



Red Hat Streams for Apache Kafka 2.7

RHEL 上 Apache Kafka 2.7 的流发行注记

Red Hat Enterprise Linux 中 Apache Kafka 发行版本的主要新功能及变化信息

Red Hat Streams for Apache Kafka 2.7 RHEL 上 Apache Kafka 2.7 的流发行注记

Red Hat Enterprise Linux 中 Apache Kafka 发行版本的主要新功能及变化信息

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

本发行注记总结了 Apache Kafka 2.7 发行版本流中引入的新功能、功能增强和修复。

目录

第 1 章 APACHE KAFKA 的名称更改为 STREAMS 的通知	3
第 2 章 功能	4
2.1. KAFKA 3.7.0 支持	4
2.2. KRAFT：从基于 ZOOKEEPER 迁移到基于 KRAFT 的 KAFKA 集群的支持	4
2.3. KRAFT：基于 KRAFT 的集群的 KAFKA 升级	4
第 3 章 功能增强	6
3.1. KAFKA 3.7.0 的改进	6
3.2. KAFKA BRIDGE 文本格式	6
第 4 章 技术预览	7
4.1. KRAFT 模式	7
4.2. KAFKA 静态配额插件配置	7
第 5 章 已弃用的功能	8
5.1. RHEL 7 在 APACHE KAFKA 2.7.0 的 STREAMS 中被弃用	8
5.2. JAVA 11 在 APACHE KAFKA 2.7.0 的流中被弃用	8
5.3. KAFKA MIRRORMAKER 2 身份复制策略	8
5.4. KAFKA MIRRORMAKER 1	8
5.5. KAFKA BRIDGE SPAN 属性	8
第 6 章 修复的问题	10
第 7 章 已知问题	11
7.1. 以 FIPS 模式运行时的 JMX 身份验证	11
第 8 章 支持的配置	12
8.1. 支持的平台	12
8.2. 支持的客户端	12
8.3. 支持的 APACHE KAFKA 生态系统	12
8.4. 其他支持的功能	12
8.5. 存储要求	13
第 9 章 组件详情	14
第 10 章 支持的与红帽产品集成	16
10.1. RED HAT BUILD OF KEYCLOAK（以前称为 RED HAT SINGLE SIGN-ON）	16

第 1 章 APACHE KAFKA 的名称更改为 STREAMS 的通知

Apache Kafka 的流被重命名为 **Apache Kafka 的流**，作为品牌工作的一部分。这个变化旨在提高 Red Hat Enterprise Linux Kafka 产品的意识。在这个转换期间，您可能会遇到到 Apache Kafka 的旧名称流的引用。我们正在积极更新文档、资源和媒体，以反映新的名称。

第 2 章 功能

Apache Kafka 2.7 的流引入了本节中介绍的功能。

RHEL 上的 Apache Kafka 2.7 流基于 Apache Kafka 3.7.0。



注意

要查看本版本中解决的所有增强和错误，请参阅 [Apache Kafka JIRA 项目的流](#)。

2.1. KAFKA 3.7.0 支持

Apache Kafka 的流现在支持并使用 Apache Kafka 版本 3.7.0。仅支持由红帽构建的 Kafka 发行版本。

有关升级说明，请参阅以下指南中的 Apache Kafka 和 Kafka 升级流的说明：

- [以 KRaft 模式在 RHEL 上使用 Streams for Apache Kafka](#)
- [在带有 ZooKeeper 的 RHEL 上使用流 for Apache Kafka](#)

如需更多信息，请参阅 [Kafka 3.7.0 发行注记](#)。

Kafka 3.6.x 仅支持升级到 Apache Kafka 2.7 的流。建议您执行滚动更新以使用新的二进制文件。



注意

Kafka 3.7.0 提供对 KRaft 模式的访问，因此 Kafka 在使用 Raft 协议的情况下运行没有 ZooKeeper。

2.2. KRAFT：从基于 ZOOKEEPER 迁移到基于 KRAFT 的 KAFKA 集群的支持

KRaft 模式 Apache Kafka [是一个技术预览](#)，有一些限制，但本发行版本引入了一些支持 KRaft 的新功能。要使用 KRaft 支持，提供了一个新的使用指南：[在 RHEL 上使用 Streams for Apache Kafka in KRaft mode](#)。

如果您使用 ZooKeeper 在 Kafka 集群中进行元数据管理，您现在可以在 KRaft 模式中迁移到使用 Kafka。

在迁移过程中，您可以执行以下操作：

1. 将控制器节点的仲裁安装为节点池，该池取代了 ZooKeeper 来管理集群。
2. 通过将 `zookeeper.metadata.migration.enable` 标志设置为 `true`，在控制器配置中启用 KRaft 迁移。
3. 通过将 `zookeeper.metadata.migration.enable` 标志设置为 `false`，将代理切换为使用 KRaft 和控制器退出迁移模式。

请参阅 [迁移到 KRaft 模式](#)。

2.3. KRAFT：基于 KRAFT 的集群的 KAFKA 升级

现在支持 KRaft 到 KRaft 升级。您可以更新安装文件，然后配置并重启所有 Kafka 节点。然后，您将基于 KRaft 的 Kafka 集群升级到更新支持的 KRaft 元数据版本。

更新 KRaft 元数据版本

```
./bin/kafka-features.sh --bootstrap-server <broker_host>:<port> upgrade --metadata 3.7
```

请参阅[升级基于 KRaft 的 Kafka 集群](#)。

第 3 章 功能增强

Apache Kafka 2.7 的流添加了很多改进。

3.1. KAFKA 3.7.0 的改进

有关 Kafka 3.7.0 中引入的增强功能概述，请参阅 [Kafka 3.7.0 发行注记](#)。

3.2. KAFKA BRIDGE 文本格式

在执行制作者操作时，**POST** 请求必须提供 **Content-Type** 标头来指定 *生成的消息的嵌入式数据格式*。在以前的版本中，JSON 和二进制是记录和键值支持的格式。现在也可以使用 **文本格式**。

表 3.1. 支持的内容类型格式

嵌入式数据格式	Content-Type 标头
JSON	content-Type: application/vnd.kafka.json.v2+json
二进制	content-Type: application/vnd.kafka.binary.v2+json
文本	content-Type: application/vnd.kafka.text.v2+json

第 4 章 技术预览

Apache Kafka 2.7 的 Streams 中包含的技术预览功能。



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. KRAFT 模式

KRaft 模式作为技术预览提供。

目前，Apache Kafka Streams 中的 KRaft 模式有以下限制：

- 不支持从 KRaft 模式降级到使用 ZooKeeper。
- 不支持带有多个磁盘的 JBOD 存储。

4.2. KAFKA 静态配额插件配置

使用 *Kafka Static Quota* 插件的技术预览，在 Kafka 集群中的代理上设置吞吐量和存储限制。您可以设置字节阈值和存储配额，以在与代理交互的客户端上放置限制。

Kafka 静态配额插件配置示例

```
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[有关使用 Kafka 静态配额插件的代理设置限制](#)。

第 5 章 已弃用的功能

本发行版本中已弃用的功能，以及之前的 Apache Kafka Streams 版本支持的功能如下所示。

5.1. RHEL 7 在 APACHE KAFKA 2.7.0 的 STREAMS 中被弃用

在 Apache Kafka 2.7.0 的 Streams 中弃用了对 RHEL 7 的支持。使用 Java 11 的 RHEL 7 已被弃用，作为 [受支持的平台](#)。预期 Apache Kafka 2.7.0 的 Streams 将是支持 RHEL 7 的最后一个版本。

5.2. JAVA 11 在 APACHE KAFKA 2.7.0 的流中被弃用

Kafka 3.7.0 和 Streams for Apache Kafka 2.7.0 中弃用了对 Java 11 的支持。对于 Apache Kafka 组件（包括客户端）的所有流（包括客户端）将不支持 Java 11。

Apache Kafka 的流支持 Java 17。在开发新应用程序时使用 Java 17。计划将当前使用 Java 11 的任何应用程序迁移到 17。



注意

在 Apache Kafka 2.4.0 的 Streams 中删除了对 Java 8 的支持。如果您当前正在使用 Java 8，计划以同样的方式迁移到 Java 17。

5.3. KAFKA MIRRORMAKER 2 身份复制策略

身份复制策略是 MirrorMaker 2 的一个功能，用于覆盖远程主题的自动重命名。主题不会使用源集群的名称添加名称，而是保留其原始名称。此设置对主动/被动备份和数据迁移场景特别有用。

要实现身份复制策略，您必须在 MirrorMaker 2 配置中指定复制策略类 (**replication.policy.class**)。在以前的版本中，您可以指定位于 Apache Kafka **mirror-maker-2-extensions** 组件的 Streams 中的 **io.strimzi.kafka.connect.mirror.IdentityReplicationPolicy** 类。但是，这个组件现已弃用，并将在以后的版本中删除。因此，建议您更新您的实现以使用 Kafka 自己的复制策略类 (**org.apache.kafka.connect.mirror.IdentityReplicationPolicy**)。

请参阅在 [MirrorMaker 2 中使用流进行 Apache Kafka](#)。

5.4. KAFKA MIRRORMAKER 1

Kafka MirrorMaker 在两个或更多活跃 Kafka 集群之间复制数据，并在数据中心之间复制数据。Kafka MirrorMaker 1 在 Kafka 3.0.0 中已弃用，并将在 Kafka 4.0.0 中删除。MirrorMaker 2 将是唯一可用的版本。MirrorMaker 2 基于 Kafka Connect 框架，连接器管理集群之间的数据传输。

因此，在 Apache Kafka 的 Streams 中也弃用了 MirrorMaker 1。如果您使用 MirrorMaker 1（称为 Apache Kafka 文档中的 Streams 中的 *MirrorMaker*），请将 MirrorMaker 2 与 **IdentityReplicationPolicy** 类搭配使用。MirrorMaker 2 将复制的主题重命名为目标集群。**IdentityReplicationPolicy** 配置会覆盖自动重命名。使用它生成与 MirrorMaker 1 相同的主动/被动单向复制。

请参阅在 [MirrorMaker 2 中使用流进行 Apache Kafka](#)。

5.5. KAFKA BRIDGE SPAN 属性

以下 Kafka Bridge span 属性已弃用，在适用的情况下显示替换：

- `http.method` 被 `http.request.method` 替代
- `http.url` 被 `url.scheme`, `url.path`, 和 `url.query` 替代
- `messaging.destination` 替换为 `messaging.destination.name`
- `http.status_code` 替换为 `http.response.status_code`
- `messaging.destination.kind=topic` 没有替换

Kafka Bridge 使用 OpenTelemetry 进行分布式追踪。更改是内联的，并带有 OpenTelemetry 语义惯例的更改。这些属性将在以后的 Kafka Bridge 发行版本中删除

第 6 章 修复的问题

RHEL 上 Apache Kafka 2.7 的 Streams 中修复的问题。

有关 Kafka 3.7.0 中修复的问题的详情，请参考 [Kafka 3.7.0](#) 发行注记。

表 6.1. 修复的问题

问题号	描述
ENTMQST-5839	OAuth 问题修复： <code>oauth.fallback.username.prefix</code> 没有效果
ENTMQST-5753	使用多个 HTTP 请求中的不同嵌入式格式生成不会导致
ENTMQST-5504	在启用 KRaft 时添加对 Kafka 和 Strimzi 升级的支持
ENTMQST-3994	zookeeper 到 KRaft 迁移

表 6.2. 修复的常见漏洞和风险(CVE)

问题号	描述
ENTMQST-5886	CVE-2023-43642 安全漏洞在 snappy-java 中的 SnappyInputStream 中找到
ENTMQST-5885	CVE-2023-52428 Nimbus JOSE+JWT before 9.37.2
ENTMQST-5884	CVE-2022-4899 安全漏洞在 zstd v1.4.10 中找到
ENTMQST-5883	CVE-2021-24032 安全漏洞包括在 zstd 中
ENTMQST-5882	CVE-2024-23944 Apache ZooKeeper：持久监视器处理中的信息披露
ENTMQST-5881	CVE-2021-3520 在 lz4 中存在一个缺陷
ENTMQST-5835	CVE-2024-29025 netty-codec-http: 资源分配没有限制或 Throttling
ENTMQST-5646	CVE-2024-1023 vert.x: io.vertx/vertx-core: 内存泄漏，因为在 Vertx 中使用 Netty FastThreadLocal 数据结构
ENTMQST-5667	当 TCP 服务器配置了 TLS 和 SNI 支持时，CVE-2024-1300 vertx-core: io.vertx:vertx-core: leak

第 7 章 已知问题

本节列出了 RHEL 上 Apache Kafka 2.7 的 Streams 的已知问题。

7.1. 以 FIPS 模式运行时的 JMX 身份验证

当在启用了 JMX 身份验证的 FIPS 模式下运行 Apache Kafka 时，客户端可能会失败。要临时解决这个问题，在以 FIPS 模式运行时不要启用 JMX 身份验证。我们正在调查此问题，并在以后进行解决。

第 8 章 支持的配置

Apache Kafka 2.7 发行版本流支持的配置。

8.1. 支持的平台

以下平台测试了在 Red Hat Enterprise Linux (RHEL) 版本上运行的带有 Kafka 的 Apache Kafka 2.7 的 Streams 测试。

操作系统	架构	JVM
RHEL 7 (已弃用)	x86, amd64	Java 11 (已弃用)
RHEL 8 和 9	x86, amd64, ppc64le (IBM Power), s390x (IBM Z 和 IBM® LinuxONE), aarch64 (64-bit ARM)	Java 11 和 Java 17

平台使用 Open JDK 11 和 17 测试，但 RHEL 7 和 Java 11 在 Apache Kafka 2.7.0 的 Streams 中弃用。IBM JDK 被支持，但不在每个发行版本中定期测试。不支持 Oracle JDK 11。

8.2. 支持的客户端

对于 Apache Kafka，只支持由红帽构建的客户端库。目前，Apache Kafka 的流仅提供 Java 客户端库。

客户端使用 Open JDK 11 和 17 进行测试。

8.3. 支持的 APACHE KAFKA 生态系统

在 Apache Kafka 的 Streams 中，只支持从 Apache Software Foundation 直接发布的以下组件：

- Apache Kafka Broker
- Apache Kafka Connect
- Apache MirrorMaker
- Apache MirrorMaker 2
- Apache Kafka Java Producer, Consumer, Management client, 和 Kafka Streams
- Apache ZooKeeper



注意

Apache ZooKeeper 仅作为 Apache Kafka 的实现详情支持，不应出于其他目的进行修改。另外，分配给 ZooKeeper 节点的内核数或 vCPU 不包含在订阅合规计算中。换句话说，ZooKeeper 节点不会计算客户的订阅。

8.4. 其他支持的功能

- Kafka Bridge
- Sything Control
- 分布式追踪

另请参阅 [第 10 章 支持的与红帽产品集成](#)。

8.5. 存储要求

Apache Kafka 的流已使用块存储测试，并与 XFS 和 ext4 文件系统兼容，它们都通常与 Kafka 一起使用。文件存储选项（如 NFS）不兼容。

其他资源

有关 Apache Kafka 2.5 LTS 发行版本的流支持的配置的详情，请参考 [Apache Kafka 2.5 发行注记](#)。

第 9 章 组件详情

下表显示了 Apache Kafka 的每个流的组件版本。



注意

Operator、控制台和代理等组件仅适用于在 OpenShift 中使用 Streams for Apache Kafka。

Apache Kafka Streams	Apache Kafka	Strimzi Operators	Kafka Bridge	Oauth	Sything Control	控制台 (Console)	Proxy
2.7.0	3.7.0	0.40.0	0.28	0.15.0	2.5.128	0.1	0.5.1
2.6.0	3.6.0	0.38.0	0.27	0.14.0	2.5.128	-	-
2.5.1	3.5.0	0.36.0	0.26	0.13.0	2.5.123	-	-
2.5.0	3.5.0	0.36.0	0.26	0.13.0	2.5.123	-	-
2.4.0	3.4.0	0.34.0	0.25.0	0.12.0	2.5.112	-	-
2.3.0	3.3.1	0.32.0	0.22.3	0.11.0	2.5.103	-	-
2.2.2	3.2.3	0.29.0	0.21.5	0.10.0	2.5.103	-	-
2.2.1	3.2.3	0.29.0	0.21.5	0.10.0	2.5.103	-	-
2.2.0	3.2.3	0.29.0	0.21.5	0.10.0	2.5.89	-	-
2.1.0	3.1.0	0.28.0	0.21.4	0.10.0	2.5.82	-	-
2.0.1	3.0.0	0.26.0	0.20.3	0.9.0	2.5.73	-	-
2.0.0	3.0.0	0.26.0	0.20.3	0.9.0	2.5.73	-	-
1.8.4	2.8.0	0.24.0	0.20.1	0.8.1	2.5.59	-	-
1.8.0	2.8.0	0.24.0	0.20.1	0.8.1	2.5.59	-	-
1.7.0	2.7.0	0.22.1	0.19.0	0.7.1	2.5.37	-	-
1.6.7	2.6.3	0.20.1	0.19.0	0.6.1	2.5.11	-	-
1.6.6	2.6.3	0.20.1	0.19.0	0.6.1	2.5.11	-	-

Apache Kafka Streams	Apache Kafka	Strimzi Operators	Kafka Bridge	Oauth	Sything Control	控制台 (Console)	Proxy
1.6.5	2.6.2	0.20.1	0.19.0	0.6.1	2.5.11	-	-
1.6.4	2.6.2	0.20.1	0.19.0	0.6.1	2.5.11	-	-
1.6.0	2.6.0	0.20.0	0.19.0	0.6.1	2.5.11	-	-
1.5.0	2.5.0	0.18.0	0.16.0	0.5.0	-	-	-
1.4.1	2.4.0	0.17.0	0.15.2	0.3.0	-	-	-
1.4.0	2.4.0	0.17.0	0.15.2	0.3.0	-	-	-
1.3.0	2.3.0	0.14.0	0.14.0	0.1.0	-	-	-
1.2.0	2.2.1	0.12.1	0.12.2	-	-	-	-
1.1.1	2.1.1	0.11.4	-	-	-	-	-
1.1.0	2.1.1	0.11.1	-	-	-	-	-
1.0	2.0.0	0.8.1	-	-	-	-	-



注意

Strimzi 0.26.0 包括一个 Log4j 漏洞。产品中包含的版本已更新，以取决于不包含漏洞的版本。

第 10 章 支持的与红帽产品集成

Apache Kafka 2.7 的流支持与以下红帽产品集成：

红帽构建的 Keycloak

提供 OAuth 2.0 身份验证和 OAuth 2.0 授权。

有关这些产品可在 Apache Kafka 部署的 Streams 中引入的功能信息，请参阅产品文档。

10.1. RED HAT BUILD OF KEYCLOAK（以前称为 RED HAT SINGLE SIGN-ON）

Apache Kafka 的流通过红帽构建的 Keycloak [授权服务](#) 支持基于 OAuth 2.0 令牌的授权，从而提供对安全策略和权限的集中管理。



注意

Red Hat build of Keycloak 替换了 Red Hat Single Sign-On，它现在处于维护支持中。我们正努力更新文档、资源和媒体，以反映这种转换。在此期间，在 Apache Kafka 的 Streams 中使用 Single Sign-On 的内容也适用于使用 Keycloak 的红帽构建。

其他资源

- [Red Hat build of Keycloak 支持的配置](#)
- [Red Hat Single Sign-On 支持的配置\(maintenance\)](#)

更新于 2024-04-30