



Red Hat Streams for Apache Kafka 2.7

Release Notes for Streams for Apache Kafka 2.7 on RHEL

Highlights of what's new and what's changed with this release of Streams for Apache Kafka on Red Hat Enterprise Linux

Red Hat Streams for Apache Kafka 2.7 Release Notes for Streams for Apache Kafka 2.7 on RHEL

Highlights of what's new and what's changed with this release of Streams for Apache Kafka on Red Hat Enterprise Linux

Legal Notice

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.

Abstract

The release notes summarize the new features, enhancements, and fixes introduced in the Streams for Apache Kafka 2.7 release.

Table of Contents

CHAPTER 1. NOTIFICATION OF NAME CHANGE TO STREAMS FOR APACHE KAFKA	3
CHAPTER 2. FEATURES	4
2.1. KAFKA 3.7.0 SUPPORT	4
2.2. KRAFT: SUPPORT FOR MIGRATING FROM ZOOKEEPER-BASED TO KRAFT-BASED KAFKA CLUSTERS	4
2.3. KRAFT: KAFKA UPGRADES FOR THE KRAFT-BASED CLUSTERS	5
2.4. RHEL 7 NO LONGER SUPPORTED	5
CHAPTER 3. ENHANCEMENTS	6
3.1. KAFKA 3.7.0 ENHANCEMENTS	6
3.2. KAFKA BRIDGE TEXT FORMAT	6
CHAPTER 4. TECHNOLOGY PREVIEWS	7
4.1. KRAFT MODE	7
4.2. KAFKA STATIC QUOTA PLUGIN CONFIGURATION	7
CHAPTER 5. DEPRECATED FEATURES	8
5.1. JAVA 11 DEPRECATED IN STREAMS FOR APACHE KAFKA 2.7.0	8
5.2. KAFKA MIRRORMAKER 2 IDENTITY REPLICATION POLICY	8
5.3. KAFKA MIRRORMAKER 1	8
5.4. KAFKA BRIDGE SPAN ATTRIBUTES	8
CHAPTER 6. FIXED ISSUES	10
CHAPTER 7. KNOWN ISSUES	11
7.1. INCOMPATIBILITY WITH RHEL 7	11
7.2. JMX AUTHENTICATION WHEN RUNNING IN FIPS MODE	11
CHAPTER 8. SUPPORTED CONFIGURATIONS	12
8.1. SUPPORTED PLATFORMS	12
8.2. SUPPORTED CLIENTS	12
8.3. SUPPORTED APACHE KAFKA ECOSYSTEM	12
8.4. ADDITIONAL SUPPORTED FEATURES	13
8.5. STORAGE REQUIREMENTS	13
CHAPTER 9. COMPONENT DETAILS	14
CHAPTER 10. SUPPORTED INTEGRATION WITH RED HAT PRODUCTS	16
10.1. RED HAT BUILD OF KEYCLOAK (FORMERLY RED HAT SINGLE SIGN-ON)	16

CHAPTER 1. NOTIFICATION OF NAME CHANGE TO STREAMS FOR APACHE KAFKA

AMQ Streams is being renamed as **streams for Apache Kafka** as part of a branding effort. This change aims to increase awareness among customers of Red Hat's product for Apache Kafka. During this transition period, you may encounter references to the old name, AMQ Streams. We are actively working to update our documentation, resources, and media to reflect the new name.

CHAPTER 2. FEATURES

Streams for Apache Kafka 2.7 introduces the features described in this section.

Streams for Apache Kafka 2.7 on RHEL is based on Apache Kafka 3.7.0.



NOTE

To view all the enhancements and bugs that are resolved in this release, see the [Streams for Apache Kafka Jira project](#).

2.1. KAFKA 3.7.0 SUPPORT

Streams for Apache Kafka now supports and uses Apache Kafka version 3.7.0. Only Kafka distributions built by Red Hat are supported.

For upgrade instructions, see the instructions for Streams for Apache Kafka and Kafka upgrades in the following guides:

- [Using Streams for Apache Kafka on RHEL in KRaft mode](#)
- [Using Streams for Apache Kafka on RHEL with ZooKeeper](#)

Refer to the [Kafka 3.7.0](#) Release Notes for additional information.

Kafka 3.6.x is supported only for the purpose of upgrading to Streams for Apache Kafka 2.7. We recommend that you perform a rolling update to use the new binaries.



NOTE

Kafka 3.7.0 provides access to KRaft mode, where Kafka runs without ZooKeeper by utilizing the Raft protocol.

2.2. KRAFT: SUPPORT FOR MIGRATING FROM ZOOKEEPER-BASED TO KRAFT-BASED KAFKA CLUSTERS

KRaft mode in Streams for Apache Kafka is a [technology preview](#), with some limitations, but this release introduces a number of new features that support KRaft. To support using KRaft, a new using guide is available: [Using Streams for Apache Kafka on RHEL in KRaft mode](#) .

If you are using ZooKeeper for metadata management in your Kafka cluster, you can now migrate to using Kafka in KRaft mode.

During the migration, you do the following:

1. Install a quorum of controller nodes, which replaces ZooKeeper for management of your cluster.
2. Enable KRaft migration in the controller configuration by setting the **zookeeper.metadata.migration.enable** flag to **true**.
3. Enable KRaft migration in the brokers by setting the **zookeeper.metadata.migration.enable** flag to **true**.
4. Switch the brokers to using KRaft by adding a **broker** KRaft role and node ID.

5. Switch the controllers out of migration mode by removing the **zookeeper.metadata.migration.enable** property.

See [Migrating to KRaft mode](#).

2.3. KRAFT: KAFKA UPGRADES FOR THE KRAFT-BASED CLUSTERS

KRaft to KRaft upgrades are now supported. You update the installation files, then configure and restart all Kafka nodes. You then upgrade the KRaft-based Kafka cluster to a newer supported KRaft metadata version.

Updating the KRaft metadata version

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

See [Upgrading KRaft-based Kafka clusters](#).

2.4. RHEL 7 NO LONGER SUPPORTED

RHEL 7 is no longer supported. The decision was made due to [known incompatibility issues](#).

CHAPTER 3. ENHANCEMENTS

Streams for Apache Kafka 2.7 adds a number of enhancements.

3.1. KAFKA 3.7.0 ENHANCEMENTS

For an overview of the enhancements introduced with Kafka 3.7.0, refer to the [Kafka 3.7.0 Release Notes](#).

3.2. KAFKA BRIDGE TEXT FORMAT

When performing producer operations, **POST** requests must provide **Content-Type** headers specifying the *embedded data format* of the messages produced. Previously, JSON and binary were the supported formats for record and key values. It's now possible to also use **text** format.

Table 3.1. Supported content type formats

Embedded data format	Content-Type header
JSON	Content-Type: application/vnd.kafka.json.v2+json
Binary	Content-Type: application/vnd.kafka.binary.v2+json
Text	Content-Type: application/vnd.kafka.text.v2+json

CHAPTER 4. TECHNOLOGY PREVIEWS

Technology Preview features included with Streams for Apache Kafka 2.7.



IMPORTANT

Technology Preview features are not supported with Red Hat production service-level agreements (SLAs) and might not be functionally complete; therefore, Red Hat does not recommend implementing any Technology Preview features in production environments. This Technology Preview feature provides early access to upcoming product innovations, enabling you to test functionality and provide feedback during the development process. For more information about the support scope, see [Technology Preview Features Support Scope](#).

4.1. KRAFT MODE

KRaft mode is available as a technology preview.

Currently, the KRaft mode in Streams for Apache Kafka has the following limitations:

- Downgrading from KRaft mode to using ZooKeeper is not supported.
- JBOD storage with multiple disks is not supported.

4.2. KAFKA STATIC QUOTA PLUGIN CONFIGURATION

Use the technology preview of the *Kafka Static Quota* plugin to set throughput and storage limits on brokers in your Kafka cluster. You can set a byte-rate threshold and storage quotas to put limits on the clients interacting with your brokers.

Example Kafka Static Quota plugin configuration

```
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
```

See [Setting limits on brokers using the Kafka Static Quota plugin](#) .

CHAPTER 5. DEPRECATED FEATURES

The features deprecated in this release, and that were supported in previous releases of Streams for Apache Kafka, are outlined below.

5.1. JAVA 11 DEPRECATED IN STREAMS FOR APACHE KAFKA 2.7.0

Support for Java 11 is deprecated in Kafka 3.7.0 and Streams for Apache Kafka 2.7.0. Java 11 will be unsupported for all Streams for Apache Kafka components, including clients, in the future.

Streams for Apache Kafka supports Java 17. Use Java 17 when developing new applications. Plan to migrate any applications that currently use Java 11 to 17.



NOTE

Support for Java 8 was removed in Streams for Apache Kafka 2.4.0. If you are currently using Java 8, plan to migrate to Java 17 in the same way.

5.2. KAFKA MIRRORMAKER 2 IDENTITY REPLICATION POLICY

Identity replication policy is a feature used with MirrorMaker 2 to override the automatic renaming of remote topics. Instead of prepending the name with the source cluster's name, the topic retains its original name. This setting is particularly useful for active/passive backups and data migration scenarios.

To implement an identity replication policy, you must specify a replication policy class (**replication.policy.class**) in the MirrorMaker 2 configuration. Previously, you could specify the **io.strimzi.kafka.connect.mirror.IdentityReplicationPolicy** class included with the Streams for Apache Kafka **mirror-maker-2-extensions** component. However, this component is now deprecated and will be removed in the future. Therefore, it is recommended to update your implementation to use Kafka's own replication policy class (**org.apache.kafka.connect.mirror.IdentityReplicationPolicy**).

See [Using Streams for Apache Kafka with MirrorMaker 2](#).

5.3. KAFKA MIRRORMAKER 1

Kafka MirrorMaker replicates data between two or more active Kafka clusters, within or across data centers. Kafka MirrorMaker 1 was deprecated in Kafka 3.0.0 and will be removed in Kafka 4.0.0. MirrorMaker 2 will be the only version available. MirrorMaker 2 is based on the Kafka Connect framework, connectors managing the transfer of data between clusters.

As a result, MirrorMaker 1 has also been deprecated in Streams for Apache Kafka as well. If you are using MirrorMaker 1 (referred to as just *MirrorMaker* in the Streams for Apache Kafka documentation), use MirrorMaker 2 with the **IdentityReplicationPolicy** class. MirrorMaker 2 renames topics replicated to a target cluster. **IdentityReplicationPolicy** configuration overrides the automatic renaming. Use it to produce the same active/passive unidirectional replication as MirrorMaker 1.

See [Using Streams for Apache Kafka with MirrorMaker 2](#).

5.4. KAFKA BRIDGE SPAN ATTRIBUTES

The following Kafka Bridge span attributes are deprecated with replacements shown where applicable:

- **http.method** replaced by **http.request.method**

- **http.url** replaced by **url.scheme**, **url.path**, and **url.query**
- **messaging.destination** replaced by **messaging.destination.name**
- **http.status_code** replaced by **http.response.status_code**
- **messaging.destination.kind=topic** without replacement

Kafka Bridge uses OpenTelemetry for distributed tracing. The changes are inline with changes to OpenTelemetry semantic conventions. The attributes will be removed in a future release of the Kafka Bridge

CHAPTER 6. FIXED ISSUES

The issues fixed in Streams for Apache Kafka 2.7 on RHEL.

For details of the issues fixed in Kafka 3.7.0, refer to the [Kafka 3.7.0](#) Release Notes.

Table 6.1. Fixed issues

Issue Number	Description
ENTMQST-5839	OAuth issue fix: oauth.fallback.username.prefix had no effect
ENTMQST-5753	Producing with different embedded formats across multiple HTTP requests isn't honoured
ENTMQST-5504	Add support for Kafka and Strimzi upgrades when KRaft is enabled
ENTMQST-3994	ZooKeeper to KRaft migration

Table 6.2. Fixed common vulnerabilities and exposures (CVEs)

Issue Number	Description
ENTMQST-5886	CVE-2023-43642 flaw was found in SnappyInputStream in snappy-java
ENTMQST-5885	CVE-2023-52428 Nimbus JOSE+JWT before 9.37.2
ENTMQST-5884	CVE-2022-4899 vulnerability was found in zstd v1.4.10
ENTMQST-5883	CVE-2021-24032 flaw was found in zstd
ENTMQST-5882	CVE-2024-23944 Apache ZooKeeper: Information disclosure in persistent watcher handling
ENTMQST-5881	CVE-2021-3520 a flaw in lz4
ENTMQST-5835	CVE-2024-29025 netty-codec-http: Allocation of Resources Without Limits or Throttling
ENTMQST-5646	CVE-2024-1023 vert.x: io.vertx/vertx-core: memory leak due to the use of Netty FastThreadLocal data structures in Vertx
ENTMQST-5667	CVE-2024-1300 vertx-core: io.vertx:vertx-core: memory leak when a TCP server is configured with TLS and SNI support

CHAPTER 7. KNOWN ISSUES

This section lists the known issues for Streams for Apache Kafka 2.7 on RHEL.

7.1. INCOMPATIBILITY WITH RHEL 7

There are known incompatibility issues when using RHEL 7 with Kafka 3.7. As a result, RHEL 7 is no longer supported. The issues arise due to an outdated GCC (GNU Compiler Collection) version on RHEL 7, which is incompatible with the version of the RocksDB JNI library (**org.rocksdb:rocksdbjni:7.9.2**) required by Kafka 3.7.

RocksDB JNI version 7.9.2 requires a newer version of the GCC and the associated **libstdc++** library than what is available on RHEL 7. Snappy compression and Kafka Streams, which depend on RocksDB, will not function correctly on RHEL 7 due to these outdated libraries.

Recommendation

- Upgrade your clients and brokers running on RHEL 7 to RHEL 8 to ensure compatibility with Kafka 3.7 and the latest Streams for Apache Kafka features.
- If you wish to continue using RHEL 7, consider using Streams for Apache Kafka 2.5 LTS or 2.6.

7.2. JMX AUTHENTICATION WHEN RUNNING IN FIPS MODE

When running Streams for Apache Kafka in FIPS mode with JMX authentication enabled, clients may fail authentication. To work around this issue, do not enable JMX authentication while running in FIPS mode. We are investigating the issue and working to resolve it in a future release.

CHAPTER 8. SUPPORTED CONFIGURATIONS

Supported configurations for the Streams for Apache Kafka 2.7 release.

8.1. SUPPORTED PLATFORMS

The following platforms are tested for Streams for Apache Kafka 2.7 running with Kafka on the version of Red Hat Enterprise Linux (RHEL) stated.

Operating System	Architecture	JVM
RHEL 8 and 9	x86, amd64, ppc64le (IBM Power), s390x (IBM Z and IBM® LinuxONE), aarch64 (64-bit ARM)	Java 11 (deprecated) and Java 17

Platforms are tested with Open JDK 11 and 17, though Java 11 is deprecated in Streams for Apache Kafka 2.7.0. The IBM JDK is supported but not regularly tested against during each release. Oracle JDK 11 is not supported.

FIPS compliance

Streams for Apache Kafka 2.7.0 is designed for FIPS.

To check which versions of RHEL are approved by the National Institute of Standards and Technology (NIST), see the [Cryptographic Module Validation Program](#) on the NIST website.

8.2. SUPPORTED CLIENTS

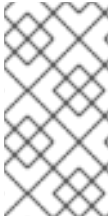
Only client libraries built by Red Hat are supported for Streams for Apache Kafka. Currently, Streams for Apache Kafka only provides a Java client library.

Clients are tested with Open JDK 11 and 17.

8.3. SUPPORTED APACHE KAFKA ECOSYSTEM

In Streams for Apache Kafka, only the following components released directly from the Apache Software Foundation are supported:

- Apache Kafka Broker
- Apache Kafka Connect
- Apache MirrorMaker
- Apache MirrorMaker 2
- Apache Kafka Java Producer, Consumer, Management clients, and Kafka Streams
- Apache ZooKeeper

**NOTE**

Apache ZooKeeper is supported solely as an implementation detail of Apache Kafka and should not be modified for other purposes. Additionally, the cores or vCPU allocated to ZooKeeper nodes are not included in subscription compliance calculations. In other words, ZooKeeper nodes do not count towards a customer's subscription.

8.4. ADDITIONAL SUPPORTED FEATURES

- Kafka Bridge
- Cruise Control
- Distributed Tracing

See also, [Chapter 10, Supported integration with Red Hat products](#) .

8.5. STORAGE REQUIREMENTS

Streams for Apache Kafka has been tested with block storage and is compatible with the XFS and ext4 file systems, both of which are commonly used with Kafka. File storage options, such as NFS, are not compatible.

Additional resources

For information on the supported configurations for the Streams for Apache Kafka 2.5 LTS release, see the [Streams for Apache Kafka 2.5 Release Notes](#) .

CHAPTER 9. COMPONENT DETAILS

The following table shows the component versions for each Streams for Apache Kafka release.

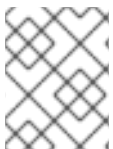


NOTE

Components like the operators, console, and proxy only apply to using Streams for Apache Kafka on OpenShift.

Streams for Apache Kafka	Apache Kafka	Strimzi Operators	Kafka Bridge	Oauth	Cruise Control	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	-	-

Streams for Apache Kafka	Apache Kafka	Strimzi Operators	Kafka Bridge	Oauth	Cruise Control	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	-	-	-	-	-



NOTE

Strimzi 0.26.0 contains a Log4j vulnerability. The version included in the product has been updated to depend on versions that do not contain the vulnerability.

CHAPTER 10. SUPPORTED INTEGRATION WITH RED HAT PRODUCTS

Streams for Apache Kafka 2.7 supports integration with the following Red Hat products:

Red Hat build of Keycloak

Provides OAuth 2.0 authentication and OAuth 2.0 authorization.

For information on the functionality these products can introduce to your Streams for Apache Kafka deployment, refer to the product documentation.

10.1. RED HAT BUILD OF KEYCLOAK (FORMERLY RED HAT SINGLE SIGN-ON)

Streams for Apache Kafka supports OAuth 2.0 token-based authorization through Red Hat build of Keycloak [Authorization Services](#), providing centralized management of security policies and permissions.



NOTE

Red Hat build of Keycloak replaces Red Hat Single Sign-On, which is now in maintenance support. We are working on updating our documentation, resources, and media to reflect this transition. In the interim, content that describes using Single Sign-On in the Streams for Apache Kafka documentation also applies to using the Red Hat build of Keycloak.

Additional resources

- [Red Hat build of Keycloak Supported Configurations](#)
- [Red Hat Single Sign-On Supported Configurations \(maintenance\)](#)

Revised on 2024-05-30 15:30:51 UTC