



Red Hat AMQ Streams 2.0

Release Notes for AMQ Streams 2.0 on RHEL

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

Red Hat AMQ Streams 2.0 Release Notes for AMQ Streams 2.0 on RHEL

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

Legal Notice

Copyright © 2022 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 2.0 release.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	3
CHAPTER 1. FEATURES	4
1.1. KAFKA 3.0.0 SUPPORT	4
1.2. SCALA UPGRADE	4
1.3. SUPPORT FOR POWERPC ARCHITECTURE	5
1.3.1. Requirements for IBM Power systems	5
1.3.2. Unsupported on IBM Power systems	5
CHAPTER 2. ENHANCEMENTS	6
2.1. KAFKA 3.0.0 ENHANCEMENTS	6
2.2. ENVIRONMENT VARIABLES CONFIGURATION PROVIDER FOR EXTERNAL CONFIGURATION DATA	6
CHAPTER 3. TECHNOLOGY PREVIEWS	7
3.1. KAFKA STATIC QUOTA PLUGIN CONFIGURATION	7
3.2. CRUISE CONTROL FOR CLUSTER REBALANCING	7
3.2.1. Enhancements to the Technology Preview	8
CHAPTER 4. DEPRECATED FEATURES	9
4.1. JAVA 8	9
4.2. KAFKA MIRRORMAKER 1	9
CHAPTER 5. FIXED ISSUES	10
5.1. FIXED ISSUES FOR AMQ STREAMS 2.0.1	10
5.2. FIXED ISSUES FOR AMQ STREAMS 2.0.0	10
CHAPTER 6. KNOWN ISSUES	12
CHAPTER 7. SUPPORTED INTEGRATION 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

The features added in this release, and that were not in previous releases of AMQ Streams, are outlined below.

The AMQ Streams 2.0.1 patch release is now available.



NOTE

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

1.1. KAFKA 3.0.0 SUPPORT

AMQ Streams now supports Apache Kafka version 3.0.0.

AMQ Streams uses Kafka 3.0.0. Only Kafka distributions built by Red Hat are supported.

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

Refer to the [Kafka 2.8.0](#) and [Kafka 3.0.0](#) Release Notes for additional information.



NOTE

Kafka 2.8.x is supported only for the purpose of upgrading to AMQ Streams 2.0.

For more information on supported versions, see the [Red Hat Streams for Apache Kafka Component Details](#) on the Customer Portal.

Kafka 3.0.0 requires ZooKeeper version 3.6.3. Therefore, you need to upgrade ZooKeeper when upgrading from AMQ Streams 1.8 to AMQ Streams 2.0, as described in the upgrade documentation.



WARNING

Kafka 3.0.0 provides early access to *self-managed mode*, where Kafka runs without ZooKeeper by utilizing the Raft protocol. **Note that self-managed mode is not supported in AMQ Streams.**

1.2. SCALA UPGRADE

AMQ Streams now uses Scala 2.13, which replaces Scala 2.12.

If you are using the Kafka Streams dependency in your Maven build, you must update it to specify Scala 2.13 and Kafka version 3.0.0. If the dependency is not updated, it will not build.

Kafka Streams dependency

```
<dependency>
  <groupId>org.apache.kafka</groupId>
```



```
<artifactId>kafka-streams-scala_2.13</artifactId>  
<version>3.0.0</version>  
</dependency>
```

See [Kafka Streams Scala dependency](#)

1.3. SUPPORT FOR POWERPC ARCHITECTURE

AMQ Streams 2.0 is enabled to run on PowerPC *ppc64le* architecture.

Support for IBM Power systems applies to AMQ Streams running with Kafka 3.0.0 and Open JDK 11 on Red Hat Enterprise Linux 8. The Kafka versions shipped with AMQ Streams 1.8 and earlier versions **do not contain** the ppc64 binaries.

1.3.1. Requirements for IBM Power systems

- Red Hat Enterprise Linux 8
- Open JDK 11
- Kafka 3.0.0

1.3.2. Unsupported on IBM Power systems

- Red Hat Enterprise Linux 7
- Kafka 2.8.0 or older
- Open JDK 8, Oracle JDK 8 & 11, IBM JDK 8
- AMQ Streams upgrades and downgrades since this is the first release on ppc64le

CHAPTER 2. ENHANCEMENTS

The enhancements added in this release are outlined below.

2.1. KAFKA 3.0.0 ENHANCEMENTS

For an overview of the enhancements introduced with Kafka 3.0.0, refer to the [Kafka 3.0.0 Release Notes](#).

2.2. ENVIRONMENT VARIABLES CONFIGURATION PROVIDER FOR EXTERNAL CONFIGURATION DATA

Use the Environment Variables Configuration Provider plugin to load configuration data from environment variables.

You can use the provider to load configuration data for all Kafka components, including producers and consumers. Use the provider, for example, to provide the credentials for Kafka Connect connector configuration.

You can use the Environment Variables Configuration Provider, for example, to load certificates or JAAS configuration using environment variables.

See [Loading configuration values from environment variables](#)

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. 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= 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](#)

3.2. CRUISE CONTROL FOR CLUSTER REBALANCING



NOTE

Cruise Control remains in Technology Preview, with some new [enhancements](#).

You can install [Cruise Control](#) and use it to rebalance your Kafka cluster using *optimization goals* – defined constraints on CPU, disk, network load, and more. In a balanced Kafka cluster, the workload is more evenly distributed across the broker pods.

Cruise Control helps to reduce the time and effort involved in running an efficient and balanced Kafka cluster.

A zipped distribution of Cruise Control is available for download from the [Customer Portal](#). To install Cruise Control, you configure each Kafka broker to use the provided Metrics Reporter. Then, you set Cruise Control properties, including optimization goals, and start Cruise Control using the provided script.

The Cruise Control server is hosted on a single machine for the whole Kafka cluster.

When Cruise Control is running, you can use the REST API to:

- Generate *dry run* optimization proposals from multiple optimization goals

- Initiate an optimization proposal to rebalance the Kafka cluster

Other Cruise Control features are not currently supported, including notifications, write-your-own goals, and changing the topic replication factor.

See [Cruise Control for cluster rebalancing](#)

3.2.1. Enhancements to the Technology Preview

Cruise Control version 2.5.73 is now available.

A zipped distribution of the latest version is available to download from the Red Hat Customer Portal.

See [Customer Portal](#)

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 for Java 8 was deprecated in Kafka 3.0.0 and AMQ Streams 2.0. Java 8 will be unsupported for all AMQ Streams components, including clients, in the future.

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.

4.2. KAFKA MIRRORMAKER 1

Kafka MirrorMaker replicates data between two or more active Kafka clusters, within or across data centers. Kafka MirrorMaker 1 is deprecated for Kafka 3.0.0 and will be removed in Kafka 4.0.0. MirrorMaker 2.0 will be the only version available. MirrorMaker 2.0 is based on the Kafka Connect framework, connectors managing the transfer of data between clusters.

As a consequence, the AMQ Streams **KafkaMirrorMaker** custom resource which is used to deploy Kafka MirrorMaker 1 has been deprecated. The **KafkaMirrorMaker** resource will be removed from AMQ Streams when Kafka 4.0.0 is adopted.

If you are using MirrorMaker 1 (referred to as just *MirrorMaker* in the AMQ Streams documentation), use the **KafkaMirrorMaker2** custom resource with the **IdentityReplicationPolicy**. MirrorMaker 2.0 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 [Kafka MirrorMaker 2.0 cluster configuration](#)

CHAPTER 5. FIXED ISSUES

The following sections list the issues fixed in AMQ Streams 2.0.x. Red Hat recommends that you upgrade to the latest patch release.

For details of the issues fixed in Kafka 3.0.0, refer to the [Kafka 3.0.0 Release Notes](#).

5.1. FIXED ISSUES FOR AMQ STREAMS 2.0.1

The AMQ Streams 2.0.1 patch release is now available.

For additional details about the issues resolved in AMQ Streams 2.0.1, see [AMQ Streams 2.0.x Resolved Issues](#).

Log4j vulnerabilities

AMQ Streams includes log4j 1.2.17. The release fixes a number of log4j vulnerabilities.

For more information on the vulnerabilities addressed in this release, see the following CVE articles:

- [CVE-2022-23307](#)
- [CVE-2022-23305](#)
- [CVE-2022-23302](#)
- [CVE-2021-4104](#)
- [CVE-2020-9488](#)
- [CVE-2019-17571](#)
- [CVE-2017-5645](#)

5.2. FIXED ISSUES FOR AMQ STREAMS 2.0.0

Log4j2 vulnerabilities

AMQ Streams includes log4j2 2.17.1. The release fixes a number of log4j2 vulnerabilities.

For more information on the vulnerabilities addressed in this release, see the following CVE descriptions:

- [CVE-2021-45046](#)
- [CVE-2021-45105](#)
- [CVE-2021-44832](#)
- [CVE-2021-44228](#)

Table 5.1. Fixed issues

Issue Number	Description
ENTMQST-3250	Changing log level does not seem to work in Kafka Exporter

Table 5.2. Fixed common vulnerabilities and exposures (CVEs)

Issue Number	Description
ENTMQST-3146	CVE-2021-34429 jetty-server: jetty: crafted URIs allow bypassing security constraints
ENTMQST-3307	CVE-2021-38153 Kafka: Timing attack vulnerability for Apache Kafka Connect and Clients
ENTMQST-3308	CVE-2021-38153 kafka-clients: Kafka: Timing attack vulnerability for Apache Kafka Connect and Clients
ENTMQST-3316	CVE-2021-37136 netty-codec: Bzip2Decoder doesn't allow setting size restrictions for decompressed data
ENTMQST-3317	CVE-2021-37137 netty-codec: SnappyFrameDecoder doesn't restrict chunk length and may buffer skippable chunks in an unnecessary way
ENTMQST-3532	CVE-2021-44228 log4j-core: Remote code execution in Log4j 2.x when logs contain an attacker-controlled string valuer
ENTMQST-3555	CVE-2021-45046 log4j-core: DoS in log4j2.x with thread context message pattern and context lookup pattern
ENTMQST-3587	CVE-2021-45105 log4j-core: DoS in log4j 2.x with Thread Context Map (MDC) input data contains a recursive lookup and context lookup pattern
ENTMQST-3602	CVE-2021-44832 log4j-core: remote code execution through JDBC Appender

CHAPTER 6. KNOWN ISSUES

There are no known issues for AMQ Streams 2.0 on RHEL.

CHAPTER 7. SUPPORTED INTEGRATION PRODUCTS

AMQ Streams 2.0 supports integration with the following Red Hat products.

Red Hat Single Sign-On 7.4 and later

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 AMQ Streams 2.0 documentation.

Additional resources

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

CHAPTER 8. IMPORTANT LINKS

- [Red Hat Streams for Apache Kafka Supported Configurations](#)
- [Red Hat Streams for Apache Kafka Component Details](#)

Revised on 2022-06-07 08:54:05 UTC