Red Hat AMQ 7.5

Release Notes for AMQ Streams 1.3 on RHEL

For use with AMQ Streams on Red Hat Enterprise Linux

Last Updated: 2022-06-29

Red Hat AMQ 7.5 Release Notes for AMQ Streams 1.3 on RHEL

For use with AMQ Streams on Red Hat Enterprise Linux

Legal Notice

Abstract

These release notes contain the latest information about new features, enhancements, fixes, and issues contained in the AMQ Streams 1.3 release.

Table of Contents

CHAPTER 1. FEATURES 3
1.1. KAFKA 2.3.0 SUPPORT 3

CHAPTER 2. ENHANCEMENTS 4
2.1. KAFKA 2.3.0 ENHANCEMENTS 4

CHAPTER 3. TECHNOLOGY PREVIEWS 5
3.1. DISTRIBUTED TRACING WITH JAEGER 5
3.2. OAUTH 2.0 AUTHENTICATION 6

CHAPTER 4. FIXED ISSUES 8

CHAPTER 5. KNOWN ISSUES 9

CHAPTER 6. SUPPORTED INTEGRATION PRODUCTS 10

CHAPTER 7. IMPORTANT LINKS 11

CHAPTER 1. FEATURES

1.1. KAFKA 2.3.0 SUPPORT

AMQ Streams now supports Apache Kafka version 2.3.0.

AMQ Streams is based on Kafka 2.3.0. Only Kafka distributions built by Red Hat are supported.

For upgrade instructions, see AMQ Streams and Kafka upgrades .

Refer to the Kafka 2.2.1 and Kafka 2.3.0 Release Notes for additional information.

NOTE

Kafka 2.2.x is supported in AMQ Streams only for upgrade purposes.

For more information on supported versions, see the Red Hat AMQ 7 Component Details Page on the Customer Portal.

CHAPTER 2. ENHANCEMENTS

The enhancements added in this release are outlined below.

2.1. KAFKA 2.3.0 ENHANCEMENTS

For an overview of the enhancements introduced with Kafka 2.3.0, refer to the Kafka 2.3.0 Release Notes.

CHAPTER 3. TECHNOLOGY PREVIEWS

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 support scope, see Technology Preview Features Support Scope.

3.1. DISTRIBUTED TRACING WITH JAEGER

NOTE

This is a Technology Preview feature.

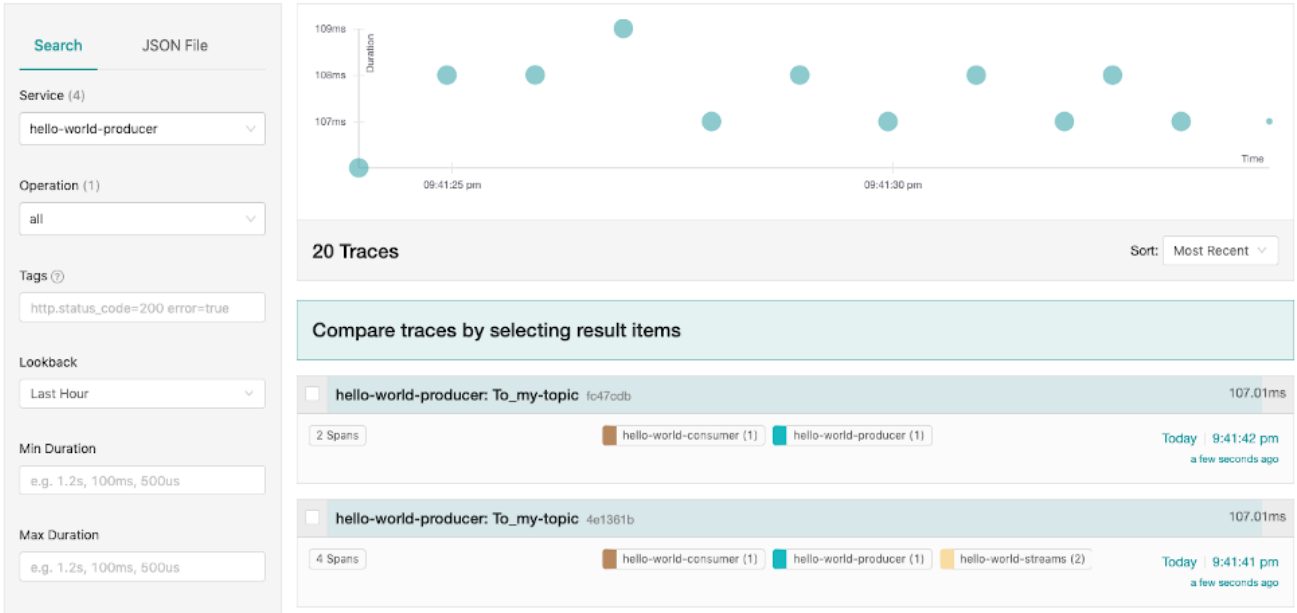
This release adds support for the distributed tracing of transactions within a typical Kafka architecture. Using an included OpenTracing Java library, you can instrument your client applications to generate traces for transactions, for example, producing and consuming messages.

Distributed tracing is supported in the following components:

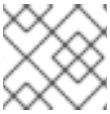
- Kafka clusters
- Producers and consumers
- Kafka Streams applications
- Mirror Maker
- Kafka Connect

Trace data is visualized in a user interface using Jaeger. You can use this information to monitor the operation of your Kafka cluster from end-to-end, and debug performance issues with target systems and applications.

An example of a query in the Jaeger user interface



3.2. OAUTH 2.0 AUTHENTICATION



NOTE

This is a Technology Preview feature.

AMQ Streams supports the use of OAuth 2.0 authentication using the *SASL OAUTHBEARER* mechanism.

Using OAuth 2.0 token based authentication, application clients can access resources on application servers (called 'resource servers') without exposing account credentials. The client presents an access token as a means of authenticating, which application servers can also use to find more information about the level of access granted. The authorization server handles the granting of access and inquiries about access.

In the context of AMQ Streams:

- Kafka brokers act as resource servers
- Kafka clients act as resource clients

The brokers and clients communicate with the OAuth 2.0 authorization server, as necessary, to obtain or validate access tokens.

For a deployment of AMQ Streams, OAuth 2.0 integration provides:

- Server-side OAuth 2.0 support for Kafka brokers
- Client-side OAuth 2.0 support for Kafka Mirror Maker, Kafka Connect and the Kafka Bridge

Red Hat Single Sign-On integration

You can deploy Red Hat Single Sign-On as an authorization server and configure it for integration with AMQ Streams.

You can use Red Hat Single Sign-On to:

- Configure authentication for Kafka brokers
- Configure and authorize clients
- Configure users and roles
- Obtain access and refresh tokens

See Using OAuth 2.0 token based authentication .

CHAPTER 4. FIXED ISSUES

The following table lists the fixed issues for AMQ Streams 1.3.

Issue Number	Description
ENTMQST-1287	The operation to create consumer should not contain HTTP code 500 in response
ENTMQST-1194	Unhandled exception when decoding invalid JSON
ENTMQST-1185	Fix consumerOrPartitionNotFound test

CHAPTER 5. KNOWN ISSUES

The following table lists the known issues for AMQ Streams 1.3.

Issue Number	Description
ENTMQST-1188	The logging level for the AMQ Streams Kafka Bridge is set to **DEBUG** by default. To reduce the size of the logs produced and improve performance, edit the **config/log4j.properties** file and set the logging level to **INFO**: `log4j.logger.io.strimzi.kafka.bridge=INFO, out` For more information, see Configuring loggers for the Kafka Bridge.

CHAPTER 6. SUPPORTED INTEGRATION PRODUCTS

AMQ Streams 1.3 supports integration with the following Red Hat products.

- **Red Hat Single Sign-On 7.3** for OAuth 2.0 support (as a Technology Preview)

For information on the functionality these products can introduce to your AMQ Streams deployment, refer to the AMQ Streams 1.3 documentation.

CHAPTER 7. IMPORTANT LINKS

- Red Hat AMQ 7 Supported Configurations
- Red Hat AMQ 7 Component Details

Revised on 2022-06-29 10:06:54 UTC