



Red Hat AMQ Streams 2.1

Release Notes for AMQ Streams 2.1 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 2.1 Release Notes for AMQ Streams 2.1 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 © 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 2.1 release.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	3
CHAPTER 1. FEATURES	4
1.1. KAFKA 3.1.0 SUPPORT	4
1.2. SUPPORT FOR IBM POWER ARCHITECTURE	4
1.2.1. Requirements for IBM Power	4
1.2.2. Unsupported on IBM Power	5
CHAPTER 2. ENHANCEMENTS	6
2.1. KAFKA 3.1.0 ENHANCEMENTS	6
2.2. OAUTH CONFIGURATION OPTIONS	6
CHAPTER 3. TECHNOLOGY PREVIEWS	7
3.1. KAFKA STATIC QUOTA PLUGIN CONFIGURATION	7
3.2. CRUISE CONTROL FOR CLUSTER REBALANCING	7
CHAPTER 4. DEPRECATED FEATURES	9
4.1. JAVA 8	9
4.2. KAFKA MIRRORMAKER 1	9
4.3. IDENTITY REPLICATION POLICY	9
CHAPTER 5. FIXED ISSUES	10
CHAPTER 6. KNOWN ISSUES	11
CHAPTER 7. SUPPORTED INTEGRATION PRODUCTS	12
CHAPTER 8. IMPORTANT LINKS	13

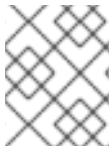
MAKING OPEN SOURCE MORE INCLUSIVE

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 2.1 introduces the features described in this section.

AMQ Streams version 2.1 is based on Strimzi 0.28.x.



NOTE

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

1.1. KAFKA 3.1.0 SUPPORT

AMQ Streams now supports Apache Kafka version 3.1.0.

AMQ Streams uses Kafka 3.1.0. Only Kafka distributions built by Red Hat are supported.

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

Refer to the [Kafka 3.0.0](#) and [Kafka 3.1.0](#) Release Notes for additional information.



NOTE

Kafka 3.0.x is supported only for the purpose of upgrading to AMQ Streams 2.1.

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

Kafka 3.1.0 requires ZooKeeper version 3.6.3, which is the same version as Kafka 3.0.0. Therefore, you will not need to upgrade ZooKeeper when upgrading from AMQ Streams 2.0 to AMQ Streams 2.1.



WARNING

Kafka 3.1.0 provides 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. SUPPORT FOR IBM POWER ARCHITECTURE

AMQ Streams 2.1 is enabled to run on IBM Power *ppc64le* architecture.

Support for IBM Power applies to AMQ Streams running with Kafka 3.0.0 and later, 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 *ppc64le* binaries.

1.2.1. Requirements for IBM Power

- Red Hat Enterprise Linux 8
- Open JDK 11

- Kafka 3.0.0 and later

1.2.2. Unsupported on IBM Power

- Red Hat Enterprise Linux 7
- Kafka 2.8.0 and earlier
- Open JDK 8, Oracle JDK 8 & 11, IBM JDK 8

CHAPTER 2. ENHANCEMENTS

AMQ Streams 2.1 adds a number of enhancements.

2.1. KAFKA 3.1.0 ENHANCEMENTS

For an overview of the enhancements introduced with Kafka 3.1.0, refer to the [Kafka 3.1.0 Release Notes](#).

2.2. OAUTH CONFIGURATION OPTIONS

New OAuth configuration properties have been introduced to the OAuth authentication configuration.

The properties related to timeouts and extracting groups information.

Timeout properties

- **oauth.connect.timeout.seconds** specifies the maximum time in seconds to connect to an authorization server before a timeout.
- **oauth.read.timeout.seconds** specifies the maximum time in seconds to read from an authorization server before a timeout.

The default is sixty seconds for both.

Groups properties

- **oauth.groups.claim** specifies a JsonPath query to extract groups information from a JWT token or introspection endpoint response. Not set by default.
- **oauth.groups.claim.delimiter** specifies a delimiter to parse groups information when returned as a single delimited string. The default value is ',' (comma).

Example OAuth configuration for a Kafka broker listener

```
listener.name.client.oauthbearer.sasl.jaas.config=org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule required \  
# ...  
oauth.connect.timeout.seconds=60  
oauth.read.timeout.seconds=60  
oauth.groups.claim="$$.groups"  
oauth.groups.claim.delimiter=","
```

See [Rebalance performance tuning](#).

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

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 [AMQ Streams software downloads page](#). 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](#).

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 3.1.0 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](#).

4.3. 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.

The AMQ Streams Identity Replication Policy class (**io.strimzi.kafka.connect.mirror.IdentityReplicationPolicy**) is now deprecated and will be removed in the future. You can update to use Kafka's own Identity Replication Policy (**class org.apache.kafka.connect.mirror.IdentityReplicationPolicy**).

See [Kafka MirrorMaker 2.0 cluster configuration](#).

CHAPTER 5. FIXED ISSUES

The issues fixed in AMQ Streams 2.1 on RHEL are shown in the following table. For details of the issues fixed in Kafka 3.1.0, refer to the [Kafka 3.1.0 Release Notes](#).

Table 5.1. Fixed common vulnerabilities and exposures (CVEs)

Issue Number	Description
ENTMQST-2851	<i>CVE-2021-3520</i> lz4: memory corruption due to an integer overflow bug caused by memmove argument
ENTMQST-3631	<i>CVE-2021-43797</i> netty: control chars in header names may lead to HTTP request smuggling

CHAPTER 6. KNOWN ISSUES

There are no known issues for AMQ Streams 2.1 on RHEL.

CHAPTER 7. SUPPORTED INTEGRATION PRODUCTS

AMQ Streams 2.1 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

- <https://access.redhat.com/articles/6644711>
- [AMQ Streams Component Details](#)

Revised on 2022-06-07 09:00:36 UTC