



Red Hat AMQ Streams 2.4

Release Notes for AMQ Streams 2.4 on RHEL

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

Red Hat AMQ Streams 2.4 Release Notes for AMQ Streams 2.4 on RHEL

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

Legal Notice

Copyright © 2023 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 AMQ Streams 2.4 release.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	3
CHAPTER 1. FEATURES	4
1.1. KAFKA 3.4.0 SUPPORT	4
1.2. SUPPORT FOR IBM Z AND LINUXONE ARCHITECTURE	4
1.2.1. Requirements for IBM Z and LinuxONE	4
1.2.2. Unsupported on IBM Z and LinuxONE	4
1.3. SUPPORT FOR IBM POWER ARCHITECTURE	5
1.3.1. Requirements for IBM Power	5
1.3.2. Unsupported on IBM Power	5
CHAPTER 2. ENHANCEMENTS	6
2.1. KAFKA 3.4.0 ENHANCEMENTS	6
2.2. OAUTH 2.0 CONFIGURATION FOR HTTP REQUESTS	6
CHAPTER 3. TECHNOLOGY PREVIEWS	8
3.1. OPENTELEMETRY FOR DISTRIBUTED TRACING	8
3.2. KRAFT MODE	8
3.3. KAFKA STATIC QUOTA PLUGIN CONFIGURATION	8
CHAPTER 4. DEPRECATED FEATURES	10
4.1. JAVA 8 SUPPORT REMOVED IN AMQ STREAMS 2.4.0	10
4.2. OPENTRACING	10
4.3. KAFKA MIRRORMAKER 2 IDENTITY REPLICATION POLICY	10
4.4. KAFKA MIRRORMAKER 1	10
CHAPTER 5. FIXED ISSUES	11
CHAPTER 6. KNOWN ISSUES	12
6.1. JMX AUTHENTICATION WHEN RUNNING IN FIPS MODE	12
CHAPTER 7. SUPPORTED INTEGRATION WITH RED HAT PRODUCTS	13
CHAPTER 8. IMPORTANT LINKS	14

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

CHAPTER 1. FEATURES

AMQ Streams 2.4 introduces the features described in this section.

AMQ Streams 2.4 on RHEL is based on Apache Kafka 3.4.0.



NOTE

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

1.1. KAFKA 3.4.0 SUPPORT

AMQ Streams now supports Apache Kafka version 3.4.0.

AMQ Streams uses Kafka 3.4.0. Only Kafka distributions built by Red Hat are supported.

For upgrade instructions, see [AMQ Streams and Kafka upgrades](#).

Refer to the [Kafka 3.4.0](#) Release Notes for additional information.



NOTE

Kafka 3.3.x is supported only for the purpose of upgrading to AMQ Streams 2.4.

For more information on supported versions, see the [AMQ Streams Component Details](#).

Kafka 3.4.0 uses ZooKeeper version 3.6.3, which is the same version as Kafka 3.3.x. We recommend that you perform a rolling update to use the new binaries.



NOTE

Kafka 3.4.0 provides access to KRaft mode, where Kafka runs without ZooKeeper by utilizing the Raft protocol. KRaft mode is available as a [Technology Preview](#).

1.2. SUPPORT FOR IBM Z AND LINUXONE ARCHITECTURE

AMQ Streams 2.4 is enabled to run on IBM Z and LinuxONE s390x architecture.

IBM Z and LinuxONE is tested for AMQ Streams running with Kafka and Open JDK 11, on Red Hat Enterprise Linux 8 and later.

1.2.1. Requirements for IBM Z and LinuxONE

- Red Hat Enterprise Linux 8 and later
- Open JDK 11

1.2.2. Unsupported on IBM Z and LinuxONE

- Red Hat Enterprise Linux 7
- Open JDK 8, Oracle JDK 8 & 11, IBM JDK 8

- FIPS mode

1.3. SUPPORT FOR IBM POWER ARCHITECTURE

AMQ Streams 2.4 is enabled to run on IBM Power ppc64le architecture.

IBM Power is tested for AMQ Streams running with Kafka and Open JDK 11, on Red Hat Enterprise Linux 8 and later.

1.3.1. Requirements for IBM Power

- Red Hat Enterprise Linux 8 and later
- Open JDK 11

1.3.2. Unsupported on IBM Power

- Red Hat Enterprise Linux 7
- Open JDK 8, Oracle JDK 8 & 11, IBM JDK 8

CHAPTER 2. ENHANCEMENTS

AMQ Streams 2.4 adds a number of enhancements.

2.1. KAFKA 3.4.0 ENHANCEMENTS

For an overview of the enhancements introduced with Kafka 3.4.0, refer to the [Kafka 3.4.0 Release Notes](#).

2.2. OAUTH 2.0 CONFIGURATION FOR HTTP REQUESTS

You can now use configuration to control HTTP requests to an OAuth 2.0 authorization server.

If you are creating a listener for your Kafka brokers that uses OAuth 2.0 authentication or authorization, you can add the following properties to the listener configuration:

- **httpRetries** to control the maximum number of times to retry a failed HTTP request to the authorization server.
- **httpRetryPauseMs** to control the time to wait in milliseconds before attempting another retry of a failed HTTP request to the authorization server.

Example listener configuration

```
sasl.enabled.mechanisms=OAUTHBEARER
listeners=CLIENT://0.0.0.0:9092
listener.security.protocol.map=CLIENT:SASL_PLAINTEXT
listener.name.client.sasl.enabled.mechanisms=OAUTHBEARER
sasl.mechanism.inter.broker.protocol=OAUTHBEARER
inter.broker.listener.name=CLIENT
listener.name.client.oauthbearer.sasl.server.callback.handler.class=io.strimzi.kafka.oauth.server.JaasServerOAuthValidatorCallbackHandler
listener.name.client.oauthbearer.sasl.jaas.config=org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule required ;
listener.name.client.oauthbearer.sasl.login.callback.handler.class=io.strimzi.kafka.oauth.client.JaasClientOAuthLoginCallbackHandler
listener.name.client.oauthbearer.sasl.jaas.config=org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule required \
# ...
  oauth.token.endpoint.uri="https://AUTH-SERVER-ADDRESS/auth/realms/REALM-NAME/protocol/openid-connect/token" \
  oauth.custom.claim.check="@.custom == 'custom-value" \
  oauth.scope="SCOPE" \
  oauth.check.audience="true" \
  oauth.audience="AUDIENCE" \
  oauth.valid.issuer.uri="https://https://AUTH-SERVER-ADDRESS/auth/REALM-NAME" \
  oauth.client.id="kafka-broker" \
  oauth.client.secret="kafka-broker-secret" \
  oauth.connect.timeout.seconds=60 \
  oauth.read.timeout.seconds=60 \
  oauth.http.retries=2 \
  oauth.http.retry.pause.millis=300 \
  oauth.groups.claim="$groups" \
  oauth.groups.claim.delimiter="," ;
```

See the following:

- [Configuring OAuth 2.0 support for Kafka brokers](#)
- [Configuring OAuth 2.0 authorization support](#)

CHAPTER 3. TECHNOLOGY PREVIEWS

Technology Preview features included with AMQ Streams 2.4.



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](#).

3.1. OPENTELEMETRY FOR DISTRIBUTED TRACING

OpenTelemetry for distributed tracing is available as a technology preview. You can use OpenTelemetry with a specified tracing system. OpenTelemetry is replacing OpenTracing for distributed tracing. [Support for OpenTracing is deprecated](#).

By Default, OpenTelemetry uses the OTLP (OpenTelemetry Protocol) exporter for tracing. AMQ Streams with OpenTelemetry is distributed for use with the Jaeger exporter, but you can specify other tracing systems supported by OpenTelemetry. AMQ Streams plans to migrate to using OpenTelemetry with the OTLP exporter by default, and is phasing out support for the Jaeger exporter.

See [Introducing distributed tracing](#).

3.2. KRAFT MODE

Apache Kafka is in the process of phasing out the need for ZooKeeper. You can now try deploying a Kafka cluster in KRaft (Kafka Raft metadata) mode without ZooKeeper as a technology preview.

CAUTION

This mode is intended **only** for development and testing, and must not be enabled for a production environment.

Currently, the KRaft mode in AMQ Streams has the following major limitations:

- Moving from Kafka clusters with ZooKeeper to KRaft clusters or the other way around is not supported.
- Upgrades and downgrades of Apache Kafka versions are not supported.
- SCRAM-SHA-512 authentication is not supported.
- JBOD storage with multiple disks is not supported.
- Many configuration options are still in development.

See [Running Kafka in KRaft mode](#) .

3.3. KAFKA STATIC QUOTA PLUGIN CONFIGURATION

Use 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= 4000000000000
client.quota.callback.static.storage.hard= 5000000000000
client.quota.callback.static.storage.check-interval= 5
```

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

CHAPTER 4. DEPRECATED FEATURES

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

4.1. JAVA 8 SUPPORT REMOVED IN AMQ STREAMS 2.4.0

Support for Java 8 was deprecated in AMQ Streams 2.0.0. Support for Java 8 is removed in AMQ Streams 2.4.0. This applies to all AMQ Streams components, including clients.

AMQ Streams supports Java 11. Use Java 11 when developing new applications. Plan to migrate any applications that currently use Java 8 to Java 11.

If you want to continue using Java 8 for the time being, AMQ Streams 2.2 provides Long Term Support (LTS). For information on the LTS terms and dates, see the [AMQ Streams LTS Support Policy](#).

4.2. OPENTRACING

Support for OpenTracing is deprecated.

The Jaeger clients are now retired and the OpenTracing project archived. As such, we cannot guarantee their support for future Kafka versions. We are introducing a new tracing implementation based on the OpenTelemetry project.

4.3. KAFKA MIRRORMAKER 2 IDENTITY REPLICATION POLICY

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

To implement an identity replication policy, you specify a replication policy class (**replication.policy.class**) in the MirrorMaker 2 configuration. The AMQ Streams **mirror-maker-2-extensions** component, which includes the **io.strimzi.kafka.connect.mirror.IdentityReplicationPolicy** class, 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**).

4.4. 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 AMQ Streams as well. If you are using MirrorMaker 1 (referred to as just *MirrorMaker* in the AMQ Streams 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 AMQ Streams with MirrorMaker 2](#).

CHAPTER 5. FIXED ISSUES

The issues fixed in AMQ Streams 2.4 on RHEL.

For details of the issues fixed in Kafka 3.4.0, refer to the [Kafka 3.4.0](#) Release Notes.

Table 5.1. Fixed issues

Issue Number	Description
ENTMQST-3757	[KAFKA] Mirror Maker 2 negative lag
ENTMQST-4107	[KAFKA] MM2 connector task stopped and didn't result in failed state
ENTMQST-4109	[KAFKA] Confusing error in MM2 when offsets for a group cannot be synced

Table 5.2. Fixed common vulnerabilities and exposures (CVEs)

Issue Number	Description
ENTMQST-4786	CVE-2022-42003 jackson-databind: deep wrapper array nesting when UNWRAP_SINGLE_VALUE_ARRAYS enabled
ENTMQST-4788	CVE-2022-42004 jackson-databind: use of deeply nested arrays
ENTMQST-4795	CVE-2023-25194: Apache Kafka: Possible RCE/Denial of service attack via SASL JAAS JndiLoginModule configuration using Kafka Connect
ENTMQST-4796	CVE-2020-36518 jackson-databind before 2.13.0 allows a Java StackOverflow exception and denial of service via a large depth of nested objects
ENTMQST-4797	CVE-2021-37137 Snappy frame decoder function doesn't restrict the chunk length which may lead to excessive memory usage
ENTMQST-4798	CVE-2021-37136 Bzip2 decompression decoder function doesn't allow setting size restrictions on the decompressed output data
ENTMQST-4799	CVE-2022-24823 Local information disclosure vulnerability in Netty
ENTMQST-4802	CVE-2022-36944 Scala 2.13.x before 2.13.9 has a Java deserialization risk via a gadget chain
ENTMQST-4803	CVE-2023-1370 JSON processor lib may cause stack exhaustion (stack overflow) due to recursive nesting of arrays/objects
ENTMQST-4804	CVE-2023-24815 Vert.x-Web apps serving files using StaticHandler on Windows issue

CHAPTER 6. KNOWN ISSUES

This section lists the known issues for AMQ Streams 2.4 on RHEL.

6.1. JMX AUTHENTICATION WHEN RUNNING IN FIPS MODE

When running AMQ Streams 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 7. SUPPORTED INTEGRATION WITH RED HAT PRODUCTS

AMQ Streams 2.4 supports integration with the following Red Hat products.

Red Hat Single Sign-On

Provides OAuth 2.0 authentication and OAuth 2.0 authorization.

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

Additional resources

- [Red Hat Single Sign-On Supported Configurations](#)

CHAPTER 8. IMPORTANT LINKS

- [AMQ Streams Supported Configurations](#)
- [AMQ Streams Component Details](#)

Revised on 2023-07-27 09:28:50 UTC