Red Hat AMQ 2021.q3

Release Notes for AMQ Streams 1.8 on RHEL

For use with AMQ Streams on Red Hat Enterprise Linux

Last Updated: 2021-12-14

Red Hat AMQ 2021.q3 Release Notes for AMQ Streams 1.8 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 1.8 release.

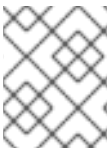
Table of Contents

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

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



NOTE

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

1.1. KAFKA 2.8.0 SUPPORT

AMQ Streams now supports Apache Kafka version 2.8.0.

AMQ Streams uses Kafka 2.8.0. Only Kafka distributions built by Red Hat are supported.

For upgrade instructions, see AMQ Streams and Kafka upgrades .

Refer to the Kafka 2.7.0 and Kafka 2.8.0 Release Notes for additional information.

NOTE

Kafka 2.7.x is supported only for the purpose of upgrading to AMQ Streams 1.8.

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

Kafka 2.8.0 requires ZooKeeper version 3.5.9. Therefore, you need to upgrade ZooKeeper when upgrading from AMQ Streams 1.7 to AMQ Streams 1.8, as described in the upgrade documentation.

WARNING

Kafka 2.8.0 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**.

CHAPTER 2. ENHANCEMENTS

The enhancements added in this release are outlined below.

2.1. KAFKA 2.8.0 ENHANCEMENTS

For an overview of the enhancements introduced with Kafka 2.8.0, refer to the Kafka 2.8.0 Release Notes.

2.2. OAUTH 2.0 AUTHENTICATION ENHANCEMENTS

Configure audience and scope

You can now configure the **oauth.audience** and **oauth.scope** properties and pass their values as parameters when obtaining a token. Both properties are configured in the OAuth 2.0 authentication listener configuration.

Use these properties in the following scenarios:

- When obtaining an access token for inter-broker authentication
- In the name of a client for OAuth 2.0 over PLAIN client authentication, using a **clientId** and **secret**

These properties affect whether a client can obtain a token and the content of the token. They do not affect token validation rules imposed by the listener.

Example configuration for oauth.audience and oauth.scope properties

```
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.scope=""SCOPE"" \
  oauth.audience="AUDIENCE" \
  oauth.check.audience="true" \
  # ...
```

Your authorization server might provide **aud** (audience) claims in JWT access tokens. When audience checks are enabled by setting **oauth.check.audience="true"**, the Kafka broker rejects tokens that do not contain the broker's **clientId** in their **aud** claims. Audience checks are disabled by default.

See Configuring OAuth 2.0 support for Kafka brokers

Token endpoint not required with OAuth 2.0 over PLAIN

The **oauth.token.endpoint.uri** parameter is no longer required when using the "client ID and secret" method for OAuth 2.0 over PLAIN authentication.

Example OAuth 2.0 over PLAIN listener configuration with token endpoint URI specified

```
listener.name.client.plain.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \
  oauth.valid.issuer.uri="https://__AUTH-SERVER-ADDRESS__" \
```

```
  oauth.jwks.endpoint.uri="https://__AUTH-SERVER-ADDRESS__/jwks" \
  oauth.username.claim="preferred_username"  \
  oauth.token.endpoint.uri="http://__AUTH_SERVER__/auth/realms/__REALM__/protocol/openid-
connect/token" ;
```

If the **oauth.token.endpoint.uri** is not specified, the listener treats the:

- **username** parameter as the account name
- **password** parameter as the raw access token, which is passed to the authorization server for validation (the same behavior as for OAUTHBEARER authentication)

The behavior of the "long-lived access token" method for OAuth 2.0 over PLAIN authentication is unchanged. The **oauth.token.endpoint.uri** is not required when using this method.

See OAuth 2.0 Kafka broker configuration

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 anomaly detection, 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 2.5.59 provides significant performance improvements, including 10% faster optimization proposal calculations.

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. DEPRECATED AND REMOVED KAFKA FEATURES

This section gives advance notice of important deprecations and removals in the Apache Kafka project.

4.1.1. Planned for removal in Kafka version 3.0

Kafka version 3.0 will be shipped with the next major release of AMQ Streams.

The following table shows methods and components that were deprecated in Kafka 2.x or earlier and will be **removed** in Kafka 3.0. This list is not exhaustive.

Table 4.1. Deprecated API methods and components that will be removed in Kafka 3.0

API or component	Issue link	Description
Admin API	KAFKA-12581	Remove deprecated Admin.electPreferredLeaders
Admin API	KAFKA-6987	Reimplement KafkaFuture with CompletableFuture (deprecate KafkaFuture.Function)
Admin client	KAFKA-12577	Remove deprecated **ConfigEntry** constructor
All clients	KAFKA-12579	Remove various deprecated methods from clients for 3.0
All clients	KAFKA-12600	Remove deprecated config value **default** for client config **client.dns.lookup**
All clients	KAFKA-12578	Remove deprecated security classes/methods
Broker	KAFKA-12591	Remove deprecated **quota.producer.default** and **quota.consumer.default** configurations
Broker	KAFKA-12592	Remove deprecated LogConfig.Compact
Broker	KAFKA-12590	Remove deprecated SimpleAclAuthorizer

API or component	Issue link	Description
Broker	KAFKA-5905	Remove PrincipalBuilder and DefaultPrincipalBuilder
Common	KAFKA-12573	Removed deprecated **Metric#value**
Consumer API	KAFKA-12637	Remove deprecated PartitionAssignor interface
Connect API	KAFKA-12482	Remove deprecated rest.host.name and rest.port Connect worker configs
Connect API	KAFKA-12945	Remove port, host.name, and related configs in 3.0
Connect API	KAFKA-12717	Remove internal converter config properties
Streams API	KAFKA-12574	Deprecate eos-alpha
Streams API	KAFKA-12808	Remove deprecated methods under StreamsMetrics
Streams API	KAFKA-7606	Remove deprecated options from StreamsResetter
Streams API	KAFKA-12796	Removal of deprecated classes under **streams-scala**
Streams API	KAFKA-12419	Remove deprecated APIs of Kafka Streams in 3.0
Streams API	KAFKA-10434	Remove deprecated methods on WindowStore
Streams API	KAFKA-12449	Remove deprecated WindowStore#put
Streams API	KAFKA-12813	Remove deprecated schedule method in ProcessorContext
Streams API	KAFKA-12809	Remove deprecated methods under Stores

API or component	Issue link	Description
Streams API	KAFKA-12814	Remove deprecated method StreamsConfig#getConsumerConfig
Streams API	KAFKA-12313	Deprecate the default.windowed.serde.inner.class configs
Streams API	KAFKA-8372	Remove deprecated RocksDB#compactRange API
Streams API	KAFKA-12584	Remove deprecated **Sum** and **Total** classes
Streams API	KAFKA-12683	Remove deprecated "UsePreviousTimeOnInvalidTimeStamp"
Streams API	KAFKA-12810	Remove deprecated TopologyDescription.Source#topics
Streams API	KAFKA-12630	Remove deprecated KafkaClientSupplier#getAdminClient
Streams API	KAFKA-10046	Deprecated PartitionGrouper config is ignored
Streams API	KAFKA-12633	Remove deprecated "TopologyTestDriver#pipeInput / readOutput"
Streams API	KAFKA-12441	Remove deprecated methods StreamsBuilder#addGlobalStore
Streams API	KAFKA-12452	Remove deprecated overloads for ProcessorContext#forward
Streams API	KAFKA-12450	Remove deprecated methods from ReadOnlyWindowStore
Streams API	KAFKA-12880	Remove deprecated Count and SampledTotal in 3.0
Streams API	KAFKA-12451	Remove deprecation annotation on long-based read operations in WindowStore

API or component	Issue link	Description
Streams API	KAFKA-12568	Remove deprecated "KStream#groupBy/join", "Joined#named" overloads
Streams API	KAFKA-12849	Migrate TaskMetadata to interface with internal implementation
Streams API	KAFKA-7785	Remove PartitionGrouper interface and it's config and move DefaultPartitionGrouper to internal package
Streams API	KAFKA-7106	Remove segment/segmentInterval from Window definition
Streams API	KAFKA-8897	Increase Version of RocksDB
Streams API	KAFKA-12909	Allow users to opt-into spurious left/outer stream-stream join improvement
Tools	KAFKA-8405	Remove deprecated **kafka-preferred-replica-election** command
Tools	KAFKA-12588	Remove deprecated --zookeeper in shell commands

4.1.2. Mirror Maker 1.0 planned for removal in Kafka version 4.0

Kafka version 4.0 will be shipped in a future major release of AMQ Streams.

The following table shows a feature that will be deprecated in Kafka 3.0 and **removed** in Kafka 4.0.

Table 4.2. Components that will be deprecated in Kafka 3.0 and removed in Kafka 4.0

Component	Link to issue	Summary
Mirror Maker 1.0	KAFKA-12436	deprecate MirrorMaker v1

CHAPTER 5. FIXED ISSUES

The following sections list the issues fixed in AMQ Streams 1.8.x. Red Hat recommends that you upgrade to the latest patch release

For details of the issues fixed in Kafka 2.8.0, refer to the Kafka 2.8.0 Release Notes .

5.1. FIXED ISSUES FOR AMQ STREAMS 1.8.4

The AMQ Streams 1.8.4 patch release is now available.

For additional details about the issues resolved in AMQ Streams 1.8.4, see AMQ Streams 1.8.x Resolved Issues.

Log4j2 vulnerability

The 1.8.4 release fixes a remote code execution vulnerability for AMQ Streams components that use log4j2. The vulnerability could allow a remote code execution on the server if the system logs a string value from an unauthorized source. This affects log4j versions between 2.0 and 2.14.1.

For more information, see CVE-2021-44228.

5.2. FIXED ISSUES FOR AMQ STREAMS 1.8.0

Table 5.1. Fixed issues

Issue Number	Description
ENTMQST-2453	The **kafka-exporter** pod restarts for no reason.
ENTMQST-2459	Running Kafka Exporter leads to high CPU usage.
ENTMQST-2511	Fine tune the health checks to stop Kafka Exporter restarting during rolling updates.
ENTMQST-1529	File Source Connector stops in the case of a large file.

Table 5.2. Fixed common vulnerabilities and exposures (CVEs)

Issue Number	Title	Description

Issue Number	Title	Description
ENTMQST-3023	CVE-2021-34428 jetty-server: jetty: SessionListener can prevent a session from being invalidated breaking logout.	A flaw was discovered in the jetty-server, where if an exception is thrown from the SessionListener#sessionDestroyed() method, then the session ID is not invalidated in the session ID manager. On deployments with clustered sessions and multiple contexts, this could result in a session not being invalidated and a shared-computer application being left logged in. The highest threat from this vulnerability is to data confidentiality and integrity.
ENTMQST-2980	CVE-2021-28169 jetty-server: jetty: requests to the ConcatServlet and WelcomeFilter are able to access protected resources within the WEB-INF directory.	-
ENTMQST-2711	CVE-2021-21409 netty: Request smuggling via content-length header.	A flaw was found in Netty. There is an issue where the content-length header is not validated correctly if the request uses a single Http2HeaderFrame with the endstream set to true. This flaw leads to request smuggling if the request is proxied to a remote peer and translated to HTTP/1.1. The highest threat from this vulnerability is to integrity.

Issue Number	Title	Description
ENTMQST-2663	CVE-2021-27568 json-smart: uncaught exception may lead to crash or information disclosure.	A flaw was found in json-smart. When an exception is thrown from a function, but is not caught, the program using the library may crash or expose sensitive information. The highest threat from this vulnerability is to data confidentiality and system availability. In OpenShift Container Platform (OCP), the Hive/Presto/Hadoop components that comprise the OCP Metering stack, ship the vulnerable version of json-smart package. Since the release of OCP 4.6, the Metering product has been deprecated [1], hence the affected components are marked as wontfix. This may be fixed in the future. [1] https://docs.openshift.com/container-platform/4.6/release_notes/ocp-4-6-release-notes.html#ocp-4-6-metering-operator-deprecated
ENTMQST-2647	CVE-2021-21295 netty: possible request smuggling in HTTP/2 due missing validation.	In Netty (io.netty:netty-codec-http2) before version 4.1.60.Final there is a vulnerability that enables request smuggling. If a Content-Length header is present in the original HTTP/2 request, the field is not validated by **Http2MultiplexHandler** as it is propagated up. This is fine as long as the request is not proxied through as HTTP/1.1. If the request comes in as an HTTP/2 stream, gets converted into the HTTP/1.1 domain objects (**HttpRequest**, **HttpContent**, etc.) via **Http2StreamFrameToHttpObjectCodec** and then sent up to the child channel's pipeline and proxied through a remote peer as HTTP/1.1 this may result in request smuggling.

Issue Number	Title	Description
ENTMQST-2617	CVE-2021-21290 netty: Information disclosure via the local system temporary directory.	In Netty there is a vulnerability on Unix-like systems involving an insecure temp file. When netty's multipart decoders are used, a local information disclosure can occur via the local system temporary directory if temporary storing uploads on the disk is enabled. On unix-like systems, the temporary directory is shared between all user. As such, writing to this directory using APIs that do not explicitly set the file/directory permissions can lead to information disclosure.
ENTMQST-2613	CVE-2020-13949 libthrift: potential DoS when processing untrusted payloads.	A flaw was found in libthrift. Applications using Thrift would not show an error upon receiving messages declaring containers of sizes larger than the payload. This results in malicious RPC clients with the ability to send short messages which would result in a large memory allocation, potentially leading to denial of service. The highest threat from this vulnerability is to system availability.
ENTMQST-1934	CVE-2020-9488 log4j: improper validation of certificate with host mismatch in SMTP appender.	-
ENTMQST-2910	CVE-2021-28163 jetty-server: jetty: Symlink directory exposes webapp directory contents.	If the **${jetty.base}** directory or the **${jetty.base}/webapps** directory is a symlink the contents of the **${jetty.base}/webapps** directory may be deployed as a static web application, exposing the content of the directory for download. The highest threat from this vulnerability is to data confidentiality.

Issue Number	Title	Description
ENTMQST-2909	CVE-2021-28164 jetty-server: jetty: Ambiguous paths can access WEB-INF.	In Jetty the default compliance mode allows requests with URIs that contain **%2e** or **%2e%2e** segments to access protected resources within the WEB-INF directory. An attacker can use this vulnerability to reveal sensitive information regarding the implementation of a web application.
ENTMQST-2908	CVE-2021-28165 jetty-server: jetty: Resource exhaustion when receiving an invalid large TLS frame.	When using SSL/TLS with Jetty, either with HTTP/1.1, HTTP/2, or WebSocket, the server may receive an invalid large (greater than 17408) TLS frame that is incorrectly handled, causing high CPU resources utilization. The highest threat from this vulnerability is to service availability.
ENTMQST-2867	CVE-2021-29425 commons-io: apache-commons-io: Limited path traversal in Apache Commons IO 2.2 to 2.6.	-
ENTMQST-2821	CVE-2021-28168 jersey-common: jersey: Local information disclosure via system temporary directory.	-

CHAPTER 6. KNOWN ISSUES

This section lists the known issues for AMQ Streams 1.8.

6.1. SMTP APPENDER FOR LOG4J

AMQ Streams ships with a potentially vulnerable version of log4j (**log4j-1.2.17.redhat-3**). The vulnerability lies with the SMTP appender functionality, which is not used by AMQ Streams in its default configuration.

Table 6.1. CVE issue

Issue Number	Description
ENTMQST-1934	CVE-2020-9488 log4j: improper validation of certificate with host mismatch in SMTP appender [amq-st-1].

Workaround

If you are using the SMTP appender, ensure that **mail.smtp.ssl.checkserveridentity** is set to **true**.

CHAPTER 7. SUPPORTED INTEGRATION PRODUCTS

AMQ Streams 1.8 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 1.8 documentation.

Additional resources

- Red Hat Single Sign-On Supported Configurations

CHAPTER 8. IMPORTANT LINKS

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

Revised on 2021-12-14 20:09:39 UTC