。因此，从 AMQ Streams 1.6.4 / 1.6.5 升级时，*您不需要* 升级 ZooKeeper。

1.1.2. AMQ Streams 1.6.4 和 1.6.5 中的 Kafka 支持

AMQ Streams 1.6.4 和 1.6.5 发行版本支持，并使用 Apache Kafka 版本 2.6.2 和 ZooKeeper 3.5.9。

有关升级说明，请参阅 AMQ Streams 和 Kafka 升级 。

如需更多信息，请参阅 Kafka 2.6.2 发行注记。

Kafka 2.6.2 需要 ZooKeeper 版本 3.5.9。因此，当从 AMQ Streams 1.6.0 升级时，您需要升级 ZooKeeper。

1.1.3. AMQ Streams 1.6.0 中的 Kafka 支持

AMQ Streams 1.6.0 支持并使用 Apache Kafka 2.6.0 版本。

有关升级说明，请参阅 AMQ Streams 和 Kafka 升级 。

如需更多信息，请参阅 Kafka 2.5.0 和 Kafka 2.6.0 发行注记。

注意

AMQ Streams 1.6.0 中只支持 Kafka 2.5.x 用于升级。

Kafka 2.6.0 需要与 Kafka 2.5.x 相同的 ZooKeeper 版本（ZooKeeper 版本 3.5.7 / 3.5.8）。因此，当从 AMQ Streams 1.5 升级时，*您不需要* 升级 ZooKeeper。

1.2. OAUTH 2.0 授权

对 OAuth 2.0 授权的支持从技术预览转变为 AMQ Streams 的通用组件。

如果您使用 OAuth 2.0 进行基于令牌的身份验证，您现在可以使用 OAuth 2.0 授权规则来限制客户端对 Kafka 代理的访问。

AMQ Streams 支持通过 Red Hat Single Sign-On Authorization Services 使用基于 OAuth 2.0 令牌的授权，它允许您集中管理安全策略和权限。

Red Hat Single Sign-On 中定义的安全策略和权限用于授予对 Kafka 代理上资源的访问权限。用户和客户端与允许对 Kafka 代理执行特定操作的策略进行匹配。

请参阅使用 OAuth 2.0 基于令牌的授权。

1.3. 开源策略代理(OPA)集成

开源策略代理(OPA)是一个开源策略引擎。您可以将 OPA 与 AMQ Streams 集成，作为基于策略的授权机制，允许在 Kafka 代理上进行客户端操作。

从客户端发出请求时，OPA 将根据为 Kafka 访问定义的策略评估请求，然后允许或拒绝请求。

您可以为 Kafka 集群、消费者组和主题定义访问控制。例如，您可以定义一个授权策略，允许从制作者客户端写入到特定代理主题的访问。

请参阅 **KafkaAuthorizationOpa** 模式参考

注意

- 红帽不支持 OPA 服务器。
- OPA 集成仅在 Open JDK 11 上受支持。

第 2 章 功能增强

下面概述了本发行版本中添加的增强功能。

2.1. KAFKA 增强

有关引入的增强的概述：

- Kafka 2.6.2，请参阅 Kafka 2.6.2 发行注记 （仅适用于 AMQ Streams 1.6.4）
- Kafka 2.6.1，请参阅 Kafka 2.6.1 发行注记 （仅适用于 AMQ Streams 1.6.4）
- kafka 2.6.0，请参阅 Kafka 2.6.0 发行注记

2.2. KAFKA 网桥的改进

此发行版本包括对 AMQ Streams 的 Kafka Bridge 组件的以下改进。

检索分区和元数据

Kafka Bridge 现在支持以下操作：

- 检索给定主题的分区列表：

```
GET /topics/{topicname}/partitions
```

- 检索给定分区的元数据，如分区 ID、领导代理和副本数：

```
GET /topics/{topicname}/partitions/{partitionid}
```

请参阅 Kafka Bridge API 参考。

支持 Kafka 消息标头

使用 Kafka Bridge 发送的消息现在可以包括 Kafka 消息标头。

在发送到 **/topics 端点** 的 POST 请求中，您可以选择在消息有效负载中指定标头，该标头包含在请求正文中。消息标头值必须采用二进制格式，并以 Base64 格式编码。

带有 Kafka 消息标头的请求示例

```
curl -X POST \
  http://localhost:8080/topics/my-topic \
  -H 'content-type: application/vnd.kafka.json.v2+json' \
  -d '{
    "records": [
        {
            "key": "my-key",
            "value": "sales-lead-0001"
            "partition": 2
            "headers": [
              {
                "key": "key1",
                "value": "QXBhY2hlIEthZmthIGlzIHRoZSBib21iIQ=="
              }
```

```
          ]
        },
    ]
}'
```

请参阅 对 Kafka 网桥的请求。

2.3. MIRRORMAKER 2.0 主题重命名更新

MirrorMaker 2.0 架构通过自动重命名远程主题来代表源集群来支持双向复制。原始集群的名称前面是主题名称的前面。

现在，您可以通过在源连接器配置中添加 **IdentityReplicationPolicy** 来覆盖自动重命名。应用此配置后，主题会保留其原始名称。

```
replication.policy.class= io.strimzi.kafka.connect.mirror.IdentityReplicationPolicy (1)
```

1. 添加可覆盖远程主题自动重命名的策略。该主题不会用源集群的名称来附加名称，而是保留其原始名称。

例如，覆盖在您要备份或将数据迁移到另一个集群的 *主动/被动* 集群配置中非常有用。在这两种情况下，您可能不希望自动重命名远程主题。

请参阅在 MirrorMaker 2.0 中使用 AMQ Streams

2.4. OAUTH 2.0 身份验证和授权

此发行版本包括对基于 OAuth 2.0 令牌的身份验证和授权的以下增强。

会话重新身份验证
AMQ Streams 中的 OAuth 2.0 身份验证现在支持 Kafka 代理的 *会话重新身份验证*。这定义了 Kafka 客户端和 Kafka 代理之间经过身份验证的 OAuth 2.0 会话的最长持续时间。会话重新身份验证支持两种类型的令牌验证：快速本地 JWT 和内省端点。

您可以在 **server.properties** 文件中为 Kafka 代理的 OAuth 2.0 配置中配置会话重新身份验证。

- 要应用到所有监听器，请以毫秒为单位设置 **connection.max.reauth.ms** 属性。
- 要应用到特定的监听程序，请设置 listen **.name.*LISTENER-NAME*.oauthbearer.connections.max.reauth.ms** 属性（毫秒）。*LISTENER-NAME* 是监听器不区分大小写的名称。

如果经过身份验证的会话超过配置的最大会话重新身份验证时间，或者达到访问令牌到期时间，则会关闭该会话。然后，客户端必须再次登录到授权服务器，获取新的访问令牌，然后重新身份验证到 Kafka 代理。这将在现有连接上创建一个新的经过身份验证的会话。

当下次需要重新身份验证时，客户端尝试的任何操作（不包括重新身份验证）将导致代理终止连接。

会话重新验证的监听程序在 6 分钟后配置示例

```
sasl.enabled.mechanisms=OAUTHBEARER
listeners=CLIENT://0.0.0.0:9092
# ...
listener.name.client.oauthbearer.sasl.jaas.config=org.apache.kafka.common.security.oauthbearer.OAut
```

```
hBearerLoginModule required \
  oauth.valid.issuer.uri="https://AUTH-SERVER-ADDRESS" \
  oauth.jwks.endpoint.uri="https://AUTH-SERVER-ADDRESS/jwks" \
  oauth.username.claim="preferred_username"  \
  oauth.client.id="kafka-broker" \
  oauth.client.secret="kafka-secret" \
  oauth.token.endpoint.uri="https://AUTH-SERVER-ADDRESS/token" ;
listener.name.client.oauthbearer.sasl.login.callback.handler.class=io.strimzi.kafka.oauth.client.JaasClien
OauthLoginCallbackHandler
listener.name.client.oauthbearer.connections.max.reauth.ms=3600000
```

请参阅：Kafka 代理的 会话重新身份验证以及 配置 Kafka 代理的 OAuth 2.0 支持。

JWKS 密钥刷新间隔

将 Kafka 代理配置为使用快速本地 JWT 令牌验证时，您现在可以在监听器配置中设置 **oauth.jwks.refresh.min.pause.seconds** 选项（在 **server.properties** 文件中）。这会定义代理尝试刷新授权服务器发布的 JSON Web 密钥集(JWKS)公钥尝试之间的最小间隔。

在这个版本中，如果 Kafka 代理检测到未知签名密钥，它会尝试立即刷新 JWKS 密钥并忽略常规刷新计划。

尝试刷新 JWKS 密钥之间暂停 2 分钟的示例

```
listener.name.client.oauthbearer.sasl.jaas.config=org.apache.kafka.common.security.oauthbearer.OAut
hBearerLoginModule required \
  oauth.valid.issuer.uri="https://AUTH-SERVER-ADDRESS" \
  oauth.jwks.endpoint.uri="https://AUTH-SERVER-ADDRESS/jwks" \
  oauth.jwks.refresh.seconds="300" \
  oauth.jwks.refresh.min.pause.seconds="120" \
  # ...
  oauth.ssl.truststore.type="PKCS12" ;
```

JWKS 密钥刷新调度在 **oauth.jwks.refresh.seconds** 选项中设置。当遇到未知的签名密钥时，会在刷新计划外调度 JWKS 密钥刷新。刷新将等到上次刷新达到 **oauth.jwks.refresh.min.pause.seconds** 中指定的间隔后才会开始。默认值为：**1**。

请参阅 为 Kafka 代理配置 OAuth 2.0 支持 。

从 Red Hat Single Sign-On 刷新授权

通过红帽单点登录，为基于 OAuth 2.0 令牌的授权添加了新的配置选项。在配置 Kafka 代理时，现在可以定义以下与从 Red Hat SSO 授权服务刷新授权相关的选项：

- **strimzi.authorization.grants.refresh.period.seconds**:连续两次的时间允许刷新运行。默认值为 **60**。如果设置为 **0** 或以下，则禁用刷新授权。
- **strimzi.authorization.grants.refresh.pool.size**:可以获取的线程数并行为活动会话授予。默认值为 **5**。

请参阅 使用基于 OAuth 2.0 令牌的授权和 配置 OAuth 2.0 授权支持

在 Red Hat Single Sign-On 中检测权限更改

在这个版本中，**KeycloakRBACAuthorizer** （红帽 SSO）授权会定期检查活动会话的权限更改。现在，可以实时检测中央用户和权限管理更改。

2.5. 在 KAFKA 管理工具中弃用 ZOOKEEPER 选项

以下 Kafka 管理工具中已弃用 **--zookeeper** 选项：

- **bin/kafka-configs.sh**
- **bin/kafka-leader-election.sh**
- **bin/kafka-topics.sh**

在使用这些工具时，现在应使用 **--bootstrap-server** 选项指定要连接的 Kafka 代理。例如：

```
/bin/kafka-topics.sh --bootstrap-server localhost:9092 --list
```

虽然 **--zookeeper** 选项仍然可以正常工作，但将在以后的 Kafka 发行版本中的所有管理工具中删除。这是 Apache Kafka 项目中持续工作的一部分，以消除 Kafka 对 ZooKeeper 的依赖。

在多个 流程中，RHEL 中使用 AMQ Streams 已更新为使用 **--bootstrap-server** 选项。

第 3 章 技术预览

重要

技术预览功能不被红帽产品服务级别协议(SLA)支持，且可能无法完成。因此，红帽不推荐在生产环境中实施任何技术预览功能。此技术预览功能为您提供对即将推出的产品创新的早期访问，允许您在开发过程中测试并提供反馈。如需有关支持范围的更多信息，请参阅技术预览功能支持范围。

3.1. 使用 CRUISE CONTROL 进行集群重新平衡

现在，您可以安装 Cruise Control，并使用它重新平衡 Kafka 集群。cruise Control 有助于减少运行高效、均衡 Kafka 集群所需的时间和工作量。

可以从 客户门户网站 下载 Cruise Control 的压缩分发包。要安装 Cruise Control，您需要将每个 Kafka 代理配置为使用提供的 Metrics Reporter。然后，您设置 Cruise Control 属性，包括优化目标，并使用提供的脚本启动 Cruise 控制。

Cruise Control 服务器托管在整个 Kafka 集群的单个机器上。

当 Cruise Control 运行时，您可以使用 REST API:

- 从多个优化目标生成空运行优化调整
- 启动优化建议以重新平衡 Kafka 集群

目前不支持其他 Cruise 控制功能，包括异常检测、通知、写入目标以及更改主题复制因素。

请参阅 Cruise Control for cluster 重新平衡。

第 4 章 已弃用的功能

AMQ Streams 1.6 没有弃用的功能。

第 5 章 修复的问题

以下小节列出了 AMQ Streams 1.6.x 中已解决的问题。如果您在 RHEL 7 和 8 中使用 AMQ Streams 1.6.x，红帽建议您升级到最新的补丁版本。

有关修复的问题的详情：

- Kafka 2.6.3，请参阅 Kafka 2.6.3 发行注记
- Kafka 2.6.2，请参阅 Kafka 2.6.2 发行注记
- Kafka 2.6.1，请参阅 Kafka 2.6.1 发行注记
- kafka 2.6.0，请参阅 Kafka 2.6.0 发行注记

5.1. 修复了 AMQ STREAMS 1.6.7 的问题

AMQ Streams 1.6.7 补丁发行版本(Long Term Support)现已正式发布。

AMQ Streams 1.6.7 是最新 Long Term Support 版本，可用于 RHEL 7 和 8。

有关 AMQ Streams 1.6.7 中问题的详情，请参阅 AMQ Streams 1.6.x 解决问题。

Log4j 漏洞

AMQ Streams 包括 log4j 1.2.17。发行版本修复了多个 log4j 漏洞。

有关本版本中解决漏洞的更多信息，请参阅以下 CVE 文章：

- CVE-2022-23307
- CVE-2022-23305
- CVE-2022-23302
- CVE-2021-4104
- CVE-2020-9488
- CVE-2019-17571
- CVE-2017-5645

5.2. 修复了 AMQ STREAMS 1.6.6 的问题

有关 AMQ Streams 1.6.6 中解决的问题的更多详情，请参阅 AMQ Streams 1.6.x Resolved 问题。

Log4j2 漏洞

AMQ Streams 包括 log4j2 2.17.1。此发行版本修复了多个 log4j2 漏洞。

有关本版本中解决漏洞的更多信息，请参阅以下 CVE 文章：

- CVE-2021-45046
- CVE-2021-45105

- CVE-2021-44832
- CVE-2021-44228

5.3. 修复了 AMQ STREAMS 1.6.5 的问题

有关 AMQ Streams 1.6.5 中解决问题的更多详情，请参阅 AMQ Streams 1.6.x 解析的问题。

Log4j2 漏洞

1.6.5 发行版本修复了使用 log4j2 的 AMQ Streams 组件的远程代码执行漏洞。如果系统从未授权来源记录字符串值，则该漏洞允许在服务器上执行远程代码。这会影响 2.0 和 2.14.1 之间的 log4j 版本。

如需更多信息，请参阅 CVE-2021-44228。

5.4. 修复了 AMQ STREAMS 1.6.4 的问题

有关 AMQ Streams 1.6.4 中解决问题的更多详情，请参阅 AMQ Streams 1.6.x 解析的问题。

5.5. 修复了 AMQ STREAMS 1.6.0 的问题

问题号	描述
ENTMQST-2049	Kafka 网桥：Kafka consumer 应该使用 group-consumerid 键进行跟踪
ENTMQST-2084	docs 上的 zookeeper 版本与 AMQ Streams 1.5 中的版本不匹配

第 6 章 已知问题

本节列出了 AMQ Streams 1.6 的已知问题。

问题号

ENTMQST-2030 - kafka-ack 报告 **javax.management.InstanceAlreadyExistsException: kafka.admin.client:type=app-info,id=<client_id> with client.id set**

描述

如果 **bin/kafka-acls.sh** 实用程序与 **--bootstrap-server** 参数结合使用以添加或删除 ACL，则操作会成功，但会生成警告。警告的原因是创建了第二个 **AdminClient** 实例。以后的 Kafka 发行版本中会解决这个问题。

第 7 章 支持的集成产品

AMQ Streams 1.6 支持与以下红帽产品集成：

- **Red Hat Single Sign-On 7.4 及更新的版本**用于 OAuth 2.0 身份验证和 OAuth 2.0 授权
- **Red Hat Debezium 1.0 及之后的版本** 用于监控数据库和创建事件流

有关这些产品可引入到您的 AMQ Streams 部署的功能信息，请参阅 AMQ Streams 1.6 文档。

第 8 章 重要链接

- Red Hat AMQ 7 支持的配置
- Red Hat AMQ 7 组件详情

修订到 2022 年 2 月 13 日 15:27:39 +1000