Red Hat AMQ Streams 2.6

Release Notes for AMQ Streams 2.6 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 2.6 Release Notes for AMQ Streams 2.6 on RHEL

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

Legal Notice

Abstract

The release notes summarize the new features, enhancements, and fixes introduced in the AMQ Streams 2.6 release.

Table of Contents

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 2.6 introduces the features described in this section.

AMQ Streams 2.6 on RHEL is based on Apache Kafka 3.6.0.

> **NOTE**
>
> To view all the enhancements and bugs that are resolved in this release, see the AMQ Streams Jira project.

1.1. KAFKA 3.6.0 SUPPORT

AMQ Streams now supports and uses Apache Kafka version 3.6.0. Only Kafka distributions built by Red Hat are supported.

For upgrade instructions, see AMQ Streams and Kafka upgrades .

Refer to the Kafka 3.6.0 Release Notes for additional information.

Kafka 3.5.x is supported only for the purpose of upgrading to AMQ Streams 2.6.

Kafka 3.6.0 uses ZooKeeper version 3.8.3, which is a different version to Kafka 3.5.x. We recommend that you perform a rolling update to use the new binaries.
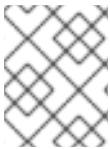
> **NOTE**
>
> Kafka 3.6.0 provides access to KRaft mode, where Kafka runs without ZooKeeper by utilizing the Raft protocol. KRaft mode is available as a Technology Preview.

CHAPTER 2. ENHANCEMENTS

AMQ Streams 2.6 adds a number of enhancements.

2.1. KAFKA 3.6.0 ENHANCEMENTS

For an overview of the enhancements introduced with Kafka 3.6.0, refer to the Kafka 3.6.0 Release Notes.

2.2. OAUTH 2.0 ACCEPT HEADER EXCLUDE OPTION

The **includeAcceptHeader** configuration property has been added as an OAuth 2.0 server-side and client-side configuration option. Some authorization servers have issues with clients sending **Accept: application/json** headers. By setting **oauth.include.accept.header="false"**, the header is not sent. The default setting is **true**.

See Using OAuth 2.0 token-based authentication and Using OAuth 2.0 token-based authorization .

CHAPTER 3. TECHNOLOGY PREVIEWS

Technology Preview features included with AMQ Streams 2.6.
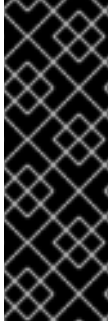
IMPORTANT

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. 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:

- Migrating from Kafka clusters with ZooKeeper to KRaft clusters is not recommended for production.

- Downgrading from KRaft mode to using ZooKeeper is not supported.

- JBOD storage with multiple disks is not supported.

- Many configuration options are still in development.

See Running Kafka in KRaft mode .

3.2. 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 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 Kafka 3.0.0 and AMQ Streams 2.0. Support for Java 8 was removed in AMQ Streams 2.4.0. This applies to all AMQ Streams components, including clients.

AMQ Streams supports Java 11 and Java 17. Use Java 11 or 17 when developing new applications. Plan to migrate any applications that currently use Java 8 to Java 11 or 17.

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 now removed.

The Jaeger clients are now retired and the OpenTracing project archived. As such, we cannot guarantee their support for future Kafka versions.

OpenTelemetry has replaced OpenTracing for distributed tracing.

See Setting up distributed tracing.

4.3. 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 AMQ Streams **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 AMQ Streams with MirrorMaker 2 .

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 2.6 on RHEL.

For details of the issues fixed in Kafka 3.6.0, refer to the Kafka 3.6.0 Release Notes.

Table 5.1. Fixed issues

Issue Number	Description
ENTMQST-5575	BackPort fix of ZOOKEEPER-4708 to AMQ Streams

Table 5.2. Fixed common vulnerabilities and exposures (CVEs)

Issue Number	Description
ENTMQST-4990	jackson-databind: denial of service via cylic dependencies
ENTMQST-4999	CVE-2023-33201 bouncycastle: potential blind LDAP injection attack using a self-signed certificate
ENTMQST-5023	netty: io.netty:netty-handler: SniHandler 16MB allocation
ENTMQST-5047	CVE-2023-2976 guava: insecure temporary directory creation
ENTMQST-5385	CVE-2023-44981 [2.6] CVE-2023-44981 zookeeper: zookeeper: Authorization Bypass in Apache ZooKeeper
ENTMQST-5139	CVE-2023-20873 spring-boot: Security Bypass With Wildcard Pattern Matching on Cloud Foundry
ENTMQST-5160	CVE-2022-46751 apache-ivy: XML External Entity vulnerability
ENTMQST-5173	CVE-2023-41080 tomcat: Open Redirect vulnerability in FORM authentication
ENTMQST-5293	CVE-2023-40167 jetty-http: jetty: Improper validation of HTTP/1 content-length
ENTMQST-5352	CVE-2023-42445 gradle: Possible local text file exfiltration by XML External entity injection
ENTMQST-5353	CVE-2023-44387 gradle: Incorrect permission assignment for symlinked files used in copy or archiving operations
ENTMQST-5398	CVE-2023-44981 zookeeper: Authorization Bypass in Apache ZooKeeper
ENTMQST-5427	CVE-2023-31582 jose4j: Insecure iteration count setting
ENTMQST-5437	CVE-2023-5072 in cruise-control

CHAPTER 6. KNOWN ISSUES

This section lists the known issues for AMQ Streams 2.6 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 CONFIGURATIONS

Supported configurations for the AMQ Streams 2.6 release.

7.1. SUPPORTED PLATFORMS

The following platforms are tested for AMQ Streams 2.6 running with Kafka on the version of Red Hat Enterprise Linux (RHEL) stated.

Operating System	Architecture	JVM
RHEL 7	x86, amd64	Java 11
RHEL 8 and 9	x86, amd64, ppc64le (IBM Power), s390x (IBM Z and IBM® LinuxONE), aarch64 (64-bit ARM)	Java 11 and Java 17

Platforms are tested with Open JDK 11 and 17. The IBM JDK is supported but not regularly tested against during each release. Open JDK 8, Oracle JDK 8 & 11, and IBM JDK 8 are not supported.

> **NOTE**
>
> Support for aarch64 (64-bit ARM) applies to AMQ Streams 2.6 when running Kafka 3.6.0 only.

7.2. SUPPORTED CLIENTS

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

Clients are tested with Open JDK 11 and 17.

7.3. SUPPORTED APACHE KAFKA ECOSYSTEM

In AMQ Streams, 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
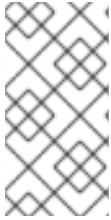
NOTE

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.

7.4. ADDITIONAL SUPPORTED FEATURES

- Kafka Bridge

- Cruise Control

- Distributed Tracing

See also, Chapter 9, *Supported integration with Red Hat products* .

7.5. STORAGE REQUIREMENTS

Kafka requires block storage; file storage options like NFS are not compatible.

Additional resources

For information on the supported configurations for the AMQ Streams 2.2 LTS release, see the AMQ Streams Supported Configurations article on the customer portal.

CHAPTER 8. COMPONENT DETAILS

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

AMQ Streams	Apache Kafka	Strimzi Operators	Kafka Bridge	Oauth	Cruise Control
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
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	–

AMQ Streams	Apache Kafka	Strimzi Operators	Kafka Bridge	Oauth	Cruise Control
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 9. SUPPORTED INTEGRATION WITH RED HAT PRODUCTS

AMQ Streams 2.6 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.

9.1. RED HAT SINGLE SIGN-ON

AMQ Streams supports the use of OAuth 2.0 token-based authorization through Red Hat Single Sign-On Authorization Services, which allows you to manage security policies and permissions centrally.

Additional resources

- Red Hat Single Sign-On Supported Configurations

Revised on 2023-12-19 17:46:56 UTC