Red Hat AMQ 2021.Q3

OpenShift 中 AMQ Streams 1.8 发行注记

用于 OpenShift Container Platform 上的 AMQ Streams

Last Updated: 2023-02-23

Red Hat AMQ 2021.Q3 OpenShift 中 AMQ Streams 1.8 发行注记

用于 OpenShift Container Platform 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 1.8 发行版中包含的新功能、增强、修复和问题的最新信息。

目录

使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。我们从这四个术语开始： master、slave、blacklist 和 whitelist。这些更改将在即将发行的几个发行本中逐渐实施。如需了解更多详细信息，请参阅 CTO Chris Wright 信息。

第 1 章 功能

AMQ Streams 版本 1.8 基于 Strimzi 0.24.x。

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. OPENSHIFT CONTAINER PLATFORM 支持

OpenShift Container Platform 4.6 和 4.8 支持 AMQ Streams 1.8。

有关支持的平台版本的更多信息，请参阅红帽知识库文章 Red Hat AMQ 7 支持的配置 。

1.2. KAFKA 2.8.0 支持

AMQ Streams 现在支持 Apache Kafka 版本 2.8.0。

AMQ Streams 使用 Kafka 2.8.0。仅支持由红帽构建的 Kafka 发行版本。

您必须将 Cluster Operator 升级到 AMQ Streams 版本 1.8，然后才能将代理和客户端应用程序升级到 Kafka 2.8.0。有关升级说明，请参阅 升级 AMQ Streams。

如需更多信息，请参阅 Kafka 2.7.0 和 Kafka 2.8.0 发行注记。

注意

仅支持将 Kafka 2.7.x 升级到 AMQ Streams 1.8。

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

Kafka 2.8.0 版本需要 ZooKeeper 版本 3.5.9。因此，当从 AMQ Streams 1.7 升级到 AMQ Streams 1.8 时，Cluster Operator 会执行 ZooKeeper 升级。

1.3. 用来打开和关闭功能的功能门

作为 Kafka 集群管理员，您现在可以在 Operator 部署配置 *中打开和关闭功能* 子集。功能门当前仅适用于 Cluster Operator。以后的发行版本可能会向其他 Operator 添加功能门。

AMQ Streams 1.8 引入了以下功能门和关联的新功能：

- **ControlPlaneListener 来** 切换 control plane 侦听程序
- 用于切换 服务帐户补丁的 **ServiceAccountPatching**

功能门的默认状态为 *enabled 或 disabled*。启用后，功能门会更改 Operator 的行为，并在 AMQ Streams 部署中启用该功能。

功能门的成熟度等级为 *Alpha*、*Beta* 或正式 *可用* (GA)。

表 1.1. 功能门的成熟度

功能门成熟度等级	描述	默认状态
alpha	控制可能实验性、不稳定或未充分测试生产用途的功能。在以后的版本中可能会更改这些功能。	Disabled
Beta	控制经过良好测试的功能。在以后的版本中这些功能不太可能改变。	Enabled
正式发行（GA）	控制稳定、经过良好测试且适合生产用途的功能。GA 功能不会在以后的版本中改变。	Enabled

配置功能门

在 Cluster Operator 的部署配置中，在 **STRIMZI_FEATURE_GATES** 环境变量中指定以逗号分隔的功能门名称和前缀列表。**+** 前缀启用功能门，而 **-** 前缀可禁用它。

示例：启用 Control Plane Listener 功能门

1. 编辑 Cluster Operator 的 Deployment：

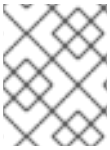
```
oc edit deployment strimzi-cluster-operator
```

2. 添加 **STRIMZI_FEATURE_GATES** 环境变量，值设为 **+ControlPlaneListener**

```
# ...
env:
 #...
- name: STRIMZI_FEATURE_GATES
  value: +ControlPlaneListener
#...
```

请参阅 Feature gates 和 Cluster Operator 配置。

1.4. CONTROL PLANE 侦听程序



注意

此功能使用 **ControlPlaneListener** 功能门控制，该功能为 alpha 阶段并默认禁用。如需更多信息，请参阅 功能门。

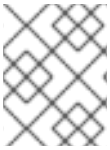
在标准的 AMQ Streams 集群中，control plane 流量和数据平面流量在端口 9091 上使用相同的代理间侦听程序。

在这个版本中，您可以配置集群，以便 control plane 流量在端口 9090 上使用专用的 *control plane 侦听程序*。数据平面流量继续在端口 9091 上使用监听程序。

使用 control plane 侦听器可能会提高性能，因为重要的控制器连接（如分区领导更改）不会因为代理间数据复制而延迟。大多数数据平面流量由此数据复制组成。

请参阅 Control plane 监听程序功能门。

1.5. 服务帐户补丁



注意

此功能使用 **ServiceAccountPatching** 功能门控制，该功能为 alpha 阶段，默认为禁用。如需更多信息，请参阅 功能门。

默认情况下，Cluster Operator 不会更新服务帐户。

在这个版本中，您可以在每次协调中启用对服务帐户的更新。例如，Cluster Operator 可向服务帐户应用自定义标签或注解。使用 **template.serviceAccount** 属性为自定义资源配置自定义标签和注解。

自定义标签和注解示例

```
# ...
template:
  serviceAccount:
    metadata:
      labels:
        label1: value1
        label2: value2
      annotations:
        annotation1: value1
        annotation2: value2
# ...
```

请参阅 服务帐户补丁功能门。

1.6. 用于更改数据捕获集成的 DEBEZIUM

红帽 Debezium 是一个分布式更改数据捕获平台。它捕获数据库中的行级更改，创建更改事件记录，并将记录流传输到 Kafka 主题。Debezium 基于 Apache Kafka 构建。您可以将 Debezium 与 AMQ Streams 部署和集成。部署 AMQ Streams 后，您可以通过 Kafka Connect 将 Debezium 部署为连接器配置。Debezium 将更改事件记录传递到 OpenShift 上的 AMQ Streams。应用程序可以读取 *这些更改事件流*，并按发生更改事件的顺序访问更改事件。

Debezium 具有多个用途，包括：

- 数据复制
- 更新缓存和搜索索引
- 简化单体式应用程序
- 数据集成
- 启用流查询

Debezium 为以下通用数据库提供连接器（基于 Kafka Connect）：

- Db2

- MongoDB
- MySQL
- PostgreSQL
- SQL Server

有关使用 AMQ Streams 部署 Debezium 的更多信息，请参阅 产品文档。

1.7. SERVICE REGISTRY

您可以将服务注册表用作数据流的集中式存储。对于 Kafka，您可以使用 Service Registry 来存储 *Apache Avro* 或 JSON 模式。

服务注册表提供 REST API 和 Java REST 客户端，用于通过服务器端端点从客户端应用注册和查询架构。

使用 Service Registry 将管理模式的过程与客户端应用程序配置分离。您可以通过在客户端代码中指定 URL 来启用应用程序从 registry 中使用 schema。

例如，消息序列化和反序列化的架构可以存储在注册表中，后者随后从使用它们的应用程序引用，以确保它们发送和接收的消息与这些模式兼容。

Kafka 客户端应用程序可以在运行时从 Service Registry 中推送或拉取其模式。

有关在 AMQ Streams 中使用 Service Registry 的更多信息，请参阅 Service Registry 文档。

第 2 章 升级要求

您必须升级自定义资源以使用 API 版本 **v1beta2**，然后才能升级到 AMQ Streams 版本 1.8。

AMQ Streams 1.7 中引入了所有自定义资源的 **v1beta2** API 版本。对于 AMQ Streams 1.8， **v1alpha1** 和 **v1beta1** API 版本已从除 **KafkaTopic 和 Kafka User** 之外的所有 AMQ Streams 自定义资源中删除。

将自定义资源升级到 **v1beta2** 准备 AMQ Streams 以迁移到 Kubernetes CRD **v1**，这是 Kubernetes v1.22 所必需的。

如果您要从 1.7 版本之前的 AMQ Streams 版本升级：

1. 升级到 AMQ Streams 1.7
2. 将自定义资源转换为 **v1beta2**
3. 升级到 AMQ Streams 1.8

请参阅 部署和升级 AMQ 流。

2.1. 将自定义资源升级到 V1BETA2 版本

为了支持将自定义资源升级到 **v1beta2**，AMQ Streams 提供了 *API 转换工具，您可以从 AMQ Streams 下载网站下载 该工具*。

您可以通过两个步骤执行自定义资源升级。

步骤一：转换自定义资源的格式

使用 API 转换工具，您可以通过以下两种方式之一将自定义资源格式转换为适用于 **v1beta2** 的格式：

- 转换描述 AMQ Streams 自定义资源配置的 YAML 文件
- 直接在集群中转换 AMQ Streams 自定义资源

另外，您可以手动将每个自定义资源转换为适用于 **v1beta2** 的格式。文档中包括了手动转换自定义资源的说明。

第 2 步：将 CRD 升级到 v1beta2

接下来，使用带有 **crd-upgrade** 命令的 API 转换工具，您必须将 **v1beta2** 设置为 CRD 中的 *存储* API 版本。您无法手动执行此步骤。

具体步骤请查看 升级 AMQ Streams。

第 3 章 增强

下面概述了本发行版本中添加的增强功能。

3.1. KAFKA 2.8.0 增强

有关 Kafka 2.8.0 引入的增强概述，请参阅 Kafka 2.8.0 发行注记。

3.2. KAFKA CONNECT 构建配置更新

您可以使用 **构建配置**，以便 AMQ Streams 自动构建带有数据连接所需的连接器插件的容器镜像。

现在，使用 Kafka Connect 构建 Pod 创建了一个专用服务帐户。服务帐户与 Kafka Connect 本身不同。在这个发行版本前，构建在 default 服务帐户下运行。在指定身份验证和访问权限时，拥有自己的身份非常有用。

如果将标准 HTTP 代理（HTTP_**PROXY、HTTPS_PROXY 和 NO_PROXY** ）设置为 AMQ Streams 部署的环境变量，Kafka Connect 构建现在也可以在代理后面正常工作。

请参阅：

- 使用 AMQ Streams 自动创建新容器镜像
- **构建** 架构参考

3.3. KUBERNETES 配置提供程序用于外部配置数据

使用 *Kubernetes Configuration Provider* 插件，从外部来源加载配置数据。您可以从 OpenShift Secret 或 ConfigMap 加载数据。

该供应商独立于 AMQ Streams 运行。它会在不需要重启 Kafka 组件的情况下加载数据，即使使用新的 Secret 或 ConfigMap 时也是如此。

您可以使用它来加载所有 Kafka 组件的配置数据，包括生产者和使用者。例如，使用它为托管多个连接器的 Kafka Connect 实例提供凭据，而无需中断

请参阅 从外部源加载配置值。

3.4. 带有标记的日志过滤器

如果您使用 ConfigMap 为 AMQ Streams Operator 配置(log4j2)日志记录级别，现在可以定义日志记录过滤器来限制日志中返回的内容。您可以在 ConfigMap 中添加过滤器属性。

过滤器使用 *标记* 来指定要包含在日志中的内容。您可以为标记指定一个 kind、namespace 和 name。例如，如果 Kafka 集群失败，您可以通过将 kind 指定为 **Kafka** 来隔离日志，并使用失败集群的命名空间和名称。

本例显示了名为 **my-kafka-cluster** 的 Kafka 集群的标记过滤器。

基本日志记录过滤器配置

```
appender.console.filter.filter1.type=MarkerFilter (1)
appender.console.filter.filter1.onMatch=ACCEPT (2)
appender.console.filter.filter1.onMismatch=DENY (3)
```

```
appender.console.filter.filter1.marker=Kafka(my-namespace/my-kafka-cluster) 4
```

1. **MarkerFilter** 类型比较了指定的过滤标记。
2. 如果标志匹配，**onMatch** 属性接受日志。
3. 如果标志不匹配，则 **onMismatch** 属性将拒绝日志。
4. 用于过滤的标记格式为 *KIND(NAMESPACE/NAME-OF-RESOURCE)*。

请参阅在 Operator 中添加日志记录过滤器

3.5. OAUTH 2.0 身份验证增强

配置使用者和范围

现在，您可以在从授权服务器获取令牌时配置 **clientAudience 和 clientScope** 属性。属性值作为 **使用者 和范围 参数**传递给 令牌端点。这两个属性都在 **Kafka** 自定义资源中的 OAuth 2.0 身份验证监听程序配置中配置。

在以下情况下使用这些属性：

- 获取用于内部代理身份验证的访问令牌时
- 在基于 PLAIN 客户端身份验证的 OAuth 2.0 客户端名称中，使用 **clientId** 和 **secret**
  具体来说，当 PLAIN 回调首次与授权服务器交换 **clientID** （作为用户名）和 **机密** （作为密码）以获取访问令牌时，现在请求中可以包含 **使用者 和范围**。

这些属性会影响客户端是否可以获取令牌和令牌的内容。它们不会影响侦听器实施的令牌验证规则。

clientAudience 和 client Scope 属性的配置示例

```
# ...
  authentication:
    type: oauth
    # ...
    clientAudience: AUDIENCE
    clientScope: SCOPE
```

授权服务器有时在 JWT 访问令牌中提供 **提示（严重）**声明。当启用用户检查时，Kafka 代理会拒绝在 A**ud** 声明中不包含代理 **客户端Id** 的令牌。要启用受众检查，请在 **oauth** 监听程序配置中将 **checkAudience** 选项设置为 **true**。默认情况下禁用受众检查。

请参阅 为 Kafka 代理和 KafkaListenerAuthenticationOAuth 模式配置 OAuth 2.0 支持
https://access.redhat.com/documentation/en-us/red_hat_amq/2021.q3/html-single/using_amq_streams_on_openshift/index#type-KafkaListenerAuthenticationOAuth-reference

为 Kafka Connect 和 Kafka Bridge 指定使用者

现在，您可以在为 Kafka Connect 或 Kafka Bridge 配置相应的自定义资源中指定 OAuth 客户端身份验证时的使用者 选项。在以前的版本中，只支持这些资源的 **scope** 选项。

请参阅 KafkaClientAuthenticationOAuth 模式参考

OAuth 2.0 over PLAIN 不需要令牌端点

在通过 PLAIN 身份验证对 OAuth 2.0 使用"客户端 ID 和 secret"方法时，不再需要 **tokenEndpointUri** 选项。

指定了令牌端点 URI 的 OAuth 2.0 over PLAIN 配置示例

```
# ...
  authentication:
    type: oauth
    # ...
    enablePlain: true
    tokenEndpointUri: https://OAUTH-SERVER-ADDRESS/auth/realms/external/protocol/openid-
connect/token
```

如果没有指定 **tokenEndpointUri**，侦听器将处理：

- **用户名** 参数作为帐户名称
- 作为原始访问令牌 **的密码** 参数，传递给授权服务器进行验证（与 OAUTHBEARER 身份验证的作用相同）

OAuth 2.0 与 PLAIN 身份验证相比的"长生命访问令牌"方法的行为没有改变。使用此方法时不需要 **tokenEndpointUri**。

请参阅 OAuth 2.0 身份验证机制

3.6. 用户配额

通过 User Operator 处理用户配额不再由 ZooKeeper 管理。相反，用户配额通过 API 处理。

另外，增加了对 Kafka 变异速率配额的支持。此配额限制每秒允许的分区变异数量。配额可防止 Kafka 集群被并发主题操作所困扰。

分区变异的数量包括以下类型的用户请求：

- 为新主题创建分区
- 将分区添加到现有主题
- 从主题中删除分区

您可以配置变异率配额来控制用户请求接受变异的速度。费率从创建或删除的分区数中累积而来。

使用 **controllerMutationRate** 选项，将配额应用到 Kafka 用户。在本例中，每秒允许 10 个分区创建和删除操作。

使用用户配额的 KafkaUser 配置示例

```
apiVersion: kafka.strimzi.io/v1beta2
kind: KafkaUser
metadata:
  name: my-user
  labels:
    strimzi.io/cluster: my-cluster
spec:
  # ...
```

```
quotas:
  #...
  controllerMutationRate: 10 (1)
```

请参阅 用户配额

3.7. 暂停自定义资源协调

您可以暂停 AMQ Streams 操作器管理的自定义资源的协调，以执行修复或更新。您还可以暂停您要创建的自定义资源的协调。自定义资源会被创建，但会被忽略。

您可以为自定义资源添加注解来暂停它。

暂停协调的注解示例

```
oc annotate KIND-OF-CUSTOM-RESOURCE NAME-OF-CUSTOM-RESOURCE strimzi.io/pause-reconciliation="true"
```

现在，可以暂停 **KafkaTopic** 自定义资源的协调。

请参阅 暂停自定义资源协调

3.8. KAFKA EXPORTER 更新

AMQ Streams 提供的自定义版本的 Kafka Exporter 已更新至 1.3.1 版本。AMQ Streams 在提供的示例中包括 Kafka Exporter 的 Grafana 仪表板示例（**示例/metrics/grafana-dashboards/strimzi-kafka-exporter.json**）。

请参阅 添加 Kafka Exporter

3.9. KAFKA CONNECT 构建使用哈希来命名下载文件

您可以配置 **KafkaConnect** 资源，以创建自定义 Kafka Connect 容器镜像。使用 **spec.build** 配置可自动执行该过程。您可以配置 **插件** 来指定实施工件和 **输出** 来引用存储镜像的容器注册表。AMQ 流下载并将连接器插件添加到新容器镜像中。

构建过程现在使用 URL 哈希来命名下载的工件文件。在以前的版本中，它使用下载 URL 的最后一部分。如果您的插件工件需要特定名称，您可以使用一个新的工 **件** 类型及其 **fileName** 字段。

插件工件命名示例

```
apiVersion: kafka.strimzi.io/v1beta2
kind: KafkaConnect
metadata:
  name: my-connect-cluster
spec:
  #...
  build:
    output:
      #...
    plugins:
      - name: my-plugin
        artifacts:
```

```
        - type: other
          url: https://my-domain.tld/my-other-file.ext
          sha512sum: 589...ab4
          fileName: name-of-file.ext
#...
```

请参阅 **构建** 架构参考

第 4 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。

4.1. KAFKA 静态配额插件配置

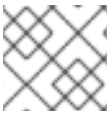
使用 *Kafka Static Quota* 插件，为 Kafka 集群中的代理设置吞吐量和存储限制。您可以通过配置 **Kafka** 资源启用插件并设置限值。您可以设置字节阈值和存储配额，对与代理交互的客户端设定限制。

Kafka 静态配额插件配置示例

```
apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
  name: my-cluster
spec:
  kafka:
    # ...
    config:
      client.quota.callback.class: io.strimzi.kafka.quotas.StaticQuotaCallback
      client.quota.callback.static.produce: 1000000
      client.quota.callback.static.fetch: 1000000
      client.quota.callback.static.storage.soft: 400000000000
      client.quota.callback.static.storage.hard: 500000000000
      client.quota.callback.static.storage.check-interval: 5
```

请参阅使用 Kafka 静态配额插件设置代理限制

4.2. 用于集群重新平衡的精简控制



注意

固态控制仍为技术预览，但有一些新的 增强功能。

您可以部署 Cruise Control，并使用它来在 CPU、磁盘、网络负载 *等上使用优化* 的 goals(pipeline)定义的限制来重新平衡 Kafka 集群。在均衡 Kafka 集群中，工作负载更加均匀地分布在代理 pod 中。

作为 **Kafka** 资源的一部分，Tithise Control 被配置和部署。您可以使用默认优化目标，或对其进行修改以符合您的要求。Cruise Control 的 YAML 配置文件示例在 example /**cruise-control/ 中提供**。

部署 Cruise Control 后，您可以创建 **KafkaRebalance** 自定义资源来：

- 从多个优化目标生成优化效果
- 基于优化建议重新平衡 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 用于集群重新平衡的 Cruise Control

4.2.1. 技术预览的改进

刷新优化建议

现在，您可以重复使用状态为 **Ready** 的现有 **KafkaRebalance** 资源，这表示集群重新平衡成功完成。您可以重复使用 **KafkaRebalance** 资源中定义的优化目标，或更改目标。

刷新优化建议：

1. 检查 **KafkaRebalance** 资源的状态：

```
oc describe kafkarebalance REBALANCE-NAME
```

2. 应用 **strimzi.io/rebalance=refresh** 注解：

```
oc annotate kafkarebalance REBALANCE-NAME strimzi.io/rebalance=refresh
```

cruise Control 会刷新优化建议来反映 Kafka 集群的最新状态。

请参阅 Ap 验证的一个优化建议

查看优化时的代理负载

优化现在包括在概述状态之外，由 *代理负载* 组成。代理负载在 ConfigMap 中返回，显示每个 Kafka 代理负载的指标，包括 CPU 使用率、磁盘用量、网络输出率等。指标分为三个类别：

之前
: 应用优化建议前的当前值

后
: 应用优化建议后的预期值

差别
: 后值和之前值之间的区别

代理负载 ConfigMap 的名称与 **KafkaRebalance** 资源相同。指标编码为 JSON 字符串。若要以人类可读的格式查看它们，请使用 **jq** 或 similiar JSON 解析器。例如：

```
oc get configmap MY-REBALANCE -o json | jq '.["data"]["brokerLoad.json"]|fromjson|.'
```

请参阅 优化概述

第 5 章 已弃用的功能

本发行版本中弃用的功能，以及之前的 AMQ Streams 版本中所支持的功能如下所示。

5.1. 通过 SOURCE-TO-IMAGE(S2I)进行 KAFKA 连接.

随着 KafkaConnect 资源引入 **构建配置**，AMQ Streams 现在可以使用数据连接所需的连接器插件自动构建容器镜像。

因此，已弃用了对使用 Source-to-Image(S2I)的 Kafka Connect 的支持。

要准备此更改，您可以将 Kafka Connect S2I 实例迁移到 Kafka Connect 实例。

请参阅 Migrating from Kafka Connect with S2I to Kafka Connect

5.2. ENABLEECDSA 属性

在 Kafka 代理的 **oauth** 监听程序配置中弃用了 **enableECDSA** 属性。此属性的值现在被忽略。

ECDSA 加密通过 Java 加密扩展(JCE)提供。Bouncy Castle Crypto 库不再与 AMQ Streams 打包。

5.3. 包含语言

KafkaMirrorMaker2 资源中的 topic **Black listPattern 和 groupsBlacklistPattern** 属性已弃用。这些属性将被 **主题ExcludePattern 和 groupsExcludePat tern** 删除和替换。

KafkaMirrorMaker 资源中的 **whitelist** 属性已弃用。该属性将替换为 **include：**

5.4. 插件工件文件的 KAFKA CONNECT 命名

Kafka Connect 构建过程现在使用 URL 哈希来命名下载的工件文件。在以前的版本中，它使用下载 URL 的最后一部分。

请参阅 Kafka Connect 构建使用哈希来命名下载文件

5.5. 弃用和删除 KAFKA 功能

本节提前通知 Apache Kafka 项目中的重要弃用和删除。

5.5.1. 计划在 Kafka 版本 3.0 中删除

Kafka 版本 3.0 将随 AMQ Streams 下一个主发行版本一起提供。

下表显示了在 Kafka 2.x 或更早版本中弃用且将在 Kafka 3.0 **中删除** 的方法和组件。这份清单并非详尽。

表 5.1. 弃用了将在 Kafka 3.0 中删除的 API 方法和组件

API 或组件	问题链接	描述
管理 API	KAFKA-12581	删除已弃用的 Admin.electPreferredLeaders

API 或组件	问题链接	描述
管理 API	KAFKA-6987	使用 CompletableFuture 重新实施 KafkaFuture（弃用 KafkaFuture.Function）
管理客户端	KAFKA-12577	删除已弃用的 **ConfigEntry** 构造器
所有客户端	KAFKA-12579	从客户端的 3.0 中删除各种弃用方法
所有客户端	KAFKA-12600	删除客户端配置客户端. **dns.lookup**的已弃用配置值 **默认值**
所有客户端	KAFKA-12578	删除已弃用的安全类/methods
broker	KAFKA-12591	删除已弃用的 **quota.producer.default** 和 **quota.consumer.default** 配置
broker	KAFKA-12592	删除已弃用的 LogConfig.Compact
broker	KAFKA-12590	Remove deprecated SimpleAclAuthorizer
broker	KAFKA-5905	删除 PrincipalBuilder 和 DefaultPrincipalBuilder
common	KAFKA-12573	删除了 deprecated **Metric#value**
使用者 API	KAFKA-12637	删除已弃用的 PartitionAssignor 接口
连接 API	KAFKA-12482	删除已弃用的 rest.host.name 和 rest.port Connect worker 配置
连接 API	KAFKA-12945	删除 3.0 中的端口、host.name 和相关配置
连接 API	KAFKA-12717	删除内部转换器配置属性
Streams API	KAFKA-12574	弃用 eos-alpha

API 或组件	问题链接	描述
Streams API	KAFKA-12808	删除 StreamsMetrics 下已弃用的方法
Streams API	KAFKA-7606	从 StreamsResetter 中删除已弃用的选项
Streams API	KAFKA-12796	在 stream **-scala 下删除**已弃用的类
Streams API	KAFKA-12419	删除 3.0 中已弃用的 Kafka Streams API
Streams API	KAFKA-10434	删除 WindowStore 上已弃用的方法
Streams API	KAFKA-12449	删除已弃用的 WindowStore#put
Streams API	KAFKA-12813	删除 ProcessorContext 中已弃用的调度方法
Streams API	KAFKA-12809	删除 Stores 中已弃用的方法
Streams API	KAFKA-12814	删除已弃用的方法 StreamsConfig#getConsumerConfig
Streams API	KAFKA-12313	弃用 default.windowed.serde.inner.class 配置
Streams API	KAFKA-8372	删除已弃用的 RocksDB#compactRange API
Streams API	KAFKA-12584	删除已弃用的 **Sum** 和 **Total** 类
Streams API	KAFKA-12683	删除已弃用的"UsePreviousTimeOnInvalidTimeStamp"
Streams API	KAFKA-12810	Remove deprecated TopologyDescription.Source#topics
Streams API	KAFKA-12630	Remove deprecated KafkaClientSupplier#getAdminClient

API 或组件	问题链接	描述
Streams API	KAFKA-10046	弃用的 PartitionGrouper 配置会被忽略
Streams API	KAFKA-12633	Remove deprecated "TopologyTestDriver#pipeInput / readOutput"
Streams API	KAFKA-12441	删除已弃用的方法 StreamsBuilder#addGlobalStore
Streams API	KAFKA-12452	为 ProcessorContext#forward 删除已弃用的过载
Streams API	KAFKA-12450	从 ReadOnlyWindowStore 中删除已弃用的方法
Streams API	KAFKA-12880	删除 3.0 中已弃用的 Count 和 SampledTotal
Streams API	KAFKA-12451	删除 WindowStore 中基于长期读取操作的弃用注解
Streams API	KAFKA-12568	删除已弃用的 "KStream#groupBy/join", "Joined#named" overloads
Streams API	KAFKA-12849	将 TaskMetadata 迁移到与内部实现的接口
Streams API	KAFKA-7785	删除 PartitionGrouper 接口及其配置，并将 DefaultPartitionGrouper 移到内部软件包
Streams API	KAFKA-7106	从窗口定义中删除片段/segmentInterval
Streams API	KAFKA-8897	增加 RocksDB 版本
Streams API	KAFKA-12909	允许用户选择-inbeious left/outer stream-stream 加入改进
工具	KAFKA-8405	删除已弃用的 **kafka-preferred-replica-election** 命令
工具	KAFKA-12588	删除 shell 命令中已弃用的 --zookeeper

5.5.2. mirror Maker 1.0 计划删除 Kafka 版本 4.0

Kafka 版本 4.0 将在 AMQ Streams 未来的主发行版本中提供。

下表显示了将在 Kafka 3.0 中弃用并在 Kafka 4.0 **中删除** 的功能。

表 5.2. 将在 Kafka 3.0 中弃用并在 Kafka 4.0 中删除的组件

组件	发布链接	概述
镜像 Maker 1.0	KAFKA-12436	deprecate MirrorMaker v1

第 6 章 修复的问题

下表中显示了 AMQ Streams 1.8 中修复的问题。有关在 Kafka 2.8.0 中修复的问题的详情，请参考 Kafka 2.8.0 发行注记。

问题号	描述
ENTMQST-1529	使用大型文件时，FileStreamSourceConnector 将停止。
ENTMQST-2359	Kafka Bridge 不处理分配和订阅。
ENTMQST-2453	**kafka-exporter** pod 不会因任何原因重启。
ENTMQST-2459	运行 Kafka Exporter 会导致 CPU 使用率高。
ENTMQST-2511	微调健康检查，以在滚动更新期间停止 Kafka Exporter 重启。
ENTMQST-2777	如果未指定服务模板，则不会设置 ENTMQST-2777 自定义网桥标签。
ENTMQST-2974	更改 Kafka Connect 连接器的日志级别只能临时正常工作。

表 6.1. 修复了常见漏洞和风险(CVE)

问题号	描述
ENTMQST-1934	CVE-2020-9488 log4j：SMTP 附加程序 [amq-st-1] 中存在主机不匹配的证书验证不正确。
ENTMQST-2613	CVE-2020-13949 libthrift：在处理不可信有效负载 [amq-st-1] 时潜在的 DoS。
ENTMQST-2617	CVE-2021-21290 netty：通过本地系统临时目录 [amq-st-1] 提供信息。
ENTMQST-2647	CVE-2021-21295 netty：如果缺少验证 [amq-st-1]，则 HTTP/2 中可能存在的请求。
ENTMQST-2663	CVE-2021-27568 json-smart：未发生异常可能会导致崩溃或信息泄露 [amq-st-1]。
ENTMQST-2711	ENTMQST-2711 CVE-2021-21409 netty：请求通过 content-length 标头 [amq-st-1].

问题号	描述
ENTMQST-2821	CVE-2021-28168 jersey-common: jersey：本地信息通过系统临时目录 [amq-st-1] 披露。
ENTMQST-2867	CVE-2021-29425 commons-io: apache-commons-io： Apache Commons IO 2.2 到 2.6 [amq-st-1] 中的有限路径遍历.
ENTMQST-2908	ENTMQST-2908 CVE-2021-28165 jetty-server: jetty: 资源耗尽（在收到无效大型 TLS 帧 [amq-st-1] 时）。
ENTMQST-2909	CVE-2021-28164 jetty-server: jetty: Ambiguous 路径可以访问 WEB-INF [amq-st-1]。
ENTMQST-2910	CVE-2021-28163 jetty-server: jetty: Symlink 目录公开 webapp 目录内容 [amq-st-1]。
ENTMQST-2980	CVE-2021-28169 jetty-server: jetty: 对 ConcatServlet 和 WelcomeFilter 的请求可以访问 WEB-INF 目录 [amq-st-1] 中的保护资源。
ENTMQST-3023	CVE-2021-34428 jetty-server: jetty: SessionListener 可以防止会话被破坏的 logout [amq-st-1] 无效。

第 7 章 已知问题

本节列出了 AMQ Streams 1.8 的已知问题。

7.1. LOG4J 的 SMTP 附加程序

AMQ Streams 附带一个潜在的存在安全漏洞的 log4j 版本(**log4j-1.2.17.redhat-3**)。漏洞在于 SMTP 附加程序功能，其默认配置中 AMQ Streams 不使用该功能。

表 7.1. CVE 问题

问题号	描述
ENTMQST-1934	CVE-2020-9488 log4j：SMTP 附加程序 [amq-st-1] 中存在主机不匹配的证书验证不正确。

临时解决方案

如果您使用 SMTP 附加程序，请确保 **mail.smtp.ssl.checkserveridentity** 设置为 **true**。

7.2. AMQ STREAMS CLUSTER OPERATOR ON IPV6 集群中

AMQ Streams Cluster Operator 不在互联网协议版本 6(IPv6)集群中启动。

临时解决方案

这个问题有两个临时解决方案。

临时解决方案之一：设置 KUBERNETES_MASTER 环境变量

1. 显示 OpenShift Container Platform 集群的 Kubernetes master 节点地址：

```
oc cluster-info
Kubernetes master is running at MASTER-ADDRESS
# ...
```

   复制 master 节点的地址。

2. 列出所有 Operator 订阅：

```
oc get subs -n OPERATOR-NAMESPACE
```

3. 编辑 AMQ Streams 的订阅 资源：

```
oc edit sub amq-streams -n OPERATOR_NAMESPACE
```

4. 在 **spec.config.env** 中，添加 **KUBERNETES_MASTER** 环境变量，设置为 Kubernetes 主机节点的地址。例如：

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: amq-streams
```

```
  namespace: OPERATOR-NAMESPACE
spec:
  channel: amq-streams-1.8.x
  installPlanApproval: Automatic
  name: amq-streams
  source: mirror-amq-streams
  sourceNamespace: openshift-marketplace
  config:
    env:
    - name: KUBERNETES_MASTER
      value: MASTER-ADDRESS
```

5. 保存并退出编辑器。

6. 检查 **订阅是否已** 更新：

```
oc get sub amq-streams -n OPERATOR-NAMESPACE
```

7. 检查 Cluster Operator **Deployment** 是否已更新为使用新的环境变量：

```
oc get deployment CLUSTER-OPERATOR-DEPLOYMENT-NAME
```

解决方法二：禁用主机名验证

1. 列出所有 Operator 订阅：

```
oc get subs -n OPERATOR-NAMESPACE
```

2. 编辑 AMQ Streams **的订阅** 资源：

```
oc edit sub amq-streams -n OPERATOR_NAMESPACE
```

3. 在 **spec.config.env** 中，添加 **KUBERNETES_DISABLE_HOSTNAME_VERIFICATION** 环境变量，设置为 **true**。例如：

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: amq-streams
  namespace: OPERATOR-NAMESPACE
spec:
  channel: amq-streams-1.8.x
  installPlanApproval: Automatic
  name: amq-streams
  source: mirror-amq-streams
  sourceNamespace: openshift-marketplace
  config:
    env:
    - name: KUBERNETES_DISABLE_HOSTNAME_VERIFICATION
      value: "true"
```

4. 保存并退出编辑器。

5. 检查 **订阅是否已** 更新：

```
oc get sub amq-streams -n OPERATOR-NAMESPACE
```

6. 检查 Cluster Operator **Deployment** 是否已更新为使用新的环境变量：

```
oc get deployment CLUSTER-OPERATOR-DEPLOYMENT-NAME
```

第 8 章 支持的集成产品

AMQ Streams 1.8 支持与以下红帽产品集成：

Red Hat Single Sign-On 7.4 及更新的版本

提供 OAuth 2.0 身份验证和 OAuth 2.0 授权。

Red Hat 3scale API Management 2.6 及更新的版本

保护 Kafka Bridge 并提供额外的 API 管理功能。

Red Hat Debezium 1.5

监控数据库并创建事件流。

Red Hat Service Registry 2.0

为数据流提供集中存储服务模式。

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 1.8 文档。

其它资源

- Red Hat Single Sign-On 支持的配置
- Red Hat 3scale API Management 支持的配置
- Red Hat Debezium 支持的配置
- Red Hat Service Registry 支持的配置

第 9 章 重要链接

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

2021-12-18 13:39:53 +1000 修订